BAYESIAN INFERENCE IN QUANTUM GENERATIVE MODELLING*

R. RAJPAL , J. ZORYK , AND U. KARADAG

Abstract. Quantum computing has the potential to revolutionize various fields, but current quantum hardware is prone to noise and errors that can limit the performance of quantum algorithms. This report explores using Bayesian inference to estimate the distribution of quantum circuit parameters in the Quantum Circuit Born Machine (QCBM) used for generative modeling. Bayesian inference allows uncertainty quantification and incorporation of prior knowledge. However, computing gradients in quantum circuits can be costly for large datasets due to the high-dimensional parameter space, so stochastic gradient approximations are often used. While this reduces computational cost, it introduces gradient noise that can bias the parameter samples. To address this, we investigate using the Stochastic Gradient Adaptive Langevin Thermostat (SGAdLT) to reduce the bias from stochastic gradient noise. Experiments are conducted on a Gaussian mixture model and the Bars-and-Stripes dataset using simulated quantum circuits. The effect of the temperature parameter β, gradient batch size, and artificial gradient noise are analyzed. Results show SGAdLT helps explore local cost minima at lower β values compared to standard SGLD. This work demonstrates the potential of combining Bayesian inference with quantum machine learning to enhance the robustness of variational quantum algorithms in the presence of hardware noise and computational limitations. Further research directions are also discussed.

1. Motivation. Quantum computing has the potential to revolutionize various fields, including cryptography, drug development, and optimization. However, the current state of quantum hardware is still prone to noise and errors, which can limit the performance and reliability of quantum algorithms. Variational quantum algorithms, such as the Quantum Circuit Born Machine (QCBM) [1] used for generative modeling, offer a promising approach to mitigate these issues by optimizing the parameters of the quantum circuit to minimise the impact of noise and improve robustness.

Bayesian inference provides a principled framework for estimating the distribution of quantum circuit parameters, allowing for the quantification of uncertainty and the incorporation of prior knowledge [2]. By sampling from the posterior distribution of circuit parameters using sampling methods, such as Langevin Monte Carlo or Hamiltonian Monte Carlo (HMC), we can explore the complex probability landscape and identify optimal parameter configurations that enhance the performance and reliability of the quantum circuit.

*RSCAM Group Project,Supervisor: Prof. Benedict Leimkuhler

However, computing gradients in variational quantum algorithms can be costly, especially for large datasets. To address this challenge, stochastic gradient approximations are often employed [2], where a subset of the data is used to estimate the gradient. While this approach reduces computational overhead, it introduces gradient noise, which can ultimately bias the samples and lead to suboptimal parameter estimates.

To overcome this limitation, various debiasing strategies have been proposed in the literature such as: stepsize decay and Multilevel Monte-Carlo [3]. These techniques aim to correct the bias introduced by stochastic gradient approximations and improve the quality of the samples. In this project, we use the Stochastic Gradient Adaptive Langevin Thermostat (SGAdLT) [4], also known as the stochastic gradient Nosé-Hoover thermostat, as our debiasing strategy.

SGAdLT is an adaptive sampling method that adjusts the noise level of the sampler based on the observed gradient noise. By incorporating a thermostat variable that controls the temperature of the system, SGAdLT can effectively counteract the bias introduced by stochastic gradients [4]. The method dynamically adapts the noise level to ensure that the samples are drawn from the correct target distribution, even in the presence of gradient noise.

The motivation for this project is to explore the effectiveness of SGAdLT in the context of generative quantum models, specifically focusing on its application in Quantum Circuit Born Machine (QCBM). By investigating the performance of SGAdLT in reducing the bias caused by stochastic gradient approximations, we aim to improve the quality of samples, leading to more accurate and robust parameter estimates.

This research has the potential to enhance the reliability and performance of quantum models in the presence of hardware noise and computational limitations. By developing more robust Bayesian inference methods for quantum circuits, we can contribute to the advancement of quantum computing and its practical applications in various domains.

2. Background.

2.1. Generative modelling. Generative AI has seen an explosion in applications over the past decades with examples such as: LLMs like ChatGPT, Diffusion Models in Images, Audio Generation models etc [5]. The premise of generative model-

ling is that we have samples X from an unknown distribution P i.e., $X \sim P$ with the aim to generate samples Y that look like X by sampling $Y \sim Q$ from some tuneable distribution Q.

The data distribution P represents the true, underlying distribution from which the observed data samples X are drawn. In real-world scenarios, the data distribution is often unknown and can only be approximated using the available data samples. The goal of generative modelling is to learn a model that can generate new samples that closely resemble the data distribution.

The generated distribution Q is a parametric distribution that the generative model learns to approximate the data distribution P. The parameters of the generated distribution are adjusted during the training process to minimize the difference between the generated samples and the data samples. Common choices for the generated distribution include Gaussian distributions, Bernoulli distributions, or more complex distributions like mixture models and implicit distributions defined by neural networks. To quantify the similarity in distributions, practitioners often use a cost function or a scoring rule such as Maximum Mean Discrepancy (MMD), where it is 0 when distributions P and Q are identical. See Appendix C for more detail on MMD.

2.2. Bayesian Learning. Consider parameters $\boldsymbol{\theta}$ as random variables with a posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{D})$, prior $p(\boldsymbol{\theta})$, likelihood $f(\boldsymbol{D}|\boldsymbol{\theta})$ and data $\boldsymbol{D}$. Using Bayes formula:

$$\pi(\boldsymbol{\theta}|\boldsymbol{D}) \propto p(\boldsymbol{\theta}) f(\boldsymbol{D}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \exp(-\beta C(\boldsymbol{\theta})) \tag{2.1}$$

where β is the reciprocal temperature and $C(\boldsymbol{\theta})$ is a cost function such as MMD.

The goal becomes sampling $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N \sim \pi(\boldsymbol{\theta}|\boldsymbol{D})$ to approximate the predictive distribution with a Bayesian Model Average (BMA). See Appendix B for a visual representation of BMA.

$$p_{\mathrm{BMA}}(y|\boldsymbol{D}) = \int_{\boldsymbol{\theta} \in \mathbb{R}^n} p(y|\boldsymbol{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{D}) \, d\boldsymbol{\theta} \approx \frac{1}{N} \sum_{i=1}^{N} p(y|\boldsymbol{\theta}_i) \tag{2.2}$$

Working in log-space, we use gradient-based stepping methods (e.g., Langevin Monte Carlo) to sample from the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{D})$:

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{D}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) - \beta \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}).$$

The reciprocal temperature β plays a crucial role in shaping the posterior distribution and controlling the exploration-exploitation trade-off during sampling. When β is high (i.e., the temperature is low), the posterior distribution becomes sharper and more peaked around the modes, as shown in Figure D.1(a). This can be useful for focusing inference on specific regions of interest or for identifying modes of the distribution with high confidence, a process known as annealing [6]. Annealing allows the sampling algorithm to gradually converge to the most probable regions of the parameter space as β increases.

On the other hand, when β is low (i.e., the temperature is high), the posterior distribution becomes smoother and more diffuse, as illustrated in Figure D.1(b). This can make the distribution more tractable for sampling algorithms by reducing the number of local maxima and making the distribution less peaked, a process called tempering [7]. Tempering allows for better exploration of the parameter space and can help prevent the sampling algorithm from getting stuck in local optima.

In the context of Stochastic Gradient Langevin Dynamics (SGLD) and Stochastic Gradient Adaptive Langevin Thermostats (SGAdLT), which will be introduced later, the choice of β can be interpreted as controlling the balance between the stochastic gradient noise and the Langevin dynamics. A higher β value will result in a more deterministic update rule, emphasizing exploitation of the current best solutions, while a lower β value will introduce more stochasticity, promoting exploration of the parameter space.

2.3. Quantum Circuit Born Machine (QCBM). Quantum Circuit Born Machine (QCBM), a special subclass of Parameterized Quantum Circuits (PQCs), can be used as an Ansatz [1] for generative modeling. A PQC consists of a sequence of parameterized quantum gates applied to a set of qubits. By adjusting the parameters of the gates, the PQC can generate different quantum states.

In a QCBM, the PQC is used to generate a probability distribution over classical bit strings. According to the Born rule, the probability of measuring a particular bit string x is given by $p_{\boldsymbol{\theta}}(x) = |\langle x|\psi_{\boldsymbol{\theta}}\rangle|^2$, where $|\psi_{\boldsymbol{\theta}}\rangle$ is the quantum state generated by the PQC with parameters $\boldsymbol{\theta}$. By sampling from this distribution, the QCBM can

[1] Ansatz: Assuming a particular structure for the solution and optimizing based on that. E.g Assume a polynomial ansatz $y(x) = ax^2 + bx + c$ for the solution of the ODE $\frac{d^2y}{dx^2} - 4y = 0$ and optimize w.r.t a, b, c.

generate classical data. Figure 2.1 shows an example QCBM with 8 qubits and a circuit depth of 3.

