

# CaloClouds II: Ultra-Fast Geometry-Independent Highly-Granular Calorimeter Simulation

Erik Buhmann<sup>1\*</sup>, Frank Gaede<sup>2,3</sup>, Gregor Kasieczka<sup>1,3</sup>,  
Anatolii Korol<sup>2</sup>, William Korcari<sup>1</sup>, Katja Krüger<sup>2</sup>,  
and Peter McKeown<sup>2</sup>

<sup>1</sup> Institut für Experimentalphysik, Universität Hamburg, Luruper  
Chaussee 149, 22761 Hamburg, Germany

<sup>2</sup> Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg,  
Germany

<sup>3</sup> Center for Data and Computing in Natural Sciences CDCS, Deutsches  
Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany

\* Author to whom correspondence should be addressed.

E-mail: [erik.buhmann@uni-hamburg.de](mailto:erik.buhmann@uni-hamburg.de)

August 2023

**Abstract.** Fast simulation of the energy depositions in high-granular detectors is needed for future collider experiments with ever increasing luminosities. Generative machine learning (ML) models have been shown to speed up and augment the traditional simulation chain in physics analysis. However, the majority of previous efforts were limited to models relying on fixed, regular detector readout geometries. A major advancement is the recently introduced CALOCLOUDS model, a geometry-independent diffusion model, which generates calorimeter showers as point clouds for the electromagnetic calorimeter of the envisioned International Large Detector (ILD).

In this work, we introduce CALOCLOUDS II which features a number of key improvements. This includes continuous time score-based modelling, which allows for a 25 step sampling with comparable fidelity to CALOCLOUDS while yielding a  $6\times$  speed-up over GEANT4 on a single CPU ( $5\times$  over CALOCLOUDS). We further distill the diffusion model into a consistency model allowing for accurate sampling in a single step and resulting in a  $46\times$  ( $37\times$ ) speed-up. This constitutes the first application of consistency distillation for the generation of calorimeter showers.

## 1. Introduction

Accurate simulations of particle physics experiments are crucial for comparing theory predictions with experimental results. With the planned high luminosity upgrade to the Large Hadron Collider (LHC) [1] and other envisioned collider experiments like those at the International Linear Collider (ILC) [2], experimental data is going to be taken at ever increasing rates. The amount of simulated events needs to keep up with these rates, which is difficult to achieve with current Monte Carlo simulations and the projected computing budgets at large experiments [3, 4].

Detector simulations, such as the simulation of the sensor response in highly granular calorimeters, can be augmented or sped up by employing modern generative machine learning methods [5–8]. Recent studies have explored the simulation of calorimeter showers with various generative models such as generative adversarial networks (GANs) [5, 9–19], autoencoders and their variants [20–25], and normalizing flows [26–33]. Additionally, diffusion models [34–38], also referred to as score-based generative models, have been shown to provide very high fidelity on calorimeter data [39–43]. Beyond detector simulation, generative models have, for example, also been explored as event generators [44–50] and parton shower simulators [51–63].

Most previous generative calorimeter models rely on a fixed data geometry, representing calorimeter showers as 3-dimensional images with the energy as the “color” channel and each pixel representing a calorimeter sensor. Modern high granularity calorimeters consist of many thousands of sensor cells or more (e.g. 6 million for the planned CMS HGCal [64]), but a given shower often deposits energy in only a small fraction of cells resulting in very sparse 3d image representations. Hence, it is much more computationally efficient to only simulate the actual energy depositions with a generative model. This can be achieved by describing the shower with only the coordinates and energies deposited — i.e. a *point cloud*. Such a multidimensional calorimeter point cloud can be represented by four features, the three-dimensional spatial coordinates and the cell energy, with the number of points equivalent to the number of cells containing hits.

In addition to computational efficiency, such point cloud showers have the major advantage that they can represent not only cell energies, but also much more granular GEANT4 step information, i.e. simulated energy depositions in the material, not accessible in experiments. Such GEANT4 step point clouds are largely independent of the cell structure within a layer of a given calorimeter, effectively allowing the projection of the shower into any part of the calorimeter, regardless

of cell type. These projections with GEANT4 step point clouds are less likely to produce artifacts due to gaps or cell staggering than cell-level point clouds would, resulting in a largely geometry-independent description of the calorimeter shower.

Previous point cloud and graph generative models explored in particle physics [55, 58–61, 63, 65] were only used for relatively small numbers of points. However, energetic calorimeter showers in high granularity calorimeters consist of  $\mathcal{O}(1000)$  points. To generate such showers, we recently introduced CALOCLOUDS [40] a generative model able to accurately generate photon showers in the form of point clouds with several thousands of points (namely clustered GEANT4 steps), in order to achieve geometry-independence. Since then, a specific comparison between a generative model for fixed geometry and a generative model for point cloud structured calorimeter showers on cell-level was performed in Ref. [41].

This CALOCLOUDS architecture consists of multiple sub-models with a diffusion model (see Sec. 3.1 for details) at its core. Most diffusion models, including the one used in CALOCLOUDS, are currently held back by their slow sampling speed, as many evaluation steps have to be performed to generate events. However, recent advances in computer vision achieve very high generative fidelity on natural image data with  $\mathcal{O}(10)$  model evaluations using advanced training paradigms and novel ordinary and stochastic differential equation solvers [66–69]. In this work, we first leverage recent advances in the training and sampling procedure of diffusion models in order to generate samples with the CALOCLOUDS II model<sup>‡</sup> using much fewer model evaluations than the original CALOCLOUDS model, by following the diffusion paradigm introduced in Ref. [67].

Another research direction to speed up generative models is the distillation of diffusion models into models which require significantly fewer function evaluations during sampling than the original model [70–74]. Recently, consistency models have been introduced as a novel kind of generative model allowing for single and multi-step data generation [75]. These consistency models can either be trained ab-initio or distilled from an already trained diffusion model. We demonstrate the ability to distill our diffusion model into a consistency model, thereby allowing data generation with a single model evaluation leading to further speed-ups.

<sup>‡</sup> The code is available at <https://github.com/FLC-QU-hep/CaloClouds-2>.

In summary, the proposed CALOCLOUDS II contains the following adjustments:

- (i) The previously used discrete-time diffusion process is replaced with the continuous-time diffusion paradigm introduced in Ref. [67]. This allows for fewer diffusion iterations during sampling.
- (ii) The common latent space is removed as we have noticed no advantage for the generative fidelity when generating photon calorimeter showers. This removal yields a simplified model architecture and improved training and sampling speeds.
- (iii) We add a calibration to the energy per calorimeter layer as well as applying a calibration to the center of gravity in the  $X$ - and  $Y$ -direction of the generated point cloud showers. This replaces the previous total energy calibration and improves the generative fidelity in the longitudinal energy distribution.
- (iv) Further, we apply consistency distillation to distill the diffusion model into a consistency model [75], allowing single step generation and therefore greatly improved sampling speed. We refer to this model as CALOCLOUDS II (CM).

In Sec. 2 we describe the point cloud dataset used for training and evaluation. The diffusion paradigm and model components of the CALOCLOUDS II model are explained in Sec. 3. We compare the generative fidelity of CALOCLOUDS II and its variant to the original CALOCLOUDS model in Sec. 4 and draw our conclusions in Sec. 5.

## 2. Data Samples

To compare the performance of our improved CALOCLOUDS II model we use the same dataset as in Ref. [40]. The data describes a calorimeter shower in the form of a point cloud. Each calorimeter shower consists of energy depositions of photons showering in a section of the high-granular electromagnetic calorimeter (ECAL) of the envisioned International Large Detector (ILD) [76]. As a sampling calorimeter, it consists of 30 layers with passive tungsten material and active silicon sensors. All individual silicon layers consist of small  $5 \times 5$  mm readout cells with a thickness of 0.5 mm. Between the first 20 active layers in the longitudinal direction there are passive layers with a thickness of 2.1 mm and between the remaining 10 layers the passive layers have a thickness of 4.2 mm. We simulated the dataset with GEANT4 Version 10.4 (using the QGSP\_BERT physics list) implemented in the iLCSoft framework [77]. The simulated geometric model is implemented

in DD4HEP [78] and includes realistic gaps between the sensors and position dependent irregularities. More simulation details can be found in Ref. [40].

During the full GEANT4 simulation up to 40,000 individual energy depositions originating from secondary particles traversing the active sensor material are registered (depending on the incident photon energy). We refer to these energy depositions as GEANT4 *steps*. All steps that fall into the volume of the same sensor are subsequently summed, resulting in the energy deposited in a cell *hit*. These cell hits (up to 1,500 at 90 GeV) are then used for downstream analysis as it is the same low-level information that is measurable in a real experimental setting.

Ideally, a generative model should produce cell-level hits to make the full GEANT4 simulation more computationally efficient. Cell-level information is also generated in all other approaches for fast calorimeter shower simulation with generative machine learning models. However, generating discrete cell hits directly in the form of a point cloud is challenging, as minor imperfections such as overlapping points can heavily impact the generative fidelity in various high level observables like the total number of hits  $N_{hits}$ .

Therefore, it could be advantageous to generate point clouds not on hit-level but on GEANT4 step-level, i.e. many simulated very granular energy depositions per cell, resulting in much larger point clouds where points are continuously distributed in space (as opposed to discrete cell hits). Yet, we found generating a point cloud with up to 40,000 steps prohibitively expensive from a computational point-of-view. Additionally such a high resolution is not necessary for good generative fidelity. Therefore in Ref. [40] we introduced a middle ground: we cluster the up to 40,000 GEANT4 steps into up to 6,000 *points*. For this clustering, the steps are grouped into their layer and their energy is binned in an ultra-high granularity grid with  $36\times$  higher granularity than the cell resolution, resulting in a square grid size of  $0.83 \times 0.83$  mm. This results in a clustered point cloud of up to 6,000 points — sufficiently small to be generated with the CALOCLOUDS model, yet distributed in discrete positions with sufficiently small separation so as to be approximately a continuous point distribution in 3d space.

In addition to a computationally efficient simulation, this makes the generated calorimeter point cloud largely geometry-independent of the actual cell layout of the calorimeter, unlike point clouds based on cell-level energy depositions. This ultra-high granularity calorimeter point cloud can be projected into any part of the calorimeter (except changing its depth), without introducing reconstruction artifacts due to i.e. gaps and cell staggering, as successfully shown in Ref. [40].

