Normalizing flows for high-dimensional detector simulations

Florian Ernst[1,2], **Luigi Favaro**[1,3], **Claudius Krause**[1,4], **Tilman Plehn**[1] and **David Shih**[5]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Experimental Physics Department, CERN, Geneva, Switzerland
3 CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium
4 Institut für Hochenergiephysik (HEPHY),
Österreichische Akademie der Wissenschaften (ÖAW), Vienna, Austria
5 NHETC, Department of Physics & Astronomy, Rutgers University, Piscataway, NJ USA

Abstract

Whenever invertible generative networks are needed for LHC physics, normalizing flows show excellent performance. In this work, we investigate their performance for fast calorimeter shower simulations with increasing phase space dimension. We use fast and expressive coupling spline transformations applied to the CaloChallenge datasets. In addition to the base flow architecture we also employ a VAE to compress the dimensionality and train a generative network in the latent space. We evaluate our networks on several metrics, including high-level features, classifiers, and generation timing. Our findings demonstrate that invertible neural networks have competitive performance when compared to autoregressive flows, while being substantially faster during generation.

Received 2023-12-22
Accepted 2025-01-13
Published 2025-03-05
doi:10.21468/SciPostPhys.18.3.081

Check for updates

Contents

1 Introduction

Simulations are a defining aspect of LHC physics, bridging experiment and fundamental theory and allowing for a proper interpretation of LHC measurements [1, 2]. The simulation chain starts from the hard interaction at the scattering vertex, and progresses through the radiation of soft particles, the decay of heavy, unstable particles, hadronization of colored states, and subsequent interaction of all particles in the event with the detector. In the development of LHC as a precision-hadron collider, the last step has become a major bottleneck in speed and precision, in particular the reproduction of the detailed interactions of incident and secondary particles within the calorimeters. Generating these calorimeter showers with GEANT4 [3–5], based on first principles, takes a substantial amount of the LHC computing budget. Without significant progress, simulations will be the limiting factor for all analyses at the high-luminosity upgrade of the LHC.

One development driving faster and more precise LHC simulations is the advent of deep generative networks. Fast detector simulations based on parametric models have been extensively used in previous measurements from LHC experiments [6, 7], although their precision can be improved with machine learning components [8]. Such networks have shown great promise for LHC physics in the past few years, providing fast and accurate surrogates for simulations in high-dimensional phase spaces [9]. They learn the underlying probability distribution of events or calorimeter showers from a reference dataset and then generate new samples based on this learned distribution [10, 11]. We have seen successful applications to all steps in the simulation chain [2], phase space integration [12–22]; parton showers [23–30]; hadronization [31–34]; detector simulations [8, 35–63]; and end-to-end event generation [64–70].

For LHC physics, it is crucial that these networks are not used as black boxes, but their performance can be investigated, understood, and improved systematically [69, 71–75]. This is especially important when their conditional counterparts are used for inference [76–78], probabilistic unfolding [79–88], or anomaly detection [89–94].

In this paper, we will focus on the problem of building fast and accurate surrogate models for calorimeter shower simulation, using the technology of normalizing flows. We have seen in a number of contexts that normalizing flows are a promising technique for fast calorimeter simulation [44, 45, 51, 75], but there are also major challenges with scaling them up to more granular (higher-dimensional) calorimeters [52, 53, 58, 63]. These challenges are especially interesting because recently, continuous-time generative models (diffusion models and continuous normalizing flows trained with flow-matching) have been tested on LHC physics [28–30, 49, 57, 59, 60, 70, 74, 78, 95–98] and show impressive performance which is not as limited by the dimensionality of the data. However, their gain in expressivity comes at the expense of slower generation, leading to an interesting trade-off between speed and quality of generated events or showers.

Table 1: Sample sizes for different incident energies in dataset 1.

E_{inc}	256 MeV ... 131 GeV	262 GeV	0.524 TeV	1.04 TeV	2.1 TeV	4.2 TeV
photons	10000 per energy	10000	5000	3000	2000	1000
pions	10000 per energy	9800	5000	3000	2000	1000

Here, we will build on previous works [52, 53, 58, 63] attempting to scale up normalizing flows to higher-granularity calorimeters. Focusing on the datasets [99–102] of the Fast Calorimeter Simulation Challenge [103, 104], we will tackle this problem in two ways.

- First, we will show how impressive gains in speed can be achieved by switching from the fully-autoregressive flows of [44, 45, 51, 53, 58] to flows based on coupling layers [105–107] which are equally fast in the sampling and density estimation directions, while retaining or improving the network accuracy. Following the terminology of [108, 109], we will refer to coupling-layer based flows as *invertible neural networks* (INNs) throughout this work. Using the INN framework, we are able to obtain state-of-the-art results on dataset 1 (pions and photons) and dataset 2 of the CaloChallenge.

- Second, to reach the dimensionality of dataset 3, we will combine the INN framework with a VAE. Conceptually similar to other approaches, [52, 110, 111], we will train the INN on the (much lower-dimensional) latent space of a VAE fit to the showers of dataset 3. Then sampling from the INN and passing this through the decoder of the VAE, we will obtain a surrogate model for dataset 3. We will see that the results here, while not state-of-the-art in terms of quality, are very fast to generate, so could fill out another point in the Pareto frontier of fast calorimeter shower simulation.

The paper starts by introducing the CaloChallenge datasets in Sec. 2. In Sec. 3 we introduce our fast INN version [108, 109] of a normalizing flow, as well as a VAE+INN combination. In Sec. 4 we discuss their performance on the different dataset, with increasing phase space dimensionality and including learned classifier weights. We conclude and provide timing information in Sec. 5. In the Appendices we provide details on the different network architectures and hyperparameters and compare the INN performance to CaloFlow.

2 Datasets

As stated above, our reference datasets are the public datasets of the Fast Calorimeter Simulation Challenge [103, 104] and represent three increasing dimensionalities from the current LHC calorimeter granularity to the ultra high granularity of future calorimeters proposed for ILC [112], CLIC [113], FCC [114] and beyond. We use the public datasets [99–102] of the Fast Calorimeter Simulation Challenge [103]. They consist of showers simulated with GEANT4 for different incident particles. The general geometry is the same across all datasets: the detector volume is segmented into layers in the direction of the incoming particle. Each layer is segmented along polar coordinates in radial (r) and angular (α) bins. A shower is given as the incident energy of the incoming particle and the energy depositions in each voxel.

Dataset 1 (DS1) provides calorimeter showers for central photons and charged pions. They have been used in ATLFAST3 [8]. The voxelizations of the 5 photon layers and 7 pion layers in radial and angular bins ($n_r \times n_\alpha$) are

$$\begin{aligned} \text{photons} \quad & 8\times 1,\ 16\times 10,\ 19\times 10,\ 5\times 1,\ 5\times 1\,, \\ \text{pions} \quad & 8\times 1,\ 10\times 10,\ 10\times 10,\ 5\times 1,\ 15\times 10,\ 16\times 10,\ 10\times 1\,. \end{aligned} \tag{1}$$

This gives 368 voxels for photons and 533 voxels for pions. The incoming particles are simulated for 15 different incident energies $E_{\text{inc}} = 256$ MeV ... 4.2 TeV, increasing by factors of two, with the sample sizes given in Tab. 1. The original ATLAS dataset does not require an energy threshold. The effect of a threshold on the shower distributions at the detector cell level requires further studies. We require $E_{\text{min}} = 1$ MeV to all generated voxels, motivated by the readout threshold of the calorimeter cells and the fact that photon showers require a minimum cell energy of 10 MeV to cluster and pion showers start clustering at 300 MeV [115]. We note that the usually called E_{inc} is in reality the momentum of the incoming particle. This has implications for pions which have to be further studied for a future deployment.

Datasets 2 and 3 (DS2/3) are not modeled after existing detectors. They assume 45 layers of active silicon detector (thickness 0.3 mm), alternating with inactive tungsten absorber layers (thickness 1.4 mm) at $\eta = 0$. Each dataset contains 100,000 GEANT4 electron showers with log-uniform $E_{\text{inc}} = 1$ ... 1000 GeV. The only difference between the two datasets is the voxelization. In dataset 2, each layer is divided into 16×9 angular and radial voxels, defining 6480 voxels in total. Dataset 3 uses 50×18 voxels per layer or 40,500 voxels in total. The minimal recorded energy per voxel for these two datasets is 15.15 keV.

3 CaloINN

We study two different network architectures. First, we benchmark a standard INN and demonstrate its precision and generation speed especially for low-dimensional phase space. Second, we embed this INN in a VAE, with the goal of describing datasets 2 and 3 with the same physics content, but a much larger phase space dimensionality.

3.1 INN

Normalizing flows describe bijective mappings between a (Gaussian) latent space r and the physical phase space x,

$$p_{\text{latent}}(r) \quad \underset{\leftarrow \overline{G}_\theta(x)}{\overset{G_\theta(r)\rightarrow}{\longleftrightarrow}} \quad p_{\text{model}}(x) \sim p_{\text{data}}(x)\,. \tag{2}$$

$\overline{G}_\theta(x)$ denotes the inverse transformation to $G_\theta(r)$. The INN variant [108,109] of normalizing flows is completely symmetric in the two directions. After training the network, $p_{\text{data}}(x) \sim p_{\text{model}}(x)$, we use the INN to sample $p_{\text{model}}(x)$ from $p_{\text{latent}}(r)$ [9].

The building block of our INN architecture is the coupling layer [105–107]. It allows for a Jacobian calculable in a single network evaluation for both $\overline{G}_\theta(x)$ and $G_\theta(r)$. Therefore, we train the INN with a likelihood loss

$$\mathcal{L}_{\text{INN}} = -\Big\langle \log p_{\text{model}}(x) \Big\rangle_{p_{\text{data}}} = -\left\langle \log p_{\text{latent}}\big(\overline{G}_\theta(x)\big) + \log \left| \frac{\partial \overline{G}_\theta(x)}{\partial x} \right| \right\rangle_{p_{\text{data}}} . \tag{3}$$

The first term ensures that the latent representation remains Gaussian, while the second term constructs the correct transformation to the phase space distribution. Given the structure of $\overline{G}_\theta(x)$ and the latent distribution p_{latent}, both terms can be computed efficiently.

Figure 1 (left) shows a schematic representation of the CaloINN architecture. In the coupling block we split the input vector, consisting of the normalized voxels x and the energy variables u, in two equally-sized vectors. The first half of the vector is not transformed and used, together with the logarithm of the incident energy E_{inc}, to predict the parameters of the transformation applied to the second half.

A standard affine transformation uses a scale and shift parameter for each voxel. Instead, we define a spline parametrized by a neural network. We employ rational quadratic splines (RQS) and cubic splines. The splines are defined piecewise in a box of size $[-B, B]$. Given the total number of bins K, the spline is parametrized by the locations of each knot and their first derivatives. In one bin k, the two transformations have the form

$$f_{\text{RQS}}(x)_k = \frac{\alpha_{0,k} + \alpha_{1,k}x + \alpha_{2,k}x^2}{\beta_{0,k} + \beta_{1,k}x + \beta_{2,k}x^2}, \qquad \text{and} \qquad f_{\text{cubic}}(x)_k = \gamma_{0,k} + \gamma_{1,k}x + \gamma_{2,k}x^2 + \gamma_{3,k}x^3, \quad (4)$$

respectively. The parameters $(\alpha, \beta)_k$ and $(\gamma)_k$ can be expressed as a function of the bin height, width, and first derivative in a stable numerical form. The complete parametric expression for the RQS can be found in [116], while the implementation of the cubic spline follows [117]. The total number of parameters predicted by the neural network, after accounting for the continuity constraints, are $3K - 1$ for each transformed variable. The large scale architecture stacks several coupling blocks each one followed by a permutation of the input and an ActNorm [107] layer for normalization purposes. The INN is implemented using the FrEIA[1] package [118]. For datasets 1, we employ a rational quadratic spline, while for dataset 2 we find cubic splines to give more stable results. A discussion on the ablation studies is provided in App. A.1 together with all the INN hyperparameters.

As a noteworthy preprocessing we normalize each shower to the layer energy. The energy information is encoded as

$$u_0 = \frac{\sum_i E_i}{E_{\text{inc}}}, \qquad \text{and} \qquad u_i = \frac{E_i}{\sum_{j \geq i} E_j}, \quad (5)$$

in terms of the energy depositions per layer E_i. The u_i are appended to the list of voxels for each shower. We do not explore a separate training for the energy and the voxel dimensions which would simplify the learning process of the energy dimensions. We train the INN on the full data, conditioned on the logarithm of the incident energies. Unlike, for instance, CaloFlow [45,51,58] we train a single network without any distillation. We provide the details of the preprocessing in App. A.1.

3.2 VAE+INN

The problem with the INN is the scaling towards dataset 3 with its high-dimensional phase space of 40k voxels. The INN scales at least linearly in time and memory with the input dimension since each voxel is processed by a spline that has to be parameterized independently. In practice, the scaling is usually worse than linear, as the number of parameters, needed to parameterize each spline, tends to grow with the number of voxels as well. To solve this scaling problem we introduce an additional VAE to reduce the dimensionality of the INN mapping. Differently from [52], we do not estimate the dimensionality of the manifold but rather optimize the reconstruction of the VAE while keeping a low-dimensional latent space. The VAE consists of a preprocessing block, an encoder-decoder combination, and a postprocessing block. Both, the decoder and the encoder are conditioned on the incident energies and additional energy variables. Therefore, we compress normalized showers in the latent space and jointly learn the energy and the latent variables with the INN. During generation, the INN samples into the latent space of the VAE, and the VAE decoder translates this information to the shower phase space. We set the latent space to 50 for dataset 1 and dataset 2, and to 300 for dataset 3. Other specifics of the network are different in the three datasets and are provided in App. A.1.

[1] We provide the code in a Github repository at https://github.com/heidelberg-hepml/CaloINN.

