Quantum-Assisted Generative AI for Simulation of the Calorimeter Response

Wojciech T. Fedorko[1,*], *J. Quetzalcoatl* Toledo-Marín[1,2], *Geoffrey C.* Fox[3], *Colin W.* Gay[4], *Hao* Jia[4], *Ian* Lu[1,5], *Roger* Melko[2], *Eric* Paquet[6], *Deniz* Sogutlu[1,4], and *Maximilian J.* Swiatlowski[1]

[1]TRIUMF, Vancouver, BC V6T 2A3, Canada
[2]Perimeter Institute for Theoretical Physics, Waterloo, ON, N2L 2Y5, Canada
[3]University of Virginia, Charlottesville, VA, 22911, USA
[4]University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[5]University of Toronto, Toronto, ON, M5S 1A1, Canada
[6]National Research Council,Ottawa, ON, K1A 0R6, Canada

Abstract. As CERN approaches the launch of the High Luminosity Large Hadron Collider (HL-LHC) by the decade's end, the computational demands of traditional simulations have become untenably high. Projections show millions of CPU-years required to create simulated datasets - with a substantial fraction of CPU time devoted to calorimetric simulations. This presents unique opportunities for breakthroughs in computational physics. We show how Quantum-assisted Generative AI can be used for the purpose of creating synthetic, realistically scaled calorimetry dataset. The model is constructed by combining D-Wave's Quantum Annealer processor with a Deep Learning architecture, increasing the timing performance with respect to first principles simulations and Deep Learning models alone, while maintaining current state-of-the-art data quality

1 Introduction

A key objective of the High-Luminosity LHC (HL-LHC) is precision Higgs boson studies. The vast dataset will allow Higgs couplings to fermions and gauge bosons to be measured with unprecedented precision. Rare decay modes such as $H \rightarrow \mu^+\mu^-$ will be accessible, and double Higgs production could be probed to extract the Higgs self-coupling.

The HL-LHC will also significantly enhance sensitivity to physics beyond the Standard Model and enable new precision tests of the Standard Model.

This tremendous opportunity brings also a huge technological and experimental challenge. Among other aspects the computational load is expected to increase to millions of CPU-core years annually [1]. This demand is in large portion driven by the need to create vast simulated datasets, needed to conduct statistical analysis of the experimental data. One of the major simulation tasks is simulation of calorimeter with first-principles simulation - GEANT4 [2]. The Machine Learning LHC community recognized this challenge and is applying a variety of Deep Generative Models to the problem in the hope of creating a surrogate

[*] e-mail: wfedorko@triumf.ca

model that would produce high quality synthetic data rapidly. Various approaches have been tried ranging from Generative Adversarial Networks (GAN) [3], Diffusion Models [4], Normalizing Flow-based models [5] and others. For a recent overview we invite reader to consult a recent benchmarking paper [6]. Notably the ATLAS experiment has deployed a GAN-based model in their fast simulation framework already [7].

In general the Deep Generative Models work by sampling from a relatively simple fixed distribution - or 'latent space' - for example a multi-dimensional Gaussian distribution, and then transforming that random number in a long sequence of steps to produce a sample representative of the target distribution - in our case a section of a calorimeter where a shower is deposited.

Our group has taken an alternative approach where we aim to generate a random number from a complex, expressive distribution that can be *trained* to represent the target distribution optimally. That random number is then processed by a relatively uncomplicated neural network that outputs data in the desired format quickly. Classically generating random numbers from arbitrary distributions is computationally intensive. However we aim to use a quantum annealer to accelerate this process - that is to employ the quantum processor (QPU) for the task which it is well suited for - sampling of random numbers from arbitrary, learnable distributions. We thus arrive at a quantum-assisted generative model. Previous efforts by this group include the application of a discrete variational encoder to this problem [8], followed by the development of an initial quantum-assisted model on a simplified dataset [9], culminating in a conditioned quantum assisted model with advancements related to technical aspects of quantum machine learning [10]. In this work we report on results with updated quantum annealing architecture with advancements on the corrected backpropagation of the gradient with respect to the latent space parameters.

In the subsequent sections we describe the dataset used, the philosophy behind the design of the model, followed by preliminary qualitative and quantitative analysis of the generated data and comparisons of selected deep generative models to our model, in terms of quality of the data generated and energy consumptions before concluding.