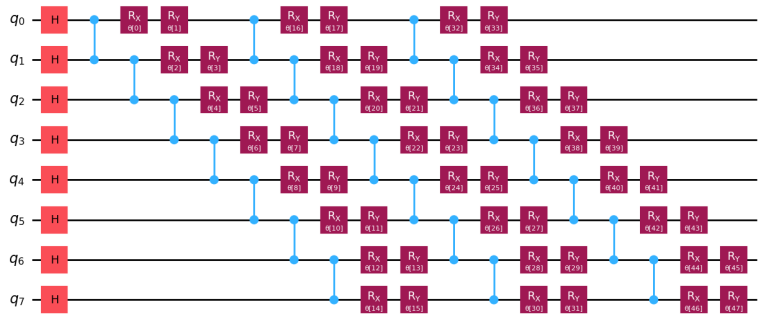


FIG. 2.1. *Quantum Circuit Born Machine with 8 Qubits and a Circuit Depth of 3.*

QCBMs have several potential advantages over classical generative models. First, they can efficiently represent certain probability distributions that are difficult to model classically, such as those with strong correlations or entanglement between variables. Second, they may be more expressive than classical models with a similar number of parameters, due to the exponential size of the Hilbert space.

However, training QCBMs can be challenging due to the following issues:

Reasons why β from 2.2 may be helpful:

- Exponentially many local minima [8]
- Barren Plateau Phenomenon [9]

To address these challenges, we propose using Bayesian learning to train QCBMs. By introducing a prior distribution over the parameters and updating it based on the observed data, we can potentially avoid getting stuck in local minima and escape barren plateaus. The posterior distribution can be approximated using Langevin dynamics, as discussed in Section 2.2.

Fig 2.2 shows an overview of our quantum Bayesian learning approach. The QCBM, denoted by $U(\boldsymbol{\theta})$, generates samples from a probability distribution parameterized by $\boldsymbol{\theta}$. The prior distribution $p(\boldsymbol{\theta})$ is updated based on the observed data $\mathcal{D}$ to obtain the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{D})$. Langevin dynamics is used to sample from the posterior and update the parameters of the QCBM.

The gradients of the cost function $C(\boldsymbol{\theta})$ required for Langevin dynamics can be computed using the parameter-shift rule, which is a finite-difference method for

estimating gradients on quantum circuits. This may add a deterministic bias in the estimated gradients. However, for the purposes of this project, we assume that this is exact as it is proven to be unbiased in [1]. See Appendix E for details.

In summary, QCBMs are a promising approach for generative modeling on quantum computers, but training can be challenging due to the complex cost landscape. By combining QCBMs with Bayesian learning, we aim to develop a more robust and effective training method.

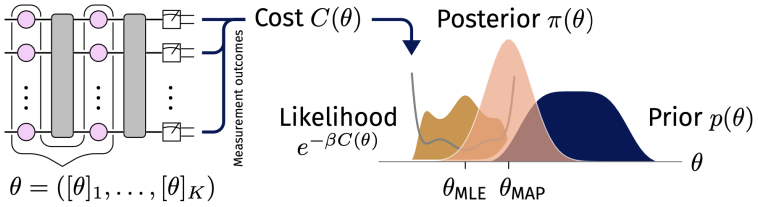


FIG. 2.2. *Overview of quantum Bayesian learning for QCBMs [2].*

3. Stochastic Gradient Langevin Dynamics (SGLD). Langevin Dynamics is used to sample the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{D})$. The parameter space of QCBM is quite high-dimensional, especially as the number of qubits increase. The QCBM in Fig 2.1 has parameter θ that is $2^8 = 64$-dimensional. In generative modelling, we often have many samples of data $\boldsymbol{D}$. It can become quite computationally costly to compute the gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$. Thus, it is beneficial to compute stochastic gradients.

SGLD [10] is based on the following stochastic differential equation:

$$
\begin{aligned}
d\boldsymbol{\theta} &= -\frac{1}{\gamma}\nabla\tilde{C}(\boldsymbol{\theta})dt + \sqrt{\frac{2}{\gamma\beta}}d\boldsymbol{W} \\
&= -\frac{1}{\gamma}\left(\frac{1}{|S|}\sum_{i\in S}\nabla C_i(\boldsymbol{\theta})\right)dt + \sqrt{\frac{2}{\gamma\beta}}d\boldsymbol{W}
\end{aligned}
$$

where $S \subseteq 1, \ldots, n$ is a random subset of indices.

- $\boldsymbol{\theta}$: circuit parameters
- γ: friction coefficent, inverse step-size
- $\tilde{C}(\boldsymbol{\theta})$: stochastic gradient of the cost function
- β: reciprocal temperature
- $d\boldsymbol{W}$: standard Wiener process

4. Gradient Noise Model (GNM). One of the biggest problems about the modelling approach here is that since it is computationally expensive the generative model utilizes the Stochastic Gradient at each step. This introduces bias to the approximations for the gradients. Below Fig 4.1 illustrates the component-wise (marginal) distributions of bias vector. The methodology we used to conduct this analysis was to take 2000 stochastic gradients with a batch size of 200 for each of the 64 components then we obtain the bias value for that stochastic gradient by subtracting the best approximation for the real gradient (i.e. using all samples to approximate the gradient):

$$h_i = \{(\nabla \tilde{C}_{|S_j|}(\boldsymbol{\theta}) - \nabla C(\boldsymbol{\theta}))_i\}_{j=1}^{2000} \text{ for } i = 1, ..., 64.$$

where h_i is the bias vector for the respective component. The histograms just illustrate the distributions of $h_1, \dots h_{64}$.

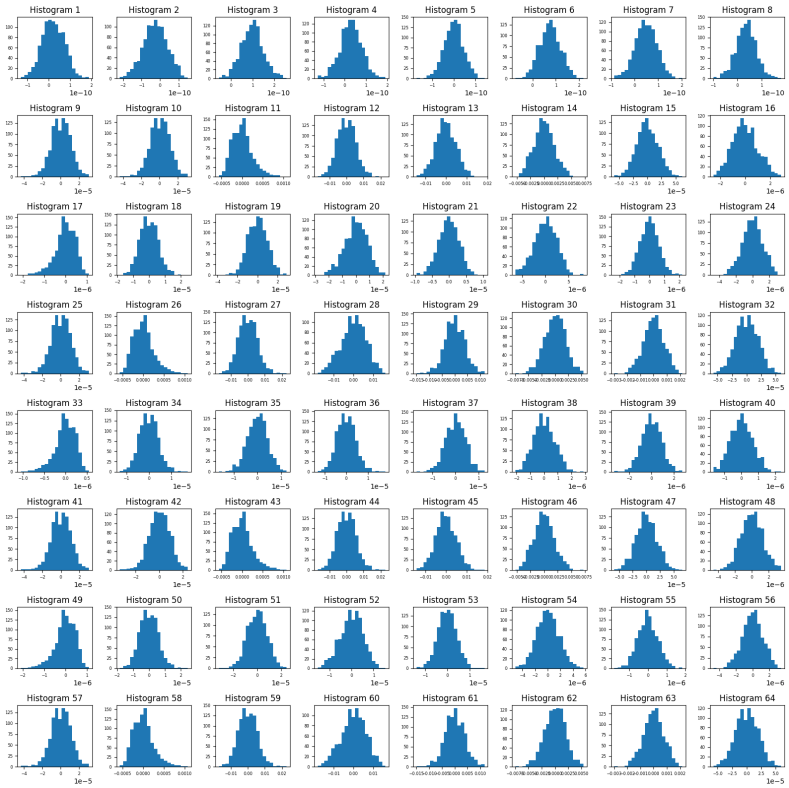


FIG. 4.1. *Gradient Noise Model for a Bimodal Gaussian with 5000 samples and batch size = 200. The component-wise biases appear to resemble the Gaussian distribution, even though some histograms are positively/negatively skewed or seem to have considerably different variations.*

5. Stochastic Gradient Adaptive Langevin Thermostat (SGAdLT).

5.1. Introduction. The Ad-Langevin Thermostat [4] debiases under the **assumption of Gaussian batch noise with constant covariance**.

It is based on the following coupled system of SDEs:

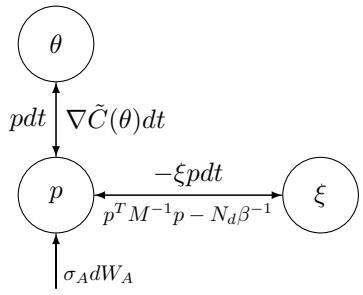
$$d\boldsymbol{\theta} = \boldsymbol{M}^{-1}\boldsymbol{p}dt, \tag{5.1}$$

$$d\boldsymbol{p} = \nabla\tilde{C}(\boldsymbol{\theta})dt - \xi\boldsymbol{p}dt + \sigma_A\sqrt{\boldsymbol{M}}d\boldsymbol{W_A}, \tag{5.2}$$

$$d\xi = \mu^{-1}\left[\boldsymbol{p}^T\boldsymbol{M}^{-1}\boldsymbol{p} - N_d\beta^{-1}\right]dt \tag{5.3}$$

- $\boldsymbol{\theta}$: circuit parameters
- $\boldsymbol{p}$: momentum variable
- $\boldsymbol{M}$: mass matrix
- $\nabla\tilde{C}(\boldsymbol{\theta})$: stochastic gradient of the cost function
- ξ: auxillary variable
- σ_A: additive noise
- $d\boldsymbol{W_A}$: standard Wiener process
- μ: thermal mass
- N_d: number of degrees of freedom
- $\beta = (k_BT)^{-1}$: reciprocal temperature

5.2. Physical Interpretation. Below is a diagram representing the interactions between the variables.



Key interactions:

- ξ acts as the friction on $\boldsymbol{p}$,
- Additive noise, σ_A, injects heat to the system,
- The dynamics of ξ are driven by the thermostat (kinetic energy difference) $(\boldsymbol{p}^T\boldsymbol{M^{-1}}\boldsymbol{p} - N_d\beta^{-1})$.

