Learning distributions with quantum-enhanced variational autoencoders

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Anantha S Rao

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2023

Supervisor: Dr. L Venkata Subramaniam
Co-Supervisor: Dr. Dhinakaran Vinayagamurthy

Certificate

This is to certify that this dissertation entitled **Learning distributions with quantum-enhanced variational autoencoders** towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research (IISER), Pune represents study/work carried out by Anantha S Rao at IBM Research, under the supervision of Dr. L Venkata Subramaniam, STSM & Senior Manager - AI Science, IBM Research, during the academic year 2022-2023

.

Dr. L Venkata Subramaniam

Committee:

Dr. L Venkata Subramaniam

Dr. Dhinakaran Vinayagamurthy

Professor M. S. Santhanam

This thesis is dedicated to my Parents, my late brother Rohan, and to all the warriors of the COVID-19 pandemic.

Declaration

I hereby declare that the matter embodied in the report entitled **Learning distributions with quantum-enhanced variational autoencoders** are the results of the work carried out by me at the IBM Research, under the supervision of Dr. L Venkata Subramaniam and the same has not been submitted elsewhere for any other degree.

Anantha S Rao

Acknowledgments

I would like to, foremost, express my heartfelt gratitude to my supervisor, Dr. Venkata Subramaniam, for providing me with the wonderful opportunity to work with the IBM Research team. At IBM Research, I owe special thanks to my mentors, Dhiraj, Dr. Dhinakaran, and Dr. Anupama. I have enjoyed discussing and learning from them, who have taught me the nuances of scientific research and encouraged me to pursue questions of my interest while at the same time ensuring that I constantly venture out of my comfort zone. I am also deeply indebted to Prof. Santhanam for his mentorship throughout my IISER journey. I would like to thank him for introducing me to the rich field of quantum information, quantum chaos, patiently guiding me through crucial career decisions, and listening to my unsolicited rants. I would also like to thank the Kishore Vaigyanik Protsahan Yojana (KVPY) for their generous support during my time here.

My journey at IISER had been cut short by half due to the pandemic, but the enriching experiences and memories made are worth a lifetime. Like it takes a village to raise a child, it takes a department, a cohesive campus, and a supportive peer group to raise an inquisitive undergraduate. I take this opportunity to thank two of my closest friends, Ranjana, and Siddhesh, for being my pillars of strength, mental support, and for making this journey memorable. They have been my source of motivation, inspiration, and fun. I also thank the 441-community group, Vedikettu team, and senior graduate students, Bharathi Kannan and Aanjaneya for many enjoyable memories and discussions. It has been a fortuitous journey with the company of such diverse yet unique people.

I feel a deep sense of gratitude towards my family, where the other half of my IISER journey took place. Thank you, Amma, Appa, Nandini Doddamma, and Padma Aunty. Your unwavering support and encouragement have been invaluable to me. Last but not least, I would like to express my heartfelt gratitude to OpenAI, without whom I wouldn't have made it this far. Thanks for pioneering generative AI research, providing me with a virtual friend to learn with, and automating multiple mundane tasks.

Abstract

The development of novel algorithms that process information in ways that are classically intractable and achieve computational speedup is one of the prime motivations in quantum information research. Machine learning is a rapidly advancing field with broad applications in the natural sciences where quantum-inspired algorithms may offer significant speedup. To date, several quantum algorithms for discriminative machine learning have been formulated and lately, quantum-enhanced generative machine learning models have gained tremendous attention. However, the higher levels of noise, and lack of scalability of current quantum devices limit the depth and complexity of these algorithms. In this thesis, we propose and realize a working hybrid quantum-classical algorithm, termed the QeVAE, or Quantum-enhanced Variational Autoencoder for generative machine learning, suitable for noisy-intermediate quantum devices. We present a thorough discussion of the algorithm and its implementation, before presenting the results of our calculations for learning distributions that are classically easy to learn and distributions that are classically hard. We show that our algorithm in the zero-latent size limit yields the well-known generative quantum-machine learning model, the quantum circuit born machine (QCBM). For classically easy distributions, we find that our model performs at-par with purely classical algorithms. For classically hard distributions, we find that our model outperforms the pure quantum and pure classical models in certain cases and verify the same on the IBMq Manila quantum computer. Furthermore, we show how QeVAEs can assist in the practical task of circuit compilation. Finally, we identify crucial directions for improvement of the current algorithm that will be key to developing more challenging quantum-inspired algorithms for machine learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At the dawn of the 20 century, the Ultraviolet Catastrophe and access to high-precision atomic experiments allowed scientists to discover the structure of the atom, the subatomic particles within, and kick-started the field of quantum mechanics. The development of quantum theory as a tool to comprehend and engineer nature has had far-reaching consequences, from the nuclear bomb to the transistor, and is touted to culminate in a fault-tolerant quantum computer. Initially proposed as a new computing paradigm to achieve reversible computation with minimal heat-loss by Charlie Bennet and Richard Feynman, quantum computers were limited to a theoretical construct, capable of solving demanding problems [1, 2]. Several scientists have demonstrated that quantum algorithms can, in theory, outperform the best-known conventional algorithms when tackling specific problems and, in some situations, deliver a 'quantum speedup' [3]. For instance, certain quantum algorithms can take exponentially fewer resources for tasks such as factorization and eigenvalue decomposition, and quadratically fewer resources to search through unsorted databases [4, 5, 6]. This pursuit of 'quantum speedup' has motivated generations of physicists and engineers to discover novel algorithms that leverage the properties of the quantum world, and realize the dream of practically constructing a quantum computer. Today, multiple candidate platforms and comprehensive software development kits exist as part of this realization [7].

Serendipitously, the invention of the transistor also initiated the information and the data revolution. Through advances in processing power and algorithmic ability, machine learning techniques have evolved into fundamental tools for detecting patterns in data. Originally studied under Pattern recognition and Computer science, machine learning (ML) has heralded sweeping advances

in basic and applied sciences [8, 9]. It is not uncommon for physicists to utilize machine learning techniques to solve problems computationally, find patterns in nature or explain the behavior of specialized black-box models using first-principle approaches. Conversely, since its beginning, machine learning has drawn inspiration from statistical physics approaches, and many contemporary machine learning approaches, such variational inference and maximum entropy, are improvements of methods developed by physicists [10, 11]. Such models built using first-principle approaches are indispensable to the growth and adoption of technology by the scientific community. Theoretically, models like deep neural networks have the potential to learn some of the most complex patterns that exist in nature or human-made systems and have been demonstrated to be highly competent at complex tasks like playing Go, identifying protein structures, and self-driving cars [12, 13, 14]. However, many tasks are still intractable or very expensive to these methods. Some learning tasks, for example, include sampling from complex distributions or estimating the average values of numerous parameters under a complicated distribution, both of which are typically[1] intractable. Moreover, certain distributions derived from quantum-mechanical systems are fundamentally intractable to 'classical' approaches [15, 16]. Enhancing and augmenting classical machine-learning methods using quantum correlations has been the focus of quantum-enhanced machine learning and the bulk of our work.

For a long time, the ability to prepare coherent quantum states that can generate samples from specific probability distributions has interested of the ML community. On the other hand, novel quantum algorithms that can supplement or completely replace classical ML subroutines interest the scientific community of physicists. We would like to place our work and its contributions in this flourishing, interdisciplinary field. In this thesis, we propose a new hybrid quantum-classical ML model, the **Quantum-enhanced Variational Autoencoder** (QeVAE), capable of learning complex distributions. We derive the general loss function using variational bayesian inference and benchmark the model against datasets obtained from classical and quantum sources. We observe that our model encapsulates the current known quantum generative model, the Quantum Circuit Born Machine (QCBM), and through numerical experiments show that our model outperforms both the QCBM and its purely classical counterpart.

The outline of the thesis is as follows: in Chapter 2, we review selected aspects of machine learning, variational quantum algorithms and the evidence lower-bound encountered in classical variational autoencoders. So as to not deviate from our main subject matter, we discuss only those topics that provide the necessary background for the rest of the thesis. Chapter 2 is also our attempt

[1] requiring exponential time or space resources

at a self-contained introduction to variational bayesian inference, particularly applied to generative modeling. With the necessary background in place, we discuss the methods followed to setup the QeVAE. The setup comprises three main tasks: (i) constructing a hybrid quantum-classical neural network, (ii) modifying the network to learn both classical and quantum datasets, and (iii) obtaining results and contrasting them against current methods. These three tasks, in that order, will be taken up in Chapters 3. In addition, we describe the different kinds of datasets we use, namely, the MNIST database, measurement distributions from product states, haar random states, quantum circuit states and quantum-kicked rotor states. In Chapter 4, we present and benchmark our results for classical distributions, and for distributions generated by quantum-mechanical systems. Our implementation uses the Quantum Information Science Kit (Qiskit)[17] and Pytorch libraries for calculations. We provide a detailed guide to implementing the ideas presented in Qiskit at the end of Chapter 4 along with a working-code implementation. We conclude with a few remarks on the limitations of our setup and directions for future work in chapter 5.

Chapter 2

Theory

Our first order of business is to review some important aspects of the theory underpinning machine learning and quantum algorithms. The elucidation of the information processing abilities of quantum systems is a rich subject, and we direct the reader to the following references for a more in-depth study [18, 19]. Instead, here we restrict ourselves to a description of those aspects that bear direct relevance to the thesis that follows.

2.1 Primer on machine learning

Machine Learning [20, 21] can be described as a mathematical model that learns the patterns and relations from available data without being explicitly programmed. A computer program (machine) is said to *learn* from experience $\mathscr{E}$, if a certain performance metric $\mathscr{P}$ improves with experience by repeatedly performing a set of tasks $\mathscr{T}$, i.e, $\mathscr{P}(\mathscr{T}) \propto \mathscr{E}$. In other words, its performance measured by $\mathscr{P}$ for task $\mathscr{T}$ improves with $\mathscr{E}$ [22]. As depicted in figure 2.1, machine learning algorithms are grouped into three types based on the type of experience $\mathscr{E}$. They can be broadly be described as:

- *Supervised learning* refers to the class of ML techniques that rely on inference modelling. Given training examples X and their corresponding labels Y from the joint distribution $P(X,Y)$, we intend to determine the probability of a label y given a new data-point $\tilde{x}$, i.e. $P(Y = y|X = \tilde{x})$. Ex: Classification, Regression.

- *Unsupervised learning* techniques examine at the structural aspects of a data set. The algorithm's objective is to discover hidden patterns in the dataset and learn the probability distribution $P(X)$ given multiple samples $x \in X$. Clustering, dimensionality reduction, and generative modeling are some prominent examples.

- *Reinforcement learning* methods do not explicitly require the dataset X nor its labels Y. An agent investigates potential actions given an available state space and attempts to learn the optimal strategy. Each action done and the state-space explored is assigned a score, and activities that result in a higher score are rewarded, while actions that result in a lower score are penalized.

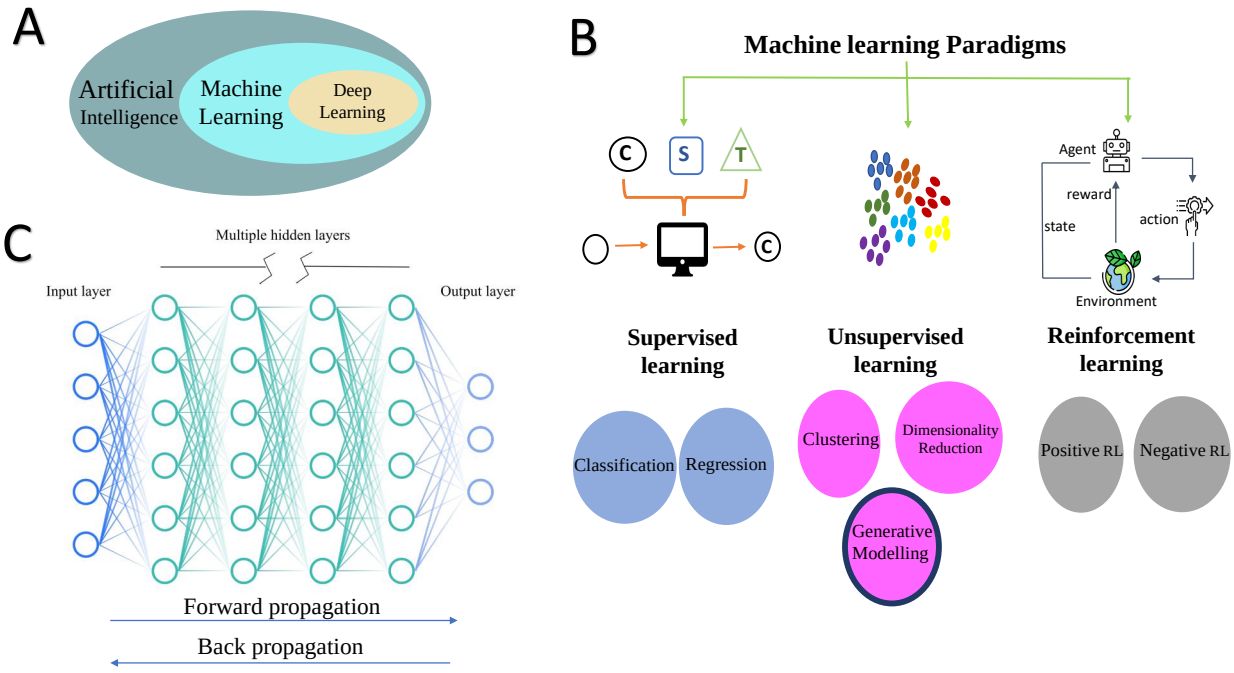

Figure 2.1: (A) The organization of Deep learing (DL), Machine learning (ML), and Artificial intelligence (AI). The field of AI encompasses ML, which encompasses DL (B) The different types of learning tasks in ML. In this thesis, our focus is on generative modeling. (C) A deep neural network with multiple layers of neurons stacked together. It has three major components: an input layer to which input data is fed, an output layer that outputs the generated signal, and multiple hidden layers that transform the input into high-level representation

Operationally, ML algorithms broadly perform two main tasks: (1) *Discriminative learning* where the focus is on learning the decision boundary between the classes within a dataset, i.e, learning $p(Y|X)$ and (2) *Generative learning* where the model aims to capture the actual distribution

of the classes in the dataset and learn the underlying distribution of data in each class i.e, learn $p(X,Y)$. The major focus of our work is on generative machine learning. For a detailed review on the above topics, we direct the interested reader to [21, 23]. A majority of modern generative algorithms utilize deep neural networks to learn from a large corpus of data and generate rich new examples which may consist of text, image, audio, or combination of them. These algorithms include Variational Auto-Encoders [24], Normalizing Flows [25], Generative Adversarial Networks (GANs) [26], and Diffusion models [11]. In the following section, we briefly review deep neural networks and then discuss the limitations of current methods.

A deep learning model consists of multiple layers of representations of the raw data in increasing levels of abstraction. Such a structure allows the model to independently learn the important features of a dataset and perform inference, whereas in simpler ML models, one is required to explicitly extract features before training. The simplest example of a deep learning model is a feed-forward neural network with dense connections as shown in figure 2.1C. Successive layers receive processed inputs based on a set of trainable parameters called *Weights*(**W**) and *Biases*(**b**). At each layer, a non-linear operation $\phi : R \rightarrow R$ with learnable parameters is applied to achieve high expressivity. A general operation performed at every layer of the neural network on data $\boldsymbol{x}$ can be given by:

$$\mathscr{L}_{n_0 \rightarrow n_1} = \phi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{2.1}$$

The parameters are trained using the gradients with respect to the output error using the chain rule of derivatives, called the backpropagation algorithm. The high-level representations learned are capable of distinguishing various patterns and suppressing noise within the raw data. In this work, we will focus on deep neural networks with dense connections; for further reading for other types of networks, we suggest [21]. Such neural networks are typically trained by minimizing a pre-defined loss function. We measure the generalization performance, a measure of performance on unseen data, of the learned map on new examples by calculating the loss for a different set of examples, often called a testing set.

Notably, many prominent ML and DL algorithms have had a deep-connection with physical principles, or ideas based on first-principles. With the objective of learning the boundary between different classes, some techniques in discriminative learning can be mapped to solving an error-minimization problem or can be viewed as a type of statistical physics model [27, 28]. Support vector machines can be thought of as a type of potential function that seeks to minimize the energy

between two classes and identifies a hyperplane that separates two regions of space. State-of-the-art generative models like GPT-3 and DALL-E rely on physical principles like diffusion and variational principles [29]. Elucidating how complex black-box models function is a job that Physicists have taken as a challenge and have had considerable success.