2 Dataset

For this project we use one of the *CaloChallenge 2022* [11] datasets namely the *Dataset 2* containing a sample of electrons simulated with GEANT4 simulation. Electrons, impinging perpendicularly on the calorimeter have a log-normal distributed energy spectrum between 1 GeV and 1 TeV. A shower development in simulated in a cylindrical volume of the detector voxelized in a cylindrical geometry with the axis centered on the direction of the impinging particle. For the purpose of the discussion here we assign the z axis in the direction of the incident electron and the x and y in the plane transverse the direction of the electron using the right-handed convention. The detector is a 45-layer tungsten-silicon sandwich. The cylinder is divided in 9 voxels radially - with each voxel radial dimension corresponding approximately to 0.5 Molière radius. The cylindrical volume is divided 16-fold in the azimuthal direction.

The dataset is divided into training, validation and testing subsets with a 80%, 10% 10% split.

Before feeding the shower and incident energy data to the model, we apply on-the-fly transformations. Given a shower $\boldsymbol{v}$ and incident energy e, we first normalize energies in each voxel i as $E_i = v_i/e$, ensuring $E_i \in [0, 1]$ range. To avoid strict bounds, we define $u_i = \delta + (1 - 2\delta)E_i$ with $\delta = 10^{-7}$ and apply the *logit* transformation:

$$x_i = \ln \frac{u_i}{1 - u_i} - \ln \frac{\delta}{1 - \delta} \tag{1}$$

which preserves zero values. The incident energy, used for conditioning, is log-transformed and scaled to $[0, 1]$. Unlike previous approaches [4, 12, 13], we omit standardization to maintain zero values in x_i.

3 Model Design Philosophy and architecture

In this work we aim to give the reader an intuitive understanding of the philosophy of the model and the architecture. The reader is referred to our prior work [10] (and references within) for the details of the model architecture and the mathematical foundations. In addition we outline the corrected strategy for calculation of the gradient of the loss with respect to the parameters of the latent space - a component of our model updated with respect to prior work.

Our model, the **Calo4pQVAE** uses an architecture similar to a variational auto-encoder (VAE). The first stage of processing uses an encoder incorporating a three dimensional convolution structure. The encoder is also referred to as the *approximating posterior* $q_\phi(\boldsymbol{z}|\boldsymbol{x}, e)$, where $\boldsymbol{z}$ is the latent variable, $\boldsymbol{x}$ is the input data - namely transformed voxel energies, and e is the encoding of the incident particle energy. The encoder is implemented as a neural network with parameters ϕ. The encoder compresses the data into latent representation appropriate for processing by the model comprising the latent space, also known as the prior $p_\theta(\boldsymbol{z})$. The feature differentiating our model from the traditional variational encoder is that the latent space is modeled by a Restricted Boltzmann Machine (RBM), a generative model capable of synthesizing arbitrary binary distributions. The decoder, represented by $p_\theta(\boldsymbol{x}|\boldsymbol{z}, e)$, also incorporates a three dimensional convolution structure with parameters θ, processes the data $\boldsymbol{z}$ from the latent space and generates synthetic data $\boldsymbol{x}$ in the format identical to the input data. During training, the latent space model learns the probability distribution of the data in the latent representation provided by the encoder, and the decoder learns to re-generate the input data. The loss used to train the model encompasses the fidelity of the data reconstruction and a regularizing term encapsulating the ability of the latent space model to re-generate the distribution provided by the encoder. During synthetic data generation the encoder is not used. Instead new samples are generated from the prior and processed by the decoder. The RBM can be sampled from using a D-Wave QPU - and therefore it must match the structure of the target QPU in terms of nodes (mapped to qubits on a QPU) and inter-node connections (mapped to couplers on a QPU). Sampling is possible using classical methods - however these are very computationally intensive. While mapped to QPU we demonstrate that samples of same quality can be obtained rapidly. This is, in fact, the defining feature of the model - an expressive latent space that is learnable and can innately model a complex probability distribution. Quantum sampling allows the model to keep the expressiveness of the prior, while accelerating the sampling.

The encoder has a hierarchical structure where each level of the hierarchy produces one partition of the latent space data. That partition of the latent space is then concatenated with the input data and passed on to the subsequent level of the hierarchy as shown in Fig. 1. This hierarchical structure is meant to reflect conditional dependencies in the input data. The convolution layers in the encoder incorporate a padding strategy. Cylindrical structure of the data is first unrolled into a cuboid structure and then the voxels corresponding to the 'cut' boundary of the cylinder are replicated on the other side of the cuboid to account for the boundary conditions in the angular direction. Voxels lying in the center of the cylindrical volume are also replicated and permuted such that neighbor relationship of voxels is preserved.