To produce the training set, a total of 524,000 showers were generated

with GEANT4, with an incident energy uniformly sampled between 10 and 90 GeV. Additionally, multiple test sets were generated: 40,000 showers uniformly distributed in energy for the figures shown in Sec. 4.1; 2,000 showers for the single energy plots at 10, 50, and 90 GeV; and 500,000 showers for calculating the evaluation metrics and the classifier score in Sec. 4.2 and Sec. 4.3.

Each point of the point cloud has four features: the  $X$ - and  $Y$ -position (transverse to the incident particle direction), the  $Z$ -position (parallel to the incident particle direction), and the *energy*. As a pre-processing step, the passive material regions are removed such that the point locations in the dataset also become continuous in the longitudinal  $Z$ -axis. The position features,  $X$ ,  $Y$ , and  $Z$ , are each normalized to the range  $[-1, 1]$ . The energy feature of the 4d point cloud is given in MeV.

As it is important for downstream analyses to accurately simulate the behaviour of photon showers on the level of the physical geometry, i.e. at cell level, all results shown in Sec. 4.1 to 4.3 are on cell-level. To this end, the calorimeter point cloud — with either up to 40,000 points for GEANT4 or with up to 6,000 points for those generated with CALOCLOUDS/ CALOCLOUDS II— are binned to the realistic ILD ECAL layout (including detector irregularities and gaps) with  $30 \times 30 \times 30$  calorimeter cells.

### 3. Generative Model

The CALOCLOUDS II model is an improved version of the original CALOCLOUDS architecture from Ref. [40]. CALOCLOUDS is a combination of two normalizing flows [79], a VAE-like encoder [80], and a discrete time Denoising Diffusion Probabilistic Model (DDPM) [37]. Specifically, it consists of the *Shower Flow*, a normalizing flow generating conditioning and calibration features; the *EPiC Encoder*, an encoder based on Equivariant Point Cloud (EPiC) layers [59] to encode calorimeter showers during training into a latent space for model conditioning; the *Latent Flow*, a normalizing flow trained to model the encoded latent space during sampling; and a diffusion model, called *PointWise Net*, which is a DDPM-based diffusion model generating each point i.i.d. based on a common latent space, incident energy and number of points conditioning. The models are implemented using PYTORCH 1.13 [81].

In the following, we outline the differences between CALOCLOUDS and CALOCLOUDS II. The largest conceptual difference is the change of the diffusion paradigm. We move from a discrete time diffusion process (DDPM), in which the

training and sampling is performed with the same number of diffusion steps, to a continuous time diffusion paradigm based on Ref [67], sometimes referred to as *EDM* diffusion or *k-diffusion*. This EDM diffusion allows for training a continuous time score function, which can be used to denoise any noise level, thereby separating the training and sampling procedure and allowing for sampling with various ordinary differential equation (ODE) and stochastic differential equation (SDE) solvers and different step sizes. Crucially, it allows to trade off sampling speed and sampling fidelity without retraining. We find good performance with the 2nd-order Heun ODE solver and the step size parameterisation suggested in Ref [67]. Additional details on the diffusion paradigm is given in the following Sec. 3.1.

As a second change, CALOCLOUDS II simplifies the original model. We noticed that for the photon calorimeter shower point clouds we are generating in this study, the shared latent space between points is not necessary for high generative fidelity. Therefore the latent dimensionality can be set to zero, so the *EPiC Encoder* and the *Latent Flow* are removed. By discarding it we achieve a simpler model as well as improved training and sampling efficiency.

Next, the *Shower Flow* for generating conditioning and calibration features is expanded to generate the total number of points, total visible energy, the relative number of points and energy of each calorimeter layer in the *Z*-direction, as well as the center of gravity in the *X*- and *Y*-direction. This flow is conditioned on the incident particle energy only. Similar to calibrating the number of points per calorimeter layer (see Ref. [40] for details), we now also calibrate the energy per layer of the generated point cloud as well as the center of gravity in the *X*- and *Y*-direction, replacing the previously performed post-diffusion calibration of the total visible energy. The total number of points generated per shower is used — together with the incident particle energy — for the conditioning of the PointWise Net diffusion model.

Overall, the Shower Flow is composed of ten blocks, each with seven coupling layers [82, 83] conditioned on the incident particle energy. It is implemented using the PYRO package [84]. The Shower Flow is trained once for 350k iterations and used for all three models (CALOCLOUDS, CALOCLOUDS II, and CALOCLOUDS II (CM)) compared in Sec. 4.

During sampling, the number of points predicted by the Shower Flow is calibrated before being used for the conditioning of the Latent Flow and the PointWise Net. The calibrated number of points is given by  $N_{\text{cal}} = N_{\text{uncal}} \cdot p_{\text{gen}}(p_{\text{data}}(N_{\text{uncal}}))$ , where  $p_{\text{data}}$  is a cubic polynomial fit of the ratio of the number

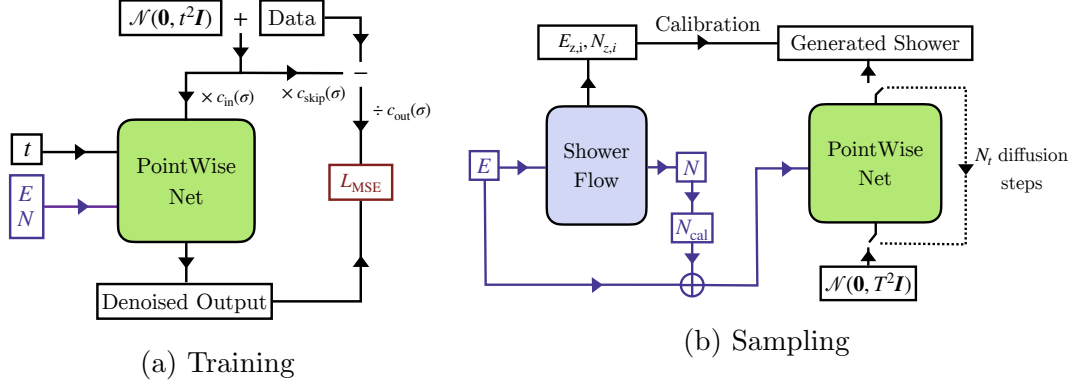


Figure 1: Illustration of the training and sampling procedure of the CALOCLOUDS II model. When sampling with CALOCLOUDS II (CM) only one denoising step is performed.

of points  $N_{\text{uncal}, \text{G4}}$  to the number of cell hits  $N_{\text{cell}, \text{G4}}$  of the GEANT4 showers and  $p_{\text{gen}}$  is a fit of the ratio of number of cell hits  $N_{\text{cell}, \text{gen}}$  to the (uncalibrated) number of points  $N_{\text{uncal}, \text{gen}}$  of a given model. Hence, this polynomial fit  $p_{\text{gen}}$  is performed for each model separately. More details on the model components and the calibrations can be found in Ref [40]. A schematic overview of the training and sampling procedure is shown in Fig. 1.

In the following Sec. 3.1 we describe the continuous time diffusion paradigm implemented in the CALOCLOUDS II model and in Sec. 3.2 we outline its distillation into a consistency model, referred to as CALOCLOUDS II (CM). Both models use the same model components outlined above. Details on the training and sampling hyperparameters are outlined in Sec. 3.3.

### 3.1. Diffusion Model

The diffusion model [34] used in the CALOCLOUDS model is a Denoising Diffusion Probabilistic Model (DDPM) with the same discrete time steps during model training and sampling [37, 85]. Since the introduction of DDPM, subsequent works, i.e. Refs. [38, 67, 86], have shown that it is advantageous to train a diffusion model with continuous time conditioning. This allows for a more flexible sampling regime for which various SDE and ODE solvers with either a fixed or an adaptive number of solving steps can be applied.

In the following, we outline the key parts of a diffusion model based on the paradigm outlined in Ref. [67]. The training of a diffusion model starts by diffusing



a data distribution  $p_{\text{data}}(x)$  with an SDE [38] in the forward direction (“data”  $\rightarrow$  “noise”) via

$$d\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t, t) dt + \sigma(t) d\mathbf{w}_t, \quad (1)$$

where  $t \in [0, T]$ ,  $T > 0$  is a fixed time step,  $\boldsymbol{\mu}(\cdot, \cdot)$  and  $\sigma(\cdot)$  denote the *drift* and *diffusion* coefficients, and  $\mathbf{w}_{t \in [0, T]}$  is the standard Brownian motion. The distribution of  $\mathbf{x}_t \sim p_t(\mathbf{x}) = p_{\text{data}}(\mathbf{x}) * \mathcal{N}(\mathbf{0}, T^2 \mathbf{I})$  ( $*$  as the convolution operator) and at time step  $t = 0$  it is identical to the data distribution  $p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$ . When reversing this diffusion process (“noise”  $\rightarrow$  “data”), a so called *probability flow ODE* emerges with a solution trajectory sampled at time step  $t$  given by

$$d\mathbf{x}_t = \left[ \boldsymbol{\mu}(\mathbf{x}_t, t) - \frac{1}{2} \sigma(t)^2 \nabla \log p_t(\mathbf{x}_t) \right] dt, \quad (2)$$

with  $\nabla \log p_t(\mathbf{x}_t)$  as the *score function* of  $p_t(\mathbf{x})$ . As suggested in Ref. [67], we set the coefficients in the SDE in Eq. 1 to  $\boldsymbol{\mu}(\mathbf{x}, t) = 0$  and  $\sigma(t) = \sqrt{2t}$  to ensure that  $p_T(\mathbf{x})$  is close to the distribution  $\mathcal{N}(\mathbf{0}, T^2 \mathbf{I})$ . Since the exact analytical score function is usually unknown, we train a neural network with weights  $\phi$  as a score model  $\mathbf{s}_\phi(\mathbf{x}, t) \approx \nabla \log p_t(\mathbf{x}_t)$  to get the empirical probability flow ODE:

$$\frac{d\mathbf{x}_t}{dt} = -t \mathbf{s}_\phi(\mathbf{x}_t, t) \quad (3)$$

For the purpose of numerically stable scaling behaviour, we follow Ref. [67] and actually train a separate network  $\mathbf{d}_\phi$  with  $t$ -dependent skip connections from which  $\mathbf{s}_\phi$  is derived:

$$\mathbf{s}_\theta(\mathbf{x}, t) = c_{\text{skip}}(t) \mathbf{x} + c_{\text{out}}(t) \mathbf{d}_\theta(c_{\text{in}}(t) \mathbf{x}, t) \quad (4)$$

The coefficients are time dependent and control the skip connection via  $c_{\text{skip}} = \sigma_{\text{data}}^2 / (\sigma_{\text{data}}^2 + t^2)$ , the input scaling via  $c_{\text{in}} = t \cdot \sigma_{\text{data}} / \sqrt{\sigma_{\text{data}}^2 + t^2}$  and the output scaling via  $c_{\text{out}} = 1 / \sqrt{\sigma_{\text{data}}^2 + t^2}$ . The hyperparameter  $\sigma_{\text{data}}$  corresponds roughly to the variance of  $p_{\text{data}}(\mathbf{x})$  and is set to  $\sigma_{\text{data}} = 0.5$ . During training a random time step is drawn from the continuous noise distribution  $\ln(t) = \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$ , with  $P_{\text{mean}} = -1.2$  and  $P_{\text{std}} = 1.2$ , and the loss is given by:

$$\mathbb{E}_{t, \mathbf{x}_t, \mathbf{x}_0} [\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2] \quad (5)$$

An illustration of this training process can be found in Fig. 1a.