Negative feedback loop enables adaptive noise dissipation.

5.3. Splitting the Ad-Langevin Thermostat. Since stochastic systems in most of the cases cannot be solved "exactly," splitting methods are often adopted in practice. For instance here, the vector field of the Ad-Langevin can be split into four pieces which are denoted as "A," "B," "O," and "D" [4]:

$$\mathrm{d}\begin{bmatrix}\boldsymbol{\theta}\\ \boldsymbol{p}\\ \xi\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{M}^{-1}\boldsymbol{p}\\ \mathbf{0}\\ 0\end{bmatrix}\mathrm{d}t}_{\mathrm{A}} + \underbrace{\begin{bmatrix}\mathbf{0}\\ -\nabla\tilde{C}(\boldsymbol{\theta})+\sigma\boldsymbol{M}^{1/2}\boldsymbol{R}\\ 0\end{bmatrix}\mathrm{d}t}_{\mathrm{B}}$$

$$+ \underbrace{\begin{bmatrix}\mathbf{0}\\ -\xi\boldsymbol{p}\mathrm{d}t+\sigma_{\mathrm{A}}\boldsymbol{M}^{1/2}\mathrm{d}\boldsymbol{W}_{\mathrm{A}}\\ 0\end{bmatrix}}_{\mathrm{O}} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mu^{-1}(\boldsymbol{p}^T\boldsymbol{M}^{-1}\boldsymbol{p}-N_d\beta^{-1})\end{bmatrix}\mathrm{d}t}_{\mathrm{D}}$$

In this scheme we note that $\mathbf{R}$ appears as opposed to the SDE representation from 5.1 to clear any confusion, here $\mathbf{R}$ is a vector of i.i.d. standard normal random variables [4]. For our experiments, we chose to use the **BADODAB** scheme as used in the Bayesian Logistic Regression example from [4].

6. Experiments. (Code can be found here)

6.1. Datasets.

We use the BARS-and-Stripes [11] Dataset and a Gaussian Mixture model. We used the noiseless simulator Qujax [12] to simulate our experiments.

6.1.1. Gaussian Mixture Model.

Our first empirical data distribution is of sample size 5000 and is a non-symmetric bimodal Gaussian with dimensionality = 1.

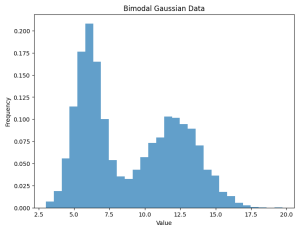


FIG. 6.1. *Bimodal Gaussian*

For our experiments on this dataset, unless otherwise mentioned, the default hyperparameters are listed in Appendix F.

6.1.2. BARS-and-Stripes (BAS).

The Bars-and-Stripes (BAS) dataset is a synthetic dataset commonly used to benchmark generative models, particularly in the quantum computing domain [11] . The dataset consists of images with either vertical bars or horizontal stripes on a grid of pixels, as seen in Figure 6.2. For an $n \times m$ grid, the total number of valid BAS patterns is $N_{BAS} = 2^n + 2^m - 2$. To generate samples from the BAS dataset, we define a target distribution $p(x)$ where $p(x_i) = 1/N_{BAS}$ if x_i is a valid BAS image, and 0 otherwise. The prior distribution is assumed to be uniform, meaning that all valid BAS patterns are expected to be generated with equal probability. The BAS patterns are encoded into the qubits, with each data qubit representing a pixel of the BAS image. By applying a series of quantum gates and sampling the circuit parameters, the QCBM can generate samples that resemble the target BAS distribution.

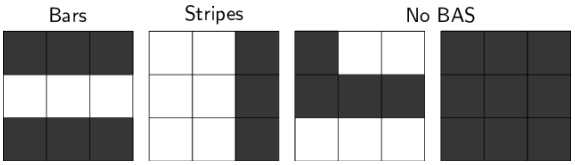


FIG. 6.2. *Bars and Stripes Dataset*

6.1.3. Experiment 1: Expressivity of Circuit. The expressive power of a generative learning machine directly determines how well the generated distribution can match the target distribution. The stronger the expressive power, the smaller the dissimilarity between the two distributions will be. In this experiment, we investigated the relationship between the expressivity of the quantum circuit by varying the number of qubits $Q \in \{2, 4, 6, 8, 10\}$ and the circuit depth $D \in \{1, 2, 3, 4, 5\}$. The quantum circuits are composed of D blocks, where each block implements a unitary operator $U(\theta_i)$. A unitary operator $U(\theta) = \prod_{i=1}^{D} U(\theta_i)$ is applied to Q input qubits. The experiment was conducted with the following hyper-parameters: number of training steps $N = 5000$, step size $h = 0.1$, and reciprocal temperature $\beta = 100$. The initial parameters of the quantum circuit were randomly initialized within a small range $(-0.001/\pi, 0.001/\pi)$ to avoid vanishing or exploding gradients. Figure A.2 displays the Predictive Distributions with a burn-in of 1000. We observe that for $Q = 2$, there is no visible Predictive Distribution $p(X)$, while for $Q = 4$, the $p(x)$ends at $x = 15$. This is because increasing Q increases the $p(x)$ space sampling grid points by a factor of 2^Q, namely $p(x) \to [0, 2^Q - 1]$, given that the grid points are evenly spaced. However, due to limitations of the Qujax package used for the simulation, increasing the number of qubits beyond a certain point may not further improve expressivity. To overcome this, we propose mapping the qubits to a targeted sub-domain of the range, such as $[4, 20]$, to allow for a more focused generative model of the target distribution. Regarding the circuit depth D, we observe that increasing it improves expressivity but also increases the computational cost per iteration, since the number of parameters grows by $|\theta| = 2Q(D + 1)$.

6.1.4. Experiment 2: Adding artificial noise σ to gradients. Assume $\tilde{C}(\boldsymbol{\theta}) = C(\boldsymbol{\theta}) + \alpha\boldsymbol{\varepsilon}$ where α is multiplicative factor and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{I})$. We assume in this case that the noise induced by taking stochastic gradients is equivalent to adding standard Gaussian noise. This gradient noise model is Gaussian with identity covariance which satisfies the assumption of Adaptive Langevin Thermostat 5. We would hence like to use this experiment as a proof of concept for the Adaptive Langevin Thermostat. First of all, take a look at Fig A.3. The marginal distribution of noises are all standard normal Gaussians i.e., equal scale equal to one and zero mean. As an extreme case, we run experiments with $\alpha = 10$ to demonstrate the benefit the thermostat has on sampling that overdamped Langevin dynamics does not offer. This is an extremely high gradient noise regime. Referring to Figures A.4, A.6, it is apparent that SGLD's cost curve (Maximum Mean Discrepancy) is unable to sample from a region of low cost (or high probability). Thus, the predictive distribution computed is quite flat. It is nearly a uniform predictive distribution which is quite poor. Interestingly though, it is able to generate two small modes similar to our data distribution. On the other hand, quite pleasantly, the cost curve appears to be sampling from a region of low cost (high probability) and attains a fairly accurate predictive distribution. It does however overfit to some extent as it contains more than two peaks as indicated in Figures A.5, A.7. It is likely overfitting to the kinks in the data distribution as indicated in the histogram as opposed to learning the underlying signal. This is perhaps due to the low temperature i.e., high value of $\beta = 100$ which has the effect of annealing the distribution resulting in overconfident predictions. We will look into this effect of β in the following experiment.

6.1.5. Experiment 3: The effect of β. As indicated in Fig D.1, increasing β has the effect of annealing the distribution while decreasing β has the effect of tempering the distribution. This forms a tradeoff between exploration and exploitation as discussed in Sec 2.2. Referring to Fig A.8 and A.9, we observe the loss curve of SGLD (row 1) and SGAdLT (row 2) where we take stochastic gradients of batch size 200. Note that we are no longer adding artificial noise to the gradients. The gradient noise model now looks like Fig 4.1 which looks approximately Gaussian with non-constant covariance, thus violating the assumptions of Adaptive Langevin Thermostat. However, according to [13], it is possible for the method to work for even non-Gaussian and non-constant covariance with a minor modification. We decide to proceed with the experiments despite the violation in assumption by varying $\beta \in \{1, 100, 1000, 10000\}$.

As we increase β, SGLD begins to converge very quickly to nearly zero cost for $\beta = 10000$, with very little variation. This implies that it acts as an optimizer for very large β. For $\beta = 1$, which is the standard temperature regime, it is unable to sample from a region of high probability. Note the corresponding predictive distributions in Fig A.10 and A.11. For $\beta = 1$ and 100, the predictive distribution appears to be nearly uniform. The predictive distribution for $\beta = 1000$ begins to show some, albeit poor, fitting to the data distribution. At $\beta = 10000$, it appears to give an excellent predictive distribution. However, this is most likely because it is sampling just around the maximum a posteriori estimate θ_{MAP}. This is an insufficient exploration of the parameter space θ. We cannot rely on it to provide us with accurate uncertainty estimates. When we baseline against Stochastic Gradient Adaptive Langevin Thermostat, we observe that for $\beta = 1$, it is unable to sufficiently explore a local basin of attraction, thus yielding a uniform predictive distribution. However, for $\beta = 100$ and higher, we observe that it begins to explore to some extent around the local minima, thus yielding reasonable predictive distributions. However, these appear to be overfitting to the data distribution. This is a standard effect of annealing. Moreover, we have chosen an arbitrary set of default parameters for $\mu^{-1}, \sigma_A, etc.$ It would be beneficial to first find the accurate values for these hyper-parameters. On a positive note however, we observe that SGAdLT is able to begin exploring local minima for lower β than for SGLD. This is a promising result!

