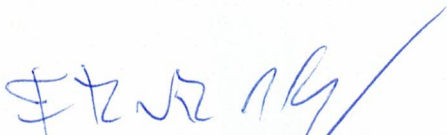


"MAJORIZATION-BASED BENCHMARKING"

NINA MACHADO O'NEILL

Tese de Doutorado em Física apresentada no
Centro Brasileiro de Pesquisas Físicas do
Ministério da Ciência Tecnologia e Inovação.
Fazendo parte da banca examinadora os
seguintes professores:



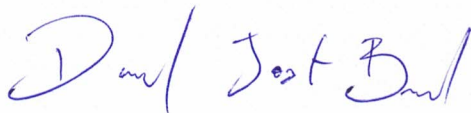
Fernando da Rocha Vaz Bandeira de Melo - Presidente/CBPF



Raúl Oscar Vallejos – Coorientador/ CBPF

Nadja Kolb Bernardes – UFPE

Eduardo Inacio Duzzioni - UFSC



Daniel Jost Brod – UFF



Marcos Cesar de Oliveira – UNICAMP

Rio de Janeiro, 06 de maio de 2025.

CENTRO BRASILEIRO
DE PESQUISAS FÍSICAS



Majorization-based Benchmarking

Assessing computational complexity in near-term quantum devices

Nina M. O'NEILL

Fernando DE MELO
Raúl O. VALLEJOS

Preface

One of the main challenges I faced while writing this thesis was reconciling the two distinct research directions I pursued during my PhD. The first, building on my master's work, explored quantum state assignment based on limited information and resulted in a publication in *Physical Review A* [1]. The second focused on developing a complexity benchmarking protocol for contemporary quantum processors, leading to a publication in *Quantum Information Processing* [2]. While there is a common thread connecting both projects—my interest in bridging theoretical models and experimental implementation through computational investigation—I ultimately concluded that attempting to cover both in a single thesis would not allow me to explore either with the appropriate level of depth. For the sake of thematic unity, I chose to focus solely on my benchmarking research, which comprised the majority of my PhD and aligns more closely with my future research interests.

Abstract

Quantum computing has become a rapidly evolving field, with researchers continually pushing the limits of processor size and capability. Many companies have published ambitious roadmaps to build larger, faster, and more precise quantum processors, aiming to achieve practical applications beyond fundamental research. These advances naturally demand metrics and tests to evaluate the performance of these emerging technologies. To meet this need, we introduce an architecture-independent benchmarking method based on majorization, a mathematical framework for quantitatively assessing the uniformity of probability distributions. Building on prior work that links the complexity of random quantum circuits to majorization relations in their outputs, we propose a complexity indicator capable of benchmarking quantum devices. To establish the protocol's experimental viability, we employ classical machine learning techniques to mitigate the effects of finite measurement statistics and limited gate depth. Numerical simulations accounting for experimental constraints such as noise and hardware connectivity suggest that majorization-based benchmarking provides meaningful complexity assessments of current quantum devices, and can help determine noise thresholds necessary for reliable complex computations.

Keywords: majorization, benchmarking, quantum computing, machine learning, quantum processors, complexity

Resumo

A computação quântica se tornou um campo de rápida evolução, onde pesquisadores continuamente desafiam os limites no tamanho e capacidade dos processadores. Muitas empresas publicaram *roadmaps* ambiciosos para desenvolver processadores quânticos maiores, mais rápidos e mais precisos, visando aplicações práticas para além da pesquisa de base na área. Esses avanços naturalmente demandam métricas e testes para avaliar a performance dessas tecnologias emergentes. Para atender a essa necessidade, apresentamos um método de *benchmarking* independente de *hardware* baseado em majorização, uma ferramenta matemática para avaliar quantitativamente a uniformidade de distribuições de probabilidade. Com base em resultados prévios ligando a complexidade de circuitos quânticos aleatórios a relações de majorização entre seus *outputs*, propomos um indicador capaz de mensurar a complexidade computacional alcançável pelos dispositivos quânticos de hoje. Para estabelecer a viabilidade experimental do protocolo, empregamos técnicas de *machine learning* clássico para mitigar os efeitos da estatística finita de medida e limites na profundidade dos circuitos. Simulações numéricas que levam em conta restrições experimentais, como ruído e conectividade do hardware, sugerem que o protocolo de benchmarking baseado em majorização fornece avaliações substanciais da complexidade de dispositivos quânticos atuais e pode ajudar a estabelecer limites no ruído necessário para realizar computações complexas com confiança.

Palavras-chave: majorização, benchmarking, computação quântica, machine learning, processadores quânticos, complexidade

Contents

1	Introduction	5
2	RQC-based benchmarking	8
2.1	Randomized benchmarking	8
2.2	Quantum volume	10
2.3	Cross-entropy benchmarking	11
3	The majorization-based complexity indicator	15
3.1	The concept of majorization	15
3.2	Majorization in quantum information theory	17
3.3	The principle of majorization	19
3.4	Majorization and complexity in random quantum circuits	21
4	The majorization-based benchmarking protocol	26
4.1	The benchmarking protocol	27
4.2	Noiseless simulations	29
4.3	Noisy simulations	32
4.4	Conclusion	36
5	Accounting for finite statistics	38
5.1	The challenge of finite statistics	38
5.2	Generating Haar- n under finite statistics	40
5.3	Differences between Haar- n and Cliff- n under finite statistics	45
5.4	Conclusion	48
6	An implementable benchmarking proposal	49
6.1	Motivation	49
6.2	Support vector machines	50
6.3	The modified benchmarking protocol	66
6.4	Conclusion	79
7	Conclusion	82

Chapter 1

Introduction

Quantum computing has undergone remarkable growth over the past two decades. Processors with tens [3–9] or even hundreds [10, 11] of qubits have become a reality. A significant milestone was achieved in 2023, when IBM announced the Condor, its first quantum computing processor to exceed 1,000 qubits [12]. As of December 2024, when this thesis is being written, Google has unveiled its quantum chip, Willow—a 105-qubit processor capable of executing a quantum error correction code below the necessary noise threshold [13]. These recent developments exemplify the rapid progress in the field, a trend which is expected to continue as many companies have published ambitious roadmaps [14–17] to develop larger, faster, and more precise quantum processors, striving to build machines with practical applications beyond quantum computing research itself.

This context naturally drives the development of metrics and tests to evaluate the performance of these rapidly advancing technologies. In recent years, various benchmarking techniques have been proposed for this purpose. In a recent review [18], Wang, Guo and Shan classify benchmarks into three types: *physical*, *aggregated* and *application-level* benchmarks.

Physical benchmarks are metrics that reflect the physical properties of a quantum processor. The most well-known benchmark of this type is the number of qubits. Based on the classical concept of a *bit*, which can assume the value of 0 or 1 to perform logical operations, a qubit is a two-level quantum system that can exist in a superposition of states $\alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. A processor with n qubits can prepare a state that exists in a superposition of 2^n classical states, enabling it to process information in ways that are infeasible for classical computers. Due to this exponential scaling of the state space, the number of qubits provides an intuitive sense of the system’s potential scale. However, the performance of a quantum computer actually depends on a variety of factors. Besides the number of physical qubits, important metrics include the device’s connectivity, the number of operations it can execute in parallel, which gates it can natively perform and how many of them can be applied before quantum effects are lost due to errors.

Meanwhile, aggregated benchmarks are metrics proposed with the objective of providing a concise assessment of a device’s performance, accounting for the combined influence of multiple physical properties. In contrast to physical benchmarks, which can be measured directly, aggregated benchmarks must be estimated

from a processor’s basic physical characteristics, or calculated using specific protocols. A notable example of this type of benchmark is quantum volume, a metric that expresses the maximum size of square quantum circuits that can be implemented successfully by the computer, and has been adopted by various quantum computing companies [19–22].

Lastly, application-level benchmarks evaluate performance by running real-world applications on quantum computers. Most of these concern assessing a processor’s capacity for solving optimization problems, which have a wide range of real-world applications [23, 24]. Other areas of interest for application-level benchmarks include machine learning [25] and quantum chemistry [26].

Given the diversity of benchmarking approaches and current quantum device architectures, a comprehensive review of these techniques lies beyond the scope of this thesis. Instead, we focus on aggregated benchmarks that use random quantum circuit-based (RQC-based) protocols to assess a processor’s general computational capacity. During the last few years, RQC-based benchmarks have attracted attention due to their successful experimental implementation and widespread adoption throughout the quantum computing industry [19–22, 27–31]. Due to their unstructured nature, RQC-based protocols are widely used as benchmarks to estimate a quantum device’s overall performance, including its ability to handle complex interactions between qubits. These protocols provide insights into the device’s error rates and fidelity, which are critical for assessing its suitability for running general quantum algorithms. They have also shown promise as a tool for demonstrating *quantum advantage*.

One of the most ambitious goals of the field, quantum advantage refers to the realization of a programmable quantum computer capable of solving problems that are intractable for classical computers within any feasible timeframe. Sampling from a high-dimensional Hilbert space, a task widely believed to be intractable for classical computers [32], could theoretically be achieved by implementing random quantum circuits with a sufficiently large number of qubits. Thus, the construction of a machine capable of implementing large RQCs with small enough error would demonstrate quantum advantage.

In practice, quantum processors are prone to errors. As a result, it is not only necessary to implement a sufficiently large random quantum circuit but also to verify that its output matches the intended probability distribution. A key metric for assessing whether the output of a quantum circuit aligns with the expected outcome is the *fidelity*, defined as [33]:

$$F = \langle \psi | \rho | \psi \rangle, \quad (1.1)$$

where ρ is the quantum state produced by a physical noisy quantum device and $|\psi\rangle$ is the ideal state expected from a noiseless quantum circuit.

However, the exponential scaling of the state space makes measuring fidelity exactly through quantum state tomography infeasible for quantum processors with a large number of qubits. This limitation has led to the development of alternative benchmarks, such as cross-entropy benchmarking (XEB) fidelity, which verify that the output probability distribution of a random quantum circuit follows the expected distribution. XEB has been prominently used in recent quantum advantage experiments [34–37].

We introduce majorization-based benchmarking as an alternative or complementary method to these approaches. Based on the theory of majorization, a mathematical framework for quantitatively assessing the uniformity or inequality of probability distributions, the majorization-based benchmarking protocol analyzes the output of random quantum circuits to characterize a quantum device’s capacity for complex

computation. By requiring only measurements in the computational basis, it is practical to implement, architecture-independent, and less experimentally demanding than computing fidelity. The protocol’s experimental viability can be further enhanced by employing classical machine learning techniques to mitigate the effects of finite measurement statistics and limited gate depth.

Numerical simulations accounting for experimental constraints such as noise and hardware connectivity suggest that majorization-based benchmarking provides meaningful complexity assessments of current quantum devices, and can help determine noise thresholds necessary for reliable complex computations. These features make it a promising tool for benchmarking current quantum processors.

Given this introduction, let us lay out the structure of the thesis. Chapters 2 and 3 establish the theoretical basis for majorization-based benchmarking. Chapter 2 provides a brief overview of notable RQC-based benchmarking protocols currently used in the industry, such as randomized benchmarking, quantum volume and cross-entropy benchmarking, which are widely adopted due to their scalability and ease of implementation. This provides a strong motivation for the RQC-based approach used in our method. Chapter 3 reviews the concept of majorization, explores its correlation with complexity, and presents the majorization-based indicator used in our benchmarking procedure, establishing the mathematical foundation for the benchmark.

Chapter 4 focuses on the work published in Reference [2]. It explains the majorization-based protocol and simulates its application on a specific architecture, the 8-qubit Rigetti Agave. Using numerical results, we demonstrate that majorization-based benchmarking can determine the number of gates required for a noiseless processor to sample from its full Hilbert space. We also simulate this specific architecture in the presence of typical noise, demonstrating the majorization-based indicator’s capacity to detect loss of complexity due to error. These results establish majorization-based benchmarking as a potential means for determining noise thresholds necessary for reliable complex computations.

Building on these results, Chapter 5 studies the viability of majorization-based benchmarking in a finite statistics regime. Using simulated experiments, it outlines the challenges of implementing the benchmark under finite statistics, such as increased measurement overhead as the number of qubits rises, which can hinder scalability. Chapter 6 proposes the use of classical machine learning techniques to address these issues. It introduces a modified benchmarking protocol, which allows a simple machine learning classifier (Support Vector Machine) to be trained to distinguish between RQC outputs coming from universal and non-universal gate sets, using a moderate amount of gates and measurements.

The thesis concludes with a reflection on the strengths of our method, motivating experimental tests of majorization-based benchmarking on actual quantum processors and paving the way for its adoption in the industry.

Chapter 2

RQC-based benchmarking

A variety of benchmarking protocols have been developed to assess the performance of quantum processors, each with distinct advantages and limitations. Among them, methods based on random quantum circuits have gained prominence due to their scalability and practical feasibility. This chapter presents an overview of three widely used RQC-based benchmarking techniques—randomized benchmarking, quantum volume, and cross-entropy benchmarking—highlighting their fundamental principles and roles in current quantum computing research.

2.1 Randomized benchmarking

Randomized benchmarking (RB) [27] is a widely-used protocol to estimate the average error rates of quantum computing hardware platforms [27–31]. Conceptually, the protocol is based on the idea that any sequence of unitary operations can be reversed by applying an appropriate final operation. Consider an arbitrary transformation on a quantum system given by a sequence of unitary operators:

$$U = U_1 U_2 \dots U_s. \quad (2.1)$$

Since unitary operators are invertible, it is always possible to apply an additional operation U_{s+1} such that

$$U_{s+1} = (U_1 U_2 \dots U_s)^\dagger. \quad (2.2)$$

This ensures that the overall transformation acts as the identity in the absence of noise. Randomized benchmarking exploits this principle by analyzing the deviations that arise when noise is present in the system.

The first step of the protocol is to generate a sequence of $s + 1$ quantum operations. The first s are chosen uniformly at random from the Clifford group on n qubits. The final operation is selected so that the net sequence, in the absence of errors, implements the identity operation.

In practice, each operation is affected by noise. Denoting the ideal Clifford operations as C_j and their associated noise processes as Λ_j , the full sequence can be written as:

$$S = (\Lambda_{s+1} \circ C_{s+1}) \circ (\Lambda_s \circ C_s) \circ \cdots \circ (\Lambda_1 \circ C_1) , \quad (2.3)$$

with

$$(\Lambda_{s+1} \circ C_{s+1}) = [(\Lambda_s \circ C_s) \circ \cdots \circ (\Lambda_1 \circ C_1)]^\dagger . \quad (2.4)$$

The second step of the RB protocol is to measure the probability of obtaining the expected measurement outcome after applying the generated sequence. This is also called the *survival probability*. Let ρ_ψ denote the initial state of the quantum system taking into account preparation errors, while E_ψ is the POVM element that accounts for measurement errors. Then

$$\text{Tr} [E_\psi S (\rho_\psi)] \quad (2.5)$$

gives the survival probability of ρ_ψ after applying the sequence S . In the absence of errors, the measurement is assumed to be ideal, meaning E_ψ reduces to the projector $|\psi\rangle\langle\psi|$, the state of the system returns to $|\psi\rangle$ and the survival probability is 1.

Next, the average survival probabilities

$$F_{seq}(\psi, s) = \langle \text{Tr} [E_\psi S (\rho_\psi)] \rangle , \quad (2.6)$$

are measured for a set of ensembles of sequences with varying lengths s .

Under the assumptions of gate-independent and time-independent errors, the random Clifford sequences effectively twirl the noise channel. This causes the average effect of the noise to behave like a depolarizing channel, in the sense that the average survival probability decays exponentially with the sequence length s [27, 38]. Hence, the measured average survival probabilities are fit to an exponential decay model of the form

$$F_{seq}(\psi, s) = Ap^s + B , \quad (2.7)$$

where A and B are coefficients that absorb state-preparation, measurement errors and edge effects. The parameter p determines the noise strength and can be used to determine the average error rate r by the relation [38]

$$r = 1 - p - \frac{1 - p}{2^n} . \quad (2.8)$$

The advantage of RB is that it allows for the estimation of error rates without the need to directly compute the fidelity of the quantum operations, which, as discussed previously, becomes intractable as the number of qubits in the processor increases. Furthermore, by fitting the averaged sequence fidelity to the model given by Equation (2.7), it is possible to separately estimate gate and measurement errors. The scalability of RB, combined with its ability to disentangle different error sources, makes it a powerful and widely adopted tool for assessing and optimizing quantum hardware performance.

2.2 Quantum volume

The performance of a quantum computer depends on various factors, including the number of physical qubits, the amount of gates that can be applied before quantum effects are lost due to errors, the device’s connectivity, and the number of operations it can execute in parallel. The objective of quantum volume is to provide a single-number metric that captures a quantum computer’s overall performance, accounting for all these factors [39, 40].

Quantum volume is defined as

$$V_Q = 2^{d_{max}}, \quad (2.9)$$

where d_{max} is the number of layers of the largest square *model circuit* that processor can successfully implement.

Model circuits are a class of random quantum circuits with a fixed yet generic structure. A model circuit with d layers acting on q qubits is constructed as follows: each layer begins with a random permutation of the qubit labels, determining which pairs will interact. Then, two-qubit gates are applied to each neighboring pair according to this new ordering. These gates are independently sampled from the Haar measure on $SU(4)$, ensuring they are fully random unitary operations. A square model circuit is defined as one for which the width and depth are equal, i.e., $d = q$. A schematic representation of a model circuit can be seen in Figure 2.1.

Cross et al. [40] argue that a processor’s ability to implement these circuits is a meaningful benchmark because, while real quantum algorithms are not purely random, they can be expressed as polynomial-sized quantum circuits composed of two-qubit unitary gates. Consequently, similar structures appear in practical quantum algorithms and can also be used to model generic state preparations.

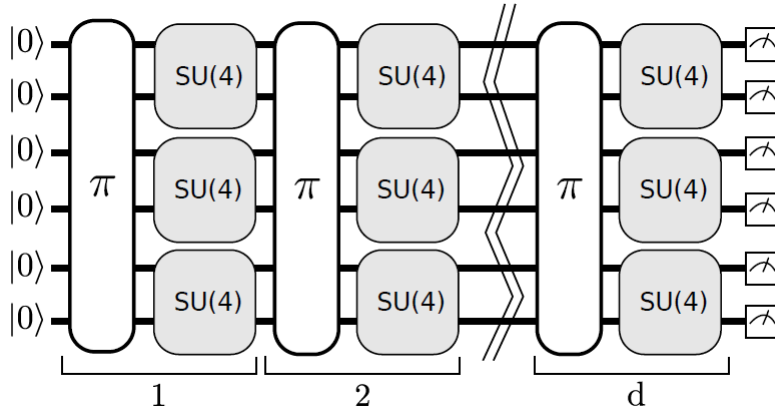


Figure 2.1: Schematic representation of a model circuit, reproduced from Reference [40].

To verify the successful implementation of a model circuit, researchers use the heavy output generation (HOG) test [20, 40], which exploits the statistical properties of model circuit outputs for practical experimental validation.

Consider a random quantum circuit U composed of q qubits. At the end of the circuit, when all qubits are measured, the result will be one of the 2^q possible bit-strings of length q . The probability distribution of all possible outputs can be represented by a 2^q -dimensional vector \mathbf{p}_U , where each component p_x corresponds

to the probability of obtaining the bit-string x . Sorting these components in non-decreasing order,

$$p_0 \leq p_1 \leq \dots \leq p_{2^q-1}, \quad (2.10)$$

the median of the ordered distribution \mathbf{p}_U is given by

$$p_{med} = \frac{(p_{2^{(q-1)}} + p_{2^{(q-1)}-1})}{2}. \quad (2.11)$$

Heavy outputs are defined as the bit-strings x for which $p_x > p_{med}$. To pass the HOG test, at least two-thirds of the measured bit-strings must be heavy.

Asymptotically, an ideal quantum device is expected to produce heavy outputs with a probability of approximately 0.85, whereas a completely depolarized system yields around 0.5 [41]. By evaluating a processor's ability to implement model circuits and pass the heavy output generation test, quantum volume provides a practical and scalable measure of performance. These qualities have lead to its implementation and adoption by various quantum computing companies [19–22].

2.3 Cross-entropy benchmarking

Cross-entropy benchmarking (XEB) is a technique for assessing how accurately a quantum processor executes a random quantum circuit. The method has gained prominence due to its successful application in recent quantum advantage experiments [34,37]. It is widely believed that sampling from a high-dimensional Hilbert space becomes intractable for classical computers as the state space dimension grows [32,41]. Consequently, a device capable of successfully implementing a random quantum circuit with a sufficiently large number of qubits would provide evidence of quantum advantage.

Before proceeding, it is helpful to define more precisely what is meant by random circuit sampling. Following Ref. [42], consider an n -qubit system in a 2^n dimensional Hilbert space \mathcal{H}^n . A quantum circuit U acts on this system, evolving it from an initial state $|\psi_0\rangle$ to a final state $|\psi\rangle = U|\psi_0\rangle$. After executing the circuit and measuring each of the qubits in the computational basis, we recover a bitstring $x = a_1 a_2 \dots a_n$. This bitstring corresponds to a basis state $|x\rangle = |a_1 a_2 \dots a_n\rangle$ in a 2^n dimensional space. The probability of obtaining the measurement $|x\rangle$ after the application of U is given by

$$p_U(x) = |\langle a_1 a_2 \dots a_n | \psi \rangle|^2 = |\langle x | \psi \rangle|^2. \quad (2.12)$$

The task of obtaining a sample x from a probability distribution p_U is known as sampling from the circuit U .

If we consider an ensemble of quantum circuits $\{U_i\}$, it is possible to treat the corresponding probabilities p as random variables following their own probability distribution $P(p)$. In the case of uniformly distributed random quantum circuits, this distribution has the simple form of

$$P(p(x)) = D e^{-D p(x)}, \quad (2.13)$$

where $D \equiv 2^n$ represents the dimension of the Hilbert space. This is known as the Porter-Thomas distribution.

This behavior is notably distinct from that of classical circuits. Consider a set of randomly selected classical functions f mapping Boolean inputs to $\{0, 1\}^n$. Since no output bitstring is favored, the expectation value $\mathbb{E}[p(x)] = 2^{-n}$ for any bitstring x . In other words, in the classical case, the probabilities $p(x)$ are uniformly distributed. For quantum circuits, however, the situation is more complicated. Before measurement, the system is not necessarily in one of the 2^n basis states. Instead, a random wavefunction $|\psi\rangle$ in the computational basis can be written

$$|\psi\rangle = \sum_x (a_x + ib_x) |x\rangle, \quad (2.14)$$

where $a_x, b_x \in \mathbb{R}$ and satisfy

$$\sum_x a_x^2 + b_x^2 = 1. \quad (2.15)$$

The amount of states satisfying these conditions is infinite, so the probability $P(p)$ is given by

$$P(p) = \frac{\text{Vol}(\mathcal{H}_{p,1})}{\text{Vol}(\mathcal{H}_1)}, \quad (2.16)$$

where $\mathcal{H}_{p,1}$ is the subspace of normalized states with probability p and \mathcal{H}_1 the subspace of normalized states. Following the supplementary information in Ref. [36], the numerator is given by

$$\text{Vol}(\mathcal{H}_{p,1}) = \int_{-\infty}^{\infty} \prod_x \delta\left(\sum_x a_x^2 + b_x^2 - 1\right) \delta\left(|\langle x|x\rangle|^2 - p\right) da_x db_x, \quad (2.17)$$

and the denominator,

$$\text{Vol}(\mathcal{H}_1) = \int_{-\infty}^{\infty} \prod_x \delta\left(\sum_x a_x^2 + b_x^2 - 1\right) da_x db_x. \quad (2.18)$$

The first delta function in Eq.2.17 imposes the normalization condition, while the second selects the values of x corresponding to a specific probability p .

Using the Cauchy equation, the function $\delta(x - \alpha)$ can be expressed as

$$\delta(x - \alpha) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{it(x-\alpha)} dt,$$

and so

$$\delta\left(\sum_x a_x^2 + b_x^2 - 1\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{it(\sum_x a_x^2 + b_x^2 - 1)} dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-it} \prod_x e^{it(a_x^2 + b_x^2)}.$$

Plugging this result into Eq.2.18, we obtain

$$\text{Vol}(\mathcal{H}_1) = \int_{-\infty}^{\infty} \prod_x \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-it} \prod_x e^{it(a_x^2 + b_x^2)} \right) da_x db_x \quad (2.19)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-it} dt \left(\int_{-\infty}^{\infty} \prod_x e^{it(a_x^2 + b_x^2)} da_x db_x \right). \quad (2.20)$$

The term in parentheses can be rewritten as

$$\int_{-\infty}^{\infty} \prod_x e^{it(a_x^2 + b_x^2)} da_x db_x = \prod_x \int_{-\infty}^{\infty} e^{ita_x^2} da_x \cdot \int_{-\infty}^{\infty} e^{itb_x^2} db_x.$$

Using

$$\int_{-\infty}^{\infty} e^{its} ds = \sqrt{\frac{i\pi}{t}},$$

we find that

$$\text{Vol}(\mathcal{H}_1) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-it} \left(\frac{i\pi}{t} \right)^D dt = \frac{(i\pi)^D}{2\pi} \int_{-\infty}^{\infty} \frac{e^{-it}}{t^D} dt.$$

The integral in the above expression can be evaluated using Cauchy's residue theorem (See Ref. [42] for details), and the final expression for $\text{Vol}(\mathcal{H}_1)$ is given by

$$\text{Vol}(\mathcal{H}_1) = \frac{(i\pi)^D}{2\pi} \cdot \frac{-2\pi i}{(N-1)!} \cdot (-i)^{N-1} = \frac{\pi^D}{(D-1)!}.$$

The integration of the numerator can be performed through a similar procedure, yielding

$$\text{Vol}(\mathcal{H}_{p,1}) = \frac{\pi^D}{(D-2)!} (1-p)^{D-2}.$$

Thus,

$$P(p) = (D-1)(1-p)^{D-2}. \quad (2.21)$$

Given the expansions

$$e^{-Dp} = 1 - Dp + \frac{(Dp)^2}{2!} - \frac{(Dp)^3}{3!} + \dots, (1-p)^D = 1 - Dp + \frac{D(D-1)}{2!} p^2 - \frac{D(D-1)(D-2)}{3!} p^3 + \dots, \quad (2.22)$$

we see that, as D grows,

$$(1-p)^{D-2} \approx (1-p)^D \approx e^{-Dp},$$

and Eq.2.21 converges to the Portier-Thomas distribution.

In the presence of depolarizing noise, the probabilities $p(x)$ deviate from Portier-Thomas and become more uniformly distributed [43]. To systematically assess how well a quantum device implements a random circuit, cross-entropy benchmarking quantifies this deviation using the cross-entropy benchmark fidelity, defined as

$$\mathcal{F}_{XEB} = D \langle p(x) \rangle - 1. \quad (2.23)$$

This metric compares the measured bit-string probabilities against the theoretical Porter-Thomas distribution, providing an efficient way to estimate fidelity without requiring full quantum state tomography. This makes XEB particularly valuable for benchmarking large-scale quantum processors.

To better understand this metric, let us consider two limiting cases. First, suppose the quantum device is so noisy that the output distribution becomes uniform. In this case, the probabilities satisfy $p(x) = 1/D$ for every bit-string, leading to

$$\mathcal{F}_{XEB} = D \langle p(x) \rangle - 1 = D \times \frac{1}{D} - 1 = 0. \quad (2.24)$$

Next, consider an ideal, noiseless device where the probabilities $p(x)$ follow the expected Porter-Thomas distribution. By Equation (2.13), the fraction of bit-strings with probability in the interval $[p, p+dp]$ is

given by:

$$P(p) dp = D e^{-Dp} dp. \quad (2.25)$$

To find the total number of such bit-strings, we multiply this by the total number of possible bit-strings, $2^n = D$. This yields

$$N(p) dp = D^2 e^{-Dp} dp. \quad (2.26)$$

Using this expression, we can calculate the probability that one of these bit-strings is sampled:

$$p \cdot N(p) dp = p D^2 e^{-Dp} dp. \quad (2.27)$$

We define

$$f(p) = p D^2 e^{-Dp} \quad (2.28)$$

as the probability density function for the ideal probability of a sampled bit-string. This function satisfies the normalization condition:

$$\int_0^1 f(p) dp = 1. \quad (2.29)$$

The probability density $f(p)$ describes the likelihood that a randomly chosen bit-string from the output distribution has an ideal probability p when sampling from an ideal quantum device. Using this result, the average probability of a bit-string sampled from a noiseless device is given by

$$\langle p \rangle = \int_0^1 p f(p) dp = \int_0^1 p D^2 e^{-Dp} dp. \quad (2.30)$$

Evaluating this integral yields [43]

$$\langle p \rangle = \frac{2}{D} \left(1 - e^{-D} \left(\frac{D^2}{2} + D + 1 \right) \right). \quad (2.31)$$

For large D , this can be approximated as

$$\langle p \rangle \approx \frac{2}{D}. \quad (2.32)$$

Substituting this into Equation (2.23) gives

$$\mathcal{F}_{XEB} = D \langle p(x) \rangle - 1 = D \times \frac{2}{D} - 1 = 1. \quad (2.33)$$

Thus, for an ideal quantum device, $\mathcal{F}_{XEB} = 1$.

In the general case, where depolarizing noise is present, the fidelity can be understood as interpolating between the two limiting cases (uniform noise and ideal performance) by a convex combination [43]. This result highlights the role of \mathcal{F}_{XEB} as a practical metric for assessing quantum processor performance. In an ideal, noiseless scenario, the fidelity reaches its upper bound of 1, while in the presence of noise, it decreases proportionally to the probability of errors occurring during the circuit execution. This makes XEB a valuable tool for benchmarking large-scale quantum devices, providing a direct measure of how closely a device's output distribution aligns with theoretical expectations.

Chapter 3

The majorization-based complexity indicator

In many different fields of science, determining how unequal one distribution is when compared to another is often crucial. Whether it be in physics, economics, or information theory, understanding how resources, values, or phenomena are spread across a population can provide significant insights. For instance, in equilibrium statistical mechanics, entropy quantifies how spread out the probability distribution of the system is over its possible microstates, providing a measure of how much work can be extracted [44]. In economics, measuring income inequality can help assess the fairness of wealth distribution [45]. Similarly, in machine learning, the distribution of data across different categories or classes influences algorithm design and performance [46].

Majorization is a mathematical framework for comparing the degree of inequality or uniformity in distributions with precision. Given its role in quantifying inequalities, majorization has found applications across diverse fields [47], including pure mathematics [48], biology [49] and physics [50–53]. Recent findings [54] demonstrate a correlation between majorization and complexity in the context of random quantum circuits. These results provide the mathematical foundation for majorization-based benchmarking.

This chapter begins with a review of the concept of majorization, followed by a brief discussion of its relevance in quantum information theory. We then explore the relationship between majorization and complexity, leading to the findings of Reference [54], which establish the majorization-based indicator used in our benchmarking procedure.