Nonetheless, current ML methods lack in several aspects and the field remains open for novel contributions. While a lot of work has focused on improving and building the best ML algorithms using neural networks, there are still tasks where a processing-speedup or accuracy-advantage is desirable. For instance, conventional sampling techniques in machine learning are often intractable (Ex: computing partition functions) or very expensive/time-consuming (Ex: markov-chain-monte-carlo calculations). Moreover, classical machine learning techniques require exponential number of parameters to learn the probability distributions produced by quantum-mechanical systems [30, 31]. Recent research has indicated that generative machine learning is the field where an advantage can be seen by using principles from quantum algorithms [32]. Variational bayesian inference (VBI) provides a method to learn the joint probability distribution $p(X,Y)$ and this thesis is dedicated to the specific problem of understanding how quantum models can assist in VBI. In the following sections, we introduce the basic ideas in quantum computation, methods in quantum machine learning, and those algorithms suitable for current-era quantum devices. We conclude this chapter by introducing the variational bayesian inference technique and how quantum models can potentially offer an advantage.

2.2 Primer on quantum algorithms

Quantum computation can be realized through many different frameworks and the two most popular ones are (1)gate-based and (2)annealing-based approaches. The former operates by applying a sequence of *quantum gates* to a set of *qubits* initialized in a known state. The later is a method of solving optimization problems by encoding the problem into the energy levels of a physical system and then adiabatically evolving the system towards the global minimum of the energy landscape. Our work focusses primarily on the gate-based model of computation. The basic primitives in this model of computation include the notion of *qubits*, *gates*, and *measurements*. The following is a self-contained primer on the above alluded concepts, including variational quantum circuits and quantum machine learning. We refer the interested reader to [18, 33, 34] for a more in-depth review of same topics.

2.2.1 Quantum computation

Classical computation and information processing is through the notion of bits, essentially restricting *packets* of information to be binary: either a *yes*, or a *no*. Mathematically, such a bit is a binary digit that takes one of two values: 0 or 1. Quantum information introduces the notion of *qubits* (quantum bits), allowing a *packet* of information to be both a 0 or 1. More formally, a qubit is a normalized two-dimensional complex vector, called a *wavefunction*, ($|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$), such that $|\psi|^2 = |\alpha|^2 + |\beta|^2 = 1$ where $\{|0\rangle, |1\rangle\}$ are any orthonormal basis in two-dimensions. Conservation of probability ($\sum_i |\psi_i(t)|^2 = 1$) ensures that a closed quantum system undergoes unitary time evolution. A pure quantum state made up of of n qubits can then be expressed through a tensor product as:

$$|\psi\rangle = \otimes_{i=0}^{n-1}(\alpha_i|0\rangle + \beta_i|1\rangle), \tag{2.2}$$

and occupies a Hilbert space that has dimension 2^n. Qubits exhibit two properties that classical bits lack, namely, *superposition* and *entanglement*. The former allows a quantum state to exist in a linear combination of orthogonal states, allowing a quantum state/wavefunction to encode much more information than a classical bit. The later describes a form of correlation between many quantum systems that is strictly non-classical and stronger than any known correlation of classical systems. Such entangled states cannot be decomposed as a product over the component quantum systems. Simple examples of entangled states include the the bell state ($|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$), and the GHZ state ($|\psi\rangle = (|000\rangle + |111\rangle)/\sqrt{(2)}$). Apart from pure quantum states, a classical mixture of pure quantum states, called *mixed quantum states* can be represented by density matrices such as:

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j| \tag{2.3}$$

where p_j is the probability of choosing a state $|\psi_j\rangle$ from a mixture of states such that $Tr[\rho] = 1$. Lastly, measuring a quantum state with a Hermitian operator $\hat{O}$ equates to $\langle\psi|\hat{O}|\psi\rangle$ or Tr$[\hat{O}\rho]$.

The Bloch sphere is a useful geometrical representation of a two-level quantum system, such as a qubit, representing a unit sphere with each point on the surface corresponding to a pure quantum state. The north and south poles of the sphere correspond to the basis states $|0\rangle$ and $|1\rangle$, and any other pure state can be written as a superposition of these states, with some amplitude and phase. For instance, the state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ lies on the equator of the sphere at an azimuthal angle

of 0. In addition, operations on the qubit can be visualized as rotations to the Bloch vector. For example, Hadamard gate transforms a qubit in state $|0\rangle$ to a superposition state $(|0\rangle + |1\rangle)/\sqrt{2}$). Such unitary operations are called *gates*. The U_3 gate is a single-qubit gate with three tunable parameters (θ, ϕ, λ) that specifies the position of a qubit on a Bloch sphere. The U_3 gate is a generalization of the R_x, R_y and R_z gates (discussed under section2.2.3) and can be used to create any single-qubit unitary operation.

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\phi}\sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)}\cos(\frac{\theta}{2}) \end{pmatrix} \tag{2.4}$$

A sequence of such gates or unitary operations with tunable parameters can be applied to a qubit instantiated in $|0\rangle$ to reach a desired quantum state pre-measurement. Having looked at the necessary concepts, we now move on to learn about how parameterized quantum gates are amenable to machine learning methods.

2.2.2 Quantum machine learning

ML algorithms are known to be eloquent at finding atypical patterns. Given that quantum processes naturally generate complex patterns, such systems might handle operations and perform better than conventional methods. *Quantum machine learning* (QML) is a new discipline that addresses the use of quantum computers for finding patterns in data. Formed by the union of quantum computation and machine learning, they include both quantum algorithms that process classical data as well as classical algorithms that process data from a quantum system. We are specifically interested in the question: How can the power of quantum computation solve problems in machine learning in lesser time, or with a better accuracy? This pursuit is called a *quantum speedup*. *Quantum advantage* refers to a calculation employing a quantum device that cannot be accomplished classically with a realistic amount of resources [33].

Multiple QML algorithms have been proposed that are provably advantageous. The key components of these algorithms involve subroutines such as the quantum phase estimation [35], quantum amplitude estimation [36], or Grover's search algorithm [6]. Majority of ML problems utilize linear-algebraic subroutines like Fourier transforms, solving a system of linear equations, or an eigen-decomposition of matrices to process and infer from data. Quantum basic linear algebra (qBLAS) subroutines like the HHL algorithm, quantum PCA, and quantum fourier transform

exhibit exponential speedup over the best-known classical algorithms for such tasks. However, these algorithms require millions of qubits with long coherence times and very low error levels. Since current noisy-intermediate scale quantum (NISQ) devices suffer from (1)limited connectivity, (2)qubit numbers upto 100 and are (3)prone to large errors and limited coherence times, the above mentioned algorithms cannot be realized physically [37]. Moreover, these algorithms face challenges with input-loading, output-measurement, and lack of bench-marking [34]. This raises the question of how we might make the most of the limited resources at our disposal to accomplish tasks that are classically challenging.

Quantum models cannot learn and generalize quantum data using just quantum processors alone since they are still rather tiny and noisy. Thus, existing NISQ methods must use a hybrid quantum-classical setup to leverage the potential of quantum computers. This is achieved by integrating conventional and quantum computing resources to train a parameterized model based on the problem at hand. We can thus partition current QML algorithms into two types: those that require fault-tolerant quantum computers and those that work with first-generation quantum hardware. The later often use variational quantum circuits, which are shallow, parameterized quantum circuits. These algorithms perform a portion of the calculation that is classically challenging on the quantum device and the remainder on a classical computer. Since optimization is performed by varying the parameters and updating the variables to minimize a loss function, these algorithms are known as *variational quantum algorithms* (VQA). Moreover, a *quantum neural network* (QNN) also refers to a parameterized quantum-classical model that is optimized using a classical device and run on a quantum computer. In the following section, we discuss the building blocks of a VQA.

2.2.3 Variational Quantum Algorithms

Variational quantum algorithms (VQAs), as previously indicated, employ a conventional optimizer to train a parameterized quantum circuit and provide a solution to the constraints of NISQ technology. VQAs offer the best approach to achieve quantum advantage with NISQ technology, and they have already been proposed for applications in quantum chemistry, machine learning, and binary optimization problems. A VQA is made up of a number of modular parts that may be easily merged, expanded, and enhanced as the hardware and algorithms advance. These elements are the following: (1) the objective function, which essentially encodes an optimization problem or a hermitian unitary as a cost function that is to be variationally minimized; (2) the parametrized

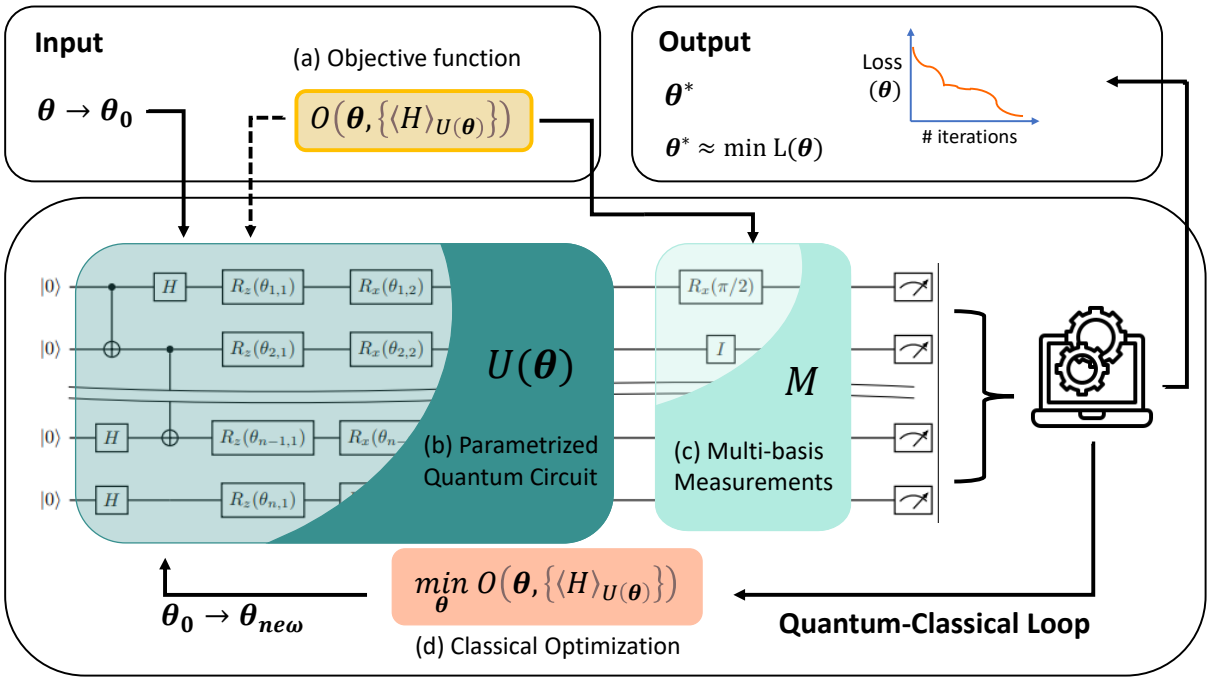

Figure 2.2: Structure of a variational quantum algorithm (VQA). (a) The input includes the the objective function and the set of variational parameters to be optimized. (b) Evaluation of the objective function involves computing the expectation value through (c) multi-basis measurements of an operator or the quasi-probabilities obtained from a quantum computer. (d) The value of the objective function is minimized by computing gradients with respect to the input or through gradient-free approaches. This encapsulates the classical optimization step. Finally, after certain number of iterations, a local minima of the objective and the corresponding variational parameters are obtained.

quantum circuit (PQC), which contains unitaries like the U_3 gate and quantum embedding layers that are manipulated during the training phase; (3) multi-basis measurements, which yield quasi-probabilities or average values of operators required to evaluate the cost function; and (4) a classical learner, that determines the best circuit parameters to minimize the cost function. For a more detailed review of VQAs and its applications we direct the reader to [33].

Objective function: The optimization problem is formulated as the minimization of a predefined objective/cost function : $\min_{\theta} \mathscr{O}(\theta, \{p(\theta)\})$. The collection of variational parameters $\{\boldsymbol{\theta}\}$ determine the value of the objective function $\mathscr{O}$ and the measurement results $p(x|\boldsymbol{\theta})$. In addition, certain objective functions aim to find the ground state energy and wavefunction by minimizing the expectation value of a Hamiltonian $\hat{H}$. They vary $\{\boldsymbol{\theta}\}$ such that $\langle \hat{H} \rangle_{U(\theta)} \equiv \langle 0|U^{\dagger}(\theta)\hat{H}U(\theta)|0\rangle$ is minimized according to the Rayleigh-reitz criterion [18]. It is important to note that the choice

of the objective function is critical to achieve the desired convergence. Global objective functions, such as finding the smallest positive eigenvalue, are prone to the vanishing gradient problem during optimization [38]. Following are some example of objective functions are:

(1) Pauli Strings: An operator from the full n-qubit Pauli group $P_n = (-i)^q P_{n-1} \otimes ... \otimes P_0$, where $q \in \mathbb{Z}_4$ and $P_i \in \{\sigma_I, \sigma_X, \sigma_Y, \sigma_Z\}$ are single-qubit Pauli matrices. The objective function is usually the expectation value of a manybody state with respect to the Pauli string ie $\langle\psi|P_n|\psi\rangle$ where

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma_I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.5}$$

(2) State fidelity: The state fidelity for density matrix input states ρ_1, ρ_2 is:

$$F(\rho_1, \rho_2) = Tr\left(\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right)^2 \tag{2.6}$$

If either of the states is pure then $F(\rho_1, \rho_2) = \langle\psi_1|\rho_2|\psi_1\rangle$ where ρ_2 is $|\psi_2\rangle\langle\psi_2|$

Parametrized quantum circuits: The quantum circuit that sets up the wave function to be optimized is known as the *parametrized quantum circuit* (PQC). It is a unitary operation $\hat{U}$ that depends on a set of free parameters $\boldsymbol{\theta} = (\theta_1, ...\theta_n)$, yielding $\hat{U}(\boldsymbol{\theta})$. The VQA can commence the search in an area of the parameter space that is closest to the optimum by selecting a good initial state, specified by $\{\boldsymbol{\theta}\}$. When the QML model is to be trained on a certain dataset, the PQC can further be broken down into two parts: (1) The Feature Map, $U_F(\boldsymbol{x}; \boldsymbol{\mu})$ that encodes classical data $\boldsymbol{x}$ onto a quantum circuit, embedding the data in a high-dimensional Hilbert space and (2) the Ansatz, $U_A(\boldsymbol{\nu})$ that contains arbitrary rotation and entanglement gates parametrized by $\boldsymbol{\nu}$. The PQC can thus be concisely represented as

$$U(\boldsymbol{x}, \boldsymbol{\theta})|\psi_0\rangle = U_A(\boldsymbol{\nu})U_F(x; \boldsymbol{\mu})|\psi_0\rangle \tag{2.7}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\nu}\}$

The choice of the Ansatz (German for 'educated guess') $U_A(\boldsymbol{\nu})$ is known to greatly affect the performance of a VQA. Most Ansatz contain arbitrary rotation and entangling gates operating between two or more qubits. The Rx, Ry and Rz gates are single-qubit rotation gates where each R_i

gate is a rotation around the i^{th}-axis of the Bloch-sphere by an angle (θ) (radians).

$$R_x(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \quad R_y(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \quad R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{2.8}$$

Moreover, the Ansatz governs the convergence speed, the expressibility and entanglement capacity of the desired output state $\psi(\boldsymbol{\theta})$ [39]. Although some problems, such as those in quantum chemistry, require the ansatz to have certain symmetries or a particular entanglement structure, such circuits might not be compatible with current quantum hardware. This is because deeper circuits tend to be more susceptible to errors, and only few types of native gates are executable on hardware. Consequently, there is a trade-off between choosing an ansatz suitable for a particular problem vs choosing it to be suitable to be executed on NISQ hardware efficiently. We will describe the feature map and ansatz used for our problems in the Methods chapter. Interested readers can refer to [40, 41, 42] for a detailed guide on choosing the right ansatz, and quantum embedding.

