Machine Learning Enhanced Quantum State Tomography

Steven Richard Glass

25880586@sun.ac.za

Honours Thesis

Bachelor of Data Science, Focal Area: Statistical Physics

Stellenbosch University

Supervisor: Prof. Francesco Petruccione

November 16, 2025

Contents

1 Introduction

Quantum Computing is an emergent field existing at the cutting-edge intersection of Computer Science, Mathematics and Physics. What was originally theoretical speculation has recently transformed into real-world technology, as continuous breakthroughs in hardware development brings humans ever closer to scalable quantum computers in the real world. Despite this rapid progress, modern day quantum computers remain largely constrained by environmental noise, which disrupts the fragile quantum states used in computation.

At the heart of quantum computing lies the task of reliably preparing, manipulating, and measuring quantum states. However, when subject to environmental noise these states may suffer unintended changes, and therefore the problem of verifying the actual state prepared by hardware is crucial. The goal of Quantum State Tomography (QST) is to prepare an ensemble of identical copies of a quantum state, perform repeated measurements, and create a reconstruction of the most plausible physical state according to the measurements. QST is an essential benchmark for researchers to ensure that quantum states prepared by hardware are in line with theoretical expectations.

Numerous approaches have been developed for quantum state reconstruction from experimental data, each balancing computational feasibility, physical constraints, and statistical robustness. In this work, three frameworks for reconstruction will be analysed. Maximum Likelihood Estimation (MLE), which is a classical optimisation based scheme that enforces physical constraints. Bayesian Mean Estimation (BME), which provides a probabilistic characterisation of uncertainty through posterior inference. Lastly, a Deep Neural Network (DNN) approach, which learns via a training procedure the mapping from measurements to physical states. By evaluating these methods under realistic noise and scaling conditions, this work aims to assess the trade-offs between interpretability, robustness, and computational feasibility for performing QST, and to identify promising pathways toward scalable, noise-resilient state reconstruction in practical quantum computing environments.

2 Quantum Mechanics Background

A rigorous discussion of QST first requires the establishing of a mathematical framework for quantum mechanics. What follows is a review of the formal representation of quantum states and the postulates governing their measurement. These theoretical foundations underpin how measurements of a quantum system arise, and how they may be leveraged in reconstructing the quantum state.

2.1 Quantum States

2.1.1 Pure States and Measurement

In quantum mechanics, the state of a physical system is represented by a normalised vector $|\psi\rangle \in \mathcal{H}$, where $\mathcal{H}$ denotes a complex Hilbert space. The state is said to be *pure* when it can be represented by a single state vector $|\psi\rangle$, meaning there is no classical statistical uncertainty regarding the system.

Each physical observable A of the system is represented by a Hermitian operator whose real eigenvalues correspond to possible measurement outcomes. The eigenvalue equation

$$A\,|\lambda_i\rangle = \lambda_i\,|\lambda_i\rangle\,, \tag{2.1}$$

defines the set of measurement outcomes $\{\lambda_i\}$ and their corresponding eigenstates $\{|\lambda_i\rangle\}$, which form an orthonormal basis of $\mathcal{H}$ [2, pg. 33]. States can be expanded as a linear combination of the eigenstates

$$|\psi\rangle = \sum_i c_i\,|\lambda_i\rangle\ , \tag{2.2}$$

where $c_i = \langle\lambda_i|\psi\rangle$ are complex probability amplitudes. Upon measuring A on the system $|\psi\rangle$, the probability of obtaining the outcome λ_i is given by

$$P(|\psi_i\rangle) = |\,\langle\lambda_i|\psi\rangle\,|^2 = |c_i|^2, \tag{2.3}$$

which is subject to $\sum_i |c_i|^2 = 1$. In practice, the state $|\psi\rangle$ of the system may be unknown. The probabilities need to be estimated empirically by performing repeated measurements on identical copies of the system. The goal of QST is to invert this relationship and estimate the state $|\psi\rangle$ of the system using empirical estimates of these probabilities.

2.1.2 Mixed States

Mixed states represent broader notion of quantum systems than pure states, as there may be a classical statistical uncertainty regarding what pure state the system may be prepared in. Suppose a system can be prepared in one of a finite set of pure states $\{\,|\psi_i\rangle\}_{i=1}^k$, with corresponding probability $\{p_i\}_{i=1}^k$ satisfying $\sum_{i=1}^k p_i = 1$. This probabilistic mixture of states is referred to as the *density matrix* (or density operator) ρ, which is explicitly written as [23, pg. 99]

$$\rho \equiv \sum_{i=1}^{k} p_i\,|\psi_i\rangle\,\langle\psi_i|\,. \tag{2.4}$$

The density matrix ρ provides a general representation of any quantum state and is physically valid if and only if it is Hermitian, positive semi-definite ($\rho \succ 0$) and has unit-trace ($\mathrm{Tr}(\rho) = 1$) [23, pg. 101]. Importantly, the ensemble description of ρ is not unique, as ρ can always be diagonalised as

$$\rho\ =\ \sum_{i=1}^{d} \lambda_i\ |\psi_i\rangle\,\langle\psi_i|\,, \tag{2.5}$$

where $d = \dim(\mathcal{H}) = 2^n$. The eigenvectors $\{|\psi_i\rangle\}$ of ρ form an orthonormal set, and the eigenvalues $\{\lambda_i\}$ are non-negative real numbers which sum to 1. The eigenvalues capture how the state's probability weight is distributed across its orthogonal components and determines central quantities, such as the entropy and the purity [23].

The purity γ of a quantum state ρ is defined as $\gamma \equiv \mathrm{Tr}(\rho^2)$, which serves as a measure of the mixedness of the state and always satisfies the bound $1/d \leq \gamma \leq 1$. A state is pure if $\gamma = 1$ and mixed if $\gamma < 1$, where smaller values of γ correspond to a higher degree of mixedness.

2.2 Single Qubit States

The fundamental unit of information in quantum computing is the quantum bit (qubit), which represents the quantum analogue of a classical bit. Classical computing would dictate that a bit deterministically assumes either a 0 or 1 state, corresponding to distinct physical states of the system such as low or high voltage across a wire. Qubits, however, may be in a coherent superposition of both logical states simultaneously, and only assume a specific outcome probabilistically upon measurement. A single qubit pure state can be represented by the normalised vector $|\psi\rangle \in \mathcal{H} = \mathbb{C}^2$, where the basis vectors of this Hilbert space are defined as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.6}$$

The quantum superposition of a single qubit system can be represented as a linear combination of these basis states

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\ , \tag{2.7}$$

where α and β are both complex numbers, and represent the probability amplitudes of each respective measurement outcome (0 or 1). Recall that Eq. (2.3) showed it is a necessary condition that $|\alpha|^2 + |\beta|^2 = 1$. If one writes the normalisation implicitly and ignores the global phase, one can express the state as

$$|\psi\rangle = \cos\left[\frac{\theta}{2}\right]|0\rangle + e^{i\phi}\sin\left[\frac{\theta}{2}\right]|1\rangle\,, \tag{2.8}$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$ [1, pg. 4]. These parameters define the Bloch vector

$$\mathbf{r} = \big(\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\theta\big),$$

which satisfies $\|\mathbf{r}\| = 1$ for pure states. The corresponding density matrix ρ of this pure state is given by

$$\rho = |\psi\rangle\langle\psi| = \begin{bmatrix} \cos^2\left(\frac{\theta}{2}\right) & e^{-i\phi}\,\dfrac{\sin\theta}{2} \\ e^{i\phi}\,\dfrac{\sin\theta}{2} & \sin^2\left(\frac{\theta}{2}\right) \end{bmatrix}. \tag{2.9}$$

For a single qubit state, the set of operators $\{\mathbb{I}, \sigma_x, \sigma_y, \sigma_z\}$, where $\sigma_x, \sigma_y, \sigma_z$ are the Pauli matrices and $\mathbb{I}$ is the identity operator, form an orthogonal basis for any 2×2 Hermitian matrix [2, pg. 46]. The single qubit density matrix, being a 2×2 Hermitian matrix, can be written as a linear expansion of these basis states as

$$\rho = \frac{1}{2}\left(\mathbb{I} + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\right), \tag{2.10}$$

where $r_i = \mathrm{Tr}(\sigma_i\rho)$ are the components of the Bloch vector. For the pure state above, $r_x = \sin\theta\cos\phi$, $r_y = \sin\theta\sin\phi$, and $r_z = \cos\theta$.

2.3 Multi-Qubit States

For n-qubit systems, the Hilbert space $\mathcal{H}^{(n)}$ is given by the tensor product of the constituent single-qubit Hilbert spaces

$$\mathcal{H}^{(n)} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n \cong \mathbb{C}^{2^n}. \tag{2.11}$$

The corresponding computational basis states are therefore written as

$$|q_1\rangle \otimes |q_2\rangle \otimes \cdots \otimes |q_n\rangle = |q_1 q_2 \ldots q_n\rangle, \tag{2.12}$$

where $q_i \in \{0, 1\}$. Consequently, the general n-qubit pure state is expressed as the superposition of these computational basis states

$$|\psi\rangle \;=\; \sum_{q_1,\ldots,q_n\in\{0,1\}} \alpha_{q_1,\ldots,q_n} \left|q_1 \ldots q_n\right\rangle, \tag{2.13}$$

where $\alpha_{q_1,\ldots,q_n}$ are the complex probability amplitudes, satisfying $\sum_{q_1,\ldots,q_n\in\{0,1\}} |\alpha_{q_1,\ldots,q_n}|^2 = 1$. The density matrix representation of a multi-qubit state is given by a statistical mixture, as seen in Eq. (2.4), of multi-qubit pure states. It is positive semi-definite, Hermitian, trace-one, and acting on the complex Hilbert space $\mathcal{H} = \mathbb{C}^{2^n\times 2^n}$.

The tensor product of the Pauli matrices forms an orthogonal basis for the space of Hermitian operators acting on the n-qubit Hilbert space. When used as a measurement setting, they constitute an informationally complete set of measurements referred to as *cube* measurements [6]. Let $\{\sigma_t\}_{t\in\{\mathbb{I},x,y,z\}}$ denote the single–qubit Pauli operators, with $\sigma_{\mathbb{I}} = \mathbb{I}$. The corresponding operator space is spanned by the tensor product of the single-qubit Pauli matrices [9, pg. 3]

$$\mathcal{P} = \left\{\sigma_{t_1} \otimes \sigma_{t_2} \otimes \cdots \otimes \sigma_{t_n} \,\middle|\, t_i \in \{\mathbb{I}, x, y, z\}\right\}. \tag{2.14}$$

This results in 4^n many basis operators [9, pg. 3]. This set of measurement settings is known as the Pauli-tensor basis [9], and under this basis the density matrix can be expanded as

$$\rho = \frac{1}{2^n} \sum_{t_1,t_2,\ldots,t_n=0}^{3} S_{t_1,t_2,\ldots,t_n} \left(\sigma_{t_1} \otimes \sigma_{t_2} \otimes \cdots \otimes \sigma_{t_n}\right). \tag{2.15}$$

Here the coefficients $S_{t_1,t_2,\ldots,t_n}$ are the Pauli expansion coefficients of the state. This basis is information-

ally complete, meaning measurement statistics given by the POVM elements are sufficient to uniquely reconstruct any state ρ [1]. Although other informationally complete bases exist, such as Symmetric Informationally Complete POVMs (SIC-POVMs), or Mutually Unbiased Bases (MUB), this work will only utilise the Pauli-tensor basis due to the ease of implementation and compatibility with the Pauli framework.

2.4 Quantum Measurements

Previous sections have detailed how quantum states are represented mathematically, however, it is necessary to establish how measurements performed on the systems act as the interface between the quantum and classical realms. Quantum states exist in a superposition of possible outcomes and upon measuring the system, one of these outcomes is assumed with a corresponding probability. As experimentalists, one can only observe the outcome assumed after measuring the system, but never the full picture described by the density matrix. The postulates that follow mathematically formalise the interface between the system before and after a measurement is performed.

