

Conditional Born machine for Monte Carlo events generation

Oriel Kiss^{1,2,*}, Michele Grossi¹, Enrique Kajomovitz³ and Sofia Vallecorsa¹

¹European Organization for Nuclear Research (CERN), Geneva 1211, Switzerland

²Department of Computer Science, University of Geneva, Geneva 1211, Switzerland

³Department of Physics, Technion, Haifa 32000, Israel

(Dated: May 17, 2022)

Generative modeling is a promising task for near-term quantum devices, which can use the stochastic nature of quantum measurements as random source. So called Born machines are purely quantum models and promise to generate probability distributions in a quantum way, inaccessible to classical computers. This paper presents an application of Born machines to Monte Carlo simulations and extends their reach to multivariate and conditional distributions. Models are run on (noisy) simulators and IBM Quantum superconducting quantum hardware.

More specifically, Born machines are used to generate muonic force carriers (MFC) events resulting from scattering processes between muons and the detector material in high-energy-physics colliders experiments. MFCs are bosons appearing in beyond the standard model theoretical frameworks, which are candidates for dark matter. Empirical evidences suggest that Born machines can reproduce the underlying distribution of datasets coming from Monte Carlo simulations, and are competitive with classical machine learning-based generative models of similar complexity.

I. INTRODUCTION

Quantum computers have the potential to solve problems that are difficult for classical computers, such as factoring [1] or simulation of quantum systems [2]. However, the unavailability of error-correcting codes and limited qubit connectivity prevents them from being used. Nevertheless, noisy-intermediate-scale-quantum (NISQ) [3] devices, characterized by their low number of noisy qubits and short decoherence time, have already been successfully proven in domains such as machine learning [4–13] or quantum chemistry [14–16].

The present manuscript focuses on generative modeling in quantum machine learning (QML), which is the task of learning the underlying probability distribution $\pi(y)$ of a given dataset and generating samples from it. In the classical regime, generative models are often expressed as neural networks. For instance, generative adversarial networks (GAN) [17] and variational autoencoder [18] have been successfully applied in a variety of fields, ranging from computer vision [19] to natural sciences [20]. In High Energy Physics (HEP), generative models have been proposed as an alternative to Monte Carlo (MC) simulations, e.g., to simulate detectors [21–23] and very recently as a method to load distributions of elementary particle-physics processes [24]. MC calculations in HEP, such as Geant4 [25] or MadGraph [26] are usually expensive in time and CPU resources [27]. Generative models provide a solution, e.g., by augmenting small MC datasets or inter/extrapolating to different regimes.

The probabilistic nature of quantum mechanics allows us to define a new class of generative models: *quantum circuit Born machine* (QCBM). These models

use the stochastic nature of quantum measurement as random-like sources and have no classical analog. More specifically, they produce samples from the underlying distribution of a pure quantum state by measuring a parametrized quantum circuit [28] with probability given by the Born rule $p_\theta(x) = |\langle x|\psi(\theta)\rangle|^2$. Born machines have been proposed as Bayesian models [13], using an adversarial training strategy [11, 29], optimal transport [12] and adapted to continuous data [30, 31]. Quantum neural networks using a Gaussian noise source [32], or quantum Boltzmann machines [33] are both viable alternatives for quantum generative modeling but will not be addressed in the present manuscript. Quantum generative models also have the ability to load probability distribution on a quantum computer [11], which can then be used to integrate elementary processes via quantum amplitude estimation [24] or for finance applications [34].

Here, an extension to multivariate and conditional probability distributions is proposed, exploring limitations of NISQ devices. Even if generating multivariate distributions with Born machines has already been explored by Zhu *et al.* [35], we propose an alternative circuit design with a reduced connectivity, better suited for NISQ devices. Noteworthy, conditional distributions have never been explored in this context, and this work is thus a first step in that direction. Experiments were conducted on (noisy) simulators and superconducting devices from IBM Quantum using `qiskit runtime`, the recent serverless architecture framework which handles classical and quantum computations simultaneously on a dedicated cloud instance. Noisy simulators incorporate gates and readout errors, approaching real device performances. However, their behavior can be genuinely different from noisy simulators. Hence, this work emphasizes the use of real quantum hardware and address related challenges.

This paper is organised as follows. Section II introduces the physical use case of muonic force carriers and

* oriel.kiss@cern.ch

the preprocessing of the dataset. The models are introduced in Sec. III. More specifically, Sec. III A introduces the quantum circuit Born machine and Sec. III B and III C the multivariate and conditional versions, respectively. Results are shown in Sec. V for all models on (noisy) simulators and real quantum hardware.

II. MUONIC FORCE CARRIERS

A. Physical Setting

Muon force carriers (MFC) are theorized particles that could be constituents of dark matter and explain some anomalies in the measurement of the proton radius and the muon’s magnetic dipole, making them exciting candidates for new physics searches.

Following Galon *et al.* [36], we consider a muon fixed-target scattering experiment between muons produced at the high-energy collisions of the LHC and the detector material of the ForwARD Search ExpeRiment (FASER) or the ATLAS calorimeter. In the ATLAS case [36], independent muon measurements performed by the inner detector and muon system can help observe new force carriers coupled to muons, which are usually not detected. In the FASER experiment, the high resolution of the tungsten/emulsion detector is used to measure muons trajectories and energies.

B. Dataset

The dataset, produced with MadGraph simulations [26], is composed of samples with the following variables: the energy (E), transversal momentum (pt) and pseudo-rapidity (η) of the outgoing muon and MFC, conditioned on the energy of the incoming muon. The data is made more Gaussian-shaped by being preprocessed in the following way: the energy is divided by the mean of the incoming energy, the transverse momentum is elevated to the power of 0.1 [37], and everything is standardized to zero mean and unit variance. The dataset is composed of 10240 distinct events, and it is splitted into a training and testing set of equal size.

