CaloDVAE : Discrete Variational Autoencoders for Fast Calorimeter Shower Simulation

Abhishek Abhishek[1]*, **Eric Drechsler**[1,2], **Wojciech Fedorko**[1], **Bernd Stelzer**[1,2]
[1] TRIUMF, Vancouver, BC V6T 2A3
[2]Simon Fraser University, Burnaby, BC V5A 1S6

Abstract

Calorimeter simulation is the most computationally expensive part of Monte Carlo generation of samples necessary for analysis of experimental data at the Large Hadron Collider (LHC). The High-Luminosity upgrade of the LHC would require an even larger amount of such samples. We present a technique based on Discrete Variational Autoencoders (DVAEs) to simulate particle showers in Electromagnetic Calorimeters. We discuss how this work paves the way towards exploration of quantum annealing processors as sampling devices for generation of simulated High Energy Physics datasets.

1 Introduction

With the advent of the High-Luminosity upgrade [1] and the expected increase in luminosity, experiments at the LHC are facing difficult computational challenges. A limiting factor on the precision of physics results is the lack of detailed Monte Carlo (MC) simulation in relevant phase spaces. This introduces a statistical uncertainty on measurements and hypothesis tests, limiting the sensitivity of LHC experiments. At the moment, billions of CPU hours [2, 3] are used by LHC experiments for MC simulation annually.

The simulation of particles interacting with the calorimeter system is computationally demanding. In sampling calorimeters, particles interact electromagnetically or hadronically with dense absorber material, resulting in a cascade of subsequent particles - a particle shower. Active layers, like liquid Argon [4], provide energy and location measurements from electric signals proportional to the number of particles produced in the cascade. The shower propagation and deposition of energy is an intrinsically probabilistic process. A full physics-based simulation with the state-of-the-art toolkit GEANT4 [5] can take minutes per event on current high-performance computing platforms [6, 7]. Approximate algorithms based on parameterizations [8, 9] have been used in many applications and significantly decrease the runtime at the expense of accuracy. However, the developments of new methods involving hadronic and tau-jet sub-structure information may require the complete shower to be simulated [10], rendering such approximations insufficient.

Recent developments [11–14] suggest, that deep generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are able to provide approximations to underlying probability distributions of calorimeter shower data. Generating independent random samples from such models is computationally cheap, thus rendering them promising candidates for replacing parts of the default simulation framework. The ATLAS experiment has incorporated a GAN for calorimeter shower generation in a recent update to their simulation infrastructure [14].

Inspired by these remarkable successes, we introduce a Discrete Variational Autoencoder (DVAE) [15–17] based model with hierarchical dependencies of latent variables in the approximate posterior and a Restricted Boltzmann Machine (RBM) latent prior. We study the qualitative performance of

*abhishek@myumanitoba.ca

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

this model on an idealized calorimeter dataset [18] where electromagnetic calorimeter showers are simulated. We demonstrate that this model tackles the challenges brought on by the non-uniformly segmented nature of the calorimeter, dependence between energy deposits in sequential layers and varying sparsity of the activated calorimeter cells.

Quantum Variational Autoencoders (QVAEs) [19, 20] are hybrid Quantum-Classical generative models which may be able to exploit quantum phenomena such as superposition and tunneling in quantum annealers to achieve better generative performance than their classical counterparts. This work is a first step towards applying QVAEs for calorimeter shower simulation and paves the way for future exploration and application of quantum annealing processors for generation of simulated High Energy Physics (HEP) datasets.

2 Methodology

2.1 Dataset and Preprocessing

We use the Electromagnetic Calorimeter Shower Images dataset [18] previously studied in [12]. The dataset contains energy deposits from positrons, photons and charged pions in an idealised, longitudinally segmented EM calorimeter. An incident particle of certain type, energy and direction is generated and its interaction with the calorimeter material simulated using the GEANT4 10.2.0 toolkit [5] with the FTFP_BERT physics list [21–27] using the electromagnetic physics package [28]. The calorimeter is a cube of volume 480 mm^3 with three non-uniformly segmented layers. The exact geometry can be found in the appendix Table 5.