2.4.1 General Measurement Postulate

Any measurement performed on a quantum system can be described by a set of operators $\{M_m\}$ defined on the system's Hilbert space $\mathcal{H}$. The probability of realising outcome m when measuring the state ρ is

$$P(m) \;=\; \mathrm{Tr}\big(M_m^\dagger M_m \rho\big) \;=\; \mathrm{Tr}\big(M_m \rho M_m^\dagger\big). \tag{2.16}$$

Now suppose we measure the system, and observe the outcome m, then the state of the system after the measurement is performed must collapse to

$$\rho_m \;=\; \frac{M_m \rho M_m^\dagger}{P(m)}\,. \tag{2.17}$$

The measurement operators must satisfy the completeness condition $\sum_m M_m^\dagger M_m \;=\; \mathbb{I}$ [23], with each $M_m^\dagger M_m \succeq 0$. This is the most general form of the measurement postulate, and will lay the foundations for the two specific cases of measurements: projective measurements and positive operator value measurements.

2.4.2 Projective Measurement Postulate

When the measurement operators correspond to the projectors onto the eigenstates of a Hermitian observable A, the measurements are said to be projective. Here, the measurement operators $P_i \equiv |\lambda_i\rangle\langle\lambda_i|$ are the projectors onto the i'th orthonormal eigenstate of A, whereas the outcome is the corresponding eigenvalue λ_i. The probability of obtaining the i'th outcome following a measurement on the system is given by the Born rule [23, pg. 102]

$$P(i) = \mathrm{Tr}(P_i\,\rho), \tag{2.18}$$

and the system collapses to the corresponding eigenstate following the measurement. Projective measurements represent sharp idealised measurements on a system, in which the state collapses perfectly onto a member of the orthonormal eigenbasis of A.

2.4.3 POVM Measurement Postulate

While projective measurements are perfectly idealised representations of the interface between an experimentalist and the quantum system, realistic measurements are seldom ideal and may be noisy or involve partial access to a system. A more generalised notion of measurements will be formulated, known as positive operator valued measures (POVM).

As introduced in the general measurements section, each measurement on a system can be described by an operator M_m, with a corresponding positive operator $E_m = M_m^\dagger\, M_m$, which determines the probability of each outcome through $P(m) = \mathrm{Tr}(E_m\rho)$. Importantly, these operators are the only quantities one requires to determine the measurement probabilities, so therefore the entire post collapse process can be ignored.

The set $\{E_m\}$ still satisfies the conditions $\sum_m E_m = \mathbb{I}$, and $E_m \succeq 0$ ensuring the measurement outcomes form a valid probability distribution. POVMs generalise projective measurements, as by abstracting the dynamics of the post measurement state, you can allow for measurements which are non-orthogonal or even probabilistic, making it a crucial tool for QST [23, pg. 91].

2.5 Quantum Circuits

A quantum circuit is a model for quantum computation, that displays logical operations on quantum states. Quantum circuits are comprised of a series of quantum gates, which physically alter the state of the qubits through controlled unitary evolutions. Sequentially applying quantum gates to an initial state can be thought of analogously to the manipulation of classical bits using logic gates but rather acts on quantum superpositions and entanglements.

2.5.1 Single Qubit Gates

Single qubit gates represent controlled evolutions on single qubit states, where the evolution of a closed quantum system can be described by a unitary transformation U such that $U = UU^\dagger = \mathbb{I}$. $U^\dagger$ is the adjoint of U and $\mathbb{I}$ is the 2×2 identity matrix [23, pg. 18]. For a single qubit in the state $|\psi\rangle$, the unitary transformation equation can be written as [23, pg. 81]

$$|\psi\rangle \mapsto U\,|\psi\rangle\ , \tag{2.19}$$

and for the density matrix can be written as [23, pg. 102]

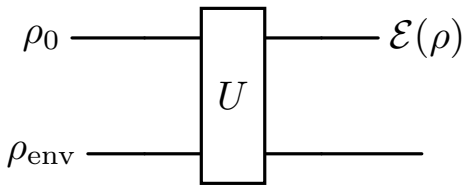

Figure 2.1: Representation of an open quantum system

$$\rho \mapsto U\,\rho\,U^{\dagger}\,. \tag{2.20}$$

Some of the fundamental single-qubit gates, which serve as the building blocks of more complicated quantum circuits, are listed as:

1. The Pauli-X gate: $\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is analogous to the classical NOT gate, and represents flipping the qubit from $|0\rangle$ to $|1\rangle$.
2. The Pauli-Y gate: $\sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$.
3. The Pauli-Z gate: $\sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.
4. The Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.
5. The general rotation gate: $R_{\hat{r}}[\theta] = e^{-i\frac{\theta}{2}\hat{r}\cdot\boldsymbol{\sigma}} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) - ir_z \sin\left(\frac{\theta}{2}\right) & (-ir_x - r_y)\sin\left(\frac{\theta}{2}\right) \\ (-ir_x + r_y)\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) + ir_z \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$.

An important theorem of general parameterised rotations about the Bloch sphere is known as the Z-Y decomposition. Suppose U is a unitary operation on a single qubit, then there exists real numbers $\alpha, \beta, \gamma, \delta$ such that U can be decomposed into

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} \end{bmatrix}. \tag{2.21}$$

2.6 Quantum Operations and the Phase Damping Channel

Closed quantum systems represent the dynamics of the system purely through a unitary operation, as it is assumed there are no interactions between the system and the environment. However, in realistic quantum systems, the state of the system is often coupled with its environment, so a more general notion of operation is required that accounts for the environment. Suppose a quantum state ρ is coupled to an environment ρ_{env} and undergoes a joint evolution determined by U as visualised in Figure 2.1. The quantum channel $\mathcal{E} : \rho \longmapsto \mathcal{E}(\rho)$ represents the reduced state of the system after undergoing an evolution paired with it's environment, while ignoring the environmental degrees of freedom. $\mathcal{E}(\rho)$ is written as

$$\mathcal{E}(\rho) = \mathrm{Tr}_{\mathrm{env}}\big(U(\rho \otimes \rho_{\mathrm{env}})U^{\dagger}\big) \tag{2.22}$$

[23, pg. 358]. Any such quantum operation can be represented using the operator-sum representation as

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^{\dagger}, \tag{2.23}$$

where $\{E_k\}$ are the Kraus operators associated with the channel, which satisfy $\sum_k E_k^{\dagger} E_k = \mathbb{I}$ [23, pg. 360]. From this formalism, many forms of error channels can be modelled, however, this work will investigate the specific case of *phase damping channels*. These channels model the loss of phase information due to environmental decoherence. Importantly, the energy eigenstates remain the same, however, the information regarding the relative phases between the eigenstates is lost [23]. The phase damping channel can be written as

$$\mathcal{E}(\rho) = E_0 \rho E_0^{\dagger} + E_1 \rho E_1^{\dagger}, \tag{2.24}$$

with Kraus operators

$$E_0 = \sqrt{1-p}\,\mathbb{I}, \qquad E_1 = \sqrt{p}\,Z, \tag{2.25}$$

where $p \in [0,1]$ represents the probability of a phase error. Acting on a general single-qubit density matrix

$$\rho = \begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{bmatrix},$$

the channel yields the transformation

$$\mathcal{E}(\rho) = \begin{bmatrix} \rho_{00} & (1-2p)\rho_{01} \\ (1-2p)\rho_{10} & \rho_{11} \end{bmatrix}. \tag{2.26}$$

This channel is particularly important in quantum optics and superconducting qubit experiments, as it models random phase fluctuations in systems weakly coupled to their environment. Such an error channel is particularly relevant in practical QST scenarios, as it can help model realistic noise that may be encountered in experiments.

3 Classical Statistical Inference Methods

The reconstruction of a quantum state from a finite set of measurement data is inherently a statistical parameter inference procedure. Within statistics, frequentism and Bayesian inference are two distinct frameworks for reasoning about probability and statistical estimation. A frequentist approach to performing QST such as maximum likelihood estimation, would treat the state ρ as some fixed but unknown quantity, and guides inference by the data only. A Bayesian approach instead treats the state as a random variable with uncertainty, and guides inference on the state not only by the data, but also by whatever prior knowledge one might have regarding it.

3.1 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a foundational method in statistics and data driven modelling, and offers a sensible approach for estimating the parameters of a statistical model given a set of observed data. Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a set independent and identically distributed (iid) random variables with probability density function (pdf) $f(x_i \,|\, \theta)$. Observed values of the random variables are written as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The likelihood of observing the data given a specified value of θ, is defined as the product over the pdf's [28]

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(x_i \,|\, \theta)\,. \tag{3.1}$$

Often one works with the logarithm of the likelihood (log-likelihood), as due to the monotonicity of the logarithm, the maximum occurs at the same value of θ, and allows for the factorisation into a summation

$$\ell(\theta) \equiv \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log f(x_i \,|\, \theta), \tag{3.2}$$

where the MLE estimator is the value of parameter θ which maximises the log-likelihood. Formally

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta} \; \ell(\theta). \tag{3.3}$$

This yields a sensible estimate for θ, as it represents the parameter of the pdf's that make the observed data most probably [28]. Another important result of the MLE is that it is asymptotically efficient (see the Cramer-Rao lower bound in Appendix A) and asymptotically normal, which means that

$$\hat{\theta}_{\mathrm{MLE}} \approx \mathcal{N}\left(\theta, \frac{1}{n\,\mathrm{I}(\theta)}\right)\,, \tag{3.4}$$

where $\mathrm{I}(\theta) = \mathbb{E}[-\frac{\partial^2}{\partial\theta^2} \log\big(f(\mathbf{x} \,|\, \theta)\big)]$ is the Fisher information, and $\mathbb{E}$ is the expectation value. This defines the theoretical variance of any MLE as $(n\,\mathrm{I}(\theta))^{-1}$.

The Fisher information cannot be computed exactly at the true parameter value θ, as this is unknown in practice. As an approximation, we substitute the MLE estimator $\hat{\theta}_{\mathrm{MLE}}$ in for θ. Additionally, the expectation value takes an integral over the space of all possible observed values, which for many purposes is intractable. Therefore when computing the Fisher information in practice, an empirical estimate over the observed data is calculated by taking the negative-Hessian at the MLE estimator

$$\hat{\mathrm{I}}(\hat{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial^2}{\partial\theta^2} \log\big(f(\mathbf{x_i} \,|\, \theta)\big)\,\big|_{\theta=\hat{\theta}}\,. \tag{3.5}$$

Here $\hat{\mathrm{I}}$ is known as the observed information. The resulting error bars for an 95% confidence interval can be written as

$$\theta \approx \hat{\theta}_{\text{MLE}} \pm 1.96 \sqrt{\frac{1}{n\hat{\mathrm{I}}(\hat{\theta})}}\,, \tag{3.6}$$

3.2 Bayesian Inference

Suppose again that $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ are iid random variables with pdf $f(x_i \,|\, \theta)$. Bayesian inference treats the statistical parameter of interest θ not as a singular unknown value (like MLE), but instead as a random variable with a probability distribution. Bayesian inference is centred around Bayes theorem, which is defined as

$$P(\theta \,|\, \mathbf{x}) = \frac{P(\theta)\, P(\mathbf{x} \,|\, \theta)}{P(\mathbf{x})} = \frac{P(\theta)\, P(\mathbf{x} \,|\, \theta)}{\int_\Theta P(\theta)\, P(\mathbf{x} \,|\, \theta)\, d\theta}\,. \tag{3.7}$$

$P(\theta)$ is known as the prior distribution function, and represents any a priori information regarding θ. $P(\mathbf{x} \,|\, \theta)$ is the likelihood of the data $\mathbf{x}$ given θ, and is defined in Eq. (3.1). The posterior distribution $P(\theta \,|\, \mathbf{x})$ is proportional to the product of these two terms, and represents how the prior belief of the distribution of θ should look upon observation of data. The evidence $P(\mathbf{x}) = \int_\Theta P(\theta) P(\mathbf{x}|\theta)\, d\theta$ is constant with respect to θ, and can be seen as simply a normalisation constant. Therefore one can write the equation up to proportionality as

$$P(\theta \,|\, \mathbf{x}) \propto P(\theta)\, P(\mathbf{x} \,|\, \theta)\,. \tag{3.8}$$

Computing the posterior distribution exactly is often analytically intractable, as the evidence involves taking an integral over the entire parameter space. However, Monte Carlo simulation methods offer an alternative approach to sampling from the posterior by using ergodic Markov Chains. The Metropolis algorithm, first developed by Metropolis et al. [21], and later generalised by Hastings [11], defines an ergodic Markov process for sampling from any target distribution π.