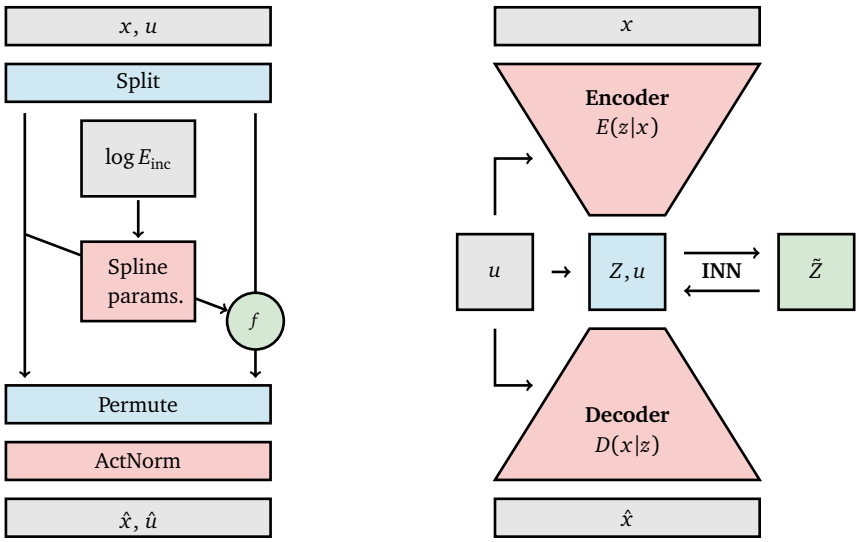

Figure 1: Schematic representation of the CaloINN (left) and the CaloVAE+INN (right) architectures.

The goal of our β-VAE [119, 120] is to learn to reconstruct the input data. We assume a Gaussian distribution for the encoder network $E(z|x)$. The VAE loss for the compression is

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \beta D_{\text{KL}}[E(z|x), p_{\text{latent}}(z)], \tag{6}$$

with the usual binary cross entropy loss and the Gaussian prior. For a Gaussian encoder the KL-divergence can be computed analytically, and the coupling strength is $\beta = 10^{-9}$. We select this small value as the KL part is only a regularization in our setup. We do not need a Gaussian latent space since a more expressive mapping is learned by the INN. However, we need very accurate decoding abilities from the latent space. The only requirement, that is ensured by the small KL term, is a compact well-behaved space which can be learned by a generative model. For numerical stability we split it into an upscaling factor for the BCE part and a downscaling factor for the KL part.

For the decoder we use a Bernoulli likelihood, because it outperforms other models. For example the Gaussian and the continuous Bernoulli [121] approach. The Gaussian decoder does not model the shower geometry well, and it under-populates the low-energy regions. The continuous Bernoulli distribution leads to instabilities, as the average energy deposition in the normalized space is close to zero. We use a Bernoulli decoder,

$$D(x|\lambda(z)) = \lambda(z)^x (1-\lambda(z))^{1-x}, \tag{7}$$

defining the combined VAE loss

$$\mathcal{L}_{\text{VAE}} = \left\langle \left\langle x \log \lambda + (1-x) \log (1-\lambda) \right\rangle_{z\sim E(z|x)} + \beta \left[1 + \log \sigma_E^2 - \mu_E^2 - \sigma_E^2 \right] \right\rangle_{x\sim p_{\text{data}}}. \tag{8}$$

Because the Bernoulli distribution gives a binary probability we use its continuous mean λ as the prediction for the individual voxels.

The remaining differences between the unit-Gauss prior in the latent space and the encoder are mapped by the INN. Applying a 2-step training, we first train the VAE and then train the INN given the learned latent space. This means we pass the encoder means and the standard deviations, as well as the energy variables to the INN. The INN is trained as described above, mapping the latent representation of the VAE to a standard Gaussian. As for the full INN, the

energy information is encoded following Eq (5) and learned by the latent flow. Both encoder and decoder of the VAE are conditioned to these variables.

For the larger datasets 2 and 3, we employ a mixture of a convolutional and a fully connected VAE. Our assumption is that the calorimeter layers do not require a full correlation, but that only neighboring layers are strongly correlated. This assumption is simply implementing locality, which should be given due to the causal propagation of the shower through the calorimeter layers and the regular structure of electromagnetic showers. Therefore, we can simplify the structure by compressing consecutive layers jointly in a first-step compression. We use an architecture with fully connected sub-blocks, resembling a kernel architecture with a kernel size k (number of jointly encoded calorimeter layers) and a stride s (distance between two neighboring kernel blocks). After this first compression we concatenate these latent subspaces and compress them a second time into our final latent space. For the decoding we reverse this two-step structure. The overlapping regions of the fully connected kernel blocks are summed over. It should be noted, that the large scale correlations are not completely ignored in this approach. They can still be learned as correlations between the kernel blocks in the second stage. However, it is harder to model them as the information content is already compressed by the first dimensionality reduction.

4 Results

The main physics reason for specific shower features is the incident energy. Low-energy showers will interact with only a few layers of the calorimeter and quickly widen, leading to a broad center-of-energy distribution in earlier calorimeter layers and a high sparsity in the given voxelization. High-energy showers penetrate the calorimeter more deeply. They will be collimated in the initial layers and have low sparsity since each shower is likely to deposit energy in each voxel.

To see if the ML-learned showers reflect these physics properties, we look at physics-motivated and high-level features. Given a shower with energy depositions $\mathcal{I}$, we look at the center of energy and its width for each layer,

$$\langle \zeta \rangle = \frac{\zeta \cdot \mathcal{I}}{\sum_i \mathcal{I}_i}, \qquad \text{and} \qquad \sigma_{\langle \zeta \rangle} = \sqrt{\frac{\zeta^2 \cdot \mathcal{I}}{\sum_i \mathcal{I}_i} - \langle \zeta \rangle^2}, \qquad \text{for} \qquad \zeta = \eta, \phi\,, \tag{9}$$

where $\sum_i$ runs over the voxels in one layer. We also look at the energy deposition in each layer; the layer sparsity; and for dataset 1, the ratio $E_{\text{tot}}/E_{\text{inc}}$ for each discrete incident energy.

To analyze the quality of our generative networks in more detail and to identify failure modes, we train a classifier $D(x)$ on the voxels, to distinguish GEANT4 showers from generated showers [44, 45, 75]. By the Neyman-Pearson lemma the trained classifier approximates the likelihood-ratio. This means we can compute the correction weight [75] and use the weight distributions as an evaluation metric

$$w(x) = \frac{D(x)}{1 - D(x)} \approx \frac{p_{\text{data}}}{p_{\text{model}}}(x)\,. \tag{10}$$

For these weights it is crucial that we evaluate them on the training and on the generated datasets combined, because typical failure modes correspond to tails for one of the two datasets [75]. In addition, we always check if showers with especially small or large weights cluster in phase space, allowing us to identify failure modes of the respective generative network. We also report the Area-Under-the-Curve (AUC) score, which is calculated after training ten classifiers from different initializations and averaging the obtained AUC scores.

4.1 Dataset 1 photons

We start with the photons in dataset 1. At high energy the interactions with matter in a photon shower are dominated by pair production and Bremsstrahlung, making this dataset the simplest in terms of dimensionality and complexity. We summarize the most interesting high-level features for the GEANT4 training data, the INN generator, and the VAE+INN generator in Fig. 2.

We first look at the shower shape in rapidity for the layer with the largest energy deposition for $E_{\text{inc}} = 0.256, 8.2, 262.1$ GeV. These energies provide insights on the generation over the entire spectrum of low, medium, and high energetic showers. For instance, we show the center of energy and its width in the calorimeter layer-1 for $E_{\text{inc}} = 256$ MeV, while, for the remaining energies, we show the layer 2. Dataset 1 is not symmetric in η and ϕ, because the shower were not generated around $\eta = \phi = 0$. All showers have the same mean width, regardless of the incident energy. This is captured by both networks at the level of 5% to 20%. A failure mode of the INN is the region $\sigma_{\langle\eta\rangle} < 20$ mm for low energies, where the network undershoots the training data by up to 50% in the first bin. A peculiar feature of these distributions is a small peak at zero, which occurs when at most one voxel per layer receives a hit. These cases are better reproduced by the VAE, whereas the INN tends to produce slightly more collimated showers. The INN is able to reproduce the collimated showers at higher energies always within the statistical uncertainties of the training data, both in the center of energies and their widths.

Next, we show the energy depositions in layers 1. Although the energy deposited in layer-2 is larger for the intermediate and the large incident energies, we focus on layer-1 to showcase the performance over a larger range of energies. Both networks show comparable performance over the entire energy range for $E_{\text{inc}} = 8.2, 262.1$ GeV, while the VAE has larger deviations at lower energies.

Finally, we look at observables inclusive in the energy. The sparsity λ_2 in the same layer is determined by the energy threshold of 1 MeV. The INN matches the truth over the entire λ-range to 10%, while the VAE struggles. In particular, its showers have too many active voxels, leading to the mis-modeled peak close to zero.

The ratio $E_{\text{tot}}/E_{\text{inc}}$ exhibits a small bias in the energy generation for the VAE+INN towards low energies, artifact of the final threshold in the architecture. For smaller incident energies, more voxels are zero [53], which causes a problem for the VAE+INN because the showers are more sparse which is the weakness of the VAE.

To illustrate the discrete structure of the incident energies in dataset 1, we collect $E_{\text{tot}}/E_{\text{inc}}$ for each incident energy in Fig. 3. The incident energy, provided during training and generation, carries energy-dependent information about the shower. For instance, low-energy showers have a much broader energy ratio distribution, in contrast to high-energy showers. Both generative networks learn the conditional distribution on E_{inc} with deviations up to 30% in the tails. We include a set of shower shape histograms inclusive in the energy in App. A.3 and the full set of histograms for each energy with the published samples [122].

4.2 Dataset 1 pions

The physics of hadronic showers is significantly more complex than photon showers, so it is interesting to see how our INNs perform for a low-dimensional calorimeter simulation of pions. As before, we show shower shapes, sparsity, energy depositions, and the fraction of deposited energy in Fig. 4. We focus on three distinct incident energies, 256 MeV, 8.2 GeV, and 262.1 GeV. In particular we show the layers with the largest energy deposition. For the lowest incident energy, a large fraction of energy is deposited in layer-2 while for the other two cases we show layer-12 and layer-13, respectively.

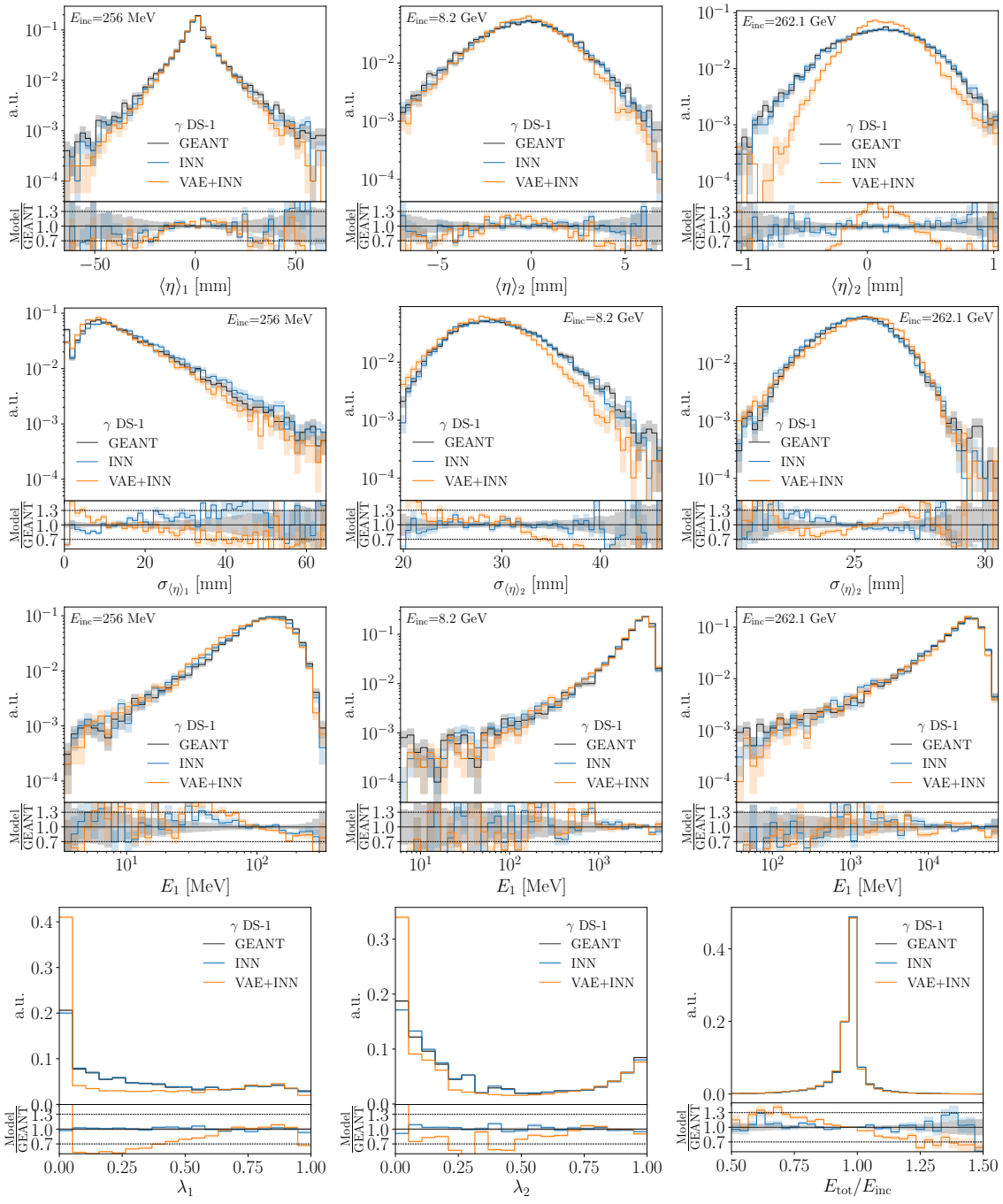

Figure 2: Set of high-level features for γ showers in dataset 1, compared between GEANT4, INN, and VAE+INN. We show the energy deposition, the center of energy, and the width of the center of energy in layer-1 for the incident energies 256 MeV, 8.2 GeV, and 262.1 GeV. The last row contains the inclusive sparsity in layer-1 and layer-2, and the inclusive energy ratio E_{tot}/E_{inc}.