III. MODELS

A. Quantum Circuit Born Machine

A Born machine represents a probability distribution as a quantum pure state and can generate samples via projective measurements. The Born machine outputs binary strings, which can be interpreted as a sample from the generated discrete probability distribution. Similar to a classification task, the target distribution is discretized into 2^N bins, which are associated to the different binary strings of size N . The quantum state can take

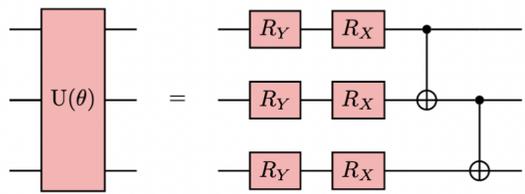


Figure 1: Ansatz constructed with $R_Y(\theta)$ and $R_X(\theta)$ single-qubit rotation and CNOT interactions with a linear connectivity.

the form of a quantum circuit [13] or a tensor network [38], acting on some initial state, e.g. $|0\rangle^{\otimes N}$. Following the former approach, this manuscript considers quantum circuit Born machines (QCBM), where the quantum circuit is constructed, for convenience, using repetition of basic layers $U(\theta)$. These building blocks are chosen to be the time evolution of an Ising Hamiltonian, which is conjectured to be difficult to simulate classically [6, 12]. On the other hand, $U(\theta)$ can be written in a hardware-friendly fashion for NISQ superconducting circuits devices, using the Baker-Campbell-Hausdorff formula. An example, constructed with $R_Y(\theta) = \exp(-i\theta\sigma_y/2)$ and $R_X(\theta) = \exp(-i\theta\sigma_x/2)$ single-qubit rotations and CNOT interaction between two qubits with linear connectivity is shown in Figure 1. A $R_Y(\theta)$ rotation is always added before the measurements.

B. Correlated Features

A simple way to extend the above-defined QCBM to generate correlated features is to use different registers for each of them, as proposed in Ref. [35]. In this scenario, a global unitary C entangles the registers while a local operator $U(\theta)$ learns the individual distributions, as shown in Figure 2. Zhu *et al.* [35] propose to use a GHZ state

$$|\text{GHZ}\rangle = \frac{|0\rangle^{\otimes 2n} + |1\rangle^{\otimes 2n}}{\sqrt{2}}, \quad (1)$$

which is a generalization of a Bell state on $N = 2n$ qubits. However, our investigations empirically suggest that the long-range interactions do not bring clear benefits and are expensive to be implemented on NISQ devices. Moreover, in this case using a GHZ state led to circuits exceeding the coherence time of the device, resulting in uniform distributions coming from maximally mixed states. Empirical evidence from this work suggests that a trainable C easily leads to the appearance of the well-known barren plateaus phenomena [39]. These reasons suggest the use for more hardware-friendly alternatives which are better suited for NISQ devices. Therefore, a fixed state is constructed only on the first qubits of each pair of reg-

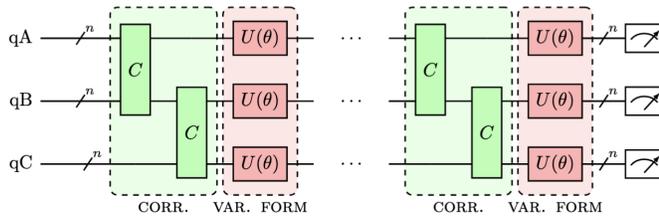


Figure 2: QCBM for a multivariate probability distribution. The fixed C gates create entanglement between the registers while the trainable $U(\theta)$ gates learn the distributions.

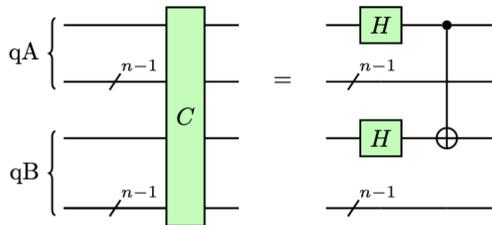


Figure 3: Correlation gates C between two quantum registers qA and qB, with n qubits each.

isters of size n . We choose

$$|C\rangle = \left[\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \otimes |0\rangle^{\otimes n-1} \right]^{\otimes 2}. \quad (2)$$

Only a linear connectivity is considered between the registers. The circuit to construct this state is shown in Figure 3.

It is possible, by using the $U(\theta)$ from Figure 1, to map the multivariate Born machine, from Figure 2, to an IBM Quantum chip without using any swap gates. This is due to the linear connectivity of all the components and the T topology of the device. Figure 4 shows a possible way to do so onto a 27 qubits architecture, using three registers (qA in red, qB in blue, and qC in green) with $n = 3$ qubits each. The elimination of swap gates diminishes the amounts of errors made on the quantum devices, by reducing the number of double-qubit gates and depth.

C. Conditional Born Machine

Conditional generative models, such as conditional generative adversarial networks (C-GAN) [40] produce samples x according to some conditions y . This task is more challenging since $p(x|y)$ has to be captured, instead of only $p(x)$. Conditional generative models may be able to reduce over-fitting by sharing weights across different value of the conditioning variable y . The flexibility of conditional generative models compared to MC simulations is advantageous in terms of computational and time resources needed to generate complex events. For

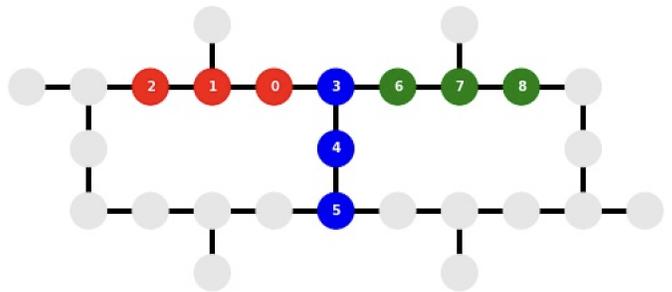


Figure 4: Mapping of the multidimensional QCBM onto a 27 qubits IBM quantum chip. The different colors refer to different registers and the ordering follows the convention of Qiskit.

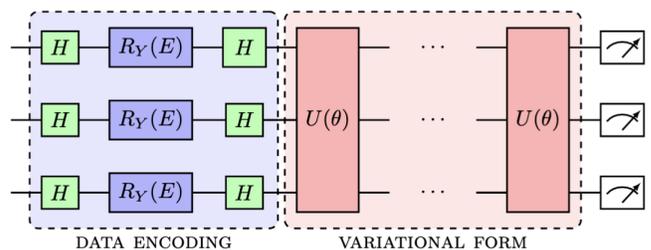


Figure 5: Conditional Born machine. The data dependent block (blue) acts as a feature map while the trainable gates (red) learn the distribution.

instance, in the MC simulations used in this work, the initial energy has to be fixed, while left as a variable in ML-based techniques. Hence, a conditional model could interpolate or extrapolate, reducing the time consumption needed for MC simulations.

