Quantum Generative Modelling Using the Kicked Ising Spin Chain Dynamics

A Thesis

submitted to

Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
Master of Science in Quantum Technology

by

Janhavi J

Indian Institute of Science Education and Research Pune

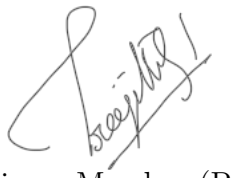
April, 2026

Supervisor: Prof. M. S. Santhanam

Certificate

This is to certify that this dissertation entitled "Quantum Generative Modelling Using the Kicked Ising Spin Chain Dynamics" submitted towards the partial fulfilment of the Master of Science in Quantum Technology represents study/work carried out by Janhavi J from Indian Institute of Science Education and Research Pune under the supervision of Dr Prof. M. S. Santhanam (Department of Physics, IISER Pune) during the academic year 2025–2026.

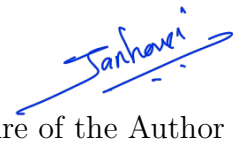
Signature of Supervisor (Prof. M. S. Santhanam) Date: 21.4.2026



Signature of Advisory Member (Prof. Sreejith G. J.) Date: 7 May 2026

Declaration

I hereby declare that the matter embodied in the report entitled "Quantum Generative Modelling Using the Kicked Ising Spin Chain Dynamics" are the results of the work carried out by me at the Indian Institute of Science Education and Research, Pune under the supervision of Prof. M. S. Santhanam (Department of Physics, IISER Pune) and the same has not been submitted elsewhere for any other degree.



Signature of the Author (Janhavi J) Date: 21-04-2026

Abstract

Quantum Machine Learning is a paradigm that aims to harness natural quantum dynamics to perform various machine learning tasks. Variational quantum algorithms, which form a hybrid classical–quantum framework, are a promising area of research, especially in the current NISQ era where fully fault-tolerant quantum computation is not yet available.

In this work, we replace the typical randomly parametrised circuits with the kicked Ising spin system in the Quantum Circuit Born Machine (QCBM), which is used as the ansatz. This choice is motivated by the structured dynamics of the system, which can potentially offer better expressibility with a reduced parameter space. We have treated it as a toy model and have analysed its performance for supervised generative tasks, such as learning analytical distributions and the Bars and Stripes (BAS) dataset. The MMD loss function is used for training purposes. Finally, we have presented the results observed and directions for improvement in its performance and future work.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. M. S. Santhanam, for his invaluable guidance, patience, and insightful feedback throughout the course of this research. I am equally indebted to my thesis advisory member, Prof. Sreejith G. J, for his unwavering support and mentorship throughout my journey at IISER Pune.

I am grateful to Bharathi Kannan and Nisarg Vyas for the stimulating discussions that helped shape this work, and to IISER Pune for providing the ideal environment and resources to conduct this research.

Finally, I want to thank my family who have always been my greatest strength and source of encouragement. A special mention goes to Dharani Kumar, who has been my constant pillar of strength throughout this endeavour.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum processors and their computational power have been a major topic of research recently. Significant work is being dedicated to demonstrating that the quantum-inspired machine learning models[6] exhibit a quantum advantage over their classical counterparts. Quantum computing aims to solve problems that are otherwise intractable for classical computers [7]. Progress in the fields of material science, hardware fabrication, and the techniques for quantum control, along with the developments in error correction and compilation, has brought us nearer to large-scale fault-tolerant quantum computers. However, completely noise and error-resilient quantum devices remain a long-term goal to be achieved. The quantum processors available today are limited to a few hundred physical qubits and can execute only shallow circuits before the system gets dominated by the noise from the environment and decoherence, which degrades the quantum state.

The current Noisy Intermediate-Scale Quantum (NISQ) era has significantly reshaped the direction of research. Instead of waiting for ideal hardware that is resilient to noise and errors, there has been immense effort to design algorithms compatible with existing constraints and available systems, while simultaneously demonstrating a quantum advantage. It is believed that NISQ devices may still have applications that are beneficial in the near term.

The central idea of this approach is the hybrid quantum-classical algorithm[8], where Variational Quantum Algorithms[9] have been the most widely explored strategy of using a quantum processor and a classical computer working together in an iterative loop. Variational quantum algorithms(VQAs) have been the most explored strategy developed to attain the quantum advantage on NISQ devices. VQAs, in many ways, are the quantum analogue of neural networks. Using a quantum computer to execute these parameterised circuits, which act as the variational ansatz and then leveraging the power of classical optimisers for parameter updates, thus outsourcing the optimisation task. Thus, its quantum and classical parts work hand in hand. This also has an advantage, as the circuits can be kept shallow enough to mitigate noise

while still performing complex computational tasks.

Parametrised Quantum circuits (PQCs) are those that consist of adjustable single-qubit and entangling gates in a layered architecture. This structure provides the flexibility for diverse computational tasks. PQCs thus have been used as learning models for various machine learning tasks such as classification, regression, clustering, and generative modelling.

Generative models are of particular interest, as they are considered a promising avenue for the realisation of quantum advantage on classical data. This is due to the inherent difficulty of modelling high-dimensional probability distributions, which quantum systems can naturally represent. Generative models are those that learn an unknown target probability distribution of the data, which is π_{target}, and the model thereby produces samples from the approximate probability distribution p_{model}, which are new and previously unseen. Unlike classification or regression tasks, which focus on predicting the outputs based on inputs, generative models require capturing the overall structure of the distribution, which makes the task much more rigorous, but at the same time can be used in various tasks such as data synthesis, pattern recognition, unsupervised learning and so on. Previously, these tasks have been performed on Quantum Circuit Born Machines(QCBMs), which act as the quantum generative models.

The central approach of quantum generative models is such that this probability distribution is encoded into the quantum states. The large state space, which grows exponentially, provides the capacity to represent complex distributions. The probability of obtaining a specific measurement outcome is given by the square of the amplitude of the corresponding component of the state, as described by the Born rule. Thus, quantum states by nature are probabilistic, and we leverage this property for generative tasks. In addition to the large Hilbert space, the entanglement present enables the quantum models to capture the correlations that might be difficult to represent classically.

VQAs have a lot of applications in various fields, in solving various problems. Despite their versatility and applications, VQAs face a lot of difficulties, including trainability, efficiency and accuracy. The inherent random nature of the parametrised circuits, since they are chosen in an ad-hoc way, and the exponentially growing dimension of the Hilbert space with the number of qubits, make the training challenging due to the emergence of barren plateaus. Variational Quantum Circuits (VQCs) are optimised by iteratively updating the parameters based on the cost function. However, this adds to its challenges since such explicit determination of losses requires evaluating model probabilities, which can be very expensive in high-dimensional spaces. This raises the question of whether a more structured ansatz, which is the quantum systems, should be used instead.

While Born machines with parametrised circuits as ansatz have been illustrated and studied, we address here the question if other quantum many-body systems can serve as an effective ansatz for Born machines and investigate their ability to learn the data distribution and their performance as a generative model. Here we are replacing the random parametrised circuits, which lack structure, with a structured ansatz.

In this work, we explore a physically motivated alternative to random parametrised circuits. Instead of an unstructured ansatz, we have used a Floquet system, specifically a kicked Ising spin chain, a system which is periodically driven. The system is equivalent to that of a set of coupled kicked tops with variable coupling. The time-evolution operator here is the Floquet unitary. Thus, this provides us with a perspective to study if a dynamical system can be used as a generative model and its performance can be evaluated. Quantum many-body systems have inherent structure as a result of local interactions and controlled dynamics, and thus exhibit an ability to generate complex distributions and states. Using this circuit as an ansatz in the QCBM framework and demonstrating its ability to learn simple distributions has been done in this work. This allows us to examine if physically inspired circuits offer advantages in terms of expressibility or interpretability. We have done two main tasks to evaluate this idea. Firstly, we have also used the bars and stripes dataset(2x2), which is the standard benchmark for generative models and looked at the samples produced by the model and tested if it is able to produce valid samples. Secondly, the model is trained to learn simple distributions, which are predefined target distributions and assess its ability to approximate them. With these tasks, we try to analyse and understand the generative performance and the learning capability of the following approach.

The thesis is structured as follows. In Chapter 2, a general introduction to classical and quantum chaos, Floquet formalism, kicked top model and its spectral properties, and preliminaries of Quantum Machine Learning, along with an explanation of the architecture of Variational Quantum Algorithms and Born Machines, is provided. Chapter 3 presents the framework and methodology used in this work. The kicked Ising spin chain model used for our work is introduced. Subsequently, the loss function adopted for this work, Maximum Mean Discrepancy (MMD), is explained, including its formulation and key components. The chapter concludes with a discussion of the implementation details, including basis encoding, the learning algorithm, and parameter choices. In Chapter 4, we present and benchmark our results for classical distributions. We analyse the results obtained from training the proposed quantum generative model on the BAS datasets. The ability to learn different target distributions of the kicked Ising ansatz is examined and discussed. In the Conclusions, the results presented in the thesis are summarised, and possible directions for future work are discussed.

Chapter 2

Background: Quantum Chaos and Quantum Machine Learning

2.1 Classical Chaos

In classical mechanics, a physical system is described by the Hamiltonian H, which is a scalar function on the phase space of d degrees of freedom, with $(q_1, q_2, \ldots, q_d)$ as the coordinates and $(p_1, p_2, \ldots, p_d)$ as the conjugate momenta. The dynamics in the phase space are governed by the Hamilton equations.

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \tag{2.1}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \quad 1 \leq i \leq d. \tag{2.2}$$

Every point in phase space is well-defined. The evolution of any initial point $(q(0), p(0))$ to a point $(q(t), p(t))$ after time t is described by Hamilton's equations of motion.

There are a lot of classical systems whose dynamics are chaotic[10]. The examples are the double pendulum [11], Sinai's billiard [12] and so on. The chaotic behaviour arises from the nonlinear form of interaction between two or more components in the system. Classical chaos is defined as the sensitive dependence on the initial condition. In regular systems, smaller uncertainties in the initial condition do not have a profound impact on the final state of the system, whereas in a chaotic system, the effect is profound. These systems are non-integrable, and in this sense, their evolution becomes non-deterministic; that is, they are unpredictable due to the inability to determine the initial state with infinite precision.

The behaviour can be observed with the example of billiards in the Fig. 2.1 [1], illustrating the regular and chaotic trajectories in the case of the billiards.

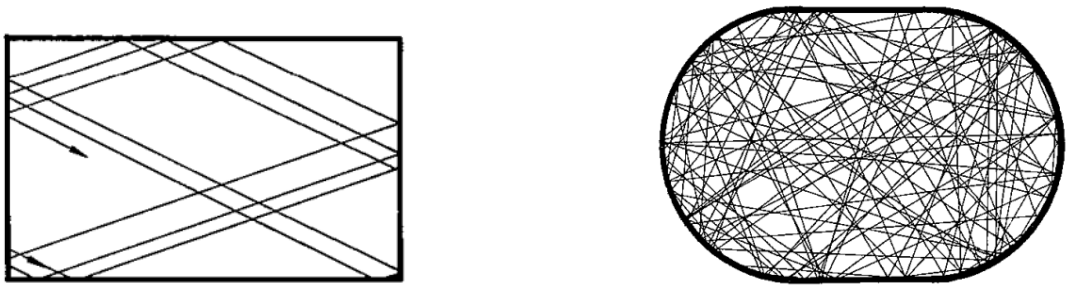
In classical chaotic systems, the small initial difference in their states in the classi-

cal phase space grows exponentially over time, and their trajectories diverge, and the Lyapunov exponent[13] decides the rate of divergence. The phase space trajectories and the Lyapunov exponent essentially describe the chaotic nature of the system.

Consider the case of two trajectories in phase space with initial separation vector $\delta\mathbf{X}_0$, the separation over time grows as given by the equation below.

$$|\delta\mathbf{X}(t)| \approx e^{\lambda t}|\delta\mathbf{X}_0| \tag{2.3}$$

where λ is the Lyapunov exponent. Systems with a positive Lyapunov exponent exhibit chaotic behaviour.



(a) Regular trajectory of rectangular billiard **(b)** Chaotic trajectory of Sinai billiard

Figure 2.1: Trajectory of particles in billiards taken from Ref.[1].

2.2 Quantum Chaos

The quantum dynamics of classically chaotic systems and their manifestations are a key area of focus, as it seeks to bridge the conceptual gap between classical chaos and its quantum mechanical manifestations. A primary objective is to determine how a system in the quantum limit mimics chaotic dynamics and to establish criteria such as spectral statistics and out-of-time-ordered correlators (OTOCs) to distinguish between regular and irregular behaviour.

Quantum chaos[14] studies the dynamics and properties of a quantum system in which its classical counterparts are capable of showing chaotic attributes. In quantum mechanics, what we have is Hilbert space in place of phase space, and instead of trajectories, we have evolving states. The classical context of chaos cannot be applied to describe quantum chaos, as the overlap between two states does not change since the evolution is unitary. For any two states at time $t = 0$, $|\psi_A(0)\rangle$ and $|\psi_B(0)\rangle$, after evolution at time t the inner product is,

$$\langle\psi_B(t)|\psi_A(t)\rangle = \langle\psi_B(0)|U^\dagger(t)U(t)|\psi_A(0)\rangle = \langle\psi_B(0)|\psi_A(0)\rangle \tag{2.4}$$

This proves that the overlap never changes because of the unitary evolution.

Thus, unlike classical chaos, the notion of sensitivity to initial conditions, typically characterised by exponential divergence of trajectories, does not directly apply in quantum systems due to their linear and unitary evolution. In quantum systems, those exhibiting regular characteristics show no level repulsion; the quasienergy levels tend to cluster and thus have no gaps. In contrast, chaotic quantum systems exhibit level repulsion. Thereby, chaotic systems follow Wigner–Dyson statistics, whereas regular systems show Poissonian statistics for their level spacing. In our work, the chaoticity of the model is analysed by looking at its spectral statistics. Thus, to see if a quantum system is regular or chaotic, we need to look at its spectral properties, such as level spacing statistics and eigenvector statistics. A quantum system is chaotic when its spectral properties are in agreement with those of the Random Matrix Theory (RMT). A comprehensive description of this is given in the section 2.4.

We have investigated periodically driven quantum systems, specifically the kicked Ising spin chain. This model serves as a foundational benchmark for understanding quantum chaos, providing the necessary framework for transitioning into generative modelling using variational quantum circuits.

2.3 Floquet Theory

Consider the time-dependent Hamiltonian, $H(t)$, which is periodic and thereby written as

$$H(t+\tau) = H(t) \tag{2.5}$$

where τ is the period of driving. The general Hamiltonian can be written as

$$H(t) = H_0 + V(t) \tag{2.6}$$

H_0 is the Hamiltonian of free evolution of the system, which could be atoms or nuclei, and $V(t)$ describes the coupling to the oscillating field, which is periodic, and is given as

$$V(t+\tau) = V(t) \tag{2.7}$$

In those cases of delta-kicked systems, which will be our topic of interest, the periodic potential $V(t)$ takes the form of a sequence of pulses, which leads to a stroboscopic map of the dynamics. Using the Floquet theory, now we can define the time evolution of the system in terms of the Floquet maps, which connect the state $\psi(x,t)$ to $\psi(x,t+\tau)$ at times t and $t+\tau$.

