UNIVERSITY OF TRENTO

Department of Information Engineering and Computer Science

DISTRIBUTED SIMULATION METHODS AND ALGORITHMS FOR SUPERCONDUCTING QUANTUM COMPUTER RELIABILITY

ON THE EFFECTS OF RADIATION ON QUANTUM INFORMATION, ERROR CORRECTION AND HYBRID SYSTEMS

MARZIO VALLERO

DOCTORATE IN INFORMATION ENGINEERING AND COMPUTER SCIENCE
SPECIALIZATION IN QUANTUM SCIENCE AND TECHNOLOGIES

University of Trento

UNIVERSITY OF TRENTO | Department of Information Engineering and Computer Science

DISTRIBUTED SIMULATION METHODS AND ALGORITHMS FOR SUPERCONDUCTING QUANTUM COMPUTER RELIABILITY

ON THE EFFECTS OF RADIATION ON QUANTUM INFORMATION, ERROR CORRECTION AND HYBRID SYSTEMS

MARZIO VALLERO

Adviser: Flavio Vella
Associate Professor, University of Trento

Co-adviser: Paolo Rech
Associate Professor, University of Trento

Examination Committee and Reviewers

Chair: Stefano Cherubin
Associate Professor, Norwegian University of Science and Technology

Rapporteur: Enrico Blanzieri
Associate Professor, University of Trento

Reviewers: Rosa Maria Badia Sala
Full Professor, Polytechnic University of Catalonia

Robert Wille
Full Professor, Technical University of Munich

DOCTORATE IN INFORMATION ENGINEERING AND COMPUTER SCIENCE
SPECIALIZATION IN QUANTUM SCIENCE AND TECHNOLOGIES

University of Trento

Submitted on the 19th of January, 2026

Defended on the 29th of April, 2026

I hereby declare that all the contents of this work are original and novel, in accordance with the code of conduct of the University of Trento and the Italian and international laws on copyright. The reporting of sentences or figures already object of publication is made with the express authorisation from all the involved authors and copyright holders.

Marzio Vallero, the author

Distributed Simulation Methods and Algorithms for Superconducting Quantum Computer Reliability

This document was formatted with the pdfLaTeX processor.
The cover images have been personally formatted by the author, under fair use from the original creators.

A mio nonno Aldo,
per la sua contagiosa curiosità.

”

“It’s a dangerous business, Frodo, going out your door. You step onto the road, and if you don’t keep your feet, there’s no knowing where you might be swept off to.”

— **John Ronald Reuel Tolkien**, The Fellowship of the Ring (1954)

AI Statement

No portion of this thesis, be it conceptual, textual or visual,
is the product of generative artificial intelligence.

> *"Once men turned their thinking over to machines*
> *in the hope that this would set them free.*
> *But that only permitted other men*
> *with machines to enslave them."*
>
> — **Frank Herbert**, Dune
> (1965)

Abstract

Superconducting quantum computers are yet to reach the fault-tolerant regime, and thus the tantalising promises behind quantum algorithms are laid barren. To cross this knowledge-concealing wasteland, one must first understand how both intrinsic and extrinsic noise hinders the reliability of quantum computing systems. Most recently, in fact, external impinging particle events have been discovered to be one of the causes behind sudden bursts of information loss in these devices. Modelling such events with standard methods is a masterfully complex endeavour, which is ultimately hampered by computational costs in terms of memory and time. There is a growing requirement for an accurate fault model able to characterise a qubit's interaction with impinging particles, and more generally, the performance of full algorithms on quantum computers in the presence of radiation. Such a fault model would be the first of many other steps, paving the way for research on new quantum error correction and mitigation algorithms, and ultimately giving scientists a new instrument for tackling and understanding this ever growing issue. Reaching such goal requires the efficient simulation of quantum systems, both at the algorithmic and device level, pushing against the known limitations and bottlenecks imposed by these methods.

The first objective is to understand the feasibility of quantum circuit simulations with the available computational methods, underlining bounds and limitations. This paved the way to provide a physically accurate noise model for the interaction of impinging particles on superconducting qubits, the most widespread and scalable technology for building quantum computers. The model has been devised to be easy to interpret and simulate, whilst being highly expressive and tunable, in order to keep up with future technological advancements in the field. The following step has been to leverage the fault model to develop methods and techniques apt at testing the reliability of Quantum Error Correction (QEC) and quantum algorithms alike. This includes methods for detecting the presence of radiation-induced faults at runtime, and applying partial information reconstruction methods. A deeper understanding of the effects of radiation also prompted the modelling and study of hardware hardening techniques. At last, the interoperation of quantum and classical computing systems has

been investigated in the context of the reliability of hybrid machine learning algorithms, which are going to play an ever important role in the high performance computing systems of the future.

These contributions press on in the quest for knowledge, impacting the quantum computing stack, correlating logical and physical quantum circuit design, and widening the understanding of how to prevent faults in quantum computing systems, quickening the advent of fault-tolerant quantum computation.

Keywords: quantum computing, reliability, superconducting, distributed computing, fault modelling

Contents

List of Figures

List of Tables

Acronyms

ASIC Application specific integrated circuit *(p. 72)*
AVF Architecture Vulnerability Factor *(p. 118)*
BF Belief Find *(p. 87)*
BM Belief Matching *(pp. 72, 87)*
BP Belief Propagation *(p. 72)*
BQP Bounded Quantum Polynomial time *(p. 111)*
CMOS Complementary Metal-Oxide-Semiconductor *(pp. 11, 17–19, 117, 122, 127, 129)*
CNN Convolutional Neural Network *(pp. 114, 122)*
CNOT Controlled-not *(pp. 6, 66, 114)*
CPU Central Processing Unit *(pp. 10, 32, 33, 36)*
DAG Directed Acyclic Graph *(pp. 26, 66)*
EPR Einstein-Podolsky-Rosen *(p. 3)*
FIFO First In First Out *(p. 77)*
FLOPS Floating Point Operations per Second *(pp. 40, 45)*
GHZ Greenberger-Horne-Zeilinger *(p. 55)*
GPU Graphical Processing Unit *(pp. 9, 10, 23, 32, 35, 37, 40–44, 46–48, 132, VIII)*
HPC High Performance Computing *(p. 22)*
I/O Input/Output *(p. 37)*
LRC Inductance-Resistance-Capacitance *(p. 11)*
ML Machine Learning *(p. 115)*
MNIST Modified National Institute of Standards and Technology *(pp. 114, 115, 125)*
MPI Message Passing Interface *(pp. 40, 43)*
MPO Matrix Product Operator *(p. 9)*
MPS Matrix Product State *(pp. 9, 38)*
MWPM Minimum Weight Perfect Matching *(pp. 16, 54, 57, 72, 73, 84, 85, 87, 89)*
NCCL NVIDIA Collective Communications Library *(p. 40)*
NISQ Noisy Intermadiate Scale Quantum *(pp. 2, 14, 17, 22, 109, 113)*

NP Nondeterministic Polynomial time *(pp. 22, 27, 46, 72)*

PVF Program Vulnerability Factor *(p. 118)*

QAC Quantum Adiabatic Computing *(p. 7)*

QAOA Quantum Approximate Optimisation Algorithm *(pp. 10, 30, 33, 35, 38, 39, 41, 43, 44, 46)*

QC Quantum Computing *(pp. 1, 22, 129)*

QEC Quantum Error Correction *(pp. 2, 10, 14–17, 20, 21, 49, 50, 54, 67–77, 82–85, 87, 89–92, 95, 99, 100, 103–109, 113, 131, 132, V)*

QFT Quantum Fourier Transform *(pp. 30, 31, 33, 35, 37, 39, 41, 43, 45)*

QML Quantum Machine Learning *(pp. 111, 113, 118, 129)*

QNN Quanvolutional Neural Network *(pp. 109, 110, 112–119, 121–123, 125, 127–130)*

QPE Quantum Phase Estimation *(pp. 30, 31, 33, 35, 39, 41, 43, 45)*

QVF Quantum Vulnerability Factor *(pp. 118–120)*

RAM Random Access Memory *(p. 32)*

REI Radiation Event Identification *(pp. 70, 76–78, 80–87, 89)*

UF Union Find *(pp. 72, 87)*

VQE Variational Quantum Algorithm *(pp. 10, 30, 31, 33, 35, 36, 38, 39, 41, 44)*

VRAM Video Random Access Memory *(p. 9)*

Chapter 1

Introduction

Do we really need quantum computers?
One could blindly argue in favour of this, however, the real boundary that separates classical computation from quantum computation is yet to be thoroughly defined. The quantum computing paradigm, since its theoretical inception by one of the forefathers of modern physics R. Feynman in 1982 [1], served to extend the classical definition of computation. This was done in order to better describe the quantum properties of nature, namely leveraging *superposition* and *entanglement*. This followed the hypothesis that *an experiment*, a purposefully built physical system in that sense, can be said to be *performing computations* under specific conditions [2]. Many quantum algorithms would then be proposed, promising quadratic or exponential speedups, and never-before-seen efficiency per operation when compared to classical compute systems [3, 4], a highly debated concept, later generalised as quantum advantage [5]. In recent years, thanks to the widespread access to simulators and quantum devices over the cloud, researchers have been able to quickly expand the reach of Quantum Computing (QC) to fields such as finance [6], chemistry [7], biomechanics [8] and machine learning [9, 10].

It is well known that simulating quantum systems with classical machines, namely through the Schrödinger-Feynman method [11], incurs in exponential bottlenecks both in terms of memory and compute time, scaling according to the size of the encoded Hilbert space [12]. Although alternative simulation techniques have been proposed over recent decades, such as tensor network contraction [13, 14] and decision-diagram methods [15, 16], they all feature unique limitations that prevent them from simulating arbitrarily complex universal

quantum circuits. Such limitations put forth the need to build *real* quantum computers, where quantum algorithms pose the stark requirement of *fault tolerance* onto the quantum bits (qubits) used for execution.

In the last four decades, research has shifted from a few experimental hardware implementations to a blossoming of industry-grade quantum computers [17, 18], albeit with limited functionality. This widened the reach of such technologies, making quantum computers available to researchers around the world via cloud and in-premises access. However, the nature of current quantum devices is still a far cry from the expected reliability requirements of quantum algorithms. Currently, Noisy Intermadiate Scale Quantum (NISQ) [19] devices make use of error mitigation at the physical qubit level, to reduce the statistical impact of fault events on the computation results provided by the quantum computers [20]. Further hardware-level noise reduction comes at a high cost, being limited by engineering constraints, thus prompting the adoption of software-level solutions, namely QEC codes [21]. The simplest error correction code requires 5 physical qubits to encode a single bit of quantum information, with considerably limited fault tolerance [22]. While intrinsic noise is a generally well understood phenomenon over which QEC is developed and tested, extrinsic fault events are yet to be fully understood. In the context of superconducting quantum computers, a notable sensitivity to extrinsic radiation events has been recently highlighted [23, 24], as detailed later. Overcoming this source of faults is a critical step in paving the way to reach fault-tolerant quantum algorithms.

1.1 Literature review

This section acts as the backbone to follow the topics of interest of the thesis. Starting from an outline of quantum computing theory and simulation, follows a general description of the major quantum computer technologies. After a more detailed grounding on superconducting qubits, follows a presentation of intrinsic noise, immediately followed by a quantum error mitigation and correction. The topic of the interactions between radiation and superconducting qubits is covered at the end of this section.

1.1.1 Quantum information theory

Quantum computing, from an information standpoint, is an extension over $\mathbb{C}^2$ of classical binary computing. The minimum unit of quantum information is a *qubit*, a controllable two-level quantum mechanical system, acting as the counterpart of the classical *bit*. Any classical algorithm can thus be easily mapped onto a quantum algorithm, although this leads to no

advantage to be gained *per se*. The alleged advantages of quantum computing stem from the exploitation of properties unavailable in a classical information setting, namely *superposition* and *entanglement*. The former allows a qubit to partake multiple different states at once, whilst the latter manifests a linking between multiple qubits into a higher-level object which displays non-classical correlation patterns amongst its constituent elements.

It is worth specifying the differences between a bit and a qubit from an information standpoint. In the classical domain, a binary state ψ is defined as a linear combination over two bases, exclusively taking the value of either one. These bases are vectors in a linear space, and can be defined using Dirac's notation.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.1}$$

Conceptually, it may help to think of a classical binary state as having unit probability of being in either base, and zero probability of being in the other, at any point in time. This implies that the sum of the two squared probabilities equals unity, as such $a^2 + b^2 = 1$

$$\psi = a\,|0\rangle + b\,|1\rangle\,, \quad a, b \in \{0, 1\}. \tag{1.2}$$

In the quantum information domain, a qubit $\left|\psi\right\rangle$ is defined over those same two bases, while withholding the exclusivity requirement. This property is called superposition, implying that a qubit can simultaneously exist in the two basis states with a given probability amplitude, by defining $\alpha, \beta \in \mathbb{C}$

$$\left|\psi\right\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C}. \tag{1.3}$$

Again, the square of an amplitude represents the probability for the qubit to collapse on that basis state when observed. The conservation of the sum of probabilities is still enforced, as such $\alpha^2 + \beta^2 = 1$.

Entanglement describes the ability of two qubits to share a non-classical correlation that acts on the information stored in two or more qubits. This makes it so that, by observing one of the entangled qubits, one can infer information regarding the state of the other qubits without observing them. This second property sprouted issues with the locality of quantum mechanics in the well-known Einstein-Podolsky-Rosen (EPR) paradox [25], later solved by J.S. Bell [26], which defined the minimum set of these namesakes entangled quantum states. These two qubit states cannot be decomposed in a linear combination of two independent qubits, such as

$$|\Phi^+\rangle = \alpha\,|0\rangle \otimes |0\rangle + \beta\,|1\rangle \otimes |1\rangle = \alpha\,|00\rangle + \beta\,|11\rangle\,. \tag{1.4}$$

Entanglement is believed to be the fundamental resource responsible for achieving algorithmic improvements over classical information processing, although superposition plays an important role as well. This is reflected in the fact that circuits with low entanglement have been proven to be trivial to simulate with classical algorithms in logarithmic time [27].

These two fundamental qubit properties are retained for as long as a qubit is not observed directly. In fact, measuring the state of a qubit inherently destroys the quantum information stored therein, forcing it to collapse in either of the measurement bases, due to the *observer effect* [28]. Any following measurement will always yield the same classical output with unit probability. Retrieving the output distribution of a qubit thus involves the extraction of multiple separate samples. Given an arbitrary quantum state $\left|\psi\right\rangle$, the measurement operator M_m and its conjugate transpose $M_m^\dagger$ define the probability of observing m as $P : \mathcal{F} \longrightarrow [0, 1]$, where the function $\mathcal{F}$ represents the collection of possible outcome events

$$P(m) = \left\langle\psi\right| M_m^\dagger M_m \left|\psi\right\rangle\,. \tag{1.5}$$

Following this first measurement, the system collapses in a classical state and is conditioned on having measured m, according to the *Born rule* [29]

$$\left|\psi\right\rangle \rightarrow \frac{M_m \left|\psi\right\rangle}{\sqrt{\left\langle\psi\right| M_m^\dagger M_m \left|\psi\right\rangle}}. \tag{1.6}$$

Measurement is thus a unitary operator performing a projection onto an orthonormal basis. The matrix form of the measurement operators M_0 and M_1 for the Z basis is defined as the self outer product of the basis states

$$M_0 = |0\rangle\langle 0|\ , \quad M_1 = |1\rangle\langle 1|\,. \tag{1.7}$$

One can compute the probability of a qubit to be measured in state $|1\rangle$ as follows

$$\begin{aligned}\left\langle\psi\right| M_1^\dagger M_1 \left|\psi\right\rangle &= \left\langle\psi\right| |1\rangle\langle 1| \left|\psi\right\rangle \\ &= (\alpha^*\langle 0| + \beta^*|1\rangle)(|1\rangle\langle 1|)(\alpha|0\rangle + \beta|1\rangle) \\ &= |\beta|^2.\end{aligned} \tag{1.8}$$

Substituting this result in the measurement equation, conditioned on having measured $|1\rangle$, always returns the same classical state

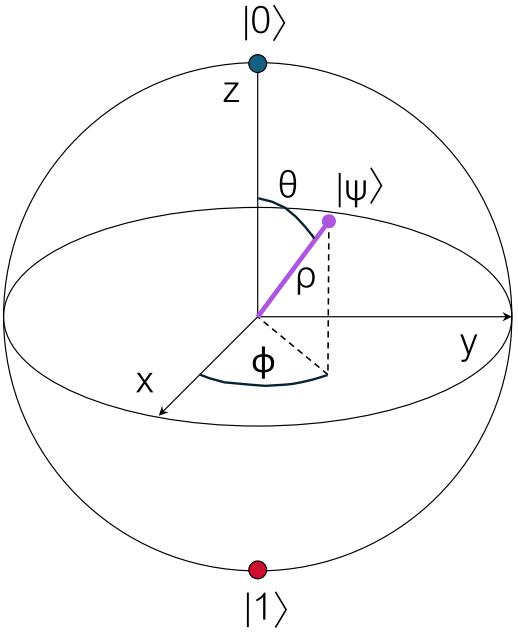


Figure 1.1: Bloch sphere of an arbitrary quantum state.

$$|\psi\rangle \rightarrow \frac{M_1 |\psi\rangle}{\sqrt{\langle\psi| M_1^\dagger M_1 |\psi\rangle}} = \frac{1}{|\beta|} |1\rangle \langle 1| (\alpha |0\rangle + \beta |1\rangle) = \frac{\beta}{|\beta|} |1\rangle \approx |1\rangle . \tag{1.9}$$

A qubit can be visualised on the Bloch sphere with unit radius, mapping the quantum state onto a vector via spherical coordinates, thus parameterised by the radial distance ρ, the polar angle θ and the azimuthal angle ϕ. This mapping lets one re-parametrise the α and β probability amplitudes in terms of θ and ϕ as follows

$$\alpha = \cos\left(\frac{\theta}{2}\right), \quad \beta = e^{i\phi} \sin\left(\frac{\theta}{2}\right). \tag{1.10}$$

as defined over the closed intervals $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$ in $\mathbb{R}$. As such, $\alpha \in \mathbb{R}$, whilst $\beta \in \mathbb{C}$. An arbitrary quantum state can be written as

$$|\psi\rangle = \rho \left(\cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \right). \tag{1.11}$$

The quantum states defined over $\rho = 1$ are called *pure states*, and lie on the surface of the Bloch sphere, as shown in Figure 1.1. These states describe with unit probability the information encoded in the quantum system. Points inside the sphere's volume are *mixed states* [12], with $\rho \in [0, 1)$. These states are instead described as groupings of multiple pure states, each with its own probability, summing to one over the grouping.

1.1.2 Quantum circuits

An algorithm employing qubits is generally expressed as a quantum circuit. These circuit representation are derived from Penrose's notation [30], and are read from left to right, following the flow of information. Quantum circuits involve an ensemble of qubits being addressed through the sequential application of unitary operators, or *quantum gates*. These gates can operate on single qubits, or on multiple qubits, and are represented by $2^N \times 2^N$ unitary matrices, with N being the number of qubits addressed by the gate. The most basic single-qubit operators include the Pauli X, Y, Z gates and the basis-swap Hadamard gate, represented by the following matrices

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{1.12}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.13}$$

Multiple qubit gates are generally employed to generate entangled states, and are usually composed of one or more *control* qubits that condition the execution of a certain operation on one or more *target* qubits. A prime example of this is the Controlled-not (CNOT) gate, that conditionally applies an X gate to a single target qubit if the control qubit is in the $|1\rangle$ state. The matrix representation of the CNOT gate is

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.14}$$

Representing any arbitrary quantum algorithm requires the definition of a universal quantum gate set, similarly to the classical information domain. The fundamental property of this set is its ability to express arbitrary operations as the composition of a limited number of instructions, a key requirement for universal quantum computers. The Clifford gate set, which contains all the gates composed of H, S ($S = \sqrt{Z}$) and CNOT gates, is a non-universal quantum gate set. Notably, quantum circuits decomposable to the Clifford gate set can be classically simulated in polynomial time [12].

Figure 1.2 shows the quantum circuit that encodes the $|\Phi^+\rangle$ Bell state: the two qubits are initialised in state $|0\rangle$, then the first qubit is put in the equiprobable superposition state $|+\rangle$ and is used as the control of a CNOT gate, after which the second qubit is entangled

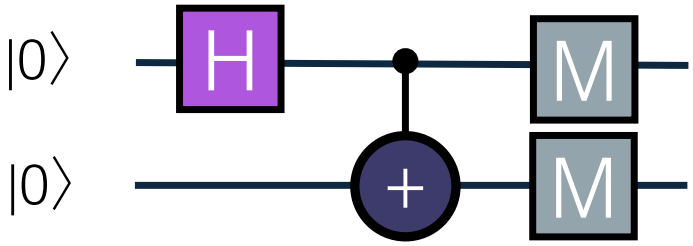


Figure 1.2: $|\Phi^+\rangle$ Bell state quantum circuit.

with the first. The sequence of gates that compose this quantum circuit are represented by the following equation:

$$|\Phi^+\rangle = CNOT \cdot (H \otimes I) \cdot |00\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{1.15}$$

Since the measurement of a qubit yields a classical bit, the execution of a quantum circuit on a real quantum device yields a single *classical* output bit string. In order to extract an approximate output distribution of the possible quantum states, one must retrieve multiple samples of the same quantum circuit. This collection of samples converges to the element-wise square of the state vector made of the probability amplitudes of each possible quantum state in the system. Experimental results and expected theoretical results oftentimes differ, due to noise phenomena such as imperfections in the qubits, in the execution of the quantum circuit, and the number of samples made. An example of this process is presented in Figure 1.3, depicting the theoretical output distribution of the Bell circuit on the top and the experimental output distribution obtained from a real quantum device, an IBM Falcon r4T processor over 1024 measurement shots, on the bottom.

Quantum circuits are thus defined at a logical, high abstraction level, only to be later converted into sequences of quantum gates, the ones that can be directly carried out by the architecture of each specific quantum device. This *transpilation* process involves mapping the quantum circuit onto a system of imperfect components, taking into account optimisations, noise reduction metrics[31], and hardware routing constraints.

1.1.3 Quantum simulation methods

Quantum computation can be used to refer to multiple and well separated methods. The most widespread and well-known approach is that of universal gate-based quantum computing, which employs a set of quantum operators sequentially applied to qubits. Other approaches include Quantum Adiabatic Computing (QAC) and Quantum Annealing. The

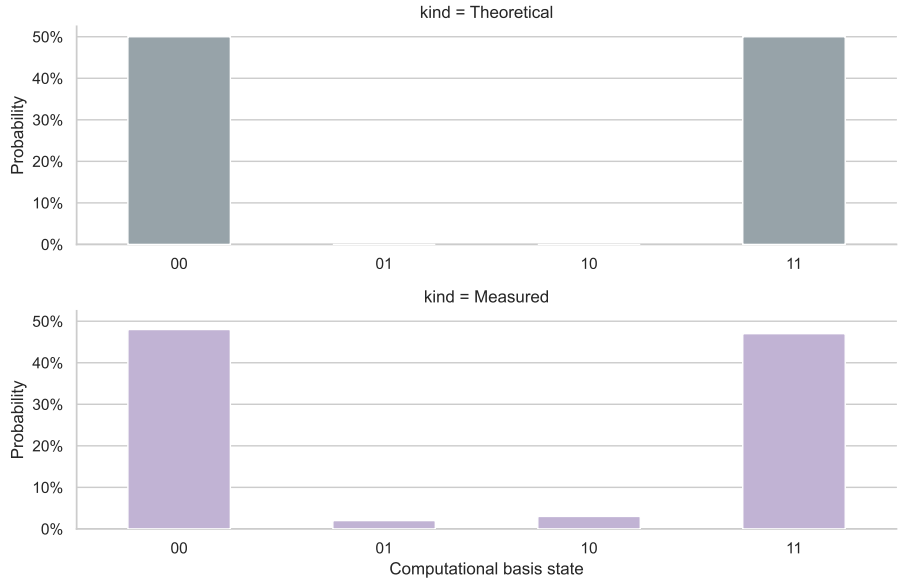


Figure 1.3: Theoretical and hardware measured outputs of the Bell circuit.

former evolves a system according to Quantum Adiabatic theorem from a starting Hamiltonian to a target Hamiltonian by never leaving the ground state. The latter employs the properties of quantum tunnelling to evolve a Hamiltonian towards its ground state. This thesis will solely cover universal gate-based quantum computation.

The scientific achievement of *exascale* compute capability has been just recently reached [32], and one may argue that conquering the next computational threshold will involve hybrid quantum-classical systems [33]. Supercomputing systems that leverage this hybrid union are already in place, although with limited capacity on the quantum accelerator side of things [34]. It is thus important to understand which algorithms should be executed on a quantum accelerator, and which algorithms should be simulated, as the gap steadily shrinks in select scenarios [35], converting them into quantum-inspired classical algorithms. Crucially, testing and validation of quantum algorithms, and fidelity measurement of real quantum devices' outputs must still be done through classical simulation, to double-check results and ensure correct operation. With the increasingly higher optimisation of specialised software libraries over commodity hardware accelerators, the rapid simulation of *small* quantum algorithms is becoming more and more easy to accomplish [36–39].

Multiple simulation methods for universal gate-based quantum computing have been developed to perform quantum information processing, from state vector simulation and tensor network contractions, to stabilisers theory, p-block simulation [15, 40], or branching-based techniques [41], as described in Table 1.1. The scaling of those methods depends on specific factors: the number of qubits N, the number of quantum gates m, the bond

Quantum simulation methods			
Methods	Memory	Run time	Precision
State Vector	$O(2^N)$	$O(m2^N)$	Exact
Density Matrix	$O(2^{2N})$	$O(m2^{2N})$	Exact
MPS/MPO	$O(N\chi^2)$	$O(N\chi^6)$	Approx.
Tensor Network	$O(2N+m)$	$O(e^W)$	Both
Stabiliser	$O(e^{m_T})$	$O(e^{m_T})$	Approx.

Table 1.1: Properties of quantum simulation methods.

dimension χ, the treewidth of the tensor network derived graph W and the number of T gates m_T. They can be compared over the scaling of memory and time requirements, the approximate or exact nature of results, and the capability of simulating the noise present in a quantum computer. A better exploitation of current classical computation resources may lead to efficient simulation of small quantum subroutines, or the offloading of part of the quantum information processing, possibly reducing or annulling the queries made to cloud-based quantum computers for problems requiring a low number of qubits.

Full state simulators leverage the Schrödinger-Feynman method, which keeps track of the whole description of a quantum system, either in the form of a *state vector* or a *density matrix*: this approach scales exponentially in terms of time and memory, but provides exact results, and may be adapted to simulate the noise regimes of current quantum devices. The simulation of quantum circuits using these approaches involves dense vector-matrix multiplications. This is the most common and general technique used to solve quantum circuits, and numerous implementations have been made available over the years, with various degrees of performance and efficiency [36, 42–44]. A single datacentre-grade GPU with 64 GB of Video Random Access Memory (VRAM) can simulate up to 32 qubits with the state vector approach, or 16 qubits with the density matrix approach, as 16 bytes are needed to store a double precision complex number. This technique can be efficiently distributed over multiple accelerators and compute nodes by slicing the state vector or the density matrix, though it suffers from diminishing returns in strong scaling, as increasing the qubit register size by one requires doubling the compute and memory resources.

Tensor based approaches, such as *Matrix Product States* and *Operators* (MPS/MPO) and *Tensor Network contractions* trade accuracy and expressiveness for computation speed, and provide partial results by probing a small portion of the final quantum state's probability distribution [13, 14]. The input qubits are converted in one-dimensional tensors, while single and double-qubit gates are encoded in two-dimensional and four-dimensional tensors,

respectively. Tensors are arranged and multiplied together following a specific ordering, which impacts the dimensionality of the intermediate tensors, in turn affecting performance [45–48]. These techniques are generally employed to simulate quantum algorithms which boast low circuit-level entanglement, achieving considerable speedups in time, and lighter memory requirements [10, 36, 47, 49–53]. As such, the result provided by these techniques will be the amplitude of a subset of correlated bit strings, or the expectation value of a measurement operator, both describing a subset of the solution's probability distribution. Algorithms such as the Variational Quantum Algorithm (VQE) or the Quantum Approximate Optimisation Algorithm (QAOA) are encoded in circuits that can easily take advantage of this reduced solution expressiveness.

At last, *Stabiliser simulators* work on the assumption that most or all the operations in a circuit are contained in the Clifford group [12], a *non-universal* set of gates which can be efficiently applied by following predefined rules and requiring little to no actual multiplication of variables. The applicability of this last technique is mainly geared towards QEC algorithms [54], which predominantly leverage Clifford gates. However, it is not efficient for simulating quantum circuits which leverage extensive superposition, a foundational property of most computationally-relevant quantum circuits.

Recent papers have demonstrated that GPU-based quantum simulation approaches can bridge the gap between the capabilities of current quantum computers and classical supercomputers [55], although hardware-specific parallel quantum simulation methods had been introduced long time before in both the single-GPU [56, 57] and distributed settings [58]. Plenty of review articles have addressed and measured the effectiveness of classical approaches for quantum circuit simulation. Some of those works were limited to state vector simulation approaches only [59–61], comparing the overall computational efficiency for a small set of quantum circuits on GPU accelerators, whilst others proposed hybrid Central Processing Unit (CPU)-GPU systems for heterogeneous quantum simulations that could outperform GPU-only implementations [62] alongside communication optimisations for this hybrid setting [63]. Other works have, instead, only considered the performance of tensor network contractions, focusing on problem-specific optimisations to improve GPU-based simulation performance [64], or on general approaches for tensor network simulation [65]. It must be noted that most of the benchmarking suites available in the literature are meant to test real hardware performance [66, 67], without focusing on the performance boundaries between classical simulation techniques and quantum computers.

1.1.4 Quantum computer technologies

In the last decade, progress in the various technologies used for building quantum devices has reached commercial applications. Three main candidate technologies can be identified at present time, namely photonic circuits, ion traps, neutral atoms and superconducting chips [68].

Photonic circuits generally make use of Mach-Zehnder interferometers and waveguides to map probabilistic quantum gates through photon-photon interactions, which are then measured through single photon detectors to reconstruct the circuit's output distribution. The optical properties of these devices are strongly dependent on both the materials and the production processes employed, and in turn impact the accuracy of the devices themselves. Although this technology features the fastest gate times, researchers are facing fabrication and scaleability concerns, which, together with their characteristic limited programmability, is hampering their adoption.

Ion traps use electromagnetic pulses to spatially constrain single ions, which are then addressed through specific laser wavelengths for the application of quantum gates, measuring their fluorescence to reconstruct the state of the qubits. Despite their excellent quantum gate fidelity, ion traps face serious scaling issues, besides the comparatively slow operational times with respect to other technologies.

Neutral atoms refer to the usage of Rydberg atoms in opto-magnetic traps, which lock in place, address and move atoms. This approach shares some conceptual similarities with ion traps, with the advantage of guaranteeing all to all qubit connectivity. However, implementation of quantum gates are severely limited, and multi qubit gates require a slow interaction between proximal atoms.

At last, superconducting quantum computers make use of Josephson junctions to build superconducting current loops which boast separate energy levels addressable via microwave pulses. The cheap costs of fabrication, fast gate implementations, and ease of scaleability, since it can be manufactured with a nanoscale photolithography process similar to those of classical Complementary Metal-Oxide-Semiconductor (CMOS) transistors [69], have made this technology more accessible and compelling than the others, despite the relatively high error rates and low coherence of information. Amongst the superconducting qubit implementations, one of the most widely employed is the transmon qubit. It acts as an anharmonic quantum oscillator, where a Josephson junction, a superconductive electronic component, introduces a non-linearity in a standard Inductance-Resistance-Capacitance (LRC) circuit that significantly enlarges the energy band gap between the first and the second excited states. This results in a quantum system with two discrete states, a so-called $|0\rangle$

ground state and an excited $|1\rangle$ excited state, whilst higher energy quantum states remain unused under nominal conditions. Superconductive qubits need to be operated at extremely low temperatures, in the milli-Kelvin regime, in order for the Josephson junctions to reach past their critical temperature regime.

Amongst these candidate options, a large portion of the research and industrial interest has been devoted to the latter technology. Given this, and the aforementioned advantages of the technology, this thesis mainly focuses on the superconducting quantum computers. In this context, universal gate-based quantum computers have surpassed the one-thousand *noisy* qubits mark [70], and classical computers have already fallen behind in terms of fully simulating the behaviour of such large machines in specific problems [71], although some of these demonstrations have been highly contested [18]. However, quantum computers are still prone to errors and faults, as the technology is far from being mature.

1.1.5 Intrinsic noise characterisation

Each qubit implementation faces reliability challenges, as qubits are exceptionally complex to implement and control. Amongst the available technologies, the superconducting transmon qubit is highly promising and widely adopted. Superconducting qubits encode quantum information in the two-level system of an anharmonic oscillator circuit, which is built using Josephson junctions. The main engineering challenge of this technology stems from the requirement of keeping the whole quantum chip well below the critical temperature necessary for observing a supercurrent. This generally leads to operational temperatures of around $10\ mK$, but novel technologies show promising performance even at temperature over $200\ mK$ [72]. The biggest hurdle is thus to isolate the system enough to preserve its quantum properties [73]. However, in order to couple the system with control logic, isolation from the environment is inevitably compromised. This makes the whole system extremely susceptible to manufacturing quality of the qubits, due to the complexity of controlling the stimuli, cross talk amongst proximal qubits, and the readout of the results, in addition to introducing inevitable interactions with the environment through slight differentials in temperature and pressure, and electromagnetic interference [73, 74]. These spurious interactions thus introduce unwanted noise into the system.

A superconducting device's ability to retain information varies over time. It can be quantified through two separate metrics, the *spin-lattice coherence time (τ_1)* and *spin-relaxation time (τ_2)* [19]. The former, τ_1, refers to the natural energy decay time of an excited qubit in state $|1\rangle$ to the ground state $|0\rangle$. It is the argument of the inversely decaying exponential defining the probability of a qubit to have collapsed to the ground state as a function of

time. The latter, τ_2, is the minimum interval before a qubit's state gets affected by external interference or by neighbouring qubits, thus degrading to a classical mixture of states. It is the argument of another inversely decaying exponential, which instead represents the probability over time for a qubit to have degraded into a classical mixture of states. These metrics are used to model the decoherence of a two-level quantum system. Current qubit implementations grant a quantum state stability slowly transitioning towards coherence times greater than 1.4 *ms* [75–77], and thanks to special gate composition and pulse scheduling techniques, quantum gate fidelity now ranges from $\sim 85\%$ to upwards of $\sim 99\%$ [78–82]. Technological improvements, however, are reaching their limits, and any further upgrade is bound by design and development costs [68, 83, 84]. As a consequence, retention and relaxation errors shorten significantly the computationally-useful lifetime of a qubit, inducing an alteration in the qubit state, together with independent gate and measurement error rates, which are referred to as *intrinsic noise*, a general phenomenon that inevitably reduces quantum information retention in the qubit, ultimately leading to information loss. Superconducting quantum computers are also subject to imperfections in the application of quantum gates. This leads to both errors in the preparation of quantum states and errors in the output measurements.

The intrinsic noise profile of a quantum computer is rarely simulated up to this degree of precision, as doing so requires a thorough physics-informed model. It is common practice in the literature to compose such *intrinsic noise models* through the usage of *Pauli operators*, under the umbrella terminology of *depolarisation error models* [85]. The uncorrelated nature of these models follows the definitions of intrinsic noise provided in the literature [19]. A general depolarising noise model is parameterised by a *physical error rate* p, which is meant to match the average measured error rate of a real quantum computer. This physical error rate rules the probabilistic insertion of a Pauli operator after each gate operation $\mathcal{O}$, or before each measure operator $\mathcal{M}$ in a quantum circuit. When performing two-qubit gate operations, an equally sized error gate is similarly employed, defined as the tensor product of two independent $\mathcal{E}$ noise operators: $\mathcal{E}_2 \doteq \mathcal{E} \otimes \mathcal{E}$. It is important to underline that this model introduces errors *independently* across all qubits, such that those errors show *no correlation* amongst each other.

An accurate definition of this intrinsic noise model would require the physical error rate to increase over time, following the same behaviour of τ_1 and τ_2. Considering that the coherence time of modern superconducting quantum computers is orders of magnitude longer than the execution time of a single quantum circuit, it is reasonable to assume for the physical error rate to remain constant throughout a simulation. One of the most widely employed depolarising noise models is the *superconducting-inspired 1000 ns cycle* (SI1000)

noise model [86].

1.1.6 Quantum error mitigation and correction

The most notable obstacle preventing quantum technology from thriving is reliability. Despite the tremendous technological advancements of recent years, quantum hardware designers are still facing issues in merging the reliability and scalability aspects of quantum computers [87–92]. Significant improvements are still necessary to fill the gap between current NISQ devices and large-scale fault-tolerant requirements of quantum algorithms. To reach past this limit in the quantum setting, the main solution is to employ error detection and correction algorithms able to work with quantum information.

Single qubit-level errors can be mitigated and corrected via iterative hardware-level error mitigation and calibration techniques [93–95]. This is generally done by leveraging extensive physical measurement data to calibrate control signals and optimise quantum gate implementations. These approaches are highly resource intensive, and aim at normalising qubit accuracy and quality across a single quantum chip, though their effect on improving quantum information retention are limited.

Error correction for classical information, in a simplified picture, makes use of data replication and parity measurements to reduce the error rate associated to memories or communication protocols: making a reliable system out of a set of unreliable parts [96]. Similarly, QEC [21, 97, 98] employs multiple physical qubits to encode quantum information in a higher level structure, a logical qubit [21, 99–101]. Due to the no-cloning theorem [102], however, there is no way to accurately replicate *arbitrary* quantum information, so one does not simply reuse classical error correction algorithms. As such, a set of *data* qubits are used to encode information, *observer* qubits are used to detect joint parity changes of the data qubits. This parity information is obtained from stabiliser measurements, which are non-entangling projective measurements storing the joint parity of a set of *data* qubits into an *observer* qubit that act as the foundation of QEC [96, 103]. Each observer qubit shares one data qubit with *at least* one other observer qubit, and computes the joint parity of *at least* two data qubits. An example of parity extraction for a bit-flip repetition code is presented in Figure 1.4

Once a QEC measurement round has been completed, the parity information of all the stabilisers is collected into a *error syndrome*, a classical vector of boolean variables. A *true* value represents an odd parity for that stabiliser, whilst a *false* value represents an even parity. In this context, stabilisers that have measured an odd parity are labelled as *defects* of the syndrome. The *error syndrome* is then processed to locate and correct errors in the

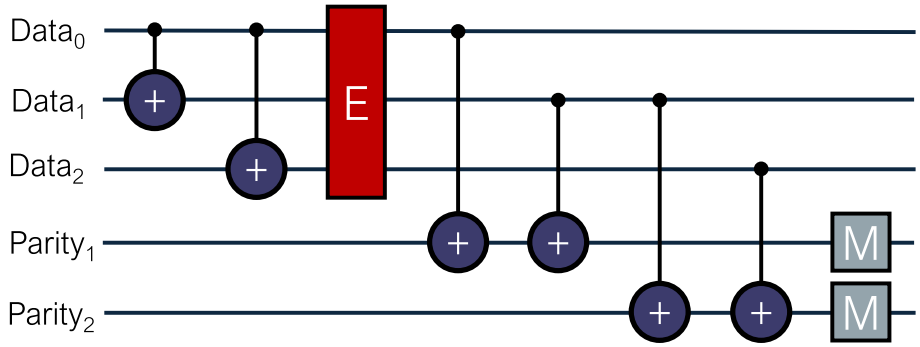


Figure 1.4: Parity extraction in the repetition code.

data qubits for a given computational basis [99, 104, 105]. Commonly used syndrome bases are the Z-basis for bit-flips and the X-basis for phase flips, although it must be noted that the set of possible syndromes is infinite, as it is possible to define an infinite number of arbitrary unitary operators on a qubit. As such, the error correction capabilities of surface codes are always going to be limited by the expressiveness of the syndrome set chosen.

The first implementation of a surface code was presented on a bidimensional mesh with periodic boundaries (i.e. a torus), distributing data and observer qubits in a draughtboard pattern [106], later followed by a single error correcting code [107]. Subsequent implementations circumvented the periodic boundary condition by rotating the pattern, as in the rotated surface code [108], opening the way for engineering real device implementations [98]. This reduces the planar connectivity requirement amongst qubits to a degree of at most four, making this code easily implementable in current quantum hardware. The idea of this approach is to spatially relate all the stabiliser measurements that detect an error over a decoding step. This is done by matching the error syndromes closest to each other, forming loops that can be shrunk down to a point and then applying the corresponding syndrome correcting operators. Quantum error correction codes are generally parameterisable by *code distance* d and *number of repetitions* r. The distance measure is directly proportional to the number of data qubits over which quantum information is replicated: the surface code requires 2^d physical qubits for each logical qubit at distance d. The number of repetitions identifies how many times the stabilisers in the QEC code must be re-measured over time to complete a round of correction, and generally scales as $O(d)$. As such, a single stabiliser qubit may be measured multiple times during the execution of a single round of correction, producing separate stabiliser measurements.

The process of decoding an error syndrome involves making a prediction of its most likely generators, in order to issue corrective operators apt at reverting its effects and preserving the quantum information encoded therein [109, 110]. The set of stabilisers includes the

generators of the parity check matrix of the QEC code. The relations in the parity check matrix can also be represented in a Tanner graph, a connected graph where each vertex represents one stabiliser qubit and each edge represents an error mechanism between two stabilisers. Edges may thus be due to error mechanisms on a data qubit shared between two stabilisers qubits, or due to error mechanisms on the stabiliser itself. The Minimum Weight Perfect Matching (MWPM) decoder, one of the most commonly used decoders, leverages the Tanner graph to infer the errors sources from the error syndrome [110–114]. Other decoding approaches include tensor network predictors [115], union-finding [116], belief propagation [117], and machine learning [118].

There is experimental evidence of QEC lowering both the theoretical and experimental the error rate of quantum devices [99, 104, 105, 119–121] by assembling logical qubits that improve onto the error rate of the noisy physical qubits that constitute them [93–95]. However, this comes at a high resource cost: multiple physical qubits are needed to encode a single quantum logical qubit, with an overhead going from $7\times$ [22] to upwards of $49\times$ [98, 120, 121]. Moreover, QEC requires high bandwidth and low latency classical processing power to keep up with the data generation rate of quantum computers at the decoding step. This throughput requirement scales with the dimension of the surface code, to the point where *ad hoc* hardware solutions are being proposed and developed [122–124].

1.1.7 Radiation induced faults

Despite recorded successes regarding quantum fault tolerance, the lurking shadow of radiation-induced faults remains a widely undiscussed topic, whilst being a serious threat to quantum computing progress. In fact, recent experiments and simulations have highlighted the high susceptibility of superconducting qubits to natural ionising radiation [21, 23, 24, 91, 92, 101, 125–132]. These unpredictable stochastic events alter the state of qubits by forcing them into decoherence for long periods of time. From a physics standpoint, the fault mechanism involves the generation of electron-hole ($e^- - h^+$) pairs in the silicon substrate of the quantum chip, which in turn give rise to phonons travelling in the device. Phonons lead to the localised breaking of Cooper pairs in the Josephson junctions atop the substrate, forming quasiparticles that disturb the encoded quantum information. Such radiation events have been unequivocally identified as one of the root causes of spatio-temporally correlated faults. An example of this physical process is schematised in Figure 1.5.

Most particle interactions deposit enough energy to overcome the threshold and trigger a fault, with the rate of occurrence of such events measured at once every *ten seconds* [21]. Radiation induced effect are transient in nature, as the energy absorbed by the substrate is

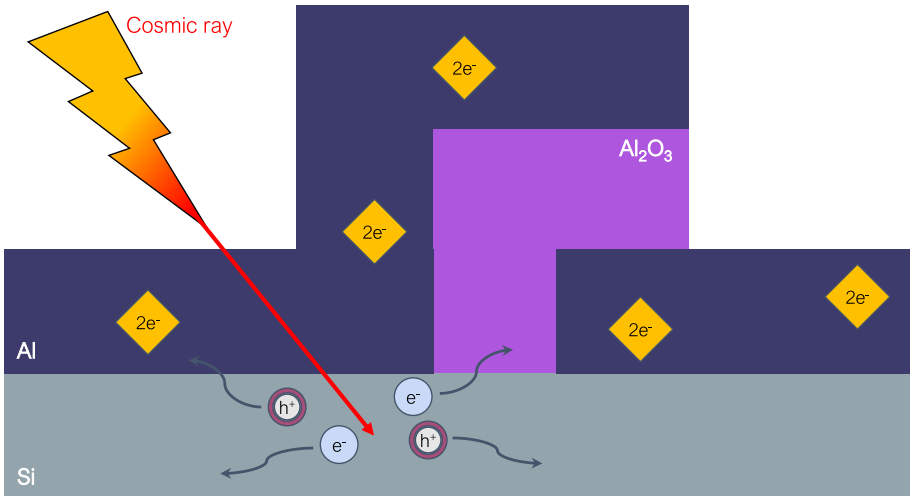


Figure 1.5: Radiation impact and generation of electron-hole pairs.

gradually dissipated by recombination soon after the impact and by diffusion later [133]. This forces qubits into a decoherent state lasting from 25 *ms* to upwards of 100 *seconds* [134–136]. For reference, a single NISQ quantum circuit execution lasts for a few hundred *nanoseconds*, as such these kinds of faults can corrupt the execution of hundreds to thousands of measurement shots, making sampling ineffective. Such intensity and persistence is orders of magnitude larger than standard device-intrinsic noise events, effectively impacting multiple qubits simultaneously, proportionally to the charge deposited on the silicon substrate of the quantum chip, with aluminium-based ground plates being the most prone to incur in long-lasting quasiparticle poisoning [137]. Since the logic state of the qubit is not deterministic, rather than simply inverting a bit-state as it would happen in classical computing, ionising radiation probabilistically modifies or erases quantum information.

Observation I.I

Radiation induces correlated faults in multiple physical qubits in current NISQ machines, a type of error syndrome that lies outside of most QEC codespaces.

The effects of noise and radiation on an arbitrary logical qubit $|\Psi\rangle$ are compared in Fig. 1.6, implemented with five physical qubits correlated by a QEC code. Noise affecting one physical qubit, being well characterised, can be compensated for without affecting the logical qubit state. The charge deposited by the radiation event instead spreads across the whole physical substrate, jeopardising QEC efficacy by generating correlated alterations in multiple physical qubits, which translates at the logical qubit level in an error syndrome lying outside the codespace.

This goes in stark contrast with CMOS transistors, where a fault happens only if the

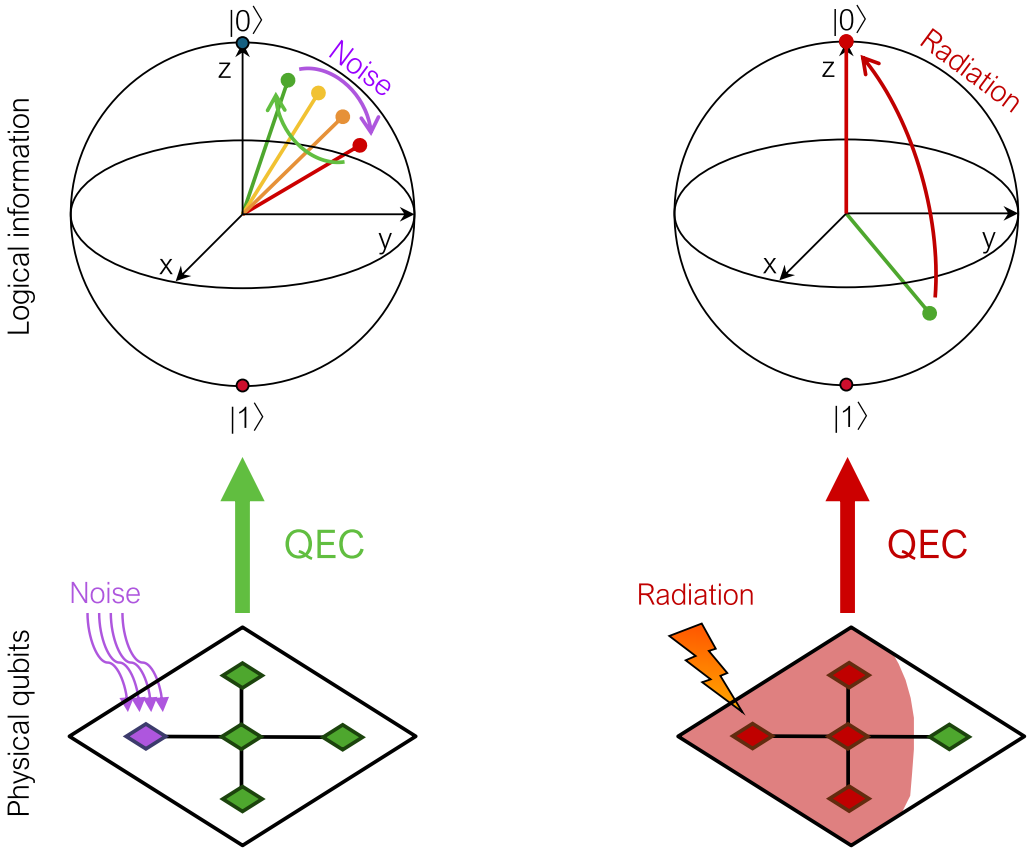


Figure 1.6: Comparison of the effects of noise and radiation on logical qubits.

deposited energy is sufficient to cross a critical charge threshold and reverse the transistor state [133, 138]. With the CMOS feature size shrinking trend, the probability of having multiple bit upsets is increasing [133], being already over 10% in 28nm technologies [139]. However, most interactions almost exclusively caused by high-energy neutrons, while other impinging particles such as muons, gamma rays, or low energy neutrons do not deposit sufficient charge to trigger the fault in a transistor [133].