Consider a sequence of random variables $\left(\theta^{(t)}\right)_{t=0}^{K}$, this sequence is said to be Markovian if the probability of each $\theta^{(t)}$ depends only on $\theta^{(t-1)}$ that precedes it [11]. Additionally, an ergodic process implies that as the number of samples K drawn from an ergodic chain with stationary distribution π tends to infinity, the time average of a function g along the trajectory of samples will converge in probability to the true ensemble average. Mathematically,

$$\lim_{K\to\infty} \frac{1}{K} \sum_{t=0}^{K-1} g\left(\theta_t\right) = \mathbb{E}_\pi\left[\, g(X) \,\right]\,, \qquad \text{almost surely.}$$

The process samples values of θ by generating proposal points $\theta^\star$ according to a proposal distribution $\theta^\star \sim q(\theta \,|\, \theta^{(t-1)})$, where q can be any normalised probability distribution. The proposal point is then accepted as a sample based on the ratio

$$r = \frac{\pi(\theta^\star)}{\pi(\theta^{(t-1)})} \cdot \frac{q(\theta^{(t-1)} \,|\, \theta^\star)}{q(\theta^\star \,|\, \theta^{(t-1)})}\,, \tag{3.9}$$

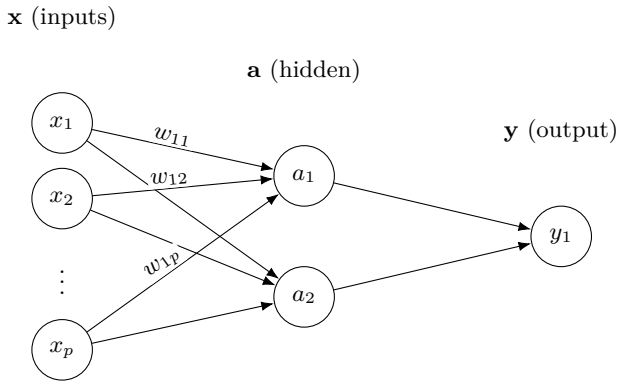

Figure 4.1: Simple feed-forward neural network with 1 hidden layer, which maps p-dimensional inputs $\mathbf{x}$ to a 1 dimensional output $\mathbf{y}$.

where the probability of accepting the proposal point is $P(\text{accept}) = \min\{1, r\}$. The Metropolis-Hastings algorithm defines an ergodic Markov process with stationary distribution $\pi(\theta)$ [11], and because of the ergodic theorem the resultant samples are treated as being effectively drawn from the stationary distribution.

Suppose we can substitute the the posterior $P(\theta \,|\, \mathbf{x}) = \frac{P(\theta)\, P(\mathbf{x} \,|\, \theta)}{P(\mathbf{x})}$ in for π and use a symmetric proposal distribution $q(\theta \,|\, \theta^{(t-1)}) = q(\theta^{(t-1)} \,|\, \theta)$, then the acceptance ratio r becomes

$$r = \frac{P(\theta^\star)\, P(\mathbf{x} \,|\, \theta^\star)}{P(\theta^{(t-1)})\, P(\mathbf{x} \,|\, \theta^{(t-1)})}\,. \tag{3.10}$$

It follows that by sampling points using the Metropolis Hastings algorithm using a symmetric proposal and acceptance ratio seen in Eq. (3.10) using the unnormalised form of the posterior, the samples will be drawn from a Markov chain with the posterior as the stationary distribution.

4 Deep Learning Background

Deep learning is a branch of machine learning which builds data driven learning models inspired by the structure of neurons in the human brain. The Multi-Layer Perceptron (MLP), seen as an example in Figure 4.1, is the quintessential architecture of deep learning, which consists of many sequential layer's of neurons connected by weighted edges and biases, which lead to a final layer of neurons which are used to perform prediction given their initial inputs. Deep learning has shown great promise in many fields of science due to it's ability to learn fast mappings and representations driven by large amounts of data, and therefore it's application to QST is investigated as a method for reconstructing quantum states.

4.1 Supervised Machine Learning

Machine Learning is a subfield of artificial intelligence which focuses on building algorithms that extract meaningful patterns from data. Supervised machine learning focuses on training models using a labelled dataset, which implies that the model learns using input predictor variables, and a ground truth target variable which the model can compare it's prediction to.

Formally, we denote the dataset as $\mathcal{D} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{n}$ where $\mathbf{x_i} \in \mathbb{R}^p$ denotes the i'th observation of the predictor variables and $\mathbf{y_i} \in \mathbb{R}^k$ denotes the corresponding i'th target. We assume the data to be generated from some unknown probability distribution: $(\mathbf{X}, \mathbf{Y}) \sim P_{\mathbf{X},\mathbf{Y}}$. The primary goal of discriminative supervised learning is to approximate a function mapping $f_\theta : \mathbb{R}^p \to \mathbb{R}^k,\ f \in \mathcal{F}$ from a hypothesis space $\mathcal{F} = \{f_\theta,\ \theta \in \Theta\}$. The model f_θ is chosen such that it minimises the *risk functional*

$$R(f_\theta) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[L(f_\theta(\mathbf{X}), \mathbf{Y})], \tag{4.1}$$

where L represents the model's loss function, which compares the ground truth target $\mathbf{Y}$ with the model's current estimate of the target $f_\theta(\mathbf{X})$, and quantifies how incorrect the model's estimation was. This expectation is approximated by averaging the empirical loss observed over each training observation

$$\hat{R}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} L(f_\theta(\mathbf{x}_i), \mathbf{y}_i). \tag{4.2}$$

The task of the statistical learning algorithm, is to choose the function f_θ such that it minimises the empirical risk. Mathematically, this is written as

$$f_\theta = \arg\min_f(\hat{R}(f))\,. \tag{4.3}$$

This process is known as *Empirical Risk Minimisation*, and provides a methodological approach for choosing the model's function f_θ.

4.2 Feed Forward Neural Networks

The Multi-Layer Perceptron (MLP) is the quintessential deep learning model, and it's architecture is the basis of all other modern deep learning approaches (Convolutional Neural Networks, Recurrent Neural Networks, etc.) [7, Ch. 6]. In the following sections, we will explore in detail how the MLP architecture and learning algorithm works, and why this provides robust, generalisable function mapping approximates.

4.2.1 Input and Hidden Units

The model takes as input the vector $\mathbf{x} = \big(x_1, x_2, \ldots, x_p\big)$, $\mathbf{x}_k \in \mathbb{R}^p$ and performs a weighted sum at each hidden layer based on the inputs of the previous layer

$$z_k = \sum_{j=1}^{p} w_{kj}\, x_j\ + b_k = \mathbf{w}_k^T\mathbf{x}\ + b_k\,, \tag{4.4}$$

where w_{kj} is the weight corresponding to the edge connecting input vertex j to hidden vertex k, which comprise the vector $\mathbf{w}_k = \big(w_{k1}, w_{k2}, \ldots w_{kp}\big)$, and b_k is the bias corresponding to the k'th hidden vertex.

It follows that these output values z_k can be written compactly in vector format

$$\mathbf{z}^{(1)} = \mathbf{W}^{(1)\,T}\,\mathbf{x} + \mathbf{b}^{(1)}\,, \tag{4.5}$$

where the subscripts of 1 indicate the first hidden layers respective outputs, weights and biases, and $\mathbf{z}^{(1)} \in \mathbb{R}^{K_1}$ (K_1 is the number of nodes in the first hidden layer). Written explicitly, the weights and biases for the first layer are

$$\mathbf{W}^{(1)\,T} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K_1 1} & w_{K_1 2} & \dots & w_{K_1 p} \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{K_1} \end{bmatrix}.$$

The output is then generally applied with an element-wise non-linear transformation function labelled ϕ, known in the literature as an *activation* function [15]. Without any activation functions, the model simply makes linear weighted predictions, so by introducing non-linear functions at each layer, more complex patterns may be extracted from the data. This function is typically specified before training, and is applied to give the final values of the hidden layer

$$\mathbf{a}^{(1)} = \phi\big(\mathbf{z}^{(1)}\big)\,. \tag{4.6}$$

This process may repeat for an arbitrary number of hidden layers, until one obtains the output layer $\hat{\mathbf{y}}$ which is treated in the exact same manner as the hidden layers (a weighted sum of the previous layer combined with a non-linear activation function).

4.2.2 Output and Loss Functions

For a neural network with K layers, the final output layer represents the final prediction of the model. For regression problems, the final layer is simply the weighted sum from the previous layer without any activation, i.e. $\mathbf{z}^{(K)} = \mathbf{W}^{(K)\,T}\,\mathbf{a}^{(K-1)} + \mathbf{b}^{(K)}$.

The model learns by minimising the empirical risk seen in Eq. (4.2), given it's current set of weights and biases. The loss function determines how risk is defined in the optimisation scheme. The loss function that will be utilised in this investigation is the Mean Squared Error (MSE), and defines the squared difference between the true observed value and the prediction as

$$L_{\text{MSE}}(\theta, \hat{\theta}) = \Big(\theta - \hat{\theta}\Big)^2\,. \tag{4.7}$$

5 Quantum State Tomography

Quantum State Tomography (QST) refers to the experimental procedure in which an ensemble of unknown and identical quantum states is repeatedly measured, and the measurements are then used to reconstruct a plausible estimate of the underlying state [13]. We will investigate QST as it is applied

to quantum computing, as it serves as a tool in experimentally validating whether quantum circuitry is behaving in accordance with theoretical expectations.

5.1 Problem Overview

For n-qubits, the full Pauli operator basis contains 4^n elements, corresponding to all tensor products of $\{\sigma_{\mathbb{I}}, \sigma_x, \sigma_y, \sigma_z\}$. However, in the Pauli tensor basis for QST, there exists only $B = 3^n$ basis settings [24], as observables including the identity operator $\sigma_{\mathbb{I}}$ do not require their own measurement settings and can be reconstructed from marginals via partial traces. For each basis setting, there exists $O = 2^n$ possible outcomes per basis. Considering all the possible outcomes across the settings, one obtains the POVM set $\{E_j\}_{j=1}^{m}$, where $m = 3^n \times 2^n$ is the total number of possible outcomes in the experiment.

The measurement probabilities are given by the Born rule from Eq. (2.18) as $p_j = \mathrm{Tr}(E_j\,\rho)\,,\ j = 1, 2, \ldots, m$, where p_j represents the probability of the outcome j. These are known as the *ideal measurement statistics*. Experimentally, these probabilities are inaccessible, as the density matrix is unknown, and instead need to be approximated using experimental frequency counts. Repeatedly measuring identical copies of a system yields a vector of measurement counts in each basis. For the b'th measurement setting, this can be expressed explicitly as

$$\mathbf{n}^{(b)} = \big(n_1, n_2, \ldots, n_O\big),$$

which over all the bases yields the set $\{\mathbf{n}^{(b)}\}_{b=1}^{B}$. The core task of QST is the mapping of the measurement counts to a realistic estimate of the quantum state ρ that generated them in the first place. The estimate should obey physicality constraints discussed in Section 2.1.2, and correctly acknowledge the degree of uncertainty regarding it's value given the data used for reconstruction. Three models of noise are investigated, (i) shot-based noise, which is characteristic of experiments where few copies of the system can be prepared [18], (ii) measurement miscalibration noise, where the POVM set is rotated using a Gaussian rotation matrix [18], (iii) phase damping channel, which is applied to the specific case of benchmark states.

