Towards a Fast Simulation of the ATLAS Calorimeter with Invertible Neural Networks

Dovydas Zemaitaitis[1,2]
Supervisors: Joshua Falco Beirer[1], Florian Ernst[1,3]

[1]CERN, Geneva, Switzerland
[2]University of Edinburgh, Edinburgh, United Kingdom
[3]Heidelberg University, Heidelberg, Germany

2024 September 9

Abstract

Producing accurate simulations faster than the traditional Monte-Carlo-based GEANT4 toolkit is critical for the LHC physics programme to keep up with the increasing demand for simulations. This report presents an ongoing effort to study the usage of invertible neural networks for the fast simulation of particle showers in the ATLAS calorimeter. An implementation of performance metrics and an experiment tracking platform with flexible model comparison capabilities are presented. Outcomes of hyperparameter tuning experiments are discussed.

1 Introduction

Monte Carlo (MC) simulations play a critical role in the Large Hadron Collider (LHC) physics programme, linking fundamental theory and experimental measurements [1–3]. The simulation of proton-proton collisions encompasses the generation of events at the parton level, simulating parton showering, hadronisation, decay of unstable particles, and interaction of the final-state particles with the detector matter [1]. The detector response is traditionally simulated using the GEANT4 toolkit [4, 5], which is highly accurate but extremely CPU-intensive, making it the primary bottleneck in the simulation process.

Approximately 80% of the CPU resources in ATLAS detector simulations are consumed by the simulation of electromagnetic and hadronic showers in the calorimeter [2]. The number of final-state particles increases with incident energies, making simulating the detector response more complex and time-consuming.

Instead of simulating the interaction of every particle in a shower with the detector material, the detector's response can be parameterised based on the incident particle's energy and direction by leveraging machine learning (ML) methods. Over the past decade, ATLAS has developed and employed faster calorimeter shower simulation methods, such as AtlFast3, which integrates both classical parameterisation and ML-based models, specifically generative adversarial networks (GANs) [6].

To further enhance the performance of detector simulations, various ML architectures are being explored. The current state-of-the-art includes deep generative networks such as diffusion models [7], normalising flows [8], variational autoencoders (VAEs) [9], and GANs [10]. A notable initiative in this field is the *Fast Calorimeter Simulation Challenge 2022* (further referred to as *CaloChallenge*), which aims to identify the most effective models for calorimeter shower generation [11, 12]. This report investigates the usage of one of the proposed architectures, invertible neural networks, for the fast simulation of calorimeter showers in ATLAS [13].

2 Technical background

2.1 Invertible neural networks

Invertible neural networks (INNs) are a specific type of normalising flows (NFs) [14–16]. These neural networks are designed to be bijective, meaning they can uniquely map each input to an output and vice versa, making the network invertible.

The basic concept of NFs is to learn a sequence of invertible, differentiable transformations that map the data distribution into a simple, arbitrary distribution, such as a Gaussian, which defines the latent space. Because these transformations are invertible, a sample can be drawn from the latent space, and inverse transformations applied in reverse to generate a new data point with the characteristics of the training data, effectively sampling its underlying probability distribution.

While the basic building block of normalising flows can be any invertible, piecewise differentiable object, INNs use coupling blocks. Compared to other NFs, like masked autoregressive flows, this design offers a significant speed advantage by requiring only a single forward pass, with the inverse computation being equally straightforward [17]. The simplest example is the affine coupling layer, shown in Figure 1. The block splits the input vector into two halves, then takes one half to parameterise itself and transforms the second half accordingly [13].

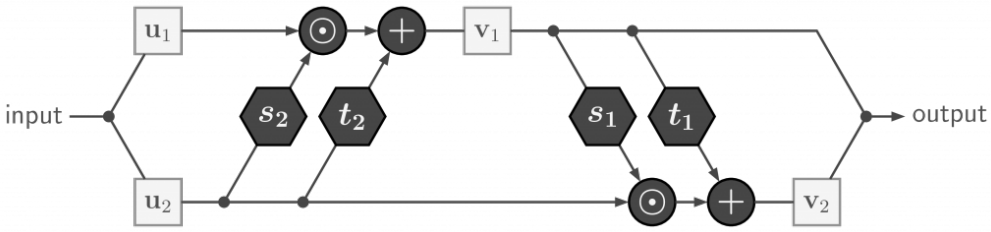

Figure 1: A schematic of an affine coupling layer. A component of the input layer, u_2 is used to generate scaling (s_2) and translation (t_2) factors to transform the component u_1, and then a corresponding action is performed for u_2. $\odot$ denotes the element-wise multiplication. This operation is easily invertible as the addition and multiplication operations are inherently invertible [18].