As can be observed, H is no longer time-independent. Since the Hamiltonian is

periodic, we can solve the corresponding time-dependent Schrodinger equation,

$$i\hbar\frac{\partial\psi}{\partial t} = H\psi \tag{2.8}$$

by separation of variables,

$$\psi_n(x,t) = \exp\left(-\frac{iE_nt}{\hbar}\right)\psi_n(x) \tag{2.9}$$

Thus, E is not a constant of motion. But H is still invariant under a discrete time shift τ. This makes the Hamiltonian commute with the shift operator T_τ, and thus we can find solutions $\psi_n(x,t)$ to the time-dependent Schrodinger equation since the Hamiltonian and T_τ can be simultaneously diagonalised. To sum up, we have solutions to $\psi_n(x,t)$ which are the simultaneous eigenfunctions of the Hamiltonian and the time-translation operator.

$$T_\tau\psi_n(x,t) = \psi_n(x,t+\tau) = \lambda_n\psi_n(x,t) \tag{2.10}$$

It becomes necessary for λ_n to be a pure phase factor for stationary solutions. Thus, we have

$$\psi_n(x,t+\tau) = e^{-i\phi_n}\psi_n(x,t) \tag{2.11}$$

$$\psi_n(x,t) = e^{-i\omega_n t}u_n(x,t) \tag{2.12}$$

where $u_n(x,t+\tau) = u_n(x,t)$ making it periodic and $\omega_n = \phi_n/\tau$, this is the Floquet theorem analogous to that of the Bloch theorem in the cases of periodic lattices.

The quasi-energy,

$$E_n = \hbar\omega_n = \frac{\hbar\phi_n}{\tau} \tag{2.13}$$

Similar to the Brillouin zone in solid-state physics, these quasi-energies are defined only within a range of integer multiples of h/τ, with values outside this interval being physically redundant. This is because the Floquet phases ϕ_n are defined only up to the integer multiples of 2π.

Now, let's look at the time evolution operator $U(t)$.

$$\psi(x,t) = U(t)\psi(x,0) \tag{2.14}$$

Plugging this onto the time-dependent Schrodinger eq(2.8) we get,

$$i\hbar\frac{dU}{dt} = HU \tag{2.15}$$

H being hermitian we see that U is unitary. This has been proved here. Adjoint of

eq.(2.15) is given by,

$$-i\hbar\frac{dU^\dagger}{dt} = U^\dagger H \tag{2.16}$$

Multiplying $U^\dagger$ from left to eq.(2.15) and U from right to eq.(2.16) on both sides and then taking the difference, we get

$$\frac{d(U^\dagger U)}{dt} = 0 \tag{2.17}$$

which essentially means $U^\dagger U$ is a constant and putting the initial condition $U(0)$=1 implies

$$U^\dagger U = 1 \tag{2.18}$$

Hence, we have the equation of the form,

$$\psi(x,\tau) = U(\tau)\psi(x,0) \tag{2.19}$$

This can be generalised as given below

$$\psi(x,n\tau) = [U(\tau)]^n\psi(x,0) \tag{2.20}$$

and this gives us the time-evolution operator as

$$U(n\tau) = [U(\tau)]^n \tag{2.21}$$

The unitarity of $U(\tau)$ ensures that it can be diagonalised. This operator is called the Floquet operator. By performing a unitary transformation, we obtain the Floquet phases, which lie on the unit circle in the complex plane and are used to determine the dynamics of Floquet systems. [14]

$$(U_D)_{nn} = e^{-i\phi_n} \tag{2.22}$$

Since we will be looking at kicked systems, the Hamiltonian for those systems is described as

$$H(t) = H_0 + V_0\sum_{n=-\infty}^{\infty}\delta(t-n\tau) \tag{2.23}$$

Given that the Hamiltonian $H(t)$ remains constant over distinct intervals, the time-evolution operator is obtained by the direct integration method and in the limit of $\Delta\tau \to 0$, we get the form of the Floquet unitary,

$$U = \exp\left(\frac{-iV_0}{\hbar}\right)\exp\left(\frac{-iH_0\tau}{\hbar}\right) \tag{2.24}$$

We will study the kicked tops in section 2.5, which are one of the popular classes of

these systems.

2.4 Spectral Statistics

In the previous section of Floquet theory, we defined the Floquet operator and Floquet phases. Hereafter, we will focus on quantum systems with U as the time evolution operator, describing the dynamics as

$$U|\psi_n\rangle = e^{i\phi_n}|\psi_n\rangle \tag{2.25}$$

with $n = 1, 2, 3....., N$, where N denotes the dimension of the Hilbert space. The eigenstates ϕ_n are normalised, thereby the eigenphases $|\psi_n\rangle$ are constrained to the range $[-\pi, \pi)$. We analyse the statistics of the phases $(\phi_1, \phi_2, \phi_3, ..., \phi_N)$ sorted in ascending order. The statistical properties for a system to be chaotic in nature are expected to follow the universal predictions from Random Matrix Theory. Those systems with time-reversal symmetry obey the results of the Gaussian Orthogonal Ensemble(GOE), whereas those without time-reversal symmetry obey the Gaussian Unitary Ensemble(GUE)[15]. Since we will only be studying systems with time-reversal invariance, we restrict our discussion to those of the GOE.

2.4.1 Level Spacing Statistics

An easy way to look at the spectral statistics is to analyse the spacing between consecutive levels. The spacing from the eigenphases is given as

$$s_n = \frac{N}{2\pi}(\phi_{n+1} - \phi_n) \tag{2.26}$$

where the factor $N/2\pi$ is for unfolding, thus ensuring the mean spacing to be 1. The Wigner distribution describes the distribution [16] in the limit where $N \to \infty$, which is given by

$$P_{\text{GOE}}(s) \approx \frac{\pi}{2} s \exp\left(-\frac{\pi}{4}s^2\right) \tag{2.27}$$

The probability of zero spacing between adjacent levels is negligible, exhibiting level repulsion.

In addition to the level spacing statistics, one can also look at the adjacent spacing ratio statistics of the levels [17]. Ratios, $\tilde{r}_n$ with $r_n = s_n/s_{n-1}$ are given as

$$\tilde{r}_n = \frac{\min(s_n, s_{n-1})}{\max(s_n, s_{n-1})} = \min\left(r_n, \frac{1}{r_n}\right) \tag{2.28}$$

From the prediction of the distribution for the spacing ratio, a chaotic system is

expected to have

$$\langle \tilde{r} \rangle_{\text{GOE}} \approx 0.5359 \tag{2.29}$$

While regular systems show Poissonian level statistics such that

$$\langle \tilde{r} \rangle_{\text{Poisson}} \approx 0.3863 \tag{2.30}$$

One can refer to Ref.[18] for an elaborate analytical derivation.

2.4.2 Eigenvector statistics

The next statistical property we will discuss concerns the eigenvectors. Consider the $\{|m\rangle\}_{m=1,\ldots,N}$ as the orthonormal basis of the eigenstates. We can then write $|\psi_n\rangle$ as

$$|\psi_n\rangle = \sum_{m=1}^{N} c_m^{(n)} |m\rangle \tag{2.31}$$

and the coefficients are given as

$$c_m^{(n)} = \langle m|\psi_n\rangle \tag{2.32}$$

it is normalised such that

$$\sum_m |c_m^{(n)}|^2 = 1 \tag{2.33}$$

Defining

$$\eta_{nm} = N|c_m^{(n)}|^2 \tag{2.34}$$

where N is for normalisation. We look at the distribution of η_{nm} given by $P(\eta)$ to determine if the system is chaotic or regular. A chaotic system with time-reversal invariance, it shows Porter-Thomas distribution[19] in the limit $N \to \infty$.

$$P_{\text{GOE}}(\eta) = \frac{\exp(-\eta/2)}{\sqrt{2\pi\eta}} \tag{2.35}$$

.

2.5 Kicked Top Model

This section discusses a top that precesses in a magnetic field and periodically experiences impulsive nonlinear kicks. The system is three-dimensional, with the components of the angular momentum J, (J_x, J_y, J_z) as the dynamic variables. From the

angular momentum commutators we have,

$$[J_i, J_j] = i\epsilon_{ijk} J_k \tag{2.36}$$

The dynamics of the top are such that the square of J is a constant of motion.

$$[J^2, H] = 0 \tag{2.37}$$

What this means quantum mechanically is that we have an operator J^2 which is conserved. Since J^2 AND H commute, they are simultaneously diagonalisable. The eigenvalues of the operator J^2 are

$$J^2 = j(j+1) \quad , \quad j = \frac{1}{2}, 1, \frac{3}{2}, \dots \tag{2.38}$$

Thus, the dimension of the Hilbert space is finite, which is $2j+1$. Putting $N = 2j$, the dimension becomes $N+1$ .The Hamiltonian chosen is such that the quantum behaviour approaches the classical behaviour in the limit $j \to \infty$. We can simplify the dynamics by adopting a stroboscopic view, where we focus exclusively on the state of the system at the moment of each pulse. The operator we analyse numerically is the Floquet operator U.

2.5.1 Quantum Dynamics

The Hamiltonian of the system[20] is

$$H(t) = \frac{\hbar p}{\tau} J_y + \frac{\hbar k}{2j} J_z^2 \sum_{n=-\infty}^{+\infty} \delta(t - n\tau) \tag{2.39}$$

The first term in the Hamiltonian denotes the precession around the y-axis with p/τ as its angular frequency, and the second term describes the impulsive kicks at a temporal distance of τ and kick strength k/j, which is dimensionless. This can be viewed as impulsive rotations around the z axis at an angle proportional to J_z.

The Floquet unitary of this system U is

$$U = e^{-i(k/2j)J_z^2} e^{-ipJ_y} \tag{2.40}$$

The initial state $|\psi_0\rangle$ is chosen as the spin-coherent state given as,

$$|\psi_0\rangle = |\theta, \phi\rangle = \left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right)^{\otimes N} \tag{2.41}$$

The coherent state is the minimum-uncertainty state and thus is very similar to that

of its classical counterpart. Thus, a comparison between the quantum and classical behaviour of the model can be done in an efficient way [21]. Setting $\tau = 1$ and $\hbar = 1$ makes the effective Planck's constant as $1/j$.

In the Schrodinger picture, we have the state evolving at n-time steps as

$$|\psi_n\rangle = U^n|\psi_0\rangle, \quad n = 0, 1, 2, \ldots \tag{2.42}$$

But we are interested in looking at the Heisenberg picture. The Heisenberg evolution of the angular momentum operator J_i where $i \in \{x, y, z\}$ after n periods is given as,

$$J_{in} = U^{\dagger n} J_i U^n \tag{2.43}$$

for the discrete time evolution.

As mentioned earlier, we look at the spectral statistics of the eigenphases of U and learn the nature of the system. By exact diagonalisation of the time-evolution operator U over a one-time step, we obtain the eigenvalues and eigenstates. Using the eigenvalues, we look at the spectral statistics such as level-spacing and the mean spacing ratio as described before in section 2.4.

The Hamiltonian of the kicked top has a discrete symmetry under the parity operator,

$$\hat{\Pi} = (-1)^{j+J_y} \tag{2.44}$$

Thus, the Floquet unitary commutes with the parity operator further

$$[\hat{F}, \hat{\Pi}] = 0. \tag{2.45}$$

Since the parity quantum number is conserved, the Hilbert space decomposes into two independent invariant subspaces corresponding to the eigenvalues ± 1 of the parity operator. The dynamics within each subspace evolve independently, and therefore, we analyse the spectral statistics separately for each parity sector.

Here we have set the precession angle $p = \pi/3$ and $j = 800$. The full Hilbert space has a dimension of 1601, but after parity separation, the two subspaces have dimensions of 801 and 800 states, respectively. The following results are for the $+1$ parity sector, which had a dimension of 801.

Fig.2.2 depicts the values of the mean spacing ratio over different values of k. We can observe the clear transition from integrable to a chaotic regime.

For $k < 5$, the system shows near-integrable behaviour with $\langle r \rangle$ close to the Poisson prediction, which is $\langle r \rangle = 0.386$. When k increases above 5, there is a rapid transition to the chaotic regime, where $\langle r \rangle$ saturates approximately 0.53. This value is in agreement with that of the GOE prediction characterising chaotic systems with

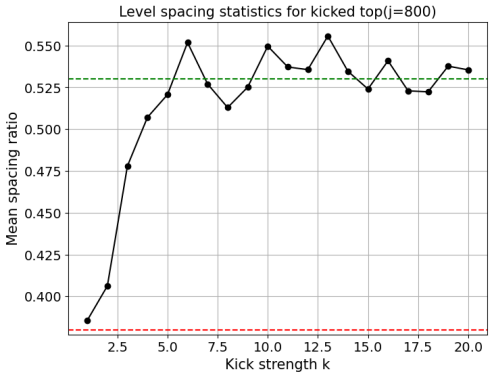


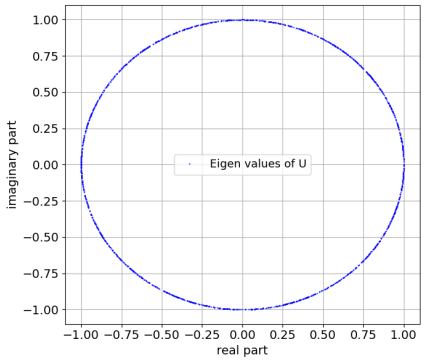
Figure 2.2: Mean level spacing ratio $\langle r \rangle$ as a function of the kick strength k of a kicked-top system with $j = 800$.

time-reversal symmetry. Small deviation from GOE is seen due to the finite-size effects inherent to the finite-dimensional Hilbert space.

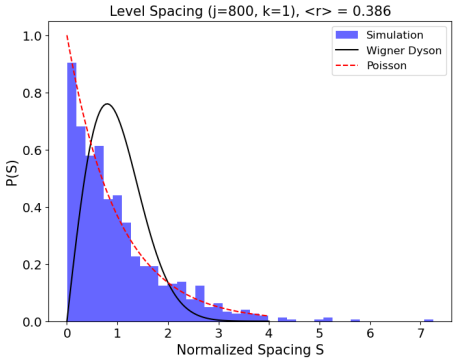
Fig.2.3 shows how the kicked top system changes its behaviour as we increase the kick strength from k = 1 to k = 6. We're looking at the same system but at two different kick strengths. Since the Floquet operator is unitary, all its eigenvalues must sit on the unit circle in the complex plane, having unit magnitude but different phases. This is what is shown in the plots (a) and (c). The plots (b) and (d) show histograms of the spacings between neighbouring eigenvalues, which tell us whether the system is chaotic or not.