In a qubit, even a single Cooper pair breakage is sufficient to disturb the equilibrium, thus affecting the encoded quantum information [23, 24]. Since the energy required to break a Cooper pair is in the order of milli-electronvolts (meV), even very light interactions are sufficient to induce a fault [134]. This makes quantum devices prone to errors generated not only by relatively rare highly interacting particles such as neutrons ($13\ n/cm^2/h$ at sea level), but also from the overly abundant flux of muons ($60\ \mu/cm^2/h$ at sea level) [140], which end up being the first cause of radiation events [129], exacerbating the probability of observing a radiation-induced fault in quantum devices. Moreover, given that this deposited charge spreads isotropically in the silicon substrate from the particle impact point, the documented area-of-effect of these events usually involves most, if not all, of the qubits present on

the quantum chip. Field experiments on a 25 qubits array showed simultaneous chip-wide radiation-induced faults every tens of seconds [126]. The reported error rate is several orders of magnitude higher than the one of modern CMOS technology. As a reference, the whole Titan supercomputer (composed of 14,000 nodes) has an error rate in the order of one error every few hours [141]. Crucially, while this topic has been deeply investigated for mission-critical applications on classical transistor-based systems, such as those used in high performance computing, space exploration, real-time embedded systems or remote sensing control, the detection and correction of transient faults in quantum computers and quantum circuits is yet to be fully characterised.

Observation I.II

The radiation-induced fault rate of qubits is orders of magnitude higher than the one of traditional transistors.

Some of the currently proposed hardware-level solutions to radiation either leverage underground facilities to reduce the external radiation flux [129, 135, 142–145], employ on-chip simultaneous error correction specific to frequent and small independent errors [146, 147], sever the superconductor-substrate coupling [148], engineer higher energy gaps [21, 101, 149, 150], use alternative materials and construction techniques for the quantum chip's substrate [151], or employ per-qubit quasiparticle barriers and traps [130, 152–156]. Notably, shielding techniques are impractical on a scaling standpoint, and can not prevent all radiation events from reaching the substrate, while gap engineering, suspended qubits, and quasiparticle traps incur in additional manufacturing costs and scaleability constraints of superconducting quantum chips on a *per qubit* basis. Another option would be to replicate quantum chips, but the redundant chips, to maintain quantum properties, should share a quantum network and should be able to entangle qubits amongst different chips [157]. These solutions are far from being final, as high energy events are still very common and disruptive.

The presence of radiation-induced faults prompts the usage of post-selection, discarding results altered by radiation poisoning [21], thus indirectly increasing the already exorbitant operation cost of quantum computers. Albeit being helpful in the short term, this is not a definitive solution to the problem of radiation and has very limited effectiveness, all the while being inefficient for scaling purposes and wasteful in terms of resources.

1.2 Research objectives

The main topic of the thesis revolves around how to model, simulate and compensate radiation faults in superconducting gate-based quantum computers, both at the qubit and algorithmic levels of abstraction. This follows the research gap between the physics of radiation events in quantum computers and current reliability solutions in the field. In fact, while intrinsic noise suppression in quantum computers is well studied, the areas involving modelling of radiation events, their interaction with quantum algorithms, QEC codes and classical decoders remain uncharted, since none of them have been designed with multiple correlated qubit errors in mind. Investigating the correct simulation method to enact this fault model is another important step in creating a useful and practical tool for the scientific community. This means first understanding how to represent radiation's effects at the quantum circuit level of abstraction. This characterisation opens up new avenues of research, testing and validating software and hardware solutions, and considering the effects of logical-level faults in hybrid quantum algorithms. Moreover, it will stimulate new *algorithmic* solutions to detect and eventually correct radiation-induced faults in quantum devices.

The main contribution of this thesis is to create a common ground for radiation reliability research between the computer science and physical levels of abstraction. Such contribution will spring forth transdisciplinary research, moving the community closer to the goal of fault-tolerant quantum algorithms.

1.3 Outline

The thesis is structured as follows. In Chapter 2, a general characterisation of a set of quantum circuits is presented. From there, the experimental grounding of classical state vector and tensor network contraction methods is provided, with experimental measures of distributed quantum simulation performance on modern accelerators. Chapter 3 introduces the very first iteration of the radiation fault model, compatible with state vector simulators. This model is first used to investigate the efficacy of small QEC codes, namely repetition and surface codes, in compensating simulated radiation faults. Afterwards, Chapter 4 defines a more fine-grained evolution of the radiation fault model, specifically adapted for stabiliser-based simulation methods. Thanks to better scaleability, this second and more flexible model is then used to test a suite of large rotated surface codes over a set of syndrome decoders. A novel algorithm for real-time detection and identification of radiation events in subround time is presented, along with a tentative approach to reduce the logical error rate of QEC

at the decoder level. In Chapter 5, the quantum chip representation in the radiation fault model is extended to accommodate for hardware hardening solutions. The effectiveness of this cross-layer hardening is measured over a set of rotated surface codes, showing notable improvements with respect to the baseline non-hardened quantum chip. Reaching outward from QEC level-faults, Chapter 6 models the effects of radiation at the logical, post-QEC qubit level, investigating the effects of fault propagation in hybrid algorithms. Specifically, faults in the quantum portion of an image classification neural network are propagated to the following classical layers, outlining the misclassification rate of the whole model over two datasets. Alas, the thesis' conclusions are drawn in Chapter 7.

Chapter 2

Methods for distributed quantum simulation

The frontier of QC simulation on classical hardware is quickly saturating the hard scalability limits imposed by computation and memory bottlenecks. Nonetheless, the need to simulate large quantum systems classically has never been more pressing, as the NISQ devices are yet to reach fault tolerance, thus still requiring validation. The two most widely employed exact simulation methods, state vector and tensor network contractions, boast specific limitations. The exponential memory requirement of state vector simulation, when compared to the qubit register sizes of currently available quantum computers, quickly saturates the memory capacity of the top High Performance Computing (HPC) machines. Tensor network contraction approaches, which encode quantum circuits into tensor networks and then contract them over an output bit string to obtain its probability amplitude, still fall short on finding an optimal contraction path, a notably Nondeterministic Polynomial time (NP)-hard problem.

2.1 Objectives

The purpose of this chapter is to understand where the limit for efficient quantum simulation on classical hardware lies, emphasising the computational aspects, such as distributed performance, scaleability, time and memory footprints of quantum algorithms, with the

This chapter refers to the contents of the article "State of practice: Evaluating GPU performance of state vector and tensor network methods", written by M. Vallero et al. and published in the Future Generation Computer Systems journal [158].

objective to find quantum circuit features that correlate to simulation performance. The questions I seeked an answer for are:

- **RQ1**: What is the performance of state-of-the-art quantum simulation methods?
- **RQ2**: Which topological features of a quantum circuit correlate to simulation performance, and which simulation approach is more suitable?
- **RQ3**: Are there limitations to distributed quantum simulation, and if so, can those be predicted?

I have considered a set of eight widely used quantum subroutines, each in different configurations, performed both single and distributed scaleability experiments on the Leonardo supercomputer provided by CINECA, correlating performance measures with the metrics characterising the circuits in the benchmark, and identifying what rules the observed performance trends. Specifically, I have performed distributed sliced tensor contractions, analysing the impact of pathfinding quality on contraction time and correlating both results with topological circuit metrics. From such observations, given the structure of a quantum circuit and the number of qubits, I highlight how to select the best simulation strategy, showing how pre-execution circuit analysis can guide and improve simulation performance by more than an order of magnitude.

I show that, by profiling quantum circuits with the approach presented in this chapter, the simulation time can reach a speedup of *up to one order of magnitude*, especially for large quantum circuits, on a single GPU. Furthermore, results from distributed tensor contraction simulations highlighted speedups of more than $364\times$ with respect to single GPU performance, tracing the impact of pathfinding quality on the contraction performance, obtaining speedups of up to $4.79\times$ through tuning. The proposed circuit metrics to performance correlation is achieved by characterising a purposefully selected suite of well-known quantum circuit subroutines according to objective metrics, and checking how those scale with respect to the size of the quantum circuit. All the circuits considered are parameterisable over the number of qubits in the system, and some of them feature additional customisation parameters, such as layer repetition. Moreover, they have been selected as to have practical applicability in terms of exact simulation. These same circuits have been simulated on CINECA's Leonardo supercomputer, using both state vector and tensor network contraction methods through NVIDIA's *cuQuantum* library [36], highlighting which one boasts the better performance for each workload. This chapter proposes a practical methodology to pick the most efficient simulation strategy according to a given set of static characteristics of the circuit.

The foundational topics on quantum information theory and quantum circuits have been introduced in Chapter 1, Sections 1.1.1 and 1.1.2. The rest of the chapter is organized as follows. Section 2.2 introduces two classical algorithms for circuit based quantum simulation, state vectors and tensor network contractions. Section 2.3 gives a definition of the metrics and of the quantum circuits considered for this study. Section 2.4 characterises the quantum circuits according to the aforementioned metrics, then presents performance results with respect to execution times and peak memory occupancy, scaling of distributed tensor network contractions and impact of pathfinding resources on tensor network contraction times. Lastly, Section 2.5 concludes the chapter by expanding on the hereby presented chapter by opening new paths for investigation.

2.2 Quantum circuit simulation

Quantum simulation can refer to two concepts: either the usage of real quantum devices to simulate other quantum systems, or the usage of classical machines to compute the theoretical output of a quantum algorithm. For the sake of clarity, this thesis will always refer to the latter when talking about *quantum circuit simulation*. The objective of simulation is not to thwart the development of real quantum devices, but rather to validate the outputs of such machines against their theoretical expected outputs. Moreover, given the still relatively scarce availability of real quantum devices and the limitations of current device technology, such as low coherence times, quantum simulators provide a means for validating new and possibly deeper quantum algorithms. There are various approaches to simulate a quantum circuit, the two main ones that provide exact results are: *state vector* simulation [44] and *tensor network* contraction [45], which are detailed in the following subsections.

The aim of this chapter is to identify how the inherent structure of a quantum circuit can affect the execution time, so as to preemptively identify which simulation strategy works best for which kind of circuit, by making use of *ad hoc* metrics. These metrics provide a description of the overall structure of the quantum circuit, highlighting critical areas for the improvement of modern simulators. It will be possible to infer that any other quantum circuit, reflecting the characteristics provided in this chapter, will scale similarly in terms of simulation. The main current limitation of state vector simulators is the inherent exponential memory blowout linked to the system size, that has been tentatively compensated through state vector compression [43], however the distributed application of vector-matrix multiplications still scales exponentially on the system size. Tensor networks have already shown promising results, with useful applications in the field of verification of real quantum computer's

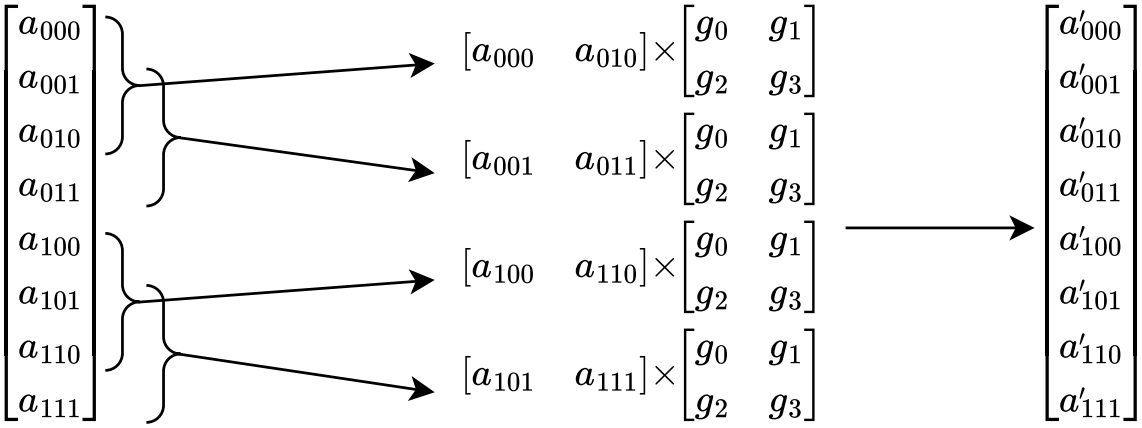


Figure 2.1: State-vector multiplication of a single-qubit gate.

outputs [159–161], however due to the limited exploitation of the internal structures of the circuit-derived graph representation, the contraction path used is not granted to be optimal.

2.2.1 State vector simulation

Quantum states are represented by a wave function, which can be encoded into a state vector. Given any quantum system of N qubits, its corresponding state vector will contain 2^N complex probability amplitudes, one for each possible output bit string. Quantum gates are applied by splitting the state vector into smaller vectors of size equal to that of the gate to be applied, then each sub-vector is multiplied with the gate matrix and the resulting sub-vectors are reassembled in the evolved state vector. The splitting operation is performed according to the qubits over which the operator is applied. This can be intuitively understood by considering the ordered set of output bit strings: the probability amplitudes corresponding to a given sequence of qubits, which depends on the qubit indices the operator acts onto, are grouped together. An example of this process for both single and two qubit gates is depicted in Figures 2.1 and 2.2 for a three qubits system. In the former case, a single-qubit gate is applied to qubit 1, so the amplitude pairs are grouped by following a *0-1* repeated scheme for the amplitude's index. In the latter case, a double-qubit gate is applied on qubits 0 and 1, so the amplitude pairs are grouped following a *00-01-10-11* scheme. All amplitude indices are written in little-endian and ordered top to bottom.

The state vector simulation's complexity scales linearly in time with respect to the number of gates [44]. However, the memory footprint of the state vector and the number of vector-matrix multiplications performed increase exponentially with the number of qubits present in the system to be simulated, so this approach is not scaleable indefinitely. To put that into perspective, it is possible to roughly estimate the number of atoms in the

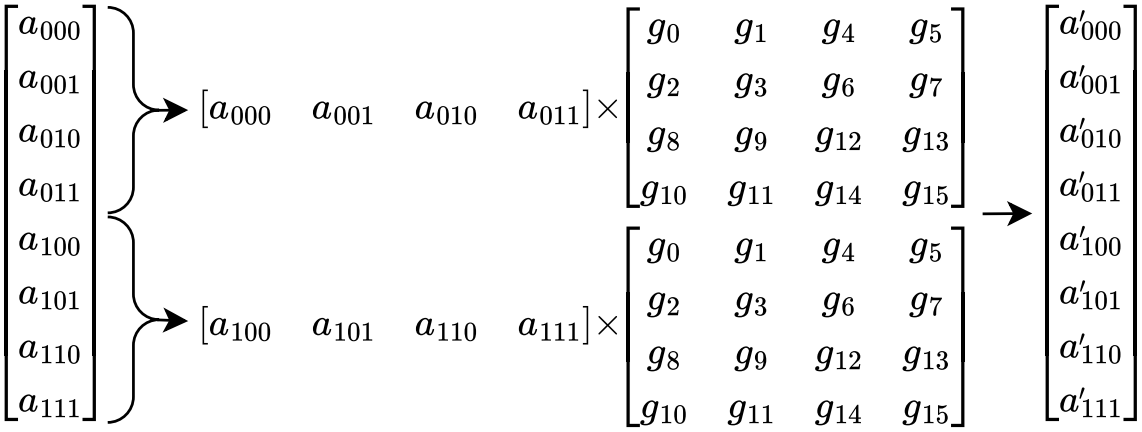


Figure 2.2: State-vector multiplication of a two qubits gate.

observable universe to be $10^{82} \approx 2^{270}$ [162]: this means that, by storing a single amplitude value inside each of them to build a state vector, one could only represent systems with upwards of 270 qubits. Well known quantum algorithms need significantly more logical qubits [163], and this is without considering the cost in terms of classical computation time, which may add up to reach unfathomable time scales [18]. Overall, the state vector approach is generally convenient when simulating small quantum systems, as it produces a full description of the output wavefunction. The task of state vector simulation inherently exposes chances for parallelisation, both for circuit specific approaches [64, 164], and general circuit-agnostic approaches [36, 165]. Despite the hindrances induced by communication and memory requirements, state vector simulation can still efficiently simulate quantum systems of limited size better than other competing methods [60, 166].

2.2.2 Tensor network simulation

Quantum gates and quantum basis states are represented by tensors. The graphical representation of a quantum circuit can be read as a Directed Acyclic Graph (DAG), where the vertices are represented by quantum gates or basis states and the edges are represented by the qubit *wires*. The input tensors are the basis states, encoded as follows

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.1}$$

All other gates use their standard matrix representation. The contraction of an edge corresponds to the multidimensional generalisation of the dot product between two tensors over a shared index. The measurement operators at the end of the quantum circuit are substituted by open indices. Whenever a full network contraction operation is performed,

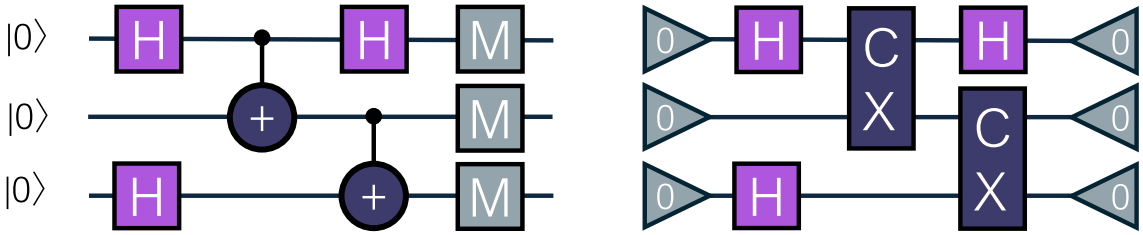


Figure 2.3: Quantum circuit conversion to tensor network.

the output open indices are closed with the conjugate tensors of the basis states that encode a specific bit string [13]. Doing so, followed by the contraction of the newly closed indices, produces the probability amplitude of the chosen bit string. Figure 2.3 provides graphical insight over this process. On the left, an example of a quantum circuit. In the centre, the circuit gets converted into a tensor network representation, where single and double-qubit gates become order-2 and order-3 tensors, respectively. On the right, after the tensor network contraction, one gets the probability amplitude of a specific bit string: by repeating this process over all output bit strings, one may reconstruct the whole state vector.

It is possible to use tensor networks to reconstruct the whole state vector, by closing and contracting the output indices over different bit strings, however doing so incurs in the same limits of the state vector simulator for storing the final vector.

Observation II.I

Tensor network contractions can reconstruct the full state vector, and such process can be trivially parallelised over different bit-strings.

The memory occupancy of the tensor network grows linearly with respect to the number of quantum gates and qubits in the system. This approach moves the complexity of simulation to that of finding an optimal contraction path for the tensor network, which is known to be a NP-hard problem [167]. Efficient heuristics specialised for quantum circuit-derived tensor networks have been proposed [50, 168], however there is no *catch-all* solution for this kind of problem. The pathfinding algorithm used in this chapter [45], despite representing the state-of-the-art for this class of problems, only strives to optimise having the lowest possible amount of floating-point operations across the whole contraction process, which does not prevent the formation of large intermediate tensors, something that inevitably reduces contraction performance. Besides, as it will be discussed in Section 2.4.4, the optimiser may easily be locked in a local minimum in some problems, whereas other problems feature smoother landscapes in terms of pathfinding complexity. If the contraction path is not optimal, it may lead to increased computation time, possibly making it less efficient

than state vector simulation altogether. There is abundant evidence supporting the fact that tensor contraction can simulate quantum circuits and systems that lie outside the reach of state vector methods [159, 169]. To the interested reader, I suggest some resources for tensor network theory by J. Biamonte [14, 170].

2.3 Benchmarks and metrics

To assess the performance of current state-of-the-art simulators and to select the most efficient one, it is necessary to use a set of metrics and quantum circuits which are relevant and well established in the quantum computing field. I relied on two quantum circuit benchmarking suites, which are widely recognised in the literature: SupermarQ [66] and QASMBench [67]. Both of these suites provide their own sets of quantum circuits, that, however, have been specifically selected for testing the hardware performance of real quantum devices. For this reason, some of these circuits boast little to no practical use in the context of noiseless exact simulation, such as the error correcting code circuits in SupermarQ, or the Greenberger-Horne-Zeilinger.

Observation II.II

Not all quantum circuits generally used for benchmarking are computationally representative in a classical simulation environment.

Furthermore, both suites introduce a list of metrics that characterise the topological nature of static quantum circuits. These metrics provide a measure of topological properties of the graph derived from the quantum circuit representation, letting me correlate such properties with the runtime performance statistics.

2.3.1 SupermarQ

In the SupermarQ [66] suite, six metrics are introduced, however I only considered the ones that have topological significance, referring to all elements which may alter the circuit-derived graph, that is the relative presence and the disposition of two-qubit or higher size quantum gates. All metrics range in $[0, 1]$, where higher is closer to 1.

Program communication: This metric measures the amount of interconnections present in a quantum program, computed as the ratio of the average degree of interaction of the quantum circuit in graph form with that of a maximally connected graph with a number of nodes equal to the number of qubits in the circuit. The term $d(q_i)$ is the degree

of the *i-th* qubit.

$$PC = \frac{\sum_i^N d(q_i)}{N(N-1)} \tag{2.2}$$

Critical depth: The critical depth represents the ratio between the longest chain of two-qubit operators and the total number of two qubit gates in the circuit. It gives a measure on whether the program's output heavily relies on distributed entanglement or not. n_{e_d} is the total number of two qubit gates on the circuit's critical depth path, while n_e is the number of two qubit gates in the circuit.

$$CD = n_{e_d}/n_e \tag{2.3}$$

Entanglement ratio: This measure is the ratio of the number of entanglement operators, n_e, with the total number of gates in the quantum circuit, n_g.

$$E = n_e/n_g \tag{2.4}$$

Parallelism: It is a measure of the number of concurrent operations made in the same time step, intuitively understood as the degree of *compression* of the quantum circuit. The number of gates n_g is compared with the depth d of the program, then such value is normalised with respect to the number of qubits n.

$$P = \left(\frac{n_g}{d} - 1\right)\frac{1}{n-1} \tag{2.5}$$

2.3.2 QASMBench

The metrics introduced in the QASMBench [67] suite are more tied to the architectural implementation of physical quantum devices. Follows the definition of the only topologically significant metric.

Entanglement variance: This metric defines the spread of entanglement amongst the qubits in the quantum circuit. It checks whether there are a few qubits which feature most of the connections towards the others, or if all qubits are sharing the same amount of entangling connections. In a quantum program with N qubits, the number of two-qubit gates acting on the *i-th* qubit is $n_{g_2}(q_i)$, while the average number of two-qubit gates per qubit is $\overline{n_{g_2}}$.

$$EV = \frac{\log(\sum_{i=0}^N (n_{g_2}(q_i) - \overline{n_{g_2}})^2 + 1)}{N} \tag{2.6}$$

Quantum circuits used as benchmark				
Circuit name	**Description**	**Total gates**	**Total multi-qubit gates**	**Ref.**
QAOA	Quantum Approximate Optimisation Algorithm	$\frac{3}{2}PN(N-1)+2N$	$PN(N-1)$	[171]
Random	Random quantum supremacy circuit	$(1-k)(N(\lfloor N/2\rfloor+N\%2))+kN^2$	$kN(\lfloor N/2\rfloor)$	[18]
QPE	Quantum Phase estimation	$\frac{N(N-1)}{2}+2N-1+\lfloor\frac{(N-1)}{2}\rfloor$	$\frac{(N^2-N)}{2}+N-2+\lfloor\frac{(N-1)}{2}\rfloor$	[172]
QFT	Quantum Fourier transform	$\frac{1}{2}N(N+1)+\lfloor N/2\rfloor$	$\frac{1}{2}(N^2-N)+\lfloor N/2\rfloor$	[173]
VQE	Variational Quantum Eigensolver	$L(5N-1)+N$	$L(N-1)$	[7]
Hamiltonian simulation	One-dimensional Hamiltonian time evolution	$3T(2N-1)$	$T(N-1)$	[66]
Hidden Shift	Find the shift s such that $g(x)=f(x+s)$	$3N+2M+\lfloor N/2\rfloor$	$\lfloor N/2\rfloor$	[174]
Bernstein-Vazirani	Hidden bit string extraction	$2N+M$	M	[11]

Table 2.1: The quantum circuits included in the benchmark suite.

2.3.3 Benchmark circuits

In order to provide a broad, extensive and scalable evaluation, I considered a specific set of circuits, selected to encompass some of the applications for quantum computing that do not leverage the presence of quantum noise. As such, their results are significant in terms of exact theoretical simulation. The list of circuits, with information regarding usage, scaling of the number of gates and references, is detailed in Table 2.1. All circuits considered can be freely expanded over any problem size, making them easily adaptable to benchmark future hardware and simulation platforms. For the sake of the hereby presented analysis, I tested all circuit qubit sizes in the range $[2-32]$, and subset of those quantum circuits in the range $[32, 40]$ using tensor contraction.

QAOA: The *Quantum Approximate Optimisation Algorithm* is a variational circuit that uses all-to-all connectivity to encode classical problems, such as the max-cut problem. The algorithm version used in this chapter is the vanilla one [171] with parameter $P=1$, although other versions exist, such as the one with ZZ-Swap gates [175].

Random: The *Random* quantum circuit, notably dubbed *Quantum Supremacy circuit* in its original formulation [18], is composed of multiple repeated layers of random gates picked from the set $\mathcal{G}=\{H, X, RZ, RX, RY, CNOT, CZ, SWAP\}$. The number of layers has been set equal to the number of qubits in the system. Given the random nature of the

circuit, it is possible to define a lower and an upper bound for the number of gates that can be found in the circuit. This circuit has been purposefully built to avoid any internal structure, so as to be as complex as possible to simulate. Despite achieving such a goal, the applicability for this subroutine to real world problems remains questionable at best.

QPE: The *Quantum Phase Estimation* subroutine is one of the foundational steps in Shor's algorithm, able to solve the order finding problem of a modulo function [3, 172].

QFT: The *Quantum Fourier Transform* is one of the most widely known quantum subroutines, which uses phase encoding to efficiently perform the Fourier transform [173]. This quantum circuit is a good candidate for simulation, since it is built by the recursive application of the same operator, possibly giving rise to exploitable internal structures.

VQE: The *Variational Quantum Eigensolver* is an hybrid quantum-classical algorithm that uses iterative optimisation to find the ground state of a molecule encoded in a quantum register [7]. The circuit is built by repeating L times a given structure. For the sake of simplicity, the version used in this chapter assumes $L = 1$. Its applications focus mainly on, but are not limited to, the simulation of the bond energies of chemical compounds.

Hamiltonian Simulation: This circuit encompasses a general approach for the encoding and simulation of the time evolution of a given Hamiltonian in a quantum computer [66]. The circuit is characterised by the repeated application of a quantum subroutine over $T = \textit{total\_time}/\textit{time\_step}$ iterations. The benchmark I considered computes the magnetic interactions of a monodimensional chain of spins.

Hidden Shift: The *Hidden Shift* quantum circuit is able to find the value s of a function $g(x) = f(x + s)$ by performing a single query to the function, leveraging the superposition of all the possible inputs [174]. The term k represents the percentage of bits equal to 1 in the binary representation of the shift value.

Bernstein-Vazirani: This quantum algorithm can solve the problem of finding out the bit string that satisfies a given function by performing a single query, whilst the classic algorithm would require at most N queries, where N is the total number of possible bit strings [11]. The term k is the percentage of bits set to 1 in the binary representation of the solution.

2.3.4 Leonardo supercomputer

All the simulation results presented in this chapter have been carried out on the Leonardo supercomputer managed by CINECA, in Italy. This machine is part of the network of pre-exascale supercomputers financed by the EuroHPC consortium. At the time of its inauguration in 2022, Leonardo was the fourth ranking supercomputer in the Top500 list, while

at the time of writing this thesis, it now occupies the tenth spot on that same leaderboard.

The architecture is subdivided in two main compute partitions. The first is the Data Centric General Purpose partition, boasting 1536 compute nodes equipped with dual Intel Sapphire Rapids 56-core Xeon Platinum 8480+ processors and 512 (8x64) GiB of DDR5 4800 MHz volatile RAM. The second is the Booster partition, with 3456 compute nodes equipped with four NVIDIA Ampere100 80 GB custom GPUs, and Intel Ice Lake Xeon Platinum 8358 CPU and 512 (8x64) GiB of DDR4 3200 MHz volatile RAM. All compute nodes are interconnected in a Dragonfly+ topology, via 200 Gbps NVIDIA Mellanox High Data Rate Infiniband switches.

Leonardo features a measured Linpack performance of 241.20 PFlop/s, and a theoretical peak of 306.31 PFlop/s, with a peak power consumption at maximum system load of about 7.5 MW.

2.4 Results

In order to find an answer to the research questions of this chapter, I started by characterising the quantum circuits according to the metrics introduced in Section 2.3, to later perform various simulations, with the objective to relate the metrics with execution time, memory occupancy, and in the specific case of tensor networks, distributed sliced contraction performance and pathfinding efficacy. The simulations have been run using the NVIDIA cuQuantum library (version v24.03)[36], adapted to the specific needs of the hereby presented analysis. This gave me access to three different GPU accelerated simulation backends:

- *qsim-cusv*: a state vector simulator that uses the *cuStateVec* backend.
- *qsim-cuda*: a state vector simulator that uses the *cupy* backend.
- *cutn*: a tensor network simulator that uses the *cuTensorNet* backend for contraction.

All experiments have been run on CINECA's Leonardo supercomputer. Apart from the distributed experiments, all other experiments have been run on a single node, using 8 cores of an Intel Xeon Platinum 8358 CPU, 128 GB of Random Access Memory (RAM) and one NVIDIA Ampere A100 64 GB GPU. The comparative simulations between state vectors and tensor contractions have all been limited to a problem size of 32 qubits, as that is the largest state vector that can be represented in a single available GPU, while larger tensor networks have been simulated later. Given the high computational cost of state vector and

tensor network methods, CPU-based simulation algorithms have not been considered, since they would not provide any meaningful comparison in terms of performance.

2.4.1 Quantum circuit properties

Following the results reported in Figure 2.4, each metric has been analysed independently. The metrics for the Random, Bernstein-Vazirani and Hidden Shift benchmarks have been averaged over 100 circuit samples, to compensate for the fact that these circuits do not have a constant topology.

The *program communication* keeps a constant value of 1 for the QAOA, QPE and QFT circuits, suggesting that those three algorithms feature at least a two-qubit operation with each of the other qubits in the quantum register. This means that the resulting topology of the circuit will be that of a fully connected graph. All other circuits, on the other hand, quickly drop towards values proximal to 0.1 as soon as the number of qubits in the system increases, meaning that most of the qubits do not interact directly. This can be explained by circuit structures where a few qubits interact with all the others, or by circuit structures where all qubits interact only with their closest neighbours. The main outlier is the Random circuit, which stabilises at a value of about 0.5, implying that, on average, each qubit interacts with at least half of the total number of qubits in the circuit.

The *critical depth* starts at value 1 for most circuits at low qubit sizes and rapidly drops towards the range $[0.08, 0.22]$ for the QAOA, QPE, QFT, Random and Hidden Shift benchmarks. This, together with the *program communication* score, means that the highly entangled structure of the first three circuits is not due to a chain of two qubit operators. The Hidden shift and the Random circuits, having both low *program communication* and *critical depth* scores, imply that the derived graph structure is sparsely connected, with a few "central" qubits sporting most of the two qubit gates towards all other qubits. The remainder of the quantum circuits in the test suite maintain a constant value at 1.0. This, together with the *program communication* metric implies that the graph structure of the VQE, Hamiltonian simulation and Bernstein-Vazirani circuits can be reduced to that of a single chain of nodes.

The *entanglement ratio* attains its maximum value in the QPE and QFT benchmarks, as those circuits are mainly composed of two-qubit gates. The QAOA and Random circuits are composed of 40% to 60% by multiple qubit gates, with the former saturating at 60% as the size of the system increases to 32 qubits, whilst the latter, given its non-deterministic structure, boasts an average of about 50%. The other circuits, the Hamiltonian simulation, VQE, Hidden shift and Bernstein-Vazirani, are mainly made of single-qubit gates, which can

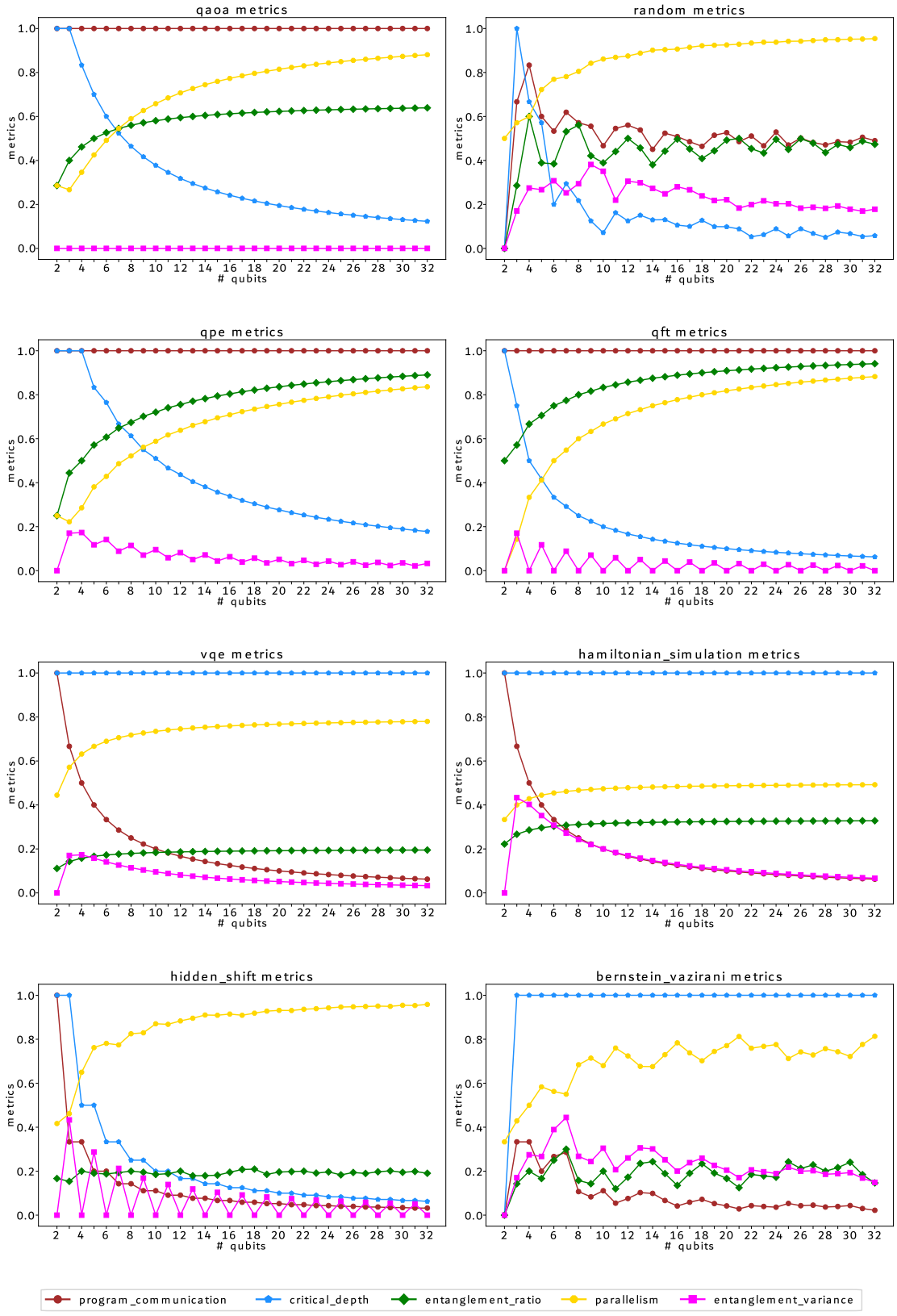


Figure 2.4: The scaling of the considered metrics on the benchmark set.

get easily processed during tensor network contraction [45], as such their *ER* scores are lower, ranging $[0.15 - 0.35]$.

The *parallelism* metric grows with respect to the circuit size for all the algorithms considered, with initial values ranging from 0.0 for the QFT to 0.5 for the Random. Generally, however, the metric's value saturates to different levels, with the QAOA, Random, QPE, QFT and Hidden Shift circuits passing the threshold $P > 0.8$ for systems sizes of 32 qubits, suggesting that the density of their derived topology is very high. On the other hand, the VQE and Bernstein-Vazirani circuits saturate in the range $[0.6 - 0.8]$, suggesting a slightly more sparse topology. The Hamiltonian simulation circuit saturates at value $P \approx 0.5$, highlighting its dependence on sequential processing of quantum information and a lower topological density.

The *entanglement ratio* rapidly approaches *zero* for almost all circuits considered in the benchmark suite. Notably, the QAOA circuit has a *constant* variance value of 0.0, meaning that independently of the circuit size, the number of two qubit operators is evenly split amongst all the qubit in the system. The QPE, QFT, VQE and Hidden Shift algorithms see a rapid decrease in the metric's value, approaching the range $[0.0 - 0.05]$ for circuits of 32 qubits, again hinting at the fact that most of the qubits take part in a similar amount of multi-qubit operations. The only two exceptions are the Random and the Bernstein-Vazirani circuits, which instead have a higher value of $ER \in [0.18 - 0.25]$. In the case of the Random circuit, this is due to the fact that the structure of the circuit does not follow a predefined scheme, whilst in the Bernstein-Vazirani circuit it is directly depended on the number of 1s present in the solution binary bit-string of the oracle function.

2.4.2 Single GPU simulation performance

Figure 2.5 details the memory requirements scaling for a general state vector simulator and the tensor network representations of all the circuits considered in the benchmark.

The memory occupancy for both the *qsim-cuda* and the *qsim-cusv* simulators is the same, as they both have to store in memory the whole state vector of complex probability amplitudes. Each probability amplitude is stored as a *complex-64* single precision binary number, scaling exponentially in memory, since the size of the state vector is $N = 2^n$, with n being the number of qubits in the system. The state vector is updated when applying new quantum gates, however its size remains unaltered, regardless of how many subsequent operations are applied to it.

The tensor network representation instead stores the initial state as a sequence $(1, 2)$ tensors, and each quantum gate as either a $(2, 2)$ order-2 tensor, in the case of single-qubit

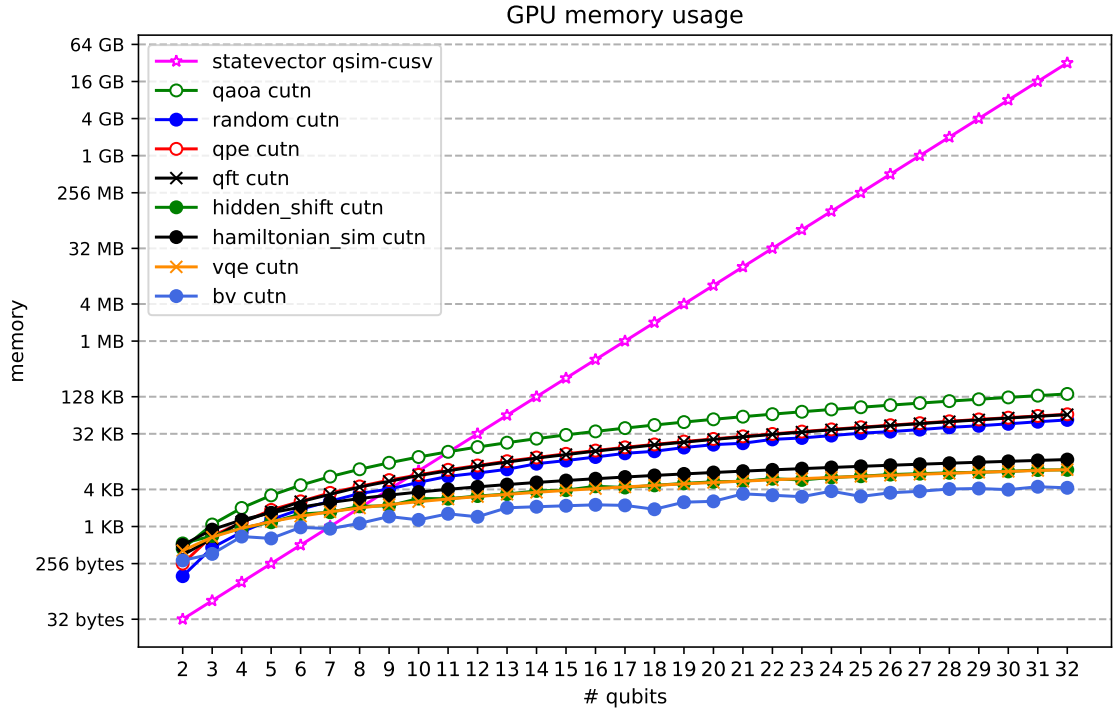


Figure 2.5: Memory usage scaling for state vectors and tensor networks.

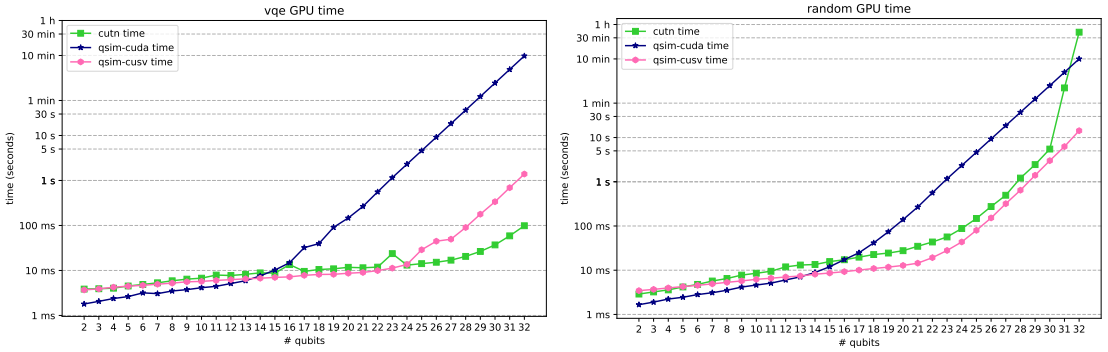


Figure 2.6: Time to solution for the VQE (left) and Random (right) circuits.

operators, or as a $(2, 2, 2, 2)$ order-4 tensor, in the case of a controlled gate, adding two dimensions of size 2 for each additional input of the operator. As such, the size of the tensor network representation scales linearly with the number of gates in the quantum circuit and the number of qubits in the system.

Figure 2.6 provides a side by side comparison of the execution times of the VQE and the Random circuits. The time performance data has been collected, for each circuit configuration, as the average time over 10 runs after having performed 3 warm-up runs. The tensor network time is the sum of the pathfinding time, done in CPU, and the contraction

Pofiler results for the QFT circuit						
qubits	qsim-cuda			qsim-cusv		
	memory	kernels	memory (KiB)	memory	kernels	memory (KiB)
13	1.6 %	98.4 %	88.16	38.7 %	61.3 %	26839.04
14	1.2 %	98.8 %	152.28	39.1 %	60.9 %	26900.48
15	33.2 %	66.8 %	280.53	29.3 %	70.7 %	27033.6
16	33.9 %	66.1 %	537.03	30.1 %	69.9 %	27289.6
21	8.5 %	91.5 %	16435.2	14.3 %	85.7 %	43212.8
22	3.9 %	96.1 %	32952.32	8.0 %	92.0 %	59607.04
23	2.1 %	97.9 %	65873.92	4.0 %	96.0 %	92436.48
24	1.1 %	98.9 %	131717.12	2.1 %	97.9 %	158105.6
25	0.6 %	99.4 %	263413.76	1.1 %	98.9 %	289433.6

Table 2.2: Nsight profile data for the QFT circuit.

time, done in GPU. More than 3244 quantum circuit simulations have been collectively performed for this experiment. The *Nsight* profile data gathered from the QFT benchmark on the *qsim-cuda* and the *qsim-cusv* simulators are summed up in Table 2.2. It is noticeable how the *qsim-cuda* simulator has the overall fastest performance for circuits with size of 14 qubits or less, after which the performance of the simulator degrades significantly.

The *qsim-cuda* simulator moves the whole state vector in memory through cache at once using synchronous `memcpy` calls, while the *qsim-cusv* simulator instead splits the state vector into multiple sub-state vector, invoking multiple asynchronous `memcpy` operations and applying the gate separately on each sub-state vector. This leads to overall better performance for the *qsim-cuda* simulator in quantum circuits that use less than 14 qubits, as a single synchronous `memcpy` operation is required, whilst *qsim-cusv* suffers the high number of asynchronous `memcpy` operations, which outpower the computational intensity of applying one quantum gate. The shared memory size of the NVIDIA A100, the device used for all the simulations, is only configurable up to 164 KB: the state vector of a 14 qubits system, at *complex-64* single precision, occupies 128 KB. As soon as the state vector size exceeds that of the cache, the *qsim-cuda* simulator is forced to split the state vector into sub-state vectors (like in the qsim-cusv case), but keeps doing so with synchronous `memcpy` operations. On the other hand, the qsim-cusv simulator manages to hide more of the computational complexity of applying quantum gates by overlapping asynchronous `memcpy` operations. With problem sizes larger than 22 qubits, however, the computational intensity of applying a quantum gate starts to overshadow the advantage of asynchronous Input/Output (I/O), with a degradation of performance. This performance difference between *qsim-cuda* and *qsim-cusv* is largely

due to the memory access patterns of those two simulators. The *qsim-cusv* simulator can make use of index bit swapping, which gives it an advantage in cache locality with respect to the *qsim-cuda* simulator [36].

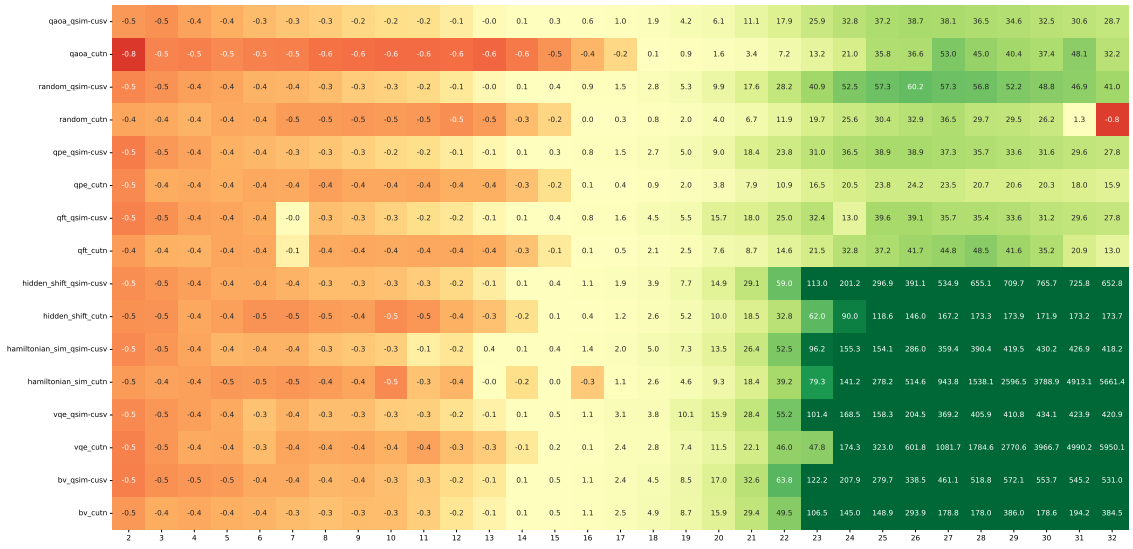


Figure 2.7: Speedup heatmap of qsim-cusv and cutn with respect to qsim-cuda.