The prior is modeled by an RBM. The usual structure of an RBM is bipartite, however as in our prior work [10], the RBM structure implemented is 4-partite. In contrast to [10] which was implemented using the connectivity structure of a Pegasus architecture D-Wave processor [14] this work uses the connectivity structure of the new Zephyr [15] architecture

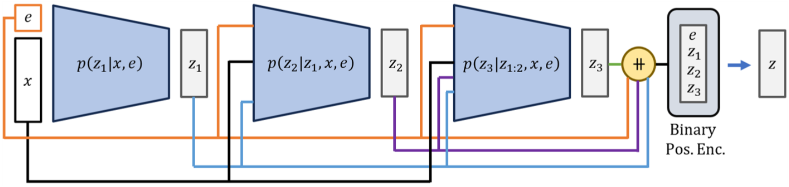


Figure 1. Hierarchical, autoregressive architecture of the encoder. The'#' symbol indicates concatenation.

processor, incorporating a higher degree of connectivity between nodes. In an RBM the nodes are binary valued and connections exist between partitions but not within partitions. This partitioned structure enables sampling from the RBM using so called block Gibbs sampling. The probability to sample a state is proportional to the negative exponent of the energy of the state - that is states are Boltzmann distributed. The energy of the state is expressed by the Eq 2:

$$
\begin{aligned}
E(\mathbf{v},\mathbf{h},\mathbf{s},\mathbf{t}) = & -a_i v_i - b_i h_i - c_i s_i - d_i t_i - v_i W^{(0,1)}_{ij} h_j - v_i W^{(0,2)}_{ij} s_j \\
& -v_i W^{(0,3)}_{ij} t_j - h_i W^{(1,2)}_{ij} s_j - h_i W^{(1,3)}_{ij} t_j - s_i W^{(2,3)}_{ij} t_j \ ,
\end{aligned}
\tag{2}
$$

where $\mathbf{v}$, $\mathbf{h}$, $\mathbf{s}$, $\mathbf{t}$ denote vectors of nodes in the four partitions and vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ and matrices $W^{(p,q)}$ are node biases and weight matrices connecting the partitions, respectively.

The binary nature of the nodes and the stochastic nature of the RBM necessitate special treatment to enable the propagation of the gradient and training of the full model through gradient descent. For this purpose we used the Gumbel trick [16] where smoothed versions of binary-valued RBM nodes are used and 'perturbed' with random noise. The smoothness is controlled by an inverse temperature parameter. We linearly anneal the Gumbel trick smoothness during training reaching very close approximation to the binary-valued variable.

We train the RBM using the *enhanced gradient* method, which has been shown to be invariant to bit flips and more robust during training [17, 18]. For this purpose, we hard-coded the RBM gradient while all other gradients were computed using automatic differentiation. In addition, we freeze the RBM parameters to the effect of automatic differentiation, to prevent updating the RBM parameters twice per batch.

The decoder incorporates a three dimensional convolutional architecture. Due to sparsity of the data - that is presence of multiple voxels not having any energy deposit in any given event, we introduced a binary valued mask in addition to the real valued output of the decoder. The mask and the real valued output are multiplied element-wise. The mask is trained using Binary Cross Entropy loss and the real valued output using the Mean Squared Error loss. The Gumbel trick is used again for the mask during training to enable gradient backpropagation.

All components of the model are 'conditioned' on the incident energy of the particle - that is their behavior is made dependent on the particle energy. The encoder and decoder are conditioned by concatenating the energy to the inputs of these networks. The latent space is conditioned by dedicating one partition to a binary encoding of the incident energy and fixing that partition during either classical or quantum sampling. Fixing the designated partition in the quantum implementation is accomplished (as developed in [10]) through setting of so called *flux bias* - effectively generating biases on qubits within the conditioning partition which are much larger than the sum of coupling terms connecting the conditioning qubits to qubits in other partitions.