In this work, we use a flattened representation where the energy deposits in each layer are "unrolled" and concatenated into a single feature vector for each example. The broad dynamic range of the energy deposited in a given calorimeter cell and the differences of energy deposit scales between different cells (e.g. cell near the middle vs near the edge of a layer) pose a challenge during training. Standardization is a common technique which makes data features approximately standard normally distributed, however is not suitable for highly sparse and either 0 or strictly positive calorimeter shower data. We modify the standardization procedure to work with calorimeter shower data as described in Appendix 6.1.

2.2 Deep Generative Models

Variational Autoencoders (VAEs) [29, 30] are a class of deep generative models that approximate the data distribution by optimizing an evidence lower bound (ELBO), $\mathcal{L}_{\phi,\theta}(\mathbf{x})$ to the log-likelihood of the data under the model distribution, $\log p_\theta(\mathbf{x})$:

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{autoencoding term}} - \underbrace{\mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}_{\text{kl term}} \leq \log p_\theta(\mathbf{x})$$

In the simplest case, the approximate posterior and prior over the latent variables are assumed to be factorized Gaussian distributions, $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), diag(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))$ and $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ respectively. The approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$ and generative, $p_\theta(\mathbf{x}|\mathbf{z})$ distributions are often parametrized using deep neural networks. The parameters ϕ and θ are optimized by minimizing the negative ELBO, $-\mathcal{L}_{\phi,\theta}(\mathbf{x})$ using stochastic gradient descent.

Discrete Variational Autoencoders (DVAEs) extend the VAE framework to allow discrete variables in the latent space [15–17]. The non-differentiability of discrete variables does not allow for the reparameterization trick [29, 30] to be used to compute low-variance gradient estimates of the autoencoding term w.r.t ϕ. In this work, we focus on the GumBolt-DVAE model [17] which extends the Gumbel trick [31, 32] for relaxing discrete distributions to work with Boltzmann machine (BM) priors. In GumBolt-DVAE, continuous proxy variables ζ are used in replacement of discrete variables $\mathbf{z}$ during training while the discrete variables $\mathbf{z}$ are used during validation and generation. The approximate posterior has a hierarchical structure, $q_\phi(\mathbf{z}|\mathbf{x}) = \prod_i q_{\phi_i}(\mathbf{z}_i|\mathbf{z}_{j<i}, \mathbf{x}), \mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ and the latent generative process is implemented by a restricted Boltzmann machine (RBM), $p_{\theta_{RBM}}(\mathbf{z}) = e^{-E_{\theta_{RBM}}(\mathbf{z})}/Z_\theta = e^{\mathbf{a}_\mathrm{l}^T\mathbf{z}_\mathrm{l} + \mathbf{a}_\mathrm{r}^T\mathbf{z}_\mathrm{r} + \mathbf{z}_\mathrm{l}^T\mathbf{W}\mathbf{z}_\mathrm{r}}/Z_\theta$, where Z_θ is the partition function. The RBM parameters (a_l, a_r, W) are jointly trained with the parameters (ϕ, θ). The complete set of latent variables predicted by the approximate posterior, $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ is partitioned into two equal

subsets which form the two sides $\{\mathbf{z}_\mathrm{r}, \mathbf{z}_\mathrm{l}\}$ of the RBM. The hierarchical approximate posterior and RBM prior allow for rich latent space distributions and improve the generative performance of the model [15–17].