In the case of the Hamiltonian simulation, VQE and QAOA circuits, the tensor network outperforms both state vector simulators. This is a consequence of the fact that these circuits are mainly composed of single-qubit gates, corresponding to an *entanglement ratio* score lower than 0.5, and have a well distributed amount of entanglement across the system, meaning an *entanglement ratio* score lower than 0.2. This leads to having small intermediate tensors during the contraction step, that eventually get merged together into larger tensors right towards the end of the process. As soon as one of those two metrics grows, the employed pathfinding algorithm [45] struggles find an optimal contraction path. Moreover, once all the order-2 tensors have been contracted, the resulting topology resembles that of a MPS, which is optimal for contraction. This is important information, as the contractions of order-2 tensors are easy to perform and do not increase the intermediate tensor order during the contraction process.

Observation II.III

Unstructured and unbalanced tensor networks give rise to large intermediate tensors during contraction in function of their qubit size and *program communication* metric, hindering performance.

The circuit with the worst performance for the *cutn* simulator is the Random circuit, which is the only problem where performance degrades by more than an order of magnitude,

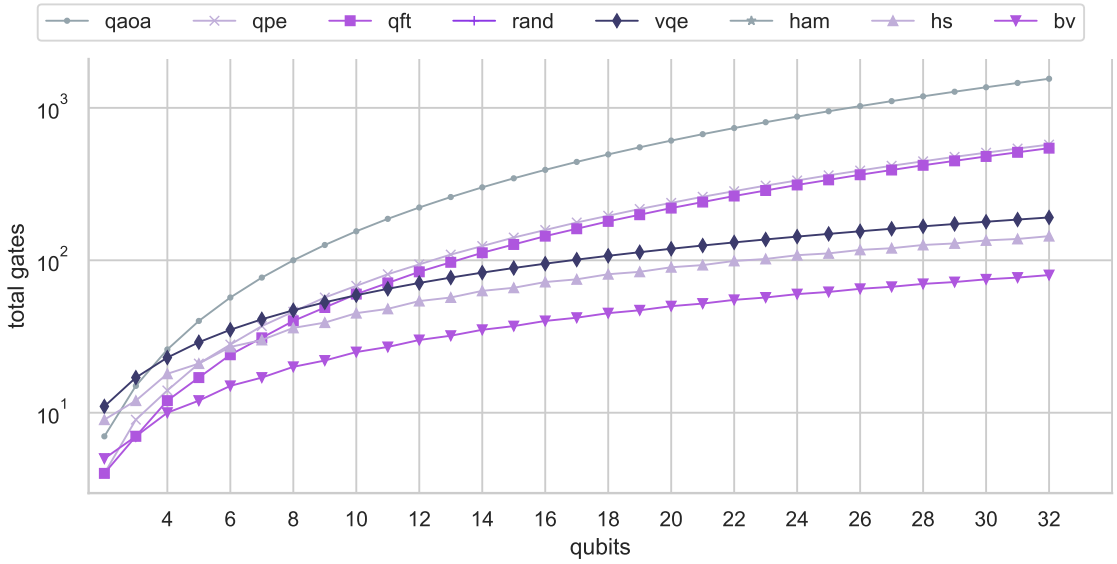


Figure 2.8: Gate usage scaling of the benchmarked circuits by input size. The Y-axis is in logarithmic scale.

mainly due to its lack of an internal structure, something which can be clearly noticed on the right of Figure 2.6. In all other benchmark circuits, the performance in terms of time is comparable to that of a state vector simulator, whilst keeping the reduced memory footprint of the tensor network representation.

In Figure 2.7, once can see the relative speedup of the *qsim-cusv* and the *cutn* simulators when compared to the *qsim-cuda* simulator, whose baseline performance is mostly dependent on the system size, rather than on the number of gates to be processed. Negative speedups can be observed in the left half of the heatmap, as the *qsim-cuda* backend outperforms the two other simulators. On the second half of the heatmap, an appreciable speedup on both backends can be observed. In the QAOA, Random, QPE and QFT circuits the speedups range to up to $60\times$ for the *cusv* backend and up to $53\times$ for the *cutn* backend. The largest speedups are measured on the Hidden Shift, Hamiltonian Simulation, VQE and Bernstein-Vazirani circuits, exceeding by $5000\times$ the speedup value for Hamiltonian Simulation and the VQE. These last four circuits are the ones that scale more slowly in terms of the number of quantum gates, as previously seen in Figure 2.8. The circuits have been generated using $P = 1$, $k = 0.5$, $M = \lfloor kN \rfloor$, $L = 1$, and $T = 1$. The main outlier is the Random circuit, which has poor performance on the *cutn* backend, with a negative speedup at high qubit sizes.

Due to limitations in the software environment, given the at-the-time still prototypical nature of the simulation library, I was unable to extend the analysis to distributed state

vector simulation, and as such to test larger quantum circuits with this approach. However, the time to solution trend is evident for both state vector simulators. As previously discussed in this chapter, and as highlighted in Figure 2.5, the state vector method suffers from diminishing returns as the problem size grows. This follows the results of large scale distributed simulations in the literature, which advertise the same single-GPU performance trends [61].

2.4.3 Distributed sliced tensor contraction performance

In order to understand how the performance of tensor network contraction scales the number of compute units (GPUs) increases, a strong scaling experiment has been designed. I have implemented a distributed version of the tensor network contraction algorithm, by leveraging the cuTensorNet library, Message Passing Interface (MPI) and NVIDIA Collective Communications Library (NCCL). This let me run scaling experiments for all the circuits in the benchmark at size 32 qubits, exception made for the Random circuit which was limited at size 28 qubits. The objective was to test the efficacy of tensor network slicing in improving contraction efficacy by using an increasing number of GPUs and compute nodes on the Leonardo supercomputer. The algorithm starts by spawning one MPI process for each available GPU, and first performs a distributed pathfinding on the whole network. The best path, selected according to the lowest Floating Point Operations per Second (FLOPS) count, is broadcast to all other MPI processes through the MPI communicator. The tensor network is then sliced in a number of subnetworks equal to the number of MPI processes, in order to provide to each MPI process, and thus each GPU, a comparable amount of FLOPS to be performed. Each GPU contracts its own subnetwork, and all the partial results are reduced with a sum operation through the NCCL communicator, that yields the final amplitude result.

In Figure 2.9, I present the strong scaling of distributed sliced tensor network contraction performance on circuits of size 32 qubits, with the exclusion of the Random circuit at size 28 qubits, from using 1 GPU going up to 256 GPUs. The number of GPUs (*# GPUs*) also corresponds the number of MPI processes, and each point represents the mean of 30 datapoints. The node configuration on Leonardo includes 4 GPUs per node, with each node being interconnected with an Nvidia Mellanox network, reaching up to 200 Gbit/s node to node transfer rates. Given the size of the partial results being reduced, network bandwidth does not act as a computational bottleneck, but rather the arithmetic intensity of the contraction of the *worst* subnetwork. The strong scaling results show that not all the quantum circuits selected in the benchmark exhibit large performance gains, particularly

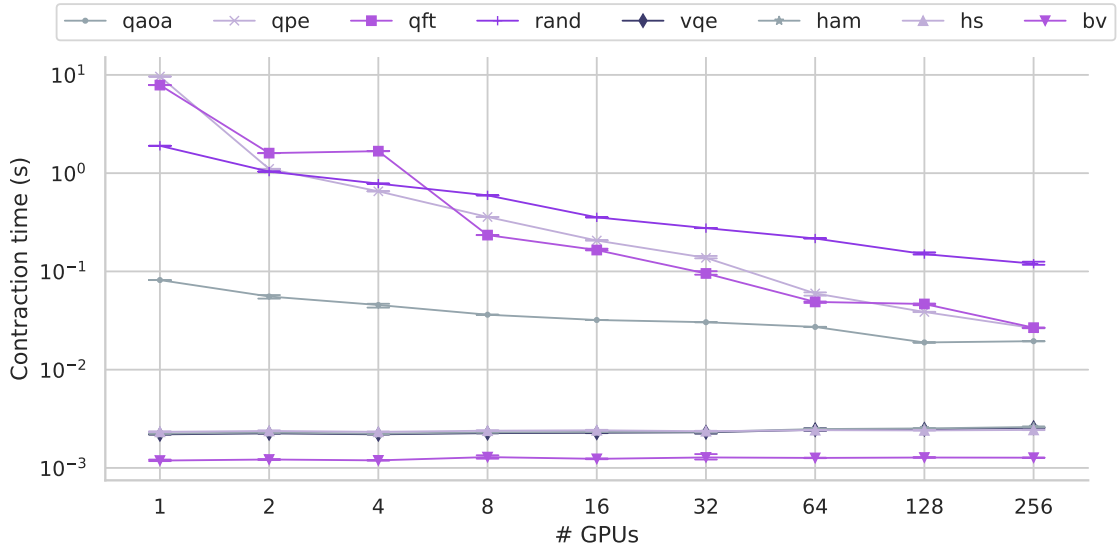


Figure 2.9: Strong scaling of tensor contraction at 32 qubits.

the Hidden Shift, VQE, Bernstein-Vazirani and Hamiltonian simulation. These circuits lack complexity in the structure of their circuit-derived tensor networks, as reported by their asymptotically decreasing program communication metric in relation to the size of the quantum circuit: this makes it so that the overall arithmetic intensity of each subnetwork fails to saturate the GPU's computational capacity. The QAOA sees a noticeable improvement in terms of contraction performance, steadily lowering the contraction time from 81 ms on a single GPU to as low as 19 ms on 16 GPUs, a speedup of more than $4.2\times$. Likewise, the Random circuit's contraction time goes from a mean of 1.89 s when using a single GPU to about 120 ms in the case with 256 GPUs, a speedup of about $15\times$. The largest improvements in the contraction times can be obtained on the QFT and QPE quantum circuits, which drop more than one order of magnitude in contraction time when going from running on one GPU to 16 GPUs. This translates in a speedup of more than $294\times$ on the QFT circuit and a speedup of more than $364\times$ on the QPE circuit, respectively, in lieu of just increasing by $256\times$ the computational resources available. The reason behind this is that those quantum circuits feature high *program communication* and *entanglement ratio* scores, which means that most of the contraction operations will involve order-4 tensors, which quickly ramps up the size of the intermediate tensor during contraction. The higher the number of sub-tensor networks being contracted, the smaller the overall order of the final sub-tensor is going to be. Scaling the number of GPUs thus prevents the formation of these large sub-tensors, which get contracted only once in the final reduce step, lowering the overall contraction time.

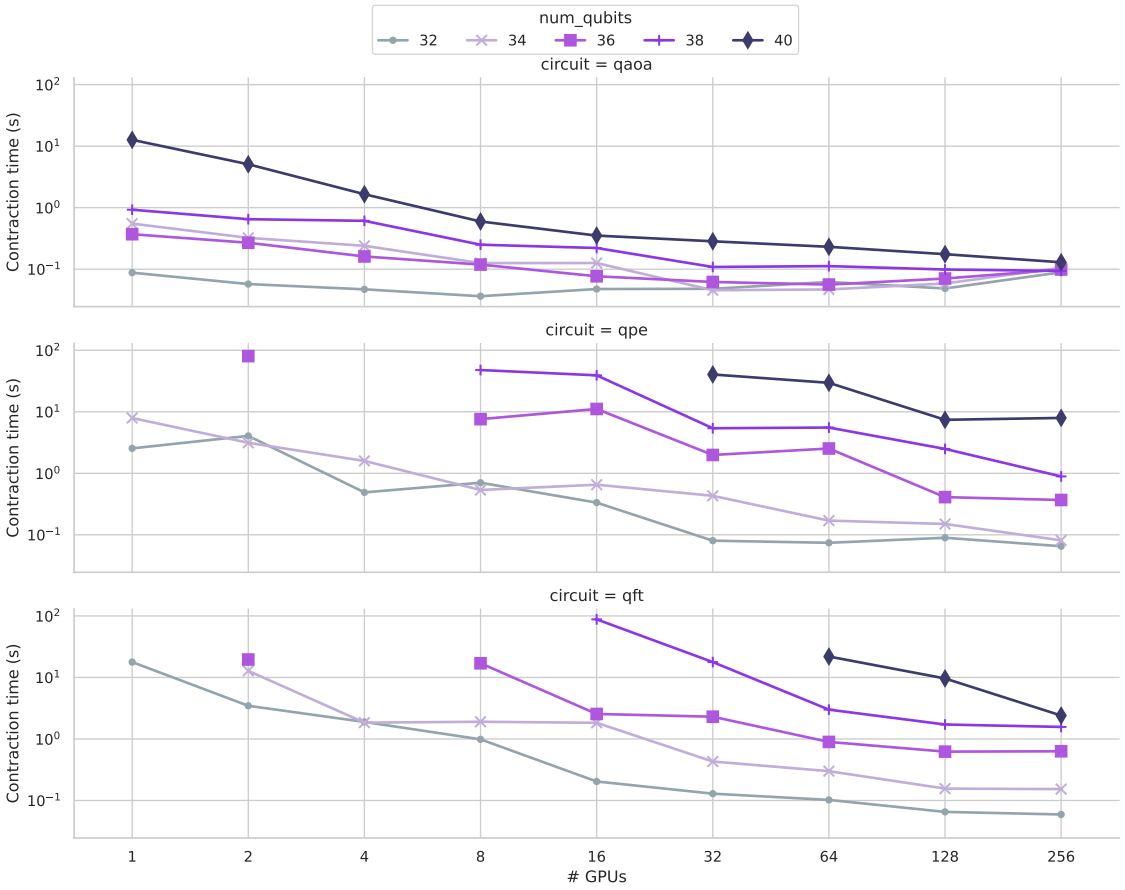


Figure 2.10: Strong scaling of distributed sliced tensor network contraction.

Observation II.IV

Quantum circuits with large program communication and entanglement ratio scores benefit the most from sliced distributed contraction, reaching super-linear speedups with respect to a linear increase in computational power.

Albeit some of the benchmark circuits considered have been found to be trivially solvable even by a single GPU, the presented results show how specific descriptive metrics of a quantum circuit, namely the program communication and entanglement ratio metrics, can let one foresee the presence or absence of a computational gain with a distributed sliced tensor network contraction.

On the topic of evaluating distributed sliced performance, it is still unclear how to measure weak scaling performance of different quantum circuits, that is measuring the performance of a problem when scaling equally the problem's complexity and the available computing resources. The problems considered in this benchmark are parameterised by the number of qubits, which does not provide direct control over the problem's contraction

complexity, which mainly depends on the treewidth of the tensor network [45]. Nonetheless, Figure 2.10 reports the strong scaling of distributed sliced tensor network contraction performance on the QAOA, QPE and QFT circuits of sizes ranging from 32 to 40 qubits, scaling from one to 256 GPUs, where the number of GPUs (*# GPUs*) also corresponds the number of MPI processes. A contraction time limit of 120 seconds has been imposed on the pathfinder. In the leftmost subplot, the QAOA circuit, which is the least computationally intense of the three, is easily handled by a single GPU up to 40 qubits. Most importantly, a communication bottleneck is observed at 256 GPUs, since the intensity of contracting the sub-tensor networks is smaller than the time required to reduce all of them to a single scalar. In the central and rightmost plots, the QPE and QFT circuits, boasting higher computational intensity, start to exceed the imposed time limit at 36 and 34 qubits, respectively. Notably, both circuits take better advantage of the large scale, solving 40 qubit circuits when using 32 and 64 GPUs, reaching a communication lower bound at 128 GPUs. Based on these results, I argue that it could be possible to develop a synthetic parameterisable quantum circuit that grows in terms of the circuit derived tensor network's treewidth to actively measure the weak scaling performance of tensor network contractions. This would most probably end up being a variation of a Random circuit, which however holds no meaning in terms of problem solution.

Observation II.V

The complexity of contracting quantum circuit derived tensor networks does not scale with the problem size, i.e. the number of qubits. As such, specialised synthetic benchmarks are needed to measure the weak scaling of tensor network contraction.

For the sake of this thesis, following the results hereby presented, I can foresee a correlation with the strong scaling capacity of the cuTensorNet library in relation to the program communication metric of a quantum circuit. For quantum circuits with program communication scores of one can efficiently leverage large multi-GPU acceleration. On the opposite case, when the program communication approaches zero, one GPU can suffice, and no advantage is gained from increasing computational resources.

2.4.4 Pathfinding impact on tensor contraction performance

There is a need to classify quantum circuit derived tensor networks according to their pathfinding complexity. Specifically, was interested in predicting which circuits can be contracted with higher efficiency in correlation to an increase in the resources available to the pathfinding algorithm. This has been done by investigating how variations in the resources

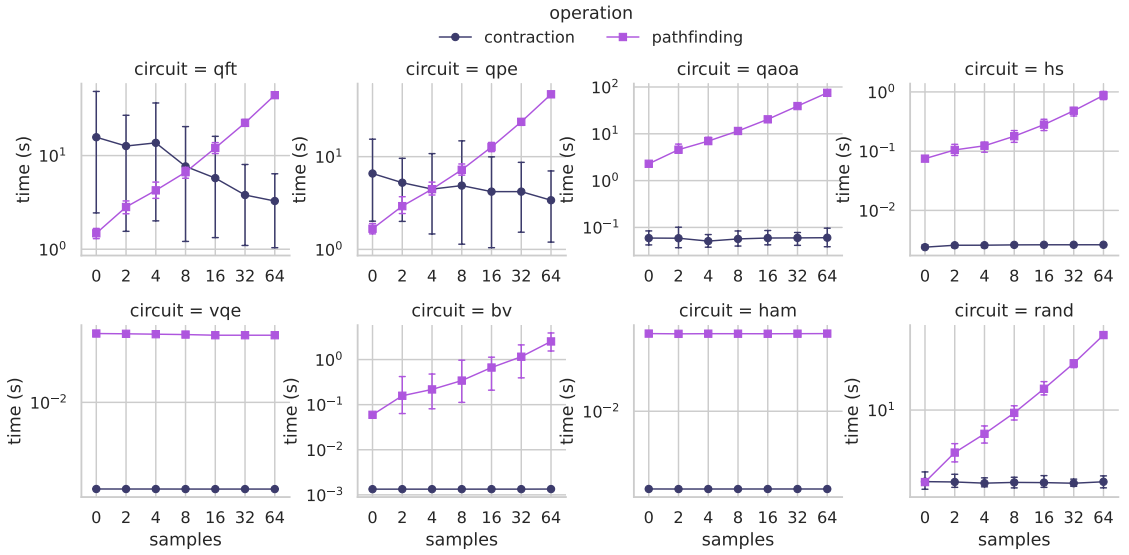


Figure 2.11: Pathfinding and contraction time correlation.

available to the cuTensorNet pathfinding algorithm, namely the number of additional samples performed, impact the contraction time of the quantum circuit derived tensor networks in the benchmark. In this context, a sample size of n indicates that $n+1$ pathfinding calls have been made. The metric object of evaluation is the total time required by the pathfinding algorithm to sequentially compute all the samples by enforcing a single thread. Although this search may be easily distributed, the total computation time required by the pathfinder is still the same. The size of each quantum circuit is kept fixed at 32 qubits, apart from the random circuit of size 28 qubits, due to representability reasons.

In Figure 2.11, the variation in the pathfinding and contraction time with respect to an increase in the number of samples is presented, with the pathfinding time in orange and contraction time in blue, for tensor networks derived from circuits of size 32 qubits, where vertical bars represent the 90th percentile interval. The number of samples performed by the pathfinder are executed sequentially in order to extrapolate the total time required for this computation, whilst contractions are performed using a single GPU, as such different y-axis limits are used in each separate subplot. These results let me identify three categorisations for the quantum circuits used in this benchmark: pathfinding-bound problems, contraction-bound problems and unbounded problems. Pathfinding-bound problems include the VQE and Hamiltonian simulation circuits, since the complexity landscape of their possible contraction paths is mostly flat. This means that such circuits provide no advantage in terms of contraction time when assigning additional resources to the pathfinding algorithm. Contraction-bound problems include the QAOA, Hidden Shift, Bernstein-Vazirani and

Random circuits, and they are characterised by having a non-trivial pathfinding complexity landscape, but trivial contraction complexity. This means that providing additional resources to the pathfinder indeed gives rise to better solutions in terms of total FLOPS in contraction, but the overall difference in terms of contraction efficiency is unnoticeable. Unbounded problems include The QFT and QPE problems, since they show an direct correlation between the amount of resources provided to the pathfinding algorithm and a reduction in the contraction time. In fact, as the number of samples increases, a speedup can be observed, both in terms of contraction time of $1.9\times$ on the QPE and of $4.79\times$ on the QFT by increasing the pathfinding time by about $29\times$. From these observations it follows that quantum circuits characterised by an *entanglement ratio* metric that converges to *one* map to more complex pathfinding problems that can efficiently leverage additional pathfinding resources, whilst if the same metric saturates to values lower than 0.4, the problem is bounded, either in pathfinding or contraction.

Observation II.VI

Quantum circuit derived tensor networks can be classified in pathfinding-bound problems, contract-bound problems, and unbounded problems. Only the latter show an anticorrelation between pathfinding time and contraction time.

Although this result might point towards diminishing returns, it must be noted that the pathfinding time was purposefully measured sequentially on a single thread, but each sample is indeed independent of the other, as such using one independent thread and core per sample is going to keep the real-time pathfinding time constant. Moreover, in terms of real-world tensor contraction, contractions are performed with a batched approach, where the contraction path is computed only once, but the actual contraction may be performed hundreds or thousands of times, thus outweighing the steep initial pathfinding cost. Given the intense computational cost of pathfinding and its reliance on performing numerous samples, an open question remains on whether this algorithm can be further accelerated using GPUs in order to increase the number of concurrent samples while still fitting a low time bound.

2.4.5 Lessons learnt

The simulation performance is ruled by a plethora of factors. State vector simulators are inherently limited by the problem size, given the exponential memory requirement, thus trying to simulate systems larger than ~ 50 qubits with state vectors suffers from diminishing returns [43]. Moreover, the distribution of the subvector-matrix multiplications

scales exponentially over the problem size, hindering the time performance. The main advantage lies in being able to access the complete set of information encoded in the wave function. Modern state vector simulators can take advantage of GPU caching in order to limit the computation time in small scale problems, but still hit a hard memory bandwidth limit as soon as the cache's capacity is saturated. Tensor Network contraction has a lot of potential for problems which embody a structure with well-balanced connectivity, as that of a structured mesh, as this generates small sized intermediate tensors during the contraction process. As such, current pathfinding algorithms are optimised for finding such structures, and struggle whenever the topology becomes more irregular, such as when most of the entanglement operations in a circuit are concentrated on a few qubits. This is because the size of the intermediate tensors immediately spikes up, slowing the dot product with the remainder of tensors. The main advantage of this approach is the fact that it scales linearly in memory, thus allowing the simulation of larger systems, so long as the circuit's structure is favourable for contraction. In fact, by scaling the number of qubits in order to saturate the memory capacity of an 80GB NVIDIA A100 GPU, assuming that connectivity information is stored in the shared memory of the machine, one would need more than $7.5k$ logical qubits for a QAOA circuit with $P = 1$, or more than $40k$ qubits for the Random circuit, the two circuits in the benchmark with the highest gate scaling per qubit. Despite the fact that there is no guarantee of convergence towards an optimal path, tensor network contraction is the only approach to exact simulation of quantum circuits that can scale favourably to circuits with dimensions larger than 50 qubits. Previous works show that parallelising the contraction process is trivial, thus the main bottleneck remains the pathfinding algorithm, a well-known NP-hard problem [176].

Following the observations in this chapter, circuits which are characterised by an *entanglement ratio* score greater than 0.2 have a highly unbalanced structure, reducing the efficacy of the pathfinding algorithm for tensor network contraction. Likewise, if more than half of the circuit's gates are double-qubit gates, which amounts to a *entanglement ratio* score greater than 0.5, the tensor network approach starts to scale poorly, as stated in Observation 4. This is due to the presence of large tensors early on during the contraction, slowing down the overall process. In both of the previous cases is thus suggested to use a state vector solver. However, if these first two conditions are not met, one must check for the *program communication* and the *critical depth* scores. If they are opposite to one another, with one being smaller than 0.2 and the other being larger than 0.9, then the best simulation method is the tensor network contraction. This is due to the fact that, if the previous conditions about the *entanglement ratio* and the *entanglement ratio* scores are met, a *program communication* score greater than 0.9 and a *critical depth* score lower than 0.2

indicate a circuit structure in which most qubits interact with each other, but there are little to no repeated interactions. On the other hand, a *program communication* smaller than 0.15 together with a *critical depth* score larger than 0.9 indicate a circuit structure composed of many chained two-qubit interactions amongst the same pairs of qubits. In both cases the tensor network becomes a pseudo-regular grid, which can be efficiently contracted whilst keeping intermediate tensor sizes at bay, leading to time performance gains of up to one order of magnitude.

Program communication is especially important in determining whether a quantum circuit contraction can be efficiently contracted in a distributed sliced setting. Those circuits display super-linear speedups with respect to the available compute resources. Moreover, I showed how this is only justifiable to provide additional pathfinding resources to unbounded quantum circuit tensor contractions, as they can provide further speedups, whilst all other circuits can save on using additional resources on pathfinding.

2.5 Chapter summary

This chapter characterised the performance of a selected suite of relevant quantum circuits when simulated on state-of-the-art simulator backends and high performance hardware. At first, the circuits have been characterised according to objective metrics that could describe their topological structure, to later correlate them to the performance of the simulators. For what concerns **RQ1**, the results point towards the fact that state vector simulators become heavily communication bound as soon as the size of the state vector exceeds that of the GPU's cache. The overall performance of tensor network contractions can already outpace the state vector simulator in some of the benchmark problems, whilst keeping a comparable, albeit slower performance in the remainder of the test runs. The answer to **RQ2** mostly underlined that tensor network contractions perform better in circuits that have well distributed entanglement amongst the qubits, with a low overall number of two qubit gates in relation to the total number of gates. In the opposite case, state vector methods should be adopted for those hard to tackle problems instead. An important observation relative to **RQ3** is that tensor network contractions proved to be highly dependent on the efficacy of the pathfinding algorithm, while state vectors methods are mostly limited by qubit count.

It is reasonable to assume that by improving the memory access pattern of a state vector backend, one may improve the time performance of any benchmark to be simulated. Similarly, the development of further tensor contraction pathfinding optimisations for harder

to tackle topologies may push the performance of such backend to become the fastest simulation approach for quantum circuits, for example by limiting the growth of intermediate tensors during contraction. This would let one leverage the fact that the representation of tensor networks in memory enables the validation of larger quantum circuits and computers. Moreover, the promising scaling of distributed pathfinding and sliced contraction approaches over large tensor networks have the chance to further close the gap on simulating real quantum computer. It must be noted that there is an absence of a real-world quantum algorithm class with parameterisable contraction complexity. The development of such a class of algorithms would provide means for measuring weak scaling performance of distributed tensor network contraction libraries, easing performance comparisons between quantum and classical systems. Future works could investigate the applicability of GPU accelerated pathfinding algorithms, so as to further improve the path quality and reduce the overall contraction time in unbounded problems.

Chapter 3

Radiation Faults and Quantum Error Correction

Reliability is fundamental for developing large-scale quantum computers. Since the benefit of technological advancements to the qubit's stability is saturating, algorithmic solutions, such as QEC codes, are needed to bridge the gap to reliable computation. Unfortunately, the deployment of the first quantum computers has identified faults induced by natural radiation as an additional threat to qubits reliability. The high sensitivity of qubits to radiation hinders the large-scale adoption of quantum computers, since the persistence and area-of-effect of the fault can potentially undermine the efficacy of the most advanced QEC.

3.1 Objectives

In this chapter, I investigate the resilience of various implementations of state-of-the-art QEC codes to modelled radiation-induced faults through simulation. Given the advances in software QEC and the available knowledge about radiation corruption in quantum chips, this chapter seeks to answer the following research questions:

- **RQ1**: Are state-of-the-art error correction codes, designed for intrinsic noise, effective for radiation-induced events?

This chapter refers to the contents of the article "On the Efficacy of Surface Codes in Compensating for Radiation Events in Superconducting Devices", written by M. Vallero et al., published in the Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, and winner of the Best Student Paper Award [177].

- **RQ2**: How can one tune and configure the surface code to improve the chance of correcting also radiation strikes?

- **RQ3**: Which are the main insights to design future reliability solutions for radiation-induced corruptions?

The goal is to understand the efficacy of state-of-the-art surface codes against radiation-induced events, to provide indications on how to configure the code to increase the chances of correcting the radiation strike, and to give insights on how to design future reliability solutions. To provide a realistic evaluation, I modelled the radiation-induced fault following its theoretical definition and experimental observation, considering both the fault's temporal and spatial distributions. The model has been translated into a flexible and easy-to-use quantum fault injection toolkit, which has already been disclosed as open-source library.

The hereby presented extensive analysis is based on over *400 million* faults injections and considers both the repetition and XXZZ surface codes implemented in various fashions. I detail how physical level faults spread up to the higher-abstraction logical layer and correlate the code performance with the surface code distance, the number of physical qubits affected, and the architectural interconnection pattern of qubits. Overall, more than 12 different configurations have been considered.

I show how surface code performance degrades significantly in the presence of radiation-induced transient faults, reaching logical error rates of up to 54% and that a single particle interaction that spreads to neighbour qubits has an effect on the code output which is worse than corrupting half of the available qubits. The hereby presented analysis highlights that, with a constant number of physical qubits, the bit-flip repetition code is up to 10% more effective against radiation than XXZZ. Moreover, by properly selecting the underlying hardware topology, one can improve the radiation fault correction by up to 7% in the repetition code and 9% in the XXZZ code.

While several works have evaluated radiation's effect in quantum devices [21, 23, 24, 91, 92, 125, 127–131] and proposed some physical implementation improvements [21, 146, 147, 150–152], this is the first investigation targeting the effectiveness of surface codes in correcting radiation-induced events. Some works have considered the impact of an *ideal erasure error* due to fabrication errors on QEC, however there is still much to be questioned about the applicability of these models to the specific issue of radiation [178, 179].

Specifically, I aim at broadening the understanding of surface code performance by:

- Describing a methodology for modelling and injecting particle-induced transient faults in transmon quantum computers.

- Analysing the impact of radiation-induced faults over both the temporal and spatial domain;
- Measuring the conversion rate of a physical level transient fault into the logical level error of a surface code;
- Detailing how and why surface codes fail, providing guidelines on how to configure the available codes and fostering the development of new and improved error correction techniques;
- Providing an easy-to-use tool to test future surface codes implementations with realistic fault models;

Since the methodology and framework hereby provided work at software abstraction layer, it can be directly extended and adapted to any other quantum computer implementation.

The chapter refers to the background concepts on quantum circuits, intrinsic noise and radiation events, which have been previously introduced in Chapter 1, Sections 1.1.2, 1.1.5 and 1.1.7. The remainder of the chapter is organised as follows. Section 3.2 describes the implementation of both the intrinsic noise and radiation-induced fault models. In Section 3.3, I define the surface code classes that have been tested, to later present an analysis of the collected data in Section 3.4. The chapter is concluded by summarising the main results and digressing on the future investigation paths in Section 3.5.

3.2 Noise and Fault Model Formalisation

This Section describes how the previously introduced intrinsic and radiation noise have been modelled to perform efficient and accurate simulations of the quantum computer's behaviour.

3.2.1 Intrinsic noise model

A superconducting device's ability to retain information varies over time, and depends on the accuracy associated with performing each quantum gate operation. As such, noise models are necessary in order to accurately simulate the behaviour of real quantum computers.

Following the common practices found in the literature, I have decided to use a depolarisation error model based on *unitary* Pauli operators, adapted from [85]. The uncorrelated nature of this noise model's faults follows the definitions of intrinsic noise provided in the

literature [19], and surface codes are built and optimised against this kind of depolarisation noise. The model I have used is parameterised over a *physical error rate* p, and acts by appending an X, Y or Z operator after each gate operation O with probability $\frac{p}{3}$, thus producing uncorrelated errors in time and space:

$$O\left|\psi\right\rangle \rightarrow \mathcal{E}O\left|\psi\right\rangle \text{ with } \mathcal{E} \doteq \sqrt{1-p}\,\mathbb{I} + \sqrt{p/3}\,(X+Y+Z)\,. \tag{3.1}$$

When performing two-qubit gate operations, an error gate is appended after each one of them. This error gate is defined as the tensor product of two independent $\mathcal{E}$ noise operators: $\mathcal{E}_2 \doteq \mathcal{E} \otimes \mathcal{E}$.

Such an error model is frequently used in literature to benchmark the performance of surface codes, and is inherently defined as one of the main examples of a *nice* error basis. Given that surface codes have relatively low circuit depth, including those analysed in this chapter, their execution time is orders of magnitude lower than the characteristic T1 and T2 times of modern superconducting quantum computers [77, 180]. As such, the state coherence difference between the first and last gate operation in the circuit can be approximated as being constant without incurring a loss in simulation fidelity.

3.2.2 Radiation-Induced fault model

As previously stated, the impact of radiation breaks the quantum equilibrium, causing a loss of coherence. To model the impact of radiation in the logic state of a qubit, a *non-unitary* reset operation is appended to each quantum gate acting on that qubit with probability p_{q_i}, with i being the qubit's index. The energy deposited by the impinging particle, and thus the probability of applying the reset operation on a qubit, depend on the distance from the point of impact and decays over time. As such, the fault event hereby modelled evolves both in the spatial and temporal domains from the *locus of radiation*, i.e. the qubit from which the fault spreads.

In the **time domain**, since the deposited charge in silicon recombines and diffuses [133], the fault event evolves as a decaying exponential [132, 133, 137] that spikes at the locus of radiation and wears off to zero as time goes on. Equation 3.2 details the temporal decay function $T(t)$ over continuous time $t \in [0, 1]$, that outputs the probability of generating quasiparticles in the silicon substrate:

$$T(t) = e^{-\gamma t}, \qquad \gamma = 10, \qquad t \in [0, 1]\,. \tag{3.2}$$

The factor $\gamma = 10$ defines the exponentially decaying presence of quasiparticles in the silicon substrate, following the experimental rates highlighted in the literature [24, 126, 131].

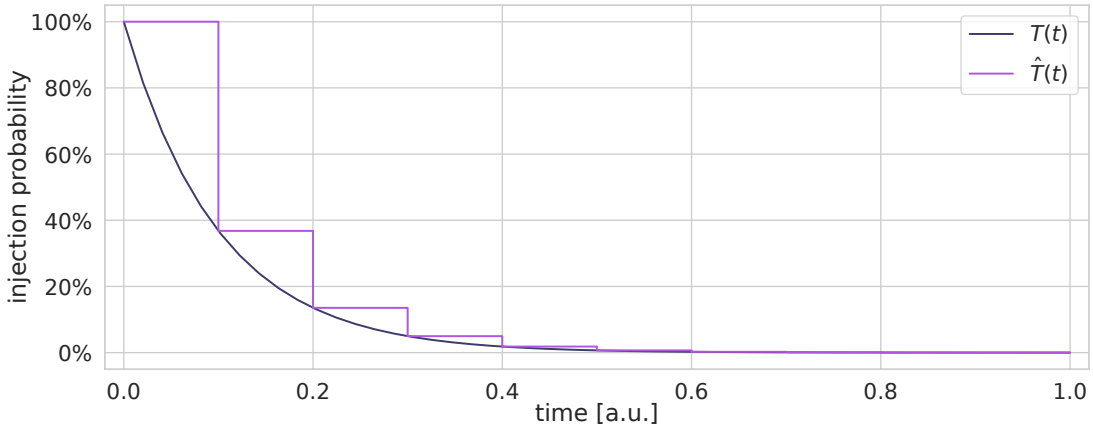


Figure 3.1: Intensity of radiation-induced faults over time.

The estimated time required to execute a shot of the surface code on a real quantum computer ranges, on average, from $\sim 14\ \mu s$ to $\sim 125\ \mu s$ [21, 181, 182]. Transient radiation-induced faults can last for upwards of 100 seconds [126, 134]. Given this order of magnitude difference, one can approximate the time evolution of $T(t)$ with a step function $\hat{T}(t)$, by sampling the temporal decay $T(t)$ over n_s number of samples, as shown in Figure 3.1. For the sake of this chapter's analysis, I have selected $n_s = 10$, meaning that the function $T(t)$ has been sampled over 10 equidistant points in time, granting a reasonable trade-off between accuracy and performance. Increasing the number of samples comes at the expense of computational overhead.

In the **spatial domain**, the deposited charge spreads from the locus of radiation throughout the quantum chip, diminishing in intensity the further a qubit lies from the impact point [23]. By following the qubit interconnections in the quantum computer's architecture, an undirected graph with fixed weight on each edge of $n = 1$ can be devised, called coupling map. This behaviour approximates the electron hole pair distributions induced by particle impacts in silicon over a normalised integer distance d [23]. The spatial damping function S, parameterised by the minimum distance between two qubits in the coupling map, is defined as

$$S(d) = \frac{n^2}{(d+n)^2}, \quad n = 1\ . \tag{3.3}$$

To parameterise the application of faults on a qubit, the root injection probability sampled from T is thus multiplied by the output of S. The product of the temporal and spatial domain fault evolution functions is collectively defined as F, the transient error decay function

$$F(t,d) = T(t)S(d)\ . \tag{3.4}$$

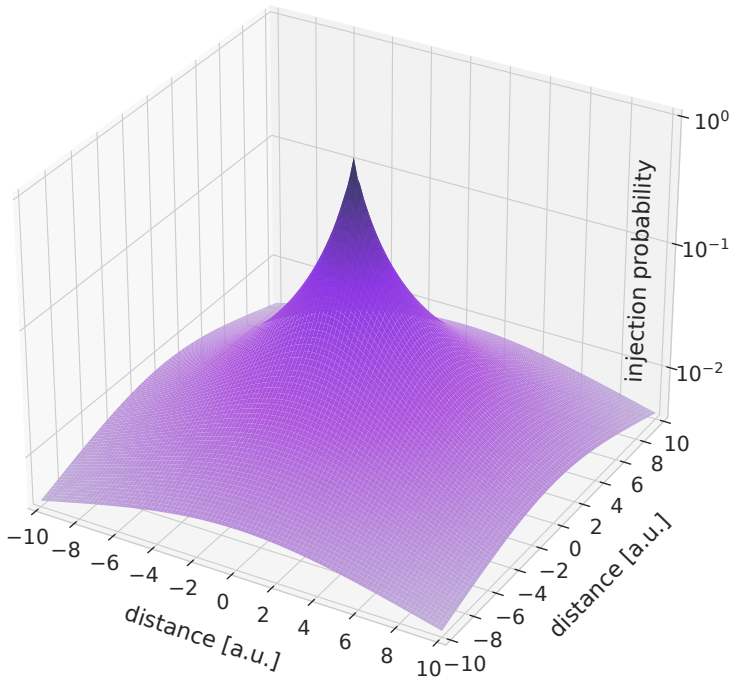


Figure 3.2: Intensity of radiation-induced fault with respect to the distance.

The fault probability p_{q_i}, obtained from sampling $F(t, d)$, is computed on a per-qubit basis, and it is used to parameterise the application of a reset operator after each gate applied to that qubit. An example of the spatial evolution of the root injection probability at time $t = 0$ is presented in Figure 3.2. The spatial decay function $S(d)$ parameterised by the distance from the locus of radiation at coordinate $(0, 0)$, with a peak of 100%.

3.3 Exploration of Design Space

For this analysis I have considered two of the most widely employed surface codes: the repetition [21, 183] and the XXZZ surface code [181, 182, 184]. The library used to generate the parameterised correction codes and decoding their output with the MWPM algorithm is courtesy of the Qiskit Topological Codes project [185]. While presenting an extensive evaluation on two cornerstone codes, the hereby described methodology and the analyses are not tied to a specific code or implementation, and can be adapted to any other current or future QEC code.

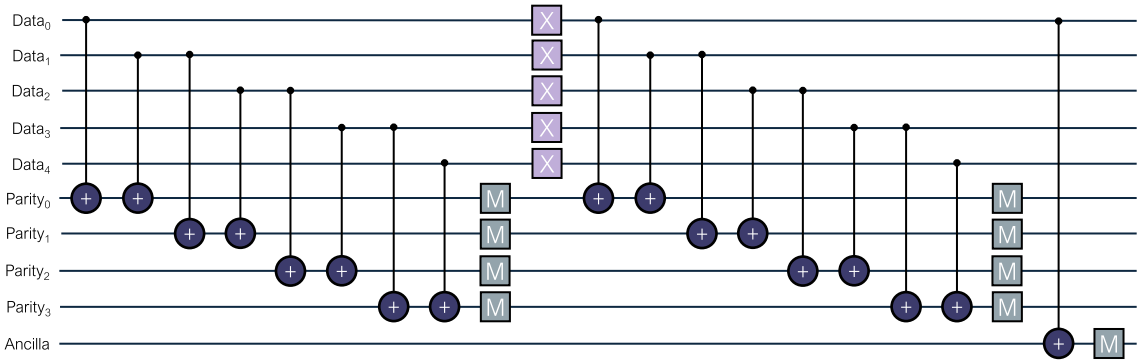


Figure 3.3: Distance-(5,1) bit-flip repetition code.

3.3.1 Repetition code

The quantum repetition code employs *n* data qubits to encode the state of one single logical qubit, by repeating information in what is called a *Greenberger-Horne-Zeilinger (GHZ) state*, and additional *n* qubits to perform stabiliser measurements, necessary to extract error syndromes. The overall distance of the code is defined as $d = (d_Z, d_X)$, with either $d_Z = 1$ for phase-flip protection codes and $d_X = 1$ for bit-flip protection codes.

$$|0\rangle \rightarrow \left|\psi_0\right\rangle^{\otimes n}, \quad |1\rangle \rightarrow \left|\psi_1\right\rangle^{\otimes n} \tag{3.5}$$

This code can either offer protection from bit-flips or phase-flips, according to the basis chosen for the GHZ state: by using the Z-basis, one encodes bit-flip protection, whilst by choosing the X-basis one encodes phase-flip protection. The repetition code can detect only the corresponding encoding basis error on up to $\lfloor (n-1)/2 \rfloor$ qubits, so long as those error events are uncorrelated (whilst radiation events are correlated). The total number of qubits required to encode a repetition code is $q_{rep} = 2n$, with $n = \max(d_Z, d_X)$ being odd, and either d_Z or d_X being equal to 1. This code is one of the few that have been extensively tested on superconducting quantum computers [21, 131, 186].

As an example, the circuit structure for the distance-$(5, 1)$ quantum circuit bit-flip protected repetition code, which uses 10 qubits, is shown in Figure 3.3. The circuit pattern contains a first stabilisation component, represented by the chain of nearest neighbour CNOTs controlled by the data qubits and targeting the stabilisation qubits, followed by a round of syndrome measurements. At the centre of the quantum circuit, in green, one can observe the repeated application of a logical operation (an X gate) to all the logical qubits, followed by a second round of syndrome measurement. The code raw output is extracted by applying an ancilla readout.

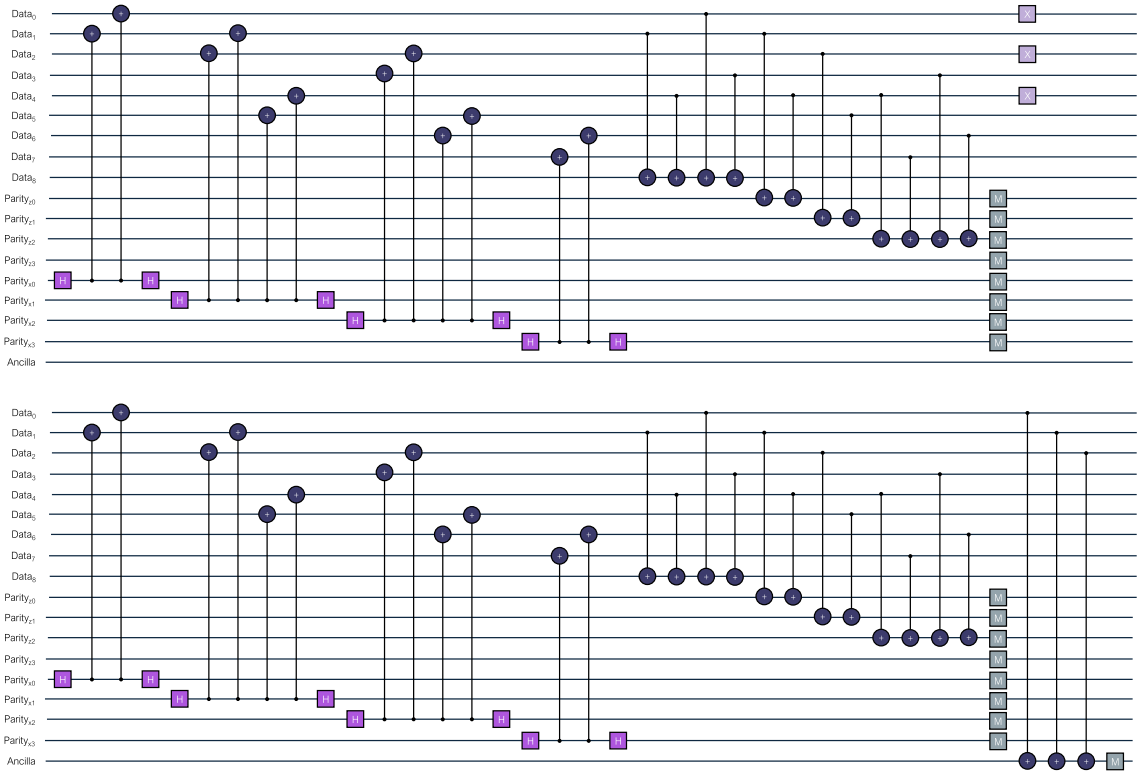


Figure 3.4: Distance-(3,3) XXZZ surface code.

3.3.2 XXZZ code

The XXZZ surface code is a rotated surface code generated by XXZZ and ZZXX Pauli strings, clockwise associated with the vertices of each face in a two-dimensional mesh, with one qubit in each vertex. It is virtually identical to the XZZX code, only varying in terms of Pauli strings generators for the stabiliser plaquettes [98]. It is an adaptation of the toric code with non-periodic boundaries [106]. The mesh is defined over two odd integers d_Z and d_X, one for the Z-error stabilisers (i.e. bit flips) and one for the X-error stabilisers (i.e. phase flip). The overall distance of the code is defined as $d = (d_Z, d_X)$, with the total number of qubits required to encode these circuits being $q_{XXZZ} = 2d_Z d_X$. This code has been tested on superconducting quantum computers, letting researchers achieve logical error rates lower than those of the physical qubits they are encoded with [182, 184].

As an example of the code's structure, specifically for the distance-$(3, 3)$ XXZZ surface code, is shown in Figure 3.4. A total of $n = d_Z d_X$ qubits are used to encode the data, $m = \left\lfloor \frac{(d_Z d_X)-1}{n} \right\rfloor$ qubits are used for measuring Z-basis errors, m qubits are used to detect X-basis errors and one final ancilla qubit is used to perform the raw code readout.

3.3.3 Simulation parameters

To provide realistic injection data, the quantum computer's intrinsic noise has been modelled as a depolarising channel, following the description provided in Section 3.2.1. This intrinsic noise model is parameterised with probability $p = 1\%$[110], unless otherwise noted. In absence of radiation-induced events all the tested configurations do not present output errors.

The approximated temporal damping function $\hat{T}(t)$ has been sampled over $n_s = 10$ equidistant points in time.

All simulations refer to a 5×6 bidimensional lattice as the coupling map, except for the architectural analysis in Section 3.4.4.