5.1.1 Sampling Noise

Suppose one performs a total of N_{shots} measurements for each measurement setting on an unknown ρ, obtaining n_j many observations of the outcome j. Because each outcome has a corresponding probability p_j according to the Born rule, each of these vectors are distributed according to the multinomial distribution

$$\mathbf{n}^{(b)} \sim \text{Multinomial}\big(N_{\text{shots}}\,, p_1, \ldots\, p_O\big), \qquad \text{for } b = 1, \ldots, B. \tag{5.1}$$

By applying the weak law of large numbers seen in Appendix B, one can derive the relationship between N_{shots} and the sampling noise. The measurement statistics $\mathbf{n}^{(b)}$ for a basis setting b have expectation $\mathbb{E}[\mathbf{n}^{(b)}] = \mathbf{p}^{(b)}\, N_{\text{shots}}$, where $\mathbf{p}^{(b)} = (p_1, p_2, \ldots, p_O)$ is the vector of probabilities for that basis setting. Applying the weak law of large numbers from Eq. (3) it follows

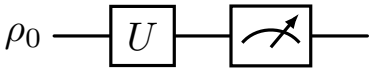

Figure 5.1: Idealised circuit

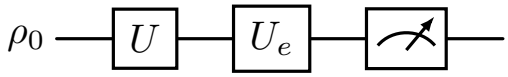

Figure 5.2: Noisy measurement circuit model

$$\lim_{N\to\infty} P\big(\big|\frac{1}{N}\sum_{i=1}^{N}\mathbf{n}_i^{(b)} - \mathbf{p}^{(b)}\, N_{\text{shots}}\big| \leq \epsilon\big) = 1 \tag{5.2}$$

$$\Rightarrow \lim_{N\to\infty} P\big(\big|\frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i^{(b)} - \mathbf{p}^{(b)}\ \big| \leq \epsilon^{\star}\big) = 1\,. \tag{5.3}$$

The observed frequencies $\mathbf{f}^{(b)} = \mathbf{n}^{(b)} \,/\, N_{\text{shots}}$ for the b'th basis setting will therefore converge in probability to the ideal measurement statistics. Resultantly, the ideal measurement statistics may be approximated using the empirical frequency of each outcomes occurrence. The empirical frequency has expectation $\mathbb{E}[\mathbf{f}^{(b)}] = \mathbf{p}^{(b)}$, and variance $\text{Var}(\mathbf{f}^{(b)}) = \frac{\mathbf{p}^{(b)}(1-\mathbf{p}^{(b)})}{N_{\text{shots}}}$, which scales inversely with the number of shots. If the number of shots is too small, the resulting variance in the estimate of the Born-probabilities may be too large. This noise is known as *shot noise*, and is intrinsic to the system.

5.1.2 Measurement Miscalibration Noise

Now consider the presence of noise in the measurement operators themselves. Suppose the quantum state is initialised as ρ_0, undergoes evolution given by the unitary U, and is measured by the ideal projective measurement operators $\{P_i\}_{i=1}^{m}$ given by the Pauli tensor basis. This process can be visualised in Figure 5.1. Now, consider the same setup, however, prior to measurement the system experiences a noise perturbation given by the unitary operator U_e. This is more realistic in real physical systems, as this models a slightly miscalibrated measurement device which is subject to imperfections. This can be visualised in Figure 5.2.

The composite measurement operators including this noise can be represented by $P_j\, U_e$, as first the noisy miscalibration is applied, and then the measurement is performed. From the POVM formalism in Section 2.4.3 it follows that the resulting POVM elements become

$$E_j = \big(P_j\, U_e\big)^{\dagger} P_j\, U_e = U_e^{\dagger} P_j^{\dagger} P_j\, U_e = U_e^{\dagger}\, P_j\, U_e\,. \tag{5.4}$$

Note that $P_j^{\dagger} P_j = P_j$, as P_j is a projective operator. Using the Z-Y decomposition [23, pg. 176], one can parameterise any physical rotation around the Bloch sphere using 3 free parameters, namely $\omega_1, \omega_2, \omega_3$. The rotation matrix U_e can therefore be written as

$$U_e(\omega_1, \omega_2, \omega_3) = \begin{bmatrix} e^{i\frac{\omega_1}{2}}\cos(\omega_3) & e^{-i\omega_2}\sin(\omega_3) \\ -i\,e^{i\omega_2}\sin(\omega_3) & e^{-i\frac{\omega_1}{2}}\cos(\omega_3) \end{bmatrix}. \tag{5.5}$$

Noise in the system will be simulated by sampling the three rotational parameters independently as $\omega_i \sim \mathcal{N}(0, \xi)$ [5], where ξ will control the variance of each axis rotation prior to measurement. This noise was introduced into the dataset generation by sampling a new rotation operator U_e for each ideal measurement projector, and then transforming via Eq. (5.4) to obtain the now noisy POVM set, which is used to sample measurement statistics.

5.2 Stokes Parameter Reconstruction

Now let us consider a practical implementation of how one can go about mapping empirical measurement statistics to an actual reconstruction of the quantum state. Recall from Section 2.2 that a single qubit state can be expressed in terms of the parameterisation

$$\rho = \frac{1}{2}\left(\mathbb{I} + \sum_{i=x,y,z} r_i\sigma_i\right). \tag{5.6}$$

The real coefficients $r_i = \mathrm{Tr}(\sigma_i\rho)$ known as the Stokes parameters, correspond to the expectation value of the Pauli observables [14]. In an ideal, noise free setting, these expectation values can be estimated directly from the measurement statistics as

$$\langle\sigma_i\rangle = P(\lambda = 1) - P(\lambda = -1),$$

where $P(\lambda = \pm 1)$ is the probability of obtaining each eigenvalue upon measurement in the corresponding Pauli basis. Experimentally, the probabilities are estimated using the empirical frequencies of each outcome, so the estimated Stokes parameters become $r_i \approx f^{(i)}(+1) - f^{(i)}(-1)$, where $f^{(i)}(\pm 1)$ represents the frequency of each observable in the corresponding Pauli basis. In the limit of infinite measurements, the Stokes parameters fully determine the quantum state ρ, and this may be extended to multiple qubits by measuring in the Pauli-tensor basis.

However, in practical QST this reconstruction procedure becomes unreliable in the presence of a finite set of measurements subject to both statistical sampling noise and experimental noise. Resulting estimates may lie outside of the physically valid Bloch sphere, and may then potentially violate physicality constraints such as having a trace not equal to one, or yielding negative eigenvalues, which violates positive semi-definiteness [14, pg. 2]. Therefore, more advanced statistical methods are required to obtain robust and physically valid reconstructions of the density matrix.

5.3 Fidelity as a Distance Measure

In quantum information theory, there exists many measures which define the similarity of two states, namely ρ and ρ' on a Hilbert space $\mathcal{H}$. In this work, we will consider the fidelity $F(\rho, \rho')$ of two states,

defined as [23]

$$F(\rho, \rho') \equiv \left(\mathrm{Tr}\left(\sqrt{\rho^{1/2}\,\rho'\,\rho^{1/2}}\right)\right)^2. \tag{5.7}$$

This expression defines the overlap between the two quantum states bounded by $0 \leq F(\rho, \rho') \leq 1$, with $F = 1$ if and only if $\rho = \rho'$. In QST, fidelity is a useful measure for evaluating the performance of the reconstruction method, as it captures the distinguishability of two states in an intuitive manner.

6 Methodology

This section outlines the methodological framework of the investigation into QST. The procedure for generating the quantum states under different forms of noise is detailed, and how the subsequent measurement statistics are obtained to build the data set. Additionally, the implementation of each statistical reconstruction method is described and analysed in detail, resulting in a reproducible workflow that displays exactly how the results are obtained.

6.1 Dataset Generation

The data $\mathcal{D} = \{(\mathbf{f}_i, \rho_i)\}_{i=1}^{N}$ consists of N independently sampled n-qubit density matrices $\rho_i \in \mathbb{C}^{2^n \times 2^n}$, and the corresponding observed frequency of each measurement outcome $\mathbf{f}_i = \frac{1}{N_{\text{shots}}}(n_1, n_2, \ldots, n_m) \in \mathbb{R}^m$.

The data is simulated using a `Python-3` and `NumPy` [10] on a classical computer contained within a single script. The script builds a dataset of N density matrices, and performs N_{shots} measurement shots within each computational basis. The following section details how the density matrices are sampled, and how the measurement is simulated to include both statistical sampling noise and measurement miscalibration noise.

6.1.1 Pure and Mixed State generation

The density matrices simulated for the purpose of evaluating the generalisability of each reconstruction method should be unbiased and uniform in Hilbert-space, as the purpose of each method is to generalise well to any unseen density matrix.

Let $\mathbb{U}(d)$ denote the set of all unitary operators acting on $\mathcal{H} = \mathbb{C}^d$ where for n-qubits $d = 2^n$. For any $U \in \mathbb{U}^d$ it is always true that

$$UU^\dagger = U^\dagger U = \mathbb{I}. \tag{6.1}$$

We are interested in uniformly and randomly sampling a unitary from this space, and this notion arises through the *Haar measure*. The Haar measure μ is the unique probability measure on $\mathbb{U}(d)$ that is invariant under left and right multiplication [22]

$$d\mu(VS) = d\mu(S) = d\mu(SV) \qquad \forall\, S \subseteq \mathbb{U}(d),\ V \in \mathbb{U}(d). \tag{6.2}$$

Here, $d\mu$ represents the infinitesimal induced Haar volume element. This invariance ensures that Haar distributed unitaries are unbiased in the Hilbert space [20]. A Haar random unitary U_{haar} induces a corresponding uniform distribution over pure states by applying it to an arbitrary fixed state $|\psi_0\rangle$ we obtain

$$|\psi_{\text{haar}}\rangle = U_{\text{haar}} |\psi_0\rangle , \tag{6.3}$$

where the resulting state $|\psi_{\text{haar}}\rangle$ is a Haar distributed pure state [22]. The corresponding density matrix representing this pure state is

$$\rho_{\text{haar}} = |\psi_{\text{haar}}\rangle \langle\psi_{\text{haar}}| . \tag{6.4}$$

This work uses the `Qiskit` function `random_statevector` from the quantum information package [16], which samples pure states according to the Haar measure. The algorithm for efficiently generating these matrices can be seen in Appendix C.

We use U_{haar} to uniformly sample pure states on the Bloch sphere by applying the unitary transformation $|\psi_{\text{haar}}\rangle = U_{\text{haar}} |\psi_0\rangle$ where $|\psi_0\rangle$ is an arbitrary fixed state, and $|\psi_{\text{haar}}\rangle$ represents the Haar random state [29, pg. 5]. The corresponding density matrix representing this pure state is $\rho_{\text{haar}} = |\psi_{\text{haar}}\rangle \langle\psi_{\text{haar}}|$. This work uses the Qiskit function `random_statevector` from the quantum information package, which samples pure states according to the Haar measure. The algorithm for efficiently generating these matrices can be seen in Appendix C.

Extending this concept to mixed states, we consider random mixed states sampled uniformly using the Hilbert Schmidt metric [30]

$$D_{HS}(\rho_1, \rho_2) = \sqrt{\text{Tr}[(\rho_1 - \rho_2)^2]} . \tag{6.5}$$

The Hilbert Schmidt metric defines a Euclidean geometry on the space of density matrices. Uniformly sampling according to this metric means sampling density matrices proportionally to the induced volume element from this geometry. Density matrices can be sampled according to the Hilbert-Schmidt metric using the Ginibre ensemble Z [30, pg. 7119], where Z is the $d \times K$ complex matrix, with independently sampled complex Gaussian entries $z_{ij} \sim \mathcal{N}(0,1) + i\mathcal{N}(0,1)$. The mixed states are then computed as [30, pg. 7119]

$$\rho_{\text{mixed}} = \frac{ZZ^\dagger}{\text{Tr}(ZZ^\dagger)} . \tag{6.6}$$

The parameter K controls the rank of the states that are generated. When $K = 1$, the generated state will have rank ≤ 1, and will always yield a pure state. However, when $K = d$, the generated states will be full-rank mixed states distributed according to the Hilbert Schmidt metric [30].

The purity of states sampled according the Ginibre Ensemble cannot be controlled, so we introduce an alternate method that is able to control the purity explicitly of the resultant state. This can be accomplished using a depolarising channel on the pure states generated according to the Haar random unitary as follows

$$\rho_{\text{mixed}} = \mathcal{E}\left(\rho_{\text{haar}}\right) = (1-p)\rho_{\text{haar}} + p\frac{\mathbb{I}}{d}, \tag{6.7}$$

which yields an isotropic mixture between the pure state and the maximally mixed state [23, pg. 397]. In this approach p directly controls the purity γ of the output state, as $\gamma = p^2(1-1/d)+1/d$. Therefore, when generating a generalised set of uniformly sampled states the former Ginibre ensemble method of sampling is used, however when directly testing the performance of different methods under a controlled variety of purities, the latter approach is used.

6.1.2 Noise Generation

Sampling noise is simulated in the system by sampling from quantum states ρ using a variable numbers of measurement shots per basis. This work investigates the effects of sampling noise over the range

$$N_{\text{shots}} \in \{10, 20, 50, 100, 1000, 10\,000\}, \tag{6.8}$$

to investigate the performance of different reconstruction methods subject to varying degrees of sampling noise.