2.2 CaloINN

More complex invertible transformations can be employed to enhance the expressivity of the coupling block architecture. CaloINN uses splines with N_{knots} number of knots, which can be rational quadratic (RQS) [19, 20] or cubic [21, 22]. They transform the modified half of the features element-wise. Importantly, these splines are parameterised in a way that ensures monotonicity and thus invertibility for all possible parameter configurations passed from the non-invertible sub-networks. CaloINN is implemented using the Framework for Easily Invertible Architectures (FrEIA) [23] and PyTorch [24].

To simulate the calorimeter response for various incident energies, CaloINN must be conditioned on these energy values. This is achieved by passing the conditions, concatenated with the input, to the coupling blocks [25].

2.3 The ATLAS calorimeter system

The ATLAS detector at the LHC is a cylindrical multipurpose detector comprising subdetectors arranged in concentric layers that cover almost an entire solid angle around the interaction point. Its primary components include the inner detector, the calorimeter system and the muon spectrometer.

The detector uses a right-handed coordinate system with the origin at the collision point and the z-axis aligned with the beam direction. The following kinematic observables are defined: the azimuthal angle ϕ, measured in the (x, y) plane, and the pseudorapidity η, given by $\eta = -\ln\left(\tan\frac{\theta}{2}\right)$, where θ is the polar angle in the (y, z) plane [26].

The ATLAS calorimeter system covers a total pseudorapidity range of $|\eta| < 4.9$ and is designed to absorb most of the particles produced in a collision. It is composed of an electromagnetic calorimeter (ECAL) followed by a hadronic calorimeter (HCAL). Both are sampling calorimeters, meaning, they consist of alternating layers of dense material to absorb the incoming particles and active material to generate measurable electronic signals corresponding to the energy deposited.

The ECAL, with its fine granularity, is optimised for precision measurements of photons and electrons. It comprises the barrel (EMB) and endcap (EMEC) modules that utilise lead/liquid argon (LAr) as an active material. The ECAL's accordion-shaped geometry, as depicted in Figure 2, allows for higher packing density and improved access for readout electronics. Each layer is divided into groups of cells, known as towers. For higher incident energies, this complex geometry leads to event generation times with GEANT4 on the order of minutes for a typical sample involving the production of top-anti-top quark pairs ($t\bar{t}$) [6]. As a result, faster calorimeter simulation methods become essential.

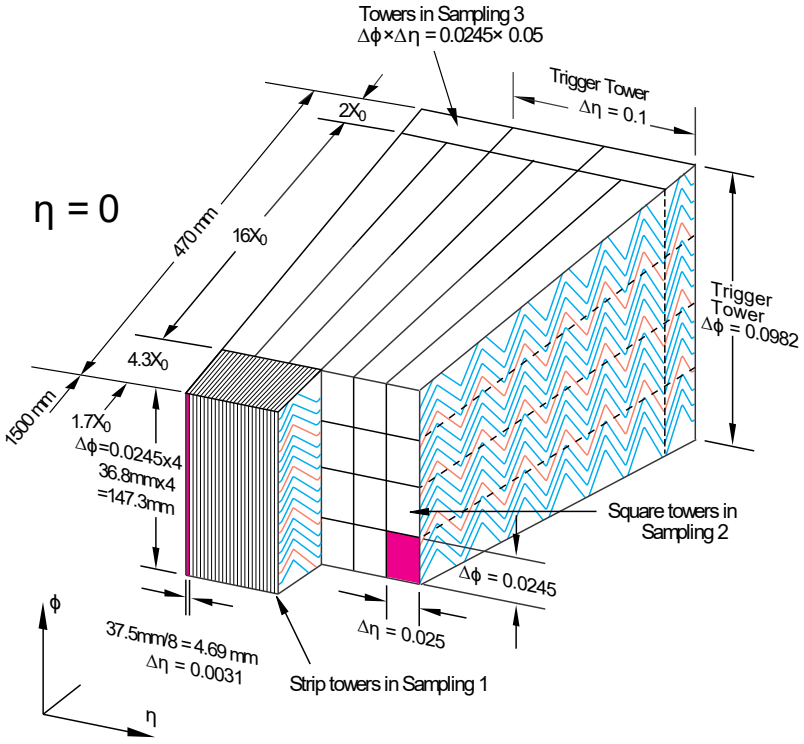

Figure 2: A cut-away view of the ATLAS ECAL tower, illustrating the accordion-shaped geometry that uses lead/liquid argon (LAr) as the active material (highlighted in blue and orange). This design enables closer packing and improved access to the readout electronics. The thickness of the calorimeter is measured in radiation lengths, X_0 [6].