For sampling from the trained score model, one samples from noise at time step  $T$  as  $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, T^2 \mathbf{I})$  and integrates the probability flow ODE in Eq. 2 over

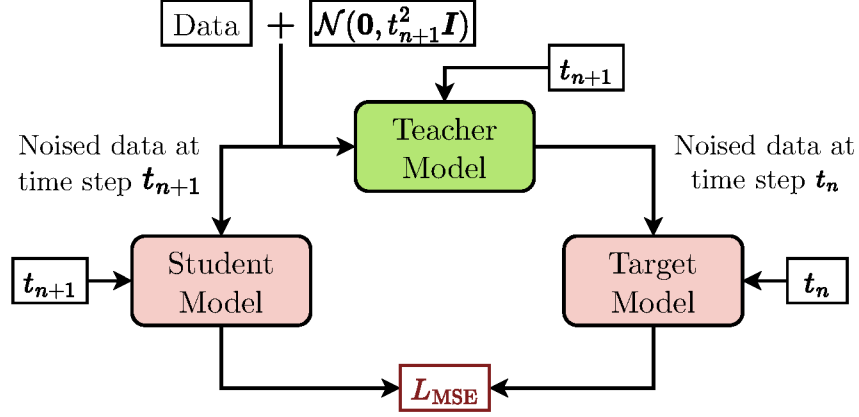


Figure 2: Illustration of the consistency distillation process distilling the diffusion model of CALOCLOUDS II (teacher model) into a consistency model (student and target model). The student model is updated via gradient descent and the target model is updated as an exponential moving average of the student model weights.

discrete time steps backwards in time using a numerical ODE solver. This results in a sample  $\hat{\mathbf{x}}_0$  which provides a good approximation of a sample from the data distribution  $p_{\text{data}}(\mathbf{x})$ . In practice the solver is usually stopped at a small positive value  $\epsilon > 0$  to avoid numerical instabilities resulting in the approximate sample  $\hat{\mathbf{x}}_\epsilon \approx \hat{\mathbf{x}}_0$ . For our sampling, we use the suggested values and step scheduling from Ref. [67] with  $T = 80$  and  $\epsilon = 0.002$ , and apply the 2<sup>nd</sup> order Heun ODE solver.

### 3.2. Consistency Model

Consistency Models (CM) [75] are a recently introduced generative architecture. They allow for single-step or multi-step generation with the same model and can be trained standalone or distilled from a diffusion model that has already been trained. A consistency model  $\mathbf{f}_\Phi$  with weights  $\Phi$  is trained to estimate the consistency function  $\mathbf{f}$  from data. The consistency function is defined as  $\mathbf{f} : (\mathbf{x}_t, t) \rightarrow \mathbf{x}_\epsilon$  and is *self-consistent* in the sense that any pair of  $(\mathbf{x}_t, t)$  belong to the same probability flow ODE trajectory. This means that the result of a function evaluation at any point on this trajectory leads to the same result, i.e.  $\mathbf{f}(\mathbf{x}_t, t) = \mathbf{f}(\mathbf{x}'_{t'}, t')$  for all  $t, t' \in [\epsilon, T]$ .

For sampling from a trained consistency model in a single model pass,

one initializes  $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, T^2 \mathbf{I})$  and performs one function evaluation to get  $\hat{\mathbf{x}}_\epsilon = \mathbf{f}_\Phi(\mathbf{x}_T, T)$ . It is also possible to sample with multiple model passes by first evaluating  $\mathbf{f}_\Phi(\mathbf{x}_T, T)$ , and then adding noise again from  $\mathcal{N}(\mathbf{0}, t^2 \mathbf{I})$  to denoise a second time. This can be done in an alternating fashion for an arbitrary number of steps. Often multi-step generation appears to improve sample fidelity [63, 67], however we are able to achieve comparable fidelity to the original diffusion model with only a single model evaluation and therefore limit ourselves to this most efficient scenario.

In line with Ref. [75], we found improved training fidelity when distilling the consistency model from a diffusion model instead of training it individually. For this purpose we distill the consistency model  $\mathbf{f}_\Phi(\mathbf{x}, t)$  from the diffusion model  $\mathbf{s}_\phi(\mathbf{x}, t)$  based on the PointWise Net of CALOCLOUDS II introduced in the previous Sec. 3.1. The distillation is performed by separating the continuous time space  $[\epsilon, T]$  into  $N - 1$  sub intervals (we use  $N = 18$ ). The interval boundaries are determined by the same step size parameterisation as in the diffusion model sampling formulation [67]. During training a random boundary time step  $t_{n \in [1, N]}$  is chosen to perform the distillation. We refer to the original diffusion model here as the *teacher* model  $\mathbf{s}_\phi(\mathbf{x}, t)$  and to the distilled consistency model during distillation as the *student* model  $\mathbf{f}_\Phi(\mathbf{x}, t)$ . Additionally, we call the final distilled consistency model the *target* model  $\mathbf{f}_{\Phi^-}(\mathbf{x}, t)$ . We use the self-consistency property of the consistency model for training since it requires a well trained model to obey  $\mathbf{f}_\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}) = \mathbf{f}_\Phi(\mathbf{x}_{t_n}, t_n)$ . These neighboring points  $(\mathbf{x}_{t_{n+1}}, \mathbf{x}_{t_n})$  on the probability flow ODE trajectory we obtain by sampling  $\mathbf{x} \sim p_{\text{data}}$ , adding noise to it to get  $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}, t_{n+1}^2 \mathbf{I})$  and performing one ODE solver step with the teacher diffusion model to compute  $\mathbf{x}_{t_n} = \mathbf{s}_\phi(\mathbf{x}_{t_{n+1}}, t_{n+1})$ . This allows the student consistency model  $\mathbf{f}_\Phi(\mathbf{x}, t)$  to be trained with the loss:

$$\mathbb{E}_{t, \mathbf{x}_t, \mathbf{x}_0} [\|\mathbf{f}_\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}) - \mathbf{f}_{\Phi^-}(\mathbf{x}_{t_n}, t_n)\|_2^2] \quad (6)$$

The target model  $\mathbf{f}_{\Phi^-}(\mathbf{x}_t, t)$  weights  $\Phi^-$  are updated after every iteration as a running average of the student model weights  $\Phi$ . An overview of the distillation procedure is illustrated in Fig. 2.

### 3.3. Training and Sampling

The diffusion model in CALOCLOUDS II was trained for 2M iterations with a batch size of 128 using the Adam optimizer [87] with a fixed learning rate of  $10^{-4}$ . As the final model, we use an exponential moving average (EMA) of the model

weights. We scan several values for the number of ODE solver steps  $N$  and find  $N = 13$  optimal as with fewer steps than this, the generative fidelity as probed by the correct learning of physically relevant shower shapes with CALOCLOUDS II deteriorates. This results in  $2N - 1$  diffusion model evaluations as the last step of the Heun ODE solver does not perform a 2nd order correction. Compared to CALOCLOUDS with 100 function evaluations this already hints at a significant computational speed-up.

The diffusion model used in CALOCLOUDS II was distilled into a consistency model for CALOCLOUDS II (CM) by using the Adam optimizer with a fixed learning rate of  $10^{-4}$  for 1M iterations with a batch size of 256. Notably, only a single training is necessary for distilling a model which is able to perform single step generation, as opposed to the multiple trainings required for e.g. progressive distillation [42, 65, 71].

## 4. Results

In the following, we compare the original CALOCLOUDS model with the improved CALOCLOUDS II model and its distilled variant CALOCLOUDS II (CM). To achieve a fair comparison between the three models, we use the same training of the Shower Flow and the same calibration procedure for all three models. Hence, the Shower Flow from the CALOCLOUDS II model was also used for generating samples with the CALOCLOUDS model — a slight modification compared to the original CALOCLOUDS model in Ref. [40]. This means that the samples generated with the CALOCLOUDS model also include the energy per layer and center of gravity in  $X$  and  $Y$  calibration. For the Latent Flow and the PointWise Net of CALOCLOUDS the same model weights as in Ref. [40] were used.

We first show the performance of our generative models based on the same observables as discussed in Ref. [40] in Sec. 4.1. Next, in Sec. 4.2, we quantify the performance of the models with multiple Wasserstein-distance-based scores for the usual set of calorimeter shower observables and in Sec. 4.3 we use a classifier to distinguish between simulated GEANT4 showers and generated showers based on the calculated shower observables. Finally in Sec. 4.4 we benchmark the computational efficiency of our models and compare them to the baseline simulation timing with GEANT4.