Measurement miscalibration noise is simulated in the system by sampling the U_e matrices once per POVM element, and computing $\tilde{E}_j = U_e^\dagger E_j U_e$ to transform to the new noisy POVM. Each U_e is generated independently by sampling the independent Gaussian rotation angles $\omega_i \sim \mathcal{N}(0, \xi)$ for $i = 1, 2, 3$ to build the matrix seen in Eq. (5.5). The reason behind only generating the noisy POVM set $\{\tilde{E}_j\}$ once, is this more accurately simulates a measurement device that is systematically miscalibrated, instead of a device that projects on new randomly rotated axes every time. The variance in the Gaussian rotation angles ξ generally controls the strength of the random rotation angle. Therefore, the effects of miscalibration noise over the range

$$\xi \in \{0.01, 0.02, 0.05, 0.10, 0.15, 0.20\}, \tag{6.9}$$

is tested to investigate the effect of varying degrees of this noise on the different reconstruction methods.

6.1.3 Sampling Measurement Statistics

Once the quantum state has been generated, and the miscalibration noise has optionally been applied, then the measurement counts are sampled using the data generator coded in Python. Each POVM element E_j has a corresponding basis setting b and measurement outcome o, which are encoded by the tuple (b, o). The function `group_by_basis` takes as input a list of all these tuples, and outputs a dictionary containing each basis, and the corresponding index of it's POVM element. For example, for

1-qubit it would return

```
{X: [0, 1] , Y:[2, 3], Z:[4, 5]}.
```

The function `simulate_counts` iterates over each basis setting, and computes the vector of probabilities $\mathbf{p}^{(b)}$ with each element the probability according to the Born rule. Finally, a vector of counts $\mathbf{n}^{(b)}$ is sampled according to the Multinomial$(N_{\text{shots}}, \mathbf{p}^{(b)})$ distribution using the `numpy.random.multinomial` function from the `NumPy` package [10].

6.2 Reconstruction Methods

6.2.1 Maximum Likelihood Estimation

We now investigate the application of MLE for performing QST by now treating the density matrix ρ as the statistical parameter, and the observed measurement counts $\{\mathbf{n}^{(b)}\}_{b=1}^{B}$ as the data. Rewriting Eq. 3.1 in terms of these quantities, we obtain

$$\mathcal{L}(\rho \,|\, \mathbf{n}) = \prod_{j=1}^{m} p_j^{n_j} = \prod_{j=1}^{m} [\mathrm{Tr}(E_j\, \rho)]^{n_j}. \tag{6.10}$$

Taking the logarithm $\ell(\rho \,|\, \mathbf{n}) \equiv \log \mathcal{L}(\rho \,|\, \mathbf{n})$ of either side yields

$$\ell(\rho \,|\, \mathbf{n}) = \sum_{j=1}^{m} n_j \log\big(\mathrm{Tr}(E_j \rho)\big). \tag{6.11}$$

The MLE estimator is subject to the constraints $\rho \succeq 0$, $\mathrm{Tr}(\rho) = 1$. Prior to optimising over the log-likelihood, we parameterise the density matrix using the Cholesky decomposition [26] [1]

$$\rho = \frac{T(\theta)\, T(\theta)^{\dagger}}{\mathrm{Tr}(T(\theta)\, T(\theta)^{\dagger})}. \tag{6.12}$$

$T(\theta)$ is a unique lower triangular matrix with entries represented by the symbol θ, and any matrix ρ that can be decomposed into $\rho = T(\theta)\, T(\theta)^{\dagger}$ is considered Hermitian and positive semi-definite ($\rho \succeq 0$) [26], and we divide by the trace to ensure normalisation ($\mathrm{Tr}(\rho) = 1$). By decomposing the density matrix into this Cholesky form parameterised by θ, we have now ensured that all the physicality constraints of the density matrix are satisfied. The task is therefore estimating the parameters θ of T, and consequently reconstructing the physical density matrix ρ. The algorithm minimises the negative log-likelihood of the observed measurement counts given the current estimate of the parameterised density matrix. The optimisation goal of the MLE for QST can be written in full form as

$$\hat{\rho}_{\mathrm{MLE}} = \arg\min_{\theta} \left\{ \sum_{j=1}^{m} n_j \, \log \, \mathrm{Tr}\left(E_j \, \frac{T(\theta)\, T(\theta)^{\dagger}}{\mathrm{Tr}(T(\theta)\, T(\theta)^{\dagger})} \right) \right\}, \tag{6.13}$$

where $\hat{\rho}_{\mathrm{MLE}}$ represents the MLE estimator of the density matrix. The negative-log likelihood function along with the POVM set are implemented using `NumPy` [10] in `Python`, and the optimisation done using the `minimise` function within the `scipy.optimize package` [25]. The minimisation starts by initialising the density matrix with the uniform parameterisation, and then iteratively minimises the negative log-likelihood using the `L-BFGS-B` method from `SciPy`, which uses a Newton-Raphson approach by building an approximation of the inverse Hessian matrix (second order gradient information) at each iteration. The reason for using the Newton-Raphson approach here instead of normal gradient-descent is that the Hessian includes information regarding the curvature of the likelihood, which enables a more fast and stable convergence to the minimum [3, pg. 240].

When reporting uncertainty on a derived quantity such as the fidelity $F(\hat{\theta})$, we propagate the uncertainty of the MLE estimate using the multivariate delta method. The variance of the fidelity is approximated using [27]

$$\mathrm{Var}\big[F(\hat{\theta})\big] \;\approx\; \nabla_{\theta} F(\hat{\theta})^{\top} \, \mathrm{Cov}(\hat{\theta}) \, \nabla_{\theta} F(\hat{\theta}) \,, \tag{6.14}$$

where $\mathrm{Cov}(\hat{\theta}) \approx [n\, \hat{\mathrm{I}}(\hat{\theta})]^{-1}$ is the asymptotic covariance of the MLE. The corresponding 95% confidence interval for the fidelity is then

$$F(\theta) \;\approx\; F(\hat{\theta}_{\mathrm{MLE}}) \;\pm\; 1.96 \sqrt{\mathrm{Var}\big[F(\hat{\theta})\big]} \,, \tag{6.15}$$

providing error bars that reflect the propagated statistical uncertainty in the estimated state parameters.

6.2.2 Bayesian Inference

In practice, the MLE method often yields rank deficient estimates with one or more zero-eigenvalues [4]. Such an estimate is implausible, as it would mean an experimentalist would infer certainty that some measurement outcomes are impossible. Bayesian Mean Inference (BME) overcomes these issues by sampling values from the posterior distribution using the Metropolis-Hastings algorithm, resulting in an empirical distribution of the estimate with a probabilistic error-bound on the estimate.

Transferring Eq. (3.7) to the QST problem, the posterior up to a normalisation constant may be written as

$$P(\rho \,|\, \mathbf{n}) \propto P(\rho) \prod_{j=1}^{m} [\mathrm{Tr}(E_j \, \rho)]^{n_j}. \tag{6.16}$$

The likelihood is the same as described in previous sections, which is multiplied by the prior probability $P(\rho)$ which is a distribution encoding our prior beliefs about the state ρ. This expression is written in logarithmic form, as it helps prevent numerical underflow issues from the likelihood term

$$\log P(\rho \,|\, \mathbf{n}) \propto \; \log P(\rho) \; + \sum_{j=1}^{m} n_j \log \left(\mathrm{Tr}(E_j \, \rho) \right). \tag{6.17}$$

Algorithm 1 Metropolis-Hastings algorithm for QST

Require: POVM $\{E_j\}_{j=1}^m$, counts $\{n_j\}_{j=1}^m$ with $N = \sum_j n_j$; log-prior $\log(\pi_0(\theta))$; proposal kernel on the unconstrained parameters $q(\theta^\star \,|\, \theta)$; Cholesky decomposition inverter $\phi : \theta \mapsto \rho$; initial state $\theta^{(0)}$; iterations K; burn-in B.

Define log-likelihood $\ell(\theta) = \sum_{j=1}^m n_j \log\big(\mathrm{Tr}(E_j\, \phi(\theta))\big)$ and the log-posterior up to a normalising constant $\log(\pi(\theta)) \propto \ell(\theta) + \log(\pi_0(\theta))$.

for $t = 1, 2, \ldots, K$ **do**

Propose parameterisation $\theta^\star \sim q(\theta \,|\theta^{(t-1)})$, where $q(\theta \,|\, \theta^{(t-1)}) \equiv \mathcal{N}(\theta^{(t-1)}, \sigma^2\, \mathbb{I}_d)$

Compute acceptance ratio

$$r \;=\; e^{\log(\pi_0(\theta^\star)) - \log(\pi_0(\theta^{(t-1)})) + \ell(\theta^\star) - \ell(\theta^{(t-1)})}$$

Reconstruct from parameterisation:

$$\rho^{(t-1)} = \phi(\theta^{(t-1)})\,, \qquad \rho^\star = \phi(\theta^\star)$$

Draw $u \sim \mathrm{Uniform}(0, 1)$.

if $u < \min\{1, r\}$ **then**

$\rho^{(t)} \leftarrow \rho^\star$ ▷ accept

else

$\rho^{(t)} \leftarrow \rho^{(t-1)}$ ▷ reject

end if

end for

Keep samples $\{\rho^{(t)} : t > B,\ t \equiv 0 \pmod{\tau}\}$.

Posterior mean (BME): $\widehat{\rho}_{\mathrm{BME}} = \dfrac{1}{M} \displaystyle\sum_{t \in \mathcal{I}} \rho^{(t)}$ where $\mathcal{I}$ indexes kept samples.

The Metropolis-Hastings algorithm is utilised for sampling points from the posterior distribution, as given in Algorithm 1. The final estimator using the BME method is taken as the empirical average of the posterior samples taken according to Algorithm 1, by computing [4]

$$\hat{\rho}_{\mathrm{BME}} = \frac{1}{K} \sum_{k=1}^{K} \rho^{(k)}, \tag{6.18}$$

where K represents the number of samples drawn after the burn-in samples. Given a collection of posterior samples $\{\rho^{(k)}\}_{k=1}^K$ drawn from $P(\rho \,|\, \mathbf{n})$, the fidelity $F^{(k)}$ with respect to the target state ρ_{true} is computed for each sample. The resulting set $\{F^{(k)}\}$ forms an empirical posterior distribution over the fidelity. The Bayesian analogue of a confidence interval is known as a credible interval, which is a set of bounds containing a specified proportion of the posterior probability distribution. It represents the probability that the parameter of interest lies within a set of bounds, conditioned on the data. The 95% credible interval is taken by finding the 2.5th and 97.5th percentiles of the set $\{F^{(k)}\}$, which represent the bounds for which there exists a 95% posterior probability the fidelity lies within these bounds. This provides a much more intuitive interpretation of the error bounds of our estimate, as the bounds represents a probability of being within a certain set of values, instead of a confidence.

6.2.3 Deep Neural Network

The Deep Neural Network (DNN) is implemented using `tensorflow.keras` [19] in Python, and consists of an initial input layer with a size $m = 3^n \times 2^n$, corresponding to the number of measurement outcomes. The input consists of the obtained measurement frequencies $\mathbf{f} = \frac{1}{N_{\mathrm{shots}}}\big(n_1, n_2, \ldots, n_m\big) \in \mathbb{R}^m$. The data

generated to train the neural network is sampled using a split of 80% mixed states sampled according to the Ginibre ensemble, and 20% states sampled according to the Haar measure. This split should diversify the states seen by the DNN seen during training, and improve the generalisation capabilities.

Each dense layer in the network is followed by a batch normalisation and dropout layer to prevent the model from overfitting to the data, and ensure robustness to any variance in the training data. Batch normalization standardises activations within each batch by centring and scaling them, and dropout regularises the network by randomly deactivating a subset of neurons during training. Each dense layer has the Rectified Linear Unit (ReLU) activation function applied to increase the non-linear expressivity of the model, where the ReLU function is defined as

$$\phi(x) = \max(0, x)\,. \tag{6.19}$$

The final output layer is a linear dense layer that predicts the parameters of the Cholesky decomposition of $\hat{\rho}$ as seen in Eq. (6.12), which is given by the vector $\hat{\theta} \in \mathbb{R}^{4^n}$.

