CaloFFJORD: High Fidelity Calorimeter Simulation Using Continuous Normalizing Flows

Chirag S. Furia
Brown University
Providence, RI 02912
chirag_furia@brown.edu

Vinicius Mikuni
National Energy Research Scientific Computing Center
Berkeley, CA 94720
vmikuni@lbl.gov

Abstract

High fidelity simulation of detector components in collider physics is computationally expensive and often not scalable to the requirements of future experimental facilities. In this work, we present a fast and accurate alternative for detector simulation based on continuous normalizing flows for calorimeter simulation named CALOFFJORD, able to reproduce high-fidelity calorimeter responses in a fraction of the time compared to full simulation routines. We evaluate our model using different detector simulations and show that CALOFFJORD can improve the fidelity of detector simulations by incorporating data symmetries that are harder to encode within standard normalizing flow architectures.

1 Introduction

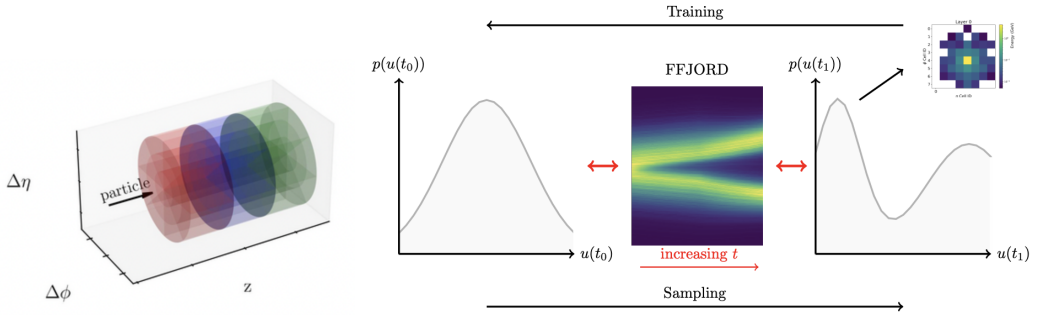

Figure 1: On the left is a 3D diagram of a calorimeter showing the direction of an incident particle. On the right is a visualization of CALOFFJORD. In the training step, the likelihood of calorimeter showers within the target space $u(t_1)$ is maximized. Afterwards, samples from the latent space $u(t_0)$ can be passed through the model to generate new showers. The regularized path between the two distributions represents the continuous change of variables parameterized by t.

The simulation of particle interactions is a fundamental task in collider physics to compare theories with experimental results. Among the different aspects to be simulated, the detector response is often the most computationally expensive (1) due to millions of detector readouts that need to be reproduced with precision by full physics simulators such as GEANT4(2; 3; 4). In particular, calorimeter simulations are challenging since particle interactions with the detector material lead to cascades of particles being created and recorded by the hundreds of thousands of detector readouts from modern calorimeter detectors. This challenge motivates the development of fast surrogate models that are able to reproduce the detector response much faster than a full detector simulation.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

Machine learning models have the capability of addressing this problem by providing an accurate description of the full detector complexity and leveraging hardware accelerators, such as Graphics Processing Units (GPUs), resulting in both accurate and hundreds to thousands of times faster generation of new physics observables. In this work, we focus on continuous normalizing flows based on the idea proposed in (5), a class of generative algorithms based on invertible transformations that were shown to produce high fidelity simulations (6) by directly learning the likelihood of the data. The transformation of the probability density is described by the following equation:

$$\log p_x(x) = \log p_z(z) - \log \det \left| \frac{\partial f(z)}{\partial z} \right|, \tag{1}$$

for data $z \sim p_z(z)$ that is transformed through the invertible function f to $x = f(z) \sim p_x(x)$. While the bijective function f can represent any invertible transformation, the complexity of the Jacobian calculation often restricts the class of allowed functions that can be used. This restriction can be lifted by leveraging the instantaneous change of variables formula:

$$\frac{\partial \log p(z(t))}{\partial t} = -\mathrm{Tr}\left(\frac{\partial f}{\partial z(t)} \right), \tag{2}$$

where the parameter t has been introduced as an intermediate variable that upon integration results in:

$$\log p(z(t_1)) = \log p(z(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial z(t)} \right) dt. \tag{3}$$