3.1 The concept of majorization

One of the earliest recorded applications of majorization was by economist Max Otto Lorenz, who proposed it as a tool to measure wealth concentration [45]. Lorenz writes that one should plot, on one axis, the cumulative percentage of the population ordered from poorest to richest, and on the other, the total wealth held by this percentage of the population.

In other words, consider a population of N individuals, each holding a certain percentage p_i of the total wealth, with $i = 1, \dots, n$, where the indices are ordered in such a way that $i = 1$ is the poorest and $i = N$ the richest individual. To construct the Lorenz curve of this population, one should plot, on the horizontal axis, the quantity k/N , with $k = 0, \dots, N$, and on the vertical axis, the quantity S_k/S_N , where $S_0 = 0$ and $S_k = \sum_{i=1}^k p_i$. When wealth is uniformly distributed in the population, this procedure yields a straight line. Otherwise, the curve is convex and lies under this straight line. According to Lorenz [45] (p. 217), “With an unequal distribution, the curves will always begin and end in the same points as with an equal distribution, but they will be bent in the middle; and the rule of interpretation will be, as the bow is bent, concentration increases.” An example of the curves used in Lorenz’ original formulation can be seen in Figure 3.1. Curve A corresponds to the case in which wealth is uniformly distributed among the population, while curves B and C are unequal distributions, and curve B represents a more even wealth distribution than that represented by curve C.

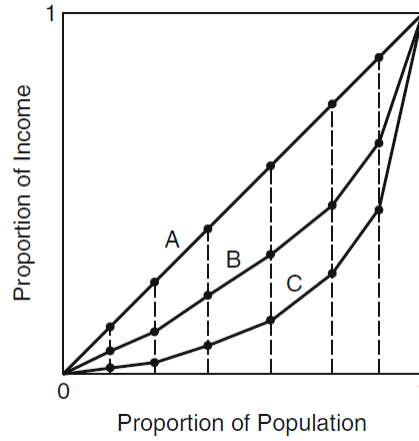


Figure 3.1: An example of the curves used in Lorenz’s original formulation, reproduced from Reference [47]. The uniform probability distribution is represented by a straight line labeled A. All other probability distributions are represented as curves that lie below A.

The definition that will be used in this work closely follows the procedure originally outlined by Lorenz, but with one key difference: while Lorenz sorted individuals in non-decreasing order of income, we sort the components of probability distributions in non-increasing order to align with the usual convention in quantum information.

Consider two vectors, \mathbf{x} and $\mathbf{y} \in \mathbb{R}^N$ with components x_i and y_i , $i = 0, \dots, N$. We say that \mathbf{x} is majorized by \mathbf{y} (or \mathbf{y} majorizes \mathbf{x}), denoted $\mathbf{x} \prec \mathbf{y}$, when

$$\sum_{i=1}^k x_i^\downarrow \leq \sum_{i=1}^k y_i^\downarrow, \quad 1 \leq k < N, \quad (3.1)$$

$$\sum_{i=1}^N x_i = \sum_{i=1}^N y_i. \quad (3.2)$$

Here, the superscript \downarrow signifies that the vector components are sorted in non-increasing order. The partial sums in (3.1) will be denoted $F_x(k)$ and $F_y(k)$, and referred to as the k -th cumulants of \mathbf{x} and \mathbf{y} , respectively.

By plotting the curves of $F(k)$ vs. k/N , majorization relations can be visualized in a straightforward manner. Figure 3.2 illustrates this with two 100-component vectors. The curve labeled ‘uniform’ represents a

uniform distribution, while 'random' corresponds to a vector whose components were sampled from a Gaussian distribution and then normalized. According to definition (3.1), in Figure 3.2, the random distribution majorizes the uniform distribution. Notably, under the quantum information sorting convention, if \mathbf{x} majorizes \mathbf{y} , the Lorenz curve of \mathbf{x} lies above that of \mathbf{y} . The Lorenz curve of the uniform distribution, which is majorized by all other comparable probability distributions, always lies below the others.

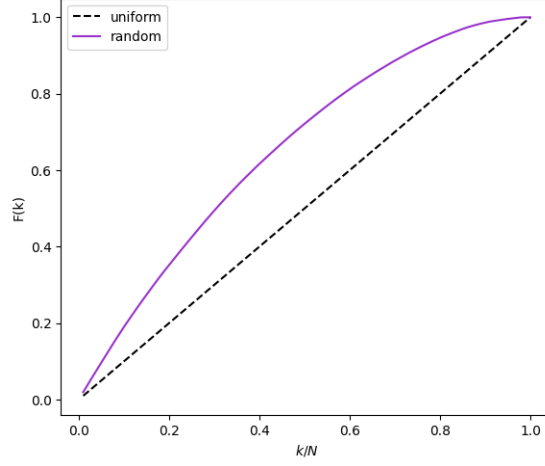


Figure 3.2: An example of Lorenz curves that will be used in this work, in which the coefficients of probability distribution vectors are sorted in non-increasing order.

An equivalent definition of majorization is given by a theorem of Hardy, Littlewood, and Pólya [55]. According to this theorem, for two vectors \mathbf{x} and $\mathbf{y} \in \mathbb{R}^N$,

$$\mathbf{x} \prec \mathbf{y} \iff \mathbf{x} = P\mathbf{y} \quad (3.3)$$

for some square matrix $P = (p_{ij})$ of nonnegative real numbers, where each row and column sums to 1,

$$\sum_i p_{ij} = \sum_j p_{ij} = 1. \quad (3.4)$$

Such a matrix is called bistochastic. In other words, we say that \mathbf{y} majorizes \mathbf{x} if it is possible to construct \mathbf{x} by multiplying \mathbf{y} by some bistochastic matrix P . This definition will be important when covering different applications of majorization in physics, as we shall see in the following subsections.

3.2 Majorization in quantum information theory

To better understand the connections between majorization and quantum mechanics, let us first formulate an operator-based definition of majorization. This will allow us to evaluate majorization relations between density matrices, which in quantum mechanics often perform a somewhat analogous role to that of probability distributions. Let R and S be d -dimensional Hermitian operators. We say [53]

$$R \prec S \iff \lambda(R) \prec \lambda(S), \quad (3.5)$$

where $\lambda(R)$ denotes the vector whose components are the eigenvalues of R , arranged in non-increasing order.

In Reference [53], Nielsen and Vidal shed light on the close connection between majorization and quantum mechanics through the combination of two results, Horn's lemma [56] and Uhlmann's theorem [57]. Horn's lemma states that, for vectors \mathbf{r} and \mathbf{s} ,

$$\mathbf{r} \prec \mathbf{s} \iff r_i = \sum_j |u_{ij}|^2 s_j \quad (3.6)$$

for some unitary matrix $U = (u_{ij})$ of complex numbers. A matrix that satisfies this property is necessarily bistochastic, making Horn's Lemma a special case of Equation (3.3). In turn, Uhlmann's theorem states that $R \prec S$ for Hermitian matrices R and S if and only if there exist unitary matrices U_j and a probability distribution $\{p_j\}$ such that

$$R = \sum_j p_j U_j S U_j^\dagger. \quad (3.7)$$

Due to the fundamental role of unitarity in quantum mechanics, relations of the type found in Horn's lemma and Uhlmann's theorem arise frequently. According to Nielsen and Vidal, this accounts for many of the applications of majorization in quantum mechanics.

One of the most significant areas where majorization relations emerge is the study of quantum entanglement. Specifically, they provide a framework for understanding the feasibility of state transformations in bipartite systems by means of local operations on the two subsystems aided by classical communication (LOCC). Consider a bipartite system AB in an entangled pure state ψ . The feasibility of the transformation of ψ into another state ϕ by means of LOCC depends on a majorization relation between the reduced density matrix of part A of the system for states ψ and ϕ . Namely, the transformation $\psi \rightarrow \phi$ through LOCC is feasible if, and only if [53]

$$\rho_A^\psi \prec \rho_A^\phi, \quad (3.8)$$

where ρ_A^ψ and ρ_A^ϕ are the reduced density matrices of part A of the system for the states ψ and ϕ , respectively.

Another property of majorization with interesting implications for quantum information theory is its relation to a Schur-convex functions. A real-valued function f defined on \mathbb{R}^n is said to be Schur-convex if, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\mathbf{x} \prec \mathbf{y} \implies f(\mathbf{x}) \leq f(\mathbf{y}). \quad (3.9)$$

If, in addition, $f(\mathbf{x}) < f(\mathbf{y})$ whenever $\mathbf{x} \prec \mathbf{y}$ but \mathbf{x} is not a permutation of \mathbf{y} , then f is said to be strictly Schur-convex. Similarly, f is said to be Schur-concave if

$$\mathbf{x} \prec \mathbf{y} \implies f(\mathbf{x}) \geq f(\mathbf{y}), \quad (3.10)$$

and strictly Schur-concave if strict inequality $f(\mathbf{x}) > f(\mathbf{y})$ holds when \mathbf{x} is not a permutation of \mathbf{y} . Evidently, f is Schur-concave if and only if $-f$ is Schur-convex. Notably, the Shannon entropy, defined as

$$H(\mathbf{x}) \equiv - \sum_i x_i \log x_i \quad (3.11)$$

is strictly Schur-concave [47]. As a consequence, If a probability distribution \mathbf{y} associated to a random variable Y majorizes the probability distribution \mathbf{x} of a random variable X , then $H(\mathbf{x}) > H(\mathbf{y})$. However, the opposite is not necessarily true: $H(\mathbf{x}) > H(\mathbf{y})$ does not imply that \mathbf{y} majorizes \mathbf{x} . This asymmetry

highlights majorization's finer granularity compared to entropy in quantifying the spread of probability distributions.

Recognizing this potential, Ruch [58] used majorization relations to propose an alternative to the second law of thermodynamics, known as the principle of *increasing mixing character*. Inspired by this work, Órus, Latorre and Martín-Delgado [51] proposed a principle of majorization in the context of quantum algorithms, which will be detailed in the next section.

3.3 The principle of majorization

Before explaining Órus, Latorre and Martín-Delgado's majorization principle, we briefly discuss Ruch's principle of increasing mixing character. According to Ruch [58] (p. 182), "The time development of a statistical (Gibbs) ensemble of isolated systems (microcanonical ensemble) proceeds in such a way that the mixing character increases monotonically".

To better illustrate Ruch's principle, we use as an example a particular formulation of the master equation, known as the Pauli master equation [59]. Consider a distribution of n particles among m states. Assume that the number of particles is constant and the transitions are governed by constant probabilities w_{ik} . Furthermore, suppose n is large enough that differentiation with respect to time is justified. This formulation can be expressed as

$$\dot{n}_i(t) = \sum_{k=1}^m w_{ik} n_k(t) - \sum_{l=1}^m w_{li} n_i(t), \quad w_{ik} \geq 0, \quad (3.12)$$

or

$$\dot{\mathbf{n}}(t) = \sum_{k=1}^m W_{ik} n_k(t), \quad W_{ik} = w_{ik} - \delta_{ik} \sum_{l=1}^m w_{li}. \quad (3.13)$$

Defining the vector $\mathbf{n} = \{n_i\}$, we may write Equation (3.13) as $\dot{\mathbf{n}}(t) = W\mathbf{n}(t)$. The solution to this equation is given by

$$\mathbf{n}(t) = e^{W(t-t_0)} \mathbf{n}(t_0). \quad (3.14)$$

At equilibrium, $\dot{\mathbf{n}}(t) = 0$. Imposing this condition on Equation (3.13), the matrix W satisfies

$$\sum_{i=1}^m W_{ik} = \sum_{i=1}^m w_{ik} - \sum_{l=1}^m w_{lk} = 0; \quad (3.15)$$

$$\sum_{k=1}^m W_{ik} = \sum_{k=1}^m w_{ik} - \sum_{k=1}^m w_{lk} = 0. \quad (3.16)$$

that is, the sum of the elements in each of its rows and columns is zero. The expansion $e^{W(t-t_0)}$ is given by

$$e^{W(t-t_0)} = \mathbb{1} + (t-t_0)W + (t-t_0)^2 W^2 + \dots \quad (3.17)$$

where $\mathbb{1}$ is the identity matrix.

When multiplying two matrices for which the elements of each of their rows and columns sum to zero, the resulting matrix also retains this property. Consider two $m \times m$ matrices A and B . The entries of the matrix $C = AB$ are given by

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj} . \quad (3.18)$$

Now, consider the sum of the i -th row of C :

$$\sum_{j=1}^m c_{ij} = \sum_{j=1}^m \sum_{k=1}^m a_{ik} b_{kj} = \sum_{k=1}^m a_{ik} \sum_{j=1}^m b_{kj} . \quad (3.19)$$

If the sum of the elements in each row of B is zero, we have:

$$\sum_{j=1}^m b_{kj} = 0 \quad \forall k . \quad (3.20)$$

This implies that

$$\sum_{j=1}^m c_{ij} = 0 \quad \forall i , \quad (3.21)$$

meaning that each row of C also sums to zero.

Similarly, the sum of the j -th column of C is given by:

$$\sum_{i=1}^m c_{ij} = \sum_{i=1}^m \sum_{k=1}^m a_{ik} b_{kj} = \sum_{k=1}^m a_{ik} \sum_{i=1}^m b_{kj} . \quad (3.22)$$

If the sum of the elements in each column of A is zero, we have:

$$\sum_{i=1}^m a_{ik} = 0 \quad \forall k . \quad (3.23)$$

This implies that

$$\sum_{i=1}^m c_{ij} = 0 \quad \forall j , \quad (3.24)$$

meaning that each column of C also sums to zero.

Thus, since the sum of the elements of each of the rows and columns of W is zero, the same is true for all powers of W . Consequently, the sum of the elements of each row and column of $e^{W(t-t_0)}$ is one. Because the number of particles in each state can never be negative, there are no negative coefficients and we conclude that $e^{W(t-t_0)}$ is a bistochastic matrix. Given this, Equation (3.14) implies that

$$\mathbf{n}(t) \prec \mathbf{n}(t_0) \quad \text{if } t \geq t_0 , \quad (3.25)$$

and the time development of the system proceeds in the direction of monotonically increasing statistical disorder, i.e. decreasing majorization.

Inspired by the principle of increasing mixing character, Órus, Latorre and Martín-Delgado proposed a majorization principle for the behavior of quantum algorithms. To understand this principle, let us first

consider a quantum computer executing an algorithm that uses n qubits and has N_s computational steps. At step s of the computation, the register is in a pure state $|\Psi_s\rangle$, with $s = 0, 1, \dots, N_s$. It is possible to associate a probability distribution to this state in the following way. First, we decompose the register state in the computational basis,

$$|\Psi_s\rangle = \sum_{i=0}^{2^n-1} c_i |\phi_i\rangle, \quad (3.26)$$

where each $|\phi_i\rangle$, denotes a basis state. Given this, we can define an ordered vector \mathbf{p}_s^\downarrow composed of the coefficients $|c_i|^2$ in nonincreasing order. This distribution characterizes the probability of obtaining each individual state of the computational basis when performing a measurement on the register at step s of the computation.

Órus, Latorre and Martín-Delgado claim [51] that, for efficient quantum algorithms (those which offer a significant speedup compared to the best known classical counterparts), the time development of the system proceeds in the direction of monotonically increasing statistical order, i.e. *increasing* majorization. The probability distributions \mathbf{p}_s^\downarrow associated to the state of quantum registers at different operating stages satisfy

$$\mathbf{p}_s^\downarrow \prec \mathbf{p}_{s+1}^\downarrow, \forall s = 1, \dots, N_s - 1, \quad (3.27)$$

where the index s runs from 1 to $S - 1$ so that the first step is excluded.

To corroborate this, the authors demonstrated that this principle was satisfied by a variety of algorithms known to be efficient in the previously described sense, such as Grover's, quantum adiabatic evolution and phase-estimation algorithms, including Shor's [51, 52].

3.4 Majorization and complexity in random quantum circuits

Órus, Latorre and Martín-Delgado's result suggests that majorization relations play a meaningful role in the behavior of efficient quantum algorithms. Given that random quantum circuits are frequently used in practical quantum algorithms and play a key role in quantum advantage experiments, Vallejos, Melo, and Carlo investigated how majorization evolves in these systems [54]. The study considered RQCs constructed using three finite gate sets, $G1=\{\text{CNOT}, \text{H}, \text{NOT}\}$, $G2=\{\text{CNOT}, \text{H}, \text{S}\}$ and $G3=\{\text{CNOT}, \text{H}, \text{T}\}$, as well as matchgates and diagonal gates. The evolution of majorization in these systems was studied in relation to the complexity class of the gate sets, which fall into three categories: universal, classically simulable, or neither universal nor classically simulable.

A set of quantum gates is universal for quantum computation if any unitary operation can be approximated to arbitrary accuracy using only those gates [33]. In contrast, a set of gates is classically simulable if the measurement outcomes of quantum circuits composed solely of these gates can be efficiently computed using classical resources. A simulation is deemed efficient if it runs in polynomial time with respect to the system size.

There are two main notions of classical simulability: *weak* and *strong*. Consider a set of quantum circuits U which can be efficiently generated by a classical algorithm, with both the input states and the types of measurements performed on the output being restricted to specific classes. The family U is strongly simulable under these constraints if the exact probabilities of measurement outcomes can be computed efficiently

using a classical algorithm [60]. In contrast, U is weakly simulable if a sample from its output distribution can be generated efficiently using classical means [61].

The gate set G3 is universal [33], while the sets G1 and G2 contain only Clifford gates and are thus nonuniversal and classically simulable [62]. Matchgate (MG) circuits are a class of 2-qubit quantum gates, which can be represented by the following matrix:

$$G(A, B) = \begin{pmatrix} p & 0 & 0 & q \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ r & 0 & 0 & s \end{pmatrix}, \quad (3.28)$$

where A and B are unitary 2×2 matrices with equal determinant. These matrices are defined as:

$$A = \begin{pmatrix} p & q \\ r & s \end{pmatrix} \quad B = \begin{pmatrix} w & x \\ y & z \end{pmatrix}. \quad (3.29)$$

To construct random quantum circuits using matchgates, the matrices A and B are first selected randomly according to the Haar measure in the unitary group $U(2)$. These gates are then applied to randomly selected pairs of qubits, with each pair having an equal probability of being chosen. Matchgate circuits are of particular interest in this context because their complexity class changes depending on the connectivity conditions. Circuits that only act on nearest-neighbor qubits are classically simulable. However, when the nearest-neighbor restriction is lifted, the resulting circuits become universal for quantum computation, as shown in Reference [60].

Finally, diagonal-gate circuits are made up from gates which are diagonal in the computational (Z) basis. The initial state is set to $|0\rangle^{\otimes n}$ and Hadamard gates are placed at the beginning and ending of each line. Diagonal circuits cannot perform universal computation, however, it has been shown that they are not classically simulable under plausible assumptions in computational complexity theory [63]. The authors used a particular subclass of diagonal circuits: the r -qubit phase-random circuits, which have the form [64]

$$W_r = \text{diag} \{ e^{i\phi_1}, e^{i\phi_2}, \dots, e^{i\phi_r} \}, \quad (3.30)$$

where the ϕ 's are independent, random and uniformly distributed in $[0, 2\pi)$. Since diagonal gates commute, the order of application does not affect the resulting circuit. The gates were applied on all combinations of r (out of n) qubits, the ordering being random. The study used only three values for r , namely $r = 2, 3, n$. These types of circuits will be referred to as D2, D3 and Dn, respectively.

It was found that, for all types of circuits, the averages of the ordered probability distributions \mathbf{p}_s^\downarrow exhibited decreasing majorization as the number of computational steps (in this case, given by the number of gates) increased [54].

To illustrate this result, we present the results of an analogous simulated experiment using the gates in the G1 gate set. The random quantum circuits were composed of 8 qubits and the computational procedure, which will be described shortly, was performed 500 times. First, the circuit was initialized in a random product state and its output probability distribution calculated. Next, random gates from G1 were applied. For every 25 gates applied, the output probability distribution was calculated once more. Finally, the average

probability distributions for the outputs was computed at computational steps $\{0, 25, 50, 75, 100, 125\}$. The resulting average Lorenz curves can be seen in Figure 3.3.

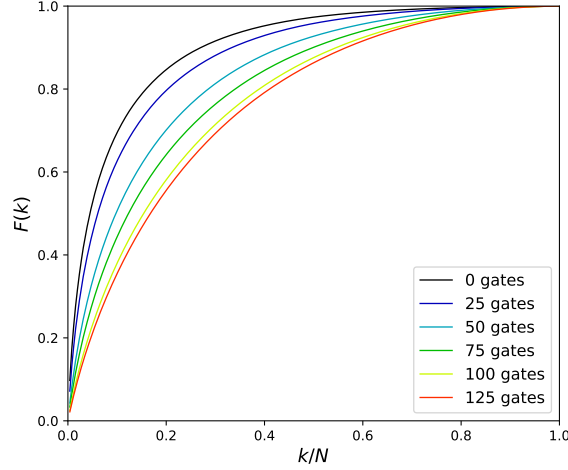


Figure 3.3: Average Lorenz curves of the probability distributions associated to the output of random quantum circuits constructed with the gate set $G1 = \{\text{CNOT}, \text{H}, \text{NOT}\}$, exhibiting decreasing majorization with the number of gates.

In this plot, it is possible to see that, for any given number of gates, the average Lorenz curve representing the output probability of the circuit at that particular computational step is below the one associated to the previous step. In other words, the output probability distributions exhibit decreasing majorization with the number of gates.

In Reference [54], the authors also observed that, for a sufficiently large number of gates, the averaged Lorenz curves for all the studied circuit families converge to an asymptotic curve. While the rate of change in the Lorenz curves varies depending on the circuit type, many of the asymptotic curves closely resemble those constructed from randomly sampled vectors in the Hilbert space using the Haar measure. These results are illustrated in Figure 3.4. The abbreviations in the graph are structured in the form A-B-C. A represents the type of circuit: G1, G2, G3, MG, D2, D3 or Dn. B refers to the connectivity, and can be NN (nearest neighbors), RN (random neighbors) or ALL (all combinations of r qubits). C refers to the initial state of the circuit. When the initial state is a random product state, C will be equal to RS, and when it is set to $|0\rangle^{\otimes n}$, C will be 0. The curves labeled Haar- n refer to those constructed from randomly sampled vectors, where 2^n is the dimensionality of the Hilbert space from which the vectors were sampled.

Since all the circuits simulated in Reference [54] had 8 qubits, the asymptotic average Lorenz curves for most of them coincided with Haar-8. Those that did not coincide with Haar-8 still exhibited only minimal deviation from the curve. In other words, for a sufficiently large number of gates, the averaged Lorenz curves of all studied circuit families closely resemble those of random vectors in an 8-dimensional Hilbert space.

However, studying the standard deviation of the asymptotic Lorenz curves revealed an important finding: the fluctuations in Lorenz curves for the asymptotic states of RQCs from different complexity classes show noticeably distinct qualitative behavior. Formally, let \mathbf{p}_U be the probability distribution corresponding to the asymptotic state of a random quantum circuit U . The k -th cumulant, $F_{p_U}(k)$, of this distribution is defined as:

$$F_{p_U}(k) = \sum_{i=1}^k p_{U_i}^\downarrow, \quad (3.31)$$

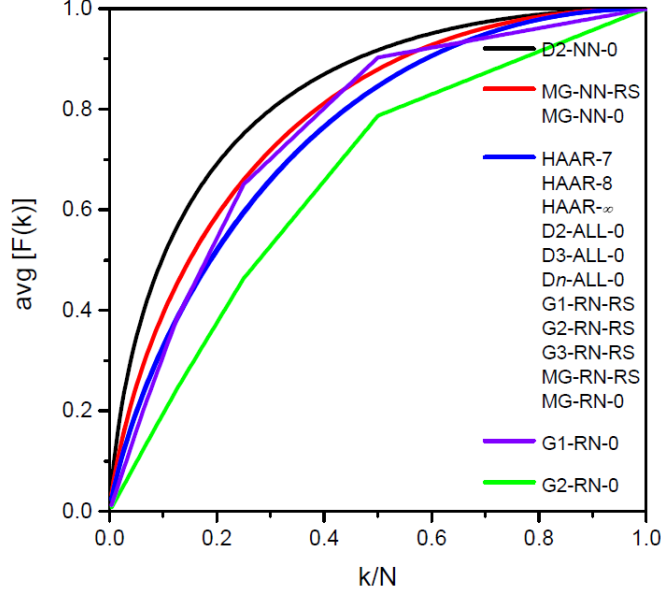


Figure 3.4: Average Lorenz curves for asymptotic states generated by various families of random circuits of 8 qubits, reproduced from Reference [54].

where $p_{U_i}^\downarrow$ is the i -th component of the sorted probability vector \mathbf{p}_U^\downarrow . The fluctuations in $F_{p_U}(k)$ are quantified by the standard deviation:

$$\text{std}[F_{p_U}(k)] = \sqrt{\langle F_{p_U}^2(k) \rangle - \langle F_{p_U}(k) \rangle^2}, \quad (3.32)$$

which is calculated over an ensemble of circuits generated from the same gate sets.

By plotting the fluctuations against k/N , the authors observed distinct qualitative behaviors in the curves for ensembles of random quantum circuits constructed from gate sets of different complexity classes, as shown in Figure 3.5. Notably, the fluctuation curves for G3 circuits, diagonal circuits, and matchgate circuits without nearest-neighbor restrictions—all of which are not classically simulable—aligned closely with those of random vectors of 8 qubits (Haar-8). In contrast, circuits known to be classically simulable, such as matchgate circuits with nearest-neighbor connectivity and circuits composed of the G1 and G2 gate sets, exhibited significant deviations from Haar-8.

To further validate the correlation between the majorization-based indicator and computational complexity, the authors compared fluctuations in the Lorenz curves with entanglement spectrum statistics, which have also been shown to correlate with complexity [65, 66]. To define the entanglement spectrum, consider a system partitioned into two subsystems, A and B , associated with Hilbert spaces \mathcal{H}_A and \mathcal{H}_B , respectively. Let n_A and n_B denote the number of qubits in each subsystem, with the total number of qubits given by $n = n_A + n_B$. The total system is described by the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. For a pure state $|\psi\rangle \in \mathcal{H}$, the reduced density matrix ρ_A is obtained by tracing out subsystem B :

$$\rho_A = \text{Tr}_B |\psi\rangle \langle \psi|. \quad (3.33)$$

The eigenvalues of ρ_A , denoted $\text{spec}\{\rho_A\} = \{\lambda_1, \dots, \lambda_{n_A}\}$, constitute the reduced density matrix spectrum. The entanglement spectrum of an RQC is the spectrum of the reduced states ρ_A^U after the circuit U is

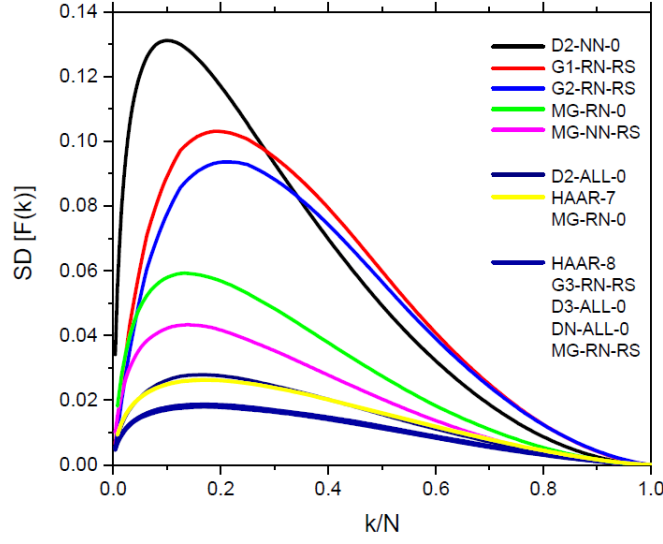


Figure 3.5: Standard deviation of Lorenz curves for asymptotic states generated by various families of random circuits of 8 qubits, showing distinct behaviors for RQCs constructed from gate set of different complexity classes. Reproduced from Reference [54].

applied to the initial (pure) state. Shaffer et al. [65] found that the entanglement spectra of RQCs composed of universal gates follow Wigner-Dyson statistics, while those constructed from non-universal gates exhibit Poisson-like statistics. Following this methodology, Vallejos, Melo, and Carlo analyzed a 50-50 bipartition of the asymptotic states of circuits from different families and calculated their entanglement spectra. Their results aligned with the classification obtained using the majorization-based indicator.

These results provide compelling evidence of the correlation between majorization relations and computational complexity in random quantum circuits. The fluctuations in the asymptotic Lorenz curves serve as a reliable indicator that can distinguish between RQCs composed of gates from different complexity classes. Furthermore, as it is not based on entanglement, which can degrade quickly even in weakly dissipative environments [67], the majorization-based indicator shows more promise for practical implementations, where noise is often a crucial factor. In the next chapter, we will discuss how this majorization-based indicator can be used in an RQC-based benchmarking protocol to assess the computational capacity of current quantum processors for reliable, complex computation.

Chapter 4

The majorization-based benchmarking protocol

Chapter 2 reviewed several notable RQC-based methods for benchmarking quantum processors. The success of protocols such as randomized benchmarking, quantum volume, and cross-entropy benchmarking demonstrates the scalability and practicality of RQC-based approaches. In particular, the XEB protocol illustrates how statistical properties of random quantum circuits can be leveraged to extract meaningful information about a device’s performance.

Meanwhile, the findings of Reference [54] demonstrate that it is possible to discriminate between random quantum circuits of different complexity classes by plotting the fluctuations in the Lorenz curves of their asymptotic output states. While these results come from different contexts, they suggest a natural connection: if random quantum circuits are effective for benchmarking and majorization provides a meaningful way to analyze them, the majorization-based indicator may offer a practical way to assess a quantum device’s ability to perform complex computations.

Building on this logic, we introduce the majorization-based benchmarking protocol [2]. Rooted in the theory of majorization, this tool provides an alternative approach to characterizing the computational capabilities of current quantum devices, focusing on a processor’s ability to sample from its full Hilbert space.

In this chapter, we detail the protocol’s implementation and simulate its use on an existing device, the 8-qubit Rigetti Agave. Using the classical supercomputer KUATOMU together with the ATOS QLM system, we model the benchmarking procedure while incorporating experimental constraints such as noise and hardware connectivity. Our results demonstrate how majorization-based benchmarking can help determine noise thresholds necessary for reliable complex computations.

4.1 The benchmarking protocol

As discussed in the previous chapter, the fluctuations in Lorenz curves derived from the output probabilities of random quantum circuits composed of universal gate sets exhibit a limiting behavior qualitatively similar to that of random n -qubit vectors. This heuristic observation suggests a correlation between the qualitative characteristics of the Lorenz curve fluctuations and the circuit’s capacity to sample from the full Hilbert space.

Protocols such as quantum volume and XEB establish successful sampling from a universal distribution as a key benchmark for the computational capacity of quantum processors. Proponents of quantum volume argue that random quantum circuits model arbitrary quantum operations, offering a general measure of a device’s computational capabilities [39]. Recent efforts to demonstrate quantum advantage have also focused on proving a device’s capability of sampling from a universal distribution [34,35,37]. In this context, XEB has been implemented as an alternative metric to fidelity, verifying the successful implementation of random quantum circuits.

