Fall 10-1-2021

# Practicality of Quantum Random Access Memory

Connor Hann
*Yale University Graduate School of Arts and Sciences*, chann1127@gmail.com

# Abstract

## Practicality of Quantum Random Access Memory

Connor T. Hann

2021

Quantum computers are expected to revolutionize the world of computing, but major challenges remain to be addressed before this potential can be realized. One such challenge is the so-called data-input bottleneck: Even though quantum computers can quickly solve certain problems by rapidly analyzing large data sets, it can be difficult to load this data into a quantum computer in the first place. In order to quickly load large data sets into quantum states, a highly-specialized device called a Quantum Random Access Memory (QRAM) is required. Building a large-scale QRAM is a daunting engineering challenge, however, and concerns about QRAM's practicality cast doubt on many potential quantum computing applications.

In this thesis, I consider the practical challenges associated with constructing a large-scale QRAM and describe how several of these challenges can be addressed. I first show that QRAM is surprisingly resilient to decoherence, such that data can be reliably loaded even in the presence of realistic noise. Then, I propose a hardware-efficient error suppression scheme that can further improve QRAM's reliability without incurring substantial additional overhead, in contrast to conventional quantum error-correction approaches. Finally, I propose experimental implementations of QRAM for hybrid quantum acoustic systems. The proposed architectures are naturally hardware-efficient and scalable, thanks to the compactness and high coherence of acoustic modes. Taken together, the results in this thesis both pave the way for small-scale, near-term experimental demonstrations of QRAM and improve the reliability and scalability of QRAM in the long term.

# Practicality of Quantum Random Access Memory

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Connor T. Hann

Dissertation Director: Liang Jiang, Steven M. Girvin

December, 2021

# Contents

# List of Figures

# List of Tables

# Acknowledgements

First, I would like thank my advisors, Liang Jiang and Steve Girvin. Both Liang and Steve have been incredible mentors to me over the last five years. When I started as a graduate student, I hardly knew anything about the fields of quantum information and quantum computing, but Liang and Steve were always incredibly patient and provided me with excellent guidance. I really appreciated the freedom that they afforded to me to pursue my own interests and the encouragement they provided along the way. I always left our meetings feeling excited about the research ahead.

I would also like to thank my other two committee members, Shruti Puri and Rob Schoelkopf. Shruti has been so incredibly generous in taking time to teach me about everything from cat qubits, to error correction, to quantum algorithms, and it was always enjoyable puzzling over problems together. Similarly, it was a great privilege to be able to work so closely with an experimental group like Rob's, and I'm grateful for all that Rob taught me about circuit QED and effective academic writing. I count myself as extremely fortunate to have been able to work so closely with both Shruti and Rob over the course of my PhD.

I want to thank all of my friends and collaborators at the Yale Quantum Institute, the University of Chicago, and the AWS Center for Quantum Computing. All of these communities have been so welcoming and supportive, and it was a pleasure to learn from and collaborate with such amazing groups of people. I also want to thank the administrators at each of these institutes for all that they do to create such wonderful communities.

Finally, I want to thank my friends and family. My friends at Yale and beyond have made the past five years memorable and enjoyable, and my parents and sisters always supported and encouraged me. I couldn't have done this without the love and support of my fiancé, Emily, and this thesis is dedicated to her.

# Chapter 1

# Introduction

## 1.1 Quantum computing

Quantum computers are devices that exploit quantum phenomena—such as superposition, entanglement, and interference—in order to perform computations. In principle, these quantum effects can be leveraged to solve certain problems much faster than is possible with conventional classical computers. For example, Shor's quantum factoring algorithm can factor large integers exponentially faster than the best-known classical algorithms [1]. The presumption that factoring large numbers is difficult underlies many modern-day cryptography schemes, including RSA encryption [2], so quantum computers are poised to have a significant impact on the field of cryptography as a result [3]. Similarly, quantum computers can be programmed to simulate the dynamics of other quantum systems, with simulation times exponentially faster than what is achievable classically [4]. The ability to efficiently simulate large quantum systems would be transformative for the fields of chemistry, physics, and materials science [5]. There exist numerous other examples of quantum algorithms that provide speedups over their classical counterparts [6].

Fig. 1.1 illustrates a typical workflow for solving some computational problem

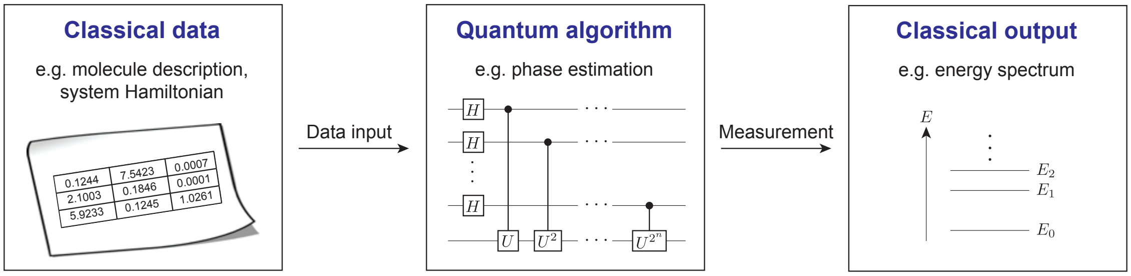


Figure 1.1: Typical quantum-computing workflow. A classical description of the problem is specified, and this classical data is input into a quantum processor. Then, a quantum algorithm is run, and measurements are performed to extract the desired result.

with a quantum computer. The problem-solving process beings with some classical data that constitutes a description of the problem to be solved. This data is fed into a quantum computer, where it serves as the input to an appropriate quantum algorithm. The quantum algorithm is executed, and finally the system is measured in order to extract a classical output that encodes the solution to the problem.

As an example, suppose that we wish to calculate the ground state energy of some complicated quantum system, such as an interacting many-body system or molecule. We begin by first constructing a classical description of this system. We enumerate the interactions between the various particles and their strengths, or, more generally, we write down the system's Hamiltonian. Next, this classical data is loaded into the quantum processor and used as input to the quantum phase estimation algorithm [7–9]. This algorithm can calculate the ground state energy in a time that scales only polynomially with the system size[1], in contrast to the exponential time required by classical algorithms. After the algorithm has been run, the solution to the problem—the ground state energy—is encoded in the state of the quantum processor. We extract this solution by measuring the qubits that comprise the processor.

[1] This assumes the ability to efficiently prepare the system in a state that has a large overlap with the ground state—a non-trivial assumption.

## 1.2 The data-input bottleneck

In practice, the process of loading classical data into a quantum processor can sometimes be quite difficult, a problem referred to as the data-input bottleneck. For example, there could be a very large amount of data to load, as is frequently the case in quantum algorithms for machine learning [10]. Alternatively, the quantum algorithm might require that the data is presented in a very particular form, e.g., encoded in the amplitudes of a quantum state (see Chapter 2).

Frequently, this input bottleneck problem is abstracted away by invoking a *quantum oracle* [11–13]. An oracle is a theoretical device that makes the input data accessible to the quantum computer in a suitable way, but the mechanism by which the oracle enables this access—how the oracle actually operates—is left unspecified. This abstraction can be quite useful, for example, when analyzing the complexity of quantum algorithms [12].

If quantum computers are to be used to solve problems faster than their classical counterparts, however, it is crucial that we specify how to implement every step of problem-solving process (Fig. 1.1). In particular, the question of how data is to be loaded into the quantum processor must be explicitly addressed. Indeed, any quantum algorithms that obtain speedups with the aid of abstract oracles are necessarily incomplete; an implementation of the requisite oracles is required to apply the algorithm in practice.

## 1.3 Quantum random access memory (focus of this thesis)

Quantum random access memory (QRAM) [14–19] is a highly-specialized quantum architecture that could solve many of the challenges associated with loading classical data into quantum processors. With a classical RAM, any single data item stored in memory can be quickly loaded into the central processor. In contrast, with a QRAM, multiple different classical data items can be loaded simultaneously *in superposition*. QRAM thus acts as a link between the classical and quantum worlds. Indeed, QRAM can serve as a fast and general-purpose implementation of a quantum oracle, and access to QRAM would solve the data-input bottleneck problem in many applications.

Actually constructing a large QRAM, however, is expected to be very challenging. The main obstacle is that building a QRAM requires a number of qubits that scales linearly in proportion to the size of the data set being loaded. As one considers applications involving larger data sets, the hardware cost of QRAM grows, and issues of scalability become increasingly important. For example, if the QRAM is to be both reliable and hardware efficient, the underlying quantum hardware must be simultaneously highly coherent and highly compact. Moreover, even with highly-coherent components, errors will become inevitable as the size of the QRAM grows, so it is crucial that some means of reliably loading data even in the presence of errors is developed.

These scalability problems are not fundamentally different from those faced by universal quantum computers. Solutions to these problems developed in context of universal quantum computing, however, cannot always be practically applied to QRAM. For instance, conventional approaches to quantum error correction [20] can result in impractically high overheads when applied to QRAM [18]. At the same time, QRAM is a highly-specialized architecture that serves a limited purpose, and the challenges of scalability can and should be addressed with this specialization in mind. Implementing QRAM does not require a universal set of quantum operations, for example, and this fact can be exploited to simplify QRAM architectures.

Ultimately, in order to make constructing a large-scale QRAM practical, specifically-tailored solutions to QRAM's scalability problems will be required. Developing such solutions is the main focus of this thesis.

## 1.4 Summary of main results and thesis organization

In Chapter 2, we begin by providing a basic review of quantum oracles and the query model of computation. We also review a representative sample of quantum algorithms in order to illustrate the ubiquity and utility of quantum oracles. We then introduce the notion of QRAM, and give a self-contained review of the topic. We conclude the chapter by enumerating several of the practical challenges associated with QRAM's implementation.

In Chapter 3, we study the effects of noise and decoherence on QRAM. We show that QRAM can be surprisingly resilient to decoherence, such that high-fidelity queries can be performed even in the presence of realistic decoherence. More precisely, we prove that the infidelity a QRAM query scales only polylogarithmically with the memory size (i.e. polynomially in the number of address bits) even when all components are subject to arbitrary noise channels, and we verify this scaling numerically. Further, we describe several corollaries of this result that enable significant architectural simplifications for QRAM.

In Chapter 4, we present an efficient scheme for further suppressing errors in QRAM queries. In contrast to quantum error correction, which typically entails a large overhead when applied to QRAM, our scheme is hardware efficient, with an overhead that is independent of the memory size. We first quantify the error suppression capabilities of our scheme for general quantum operations, then tailor our analysis to QRAM. Though our scheme cannot match the exponential error suppression achieved by quantum error correction, it is suitable for use in near-term devices. Indeed, taken together, the results of Chapters 3 and 4 demonstrate that small- to medium- scale QRAM can be reliably implemented using realistically-noisy quantum hardware available today.

Finally, in Chapter 5 we propose experimental implementations of QRAM in circuit quantum acoustodynamics systems. In these systems, highly-compact acoustic modes are the primary carriers of quantum information, and the architectures we propose are naturally hardware-efficient as a result. We describe two distinct architectures: The first is based on Hamiltonian engineering in multimode systems and is better suited for near-term implementation. The second is based on dissipative cat qubits and enables fault-tolerant QRAM queries with low overhead.

For Chapters 3 to 5, we discuss conclusions, open questions, and directions for future research at the end of each chapter.

Throughout this thesis, we assume familiarity with the basic notions of quantum computing. Ref. [21] provides an excellent introduction to the topic. Additionally, Chapter 5 assumes familiarity with circuit quantum electrodynamics. We refer the reader to Ref. [22] for a pedagogical introduction and to Ref. [23] for a recent review.

# Chapter 2

# Quantum random access memory

## 2.1 Quantum oracles

### 2.1.1 The query model

The computational power of quantum computers is frequently analyzed in the so-called query model of computation. In this section, we describe the query model, introducing the concept of an oracle and the notion of query complexity. Refs. [11–13] all provide excellent reviews of these topics.

Suppose that we wish to solve some computational problem. Without loss of generality[1], we may assume that the input to the problem—a specification of the problem instance to be solved—is some classical data vector $\mathbf{x}$. In the query model, this input is only accessible through an *oracle* (sometimes also referred to as a black box) that can be queried to reveal information about $\mathbf{x}$. Though the oracle will provide information about $\mathbf{x}$ when prompted, how exactly it retrieves this information is left unspecified. The goal is to solve the problem using as few queries to the oracle as

[1] In some settings, it is more natural to assume that the input is some function $f$, rather than a data vector $\mathbf{x}$. Provided that the domain of $f$ is a set of consecutive integers, then the latter can be reduced to the former by taking a data vector that specifies the outputs of the function over the set, $x_i = f(i)$.

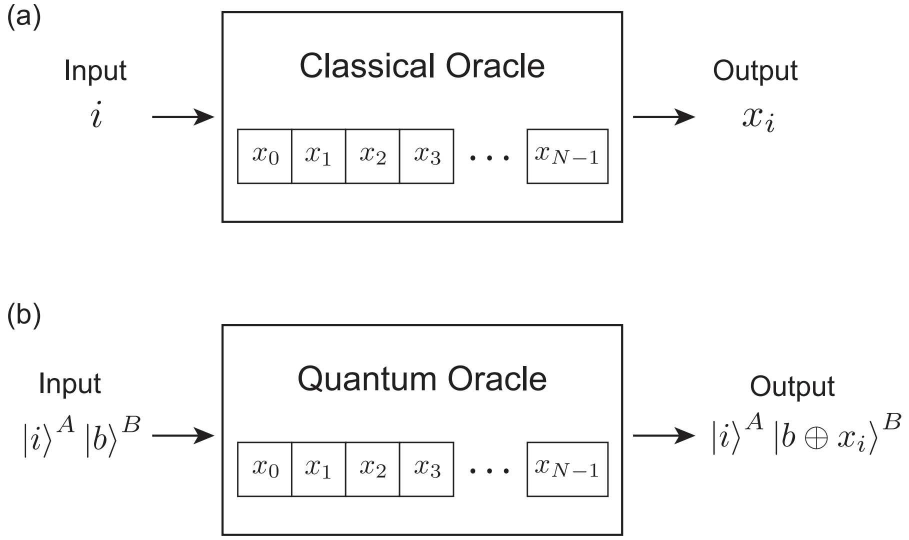


Figure 2.1: Classical and quantum data-lookup oracles. Both the classical (a) and quantum (b) oracles provide access to a *classical* data vector $\mathbf{x} = (x_0, \dots, x_{N-1})$. For a classical oracle, the input and output of a query are both classical numbers, while for a quantum oracle, the input and output of a query are both quantum states.

possible, without exploiting any details about how the oracle might operate.

Oracles can be either classical or quantum. The simplest example of a classical oracle is a so-called *data-lookup oracle*, illustrated in Fig. 2.1(a). The oracle is queried by providing it with an index $i$ as input, and the oracle subsequently outputs the corresponding vector element $x_i$.

A natural generalization of this classical data-lookup oracle is the quantum data-lookup oracle, illustrated in Fig. 2.1(b). In the quantum case, the inputs and outputs of the oracle query are quantum states, and the query itself is some unitary operation, $O_{\mathbf{x}}^{(\mathrm{DL})}$, that implements the mapping

$$O_{\mathbf{x}}^{(\mathrm{DL})} |i\rangle^A |b\rangle^B = |i\rangle^A |b \oplus x_i\rangle^B , \tag{2.1}$$

where the label $b$ denotes an arbitrary computational basis state, and $\oplus$ denotes addition modulo 2. The superscripts $A$ and $B$ denote two quantum registers; the state of register $A$ specifies which element to look up, and the query encodes this element into the state of register $B$. Note that applying the oracle a second returns the system to its initial state

$$O_{\mathbf{x}}^{(\mathrm{DL})} \left|i\right\rangle^{A} \left|b \oplus x_i\right\rangle^{B} = \left|i\right\rangle^{A} \left|b \oplus x_i \oplus x_i\right\rangle^{B} = \left|i\right\rangle^{A} \left|b\right\rangle^{B}. \tag{2.2}$$

It follows that $O_{\mathbf{x}}^{(\mathrm{DL})}$ is unitary (and involutory), independent of the details of the classical data. It should be emphasized that, though the inputs and outputs of the query are quantum, the data being queried is classical. In this way, quantum oracles act as an interface between classical data and quantum algorithms.

As an aside, let us comment on the dimensionality of the quantum registers $A$ and $B$. Suppose $\mathbf{x}$ is a length-$N$ vector, with entries $x_i$ each specified by $d$ binary digits. It then suffices to choose $A$ to be $n$-qubit register, where $n \equiv \log_2 N$. This way, the Hilbert space dimension of $A$ is $N$, which is sufficient to index all elements of $\mathbf{x}$. It suffices to choose $B$ to be a $d$-qubit register because a single qubit is sufficient to store a single classical bit. As an example, suppose that we wish to query the 5-th element of a length-8 vector ($N = 8$), where the vector elements are specified by 2 binary digits ($d = 2$), e.g. $x_5 = 01$. The corresponding query is,

$$O_{\mathbf{x}}^{(\mathrm{DL})} \left|101\right\rangle^{A} \left|00\right\rangle^{B} = \left|101\right\rangle^{A} \left|01\right\rangle^{B}, \tag{2.3}$$

where for simplicity we have set $\left|b\right\rangle^{B} = \left|00\right\rangle^{B}$. Note that, for the $A$ register, the index $i = 5$ is specified by the corresponding binary decomposition, 101.

Quantum data-lookup oracles are strictly more powerful than their classical counterparts. Indeed, a quantum data-lookup oracle can easily be used to emulate the corresponding classical data-lookup oracle; simply prepare the input state $\left|i\right\rangle^{A} \left|0\right\rangle^{B}$, query $O_{\mathbf{x}}^{(\mathrm{DL})}$, and measure the $B$ register of the resultant state $\left|i\right\rangle^{A} \left|x_i\right\rangle^{B}$. The increased power of quantum oracles derives from the fact that they may be queried in superposition. If one prepares the register $A$ in a superposition of different states, it follows from the linearity of quantum mechanics that $O_{\mathbf{x}}^{(\mathrm{DL})}$ will look up the corresponding vector elements in superposition,

$$O_{\mathbf{x}}^{(\mathrm{DL})} \sum_{i=0}^{N-1} \alpha_i \left|i\right\rangle^A \left|b\right\rangle^B = \sum_{i=0}^{N-1} \alpha_i \left|i\right\rangle^A \left|b \oplus x_i\right\rangle^B . \tag{2.4}$$

As we discuss later in this chapter, this ability to perform queries in superposition is exploited by a great many quantum algorithms in order to reduce the number of queries required to solve a problem, i.e. to provide quantum speedups.

In the query model, the complexity of a problem is naturally quantified by the number of queries that are required to solve it. This number is referred to as the *query complexity* of the problem. Upper bounds on the query complexity can be obtained by constructing specific algorithms to solve the problem, while lower bounds can be obtained through a variety of methods (e.g., the classical [24] and quantum [25] adversary methods, or polynomial methods [26]; see Refs. [12, 27] for reviews). Though such bounds are intrinsically interesting in the context of quantum complexity theory, upper bounds can also be translated into more practical statements in the context of the usual circuit model of quantum computation [21]. For example, suppose an algorithm solves some problem using $O(Q)$ queries to the oracle $O_{\mathbf{x}}^{(\mathrm{DL})}$ and $O(S)$ additional constant-depth operations. Then, given a depth-$T$ quantum circuit to implement $O_{\mathbf{x}}^{(\mathrm{DL})}$, we can construct a quantum circuit with depth $O(QT + S)$ that solves the same problem (simply replace each oracle query by the circuit implementing $O_{\mathbf{x}}^{(\mathrm{DL})}$). This statement has an important implication: any efficient algorithm in the query model can be immediately translated into an efficient algorithm in the circuit model if efficient implementations of the requisite oracles are available.

### 2.1.2 The versatility of data-lookup oracles

In this section, we describe how data-lookup oracles can be leveraged to perform a variety of other interesting functions. Specifically, we show how querying a data-lookup oracle enables one to implement analogous phase-flip oracles, encode classical data in the amplitudes of quantum states, and synthesize unitary operations. Such techniques are used in a wide variety of quantum algorithms, as discussed later in this chapter.

#### Phase-flip oracles

A phase-flip oracle, $O_{\mathbf{x}}^{(\mathrm{PF})}$, is defined to be the unitary operator which applies a $-1$ phase to a computational basis state conditioned on the corresponding element of the binary data vector $\mathbf{x}$,

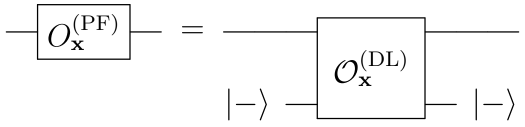
$$O_{\mathbf{x}}^{(\mathrm{PF})} \left|i\right\rangle = (-1)^{x_i} \left|i\right\rangle . \tag{2.5}$$

That is, $O_{\mathbf{x}}^{(\mathrm{PF})}$ applies a $-1$ phase to computational basis state $\left|i\right\rangle$ if $x_i = 1$. Otherwise, if $x_i = 0$, the oracle acts trivially. Grover's algorithm for searching an unstructured database famously employs such an oracle [28], and we discuss this application further in Section 2.1.3.

There is a well-known construction [21] that allows one to implement $O_{\mathbf{x}}^{(\mathrm{PF})}$ using a single query to the corresponding data-lookup oracle $O_{\mathbf{x}}^{(\mathrm{DL})}$. The trick is to prepare the output qubit in the state $\left|-\right\rangle \equiv \left(\left|0\right\rangle - \left|1\right\rangle\right)/\sqrt{2}$, then query $O_{\mathbf{x}}^{(\mathrm{DL})}$,

$$\begin{aligned} O_{\mathbf{x}}^{(\mathrm{DL})} \left|i\right\rangle^{A} \left|-\right\rangle^{B} &= \frac{1}{\sqrt{2}} \left|i\right\rangle^{A} \left( \left|0 \oplus x_i\right\rangle^{B} - \left|1 \oplus x_i\right\rangle^{B} \right) \\ &= (-1)^{x_i} \left|i\right\rangle^{A} \left|-\right\rangle^{B} . \end{aligned} \tag{2.6}$$

The second line is obtained by observing that interchanging $\left|0\right\rangle$ and $\left|1\right\rangle$ in the state $\left|-\right\rangle$ gives $-\left|-\right\rangle$. Notice that the state of the qubit $B$ is not changed by the query (this is an example of "phase kickback," a phenomenon widely exploited in quantum computing). It follows that the query does not entangle the qubit $B$ with the register $A$, and so the qubit $B$ can be discarded. The phase-flip oracle may thus be implemented as

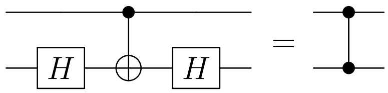

$O_{\mathbf{x}}^{(\mathrm{PF})}$
=
$\mathcal{O}_{\mathbf{x}}^{(\mathrm{DL})}$
$|-\rangle$
$|-\rangle$

(2.7)

More generally, by conjugating the $B$ register with Hadamard gates, $O_{\mathbf{x}}^{(\mathrm{DL})}$ can be used to apply a $-1$ phase to the joint state $|i\rangle^A |b\rangle^B$ conditioned on both the corresponding data vector element and the state of the $B$ register,

$$(I \otimes H) O_{\mathbf{x}}^{(\mathrm{DL})} (I \otimes H) |i\rangle^A |b\rangle^B = (-1)^{x_i b} |i\rangle^A |b\rangle^B , \tag{2.8}$$

where the implementation of $O_{\mathbf{x}}^{(\mathrm{PF})}$ described above corresponds to the case of $b = 1$. Eq. (2.8) can be understood as analogous to the statement that conjugating the target qubit of a CNOT gate with Hadamards yields a CZ gate,



(2.9)

The generalized phase-flip oracle described by Eq. (2.8) is also frequently used in the quantum algorithms literature.

### Amplitude encoding oracles

An amplitude encoding oracle, $O_{\mathbf{x}}^{(\mathrm{AE})}$, is defined to be a unitary operator which encodes the entries of a length-$N$ data vector, $\mathbf{x}$, into the amplitudes of a $\log N$-qubit quantum state,

$$O_{\mathbf{x}}^{(\mathrm{AE})} \left( |0\rangle^{\otimes \log N} \right) = \frac{1}{|\mathbf{x}|_2} \sum_{i=0}^{N-1} x_i |i\rangle \equiv |\psi(\mathbf{x})\rangle , \tag{2.10}$$

where $|\mathbf{x}|_2 = (\sum_i |x_i|^2)^{1/2}$. The ability to encode a data vector in the amplitudes of a quantum state is frequently assumed in quantum algorithms for linear algebra and machine learning [10], and quantum chemistry [29], for example. More generally, the ability to implement $O_{\mathbf{x}}^{(\mathrm{AE})}$ for any $\mathbf{x}$ enables one to prepare arbitrary quantum states. We now describe two well-known methods for implementing $O_{\mathbf{x}}^{(\mathrm{AE})}$ using related data-lookup oracles.

The first method requires only two queries to a data-lookup oracle but requires postselection, with a success probability that depends on $\mathbf{x}$. First, Hadamard gates are applied to all $\log N$ qubits to prepare them in an equal superposition over all $N$ computational basis states,

$$(H\,|0\rangle)^{\otimes \log N} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle^S \,, \tag{2.11}$$

where the superscript $S$ will be used to distinguish the $(\log N)$-qubit system register from ancillary registers. Then, an ancillary register $A_1$ is added, and a data-lookup oracle $O_{\mathbf{y}}^{(\mathrm{DL})}$ is queried,

$$O_{\mathbf{x}}^{(\mathrm{DL})} \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle^S |0\rangle^{A_1} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle^S |y_i\rangle^{A_1} \,. \tag{2.12}$$

where $\mathbf{y} \equiv \mathbf{x}/|\mathbf{x}|_\infty$, and $|\mathbf{x}|_\infty = \max_i |x_i|$ (the vector $\mathbf{y}$ is proportional to $\mathbf{x}$, but normalized so that $|y_i| \leq 1$ for all $i$). Next, another ancillary qubit, $A_2$, is added, and $A_2$ is rotated conditioned on the state of $A_1$, resulting in the state

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle^S |y_i\rangle^{A_1} \left( y_i\,|0\rangle^{A_2} + \sqrt{1-|y_i|^2}\,|1\rangle^{A_2} \right) . \tag{2.13}$$

One then measures the qubit $A_2$ in the standard basis and postselects on obtaining the outcome $|0\rangle$. $A_2$ is then discarded, yielding the (normalized) state

$$\frac{1}{|\mathbf{x}|_2} \sum_{i=0}^{N-1} x_i \, |i\rangle^S \, |y_i\rangle^{A_1} . \tag{2.14}$$

Finally, $O_{\mathbf{x}}^{(\mathrm{DL})}$ is queried again to erase the data stored in the register $A_1$,

$$O_{\mathbf{x}}^{(\mathrm{DL})} \frac{1}{|\mathbf{x}|_2} \sum_{i=0}^{N-1} x_i \, |i\rangle^S \, |y_i\rangle^{A_1} = \frac{1}{|\mathbf{x}|_2} \sum_{i=0}^{N-1} x_i \, |i\rangle^S \, |0\rangle^{A_1} . \tag{2.15}$$

Discarding $A_1$ yields the desired state $|\psi(\mathbf{x})\rangle$.

In the above procedure, the probability of successful postselection is

$$\frac{1}{N} \sum_{i=0}^{N-1} |y_i|^2 = \frac{1}{N} \frac{|\mathbf{x}|_2^2}{|\mathbf{x}|_\infty^2}. \tag{2.16}$$

When the entries of $\mathbf{x}$ are relatively uniform, with no entries significantly larger than the others, this success probability can approach 1. In such cases, one can efficiently encode the vector $\mathbf{x}$ in the amplitudes of a quantum state [30]. However, in the worst case, the success probability is only $1/N$, as $|\mathbf{x}|_\infty \leq |\mathbf{x}|_2$. Thus efficient amplitude encoding is not possible in general using this approach. We note that the success probability can be improved to $O(1)$ using amplitude-amplification, at the cost of $O(\sqrt{N})$ additional oracle queries [31, 32]. In fact, the quantum search lower bound of Ref. [24] implies that at least $\Omega(\sqrt{N})$ queries to $O_{\mathbf{x}}^{(\mathrm{DL})}$ are required to prepare $|\psi(\mathbf{x})\rangle$ with near-unit probability.

As a brief aside, the big-$O$ and big-$\Omega$ notation is defined as follows. We say $f(x) = O(g(x))$ if $|f(x)| \leq cg(x)$ for all $x \geq x_0$, where $c$ and $x_0$ are constants. Similarly, we say $f(x) = \Omega(g(x))$ if $|f(x)| \geq cg(x)$ for all $x \geq x_0$. Put simply, $O$ denotes an asymptotic upper bound, while $\Omega$ denotes an asymptotic lower bound.

The second method for implementing $O_{\mathbf{x}}^{(\mathrm{AE})}$ requires $O(\log N)$ queries to data-lookup oracles and is deterministic. However, the approach requires that additional information about $|\mathbf{x}|$ (namely, its sub-norms) also be accessible via data-lookup oracles. This procedure is described in Refs. [33–36], and we summarize it below.

First, define $p_j$ to be the probability that the first $w$ qubits of $|\psi(\mathbf{x})\rangle$ are in state $|j\rangle$,

$$p_j = \frac{1}{|\mathbf{x}|_2^2} \sum_{\text{prefix}(i)=j} |x_i|^2 \tag{2.17}$$

where $j \in \{0,1\}^w$, and $\text{prefix}(i) = j$ denotes the set of all bit strings $i$ that have their first $w$ bits equal to $j$. The procedure builds up $|\psi(\mathbf{x})\rangle$ qubit by qubit, starting by preparing single qubit in the state

$$|\psi_1\rangle = \sqrt{p_0}\,|0\rangle + \sqrt{p_1}\,|1\rangle\,. \tag{2.18}$$

Observe that the probability that this qubit is $|0\rangle$ is equivalent to the probability that the first qubit of $|\psi(\mathbf{x})\rangle$ is 0.

To add the remaining qubits, we require data-lookup oracles $O^{(\text{DL})}_{\boldsymbol{\theta}^{(w)}}$. Here, the entries of the data vector $\boldsymbol{\theta}$ are defined as $\theta_j = \cos^{-1}\sqrt{p_{j0}/p_j}$, where $j$ is a $w$-bit string, and $j0$ is a $(w+1)$-bit string with the last bit equal to 0. Qubits are then added to the state via the following recursive procedure,

$$\begin{aligned}
|\psi_w\rangle &= \sum_{j\in\{0,1\}^w} \sqrt{p_j}\,|j\rangle \\
&\xrightarrow{O^{(\text{DL})}_{\boldsymbol{\theta}^{(w)}}} \sum_{j\in\{0,1\}^w} \sqrt{p_j}\,|j\rangle\,|\theta_j\rangle \\
&\longrightarrow \sum_{j\in\{0,1\}^w} \sqrt{p_j}\,|j\rangle \left(\sqrt{\frac{p_{j0}}{p_j}}\,|0\rangle + \sqrt{\frac{p_{j1}}{p_j}}\,|1\rangle\right)|\theta_j\rangle \\
&\xrightarrow{O^{(\text{DL})}_{\boldsymbol{\theta}^{(w)}}} \sum_{j\in\{0,1\}^{(w+1)}} \sqrt{p_j}\,|j\rangle = |\psi_{w+1}\rangle\,.
\end{aligned} \tag{2.19}$$

In the second line, the oracle $O^{(\text{DL})}_{\boldsymbol{\theta}^{(w)}}$ is queried, and the result is stored in an ancillary register. In the third line, a new qubit is added to the state, and this qubit is rotated (conditioned on the state of the ancillary register) by an angle $\theta_j$. Finally, in the third line, $O^{(\mathrm{DL})}_{\boldsymbol{\theta}^{(w)}}$ is queried again to erase the data stored in the ancillary register, which is subsequently discarded. Starting with $|\psi_1\rangle$, the above procedure can be repeatedly applied to prepare states $|\psi_2\rangle, \ldots |\psi_w\rangle$, where $|\psi_w\rangle$ has the same measurement statistics as the first $w$ qubits of the state $|\psi(\mathbf{x})\rangle$. After $\log N - 1$ rounds, corresponding to $2(\log N - 1)$ oracle queries, the desired state $|\psi_{\log N}\rangle = |\psi(\mathbf{x})\rangle$ has been prepared.

This procedure illustrates that it is possible to efficiently prepare an arbitrary quantum state when equipped with suitable data-lookup oracles [33]. In this context, efficient refers to the query complexity; the total number of oracle queries scales only polynomially with the number of qubits. If the oracles themselves can be implemented via polynomial-depth circuits, then it follows that the above procedure furnishes polynomial-depth state-preparation circuits. For example, Ref. [37] employs this procedure to find polynomial-depth state-preparation circuits in the situation where the amplitudes of $|\psi(\mathbf{x})\rangle$ are related to some efficiently-integrable probability distribution.

### Unitary synthesis

The problem of unitary synthesis is to implement an arbitrary $N \times N$ unitary $U$, specified by a list of its matrix elements. That is, if you are handed matrix representation of $U$ written out in full on a sheet of paper (or, more practically, stored in some classical data structure), how can you construct a quantum circuit that implements $U$? In answer to this question, explicit *gate-based* constructions have been found that allow one to decompose an arbitrary $U$ into sequences of $O(N^2)$ single- and two-qubit gates [38–42]. A theoretical lower bound of $\Omega(N^2)$ gates can be obtained via counting arguments [43, 44], so these constructions are optimal.

Alternatively, there exist *oracle-based* constructions that allow one to implement an arbitrary unitary $U$ assuming that access to the matrix elements is provided by suitable oracles [36, 45–47]. We describe two such constructions below, thereby demonstrating that suitable oracles enable the synthesis of arbitrary unitaries. Moreover, the number of oracle queries required in these constructions is generally[2] $O(N)$. Therefore, in special cases where the requisite oracles can be implemented using only poly($n$) gates, these oracle-based construction can provide polynomial reductions in the gate complexity relative to the gate-based constructions of Refs. [38–42].

The first construction is based on a reduction [46] of unitary synthesis to Hamiltonian simulation. Given a unitary $U$, we define a corresponding Hamiltonian

$$H = \begin{pmatrix} 0 & U \\ U^\dagger & 0 \end{pmatrix}, \tag{2.20}$$

where $H$ acts on a Hilbert space that twice as large as that which $U$ acts on, i.e. with one additional qubit. Because $H^2 = I$, we have that

$$e^{-iHt} = \cos(t)I - i\sin(t)H, \tag{2.21}$$

and hence that

$$e^{-iH\pi/2}\left|1\right\rangle\left|\psi\right\rangle = -i\left|0\right\rangle\left(U\left|\psi\right\rangle\right). \tag{2.22}$$

Thus, the ability to simulate evolution under $H$ enables one to apply $U$ to an arbitrary state $\left|\psi\right\rangle$. Now, assuming a data-lookup oracle that accesses matrix elements of $U$ is available, one can easily construct a data-lookup oracle that accesses matrix elements of $H$. Hence, oracle-based algorithms for Hamiltonian simulation (discussed further in Section 2.1.3) can be applied to implement $U$. With an optimal algorithm, e.g. quantum signal processing [48], the required number of queries is $O(N)$.

[2] The counting arguments of Ref. [43, 44] do not apply to the situation where $U$-dependent oracles are invoked.

The second construction [36, 45, 47] is based on the decomposition of $U$ into a product of Householder reflections [49]. First, define

$$U' = |0\rangle \langle 1| \otimes U + |1\rangle \langle 0| \otimes U^{\dagger}, \tag{2.23}$$

and observe that the ability to implement $U'$ enables one to apply $U$ to an arbitrary state $|\psi\rangle$,

$$U' |1\rangle |\psi\rangle = |0\rangle (U |\psi\rangle). \tag{2.24}$$

Now, using the Householder reflection decomposition, one can show [47] that $U'$ can be expressed as a product of $N$ reflections,

$$U' = \prod_{i=1}^{N} R_{w_i}, \tag{2.25}$$

where

$$R_{w_i} = I - 2 |w_i\rangle \langle w_i|, \tag{2.26}$$

and

$$|w_i\rangle = (|1\rangle |i\rangle - |0\rangle |U_i\rangle)/\sqrt{2}. \tag{2.27}$$

Here, $|U_i\rangle$ is an amplitude encoding of $U_i$, the $i$-th column of $U$,

$$|U_i\rangle = \frac{1}{|U_i|_2} \sum_{j=0}^{N-1} U_{ji} |j\rangle. \tag{2.28}$$

The reflection $R_{w_i}$ can be implemented with the aid of an amplitude encoding oracle $O_{w_i}^{(\mathrm{AE})}$,

$$R_{w_i} = O_{w_i}^{(\mathrm{AE})}(I - 2 |0\rangle \langle 0|)(O_{w_i}^{(\mathrm{AE})})^{\dagger}, \tag{2.29}$$

where $(I - 2 |0\rangle \langle 0|)$ is a multiply-controlled phase gate (it imparts a $-1$ phase to

$|0\rangle$, but does nothing to orthogonal states). Finally, we note that $O_{w_i}^{(\mathrm{AE})}$ can be straightforwardly implemented using an oracle, $O_{U_i}^{(\mathrm{AE})}$ that implements an amplitude encoding of the $i$-th column of $U$ [36]. Therefore, amplitude encoding oracles for the columns of $U$ enable one to implement $U$ itself. The total number of oracle queries required is $O(N)$, since each of the $N$ reflections $R_{w_i}$ can be implemented in a constant number of queries.

### 2.1.3 Oracles in context: use in quantum algorithms

In this section, we review several quantum algorithms that invoke quantum oracles. The purpose of this review is two-fold: First, these examples illustrate the ubiquity of oracles in the quantum algorithms literature. Second, these examples highlight the important role that oracles play as an interface between classical data and quantum algorithms.

We review algorithms for period-finding (factoring) [1], unstructured search [28], and Hamiltonian simulation [29], and we justify this selection as follows. Perhaps surprisingly, there are only three main classes of quantum algorithm from which nearly[3] all modern quantum algorithms are derived: factoring, search, and simulation [50]. For example, factoring gave rise to the more general procedure of quantum phase estimation [7], which, together with algorithms for Hamiltonian simulation, forms the basis of the so-called HHL algorithm for solving linear systems [51]. The HHL algorithm, in turn, underlies nearly all quantum algorithms for linear algebraic problems, with numerous applications in data analysis and machine learning [10, 52, 53]. Below, we discuss one example from each of the three classes (factoring, search, and simulation), with the understanding that these examples are representative of a much broader set of algorithms.

[3] Notably, these classes do not include variational algorithms or algorithms for demonstrating quantum computational supremacy.

Importantly, we show how each of the three paradigmatic examples can be framed in the query model, and by extension that nearly all algorithms can be framed in this model. However, we caution the reader that this framing does *not* imply that a general-purpose oracle implementation (such as QRAM) is required to run most quantum algorithms. In some cases, the structure of a problem can be exploited to construct efficient implementations of the requisite oracles. On the other hand, there are many applications where no such structure exists, and for these applications a general-purpose oracle implementation is necessary. Refs. [34, 51, 54–66] provide examples of algorithms that require a general-purpose oracle implementation because there is no obviously-exploitable structure in the data.

### Period finding

The problem statement of the period-finding algorithm is as follows. Suppose that you are presented with a length-$N$ classical data vector $\mathbf{x}$ and promised that the data is periodic,

$$x_i = x_{i+r}, \text{ for all } i, \tag{2.30}$$

for some unknown period $r$. Further suppose that this data vector is accessible only through an oracle. The goal is to determine $r$ with as few oracle queries as possible.

Classically, any algorithm to solve this problem necessarily requires at least $\Omega(r)$ queries. An optimal approach is thus simply to query each element of $\mathbf{x}$ sequentially, until a repeated element is obtained (we assume that there are no repeated values within a single period for simplicity). Because $r \sim N$ in the worst case, the query complexity of the classical algorithm is $O(N)$.

With the assistance of a quantum oracle, this problem can be solved with only a single query [1, 21]. The algorithm uses two registers, $A$ and $B$, with $n = \log N$ qubits and $d$ qubits, respectively. Here $d$ denotes the number of binary digits needed to specify a data vector element $x_i$. The algorithm begins by initializing both registers in the all-$|0\rangle$ state, and a layer of Hadamard gates is applied to register $A$ to prepare it in an equal superposition state. Then, a data-lookup oracle $O_{\mathbf{x}}^{(\mathrm{DL})}$ is queried. After the query, the state of the system is

$$O_{\mathbf{x}}^{(\mathrm{DL})}(H^{\otimes n} \otimes I) \left|0\right\rangle^A \left|0\right\rangle^B = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \left|j\right\rangle^A \left|x_j\right\rangle^B . \tag{2.31}$$

To proceed, we need to introduce the quantum Fourier transform. The quantum Fourier transform, QFT, is a unitary operation that we define through its action on the basis states,

$$\mathrm{QFT}\left|j\right\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i\, jk/N} \left|k\right\rangle , \tag{2.32}$$

where the coefficients on the right hand side can be recognized as those which appear in the classical discrete Fourier transform. Now, in anticipation of applying the quantum Fourier transform, observe that $\left|x_j\right\rangle^B$ can be expressed as

$$\left|x_j\right\rangle^B = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i\, js/r} \left|x_s\right\rangle^B , \tag{2.33}$$

where

$$\left|x_s\right\rangle^B \equiv \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-2\pi i\, js/r} \left|x_j\right\rangle^B . \tag{2.34}$$

Inserting the expression (2.33) into Eq. (2.31), the state of the system at this point in the algorithm is

$$\begin{aligned} \frac{1}{\sqrt{rN}} \sum_{j=0}^{N-1} \sum_{s=0}^{r-1} e^{2\pi i\, js/r} \left|j\right\rangle^A \left|x_s\right\rangle^B &= \frac{1}{\sqrt{rN}} \sum_{s=0}^{r-1} \left( \sum_{j=0}^{N-1} e^{2\pi i\, js/r} \left|j\right\rangle^A \right) \left|x_s\right\rangle^B \\ &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left( \mathrm{QFT} \left|s/r\right\rangle^A \right) \left|x_s\right\rangle^B , \end{aligned} \tag{2.35}$$

where for simplicity we have assumed that $N$ is an integer multiple of $r$ in order to express the state on the second line directly in terms of QFT. Applying the inverse quantum Fourier transform, $(\mathrm{QFT})^\dagger$, to register $A$ then yields,

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|s/r\right\rangle^A\left|x_s\right\rangle^B. \tag{2.36}$$

Finally, register $A$ is measured, and one of the different possible outcomes, $s/r$, is obtained (each outcome occurs with probability $1/r$). Remarkably, most of these different measurement outcomes provide sufficient information to determine $r$, which can be obtained through the continued fractions algorithm [21]. Thus, the period can be determined with only a single query to a quantum oracle.

This period-finding algorithm is the core of Shor's famous factoring algorithm [1]. In that context, the structure of the problem can be exploited to develop an efficient implementation of the requisite oracle. In the case of Shor's algorithm, the periodic data is of the specific form

$$x_j = a^j \mathrm{mod} M, \tag{2.37}$$

where $a$ and $M$ are positive integers that have no common factors, with $a < M$. For data of this form, the corresponding oracle,

$$\left|j\right\rangle\left|0\right\rangle \rightarrow \left|j\right\rangle\left|a^j \mathrm{mod} M\right\rangle, \tag{2.38}$$

can be implemented efficiently [21]. In contrast, in order to apply the period-finding algorithm to a periodic but otherwise unstructured data vector $\mathbf{x}$, a general-purpose implementation of $O_{\mathbf{x}}^{(\mathrm{DL})}$ would be required, i.e. an implementation that works for any data vector $\mathbf{x}$.

### Grover's algorithm

The problem statement of Grover's algorithm is as follows. Suppose that you are presented with a classical database and promised that an element of interest is contained somewhere within the database. Further suppose that the database is accessible through an oracle. The goal is to determine the location of the element of interest with as few queries to the oracle as possible. In the simplest incarnation, the database is a length-$N$ binary vector $\mathbf{x}$, with $(N-1)$ of the entries equal to 0 and a single entry equal to 1. The element of interest is the single vector element for which $x_{i^*} = 1$, and the goal is to find the index $i^*$ of this element. For brevity, we present the algorithm in this simplified context.

Any classical algorithm to solve this problem requires $O(N)$ queries. The optimal approach is simply to check entries of $\mathbf{x}$ one-by-one until finding the marked element. Clearly, this approach requires checking $O(N)$ elements of the vector on average.

Grover's algorithm [28], solves this problem using only $O(\sqrt{N})$ oracle queries. The algorithm begins by initializing a register of $n = \log N$ qubits in the state $|0\rangle^{\otimes n}$. A layer of Hadamard gates is applied to prepare these qubits in the equal superposition state

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \, . \tag{2.39}$$

After this preparation, a unitary operator $G$, called the Grover operator, is repeatedly applied to the state. The Grover operator is defined as

$$G = (2\,|\psi\rangle\,\langle\psi| - I)O_{\mathbf{x}}^{(\mathrm{PF})}, \tag{2.40}$$

where $O_{\mathbf{x}}^{(\mathrm{PF})}$ is a phase-flip oracle defined in the previous section. The operator $(2\,|\psi\rangle\,\langle\psi| - I)$ is sometimes referred to as an inversion about the mean because of its effect when applied to a generic state,

$$(2\,|\psi\rangle\,\langle\psi| - I) \sum_{i=0}^{N-1} \alpha_i\,|i\rangle = \sum_{i=0}^{N-1} (2\overline{\alpha} - \alpha_i)\,|i\rangle \tag{2.41}$$

where $\overline{\alpha} = \sum_{i=0}^{N-1} \alpha_i/N$ denotes the mean value of the coefficients $\alpha_i$.

The effect of the Grover operator on the state $\left|\psi\right\rangle$ has an elegant geometric interpretation [21]. Let us define

$$\left|\text{good}\right\rangle = \left|i^*\right\rangle , \tag{2.42}$$

$$\left|\text{bad}\right\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \neq i^*} \left|i\right\rangle . \tag{2.43}$$

The initial state $\left|\psi\right\rangle$ can thus be expressed as

$$\left|\psi\right\rangle = \sqrt{\frac{N-1}{N}} \left|\text{bad}\right\rangle + \sqrt{\frac{1}{N}} \left|\text{good}\right\rangle , \tag{2.44}$$

hence $\left|\psi\right\rangle$ lies in the plane spanned by $\left|\text{good}\right\rangle$ and $\left|\text{bad}\right\rangle$. As illustrated in Fig. 2.2, $G$ has the net effect of rotating $\left|\psi\right\rangle$ in this plane. This is because both $O_{\mathbf{x}}^{(\text{PF})} = (I - 2\left|i^*\right\rangle\left\langle i^*\right|)$ and $(2\left|\psi\right\rangle\left\langle\psi\right| - I)$ act as reflections within this plane. This geometric reasoning can be applied to show that

$$G^k \left|\psi\right\rangle = \cos\left(\frac{2k+1}{2}\theta\right) \left|\text{bad}\right\rangle + \sin\left(\frac{2k+1}{2}\theta\right) \left|\text{good}\right\rangle , \tag{2.45}$$

where $\cos(\theta/2) = \left\langle\psi|\text{bad}\right\rangle = \sqrt{1 - 1/N}$. Thus, each application of $G$ rotates the state by an angle of $\theta$ towards the state $\left|\text{good}\right\rangle$.

Leveraging these results, we see that in the limit of $N \gg 1$, the state after $k = \sqrt{N}\pi/4$ applications of $G$ is

$$G^{(\sqrt{N}\pi/4)} \left|\psi\right\rangle \approx \left|\text{good}\right\rangle = \left|i^*\right\rangle . \tag{2.46}$$

Measuring the system then reveals the location $i^*$ of the marked item. Because each application of $G$ requires only a single query to $O_{\mathbf{x}}^{(\text{PF})}$, the total number of queries required is only $O(\sqrt{N})$.

If Grover's algorithm is to be applied in practice, a means of implementing $O_{\mathbf{x}}^{(\text{PF})}$ is

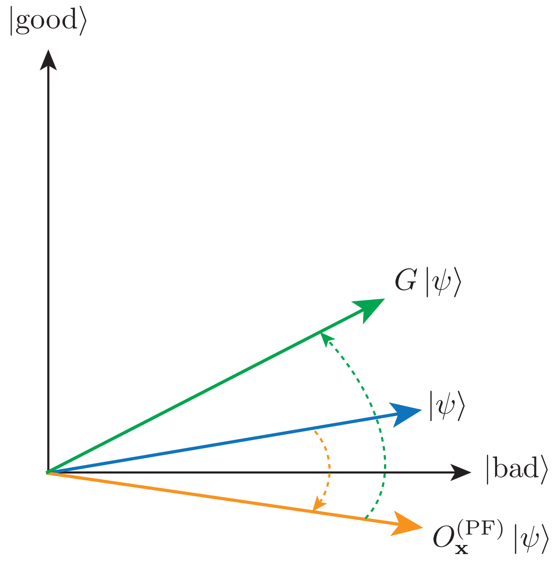


Figure 2.2: Geometric interpretation of the Grover operator (adapted from Ref. [21]). The Grover operator is a product of two reflections: a reflection along $|\text{good}\rangle$, followed by a reflection about $|\psi\rangle$. These reflections act as rotations within the plane spanned by $|\text{good}\rangle$ and $|\text{bad}\rangle$.

required. In some settings, the structure of the problem can be exploited to provide an efficient implementation. For example, Grover's algorithm can be applied to provide quadratic speedups in the search for solutions to problems in the complexity class NP [21]. The relevant property of problems in this complexity class is that potential solutions can be efficiently checked. That is, there exists some efficiently-computable function $f(i)$ such that $f(i) = 1$ if $i$ is a solution to the problem, and $f(i) = 0$ otherwise. Now, because any efficient classical circuit can be made reversible and hence mapped to an efficient quantum circuit, the operation

$$|i\rangle\,|0\rangle \to |i\rangle\,|f(i)\rangle\,, \tag{2.47}$$

can be performed efficiently. This operation is simply a data-lookup oracle, which can be used to realize the phase-flip oracle required by Grover's algorithm. Thus, for problems in NP, the ability to efficiently check solutions classically directly furnishes an efficient quantum circuit to implement the requisite oracle. In contrast, if Grover's algorithm is applied to search an unstructured database, there is no structure that can be readily exploited to construct an oracle implementation. In this case, a general-purpose implementation of $O_{\mathbf{x}}^{(\mathrm{PF})}$ is required, i.e. an implementation that works for any data vector $\mathbf{x}$.

### Hamiltonian simulation

The problem statement of Hamiltonian simulation is as follows. You are presented with an arbitrary quantum state $|\psi\rangle$, as well as a description of some Hamiltonian, $H$, where this description is accessible through a set of oracles (defined below). The goal is to construct a unitary $U$ which simulates the evolution under $H$ for a time $t$ in order to prepare the state

$$U\,|\psi\rangle = e^{-iHt}\,|\psi\rangle\,, \tag{2.48}$$

using as few queries to the oracles as possible. Furthermore, the error in the simulation should be bounded so that $|U - e^{-iHt}| \leq \varepsilon$, according to some metric.

In general, simulating quantum systems using classical computers is hard—the required resources scale exponentially in the size of the system. However, as suggested by Feynman [67], and first shown explicitly by Lloyd [4], quantum computers can perform such simulations efficiently. In the years since, numerous quantum algorithms for Hamiltonian simulation have been proposed [29, 46, 48, 68–70], based on a variety of different methodologies. To review all of these different approaches is beyond the scope of this thesis. Instead, as an illustrative example, we review the algorithm of Ref. [29], chosen for its relative simplicity and near optimality. We highlight the role that oracles play in this algorithm, and we also describe other types of oracles that are frequently used in Hamiltonian simulation algorithms.

We begin by first outlining the main steps of Ref. [29]'s algorithm, then discussing the required oracles and how they are used. The algorithm of is based on a Trotterization approach, where the evolution for a time $t$ is decomposed into $r$ intervals, each of length $t/r$,

$$U = e^{-iHt} = \left(e^{-iHt/r}\right)^r. \tag{2.49}$$

The operator $U_r \equiv e^{-iHt/r}$ can be expanded in a Taylor series as

$$U_r \approx \sum_{k=0}^{K} \frac{1}{k!}(-iHt/r)^k, \tag{2.50}$$

where the series is truncated at order $K$. Together, the parameters $r$ and $K$ determine the accuracy of the simulation, and we describe how they are to be chosen below. Next, the algorithm leverages the fact that any Hamiltonian may be decomposed into a linear combination of unitary operators,

$$H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell, \tag{2.51}$$

where each $H_\ell$ is unitary. Inserting this expression into Eq. (2.50) yields,

$$U_r \approx \sum_{k=0}^{K} \sum_{\ell_1,\ldots,\ell_k=1}^{L} \frac{(-it/r)^k}{k!} \alpha_{\ell_1} \ldots \alpha_{\ell_k} H_{\ell_1} \ldots H_{\ell_k}. \tag{2.52}$$

It will be convenient to define

$$\beta_j \equiv \frac{(t/r)^k}{k!} \alpha_{\ell_1} \ldots \alpha_{\ell_k}, \tag{2.53}$$

$$V_j \equiv (-i)^k H_{\ell_1} \ldots H_{\ell_k}, \tag{2.54}$$

where the index $j$ is used as a shorthand for the indices $\{k, \ell_1, \ldots, \ell_k\}$, and we note that each $V_j$ is unitary. With these definitions,

$$U_r \approx \sum_j \beta_j V_j. \tag{2.55}$$

At the core of the algorithm is a technique for realizing unitary[4] transformations of the form (2.55), which we describe below. Using this technique, the algorithm simply applies $U_r$ a total of $r$ times to perform the simulation.

To implement unitaries of the form (2.55) we assume access to two oracles, $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$ and select$(V)$. The first oracle, $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$, is an amplitude encoding oracle,

$$O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}} \left|0\right\rangle = \frac{1}{\sqrt{s}} \sum_j \sqrt{\beta}_j \left|j\right\rangle , \tag{2.56}$$

where we have defined $s \equiv |\sqrt{\boldsymbol{\beta}}|_2^2$. The second oracle, select$(V)$, acts on two registers and is defined as

$$\mathrm{select}(V) = \sum_j \left|j\right\rangle \left\langle j\right| \otimes V_j. \tag{2.57}$$

Now, we define the operator

$$W \equiv \left[ \left( O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}} \right)^\dagger \otimes I \right] \mathrm{select}(V) \left[ O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}} \otimes I \right], \tag{2.58}$$

and observe that

$$W \left|0\right\rangle \left|\psi\right\rangle = \frac{1}{s} \left|0\right\rangle U_r \left|\psi\right\rangle + \sqrt{1 - \frac{1}{s^2}} \left|\Phi\right\rangle , \tag{2.59}$$

where $\left|\Phi\right\rangle$ is some state orthogonal to $\left|0\right\rangle$ in the first register. To proceed, it is convenient to choose the number of Trotter steps $r$ such that $s = 2$, and one can show that $s = 2$ is obtained for the choice $r = \ln(2)T$, with $T \equiv |\boldsymbol{\alpha}|_1 t$. This choice of $r$ dictates that we must choose $K = O(\log(T/\varepsilon))$ to guarantee an error of at most $\varepsilon$. With these choices, one can verify that

$$- WRW^\dagger RW \left|0\right\rangle \left|\psi\right\rangle = \left|0\right\rangle U_r \left|\psi\right\rangle , \tag{2.60}$$

[4] We note that $\sum_j \beta_j V_j$ is not necessarily unitary, but will be approximately unitary for sufficiently large $K$. For simplicity, we neglect subtleties associated with nonunitarity.

where $R = (I - 2\left|0\right\rangle\left\langle 0\right|) \otimes I$ is a controlled-phase gate on the first register. The construction in Eq. (2.60) is an instance of a more general framework, oblivious amplitude amplification [69], which itself is an extension of Grover's algorithm. For the sake of brevity, however, we do not describe the framework here. Finally, we see from Eq. (2.60) that the operator $(-WRW^{\dagger}RW)$ constitutes an implementation of $U_r$. Moreover, the implementation requires only a constant number of queries to the oracles $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$ and select$(V)$.

Let us analyze the query complexity of this algorithm. The algorithm performs $r = \ln(2)T$ applications of $U_r$, each of which requires only a constant number of oracle queries to implement. The query complexity is thus $O(T)$, i.e. linear in $t$. This complexity has no dependence on $\varepsilon$ because the oracles $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$ and select$(V)$ depend implicitly on $\varepsilon$ through the parameter $K = O(\log(T/\varepsilon))$. For this reason, a more commonly used input model is to assume access to analogous oracles, $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\alpha}}}$ and select$(H)$, that do not depend on $\varepsilon$. As described in Ref. [29], $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$ and select$(V)$ can be implemented using $O(K)$ queries to $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\alpha}}}$ and select$(H)$. Thus, with respect to these latter oracles the query complexity is $O(T\log(T/\varepsilon))$. This complexity is not quite optimal, but there now exist more sophisticated algorithms [70] that can achieve a provably optimal query complexity of $O(T+\log(1/\varepsilon))$, i.e. additive in $t$ and $\log(1/\varepsilon)$.