6.1.6. Experiment 4: Varying batch size to see how the GNM varies. As indicated in Sec 4, the smaller our batch size, the noisier our stochastic gradients are. The gradient noise model in Fig 4.1 is of batch size 200 for a specific configuration of parameters θ_{init}. We wish to study what kind of distribution is formed by the errors of the stochastic gradients and how they vary with batch size. The Adaptive Langevin Thermostat method assumes zero-mean Gaussian noise with constant covariance. We hypothesize that the scale of the error distributions should get smaller as the batch size increases since there is less uncertainty and hence less variance. We varied our batch size $S \in \{100, 200, 500, 1000, 2000\}$.

To analyze the gradient noise model, we plot the marginal histograms with 2000 samples each. Due to our quantum circuit having 64 parameters to optimize over, we only present batch size 100 and 2000 to represent the disparity in Fig **??**. The rest can be found in Experiment 4 notebook of the Github repository. After assessing the grids thoroughly, we were unable to find a discernible pattern in the change of the scales as the batch size varied. It is unclear to us why this is the case. Some of the scales of the parameters remain the same, some increase or decrease arbitrarily irrespective of whether the batch size is decreasing or increasing. However, we must recall that these marginals only represent the joint distribution of the gradient noise if they are all independent. In the case of Gaussian distributions, being uncorrelated implies independence of the marginals. First of all, it is unclear whether this distribution is Gaussian. If we make the assumption that it is Gaussian based on a qualitative visual assessment of the histograms, the arbitrary change of scale with batch size indicates that the components of the gradient noise are correlated. Thus, it would make more sense to validate whether the joint distribution of the gradient noise model is Gaussian with tests like the Kolmogorov-Smirnov test. However, this is expensive in high-dimensions (i.e., 64) and due to the time constraint, we leave this for future work. The stabilizing thermostat term ξ can also be studied more extensively to derive key insights.

We must also note that many of the scales have order of magnitudes in the range of $[-6, -3]$, which is very small. This means that the noise introduced by the stocastic gradients is quite small. This is most likely because our dataset is 1-dimensional. The bias introduced by the stochastic gradients may thus be very small as well.

6.1.7. Experiment 5: The effect of stepsize h. In this experiment, we investigated the effect of the stepsize h on the performance of SGLD and SGAdLT. We varied $h \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$ while keeping other hyper-parameters fixed at: number of qubits $Q = 8$, circuit depth $D = 3$, reciprocal temperature $\beta = 100$, training steps $N = 5000$, and batch size = 200. Figure **??** shows the MMD loss curves for SGLD and SGAdLT at different step sizes. For smaller h (0.01 and 0.05), both methods exhibit slower convergence. As h increases to 0.1, convergence speed improves. However, large step sizes (0.5 and 1.0) lead to instability and fluctuations, more pronounced in SGLD. The choice of h in SGLD and SGAdLT controls the balance between stochastic gradient noise and Langevin dynamics, similar to the role of β. A smaller h promotes exploitation of the current best solutions, while a larger h encourages exploration of the parameter space. In the context of sampling, h controls the "energy landscape" of the posterior distribution. Smaller h values result in a more rugged landscape with well-defined minima, emphasizing the exploitation of local optima. Larger h values lead to a smoother landscape, allowing for better exploration and preventing the algorithm from getting stuck in suboptimal regions. SGAdLT generally achieves lower MMD values and faster convergence than SGLD across different step sizes due to its adaptive noise adjustment based on the observed gradient noise. The optimal h depends on the problem and may require tuning. Further work may include using an adaptive step size control.

6.1.8. Experiment 6: Thermal Mass μ vs Additive noise σ_A. Previously, we arbitrarily set $\mu^{-1} = \sigma_A = 1$. We perform a grid search to determine the optimal hyperparameter configuration (μ^{-1}, σ_A) at $\beta = 100$. Thermal mass μ regulates how particles interact with the heat bath, affecting equilibration rate. Additive noise σ_A refers to random fluctuations added to the particle dynamics to mimic thermal fluctuations from the environment. Dynamically adjusting these two hyperparameters optimizes sampling efficiency and convergence in molecular dynamics simulations. We vary $(\mu^{-1}, \sigma_A) \in \{0.01, 0.1, 1, 10\}^2$.

Figure A.18 contains the cost curves and Bayesian model averages in a grid format. Based on the cost curves, it appears that as σ_A increase while μ^{-1} is kept constant (Row 1), there is more noise in the system, thus resulting in an inability to explore a region of high probability. As we increase μ^{-1} (i.e., decrease the thermal mass), we coerce exploration of a region of low cost. There is thus a tradeoff between exploration and exploitation similar to that of β. We argue that for our purposes, the configurations $(\mu^{-1}, \sigma_A) = (0.1, 1.0)$ or $(1.0, 1.0)$ achieves the desired tradeoff for exploration while remaining in a region of low cost. We then compare their Bayesian Model Averages and conclude that the configuration $(\mu^{-1}, \sigma_A) = (0.1, 1.0)$ is more favorable than $(\mu^{-1}, \sigma_A) = (1.0, 1.0)$ because it overfits to kinks in the histogram. We proceed to use this configuration in future experiments. This decision, we admit, is somewhat subjective and may benefit from further objective and empirical analysis.

6.1.9. Experiment 7: Preconditioning using mass matrix M. From Experiment 4, we know that the Gradient noise model has different variance along each component and are possibly correlated. To remedy this, we can precondition the algorithm using mass matrix $\boldsymbol{M}$ [14]. The choice of $\boldsymbol{M}$ can have drastic effects on convergence and the quality of the samples, particularly for noise distributions which are very anisotropic. A typical choice of $\boldsymbol{M}$ is the covariance of the noise distribution in the convex setting. The effect of $\boldsymbol{M}^{-1}$ is to make the dynamics affine invariant. In summary, we should expect Langevin Dynamics to be most efficient when applied to roughly isotropic distributions with roughly Gaussian tail behavior. We decide to use a batch size of 200 and approximate its stochastic gradient noise distribution with a diagonal matrix $\boldsymbol{M}$ where

$$\boldsymbol{M} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{64} \end{bmatrix}$$

where σ_i is the variance of histogram h_i for batch size 200 from Fig 4.1.

We set $\mu^{-1} = 0.1$ and $\sigma_A = 1.0$ based on Experiment 6 and re-run our experiment for the bimodal Gaussian with $\beta = 100$. Regrettably, this did not yield favorable results as indicated in Fig A.19. As indicated by the figure, it never explores a region (basin) of high probability (low cost). It is unclear why this might be the case. Potential reasons may include:

- The diagonal approximation to the covariance matrix of the noise distribution is a crude one, due to the correlation between components.
- The scales of the histograms are of order of magnitude between $[-6, -3]$ which may result in numerical errors when performing operations with the inverted matrix such as $\boldsymbol{M}^{-1}\boldsymbol{p}$.
- We assumed that the gradient generated by the parameter shift rule in Appendix E is exact, which may not be the case, resulting in erroneous variances σ_i.

6.2. BARS-and-STRIPES. To evaluate the performance of the QCBM on the BAS dataset, we conducted experiments on 2x2 and 3x3 grid sizes. The QCBM was trained using the SGAdLTalgorithm with a step size of $h = 0.5$ and 20,000 training steps. The reciprocal temperature β was set to 1000, the inverse thermal mass μ^{-1} was set to 10, and the additive noise parameter σ_a was set to 0.1.

For the 2x2 case, the total number of valid BAS patterns is $N_{BAS} = 6$, while for the 3x3 case, $N_{BAS} = 14$. The goal of the QCBM is to learn the probability distribution of the valid BAS patterns and generate samples that closely resemble the target distribution.

After training, the QCBM parameters corresponding to the minimum MMD cost were selected as the final parameters. Using these parameters, we generated samples from the QCBM and visualized them in a 2x2 grid (Figure A.22) and a 3x3 grid (Figure A.25). The corresponding probabilities of each generated sample were also calculated and displayed.

The results showed that the QCBM, trained using SGAdL, was able to generate valid BAS patterns in some cases, while in other cases, the generated patterns were close to the valid ones. However, further hyperparameter analysis and tuning could potentially improve the results for this dataset.

Upon observing the MMD loss plot, we noticed a rapid descent at the beginning, followed by a relative plateauing for the rest of the plot. The MMD did not reach close to zero as desired, indicating that the QCBM may have encountered challenges in fully capturing the target distribution. One hypothesis for this behavior is that the sampling particle may have gotten trapped in a local minimum during the training process.

To address this issue, a potential future work could involve implementing a parallel tempering algorithm [15]. In this approach, multiple sampling particles are run simultaneously with different hyperparameters. After a certain number of iterations, the hyperparameters are swapped between the particles. The desired behavior of this algorithm is that some particles can jump between different basins in the loss landscape, while other particles can explore the local minima within each basin. This approach could help the QCBM escape local minima and improve its ability to learn the target distribution.