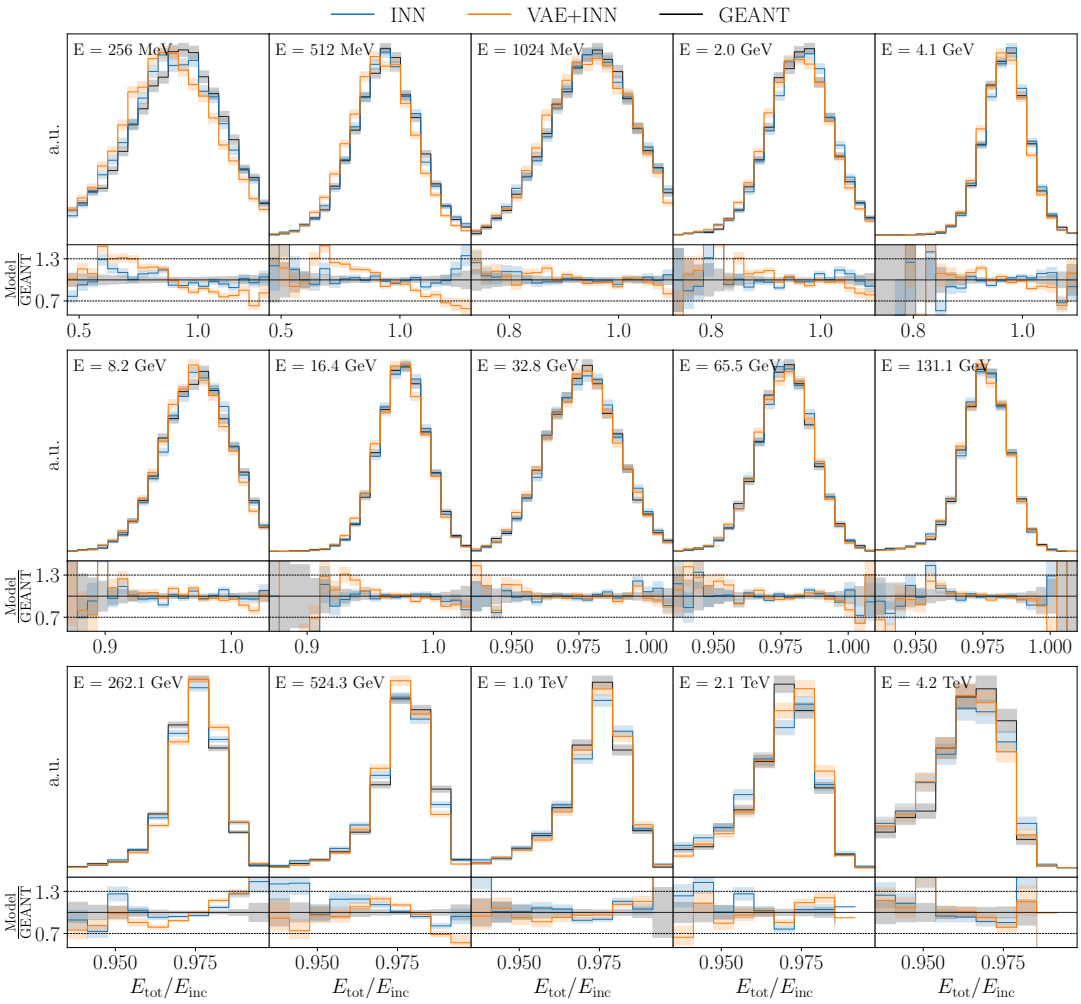

Figure 3: Energy ratio $E_{\text{tot}}/E_{\text{inc}}$ for each discrete incident energy, compared between GEANT4, INN, and VAE+INN for γ showers.

For the shower shapes, both networks show small, percent-level deviations in the bulk of the distributions at larger energies, while a larger discrepancy is found in the low-energy regime. In addition, the VAE+INN is smearing out secondary peaks of the distributions. Both networks generate slightly too wide showers. This effect is evident only at $E_{\text{inc}} = 256$ MeV and the more physically interesting energy region is modeled within statistical uncertainties for the INN, besides very narrow showers with width of the center of energy close to zero.

In the energy distributions we see the benefit of a smaller latent space. The energy variables of the VAE+INN are modeled within statistical uncertainties in layer-12 and layer-13 while the INN shows a few bins with deviations up to 30%. Additionally, the sharp cut at low energy is smeared to a different extent by the networks. Finally, we show global sparsity features and the $E_{\text{tot}}/E_{\text{inc}}$ ratio. As before the VAE+INN is unable to capture the correct number of active voxels. From the ratio $E_{\text{tot}}/E_{\text{inc}}$ we see that at all energies the fraction of deposited energy can be very different from shower to shower, leading to the wide energy distribution far from one. We collect in App. A.3 a set of inclusive histograms and the $E_{\text{tot}}/E_{\text{inc}}$ ratio for single incident energies.

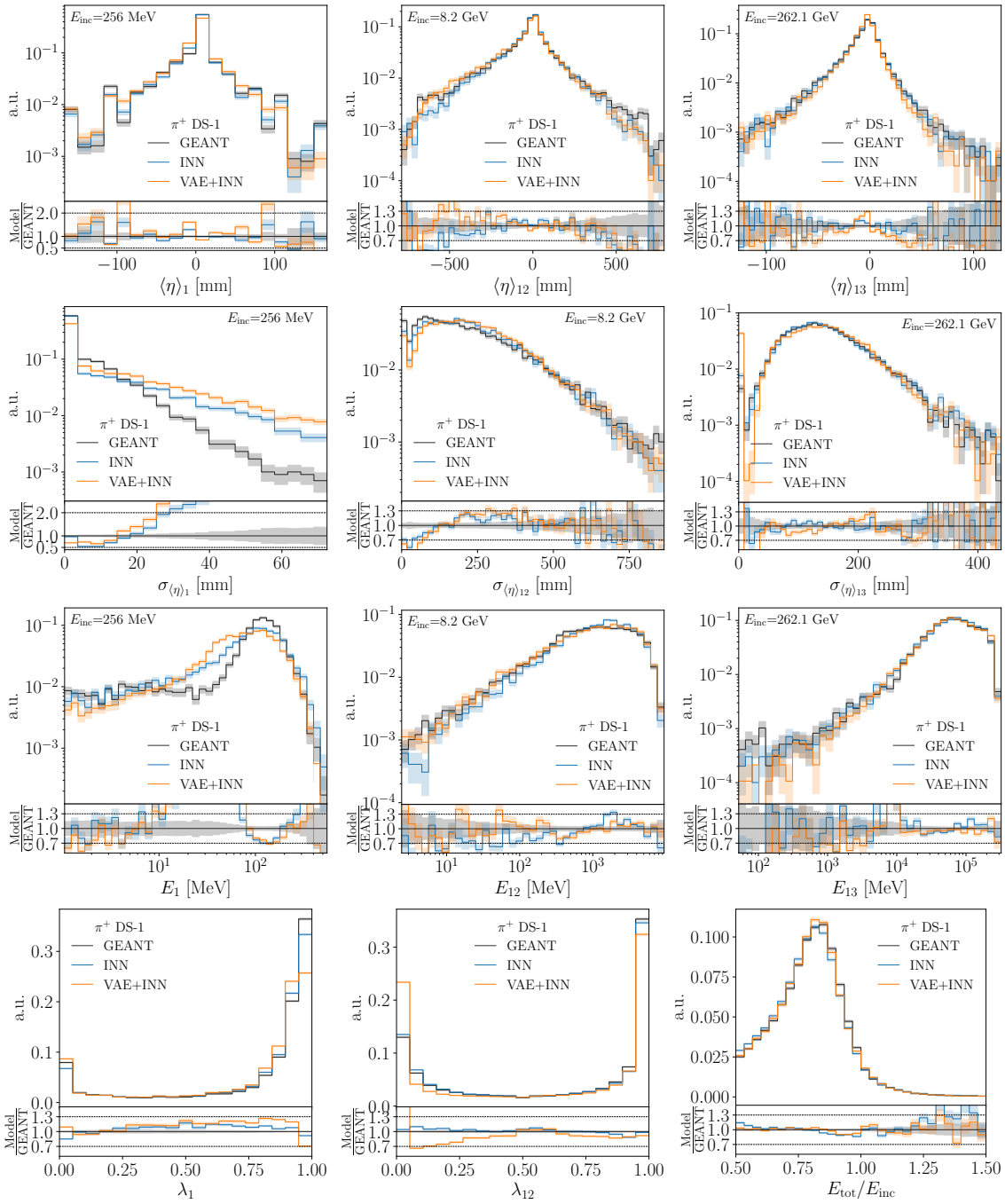

Figure 4: Set of high-level features for pion showers in dataset 1, compared between GEANT4, INN, and VAE+INN. We show the energy deposition, the center of energy, and the width of the center of energy in layer-1, layer-12, and layer-13. For each layer, we show a single incident energy. The last row contains the inclusive sparsity in layer-1 and layer-12, and the inclusive energy ratio E_{tot}/E_{inc}.

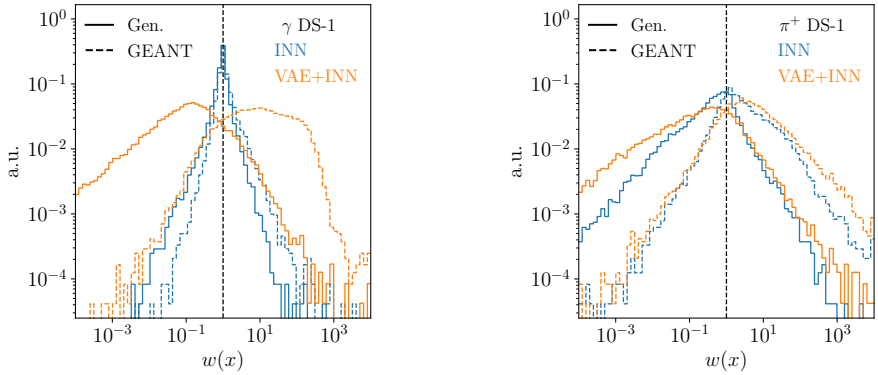

Figure 5: Classifier weight distributions in dataset 1. Classifier trained on low-level features for γ showers (left) and π showers (right).

Low-level classifier

To evaluate the performance of our generative networks on dataset 1 systematically, we train a network to learn the classifier weights defined in Eq.(10) over the voxel space. In the left panel of Fig. 5 we show the weights for the γ-shower. We clearly see that the INN outperforms the VAE+INN. Its weight distribution peaks much closer to 1 and the corresponding AUC of 0.603(2) is substantially better than the corresponding AUC of 0.937(2) of the VAE+INN.

More importantly, the INN does not show significant tails at large or small weights, which would indicate distinct failure modes. The peak of the VAE+INN, on the other hand, has moved away from 1. The tail at small weights indicates regions that are overpopulated by the network. We already know that this is the case for the sparsity. Large weights appear in phase space regions which the VAE+INN fails to populate, for instance the widths of the centers of energy.

In the right panel of Fig. 5 we see that the two generators perform more similar for π-showers. Both networks now show tails at small and large weights, two orders of magnitude away from one. This means there are regions that are over- and underpopulated by the generative networks. The fact that small weights appear for generated showers and large weights appear for the training data is generally expected. The AUCs for the INN and the VAE+INN are 0.804(2) and 0.864(2), respectively. The INN weight distribution is sharper around one, resulting in the smaller AUC compared to the combined VAE+INN approach. Altogether, we find that for dataset 1 with its limited dimensionality of 368 voxels for photons and 533 voxels for pions the INN works well, and that adding a VAE hinders the generative model because of a more complex latent space and a limited reconstruction quality of the autoencoder.

4.3 Dataset 2 electrons

Dataset 2 is given in terms of 6480 voxels, the kind of dimensionality which will probe the limitations of the regular INN. The number of parameters for this network approaches 200M. The question will be, if the VAE+INN condensation helps the performance of the network. As before, we show a representative set of high-level features in Fig. 6. This time we group the showers in three equally spaced energy windows in $\log E_{\text{inc}}$. For brevity, we only focus on three different layers representative for the interaction of the incident particle with the detector in each energy window. We include all the remaining histograms together with the published samples. We choose layer 1 for $E_{\text{inc}} \in [10^0, 10^1]$ GeV, layer-10 for $E_{\text{inc}} \in [10^1, 10^2]$ GeV, and layer-20 for showers with incident energy in $E_{\text{inc}} \in [10^2, 10^3]$ GeV.