The algorithm that we have just described operates in the so-called linear combination of unitaries (LCU) input model. The defining feature of the LCU model is that the Hamiltonian is decomposed into a linear combination of unitaries as in Eq. (2.51), and information about the Hamiltonian is accessed through two oracles: an amplitude encoding oracle that provides information about the coefficients (e.g., $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\alpha}}}$ or $O^{(\mathrm{AE})}_{\sqrt{\boldsymbol{\beta}}}$), and a select-type oracle that provides information about the unitaries (select$(H)$ or select$(V)$). There exists another common input model that is worth mentioning: the $d$-sparse Hamiltonian input model [46, 48]. In this model, the

Hamiltonian is assumed to have at most $d$ non-zero matrix elements per row, and information about these matrix elements is accessible through two oracles,

$$O_H^{(\mathrm{DL})} \left|j, k\right\rangle \left|0\right\rangle = \left|j, k\right\rangle \left|H_{jk}\right\rangle, \tag{2.61}$$

$$O_f \left|j\right\rangle \left|k\right\rangle = \left|j\right\rangle \left|f(j, k)\right\rangle. \tag{2.62}$$

The first, $O_H^{(\mathrm{DL})}$, is simply a data-lookup oracle that, given row and column indices $j$ and $k$, looks up the corresponding matrix element of $H$. The second oracle looks up the locations of non-zero matrix elements: given row and column indices $j$ and $k$, $O_f$ computes the index of the $k$-th non-zero entry in row $j$, denoted by $f(j, k)$. Note that $O_f$ computes the index $f(j, k)$ *in place*, i.e. it overwrites the input $k$.

Whatever the input model, implementations of the requisite oracles are required to deploy a Hamiltonian simulation algorithm. In some situations, additional promises on the structure of $H$ can be exploited to develop efficient oracle implementations. For example, if $H$ corresponds to the Hamiltonian of some physical system (as opposed to an arbitrary Hermitian matrix), symmetry or locality constraints can facilitate efficient oracle implementations. However, it is not always the case that the $H$ under consideration corresponds to the Hamiltonian of some well-structured physical system. Indeed, Hamiltonian simulation is a widely-used subroutine in other quantum algorithms, many of which have nothing to do with simulating physical systems. In cases where structure cannot be exploited to develop efficient oracle implementations, general-purpose oracle implementations are required. We describe such implementations in the next section.

As an additional remark: For the rest of the thesis, we consider general-purpose implementations of data-lookup oracles. But as the above example demonstrates, Hamiltonian simulation algorithms tend to require more exotic oracles, namely select-type oracles such as select$(H)$ and in-place data-lookup oracles such as $O_f$. The very same architectures we develop for implementing data-lookup oracles can be straightforwardly extended to also implement these other oracles.

## 2.2 QRAM: an architecture for implementing quantum oracles

In this section, we introduce quantum random access memory (QRAM) [14–19], which is a general-purpose architecture for the implementation of quantum oracles. QRAM can be understood as a generalization of classical RAM; the classical addressing scheme in the latter is replaced by a quantum addressing scheme in the former. More precisely, in the case of classical RAM, an address $i$ is provided as input, and the RAM returns the memory element $x_i$ stored at that address. Analogously, in the case of QRAM, a quantum superposition of different addresses $|\psi_{\text{in}}\rangle$ is provided as input, and the QRAM returns an entangled state $|\psi_{\text{out}}\rangle$ where each address is correlated with the corresponding memory element,

$$|\psi_{\text{in}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A |0\rangle^B \xrightarrow{\text{QRAM}} |\psi_{\text{out}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A |x_i\rangle^B \,, \tag{2.63}$$

where $N$ is the size of the data vector $\mathbf{x}$, and the superscripts $A$ and $B$ stand for "address" and "bus" respectively. The reader will recognize Eq. (2.63) as the action of the data-lookup oracle $O_{\mathbf{x}}^{(\text{DL})}$ defined in Eq. (2.4); QRAM is an architecture specifically designed to implement such oracles.

QRAM has two features that make it particularly appealing: general applicability and efficiency. QRAM can implement $O_{\mathbf{x}}^{(\text{DL})}$ for arbitrary $\mathbf{x}$, and the time required to implement this oracle is only $O(\log N)$ (albeit at the cost of $O(N)$ ancillary qubits). Together, these two features make QRAM appealing for use as an oracle implementation in a wide variety of quantum algorithms, especially those that require $O(\log N)$

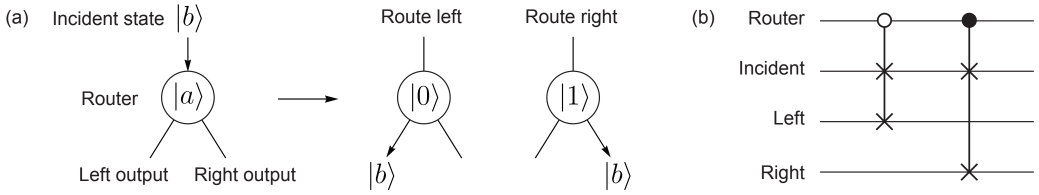


Figure 2.3: Quantum router. (a) Schematic of a quantum router. The router directs an incident qubit $|b\rangle$ at its top port out of either the left or right output ports conditioned on the state $|a\rangle$ of the router. When $|a\rangle = |0\rangle\,(|1\rangle)$, the incident qubit leaves out of the left (right) port. (b) Example of a quantum circuit that implements the routing operation using two controlled-SWAP gates, one conditioned on the control being $|0\rangle$ (open circle) and the other conditioned on the control being $|1\rangle$ (filled circle).

query times in order to claim exponential speedups over their classical counterparts. QRAM can serve as an oracle implementation in quantum algorithms for machine learning [10, 34, 52, 53, 58, 60, 66, 71], chemistry [72, 73], and a host of other areas [28, 36, 51, 54, 55, 63, 74–76], as described in the previous section.

Below, we describe several variants of the QRAM architecture. We begin by introducing the basic building blocks of QRAM, quantum routers, in Section 2.2.1. Next, in Sections 2.2.2 and 2.2.3 we describe the *fanout QRAM* and the *bucket-brigade QRAM* architectures, both based on quantum routers. Finally, in Section 2.2.4, we describe quantum read-only memory (QROM) and hybrid architectures, which can perform operation (2.63) using fewer qubits but longer query times.

### 2.2.1 Quantum routers

In both classical and quantum random accesses memories, each location in memory is indexed by a unique binary address. To read from the memory, an address is provided as input, and the memory element located at that address is returned at the output. In the classical case, transistors are the physical building blocks of the addressing scheme: they act as classical routers, directing electrical signals to the memory location specified by the address bits. Analogously, in the quantum case, quantum routers are the fundamental building blocks of the addressing scheme.

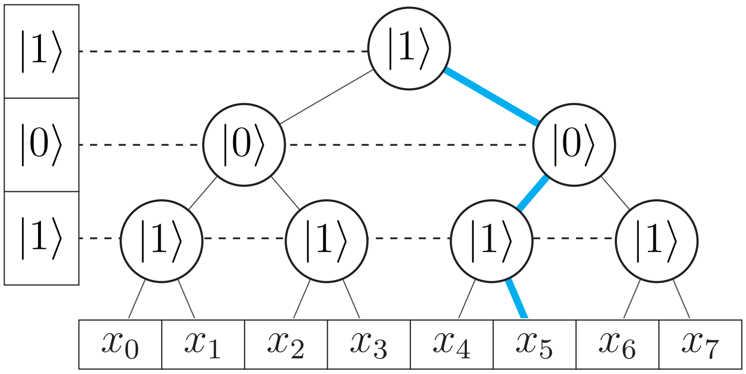


Figure 2.4: Fanout QRAM. Each address qubit controls the states of all routers within the corresponding level of the binary tree. A bus qubit injected at the top node then follows the path (blue) to the specified memory element.

As shown in Fig. 2.3(a), a quantum router is a device that directs incident signals along different paths in coherent superposition, conditioned on the state of a routing qubit. For example, if the routing qubit is in state $|0\rangle$ ($|1\rangle$), then a qubit incident on the router is routed to the left (right). If the routing qubit is in a superposition of these states, then the incident qubit is routed in both directions in superposition, becoming entangled with the routing qubit in the process. Quantum routers can also be understood through the language of quantum circuits Fig. 2.3(b); the routing operation is a unitary that can be implemented via a sequence of controlled-SWAP gates (Fredkin gates).

### 2.2.2 Fanout QRAM

A QRAM can be constructed out of quantum routers as shown in Fig. 2.4 (see Chapter 6 of Ref. [21]). A collection of routers is arranged in a binary tree, with the outputs of routers at one level of the tree acting as inputs to the routers at the next level down. The memory is located at the bottom of the tree, with each of the $N$ memory cells connected to a router at the bottom level. To query the memory, all routing qubits are initialized in $|0\rangle$, and a register of $\log N$ address qubits is prepared in the desired state. All routing qubits at level $\ell$ of the tree are then flipped from $|0\rangle$ to $|1\rangle$ conditioned on the $\ell$-th address qubit. To retrieve the memory contents, a so-called bus qubit is prepared in the state $|0\rangle$ and injected into the tree at the top node. The bus follows the path indicated by the routers down to the memory. Upon reaching a memory cell, the contents of that memory cell are copied into the state of the bus (more on this below). Note that because we consider *classical* data, the data can be copied without violating the no-cloning theorem. For simplicity, we assume that each memory element $x_i$ is a single bit, in which case a single bus qubit suffices to store the memory element (higher-dimensional data can be retrieved using multiple bus qubits). Finally, the bus is routed back out of the tree via the same path, and all routers are flipped back to $|0\rangle$ in order to disentangle them from the rest of the system.

Importantly, because the routers operate coherently, the above procedure allows one to query multiple memory elements in superposition, as in Eq. (2.63). If the address qubits are prepared in a superposition of different computational basis states, the bus is routed to a superposition of different memory locations.

In this architecture, the total time required to perform a query (or, equivalently, the circuit depth) is only $O(\log N)$. The ability to perform queries in logarithmic time can be crucial for algorithms that invoke QRAM in order to claim exponential speedups over their classical counterparts. However, this speed comes at the price of a high hardware cost. To perform operation (2.63), both the fanout and bucket-bridgade architectures require $O(N)$ ancillary qubits to serve as routers.

We have described the operation of the fanout QRAM in the language of quantum routers for simplicity. Of course, the QRAM's operation can be equivalently described in the usual circuit model, and in Fig. 2.5 we provide an equivalent circuit for the case of $N = 8$. The circuit is divided into several stages. During the first stage, labelled $U_1$, the circuit flips the routers at each level of the tree conditioned on the

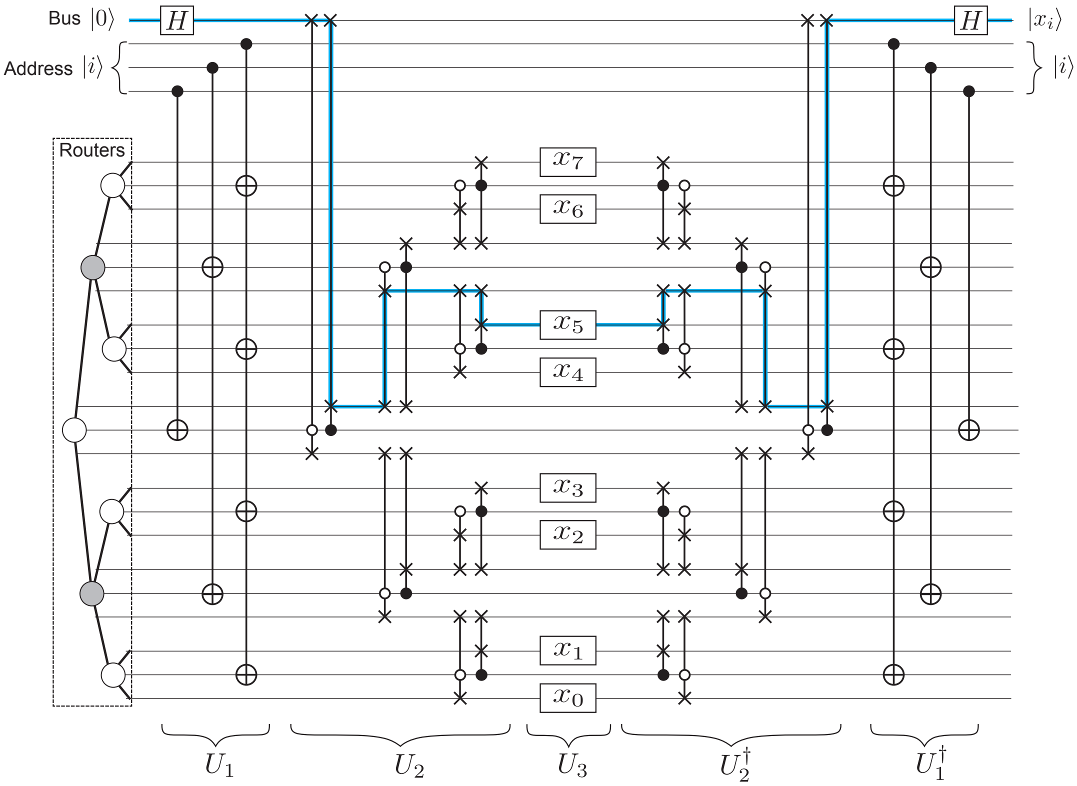


Figure 2.5: Fanout QRAM circuit for $N = 8$. The bus and address registers are indicated by rails at the top of the diagram, and the routers are indicated by the rails below. For each router shown on the left, there are three rails: one for the router's internal state, and two for the router's two output modes. All qubits comprising the routers are initialized to $|0\rangle$. The path of the bus is highlighted in blue for the case where the three address qubits are initialized to $|i\rangle = |101\rangle$. The action of the $x_i$ gates in the middle of the circuit is defined in Fig. 2.6.

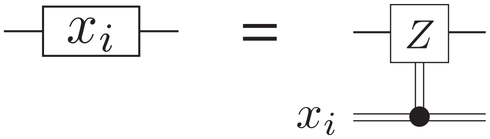


Figure 2.6: Data-copying circuit. A $Z$ gate is applied to the qubit conditioned on the value of $x_i$.

corresponding address qubit. All of the multi-target CNOT gates can be applied in parallel, and this stage can be decomposed into a sequence of single-target CNOT gates with depth $O(\log N)$. Next, during the stage labelled $U_2$, the bus qubit is routed to the appropriate position by the quantum routers. Note that, because the destination of the bus is not known *a priori*, routing operations must be performed for all routers.

During the next stage, $U_3$, the classical data is copied to the state of the bus. This copying is accomplished with the aid of the circuit shown in Fig. 2.6. The circuit applies the Pauli operator $Z$ to a qubit conditioned on the classical value $x_i$. To see how this circuit implements the required data-copying operation, observe that the bus qubit in Fig. 2.5, initialized to $|0\rangle$, is mapped to $|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$ by the first Hadamard gate. For input state $|+\rangle$, the circuit of Fig. 2.6 leaves the state as $|+\rangle$ if $x_i = 0$ and flips the state to $|-\rangle$ if $x_i = 1$. Thus, the circuit encodes the classical value $x_i$ into the qubit state in the $\{|+\rangle, |-\rangle\}$ basis. See Appendix A for further details on this data-copying procedure, including an explanation for why data is copied in the $\{|+\rangle, |-\rangle\}$ basis, as opposed to $\{|0\rangle, |1\rangle\}$.

During the final two stages the bus qubit is routed back to its original position ($U_2^\dagger$), and the states of all routers are reset to $|0\rangle$ ($U_1^\dagger$). At the conclusion of the circuit, the bus qubit contains the data specified by the address register. Thus, the circuit implements the desired operation (2.63).

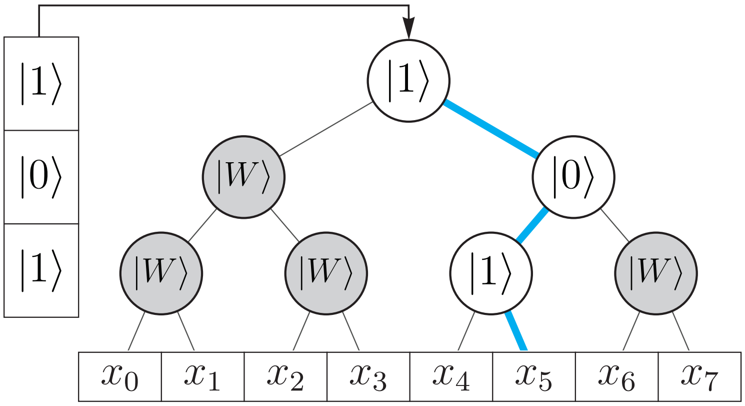


Figure 2.7: Bucket-brigade QRAM, utilizing routers with three sates: wait $|W\rangle$, route left $|0\rangle$, and route right $|1\rangle$. The address qubits themselves are routed into the tree, carving out a path to the memory.

### 2.2.3 Bucket-brigade QRAM

As we will describe in Chapter 3, the fanout QRAM architecture is impractical due to a high susceptibility to decoherence. Refs. [14] proposed the so-called "bucket-brigade" QRAM architecture as a potential solution to this decoherence problem. We describe the architecture in this section, deferring the discussion of decoherence to Chapter 3.

The bucket-brigade architecture of Ref. [14] is a variant of the fanout architecture with two major modifications. The first modification is that the two-level routing qubits are replaced with three-level routing qutrits. In addition to the $|0\rangle$ (route left) and $|1\rangle$ (route right) states, each router also has a third state, $|W\rangle$ (wait). We refer to the states $|0\rangle, |1\rangle$ as *active*, and the state $|W\rangle$ as *inactive*. We assume that all routers are initialized in the $|W\rangle$ state, and that the action of the routing operation is trivial when the routing qutrit is in the $|W\rangle$ state. Each router's incident and output modes are also now taken to be physical three-level systems, and each address qubit is encoded within a two-level subspace of a physical three-level system.

The second modification is that the address qubits are themselves routed into the tree during a query. When an address qubit encounters a router in the $|0\rangle$ ($|1\rangle$) state, it is routed to the left (right) as usual. When an address qubit encounters a router in the $|W\rangle$ state, the states of the router and incident mode are swapped, so that the router's state becomes $|0\rangle$ ($|1\rangle$) when the incident address was $|0\rangle$ ($|1\rangle$). The physical implementation described in Ref. [14] provides a helpful example to visualize how these operations could be realized: the authors envisage the routers as three-level atomic systems, with the address qubits encoded in the polarization states of flying photons. (Note that the two polarization states constitute a two-level subspace of a physical three-level system, since the photonic mode may also be in the vacuum state.) When a photon encounters an atom in the $|W\rangle$ state, it is absorbed, and in the process it excites the atom to the $|0\rangle$ or $|1\rangle$ state conditioned on its polarization. When subsequent photons encounter the excited atom, they are routed accordingly.

To query the memory, the address qubits are sequentially injected into the tree at the root node. The first address qubit is absorbed by the router at the root node, exciting it from $|W\rangle$ to the $\{|0\rangle, |1\rangle\}$ subspace in the process. When the second address qubit is injected into the tree, is routed left or right, conditioned on the state of the router at the root node. The state of the first address qubit thereby dictates the routing of the second. The second address is subsequently absorbed by one of the routers at the second level of the tree. The process is repeated, with the earlier addresses controlling the routing of later ones, carving out a path of active routers from the root node to the specified memory element. Once all address qubits have been routed into the tree, the bus qubit is routed down to the memory and the data is copied as before. Finally, the bus and all address qubits are routed back out of the tree in reverse order to disentangle the routers. Here again, we emphasize that multiple memory elements can be queried in superposition, as in Eq. (2.4), because all routing operations are performed coherently.

As with the fanout QRAM, the operation of the bucket-brigade QRAM can be equivalently described in the language of quantum circuits. We provide an example

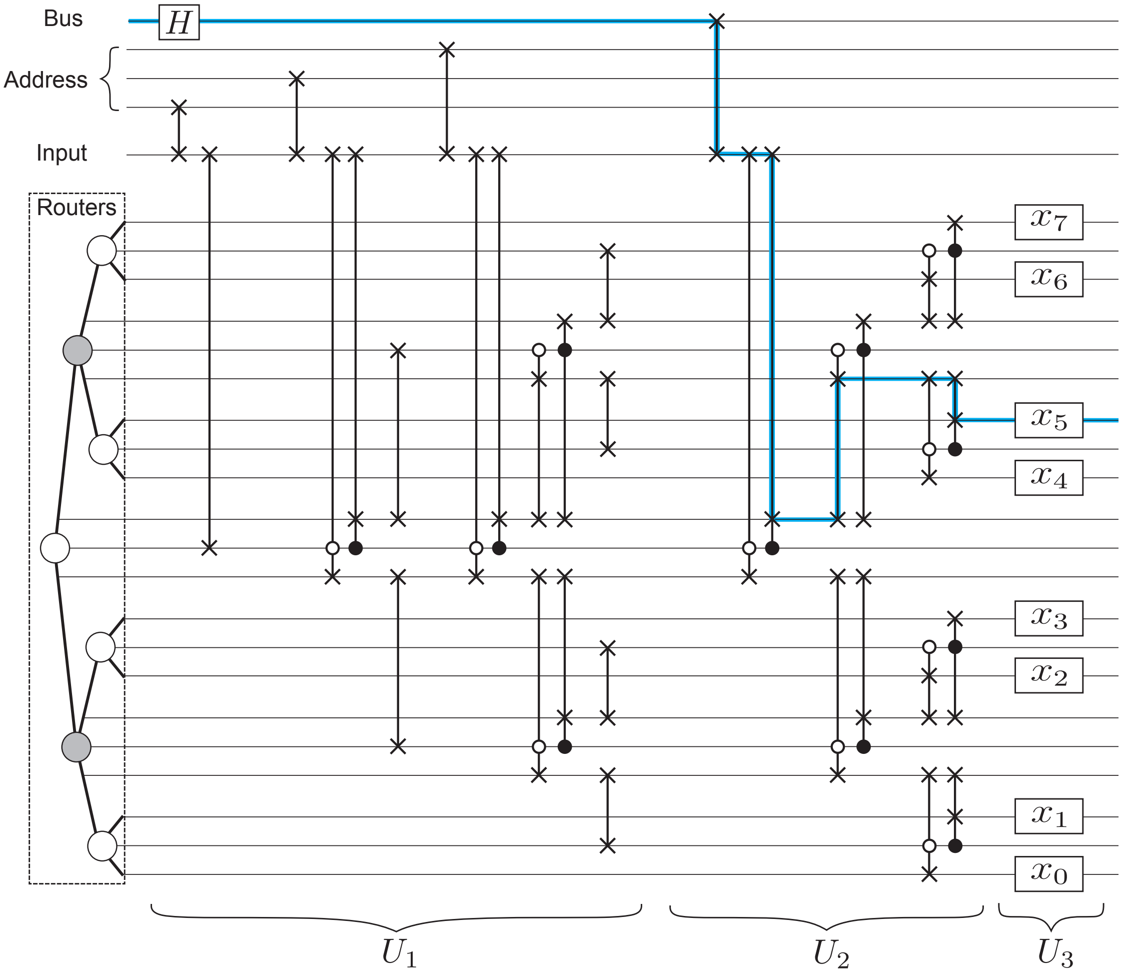


Figure 2.8: Bucket-brigade QRAM circuit for $N = 8$. The bus and address registers are indicated by rails at the top of the diagram, and the routers are indicated by the rails below. For each router shown on the left, there are three rails: one for the router's internal state, and two for the router's two output modes. All qubits comprising the routers are initialized to $|W\rangle$. The path of the bus is highlighted in blue for the case where the three address qubits are initialized to $|i\rangle = |101\rangle$. To complete the query, operations $U_2^\dagger$ and $U_1^\dagger$ must subsequently be applied, but we omit them here for clarity.

circuit for $N = 8$ in Fig. 2.8. The circuit is divided into several stages. During the first stage, labelled $U_1$, the address qubits are routed into the tree one by one. When the $\ell$-th address qubit reaches the incident port of a router at level $\ell$ of the tree, a swap gate is performed that exchanges the state of the router and its incident port. This way, the $\ell$-th address qubit is stored in a router at level $\ell$, and this router may be used to route subsequent address qubits to lower levels. This stage of the circuit can be performed in $O(\log N)$ depth [77].

The stages labelled $U_2$ and $U_3$ are the same as in the fanout QRAM; they route the bus qubit to the appropriate destination, then copy data to the bus (see Appendix A for further details on data copying). Next, the operation $U_2^\dagger$ is applied to route the bus back to its original location. Finally, to complete the query, $U_1^\dagger$ is applied to return the address qubits to their original locations and reset the routers to their initial states. For simplicity, these last two stages, $U_2^\dagger$ and $U_1^\dagger$, are omitted from Fig. 2.8.

### Quantum walk implementation

For context, we note that a variant of the bucket-brigade QRAM based on quantum walks was recently proposed in Ref. [78]. In that work, the bus consists of a quantum particle that executes a quantum walk on a binary-tree graph. In particular, the particle is imbued with a property, "chirality," that dictates whether it moves left or right at each vertex, and this property is controlled by the initial address in such a way that the bus is routed to the appropriate memory location as the quantum walk proceeds. We refer the interested reader to Ref. [78] for further details.

The authors of Ref. [78] claim three potential benefits of their scheme. The first is that the scheme is highly parallelized, such that queries only require $O(\log N)$ time to perform. While this does constitute an improvement over the original bucket-brigade QRAM papers [14, 15], which claimed $O(\log^2 N)$ query times, it matches the $O(\log N)$ query time of the circuit of Fig. 2.8.

The second claimed benefit is an improved resilience to decoherence relative to the standard bucket-brigade QRAM. The basis for this claim is that the quantum walk implementation does not utilize quantum routers, so the bus is never entangled with all $O(N)$ nodes of the binary tree. Instead, the bus is only ever entangled with the $O(\log N)$ qubits comprising the initial address. As a result, the infidelity of a query is expected to scale as $O(T \log N)$, where $T = O(\log N)$ is the query time. As we show in Chapter 3, however, the infidelity of the standard bucket-brigade QRAM has the same scaling; despite the fact that the standard bucket-brigade QRAM entangles the bus with $O(N)$ routers, the infidelity of a query only scales as $O(T \log N)$. Thus, in light of our results, the quantum-walk implementation does not seem to provide any advantage with respect to decoherence.

The third claimed benefit is that the use of quantum walks could allow for a simpler implementation of QRAM. For example, time-dependent control may not be required in the quantum-walk implementation [79], and it is conceivable that implementing QRAM via quantum walks could be easier in some experimental platforms. Further work is required to determine whether the quantum walk implementation would provide real practical benefits for these or other reasons.

### 2.2.4 QROM and hybrid architectures

The fanout and bucket-brigade architectures allow one to perform queries in $O(\log N)$ time using $O(N)$ qubits. These fast query times are essential for algorithms that must rapidly load large classical data sets in order to claim exponential speedups over their classical counterparts, e.g., quantum machine learning algorithms [10, 52, 53, 58, 71]. However, in algorithms that only require comparatively small data sets to be loaded, e.g. simulating local Hamiltonians [29, 46, 70, 73, 80, 81], slower query times can be sufficient. Circuits that use fewer qubits at the price of longer query times can be better suited for such algorithms.

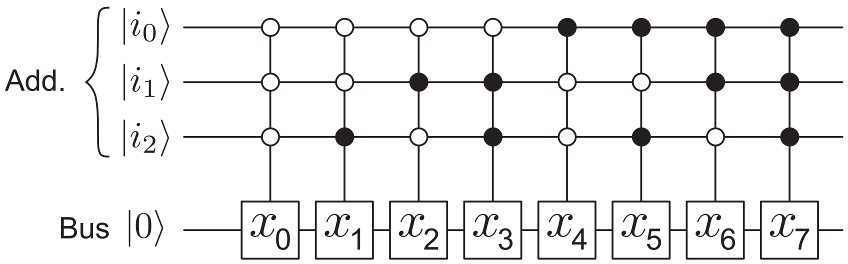


Figure 2.9: QROM circuit. The circuit implements operation (2.63) by iterating over all $N$ possible states of the address register. The $j$-th gate flips the state of the bus qubit if the address register (Add.) is in state $|j\rangle$ and $x_j = 1$, otherwise the gate acts trivially.

Indeed, this allocation of resources, $O(\log N)$ time and $O(N)$ qubits, represents one extreme; at the other extreme are architectures that perform queries in $O(N \log N)$ time using $O(\log N)$ qubits [81–83]. In fact, there exists a family of architectures that interpolate between these two extremes to leverage this space-time trade-off [18, 19, 36, 84]. We refer to these as *hybrid architectures*, and we describe them in this section.

Fig. 2.9 provides a straightforward example of a circuit that performs queries in $O(N \log N)$ time using only $O(\log N)$ qubits. To query a memory of size $N$, a sequence of $N$ multiply-controlled Toffoli gates is applied, where each gate has $\log N$ controls (the address qubits) and one target (the bus qubit). The circuit sequentially iterates over all $N$ possible addresses, flipping the bus qubit conditioned on the corresponding classical data. The circuit requires only $O(\log N)$ qubits, but it has depth $O(N \log N)$, since each multiply-controlled Toffoli gate can be performed in depth $O(\log N)$ [85]. Adopting the nomenclature introduced in Ref. [81], we refer to such circuits as Quantum Read-Only Memory (QROM)[5].

More generally, circuits can be constructed that trade longer query times for fewer qubits by combining QROM and QRAM, as shown in Fig. 2.10. We introduce a tunable parameter $M \leq N$, defined to be a power of 2. That is, $M = 2^m$, with $m$ an

[5] The terminology “read-only” is somewhat misleading. In this thesis, we only consider reading data from QRAM/QROM, as in Eq. (2.63). One could also use variants of QRAM/QROM to write to a classical memory, but writing multiple different elements to a *classical* memory *in superposition* is not possible.

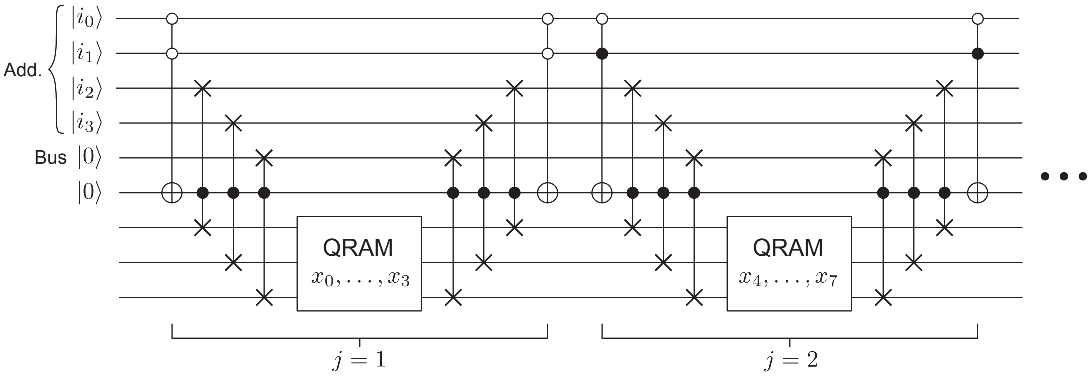


Figure 2.10: Hybrid circuit. All $M = 2^m$ possible states of the first $m$ address qubits are iterated over sequentially, as in QROM. Conditioned on these qubits, the remaining address qubits are used to query an $(N/M)$-cell classical memory via QRAM. In the circuit shown, $\log N = 4$ and $m = 2$. The boxes labelled QRAM implement (2.63), using either the fanout or bucket-brigade architecture. At the $j$-th iteration ($j \in [1, M]$), the data elements $\{x_{[(j-1)N/M]}, \ldots, x_{[j(N/M)-1]}\}$ are queried by the QRAM. Only the first two iterations are shown. The circuit depth is $O(M \log N)$, and the circuit uses $O(N/M + \log N)$ qubits, which includes the $O(N/M)$ ancillary qubits required by the QRAM (not shown).

integer in the interval $[0, \log N]$. The idea is to divide the full classical memory into $M$ blocks, each with $N/M$ entries. These blocks are queried one by one using a QRAM of size $N/M$ concatenated with a QROM-like iteration scheme. The total hardware cost of the scheme is $O(\log N + N/M)$, comprising $O(\log N)$ qubits for the address and bus registers and $O(N/M)$ ancillary qubits for the QRAM. The total circuit depth is $O(M \log N)$ because each of the $M$ iterations in the circuit can be performed in depth $O(\log N)$. Therefore, by tuning the parameter $M$, one can interpolate between large-width, small-depth circuits like QRAM, and small-width, large-depth circuits like QROM. The hybrid circuit reduces to QRAM for $M = 1$, and QROM for $M = N$.

## 2.3 Practical challenges

In this section, we describe some of the practical challenges associated with constructing a large-scale QRAM. Though the main results of this this thesis serve to mitigate some of these challenges, others remain, and the question of whether large-scale QRAM can be used to facilitate quantum speedups, either in principle or in practice, remains open.

It is important to note that most of the challenges described below only become seriously problematic for large memory sizes. For small or intermediate memory sizes, building a QRAM is not fundamentally different from building a universal quantum computer. After all, one could implement the operation $O_{\mathbf{x}}^{(\mathrm{DL})}$ simply by running the circuit Fig. 2.8 on a universal quantum computer. In fact, building a small-scale QRAM is arguably easier than building a universal quantum computer because QRAM does not require a universal gate set. As we describe below, however, at large memory sizes there arise a distinct set of practical challenges for QRAM. As a result, implementing a large-scale QRAM could prove significantly more difficult than implementing a fault-tolerant universal quantum computer. In this regard, the results of this thesis are encouraging; in the subsequent chapters we show that several of these challenges are not as problematic as was previously thought. Still, further work will be required to develop practical implementations of large-scale QRAM.

### 2.3.1 High quantum hardware cost

The central practical challenge associated with constructing a large-scale QRAM is the high quantum hardware cost: QRAM requires $O(N)$ qubits to query a memory of size $N$. This large overhead is an inevitable consequence of the fact that the query time is only $O(\log N)$. Indeed, a simple counting argument can be used to show that if the query time is $O(\log N)$ then the hardware overhead must necessarily be $O(N)$. (Of course, the hardware overhead can be reduced if longer query times can be tolerated, but this is unacceptable in some applications.)

This $O(N)$ overhead may be impractical in certain applications. For example, the relevant values of $N$ could easily reach millions or billions for quantum algorithms in big data or machine learning. At least as many qubits would be required to build the requisite QRAM. Scaling to this many qubits is a daunting engineering challenge, but it may be possible at some point in the future.

### 2.3.2 Decoherence, error correction, and fault tolerance

As a direct consequence of the large hardware overhead, QRAM implementations can be highly susceptible to decoherence. In naive implementations, the decoherence of even a single qubit can ruin an entire query. All qubits must then have decoherence rates $\varepsilon \ll 1/N$ if queries are to be performed with high fidelity. Additionally, if a QRAM is queried $Q$ times during the course of some algorithm, then the decoherence rates must further satisfy the requirement $\varepsilon \ll 1/(QN)$ because an error in a single query could, in principle, derail the entire algorithm. This becomes a very stringent requirement in applications where both $N \gg 1$ and $Q \gg 1$.

As we show in Chapter 3, this problem can be mitigated to an extent by employing the bucket-brigade QRAM architecture. With the bucket-brigade architecture, the required decoherence rate need only satisfy $\varepsilon \ll 1/(Q\,\mathrm{polylog}N)$, but even this requirement could still be challenging to satisfy. Indeed, error rates in current state-of-the art platforms are on the order of $\varepsilon \sim 10^{-3}$, and with such error rates one could only query a memory with $N = 100$ entries order $Q \sim 10$ times before decoherence becomes overwhelming. Therefore, some amount of quantum error correction (or error suppression, see Chapter 4) will likely be necessary for any application requiring more than a handful of queries [16].

In Chapter 4, we provide a detailed discussion of the challenges associated with implementing an error-corrected QRAM; we summarize this discussion here. Unfortunately, the use of quantum error correction for QRAM only serves to magnify QRAM's already large overhead. With error correction, QRAM requires $O(N)$ *logical* qubits to implement. Each logical qubit, in turn, can comprise a large number of physical qubits. While the hardware cost for error-corrected QRAM is still technically $O(N)$, the big-$O$ notation can hide a large prefactor—potentially several orders of magnitude. Furthermore, when the qubits comprising QRAM are error corrected, the logical operations performed between them must be implemented fault tolerantly. Unfortunately, the main operation used in QRAM—quantum routing—is a non-Clifford operation, so implementing this operation in the usual Clifford + T fault-tolerance model requires magic state distillation [86, 87]. The need for many magic state factories further inflates the overhead.

As we show in Chapters 4 and 5, the challenges associated with error correction can be mitigated to an extent. In Chapter 4 we show how the QRAM query infidelity can be suppressed without incurring an additional $O(N)$ overhead, and in Chapter 5 we propose hardware-efficient QRAM architectures that are compatible with low-overhead fault tolerance.

### 2.3.3 Long-range interactions

A classical data structure of size $O(N)$ is required to hold the classical data vector $\mathbf{x}$, and in order to access any part of this data structure in only $O(\log N)$ time, QRAM necessarily requires long-range interactions. Indeed, the need for long-range interactions is evident in Figs. 2.4 and 2.7. As in the figures, suppose that the classical data are stored in a one-dimensional data structure whose physical extent (i.e. its length) is $Nd$, where $d$ denotes the length of a single memory cell. The physical separation between adjacent routers at the lowest level of the tree is then only $d$, but this separation doubles at each higher level of the tree. Towards the top of the tree, routers are physically separated from one another by distances approaching $Nd$. Therefore, for sufficiently large $N$, long-range interactions will be required to connect routers at the top levels of the tree[6].

Some hardware platforms, such as Rydberg atoms [88, 89], boast native long-range

[6] Note that the distances can be reduced if a higher-dimensional data structure is used, but long-range interactions will still eventually be required.

interactions that could be exploited for this purpose. Alternatively, a large-scale QRAM can be built in a modular fashion, and entanglement between far-separated modules could be used to enable the requisite long-range connectivity [90–94]. Nevertheless, the need for long-range interactions may add additional hardware complexity that could further complicate the construction of a large-scale QRAM.

### 2.3.4 Fair comparisons with classical hardware

In order to claim that QRAM can facilitate a genuine quantum speedup for some application, a fair comparison must be made with comparable classical hardware. Refs. [30, 95] argue that, because QRAM assumes access to $O(N)$ quantum routers operating in parallel, it is appropriate to compare with a parallel classical computer with $O(N)$ processors. Indeed, if each quantum router requires a classical co-processor to control the routing operations or implement error correction, then $O(N)$ classical processors are already required to operate a QRAM. It is prudent to ask whether any purported quantum speedups still hold when assuming access to such highly-parallelized classical resources. Of course, the need to make a fair comparison with classical hardware does not affect our ability to construct a large-scale QRAM. Rather, it only constrains the potential applications where QRAM might be used to obtain a quantum speedup.

Consider the case of Grover's search algorithm as an example. Suppose a QRAM is used to search an unstructured database of size $N$. A marked element can be found in only $O(\sqrt{N})$ QRAM queries using Grover's algorithm [28]. Since each query takes $O(\log N)$ time, the total time required to find the marked element is $O(\sqrt{N}\log N)$. Given to access to a classical computer with $O(N)$ processors operating in parallel, however, it is possible to solve this same problem in only $O(\log N)$ time. One simply arranges the classical processors in a binary tree, and the address of the marked element can be passed up the tree in only $O(\log N)$ time. Thus, in this instance we see that the quantum device provides no speedup over a comparable classical device. The recent advent of so-called quantum-inspired classical algorithms [96–98] provides another example of this phenomenon. In contrast, simulating evolution under a local Hamiltonian is one example of an application where a fair comparison with classical hardware does not imperil the quantum speedup.

# Chapter 3

# Noise resilience of the bucket-brigade QRAM

The idea of QRAM has faced skepticism because of the practical challenges outlined in Section 2.3. Indeed, the question of whether QRAM can be used to facilitate quantum speedups, either in principle or in practice, has not been definitively settled. A central practical concern is the seemingly high susceptibility of QRAM to decoherence [14, 16]. As we discuss below, naive implementations of QRAM perform operation (2.63) with an infidelity that scales linearly with the size of the memory. Such implementations are not scalable. As the memory size increases, the infidelity grows rapidly without quantum error correction, yet the overhead associated with error correction can quickly become prohibitive because all $O(N)$ ancillary qubits need to be corrected [18].

Refs. [14, 15] proposed the bucket-brigade QRAM architecture as a potential solution to this decoherence problem, though this solution has also faced skepticism. Proponents argue that the bucket-brigade QRAM is highly resilient to noise, in that it can perform operation (2.63) with an infidelity that scales only polylogarithmically with the size of the memory. This favorable scaling could allow for high-fidelity queries of large memories without the need for quantum error correction, thereby mitigating the aforementioned scalability problem. This noise resilience, however, has only been derived for contrived noise models that place severe constraints on the quantum hardware [14–16], thus casting doubt on the viability of the bucket-brigade architecture. Indeed, while several proposals for experimental implementations of QRAM have been put forth [15, 17, 99–102], to our knowledge there has yet to be an experimental demonstration of even a small-scale QRAM[1]. Absent from this debate has been a fully general and rigorous analysis of how decoherence affects the bucket-brigade architecture.

In this chapter, we study the effects of generic noise on the bucket-brigade QRAM architecture. Our main result is that the architecture is far more resilient to noise than was previously thought (our main scaling results are summarized in Table 3.1). We rigorously prove that the infidelity scales only *polylogarithmically* with the memory size even when all components are subject to arbitrary noise channels, and we verify this scaling numerically. Remarkably (and perhaps counter-intuitively), this scaling holds even for noise channels where the expected number of errors scales *linearly* with the memory size. Our analysis reveals that this remarkable noise resilience is a consequence of the limited entanglement among the memory's components. We leverage this result to show that significant architectural simplifications can be made to the bucket-brigade QRAM, and that hybrid architectures [18, 19, 36, 84], which implement (2.63) with fewer qubits but longer query times, can also be made partially noise resilient. We also show that these benefits persist when quantum error correction is used. Importantly, the present work shows that a noise-resilient QRAM can be constructed from realistically noisy devices, paving the way for small-scale, near-term experimental demonstrations of QRAM.

[1] Note that the "random access quantum memories" demonstrated in Refs. [103–105] are distinct from QRAM; these experiments do not demonstrate the quantum addressing needed to perform operation (2.63).

| **Architecture** | **Infidelity scaling** |
|---|---|
| Fanout QRAM | $N \log N$ |
| Standard BB QRAM | $\log^2 N$ |
| Two-level BB QRAM | $\log^3 N$ |
| Hybrid fanout | $N \log N + M \log^2 N$ |
| Hybrid BB | $M \log^2 N$ |

Table 3.1: Infidelity scalings of QRAM architectures. $N$ denotes the size of the classical memory being queried, and bucket-brigade is abbreviated as BB. The first three architectures have circuit depth $O(\log N)$ and require $O(N)$ qubits. For the hybrid architectures, $M \leq N$ is a tunable parameter that determines the circuit depth, $O(M \log N)$, and the number of qubits, $O(N/M + \log N)$.

This chapter is organized as follows. In Section 3.1, we review prior works that studied the effects of noise on QRAM. The main result of this chapter is presented in Section 3.2: we prove that the query infidelity of the bucket-brigade architecture scales only polylogarithmically when its components are subject to generic mixed-unitary error channels (the full proof for arbitrary error channels is given in Ref. [77]). Importantly, these proofs assume that *all* components of the QRAM (both active and inactive) are susceptible to decoherence, in contrast to prior works. Next, in Section 3.3, we discuss various implications and extensions of this result. We show that the use of three-level memory elements in the original bucket-brigade architecture is superfluous and that the architecture can be significantly simplified (while maintaining noise resilience) by instead using two-level memory elements. We also show that the bucket-brigade architecture can also be employed to imbue hybrid architectures with partial noise resilience. Additionally, we prove that error-corrected implementations of the bucket-brigade architecture are resilient to logical errors, and we discuss the practical utility of error-corrected QRAM. Finally, in Section 3.4 we conclude by discussing the implications of these results in the context of potential algorithmic applications.

The results in this chapter are primarily based on Ref. [77]: CTH et al., Resilience of quantum random access memory to generic noise, PRX Quantum 2, 020311 (2021).

## 3.1 Prior studies of noise in QRAM

In this section, we review earlier results concerning the effects of noise on QRAM. We begin by illustrating that the fanout architecture is highly susceptible to decoherence and hence impractical. This impracticality motivated the development of the bucket-brigade QRAM. While earlier works claimed that the bucket-brigade architecture could be resilient to decoherence, all of the earlier arguments rested on the problematic notion of decoherence-free inactive routers; we review these arguments and their shortcomings.

### 3.1.1 Effects of noise on the fanout QRAM

The fanout architecture is impractical due to its high susceptibility to decoherence. In this architecture, each address qubit is maximally entangled with all routers at the respective level of the tree, similar to a GHZ state. As a result, the decoherence of any individual router is liable to ruin a query. As an example, suppose that the routers are subject to amplitude-damping errors. The loss of an excitation from any router at level $\ell$ collapses all other level-$\ell$ routers—and the $\ell$-th address qubit—to the $|1\rangle$ state. Any terms in the superposition where the $\ell$-th address qubit was in the $|0\rangle$ state prior to the error are thus projected out, thereby reducing the fidelity by a factor of 2 on average.

More generally, suppose that each router suffers an error with probability $\varepsilon$ at each time step during the query. The final state $\Omega$ of the full system (address, bus, and routers) can then be written as a statistical mixture

$$\Omega = (1-\varepsilon)^{T(N-1)}\,\Omega_{\text{ideal}} + \dots \tag{3.1}$$

where $\Omega_{\text{ideal}}$ is the error-free state, $T = O(\log N)$ is the number of time steps required to perform a query, and "…" denotes all states in the mixture where at least one of the $N-1$ routers has suffered an error. We define the *query fidelity* as

$$F = \langle \psi_{\text{out}} | \mathrm{Tr}_R\left(\Omega\right) | \psi_{\text{out}} \rangle\,, \tag{3.2}$$

where $\mathrm{Tr}_R$ indicates the partial trace over the routers. The routers are traced out because only the address and bus registers are passed on to whatever algorithm has queried the QRAM; the routers are ancillae whose only purpose is to facilitate the implementation of $O_{\mathbf{x}}^{(\text{DL})}$.

As illustrated by the amplitude-damping example, the problem with the fanout implementation is that the no-error state $\Omega_{\text{ideal}}$ is generally the only state in the mixture (3.1) with high fidelity. Neglecting the low-fidelity states, the query infidelity scales as

$$1 - F \sim \varepsilon N T, \tag{3.3}$$

to leading order in $\varepsilon$. We refer to this linear scaling of the infidelity with the memory size as *unfavorable* because error probabilities $\varepsilon \ll 1/NT$ are required to perform queries with near-unit fidelity. This stringent requirement severely constrains the size of fanout QRAMs. For example, error probabilities $\varepsilon \sim 10^{-3}$ would restrict the maximum size of a high-fidelity fanout QRAM to less than $N \sim 100$ memory cells. While quantum error correction can be used to suppress the error rates in principle, the additional hardware overhead can be prohibitive [18] because all $O(N)$ routers must be error corrected. Thus, because of its high susceptibility to decoherence, the fanout architecture is not regarded as scalable.

### 3.1.2 Effects of noise on the bucket-brigade QRAM

The bucket-brigade QRAM architecture was originally proposed as a means to overcome practical challenges associated with noise. As described in Section 2.2.3, all quantum routers in the bucket-brigade architecture have three states, two active states ($|0\rangle$ = route left, $|1\rangle$ = route right) and one inactive state ($|W\rangle$ = wait). At the beginning of a query, all routers are initialized to the inactive $|W\rangle$ state. As the address qubits are routed into the tree, $\log N$ routers are excited to active states, while the remaining $(N-1)-\log N$ routers remain in the inactive state[2]. This limited number of active routers is central to all prior arguments for the bucket-brigade architecture's noise resilience.

To demonstrate the noise resilience of the bucket-brigade QRAM, the original papers [14, 15] adopt an error model where only active routers are prone to decoherence. For example, the active states of a router could correspond to some energetically excited states with finite lifetimes, whereas the inactive state could correspond to a relatively stable ground state. Alternatively, the process of exciting a router to an active state could be very noisy in comparison to simply idling in the inactive state. Whatever the justification, if only active routers are prone to decoherence, then it follows from the limited number of active routers that the bucket-brigade architecture is resilient to noise. For example, Ref. [16] studied the bucket-brigade QRAM with routers subject to $|0\rangle \leftrightarrow |1\rangle$ bit-flip errors, with the $|W\rangle$ states assumed to be error free. In this case, the expected number of errors is only $\varepsilon \log N$, because only the $\log N$ active routers are prone to errors. The expected number of errors also scales

[2] When multiple different memory elements are queried in superposition, each router is generally in a superposition of active and inactive states. Only $\log N$ routers are active within in each branch of the superposition corresponding to a definite address.

with $\log N$ for the error model considered in Refs. [14, 15, 17], where gates involving only inactive routers are assumed to be error free.

For error models where only active routers are prone to decoherence, the query infidelity is

$$1 - F \sim \varepsilon T \log^{\alpha} N. \tag{3.4}$$

to leading order in $\varepsilon$, where $\alpha$ is some constant, and we recall that $T = O(\log N)$ is the number of time steps. We refer to this logarithmic scaling of the infidelity with the memory size as *favorable* because queries can be performed with near-unit fidelity so long as the error rate satisfies $\varepsilon \ll 1/T \log^{\alpha} N$. This is a much more forgiving requirement; memories of exponentially larger size can be queried relative to the fanout architecture. Indeed, the exponential improvement in scalability suggests that quantum error correction is not required to query large memories with high fidelity, provided physical error rates are sufficiently low.

Unfortunately, the above error models can be poor approximations of the noise in actual quantum hardware. In these contrived models, inactive routers are assumed to be completely free from decoherence. More realistically, all routers will be prone to decoherence, independent of whether they are active or inactive. For example, though several proposals for experimental implementations of the bucket-brigade scheme have been put forth [15, 17, 99–101], none have proposed a method of engineering routers that are free from decoherence when inactive. While one can conceive of implementations in which inactive routers have decoherence rates which are nonzero but far smaller than those of active routers, it is not obvious whether such implementations would enjoy the favorable infidelity scaling. Indeed, Ref. [14] conjectured that decoherence of inactive routers could significantly increase the infidelity in this case, owing to the exponentially larger number of inactive routers. Furthermore, Refs. [14, 16] portray the favorable infidelity scaling as a direct consequence of the assumption that inactive routers are decoherence-free.

Additionally, the above error models have troubling implications in the context of quantum error correction. Suppose that we try to use quantum error correction to further suppress errors in the bucket-brigade QRAM architecture. Each of the physical qubits comprising a quantum router must then be replaced by an error-corrected logical qubit in order to yield an error-corrected logical router. However, as argued in Ref. [16], quantum error correction is an *active* process; one must actively check for errors and correct them when they occur. As a result, even when an error-corrected quantum router is idling in the logical $|W\rangle$ state, it is very much active at the physical level. Thus, when quantum error correction is used, all routers should be regarded as physically active. If the noise resilience of the bucket-brigade QRAM is to be attributed to the limited number of active routers (as is done in Refs. [14–17, 52, 53]), then the use of quantum error correction evidently undermines this resiliency; the query infidelity would then revert to the same $1 - F \propto N$ scaling of the fanout QRAM. One is led to the paradoxical conclusion that the use of quantum error correction can actually reduce the query fidelity.

These two factors—the necessity of incorporating more realistic error models and the paradoxical implications of prior analyses for quantum error correction—motivate a critical re-examination of the effects of noise on the bucket-brigade QRAM. In particular, it is prudent to ask whether the favorable scaling still holds when inactive routers are not assumed to be decoherence-free. Relaxing this assumption causes the expected number of errors to increase *exponentially*, from $O(\log N)$ to $O(N)$. Because the expected number of errors in the fanout architecture is also $O(N)$, one might naively expect that the favorable infidelity scaling no longer holds. However, in the next section we prove that this is not the case. Perhaps surprisingly, the infidelity of the bucket-brigade architecture still scales favorably despite the exponential increase in the expected number of errors. Moreover, the favorable scaling holds for *arbitrary* error channels.

## 3.2 Noise resilience of the bucket-brigade QRAM

In this section, we prove that the bucket-brigade QRAM's query infidelity scales only polylogarithmically with the memory size, even when all routers (both active and inactive) are subject to decoherence. We begin by providing an intuitive explanation for this result based on entanglement within the bucket-brigade architecture. Then, by carefully analyzing the propagation of errors within the QRAM, we derive an upper bound on the query infidelity. Finally, we classically simulate QRAM circuits with routers subject to a variety of realistic error channels in order to verify this bound.

### 3.2.1 Intuition

The noise resilience of the bucket-brigade architecture can be understood intuitively as a consequence of the minimal entanglement among the routers, see Fig. 3.1. Suppose one queries all memory locations in equal superposition. Then in both the fanout and bucket-brigade architectures, all of the routers are entangled. However, the degree to which each router is entangled with the rest of the system is quite different between the two architectures. This difference can be quantified by computing the entanglement entropy for a given router

$$S(\rho) = -\mathrm{Tr}\left[\rho \log \rho\right] \tag{3.5}$$

where $\rho$ is the reduced density matrix of the router, obtained by tracing out the rest of the system. In the fanout architecture, each router is maximally entangled with the rest of the system; the reduced density matrix is the maximally mixed state $\rho = I/2$ (recall the fanout architecture employs two-level routers), for which $S(\rho) = 1$. In contrast, in the bucket-brigade architecture, the entanglement entropy of a router depends on its location within the tree. A router at level $\ell$ (0-indexed) of the tree is only active in $N2^{-\ell}$ of the $N$ different branches of the superposition. As a result, the entanglement entropy decreases exponentially with depth, $S(\rho) \sim 2^{-\ell}$.

Routers deeper down in the tree are nearly disentangled from the system, and their decoherence only reduces the query fidelity by an exponentially decreasing amount. Thus, despite the fact that exponentially many such errors typically occur, the overall fidelity can remain high. More precisely, if we posit that the infidelity associated with an error in a router at level $\ell$ scales as $\sim 2^{-\ell}$ due to the limited entanglement, and that $\varepsilon T\, 2^{\ell}$ such routers suffer errors on average, then the total infidelity scales as

$$1 - F \sim \sum_{\ell=1}^{\log N} \left(2^{-\ell}\right)\left(\varepsilon T 2^{\ell}\right) = \varepsilon T \log N. \tag{3.6}$$

The infidelity scales only logarithmically with $N$ because the exponential increase in the expected number of errors with $\ell$ is precisely cancelled by the exponential decrease in the infidelity associated with each. We rigorously justify these claims in the next section.

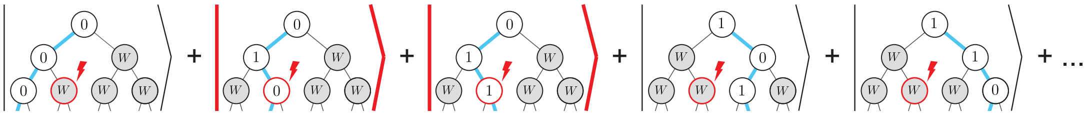


Figure 3.1: Conceptual picture of noise resilience. Each ket represents the state of the QRAM when a different memory element is queried, with the superposition of kets representing a superposition of queries to different elements. When a router $r$ suffers an error (red lightning bolt), it corrupts only the subset of queries where $r$ is active (indicated by thick red kets); other queries in the superposition succeed regardless. Because most routers are only active in a small fraction of queries, most queries succeed and the total infidelity is low.

### 3.2.2 Proof of noise resilience

In this section, we prove that the query infidelity of the bucket brigade architecture is upperbounded by

$$1 - F \leq A\varepsilon T \log N, \tag{3.7}$$

where $T = O(\log N)$ is the time required to perform a query, $\varepsilon$ is the probability of error per time step, and $A$ is a constant of order 1. This bound holds even when all $N$ memory elements are queried in superposition, and it holds for arbitrary error channels, including, e.g., depolarizing errors and coherent errors. Moreover, we assume no special structure in the classical data $x_i$, so our bounds hold independent of the data.

Our proof is based on a careful analysis of how errors can propagate throughout the QRAM. Accordingly, we begin by defining our error model. We suppose that each routing qutrit is subject to an error channel in the form of a generic completely-positive trace-preserving map,

$$\rho \to \mathcal{E}(\rho) = \sum_i K_m \rho K_m^\dagger, \tag{3.8}$$

where the Kraus operators $K_m$ obey the completeness relation $\sum_m K_m^\dagger K_m = I$. The error channel is applied simultaneously to all routers at discrete time steps throughout the query (see Eq. (3.14) below). In Ref. [77], we prove that the bound (3.7) holds for arbitrary error channels of the form (3.8). For the sake of brevity and simplicity, however, here we restrict our attention to channels where (i) there is a no-error Kraus operator, $K_0$, that is proportional to the identity, and (ii) the remaining Kraus operators are proportional to unitaries, $K_m^\dagger K_m \propto I$. Under these restrictions,

$$\mathcal{E}(\rho) = (1-\varepsilon)\rho + \sum_{m>0} K_m \rho K_m^\dagger, \tag{3.9}$$

for some $\varepsilon \in [0,1]$. An operational interpretation of this channel is that one of the errors $K_{m>0}$ occurs with probability $\varepsilon$, and no error occurs with probability $1-\varepsilon$. Experimentally relevant examples include bit-flip, dephasing, and depolarizing channels. The restriction to this form of mixed-unitary channel allows us to make two assumptions that greatly simplify the proof: (i) the probability that an error occurs is independent of the router state, and (ii) the no-error backaction $K_0 \propto I$ is trivial. We make no further assumptions about the Kraus operators, and we stress that they may act non-trivially on the inactive state $|W\rangle$, meaning that inactive routers can decohere.