D-Wave quantum annealers implement an ’initial’ Hamiltonian H_0 and a ’problem’ Hamiltonian H with annealing coefficients A and B respectively. The two Hamiltonians do not commute. The initial Hamiltonian is the transverse field while the problem Hamiltonian is an Ising model Hamiltonian - therefore a linear mapping exists between RBM Energy function and the problem Hamiltonian. The quantum annealing procedure encompasses putting the QPU in the ground state of H_0 and then annealing the coefficient A to zero while increasing the coefficient B. At the end of the anneal the generated state is a state of the problem Hamiltonian. By repeating this process, one obtains a set of states each with an energy that is Boltzmann distributed. However, the inverse temperature β^* is not known *a priori* and needs to be estimated. For optimization application the users rely on the fact that the ground state or a state close to ground state is achieved often enough for a practical application, while in our problem we want to achieve the same state distribution as in classical sampling of the RBM. To this effect we must employ an iterative procedure (fully described in [10]), where the weights and biases of the quantum Hamiltonian are successively scaled in such a way as to equate mean energy of the classical and quantum implementation of the RBM. In the previous Pegasus work we obtain the energy expectation of the quantum Hamiltonian by setting the Hamiltonian and flux bias once and obtaining a batch of samples in a given iteration. During the generation of synthetic data we found it is then necessary to introduce a wait time between successive sample generation. In this work we build a batch by setting the Hamiltonian parameters and flux bias once per sample. We then perform one iteration of the scaling procedure using a compiled batch of samples. The success of this procedure is illustrated in Fig. 2, where the Energy function of the RBM is evaluated on the encoded test data, data obtained from RBM by classical sampling and from sampling the QPU.

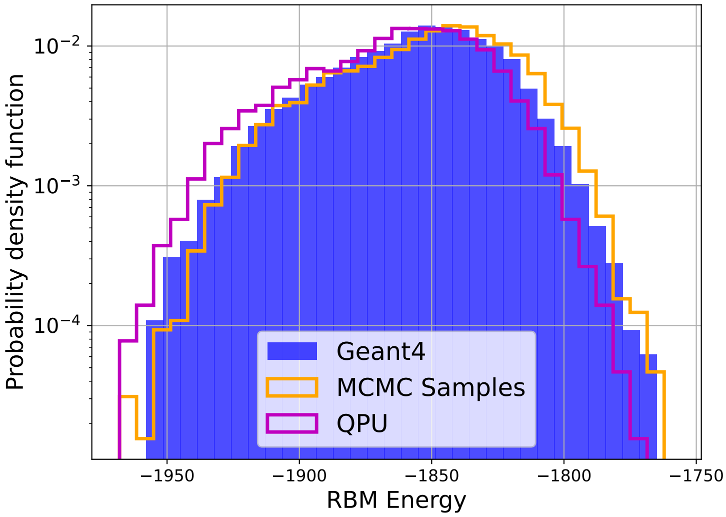
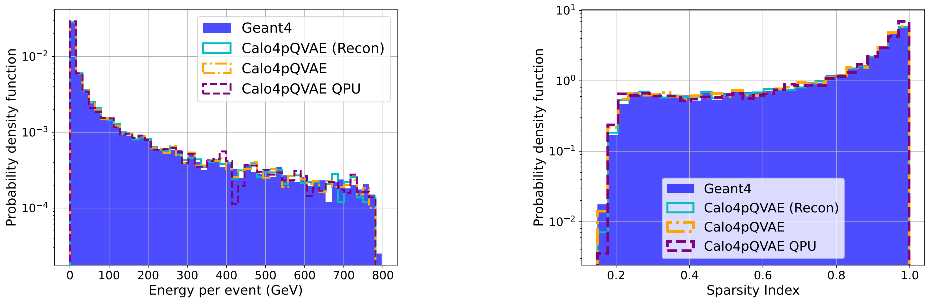


Figure 2. Comparison of energy function evaluation on the test data, data sampled classically and sampled using D-Wave QPU - indicating good performance of effective temperature scaling procedure.

4 Model performance

In this section we discuss preliminary results concentrating on the quality of the synthetic data generation as well as speed of sampling and energy cost with a comparison against selected deep generative models.

Figure 3 shows comparisons of total energy observed in the cylindrical calorimeter volume and the sparsity index computed over the whole volume. The sparsity index is defined as the ratio of voxels with zero observed energy to the total number of voxels in the volume. The histograms show the test data, test data reconstructed by our auto-encoding model, and synthetic data generated using classical and quantum sampling. Similarly in figures 4 and 5, the energy sum and the calculation of sparsity index is performed in ranges of layers of the calorimeter volume. All figures show good qualitative agreement mutually between reconstructed data, test data and synthetic data, with minor discrepancies observed in the tails of the distributions.



(a) Total energy observed in the calorimeter cylindrical volume.

(b) Sparsity index calculated over the entire calorimeter cylindrical volume

Figure 3. Energy sum over calorimeter cells and sparsity index

We also qualitatively study the performance of model conditioning. Figure 6 shows the distribution of sum of voxel energies in the entire cylindrical calorimeter volume where the incident electron energy has been restricted to a narrow window. Overlaid are distributions of the same data reconstructed by the auto-encoding model and distributions generated by classical and quantum sampling under conditioning corresponding to the test data in selected incident electron energy ranges. In the lower incident energy bins there is evident low bias, especially in the QPU samples, while in the higher energy bins qualitatively the distributions are well matched. We hypothesize that the observed bias may be due to the underlying logarithmic distribution of the training data. This remains to be confirmed in a future study - however the confirmation is not possible with the dataset used here.