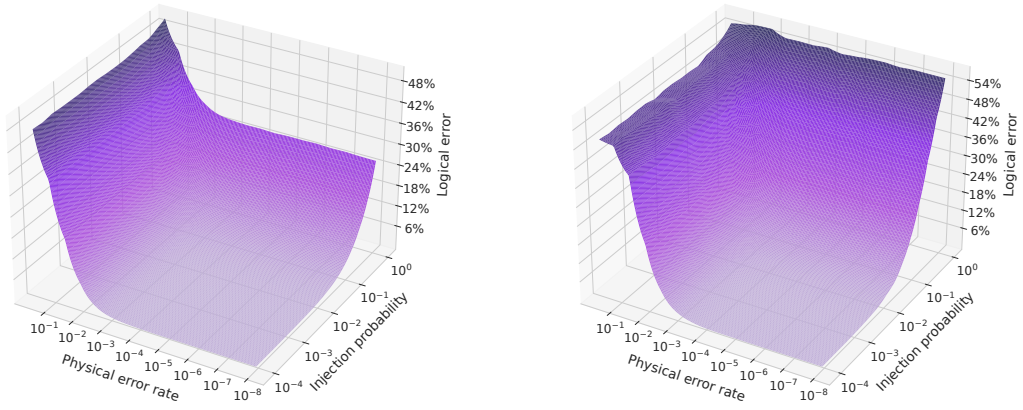
For both of the analysed surface code classes, each data qubit is initialised to $|0\rangle$, later encoding a logical X-gate operation. The surface code's circuit is repeated over multiple execution shots, over a simulated radiation event that lasts for 100 *ms*. The expected logical output post-decoding of the surface code is a logical $|1\rangle$ state, as detailed in the example circuit diagrams shown in Figures 3.3 and 3.4. Measurements are decoded through the MWPM algorithm and a logical error is detected whenever the output of the decoder is a logical $|0\rangle$ state. The logical error rate is computed as the number of shots that are decoded as a $|0\rangle$ state divided by the total number of shots.

3.4 Results

A total of four different analyses over two quantum error-correcting code classes have been performed. When referring to *injection data*, the whole time and space evolution of a given fault is being considered, unless otherwise noted.

3.4.1 Noise vs. radiation-induced faults analysis

Surface codes, in their various implementations, have been designed to correct intrinsic noise. With this first analysis, to answer the research question whether surface codes are effective in correcting radiation-induced faults, I evaluated: (1) how intrinsic noise and radiation-induced faults interoperate and (2) if the two events show interference patterns that influence the output logical error of the surface code. To do this, the logical error rate of surface codes have been correlated with the radiation-induced transient faults and the intrinsic noise model of a quantum computer. I report data on the distance-(5,1) repetition code and the distance-(3,3) XXZZ code. Both surface codes are transpiled over a square



(a) Distance-(5,1) repetition code.

(b) Distance-(3,3) XXZZ code.

Figure 3.5: Logical error landscape of the repetition and the surface code.

lattice architectural layout. Specifically, the repetition code lattice was of size 5×2, whilst the one for the XXZZ code was 5×4. Data from simulations with alternative parameters have unveiled a similar behaviour, and will not be further commented on. The root injection point has been deterministically chosen to be the qubit with index two for reproducibility reasons. A further analysis on the choice of the injection point is discussed in Section 3.4.4.

In Figure 3.5, one can observe the post-decoding logical error as a 3D plot, combining the effects of intrinsic device noise and radiation-induced faults. The bottom right axis represents the time evolution of the radiation fault parameterised by the root injection probability (100% at time of impact), which has a one-to-one mapping with time as seen in Section 3.2.2: at the time of strike, the probability to modify the qubit state is 100%, exponentially diminishing as time passes. The bottom left axis represents the physical error rate, that is the intensity p of the intrinsic noise model introduced in Section 3.2.1. I tested the intrinsic noise model over a range of values for p, the *physical error rate*, going from 10^{-8} up to 10^{-1}. The higher extreme for the physical error rate has been selected to highlight the effects of intrinsic noise on a scale similar to that of the transient error, whilst the lower extreme of 10^{-8} is the target error rate required to reach fault tolerance in quantum computers. A physical error rate on the order of 10^{-3} well approximates the noise behaviour of current quantum devices [131]. The full range of the time evolution of the transient fault has been considered, represented as the root injection probability at a given point in time. The post-decoding logical error of the surface code is then interpolated at each coordinate.

As shown in Figure 3.5, at the particle strike, when the root injection probability is close to 100%, the considered error correction codes show an average logical error rate 27% and

50%, respectively. In the case of the repetition-(5,1) code, the highest value for the logical error rate of 48% is reached when maximising both the intrinsic noise model's error rate, at 10^{-1}, and the root injection probability on the repetition qubit, at 10^{0} (100%). Similarly, the XXZZ-(3,3) code peaks at 54% under the same conditions. Both surface codes achieve a logical error rate lower than the physical error rate only when the latter is smaller than 10^{-3} [131]. This matches the surface code performance metrics presented in surface code simulation works [110]. Crucially, when reducing the physical error rate of the simulation to a regime which is unreachable for current quantum computers, such as 10^{-8}, the detrimental effects of the radiation-induced fault can still be observed, as the logical error reaches 24% for the repetition-(5,5) code and 52% for the XXZZ-distance(3,3) code. This provides a clear insight: regardless of the gate level accuracy of current or future quantum computers, radiation-induced faults will still catastrophically corrupt the outputs of error correction codes. As such, reaching extremely low qubit error rates will not be sufficient to counteract radiation-induced fault events.

Observation III.I

Particle impacts undermine surface code performances regardless of intrinsic qubit physical error rate.

The interaction of intrinsic noise and radiation-induced faults only show constructive interference, amplifying the overall error rate of the quantum computer, as no sudden pits on the surface are observed. This lets one infer that no recorded injection event has positively altered the output of the surface code. As such, the intrinsic noise model has proven to act as a lower limit to the accuracy of the surface code.

Observation III.II

Radiation-induced faults do not cause positive alterations of the surface code's output by reversing the effects of intrinsic noise, but rather amplify the logical error.

3.4.2 Code Distance analysis

The repetition and XXZZ code classes are parameterised by distance, which is represented by the tuple (d_Z, d_X). d_Z represents the number of qubits devoted to correct bit-flip errors, while d_X represents the same statistic for phase-flip errors. There are two research questions I aim to answer: (1) Does a larger surface code provide better protection from radiation-induced faults? (2) Does the use both bit-flip and phase-flip protection in the XXZZ code improve the performance over the exclusively bit-flip protected repetition code?

To answer the two research questions, the code distance has been correlated with the logical error over the two considered surface code classes. For each surface code distance, I have considered one corrupted qubit, highlighting the fault's magnitude at time of impact ($t = 0$). Furthermore, I have removed the fault's spatial expansion to neighbouring qubits, whose impact is analysed in detail in Section 3.4.3. This has been done in order to highlight the moment of maximal criticality of the *non-unitary* reset error, which occurs at the beginning of the event. Each code followed the interconnection constraints of a lattice of size 5×6, scaled down according to the qubit requirements of each code class and distance. A subset of connected subgraphs in the lattice has been selected, treating each subgraph as a hypernode inside which each qubit would undergo the same fault event. The results have then been grouped by the size of the subgraphs, extrapolating the median error across all subset sizes.

In Figure 3.6 the post-decoding logical error has been plotted on the x-axis, with respect to the surface code distance on the y-axis, while the hue representing the number of physical qubits in the surface code's circuit. The bit-flip repetition code class boasts a logical error rate of $\sim 8\%$ at distance-(3,1). For higher distance versions of the code, the logical error steadily increases, reaching a peak of 20.5% in the distance-(13,1) repetition code. A slight logical error difference is observed in the distance-(15,1) repetition code, with a logical error of 19.5%, which is to be attributed to statistical noise.

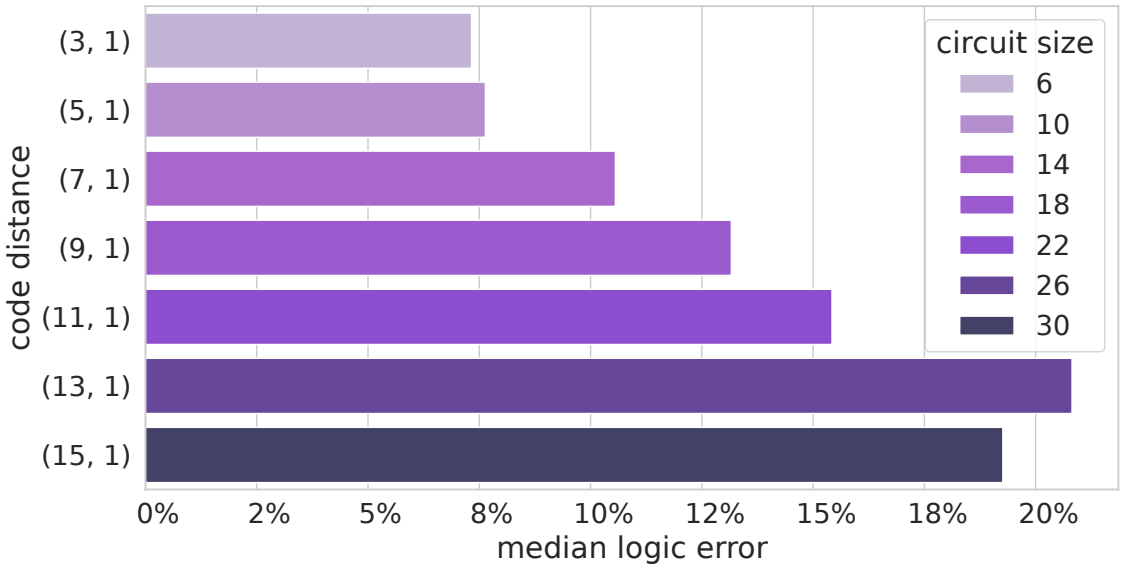
The XXZZ code boasts both bit-flip and phase-flip error correction capabilities, as detailed in Section 3.3.2. In the distance-(1,3) case, one can observe a logical error rate of $\sim 12\%$, while in the distance-(3,1) case an error rate of $\sim 7.5\%$ is registered instead. When considering the distance-(3,3) case, the logical error reaches about $\sim 21\%$. In the distance-(3,5) and distance-(5,3) codes, a behaviour similar to that of the distance-(1,3) and the distance-(3,1) codes can be noticed, albeit with higher logical error rates, of $\sim 29.5\%$ and $\sim 26\%$.

Observation III.III

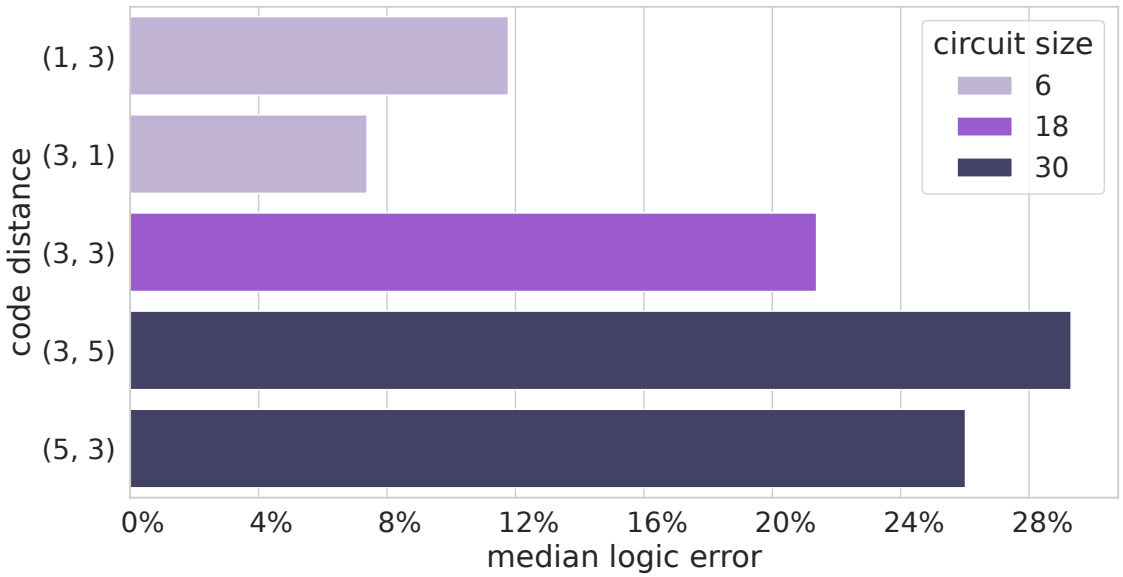
Larger surface codes are more sensitive to radiation-induced faults, reaching larger logical error rates in the presence of the same fault intensity.

This is especially highlighted in the bit-flip repetition code plot in Figure 3.6, as given a single non-unitary and non-spreading erasure error, the code will generally perform worse. This goes in contrast with the fact that under sufficiently low intrinsic noise thresholds, larger surface codes would imply lower error rates.

For like-sized surface codes, bit-flip protection stabilisers are up to 10% more effective



(a) Bit-flip repetition code.



(b) XXZZ code.

Figure 3.6: Logical error criticality by code distance.

at dealing with radiation-induced errors when compared to phase-flip protection stabilisers.

Observation III.IV

Bit-flip protection in surface codes is more efficient at dealing with radiation-induced faults.

This is noticeable in the XXZZ code plot of Figure 3.6, as the distance-(3,1) code and the

distance-(5,3) code outperform their respective distance-(1,3) and distance-(3,5) counterparts. This checks out as the erasure error introduced when modelling qubit corruption is a Z-basis transformation.

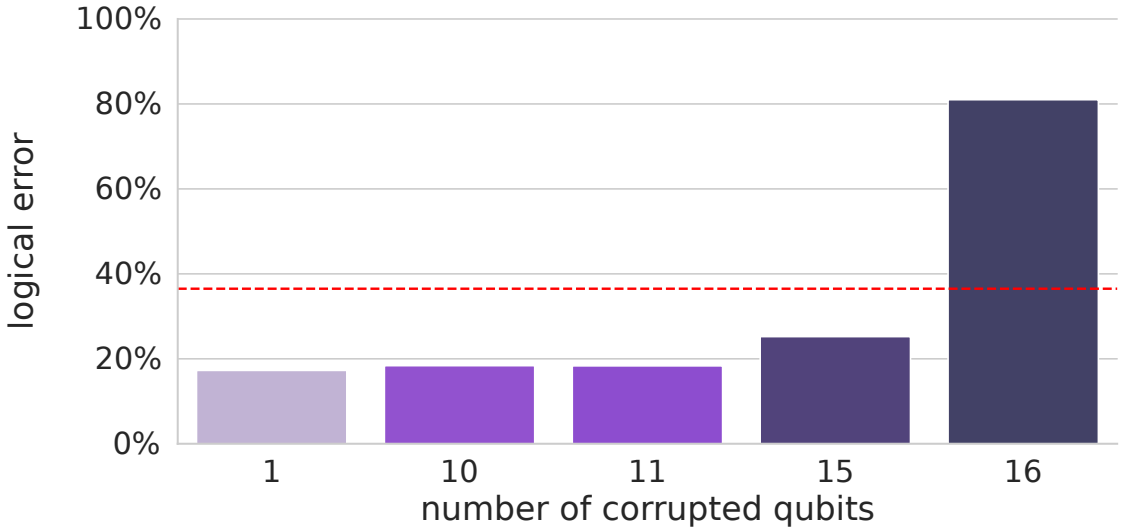
3.4.3 Spreading fault vs. erasure fault

Particle strike events can hinder multiple qubits at once, spreading throughout the quantum chip as radiation-induced faults, a behaviour strikingly different from that of events characterising intrinsic noise. This analysis seeks to answer these questions: (1) How many simultaneous reset operations are needed to approximate the effects of a single spreading radiation-induced fault? (2) What is the impact of a spreading radiation-induced fault when compared to a fault that does not evolve over the spatial domain?

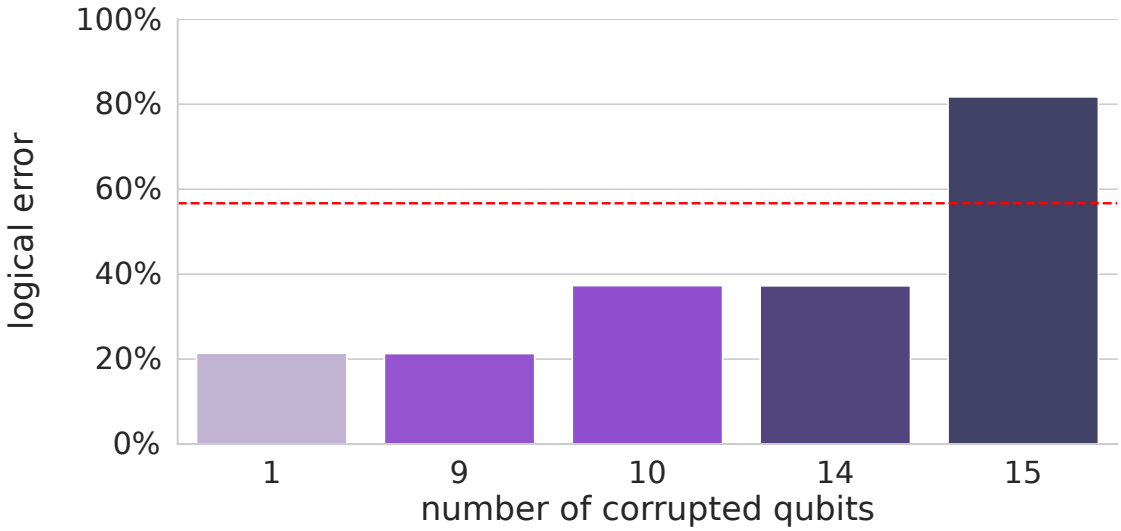
To answer these questions, I have selected the connected subgraphs over a 5×6 lattice architectural graph, then inject all qubits in the subgroup with the same reset event, extracting the median error across all subset sizes, comparing it to the horizontal line of the radiation-induced logical error. I highlight the surface code's performance at time $t = 0$, as it is the most critical moment in the evolution of the radiation-induced fault. On these configurations, I have considered the reset errors to impact only the root injection points, leaving neighbouring qubits unaffected.

In Figure 3.7, only the post-decoding logical error for the *distance-(15,1)* repetition code and the *distance-(3,3)* XXZZ code have been presented, as emblematic cases. On the x-axis, the number of corrupted qubits, which undergo an isolated reset error, is compared to the logical error of a single spreading radiation-induced fault, represented by the red line. The repetition code, when only one qubit is reset, shows a logical error rate of $\sim 17\%$, an absolute difference of $\sim 17\%$ with respect to the radiation-induced fault. Increasing the number of qubits being reset monotonically increases the logical error rate for the correlated case, reaching a value of $\sim 25\%$ when 15 qubits are erased. As soon as more than half of the total number of qubits in the circuit are reset, which in the case of the distance-(15,1) repetition code is 15 qubits, the logical error rate reaches $\sim 80\%$, a value larger than the radiation-induced logical error rate of $\sim 34\%$.

The XXZZ code, when a single qubit is reset, shows a logical error rate of $\sim 21\%$, which is almost one third of the logical error rate obtained with a radiation-induced fault. As the number of erasure errors increases, the logical error rate worsens, and once ten qubits are corrupted, the logical error rate reaches $\sim 36\%$. Performance degrades again once at least 15 qubits are corrupted, reaching an error rate of $\sim 80\%$, a logical error rate that exceeds the one of a single-qubit radiation-induced fault.



(a) bit-flip distance-(15,1) repetition code.



(b) XXZZ distance-(3,3) code.

Figure 3.7: Effect of a spreading fault on the logical error rate.

A single radiation-induced fault, despite a rapid damping in intensity as distance grows (shown in Figure 3.2), has shown significantly more detrimental effects than multiple erasure faults. This further highlights the danger of radiation-induced faults.

Observation III.V

A single spatially correlated radiation-induced fault is more detrimental than multiple uncorrelated erasure events.

The analysis highlights that limiting the spatial spread of radiation-induced faults is

crucial to guarantee the performance of surface codes. Hardware solutions that promise to prevent radiation events from spreading over the substrate [132, 150–152], can then improve the performance of surface codes. Given that qubit isolation solutions will have a *significant* impact on the production cost of quantum computers, it is fundamental to be able to test and validate their effect beforehand. While completely removing the spatial spread of transient errors through complete isolation of each qubit on the substrate is an unreachable task, these techniques may prove to have a positive impact on the error correction capabilities of surface codes. However, rarer events, that can independently corrupt multiple physical qubits at the same time, will still pose a threat to quantum reliability.

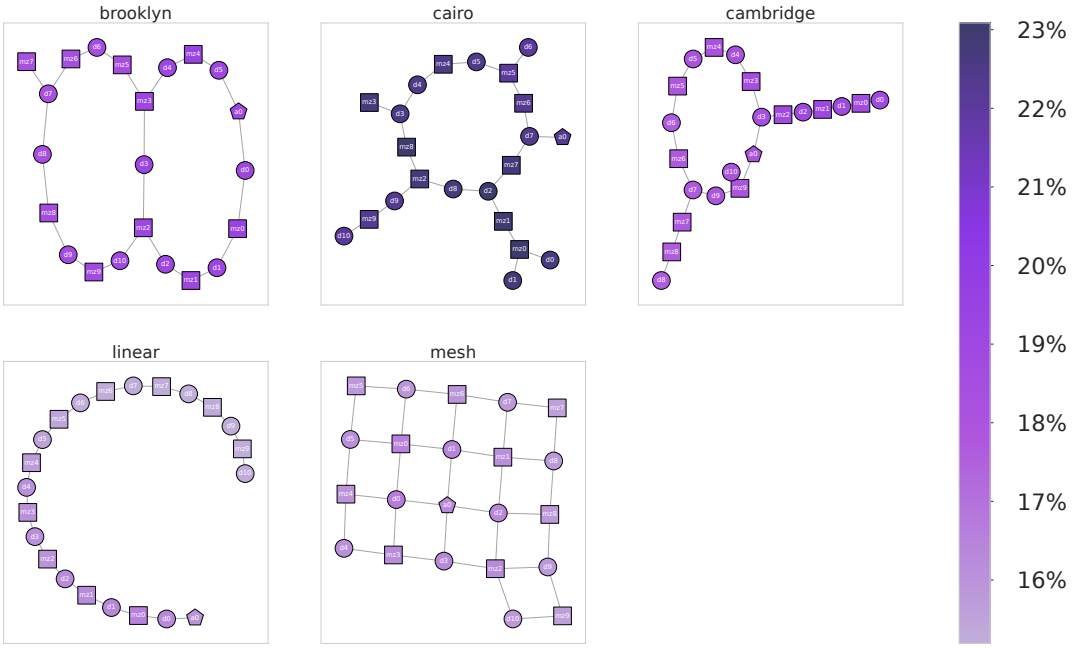
Observation III.VI

Limiting the spatial spread of the radiation-induced errors significantly improves the error correction capabilities of surface codes, increasing the threshold for the number of concurrent erasure errors that they can withstand.

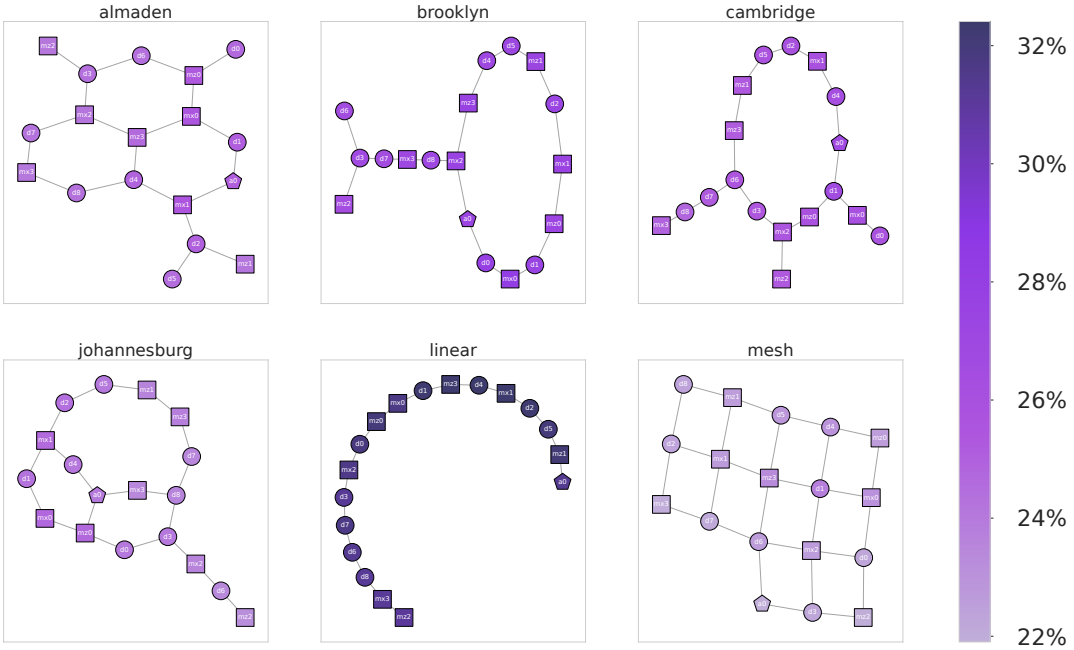
3.4.4 Hardware architecture analysis

Quantum circuits need to be transpiled following the architecture constraints of a quantum computer to be executed. The questions object of this analysis are: (1) Does the choice of the architectural connectivity graph impact the logical error rate? (2) Are specific qubits in the surface code more critical than others? To answer these questions, I have transpiled the surface codes to various architectural connectivity graphs, injecting a single radiation-induced fault on each possible root injection point, and following its evolution over the temporal and spatial domain. In those graphs, each edge represents an integer distance of one between two nodes, omitting unused qubits from the original coupling map. The transpilation process has been done with the default optimisation factor and without forcing the qubit positioning in the initial or final layouts. The tested coupling maps include a 5×6 mesh, a linear graph and a few of the hardware connectivity patterns publicly available from the Qiskit library [38]. These latter graphs have been filtered according to the number of qubits required to represent the surface code and communication constraints. I have considered a one-to-one mapping between the architectural graph and each qubit's physical position on the quantum chip, and a constant weight of one on each edge.

The results over each architectural graph are presented in Figure 3.8. The hue of each node represents the median logical error over the fault's duration obtained from a particle impact in that qubit. The shape of the node represents the qubit's function in the surface code: data qubit nodes are enclosed by a circle, stabiliser qubit nodes are enclosed by a



(a) Distance-(11,1) bit-flip repetition code.



(b) Distance-(3,3) XXZZ code.

Figure 3.8: Logical error rate by corrupted qubit on different architectures.

square and the ancilla qubit nodes are enclosed by a pentagon. Only the results for the *distance-(11,1)* repetition code and the *distance-(3,3)* XXZZ code are shown. Note the color map scale difference between the left and the right subplots.

The repetition code, due to its 22 qubit size constraint, has been tested on these coupling

maps: linear, mesh (5×6), Brooklyn, Cairo and Cambridge. The linear architecture boasts the lowest median logical error range. It ranges from $\sim 15\%$ when injecting either the ancilla qubit or the lower indexed data and stabiliser qubits, to $\sim 17\%$ for the injections on higher indexed qubits. The mesh architecture highlights a similar correlation, as the ancilla and the lower indexed data and stabilisation qubits show slightly larger median errors of $\sim 17\%$ than their higher indexed counterparts ($\sim 16\%$). This correlation is true, albeit for slightly different error ranges, across all the other coupling maps. The Cairo architecture has the worst performance, reaching 23% error at its peak, and lower thresholds of $\sim 21.5\%$. The Brooklyn architecture shows a relatively stable median logical error rate of $\sim 19\%$, while the Cambridge architecture's median logical error ranges from $\sim 17\%$ to $\sim 19\%$. The linear and mesh architectures boast the lowest error rates since they better support the nearest neighbour interactions of the repetition code, which requires that stabiliser qubits are placed on nodes with degree ≥ 2.

The XXZZ code, given the smaller size requirement of 18 qubits, has been tested on the following coupling maps: complete, linear, mesh (5×4), Almaden, Brooklyn, Cambridge and Johannesburg. In this case, the mesh architecture sports the lowest median logical error, with a peak at $\sim 24.5\%$, and the lowest logical error at $\sim 22\%$. The Johannesburg and Almaden architectures show higher error rates, ranging from $\sim 23\%$ to $\sim 26\%$. The Cambridge architecture has a higher logical error variation, which peaks on lower indexed qubits at $\sim 27\%$, whilst the Brooklyn architecture performs worse, at upwards of $\sim 28\%$ logical error rates and little variation with respect to the position of the corrupted qubit. The linear architecture has a significantly worse performance when compared with the others. This is due to the fact that the XXZZ code, being a rotated code, requires that the stabiliser qubits are placed on nodes with degree ≥ 4, while the linear architecture has an average node degree $\lesssim 2$, thus introducing a large overhead in SWAP operations.

The analysis unveiled a correlation between the index of the qubit in the surface code, and the median logical error rate registered when that qubit acts as the locus of a transient fault event. This matches the flow of information across qubits during the execution of the surface code's circuit. Such behaviour can be explained by looking at the DAG representation of the circuits analysed, which highlights the sequential dependence across multiple gate operations, as qubits get linked together by successive CNOT and SWAP gates. When a single particle hits a qubit and spreads over the code, it will affect also the descendants in the DAG. The lower error rate in higher indexed qubits is then just a matter of ordering convention in the temporal sequence of quantum gates applied.

Observation III.VII

Radiation-induced transient errors have a stronger impact on qubits that are used earlier in the sequence of gates in a quantum circuit.

No circuit rewriting or reordering technique, such as the ones performed by a transpiler, can avoid this effect. This is because the sequential dependence across gate operations in the analysed surface codes is intrinsic to their formalism.

Moreover, the choice of a quantum computer architecture significantly impacts the performance of surface codes according to their connectivity requirements, that is the average qubit degree required to represent the quantum circuit. While the repetition code works exceptionally well when transpiled on a linear architecture, the XXZZ code suffers from a large overhead of SWAP operations, thus increasing the number of gate operations and the chance for a radiation-induced fault to propagate.

Observation III.VIII

If the coupling map of a quantum computer is sufficiently connected, there will be less communication overhead, preventing radiation-induced faults from spreading.

3.5 Chapter summary

In this chapter, I have presented an analysis of the impact of radiation-induced faults on the efficacy of two classes of QEC codes, namely the repetition and XXZZ codes. Concerning **RQ1**, QEC surface codes cannot withstand the faults introduced by radiation. The observed extrapolated post-QEC logical error peaks reach 24% and 54%. Answering **RQ2** pointed to the fact that one can indeed tune the QEC code to improve the reliability to radiation strikes. The analysis shows that, given an equivalent number of physical qubits, the bit-flip only correction codes are up to 10% more effective against radiation than bit-phase correction codes. Moreover, choosing properly the underlying hardware topology can further increase the radiation fault correction capability from 7% to upwards of 10%. These improvements do not introduce any additional overhead to the QEC. At last, **RQ3** has been tackled, finding that in order to increase a surface code's resistance to radiation faults, bit-flip protection should be prioritised. Moreover, qubit charge wells are a promising solution to reduce the impact of transient faults, by preventing their spread. These insights can guide the design of future QEC codes able to cope also with radiation-induced errors.

Future research directions include the implementation and testing of new surface codes following this article's observations, and the usage of the presented post-QEC logical error

rates to perform post-QEC logical layer fault injection. I intend to propagate the logical fault induced by radiation in the coded qubit status in quantum circuits. The aim is to identify the critical logical shifts for a given circuit to further better tune the QEC correction capabilities.

Chapter 4

Radiation Event Identification at runtime

The quest for universal superconducting quantum computing is hindered by noise and errors. It has been proven that QEC codes will lay at the foundation of fault-tolerant quantum computing. However, cosmic-ray induced correlated errors, which are the most detrimental events that can impact superconducting quantum computers, are yet to be efficiently tackled, since they are sufficient to induce QEC failure [21, 101]. In order to reach fault tolerance, one must also develop radiation aware methods to complement QEC.

4.1 Objectives

In this chapter, I present the first algorithm to effectively exploit syndrome information for the efficient detection of radiation events in superconducting quantum devices at runtime. I aim to tackle these three research questions:

- **RQ1**: Can one actively detect high energy events and their area-of-effect in QEC codes at runtime?
- **RQ2**: How do high energy events affect QEC codes and decoders?
- **RQ3**: Can one leverage fault information to improve the decoder's performance?

This chapter refers to the contents of the article "Radiation-Induced Fault Detection in Superconducting Quantum Devices", written by M. Vallero et al. and published in the Advanced Quantum Technologies journal [187].

I have performed a thorough analysis of simulated Rotated Surface codes injecting over 11 million physics-modelled radiation-induced faults. This includes considering the properties of the X and Z check bases, the impact of code distance, and the decoder's time to solution constraints. The hereby presented technique detects 100% of injected faults, regardless of the impact's position. Moreover, the algorithm accurately identifies both the radiation impact centre and the area affected, with an overhead lower than 0.3% the decoding time. Additionally, the fault identification information is used to propose a radiation fault correction technique that improves of up to 20% the output correctness compared to existing decoders.

The observations in this chapter are backed by more than 11 million QEC shots simulated following a physics-derived model of the interaction of radiation with superconducting qubits. This model considers the distribution of charge deposition from the impinging particle over the quantum chip's surface, and the complete temporal evolution of its transient effects. The implementation of the fault model is a novel software written as an extension of the STIM library [54], and has been disclosed as open source software as part of this thesis' contributions. The choice of a stabiliser-based simulation method is justified by the focus of this chapter on QEC codes, as other commonly used quantum circuit simulation methods, such as tensor network or state vector, would require considerably more computational resources at the scales hereby considered [41, 158, 188].

Not only is it possible to detect radiation-induced events, but also to identify at runtime their area-of-effect (i.e., the likely set of affected qubits) through a novel QEC-agnostic algorithm that introduces minimal overhead. This algorithm, dubbed *Radiation Event Identification (REI)*, processes information from the syndrome measurements of a QEC code, correlating it with an internal representation of the qubit's position on the quantum chip. The incidence of false positives is significantly lower than the incidence of intrinsic noise in the quantum device, which in the performed experiments is set to $p = 10^{-5}$ to replicate the behaviour of near future quantum devices. I have characterised the real-time REI subroutine's predicted area-of-effect and execution time performance over different distance measures of the Rotated Surface code, injecting radiation faults in a set of positions of the quantum chip. Furthermore, I have analysed the embedding of multiple separate quantum error correction codes onto a large quantum chip, benchmarking the efficacy of the Minimum Weight Perfect Matching decoder. Its effectiveness at dealing with syndrome measurements corrupted by radiation is measured in both a single logical qubit setting and in a multiple logical qubit setting, observing how the logical error rate correlates with their position on the quantum chip. At last, the performance of a suite of classical graph-based decoders is measured against syndrome measurements corrupted by radiation. In doing

so, such decoders are compared to a novel empirical technique, dubbed *RadMatching*, that aims at correcting radiation's effects just before the decoding step.

This chapter links back to concepts previously introduced in Chapter 1, such as intrinsic noise in Section 1.1.5, radiation events in Section 1.1.7 and quantum error correction in Section 1.1.6. The remainder of the chapter is structured as follows. Section 4.2 provides a quick grounding on the rotated surface code and on syndrome decoding. Section 4.3 describes the formalism and algorithmic implementation of the intrinsic noise model and the radiation-induced fault model. The QEC codes and decoders object of analysis are detailed in Section 4.4, to later go over the results and answer the research questions in Section 4.5. Conclusions, along details on future investigations, are drawn in Section 4.6.

4.2 Background

In this section I provide a light background on the topics which are strictly necessary to follow along with the remainder of the chapter.

4.2.1 Rotated surface code

For the purposes of this chapter, I have considered the Rotated Surface code [189, 190], which is an alternative version of the surface code which has been rotated by $\pi/4$. This makes it so that data qubits are placed on the vertices of a square lattice, and the X and Z stabiliser qubits are placed at the centre of the plaquettes in a draughtboard pattern. The Rotated Surface code of distance d maps the logical state of a single qubit onto $2d^2 - 1$ qubits, with d^2 data qubits and $d^2 - 1$ stabiliser qubits.

4.2.2 Syndrome decoders

The syndrome measurement obtained from a round of QEC must be processed in order to identify which error mechanisms in the Tanner graph might have the highest chance to be responsible for the syndrome's defects. This can be visualised as finding the set of edges connecting defect pairs in the Tanner graph which covers the data qubits that are most likely to have triggered a defect, finding a *matching*. However, not all syndrome measurements have even parity, as such one odd stabiliser might be left unmatched. As such, the Tanner graph is expanded with a boundary hyper-vertex, that acts as an escape route that can match any number of defects. The process of finding the most probable set of defect-joining edges, known as maximum likelihood decoding, is known to be NP-hard, and thus alternative

and faster workaround techniques have been developed by the community. The idea behind these alternative approaches is that intrinsic noise leads to uncorrelated pairwise defects, as such the shortest path between two unmatched syndromes should cover the most probable defect source. This information is then used to both read out the logical state of the QEC code and to correct the sources of error.

Many decoding techniques are available in the literature, ranging from graph matching algorithms [109], to Machine Learning implementations that specialise on specific noise profiles and QEC codes [101, 191, 192], to Tensor Network contraction approaches [101, 193, 194]. Generally, graph-based algorithms boast the best compromise between fast time to solution and excellent accuracy [101, 195]. In fact, the decoding step is subject to very strict time constraints, since it needs to keep pace with the syndrome measurement generation rate of real quantum computers to avoid the buildup of an exponential backlog of syndromes to be processed. To fulfil these requirements, Tanner graph based algorithms have been selected as prime candidates for the construction of *ad hoc* Application specific integrated circuit (ASIC) systems [122, 124, 195]. Considering current superconducting technology, the decoding latency is expected to be less than 1 μs and the data throughput should support upwards of 1 TB/s. However, these same constraints rule out Machine Learning and Tensor Network approaches. Given the demonstrative aim of this chapter, I will only evaluate the performance of Tanner graph based approaches, which include the following.

Minimum Weight Perfect Matching. This decoder uses a variation of Edmond's Blossom algorithm [196] to find a matching of the Tanner graph which is minimal, according to the sum of the weights in the matching edge set, where the weight of one edge is the distance between its vertices.

Belief Matching. The Belief Matching (BM) decoder is composed of two subroutines, with a Belief Propagation (BP) step immediately followed by a MWPM step [197]. Similarly to the Belief Find decoder, the syndrome induced Tanner graph is first tentatively decoded by the BP subroutine. In case of failure, the information from the BP step is used to set the weights for the MWPM step.

Union Find. This decoder uses the Union Find (UF) algorithm applied to the parity check matrix of the QEC code to infer the estimated correction vector from the syndrome data [116, 198].

Belief Find. This decoder is composed of two routines. At first, the BP algorithm is used to try to find a correction vector. If BP does not converge, the edge weights obtained from that step are fed to the Union Find algorithm [197].

The implementation of the BP and UF subroutines is provided by the *ldpc* library [199], while the MWPM subroutine is provided by the *Higgott2022pymatching* library [109].

4.3 Quantum chip, Noise and Fault Modelling

This section details the general characteristics of the considered superconducting quantum chip model, the algorithmic implementation of the intrinsic noise model for a general superconducting quantum computer and the radiation fault model used in all the analyses of Section 4.5.

4.3.1 Quantum chip model

I have modelled a generic quantum chip, which is able to execute any quantum gate amongst the ones available in the STIM library. This quantum chip model follows the reported topology and hardware properties of top of the line quantum computers at the time of writing [101, 200]. The characteristic τ_1 coherence time of this simulated quantum chip has been set to be of $85\mu s$, while the gate durations have been set to $25ns$ for single-qubit gates, $32ns$ for two qubit gates and $58ns$ for measure and reset gates, respectively, following values from real world implementations [201–204]. Moreover, the topology of the interconnections between qubits has been selected so that the considered QEC codes could be easily mapped on the quantum chip without incurring in the need of SWAP operations. The mapping is found through an homomorphic subgraph solver, by searching the interconnection structure of a given quantum circuit into the quantum chip's topology. The topology is built from the repetition over two axes of a single connected component by a number or *rows* and *columns*. An example of the topology of the quantum chip, as seen from a top-down view, is provided in Figure 4.1: it has been generated from the repetition of a cross-shaped minimal connected component over 20 diagonal rows and 20 diagonal columns, for a total of 760 qubits.

Although these characteristics are representative of current quantum devices, thanks to the modular and open source design of the code library hereby presented, any current and future quantum chip can be easily modelled and tested as well, including topologies, gate timings and gate sets.

4.3.2 Intrinsic noise model

Superconducting quantum computers are subject to an ensemble of physical phenomena that hinder the accuracy of quantum gate operations. Given the complexity of all the variables at play, QEC codes are usually tested against artificial noise that optimally approximates a real quantum computer's behaviour. It is common practice in the literature to compose such *intrinsic noise models* through the usage of *Pauli operators*, under the umbrella

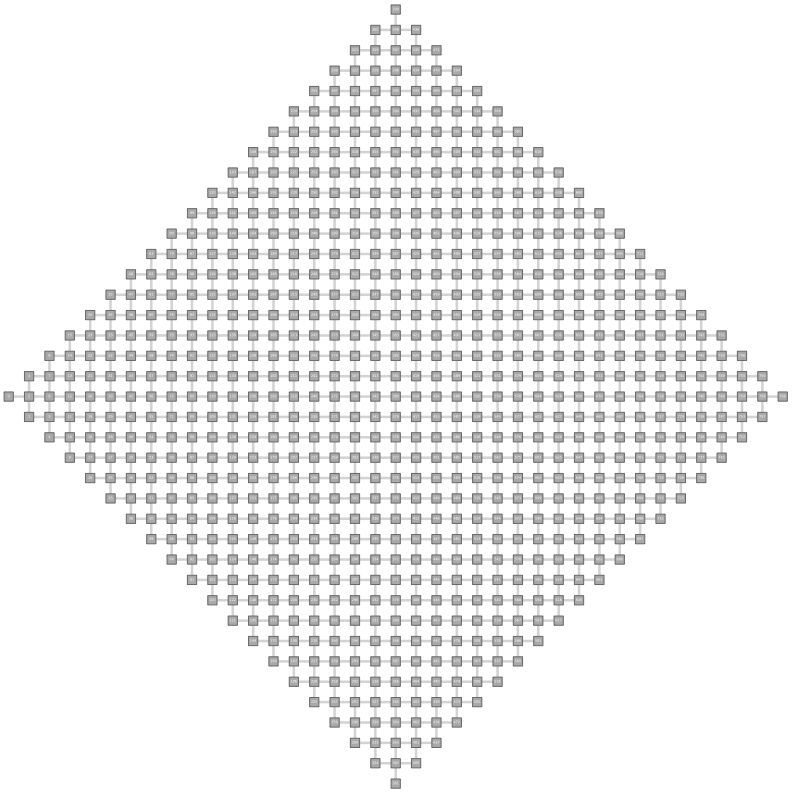


Figure 4.1: Rotated square mesh quantum chip topology.

terminology of *depolarisation error models* [85]. They are parameterised by a *noise error rate* p, which is meant to match the average measured error rate of a real quantum computer. The considered intrinsic noise model is a variation of the *superconducting-inspired 1000 ns cycle* (SI1000) noise model [86]. I have identified four sets of quantum operations: the set of single-qubit gates $\mathcal{Q}_1$, the set of two qubit gates $\mathcal{Q}_2$, the set of reset gates $\mathcal{R}$ and the set of measurement operators $\mathcal{M}$. Noise is thus introduced in the quantum circuit operator that probabilistically triggers either an I operator or an operator in the set $\mathcal{P}_e \doteq \{X, Y, Z\}$ after each qubit involved in a quantum gate or measurement operator, governed by the model's *noise error rate*. In the case of the $\mathcal{Q}_2$ set, the noise operator is the tensor product of two Pauli operators, which can either be I or an operator in the $\mathcal{P}_e$ set. Each of those noise operators is then triggered independently, giving rise to uncorrelated errors across qubits and over time, following the relations in Table 4.1.

Considering that the coherence time of modern superconducting quantum computers is orders of magnitude longer than the execution time of a single QEC code round, it is

Instrinsic noise model						
Set	I	$\mathcal{P}_e$	II	$I\mathcal{P}_e$	$\mathcal{P}_eI$	$\mathcal{P}_e\mathcal{P}_e$
$\mathcal{Q}_1$	$1-\frac{p}{10}$	$\frac{p}{30}$	–	–	–	–
$\mathcal{Q}_2$	–	–	$1-p$	$\frac{p}{5}$	$\frac{p}{5}$	$\frac{3p}{5}$
$\mathcal{M}$	$1-5p$	$\frac{5p}{3}$	–	–	–	–
$\mathcal{R}$	$1-2p$	$\frac{2p}{3}$	–	–	–	–

Table 4.1: Pauli error probability by quantum gate class.

reasonable to assume that the noise error rate p remains constant throughout the execution of the simulations. All the simulation results discussed in Section 4.5 refer to an intrinsic noise rate $p = 10^{-5}$.

4.3.3 Radiation fault model

It is known that high energy events disturb the quantum information stored in multiple qubits, ultimately reducing their coherence time. This behaviour has been modelled via a probabilistic *Y gate* prepended to each quantum gate operation, that induces a loss of coherence by erasing the information in the qubit according to a probability p_{q_i}. This probability depends on three factors, namely the time elapsed between the previous gate operation on a qubit and the current gate operation on that same qubit, the spatial distance between the impact locus of the high energy event and the qubit, and the time elapsed since the beginning of the high energy event. From this, it follows that the high energy event's intensity depends on the quantum circuit's structure, the architecture of the quantum chip and time.

Over the **temporal dimension**, the probability for a qubit to undergo decoherence depends on the τ_1 measure and the time elapsed between the previous and the current quantum gate Δt_g. High energy events significantly reduce τ_1, spiking in intensity as soon as energy is deposited and gradually wearing off over time. This reduction has been modelled through the $\tau_{rad}(t)$ function, detailed in Equation 4.1, that is parameterised over the time t at which the quantum gate is applied, and depends on three constants: the characteristic τ_1 of the quantum computer, the wall-clock time at the start of the high energy event t_{rad} and the total duration of the high energy event Δt_{rad}.

$$\tau_{rad}(t) = \tau_1 e^{10\left(\frac{t-t_{rad}}{\Delta t_{rad}}-1\right)} \tag{4.1}$$

At the locus of the high energy event, the probability p_{root} of the fault to trigger is computed

through the $T(\Delta t_g, t)$ function, detailed in Equation 4.2, which also depends on the current value of $\tau_{rad}(t)$.

$$T(\Delta t_g, t) = 1 - e^{-\frac{\Delta t_g}{\tau_{rad}(t)}} \tag{4.2}$$

Following the device model described in Section 4.3.1, the time required to execute a round of quantum error correction will range from a few hundred to a few thousand nanoseconds, depending on the depth of the quantum circuit that encodes the QEC code.

Over the **spatial dimension**, the charge deposited by a high energy event spreads across the quantum chip, where the intensity of the radiation fault becomes lower the further away a qubit is from the impact point. Knowing the planar coordinates of each qubit on the quantum chip's architecture, one can compute the Euclidean distance over two dimensions Δs from the locus of radiation to any qubit on the quantum chip. The $S(\Delta s)$ function is used to model the decreasing intensity of the radiation fault over the quantum chip, following the inverse square of the distance from the locus of radiation, as shown in Equation 4.3.

$$S(\Delta s) = \frac{1}{(\Delta s + 1)^2} \tag{4.3}$$

Piecing Equations 4.1, 4.2 and 4.3 together let me define the $P(\Delta s, \Delta t, t)$ function, which represents the probability p_{q_i} of the qubit of index i to undergo a radiation-induced erasure error, as detailed in Equation 4.4.

$$P(\Delta s, \Delta t_g, t) = S(\Delta s)T(\Delta t_g, t) \tag{4.4}$$

Observation IV.I

The radiation-induced fault intensity depends on a qubit's distance from the source of radiation, the time since the beginning of the radiation event and the idle time between gates.