Measurement: The expectation value with respect to an operator ($\langle\hat{O}\rangle_{U_\theta}$) or the associated quasi-probabilities of a prepared quantum state ($|U_\theta|\psi\rangle|^2$), must be known to learn about the prepared quantum state. They can be estimated through multiple measurement shots and a statistical average of the measured eigenvalues. A straightforward approach is to estimate the eigenvalues by transforming the quantum state to the diagonal basis of the observable $\hat{O}$ before measurement. Alternatively, since NISQ-friendly approaches involve parameterized Pauli strings, the diagonal basis can be achieved by simple single-qubit rotations. Other measurement procedures involve estimation of state overlaps, SWAP tests, or classical shadows [43].

Parameter optimization: The process of optimizing the parameters of a PQC is similar to optimizing a multivariate functional and one can leverage the extensive set of methods developed for classical optimization. Nevertheless, not all optimization algorithms work well with PQCs and must satisfy the following three criteria: (1) A shorter coherence time in NISQ-devices precludes the execution of deep analytical gradient circuits. (2) Since the measurement protocol is costly and error-prone, the number of measurements evaluations must be minimal. (3) Calibration errors necessitate that the optimizer should be robust to noisy data and the number of measurement shots. Furthermore, classical optimization is known to be intrinsically hard and that the training landscape might contain a large number of far from optimum persistent local minima [44]. This restricts the class of optimization methods and based on the method of evaluation, they are of three kinds:

- Gradient-based approaches: The cost function $\mathscr{O}(\boldsymbol{\phi})$ can be minimized by iteratively evaluating the change of the function value with respect to an infinitesimal change of its parameters $\boldsymbol{\phi} = (\phi_1, ..., \phi_M)$. With the knowledge of the gradient, the local minima of the objective function can be iteratively computed: starting from an initial vector $\boldsymbol{\phi}^{(0)}$ and iteratively updating $\boldsymbol{\phi}^{(t)}$ over many steps t. The rule to update $\boldsymbol{\phi}_i$ is:

$$\boldsymbol{\phi}_i^{(t+1)} = \boldsymbol{\phi}_i^{(t)} - l\partial_i f(\boldsymbol{\phi}) \tag{2.9}$$

  or $\boldsymbol{\phi}^{(t+1)} = \boldsymbol{\phi}_i^{(t)} - l\nabla f(\boldsymbol{\phi})$, where l is the *learning rate* and

$$\partial \equiv \frac{\partial}{\partial \phi_i}; \qquad \nabla \equiv (\partial_i, ..., \partial_M) \tag{2.10}$$

  is the partial derivative with respect to the parameter ϕ_i and the gradient vector, respectively. The gradient can be computed in multiple ways for a quantum circuit and the most relevant of them are detailed in [45]. The most popular ones are: (1) Finite difference, (2) Parameter-shift rule, (3) Quantum natural gradient, (4) Quantum analytical gradient, (5) Stochastic gradient descent.

- Gradient-free approaches: These techniques do not rely on the gradient of the cost function. Instead, approaches such as evolutionary algorithms have been demonstrated to perform similar to state-of-the-art gradient-based methods. In addition, Reinforcement learning methods and surrogate model-based optimizations have also been used to optimize PQCs.

- Resource-aware optimizers: In recent years, optimizers have been designed to reduce parameters associated with running quantum circuits on hardware, such as the number of measurement shots or real hardware attributes. Circuit compilation methods aid in reducing the depth of the circuit to be optimized. Optimizers such as ROSALIN, SPSA, and QNSPSA have been created to be noise-resilient and require fewer number of measurements.

2.2.4 Hybrid Quantum-classical neural networks

The majority of success and focus in machine learning research is due to the power of artificial neural networks. Developing such machine learning models incorporating qubits, including the phenomena of superposition and entanglement, is an active field of research. Hybrid quantum-classical neural network is one of the NISQ-friendly approaches to construct a *quantum neural*

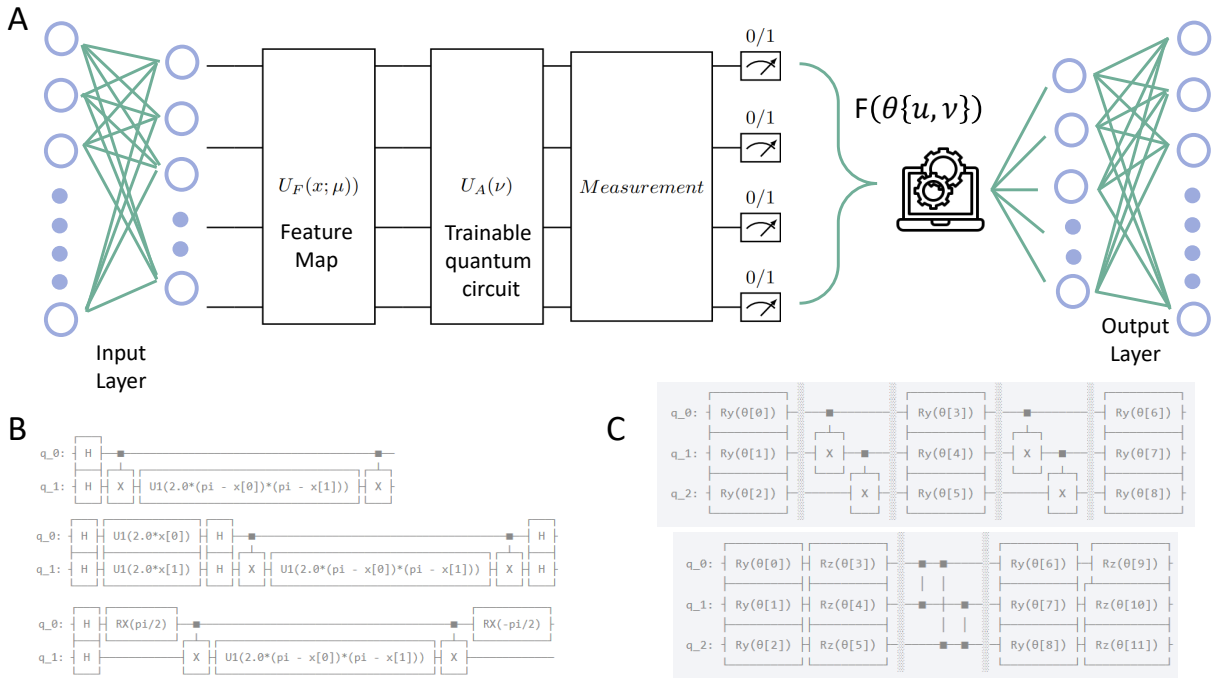

Figure 2.3: (A) Schematic of a Hybrid quantum-classical neural network. An instance of a hybrid network consisting of classical data processed by a feed-forward neural network. The processed inputs are embedded into the quantum circuit and unitary operations are performed. Measurement is like the activation function that yields outputs to be classified processed further. (B) Example feature map function $U_F(x;\mu)$ (C) Example of Trainable quantum circuits (Ansatz) $U_A(\nu)$

network, where one integrates the best of the classical and quantum resources. While classical feed-forward dense networks process parts of the computation, a quantum device handles the remainder. The combined parameters of the classical and quantum processing units can be tuned using appropriate optimizers and loss functions. More specifically, data can be processed by a neural network and embedded onto a quantum circuit, whose output can be utilized to infer the category of the data supplied. Different learners (optimization methods) can be utilized for the quantum circuit and the classical network. In figure 2.3, we represent a general structure of a hybrid quantum-classical neural network.

Now that we have reviewed the required background of ML theory and variational quantum algorithms, we move on to our generative model of interest, the variational autoencoder (VAE). In the next section, we derive the loss function for the VAE using Bayesian inference and with the help of mean-field theory, show how the loss function simplifies when the latent variables are set to be factorized Gaussian distributions.

2.3 Variational autoencoders

Variational autoencoders (VAE) [24, 46], are generative machine learning models that aim to implicitly learn the underlying distribution of the dataset $p(\boldsymbol{X}, Y)$. We assume that the distribution of observed dataset $p(\boldsymbol{X})$ can be represented by another distribution $p(\boldsymbol{Z})$, where $\boldsymbol{Z}$ is called the *hidden/latent variable*. We are interested in discovering the relationship between (X, Z), more specifically the generative process: $p(X|Z)$. We can depict this relationship through a graphical model as as shown in figure 2.4a. The edge from node Z to node X depicts the relationship between the two random variables through the conditional distribution $p(X|Z)$. From Bayes' Theorem, we know the general relationship between these random variables:

$$p(Z|X) = \frac{p(X|Z)p(Z)}{p(X)} \tag{2.11}$$

where $p(Z|X)$ is the posterior distribution, $p(X|Z)$ is the likelihood, $p(Z)$ is the prior probability distribution, and $p(X)$ is the marginal distribution. Presuming we know how to evaluate functions on the likelihood and the prior, generative learning requires us to compute functions on $p(Z|X)$. This is the problem of *posterior inference*. Several approaches exist to determine the posterior, including those from statistical physics, Laplace approximations, importance sampling, and perturbation theory [47, 48, 49, 50]. In variational inference formalism, we assume a certain form for the posterior through a known distribution $q_\phi(z|x)$ with tunable parameters ϕ and approximate it to the desired distribution $p(Z|X)$.

Variational Lower Bound for Mean-field Approximation

The joint probability of the graphical model $(Z \to X)$ can be written as $p(x,z) = p(x|z)p(z)$. The generative process entails that we draw samples $z_i \sim p(z)$ and $x_i \sim p(x|z)$. To perform inference, we use the bayes rule and find the posterior distribution by:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz} \tag{2.12}$$

Since the computation of $p(x)$ involves assessing all configurations of latent variables, the integral in the denominator (called 'evidence') is frequently intractable or takes exponential time to compute. As a result, the posterior distribution can be approximated with a family of distributions

$q_\phi(z|x) = \Pi_{j=1}^{J} q_{\phi_j}(z_j|x)$ where ϕ is a set of variational parameters and J is the latent-vector dimension. In the mean-field approximation, we assume that the latent variables can be partitioned so that each partition is independent of the others. We measure the departure of our parametric model $q_\phi(z|x)$ from the true posterior $p(z|x)$ through the reverse KL divergence which measures the amount of information required to 'distort' $p(z|x)$ into $q_\phi(z|x)$. This can be written as:

$$KL(q_\phi(z|x)||p(z|x)) = \sum_{z\in\mathbb{Z}} q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)} \tag{2.13}$$

On simplifying, we get:

$$\begin{aligned} KL(q_\phi(z|x)||p(z|x)) &= \left(\sum_{z\in\mathbb{Z}} q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z,x)}\right) + \left(\log p(x) \sum_{z\in\mathbb{Z}} q_\phi(z|x)\right) \\ &= \log p(x) + \left(\sum_{z\in\mathbb{Z}} q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z,x)}\right) \\ &= \log p(x) + \mathbb{E}_{q_\phi(z|x)}\left(\log \frac{q_\phi(z|x)}{p(z,x)}\right) \end{aligned}$$

Minimizing the LHS is equivalent to minimizing the second term in the RHS since the fist term is independent of the parameter set $\{\phi\}$. The later is equivalent to maximizing the negation of :

$$max_\phi \mathscr{L} = -\mathbb{E}_{q_\phi(z|x)}\left(\log \frac{q_\phi(z|x)}{p(z,x)}\right) \tag{2.14}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p(x|z) + \log p(z)\right] \tag{2.15}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log p(x|z) + \log \frac{p(z)}{q_\phi(z|x)}\right] \tag{2.16}$$

In ML literature, $\mathscr{L}$ is known as the *variational lower bound* or as the *(negative) variational free energy* is statistical physics literature. If we can evaluate $p(x|z), p(z), q_\phi(z|x)$, $\mathscr{L}$ is computationally tractable. We may rearrange the terms even more to produce an intuitive formula:

$$\mathscr{L} = \mathbb{E}_{q_\phi(z|x)}\left[\log p(x|z) + \log \frac{p(z)}{q_\phi(z|x)}\right] \tag{2.17}$$

$$= \mathbb{E}_{q_\phi(z|x)}(\log p(x|z)) + \sum_{Q} q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} \tag{2.18}$$

$$= \mathbb{E}_{q_\phi(z|x)}(\log p(x|z)) - KL(q_\phi(z|x)||p(z)) \tag{2.19}$$

Substituting $\mathscr{L}$ back into Eq.2.13, we have:

$$KL(q||p) = \log p(x) - \mathscr{L}$$
$$\log p(x) = KL(q||p) + \mathscr{L} \tag{2.20}$$

In equation 2.20, we note that since $KL(q||p) \geq 0$, $\log(p(x))$ must be greater than $\mathscr{L}$. Consequently, $\mathscr{L}$ is a lower bound for $\log(p(x))$ and is therefore called the *evidence lower bound* (ELBO):

$$\mathscr{L} = \log p(x) - KL(q_\phi(z|x)||p(z|x)) = \mathbb{E}_q[\log p(x|z)] - KL(q_\phi(z|x)||p(z)) \tag{2.21}$$

We have thus circumvented the problem of computing the posterior distribution by maximizing

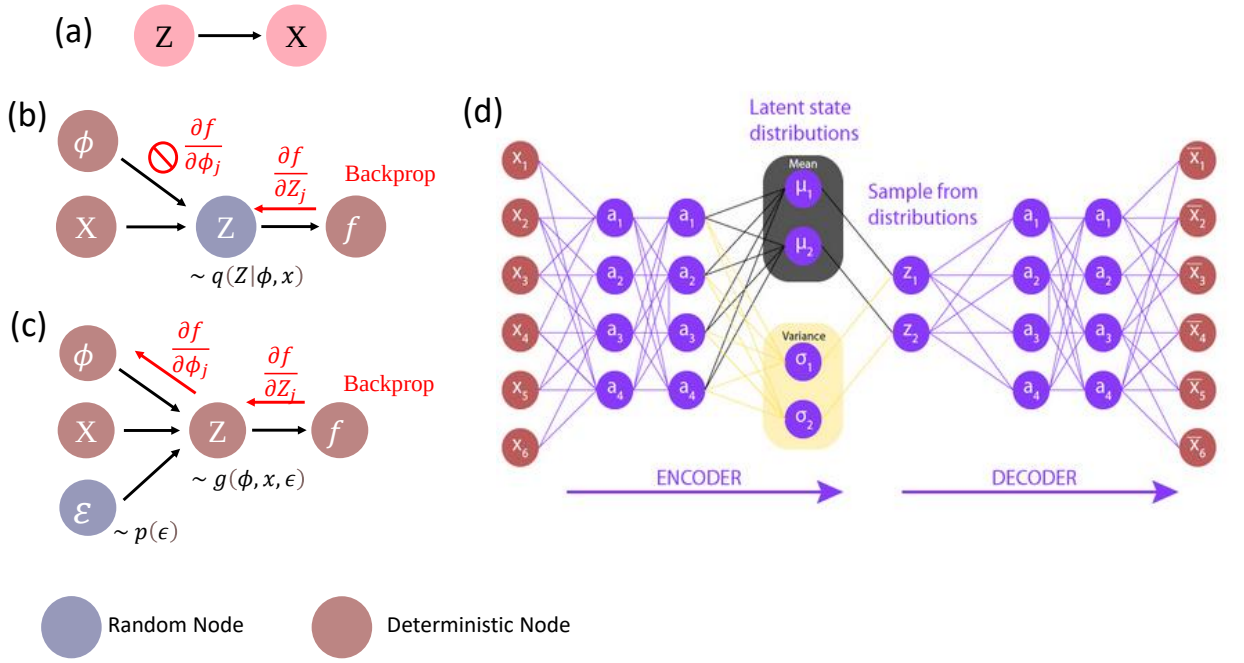

Figure 2.4: (a) Graphical model depicting the relationship between random variables (Z, X) (b,c) The reparameterization trick allows backpropagation through the latent vector Z by considering a random state ε from U(0,1) (d)Architecture of a variational autoencoder with the inference and generative neural-networks

the ELBO. Such a formulation is amenable to deep learning methods. Specifically, it entails constructing a neural network where each data-point $x \in X$ is itself the (input, output) pair, and the

network learns to reconstruct the dataset by mapping it to a lower-dimensional manifold[1] Z. The posterior $q_\theta(z|x)$ can be represented by a neural network (called 'encoder') that takes as input data x and outputs parameters z. The likelihood $p(x|z)$ can be specified, again, by a neural network (called the 'decoder') that takes latent variables z and outputs x, and the parameters to the data distribution $p_\phi(x|z)$. The encoder and decoder networks have parameters (θ) and (ϕ) respectively. Optimizing the encoder and decoder networks to maximize the ELBO solves the problem. Since this network reconstructs itself by variationally encoding X onto a manifold Z, we call this a *Variational Autoencoder* (figure 2.4(d)). We can now restate the ELBO and include the inference and generative network parameters as:

$$\boxed{ELBO_i(\theta,\phi) = E_{q_\theta(z|x_i)}[\log(p_\phi(x_i|z))] - KL(q_\theta(z|x_i)||p(z))} \tag{2.22}$$

This evidence lower bound gives the negative of the loss function for variational autoencoders. In the next section, we show how restricting the prior and parameterized distributions to a normal distribution can provide us with a closed-form expression for the loss function.