It is important to note that this error model only describes decoherence of the routing qutrits; a router's incident and output modes may also decohere, and there may be errors in the gates that implement the routing operation. At the end of this section, we prove that the bound (3.7) still holds when including these other errors, but we neglect them for now to simplify the discussion.

The proof proceeds by direct calculation. To bound the infidelity, we first write the final state $\Omega$ as a sum over different *error configurations*,

$$\Omega = \sum_c p(c)\Omega(c), \tag{3.10}$$

where an error configuration $c$ specifies which Kraus operator is applied to each router at each time step. Here, $p(c)$ is the probability of configuration $c$, and the pure state $\Omega(c) = |\Omega(c)\rangle\langle\Omega(c)|$ is the corresponding final state of the system (both quantities are defined more formally below). The fidelity is thus given by,

$$F = \sum_c p(c)F(c), \tag{3.11}$$

where

$$F(c) = \langle\psi_{\text{out}}|\text{Tr}_R\Omega(c)|\psi_{\text{out}}\rangle \tag{3.12}$$

is the query fidelity of the state $\Omega(c)$. Our approach is to place an upper bound on the infidelity by deriving an upper bound on $1 - F(c)$.

Let us formally define $\Omega(c)$ and $p(c)$. A QRAM query consists of $O(N)$ routing operations performed in a predetermined sequence. By design, many of these oper-

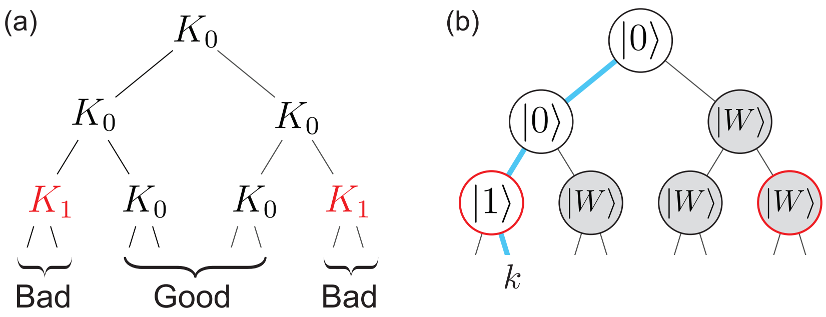


Figure 3.2: Error configurations. (a) Example composite Kraus operator $K_{c(t)}$. The single-router Kraus operators $K_{c(r,t)}$ comprising the tensor product $K_{c(t)}$ are arranged geometrically according to the routers on which they act. Branches of the tree are classified as either good or bad according to the locations of the errors $K_{m>0}$. (b) Query to an element $k \notin g(c)$. Routers are labelled with their ideal, error-free states, and routers outlined in red suffer errors. Because one of the active routers suffers an error, the query is liable to fail.

ations commute and can be performed in parallel, so that the entire operation can be written as a quantum circuit with depth $T = O(\log N)$ (see circuit diagram in Fig. 2.8). More precisely, a bucket-brigade QRAM query can be written as,

$$|\psi_{\text{out}}\rangle \, |\mathcal{W}\rangle = U_T \dots U_2 U_1 \, |\psi_{\text{in}}\rangle \, |\mathcal{W}\rangle \, , \tag{3.13}$$

where $|\mathcal{W}\rangle = |W\rangle^{\otimes(N-1)}$ is the initial state of the routers, and $U_t$ is a constant-depth circuit. Now, let $K_{c(r,t)}$ denote the Kraus operator applied to router $r$ at time step $t$, and define the composite Kraus operator $K_{c(t)} = \bigotimes_{r=1}^{N-1} K_{c(r,t)}$ [see Fig. 3.2(a)]. The final state $|\Omega(c)\rangle$ is

$$|\Omega(c)\rangle = \frac{1}{\sqrt{p(c)}} \left[ U_T K_{c(T)} \dots U_1 K_{c(1)} \right] |\psi_{\text{in}}\rangle \, |\mathcal{W}\rangle \, , \tag{3.14}$$

The requirement that $|\Omega(c)\rangle$ is normalized defines the probability $p(c)$ of obtaining state $\Omega(c)$ in the mixture (3.10). Note that $\sum_c p(c) = 1$ follows from the Kraus operators' completeness relation.

For a given error configuration $c$, it is convenient to classify branches of the tree as either *good* or *bad*, depending on whether errors $K_{m>0}$ are ever applied to the routers in the branch [Fig. 3.2(a)]. More precisely, let $\mathbf{i}$ denote the set of all routers in the $i$-th branch of the tree (corresponding to address $i$), and let $\mathbf{c}$ denote the set of all routers which have an error $K_{m>0}$ applied to them at some time step. A branch $i$ is defined to be good if $\mathbf{i} \cap \mathbf{c} = \varnothing$, and bad otherwise. To keep the notation simple, we use $g(c)$ to denote set of good branches. As illustrated in Fig. 3.2(b), queries to addresses $i \notin g(c)$ are liable to fail because they rely on routers that suffer errors.

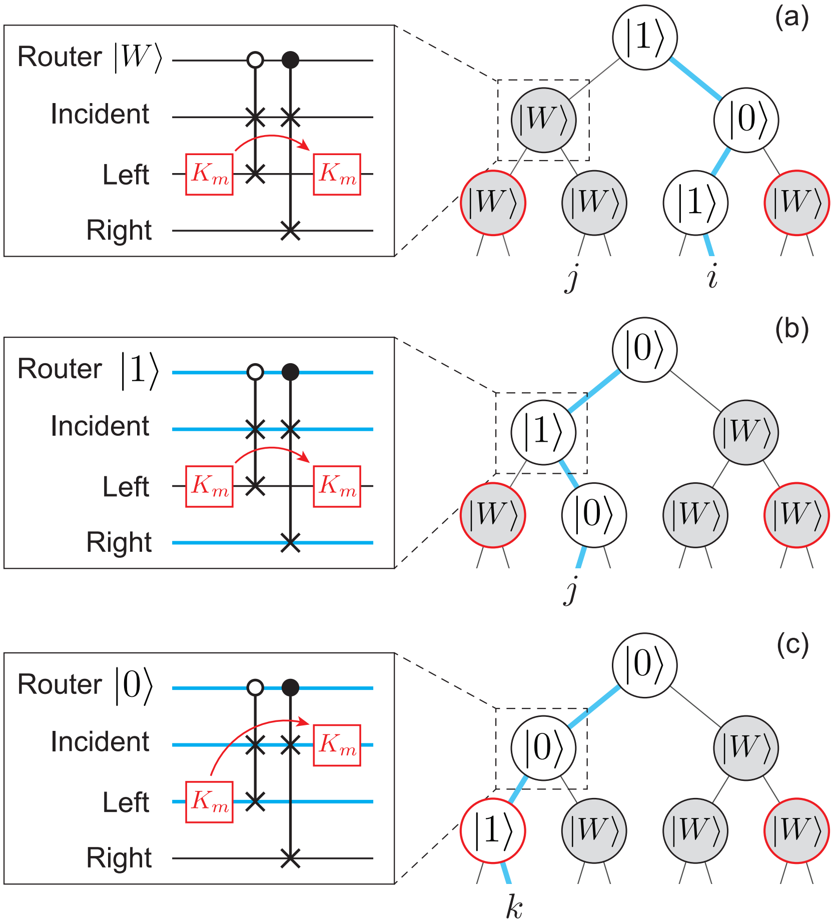


Figure 3.3: Error propagation. (a,b) Constrained propagation during queries to elements $\in g(c)$. The error in the leftmost router can propagate upward into the left output of the router indicated by the dashed box. The circuits on the left show that the error does not propagate further, regardless of whether the router is inactive (a) or active (b). In the circuit diagrams, red boxes denote errors $K_{m>0}$, and the red arrows indicate how the error propagates (i.e. how the error transforms under conjugation by the routing operation). (c) Error propagation is not constrained during queries to elements $\notin g(c)$. Note that the state of the router dictates how the error propagates in these examples.

The main observation underlying our proof is that the propagation of errors is constrained when memory elements $\in g(c)$ are queried. Roughly speaking, errors do not propagate from bad branches into good branches. More precisely, for any $i, j \in g(c)$, errors do not propagate into branch $j$ during a query to element $i$. We illustrate this fact with two examples, shown in Figs. 3.3(a,b). In general, errors in the bad branches can propagate. They can even propagate into an output mode of a router $r$ in branch $j \in g(c)$, but they can never propagate into branch $j$. Fig. 3.3(a) shows an example of how such an error propagates through $r$'s routing operation in the case where a memory element $i \neq j$ is queried. Because $j \in g(c)$, $r$'s routing qutrit suffers no errors and is thus in $|W\rangle$. The action of the routing operation is trivial for a router in $|W\rangle$, so the error does not propagate to other modes. (We reiterate that we are assuming error-free gates; gate errors are discussed at the end of this section.) Similarly, Fig. 3.3(b) shows an example of how errors propagate in the case where $j$ is queried. The error-free routing qutrit is in $|1\rangle$, so the routing operation acts non-trivially on only the incident and right output modes. The error in the left output mode does not propagate upward. For comparison, in Fig. 3.3(c) we illustrate that the propagation of errors is not constrained in this way when memory elements $k \notin g(c)$ are queried. As an aside, we note that the constrained error propagation can be understood as a sort of error transparency [106–108]: when elements ∈ g(c) are queried, the errors in the bad branches commute with the routing operations in the good branches.

The constrained propagation of errors has two important consequences. The first is that a query to memory element $i \in g(c)$ always succeeds, meaning that the address and bus registers are in the desired state $|i\rangle^A |x_i\rangle^B$ at the end of the query. This follows from the fact that errors cannot propagate to any of the routers in branch $i$. The second consequence is that, if multiple memory elements $i, j, \ldots \in g(c)$ are queried in superposition, the address and bus registers are disentangled from the routers at the end of the query. This follows from the fact that errors are restricted to propagate within the bad branches, and their propagation is unaffected by routers outside these branches. Figs. 3.3(a,b) provide an example. As a result, even though errors can propagate non-trivially among the bad branches during the query, the final state of the routers is independent of which memory element in $g(c)$ is queried.

It follows that the final state $|\Omega(c)\rangle$ can be written as

$$|\Omega(c)\rangle = |\text{good}(c)\rangle + |\text{bad}(c)\rangle\,, \tag{3.15}$$

with

$$|\text{good}(c)\rangle = \left(\sum_{i\in g(c)} \alpha_i\,|i\rangle^A\,|x_i\rangle^B\right)|f(c)\rangle^R\,. \tag{3.16}$$

Here, $|f(c)\rangle^R$ denotes the final state of the routers with respect to the good branches, and $|\text{bad}(c)\rangle$ contains the $i \notin g(c)$ terms. We now use the expression (3.15) to place a lower bound on $F(c)$. First notice that

$$F(c) \geq |\langle\psi_{\text{out}}, f(c)|\Omega(c)\rangle|^2\,, \tag{3.17}$$

which can be obtained by performing the partial trace in Eq. (3.12) using a basis that contains the state $|f(c)\rangle$ and neglecting the contributions from other states. Then, defining $\Lambda(c)$ as the weighted fraction of good branches,

$$\Lambda(c) = \langle\text{good}(c)|\text{good}(c)\rangle = \sum_{i\in g(c)} |\alpha_i|^2 \tag{3.18}$$

we have that

$$\langle\psi_{\text{out}}, f(c)|\text{good}(c)\rangle = \Lambda(c) \tag{3.19}$$

$$|\langle\psi_{\text{out}}, f(c)|\text{bad}(c)\rangle| \leq 1 - \Lambda(c). \tag{3.20}$$

To obtain the inequality (3.20) we have used the fact that $|\Omega(c)\rangle$ is normalized and that $\langle \mathrm{good}(c)|\mathrm{bad}(c)\rangle = 0$. The latter follows from the orthogonality of different initial address states, $\langle i|j\rangle^A = 0$ for $i \neq j$, and the fact that all subsequent operations, including the Kraus operators, are unitary and thus preserve inner products (this follows from our earlier restriction to mixed-unitary error channels; general channels are covered by the proof in Ref. [77]). Plugging Eqs. (3.15), (3.19) and (3.20) into the bound (3.17) and applying the reverse triangle inequality allows us to bound the infidelity as a function of $\Lambda(c)$,

$$F(c) \geq \begin{cases} (2\Lambda(c)-1)^2, & \Lambda(c) \geq 1/2, \\ 0, & \Lambda(c) < 1/2. \end{cases} \tag{3.21}$$

To proceed further, we compute the expected fraction of good branches, $\mathbb{E}(\Lambda)$, where the expectation value is taken with respect to the distribution of error configurations, i.e. $\mathbb{E}(f) = \sum_c p(c) f(c)$. This expectation value can be computed recursively for trees of increasing depth. Let $\mathbb{E}_d(\Lambda)$ denote the expected fraction of good branches for a depth-$d$ tree. For a depth-1 tree, expected fraction is equivalent to the probability that the lone router never suffers an error, $\mathbb{E}_1(\Lambda) = (1-\varepsilon)^T$. For deeper trees, the expected fraction of error-free routers at each level is $(1-\varepsilon)^T$, so we have the recursive rule

$$\mathbb{E}_{d+1}(\Lambda) = (1-\varepsilon)^T \mathbb{E}_d(\Lambda). \tag{3.22}$$

Applying this rule to the initial condition $\mathbb{E}_1(\Lambda)$, we obtain

$$\mathbb{E}_{\log N}(\Lambda) = (1-\varepsilon)^{T \log N}. \tag{3.23}$$

We can now combine the above results to bound the infidelity. We have that

$$F = \mathbb{E}(F) \geq \mathbb{E}(\sqrt{F})^2 \tag{3.24}$$

$$\geq [2\mathbb{E}_{\log N}(\Lambda) - 1]^2 \tag{3.25}$$

$$= \left[2(1-\varepsilon)^{T\log N} - 1\right]^2, \tag{3.26}$$

where the second inequality follows from (3.21) under the assumption that $\mathbb{E}(\Lambda_{\log N}) \geq 1/2$. Applying Bernoulli's inequality yields the desired result,

$$1 - F \leq 4\varepsilon T \log N, \tag{3.27}$$

which holds for $\varepsilon T \log N \leq 1/4$. This bound is our main result, and we stress that it holds even when all $N$ elements are queried in superposition, and that it was derived under the assumption that all routers are susceptible to decoherence, regardless of whether they are active or inactive.

We offer two additional remarks on the proof. First, we reiterate that while the above proof holds only for mixed-unitary error channels, in fact the favorable infidelity scaling holds for arbitrary error channels, which we prove in Ref. [77]. Second, the favorable scaling can be interpreted as a consequence of the limited entanglement among the routers, as discussed in Section 3.2.1. This limited entanglement manifests in Eqs. (3.15) and (3.16). The fact that a router at level $\ell$ is active in only $N2^{-\ell}$ of the $N$ branches implies both that the router's entanglement entropy decreases exponentially with $\ell$, and that only $N2^{-\ell}$ branches are corrupted when it suffers an error.

We conclude this section by describing four simple extensions of the proof that cover other cases of interest:

*1. Initialization errors.* Suppose that each router has some probability $\varepsilon$ of not being initialized to $|W\rangle$ prior to the query. Such errors can be viewed as router errors of the form (3.8) that occur during the 0-th time step. As such, they are also covered by the proof provided one replaces $T \to T+1$ in the equations above. In Section 3.3.1, we show that, in fact, one can make an even stronger statement: the infidelity scales favorably even when the QRAM is initialized in an arbitrary state.

*2. Gate errors.* Faulty implementation of the routing operation can be described without loss of generality as a composition $\mathcal{D} \circ \mathcal{R}$, of some error channel $\mathcal{D}$ followed by the ideal routing operation $\mathcal{R}$. Provided that $\mathcal{D}$'s Kraus operators are proportional to unitaries, and that there is a no-error Kraus operator proportional to the identity, then $\mathcal{D}$ can also be written in the form (3.9), and the proof proceeds as above. Note that the propagation of errors is still constrained in the case of gate errors because all routing gates in good branches are error-free by construction.

*3. Alternate gate sets.* We have defined the routing operation as a sequence of two controlled-SWAP gates [Fig. 2.3], but this same operation could also be decomposed into other types of gates, e.g. into Toffolis, or Clifford + T gates. The bound (3.27) holds for any choice of gate decomposition. To see that the bound holds, consider that any error that propagates non-trivially through a given routing operation can be categorized as occurring either *before* or *during* that operation. The propagation of errors that occur before the operation is determined solely by the conjugation of the error with the entire routing operation (Fig. 3.3), which is unaffected by the choice of decomposition. In contrast, the propagation of errors that occur during the operation will generally depend on the choice of the decomposition. However, such errors can equivalently be described as a faulty implementation of the routing operation itself, so they do not spoil the favorable error scaling by the argument in the previous paragraph.

*4. Correlated errors.* The noise resilience also persists in the presence of correlated errors that afflict a constant number of adjacent routers in the tree. The proof assumes that if any error (correlated or otherwise) occurs in a branch, then that branch does not contribute to the fidelity. As such, whether an error afflicts only a given router $r$ or also some of $r$'s child routers lower in the tree is irrelevant to the proof. The effects of correlated errors can thus be incorporated simply by augmenting $\varepsilon$ to also include the probability that a router is among those afflicted by a correlated error. For correlated errors afflicting only a constant number of adjacent routers, the resulting increase in $\varepsilon$ is independent of $N$, so the query infidelity still scales only polylogarithmically with $N$.

### 3.2.3 Classical simulation of noisy QRAM circuits

In this section, we verify the bound (3.27) through numerical simulation of noisy QRAM circuits. While full state vector simulations require $\exp(N)$ memory and quickly become intractable as the QRAM size grows, our simulations are enabled by a novel classical algorithm with space and time complexity $\text{poly}(N)$.

The main observation underlying the algorithm is that any quantum circuit consisting of the following elements can be simulated efficiently classically: state preparation in the computational basis, and gates from the set {SWAP, controlled-SWAP}. Such circuits are essentially classical—the system begins in a definite computational basis state, and the SWAP-type gates act only as permutations so that the system remains in a computational basis state through every step of the circuit. The simulation proceeds simply by tracking the (classical) state of the system. Furthermore, for initial states that are a superposition of polynomially-many different computational basis states, it follows from linearity that the action of any circuit composed of these SWAP-type gates can also be efficiently simulated. QRAM circuits can thus be efficiently simulated because they consist of SWAP-type gates acting on $O(N)$ qubits or qutrits, and the system is initialized in a superposition of only $O(N)$ computational basis states (one for each address). In fact, QRAM circuits are examples of so-called efficiently computable sparse (ECS) operations, whose efficient classical simulation is described in Ref. [109].

For context, we note that this approach is similar in spirit to the Gottesman-Knill theorem [110], which states that any Clifford circuit with preparation and measurement in the computational basis can be simulated classically in polynomial time. Because QRAM circuits necessarily employ non-Clifford gates (controlled-SWAP), however, the theorem does not directly apply. Still, the similarities are apparent: restricting the allowed gates and state preparations enables an efficient classical description of the system, making efficient simulation possible.

In addition, for a wide variety of error models, noisy QRAM circuits can be simulated efficiently using Monte Carlo methods. To simulate noisy circuits, the space of error configurations is randomly sampled according to the distribution $p(c)$. For each sampled configuration $c$ from a set of samples $S$, we compute the final system state $|\Omega(c)\rangle$, and we obtain the fidelity by averaging $F = \frac{1}{|S|}\sum_{c\in S} F(c)$. This sampling procedure is efficient provided that two criteria are satisfied: first, that the state $|\Omega(c)\rangle$ is efficiently computable, and second, that sampling from $p(c)$ is efficient. A sufficient condition for satisfying these two criteria is that the error channel maps computational basis states to other computational states, i.e., the channel's Kraus operators $K_m$ satisfy

$$K_m |i\rangle \propto |i'\rangle \,, \tag{3.28}$$

for all $m$, and where $|i\rangle, |i\text{’}\rangle \in \{|0\rangle, |1\rangle, |W\rangle\}$ are computational basis states. The first criterion is satisfied because Eq. (3.28) guarantees that a QRAM circuit interspersed with applications of the Kraus operators $K_m$ is still ECS. The second criterion is satisfied because the distribution $p(c)$ can be sampled efficiently by applying errors independently to each router (with appropriate probability) at each time step as the simulation proceeds. In detail, suppose that at time $t$ the system is in a state $|\psi(t)\rangle$

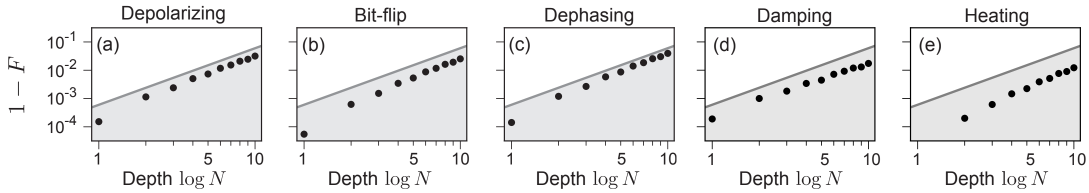


Figure 3.4: Favorable error scaling. For a variety of error channels, the query infidelity (black dots) is calculated numerically and plotted as a function of the tree depth $\log N$ (note the logarithmic scaling on both axes). The region defined by the upper bound (3.27) is shown in gray in each plot. Plotted infidelities are averages over many randomly generated binary data sets $\{x_0, \dots x_{N-1}\}$. Each such data set is generated by randomly choosing each $x_i$ to be 0 or 1 with equal probability. Error bars are smaller than the dot size. The error rate for all plots is $\varepsilon = 10^{-4}$.

that is a superposition of polynomially-many computational basis states,

$$|\psi(t)\rangle = \sum_{\{i_1,\dots i_{N-1}\}\in C} \alpha_i \left|i_1, i_2, \dots, i_{N-1}\right\rangle , \tag{3.29}$$

where $|i_r\rangle$ denotes the state of router $r$, and the cardinality of the set $C$ is $O(\text{poly}N)$. The probability that a Kraus operator $K_m$ is applied to router $r$ is

$$\text{Tr}\left[K_m^\dagger K_m \rho_r\right] , \tag{3.30}$$

where $\rho_r(t) = \text{Tr}_{\bar{r}}(|\psi(t)\rangle\langle\psi(t)|)$ is the reduced density matrix of router $r$, with $\text{Tr}_{\bar{r}}$ denoting the partial trace over the rest of the system. Eq. (3.28) guarantees that this probability is efficiently computable, so sampling from the possible errors at time $t$ is also efficient. This sampling procedure is repeated at each time step in order to sample from the full error configuration.

We apply this algorithm in order to compute the query infidelity for QRAM circuits with routers subject to a variety of noise channels. The results (Fig. 3.4) confirm that the QRAM query infidelity scales favorably in the presence of realistic noise channels acting on all of the memory's components. We stress that, for such channels, the expected number of errors generally scales linearly with $N$. Results for qutrit depolarizing, bit-flip, and dephasing channels are shown in panels (a), (b), and (c), respectively. We define the qutrit depolarizing, bit-flip, and dephasing channels as in Refs. [111], [16], and [112], respectively. In particular, we define the operators

$$A_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \; A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix}, \qquad (3.31)$$

where the matrices are written in the $\{|W\rangle, |0\rangle, |1\rangle\}$ basis, and $\omega = e^{i2\pi/3}$. The Kraus decompositions of the qutrit error channels are

$$\begin{aligned} \text{Depolarizing} = \Bigg\{ & \sqrt{1-\varepsilon} I, \sqrt{\frac{\varepsilon}{8}} A_1, \sqrt{\frac{\varepsilon}{8}} A_2, \sqrt{\frac{\varepsilon}{8}} A_1^2, \\ & \sqrt{\frac{\varepsilon}{8}} A_2^2, \sqrt{\frac{\varepsilon}{8}} A_1 A_2, \sqrt{\frac{\varepsilon}{8}} A_1^2 A_2, \sqrt{\frac{\varepsilon}{8}} A_1 A_2^2, \sqrt{\frac{\varepsilon}{8}} A_1^2 A_2^2 \Bigg\} \qquad (3.32) \\ \text{Bit-flip} = \Big\{ & \sqrt{1-\varepsilon} I, \sqrt{\varepsilon} \left( |0\rangle\langle 1| + |1\rangle\langle 0| \right) \Big\} \qquad (3.33) \\ \text{Dephasing} = \Bigg\{ & \sqrt{1-\varepsilon} I, \sqrt{\frac{\varepsilon}{2}} A_2, \sqrt{\frac{\varepsilon}{2}} A_2^2 \Bigg\} \qquad (3.34) \end{aligned}$$

Here, each channel is specified by a list of its Kraus operators $\{K_0, K_1, \ldots\}$. These channels are all of the form (3.9), so the query fidelity is subject to the bound (3.27). The numerical results are all clearly consistent with this bound, and the expected $1 - F \propto \log^2 N$ scaling is evident on the log-log scale. In panels (d) and (e), we show numerical results for qutrit decay and heating channels,

$$\begin{aligned} \text{Decay} &= \Big\{ |W\rangle\langle W| + \sqrt{1-\varepsilon} \left( |0\rangle\langle 0| + |1\rangle\langle 1| \right), \sqrt{\varepsilon} |W\rangle\langle 0|, \sqrt{\varepsilon} |W\rangle\langle 1| \Big\} \qquad (3.35) \\ \text{Heating} &= \Bigg\{ |0\rangle\langle 0| + |1\rangle\langle 1| + \sqrt{1-\varepsilon} |W\rangle\langle W|, \sqrt{\frac{\varepsilon}{2}} |0\rangle\langle W|, \sqrt{\frac{\varepsilon}{2}} |1\rangle\langle W| \Bigg\}. \qquad (3.36) \end{aligned}$$

We find that the query fidelities for these channels also satisfy the bound (3.27). Note, however, that the decay and heating channels are not mixed-unitary channels, so the query fidelities are subject to the general bound derived in Ref. [77], rather than Eq. (3.27).

## 3.3 Implications of QRAM's noise resilience

In this section, we describe a number of important implications of the bucket-brigade QRAM's resilience to noise. We show that the use of three-level routers in the bucket-brigade architecture is superfluous, that hybrid QRAM architectures (Section 2.2.4) can also be made partially resilient to noise, and that the bucket-brigade's noise resilience persists when quantum error correction is used.

### 3.3.1 Noise resilience without inactive routers

In Section 3.2.2, we proved that the query infidelity of the bucket-brigade QRAM scales favorably, even when inactive routers are subject to decoherence. It is thus natural to ask whether distinguishing between active and inactive routers is useful, and in fact whether the use of three-level routers is necessary in the first place. In this section, we show that the answer is no—the query fidelity still scales only polylogartihmically for QRAMs constructed from noisy two-level routers. As in Section 3.2.2, the argument presented to justify this claim is based on a careful analysis of how errors propagate. Furthermore, we show that this same argument also reveals that noise resilience persists when the QRAM is initialized in an arbitrary state, and when the routing circuit [Fig. 2.3(b)] is modified. Taken together, the results in this section show that the noise resilience of the bucket-brigade scheme is a robust property that is insensitive to implementation details. They also show that existing experimental proposals [17, 99] employing two-level routers are noise-resilient.

Consider a QRAM constructed from routers with only two states: $|0\rangle$ (route left) and $|1\rangle$ (route right). Routers are thus always active. For concreteness, we suppose

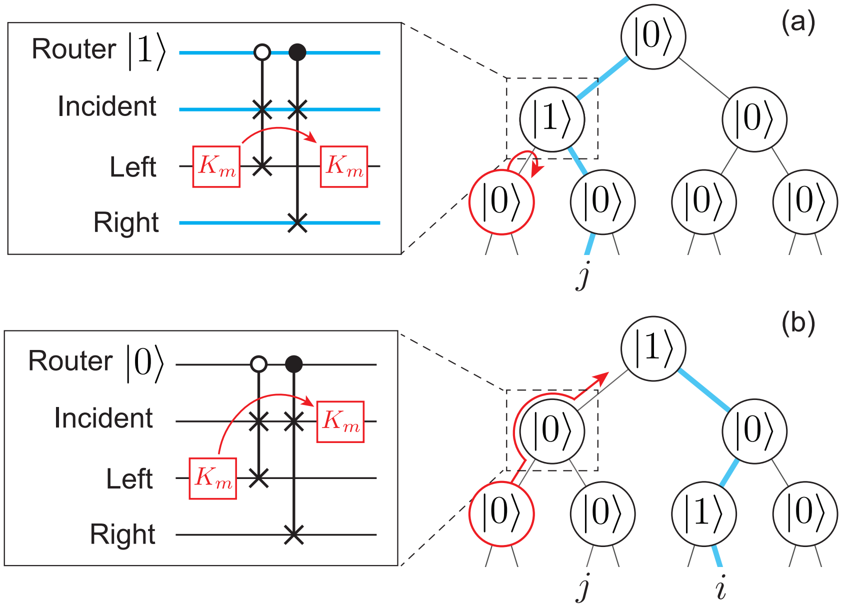


Figure 3.5: Error propagation with two-level routers. (a) A query to memory element $j \in g(c)$, with an error $K_{m>0}$ applied to the red-outlined router. The circuit on the left shows how the error propagates through the router indicated by the dashed box. In this case, the error does not propagate into branch $j$. (b) A query to a different memory element $i \in g(c)$. In this case, the error propagates upward into branch $j$, in contrast to the situation in (a).

that the routing operation is implemented using the circuit in Fig. 2.3(b), and that all routers are initialized in $|0\rangle$, though these assumptions can be relaxed. Unfortunately, the proof from Section 3.2.2 cannot be directly applied to show that the query fidelity also scales favorably in this case. The proof fails in the case of two-level routers because the propagation of errors is no longer so highly constrained. Recall that in the case of three-level routers, errors do not propagate from bad branches into good branches. More precisely, for any $i, j \in g(c)$, errors do not propagate into branch $j$ when branch $i$ is queried. This is not the case for two-level routers: while errors do not propagate into branch $i$ when branch $i$ is queried, they can propagate into other branches $j$, as illustrated in Fig. 3.5. Because of this difference, when multiple memory elements $i, j, \ldots \in g(c)$ are queried in superposition, it is not guaranteed that the address and bus registers will be disentangled from the routers at the end of the query. Thus, Eqs. (3.15) and (3.16) no longer hold. Instead, the final state $|\Omega(c)\rangle$ is given by

$$|\Omega(c)\rangle = \sum_{i\in g(c)} \alpha_i \left|i\right\rangle^A \left|x_i\right\rangle^B \left|f_i(c)\right\rangle^R + |\text{bad}(c)\rangle\,, \tag{3.37}$$

where $|f_i(c)\rangle$ denotes the now address-dependent final state of the routers, and $|f_i(c)\rangle \neq |f_j(c)\rangle$ in general. As a result, the $i, j \in g(c)$ terms are no longer guaranteed to be in coherent superposition after tracing out the routers. Rather, the final state of the address-bus system is liable to contain an incoherent mixture of these terms. That is, the final density matrix can contain terms of the form $|i, x_i\rangle\langle i, x_i|$ and $|j, x_j\rangle\langle j, x_j|$ without $|i, x_i\rangle\langle j, x_j|$ or $|j, x_j\rangle\langle i, x_i|$ terms. This loss of coherence reduces the fidelity.

We now proceed to estimate this reduction in fidelity. We find that the reduction is mild, such that the infidelity still scales only polylogarithmically with the memory size. Our approach is to isolate the subset of branches in $g(c)$ for which the sort of damaging error propagation described above does not occur. Explicitly, we define the subset $\tilde{g}(c) \subseteq g(c)$ as the largest subset such that for any $i, j \in \tilde{g}(c)$ errors do not propagate into branch $j$ during a query to element $i$. We then have that $|f_i(c)\rangle = |f_j(c)\rangle$ by the same argument as given in Section 3.2.2. It follows that, if multiple memory elements in $\tilde{g}(c)$ are queried in superposition, the address and bus registers will be disentangled from the routers at the end of the query.

Having defined $\tilde{g}(c)$ as the subset of good branches without damaging error propagation, we are free to define all other branches as bad and then proceed exactly as in Section 3.2.2. In particular, we analogously define

$$\tilde{\Lambda}(c) = \sum_{i\in\tilde{g}(c)} |\alpha_i|^2 \tag{3.38}$$

as the weighted fraction of good branches, and

$$F \geq [2\mathbb{E}(\tilde{\Lambda}) - 1]^2, \tag{3.39}$$

follows as the analog of Eq. (3.25). Because $\tilde{g}(c) \subseteq g(c)$, we have that

$$\mathbb{E}(\tilde{\Lambda}) = (1 - \delta)\mathbb{E}(\Lambda), \tag{3.40}$$

for some $\delta \in [0, 1]$ to be determined. Proceeding as in Section 3.2.2, it follows that the infidelity satisfies the bound

$$1 - F \leq 4\varepsilon T \log N + 4\delta \tag{3.41}$$

assuming $\varepsilon T \log N + \delta \leq 1/4$.

We can estimate $\delta$ by computing the average probability that errors propagate from bad branches into good branches. More specifically, we compute the probability that an error propagates into a branch $i \in g(c)$ when some other branch $j \in g(c)$ is queried. Suppose that a router $r$ suffers an error at time step $t$, and let $P_{r\to i}(t)$ denote the probability of this error propagating into branch $i$. Then to leading order in $\varepsilon$,

$$\delta = \varepsilon \sum_{r,t} P_{r\to i}(t) + O(\varepsilon^2), \tag{3.42}$$

which can be understood as the total probability that an error occurs and propagates into branch $i$. To compute $\sum_{r,t} P_{r\to i}(t)$ to leading order, we observe that errors are generally free to propagate from a router's left output to its input, as illustrated in Fig. 3.5(b). This is because, by default, all routers are initialized in $|0\rangle$, for which the routing operation swaps the states at the incident and left ports. In contrast, for an error to propagate upward from a router's right output, an additional error would be required to flip the router from $|0\rangle$ to $|1\rangle$. Thus, only the errors which can reach branch $i$ by propagating upward exclusively through the left outputs of routers contribute to $\sum_{r,t} P_{r\to i}(t)$ to leading order in $\varepsilon$. A conservative overestimate is thus obtained by first enumerating all routers $r$ that are connected to $i$ through the left ports of other routers, then pessimistically taking $P_{r\to i}(t) = 1$ for each. There are at most $\log^2 N$ such routers, so

$$\delta \leq \varepsilon T \log^2 N + O(\epsilon^2). \tag{3.43}$$

Substituting this expression into Eq. (3.41), we obtain

$$1 - F \lesssim 4\varepsilon T \left(\log N + \log^2 N\right). \tag{3.44}$$

Here we use the symbol $\lesssim$ to contrast this bound with Eq. (3.27); we proved the bound (3.27) rigorously, while we have obtained Eq. (3.44) through a scaling argument. As such, it is appropriate to focus only on the scaling of Eq. (3.44). We see that the infidelity still scales only polylogarithmically with the memory size, indicating that a bucket-brigade QRAM constructed from noisy two-level routers also exhibits noise resilience. Note, however, that the infidelity here scales with $\log^3 N$ [recall $T = O(\log N)$], as opposed to $\log^2 N$ in the case of three-level routers. Both scalings are still favorable according to our definition, but the discrepancy indicates that three-level routers impart better noise resilience than two-level routers.

We simulate noisy QRAM circuits with two-level routers in order to verify this noise resilience. Simulation results are shown in Fig. 3.6. For all noise channels simulated, the query infidelity is observed to scale polylogarithmically with the memory size, as expected. Moreover, the observed scaling exponents are $\leq 3$ in all cases, consistent with the pessimistic $1 - F \sim \log^3 N$ scaling given above.

It is interesting to note that two-level routers are more resilient to certain noise channels than others, as quantified by the observed differences in scaling exponents. For example, the infidelity under the dephasing channel is observed to scale approximately as $1 - F \sim \log^2 N$. This relatively mild scaling can be explained as follows. When the dephasing errors are propagated through the QRAM circuit, they may act

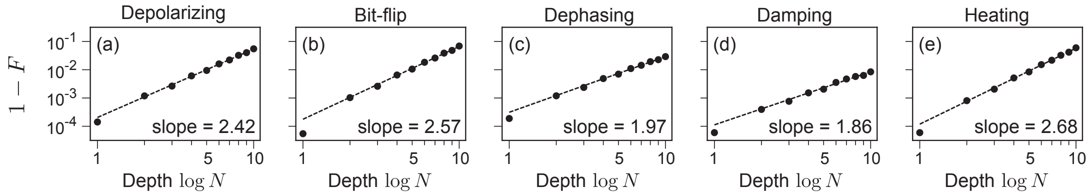


Figure 3.6: Favorable error scaling with two-level routers. For a variety of error channels, the query infidelity (black dots) is calculated numerically and plotted as a function of the tree depth $\log N$. Linear fits for each data set are shown as dashed lines, with the corresponding slopes given on each plot. Fits are performed only on data points with $\log N \geq 3$ so that the slopes are not skewed by finite-size effects at small $\log N$. Slopes $\leq 3$ are consistent with the scaling argument in the text. The error rate for all plots is $\varepsilon = 10^{-4}$.

non-trivially on the final state of the address and bus registers, but they act trivially on the final state of the routers (the all-$|0\rangle$ state). As a result, the final state of the routers is the same for every address: $|f_i(c)\rangle = |f_j(c)\rangle$ for all $i, j$. Hence, $\tilde{g}(c) = g(c)$, and the bound from Section 3.2.2 applies. For the other channels, $\tilde{g}(c) \neq g(c)$ in general, consistent with observed scaling exponents $> 2$. The case of amplitude damping is also interesting to consider: the expected number of errors for this channel is only $\varepsilon T \log N$ because only $\log N$ excitations are injected into the tree. Because $T = O(\log N)$, one expects the infidelity to scale with $\log^2 N$. The observed slope of 1.86 is somewhat smaller owing to the fact that, in our simulations, excitations are only susceptible to damping while they reside in the tree.

The scaling argument presented in this section also suffices to show that the noise resilience persists in two other interesting situations: when the QRAM is initialized in an arbitrary state, and when the routing circuit is modified. Regarding initialization, observe that the above argument is straightforwardly modified to cover the case where all routers are initialized in $|1\rangle$ rather than $|0\rangle$. Indeed, such an argument holds regardless of whether a given router is initialized in $|0\rangle$ or $|1\rangle$. It follows that the query infidelity scales favorably when the QRAM is initialized in an arbitrary state [3]

[3] This observation is distinct from the observation of Refs. [36, 84] that the ancillary qubits used to perform a query can be “dirty.” See Appendix A

(though some additional care must be taken when copying data to the bus—see Appendix A for details). This observation has great practical utility, as it means that QRAM can be constructed even from physical components that cannot reliably be initialized to a particular state.

Regarding modifications to the routing circuit, it is helpful to consider an example. In Ref. [99] a modified routing circuit was proposed in which one of the controlled-SWAP gates in Fig. 2.3(b) is replaced by a SWAP gate. This modification has nontrivial effects on how errors propagate. With the modified circuit, errors can propagate from bad branches into good branches even when three-level routers are used. However, this is the same sort of damaging error propagation as is illustrated in Fig. 3.5. Indeed, from the perspective of error propagation, the effect of this modification to the routing circuit is equivalent to replacing three-level routers with two-level routers. Accordingly, the argument above can be directly applied to show that the favorable scaling persists with the modified circuit. This example demonstrates that noise resilience is not a specific feature of the routing circuit in Fig. 2.3(b).

Taken together, the results from this section demonstrate that the noise resilience of the bucket brigade architecture is a robust property that is insensitive to implementation details. This observation affords a great deal of freedom to experimentalists in deciding how the routers and routing operations could be implemented in practice.

### 3.3.2 Noise resilience of hybrid architectures

In this section, we consider the effects of noise on hybrid QRAM architectures (Section 2.2.4). Recall that these architectures are described by a tunable parameter, $M$, that dictates the circuit width $O(\log N + N/M)$ and depth $O(M \log N)$. At the extremes, the large-width and short-depth bucket-brigade QRAM circuits (Fig. 2.8) correspond to $M = 1$, while the short-width and large-depth QROM circuits (Fig. 2.9) correspond to $M = N$. Circuits with intermediate values of $M$ are referred to as hybrid.

We first consider the effects of noise on QROM, then turn to the hybrid circuits. One can easily observe that QROM does not possess any intrinsic noise resilience. For example, when all memory elements are queried in equal superposition, a single dephasing error at any location in the QROM circuit reduces the query fidelity to 0. The effects of bit flips are similarly detrimental, assuming there is no contrived redundancy in the classical data. More generally, we can follow the approach of Section 3.2.2 and express the QROM query fidelity as $F = \sum_c p(c)F(c)$, where the error configuration $c$ specifies which Kraus operators are applied at each location in the circuit, and $F(c)$ is the final state fidelity of the address and bus registers given configuration $c$. In the case of QROM, only the error configuration with no errors is guaranteed to have unit or near-unit fidelity in general[4]. There are $O(N \log^2 N)$ possible error locations, so it follows that the QROM query infidelity scales as

$$1 - F_{\text{QROM}} \sim \varepsilon N \log^2 N, \tag{3.45}$$

to leading order. Therefore, QROM is not noise resilient, since near-unit query fidelities generally require $\varepsilon \ll 1/N$, neglecting logarithmic factors.

Similarly, the hybrid circuits do not exhibit noise resilience when the QRAM subroutines are implemented with the fanout architecture. Recall from Section 3.1 that the fanout architecture is not noise resilient; only the fanout's no-error configuration is guaranteed to have high fidelity in general. Because neither QROM nor the fanout QRAM are noise resilient, only the no-error configuration of the hybrid fanout circuit is guaranteed to have high fidelity. Since the number of possible error locations is

[4] Some other error configurations may have high fidelity for specific choices of the error channel, the initial address state, or the classical data, but we ignore this possibility to keep the analysis general and pessimistic.

$O(M \log N(\log N + N/M))$, the query fidelity scales as

$$1 - F_{\text{hybrid,fanout}} \sim \varepsilon(N \log N + M \log^2 N), \tag{3.46}$$

to leading order. Here again, error rates $\varepsilon \ll 1/N$ are required for near-unit query fidelity, neglecting logarithmic factors.

In contrast, the hybrid circuits do exhibit partial noise resilience when the QRAM subroutines are implemented with the bucket-brigade architecture. Because the bucket-brigade QRAM is resilient to noise, error configurations with errors occurring exclusively in the QRAM subroutines can still have high fidelities. We can obtain a lower bound on the query fidelity by neglecting all other configurations. Doing so allows us to bound the query fidelity by a product of two factors

$$F_{\text{hybrid,BB}} \gtrsim (1-\varepsilon)^{O(M \log^2 N)} \times (1-\varepsilon)^{O(M \log N \log N/M)} \tag{3.47}$$

The first factor is simply the probability that no errors occur outside the QRAM. The second factor is the expected fraction of error-free branches within the QRAM (each branch contains $\log N/M$ routers, and there are $T = O(M \log N)$ possible time steps at which errors may occur). We have related this expected fraction to $F_{\text{hybrid}-\text{BB}}$ by the same argument as in Section 3.2.2. Thus, to leading order,

$$\begin{aligned} 1 - F_{\text{hybrid,BB}} &\lesssim \varepsilon M \log N(\log N + \log N/M), && (3.48) \\ &\sim \varepsilon M \log^2 N. && (3.49) \end{aligned}$$

Note that we have not kept track of prefactors since we are only interested in how the infidelity scales; a strict upper bound could be rigorously derived following the approach of Section 3.2.2. Near-unit query fidelities only require error rates $\varepsilon \ll 1/M$, neglecting logarithmic factors (cf. the $\varepsilon \ll 1/N$ requirement for the other cases).

Because $M \leq N$, the infidelity of the hybrid bucket-brigade architecture scales more favorably than both QROM and the hybrid fanout architecture. Of course, the extent of the scaling advantage depends on $M$. For example, if one chooses $M = \sqrt{N}$, so that the number of qubits and circuit depth are comparable, then the hybrid bucket-brigade architecture yields a quadratic improvement in the infidelity scaling. Note that we assume three-level routers above for simplicity; for two-level routers, one should replace $\log N/M \to \log^2 N/M$ in the above expressions, in accordance with the argument from Section 3.3.1.

### 3.3.3 Resilience to logical errors in error-corrected QRAM

In this section, we show that the benefits of the bucket brigade scheme persist when quantum error correction is used. When the bucket-brigade QRAM is implemented using error-corrected routers and fault-tolerant routing operations [21, 113], the logical query infidelity scales only polylogarithmically with the memory size. Thus, error-corrected implementations of the bucket-brigade scheme can offer improved fidelity or reduced overhead relative to other implementations. In practice, these improvements may be tempered by the overhead associated with the fault-tolerant implementation of the routing operations, and we discuss the utility of the bucket-brigade architecture in light of such considerations.

While we have shown that the query infidelity of the bucket-brigade scheme scales favorably with the memory size, strategies to further suppress the infidelity are desirable, and quantum error correction provides one possible approach. Indeed, error correction may be required in cases where the physical error rate cannot be made sufficiently small, or when many queries must be performed in sequence. For example, Ref. [16] argued that error correction is likely to be needed for any algorithm that requires a number of QRAM queries that scales superpolynomially in $\log N$, e.g., Grover's algorithm [28].

It is thus natural to ask whether an error-corrected bucket brigade QRAM offers any advantages over other architectures. Indeed, this question was previously considered in Ref. [16], where the authors argue in the negative. Their argument is based on the canonical attribution [14–17, 52, 53] of the bucket brigade's noise resilience to the limited number of active routers. Error-corrected routers must be considered active, they argue, and so the number of active routers is the same in both the fanout and bucket brigade schemes. Hence, the bucket brigade scheme was not believed to provide any advantage if error correction were used.

As we have shown, however, the noise resilience of the bucket brigade scheme is not a function of the number of active routers, but rather a function of the limited entanglement among the routers. As a direct corollary of this result, we find that, in fact, the benefits of the bucket brigade scheme do persist when error correction is used. The proof from Section 3.2.2 is agnostic to whether the routers are composed of uncorrected physical qubits or error-corrected logical qubits, provided that uncorrectable logical errors occur independently with some probability $\varepsilon_L$ (which can be guaranteed by implementing the routing operations fault tolerantly). Physical errors occurring with probability $\varepsilon$ can simply be replaced by logical errors occurring with probability $\varepsilon_L$, and one obtains the corresponding bound

$$1 - F_L \leq 4\varepsilon_L T_L \log N, \tag{3.50}$$

where $F_L$ is the query fidelity of the logical QRAM circuit, and $T_L$ is the circuit depth. Thus, when implemented fault-tolerantly, the logical bucket-brigade circuits possess an intrinsic resilience to logical errors, in that the logical infidelity scales only polylogarithmically with the the size of the memory (This scaling assumes $T_L = O(\log N)$; see further discussion at the end of this section).

To provide further exposition, we give a concrete example of an error-corrected quantum router. Consider a quantum error correcting-code, with logical codewords $|0_L\rangle$ and $|1_L\rangle$ satisfying the Knill-Laflamme conditions [21, 114],

$$PK_i^\dagger K_j P = h_{ij} P, \tag{3.51}$$

where $P$ is the projector onto the code space, the $\{K_i\}$ are the set of correctable errors, and $h$ is a Hermitian matrix. A logical two-level quantum router then constitutes a single logical qubit (similarly, a logical three-level quantum router can be constructed from a pair of logical qubits, for example). Crucially, the logical routers comprising the QRAM can be corrected without revealing any information about which memory elements are being accessed. This is because the conditions (3.51) guarantee that errors can be corrected without revealing any information about the encoded state. Even when the logical router is in a superposition of different states, or entangled with other routers, syndrome measurements do not reveal information about the router state. Note that the conditions (3.51) also guarantee that information is not leaked to the environment; the states $|0_L\rangle$ and $|1_L\rangle$ necessarily have equal probability of suffering errors.

Because of the favorable logical error scaling Eq. (3.50), error-corrected implementations of the bucket-brigade scheme can offer improved fidelity or reduced overhead relative to other implementations. For instance, if the same error-correcting code is used in fault-tolerant implementations of the bucket-brigade and fanout QRAMs, the logical infidelity of bucket brigade QRAM will be lower than the logical infidelity of the fanout QRAM by a factor of $\sim 1/N$ in general. Alternatively, if a given application requires that QRAM have a logical infidelity below some threshold, the error-correction overhead required to realize such high-fidelity queries can be significantly smaller for the bucket-brigade scheme relative to the fanout scheme. Indeed, even if the reduction in error-correction overhead is fairly small for each router, the total overhead reduction considering all $N$ routers can be significant. Such reductions could be of significant practical benefit. For context, we note that detailed overhead estimates for fault-tolerant QRAM using the surface code were made in Ref. [18]; these overheads can potentially be improved by exploiting the bucket-brigade's noise resilience.

## 3.4 Conclusions and Outlook

We have shown that the bucket-brigade QRAM architecture possesses a remarkable resilience to noise. Even when all $O(N)$ components comprising the QRAM are subject to arbitrary error channels, the query infidelity scales only polylogarithmically with the memory size. As a result, the bucket-brigade architecture can be used to perform high-fidelity queries of large memories without the need for quantum error correction, provided physical error rates are low. Importantly, we prove that this noise resilience holds for *arbitrary* error channels, demonstrating that a noise-resilient QRAM can be implemented with realistically noisy devices.

In the near-term, this noise resilience could facilitate experimental demonstrations and benchmarking of numerous quantum algorithms. We are presently in the Noisy, Intermediate-Scale Quantum (NISQ) era [115], when making more qubits is easier than making better qubits. The same is likely to be true even in the era of early fault-tolerance. In these eras, the bucket-brigade architecture—with its larger overhead and noise resilience—could actually prove to be more practical than alternatives like QROM (see Section 3.3.2) that have a lower overhead but are less tolerant to noise. The bucket-brigade architecture thus more readily enables small-scale, near-term implementations of algorithms, and important practical insights are likely to be gained from such demonstrations. Schemes to further suppress the query fidelity without resorting to full error correction (chapter 4) could prove useful in this effort.

| $N$ | $\exp(n)$ | $\exp(n)$ | $\text{poly}(n)$ |
|---|---|---|---|
| Q | $\exp(n)$ | $\text{poly}(n)$ | poly(n) |
| Applicable architectures | QRAM | QRAM | QRAM, QROM, Hybrid |
| QEC required? | Yes | Maybe not | Maybe not |
| Paradigmatic example | Searching an unstructured database [28] | Solving linear systems of equations [51] | Simulating local Hamiltonians [116] |

Table 3.2: Algorithm categorization. Algorithms are sorted based on how the size of the classical memory, $N$, and the number of queries, $Q$, scale with the number of qubits, $n$. When $N = \exp(n)$, QRAM is the only suitable architecture, assuming $\text{poly}(n)$ query times are required. When $Q = \text{poly}(n)$ quantum error correction may not be required, depending on the physical error rates. For the examples in the last two rows, $Q$ also depends on the particular algorithm used and the desired precision; we assume these are chosen such that $Q = \text{poly(n)}$. We omit the case of $N = \text{poly}(n)$ and $Q = \exp(n)$, for which the query complexity is exponential in the problem size.

In the long-term, this noise resilience may prove useful in facilitating speedups for certain quantum algorithms, but it is important that the required resources be carefully assessed before a speedup via QRAM is claimed. Consider an oracle-based algorithm that requires $n$ qubits (not including ancillary qubits needed to implement the oracle). As we show in Table 3.2, such algorithms can be conveniently classified according to how the size of the classical memory being queried, $N$, and the total number of queries, $Q$, scale with $n$. Assuming $\text{poly}(n)$ query times are required, the memory size $N$ dictates whether QRAM (as opposed to QROM or a hybrid architecture) is required to implement the oracle. The number of queries $Q$ dictates whether error correction is necessarily required [16]. The noise resilience of the bucket-brigade has the biggest potential impact in case of $N = \exp(n)$ and $Q = \text{poly}(n)$. In this case, QRAM is required, and the noise-resilience of the bucket-brigade architecture, together with the comparatively small number of queries, allows for the possibility that the QRAM could be implemented without error correction. Of course, the noise resilience can also be advantageous in the other cases, where hybrid architectures may be employed (Section 3.3.2) or when error correction is used (Section 3.3.3).

Finally, it is worth emphasizing that the results in this chapter constitute general statements about the bucket-brigade architecture, independent of its application to particular algorithms. In fact, the architecture may prove useful in applications other than facilitating algorithmic speedups. For example, Ref. [75] employs the bucket-brigade architecture in a quantum cryptographic protocol. The architecture may similarly prove useful for quantum communication or metrology. Exploring applications of the bucket-brigade architecture—and the utility of its noise resilience—in these other contexts represents an interesting direction for future research. In particular, applications involving quantum queries of quantum data remain largely unexplored. Our own preliminary work indicates that the bucket-brigade architecture may also be useful for quantum communication, quantum compression, or efficient, distributed quantum information processing, for example.

# Chapter 4

# Hardware-efficient error suppression

In Chapter 3, we showed that the bucket-brigade QRAM is remarkably resilient to noise, with a query infidelity that scales only polylogarithmically with the memory size $N$. This favorable infidelity scaling is very encouraging for noisy implementations of QRAM. However, the query infidelity is ultimately still lowerbounded by the physical error rate. This lower bound on the query infidelity can limit the potential applications of QRAM. Of course, in some applications, only a small number of QRAM queries may be required, and it is conceivable that this residual infidelity may not be problematic [16]. However, in applications requiring many queries, some form of error correction or suppression will likely be required to further reduce the query infidelity.

In this chapter, we present a hardware-efficient error suppression scheme. In contrast to quantum error correction, which necessarily entails an additional $O(N)$ hardware overhead when applied to QRAM, the minimal additional hardware overhead required by our scheme is independent of the size of the QRAM itself. The price we pay for this improved hardware efficiency is that the extent of the error suppression is somewhat limited. For a base query infidelity of $p$, our scheme enables queries with an effective query infidelity of

$$1 - F_M = p/M + O(p^2), \tag{4.1}$$

where $M$ is a tunable parameter that dictates the time overhead associated with the error-suppression scheme. Thus, our scheme can provide at most a quadratic reduction of the query infidelity (for $M = 1/p$). However, even a quadratic reduction in error would have tremendous practical utility in near-term applications. Indeed, our scheme is particularly well-suited for use in near-term devices, owing to its hardware efficiency.

In Section 4.1, we motivate our scheme by describing the challenges that conventional error correction approaches face when applied to QRAM, and we introduce the basic ingredient of our suppression scheme: error symmetrization. Next, in Section 4.2, we present our scheme, and analyze its error suppression capabilities when applied to general noisy operations (not just QRAM). Finally, in Section 4.3, we apply our general analysis to the particular case of QRAM, demonstrating that the query infidelity can be suppressed in a hardware-efficient manner.

## 4.1 Motivation and background

In this section, we provide a detailed discussion of the practical challenges associated with the implementation of an error-corrected QRAM by conventional methods. These challenges serve as a motivation for our own error-suppression scheme, which is based on a fundamentally different approach. We also review the basic idea of error symemtrization, as well as some of the shortcomings in the original error symmetrization proposal of Ref. [117] (our scheme remedies these shortcomings).

### 4.1.1 Practical challenges with error-corrected QRAM

Quantum error correction can be used to suppress the QRAM query infidelity. One simply replaces each of the physical qubits comprising quantum routers with error-corrected logical qubits. Further, to prevent the uncontrolled spread of errors, the associated routing operations must be implemented in a fault-tolerant manner. There is no fundamental obstacle that would prevent one from applying these techniques in the context of QRAM. There are, however, a variety of practical concerns that could make conventional approaches to error correction infeasible.

The first practical concern is the large error-correction overhead. Building a large-scale QRAM requires a large hardware overhead, even without error correction. Without error correction, QRAM requires $O(N)$ physical qubits to serve as quantum routers in order to query a classical memory of size $N$. In big data or machine learning applications, the relevant values of $N$ could easily reach millions or billions, and comparable numbers of physical qubits would be required to apply QRAM-based quantum algorithms to such problems. Scaling to this many physical qubits is already a daunting engineering challenge.

This challenge is only magnified when error correction is used, as now $O(N)$ *logical* qubits are required. Though error correction is formally efficient [20], in the sense that exponential error suppression can be achieved with only a polynomial overhead, the overheads involved can still be quite large. For example, the most common architecture for fault-tolerant quantum computing is the surface code [118], where current estimates suggest that an overhead of $\sim 1000$ physical qubits per logical qubit is likely required to enable practical applications [119]. Thus, when applied in the context of large-scale QRAM, surface code error correction could increase the required number of physical qubits from millions or billions to billions or trillions. This crude estimate is consistent with the more detailed analysis of Ref. [18], which found that a fault-tolerant QRAM implementation using the surface code would require $10^{10}$ physical qubits for $N = 10^6$, and $10^{13}$ physical qubits for $N = 10^9$. Whether quantum computers will one day be scalable to such sizes remains an open question, but these large overheads indicate that building a large, fault-tolerant QRAM will not be feasible in the foreseeable future (at least, not with conventional surface code architectures—in Chapter 5 we describe an alternative error-correction approach based on cat qubits that is far more hardware efficient).