4.4 Radiation Event Identification (REI)

The main idea behind the REI subroutine is to detect and characterise a radiation-induced fault making use of information regarding the device's architecture, the QEC code's mapping on the device, and a dynamic backlog made of the previous syndrome measurements. This algorithm has been explicitly designed to be agnostic of the QEC class, making it a widely applicable subroutine for quantum error correction decoders. In Section 4.5.5 I also propose how to use this information to move towards radiation-aware *decoding*. The term *identification* implies both the ability of the algorithm to discern whether a radiation event is

happening in real time, and that of knowing the approximate centre of the impact location and its extension over the quantum chip. REI is composed of three steps: (1) radiation-fault detection, (2) radiation impact location identification, and (3) area affected by radiation. The latter two steps are necessary to evaluate if the unaffected qubits can still be used to reconstruct the information regardless of the radiation-induced corruption.

Given the radiation fault model presented in Section 4.3.3, it is known that high energy events show correlations in space, over neighbouring qubits on the quantum chip, and in time, across time spans that last for thousands of QEC shots. The approach is presented in Algorithm 1, jointly with the information processing scheme of Figure 4.2.

Since syndrome measurements are processed sequentially over time, the decoder keeps track of the last K_{max} syndromes with a First In First Out (FIFO) policy, in order to correlate radiation errors in time. The backlog of $K \in [1, K_{max}]$ measured syndromes of size S is stored in a dynamic matrix *s_matrix* of size $K \times S$. Whenever a new syndrome measurement is provided, the oldest syndrome measurement is discarded, and the current syndrome measurement is appended as the last row of *s_matrix*. This latter matrix is then reduced to a vector *s_vector* of size S by summing over all the column values, and then normalised by the number of rows in *s_matrix*. A vector *q_defects* of size Q is instantiated, following the number of physical qubits in the considered QEC code. Each stabiliser measurement in a QEC code is hosted on one specific physical qubit. The *qubit_to_stabilisers* dictionary maps each physical qubit index q to a vector of stabiliser indexes s that are hosted on q. Iterating over all physical qubits q used by the QEC code, the sum of the elements in *s_vector* with indexes $i \in s$, normalised by the cardinality of s are stored in *q_defects*[q]. The *q_defects* vector is pruned, keeping only those elements which are strictly larger than $alpha = 1/((rounds + 1)S)$, and its results are stored in the dictionary $pruned\_qf$. If the number of elements in $pruned\_qf$ is smaller than two, no centroid can be found, and the subroutine returns.

Otherwise, a matrix xye is instantiated, with rows equal to the number of elements in $pruned\_qf$, and three columns. Iterating with index $i \in [0, pruned\_qf.size]$ over the (q, e) pairs in $pruned\_qf$, the i-th row of xye is set as the x and y coordinate of qubit q and its defect incidence rate e. The Euclidean distance amongst all the qubits' planar coordinates in xye are then computed and stored in a distance matrix $corr\_matrix$. In doing so, the diagonal elements, which represent the distance of each qubit with itself, are ignored. The minimum value for each column in the $corr\_matrix$ is filtered and collected in the $closest\_corr\_qubits$ vector, which is then averaged to compute the $correlation\_factor$. If the latter is found to be strictly greater than the average minimum Euclidean distance between all qubits in the quantum chip's topology, then the events are

Algorithm 1: Radiation Event Identification (REI)

Input: `syndrome`
Output: `(x, y), radius`

if $s\_matrix.rows = 0$ **then**
  $s\_matrix \leftarrow syndrome$
else
  if $s\_matrix.rows \geq K_{max}$ **then**
    s_matrix`.delete(row=0)`
  end
  s_matrix`.append(`$syndrome$`)`
end
$s\_vector \leftarrow s\_matrix$`.sum(axis="cols")`
$s\_vector \leftarrow s\_vector / s\_matrix$`.rows`
$q\_defects \leftarrow$ `vector(shape=(`Q`,), init_value=0)`
for $(q, s) \in qubit\_to\_stabilisers$ **do**
  $q\_defects[q] \leftarrow s\_vector[s]$`.sum()/`$s$`.rows`
end
$alpha \leftarrow 1/((rounds + 1)s\_matrix$`.rows)`
$pruned\_qf \leftarrow q\_defects[q\_defects > alpha]$
if $pruned\_qf.rows \leq 2$ **then**
  return `NULL`
end
$xye \leftarrow$ `matrix(rows=`$pruned\_qf$`.size, cols=3)`
for $(i,(q,e)) \in pruned\_qf$ **do**
  $xye[i,:] \leftarrow$ `vector([`$q.x, q.y, e$`])`
end
$range \leftarrow xye[:,2]$`.max-`$xye[:,2]$`.min`
if $range \neq 0$ **then**
  $xye[:,2] \leftarrow (xye[:,2] - xye[:,2]$`.min)/`$range$
end
$corr\_matrix \leftarrow$ `euclidean_dist(`$xye[:,0:2]$`, `$xye[:,0:2]$`, ignore_self=True)`
$closest\_corr\_qubits \leftarrow corr\_matrix$`.min(axis="cols")`
$correlation\_factor \leftarrow$ `avg(`$closest\_corr\_qubits$`)`
if $correlation\_factor > 2 * device\_avg\_min\_dist$ **then**
  return `NULL`
end
$xye[:,2] \leftarrow$ `power(`$xye[:,2]$`, 2)`
$x \leftarrow$ `avg(`$xye[:,0]$`, weights=`$xye[:,2]$`)`
$y \leftarrow$ `avg(`$xye[:,1]$`, weights=`$xye[:,2]$`)`
$d\_vector \leftarrow$ `euclidean_dist(`$xye[:,0:2]$`, vector([`x,y`]))`
$radius \leftarrow 2*$ `avg(`d_vector`, weights=`$xye[:,2]$`)`
return `((`x`, `y`), `$radius$`)`

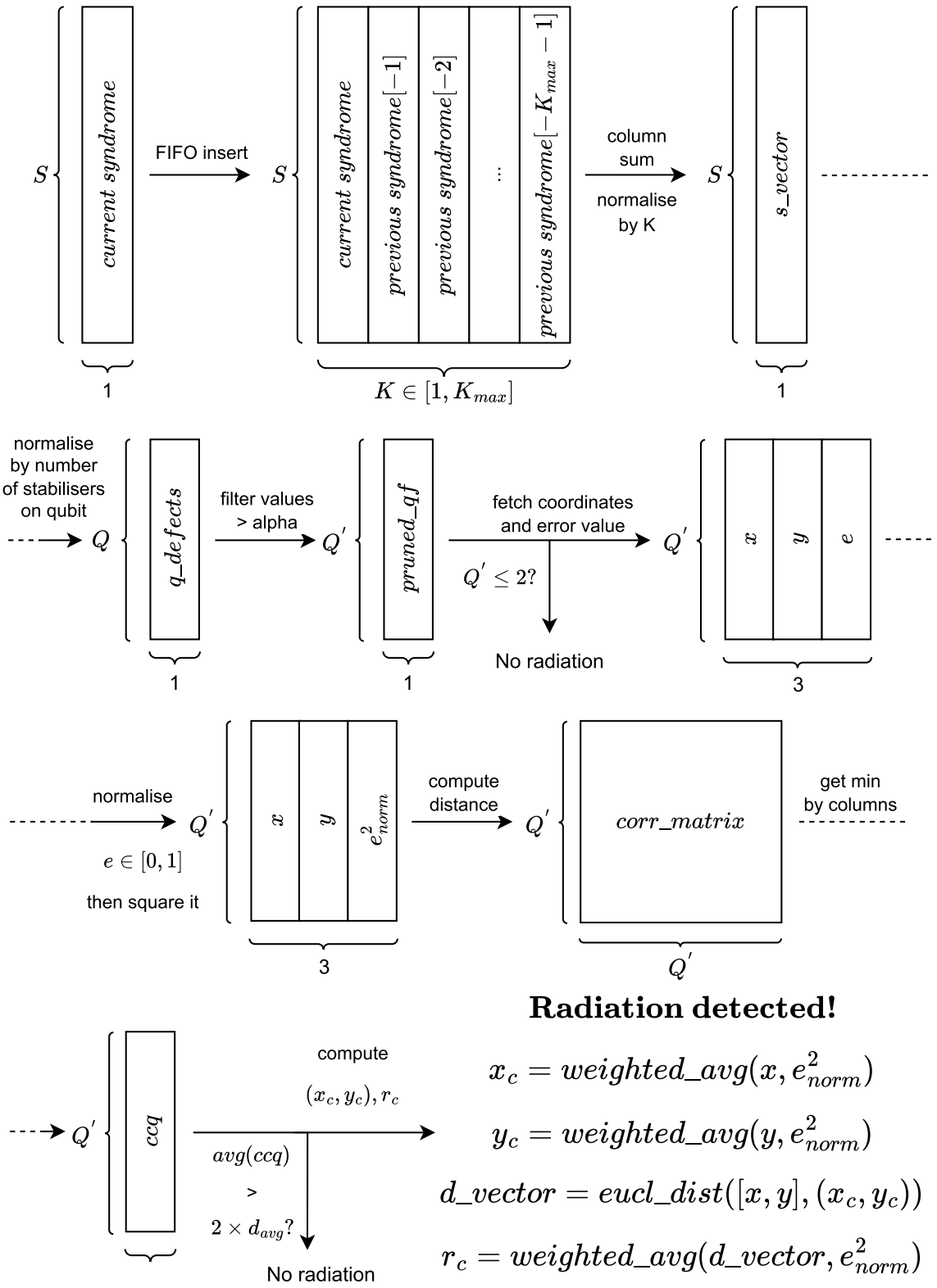


Figure 4.2: REI algorithm information processing diagram.

considered uncorrelated and the subroutine returns without detecting a radiation event. Otherwise, the physical x, y coordinates of each physical qubit, the coordinates of the locus of the high energy event are computed as the average over the x and the y coordinates of all qubits, weighted by the error column $xye[:, 2]$ which had been previously elevated to the power of two. The radius of the high energy event is computed as twice the average weighted Euclidean distance between each qubit and the newfound coordinates of the radiation error locus.

Observation IV.II

It is possible to identify a circular threshold that encloses radiation's area-of-effect, which shrinks over time. All equidistant qubits experience on average the same radiation-induced fault intensity.

4.5 Results

To show the efficacy and efficiency of the REI subroutine and its applications, I devised a set of five separate analyses, considering variable code distances for the Rotated Surface code, different loci for radiation faults, multiple Tanner graph based decoder implementations, and configurations of the mapped surface codes on the quantum chip's architecture.

4.5.1 Code distance impact on detected area-of-effect

The subject of this analysis is to compare the REI subroutine efficacy on the Rotated Surface code against the same simulated radiation event at different code distances. To keep the analysis as interesting and relevant as possible, I have selected the centre of the quantum chip as the locus of radiation, which is the worst case scenario for such an event. Each radiation event begins at time zero, with an overall duration of $1\ ms$, meaning that the intensity of the fault after that time is strictly lower than the intrinsic noise rate of the quantum device. The Rotated Surface code is considered for both the X-basis and the Z-basis, at code distances varying from 5 to 19. Statistics are drawn over 512 independent syndrome sequences over time.

Figure 4.3 compares the REI subroutine's outputs on both basis correction passes of the Rotated Surface code, on the first and the second row respectively. In the first column, one can notice how the radiation detection rate, that is the incidence by which the REI subroutine successfully spots a radiation event, suffers from no false positives outside the time window of the fault. Moreover, the identification of the fault is triggered sharply as

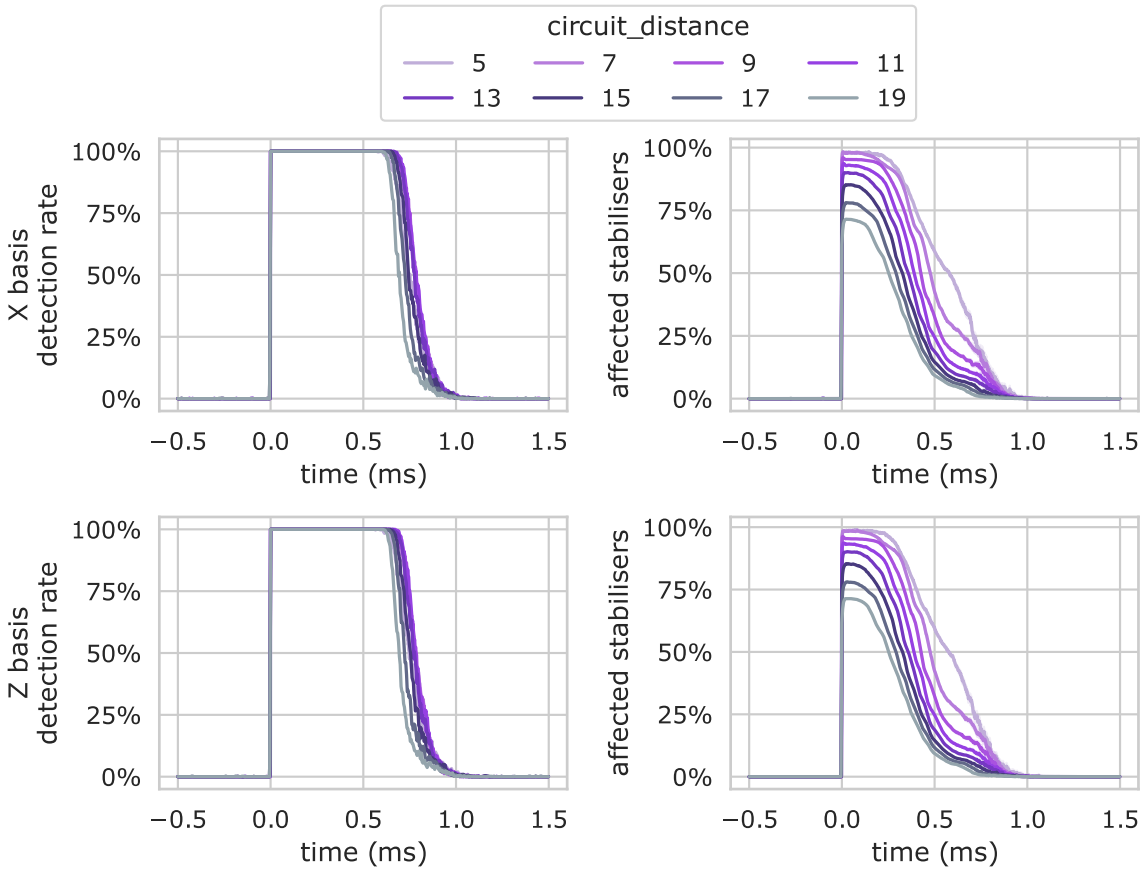


Figure 4.3: Code distance relation with radiation area-of-effect detection.

soon as it begins, regardless of the code distance and the basis considered. Notably, the tail of the radiation fault, which is exponentially less intense at the end than at the beginning of the event, stops being detected slightly earlier in higher distance codes.

Observation IV.III

Radiation event identification is code and error-basis agnostic, and can be identified at runtime and separated from regular intrinsic noise.

In the two rightmost plots of Figure 4.3, I have considered the ratio of stabilisers affected by the radiation event over the total number of stabilisers in the surface code. Given the properties of radiation faults described in Section 4.3.3, one would expect the affected area to be circle-shaped and centred on the injection point. In all the considered experiments, all code configurations are subject to a simulated radiation fault of equal position and intensity, thus inducing comparable error rates in the same qubits across simulations. Notably, the ratio of corrupted stabilisers is inversely proportional to the code distance, as they are more closely compacted around the fault's source. In fact, codes which use a lower number of

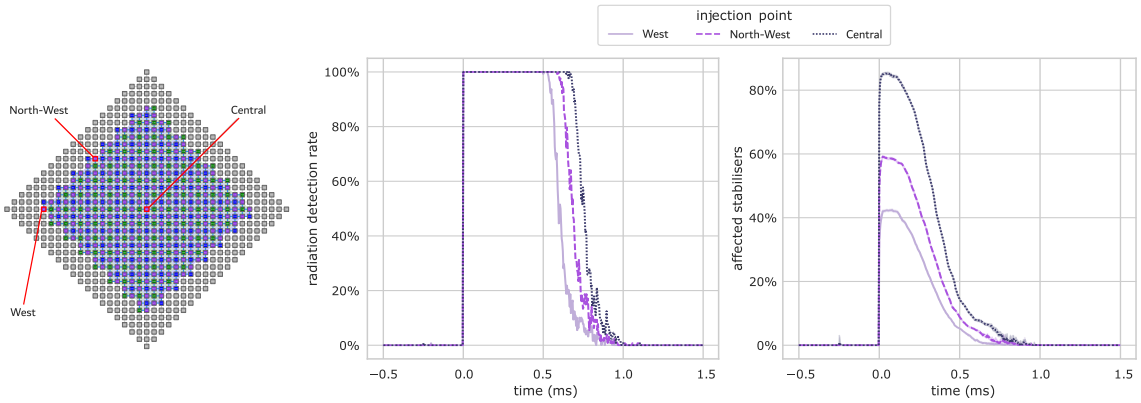


Figure 4.4: Detection of the area-of-effect of radiation faults.

qubits will inevitably incur a higher ratio of failing stabilisers, as the area of effect of radiation is the result of a physical phenomenon, unaffected by the choice of QEC code. Inversely, using a larger number of resources will in turn reduce the percentage of affected stabilisers, as the QEC code will have more stabilisers "to spare". Once again, the choice of the Rotated Surface code's basis is transparent to the measured metric.

Observation IV.IV

Larger distance Rotated Surface codes have a higher chance of preserving information from radiation events, as a lower ratio of stabiliser qubits is corrupted.

4.5.2 Effect of radiation fault position

In this analysis, I have characterised the detection capabilities of the REI subroutine. This includes testing both the incidence of detection of a high energy event and the ratio of stabiliser qubits affected by the event.

In Figure 4.4, the radiation detection efficacy on the Rotated Surface code of distance 15, drawing statistics for three separate radiation fault loci, highlighted in red in the left subplot. These injection points are labelled as *Central*, *North-West* and *West*, as outlined in the three different locations of the left subplot. The detection rate of radiation events is shown in the centre, while the estimated ratio of stabilisers affected over time by the radiation event is on the right. Statistics are drawn over 128 samples. For all of them, the radiation event is injected at time 0 *ms* and lasts for 1 *ms*. The aim is to highlight that the REI's performance is not affected by the fault location, as it can still accurately identify the fault even if corner qubits are affected.

The middle subplot represents the evolution over time of the incidence of detection of a radiation event. The REI subroutine immediately spots the presence of a radiation

event, with a detection rate that sharply falls off towards the end of the event at $1\ ms$. The North-West injection point stops being detected about $50\ \mu s$ earlier than the Central one, and the West injection point stops being detected about $100\ \mu s$ before the Central one. From this, it follows that the detection rate of the Central event, which affects a larger portion of qubits, is detected as active for a longer period of time.

In the rightmost plot the position of the radiation locus is correlated with the ratio of affected stabiliser qubits. Since the intensity of the radiation event over the quantum chip's surface area dampens with inverse square proportionality, the further a locus of radiation is from the centre of the QEC code, the lower number of qubits, and thus stabilisers, are going to be affected. One can notice this trend as the Central injection point retains the largest number of affected stabiliser qubits over time, whilst the West injection point retains the fewest. Given that every location on the quantum chip is equally likely to be hit, one can state that the rate of affected stabiliser qubits is upper bounded in the worst case by a radiation event hitting the centre of the surface code. However, a lower bound can not be identified, as the high energy event may impact a set of coordinates that lie outside the area belonging to the surface code. In this latter case, some of the stabiliser qubits might still be affected.

Observation IV.V

The temporal persistence of a radiation event is correlated to the affected area in the surface code. Peripheral faults induce transients that last for less time than those at the code's centre.

4.5.3 Radiation detection complexity and time performance

The scope of this analysis is to understand the impact of the radiation detection routine in the context of the constraints associated to QEC decoding in modern quantum computers. Most importantly, I wanted to prove that the REI subroutine is efficient enough in terms of time to solution to be possibly integrated in existing decoders, as later shown in Section 4.5.5. For this task, I once again consider the Rotated Surface code in the Z basis, at distances ranging from 5 to 19.

Figure 4.5 reports the average overhead ratio on the Z-basis pass of the Rotated Surface code for growing code distance, with black bars representing the standard deviation. This is done with an optimised C implementation of the REI subroutine, compared with the time to solution of the Minimum Weight Perfect Matching decoder, which is the fastest in terms of time to solution between the decoders reported in Section 4.2.2. For the sake of space, I

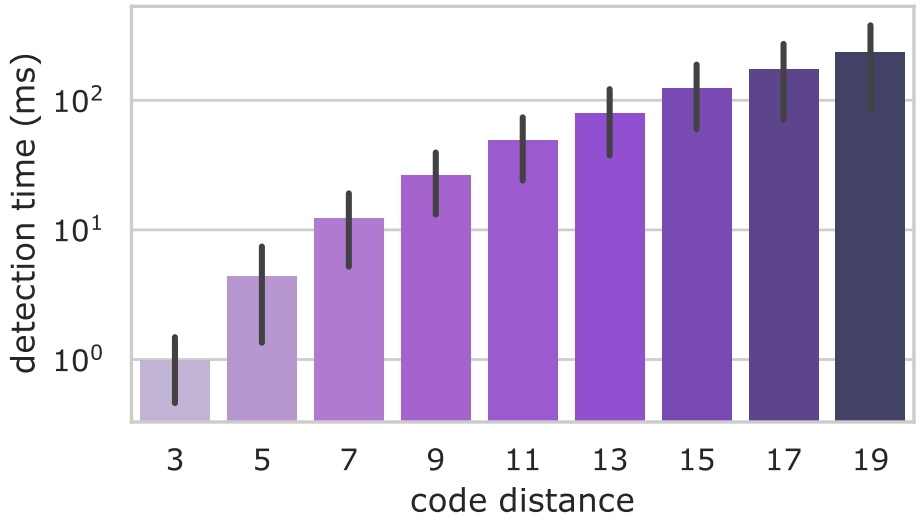


Figure 4.5: Overhead ratio of REI over the MWPM decoder.

did not report the relative overhead with respect to the other Tanner graph based decoders. In both the REI subroutine and the MWPM decoder cases, the input is a set of random stabiliser syndromes. The average overhead induced by the subroutine ranges between 0.1% for the distance three Rotated Surface code, to no more than 0.2% for the same code at distance 19, despite the quadratic increase in the number of stabiliser qubits. The REI subroutine requires from three to two orders of magnitude less time than the MWPM decoder to provide an output, thanks to early exit conditions that trigger in absence of a radiation event. It is thus reasonable to assume the insertion of the REI subroutine as a component of future QEC decoders.

Observation IV.VI

Without the need for vectorisation, the sequential implementation of the REI subroutine introduces minimal overhead with respect to the decoding step.

Most of the complexity of the REI subroutine stems from the computation of the Euclidean distance between two vectors of coordinates of size n, which add up to a complexity of $O(n^2)$ in the number of comparisons, whilst the remainder of the operations have complexity $O(n)$, where in the worst case $n = (d^2 - 1)$, with d being the distance of the surface code. This scales favourably with respect to the $O(d^6 log(d))$ complexity of the MWPM decoder for a code of distance d [109]. Since the highest complexity operations of the REI subroutine are easily vectorisable and parallelisable, the time to solution would reach similar acceleration as that of hardware based decoders in use in real-world quantum computers.

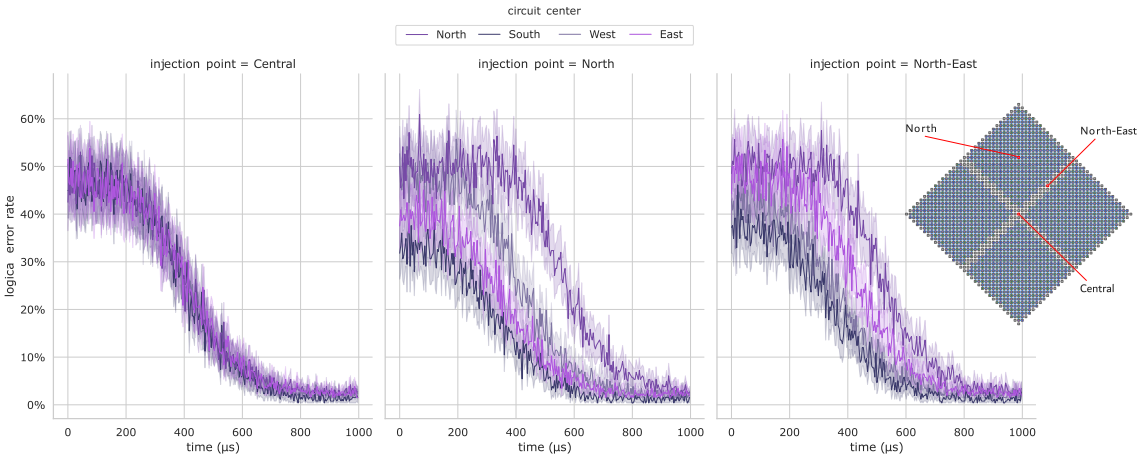


Figure 4.6: Multi-code logical error evolution.

4.5.4 Multi-code logical error correlation

Recent years have seen a surge in the number of physical qubits embedded in a single superconducting chip. There is reason to believe that this trend will continue, and that multiple QEC codes may be placed onto the same quantum chip. In this analysis, I have extrapolated the logical error rates of a future large-scale superconducting quantum chip. To do so I have considered an expanded chip structure, with the same repeating pattern as the one presented in Figure 4.1, hosting four distance 15 Rotated Surface codes, placed respectively in North, South, East and West quadrants obtained by cutting the quantum chip with two orthogonal diagonal lines. I have reported the performance of the Minimum Weight Perfect Matching decoder with respect to three independent loci for the radiation fault in separate simulations. Similarly to what had been presented in Section 4.5.2, three fault positions are identified: the *Central* fault position, equidistant to all surface codes, one fault position at the halfway *North-East* point between two surface codes, and one fault position centred onto the *North* surface code.

Figure 4.6 presents the post-decoding logical error rate of the four surface codes as a function of time, considering a radiation fault lasting for $1\ ms$, drawing statistics over 256 shots.

Observation IV.VII

Surface codes which are equidistant from a radiation event will experience correlated logical error spikes of similar intensity.

In the leftmost subplot, once can see how all surface codes are equally affected by the Central radiation fault, with logical rates that closely match one another. This behaviour is

expected, since they are all equidistant from the locus of radiation.

Conversely, when considering the North injection point in the middle subplot, the North surface code is affected to a higher degree by the radiation fault, reaching a peak of more than 53% logical error rate for more than half of the total fault's duration, whilst the second closest is the West surface code, with a peak at about 45%, closely followed by the East code with a similar peak at 41%, while the furthest surface code, in the South quadrant reaches a lower peak of 34%. Notably, the overall persistence of the peak is longer for the North code, which is affected the most, while in further away codes the logical error converges to zero faster. In the leftmost plot, when considering the North-East locus of radiation, both the North and the East codes reach a peak logic error rate of about 50%, while the West and South code reach lower peaks of about 40%. Physical distance from the impact point is thus insufficient to prevent multi-code logical errors. One can thus expect bundles of proximal logical qubits to be subject to correlated logical error spikes.

Observation IV.VIII

Radiation events reaching far away surface codes can still overcome their error correction threshold, although to a lesser degree than closer events.

4.5.5 Radiation aware decoding

I conjecture that, by using information from the REI subroutine (fault detection, fault location identification, and area affected by radiation), one can partly mitigate the effects of radiation on the Rotated Surface code during the decoding phase. Intuitively, since the stabilisers are physically interleaved with data qubits, and radiation spreads in space, I expect that if a group of stabilisers has been corrupted, the data qubits inside that area have been affected by radiation as well. Once the area of effect of the radiation event has been identified, a bitwise inversion is applied to all the stabiliser measurements in the current syndrome that were hosted on physical qubits inside the affected area, mapping *true* values to *false*, and vice versa. If a stabiliser measurement belongs to a qubit outside the radiation fault, it is left unmodified. This processed syndrome is then fed to the Minimum Weight Perfect Matching decoder, which will provide an output prediction for the state of the QEC code, to be compared with the rest of the decoders, thus labelling it the *RadMatching* decoder.

To measure if RadMatching can be effectively used to compensate for radiation-induced events, the logical error rate of the Rotated Surface code is compared with the Tanner graph based decoders presented in Section 4.2.2: MWPM, BM, Belief Find (BF) and UF. This is done considering a single radiation fault, lasting for $1\ ms$, injected at the centre of the quantum

chip, where radiation's effects are most detrimental, and a Rotated Surface code of distance 9 with 9 repetitions per round of correction. The performance of all decoders is compared onto the same temporal sequence of syndrome measurements. Statistics are drawn over 384 shots.

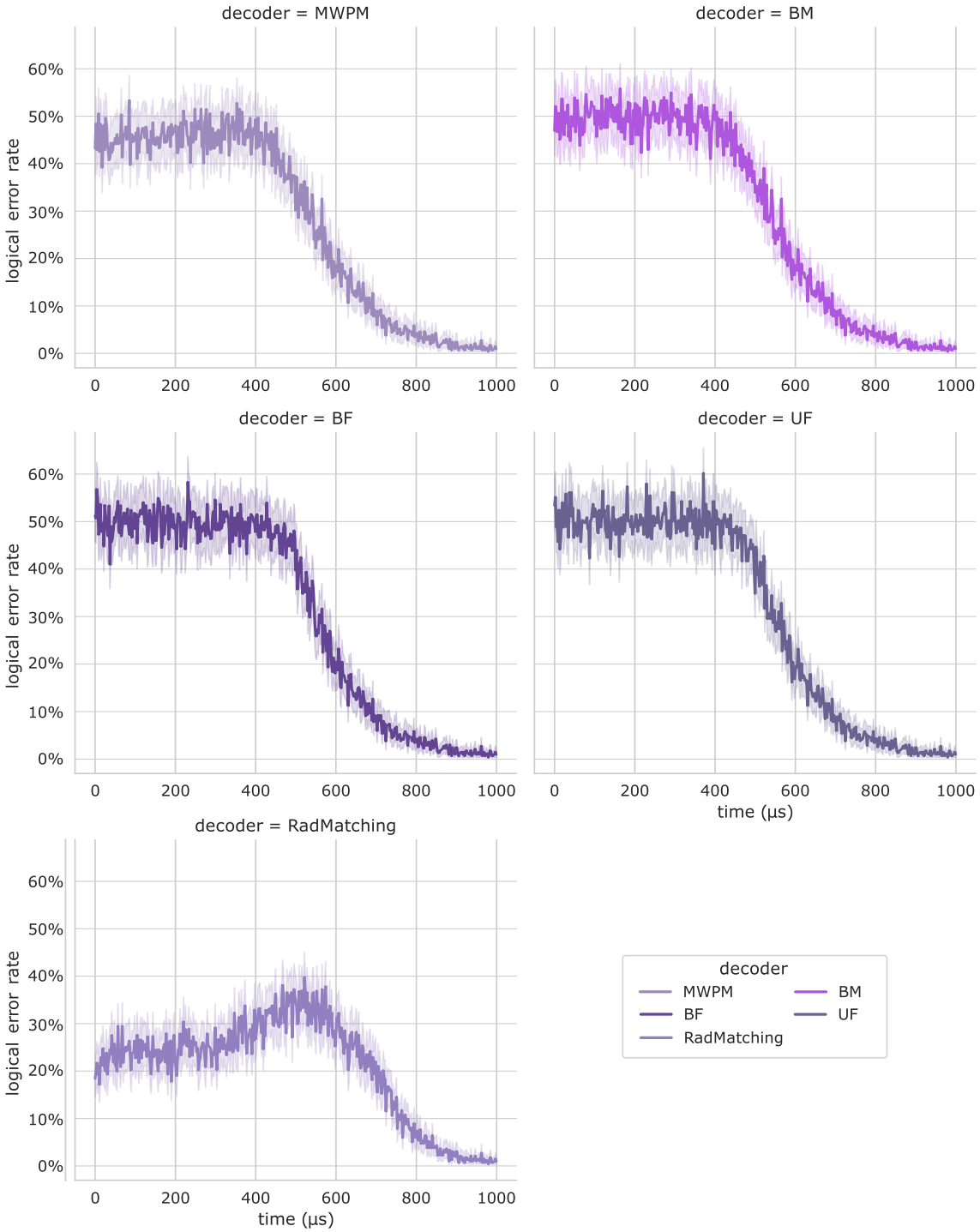


Figure 4.7: Decoder performance comparison.

Figure 4.7, presents the logical error of the Rotated Surface code over the time window of the radiation fault. Decoders are ordered from best to worst going left to right. The performance of the RadMatching implementation is shown on the leftmost subplot, using the REI subroutine together with the MWPM decoder. The presented approach improves onto the logical error of the non radiation-aware MWPM decoder by upwards of 25% at the beginning of the radiation event, when the fault's intensity is maximal.

Observation IV.IX

Identification of radiation's area-of-effect provides vital information to the RadMatching decoder, lowering the overall logical error rate of the Rotated Surface code.

All the other non-radiation aware graph based decoders considered in this analysis show marginally worse performance than the MWPM decoder, with the BM decoder reaching a peak of 50%, the BF decoder reaching a peak of 52% and the UF decoder reaching again a peak of 50%. All the four non radiation-aware decoders maintain their respective logical error peak from more than half of the total duration of the fault. On the other hand, the RadMatching decoder shows a relatively stable logical error rate of about 25% in the first third of the radiation event's duration, albeit reaching a logical error of about 35% at the halfway point, which is slightly higher than that at the beginning of the time window. This slight performance degradation (which still shows an improvement over existing decoders) is due to the lower radiation detection rate after the halfway point, since the intensity of the radiation fault gets smaller over time, as shown in Section 4.3. In the last portion of the time evolution, all the considered decoders converge to their respective nominal *radiation-free* logical error rates.

Observation IV.X

Tanner-graph based decoders which are not radiation aware can not keep up with the error rates imposed by radiation events.

4.6 Chapter summary

In this chapter, I have discussed how to model the impact of a high energy event onto a superconducting quantum chip, to later run simulations on multiple configurations of the Rotated Surface code.

Reaching back to the introductory research questions of this chapter, I have tackled **RQ1** by introducing the novel REI subroutine, which makes it possible to identify the incidence and the area-of-effect of radiation events at runtime, with a time to solution which is just

a fraction of that of decoding. Important observations relative to **RQ2** have also been underlined, stating that larger surface codes are inherently more likely to preserve stabiliser information in the event of radiation faults, and that the most detrimental location for a fault to happen is right at the centre of the surface code's area, with longer lasting transient effects with respect to peripheral positions. Moreover, I have deepened my answer to **RQ2** by considering the post-decoding logical error rates of a set of Rotated Surface codes on a shared quantum chip, noticing how their logical error rate is correlated to their respective distance from the impact point, hinting at how radiation-induced faults might propagate across logical qubits. Furthermore, I have compared a set of Tanner-graph based decoders, highlighting that standard decoding techniques are not sufficient for dealing with radiation events. At last, **RQ3** has been assessed by measuring the performance of the RadMatching decoder, a MWPM decoder made radiation-aware by the REI subroutine, showing an average reduction of the logical error rate by about 25% with respect to other approaches.

This chapter stresses the urgency of finding new and alternative solutions to the issue imposed by high energy events in superconducting quantum computers. Despite the promising results, there is still plenty of research to be done, regarding extensions and generalisations of radiation-aware decoding techniques, the verification of new QEC code classes, and the analysis of extended prototypes for quantum memories.

Chapter 5

Cross-layer hardening of qubits

The struggle of the hour in quantum computing research is achieving effective suppression of the error mechanisms induced by the interaction of external radiation with superconducting quantum devices. Despite the rapid advancements in QEC of recent years, radiation-induced faults are yet to be fully addressed. These events are known to be the cause of simultaneous correlated defects in qubits that lie onto a single substrate, ultimately annulling QEC codes effectiveness.

5.1 Objectives

In this chapter, I aim at providing an effective and affordable radiation-induced fault suppression by exploiting cross-layer hardening. This is done by *selectively* combining substrate-level phonon barriers and QEC interleaving via a planar-mesh tiling algorithm, TETRIS-Q, reaching efficient and effective suppression of radiation events. Substrate phonon-barriers [130, 152] enclose one or multiple qubits into tiles, thus limiting spatially correlated faults. The proposed cross-layer solution comes at no extra cost in terms of QEC code execution or decoding time. The objective is to merge the qualities of QEC and phonon-barriers, all while reducing the overall cost of implementing qubit hardening solutions. The investigation is guided by these research questions:

This chapter refers to the contents of the article "TETRIS-Q: Tiling-based Effective Transient-fault Reduction on Interleaved Superconducting Qubits", written by M. Vallero et al. and currently under review [205].

- **RQ1**: To what extent do phonon barriers improve QEC codes' tolerance to radiation events?
- **RQ2**: What is the optimal phonon barrier size and position to preserve QEC operativity?
- **RQ3**: Can phonon barriers improve the error threshold of QEC codes?
- **RQ4**: Does interleaving multiple independent QEC codes improve their reliability to radiation events?
- **RQ5**: What reliability improvements can be gained by using both barriers and QEC interleaving?

Following real-world implementations, barriers are characterised by a permeability quality factor, which I took as input for my simulations, only partly limiting the dispersion of radiation-induced quasiparticles to well-defined portions of the quantum chip. The theory supporting phonon-barriers is that, by disrupting the spatially correlated nature of radiation faults, the resilience of current QEC codes will be considerably boosted, without incurring in any additional runtime or decoding overhead.

With ever-increasing on-device qubit counts, more and more independent logical qubits are being embedded in independent QEC codes on a single chip. I thus also leveraged the TETRIS-Q tiling algorithm to interleave multiple separate QEC codes. The intent is to increase the physical distance between virtually-close qubits in a QEC code whilst maintaining the same total number of embedded logical qubits.

By merging phonon barriers placement and QEC interleaving, I managed to reduce the impact of simulated radiation-induced faults below the intrinsic noise floor of the superconducting quantum computer. Besides, I have optimised for the implementation cost of the phonon-barriers in order to reach the desired level of reliability without incurring in diminishing returns in effectiveness.

I have modelled and simulated radiation-induced transient faults over a plethora of barrier and QEC interleaving configurations. Through more than *51 million* radiation fault injections, I show peak logical error reductions of more than 99.8%, together with an 80% reduction of the observable transient duration with permeable barriers. I found that sparser tiling can reach comparable performance to *single qubit* tiling, prompting cost reductions of upwards of 87% in barrier tracing. By leveraging independent QEC code interleaving, I have measured up to one order of magnitude average logical error rate reductions without

the use of permeable barriers, and up to three orders of magnitude with the joint usage of barriers.

The chapter makes direct reference to the concept of radiation events in superconducting quantum computers, as described in Chapter 1, Section 1.1.7. The structure of the chapter is as follows. Section 5.2 introduces the QEC classes object of the later analyses, and details hardware hardening approaches. Following this, Section 5.3 presents the models used for the quantum computer, intrinsic noise and radiation-induced faults, then discuss the TETRIS-Q tiling strategy, the modelling of phonon barriers and the QEC interleaving approach. The main findings of the chapter are presented in Section 5.5, with final considerations along a summary in Section 5.6.

5.2 Background

This section provides a description of both radiation events and QEC to acquaint the reader with the concepts of the later sections.

5.2.1 Quantum Error Correction

This study focused its efforts on the currently most widely adopted class of QEC codes, the rotated surface code [98]. The main advantage of this code class lies in its planar connectivity requirement amongst qubits of degree ≤ 4, which makes them easily implementable in current day quantum hardware. This gives also rise to a defect decomposition that gives rise to graph-like errors, which can be easily decoded with efficient and fast decoders, such as minimum weight perfect matching [109, 110]. In the context of the considered QEC codes, a syndrome extraction round lasts around a few microseconds or less, as such radiation events are expected to hinder multiple subsequent rounds of correction [21].

5.2.2 Radiation hardening methods

Preliminary radiation-hardening solutions have been proposed. Gap-engineering, an hardware-level tuning of the energy threshold required to cross the Josephson junctions, which is designed to regulate intrinsic device noise, has shown promising results in phonon-mediated cross talk reduction [206–209]. There is, however, limited evidence of its effectiveness over the whole spectrum of radiation events [101, 149, 151, 210]. Placing quantum computers in underground facilities limits the external radiation flux reaching the device, although scaling

this approach is impractical [129, 135, 142, 144]. The decoupling of superconducting components from the substrate has shown a notable reduction of the incidence of radiation events, but this also imposes serious manufacturing quality variance and scaleability challenges [148]. The usage of geometrical trenches, insulating layers, downconversion structures or phonon barriers limits the generation and diffusion of quasiparticles in superconducting quantum computers [130, 151–156], all while increasing manufacturing costs on a *per qubit* basis, which makes them not cost-effective.

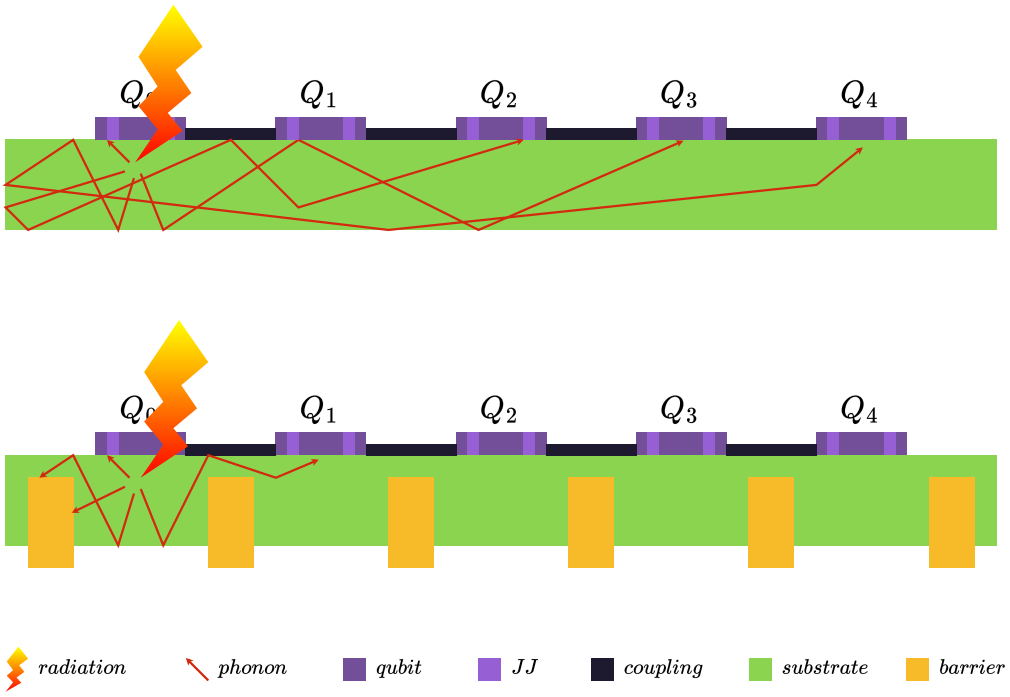


Figure 5.1: Generalised effect of substrate barriers.

5.2.3 Main contribution

The concept of substrate-level phonon barrier used in this paper refers to a hardware-level solution intent at limiting the spreading of radiation byproducts across the substrate of a quantum chip. From a higher abstraction point of view, these barriers reduce the incidence of simultaneous correlated faults at the qubit level, as exemplified in Figure 5.1. I did not focus on any specific physical implementation for such barriers, as multiple alternatives are currently being explored in the field [152, 211], but rather I looked into the higher-abstraction implications of *where* to put barriers, *how many* qubits should they enclose, and *how* to optimise their usage. The aim was to provide both cost-efficient and effective radiation suppression via hardware-software co-design.

Observation V.I

Substrate barriers limit radiation-deposited energy propagation, aiming to reduce the incidence of spatially correlated radiation-induced faults.

Any barrier is characterised by its ability to prevent the passage of quasiparticles, namely its *permeability*. The phonon flux attenuation provided by these barrier technologies has been experimentally measured to be in the order of $2-100\times$ [152, 211] with aluminium strip barriers with widths ranging between $10-100\mu m$ [212, 213], hinting at an inverse correlation between a barrier's width w and its permeability, as $b_p \propto 1/w$. In the context of a linear multiplicative model, a barrier with permeability $b_p = 1$ will not reduce the quasiparticle flux, while a barrier with permeability $b_p = 0$ is an ideal barrier, allowing no energy leakage. Realistic engineering constraints limit the range of achievable barrier permeability quality factors as $b_p \in [0, 0.01]$, starting from more permeable trenches and metallisation stripes to less permeable multi-layered etchings. I considered the increasing cost of implementing a less permeable barrier with respect to the increase in its width, in the following simple model correlating barrier width in μm and the barrier permeability quality factor b_p that reduces the phonon flux.

$$w(b_p) = 10\sqrt{\frac{1-x}{x}} \tag{5.1}$$

The cost of implementing a barrier must then be multiplied by the total length of all the barriers that need to be manufactured to create a tiling of the whole quantum chip, as later explained in Section 5.4.2.

I analysed the effectiveness of substrate-level phonon barriers by enclosing one or multiple qubits in tiles, an exclusive set of qubits which are spatially close on the quantum chip's substrate. Quasiparticles can easily propagate inside a tile, while being less likely to diffuse across adjacent tiles. In this context, a size-1 tile corresponds to a contiguous barrier enclosing a single qubit. The subdivision of the qubits on the quantum chip in separate tiles of variable size is defined as a *tiling* of the topology of the quantum chip.

5.3 Setup and methodology

This section goes over intrinsic noise and radiation fault models used for the presented simulations. I have considered a generalised superconducting quantum computer model, with square lattice connectivity amongst a quantum chip with 1741 qubits. This qubit count is not a stringent requirement, as most of the simulations required the usage of just a portion of the whole quantum chip. All the simulation data presented in this chapter has been

computed on the distributed nodes of the Leonardo supercomputer, provided by CINECA, Italy. Additional information on this supercomputer had been previously provided in Chapter 2. The simulation library employed is STIM [54].

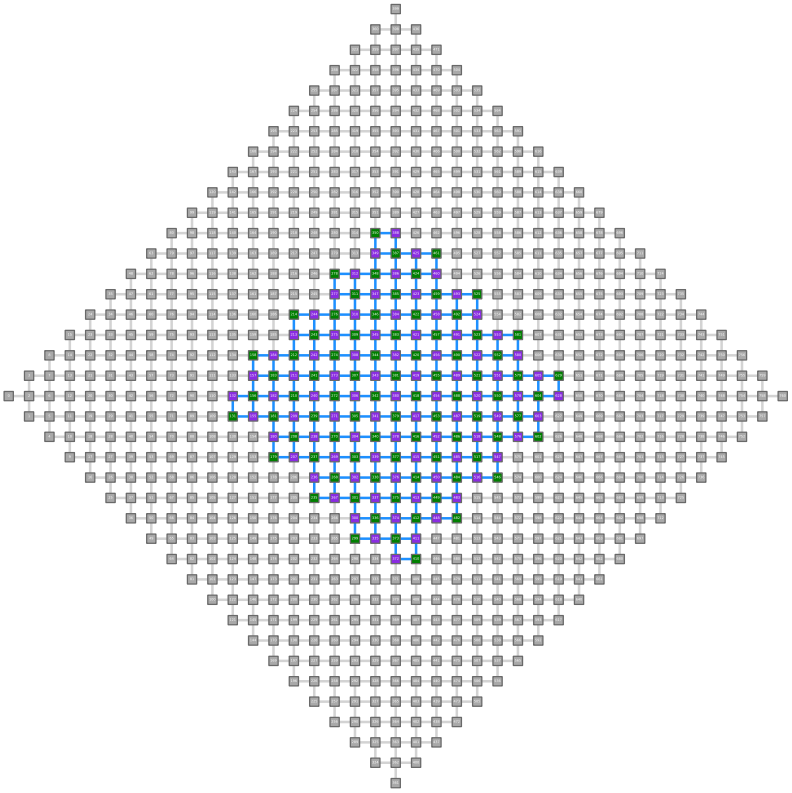


Figure 5.2: Quantum chip topology with an embedded rotated surface code.

5.3.1 Intrinsic noise model

In all the simulations, the intrinsic noise model considered was the standard *SI1000*, a commonplace representation of superconducting quantum computer inspired noise generally employed when testing QEC codes [86]. It operates by appending a randomised Pauli noise operator after each quantum gate in a quantum circuit, which is then triggered at the syndrome sampling step according to a given probability p. The intensity of intrinsic noise is thus parameterised by p, which can thus be easily swept over a continuous range of intensities. With respect to the original implementation, the intrinsic noise model used in this chapter has been extended to accommodate for all the gates employed in the considered QEC codes, but has been left otherwise unaltered.