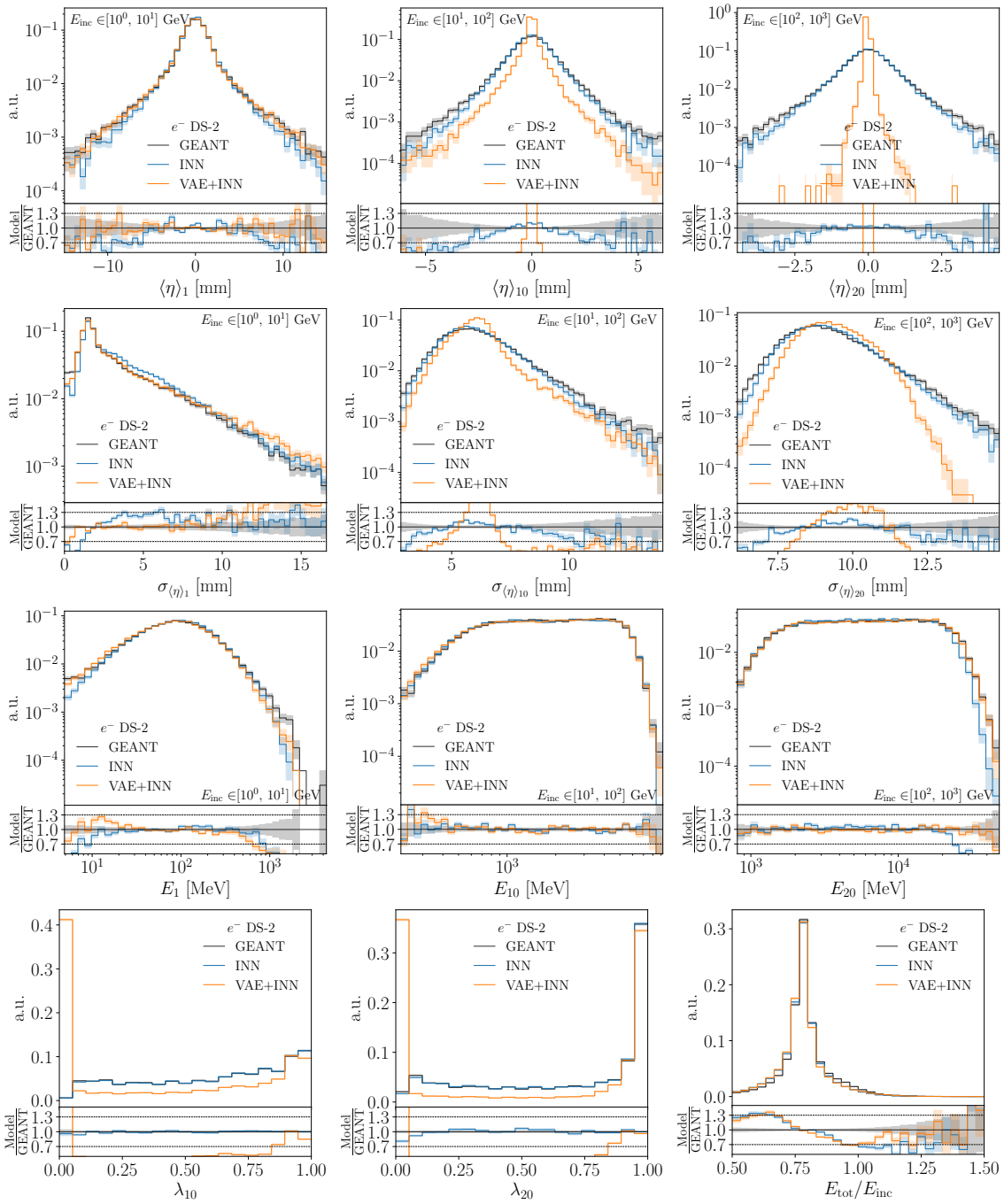

Figure 6: Set of high-level features for electron showers in dataset 2, compared between GEANT4, INN, and VAE+INN. We show the energy deposition, the center of energy, and the width of the center of energy in layer-1, layer-10, and layer-20. For each layer, we show a single incident energy range. The last row contains the inclusive sparsity in layer-10 and layer-20, and the inclusive energy ratio $E_{\text{tot}}/E_{\text{inc}}$.

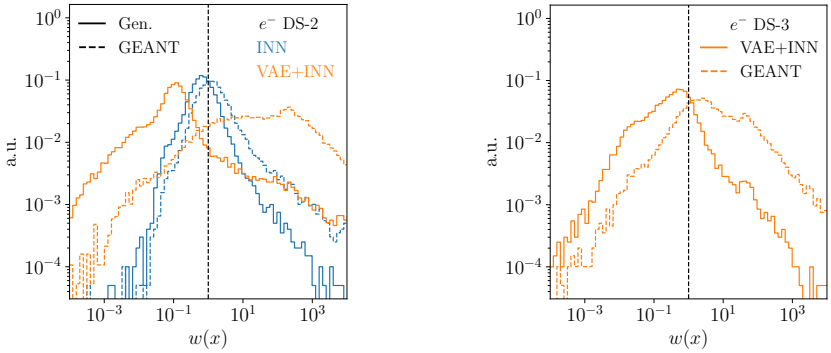

Figure 7: Classifier weight distributions. Classifier trained on e^- showers on dataset 2 (left) and dataset 3 (right). The tails of dataset 3 should be taken with a grain of salt, giving the limitations of the simple classifier architecture.

From the shower shapes we see that the INN-based architecture generates realistic showers at all energies. The training is stable and consistent across different runs of the same architecture. We observe agreement in the center of energy distributions with small deviations towards larger $\langle\eta\rangle$ and less commonly around zero, as shown in layer-10. The agreement in phase space density between GEANT4 and the INN ranges from a few percent in the bulk of the distributions to 50% in the tails, where very low statistics is available. Similar numbers apply to the width of the center of energy a shift towards wider showers is observed in all the energy windows. The failure mode of the INN, regardless of the dataset, is an under-sampling of showers with width between the peak at zero and the secondary peak, for which the location depends on the layer but not on the incident energy.

The VAE+INN shows limitations for large incident energies. As the energy increases, the generated showers only reproduce the mean value of the center of energy, i.e. showers have a rather uniform energy distribution around the center of the calorimeter resulting in $\langle\eta\rangle$ peaked around zero. A similar failure mode is observed in the width of the center of energy where the VAE+INN is more concentrated around the mean width of the GEANT4 showers. On the other hand, the compression mechanism of the VAE works well at lower energies where showers are generated by the latent INN and reconstructed by the VAE within statistical precision besides the $\sigma_{\langle\eta\rangle}$ region close to zero.

The two networks learn the energy depositions in the layers in two very different spaces. The INN extracts them with a large number of voxels, while the VAE+INN compresses them into a reduced space of around 50 features. This different dimensionality is reflected in all energy distributions in Fig. 6. While in poorly populated tails the INN does slightly better, the VAE+INN performs better for the main features in the central and high-energy regime. This is true for the layer-wise energies, but also for the ratio $E_{\text{tot}}/E_{\text{inc}}$. The last two plots show the inclusive sparsity distributions in layer-10 and layer-20 confirming that the INN reproduces the sparsity across the entire phase space while the VAE+INN struggles from the decoding step.

Low-level classifier

Again, we show a systematic comparison for dataset 2 in terms of the classifier weights in the left panel Fig. 7. Compared to dataset 1, there is a clear deterioration of the INN performance for the higher-dimensional phase space. At small weights, the tail remains narrow, indicating that there are still no phase space regions where the network over-samples the true phase space distribution. For large weights the weight tail now extends to values larger than $w \sim 10^3$. This tail can be related to a recurrent under-sampling of showers with a small width of the center of energy in each layer, as seen in Fig. 6.

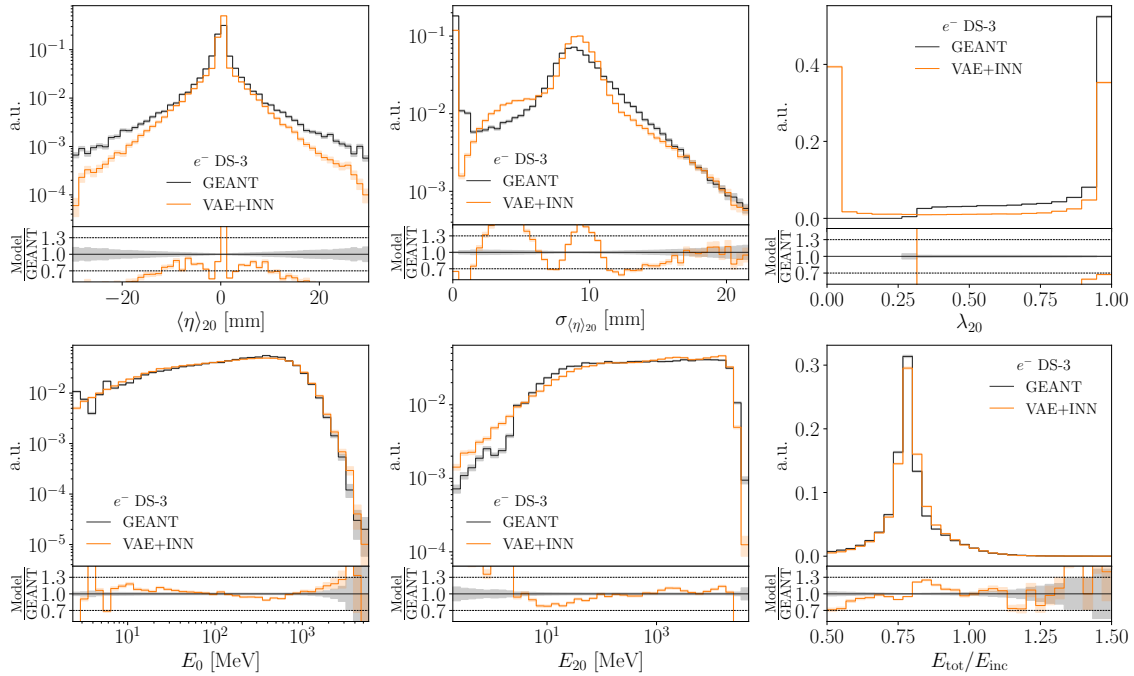

Figure 8: Set of high-level features for electron showers in dataset 3, compared between GEANT4, INN, and VAE+INN.

The classifier evaluating the VAE+INN generator highlights a few important structures as well. First, we have a clear over-sampled region in phase space with weights $w \sim 10^{-2}$, which we can relate to the center of energy distribution as well. As mentioned before, the VAE+INN over-samples showers with width close to the mean shower width. The classifier confirms this major failure mode. For the large-weight tail we checked that the under-sampled showers do not cluster in the same way, but are distributed over phase space, including tails of distributions.

The AUC values of the classifiers for dataset 2, 0.705(5) for the INN and 0.916(3) for the VAE+INN, confirm the challenge of the INN related to the size of the model and the dimensionality of dataset-2, especially relative to the well-modeled γ-showers in dataset 1. Adding a VAE improves the generation of low-energetic showers, where the low activity inside the calorimeter can be nicely compressed in the latent space, however this is out-weighted by the failure modes observed at medium and high incident energies.

4.4 Dataset 3 electrons

Finally, we tackle dataset 3, which includes the same physics as dataset 2, but over a much higher-dimensional and extremely sparsely populated phase space. For this dataset we cannot train an INN without dimensionality reduction, so we only show VAE+INN results in Fig. 8. As expected, the performance is worse than for dataset 2, but the training is stable across different training runs. One problem is a worsening reconstructions of the centroids and widths for the later layers, which is likely related to the small average energy deposition per voxel. The maximum in the width distributions is overpopulated by the VAE+INN. For the energy distributions the VAE+INN is doing reasonably well, with serious deviations only in the low-energy tails.

The classifier weights for the VAE+INN generating dataset 3 are shown in the right panel of Fig. 7. Even though the generative task is considerably harder, the learned weight distribution broadens centrally, but shows smaller tails than for dataset 2. The reason is that not only the

Table 2: Per-shower generation times in ms. We show mean and standard deviation of 10 independent runs. The star indicates that only 10k samples were generated. The CPU timings were done with an Intel(R) Core(TM) i9-7900X at 3.30 GHz, the GPU timings with an NVIDIA TITAN V with 12GB RAM.

	Batch size	INN			
		1-photon	1-pion	2-electron	3-electron
GPU	1	24.79 ± 0.49	24.76 ± 0.35	50.90 ± 0.37	
	100	0.385 ± 0.002	0.406 ± 0.003	1.900 ± 0.026	
	10000	0.162 ± 0.002	0.191 ± 0.006	exceeding memory	
CPU	1	17.48 ± 0.09	18.88 ± 0.33	117.5 ± 1.8	
	100	0.827 ± 0.028	1.004 ± 0.047	14.26 ± 0.18	
	10000	0.510 ± 0.008	0.719 ± 0.016	15.24 ± 1.36	
	Batch size	VAE+INN			
		1-photon	1-pion	2-electron	3-electron
GPU	1	33.64 ± 0.32	33.54 ± 0.23	40.55 ± 0.40	43.13 ± 1.4 *
	100	0.507 ± 0.005	0.544 ± 0.007	1.05 ± 0.02	3.44 ± 0.04
	10000	0.180 ± 0.002	0.228 ± 0.003	0.748 ± 0.018	—
CPU	1	20.83 ± 0.72	20.05 ± 0.13	28.11 ± 0.15	39.46 ± 1.1 *
	100	0.582 ± 0.005	0.886 ± 0.015	1.94 ± 0.01	4.91 ± 0.01
	10000	0.328 ± 0.004	0.426 ± 0.014	1.25 ± 0.01	4.97 ± 0.08

generative network, but also our simple classifier are reaching their limits. However, the bulk of the classifier weight distribution clearly indicates that for dataset 3 the phase space density is mis-modelled by factors as large as 100 or 0.01 over large phase space regions. While the INN description of these position showers fails altogether, the VAE+INN results do not guarantee the level of precision we would expect for generative networks at the LHC. For completeness, we include the high-level distributions divided in windows of incident energies only with the published samples [122].

4.5 Comparison

Finally, we compare our INN to other networks in two main aspects, the generation timing and the shower fidelity as measured by the AUC of a classifier. First, we do an in-depth timing study of our networks using the CaloChallenge [104] procedure. The INN architecture with modern coupling layers is ideally suited for fast and precise generation. We create a singularity container [123] of the software environment and take the time it takes to load the container, load the network, move it on the GPU, generate the samples, and save them to disk. In Tab. 2 we show the averaged results from ten runs. We observe a speed–up for increased batch size and when running on the GPU. The INN has a small advantage for dataset 1, but is unable to generate dataset 2 with the highest batch size and dataset 3 altogether. The VAE shows generation times at or below the millisecond mark.

The training time for the DS1 network on a single A30 GPU is $\sim$ 4 hours, including the validation steps which slightly increased the training time due to the large number of validation figures. However, we are unable to provide an exact number because of large fluctuations in the training time coming from the shared CPU and GPU memory of the cluster. Under the same observations, the network for dataset-2 trained on average for $\sim$ 6 hours.