At $k = 1$, the spacing distribution looks exponential, with lots of small spacings, exhibiting eigenphases clustering. The mean ratio $\langle r \rangle \approx 0.386$ exactly as the GOE statistics, which is what we'd expect for a Poisson distribution (the signature of regular, non-chaotic systems). The red-dashed curve shows this Poisson prediction, and our data fits it reasonably well. At $k = 6$, things look very different. The distribution has almost no probability near eigenphases with zero spacing, exhibiting level repulsion. Instead, we see a peak around $s \approx 1$ and then a tail-off. This matches the Wigner-Dyson distribution (black curve), which is the hallmark of quantum chaos. The mean ratio $\langle r \rangle \approx 0.552$ is very close to the GOE prediction of 0.53.

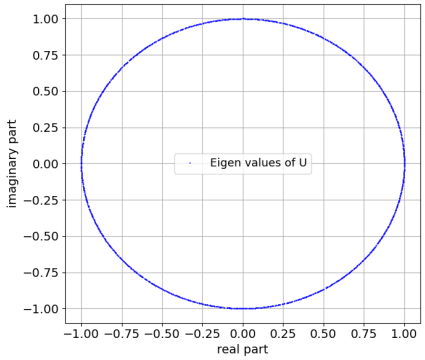
This transition in the spacing statistics from exponential clustering to level repulsion is one of the clearest signatures that our system has gone from regular motion to chaos.



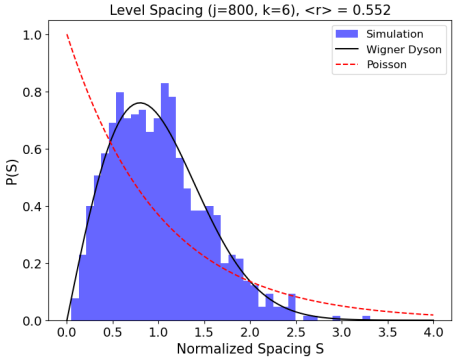
(a) Eigenvalues of U for $k = 1$



(b) Level spacing for $k = 1$



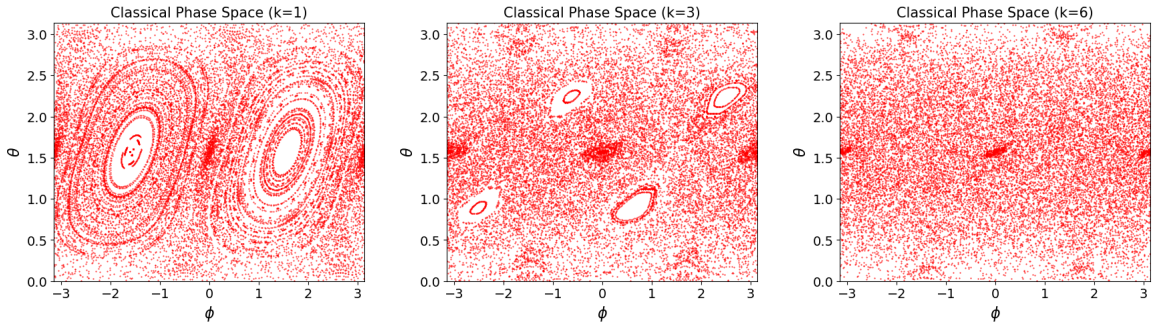
(c) Eigenvalues of U for $k = 6$



(d) Level spacing for $k = 6$

Figure 2.3: Comparison of the kicked top at weak, $k = 1$ and strong,$k = 6$ kick strengths. Left: eigenvalues on the unit circle. Right: Level spacing distribution for the consecutive kick strengths. The black curve is the analytical curve for the GOE level spacing statistics. The red dashed curve is the analytical curve for the Poissonian level spacing distribution.

2.5.2 Classical Dynamics



(a) Phase space for $k = 1$ **(b)** Phase space for $k = 3$ **(c)** Phase space for $k = 6$

Figure 2.4: Kicked top classical phase space trajectories for different kick strengths k, illustrating the regular to chaotic regime transition.

The Heisenberg evolution of the operator J was given in Eq.2.43. At the classical limit obtained by $j \to \infty$, which essentially means $N \to \infty$ and by rescaling the angular momenta using the following relations,

$$X = \frac{J_x}{j}, \quad Y = \frac{J_y}{j}, \quad Z = \frac{J_z}{j} \tag{2.46}$$

we get the equations of classical dynamics[22],

$$X_{n+1} = Z_n, \tag{2.47}$$

$$Y_{n+1} = Y_n \cos(kZ_n) + X_n \sin(kZ_n), \tag{2.48}$$

$$Z_{n+1} = -X_n \cos(kZ_n) + Y_n \sin(kZ_n). \tag{2.49}$$

The Fig.2.4 shows how the classical phase space of the kicked top evolves as we increase the chaos parameter k. We're plotting using spherical coordinates, $\theta = \cos^{-1}(Z)$, the polar angle and $\phi = \tan^{-1}(Y/X)$,the azimuthal angle. It can be observed that as the value of k is increased, the system shows a transition from regular to chaotic regime.

When $k = 1$, the phase space looks regular, indicating a regular motion because well-defined periodic orbits tracing out smooth curves can be observed. The system is predictable and orderly, which is the characteristic of a regular system. For $k = 3$, we see a mixed regime, that is, there are still some islands of regular behaviour, but they're surrounded by a sea of chaotic trajectories. It's like the system is in the intermediate regime. When $k = 6$, the phase space is completely chaotic; those orderly structures have vanished, and the trajectories now fill the space densely. There's no trace of the regular motion as we saw at k = 1.

2.6 Quantum Machine Learning

Classical Machine Learning algorithms[23] are effective at finding complex patterns in the data. Since quantum systems are naturally able to produce complex patterns, we can leverage them for identifying unknown patterns in data, expecting that it renders additional advantages and show more efficiency than the classical approach. This idea forms the basis of *QuantumMachineLearning* (QML), an emerging field that combines concepts from quantum computing and machine learning.

This integration of quantum computing and machine learning has been done in four approaches, which are shown in Fig. 2.5. They include processing classical data

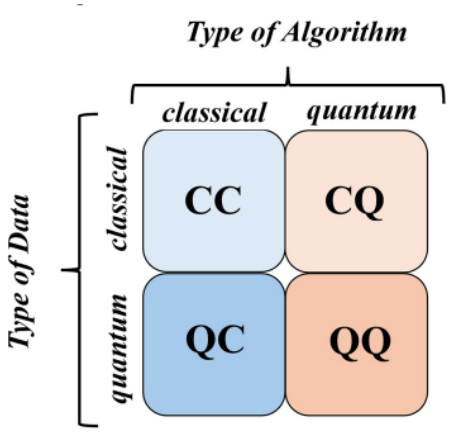


Figure 2.5: Four Approaches of Integrating Quantum Computing and Machine Learning, taken from Ref.[2]

using quantum algorithms as well as processing data from a quantum system using classical algorithms.

In our work, we have tried the former approach.

The primary question in this area is whether using quantum computation is really possible to attain any advantage in speed or accuracy over classical methods. Achieving this advantage is known as *quantum speedup*. More broadly, the term *quantum advantage*[24] describes a scenario where the quantum device performs the computation, and it outperforms its classical counterpart. They could be intractable for classical systems due to the limitations of memory, speed and so on.

Multiple quantum algorithms, such as Grover's Search Algorithm[25], Shor's Algorithm[26], Quantum Fourier Transform[27], have been theoretically proven to show this quantum speedup[28]. However, the major drawback being that all these require millions of qubits, long coherence time, error-resilience and so on for the advantage to be observed physically.

Thus, with these limitations, a purely quantum approach becomes difficult to implement effectively. As a result, current NISQ devices rely on hybrid classical–quantum frameworks, where both classical and quantum components are integrated to har-

ness the strengths of quantum computation while simultaneously utilising classical resources for tasks such as optimisation, control, and error mitigation. This is done by performing the part of the calculation that would be classically challenging on a quantum device, and the rest is performed on a classical computer. This gives the near-term advantages of using a quantum framework. Variational quantum algorithms (VQA), which are shallow parametrised circuits, are used for this purpose. In the following section, it has been discussed in detail.

2.6.1 Variational Quantum Algorithms(VQA)

Variational Quantum Algorithms (VQAs), also known as parametrised circuits and quantum neural networks, are machine learning models. It is a class of hybrid quantum-classical algorithms. They are a promising candidate to show quantum advantage and are designed to work on NISQ devices.

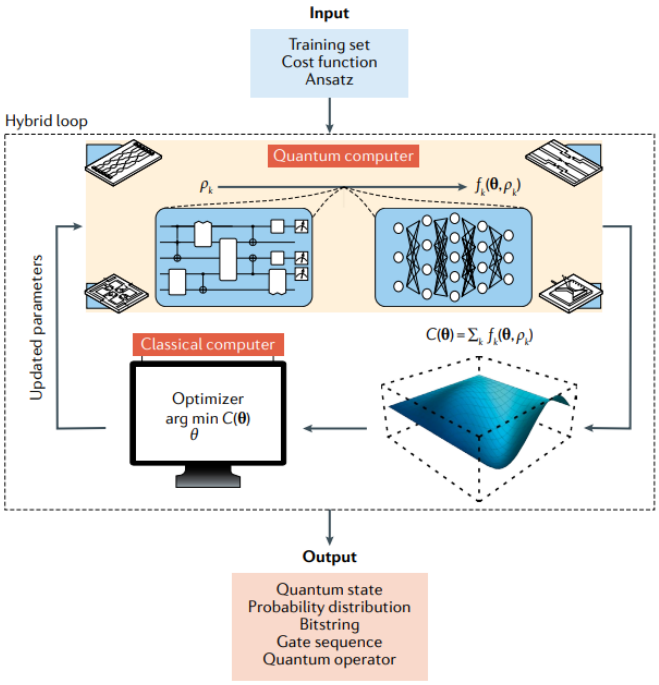


Figure 2.6: Variational Quantum Algorithm (VQA) workflow, taken from Ref.[3].

Figure 2.6 depicts the schematic of VQA. The workflow involves a parametrised quantum circuit, which is otherwise called an ansatz with variable parameters that are updated iteratively by the classical optimiser. This is done to minimise the cost function, $C(\theta)$ with θ as the set of trainable parameters, which encodes the quality measure of the model. $\{\rho_k\}$ is the set of training data used for optimisation. As we are using it for generative tasks, the objective is to minimise the divergence between the model's distribution and the target distribution of the dataset.

Let's discuss the building blocks of VQA much more elaborately.

Cost Function

The very first step involved is defining the cost function $C(\theta)$, which is also known as the loss function. Thereby, the problem is encoded into this cost function. Depending on the task, the ansatz is trained to either minimise or maximise the cost function over the trainable parameters θ.

In general, the cost function is defined as,

$$C(\theta) = f(\{\rho_k\}, \{O_k\}, U(\theta)) \tag{2.50}$$

f denotes some function, $U(\theta)$ is the unitary of the parametrised circuit, θ the trainable parameters, $\{\rho_k\}$ are the training input states and $\{O_k\}$ is the set of observables. It is the task in hand that determines the form of f. Certain cost functions aim to find the ground state energy. In those cases, the observable becomes the system's Hamiltonian H, and the cost function is the expectation value of H. The Hamiltonian is implemented as the parametrised quantum circuit, thereby the cost function is,

$$C(\theta) = \langle H \rangle_\theta \equiv \langle 0|U^\dagger(\theta) H U(\theta)|0\rangle \tag{2.51}$$

The task is to reduce this cost to find the minimum energy state by varying θ appropriately. Here, we start with the initial state $|0\rangle$.

Ansatz

The ansatz determines the trainable parameter θ. The form of ansatz is dependent on the task at hand, called as the problem-inspired ansatz. When no relevant information about the task is available, then problem-agnostic ansatzes are adopted where the ansatz architecture becomes generic. The quantum circuit used to set up the wave function intended to optimise is known as the parametrised quantum circuit (PQC) or the variational quantum circuit (VQC). The trainable parameters are encoded into the unitary of the PQC given as $U(\theta)$. Generally, the PQC is composed of single-qubit gates and entangling gates with trainable parameters and thus $U(\theta)$ can be written as a sequentially applied L unitaries,

$$U(\theta) = U_L(\theta_L) \dots U_2(\theta_2) U_1(\theta_1). \tag{2.52}$$

Optimisation

Once we have defined the cost function $C(\theta)$ and the corresponding ansatz $U(\theta)$, the next step is to optimise over these trainable parameters θ. The quantum part of the algorithm does the work of evaluating the cost function, which is then taken for optimisation by a classical optimiser, thereby updating the parameters iteratively.

This can be written as,

$$\theta^* = \arg\min_{\theta} C(\theta) \tag{2.53}$$

Measurement output that is obtained from the PQC, $U(\theta)|0\rangle$, allows for the calculation of $C(\theta)$. This information is then the input for the classical component of the optimisation loop, which explores the landscape of the cost function for the best set of parameters. Generally, gradient-based methods are used for finding the minimum loss with the parameter update rule given as,

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta C(\theta^{(t)}) \tag{2.54}$$

with η as the learning rate and $\nabla_\theta C(\theta^{(t)}$ the gradient of the cost function at iteration t.

In our work, we have employed a gradient-free optimisation method. This is preferred as they are a lot more resistant to noise and errors, and also can be employed to tackle the issues of barren plateaus. Though they may require more circuit evaluations.

2.6.2 Quantum Circuit Born Machine(QCBM)

QCBM in unsupervised generative modelling is used to learn and represent classical probability distributions using quantum pure states[29]. Thus, samples can be generated by the projective measurement on the qubits. The measurement is done in the computational basis, and probabilities are governed by the Born rule. The circuit parameters are trained to minimise the distance and thereby approximate the quantum-generated distribution and the target classical distribution. Here, we can see that born machines leverage the probabilistic nature of quantum states. Unlike classical generative models, the probability distribution is thereby encoded in the probability distribution of the quantum states, which are generated using parametrised quantum circuits [30]. It is the high expressive power of such models that made them so popular.

Let $\mathcal{D} = \{x\}$ be a classical dataset containing samples which are drawn independently from an unknown distribution, and this distribution is our target distribution $\pi(x)$. We aim to use quantum circuits to learn this target distribution and use it to generate samples similar to those of the target distribution.

The QCBM works such that it starts with qubits in a known initial state $|0\rangle$ and repeatedly applies parameterised quantum gates, which are trained. This produces a final quantum state $|\psi_\theta\rangle$. When measured in the computational basis, it yields

bitstrings x with probability $p_\theta(x)$ given by the Born rule,

$$p_\theta(x) = |\langle x|\psi_\theta\rangle|^2. \tag{2.55}$$

2.6.3 Kernel Methods

The optimisation process is performed over the parameters θ and requires a cost function that quantifies how well a given set of parameters reproduces the target distribution. The choice of this cost function is of utmost importance, as it directly influences the training performance. Kernel methods are based on the idea of defining a similarity measure between data points[31].

Kernel method functions by mapping data points to a high-dimensional feature space, but implicitly. Instead of computing these transformations $\phi(x)$, in this method, the mapping can always be defined as a kernel function $K(x, y)$ that directly computes the inner product in the feature space:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle \tag{2.56}$$

This is the kernel trick. Thus, we can work in potentially infinite-dimensional spaces without the need to explicitly construct the feature vectors, making the computations tractable. We have used a Gaussian kernel for our purpose, which will be discussed further.