The advantage of this approach is that the initial complexity of $\mathcal{O}(N^3)$ is reduced to the complexity of the trace determination that is of $\mathcal{O}(N^2)$ with the dimensionality of the data N. The complexity can be further reduced by using stochastic trace approximators such as the Hutchinson trace estimator (7) leading to a linear complexity. On the other hand, the integral of equation 3 needs to be solved at each estimation of the density, but still leads to a speed-up both in training and sampling compared to the exact trace estimation. In this work, we employ a continuous normalizing flow model known as Free-Form Jacobian of Reversible Dynamics (FFJORD) to reproduce the detector response of two different calorimeter simulations, named CALOFFJORD. The first dataset corresponds to the CALOGAN dataset (8). The second one uses a regular detector structure based on dataset 2 of the ongoing CALOCHALLENGE (9). Results are compared with a different normalizing flow implementation named CALOFLOW.

In order to stabilize and reduce training times, the FFJORD implementation of continuous normalizing flows in TensorFlow Probability is modified to incorporate two regularization terms. The first term is a kinetic penalty defined by $||f||^2$, while the second is a Jacobian penalty defined over the vector-Jacobian product, $||\epsilon^T \frac{\partial f}{\partial z(t)}||^2$ (10). Factors of 0.01 are applied to both regularization terms in CALOFFJORD.

2 Datasets

Calorimeter showers are generated when a single particle with energy E hits the detector and starts a cascading energy transfer through the detector's material. In Dataset I, the CALOGAN dataset, the calorimeter consists of 3 layers. To minimize the number of empty measurements, multiple readout channels are combined together through a voxelization procedure, resulting in a different number of components in each layer. These are subdivided into 2D Cartesian grids of voxels. The first, second, and third layers have dimensions 3×96, 12×12, and 12×6, respectively. Thus, each GEANT simulation is characterized by its incident energy E and energy depositions for the 504 voxels. In Dataset II, the calorimeter consists of 9 layers with cylindrical coordinates used to subdivide each layer in the GEANT simulations. These were converted to Cartesian coordinates in order to create uniform 8×8 2D meshes for each layer, yielding 576 voxels. Input energies used to train the model follow the preprocessing strategy introduced in CALOFLOW to simplify the comparison.

3 Model architectures and training

CALOFLOW consists of two distinct normalizing flows that each combine masked autoregressive layers for distribution estimation (MADE) and neural spline flows (6). The first flow learns the energy

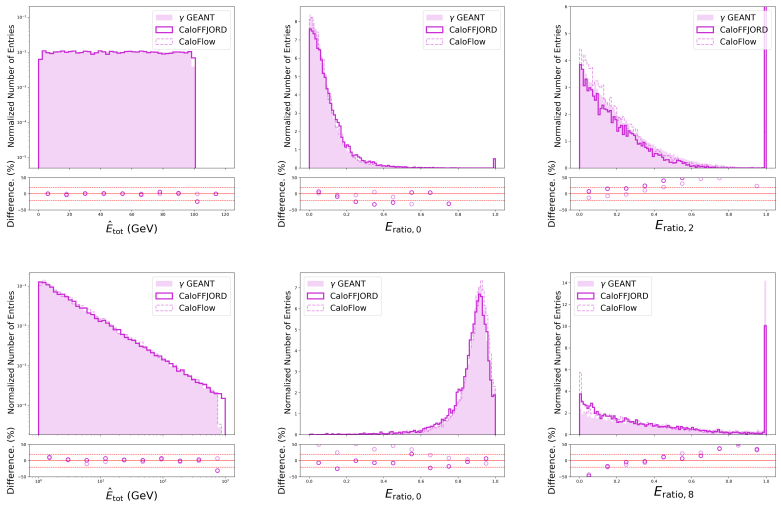

Figure 2: Distributions of the total energy and the normalized difference between the two brightest voxels for the first and last layers in Dataset I (top row) and Dataset II (bottom row).

distribution across the calorimeter layers conditioned on the initial energy of the incoming particle, while the second flow learns the normalized shower distribution conditioned on the energy per layer and initial particle energy. For both Dataset I and Dataset II, Flow I was trained with 6 MADE blocks and Flow II with 8 MADE blocks.