The DNN's optimisation task consists of building the non-linear function $g(\mathbf{f}) : \mathbf{f} \mapsto \theta$, and the reconstruction of the matrix is done post prediction, by substituting θ into the Cholesky decomposition form, as seen in Eq. (6.12). The DNN uses the Mean Squared Error (MSE) loss function in the θ parameter space. This means the predicted parameterisation $\hat{\theta}$, and the true parameterisation θ by computing the empirical average of the loss function

$$\mathbb{E}\left[L(\theta, \hat{\theta})\right] \approx \frac{1}{N}\sum_{i=1}^{N}\left(\theta_i - \hat{\theta}_i\right)^2, \tag{6.20}$$

where N is the batch size. This work uses a batch size of 32 to try and balance computational efficiency with accurate gradient updates. The Adam optimiser [17] is utilised to update the weights using a stochastic gradient descent optimisation, which helps prevent the algorithm from converging in sub-optimal local minima. Additionally, the training is run for a minimum of 30 epochs, and uses early stopping, a method that terminates the training process if no improvement is made past a certain number of epochs. This ensures that the optimisation procedure doesn't accidentally climb out of a global optima and converge later to a local optima.

7 Results

The results presented in this section analyse the performance of the MLE, BME, and DNN approaches for quantum state reconstruction. Each method is evaluated in terms of the scaling of computational efficiency, and the fidelity of reconstructions under varying levels of simulated noise. The results provide an insight into the practical feasibility of each method for realistic experimental application.

Method	Qubits (n)	Parameters	Training time (s)	Infer time (ms)
BME	2	N/A	N/A	22.0
MLE	2	N/A	N/A	98.70
NN	2	23 640	61	23.3
BME	3	N/A	N/A	3391.74
MLE	3	N/A	N/A	1690.94
NN	3	56 000	277	24.9

Table 7.1: Computational comparison between MLE, Bayes and the DNN for reconstruction of states sampled according to the Ginibre ensemble, with a shot-count of 1000, and no miscalibration noise. The table compares training times and inference times for the different methods for 2 and 3 qubit runs. The learning models are trained on 98 000 states, and each inference time is averaged over 2000 testing states, such that each predicted state must achieve a fidelity greater than 0.98.

7.1 Computational Efficiency Analysis

BME and MLE are not statistical learning schemes, meaning no parameters need to be inferred other than the parameters of the density matrix. MLE optimises the likelihood directly by incrementally updating ρ until convergence, whereas BME proposes and accepts samples according to the posterior distribution $P(\rho \,|\, \mathbf{n})$. In contrast, the DNN is a statistical learning model, which requires it be trained once on a complete set of data to infer it's model parameters, and then utilise the learned parameters to perform inference and predict ρ. Therefore, the DNN has an up-front time to train, but once trained, the inference should be virtually instant, as one simply performs a forward pass through the layers.

Table 7.1 showcases that the MLE and BME approach offer comparable performance at one qubit to the DNN, however, both increase exponentially in computation time as qubits increase. This is theoretically explained by both methods relying on the explicit form of the log-likelihood function $\ell(\rho)$. MLE optimises directly over the log-likelihood, whereas the BME draws points using the Metropolis-Hastings algorithm which requires the explicit form of the log-likelihood when computing the acceptance probability. It was shown that the number of POVM elements m grow with the number of qubits n as $m = 3^n \times 2^n$, and from Section 2.3 it follows that the size of POVM elements grows with $2^n \times 2^n$. This exponential growth explains the drastic increase in computational resources required for the algorithms to achieve the same level of fidelity across increasing qubits.

The number of training parameters of the DNN clearly must increase with the number of qubits in order to achieve the same fidelity. This increases the expressivity of the network, and it's ability to capture complex correlations between inputs measurements and output density matrices, however, it also increases the training time from 61s to 277s. Although the training time increases, the inference time per state remains virtually constant, as the increase in time for the network to perform one forward pass is clearly negligible.

7.2 Shot Noise Analysis

The results in Figure 7.1 show that the MLE systematically underperforms when estimating low-purity states, and improves in fidelity as the purity increases. As the number of shots-per-basis increase, the fidelity curves tend towards 1, and for a high enough number of shots-per-basis there is no deviation from a maximum fidelity regardless of the purity. The error bars additionally demonstrate the uncertainty in the estimates, as a fewer number of shots per basis leads to a greater uncertainty in the true fidelity.

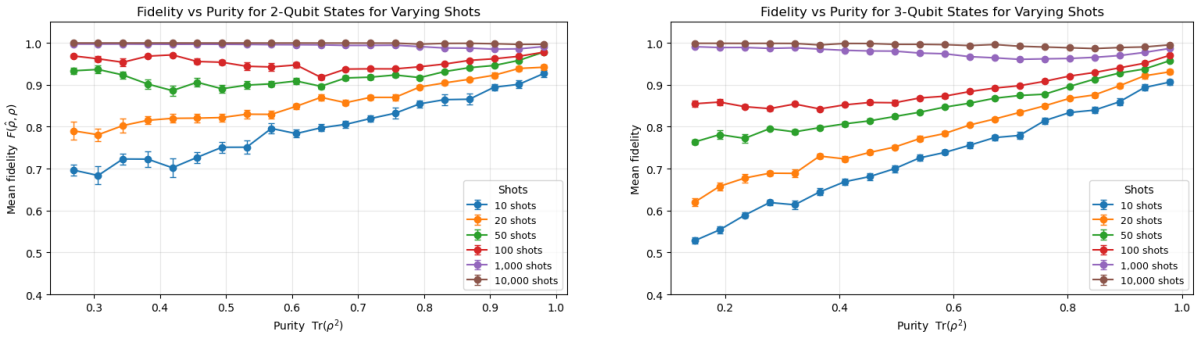

Figure 7.1: Purity vs. Fidelity for MLE reconstructions of 2-qubit (left) and 3-qubit (right) states for an increasing number of shots per basis.

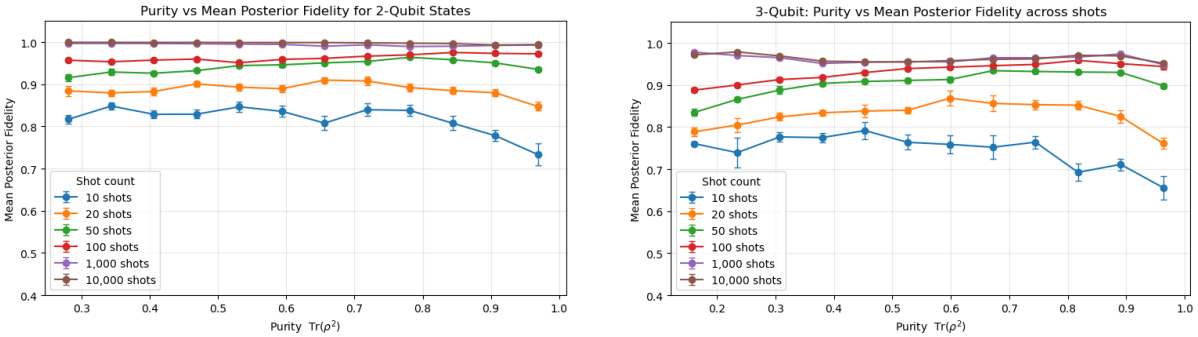

Figure 7.2: Purity vs. Expected Fidelity for Metropolis-Hastings Bayesian posterior reconstructions of 2-qubit (left) and 3-qubit (right) states for an increasing number of shots per basis.

In contrast, the BME approach shown in Figure 7.2 demonstrates a more stable trajectory across various purities for it's expected fidelity when compared to the MLE. This can likely be attributed to the BME taking an average across many candidate posterior states, as opposed to the MLE taking only one candidate state that maximises the likelihood.

In Figure 7.3 the DNN clearly displays the best performance in fidelity consistent across all purities out of all the reconstruction methods, with only a slight dip in performance at higher purities. By increasing the number of shots, the DNN can improve it's fidelity across all purities, but remains more robust to sampling noise than the other two methods as this increase is only gradual.

7.3 Basis Noise Analysis

In this section, the effect of the Gaussian rotation parameter ξ which determines the noisy measurement operators seen in Eq. (5.5) on the fidelity is investigated across varying purities. The number of shots is fixed here at $N_{\text{shots}} = 10\,000$ in order to minimise the effect of sampling noise in this experiment.

The MLE performance in Figure 7.4 shows a noticeable decrease in fidelity for increasing levels of miscalibration noise, and increasing width of the error bars on the estimations. Referring to the optimisation task established in Section 6.2.1 the log-likelihood term from Eq. (6.11) that is being optimised is dependent on the POVM set $\{E_j\}_{j=1}^{m}$. Despite rotating the basis settings using the unitary seen in Eq. (5.5), the log-likelihood term is still naively assuming that the POVM set is in the idealised form. This explains

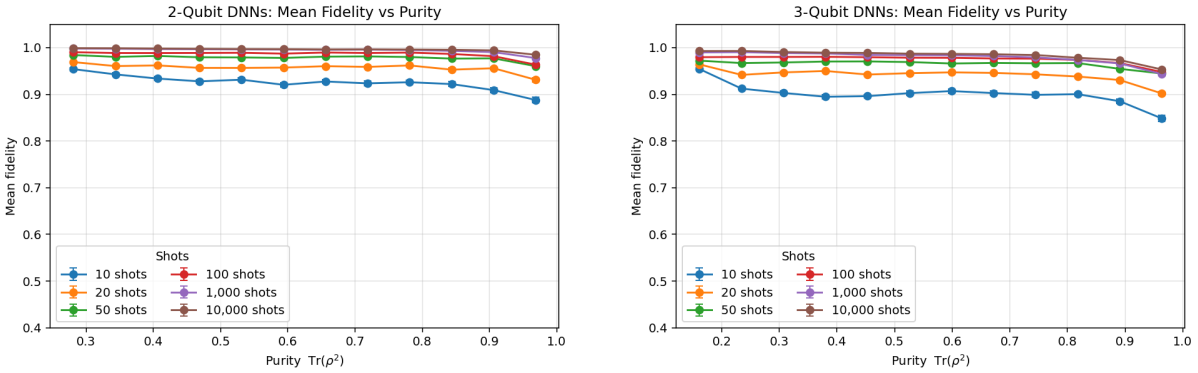

Figure 7.3: Purity vs. Fidelity for DNN reconstructions of 2-qubit (left) and 3-qubit (right) states for an increasing number of shots per basis.

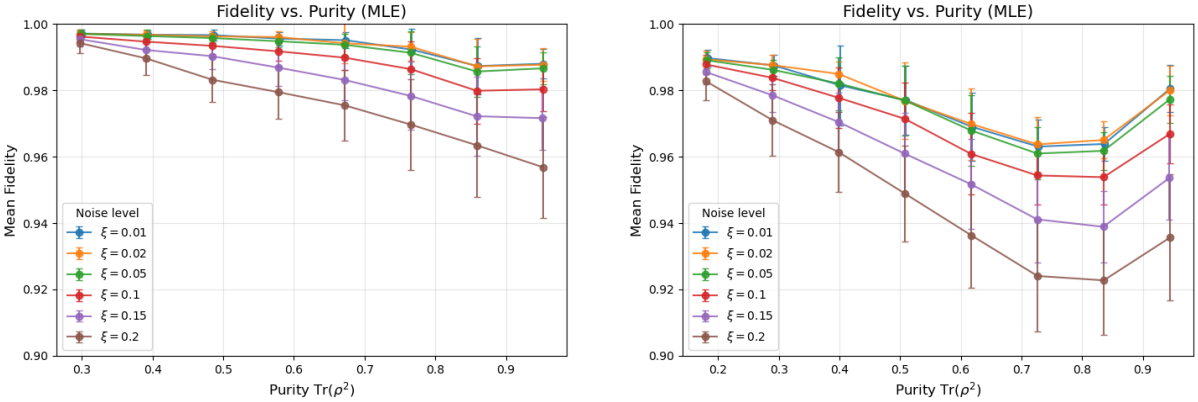

Figure 7.4: Fidelity vs Purity for 2-qubit (left) and 3-qubit (right) states for a varying amount of basis rotation noise controlled by the variance of rotation angle ξ for the MLE reconstruction.