Chapter 3

Kicked Ising Spin Chain Ansatz for Quantum Generative Modelling

The work carried out as part of this thesis was mostly computational in nature. This chapter gives a comprehensive explanation of the model employed, the numerical implementation techniques along with the frameworks employed to assess the model's performance.

3.1 Model- Kicked Ising spin chain

The many-body system considered in our work is that of a kicked Ising spin chain with open boundary conditions, where the kicks are the magnetic fields in a direction that is random and site-dependent. The Hamiltonian of the system [32], which is time-dependent, is given by,

$$H = H_{\text{free}} + H_{\text{kick}} \sum_{n=-\infty}^{\infty} \delta(t - n\tau), \tag{3.1}$$

$$H_{\text{free}} = J \sum_{i=1}^{L-1} \sigma_i^z \sigma_{i+1}^z, \tag{3.2}$$

$$H_{\text{kick}} = B \sum_{i=1}^{L} \left(\cos(\Theta_i)\sigma_i^x + \sin(\Theta_i)\sigma_i^z\right). \tag{3.3}$$

where L is the total number of spins, and thus the Hilbert space dimension is 2^L. σ_i^z and σ_i^x are the Pauli spin operators. H_{free} is the free evolution Hamiltonian, which encodes the nearest-neighbour coupling along the z direction with coupling strength of J. H_{kick} is the kick part Hamiltonian, where B is the magnetic field strength applied periodically, which is written as the sum over δ functions. The direction at which the magnetic field acts on the spins is determined by the tilt angle Θ_i for

each spin along the $x - z$ plane. This model has time-reversal symmetry. The tilt angle is site-dependent and is drawn from a uniform distribution ranging from $[0, 2\pi]$. These site-dependent tilt angles are chosen to break the external reflection symmetry inherent to the system. In contrast to the kicked-top model, where the dynamics are restricted to a reduced effective Hilbert space, this kind of disorder ensures that the system can explore the full Hilbert space of dimension 2^L.

The time-evolution operator of this periodic Hamiltonian, that is, the Floquet unitary[5], is

$$U = e^{-iH_{free}/2} e^{-H_{kick}} e^{-iH_{free}/2} \tag{3.4}$$

The eigenstates of U have real coefficients in the computational basis.

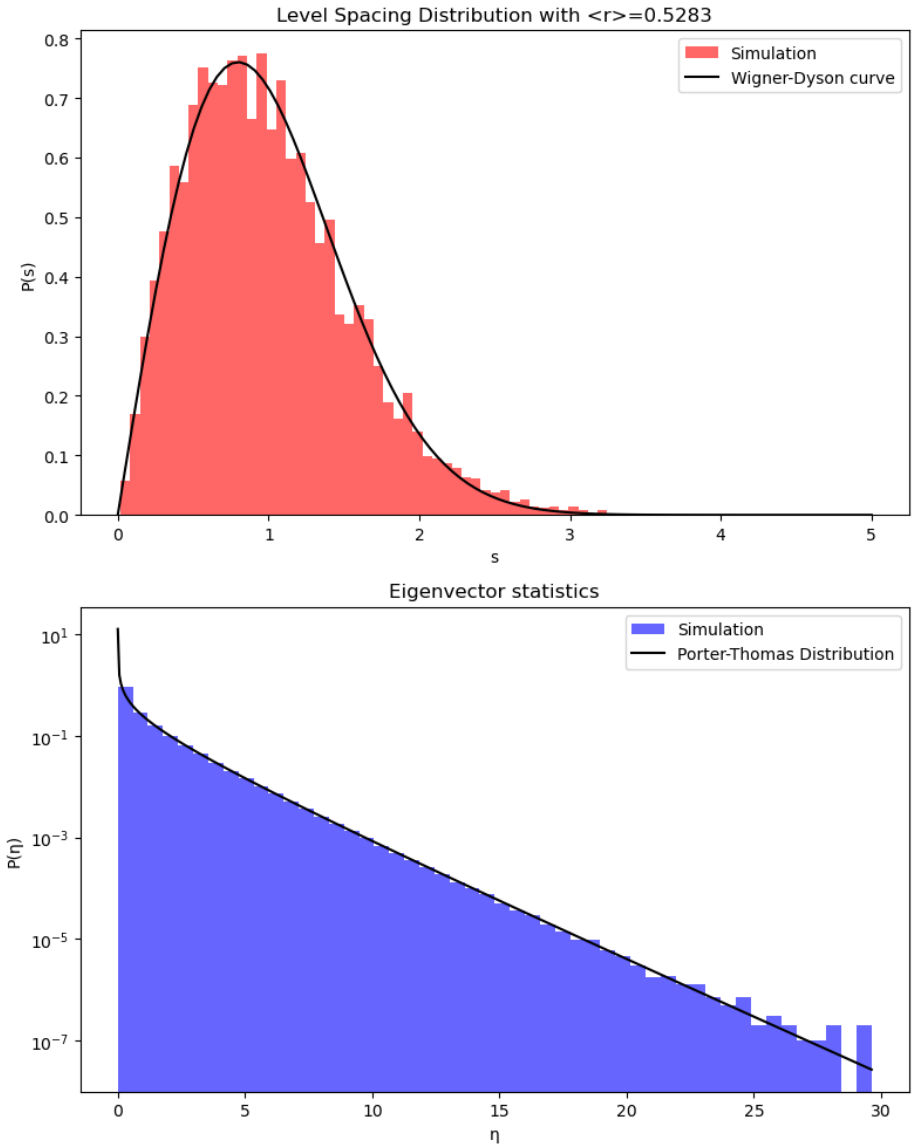


Figure 3.1: Level spacing statistics(top) and eigenvector statistics(bottom) plots for the kicked Ising spin chain of $L = 12$. The black curves in both plots show the theoretical GOE curve for level spacing and eigenvector statistics, respectively.

Fig.3.1 shows the results of numerical computation of the level spacing and eigenvector distribution for the parameter set given in Table 3.1

L	J	B	$\{\Theta_i\}$
12	0.8	1.35	$\{7,7,7,7,8,8,8,8,8,8,7,7\}\pi/32$

Table 3.1: Parameters for the kicked Ising spin chain picked from Ref.[5].

Both plots are in good agreement with the GOE predictions. The mean spacing ratio, $\langle r \rangle$ observed is 0.528 and thus qualifies as quantum chaotic. The $P(\eta)$ histogram is shown in semi-logarithmic representation and exhibits Porter-Thomas distribution.

This is done to demonstrate that the system exhibits chaotic characteristics. The set of Θ_i was chosen from the Ref.[5].

The purpose of this work is to investigate the potential of a quantum many-body system as a generative model. The kicked Ising spin chain model has been used for this. We are attempting to construct the VQAs using the Floquet unitary of this system to explore if they can be used as generative models. Thereby, here we are replacing the standard randomised layers of the variational quantum circuits with the Floquet unitary $U = e^{-iH_{free}/2}e^{-H_{kick}}e^{-iH_{free}/2}$ of the kicked Ising spin chain. Thus, we have a chaos-inspired ansatz. Unlike random circuits, this operator has inherent structure as they are made of deterministic local and non-local interactions and thus can encode correlations in the distribution and show complex scrambling dynamics without becoming fully random unitary designs.

3.2 Loss function-MMD Loss

Loss functions are necessary for choosing the best model from the model family. It is used to measure the quality of the model. Those models with the least loss denote that the model predictions and the target distribution are close. The type of loss function selected depends on the type of generative model being used and the data distribution given for the model to learn. This makes it very task-specific. For our work, we analyse the Maximum Mean Discrepancy (MMD) loss[33].

Maximum Mean Discrepancy (MMD) loss has been widely utilised for training Born Machines [34]. MMD loss quantifies the divergence between the target distribution $\pi(x)$ and the model-encoded distribution $p_\theta(x)$ by comparing their respective mean embeddings within the feature space. In practice, an empirical estimate of the MMD loss is calculated using finite samples from both distributions. In our case, since we are dealing with a small Hilbert space and we have access to the probability distribution, we have employed an analytic evaluation.

3.2.1 Feature Means

Given a random variable x, $\phi(x)$ is the feature map which maps x to feature space $\mathcal{F}$, which is the Reproducing Kernel Hilbert Space (RKHS). Using the kernel trick as mentioned in 2.56, we can compute the feature space inner product as

$$k(x, y) = \langle \phi(x), \phi(y) \rangle \tag{3.5}$$

where x,y are the random variables and $k(x, y)$ is the kernel function.

The feature means, which is also called as mean embedding, is another feature map where $\phi(x)$ is mapped to the means of every coordinate of $\phi(x)$ given the probability measure P on x. The mean embedding,

$$\mu_P(\phi(x)) = [\mathbb{E}[\phi(x_1], \ldots, \mathbb{E}[\phi(x_m]]^T \tag{3.6}$$

$\mathbb{E}$ denotes the expectation value.

Thus, the kernel function inner product becomes

$$\langle \mu_P(\phi(x)), \mu_Q(\phi(y)) \rangle_{\mathcal{F}} = \mathbb{E}_{P,Q}[\langle \phi(x), \phi(y) \rangle_{\mathcal{F}}] = \mathbb{E}_{P,Q}[k(x, y)] \tag{3.7}$$

where P and Q are the underlying probability distribution of x and y respectively and

$$\mu_P = \mathbb{E}_{x \sim P}[\phi(x)], \quad \mu_Q = \mathbb{E}_{y \sim Q}[\phi(y)] \tag{3.8}$$

defined as the mean embeddings.

In this work, the probability measures P and Q correspond to the model distribution $p_\theta(x)$ and the target distribution $\pi(x)$, respectively. The random variables x and y represent the computational basis states sampled from these distributions.

3.2.2 Maximum Mean Discrepancy

As mentioned, MMD is the distance between the model distribution P and the target distribution Q by comparing their feature means in feature space. The squared MMD loss[35] is defined as,

$$MMD^2(P, Q) = \|\mu_P(\phi(x)) - \mu_Q(\phi(y))\|^2_{\mathcal{F}} \tag{3.9}$$

Expanding the norm in Eq.3.9 we get,

$$MMD^2(P, Q) = \|\mu_P - \mu_Q\|^2 \tag{3.10}$$

$$= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle \tag{3.11}$$

using inner products and then expanding,

$$= \langle \mu_P, \mu_P \rangle - 2\langle \mu_P, \mu_Q \rangle + \langle \mu_Q, \mu_Q \rangle \tag{3.12}$$

Computing each term:

$$\langle \mu_P, \mu_P \rangle = \langle \mathbb{E}_x[\phi(x)], \mathbb{E}_{x'}[\phi(x')] \rangle \tag{3.13}$$

$$= \mathbb{E}_{x,x'}[\langle \phi(x), \phi(x') \rangle] \tag{3.14}$$

$$\mathbb{E}_{x,x'}[k(x, x')] \tag{3.15}$$

Similarly, we have,

$$\langle \mu_P, \mu_Q \rangle = \mathbb{E}_{x,y}[k(x, y)] \tag{3.16}$$

and

$$\langle \mu_Q, \mu_Q \rangle = \mathbb{E}_{y,y'}[k(y, y')]. \tag{3.17}$$

Thus,

$$MMD^2(P, Q) = \mathbb{E}_{x,x' \sim P} k(x, x') + \mathbb{E}_{y,y' \sim Q} k(x, y') - 2\mathbb{E}_{x \sim P, y \sim Q} k(x, y) \tag{3.18}$$

Intuitively, MMD measures the distance between two probability distributions by comparing their representations in a high-dimensional feature space. If the two distributions are identical, their mean embeddings coincide, resulting in zero MMD.

Here, P corresponds to the target distribution $\pi(x)$, while Q corresponds to the model distribution $p_\theta(x)$. Therefore, our loss is given by,

$$\mathcal{L}_{MMD} = \mathbb{E}_{x,x' \sim P_\theta} k(x, x') + \mathbb{E}_{y,y' \sim \pi} k(y, y') - 2\mathbb{E}_{x \sim p_\theta, y \sim \pi} k(x, y) \tag{3.19}$$

For our implementation, we have rather used the analytic form of the MMD loss. This was to attain a much more deterministic and noise-free loss. This method allows a precise evaluation of the model and the target distribution.

Thus, we have,

$$\mathcal{L}_{MMD} = p^T K p + \pi^T K \pi - 2p^T K \pi \tag{3.20}$$

with K as the kernel matrix given by,

$$K_{ij} = k(x_i, x_j) \tag{3.21}$$

This analytic approach is computationally tractable for our system size ($n = 10$), as the kernel matrix K remains of size 1024×1024. This eliminates the statistical noise associated with finite sampling, providing a smoother gradient for the Born Machine's optimisation. The kernel used for our analysis is the Gaussian Mixture Kernel given

as,

$$k(x, y) = \frac{1}{c} \sum_{j=1}^{c} \exp\left(-\frac{|x - y|^2}{2\sigma_j^2}\right). \tag{3.22}$$

where c denotes the number of radial basis functions. To improve robustness, the kernel is averaged over multiple bandwidth parameters. By employing a mixture of kernels, we ensure that the MMD is sensitive to both local structural differences and the global shift between $p_\theta(x)$ and $\pi(x)$.

3.3 Implementation

In this section, we detail the model and kernel specifications, the encoding method used, the loss computation, and the computational methods used for simulation.

3.3.1 Basis Encoding

The encoding of classical data into quantum states for our purpose is done using basis encoding. It is one of the elementary methods used widely. In this technique, the bitstring of the classical data $x = x_1 \dots x_n$ is mapped onto the computational basis states $|x\rangle = |x_1 \dots x_n\rangle$. It can be seen that n qubits are necessary for representing classical data of n bits. For example, the classical bitstring (1010) is encoded as the quantum state $|1010\rangle$ where at least 4 qubits are required to represent it. Alternatively, it can also be extended to larger systems of m-qubits ($m > n$) by padding the remaining $(m - n)$ qubits with the $|0\rangle$ state, similar to padding with zeros in the bitstring.

The entire classical dataset can be encoded into a quantum superposition in the computational basis as given below,

$$|\mathcal{D}\rangle = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} |x^{(k)}\rangle, \tag{3.23}$$

where $\mathcal{D}$ denotes the classical dataset being encoded into quantum superposition state $|\mathcal{D}\rangle$. Each input binary string $x^{(k)} = (b_1, \dots, b_m)$ with $b_i \in 0, 1$ for $i = 1, \dots, m$ can be directly mapped to the quantum state $|x^{(k)}\rangle$. Here K denotes the number of samples in the input dataset, and m is the feature size.

The basis embedding of $\mathcal{D}$ will be sparse if the number of data points K is significantly smaller than the 2^m available basis states in the m-bit Hilbert space.

3.3.2 Learning Algorithm

The primary motive to train a model, as discussed earlier, is to reduce the discrepancy between a given target distribution,$\pi(x)$ and the model distribution $p_\theta(x)$. In our case, the training of the quantum system as a model is done by time-evolving it using the Hamiltonian given in Eq. 3.1, and then optimising Θ_m, which denotes the set of $\{\Theta_i\}$ for L sites in the mth quench, over N different realisations in each quench.