2.3 CaloDVAE

Our simulation technique is based on the GumBolt-DVAE framework. Figure 1 shows a graphical description of our model. We employ energy conditioning in a similar fashion to [13]. The approximate posterior distribution at a given hierarchy level i and the generative distribution are specified as $q_{\phi_i}(\mathbf{z}_i|\mathbf{z}_{j<i}, \mathbf{x}, e)$ and $p_\theta(\mathbf{x}|\mathbf{z}, e)$ respectively, where e is the true energy of the incident particle in GeV. Fully connected neural networks (FCNNs) with ReLU activation functions are used to parametrize $q_{\phi_i}(\mathbf{z}_i|\mathbf{z}_{j<i}, \mathbf{x}, e), i = 1, \ldots, n$ and $p_\theta(\mathbf{x}|\mathbf{z}, e)$. In practice, since we use a flattened representation, the true incident particle energy, e is simply concatenated to the input feature vector. The resulting vector, concatenated with $\{\mathbf{z}_{j<i}\}$ is passed through a sequence of non-linear fully connected layers to obtain approximate posterior samples $\mathbf{z}_i$ at a given hierarchy level i. During the autoencoding phase, approximate posterior samples $\{\mathbf{z}_i, i = 1, \ldots, n\}$ are concatenated with e and passed through a sequence of non-linear fully connected layers to obtain a resampled version of the input x. During the generation phase to obtain new samples, RBM latent variable samples $\mathbf{z} \sim p_{\theta_{RBM}}(\mathbf{z})$ obtained using block Gibbs sampling, concatenated with the requested incident particle energy e, are passed through a sequence of non-linear fully connected layers.

Output masking Previous studies on the calorimeter dataset identified limitations in capturing the layer sparsity distributions in particular for charged pions[12]. To overcome this, we introduce stochastic discrete variables $\mathbf{x}_m$ in the generative model where each $x_{m,i} \in \{0, 1\}$ determines whether the calorimeter cell i is hit. These variables, in addition to the generated energy deposits $\mathbf{x}_e$ are used to produce the final output, $\mathbf{x} = \mathbf{x}_m \odot \mathbf{x}_e$, $\odot$ denotes the Hadamard product. In practice, a hidden vector $\mathbf{x}_0$ is first obtained by passing either approximate posterior samples or RBM samples concatenated with the incident particle energy through a sequence of non-linear fully connected layers. $\mathbf{x}_0$ is then passed through a second set of non-linear fully connected layers to obtain $\mathbf{x}_m$ and $\mathbf{x}_e$ independently. A ReLU activation function is applied to $\mathbf{x}_e$ since $\forall i, x_{e_i} \geq 0$ and to encourage sparsity [11, 12]. We use the Gumbel trick [31, 32] for discrete variables $\mathbf{x}_m$ during training to ensure differentiability (i.e. continuous proxy variables are used instead during training). Binary Cross Entropy (BCE) loss applied to $\mathbf{x}_m$ and Mean Squared Error (MSE) loss applied to $\mathbf{x}$ are summed to compute the total autoencoding loss used to train the model.

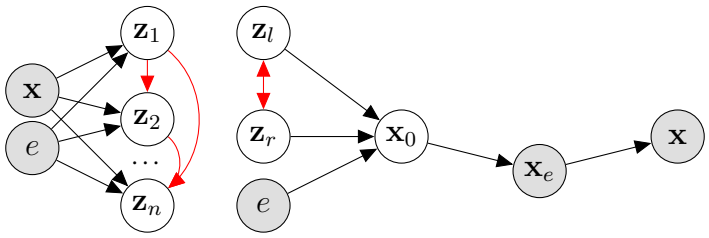

Figure 1: Graphical description of the hierarchical approximate posterior $q_\phi(\mathbf{z}|\mathbf{x}, e)$ (left) and generative model $p_\theta(\mathbf{x}|\mathbf{z}, e)$ (right) to generate new synthetic samples. In the inference model (left), continuous proxies $\boldsymbol{\zeta}_i$ are used instead of discrete $\mathbf{z}_i$ during training. e is the true or requested energy of the incident particle in GeV.

3 Preliminary Results

We performed a grid search to heuristically determine the best hyperparameter setting for each particle type. A separate model with the best hyperparameter setting is trained for each particle type and used to produce the results. We include the details on the hyperparameter scans and settings used to produce the following results in Appendix 6.4.

Qualitative assessment of shower images of CaloDVAE samples (Appendix 6.2, Figure 3) reveals that a broad variety of samples are generated by our model, reproducing features such as the patterns

of activated and non-activated cells, centrality and lateral width of the clusters, as well as longitudinal behaviour of the shower. Of note is the behaviour in the last two layers in the pion sample, where some of the generated showers have very little deposit in these two layers, whereas in other samples large energy deposit is seen - a feature observed in the simulated training data. Our model also displays good energy conditioning and extrapolation behavior beyond the energy region in which it has been trained (*cf.* Appendix 6.5).