The condition y in MFC events is the energy E_{in} of the incoming muon. Different experimental values for E_{in} are considered, which ranges from 50 GeV to 200 GeV in steps of 25 GeV. The conditional QCBM tries to generate the correct distributions when been given access to the incoming muon's energy. In practice, E_{in} is first scaled between $[0,1]$, transformed with the function arcsine, as used in [7], and then encoded into the QCBM via repeated $R_Y(E_{\text{in}})$ rotations on all qubits, as shown in Figure 5. Overall, the model consists of a feature map that encodes the data and trainable gates that learn the probability distribution. More complex feature maps [10], or data re-uploading [41] strategies were used, but did not show any improvement.

IV. TRAINING STRATEGY

A. Optimization

The QCBM is trained using a two-samples test, with a Gaussian kernel

$$K(x, y) = \exp\left(-\frac{(x - y)^2}{2\sigma}\right), \quad (3)$$

by comparing the distance between two samples x and y in the kernel feature space. Concretely, the maximum mean discrepancy (MMD) [42] loss function

$$\mathcal{L} = \mathbb{E}[K(x, y)]_{x \sim p_\theta, y \sim p_\theta} - 2\mathbb{E}[K(x, y)]_{x \sim p_\theta, y \sim \pi} + \mathbb{E}[K(x, y)]_{x \sim \pi, y \sim \pi} \quad (4)$$

is used, with bandwidth

$$\sigma \in [0.01, 0.1, 1, 10, 100]. \quad (5)$$

In this way, the difference of all the moments between the target and model probability distribution are efficiently compared at different scales. Advantages of the MMD include its metric properties and the training stability it provides, making it a suitable option in the NISQ era.

The gradient can be computed [13] using the shift rule [43], as

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \mathbb{E}[K(x, y)]_{x \sim p_{\theta^+}, y \sim p_{\theta^+}} - \mathbb{E}[K(x, y)]_{x \sim p_{\theta^-}, y \sim p_{\theta^-}} - \mathbb{E}[K(x, y)]_{x \sim p_{\theta^+}, y \sim \pi} + \mathbb{E}[K(x, y)]_{x \sim p_{\theta^-}, y \sim \pi}, \quad (6)$$

where p_{θ^\pm} are QCBMs with parameters $\theta^\pm = \theta \pm \pi/2 \cdot \hat{e}_i$ with \hat{e}_i the i^{th} unit basis vector in parameters space, i.e. $(\hat{e}_i)_j = \delta_{ij}$.

Alternatively, the Simultaneous Perturbation Stochastic Approximation (SPSA) [44] algorithm is also considered to optimize the QCBM in a gradient-free fashion. SPSA efficiently approximates the gradient with two sampling steps by perturbing the parameters in all directions simultaneously. While the convergence is slower than using the exact gradient, fewer circuit evaluations are needed for each epoch. Moreover, the stochastic nature of SPSA makes it more resilient to hardware and statistical noise. We found on simulations that the gradient-based algorithm outperforms SPSA, as SPSA sometimes gets trapped in local minima. However the gradient-based algorithm is more resource intensive than SPSA, and in this regime SPSA is preferred as it is better suited for quantum hardware.

Therefore, a mixed training scheme is used, where the models are first trained on (noisy) simulators using the ADAM optimizer [45], and then fine-tuned for a few epochs on quantum hardware using SPSA. A readout error mitigation scheme [46] is used on the measurements. Details about the implementation, training and resources can be found in Appendix A.

B. Classical Baseline

Classical generative models trained using the MMD loss function (GMMD models) [47, 48] are used as a baseline. They are trained on continuous data since the performance is usually higher than for discrete samples. The impact of the classical models' complexity is investigated by comparing QCBMs with a classical model of approximately the same size and GMMD models which have two orders of magnitude more parameters than the aforementioned QCBMs. Since currently available quantum hardware are unable to accommodate large scale versions of the proposed algorithms, we are forced to perform proof of concept experiments. The goal of the classical baseline, at this stage, is to give an indication of the current level of deployability of quantum machine learning models and not to predict quantum advantage.

V. RESULTS

A. One Dimensional Distribution

As a first demonstration, the QCBM is trained on a one-dimensional distribution: the energy of the outgoing muon discretized on $2^4 = 16$ bins. The QCBM is built with one repetition of $\text{RY}(\theta)$, $\text{RX}(\theta)$, and $\text{RZZ}(\theta) = \exp(-i\theta\sigma_z \otimes \sigma_z)$ gates using a full entanglement scheme. Empirical evidence suggests that this circuit is better suited to this task than the one proposed in Figure 1. The small number of two qubits gates enables the use of real quantum hardware without severe complications due to the noise. Results obtained with an ideal simulator, noisy simulator, superconducting circuits (ibmq-montreal) and classical GMMD are shown in Figure 6. The histograms display the number of generated events and the ratios with the dataset as a function of energy (GeV), with error bars corresponding to one standard deviation from ten sampling processes. The GMMD is chosen to be a neural network with 4 hidden layers of size [64, 128, 64, 16], each with a sigmoid activation function, and a latent space of dimension 15.

The total variance (TV) $\in [0, 1]$ with sample set Ω

$$\text{TV}(p, \pi) = \frac{1}{2} \sum_{x \in \Omega} |p(x) - \pi(x)| \quad (7)$$

is used as a comparison metric and results are shown in Table I.

The QCBM is competitive with the GMMD of higher complexity and superior in the under-parameterized regime, as outlined by the results of the *easy* GMMD. The noise does not negatively contribute to the performance, as emphasized by the noisy simulations and quantum hardware results.