Drawing from these results, we propose a majorization-based benchmarking protocol. First, we implement random quantum circuits using the native gates of a given device. The output probabilities of this experiment are then compared against two reference curves: (i) n -qubit Haar-random pure states, and (ii) randomized n -qubit Clifford circuits. These curves are denoted Haar- n and Cliff- n , respectively.

As noted earlier, random quantum circuits composed entirely of Clifford gates are strongly simulable. Furthermore, these circuits exhibit a characteristic limiting behavior that is qualitatively distinct from random n -qubit vectors [54]. Thus, while the Haar- n curve characterizes the sampling capabilities of a noiseless quantum device able to sample from its full Hilbert space, the asymptotic Cliff- n curve serves as a natural reference for sampling capabilities that can be efficiently sampled by classical means.

The Clifford circuits used in this procedure are generated from the set $\{\text{CNOT}, \text{H}, \text{S}\}$, starting from a random pure separable state and no qubit connectivity constraint. The Haar- n line, on the other hand, is generated by sampling an ensemble of random n -qubit states from the Haar measure and computing its cumulant fluctuations. An example of these characteristic fluctuation curves for 8-qubit systems can be seen in Figure 4.1. These curves were computed from an ensemble of 10^4 random vectors. The Cliff-8 curve was computed from the output probabilities of Clifford circuits composed of 2000 gates, while the Haar-8 curve was calculated using an ensemble of Haar-random 8-qubit pure states.

The fluctuations in Lorenz curves computed from the RQC output probabilities depend on various factors, including the number of circuits used to compute the standard deviation, the number of gates per circuit, qubit connectivity, and noise levels. To isolate these factors and study their individual effects, we first analyzed noiseless quantum processors.

Using SENAI CIMATEC’s quantum computing simulator, KUATOMU, we simulated the operation of several currently available universal, circuit-model quantum processing units (QPUs). I conducted simulations of Rigetti’s Agave 8-qubit QPU [68], as well as theoretical architectures inspired by this device. My collaborator, Alexandre B. Tacla of SENAI-CIMATEC, analyzed IBM’s architectures, including the 5-qubit IBM Q Yorktown and the 7-qubit IBM Perth. In this thesis, I will focus on the simulations of Rigetti’s devices. The results for IBM’s architectures, which are qualitatively similar to those obtained for Rigetti’s, can be found in Reference [2].

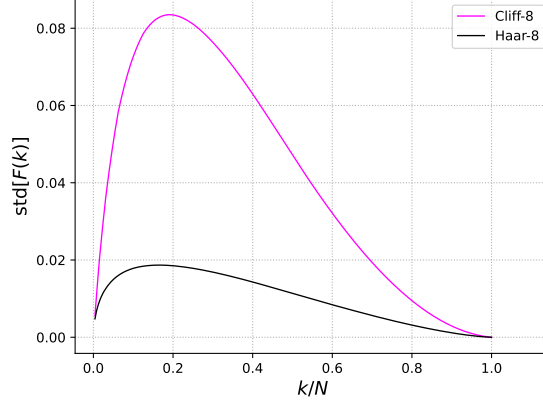


Figure 4.1: Illustration of the characteristic curves used in the benchmarking procedure for the case of an 8-qubit system.

The circuits used in my benchmarking experiments were constructed by sampling from the gates of the set $G_{\text{Rig}} = \{\text{RX}, \text{RZ}, \text{CZ}\}$, where RZ rotations can be of an arbitrary angle, but RX rotations are restricted to $\pm\frac{\pi}{2}$ and $\pm\pi$. The angles of the RZ rotations are sampled uniformly from $[-\pi, \pi)$ and the angles of the RX rotations, from $\{-\pi, -\frac{\pi}{2}, \frac{\pi}{2}, \pi\}$. These are the gates which can be natively implemented by Rigetti’s older quantum processors [69].

Our initial objective was to determine the number of gates required to achieve the desired level of complexity in the output while taking into account the Agave’s native gate set and connectivity constraints. To address this, we systematically varied these parameters in our simulations, analyzing their impact on the fluctuation curves. After this, we explored the role of qubit connectivity by simulating other possible architectures. Figure 4.2 illustrates the studied configurations. Graph (a) represents the architecture of the Agave device, which has a “ring” geometry with nearest-neighbor connections, while graphs (b)–(d) represent hypothetical 8-qubit devices with an increasing number of connections between the qubits.

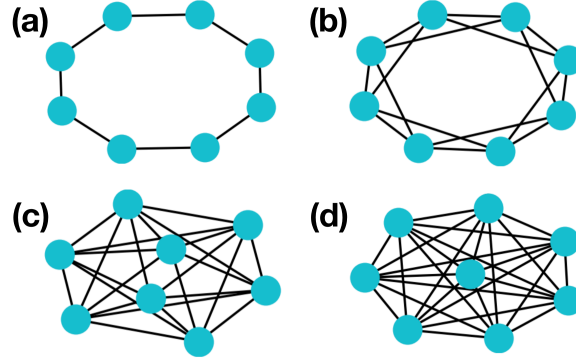


Figure 4.2: Connectivity maps for (a) Rigetti Agave 8-qubit, “ring” QPU, with nearest-neighbor connectivity ($n_c = 2$), and other examples of 8-qubit QPU geometries with increasing qubit connectivity: (b) $n_c = 4$, (c) $n_c = 6$, and (d) $n_c = 7$ (all-to-all).

These simulations allowed us to systematically assess how circuit depth and qubit connectivity influence the complexity of the output distribution. By comparing fluctuation curves across different configurations, we identified the minimal conditions required for the circuits to exhibit behavior consistent with universal sampling. In the following section, we present our findings, discussing how these factors impact the effectiveness of the benchmarking protocol in a noiseless setting.

4.2 Noiseless simulations

In majorization-based benchmarking, the asymptotic fluctuations in the Lorenz curves of Haar-random vectors define the reference curve for universal sampling capabilities. To ensure the reliability of the benchmarking procedure, this reference curve must be stable. As a first step, we analyzed how ensemble size affects the statistical variation of the Haar-8 curve, determining the minimum number of samples needed to make these fluctuations negligible. Figure 4.3 illustrates how the fluctuations in the Lorenz curves of Haar-random 8-qubit probability vectors vary with ensemble size. For small ensembles, the curve exhibits significant statistical variation but stabilizes as the number of samples increases.

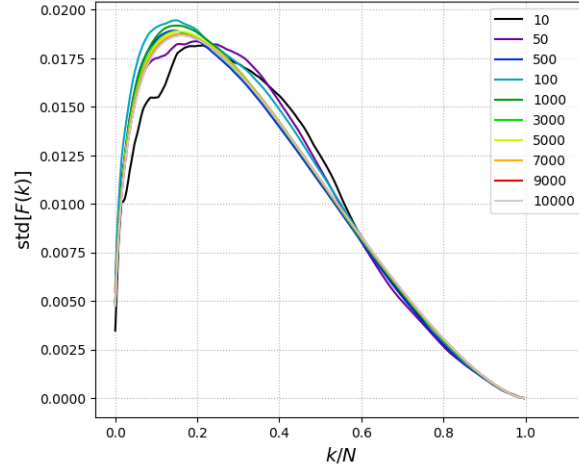


Figure 4.3: Fluctuations in the Lorenz curves of Haar-random 8-qubit probability vectors for different ensemble sizes. As the number of samples increases, the curves converge toward the Haar-8 benchmark line.

To study this phenomenon systematically, we generated a total of 10^4 Haar-random 8-qubit probability vectors and divided them into batches with an increasing number of samples. The first batch consisted of 500 samples, the second of 1,000, and so on, with the final batch comprising all 10^4 samples. This produced a total of 20 batches, which were labeled by an index $i \in \{1, \dots, 20\}$. Next, we computed the standard deviation of the Lorenz curves within each batch. To quantify the convergence to the asymptotic Haar-8 curve, we measured the vertical distance between the peak standard deviations of consecutive batches, defined as:

$$\Delta P_i = \left| \max [\text{std} [F(k)]_i] - \max [\text{std} [F(k)]_{i-1}] \right|. \quad (4.1)$$

In this formula, $\text{std}[F(k)]_i$ represents the standard deviation of the Lorenz curves in the batch labeled by the index i .

A plot showing the distances ΔP_i versus the number of circuits contained in batch i can be seen in Figure 4.4. For ensembles using 5000 circuits or more, the vertical distance between the peak fluctuations is less than 10^{-3} . This variation is small enough to ensure that the Haar-8 benchmark line remains fixed within visual resolution. As a result, all Lorenz curve fluctuations used in these simulations were calculated using an ensemble of 5,000 circuits

After determining the appropriate ensemble size, we began investigating the number of gates required for a noiseless 8-qubit Agave device to produce fluctuations in Lorenz curves that closely align with the Haar-8 benchmark line, thereby achieving behavior consistent with universal sampling capabilities. The number

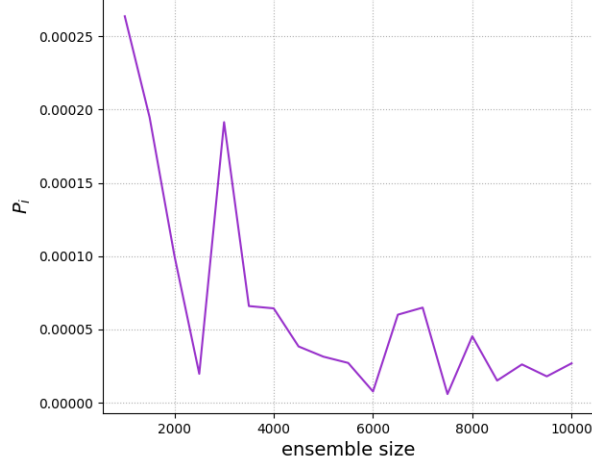


Figure 4.4: Plot of P_i (the vertical distance between the peak standard deviations of consecutive batches) versus the number of samples in the largest batch, showing the convergence of the Lorenz curve fluctuations to the asymptotic Haar-8 benchmark.

of gates ranged from 100 to 2000, increasing in increments of 100 gates. In order to simulate the topology of the Agave device, the connectivity was initially restricted to nearest neighbors. Figure 4.5 shows the results of these simulations. For clarity and to avoid overloading the graph with excessive information, it illustrates only the fluctuations for circuits with gate counts ranging from 100 to 1900, with increments of 200 gates.

The characteristic fluctuations approach the Haar-8 curve as the number of gates increases. This provides us with further evidence that the Haar-8 curve can be considered a lower limit for universal (noiseless) gate sets, and thus, serve as our reference for identifying a processor’s capacity to sample from the full Hilbert space.

For circuits composed of 100 and 300 gates, the fluctuations are significantly above those typical of Cliff-8. This behavior is similar to that of the 2-qubit phase-random diagonal gate circuits studied in Reference [54]. As was stated previously, this gate set cannot perform universal computation, but is not classically simulable in the strong sense. Hence, for this region, the results of the benchmarking procedure are inconclusive. At 500 gates, the Agave reaches a level of complexity comparable to the Cliff-8 benchmark. Though clearly distinguishable from each other, both curves are similar in shape and height. As the number of gates increases further to 700 and 900, fluctuations decrease below the Clifford mark and approach the Haar-8 limit. Note that fluctuations for 1100 gates nearly coincide with Haar-8. For 1300 gates or more, the Lorenz fluctuation curves become visually indistinguishable from Haar-8. These results suggest that a noiseless Agave processor is theoretically capable of efficiently sampling from a complex probability distribution, but achieving this would require a random quantum circuit comprising over 1000 noiseless gates.

To gain a more quantitative understanding of this phenomena, I computed the distance between the fluctuations for each number number of gates and the Haar-8 benchmark line, defined as

$$D_H = \sqrt{\sum_{k=1}^{2^n} \left(\text{std}[F(k)] - \text{std}[F_H(k)] \right)^2}. \quad (4.2)$$

Figure 4.6 shows a plot of D_H versus the number of gates in the random quantum circuits. As the size of the circuit increases, the distance to the Haar-8 line decreases, a behavior also observed in Figure 4.5.

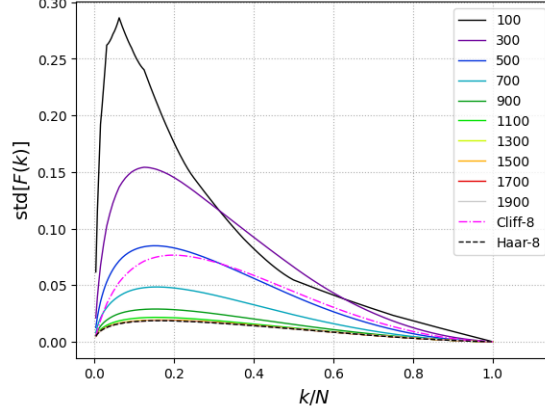


Figure 4.5: Fluctuations of the Lorenz curves for random quantum circuits with an increasing number of gates. The result suggests that a noiseless Rigetti Agave QPU would need to execute RQCs of 1300 gates or more to sample from its full Hilbert space.

Comparing the two figures, we observe that visual coincidence with the Haar-8 line occurs in the region where $D_H < 10^{-2}$.

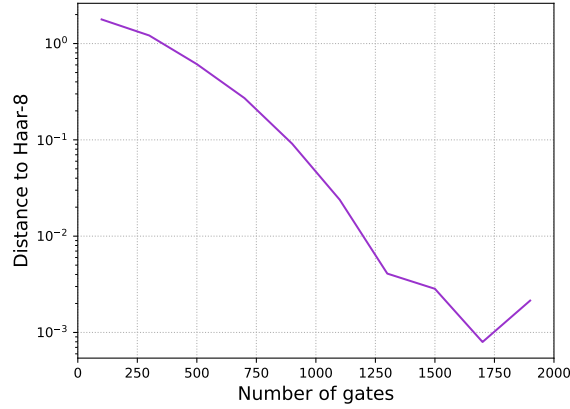


Figure 4.6: Distance to Haar-8, as defined in Equation 4.2, as a function of the number of gates in random quantum circuits composed of Rigetti's native gates. By comparing to Figure 4.5 it is possible to see that visual coincidence with the Haar-8 line occurs when the distance is below 10^{-2} .

To analyze the effect of qubit connectivity on a processor's capacity for complex computation, we calculated the fluctuations in Lorenz curves for RQCs of different gate depths on theoretical Agave-inspired processors with increasing qubit connectivity. By plotting D_H as a function of gate depth for these architectures, we were able to compare their performance. These results can be seen in Figure 4.7, where each configuration is labeled by its qubit connectivity number n_c . We observed that convergence to Haar-8 requires less gates when qubit connectivity is larger. Fluctuations for the $n_c = 6$ and $n_c = 7$ configurations coincide with Haar-8 for 700 gates. For the $n_c = 4$ geometry, the coincidence happens at 900 gates. All of the theoretical architectures show an improved capacity for handling complex computation when compared to the Agave, which requires circuits of over 1000 gates to achieve similar results.

The results show that the majorization-based indicator reacts to changes in qubit connectivity in a manner consistent with expectations. As the connectivity between qubits increases, fewer gates are required to produce outputs aligning with the Haar-8 benchmark line. This indicates a processor's improved ability to handle more complex quantum computations. This behavior aligns with the intuitive idea that more

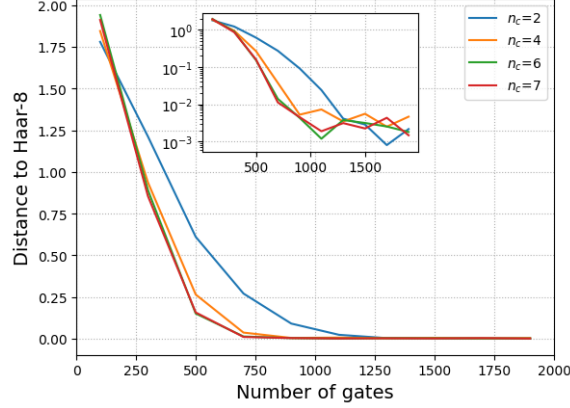


Figure 4.7: Distance to Haar-8, as defined in Equation (4.2), as a function of circuit depth for all 8-qubit QPU designs shown in Figure 4.2, with the Rigetti native gate set. The inset shows the same results in logarithmic scale.

connected processors can more efficiently process information, as higher connectivity allows for more flexible quantum operations and reduced circuit depth [70]. It is important to note that these results were obtained in the absence of noise, focusing purely on the architectural effects. These findings suggest that the majorization-based indicator is a useful tool for evaluating a processor’s performance, as it provides meaningful insights into how well a quantum processor can scale its computational capabilities in response to architectural improvements.

4.3 Noisy simulations

To develop a benchmarking method suitable for real quantum processors, it is crucial to understand how the majorization-based indicator responds to typical noise. In this work, I considered a simplified noise model in which one assumes all quantum gates operate perfectly, meaning a processor’s qubits only experience noise through interaction with the environment during idle periods. The two main types of noise these qubits can experience are described by the quantum operations of *amplitude damping* and *dephasing*.

Following Reference [33], a quantum operation ε is a completely positive trace-preserving (CPTP) map that acts on the state ρ of the affected qubit as

$$\varepsilon(\rho) = \sum_i E_k \rho E_k^\dagger. \quad (4.3)$$

To ensure trace preservation, the set of Kraus operators $\{E_k\}$ must satisfy the completeness relation

$$\sum_i E_k^\dagger E_k = \mathbb{1}. \quad (4.4)$$

Amplitude damping is a quantum noise channel that models energy dissipation in open quantum systems. It describes processes such as spontaneous photon emission by an atom or a spin system at high temperature reaching thermal equilibrium with its environment. Formally, the amplitude damping operation is given by

$$\varepsilon_A(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger \quad (4.5)$$

where

$$\begin{aligned} E_0 &= |0\rangle \langle 0| + \sqrt{1-p} |1\rangle \langle 1|, \\ E_1 &= \sqrt{p} |0\rangle \langle 1|, \end{aligned}$$

and p denotes the probability of error. This probability increases with the duration of the idle period according to

$$p = 1 - \exp(-t/T_1) \quad (4.6)$$

where T_1 is a characteristic time constant of the process. A detailed discussion of amplitude damping can be found in section 8.3.5 of Reference [33].

Dephasing, in contrast, is a quantum noise channel that models the loss of quantum information without energy dissipation. It describes processes such as photon scattering in a waveguide or the perturbation of atomic electronic states due to distant electrical charges. Like amplitude damping, dephasing can be described as a quantum operation of the form

$$\varepsilon_D(\rho) = \tilde{E}_0 \rho \tilde{E}_0^\dagger + \tilde{E}_1 \rho \tilde{E}_1^\dagger, \quad (4.7)$$

where the Kraus operators are given by

$$\begin{aligned} \tilde{E}_0 &= \sqrt{1-\tilde{p}} \mathbb{1}, \\ \tilde{E}_1 &= \sqrt{\tilde{p}} Z, \end{aligned} \quad (4.8)$$

with the probability

$$\tilde{p} = \frac{1}{2}(1 - \sqrt{1 - \exp(-2t/T_2)}). \quad (4.9)$$

Here, as before, T_2 is a characteristic time constant of the process. A detailed discussion of dephasing can be found in section 8.3.6 of Reference [33].

Observing the mathematical descriptions of the amplitude damping and dephasing processes, we can see that the constants T_1 and T_2 implicitly describe the evolution of error in these processes. These characteristic times are inversely related to the probability of error: higher values of T_1 and T_2 indicate a lower noise environment, whereas lower values correspond to higher noise levels. As such, these constants quantify the noise levels in a quantum processor, which is why they often appear in processor specifications. For instance, in the case of the Agave processor, Rigetti reports an average value of $T_1 = 13.38 \mu\text{s}$, while the average for T_2 is $15.05 \mu\text{s}$ [71].

The idle time for each qubit depends on the configuration of the circuit being implemented, as some gates can be applied simultaneously while others must wait their turn, leaving certain qubits idle. The duration of each qubit's idle period is determined by the gates applied to other qubits during its waiting time. Since the idle time of each qubit depends on the durations of gates applied to others, the ATOS QLM simulations for noisy quantum processors use the duration of a single-qubit gate as the unit of time when applying the noise channel.

For the Rigetti Agave processor, the average duration of one-qubit gates is 50 ns, and for two-qubit gates, it is 160 ns. Since the unit of time in the simulations is based on the duration of a single-qubit gate,

the duration of the two-qubit gate was approximated to 150 ns, the closest multiple of the one-qubit gate duration.

Our analysis considered two different scenarios: one in which only amplitude damping was present, and another where only dephasing occurred. This approach was used to isolate the effect of each type of noise on the system’s behavior, allowing for a clearer understanding of how each noise channel influences the performance of the quantum processor. By separately analyzing amplitude damping and dephasing, we could better identify the individual contributions to the overall noise profile and assess the processor’s resilience to each type of error.

The noise time scales T_1 and T_2 varied from $1\ \mu\text{s}$ to 1 ms, with the ensemble size fixed at 5000 circuits, each composed of 1500 gates. As demonstrated in the previous section, this circuit depth results in a fluctuation curve with a Euclidean distance from Haar-8 that is less than 10^{-2} . This distance leads to a curve that is visually indistinguishable from Haar-8 when an appropriate scale is chosen.

Figure 4.8 shows the results when only amplitude damping is present. As T_1 increases, corresponding to a decrease in the intensity of the noise, the fluctuation curves approach the Haar-8 benchmark line. For $T_1 = 1\ \mu\text{s}$, the fluctuation curve is significantly above the Cliff-8 line, and our benchmarking procedure is inconclusive. When T_1 is increased to $10\ \mu\text{s}$, the fluctuations reduce dramatically, and the curve shape changes. At this noise level, the benchmarking procedure remains inconclusive, as the curve does not align with either Haar-8 or Cliff-8. At $T_1 = 100\ \mu\text{s}$, the shape of the curve and the magnitude of the peak standard deviation align more closely with Haar-8, suggesting that the sampling capabilities are starting to resemble what is expected. Finally, for $T_1 = 1\ \text{ms}$, there is no apparent deviation from the Haar-8 curve. This result strongly suggests that this level of noise ensures the Agave processor’s capability to sample from its full Hilbert space is not compromised.

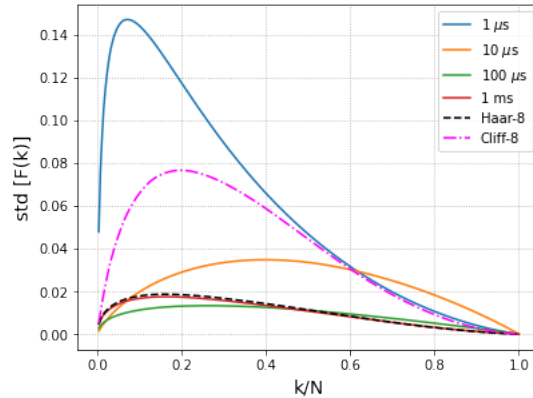


Figure 4.8: Fluctuations in Lorenz curves for an ensemble size of 5000 circuits and circuit depth of 1500 gates, characterizing the RQC outputs of a Rigetti Agave processor under the influence of amplitude damping.

Figure 4.9 presents the results for pure dephasing. Similar to the case of amplitude damping, the fluctuation curves approach the Haar-8 benchmark line as T_2 increases. However, for pure dephasing, the fluctuations are zero for $T_2 = 1\ \mu\text{s}$ and increase monotonically as the error decreases. This behavior can be explained by the Kraus operators for the dephasing channel, as given in Equation (4.8). The dephasing channel effectively nullifies the off-diagonal terms of the density matrix in the computational basis [72]. As a result, when T_2 is low, the output probabilities of random quantum circuits tend toward a uniform

distribution, which causes the standard deviation to be zero. This contrast between the fluctuation behaviors in pure dephasing and pure amplitude damping underscores that the key indicator of complexity lies in how much the fluctuations deviate from Haar-8, rather than their absolute size.

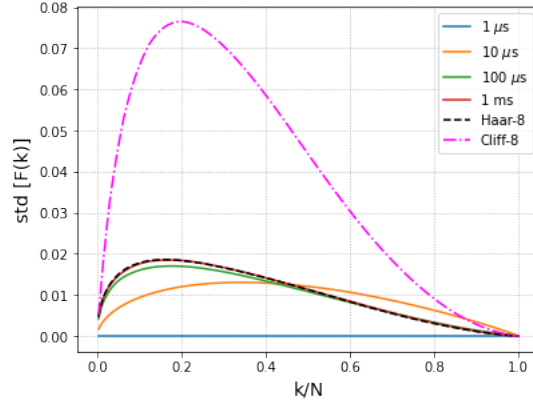


Figure 4.9: Fluctuations in Lorenz curves for an ensemble size of 5000 circuits and circuit depth of 1500 gates, characterizing the RQC outputs of a Rigetti Agave processor under the influence of dephasing.

When T_2 increases to $10 \mu s$, the fluctuations also increase, approaching the magnitude of those in the Haar-8 benchmark. However, the shape of the curve remains visibly different from the reference line, making it impossible to confirm that the Agave is still able to sample from its full Hilbert space. At $T_1 = 100 \mu s$, the curve aligns more closely with Haar-8, though there is still a noticeable difference in the magnitude of the peak fluctuations. At this noise level, the Agave begins to recover its ability to sample from its full Hilbert space. Finally, at $T_1 = 1 ms$, there is no apparent deviation from the Haar-8 curve, suggesting that the Agave processor can reliably sample from its full Hilbert space at this noise level.

To validate these results, we calculated the average purity of the output states (before measurement) and the average fidelity between the noisy and noiseless output states (for the same circuit) for all studied values of T_1 and T_2 . Figure 4.10 shows the variation of these metrics with increasing $T_{1(2)}$, corresponding to a decrease in noise levels. The behaviors of these metrics are similar for both the amplitude damping and pure dephasing noise profiles. For $T_{1(2)} < 10 \mu s$, noise is very strong, leading to very low values of purity and fidelity. For $10 \mu s \lesssim T_{1(2)} \lesssim 1 ms$, we observe an intermediate noise regime, where purity and fidelity increase rapidly from low to high values. For $T_{1(2)} \gtrsim 10 ms$, noise is very low, allowing for computations with near-unity fidelity and purity.

These results support the conclusions drawn from the benchmarking procedure and corroborate the correlation between deviations from the Haar-8 curve and the amount of noise the circuit experiences. Small deviations are associated with high purity and fidelity, while larger fluctuations in Lorenz curves correspond to decreases in both purity and fidelity. This suggests that majorization-based benchmarking can assess a quantum processor's resilience to noise. By observing how much noise the processor can tolerate before the cumulant fluctuations significantly deviate from the Haar curve, we can gauge its ability to handle complex computations and establish thresholds for noise levels that ensure its capability for reliable complex computation. This provides valuable insight into the processor's practical applicability in real-world scenarios, where noise is an inherent challenge.

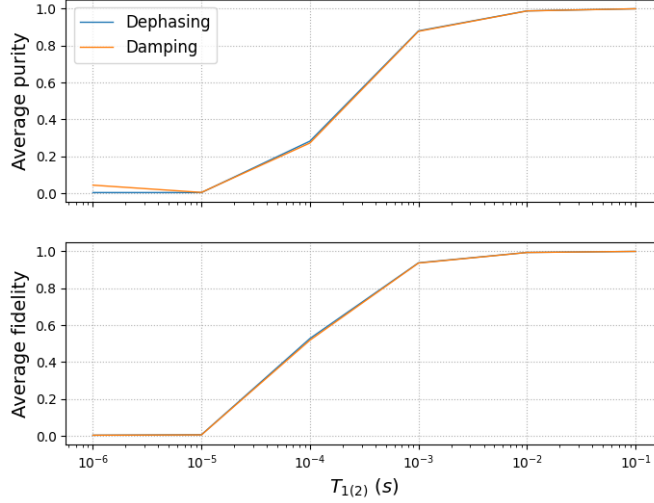


Figure 4.10: Average purity of the circuits’ output states (top) and average fidelity between noisy and noiseless output states (bottom) as a function of noise times T_1 and T_2 , for the Rigetti Q8 Agave processor. These results support the conclusions drawn from the benchmarking procedure.

An interesting observation is that the majorization-based indicator behaves differently for different types of noise, while the purity and fidelity metrics exhibit similar trends across both amplitude damping and dephasing. This contrast suggests the potential of the majorization-based indicator for distinguishing between noise profiles, offering additional insight into the processor’s resilience to different noise types. While purity and fidelity provide a general sense of the processor’s performance, the majorization-based criteria could help identify specific noise-induced behaviors that may otherwise be overlooked.

4.4 Conclusion

In the last two decades, RQC-based benchmarking protocols have established themselves as indispensable tools for evaluating the performance of quantum gates and processors. Notable examples include randomized benchmarking, quantum volume and cross-entropy benchmarking, each of which has been successfully implemented across the quantum computing industry to characterize and compare the performance of existing devices.

The theoretical framework behind protocols such as quantum volume and XEB reveals an intriguing property of universal RQCs: in the absence of noise, the probability $p(x)$ of sampling a bit-string x over an ensemble of quantum-circuits is distributed according to the Porter-Thomas distribution. This is a distinctly quantum behavior. In contrast, for a set of randomly selected classical functions mapping Boolean inputs, $p(x)$ follows the *uniform* distribution, meaning no output bit-string is favored. This quantum signature—rooted in the statistical properties of RQCs—serves as the foundation for many benchmarking protocols, including XEB and quantum volume.

While quantum volume and XEB leverage this property to estimate circuit fidelity, the majorization-based indicator focuses on discriminating between RQCs of different complexity classes. By evaluating a processor’s capacity for universal computation, the majorization-based benchmarking protocol provides insight into the complexity of operations that the device can perform, offering a complementary perspective on quantum device performance.

The numerical simulations presented in this chapter show that the majorization-based indicator is sensitive to multiple relevant quantities for quantum processor performance, such as noise, connectivity and the number of gates the device can effectively implement. It is correlated with purity and fidelity, which are desirable properties for a benchmarking metric. Furthermore, it requires only taking measurements on the computational basis, making it architecture independent and less experimentally costly to measure than directly computing purity and fidelity.

These characteristics make majorization-based benchmarking a promising method for assessing capacity for universal computation in near-term devices. However, to apply this benchmark to actual quantum processors, it is crucial to address the challenges posed by finite statistics—a topic I will explore in detail in the next chapter. Understanding these practical limitations will be crucial to ensuring the applicability of majorization-based benchmarking in real-world devices.

Chapter 5

Accounting for finite statistics

The promising results of Reference [2] motivate testing the majorization-based benchmarking procedure on actual quantum processors. In the previous chapter, we examined two key resources for characterizing complex outputs: the number of gates in the RQC and the number of circuits used to compute the standard deviation. However, before proceeding to experimental implementation, an additional factor must be studied: the influence of finite statistics.