Shower shape variables are determined by the transverse and longitudinal profile of the shower, and are useful for particle identification and energy calibration [12]. We present 1D histograms for a subset of these variables in Figure 2 and their description in Appendix 6.3, Table 1. GEANT4 samples from the test subset of the dataset and CaloDVAE samples with $e \sim \mathcal{U}[1, 100]$ GeV were used to fill the histograms. The shower shape distributions approximately match at different scales and in particular, layer sparsity distributions which were previously observed to be challenging are recovered faithfully. Correct modelling of the bi-modal sparsity distribution for charged pions in layer 1 is quite notable since they undergo both hadronic and electromagnetic interactions.

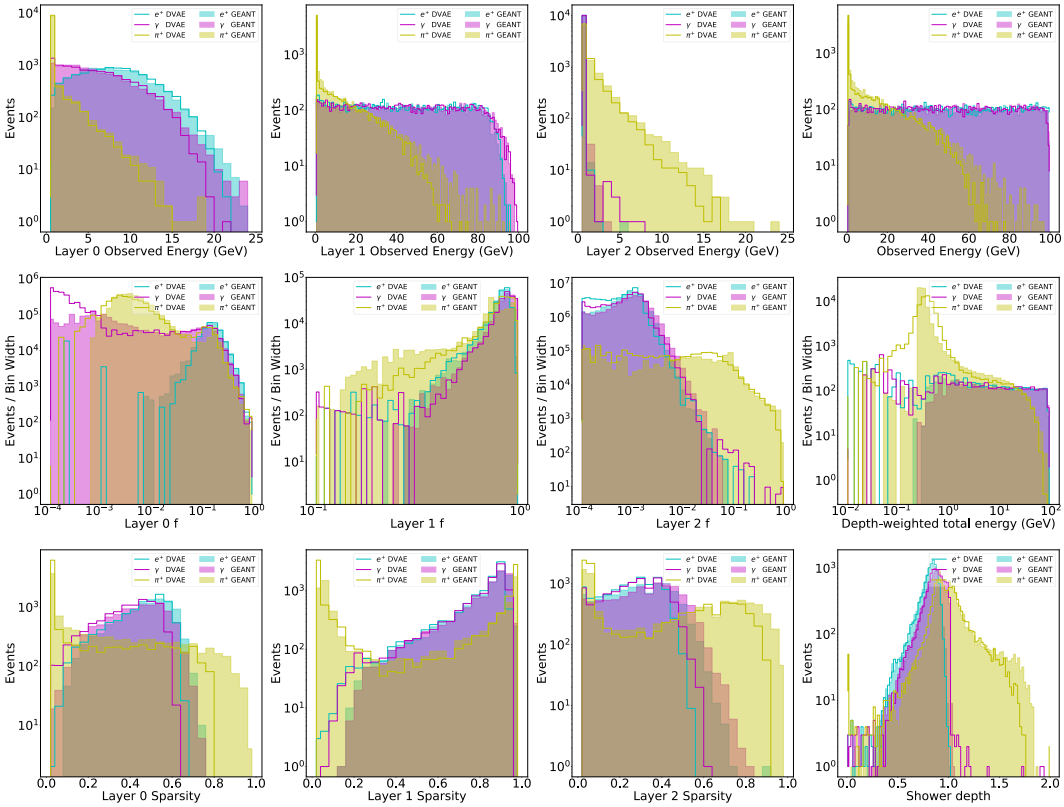

Figure 2: Shower shape variables (Appendix 6.3, Table 1) and layer sparsity (fraction of cells hit) distributions for GEANT4 and CaloDVAE samples.