7. Conclusions. In this report, we have reviewed the application of Bayesian inference to generative modelling using QCBMs. Our primary focus was to investigate the effectiveness of the SGAdLT in addressing the existing challenges posed by the stochastic gradient noise in these models. By conducting extensive experiments on different types of datasets such as the simpler Bimodal Gaussian data and the more complex BARS-and-STRIPES dataset, we aimed to provide valuable insights into the performance and utility of SGAdLT as an alternative to sampling methods used prior in the Quantum Machine Learning space.

The results from our experiments demonstrated that SGAdLT exhibits promising capabilities in mitigating the bias introduced by stochastic gradients. We observed that SGAdLT can effectively explore local cost minima at lower β values compared to the SGLD, the recently proposed sampling algorithm for this type of model. These findings highlight the potential of SGAdLT to improve the sampling process and enhance the overall performance of Quantum Generative Models in general.

Throughout the experimentation phase, we also analysed the impact of various hyper-parameters on the performance and generalisation of SGAdLT. We mainly investigated the effects of the reciprocal temperature (β) and gradient batch size S. Our study of the hyper-parameters followed a key theme: "exploitation versus exploration while sampling", by tuning the hyper-parameters in a specific configuration one can drive SGAdLT to explore high probability posterior landscapes (exploitation), however if the task requires the parameter space to be extensively explored one can also choose a configuration of hyper-parameters to achieve the desired effect.

It is important to note that in our experiments, we employed uniform priors for the Bayesian inference/learning process. However as suggested in [2] exploring the use of Laplace could potentially come with further enhancements to the performance of SGAdLT as it was a great choice to use in conjunction with SGLD. We assume the main benefit of incorporating the Laplace Prior to our model would be better generalisation and regularisation by imposing sparsity.

Moreover, while our experiments were conducted on simulated quantum circuits via the use of Qujax [12], it is critical for us to extend this research to actual quantum hardware. Running our model on real quantum devices would allow us to assess the performance and scalability in the presence of hardware noise and see how well SGAdLT works for mitigating noise that does not fit into the prior assumptions (non-

Gaussian, non-constant covariance). This understanding could lead to the development of more bespoke variations to use for quantum machine learning but before that we might also benefit from exploring the use of modified variations of Ad-Langevin Thermostat, such as the one proposed by Sekkat and Stoltz [13].

Another note about noise is that it is an important aspect of the work that warrants further research is the gradient noise model. Our experiments highlighted the significance of understanding the characteristics and behaviour of gradient noise in QCBMs. By conducting a more-in depth analysis of the GNM, we can gain valuable insights into its impact on the convergence and stability of SGAdLT. These algorithmic advancements may help in addressing specific challenges encountered in quantum generative modeling, such as the barren plateau phenomenon and the presence of local minima in the cost landscape.

In conclusion, our research showcases the immense potential of using a Bayesian framework within quantum machine learning, particularly in the context of quantum generative modeling using QCBMs. The application of SGAdLT demonstrates promising results in enhancing the robustness and performance of these models in the presence of stochastic gradient noise. However, this work also highlights the need for further research and exploration in several key areas, including the incorporation of Laplace priors, simulation on actual quantum hardware, in-depth analysis of the gradient noise model, and the investigation of modified algorithms. By addressing these aspects, we can continue to push the boundaries of quantum generative modeling and unlock its full potential for various practical applications. As quantum computing technologies continue to advance, the integration of Bayesian methods with quantum machine learning will hopefully play a pivotal role in shaping the future of this exciting field.

REFERENCES

[1] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6), December 2018.

[2] Samuel Duffield, Marcello Benedetti, and Matthias Rosenkranz. Bayesian learning of parameterised quantum circuits. *Machine Learning: Science and Technology*, 4(2):025007, Apr 2023.

[3] Neil K. Chada, Benedict Leimkuhler, Daniel Paulin, and Peter A. Whalley. Unbiased kinetic langevin monte carlo with inexact gradients, 2023.

[4] Benedict Leimkuhler and Xiaocheng Shang. Adaptive thermostats for noisy gradient systems. *SIAM Journal on Scientific Computing*, 38(2):A712–A736, Jan 2016.

[5] Maria Teresa Baldassarre, Danilo Caivano, Berenice Fernandez Nieto, Domenico Gigante, and Azzurra Ragone. The social impact of generative ai: An analysis on chatgpt. In *Proceedings of the 2023 ACM Conference on Information Technology for Social Good*, pages 363–373, 2023.

[6] Aram W. Harrow and Annie Y. Wei. *Adaptive Quantum Simulated Annealing for Bayesian Inference and Estimating Partition Functions*, page 193–212. Society for Industrial and Applied Mathematics, January 2020.

[7] Nanyang Ye, Zhanxing Zhu, and Rafal K. Mantiuk. Langevin dynamics with continuous tempering for training deep neural networks, 2017.

[8] Zhan Yu, Qiuhao Chen, Yuling Jiao, Yinan Li, Xiliang Lu, Xin Wang, and Jerry Zhijian Yang. Provable Advantage of Parameterized Quantum Circuit in Function Approximation. 10 2023.

[9] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), November 2018.

[10] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[11] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Physical Review Research*, 2(3), July 2020.

[12] Samuel Duffield, Gabriel Matos, and Melf Johannsen. qujax: Simulating quantum circuits with JAX. *Journal of Open Source Software*, 8(89):5504, September 2023.

[13] Inass Sekkat and Gabriel Stoltz. Quantifying the mini-batching error in bayesian inference for adaptive langevin dynamics, 2023.

[14] James M Flegal, Kshitij Khare, and Yan Zhou. Preconditioned langevin monte carlo for bayesian inference. *arXiv preprint arXiv:1905.11503*, 2019.

[15] Rohitash Chandra, Konark Jain, Ratneel V. Deo, and Sally Cripps. Langevin-gradient parallel tempering for bayesian neural learning. *Neurocomputing*, 359:315–326, 2019.

Appendix A. Figures for Experiments.

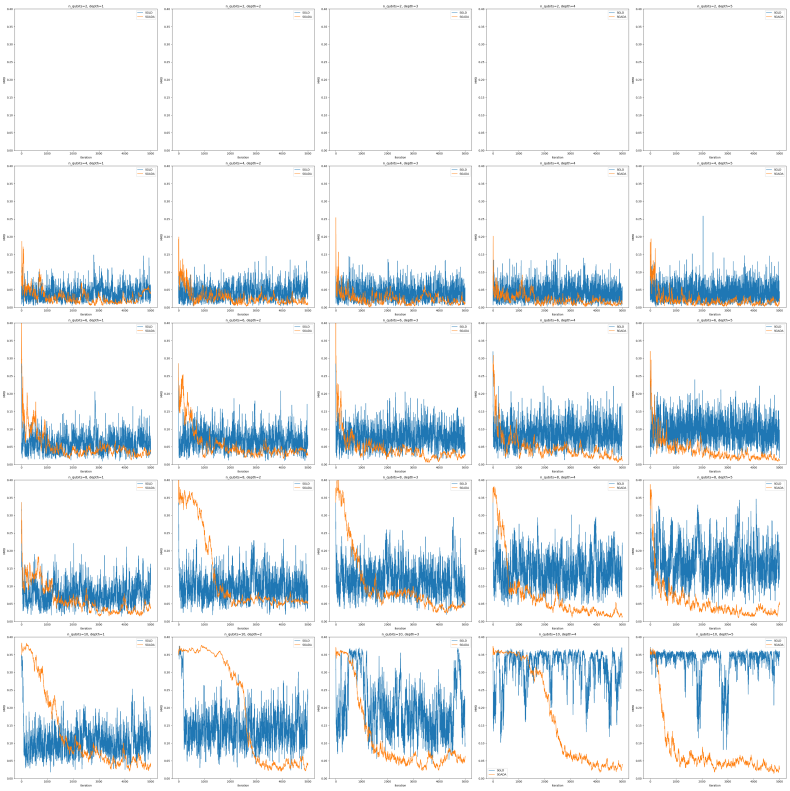


FIG. A.1. *Maximum Mean Discrepancy (MMD) convergence plots for SGLD (Blue) and SGADA (Orange) algorithms across different numbers of qubits(Rows, 2-10) and circuit depths (columns 1-5).*