In a real experiment, accessing the exact probability distribution of a RQC’s output is not feasible. Instead, researchers prepare an ensemble of identical circuits and record the relative frequencies of their outcomes. Larger ensembles yield frequency estimates that more closely approximate the probability distribution. However, in practical settings, resources are limited, raising the question of how many measurements are necessary to obtain reliable benchmarking results and how this requirement scales with the processor’s qubit count. Understanding and addressing this challenge is crucial for the successful application of majorization-based benchmarking to real quantum processors.

5.1 The challenge of finite statistics

Consider a quantum processor with n qubits. The measurement at the end of the quantum circuit projects the system’s state onto one of the 2^n possible states of the computational basis. Since quantum measurement is inherently probabilistic, the outcome is sampled from a probability distribution determined by the quantum state. The result is a bit string of length n , which can be represented as a 2^n -dimensional binary vector \mathbf{x} , where the component corresponding to the observed bit string is 1 and all other components are 0.

To estimate a quantum circuit’s output probability, we perform multiple experiments under identical conditions. Suppose we execute the same quantum circuit m times, each acting on the same initial state, yielding m measurement outcomes. The quantity

$$\text{Freq}(x) = \sum_{i=1}^m \frac{1}{m} y_x^i \quad (5.1)$$

represents the relative frequency of obtaining a certain state $|x\rangle$, where y_x^i is an indicator variable that is 1 if the output $|x\rangle$ is obtained in the i -th experiment and 0 otherwise. Since quantum measurements project the state onto one of the computational basis states, we have the constraint that if $y_x^i = 1$ for some basis state $|x\rangle$, then $y_{x'}^i = 0$ for all $x' \neq x$.

Since $\text{Freq}(x)$ is an empirical estimate of $\text{Prob}(x)$, we can use Hoeffding's inequality to bound its deviation from the true probability. According to Hoeffding's inequality, for a sum

$$S_m = X_1 + X_2 + \cdots + X_m, \quad (5.2)$$

where each X_i is an independent random variable satisfying $a_i \leq X_i \leq b_i$, we have

$$\Pr(|S_m - E[S_m]| \geq \epsilon) \leq 2 \exp \left(-\frac{2\epsilon^2}{\sum_{i=1}^m (b_i - a_i)^2} \right), \quad (5.3)$$

where $E[S_m]$ is the expected value of S_m [73].

The frequency $\text{Freq}(x)$ can be expressed as a sum of variables $X_1 + X_2 + \cdots + X_m$ such that

$$X_i = \frac{1}{m} y_x^i, \quad (5.4)$$

Since each X_i satisfies $0 \leq X_i \leq \frac{1}{m}$, we substitute into Hoeffding's inequality:

$$\Pr(|\text{Freq}(x) - \text{Prob}(x)| \geq \epsilon) \leq 2 \exp \left(-\frac{2\epsilon^2}{\sum_{i=1}^m (\frac{1}{m})^2} \right). \quad (5.5)$$

And as

$$\sum_{i=1}^m \left(\frac{1}{m} \right)^2 = m \left(\frac{1}{m} \right)^2 = \frac{1}{m}, \quad (5.6)$$

we obtain the final bound:

$$\Pr(|\text{Freq}(x) - \text{Prob}(x)| \geq \epsilon) \leq 2e^{-2\epsilon^2 m}. \quad (5.7)$$

This result shows that the probability of the empirical frequency deviating from the true probability by more than a fixed threshold ϵ decreases exponentially with the number of measurements. Consequently, increasing the number of measurements significantly improves the accuracy of our estimate.

However, this bound only accounts for the error in estimating a single component of the probability distribution. To obtain a reliable estimate of the full probability distribution, we need to ensure that all components are simultaneously bounded by a similar inequality. According to Boole's inequality, for any finite or countable set of events A_1, A_2, \dots, A_n ,

$$P \left(\bigcup_{i=1}^n A_i \right) \leq \sum_{i=1}^n P(A_i). \quad (5.8)$$

That is, probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events [74].

Since there are N possible measurement outcomes, we apply this inequality to control the probability that at least one estimate deviates beyond ϵ :

$$\Pr(\exists x \text{ such that } |\text{Freq}(x) - \text{Prob}(x)| \geq \epsilon) \leq 2Ne^{-2\epsilon^2 m}. \quad (5.9)$$

This result highlights that the required number of measurements m depends not only on the desired accuracy ϵ but also on the number of possible measurement outcomes. In the context of quantum computing, this poses a significant challenge, as the number of outcomes scales exponentially with the number of qubits, following $N = 2^n$.

Consequently, any practical complexity benchmark must remain viable even when the number of measurements is limited. Understanding how finite measurement statistics impact benchmarking results is therefore crucial for assessing its real-world applicability.

5.2 Generating Haar- n under finite statistics

Following a similar methodology as used in Reference [2], we used the quantum simulator KUATOMU to conduct simulated experiments. We begin by studying the behavior of Lorenz curve fluctuations corresponding to frequency distributions of Haar-random vectors for different values of m . These distributions, which we refer to as *Haar-like*, can be considered estimates of the Haar- n benchmark line in the finite statistics regime. The objective of this analysis is to determine how large m must be so that Haar-like curves converge to the Haar- n benchmark line, serving therefore as adequate estimates. My primary focus was to determine the qualitative scaling behavior of the number of measurements m necessary to achieve visual coincidence with the appropriate Haar- n benchmark line as the number of qubits n increased.

As n varies, it is essential to ensure that the ensemble size used to calculate the Lorenz curve fluctuations remains adequate across all values of n under consideration. This is particularly important because, as n increases, the fluctuations in Lorenz curves of Haar-random vectors tend to decrease, a phenomenon rooted in the *concentration of measure* [75]. In the context of quantum theory, this principle implies that certain properties of random quantum states become increasingly predictable as the dimension of the system grows [76].

To understand this behavior more deeply, recall that any pure n -qubit state can be represented as a point on an D -dimensional hypersphere, with $D = 2^{n+1} - 1$ [77]. This means that, as the number of qubits rises, the dimension of the hypersphere increases exponentially. For high dimensional-spaces, geometric and statistical properties often exhibit sharp concentration around their average values. This is formalized by Levy's lemma [75], which states that, for a specific class of functions acting on points on a high-dimensional sphere, the probability of deviating significantly from the mean value decreases exponentially with the dimension.

Levy's lemma applies to functions that are Lipschitz continuous. A function f is said to be Lipschitz continuous if there exists a constant L such that for all points x and y , the following inequality holds:

$$|f(x) - f(y)| \leq L |x - y|. \quad (5.10)$$

This condition ensures that the function does not vary too wildly, allowing Levy’s lemma to guarantee the strong concentration of its values around the mean. The k -th cumulants, defined in Equation (3.31), are continuous in this sense, since the function is bounded between zero and one. Therefore, as the dimension of the random vectors rises, the cumulants tend to concentrate more strongly around the mean, leading to a decrease in standard deviations.

An example of this phenomena for $n \in \{5, 6, 7, 8, 9\}$ is shown in Figure 5.1. In practice, this implies that the ensemble size needs to be sufficiently large to minimize the impact of statistical variability when calculating the standard deviation, even for larger values of n .

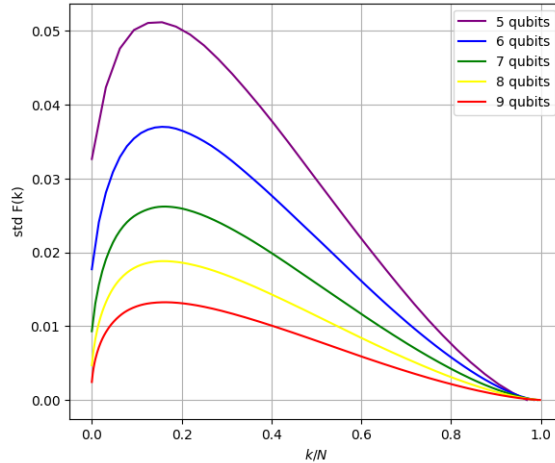


Figure 5.1: The Haar- n benchmark for different values of n . Fluctuations decrease as n increases, a consequence of the concentration of measure.

In our previous work, we determined the appropriate ensemble size using a procedure (detailed in section 4.2) where we generated a sequence of standard deviations, each computed using incrementally larger ensemble sizes. To quantify the convergence of these standard deviations, we calculated the vertical distance between the peak standard deviations of consecutive batches, defined as:

$$\Delta P_i = |P_i - P_{i-1}|, \quad (5.11)$$

where $P_i = \max[\text{std}[F(k)]_i]$ is the peak value of the standard deviation of the Lorenz curves in the batch labeled by the index i . This approach allowed us to identify the ensemble size at which further increases no longer significantly reduced the statistical variability, ensuring robust results without unnecessary computational cost.

To extend this analysis to circuits with $n \in \{5, 6, 7, 8, 9\}$ qubits, it is necessary to normalize ΔP_i for meaningful comparisons across different values of n . To this end, we divide each ΔP_i by the appropriate peak component of the standard deviation calculated using all of the generated vectors. In the previous study, we generated a total of 10^4 Haar-random 8-qubit probability vectors, dividing them into batches with an increasing number of samples. The first batch contained 500 samples, the second 1,000, and so on, with the final batch comprising all 10^4 samples. This procedure was repeated for $n \in \{5, 6, 7, 9\}$, so that each ΔP_i was normalized as:

$$\Delta_P = \frac{P_i - P_{i-1}}{P_{20}}, \quad (5.12)$$

where $i \in \{1, \dots, 20\}$ and P_{20} the peak component of the standard deviation calculated using all 10^4 vectors. This ensures that the distance ΔP_i is scaled relative to the peak of the appropriate benchmark line for each n . The resulting plot can be seen in Figure 5.2.

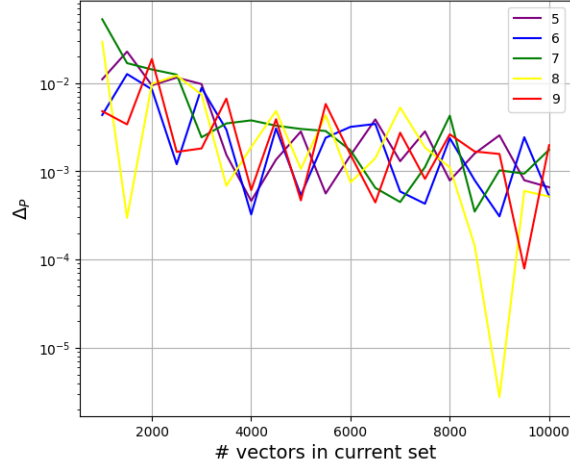


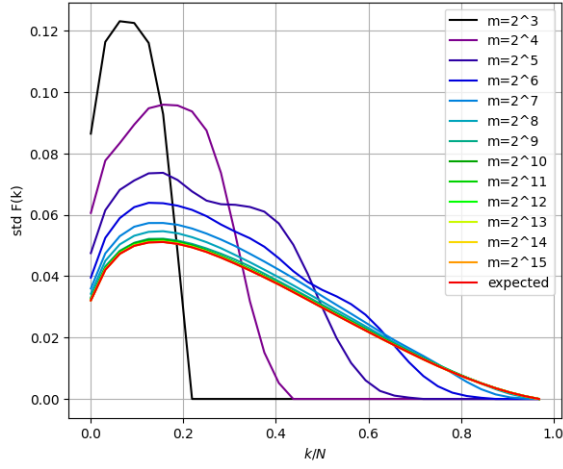
Figure 5.2: Normalized vertical distance ΔP between the peak values of fluctuations P_i and P_{i-1} , calculated using batches i and $i-1$, as a function of the number of circuits in batch i for a varying number of qubits. The results show that for ensembles of 5000 vectors or more, the variation between peaks is less than 1% of the corresponding Haar- n benchmark line, for all values of n under consideration.

The results indicate that the number of circuits required to generate a stable benchmark line is not significantly dependent on the number of qubits in the range under consideration. This is evidenced by the significant overlap in the curves depicting the variation of ΔP as the number of circuits increases. For ensembles of 5000 vectors or more, the variation between peaks is less than 1% of the corresponding Haar- n benchmark line for all considered values of n . Therefore, the ensemble size of 5000 vectors, used in the previous study, remains adequate even for $n = 9$.

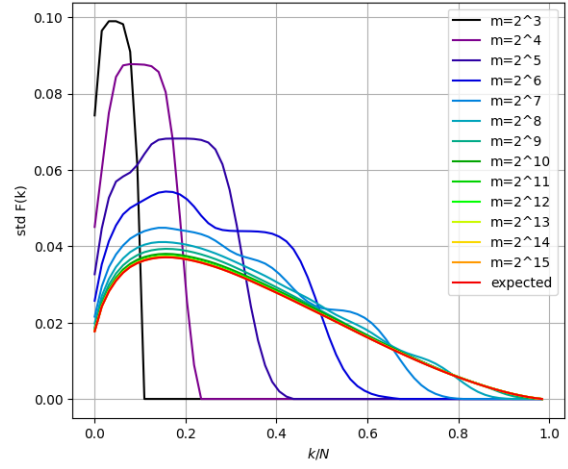
After determining the appropriate ensemble size, I computed the cumulant fluctuations for frequency distributions of n -qubit Haar-random pure states, calculated using an increasing number of measurements m . The number of measurements was increased in powers of 2, ranging from 2^3 to 2^{15} . This range was chosen to facilitate comparison between the size of the probability vector, given by 2^n , and the number of measurements required to achieve visual coincidence with the benchmark line. The results are illustrated in Figure 5.3.

To understand the qualitative behavior of the fluctuation curves as m increases, it is helpful to examine the averages of these Lorenz curves for varying m . For concreteness, Figure 5.4 shows the frequency distributions of 7-qubit Haar-random pure states as an example. While this figure illustrates the 7-qubit case, the conclusions drawn from it apply to systems with any number of qubits, as the behavior arises from general statistical principles governing the sampling of Haar-random states.

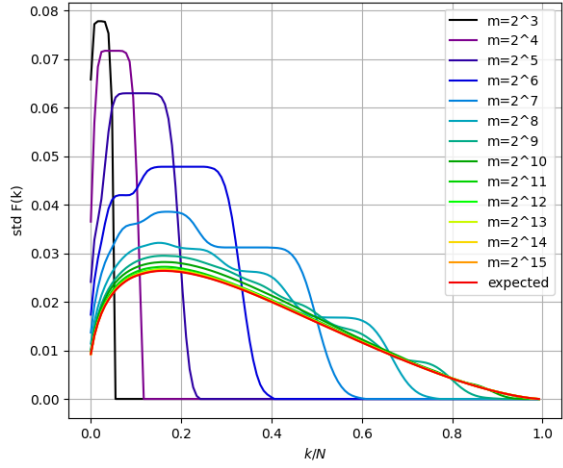
When m is small compared to $N = 2^n$, the frequency vector is sparse, with only a few non-zero components. As a result, the partial sum $F(k)$ reaches its maximum value (one) for small k , leading to a sharp peak in the standard deviation at small k/N , followed by an abrupt drop to zero as $F(k)$ remains constant. The Lorenz curves in this regime are characterized by a steep initial rise, reflecting the concentration of probability in a small number of outcomes. As m becomes large compared to N , the frequency vector becomes densely populated, and the partial sums $F(k)$ increase smoothly with k/N . The Lorenz curves converge to



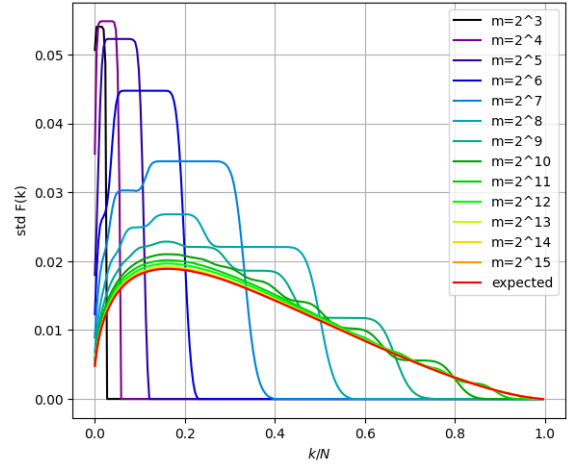
(a) $n = 5$



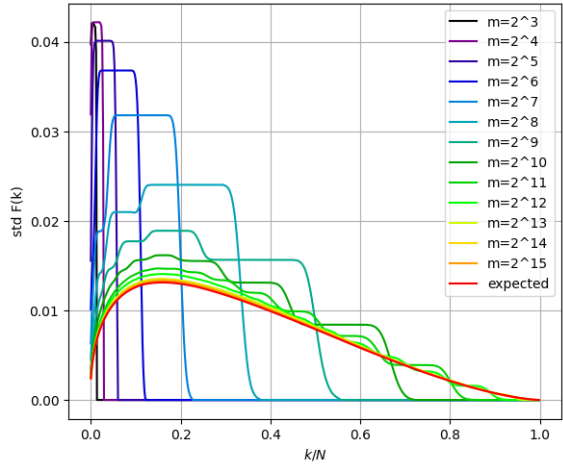
(b) $n = 6$



(c) $n = 7$



(d) $n = 8$



(e) $n = 9$

Figure 5.3: Fluctuations in the Lorenz curves for frequency distributions of n -qubit Haar-random pure states, calculated using an increasing number of measurements m . The curve labeled “expected” corresponds to the infinite statistics scenario. For $n = 5, 6$, visual coincidence occurs at $m \sim 2^{11}$, while for $n = 7, 8, 9$, it occurs at $m \sim 2^{13}$.

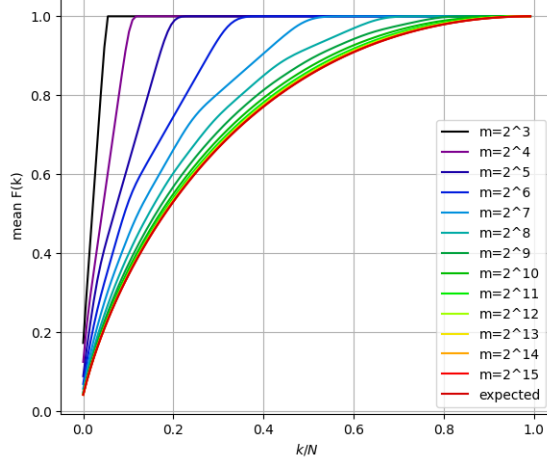


Figure 5.4: Average Lorenz curves for frequency distributions of 7-qubit Haar-random pure states, calculated using an increasing number of measurements m . For $m \in \{2^6, 2^7, 2^8\}$, the average Lorenz curves display a faceted behavior.

a characteristic curve that reflects the statistics of Haar-random vectors. This curve is determined by the underlying Haar measure and represents the typical distribution of probabilities for Haar-random states. Consequently, the standard deviation of $F(k)$ exhibits a smooth, monotonic behavior, consistent with the convergence to this characteristic curve. In the intermediate regime, where $m \sim N$, the frequency vector does not yet fully resolve the underlying probability distribution. Some outcomes are overrepresented, while others are underrepresented, leading to abrupt changes in the rate of increase of $F(k)$. This results in faceted Lorenz curves, with distinct linear segments corresponding to different regimes of k/N . The standard deviation of $F(k)$ in this regime is undulating, reflecting the transitional behavior of the frequency vector as it moves from sparse to dense sampling.

Since the minimum m required for overlap with the Haar- n benchmark line cannot be easily determined by visual inspection, we instead consider the Euclidean distance between the frequency distributions of n -qubit Haar-random pure states and the ideal Haar- n curve. This distance, denoted $D_H(n)$, is computed as given by Equation (4.2). A plot of $D_H(n)$ across varying qubit counts is shown in Figure 5.5. The range of m was limited to $m \in \{2^8, 2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}\}$ to ensure greater resolution in identifying the point where $D_H(n) < 10^{-2}$ for each value of n . For $n = \{5, 6, 7, 8, 9\}$, the intercepts occur at $m \in [250, 1000], [500, 1000], [1000, 2000], [2000, 3000]$ and $[4000, 6000]$, respectively.

This scaling behavior poses a significant challenge to the applicability of the majorization-based benchmarking protocol as proposed in Reference [2]. It suggests that, for larger devices, the number of measurements required to generate the Haar- n benchmark line and demonstrate the capability of sampling from the full Hilbert space may be impractically large. This highlights the need to modify the protocol to ensure its feasibility in real-world applications.

To identify potential modifications, we study the frequency distributions of Clifford circuits for varying values of m and across different qubit counts. By comparing these Clifford-like curves to the corresponding Haar-like ones, we aim to determine whether the requirement of reaching the asymptotic Haar- n curve can be relaxed. This analysis is the focus of the next section.

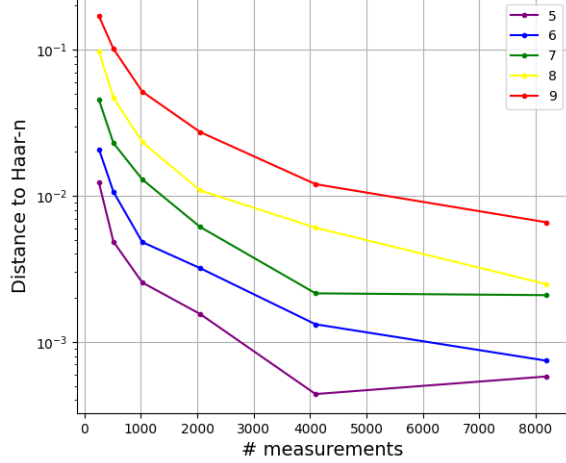


Figure 5.5: Distance $D_H(n)$ as a function of the number of measurements m used to calculate the frequency distributions.

5.3 Differences between Haar- n and Cliff- n under finite statistics

Next, we study the behavior of fluctuations in Lorenz curves for frequency distributions of Cliff- n circuits, or *Clifford-like* distributions, with the aim of determining how large m must be for Haar-like and Clifford-like curves to be distinguishable. This builds on our earlier work [2], where we defined the asymptotic Cliff- n curve as a reference for sampling capabilities that can be efficiently simulated by classical computers. By computing these distributions for varying values of m and across different qubit counts, we aim to uncover how the distinguishability between Haar-like and Clifford-like curves evolves as m increases, shedding light on the boundary between simulable and non-simulable computation in the context of finite statistics.

As in our previous work [2], the Clifford circuits consisted of 500 gates from the gate set $\{\text{CNOT}, \text{H}, \text{S}\}$, starting from a random pure separable state and without qubit connectivity constraints. The number of measurements was increased in powers of 2, ranging from 2^3 to 2^{15} , and the values of n were chosen as $n \in \{5, 6, 7, 8\}$. Plots of the Lorenz curve fluctuations for these frequency distributions, corresponding to each value of n , are shown in Figure 5.6.

Compared to the Haar-like fluctuations shown in Figure 5.3, the Clifford-like fluctuation curves exhibit distinct qualitative behavior as m increases. While the fluctuations for Haar-like distributions decrease monotonically, eventually approaching their asymptotic value, the fluctuations for Clifford-like distributions initially decrease before rising toward their asymptotic value. For $n \in \{5, 6, 7\}$, we see the peak fluctuations begin to rise at $m = 64$, while for $n = 8$, this behavior starts at $m = 128$.

This result is favorable because, as the peaks in Clifford-like fluctuations rise while those in Haar-like fluctuations fall, the two behaviors tend to diverge, making them easier to distinguish. The fact that the decreasing behavior of Clifford-like fluctuations stops relatively early (at small m) further enhances this distinguishability.

To analyze this behavior quantitatively, we introduce the normalized difference $\frac{d}{d_0}$, which compares the peaks of Haar-like and Clifford-like fluctuation curves. Here, d_0 represents the difference between the

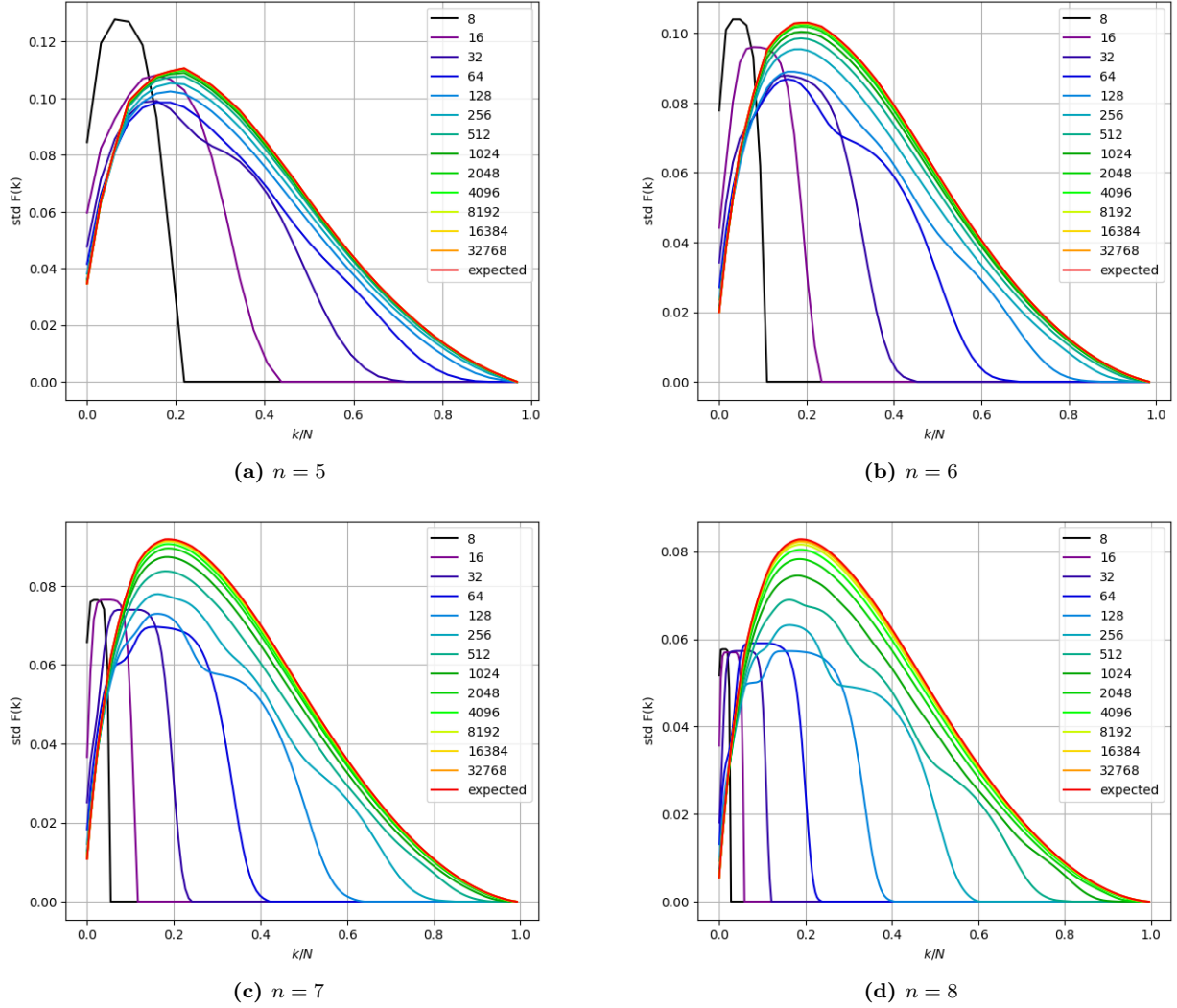


Figure 5.6: Fluctuations in the Lorenz curves for frequency distributions of n -qubit randomized Clifford circuits, calculated using an increasing number of measurements m . The curve labeled “expected” corresponds to the infinite statistics scenario.

peaks of the ideal fluctuation curves, defined as:

$$d_0 = \max(\text{std}[F_C(k)]) - \max(\text{std}[F_H(k)]). \quad (5.13)$$

where $F_C(k)$ and $F_H(k)$ denote the cumulants of the of the Cliff- n and Haar- n distribution, respectively. The difference d is defined analogously for finite statistics distributions. A visual representation of d and d_0 for 7 qubits and $m = 2^5$ is provided in Figure 5.7.

It is important to note that this quantity will be negative when the Clifford-like distribution lies below the Haar-like distribution. This intentional design not only signals when the Clifford curve is under the Haar curve but also allows us to observe the transition from decreasing to increasing behavior in the Clifford-like curves. Figure 5.8 shows a plot of $\frac{d}{d_0}$ for $n \in \{5, 6, 7, 8\}$. While the difference was calculated for all values of m , the plot focuses on the range $m = 2^3$ to $m = 2^{10}$, as we are particularly interested in the behavior of the curves at lower values of m .

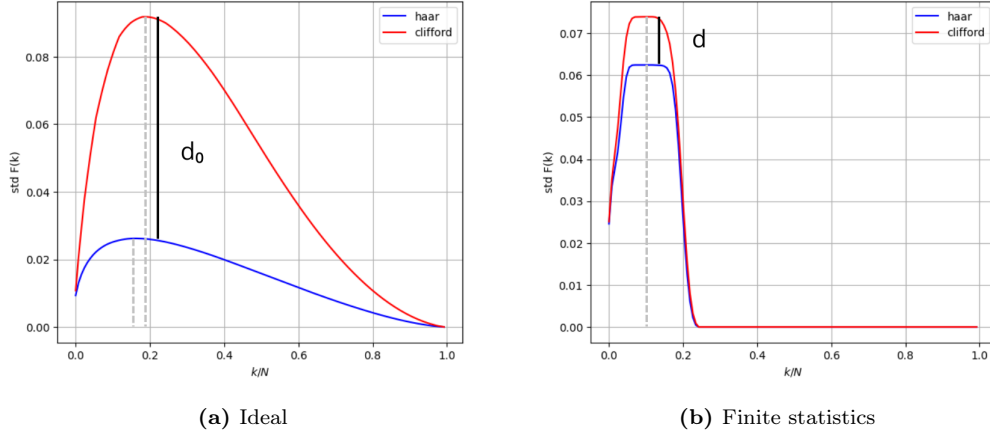


Figure 5.7: Fluctuations in Lorenz curves for Haar-7 and Cliff-7 probability distributions, compared to the frequency distributions of Haar-like and Clifford-like curves for $m = 128$, illustrating the variation in vertical peak distances.

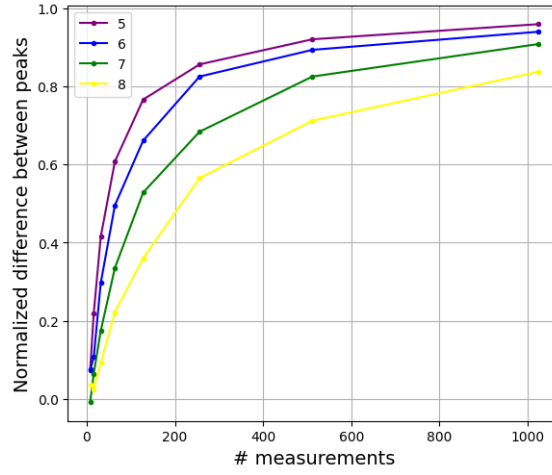


Figure 5.8: Normalized difference between peaks of fluctuation curves for Clifford-like and Haar-like frequency distributions, calculated for an increasing number of measurements.