5.3.2 Radiation fault model

Radiation events induce a spatio-temporally correlated reduction of the coherence time τ_1 of multiple qubits [24, 126]. The radiation fault model makes use of information from the properties of a modelled radiation event, the physical placement of qubits on the quantum chip, and the gate and coherence times of a generalised superconducting quantum computer [177, 187]. This model represents the effect of correlated radiation faults by injecting correlated Y-ERROR operators during the execution of a quantum circuit, which trigger according to a given probability.

The reduction in the characteristic τ_1 time of a qubit follows

$$\tau_{rad}(t) = \tau_1 e^{10((t-t_{rad})/\Delta t_{rad}-1)}, \tag{5.2}$$

taking into account the current time t, the beginning time of the radiation event t_{rad}, and the overall duration of the radiation event Δt_{rad}. The $\tau_{rad}(t)$ coherence time is used as an argument of

$$T(\Delta t_g, t) = 1 - e^{-\Delta t_g/\tau_{rad}(t)}, \tag{5.3}$$

that together with the elapsed time since the last gate on a qubit, governs the probability of that qubit to undergo a radiation-induced fault. The distance of a qubit Δs from the impact point of the radiation event, also called locus of radiation, introduces the dampening factor

$$S(\Delta s) = 1/(\Delta s + 1)^2, \tag{5.4}$$

that represents how energy dissipated in the quantum chip's substrate. As such, the probability for a qubit to undergo a radiation-induced fault follows Equation 5.5.

$$P(\Delta t_g, t, \Delta s) = T(\Delta t_g, t)S(\Delta s) \tag{5.5}$$

The radiation fault model automatically introduces these faults only during the Δt_{rad} time window.

5.4 Tiling, barriers and interleaving

This Section describes the effects of barriers in the radiation-induced fault simulation model, and goes into detail over the tiling and interleaving strategy employed in this paper.

5.4.1 Tiling strategy

The TETRIS-Q algorithm I used is a general algorithm working over a coordinate-aware planar square lattice graph. The algorithm takes as input a graph, a starting vertex v_s, usually at the centre of the graph, and a tile size t_{size}. At first, it identifies the shape of the *master tile*, which is composed of the largest complete square of vertexes, of side $s_{size} = \lfloor \sqrt{t_{size}} \rfloor$, plus the remaining vertexes $V_r = t_{size} - s_{size}^2$ wrapped around the upper-right side of the square. The first tile is placed onto the coordinates associated to the vertex v_s, then all other non-overlapping tiles are discovered via a breadth-first search. With this configuration, if multiple tiles are adjacent to one another, they share at least one side, and each tile contains the same number of vertexes, exception made for the vertexes on the outer portions of the graph. The algorithm produces a hypergraph, where each vertex is a contraction of a group of vertexes from the input graph.

5.4.2 Barriers

Barriers are thus placed onto the perimeters of the tiles computed by the TETRIS-Q algorithm, which can easily extend the tiling to any quantum chip topology. In the hypergraph, each vertex represents a tile, and the edges represent a shared barrier with a physically adjacent tile. An example of tiling pattern and hypergraph generation is presented in Figure 5.3. In the context of barriers t_{size} directly amounts to the total number of qubits inside each tile.

In the context of barriers, each tile introduces a barrier on its perimeter. This imposes one additional dampening factor on the probability of a qubit to experience a radiation-induced fault, as outlined in Equation 5.6, where Δl is the path length between the tile of a qubit and locus of radiation's tile, and b_p is the parameterisable barrier permeability. Notably, a $b_p = 1$ means that the barriers provide no damping.

$$B(\Delta l) = b_p^{\Delta l}, \qquad \Delta l \in \mathbb{N} \tag{5.6}$$

This let me extend Equation 5.5, adding the $B(\Delta t)$ damping parameter induced by the presence of tiles, giving rise to Equation 5.7, which has been used in this paper.

$$P(\Delta t_g, t, \Delta s, \Delta l) = T(\Delta t_g, t) S(\Delta s) B(\Delta l) \tag{5.7}$$

The fabrication costs of barriers scale linearly with the total perimeter of all the barriers that need to be put onto the quantum chip. To compute do this, one must thus resort to the barrier hypergraph, since it provides information about the adjacency of barriers. The

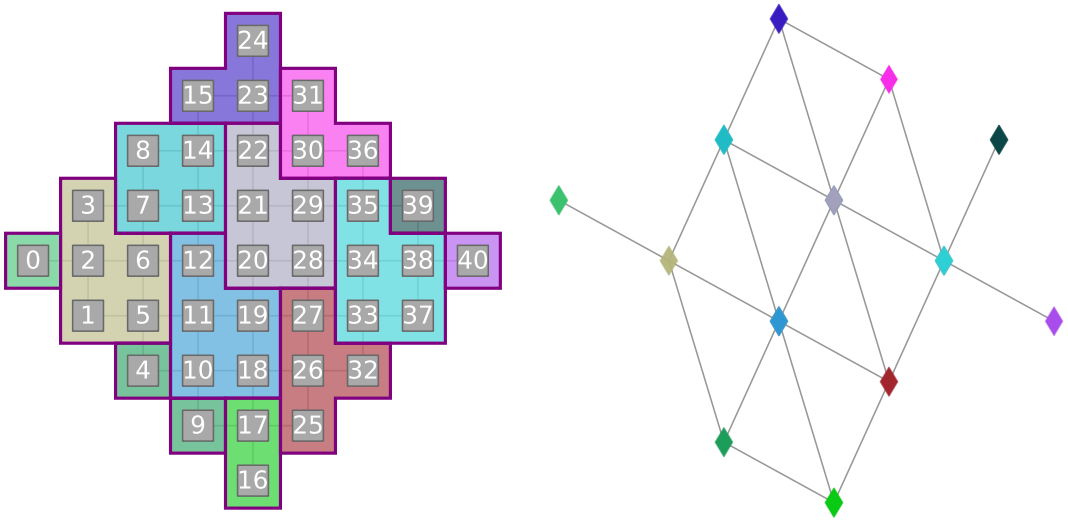


Figure 5.3: The quantum chip's topology (left) is used to create the barrier hypergraph (right) with a tiling pattern of dimension $tile_{size} = 5$, where each tile contains at most five qubits. The total phonon barrier perimeter is highlighted in purple.

total length P_{rs} is thus given by the summation of the perimeter of all barriers, minus all the overlapping portions of the perimeter of each tile in excess of one.

Observation V.II

The cost of a tiling depends on the size and shape of the tile, and the relative placement of the tiling with respect to the quantum chip's topology.

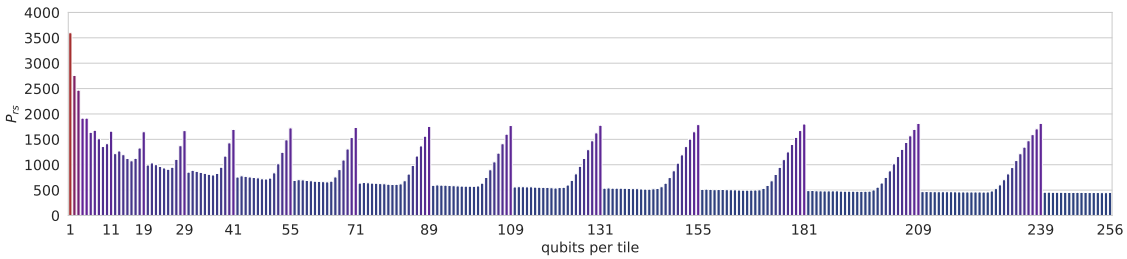


Figure 5.4: The P_{rs} barrier cost per tile size, onto a square lattice topology boasting 1741 qubits. Due to quantum chip topology boundaries, the tiling cost does not decrease monotonically. X-axis labels represent local maximum P_{rs} cost.

In Figure 5.4 one can observe the cost P_{rs}, i.e. the total length of the barrier traces in relative units, in function of the tile size, on the quantum computer topology, for a total of 1741 physical qubits. The maximum cost is obtained when each qubit has its separate tile, while the minimum cost is obtained for a tile size that encompasses all qubits. Besides,

the goal of tiling barriers is to simultaneously minimise both the P_{rs} cost and the expected logical error rate of QEC, as I will discuss in Section 5.5.2.

5.4.3 Interleaving

Quantum circuit interleaving is a software-hardware co-design approach that aims at finding suitable quantum circuits mappings over a grid of qubits. Given the rise of hardware-software co-design solutions for QEC codes, this can be seen as a further possible improvement to disentangle spatially close qubits from virtually correlated errors in QEC codes. This specialisation is deemed extremely important in the context of future fault-tolerant quantum computers, where multiple QEC codes, and thus logical qubits, may share the same purpose-built quantum chip. From a topological standpoint, only qubits directly involved in the operation of a specific QEC code will be interconnected, thus giving rise to coupling line overlaps to be managed via three-dimensional coupling lines over the substrate. The implications of this choice from an engineering standpoint will not be made subject of analysis in this paper. From an higher abstraction standpoint, interleaving independent QEC codes is the dual of the base case, which corresponds to relegate independent QEC codes in non-overlapping portions of the quantum chip.

The hypergraph can also be used to identify interleaving patterns to map multiple quantum circuits sharing the same dependency graph. The spatial sorting of vertexes, i.e. qubits, in each tile of the hypergraph is preserved. This makes it so that one can map a circuit onto the hypergraph derived from the quantum chip's topology, and then shift this mapping for multiple circuits amongst all the positions inside each tile. In a top-down view of the quantum chip, this amounts to interleaving multiple independent quantum circuits.

5.5 Results

This Section goes over the main findings of the chapter. The quantum computer has been modelled with a characteristic τ_1 time of $85\ \mu s$, a single-qubit/two-qubit/measure-reset gate duration of $25/32/58\ ns$. The SI1000 [86] intrinsic noise rate is set to $p = 10^{-5}$, unless otherwise noted. All the simulated radiation faults considered have been injected at the centre of the quantum chip, where the reach and intensity of the fault is maximal across the largest number of physical qubits. The duration of the radiation fault is set to $100\ ms$, following both results from the literature [23, 135, 137, 214, 215] and similar simulations of the same phenomena [177, 187]. The QEC codes considered have been decoded using the PyMatching MWPM decoder [109], as previous simulations from Chapter 4 using other

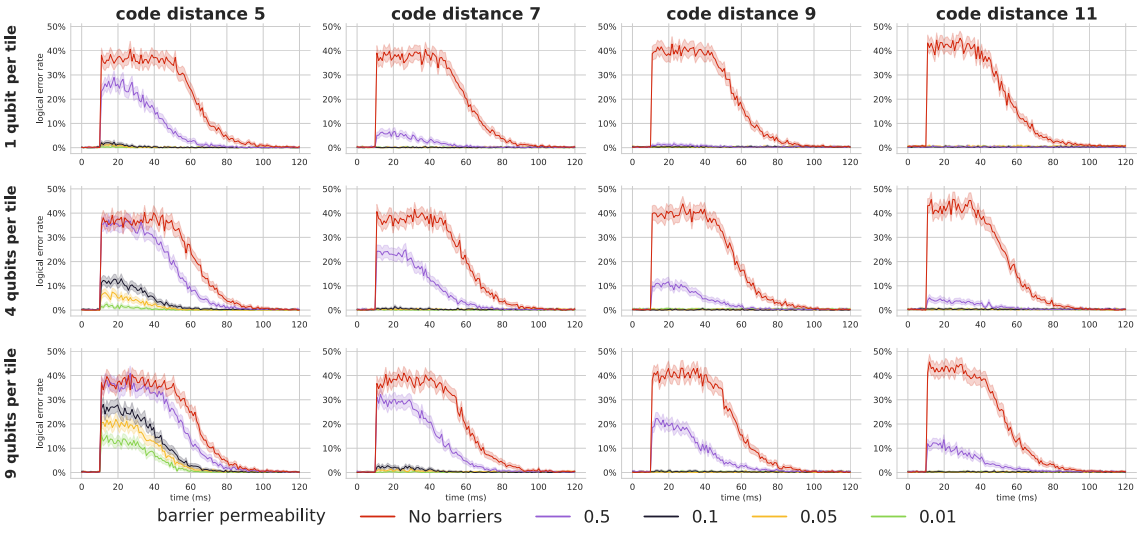


Figure 5.5: Rotated surface code logical error rates with 1, 4 and 9 qubit tiles (top to bottom), varying over increasing rotated surface code distance d (left to right) and barrier permeability b_p (plot hues). The case with no barriers acts as the base case, while barrier permeability costs follow Equation 5.1. The tiling with 4-qubit tiles and 9-qubit tiles incur in barrier tracing costs of 47% and 61%, respectively.

graph-based decoders showed no evident performance difference in the context of simulated radiation-induced faults.

5.5.1 Effect of barriers over time

This first analysis I considered a set of rotated surface code distances d, where $d \in \{5, 7, 9, 11\}$, and measure the post decoding logical error rate in the presence of an identical radiation fault injected at $t = 10\ ms$ and lasting for $100\ ms$. The tiling patterns considered in this analysis are square shaped containing 1, 4 and 9 qubits, respectively, whilst performing a sweep over the barrier permeability factors with $b_p \in \{1, 0.5, 0.1, 0.05, 0.01\}$. I measured 141 time steps and 1024 shots per time step.

In Figure 5.5, I considered increasing phonon barrier tile sizes growing from top to bottom and rotated surface code distances growing from left to right, whilst logical error rate hues represent the barrier permeability used for that configuration. In each configuration, I reported the base case for barrier permeability $b_p = 1$, equivalent to the base case with no barriers, characterised by a peak in the logical error rate at the beginning of the radiation event ($10\ ms$) at about 40%. This peak means that the QEC code is providing a performance slightly better than a random guess. The logical error rate peak of this base case presents an inflection point at about 40% of the fault's duration, signalling a rapid reduction in the radiation fault's intensity, and a second inflection point at 65% of the fault's duration, with

a more slowly fading tail until the end of the event. With the introduction of single qubit barriers with a barrier permeability factor $b_p = 0.1$, one can notice a considerably smaller peak in the logical error rate of about 3% for single qubit tiling at code distance-5, while the transient's duration dissipates just after 20% of the temporal duration of the fault. As the code distance increases, the effectiveness of barriers improves, with logical error rates not surpassing the 1% threshold at distance-11, with a barrier permeability $b_p = 0.5$, whilst the same performance in the distance-5 code can only be reached with barrier permeabilities $b_p = 0.01$ more than one order of magnitude smaller.

Observation V.III

Phonon barriers produce observable effects even at relatively high permeability, reducing the peak and duration of the effects of radiation-induced faults.

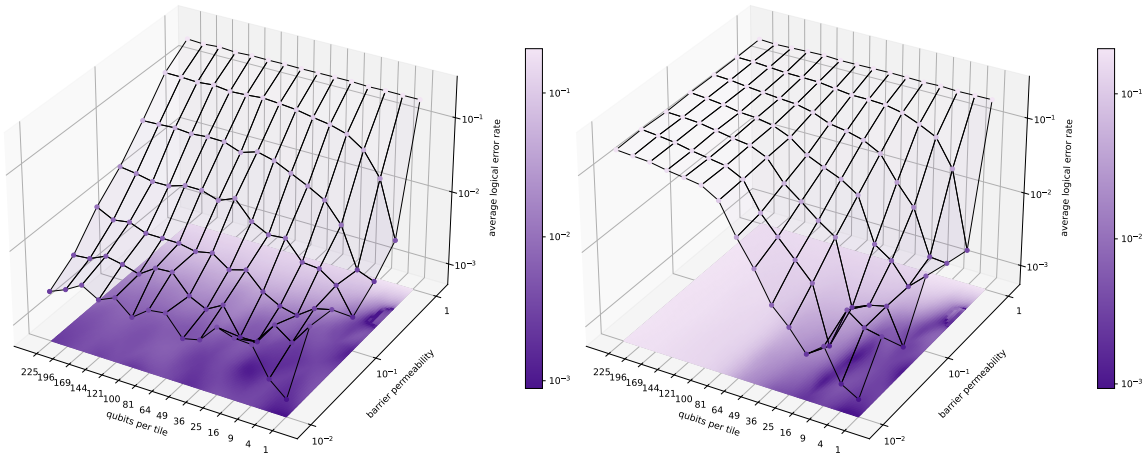
In the case of a larger tiling pattern with 4 and 9 qubits per tile, smaller barrier permeabilities are required to keep the logical error rate under the 1% threshold. This is noticeable in the comparable performance of the distance-5 surface code on single-qubit tiles, the distance-7 code on 4-qubit tiles and the distance-9 code with 9-qubit tiles, all at the same barrier permeability $b_p = 0.5$. This is due to the fact that by simultaneously increasing code distance and phonon barrier tile, each tile encloses a comparable percentage of all the qubits in the code. With decreasing barrier permeability and phonon tile size and increasing code distance, the logical error rate peak and the overall witnessed duration of the radiation faults shrink under the code's threshold. Notably, however, it is not necessary to do all those things at once to reach the desired radiation-induced fault response, prompting cost-saving strategies for what concerns phonon barrier tile sizes and permeabilities, and qubit requirements for code distances. I performed other similar full timescale simulations for other tile sizes, which have not been reported for the sake of brevity, as the effect of tile size is discussed in the following Subsection.

Observation V.IV

Lower barrier permeability and smaller phonon barrier tiles shunt the logical error rate peaks induced by radiation.

5.5.2 Effect of tiling size and position

This second analysis investigates the effects of the size of square phonon barrier tiles sweeping over the perfect squares $\{t_{size} \in \mathbb{N} : \sqrt{t_{size}} \in \mathbb{N}, t_{size} \leq 225\}$, in relation to the barrier permeability $b_p \in [1, 0.01]$ and the positioning of phonon barrier tiles. I considered



(a) Non-centred phonon barrier tiling.

(b) Centred phonon barrier tiling.

Figure 5.6: Distance-9 rotated surface code average logical error rate (Z-axis, vertical), varying over qubits per tile (X-axis, left) and barrier permeability b_p (Y-axis, right), differentiated by tiling positioning.

a distance-9 rotated surface code, and measure the average post-decoding logical error rate in the presence of an identical radiation fault. I measured 20 time steps and 1024 shots per time step.

In Figure 5.6, I considered the case of growing square along two edges only (a), and the case of growing square tiles from all edges simultaneously (b), in both cases by starting the tiling from the centre of the quantum chip. In both subfigures, I swept over the X-axis (left) with the number of qubits per tile, and over the Y-axis over the barrier permeability b_p (right), whilst representing the average logical error rate during a radiation-induced fault on the Z-axis. The barrier permeability has the most noticeable impact on the average logical error rate, with a stark correlation in both the non-centre (a) and centre (b) growing phonon barrier tiles. This is confirmed by the fact that, when the barrier permeability $b_p = 1$, the average logical error rate remains constant, regardless of the barrier size. The placement of phonon barrier tiles is especially important when larger tiles are taken into consideration. When the square tiles are centred with respect to the rotated surface code (b), the size of the phonon barrier tile show an improvement with respect to the baseline only if a single tile encloses at most 50% the surface code, otherwise no gain is recorded, regardless of the barrier permeability. In fact, the distance-9 rotated surface code employs 188 physical qubits, and for QEC-centred phonon barrier tiles holding more than 100 qubits no benefit is observed.

Observation V.V

Regardless of barrier permeability, centred phonon barrier tiles must enclose less than 50% of the QEC code's qubits to produce observable effects.

Non-centred phonon barrier tiling is considerably more effective, as with the growing size of the phonon barrier tiles, each tile holds at most about 25% of all the qubits in the rotated surface code. Intuitively, the lowest average logic error rate of 10^{-3} is reached with single qubit tiles and the lowest barrier permeability $b_p = 0.01$ in both cases. To reach an average logical error rate lower than 10^{-2} with a barrier permeability $b_p = 0.1$, it is sufficient to employ non-centred tiles containing at most 25 qubits, with a P_{rs} cost reduction of more than 75% with respect to tiles of size one. Centred tiles reach the same average logical error rate only with 4-qubit tiles at the same barrier permeability. This difference becomes even more evident with barrier permeability $b_p = 0.01$, whereby using non-centred tiling, an average logical error rate approaching 10^{-3} can be guaranteed with tiles of up to 225 qubits, prompting a cost reduction of more than 87%. As such, phonon barrier tile placement has a fundamental impact in the effectiveness of radiation-induced fault tolerance. Through further similar simulations, not reported in this manuscript for the sake of brevity, I conclude that the maximal effectiveness of a barrier is achieved for a tile sizes containing less than a quarter of the QEC code's total qubits.

Observation V.VI

Phonon barrier tiles containing up to 25% of a QEC code's qubits are comparatively effective to single-qubit phonon barrier tiles.

5.5.3 Effect of barrier permeability with respect to noise

In this third analysis, I correlate intrinsic noise and the average logical error rates over the whole duration of a radiation-induced fault. Specifically, I swept over the intrinsic noise model's probability $p \in [10^{-5}, 10^{-2}]$, for codes of distance $d \in \{5, 7, 9, 11\}$, and phonon barrier tiles sizes $tile_{size} \in \{1, 4, 9, 16\}$, to identify the rotated surface code's threshold when affected by radiation. I measured 20 time steps per configuration, with 1024 samples per time step.

In Figure 5.7 I identified the code thresholds in the considered combinations of barrier permeability and phonon barrier tile size. If a QEC's codes logical error rate threshold under the effects of radiation persists beyond the intrinsic noise rate, the QEC code becomes non-operational, accumulating more errors than can be corrected, as stated by observations [177]. In the base case with barrier permeability $b_p = 1$, corresponding to fully permeable barriers,

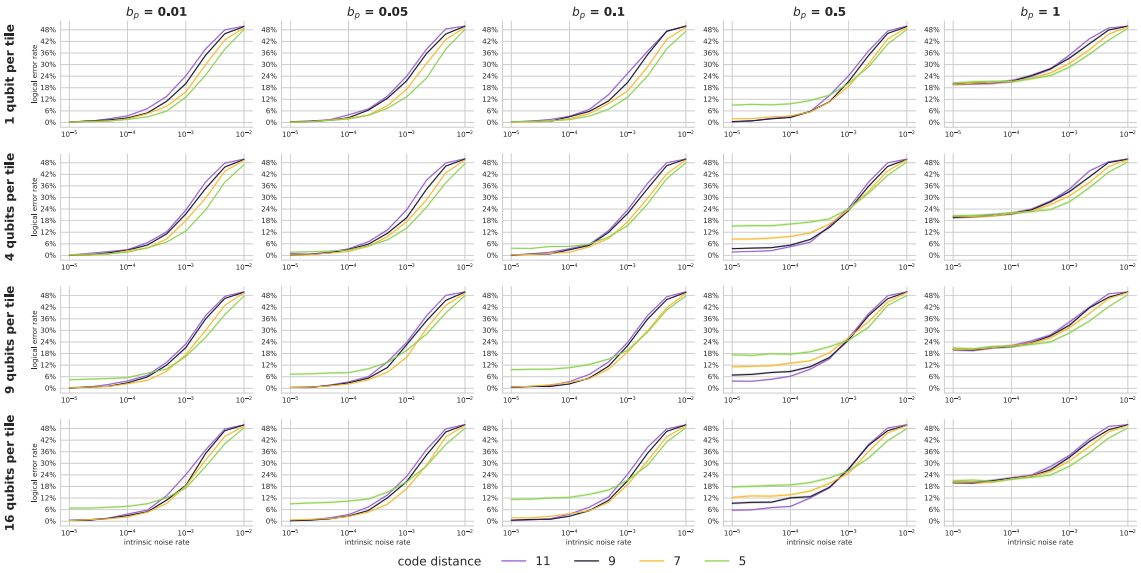


Figure 5.7: Average logical error rate of different rotated surface code distances with respect to intrinsic noise intensity, varying over barrier permeability b_p (increasing from left to right) and $tile_{size}$ (increasing from top to bottom), averaged over the total duration of identical radiation-induced faults.

the crossing points amongst the considered code distances is at 2.1×10^{-1} over an intrinsic noise intensity of 10^{-4}. By introducing barriers with permeability $b_p = 0.5$, the crossing point stabilises between 1.8×10^{-1} and 2.4×10^{-1} with a one-order of magnitude more intense intrinsic noise rate. Further reducing the barrier permeability correlates to lowering the code's threshold under the effect of radiation towards an error threshold comparable to the intrinsic noise intensity.

Observation V.VII

Phonon barrier tiling can preserve the characteristic efficiency threshold of QEC codes to intrinsic noise, guaranteeing radiation-fault tolerance.

Given the temporally bound nature of radiation-induced faults, QEC codes can bear a temporary increase in the logical error rate past the nominal error threshold. Although the best performing combination of barrier permeability and phonon barrier tiling size corresponds to single-qubit tiles with $b_p = 0.01$, the combination with the same barrier permeability and 4-qubits per tile performs comparatively well, with a 48% reduction in the implementation cost of the tiling. The distance-5 code responds less effectively to phonon barrier tiles, given its lower number of stabiliser qubits, and thus its crossing point with other code distances happens at a larger intrinsic noise rate. When considering only code distances $d \in \{7, 9, 11\}$, the crossing point reaches much better regimes, comparable with

the single-qubit tiles with $b_p = 0.01$, with 9-qubits per phonon barrier tile and a barrier permeability one order of magnitude larger, prompting a cost saving of more than 60% and guaranteeing operativity under the effects of radiation.

Observation V.VIII

A QEC code's distance limits the efficacy of phonon barriers with lower permeability, regardless of the intrinsic noise intensity.

5.5.4 Effect of barriers and interleaving

In the fourth analysis, I moved towards the concept of quantum circuit interleaving, observing the logical error rate over the duration of a radiation-induced fault of four independent distance-7 rotated surface codes. Specifically, I compared the base case without any interleaving, with the four interleaved QEC codes, over three different scenarios: a quantum chip without phonon barriers, a quantum chip with square tiles containing 16 qubits and a barrier permeability of 0.1, and a quantum chip with square tiles containing 4 qubits and the same barrier permeability. For each scenario, I recoded more than a thousand samples per time step, and 100 time step samples.

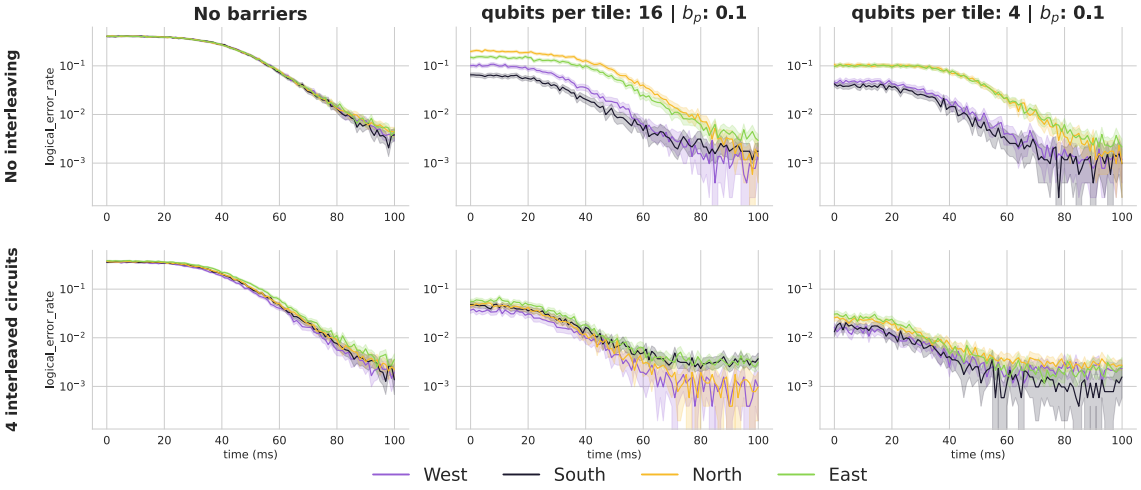


Figure 5.8: Logical error rate over time of four independent rotated surface codes subject to a radiation induced fault with respect the absence (top) or presence (bottom) of interleaving, varying over qubits per tile (from left to right), with a constant barrier permeability $b_p = 0.1$, where applicable.

In Figure 5.8, the top row contains the base cases with the four rotated surface codes placed side by side, whilst the bottom row are the cases with the four interleaved quantum circuits. When comparing the two cases with no barriers, by using interleaving, the logical

error rate peak sees a reduction of about 3%, but most importantly, the inflection point that signals the start of the dissipation of the radiation-induced fault appears 7% earlier in the total duration of the fault. This effect becomes even more evident with the introduction of phonon barriers.

Observation V.IX

Phonon barrier tiles and QEC interleaving show positive interference in mitigating radiation-induced faults.

With a tiling containing 16 qubits, the peak logical error rate gets reduced by more than 10% with the usage of interleaving, with an anticipation of the inflection point of up to 40% of the total duration of the radiation-induced fault. The third case, with a smaller tiling of 4 qubits, anticipates the position of the second logical error rate inflection point, after which the logical error rate reaches the logical error rate floor mandated by the intrinsic noise rate, by about 20% with respect to the 16-qubits tiling and of about 40% with respect to the 4 qubit tiling without interleaving.

5.5.5 Optimising barrier and interleaving cost

This last analysis measures the average logical error rate of one distance-7 rotated surface code over a sweep from 1 (base case) to 9 interleaved quantum circuits and $b_p = 0.1$ phonon barriers with tile sizes ranging from 1 to 16 qubits, plus the base case with no barriers. For each combination, I considered 1024 samples per time step, and 20 time step samples.

In Figure 5.9, the base case without phonon barriers presents the worst performance, acting as the comparison baseline. Following the columns of the topmost row from left to right indicates ever smaller tiling for the phonon barriers, and likewise a reduction in the average logical error rate of the rotated surface code, up to a minimum of 2.14×10^{-3} in the rightmost case of single-qubit tiles. Nonetheless, the efficacy of tiling does not scale linearly with the number of qubits contained therein, as the tiles which present a square or rectangular shape are more effective at preventing the spread of phonons. This can be noticed in the cases where the number of qubits per tile is either a perfect square, or the summation of the square of a perfect square with itself.

Observation V.X

Minimising the number of adjacent phonon barrier tiles improves their effectiveness.

Figure 5.9: Average logical error rate of a distance-7 rotated surface code with respect to the number of interleaved independent surface codes (rows) and the number of qubits enclosed in each phonon barrier (columns) with a constant barrier permeability $b_p = 0.1$, averaged over the total duration of identical radiation-induced faults.

The first column presents the isolated effect of interleaving without phonon barriers, reaching up to an 11% reduction of the average logical error rate when 9 surface codes are interleaved with respect to the base case. The rest of the heatmap presents all the combinations of tiling dimensions and number of interleaved QEC codes. Notably, the best combinations of interleaving and phonon barrier tiles do not necessarily require many interleaved QEC codes and small tiling patterns. In the context of the modelled quantum chip, the best average logical error rate of 6.51×10^{-4} is achieved for 14-qubits phonon barrier tiles with 8 interleaved rotated surface codes. However, by interleaving 4 rotated surface codes, and using phonon barriers around 15 qubits tiles, the average logical error rate is 9.3×10^{-4} during a radiation-induced fault, a performance which is comparable to the combination of single-qubit tiles and two interleaved rotated surface codes, whilst boasting a reduction of more than 70% of the total cost of implementing phonon barriers on-chip.

Observation V.XI

The cost of phonon barrier tiles can be further reduced via interleaving and the usage of larger distance codes, whilst reaching comparable radiation-induced fault tolerance.

Radiation event tolerance can thus be reached by employing a combination of both quantum circuit interleaving and phonon barrier tiling without incurring in prohibitively costed solutions.

5.6 Chapter summary

I have modelled and simulated substrate barriers in a superconducting quantum chip, with the intent of reducing the spatial correlation of radiation-induced faults in QEC codes. Though a tiling of the device's substrate, group one or multiple physical qubits in tiles delimited by a barrier. This barrier reduces the rate at which energy deposited by radiation events can spread across the quantum chip, without incurring in any additional overhead for QEC codes. I considered the rotated surface code, measuring variations in the logical error rate across multiple code distances, barrier permeability factors, and intrinsic noise intensities.

Linking back to the research questions posed at the beginning of this chapter, in the context of **RQ1** I have characterised the immediate effect of barriers in limiting the temporal duration of radiation-induced faults, over a range of barrier permeability factors. Regarding **RQ2**, as the barrier permeability diminishes, so does the peak logical error rate, eventually reaching the QEC code's characteristic noise floor. I have highlighted how tiling size, up to one quarter of the QEC code's qubits, is sufficient to reach appreciable logical error rate reduction, prompting cost reductions, while tile positioning has an even more fundamental barrier efficacy impact. Moreover, I have answered **RQ3** by correlating the benefits of lowering the barrier permeability and lowering the intrinsic noise rate of the quantum computer, showing that a QEC code's distance impacts the efficacy of barriers, but can not improve their intrinsic noise error threshold. As such, less permeable barriers should be employed together with less intrinsically noisy quantum hardware to observe significant improvements. On the topic of **RQ4**, I have considered the co-design approach of interleaving of multiple independent QEC codes, with the aim of increasing the spatial separation of qubits that refer to the same QEC code. This method proved to be a viable to improve radiation-induced fault tolerance, reducing both the peak logical error rate and the persistence of the fault's temporal tail. At last, regarding **RQ5**, the usage of both phonon barrier tiles and interleaving shows positive interference, further limiting the effects of radiation-induced faults below the considered rotated surface codes' thresholds.

This chapter underlines the importance of addressing radiation-induced faults, providing a model with simulation results supporting a solution fit for addressing their correlated nature. Albeit the results hereby presented are very promising, other approaches must also be investigated. This includes other phonon barrier tiling methods, other hardware hardening methods, possibly radiation-aware decoding-techniques and QEC codes, as many other factors may come to play a role in ultimately extirpating the scourge of radiation-induced faults from superconducting quantum computers.

Chapter 6

Fault propagation in hybrid algorithms

Hybrid quantum-classical machine learning models, called *Quanvolutional* Neural Networks (QNNs) [217], deliver promising speedups in terms of convergence and inference times over the classical Convolutional Neural Networks (CNNs) while maintaining a very similar classification accuracy [218]. However, while QEC and mitigation strategies have been developed for tackling intrinsic noise, their overhead is not yet compatible with current NISQ machines and algorithms. Additionally, the transient, correlated, and stochastic nature of radiation-induced faults would in any case make QEC ineffective, since multiple qubits would be affected by the charge deposited by the particle. Thus, the current and foreseeable quantum technology will still need to deal with logical-shift errors, that come to model these radiation-induced logical-shift errors.

6.1 Objectives

This chapter aims at investigating the propagation of logical-shift errors in Quanvolutional Neural Network (QNN)s, with the goal of understanding if and how faults impact execution in an hybrid algorithm setting. The quantum circuit subject of analysis is the starting point of multiple current (and future) QNN models [218–223]. Despite the fact that extensive research

This chapter refers to the contents of the article "Understanding Logical-Shift Error Propagation in Quanvolutional Neural Networks", written by M. Vallero et al. and published in the IEEE Transactions on Quantum Engineering journal [216].

to understand and improve the reliability of traditional neural networks has been triggered already [224–226], studies about fault propagation in QNNs are still lacking. Promptly addressing the issue imposed by both intrinsic and extrinsic radiation-induced faults is the starting point for the development of new reliability solutions, before its impact becomes evident in the field. This investigation seeks to advancing the knowledge of QNNs reliability by addressing the following research questions:

- **RQ1**: What is the candidate methodology to evaluate the reliability of QNNs to logical-shift faults with fault injection?
- **RQ2**: What are the most critical qubit(s) components of the qLayer and how logical-shift faults modify its output?
- **RQ3**: What is the probability for a logical-shift fault in the qLayer to cause a misclassification a QNN?
- **RQ4**: Do other factors such as the input image, the data set, the affected image subgrid and the QNN design impact its reliability?

At first, I showcase a methodology to track fault propagation in QNNs by considering three different implementations of the very first such model ever designed [217]. The analysis is carried out by injecting more than 13 *billion* logical-shift faults in the quantum layer, the Hardware Efficient Ansatz used for implementing the quantum convolution, adapting to QNNs an open-source fault-injector for quantum circuits (*QuFI*) [92]. The objective is to track how faults in the quantum layer propagate in the network during inference and why they cause misclassifications. The error propagation is first studied in the quantum circuit implementing a single convolution, and later in various designs of the same QNN, varying the dataset and the network depth. By tracking the propagation through the qubits, channels, and subgrids, I identify the faults that are more likely to cause misclassifications. In fact, up to 10% of the injections in the quanvolutional layer cause misclassification and even logical-shifts of small magnitude can be sufficient to disturb the network functionality. Corruptions in the qubits' state that alter their probability amplitude are more critical than the ones altering their phase, that some object classes are more likely than others to be corrupted, that the criticality of subgrids depends on the dataset, and that the control qubits, once corrupted, are more likely to modify the QNN output than the target qubits.

The foundational topics on quantum information theory, quantum circuits, intrinsic noise and radiation-induced faults have been introduced in Chapter 1, Sections 1.1.1, 1.1.2, 1.1.5 and 1.1.7, respectively. The rest of the chapter is organised as follows. Section 6.2

provides background on quanvolutional neural networks, while Section 6.3 outlines the design space exploration of this chapter's evaluation. Then, Section 6.4 describes the adopted experimental setup, and Section 6.5 presents and discusses the experimental results and their implications. Section 6.6 highlights the impact of the proposed methodology. Finally, Section 6.7 draws conclusions and paves a path for future work.

6.2 Background

This Section covers the fundamentals of quanvolutional neural networks, providing the necessary information about the context on the work that has been carried out.

6.2.1 Quantum Machine Learning

Quantum Machine Learning (QML) explores how to devise and implement efficient quantum circuits that offer advantages over classical machine learning algorithms [227, 228]. The classical machine learning neuron operation is encoded in a binary fashion as *active* or *resting*, which could intuitively be translated to the basis states $|0\rangle$ and $|1\rangle$ of a qubit. This theoretically allows learning models to exploit quantum features like superposition and entanglement, possibly providing speedups or new processing approaches [217, 219, 229].

Li *et al.* [230] present an exciting and long-awaited application of quantum multiplicative weight primal-dual ideas in supervised machine learning, achieving a quadratic improvement over classical counterparts. In addition, Kerenidis *et al.* [231] propose quantum classification via Slow Feature Analysis, while Havlíek *et al.* [232] have developed and tested fully quantum neural networks, such as quantum support vector machines, on real quantum hardware, showing how an ever-increasing number of approaches are being adapted and tested with success in the quantum computing field. Recently, also Convolutional Neural Networks (CNNs) have been mapped on quantum circuits. The quantum convolutional layer (quanvolutional layer or qLayer for short) encodes a convolution kernel and a max pooling operation in the structure of a Bounded Quantum Polynomial time (BQP) circuit, called Hardware Efficient Ansatz, and applies it to local subsections of an input, producing an output of higher-level features. The substitution of a classical convolutional layer with a quanvolutional layer maintains the accuracy unaltered (since the two layers perform a comparable operation), but the network with the quanvolutional layer still presents a lower loss and a faster convergence [217–219, 229, 233]. The models proposed in these papers make use of the Hardware Efficient Ansatz circuit, which is extensively analysed in this chapter, to derive the quantum layer. The detailed reliability evaluation of the quantum

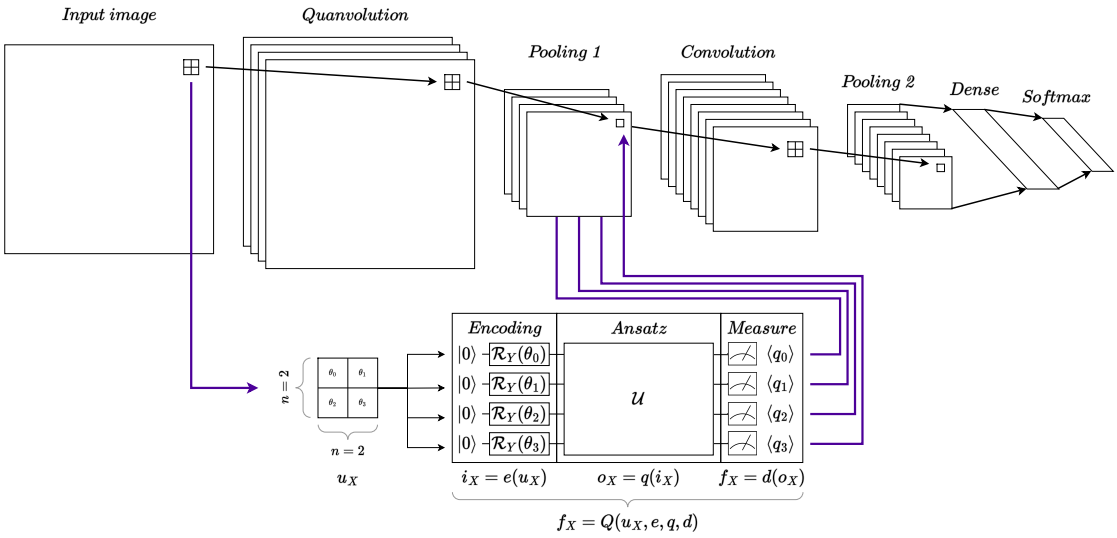


Figure 6.1: Quanvolutional Neural Network architecture.

layer together with the fault effect characterisation proposed can be directly applied to most of the available QNN models. To showcase how these results and observations can be used to evaluate the quantum fault propagation in QNN models, the original implementation is selected as a case study [217]. An exhaustive fine-grain fault injection campaign, considering three incrementally complex versions of the original design, is performed so as to let the reader compare the results with traditional convolution fault propagation.

A generic hybrid architecture example is depicted in Figure 6.1, with details of the quanvolutional layer. The input image is divided into 2×2 subgrids, then fed to a four-qubit quantum circuit performing both a 2×2 convolution and pooling operation on each subgrid. The output of the quanvolutional layer is a tensor of 4 channels representing the extracted feature map. The combination of all subgrids is the output feature map that is propagated to the downstream layer. As suggested by the results, logical-shift faults as the one caused by intrinsic noise or natural radiation, can potentially corrupt the output prediction, therefore justifying the reason for studying faults' impact in QNNs.

6.3 Exploration of Design Space

To have a thoughtful understanding of logic-shift error propagation I proposed a bottom-up approach, starting from a per-qubit reliability characterisation of the qLayer circuit, to later consider the fault propagation in the QNN and its impact on the final classification. This includes studying three network designs with incremental depths and two datasets. The hereby proposed methodology can be adapted and easily applied to test fault propagation

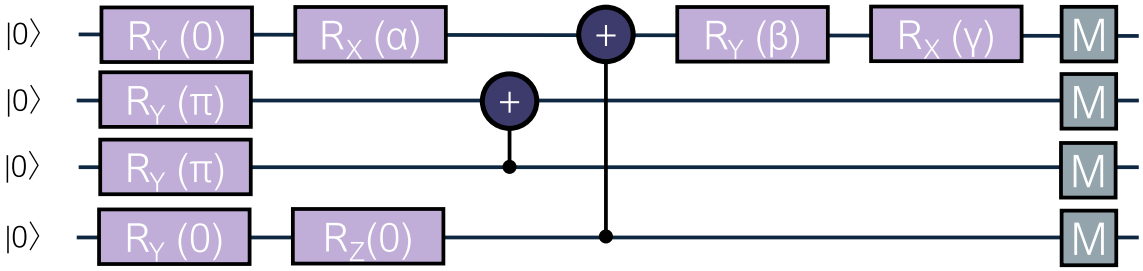


Figure 6.2: Ansatz circuit implementing quanvolution.

in any other QML model, although such extensive analysis exceeds the scope of this chapter. Several aspects that can impact the fault effect on the QNN operation have been highlighted, from the dependence of error propagation with the input image to the vulnerability of different qubits and different subgrids (position of the corrupted quanvolution in the feature map). Only faults affecting the quantum part of the QNN are considered, with no fault directly introduced in the classical layers. The injection methodology considers a one to one mapping of logical qubits to physical ones, without QEC, as is the custom for current NISQ algorithms, following hardware constraints. Given the susceptibility of error correction to radiation, however, the results are still to be held relevant for future error corrected quantum machines as well.

6.3.1 Quanvolutional layer

The first proposed evaluation is the characterisation of the reliability profile of the ansatz 4-qubit quantum circuit implementing the *quanvolution layer* (qLayer), shown in Figure 6.2. The values that parametrise the gates are randomly generated and kept constant during the experiments, set as follows: $\alpha = 2.353$, $\beta = 4.599$, $\gamma = 3.761$, and $\delta = 5.974$. The objective of this first exclusively quantum analysis is that of understanding the inner workings of fault propagation in this quantum circuit. The qLayer is composed of three main sections: encoding, the actual random circuit, and measurement. The sequence of these elements produces an output tensor of size comparable to a classical convolution *and* pooling operator on 2×2 subgrids with a stride of 2. The qLayer is not a direct quantum translation of the convolution operation for CNNs, but rather it is the standard quantum dual of a convolution kernel for QNNs, as per the works of [220–223, 234–236]. Each of the 4 qubits calculates one of the 4 channels of the feature map. Larger qLayers are possible, but in all the available QNN implementations the size of the subgrids is kept to 2×2, which provides the best trade-off between accuracy, circuit complexity, and performance [217, 219]. Thus, for this chapter, the qLayer size is kept constant at 2×2. The proposed methodology

and the following insights can be applied to any current and future qLayer sizes.

The circuit contains two CNOT gates, each controlled by qubits 2 and 3, respectively targeting qubits 1 and 0. The CNOT gate, a Multi-Qubit gate, will perform an X-gate (the equivalent of the NOT gate in classical computing) on the target qubit if the state of the control qubit is $|1\rangle$. Given the entanglement caused by these two gates, the propagation of faults from control qubits to target ones is tested as well. At the end of the circuit, the expectation of each qubit is extracted, by running the circuit on a minimum of 1024 shots.

To have a fine-grain evaluation of the reliability of the quanvolution operation, the analysis carried out in Section 6.5.1 considered a fixed input subgrid and injected a fault in each of the 4 qubits (one qubit corrupted at a time). The aim of the per-qubit evaluation is to understand which channel (qubit) is less reliable and if there is a difference between control and target qubits. As detailed in Section 6.5.1, it followed that faults in control qubits have a more significant impact on the QNN output, since they get propagated to the target qubit, and that the injection on one qubit affects only the channel associated with the corrupted qubit, with negligible effects on the other channels.

6.3.2 QNN and input data set

To understand how faults occurring in the qLayer propagate in the QNN, the insights gained from the previous analysis are leveraged by testing three hybrid models, to make a direct comparison with the well studied fault propagation mechanisms of convolution layers in classical CNNs. Logic shifts have been injected only at inference time, not during training, following the common practices of traditional Convolutional Neural Network (CNN) reliability evaluation [224, 225]. In fact, while errors during training can potentially reduce the performance or increase the convergence time, these effects are easily detectable and solved with additional training steps. On the contrary, silent errors during inference can lead to potentially harmful real-time mispredictions and should be strictly avoided.

Recent experiments showed that the charge deposited by radiation migrates in the silicon substrate, eventually affecting physically close qubits [126, 134]. Since the 4 qubits implementing the qLayer must be connected and close to each other, the single-particle interaction is expected to corrupt them all. As such, in the QNN reliability evaluation, all four qubits will be simultaneously corrupted during subgrid computation.

The QNN design available in [217], a hybrid classical-quantum adaptation of the Le-Net model [237] for image classification, is taken as the baseline for one of the first (classical and quantum) models to be designed. The inputs used are taken from the Modified National Institute of Standards and Technology (MNIST) handwritten digits and fashion data sets [238],

both consisting of $70,000$ 28×28 pixels greyscale images representing either handwritten digits or clothing apparel.

Training and testing are performed on the MNIST data sets (handwritten digits and fashion items), since they are widely regarded as a cornerstone of classical Machine Learning (ML) research. Additionally, the current scale of quantum devices does not yet allow for the usage of state-of-the-art, high-resolution data sets. Nevertheless, the provided results and insights are still fundamental for characterising the considered quantum design.