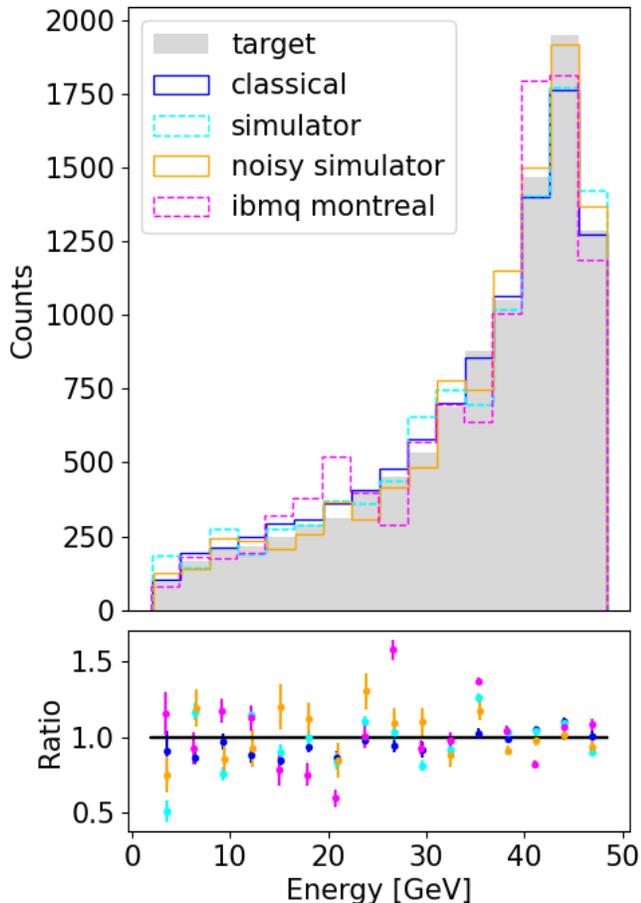


Figure 6: Outgoing muon’s energy at 50 GeV. Comparison of QCBMs run on perfect simulator (dashed cyan), noisy simulator (orange), ibmq_montreal (dashed pink) with the classical GMMD (blue).

backend	TV
simulator	0.055
noisy simulator	0.043
ibmq_montreal	0.074
GMMD	0.028
easy GMMD	0.290

Table I: Total variance for the one dimensional distribution on different backends.

B. Multivariate Distribution

As a next step, we consider a multivariate distribution, namely the energy, transverse momentum, and pseudorapidity of the outgoing muon with incoming energy of 125 GeV, using $2^3 = 8$ bins. The QCBM is designed with five repetitions of entangling C and local $U(\theta)$ layers, as seen in Figure 2. The former creates an entangling state, while the latter consists of $\text{RY}(\theta)$ and CNOT interaction with linear connectivity. The results for the simulator,

backend	TV(E)	TV(pt)	TV(η)
simulator	0.055	0.05	0.052
noisy simulator	0.075	0.12	0.06
ibmq_mumbai	0.078	0.097	0.13
GMMD	0.036	0.017	0.063
easy GMMD	0.360	0.040	0.110

Table II: Total variance for the individual multivariate distributions on different backends.

noisy simulator, ibmq_mumbai, and GMMD are shown in Figure 7 and the total variance for the individual distributions are presented in Table II. The GMMD is constructed similarly as above but with three hidden layers of size [128, 256, 128]. Even if the GMMD achieves the best accuracy, the QCBM is still competitive despite its small number of learned parameters or the presence of noise.

An important factor for the performance of generative models is their ability to learn the correlations between the variables, which is not reflected in the total variance. To this point, the correlations in the target datasets are compared to those in the generated datasets. Figure 8 shows the difference between the correlation matrices calculated for the classical GMMD, (noisy) quantum simulators, and the quantum backend ibmq_mumbai. The correlation matrices are computed with the Pearson product-moment methods

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} \cdot C_{jj}}}, \quad (8)$$

where C is the covariance matrix.

We observe that the QCBMs trained on the different backends can reproduce the correlations, even if the classical GMMD is better.

C. Conditional Distribution

Finally, the conditioning on the initial muon’s energy is encoded into the QCBM via parametrized-rotations. The QCBM as outlined in Figure 5 contains four repetitions of $U(\theta)$ while the GMMD has 2 hidden layers of size [8, 8]. The training is performed on the whole dataset except at 125 GeV, which is left to test the interpolation capabilities of the models. Results are shown in Figure 9, and the values of the total variance are reported in Table III. All models achieve good performance for the interpolation. The results on the quantum hardware could be slightly improved for some histogram binned values. However, the performance is similar on training and testing energy bins, suggesting that the QCBM is able to interpolate yet suffers from the noise.

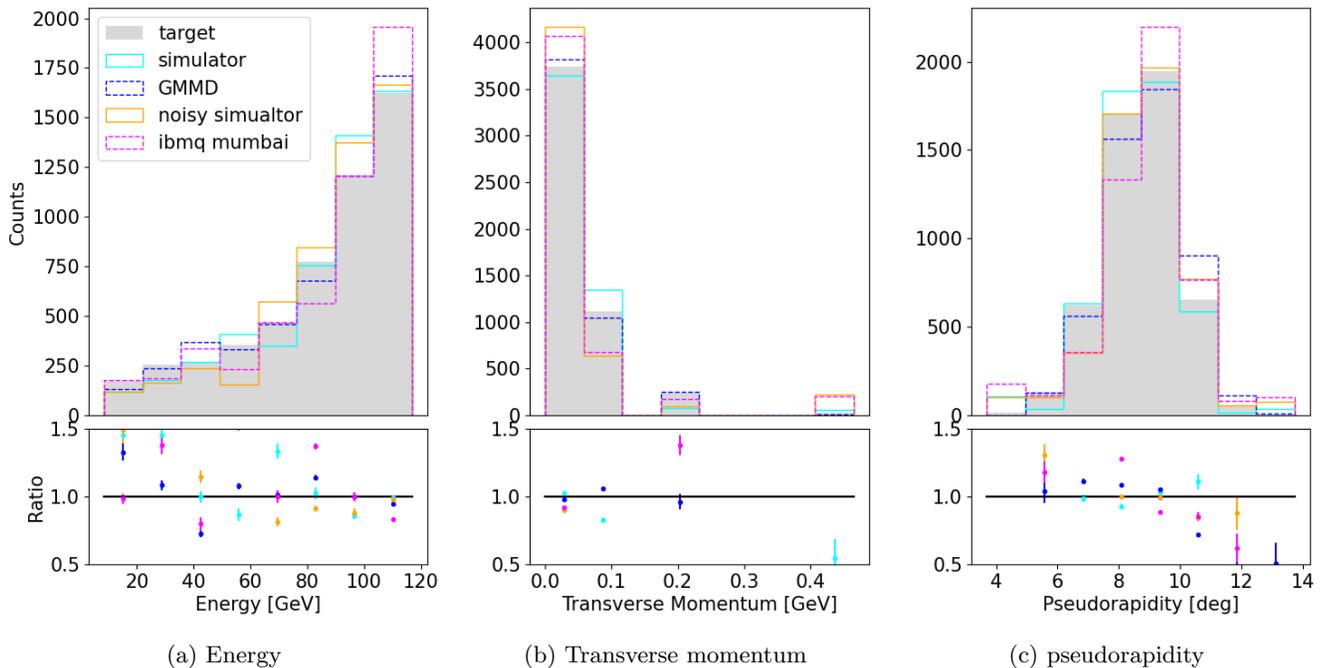


Figure 7: Outgoing muon’s energy (left), transverse momentum (middle) and pseudorapidity (right) at 125 GeV, generated by the multidimensional QCBM on a perfect simulator (cyan), noisy simulator (orange) and `ibmq_mumbai` (dashed pink) compared to a classical GMMD (dashed blue).