The generation time from different published networks can vary because of varying hardware, making a fair comparison laborious. To avoid generating samples from different networks, we base our timing comparison on the result of the CaloChallenge [104]. We look at

two well-known architectures based on autoregressive normalizing flows, CaloFlow teacher and student [51], and the diffusion model CaloDiffusion [60], which provides benchmark results for this class of neural networks. From Tab. 3, we observe that the coupling block structure provides a generation speed-up when compared to the autoregressive counterpart, as studied in the CaloFlow architecture. As expected, our model is also substantially faster than a diffusion model due to the additional function evaluation needed to revert the diffusion process.

We summarize the shower fidelity results in Tab. 4. Our figure of merits are a classifiers trained on all the voxels, i.e. low-level features, and a second one trained on high-level observables. These include the layer energy deposition, the center of energy and the width of the center of energy in both (η, ϕ) directions, and the incident energy. From the AUC score, our photon showers on dataset-1 show the best performance on the low-level feature, even when compared to current diffusion networks, and high-level features comparable to CaloFlow. For pions, the complex low-level shower structure is better captured by CaloDiffusion, while flow networks still retain good high-level shower quality. Lastly, the training challanges encountered while training on dataset-2 are also reflected on the ability of the CaloINN network to generate high-dimensional electron showers.

Table 3: Generation time for networks trained on the full space. We compare our network CaloINN, CaloFlow teacher and student, and CaloDiffusion. Numbers are taken from the CaloChallenge [104] review.

	Generation time in ms per shower (CPU/GPU)		
	Batch size 1		
	1-photon	1-pion	2-electron
CaloINN	38(3) / 25(2)	43(3) / 25(2)	$3.9(3)\cdot10^2$ / 53(1)
CaloFlow teacher [51]	$4.3(3)\cdot10^4$ / $4.2(1)\cdot10^3$	$2.0(3)\cdot10^5$ / $6.2(1)\cdot10^3$	—
CaloFlow student [51]	$5.7(2)\cdot10^2$ / 56.9(5)	$6.2(2)\cdot10^2$ / 77(4)	—
CaloDiffusion [60]	$1.57(6)\cdot10^4$ / $5.59(6)\cdot10^3$	$1.5(1)\cdot10^4$ / $5.67(5)\cdot10^3$	$3.6(2)\cdot10^4$ / $5.29(8)\cdot10^3$
	Batch size 100		
	1-photon	1-pion	2-electron
CaloINN	2.7(3) / 0.51(3)	3.9(4) / 0.44(1)	60(10) / 1.18(3)
CaloFlow teacher [51]	$2.0(1)\cdot10^3$ / 45(1)	$5.4(5)\cdot10^3$ / 70(1)	—
CaloFlow student [51]	11(1) / 0.79(1)	14(2) / 1.00(2)	—
CaloDiffusion [60]	$4.6(3)\cdot10^3$ / 75(2)	$1.57(6)\cdot10^4$ / 77(2)	$2.3(3)\cdot10^4$ / 99(2)

Table 4: Summary of low-level (LL) and high-level (HL) classifier scores for our networks, where errors are extracted from 10 different classifiers. We compare to CaloFlow [51] and CaloDiffusion [60]. An in-depth comparison between different architectures, including latent models, is part of [104].

	AUC (LL/HL)		
	1-photon	1-pion	2-electron
CaloINN	0.603(2) / 0.563(3)	0.804(2) / 0.692(1)	0.705(5) / 0.891(2)
CaloFlow teacher [51]	0.701(3) / 0.551(3)	0.827(3) / 0.692(2)	—
CaloDiffusion [60]	0.62(1) / 0.62(1)	0.65(1) / 0.65(1)	0.56(1) / 0.56(1)

5 Conclusions

Simulations are at the heart of the LHC program. Modern generative networks are showing great promise to improve their quality and speed, allowing them to meet the requirements of the high-luminosity LHC. In this paper, we have studied fast and precise normalizing flows, specifically an INN and a VAE+INN combination to generate calorimeter showers in high-dimensional phase spaces. As reference datasets we use the CaloChallenge datasets 1 to 3, with an increasing number of 368, 533, 6480, and 40,500 voxels.

dataset 1

For the simplest dataset 1 photon showers, we have found that the INN generated high-fidelity showers and learns the phase space density of high-level features at the 10% level, except for failure modes which we can identify using high-level features and classifier weights over the low-level phase space. We have found that the INN provides unmatched speed with, for instance, $\mathcal{O}(10)$ ms generation time on a single CPU for a single shower. At the same time the shower quality is comparable or even better than other deep generative networks, including diffusion models.

For the pions in dataset 1 the INN faces more serious challenges, including mis-modeled features, and a wider range of learned classifier weights. The performance difference between the INN and the VAE+INN is limited by the expressivity of the latent network, with the additional sparsity failure mode introduced by the autoencoder. Also in this case our networks show shower accuracy comparable to other normalizing flows while providing faster generation time.

The electron showers in dataset 2 introduced technical challenges for the full-space INN. We have observed that the compression by the VAE+INN helps learning simple showers in the high-dimensional calorimeter. In particular, the main shape features of low-energetic showers are improved by the VAE+INN, including all the energy variables which are learned in a smaller latent space. However, the compression introduces more complex features for higher energies, where we observe a substantial deterioration of the VAE+INN. The second issue is introduced by the decoding step, which limits the reconstruction of low-energy depositions and the sparsity. The INN produces good-quality showers across the entire phase space. Although the physics is similar to the photons in dataset-1, we have not matched the same shower quality. This is attributed to difficulties in the optimization task of a much larger INN. In this paper we focused on the improvements provided by an INN for fast detector simulation and we believe other architectures can also benefit from our observations. For instance, a super-resolution approach, as in [63], would show better scaling properties to higher-dimensional calorimeters while retaining the improvement proposed by our CaloINN. Additionally, having a separate network which learns only the energy variables will further increase the overall fidelity in terms of both layer energy deposition and shower shape observables.

Finally, the electrons in dataset 3 exceed the power of the plain INN, leaving us with the VAE+INN as the remaining option. As for dataset-2, we have observed an intrinsic limitation of the latent INN to learn the compressed features which we partially addressed by moving to a kernel-based autoencoder architecture. While still providing fast generation, the VAE+INN is not able to generate high-fidelity showers. Although the expressivity of the network can be improved, as it has been done in [98], generating the correct sparsity remains an open question for latent models.

The generated samples used in this paper are published on Zenodo at 10.5281/zenodo.14178546. We also include the complete set of high-level features for the studied incident energies and the inclusive results.

Acknowledgments

We would like to thank Theo Heimel, Stefan Radev and Peter Loch for helpful discussions. We would like to thank Thorsten Buss for collaborating in an early phase of the project.

Funding information We would like to thank the Baden-Württemberg Stiftung for financing through the program *Internationale Spitzenforschung*, project *Uncertainties – Teaching AI its Limits* (BWST_ISF2020-010). DS is supported by the U.S. Department of Energy under Award Number DOE-SC0010008. This research is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 – TRR 257: *Particle Physics Phenomenology after the Higgs Discovery* and through Germany's Excellence Strategy EXC 2181/1 – 390900948 (the *Heidelberg STRUCTURES Excellence Cluster*).

A Appendices

A.1 Network details

In this appendix we give details on the network architectures and the preprocessing. The INN and the VAE+INN take voxels normalized by the layer energy as input. The extra energy dimensions, calculated as in Eq. 5, are appended to the feature vector.

In the INN, we apply uniform noise and and a regularized logarithmic transformation with strength α. The transformation applied to the features is a rational quadratic spline [116] for dataset 1 and a cubic spline [117] for dataset 2. The prediction of the spline parameters is obtained with an MLP sub-network with 256 nodes for each hidden layer. To equally learn each dimension, we permute the order of the features after a transformation and normalize the output to mean zero and unit standard deviation with an ActNorm [107] layer. In the large-scale architecture, we stack twelve blocks to construct the INN with the additional preprocessing block.

The VAE preprocessing has a similar structure. After normalization, we apply an α-regularized logit transformation and a normalization to zero mean and unit standard deviation to each feature. We do not add noise during training and we set the latent dimension to 50 for dataset 1 and 2, and to 300 for dataset 3. We provide the full list of parameters in Tabs. 5 and 6.

The selection of the hyperparameters is based on heuristic observations from which we performed a rather limited grid search around the starting set. For instance, the hidden dimension of the layers is chosen to be similar to the number of input features used to predict the spline parameters and the selection of the number of layers is based on previous experience on INNs [69, 124]. While RQS are currently one of the most expressive transformations, we found occasional run-to-run instabilities while training on DS2 while the cubic spline consistently converged. A change in batch size is accompanied by a change in the number of epochs, such that the number of iterations is approximately constant,

The initial conditions for the number of bins and the number of blocks is based on the number of parameters used in [44], which already showed great generation performance. Additionally we tested $b \in \{1 \cdot 10^{-6}, 5 \cdot 10^{-6}\, 1 \cdot 10^{-5}\}$, a batch size $\in \{64, 128, 256, 512\}$, and change the number of blocks by two units.
For the VAE we additionally varied the latent space by a factor of 2 up and down without any visible improvements. The size of the embedding and decoding networks was not the goal of a larger optimization. It turned out beneficial to inflate the dimensionality in the first layer in all tried configurations. Apart from that we found generally better results the larger the

Table 5: Network and training parameters for the pure INN.

Parameter	INN DS1/DS2	INN (with VAE)
coupling blocks	RQS / Cubic	RQS
# layers	4 / 3	3
hidden dimension	256	32
# of bins	10	10
# of blocks	12/14	18
# of epochs	450 / 200	200
batch size	512 / 256	256
lr scheduler	one cycle	one cycle
max. lr	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
$\beta_{1,2}$ (ADAM)	$(0.9, 0.999)$	$(0.9, 0.999)$
b	$5 \cdot 10^{-6}$	/
α	$1 \cdot 10^{-8}$	$1 \cdot 10^{-6}$

Table 6: Network and training parameters for the VAE-INN.

Parameter	VAE	
lr scheduler	Constant LR	Inner VAE
lr	$1 \cdot 10^{-4}$	
hidden dimension	5000, 1000, 500 (Set 1) 1500, 1000, 500 (Set 2) 2000, 1000, 600 (Set 3)	
latent dimension	50 (Set 1,2) / 300 (Set 3)	
# of epochs	1000	
batch size	256	
β	$1 \cdot 10^{-9}$	
threshold t [keV]	2 (Set 1) / 15.15 (Set 2,3)	
hidden dimension	1500, 800, 300	Kernel
kernel size	7	
kernel stride	3 (Set 2), 5 (Set 3)	

encoder and decoder networks were constructed. So we chose the parameter number based on the available GPU RAM.
We avoided large resources consumption which would be required for a finer ablation study. Therefore, we expect that the results can be improved in terms of timings, network complexity, and performance.

The classifiers trained for the evaluation of the generative networks are simple MLP networks with leaky ReLU. We use three layers with 512 nodes each and a batch size of 1000. The network is trained for 200 epochs with a learning rate of $2 \cdot 10^{-4}$ and the Adam optimizer with standard parameters. To prevent overfitting, especially for the larger datasets, we apply 30% dropout to each layer, and we reduce the learning rate on plateau with a decay factor of 0.1 and decay patience of 10. The splitting between training, validation, and testing is 60/20/20%. The selection of the best network is based on the best validation loss.

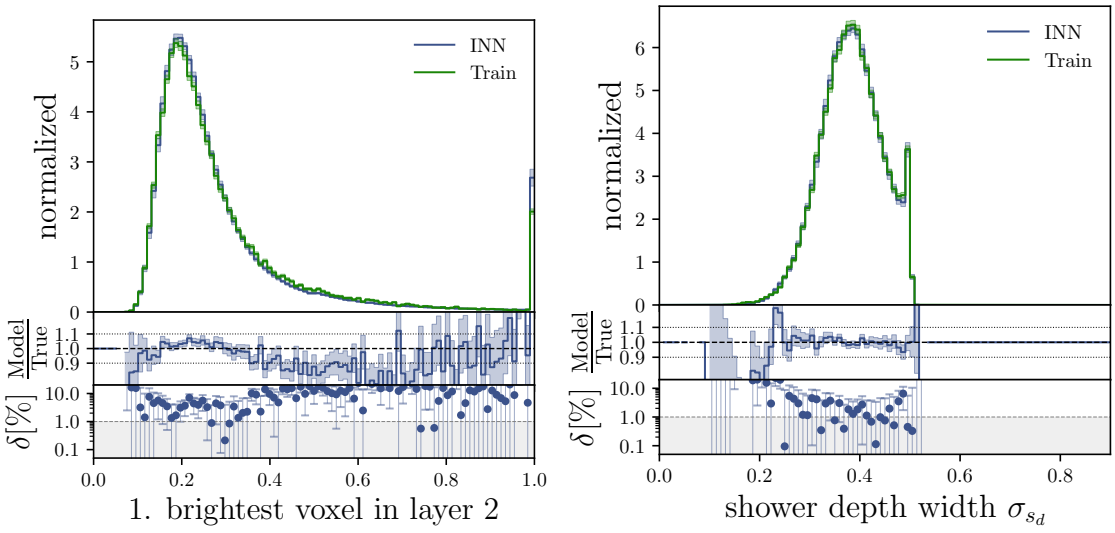

Figure 9: Comparison between CaloINN and GEANT4 on two high level features. Brightest voxel distribution in layer-1 (left), and width of the shower depth (bottom). Error bars on the INN are calculated after sampling from the Bayesian network $N = 50$ times.

A.2 CaloGAN dataset

In this section we discuss the INN performance on the even simpler CaloGAN dataset [35,37]. The INN architecture is described in Sec. 3. To extract uncertainties from the generative network, we promote the deterministic INN to its Bayesian counterpart [69,124]. The implementation follows the variational approximation substituting the linear layer with a mixture of uncorrelated Gaussians with learnable means and a diagonal covariance matrix. In practice, we only upgrade the last layer of each sub-network to a Bayesian layer [125].