The QNN receives, as input, greyscale images with values ranging between 0 and 255. For each 2×2 subgrid in the input image, each pixel is encoded using amplitude embedding through a parameterised rotation $\mathcal{R}_Y$ around the Y-axis, mapping each value linearly to the range $[0, \pi]$. In the qLayer, the quanvolution circuit is executed for each subgrid and the resulting tensor is propagated to the downstream layers.

The fault propagation during the QNN inference is traced to the output correctness. A distinction is made between masked faults (the output is unaffected), tolerable Silent Data Corruptions (the output is altered, but the correct class is selected), and misclassifications.

To have an overview of possible logical-shift errors propagation an exhaustive fault injection is performed in at least 30 random images from each data set, meaning more than $273,646,592$ injected faults per image. In other words, once the injection site has been selected (qubit, channel, grid, etc...) a complete fault injection is performed, considering all the possible parameterised rotations, for each input image. Then, to understand the impact of error propagation from the input frame, further experiments have been performed on 100 images. No significant dependence of fault propagation with the input image class has been observed.

6.3.3 QNN Models

The error propagation in classical CNNs is known to be dependent on the network depth (i.e., the number of layers the fault needs to traverse to reach the output) [224, 225]. In particular, convolution tends to spread the faults happening in upstream (traditional) layers. With the aim of understanding the dependence of logical-shift error propagation on the network depth, three designs of increased complexity of the same QNN have been considered (based on [217]), hereby called *ModelA*, *ModelB*, and *ModelC*.

ModelA, whose structure is represented in Figure 6.3, is the quintessential Quanvolutional Neural Network, composed of the minimum number of layers. The qLayer takes as input a $(28, 28, 1)$ tensor and outputs a $(14, 14, 4)$ tensor, whose output is then flattened and redirected into a softmax dense layer.

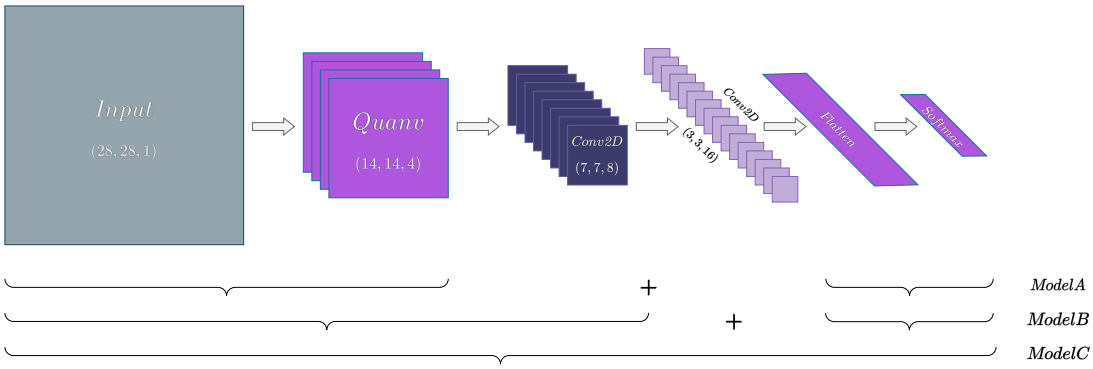


Figure 6.3: Quanvolutional neural network model composition.

ModelB and ModelC are derived from the barebone ModelA by adding respectively one and two cascaded Conv2D operators between the quanvolutional and flatten layers, respectively. The concatenation of a qLayer with classical convolutional layers has been done following state-of-the-art approaches in literature [217], choosing suitable filter sizes for the classical layers in the networks: each additional Conv2D layer doubles the number of filters used in the preceding operator and uses a filter size of 2×2, with a stride of 2. It is worth noting that multiple cascaded qLayer have not been considered, following the approach in [217].

Each of the derived designs has been re-trained to adapt the weights to the network depth. The accuracy on both the training and validation data sets obtained after the training of the three QNN designs is similar (at most 3% of difference) and comparable to the performance of the corresponding fully classical implementation.

Interestingly, as detailed in Section 6.5.4, increasing the depth of the QNN by adding cascaded traditional convolutional layers *reduces* the quantum transient fault impact on the output, masking some faults and reducing the probability to have misclassifications.

6.3.4 Single and Multiple Subgrid Injections

Finally, I have also compared the reliability of QNNs when multiple subgrids are corrupted. In fact, the near-future prospect of highly integrated quantum chips justified the possibility multiple subgrid corruptions at inference time, as detailed in Section 6.5.5. For this reason, additional experiments injecting on two distinct subgrids have been performed. As shown, when multiple subgrids are corrupted, the impact on the QNN's output is higher, increasing the probability of having misclassifications.

6.4 Experimental Setup

This section describes the setup used to conduct the experiments, providing details on the framework used to model the transient fault's effect.

6.4.1 Logical-Shift Error Model

Fault injection in quantum circuits is more complex than in classical CMOS devices. In fact, the classical bit has only two states (0 and 1) and, thus, a bit-flip fault model is sufficient to study the reliability of CMOS devices. As seen in Observation I, for quantum bits in a superposition, the interaction of ionising particles can modify the quantum state by inducing a parametrised rotation. The magnitude of such parametrised rotations depends on the deposited charge, as shown with simulations [23] and experimentally validated [129], which can range from *meV* to *GeV*[135]. Thus, in contrast to classical computing, the quantum fault model has to take into account many more possible state changes than a "simple" bit-flip (i.e. the X-Pauli gate), as a particle impact can induce *any* given parametrised rotation.

Since the energy of the impinging particle is continuous in a wide range (meV to GeV) [239], the fault's rotation range will also be continuous. As such, all parametrised rotation magnitudes have been considered in the fault injection. This makes for a systematic analysis which is as general as possible, without being tied to a specific particle energy range. The fault model and the results hereby presented can be easily weighted or normalised once more information on the correlation between exact impinging particle energy and fault amplitude will be known.

6.4.2 Logical-Shift Injection and Simulation

This chapter only considers faults affecting the quantum part of the QNN. The effect of faults in the classical parts of CNNs has already been investigated deeply [224–226]. The simulations have been carried out without considering a device-level noise profile, as it is a well-separated event with respect to particle-impacts, and its effects would add up to those of transient faults. In addition to this, one must recall that noise has close to no impact on the Ansatz circuit, as previously stated in Section 6.3.

To inject logical-shift errors into the quantum convolution circuit during the QNN inference by applying a tuned stimulus to modify the qubit state at inference time, while no fault is injected at training time. In practical terms, faults are represented via a parameterisable U3 gate, which can induce rotations of arbitrary magnitude, modify the ϕ and/or θ that encode the qubit's information [92]. The ϕ angle modifies the phase of a qubit, and the

θ angle changes the $|0\rangle - |1\rangle$ probability. The possible range for each angle without state duplication are $\phi = [0, 2\pi]$, and $\theta = [0, \pi]$. The discretisation of the angles range over a $\frac{\pi}{12}$ step size results in 325 possible configurations (i.e., distinct fault magnitudes to be injected).

To track fault propagation in QNNs, the applicability spectrum of the open source QuFI has been broadened by porting it to the *Pennylane* [39] framework. This opens up the possibility of running quantum circuits on devices provided by different vendors implementing different technologies, not to be limited to IBM machines, and a more direct QML-oriented development, since Pennylane inherently supports multiple libraries dedicated to the task.

6.4.3 Fault Effect Evaluation

As previously stated, the quantum circuit output is probabilistic, with each possible state having a certain probability to be selected. For instance, a 2-qubit circuit has 4 possible states: $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Ideally, the correct state will have the highest probability so it can be selected as the output. The *Quantum Vulnerability Factor (QVF)* [91] metric has been selected to measure the impact of a transient fault in the output probability distribution. The QVF, corresponding to the Architecture Vulnerability Factor (AVF) [240] and the Program Vulnerability Factor (PVF) [241] in traditional computing systems, ranges from $[0, 1]$, and indicates the probability of a fault to propagate affecting the output. In other words, the QVF indicates how likely the fault is, given the probabilistic output, to induce the selection of a corrupt state. A QVF close to zero indicates a high probability of selecting the correct state. Values close to one indicate that an incorrect state is likely to be selected. QVF values around 0.5 mean that the correct state and at least one incorrect state have similar probabilities, which makes the identification of correct states dubious.

To evaluate the effect of the propagation of logical-shifts in the qLayer to the downstream layers the *misclassification* rate of the tested QNNs has also been measured. Faults have been injected into the qLayer at inference time, letting the corrupted output feed the downstream operations. Then, the classification of the faulty execution is compared with the fault-free one. However, the classification accuracy with respect to the ground truth since is not subject of evaluation, as the objective of the analysis is tacking the impact of faults in the execution of a QNN. The unlikely event of a fault improving accuracy is purely stochastic and not scientifically relevant, as one cannot rely on radiation to improve the QNN's accuracy. Moreover, no such events have been observed.

6.5 Results

This section details the experimental results obtained from 13 billion logical-shift fault injection simulations ($267,233$ quantum circuit injections per input image, per configuration). This extensive evaluation provides a very accurate evaluation, with the statistical error being lower than 1% [242]. The bottom-up evaluation starts from the characterisation of the reliability of the quanvolution circuit, then understanding the fault effect on the QNN's output, and identifying how many faults induce misclassification. Then, the QNN's reliability correlation with the data sets, the input images, and the injected subgrid are considered. Finally, the fault propagation is tracked in three different QNN designs of increasing complexity (ModelA, ModelB, ModelC), to later address the impact of double faults.

6.5.1 Quanvolutional Layer Reliability

This first reliability evaluation details the propagation of logical-shift faults in the quantum computation core of QNNs, that is, the quanvolutional layer implemented with the Ansatz circuit depicted in Figure 6.2. For this evaluation, the quanvolutional layer is analysed as a standalone quantum circuit, i.e. without the integration with the upstream and downstream portions of the QNN. Each qubit has been separately injected to reach a finer grain set of results.

To assess the resilience profile of the circuit, the chosen input is a constant 2×2 subgrid, with the top-right and bottom-left pixels as white (value 255) and the other two as black (value 0), i.e., a diagonal black and white subgrid. This corresponds to encoding qubits 0 and 3 of Figure 6.2 in state $|0\rangle$, whilst qubits 1 and 2 are encoded in state $|1\rangle$, since they are prepared by rotations around the Y-axis of 0 and π radians, respectively.

Figure 6.4 presents, for each (θ, ϕ) logical-shift, the QVF for the qLayer circuit, increasing the logical-shift in θ (0 to π) and ϕ (0 to 2π). Each qubit has been subject to a separate fault injection. A QVF close to 1 (red) indicates a shift that entails a high probability of selecting the wrong output, while values close to 0 (green) indicate shifts that do not modify the output selection.

In Figure 6.4 one can see that the QVF increases (worsens) moving to the right of the picture, while being almost unaltered moving up in the picture. This means that the qLayer circuit becomes highly affected by the azimuthal faults (θ logical-shift) for values greater than $\frac{\pi}{2}$. While this result might seem obvious and intuitive (a higher modification leads to a higher impact on the output), it has been shown that for quantum circuits logical-shifts of higher

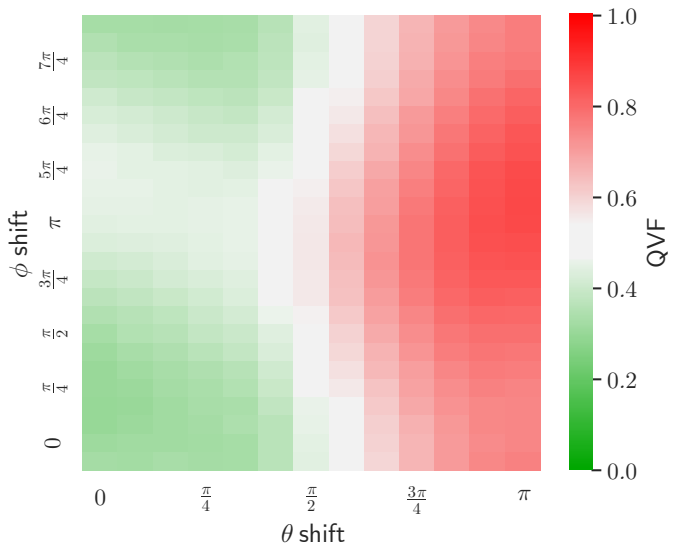


Figure 6.4: Quantum Vulnerability Factor heatmap of the quanvolutional layer.

magnitude do not necessarily have a higher probability to modify the circuit output [92].

Interestingly, the qLayer shows a relatively low vulnerability to the polar angle (ϕ), albeit a small QVF rise between $\frac{3\pi}{4}$ and $\frac{5\pi}{4}$. Analysing the details of the single-qubit QVF heatmaps highlighted that qubits 0 (target) and 3 (control) are responsible for lowering the average resilience of the circuit for $0 < \theta < \frac{\pi}{4}$ and $\frac{3\pi}{4} < \phi < \frac{5\pi}{4}$ (white region). This is because those two qubits undergo more quantum gates than qubits 1 and 2.

The QVF heatmap suggests that the θ shifts are critical, whilst ϕ logical-shifts are not. This property has been further investigated at the network level in the following Section 6.5.2.

Observation VI.I

Due to the usage of amplitude embedding, ϕ logical-shifts do not significantly modify the qLayer output, while θ shifts cause an effect on the output that is proportional to the shift magnitude.

From this it also followed that a single injected fault in a qubit of the qLayer circuit modifies all its logically connected qubits, and consequently the output bit-string. This means that the computation of the qLayer is likely to spread the fault, corrupting the cascaded layers in the network's architecture.

Observation VI.II

A fault in a single qubit of the qLayer spreads to all its logically connected qubits.

Phase-shift fault-induced misclassification probability			
QNN design	data set	Single grid	Double grid
ModelA	digits	3.49%	6.05%
	fashion	1.23%	1.94%
ModelB	digits	5.52%	10.65%
	fashion	3.99%	6.58%
ModelC	digits	1.69%	8.62%
	fashion	3.16%	6.33%

Table 6.1: The misclassification probability is always higher with double grid faults, and the *fashion* dataset shows lower average misclassifications.

6.5.2 Fault Propagation in QNNs

To understand how faults propagate in QNNs and identify the faults that generate misclassifications, an extensive fault injection campaign is performed, by injecting a logical-shift fault in each of the 4 qubits executing one quanvolution (i.e., calculating one subgrid). This includes the three network models, with an increasing depth, on both input data sets and over single and double subgrids injected. Faults that did not corrupt the softmax vector output of the neural network have been labelled as *masked*. Faults that modified the output vector have been labelled as either *tolerable* if they did not alter the output predicted class, or *misclassified* otherwise.

It has been observed that *all* of the θ logical-shifts propagate to ModelA's output (not necessarily modifying the classification) while *none* of the injections of ϕ logical-shift causes an observable effect on the network output. The fact that the injections of ϕ logical-shift do not propagate should not surprise. As discussed in Section 6.4, the qLayer circuit uses amplitude embedding, i.e. maps the convolution data in the θ angle of the qubit state, the $|0\rangle - |1\rangle$ probability. Thus, changes to the phase (ϕ angle) of a qubit state are expected to have a small impact on the qLayer output (as confirmed in Observation II) and, as the fault injection in the QNN shows, ϕ polar shifts do not modify the inference. In the following, *only θ shift injections have been reported*.

Observation VI.III

In a simple QNN with just one qLayer no ϕ logical-shift modifies the output but all θ logical-shifts propagate to the output.

Table 6.1 shows the measured average probability amongst all the logical-shift faults injected in the qLayer circuit to induce a misclassification across all the possible configurations

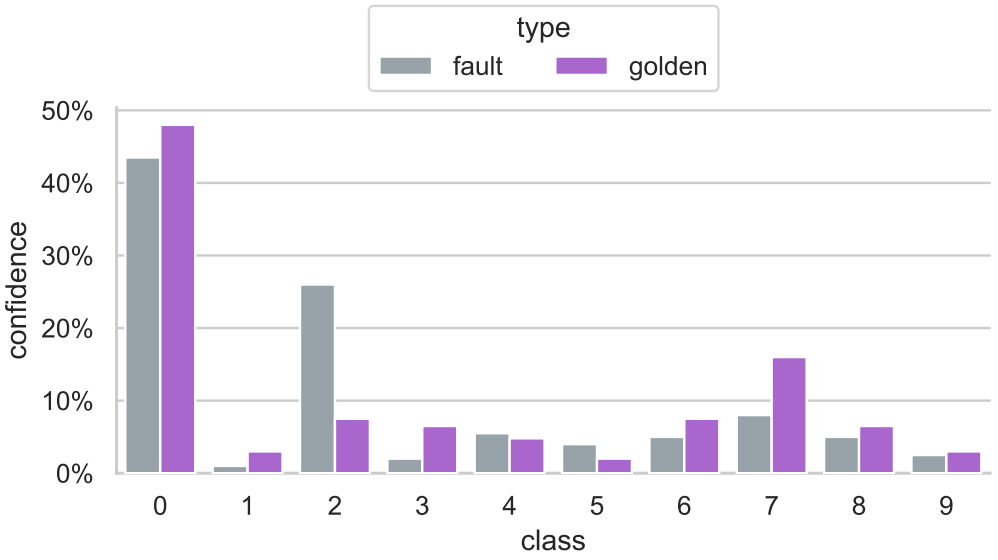


Figure 6.5: Masked event in the softmax layer's output.

of data sets, models, and the number of subgrids injected at a time. The analysis shows that the misclassification rate can vary from 1.23% to up to 10.65%, depending on the QNN design and data set. This misclassification probability is the result of the interaction of a plethora of factors: the following subsections detail the dependency of the misclassification rate from the logical-shift magnitude, network design, and the number of simultaneously injected subgrids.

The measured misclassification rates for QNNs, shown in Table 6.1, are comparable to the ones of classical CNNs, that range from 1% (floating-point) to 7% (with a specific fixed-point data type) [224]. Observation II underlines that the CMOS error rate is *orders of magnitude lower* than the one of a superconducting transmon qubit. Thus, while CNNs and QNNs have similar misclassification probability, the latter are much more likely to experience a fault, thus experiencing a considerably higher misclassification rate.

Observation VI.IV

The probability for a fault to generate a misclassification in a QNN or in a CNN is comparable. However, in QNN the fault rate is orders of magnitude higher.

Figures 6.5 and 6.6 provide an example of the effects of a *tolerable* fault and of a *misclassification* fault on the softmax vector output, respectively. To better understand the effect of fault propagation, Figure 6.5 shows an example of a fault that does not induce misclassification whilst modifying significantly the classes' probability distribution. The plotted data refers to a $\theta = \frac{\pi}{2}$ fault injected in all 4 qubits of the qLayer applied to a single subgrid

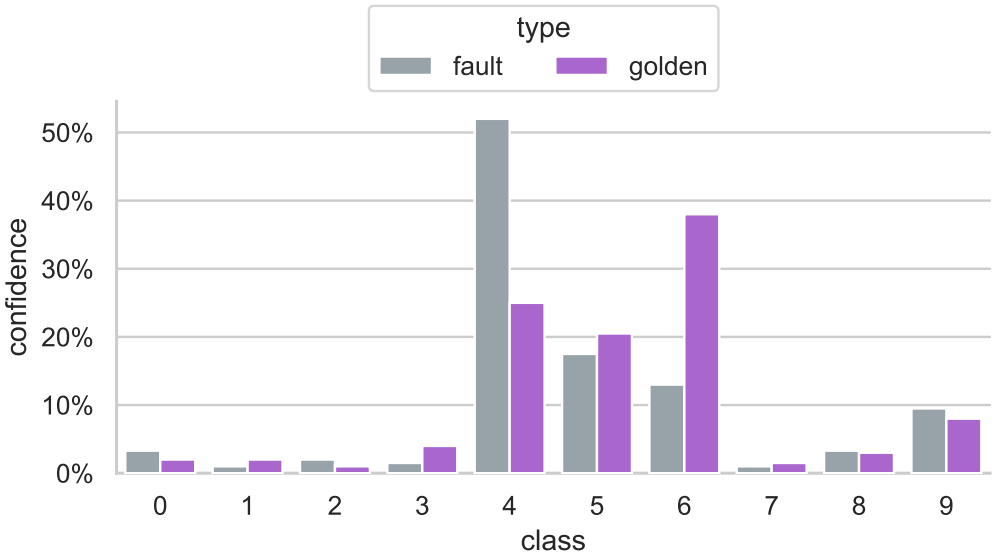


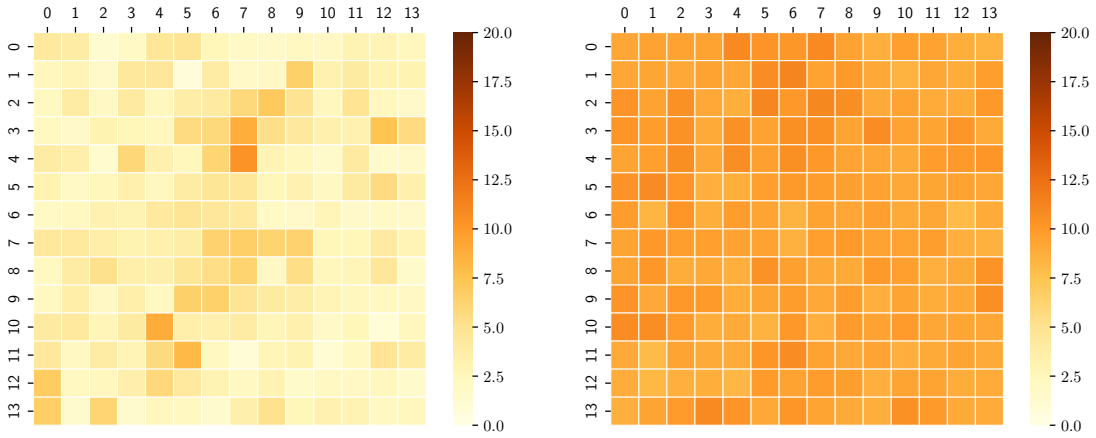
Figure 6.6: Misclassification event in the softmax layer's output.

out of the 196 possible subgrids of the input image. The softmax layer outputs for the baseline fault-free are labelled as "golden" and coloured in green, while the faulty executions are coloured in red. In the baseline fault-free execution, *class 0* is selected with very high confidence (0.48 vs 0.18 of the second class). The fault triplicates the confidence for *class 2* to be selected while reducing the one for *class 0*. Nonetheless, despite a significant reduction in the classification confidence (0.44 of *class 0* vs 0.26 of *class 2*), *class 0* is still the one with the highest probability.

Figure 6.6 shows an example of a misclassification fault. Once again, the outputs for the baseline fault-free are labelled as "golden" and coloured in green, while the faulty executions are coloured in red, considering the same $\theta = \frac{\pi}{2}$ fault amplitude. The baseline fault-free execution classifies the input as *class 6*, but with low confidence (0.38), since both *class 4* and *class 5* have a high probability at the QNN output. The $\theta = \frac{\pi}{2}$ fault injected in the qLayer reduces to 0.33% the probability of *class 6* and doubles *class 4* probability, promoting it to the selected output class leading to a misclassification.

6.5.3 Misclassification Dependence on Subgrid and Input

To understand possible QNN reliability dependencies from the input frame and the corrupted subgrid, an extensive fault injection has been performed, by considering 100 images for each data set and injecting a fault in every single subgrid of the input image. Since each image has 196 subgrids, this campaign is computationally demanding to execute, requiring a total



(a) Handwritten digits data set

(b) Fashion data set

Figure 6.7: Misclassification probability heatmap by dataset and subgrid

of more than 7 billion injections for both data sets.

Figure 6.7 shows the average misclassification probability for each subgrid on the digits (a) and fashion (b) data sets. Data has been obtained testing 100 images of (a) digits and (b) fashion data sets. To ease visualisation, the misclassification rate has been plotted as a heatmap, where the (row, column) are the coordinates of the subgrid location. As can be seen by comparing Figures 6.7a and 6.7b, the two data sets have a completely different reliability dependence on the corrupted subgrids.

In the handwritten digits data set (Figure 6.7a) some subgrids are extremely likely to generate misclassification while others, even if corrupted, have a low probability to impact the network output. For instance, the subgrid in (row: 4, column: 7) has a misclassification ratio of 10.3% whilst a fault in the subgrid (row: 1, column: 5) has a 0.8% probability to induce a misclassification. In the fashion data set (Figure 6.7b) the heatmap has a homogeneous distribution of misclassification ratios, suggesting that the probability of incorrectly labelling an image on this second data set is not significantly dependent on the corrupted subgrid. Finally, no input image class has been registered to have had a correlation with the misclassification rate.

Observation VI.V

The misclassification probability depends on the corrupted subgrid in the digits data set, while there is no dependence between misclassification and object class.

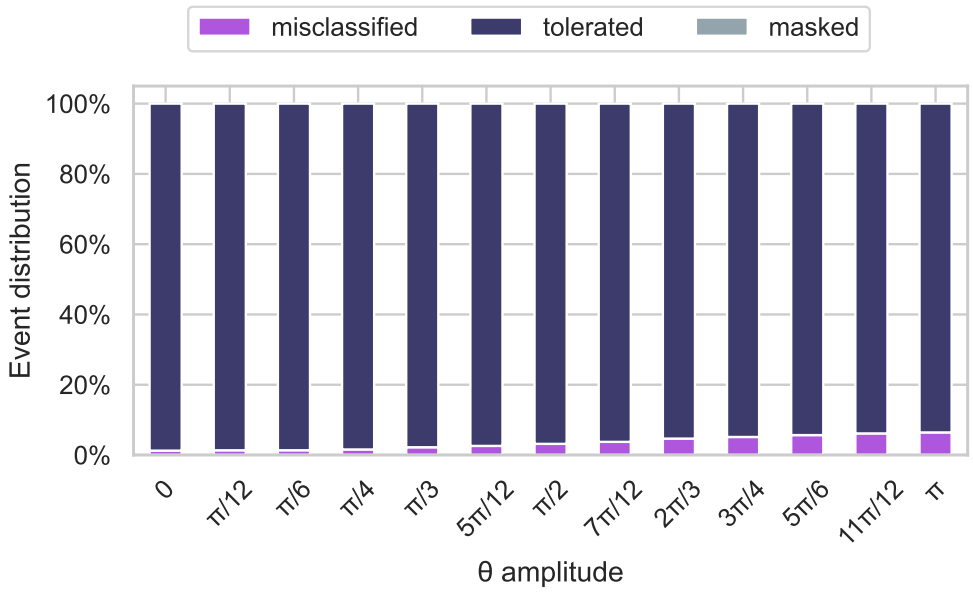


Figure 6.8: Model A misclassification ratio by fault amplitude.

6.5.4 Fault Propagation Dependence on QNN Design

To understand if the QNN design impacts the fault propagation, a single *random* subgrid has been subject to fault injection in the qLayer of ModelA (one qLayer), ModelB (one qLayer and one Conv2D layer), and ModelC (one qLayer and two Conv2D layers), with data set partitions of size 30. Details about the three QNN designs can be found in Section 6.3.3.

The analysis of ModelA on the MNIST handwritten data set partition is presented in Figure 6.8, highlighting the percentage of *misclassified*, *tolerable*, and *masked* faults with respect to the amplitude of the angle θ in the parameterisable U3 fault gate. There is an evident correlation between the amplitude of θ and the incidence of misclassifications in the network's output. Faults with an amplitude of just $\theta = \frac{\pi}{2}$ produce a 3.18% misclassification ratio, which bumps up to 6.43% for a fault amplitude of $\theta = \pi$. Moreover, given the relatively shallow architecture of ModelA, the classical part of the network cannot sufficiently compensate for the fault and no *masked* event is ever registered. All the injected faults in fact produce a variation in the output softmax vector.

In Figure 6.9, once again computed on the handwritten digits data set, ModelB undergoes a single fault iin one of the subgrids of the qLayer, which gets propagated first through the Conv2D layer and later in the Flatten and Softmax layers. On a fault gate amplitude of $\theta = \frac{\pi}{2}$, the misclassification ratio is valued at 5.84%, rising to 7.39% when considering the maximum fault amplitude. Much like for ModelA, it is once again clear to see that

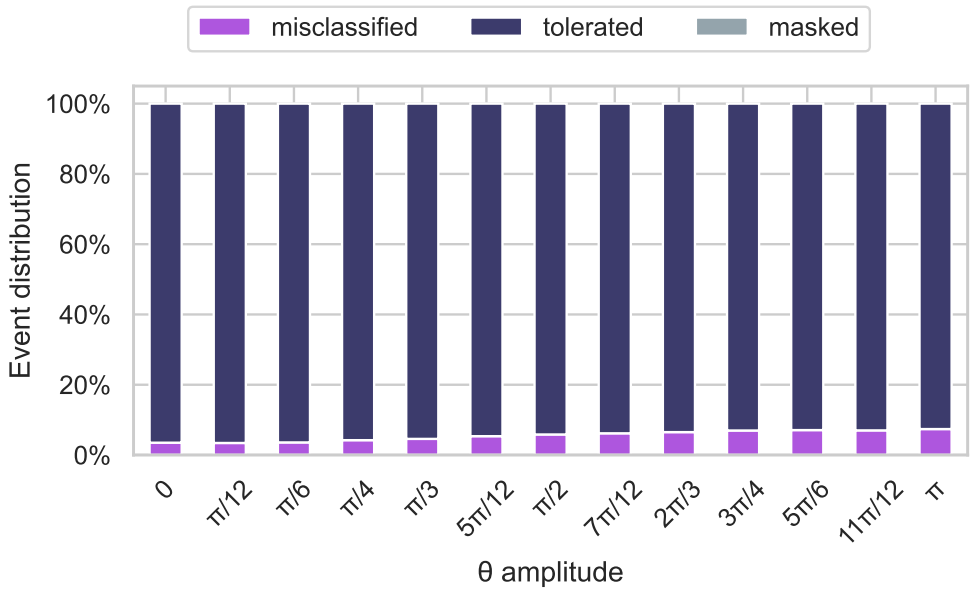


Figure 6.9: Model B misclassification ratio by fault amplitude.

there is a correlation between the azimuthal angle of the U fault gate θ and a rise in the misclassification ratio. No *masked* event has been observed. On average, as seen in Table 6.1, the misclassification ratio for ModelB is 5.52% on the handwritten digits data set, whilst the same analysis on the fashion data set boasts a slightly lower average rate of 3.99%. The average probability for ModelB to produce a wrong output class prediction increases by a significant margin in both data sets with respect to ModelA.

ModelC's reliability response to a single qLayer fault is detailed in Figure 6.10, once again on the handwritten digits data set. Unlike the other experiments, *masked* events have been registered with an average probability of 26.67%: this can be explained by the fact that the increasing number of filters in the Conv2D operators eventually disperses the effect of a portion of the faults introduced at the quantum layer and eventually those get cancelled out by undergoing a product operation with weights or kernel parameters equal to zero.

Observation VI.VI

Downstream Conv2D layers can help in masking some qLayer faults.

It is important to note that this event depends on the qLayer, as it is not the direct quantum translation of a convolution and thus boasts a different behaviour. Moreover, a significant drop in the overall misclassification rate is observed, with average values of 1.69% for the handwritten digits data set, and a maximum registered at 3.17% at the

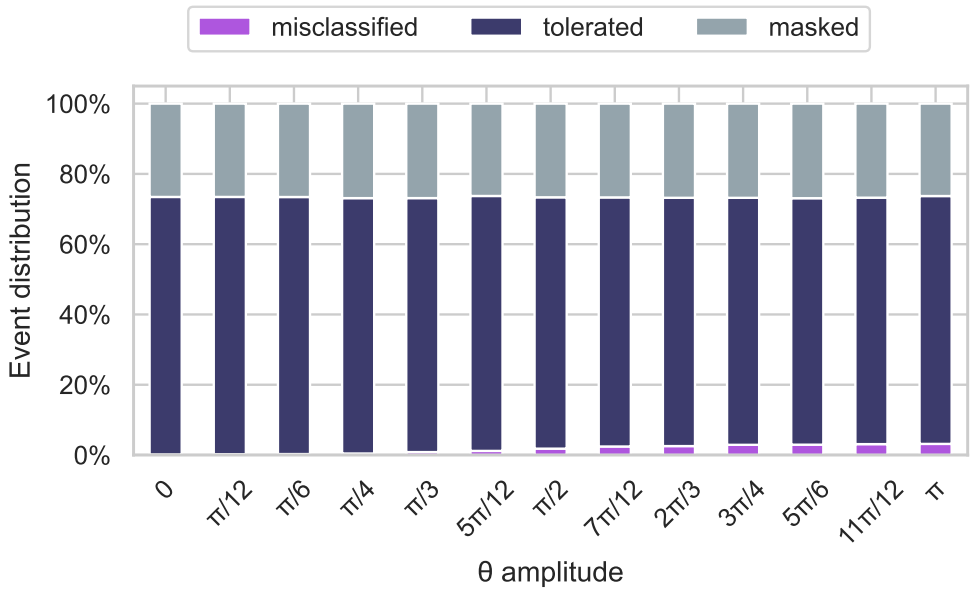


Figure 6.10: Model C misclassification ratio by fault amplitude.

highest fault gate amplitude of $\theta = \pi$. Similarly, on the fashion data set, an average of 3.16% misclassifications is registered, with a *masked* events ratio of 26.59%.

Observation VI.VII

Larger θ logical-shifts increase the misclassification probability, in all the tested QNN designs.

6.5.5 Double sub-grids corruption

In a general quantum workload, one cannot rule out the possibility to experience multiple radiation-induced corruptions across the whole execution, especially in iterative approaches such as QNNs or in deep quantum circuits. CMOS devices, in terrestrial applications, can be corrupted mostly by neutrons and the probability for a CMOS-based chip (even large GPUs) to be corrupted by an impinging neutron is very low, in the order of 10^{-6} to 10^{-8} [133, 243]. Since the flux of neutrons at sea level is about $13n/cm^2/h$, the error rate of a CMOS chip is in the order of 10^{-5} to 10^{-9} errors per hour [133], making it highly unlikely to observe two events in a single computation. Unfortunately, this does not hold for qubits, since they have an intrinsic coherent time in the order of ms and a sensitivity to radiation that is much higher than CMOS transistors (Observation I) and, moreover, they can be affected by various uncorrelated radiation sources [126, 129]. Additionally, quantum chips to are expected to

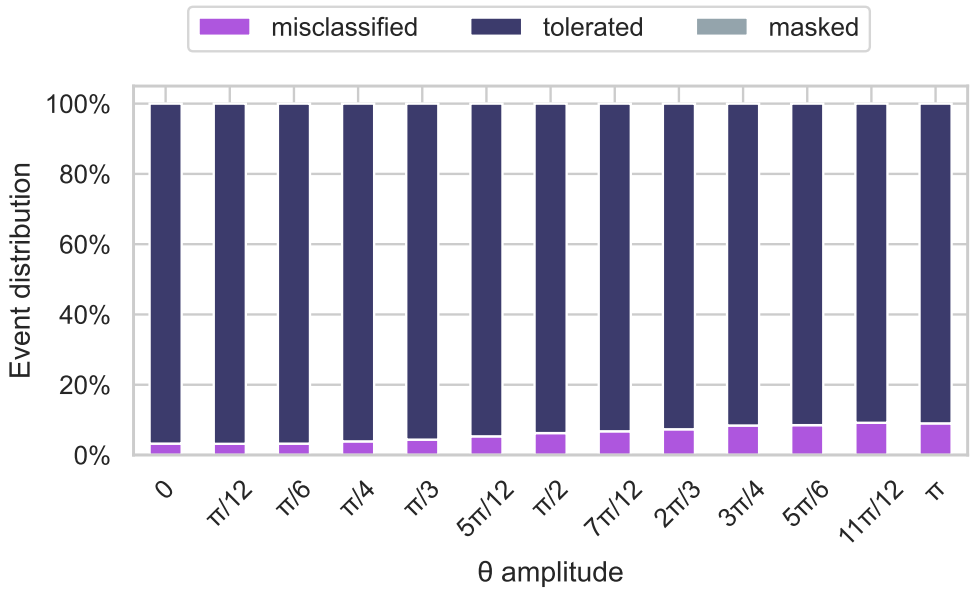


Figure 6.11: Model A misclassification ratio by fault amplitude, double fault.

become highly integrated in the near future, possibly including multiple qLayer circuits on a small surface area. As a result, it is reasonable for a single particle to corrupt multiple logical qubits or possibly even multiple qLayer circuits.

Therefore, as a final analysis, two separate random subgrids have been subject to corruption at the same time, considering all QNN models and input data sets. The results presented in Figure 6.11 refer to ModelA on the handwritten digits data set partition of size 30. Much like the single corrupted subgrid ca, a correlation between the amplitude and the misclassification ratio is evident, where a fault amplitude of $\theta = \frac{\pi}{2}$ is responsible for changing the output predicted class in 6.24% of cases, almost doubled with respect to the previous experiment on single subgrid injections. The misclassification ratio tops out at 9.0% with the highest amplitude injection of $\theta = \pi$. No *masked* injections have been observed.

Additional experiments obtained by testing double subgrid injections on both ModelB and ModelC have been performed, boasting a steady increase in the rate of misclassification events. Moreover, ModelC undergoes a reduction in the number of *masked* events when the number of injected subgrids is doubled. The average rates for these experiments are reported in Table 6.1.

Observation VI.VIII

The corruption of two subgrids significantly increase the misclassification probability.

6.6 Discussion and Projections

The considered QNN architecture is the first model of its kind ever proposed [217]. This design is the cornerstone over which vibrant and rapidly growing research is being carried out [220–223, 234–236]. In particular, the structure of the Hardware Efficient Ansatz is being used to implement quanvolution in the vast majority of QNNs models. For this reason, the hereby presented methodology and analyses can be used to understand the reliability behavior of current and future QNNs, either making use of the same quanvolutional layer or of layers derived from it.

Thanks to the continuous advancements in quantum computing technology, the application landscape for QNNs keeps broadening. However, as it has been shown, the widespread adoption of QC could be stifled by logical-shifts caused by either intrinsic noise or cosmic rays, particularly on superconducting transmon quantum devices [23, 24, 92, 126–131]. Despite the fact that QNNs have a misclassification ratio comparable to that of CNNs, their reliability is much more significantly hindered with respect to their classical counterparts, given that the radiation-induced fault rate for quantum devices is orders of magnitude higher with respect to CMOS. The usage of surface codes along scalability and construction quality improvements may have a positive role in improving the reliability of many QML models, at which point the hereby presented systematic results may simply be re-weighted according to the way in which they impact the output distribution. At the moment, however, there is no guarantee that surface codes will not fail in the event of a particle impact, and may as well worsen the results in this circumstance.

Hardware/software co-design has been demonstrated to be critical for quantum computers [31, 67, 244–248]. This chapter's analysis adds the logical-shift fault issue to the reliability assessment of these devices and architectures. This work's results, alongside the methodology employed, can direct algorithm design, innovative software/hardware hardening solutions development, and more robust circuit architecture implementation. For instance, quantum circuit designers could leverage the presented framework to implement and test purposefully made quantum error correction codes, adding redundancy in the most critical part or duplicating only the most critical quanvolutions, and thus largely reducing the misclassification ratio. The information regarding subgrid criticality can help, knowing the data set used in the field, in designing a future scheduler or optimiser for Quantum Machine Learning workloads to map each subgrid execution onto more or less reliable quantum hardware with respect to their impact in case of a fault. Moreover, I envision that transpilers may leverage this analysis as an additional heuristic metric, aimed at reducing the impact of radiation-induced faults, and adaptable to any physical quantum device. The final highlight

is that better training, or a different QNN design, might increase the classification confidence and reduce radiation-induced misclassifications, albeit this can hardly solve the faults issue altogether.

6.7 Chapter summary

This chapter proposed a methodology to deeply investigate the propagation of logical-shift faults in Quanvolutional Neural Networks, thus directly tackling **RQ1**. By using a fault model derived from experiments and simulations, I presented results that show how the corruption of the quanvolutional layer significantly impacts QNNs' operations and classification. Concerning **RQ2**, the data shows that θ logical-shifts are very likely to propagate in the QNN, and following with **RQ3**, up to 10% of injections induce misclassification. Regarding **RQ4**, the misclassification probability depends on the logical-shift magnitude, on the corrupted subgrid, on the data set, and on the number of classical layers that follow the corrupted layer.

Future research directions include the proposal of mitigation or hardening solutions for QNNs. Blocking the fault propagation in the quanvolutional layer may lead to a reduction in the measured misclassification events.

Chapter 7

Conclusions

Achieving fault tolerance in quantum computing, which originally seemed such a well-defined task in its formulation, revealed more and more questions as time went by. I investigated how quantum computers work in order to simulate them with different methods and varying degrees of accuracy. This led me to study state-of-the-art simulation methods at large scale, correlating quantum algorithms to classical simulation efficacy. I devised methodologies to characterise the effect of radiation events on quantum information at various layers of abstraction. This meant providing the scientific community with a comprehensive fault model, yet simple and understandable enough to prompt more research. I later leveraged this model to investigate theories on how to better reach reliability at both the hardware and QEC layer. Only a portion of those theories actually made it past the point of becoming scientific contributions. Amongst them, I proposed ways to detect radiation events and solutions to improve the reliability of quantum computers and algorithms.

The main contribution of the thesis has been to close the gap between the physics of energy depositions on semiconductors caused by radiation events and a representative approximate fault model to be leveraged from the computer science and engineering side of things. This opened up the theorisation and testing of classical computer science bound methods to the issue of quantum computer reliability, yet remaining open for further discoveries, further strengthening the bond between these two disciplines.

I am aware that my work mostly serves as a proof of concept, being a rickety bridge between different topics that hopefully can be reinforced with time, further research and new ideas. The accuracy of the dispersion of charge in the depositing medium, other than

additional considerations about the implementation and placement of single qubits can still be expanded and improved, with the aim of ideally modelling a real quantum computer in its finer physical details. The measures of algorithmic reliability are to be modified to support larger quantum circuits with favourable scaling in the size of the algorithm.

The research I carried out, more often than not, led to more questions and ideas. For what concerns distributed simulation methods for quantum algorithms, further research directions would include the parallelisation and GPU optimisation of the pathfinding algorithms for tensor network contraction, with the aim of improving overall path quality and reduce contraction times on unbounded problems. This could be of high interest when considering the ever-challenged boundaries of so-called *quantum supremacy*. Conversely, for what concerns sampling QEC codes, Clifford based simulators have proven to be sufficiently performant, with limited margin for improvement. The information extracted from the many QEC-level fault injections performed should be converted, together with the hardware level fault model, in a sound post-QEC logical qubit fault model that takes into account more hardware and algorithmic features. This in turn should open up to the identification of the most critical QEC failures from the perspective of a quantum algorithm running atop the QEC, and better prepare and compensate for those faults. Albeit most of the research ideas prompted by this thesis work lie at the QEC and decoder level, there is still much to be done towards radiation aware compilers, in terms of qubit placement, and radiation aware decoders, in terms of both compensation and reconstruction of information loss at the QEC syndrome level. This in general also encompasses a deeper dive into the vast landscape of QEC codes, especially high density codes, in order to characterise their reliability to radiation events. Hardware based solutions for radiation event resistance should be further optimised to reach case by case cost-effectiveness, ideally leveraging cross-layer design and hardening principles. At last, hybrid quantum-classical algorithms implemented must be reinforced with logical level mitigation components, such as specialised classical post-processing passes.

As the terminus of my doctorate rapidly approaches, despite all the research results hereby presented, quantum fault tolerance yet stands behind a wall of hypotheticals. A wall that now presents a dent on its surface. Whether this will lead to an opening to the other side, or not, eludes my understanding.

Epilogue

This thesis is the product of almost four years of work. It all started from my Master's thesis in February 2022, which evolved into a doctorate proposal in September of the same year. The path I naïvely chose to follow was one of knowledge, fuelled by curiosity.

My illusions of grandeur met a quick fate, as the first year of the doctorate was by far the most challenging. A new city, new university, new people, and the same old me. Struggling to adapt, I held on and waited for it to become routine.

This journey was a test of will and a chance at introspection. I always felt the need to prove that I could achieve ever greater results, reaching higher than the others. This drive towards a metaphorical concept of altitude is frankly amusing, given my dread of heights. After all this time, I came to the realisation that any height is only a matter of perspective. I proved to myself that I could get to the end of it, and that is enough.

To prevent me from severing the bonds with the places and people I hold dear, these last years have seen me go back and forth between my hometown and Trento way more than I would have imagined. This led me to cover more than 43 thousand kilometres, which is about $1.08\times$ the circumference of the Earth. I did not travel *each and every highway*, nonetheless, I managed to do a trip around the world in *my way*. The mesmerising rhythm of this needle, moving so restlessly across time and space, has silently woven new bonds. In trying to strike a delicate balance between two selves, I focused on the sewing thread, and I failed to notice that I had reached the end of the spool.

I have always struggled to accept the end.
The end of playtime, the end of school years, the end of friendships, the end of relationships, the end of life. For most people, an end is a new beginning, whilst I find myself more and more entangled with lost chances as time flows forth. On which end of the seam should I tie the next spool's thread? These consuming choices all lead to similar outcomes, it seems.

I hold no fear for the unknown or in challenging ideals: I merely seek refuge against inevitability. May I laugh at these words, at the end of days, after a life well spent.

Acknowledgements

> *"Sei come una Ferrari senza le ruote!"*
>
> — **Flavio Vella**, trying to encourage me (?)
> (15th of March 2023)

My academic acknowledgements start from my adviser, Professor Flavio Vella, for his restless patience in having borne my natural talent to whine about each and every single bump in my research path, for having shared my first intercontinental flight to the United States, and for having looked past my *Sabaudian* lineage to highlight my talents. Although I feel like we have never fully understood each other, I am grateful for the way our collaboration turned out to be, however odd it might have been.

> *"Come dicevano i romani, il passo più difficile è quello per superare la soglia..."*
>
> — **Paolo Rech**, whilst on a chairlift in Trentino
> (3rd of August 2022)

I want to thank my co-adviser, Professor Paolo Rech, for having convinced me to partake the path of the doctorate, for his abrasive and constructive honesty, for the nice trips around New York and Estonia, and for having mostly always managed to understand what was on my mind, at times when I did not even know myself. Despite his obscure sense of humour, he also gave me a very important teaching: that my future, however perilous or adventurous it might be, lies in my hands and my hands only.

I thank Professor Stefano Cherubin for the talks we had in Krakow at the HiPEAC conference, and for his contribution as chair of the examination committee.
I thank Professor Enrico Blanzieri for his contribution as rapporteur of the examination committee.
I thank Professor Rosa Badia for the time she devised in reviewing this thesis, and for her contribution in quality of external examination committee member.

I thank Professor Robert Wille for having hosted me at the Chair of Design Automation at TUM, in Munich, and for the help he provided in reviewing this thesis.

My personal thanks continue to the wonderful people with whom I had the pleasure to share my working hours, the highs and, more often than not, the lows, of the last three years.

I thank Bresco for his sincerity, his great storytelling skills and his highly contagious laughter. Will I ever find someone else to ask my thoughts on the theory of chaotic attractors first thing in the morning? I doubt it. To the man that has made me aware of so many nerdy and interesting things, and that has shown me the extent of how much truly I still do not know, both scientifically and personally speaking, I utter the word of power: "Boia".

I thank Gioele for his kindness, his determination and his keen attention to details, which so often made me feel understood in my own little obsessions. From the long talks ranging from actual work related conundrums to the small discussions involving more personal matters, he never failed to be there. To the best Geotastic player I know, thank you.

I thank Pico, for having shared with me most of the time of this journey, since day one. He managed to convince me to start rock climbing, and also introduced me to group theory, the former being a much more daunting task than the latter. Most importantly, he taught me by example how to be patient, how to understand perspectives different from my own, and how to share, making me better for it.

I thank Tom for having been the best wingman ever, and having shared so many beers and drinks and laughs together. To *la raclette*, good company, the sleepless nights spent chasing papers and girls, and the Italian-French cuisine brotherhood.

I thank Giulia and Michele for having been the very best first office mates in room 127, at a time when we all knew too little about what would have awaited us in the years to come, and for having shared the burden of my first steps at the climbing gym.

I thank Rawlings and Anish for having welcomed me as an office mate in room 175, right after the first great Ph.D. office migration of 2023. They never held back from sharing tips from their experience as Ph.D. students, helping me understand a world that still seemed so alien, all with a positive attitude and a smile.