Essentially, we are using the Monte Carlo method here. The core idea of the Monte Carlo method is repeated random sampling. That is, random samples are generated, the loss for each sample is computed, and the one with the lowest loss is selected. This step is repeated again and again. We have N samples, i.e., realisations, for each quench, and a total of M quenches. For an L qubit system, the number of trainable parameters is L.

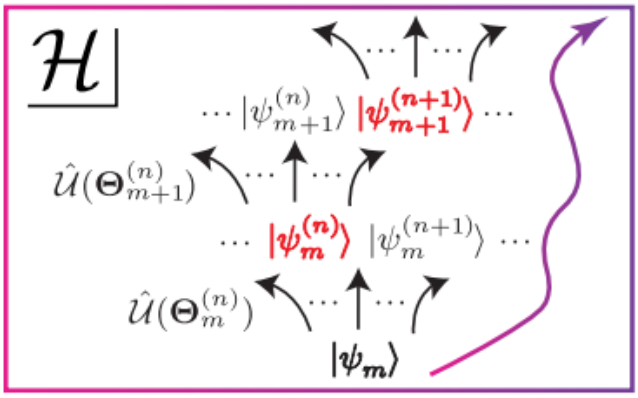


Figure 3.2: Schematic of learning algorithm, taken from Ref.[4].

Fig.3.2 depicts the training procedure. The algorithm proceeds iteratively through M quenches. During quench m, we prepare N independent copies of the quantum state $|\psi_m\rangle$ and apply distinct time-evolution operators $U(\Theta_m^{(n)})$ to each copy, where the $\Theta_m^{(n)}$ values are drawn from an identical probability distribution. After evaluating the loss across all N evolved states, we identify the state that achieves the minimum loss. This state $|\psi_m^{(n)}\rangle$ then serves as the initial condition for the subsequent quench at step $(m+1)$. Through successive iterations of this selection and evolution process over M quenches, the quantum state distribution progressively converges toward the target distribution. The overall training dynamics resemble a biased random walk through the system's Hilbert space, where the bias is introduced by the loss-based selection mechanism and thereby approaches the best parameter set.

Illustration.1 summarises the pseudocode for the learning algorithm.

Algorithm 1 Pseudocode for the Learning algorithm

Start with some initial state $|\psi(\mathbf{\Theta}_{m=0})\rangle \equiv |\psi_0\rangle$
while $m < M$ **do**
  while $n < N$ **do**
    Sample $\mathbf{\Theta}_m^{(n)}$ uniformly from $[0, 2\pi]$;
    Evolve the state $|\psi_{m+1}^{(n)}\rangle = U(\mathbf{\Theta}_m^{(n)})|\psi_m\rangle$ with
      $U = e^{-iH_{free}/2} e^{-H_{kick}} e^{-iH_{free}/2}$;
    Compute $\mathcal{L}(\mathbf{\Theta}_m^{(n)})$
    $n \leftarrow n + 1$
  end while
  $\mathbf{\Theta}_m = \arg\min_{\mathbf{\Theta_m^{(n)}}} \mathcal{L}(\mathbf{\Theta}_m^{(n)})$;
  $|\psi_{m+1}\rangle = U(\mathbf{\Theta}_m)|\psi_m\rangle$
  $m \leftarrow m + 1$
end while
We have the training outcome $|\psi_M\rangle$.

3.3.3 Parametric Regime

In the training of the kicked Ising spin chain model, we have kept the parameters J and B fixed at 0.2. This choice was made with the intention of positioning the system in the intermediate regime. Thus, the system would neither be in the Poissonian localised regime nor the chaotic regime. By operating in this transitional phase, we aim to ensure that the model has sufficient expressivity to learn different distributions, and at the same time, it remains responsive to the optimisation process, which would be difficult to attain in the limiting regimes.

We have taken 4 channels, $c = 4$ each of bandwidths, $\sigma_j^2 = [0.1, 0.25, 4, 10]$(taken from Ref.[4]). This choice of the bandwidths is done to ensure that the kernels are capable of capturing the local and global features in the target distribution. MMD loss is the best choice when dealing with large and complex distributions.

We have used two distinct methodologies to evaluate the performance of the model. Firstly, the most common benchmarking method of using the BAS dataset has been used. We train the model on the Bars and Stripes dataset. The BAS dataset has binary black and white images of size corresponding to $n \times n$ pixels. A simple case of 2×2 Bars and Stripes (BAS) has been learned using this model and turned out to be successful, and we have also looked at the 3×3 case as well. Secondly, the model was given a predefined target distribution, and its ability to match this distribution was evaluated. Though it functions as a supervised generative model, this test enables us to assess the model's accuracy in approximating a known analytical probability distribution.

In both cases, $M = 100$ quenches and $N = 500$ different realisations are used, otherwise specified.

3.3.4 Runtime Scaling

The complexity of the runtime of the learning algorithm is mainly decided by the number of quenches, M and the number of realisations/copies in each quench, which is N. Thus, the overall complexity becomes $\mathcal{O}(TMN)$. The major computational cost within each circuit evaluation is contributed by the time required to simulate the Floquet unitary of the model and the state vector evolution. T here represents the system evolution time. Since the MMD loss is evaluated analytically from the full state vector, it inherits the $\mathcal{O}(2^n)$ scaling as well.

Chapter 4

Results and Discussion

4.1 Learning the BAS data distribution

4.1.1 Learning 2×2 BAS Dataset

The Bars and Stripes dataset, composed of 2×2 binary images, is used to train the QCBM. This dataset contains 6 valid configurations. The valid configurations are shown in the Fig. 4.1.

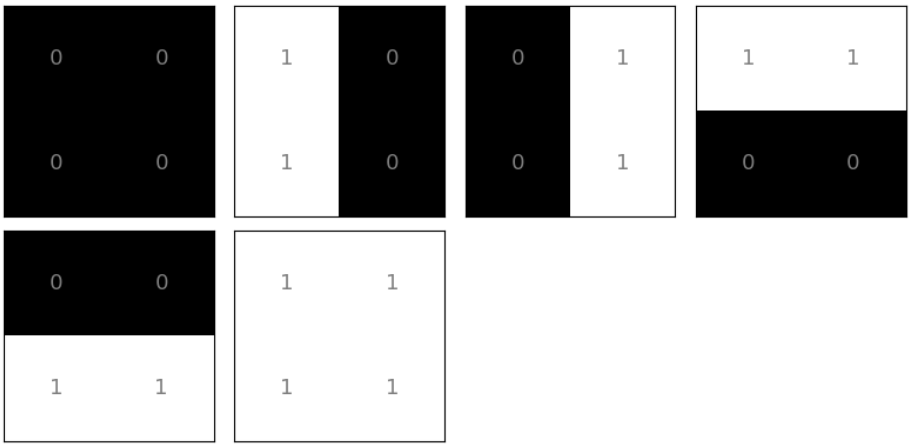


Figure 4.1: 2×2 BAS Dataset

This is encoded in a 4-qubit system, where each image is represented as a flattened bitstring. Thus, each point in the dataset has 4 features. As we have employed a 4-qubit model, there are a total of $2^4 = 16$ possible basis states. These basis states are used for encoding the binary string out of which only 6 are valid. The set of bitstrings for the valid configurations is $[0000, 1010, 0101, 1100, 0011, 1111]$ with $[0, 10, 5, 12, 3, 15]$ as the corresponding integers. The basis encoded quantum superposition state thus looks like,

$$|\psi_{target}\rangle = \frac{1}{\sqrt{6}}(|0000\rangle + |1010\rangle + |0101\rangle + |1100\rangle + |0011\rangle +)|1111\rangle) \tag{4.1}$$

The target probability distribution $\pi(x)$ using the dataset is computed and is shown in Fig. 4.2. Here x denotes the basis states.

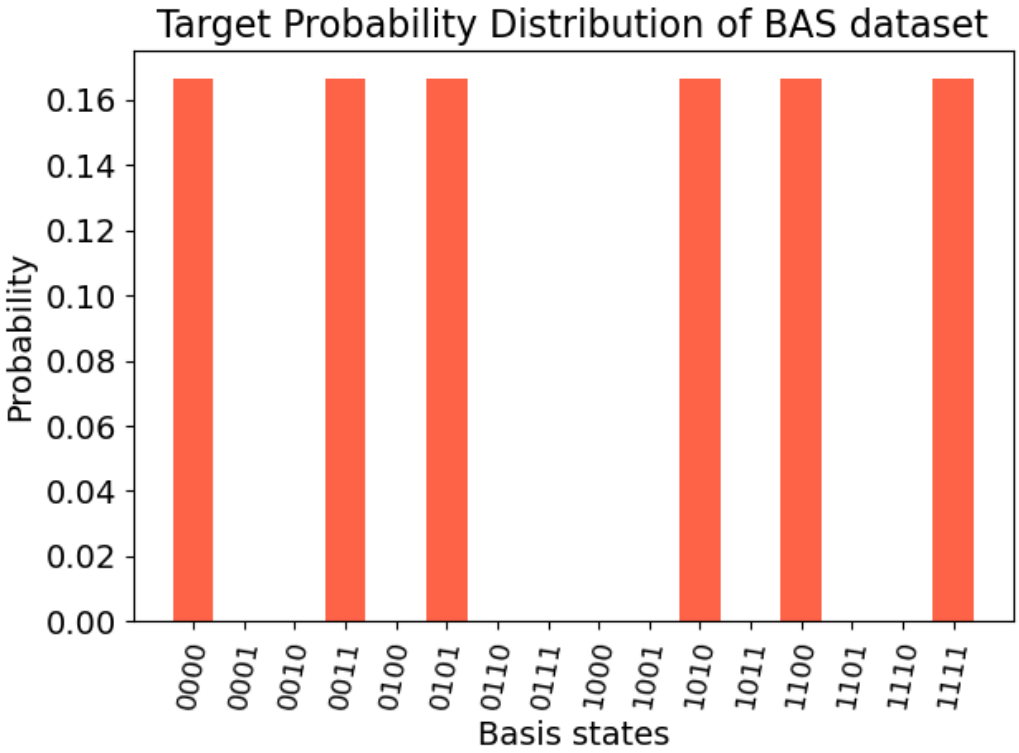


Figure 4.2: 2×2 BAS Dataset Target Probability Distribution

The probabilities are non-zero only for the valid configurations and zero for the rest. And it is a uniform distribution over the valid states.

We train our model to learn this probability distribution. The learning algorithm has already been discussed in the section 3.3.2, and by employing the MMD loss function discussed in section 3.2, which computes the distance between the model distribution $p_\theta(x)$ and target distribution $\pi(x)$, the generative model is trained to minimise this loss.

The training results are shown in Fig 4.3.

The training results for the 4-qubit model on the Bars and Stripes(BAS) dataset show a successful convergence of the model distribution toward the target distribution. The MMD loss trajectory shows a consistent decline over 100 quenches, with each quench consisting of 500 realisations corresponding to different parameter sets sampled, starting from a high initial value of approximately 0.05 and approaching near zero, which indicates that our model effectively minimises the discrepancy with the target distribution.

The MMD loss values obtained across 100 independent quenches are presented below:

```
Quench: 0    MMD Loss: 0.0529
```

```
Quench: 10   MMD Loss: 0.0164
Quench: 20   MMD Loss: 0.0137
Quench: 30   MMD Loss: 0.0130
Quench: 40   MMD Loss: 0.0082
Quench: 50   MMD Loss: 0.0058
Quench: 60   MMD Loss: 0.0050
Quench: 70   MMD Loss: 0.0037
Quench: 80   MMD Loss: 0.0026
Quench: 90   MMD Loss: 0.0019
Quench: 100  MMD Loss: 0.0021
```

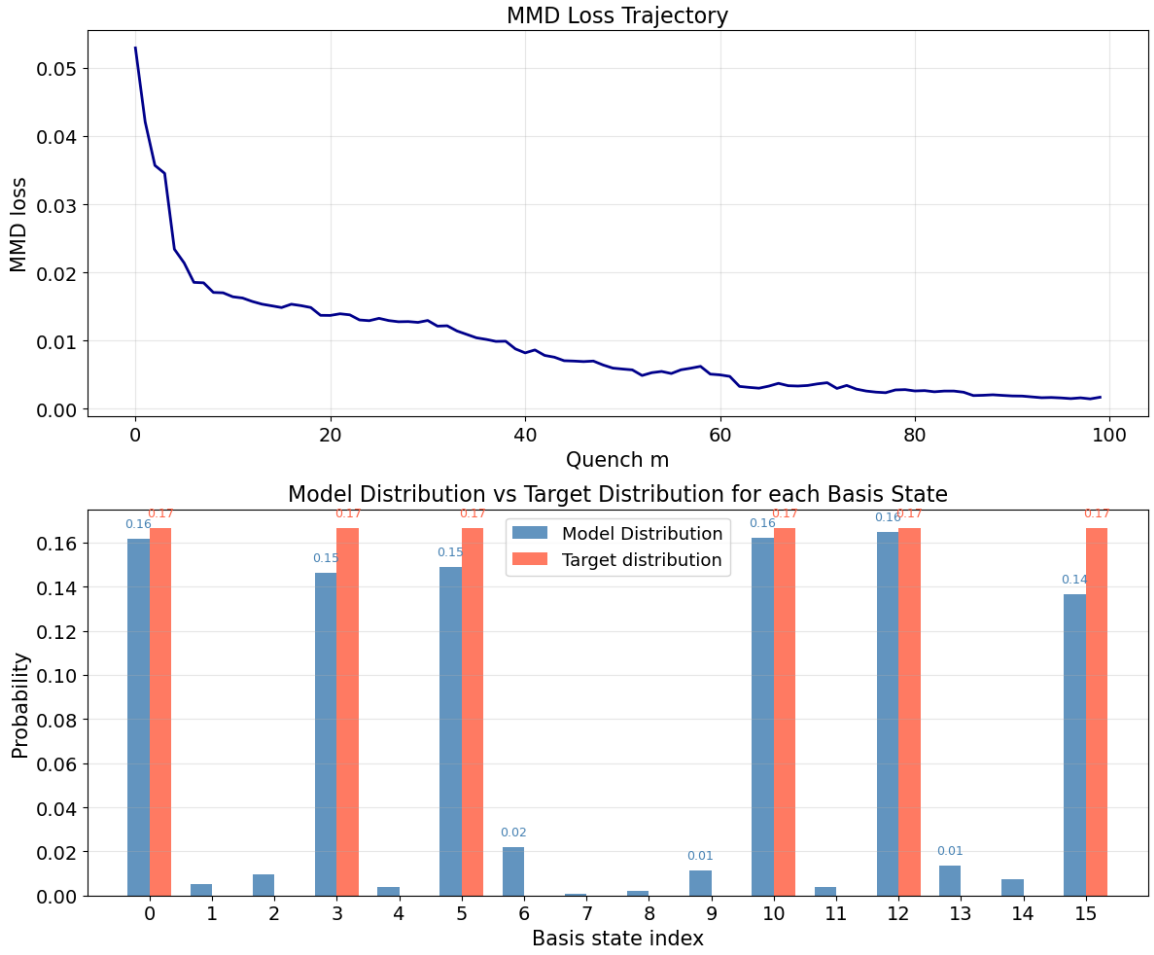


Figure 4.3: Plots showing the training results for the BAS 2×2 Dataset. The top plot depicts the trajectory of the Maximum Mean Discrepancy(MMD) loss over 100 quenches.In each quench, 500 realizations corresponding to different parameter sets are sampled to estimate the loss. The bottom plot compares the probability distribution of the final state $|\psi_{model}\rangle$ obtained after training with the target BAS distribution.