4 Discussion and Future Outlook

QVAE In QVAEs, a Quantum Boltzmann machine (QBM) [33] replaces the restricted Boltzmann machine (RBM) implementing the latent generative process in the DVAE framework [19, 20] and offloads the classical latent space sampling to a quantum annealer. Quantum annealers operated as sampling devices may provide a computational advantage over Markov Chain Monte Carlo (MCMC) techniques when using large latent-space BMs. Previous work [20] has shown remarkable success on the MNIST and FMNIST datasets but notes that more complex datasets are required to fully exploit the large BMs in the latent space. The successful reproduction of high level physics observables by our model indicates that models of this class have high enough expressibility (enabled by the trainable complex prior) to model High Energy Physics (HEP) datasets such as calorimeter showers. Therefore,

generative modelling of HEP datasets may benefit from using quantum annealers as Boltzmann sampling devices. Our work provides a template for application of similar techniques to other HEP datasets and is a necessary first step towards the exploration and application of quantum annealing processors for generation of simulated HEP datasets.

5 Broader Impact

We considered potential negative impacts of the research presented. Indirectly, the method presented can be used to create deceptive fake data - a concern common to most generative methods. We note that this concern already exists with the foundational works [15–17] upon which this application work builds. Within the presented application domain the negative impact of the work may include production of biased simulation samples and thus affecting scientific results that rely on these samples. This concern would also apply to any traditional or novel method of generating simulated data.

We believe the potential negative impacts are offset by the positive impacts on science and society. If the full potential of the work presented here is eventually realized - i.e. if quantum processors can be harnessed for the generation of synthetic data - millions of CPU years per year could be saved that would have to be otherwise devoted to the task of simulated data generation for the HL-LHC experiments. This saving can have enabling impact in terms of fiscal considerations, but also will contribute to reduced environmental footprint. The sensitivity of physics analysis could also be improved through the availability of large synthetic datasets. We also note the potential for the development of semi-supervised methods based on the methodology presented, thus enabling learning on real experimental data and potential reduction of systematic uncertainties in the final physics analyses in HEP experiments.

Acknowledgements

We gratefully acknowledge the support of NSERC and Compute Canada. The authors would like to thank Olivia Di Matteo, Mohammad Amin and Walter Vinci for helpful discussions. Weights and Biases with an academic license was used for experiment tracking [34].

References

[1] P. Calafiura et al. ATLAS HL-LHC Computing Conceptual Design Report. Technical report, CERN, Geneva, Sep 2020.

[2] E. Karavakis et al. Common Accounting System for Monitoring the ATLAS Distributed Computing Resources. *J. Phys. Conf. Ser.*, 513:062024, 2014.

[3] C. Bozzi. LHCb Computing Resource usage in 2014 (II). Technical report, CERN, Geneva, Jan 2015.

[4] T. G. McCarthy. Upgrade of the ATLAS Liquid Argon Calorimeters for the High-Luminosity LHC. In *2016 IEEE Nuclear Science Symposium and Medical Imaging Conference*, page 8069859, 2016.

[5] S. Agostinelli et al. GEANT4–a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250–303, 2003.

[6] G. Aad et al. The atlas simulation infrastructure. *The European Physical Journal C*, 70(3):823–874, Sep 2010.

[7] R. Rahmat et al. The fast simulation of the CMS experiment. *J. Phys. Conf. Ser.*, 396:062016, 2012.

[8] G. Grindhammer and S. Peters. The Parameterized simulation of electromagnetic showers in homogeneous and sampling calorimeters. In *International Conference on Monte Carlo Simulation in High-Energy and Nuclear Physics - MC 93*, 2 1993.

[9] M. Beckingham et al. The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim. 10 2010.

[10] F. Dias. The new ATLAS Fast Calorimeter Simulation. *PoS*, ICHEP2016:184, 2016.

[11] L. de Oliveira et al. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science*, 1(1):1–24, 2017.

[12] M. Paganini et al. Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D*, 97(1), January 2018.

[13] Deep generative models for fast shower simulation in ATLAS. Technical report, CERN, Geneva, Jul 2018. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2018-001.

[14] ATLAS Collaboration. Atlfast3: the next generation of fast simulation in atlas, 2021.

[15] J. T. Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[16] A. Vahdat et al. Dvae++: Discrete variational autoencoders with overlapping transformations. In *International Conference on Machine Learning*, pages 5035–5044. PMLR, 2018.