CALOFFJORD uses a visual transformer (11) architecture with initial image patches built using 1D convolutional layers for the CaloGAN dataset and 3D convolutional layers in Dataset II, resulting in 42 patches in Dataset I and 36 for Dataset II. A position embedding is added to the created patches before the transformer with a single attention head. The output of the transformer is then combined with two fully connected layers of size 1024 and hyperbolic tangent activation before the output layer with linear activation. The time information is concatenated to the data at the input of each hidden layer. CALOFFJORD uses the same strategy of decomposing the generative model into two components. The direct estimation of the energy per layer is achieved using a RESNET-like (12) architecture with 8 fully connected layers and skip connections between every 2 layers.

A classifier with the same architecture used to evaluate CALOFLOW (6) is applied to evaluate generated showers from both CALOFFJORD and CALOFLOW against ground-truth showers from GEANT for Dataset II and is used as a quantitative metric to compare the quality of the generation.

4 Results

The top row of Figure 2 contains plots of multiple observables from Dataset I. The histogram on the left shows the distribution of the total energy $\hat{E}_{tot}$ distributed across all of the layers given incident energies following a uniform distribution over 1-100 GeV. The center and right histograms show the normalized difference between the brightest and second-brightest voxels for layers 0 and 2. As highlighted by the ratio plots beneath the histograms that compare both models against the ground truth from GEANT, CALOFFJORD is more accurate at the extreme $E_{ratio} = 0$ when two nearby voxels share similar energies, while CALOFLOW has an advantage for larger values as CALOFFJORD yields peaks at the opposing extreme $E_{ratio} = 1$, where all of the energy is deposited in one voxel.

The bottom row contains plots of the same observables from Dataset II. The total energy histogram is now based on incident energies that follow a log-uniform distribution over 1-1000 GeV. CALOFFJORD shows small discrepancies at the higher energies, as evidenced by the corresponding ratio plot. The center and right histograms show the E_{ratio} distributions for layers 0 and 8. Both models perform similarly in layer 0, while discrepancies are observed in layer 8 for both models. However, CALOFFJORD is more accurate at the extreme $E_{ratio,8} = 0$. The average voxel energy is shown in Figure 3 for Dataset II. In the first layer of the detector, a similar pattern is observed in both

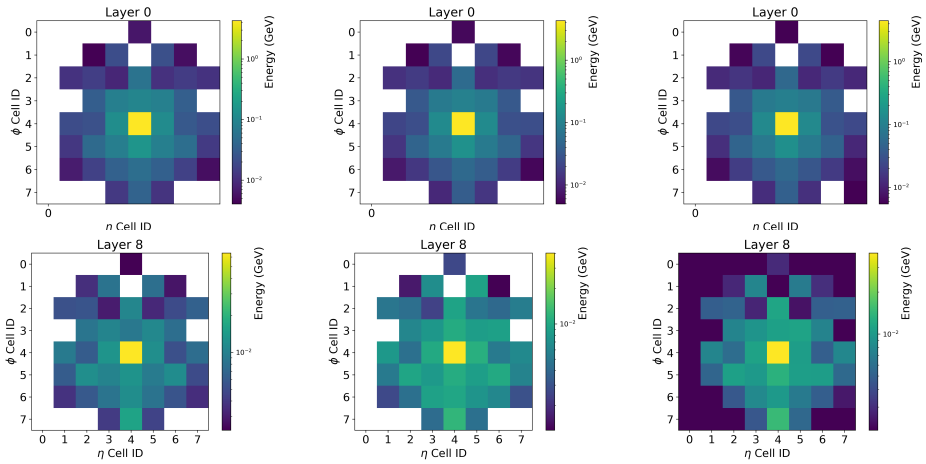

Figure 3: Average energy depositions in layers 0 (top row) and 8 (bottom row) in Dataset II. From left to right are GEANT, CALOFFJORD, and CALOFLOW.