It can be observed that the training loss trajectory shows a clear downward trajectory, indicating that the model distribution is converging to the target distribution. Yet, a closer examination of the final quench loss points to some issues encountered

during the training of the model. A major issue was the quench-to-quench variation of the loss, where there was no monotonic decrease. As it can be observed between quench 90 (0.0019) and quench 100 (0.0021), there is a slight upward fluctuation in the loss. The loss is not guaranteed to decrease over the quenches, though the difference is not profound. This issue was seen in training throughout. This might be due to the stochasticity of the learning algorithm. Unlike gradient-based methods, where the trajectory follows a deterministic path down the steepest slope, the Monte Carlo method, which we have used, relies on random parameter sets. Even when the model nears the global minimum, the algorithm continues to search the neighbouring parameter space. Therefore, the slight increase in loss observed in the final quench (from 0.0019 to 0.0021) is a signature of this stochastic exploration, where in any step it is possible for it to land in a slightly less optimal region occasionally before the algorithm finds the point of least loss.

The model is further validated by comparing the final probability distribution of the model to the target probability distribution, where the model accurately assigns high probabilities ranging from $0.14 - 0.16$ for the correct basis states identified in the target distribution $\pi(x)$, which has a probability of 0.17. Though minor residual probabilities in non-target states can be seen, the alignment between the blue model bars and orange target bars confirms good approximation between the distributions, thereby showing the model's ability to learn the underlying structure of the 2×2 pixel dataset.

Further, the generative performance of the model was also evaluated. This was done by generating samples. A sampling-based hit rate offers a measure for the model's quality in generating valid dataset patterns. We generated 1000 samples using the probability distribution obtained from the final quantum state $|\psi_{model}\rangle$ of the system. A hit is recorded when the sample is a bar or a stripe. The sampling test yielded a hit rate of approximately 92%. This value remained stable as the number of samples was increased, indicating statistical convergence. This high accuracy indicates that the model has successfully concentrated the majority of its probability mass on the valid basis states as required in the BAS dataset.

Some of the samples are attached in Fig. 4.4. Those with a green border indicate valid samples, and those with a red border indicate invalid samples.

4.1.2 Learning 3×3 BAS Dataset

Now lets look into the Bars and Stripes dataset composed of 3×3 binary images. This dataset consists of 14 valid configurations, which are shown in Fig. 4.5. We use the probability distribution of this dataset for the model to learn and approximate.

This dataset distribution is encoded in a 9-qubit system in the same way as the

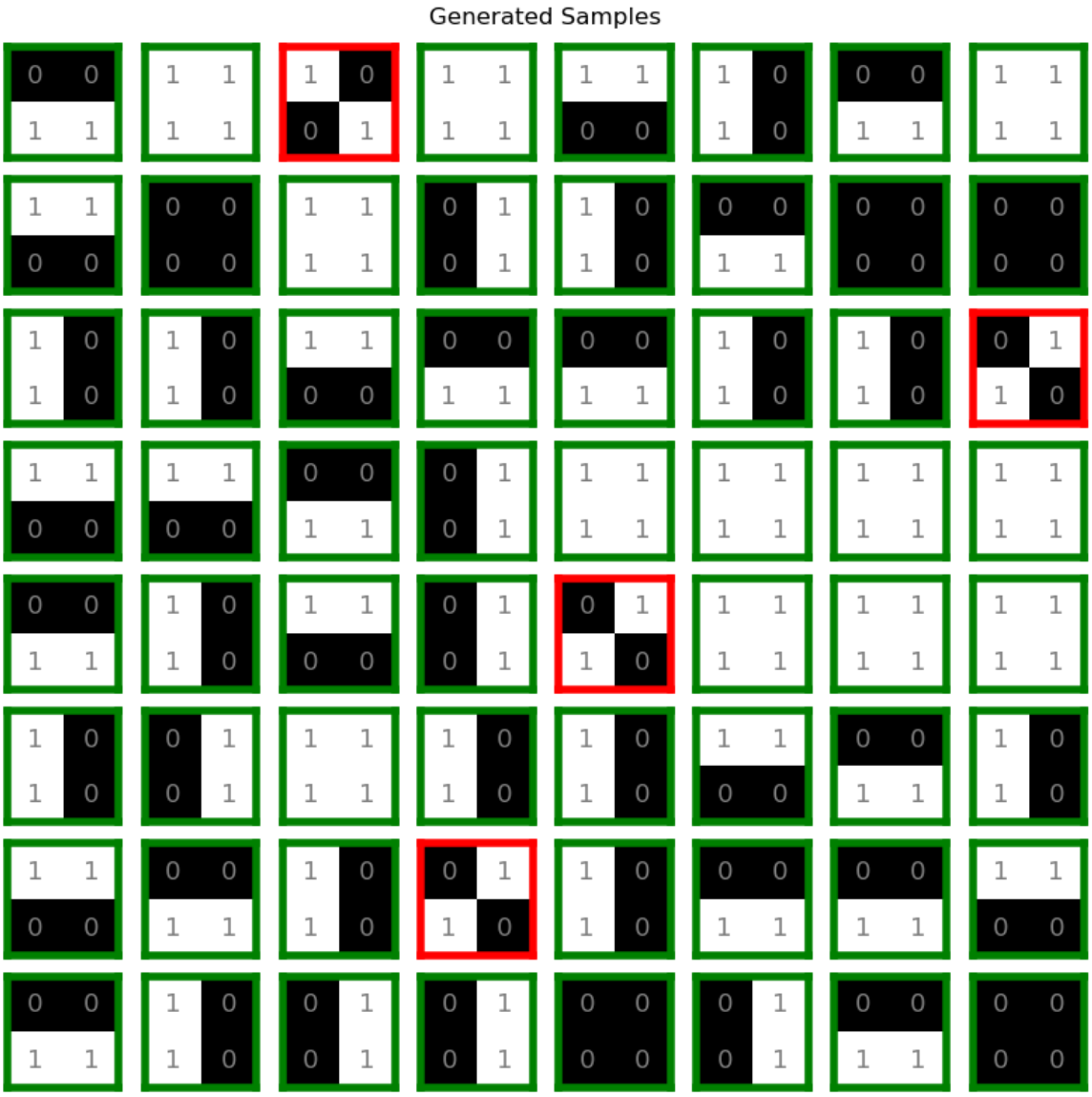


Figure 4.4: Samples generated by the trained model using the probability distribution of the 2×2 BAS dataset. Those with a green border are valid samples, and those with a red border are invalid samples.

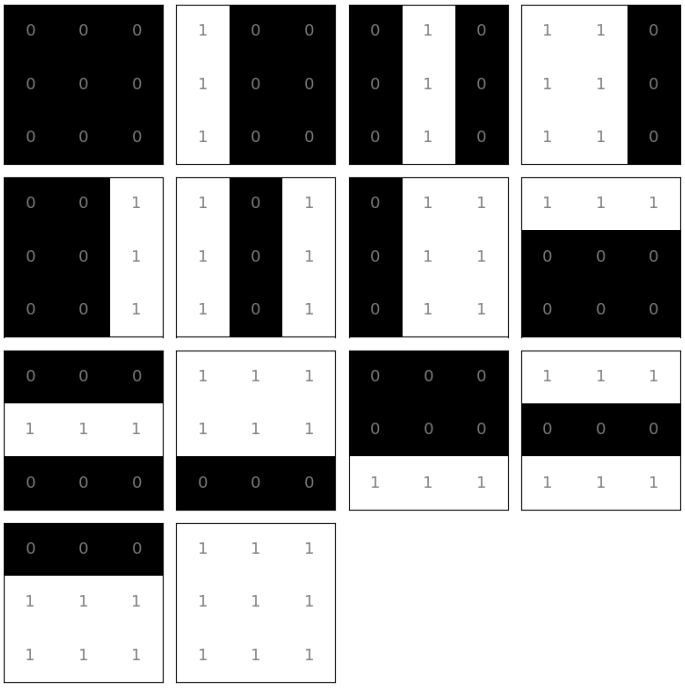


Figure 4.5: 3×3 BAS Dataset

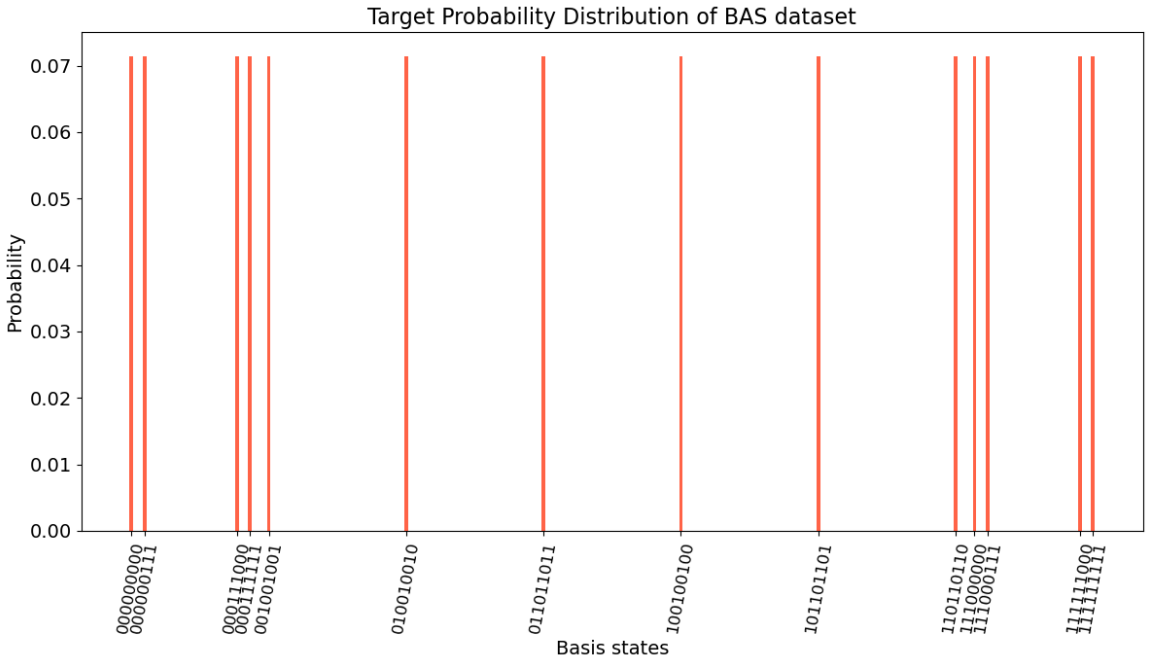


Figure 4.6: 3×3 BAS Dataset Target Probability Distribution

2×2 BAS dataset was done. The model is $2^9 = 512$ dimensional with 9 trainable parameters.

The target distribution $p(x)$ in this case is shown in Fig. 4.6. The probabilities are non-zero only for the valid configurations and zero for the rest. And it is uniformly distributed over the valid basis states.

The model has to be trained to learn and represent this distribution. Here, the number of realisations is $N = 500$ and the number of quenches, $M = 100$. The MMD loss values obtained across 100 independent quenches have been analysed. The following are the MMD losses observed corresponding to the quenches.

```
Quench: 0    MMD Loss: 0.0638
Quench: 10   MMD Loss: 0.0417
Quench: 20   MMD Loss: 0.0372
Quench: 30   MMD Loss: 0.0360
Quench: 40   MMD Loss: 0.0328
Quench: 50   MMD Loss: 0.0328
Quench: 60   MMD Loss: 0.0329
Quench: 70   MMD Loss: 0.0328
Quench: 80   MMD Loss: 0.0326
Quench: 90   MMD Loss: 0.0333
Quench: 100  MMD Loss: 0.0333
```

The loss is seen to drop from 0.0638 to 0.0328 within the first 40 quenches. But then the rate of decline of the loss trajectory slows down significantly. The loss does not go any lower, but rather increases to 0.0333. The loss just starts to fluctuate non-monotonically. Thus, the loss shows a prolonged period of stagnation and stochastic oscillation. This behaviour hints at the inability of the system to learn such a distribution, thus the ansatz lacks the necessary expressivity to achieve convergence ($MMD \to 0$) required for a perfectly sparse 3×3 BAS distribution, which could be due to the limited parametrisation.

The training results are shown in Fig.4.7.

The second plot compares the target distribution against the learned model distribution. While the target is a perfectly sparse uniform distribution over 14 states (each with $p \approx 0.071$), the model distribution is significantly noisier and spread out across the whole 512-dimensional Hilbert space basis states. There is a high probability even at basis states where the target probability is zero.

The same sampling test performed for the 2×2 BAS-trained 4-qubit model was also done for this model. The sampling test yielded a hit rate of approximately 45% only, which indicates the sampling efficiency of the model. This was a major drawback for this case. The presence of significant probability 'leakage' into invalid basis states explains the recorded 45% sampling efficiency.