[17] A. H. Khoshaman and M. H. Amin. Gumbolt: Extending gumbel trick to boltzmann priors. *arXiv:1805.07349*, 2018.

[18] B. Nachman et al. Electromagnetic calorimeter shower images, 2017. Mendeley Data, V1.

[19] A. Khoshaman et al. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1):014001, 2018.

[20] W. Vinci et al. A path towards quantum advantage in training deep generative models with quantum annealers. *Machine Learning: Science and Technology*, 1(4):045028, 2020.

[21] B. Andersson et al. Final state interactions in the (nuclear) FRITIOF string interaction scenario. *Z. Phys. C*, 70:499–506, 1996.

[22] B. Andersson et al. A Model for Low p(t) Hadronic Reactions, with Generalizations to Hadron - Nucleus and Nucleus-Nucleus Collisions. *Nucl. Phys. B*, 281:289–309, 1987.

[23] B. Nilsson-Almqvist and E. Stenlund. Interactions Between Hadrons and Nuclei: The Lund Monte Carlo, Fritiof Version 1.6. *Comput. Phys. Commun.*, 43:387, 1987.

[24] B. Ganhuyag and V. Uzhinsky. Modified FRITIOF code: Negative charged particle production in high energy nucleus nucleus interactions. *Czech. J. Phys.*, 47:913–918, 1997.

[25] M. P. Guthrie et al. Calculation of the capture of negative pions in light elements and comparison with experiments pertaining to cancer radiotherapy. *Nucl. Instrum. Meth.*, 66:29–36, 1968.

[26] H. W. Bertini and M. P. Guthrie. News item results from medium-energy intranuclear-cascade calculation. *Nucl. Phys. A*, 169:670–672, 1971.

[27] V. A. Karmanov. Light-front wave function of a relativistic composite system in an explicitly solvable model. *Nucl. Phys. B*, 166:378–398, 1980.

[28] H. Burkhardt et al. Geant4 standard electromagnetic package for HEP applications. In *2004 IEEE Nuclear Science Symposium and Medical Imaging Conference*, number 3, pages 1907–1910, 2004.

[29] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2014.

[30] D. J. Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014.

[31] E. Jang et al. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[32] C. J. Maddison et al. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[33] M. H. Amin et al. Quantum boltzmann machine. *Physical Review X*, 8(2):021050, 2018.

[34] L. Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We believe the scope of the work is accurately represented in the abstract.
   (b) Did you describe the limitations of your work? [Yes] See section 4
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We submit that the work presented adheres to the ethical standards outlined in the 'Ethics Guidelines' document
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Release of the code at this time proved impractical as the work is progressing toward more advanced applications
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We outline the method of data pre-processing and hyperparameters used and methodology for selecting the final choice in Appendices 6.1 and 6.4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We do not repeat the training sessions with multiple random seeds as precise quantification of the agreement of the generated dataset with the primary samples is beyond the scope of the paper. We argue that quantification of such agreement over multi-dimensional feature space is challenging in principle, and generally not done in similar expense. Even barring this the toy nature of the dataset would not justify the expense of repeat trials to estimate the 'error bar' on such agreement. The scope of the paper includes only the assesment if the discrete latent variable generative models are capapble in principle of reproducing complex physics distributions.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not rigorously track resource and computing time used on the variety of assets we used for training and evaluation of our model as this is out of scope of the current state of the study - which concentrated on establishing the feasibility of discrete latent space model to generate complex physics datasets. We do not claim that processing/sample generation time of this particular model would be competitive in relation to other generative models. This is because of the Monte Carlo sampling stage inherently built into sampling of the RBM - however the application of quantum processors is expected to address this.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 2.1
   (b) Did you mention the license of the assets? [No] We do not mention the license, however it is easily found by following the reference in the text.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Data is available under CC BY 4.0 licence

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

6 Appendix

6.1 Dataset standardization

Applying standardization to data features makes them standard normally distributed by removing the mean and scaling to unit variance. However, in the case of calorimeter shower data, a value of 0 for a given cell denotes a non-hit cell. Since the raw values in our data correspond to energy depositions, they are either 0 or strictly positive. To achieve some form of standardization while preserving the "physical" characteristics, we scale each cell independently using the following steps.