VAE loss-function with Gaussian latent variables: Suppose the prior and the approximate posterior distributions are considered to be multivariate Gaussian distributions. Then, we have:

$$p(z) \to \frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right) \quad \text{and} \quad q_\theta(z|x_i) \to \frac{1}{\sqrt{2\pi\sigma_q^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \tag{2.23}$$

the KL divergence term modifies into:

$$-KL(q_\theta(z|x_i)||p(z)) = \sum_Q \frac{1}{\sqrt{2\pi\sigma_q^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \times \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_q^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}\right)$$

$$= \sum_Q \frac{1}{\sqrt{2\pi\sigma_q^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \times \left(-\frac{1}{2}\log 2\pi - \log\sigma_p - \frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2}\log 2\pi + \log\sigma_q + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right)$$

[1] a higher dimension can also be used to build a model that mitigates noise

The above equation simplifies to:

$$
\begin{aligned}
-KL(q_\theta(z|x_i)||p(z)) &= \sum_Q \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \times \left\{\log\frac{\sigma_q}{\sigma_p} - \frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right\} \\
&= \mathbb{E}_q\left\{\log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right\} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left\{(x-\mu_p)^2\right\} + \frac{1}{2\sigma_q^2}\mathbb{E}_q\left\{(x-\mu_q)^2\right\} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left\{[(x+\mu_q)-(\mu_p-\mu_p)]^2\right\} + \frac{\sigma_q^2}{2\sigma_q^2}
\end{aligned}
$$

Expanding the terms within the expectation term, we can simplify the RHS as:

$$
\begin{aligned}
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{1}{2} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left\{(x-\mu_q)^2 + 2(x-\mu_q)(\mu_q-\mu_p) + (\mu_q-\mu_p)^2\right\} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{1}{2} - \frac{1}{2\sigma_p^2}\left[\mathbb{E}_q\left\{(x-\mu_q^2)\right\} + 2\mathbb{E}_q\left\{(x-\mu_q)(x-\mu_p)\right\} + \mathbb{E}_q\left\{(\mu_p-\mu_q)^2\right\}\right] \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{1}{2} - \frac{1}{2\sigma_p^2}\left[\sigma_q^2 + 2*0*(\mu_q-\mu_p) + (\mu_q-\mu_p)^2\right] \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{\sigma_q^2 + (\mu_q-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2}
\end{aligned}
$$

When we set the prior distribution to a standard Gaussian, we set $\sigma_p = 1$ and μ_p=0, yielding:

$$
-KL(q_\theta(z|x_i)||p(z)) = \frac{1}{2}\left[1 + \log(\sigma_q^2) - \sigma_q^2 - \mu_q^2\right] \tag{2.24}
$$

Consequently, the ELBO due to a set of data points x_i is given by:

$$
\frac{1}{2}\left[1 + \log(\sigma_q^2) - \sigma_q^2 - \mu_q^2\right] + \mathbb{E}_{q_\theta(z|x_i)}\left[log p_\phi(x_i|z)\right] \tag{2.25}
$$

where σ_q^2 and μ_q are parameters of the approximate distribution, $q_\theta(z|x)$. The loss function is the negative of the previous equation and is given by:

$$
\boxed{\mathbb{L} = -\sum_{j=1}^{J}\frac{1}{2}\left[1 + \log(\sigma_q^2) - \sigma_q^2 - \mu_q^2\right] - \frac{1}{L}\mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]} \tag{2.26}
$$

where J is the latent vector dimension, and L is the batch-size.

Reparameterization trick: Variational Autoencoders sample a random vector z from the parametric model $q_\phi(z|x)$, representing the true posterior. We need to backpropogate via the random sampling step to optimize the encoder and decoder neural networks, which is an issue since gradients cannot propagate through random nodes (as demonstrated in figure 2.4(b,c). To get around this, we employ the reparameterization method. We propose a new parameter ε, which allows us to reparameterize z so that the backpropagation algorithm can flow through the deterministic nodes. Specifically, we use $\varepsilon \sim N(0,1)$ sampled from a Normal distribution in (0,1) and set the 'sampled' $z \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma} \otimes \varepsilon)$ where $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ are learnt during training.

2.3.1 IBM Hardware

IBM Quantum provides cutting-edge superconducting quantum computers that are incorporated into classical computing workflows. There are around 20 quantum systems offered through IBM Cloud, some of which are open to the public and others of which are reserved for premium access. The IBM Quantum devices are made up of many qubit and LC oscillator components, each with a specific frequency. The qubits are built with Josephson junctions, which are nonlinear electrical devices capable of exhibiting quantum behavior at low temperatures. The linear resonant circuits, or LC oscillators, link to the qubits and serve as readout devices. IBM's superconducting qubits offer several advantages over other systems. These include: (1)high coherence times: preserving the quantum state for longer periods without being disturbed by noise. (2) Scalability, (3) Rich implementation of set of single-qubit and two-qubit gates, and (4) Qiskit Runtime, an open-source cloud-based platform that allows users to run quantum programs faster and more efficiently. These merits motivate us to test our proposed quantum algorithms on IBM hardware and verify our results. Nonetheless, these platforms suffer from issues of noise and specific hardware constraints. For instance, they require (1) extremely low temperatures (around 15 mK) to function, which poses challenges for scalability and maintenance. They (2) suffer from crosstalk, wherein unwanted interaction between neighboring qubits introduces errors during computations. They have (3) limited connectivity, and have (4) finite gate fidelities. Thus they do not perform perfectly every time. This may limits our ability to achieve fault tolerance or error correction.

2.4 Previous work and our contribution

Quantum-enhanced machine learning is an area of active interest at the confluence of machine learning, quantum computing and engineering research. Many previous works have looked at incorporating quantum correlations and building generative models [51]. While a majority of such works on generative QML focus on quantum Generative Adversarial Networks (QGANs) and Quantum Circuit Born Machines (QCBMs), very few focus on the quantizing the classical variational autoencoder. In particular, previous works incorporating the ELBO framework was focused mainly for annealing based computation [52]. However, only recently works based on the gate-based model of computation have been realized. A recent work focused on improving the latent space representation of classical VAEs through parameterized quantum circuits (PQCs) [53]. Another work considered a fully quantum model where both the encoder and decoder are PQCs with continuous Gaussian latent space variables formed by the expectation values from the encoder [54]. The later work focused on determining molecular properties on graphs, where no advantage in terms of accuracy or speed was reported. Our work focuses on incorporating the merits of both quantum and classical models in a hybrid fashion and on both quantum and classical datasets. We look at both discrete and continuous latent spaces and highlight the changes in performance in both cases. In the next section, we detail our motivation on building our QeVAE and how it can be used to solve a problem intrinsic to the ELBO framework that is still to be addressed in literature.

Why the Vanilla Variational Autoencoder fails and how can quantum correlations help? In traditional VAEs, the true posterior is estimated using diagonal Gaussian latent variables. The reason factorized Gaussian distributions are used is because they are (a) computationally cheap to compute and differentiate the posterior (a non-diagonal covariance matrix would require $O(n^2)$ parameters, whereas the current arrangement requires just $O(n)$), and (b) straightforward to sample at each mini-batch. Nevertheless, completely factorized diagonal Gaussian distributions cannot mimic all distributions. The fact that they are entirely factorized, in particular, restricts their capacity to simulate the genuine posterior. To this effect, ML practitioners have advocated building auto-regressive models where consecutive latent space nodes share some dependence. However, this still limits the expressive power of the model. In theory, if one can more precisely approximate the genuine posterior, the generator network should be able to train more easily, improving the overall output. So, how can we use a more complicated distribution? Motivated by the fact that quantum systems can produce complex distributions, we hypothesize that a parameterized quan-

tum circuit (PQC) as our posterior ansatz *can* provide a better result. In other words, our primary motivation for the study is: Can an ansatz or $q_\phi(z|x)$ given by a variational quantum circuit approximate the true posterior better than a purely classical model? Another related question, we would like to ask is if such quantum models can learn distributions which classical VAEs cannot. If so, what are those distributions? What is the performance improvement? In the next section, we provide a more elaborate description of the setup that will assist the reader in setting up a quantum neural network within the variational autoencoder framework.

Chapter 3

Methods

This chapter describes the theoretical and numerical methods used in this thesis. The first section describes how to construct a quantum-enhanced variational autoencoder using parameterized quantum circuits, why they are required and how they are beneficial. The second section describes how we construct the Quantum-enhanced Variational Autoencoder (QeVAE) to learn a classical distribution of pixels present in the MNIST database. We derive the loss function for the QeVAE in the variational Bayesian approach and show how discrete latent variables modify the final loss function. In the next section, we describe how we to construct QeVAEs to tackle problems that classical VAEs fail at. The task involves learning the measurement distributions of quantum-mechanical states. We also give a brief description of how these states and distributions are obtained. We conclude this chapter with some notes on practically implementing our algorithm on a quantum device.

3.1 Towards a Quantum Variational Autoencoder

The classical VAE can be quantized by substituting either the encoder, decoder or both by parameterized quantum circuits. The quantum-nature of these novel hybrid machine learning models endow two important advantages: (1) Efficient sampling (performed by a quantum circuit) and (2) larger latent space volume (through the exponentially growing Hilbert space of qubits). We exploit these features and numerically benchmark the performance of QeVAEs and contrast them against classical VAEs for two types of datasets: classical and quantum. For the classical dataset, we use

the Modified National Institute of Standards and Technology (MNIST) database and depict how the quantum-enhanced model performs relative to the classical VAEs. Since, our construction of a QeVAE uses a discrete latent space, we benchmark it with a classical VAE with discrete latent vectors instead of a factorized multivariate Gaussian distribution (which is used in the next section). In the following sections, we give a more detailed report to assist a reader aiming to reproduce our results.

3.2 Learning Classical distributions

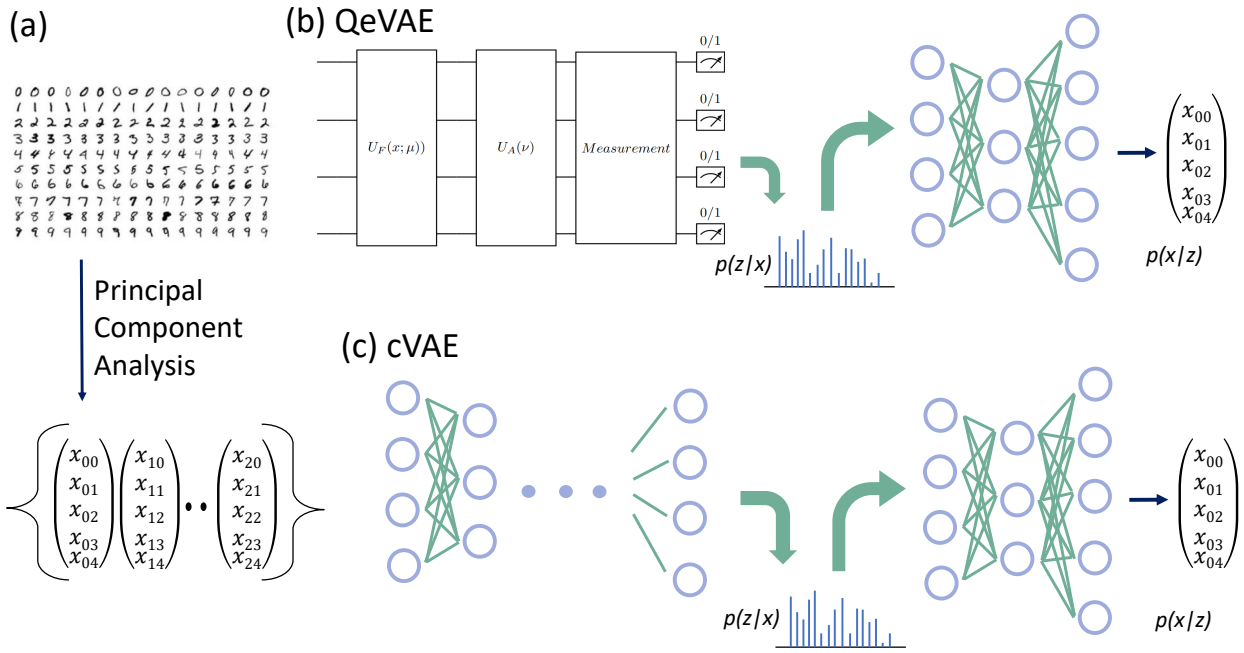

Figure 3.1: **QeVAE for learning classical distributions**: (a) MNIST images are preprocessed by reducing the dimension of each image through Principal Component Analysis (PCA). (b) The PCA vectors of each image are the features that are fed as input to the VQA in our QeVAE setup. The feature map loads the vector, and the ansatz performs a set of rotation and entangling operations, followed by a measurement scheme. Binary vectors from the latent space are sampled based on their probability of occurrence and passed through the decoder (a feed-forward neural network). The output is a weighted average of many latent vectors sampled and processed by the decoder. (c) A classical VAE with discrete latent variables.

We implement a QeVAE by substituting the encoder of a classical neural network $q_\phi(z|x)$ with a parameterized quantum circuit (PQC) as shown in figure 3.1. The PQC consists of three

components: the feature map, the ansatz and the measurement. The feature map is responsible for loading classical data onto the quantum circuit, the ansatz contains arbitrary rotation and entangling gates that determine the expressibility and entangling capacity of the circuit. Finally, a projective measurement yields a binary vector based on the Born rule. By varying the parameters of the PQC ($U(x;\phi)$), we optimize a variational family given by $q_\phi(z|x) = ||\psi(x;\phi)||^2 = ||U(x;\phi)|0\rangle||^2$. In our setup, we choose to work with a discrete latent space because of two main reasons: (1) The latent space dimension grows exponentially with the number of qubits ($|z| = O(2^n)$). This allows us to encode a large dataset into a small qubit system (For instance, 1024 unique elements can be easily encoded in a 10 qubit system). In addition, multi-basis measurement of the system (ie performing measurements of different positive-operator valued measures (POVMs) on the same qubit) can increase the size further. (2) Quantum models intrinsically produce discrete distribution and any attempt at transforming this into a continuous distribution will reduce the amount of information stored. Using latent variables also prevents the use of the reparameterization rule, and we can write the loss-function as:

$$L = -\mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z)) + \beta KL(q_\phi(z|x)||p(z))$$

For discrete latent variables, we can choose our prior $p(z)$ to be a Uniform distribution over all the 2^n states available to simplify the second term as:

$$\begin{aligned}
KL(q_\phi(z|x)||p(z)) &= \sum_{z=1}^{2^n} q_\phi(z|x) \log\left(\frac{q_\phi(z|x)}{p(z)}\right) \\
&= \sum_{z=1}^{2^n} |\psi_z(x;\phi)|^2 \log\left(2^n |\psi_z(x;\phi)|^2\right) \\
&= \sum_{z=1}^{2^n} |\psi_z(x;\phi)|^2 \left(n\log 2 + 2\log|\psi_z(x;\phi)|\right) \\
&= n\log 2 + 2\sum_{z=1}^{2^n} |\psi_z(x;\phi)|^2 \log\left(|\psi_z(x;\phi)|\right)
\end{aligned}$$

Note that the second term in the RHS is actually the negative self-entropy of the measurement distribution produced by the PQC and can be written as $-\mathscr{H}(|\psi(x,\phi)|^2)$. Furthermore, the difference between this term and $n\log 2$ computes the difference between the state of maximum entropy (uniform) and the current state. To achieve a balance between the reconstruction loss and the KL-divergence loss, we use an additional hyper-parameter β. During training, β can be slowly changed from $0 \to 1$, resulting in focusing on reconstructing the input initially, and learning the