More quantitative analysis is performed by computing *Kernel Physics Distance* and *Fréchet Physics Distance* [19]. These metrics are computed on high level quantities characterizing shower development in the calorimeter and are sensitive to mismodeling of individual features, correlations between features as well as so called *mode collapse* - where the generative model generates only a limited variety of samples instead of full target distribution. Table 1 shows the performance of our model contrasted to selected deep generative models based on metrics computed in [6]. Models selected for comparison include best on these metrics as well as ones performing similarly to our model in terms of energy cost.

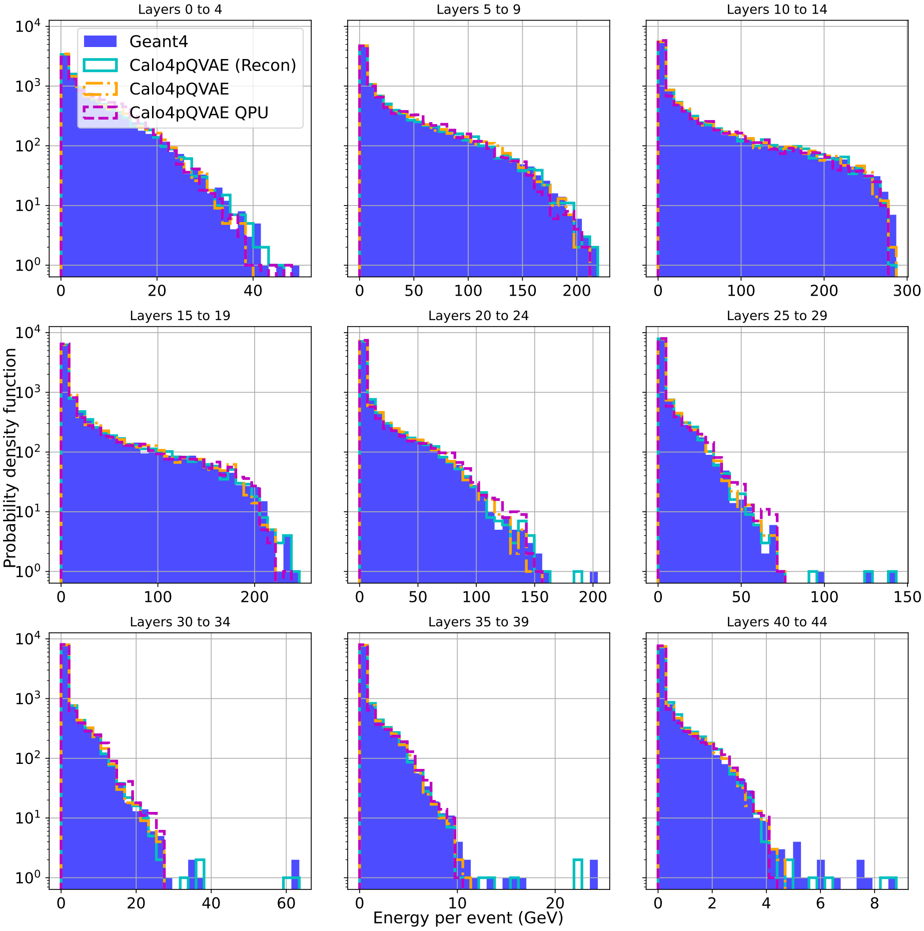


Figure 4. Observed energy sums in succesive layers of the cylindrical calorimeter volume.

Model	**FPD x** 10^3	**KPD x** 10^3
Calo4pQVAE (QPU)	328±3	0.49±0.16
CaloDream [20]	24±1	0.02±0.04
CaloDiffusion [4]	146±1	0.17±0.04
Convolutional L2LFlows [5]	157±1	0.27±0.09
CaloScore (single-shot) [13]	546±2	0.93±0.07

Table 1. Comparison of different models based on FPD and KPD metrics. All values except for our model summarized from [6]

We present preliminary results on energy consumption of our model as compared to deep generative models. The results here were obtained under a number of assumptions and do not encompass any embedded carbon costs. We converted per sample timing quoted in [6] for the deep generative models shown using the maximum power consumption of the GPU the models were evaluated on (A100-SXM-4 40 GB) with the maximal batch size that was