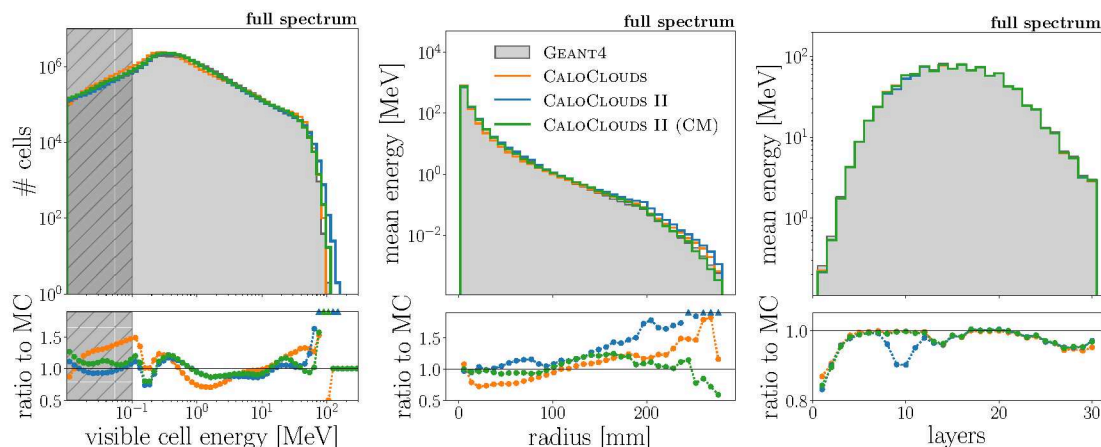


Figure 3: Histogram of the cell energies (left), radial shower profile (center), and longitudinal shower profile (right) for GEANT4, CALOCLOUDS, CALOCLOUDS II, and CALOCLOUDS II (CM). In the cell energy distribution, the region below 0.1 MeV is grayed out (see main text for details). All distributions are calculated with 40,000 events sampled with a uniform distribution of incident particle energies between 10 and 90 GeV. The bottom panel provides the ratio to GEANT4. Values outside the range are indicated by small triangles.

#### 4.1. Physics Performance

In this Section, we compare various calorimeter shower distributions from Ref. [40] between the GEANT4 test set and datasets generated using CALOCLOUDS, CALOCLOUDS II, and CALOCLOUDS II (CM). First, we compare various cell-level and shower observables calculated from the model generated showers to GEANT4 simulations with samples of incident photons with energies uniformly distributed between 10 and 90 GeV (also referred to as *full spectrum*). In Fig. 3 we investigate three representations of the energy distributed in the calorimeter cells, namely the per-cell energy distribution (left), the radial shower profile (center) and the longitudinal shower profile (right). The per-cell energy distribution contains the energy of the cells of all showers in the test dataset. The peak of the distribution at about 0.2 MeV corresponds to the most probable energy deposition of a minimum ionising particle (MIP) in the silicon sensor. For downstream analyses a cell energy cut at half a MIP is applied, since below this threshold the sensor response is indistinguishable from electronic noise. Hence this cut was applied to all showers when calculating the shower observables and scores in this section. All models

describe the cell energy distribution reasonably well. For most of the range the CALOCLOUDS II models perform better than CALOCLOUDS, however there are a few outliers with energies which are too high produced by CALOCLOUDS II compared to the other two models.

The radial shower profile describes the average radial energy distribution around the central shower axis (in Z-direction) of the ECAL. It is well described by all models, although overall the CALOCLOUDS II (CM) model represents the GEANT4 distribution most closely. Note that this is a distribution that is not directly impacted by any of the post-diffusion calibrations performed and is therefore a good benchmark for the effectiveness of the point cloud diffusion approach alone.

The longitudinal shower profile describes how much energy is deposited on average in each of the 30 calorimeter layers. In the previous iteration of CALOCLOUDS it was not well modeled, but since we now calibrate the energy per layer with the improved Shower Flow for the generated point clouds it is very well modelled. However, a small outlier can be seen for the CALOCLOUDS II model around the 10<sup>th</sup> layer. The alternating higher and lower energy depositions per layer are due to the fact that for technical reasons, pairs of silicon sensors surrounding one tungsten absorber layer and facing opposite directions are installed into a tungsten structure with every other absorber layer. This results in the observed pair-wise difference in the sampling fraction between consecutive layers.

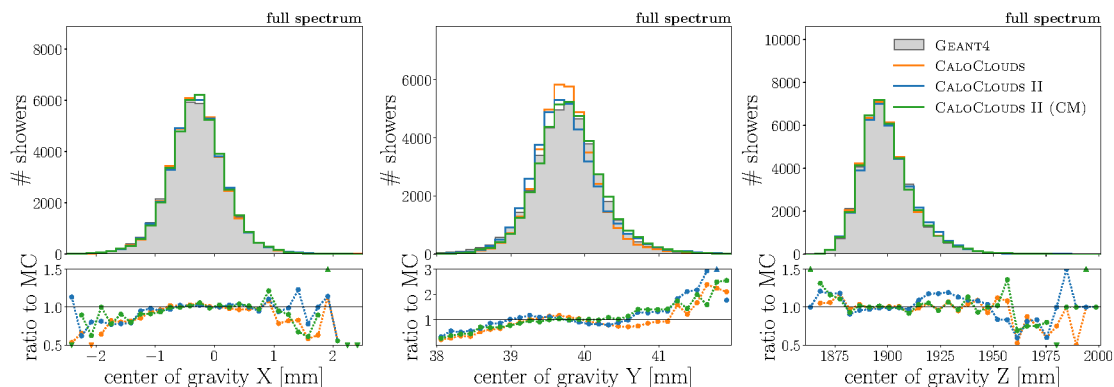


Figure 4: Position of the center of gravity of showers along the X (left), Y (center), and Z (right) directions. All distributions are calculated for 40,000 showers with a uniform distribution of incident particle energies between 10 and 90 GeV.

In Fig. 4 we show the center of gravity distribution  $m_{1,i \in \{X,Y,Z\}}$  (the energy weighted shower centroid) in the  $X$ -,  $Y$ -, and  $Z$ -directions. Note that in the  $X$ - and  $Y$ -directions these distribution are calibrated for the original point cloud, before the cell-level observables are calculated. While the  $m_{1,X}$  distribution is very well modelled by all three generative models,  $m_{1,Y}$  is slightly shifted to lower center of gravity values for all models with the CALOCLOUDS distribution additionally being marginally too narrow. The centers of gravity in  $X$  and  $Y$  behave slightly different as a magnetic field is simulated in the  $Y$ -direction and the active sensors are staggered in the  $Y$ -direction while they are all aligned in the  $X$ -direction. Due to the number of hits and energy per layer calibrations applied, the distribution of  $m_{1,Z}$  is very well modelled. Only in the region around 1925 mm is the CALOCLOUDS II model slightly worse than the other two models. Overall, the three models are reasonably close to the GEANT4 simulation in all six observables.

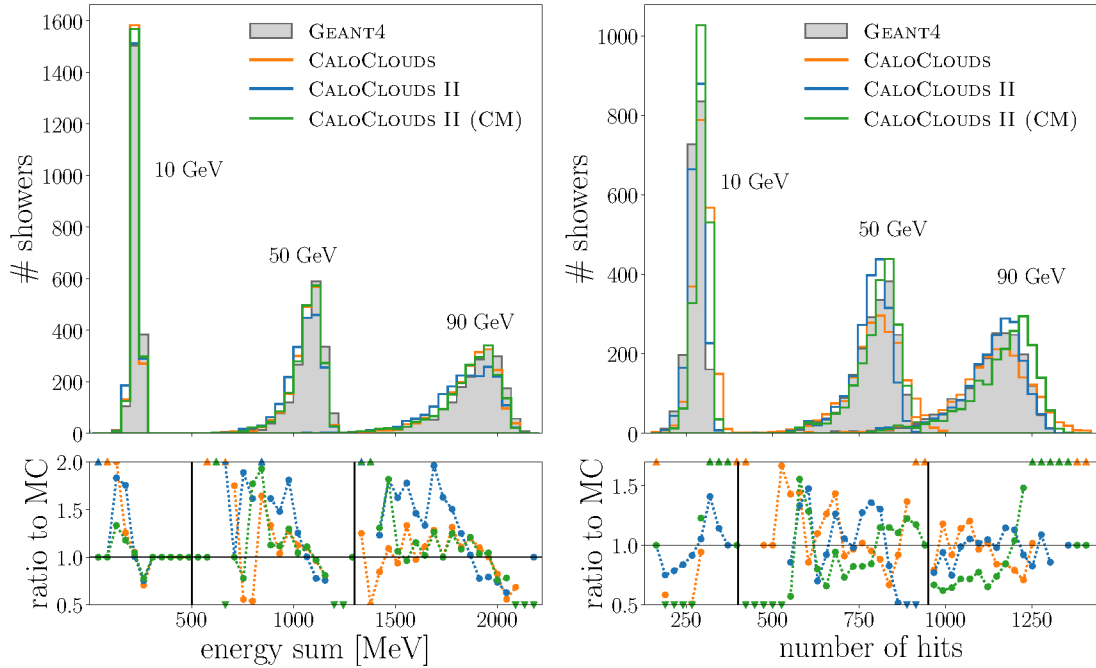


Figure 5: Visible energy sum (left) and the number of hits (right) distributions, for 10, 50, and 90 GeV showers. For each energy and model, 2,000 showers are shown.

Next, we investigate the models’ fidelity for single incident photon energies of 10, 50, and 90 GeV. In Fig. 5 we show the distributions of the total visible energy (left) and the distributions of the number of hits (cells with deposited energy above the half MIP threshold) for the three single energy datasets of 2k showers each. The total energy is well represented by all three models, with a few outliers visible in the ratio between the generated output and the GEANT4 simulation. The number of hits on the other hand is one of the most difficult distributions to represent well with a point cloud generative model. Here high fidelity is still achieved by applying the number of points re-scaling discussed in Sec. 3. Overall the CALOCLOUDS distributions are slightly too wide as was observed already in Ref. [40]. In comparison, CALOCLOUDS II represents the shape of the distribution better, yet in particular for 10 and 90 GeV showers the mean is a bit too large for the CALOCLOUDS II (CM) generated events. This is explainable due to the nature of the polynomial fit used for the number of points re-scaling. The fit does not perform very well at the edges of the incident energy space. It is known that extrapolation is difficult for generative models, therefore we conjecture that with a training set including lower and higher energies, the fidelity at 10 and 90 GeV would approach the performance at 50 GeV. Overall the CALOCLOUDS II models perform very well and are comparable in fidelity to the CALOCLOUDS model.