Table 1: DNN classifier results for Dataset II

Models (vs GEANT)	AUC	JSD
CALOFFJORD	0.824 ± 0.003	0.256 ± 0.005
CALOFLOW	0.803 ± 0.003	0.224 ± 0.006

CALOFFJORD and CALOFLOW where the expected energy deposited is large. At the last layer of the detector, however, where low to no energy is deposited, we observe CALOFLOW producing noisier samples, resulting in additional spurious energy deposits which are not present in CALOFFJORD. When evaluating the classifier, CALOFLOW yields lower AUC and JSD metrics, averaged over 10 runs of the DNN classifier, as shown in Table 1. Nevertheless CALOFFJORD achieves similar performance in these metrics, but having sampling speeds over 10^3 times quicker according to Table 2[1].

Table 2: Sampling times and number of parameters for CALOFFJORD and CALOFLOW

	Time per shower		No. of Parameters	
Model	Dataset I	Dataset II	Dataset I	Dataset II
Full Simulation	1772 ms	$\mathcal{O}(10^5)$ ms	-	-
CALOFLOW	14.90 ms	16.75 ms	37.8M	43.1M
CALOFFJORD	4.72 μs	5.26 μs	6.5M	7.4M

4.1 Conclusions

In this work we introduced CALOFFJORD, a fast surrogate model for calorimeter generation using continuous normalizing flows. We train and evaluate the performance of our model using two public datasets. CALOFFJORD achieves similar performance to CALOFLOW with a distinct advantage in sampling speed and reduced number of model parameters. Since CALOFFJORD relies on continuous normalizing flows, the backbone network architecture is more flexible, allowing the use of network architectures that match the data structure. For future work, we plan to investigate other datasets and compare the proposed approach with other generative models such as flow matching (13), which uses a different strategy to train the continuous normalizing flow.

[1]An update to CALOFLOW introduces an additional distillation model that is able to speed up the generation time and make it competitive with CALOFFJORD. Since we focus on direct comparison with the baseline flow models, we have not investigated the distillation model

References

[1] Apostolakis, J. *et al.* HEP Software Foundation Community White Paper Working Group - Detector Simulation (2018). 1803.04165.

[2] Agostinelli, S. *et al.* Geant4 - a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506**, 250 – 303 (2003). URL http://www.sciencedirect.com/science/article/pii/S0168900203013688.

[3] Allison, J. *et al.* Geant4 developments and applications. *IEEE Transactions on Nuclear Science* **53**, 270–278 (2006).

[4] Allison, J. *et al.* Recent developments in geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **835**, 186–225 (2016). URL https://www.sciencedirect.com/science/article/pii/S0168900216306957.

[5] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I. & Duvenaud, D. FFJORD: free-form continuous dynamics for scalable reversible generative models. *CoRR* **abs/1810.01367** (2018). URL http://arxiv.org/abs/1810.01367. 1810.01367.

[6] Krause, C. & Shih, D. Fast and accurate simulations of calorimeter showers with normalizing flows. *Phys. Rev. D* **107**, 113003 (2023). 2106.05285.

[7] Hutchinson, M. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation* **19**, 433–450 (1990). URL https://doi.org/10.1080/03610919008812866.

[8] Paganini, M., de Oliveira, L. & Nachman, B. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev. D* **97**, 014021 (2018). 1712.10321.

[9] Fast calorimeter simulation challenge 2022. URL https://calochallenge.github.io/homepage/.

[10] Finlay, C., Jacobsen, J.-H., Nurbekyan, L. & Oberman, A. M. How to train your neural ode: The world of jacobian and kinetic regularization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20 (JMLR.org, 2020).

[11] Arnab, A. *et al.* Vivit: A video vision transformer. *CoRR* **abs/2103.15691** (2021). URL https://arxiv.org/abs/2103.15691. 2103.15691.

[12] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition (2015). 1512.03385.

[13] Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M. & Le, M. Flow matching for generative modeling (2022). 2210.02747.