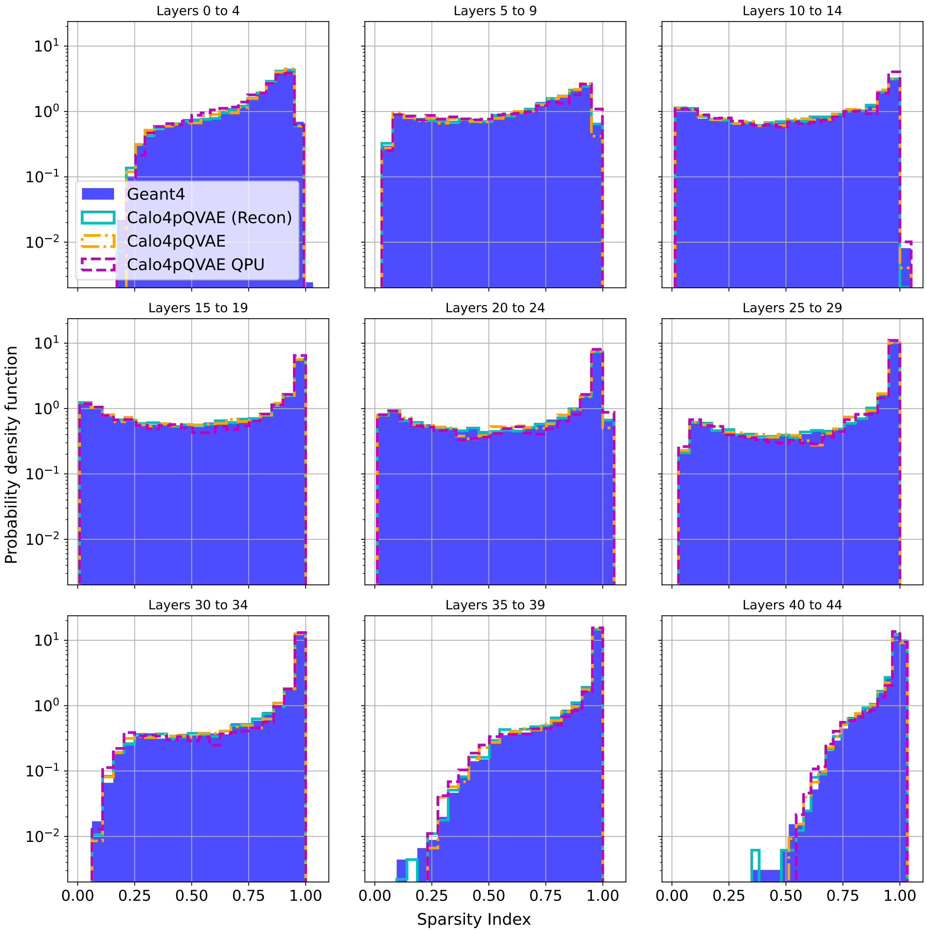
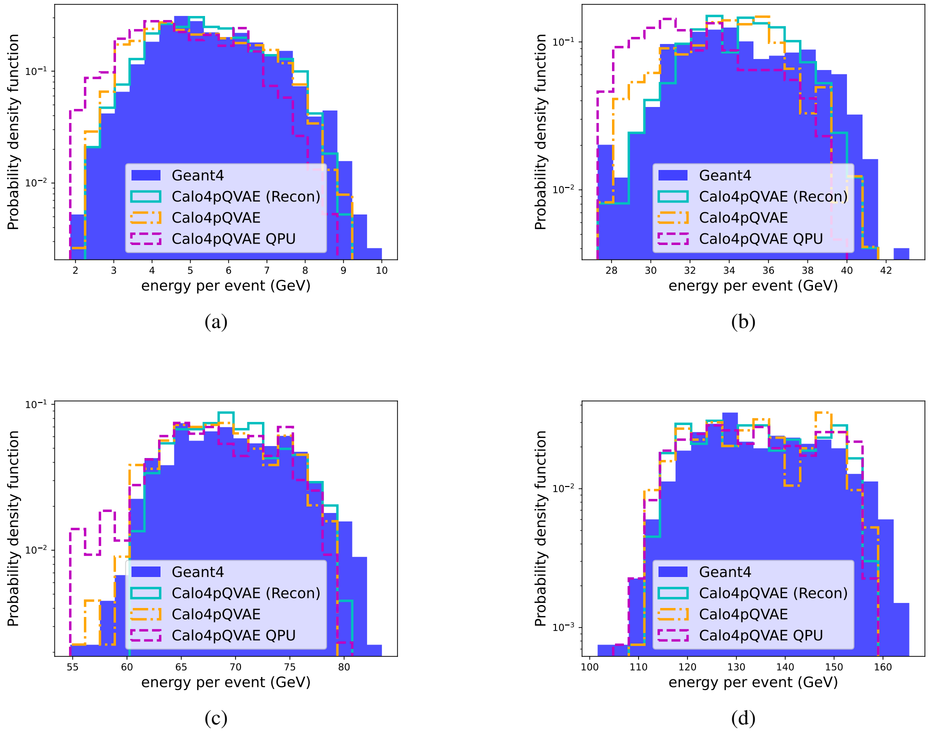