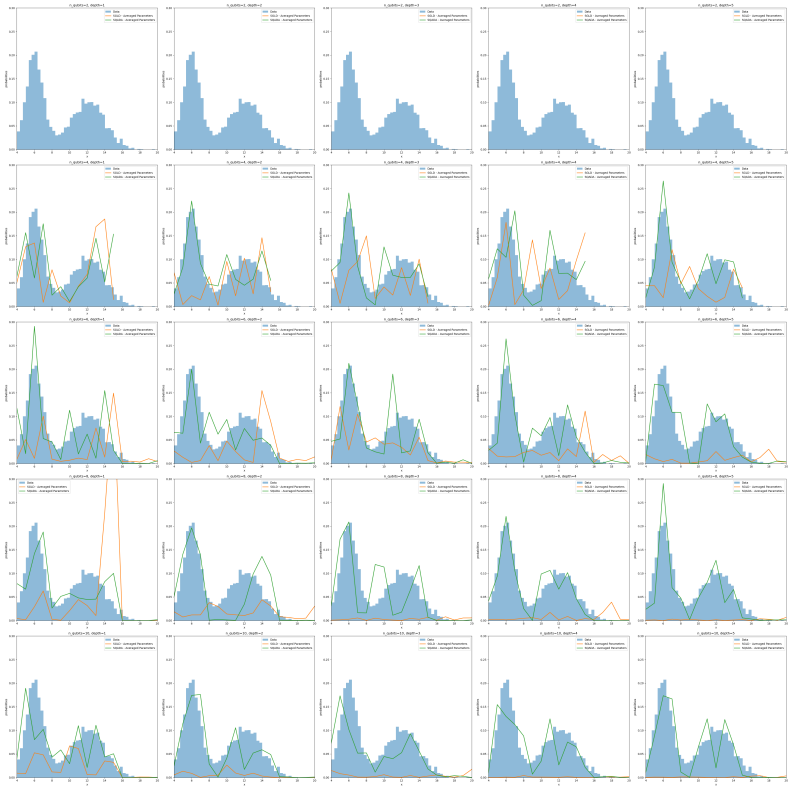


FIG. A.2. *Probability distributions of the generated samples for SGLD (Orange) and SGADA (Green) algorithms across different numbers of qubits(Rows, 2-10) and circuit depths (columns 1-5).*

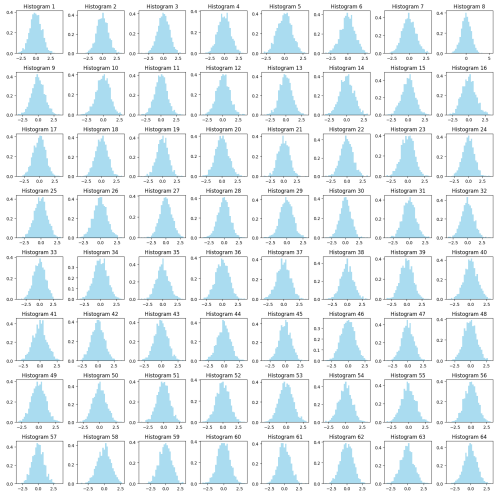


FIG. A.3. *Gradient noise model for artificial noise*

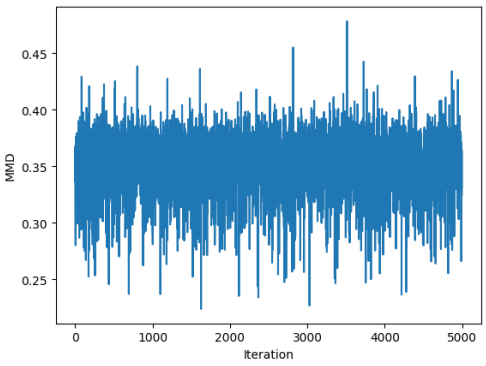


FIG. A.4. *Cost Curve for* ***SGLD***

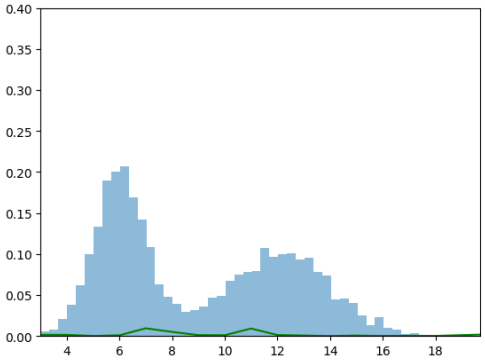


FIG. A.5. $p(y|\boldsymbol{D}) \approx \frac{1}{N}\Sigma_{i=1}^{N} p(y|\boldsymbol{\theta}_i^{SGLD})$

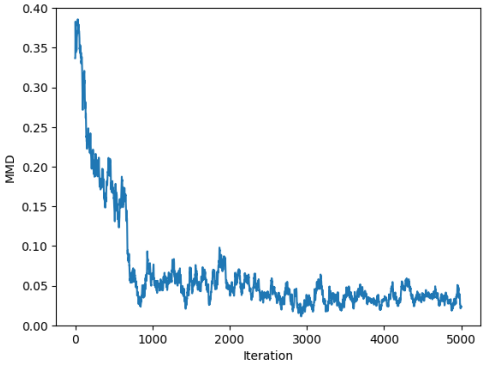


FIG. A.6. *Cost Curve for* ***SGAdLT***

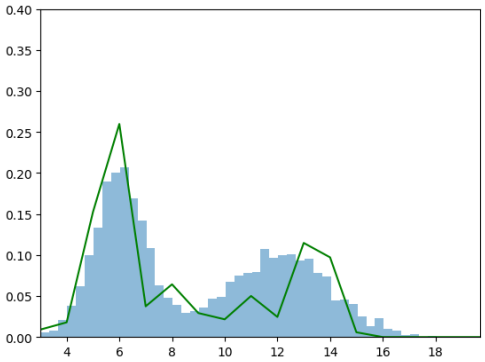


FIG. A.7. $p(y|\boldsymbol{D}) \approx \frac{1}{N}\Sigma_{i=1}^{N} p(y|\boldsymbol{\theta}_i^{SGAdLT})$