For all values of n , the normalized difference $\frac{d}{d_0}$ rises monotonically, approaching its asymptotic value of 1. However, the number of measurements m required for $\frac{d}{d_0}$ to approach this asymptotic value increases with larger n . A promising finding is that, for all n considered, the number of measurements needed to distinguish between the curves is significantly less than the dimension 2^n of the corresponding probability vector. For example, consider the case of $n = 8$. As shown in Chapter 4, the difference between the Cliff-8 and Haar-8 curves is approximately $d_0 \sim 0.06$ (See Figure 4.1). At $d = 0.01$, or $\frac{d}{d_0} \sim 0.16$, the curves are already easily distinguishable. Figure 5.8 demonstrates that $\frac{d}{d_0} > 0.2$ for $m = 2^6$, meaning distinguishability is achieved for m considerably smaller than 2^n .

This result demonstrates that the majorization-based indicator not only captures differences in complexity in the finite-statistics regime but also achieves this without requiring the generation of the asymptotic Haar-random benchmark line. This is particularly encouraging from a scalability perspective, as it suggests that the indicator remains practical even as the system size grows.

5.4 Conclusion

Combining the results from this study with those of Chapter 4, we conclude that achieving the Haar- n benchmark line is likely unfeasible for most current quantum devices. Realizing the asymptotic behavior of fluctuations in the Lorenz curves of Haar-random vectors on a noiseless 8-qubit processor would require thousands of high-depth unitary circuit realizations and on the order of 2^{14} measurements. These challenges are further exacerbated by the presence of noise.

Moreover, the requirement to generate the Haar- n benchmark line is not only excessively stringent but also overlooks a critical insight: fluctuations in the Lorenz curves of Haar-like and Clifford-like frequency distributions are distinguishable well before the asymptotic behavior of the curves is fully realized. This finding suggests that the majorization-based indicator can assess differences in complexity with a higher resolution than previously assumed, offering a more practical and nuanced tool for benchmarking quantum devices.

The primary limitation of the majorization-based indicator lies in its heuristic nature. While visual inspection of fluctuations in Lorenz curves plotted against k/N allows for the distinction between Haar-like and Clifford-like behaviors, there is currently no precise mathematical formula to quantify this distinction. A comprehensive study accounting for the simultaneous influence of the number of gates, measurements, and experimental realizations required to differentiate between Haar-like and Clifford-like fluctuation curves would necessitate analyzing a large number of plots. This process is not only labor-intensive but also introduces uncertainty, as it is difficult to ensure that all images are evaluated using consistent criteria.

This type of heuristic challenge, where visual inspection and pattern recognition play a central role, has been successfully addressed in other fields using machine learning algorithms [78,79]. A well-known example is the classification of handwritten digits or other images, where machine learning models are trained to identify patterns and make distinctions that are difficult to quantify with explicit mathematical formulas. Inspired by these successes, we propose the use of a machine-learning classifier to analyze the fluctuations in Lorenz curves and distinguish between Haar-like and Clifford-like behaviors. By leveraging the ability of machine learning to extract subtle patterns from complex data, we aim to overcome the limitations of manual inspection and provide a more systematic and reliable approach to this problem.

In the next chapter, we will introduce the machine learning algorithm chosen for this study: Support Vector Machines (SVMs). We will discuss the rationale behind selecting SVMs, including their ability to handle high-dimensional data and their effectiveness in binary classification tasks. A detailed explanation of how SVMs work will be provided, along with the specific strategy we employed to train and evaluate the model for distinguishing Haar-like and Clifford-like behaviors in Lorenz curves. Finally, we will present and analyze the results of our study, demonstrating the potential of this approach to overcome the heuristic limitations of manual inspection and provide a more robust framework for benchmarking quantum devices.

Chapter 6

An implementable benchmarking proposal

The results of Chapter 5 show that the majorization-based indicator is capable of capturing the differences between Haar-like and Clifford-like frequency distributions well before the asymptotic behavior of the Lorenz curve fluctuations is fully realized. This finding motivates a comprehensive analysis into the distinguishability of Haar-like and Clifford-like curves across various experimental setups—with varying numbers of gates, measurements and experimental realizations. Such a study would necessitate analyzing a large number of plots, a process which is not only labor-intensive, but also introduces uncertainty regarding the consistency of the criteria used to evaluate the images.

To address this challenge, we employed a machine learning-based strategy. A support vector machine (SVM) was trained to distinguish between Haar-like and Clifford-like fluctuation curves, with the objective of determining the minimum number of measurements, gates, and circuits necessary for the SVM to discriminate between Haar-like and Clifford-like fluctuation curves with a success rate above a predefined threshold. This approach allowed us to gain insights into the resources necessary to reliably differentiate between quantum and classical sampling behaviors in the finite statistics regime.

6.1 Motivation

In recent years, numerous machine learning algorithms have been developed and successfully applied to pattern recognition tasks [78]. Among these, the Support Vector Machine (SVM) stands out as particularly well-suited for our problem. Introduced by Cortes and Vapnik in 1995 [80], SVMs classify data by identifying the optimal hyperplane that maximizes the margin between classes. As a deterministic model, the SVM offers a significant advantage in our study, which already involves a multitude of statistically varying quantities. This determinism ensures a stable and interpretable framework for distinguishing between Haar-like and Clifford-like behaviors, complementing the heuristic nature of our majorization-based indicator.

Furthermore, SVMs are capable of handling high-dimensional data efficiently using a technique called the *kernel trick*. This technique relies on a positive-definite kernel function, which maps the training data onto

a higher-dimensional feature space without explicitly computing the coordinates of the data in that space. Instead, the kernel function represents the data through pairwise comparisons that measure the similarity between observations, much like comparing the distances between points in a geometric space. Once the data is transformed, the SVM identifies the optimal hyperplane that maximizes the margin between the classes. This ability to operate effectively in high-dimensional spaces makes SVMs particularly well-suited for handling the Lorenz curve fluctuation vectors analyzed in this study, as their dimension scales exponentially with the number of qubits.

Finally, SVMs are also versatile in their learning paradigms, supporting both *supervised* and *unsupervised* approaches. The support vector machine was originally proposed as a supervised learning algorithm, designed to learn from labeled training data to predict outcomes for new, unseen samples [80]. However, formulations such as the one-class SVM can be used for unsupervised learning, where the goal is to model the underlying distribution of the data and identify anomalies or outliers [81, 82]. The particular formulation chosen for this study, Schölkopf et. al’s ν -SVM [83], can be used for both supervised and unsupervised learning tasks [81, 83].

This enabled the comparison of two different approaches for data classification. In the supervised setting, the SVM is trained on both Haar-like and Clifford-like curves, learning to classify and distinguish between the two. In contrast, the unsupervised setting involves training the SVM exclusively on Haar-like curves, modeling their distribution to detect deviations from universal behavior. While the supervised approach focuses on classifying data as either Haar-like or Clifford-like, the unsupervised approach seeks to identify features unique to RQCs composed of universal gate sets. Although the difference between these tasks is subtle, it enables distinct inferences about the data, offering complementary insights into the complexity of the analyzed outputs.

This chapter presents a modified version of the majorization-based benchmarking protocol, using machine learning to distinguish between RQC outputs of different classes of complexity. It begins by exploring the mathematical foundations of SVMs, establishing the theoretical framework necessary to understand their application in this study. Next, we present the modified benchmarking protocol, followed by an analysis of simulated experiments. The results demonstrate the experimental viability of majorization-based benchmarking and highlight the potential of machine learning to enhance its resolution and applicability.

6.2 Support vector machines

Support vector machines are a class of machine learning models primarily used for classification and regression tasks. They have been successfully applied to a wide range of real-world problems, including text categorization, image recognition [79], and fraud detection [84]. In addition, SVMs have demonstrated significant utility in the life sciences, such as in the automated classification of microarray gene expression profiles, which can aid medical professionals in cancer diagnosis and treatment planning [85].

This section explains the fundamental mathematical principles of SVMs, providing the necessary foundation for understanding their role in our study. It begins by tackling the problem of binary classification for linearly separable data, followed by a formal mathematical treatment of the kernel trick. These concepts enable us to understand the specific formulation chosen for this study—Schölkopf et al.’s ν -SVM [83]—and its applicability to both supervised and unsupervised learning tasks. The derivations in this section largely

follow the approach presented in Reference [78], with additional insights drawn from Reference [86], an online lecture series that provides a comprehensive introduction to this topic.

The simplest formulation of the support vector machine addresses the problem of binary classification for linearly separable data. The goal of this type of classification problem is to find a discriminant function $y : \mathbb{R}^D \rightarrow \mathbb{R}$ that takes a D -dimensional input vector \mathbf{x} and assigns it to one of two classes. The simplest representation of such a function is given by

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b, \quad (6.1)$$

where \mathbf{w} is called a *weight* vector, b is called the *bias*. An input vector is assigned to class \mathcal{C}_1 if $y(\mathbf{x}) \geq 0$, and to class \mathcal{C}_2 otherwise. The equation $y(\mathbf{x}) = 0$ defines a hyperplane which serves as the decision boundary separating the two classes.

To develop an intuitive understanding of the SVM's fundamental principles, we begin by examining a simple scenario: the classification of linearly separable 2-D data with non-overlapping class distributions. In this case, $y(\mathbf{x}) = 0$ represents a line that separates the training data into two regions of the \mathbb{R}^2 plane. There exist infinitely many combinations of \mathbf{w} and b that can separate the training data, corresponding to lines with varying slopes and intercepts. However, not all such separating lines perform equally well as classifiers.

A desirable classifier should exhibit robustness: when a small amount of noise is introduced to the dataset, the classification of the data points should remain unchanged. Geometrically, this implies that the points should remain on the same side of the hyperplane as they were prior to the addition of noise. The hyperplane that best achieves this robustness is the one that maximizes the margin—the distance between the hyperplane and the closest data points of each class. Intuitively, a larger margin provides a buffer against noise, reducing the risk of misclassification and improving the model's generalization to unseen data. This concept is illustrated in Figure 6.1.

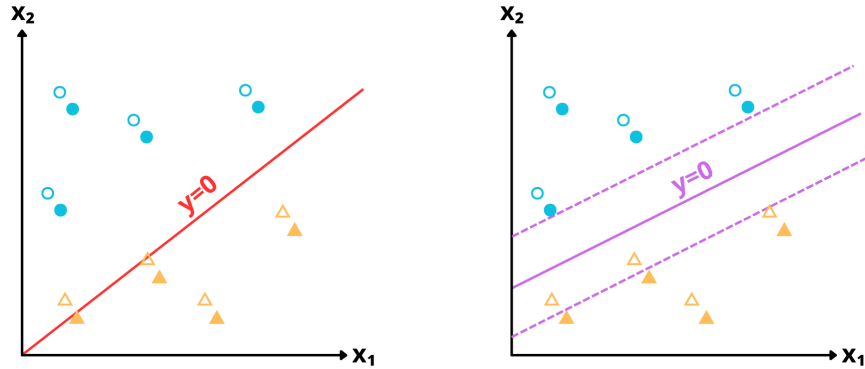


Figure 6.1: Diagram illustrating two choices of decision boundaries (lines) for separating data in the \mathbb{R}^2 plane. Solid shapes (blue circles and yellow triangles) represent the initial data, while outlined shapes represent the data after the addition of noise. For the line on the left, the addition of noise causes some data points to shift across the boundary, resulting in misclassification. For the line on the right, the presence of a margin prevents misclassification, demonstrating the importance of margin maximization in robust classification.

While we have illustrated this concept in two dimensions for intuitive understanding, it generalizes naturally to data of arbitrary dimensionality. In higher-dimensional spaces, the line becomes a hyperplane that separates the data into two regions, and the margin remains the minimal distance between the hyperplane

and the closest data points of each class. Maximizing this margin ensures robustness and generalization, regardless of the dimensionality of the input space.

This approach is known as the **hard-margin SVM** and is suitable for the classification of linearly separable data of any dimensionality, provided that the class distributions do not overlap. In the next section, we will discuss the mathematical formalism of the hard-margin SVM in detail, which will allow us to derive meaningful insights into the geometric properties of the problem and establish the theoretical foundation for extending the method to cases involving non-linearly separable data and overlapping class distributions.

6.2.1 Hard margin SVM

Consider a binary classification problem with a discriminant function given by:

$$y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) . \quad (6.2)$$

The function assigns a class label of $+1$ or -1 to each input. Notably, the classifier's behavior remains unchanged if \mathbf{w}^T and b are scaled by any constant $a > 0$, since

$$\text{sign}(a(\mathbf{w}^T \mathbf{x} + b)) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) . \quad (6.3)$$

To remove this degree of freedom and simplify the optimization problem, a scaling constraint is imposed on \mathbf{w} and b . The standard practice is to require the decision hyperplane to satisfy the *canonical representation*, defined as:

$$l_n y(\mathbf{x}_n) = l_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 , \quad (6.4)$$

where $\mathbf{x}_1, \dots, \mathbf{x}_N$ is the training set with corresponding labels $l_1, \dots, l_N \in \{-1, 1\}$. Here, l_n represents the class label of the n -th training sample, \mathbf{x}_n . The inequalities above are referred to as the *constraints* of the optimization problem. A constraint is said to be *active* for a data point if the equality holds, and *inactive* otherwise.

With the canonical representation established, we now turn to the concept of the *margin*, which is central to the formulation of the support vector machine. The margin is defined as the distance between the decision hyperplane and the closest data points of each class. The size of the margin, ρ , can be computed as the distance between the closest points on either side of the decision boundary, projected onto the direction of the decision boundary, which is given by the weight vector \mathbf{w} . This statement can be visualized in Figure 6.2, which illustrates the example in which the data is 2-dimensional.

Let \mathbf{x}^+ and \mathbf{x}^- denote the closest data samples on each side of the margin, with labels $+1$ and -1 , respectively. By the canonical representation, these points satisfy:

$$\mathbf{w}^T \mathbf{x}^\pm + b = \pm 1 . \quad (6.5)$$

The margin size is given by

$$\rho = \frac{1}{2}(\mathbf{x}^+ - \mathbf{x}^-) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T \mathbf{x}^+ - \mathbf{w}^T \mathbf{x}^-}{2 \|\mathbf{w}\|} . \quad (6.6)$$

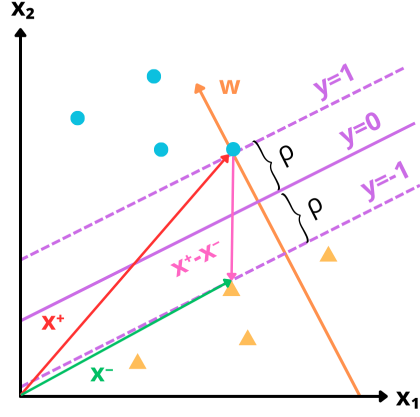


Figure 6.2: Illustration of the decision boundary and margin for a 2-D example. The margin ρ is the perpendicular distance between the hyperplane and the closest points, projected onto the weight vector \mathbf{w} .

Substituting the conditions given by Equation (6.5), we obtain

$$\rho = \frac{1 - b - (-1 - b)}{2 \|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}. \quad (6.7)$$

The objective of the hard-margin SVM is to maximize ρ with respect to \mathbf{w} and b . For mathematical convenience, this is reformulated as the following optimization problem:

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2, \quad (6.8)$$

subject to

$$l_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1. \quad (6.9)$$

This formulation is called the primal hard-margin SVM problem. While the primal problem is intuitive and directly optimizes the margin, deriving the dual optimization problem is essential to uncover the most important properties of the SVM, such as the role of support vectors and the application of the kernel trick. Moreover, as the SVM optimization problem is a quadratic optimization problem, it satisfies strong duality [79]. Thus, solving the dual problem provides the same optimal decision boundary as the primal formulation.

To derive the dual problem, we introduce the Lagrange multipliers a_1, \dots, a_N , each satisfying $a_n \geq 0$. The Lagrangian for the primal problem is given by:

$$L(\mathbf{w}, b, a_n) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{l_n(\mathbf{w}^T \mathbf{x}_n + b) - 1\}. \quad (6.10)$$

To solve the optimization problem, the Lagrangian must be minimized with respect to \mathbf{w} and b , and maximized with respect to the Lagrangian multipliers. Let $\nabla_{\mathbf{w}}$ denote the gradient of L with respect to \mathbf{w} , defined as

$$\nabla_{\mathbf{w}} L := \left(\frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_D} \right). \quad (6.11)$$

Setting $\nabla_{\mathbf{w}} L = 0$ yields

$$\mathbf{w} = \sum_{n=1}^N a_n l_n \mathbf{x}_n, \quad (6.12)$$

and $\frac{\partial L}{\partial b} = 0$ leads to

$$\sum_{n=1}^N a_n l_n = 0. \quad (6.13)$$

These results allow us to rewrite the Lagrangian solely in terms of the Lagrange multipliers $\mathbf{a} = (a_1, \dots, a_N)$, simplifying the optimization problem. This reformulation is called the dual Lagrangian.

First, we use Equation (6.12) to express the squared norm of \mathbf{w} in terms of the dual variables a_n :

$$\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w} = \sum_{n=1}^N a_n l_n \mathbf{w}^T \mathbf{x}_n. \quad (6.14)$$

Next, we rewrite a portion of the Lagrangian to simplify subsequent calculations. Specifically, consider the following term:

$$\sum_{n=1}^N a_n \{l_n (\mathbf{w}^T \mathbf{x}_n + b) - 1\} = \sum_{n=1}^N a_n l_n \mathbf{w}^T \mathbf{x}_n + \sum_{n=1}^N a_n l_n b - \sum_{n=1}^N a_n. \quad (6.15)$$

Substituting Equations (6.13) and (6.14) into this expression, we obtain:

$$\sum_{n=1}^N a_n l_n \mathbf{w}^T \mathbf{x}_n + \sum_{n=1}^N a_n l_n b - \sum_{n=1}^N a_n = \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n. \quad (6.16)$$

This result allows us to rewrite the Lagrangian in a simplified form:

$$L(\mathbf{w}, a_n) = \frac{1}{2} \|\mathbf{w}\|^2 - \|\mathbf{w}\|^2 + \sum_{n=1}^N a_n = -\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{n=1}^N a_n. \quad (6.17)$$

Finally, using (6.12) once more, we derive the dual Lagrangian:

$$\tilde{L}(a_n) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m l_n l_m \mathbf{x}_n^T \mathbf{x}_m, \quad (6.18)$$

where the Lagrange multipliers a_n are subject to the constraints

$$\begin{cases} a_n \geq 0, & n = 1, \dots, N, \\ \sum_{n=1}^N a_n l_n = 0. \end{cases}$$

The first condition, $a_n \geq 0$, ensures that the Lagrange multipliers are non-negative, which is a requirement for dual feasibility in constrained optimization. The second condition, $\sum_{n=1}^N a_n l_n = 0$, ensures that the solution respects the linear constraint derived from the primal problem, as previously shown in Equation (6.13). Together, these constraints guarantee that the dual solution is both feasible and consistent with the primal problem.

To fully characterize the optimal solution, the solution must also satisfy the **Karush-Kuhn-Tucker (KKT) conditions**, which include the above constraints and add further requirements for optimality. For this problem, the KKT conditions are given by:

$$\begin{cases} a_n \geq 0, \\ l_n y(\mathbf{x}_n) - 1 \geq 0, \\ a_n \{l_n y(\mathbf{x}_n) - 1\} = 0. \end{cases}$$

The third condition, known as *complementary slackness*, has an important geometric interpretation. It states that for each data point \mathbf{x}_n , either the Lagrange multiplier a_n is zero, or the point lies exactly on the margin boundary ($l_n y(\mathbf{x}_n) = 1$). This implies that only a subset of the data points—those for which $a_n > 0$ —actively contribute to the solution. These points, which lie on the margin boundary, are called **support vectors**. All other points, for which $a_n = 0$, do not influence the final decision boundary. This is the property which gives the support vector machine its name.

To better visualize the effect of this condition on the model, we use Equation (6.12) to rewrite the discriminant function solely in terms of a_n and the bias term b :

$$y(x) = \text{sign} \left(\sum_{n=1}^N a_n l_n \mathbf{x}_n^T \mathbf{x} + b \right). \quad (6.19)$$

This equation demonstrates that training samples with $a_n = 0$ do not contribute to predictions for new data points \mathbf{x} . Only the support vectors, which satisfy $l_n y(x_n) = 1$, contribute to the final decision boundary. This property is known as *sparsity*, and is computationally advantageous, as it reduces the complexity of the model by focusing only on the most informative training samples.

Now, to complete the formulation of the support vector machine, it remains to derive an expression for the bias term b . This term is crucial for ensuring that the decision boundary is correctly positioned relative to the support vectors. Starting from the condition for support vectors, we can derive a numerically stable solution for b that accounts for all support vectors in the dataset.

Substituting Equation (6.19) into the condition for support vectors yields

$$l_n \left(\sum_{m \in S} a_m l_m l_n \mathbf{x}_n^T \mathbf{x}_m + b \right) = 1, \quad (6.20)$$

where S denotes the set of indices corresponding to the support vectors. Since all $l_n \in \{-1, 1\}$, all l_n satisfy the property $l_n^2 = 1$. Thus, we can simplify the above equation by multiplying through by l_n :

$$\sum_{m \in S} a_m l_m \mathbf{x}_n^T \mathbf{x}_m + b = l_n. \quad (6.21)$$

A numerically stable solution for b can be obtained by averaging over all support vectors. Letting N_S be the total number of support vectors, this solution is given by

$$b = \frac{1}{N_S} \sum_{n \in S} \left(l_n - \sum_{m \in S} a_m \mathbf{x}_n^T \mathbf{x}_m \right). \quad (6.22)$$

The bias term is computed after the a_n are determined from the dual optimization problem. This result ensures that the bias term b is computed in a way that reduces variance among support vectors, which is crucial for numerical stability and generalization. The averaging procedure mitigates the effects of outliers and ensures robustness in the solution.

6.2.2 Soft margin SVM

Thus far, we have operated under the assumption of linearly separable, non-overlapping class distributions. However, real-world datasets often exhibit overlapping class-conditional distributions due to factors such as noise, outliers, or inherent variability in the data. In such cases, enforcing an exact separation of the training data in the input space may require a highly complex decision boundary, which can lead to overfitting and poor generalization performance. To address this issue, the support vector machine is extended to allow for some misclassification of training points. This relaxed formulation, known as the **soft margin SVM**, introduces a trade-off between maximizing the margin and minimizing classification errors, thereby improving the model's robustness and generalization capability.

To achieve this, the classification constraints given by Equation (6.4) are relaxed by introducing **slack variables** $\xi_n \geq 0$, which are optimization variables that measure the degree of misclassification for each data point. The new optimization problem is formulated as:

$$\arg \min_{\mathbf{w}, b, \xi_n} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \right), \quad (6.23)$$

where $C > 0$ is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors. The new constraints are given by:

$$\begin{cases} l_n y(\mathbf{x}_n) \geq 1 - \xi_n, & n = 1, \dots, N, \\ \xi_n \geq 0. \end{cases}$$

From these conditions, it follows that for points on or inside the correct margin boundary, $\xi_n = 0$; otherwise, $\xi_n = |l_n - y(\mathbf{x}_n)|$. Together, these constraints represent a slackening of the requirement imposed by Equation (6.4). They allow for some points to be misclassified or to lie within the margin in order to improve the model's robustness. To better understand how this works, let us examine the impact of these new constraints on the classification of the data points.

When $\xi_n = 0$, the first constraint reduces to the canonical representation, as given by Equation (6.4). Consequently,

$$\begin{cases} l_n = 1 & \Rightarrow y(\mathbf{x}_n) \geq 1, \\ l_n = -1 & \Rightarrow y(\mathbf{x}_n) \leq -1. \end{cases}$$

This implies that these data points will be correctly classified, lying either on or within the margin boundary. When $0 < \xi_n \leq 1$,

$$\begin{cases} l_n = 1 & \Rightarrow y(\mathbf{x}_n) \geq 0, \\ l_n = -1 & \Rightarrow y(\mathbf{x}_n) \leq 0. \end{cases}$$

These points will also be correctly classified but will lie within the margins. Finally, points with $\xi_n > 1$ will be misclassified, as the condition $|l_n - y(\mathbf{x}_n)| > 1$ can only be satisfied if the signs of l_n and $y(\mathbf{x}_n)$ are opposite. Points satisfying either $0 < \xi_n \leq 1$ (margin violations) or $\xi_n > 1$ (misclassifications) are referred to as *margin errors*, as they deviate from the ideal margin boundary defined by $\xi_n = 0$.

The trade-off between margin violations and model robustness can be intuitively grasped by visualizing the geometric relationship between the decision boundary and training samples satisfying each of these conditions. Figure 6.3 provides a schematic representation of these cases, illustrating the relationship between ξ_n and the classification outcomes for the case of 2-dimensional data. Points 1 and 2 represent cases in which $\xi_n = 0$; they are correctly classified and lie outside the margin and on the margin boundary, respectively. Point 3 corresponds to a case where $0 < \xi_n \leq 1$; it is correctly classified but violates the margin boundary. Finally, point 4 represents a case in which $\xi_n > 1$, resulting in a misclassification. Points 3 and 4 are both examples of margin errors. This visualization highlights the practical implications of the trade-off between margin violations and classification accuracy, providing insight into how soft-margin SVMs balance robustness and generalization.

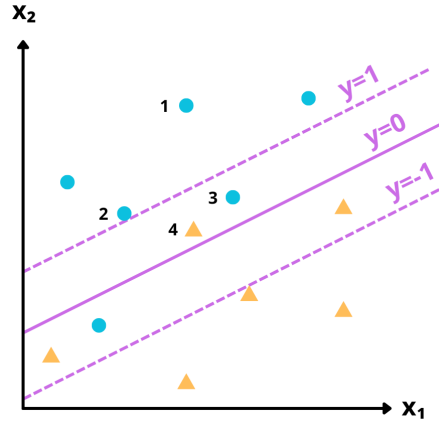


Figure 6.3: Schematic illustration of the relationship between ξ_n and classification outcomes for 2D data. Points 1 and 2 represent correctly classified points ($\xi_n = 0$) lying outside the margin and on the margin boundary, respectively. Point 3 shows a correctly classified point with $0 < \xi_n \leq 1$, violating the margin boundary. Point 4 represents a misclassified point where $\xi_n > 1$. Points 3 and 4 are examples of margin errors. This visualization highlights the practical implications of the trade-off between margin violations and classification accuracy.

Having established the soft-margin SVM formulation and its constraints, we now turn to solving the associated optimization problem. As in the hard-margin case, we employ the method of Lagrange multipliers, which allows us to incorporate the constraints into the objective function and derive the optimal solution. To extend the principles used for the hard-margin SVM to the more general soft-margin case, we introduce two sets of Lagrange multipliers, $a_n \geq 0$ and $\mu_n \geq 0$. The Lagrangian for this optimization problem is given by

$$L(\mathbf{w}, b, a_n, \xi_n, \mu_n) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \{l_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^N \mu_n \xi_n. \quad (6.24)$$

As before, the Lagrangian multipliers must be non-negative and the dual problem must satisfy the requirements of the original optimization problem, $l_n y(\mathbf{x}_n) \geq 1 - \xi_n$ and $\xi_n \geq 0$. Finally, we impose complementary slackness, which implies $a_n(l_n y(\mathbf{x}_n) - 1 + \xi_n) = 0$ and $\mu_n \xi_n = 0$. The combined set of conditions that must

be satisfied by the soft-margin SVM optimization problem is given by:

$$\begin{cases} a_n \geq 0, \\ l_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0, \\ a_n(l_n y(\mathbf{x}_n) - 1 + \xi_n) = 0, \\ \mu_n \geq 0, \\ \mu_n \xi_n = 0, \\ \xi_n \geq 0, \end{cases}$$

with $n = 1, \dots, N$.

The optimization with respect to \mathbf{w} and b is performed in the same manner as in the hard-margin case, and recovers the conditions given by Equations (6.12) and (6.13). Since the expressions for a_n and \mathbf{w} are the same as in the hard-margin case, the substitutions can be performed the same way (see Subsection 6.2.1) and the Lagrangian can be rewritten as

$$L(\mathbf{w}, a_n, \xi_n, \mu_n) = -\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{n=1}^N a_n + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \xi_n - \sum_{n=1}^N \mu_n \xi_n, \quad (6.25)$$

which simplifies to

$$L(\mathbf{w}, a_n, \xi_n, \mu_n) = -\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{n=1}^N a_n + \sum_{n=1}^N \xi_n (C - \mu_n - a_n). \quad (6.26)$$

Setting $\frac{\partial L}{\partial \xi_n} = 0$ yields:

$$a_n = C - \mu_n, \quad (6.27)$$

implying that $C - \mu_n - a_n = 0$. Substituting this condition into the Lagrangian, we recover the dual Lagrangian of the hard-margin case:

$$\tilde{L}(a_n) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m l_n l_m \mathbf{x}_n^T \mathbf{x}_m. \quad (6.28)$$

However, this is not the same optimization problem, as the constraints on the dual Lagrangian are different. Since a_n , C and μ_n are all non-negative, it follows that a_n is bounded between 0 and C . Additionally, $\sum_{n=1}^N a_n l_n = 0$, ensures that the solution respects the linear constraint derived from the primal problem, as shown in the previous section. The resulting set of constraints for the dual problem is

$$\begin{cases} 0 \leq a_n \leq C, \\ \sum_{n=1}^N a_n l_n = 0, \end{cases}$$

for $n = 1, \dots, N$.

As before, substituting Equation (6.12) into discriminant function leads to:

$$y(x) = \text{sign} \left(\sum_{n=1}^N a_n l_n \mathbf{x}_n^T \mathbf{x} + b \right). \quad (6.29)$$

Hence, the soft-margin SVM maintains the sparsity of the solution, as only training samples with $a_n > 0$ contribute to the model's predictions. These samples are the support vectors, which include points lying on the margin boundary ($\xi_n = 0$), within the margin ($0 < \xi_n \leq 1$) and misclassified points ($\xi_n > 1$).

As in the hard-margin SVM case, there is a relationship between the values of a_n , ξ_n and the geometric placement of the training samples relative to the margin. To understand this, we first compile all the conditions derived by solving the optimization problem

$$\begin{cases} 0 \leq a_n \leq C, \\ l_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0, \\ a_n(l_n y(\mathbf{x}_n) - 1 + \xi_n) = 0, \\ \mu_n \geq 0, \\ \mu_n \xi_n = 0, \\ \xi_n \geq 0, \\ a_n = C - \mu_n. \end{cases}$$

When both a_n and ξ_n equal zero, the training sample is correctly classified and lies outside the margin. The third condition, together with the requirement that a_n be non-negative implies that the training samples for which $a_n > 0$ satisfy

$$l_n y(\mathbf{x}_n) = 1 - \xi_n. \quad (6.30)$$

This is also known as the *support vector condition*. When $a_n < C$, the last constraint implies that μ_n is strictly positive. According to the fifth condition, $\mu_n \xi_n = 0$, meaning $\xi_n = 0$. Hence, points for which $\xi_n = 0$ and $0 < a_n < C$ will lie exactly on the margin. Points with $a_n = C$ are either correctly classified, but violate the margin (if $0 < \xi_n \leq 1$) or are misclassified (if $\xi_n > 1$).