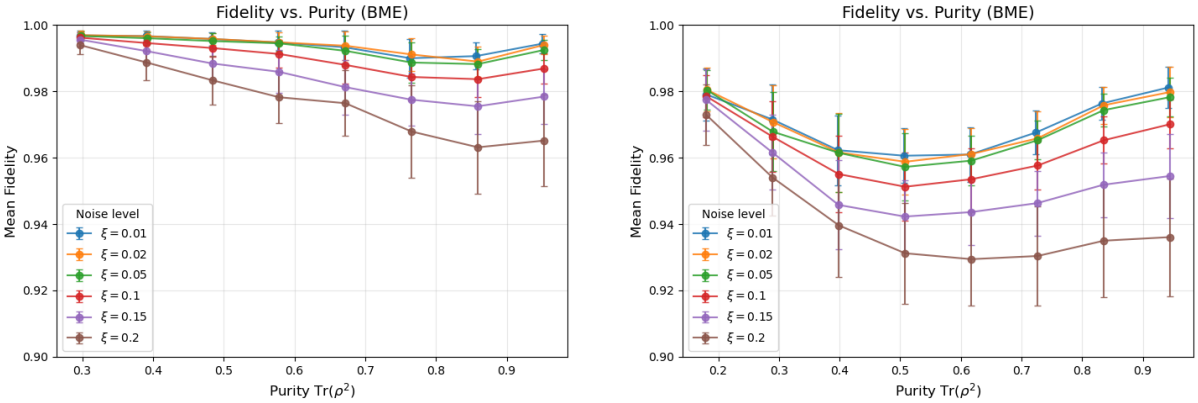

Figure 7.5: Fidelity vs Purity for 2-qubit (left) and 3-qubit (right) states for a varying amount of basis rotation noise controlled by the variance of rotation angle ξ for the BME reconstruction.

the decrease in reliable estimates, as the MLE has become systematically biased towards predicting ideal states, instead of realistic ones. This decrease in performance can be seen across all purities.

Similarly, in Figure 7.5 the BME also showcases the same drop in performance across increasing rotation noise, as the log-likelihood term in the posterior distribution still makes the same naive assumption

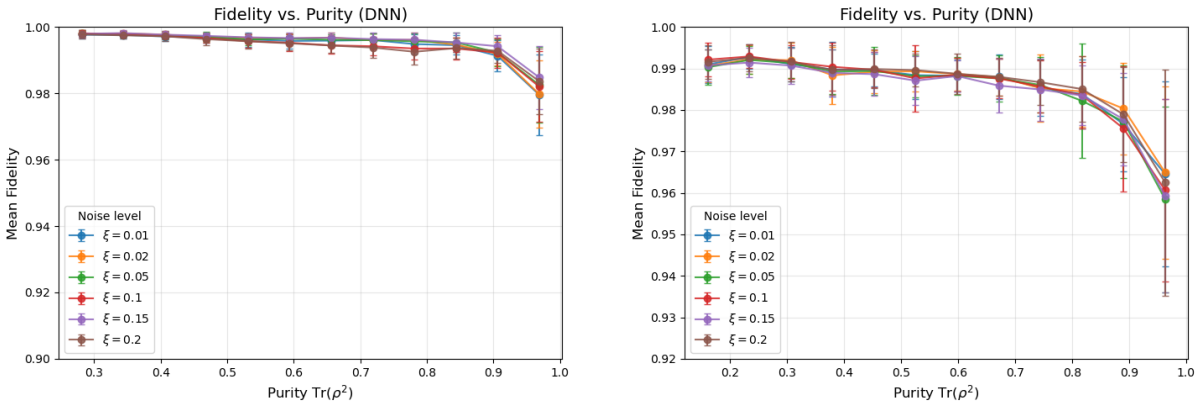

Figure 7.6: Fidelity vs Purity for 2-qubit (left) and 3-qubit (right) states for a varying amount of basis rotation noise controlled by the variance of rotation angle ξ for the DNN reconstruction.

regarding the ideal form of the POVM set. Therefore, it follows logically that the estimates of the fidelities worsen across increasing miscalibration noise for all purities, with increasing error-bounds.

In contrast, Figure 7.6 the DNN appears to show no noticeable decrease in performance across increasing miscalibration noise. This may likely be attributed to the fact that the optimisation procedure of the DNN is completely agnostic to the form of the POVM set used to perform measurements (unless it was to be explicitly included using a custom loss function), and instead focuses on minimising the squared error loss as seen in equation 6.20. Therefore, the DNN implicitly learns the rotated POVM set to be the origin of the measurement counts, and does not suffer from the systematic bias seen in the MLE and BME methods. The predicted fidelity for high purity states does however drastically decrease, and the error bars grow significantly. This can likely be attributed to the large bias towards states sampled according to the Ginibre ensemble during training, as the DNN is trained on 80% mixed states, and only 20%. In future research, a higher number of pure states should be included during training in order to combat this bias.

7.4 Eigenvalue Analysis

Each density matrix has 4 possible eigenvalues according to the orthonormal decomposition seen in Eq. (2.5), whose spread and concentration characterise the statistical mixedness of the state. Figure 7.7 displays that for each method, the lowest purity band predicts a relatively uniform distribution of eigenvalues, which is consistent with mixed states having a more even distribution of probability weight across each eigenstate. Importantly, as the purity increases, the MLE method consistently produces more estimates with exactly zero eigenvalues, which is consistent with the findings in the literature, as the MLE method tends to converge to the boundary of the physical state space [4]. In contrast, the BME spectra appear much smoother with less of a spike near zero, reflecting the Bayesian averaging of posterior values that regularises any non-physical state samples. The DNN method predicts less states with exactly zero eigenvalues than the MLE method, however, the spread is still highly jagged with a high concentration of eigenvalues near but not exactly at zero. The predicted eigenvalues of the DNN method is largely dependent on the data it was trained on, and because this DNN was trained on states with varying purity, it has learnt to not produce estimates with exactly zero-eigenvalues. However, if one was to train a DNN on entirely pure states, it could be hypothesised that subsequent predictions would

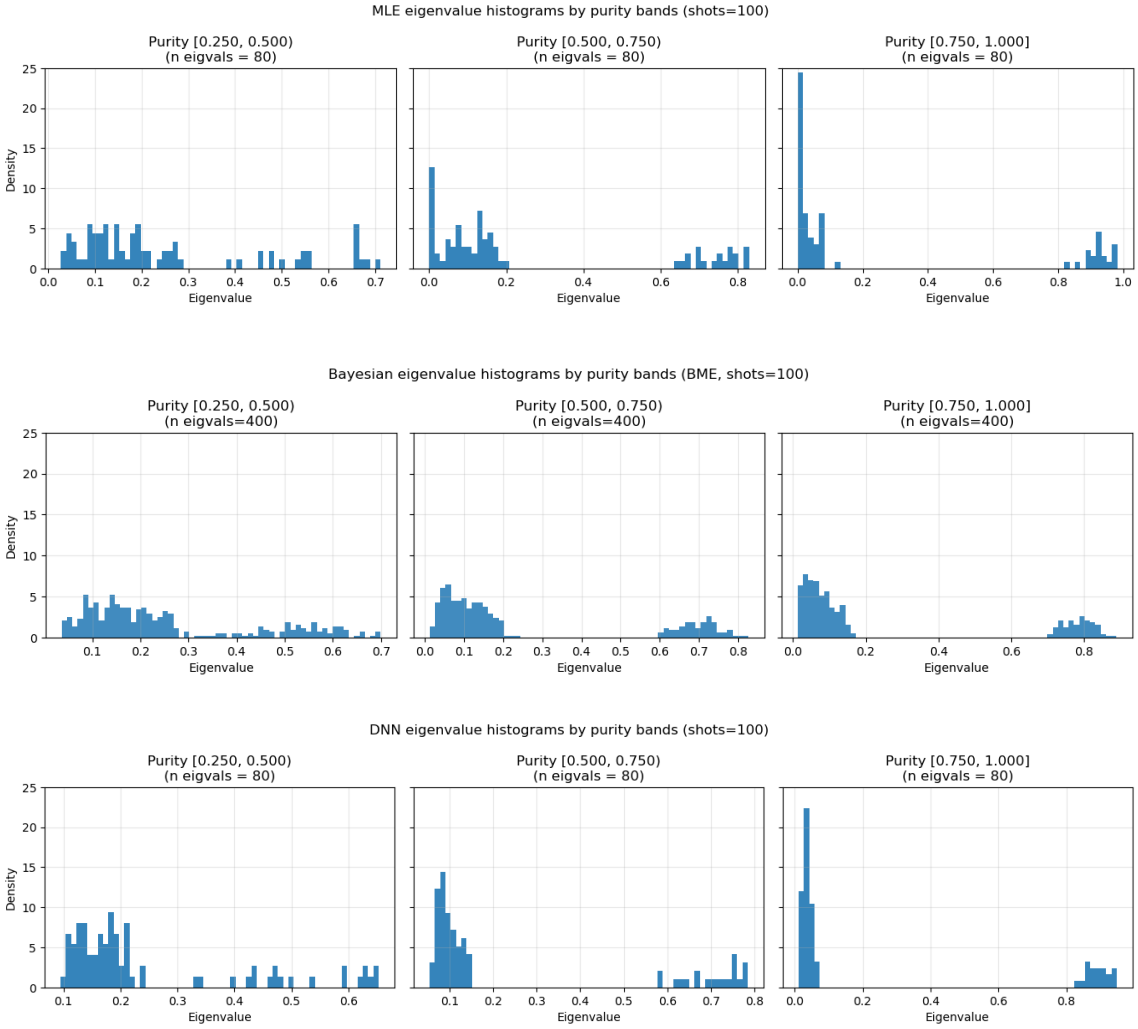

Figure 7.7: Normalised histograms of eigenvalues for 2-qubit reconstructions using the MLE (top), BME (middle) and DNN (bottom) methods, each reconstructed with 100 shots per basis over 1 000 states. The three panels in each subplot correspond to different purity ranges, showing how the eigenvalue spectrum evolves from mixed to pure states.

contain a large amount of exactly zero-eigenvalues.

7.5 Generalised QST Model using Transfer Learning

This section focuses on the generalisation capabilities of the DNN model discussed in previous sections, by applying it to a specific family of known quantum states. The Greenberger-Horne-Zeilinger (GHZ) state [8] for 3-qubits can be written as

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}\big(|000\rangle + |111\rangle\big)\,, \tag{7.1}$$

where $|\mathrm{GHZ}\rangle$ is used to denote the pure GHZ state. The GHZ states can be generalised to n-qubits, moving past Bell's theorem in generalising maximally entangled states past 2-qubit systems. GHZ states

are a canonical benchmark family of states for multipartite entanglement, and provide a practical test case to evaluate the DNN's performance outside of uniformly sampled states seen in Section 6.1.1.

The manifold of testing GHZ states are sampled using the standard GHZ form from Eq. (7.1), introducing a relative phase term between the components, and by applying a phase damping channel to the system. The initial GHZ state with a relative phase term is

$$|\mathrm{GHZ}(\phi)\rangle = \frac{1}{\sqrt{2}}\big(|000\rangle + e^{i\phi}\,|111\rangle\big), \tag{7.2}$$

with associated density matrix

$$\rho_{\mathrm{GHZ}}(\phi) = |\mathrm{GHZ}(\phi)\rangle\langle\mathrm{GHZ}(\phi)|\,. \tag{7.3}$$

The phase damping channel is then applied to the density matrix to obtain the final testing states

$$\rho_{\mathrm{test}} = \mathcal{E}\big(\rho_{\mathrm{GHZ}}(\phi)\big) = E_0\rho_{\mathrm{GHZ}}(\phi)E_0^\dagger + E_1\rho_{\mathrm{GHZ}}(\phi)E_1^\dagger\,, \tag{7.4}$$

where ρ_{test} is a testing GHZ state, $\mathcal{E}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger$ is the phase damping channel with Kraus operators $E_0 = \sqrt{1-p}\,\mathbb{I}, \quad E_1 = \sqrt{p}\,Z$. The real valued $\phi \in [0, 2\pi]$ controls the relative phase between the two components, and the real valued $p \in [0, 1]$ controls the amount of phase damping.

Including a relative phase term factors for more general cases of the GHZ-state, in which the coherence acquires an arbitrary phase. The phase damping channel as seen in Section 2.6, destroys the off-diagonal terms, while leaving the diagonal terms untouched, effectively destroying information regarding the relative phase. Because the GHZ states' entanglement relies primarily on maintaining the phase coherence between the $|000\rangle$ and $|111\rangle$ term, it is extremely sensitive to such noise. Upon sampling each GHZ-state, a relative phase is sampled according to $\phi \sim \mathrm{Uniform}[0, 2\pi]$, and a probability of phase error term is sampled according to $p \sim \mathrm{Uniform}[0, \frac{1}{2}]$, and the state formed from Eq. (7.4) is measured according to the method described in Section 6.1.3.