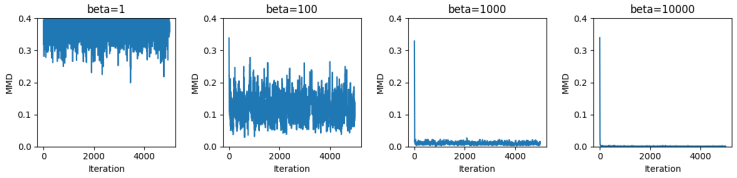


FIG. A.8. *(a) Cost Curves for SGLD*

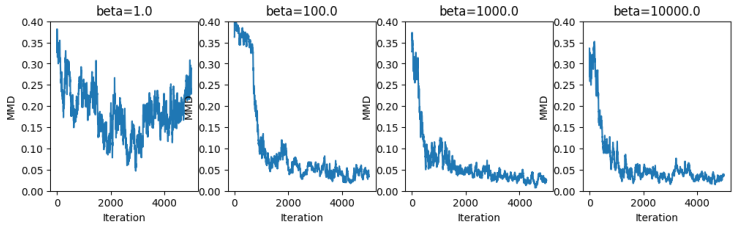


FIG. A.9. *(b) Cost Curves for SGAdLT*

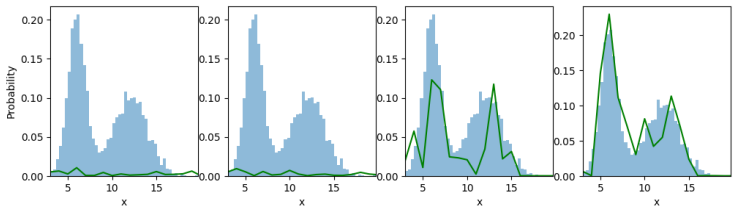


FIG. A.10. *Predictive Distribution $p(y|\boldsymbol{D})$ for SGLD*

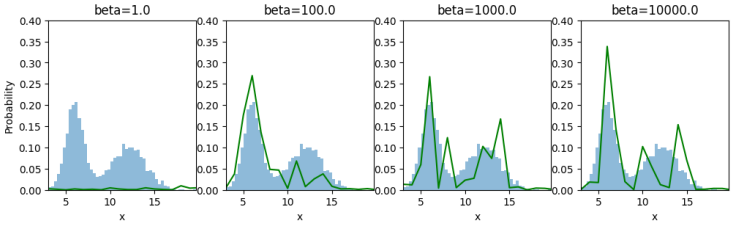


FIG. A.11. *Predictive Distribution $p(y|\boldsymbol{D})$ for SGAdLT*

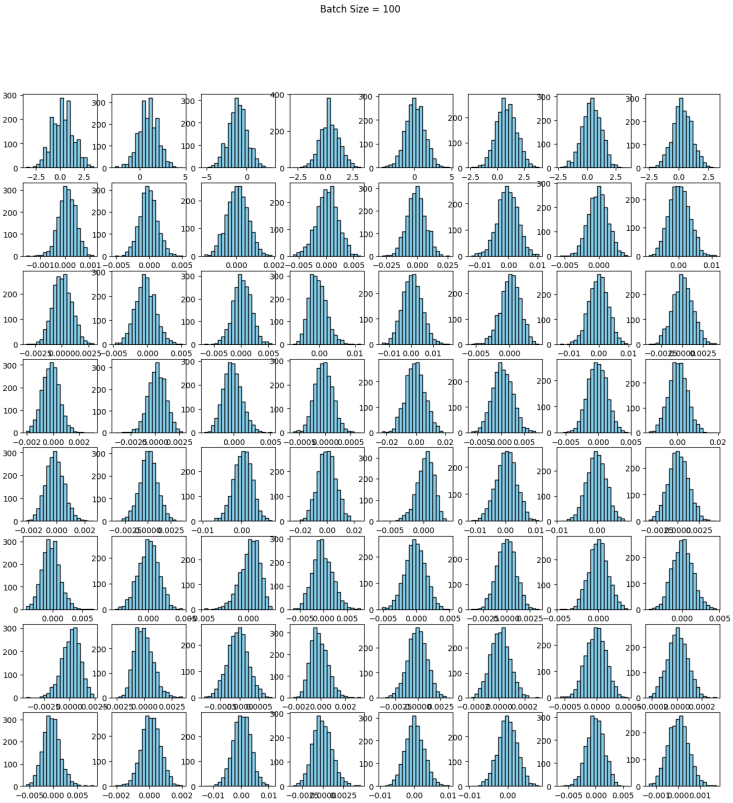


FIG. A.12. *Batch Size = 100*

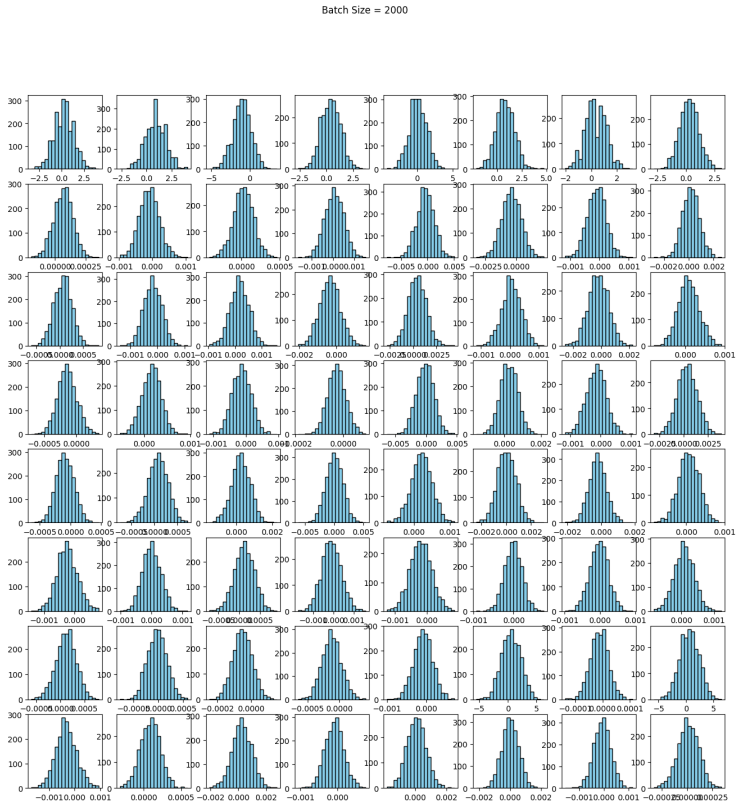


FIG. A.13. *Batch Size = 2000*

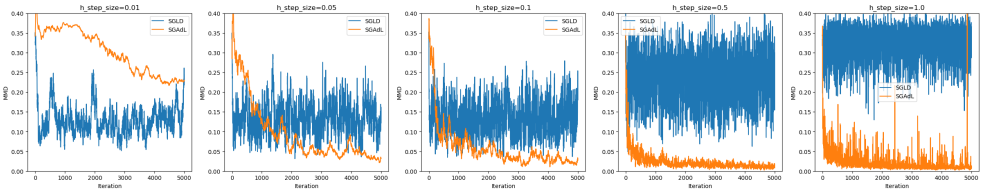


FIG. A.14. *Maximum Mean Discrepancy (MMD) convergence plots for SGLD (Blue) and SGADA (Orange) algorithms across different range of step size h, (0.01, 0.05, 0.1, 0.5, 1)*

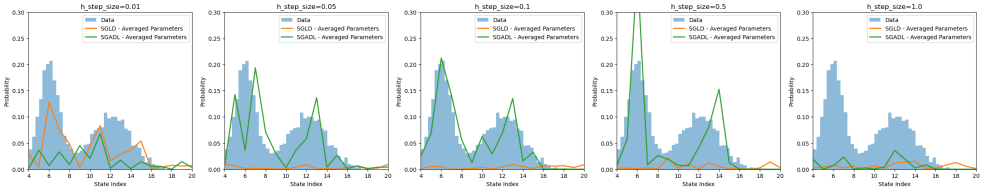


FIG. A.15. *Probability distributions of the generated samples for SGLD (Orange) and SGADA (Green) algorithms across different range of step size h, (0.01, 0.05, 0.1, 0.5, 1)*