Another practical concern is the fact that quantum routing (Fig. 2.3) is a non-Clifford operation. As a result, magic state distillation [86, 87] is required to implement the routing fault-tolerantly in the usual Clifford+T fault-tolerance model. In total, $O(N)$ magic states are required to perform a query. If queries are to be performed in time $T_L = O(\log N)$, these magic states must be distilled in parallel, so $O(N)$ magic state factories are required. The additional overhead associated with these factories could be prohibitive for large $N$, however, potentially limiting the extent to which such parallelism can be exploited. That said, it should be noted that though the routing operation is non-Clifford, it is also not universal for quantum computing. An important open question concerning fault-tolerant QRAM is thus whether fault-tolerant implementations of this specific operation can be designed that are more efficient than generic fault-tolerant operations. Schemes for pieceable fault-tolerance [120], flag qubits [121, 122], or noise-bias preserving gates [123–125] may prove useful in this regard.

### 4.1.2 Error symmetrization

We now describe the error-suppression scheme of Ref. [117], which serves as the motivation for our own error-suppression scheme.

### The symmetric subspace

We begin by defining the symmetric subspace, $\mathcal{S}$, and examining a few of its relevant properties. We refer the interested reader to Ref. [126] for further details.

Consider a collection of $M$ quantum systems, each described by a $d$-dimensional Hilbert space $\mathcal{H}$. We define the symmetric subspace, $\mathcal{S}$, as the subspace of the joint Hilbert space $\mathcal{H}^{\otimes M}$ that is invariant under permutations of the $M$ systems,

$$\mathcal{S} = \{|\psi\rangle \in \mathcal{H}^{\otimes M} \, : P(\pi)\,|\psi\rangle = |\psi\rangle \;\forall\, \pi \in \mathcal{S}_M\}. \tag{4.2}$$

Here, $\mathcal{S}_M$ is the symmetric group over $M$ symbols, $\pi$ is an element of this group (i.e. a permutation), and $P(\pi)$ is the corresponding permutation operator on the space $\mathcal{H}^{\otimes M}$,

$$P(\pi) = \sum_{i_1,\ldots,i_M=0}^{d-1} |i_{\pi^{-1}(1)},\ldots,i_{\pi^{-1}(M)}\rangle \langle i_1,\ldots,i_M| \,. \tag{4.3}$$

As examples, for the case of $M = 3$ and $d = 2$, the states $|s_0\rangle = |000\rangle$, $|s_3\rangle = |111\rangle$, and

$$|s_1\rangle = \frac{1}{\sqrt{3}}\left(|100\rangle + |010\rangle + |001\rangle\right)$$

are all contained in $\mathcal{S}$. In fact, the states

$$|s_t\rangle \equiv \sqrt{\frac{t!(M-t)!}{M!}} \sum_{\vec{i}\in[t]} |i_1,\ldots,i_M\rangle \,, \tag{4.4}$$

form a basis for $\mathcal{S}$ in the case of $d = 2$ (the basis states can be expressed in a similar form for the case of general $d$ [126]). Here, $[t]$ denotes the set of all bit strings with exactly $t$ 1's and $(M-t)$ 0's.

$\mathcal{S}$ can be equivalently defined as the smallest subspace of $\mathcal{H}^{\otimes M}$ that contains all states of the form $|\psi\rangle\,|\psi\rangle \dots |\psi\rangle$, for arbitrary $|\psi\rangle$. That is,

$$\mathcal{S}' = \text{span}\{|\psi\rangle^{\otimes M} \,:\, |\psi\rangle \in \mathcal{H}\}. \tag{4.5}$$

One can prove that $\mathcal{S}' = \mathcal{S}$, and we sketch the basic idea of the proof. (For simplicity, we consider the case of $d = 2$; the proof for general $d$ is similar [126]). First observe that $P(\pi)\,|\psi\rangle^{\otimes M} = |\psi\rangle^{\otimes M}$ for arbitrary $\pi$, which shows that $\mathcal{S}' \subseteq \mathcal{S}$. To show containment in the other direction, we consider a generic product state,

$$|p(x_0, x_1)\rangle = \left(\sum_{i=0}^{1} x_i\,|i\rangle\right)^{\otimes M}, \tag{4.6}$$

where clearly $|p(x_0, x_1)\rangle \in \mathcal{S}'$ for all $x_0, x_1$. Expanding out this expression, the coefficient of $x_0^{t_0} x_1^{t_1}$ is

$$\sqrt{\binom{M}{t_1}}\,|s_{t_1}\rangle\,.$$

Now, the main observation of the proof is that $|p(x_0, x_1)\rangle \in \mathcal{S}'$ for all $x_0, x_1$ implies that $|s_{t_1}\rangle \in \mathcal{S}'$. To see this fact, notice that

$$\frac{\partial^{t_0}}{\partial x_0^{t_0}} \frac{\partial^{t_1}}{\partial x_1^{t_1}}\,|p(x_0, x_1)\rangle \in \mathcal{S}' \tag{4.7}$$

because $\partial\,|p(x_0, x_1)\rangle / \partial x_i$ can be expressed as linear combinations of $|p(x_0, x_1)\rangle$ for different values of $x_0, x_1$. At the same time,

$$|s_{t_1}\rangle = \frac{\partial^{t_0}}{\partial x_0^{t_0}} \frac{\partial^{t_1}}{\partial x_1^{t_1}}\,|p(x_0, x_1)\rangle\Big|_{x_0, x_1 = 0}, \tag{4.8}$$

so $|s_{t_1}\rangle \in \mathcal{S}'$. Because the $|s_{t_1}\rangle$ form a basis for $\mathcal{S}$, we have that $\mathcal{S} \subseteq \mathcal{S}'$, completing the proof.

Another useful property of $\mathcal{S}$ is that the operator,

$$\Pi_{\mathcal{S}} = \frac{1}{M!} \sum_{\pi \in \mathcal{S}_M} P(\pi) \tag{4.9}$$

is the orthogonal projector onto $\mathcal{S}$. This fact can be proven as follows. For any $\pi \in \mathcal{S}_m$, we have

$$\begin{aligned} P(\pi)\Pi_{\mathcal{S}} &= \frac{1}{M!} \sum_{\pi' \in \mathcal{S}_M} P(\pi)P(\pi') \\ &= \frac{1}{M!} \sum_{\pi' \in \mathcal{S}_M} P(\pi\pi') \\ &= \frac{1}{M!} \sum_{(\pi^{-1}\pi') \in \mathcal{S}_M} P(\pi') \\ &= \Pi_{\mathcal{S}}, \end{aligned} \tag{4.10}$$

and similarly $\Pi_{\mathcal{S}} P(\pi) = \Pi_{\mathcal{S}}$. It follows that $\Pi_{\mathcal{S}}^\dagger \Pi_{\mathcal{S}} = \Pi_{\mathcal{S}}$, so $\Pi_{\mathcal{S}}$ is an orthogonal projector. Now, Eq. (4.10) further implies that

$$P(\pi)\Pi_{\mathcal{S}} \left|\psi\right\rangle = \Pi_{\mathcal{S}} \left|\psi\right\rangle, \tag{4.11}$$

for arbitrary $\left|\psi\right\rangle \in \mathcal{H}^{\otimes M}$. Thus, the image of $\Pi_{\mathcal{S}}$ is contained in $\mathcal{S}$. To show containment in the other direction, we observe that

$$\Pi_{\mathcal{S}} \left|\psi\right\rangle = \frac{1}{M!} \sum_{\pi \in \mathcal{S}_M} P(\pi) \left|\psi\right\rangle = \left|\psi\right\rangle, \tag{4.12}$$

for any $\left|\psi\right\rangle \in \mathcal{S}$. Thus, the image of $\Pi_{\mathcal{S}}$ is $\mathcal{S}$, which completes the proof.

### Error suppression via $\mathcal{S}$ projection

Having defined $\mathcal{S}$, we may now describe the error-suppression scheme of Ref. [117], which is illustrated schematically in Fig. 4.1. Suppose that we have a collection of

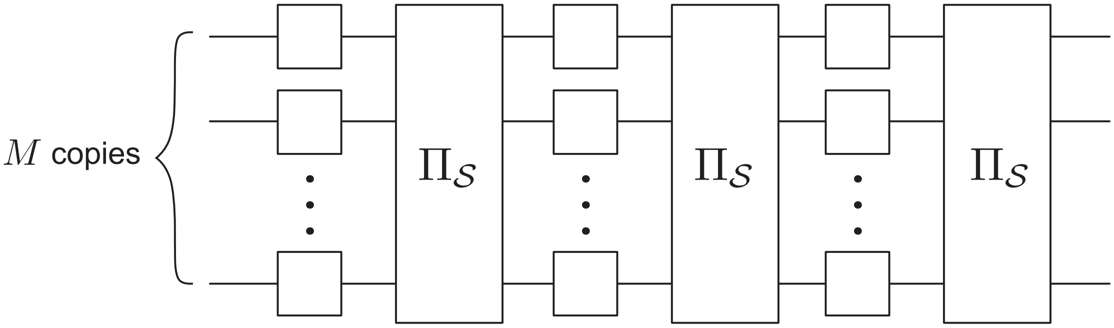


Figure 4.1: Error-suppression scheme of Ref. [117]. A collection of $M$ quantum computers perform a the same computation in parallel (indicated by the blank boxes). The parallel computations are interspersed by repeated projections onto $\mathcal{S}$.

$M$ quantum computers, all performing the same computation in parallel. Errors in these computations can be suppressed by frequently and repeatedly projecting the joint system into $\mathcal{S}$. Conceptually, the idea is that under error-free operation, the states of all the quantum computers will be identical throughout the computation, so projecting the joint system onto $\mathcal{S}$ will have no effect. However, an error in one of the quantum computers can give rise to a component of the joint state which lies outside $\mathcal{S}$. Successful projection onto $\mathcal{S}$ can eliminate this component, and ideally this brings the joint system back to its error-free state.

Before analyzing the efficacy of this scheme, let us explain how a projection onto $\mathcal{S}$ can be realized (see Fig. 4.2). One begins by preparing a register, $A$, of $\log(M!) = O(M)$ ancillary qubits in $|0\rangle^{\otimes M}$. Then a unitary operation $U$ is applied to this register to prepare it in the equal superposition state,

$$U\,|0\rangle^{A} = \frac{1}{\sqrt{M!}} \sum_{i=0}^{M!-1} |i\rangle^{A} \,. \tag{4.13}$$

In the case where $M!$ is a power of 2, the operation $U$ can be implemented by a single layer of Hadamard gates, and otherwise it can be implemented using the quantum Fourier transform. Next, a controlled-permutation operation is applied to the system,

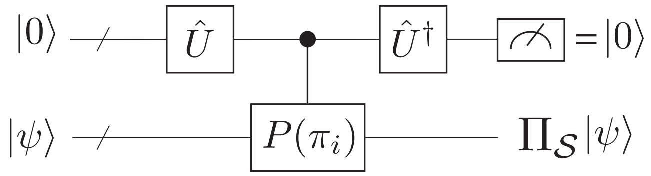


Figure 4.2: Quantum circuit for realizing a projection on the subspace $\mathcal{S}$.

$S$, of $M$ quantum computers

$$\frac{1}{\sqrt{M!}}\sum_{i=0}^{M!-1}|i\rangle^A|\psi\rangle^S \to \frac{1}{\sqrt{M!}}\sum_{i=0}^{M!-1}|i\rangle^A P(\pi_i)|\psi\rangle^S. \tag{4.14}$$

We note that Ref. [117] provides an explicit circuit for realizing this controlled-permutation operation. Then, $U^\dagger$ is applied to $A$,

$$\begin{aligned}(U^\dagger\otimes I)\frac{1}{\sqrt{M!}}\sum_{i=0}^{M!-1}|i\rangle^A P(\pi_i)|\psi\rangle^S &= |0\rangle^A\left(\frac{1}{M!}\sum_{i=0}^{M!-1}P(\pi_i)\right)|\psi\rangle^S+\dots\\ &= |0\rangle^A\Pi_{\mathcal{S}}|\psi\rangle^S+\dots,\end{aligned} \tag{4.15}$$

where "…" denotes terms orthogonal to $|0\rangle^A$. Postselecting on $|0\rangle^A$ and discarding the $A$ register thus yields

$$\Pi_{\mathcal{S}}|\psi\rangle^S, \tag{4.16}$$

as desired. This procedure can be understood simply as a generalized Hadamard test that projects the system onto the image of $\Pi_{\mathcal{S}}$ when passed.

Now let us quantify the error suppression associated with the successful projection into $\mathcal{S}$. Let $|\psi\rangle$ denote the ideal state of each of the $M$ quantum computers. For simplicity, we consider an error model where the computers are independently subject to a channel $\mathcal{E}$ that maps $|\psi\rangle$ to some orthogonal state $|\psi^\perp\rangle$ with probability $p$,

$$|\psi\rangle\to\mathcal{E}(|\psi\rangle\langle\psi|)=(1-p)|\psi\rangle\langle\psi|+p|\psi^\perp\rangle\langle\psi^\perp|. \tag{4.17}$$

Under this model, the infidelity of any one of the $M$ quantum computers is $p$, and the goal of the error suppression procedure is to reduce this infidelity by projecting the joint state onto $\mathcal{S}$. After the error channel is applied to all quantum computers, the joint state of the system is

$$\begin{aligned}\mathcal{E}(|\psi\rangle\langle\psi|)^{\otimes M} &= (1-p)^M(|\psi\rangle\langle\psi|)^{\otimes M} \\ &\quad + p(1-p)^{M-1}\sum_{i=1}^{M} |\psi^\perp(i)\rangle\langle\psi^\perp(i)| + O(p^2), \end{aligned} \tag{4.18}$$

where

$$|\psi^\perp(i)\rangle \equiv |\psi\rangle_{S_1}|\psi\rangle_{S_2}\dots|\psi^\perp\rangle_{S_i}\dots|\psi\rangle_{S_M} \tag{4.19}$$

denotes the state in which the $i$-th subsystem, $S_i$, has suffered an error, but all other subsystems are error free. Projecting onto $\mathcal{S}$ yields the (unnormalized) state,

$$\begin{aligned}\rho &= \Pi_{\mathcal{S}}\left(\mathcal{E}(|\psi\rangle\langle\psi|)^{\otimes M}\right)\Pi_{\mathcal{S}} \\ &= (1-p)^M(|\psi\rangle\langle\psi|)^{\otimes M} + p(1-p)^{M-1}\sum_{i=1}^{M}\Pi_{\mathcal{S}}|\psi^\perp(i)\rangle\langle\psi^\perp(i)|\Pi_{\mathcal{S}} + O(p^2) \\ &= (1-p)^M(|\psi\rangle\langle\psi|)^{\otimes M} + \frac{1}{M}p(1-p)^{M-1}\sum_{i=1}^{M}|s_1^{(\psi)}\rangle\langle s_1^{(\psi)}| + O(p^2), \\ &= (1-p)^M(|\psi\rangle\langle\psi|)^{\otimes M} + p(1-p)^{M-1}|s_1^{(\psi)}\rangle\langle s_1^{(\psi)}| + O(p^2) \end{aligned} \tag{4.20}$$

where

$$|s_1^{(\psi)}\rangle \equiv \frac{1}{\sqrt{M}}\sum_{i=1}^{M}|\psi^\perp(i)\rangle \in \mathcal{S} \tag{4.21}$$

is a symmetric superposition of single-error states $|\psi^\perp(i)\rangle$.

We can now use Eq. (4.20) to compute the probability of successful postselection and the fidelity of the postselected state to leading order. The probability of successful postselection is

$$\begin{aligned}\mathrm{Tr}[\rho] &= (1-p)^M + p(1-p)^{M-1} + O(p^2) \\ &= 1-(M-1)p + O(p^2). \end{aligned} \tag{4.22}$$

Note that this probability decreases with $M$. To compute fidelity of the postselected state, we first trace out all but one of the subsystems. Which subsystem we choose to retain is inconsequential; the joint state is symmetric, so the reduced states of all subsystems are the same. For simplicity, we trace out all subsystems except for the first,

$$\begin{aligned}\mathrm{Tr}_{S_2,\ldots,S_M}[\rho] &= (1-p)^M \left|\psi\right\rangle \left\langle\psi\right| + p(1-p)^M \mathrm{Tr}_{S_2,\ldots,S_M}\left[\left|s_1^{(\psi)}\right\rangle \left\langle s_1^{(\psi)}\right|\right] + O(p^2) \\ &= (1-p)^M \left|\psi\right\rangle \left\langle\psi\right| + p(1-p)^M \left[\frac{1}{M}\left|\psi^\perp\right\rangle \left\langle\psi^\perp\right| + \frac{M-1}{M}\left|\psi\right\rangle \left\langle\psi\right|\right] + O(p^2) \\ &= \left(1 - Mp + p\frac{M-1}{M}\right)\left|\psi\right\rangle \left\langle\psi\right| + \frac{p}{M}\left|\psi^\perp\right\rangle \left\langle\psi^\perp\right| + O(p^2) \\ &\equiv \rho_{S_1} \end{aligned} \tag{4.23}$$

Now, the infidelity can be expressed as

$$\begin{aligned}1-F &= 1 - \frac{1}{\mathrm{Tr}[\rho_{S_1}]}\left\langle\psi\right|\rho_{S_1}\left|\psi\right\rangle \\ &= \frac{1}{\mathrm{Tr}[\rho_{S_1}]}\left(\mathrm{Tr}[\rho_{S_1}] - \mathrm{Tr}[\Pi_{|\psi\rangle}\rho_{S_1}]\right) \\ &= \mathrm{Tr}\left[(1-\Pi_{|\psi\rangle})\rho_{S_1}\right] + O(p^2), \end{aligned} \tag{4.24}$$

where $\Pi_{|\psi\rangle} \equiv \left|\psi\right\rangle \left\langle\psi\right|$. To obtain the last line, we have used the fact that

$$\mathrm{Tr}[\rho_{S_1}] = \mathrm{Tr}[\rho] = 1 - O(p) \tag{4.25}$$

together with the fact that

$$\mathrm{Tr}\left[(1-\Pi_{|\psi\rangle})\rho_{S_1}\right] = O(p). \tag{4.26}$$

Inserting in Eq. (4.23) into Eq. (4.24) yields,

$$1 - F = \frac{p}{M} + O(p^2). \tag{4.27}$$

Thus, to leading order, the infidelity decreases as $1/M$.

Let us summarize the results of this analysis and discuss its implications (see Table 4.1). The error suppression that this scheme realizes is embodied in the $1/M$ dependence of the infidelity. As the number of copies $M$ is increased, the infidelity correspondingly decreases, allowing one to perform higher fidelity computations than would be possible with a single noisy device. At best, the infidelity can be suppressed to $O(p^2)$, which constitutes a quadratic improvement. Unfortunately, the failure probability and associated overhead pose practical challenges for this scheme. The failure probability increases proportionately with $M$, so that greater error suppression also implies an increased probability of failure. The only way to mitigate this failure probability is to apply the projections more frequently, so that the probability of error in the time between projections (i.e. $p$) is reduced. Moreover, the $O(M)$ overhead could be practically prohibitive. This scheme requires $M$ full copies of a quantum computer all running the same algorithm in parallel. For large $M$, it is likely more practical to use these resources to achieve exponential error suppression in a single quantum computer via quantum error correction, rather than a $1/M$ suppression via error symmetrization on an ensemble of noisy quantum computers.

In the next section, we present an error-suppression scheme that improves on the scheme of Ref. [117] with respect to both the failure probability and hardware overhead. As we discuss later in this chapter, these practical improvements makes

| Infidelity | Failure probability | Overhead |
|---|---|---|
| $\frac{p}{M} + O(p^2)$ | $(M-1)p + O(p^2)$ | $O(M)$ |

Table 4.1: Summary of the error-suppression scheme of Ref. [117]. Though the infidelity decreases with $M$, the failure probability and hardware overhead both increase linearly with $M$.

the scheme suitable for suppressing the query infidelity of QRAM.

## 4.2 A general scheme for hardware-efficient error suppression

In this section, we present a novel, hardware-efficient scheme for error suppression. This scheme can be viewed as a fusion of the error symmetrization scheme of Ref. [117] with QROM (Section 2.2.4).

Our scheme accomplishes the following task. One is given a quantum state $|\psi\rangle$ and access to a channel, $\mathcal{U}$, which constitutes a noisy implementation of a target unitary operation $U$. The goal is to prepare the state $U|\psi\rangle$ with as high fidelity as is possible. Any procedure that uses these resources to prepare a state $\rho_\psi$ satisfying

$$\langle\psi|U^\dagger \rho_\psi U|\psi\rangle > \langle\psi|U^\dagger \mathcal{U}(\psi) U|\psi\rangle , \tag{4.28}$$

will be referred to as an error-suppression scheme. Note that this task differs from the one considered in Ref. [117] in that we only assume access to a single copy of the state $|\psi\rangle$, as opposed to $M$ copies. Having only one copy of $|\psi\rangle$ is a more restrictive assumption, which means that the suppression scheme we develop is more widely applicable. Indeed, our scheme only requires a single quantum computer, as opposed to $M$ quantum computers operating in parallel.

We emphasize that this error suppression task should not be confused with the related but distinct task of error mitigation [127–129]. Error mitigation protocols provide means of reducing the error in measured expectation values, a feat which is particularly useful in the context of near-term variational quantum algorithms. However, error mitigation protocols do not generally enable one to prepare quantum states or implement quantum operations with higher fidelity. For this reason, error mitigation protocols cannot be applied to improve the query fidelity of QRAM.

While we do not discuss error mitigation in this thesis, we note that a number of recent error mitigation protocols exploit similar ideas [130–133]. These protocols, termed virtual distillation, employ symmetrization in order to achieve an impressive *exponential* suppression of errors in measured expectation values. Unfortunately, they cannot directly be applied to prepare quantum states or implement quantum operations with higher fidelity. As a result, these schemes cannot be directly compared with our own.

### 4.2.1 A simple example

In order to provide a pedagogical introduction to our error-suppression scheme, we begin by describing its simplest incarnation, illustrated in Fig. 4.3. The circuit in the figure has three registers: a single-qubit register $A$ (initialized in $|0\rangle$), an $n$-qubit register $B$ (containing the target state $|\psi\rangle$), and another $n$-qubit register $C$ (initialized in an arbitrary state $|\phi\rangle$). At the end of the circuit, the register $C$ is discarded, and the qubit $A$ is measured. We postselect on obtaining the outcome $|0\rangle$. After successful postselection, the reduced state of register $B$, $\rho_\psi$, constitutes the output. Note that the circuit's sequential iteration over the different computational basis states of $A$ is similar to QROM, while the postselected measurement turns out to enact a kind of symmetrization reminiscent of Ref. [117]. For these reasons, this scheme can be viewed as a fusion of QROM and error symmetrization.

The operation circuit in Fig. 4.3 can be understood as follows. The initial Hadamard

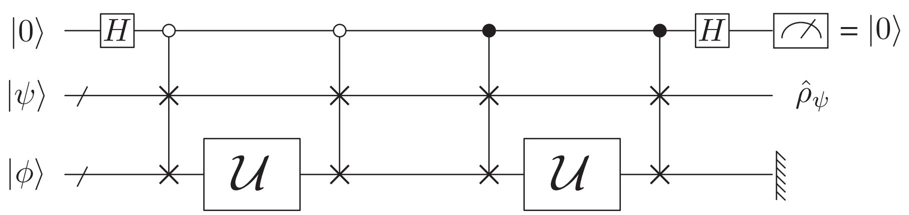


Figure 4.3: A minimal error-suppression circuit. The circuit uses $M = 2$ applications of the channel $\mathcal{U}$ to suppress the infidelity of the output state $\rho_\psi$ by a factor of 1/2 (for applicable channels).

prepares the $A$ qubit in an equal superposition. Then, conditioned on the state of this qubit, the channel $\mathcal{U}$ is applied to $|\psi\rangle$ and $|\phi\rangle$ with different ordering. If the $A$ qubit is $|0\rangle$, the first pair of controlled-SWAP gates is triggered, resulting in $\mathcal{U}$ being applied to $|\psi\rangle$ first. Then the next pair of controlled-SWAP gates is not triggered, resulting in $\mathcal{U}$ being applied to $|\phi\rangle$ second. On the other hand, if the $A$ qubit is $|1\rangle$, this ordering is reversed: $\mathcal{U}$ is applied to $|\phi\rangle$ first then $|\psi\rangle$ second. Note that, regardless of the state of the $A$ qubit, the channel $\mathcal{U}$ is applied to $|\psi\rangle$ exactly once, hence it is reasonable to expect that $\rho_\psi$ would be close to $U\,|\psi\rangle$. What is perhaps less obvious is why $\rho_\psi$ should be closer to the desired state, $U\,|\psi\rangle$, than $\mathcal{U}(\psi)$. As we show below, this is because the postselection enacts an effective "symmetrization" of the channel applied to $|\psi\rangle$, such that the effects of errors are suppressed.

To analyze this scheme quantitatively, we adopt the following error model. We express the channel $\mathcal{U}$ as a completely positive trace preserving map with Kraus representation

$$\mathcal{U}(\rho) = \sum_i K_i \rho K_i^\dagger, \tag{4.29}$$

and the fidelity of the channel with the target state $U\,|\psi\rangle$ is defined to be

$$\langle\psi|U^\dagger \mathcal{U}(\psi) U|\psi\rangle = \sum_i \langle\psi|U^\dagger K_i|\psi\rangle \, \langle\psi|K_i^\dagger U|\psi\rangle \equiv 1 - p. \tag{4.30}$$

We stress that this is a Markovian error model and that we do not consider situations where there could be temporal correlations in the noise from one application of $\mathcal{U}$ to the next[1]. To encompass errors at points in the circuit other than $\mathcal{U}$, we suppose that all other operations (single- and multi-qubit gates, measurements, and idling) are also each subject to error with some probability $p'$. Our scheme is effective at suppressing errors when $p' \ll p$. This hierarchy of error rates could be engineered in a variety of ways. It may emerge naturally if, for example, the operation $\mathcal{U}$ requires many ancillary qubits to implement, such that it is significantly noiser than the other operations in the circuit. Alternatively, error correction and fault-tolerant gadgets could be used to suppress errors during the non-$\mathcal{U}$ operations, while the operation $\mathcal{U}$ could be implemented either without error correction or in a way that is not fault-tolerant. (One could, for example, implement operation $\mathcal{U}$ by decoding the logical information in register $C$, applying a transformation to the unencoded physical qubits, then re-encoding the information. As discussed in Section 4.3, this approach could be useful in the context of QRAM, which may be prohibitively expensive to implement in a fault-tolerant manner.) For the purpose of our analysis, we do not concern ourselves with the specifics of how this hierarchy of error rates is engineered. Instead, we simply analyze the circuit of Fig. 4.3 under the assumption that $p'$ errors have a negligible impact.

With this error model, we proceed to calculate the final state of the system, from which we can ascertain effectiveness of the error suppression. Prior to the final Hadamard gate, the state of the full $A$, $B$, $C$ system is

$$\frac{1}{2}\sum_{i_1,i_2}\Big[|0\rangle^A (K_{i_1}|\psi\rangle^B K_{i_2}|\phi\rangle^C) + |1\rangle^A (K_{i_2}|\psi\rangle^B K_{i_1}|\phi\rangle^C)\Big]\Big[\text{H.c.}\Big], \tag{4.31}$$

where [H.c.] denotes the Hermitian conjugate of the state in the first set of brackets.

[1] It is not difficult to see that our scheme would fail to suppress such errors. In effect, our scheme works by correcting broken symmetries that can arise in the presence of Markovian errors (e.g., an error occurs during one application of $\mathcal{U}$ but not the other). If the same error always occured during applications of $\mathcal{U}$, there is no broken symmetry to fix, and the error suppression fails.

After the last Hadamard and successful postselection, the resulting (unnormalized) state of the $B$ and $C$ systems, which we denote $\rho_{BC}$, is

$$\begin{aligned}\rho_{BC} =& \sum_{i_1,i_2}\left[\frac{1}{2}\left(K_{i_1}\otimes K_{i_2}+K_{i_2}\otimes K_{i_1}\right)\left|\psi\right\rangle^B\left|\phi\right\rangle^C\right]\Big[\text{H.c.}\Big]\\ =& \sum_{i_1,i_2}\left[S_{i_1,i_2}\left|\psi\right\rangle^B\left|\phi\right\rangle^C\right]\Big[\text{H.c.}\Big],\end{aligned} \tag{4.32}$$

where we have defined the symmetrized joint Kraus operators

$$S_{i_1,i_2} = \frac{1}{2}\left(K_{i_1}\otimes K_{i_2}+K_{i_2}\otimes K_{i_1}\right). \tag{4.33}$$

We say that these joint Kraus operators are symmetrized because they are invariant under permutations of the indices, i.e., $S_{i_1,i_2} = S_{i_2,i_1}$. Thus, one interpretation of the circuit in Fig. 4.3 is that it uses two applications of the channel $\mathcal{U}$ to synthesize a more symmetric joint channel that is described by the (subnormalized) Kraus operators $S_{i_1,i_2}$. To proceed, it is useful to expand out the terms in $\rho_{BC}$, then group them into two classes as follows,

$$\begin{aligned}\rho_{BC} =& \frac{1}{2}\sum_{i_1,i_2}\Big(K_{i_1}\left|\psi\right\rangle\left\langle\psi\right|K_{i_1}^\dagger\otimes K_{i_2}\left|\phi\right\rangle\left\langle\phi\right|K_{i_2}^\dagger\Big)\\ &+\frac{1}{2}\sum_{i_1,i_2}\Big(K_{i_1}\left|\psi\right\rangle\left\langle\psi\right|K_{i_2}^\dagger\otimes K_{i_2}\left|\phi\right\rangle\left\langle\phi\right|K_{i_1}^\dagger\Big).\end{aligned} \tag{4.34}$$

We refer to terms on the first line as the *paired terms*, and those on the second line as the *cross terms*. This grouping is convenient because it allows us to separately quantify the contributions of the different terms to the infidelity.

### Infidelity

Let us calculate the infidelity of the state reduced state $\rho_\psi = \mathrm{Tr}_C[\rho_{BC}]$. Employing Eq. (4.24), we have that

$$1 - F = \mathrm{Tr}\left[\left(\Pi^{\perp}_{U|\psi\rangle} \otimes I\right)\rho_{BC}\right] + O(p^2), \tag{4.35}$$

where we have defined

$$\Pi^{\perp}_{U|\psi\rangle} \equiv 1 - \Pi_{U|\psi\rangle}. \tag{4.36}$$

Next, we evaluate the contributions to the infidelity from the paired terms and cross terms separately. Starting with the paired terms, we obtain

$$\begin{aligned}
&\mathrm{Tr}\left[\frac{1}{2}\sum_{i_1,i_2}\left(\Pi^{\perp}_{U|\psi\rangle}K_{i_1}|\psi\rangle\langle\psi|K^{\dagger}_{i_1} \otimes K_{i_2}|\phi\rangle\langle\phi|K^{\dagger}_{i_2}\right)\right] \\
&= \frac{1}{2}\mathrm{Tr}\left[\sum_{i_1}\left(\Pi^{\perp}_{U|\psi\rangle}K_{i_1}|\psi\rangle\langle\psi|K^{\dagger}_{i_1}\right)\right]\mathrm{Tr}\left[\sum_{i_2}\left(K_{i_2}|\phi\rangle\langle\phi|K^{\dagger}_{i_2}\right)\right] \\
&= \frac{1}{2}\mathrm{Tr}\left[\Pi^{\perp}_{U|\psi\rangle}\mathcal{U}(\psi)\right]\mathrm{Tr}\Big[\mathcal{U}(\phi)\Big] \\
&= p/2.
\end{aligned} \tag{4.37}$$

To obtain the last line we have used Eq. (4.30) and the fact that $\mathcal{U}$ is trace-preserving. Similarly, we evaluate the contribution from the cross terms,

$$\begin{aligned}
&\mathrm{Tr}\left[\frac{1}{2}\sum_{i_1,i_2}\left(\Pi^{\perp}_{U|\psi\rangle}K_{i_1}|\psi\rangle\langle\psi|K^{\dagger}_{i_2} \otimes K_{i_2}|\phi\rangle\langle\phi|K^{\dagger}_{i_1}\right)\right] \\
&= \frac{1}{2}\sum_{i_1,i_2}\langle\psi|K^{\dagger}_{i_2}\Pi^{\perp}_{U|\psi\rangle}K_{i_1}|\psi\rangle\langle\phi|K^{\dagger}_{i_1}K_{i_2}|\phi\rangle\,.
\end{aligned} \tag{4.38}$$

Combining these two contributions, we have

$$1 - F = \frac{p}{2} + \frac{1}{2}\sum_{i_1,i_2}\langle\psi|K^{\dagger}_{i_2}\Pi^{\perp}_{U|\psi\rangle}K_{i_1}|\psi\rangle\langle\phi|K^{\dagger}_{i_1}K_{i_2}|\phi\rangle + O(p^2). \tag{4.39}$$

From Eq. (4.39), we see that the circuit of Fig. 4.3 successfully suppresses errors whenever the second term, that from the cross terms, is $< p/2$. In particular, we find that the infidelity is maximally suppressed (i.e., reduced by a factor of 2) whenever

$$\mathcal{C}_2 \equiv \frac{1}{2} \sum_{i_1, i_2} \langle\psi| K_{i_2}^\dagger \Pi_{U|\psi\rangle}^\perp K_{i_1} |\psi\rangle \, \langle\phi| K_{i_1}^\dagger K_{i_2} |\phi\rangle = O(p^2), \tag{4.40}$$

to leading order. We refer to Eq. (4.40) as the criterion for maximal suppression, and we find that this criterion is satisfied for many channels of practical relevance. As an example, consider the class of mixed-unitary channels for which

$$K_0 = \sqrt{1-p}\, U, \tag{4.41}$$

and the remaining Kraus operators $K_{i>0}$ are proportional to unitary operators. This corresponds to the situation where the desired operation $U$ is implemented with probability $(1-p)$ and some other operation is implemented with probability $p$. Such a model can be used to describe quantum operations subject to bit-flip, dephasing, or depolarizing errors, for example. Restricting our attention to channels of this form, let us evaluate the contribution to the infidelity from the cross terms. We can immediately leverage the fact that

$$\Pi_{U|\psi\rangle}^\perp K_0 \,|\psi\rangle = 0, \tag{4.42}$$

to eliminate the terms where either $i_1 = 0$ or $i_2 = 0$. For the remaining terms, note that the Kraus' operator's completeness relation implies,

$$p = \sum_{i>0} K_i^\dagger K_i. \tag{4.43}$$

Thus each $K_{i>0}$ is proportional to a unitary with a constant of proportionality that is upper-bounded by $\sqrt{p}$. It follows that

$$\mathcal{C}_2 = \langle\psi|K_{i_2}^\dagger \Pi_{U|\psi\rangle}^\perp K_{i_1}|\psi\rangle \langle\phi|K_{i_1}^\dagger K_{i_2}|\phi\rangle = O(p^2) \tag{4.44}$$

for all $i_1, i_2 > 0$. The maximal suppression criterion [Eq. (4.40)] is therefore satisfied, and the infidelity of the output state is

$$1 - F = \frac{p}{2} + O(p^2). \tag{4.45}$$

Let us provide a contrasting example—one where the maximal suppression criterion is not satisfied. Consider the channel $\mathcal{U}$ defined by the single non-zero Kraus operator,

$$K_0 = R_p U, \tag{4.46}$$

where $R_p$ is small coherent rotation in the plane containing $|\psi\rangle$ and some orthogonal state $|\psi^\perp\rangle$,

$$R_p |\psi\rangle = \sqrt{1-p}\,|\psi\rangle + \sqrt{p}\,|\psi^\perp\rangle . \tag{4.47}$$

This channel corresponds to an application of $U$ followed by some deterministic coherent error, e.g., an over-rotation due to parameter miscalibration. Evaluating the maximal suppression criterion [Eq. (4.40)] for this channel yields,

$$\mathcal{C}_2 = \frac{1}{2} \langle\psi|(R_pU)^\dagger \Pi_{U|\psi\rangle}^\perp (R_pU)|\psi\rangle \langle\phi|(R_pU)^\dagger(R_pU)|\phi\rangle = \frac{p}{2}, \tag{4.48}$$

so that the infidelity of the final state is given by

$$1 - F = \frac{p}{2} + \frac{p}{2} = p. \tag{4.49}$$

For this sort of coherent error, the scheme does not yield any infidelity suppression. This is to be expected, because in this case $\mathcal{U}$ is an *entropy-non-increasing* channel.

That is, $S(\rho) = S(\mathcal{U}(\rho))$ for arbitrary $\rho$, where $S = -\text{Tr}[\rho \log \rho]$ is the von Neumann entropy. For such channels, there is no inherent randomness, so repeated applications of the channel never give rise to any asymmetries that could be removed by our post-selection scheme. This example illustrates that our scheme is limited to suppressing the infidelity associated with stochastic errors (those which can increase entropy).

To summarize, we have shown that the output state of the circuit in Fig. 4.3 has an infidelity given by Eq. (4.39). Moreover, we find that for any channels which satisfy the criterion Eq. (4.40), the infidelity is reduced by a factor of 2. (In Section 4.2.2, we show how the scheme can be generalized to achieve greater error suppression.)

### Failure probability

Before moving to our general error-suppression scheme, we calculate the failure probability of the simplified scheme of Fig. 4.3. The failure probability, $P_{\text{fail}}$, is the probability that the measurement does not yield $|0\rangle$, and is given by

$$P_{\text{fail}} = 1 - \text{Tr}[\rho_{BC}]. \tag{4.50}$$

We have

$$\begin{aligned} P_{\text{fail}} = 1 - \frac{1}{2} \sum_{i_1, i_2} \Big( \langle \psi | K_{i_1}^\dagger K_{i_1} | \psi \rangle \, \langle \phi | K_{i_2}^\dagger K_{i_2} | \phi \rangle \Big) \\ - \frac{1}{2} \sum_{i_1, i_2} \Big( \langle \psi | K_{i_2}^\dagger K_{i_1} | \psi \rangle \, \langle \phi | K_{i_1}^\dagger K_{i_2} | \phi \rangle \Big) . \end{aligned} \tag{4.51}$$

We can simplify this expression by observing that the first line is equivalent to

$$1 - \frac{1}{2} \text{Tr}\Big[ \mathcal{U}(\psi) \Big] \text{Tr}\Big[ \mathcal{U}(\phi) \Big] = \frac{1}{2}, \tag{4.52}$$

so we have

$$P_{\text{fail}} = \frac{1}{2} - \frac{1}{2}\sum_{i_1,i_2} \langle\psi|K_{i_2}^\dagger K_{i_1}|\psi\rangle \langle\phi|K_{i_1}^\dagger K_{i_2}|\phi\rangle\,. \tag{4.53}$$

We see that the failure probability is dependent on the channel $\mathcal{U}$, as well as the states $|\psi\rangle$ and $|\phi\rangle$. For mixed-unitary channels with $K_0 = \sqrt{1-p}\,U$ we can straightforwardly obtain an upper bound on this failure probability by including only the $i_1 = 0$, $i_2 = 0$ term of the sum,

$$\begin{aligned} P_{\text{fail}} &\le \frac{1}{2} - \frac{1}{2}\langle\psi|K_0^\dagger K_0|\psi\rangle \langle\phi|K_0^\dagger K_0|\phi\rangle \\ &= \frac{1}{2} - \frac{1}{2}(1-p)^2 \\ &= p + O(p^2). \end{aligned} \tag{4.54}$$

We provide a more comprehensive analysis of the failure probability in the next section.

### 4.2.2 General error-suppression scheme

The circuit in Fig. 4.3 uses 2 applications of the channel $\mathcal{U}$ to suppress the infidelity of the output state by a factor of 1/2. Generalizing this scheme, the circuit in Fig. 4.4 uses $M$ applications of the channel to suppress the infidelity of the output state by a factor of $1/M$ (the generalized circuit reduces to that in Fig. 4.3 for $M = 2$). The generalized circuit has three registers: a $(\log M)$-qubit register $A$ (initialized in $|0\rangle^{\otimes \log M}$), an $n$-qubit register $B$ (containing the target state $|\psi\rangle$), and another $n$-qubit register $C$ (initialized in an arbitrary state $|\phi\rangle$). As before, at the end of the circuit, register $C$ is discarded, and register $A$ is measured. We postselect on obtaining the outcome $|0\rangle^{\otimes \log M}$. After successful postselection, the reduced state of register $B$, $\rho_\psi$, constitutes the output. Here again, the scheme can be viewed as a fusion of QROM with the error symmetrization scheme of Ref. [117]: the circuit's sequential iteration

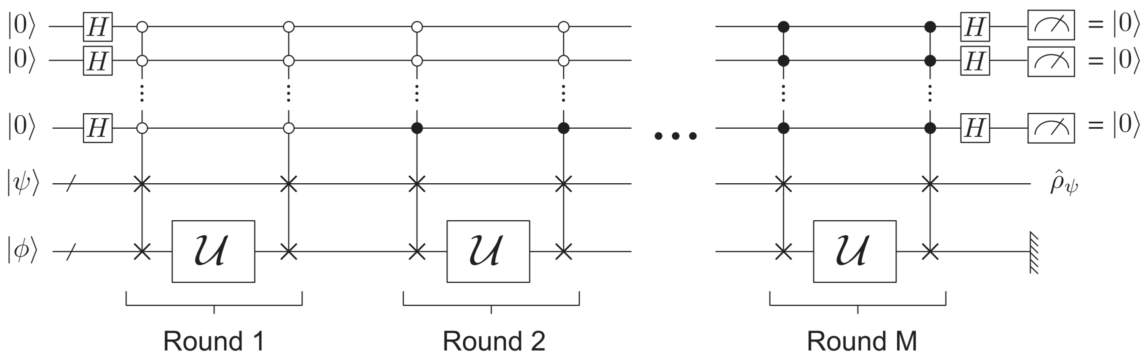


Figure 4.4: General error suppression circuit. The circuit uses $M$ applications of the channel $\mathcal{U}$ to suppress the infidelity of the output state $\rho_\psi$ by a factor of $1/M$ (for applicable channels).

over the different computational basis states of $A$ is similar to QROM, while the postselected measurement enacts a symmetrization reminiscent of Ref. [117].

To analyze this circuit, we adopt the same error model as before. We assume that the channel $\mathcal{U}$ has a Kraus decomposition with Kraus operators $K_i$, and that all operations in the circuit other than $\mathcal{U}$ are subject to a negligibly small error $p'$ (specifically, we now require $p' \ll p/M$ for $p'$ errors to be negligible).

With this error model, we proceed to calculate the final state of the system. Prior to the final layer of Hadamard gates, the state of the full $A$, $B$, $C$ system is

$$\frac{1}{M} \sum_{i_0,\ldots,i_{M-1}} \left[ \sum_{j=0}^{M-1} |j\rangle^A K_{i_j} |\psi\rangle^B \overline{K}_{i_j} |\phi\rangle^C \right] \left[ \text{H.c.} \right], \tag{4.55}$$

where we have defined

$$\overline{K}_{i_j} = \prod_{n \neq j} K_{i_n}. \tag{4.56}$$

This expression for the final state can be derived using a quantum trajectory picture. At each round $j$, we replace the channel $\mathcal{U}$ with a corresponding Kraus operator $K_{i_j}$. Together, these Kraus operators specify the quantum trajectory of the system, and the final state can be calculated by incoherently adding all such trajectories (i.e., by summing over $i_0, \ldots i_{M-1}$ in the final density matrix). In this picture, when register

$A$ is in state $|j\rangle$, the operator $K_{i_j}$ is applied to $|\psi\rangle$, and all other Kraus operators are applied to $|\phi\rangle$. Now, after the last layer of Hadamards and successful postselection, the resulting (unnormalized) state of the $B$ and $C$ systems is

$$\begin{aligned}\rho_{BC} &= \sum_{i_0,\ldots,i_{M-1}} \left[\frac{1}{M}\sum_{j=0}^{M-1}\left(K_{i_j}\otimes\overline{K}_{i_j}\right)|\psi\rangle^B|\phi\rangle^C\right]\left[\text{H.c.}\right] \\ &= \sum_{i_0,\ldots,i_{M-1}}\left[S_{i_0,\ldots,i_{M-1}}|\psi\rangle^B|\phi\rangle^C\right]\left[\text{H.c.}\right], \end{aligned} \tag{4.57}$$

where we have defined the joint Kraus operators

$$S_{i_0,\ldots,i_{M-1}} \equiv \frac{1}{M}\sum_{j=0}^{M-1}\left(K_{i_j}\otimes\overline{K}_{i_j}\right). \tag{4.58}$$

As examples, for the case of $M=2$ we have

$$S_{i_0,i_1} = \frac{1}{2}\left(K_{i_0}\otimes K_{i_1} + K_{i_1}\otimes K_{i_0}\right), \tag{4.59}$$

and for $M=3$,

$$S_{i_0,i_1,i_2} = \frac{1}{3}\left(K_{i_0}\otimes K_{i_2}K_{i_1} + K_{i_1}\otimes K_{i_2}K_{i_0} + K_{i_2}\otimes K_{i_1}K_{i_0}\right). \tag{4.60}$$

We note that $S_{i_0,i_1}$ is symmetric under permutation of its indices and can thus properly be called a symmetrized operator. In contrast, for $M>2$, $S_{i_0,\ldots,i_{M-1}}$ is not technically symmetrized according to this definition. Nevertheless, the the joint channel described by the Kraus operators $S_{i_0,\ldots,i_{M-1}}$ can still suppress errors, as we show below.

To proceed, it is again useful to expand out the terms in $\rho_{BC}$,

$$\rho_{BC} = \frac{1}{M^2}\sum_{i_0,\ldots,i_M}\sum_{a,b=0}^{M-1} K_{i_a}|\psi\rangle\langle\psi|K_{i_b}^\dagger\otimes\overline{K}_{i_a}|\phi\rangle\langle\phi|\overline{K}_{i_b}^\dagger \tag{4.61}$$

then group them into two classes as follows,

$$\rho_{BC} = \sum_{i_0,\ldots,i_M} \left[ \frac{1}{M^2} \sum_{a,b=0}^{M-1} (\delta_{a,b}) \left( K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_b}^\dagger \otimes \overline{K}_{i_a} \left|\phi\right\rangle \left\langle\phi\right| \overline{K}_{i_b}^\dagger \right) \right.$$

$$\left. + \frac{1}{M^2} \sum_{a,b=0}^{M-1} (1-\delta_{a,b}) \left( K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_b}^\dagger \otimes \overline{K}_{i_a} \left|\phi\right\rangle \left\langle\phi\right| \overline{K}_{i_b}^\dagger \right) \right], \tag{4.62}$$

$$= \sum_{i_0,\ldots,i_M} \left[ \frac{1}{M^2} \sum_{a=0}^{M-1} \left( K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_a}^\dagger \otimes \overline{K}_{i_a} \left|\phi\right\rangle \left\langle\phi\right| \overline{K}_{i_a}^\dagger \right) \right.$$

$$\left. + \frac{1}{M^2} \sum_{a\neq b} \left( K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_b}^\dagger \otimes \overline{K}_{i_a} \left|\phi\right\rangle \left\langle\phi\right| \overline{K}_{i_b}^\dagger \right) \right]. \tag{4.63}$$

We refer to terms on the first line of Eq. (4.63) as the *paired terms*, and those on the second line as the *cross terms*. This grouping is convenient because it allows us to separately quantify the contributions of the different terms to the infidelity.

### Infidelity

Let us calculate the infidelity of the reduced state $\rho_\psi = \mathrm{Tr}_C[\rho_{BC}]$. As before, we have that

$$1 - F = \mathrm{Tr}\left[ \left( \Pi_{U\left|\psi\right\rangle}^\perp \otimes I \right) \rho_{BC} \right] + O(p^2), \tag{4.64}$$

The contribution to the infidelity from the paired terms is

$$\mathrm{Tr}\left[ \sum_{i_0,\ldots,i_{M-1}} \frac{1}{M^2} \sum_{a=0}^{M-1} \left( \Pi_{U\left|\psi\right\rangle}^\perp K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_a}^\dagger \otimes \overline{K}_{i_a} \left|\phi\right\rangle \left\langle\phi\right| \overline{K}_{i_a}^\dagger \right) \right]$$

$$= \frac{1}{M^2} \sum_{a=0}^{M-1} \mathrm{Tr}\left[ \sum_{i_a} \Pi_{U\left|\psi\right\rangle}^\perp K_{i_a} \left|\psi\right\rangle \left\langle\psi\right| K_{i_a}^\dagger \right] \mathrm{Tr}\left[ \sum_{i_{b\neq a}} \left( \prod_{b\neq a} K_{i_b} \right) \left|\phi\right\rangle \left\langle\phi\right| \left( \prod_{b\neq a} K_{i_b} \right)^\dagger \right]$$

$$= \frac{1}{M^2} \sum_{a=0}^{M-1} \mathrm{Tr}\left[ \Pi_{U\left|\psi\right\rangle}^\perp \mathcal{U}(\psi) \right] \mathrm{Tr}\Big[ \mathcal{U}^{(M-1)}(\phi) \Big]$$

$$= \frac{p}{M}, \tag{4.65}$$

where in the second-to-last line we use $\mathcal{U}^{(M-1)}$ to denote $M-1$ successive applications of the channel $\mathcal{U}$. Similarly, the contribution from the cross terms is,

$$\begin{aligned}&\mathrm{Tr}\left[\sum_{i_0,\ldots,i_{M-1}}\frac{1}{M^2}\sum_{a\neq b}\left(\Pi^{\perp}_{U|\psi\rangle}K_{i_a}\left|\psi\right\rangle\left\langle\psi\right|K^{\dagger}_{i_b}\otimes\overline{K}_{i_a}\left|\phi\right\rangle\left\langle\phi\right|\overline{K}^{\dagger}_{i_b}\right)\right]\\&=\frac{1}{M^2}\sum_{i_1,i_2}\sum_{a\neq b}\langle\psi|K^{\dagger}_{i_b}\Pi^{\perp}_{U|\psi\rangle}K_{i_a}|\psi\rangle\,\langle\phi|\overline{K}^{\dagger}_{i_b}\overline{K}_{i_a}|\phi\rangle\,.\end{aligned} \tag{4.66}$$

Combining these two contributions, we have

$$1-F=\frac{p}{M}+\frac{1}{M^2}\sum_{i_1,i_2}\sum_{a\neq b}\langle\psi|K^{\dagger}_{i_b}\Pi^{\perp}_{U|\psi\rangle}K_{i_a}|\psi\rangle\,\langle\phi|\overline{K}^{\dagger}_{i_b}\overline{K}_{i_a}|\phi\rangle+O(p^2). \tag{4.67}$$

From Eq. (4.67), we see that the infidelity is maximally suppressed whenever

$$\mathcal{C}_M\equiv\frac{1}{M^2}\sum_{i_1,i_2}\sum_{a\neq b}\langle\psi|K^{\dagger}_{i_b}\Pi^{\perp}_{U|\psi\rangle}K_{i_a}|\psi\rangle\,\langle\phi|\overline{K}^{\dagger}_{i_b}\overline{K}_{i_a}|\phi\rangle=O(p^2), \tag{4.68}$$

to leading order. Thus Eq. (4.68) is criterion for maximal suppression for general $M$. As before, we find that this criterion is satisfied for many channels of practical relevance. For example, by the same argument as for the $M=2$ case, we have $\mathcal{C}_M=O(p^2)$ for the class of mixed-unitary channel with $K_0=\sqrt{1-p}\,U$. Interestingly, in certain situations, we can also have $\mathcal{C}_M=\mathcal{C}_2$. This equivalence holds, for example, if the Kraus operators are mutually commuting ($[K_i,K_j]=0$ for all $i,j$), or if $|\phi\rangle$ is stationary under the action of the channel ($K_i\left|\phi\right\rangle\propto\left|\phi\right\rangle$). In such cases, one needs only to check the comparatively simpler criterion of Eq. (4.40), rather than the general criterion of Eq. (4.68). This situation is relevant to the case of QRAM, for which it is possible to choose a state $|\phi\rangle$ that is invariant under the QRAM channel (Section 4.3).

### Failure probability

Let us calculate the failure probability,

$$P_{\text{fail}} = 1 - \text{Tr}[\rho_{BC}], \tag{4.69}$$

of the general scheme (Fig. 4.4). Inserting Eq. (4.61) into the above expression yields,

$$P_{\text{fail}} = 1 - \frac{1}{M^2} \sum_{i_0,\ldots,i_{M-1}} \sum_{a,b=0}^{M-1} \langle\psi| K_{i_b}^\dagger K_{i_a} |\psi\rangle \langle\phi| \overline{K}_{i_b}^\dagger \overline{K}_{i_a} |\phi\rangle \,. \tag{4.70}$$

Unfortunately, this expression does not readily lend itself to additional simplification, but it can be evaluated or bounded in specific cases of interest. For example, let us return to the example of a mixed-unitary channel with $K_0 = \sqrt{1-p}\,U$. For this channel, a trivial upper bound on the failure probability is obtained by consider only the terms in the sum for which $i_0, \ldots, i_{M-1} = 0$,

$$\begin{aligned} P_{\text{fail}} &\leq 1 - \frac{1}{M^2} \sum_{a,b=0}^{M-1} \langle\psi| K_0^\dagger K_0 |\psi\rangle \langle\phi| K_0^{\dagger(M-1)} K_0^{(M-1)} |\phi\rangle \\ &= 1 - (1-p)^M = Mp + O(p^2). \end{aligned} \tag{4.71}$$

We note that this linear scaling with $M$ matches the scaling of Ref. [117].

Remarkably, in some situations the failure probability of our scheme can be bounded by a constant, i.e. $P_{\text{fail}}$ does not increase with $M$. This is a highly desirable property, because it means that the infidelity can be significantly suppressed in a *near-deterministic* manner. Indeed, a near-deterministic error-suppression scheme could be used multiple times during the course of a quantum algorithm without significantly degrading the algorithm's overall success probability. Let us enumerate some situations where this favorable bound can be obtained.

**Commuting Kraus operators.** Suppose that all of $\mathcal{U}$'s Kraus operators mutually commute, $[K_i, K_j] = 0$ for all $i, j$. Under this assumption we have that

$$\begin{aligned} P_{\text{fail}} &= 1 - \frac{1}{M^2} \sum_{i_0,\ldots,i_{M-1}} \sum_{a,b=0}^{M-1} \langle\psi|K_{i_b}^\dagger K_{i_a}|\psi\rangle \langle\phi|K_{i_a}^\dagger \left(\prod_{n\neq a,b} K_{i_n}\right)^\dagger \left(\prod_{n\neq a,b} K_{i_n}\right) K_{i_b}|\phi\rangle \\ &= 1 - \frac{1}{M^2} \sum_{i_a,i_b} \sum_{a,b=0}^{M-1} \langle\psi|K_{i_b}^\dagger K_{i_a}|\psi\rangle \langle\phi|K_{i_a}^\dagger K_{i_b}|\phi\rangle, \end{aligned} \tag{4.72}$$

where we have used the Kraus operators' completeness relation, $\sum_i K_i^\dagger K_i = 1$, to obtain the second line. We proceed by breaking this expression into two parts,

$$\begin{aligned} P_{\text{fail}} = 1 - \Bigg[ &\frac{1}{M} \sum_{i_a} \langle\psi|K_{i_a}^\dagger K_{i_a}|\psi\rangle \langle\phi|K_{i_a}^\dagger K_{i_a}|\phi\rangle \\ &+ \frac{1}{M^2} \sum_{i_a,i_b} \sum_{a\neq b} \langle\psi|K_{i_b}^\dagger K_{i_a}|\psi\rangle \langle\phi|K_{i_a}^\dagger K_{i_b}|\phi\rangle \Bigg] \\ = 1 - \Bigg[ &\frac{1}{M} + \frac{1}{M^2} \sum_{i_a,i_b} \sum_{a\neq b} \langle\psi|K_{i_b}^\dagger K_{i_a}|\psi\rangle \langle\phi|K_{i_a}^\dagger K_{i_b}|\phi\rangle \Bigg]. \end{aligned} \tag{4.73}$$

Now, let us further suppose that $K_0 = \sqrt{1-p}\,U$, as before. Then we obtain the bound

$$\begin{aligned} P_{\text{fail}} &\leq 1 - \left[\frac{1}{M} + \frac{1}{M^2} \sum_{a,b=0}^{M-1} \langle\psi|K_0^\dagger K_0|\psi\rangle \langle\phi|K_0^\dagger K_0|\phi\rangle \right] \\ &= 1 - \left[\frac{1}{M} + \frac{M(M-1)}{M^2}(1-p)^2\right] = 2p\left(1 - \frac{1}{M}\right) + O(p^2). \end{aligned} \tag{4.74}$$

As $M$ increases, this bound approaches a constant, $2p$. In Section 4.2.3, we demonstrate numerically that this bound is tight for some channels.