For a given cell i, let $X_{i\neq 0}$ and $X_{i=0}$ denote the sets of dataset samples in which cell i is hit and not hit respectively. The values of cell i for samples in $X_{i\neq 0}$ are standardized using mean μ_i and variance σ_i^2 computed over $X_{i\neq 0}$. Additionally, to maintain a distinction b/w samples in which cell i is hit and not hit, we shift the values of cell i for samples in $X_{i\neq 0}$ by their smallest value plus an ϵ if the smallest value is negative. ϵ is a small positive number, e.g. 0.01. This allows to maintain a distinction between hit and non-hit cells.

6.2 Shower images

An example in this dataset can be represented as 3 grayscale 2D images in the $(\eta - \phi)$ space, where η is the beam direction in an experiment and ϕ is direction perpendicular to both η and z, the particle propagation direction. The intensity of a pixel is the amount of energy deposited in the corresponding calorimeter cell.

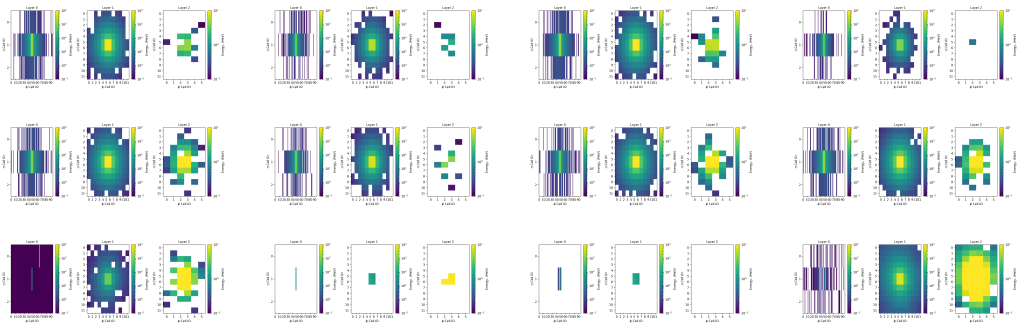

Figure 3: Examples of shower images of CaloDVAE samples for e^+ (top), γ (middle) and π^+ (bottom) with incident particle energy $e \sim \mathcal{U}[0, 100]$ GeV.

6.3 Shower Shape Variables

Shower Shape Variable	Formula	*Notes*
E_i	$E_i = \sum_{\text{pixels}} \mathcal{I}_i$	Energy deposited in the i^{th} layer of calorimeter
E_{tot}	$E_{\text{tot}} = \sum_{i=0}^{2} E_i$	Total energy deposited in the electromagnetic calorimeter
f_i	$f_i = E_i / E_{\text{tot}}$	Fraction of measured energy deposited in the i^{th} layer of calorimeter
Depth-weighted total energy, l_d	$l_d = \sum_{i=0}^{2} i \cdot E_i$	The sum of the energy per layer, weighted by layer number
Shower Depth, s_d	$s_d = l_d / E_{\text{tot}}$	The energy-weighted depth in units of layer number

Table 1: Variables characterizing the properties of the simulated showers. [12]

6.4 Hyperparameter Scan

We split the 100,000 GEANT4-simulated event dataset using a 80%-10%-10% split for train, validation, and test subsets, respectively. An optimal setting for the model and training parameters was determined heuristically independently for each incident particle type. Three qualitatively different model settings were investigated; their specifications are listed in Table 2. The models' performances were evaluated over a grid of training hyperparameters, summarised in Table 3. In order to determine the best parameter sets, the distributions of shower shape variables for GEANT4 samples from the validation subset of the dataset and CaloDVAE samples were compared using a Kolmogorov-Smirnov (KS) test. The optimal setting selected as the one maximising the KS probability over the complete set of shower variables.

It was observed that for all particle types the model architecture III and IV generalised best. This indicates that additional depth in encoder and decoder, as well as an increase in dimensionality in the latent space provides a more powerful model, capturing the underlying dataset complexity more efficiently.