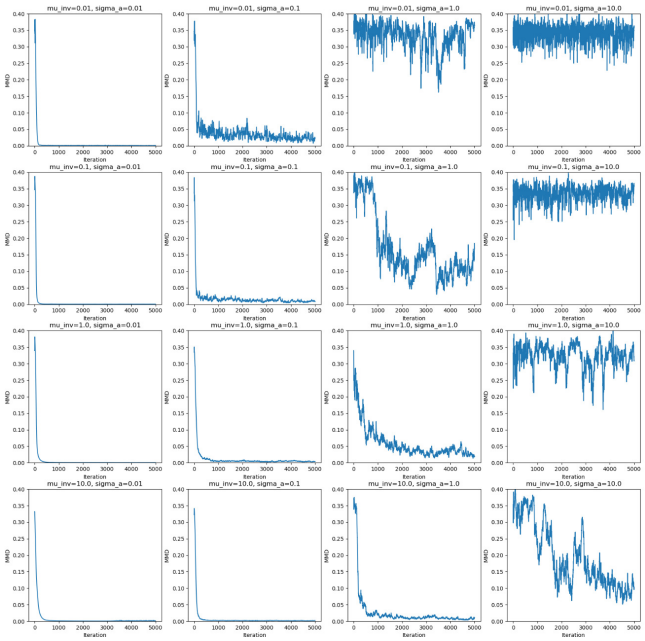


FIG. A.16. *Maximum Mean Discrepancy Curves*

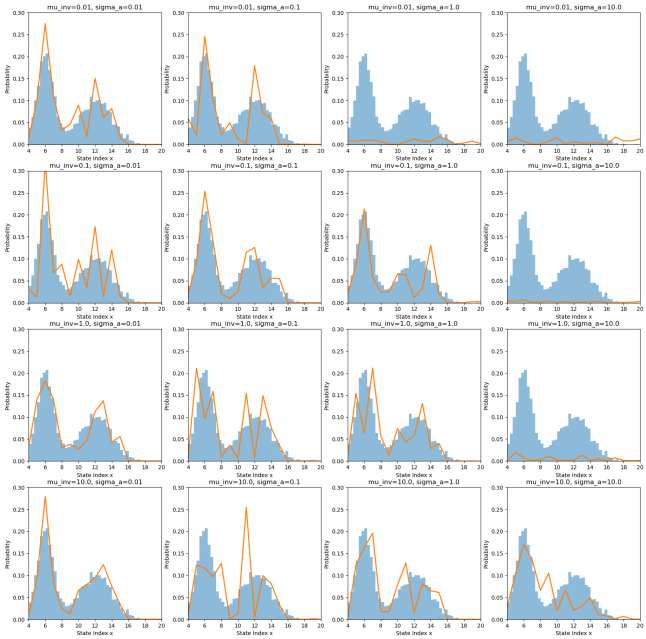


FIG. A.17. *Bayesian Model Averages*

FIG. A.18. *Thermal Mass μ vs Additive Noise σ_A for* $\{0.01, 0.1, 1, 10\}$

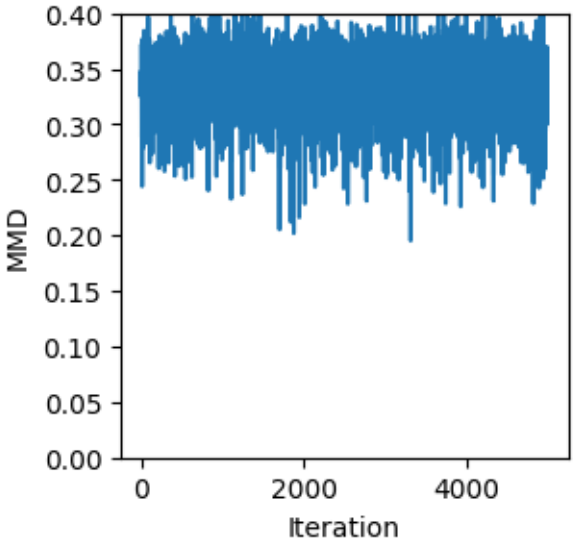


FIG. A.19. *Loss curve for preconditioned SGAdLT*

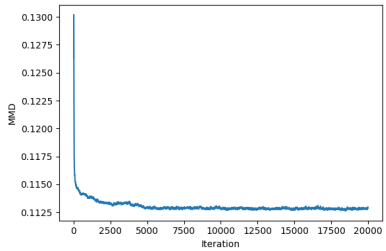


FIG. A.20. *MMD plot*

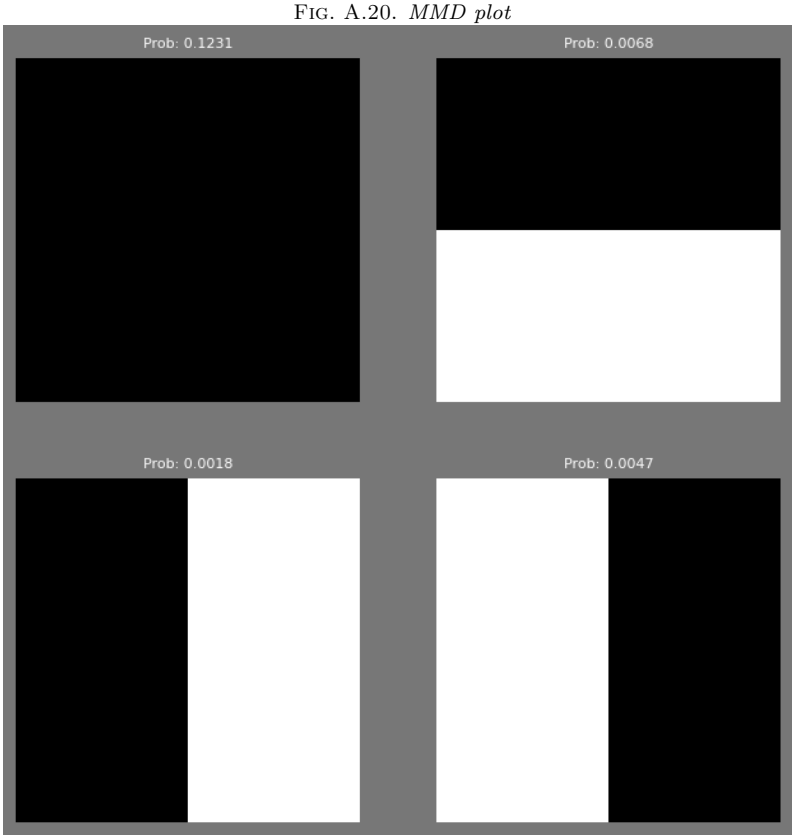


FIG. A.21. *Generated samples with probabilities*

FIG. A.22. *Comparison of MMD plot and generated samples with probabilities for the 2x2 Bars and Stripes dataset using SGAdL. (a) The MMD plot shows the convergence of the SGAdLTalgorithm over iterations. (b) The generated samples demonstrate the learned probability distribution of the Bars and Stripes patterns.*

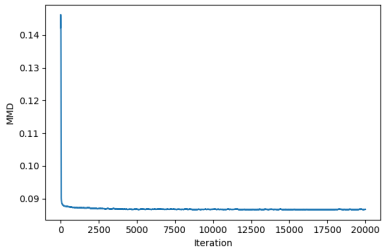


FIG. A.23. *MMD plot*

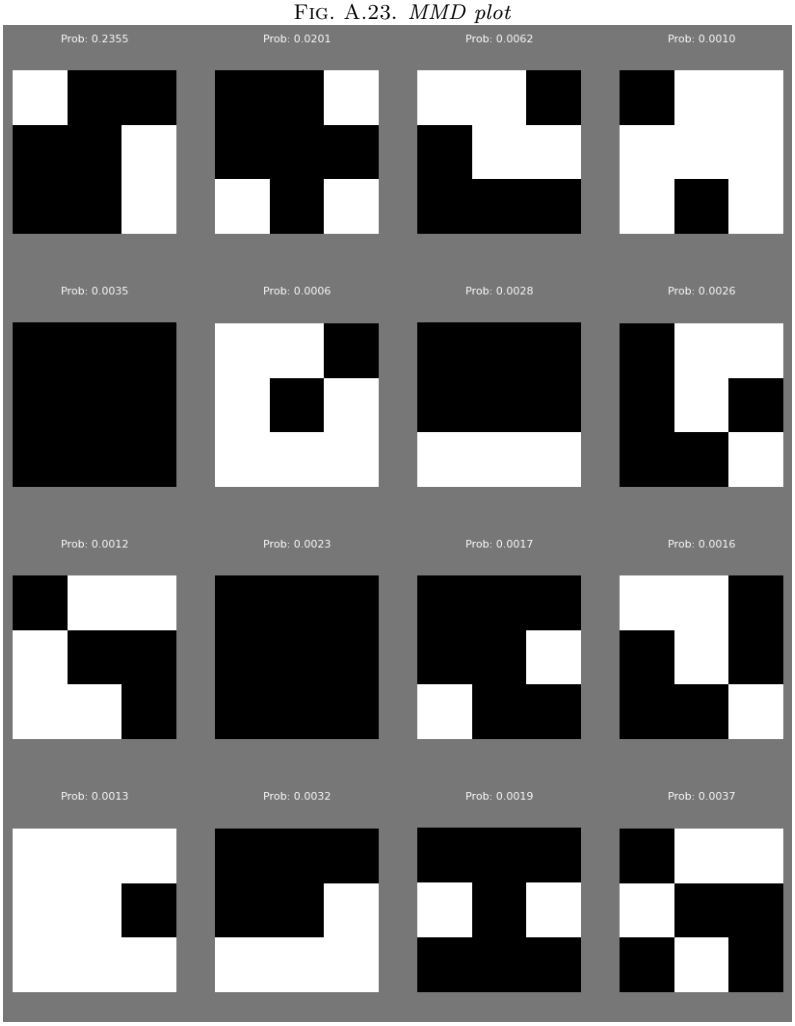


FIG. A.24. *Generated samples with probabilities*

FIG. A.25. *Comparison of MMD plot and generated samples with probabilities for the 3x3 Bars and Stripes dataset using SGAdL. (a) The MMD plot shows the convergence of the SGAdLTalgorithm over iterations. (b) The generated samples demonstrate the learned probability distribution of the Bars and Stripes patterns.*

Appendix B. Bayesian Model Averaging.

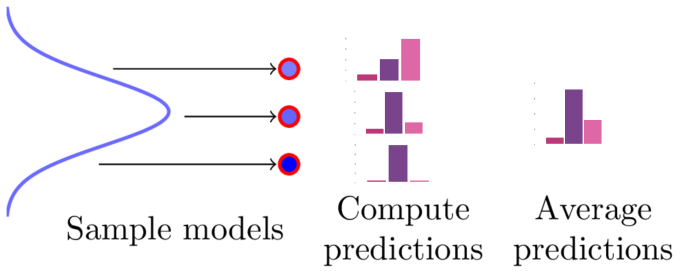


FIG. B.1. *Bayesian Model Averaging (BMA) procedure. The models w are sampled N times leading to predictions $p(y|w_i, x)$ which are uniformly averaged to give $p_{BMA}(y|x, D)$*

Appendix C. Maximum Mean Discrepancy.

The Maximum Mean Discrepancy (MMD) is used as a cost function by computing the difference in means of the distributions P and Q lifted to feature space H_k.

$$\begin{aligned} MMD^2(P,Q) &= \|\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q\|^2_{H_k} \\ &= \langle \boldsymbol{\mu}_P, \boldsymbol{\mu}_P \rangle_{H_k} + \langle \boldsymbol{\mu}_Q, \boldsymbol{\mu}_Q \rangle_{H_k} - 2\langle \boldsymbol{\mu}_P, \boldsymbol{\mu}_Q \rangle_{H_k} \\ &= \boldsymbol{E}_P[k(X,X')] + \boldsymbol{E}_Q[k(Y,Y')] + \boldsymbol{E}_{P,Q}[k(X,Y)] \end{aligned}$$

where $k(\boldsymbol{z}, \boldsymbol{z'}) = e^{-\frac{(\boldsymbol{z}-\boldsymbol{z'})^2}{2\sigma^2}}$ is Gaussian kernel with bandwidth σ. We chose this to be the median of the dataset.

Appendix D. Reciprocal Temperature β.

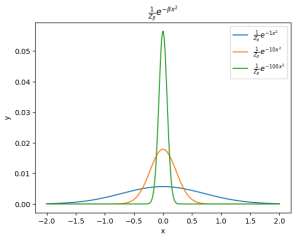


FIG. D.1. *Effect of β*

Appendix E. Parameter-Shift Rule for Gradient Estimation.

To perform gradient-based optimization of QCBMs, we need to compute the gradients of the loss function with respect to the circuit parameters. However, unlike

classical neural networks, quantum circuits do not allow for direct back-propagation of gradients through the gates. Instead, we can use the parameter-shift rule [2], which is a finite-difference method for estimating gradients on quantum circuits.

Consider a parameterized quantum circuit $U(\theta)$ that consists of a sequence of parameterized gates $U_i(\theta_i)$, where $\theta = (\theta_1, \ldots, \theta_n)$ is the vector of parameters. The parameter-shift rule states that the gradient of the expectation value of an observable O with respect to a parameter θ_i can be estimated as:

$$\frac{\partial \langle O \rangle}{\partial \theta_i} = \frac{1}{2} \left(\langle O \rangle_{\theta_i + \frac{\pi}{2}} - \langle O \rangle_{\theta_i - \frac{\pi}{2}} \right), \tag{E.1}$$

where $\langle O \rangle_{\theta_i \pm \frac{\pi}{2}}$ denotes the expectation value of O when the parameter θ_i is shifted by $\pm \frac{\pi}{2}$.

To estimate the gradients using the parameter-shift rule, we need to evaluate the expectation values $\langle O \rangle_{\theta_i \pm \frac{\pi}{2}}$ for each parameter θ_i. This can be done by running the quantum circuit twice for each parameter, with the corresponding parameter shifted by $\pm \frac{\pi}{2}$, and measuring the observable O. The gradients can then be computed using the finite-difference formula. The parameter-shift rule provides a way to estimate gradients on quantum circuits without the need for ancilla qubits or complex quantum operations.

Appendix F. Hyperparameters.

Unless otherwise specified, the default hyperparameters are:

- Number of Qubits = 8
- Circuit Depth = 3
- Cost Function: MMD
- Prior: Uniform
- Mass Matrix $\boldsymbol{M} = \boldsymbol{I}_{64}$
- Batch Size $S = 200$
- Degrees of Freedom $N_d = 1$
- Additive Noise $\sigma_A = 1.0$
- Thermal Mass $\mu = 1.0$
- Number of samples = 5000
- Burn-in Period = 1000
- Stepsize $h = 0.1$