The relationship between a_n , ξ_n and the geometric placement of the training samples relative to the margin is especially convenient because, once the values of a_n are calculated by solving the dual optimization problem, it is possible to determine which of the support vectors lie exactly on the margin. These support vectors, which satisfy $0 < a_n < C$, can then be used to derive a numerically stable solution for b . Substituting Equation (6.12) into the support vector condition yields

$$l_n \left(\sum_{m \in S} a_m l_m \mathbf{x}_n^T \mathbf{x}_m + b \right) = 1, \quad (6.31)$$

where S denotes the set of indices corresponding to the support vectors. The final expression for b is given by

$$b = \frac{1}{N_M} \sum_{n \in M} \left(l_n - \sum_{m \in S} a_m l_m \mathbf{x}_n^T \mathbf{x}_m \right), \quad (6.32)$$

with M denoting the set of indices of data points having $0 < a_n < C$.

Note that, in both the hard-margin and soft-margin SVM formulations, neither the optimization problem nor the final expression for the discriminant function depend explicitly on the coordinates or the

training samples themselves. Instead, these quantities depend solely on scalar products between coordinates. This property will prove crucial later when we generalize the SVM formulation to handle nonlinear data.

6.2.2.1 The ν -SVM formulation

Having discussed the original soft-margin SVM, we now turn our attention to the specific algorithm employed in this work: the ν -SVM formulation, proposed by Schölkopf et. al [83]. This formulation is a reparameterization that substitutes the parameter C with ν , whose relationship to the number of margin errors and support vectors is more interpretable, making it more convenient for users to apply and tune.

In the linear case, the ν -SVM formulation involves maximizing the Lagrangian [78]

$$\tilde{L}(a_n) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m l_n l_m \mathbf{x}_n^T \mathbf{x}_m, \quad (6.33)$$

subject to the constraints

$$\begin{cases} 0 \geq a_n \geq \frac{1}{N}, \\ \sum_{n=1}^N a_n l_n = 0, \\ \sum_{n=1}^N a_n \geq \nu. \end{cases}$$

These conditions enable the parameter ν to be interpreted as both an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. As was discussed in Subsection (6.2.2), support vectors include all the points for which $a_n > 0$, while margin errors include only those for which a_n is at its maximum value. This means that all margin errors are support vectors, although not all support vectors are margin errors.

Let N_{SV} denote the total number of support vectors, and N_{ME} , the number of margin errors. It follows that $N_{SV} \geq N_{ME}$. In the extreme case in which all support vectors are margin errors, each term a_n will either be 0 or $1/N$. In this case, the sum $\sum_{n=1}^N a_n$ reaches its maximum value:

$$\max \sum_{n=1}^N a_n = \frac{N_{SV}}{N}. \quad (6.34)$$

Recalling that

$$\sum_{n=1}^N a_n \geq \nu, \quad (6.35)$$

it follows that

$$\frac{N_{SV}}{N} \geq \nu. \quad (6.36)$$

Thus, ν serves as a lower bound for the fraction of support vectors.

Now, consider the case where not all support vectors are margin errors. In this situation, the sum can be rewritten as

$$\sum_{n=1}^N a_n = \frac{N_{ME}}{N} + \Delta, \quad (6.37)$$

where Δ accounts for the contribution from support vectors that are not margin errors. Rearranging,

$$\frac{N_{ME}}{N} = \sum_{n=1}^N a_n - \Delta. \quad (6.38)$$

To find an upper bound for the fraction of margin errors, we consider the case in which the sum $\sum_{n=1}^N a_n$ is at its minimum. From Equation (6.35), we have:

$$\frac{N_{ME}}{N} = \nu - \Delta. \quad (6.39)$$

Since $\Delta \geq 0$, it follows that

$$\frac{N_{ME}}{N} \leq \nu. \quad (6.40)$$

Thus, ν serves as an upper bound for the fraction of margin errors.

This property of the ν -SVM formulation is particularly interesting for the purposes of our study, as it provides a straightforward interpretation for one of the model's key parameters. By fixing ν , we directly establish an upper bound on the fraction of data points that can be margin violations or misclassifications. For instance, setting $\nu = 0.2$ ensures that at most 20% of the data points violate the margin or are misclassified.

Controlling this percentage is crucial because it directly impacts the accuracy of the classification: allowing too many misclassifications would degrade the performance of the classifier, while too few might lead to *overfitting*—a scenario where the model performs well on the training data but fails to generalize to unseen data—or an overly rigid model. This interpretation offers a clear criterion for selecting ν , which is especially valuable in a study that must account for numerous variables. Moreover, the ability to control the fraction of margin violations aligns with our goal of achieving a balance between model robustness and classification accuracy.

6.2.3 The kernel trick

So far, we have focused on the case of linearly separable data, where the SVM formulation provides a theoretically sound and interpretable framework for classification. However, real-world applications often involve data that is not linearly separable, exhibiting complex, nonlinear patterns. A common strategy in machine learning to address this challenge is to transform the data by mapping it onto a higher-dimensional space, where it may become linearly separable. This new space is called the feature space, and the transformation introduces new feature representations that better capture the data's structure.

To illustrate this idea, consider a simple example from classical mechanics [79]: Newton's law of gravitation, which describes the gravitational force f between two masses m_1 and m_2 separated by a distance r . The relationship is given by:

$$f(m_1, m_2, r) = C \frac{m_1 m_2}{r^2}, \quad (6.41)$$

where C is a constant. In its current form, this relationship is nonlinear, and a linear model would struggle to learn it directly. However, by applying a logarithmic transformation to the variables:

$$(m_1, m_2, r) \rightarrow (x, y, z) = (\ln m_1, \ln m_2, \ln r), \quad (6.42)$$

we can rewrite the relationship as:

$$g(x, y, z) = \ln f(m_1, m_2, r) = \ln C + \ln m_1 + \ln m_2 - 2 \ln r, \quad (6.43)$$

This transformed equation is linear in the new variables x , y , and z , meaning a linear model could now learn the relationship.

In this example, the transformation does not increase the dimensionality of the data; instead, it maps the original input space X onto a different feature space F of the same dimensionality. However, in many real-world situations, it is not possible to find such a mapping that allows the relationship between the features to be expressed in a linear form without increasing the dimensionality of the data.

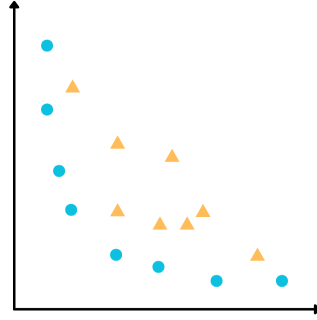


Figure 6.4: A training set composed of 2D data that is not linearly separable in the original input space. No straight line can classify the data without significant misclassification. Adapted from Reference [87].

Consider, for example, the scenario illustrated in Figure 6.4 [87], where we have a training set of 2-dimensional data that is not linearly separable in the original 2D space. In this case, no straight line can be drawn to separate the two classes without misclassifying a significant portion of the data. As a result, the SVM cannot effectively classify the data in its current form. However, we can define a transformation that maps this 2-dimensional data onto a 3-dimensional feature space, where the data becomes linearly separable by a plane. This transformation is achieved by applying the function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, defined as:

$$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2). \quad (6.44)$$

As shown in Figure 6.5, this transformation captures the nonlinear relationships in the original data, enabling the transformed data to be separated by a linear decision boundary in the 3D feature space. This illustrates the power of mapping data into higher-dimensional spaces to achieve linear separability, a key idea behind the extension of SVMs to nonlinear classification tasks.

Such explicit mappings, however, become impractical as the dimensionality of the data and the complexity of the transformations increase. Computing and storing high-dimensional feature vectors can be computationally prohibitive, especially for large datasets or highly complex feature spaces. The key innovation of the SVM formulation lies in its ability to circumvent this challenge. As discussed earlier, neither

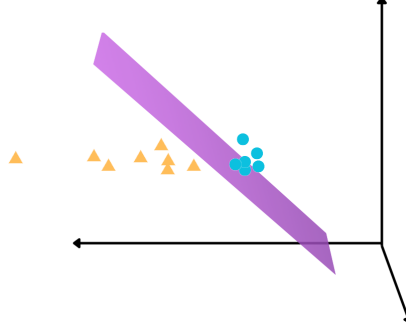


Figure 6.5: Visualization of the transformed data in a 3D feature space, where the data becomes linearly separable by a plane. Adapted from Reference [87].

the SVM optimization problem nor the final expression for the discriminant function depend explicitly on the coordinates or the training samples themselves. Instead, these quantities depend solely on scalar products between coordinates. This observation makes it possible to implicitly map the data onto a higher-dimensional feature space through the use of **kernel functions**, avoiding the need to explicitly compute the transformed feature vectors.

Consider a nonlinear mapping $\phi : X \rightarrow F$, chosen so that the transformed data is linearly separable in the feature space F . The new discriminant function is given by:

$$y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) + b) . \quad (6.45)$$

Using this, the dual Lagrangian can be written as:

$$\tilde{L}(a_n) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m l_n l_m \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) . \quad (6.46)$$

Instead of explicitly constructing $\phi(\mathbf{x}_n)$ and $\phi(\mathbf{x}_m)$ for all values of n and m , we can define a kernel function:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) , \quad (6.47)$$

which allows the inner product to be computed implicitly. This means that rather than explicitly defining a feature space, computing the inner product in that space, and then finding a way to compute this value directly from the original inputs, we can define a kernel function directly. This bypasses the need for explicit feature mapping, significantly reducing computational costs.

To ensure that a function is a valid kernel, we need to establish the properties that guarantee it corresponds to a valid inner product in some feature space. Let X be any finite input space. According to Mercer's theorem [79], a symmetric function $k : X \times X \rightarrow \mathbb{R}$ is a kernel function if, for all $n \geq 1$, $x_1, x_2, \dots, x_n \in X$ and $c_1, c_2, \dots, c_n \in \mathbb{R}$:

$$\sum_{i,j=1}^n c_i c_j k(x_i, x_j) \geq 0 . \quad (6.48)$$

This condition ensures that the kernel function corresponds to a valid inner product in some feature space. Equivalently, the matrix K with entries $k_{ij} = k(x_i, x_j)$ must be positive semi-definite (i.e., it has non-negative eigenvalues).

The most commonly used kernel function is the Gaussian kernel, defined as:

$$k(\mathbf{x}_n, \mathbf{x}_m) = e^{-\gamma \|\mathbf{x}_n - \mathbf{x}_m\|^2}, \quad (6.49)$$

where γ is a free parameter. This is also called the Radial Basis Function (RBF) kernel. Conceptually, kernels are often presented as a measure of similarity between points [88]. The Gaussian kernel provides an example where this notion can be intuitively grasped. The function has its maximum value, 1, when the Euclidian distance $\|\mathbf{x}_n - \mathbf{x}_m\|$ between \mathbf{x}_n and \mathbf{x}_m is zero, and it decreases exponentially as this distance increases.

This interpretation allows us to gain intuition on the role of the parameter γ . For large values of γ , $k(\mathbf{x}_n, \mathbf{x}_m)$ will decrease rapidly as the distance between \mathbf{x}_n and \mathbf{x}_m increases. Conversely, for small values of γ , the decrease will be slower. Intuitively, γ can be interpreted as defining how far the influence of a single training example reaches, with low values meaning “far” and high values meaning “close”. The γ parameter can be seen as the inverse of the radius of influence of samples selected by the model as support vectors [89].

Choosing an appropriate value for γ is crucial for the performance of the SVM. If γ is too large, the model may overfit to the training data, capturing noise and leading to poor generalization. On the other hand, if γ is too small, the model may fail to capture important patterns in the data.

In conclusion, kernel functions play a fundamental role in enabling SVMs to handle complex, nonlinear data. By implicitly mapping input data into high-dimensional feature spaces, they allow the SVM to efficiently construct nonlinear decision boundaries, as illustrated in Figure 6.6. This understanding of kernels is particularly crucial for comprehending one-class SVMs, which extend the SVM methodology to unsupervised learning tasks. In one-class SVMs, kernels are essential for defining a boundary around the data, enabling the model to identify outliers or novel patterns. This mechanism will be explored in detail in the following section.

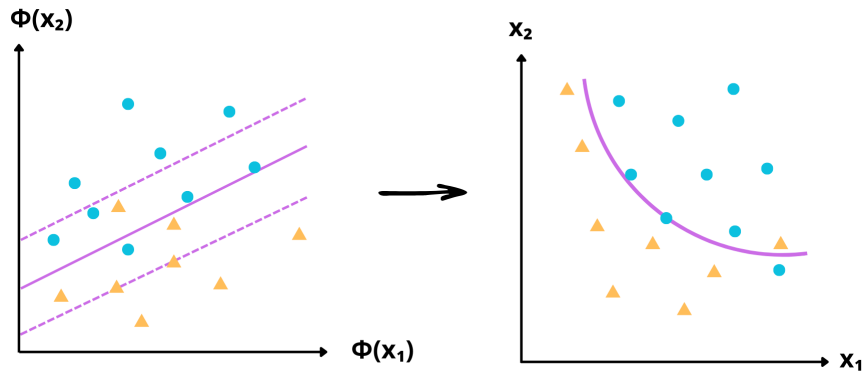


Figure 6.6: Schematic representation of how an SVM can be used to classify nonlinear data. By employing the kernel trick, the data is implicitly mapped into a higher-dimensional feature space, where it becomes linearly separable and the SVM identifies a separating hyperplane (left). When transformed back into the original input space, this hyperplane corresponds to a nonlinear decision boundary (right).

6.2.4 One-class SVMs

Imagine a door equipped with a facial recognition system designed to unlock only for a specific authorized user. For this system to work effectively, it must learn to recognize the unique features of the authorized user’s face and distinguish it from all other possible faces, including those it has never encountered before. When presented with a new image captured by the device’s camera, the system should then determine whether the face belongs to the authorized user or not. This is a classic example of a one-class classification problem. In such problems, the goal is to learn a model that identifies objects belonging to a specific target class (in this case, the authorized user’s face) while rejecting all other objects (all other faces) as outliers or anomalies.

Unlike traditional classification tasks, where the model is trained on examples from multiple classes, one-class classification focuses solely on defining the boundaries of a single class, making it particularly useful for tasks like anomaly detection [90], fraud prevention [91], and novelty identification [82,92].

In Reference [81], Schölkopf et al. proposed a method for solving one-class classification problems using the ν -SVM algorithm. In binary classification, the SVM is trained on data from two distinct classes, with each training sample labeled as belonging to one class or the other. The algorithm’s goal is to find a hyperplane that separates the two classes with the maximum possible margin. In contrast, one-class classification involves training the SVM on data from only one class—the target class. The goal here is not to separate two classes but to define a boundary around the target class that distinguishes it from all other possible data points. To achieve this, the one-class SVM constructs a hyperplane in the feature space that separates the training data from a fixed reference point (defined as the origin) with the maximum possible margin. The viability of this approach relies on two factors: the mathematical framework provided by the soft-margin SVM optimization problem and the ability to create nonlinear decision boundaries using the kernel trick.

First, let us discuss the importance of the soft-margin SVM framework. In a hard-margin SVM, where no violations are allowed, attempting one-class classification by maximizing the margin between the training data and the origin would not make sense. Because the origin is arbitrarily defined, the hyperplane could be placed infinitely far away from both the origin and the training data, resulting in a trivial solution where all points are classified as part of the target class. This would render the algorithm useless for practical applications. The soft-margin formulation addresses this issue by allowing a fraction of the training data to violate the margin, ensuring that the hyperplane is positioned at a finite distance from the origin and effectively defining a meaningful boundary around the target class.

To illustrate this, let us return to the facial recognition system example. Even if the system is trained on images of the authorized user’s face, some inputs—such as the user wearing a mask, bad lighting, or unusual facial expressions—may deviate significantly from the training data. These inputs can be considered outliers or anomalies. By allowing a small fraction of such points to violate the margin, the one-class SVM ensures that the hyperplane is positioned in a way that balances the goal of maximizing the margin with the need to minimize the number of outliers. Given this, the interpretation of ν as an upper bound for the fraction of margin errors becomes particularly important in this application. The parameter ν controls the trade-off between maximizing the margin and allowing for outliers. A smaller value of ν results in a narrower margin and fewer allowed outliers, while a larger value of ν permits a wider margin and more outliers. Intuitively, one might expect that a smaller ν would reduce false positives by enforcing a stricter boundary around the

training data. However, as ν approaches zero, the soft-margin SVM behaves increasingly like a hard-margin SVM, which, as discussed, can unexpectedly lead to more false positives. Therefore, ν must be chosen carefully to balance reliability (fewer false positives) with usability (fewer false negatives).

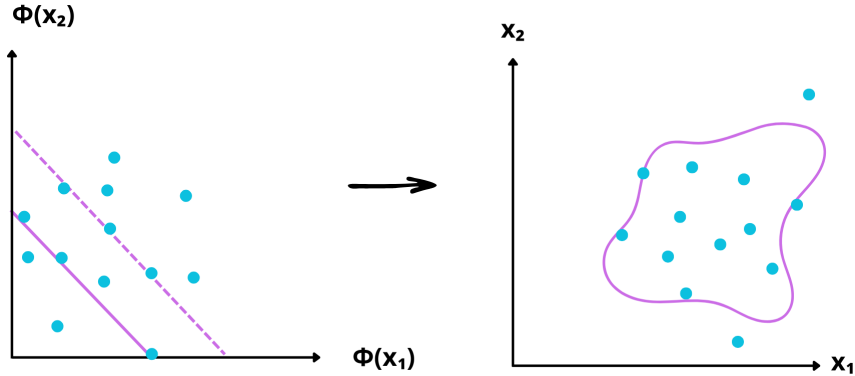


Figure 6.7: Schematic illustration of one-class SVM classification. Using the kernel trick, the data is implicitly mapped into a higher-dimensional feature space, where the SVM identifies a hyperplane separating the data from the origin (left). When projected back into the original input space, this hyperplane forms a closed decision boundary (right).

Another key factor contributing to the success of the one-class SVM is the kernel trick, which enables the construction of nonlinear decision boundaries. A linear SVM is ill-suited for one-class classification, as a hyperplane in the input space cannot effectively encapsulate the data distribution. However, by mapping the data into a higher-dimensional feature space and separating it from the origin, the one-class SVM can generate a closed decision boundary. This boundary effectively isolates the target class data from outliers or other types of data, making it a powerful tool for anomaly detection or novelty detection tasks. Together, the soft-margin framework and the kernel trick provide a robust mathematical foundation for one-class SVM, allowing it to adapt to complex data distributions and real-world applications. An illustration of the one-class SVM procedure is provided in Figure 6.7.

6.3 The modified benchmarking protocol

Having established the mathematical foundations of support vector machines, we now turn to their application in majorization-based benchmarking. In Chapter 4, we showed that the majorization-based indicator can distinguish between random quantum circuit outputs generated from universal and non-universal gate sets, even when the probability distribution is estimated from a finite number of measurements. While our initial analysis relied on visual inspection and heuristic criteria to judge when the fluctuation curves became distinguishable, the introduction of machine learning allows for a more systematic approach. To determine the amount of experimental resources necessary for distinguishing Haar-like and Clifford-like curves, we study the SVM’s classification error in relation to the number of gates, measurements, and experimental realizations used to generate the training data.

To enable a meaningful comparison between curves of different classes, the definitions of Haar-like and Clifford-like curves must be slightly adjusted. Previously, Haar-like curves were defined as the fluctuations in Lorenz curves of the frequency distributions of Haar-random vectors. In this study, however, Haar-like curves are derived from the Lorenz curves of output frequencies from random quantum circuits composed of universal gate sets. Similarly, Clifford-like curves, which were previously obtained from circuits composed

entirely of Clifford gates initialized from a random product state, are now generated from Clifford circuits initialized in $|0\rangle^{\otimes n}$, with n being the number of qubits. This ensures that the number of gates used to generate Haar-like and Clifford-like curves is directly comparable.

The Haar-like curves used in our study were constructed using the native gate set of IBM's quantum processors, $G_{\text{IBM}} = \{X, \sqrt{X}, \text{RZ}, \text{CNOT}\}$, where RZ rotations can be of an arbitrary angle. Since \sqrt{X} can generate Pauli X, we omit the latter in our simulations.

The present study focuses on circuits with $n = 5$ qubits, as a first step toward a more comprehensive analysis that considers different system sizes. Ideally, varying n would allow us to explore how the classification accuracy scales with the number of qubits and whether the majorization-based indicator remains effective in distinguishing between Haar-like and Clifford-like distributions in larger systems. However, due to time constraints, we have not yet conducted this extended analysis. Additionally, we are not considering connectivity constraints at this stage, which could influence the complexity of circuit implementations in real quantum hardware. The results presented here should therefore be regarded as preliminary, with the expectation that future work will assess the impact of system size more systematically.

Before beginning our analysis for the $n = 5$ qubit case, we verify whether Haar-like and Clifford-like curves remain distinguishable under these revised definitions. To do this, we generate the probability distributions for ensembles of 5000 circuits of each class, all composed of 500 gates. As demonstrated in our previous work [2], these values are sufficient for obtaining asymptotic curves in the infinite statistics regime, where probabilities are computed directly from the quantum state rather than estimated from measurements. These distributions serve as a reference for comparison with finite-sample estimates.

We observed that initializing the Clifford circuits in $|0\rangle^{\otimes n}$ instead of a random product state significantly increases the distinguishability between the asymptotic curves. While the difference d_0 between the peak fluctuations of the asymptotic Lorenz curves is of the order of $d_0 \sim 0.06$ when the Clifford circuits are initialized in a random product state, this distance increases to $d_0 \sim 0.2$ when the Clifford circuits are initialized in $|0\rangle^{\otimes n}$ instead. These results are illustrated in Figure 6.8.

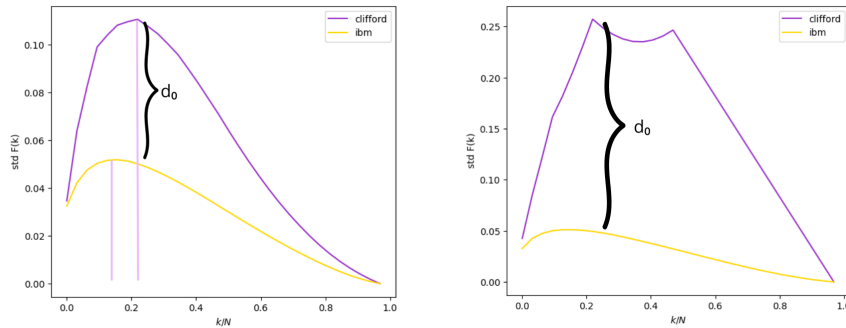


Figure 6.8: Comparison between the fluctuations in asymptotic Lorenz curves for RQCs constructed from Clifford and universal gate sets. The left panel shows results when Clifford circuits are initialized in a random product state, while the right panel corresponds to initialization in the $|0\rangle^{\otimes n}$ state. The difference d_0 between peak fluctuations increases significantly in the latter case.

The difference d between the peak fluctuations in Haar-like and Clifford-like Lorenz curves was computed across various experimental configurations, considering both the finite statistics case and the infinite statistics regime. For each experimental configuration, 50 independent sets of fluctuation curves were generated, and the average d value was computed over these sets. The number of gates varied from 30 to 200

in increments of 10, while the ensemble size—i.e., the number of circuits over which the standard deviation was computed—ranged from 20 to 1000 in increments of 20. In the finite statistics regime, the number of measurements per circuit ranged from 8 to 32.

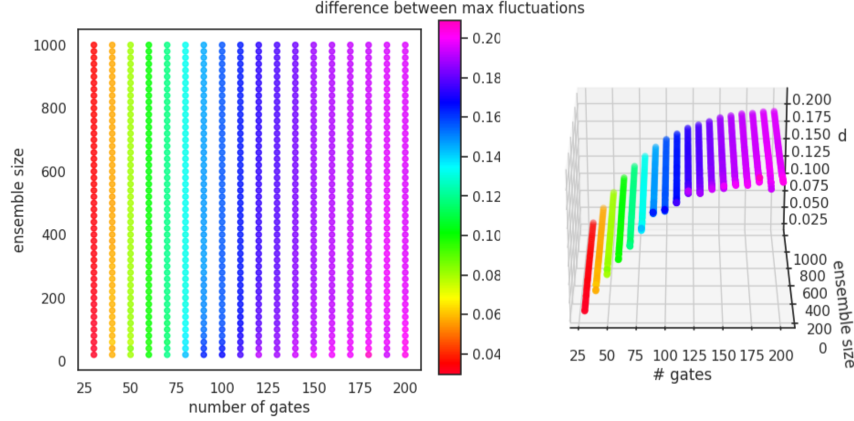


Figure 6.9: Difference between peak fluctuations of Clifford-like and Haar-like curves as a function of the ensemble size and the number of gates per circuit, in the infinite statistics regime. Both plots represent the same data but offer alternative visualizations.

Figure 6.9 presents the results in the infinite statistics regime, where only two variables are considered: the number of gates and the ensemble size. These results are visualized as a 2D color plot, with the x -axis representing the number of gates, the y -axis representing the ensemble size, and the value of d indicated by the color scale. Alternatively, the data can be represented in a 3D plot, where d is mapped to the z -axis instead.

The results show that in this regime, d reaches approximately 0.06—the asymptotic value for the case in which Clifford circuits are initialized in a random product state—for ensemble sizes of 20 circuits with 40 gates. These results suggest that significantly fewer resources are required to achieve distinguishability when Clifford circuits are initialized in the $|0\rangle^{\otimes n}$ state. Furthermore, the plots indicate that ensemble size has a significantly smaller impact on the distinguishability of the curves compared to the number of gates per circuit. This is evident from the fact that d , represented by the color scale in the 2D plot and by the z -axis value on the 3D plot, remains relatively stable as the ensemble size varies.

In the finite statistics regime, we consider three variables: the number of gates, the ensemble size, and the number of measurements per circuit. As a result, the data must be visualized in a 3D color plot, where the x -axis represents the number of gates, the y -axis represents the ensemble size, the z -axis represents the number of measurements per circuit, and the value of d is indicated by the color scale. The color distribution in the plot reveals that, among these resources, the ensemble size has the least impact on distinguishability. As before, the colors of the points, representing the distance d , remain relatively stable as the ensemble size varies. Instead, the variation in d is primarily influenced by changes in the number of gates and measurements. Although the color variation across these two variables may appear balanced, it is important to note that m varies in much smaller increments than the number of gates. Therefore, the number of measurements is the dominant factor in determining distinguishability.

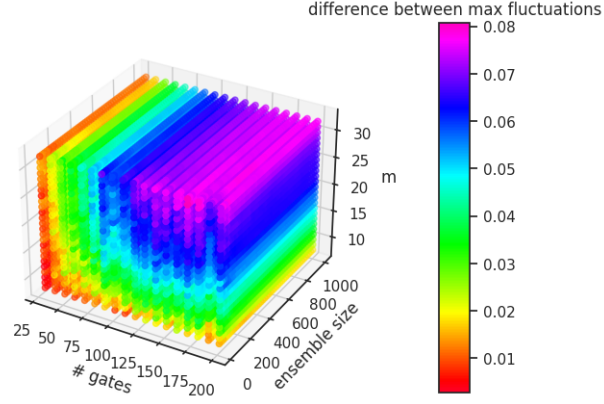


Figure 6.10: Difference between peak fluctuations of Clifford-like and Haar-like curves as a function of the ensemble size and the number of gates and measurements per circuit.

For ensemble sizes of 20 circuits with 40 gates and 8 measurements per circuit, d is already on the order of 0.01, which, as shown in the previous chapter, is distinguishable by visual inspection. Having established that Haar-like and Clifford-like Lorenz curves remain distinguishable under the new definitions, we now turn to the task of training an SVM to classify them. In the following subsections, we describe the training procedure, including the choice of parameters, the dataset construction, and the impact of different experimental configurations on classification accuracy.

6.3.1 Training set construction and parameter tuning

The training data used in this study consists of fluctuations in the Lorenz curves of RQC output frequencies. Each training sample \mathbf{x} is constructed by computing the standard deviation over c Lorenz curves, obtained from frequency distributions of circuits within the same class, each composed of g gates and estimated using m measurements. A schematic representation of how the training data is constructed can be seen in Figure 6.11.

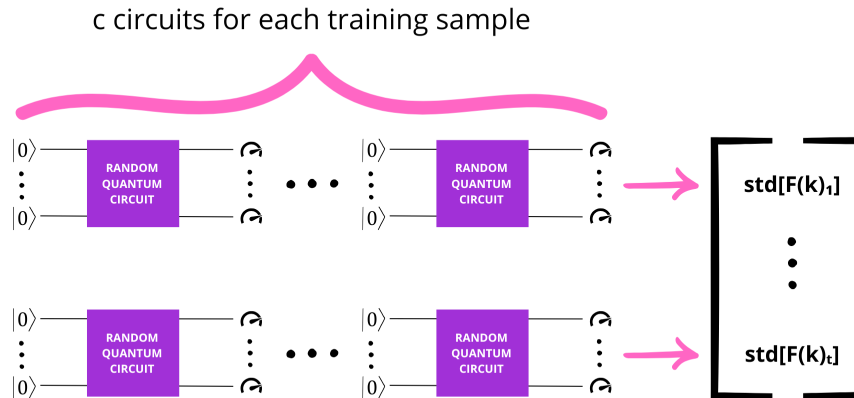


Figure 6.11: Schematic representation of the training data construction. Each of the t training vectors is obtained by computing the standard deviation over c Lorenz curves, derived from frequency distributions of random quantum circuit outputs.

The parameter ν of the ν -SVM model was fixed at 0.2. This imposes an upper bound of 20% on the fraction of margin errors allowed by the soft-margin SVM model. Since this study is still in an exploratory stage, we adopt a standard approach for support vector machine classification while leaving a more detailed analysis of kernel selection and parameter optimization for future work. Given that the Gaussian (RBF) kernel is a well-established choice for SVM classification in cases where the decision boundary is not necessarily linear [78, 79, 87], we adopt it as the default kernel for our model. This choice ensures flexibility in capturing complex relationships in the data while maintaining consistency with standard approaches in similar classification tasks.

To determine the γ parameter, we performed an initial tuning using a fixed configuration in the infinite statistics regime, where each standard deviation was computed over an ensemble of 10 circuits composed of 30 gates. This configuration was chosen because it represents the most challenging case for classification, as the Lorenz curves for Haar-like and Clifford-like distributions are closest together at lower gate depths. By tuning γ in a scenario where distinguishing between the two classes is hardest, we ensure that the chosen parameter remains effective in more favorable conditions as well. The value of γ varied from 0.1 to 0.5 in increments of 0.01. This range was chosen based on practical considerations. It was broad enough to cover a reasonable span of values typically used in similar SVM applications, ensuring that we didn't overlook a potentially optimal choice, while also keeping the search space manageable. Given that γ controls the influence of individual training samples in the SVM, starting from a small value (0.1) and increasing in small steps allowed for a systematic exploration without excessive computational cost.