Parameter	Encoder Layers	Decoder Layers	Hierarchy Levels	Latent Nodes Per Hierarchy Level
Model I	$[400, 300, 200]$	$[200, 300, 400]$	2	64
Model II	$[400, 350, 300, 200]$	$[200, 300, 350, 400]$	4	128
Model III	$[400, 350, 300, 250, 200]$	$[200, 250, 300, 350, 400]$	4	128
Model IV	$[500, 450, 400, 350, 300]$	$[300, 350, 400, 450, 500]$	6	150

Table 2: Different model architectures explored in the hyper-parameter scan. Each successive model grows in complexity by adding layers in encoder and decoder, introducing additional hierarchy levels and increasing the latent space dimensionality.

Parameter	Range
Learning Rate	$[0.01, 0.005, 1.e^{-3}, 10^{-4}, 0.5 \times 10^{-4}]$
Epochs	$[25, 50, 75, 100]$
Batch Size	$[50, 64, 75, 100, 128, 192]$
Latent smoothing temp. $\tau_{\mathbf{z}}$	$[1/5, 1/7, 1/9]$
Output mask smoothing temp. $\tau_{\mathbf{x}_m}$	$[1/5, 1/7, 1/9]$

Table 3: Grid of parameters considered for the hyperparameter optimization using the three model definitions in Table 2. Latent smoothing temp. $\tau_{\mathbf{z}}$ and output mask smoothing temp. $\tau_{\mathbf{x}_m}$ are parameters of the Gumbel trick used to control the "smoothness" of the continuous proxy variables. [31, 32]

6.5 Energy conditioning

We study the performance of energy conditioning of the model. We sample from the model requesting specific values of true energy (1, 25, 50, 100 and 150 GeV) of the incident particle and histogram total observed energy in the cluster. As shown in Fig 4 photon and electron clusters display sharp peaks at the requested energy values, whereas pion samples display broadened response - however this is due to the nature of the uncontained charged pion shower in the electromagnetic calorimeter and not due to poor model conditioning. Note that the 150 GeV exceedes the energy range where the models have been trained.

	Positron e^+	Photon γ	Charged Pion π^+
Model Type	Model II	Model IV	Model IV
Learning Rate	10^{-4}	0.5×10^{-4}	10^{-4}
Epochs	100	100	100
Batch Size	100	100	100
Latent smoothing temp. $\tau_{\mathbf{z}}$	$1/5$	$1/7$	$1/5$
Output mask smoothing temp. $\tau_{\mathbf{x}_m}$	$1/5$	$1/5$	$1/9$

Table 4: Selected Models per incident particle type after heuristic evaluation of the hyperparameter scan used to produce the preliminary results. For definitions and ranges, see Table 2 and Table 3.

Layer	Δz [mm]	$\Delta\eta$ [mm]	$\Delta\phi$ [mm]
0	90	5	160
1	347	40	40
2	43	80	40

Table 5: The geometry of the calorimeter. The z-axis corresponds to the direction of particle propagation, the η- and ϕ-axes are perpendicular to this [12].

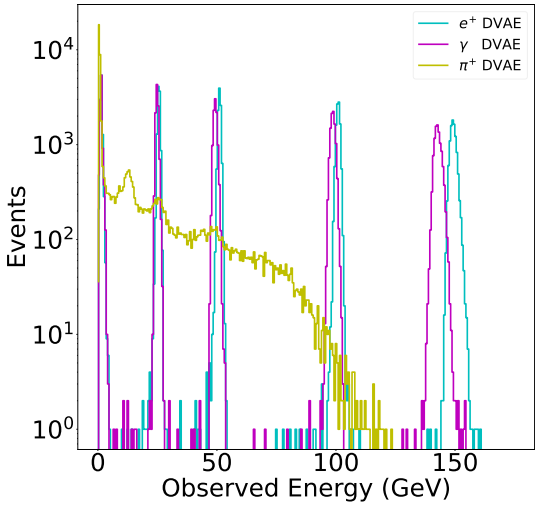

Figure 4: Observed energy spectra for synthetic CaloDVAE e^+, γ and π^+ samples generated with true incident energies of 1, 25, 50, 100 and 150 GeV.