#### 4.2. Evaluation Scores

| Simulator          | $W_1^{N_{\text{hits}}} (\times 10^{-3})$ | $W_1^{E_{\text{vis}}/E_{\text{inc}}} (\times 10^{-3})$ | $W_1^{E_{\text{cell}}} (\times 10^{-3})$ | $W_1^{E_{\text{long}}} (\times 10^{-3})$ | $W_1^{E_{\text{radial}}} (\times 10^{-3})$ | $W_1^{m_{1,x}} (\times 10^{-3})$ | $W_1^{m_{1,y}} (\times 10^{-3})$ | $W_1^{m_{1,z}} (\times 10^{-3})$ |
|--------------------|------------------------------------------|--------------------------------------------------------|------------------------------------------|------------------------------------------|--------------------------------------------|----------------------------------|----------------------------------|----------------------------------|
| GEANT4             | $0.7 \pm 0.2$                            | $0.8 \pm 0.2$                                          | $0.9 \pm 0.4$                            | $0.7 \pm 0.8$                            | $0.7 \pm 0.1$                              | $0.9 \pm 0.1$                    | $1.1 \pm 0.3$                    | $0.9 \pm 0.3$                    |
| CALOCLOUDS         | <b><math>2.5 \pm 0.3</math></b>          | $11.4 \pm 0.4$                                         | $15.9 \pm 0.7$                           | <b><math>2.0 \pm 1.3</math></b>          | $38.8 \pm 1.4$                             | $4.0 \pm 0.4$                    | $8.7 \pm 0.3$                    | $1.4 \pm 0.5$                    |
| CALOCLOUDS II      | $3.6 \pm 0.5$                            | $26.4 \pm 0.4$                                         | <b><math>15.3 \pm 0.6</math></b>         | $3.7 \pm 1.6$                            | $11.6 \pm 1.5$                             | <b><math>2.4 \pm 0.4</math></b>  | <b><math>7.6 \pm 0.2</math></b>  | $3.9 \pm 0.4$                    |
| CALOCLOUDS II (CM) | $6.1 \pm 0.7$                            | <b><math>9.8 \pm 0.5</math></b>                        | $16.0 \pm 0.7$                           | <b><math>2.0 \pm 1.4</math></b>          | <b><math>8.3 \pm 1.9</math></b>            | $3.0 \pm 0.4$                    | $9.5 \pm 0.6$                    | <b><math>1.2 \pm 0.5</math></b>  |

Table 1: Model performance comparison with 1-Wasserstein distance based scores for various standardized shower observables. The values presented are the mean and standard deviation of 10 calculated scores comparing 50k GEANT4 and 50k generated showers.

We next investigate the performance of all three CALOCLOUDS models by calculating scores from the high level calorimeter shower observables considered in



the previous Section. This allows us to put a number on the fidelity observed in plots presented in the previous Sec. 4.2 and not only rely on comparing distributions by eye.

The following observables are considered in order to calculate the one-dimensional scores: The number of hits (cells with energy depositions above the half MIP threshold)  $N_{\text{hits}}$ , the sampling fraction (the ratio of the visible energy deposited in the calorimeter to the incident photon energy)  $E_{\text{vis}}/E_{\text{inc}}$ , the cell energy  $E_{\text{cell}}$ , the center of gravity in the  $X$ -,  $Y$ -, and  $Z$ -directions  $m_{1,i \in \{X,Y,Z\}}$ , and ten observables each for the longitudinal energy  $E_{\text{long},i \in [1,10]}$  and for the radial energy  $E_{\text{radial},i \in [1,10]}$ . The ten observables for the longitudinal (radial) energy depositions are computed with the energy clustered in consecutive layers (concentric regions) such that on average all 10 observables  $E_{\text{long},i \in [1,10]}$  and  $E_{\text{radial},i \in [1,10]}$  are computed with the same statistics. Further details on these in total 20 observables can be found in Appendix A.

To compare the distributions of these observables between GEANT4 and the three generative models, we calculate the 1-Wasserstein distance  $W_1$  — also known as the earth movers distance — between each pair of distributions. The advantages of the Wasserstein distance are that it is an unbinned estimator, for one-dimensional distributions it is computationally efficient to calculate, and no hyperparameter choices have to be made apart from the number of events used for comparison.

Following earlier works using Wasserstein distance based model evaluation scores to compare generative models [55, 58], we calculate the distance between observables calculated from 50k GEANT4 and 50k model generated showers. This is done  $10\times$  for independent uniformly distributed samples and we report the mean and standard deviation of the scores in Tab. 1. To have all scores in a similar order of magnitude, we standardize each observable before we calculate the  $W_1$  score. For the layer energy and radial energy scores,  $W_1^{E_{\text{long}}}$  and  $W_1^{E_{\text{radial}}}$ , we report the average Wasserstein distance over all ten bins. The hit energy score  $W_1^{E_{\text{cell}}}$  is calculated for 50k cell hits. In addition to the generative model scores, we also calculate the scores for GEANT4 itself, comparing 50k GEANT4 showers to a separate set of 50k GEANT4 showers.

As can be seen in Tab. 1, with a few exceptions all model scores are quite close together, often within the margin of error. Overall, CALOCLOUDS II (CM) appears to produce slightly higher fidelity showers than the other two models. In particular the radial energy distribution appears to be modelled worse by CALOCLOUDS and best modelled by CALOCLOUDS II (CM), which is in line with

the histogram shown in Fig. 3. However, as can also be seen in the histograms in Sec. 4.1, none of the scores — with the exception of  $W_1^{m_1, z}$  — quite reaches the fidelity of the GEANT4 truth itself. Hence we conclude that all models perform quite well already in generating high fidelity ECAL showers, but should be further improved to match GEANT4 more exactly in the future.

As a side note, the Wasserstein distance can be heavily impacted by outliers in the distributions. Therefore it does not always correlate well with the distribution shape observed in histograms. However, the scores complement the visual inspections of histograms and distributions shown in Sec. 4.1 well.

While useful for comparing generative architectures, 1-Wasserstein distances only consider each dimension of the problem individually. Of course, a successful generative model should also accurately describe higher order correlations. We investigate this in the next Section.

### 4.3. Classifier Scores

| Simulator          | AUC                                 |
|--------------------|-------------------------------------|
| CALOCLOUDS         | $0.999 \pm 0.001$                   |
| CALOCLOUDS II      | $0.928 \pm 0.001$                   |
| CALOCLOUDS II (CM) | <b><math>0.923 \pm 0.001</math></b> |

Table 2: Model performance comparison with area under the receiver operating characteristic curve (AUC) score.

We further compare the model generated showers to the GEANT4 simulation by training a fully connected high-level classifier using the shower observables discussed in the previous Sec. 4.2 to distinguish between model generated and GEANT4 simulated showers. The 25 input shower observables are the ten radial and longitudinal energy observables, as well as the three center of gravity variables and the number of hits and total visible energy. For the datasets, we use 500k GEANT4 showers and 500k showers generated by each generative model. A 80%, 10%, 10% data split is applied, resulting in a training set of 800k showers and a validation and test set with 100k showers each.

The classifier is implemented as a fully connected neural network with three

layers (containing 32, 16, 8 nodes respectively) with LeakyReLU [88] activation functions, and one output node with a Sigmoid activation. It is trained with the Adam optimizer [87] for 10 epochs for each dataset using a binary cross-entropy loss. The final model epoch is chosen based on the lowest validation loss.

To evaluate the classifier we use the area under the receiver operating characteristic curve (AUC) score calculated on the test set. This kind of *classifier score* is also used in other publications evaluating generative models in high energy physics such as Ref. [24, 27, 28, 30, 33, 58, 89]. In case the classifier can perfectly separate the GEANT4 and model generated datasets, it will result in an  $AUC = 1.0$ . For a generated dataset that is indistinguishable from GEANT4 simulation, we expect a confused classifier with an  $AUC = 0.5$ . Values in between are difficult to interpret in absolute terms, but can give a rough indication of how well the generative models are performing compared to each other.

We trained the classifier ten times with a different train/ test/ validation data split each time. In Tab. 2 we present the mean AUC and standard deviation of these ten classifier trainings. The CALOCLOUDS generated dataset performs the worst, leading to an almost perfect classification with  $AUC = 0.999$ . The two CALOCLOUDS II variants both have a better score clearly separated from an  $AUC = 1.0$ . With an  $AUC = 0.923$  the CALOCLOUDS II (CM) model performs slightly better than the CALOCLOUDS II model. For most events, both models result in clear separability from the GEANT4 simulated showers, but constitute a clear improvement over the baseline CALOCLOUDS implementation. The better performance of the CALOCLOUDS II variants is likely due to the improved radial energy distribution, as we observed a rather large deviation in the  $W_1^{E_{\text{radial}}}$  score and in the radial distributions in Fig. A1.

#### 4.4. Timing

In this Section, we benchmark the average time to produce a single calorimeter shower with the three models considered and investigate the speed-up over the baseline GEANT4 simulation. The timing results are presented in Tab. 3.

On both a single CPU and on an NVIDIA<sup>®</sup> A100 GPU we generated  $25 \times 2,000$  showers with the same uniform energy distribution between 10 and 90 GeV. We report the mean and standard deviation of generating these showers. In particular the timing on a single CPU is interesting for current applications of generative models in high energy physics, as CPUs are much more widely available than GPUs and the current computing infrastructure relies on simulations run on

| Hardware | Simulator          | NFE | Batch Size | Time / Shower [ms]  | Speed-up      |
|----------|--------------------|-----|------------|---------------------|---------------|
| CPU      | GEANT4             |     |            | $3914.80 \pm 74.09$ | $\times 1$    |
|          | CALOCLOUDS         | 100 | 1          | $3146.71 \pm 31.66$ | $\times 1.2$  |
|          | CALOCLOUDS II      | 25  | 1          | $651.68 \pm 4.21$   | $\times 6.0$  |
|          | CALOCLOUDS II (CM) | 1   | 1          | $84.35 \pm 0.22$    | $\times 46$   |
| GPU      | CALOCLOUDS         | 100 | 64         | $24.91 \pm 0.72$    | $\times 157$  |
|          | CALOCLOUDS II      | 25  | 64         | $6.12 \pm 0.13$     | $\times 640$  |
|          | CALOCLOUDS II (CM) | 1   | 64         | $2.09 \pm 0.13$     | $\times 1873$ |

Table 3: Comparison of the computational performance of CALOCLOUDS, CALOCLOUDS II, and CALOCLOUDS II (CM) to the baseline GEANT4 simulator on a single core of an Intel<sup>®</sup> Xeon<sup>®</sup> CPU E5-2640 v4 (CPU) and on an NVIDIA<sup>®</sup> A100 with 40 GB of memory (GPU). 2,000 showers were generated with incident energy uniformly distributed between 10 and 90 GeV. Values presented are the means and standard deviations over 10 runs. The number of function evaluations (NFE) indicate the number of diffusion model passes.