**Stationary $|\phi\rangle$.** Suppose that the state $|\phi\rangle$ is stationary under the channel $\mathcal{U}$, that is, $K_i |\phi\rangle \propto |\phi\rangle$ for all $i$. This property implies that

$$K_i K_j |\phi\rangle = K_j K_i |\phi\rangle \tag{4.75}$$

for all $i, j$. Equivalently,

$$[K_i, K_j] \left|\phi\right\rangle = 0 \tag{4.76}$$

for all $i, j$. Notice that this is a weaker assumption than that considered in the previous paragraph. Nevertheless, we can then employ the exact same argument to show that

$$P_{\text{fail}} \leq 2p\left(1 - \frac{1}{M}\right) + O(p^2), \tag{4.77}$$

for this case as well.

**Bias-preserving circuits with infinite noise bias.** Suppose that the operation $U$ is bias-preserving, meaning that it does not convert phase-flip errors to bit-flip errors (we discuss bias-preserving operations at length in Section 5.3). More precisely, we suppose

$$UZ_iU^\dagger = P_Z, \tag{4.78}$$

for all $i$. Here, $Z_i$ is a phase-flip error acting on the $i$-th qubit, and $P_Z$ denotes an arbitrary linear combination of $n$-qubit Pauli operators from the set $\{I, Z\}^{\otimes n}$. We say that the channel $\mathcal{U}$ exhibits infinite noise bias if its Kraus operators can be expressed as

$$K_i = P_Z^{(i)}U, \tag{4.79}$$

where again $P_Z^{(i)}$ denotes an arbitrary linear combination of phase-flip errors. That is, $\mathcal{U}$ implements the ideal operation $U$ followed by some combination of phase-flip errors. For such channels, $P_{\text{fail}}$ can be bounded by a constant for the specific choice of $|\phi\rangle = |0\rangle^{\otimes n}$. The bound follows from the fact that,

$$\begin{aligned} K_i K_j |0\rangle^{\otimes n} &= (P_Z^{(i)} U)(P_Z^{(j)} U) |0\rangle^{\otimes n} \\ &= U^2 |0\rangle^{\otimes n} \\ &= (P_Z^{(j)} U)(P_Z^{(i)} U) |0\rangle^{\otimes n} \\ &= K_j K_i |0\rangle^{\otimes n}, \end{aligned} \tag{4.80}$$

where, to obtain the second line, we have used the fact that $U$ is bias-preserving together with the observation that $P_Z |0\rangle^{\otimes n} = |0\rangle^{\otimes n}$. Thus, we have

$$[K_i, K_j] |0\rangle^{\otimes n} = 0, \tag{4.81}$$

so the arguments of the previous two paragraphs may be directly applied to obtain

$$P_{\text{fail}} \leq 2p \left(1 - \frac{1}{M}\right) + O(p^2), \tag{4.82}$$

in this case as well.

### 4.2.3 Numerical demonstrations

To demonstrate our error-suppression scheme, we numerically simulate the circuit of Fig. 4.4 for several simple single-qubit channels. Results are shown in Fig. 4.5. In panel (a), we plot infidelity as a function of $M$, for bit-flip, phase-flip, and amplitude-damping channels. These channels are defined by the Kraus operators,

$$\text{bit flip} = \{\sqrt{1-p}I, \sqrt{p}X\} \tag{4.83}$$

$$\text{phase flip} = \{\sqrt{1-p}I, \sqrt{p}Z\} \tag{4.84}$$

$$\text{amplitude damping} = \{|0\rangle\langle 0| + \sqrt{1-p}\,|1\rangle\langle 1|, \sqrt{p}\,|0\rangle\langle 1|\}. \tag{4.85}$$

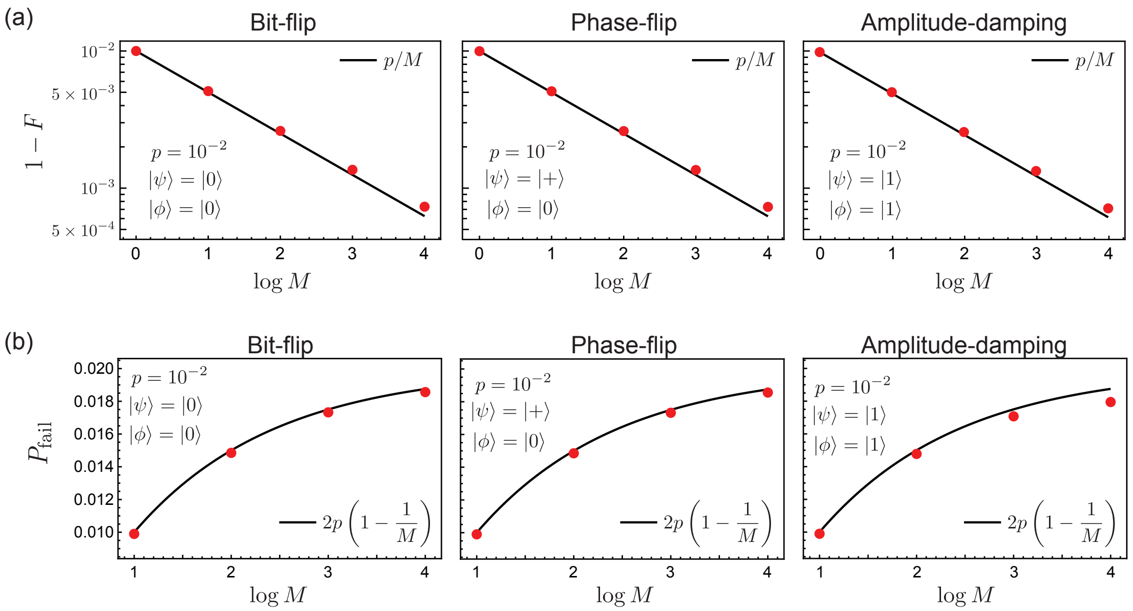


Figure 4.5: Infidelity (a) and failure probability (b) of the general error-suppression scheme for various noise channels. Red dots indicate exact numerical results (performed by enumerating all possible quantum trajectories), while solid lines correspond to the analytical expressions derived in Section 4.2.2. Note that $\log M = 0$ (equivalently, $M = 1$) corresponds to the case where no error suppression is used, for which $1 - F = p$ and $P_{\text{fail}} = 0$.

The infidelity is observed to scale as $p/M$ for all of these channels, which is to be expected because they all satisfy the criterion for maximal suppression, $\mathcal{C}_M = O(p^2)$. For the parameters in the plot, over an order-of-magnitude decrease in the infidelity is observed as $M$ is increased from $M = 1$ to $M = 16$. We note small deviations from the $p/M$ infidelity scaling are evident at the larger values of $M$. Such deviations are to be expected, as the infidelity is only suppressed to leading order. Ultimately, the infidelity is bounded from below by the next-order $O(p^2)$ contribution, which is $\sim 10^{-4}$ for the parameters shown in the plot.

In panel (b) of the figure, we plot the failure probability of the error-suppression scheme for the same three channels. In all cases, we observe that $P_{\text{fail}}$ closely follows the bound derived in Section 4.2.2, namely $P_{\text{fail}} \leq 2p(1 - 1/M)$. Indeed, the failure probability only increases mildly with $M$, approaching the constant value $2p$, even though the infidelity decreases by more than an order of magnitude. Recall that this favorable bound does not hold in general, but rather only in specific situations (e.g., when the Kraus operators commute, when $|\phi\rangle$ is stationary under the channel, or when the operation is bias-preserving). We have deliberately chosen two channels (bit-flip and phase-flip) with commuting Kraus operators in order to demonstrate this favorable error-probability scaling. The third channel, amplitude-damping, does not satisfy any of the criteria for constant failure probability identified in Section 4.2.2. Nevertheless, we observe that $P_{\text{fail}}$ seems to obey the same bound for amplitude damping as well. This observation demonstrates that our list of constant failure probability examples is not exhaustive, and that there are other cases of practical relevance with constant failure probability.

### 4.2.4 Comparison with error symmetrization

In Table 4.2, we summarize the analysis of our general error-suppression scheme (Fig. 4.4), and we compare our scheme with that of Ref. [117]. Both schemes can suppress the infidelity by a factor of $1/M$ to leading order, but the failure probabilities and hardware overheads differ substantially. While in both schemes the failure probability increases linearly with $M$ in general, there are special cases where the failure probability of our scheme satisfies $P_{\text{fail}} \leq 2p$. In such cases, the error suppression can be performed near-deterministically, provided the initial infidelity $p$ is small. The ability to perform near-deterministic error suppression is one significant advantage of our scheme.

Another significant advantage of our scheme is the exponential reduction in hardware overhead with respect to the parameter $M$. The scheme of Ref. [117] requires $M$ identical quantum computers all operating in parallel, plus an additional $O(M)$ ancillary qubits, so the total hardware overhead is $O(M)$. In contrast, our scheme requires only $\log M$ ancillary qubits, plus a constant number of additional qubits to hold the state $|\phi\rangle$. Thus, the total hardware overhead is only $O(\log M)$. This

| | Infidelity | Failure probability | Overhead |
|---|---|---|---|
| Ref. [117] | $p/M + O(p^2)$ | $(M-1)p + O(p^2)$ | $O(M)$ |
| Our scheme (Fig. 4.4) | $p/M + O(p^2)$ | $Mp + O(p^2)$ (general case)<br>$2p(1-1/M) + O(p^2)$ (special cases) | $O(\log M)$ |

Table 4.2: Comparison of our error-suppression scheme with that of Ref. [117]. Both schemes provide a $1/M$ suppression of the infidelity, but our scheme offers improvements in failure probability and overhead.

dramatic improvement in hardware efficiency makes our scheme suitable for use in near-term devices.

## 4.3 Hardware-efficient error suppression applied to QRAM

In this section, we demonstrate how our hardware-efficient error-suppression scheme can be applied to boost the query fidelity of QRAM. To do so, in Section 4.3.1 we begin by using the circuit-level noise model described in Chapter 3 to derive an effective channel describing a noisy QRAM query. Then, in Section 4.3.2 we verify that this effective channel satisfies the criterion for maximal suppression [Eq. (4.68)], from which it follows that the query infidelity is suppressed by a factor of $1/M$. Next, in Section 4.3.3 we demonstrate that the failure probability can be bounded by a constant in the case of QRAM, so that error suppression can be performed near deterministically. Finally, we verify these results through numerical simulations in Section 4.3.4.

### 4.3.1 Effective QRAM channel

We consider the situation where the target state is of the form

$$|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |0\rangle^B \,, \tag{4.86}$$

where $A$ and $B$ denote the address and bus registers. We wish to apply a data-lookup operation ($U = O_{\mathbf{x}}^{(\mathrm{DL})}$) to this state,

$$U \, |\psi\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |x_i\rangle^B \,. \tag{4.87}$$

Instead of the ideal operation $U$, we are given access to a noisy approximation of the operation, $\mathcal{U}$, that is physically implemented using a noisy bucket-brigade QRAM. In particular, we adopt the error model described in Chapter 3, where each router of the QRAM is subject to some mixed-unitary error channel.

In order to study the applicability of our error-suppression scheme to QRAM, we must first translate the circuit-level noise model of Chapter 3 into an effective channel that acts only on the address and bus registers. That is, we must calculate an explicit Kraus-operator representation for the channel $\mathcal{U}$. Recall that the final state $\Omega$ of the full system (address, bus, and routers) after a QRAM query can be written as

$$\Omega = \sum_c p(c) \, |\Omega(c)\rangle \, \langle \Omega(c)| \,, \tag{4.88}$$

where $c$ indexes different error configurations. Here,

$$|\Omega(c)\rangle = |\mathrm{good}(c)\rangle + |\mathrm{bad}(c)\rangle \,, \tag{4.89}$$

where

$$\left|\mathrm{good}(c)\right\rangle = \left(\sum_{i\in g(c)} \alpha_i \left|i\right\rangle^A \left|x_i\right\rangle^B\right) \left|f(c)\right\rangle^R, \tag{4.90}$$

and $\left|\mathrm{bad}(c)\right\rangle$ denotes the state of the full system with respect to the "bad" branches (branches $i \notin g(c)$, see Chapter 3 for details). The effective channel acting on only the address-bus system is obtained by tracing out the routers,

$$\mathcal{U}(\psi) = \mathrm{Tr}_R\left[\Omega\right] = \sum_c p(c)\mathrm{Tr}_R\left[\left|\Omega(c)\right\rangle\left\langle\Omega(c)\right|\right] \tag{4.91}$$

To proceed, it is convenient to introduce an orthonormal basis for the routers' Hilbert space. We denote elements of this basis as $\left|R_l(c)\right\rangle$, where $l$ indexes the different basis states, and we define $\left|R_0(c)\right\rangle \equiv \left|f(c)\right\rangle$. We have

$$\begin{aligned}\mathcal{U}(\psi) &= \sum_c p(c)\left[\left\langle f(c)|\Omega(c)\right\rangle\left\langle\Omega(c)|f(c)\right\rangle + \sum_{l>0}\left\langle R_l(c)|\Omega(c)\right\rangle\left\langle\Omega(c)|R_l(c)\right\rangle\right]\\ &= \sum_c p(c)\left[\left(\sum_{i\in g(c)} \alpha_i \left|i\right\rangle^A \left|x_i\right\rangle^B + \left\langle f(c)|\mathrm{bad}(c)\right\rangle\right)\left(\mathrm{H.c.}\right)\right.\\ &\quad \left. + \sum_{l>0}\left\langle R_l(c)|\mathrm{bad}(c)\right\rangle\left\langle\mathrm{bad}(c)|R_l(c)\right\rangle\right]\end{aligned} \tag{4.92}$$

This channel can be equivalently written as,

$$\mathcal{U}(\psi) = \sum_{c,l} K(c,l)\left|\psi\right\rangle\left\langle\psi\right| K(c,l)^\dagger, \tag{4.93}$$

where the operators $K(c,l)$ constitute a Kraus representation of the channel $\mathcal{U}$ and act as

$$K(c,l)\sum_{i\in g(c)} \alpha_i \left|i\right\rangle^A \left|0\right\rangle^B = \begin{cases}\sqrt{p(c)}\sum_{i\in g(c)} \alpha_i \left|i\right\rangle^A \left|x_i\right\rangle^B, & \text{for } l=0\\ 0, & \text{for } l\neq 0\end{cases} \tag{4.94}$$

and

$$K(c,l)\sum_{i\notin g(c)}\alpha_i\left|i\right\rangle^A\left|0\right\rangle^B=\sqrt{p(c)}\left|R_l(c)|\mathrm{bad}(c)\right\rangle, \tag{4.95}$$

so that

$$K(c,0)\left|\psi\right\rangle=\sqrt{p(c)}\left[\sum_{i\in g(c)}\alpha_i\left|i\right\rangle^A\left|x_i\right\rangle^B+\left|f(c)|\mathrm{bad}(c)\right\rangle\right] \tag{4.96}$$

$$K(c,l>0)\left|\psi\right\rangle=\sqrt{p(c)}\left|R_l(c)|\mathrm{bad}(c)\right\rangle. \tag{4.97}$$

Physically, $K(c,l)$ corresponds to performing a noisy query with error configuration $c$, projecting the routers onto the state $\left|R_l(c)\right\rangle$, then discarding the routers.

Thus far, we have calculated the action of the Kraus operators $K(c,l)$ on the state $\left|\psi\right\rangle$. To analyze our error-suppression scheme, we also need to specify another state $\left|\phi\right\rangle$ (of the same dimension as $\left|\psi\right\rangle$), and calculate the action of the Kraus operators on this state. Recall that the choice of $\left|\phi\right\rangle$ can affect both the infidelity and the failure probability. To minimize both of these quantities, it is prudent to choose a state $\left|\phi\right\rangle=\left|\psi^\perp\right\rangle$ that is orthogonal to $\left|\psi\right\rangle$. With such a choice for $\left|\phi\right\rangle$, the action of the Kraus operators on this state can satisfy,

$$K(c,0)\left|\phi\right\rangle=\sqrt{p(c)}\left|\phi\right\rangle \tag{4.98}$$

$$K(c,l>0)\left|\phi\right\rangle=0. \tag{4.99}$$

That the Kraus operators act on $\left|\phi\right\rangle$ in this way is not immediately obvious and remains to be justified. We provide further exposition below. Before doing so, however, we remark that our choice of a state $\left|\phi\right\rangle$ satisfying Eqs. (4.98) and (4.99) is crucial to the minimization of both the infidelity and failure probability.

When $\left|\phi\right\rangle$ is orthogonal to $\left|\psi\right\rangle$, these two states can be distinguished, and this property can be exploited to ensure that Eqs. (4.98) and (4.99) hold. The idea is to

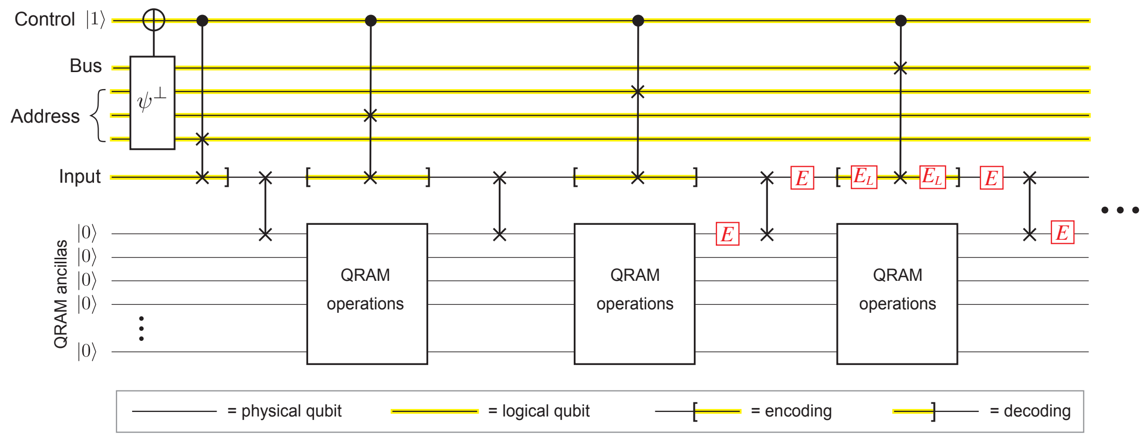


Figure 4.6: Queries with constrained error propagation. The circuit illustrates how the address and bus qubits can be injected into the QRAM when the input state is $|\psi\rangle$, without allowing errors to propagate back when the input state is $|\phi\rangle = |\psi^{\perp}\rangle$. When the address and bus registers are initialized in $|\psi^{\perp}\rangle$, the first gate in the circuit flips the control qubit from $|1\rangle$ to $|0\rangle$. All of the controlled-SWAP gates in the circuit then act trivially, so errors from the QRAM cannot propagate back to the address and bus registers. An example error and its subsequent propagation are illustrated by the red boxes labelled $E$. The address and bus registers can be error corrected in order ensure that they are not themselves subject to errors. Errors from the QRAM can then propagate to logical errors (denoted $E_L$) on the "input" rail, but these logical errors do not propagate to the other logical qubits provided the controlled-SWAP gates are implemented fault tolerantly.

first check whether the QRAM is being queried with the state $|\psi\rangle$ or $|\phi\rangle$, then only inject the address and bus qubits into the tree if the input state is $|\psi\rangle$ and not $|\phi\rangle$. The procedure is illustrated in Fig. 4.6, which is a slightly modified version of the usual bucket-brigade QRAM circuit (Fig. 2.8). The first gate of this circuit checks whether the input state is $|\phi\rangle = |\psi^{\perp}\rangle$, and if so a control qubit flipped from $|1\rangle$ to $|0\rangle$. Then, the injection of the address and bus qubits into the QRAM is coherently conditioned on the state of the control qubit. This way, the address and bus qubits are *not* sent into the tree if the input state is $|\phi\rangle$. As illustrated in the figure, it follows that errors occurring within the QRAM cannot propagate back to the address or bus registers when the QRAM is queried with $|\phi\rangle$. This is another example of the *constrained error propagation* that was crucial to our proof of QRAM's noise resilience in Chapter 3.

In this context, the constrained error propagation allows us to justify Eqs. (4.98) and (4.99). When the QRAM is queried with $|\phi\rangle$, the address and bus qubits are not routed into the tree, so the query acts trivially. Moreover, because errors cannot propagate back to the address and bus registers, it follows that $|\phi\rangle$ is invariant under the channel,

$$K(c,l)\,|\phi\rangle \propto |\phi\rangle\,,\ \text{for all } c,\ l. \tag{4.100}$$

Furthermore, because no qubits are routed into the tree, all routers remain in the wait state in the absence of errors. As discussed in Chapter 3, errors propagate identically through inactive routers (those in the wait state), and active routers that lie in an error-free branch. Therefore, the final state of the routers is the same whether the QRAM was queried with $|\phi\rangle$ or with a good address (i.e., an address $|i\rangle^A$ with $i \in g(c)$). The final state of the routers is $|f(c)\rangle$ in either case. Since $K(c,0)$ corresponds to this final state, while $K(c,l>0)$ correspond to other final sates, Eqs. (4.98) and (4.99) immediately follow.

Our justification of Eqs. (4.98) and (4.99) assumes that no errors occur in the address and bus registers themselves. This assumption can be justified approximately if the physical qubits comprising these register have a much lower error rate than those comprising the QRAM. Alternatively, as illustrated in Fig. 4.6, errors in the address and bus registers can be suppressed using error correction, while the QRAM itself can be implemented without error correction. Repeated encoding and decoding is then required to convert between logical qubits and physical qubits, but the error propagation is sufficiently constrained regardless.

We have now specified a Kraus decomposition for the noisy QRAM channel [Eq. (4.93)], and we have computed how these Kraus operators act on the relevant states $|\psi\rangle$ [Eqs. (4.96) and (4.97)] and $|\phi\rangle$ [Eqs. (4.98) and (4.99)]. With these results, we proceed to show that the criterion of maximal suppression is satisfied (Section 4.3.2) and that the failure probability can be bounded by a constant (Section 4.3.3).

### 4.3.2 Infidelity of QRAM with error suppression

As shown in Section 4.2.2, the infidelity of our error-suppression scheme when applied to a channel $\mathcal{U}$ is

$$1 - F = \frac{p}{M} + \mathcal{C}_M + O(p^2), \tag{4.101}$$

where $p$ denotes the infidelity of the channel. In the case of the bucket-brigade QRAM, we showed in Chapter 3 that

$$p = \varepsilon \, \text{polylog}(N), \tag{4.102}$$

where $\varepsilon$ denotes the physical error rate of the quantum routers, and $N$ denotes the size of the memory. Thus, when error suppression is applied to a QRAM query, the infidelity scales as

$$1 - F = \frac{\varepsilon}{M} \, \text{polylog}(N) + O[\varepsilon^2 \text{polylog}(N)], \tag{4.103}$$

provided that the criterion for maximal suppression, $\mathcal{C}_M = O(p^2)$, is satisfied. In the remainder of this section, we prove that this criterion is satisfied.

Recall from Section 4.2.2 that $\mathcal{C}_M = \mathcal{C}_2$ when $|\phi\rangle$ is invariant under the channel $\mathcal{U}$. This is the case for QRAM, since $K(c,l)\,|\phi\rangle \propto |\phi\rangle$ for all $c, l$. Thus, it remains to show that

$$\mathcal{C}_2 = \frac{1}{2} \sum_{c,c'} \sum_{l,l'} \langle\psi|K(c',l')^\dagger \Pi^\perp_{U|\psi\rangle} K(c,l)|\psi\rangle \, \langle\phi|K(c,l)^\dagger K(c',l')|\phi\rangle = O(p^2). \tag{4.104}$$

It follows from Eqs. (4.98) and (4.99) that

$$\langle\phi|K(c,l)^\dagger K(c',l')|\phi\rangle = \sqrt{p(c)p(c')}\delta_{l,0}\delta_{l',0}, \tag{4.105}$$

so

$$\mathcal{C}_2 = \frac{1}{2}\sum_{c,c'} \sqrt{p(c)p(c')}\, \langle\psi|K(c',0)^{\dagger}\Pi^{\perp}_{U|\psi\rangle}K(c,0)|\psi\rangle\,. \tag{4.106}$$

To proceed, we introduce some convenient shorthand notation,

$$|\Psi\rangle \equiv U\,|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i\,|i\rangle\,|x_i\rangle \tag{4.107}$$

$$|g_c\rangle \equiv \sum_{i\in g(c)} \alpha_i\,|i\rangle\,|x_i\rangle\,, \tag{4.108}$$

$$|\bar{g}_c\rangle \equiv \sum_{i\notin g(c)} \alpha_i\,|i\rangle\,|x_i\rangle\,, \tag{4.109}$$

$$|\gamma_c\rangle \equiv \langle f(c)|\mathrm{bad}(c)\rangle\,. \tag{4.110}$$

From these definitions we have the following useful relations,

$$|\Psi\rangle = |g_c\rangle + |\bar{g}_c\rangle\,, \tag{4.111}$$

and

$$\langle g_c|\bar{g}_c\rangle = 0, \tag{4.112}$$

and

$$K(c,0)\,|\psi\rangle = \sqrt{p(c)}(|g_c\rangle + |\gamma_c\rangle). \tag{4.113}$$

Additionally, we note that

$$\langle\gamma_c|\,(|i\rangle^A\,|x_i\rangle^B) = 0, \quad \text{for all } i\in g(c). \tag{4.114}$$

This last statement follows from the fact that we assume the components of the QRAM are subject to mixed-unitary error channels, which preserve orthogonality.

Inserting these definitions into Eq. (4.106) yields

$$\begin{aligned}\mathcal{C}_2 &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left(\langle g_{c'}| + \langle \gamma_{c'}|\right) \Pi^{\perp}_{|\Psi\rangle} \left(|g_c\rangle + |\gamma_c\rangle\right) \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \Bigg[ \langle g_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle + \langle \gamma_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle + \langle g_{c'}|\Pi^{\perp}_{|\Psi\rangle}|\gamma_c\rangle + \langle \gamma_{c'}|\Pi^{\perp}_{|\Psi\rangle}|\gamma_c\rangle \Bigg].\end{aligned} \tag{4.115}$$

We proceed to show that each of the four terms in the above expression is $O(p^2)$. For easy reference, we define

$$\text{Term}_1 \equiv \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle g_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle \tag{4.116}$$

$$\text{Term}_2 \equiv \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle \tag{4.117}$$

$$\text{Term}_3 \equiv \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle g_{c'}|\Pi^{\perp}_{|\Psi\rangle}|\gamma_c\rangle \tag{4.118}$$

$$\text{Term}_4 \equiv \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|\Pi^{\perp}_{|\Psi\rangle}|\gamma_c\rangle . \tag{4.119}$$

We being with the first term,

$$\begin{aligned}\text{Term}_1 &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle g_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left(\langle g_{c'}|g_c\rangle - \langle g_{c'}|\Psi\rangle \langle \Psi|g_c\rangle\right) \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left(\langle g_{c'}|g_c\rangle - \langle g_{c'}|g_{c'}\rangle \langle g_c|g_c\rangle\right),\end{aligned} \tag{4.120}$$

where to obtain the last line we have used Eqs. (4.111) and (4.112). Then, using the definition of $|g_c\rangle$,

$$\begin{aligned}
\text{Term}_1 &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left[ \sum_{i\in g(c)\cap g(c')} |\alpha_i|^2 - \sum_{i\in g(c)} |\alpha_i|^2 \sum_{i\in g(c')} |\alpha_i|^2 \right] \\
&= \frac{1}{2}\sum_{c,c'} \sum_{i\in g(c)\cap g(c')} p(c)p(c')|\alpha_i|^2 - \frac{1}{2}\left[\sum_{c}\sum_{i\in g(c)} p(c)|\alpha_i|^2\right]\left[\sum_{c'}\sum_{i\in g(c')} p(c')|\alpha_i|^2\right] \\
&= \frac{1}{2}\sum_{c,c'} \sum_{i\in g(c)\cap g(c')} p(c)p(c')|\alpha_i|^2 - \frac{1}{2}\left[\sum_{c}\sum_{i\in g(c)} p(c)|\alpha_i|^2\right]^2 . \qquad (4.121)
\end{aligned}$$

The term in brackets on the last line can be simplified using the results from Chapter 3. Recall that $\Lambda(c) = \sum_{i\in g(c)} |\alpha_i|^2$ denotes the fraction of good branches, and that the expected fraction of good branches is

$$\sum_c p(c)\Lambda(c) = 1 - p \qquad (4.122)$$

The other contribution to $\text{Term}_1$ can be simplified by introducing the function

$$I(i,c) \equiv \begin{cases} 1, & i \in g(c), \\ 0, & i \notin g(c). \end{cases} \qquad (4.123)$$

Using this function,

$$\begin{aligned}
\sum_{c,c'} \sum_{i\in g(c)\cap g(c')} p(c)p(c')|\alpha_i|^2 &= \sum_{i=0}^{N-1} |\alpha_i|^2 \sum_{c,c'} p(c)p(c')I(i,c)I(i,c') \\
&= \sum_{i=0}^{N-1} |\alpha_i|^2 \left(\sum_c p(c)I(i,c)\right)\left(\sum_{c'} p(c')I(i,c')\right) \\
&= \sum_{i=0}^{N-1} |\alpha_i|^2 \left(\sum_c p(c)I(i,c)\right)^2 \\
&= (1-p)^2. \qquad (4.124)
\end{aligned}$$

To obtain the last line, note that $\sum_c p(c) I(i,c)$ is the probability that $i \in g(c)$, averaged over all error configurations. This is simply the expected fraction of good branches, $(1-p)$. Thus, we have

$$\begin{aligned}\text{Term}_1 &= \frac{1}{2}\sum_{c,c'}\sum_{i\in g(c)\cap g(c')} p(c)p(c')|\alpha_i|^2 - \frac{1}{2}\left[\sum_c \sum_{i\in g(c)} p(c)|\alpha_i|^2\right]^2 \\ &= \frac{1}{2}(1-p)^2 - \frac{1}{2}(1-p)^2 = 0. \end{aligned} \tag{4.125}$$

Next, we consider Term$_2$ and Term$_3$. Note that these two terms are actually equivalent to one another, so it suffices to consider only one of them,

$$\begin{aligned}\text{Term}_2 &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|\Pi^{\perp}_{|\Psi\rangle}|g_c\rangle \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left[\langle \gamma_{c'}|g_c\rangle - \langle \gamma_{c'}|\Psi\rangle \langle \Psi|g_c\rangle\right] \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left[\langle \gamma_{c'}|g_c\rangle - \left(\langle \gamma_{c'}|g_c\rangle + \langle \gamma_{c'}|b_c\rangle\right) \langle g_c|g_c\rangle\right]. \end{aligned} \tag{4.126}$$

To proceed, we use the result from the previous paragraph that $\langle g_c|g_c\rangle = 1-p$,

$$\begin{aligned}\text{Term}_2 &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left[\langle \gamma_{c'}|g_c\rangle - \left(\langle \gamma_{c'}|g_c\rangle + \langle \gamma_{c'}|b_c\rangle\right)(1-p)\right] \\ &= \frac{1}{2}\sum_{c,c'} p(c)p(c') \left[p \langle \gamma_{c'}|g_c\rangle + \langle \gamma_{c'}|b_c\rangle (1-p)\right]. \end{aligned} \tag{4.127}$$

Now, one can show that

$$\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|g_c\rangle = O(p) \tag{4.128}$$

$$\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|b_c\rangle = O(p^2). \tag{4.129}$$

For brevity, we only sketch the proof of the first statement; the second follows from a similar calculation. We have

$$\begin{aligned}\left|\sum_{c,c'} p(c)p(c') \langle \gamma_{c'}|g_c\rangle\right| &= \left|\sum_{c,c'} p(c)p(c') \sum_{i\in g(c),\notin g(c')} \alpha_i \langle \gamma_{c'}| (|i\rangle |x_i\rangle)\right| \\ &\leq \sum_{c,c'} p(c)p(c') \left|\sum_{i\in g(c),\notin g(c')} \alpha_i \langle \gamma_{c'}| (|i\rangle |x_i\rangle)\right| \\ &\leq \sum_{c,c'} p(c)p(c') \left[\langle \gamma_{c'}|\gamma_{c'}\rangle \sum_{i\in g(c),\notin g(c')} |\alpha_i|^2\right]^{1/2}, \end{aligned} \tag{4.130}$$

where we have used Eq. (4.114) to obtain the first line, the triangle inequality to obtain the second, and the Cauchy-Schwarz inequality to obtain the third. Continuing,

$$\begin{aligned} &\leq \sum_{c,c'} p(c)p(c') \left[\sum_{j\notin g(c')} |\alpha_j|^2 \sum_{i\in g(c),\notin g(c')} |\alpha_i|^2\right]^{1/2} \\ &\leq \sum_{c,c'} p(c)p(c') \sum_{i\notin g(c')} |\alpha_i|^2 \\ &\leq \sum_i |\alpha_i|^2 \sum_{c'} p(c')[1 - I(i,c')] = p, \end{aligned} \tag{4.131}$$

where we have used $\sum_c p(c)I(i,c) = 1-p$ to obtain the final equality. This concludes the proof of Eq. (4.128). The proof of Eq. (4.129) is similar. Inserting Eqs. (4.128) and (4.129) into Eq. (4.127) yields the desired result,

$$\text{Term}_2 = \text{Term}_3 = O(p^2). \tag{4.132}$$

No new conceptual insights are required for the calculation of $\text{Term}_4$, so we omit this calculation for brevity. The result is similarly that $\text{Term}_4 = O(p^2)$.

Combining these results together, we have

$$\mathcal{C}_2 = \text{Term}_1 + \text{Term}_2 + \text{Term}_3 + \text{Term}_4 = O(p^2), \tag{4.133}$$

so we see that QRAM does indeed satisfy the criterion for maximal error suppression. We therefore have

$$\begin{aligned}1-F &= \frac{p}{M}+\mathcal{C}_M+O(p^2)=\frac{p}{M}+O(p^2)\\ &= \frac{\varepsilon}{M}\text{polylog}(N)+O[\varepsilon^2\text{polylog}(N)],\end{aligned} \tag{4.134}$$

and we see that our error-suppression scheme suppresses the query infidelity by a factor of $1/M$, to leading order.

### 4.3.3 Failure probability of QRAM error suppression

In this section, we compute the failure probability $P_{\text{fail}}$ of QRAM error suppression. We find that

$$P_{\text{fail}}=O(p), \tag{4.135}$$

is independent of $M$. Therefore, near-deterministic error suppression is possible.

The failure probability can be computed from Eq. (4.70),

$$P_{\text{fail}}=1-\frac{1}{M^2}\sum_{c_0,\dots c_{M-1}}\sum_{l_0,\dots l_{M-1}}\sum_{a,b=0}^{M-1}\langle\psi|K(c_b,l_b)^\dagger K(c_a,l_a)|\psi\rangle\,\langle\phi|\overline{K}(c_b,l_b)^\dagger\overline{K}(c_a,l_a)|\phi\rangle\,. \tag{4.136}$$

We can exploit Eqs. (4.98) and (4.99) to simplify this expression,

$$\begin{aligned}P_{\text{fail}} &= 1-\frac{1}{M^2}\sum_{a,b=0}^{M-1}\sum_{c_a,c_b}\sqrt{p(c_a)p(c_b)}\,\langle\psi|K(c_b,0)^\dagger K(c_a,0)|\psi\rangle\\ &= 1-\sum_{c_a,c_b}\sqrt{p(c_a)p(c_b)}\,\langle\psi|K(c_b,0)^\dagger K(c_a,0)|\psi\rangle\,.\end{aligned} \tag{4.137}$$

Note that this expression is already independent of $M$.

It remains to show that $P_{\text{fail}} = O(p)$. Recall from the previous section that

$$K(c,0)\left|\psi\right\rangle = \sqrt{p(c)}(\left|g_c\right\rangle + \left|\gamma_c\right\rangle). \tag{4.138}$$

Inserting this expression into Eq. (4.137) yields,

$$P_{\text{fail}} = 1 - \sum_{c_a,c_b} p(c_a)p(c_b)\left(\left\langle g_{c_b}|g_{c_a}\right\rangle + \left\langle \gamma_{c_b}|g_{c_a}\right\rangle + \left\langle g_{c_b}|\gamma_{c_a}\right\rangle + \left\langle \gamma_{c_b}|\gamma_{c_a}\right\rangle\right). \tag{4.139}$$

From the analysis in the previous section, we know that only the first term in the sum yields a contribution which is $O(1)$; the contributions from all the other terms are $O(p)$. Thus,

$$P_{\text{fail}} = 1 - \sum_{c_a,c_b} p(c_a)p(c_b)\left\langle g_{c_a}|g_{c_b}\right\rangle + O(p). \tag{4.140}$$

Note the sum in the above equation is equivalent to that in Eq. (4.124), which we calculated to be $(1-p)^2$. Inserting this expression yields the desired result,

$$P_{\text{fail}} = 1 - (1-p)^2 + O(p) = O(p). \tag{4.141}$$

### 4.3.4 Numerical demonstrations

We numerically simulate the application of our error suppression scheme to QRAM, with the QRAM subject to the the circuit-level noise model described in Chapter 3. To do so, we adapt the efficient classical simulation algorithm described in that chapter. Our simulation proceeds by first sampling from the set of possible error configurations at each of the $M$ rounds, then tracking the evolution of different computational basis states through the noisy circuit (see Chapter 3 for further details). The simulation cost scales polynomially in both $N$ and $M$.

Simulation results are shown in Fig. 4.7. We simulate a bucket-brigade QRAM with $N = 8$ memory locations, where each router is subject to either bit-flip, phase-

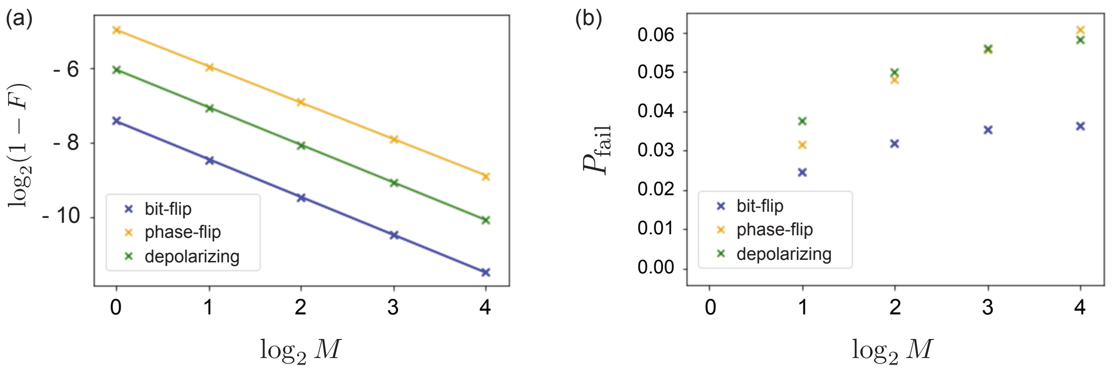


Figure 4.7: Error suppression applied to QRAM. (a) Query infidelity. We plot $\log_2(1-F)$ as a function of $\log_2(M)$, where $F$ denotes effective QRAM query fidelity obtained via error suppression. The solid lines indicate linear fits, and the fitted slopes of -1.00, -0.97, -1.00 demonstrating good agreement with the expected $1/M$ suppression. (b) Failure probability. The failure probabilities for all channels appear to approach constants, consistent with the expectation that $P_{\mathrm{fail}} = O(p)$ independent of $M$.

flip, or depolarizing errors at a rate of $\varepsilon = 0.001$. For these parameters, the infidelity of a single query ranges from $p \sim 1\%$ - 5%, depending on error channel and the data being queried. With error suppression, we observe over an order-of-magnitude decrease in the query infidelity as $M$ is increased from 0 to 16, in good agreement with the expected $1/M$ scaling. We also calculate the failure probability, and the results are consistent with the expectation that $P_{\mathrm{fail}}$ approaches a constant of order $p$. These results demonstrate that the QRAM query infidelity can be significantly suppressed in a hardware-efficient and near-deterministic manner.

## 4.4 Conclusions and Outlook

In this chapter, we have proposed a hardware-efficient error-suppression scheme that can reduce the infidelity of quantum operations. Our scheme uses $M$ applications of a channel to distill an effective channel whose infidelity is reduced by a factor of $1/M$. This scheme is hardware efficient, with the required hardware overhead scaling only logarithmically with $M$. Moreover, in several situations of practical interest, the failure probability can be shown to be independent of $M$, so that error suppression can be performed near deterministically. The hardware efficiency and near determinism of our scheme not only constitute significant improvements over the error-symmetrization scheme of Ref. [117], but they are also important practical benefits that make our scheme applicable to near-term devices. Indeed, our scheme is best suited for noisy intermediate-scale devices, where both the number and quality of qubits are limited. In this context, the ability to achieve even a quadratic suppression of the infidelity [for $M = 1/p$ we have $p/M = O(p^2)$] could be extremely useful.

As an example of a practical use case, we have described the application of our scheme to QRAM. While prohibitive overheads make large-scale, error-corrected QRAM impractical (at least with conventional error-correction approaches), our error-suppression scheme allows one to suppress the query infidelity without an additional $O(N)$ hardware overhead. Moreover, we have shown that this suppression succeeds with probability $1 - O(p)$, where $p$ is the query infidelity. Thus, if the query infidelity is already low (due to the noise resilience of the bucket-brigade architecture, for example), then the error suppression can be performed near deterministically. As a result, our scheme is suitable for use in algorithms that involve many QRAM queries. For an algorithm with $Q$ queries, the total success probability, $1 - O(Qp)$, is of order unity so long as $Q \ll 1/p$. If the base query infidelity is low, $p \ll 1$, then the number of queries can be large. On the other hand, one downside of our scheme is that the required time overhead for error suppression is proportional to $M$. Because QRAM queries are already fast [$T = O(\log N)$], however, this additional time overhead may not be problematic in many situations.

One aspect of our scheme that requires further analysis is the effect of errors in other parts of the circuit. We have assumed that these errors occur with some rate $p'$ that is sufficiently small so that these effects are negligible. To obtain a pessimistic estimate of these effects, we can assume that any such error reduces the fidelity of the final state to 0. Because there are $O(M)$ different locations for such errors, a pessimistic estimate for the infidelity is

$$1 - F = \frac{p}{M} + AMp' + O(p^2) + O(p'^2), \tag{4.142}$$

where $A$ is some constant. The optimal choice of $M$ is thus

$$M = \min\left\{(p/Ap')^{1/2}, O(1/M)\right\}, \tag{4.143}$$

and the minimal infidelity is

$$1 - F = \min\left\{2(Ap'p)^{1/2}, O(p^2)\right\}. \tag{4.144}$$

An estimate of the parameter $A$ is thus required to obtain a lower bound on the achievable infidelity. While an upper bound on $A$ can easily be obtained simply by enumerating all possible error locations, it is unlikely that such a bound would be tight. For example, many of the possible errors could be detected by the postselection, so that they would increase the failure probability rather than the infidelity. A precise estimate of $A$ is the subject of ongoing work.

Another important direction for future work will be to develop a procedure for determining which QRAM architecture minimizes the query infidelity as a function of the available resources. That is, suppose that a particular application requires that the query time must be less than $T_{\max}$, and that no more than $N_{\max}$ qubits can be used to perform the query. What architecture (QRAM, QROM, or hybrid) and what means of error reduction (error suppression, quantum error correction, or both) should be employed to minimize the query infidelity subject to these constraints? For example, for certain $N_{\max}$ and $T_{\max}$, it may be possible to use QRAM with error suppression, QROM with error correction, or even a hybrid QRAM-QROM architecture that employs both error suppression and error correction. Understanding this optimization landscape is necessary if we are to fully exploit the limited resources of near- and intermediate-term quantum devices.

# Chapter 5

# Quantum acoustic implementations of QRAM

In the preceding chapters, we have shown that QRAM can be remarkably resilient to noise, and we have presented a hardware-efficient scheme to further suppress errors in QRAM queries. Now, in this chapter, we turn to the question of how a large-scale QRAM can be constructed. Of course, QRAM's only function is to implement the unitary operation $O_{\mathbf{x}}^{(\mathrm{DL})}$, and in principle any universal quantum computer could fulfill this function. However, because QRAM is a highly-specialized architecture with a very specific purpose, using a general-purpose quantum computer to emulate QRAM is highly inefficient. For example, implementations of error-corrected QRAM using the surface code—the code most commonly considered for universal fault-tolerant quantum computing—incur massive overheads that make scaling unfeasible [18].

A more prudent approach to building a QRAM is to specifically tailor the underlying hardware to the task at hand. In this spirit, in this chapter we propose implementations of QRAM based on hybrid quantum acoustic hardware. As we show below, quantum acoustic systems are naturally well-suited to the task of implementing QRAM due to their compactness and high coherence. Because of the small size of the acoustic components, quantum acoustic devices are highly scalable. At the same time, the highly coherent nature of acoustic modes minimizes the amount of error correction that is required to realize high-fidelity queries. We leverage these appealing properties to propose experimental implementations of QRAM that are hardware-efficient and scalable.

This chapter is organized as follows. In Section 5.1 we review the recent experimental progress in the field of quantum acoustics that motivates our proposals. Next, in Sections 5.2 and 5.3, we present two schemes for quantum computing with quantum acoustics. The first scheme, in Section 5.2, is based on Hamiltonian engineering in multimode acoustic systems, and the relative simplicity of this approach makes it better suited for near-term experiments. The second scheme, in Section 5.3, is based on stabilized cat qubits, and in the long term this scheme presents a viable path toward hardware-efficient fault-tolerant quantum computing. Finally, in Section 5.4 we describe how both both schemes can be used to realize a modular, hardware-efficient QRAM.

The results in this chapter are primarily based on Ref. [99]: CTH et al., Hardware-efficient quantum random access memory with hybrid quantum acoustic systems, Phys. Rev. Lett. 123, 250501 (2019), and Ref. [125]: Chamberland et al. (including CTH), Building a fault-tolerant quantum computer using concatenated cat codes, arXiv:2012.04108.

## 5.1 Recent experimental progress in quantum acoustics

The coupling of superconducting qubits to microwave resonators, termed circuit quantum electrodynamics (cQED) [134, 135], constitutes one of today's most promising quantum computing architectures. Microwave modes provide good quantum memories [136], while superconducting nonlinearities enable the initialization [137], manipulation [138, 139], readout [140], and protection [141, 142] of quantum states encoded in microwave photons. However, long microwave wavelengths pose a potential limitation to the scalability of cQED systems. On-chip resonators face trade-offs between compactness and quality factor [143, 144], and microwave modes with millisecond coherence or better have thus far only been demonstrated in large 3D cavities [136, 145].

Recently, coherent couplings between superconducting qubits and acoustic resonators have been demonstrated in a remarkable series of experiments [146–158]. These so-called circuit quantum acoustodynamic (cQAD) systems (Fig. 5.1) possess many of the advantageous properties of cQED systems, e.g., superconducting qubits can be used to generate arbitrary superpositions of acoustic Fock states [150, 154], and phonon-number resolving measurements can be performed in the dispersive regime [157, 158].

Relative to electromagnetic modes, acoustic modes can provide dramatic benefits in terms of size and coherence times. The velocities of light and sound differ by five orders of magnitude, and the correspondingly short acoustic wavelengths enable the fabrication of ultra-compact phononic resonators [159]. Furthermore, acoustic modes can be exceptionally well-isolated from their environments—quality factors in excess of $10^{10}$ were recently demonstrated in GHz frequency phononic crystal resonators [160]. A variety of applications for such platforms have been proposed, including quantum transduction [161], entanglement generation [162, 163], and quantum signal processing [164, 165]. Only recently has the direct use of cQAD systems for quantum computing started to receive attention [99, 125, 166].

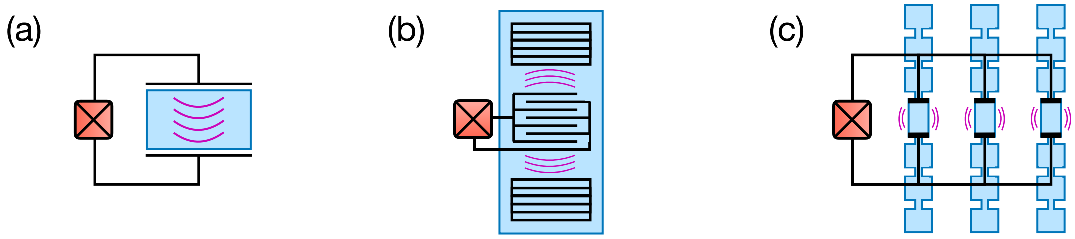


Figure 5.1: Multimode cQAD. A nonlinear superconducting circuit (red) is piezoelectrically coupled to (a) a bulk acoustic wave resonator, (b) a surface acoustic wave resonator, or (c) an array of phononic crystal resonators.

## 5.2 Quantum computing with acoustics, approach 1: multimode Hamiltonian engineering

In this section, we propose a hardware-efficient and scalable quantum computing architecture for multimode cQAD systems. Quantum information is stored in high-quality acoustic modes, and interactions between modes are engineered by applying off-resonant drives to an ancillary superconducting transmon qubit. During these operations, the transmon is only virtually excited, so the effects of transmon decoherence are mitigated. This is a crucial property, since the transmon's decoherence rate can exceed that of the phonons by orders of magnitude. In comparison to existing proposals that involve directly exciting the transmon [103, 166], this virtual approach can offer substantial improvement in gate fidelity for long-lived phonons. This scheme is also directly applicable to multimode cQED [103].

In Section 5.2.1 we provide a broad overview of our scheme, and in the sections that follow, we explore several practical aspects the scheme in further detail. In Section 5.2.2, we describe how inter-mode couplings can be selectively engineered in situations where the phonon mode frequencies are naturally evenly spaced. In Section 5.2.3 we derive the expressions for the coupling rates and verify their accuracy with numerical simulations. And finally in Section 5.2.4 we leverage these results to estimate achievable gate fidelities.

### 5.2.1 Hamiltonian engineering in multimode cQAD

In multimode cQAD, a transmon qubit (or some other superconducting circuit) is piezoelectrically coupled to a collection of acoustic modes. These modes can be supported in bulk acoustic wave (BAW) [149–151] or surface acoustic wave (SAW) [152–157] resonators, or in an array of phononic crystal (PC) resonators [158] (Fig. 5.1). Quality factors of $\approx 10^5$, $10^8$, and $10^{10}$ have been measured at GHz frequencies in SAW [167, 168], BAW [169, 170], and PC resonators [160], respectively, and the transmon can be simultaneously coupled to large numbers of high-Q modes on a single chip, even hundreds at once [149]. These systems can be described by the Hamiltonian

$$H = \omega_q b^\dagger b - \frac{\alpha}{2} b^\dagger b^\dagger b b + \sum_k \left( \omega_k a_k^\dagger a_k + g_k b^\dagger a_k + g_k^* b a_k^\dagger \right) + H_d, \tag{5.1}$$

where we take $\hbar = 1$ throughout this chapter to simplify notation. Here, $b$ and $a_k$ denote the annihilation operators for the transmon and phonon modes, respectively. The transmon is modeled as an anharmonic oscillator with Kerr nonlinearity $\alpha$ and is coupled to the $k$-th phonon mode with strength $g_k$ (typically a few MHz [152, 158, 171]). In combination with external drives on the transmon,

$$H_d = \sum_j \Omega_j b^\dagger e^{-i\omega_j t} + \text{H.c.}, \tag{5.2}$$

this coupling provides the basic tool to initialize, manipulate, and measure phononic qubits [150, 154]. For example, itinerant photon-encoded qubits sent to the system can be routed into a particular phonon mode via pitch-and-catch schemes [172–176].

Interactions between phonon modes can be engineered by applying off-resonant drives to the transmon, and we use these interactions to implement a universal gate set for phononic qubits. The main idea is that the transmon's Kerr nonlinearity enables it to act as a four-wave mixer [177–180], so phonons can be converted from

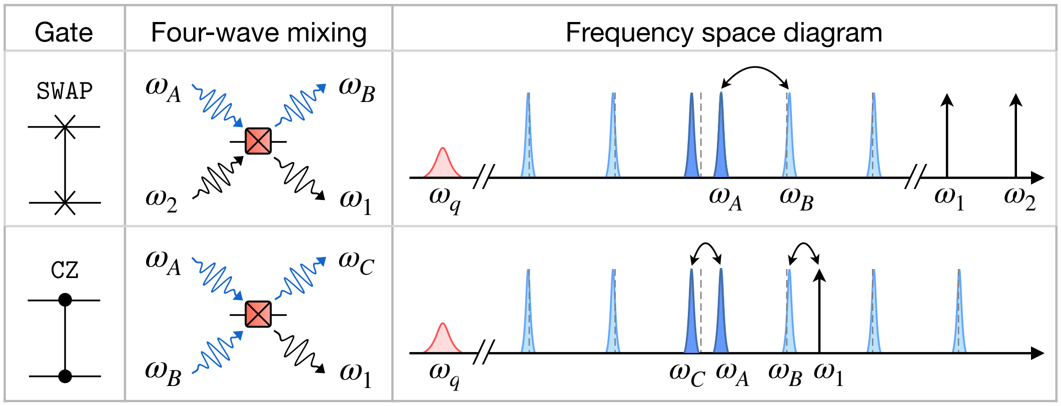


Figure 5.2: Phonon-phonon gates. SWAP: Applying two drives with $\omega_2 - \omega_1 = \omega_B - \omega_A$ creates an effective coupling between modes $A$ and $B$. CZ: Applying a single drive with $\omega_1 = \omega_A + \omega_B - \omega_C$ creates an effective three-mode coupling between modes $A$, $B$, and $C$. Frequency shifts of strongly hybridized modes (dark blue) can enable selective coupling when the modes are otherwise uniformly spaced (dashed lines denote uniform spacing). See Section 5.2.2 for further details.

one frequency to another by driving the transmon. For example, phonons can be converted from frequency $\omega_A$ to $\omega_B$ by applying two drive tones whose frequencies $\omega_{1,2}$ satisfy the resonance condition

$$\omega_2 - \omega_1 = \omega_B - \omega_A, \tag{5.3}$$

see Fig. 5.2. This driving gives rise to an effective Hamiltonian

$$H = g_v^{(1)} a_A a_B^\dagger + \text{H.c.}, \tag{5.4}$$

where

$$g_v^{(1)} = -2\alpha \frac{g_A}{\delta_A} \frac{g_B^*}{\delta_B} \frac{\Omega_1^*}{\delta_1} \frac{\Omega_2}{\delta_2} (1 - \beta^{(1)}) \tag{5.5}$$

Here, $\delta_j \equiv \omega_j - \omega_q$, and $\beta^{(1)}$ is a drive-dependent correction factor (See Section 5.2.3 for a detailed discussion of the coupling rates). Evolution under this coupling for a time $\pi/2g_v^{(1)}$ implements a SWAP gate, which exchanges the states of modes $m_A$ and $m_B$, while evolution for a time $\pi/4g_v^{(1)}$ implements a 50:50 beamsplitter operation [179].

Three-mode interactions can be similarly engineered (Fig. 5.2). Applying a single drive tone with frequency

$$\omega_1 = \omega_A + \omega_B - \omega_C \tag{5.6}$$

gives rise to the effective Hamiltonian

$$H = g_v^{(2)} a_A a_B a_C^\dagger + \text{H.c.}, \tag{5.7}$$

where

$$g_v^{(2)} = -2\alpha \frac{g_A}{\delta_A} \frac{g_B}{\delta_B} \frac{g_C^*}{\delta_C} \frac{\Omega_1^*}{\delta_1} (1 - \beta^{(2)}). \tag{5.8}$$

(See Section 5.2.3 for derivations of the coupling rates.) This three-mode interaction can be used to implement a controlled-phase (CZ) gate for qubits encoded in the $|0, 1\rangle$ phonon Fock states [181]. To perform a CZ gate between qubits in modes $A$ and $B$, mode $C$ is used as an ancilla and initialized in $|0\rangle$. Evolving for a time $\pi/g_v^{(2)}$ then enacts the mapping $|110\rangle_{ABC} \to |001\rangle \to -|110\rangle$, while leaving all other initial states unaffected. The state $|11\rangle_{AB}$ acquires a relative geometric phase, thereby implementing the CZ gate.

A variety of other operations can be similarly implemented. Single- and two-mode squeezing can be implemented by driving the transmon at appropriate frequencies, and phase shifts can be implemented in software by tuning the drive phases. Together, these two- and three-mode interactions are sufficient for universal quantum computation [182]. In the remainder of this work, however, we focus on the beam-splitter, SWAP, and CZ operations, as these are the only operations we require to implement a QRAM.

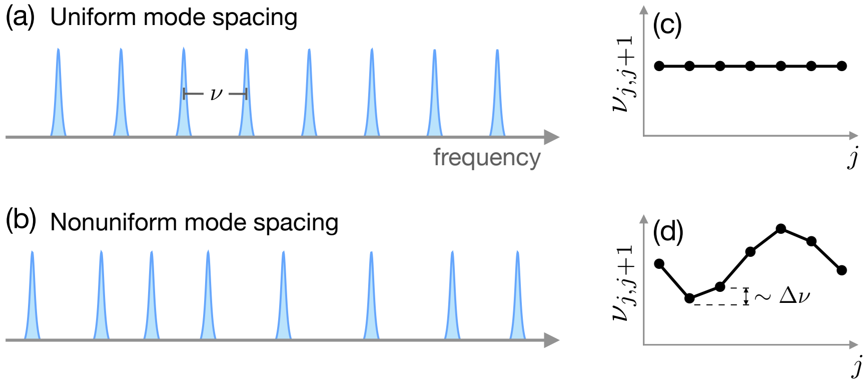


Figure 5.3: Sets of (a) uniformly and (b) nonuniformly spaced modes. (c,d) The frequency differences between successive modes are plotted to illustrate the behavior of $\nu_{j,j+1}$. (c) For uniformly spaced modes, $\nu_{j,j+1}$ is constant. (d) For nonuniformly spaced modes, $\nu_{j,j+1}$ varies on the scale of $\Delta\nu$.