Figure 9 showcases two high–level features as examples of the performance of the CaloINN as compared to the training data distribution generated by GEANT4. We show the brightest voxel distribution in layer 0 and the width of the shower depth width defined as the standard deviation of s_d [44], with

$$s_d = \frac{\sum_{k=0}^{2} k E_k}{\sum_{k=0}^{2} E_k} \,. \tag{A.1}$$

The error bars in the GEANT4 distribution are the statistical errors while for the INN we estimate the uncertainties by sampling $N = 50$ times from the network and resampling the network parameters each time.

To evaluate our networks on low-level observables, we resort again to classifier-based metrics. As already studied in a previous work [75], the INN samples are indistinguishable from the GEANT4 counterpart besides a few specific phase-space regions. We train a classifier on the CaloFlow samples and find a large tail towards small weights. From clustering of the tail, we observe a clear dependence on the energy deposition total energy deposition. We link this effect to the learned energy variable $u_2 = E_1/(E_1 + E_2)$ and the noise injection procedure. If the noise is added at voxel-level, before calculating the additional energy variables, the flow learns distorted energy ratio distributions. Especially in the last layer, where the average energy deposition is smaller, this effect is larger. We summarize this effect in Fig. 10. We also provide the AUCs and the generation timings in Tab. 7.

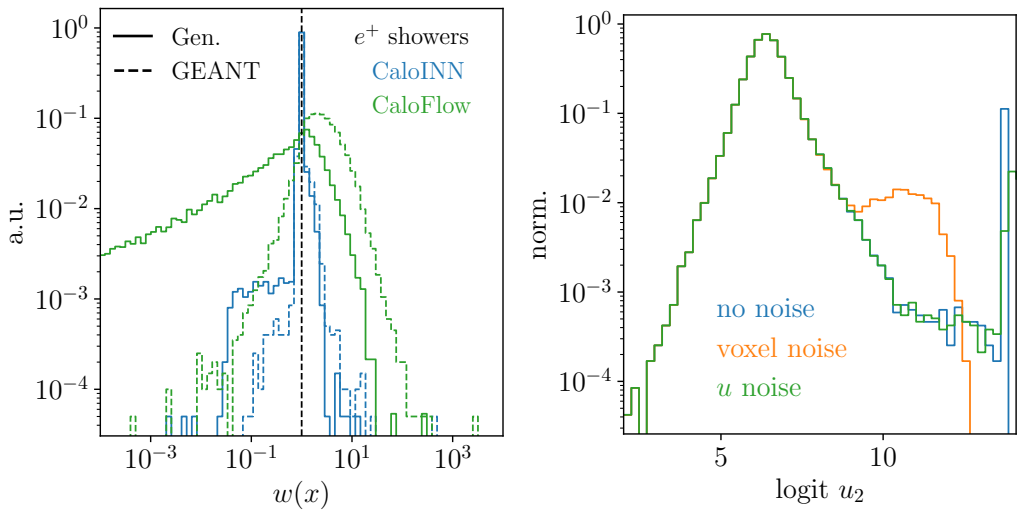

Figure 10: (left) Weight distribution of CaloFlow and CaloINN for e^+ showers. (right) u_2 distribution with different noise injections.

Table 7: (left) AUC of the two classifiers trained on the CaloFlow teacher and CaloINN samples. (right) Per shower generation timings in ms. We show mean and standard deviation of 10 independent runs of generating 100k showers. The star indicates that only 10k samples were generated in total.

AUC		CaloFlow [44]	CaloINN
e^+	unnorm.	0.859(10)	0.525(2)
	norm.	0.870(2)	0.598(3)
	hlf	0.795(1)	0.656(2)
γ	unnorm.	0.756(50)	0.530(2)
	norm.	0.796(2)	0.584(2)
	hlf	0.727(2)	0.671(2)
π^+	unnorm.	0.649(3)	0.662(2)
	norm.	0.755(3)	0.735(4)
	hlf	0.888(1)	0.786(4)

	Batch size	CaloFlow [45]	CaloINN
GPU	1	$55.12 \pm 0.19^*$	$23.79 \pm 0.10^*$
	100	0.744 ± 0.04	0.425 ± 0.005
	10000	0.249 ± 0.003	0.211 ± 0.003
CPU	1	$119.9 \pm 0.9^*$	$46.39 \pm 3.18^*$
	100	3.13 ± 0.11	1.14 ± 0.03
	10000	1.681 ± 0.004	0.72 ± 0.01

A.3 Additional histograms

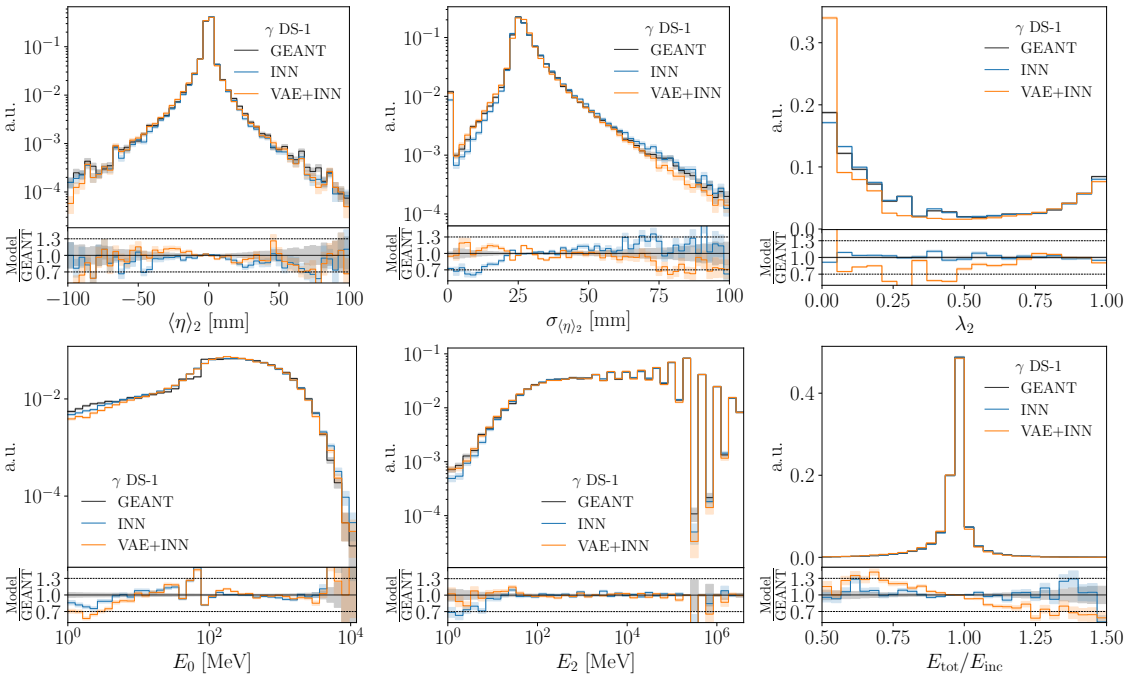

Figure 11: Set of high-level features for γ showers in dataset 1 inclusive in E_{inc}, compared between GEANT4, INN, and VAE+INN.

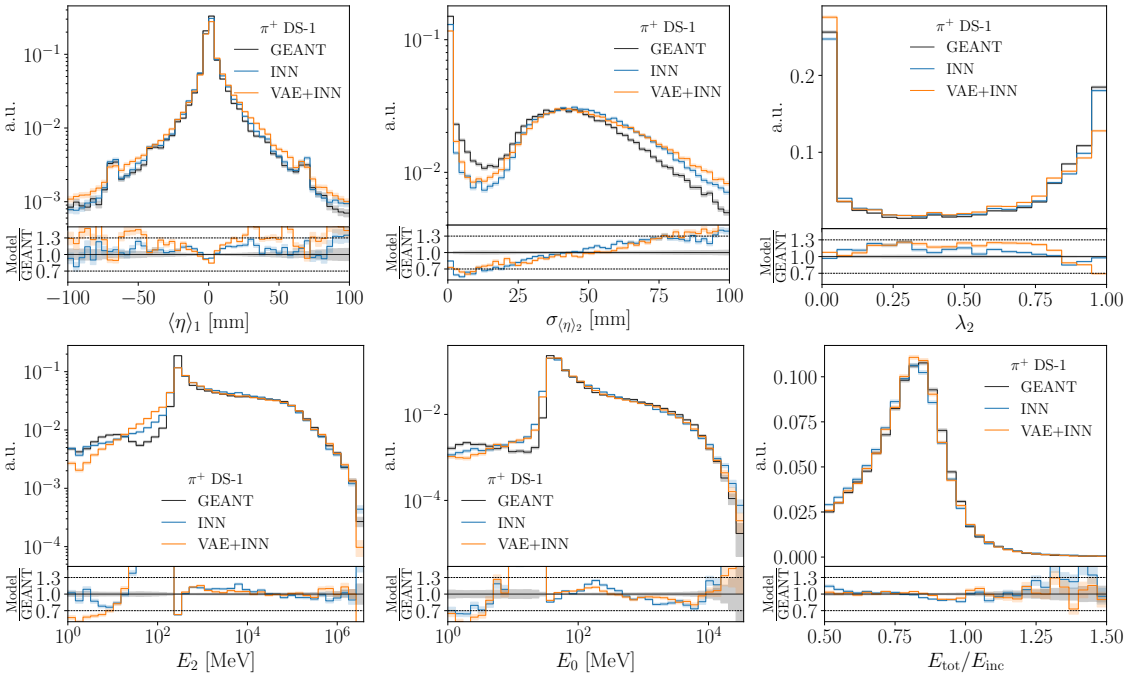

Figure 12: Set of high-level features for pion showers in dataset 1 inclusive in E_{inc}, compared between GEANT4, INN, and VAE+INN.

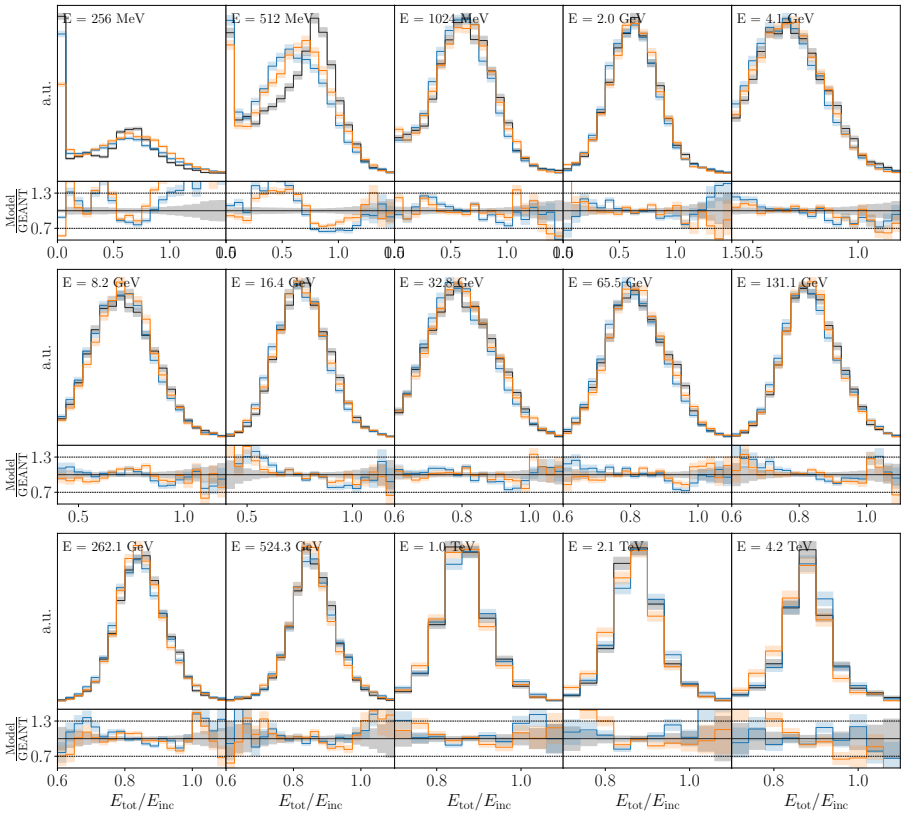

Figure 13: Energy ratio $E_{\text{tot}}/E_{\text{inc}}$ for each discrete incident energy, compared between GEANT4, INN, and VAE+INN for π^+ showers.

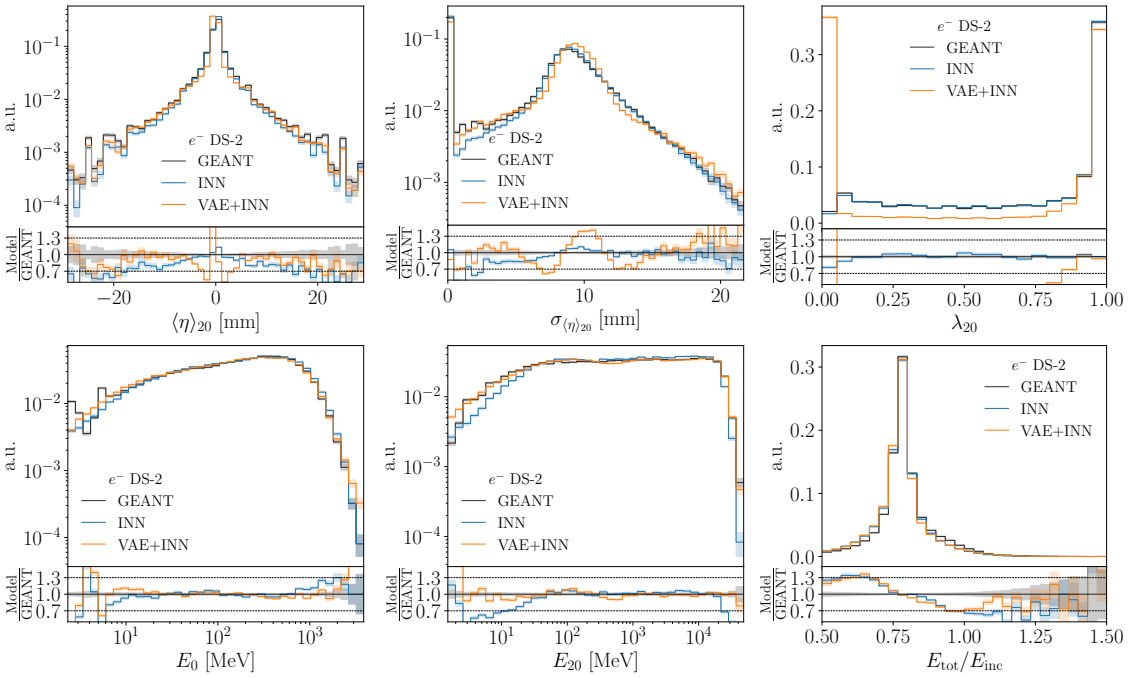

Figure 14: Set of high-level features for electron showers in dataset 2 inclusive in E_{inc}, compared between GEANT4, INN, and VAE+INN.