This tuning was performed for both the binary and one-class SVM. The training set consisted of 800 samples, with 400 Haar-like and 400 Clifford-like samples for the binary SVM, and only Haar-like samples for the one-class SVM. The test set contained 200 samples, equally split between Haar-like and Clifford-like circuits. The classification error was measured as the percentage of misclassified test samples. The results can be visualized in Figure 6.12.

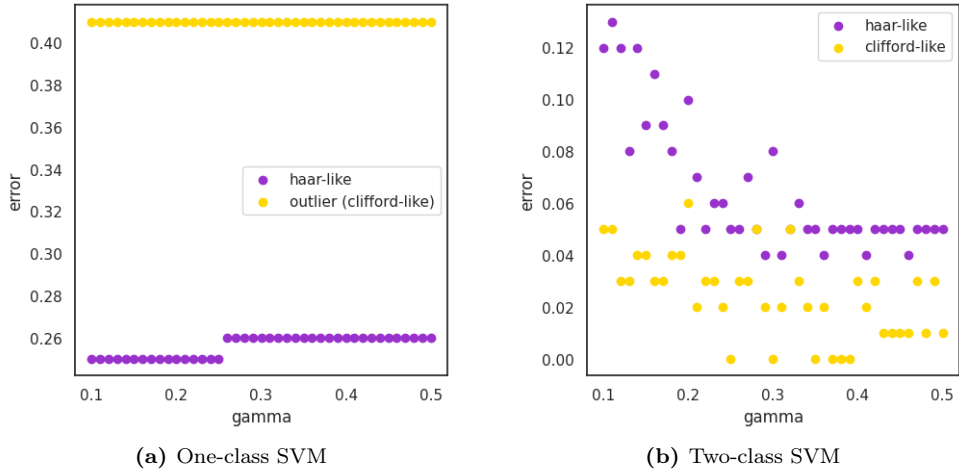


Figure 6.12: Results of γ tuning for both SVM classifiers, with ν fixed at 0.2.

For the binary SVM, the percentage of misclassified Haar-like samples starts at approximately 12% and drops as γ increases, eventually converging to approximately 5%. The percentage of misclassified Clifford-like samples remains below 6% for all values of γ in the considered range. For the one-class SVM, the

parameter γ exerts considerably less influence on the classification accuracy. The percentage of misclassified Haar-like samples (false negatives) remains at 25% for $0.1 \leq \gamma \leq 0.25$ and increases to 26% for $0.26 \leq \gamma \leq 0.5$. The percentage of misclassified Clifford-like samples (false positives) remains fixed at 41%.

To understand the results for the one-class SVM, recall that in an unsupervised learning setting, the SVM is trained exclusively on Haar-like data. As a result, the parameter ν plays a more significant role in defining the margin, as it directly controls the fraction of allowed outliers in the absence of explicitly labeled negative samples. Accordingly, the percentage of misclassified Haar-like samples is relatively close to ν , which sets the upper bound on the fraction of margin errors. To confirm this relationship, we repeated the tuning for the one-class SVM setting $\nu = 0.1$. As expected, the qualitative behavior remained unchanged, with γ exerting little influence on the classification errors. The percentage of misclassified Haar-like samples (false negatives) remained at 14% for $0.1 \leq \gamma \leq 0.36$ and increases to 15% for $0.36 \leq \gamma \leq 0.5$, aligning with the new value of ν . The results are shown in Figure 6.13.

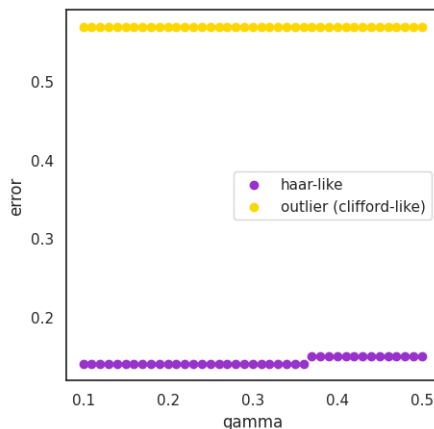


Figure 6.13: Results of γ tuning for the one-class SVM with ν fixed at 0.1. The classification accuracy depends significantly more on ν than on γ .

It is also notable that decreasing ν increased the percentage of misclassified Clifford samples (false positives) to 57%. At first glance, this may seem counterintuitive, as a smaller ν results in a narrower margin, allowing fewer outliers. However, as ν decreases, the soft-margin SVM increasingly resembles a hard-margin SVM. As discussed in subsection 6.2.4, the one-class SVM algorithm fails in the hard-margin limit because, without a sufficient number of outliers for reference, maximizing the margin pushes the decision boundary infinitely far from the origin. As a result, the model effectively classifies all samples as belonging to the target class, leading to a higher false positive rate. Therefore, when using the one-class SVM, it is crucial to choose a value of ν that balances the number of false negatives and false positives.

For the binary SVM, the margin width is primarily determined by the differences in the distributions of the two classes, making the influence of ν less crucial. As a result, the effect of γ becomes more noticeable. For the Gaussian kernel, γ can be interpreted as the inverse of the radius of influence of the support vectors selected by the model. Large values of γ can lead to overfitting and poor generalization, while small values may prevent the model from capturing important patterns in the data. Based on the results from both the one-class and binary SVM, we chose to fix γ at 0.3. This value improves the classification accuracy of Haar-like samples for the binary SVM while avoiding the risk of overfitting.

6.3.2 Training and classification accuracy

After fixing the model parameters, the next step is to determine the size of the training and test sets. In our study, we considered 2000 experimental configurations, varying the ensemble size from 10 to 100 (in increments of 10), the number of gates from 30 to 200 (also in increments of 10), and the number of measurements per circuit from 8 to 32. For each configuration, we generated 900 Haar-like samples and 500 Clifford-like samples. The training set contained 800 samples per configuration—400 Haar-like and 400 Clifford-like for the binary SVM, and only Haar-like samples for the one-class SVM. The test set consisted of 200 samples, equally divided between Haar-like and Clifford-like circuits.

To analyze classification accuracy across different experimental setups, we adopted two approaches. First, we investigated how accuracy varies with the total amount of experimental resources. To quantify this, we defined the **resource volume** V_r as:

$$V_r = c \cdot m \cdot g, \quad (6.50)$$

where c is the number of circuits in the ensemble over which the standard deviation of the Lorenz curves is computed, m is the number of measurements per circuit, and g is the number of gates in each circuit.

Second, we explored how accuracy depends on each of these resources individually while considering their combined effects. To do this, we used a 3D color plot to visualize classification accuracy as a function of c , m and g , allowing us to capture trends and interactions between these variables in a more comprehensive way.

6.3.2.1 Two-class classification

We begin by examining the results of the binary classification experiments, where the SVM was trained on both Haar-like and Clifford-like samples. The model was then tasked with distinguishing between the two classes, offering insights into the distinguishability of Haar-like and Clifford-like curves across various experimental setups. The classification error is defined as the total percentage of misclassified samples from both classes.

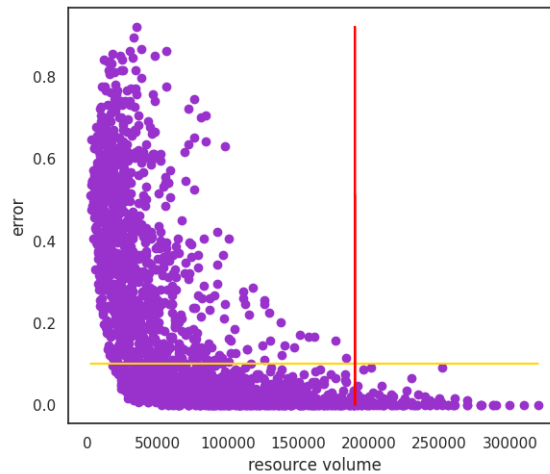


Figure 6.14: Scatter plot comparing classification error to resource volume. For small values of V_r , the error varies widely, but as V_r increases, the variation decreases, suggesting convergence toward 0%. The red line indicates the threshold of $V_r = 190,000$, and the yellow line marks the 10% error threshold. For $V_r \geq 190,000$, the error remains consistently within 10% or less.

A scatter plot comparing classification error to resource volume is shown in Figure 6.14. For small values of V_r , the error varies widely, ranging from 0% to as high as 92%. Each point in the scatter plot represents a different experimental configuration, meaning that among the setups with low V_r , some produce fluctuation curves that are significantly more distinguishable than others. This suggests that certain experimental resources have a much greater impact on the distinguishability of the curves than others.

As V_r increases, the variation in error decreases in a way that suggests convergence toward 0%. For $V_r \geq 190,000$, the error remains consistently within 10% or less. This implies that beyond a certain threshold of resources, the distinguishability between the curves no longer improves significantly. This result aligns with the expectation that the Haar-like and Clifford-like curves will eventually converge to the Haar-5 and Cliff-5 benchmark lines.

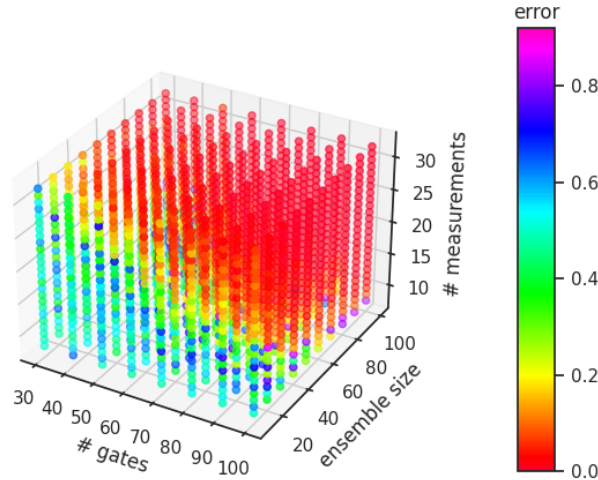


Figure 6.15: Classification error for the 2-class SVM, quantified by the percentage of misclassified samples, as a function of the ensemble size and the number of gates and measurements per circuit. Hues ranging from orange to red represent configurations for which the error is 20% or less.

To better understand how different resources impact distinguishability, we plot the classification error as a function of c , m , and g . As before, the classification error is defined as the percentage of misclassified samples. Figure 6.15 shows the 3D color plot representing the results, where the x -axis corresponds to the number of gates, the y -axis to the ensemble size, and the z -axis to the number of measurements per circuit, with classification error indicated by the color scale.

In the previous section, when analyzing the dependence of the difference d in peak fluctuations on c , m , and g , we observed that the number of measurements had a greater impact on distinguishability compared to the other resources. This effect is also reflected in the classification accuracy of the SVM. For smaller values of m , larger values of c and g are required to achieve an error of 20% or less. This can be observed in the distribution of points with hues ranging from orange to red. Conversely, when both c and g are large but m is small, the error may still be high. This behavior is especially pronounced for low values of m . For instance, the sample corresponding to $m = 8$, $c = 100$, and $g = 100$ is represented by a purple point, indicating an error of 80% or more. Meanwhile, the sample for $m = 9$, $c = 100$, and $g = 100$ is represented by an orange point, showing an error of 20% or less. This example highlights how even a slight increase in the number of measurements can significantly improve classification accuracy.

Given this, it is worth noting that for $n = 5$, the number of measurements required to distinguish Haar-like and Clifford-like behaviors is significantly smaller than the dimension of the probability vector. For comparison, the probability vector of a 5-qubit random state has 32 components. With only 9 measurements per circuit, a simple machine learning classifier is able to distinguish between Haar-like and Clifford-like fluctuation curves with an 80% success rate. This result is particularly encouraging, as it shows that the benchmarking procedure can achieve a high success rate with a small number of measurements relative to the size of the probability vector. Moreover, the accuracy improves quickly for low values of m and then stabilizes, which is beneficial for the experimental viability of the method, as it means fewer measurements are required to achieve a reliable classification.

To further investigate the relationship between classification error and experimental resources, we analyzed how the error varies when one parameter is fixed while the others change. This approach provides more information of the relative impact of each resource on classification accuracy. For each case, we provide two visualizations: a 2D color plot, where the classification error is indicated by the color scale, and a 3D plot, where it is mapped to the z -axis.

Figure 6.16a illustrates the relationship between classification error, ensemble size and number of gates per circuit when m is fixed at 16. The results indicate that, when m is fixed, both the ensemble size and the number of gates have a comparable influence on classification error. A decrease in the number of gates per circuit can be compensated by an increase of the same magnitude in the ensemble size, and vice-versa.

Similarly, Figure 6.16b depicts the relationship between classification error, ensemble size, and the number of measurements per circuit when g is fixed at 50. With g held constant, it is possible to see that the number of measurements has a larger influence on classification error than the ensemble size. This is evident from the steep drop in error as m increases, while changes in c alone result in more gradual improvements. Compensating for a decrease in the number of measurements by increasing the ensemble size is possible, but inefficient. On the other hand, it is possible to reduce the ensemble size significantly by increasing the number of measurements per circuit.

Finally, Figure 6.16c presents the classification error as a function of the number of gates and the number of measurements per circuit, with the ensemble size fixed at 50. Holding c constant allows us to directly compare the influence of g and m on classification accuracy. The influence of the number of gates is more comparable to that of the number of measurements than in the previous case. This means that compensating for a decrease in the number of measurements by increasing the number of gates is more efficient than by increasing the ensemble size. This result makes sense from a physical perspective: as the number of gates increases, the computation implemented in each circuit becomes more complex. This, in turn, has a more pronounced effect on the fluctuations in the Lorenz curves compared to simply increasing the ensemble size.

Still, the number of measurements remains the dominant factor influencing classification accuracy. This is evident because, although the error variation appears more balanced between g and m compared to the plot with fixed g , the range of values for m is still much smaller than that for g , meaning that smaller changes in m have a stronger impact on classification accuracy.

In conclusion, these results demonstrate that 5-qubit RQCs constructed using IBM's native gate set exhibit distinct behavior from those built solely from Clifford gates. In a noiseless scenario, this difference

can be identified with over 80% accuracy using a simple machine learning classifier and a moderate amount of resources. By analyzing the frequency distributions of real random quantum circuits and classifying fluctuations in Lorenz curves, this approach could assess whether the desired level of complexity is being successfully implemented by comparing classification accuracy for noiseless and experimental data.

A key advantage of this approach is its flexibility in adapting to different experimental constraints. Depending on the setup, increasing the number of measurements, implementing more gates per circuit, or expanding the ensemble size may be more practical. By examining classification error as a function of resource volume, this method enables researchers to tailor the benchmarking strategy to their specific experimental conditions, optimizing the configuration based on feasibility.

6.3.2.2 One-class classification

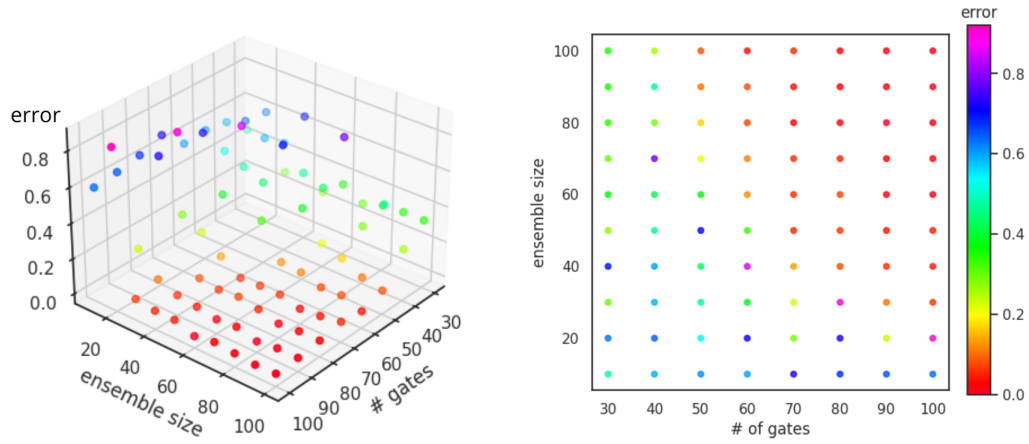
We now turn our attention to the one-class classification scenario, where the SVM is trained exclusively on Haar-like samples and tasked with identifying outliers corresponding to Clifford-like data. This approach differs from the binary classification case in that the model does not have prior exposure to both classes during training. Instead, it learns the characteristics of Haar-like curves and classifies deviations from this learned distribution as anomalies. By applying the same resource analysis as before, we examine how classification performance depends on the ensemble size, number of measurements and gates per circuit, identifying the conditions under which the model reliably distinguishes Clifford-like curves as outliers.

The classification error is again defined as the percentage of misclassified samples. In this case, however, the model does not explicitly recognize Clifford-like curves; it only determines whether a given sample belongs to the Haar-like class. Misclassifications include Haar-like curves that are incorrectly rejected as outliers (false negatives) and Clifford-like curves that are incorrectly classified as Haar-like (false positives).

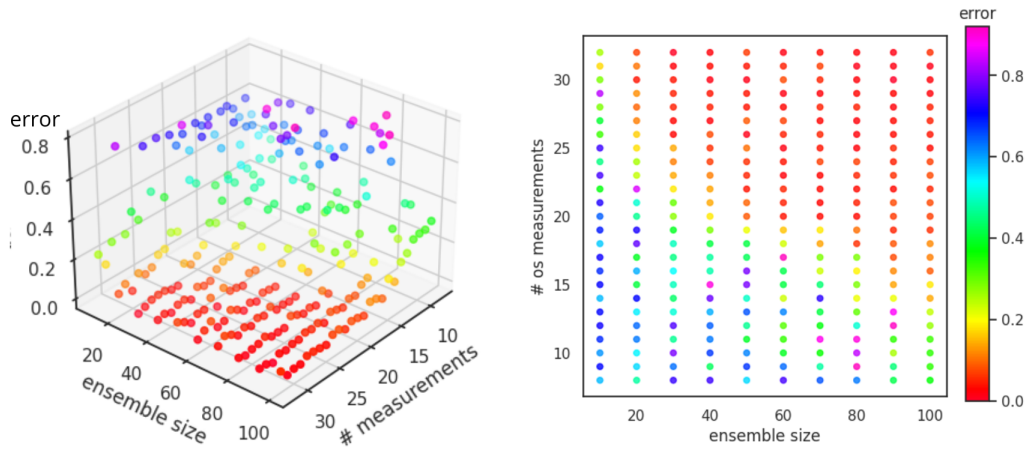
A scatter plot of the classification error as a function of V_r reveals a similar qualitative pattern to the binary SVM case: the error starts with a relatively high variation and gradually stabilizes at a constant value. However, two key differences emerge. First, the classification error in the one-class SVM is consistently lower than in the binary case, reaching a maximum of 56%—significantly below the 92% observed in the two-class SVM for small V_r . Second, unlike in the binary case, the error does not converge to zero.

One reason for this behavior is that, for the one-class SVM to function correctly, some Haar-like curves must be rejected as outliers, meaning false negatives will always be present. As a result, the error stabilizes just below 20%, aligning with the model’s built-in tolerance for margin errors. For $V_r \geq 230,000$, the error stabilizes and remains consistently at or below 20%.

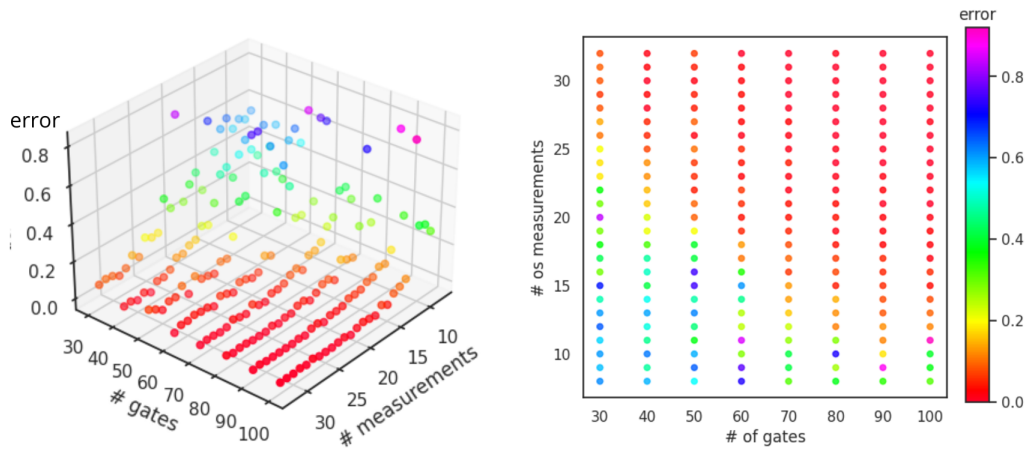
Analyzing the classification error separately for Haar-like and Clifford-like training samples reveals two distinct behaviors that further explain this trend. Figure 6.18 presents two scatter plots of classification error as a function of V_r : one considering only Haar-like samples in the test set and another considering only Clifford-like samples. In this context, the classification error corresponds to the percentage of false negatives and false positives, respectively. As V_r increases, the false negative rate remains between 8% and 37%, clustering around 20%, which aligns with the model’s allowed margin of error. In contrast, the false positive rate fluctuates significantly for small V_r , ranging from 0% to 86%, before steadily converging to 0% as V_r increases.



(a) number of measurements per circuit, $m = 16$



(b) number of gates per circuit, $g = 50$



(c) number of circuits in the ensemble, $c = 50$

Figure 6.16: Classification accuracy for the two-class SVM as a function of two parameters while the third is kept fixed. In each figure, both plots represent the same data but offer alternative visualizations.

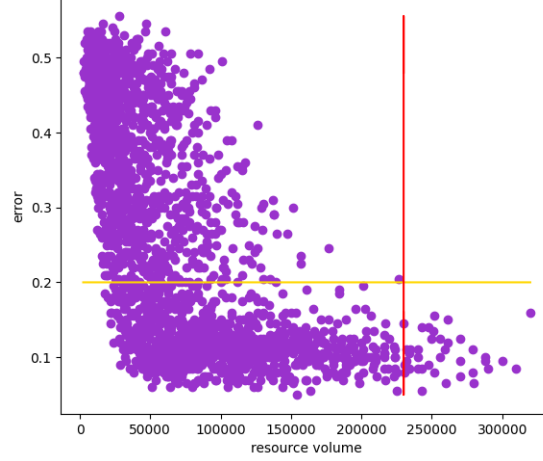


Figure 6.17: Scatter plot of classification error as a function of resource volume. The red line indicates the threshold of $V_r = 230,000$, and the yellow line marks the 20% error threshold. For $V_r \geq 230,000$, the error stabilizes and remains consistently at or below 20%.

Due to the unsupervised nature of the one-class SVM, the classification accuracy for Haar-like samples is less sensitive to the experimental setup compared to the two-class case. Since the model is trained only on Haar-like samples, its performance depends primarily on how well it captures the underlying probability distribution rather than on the specific experimental conditions that generate the data. In other words, the effectiveness of the classifier is more influenced by the optimization of the model itself than by variations in the experimental setup. This accounts for the fairly stable false negative rate.

In contrast, the classification accuracy for Clifford-like samples depends on the experimental configuration because, as more gates, measurements, or ensemble size are used, the underlying probability distribution that characterizes Clifford-like behavior becomes more distinct from that of Haar-like samples. As a result, the model is better able to differentiate between the two, leading to a steady decrease in the false positive rate.

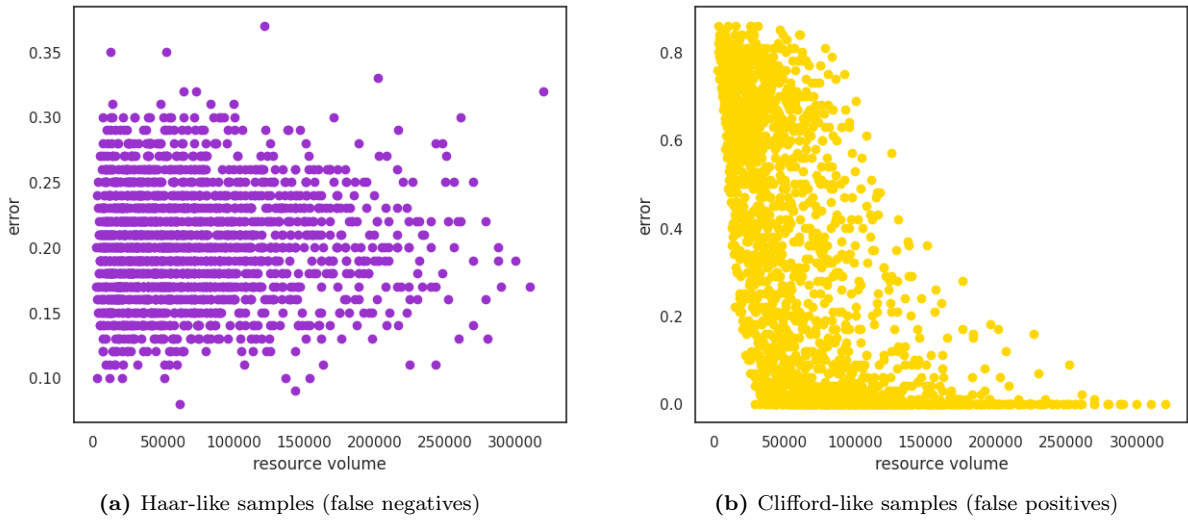


Figure 6.18: Scatter plots of classification error as a function of resource volume for Haar-like and Clifford-like samples, showing two distinct behaviors for false negatives and false positives.

Next, we analyze the classification accuracy as a function of c , m , and g . Figure 6.19 presents a 3D color plot of the results, where the x -axis represents the number of gates, the y -axis corresponds to the ensemble size, and the z -axis maps the number of measurements per circuit, with classification error indicated by the color scale. In this plot, points with hues ranging from yellow to red represent classification errors below 20%. These points are predominantly concentrated in the region with high values of both gates and measurements. By contrast, in the binary classification case, classification errors below 20% were observed for configurations with a high number of measurements per circuit, even in occasions where the number of gates was relatively low. As before, the number of measurements continues to have the most influence on classification accuracy.

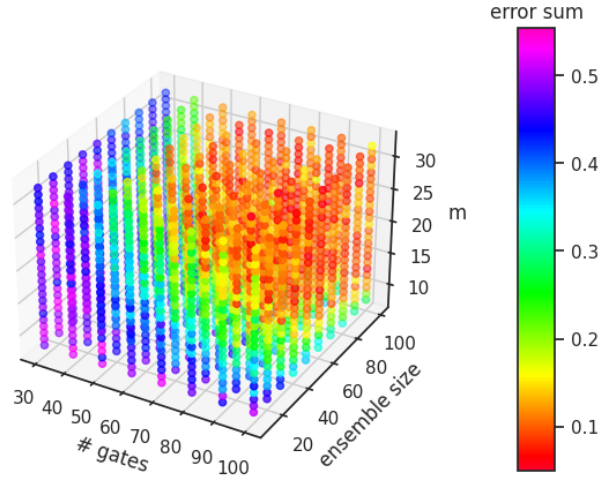


Figure 6.19: Classification error, quantified by the percentage of misclassified samples, for the one-class SVM as a function of the ensemble size and the number of gates and measurements per circuit. Hues ranging from yellow to red represent configurations for which the error is less than 20%.

Following the methodology used for the analysis of the binary SVM classifier, we studied how the error varies when one of the three parameters is fixed. Figure 6.20a shows the relationship between ensemble size, number of gates per circuit, and classification error when m is fixed at 16. The observed behavior is similar to that of the binary SVM case. When m is fixed, both the ensemble size and the number of gates exert a comparable influence on classification error.

Next, we fix the number of gates at 50. Figure 6.20b illustrates the relationship between ensemble size, number of measurements per circuit, and classification error at fixed g . Compared to the binary SVM case, where a steep drop in error was clearly observed as m increases, the decrease for the one-class SVM is more gradual. This means that, for the one-class SVM, compensating a decrease in measurements with an increase in gates to achieve greater classification accuracy is less inefficient than it would be when using a two-class SVM.

Finally, Figure 6.20c shows the classification error as a function of the number of gates and the number of measurements per circuit, with the ensemble size fixed at 50. As in the previous case, the number of measurements continues to have the most significant influence on classification accuracy.

Compared to the two-class SVM, the error rates obtained by the one-class SVM are higher. This is not unexpected, as one-class classification is inherently more stringent. In two-class classification, a sample only needs to exhibit properties more similar to Haar-like samples than to Clifford-like ones to be correctly

classified. In contrast, one-class classification requires the sample to be compared not only to Clifford-like samples but also to all other potential outliers. This makes the classification task more challenging. Additionally, we did not conduct a detailed study of the kernel choice, which is especially important for the one-class SVM. Since the one-class SVM’s performance is more sensitive to the kernel selection, the lack of kernel optimization likely contributed to its lower accuracy.

Given these factors, the results are quite encouraging. The one-class classifier managed to correctly classify 80% of the samples using a moderate number of resources, even without full optimization of the model. This demonstrates the existence of underlying patterns in the data that are learnable, even with a less-than-optimal machine learning setup.

These results confirm the findings given by the two-class SVM study. Random quantum circuits of 5 qubits constructed using IBM’s native gate set exhibit distinct behavior from those built solely from Clifford gates. In a noiseless scenario, this difference can be identified with close to 80% accuracy using a simple machine learning classifier and a moderate amount of resources.

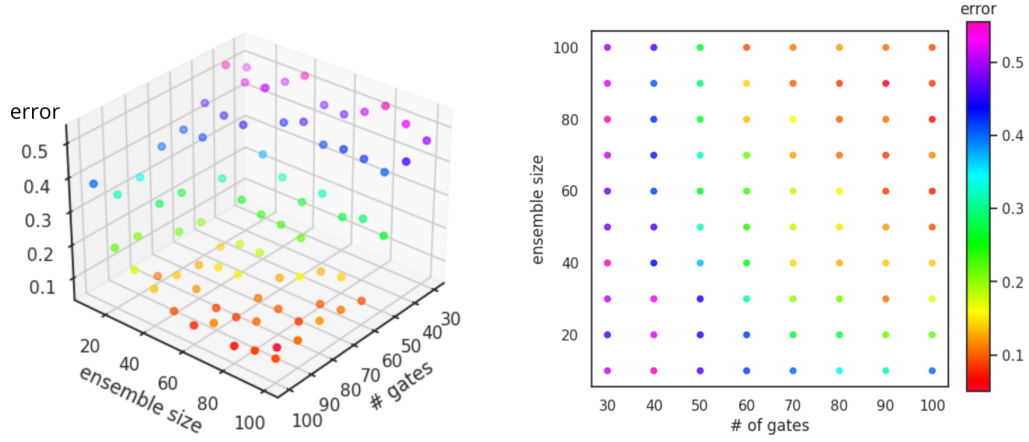
6.4 Conclusion

As discussed in the previous section, the ensemble size, number of gates, and number of measurements per circuit all play interconnected roles in determining the distinguishability between Haar-like and Clifford-like curves. This is reflected in the classification accuracy of both the one-class and two-class SVMs. Despite their different training approaches, both models lead to similar qualitative conclusions, further confirming that the complexity of a quantum state manifests in the fluctuations of Lorenz curves from measurement outputs.