Figure 5. Sparsity in succesive layers of the cylindrical calorimeter volume.

possible and attempted in [6] - as this utilizes the GPU most efficiently. Further optimization of energy consumption of these deep generative models may be possible by fine-tuning the batch size. Our model timing was converted to energy consumption using the 16kW average power consumption of the dilution refrigerator [21]. Energy consumption of the QPU itself is negligible. We do not take into account the time spend re-programming the QPU for each sample, network latencies or similar engineering challenges - though naturally these will have to be addressed in the near future if the methodology described here is to be used at one of the HL-LHC experiments. We note that for our model the dominant time and energy cost is QPU readout, which could be an area of future optimization. The time taken to generate a single shower using the first principles GEANT4 [2] simulation varies vastly depending on the type and energy of the incident particle and the geometry of the detector being simulated. Different references [3], [13], give values ranging from 1s to 100s per shower for CPU generation. Here we take 1s per shower as a first approximation and multiply by average per core power consumption from a TRIUMF Tier 1 rack used for such simulations. In contrast, the timing and energy consumption of deep generative models, as well as our quantum-assisted genera-



(a)

(b)

(c)

(d)

Figure 6. Energy observed in full cylindrical calorimeter volume, while selecting events or conditioning the model according to the incident energy spectrum in the test set in successive incident electron energy ranges. 6(a): 5-10 GeV, 6(b): 40-50 GeV, 6(c): 80-100 GeV, 6(d): 150-200 GeV,

tive model ought to be independent of the conditioning variables. The results are summarized in Tab. 2

5 Discussion

We have outlined our quantum assisted model for generating synthetic calorimeter shower samples incorporating D-Wave Zephyr architecture. These preliminary results indicate that our model displays qualitatively good results and desired conditioning behavior (though improvement is still needed). According to the KPD and FPD metrics it is competitive with models consuming similar amount of energy per sample. Better models exist - however these may outstrip energy consumption of first principles simulation.

The work shown here motivates further studies of the quantum annealing based, quantum assisted generative models, and suggests the model has sufficient expressivity to tackle such datasets. Both present day model quality and energy consumption suggest that, on the timescales of the HL-LHC, models derived from this work may reach performance practicable for deployment.

Model	**Time / sample**	**Energy / sample [J]**
GEANT4	1 s	8
Calo4pQVAE (QPU)		
Annealing	20 μs	0.3
Readout	87 μs	2.2
Wait	20 μs	0.3
GPU postprocess	54 μs	<0.1
Total	**181 μs**	**2.0**
CaloDream [20]	74.3 ms	30
CaloDiffusion [4]	99.5 ms	40
conv. L2LFlows [5]	1.6 ms	0.6
CaloScore (single-shot) [13]	2.5 ms	1

Table 2. Comparison of models based on time and energy per sample. Timing values for the deep generative models summarized from [6]

6 Acknowledgments

We gratefully acknowledge Jack Raymond, Mohammad Amin, Trevor Lanting and Mark Johnson for their feedback and discussions. We gratefully acknowledge funding from the National Research Council (Canada) via Agreement AQC-002, Natural Sciences and Engineering Research Council (Canada), in particular, via Grants SAPPJ-2020-00032 and SAPPJ-2022-00020. This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Research, Innovation and Science. The University of Virginia acknowledges support from NSF 2212550 OAC Core: Smart Surrogates for High Performance Scientific Simulations and DE-SC0023452: FAIR Surrogate Benchmarks Supporting AI and Simulation Research. JQTM acknowledges a Mitacs Elevate Postdoctoral Fellowship (IT39533) with Perimeter Institute for Theoretical Physics.

References

[1] ATLAS Collaboration, Tech. rep., Technical report, CERN, Geneva. http://cds. cern.ch/record/2802918 (2022)

[2] S. Agostinelli, J. Allison, K.a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand et al., Geant4—a simulation toolkit, Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506**, 250 (2003).

[3] M. Paganini, L. de Oliveira, B. Nachman, Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks, Physical Review D **97**, 014021 (2018).