backend	TV(100)	TV(150)	TV(125)
simulator	0.033	0.016	0.033
noisy simulator	0.067	0.046	0.035
<code>ibmq_mumbai</code>	0.15	0.13	0.094
GMMD	0.016	0.032	0.034

Table III: Total variance for the conditional distribution on different backends.

VI. DISCUSSION

The results presented in the above section suggest that QCBMs are competitive with GMMD models with a larger number of parameters. For instance, the ratio between the number of parameters of the two models is approximately 1:600 in the one dimensional case, and 1:1000 in the multivariate case. Only in the conditional case is the complexity comparable. These results indicate that GMMD models need more parameters than QCBMs for similar performance. Indeed, reducing the complexity to the level of the QCBM was subject to heavy underfitting, as outlined by the performance of the *easy* GMMDs. This suggests that QCBMs are more expressive than classical models, as outlined by Abbas *et al.* [49], and outperform them in the under-parametrized regime, pointing towards a quantum advantage in terms of model’s complexity and performance for large scale QCBMs.

Moreover, the presence of noise does not seem to be an obstacle in training of QCBMs. It is noteworthy that the

results obtained on quantum hardware are close to the one obtained on the simulator, which suggests a sufficient device quality for this task and an ability to deal with incoherent noise. The hardware results are slightly worse in the conditional case which can be explained by the reduced number of epochs performed on the quantum hardware. In fact, the loop over the training energy bins increases the resources needed for one epoch, and thus reduces the number of epochs performed.

These observations suggest that the noise is assimilated during the training, underlining the importance of using actual quantum hardware. This supports the findings of Borrás *et al.* [50], which empirically found that quantum generative adversarial networks can be efficiently trained on quantum hardware if the readout noise is smaller than 0.1. Thus, QCBMs seem to be an appealing application for NISQ devices.

Barren plateaus (BP) are large portions of the training landscape where the loss function’s gradient variance vanishes. As shown in [39], BP appear exponentially fast in the depth and number of qubits for generic quantum circuits, which makes the training of large-scale quantum variational algorithms generally difficult. Solutions to this issue, such as quantum convolutional neural networks [9] or local loss functions [51] are not applicable in this case since the measurements on all qubits are needed.

The circuits used in this work are shallow enough to avoid this effect. Nevertheless, difficulties are observed during the training on hardware and noisy simulators, which could be an effect of noise-induced BP [52]. A