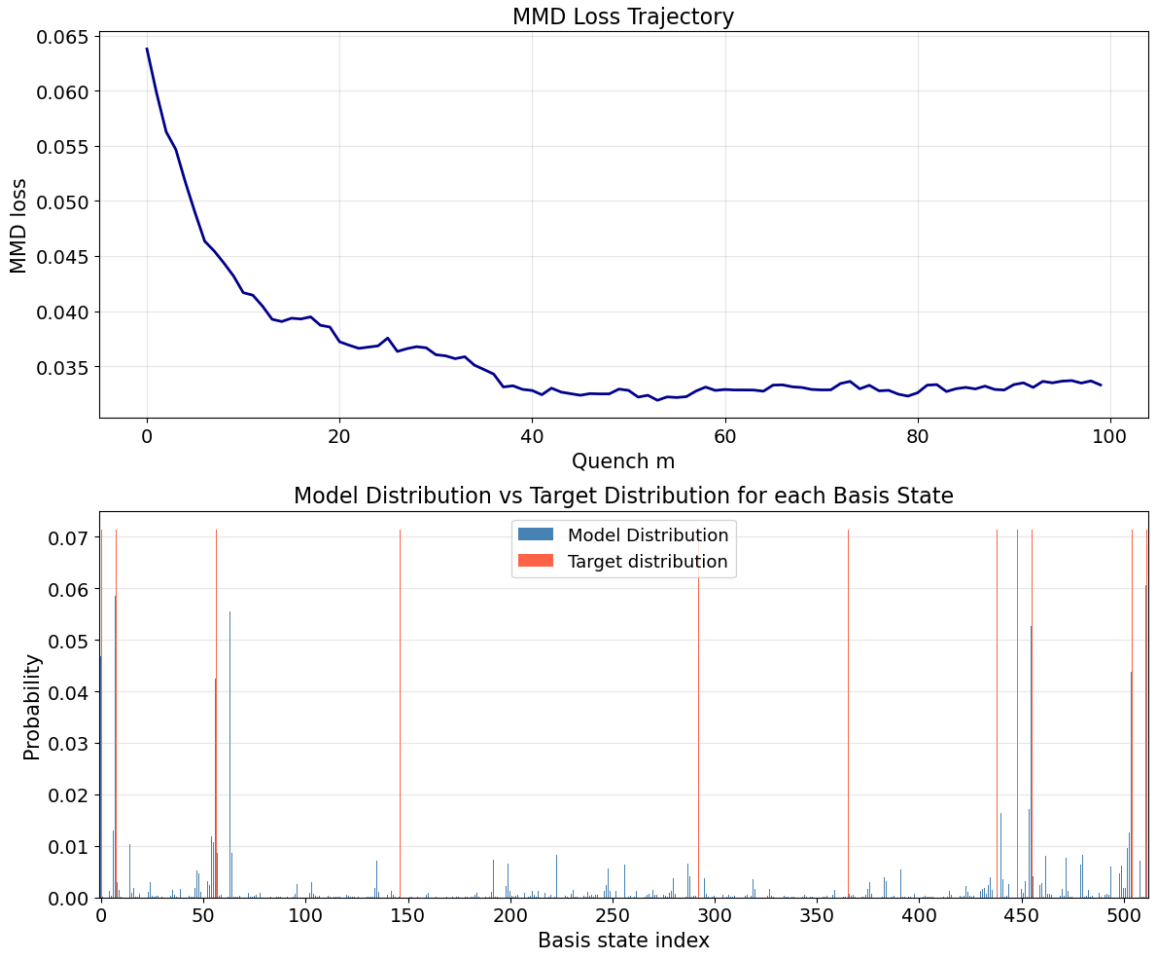


Figure 4.7: Plots showing the training results for the BAS 3×3 Dataset. The top plot depicts the trajectory of the Maximum Mean Discrepancy(MMD) loss over 100 quenches.In each quench, 500 realizations corresponding to different parameter sets are sampled to estimate the loss. The bottom plot compares the probability distribution of the final state $|\psi_{model}\rangle$ obtained after training with the target BAS distribution.

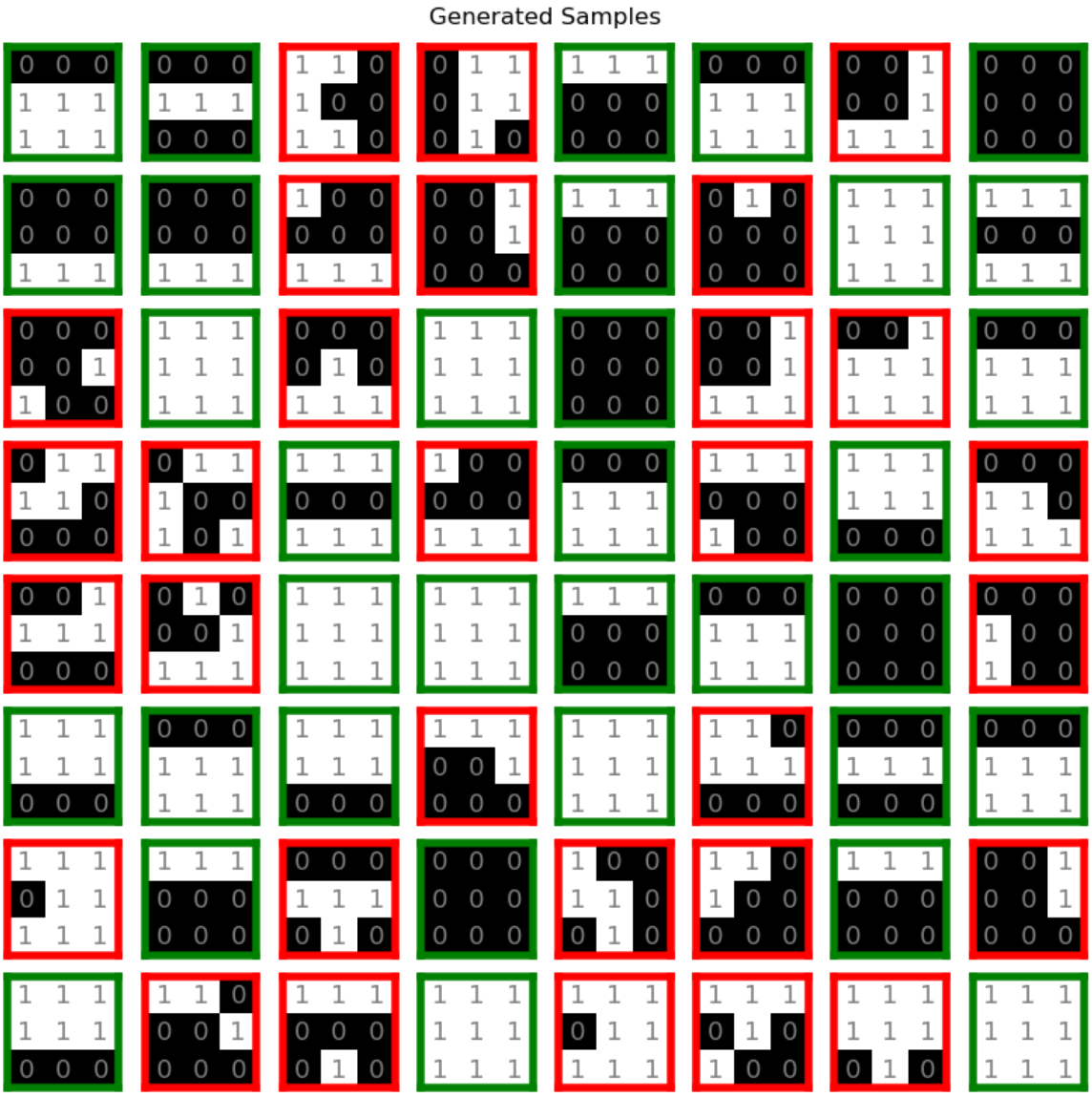


Figure 4.8: Samples generated by the trained model using the probability distribution of the 3×3 BAS dataset. Those with a green border are valid samples, and those with a red border are invalid samples.

Some of the samples are attached in Fig. 4.8.

4.1.3 Discussion

The main objective was to implement a quantum many-body system as a toy generative model and examine its potential in generative modelling. In the case of the 2×2 BAS dataset implemented on a 4-qubit sytsem, the target distribution consisted of uniform probability mass distributed across 6 valid basis states within a 16-dimensional Hilbert space. From the results, we could observe that the model with only a total of 4 trainable parameters possesses sufficient expressivity to approximate this distribution. Thus showing a high fidelity in both the analytical loss minimisation as well as in the valid data sampling tasks. The model distribution exhibited strong overlap with the target distribution, achieving a sampling efficiency of 92% for valid configurations.

In the case of the 3×3 BAS dataset, which was implemented on a 9-qubit sytsem, the target distribution consisted of uniform probability mass distributed across 14 valid basis states within a 512-dimensional Hilbert space. That means there are only 14 valid configurations among the 512 possible configurations. Mathematically, this means that 97.3% of the basis states are invalid. This is a highly sparse distribution. Since the valid configurations occupy only $\approx 2.7\%$ of the basis state, a stochastic kind of search through this Hilbert space makes it difficult to exactly locate and amplify the probability of the non-zero basis states. The plots of the target and model distributions we obtained, as mentioned already, do not agree well. Also, the sampling efficiency was found to be just 45%, which is low.

The trainable parameter space dropped significantly while scaling the system size. In the 4-qubit system BAS dataset, there were 4 trainable parameters, which were sufficient enough to navigate a 16-dimensional space. Whereas in the 9-qubit system BAS dataset, 9 parameters provide insufficient degrees of freedom to independently vary the amplitudes of 512 basis states, especially when the target requires zeroing out the rest of the 498 basis states. This indicates that the 9-qubit system with 9 trainable parameters is not expressive enough to learn such a sparse distribution.

Even though we are using a physically motivated Floquet-based ansatz, the system could have some inherent symmetries that do not align well with this kind of highly sparse distribution of the 3×3 BAS dataset over 9 qubits. This could limit the model's ability to learn the 14 valid configurations and thus cause probability leakage to the invalid states. A Monte-Carlo search would be wandering around in the 97.3% space of invalid states.

The decline in the sampling efficiency from 92% in the 4-qubit case to 45% in the 9-qubit case indicates that for larger grids, it becomes necessary to increase the number

of trainable parameters so that it can capture such kinds of complex distributions.

4.2 Learning Distributions

The model is now used to approximate various smooth, continuous functions. Here we have done this on a 10-qubit system. The number of realisations for each quench is $N = 500$ with $M = 100$ quenches in total.

4.2.1 Target Distributions

The system is trained to learn the following known probability distributions as target distributions. The distribution is mapped across the $2^{10} = 1024$ basis states of the 10-qubit system used as the learning model.

Gaussian Distribution

We have implemented a Gaussian distribution as the target distribution. The target probability $P(i)$ for each basis state $i \in \{0, 1, \ldots, 1023\}$ is calculated using the discrete Gaussian function

$$P(i) \propto \exp\left(-\frac{(i-\mu)^2}{2\sigma^2}\right) \tag{4.2}$$

where we have selected the mean $\mu = 512$ and the standard deviation $\sigma = 50$. The distribution was normalised such that $\sum_i P(i) = 1$ so that it represents a valid probability distribution.

Exponential Decay Distribution

The exponential decay profile is used for training. The probability $P(i)$ for each basis state is defined by,

$$P(i) \propto \exp(-\lambda i) \tag{4.3}$$

where the decay constant, $\lambda = 0.02$.

Bimodal Distribution

The Bimodal Gaussian distribution is used to test the model's ability to represent multi-modal probability distributions. Unlike a simple Gaussian distribution, here the model requires distributing its probability mass into two distinct, non-overlapping regions of the Hilbert space. The means used for this purpose were $\mu_1 = 256$ and $\mu_2 = 768$, with standard deviations $\sigma_1 = \sigma_2 = 50$ thus we have a symmetric bimodal.

The target distribution over the basis state is defined as

$$P(i) \propto \exp\left(-\frac{(i-\mu_1)^2}{2\sigma_1^2}\right) + \exp\left(-\frac{(i-\mu_2)^2}{2\sigma_2^2}\right). \tag{4.4}$$

which is also normalised.

Uniform Distribution

The distribution used is the uniform distribution over a finite interval. This distribution is used to check the ability of the model to learn sharp discontinuities. Here, the target is defined as a constant probability assigned to the indices in the range $[150, 350]$, with zero probability for the rest of the states in the 1024-dimensional Hilbert space.

The results of training the model for each of the above target distributions are shown in Figure 4.9.

Further, we have monitored the final and the global minimum MMD loss while training the model for each target distribution. This data is given in the Table 4.1.

Table 4.1: Comparison of the final and global minimum loss.

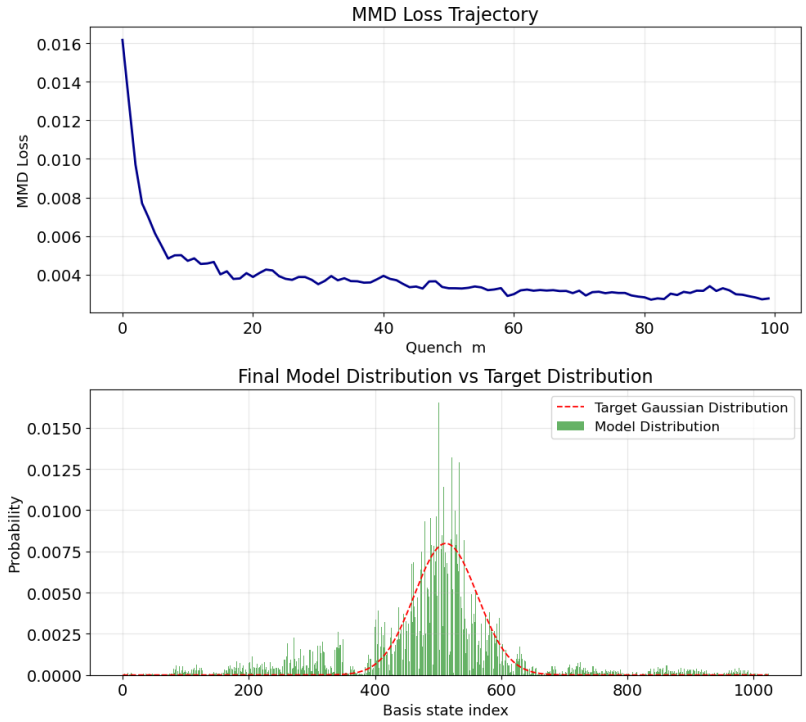
Target Distribution	**Qubits**	**Global Minimum Loss**	**Final MMD Loss**
Gaussian Distribution	10	0.002695 at Quench 82	0.002756
Bimodal Distribution	10	0.001342 at Quench 43	0.001465
Exponential Distribution	10	0.001140 at Quench 22	0.001631
Uniform Distribution	10	0.002892 at Quench 100	0.002892

4.2.2 Discussion

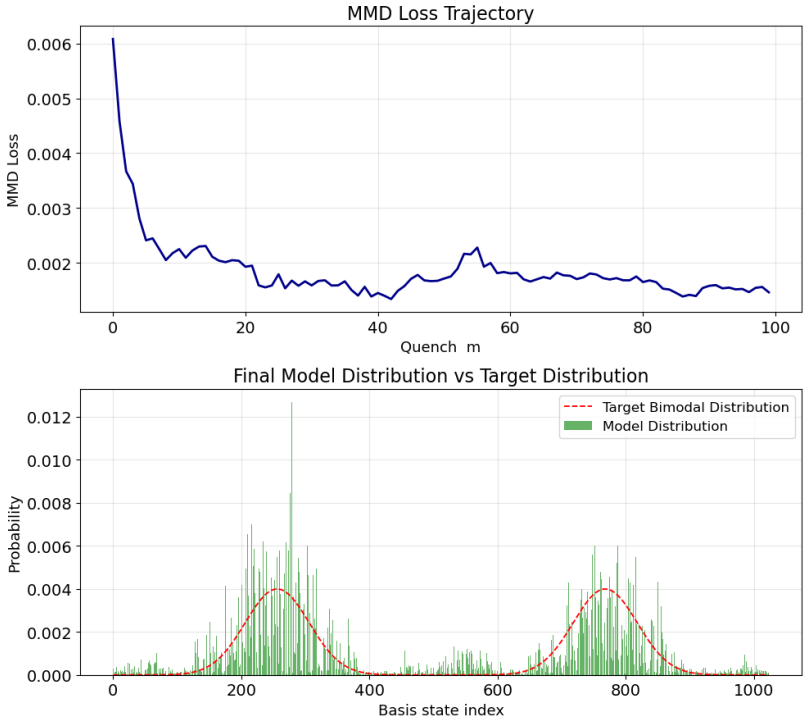
The primary goal of using known continuous distributions as targets was to understand the ability of the model to learn smooth and non-smooth distributions.