References

[1] J. M. Campbell et al., *Event generators for high-energy physics experiments*, SciPost Phys. **16**, 130 (2024), doi:10.21468/SciPostPhys.16.5.130.

[2] A. Butter et al., *Machine learning and LHC event generation*, SciPost Phys. **14**, 079 (2023), doi:10.21468/SciPostPhys.14.4.079.

[3] S. Agostinelli et al., *Geant4 – A simulation toolkit*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **506**, 250 (2003), doi:10.1016/S0168-9002(03)01368-8.

[4] J. Allison et al., *Geant4 developments and applications*, IEEE Trans. Nucl. Sci. **53**, 270 (2006), doi:10.1109/TNS.2006.869826.

[5] J. Allison et al., *Recent developments in Geant4*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **835**, 186 (2016), doi:10.1016/j.nima.2016.06.125.

[6] G. Aad et al., *The ATLAS simulation infrastructure*, Eur. Phys. J. C **70**, 823 (2010), doi:10.1140/epjc/s10052-010-1429-9.

[7] R. Rahmat, R. Kroeger and A. Giammanco, *The fast simulation of the CMS experiment*, J. Phys.: Conf. Ser. **396**, 062016 (2012), doi:10.1088/1742-6596/396/6/062016.

[8] G. Aad et al., *AtlFast3: The next generation of fast simulation in ATLAS*, Comput. Softw. Big Sci. **6**, 7 (2022), doi:10.1007/s41781-021-00079-7.

[9] T. Plehn, A. Butter, B. Dillon, T. Heimel, C. Krause and R. Winterhalder, *Modern machine learning for LHC physicists*, (arXiv preprint) doi:10.48550/arXiv.2211.01421.

[10] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, SciPost Phys. **10**, 139 (2021), doi:10.21468/SciPostPhys.10.6.139.

[11] S. Bieringer et al., *Calomplification – The power of generative calorimeter models*, J. Instrum. **17**, P09028 (2022), doi:10.1088/1748-0221/17/09/P09028.

[12] J. Bendavid, *Efficient Monte Carlo integration using boosted decision trees and generative deep neural networks*, (arXiv preprint) doi:10.48550/arXiv.1707.00028.

[13] M. Klimek and M. Perelstein, *Neural network-based approach to phase space integration*, SciPost Phys. **9**, 053 (2020), doi:10.21468/SciPostPhys.9.4.053.

[14] I.-K. Chen, M. Klimek and M. Perelstein, *Improved neural network Monte Carlo simulation*, SciPost Phys. **10**, 023 (2021), doi:10.21468/SciPostPhys.10.1.023.

[15] C. Gao, J. Isaacson and C. Krause, *i-flow: High-dimensional integration and sampling with normalizing flows*, Mach. Learn.: Sci. Technol. **1**, 045023 (2020), doi:10.1088/2632-2153/abab62.

[16] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale and S. Schumann, *Exploring phase space with neural importance sampling*, SciPost Phys. **8**, 069 (2020), doi:10.21468/SciPostPhys.8.4.069.

[17] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Event generation with normalizing flows*, Phys. Rev. D **101**, 076002 (2020), doi:10.1103/PhysRevD.101.076002.

[18] K. Danziger, T. Janßen, S. Schumann and F. Siegert, *Accelerating Monte Carlo event generation – Rejection sampling using neural network event-weight estimates*, SciPost Phys. **12**, 164 (2022), doi:10.21468/SciPostPhys.12.5.164.

[19] T. Heimel, R. Winterhalder, A. Butter, J. Isaacson, C. Krause, F. Maltoni, O. Mattelaer and T. Plehn, *MadNIS – Neural multi-channel importance sampling*, SciPost Phys. **15**, 141 (2023), doi:10.21468/SciPostPhys.15.4.141.

[20] T. Janßen, D. Maître, S. Schumann, F. Siegert and H. Truong, *Unweighting multijet event generation using factorisation-aware neural networks*, SciPost Phys. **15**, 107 (2023), doi:10.21468/SciPostPhys.15.3.107.

[21] E. Bothmann, T. Childers, W. Giele, F. Herren, S. Höche, J. Isaacson, M. Knobbe and R. Wang, *Efficient phase-space generation for hadron collider event simulation*, SciPost Phys. **15**, 169 (2023), doi:10.21468/SciPostPhys.15.4.169.

[22] T. Heimel, N. Huetsch, F. Maltoni, O. Mattelaer, T. Plehn and R. Winterhalder, *The MadNIS reloaded*, SciPost Phys. **17**, 023 (2024), doi:10.21468/SciPostPhys.17.1.023.

[23] L. de Oliveira, M. Paganini and B. Nachman, *Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis*, Comput. Softw. Big Sci. **1**, 4 (2017), doi:10.1007/s41781-017-0004-6.

[24] A. Andreassen, I. Feige, C. Frye and M. D. Schwartz, *JUNIPR: A framework for unsupervised machine learning in particle physics*, Eur. Phys. J. C **79**, 102 (2019), doi:10.1140/epjc/s10052-019-6607-9.

[25] E. Bothmann and L. Del Debbio, *Reweighting a parton shower using a neural network: The final-state case*, J. High Energy Phys. **01**, 033 (2019), doi:10.1007/JHEP01(2019)033.

[26] K. Dohi, *Variational autoencoders for jet simulation*, (arXiv preprint) doi:10.48550/arXiv.2009.04842.

[27] E. Buhmann, G. Kasieczka and J. Thaler, *EPiC-GAN: Equivariant point cloud generation for particle jets*, SciPost Phys. **15**, 130 (2023), doi:10.21468/SciPostPhys.15.4.130.

[28] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch and T. Golling, *PC-JeDi: Diffusion for particle cloud generation in high energy physics*, SciPost Phys. **16**, 018 (2024), doi:10.21468/SciPostPhys.16.1.018.

[29] V. Mikuni, B. Nachman and M. Pettee, *Fast point cloud generation with diffusion models in high energy physics*, Phys. Rev. D **108**, 036025 (2023), doi:10.1103/PhysRevD.108.036025.

[30] E. Buhmann et al., *EPiC-ly fast particle cloud generation with flow-matching and diffusion*, (arXiv preprint) doi:10.48550/arXiv.2310.00049.

[31] P. Ilten, T. Menzo, A. Youssef and J. Zupan, *Modeling hadronization using machine learning*, SciPost Phys. **14**, 027 (2023), doi:10.21468/SciPostPhys.14.3.027.

[32] A. Ghosh, X. Ju, B. Nachman and A. Siodmok, *Towards a deep learning model for hadronization*, Phys. Rev. D **106**, 096020 (2022), doi:10.1103/PhysRevD.106.096020.

[33] J. Chan, X. Ju, A. Kania, B. Nachman, V. Sangli and A. Siódmok, *Fitting a deep generative hadronization model*, J. High Energy Phys. **09**, 084 (2023), doi:10.1007/JHEP09(2023)084.

[34] C. Bierlich, P. Ilten, T. Menzo, S. Mrenna, M. Szewc, M. K. Wilkinson, A. Youssef and J. Zupan, *Towards a data-driven model of hadronization using normalizing flows*, SciPost Phys. **17**, 045 (2024), doi:10.21468/SciPostPhys.17.2.045.

[35] M. Paganini, L. de Oliveira and B. Nachman, *Accelerating science with generative adversarial networks: An application to 3D particle showers in multilayer calorimeters*, Phys. Rev. Lett. **120**, 042003 (2018), doi:10.1103/PhysRevLett.120.042003.

[36] L. de Oliveira, M. Paganini and B. Nachman, *Controlling physical attributes in GAN-accelerated simulation of electromagnetic calorimeters*, J. Phys.: Conf. Ser. **1085**, 042017 (2018), doi:10.1088/1742-6596/1085/4/042017.

[37] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, Phys. Rev. D **97**, 014021 (2018), doi:10.1103/PhysRevD.97.014021.

[38] M. Erdmann, L. Geiger, J. Glombitza and D. Schmidt, *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*, Comput. Softw. Big Sci. **2**, 4 (2018), doi:10.1007/s41781-018-0008-x.

[39] M. Erdmann, J. Glombitza and T. Quast, *Precise simulation of electromagnetic calorimeter showers using a Wasserstein generative adversarial network*, Comput. Softw. Big Sci. **3**, 4 (2019), doi:10.1007/s41781-018-0019-7.

[40] D. Belayneh et al., *Calorimetry with deep learning: Particle simulation and reconstruction for collider physics*, Eur. Phys. J. C **80**, 688 (2020), doi:10.1140/epjc/s10052-020-8251-9.

[41] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol and K. Krüger, *Getting high: High fidelity simulation of high granularity calorimeters with high speed*, Comput. Softw. Big Sci. **5**, 13 (2021), doi:10.1007/s41781-021-00056-0.

[42] ATLAS collaboration, *Fast simulation of the ATLAS calorimeter system with generative adversarial networks*, Tech. Rep. ATL-SOFT-PUB-2020-006, CERN, Geneva, Switzerland (2020), https://cds.cern.ch/record/2746032.

[43] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol and K. Krüger, *Decoding photons: Physics in the latent space of a BIB-AE generative network*, Europhys. J. Web Conf. **251**, 03003 (2021), doi:10.1051/epjconf/202125103003.

[44] C. Krause and D. Shih, *Fast and accurate simulations of calorimeter showers with normalizing flows*, Phys. Rev. D **107**, 113003 (2023), doi:10.1103/PhysRevD.107.113003.

[45] C. Krause and D. Shih, *Accelerating accurate simulations of calorimeter showers with normalizing flows and probability density distillation*, Phys. Rev. D **107**, 113004 (2023), doi:10.1103/PhysRevD.107.113004.

[46] E. Buhmann et al., *Hadrons, better, faster, stronger*, Mach. Learn.: Sci. Technol. **3**, 025014 (2022), doi:10.1088/2632-2153/ac7848.

[47] C. Chen, O. Cerri, T. Q. Nguyen, J. R. Vlimant and M. Pierini, *Analysis-specific fast simulation at the LHC with deep learning*, Comput. Softw. Big Sci. **5**, 15 (2021), doi:10.1007/s41781-021-00060-4.

[48] A. Adelmann et al., *New directions for surrogate models and differentiable programming for high energy physics detector simulation*, (arXiv preprint) doi:10.48550/arXiv.2203.08806.

[49] V. Mikuni and B. Nachman, *Score-based generative models for calorimeter shower simulation*, Phys. Rev. D **106**, 092009 (2022), doi:10.1103/PhysRevD.106.092009.

[50] ATLAS collaboration: G. Aad et al., *Deep generative models for fast photon shower simulation in ATLAS*, Comput. Softw. Big Sci. **8**, 7 (2024), doi:10.1007/s41781-023-00106-9.

[51] C. Krause, I. Pang and D. Shih, *CaloFlow for CaloChallenge dataset 1*, SciPost Phys. **16**, 126 (2024), doi:10.21468/SciPostPhys.16.5.126.

[52] J. C. Cresswell, B. L. Ross, G. Loaiza-Ganem, H. Reyes-Gonzalez, M. Letizia and A. L. Caterini, *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*, (arXiv preprint) doi:10.48550/arXiv.2211.15380.

[53] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, C. Krause, I. Shekhzadeh and D. Shih, *L2LFlows: Generating high-fidelity 3D calorimeter images*, J. Instrum. **18**, P10017 (2023), doi:10.1088/1748-0221/18/10/P10017.

[54] B. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn and T. Kuhr, *Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning*, Nat. Commun. **15**, 4916 (2024), doi:10.1038/s41467-024-49104-4.

[55] A. Xu, S. Han, X. Ju and H. Wang, *Generative machine learning for detector response modeling with a conditional normalizing flow*, J. Instrum. **19**, P02003 (2024), doi:10.1088/1748-0221/19/02/P02003.

[56] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger, P. McKeown and L. Rustige, *New angles on fast calorimeter shower simulation*, Mach. Learn.: Sci. Technol. **4**, 035044 (2023), doi:10.1088/2632-2153/acefa9.

[57] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasicezka, A. Korol, W. Korcari, K. Krüger and P. McKeown, *CaloClouds: Fast geometry-independent highly-granular calorimeter simulation*, J. Instrum. **18**, P11025 (2023), doi:10.1088/1748-0221/18/11/P11025.

[58] M. R. Buckley, I. Pang, D. Shih and C. Krause, *Inductive simulation of calorimeter showers with normalizing flows*, Phys. Rev. D **109**, 033006 (2024), doi:10.1103/PhysRevD.109.033006.

[59] V. Mikuni and B. Nachman, *CaloScore v2: Single-shot calorimeter shower simulation with diffusion models*, J. Instrum. **19**, P02001 (2024), doi:10.1088/1748-0221/19/02/P02001.

[60] O. Amram and K. Pedro, *Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation*, Phys. Rev. D **108**, 072014 (2023), doi:10.1103/PhysRevD.108.072014.