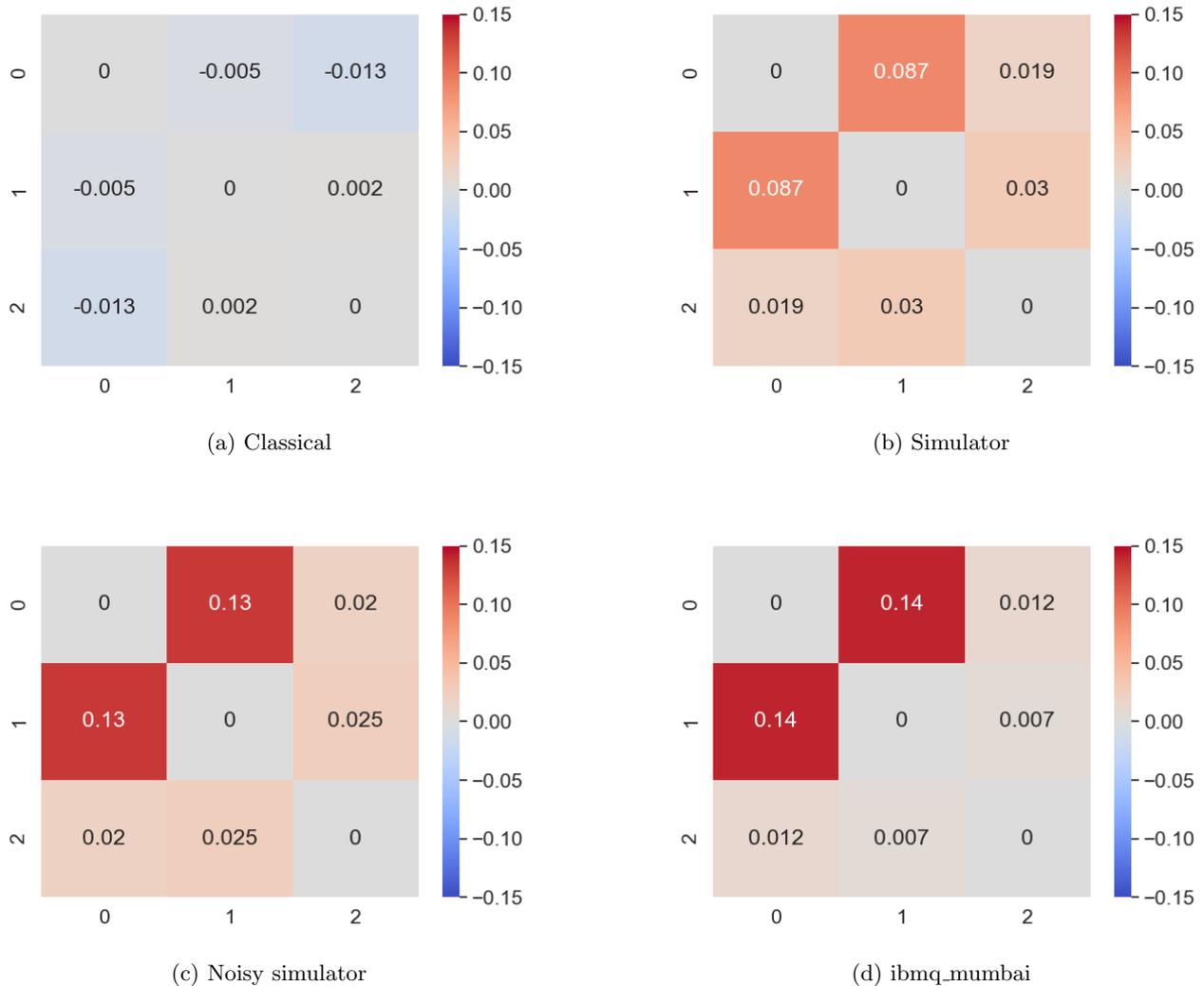


Figure 8: Difference of the correlation matrix from the dataset and the samples generated by the classical GMMD model (top left), perfect simulator (top right), noisy simulator (bottom left) and ibmq_mumbai (bottom right). The labels (0,1,2) refer to the energy (E), transverse momentum (p_t), and pseudorapidity (η), respectively.

more significant number of epochs were needed to mitigate this effect. Increasing the number of qubits to five or six, for the one-dimensional case, deteriorates the ratio between the generated and target samples for each bins, even if the loss function converged after a hundred of epochs. The same problem appeared when increasing the number of features in the multivariate case or mixing multiple features with the conditioning. Since the gradient never vanished at the beginning of the training, BP are probably not be the most critical issue, on the other hand the MMD may not be the most suitable loss function for large-scale QCBMs.

Alternative training strategies have been proposed in [11, 12], with optimal transport and an adversarial training strategy, respectively. Hence, empirical evidences suggest that the strong theoretical properties of the

MMD loss function are not met in practice, as outlined by some benchmarks [53]. Hence the performance of GMMD and GAN are similar for simple problems but the latter is superior for complex tasks. Li *et al.* [54] propose an adversarial strategy to optimize the kernel as an efficient way to improve the performance of GMMD models.

VII. CONCLUSION

The present manuscript presents the application and further development of a quantum circuit Born machine to generate Monte Carlo events in HEP, specifically muon force carriers. An efficient way to generate multivariate distributions, requiring only linear connectivity and thus suitable for NISQ devices, is proposed. Addition-

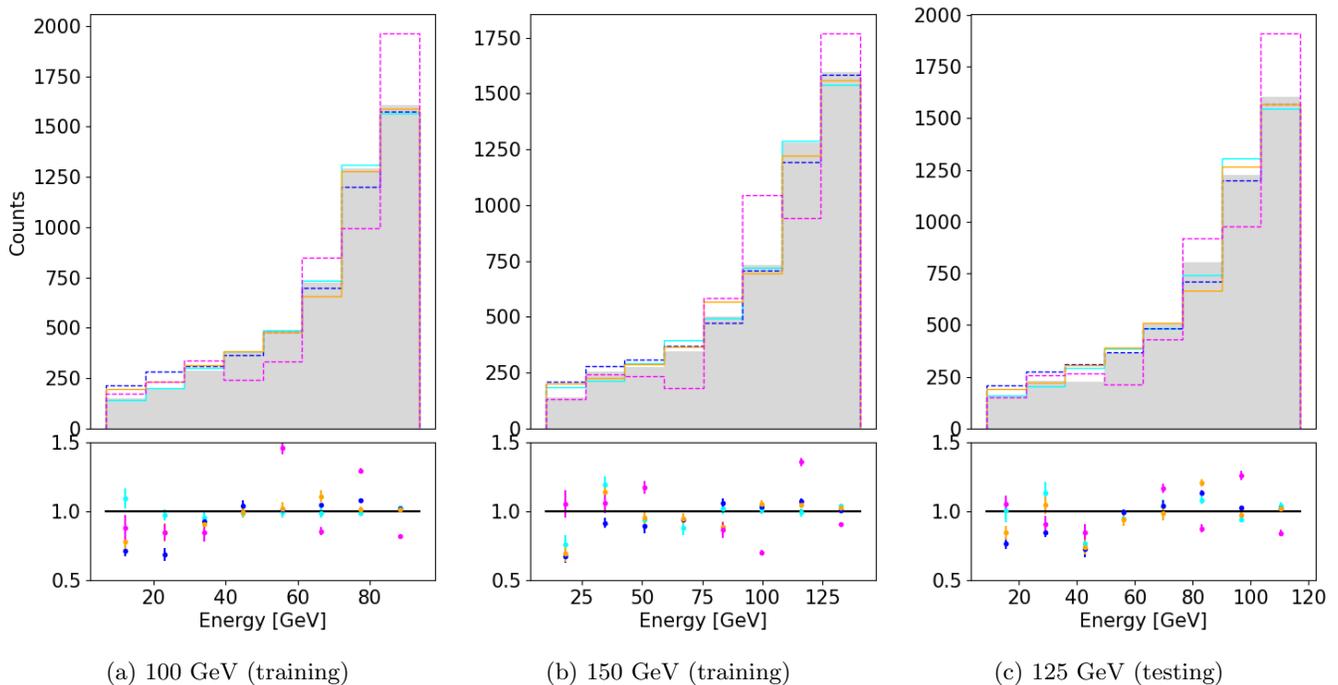


Figure 9: Outgoing muon’s energy with an initial energy of 100 GeV (left), 150 GeV (middle), and 125 GeV (right) generated by the QCBM on a perfect simulator (cyan), noisy simulator (orange) and IBMQ Mumbai (dashed pink) compared to a classical GMM (dashed blue). The models are tested on samples with an energy of 125 GeV while they are trained on samples with the remaining energies.

ally, the present paper is a first step towards generating conditional probability distributions with quantum circuit Born machines. Numerical evidence demonstrates that QCBMs can efficiently generate joint and conditional distributions and that they are competitive against classical neural networks of similar complexity. Moreover, the experiments are run successfully on quantum hardware, finding that QML algorithms can mitigate the effect of the noise during the training. Quantum generative models are consequently appealing for NISQ devices since they can manage noisy qubits without the need of expensive error mitigation techniques. QCBMs also have the advantage of needing two or three orders of magnitude less parameters while still being competitive. This emphasizes the importance of developing large scale QCBMs for potential quantum advantage.