### 5.2.2 Frequency selectivity

In BAW and SAW resonators, phonon mode frequencies are approximately uniformly spaced, i.e. $\omega_{j+1} - \omega_j = \nu$, where $\nu$ is the free spectral range. This uniform spacing can lead to problematic degeneracies in the resonance conditions above. Nonuniform mode spacing is thus necessary in order to ensure that the resonance conditions are nondegenerate, i.e. to ensure that a given pair or triple of modes can be selectively coupled. In this Section, we formalize the meaning of nonuniform, then present several schemes for engineering nonuniformity in BAW and SAW systems. For concreteness, we also provide example schematics for BAW and SAW devices with engineered nonuniformity. Note that phononic crystal resonators are not generally plagued by such degeneracies, since mode frequencies can be controlled by engineering the geometry of each individual phononic resonator.

As shown in Fig. 5.3, a set of modes is nonuniformly spaced if there exist mode pairs $\{i, j\}$ and $\{k, \ell\}$ for which $\nu_{ij} \neq \nu_{k\ell}$, where $\nu_{ij} = |\omega_i - \omega_j|$ is the frequency spacing between modes $i$ and $j$. In the context of multimode coupling, it is useful to quantify this nonuniformity as follows. Let $\mathcal{S}$ denote the set of all modes that are used to store quantum information, and let $\mathcal{P}$ denote the set of all mode pairs that one chooses to couple (we provide examples below). The connectivity of the system is then described by a graph with vertices $\mathcal{S}$ and edges $\mathcal{P}$. As a practically relevant measure of the nonuniformity, we define the quantity

$$\Delta\nu = \min_{\{i,j\}\in\mathcal{P}} \left[ \min_{\{k\in\mathcal{S},\ell\}\neq\{i,j\}} |\nu_{ij} - \nu_{k\ell}| \right], \tag{5.9}$$

which lowerbounds the frequency selectivity of two-mode couplings. Explicitly, the beamsplitter resonance condition, $\omega_2 - \omega_1 = \omega_B - \omega_A$ for a pair of modes $\{A, B\} \in \mathcal{P}$ is detuned from all other beamsplitter resonance conditions involving any mode in $\mathcal{S}$ by at least $\Delta\nu$. Highly selective virtual couplings thus require $g_v/\Delta\nu \ll 1$. Note that since $\Delta\nu$ depends on the choices of $\mathcal{S}$ and $\mathcal{P}$, there can exist a tradeoff between selectivity and the effective size and connectivity of the system. The definition of $\Delta\nu$ can be straightforwardly generalized to the case of three-mode couplings.

Whether a given pair or triplet of modes can be selectively coupled depends on the structure of the nonuniformity, and in this regard it is convenient to classify different sorts of nonuniformity according to properties of $\nu_{j,j+1}$. We study two such classes in the examples below: *point defect nonuniformities,* for which $\nu_{j,j+1}$ is constant except in the vicinity of a single defect, and *periodic nonuniformities,* for which $\nu_{j,j+1}$ is periodic. Of course, other classes exist, but we focus on these two classes since instances can readily be engineered in cQAD systems.

### External mode hybridization

A point defect nonuniformity can be created by coupling the phonons to some external mode, such as a microwave resonator. As demonstrated in Ref. [171], and sketched in Fig. 5.4(a,b), the resulting mode hybridization can significantly shift phonon mode frequencies within some bandwidth $\mathcal{D}$ of the external mode. The nonuniformity $\Delta\nu$ is dictated by the magnitude of these frequency shifts. For example, frequency shifts of order 1MHz were demonstrated in Ref. [171].

This class of nonuniformity can enable selective coupling: selective two-mode coupling is possible if one or both involved modes lie in $\mathcal{D}$, and selective three-mode coupling is possible if two of the three involved modes lie in $\mathcal{D}$. Hence, the set $\mathcal{S}$ can include arbitrarily many modes, but the set $\mathcal{P}$ can only include mode pairs with at least one mode in $\mathcal{D}$. While modes outside of $\mathcal{D}$ cannot be directly coupled to one another, information from these modes can instead be swapped into modes in $\mathcal{D}$, manipulated, and swapped back. Note that the coherence of the external mode should be comparable to that of the phonons, lest the hybridization result in a significant increase in effective decay rates, and in general there may exist a tradeoff between increased nonuniformity and enhanced decay.

### Two phonon mode families

Another approach is to create a periodic nonuniformity by simultaneously coupling the transmon to two families of phonon modes [151] with different free spectral ranges (FSRs). While modes within each family are uniformly spaced, the FSR difference causes the spacing between modes from different families to vary, as shown in Fig. 5.4(c,d). This nonuniformity enables two modes from different families to be selectively coupled. But because of the periodicity, selectivity is only guaranteed over a finite bandwidth smaller than one period. With two mode families, a set $\mathcal{S}$ containing $\approx \nu/\Delta\nu$ modes can be found wherein any two modes from different families can be selectively coupled with $\Delta\nu = |\nu_1 - \nu_2|$, where $\nu_{1,2}$ are the FSRs of the two families.

By itself, the use of two mode families does not enable selective three-mode coupling[1], but this limitation can be circumvented by coupling the transmon to one or

[1] At least two modes out of any three come from the same family, and since the modes in each family are uniformly spaced, there necessarily exists another set with the same resonance condition.

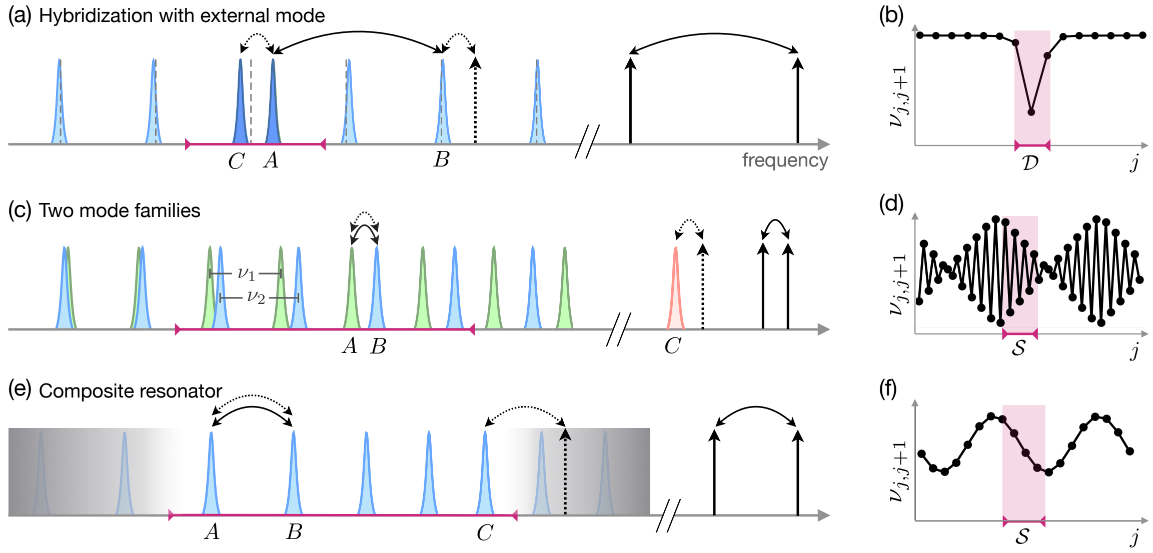


Figure 5.4: Nonuniform mode spacing. (a) External mode hybridization. The coupling between phonons and an external mode causes strongly hybridized modes (dark blue) to deviate from the otherwise uniform spacing (dashed lines). The arrows show examples of how this nonuniformity gives rise to nondegenerate resonance conditions: modes $A$ and $B$ can be coupled by the applying drives indicated by solid arrows, while modes $A$, $B$, and $C$ can be coupled by applying the drive indicated by the dashed arrow. (b) Frequency differences shrink significantly within a bandwidth $\mathcal{D}$ of the external mode. (c) Two mode families. Simultaneously coupling the transmon to two mode families (blue, green) enables selective two-mode coupling between modes from different families. Selectivity is only guaranteed in a finite region $\mathcal{S}$, and an example of such a region is highlighted in (d). The use of an external mode $C$ enables selective three-mode coupling. (e) Composite resonator. Nonuniform mode spacing in composite resonators arises due to partial reflections at the interface(s). For example, with a single interface, a simple transfer matrix treatment [183] reveals that the FSR is periodically modulated, as in (f). Selective three-mode coupling can be enabled by restricting the transmon phonon-coupling bandwidth (regions with negligible coupling are shaded in gray), or by using an external mode as in (c).

more external modes. For example, the BAW devices of Refs [149, 150] are housed in microwave cavities, and coupling the transmon to a high-Q cavity mode can enable selective three-mode coupling between the cavity and any pair of modes in $\mathcal{S}$. In a SAW device, the additional mode could come from another SAW resonator or a microwave resonator. The transmon itself could even serve as the external mode, but gate fidelities would then be directly limited by transmon coherence. Ideally, the coherence of the external mode should be comparable to that of the phonons, lest it limit gate fidelity.

### Composite resonators

Yet another approach is to employ a composite acoustic resonator, in which phonons propagate in media with different indices of refraction [Fig. 5.4(e,f)]. Reflections at the interfaces can give rise to a periodic modulation of the FSR [183]. As in the case of two mode families, this periodic nonuniformity can enable selective two-mode coupling within a finite bandwidth $\mathcal{S}$, though the magnitudes of both $\mathcal{S}$ and $\Delta\nu$ depend on the nature of the modulation.

Whether selective three-mode coupling within $\mathcal{S}$ is feasible depends on the of the specific nature of the FSR modulation. In cases where it is not already possible, selective three-mode coupling can be enabled by either coupling the transmon to some external mode, as previously described, or alternatively by restricting the bandwidth over which the transmon-phonon coupling is appreciable. For example, if the transmon-phonon coupling is only appreciable within $\mathcal{S}$, as in Fig. 5.4(e), then selective three-mode coupling is possible since the system contains an effectively finite number of nonuniformly spaced modes. In SAW systems, the coupling bandwidth can be tuned by changing the number of fingers in the interdigitated transducer[2] [152, 168].

[2] Because SAW resonators have finite bandwidth, care should be taken to avoid coupling to unconfined modes. This problem can be solved in general by engineering the transmon-phonon coupling bandwidth to lie within the SAW resonator bandwidth. The size of both bands can be tuned by varying the number of fingers in the respective interdigitated transducers [152, 168].

In BAW systems, the coupling bandwidth can be similarly tuned by changing the electromechanical transducer's geometry. For instance, in a transducer comprised of alternating layers of piezoelectric and non-piezoelectric materials, the spacing, thickness, and number of such layers could be chosen so that the coupling has a narrow response centered at a particular frequency, as in a Bragg reflector.

### Example schematics

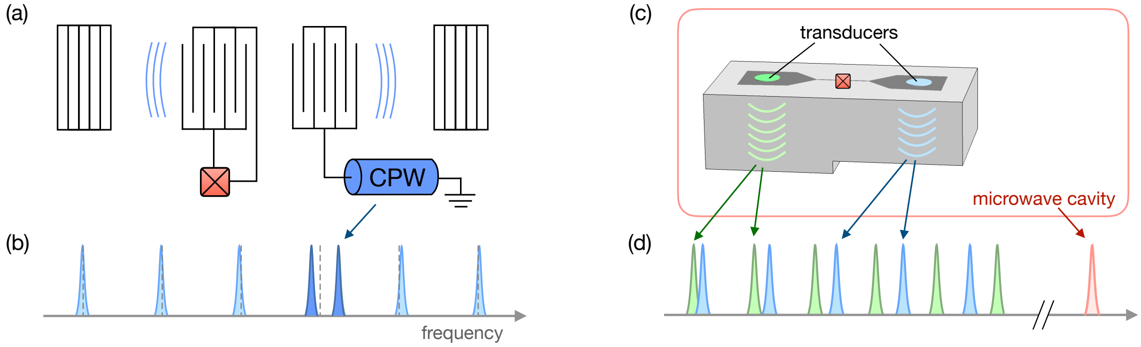


Figure 5.5: SAW and BAW devices with engineered nonuniformity. (a) The modes of a SAW resonator are coupled to both a transmon and a coplanar waveguide (CPW) resonator. Hybridization with the resonator mode creates nonuniformity. (b) Mode frequencies of the device in (a). The CPW resonator mode and the phonon mode with which it most strongly hybridizes are shown in dark blue. (c) A 3D transmon couples to both a microwave cavity mode and to phonon modes from two BAW resonators with different FSRs (the difference is engineered by reducing the thickness of the substrate under one of the transducers). (d) Mode frequencies of the device in (c).

For concreteness, in Fig. 5.5 we provide example schematics for SAW and BAW devices in which nonuniformity is engineered according to the strategies described above. Fig. 5.5(a) shows a SAW device that exploits the external mode hybridization strategy. A SAW resonator is fabricated on a piezoelectric substrate, and coupling between the transmon and the phononic modes is enabled by an interdigitated capacitor. A superconducting coplanar waveguide resonator is also coupled to the phononic modes, and the hybridization of the phononic modes with the resonator mode creates the necessary nonuniformity.

Fig. 5.5(c) shows a BAW device that exploits the two mode families strategy. The device is based on those demonstrated in Refs. [149, 150]; a three-dimensional (3D) transmon is housed inside a microwave cavity, and thin disks of piezoelectric material (transducers) fabricated in the transmon's pads enable the transmon to couple to BAW modes in the substrate. Two modifications have been made relative to the devices in Refs. [149, 150]. First, an additional transducer has been added so that the transmon simultaneously couples to two families of modes. Second, the thickness of the substrate beneath one of the transducers has been reduced so that the two families have different FSRs. The microwave cavity mode, which dispersively couples to the transmon, provides the external mode necessary to enable selective three-mode couplings. We note that other elements, e.g. a separate readout resonator for the transmon, can be integrated into 3D architectures in such a way that the transmon can be driven and measured without involving the cavity mode [184].

### 5.2.3 Estimates of achievable coupling rates

In this section, we study the virtual coupling rates

$$g_v^{(1)} = -2\alpha\xi_1^*\xi_2\lambda_A\lambda_B^*(1-\beta^{(1)}), \tag{5.10}$$

$$g_v^{(2)} = -2\alpha\xi_1^*\lambda_A\lambda_B^*\lambda_C(1-\beta^{(2)}). \tag{5.11}$$

Below, we define the notation, derive these expressions, and discuss the importance of the corrections $\beta^{(1,2)}$ for cQAD systems. Then, in order to verify the accuracy of these expressions, we compare them to numerical results obtained using the Floquet theory methods of Ref. [180].

#### Derivation of the virtual coupling rates

To derive the expressions (5.10) and (5.11), we begin with the multimode cQAD Hamiltonian (Eq. (5.1)) and perform a unitary transformation defined by $U_1 =$

$\exp iH_0t$, where $H_0 = \omega_q b^\dagger b + \sum_k \omega_k a_k^\dagger a_k$. Thus,

$$H = \sum_j \left(\Omega_j b^\dagger e^{-i\delta_j t} + \text{H.c.}\right) + \sum_k \left(g_k a_k b^\dagger e^{-i\delta_k t} + \text{H.c.}\right) - \frac{\alpha}{2} b^\dagger b^\dagger b b, \tag{5.12}$$

where $\delta_k = \omega_k - \omega_q$ is the detuning of the $k^{th}$ phonon mode, while $\delta_j = \omega_j - \omega_q$ and $\Omega_j$ are the detuning and the strength of the $j^{th}$ drive tone, respectively. In the spirit of Ref. [185], we first perform unitary transformations to eliminate the qubit-phonon couplings and drive terms then consider the effects of the anharmonicity. For convenience of notation, we introduce the dimensionless parameters $\lambda_k \equiv g_k/\delta_k$ and $\xi_j \equiv \Omega_j/\delta_j$. To leading order in $\lambda_k \ll 1$, the unitary that eliminates the couplings is $U_2 = \exp \sum_k (\lambda_k^* a_k^\dagger b e^{i\delta_k t} - H.c)$, and that which eliminates the drives is $U_3 = \exp \sum_j (\xi_j^* b e^{i\delta_j t} - H.c)$. The combined effect of these two transformations is to enact the mapping

$$q \to b + \sum_j \xi_j e^{-i\delta_j t} + \sum_k \lambda_k a_k e^{-i\delta_k t} \equiv Q, \tag{5.13}$$

so that the Hamiltonian becomes

$$H = -\frac{\alpha}{2} Q^\dagger Q^\dagger Q Q. \tag{5.14}$$

Note that we have neglected linear terms of the form $(\Omega_j^* \lambda_k a_k e^{i(\delta_j - \delta_k)t} + \text{H.c.})$. This omission is justified in the RWA provided that $|\delta_j - \delta_k| \gg \lambda_k \Omega_j$, i.e. that the drives are sufficiently far detuned from any modes in which we are interested. For simplicity, we also neglect frequency (Stark) shifts of the phononic eigenmodes—we describe their effects in Ref. [99].

When two drive tones are applied whose frequencies satisfy the resonance condition $\omega_2 - \omega_1 = \omega_B - \omega_A$, the Hamiltonian (5.14) contains a resonant beamsplitter-type coupling, $g_v^{(1)} a_A a_B^\dagger + \text{H.c.}$, where

$$g_v^{(1)} = -2\alpha \xi_1^* \xi_2 \lambda_A \lambda_B^*. \tag{5.15}$$

Similarly, when a single drive tone is applied with frequency[3] $\omega_1 = \omega_A + \omega_C - \omega_B$, the Hamiltonian contains a resonant three-mode coupling $g_v^{(2)} a_A a_B^\dagger a_C + \text{H.c.}$, where

$$g_v^{(2)} = -2\alpha \xi_1^* \lambda_A \lambda_B^* \lambda_C. \tag{5.16}$$

### Corrections to the virtual coupling rates

The Hamiltonian (5.14) contains many terms beyond just the resonant terms discussed above (see Table 5.1). Most of these terms are rapidly-rotating and can be neglected in the RWA assuming dispersive coupling ($\lambda \ll 1$) and weak drives ($\xi \ll 1$). However, other terms can produce corrections $\beta^{(1,2)}$ to the coupling rates. In this section, we first calculate these corrections to leading order in $\lambda$ and $\xi$. Then, we derive nonperturbative contributions associated with the AC Stark shift.

Table 5.1: Catalog of terms in the Hamiltonian (5.14). Summations run over all drives and all modes, including the transmon mode $q$, for which $\lambda_q = 1$ and $\delta_q = 0$.

| Term | Description |
|---|---|
| $\frac{\alpha}{2}\sum_{i,j,k,l} \xi_i^* \xi_j^* \xi_k \lambda_l a_l e^{i(\delta_i+\delta_j-\delta_k-\delta_l)t} + \text{H.c.}$ | Drive |
| $\frac{\alpha}{2}\sum_{i,j,k,l} \xi_i^* \xi_j \lambda_k^* \lambda_l a_k^\dagger a_l e^{i(\delta_i-\delta_j+\delta_k-\delta_l)t} + \text{H.c.}$ | Beamsplitter |
| $\frac{\alpha}{2}\sum_{i,j,k,l} \xi_i^* \xi_j^* \lambda_k \lambda_l a_k a_l e^{i(\delta_i+\delta_j-\delta_k-\delta_l)t} + \text{H.c.}$ | Two-mode squeezing |
| $\frac{\alpha}{2}\sum_{i,j,k,l} \xi_i^* \lambda_j^* \lambda_k \lambda_l a_j^\dagger a_k a_l e^{i(\delta_i+\delta_j-\delta_k-\delta_l)t} + \text{H.c.}$ | $\chi^{(2)}$ nonlinearity |
| $\frac{\alpha}{2}\sum_{i,j,k,l} \lambda_i^* \lambda_j^* \lambda_k \lambda_l a_i^\dagger a_j^\dagger a_k a_l e^{i(\delta_i+\delta_j-\delta_k-\delta_l)t} + \text{H.c.}$ | $\chi^{(3)}$ nonlinearity |

The leading order contribution to $\beta^{(1,2)}$ is zeroth order in both $\lambda$ and $\xi$. The only terms in the Hamiltonian (5.14) which contribute to $\beta^{(1)}$ and $\beta^{(2)}$ at this order are,

[3] In this section we consider the case $\omega_A < \omega_B < \omega_1 < \omega_C$, which nicely highlights the similarities between $g_v^{(1)}$ and $g_v^{(2)}$. The derivations proceed analogously for other cases, such as the case of $\omega_C < \omega_A < \omega_B < \omega_1$ shown in Fig. 2 of the main text.

respectively,

$$\left[-\alpha(b^{\dagger 2}\xi_2\lambda_A a_A + b^{\dagger 2}\xi_1\lambda_B a_B)e^{-i(\delta_B+\delta_1)t} + \text{H.c.}\right], \tag{5.17}$$

$$\left[-\alpha(b^{\dagger 2}\lambda_A a_A\lambda_C a_C + b^{\dagger 2}\xi_1\lambda_B a_B)e^{-i(\delta_B+\delta_1)t} + \text{H.c.}\right]. \tag{5.18}$$

The corrections from these terms can be calculated via standard perturbation theory,

$$\beta^{(1,2)} = \frac{\alpha}{\delta_B + \delta_1 + \alpha}.$$

For the SWAP operation, where the drives are far-detuned, this correction is typically negligible. However, for the CZ operation, this correction can significantly reduce the coupling rate since $\delta_1, \delta_B$ can be comparable to $\alpha$. We note that the expression for $\beta^{(1)}$ matches the leading order expression derived in Ref. [180].

Contributions to $\beta^{(1,2)}$ at higher orders in $\lambda$ can be neglected since we have assumed the dispersive regime, $\lambda \ll 1$. Contributions at higher orders in $\xi$ can be systematically calculated with perturbation theory in principle, but such calculations quickly become tedious. Here, we employ an alternative approach. We consider the AC Stark shift type terms, $-2\alpha\sum_j |\xi_j|^2 Q^\dagger Q$, and compute their contributions to $\beta^{(1,2)}$ nonperturbatively by working in a rotating frame.

Let $S$ denote the qubit's AC Stark shift. In the frame where the qubit mode rotates at its Stark-shifted frequency, $\tilde{\omega_q} = \omega_q + S$, the system Hamiltonian is

$$H = -Sb^\dagger b + \sum_j \left[\Omega_j b^\dagger e^{-i\tilde{\delta}_j t} + \text{H.c.}\right] + \sum_k \left[g_k a_k b^\dagger e^{-i\tilde{\delta}_k t} + \text{H.c.}\right] - \frac{\alpha}{2}b^\dagger b^\dagger b b \tag{5.19}$$

where $\tilde{\delta} = \omega - \tilde{\omega_q}$. Performing unitary transformations analogous to those above eliminates the coupling and drive terms so that $H = -S\tilde{Q}^\dagger\tilde{Q} - \frac{\alpha}{2}\tilde{Q}^\dagger\tilde{Q}^\dagger\tilde{Q}\tilde{Q}$, where $\tilde{Q} = q + \sum_j \tilde{\xi}_j e^{-i\tilde{\delta}_j t} + \sum_k \tilde{\lambda}_k a_k e^{-i\tilde{\delta}_k t}$. Here, $\tilde{\xi}_j = \Omega_j/\tilde{\delta}_j$ and $\tilde{\lambda}_k = g_k/\tilde{\delta}_k$. The Stark shift terms can then be cancelled by setting[4] $S = -2\alpha \sum_j |\tilde{\xi}_j|^2$. In the frame where the Stark shift terms are eliminated, one finds modified expressions for the corrections,

$$\beta^{(1)} = 1 - \frac{\delta_1 \delta_2 \delta_A \delta_B}{\tilde{\delta}_1 \tilde{\delta}_2 \tilde{\delta}_A \tilde{\delta}_B} \frac{\tilde{\delta}_B + \tilde{\delta}_1}{\tilde{\delta}_B + \tilde{\delta}_1 + \alpha} \tag{5.20}$$

$$\beta^{(2)} = 1 - \frac{\delta_1 \delta_A \delta_B \delta_C}{\tilde{\delta}_1 \tilde{\delta}_A \tilde{\delta}_B \tilde{\delta}_C} \frac{\tilde{\delta}_B + \tilde{\delta}_1}{\tilde{\delta}_B + \tilde{\delta}_1 + \alpha}. \tag{5.21}$$

Hence, the coupling rates are

$$g_v^{(1)} = -2\alpha \tilde{\xi}_1^* \tilde{\xi}_2 \tilde{\lambda}_A \tilde{\lambda}_B^* \frac{\tilde{\delta}_B + \tilde{\delta}_1}{\tilde{\delta}_B + \tilde{\delta}_1 + \alpha} \tag{5.22}$$

$$g_v^{(2)} = -2\alpha \tilde{\xi}_1^* \tilde{\lambda}_A \tilde{\lambda}_B^* \tilde{\lambda}_C \frac{\tilde{\delta}_B + \tilde{\delta}_1}{\tilde{\delta}_B + \tilde{\delta}_1 + \alpha}. \tag{5.23}$$

These expressions have the same form as above, but with the replacements $\delta \to \tilde{\delta}$, i.e. detunings are now defined relative to the qubit's Stark-shifted frequency. It follows that there also exists a Stark shift correction $\beta^{(\gamma)}$ to the inverse-Purcell enhancement

$$\kappa_\gamma = \kappa + \gamma (g/\delta)^2 (1 + \beta^{(\gamma)}), \tag{5.24}$$

where $\beta^{(\gamma)} = (\delta/\tilde{\delta})^2 - 1$, i.e. $\kappa_\gamma = \kappa + \gamma(g/\tilde{\delta})^2$. These corrections are important whenever the drives are strong enough that the qubit's Stark shift becomes comparable to the drive or mode detunings. Expressions (5.22), (5.23), and (5.24) are used to produce the plots in this chapter.

[4] This equation determines $S$ implicitly; to leading order in the drives, $S = -2\alpha \sum_j |\Omega_j|^2/\delta_j^2$. However, the Hamiltonian (5.14) contains the terms $(\alpha \xi_{1,2} q^{\dagger 2} q e^{-i\delta_{1,2} t} + \text{H.c.})$, which also contribute to $S$ at this order. Employing perturbation theory, one finds $S = -2\alpha \sum_j |\Omega_j|^2/\delta_j(\delta_j + \alpha)$, which matches the leading order calculation in Ref. [180]. This latter expression is used in the numerics throughout this work.

### Comparison with numerical Floquet calculation

To assess the accuracy of the expressions (5.22) and (5.23), we compare with numerical calculations of the coupling rates using the methods developed in Ref. [180]. First, we briefly summarize the main results of that work. The authors consider the process of engineering a bilinear interaction between two microwave cavity modes that are mutually coupled to a transmon qubit. Treating the couplings as a perturbation, they calculate the linear response of the driven transmon. This perturbative treatment is justified in the dispersive regime. They show that $g_v^{(1)}$ can be calculated in terms of a susceptibility matrix $\chi^{(1)}(\omega_A, \omega_B; \omega_1, \omega_2)$, that describes the response of the driven transmon at frequency $\omega_A$ to a weak probe field at $\omega_B$, when subject to drives at $\omega_1$ and $\omega_2$. The susceptibility can then be computed numerically to all orders in the drive amplitudes using Floquet theory. The authors find good quantitative agreement between their theoretical predictions and experimental results, even for strong drives ($\xi > 1$).

This approach can be directly applied to calculate $g_v^{(1)}$. To calculate $g_v^{(2)}$, we analogously define a higher-order susceptibility matrix $\chi^{(2)}(\omega_A, \omega_B, \omega_C; \omega_1)$ that captures the response of the transmon at frequency $\omega_A$ to weak probes at $\omega_B$ and $\omega_C$, when subject to a drive at $\omega_1$. Rather than computing $\chi^{(2)}$ directly, which can be numerically tedious, we note that $\chi^{(2)}$ can be computed in terms of $\chi^{(1)}$. In the calculation of $\chi^{(1,2)}$, the drives and probes are treated identically at the Hamiltonian level; both the drive and probe terms are of the form $H = f_j\, b^\dagger e^{-i\omega_j t} + \text{H.c.}$. For the drives, $f_j = \Omega_j$, while for the probes, $f_j = g a_j$. Since the susceptibility is calculated to all orders in the drive fields but only to leading order in the weak probe fields, going beyond leading order does not change the result in the limit where the field $f_j$ is weak. Weak probes and weak drives are thus interchangable, the only difference being a matter of interpretation. It follows that

$$\chi^{(2)}(\omega_A, \omega_B, \omega_C; \omega_1) = \chi^{(1)}(\omega_A, \omega_B; \omega_1, \omega_C). \tag{5.25}$$

This equivalence holds for $g_C \ll \delta_C$, which is the same limit that was already assumed to justify the perturbative treatment. Thus, the numerical procedure for calculation of $g_v^{(1)}$ can also be straightforwardly applied to calculate $g_v^{(2)}$.

In Figs. 5.6(a) and (b), we calculate $g_v^{(1,2)}$ numerically as described above, and we compare the results with the analytical expressions (5.22) and (5.23). Good agreement is observed for weak drives ($\xi \lesssim 0.4$ for the parameters used in the plots). Discrepancies emerge at stronger drives, but this is expected because the corrections are obtained perturbatively. In Fig. 5.6 (c), (d), the coupling rates are plotted as a function of $\delta_A$ to make apparent the importance of the AC Stark shift corrections. Due to the Stark shift, the corrected expressions and numerics are both red-shifted relative to the uncorrected expressions. Were the corrections not included, this relative shift would result in a systematic overestimation of the coupling rates for blue-detuned phonon modes.

The AC Stark shift is responsible for the interesting non-monotonic behavior of expressions (5.22) and (5.23) with $\xi$. Intuitively, this behavior is explained by the fact that the Stark shift causes the qubit to move *away* from the phonon modes in frequency space. This reduces the participation of the phonons in the qubit mode, therby reducing the coupling rate. When optimizing $g_v$ so as to minimize the SWAP or CZ infidelity, the non-monotonicity effectively restricts the drive amplitudes to the range $\xi \leq \xi_{\text{crit.}}$, where $\xi_{\text{crit.}}$ is the value of $\xi$ for which $g_v$ is maximal. For the parameters consider in Fig. 5.6, the virtual couplings rates are thus restricted to $|g_v^{(1)}|/2\pi < 100\,\text{kHz}$ and $|g_v^{(2)}|/2\pi < 25\,\text{kHz}$. Good agreement between the analytics and numerics is observed for $\xi \lesssim \xi_{\text{crit.}}$, validating the use of expressions (5.22) and

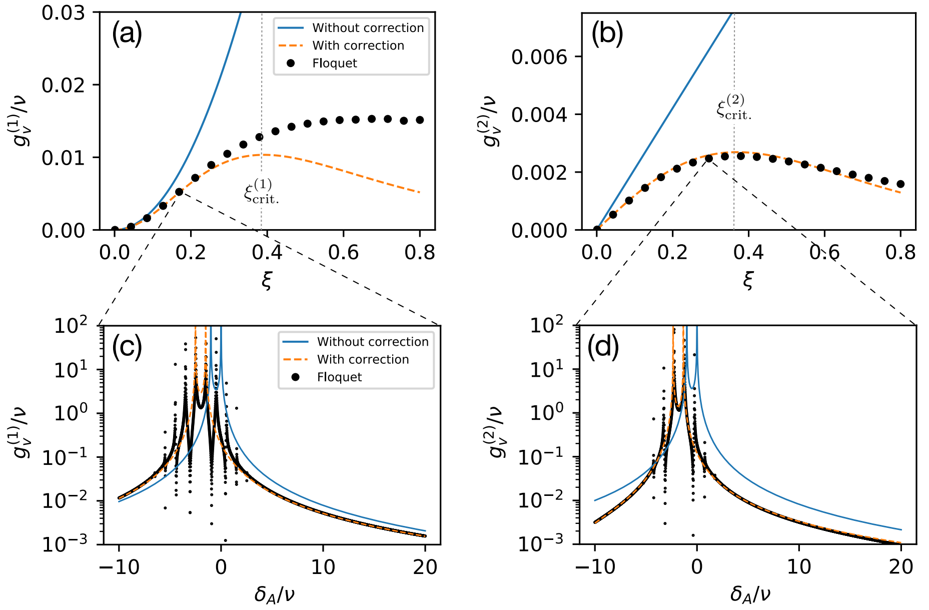


Figure 5.6: Comparison of the coupling rate expressions with numerical Floquet calculations. (a), (b) Coupling rates $g_v^{(1,2)}$ plotted as a function of drive strength. (c), (d) Coupling rates plotted as a function of the phonon mode detuning $\delta_A$. The uncorrected coupling rates exhibit two resonant peaks, at $\delta_A = 0$ and $\delta_A + \nu = \delta_B = 0$, corresponding to resonant processes where phononic excitations in modes $A$ or $B$ are converted to transmon excitations. Because of the AC Stark shift, these peaks are red-shifted in both the numerical Floquet calculation and the corrected expressions. The additional resonant peaks in the numerical calculation correspond to multiphoton resonances where phononic excitations are converted to transmon excitations by exchanging an integer number of photons between the two drive fields [180]. It is important to carefully avoid these peaks in the experiments. Parameters for all plots: $g_k/2\pi = 10\text{MHz}$, $\delta_A/2\pi = 100\text{MHz}$, $\nu/2\pi = 10\text{MHz}$, $\Delta\nu = \nu/10$. In order to account for the AC Stark shift, we also specify $\alpha/2\pi = 150\text{MHz}$, and we take $\delta_1/2\pi = 1\text{GHz}$ in the calculation of $g_v^{(1)}$. In (c), $\xi_{1,2} = 0.17$, and in (d) $\xi_1 = 0.27$.

(5.23).

This comparison illustrates the importance of the corrections derived above and confirms that the virtual coupling rates are well-described by expressions (5.22) and (5.23) for the drive strengths considered in this chapter.

### 5.2.4 Estimates of gate fidelities

During the gates described in Section 5.2.1, the transmon is never directly excited; instead, it is only *virtually* excited, so infidelity attributable to transmon decoherence is suppressed. These virtual gates can thus provide great advantage in cQAD systems, where transmon decoherence is likely to be the limiting factor. This is in contrast to existing proposals [103, 166], in which gates between resonator mode qubits are implemented by swapping information *directly* into the transmon using resonant interactions of the form $g_d(b^\dagger a + ba^\dagger)$, which can be engineered, e.g., by modulating the transmon's frequency. In the following, we compare the predicted fidelities of the *virtual* gates proposed here and the *direct* gates considered in Refs. [103, 166].

In a multimode architecture, there exists a fundamental tradeoff between decoherence and spectral crowding. Slower gates are more prone to decoherence, while faster gates have reduced frequency resolution and can disrupt other modes. We can quantify these effects as follows. Let $\kappa$ and $\gamma$ denote the bare phonon and transmon decoherence rates, respectively. Similarly, let $\kappa_\gamma^j = \kappa + \gamma(g_j/\delta_j)^2(1+\beta^{(\gamma)})$ denote the dressed decay rate of phonon mode $j$, which includes a contribution from the inverse Purcell effect [136, 180] and a drive-dependent correction $\beta^{(\gamma)}$ as described in Section 5.2.3. The contributions to the direct and virtual gate infidelities from decoherence are, respectively,

$$(\kappa+\gamma)t_d \quad \text{and} \quad \bar{\kappa}_\gamma t_v.$$

Here, $t_d = (c_d\pi/2g_d)$ and $t_v = (c_v\pi/2g_v)$ are the total gate times, where $c_{d,v}$ are gate-dependent constants. ($c_v = 1$ for SWAP, and $c_v = 2$ for CZ, as these gates have durations $\pi/2g_v$ and $\pi/g_v$ respectively. As discussed in Ref. [103], $c_d = 5$ for SWAP and $c_d = 4$ for CZ.) To estimate the gate infidelity, we have multiplied these gate times by the corresponding total decoherence rates. During direct gates, information spends roughly equal time in the phonon and transmon modes, so the total decoherence rate is $\kappa + \gamma$ (we assume the inverse Purcell enhancement to $\kappa$ is negligible relative to $\gamma$). During virtual gates, the total decoherence rate $\bar{\kappa}_\gamma$ depends on whether the gate is implemented using a two- or three-mode coupling. For two-mode couplings, the total decoherence rate is $\bar{\kappa}_\gamma = \kappa_\gamma^A + \kappa_\gamma^B$, while for three-mode couplings the rate is $\bar{\kappa}_\gamma = (\kappa_\gamma^A + \kappa_\gamma^B + \kappa_\gamma^C)/2$. The factor of 1/2 in the latter expression results from averaging over the gate duration; when there is a phonon each in modes $A$ and $B$, the rate is $\kappa_\gamma^A + \kappa_\gamma^B$, but once these two phonons have been converted into a single phonon in mode $C$ the rate is $\kappa_\gamma^C$.

The presence of other modes also contributes to the infidelity, regardless of whether these other modes are used to store quantum information. When performing a gate, transitions between the modes involved in the gate and the other modes are driven off-resonantly, and we approximate the spectral crowding infidelity as the probability that one of these unwanted transitions occurs. This probability is computed in Ref. [166] for direct gates: assuming a set of uniformly spaced modes with free spectral range $\nu$, the infidelity is approximately[5]

$$\sum_n \left(\frac{g_d}{\delta_n}\right)^2 \approx \left(\frac{g_d}{\nu}\right)^2,$$

where the sum on the left runs over all unwanted transmon-mode transitions, each

[5] Following Ref. [166], we neglect a constant prefactor of order 1 on the right hand side, with the justification that $(g_d/\nu)^2$ is actually a pessimistic upper bound; the spectral crowding infidelity can be reduced by smoothly ramping up the drives.

detuned by successive multiples of $\nu$, i.e. $\delta_n = \{\pm\nu, \pm 2\nu, \ldots\}$. Importantly, this infidelity is independent of the total number of modes in the system. Similarly, for virtual gates the spectral crowding infidelity is approximately

$$\sum_n \left(\frac{g_v}{\delta_n'}\right)^2 \approx \left(\frac{g_v}{\Delta\nu}\right)^2,$$

where the sum runs over all unwanted virtual couplings that affect modes involved in the gate. More precisely, $n$ indexes all unwanted two-mode (three-mode) couplings which involve at least one mode from the set $\{A, B\}$ ($\{A, B, C\}$), and $\delta_n'$ is the resonance condition detuning between the $n$-th unwanted coupling and the desired coupling. For example, in the two-mode case, these detunings are of the form $\delta_n' = (\omega_B - \omega_A) - (\omega_i - \omega_j)$, with either $i$ or $j \in \{A, B\}$. In performing the sum we have assumed that unwanted couplings are detuned by successive multiples of $\Delta\nu$, i.e. $\delta_n' = \{\pm\Delta\nu, \pm 2\Delta\nu, \ldots\}$. In general the $\delta_n'$ depend on the specific structure of the nonuniformity, but the scaling $(g_v/\Delta\nu)^2$ holds regardless.

Summing the contributions from decoherence and spectral crowding yields estimates for the infidelity of direct and virtual gates,

$$1 - \mathcal{F}_d \approx (\gamma + \kappa_\gamma)\left[\frac{c_d \pi}{2 g_d}\right] + \left(\frac{g_d}{\nu}\right)^2, \tag{5.26}$$

$$1 - \mathcal{F}_v \approx \bar{\kappa}_\gamma \left[\frac{c_v \pi}{2 g_v}\right] + \left(\frac{g_v}{\Delta\nu}\right)^2. \tag{5.27}$$

Evidently, the competition between decoherence and spectral crowding results in an optimal coupling rate [166]. By adjusting the drive strengths, $g_{d,v}$ can be tuned to their respective optima. The optimal infidelities are

$$1 - \mathcal{F}_d \approx \frac{3}{2}\left[\frac{c_d \pi (\kappa + \gamma)}{\sqrt{2}\nu}\right]^{2/3}, \tag{5.28}$$

$$1 - \mathcal{F}_v \approx \frac{3}{2}\left[\frac{c_v \pi \bar{\kappa}_\gamma}{\sqrt{2}\Delta\nu}\right]^{2/3}. \tag{5.29}$$

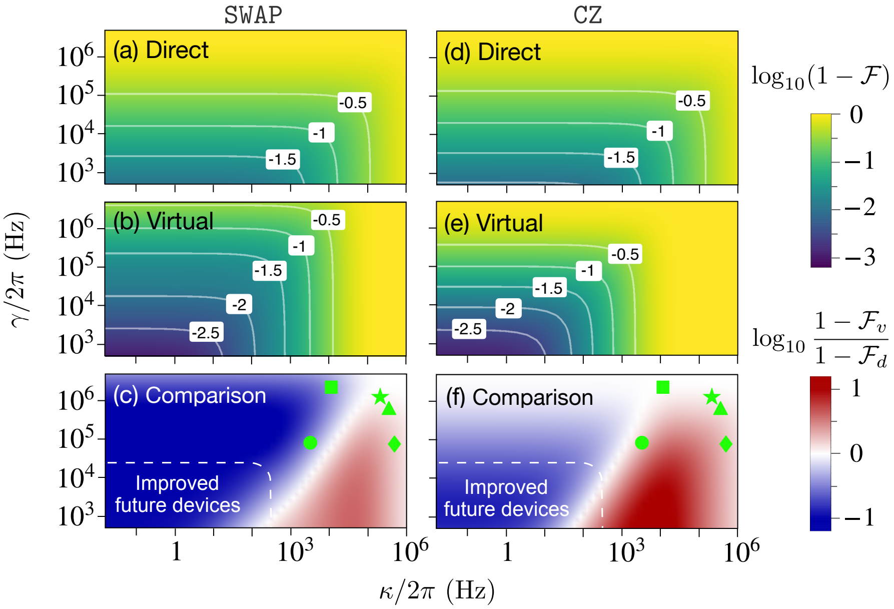


Figure 5.7: Comparison of direct and virtual operations. (a,b) $\log_{10}(1-\mathcal{F})$ for the direct and virtual SWAP operations, respectively. The couplings are optimized subject to constraints ($g_d \in [0, g]$, constraints on $g_v$ are discussed in Section 5.2.3). (c) Comparison of direct and virtual SWAP operations. The log ratio of the infidelities is plotted, with the virtual operations attaining higher fidelities in the blue region. (d,e) Log$_{10}$ infidelity for the direct and virtual CZ operations. (f) Comparison of CZ operations. For reference, the symbols {●,■,▲,♦,★} respectively denote the decoherence rates $\kappa$ (phonon) and $\gamma$ (transmon) measured in Refs. [150], [152], [158], [154], and [155]. Note, however, that the plots are generated using typical parameter values, not specific values from any one experiment. Parameters: $g/2\pi = 10$MHz, $\delta/2\pi = 100$MHz, $\nu/2\pi = 10$MHz, and $\Delta\nu/2\pi = 1$MHz.

While transmon and phonon decoherence contribute equally to $1-\mathcal{F}_d$, transmon decoherence only makes a small contribution to $1-\mathcal{F}_v$ via the inverse Purcell effect, wherein $\gamma$ is suppressed by a factor of $(g/\delta)^2 \ll 1$. The virtual gates can thus be expected to attain higher fidelities when there is a large disparity between $\gamma$ and $\kappa$, i.e. for sufficiently long-lived phonon modes. Indeed, $\mathcal{F}_v > \mathcal{F}_d$ whenever $\kappa_\gamma \lesssim (\kappa+\gamma)\Delta\nu/\nu$, provided the optimal coupling rates can be reached.

In Fig. 5.7, we plot the optimal infidelities of direct and virtual gates as a function of $\kappa$ and $\gamma$ for realistic experimental parameters. The comparison reveals that virtual gates can be performed with high fidelity (>99%) given long-lived phonons, and that virtual gates attain higher fidelities than direct gates in the same regime. Indeed, realistic improvements in phonon coherence are likely to bring near-term devices into this $\mathcal{F}_v \gg \mathcal{F}_d$ regime (Fig. 5.7c,f).

We briefly note other factors relevant to the comparison of direct and virtual gates. *Multi-phonon encodings:* Direct gates require that qubits be encoded in the $|0,1\rangle$ Fock states, while virtual operations are compatible with multi-phonon encodings, including some bosonic quantum error-correcting codes [182, 186]. *Parallelism:* Direct gates must be executed serially, while virtual gates can be executed in parallel by simultaneously applying the requisite drives (though care must be taken to ensure that the additional drives do not bring spurious couplings on resonance). *Speed:* Virtual gates are inherently slower than direct gates, with realistically attainable virtual coupling rates on the order of $g_v/2\pi \sim 10 - 100\,\mathrm{kHz}$ (see Section 5.2.3).

## 5.3 Quantum computing with acoustics, approach 2: stabilized cat qubits

In this section, we propose a quantum computing architecture for multimode cQAD systems based on dissipatively stabilized cat qubits. As in the previous section (Section 5.2), the use of acoustic systems naturally affords our architecture improved hardware efficiency and scalability. The use of cat qubits further improves scalability; cat qubits' biased noise can be exploited to reduce error correction overheads and improve code thresholds. In order to scale up this cat-qubit architecture, it is crucial to stabilize and couple multiple cat qubits in a way that both maximizes connectivity and minimizes crosstalk. To this end, we show how multiple cat qubits can be simultaneously stabilized by a single, shared nonlinear element, enabling increased connectivity and hardware efficiency. Further, we enumerate the sources of crosstalk in such architectures and show how the dominant sources can be effectively suppressed via filtering. We note that, though we tailor our analysis specifically to acoustic systems, these results are also applicable to multimode cQED systems.

In Section 5.3.1 we provide a brief review of cat qubits, stabilization methods, and bias-preserving gates. Then, in Section 5.3.2 we give a broad overview of our proposed architecture. In the sections that follow, we explore several practical aspects of the proposal in further detail. In Section 5.3.3, we describe our multiplexed stabilization scheme, and in Section 5.3.4 we enumerate the associated sources of crosstalk. Finally in Sections 5.3.5 and 5.3.6 we present two strategies for mitigating crosstalk that, when used in conjunction, enable one to suppress all dominant sources of crosstalk in the architecture.

### 5.3.1 Review of cat qubits

Cat qubits [141, 187–189] are examples of so-called bosonic qubits [190, 191], where a qubit is encoded within some two-level subspace of a bosonic mode's infinite dimensional Hilbert space. In the particular case of two-component cat qubits, we define the $|+\rangle$ and $|-\rangle$ logical states of the cat qubit to be superpositions of coherent states,

$$|\pm\rangle \equiv |C_\alpha^\pm\rangle \equiv \mathcal{N}_\pm(|\alpha\rangle \pm |{-\alpha}\rangle), \tag{5.30}$$

where $|\alpha\rangle$ denotes a coherent state with complex amplitude $\alpha$, and

$$\mathcal{N}_\pm = 1/\sqrt{2(1 \pm e^{-2|\alpha|^2})}. \tag{5.31}$$

The $|0,1\rangle$ logical states of the cat qubit are given by

$$|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) = |+\alpha\rangle + O(e^{-2|\alpha|^2})\,|{-\alpha}\rangle \tag{5.32}$$

$$|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) = |{-\alpha}\rangle + O(e^{-2|\alpha|^2})\,|{+\alpha}\rangle\,. \tag{5.33}$$

These states become orthogonal in the limit of $|\alpha|^2 \gg 1$, approaching $|+\alpha\rangle$ and $|-\alpha\rangle$ respectively.

Cat qubits are particularly interesting because they can exhibit *biased noise*, meaning that the rates of bit-flip and phase-flip errors differ. Physically, the reason for this discrepancy in error rates is that realistic errors in bosonic modes, such as excitation loss and dephasing, tend to act locally in phase space. As such, the probability that an error maps the system from $|0\rangle \approx |{+\alpha}\rangle$ all the way to $|1\rangle \approx |{-\alpha}\rangle$ or vice versa (i.e., a bit flip) is highly suppressed as $|\alpha|^2$ increases. On the other hand, the probability that an error maps $|+\rangle$ to $|-\rangle$ or vice versa (i.e., a phase flip) is not suppressed, because these two states remain close to one another in phase space as $|\alpha|^2$ increases. For example, consider excitation loss errors, which correspond to the application of the bosonic mode's annihilation operator $a$. Within the cat qubit code space, this error operator can be expressed as

$$\begin{aligned} a &= \alpha\left[\frac{\mathcal{N}_+}{\mathcal{N}_-}\left|C_\alpha^-\right\rangle\left\langle C_\alpha^+\right| + \frac{\mathcal{N}_-}{\mathcal{N}_+}\left|C_\alpha^+\right\rangle\left\langle C_\alpha^-\right|\right] \\ &= \alpha\left[\frac{1}{2}\left(\frac{\mathcal{N}_-}{\mathcal{N}_+} + \frac{\mathcal{N}_+}{\mathcal{N}_-}\right) Z + \frac{1}{2}\left(\frac{\mathcal{N}_-}{\mathcal{N}_+} - \frac{\mathcal{N}_+}{\mathcal{N}_-}\right) iY\right] \\ &= \alpha\left[Z + O(e^{-2|\alpha|^2})iY\right], \end{aligned} \tag{5.34}$$

where $Z$ and $Y$ are the usual Pauli matrices. If excitation losses occur at a rate $\kappa$, the associated bit-flip and phase-flip rates in the cat qubit will scale as

$$\Gamma_{\text{bit-flip}} \sim \kappa|\alpha|^2 e^{-4|\alpha|^2}, \tag{5.35}$$

$$\Gamma_{\text{phase-flip}} \sim \kappa|\alpha|^2. \tag{5.36}$$

Thus, provided that the system remains in the cat-qubit code space, the rate of bit-flip errors is exponentially suppressed relative to the rate of phase-flip errors.

Biased noise can be extremely useful in the context of quantum error correction. To further suppress errors, cat qubits can be concatenated with other quantum error correcting codes. If bit-flip errors are already adequately suppressed, the outer code needs only to correct phase-flip errors. A simple repetition code is already sufficient for this purpose. If some residual bit-flip errors remain to be corrected, more hardware-efficient variants of the surface code can be employed [125, 192, 193], and biased noise can lead to greatly improved error correction thresholds for such codes [192–194]. As a result, cat qubits provide a practical path towards low-overhead fault-tolerant quantum computation [123–125, 193, 195].

These appealing features can only be exploited, however, if two additional criteria are met. First, the system must remain stabilized in the cat-qubit code space even in the presence of errors or other external perturbations. The exponential suppression of bit-flip errors is only guaranteed within the code space, so if some perturbation pushes the system outside of the code space, the noise bias is liable to disappear. Second, the gates performed on cat qubits must preserve the noise bias. The cat qubit's noise channel can become unbiased if gates are applied that can convert phase-flip errors to bit-flip errors, thus *bias-preserving* gates are required. We describe how these two criteria can be satisfied below.

### Stabilization of cat qubits

One way to stabilize a bosonic mode within the cat-qubit code space is through the use of engineered dissipation. Suppose that a bosonic mode can be engineered to undergo evolution according to the master equation

$$\dot{\rho} = \kappa_2 \mathcal{D}[a^2 - \alpha^2](\rho), \tag{5.37}$$

where $\dot{\rho} = d\rho/dt$, and

$$\mathcal{D}[L]\rho \equiv L\rho L^\dagger - \frac{1}{2}(L^\dagger L\rho + \rho L^\dagger L). \tag{5.38}$$

Because $a^2 |\psi\rangle = \alpha^2 |\psi\rangle$ for any state $|\psi\rangle$ in the cat code subspace, this code space is a steady state of the above dissipative dynamics. Indeed, if errors push the system outside of the code space, the dissipation $\mathcal{D}[a^2 - \alpha^2]$ will bring the system back, thereby stabilizing the cat qubit.

Physically, the dissipator $\mathcal{D}[a^2 - \alpha^2]$ corresponds to a situation where the bosonic mode is subject to a two-photon drive and two-photon losses,

$$\mathcal{D}[a^2 - \alpha^2](\rho) = -i[\epsilon_2 a^{\dagger 2} + \epsilon_2^* a^2, \rho] + \kappa_2 D[a^2]\rho, \tag{5.39}$$

where $\varepsilon_2 \equiv i\alpha^2\kappa_2/2$. One convenient way [178, 189, 196, 197] to realize such two-photon processes is to engineer a nonlinear interaction of the form $(g_2^* a_2 b^\dagger + \text{H.c.})$. This interaction converts two excitations from the "storage" mode $a$ into a single excitation in an ancillary "buffer" mode $b$. Let us suppose that this buffer mode is strongly coupled to its environment, such that it experiences single-excitation losses at a rate $\kappa_b \gg g_2$. In this case, the buffer mode can be adiabatically eliminated (see Appendix B), and the system behaves as if the storage mode loses pairs of excitations directly. More precisely, in the presence of strong loss in the buffer mode, unitary dynamics generated by the Hamiltonian

$$H = g_2 a^2 b^\dagger + \epsilon_d b^\dagger + \text{H.c.}$$

induce effective dissipative dynamics for the storage mode of the form

$$\kappa_2 \mathcal{D}[a^2 - \alpha^2], \tag{5.40}$$

where $\kappa_2 \approx 4|g_2|^2/\kappa_b$ and $\alpha^2 = -\varepsilon_d/g_2$ (we provide more detailed derivations below). Note that the above Hamiltonian contains a linear drive on the buffer mode with amplitude $\epsilon_d$; this drive results in the effective two-excitation drive on the storage mode, owing to the nonlinear coupling between the two. A number of recent experiments have successfully demonstrated the stabilization of cat qubits following this approach [178, 196, 197], with Ref. [197] observing the resultant exponential suppression of bit-flip errors.

An alternate paradigm for cat-qubit stabilization involves using a strong Kerr nonlinearity in conjunction with two-excitation driving. In this "Kerr cat" approach [124, 193, 198–200], one engineers a Hamiltonian of the form

$$\begin{aligned} H &= -Ka^{\dagger 2}a^2 + \epsilon_2 a^{\dagger 2} + \epsilon_2^* a^2 \\ &= -K(a^{\dagger 2} - \alpha^{*2})(a^2 - \alpha^2) + \frac{|\epsilon_2|^2}{K}, \end{aligned} \tag{5.41}$$

where $\alpha = \sqrt{\epsilon_2/K}$ in this case. The coherent states $|\pm\alpha\rangle$ are degenerate eigenstates of this Hamiltonian, and these states are separated from the rest of Hilbert space by an energy gap $\propto 4K|\alpha|^2$. This large energy gap prevents resonant transitions from the degenerate cat subspace to the rest of Hilbert space, thereby stabilizing the cat qubit. While in this thesis we focus on the aforementioned "dissipative cat" approach rather than the Kerr cat approach, we note that both approaches constitute promising paths towards low-overhead fault-tolerant quantum computation [125, 193, 195].

### Bias-preserving gates

As previously mentioned, it is crucial that the gates performed on cat qubits be bias-preserving. That is, the implementation of the gate must not convert phase-flip errors into bit-flip errors or vice versa. Otherwise, the application of gates would un-bias the noise channel. There are some operations which are trivially bias preserving. For example, $Z$ rotations or controlled-$Z$ rotations are trivially bias preserving because these operations commute with phase-flip errors (i.e., Pauli $Z$ errors). State preparations and measurements in the $X$ basis are also bias preserving because bit-flip errors (i.e. Pauli $X$ errors) act trivially on these operations.

Unfortunately, though, these are the only bias-preserving operations that exist for qubits supported in physical two-level systems. Naively, one might expect that some other operations, such as CNOT, can also be implemented in a bias-preserving manner. After all, phase-flip errors are only mapped to other phase-flip errors under conjugation by CNOT. In practice, however, any non-trivial unitary operation such as CNOT must be implemented via evolution under some Hamiltonian for a finite time, and phase errors which occur *during* this operation are liable to propagate into bit-flip errors [201]. For example, a phase-flip error which occurs during a rotation about a qubit's $X$ axis generally propagates to a combination of bit-flip and phase-flip errors. In fact, there is a no-go theorem which states that, for physical two-level systems, a CNOT gate cannot be obtained via finite-time Hamiltonian evolution in a bias-preserving manner [123]. The same is true for the Toffoli gate. Such limitations on the allowed set of bias-preserving gates potentially undermine the promise of hardware-efficient fault tolerance because additional concatenation schemes are then required to build a universal gate set [201, 202].

Remarkably, when biased noise qubits are implemented using the infinite-dimensional Hilbert space of a bosonic mode, as opposed to some physical two-level system, it is possible to circumvent this no-go theorem. Indeed, with cat qubits, it is possible to perform a bias-preserving CNOT and Toffoli gates [123, 124, 199]. The existence of a bias-preserving CNOT and Toffoli at the physical level allows one to greatly reduce the required overhead for error correction. For example, Ref. [124] shows how a bias-preserving CNOT enables one to greatly simplify the fault-tolerant gadgets used in the concatenated schemes of Refs. [201, 202]. Similarly, Ref. [123] shows that bias-preserving CNOTs and Toffolis enable a universal gate set to be obtained at the level of a repetition code [123], i.e. without additional concatenation. These additional bias-preserving gates are thus crucial to reducing the overhead of fault-tolerant computation.