Transfer learning is an extension of the DNN framework in which knowledge acquired from one task can be employed to increase performance on a related task. Transfer learning is generally used to transfer a model trained on a large amount of generalised data, to a task that is concerned with a smaller manifold of task data. The QST transfer learning model is implemented in the same fashion as the DNN described in Section 6.2.3, however, the model is now divided into two separate neural-network components:

- Backbone: The backbone is built from multiple dense neural network layers with ReLU activations, batch normalisation, and dropout regularisation, as before in Section 6.2.3. The backbone effectively learns a dense representation of the measurement statistics.
- Task head W: a single dense linear layer mapping the extracted features from the backbone to the θ parameter space

$$\hat{\theta} = Wh + b, \qquad \hat{\rho} = \frac{T(\hat{\theta})\,T(\hat{\theta})^\dagger}{\mathrm{Tr}(T(\hat{\theta})\,T(\hat{\theta})^\dagger)}. \tag{7.5}$$

Training proceeds in two stages. First the backbone is trained on a dataset of 100 000 sampled density

Method	Training time (s)	Infer time (ms)
Backbone-DNN	277.27	27.14
Transfer-DNN	2.56	20.64

Table 7.2: Computational comparison between Backbone-DNN, and Transfer-DNN. The Backbone is trained on 100 000 states sampled according to the Haar and Ginibre ensemble, and task head is trained on 2 000 phase-damped GHZ states. The infer time is run on a single testing GHZ state.

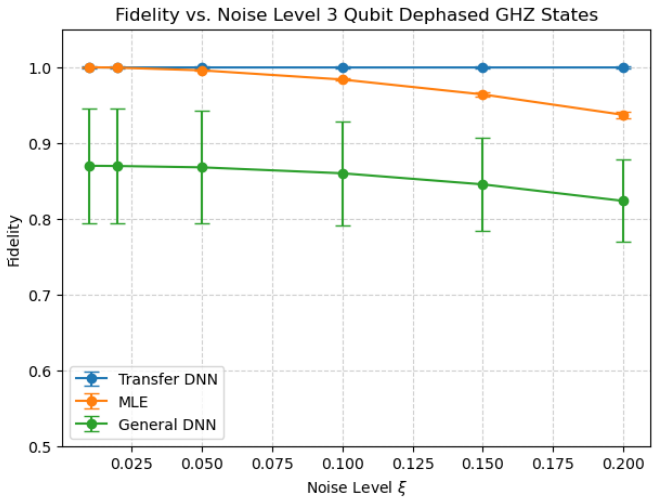

Figure 7.8: Fidelity vs. Noise level for the MLE (baseline), DNN before transfer learning (General DNN), and DNN after transfer learning (Transfer DNN). Each method is tested on the same testing set of 1 000 phase-damped GHZ states, and the average fidelity of the predictions with error-bars is computed.

matrices ρ, generated according to the Ginibre ensemble. Next, the weights of the backbone are frozen (i.e. cannot be updated), and the task head W is retrained on a set of 2 000 GHZ-states.

The loss function remains unchanged from Section 6.2.3, as well as the model architecture, batch size, optimiser, and epoch count. Theoretically, this two-phase procedure should allow the model to retain general QST knowledge, while efficiently adapting to new families of quantum states.

Table 7.2 shows the training time of the transfer model is substantially less than the time it takes to train the backbone DNN. This matches our theoretical expectations, as the transfer learning phase only updates one layer's parameters when transferring.

The results in Figure 7.8 show the fidelity of the MLE (as a baseline), pre-trained generalised DNN, and transfer-learning DNN on the same set of de-phased GHZ states for varying levels of miscalibration noise ξ. It can be observed that the generalised DNN performs the worst out of the three, as it only possesses a general understanding of a finite set uniformly sampled states, performing poorly on the phase-damped GHZ-states across the different noise levels. The MLE exhibits strong performance for low amounts of noise, however, as established in Section 7.3 breaks apart at high levels of miscalibration noise. The transfer learning model, however, outperforms all of them, exhibiting robustness to the miscalibration noise, and generalising well to the GHZ states.

8 Conclusion

This study aimed to investigate the implementation of three quantum state reconstruction methods, under realistic experimental conditions modelled by finite sampling noise and measurement miscalibration noise. Finally, a novel transfer learning approach was applied to maximally entangled states subject to a phase damping channel, to model a real world application of the deep learning framework for QST.

The MLE method demonstrates high fidelity reconstructions, and remains an efficient and conceptually intuitive estimation approach for lower dimensional states. However, MLE is prone to generating rank deficient states, and with finite data these predictions cannot be justified with certainty. Moreover, the exponential scaling of computational requirements for increasing number of qubits and sensitivity to measurement miscalibration significantly limits it practical usability. The BME method addresses some of these key issues by introducing a Bayesian framework to QST by averaging over posterior samples. Resultantly, BME yields states that suffer less from rank deficiency, and performs consistently across all purities. Nonetheless, BME still suffers the same computational scaling issues as MLE, as convergence of the algorithm depends on the number of posterior samples with a pre-specified burn in, and sensitivity to miscalibration noise is still present.

The DNN-based reconstruction offered a unique perspective, as the explicit likelihood optimisation is replaced with a learnt mapping from measurement statistics to valid density matrices. Once trained, the model performed state reconstructions almost instantly for both two and three qubits, and achieved fidelities comparable to the classical estimators. Although the required number of parameters and training time increases between two and three qubits, once trained the network was able to perform inference instantly and repeatedly, and demonstrated a greater robustness to miscalibration noise than the classical approaches. This general DNN approach was then extended to a novel implementation of transfer learning for QST, which was tested on the GHZ states with phase damping to illustrate it's practical effectiveness. The generalisability of the DNN is fine tuned to the set of GHZ states, which displayed an increase in fidelity and noise robustness relative to an MLE and general DNN approach. This demonstrates the potential of learning-based methods to generalise across noise models and specific testing states more effectively than purely statistical ones.

In future work, it would be important to address the exponential scaling issue of the inputs for the DNN. This occurs due to the number of observables growing exponentially with the number of qubits. A promising approach would be to explore *shot-based* learning approaches, which trains models on individual measurement outcomes instead of aggregated measurement histograms. By training a model on dense representations of individual measurement outcomes, one could potentially avoid the exponential scaling problems, and therefore run an even more efficient neural network approach for QST.

Code Availability

All the code used in this work is available publicly on GitHub at the following repository https://github.com/SteveG365/QST-Honours-Project/tree/main.

Bibliography

[1] J.B Altepeter. Photonic State Tomography. In *Advances In Atomic, Molecular, and Optical Physics*, pages 105–159. Elsevier, 2005. ISSN: 1049-250X.

[2] Stephen M. Barnett. *Quantum information*. Oxford scholarship online. Oxford University Press, Oxford, 2020.

[3] Christopher M. Bishop. *Pattern recognition and machine learning (information science and statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[4] Robin Blume-Kohout. Optimal, reliable estimation of quantum states. *New Journal of Physics*, 12(4):043034, April 2010.

[5] A. Czerwinski. Quantum state tomography of four-level systems with noisy measurements. *Acta Physica Polonica A*, 139(6):666–672, June 2021. Publisher: Institute of Physics, Polish Academy of Sciences.

[6] Mark D. de Burgh, Nathan K. Langford, Andrew C. Doherty, and Alexei Gilchrist. Choice of measurement sets in qubit tomography. *Physical Review A*, 78(5), November 2008. Publisher: American Physical Society (APS).

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[8] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going beyond bell's theorem. In Menas Kafatos, editor, *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer Netherlands, Dordrecht, 1989.

[9] Lukas Hantzko, Lennart Binkowski, and Sabhyata Gupta. Fast generation of Pauli transfer matrices utilizing tensor product structure. *Physica Scripta*, 100(7):075125, July 2025. Publisher: IOP Publishing.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[11] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[12] Oliver C. Ibe. Chapter 6 - functions of random variables. In Oliver C. Ibe, editor, *Fundamentals of applied probability and random processes (second edition)*, pages 185–223. Academic Press, Boston, second edition edition, 2014.

[13] Nouhaila Innan, Owais Ishtiaq Siddiqui, Shivang Arora, Tamojit Ghosh, Yasemin Poyraz Koçak, Dominic Paragas, Abdullah Al Omar Galib, Muhammad Al-Zafar Khan, and Mohamed Bennai. Quantum state tomography using quantum machine learning. *Quantum Machine Intelligence*, 6(1):28, May 2024.

[14] Daniel F. V. James, Paul G. Kwiat, William J. Munro, and Andrew G. White. Measurement of qubits. *Physical Review A*, 64(5), October 2001. Publisher: American Physical Society (APS).

[15] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An introduction to statistical learning with applications in python.* Springer texts in statistics. Springer, Cham, 2023.

[16] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, James Lishman, Julien Gacon, Simon Martiel, Paul Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with qiskit, May 2024.

[17] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization, 2017. arXiv: 1412.6980 [cs.LG].

[18] Hailan Ma, Daoyi Dong, Ian R. Petersen, Chang-Jiang Huang, and Guo-Yong Xiang. Neural networks for quantum state tomography with constrained measurements. *Quantum Information Processing*, 23(9):317, September 2024.

[19] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[20] Olivia Di Matteo. Understanding the Haar measure. *PennyLane Demos*, March 2021. Publisher: Xanadu.

[21] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. 21(6):1087–1092, June 1953.

[22] Francesco Mezzadri. How to generate random matrices from the classical compact groups, February 2007. arXiv:math-ph/0609050.

[23] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition, December 2010. ISBN: 9780511976667 Publisher: Cambridge University Press.

[24] Luciano Pereira, Leonardo Zambrano, and Aldo Delgado. Scalable estimation of pure multi-qubit states. *npj Quantum Information*, 8(1):57, May 2022.

[25] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[26] Prateek Kumar Vishwakarma. Cholesky decomposition for symmetric matrices over finite fields, 2025.

[27] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer texts in statistics. Springer, New York, NY, 2004.

[28] Lisa Yan. Maximum Likelihood Estimation.

[29] Karol Zyczkowski, Karol A. Penson, Ion Nechita, and Benoit Collins. Generating random density matrices, 2010. tex.howpublished: arXiv preprint arXiv:1010.3570.

[30] Karol Zyczkowski and Hans-Jürgen Sommers. Induced measures in the space of mixed quantum states. *Journal of Physics A: Mathematical and General*, 34(35):7111–7125, August 2001. Publisher: IOP Publishing.

9 Appendix

A Cramer Rao Lower Bound

The lower bound of the variance of any unbiased estimator T of the parameter θ is given by the Cramér-Rao lower bound

$$\mathrm{Var}(T) \geq \frac{1}{n\,\mathrm{I}(\theta)}, \tag{1}$$

where $\mathrm{I}(\theta) = \mathbb{E}[-\frac{\partial^2}{\partial\theta^2} L(\theta \,|\, \mathbf{x})]$ is the Fisher information. The MLE $\hat{\theta}_{\mathrm{MLE}}$ reaches this lower bound

$$\mathrm{Var}(\hat{\theta}_{\mathrm{MLE}}) = \frac{1}{n\,\mathrm{I}(\theta)} \tag{2}$$

B Weak Law of Large Numbers

The weak law of large numbers states that for a sequence of identical and independently distributed random variables $\big(X_1, \ldots, X_N\big)$, where $X_i \sim f(x)$ with mean μ_X, that for the unbiased estimator $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$ and any $\epsilon > 0$ [12]

$$\lim_{N\to\infty} P(|\bar{X} - \mu_X| \leq \epsilon) = 1\,. \tag{3}$$

C Haar Random Unitary Sampling

The algorithm for sampling Haar random unitaries is as follows:

1. Define the $d \times d$ complex Gaussian matrix Z with independent and identically distributed entries $z_{ij} \sim \mathcal{N}(0,1) + i\mathcal{N}(0,1)$. This is the Ginibre Ensemble [30].
2. Apply a QR decomposition $Z = QR$.
3. Compute the diagonal matrix $\Lambda = \text{diag}(\frac{R_{ii}}{|R_{ii}|})$ to ensure the phases of R's diagonal terms are normalised.
4. Compute $U_{\text{haar}} = Q\Lambda$.

[22], [20].