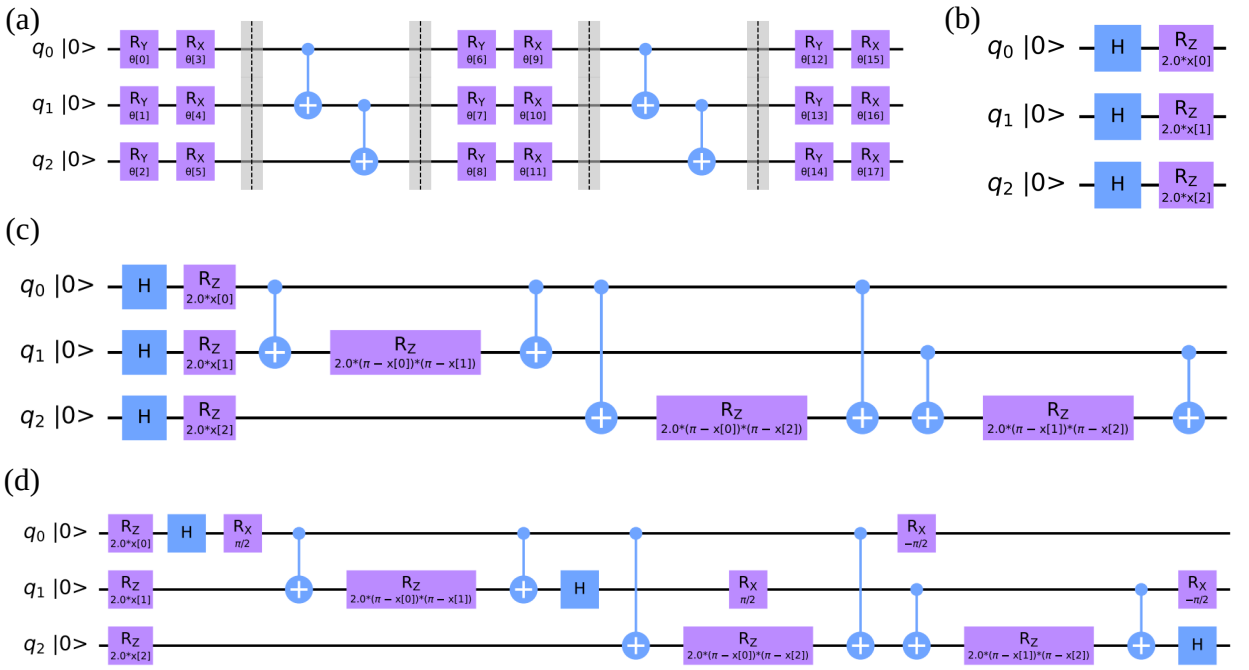

Figure 3.2: **Ansatz and Feature maps for learning classical distributions** (a) A two-local ansatz on three qubits with two repeating layers of *Rx* and *Ry* gates along with linear entanglement (b) A Pauli-Z feature map that embeds a three-dimensional vector. 'H' represents the Hadamard gate. (c) A Pauli-ZZ feature Map (d) A Pauli-(X, XY) feature map on a three qubit system.

latent space distribution only at the end. Thus, the loss function to be minimized becomes:

$$\boxed{L = -\mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z)) + \beta\left(n\log 2 - \mathscr{H}(|\psi(x,\phi)|^2)\right)} \tag{3.1}$$

Notes on implementation

While using the MNIST dataset, we note that each image is a binary matrix of size 28*x*28, flattening which, yields a 784-dimensional binary vector. The first-step of our algorithm is to encode each data-point and forward propagate it through the PQC. To encode the 784-dimensional vector, one needs to resort to *amplitude-encoding* on at least $\log_2(784) \sim 10$ qubits. Since qiskit does not allow for backpropagation with amplitude encoding, we resort to a different encoding scheme using Pauli feature map depicted in figure 3.2. Our proposed preprocessing scheme is as follows: We consider a dataset of 1000 images of digits (6,9) [since they are the most difficult digits for a classical algorithm to distinguish] and partition it into 70% training set and 30% validation set

images. After mean-normalizing the training dataset, we perform a principal component analysis (PCA) on the training set and reduce the dimension of each image to n_Q, by choosing the first n_Q principal vectors (n_Q= number of qubits). To avoid data leakage, we apply the same PCA transformation to the validation set. Now, our training set is of size (n_Q, 700) and the validation set is of size (n_Q, 300). This completes the data-preprocessing step. We are now ready to train the algorithm.

Next, each image is passed through the encoder, namely, the feature-map which prepares a complex-valued vector $U_F(x;\nu)|\psi_0\rangle = |\psi(x;\nu)\rangle$, the Ansatz is randomly initialized with parameters in (-1,1) and applies the parameterized gates to yield $U_A(\mu)|\psi(x;\mu)\rangle = |\psi(x;\{\nu,\mu\})\rangle$. This state is measured in the computational basis to yield the distribution $|\psi(x;\{\mu,\nu\})|^2$. This distribution is used to compute the second term in equation 3.3. Instead of sampling a random binary vector from the latent space, and computing the expectation (first term in equation 3.3), we compute a closed form expectation by passing every latent-vector through the decoder and computing a weighted average using $|\psi(x;\{\phi,\theta\})|^2$. It is crucial to note that with discrete latent variables, the size of the latent dimension is bounded by 2^n where n is the number of qubits. Furthermore, instead of taking all the latent vectors, we choose to pass the top $\eta\%$ of the latent-vectors, arranged in descending order of their probability. The weighted average is the output obtained from the decoder. We model the distribution of output as a factorized Gaussian distribution with mean equal to the value of the corresponding entry in output vector and standard deviation 1. Repeating this for all the neurons in the output layer, we reduce the first term in Equation3.3 to:

$$
\begin{aligned}
-\mathbb{E}_{q_\phi(z|x)} \log\left(p_\theta(x|z)\right) &= -\sum_{i=1}^{L} q_\phi(z_i|x) \log\left(p_\theta(x|z_i)\right) \\
\text{We choose: } p_\theta(x|z_i) &= \Pi_{j=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x_{0j}-x_j(\theta))^2}{2}\right) \\
\text{Then: } -\mathbb{E}_{q_\phi(z|x)} \log\left(p_\theta(x|z)\right) &= -\sum_{i=1}^{L} q_\phi(z_i|x) \sum_{j=1}^{n} \left\{\log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{(x_{0j}-x_j(\theta))^2}{2}\right\} \\
&= \sum_{i=1}^{L}\sum_{j=1}^{n} q_\phi(z_i|x_j) \frac{(x_{0j}-x_j(\theta))^2}{2} + \sum_{z=1}^{L}\sum_{j=1}^{n} q_\phi(z_i|x_j) \log\sqrt{2\pi} \\
&= \sum_{i=1}^{L}\sum_{j=1}^{n} q_\phi(z_i|x_j) \frac{(x_{0j}-x_j(\theta))^2}{2} + \log\sqrt{2\pi} \sum_{z=1}^{L} q_\phi(z_i|x_j) \sum_{j=1}^{n} \mathbb{I} \\
&= \frac{1}{2} \sum_{i=1}^{L}\sum_{j=1}^{n} |\psi_z(x_i,\phi)|^2 (x_{0j}-x_j(\theta))^2 \qquad (3.2)
\end{aligned}
$$

Neglecting the constant terms, the loss function for each sample x^l reduces to:

$$\boxed{L_{\phi,\theta}(x^l) = \left(\sum_{i=1}^{L}\sum_{j=1}^{n} |\psi_{z_i}(x_j^l,\phi)|^2 (x_{0j}^l - x_j^l(\theta))^2\right) - \beta\mathscr{H}(|\psi(x,\phi)|^2)} \tag{3.3}$$

where L is the size of the latent dimension, n is the size of the input vector, x_{0j} is the entry in the j^{th}column of the input vector, β is the relative weight given to the KL-divergence term and $|\psi_z(x_i,\phi)|^2$ is the distribution produced by the PQC.

Now, we return to describe the feature map and the ansatz. To encode classical data x onto the quantum circuit, we use the Pauli Feature Map (figure 3.2(b,c,d)) - whose general form is given by:

$$U_{\Phi(\vec{x})} = \exp\left(i\sum_{S\subseteq[n]}\phi_S(\vec{x})\prod_{i\in S}P_i\right) \tag{3.4}$$

If the variable $P_i = Z$, then U denotes the Pauli Z-feature map. The index S describes connectivities between different qubits. Using this we can define both first-order (without entangling gates) and second-order (with entangling gates) Pauli Z-evolution circuits as seen in figure 3.2 and figure 2.3. We use a two-local ansatz with linear entanglement since it is particularly suitable for NISQ-devices.

Classical VAE with discrete latent variables

To compare the results from QeVAE, we setup a classical VAE with discrete latent variables where both the encoder and decoder contain feed-forward neuronal networks. Equation 2.25 is not valid here and we numerically compute the KL divergence with the variational posterior and the prior. The output of the encoder is discretized using the sigmoid activation function ($\phi(x) = 1/(1+e^{-x})$) and the joint probability distribution is computed. Next, all samples upto a threshold are passed through the decoder and a closed form average is computed similar to the QeVAE setup described earlier. The resulting loss function is similar to 3.3 but the KL-term changes:

$$L_{\theta,\phi}(x^l) = \left(\sum_{i=1}^{L}\sum_{j=1}^{n} |\psi_{z_i}(x_j^l,\phi)|^2 (x_j^l - x_{0j}^l)^2\right) + \beta\left(n\ln 2 - \mathscr{H}(q_\phi(z|x))\right) \tag{3.5}$$

Here, since the discrete latent variables can be modeled as independent Bernoulli variables (there is no correlation between different latent neurons), we can rewrite the entropy term as $\mathscr{H}(q_\phi(x|z)) = \sum_i \{p_i(\phi)\ln p_i(\phi) + (1-p_i(\phi))\log(1-p_i(\phi))\}$, where $p_i(\phi)$ is the probability of the variable z being equal to 1 (learnt by the encoder).

Using a gradient based learning algorithm for the classical neural network (ADAM, SGD) [55, 56] and gradient-free learners for the quantum-neural network (SPSA, QNSPSA) [57, 58], we train the hybrid model and the classical model to minimize the loss function given by equation 3.3 or equation 3.5 respectively. To prevent over-fitting on the training dataset, we employ the early-stopping criterion, wherein we halt the training process when the validation loss does not decrease for δ number of epochs where δ is a hyper-parameter, called the *patience-factor*. The trained model is reconstructs the original dataset and also does not over-fit. After training, the encoder (quantum processor) can be discarded and only the decoder (classical neural network) can be used as a sampler to generate novel samples. To compare the output distribution after training, we sample latent vectors and pass them through the decoder. After performing an inverse-PCA transform on the output vectors, one can visually see the original, the reconstructed, and sampled images from the generative model.

3.3 Learning Quantum distributions

The study of quantum states has been a topic of great interest among physicists due to many of its exotic properties and potential applications. However, describing general quantum states can be a challenging task as the number of parameters required to represent a many-body spin system scales exponentially with the number of qubits. Although classical generative learning methods have been used to learn the measurement distribution of general quantum states, they require an exponentially number of parameters and sometimes they cannot learn the distribution with high fidelity [30, 31]. Recent research based on the Probably Approximately Correct (PAC) framework has shown that there exist classes of probability distributions that can be efficiently learned with quantum resources, but not with purely classical approaches. Furthermore, it is known that simulability does bot guarantee learnability of certain class of states (Ex: Clifford circuits) [16, 59, 60]. However, the description of such a quantum learner requires the existence of a fault-tolerant quantum computer. The canonical generative learning model suitable for NISQ devices is the Quantum circuit Born machine (QCBM) that contains a quantum circuit with tunable layers

of rotation gates and entanglement gates. However, these are input-agnostic and solely aim at recreating a desired target distribution.

In this part of the thesis, we explore the question of whether near-term quantum learners, can exhibit an improved performance over classical learning algorithms, in a generative modeling problem. Our focus is on the task of reconstructing a state/its measurement distribution via an iterative learning process, which has applications in depth-circuit compressing, quantum metrology, and sensing. We also show that our algorithm leads to the Quantum circuit Born machine (QCBM) in a certain limit. The goal is learn a quantum circuit that generates the target distribution. Although classical learners cannot efficiently learn the distribution of certain quantum states, we postulate that quantum learners can reliably reproduce these distributions.

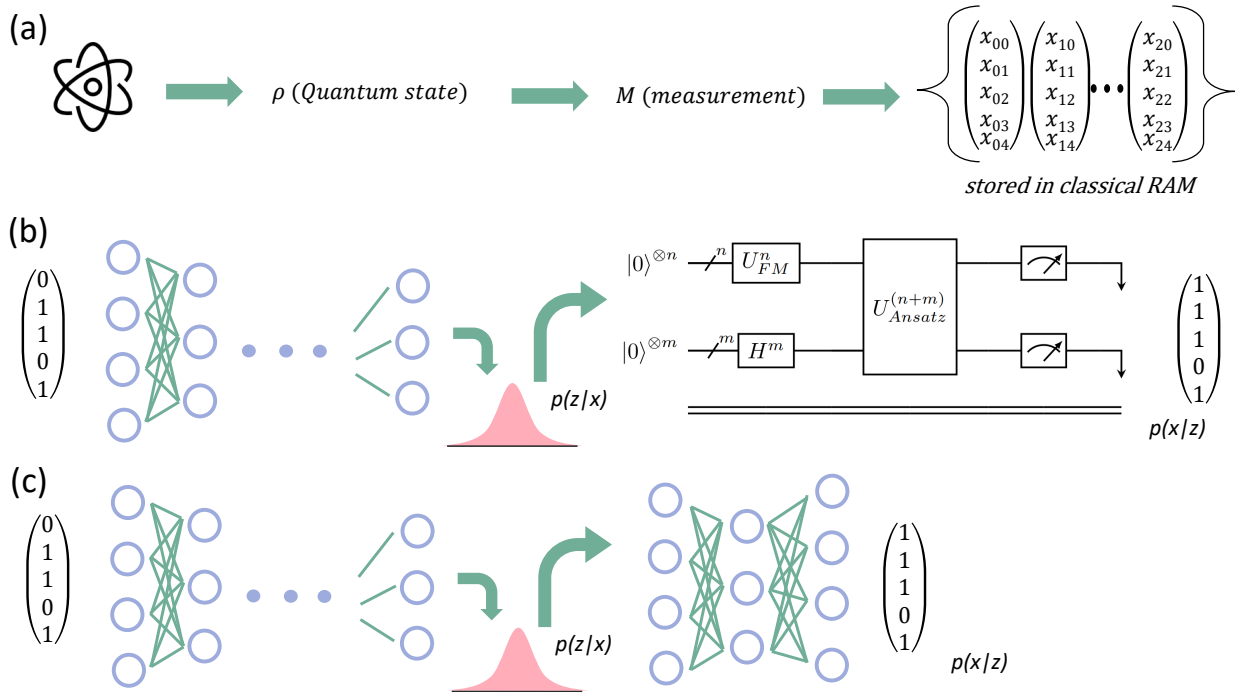

Figure 3.3: **QeVAE for learning quantum state distributions**: (a) Multiple copies of a quantum state ρ is obtained naturally from a quantum sensor and is measured through different POVMs in a lab. The measurement dataset is stored on a classical computer. (b) The QeVAE with a parameterized quantum circuit as the generative network and a classical feed-forward neural network as the inference network can be used to recrete the distribution. After training, the circuit can be used to generate the original distribution through any quantum computer and generate states amenable for downstream processing. (c) A classical VAE with continuous Gaussian latent variables that perform the same task.

To learn the distribution of unknown quantum states and realize it in different systems, we implement a QeVAE by replacing the decoder of a classical VAE with a PQC as shown in figure 3.3. The algorithm (agent) is only given access to the measurement distribution of various POVMs acting on an unknown quantum state. The agent learns to mimic the original distribution by repeatedly verifying the measurement distribution produced by the QeVAE with the original results and iteratively altering the ansatz. In such a scenario, the encoder acts as a classical post-processor whereas the decoder outputs probabilities associated with a learnt quantum state.