Let us describe how a bias-preserving CNOT gate can be implemented within the paradigm of dissipative cat qubits [123]. (Note that Ref. [124] describes how to implement a bias-preserving CNOT for Kerr cats.) The idea is to engineer a two-mode system which evolves according to the master equation

$$\dot{\rho} = \mathcal{D}[L_1](\rho) + \mathcal{D}[L_2(t)](\rho), \tag{5.42}$$

where the jump operators are given by

$$L_1 = a_1^2 - \alpha^2 \tag{5.43}$$

$$L_2(t) = a_2^2 - \frac{1}{2}\alpha(a_1 + \alpha) + \frac{1}{2}\alpha e^{2i\frac{\pi}{T}t}(a_1 - \alpha). \tag{5.44}$$

We can see that these dissipative dynamics enact a CNOT operation as follows. First, note that $L_1$ simply serves to stabilize mode 1 into a cat state, while the non-trivial dynamics are induced by $L_2(t)$. Consider the case where mode 1 is prepared in the state $|{+\alpha}\rangle$. For this initial state, we can replace the operators

$$(a_1 + \alpha) \to 2\alpha \tag{5.45}$$

$$(a_1 + \alpha) \to 0 \tag{5.46}$$

so that

$$L_2(t) \to a_2^2 - \alpha^2. \tag{5.47}$$

Thus, when mode 1 is prepared in $|{+\alpha}\rangle$, the jump operator $L_2(t)$ simply serves to stabilize mode 2 into a cat state but otherwise does nothing. In contrast, if mode 1 is prepared in $|-\alpha\rangle$ we can make the replacements

$$(a_1 + \alpha) \to 0 \tag{5.48}$$

$$(a_1 + \alpha) \to -2\alpha \tag{5.49}$$

so that

$$L_2(t) \to a_2^2 - \alpha^2 e^{2i\frac{\pi}{T}t}. \tag{5.50}$$

This dissipator acts non-trivially on mode 2. The dissipation

$$\mathcal{D}[a_2^2 - \alpha^2 e^{2i\frac{\pi}{T}t}]$$

stabilizes mode 2 to one of the instantaneous steady states, $|\pm\alpha e^{i\frac{\pi}{T}t}\rangle$. In the limit of large $T$, these dynamics will adiabatically drag the initial states $|\pm\alpha\rangle$ along to these instantaneous steady states. In particular, after a time $t = T$, the system will evolve from $|\pm\alpha\rangle$ to $|\mp\alpha\rangle$, which constitutes a bit flip. Therefore, the dissipative dynamics generated by $\mathcal{D}[L_1]$ and $\mathcal{D}[L_2(t)]$ does indeed enact a CNOT gate. Crucially, this is a bias-preserving implementation because the instantaneous steady states $|\pm\alpha e^{i\frac{\pi}{T}t}\rangle$ always remain well-separated from one another in phase space during the course of the gate. As a result, the probability of population transfer between the two states (i.e. of a bit-flip error), remains exponentially suppressed. A bias-preserving Toffoli gate can be implemented using a similar approach.

### 5.3.2 Stabilized cat qubits in multimode cQAD

Our proposal [125] for a multimode cQAD cat-qubit architecture is illustrated in Fig. 5.8. Phononic resonators[6] constitute the storage modes that support the cat

[6] We consider phononic resonators, as opposed to BAW or SAW resonators, to avoid the challenges described in Section 5.2.2 and because of phononic resonators' long lifetimes.

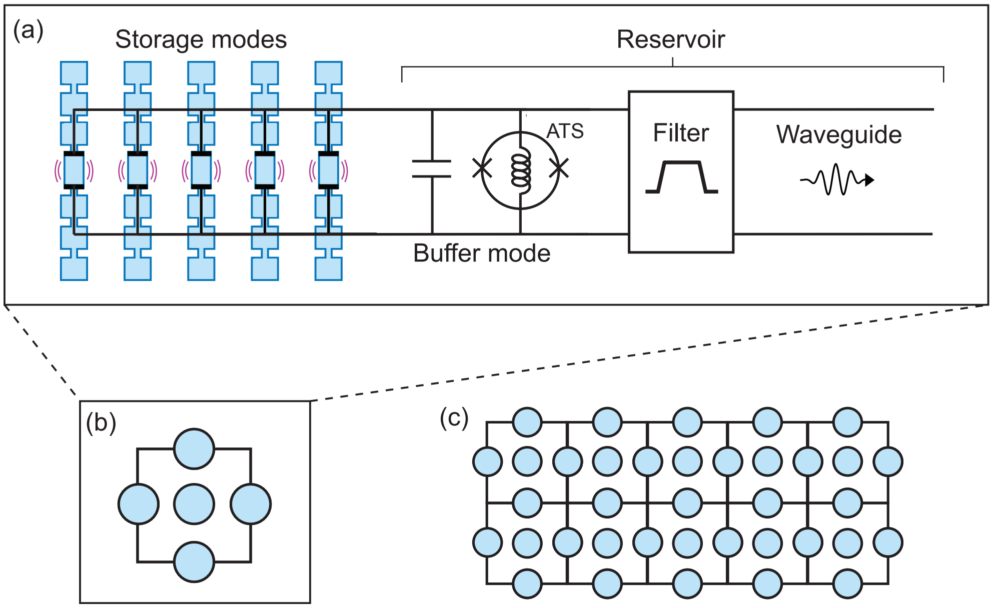


Figure 5.8: Multimode cQAD cat-qubit architecture (adapted from Ref. [125]). (a) Unit cell. A collection of phononic resonators couples to a reservoir that consists of a nonlinear buffer mode, filter, and waveguide. This single unit cell may be represented schematically as in (b) and tiled in one or two dimensions as in (c) in order to scale.

qubits. A collection of phononic resonators is coupled to a *reservoir*, which is responsible for stabilizing the phononic modes into cat states. The reservoir itself consists of three components: a nonlinear buffer mode, a filter, and a waveguide.

The buffer mode consists of a nonlinear circuit element called an Asymmetrically Threaded Squid (ATS) shunted by a capacitor, following the approach of Ref. [197]. With appropriate flux bias, the potential energy of the ATS is approximately

$$-2E_J\epsilon(t)\sin(\phi), \tag{5.51}$$

where $E_J$ is the Josephson energy, $\epsilon(t)$ is a flux pump, and $\phi$ is the superconducting phase across the ATS. This phase contains contributions from both the buffer and storage modes,

$$\phi = \varphi_b b + \sum_n \varphi_{a_n} a + \text{H.c.}, \tag{5.52}$$

where the vacuum fluctuation amplitudes $\varphi_{b,a_n}$ quantify the respective contributions of the buffer and storage modes to the total phase across the ATS. The $\sin(\phi)$ nonlinearity contains the requisite $(a_n^2 b^\dagger + \text{H.c.})$ interactions, and these interactions may be brought on resonance by pumping the system at specific frequencies (we describe this process in detail in Section 5.3.3). We note that, in comparison to the $E_J \cos(\phi)$ potential energy of a single junction (as in a transmon), the $E_J \sin(\phi)$ potential of the ATS is advantageous in this context because it does not contain deleterious cross-Kerr terms (e.g., $a^\dagger a b^\dagger b$) that can limit the stabilization [196, 203].

The buffer mode, in turn, is coupled to a waveguide through a bandpass filter. The coupling to the waveguide serves to imbue the buffer mode with a large single-excitation loss rate $\kappa_b$, as is required for the dissipative stabilization scheme described above. The bandpass filter plays a dual role: it not only serves to shield the storage modes from direct single-excitation losses into the waveguide, but it also suppresses crosstalk among the storage modes, as described further in Section 5.3.5.

In our architecture, each reservoir is responsible for stabilizing a small collection of storage modes into cat states. Ultimately, the number of modes that can be stabilized by a single reservoir is limited by crosstalk, so the number of reservoirs must be increased as the architecture is scaled. Fig. 5.8(b,c) illustrates how the architecture can be scaled up by taking a single unit cell—one reservoir and its associated storage modes—and tiling this cell in a one- or two-dimensional grid. The cat qubits can then be concatenated into a repetition or surface code. (We note that the use of a five-mode unit cell—as opposed to a four-mode unit cell—enables a more convenient readout scheme for the cat qubits, as described in Ref. [125]. The additional mode does not need to be stabilized in a cat state to facilitate readout.)

In the remainder of this section, we highlight two features of this architecture that are crucial to its scalability. The first is multiplexed stabilization, i.e. the ability to stabilize multiple storage modes into cat states using only a single nonlinear element. This multiplexed stabilization simultaneously reduces hardware complexity and improves connectivity. The second feature is crosstalk mitigation. We find that all dominant sources of crosstalk in the architecture can be suppressed through a careful combination of filtering and phonon mode frequency optimization. We provide high-level summaries of these two features below, with more detailed technical analyses given in Sections 5.3.3 to 5.3.6.

### Summary: Multiplexed stabilization

In our architecture, each reservoir is responsible for stabilizing multiple storage modes simultaneously. This multimode stabilization can be implemented via a simple extension of the single-mode stabilization scheme demonstrated in Ref. [197]. The main idea is to use *frequency-division multiplexing* to stabilize different modes independently. Here, multiplexing refers to the fact that different regions of the filter passband are allocated to the stabilization of different modes. When the bandwidth allocated to each stabilization process is sufficiently large, multiple modes can be stabilized simultaneously and independently, as we now show.

To stabilize the $n$-th mode coupled to a given reservoir, we apply a pump frequency $\omega_p^{(n)} = 2\omega_a - \omega_b + \Delta_n$, and drive the buffer mode at frequency $\omega_d^{(n)} = \omega_b - \Delta_n$, where $\Delta_n$ denotes a detuning. Analogously to the single-mode stabilization case, due to the nonlinear mixing of the ATS these pumps and drives give rise to an interaction Hamiltonian of the form

$$H = \sum_n g_2 \left(a_n^2 - \alpha^2\right) b^\dagger e^{i\Delta_n t} + \text{H.c.} \tag{5.53}$$

See Section 5.3.3 for a derivation of Eq. (5.53) as well as Eqs. (5.54) and (5.55) below. The sum does not run over all modes coupled to the ATS, but rather only over the modes stabilized by that ATS. In our architecture, though five modes couple to each

ATS, only two must be stabilized simultaneously, so the sum contains only two terms. [The other modes are stabilized by adjacent ATS's, see Fig. 5.8(c).] By adiabatically eliminating the lossy buffer mode, one obtains an effective master equation describing the evolution of the storage modes

$$\frac{d\rho}{dt} \approx \mathcal{D}\left[\sum_n \sqrt{\kappa_{2,n}}(a_n^2 - \alpha^2)e^{i\Delta_n t}\right]\rho(t), \tag{5.54}$$

where $\kappa_{2,n} \approx 4|g_2|^2/\kappa_b$ if the corresponding detuning falls inside the filter passband ($|\Delta_n| < 2J$), and $\kappa_{2,n} \approx 0$ otherwise, see Section 5.3.3. If the detunings are chosen such that $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$ for all $m \neq n$, then Eq. (5.54) can be approximated by

$$\frac{d\rho}{dt} \approx \sum_n \kappa_{2,n}\mathcal{D}\left[a_n^2 - \alpha^2\right]\rho(t), \tag{5.55}$$

which is obtained by neglecting the fast-rotating terms in (5.54) via a rotating-wave approximation. The dynamics (5.55) stabilize cat states in different modes independently and simultaneously. Thus, by simply applying additional pumps and drives with appropriately chosen detunings, multiple modes can be simultaneously stabilized by a single ATS.

The efficacy of this multiplexed stabilization scheme can be understood intuitively by considering the frequencies of photons that leak from the buffer mode to the filtered bath. In the case of $\Delta_n = 0$, a pump applied at frequency $2\omega_a - \omega_b$ facilitates the conversion of two phonons of frequency $\omega_a$ to a single photon of frequency $\omega_b$. As a result, photons that leak from the buffer to the bath have frequency $\omega_b$. If instead the pump is detuned by an amount $\Delta_n \neq 0$, it follows from energy conservation that the corresponding emitted photons have frequency $\omega_b + \Delta_n$. When the differences in these emitted photon frequencies, $\Delta_n - \Delta_m$, are chosen to be much larger than the emitted photon linewidths, $4|\alpha|^2\kappa_2$, emitted photons associated with different storage modes are spectrally resolvable by the environment. Therefore, when the stabilization of mode $n$ causes a photon to leak to the environment, there is no back-action on modes $m \neq n$. These ideas are illustrated pictorially in Fig. 5.9(a). The figure emphasizes an important additional point: the emitted photon frequencies must lie inside the filter bandwidth, lest the engineered dissipation be suppressed by the filter.

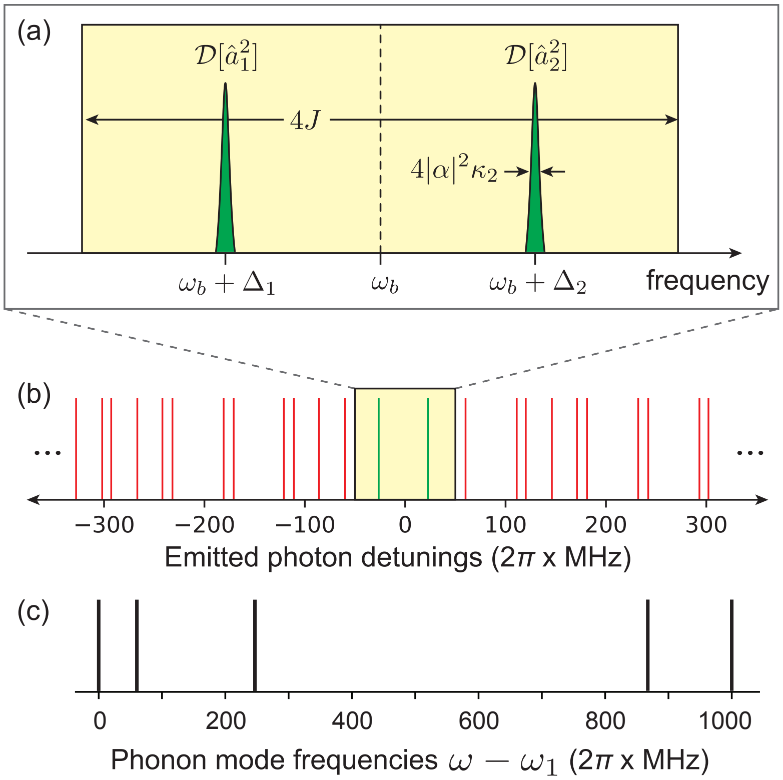

(a)
$\mathcal{D}[\hat{a}_1^2]$
$\mathcal{D}[\hat{a}_2^2]$
$4J$
$4|\alpha|^2\kappa_2$
$\omega_b + \Delta_1$
$\omega_b$
$\omega_b + \Delta_2$
frequency
(b)
−300
−200
−100
0
100
200
300
Emitted photon detunings (2π x MHz)
(c)
0
200
400
600
800
1000
Phonon mode frequencies $\omega - \omega_1$ (2π x MHz)

Figure 5.9: Multiplexed stabilization and crosstalk mitigation. (a) Frequency multiplexing. Because the desired couplings ($g_2 a_n^2 b^\dagger e^{i\Delta_i t}$+H.c.) are detuned by different amounts, photons lost to the environment via the buffer have different frequencies. When the corresponding emitted photons (green lines) are spectrally well resolved, $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$, the modes are stabilized independently. Dissipation associated with photon emissions at frequencies inside the filter passband (yellow box) is strong, while dissipation associated with emission at frequencies outside the passband is suppressed. (b),(c) Crosstalk suppression. Red lines in (b) denote photon emission frequencies associated with various correlated errors, calculated for the specific phonon mode frequencies plotted in (c). The mode frequencies are deliberately chosen so that *all* emissions associated with correlated errors occur at frequencies outside the filter passband (no red lines fall in the yellow box). In other words, Eqs. (5.61) and (5.62) are simultaneously satisfied for any choices of the indices that lead to nontrivial errors in the cat qubits. See Section 5.3.3 for further details.

### Summary: crosstalk mitigation

In acting as a nonlinear mixing element, the ATS not only mediates the desired ($g_2 a_n^2 b^\dagger$+H.c.) interactions, but it also mediates spurious interactions between different storage modes. We now describe how such interactions can give rise to crosstalk among the cat qubits, and subsequently how this crosstalk can be mitigated through a combination of filtering and phonon-mode frequency optimization.

While most spurious interactions mediated by the ATS are far detuned and can be safely neglected in the rotating-wave approximation, there are others which cannot be neglected. Most concerning among these are interactions of the form

$$g_2 a_j a_k b^\dagger e^{i\delta_{ijk}t} + \text{H.c.}, \tag{5.56}$$

for $j \neq k$, where $\delta_{ijk} = \omega_p^{(i)} - \omega_j - \omega_k + \omega_b$. This interaction converts two phonons from different modes, $j$ and $k$, into a single buffer mode photon, facilitated by the pump that stabilizes mode $i$. These interactions cannot be neglected in general because they have the same coupling strength as the desired interactions (5.53), and they can potentially be resonant or near-resonant, depending on the frequencies of the phonon modes involved.

There are three different mechanisms through which the interactions (5.56) can induce crosstalk among the cat qubits. These mechanisms are described in detail in Section 5.3.4, and we summarize them here. First, analogously to how the desired interactions (5.53) lead to two-phonon losses, the undesired interactions (5.56) lead to correlated, single-phonon losses

$$\kappa_{\text{eff}} \mathcal{D}[a_j a_k] \to \kappa_{\text{eff}} |\alpha|^4 D[Z_j Z_k] \tag{5.57}$$

where the rate $\kappa_{\text{eff}}$ will be discussed shortly. The arrow denotes projection onto the code space, illustrating that these correlated losses manifest as *stochastic*, correlated phase errors in the cat qubits.

Second, the interplay between different interactions of the form (5.56) gives rise to new effective dynamics [204–206] generated by Hamiltonians of the form

$$\begin{aligned} H_{\text{eff}} =& \chi a_i^\dagger a_j^\dagger a_m a_n e^{i(\delta_{\ell mn} - \delta_{ijk})t} + \text{H.c.}, && (5.58) \\ \to& \chi |\alpha|^4 Z_i Z_j Z_k Z_l e^{i(\delta_{\ell mn} - \delta_{ijk})t} + \text{H.c.}, && (5.59) \end{aligned}$$

where the coupling rate $\chi$ is defined in Section 5.3.4. The projection onto the code space in the second line reveals that $H_{\text{eff}}$ can induce undesired, *coherent* evolution within the code space.

Third, $H_{\text{eff}}$ can also evolve the system out of the code space, changing the phonon-number parity of one or more modes in the process. Though the engineered dissipation subsequently returns the system to the code space, it does not correct changes to the phonon-number parity. The net result is that $H_{\text{eff}}$ also induces *stochastic*, correlated phase errors in the cat qubits,

$$\gamma_{\text{eff}} \mathcal{D}[Z_i Z_j Z_k Z_\ell], \tag{5.60}$$

where the rate $\gamma_{\text{eff}}$ will be discussed shortly.

Remarkably, all of these types of crosstalk can be suppressed through a combination of filtering and phonon-mode frequency optimization. In Section 5.3.5, we show that both $\kappa_{\text{eff}} \approx 0$ and $\gamma_{\text{eff}} \approx 0$, provided

$$|\delta_{ijk}| > 2J, \tag{5.61}$$

$$|\delta_{ijk} - \delta_{\ell mn}| > 2J, \tag{5.62}$$

respectively. This suppression can be understood as follows. The decoherence associated with $\kappa_{\text{eff}}$ and $\gamma_{\text{eff}}$ results from the emission of photons at frequencies $\omega_b + \delta_{ijk}$ and $\omega_b \pm (\delta_{ijk} - \delta_{\ell mn})$, respectively. When the frequencies of these emitted photons lie outside the filter passband, their emission (and the associated decoherence) is suppressed. Crucially, we can arrange for all such errors to be suppressed *simultaneously* by carefully choosing the frequencies of the phonon modes, as shown in Fig. 5.9(b,c). The configuration of mode frequencies in Fig. 5.9(c) was found via a numerical optimization procedure described in Section 5.3.6. The optimization also accounts for the undesired coherent evolution (5.59): the detunings $\delta_{ijk} - \delta_{\ell mn}$ are maximized so that $H_{\text{eff}}$ is rapidly rotating and its damaging effects are mitigated (this suppression is quantified in Section 5.3.6). Additionally, we note that in Fig. 5.9(b) all emitted photon frequencies associated with crosstalk lie at least 10 MHz outside of the filter passband. As a result, the crosstalk suppression is robust to variations in the phonon mode frequencies of the same order. Larger variations in the phonon mode frequencies can be accommodated by reducing the filter bandwidth.

We have demonstrated that crosstalk can be largely suppressed within the five-mode unit cells of our architecture. It is tempting to consider whether more modes could be added to each unit cell to improve hardware efficiency or connectivity, but we find that crosstalk is a limiting factor in this regard. As more modes are added, the number of undesired terms (5.56) grows combinatorially, increasing the total number of constraints, Eqs. (5.61) and (5.62). At the same time, the filter bandwidth must be increased to accommodate the stabilization of additional modes, making each constraint more challenging to satisfy. Thus, it rapidly becomes difficult or impossible to satisfy all constraints, and crosstalk can become significant. We have accordingly chosen five modes per unit cell because this is the maximum number consistent with our 2D square grid layout for which all crosstalk constraints can be satisfied. While frequency crowding and bandwidth constraints are characteristic of multimode architectures generally [99, 103, 166], resonators with additional terminals, or tunable couplers [207, 208], could be employed in future designs to further suppress crosstalk and increase the number of modes per unit cell.

### 5.3.3 Multiplexed stabilization of cat qubits

In this section, we provide a detailed analysis of our multiplexed stabilization scheme. We note that we frequently employ *adiabatic elimination* as a tool to analyze dissipative dynamics throughout this section. We perform this adiabatic elimination using the effective operator approach of Ref. [206], which is summarized in Appendix B.

We consider a collection of $N$ storage modes mutually coupled to a common reservoir. For the moment, we take reservoir to be a capacitively-shunted ATS (buffer resonator) with a large decay rate. The Hamiltonian of the system is

$$H = H_d + \omega_b b^\dagger b + \sum_{n=1}^{N} \omega_n a_n^\dagger a_n - 2E_J \epsilon_p(t) \sin\left(\phi_b + \sum_{n=1}^{N} \phi_n\right),$$

where $H_d$ is a driving term (defined below), $a_n$ ($b$) is the annihilation operator for the $n$-th storage mode (buffer mode) with frequency $\omega_n$ ($\omega_b$), and $\phi_n = \varphi_n(a_n + a_n^\dagger)$ is the phase across the ATS due to mode $n$, with vacuum fluctuation amplitudes $\varphi_n$. To stabilize multiple storage modes simultaneously, we apply separate pump and drive tones for each mode. Explicitly,

$$\epsilon_p(t) = \sum_n \epsilon_p^{(n)} \cos\left(\omega_p^{(n)} t\right), \tag{5.63}$$

and

$$H_d = \sum_n \left(\epsilon_d^{(n)} b\, e^{i\omega_d^{(n)} t} + \text{H.c.}\right). \tag{5.64}$$

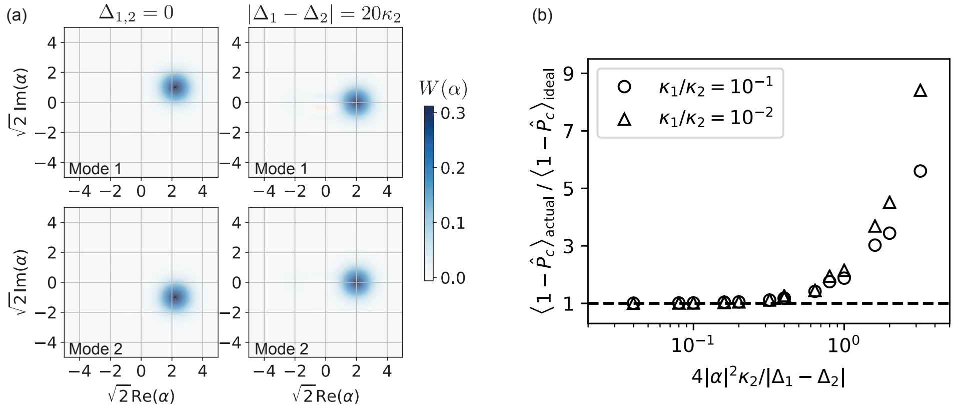


Figure 5.10: Multiplexed stabilization. (a) Comparison of stabilization for $\Delta_n = 0$ and $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$. Wigner plots are shown of two storage modes after evolution under the master equation $\dot{\rho} = -i[H, \rho] + \kappa_b \mathcal{D}[b]$, with $H$ given by (5.68). The storage modes are initialized in a product state $|\beta_1\rangle\,|\beta_2\rangle$ that does not lie in the code space but which is a steady state of (5.73). Thus, when $\Delta_n = 0$ (left plots), the evolution is (approximately) trivial. The left two plots thus also serve as Wigner plots of the initial state $|\beta_1\rangle\,|\beta_2\rangle$. However, when $|\Delta_1 - \Delta_2| \gg 4|\alpha|^2\kappa_2$ (right plots), the system evolves to the code space, defined here by $\alpha = \sqrt{2}$. (b) Validity of approximating Eq. (5.73) by Eq. (5.75). Master equations (5.73,5.75) are simulated (with decoherence added to each mode via the dissipators $\kappa_1\mathcal{D}[a]$ and $\kappa_1\mathcal{D}[a^\dagger a]$), and the expectation value of $1 - P_c$ is computed once the system reaches its steady state. Here $P_c$ denotes the projector onto the cat code space, and the subscripts "actual" and "ideal" denote expectation with respect to the steady states of (5.73) and (5.75), respectively. The ratio of expectations, plotted on the vertical axis, quantifies the relative increase in population outside the code space. A ratio $\sim 1$ indicates the approximation works well. Parameters are chosen from the ranges $|\alpha|^2 \in [1, 4]$ and $|\Delta_1 - \Delta_2|/\kappa_2 \in [5, 100]$.

We choose the frequencies of the $n$-th pump and drive tones, respectively, as

$$\omega_p^{(n)} = 2\omega_n - \omega_b + \Delta_n, \tag{5.65}$$

$$\omega_d^{(n)} = \omega_b - \Delta_n, \tag{5.66}$$

where $\Delta_n$ denote detunings whose importance will be made clear shortly.

To proceed, we expand the sine to third order and move to the frame where each mode rotates at its respective frequency. The resultant Hamiltonian is

$$\begin{aligned} H \approx & \sum_n \left( \epsilon_d^{(n)} b\, e^{-i\Delta_n t} + \text{H.c.} \right) \\ & - 2E_J \epsilon_p(t) \left[ \varphi_b b\, e^{-i\omega_b t} + \sum_n \varphi_n a_n\, e^{-i\omega_n t} + \text{H.c.} \right] \\ & + \frac{E_J}{3} \epsilon_p(t) \left[ \varphi_b b\, e^{-i\omega_b t} + \sum_n \varphi_n a_n\, e^{-i\omega_n t} + \text{H.c.} \right]^3 \end{aligned} \tag{5.67}$$

This Hamiltonian contains terms that lead to the required two-photon dissipators for each storage mode,

$$\sum_n \left[ g_{2,n} \left( a_n^2 - \alpha_n^2 \right) b^\dagger e^{i\Delta_n t} + \text{H.c.} \right], \tag{5.68}$$

with

$$g_{2,n} = E_J \epsilon_p^{(n)} \varphi_n^2 \varphi_b / 2, \tag{5.69}$$

$$\alpha_n^2 = - \left( \epsilon_d^{(n)} \right)^* / g_{2,n}. \tag{5.70}$$

However, the Hamiltonian (5.67) contains numerous other terms. While many of these other terms are fast-rotating and can be neglected in the rotating wave approximation (RWA), others can have non-trivial effects. For example, the interplay between the terms in the second and third lines of (5.67) gives rise to effective frequency shifts (a.c. Stark shifts) of the buffer and storage modes, which modify the resonance conditions (5.65) and (5.66). One can calculate the magnitudes of these shifts (and hence compensate for them) by applying the effective operator approach of Refs. [204, 205], in which case the Stark shifts are given by the coefficients of the $b^\dagger b$ and $a^\dagger a$ terms that arise in the effective Hamiltonian. Alternatively, the shifts can be calculated by moving to a displaced frame with respect to the linear terms on the second line of (5.67), as is done in Ref. [197]. The Hamiltonian (5.67) also contains terms which lead to crosstalk, but we defer the discussion of these terms to the next section. For now, we keep only the desired terms (5.68).

We proceed by adiabatically eliminating the lossy buffer mode $b$, following the approach described in Appendix B. Specifically, we designate the the ground subspace as the subspace where the buffer mode is in the vacuum state, and the excited subspace as the subspace where the buffer mode contains at least one excitation. We find that the effective dynamics of the storage modes within the ground subspace are described by the master equation

$$\dot{\rho} = -i[H_{\text{eff}}, \rho] + \mathcal{D}\left[\sum_n \frac{g_{2,n}}{\Delta_n - i\kappa_b/2}\left(a_n^2 - \alpha_n^2\right)e^{i\Delta_n t}\right](\rho), \tag{5.71}$$

where

$$\begin{aligned} H_{\text{eff}} = -\frac{1}{2}\sum_{m,n}\Bigg\{ & g_{2,n}^* g_{2,m}(a_n^2 - \alpha_n^2)^\dagger (a_m^2 - \alpha_m^2) \\ & \times \left[\frac{1}{\Delta_m - i\kappa_b/2} + \frac{1}{\Delta_n + i\kappa_b/2}\right] e^{i(\Delta_m - \Delta_n)t}\Bigg\}. \end{aligned} \tag{5.72}$$

To understand these dynamics, let us first consider the simple case where $\Delta_n = 0$. The above master equation reduces to

$$\dot{\rho} = \kappa_2 D\left[\sum_n \left(a_n^2 - \alpha_n^2\right)\right](\rho), \tag{5.73}$$

where $\kappa_2 = 4|g_2|^2/\kappa_b$. Any product of coherent states

$$|\beta_1\rangle \otimes |\beta_2\rangle \otimes \ldots \otimes |\beta_N\rangle \tag{5.74}$$

that satisfies $\sum_n \beta_n^2 = \sum_n \alpha_n^2$ is a steady state of (5.73). The subspace of steady states includes states in the code space, for which $\beta_n^2 = \alpha_n^2$, but it also includes states outside of the code space. Because a strictly larger space is stabilized, when noise pushes the system outside of the code space, the stabilization is not guaranteed to return the system to the code space. The coherent dissipation in Eq. (5.73) is thus not sufficient for our purposes.

Consider instead the case where the detunings are chosen to be distinct, satisfying $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$. In this limit, we can drop the now fast-rotating cross terms in the dissipator in Eq. (5.71), and the effective master equation becomes

$$\dot{\rho} = \sum_n \kappa_{2,n} D\left[a_n^2 - \alpha_n^2\right](\rho), \tag{5.75}$$

where

$$\kappa_{2,n} = \frac{\kappa_b |g_{2,n}|^2}{\Delta_n^2 + \kappa_b^2/4}. \tag{5.76}$$

The incoherent dissipator eq. (5.75) stabilizes cat states in each mode, as desired. Thus, by simply detuning the pumps and drives used to stabilize each mode, multiple modes can be stabilized simultaneously and independently by a single ATS.

Two remarks about the approximation of Eq. (5.73) by Eq. (5.75) are necessary. First, the condition $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$ can be derived by expressing the operators in Eq. (5.73) in the displaced Fock basis [125]. Roughly speaking, the condition dictates that $|\Delta_n - \Delta_m|$ be much larger than the rate at which photons are lost from the stabilized modes. Second, we have neglected $H_{\text{eff}}$; the rotating terms in $H_{\text{eff}}$ can be dropped in the RWA in the considered limit, and the non-rotating terms provide an additional source of stabilization [198] that we neglect for simplicity. It is also worth noting that the two-photon dissipation rate, $\kappa_{2,n}$, decreases monotonically with $\Delta_n$. To avoid significant suppression of this engineered dissipation, one can choose $\Delta_n \lesssim \kappa_b$ so that $\kappa_{2,n}$ remains comparable to $\kappa_2$, or alternatively one can exploit the filtering procedure described in Section 5.3.5 which enables strong effective dissipation even for $\Delta_n > \kappa_b$.

We demonstrate our scheme for multiplexed stabilization numerically in Fig. 5.10. Through master equation simulations, we observe good stabilization for $|\Delta_1 - \Delta_2| \gg$

$4|\alpha|^2\kappa_2$, but not $\Delta_{1,2} = 0$, as expected. Moreover, we also quantify the validity of approximating Eq. (5.73) by Eq. (5.75). Strictly speaking, the approximation is valid only in the regime $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$, but we find that even for $|\Delta_n - \Delta_m| \sim 4|\alpha|^2\kappa_2$ the stabilization works reasonably well, by which we mean that the population that leaks out of the code space is comparable for the two dissipators (5.73) and (5.75), see Fig. 5.10(b). The approximation breaks down beyond this point, and accounting for the additional terms in Eq. (5.73) becomes increasingly important.

We conclude this section by providing some physical intuition as to why detuning the pumps and drives allows one to stabilize multiple cat qubits simultaneously. When $\Delta_n = 0$, photons lost from different storage modes via the buffer cannot be distinguished by the environment. As a result, we obtain a single coherent dissipator $L \propto \sum_n (a_n^2 - \alpha_n^2)$. When distinct detunings are chosen for each mode, however, photons lost from different modes via the buffer are emitted at different frequencies. When these photons are spectrally resolvable, the environment can distinguish them, resulting in a collection of independent, incoherent dissipators $L_n \propto (a_n^2 - \alpha_n^2)$ instead. The emitted photon linewidth is $4|\alpha|^2\kappa_2$, which can be seen by expressing $\kappa_2\mathcal{D}[a^2 - \alpha^2]$ in the displaced Fock basis, as described in Ref. [125]. Thus, the emitted photons are well-resolved when $|\Delta_n - \Delta_m| \gg 4|\alpha|^2\kappa_2$, which is the same condition assumed in the derivation of (5.75).

### 5.3.4 Sources of crosstalk

In this Section we describe how undesired terms in the Hamiltonian (5.67) lead to crosstalk among modes coupled to the same ATS. In particular, we show that these undesired terms lead to effective dissipators and effective Hamiltonians that can cause correlated phase errors in the cat qubits.

The dominant sources of crosstalk are undesired terms in the Hamiltonian (5.67) of the form

$$g_2\, a_i a_j b^\dagger e^{i\delta_{ijk} t} + \text{H.c.}, \tag{5.77}$$

where

$$\delta_{ijk} = \omega_k^{(p)} - \omega_i - \omega_j + \omega_b, \tag{5.78}$$

and we have neglected the dependence of $g_2$ on the indices $i, j$ for simplicity. In contrast to the other undesired terms in (5.67), these terms have the potential to induce large crosstalk errors because they both (i) have coupling strengths comparable to the desired terms (5.68), and (ii) can be resonant or near-resonant. In particular, the undesired term is resonant ($\delta_{ijk} = 0$) for $2\omega_k + \Delta_k = \omega_i + \omega_j$. This resonance condition can be satisfied, for example, when the storage modes have near uniformly-spaced frequencies.

These unwanted terms may not be exactly resonant in practice, but we cannot generally guarantee that they will be rotating fast enough to be neglected in the RWA either. In contrast, all other undesired terms in (5.67) are detuned by at least $\min_n |\omega_n - \omega_b|$, which is on the order of $\sim 2\pi \times 1\,\text{GHz}$ for the parameters considered in this work. We therefore focus on crosstalk errors induced by the terms (5.77).

The terms (5.77) can lead to three different types of correlated errors:

- Type I: Stochastic errors induced by effective dissipators
- Type II: Stochastic errors induced by effective Hamiltonians
- Type III: Coherent errors induced by effective Hamiltonians

We describe each type of error in turn. Without mitigation (see Section 5.3.5), these correlated phase errors could be a significant impediment to performing high-fidelity operations.

### Type I: stochastic errors induced by effective dissipators

The terms (5.77) can lead to correlated photon losses at rates comparable to $\kappa_2$, resulting in significant correlated phase errors in the cat qubits. These deleterious effects manifest when one adiabatically eliminates the buffer mode. Explicitly, we apply the effective operator formalism described in Appendix B to the operators

$$H^{(1)} = g_2\, a_i a_j b^\dagger e^{i\delta_{ijk}t} + \text{H.c.}, \tag{5.79}$$

$$L^{(1)} = \sqrt{\kappa_b}\, b \tag{5.80}$$

and obtain the effective operators

$$H^{(1)}_{\text{eff}} = -\frac{|g_2|^2 \delta_{ijk}}{\delta_{ijk}^2 + \kappa_b^2/4}(a_i a_j)^\dagger (a_i a_j) + \text{H.c.}, \tag{5.81}$$

$$L^{(1)}_{\text{eff}} = \frac{g_2\sqrt{\kappa_b}}{\delta_{ijk} - i\kappa_b/2} a_i a_j e^{i\delta_{ijk}t}. \tag{5.82}$$

The effective Hamiltonian preserves phonon-number parity and thus does not induce phase flips. The effective jump operator $L_{\text{eff}}$ describes correlated single-phonon losses in modes $i$ and $j$ at a rate

$$\kappa_{\text{eff}} = \frac{\kappa_b |g_2|^2}{\delta_{ijk}^2 + \kappa_b^2/4} \tag{5.83}$$

which is comparable to $\kappa_2$ for $\delta_{ijk} \lesssim \kappa_b$. These correlated single photon losses induce correlated phase flips in the cat qubits, which can be seen by projecting $L_{\text{eff}}$ into the code space,

$$L^{(1)}_{\text{eff}} \to \sqrt{\kappa_{\text{eff}}}\, \alpha^2 Z_i Z_j e^{i\delta_{ijk}t}. \tag{5.84}$$

### Type II: stochastic errors induced by effective Hamiltonians

The interplay between different terms of the form (5.77) can lead to further correlated errors. As an example, consider the operators

$$H^{(2)} = g_2\, a_i a_j b^\dagger e^{i\delta_{ijk}t} + g_2\, a_\ell a_m b^\dagger e^{i\delta_{\ell mn}t} + \text{H.c.}, \tag{5.85}$$

$$L^{(2)} = \sqrt{\kappa_b}\, b. \tag{5.86}$$

Adiabatically eliminating the buffer mode yields,

$$\begin{aligned} H^{(2)}_{\text{eff}} &= \left[\chi(a_i a_j)^\dagger (a_\ell a_m) e^{i(\delta_{\ell mn}-\delta_{ijk})t} + \text{H.c.}\right] + \ldots, \\ L^{(2)}_{\text{eff}} &= \frac{g_2\sqrt{\kappa_b}}{\delta_{ijk} - i\kappa_b/2} a_i a_j e^{i\delta_{ijk}t} + \frac{g_2\sqrt{\kappa_b}}{\delta_{\ell mn} - i\kappa_b/2} a_\ell a_m e^{i\delta_{\ell mn}t}. \end{aligned} \tag{5.87}$$

where

$$\chi = -\frac{|g_2|^2}{2}\left[\frac{1}{\delta_{ijk} - i\kappa_b/2} + \frac{1}{\delta_{\ell mn} + i\kappa_b/2}\right]$$

and "…" denotes additional terms in the effective Hamiltonian that preserve phonon-number parity. Note that the effective dissipator $L^{(2)}_{\text{eff}}$ leads to Type I correlated phase errors. Indeed, for sufficiently large $|\delta_{ijk} - \delta_{\ell mn}|$, the action of $L^{(2)}_{\text{eff}}$ can be approximated by replacing it with two independent dissipators of the form (5.82).

What is different about this example is that the effective Hamiltonian $H^{(2)}_{\text{eff}}$ contains terms $\propto (a_i a_j)^\dagger (a_\ell a_m)$ that generally do not preserve phonon-number parity. Such terms can unitarily evolve the system out of the code space, changing the parity in the process. In turn, the engineered dissipation returns the system to the code space, but it does so without changing the parity. Therefore, the net effect of such excursions out of the code space and back is to induce *stochastic* parity-flips in the storage modes, which manifest as correlated phase errors on the cat qubits. The errors are stochastic even though the evolution generated by $H^{(2)}_{\text{eff}}$ is unitary because the stabilization itself is stochastic. Specifically, the errors are of the form $\mathcal{D}[Z_i Z_j Z_\ell Z_m]$, which one can show by adiabatically eliminating the excited states of the storage modes (see Ref. [125] for details).

**Type III: coherent errors induced by effective Hamiltonians**

The parity-non-preserving effective Hamiltonian $H_{\text{eff}}^{(2)}$ also induces non-trivial coherent evolution within the code space. This can be seen by projecting $H_{\text{eff}}^{(2)}$ into the code space

$$H_{\text{eff}}^{(2)} \to (|\alpha|^4 \chi Z_i Z_j Z_\ell Z_m e^{i(\delta_{\ell mn} - \delta_{ijk})t} + \text{H.c.}). \tag{5.88}$$

This undesired evolution does not decohere the system but can nevertheless degrade the fidelity of operations. See further discussion in Section 5.3.5.

### 5.3.5 Crosstalk mitigation: filtering

In this subsection, we show how Type I and Type II crosstalk errors can be suppressed by placing a bandpass filter at the output port of the buffer mode. The purpose of the filter is to allow photons of only certain frequencies to leak out of the buffer, such that the desired engineered dissipation remains strong but spurious dissipative processes are suppressed. A crucial requirement of this approach is that the desired dissipative processes be spectrally resolvable from the undesired ones, and we show that adequate spectral resolution is achievable in the next section (Section 5.3.6).

We begin by providing a quantum mechanical model of a bandpass filter [209, 210]. While a detailed classical analysis of the filter is given in [125], here we employ a complementary quantum model. The quantum model not only allows us to study the filter's effects numerically via master equation simulations, but it is also sufficiently simple so as to enable a straightforward analytical treatment via the effective operator formalism described in Appendix B.

Motivated by the filter designs described in Refs. [125, 210], we employ a tight-

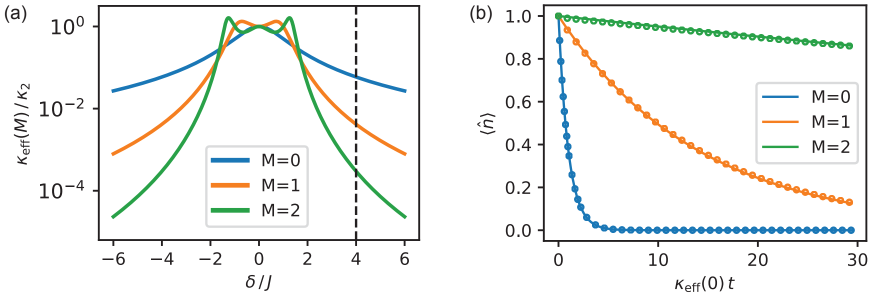


Figure 5.11: Suppression of Type I errors. (a) Plots of $\kappa_{\text{eff}}(M)$ as a function of the detuning, $\delta$, of the unwanted term. (b) Master equation simulations. The system is initialized with a single excitation in the storage mode and evolved according to the dynamics $\dot{\rho} = -i[(g_2 ab^\dagger e^{i\delta t} + \text{H.c.}) + H_{\text{buffer+filter}}, \rho] + \mathcal{D}[L^{(3)}](\rho)$. These dynamics are analogous to those generated by $H^{(3)}$ and $L^{(3)}$; in both cases the unwanted term induces losses at rates $\kappa_{\text{eff}}(M)$. Simulation results are indicated by open circles, and the analytical expressions for $\kappa_{\text{eff}}(M)$ are plotted as solid lines. Parameters: $\alpha = \sqrt{2}, \kappa_c/g_2 = 10, J/g_2 = 5$. For (b), $\delta = 4J$, as indicated by the dashed line in (a).

binding model where the filter consists of a linear chain of $M$ bosonic modes with annihilation operators $c_i$, and each with the same frequency $\omega_b$. Modes in the chain are resonantly coupled to their nearest neighbors with strength $J$. The first mode in the chain couples to the buffer mode $b$, which is no longer coupled directly to the open waveguide. Instead, the $M$-th mode is now the one which couples strongly to the waveguide, such that its single-photon loss rate is given by $\kappa_c$. The buffer-filter system is described by the Hamiltonian (in the rotating frame)

$$H_{\text{buffer+filter}} = J(c_1^\dagger b + c_1 b^\dagger) + \sum_{i=1}^{M-1} J(c_{i+1}^\dagger c_i + c_{i+1} c_i^\dagger), \tag{5.89}$$

together with the dissipator $\kappa_c \mathcal{D}[c_M]$. We show below that these additional modes act as a bandpass filter, with center frequency $\omega_b$ and bandwidth $4J$, and they suppresses the emission of photons with frequencies outside of this passband.

### Suppression of Type I errors

To illustrate the suppression of Type I errors, we consider the operators

$$H^{(3)} = \big(g_2\, a_i a_j b^\dagger e^{i\delta_{ijk}t} + \text{H.c.}\big) + H_{\text{buffer+filter}}, \tag{5.90}$$

$$L^{(3)} = \sqrt{\kappa_c}\, c_M \tag{5.91}$$

where the first term in $H^{(3)}$ is the same as the unwanted term $H^{(1)}$ from Section 5.3.4. We adiabatically eliminate *both the buffer and filter modes* in order to obtain an effective dynamics for only the storage modes. We note that adiabatically eliminating the buffer and filter modes together is not fundamentally different from adiabatically eliminating the buffer; both calculations are straightforward applications of the methods in Appendix B. We obtain the effective dissipator

$$L^{(3)}_{\text{eff}} = \sqrt{\kappa_{\text{eff}}(M)}\, a_i a_j e^{i\delta_{ijk}t} \tag{5.92}$$

where the rates for the first few values of $M$ are

$$\kappa_{\text{eff}}(0) = \frac{\kappa_c |g_2|^2}{\delta_{ijk}^2 + \kappa_c^2/4} \approx \kappa_c \frac{|g_2|^2}{\delta_{ijk}^2} \tag{5.93}$$

$$\kappa_{\text{eff}}(1) = \frac{\kappa_c |g_2|^2 J^2}{(J^2 - \delta_{ijk}^2)^2 + \delta_{ijk}^2\kappa_c^2/4} \approx \kappa_{\text{eff}}(0)\left(\frac{J}{\delta_{ijk}}\right)^2 \tag{5.94}$$

$$\kappa_{\text{eff}}(2) = \frac{\kappa_c |g_2|^2 J^4}{(2J^2\delta_{ijk} - \delta_{ijk}^3)^2 + (J^2 - \delta_{ijk}^2)^2\kappa_c^2/4} \approx \kappa_{\text{eff}}(0)\left(\frac{J}{\delta_{ijk}}\right)^4, \tag{5.95}$$

where the approximations assume that $\delta_{ijk} \gg J, \kappa_c$. In this regime, $\kappa_{\text{eff}}(M)$ is exponentially suppressed with increasing $M$ via the factor $(J/\delta_{ijk})^{2M}$.

We plot these rates as a function of $\delta_{ijk}$ in Fig. 5.11(a), where the exponential suppression of the decoherence rates outside the filter band is evident. Fig. 5.11(b) shows the results of analogous master equation simulations; good quantitative agreement with the analytical expressions is observed. Thus we conclude that Type I errors are indeed suppressed by the filter, provided $|\delta_{ijk}| > 2J$.

### Suppression of Type II errors

To illustrate the suppression of Type II errors, we construct a simple toy model that both captures the relevant physics and is easy to study numerically. Consider the operators

$$H^{(4)} = \big(g\, ab^\dagger e^{i\delta_1 t} + g\, b^\dagger e^{i\delta_2 t} + \text{H.c.}\big) + \big[g_2(a^2 - \alpha^2)b^\dagger + \text{H.c.}\big] + H_{\text{buffer+filter}} \tag{5.96}$$

$$L^{(4)} = \sqrt{\kappa_c}\, c_M. \tag{5.97}$$

where $a$ is the annihilation operator for the single storage mode that we consider in this model. In this toy model, the terms in parentheses in$H^{(4)}$ should be understood as analogous to $H^{(2)}$. Indeed we obtain the former from the latter by replacing $a_i a_j \to a$ and $a_\ell a_m \to 1$.

Adiabatically eliminating the buffer and filter modes yields the effective operators

$$H^{(4)}_{\text{eff}} = \big[\chi_{\text{eff}}(M)\, a\, e^{i(\delta_1 - \delta_2)t} + \text{H.c.}\big] + \ldots \tag{5.98}$$

$$L^{(4)}_{\text{eff}} = \sqrt{\kappa^{(\delta_1)}_{\text{eff}}(M)}\, a\, e^{i\delta_1 t} + \sqrt{\kappa^{(0)}_{\text{eff}}(M)}(a^2 - \alpha^2). \tag{5.99}$$

Here, "…" denotes a parity-preserving term ($\propto a^\dagger a$) that we neglect, $\kappa^{(\delta)}_{\text{eff}}(M)$ denotes the effective loss rate [eqs. (5.93) to (5.95)] with the replacement $\delta_{ijk} \to \delta$, and

$$\chi_{\text{eff}}(M) \approx -\frac{|g|^2}{2}\left(\frac{1}{\delta_1} + \frac{1}{\delta_2}\right) \tag{5.100}$$

is independent of $M$ in the limit $\delta_{1,2} \gg J, \kappa_b$. The first term in $L^{(4)}_{\text{eff}}$ gives rise to the Type I errors that are suppressed by the filter, as already discussed. Our present interest is the Type II errors induced by the interplay of $H^{(4)}_{\text{eff}}$, the stabilization, and the filter.

Unfortunately, the effective operators $H_{\text{eff}}^{(4)}$ and $L_{\text{eff}}^{(4)}$ do not properly capture this interplay. In particular, it follows from energy conservation that Type II errors induced by $H_{\text{eff}}^{(4)}$ result in photon emissions at frequency $\omega_b + \delta_2 - \delta_1$. Intuitively, such emissions should be exponentially suppressed when this frequency lies outside the filter band. However, this suppression is not apparent in the operators $H_{\text{eff}}^{(4)}, L_{\text{eff}}^{(4)}$ because, in the course of deriving $H_{\text{eff}}^{(4)}$, we already eliminated the filter. After adiabatic elimination the only vestige of the filter is the term $\sqrt{\kappa_{\text{eff}}^{(0)}(M)}(a^2 - \alpha^2)$, which embodies the behavior of the filter at frequency $\omega_b$, *but not at frequency* $\omega_b + \delta_2 - \delta_1$. As such, proceeding to calculate the Type II error rate from these operators is not valid, and an alternate approach is required.

In order to properly capture the subtle interplay between the effective Hamiltonian, the stabilization, and filter, we defer adiabatic elimination and instead begin by calculating an effective Hamiltonian that describes the time-averaged dynamics generated by $H^{(4)}$. We restrict our attention to a regime where the terms in parentheses in Eq. (5.96) are rapidly rotating, so that evolution generated by $H^{(4)}$ is well approximated by its time average. We calculate the time-averaged effective Hamiltonian $\bar{H}^{(4)}$ following the approach described in Refs. [204, 205],

$$\begin{aligned}\bar{H}^{(4)} &= \left[g_2(a^2 - \alpha^2)b^\dagger + \text{H.c.}\right] + H_{\text{buffer+filter}} \\ &- \frac{|g|^2}{2}\left(\frac{1}{\delta_1} + \frac{1}{\delta_2}\right)\left(2b^\dagger b + 1\right)\left(ae^{i(\delta_1 - \delta_2)t} + \text{H.c.}\right)\end{aligned} \tag{5.101}$$

where we have neglected a parity-preserving term ($\propto a^\dagger a$), and terms rotating at the fast frequencies $\delta_{1,2}$. Notice that

$$\bar{H}^{(4)} \approx \left[g_2(a^2 - \alpha^2)b^\dagger + \text{H.c.}\right] + H_{\text{buffer+filter}} + H_{\text{eff}}^{(4)}, \tag{5.102}$$

where the approximation is obtained by preemptively replacing $b^\dagger b$ with its expected value of 0. Doing so reveals that $H_{\text{eff}}^{(4)}$ can be understood as arising from the time-averaged dynamics of the the unwanted terms in $H^{(4)}$ in the limit of large $\delta_{1,2}$. In effect, time averaging provides a way of introducing $H_{\text{eff}}^{(4)}$ into the dynamics without having to eliminate the filter, thereby allowing us to study the interplay of the filter and effective Hamiltoninan.

We proceed by taking the operators $\bar{H}^{(4)}$ and $L^{(4)}$ and adiabatically eliminating the buffer, the filter, and all excited states of the storage mode, i.e. all states that do not lie in the code space. Adiabatically eliminating the storage mode excited states is valid in the regime where the engineered dissipation is strong relative to couplings that excite the storage mode ($H_{\text{eff}}^{(4)}$ in this case), such that these excited states are barely populated. We obtain

$$\bar{H}_{\text{eff}}^{(4)} = \chi_{\text{eff}}(M)\,\alpha Z\,e^{i(\delta_1-\delta_2)t} + \text{H.c.}, \tag{5.103}$$

$$\bar{L}_{\text{eff}}^{(4)} = \sqrt{\gamma_{\text{eff}}(M)}Z. \tag{5.104}$$

The rates for the first few values of $M$ are

$$\gamma_{\text{eff}}(0) = \frac{4\kappa_c|2g_2\alpha\,\chi_{\text{eff}}(0)|^2}{4\left(|2g_2\alpha|^2-\delta_{12}^2\right)^2+\delta_{12}^2\kappa_c^2}, \tag{5.105}$$

$$\begin{aligned}\gamma_{\text{eff}}(1) &= \frac{4J^2\kappa_c|2g_2\alpha\,\chi_{\text{eff}}(1)|^2}{4\delta_{12}^2\left(J^2+|2g_2\alpha|^2-\delta_{12}^2\right)^2+\left(|2g_2\alpha|^2-\delta_{12}^2\right)^2\kappa_c^2}\\ &\approx \gamma_{\text{eff}}(0)\left(\frac{J}{\delta_{12}}\right)^2,\end{aligned} \tag{5.106}$$

$$\begin{aligned}\gamma_{\text{eff}}(2) &= \frac{4J^4\kappa_c|2g_2\alpha\,\chi_{\text{eff}}(2)|^2}{4\left(|2g_2\alpha|^2(J-\delta_{12})(J+\delta_{12})+\delta_{12}^4-2J^2\delta_{12}^2\right)^2+\delta_{12}^2\left(|2g_2\alpha|^2+J^2-\delta_{12}^2\right)^2\kappa_c^2}\\ &\approx \gamma_{\text{eff}}(0)\left(\frac{J}{\delta_{12}}\right)^4,\end{aligned} \tag{5.107}$$

where we have used the shorthand $\delta_{12}\equiv\delta_1-\delta_2$ to simplify the expressions, and the approximations are obtained in the in the limit of large $|\delta_1-\delta_2|$. In this limit, we

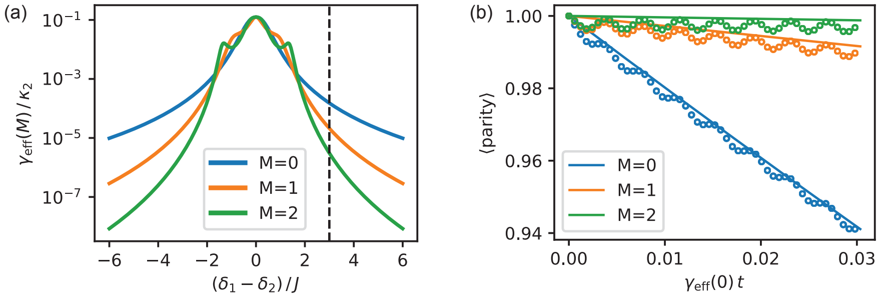


Figure 5.12: Suppression of Type II errors. (a) Plots of $\gamma_{\rm eff}(M)$ as a function of the detuning, $\delta_1-\delta_2$, of the effective Hamiltonian. (b) Master equation simulations. The storage mode is initialized in the even parity cat state and evolved according to the dynamics $\dot{\rho} = -i[\bar{H}^{(4)}, \rho] + \mathcal{D}[L^{(4)}](\rho)$. Simulation results are indicated by open circles, and the analytical expressions for $\gamma_{\rm eff}(M)$ are plotted as solid lines. Parameters: $\alpha = \sqrt{2}, \kappa_c/g_2 = 10, J/g_2 = 5$. Rather than specify values for $g$ and $\delta_{1,2}$, we simply fix $\chi_{\rm eff}(M)/g_2 = 0.2$. For (b), $\delta = 3J$, as indicated by the dashed line in (a).

find that the phase flip rate is exponentially suppressed by the filter,

$$\gamma_{\rm eff}(M) \approx \gamma_{\rm eff}(0)\left(\frac{J}{\delta_1-\delta_2}\right)^{2M}, \tag{5.108}$$

as expected.

We plot the rates $\gamma_{\rm eff}(M)$ as a function of $\delta_1-\delta_2$ in Fig. 5.12(a), where the exponential suppression of the decoherence rates outside the filter band is again evident. Fig. 5.12(b) shows the results of corresponding master equation simulations. Good quantitative agreement with the analytical expressions is observed. (Note that the small parity oscillations in the simulation results are Type III errors—coherent micro-oscillations due to evolution generated by the effective Hamiltonian within the code space. These errors are not suppressed by the filter.) Thus we find that Type II errors are also suppressed by the filter, provided the effective Hamiltonian detuning lies outside the filter passband.