CPUs. Further, the single CPU timing facilitates a direct comparison to the GEANT4 simulation. Here CALOCLOUDS already yields a speed up of  $1.2\times$ , but with less sampling steps CALOCLOUDS II achieves a speed up of  $6.0\times$ . However, when implementing the consistency distillation, we achieve a speed up of  $46\times$  with the CALOCLOUDS II (CM) model even surpassing previous generative models on the same kind of dataset such as the BIB-AE [20] by about a factor 5.

On GPU the CALOCLOUDS model achieves a speed up of  $157\times$ , CALOCLOUDS II achieves  $640\times$ , and CALOCLOUDS II (CM) achieves  $1873\times$  speed up over the baseline GEANT4 simulation on a single CPU. Note that GEANT4 is currently not compatible with GPUs and that GPUs are significantly more expensive than CPUs.

For reference, the training of the CALOCLOUDS model on similar NVIDIA<sup>®</sup> A100 GPU hardware took around 80 hours for 800k iterations with a batch size of 128, while training of the CALOCLOUDS II model took around 50 hours for 2 million iterations with the same batch size. The consistency distillation for 1 million iterations with a batch size of 256 took about 100 hours.

The speed up between CALOCLOUDS and CALOCLOUDS II is the result of a combination of the improved diffusion paradigm requiring a reduced number of function evaluations as well as the removal of the latent flow. The speed up due to the consistency model in CALOCLOUDS II (CM) yields another large factor, since only a single model evaluation is performed. Both models would be slightly slower when applied in conjunction with the Latent Flow of the CALOCLOUDS model as one evaluation of the Latent Flow is about 50% slower than a single evaluation of the PointWise Net. For a large number of model passes of the PointWise Net in the diffusion framework, the efficiency of the Latent Flow is negligible. However when we consider CALOCLOUDS II (CM) with a single model pass, the application of the Latent Flow would have a noticeable impact on computational performance. Therefore, we removed the Latent Flow in favour of model efficiency as we did not see any improvement in generative fidelity when using it in the CALOCLOUDS II framework.

## 5. Conclusions

CALOCLOUDS was the first generative model to achieve high-fidelity highly-granular photon calorimeter showers in the form of point clouds with a number of points of  $\mathcal{O}(1,000)$ . Due to their sparsity, describing calorimeter showers as point clouds is computationally more efficient than describing them with fixed data structures, i.e. 3d images. Additionally, as the point clouds are based on clustered GEANT4 steps, they allow for a translation-invariant and geometry-independent shower representation. Such cell-geometry-independent models could be easily adapted for fast simulations of calorimeters with non-square cell geometries, i.e. hexagonal cells as used in the envisioned CMS HGCal [64].

With CALOCLOUDS II we introduce a more streamlined version of CALOCLOUDS utilizing the advanced diffusion paradigm from Ref. [67]. It allows for sampling with less model evaluations and for distillation into a consistency model. Using the consistency model in CALOCLOUDS II (CM), generation with a single model evaluation is possible and results in a greatly improved computational efficiency and a speed up of  $46\times$  over GEANT4 on a single CPU. This single event CPU performance is particularly promising for introducing a generative model into existing GEANT4-based simulation pipelines. As opposed to other diffusion distillation methods like progressive distillation, consistency distillation only requires a single training to distill the diffusion model in CALOCLOUDS II into a single step generative model, further emphasising the computational advantage of

the models presented here. To our knowledge, this constitutes the first application of a consistency model to calorimeter data.

We compare all three point cloud generative models using one-dimensional distributions and a classifier-based measure and find comparable performance with a slight advantage for the CALOCLOUDS II variants. In particular, the CALOCLOUDS II (CM) model exhibits superior performance while being significantly more computationally efficient. Yet, slight deviations from the GEANT4 simulations are still visible in various shower observables. Further improvements could likely be achieved by investigating more complex architectures for the diffusion model such as fast transformer implementations [90], equivariant point cloud (EPiC) layers [59], or cross-attention [91].

During the completion of this manuscript, another EDM diffusion based model with subsequent consistency distillation was shown to achieve good fidelity when generating particle jets in the form of point clouds with up to 150 points [63]. While technically similar approaches, in our case the consistency model does not lose generative fidelity compared to the diffusion model and we demonstrate the generation of two orders of magnitude more points (6000 vs 150).

In conclusion, the CALOCLOUDS II model generates high fidelity electromagnetic showers when benchmarked on various shower observables against the baseline GEANT4 simulation. In combination with consistency distillation the CALOCLOUDS II (CM) model yields an accurate simulator, which is two orders of magnitude faster than GEANT4 on identical hardware. This constitutes an important step towards the integration of point-cloud based generative models in actual simulation workflows.

## Acknowledgments

We thank Dirk Krücker for the valuable comments on the manuscript. This research was supported in part by the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany. This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement No 101004761. We acknowledge support by the Deutsche Forschungsgemeinschaft under Germany’s Excellence Strategy – EXC 2121 Quantum Universe – 390833306 and via the KISS consortium (05D23GU4, 13D22CH5) funded by the German Federal Ministry of Education and Research BMBF in the ErUM-Data action plan. E.B. is partially funded by a scholarship from the Friedrich Naumann Foundation for Freedom. A.K. has

received support from the Helmholtz Initiative and Networking Fund's initiative for refugees as a refugee of the war in Ukraine.

## References

- [1] I. Zurbano Fernandez *et al.* edited by I. Béjar Alonso, O. Brüning, P. Fessia, L. Rossi, L. Tavian and M. Zerlauth 2020 **High-Luminosity Large Hadron Collider (HL-LHC): Technical design report 10/2020** doi: 10.23731/CYRM-2020-0010
- [2] T. Behnke, J. E. Brau, B. Foster, J. Fuster, M. Harrison, J. M. Paterson, M. Peskin, M. Stanitzki, N. Walker and H. Yamamoto 2013 **The International Linear Collider Technical Design Report - Volume 1: Executive Summary** e-Print: 1306.6327 doi: 10.48550/arXiv.1306.6327
- [3] J. Albrecht *et al.* (HEP Software Foundation) 2019 **A Roadmap for HEP Software and Computing R&D for the 2020s** *Comput. Softw. Big Sci.* **3** 7. e-Print: 1712.06982 doi: 10.1007/s41781-018-0018-8
- [4] A. Boehnlein *et al.* 2022 **HL-LHC Software and Computing Review Panel Report** URL: <http://cds.cern.ch/record/2803119/>
- [5] M. Paganini, L. de Oliveira and B. Nachman 2018 **Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters** *Phys. Rev. Lett.* **120** 042003. e-Print: 1705.02355 doi: 10.1103/PhysRevLett.120.042003
- [6] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn 2021 **GAN-  
plifying event samples** *SciPost Phys.* **10** 139. e-Print: 2008.06545 doi: 10.21468/SciPostPhys.10.6.139
- [7] S. Bieringer, A. Butter, S. Diefenbacher, E. Eren, F. Gaede, D. Hundhausen, G. Kasieczka, B. Nachman, T. Plehn and M. Trabs 2022 **Calomplification – The Power of Generative Calorimeter Models** *JINST* **17** P09028. e-Print: 2202.07352 doi: 10.1088/1748-0221/17/09/P09028
- [8] A. Adelmann *et al.* 2022 **New directions for surrogate models and differentiable programming for High Energy Physics detector simulation** *Snowmass 2021*. e-Print: 2203.08806
- [9] M. Paganini, L. de Oliveira and B. Nachman 2018 **CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks** *Phys. Rev. D* **97** 014021. e-Print: 1712.10321 doi: 10.1103/PhysRevD.97.014021
- [10] L. de Oliveira, M. Paganini and B. Nachman 2018 **Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters** *J. Phys. Conf. Ser.* **1085** 042017. e-Print: 1711.08813 doi: 10.1088/1742-6596/1085/4/042017
- [11] M. Erdmann, L. Geiger, J. Glombitza and D. Schmidt 2018 **Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks** *Comput. Softw. Big Sci.* **2** 4. e-Print: 1802.03325 doi: 10.1007/s41781-018-0008-x
- [12] M. Erdmann, J. Glombitza and T. Quast 2019 **Precise simulation of electromagnetic**

- calorimeter showers using a Wasserstein Generative Adversarial Network *Comput. Softw. Big Sci.* **3** 4. e-Print: 1807.01954 doi: 10.1007/s41781-018-0019-7
- [13] D. Belayneh, F. Carminati, A. Farbin, B. Hooberman, G. Khattak, M. Liu, J. Liu, D. Olivito, V. B. Pacela, M. Pierini, A. Schwing, M. Spiropulu *et al.* 2020 **Calorimetry with deep learning: particle simulation and reconstruction for collider physics** *The European Physical Journal C* **80** doi: 10.1140/epjc/s10052-020-8251-9
- [14] 2020 **Fast simulation of the ATLAS calorimeter system with Generative Adversarial Networks** Tech. rep. CERN Geneva all figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2020-006> URL: <https://cds.cern.ch/record/2746032>
- [15] F. Carminati, A. Gheata, G. Khattak, P. Mendez Lorenzo, S. Sharan and S. Vallecorsa 2018 **Three dimensional Generative Adversarial Networks for fast simulation** *J. Phys. Conf. Ser.* **1085** 032016 doi: 10.1088/1742-6596/1085/3/032016
- [16] P. Musella and F. Pandolfi 2018 **Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks** *Comput. Softw. Big Sci.* **2** 8. e-Print: 1805.00850 doi: 10.1007/s41781-018-0015-y
- [17] The ATLAS collaboration 2018 **Deep generative models for fast shower simulation in ATLAS** Tech. rep. CERN Geneva URL: <http://cds.cern.ch/record/2630433>
- [18] ATLAS Collaboration 2022 **AtlFast3: The Next Generation of Fast Simulation in ATLAS** *Comput. Softw. Big Sci.* **6** doi: <https://doi.org/10.1007/s41781-021-00079-7>
- [19] H. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn and T. Kuhr 2023 **Ultra-High-Resolution Detector Simulation with Intra-Event Aware GAN and Self-Supervised Relational Reasoning** e-Print: 2303.08046
- [20] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol and K. Krüger 2021 **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed** *Comput. Softw. Big Sci.* **5** 13. e-Print: 2005.05334 doi: 10.1007/s41781-021-00056-0
- [21] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol and K. Krüger 2021 **Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network** *EPJ Web Conf.* **251** 03003. e-Print: 2102.12491 doi: 10.1051/epjconf/202125103003
- [22] E. Buhmann, S. Diefenbacher, D. Hundhausen, G. Kasieczka, W. Korcari, E. Eren, F. Gaede, K. Krüger, P. McKeown and L. Rustige 2022 **Hadrons, better, faster, stronger** *Mach. Learn. Sci. Tech.* **3** 025014. e-Print: 2112.09709 doi: 10.1088/2632-2153/ac7848
- [23] A. Collaboration 2022 **Deep generative models for fast photon shower simulation in ATLAS.** e-Print: 2210.06204
- [24] J. C. Cresswell, B. L. Ross, G. Loaiza-Ganem, H. Reyes-Gonzalez, M. Letizia and A. L. Caterini 2022 **CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds.** e-Print: 2211.15380
- [25] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger, P. McKeown and L. Rustige 2023 **New Angles on Fast Calorimeter Shower Simulation** e-Print: 2303.18150
- [26] C. Chen, O. Cerri, T. Q. Nguyen, J. R. Vlimant and M. Pierini 2021 **Analysis-Specific Fast Simulation at the LHC with Deep Learning** *Computing and Software for Big*