The metric we use to quantify the generated distribution is the fidelity between two discrete distributions. If ψ and ϕ are n-qubit states, then ϕ is a similar to ψ if the fidelity $F = \mathrm{Tr}(\sqrt{\psi^{1/2}\phi\psi^{1/2}}) > 1-\varepsilon$ for an $\varepsilon > 0$. The fidelity can be written in terms of the probability distributions over a measurement that maximally distinguishes the two states [61]. Thus given two random variables X, Y with probabilities $p = (p_1, p_2, ..p_n)$ and $q = (q_1, q_2, ..q_n)$, the fidelity of X and Y is defined to be the quantity:

$$F(X,Y) = \left(\sum_i \sqrt{p_i q_i}\right)^2 \tag{3.6}$$

where the measure $\sum_i \sqrt{p_i q_i}$ is known as the Bhattacharyya coefficient between the two distributions. We now briefly describe the four distinct kinds of states we use: Random product states, Haar random states, Quantum circuit states, and Quantum-kicked rotor states. Product states are easy to learn. It is known that conventional VAEs can learn to represent such quantum states but require exponential parameters (2^n where n in the number of qubits) to learn the distribution of Haar states and states postulated to be intractable on classical devices [15, 31].

3.3.1 Quantum measurement datasets

We benchmark the performance of QeVAE on several datasets. In the following sections, we detail the types of datasets used and how they are generated.

Random product states: Product states are classically easy to simulate and are empirically found to be *classically easy* to learn. We generate random product states by simulating quantum circuits with only single qubit gates with arbitrary angles of rotation, generated according to a random seed (As shown in figure 3.4(a)). The state prepared is on the form: $|\psi\rangle = \otimes_{i=0}^{n-1} \{\alpha_i|0\rangle + \beta_i|1\rangle\}$, where n is the number of qubits. Projective Z-basis measurements generat-

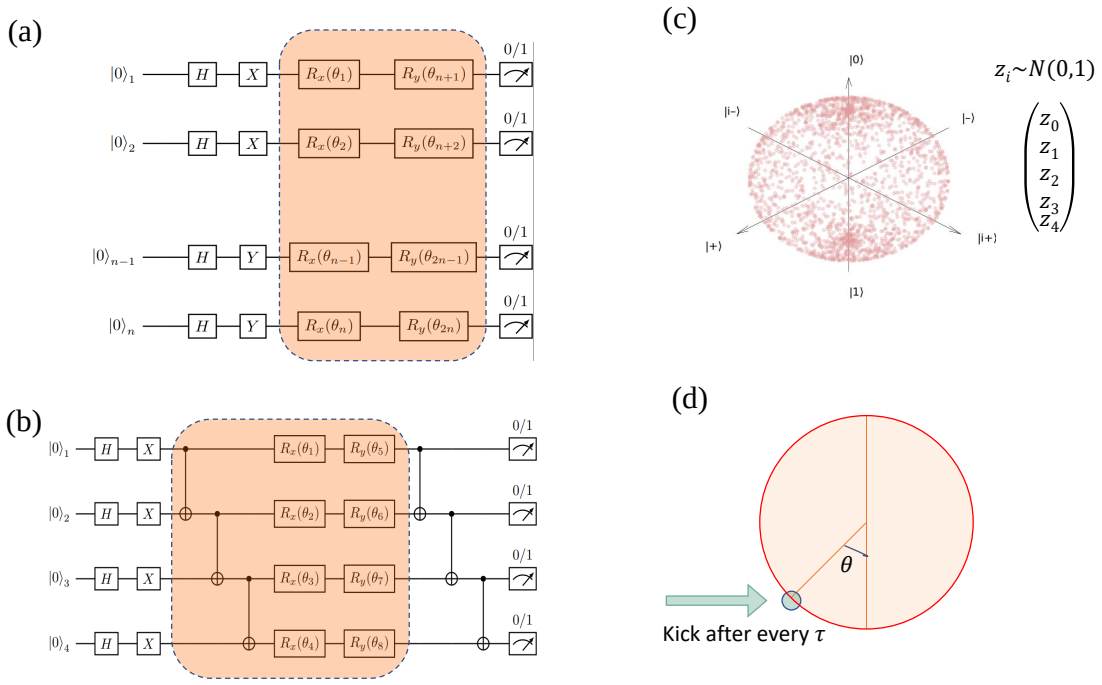

Figure 3.4: **Different types of measurement datasets**: (a) Product states obtained by a combination of arbitrary single qubit gates. The orange box represents repeatable layers of gates. (b) Quantum circuit states obtained from circuits with local (nearest-neighbor) entanglement. (c) Haar states obtained from a pure-quantum state by normalizing a 2^n-dimensional complex vector. (d) Quantum-kicked rotor states obtained by the time-evolution of an initial state $|0\rangle$.

ing the distribution.

Haar random states are quantum states that are uniformly distributed over the Hilbert space according to the Haar measure. Haar states represent *classically hard states* i.e, they require exponential number of parameters in the number of qubits to learn. These states can either be generated by first creating a Haar unitary U and then applying it on an initial state of dimension 2^n or by normalizing a complex-valued vector of dimension 2^n. We use the later method, where a complex-valued vector of $|\psi\rangle = \sum_{l=1}^{n}(c_{1l} + ic_{2l})|l\rangle$ is initialized with $|l\rangle$ corresponding to the orthonormal basis vector in the 2^n-dimensional Hilbert space, $\mathbb{C}^n$, and c_{1l}, c_{2l} are real numbers chosen independently from a standard Gaussian distribution. This vector is normalized to yield a quantum state by using the constraint: $\langle\psi|\psi\rangle = 1$. After normalization, the states are uniformly distributed on a unit hyper-sphere.

Random quantum circuit states are obtained from random quantum circuits with a pre-

defined entanglement structure and circuit depth, as shown in figure 3.3(b). These states are useful for circuit compression and circuit compilation.

Quantum kicked rotor states are obtained from the quantum kicked rotor (QKR), a quintessential model for quantum chaos in floquet systems, and are known to produce rich dynamical behavior under time evolution. The Hamiltonian of the system is given by:

$$\bar{H} = \frac{p^2}{2} + \kappa \cos x \sum_n \delta(t - n\tau) \tag{3.7}$$

and the Floquet operator is given by:

$$\hat{F} = \exp\left(-\frac{i}{\hbar_s} K \cos \hat{x}\right) \exp\left(-\frac{i}{2\hbar_s} \hat{p}^2\right) \tag{3.8}$$

where $\hbar_s$ is the scaled Planck's constant and K is the effective kicking strength, Under time evolution, the wavefunction exhibits classical diffusion in the weak-kicking (K) regime. Under the strong kicking regime, the system exhibits localization in momentum space, contrary to the chaotic behavior observed in its classical counterpart. For a more detailed review on the quantum kicked rotor, we refer the interested reader to [62]. To examine if the distribution of the wavefunction can be learnt by a generative model, we evolve an initial wavefunction initialized at $|\psi_p(0)\rangle = 0$ until 1000 kicks and then store the the probability distribution $|\psi_p|^2$. We train both classical and quantum models to reproduce these distributions. We are interested in discovering if a classical learner can learn the same distribution and how the number of parameters required scales with the size of the system.

3.3.2 Notes on implementation

Substituting the decoder of a classical VAE with a PQC yields a QeVAE suitable for learning the distribution of quantum states. Here we use continuous latent variables and the loss function is given by equation 2.26. We also create a QeVAE with a preprocessing layer before the quantum circuit, after sampling a latent vector. This provides two benefits: (a) Flexibility in choosing a latent size (b) Linearly transforming the latent vector of a different size to fit the input requirements of the quantum circuit.

The training pipeline is as follows: A dataset contains keys and values where keys represent

a bit-string of size n, the number of qubits and values represent the quasi-probability (number of times of occurrence) of measuring that bit-string in a measurement protocol. Such a dataset is expanded proportionate to its quasi-probability before training, where each bit-sting is repeated by its key value and permuted randomly. Sequentially, each bitstring is converted into a vector of size $(n, 1)$ and passed through the encoder, a classical neural network with input size n. The encoder terminates into two layers, representing the mean and log variance of a standard Gaussian. Using the reparameterization trick, a vector z is sampled from the latent space and passed to the preprocessor layer. In the absence of the preprocessor layer, the latent vector is passed to the variational quantum circuit directly where it is encoded using a Pauli feature-map, followed by layers of learnable rotation and entangling gates. Now the state is measured with multiple measurement shots (usually ~ 1024). This produces a distribution over all 2^n states. To minimize the loss function, we consider the probability of obtaining the state that was fed as input i.e $\log p(x|z)$. This process is repeated for all the samples with a predefined batch-size. The loss function is computed by summing the KL-divergence of the latent vectors and the network is optimized.

Chapter 4

Results and Discussion

In this chapter, we summarize the results obtained from our experiments. In the first section, we discuss the results on learning classical distributions. We observe how the QeVAE algorithm performs relative to the discrete CVAE on the MNIST dataset. In the second section, we discuss our findings on learning the distributions of quantum states with the QeVAE.

4.1 Learning Classical distributions

In the introduction and methods section of this thesis, we showed that quantum models might perform better than classical models since they are more expressive. In other words, since variational quantum circuits can learn complex distributions, we can obtain a model that provides a tighter bounder for the evidence lower bound loss introduced in section 2.22. Using methods detailed in chapter 3, we compare the performance of a QeVAE with a classical discrete VAE and the results are depicted in figure 4.1 and figure 4.2. Our keys observations are as follows:

1. Our proposed hybrid quantum-classical model can learn and generate images from the MNIST (6,9) database reliably. Furthermore, as shown in figure 4.2, we notice that the overall loss function decreases with the number of epochs (In ML literature, an epoch is one pass through the entire training dataset) and saturates until a point.

2. Based on the type of images generated, we find that both models give similar quality of

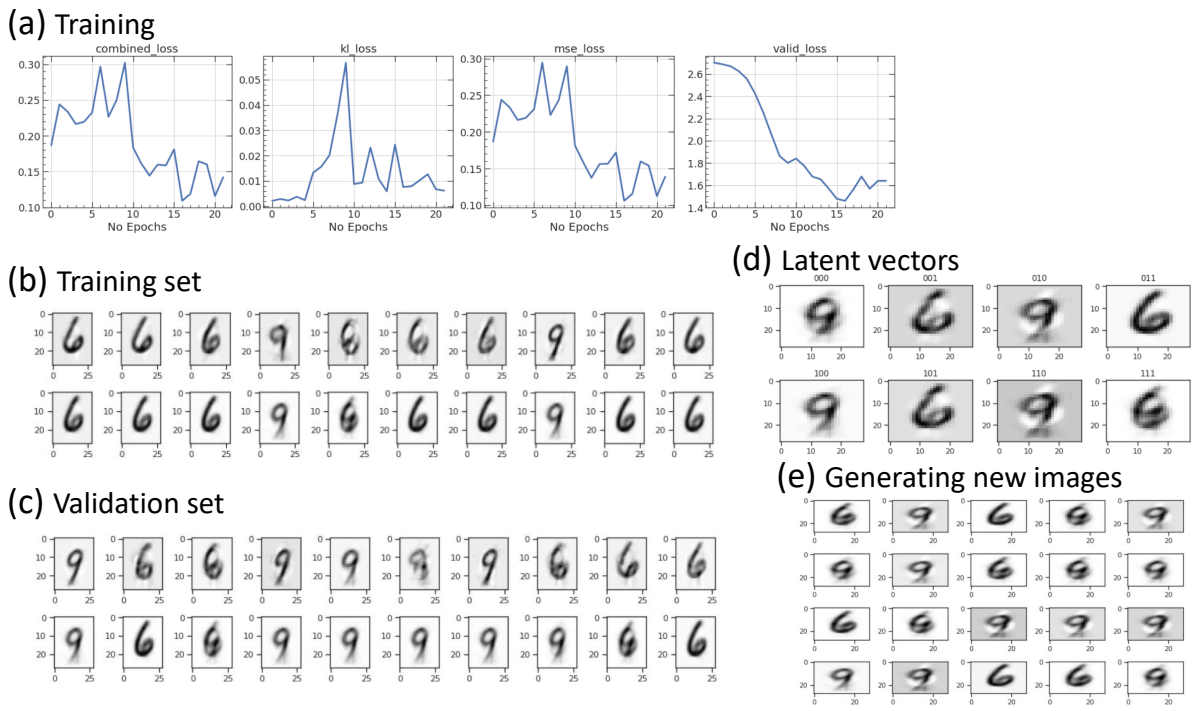

Figure 4.1: **Learning the MNIST-(6,9) dataset with a cVAE**: (a) While training the classical VAE, we observe a gradual decrease in the combined loss (reconstruction loss (mse loss) + β Kl-divergence loss) and training is halted based on the early stopping criterion on the validation loss. (b,c) After training, samples from the training set and validation set are passed through the neural network, here the top panel in (b,d) represent the input image and the bottom panel is the image recovered from the decoder. (d) For a latent-dimension containing only 8 bit-strings, sampling each bit-string leads to a unique image. We see that some bit-strings encode a '6' while some encode an '9' (e) Sampling random vectors from the latent space generates new combinations of images.

images. The generated images are blurry since the model actually only produces a vector of a very small dimension compared to the size of the image (28x28). The image is constructed from the vector via an inverse-PCA transform that is itself lossy. The final ELBO values of both the methods are in the similar range, with those of the classical VAE being higher in some cases.

3. We use an early stopping criterion on the validation set loss to prevent over-fitting. However, we find that early stopping does not work for quantum models, and learning is halted very early before. As shown in figure 4.3, the validation loss increases rapidly and then decreases for all the models we have trained, and the results produced through early stopping are very blurry. This indicates that our quantum models over-fit the training data and find it hard to

generalize on the validation set.

4. Although it appears that the classical generative model take lesser number of epochs, the difference can be attributed to the difference in the learning rates used to train the two models. Learning rates are tuned to achieve optimal performance and a balance between over-fitting or under-fitting.

5. Finally, since encoders in both the QeVAE and CVAE have the same number of parameters, we do not find any advantage in the number of parameters required to learn the posterior distribution on the *modified* MNIST dataset.

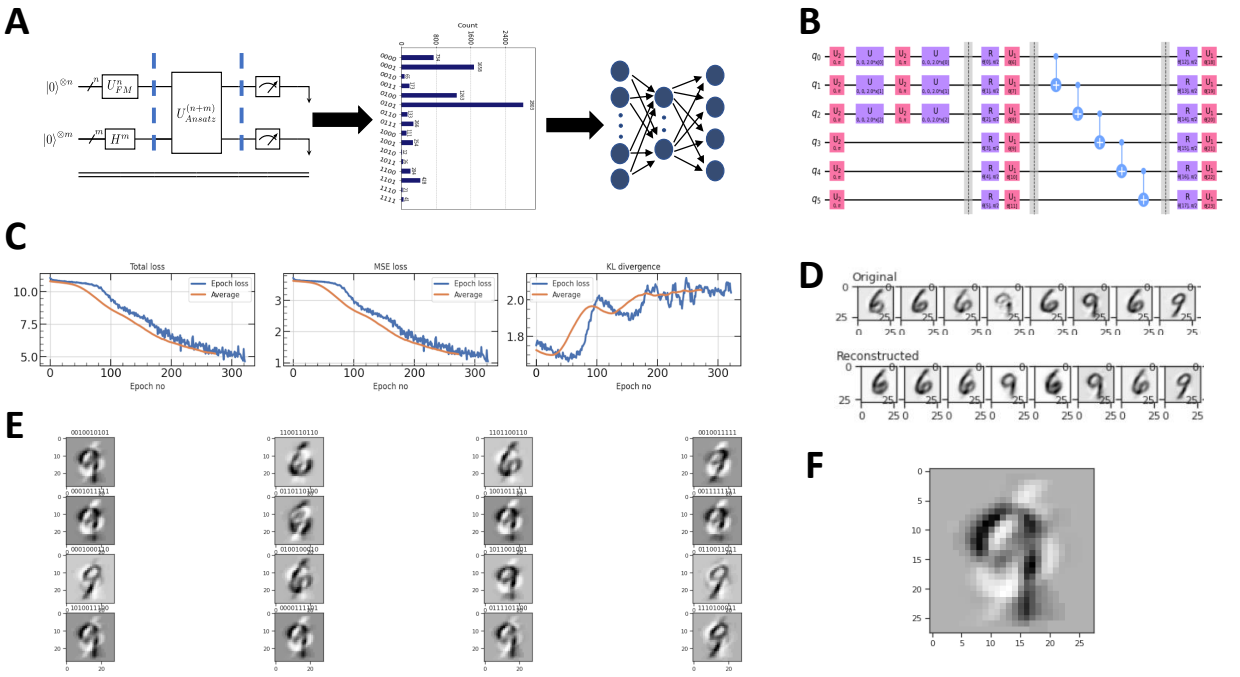

Figure 4.2: **Learning the MNIST-(6,9) dataset with QeVAE** (A) Hybrid quantum-classical neural network with a parameterized quantum circuit as the encoder, with discrete latent variables and a classical decoder. (B) The structure of the PQC until measurement with a Pauli-Z feature map on first three qubits with three ancilla qubits, followed by learnable rotation gates with linear entanglement. (C) Loss function curves during learning i.e, the Total Loss, Mean-square error or reconstruction loss and the KL loss ($1/\beta$=3), the average loss computes an moving average of 50 points and is offset to the left by 50 epochs, each epoch is one pass through the entire dataset (D) Results of training, top panel depicts the original images while the bottom panel shows the reconstructed images (E) New sample images by sampling latent vectors (F) A new sample image generated by sampling a random vector $v \in R^{10}$ where each $v_i \in (0,1)$