### 5.3.6 Crosstalk mitigation: mode frequency optimization

We have shown that stochastic correlated phase errors (Types I and II) can be suppressed by a filter if the corresponding emitted photons have frequencies outside the filter passband. We now show that it is possible to suppress *all* such errors simultaneously by carefully choosing the frequencies of the phonon modes. In doing so, the effects of Type III errors can also be simultaneously minimized, but the specifics of this minimization will depend on other architectural choices, such as whether the cat qubits are concatenated with a repetition or surface code. For the sake of brevity and simplicity, we thus focus only on the suppression of Type I and II errors, and we refer the interested reader to Ref. [125] for further details on the suppression of Type III errors.

To minimize the effects of Type I and II errors, we define a binary cost function, $C$, that quantifies these errors as a function of the phonon mode frequencies $\omega_n$ and the pump detunings $\Delta_n$. Intuitively, $C$ should be large if any emitted photons associated with Type I and II errors lie inside the filter's bandwidth $4J$. We thus take $C = 1$ if any of the following conditions are met:

- $|\delta_{ijk}| < 2J$ (Type I errors not suppressed)
- $|\delta_{ijk} - \delta_{\ell mn}| < 2J$ (Type II errors not suppressed)
- $|\delta_{iii}| > 2J$ (desired dissipation suppressed)

In other words, we set $C = 1$ if any Type I or II errors are not suppressed by the filter, or if any of the desired engineered dissipation is suppressed by the filter. Otherwise, in the ideal situation where all crosstalk errors are suppressed by the filter but the desired engineered dissipation is not, we take $C = 0$.

It is important to note that this cost function depends both on how many modes are coupled to an ATS, and on how many modes are stabilized by that ATS. This is because the total number of different photon emission frequencies depends on both the total number of phonon modes and on the number of pump tones applied to the ATS. In particular, if out of the $N$ phonon modes coupled to an ATS, only $M < N$ need to be stabilized, then only $M$ pump tones are needed. Fig. 5.8(c) provides an example of a situation where only a subset of the modes coupled to an ATS need to be stabilized. The figure shows how our architecture can be scaled by tiling multiple unit cells in a two-dimensional grid layout, where each unit cell consists of an ATS and five phonon modes to which the ATS couples. In this layout, each phonon mode is simultaneously coupled to two ATS's. However, only one ATS is required to stabilize any given phonon mode. Thus, the responsibility for stabilizing the phonon modes can be shared among the different ATS's, such that each ATS need only stabilize at most two out of the five modes to which it couples (see Ref. [125] for further details). In accordance with this example, we assume that only two out of the five phonon modes need to be stabilized by the ATS in the optimization below.

Having defined the cost function $C$, we perform a numerical search for the values of the mode frequencies and pump detunings which minimize the cost. In performing this optimization, we place two additional restrictions on allowed frequencies and detunings. First, we restrict the mode frequencies to lie within a 1 GHz bandwidth. This is done because the modes are supported by phononic crystal resonators, and as such all mode frequencies must lie within the phononic bandgap. These bandgaps are typically not more than 1 GHz wide for the devices we consider [158]. Second, we restrict the values of the detunings to $\Delta = \pm J$. This is done to maximize use of the filter bandwidth; emitted photons are detuned from one another by $2J$ and from the nearest band edge by $J$.

The optimization results are illustrated in Fig. 5.13. We find that $C = 0$ for the optimal configuration, indicating that *all* Type I and Type II errors are simultaneously suppressed by the filter. Additionally, all emitted photon frequencies associated with Type I or II errors lie at least 10 MHz outside the filter passband. As a result, the

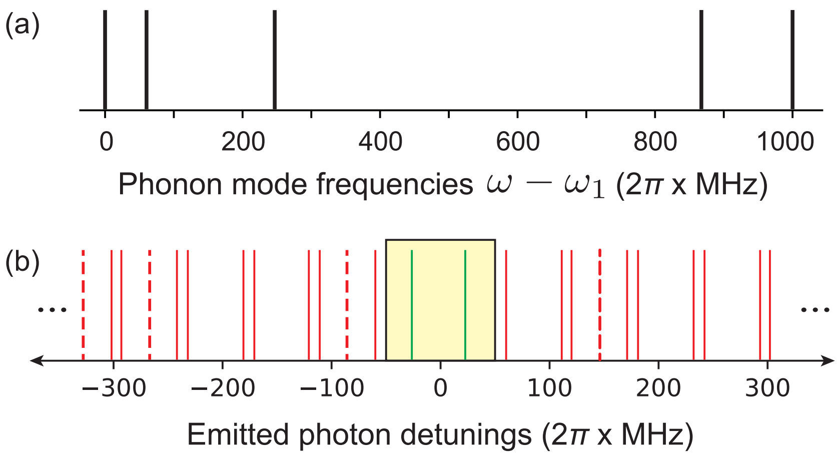


Figure 5.13: Optimized mode frequencies. (a) Plot of the optimized frequencies of the five phonon modes. (b) Emitted photon detunings. Red dashed (solid) lines indicate photons emitted via parity-non-preserving Type I (Type II) processes. The yellow box covers the region $[-50, 50]$ ($2\pi \times$ MHz), representing a bandpass filter with center frequency $\omega_b$ and a $4J = 2\pi \times 100$ MHz passband. The fact that no red lines lie inside the yellow box indicates that all Type I and II processes are sufficiently far detuned so as to be suppressed by the filter.

optimized configuration is robust to deviations in the mode frequencies of the same order, and larger deviations can be tolerated by decreasing the filter bandwidth. Moreover, as described in Ref. [125], Type III errors area also strongly suppressed in these configurations. Therefore, all dominant sources of crosstalk are strongly suppressed.

## 5.4 Hardware-efficient QRAM architectures with quantum acoustics

In this section, we describe how a QRAM can be constructed using the cQAD architectures described in Sections 5.2 and 5.3. The resulting QRAM implementations are naturally hardware efficient and scalable, thanks to the compact size and long lifetime of the acoustic modes. Furthermore, the proposed implementations are well-suited for near-term experiments. Indeed, per our proposals, a small-scale QRAM could

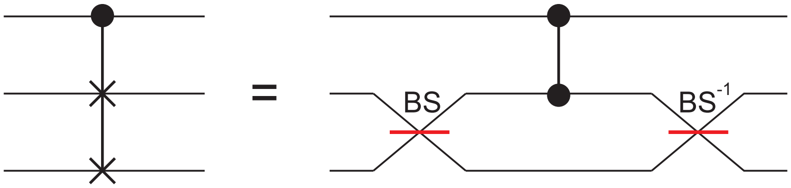


Figure 5.14: A controlled-SWAP gate can be implemented using a combination of beamsplitter and phase-shift operations [211].

already be implemented on a single chip.

First, in Section 5.4.1, we describe how the architectures from Sections 5.2 and 5.3 can be used to implement quantum routers. Then, in Section 5.4.2, we describe how these routers can be integrated to build a QRAM.

### 5.4.1 Constructing quantum routers from acoustic modes

As described in Chapter 2, the fundamental gate operation required to implement a quantum router is a controlled-SWAP gate. Below, we describe how this operation can be implemented in the architectures of Sections 5.2 and 5.3.

We begin with the architecture of Section 5.2. For this architecture, the native operations are beamsplitters (generated by Hamiltonians of the form $H \propto a_1 a_2^\dagger + \text{H.c.}$) and CZ gates (generated by Hamiltonians of the form $H \propto a_1 a_2 a_3^\dagger + \text{H.c.}$). These operations can be combined to implement a controlled-SWAP gate [211], as illustrated in Fig. 5.14. The behavior of the circuit can be understood as follows. The initial 50-50 beam-splitter maps the modes $a_1$ and $a_2$ to the linear combinations

$$\begin{pmatrix} a_1 \\ \\ a_2 \end{pmatrix} \to U^\dagger \begin{pmatrix} a_1 \\ \\ a_2 \end{pmatrix} U = \begin{pmatrix} \frac{1}{\sqrt{2}}(a_1 + i a_2) \\ \\ \frac{1}{\sqrt{2}}(a_2 + i a_1), \end{pmatrix} \tag{5.109}$$

where $U = \exp[i\pi/4(a_1^\dagger a_2 + a_2^\dagger a_1)]$ is the beamsplitter unitary. If the control mode is in the vaccum state, the CZ gate acts trivially, and the subsequent inverse beamsplitter acts as

$$\begin{pmatrix} \frac{1}{\sqrt{2}}(a_1 + ia_2) \\ \\ \frac{1}{\sqrt{2}}(a_2 + ia_1), \end{pmatrix} \to U \begin{pmatrix} \frac{1}{\sqrt{2}}(a_1 + ia_2) \\ \\ \frac{1}{\sqrt{2}}(a_2 + ia_1), \end{pmatrix} U^\dagger = \begin{pmatrix} a_1 \\ \\ a_2 \end{pmatrix} \tag{5.110}$$

so the gate acts trivially when the control is $|0\rangle$. In contrast, if the control is $|1\rangle$, the CZ gate applies a $-1$ phase to the $a_1$ mode before the second beamsplitter operation. Incorporating this phase, the final state of the system is,

$$\begin{pmatrix} \frac{1}{\sqrt{2}}(-a_1 + ia_2) \\ \\ \frac{1}{\sqrt{2}}(a_2 - ia_1), \end{pmatrix} \to U \begin{pmatrix} \frac{1}{\sqrt{2}}(-a_1 + ia_2) \\ \\ \frac{1}{\sqrt{2}}(a_2 - ia_1), \end{pmatrix} U^\dagger = \begin{pmatrix} ia_2 \\ \\ -ia_1 \end{pmatrix} \tag{5.111}$$

so the modes $a_1$ and $a_2$ are swapped (up to local phases).

The implementation of Fig. 5.14 is not precisely equivalent to a controlled-SWAP gate, but in the context of QRAM the differences are inconsequential. For example, the $\pm i$ phases that arise when the swap occurs are subsequently cancelled by a corresponding inverse operation that occurs later on in the QRAM circuit. Similarly, the controlled-SWAP implementation of Fig. 5.14 only performs the correct operation in the subspace with $< 2$ excitations in the modes to be swapped. This is because our CZ gate implementation only works properly within the single excitation subspace, and the system can leave this subspace if both modes to be swapped are excited. Indeed, the initial beamsplitter operation can place the two excitations in the same mode, as in the Hong-Ou-Mandel effect [212] (note that this fact clearly demonstrates that the 50-50 beamsplitter operation is not simply equivalent to a $\sqrt{\text{SWAP}}$ gate acting on the single-excitation subspaces of the two modes). However, in our QRAM implementation, one of the input modes is always in the vaccuum state, so this subtlety is irrelevant.

Next, we describe how a controlled-SWAP gate can be implemented in the cat-

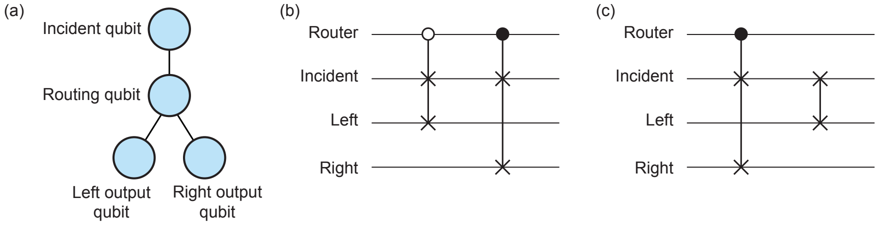
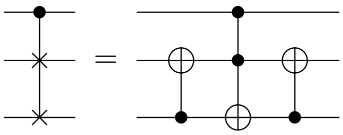


Figure 5.15: A cQAD quantum router. The router consists of four acoustic qubits, illustrated schematically in (a). These qubits are equipped with a quantum routing operation, which can be implemented using either of the circuits in (b) or (c). The circuit in (c) uses one fewer controlled-SWAP gate, which may be advantageous for near-term demonstrations.

qubit architecture of Section 5.3. As described in that section, cat qubits enable the implementation of both CNOT and Toffoli gates in a bias-preserving manner. Combining these operations, a controlled-SWAP gate can be implemented as



(5.112)

If the CNOT and Toffoli gates in this circuit are physical bias-preserving gates, then we have a bias-preserving implementation of controlled-SWAP at the physical level. Alternatively, if error correction is used, the same circuit can be implemented at the logical level. In this context, implementing the non-Clifford Toffoli gate fault-tolerantly requires magic state injection (or some other equivalent fault-tolerant construction). Magic state distillation can be done in a relatively hardware-efficient manner with cat qubits [125].

Assuming access to controlled-SWAP gates, a quantum router can be constructed from acoustic qubits as shown in Fig. Fig. 5.15. We use the term *acoustic qubits* to encompass the various different qubit implementations described in Sections 5.2 and 5.3. An acoustic qubit could be a single phononic mode with information encoded in the single-phonon subspace (Section 5.2); a single phononic mode with information encoded in the cat-qubit code space (Section 5.3); or a logical qubit comprised of many

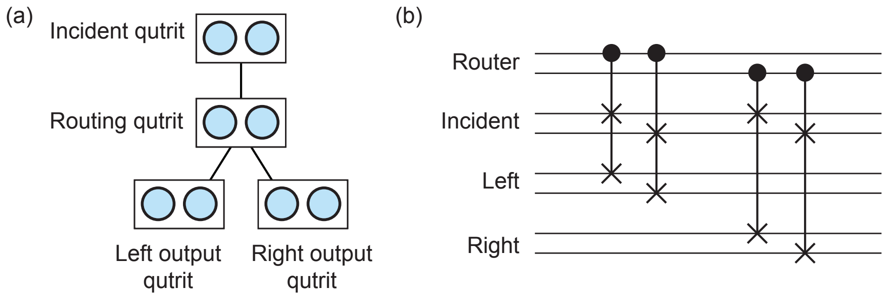


Figure 5.16: A qutrit-based cQAD quantum router. The router consists of four acoustic qutrits, each composed of two acoustic qubits, as illustrated schematically in (a). The routing operation can be implemented with the circuit shown in (b).

physical phononic modes, each encoding a cat qubit (Section 5.3). In its simplest incarnation, the router consists of four acoustic qubits, one of which controls the routing direction, while the other three serve as the router's input and output modes. The routing operation is realized by performing controlled-SWAP gates among these qubits.

As discussed in Chapter 3, QRAM is resilient to noise regardless of whether the routers are implemented using qubits or qutrits. However, the error scaling is somewhat more favorable when qutrits are used. Accordingly, we also propose a qutrit-based cQAD quantum router in Fig. 5.16. Each qutrit consists of two acoustic qubits, with the encoding

$$\text{wait} = |00\rangle \tag{5.113}$$

$$\text{route left} = |10\rangle \tag{5.114}$$

$$\text{route right} = |01\rangle\,. \tag{5.115}$$

This specific encoding of a qutrit into the two-qubit Hilbert space enables a straightforward implementation of the routing operation, as shown in Fig. 5.16(b). In comparison to the qubit-based router of Fig. 5.15, the qutrit-based router has more favorable error propagation properties but a larger hardware overhead.

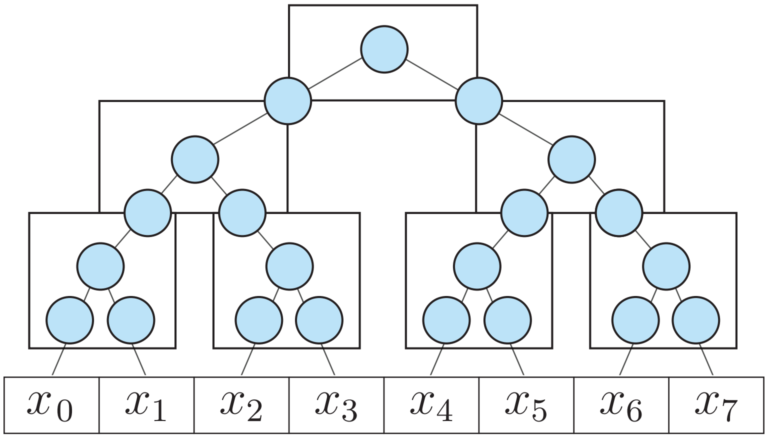


Figure 5.17: A cQAD QRAM. Each box denotes a single quantum router, with the outputs of routers at one level of the tree acting as inputs to the routers at the next level down.

### 5.4.2 A cQAD QRAM implementation

The quantum routers described in the previous section can be assembled together to build a QRAM, as shown in Fig. 5.17. A collection of routers is arranged in a binary tree with the output qubits of routers at one level acting as the input qubits for routers at the next level down. For small- to medium-size QRAMs, the acoustic modes comprising QRAM could all be implemented on a single chip. Alternatively, the QRAM could be constructed out of several physically distinct modules, with each module containing one or more quantum routers. If qubits are encoded within the single-phonon subspaces of phononic resonators, as in Section 5.2, connections between modules could be implemented using pitch-and-catch schemes [172–174, 176, 213]. More generally, connections between modules could be implemented using teleportation [21], such that connected modules need not be physically adjacent to one another. Independent of the implementation details, the proposed cQAD QRAM inherits the appealing properties of high coherence, scalability, and hardware efficiency from the underlying acoustic hardware.

## 5.5 Conclusions and Outlook

We have proposed quantum computing architectures for multimode cQAD and demonstrated how they can be used to implement a QRAM. The proposed implementations are naturally hardware efficient, owing to the compactness of multimode cQAD systems that is enabled by small acoustic wavelengths. We emphasize that hardware efficiency is not only crucial for scaling to large system sizes, but that it is also particularly advantageous for near-term experiments. Indeed, a small-scale QRAM can even be implemented even with just a single multimode resonator. In the long term, the use of bosonic quantum error correcting codes, and cat codes in particular, provides a promising path towards a QRAM implementation that is simultaneously hardware efficient and fault-tolerant.

An important direction for future work will be to precisely quantify the hardware cost of the proposed QRAM architecture, especially in the fault-tolerant regime. That is, what is the hardware cost required to query a memory of size $N$ with a logical query infidelity below some threshold value? It is not unreasonable to expect that the noise resilience of the bucket-brigade QRAM (Chapter 3), together with the low-overhead fault tolerance implementations based on cat qubits, could lead to orders-of-magnitude reduction in hardware cost relative to standard surface code implementations [18]. Despite these significant hardware efficiency improvements, building a QRAM that can address millions or billions of different memory elements is not likely feasible in the foreseeable future. Thus, another important direction for future work will be to identify applications where small- to medium-sized QRAMs can already be useful (e.g. algorithms for simulating local Hamiltonians) and to perform tailored resource estimates for these applications.

# Appendix A

# Copying classical data to the bus

In this Appendix, we explicitly describe various ways in which classical data can be copied into the state of the bus during a QRAM query. Slightly different procedures are required depending on whether the QRAM is implemented with two-level or three-level systems, and whether the QRAM is initialized in a known state or in some arbitrary state.

We begin with the case where the QRAM is implemented with two-level routers, as described in Section 3.3.1. Each router's incident and output modes are also taken to be physical two-level systems. All routers and their respective modes are initialized to $|0\rangle$. For reasons that will become apparent shortly, we suppose that the bus qubit is initialized to $|0\rangle$, but then immediately mapped to $|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$ using a Hadamard gate (see the circuit diagram in Fig. 2.8). During the query, this bus qubit is routed down the tree, to an output mode of some router at the bottom level. At this point, classical data is encoded into the state of the bus qubit by applying classically controlled $Z$ gates, as illustrated in Fig. A.1(a). If the memory element being queried is 1, a $Z$ gate is applied, and the state of the bus is flipped from $|+\rangle$ to $|-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$. If the memory element queried is 0, no $Z$ gate is applied, and the bus remains in $|+\rangle$. In this way, the classical bit is encoded in the $|\pm\rangle$ basis

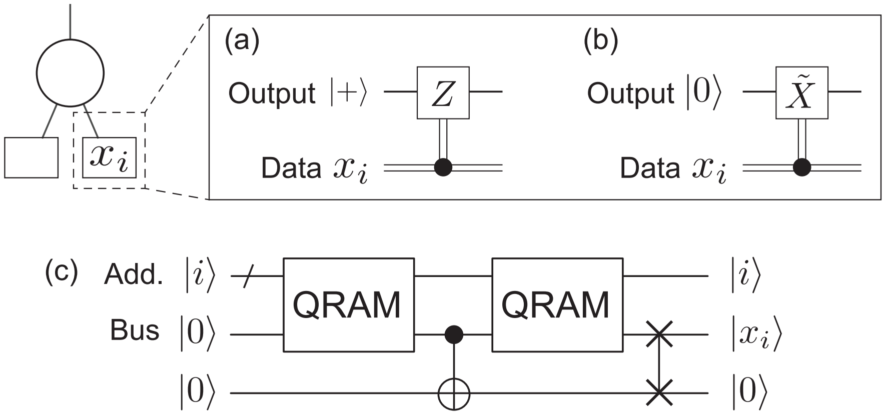


Figure A.1: Circuits for copying classical data. (a) Two-level circuit. The bus qubit is encoded within a physical two-level system and initialized in $|+\rangle$. A $Z$ gate flips the bus to $|-\rangle$ conditioned on the classical data. (b) Three-level circuit. The bus qubit is encoded within a two-level subspace of a physical three-level system and initialized in $|0\rangle$. The $\tilde{X}$ gate (see text) flips the bus to $|1\rangle$ conditioned on the classical data. (c) Query circuit for QRAM initialized in an arbitrary state. The circuits assumes three-level routers, so the bus is initialized in $|0\rangle$ and circuit (b) is employed within each QRAM block to copy data to the bus. An analogous circuit can be constructed for two-level routers. The ancillary qubits comprising the QRAM's routers (not shown) can be initialized in an arbitrary state.

of the bus qubit. Note, however, that because the location of the bus is not known, classically controlled $Z$ gates must be applied to the output modes of all routers at the bottom level of the tree.

This data copying operation has a crucial property, which we call *no extra copying*: in the absence of errors, the copying operation acts trivially on all modes that do not contain the bus qubit. In the above case, all modes that do not contain the bus are in $|0\rangle$, so they are unaffected by the $Z$ gates, hence why we use the $|\pm\rangle$ basis for the bus. The no extra copying property is crucial because it guarantees that the final state of the tree is the same across all good (error-free) branches, as required by the noise-resilience arguments in Chapter 3. Were this property not to hold, the final state of the tree would depend on which element was queried, so the bus would remain entangled with the routers after the query, even in the absence of errors.

Now let us consider the case where the QRAM is implemented with three-level routers, as described in Section 3.2.2. Each router's incident and output modes are taken to be physical three-level systems, whose basis states we also label as $|0\rangle$, $|1\rangle$, and $|W\rangle$. The address and bus qubits are encoded within the $|0,1\rangle$ subspace of such three-level systems. Prior to the query, all routers, as well as their incident and output modes, are initialized to $|W\rangle$, and the bus is initialized to $|0\rangle$. During the query, the bus is routed to an output mode of some router at the bottom level of the tree. Data is copied into the bus by applying classically controlled $\tilde{X}$ gates to the output modes [Fig. A.1(b)], where

$$\tilde{X} = |1\rangle\langle 0| + |0\rangle\langle 1| + |W\rangle\langle W| . \tag{A.1}$$

If the memory element being queried is 1, the $\tilde{X}$ gate is applied, and the state of the bus is flipped from $|0\rangle$ to $|1\rangle$. If the memory element queried is 0, no $\tilde{X}$ gate is applied, and the bus remains in $|0\rangle$. In this way, the classical bit is encoded in the $|0,1\rangle$ basis of the bus qubit (one could also choose to encode the information in the $|\pm\rangle$ basis by constructing an analogous $\tilde{Z}$ gate). Here again, the classically controlled gates must be applied to the output modes of all routers at the bottom of the tree. This operation satisfies the no extra copying property because, in the absence of errors, all modes not containing the bus are in $|W\rangle$, on which $\tilde{X}$ acts trivially.

In order to enforce the no extra copying property, both of the above data copying operations rely on the fact that the routers, as well as their input and output modes, are initialized to some known state. When the QRAM is initialized in an arbitrary state (see Section 3.3.1), however, additional care must be taken to ensure this property still holds. The challenge is that the mode that actually contains the bus must somehow be distinguished from all the other modes, which may have been initialized in the same state as the bus. This problem is solved by the circuit in Fig. A.1(c). The QRAM is queried twice, and the no extra copying property is guaranteed by the fact that the entire QRAM unitary operation is involutory. In particular, even if the process of copying data during the first query acts non-trivially on modes not containing the bus, these modes are always reset to their initial states by the second query. In fact, even the bus is reset to its initial state by the second query. Thus, the information stored in the bus is copied to an ancillary qubit in between the two queries, then swapped back into the bus after the second query. We emphasize that the query fidelity of this circuit scales favorably, which can be shown by simply replacing $T \to 2T$ in the scaling argument from Section 3.2.2 to account for the fact that the QRAM is called twice.

As an aside, let us distinguish between our observation that QRAM is resilient to noise even when initialized in an arbitrary state (Section 3.3.1), and the observation of Refs. [36, 84] that the ancillary qubits used to perform a query can be "dirty." The latter states that circuits can be designed such that, *in the absence of errors*, any ancillary qubits used during the query are returned to their initial state after the query, regardless of what the initial state was (note the circuit in Fig. A.1(c) has this property). In contrast, our observation concerns what happens when errors occur during the query: the query infidelity of the circuit Fig. A.1(c) scales favorably even when the QRAM is initialized in an arbitrary state.

# Appendix B

# Effective operator formalism

In Chapter 5, we frequently use adiabatic elimination as a tool to extract the effective dynamics of an open quantum system within some stable subspace. The purpose of this Appendix is to describe the effective operator formalism that we employ in order to perform this adiabatic elimination. While adiabatic elimination has been described in a variety of prior works (see, e.g., [206, 214, 215]), we privilege the treatment in Ref. [206] due to its simplicity and ease of application. We briefly review the relevant results.

Consider an open quantum system evolving according to the master equation

$$\dot{\hat{\rho}} = -i[\hat{H}, \hat{\rho}] + \sum_i \mathcal{D}[\hat{L}_i](\hat{\rho}), \tag{B.1}$$

with Hamiltonian $\hat{H}$, jump operators $\hat{L}_i$, and where $\mathcal{D}[\hat{L}](\hat{\rho}) = \hat{L}\hat{\rho}\hat{L}^\dagger - \frac{1}{2}\left(\hat{L}^\dagger\hat{L}\hat{\rho} + \hat{\rho}\hat{L}^\dagger\hat{L}\right)$. We suppose that the system can be divided into two subspaces: a stable ground subspace, and a rapidly-decaying excited subspace, defined by the projectors $\hat{P}_g$ and $\hat{P}_e$, respectively. The Hamiltonian can be written in block form with respect to these subspaces as

$$\hat{H} = \begin{pmatrix} \hat{H}_g & \hat{V}_- \\ \\ \hat{V}_+ & \hat{H}_e \end{pmatrix} \tag{B.2}$$

where $\hat{H}_{g,e} = \hat{P}_{g,e}\hat{H}\hat{P}_{g,e}$, and $\hat{V}_{+,-} = \hat{P}_{e,g}\hat{H}\hat{P}_{g,e}$. We also suppose that the jump operators take the system from the excited to the ground subspace, i.e., $\hat{L}_i = \hat{P}_g\hat{L}_i\hat{P}_e$, and we define the non-Hermitian Hamiltonian

$$\hat{H}_{\mathrm{NH}} = \hat{H}_e - \frac{i}{2}\sum_i \hat{L}_i^\dagger \hat{L}_i. \tag{B.3}$$

$\hat{H}_{\mathrm{NH}}$ describes the evolution within the excited subspace; unitary evolution is generated by $\hat{H}_e$, while the remaining term describes the non-unitary, deterministic "no jump" evolution induced by the dissipators $\mathcal{D}[\hat{L}_i]$.

The authors of Ref. [206] consider the case where the evolution between the subspaces induced by $\hat{V}_{+,-}$ is perturbatively weak relative to the evolution induced by $\hat{H}_0 \equiv \hat{H}_g + \hat{H}_{\mathrm{NH}}$. Because the excited subspace is barely populated due to the rapid decays, the dynamics of the system are well-approximated by those within the ground subspace, governed by the effective master equation

$$\dot{\hat{\rho}} = -i[\hat{H}_{\mathrm{eff}}, \hat{\rho}] + \sum_i \mathcal{D}[\hat{L}_{\mathrm{eff},i}](\hat{\rho}), \tag{B.4}$$

where

$$\hat{H}_{\mathrm{eff}} = -\frac{1}{2}\hat{V}_- \left[\hat{H}_{\mathrm{NH}}^{-1} + \left(\hat{H}_{\mathrm{NH}}^{-1}\right)^\dagger\right]\hat{V}_+ + \hat{H}_g, \tag{B.5}$$

and

$$\hat{L}_{\mathrm{eff},i} = \hat{L}_i\hat{H}_{\mathrm{NH}}^{-1}\hat{V}_+. \tag{B.6}$$

These expressions apply for time-independent Hamiltonians. However, we will also be interested in situations where the perturbations $\hat{V}_{+,-}$ are time-dependent and take the form

$$\hat{V}_{+}(t) = \sum_{n} \hat{V}_{+,n} e^{i\delta_n t}, \tag{B.7}$$

$$\hat{V}_{-}(t) = \sum_{n} \hat{V}_{-,n} e^{-i\delta_n t}. \tag{B.8}$$

In this case, the effective Hamiltonian and jump operators are given by

$$\begin{aligned}\hat{H}_{\text{eff}} &= \hat{H}_g \\ &- \frac{1}{2} \sum_{m,n} \hat{V}_{-,n} \left[ \hat{H}_{\text{NH},m}^{-1} + \left( \hat{H}_{\text{NH},n}^{-1} \right)^{\dagger} \right] \hat{V}_{+,m} e^{i(\delta_m - \delta_n)t},\end{aligned} \tag{B.9}$$

and

$$\hat{L}_{\text{eff},i} = \hat{L}_i \sum_{n} \hat{H}_{\text{NH},n}^{-1} \hat{V}_{+,n} e^{i\delta_n t}, \tag{B.10}$$

where $\hat{H}_{\text{NH},n} = \hat{H}_{\text{NH}} + \delta_n$.

# Bibliography

[1] P. W. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.

[2] R. L. Rivest, A. Shamir, and L. Adleman, Commun. ACM **21**, 120 (1978).

[3] D. J. Bernstein, in *Post-Quantum Cryptography*, edited by D. J. Bernstein, J. Buchmann, and E. Dahmen (Springer, Berlin, Heidelberg, 2009) pp. 1–14.

[4] S. Lloyd, Science **273**, 1073 (1996).

[5] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Chemical Reviews **120**, 12685 (2020).

[6] A. Montanaro, Npj Quantum Inf. **2**, 15023 (2016).

[7] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. Lond. A **454**, 339 (1998).

[8] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. **83**, 5162 (1999).

[9] A. Aspuru-Guzik, Science **309**, 1704 (2005).

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).

[11] R. Cleve, arXiv:quant-ph/9906111 .

[12] A. Ambainis, in *Classical and New Paradigms of Computation and Their Complexity Hierarchies* (Springer, Dordrecht, 2004) pp. 15–32.

[13] M. Mosca, arXiv:0808.0369 .

[14] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).

[15] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. A **78**, 052310 (2008).

[16] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, New J. Phys. **17**, 123010 (2015).

[17] F.-Y. Hong, Y. Xiang, Z.-Y. Zhu, L.-z. Jiang, and L.-n. Wu, Phys. Rev. A **86**, 010306 (2012).

[18] O. Di Matteo, V. Gheorghiu, and M. Mosca, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[19] A. Paler, O. Oumarou, and R. Basmadjian, Phys. Rev. A **102**, 032608 (2020).

[20] B. M. Terhal, Rev. Mod. Phys. **87**, 307 (2015).

[21] M. A. Nielsen and I. L. Chuang, *Quantum Information and Quantum Computation: 10th Anniversary Edition* (Cambridge University Press, 2000).

[22] S. M. Girvin, in *Quantum Machines: Measurement and Control of Engineered Quantum Systems*, edited by M. Devoret, B. Huard, R. Schoelkopf, and L. F. Cugliandolo (Oxford University Press, 2014) pp. 113–256.

[23] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, arXiv:2005.12667 .

[24] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM J. Comput. **26**, 1510 (1997).

[25] A. Ambainis, arXiv:quant-ph/0002066 .

[26] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, arXiv:quant-ph/9802049 .

[27] P. Hoyer and R. Spalek, arXiv:quant-ph/0509153 .

[28] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (ACM, New York, NY, USA, 1996) pp. 212–219.

[29] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015).

[30] S. Aaronson, Nat. Phys. **11**, 291 (2015).

[31] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, arXiv:quant-ph/0005055 .

[32] L. K. Grover, Phys. Rev. Lett. **85**, 1334 (2000).

[33] S. Aaronson, arXiv:1607.05256 .

[34] I. Kerenidis and A. Prakash, arXiv:1603.08675 .

[35] S. Chakraborty, A. Gilyén, and S. Jeffery, ArXiv180401973 Quant-Ph , 14 pages (2019), arXiv:1804.01973 [quant-ph] .

[36] G. H. Low, V. Kliuchnikov, and L. Schaeffer, arXiv:1812.00954 .

[37] L. Grover and T. Rudolph, arXiv:quant-ph/0208112 .

[38] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Phys. Rev. Lett. **73**, 58 (1994).

[39] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

[40] V. V. Shende, I. L. Markov, and S. S. Bullock, Phys. Rev. A **69**, 062321 (2004).

[41] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Phys. Rev. Lett. **92**, 177902 (2004).

[42] S. Bullock, D. O'Leary, and G. Brennen, Phys. Rev. Lett. **94**, 230502 (2005).

[43] E. Knill, arXiv:quant-ph/9508006 .

[44] A. W. Harrow, B. Recht, and I. L. Chuang, Journal of Mathematical Physics **43**, 4445 (2002).

[45] P. A. Ivanov, E. S. Kyoseva, and N. V. Vitanov, Phys. Rev. A **74**, 022323 (2006).

[46] D. W. Berry and A. M. Childs, QIC **12** (2012), 10.26421/QIC12.1-2, arXiv:0910.4157 .

[47] V. Kliuchnikov, arXiv:1306.3200 .

[48] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).

[49] A. S. Householder, J. ACM **5**, 339 (1958).

[50] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, arXiv:2105.02859 .

[51] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[52] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic, arXiv:1512.02900 .

[53] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Proc. R. Soc. A **474**, 20170551 (2018).

[54] R. Schützhold, Phys. Rev. A **67**, 062311 (2003).

[55] G. Schaller and R. Schützhold, Phys. Rev. A **74**, 012303 (2006).

[56] A. M. Childs, B. W. Reichardt, R. Spalek, and S. Zhang, arXiv:quant-ph/0703015 .

[57] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).

[58] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv:1307.0411 .

[59] N. Wiebe, A. Kapoor, and K. M. Svore, (), arXiv:1412.3489 .

[60] N. Wiebe, A. Kapoor, and K. Svore, (), arXiv:1401.2142 .

[61] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).

[62] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[63] S. Lloyd, S. Garnerone, and P. Zanardi, Nat. Commun. **7** (2016), 10.1038/ncomms10138.

[64] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, arXiv:1710.02581 .

[65] A. M. Childs and J.-P. Liu, ArXiv190100961 Quant-Ph (2019), arXiv:1901.00961 [quant-ph] .

[66] I. Kerenidis and A. Prakash, Phys. Rev. A **101**, 022316 (2020).

[67] R. P. Feynman, Optics News, ON **11**, 11 (1985).

[68] A. M. Childs and N. Wiebe, Quantum Info. Comput. **12**, 901 (2012).

[69] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Proc. 46th Annu. ACM Symp. Theory Comput. - STOC 14 , 283 (2014), arXiv:1312.1414 .

[70] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019), arXiv:1610.06546 .

[71] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).

[72] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Chem. Rev. **119**, 10856 (2019).

[73] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, arXiv:2001.03685 .

[74] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Proc. Thirty-Fifth ACM Symp. Theory Comput. - STOC 03 , 59 (2003), arXiv:quant-ph/0209131 .

[75] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 230502 (2008).

[76] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109** (2012), 10.1103/PhysRevLett.109.050505.

[77] C. T. Hann, G. Lee, S. Girvin, and L. Jiang, PRX Quantum **2**, 020311 (2021).

[78] R. Asaka, K. Sakai, and R. Yahagi, Quantum Sci. Technol. **6**, 035004 (2021).

[79] A. M. Childs, D. Gosset, and Z. Webb, Science **339**, 791 (2013).

[80] D. W. Berry, A. M. Childs, and R. Kothari, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (2015) pp. 792–809.

[81] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Phys. Rev. X **8**, 041015 (2018).

[82] D. K. Park, F. Petruccione, and J.-K. K. Rhee, Sci Rep **9**, 1 (2019).

[83] T. M. L. de Veras, I. C. S. de Araujo, D. K. Park, and A. J. da Silva, arXiv:2011.07977 .

[84] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, Quantum **3**, 208 (2019).

[85] M. Saeedi and M. Pedram, Phys. Rev. A **87**, 062318 (2013).

[86] A. G. Fowler, S. J. Devitt, and C. Jones, Sci. Rep. **3** (2013), 10.1038/srep01939.

[87] J. O'Gorman and E. T. Campbell, Phys. Rev. A **95**, 032338 (2017).

[88] M. Saffman, T. G. Walker, and K. Mølmer, Rev. Mod. Phys. **82**, 2313 (2010).

[89] M. Saffman, J. Phys. B: At. Mol. Opt. Phys. **49**, 202001 (2016).

[90] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, Phys. Rev. A **76**, 062323 (2007).

[91] H. J. Kimble, Nature **453**, 1023 (2008).

[92] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, Phys. Rev. A **89**, 022317 (2014).

[93] C. R. Monroe, R. J. Schoelkopf, and M. D. Lukin, Sci. Am. **314**, 50 (2016).

[94] K. S. Chou, J. Z. Blumoff, C. S. Wang, P. C. Reinhold, C. J. Axline, Y. Y. Gao, L. Frunzio, M. H. Devoret, L. Jiang, and R. J. Schoelkopf, Nature **561**, 368 (2018).

[95] D. Steiger, "Racing in parallel: Quantum versus Classical," (2016).

[96] E. Tang, (), arXiv:1807.04271 .

[97] E. Tang, (), arXiv:1811.00414 .

[98] A. Gilyén, S. Lloyd, and E. Tang, arXiv:1811.04909 .

[99] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, Phys. Rev. Lett. **123**, 250501 (2019).

[100] T. H. Kyaw, S. Felicetti, G. Romero, E. Solano, and L.-C. Kwek, Sci. Rep. **5**, 8621 (2015).

[101] A. Cadellans, *A Transmon-Based Quantum Switch for a Quantum Random Access Memory*, Ph.D. thesis, Leiden University (2015).

[102] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund, arXiv:2103.07623 .

[103] R. K. Naik, N. Leung, S. Chakram, P. Groszkowski, Y. Lu, N. Earnest, D. C. McKay, J. Koch, and D. I. Schuster, Nat. Commun. **8**, 1904 (2017).

[104] N. Jiang, Y.-F. Pu, W. Chang, C. Li, S. Zhang, and L.-M. Duan, Npj Quantum Inf. **5**, 28 (2019).

[105] S. Langenfeld, O. Morin, M. Körber, and G. Rempe, Npj Quantum Inf. **6**, 1 (2020).

[106] O. Vy, X. Wang, and K. Jacobs, New J. Phys. **15**, 053002 (2013).

[107] E. Kapit, Phys. Rev. Lett. **120**, 050503 (2018).

[108] W.-L. Ma, M. Zhang, Y. Wong, K. Noh, S. Rosenblum, P. Reinhold, R. J. Schoelkopf, and L. Jiang, Phys. Rev. Lett. **125**, 110503 (2020).

[109] M. V. den Nest, arXiv:0911.1624 .

[110] D. Gottesman, (), arXiv:quant-ph/9807006 .

[111] M. Ramzan and M. K. Khan, Quantum Inf Process **11**, 443 (2012).

[112] H.-R. Wei, B.-C. Ren, and F.-G. Deng, Quantum Inf Process **12**, 1109 (2013).

[113] D. Gottesman, (), arXiv:0904.2557 .

[114] E. Knill and R. Laflamme, Phys. Rev. A **55**, 900 (1997).

[115] J. Preskill, Quantum **2**, 79 (2018).

[116] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, PNAS **115**, 9456 (2018).

[117] A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello, SIAM J. Comput. **26**, 1541 (1997).

[118] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).

[119] C. Gidney and M. Ekerå, arXiv:1905.09749 .

[120] T. J. Yoder, R. Takagi, and I. L. Chuang, Phys. Rev. X **6**, 031039 (2016).

[121] R. Chao and B. W. Reichardt, Phys. Rev. Lett. **121**, 050502 (2018).

[122] C. Chamberland and K. Noh, npj Quantum Inf **6**, 91 (2020).

[123] J. Guillaud and M. Mirrahimi, Phys. Rev. X **9**, 041053 (2019).

[124] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais, S. T. Flammia, and S. M. Girvin, Sci. Adv. **6**, eaay5901 (2020).

[125] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E. T. Campbell, C. T. Hann, J. Iverson, H. Putterman, T. C. Bohdanowicz, S. T. Flammia, A. Keller, G. Refael, J. Preskill, L. Jiang, A. H. Safavi-Naeini, O. Painter, and F. G. S. L. Brandão, arXiv:2012.04108 .

[126] A. W. Harrow, arXiv:1308.6595 .

[127] K. Temme, S. Bravyi, and J. M. Gambetta, Phys. Rev. Lett. **119**, 180509 (2017).

[128] S. Endo, S. C. Benjamin, and Y. Li, Phys. Rev. X **8**, 031027 (2018).

[129] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Nature **567**, 491 (2019).

[130] B. Koczor, (), arXiv:2011.05942 .

[131] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, arXiv:2011.07064 .

[132] P. Czarnik, A. Arrasmith, L. Cincio, and P. J. Coles, arXiv:2102.06056 .

[133] B. Koczor, (), arXiv:2104.00608 .

[134] A. Blais, J. Gambetta, A. Wallraff, D. I. Schuster, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf, Phys. Rev. A **75**, 032329 (2007).

[135] R. J. Schoelkopf and S. M. Girvin, Nature **451**, 664 (2008).

[136] M. Reagor, W. Pfaff, C. Axline, R. W. Heeres, N. Ofek, K. Sliwa, E. Holland, C. Wang, J. Blumoff, K. Chou, M. J. Hatridge, L. Frunzio, M. H. Devoret, L. Jiang, and R. J. Schoelkopf, Phys. Rev. B **94**, 014506 (2016).

[137] M. Hofheinz, E. M. Weig, M. Ansmann, R. C. Bialczak, E. Lucero, M. Neeley, A. D. O'Connell, H. Wang, J. M. Martinis, and A. N. Cleland, Nature **454**, 310 (2008).

[138] S. Krastanov, V. V. Albert, C. Shen, C.-L. Zou, R. W. Heeres, B. Vlastakis, R. J. Schoelkopf, and L. Jiang, Phys. Rev. A **92**, 040303 (2015).

[139] R. W. Heeres, P. Reinhold, N. Ofek, L. Frunzio, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, Nat. Commun. **8**, 94 (2017).

[140] L. Sun, A. Petrenko, Z. Leghtas, B. Vlastakis, G. Kirchmair, K. M. Sliwa, A. Narla, M. Hatridge, S. Shankar, J. Blumoff, L. Frunzio, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, Nature **511**, 444 (2014).

[141] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, Nature **536**, 441 (2016).

[142] L. Hu, Y. Ma, W. Cai, X. Mu, Y. Xu, W. Wang, Y. Wu, H. Wang, Y. P. Song, C.-L. Zou, S. M. Girvin, L.-M. Duan, and L. Sun, Nat. Phys. **15**, 503 (2019).

[143] K. Geerlings, S. Shankar, E. Edwards, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret, Appl. Phys. Lett. **100**, 192601 (2012).

[144] J. Wenner, R. Barends, R. C. Bialczak, Y. Chen, J. Kelly, E. Lucero, M. Mariantoni, A. Megrant, P. J. J. O'Malley, D. Sank, A. Vainsencher, H. Wang, T. C. White, Y. Yin, J. Zhao, A. N. Cleland, and J. M. Martinis, Appl. Phys. Lett. **99**, 113513 (2011).

[145] A. Romanenko, R. Pilipenko, S. Zorzetti, D. Frolov, M. Awida, S. Posen, and A. Grassellino, arXiv:1810.03703 .

[146] A. D. O'Connell, M. Hofheinz, M. Ansmann, R. C. Bialczak, M. Lenander, E. Lucero, M. Neeley, D. Sank, H. Wang, M. Weides, J. Wenner, J. M. Martinis, and A. N. Cleland, Nature **464**, 697 (2010).

[147] J.-M. Pirkkalainen, S. U. Cho, J. Li, G. S. Paraoanu, P. J. Hakonen, and M. A. Sillanpää, Nature **494**, 211 (2013).

[148] M. V. Gustafsson, T. Aref, A. F. Kockum, M. K. Ekström, G. Johansson, and P. Delsing, Science **346**, 207 (2014).

[149] Y. Chu, P. Kharel, W. H. Renninger, L. D. Burkhart, L. Frunzio, P. T. Rakich, and R. J. Schoelkopf, Science **358**, 199 (2017).

[150] Y. Chu, P. Kharel, T. Yoon, L. Frunzio, P. T. Rakich, and R. J. Schoelkopf, Nature **563**, 666 (2018).

[151] M. Kervinen, I. Rissanen, and M. Sillanpää, Phys. Rev. B **97**, 205443 (2018).

[152] R. Manenti, A. F. Kockum, A. Patterson, T. Behrle, J. Rahamim, G. Tancredi, F. Nori, and P. J. Leek, Nat. Commun. **8**, 975 (2017).

[153] A. Noguchi, R. Yamazaki, Y. Tabuchi, and Y. Nakamura, Phys. Rev. Lett. **119**, 180505 (2017).

[154] K. J. Satzinger, Y. P. Zhong, H.-S. Chang, G. A. Peairs, A. Bienfait, M.-H. Chou, A. Y. Cleland, C. R. Conner, É. Dumur, J. Grebel, I. Gutierrez, B. H. November, R. G. Povey, S. J. Whiteley, D. D. Awschalom, D. I. Schuster, and A. N. Cleland, Nature **563**, 661 (2018).

[155] B. A. Moores, L. R. Sletten, J. J. Viennot, and K. W. Lehnert, Phys. Rev. Lett. **120**, 227701 (2018).

[156] A. N. Bolgar, J. I. Zotova, D. D. Kirichenko, I. S. Besedin, A. V. Semenov, R. S. Shaikhaidarov, and O. V. Astafiev, Phys. Rev. Lett. **120**, 223603 (2018).

[157] L. R. Sletten, B. A. Moores, J. J. Viennot, and K. W. Lehnert, Phys. Rev. X **9**, 021056 (2019).

[158] P. Arrangoiz-Arriola, E. A. Wollack, Z. Wang, M. Pechal, W. Jiang, T. P. McKenna, J. D. Witmer, and A. H. Safavi-Naeini, Nature **571**, 537 (2019).

[159] A. H. Safavi-Naeini, D. V. Thourhout, R. Baets, and R. V. Laer, Optica **6**, 213 (2019).

[160] G. S. MacCabe, H. Ren, J. Luo, J. D. Cohen, H. Zhou, A. Sipahigil, M. Mirhosseini, and O. Painter, Science **370**, 840 (2020).

[161] M. J. A. Schuetz, E. M. Kessler, G. Giedke, L. M. K. Vandersypen, M. D. Lukin, and J. I. Cirac, Phys. Rev. X **5**, 031031 (2015).

[162] A. N. Cleland and M. R. Geller, Phys. Rev. Lett. **93**, 070501 (2004).

[163] A. Bienfait, K. J. Satzinger, Y. P. Zhong, H.-S. Chang, M.-H. Chou, C. R. Conner, É. Dumur, J. Grebel, G. A. Peairs, R. G. Povey, and A. N. Cleland, Science **364**, 368 (2019).

[164] L. Guo, A. Grimsmo, A. F. Kockum, M. Pletyukhov, and G. Johansson, Phys. Rev. A **95**, 053821 (2017).

[165] G. Andersson, B. Suri, L. Guo, T. Aref, and P. Delsing, arXiv:1812.01302 .

[166] M. Pechal, P. Arrangoiz-Arriola, and A. H. Safavi-Naeini, Quantum Sci. Technol. **4**, 015006 (2018).

[167] R. Manenti, M. J. Peterer, A. Nersisyan, E. B. Magnusson, A. Patterson, and P. J. Leek, Phys. Rev. B **93**, 041411 (2016).

[168] T. Aref, P. Delsing, M. K. Ekström, A. F. Kockum, M. V. Gustafsson, G. Johansson, P. J. Leek, E. Magnusson, and R. Manenti, in *Superconducting Devices in Quantum Optics*, edited by R. H. Hadfield and G. Johansson (Springer International Publishing, Cham, 2016) pp. 217–244.

[169] W. H. Renninger, P. Kharel, R. O. Behunin, and P. T. Rakich, Nat. Phys. **14**, 601 (2018).

[170] P. Kharel, Y. Chu, M. Power, W. H. Renninger, R. J. Schoelkopf, and P. T. Rakich, APL Photonics **3**, 066101 (2018).

[171] X. Han, C.-L. Zou, and H. X. Tang, Phys. Rev. Lett. **117**, 123603 (2016).

[172] T. A. Palomaki, J. W. Harlow, J. D. Teufel, R. W. Simmonds, and K. W. Lehnert, Nature **495**, 210 (2013).

[173] M. Pechal, L. Huthmacher, C. Eichler, S. Zeytinoğlu, A. A. Abdumalikov, S. Berger, A. Wallraff, and S. Filipp, Phys. Rev. X **4**, 041010 (2014).

[174] S. J. Srinivasan, N. M. Sundaresan, D. Sadri, Y. Liu, J. M. Gambetta, T. Yu, S. M. Girvin, and A. A. Houck, Phys. Rev. A **89**, 033857 (2014).

[175] C. J. Axline, L. D. Burkhart, W. Pfaff, M. Zhang, K. Chou, P. Campagne-Ibarcq, P. Reinhold, L. Frunzio, S. M. Girvin, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, Nat. Phys. **14**, 705 (2018).

[176] P. Kurpiers, P. Magnard, T. Walter, B. Royer, M. Pechal, J. Heinsoo, Y. Salathé, A. Akin, S. Storz, J.-C. Besse, S. Gasparinetti, A. Blais, and A. Wallraff, Nature **558**, 264 (2018).

[177] J. Y. Mutus, T. C. White, E. Jeffrey, D. Sank, R. Barends, J. Bochmann, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, J. Kelly, A. Megrant, C. Neill, P. J. J. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, I. Siddiqi, R. Vijay, A. N. Cleland, and J. M. Martinis, Appl Phys Lett **103**, 122602 (2013).

[178] Z. Leghtas, S. Touzard, I. M. Pop, A. Kou, B. Vlastakis, A. Petrenko, K. M. Sliwa, A. Narla, S. Shankar, M. J. Hatridge, M. Reagor, L. Frunzio, R. J. Schoelkopf, M. Mirrahimi, and M. H. Devoret, Science **347**, 853 (2015).

[179] Y. Y. Gao, B. J. Lester, Y. Zhang, C. Wang, S. Rosenblum, L. Frunzio, L. Jiang, S. M. Girvin, and R. J. Schoelkopf, Phys. Rev. X **8**, 021073 (2018).

[180] Y. Zhang, B. J. Lester, Y. Y. Gao, L. Jiang, R. J. Schoelkopf, and S. M. Girvin, Phys. Rev. A **99**, 012314 (2019).

[181] N. K. Langford, S. Ramelow, R. Prevedel, W. J. Munro, G. J. Milburn, and A. Zeilinger, Nature **478**, 360 (2011).

[182] M. Y. Niu, I. L. Chuang, and J. H. Shapiro, Phys. Rev. Lett. **120**, 160502 (2018).

[183] P. Kharel, G. I. Harris, E. A. Kittlaus, W. H. Renninger, N. T. Otterstrom, J. G. E. Harris, and P. T. Rakich, arXiv:1809.04020 .

[184] C. Axline, M. Reagor, R. Heeres, P. Reinhold, C. Wang, K. Shain, W. Pfaff, Y. Chu, L. Frunzio, and R. J. Schoelkopf, Appl. Phys. Lett. **109**, 042601 (2016).

[185] S. E. Nigg, H. Paik, B. Vlastakis, G. Kirchmair, S. Shankar, L. Frunzio, M. H. Devoret, R. J. Schoelkopf, and S. M. Girvin, Phys. Rev. Lett. **108**, 240502 (2012).

[186] M. Y. Niu, I. L. Chuang, and J. H. Shapiro, Phys. Rev. A **97**, 032323 (2018).

[187] P. T. Cochrane, G. J. Milburn, and W. J. Munro, Phys. Rev. A **59**, 2631 (1999).

[188] H. Jeong and M. S. Kim, Phys. Rev. A **65**, 042305 (2002).

[189] M. Mirrahimi, Z. Leghtas, V. V. Albert, S. Touzard, R. J. Schoelkopf, L. Jiang, and M. H. Devoret, New J. Phys. **16**, 045014 (2014).

[190] V. V. Albert, K. Noh, K. Duivenvoorden, D. J. Young, R. T. Brierley, P. Reinhold, C. Vuillot, L. Li, C. Shen, S. M. Girvin, B. M. Terhal, and L. Jiang, Phys. Rev. A **97**, 032346 (2018).

[191] A. Joshi, K. Noh, and Y. Y. Gao, Quantum Sci. Technol. **6**, 033001 (2021).

[192] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, Nat. Commun. **12**, 2172 (2021).

[193] A. S. Darmawan, B. J. Brown, A. L. Grimsmo, D. K. Tuckett, and S. Puri, arXiv:2104.09539 .

[194] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Phys. Rev. Lett. **120**, 050505 (2018).

[195] J. Guillaud and M. Mirrahimi, arXiv:2009.10756 .

[196] S. Touzard, A. Grimm, Z. Leghtas, S. O. Mundhada, P. Reinhold, C. Axline, M. Reagor, K. Chou, J. Blumoff, K. M. Sliwa, S. Shankar, L. Frunzio, R. J. Schoelkopf, M. Mirrahimi, and M. H. Devoret, Phys. Rev. X **8**, 021005 (2018).

[197] R. Lescanne, M. Villiers, T. Peronnin, A. Sarlette, M. Delbecq, B. Huard, T. Kontos, M. Mirrahimi, and Z. Leghtas, Nat. Phys. **16**, 509 (2020).

[198] S. Puri, S. Boutin, and A. Blais, Npj Quantum Inf. **3**, 18 (2017).

[199] S. Puri, A. Grimm, P. Campagne-Ibarcq, A. Eickbusch, K. Noh, G. Roberts, L. Jiang, M. Mirrahimi, M. H. Devoret, and S. M. Girvin, Phys. Rev. X **9**, 041009 (2019).

[200] A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, and M. H. Devoret, Nature **584**, 205 (2020).

[201] P. Aliferis and J. Preskill, Phys. Rev. A **78**, 052331 (2008).

[202] P. Webster, S. D. Bartlett, and D. Poulin, Phys. Rev. A **92**, 062309 (2015).

[203] Z. Leghtas, G. Kirchmair, B. Vlastakis, R. Schoelkopf, M. Devoret, and M. Mirrahimi, Phys. Rev. Lett. **111**, 120501 (2013).

[204] D. F. V. James and J. Jerke, Can. J. Phys. **85**, 625 (2007).

[205] O. Gamel and D. F. V. James, Phys. Rev. A **82**, 052106 (2010).

[206] F. Reiter and A. S. Sørensen, Phys. Rev. A **85**, 032111 (2012).

[207] P. Mundada, G. Zhang, T. Hazard, and A. Houck, Phys. Rev. Appl. **12**, 054023 (2019).

[208] Y. Chen, C. Neill, P. Roushan, N. Leung, M. Fang, R. Barends, J. Kelly, B. Campbell, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, A. Megrant, J. Y. Mutus, P. J. J. O'Malley, C. M. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, M. R. Geller, A. N. Cleland, and J. M. Martinis, Phys. Rev. Lett. **113**, 220502 (2014).

[209] E. A. Sete, J. M. Martinis, and A. N. Korotkov, Phys. Rev. A **92**, 012325 (2015).

[210] V. S. Ferreira, J. Banker, A. Sipahigil, M. H. Matheny, A. J. Keller, E. Kim, M. Mirhosseini, and O. Painter, arXiv:2001.03240 .

[211] Y. Y. Gao, B. J. Lester, K. S. Chou, L. Frunzio, M. H. Devoret, L. Jiang, S. M. Girvin, and R. J. Schoelkopf, Nature **566**, 509 (2019).

[212] C. K. Hong, Z. Y. Ou, and L. Mandel, Phys. Rev. Lett. **59**, 2044 (1987).

[213] C. J. Axline, L. D. Burkhart, W. Pfaff, M. Zhang, K. Chou, P. Campagne-Ibarcq, P. Reinhold, L. Frunzio, S. M. Girvin, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, Nat. Phys. **14**, 705 (2018).

[214] R. Azouit, A. Sarlette, and P. Rouchon, arXiv:1603.04630 .

[215] R. Azouit, F. Chittaro, A. Sarlette, and P. Rouchon, Quantum Sci. Technol. **2**, 044011 (2017).