- Science* **5** 15 doi: 10.1007/s41781-021-00060-4
- [27] C. Krause and D. Shih 2021 **CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows** e-Print: 2106.05285 doi: 10.48550/arXiv.2106.05285
  - [28] C. Krause and D. Shih 2021 **CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows** e-Print: 2110.11377 doi: 10.48550/arXiv.2110.11377
  - [29] S. Schnake, D. Krücker and K. Borras 2022 **Generating Calorimeter Showers as Point Clouds** URL: [https://ml4physicalsciences.github.io/2022/files/NeurIPS\\_ML4PS\\_2022\\_77.pdf](https://ml4physicalsciences.github.io/2022/files/NeurIPS_ML4PS_2022_77.pdf)
  - [30] C. Krause, I. Pang and D. Shih 2023 **CaloFlow for CaloChallenge Dataset 1.** e-Print: 2210.14245
  - [31] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, C. Krause, I. Shekhzadeh and D. Shih 2023 **L2LFlows: Generating High-Fidelity 3D Calorimeter Images** e-Print: 2302.11594
  - [32] A. Xu, S. Han, X. Ju and H. Wang 2023 **Generative Machine Learning for Detector Response Modeling with a Conditional Normalizing Flow.** e-Print: 2303.10148
  - [33] M. R. Buckley, C. Krause, I. Pang and D. Shih 2023 **Inductive CaloFlow.** e-Print: 2305.11934
  - [34] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan and S. Ganguli 2015 **Deep Unsupervised Learning using Nonequilibrium Thermodynamics.** e-Print: 1503.03585
  - [35] Y. Song and S. Ermon 2020 **Generative Modeling by Estimating Gradients of the Data Distribution.** e-Print: 1907.05600
  - [36] Y. Song and S. Ermon 2020 **Improved Techniques for Training Score-Based Generative Models.** e-Print: 2006.09011
  - [37] J. Ho, A. Jain and P. Abbeel 2020 **Denoising Diffusion Probabilistic Models.** e-Print: 2006.11239
  - [38] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon and B. Poole 2021 **Score-Based Generative Modeling through Stochastic Differential Equations.** e-Print: 2011.13456
  - [39] V. Mikuni and B. Nachman 2022 **Score-based generative models for calorimeter shower simulation** *Phys. Rev. D* **106** 092009. e-Print: 2206.11898 doi: 10.1103/PhysRevD.106.092009
  - [40] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, W. Korcar, K. Krüger and P. McKeown 2023 **CaloClouds: Fast Geometry-Independent Highly-Granular Calorimeter Simulation.** e-Print: 2305.04847
  - [41] F. T. Acosta, V. Mikuni, B. Nachman, M. Arratia, B. Karki, R. Milton, P. Karande and A. Angerami 2023 **Comparison of Point Cloud and Image-based Models for Calorimeter Fast Simulation.** e-Print: 2307.04780
  - [42] V. Mikuni and B. Nachman 2023 **CaloScore v2: Single-shot Calorimeter Shower Simulation with Diffusion Models.** e-Print: 2308.03847
  - [43] O. Amram and K. Pedro 2023 **CaloDiffusion with GLaM for High Fidelity Calorimeter Simulation.** e-Print: 2308.03876
  - [44] A. Butter, T. Plehn and R. Winterhalder 2019 **How to GAN LHC Events** *SciPost Phys.*

- 7 075. e-Print: 1907.03764 doi: 10.21468/SciPostPhys.7.6.075
- [45] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini 2019 **LHC analysis-specific datasets with Generative Adversarial Networks** e-Print: 1901.05282
  - [46] R. Di Sipio, M. Fauci Giannelli, S. Ketabchi Haghighat and S. Palazzo 2019 **DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC** *JHEP* **08** 110. e-Print: 1903.02433 doi: 10.1007/JHEP08(2019)110
  - [47] J. Arjona Martínez, T. Q. Nguyen, M. Pierini, M. Spiropulu and J.-R. Vlimant 2020 **Particle Generative Adversarial Networks for full-event simulation at the LHC and their application to pileup description** *J. Phys. Conf. Ser.* **1525** 012081. e-Print: 1912.02748 doi: 10.1088/1742-6596/1525/1/012081
  - [48] Y. Alanazi, N. Sato, T. Liu, W. Melnitchouk, P. Ambrozewicz, F. Hauenstein, M. P. Kuchera, E. Pritchard, M. Robertson, R. Strauss, L. Velasco and Y. Li 2021 **Simulation of Electron-Proton Scattering Events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN)** *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21* edited by Z.-H. Zhou (IJCAI) p 2126. e-Print: 2001.11103 doi: 10.24963/ijcai.2021/293
  - [49] S. Otten, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen, D. Podareanu, R. Ruiz de Austri and R. Verheyen 2021 **Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer** *Nature Commun.* **12** 2985. e-Print: 1901.00875 doi: 10.1038/s41467-021-22616-z
  - [50] A. Butter, T. Heimel, S. Hummerich, T. Krebs, T. Plehn, A. Rousselot and S. Vent 2021 **Generative Networks for Precision Enthusiasts** e-Print: 2110.13632
  - [51] L. de Oliveira, M. Paganini and B. Nachman 2017 **Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis** *Computing and Software for Big Science* **1** doi: 10.1007/s41781-017-0004-6
  - [52] A. Andreassen, I. Feige, C. Frye and M. D. Schwartz 2019 **JUNIPR: a framework for unsupervised machine learning in particle physics** *The European Physical Journal C* **79** doi: 10.1140/epjc/s10052-019-6607-9
  - [53] E. Bothmann and L. D. Debbio 2019 **Reweighting a parton shower using a neural network: the final-state case** *Journal of High Energy Physics* **2019** doi: 10.1007/jhep01(2019)033
  - [54] K. Dohi 2020 **Variational Autoencoders for Jet Simulation** e-Print: 2009.04842
  - [55] R. Kansal, J. Duarte, H. Su, B. Orzari, T. Tomei, M. Pierini, M. Touranakou, J.-R. Vlimant and D. Gunopulos 2022 **Particle Cloud Generation with Message Passing Generative Adversarial Networks.** e-Print: 2106.11535
  - [56] B. Käch, D. Krücker, I. Melzer-Pellmann, M. Scham, S. Schnake and A. Verney-Provatas 2022 **JetFlow: Generating Jets with Conditioned and Mass Constrained Normalising Flows** e-Print: 2211.13630
  - [57] B. Käch, D. Krücker and I. Melzer-Pellmann 2022 **Point Cloud Generation using Transformer Encoders and Normalising Flows.** e-Print: 2211.13623
  - [58] R. Kansal, A. Li, J. Duarte, N. Chernyavskaya, M. Pierini, B. Orzari and T. Tomei 2023 **Evaluating generative models in high energy physics** *Physical Review D* **107** doi: 10.1103/physrevd.107.076017

- [59] E. Buhmann, G. Kasieczka and J. Thaler 2023 **EPiC-GAN: Equivariant Point Cloud Generation for Particle Jets** e-Print: 2301.08128
- [60] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch and T. Golling 2023 **PC-JeDi: Diffusion for Particle Cloud Generation in High Energy Physics.** e-Print: 2303.05376
- [61] B. Käch and I. Melzer-Pellmann 2023 **Attention to Mean-Fields for Particle Cloud Generation.** e-Print: 2305.15254
- [62] A. Butter, N. Huetsch, S. P. Schweitzer, T. Plehn, P. Sorrenson and J. Spinner 2023 **Jet Diffusion versus JetGPT – Modern Networks for the LHC.** e-Print: 2305.10475
- [63] M. Leigh, D. Sengupta, J. A. Raine, G. Quétant and T. Golling 2023 **PC-Droid: Faster diffusion and improved quality for particle cloud generation.** e-Print: 2307.06836
- [64] The CMS collaboration 2017 **The Phase-2 Upgrade of the CMS Endcap Calorimeter** Tech. rep. CERN Geneva doi: 10.17181/CERN.IV8M.1JY2 URL: <https://cds.cern.ch/record/2293646>
- [65] V. Mikuni, B. Nachman and M. Pettee 2023 **Fast Point Cloud Generation with Diffusion Models in High Energy Physics.** e-Print: 2304.01266
- [66] Q. Zhang and Y. Chen 2023 **Fast Sampling of Diffusion Models with Exponential Integrator.** e-Print: 2204.13902
- [67] T. Karras, M. Aittala, T. Aila and S. Laine 2022 **Elucidating the Design Space of Diffusion-Based Generative Models.** e-Print: 2206.00364
- [68] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li and J. Zhu 2022 **DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps.** e-Print: 2206.00927
- [69] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li and J. Zhu 2023 **DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models.** e-Print: 2211.01095
- [70] E. Luhman and T. Luhman 2021 **Knowledge Distillation in Iterative Generative Models for Improved Sampling Speed.** e-Print: 2101.02388
- [71] T. Salimans and J. Ho 2022 **Progressive Distillation for Fast Sampling of Diffusion Models.** e-Print: 2202.00512
- [72] X. Liu, C. Gong and Q. Liu 2022 **Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow.** e-Print: 2209.03003
- [73] H. Zheng, W. Nie, A. Vahdat, K. Azizzadenesheli and A. Anandkumar 2023 **Fast Sampling of Diffusion Models via Operator Learning.** e-Print: 2211.13449
- [74] D. Berthelot, A. Autef, J. Lin, D. A. Yap, S. Zhai, S. Hu, D. Zheng, W. Talbott and E. Gu 2023 **TRACT: Denoising Diffusion Models with Transitive Closure Time-Distillation.** e-Print: 2303.04248
- [75] Y. Song, P. Dhariwal, M. Chen and I. Sutskever 2023 **Consistency Models.** e-Print: 2303.01469
- [76] The ILD Concept Group edited by T. Behnke *et al.* 2020 **International Large Detector: Interim Design Report** e-Print: 2003.01116
- [77] The iLCSoft Group 2016 **iLCSoft Project Page** [Online; accessed 10-September-2022] URL: <https://github.com/iLCSoft>
- [78] M. Frank, F. Gaede, C. Grefe and P. Mato 2014 **DD4hep: A Detector Description Toolkit for High Energy Physics Experiments** *Journal of Physics: Conference*