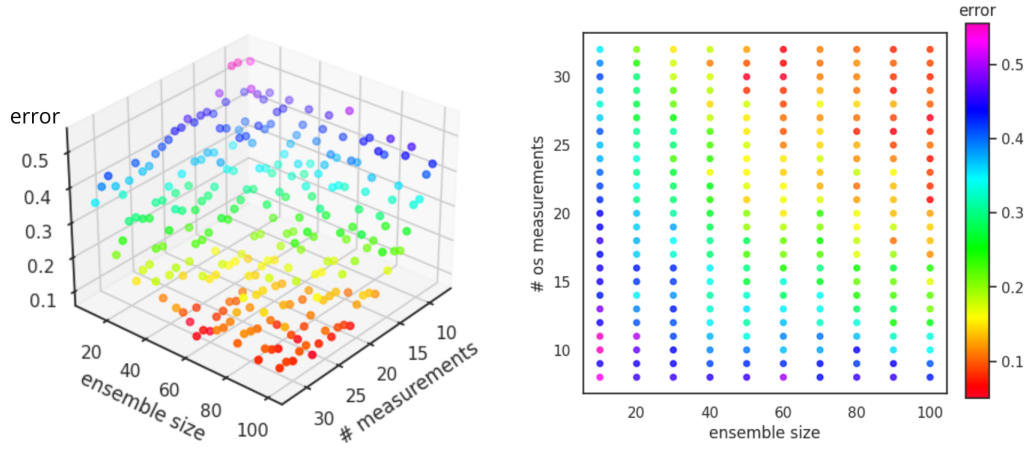
The main difference between the results of the binary and one-class SVM models lies in the relationship between ensemble size, number of measurements per circuit, and classification error when the number of gates is fixed. Compared to the binary SVM, the one-class SVM’s classification accuracy is more strongly influenced by the number of gates. This can be attributed to the one-class SVM’s unsupervised nature: it trains only on Haar-like curves, aiming to recognize patterns specific to this target class. As the number of gates increases, Haar-like curves approach their asymptotic behavior, making it easier for the model to learn their characteristics. In contrast, the two-class SVM simply needs to distinguish Haar-like curves from Clifford-like curves, which, as seen in Section 6.3, are separable well before the asymptotic regime is reached.

Having established how classification accuracy depends on resource allocation, we now turn to the practical applications of this benchmarking protocol and how it can be implemented in real quantum processors. As demonstrated in Reference [2], the majorization-based indicator is able to detect loss of complexity in the output due to noise. Therefore, it can be used to set target values for noise levels in current quantum devices. Using a machine-learning classifier, the protocol can be extended to benchmark devices even when the amount of measurements or gates implemented per circuit is not sufficient to realize the fluctuation curve’s full asymptotic behavior.

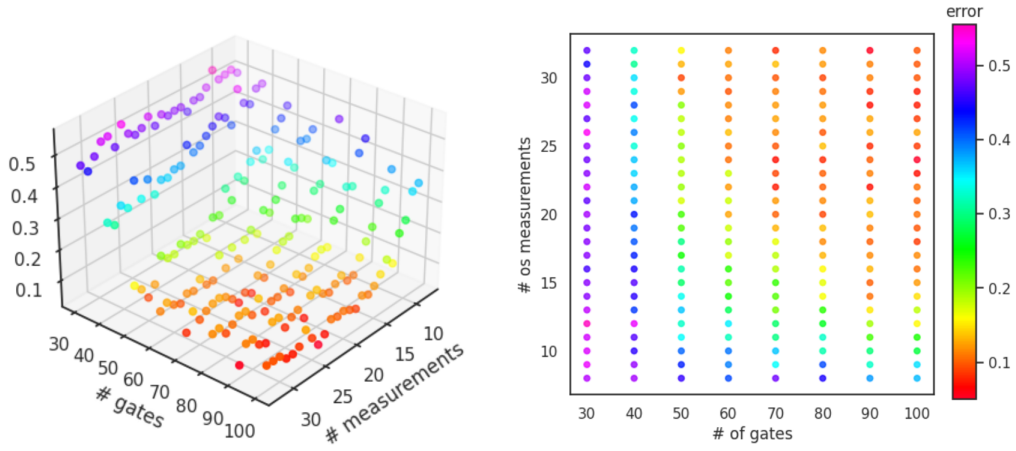
Using classical computation, it is possible to create a training set of noiseless fluctuations in Lorenz curves, taking into consideration the ensemble sizes and the number of gates and measurements that can be implemented per circuit. Then, a test set composed of fluctuations in Lorenz curves generated from the actual quantum processor is generated. If the machine learning classifier’s error rate is within the expected



(a) number of measurements, $m = 16$



(b) number of gates per circuit, $g = 50$



(c) number of circuits in the ensemble, $c = 50$

Figure 6.20: Classification accuracy for the one-class SVM as a function of two parameters while the third is kept fixed. In each figure, both plots represent the same data but offer alternative visualizations.

threshold for that particular resource volume, one may conclude that the processor achieves the desired level of complexity.

Naturally, this procedure is limited to system sizes where classical simulation of random quantum circuits remains feasible. However, many current quantum devices are still within this regime, making the approach practical for benchmarking today’s hardware. A similar limitation exists in cross-entropy benchmarking, which remains widely used and valuable despite its reliance on classical simulations.

Furthermore, using the one-class classification approach, it may be possible to extend the applicability of this benchmarking procedure to larger system sizes. Instead of training the model on Haar-like distributions, the training set would be constructed from RQCs known to be classically simulable. By generating standard deviation samples from Lorenz curves based on experimental data from real quantum circuits and testing them with this machine learning model, it may be possible to identify behavior that is not classically simulable.

Although preliminary, these insights provide strong motivation to continue this research. For $n = 5$, the number of measurements required to distinguish between Clifford-like and Haar-like behaviors is relatively low compared to the size of the probability vector. This is a promising result that encourages further investigation of larger system sizes to determine if the trend persists. Even though the machine learning model used in this analysis was simple and not fully optimized, it still yielded relatively good results, achieving classification accuracies of 80% or more with a moderate number of resources. This suggests that further optimization of the model could lead to even more accurate and experimentally viable benchmarking methods.

Chapter 7

Conclusion

The findings of this thesis establish majorization-based benchmarking as a promising tool for evaluating a quantum processor’s capacity for reliable, complex computation. The correlation between majorization relations and computational complexity in random quantum circuits was first introduced in Reference [54]. Initially, it was shown that the fluctuations in Lorenz curves of RQC outputs were sensitive to the gate set used in circuit generation. Building on this research, we sought to demonstrate that these fluctuations, which we refer to in this thesis as the majorization-based indicator, could serve as a benchmark for evaluating quantum processors. Throughout this thesis, we showed that the majorization-based indicator is not only sensitive to the gate set used to generate the RQC, but also responsive to a broader set of factors, including qubit connectivity, finite measurement statistics, and typical noise. These interrelated factors play a crucial role in determining a quantum processor’s computational capabilities. This provides strong evidence that the majorization-based indicator can be used to assess the global performance of a quantum processor, not just specific quantum circuits.

In Chapter 4, we demonstrate, using numerical results, that majorization-based benchmarking can determine the number of gates required for a noiseless processor to sample from its full Hilbert space, given the native gates and specific connectivity constraints of the hardware. We also simulate this specific architecture in the presence of typical noise, showing the majorization-based indicator’s capacity to detect a loss of complexity due to error. By determining the T_1 and T_2 values necessary to sample from the full Hilbert space, the majorization-based benchmarking protocol can be used to set thresholds for noise. The assessments provided by the majorization-based procedure align with those inferred by computing the average purity of the output states (before measurement) and the average fidelity between the noisy and noiseless output states (for the same circuit). Both of these quantities are far more experimentally costly to compute than the majorization-based indicator, underscoring its utility.

Finally, in Chapter 6, we show how majorization-based benchmarking can be implemented in the finite statistics regime. Our findings, although preliminary, provide strong motivation to continue this research, aiming towards experimental implementation. For $n = 5$, the number of measurements required to distinguish between Clifford-like and Haar-like behaviors is relatively low compared to the size of the probability vector. This is a favorable result that encourages further investigation of larger system sizes to determine if the trend persists.

Using a simple support vector machine with a Gaussian kernel, we achieved classification accuracies of 80% or higher in distinguishing Clifford-like and Haar-like behaviors while using a moderate number of resources. There are many opportunities for further optimization, particularly in the unsupervised approach, where a deeper understanding of the optimal kernel choice for this problem could significantly enhance performance.

To conclude, this thesis establishes majorization-based benchmarking as a powerful and experimentally viable tool for assessing the computational capabilities of quantum processors. By leveraging statistical properties of random quantum circuits, this method provides a scalable and efficient alternative to traditional benchmarking techniques, offering insights into the interplay between noise, circuit complexity, and hardware constraints. The promising results obtained here pave the way for further refinement and experimental validation, with the potential to establish majorization-based benchmarking as a standard for evaluating quantum devices. As quantum technologies continue to advance, investing in benchmarking methods that are both rigorous and practical will be crucial for guiding hardware development and unlocking the full potential of quantum computation.

Bibliography

- [1] Raúl O Vallejos, Pedro Silva Correia, Paola Concha Obando, Nina Machado O'Neill, Alexandre B Tacla, and Fernando de Melo. Quantum state inference from coarse-grained descriptions: Analysis and an application to quantum thermodynamics. *Physical Review A*, 106(1):012219, 2022.
- [2] Alexandre B Tacla, Nina M O'Neill, Gabriel G Carlo, Fernando de Melo, and Raúl O Vallejos. Majorization-based benchmark of the complexity of quantum processors. *Quantum Inf. Process.*, 23(6), June 2024.
- [3] Jeremy Hsu. CES 2018: Intel's 49-qubit chip shoots for quantum supremacy. <https://spectrum.ieee.org/intels-49qubit-chip-aims-for-quantum-supremacy>, 2018. [Accessed 05-12-2024].
- [4] IonQ. Compare quantum systems. <https://ionq.com/quantum-systems/compare>. [Accessed 05-12-2024].
- [5] OQC. OQC launches OQC Toshiko, the world's first enterprise ready quantum platform. <https://oqc.tech/company/newsroom/toshiko-the-worlds-first-enterprise-ready-quantum-platform/>, 2023. [Accessed 05-12-2024].
- [6] Matthew DeCross, Reza Haghshenas, Minzhao Liu, Enrico Rinaldi, Johnnie Gray, Yuri Alexeev, Charles H Baldwin, John P Bartolotta, Matthew Bohn, Eli Chertkov, Julia Cline, Jonhas Colina, Davide DelVento, Joan M Dreiling, Cameron Foltz, John P Gaebler, Thomas M Gatterman, Christopher N Gilbreth, Joshua Giles, Dan Gresh, Alex Hall, Aaron Hankin, Azure Hansen, Nathan Hewitt, Ian Hoffman, Craig Holliman, Ross B Hutson, Trent Jacobs, Jacob Johansen, Patricia J Lee, Elliot Lehman, Dominic Lucchetti, Danylo Lykov, Ivaylo S Madjarov, Brian Mathewson, Karl Mayer, Michael Mills, Pradeep Niroula, Juan M Pino, Conrad Roman, Michael Schechter, Peter E Siegfried, Bruce G Tiemann, Curtis Volin, James Walker, Ruslan Shaydulin, Marco Pistoia, Steven A Moses, David Hayes, Brian Neyenhuis, Russell P Stutz, and Michael Foss-Feig. The computational power of random quantum circuits in arbitrary geometries. 2024.
- [7] QuantWare. Meet Tenor | the 64-qubit QPU by QuantWare. <https://www.quantware.com/product/tenor>. [Accessed 05-12-2024].
- [8] Rigetti Computing Inc. Rigetti announces public availability of Ankaa-2 system with a 2.5x performance improvement compared to previous QPUs. <https://www.globenewswire.com/news-release/2024/>

- 01/04/2804006/0/en/Rigetti-Announces-Public-Availability-of-Ankaa-2-System-with-a-2-5x-Performance-Improvement-Compared-to-Previous-QPUs.html, 2024. [Accessed 05-12-2024].
- [9] RIKEN Fujitsu Limited. Fujitsu and RIKEN develop superconducting quantum computer at the RIKEN RQC-Fujitsu collaboration center, paving the way for platform for hybrid quantum computing. <https://www.fujitsu.com/global/about/resources/news/press-releases/2023/1005-01.html>, 2023. [Accessed 05-12-2024].
 - [10] Lars S Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob F F Bulmer, Filippo M Miatto, Leonhard Neuhaus, Lukas G Helt, Matthew J Collins, Adriana E Lita, Thomas Gerrits, Sae Woo Nam, Varun D Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, June 2022.
 - [11] Xinhua. China launches 504-qubit quantum chip, open to global users. <https://www.chinadaily.com.cn/a/202404/26/WS662b15dfa31082fc043c431e.html>, 2024. [Accessed 05-12-2024].
 - [12] Jay Gambetta. IBM Quantum System Two: the era of quantum utility is here. <https://www.ibm.com/quantum/blog/quantum-roadmap-2033>, 2023. [Accessed 05-12-2024].
 - [13] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*, December 2024.
 - [14] Google. Roadmap | Google Quantum AI. <https://quantumai.google/roadmap>. [Accessed 05-12-2024].
 - [15] Erin Angelini and Hugh Collins. IBM debuts next-generation quantum processor & IBM Quantum System Two, extends roadmap to advance era of quantum utility. <https://newsroom.ibm.com/2023-12-04-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility>, 2023. [Accessed 05-12-2024].
 - [16] Microsoft. Azure Quantum | Quantum Roadmap. <https://quantum.microsoft.com/en-us/vision/quantum-roadmap>. [Accessed 05-12-2024].
 - [17] Doug Finke. Quantinuum announces its processor development roadmap. <https://quantumcomputingreport.com/quantinuum-announces-its-processor-development-roadmap/>, 2024. [Accessed 05-12-2024].
 - [18] Junchao Wang, Guoping Guo, and Zheng Shan. SoK: Benchmarking the performance of a quantum computer. *Entropy (Basel)*, 24(10):1467, October 2022.
 - [19] Carina. State of Quantum Computing in Europe: AQT pushing performance with a Quantum Volume of 128 - AQT - Alpine Quantum Technologies — aqt.eu. <https://www.aqt.eu/aqt-pushing-performance-with-a-quantum-volume-of-128/>. [Accessed 22-01-2025].
 - [20] Petar Jurcevic, Ali Javadi-Abhari, Lev S Bishop, Isaac Lauer, Daniela F Bogorin, Markus Brink, Lauren Capelluto, Oktay Günlük, Toshinari Itoko, Naoki Kanazawa, Abhinav Kandala, George A Keefe, Kevin Krsulich, William Landers, Eric P Lewandowski, Douglas T McClure, Giacomo Nannicini, Adinath Narasgond, Hasan M Nayfeh, Emily Pritchett, Mary Beth Rothwell, Srikanth Srinivasan, Neereja

- Sundaresan, Cindy Wang, Ken X Wei, Christopher J Wood, Jeng-Bang Yau, Eric J Zhang, Oliver E Dial, Jerry M Chow, and Jay M Gambetta. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Sci. Technol.*, 6(2):025020, April 2021.
- [21] IQM. IQM Quantum Reports Benchmarks on 20-Qubit System - High-Performance Computing News Analysis | insideHPC — insidehpc.com. <https://insidehpc.com/2024/02/iqm-quantum-reports-benchmarks-on-20-qubit-system/>. [Accessed 22-01-2025].
- [22] savoryjo q. GitHub - CQCL/quantinuum-hardware-quantum-volume: Repository for sharing Quantinuum’s Quantum Volume data — github.com. <https://github.com/CQCL/quantinuum-hardware-quantum-volume>. [Accessed 22-01-2025].
- [23] Simon Martiel, Thomas Ayrat, and Cyril Allouche. Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way. *IEEE Transactions on Quantum Engineering*, 2:1–11, 2021.
- [24] Koen Mesman, Zaid Al-Ars, and Matthias Möller. Qpack: Quantum approximate optimization algorithms as universal benchmark for quantum computers. *arXiv preprint arXiv:2103.17193*, 2021.
- [25] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum information*, 5(1):45, 2019.
- [26] Alexander J McCaskey, Zachary P Parks, Jacek Jakowski, Shirley V Moore, Titus D Morris, Travis S Humble, and Raphael C Pooser. Quantum chemistry as a benchmark for near-term quantum computers. *npj Quantum Information*, 5(1):99, 2019.
- [27] Joseph Emerson, Robert Alicki, and Karol Życzkowski. Scalable noise estimation with random unitary operators. *J. Opt. B Quantum Semiclassical Opt.*, 7(10):S347–S352, October 2005.
- [28] Andrew W Cross, Easwar Magesan, Lev S Bishop, John A Smolin, and Jay M Gambetta. Scalable randomised benchmarking of non-clifford gates. *Npj Quantum Inf.*, 2(1), April 2016.
- [29] Easwar Magesan, Jay M Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Phys. Rev. A*, 85(4), April 2012.
- [30] E Onorati, A H Werner, and J Eisert. Randomized benchmarking for individual quantum gates. *Phys. Rev. Lett.*, 123(6):060501, August 2019.
- [31] Tobias Chasseur, Daniel M Reich, Christiane P Koch, and Frank K Wilhelm. Hybrid benchmarking of arbitrary quantum gates. *Phys. Rev. A (Coll. Park.)*, 95(6), June 2017.
- [32] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nat. Phys.*, 15(2):159–163, February 2019.
- [33] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [34] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G S L Brandao, David A Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben

- Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P Harrigan, Michael J Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S Humble, Sergei V Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C Platt, Chris Quintana, Eleanor G Rieffel, Pedram Roushan, Nicholas C Rubin, Daniel Sank, Kevin J Satzinger, Vadim Smelyanskiy, Kevin J Sung, Matthew D Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019.
- [35] C Neill, P Roushan, K Kechedzhi, S Boixo, S V Isakov, V Smelyanskiy, A Megrant, B Chiaro, A Dunsworth, K Arya, R Barends, B Burkett, Y Chen, Z Chen, A Fowler, B Foxen, M Giustina, R Graff, E Jeffrey, T Huang, J Kelly, P Klimov, E Lucero, J Mutus, M Neeley, C Quintana, D Sank, A Vainsencher, J Wenner, T C White, H Neven, and J M Martinis. A blueprint for demonstrating quantum supremacy with superconducting qubits. *Science*, 360(6385):195–199, April 2018.
- [36] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.
- [37] A Morvan, B Villalonga, X Mi, S Mandrà, A Bengtsson, P V Klimov, Z Chen, S Hong, C Erickson, I K Drozdov, J Chau, G Laun, R Movassagh, A Asfaw, L T A N Brandão, R Peralta, D Abanin, R Acharya, R Allen, T I Andersen, K Anderson, M Ansmann, F Arute, K Arya, J Atalaya, J C Bardin, A Bilmes, G Bortoli, A Bourassa, J Bovaird, L Brill, M Broughton, B B Buckley, D A Buell, T Burger, B Burkett, N Bushnell, J Campero, H-S Chang, B Chiaro, D Chik, C Chou, J Cogan, R Collins, P Conner, W Courtney, A L Crook, B Curtin, D M Debroy, A Del Toro Barba, S Demura, A Di Paolo, A Dunsworth, L Faoro, E Farhi, R Fatemi, V S Ferreira, L Flores Burgos, E Forati, A G Fowler, B Foxen, G Garcia, É Genois, W Giang, C Gidney, D Gilboa, M Giustina, R Gosula, A Grajales Dau, J A Gross, S Habegger, M C Hamilton, M Hansen, M P Harrigan, S D Harrington, P Heu, M R Hoffmann, T Huang, A Huff, W J Huggins, L B Ioffe, S V Isakov, J Iveland, E Jeffrey, Z Jiang, C Jones, P Juhas, D Kafri, T Khattar, M Khezri, M Kieferová, S Kim, A Kitaev, A R Klots, A N Korotkov, F Kostritsa, J M Kreikebaum, D Landhuis, P Laptev, K-M Lau, L Laws, J Lee, K W Lee, Y D Lensky, B J Lester, A T Lill, W Liu, W P Livingston, A Locharla, F D Malone, O Martin, S Martin, J R McClean, M McEwen, K C Miao, A Mieszala, S Montazeri, W Mruczkiewicz, O Naaman, M Neeley, C Neill, A Nersisyan, M Newman, J H Ng, A Nguyen, M Nguyen, M Yuezhen Niu, T E O’Brien, S Omonije, A Opremcak, A Petukhov, R Potter, L P Pryadko, C Quintana, D M Rhodes, C Rocque, E Rosenberg, N C Rubin, N Saei, D Sank, K Sankaragomathi, K J Satzinger, H F Schurkus, C Schuster, M J Shearn, A Shorter, N Shutty, V Shvarts, V Sivak, J Skrzynny, W C Smith, R D Somma, G Sterling, D Strain, M Szalay, D Thor, A Torres, G Vidal, C Vollgraft Heidweiller, T White, B W K Woo, C Xing, Z J Yao, P Yeh, J Yoo, G Young, A Zalcman, Y Zhang, N Zhu, N Zobrist, E G Rieffel, R Biswas, R Babbush, D Bacon, J Hilton, E Lucero, H Neven, A Megrant, J Kelly, P Roushan, I Aleiner, V Smelyanskiy, K Kechedzhi,

- Y Chen, and S Boixo. Phase transitions in random circuit sampling. *Nature*, 634(8033):328–333, October 2024.
- [38] Easwar Magesan, J. M. Gambetta, and Joseph Emerson. Scalable and robust randomized benchmarking of quantum processes. *Phys. Rev. Lett.*, 106:180504, May 2011.
- [39] Lev S Bishop, Sergey Bravyi, Andrew Cross, Jay M Gambetta, and John Smolin. Quantum volume. *Quantum Volume. Technical Report*, 2017.
- [40] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. *Phys. Rev. A (Coll. Park.)*, 100(3), September 2019.
- [41] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. *arXiv preprint arXiv:1612.05903*, 2016.
- [42] Sean Mullane. Sampling random quantum circuits: a pedestrian’s guide. *arXiv preprint arXiv:2007.07872*, 2020.
- [43] F Arute, K Arya, R Babbush, et al. Supplementary information for ‘quantum supremacy using a programmable superconducting processor’. *Nat. Int. Wkly. J. Sci*, 574:505–505, 2020.
- [44] Roger Balian. *From microphysics to macrophysics*. Theoretical and Mathematical Physics. Springer, Berlin, Germany, 1 edition, November 2006.
- [45] Max O. Lorenz. Methods of measuring the concentration of wealth. *Publ. Am. Stat. Assoc.*, 9(70):209–219, June 1905.
- [46] Alexander Amini, Ava P Soleimany, Wilko Schwarting, Sangeeta N Bhatia, and Daniela Rus. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, New York, NY, USA, January 2019. ACM.
- [47] Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: Theory of majorization and its applications*. Springer series in statistics. Springer, New York, NY, 2 edition, December 2010.
- [48] Amarjit Kundu, Shovan Chowdhury, Asok K. Nanda, and Nil Kamal Hazra. Some results on majorization and their applications. *Journal of Computational and Applied Mathematics*, 301:161–177, 2016.
- [49] Alan J Aw and Noah A Rosenberg. Bounding measures of genetic similarity and diversity using majorization. *J. Math. Biol.*, 77(3):711–737, September 2018.
- [50] Gilad Gour, David Jennings, Francesco Buscemi, Runyao Duan, and Iman Marvian. Quantum majorization and a complete set of entropic conditions for quantum thermodynamics. *Nat. Commun.*, 9(1):5352, December 2018.
- [51] J I Latorre and M A Martín-Delgado. Majorization arrow in quantum-algorithm design. *Phys. Rev. A*, 66(2), August 2002.
- [52] Román Orús, José I Latorre, and Miguel A Martín-Delgado. *Quantum Inf. Process.*, 1(4):283–302, 2002.

- [53] Michael Nielsen and Guifre Vidal. Majorization and the interconversion of bipartite states. *Quantum Inf. Comput.*, 1(1):76–93, July 2001.
- [54] Raúl O Vallejos, Fernando de Melo, and Gabriel G Carlo. Principle of majorization: Application to random quantum circuits. *Phys. Rev. A (Coll. Park.)*, 104(1), July 2021.
- [55] G.H. Hardy, J.E. Littlewood, and G. Pólya. *Some Simple Inequalities Satisfied by Convex Functions*. 1929.
- [56] Alfred Horn. Doubly stochastic matrices and the diagonal of a rotation matrix. *American Journal of Mathematics*, 76(3):620–630, 1954.
- [57] Peter M Alberti and Armin Uhlmann. *Stochasticity and partial order*. Deutscher Verlag der Wissenschaften Berlin, 1982.
- [58] Ernst Ruch. The diagram lattice as structural principle a. new aspects for representations and group algebra of the symmetric group b. definition of classification character, mixing character, statistical order, statistical disorder; a general principle for the time evolution of irreversible processes. *Theoret. Chim. Acta*, 38(3):167–183, 1975.
- [59] Carsten Timm. Random transition-rate matrices for the master equation. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 80(2 Pt 1):021140, August 2009.
- [60] Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. *Proc. Math. Phys. Eng. Sci.*, 464(2100):3089–3106, December 2008.
- [61] Richard Jozsa and Maarten Van den Nest. Classical simulation complexity of extended clifford circuits. *arXiv preprint arXiv:1305.6190*, 2013.
- [62] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(5), November 2004.
- [63] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Math. Phys. Eng. Sci.*, 467(2126):459–472, February 2011.
- [64] Yoshifumi Nakata and Mio Murao. Diagonal quantum circuits: Their computational power and applications. *Eur. Phys. J. Plus*, 129(7), July 2014.
- [65] Daniel Shaffer, Claudio Chamon, Alioscia Hamma, and Eduardo R Mucciolo. Irreversibility and entanglement spectrum statistics in quantum circuits. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(12):P12007, 2014.
- [66] Claudio Chamon, Alioscia Hamma, and Eduardo R Mucciolo. Emergent irreversibility and entanglement spectrum statistics. *Physical review letters*, 112(24):240501, 2014.
- [67] Ting Yu and JH Eberly. Sudden death of entanglement. *Science*, 323(5914):598–601, 2009.
- [68] Matthew Reagor, Christopher B Osborn, Nikolas Tezak, Alexa Staley, Guenevere Prawiroatmodjo, Michael Scheer, Nasser Alidoust, Eyob A Sete, Nicolas Didier, Marcus P da Silva, Ezer Acala, Joel

- Angeles, Andrew Bestwick, Maxwell Block, Benjamin Bloom, Adam Bradley, Catvu Bui, Shane Caldwell, Lauren Capelluto, Rick Chilcott, Jeff Cordova, Genya Crossman, Michael Curtis, Saniya Deshpande, Tristan El Bouayadi, Daniel Girshovich, Sabrina Hong, Alex Hudson, Peter Karalekas, Kat Kuang, Michael Lenihan, Riccardo Manenti, Thomas Manning, Jayss Marshall, Yuvraj Mohan, William O'Brien, Johannes Otterbach, Alexander Papageorge, Jean-Philip Paquette, Michael Pelstring, Anthony Polloreno, Vijay Rawat, Colm A Ryan, Russ Renzas, Nick Rubin, Damon Russel, Michael Rust, Diego Scarabelli, Michael Selvanayagam, Rodney Sinclair, Robert Smith, Mark Suska, Ting-Wai To, Mehrnoosh Vahidpour, Nagesh Vodrahalli, Tyler Whyland, Kamal Yadav, William Zeng, and Chad T Rigetti. Demonstration of universal parametric entangling gates on a multi-qubit lattice. *Sci. Adv.*, 4(2):eaao3603, February 2018.
- [69] Gates and Instructions; pyQuil 2.7.0 documentation — pyquil-docs.rigetti.com. <https://pyquil-docs.rigetti.com/en/v2.7.0/apidocs/gates.html#native-gates-for-rigetti-qpus>. [Accessed 27-12-2024].
- [70] Adam Holmes, Sonika Johri, Gian Giacomo Guerreschi, James S Clarke, and Anne Y Matsuura. Impact of qubit connectivity on quantum algorithm performance. *Quantum Science and Technology*, 5(2):025009, 2020.
- [71] Building scalable, innovative quantum systems — rigetti.com. <https://www.rigetti.com/what-we-build>. [Accessed 10-01-2025].
- [72] Mark M Wilde. From classical to quantum shannon theory. *arXiv preprint arXiv:1106.1445*, 2011.
- [73] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.
- [74] Ani Adhikari and Jim Pitman. Data 140 Textbook. <https://data140.org/textbook/content/README.html>, 2023. [Accessed 26-03-2025].
- [75] D Dubhashi and Alessandro Panconesi. Concentration of measure for the analysis of randomised algorithms. *Draft Manuscript*, <http://www.brics.dk/ale/papers.html>, 1998.
- [76] Patrick Hayden, Debbie W Leung, and Andreas Winter. Aspects of generic entanglement. *Communications in mathematical physics*, 265:95–117, 2006.
- [77] Ingemar Bengtsson and Karol Życzkowski. *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge university press, 2017.
- [78] Christopher M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, August 2016.
- [79] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, England, March 2013.
- [80] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.
- [81] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

- [82] Zineb Noumir, Paul Honeine, and Cedue Richard. On simple one-class classification methods. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2022–2026. IEEE, 2012.
- [83] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.
- [84] Nana Kwame Gyamfi and Jamal-Deen Abdulai. Bank fraud detection using support vector machine. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 37–41. IEEE, 2018.
- [85] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.
- [86] Ulrike von Luxburg and Philip Hennig. Probabilistic and statistical machine learning 2020. https://youtu.be/jFcYpBOeC0Q?si=oLn2DDw1_ty5YwCa, 2020. [Accessed 26-03-2025].
- [87] Alexandre Kowalczyk. *Support Vector Machines Succinctly*. Syncfusion, Inc, Morrisville, USA, October 2017.
- [88] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.
- [89] RBF SVM parameters — scikit-learn.org. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. [Accessed 10-03-2025].
- [90] Rui Zhang, Shaoyan Zhang, Sethuraman Muthuraman, and Jianmin Jiang. One class support vector machine for anomaly detection in the communication network performance data. In *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications*, pages 31–37. Citeseer, 2007.
- [91] Kittikun Kittidachanan, Watha Minsan, Donlapark Pornnopparath, and Phimpaka Taninpong. Anomaly detection based on gs-ocsvm classification. In *2020 12th international conference on knowledge and smart technology (KST)*, pages 64–69. IEEE, 2020.
- [92] Aleksandra Solarz, Maciej Bilicki, Mariusz Gromadzki, Agnieszka Pollo, Anna Durkalec, and Michał Wypych. Automated novelty detection in the wise survey with one-class support vector machines. *Astronomy & Astrophysics*, 606:A39, 2017.