While having strong potential in generative modeling, QCBMs still need improvement to handle a more refined binning and multivariate distribution of higher dimen-

sions. Additionally, it would be interesting to consider conditional multivariate distribution, which will be the focus of future work.

ACKNOWLEDGEMENT

The authors thank Su Yeon Chang, Dominic Pasquali, Tigran Ramazyan and Florian Rehm for valuable and stimulating discussions about the present manuscript.

This work was supported by CERN Quantum Technology Initiative. Simulations were performed on the University of Geneva’s Yggdrasil HPC cluster. Access to the IBM Quantum Services was obtained through the IBM Quantum Hub at CERN. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team.

-
- [1] Peter W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing* **26**, 1484–1509 (1997).
 [2] A.Yu.Kitaev, “Quantum measurements and the abelian stabilizer problem,” ArXiv e-prints (1995),

- [arXiv:9511026 \[quant-ph\]](https://arxiv.org/abs/9511026).
 [3] John Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum* **2**, 79 (2018).
 [4] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, “Quantum machine learning,” *Nature* **549**, 195–202 (2017).

- [5] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik, “The theory of variational hybrid quantum-classical algorithms,” *New Journal of Physics* **18**, 023023 (2016).
- [6] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” *Phys. Rev. A* **98**, 032309 (2018).
- [7] Oriol Kiss, Francesco Tacchino, Sofia Vallecorsa, and Ivano Tavernelli, “Quantum neural networks force fields generation,” ArXiv e-prints (2022), [arXiv:2203.04666 \[quant-ph\]](https://arxiv.org/abs/2203.04666).
- [8] Maria Schuld and Nathan Killoran, “Quantum machine learning in feature hilbert spaces,” *Phys. Rev. Lett.* **122**, 040504 (2019).
- [9] Iris Cong, Soonwon Choi, and Mikhail D. Lukin, “Quantum convolutional neural networks,” *Nat. Phys.* **15**, 1273–178 (2019).
- [10] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature* **567**, 209–212 (2019).
- [11] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner, “Quantum generative adversarial networks for learning and loading random distributions,” *npj Quantum Inf* **5** (2019), <https://doi.org/10.1038/s41534-019-0223-2>.
- [12] Daniel Mills Brian Coyle, Vincent Danos, and Elham Kashefi, “The born supremacy: quantum advantage and training of an ising born machine,” *npj Quantum Inf* **6**, 60 (2020).
- [13] Jin-Guo Liu and Lei Wang, “Differentiable learning of quantum circuit born machines,” *Phys. Rev. A* **98**, 062324 (2018).
- [14] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature* **549**, 242–246 (2017).
- [15] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik, “Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz,” *Quantum Sci. Technol.* **4**, 014008 (2019).
- [16] Luca Crippa, Francesco Tacchino, Mario Chizzini, Antonello Aita, Michele Grossi, Alessandro Chiesa, Paolo Santini, Ivano Tavernelli, and Stefano Carretta, “Simulating static and dynamic properties of magnetic molecules with prototype quantum computers,” *Magnetochemistry* **7** (2021), [10.3390/magnetochemistry7080117](https://doi.org/10.3390/magnetochemistry7080117).
- [17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing X, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” *NIPS’14: Proceedings of the 27th International Conference on Neural Information Processing Systems* **2**, 2672–2680 (2014).
- [18] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” *ICLR 2014 conference submission* (2014).
- [19] Zhengwei Wang, Qi She, and Tomás E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy,” *ACM Comput. Surv.* **54** (2021), [10.1145/3439723](https://doi.org/10.1145/3439723).
- [20] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová, “Machine learning and the physical sciences,” *Rev. Mod. Phys.* **91**, 045002 (2019).
- [21] Sofia Vallecorsa, “Generative models for fast simulation,” *Journal of Physics: Conference Series* **1085** (2018).
- [22] Michela Paganini, Luke de Oliveira, and Benjamin Nachman, “Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks,” *Physical Review D* **97** (2018), [10.1103/physrevd.97.014021](https://doi.org/10.1103/physrevd.97.014021).
- [23] A Maevskiy, D Derkach, N Kazeev, A Ustyuzhanin, M Artemev, and L Anderlini, “Fast data-driven simulation of cherenkov detectors using generative adversarial networks,” *Journal of Physics: Conference Series* **1525**, 012097 (2020).
- [24] Gabriele Agliardi, Michele Grossi, Mathieu Pellen, and Enrico Prati, “Quantum integration of elementary particle processes,” ArXiv e-prints (2022), [arXiv:2201.01547 \[hep-ph\]](https://arxiv.org/abs/2201.01547).
- [25] S. Agostinelli *et al.* (GEANT4), “GEANT4—a simulation toolkit,” *Nucl. Instrum. Meth. A* **506**, 250–303 (2003).
- [26] T. Stelzer and W. F. Long, “Automatic generation of tree level helicity amplitudes,” *Comput. Phys. Commun* **84**, 357–371 (1994).
- [27] A. M. Sirunyan, A. Tumasyan, W. Adam, F. Ambrogio, E. Asilar, T. Bergauer, J. Brandstetter, E. Brondolin, M. Dragicevic, and *et al.*, “Measurement of the cross section for top quark pair production in association with a w or z boson in proton-proton collisions at $\sqrt{s} = 13$ tev,” *Journal of High Energy Physics* **2018** (2018), [10.1007/jhep08\(2018\)011](https://doi.org/10.1007/jhep08(2018)011).
- [28] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini, “Parameterized quantum circuits as machine learning models,” *Quantum Science and Technology* **4** (2019), [10.1088/2058-9565/ab4eb5](https://doi.org/10.1088/2058-9565/ab4eb5).
- [29] Su Yeon Chang, Steven Herbert, Sofia Vallecorsa, Elías F. Combarro, and Ross Duncan, “Dual-parameterized quantum circuit gan model in high energy physics,” *EPJ Web Conf.* **251** (2021), <https://doi.org/10.1051/epjconf/202125103050>.
- [30] Ieva Čepaitė, Brian Coyle, and Elham Kashefi, “A continuous variable born machine,” *Quantum Mach. Intell.* **4** (2022), <https://doi.org/10.1007/s42484-022-00063-3>.
- [31] Jonathan Romero and Alan Aspuru-Guzik, “Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions,” ArXiv e-prints (2019), [arXiv:1901.00848 \[quant-ph\]](https://arxiv.org/abs/1901.00848).
- [32] Carlos Bravo-Prieto, Julien Baglio, Marco Cè, Anthony Francis, Dorota M. Grabowska, and Stefano Carrazza, “Style-based quantum generative adversarial networks for monte carlo events,” ArXiv e-prints (2021), [arXiv:2110.06933 \[quant-ph\]](https://arxiv.org/abs/2110.06933).
- [33] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner, “Variational quantum boltzmann machines,” *Quantum Machine Intelligence* **3** (2021), <https://doi.org/10.1007/s42484-020-00033-7>.
- [34] Javier Alcazar, Vicente Leyton-Ortega, and Alejandro Perdomo-Ortiz, “Classical versus quantum models in machine learning: insights from a finance application,” *Machine Learning: Science and Technology* **1**, 035003 (2020).
- [35] Elton Yechao Zhu, Sonika Johri, Mert Esencan Dave Bacon, Jungsang Kim, Mark Muir, Nikhil Murgai, Jason Nguyen, Neal Pistenti, Adam Schouela, Ksenia Sos-