[4] O. Amram, K. Pedro, Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation, Physical Review D **108**, 072014 (2023).

[5] T. Buss, F. Gaede, G. Kasieczka, C. Krause, D. Shih, Convolutional l2lflows: generating accurate showers in highly granular calorimeters using convolutional normalizing flows, Journal of Instrumentation **19**, P09003 (2024). 10.1088/1748-0221/19/09/P09003

[6] C. Krause, M.F. Giannelli, G. Kasieczka, B. Nachman, D. Salamani, D. Shih, A. Zaborowska, O. Amram, K. Borras, M.R. Buckley et al., Calochallenge 2022: A community challenge for fast calorimeter simulation, arXiv preprint arXiv:2410.21611 (2024).

[7] G. Aad, B. Abbott, D.C. Abbott, A.A. Abud, K. Abeling, D.K. Abhayasinghe, S.H. Abidi, A. Aboulhorma, H. Abramowicz, H. Abreu et al., Atlfast3: the next generation of fast simulation in ATLAS, Computing and software for big science **6**, 7 (2022).

[8] A. Abhishek, E. Drechsler, W. Fedorko, B. Stelzer, Calodvae : Discrete variational autoencoders for fast calorimeter shower simulation (2022), 2210.07430, https://arxiv.org/abs/2210.07430

[9] S. Hoque, H. Jia, A. Abhishek, M. Fadaie, J.Q. Toledo-Marín, T. Vale, R.G. Melko, M. Swiatlowski, W.T. Fedorko, Caloqvae: Simulating high-energy particle-calorimeter interactions using hybrid quantum-classical generative models, The European Physical Journal C **84**, 1 (2024).

[10] J.Q. Toledo-Marin, S. Gonzalez, H. Jia, I. Lu, D. Sogutlu, A. Abhishek, C. Gay, E. Paquet, R. Melko, G.C. Fox et al., Conditioned quantum-assisted deep generative surrogate for particle-calorimeter interactions (2024), 2410.22870.

[11] Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Dalila Salamani, David Shih, Anna Zaborowska, Fast calorimeter simulation challenge 2022 - dataset 1,2 and 3. zenodo., https://doi.org/10.5281/zenodo.8099322, https://doi.org/10.5281/zenodo.6366271, https://doi.org/10.5281/zenodo.6366324 (2022)

[12] C. Krause, D. Shih, Caloflow: fast and accurate generation of calorimeter showers with normalizing flows, arXiv preprint arXiv:2106.05285 (2021).

[13] V. Mikuni, B. Nachman, Caloscore v2: single-shot calorimeter shower simulation with diffusion models, Journal of Instrumentation **19**, P02001 (2024).

[14] D-Wave Systems, Advantage processor overview, https://www.dwavesys.com/media/3xvdipcn/14-1058a-a_advantage_processor_overview.pdf (2022), accessed: 2023-11-07

[15] D-Wave Systems, Advantage processor overview, https://www.dwavesys.com/media/2uznec4s/14-1056a-a_zephyr_topology_of_d-wave_quantum_processors.pdf (2021), accessed: 2025-02-26

[16] C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, arXiv preprint arXiv:1611.00712 (2016).

[17] K. Cho, T. Raiko, A.T. Ihler, Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines, in *Proceedings of the 28th international conference on machine learning (ICML-11)* (Citeseer, 2011), pp. 105–112

[18] J. Melchior, A. Fischer, L. Wiskott, How to center deep boltzmann machines, Journal of Machine Learning Research **17**, 1 (2016).

[19] R. Kansal, A. Li, J. Duarte, N. Chernyavskaya, M. Pierini, B. Orzari, T. Tomei, Evaluating generative models in high energy physics, Physical Review D **107**, 076017 (2023).

[20] L. Favaro, A. Ore, S.P. Schweitzer, T. Plehn, Calodream - detector response emulation via attentive flow matching, arXiv preprint arXiv:2405.09629 (2024).

[21] D-Wave Systems, Computational power consumption and speedup (2014), white paper, accessed: December 4, 2025, https://www.dwavesys.com/media/ivelyjij/14-1005a_d_wp_computational_power_consumption_and_speedup.pdf

Changes with respect to the slides presented at the conference Attentive reader may observe small differences with respect to the results shown in these proceedings and the ones shown during the conference. The improvement in the results is attributed to training the model on a Zephyr architecture processor as opposed to the Pegasus architecture processor that was shown in the slides. At the time of the conference, the Zephyr results were not ready - however they became available soon after and the research team decided to let the updated results become the standing record. The conclusions discussed at the conference remain unchanged.