[61] S. Diefenbacher, V. Mikuni and B. Nachman, *Refining fast calorimeter simulations with a Schrödinger bridge*, (arXiv preprint) doi:10.48550/arXiv.2308.12339.

[62] M. F. Giannelli and R. Zhang, *CaloShowerGAN, a generative adversarial network model for fast calorimeter shower simulation*, Eur. Phys. J. Plus **139**, 597 (2024), doi:10.1140/epjp/s13360-024-05397-4.

[63] I. Pang, D. Shih and J. A. Raine, *Calorimeter shower superresolution*, Phys. Rev. D **109**, 092009 (2024), doi:10.1103/PhysRevD.109.092009.

[64] S. Otten, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen, D. Podareanu, R. Ruiz de Austri and R. Verheyen, *Event generation and statistical sampling for physics with deep generative models and a density information buffer*, Nat. Commun. **12**, 2985 (2021), doi:10.1038/s41467-021-22616-z.

[65] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini, *LHC analysis-specific datasets with generative adversarial networks*, (arXiv preprint) doi:10.48550/arXiv.1901.05282.

[66] R. Di Sipio, M. F. Giannelli, S. K. Haghighat and S. Palazzo, *DijetGAN: A generative-adversarial network approach for the simulation of QCD dijet events at the LHC*, J. High Energy Phys. **08**, 110 (2019), doi:10.1007/JHEP08(2019)110.

[67] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC events*, SciPost Phys. **7**, 075 (2019), doi:10.21468/SciPostPhys.7.6.075.

[68] Y. Alanazi et al., *Simulation of electron-proton scattering events by a feature-augmented and transformed generative adversarial network (FAT-GAN)*, in *Proceedings of the thirtieth international joint conference on artificial intelligence*, International Joint Conferences on Artificial Intelligence, Montreal, Canada, ISBN 9780999241196 (2021), doi:10.24963/ijcai.2021/293.

[69] A. Butter, T. Heimel, S. Hummerich, T. Krebs, T. Plehn, A. Rousselot and S. Vent, *Generative networks for precision enthusiasts*, SciPost Phys. **14**, 078 (2023), doi:10.21468/SciPostPhys.14.4.078.

[70] A. Butter, N. Huetsch, S. Palacios Schweitzer, T. Plehn, P. Sorrenson and J. Spinner, *Jet diffusion versus JetGPT – Modern networks for the LHC*, (arXiv preprint) doi:10.48550/arXiv.2305.10475.

[71] S. Diefenbacher, E. Eren, G. Kasieczka, A. Korol, B. Nachman and D. Shih, *DCTRGAN: Improving the precision of generative models with reweighting*, J. Instrum. **15**, P11004 (2020), doi:10.1088/1748-0221/15/11/P11004.

[72] R. Winterhalder, M. Bellagente and B. Nachman, *Latent space refinement for deep generative models*, (arXiv preprint) doi:10.48550/arXiv.2106.00792.

[73] B. Nachman and R. Winterhalder, *Elsa: Enhanced latent spaces for improved collider simulations*, Eur. Phys. J. C **83**, 843 (2023), doi:10.1140/epjc/s10052-023-11989-8.

[74] M. Leigh, D. Sengupta, J. A. Raine, G. Quétant and T. Golling, *Faster diffusion model with improved quality for particle cloud generation*, Phys. Rev. D **109**, 012010 (2024), doi:10.1103/PhysRevD.109.012010.

[75] R. Das, L. Favaro, T. Heimel, C. Krause, T. Plehn and D. Shih, *How to understand limitations of generative networks*, SciPost Phys. **16**, 031 (2024), doi:10.21468/SciPostPhys.16.1.031.

[76] S. Bieringer, A. Butter, T. Heimel, S. Höche, U. Köthe, T. Plehn and S. T. Radev, *Measuring QCD splittings with invertible networks*, SciPost Phys. **10**, 126 (2021), doi:10.21468/SciPostPhys.10.6.126.

[77] A. Butter, T. Heimel, T. Martini, S. Peitzsch and T. Plehn, *Two invertible networks for the matrix element method*, SciPost Phys. **15**, 094 (2023), doi:10.21468/SciPostPhys.15.3.094.

[78] T. Heimel, N. Huetsch, R. Winterhalder, T. Plehn and A. Butter, *Precision-machine learning for the matrix element method*, SciPost Phys. **17**, 129 (2024), doi:10.21468/SciPostPhys.17.5.129.

[79] K. Datta, D. Kar and D. Roy, *Unfolding with generative adversarial networks*, (arXiv preprint) doi:10.48550/arXiv.1806.00433.

[80] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn and R. Winterhalder, *How to GAN away detector effects*, SciPost Phys. **8**, 070 (2020), doi:10.21468/SciPostPhys.8.4.070.

[81] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman and J. Thaler, *OmniFold: A method to simultaneously unfold all observables*, Phys. Rev. Lett. **124**, 182001 (2020), doi:10.1103/PhysRevLett.124.182001.

[82] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, A. Rousselot, R. Winterhalder, L. Ardizzone and U. Köthe, *Invertible networks or partons to detector and back again*, SciPost Phys. **9**, 074 (2020), doi:10.21468/SciPostPhys.9.5.074.

[83] M. Backes, A. Butter, M. Dunford and B. Malaescu, *An unfolding method based on conditional invertible neural networks (cINN) using iterative training*, SciPost Phys. Core **7**, 007 (2024), doi:10.21468/scipostphyscore.7.1.007.

[84] M. Leigh, J. A. Raine, K. Zoch and T. Golling, *ν-flows: Conditional neutrino regression*, SciPost Phys. **14**, 159 (2023), doi:10.21468/SciPostPhys.14.6.159.

[85] J. A. Raine, M. Leigh, K. Zoch and T. Golling, *Fast and improved neutrino reconstruction in multineutrino final states with conditional normalizing flows*, Phys. Rev. D **109**, 012005 (2024), doi:10.1103/PhysRevD.109.012005.

[86] A. Shmakov, K. Greif, M. Fenton, A. Ghosh, P. Baldi and D. Whiteson, *End-to-end latent variational diffusion models for inverse problems in high energy physics*, in *Advances in neural information processing systems 36*, Curran Associates, Red Hook, USA, ISBN 9781713899921 (2024).

[87] J. Ackerschott, R. K. Barman, D. Gonçalves, T. Heimel and T. Plehn, *Returning CP-observables to the frames they belong*, SciPost Phys. **17**, 001 (2024), doi:10.21468/SciPostPhys.17.1.001.

[88] S. Diefenbacher, G.-H. Liu, V. Mikuni, B. Nachman and W. Nie, *Improving generative model-based unfolding with Schrödinger bridges*, Phys. Rev. D **109**, 076011 (2024), doi:10.1103/PhysRevD.109.076011.

[89] B. Nachman and D. Shih, *Anomaly detection with density estimation*, Phys. Rev. D **101**, 075042 (2020), doi:10.1103/PhysRevD.101.075042.

[90] A. Hallin, J. Isaacson, G. Kasieczka, C. Krause, B. Nachman, T. Quadfasel, M. Schlaffer, D. Shih and M. Sommerhalder, *Classifying anomalies through outer density estimation*, Phys. Rev. D **106**, 055006 (2022), doi:10.1103/PhysRevD.106.055006.

[91] J. A. Raine, S. Klein, D. Sengupta and T. Golling, *CURTAINs for your sliding window: Constructing unobserved regions by transforming adjacent intervals*, Front. Big Data **6**, 899345 (2023), doi:10.3389/fdata.2023.899345.

[92] A. Hallin, G. Kasieczka, T. Quadfasel, D. Shih and M. Sommerhalder, *Resonant anomaly detection without background sculpting*, Phys. Rev. D **107**, 114012 (2023), doi:10.1103/PhysRevD.107.114012.

[93] T. Golling, S. Klein, R. Mastandrea and B. Nachman, *Flow-enhanced transportation for anomaly detection*, Phys. Rev. D **107**, 096025 (2023), doi:10.1103/PhysRevD.107.096025.

[94] D. Sengupta, S. Klein, J. A. Raine and T. Golling, *CURTAINs flows for flows: Constructing unobserved regions with maximum likelihood estimation*, SciPost Phys. **17**, 046 (2024), doi:10.21468/SciPostPhys.17.2.046.

[95] A. Butter, T. Jezo, M. Klasen, M. Kuschick, S. P. Schweitzer and T. Plehn, *Kicking it off(-shell) with direct diffusion*, SciPost Phys. Core **7**, 064 (2024), doi:10.21468/SciPostPhysCore.7.3.064.

[96] E. Buhmann, F. Gaede, G. Kasieczka, A. Korol, W. Korcari, K. Krüger and P. McKeown, *CaloClouds II: Ultra-fast geometry-independent highly-granular calorimeter simulation*, J. Instrum. **19**, P04020 (2024), doi:10.1088/1748-0221/19/04/P04020.

[97] J. Birk, E. Buhmann, C. Ewen, G. Kasieczka and D. Shih, *Flow matching beyond kinematics: Generating jets with particle-ID and trajectory displacement information*, (arXiv preprint) doi:10.48550/arXiv.2312.00123.

[98] L. Favaro, A. Ore, S. P. Schweitzer and T. Plehn, *CaloDREAM – Detector response emulation via attentive flow matching*, (arXiv preprint) doi:10.48550/arXiv.2405.09629.

[99] M. F. Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 – Dataset 1*, Zenodo (2022), doi:10.5281/zenodo.6234054.

[100] M. F. Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 – Dataset 1 version 3*, Zenodo (2023), doi:10.5281/zenodo.8099322.

[101] M. F. Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 – Dataset 2*, Zenodo (2022), doi:10.5281/zenodo.6366271.

[102] M. F. Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 – Dataset 3*, Zenodo (2022), doi:10.5281/zenodo.6366324.

[103] M. F. Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge* (2022), https://github.com/CaloChallenge/homepage.

[104] C. Krause et al., *CaloChallenge 2022: A community challenge for fast calorimeter simulation*, (arXiv preprint) doi:10.48550/arXiv.2410.21611.

[105] L. Dinh, D. Krueger and Y. Bengio, *NICE: Non-linear independent components estimation*, (arXiv preprint) doi:10.48550/arXiv.1410.8516.

[106] L. Dinh, J. Sohl-Dickstein and S. Bengio, *Density estimation using Real NVP*, (arXiv preprint) doi:10.48550/arXiv.1605.08803.

[107] D. P. Kingma and P. Dhariwal, *Glow: Generative flow with invertible 1 × 1 convolutions*, in *Advances in neural information processing systems 31*, Curran Associates, Red Hook, USA, ISBN 9781510884472 (2019).

[108] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother and U. Köthe, *Analyzing inverse problems with invertible neural networks*, (arXiv preprint) doi:10.48550/arXiv.1808.04730.

[109] L. Ardizzone, C. Lüth, J. Kruse, C. Rother and U. Köthe, *Guided image generation with conditional invertible neural networks*, (arXiv preprint) doi:10.48550/arXiv.1907.02392.

[110] P. Esser, R. Rombach and B. Ommer, *A disentangling invertible interpretation network for explaining latent representations*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, IEEE, Los Alamitos, USA, ISBN 9781728171685 (2020), doi:10.1109/CVPR42600.2020.00924.

[111] J. Q. Toledo-Marin et al., *Conditioned quantum-assisted deep generative surrogate for particle-calorimeter interactions*, (arXiv preprint) doi:10.48550/arXiv.2410.22870.

[112] P. Bambade et al., *The international linear collider: A global project*, (arXiv preprint) doi:10.48550/arXiv.1903.01629.

[113] M. Aicheler, P. N. Burrows, N. Catalan, R. Corsini, M. Draper, J. Osborne,D. Schulte, S. Stapnes and M.J. Stuart, *The compact linear collider (CLIC) – Project implementation plan*, CERN, Geneva, Switzerland, ISBN 9789290835141 (2018), doi:10.23731/CYRM-2018-004.

[114] A. Abada et al., *FCC-ee: The lepton collider*, Eur. Phys. J. Spec. Top. **228**, 261 (2019), doi:10.1140/epjst/e2019-900045-4.

[115] M. F. Giannelli, *Private communication* (2023).

[116] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios, *Neural spline flows*, in *Advances in neural information processing systems 32*, Curran Associates, Red Hook, USA, ISBN 9781713807933 (2020).

[117] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios, *Cubic-spline flows*, (arXiv preprint) doi:10.48550/arXiv.1906.02145.

[118] L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier and P. Sorrenson, *Framework for easily invertible architectures (FrEIA)* (2022), https://github.com/vislearn/FrEIA.

[119] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed and A. Lerchner, *beta-VAE: Learning basic visual concepts with a constrained variational framework*, in *International conference on learning representations*, Curran Associates, Red Hook, USA, ISBN 9781713872719 (2023).

[120] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins and A. Lerchner, *Understanding disentangling in β-VAE*, (arXiv preprint) doi:10.48550/arXiv.1804.03599.

[121] G. Loaiza-Ganem and J. P. Cunningham, *The continuous Bernoulli: Fixing a pervasive error in variational autoencoders*, in *Advances in neural information processing systems 32*, Curran Associates, Red Hook, USA, ISBN 9781713807933 (2020).

[122] L. Favaro and E. Florian, *Evaluation of the CaloINN generative network – Fast detector simulation*, Zenodo (2024), doi:10.5281/zenodo.14178546.

[123] G. M. Kurtzer, V. Sochat and M. W. Bauer, *Singularity: Scientific containers for mobility of compute*, PLOS ONE **12**, e0177459 (2017), doi:10.1371/journal.pone.0177459.

[124] M. Bellagente, M. Haussmann, M. Luchmann and T. Plehn, *Understanding event-generation networks via uncertainties*, SciPost Phys. **13**, 003 (2022), doi:10.21468/SciPostPhys.13.1.003.

[125] M. Sharma, S. Farquhar, E. Nalisnick and T. Rainforth, *Do Bayesian neural networks need to be fully stochastic?*, Proc. Mach. Learn. Res. **206**, 7694 (2023).