Gaussian Performance

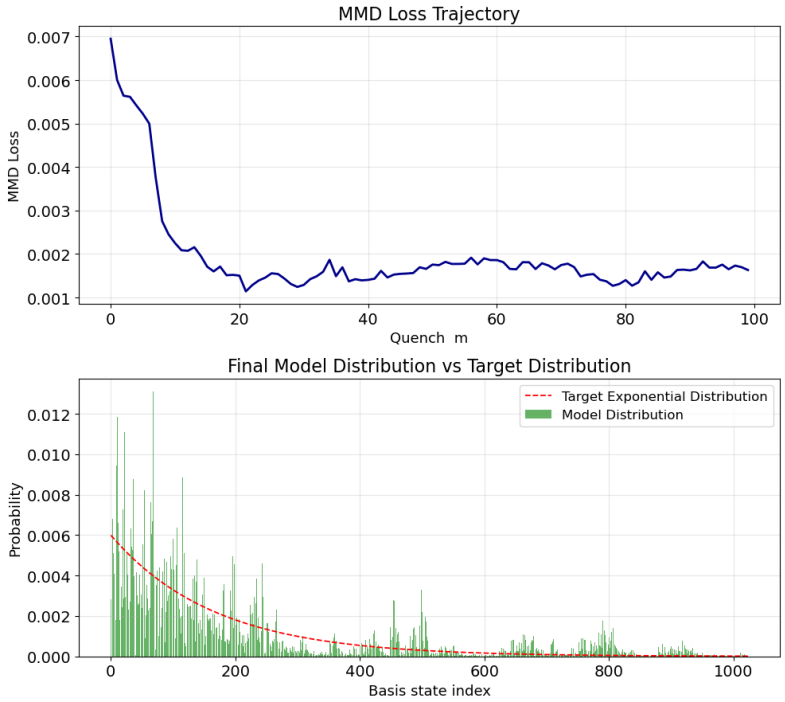
From the results of the model to learn Gaussian distributions, the final loss was found to be 0.002756, which means the model has learned the target successfully. The global minimum loss was found to be 0.002695 at Quench 82, which is quite lower than the final loss. This could be because of the stochasticity of the optimisation method chosen, which highlights the drawbacks of the optimisation method chosen. Since the loss difference is quite low, it indicates that the optimisation has succeeded in attaining a decent convergence. From the plot, it can be observed that the model



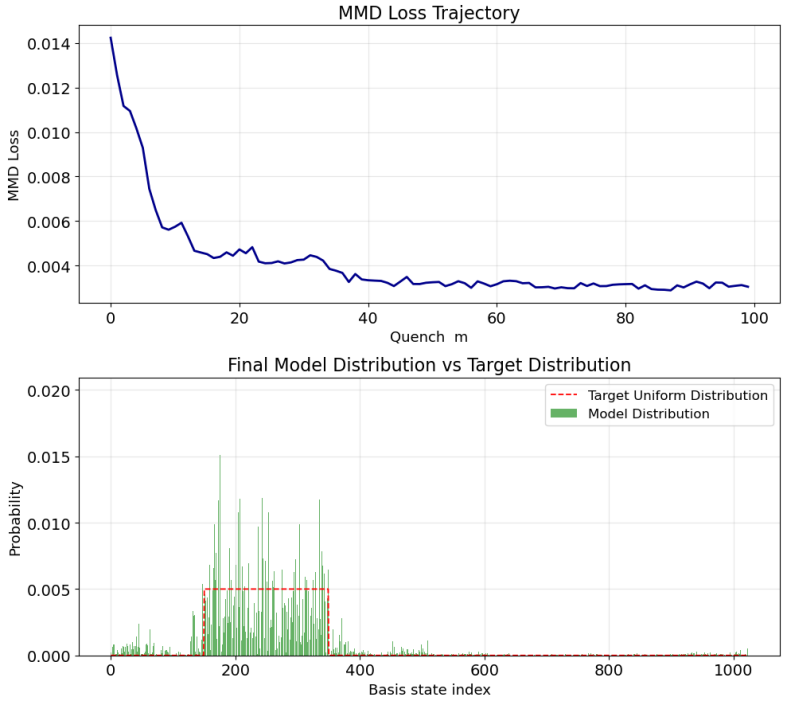
(a) Gaussian Distribution



(b) Bimodal Distribution



(c) Exponential Distribution



(d) Uniform Distribution

Figure 4.9: Benchmarks of Model Distribution against various Target Distributions. (a) Gaussian, (b) Bimodal, (c) Exponential, and (d) Uniform Window. Each plot consists of the MMD loss trajectory (top) and the final probability distribution comparison (bottom).

distribution has good overlap with that of the target distribution. This indicates that the model is capable of learning simple localised, symmetric structures.

However, in other training runs, it was also observed that the final loss had a good coincidence with the global minimum loss recorded.

Bimodal Distribution

In this case, the model achieves a Global Minimum Loss of 0.001342 and the final loss of 0.001465, successfully resolving both peaks at $\mu_1 = 256$ and $\mu_2 = 768$. The final loss is greater than the global minimum loss obtained. This kind of distribution essentially tests the entangling power of the model as it requires concentrating the probability at two distant peaks, simultaneously suppressing the probability at the central region. The plot of the comparison between the model and the target distribution shows good agreement.

Instead of collapsing to a single mean like simpler models, the Floquet-based ansatz spreads the probability across two separate regions in the Hilbert space. This suggests that the many-body interactions are strong enough to create the non-local correlations required for multi-modal states.

Exponential Distribution

From the results, the final loss obtained was 0.001631, which is much greater than the global minimum loss 0.001140 while training for an exponential target distribution. This depicts the ability of the model to learn an asymmetric distribution. Although it has the greatest loss value compared to other 4 distributions. The model was capable of creating a peak around the 0 basis states with the probability profile declining further. However, it can be observed that the global minimum loss and the final loss did not coincide. Further, there is probability leakage at other basis states as well.

The observed probability leakage into those basis states supposed to have negligible probabilities indicates a limitation in the expressivity of the current ansatz or a lack of sufficient depth to perfectly capture the sharp decay of the exponential tail.

Uniform Distribution

The final loss in this case was found to be 0.002892, and the final loss and the global minimum loss are seen to have coincided. The model was able to correctly represent the probability concentration between the indices 150 and 350, though there is a probability leakage to the rest of the basis indices. The distribution in the window is also not perfectly uniform.

As seen in all four cases of training the model to approximate a target distribution, there were instances where the final MMD loss and the global minimum MMD loss over

the 100 quenches were not the same. This discrepancy comes from the Monte Carlo method chosen. This represents the trade-off between the exploration capabilities of the search method adopted and the precision of final convergence rather than the ansatz's expressivity itself. The ansatz is seen to be efficient in learning smooth target distributions, and the uniform over a finite interval type of distributions.

Chapter 5

Conclusion and Future Work

In this thesis, the primary goal was to investigate the potential of quantum many-body systems as a generative model. We employed a Floquet system, kicked Ising spin chain as the ansatz within the Quantum Circuit Born Machine(QCBM) framework. The major motivation was to investigate whether the ability of a quantum system to naturally generate complex probability distributions could be utilised in learning classical data and distributions. Some works have essentially proved to have this advantage. Unlike conventional variational quantum circuits, in which the circuits are random and unstructured, we intend to replace this with a physically inspired quantum system that inherently has structure. Our model kicked Ising spin chain, and its potential as a generative model is what is studied in this work.

In this work, the model is trained to approximate a known probability distribution, thereby performing a supervised generative task. The model is trained using a directed random walk, a Monte Carlo-based approach, to minimise the Maximum Mean Discrepancy (MMD) loss between the generated and target probability distributions. For evaluating its performance, we have performed two primary benchmarking tasks. The first is to learn the Bars and Stripes(BAS) data set of both 2×2 and 3×3 grid sizes, where the model is supposed to capture the full probability distribution of all valid patterns. The second is to approximate analytical probability distributions, such as Gaussian, exponential, bimodal and uniform distributions.

The 2×2 BAS dataset embedded in a 4-qubit system by the method of basis encoding has shown the ability to approximate the intended target distribution. In this case, there are 4 trainable parameters and the Hilbert space dimension is 16, out of which 6 represent valid configurations. In this case, the final MMD loss observed was 0.0021. Doing tests for sampling efficiency, it showed an accuracy of 92%. Thus, it can be inferred that the model could approximate the distribution successfully, indicating its generative ability and expressivity in a low-dimensional space.

In contrast, the 3×3 BAS dataset posed significant challenges. It was embedded in a 9-qubit system similar to the previous case. The Hilbert space dimension is 512,

out of which only 14 are valid. This gives a highly spare distribution. The MMD loss through the quenches was observed to be stagnant at the value around 0.03 and did not decrease further. A sampling efficiency of only around $40 - 45\%$ was seen. This points to the inability of the system to approximate such a complex, highly sparse distribution with only 9 trainable parameters.

This performance degradation as the system scales to higher dimensions with increasingly sparse target distributions indicates that the number of trainable parameters becomes insufficient to independently modulate the amplitudes of all basis states. Also, the stochastic nature of Monte Carlo optimisation could not guarantee a monotonic decrease of MMD loss, as an absolute convergence was not observed in most of the cases while training.

The analysis of the training results of the test to match the known target analytic distributions showed that the model could successfully approximate such smooth and continuous distributions over a 10-qubit system. For each case, the model has achieved low MMD losses, exhibiting a high ability to represent localised distributions, such as Gaussian and bimodal. The multi-modal structures require the capacity to distribute probability mass across spatially separated regions. As the model is capable of doing this, it suggests that the underlying quantum ansatz is expressive enough to capture such correlations. In addition, the uniform distribution over a finite interval with sharp discontinuities and the exponential distribution, which requires distributing the probability mass over the range, have also been attained decently. The major problem observed in all these cases was probability leakage.

A recurring challenge across all tasks was the non-monotonic convergence behaviour arising from the Monte Carlo-based optimisation. The final MMD loss often did not coincide with the lowest value observed during training, which could be due to the stochastic nature of the search process. While this approach is relatively robust to noise and avoids issues associated with gradient-based methods, such as barren plateaus, it comes at the cost of precise convergence.

Future directions

The Monte-Carlo based approach used in our work is gradient-free and thus does not face issues such as barren plateaus that are commonly encountered in variational quantum algorithms. However, this comes at the cost of convergence. The optimisation process can be slow and often exhibits stochastic wandering and non-monotonic convergence, making it difficult to reach optimal solutions. One possible improvement would be to use gradient-based optimisation methods, such as parameter-shift techniques or finite-difference gradients, which could provide more directed navigation of the loss landscape.

Another direction for future work is to increase the expressibility of the model by increasing the parameter space. This can be achieved by applying the Floquet unitary over multiple time steps, effectively creating a layered structure similar to deeper circuit architectures. Finally, while the current study focuses on relatively simple, toy problems, an important next step would be to explore how this model performs on more complex and practically relevant industrial tasks, and to assess its scalability in such settings.

Bibliography

[1] M. V. Berry, in *Topics in Nonlinear Dynamics*, AIP Conference Proceedings, Vol. 46, edited by S. Jorna (American Institute of Physics, New York, 1978) pp. 16–120.

[2] M. Revythi and G. Koukiou, Machine Learning and Knowledge Extraction **7**, 75 (2025).

[3] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Nature Reviews Physics **3**, 625 (2021).

[4] W. Zhong, X. Gao, S. F. Yelin, and K. Najafi, Phys. Rev. Res. **6**, 043041 (2024).

[5] T. Herrmann, M. F. I. Kieler, and A. Bäcker, Physical Review E **108**, 044213 (2023).

[6] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, San Diego, 2014).

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, Cambridge, 2010).

[8] X.-Z. Luo, J.-G. Liu, P. Zhang, and L. Wang, Quantum **4**, 341 (2020).

[9] E. Farhi and H. Neven, arXiv preprint arXiv:1802.06002 10.48550/arXiv.1802.06002 (2018), arXiv:1802.06002 [quant-ph] .

[10] M. Robnik, in *Photophysics and Photochemistry in the Vacuum Ultraviolet*, NATO ASI Series, Vol. 142, edited by S. P. McGlynn, G. L. Findley, and R. H. Huebner (Springer, Dordrecht, 1985).

[11] T. Shinbrot, C. Grebogi, J. Wisdom, and J. Yorke, American Journal of Physics - AMER J PHYS **60**, 491 (1992).

[12] M. Berry, European Journal of Physics **2**, 91 (2000).

[13] A. Politi, Scholarpedia **8**, 2722 (2013), revision #137286.

[14] H.-J. Stöckmann, *Quantum Chaos: An Introduction* (Cambridge University Press, Cambridge, 1999).

[15] T. Guhr, A. Müller-Groeling, and H. A. Weidenmüller, Physics Reports **299**, 189 (1998).

[16] E. P. Wigner, SIAM Review **9**, 1 (1967), https://doi.org/10.1137/1009001 .

[17] V. Oganesyan and D. A. Huse, Phys. Rev. B **75**, 155111 (2007).

[18] Y. Y. Atas, E. Bogomolny, O. Giraud, and G. Roux, Phys. Rev. Lett. **110**, 084101 (2013).

[19] C. E. Porter and R. G. Thomas, Phys. Rev. **104**, 483 (1956).

[20] F. Haake, M. Kuś, and R. Scharf, Zeitschrift für Physik B Condensed Matter **65**, 381 (1987).

[21] M. C, U. Divakaran, and A. Lakshminarayan, Phys. Rev. E **112**, 044219 (2025).

[22] C. Manju, U. Divakaran, and A. Lakshminarayan, Physical Review E **112**, 044219 (2025).

[23] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. (Prentice Hall, Upper Saddle River, New Jersey, 2003).

[24] H.-Y. Huang, S. Choi, J. R. McClean, and J. Preskill, The vast world of quantum advantage (2025), arXiv:2508.05720 [quant-ph] .

[25] L. K. Grover, in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (Association for Computing Machinery, New York, NY, USA, 1996) pp. 212–219.

[26] P. W. Shor, SIAM Journal on Computing **26**, 1484 (1997).

[27] D. Camps, R. Van Beeumen, and C. Yang, Numerical Linear Algebra with Applications **28**, e2331 (2021).

[28] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Phys. Rev. X **4**, 031002 (2014).

[29] J.-G. Liu and L. Wang, Phys. Rev. A **98**, 062324 (2018).

[30] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, npj Quantum Information **5**, 45 (2019).

[31] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, 2nd ed., Quantum Science and Technology (Springer Cham, 2021).

[32] A. Lakshminarayan and V. Subrahmanyam, Phys. Rev. A **71**, 062334 (2005).

[33] Y. Li, K. Swersky, and R. Zemel, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (2015) pp. 1718–1727.

[34] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, arXiv preprint arXiv:1505.03906 (2015).

[35] O. Tunali, Maximum mean discrepancy (mmd) in machine learning, `https://onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html` (2019), updated: December 3, 2024.