- nova, and Ken Wright, “Generative quantum learning of joint probability distribution functions,” ArXiv e-prints (2021), [arXiv:2109.06315 \[quant-ph\]](https://arxiv.org/abs/2109.06315).
- [36] Iftah Galon, Enrique Kajamovitz, David Shih, Yotam Soreq, and Shlomit Tarem, “Searching for muonic forces with the atlas detector,” *Phys. Rev. D* **101**, 011701 (2020).
- [37] Yeo In-Kwon and Richard A. Johnson, “A new family of power transformations to improve normality or symmetry,” *Biometrika* **87**, 954–59 (2000).
- [38] Michael L. Wall, Matthew R. Abernathy, and Gregory Quiroz, “Generative machine learning with tensor networks: Benchmarks on near-term quantum computers,” *Phys. Rev. Research* **3**, 023010 (2021).
- [39] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven, “Barren plateaus in quantum neural network training landscapes,” *Nature Communications* **9** (2018), <https://doi.org/10.1038/s41467-018-07090-4>.
- [40] Mehdi Mirza and Simon Osindero, “Conditional generative adversarial nets,” ArXiv e-prints (2014), [arXiv:1411.1784 \[cs.AI\]](https://arxiv.org/abs/1411.1784).
- [41] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A* **103**, 032430 (2021).
- [42] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola, “A kernel two-sample test,” *Journal of Machine Learning Research* , 723–773 (2012).
- [43] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran, “Evaluating analytic gradients on quantum hardware,” *Phys. Rev. A* **99**, 032331 (2019).
- [44] J.C. Spall, “Implementation of the simultaneous perturbation algorithm for stochastic optimization,” *IEEE Transactions on Aerospace and Electronic Systems* **34**, 817–823 (1998).
- [45] D. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (2015).
- [46] Paul D. Nation, Hwajung Kang, Neereja Sundaresan, and Jay M. Gambetta, “Scalable mitigation of measurement errors on quantum computers,” *PRX Quantum* **2**, 040326 (2021).
- [47] Yujia Li, Kevin Swersky, and Richard Zemel, “Generative moment matching networks,” *Proceedings of the 32nd International Conference on Machine Learning* **37** (2015).
- [48] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani, “Training generative neural networks via maximum mean discrepancy optimization,” *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI 15)* , 258–267 (2015).
- [49] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner, “The power of quantum neural networks,” *Nature Computational Science* **1**, 403–409 (2021).
- [50] Kerstin Borrás, Su Yeon Chang, Lena Funcke, Michele Grossi, Tobias Hartung, Karl Jansen, Dirk Kruecker, Stefan Kühn, Florian Rehm, Cenk Tüysüz, and Sofia Vallecorsa, “Impact of quantum noise on the training of quantum generative adversarial networks,” (2022).
- [51] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles, “Cost function dependent barren plateaus in shallow parametrized quantum circuits,” *Nat. Commun.* **12** (2021), <https://doi.org/10.1038/s41467-021-21728-w>.
- [52] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles, “Noise-induced barren plateaus in variational quantum algorithms,” *Nat Commun* **12**, 6961 (2021).
- [53] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, “Deep learning face attributes in the wild,” *Proceedings of the Thirty-two Conference on Computer Vision and Pattern Recognition (CVPR 2015)* (2015).
- [54] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos, “Mmd gan: Towards deeper understanding of moment matching network,” *Proceedings of Thirty-first Annual Conference on Neural Information Processing Systems (NIPS 2017)* (2017).
- [55] Ville Bergholm *et al.*, “PennyLane: Automatic differentiation of hybrid quantum-classical computations,” ArXiv e-prints (2018), [arXiv:1811.04968 \[quant-ph\]](https://arxiv.org/abs/1811.04968).
- [56] James Bradbury *et al.*, “JAX: composable transformations of Python+NumPy programs,” (2018).
- [57] MD SAJID ANIS *et al.*, “Qiskit: An open-source framework for quantum computing,” (2021).

Appendix A: Implementation

The noiseless simulations are performed with PennyLane [55] powered by JAX [56], which enables an efficient gradient computation via vectorization and just-in-time compilation. The noisy simulations are performed using a fake backend tuned into the real quantum hardware, provided by Qiskit [57]. For instance, we used the noisy simulator tuned to ibmq_mumbai with a CNOT error rate of 0.085 and a readout error of 0.029 on average. The training is performed in batches composed of 512 events each, and one epoch is composed of ten batches. The learning rate is initially set to 0.01 and is halved every 20 epochs.

The resources needed to produce the presented results are presented in Table A, which show the number of parameters, the time needed for a forward and backward pass, and the number of epochs until convergence for all the quantum models trained on the simulator. Each epoch is composed of 10 batches, except the conditional model which has ten batches per training energy bin (i.e. six). Each batch contained 512 samples. Simulations were run on a single CPU on the University of Geneva’s Yggdrasil HPC cluster.

Model	Param.	Forward [s]	Backward [s]	Epochs
one dimensional	18	1.2 [s]	3.9 [s]	70
multivariate	45	1.9 [s]	9.4 [s]	100
conditional	27	1.5 [s]	4.5 [s]	30

Table IV: Number of parameters, time needed for a forward and backward pass and number of epochs until convergence for the three quantum models trained on a simulator.