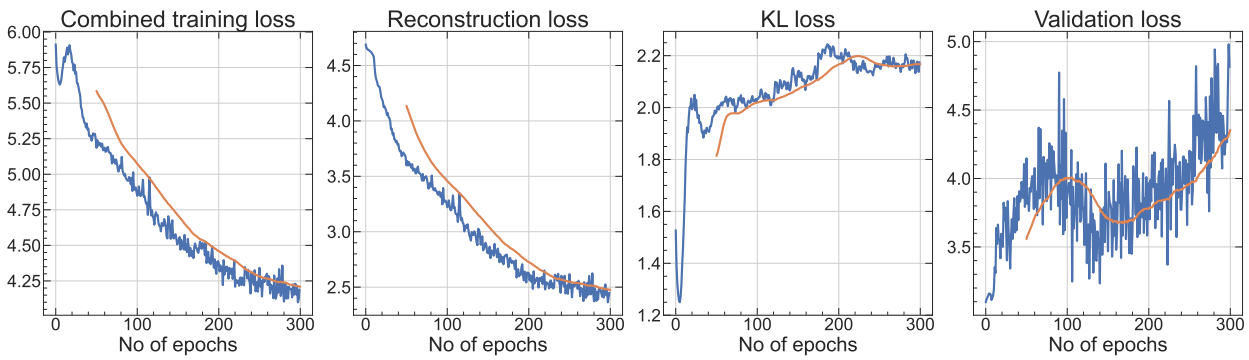

Figure 4.3: **QeVAE over-fitting the training data:** In some instances of training the QeVAE to learn MNIST images, we find that the model over-fits the training dataset: losses on the training data continue to reduce whereas the loss of the validation data increases, decreases and continues to increase after 120 epochs. The rolling-average (orange line) shows that the validation loss increasing when the training losses rapidly decrease. (Here we use a training dataset of 70 images and a validation set of 30 images. The ansatz contains 5 qubits with 20 learnable parameters. β is linearly annealed from $0 \rightarrow 1$)

4.1.1 Discussion

It is evident from the previous section that classical VAEs perform at par or better than Quantum-enhanced VAEs at learning the distribution of the modified image dataset. There are many caveats to consider and to fully arrive at an answer to the *quantum advantage* question. Firstly, we note that since images are produced by the same generator network across both the models, any performance differences must be attributed to the encoder network. Within the encoder, we note that quantum models tend to over-fit and takes a very large number of epochs to train. A possible cause for the above could be the problem of parameter initialization. Random quantum circuits initialized far from their optima are known to suffer from the problem of barren plateaus where the gradients with respect to the parameters become diminishingly low and parameter updates very slow. Our quantum models are initialized with random parameters in (-1,+1), and the final parameters learnt are very different and differ by 3x. In addition, gradients need to be propagated first through the decoder, and then the circuit, rendering the computation slow. A possible remedy is to initially train the quantum model through tensor networks and then slowly increase the entanglement structure of the encoder [63].

Secondly, the KL-divergence loss function and the relative-weight term β play an important role in the quality of images produced. A higher β prioritizes minimizing the difference between the measurement distribution produced and that of a uniform distribution. This constrains the

quantum circuit to produce a distribution closer to a uniform distribution with less emphasis on the encoded data-sample. On the other hand, the reconstruction term aims to produce a measurement distribution that efficiently encodes different aspects of the dataset. In our simulations, we find that higher β values produce better quality images whereas lower β values train better. A higher β implies that the quantum circuit is producing a distribution close to the uniform distribution, a distribution similar to uniform noise, and bulk of the generative work is performed by the decoder. This results in the decoder solely working on generating images from noise rather a latent space with rich features, a problem popularly described in literature as *posterior collapse*. To prevent the collapse of the posterior distribution, we focus on maintaining a simple generator network.

Lastly, our results substantiate that classical deep learning models are highly expressive and can encode the complex distributions produced by classical distributions (such as the distributions of pixel values in a set of images). Although VAEs share their fair set of disadvantages like factorized latent variables, and blurry images, our results highlight the fact they can learn distributions and generate data with desirable performance. However, from the above considerations it not clear if this implies that QeVAEs are better at approximating the ELBO loss encountered in VAEs. We indicate the future plans to improve and consolidate our claims in the next chapter on outlook and conclusion.

4.2 Learning Quantum distributions

In the tables presented below, we summarize the best fidelity obtained across each type of measurement dataset. We compare the final fidelity between the target distribution and that produced by a random uniform guess, a classical variational autoencoder (VAE), and a Quantum-enhanced Variational Autoencoder. For each type of state, we consider five different random seeds. QeVAE results include the best fidelity observed across different hyper-parameters like latent size, feature-map, prepossessing-layer, and relative KL-divergence term β.

Table 4.1: Fidelity for Product states

No qubits	4						8					
Seed	12	16	27	44	102	**Mean** $\downarrow$	12	16	27	44	102	**Mean** $\downarrow$
Uniform	.471	.356	.302	.156	.436	.344	.164	.306	.081	.306	.106	.193
CVAE	**.998**	**.995**	**.998**	**.995**	**.995**	**.996**	**.983**	**.979**	**.982**	**.983**	**.987**	**.983**
QeVAE	.995	.827	.882	.973	.892	.914	.875	.947	.870	.823	.957	.894

Table 4.2: Fidelity for Quantum circuit states

No qubits	4						8					
Seed	12	16	27	44	102	**Mean ↓**	12	16	27	44	102	**Mean ↓**
Uniform	.477	.366	.315	.154	.431	.348	.158	.279	.080	.309	.107	.187
CVAE	.501	.667	.597	**.925**	.758	.690	.229	.423	.079	.304	.181	.243
QeVAE	**.981**	**.976**	**.950**	.873	**.912**	**.938**	**.665**	**.654**	**.388**	**.548**	**.591**	**.569**

Table 4.3: Fidelity for Haar random states

No qubits	4						8					
Seed	42	96	27	101	102	**Mean ↓**	12	43	16	27	2	**Mean ↓**
Uniform	.772	.776	.777	.771	.768	.773	.766	.770	.773	.781	.772	.772
CVAE	.795	.798	.800	.788	.788	.794	.754	.755	.757	.766	.763	.759
QeVAE	**.839**	**.983**	**.913**	**.988**	**.932**	**.931**	**.876**	**.878**	**.887**	**.887**	**.887**	**.883**

Table 4.4: Fidelity of Quantum-kicked rotor states

No qubits	4			8		
Type	Localized (k=6)	Diffusive (k=0.5)	**Mean ↓**	Localized (k=6)	Diffusive (k=0.5)	**Mean ↓**
Uniform	.175	.838	.506	.053	.418	.236
CVAE	.723	.908	.815	.061	.406	.233
QeVAE	**.991**	**.992**	**.991**	**.912**	**.616**	**.764**

Table 4.5: Hardware results for a 4 qubit quantum circuit state

State	Fidelity	Simulator	Hardware	Suppression	Mitigation
Uniform	0.477	✓			
CVAE	0.501	✓			
QeVAE	0.981	✓			
QeVAE	0.658		✓		
QeVAE	0.642		✓	✓	✓

From the above tables, we observe that across all quantum states with entanglement, the final fidelity obtained from a QeVAE outperforms the classical VAE and a random guess. In addition, the number of parameters in the classical VAE is $O(\exp(n))$ whereas it is $O(n)$ in QeVAEs. To further validate our findings, we run the best QVAE models on real quantum devices and see that the obtained fidelity is higher than those achieved by classical methods.

What is the best feature-map to choose?: We look at two different kinds of feature-maps and latent sizes for learning the measurement distribution of Haar random states, as shown in figure 4.4. We consider the Pauli-Z and Pauli-ZZ feature map and latent sizes: 0, 4, and 8. Latent size 0

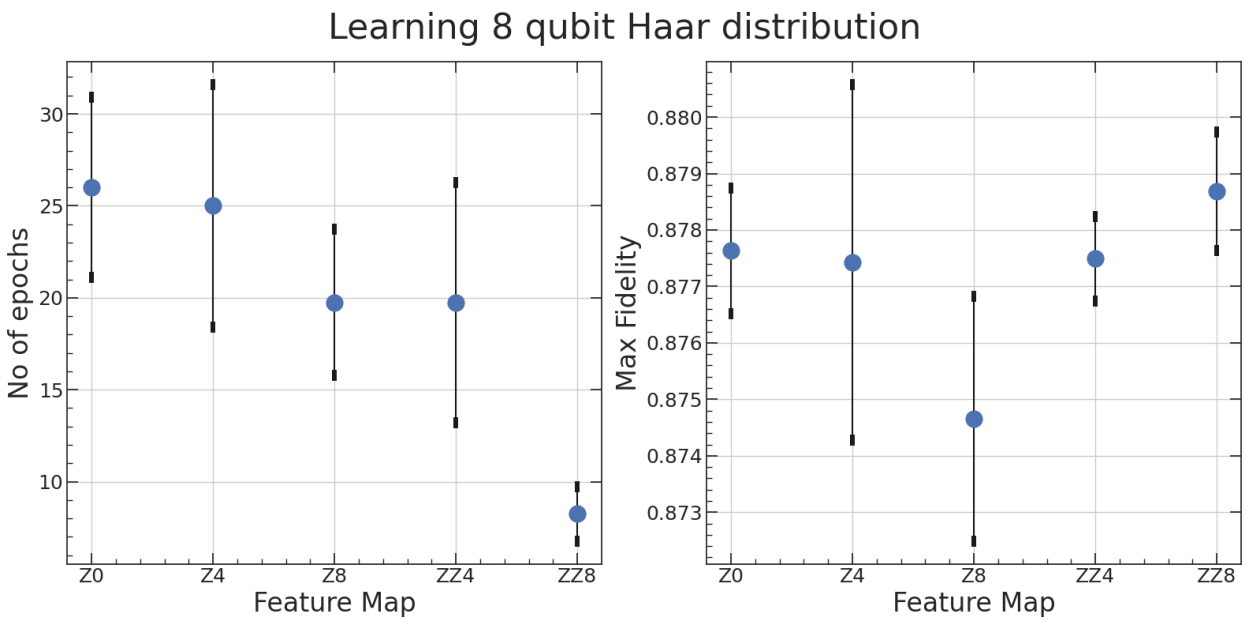

Figure 4.4: **Variation in learning Haar states with different feature maps** (Left) The average number of epochs and its standard deviation required to achieve the same fidelity across different types of quantum embeddings. The average is over five different seeds (Right) The average maximum fidelity and standard deviation for learning the measurement distribution of haar states.

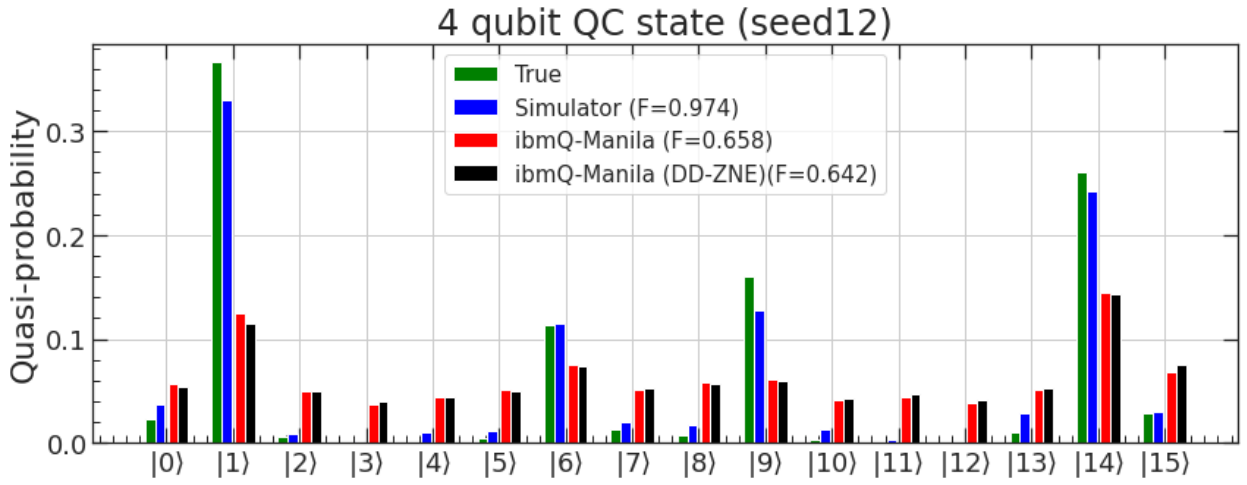

Figure 4.5: **Inference on IBMq Manila**: The true distribution produced by an unknown 4-qubit system (green) is used to train a QeVAE on IBM's qasm-simulator (blue) that results in a final of 0.97. After training, the decoder is executed on the IBM-Manila. The measurement distribution produced from hardware has a total fidelity of 0.658 which changes to 0.642 with error-mitigation (Zero noise extrapolation) and error-suppression (Dynamic decoupling)

corresponds to the QCBM case and a quantum embedding is not required. We find that all models achieve the same range of maximum fidelity and that the ZZ feature map with 8 latent variables achieves it the in the lowest number of epochs. This can be explained by the fact that the ZZ8 feature-map with 8 latent variables is the deepest and with the largest number of gates, while the QCBM requires the smallest number of gates and is the most shallow.

Hardware run: We execute the best model trained on the simulator on IBM Hardware by transpiling the decoder circuit. To generate the output distribution, random samples from $N(0,1)$ are propagated through the preprocessor linear layer and then through the circuit (executed on hardware). An average over all initial random points yields the desired output distribution. Our results are depicted in figure 4.5 and in table 4.5. We find that final fidelity is lesser than that on a simulator. Using error mitigation and suppression techniques, we are able to perform better than the classical VAE.

4.2.1 Discussion

We know from literature that the measurement distribution obtained from product states are classically *easy* and that measurements obtained from states with quantum correlations is classically *hard*. Our results not only corroborates with the above observation but also shows that quantum-enhanced classical models can overcome the drawbacks of purely classical models. Our main results and observations from data is as follows:

1. Firstly, all models are able to learn the measurement distribution obtained from various quantum states. Our proposed algorithm achieves the highest fidelity across all types of datasets, other than product states. The inherent ability of our model to learn quantum correlations i.e we are able to produce entangled multi-qubit states through variational quantum circuits and tailor the rotation gates to reproduce a desired distribution allows them to outperform the classical model for the quantum circuit, haar random and kicked rotor states. Furthermore, all quantum models require only $O(n)$ parameters where n is the number of qubits whereas classical models require $O(\exp(n))$ parameters to reproduce the same distribution with similar fidelty.

2. *Fidelity for Product states*: Since product states do not employ quantum-correlations between different qubits in a many-body system, an n body system require only $2n$ parameters

to be learnt. This polynomial dependence and presence of only classical correlations in the output distributions enables classical VAEs to efficiently learn and reproduce the distribution with very high fidelity. In addition, the ansatz in our model is not always tailored for product states, i.e the ansatz has an entanglement structure and the quantum state produced from the QeVAE will exhibit entanglement. Such a class of distribution produced cannot in general be similar to that produced by product states. This might explain the reduced fidelities for QVAEs.

3. *Fidelity for Haar, Quantum circuit and Quantum kicked rotor states*: We notice that for measurement distributions obtained from more generic quantum states, our quantum models outperform the classical models in terms of final fidelity. In some cases (like seed 44 for quantum circuit states), the score of the classical model is higher because the resulting measurement distribution is highly concentrated around a single output ($> 80\%$). In such cases, it becomes easier to just predict the output statistically than learning the intrinsic structure of the quantum circuit producing the state. Nonetheless, we find that QeVAE models require only $O(n)$ rotation gates to learn the output distribution and achieve a fidelity score that is unattainable to other methods.