I thank Helena, Giorgio, Caterina and Anelia for having helped me and Pico during the second great Ph.D. migration of 2023, by giving us a desk in room 177 and by sheltering us from the cold corridors of Povo 2 with great company, lots of jokes, amazing homemade cake, sweets and licorice.

I thank the craziest office mates I have ever had, following the last great inter-building Ph.D. migration of 2024.

To Anas and Bruno for having helped me correctly weigh the importance of work and to lessen my stress about it.
To Gabriele's bovine-based jokes and gravitational models, Nicola's calmness and (definitely not shady) business management attitude, Tadie's kindness and openness in sharing thrilling tales from the distant Ethiopia, and Matteo's naïveness and bright ideas, the four horsemen of the most blossoming research group I have ever got to meet.
To Costanza, our de facto office mate, and her crazy, energetic and fun personality, keep being awesome.
To Thomas for our shared passion for ice skating and for having been a great desk mate.
To Pedro, for his guitar skills and the beautiful music that he brought from Brazil.
Thank you all for the pranks, the post-lunch Geotastic gaming sessions, the cardboard Frisbee competitions, the supervisor-related memes, the unsolicited help, the time.

I extend my thanks to the colleagues I met at the Chair of Design Automation in Munich and their newfound love for *Nocciolini di Chivasso*. At a time when I was so afraid of being abroad alone, you all managed to make me feel included.
To Jan's high-energy positivity, Tobi's love for cars and jokes, Erik's profound thoughts, and Antonio's unstoppable partying attitude, thank you.

I want to thank those that helped me out from a technical, bureaucratic and at times personal standpoint in the last three years.
To Veronica and Davide, for the constant support in deploying and maintaining the HPC infrastructure that let me run so many simulations.
To Andrea, for having been available to unravel my doubts regarding courses, ETC credits, the *transdisciplinary quantum science and technology* programme and the qualifying exam.
To Danilo, for having been a Piedmontese bastion in foreign ground, and having always stopped by to chat with me, for the help with the HiCrest website and the Departmental Thesis template project.
To Gio, thank you for everything.

These last years saw me become the *far away* friend of some wonderful people, something that enticed me to keep going back and forth just to meet them.
To Riccardo, the best bartender and most sincere friend.
To Valerio, for our shared passion for tech and music, and his unmistakeable honesty.
To Alex, for teaching me to face problems with resolution to readily overcome them.
To Ose, for having saved me from unfortunate situations.

To Lorenzo, for all the Evangelion references and for sharing part of my awkward personality.
To Fonzie, for knowing the true value and meaning of friendship.
To Simone, for the *24/7 Ph.D. support hotline for professional and sentimental burdens*.
To Alessandro, for the shared dream of owning a sports car and our passion for video games.
To Umberto and Giulio for our long-lasting friendship, that still keeps us close 20 years later.
To all my friends, close and distant, thank you for having been there for me, in one way or another.

I want to conclude by thanking my family.
I thank my parents, Silvana and Fabio, for their unwearying presence and love for such a stubborn and indecisive son. For having helped me normalise the many changes that came and went in my life. For having always let me chase my dreams.
I thank my uncle Marco for his genuine interest in my work, for seeking connecting points between my studies and his, and for all the time he spent, together with my dad, to convince me to get on a motorcycle.
I thank my grandmother Liliana for the smiles she gifts to me when I see her, which weigh more than words, and for reminding me that the world needs art just as much as science.
I thank my grandfather Aldo, to whom I dedicated this thesis, for the invaluable wisdom that he always so effortlessly bestows onto me, and for having taught me to never stop learning.

Bibliography

[1] R. P. Feynman. “Simulating physics with computers”. In: *International journal of theoretical physics* (1982) (cit. on p. 1).

[2] D. Horsman, S. Stepney, R. C. Wagner, and V. Kendon. “When does a physical system compute?” In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* (2014) (cit. on p. 1).

[3] P. W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* (1997) (cit. on pp. 1, 32).

[4] L. K. Grover. “A fast quantum mechanical algorithm for database search”. In: *arXiv e-prints* (1996) (cit. on p. 1).

[5] M. E. Beverland et al. *Assessing requirements to scale to practical quantum advantage*. 2022 (cit. on p. 1).

[6] D. Herman et al. *A Survey of Quantum Computing for Finance*. 2022 (cit. on p. 1).

[7] A. Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Communications* (2014) (cit. on pp. 1, 31, 32).

[8] R. Mullin. “Let’s talk about quantum computing in drug discovery”. In: *C&EN Global Enterprise* (2020) (cit. on p. 1).

[9] S. Lloyd, M. Mohseni, and P. Rebentrost. *Quantum algorithms for supervised and unsupervised machine learning*. 2013 (cit. on p. 1).

[10] H.-Y. Huang et al. “Power of data in quantum machine learning”. In: *Nature Communications* (2021) (cit. on pp. 1, 10).

[11] E. Bernstein and U. Vazirani. “Quantum Complexity Theory”. In: *SIAM Journal on Computing* (1997) (cit. on pp. 1, 31, 32).

[12] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010 (cit. on pp. 1, 5, 6, 10).

[13] I. L. Markov and Y. Shi. “Simulating Quantum Computation by Contracting Tensor Networks”. In: *SIAM Journal on Computing* (2008) (cit. on pp. 1, 9, 28).

[14] J. Biamonte and V. Bergholm. *Tensor Networks in a Nutshell*. 2017 (cit. on pp. 1, 9, 29).

[15] G. F. Viamontes. “Efficient quantum circuit simulation”. PhD thesis. USA: University of Michigan, United States, 2007 (cit. on pp. 1, 8).

[16] R. Wille, S. Hillmich, and L. Burgholzer. “Decision Diagrams for Quantum Computing”. In: *Design Automation of Quantum Computers*. Ed. by R. O. Topaloglu. Cham: Springer International Publishing, 2023 (cit. on p. 1).

[17] D. Garcí a-Martín and G. Sierra. “Five Experimental Tests on the 5-Qubit IBM Quantum Computer”. In: *Journal of Applied Mathematics and Physics* (2018) (cit. on p. 2).

[18] F. Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* (2019) (cit. on pp. 2, 12, 27, 31).

[19] J. Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* (2018) (cit. on pp. 2, 12, 13, 54).

[20] S. J. Devitt, W. J. Munro, and K. Nemoto. “Quantum error correction for beginners”. In: *Reports on Progress in Physics* (2013) (cit. on p. 2).

[21] R. Acharya et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: *Nature* (2023) (cit. on pp. 2, 14, 16, 19, 52, 55–57, 71, 95).

[22] A. M. Steane. “Simple quantum error-correcting codes”. In: *Physical Review A* (1996) (cit. on pp. 2, 16).

[23] C. D. Wilen et al. “Correlated charge noise and relaxation errors in superconducting qubits”. In: *Nature* (2021) (cit. on pp. 2, 16, 18, 52, 55, 102, 121, 133).

[24] A. P. Vepsäläinen et al. “Impact of ionizing radiation on superconducting qubit coherence”. In: *Nature* (2020) (cit. on pp. 2, 16, 18, 52, 54, 99, 133).

[25] A. Einstein, B. Podolsky, and N. Rosen. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” In: *Physical Review* (1935) (cit. on p. 3).

[26] J. S. Bell. “On the Einstein Podolsky Rosen paradox”. In: *Physics Physique Fizika* (3 1964) (cit. on p. 3).

[27] G. Vidal. “Efficient Classical Simulation of Slightly Entangled Quantum Computations”. In: *Physical Review Letters* (2003) (cit. on p. 4).

[28] K. C. Young. *C191-Lectures*. 2014 (cit. on p. 4).

[29] M. Born. “On the Quantum Mechanics of Collision Processes”. In: *Zeitschrift für Physik* (1926) (cit. on p. 4).

[30] B. Coecke and R. Duncan. “Interacting quantum observables: categorical algebra and diagrammatics”. In: *New Journal of Physics* (2011) (cit. on p. 6).

[31] G. Li et al. “On the Co-Design of Quantum Software and Hardware”. In: *Proceedings of the Eight Annual ACM International Conference on Nanoscale Computing and Communication*. Virtual Event, Italy: Association for Computing Machinery, 2021 (cit. on pp. 7, 133).

[32] Top500. *El Capitan achieves top spot, Frontier and Aurora follow behind*. Top500, 2024 (cit. on p. 8).

[33] T. S. Humble et al. “Quantum Computers for High-Performance Computing”. In: *IEEE Micro* (2021) (cit. on p. 8).

[34] N. IBM. *RIKEN Selects IBMs Next Generation Quantum System to be Integrated with the Supercomputer Fugaku*. 2024 (cit. on p. 8).

[35] T. Hoefler, T. Häner, and M. Troyer. “Disentangling Hype from Practicality: On Realistically Achieving Quantum Advantage”. In: *Commun. ACM* (2023) (cit. on p. 8).

[36] H. Bayraktar et al. “cuQuantum SDK: A High-Performance Library for Accelerating Quantum Science”. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2023 (cit. on pp. 8–10, 24, 27, 33, 39).

[37] Google Quantum AI team and collaborators. *qsim*. 2020 (cit. on p. 8).

[38] G. Aleksandrowicz et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019 (cit. on pp. 8, 66).

[39] V. Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2022 (cit. on pp. 8, 122).

[40] X. Xu, S. Benjamin, J. Sun, X. Yuan, and P. Zhang. *A Herculean task: Classical simulation of quantum computers*. 2023 (cit. on p. 8).

[41] A. Ahmadzadeh and H. Sarbazi-Azad. “Fast scalable and low-power quantum circuit simulation on the cluster of GPUs platforms”. In: *Optical and Quantum Electronics* (2024) (cit. on pp. 8, 72).

[42] A. Kubicek, A. Stratikopoulos, J. Fumero, N. Foutris, and C. Kotselidis. *TornadoQSim: An Open-source High-Performance and Modular Quantum Circuit Simulation Framework*. 2023 (cit. on p. 9).

[43] X.-C. Wu et al. “Full-State Quantum Circuit Simulation by Using Data Compression”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Denver, Colorado: Association for Computing Machinery, 2019 (cit. on pp. 9, 25, 46).

[44] B. Fang, M. Y. Özkaya, A. Li, Ü. V. Çatalyürek, and S. Krishnamoorthy. *Efficient Hierarchical State Vector Simulation of Quantum Circuits via Acyclic Graph Partitioning*. 2022 (cit. on pp. 9, 25, 26).

[45] J. Gray and S. Kourtis. “Hyper-optimized tensor network contraction”. In: *Quantum* (2021) (cit. on pp. 10, 25, 28, 36, 39, 44).

[46] C. Ibrahim, D. Lykov, Z. He, Y. Alexeev, and I. Safro. *Constructing Optimal Contraction Trees for Tensor Network Quantum Circuit Simulation*. 2022 (cit. on p. 10).

[47] T. Vincent et al. “Jet: Fast quantum circuit simulations with parallel task-based tensor-network contraction”. In: *Quantum* (2022) (cit. on p. 10).

[48] R. Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* (2014) (cit. on p. 10).

[49] J. Brennan et al. *Tensor Network Circuit Simulation at Exascale*. 2021 (cit. on p. 10).

[50] F. Pan and P. Zhang. “Simulation of Quantum Circuits Using the Big-Batch Tensor Network Method”. In: *Phys. Rev. Lett.* (3 2022) (cit. on pp. 10, 28).

[51] P. Seitz, I. Medina, E. Cruz, Q. Huang, and C. B. Mendl. “Simulating quantum circuits using tree tensor networks”. In: *Quantum* (2023) (cit. on p. 10).

[52] M. Ballarin. “Quantum Computer Simulation via Tensor Networks”. MA thesis. Università degli Studi di Padova, 2022 (cit. on p. 10).

[53] T. Nguyen et al. *Tensor Network Quantum Virtual Machine for Simulating Quantum Circuits at Exascale*. 2021 (cit. on p. 10).

[54] C. Gidney. “Stim: a fast stabilizer circuit simulator”. In: *Quantum* (2021) (cit. on pp. 10, 72, 98).

[55] J. Doi, H. Horii, and C. Wood. *Efficient techniques to GPU Accelerations of Multi-Shot Quantum Computing Simulations*. 2023 (cit. on p. 10).

[56] E. Gutiérrez, S. Romero, M. A. Trenas, and E. L. Zapata. “Quantum computer simulation using the CUDA programming model”. In: *Computer Physics Communications* (2010) (cit. on p. 10).

[57] A. Amariutei and S. Caraiman. “Parallel quantum computer simulation on the GPU”. In: *15th International Conference on System Theory, Control and Computing*. 2011 (cit. on p. 10).

[58] A. Avila, A. Maron, R. Reiser, M. Pilla, and A. Yamin. "GPU-aware distributed quantum simulation". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. Gyeongju, Republic of Korea: Association for Computing Machinery, 2014 (cit. on p. 10).

[59] S. Heng, T. Kim, and Y. Han. "Exploiting GPU-based Parallelism for Quantum Computer Simulation: A Survey". In: *IEIE Transactions on Smart Processing & Computing* (2020) (cit. on p. 10).

[60] A. J. Gangapuram, A. M. Läuchli, and C. Hempel. "Benchmarking quantum computer simulation software packages: State vector simulators". In: *SciPost Phys. Core* (2024) (cit. on pp. 10, 27).

[61] J. Faj, I. Peng, J. Wahlgren, and S. Markidis. "Quantum Computer Simulations at Warp Speed: Assessing the Impact of GPU Acceleration: A Case Study with IBM Qiskit Aer, Nvidia Thrust and cuQuantum". In: *2023 IEEE 19th International Conference on e-Science (e-Science)*. 2023 (cit. on pp. 10, 41).

[62] J. Doi, H. Takahashi, R. Raymond, T. Imamichi, and H. Horii. "Quantum computing simulator on a heterogenous HPC system". In: *Proceedings of the 16th ACM International Conference on Computing Frontiers*. Alghero, Italy: Association for Computing Machinery, 2019 (cit. on p. 10).

[63] C. Jiao, W. Zhang, and L. Shen. "Communication Optimizations for State-vector Quantum Simulator on CPU+GPU Clusters". In: *Proceedings of the 52nd International Conference on Parallel Processing*. Salt Lake City, UT, USA: Association for Computing Machinery, 2023 (cit. on p. 10).

[64] D. Lykov, R. Shaydulin, Y. Sun, Y. Alexeev, and M. Pistoia. "Fast Simulation of High-Depth QAOA Circuits". In: *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*. Denver, CO, USA: Association for Computing Machinery, 2023 (cit. on pp. 10, 27).

[65] F. Pan, H. Gu, L. Kuang, B. Liu, and P. Zhang. "Efficient Quantum Circuit Simulation by Tensor Network Methods on Modern GPUs". In: *ACM Transactions on Quantum Computing* (2024) (cit. on p. 10).

[66] T. Tomesh et al. *SupermarQ: A Scalable Quantum Benchmark Suite*. 2022 (cit. on pp. 10, 29, 31, 32).

[67] A. Li, S. Stein, S. Krishnamoorthy, and J. Ang. "QASMBench: A Low-Level Quantum Benchmark Suite for NISQ Evaluation and Simulation". In: *ACM Transactions on Quantum Computing* (2023) (cit. on pp. 10, 29, 30, 133).

[68] G. Wendin. "Quantum information processing with superconducting circuits: a review". In: *Reports on Progress in Physics* (2017) (cit. on pp. 11, 13).

[69] P. Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied Physics Reviews* (2019) (cit. on p. 11).

[70] IBM. *IBM's quantum roadmap* (cit. on p. 12).

[71] Y. Kim et al. "Evidence for the utility of quantum computing before fault tolerance". In: *Nature* (2023) (cit. on p. 12).

[72] A. Anferov, S. P. Harvey, F. Wan, J. Simon, and D. I. Schuster. "Superconducting Qubits above 20 GHz Operating over 200 mK". In: *PRX Quantum* (3 2024) (cit. on p. 12).

[73] W. G. Unruh. "Maintaining coherence in quantum computers". In: *Phys. Rev. A* (2 1995) (cit. on p. 12).

[74] D. P. DiVincenzo and D. Loss. "Quantum computers and quantum coherence". In: *Journal of Magnetism and Magnetic Materials* (1999) (cit. on p. 12).

[75] R. Stassi, M. Cirio, and F. Nori. "Scalable quantum computer with superconducting circuits in the ultrastrong coupling regime". In: *npj Quantum Information* (2020) (cit. on p. 13).

[76] C. Wang et al. "Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds". In: *npj Quantum Information* (2022) (cit. on p. 13).

[77] A. Somoroff et al. "Millisecond Coherence in a Superconducting Qubit". In: *Phys. Rev. Lett.* (26 2023) (cit. on pp. 13, 54).

[78] J. Ghosh et al. "High-fidelity controlled-σ^Z gate for resonator-based superconducting quantum computers". In: *Phys. Rev. A* (2 2013) (cit. on p. 13).

[79] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen. "Gate-error analysis in simulations of quantum computers with transmon qubits". In: *Phys. Rev. A* (6 2017) (cit. on p. 13).

[80] M. A. Rol et al. "Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits". In: *Physical Review Letters* (2019) (cit. on p. 13).

[81] Y. Kim et al. "High-fidelity three-qubit iToffoli gate for fixed-frequency superconducting qubits". In: *Nature Physics* (2022) (cit. on p. 13).

[82] M. AbuGhanem and H. Eleuch. "Two-qubit entangling gates for superconducting quantum computers". In: *Results in Physics* (2024) (cit. on p. 13).

[83] J. M. Gambetta, J. M. Chow, and M. Steffen. "Building logical qubits in a superconducting quantum computing system". In: *npj Quantum Information* (2017) (cit. on p. 13).

[84] M. Vischi, L. Ferialdi, A. Trombettoni, and A. Bassi. "Possible limits on superconducting quantum computers from spontaneous wave-function collapse models". In: *Phys. Rev. B* (17 2022) (cit. on p. 13).

[85] K. Georgopoulos, C. Emary, and P. Zuliani. “Modeling and simulating the noisy behavior of near-term quantum computers”. In: *Phys. Rev. A* (6 2021) (cit. on pp. 13, 53, 76).

[86] C. Gidney, M. Newman, and M. McEwen. “Benchmarking the Planar Honeycomb Code”. In: *Quantum* (2022) (cit. on pp. 14, 76, 98, 102).

[87] J. Preskill. “Reliable quantum computers”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* (1998) (cit. on p. 14).

[88] R. Van Meter and S. J. Devitt. “The Path to Scalable Distributed Quantum Computing”. In: *Computer* (2016) (cit. on p. 14).

[89] E. A. Sete, W. J. Zeng, and C. T. Rigetti. “A functional architecture for scalable quantum computing”. In: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. 2016 (cit. on p. 14).

[90] D. Copsey et al. “Toward a scalable, silicon-based quantum computing architecture”. In: *IEEE Journal of Selected Topics in Quantum Electronics* (2003) (cit. on p. 14).

[91] D. Oliveira et al. “A Systematic Methodology to Compute the Quantum Vulnerability Factors for Quantum Circuits”. In: *IEEE Transactions on Dependable and Secure Computing* (2024) (cit. on pp. 14, 16, 52, 122).

[92] D. Oliveira et al. “QuFI: a Quantum Fault Injector to Measure the Reliability of Qubits and Quantum Circuits”. In: *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2022 (cit. on pp. 14, 16, 52, 114, 121, 124, 133).

[93] Z. Hu et al. *Toward Consistent High-fidelity Quantum Learning on Unstable Devices via Efficient In-situ Calibration*. 2023 (cit. on pp. 14, 16).

[94] S. C. Smith, B. J. Brown, and S. D. Bartlett. *Mitigating errors in logical qubits*. 2024 (cit. on pp. 14, 16).

[95] H. Ali et al. “Reducing the error rate of a superconducting logical qubit using analog readout information”. In: *Phys. Rev. Appl.* (4 2024) (cit. on pp. 14, 16).

[96] A. Chatterjee, K. Phalak, and S. Ghosh. “Quantum Error Correction For Dummies”. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2023 (cit. on p. 14).

[97] S. Bravyi, M. Englbrecht, R. König, and N. Peard. “Correcting coherent errors with surface codes”. In: *npj Quantum Information* (2018) (cit. on p. 14).

[98] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown. “The XZZX surface code”. In: *Nature Communications* (2021) (cit. on pp. 14–16, 58, 95).

[99] C. K. Andersen et al. “Repeated quantum error detection in a surface code”. In: *Nature Physics* (2020) (cit. on pp. 14–16).

[100] J. F. Marques et al. “Logical-qubit operations in an error-detecting surface code”. In: *Nature Physics* (2022) (cit. on p. 14).

[101] R. Acharya et al. “Quantum error correction below the surface code threshold”. In: *Nature* (2024) (cit. on pp. 14, 16, 19, 71, 74, 75, 95).

[102] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* (1982) (cit. on p. 14).

[103] D. Gottesman. *Stabilizer Codes and Quantum Error Correction*. 1997 (cit. on p. 14).

[104] A. Javadi-Abhari et al. “Optimized surface code communication in superconducting quantum computers”. In: *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. Cambridge, Massachusetts: Association for Computing Machinery, 2017 (cit. on pp. 15, 16).

[105] S. Krinner et al. “Realizing repeated quantum error correction in a distance-three surface code”. In: *Nature* (2022) (cit. on pp. 15, 16).

[106] A. Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* (2003) (cit. on pp. 15, 58).

[107] P. W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* (4 1995) (cit. on p. 15).

[108] S. B. Bravyi and A. Y. Kitaev. “Quantum codes on a lattice with boundary”. In: (1998) (cit. on p. 15).

[109] O. Higgott. “PyMatching: A Python Package for Decoding Quantum Codes with Minimum-Weight Perfect Matching”. In: *ACM Transactions on Quantum Computing* (2022) (cit. on pp. 15, 74, 86, 95, 102).

[110] O. Higgott and C. Gidney. “Sparse Blossom: correcting a million errors per core second with minimum-weight matching”. In: *Quantum* (2025) (cit. on pp. 15, 16, 59, 61, 95).

[111] B. J. Brown. “Conservation Laws and Quantum Error Correction: Toward a Generalized Matching Decoder”. In: *IEEE BITS the Information Theory Magazine* (2022) (cit. on p. 16).

[112] A. Márton and J. K. Asbóth. “Coherent errors and readout errors in the surface code”. In: *Quantum* (2023) (cit. on p. 16).

[113] S. Vittal, P. Das, and M. Qureshi. “Astrea: Accurate Quantum Error-Decoding via Practical Minimum-Weight Perfect-Matching”. In: *Proceedings of the 50th Annual International Symposium on Computer Architecture*. Orlando, FL, USA: Association for Computing Machinery, 2023 (cit. on p. 16).

[114] N. Sundaresan et al. “Demonstrating multi-round subsystem quantum error correction using matching and maximum likelihood decoders”. In: *Nature Communications* (2023) (cit. on p. 16).

[115] S. Bravyi, M. Suchara, and A. Vargo. “Efficient algorithms for maximum likelihood decoding in the surface code”. In: *Phys. Rev. A* (3 2014) (cit. on p. 16).

[116] N. Delfosse and N. H. Nickerson. “Almost-linear time decoding algorithm for topological codes”. In: *Quantum* (2021) (cit. on pp. 16, 74).

[117] J. Old and M. Rispler. “Generalized Belief Propagation Algorithms for Decoding of Surface Codes”. In: *Quantum* (2023) (cit. on p. 16).

[118] S. Varsamopoulos, B. Criger, and K. Bertels. “Decoding small surface codes with feedforward neural networks”. In: *Quantum Science and Technology* (2017) (cit. on p. 16).

[119] H. Goto, Y. Ho, and T. Kanao. “Measurement-free fault-tolerant logical-zero-state encoding of the distance-three nine-qubit surface code in a one-dimensional qubit array”. In: *Phys. Rev. Res.* (4 2023) (cit. on p. 16).

[120] K. Tiurev, P.-J. H. S. Derks, J. Roffe, J. Eisert, and J.-M. Reiner. “Correcting non-independent and non-identically distributed errors with surface codes”. In: *Quantum* (2023) (cit. on p. 16).

[121] M. Katsuda, K. Mitarai, and K. Fujii. “Simulation and performance analysis of quantum error correction with a rotated surface code under a realistic noise model”. In: *Phys. Rev. Res.* (1 2024) (cit. on p. 16).

[122] A. G. Fowler. “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O(1) parallel time”. In: *Quantum Info. Comput.* (2015) (cit. on pp. 16, 74).

[123] P. Das et al. *A Scalable Decoder Micro-architecture for Fault-Tolerant Quantum Computing*. 2020 (cit. on p. 16).

[124] B. Barber et al. “A real-time, scalable, fast and highly resource efficient decoder for a quantum computer”. In: *Nature Electronics* (2025) (cit. on pp. 16, 74).

[125] R. Barends et al. “Minimizing quasiparticle generation from stray infrared light in superconducting quantum circuits”. In: *Applied Physics Letters* (2011) (cit. on pp. 16, 52).

[126] M. McEwen et al. “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits”. In: *Nature Physics* (2022) (cit. on pp. 16, 19, 54, 55, 99, 118, 131, 133).

[127] A. D. Corcoles et al. “Protecting superconducting qubits from radiation”. In: *Applied Physics Letters* (2011) (cit. on pp. 16, 52, 133).

[128] L. Grünhaupt et al. “Loss Mechanisms and Quasiparticle Dynamics in Superconducting Microwave Resonators Made of Thin-Film Granular Aluminum”. In: *Phys. Rev. Lett.* (11 2018) (cit. on pp. 16, 52, 133).

[129] L. Cardani et al. “Reducing the impact of radioactivity on quantum circuits in a deep-underground facility”. In: *Nature Communications* (2021) (cit. on pp. 16, 18, 19, 52, 96, 121, 131, 133).

[130] J. M. Martinis. “Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays”. In: *npj Quantum Information* (2021) (cit. on pp. 16, 19, 52, 93, 96, 133).

[131] Z. Chen et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* (2021) (cit. on pp. 16, 52, 54, 57, 60, 61, 133).

[132] P. M. Harrington et al. *Synchronous Detection of Cosmic Rays and Correlated Errors in Superconducting Qubit Arrays*. 2024 (cit. on pp. 16, 54, 66).

[133] R. Baumann. “Soft errors in advanced computer systems”. In: *IEEE Design Test of Computers* (2005) (cit. on pp. 17, 18, 54, 131).

[134] D. Oliveira, E. Auden, and P. Rech. “Atmospheric Neutron-Induced Fault Generation and Propagation in Quantum Bits and Quantum Circuits”. In: *IEEE Transactions on Nuclear Science* (2023) (cit. on pp. 17, 18, 55, 118).

[135] L. Cardani et al. “Disentangling the sources of ionizing radiation in superconducting qubits”. In: *The European Physical Journal C* (2023) (cit. on pp. 17, 19, 96, 102, 121).

[136] E. Auden and P. Rech. “Single-Event Effects in Neutron-Irradiated High-Temperature DC Superconducting Quantum Interference Devices”. Kansas City, MO, USA, 2023 (cit. on p. 17).

[137] E. Yelton et al. “Modeling phonon-mediated quasiparticle poisoning in superconducting qubit arrays”. In: *Phys. Rev. B* (2 2024) (cit. on pp. 17, 54, 102).

[138] D. A. G. D. Oliveira et al. “Radiation-Induced Error Criticality in Modern HPC Parallel Accelerators”. In: *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 2017 (cit. on p. 18).

[139] P. Rech et al. “Measuring the Radiation Reliability of SRAM Structures in GPUS Designed for HPC”. In: *IEEE 10th Workshop on Silicon Errors in Logic - System Effects (SELSE)*. 2014 (cit. on p. 18).

[140] C. Slayman. “JEDEC Standards on Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Errors”. In: *Soft Errors in Modern Electronic Systems*. Boston, MA: Springer US, 2011 (cit. on p. 18).

[141] D. Tiwari et al. “Understanding GPU errors on large-scale HPC systems and the implications for system design and operation”. In: *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. 2015 (cit. on p. 19).

[142] F. D. Dominicis et al. “Evaluating radiation impact on transmon qubits in above and underground facilities”. In: *arXiv* (2024) (cit. on pp. 19, 96).

[143] G. Bratrud et al. *First Measurement of Correlated Charge Noise in Superconducting Qubits at an Underground Facility*. 2024 (cit. on p. 19).

[144] B. Loer et al. *Abatement of Ionizing Radiation for Superconducting Quantum Devices*. 2024 (cit. on pp. 19, 96).

[145] L. Hesla. *Fermilab is partner in Quantum Science Center based at Oak Ridge National Laboratory*. en-US. 2020. (Visited on 2022-11-29) (cit. on p. 19).

[146] G. S. Ravi et al. “Better Than Worst-Case Decoding for Quantum Error Correction”. In: *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. Vancouver, BC, Canada: Association for Computing Machinery, 2023 (cit. on pp. 19, 52).

[147] V. V. Sivak et al. “Real-time quantum error correction beyond break-even”. In: *Nature* (2023) (cit. on pp. 19, 52).

[148] C. Jünger et al. “Implementation of scalable suspended superinductors”. In: *Applied Physics Letters* (2025) (cit. on pp. 19, 96).

[149] E. T. Mannila et al. “A superconductor free of quasiparticles for seconds”. In: *Nature Physics* (2022) (cit. on pp. 19, 95).

[150] X.-G. Li et al. *Direct evidence for cosmic-ray-induced correlated errors in superconducting qubit array*. 2024 (cit. on pp. 19, 52, 66).

[151] M. McEwen et al. *Resisting high-energy impact events through gap engineering in superconducting qubit arrays*. 2024 (cit. on pp. 19, 52, 66, 95, 96).

[152] V. Iaia et al. “Phonon downconversion to suppress correlated errors in superconducting qubits”. In: *Nature Communications* (2022) (cit. on pp. 19, 52, 66, 93, 96, 97).

[153] G. Calusine et al. “Analysis and mitigation of interface losses in trenched superconducting coplanar waveguide resonators”. In: *Applied Physics Letters* (2018) (cit. on pp. 19, 96).

[154] C. E. Murray. “Analytical Modeling of Participation Reduction in Superconducting Coplanar Resonator and Qubit Designs Through Substrate Trenching”. In: *IEEE Transactions on Microwave Theory and Techniques* (2020) (cit. on pp. 19, 96).

[155] F. Henriques et al. “Phonon traps reduce the quasiparticle density in superconducting circuits”. In: *Applied Physics Letters* (2019) (cit. on pp. 19, 96).

[156] X. Pan et al. “Engineering superconducting qubits to reduce quasiparticles and charge noise”. In: *Nature Communications* (2022) (cit. on pp. 19, 96).

[157] Q. Xu et al. “Distributed quantum error correction for chip-level catastrophic errors”. In: *arXiv preprint arXiv:2203.16488* (2022) (cit. on p. 19).

[158] M. Vallero, P. Rech, and F. Vella. “State of practice: Evaluating GPU performance of state vector and tensor network methods”. In: *Future Generation Computer Systems* (2025) (cit. on pp. 23, 72, 153).

[159] G. Kalachev, P. Panteleev, and M.-H. Yung. *Multi-Tensor Contraction for XEB Verification of Quantum Circuits*. 2022 (cit. on pp. 26, 29).

[160] C. Guo, Y. Zhao, and H.-L. Huang. “Verifying Random Quantum Circuits with Arbitrary Geometry Using Tensor Network States Algorithm”. In: *Phys. Rev. Lett.* (7 2021) (cit. on p. 26).

[161] Y. Liu et al. “Verifying Quantum Advantage Experiments with Multiple Amplitude Tensor Network Contraction”. In: *Physical Review Letters* (2024) (cit. on p. 26).

[162] Planck Collaboration et al. “Planck 2015 results. XIII. Cosmological parameters”. In: *Astronomy and Astrophysics* (2016) (cit. on p. 27).

[163] J. Ha, J. Lee, and J. Heo. “Resource analysis of quantum computing with noisy qubits for Shor’s factoring algorithms”. In: *Quantum Information Processing* (2022) (cit. on p. 27).

[164] M. Morita, Y. Tomita, J. Koyama, and K. Kimura. “Simulator Demonstration of Large Scale Variational Quantum Algorithm on HPC Cluster”. In: *IEEE Access* (2024) (cit. on p. 27).

[165] T. Jones, B. Koczor, and S. C. Benjamin. *Distributed Simulation of Statevectors and Density Matrices*. 2023 (cit. on p. 27).

[166] Y. Kimura, S. Li, H. Sato, and M. Fujita. “Decision Diagram vs. State Vector: A Comparative Study on Quantum Computing Simulation Efficiency”. In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2024 (cit. on p. 27).

[167] M. X. Goemans and D. P. Williamson. “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming”. In: *J. ACM* (1995) (cit. on p. 28).

[168] R. Schutski, T. Khakhulin, I. Oseledets, and D. Kolmakov. “Simple heuristics for efficient parallel tensor contraction and quantum circuit simulation”. In: *Phys. Rev. A* (6 2020) (cit. on p. 28).

[169] E. Pednault et al. “Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits”. In: *Lawrence Livermore National Laboratory Journal* (2017) (cit. on p. 29).

[170] J. Biamonte. *Lectures on Quantum Tensor Networks*. 2020 (cit. on p. 29).

[171] E. Farhi, J. Goldstone, and S. Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014 (cit. on p. 31).

[172] A. Y. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem*. 1995 (cit. on pp. 31, 32).

[173] D. Coppersmith. *An approximate Fourier transform useful in quantum factoring*. 2002 (cit. on pp. 31, 32).

[174] W. van Dam, S. Hallgren, and L. Ip. *Quantum Algorithms for some Hidden Shift Problems*. 2002 (cit. on pp. 31, 32).

[175] I. D. Kivlichan et al. “Quantum Simulation of Electronic Structure with Linear Depth and Connectivity”. In: *Physical Review Letters* (2018) (cit. on p. 31).

[176] E. Pednault et al. *Pareto-Efficient Quantum Circuit Simulation Using Tensor Contraction Deferral*. 2020 (cit. on p. 47).

[177] M. Vallero, G. Casagranda, F. Vella, and P. Rech. “On the Efficacy of Surface Codes in Compensating for Radiation Events in Superconducting Devices”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (2024) (cit. on pp. 51, 99, 102, 106, 153).

[178] J. M. Auger, H. Anwar, M. Gimeno-Segovia, T. M. Stace, and D. E. Browne. “Fault-tolerance thresholds for the surface code with fabrication errors”. In: *Phys. Rev. A* (4 2017) (cit. on p. 52).

[179] A. Siegel, A. Strikis, T. Flatters, and S. Benjamin. “Adaptive surface code for quantum error correction in the presence of temporary or permanent defects”. In: *Quantum* (2023) (cit. on p. 52).

[180] S. Ganjam et al. *Surpassing millisecond coherence times in on-chip superconducting quantum memories by optimizing materials, processes, and circuit design*. 2023 (cit. on p. 54).

[181] S. F. Lin et al. *Codesign of quantum error-correcting codes and modular chiplets in the presence of defects*. 2024 (cit. on pp. 55, 56).

[182] Y. Xiao, B. Srivastava, and M. Granath. *Exact results on finite size corrections for surface codes tailored to biased noise*. 2024 (cit. on pp. 55, 56, 58).

[183] M. P. Stafford and N. C. Menicucci. “Biased Gottesman-Kitaev-Preskill repetition code”. In: *Phys. Rev. A* (5 2023) (cit. on p. 56).

[184] D. Forlivesi, L. Valentini, and M. Chiani. *Logical Error Rates of XZZX and Rotated Quantum Surface Codes*. 2023 (cit. on pp. 56, 58).

[185] S. Jha, A. Ebrahimi, and J. Gong. *Qiskit Topological Codes*. `https://github.com/yaleqc/qtcodes`. 2022 (cit. on p. 56).

[186] J. R. Wootton. “Benchmarking near-term devices with quantum error correction”. In: *Quantum Science and Technology* (2020) (cit. on p. 57).

[187] M. Vallero, G. Casagranda, F. Vella, and P. Rech. “Radiation-Induced Fault Detection in Superconducting Quantum Devices”. In: *Advanced Quantum Technologies* (2026) (cit. on pp. 71, 99, 102, 153).

[188] A. Leonteva, G. Masella, M. Outteryck, A. P. Orioli, and S. Whitlock. *Comparative Benchmarking of Utility-Scale Quantum Emulators*. 2025 (cit. on p. 72).

[189] H. Bombin and M. A. Martin-Delgado. “Optimal resources for topological two-dimensional stabilizer codes: Comparative study”. In: *Physical Review A* (2007) (cit. on p. 73).

[190] A. A. Kovalev and L. P. Pryadko. “Improved quantum hypergraph-product LDPC codes”. In: *2012 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2012 (cit. on p. 73).

[191] J. Bausch et al. “Learning high-accuracy error decoding for quantum processors”. In: *Nature* (2024) (cit. on p. 74).

[192] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert. “Reinforcement learning decoders for fault-tolerant quantum computation”. In: *Machine Learning: Science and Technology* (2020) (cit. on p. 74).

[193] A. J. Ferris and D. Poulin. “Tensor Networks and Quantum Error Correction”. In: *Phys. Rev. Lett.* (3 2014) (cit. on p. 74).

[194] C. T. Chubb. *General tensor network decoding of 2D Pauli codes*. 2021 (cit. on p. 74).

[195] F. Battistel et al. “Real-time decoding for fault-tolerant quantum computing: progress, challenges and outlook”. In: *Nano Futures* (2023) (cit. on p. 74).

[196] J. Edmonds. “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* (1965) (cit. on p. 74).

[197] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell. *Improved decoding of circuit noise and fragile boundaries of tailored surface codes*. 2023 (cit. on p. 74).

[198] N. Delfosse, V. Londe, and M. E. Beverland. “Toward a Union-Find Decoder for Quantum LDPC Codes”. In: *IEEE Transactions on Information Theory* (2022) (cit. on p. 74).

[199] J. Roffe. *LDPC: Python tools for low density parity check codes*. 2022 (cit. on p. 74).

[200] S. Bravyi et al. “High-threshold and low-overhead fault-tolerant quantum memory”. In: *Nature* (2024) (cit. on p. 75).

[201] H.-T. Liu et al. “Direct Implementation of High-Fidelity Three-Qubit Gates for Superconducting Processor with Tunable Couplers”. In: *Phys. Rev. Lett.* (5 2025) (cit. on p. 75).

[202] Y. Zhao et al. “Realization of an Error-Correcting Surface Code with Superconducting Qubits”. In: *Phys. Rev. Lett.* (3 2022) (cit. on p. 75).

[203] Z. Chen et al. “Efficient implementation of arbitrary two-qubit gates using unified control”. In: *Nature Physics* (2025) (cit. on p. 75).

[204] K. Kubo and H. Goto. “Fast parametric two-qubit gate for highly detuned fixed-frequency superconducting qubits using a double-transmon coupler”. In: *Applied Physics Letters* (2023) (cit. on p. 75).

[205] M. Vallero, G. Casagranda, F. Vella, and P. Rech. “TETRIS-Q: Tiling-based Effective Transient-fault Reduction on Interleaved Superconducting Qubits”. In: *review* (2026) (cit. on pp. 93, 153).

[206] J. J. Wesdorp et al. *Mitigating crosstalk errors for simultaneous single-qubit gates on a superconducting quantum processor*. 2026 (cit. on p. 95).

[207] H. D. Pinckney et al. *Characterization of Radiation-Induced Errors in Superconducting Qubits Protected with Various Gap-Engineering Strategies*. 2026 (cit. on p. 95).

[208] H. P. Binney et al. *Distinguishing types of correlated errors in superconducting qubits*. 2026 (cit. on p. 95).

[209] P. Kamenov, T. DiNapoli, M. Gershenson, and S. Chakram. *Suppression of quasiparticle poisoning in transmon qubits by gap engineering*. 2024 (cit. on p. 95).

[210] V. D. Kurilovich et al. *Correlated Error Bursts in a Gap-Engineered Superconducting Qubit Array*. 2025 (cit. on p. 95).

[211] A. Bargerbos et al. “Mitigation of Quasiparticle Loss in Superconducting Qubits by Phonon Scattering”. In: *Phys. Rev. Appl.* (2 2023) (cit. on pp. 96, 97).

[212] M. Odeh et al. *Non-Markovian dynamics of a superconducting qubit in a phononic bandgap*. 2023 (cit. on p. 97).

[213] Y. J. Rosen et al. “Protecting superconducting qubits from phonon mediated decay”. In: *Applied Physics Letters* (2019) (cit. on p. 97).

[214] G. Casagranda, M. Vallero, F. Vella, and P. Rech. “Understanding the Contributions of Terrestrial Radiation Sources to Error Rates in Quantum Devices”. In: *IEEE Transactions on Nuclear Science* (2025) (cit. on pp. 102, 153).

[215] G. Casagranda et al. “SQUID G.A.M.E.: Gamma, Atmospheric, and Mono-Energetic Neutron Effects on Quantum Devices”. In: *IEEE Transactions on Nuclear Science* (2026) (cit. on pp. 102, 153).

[216] M. Vallero, E. Dri, E. Giusto, B. Montrucchio, and P. Rech. “Understanding Logical-Shift Error Propagation in Quanvolutional Neural Networks”. In: *IEEE Transactions on Quantum Engineering* (2024) (cit. on pp. 113, 153).

[217] M. Henderson, S. Shakya, S. Pradhan, and T. Cook. “Quanvolutional neural networks: powering image recognition with quantum circuits”. In: *Quantum Machine Intelligence* (2020) (cit. on pp. 113–120, 133).

[218] K. Sooksatra, P. Rivas, and J. Orduz. “Evaluating Accuracy and Adversarial Robustness of Quanvolutional Neural Networks”. In: *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2021 (cit. on pp. 113, 115).

[219] M. Schuld, I. Sinayskiy, and F. Petruccione. “The quest for a Quantum Neural Network”. In: *Quantum Information Processing* (2014) (cit. on pp. 113, 115, 117).

[220] T. Sathya and S. Sudha. “OQCNN: optimal quantum convolutional neural network for classification of facial expression”. In: *Neural Computing and Applications* (2023) (cit. on pp. 113, 117, 133).

[221] S.-Y. Huang, W.-J. An, D.-S. Zhang, and N.-R. Zhou. “Image classification and adversarial robustness analysis based on hybrid quantum–classical convolutional neural network”. In: *Optics Communications* (2023) (cit. on pp. 113, 117, 133).

[222] Y. Dong et al. “An improved hybrid quantum-classical convolutional neural network for multi-class brain tumor MRI classification”. In: *Journal of Applied Physics* (2023) (cit. on pp. 113, 117, 133).

[223] R. Choudhuri and A. Halder. “Brain MRI tumour classification using quantum classical convolutional neural net architecture”. In: *Neural Computing and Applications* (2022) (cit. on pp. 113, 117, 133).

[224] G. Li et al. “Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Denver, Colorado: Association for Computing Machinery, 2017 (cit. on pp. 114, 118, 119, 121, 126).

[225] F. F. d. Santos et al. “Analyzing and Increasing the Reliability of Convolutional Neural Networks on GPUs”. In: *IEEE Transactions on Reliability* (2019) (cit. on pp. 114, 118, 119, 121).

[226] Y. Ibrahim et al. “Soft Error Resilience of Deep Residual Networks for Object Recognition”. In: *IEEE Access* (2020) (cit. on pp. 114, 121).

[227] J. Biamonte et al. “Quantum machine learning”. In: *Nature* (2017) (cit. on p. 115).

[228] N. Killoran et al. “Continuous-variable quantum neural networks”. In: *Phys. Rev. Research* (3 2019) (cit. on p. 115).

[229] M. Monnet et al. *Pooling techniques in hybrid quantum-classical convolutional neural networks*. 2023 (cit. on p. 115).

[230] T. Li, S. Chakrabarti, and X. Wu. "Sublinear quantum algorithms for training linear and kernel-based classifiers". In: *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*. 2019 (cit. on p. 115).

[231] I. Kerenidis and A. Luongo. "Classification of the MNIST data set with quantum slow feature analysis". In: *Phys. Rev. A* (6 2020) (cit. on p. 115).

[232] V. Havlicek et al. "Supervised learning with quantum-enhanced feature spaces". In: *Nature* (2019) (cit. on p. 115).

[233] A. Abbas et al. "The power of quantum neural networks". In: *Nature Computational Science* (2021) (cit. on p. 115).

[234] U. Ullah, A. G. O. Jurado, I. D. Gonzalez, and B. Garcia-Zapirain. "A Fully Connected Quantum Convolutional Neural Network for Classifying Ischemic Cardiopathy". In: *IEEE Access* (2022) (cit. on pp. 117, 133).

[235] A. Senokosov, A. Sedykh, A. Sagingalieva, and A. Melnikov. *Quantum machine learning for image classification*. 2023 (cit. on pp. 117, 133).

[236] A. Sagingalieva et al. "Hybrid quantum ResNet for car classification and its hyperparameter optimization". In: *Quantum Machine Intelligence* (2023) (cit. on pp. 117, 133).

[237] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* (1989) (cit. on p. 118).

[238] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* (1998) (cit. on p. 118).

[239] T. E. Roth, R. Ma, and W. C. Chew. "The Transmon Qubit for Electromagnetics Engineers: An introduction". In: *IEEE Antennas and Propagation Magazine* (2023) (cit. on p. 121).

[240] S. S. Mukherjee, C. Weaver, J. Emer, S. K. Reinhardt, and T. Austin. "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor". In: *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2003 (cit. on p. 122).

[241] V. Sridharan and D. R. Kaeli. "Eliminating microarchitectural dependency from Architectural Vulnerability". In: *2009 IEEE 15th International Symposium on High Performance Computer Architecture*. 2009 (cit. on p. 122).

[242] T. Tsai, S. K. S. Hari, M. Sullivan, O. Villa, and S. W. Keckler. "NVBitFI: Dynamic Fault Injection for GPUs". In: *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2021 (cit. on p. 123).

[243] D. A. G. Goncalves de Oliveira, L. L. Pilla, T. Santini, and P. Rech. "Evaluation and Mitigation of Radiation-Induced Soft Errors in Graphics Processing Units". In: *IEEE Transactions on Computers* (2016) (cit. on p. 131).

[244] X. Fu et al. "Quingo: A Programming Framework for Heterogeneous Quantum-Classical Computing with NISQ Features". In: *ACM Transactions on Quantum Computing* (2021) (cit. on p. 133).

[245] A. Suau, G. Staffelbach, and A. Todri-Sanial. "Qprof: A Gprof-Inspired Quantum Profiler". In: *ACM Transactions on Quantum Computing* (2022) (cit. on p. 133).

[246] D. Cuomo et al. "Optimized Compiler for Distributed Quantum Computing". In: *ACM Transactions on Quantum Computing* (2023) (cit. on p. 133).

[247] K. N. Smith et al. "TimeStitch: Exploiting Slack to Mitigate Decoherence in Quantum Circuits". In: *ACM Transactions on Quantum Computing* (2022) (cit. on p. 133).

[248] N. Casciola et al. "Understanding the Impact of Cutting in Quantum Circuits Reliability to Transient Faults". In: *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE. 2022 (cit. on p. 133).

List of Publications

- M. Vallero, P. Rech, and F. Vella. “State of practice: Evaluating GPU performance of state vector and tensor network methods”. In: *Future Generation Computer Systems* (2025)
- M. Vallero, G. Casagranda, F. Vella, and P. Rech. “On the Efficacy of Surface Codes in Compensating for Radiation Events in Superconducting Devices”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (2024)
- M. Vallero, G. Casagranda, F. Vella, and P. Rech. “Radiation-Induced Fault Detection in Superconducting Quantum Devices”. In: *Advanced Quantum Technologies* (2026)
- M. Vallero, G. Casagranda, F. Vella, and P. Rech. “TETRIS-Q: Tiling-based Effective Transient-fault Reduction on Interleaved Superconducting Qubits”. In: *review* (2026)
- M. Vallero, E. Dri, E. Giusto, B. Montrucchio, and P. Rech. “Understanding Logical-Shift Error Propagation in Quanvolutional Neural Networks”. In: *IEEE Transactions on Quantum Engineering* (2024)
- G. Casagranda, M. Vallero, F. Vella, and P. Rech. “Understanding the Contributions of Terrestrial Radiation Sources to Error Rates in Quantum Devices”. In: *IEEE Transactions on Nuclear Science* (2025)
- G. Casagranda et al. “SQUID G.A.M.E.: Gamma, Atmospheric, and Mono-Energetic Neutron Effects on Quantum Devices”. In: *IEEE Transactions on Nuclear Science* (2026)