- Series **513** 022010 doi: 10.1088/1742-6596/513/2/022010
- [79] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever and M. Welling 2016 **Improved Variational Inference with Inverse Autoregressive Flow** *Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS'16* (Red Hook, NY, USA: Curran Associates Inc.) p 4743 ISBN 9781510838819. e-Print: 1606.04934 doi: 10.48550/arXiv.1606.04934
  - [80] D. P. Kingma and M. Welling 2013 **Auto-Encoding Variational Bayes** e-Print: 1312.6114 doi: 10.48550/arxiv.1312.6114
  - [81] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf *et al.* 2019 **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. e-Print: 1912.01703
  - [82] L. Dinh, J. Sohl-Dickstein and S. Bengio 2016 **Density estimation using Real NVP** e-Print: 1605.08803
  - [83] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios 2019 **Neural Spline Flows**. e-Print: 1906.04032
  - [84] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall and N. D. Goodman 2019 **Pyro: Deep Universal Probabilistic Programming** *J. Mach. Learn. Res.* **20** 28:1–28:6
  - [85] S. Luo and W. Hu 2021 **Diffusion Probabilistic Models for 3D Point Cloud Generation**. e-Print: 2103.01458
  - [86] J. Song, C. Meng and S. Ermon 2022 **Denoising Diffusion Implicit Models**. e-Print: 2010.02502
  - [87] D. P. Kingma and J. Ba 2015 **Adam: A Method for Stochastic Optimization** *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* edited by Y. Bengio and Y. LeCun. e-Print: 1412.6980 doi: 10.48550/arXiv.1412.6980
  - [88] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.* 2013 **Rectifier nonlinearities improve neural network acoustic models** *Proc. icml* vol 30 (Citeseer) p 3 URL: [http://robotics.stanford.edu/~amaas/papers/relu\\_hybrid\\_icml2013\\_final.pdf](http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf)
  - [89] R. Das, L. Favaro, T. Heimel, C. Krause, T. Plehn and D. Shih 2023 **How to Understand Limitations of Generative Networks**. e-Print: 2305.16774
  - [90] T. Dao, D. Y. Fu, S. Ermon, A. Rudra and C. Ré 2022 **FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness**. e-Print: 2205.14135
  - [91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin 2023 **Attention Is All You Need**. e-Print: 1706.03762

## Appendix A. Radial and longitudinal energy observables

To explore the radial and longitudinal energy profile shown in Fig. 3 further and to calculate the evaluation scores in Sec. 4.2, we define ten radial and longitudinal energy observables for the calorimeter showers.

Respectively, the ten observables are defined such that energy is clustered in each observable with an equal amount of statistics. Put differently, the energy is

binned in ten quantiles with approximately the same number of cell hits in each quantile. The energy bins are defined by the quantiles calculated on the GEANT4 test set with 40,000 events. While the bin edges are precisely defined for the radial energy, we round the bin edges of the longitudinal observables to the nearest layer integer number.

Histograms of the radial energy observables  $E_{\text{radial},i \in [1,10]}$  are shown in Fig. A1 and of longitudinal energy observables  $E_{\text{long},i \in [1,10]}$  in Fig. A2. The bin edges for all observables are given in Tab. A1.

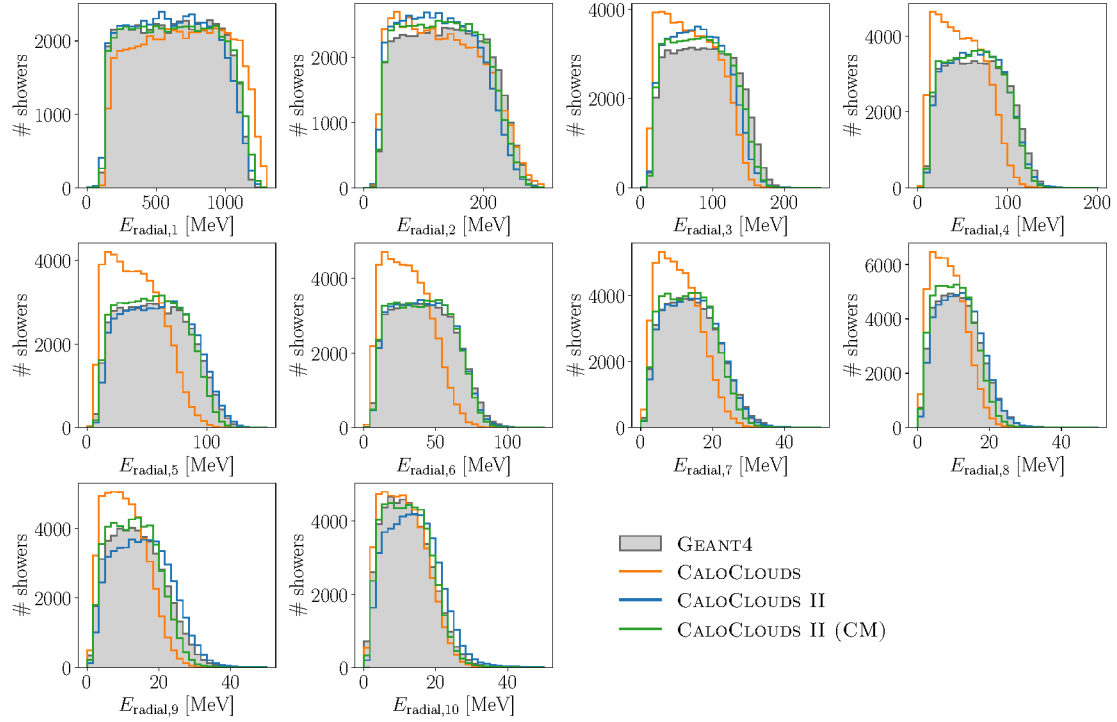


Figure A1: Radial energy observables for 50,000 showers.

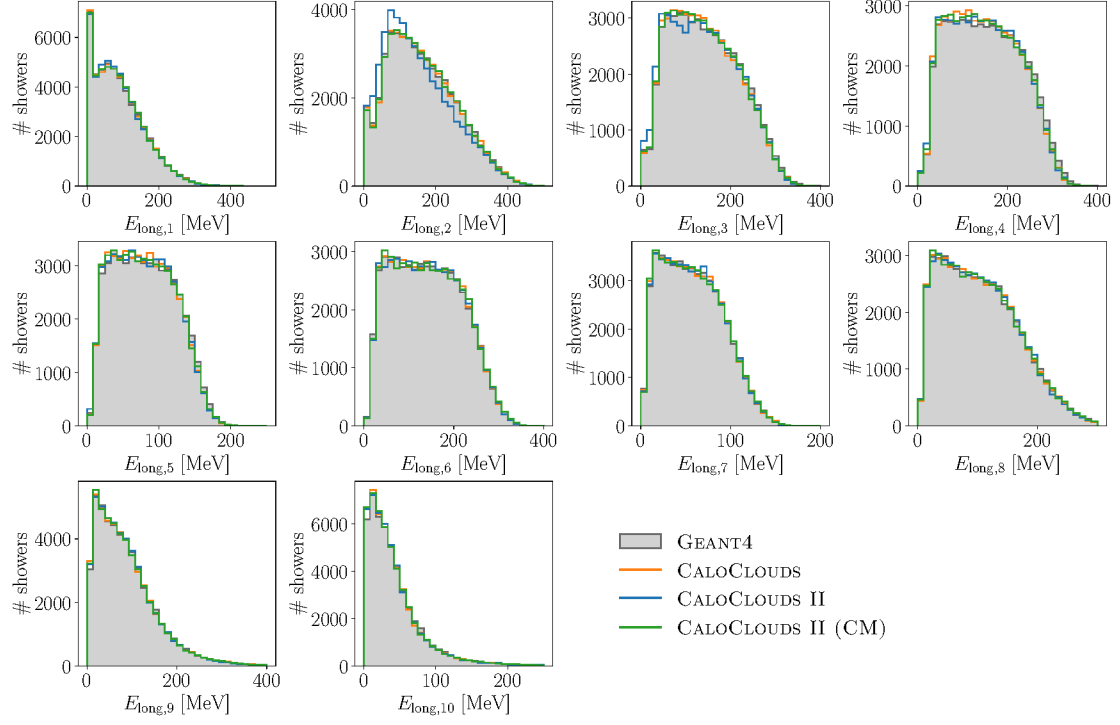


Figure A2: Longitudinal energy observables for 50,000 showers.

| Bin edges                                         | 0 | 1   | 2   | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10  |
|---------------------------------------------------|---|-----|-----|------|------|------|------|------|------|------|-----|
| Edges for $E_{\text{radial}, i \in [1,10]}$ [mm]  | 0 | 6.6 | 9.8 | 13.0 | 17.0 | 23.4 | 33.6 | 40.1 | 48.5 | 68.8 | 300 |
| Edges for $E_{\text{long}, i \in [1,10]}$ [layer] | 1 | 9   | 12  | 14   | 16   | 17   | 19   | 20   | 22   | 25   | 30  |

Table A1: Bin edges for calculating the radial and longitudinal energy observables  $E_{\text{radial}, i \in [1,10]}$  and  $E_{\text{long}, i \in [1,10]}$ . Determined for ten quantiles each including approximately the same number of cell hits. All bins are half-open, except the last bin.