4. Since QeVAE is a hybrid model, it operates through the synergy of quantum and classical resources. Classical models are well-versed at producing non-linear transformations whereas quantum models are restricted to unitary or linear transformations. Conversely, quantum models can encode quantum correlations like entanglement and discord, a phenomenon inaccessible to classical methods. An important feature of our model is the ability to leverage the merits of both the classical and quantum models, and outperform both models.

5. The QeVAE model contains many hyper-parameters that can be tuned to achieve optimal performance. In the above tables, we have presented the result for the best hyper-parameter setup. When the latent-size is set to zero, there is no contribution from the classical encoder and the output distribution is produced by the circuit alone. In such cases, the resulting model is the Quantum circuit Born Machine (QCBM), wherein the circuit parameters are iteratively updated to minimize the difference between the output and target distributions. Thus in the latent-size=0 limit, our model results in the QCBM generative model. The KL divergence term in the loss function can be neglected, and minimizing the negative expected log-likelihood becomes equivalent to minimizing the KL-divergence between the output and target distributions.

6. We would like to highlight the subtle difference between training a QCBM and our approach for QeVAEs. QCBMs are input-agnostic and the algorithm's goal is to minimize the KL-divergence betwwen the distribution produced by an ansatz and a true distribution (obtained from data). Computing the KL divergence can be costly and the number of samples to be evaluated increases exponentially with n. Alternatively, within QeVAEs we focus on maximizing the log-likelihood of producing a bitstring x, ie maximise $\log p(x|z)$. This allows the latent space Z to learn a representation for the input distribution and produce reliable a X. Furthermore, training QeVAEs is by minimizing a loss function with a particular batch-size. Thus gives one the flexibility to start training the QeVAE without having to wait for the entire measurement dataset and begin learning through iterative access to samples.

Applications: Circuit compression (a case study)

Having described the ability of our model to learn complex quantum distributions, we conclude this chapter with a practical application, on circuit compilation. Circuit compilation or circuit compression is an important area of focus in the NISQ era. Since the depth of circuits executable on hardware is limited, there is a need to transform deep circuits into shallow ones by altering the sequence of gates and reducing the overall size and complexity. Furthermore, multi-qubit gates like the CNOT gate, T gate, and SWAP gates are expensive to implement. Efficient circuit compilation assists in faster computations, and allows accurate simulation of quantum systems. This has immense applications in many-body physics, condensed matter physics, and quantum chemistry. Here, we show show QeVAEs can help in exponentially reducing the complexity of circuits by learning to reproduce measurement distributions with fewer gates. We show our results in figure 4.6.

We simulate an unknown quantum state by considering a deep quantum circuit with twenty layers of rotation and entangling gates. In reality, the form of the circuit is unknown and one only has access to the measurement data. A projective measurement on such a state produces a measurement distribution as shown in figure 4.6(e). Note that our goal here is to reproduce the measurement distribution and not to learn the original state itself. Through the QeVAE learning approach, we can learn the measurement distribution with high fidelity. After training, we can discard the encoder part of the circuit, and the decoder provides a sequence of gates that can be implemented on hardware to generate the same distribution. With this approach, we achieve a final fidelity of 0.956 (figure 4.6(d)) and a multi-fold reduction in the number of gates as seen in figure

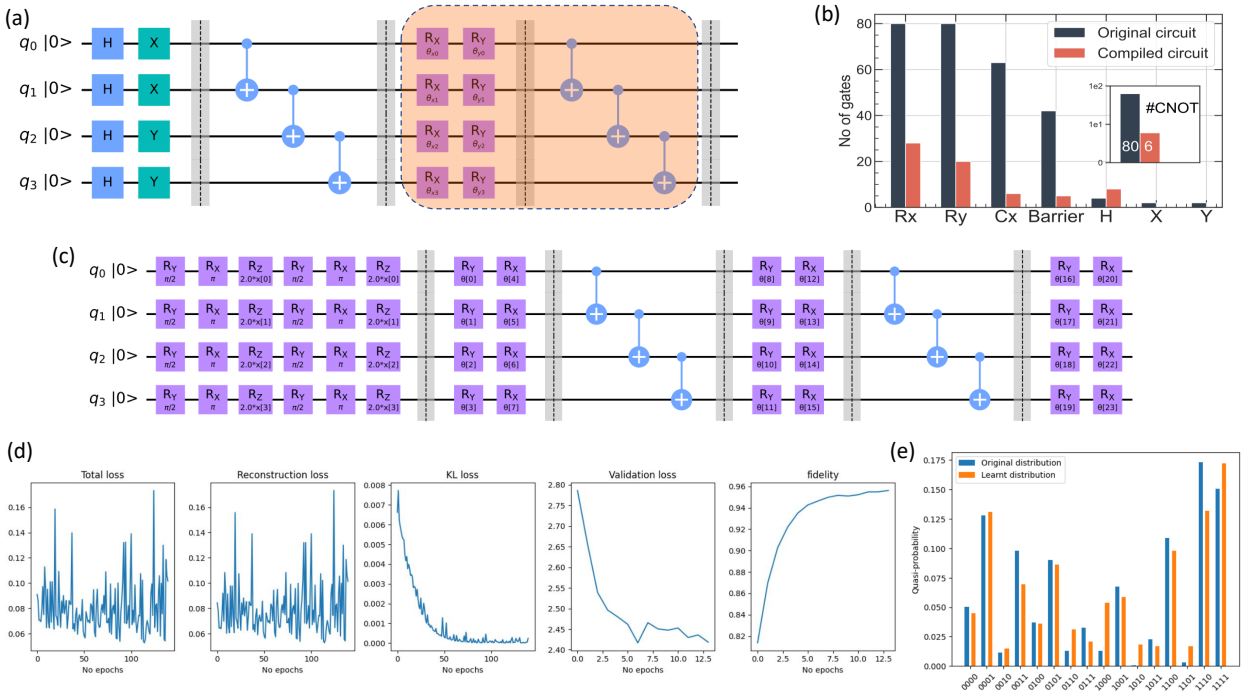

Figure 4.6: **Circuit compilation with QeVAEs** (a) Original circuit structure that produces a measurement distribution. The orange box represents a single layer of rotation and entangling layer that is repeated 20 times. (b) The compiled circuits requires very few gates when compared to the original circuit that produces the state. Particularly, the number of CNOT gates is reduced by $\sim$14X (c) The variational quantum circuit containing the Pauli-Z feature Map that embeds data $\boldsymbol{x}$ from the latent space and the ansatz with only 16 parameters. (d) Results of training the ansatz to produce the measurement distribution. We achieve a final fidelity of 0.956 (e) Original and the distribution produced by the QeVAE after training.

4.6(b). The initial circuit majorly contains: 80 R_x gates, 80 R_y gates, 63 *CNOT* gates which are reduced to 28, 20, and 6 respectively.

Chapter 5

Outlook

This chapter begins by discussing various questions raised in the Theory and Methods chapters regarding the expressivity of quantum-enhanced neural networks. Next, we discuss the implications and limitations of the new results presented in this thesis. Lastly, we provide an outlook on a few open problems within generative quantum machine learning.

5.0.1 What are the key takeaways from our work?

The major outcome of our work is a working hybrid quantum-classical machine learning model for generative learning and its bench-marking against purely classical and quantum models. We have shown that our model can learn distributions, derived from both classical sources and quantum systems. In addition, our models are particularly suitable for quantum devices with small number of noisy qubits with limited connectivity. Firstly, we have considered the MNIST database as a representative of a dataset obtained from a classical source. Here we find that classical models require similar amount of resources and can provide a better final performance than our proposed QeVAE model. Furthermore, we find that our hybrid model tends to over-fit the training dataset and that additional fine-tuning is required to generalize on the validation dataset. We do not find substantial evidence to show that quantum models can provide a tighter lower bound to infer the posterior distribution. In agreement with literature, we find that quantum models do not provide a substantial advantage in terms of accuracy for classical datasets. Thus, an implication one can work towards is that quantum generative models can learn classical distributions but it is a more

involved task to prove an accuracy advantage over the more robust classical generative models.

Secondly, we have investigated the ability of quantum generative models to learn the distributions obtained from the measurement of quantum many-body systems. Such distributions are known to be demanding for classical generative models and we verify the same in our experiments. We further go on to show that our proposed hybrid-model can learn these distributions with a much higher final fidelity. We find this trend to be universal across a range of different types of quantum states, from generic haar random states to dynamic kicked rotor states. We find that our model has multiple hyper-parameters to tune and in one such case, when the latent size is set to zero, we obtain at the Quantum circuit Born Machine, the most popular pure quantum-mechanical generative model. In addition, we have shown that QeVAEs can be useful for the practical task of circuit compilation.

To conclude, we have numerically shown that quantum-mechanical parametric models achieve higher fidelity than classical models at learning distributions produced by quantum-mechanical sources. Theoretically proving our findings is an on-going effort and future work could focus on demonstrating their capabilities.

5.0.2 What are some limitations of our work?

Our work highlights the ability of quantum-enhanced models to perform better than classical models within the variational autoencoder framework. There are some drawbacks of the VAE approach and they include: (1) information loss in the encoding and decoding process that often results in blurry images; (2) posterior collapse: if the likelihood function is much more complex than the posterior, then the encoder ignores the input data and outputs a trivial latent space, leading the decoder to reconstruct the data from noise. Other classical generative methods like GANs, Diffusion Models, and Normalizing flows have gained much traction for producing better quality images. Although, they also contain their fair share of disadvantages like mode collapse in GANs and the large number of parameters in Diffusion models, they have been shown to produce better quality images.

The QeVAE algorithm contains many tunable parameters like the learning rates of the encoder and decoder, the relative weight of the KL divergence term in the loss function β, the patience factor for early-stopping, the entanglement structure of the ansatz, the feature-map in the param-

eterized quantum circuit etc. In our calculations, we observed that obtaining an accurate final distribution requires careful fine-tuning of these hyperparameters. For example, we found out that initializing the preprocessing layer near $\mathcal{N}(0, 1)$ for our QeVAE is very essential to outperform the classical model for learning measurement distributions. When there was no preprocessing layer, the models quickly over-fit the training data and only sometimes performed better than classical VAE. However, when a preprocessing layer was added before the decoder, we observed a substantial increase in performance. In addition, we found that feature maps with entanglement (like the Pauli ZZ map) produced distributions with the same accuracy but require fewer epochs. The correct hyper-parameters have to be found out through an extensive grid-search, and through semi-empirical means.

5.0.3 Outlook

In the future, we will focus on verifying our results for learning the classical distribution on more diverse datasets to ratify the implications arrived. Particularly, text datasets can be used for discrete distributions. In addition, we will also examine if modifications in the ansatz: incorporating a non-linearity (like mid-circuit measurements) can enhance the performance of the QeVAE and provide a better bound to the ELBO. On the other hand, we have already shown quantum-enhanced models are better at learning measurement distributions. It is a challenging task to examine this behavior on real hardware and we speculate that the designed ansatz and parameters learnt must produce reliable results when executed on hardware with suitable error mitigation and suppression techniques.

Bibliography

[1] Charles H Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21:905–940, 1982.

[2] Richard P Feynman. Quantum mechanical computers. *Optics news*, 11(2):11–20, 1985.

[3] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[4] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[5] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[6] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[7] Ryan LaRose. Overview and Comparison of Gate Level Quantum Software Platforms. *Quantum*, 3:130, March 2019.

[8] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[9] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[10] Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11:501–528, 2020.

[11] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[12] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, Oct 2017.

[13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, July 2021.

[14] Safe driving cars. *Nature Machine Intelligence*, 4(2):95–96, February 2022.

[15] Bill Fefferman and Chris Umans. The power of quantum fourier sampling. *arXiv preprint arXiv:1507.05592*, 2015.

[16] Ryan Sweke, Jean-Pierre Seifert, Dominik Hangleiter, and Jens Eisert. On the Quantum versus Classical Learnability of Discrete Distributions. *Quantum*, 5:417, March 2021.

[17] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March meeting abstracts*, volume 2018, pages L58–003, 2018.

[18] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[19] Stephen Barnett. *Quantum information*, volume 16. Oxford University Press, 2009.

[20] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[21] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic Programming and Evolvable Machines*, 19(1-2):305–307, 2018.

[22] Tom Michael Mitchell et al. *Machine learning*, volume 1. McGraw-hill New York, 2007.

[23] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, November 2021.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[27] Nan Ding and S.v.n. Vishwanathan. t-logistic regression. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[28] Lourens Waldorp, Maarten Marsman, and Gunter Maris. Logistic regression and ising networks: prediction and estimation when violating lasso assumptions. *Behaviormetrika*, 46(1):49–72, August 2018.

[29] Lei Wang. Generative models for physicists. Technical report, Tech. rep., Institute of Physics, Chinese Academy of Sciences, GitHub. io, 2018.

[30] Murphy Yuezhen Niu, Andrew M Dai, Li Li, Augustus Odena, Zhengli Zhao, Vadim Smelyanskyi, Hartmut Neven, and Sergio Boixo. Learnability and complexity of quantum samples. *arXiv preprint arXiv:2010.11983*, 2020.

[31] Andrea Rocchetto, Edward Grant, Sergii Strelchuk, Giuseppe Carleo, and Simone Severini. Learning hard quantum distributions with variational autoencoders. *npj Quantum Information*, 4(1):28, 2018.

[32] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. 3(3):030502.

[33] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[34] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[35] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[36] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[37] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[38] Y Kiat, Y Vortman, and N Sapir. Feather moult and bird appearance are correlated with global warming over the last 200 years. *Nature Communications*, 10(1):2540, 2019.

[39] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[40] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

[41] Maria Schuld and Francesco Petruccione. *Supervised Learning with Quantum Computers*. Springer International Publishing, 2018.

[42] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019.

[43] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, Oct 2020.

[44] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.*, 127:120502, Sep 2021.

[45] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99:032331, Mar 2019.

[46] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[47] Jaan Altosaar, Rajesh Ranganath, and Kyle Cranmer. Hierarchical variational models for statistical physics.

[48] Jean Daunizeau. The variational laplace approach to approximate bayesian inference. *arXiv preprint arXiv:1703.02089*, 2017.

[49] Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. *Advances in neural information processing systems*, 31, 2018.

[50] Manfred Opper, Marco Fraccaro, Ulrich Paquet, Alex Susemihl, and Ole Winther. Perturbation theory for variational inference. In *Proc. Conf. Neural Inf. Process. Syst. Workshops*, 2015.

[51] Xun Gao, Eric R. Anschuetz, Sheng-Tao Wang, J. Ignacio Cirac, and Mikhail D. Lukin. Enhancing generative models via quantum correlations. *Phys. Rev. X*, 12:021037, May 2022.

[52] Amir Khoshaman, Walter Vinci, Brandon Denis, Evgeny Andriyash, Hossein Sadeghi, and Mohammad H Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1):014001, sep 2018.

[53] Pablo Rivas, Liang Zhao, and Javier Orduz. Hybrid quantum variational autoencoders for representation learning. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 52–57, 2021.

[54] Junde Li and Swaroop Ghosh. Scalable variational quantum circuits for autoencoder-based drug discovery. In *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 340–345, 2022.

[55] Stephen Boyd and Almir Mutapcic. Stochastic subgradient methods. *Lecture Notes for EE364b, Stanford University*, 2008.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[57] James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

[58] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner. Simultaneous perturbation stochastic approximation of the quantum fisher information. *Quantum*, 5:567, 2021.

[59] Marcel Hinsche, Marios Ioannou, Alexander Nietner, Jonas Haferkamp, Yihui Quek, Dominik Hangleiter, Jean-Pierre Seifert, Jens Eisert, and Ryan Sweke. Learnability of the output distributions of local quantum circuits. *arXiv preprint arXiv:2110.05517*, 2021.

[60] Mithuna Yoganathan. A condition under which classical simulability implies efficient state learnability. *arXiv preprint arXiv:1907.08163*, 2019.

[61] Christopher A Fuchs and Carlton M Caves. Ensemble-dependent bounds for accessible information in quantum mechanics. *Physical Review Letters*, 73(23):3047, 1994.

[62] M.S. Santhanam, Sanku Paul, and J. Bharathi Kannan. Quantum kicked rotor and its variants: Chaos, localization and beyond. *Physics Reports*, 956:1–87, April 2022.

[63] Manuel S Rudolph, Jacob Miller, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. Synergy between quantum circuits and tensor networks: Short-cutting the race to practical quantum advantage. *arXiv preprint arXiv:2208.13673*, 2022.