The HCAL, positioned outside the ECAL, has a coarser granularity that meets the requirements for the reconstruction of jets and the determination of missing transverse momentum. It primarily uses scintillating tile technology for energy detection. In addition to the central tile calorimeter, there are the hadronic endcap calorimeter (HEC) and the forward calorimeter (FCal) that extend the detector's coverage [6, 26].

2.4 Dataset

The dataset used to parametrise CaloINN is generated using GEANT4 10.6. The dataset is currently limited to photons, which produce simpler showers with fewer interactions in the calorimeter material. This benchmark has already shown the effectiveness of NFs for simulating calorimeter showers [1].

The dataset is structured into 24 layers and includes photons with 15 different incident energies, ranging from 256 MeV to 4.2 TeV, with each energy level doubling the previous one. Similarly to *CaloChallenge* dataset 1, the coordinate system uses angular α and radial R variables, as shown in Figure 3. However, radial binning employs irregular voxelisation with piecewise uniform binning, using finer bins at the centre and coarser bins towards the outer regions. Each R-ring has a uniform binning in alpha with $n(R)$ bins, allowing for different resolutions at varying radii. This method also addresses the issue of larger outer voxels when using a constant $n(R)$. Approximately 250 voxels per layer are defined, though the binning strategy is still under development.

For the work presented below, only the region $0.20 < |\eta| < 0.25$ was considered, but CaloINN is capable of conditioning the training on different η regions. Each region contains approximately 130,000 events, with around 10,000 events for each of the 15 different incident energies. However, the highest energies have a reduced number of events due to longer generation times.

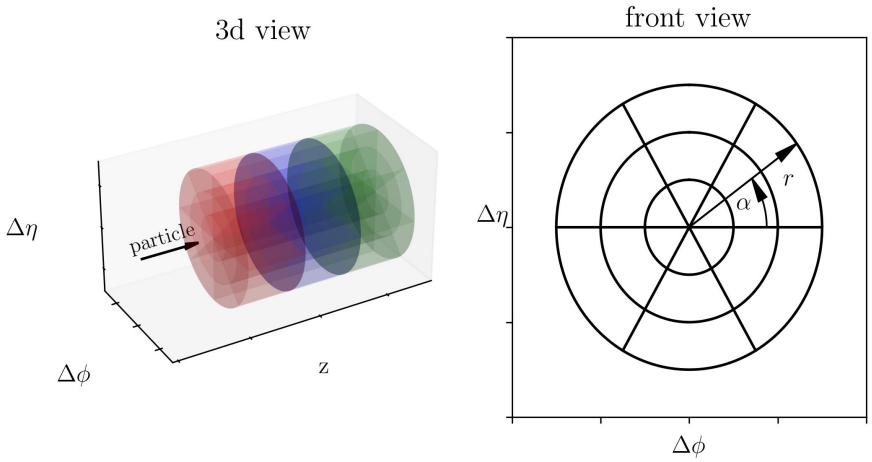

Figure 3: Schematic of the calorimeter setup used in the CaloChallenge dataset 1 [11]. While the front view illustrates uniform binning in the radial direction, it is important to note that CaloINN employs irregular binning, with finer bins closer to the centre.

3 Performance metrics

A consistent comparison method using standardised performance metrics is crucial for benchmarking the best-performing machine learning models for generating calorimeter showers in the ATLAS detector. To achieve this, performance metrics, such as energy deposition plots and numerical benchmarks detailed below, have been implemented.

3.1 Average layer energy deposition plots

To visually compare the energy depositions across each calorimeter layer of the reference GEANT4 shower samples and the CaloINN samples, average layer energy deposition plots were integrated into the CaloINN codebase.

Given the varying number of α bins $n(R)$, each plot requires determining the least common multiple (LCM) of the lengths of the corresponding R-boundaries' arrays in each R-ring, extending each array to match the LCM, doing the same extension for the corresponding energy arrays, and reshaping. An example of the plots for 256 MeV incident energy across layers 0, 1 and 2 is shown in Figure 4. Most of the energy is deposited in the shower core, and the energy is distributed roughly evenly in the angular direction.

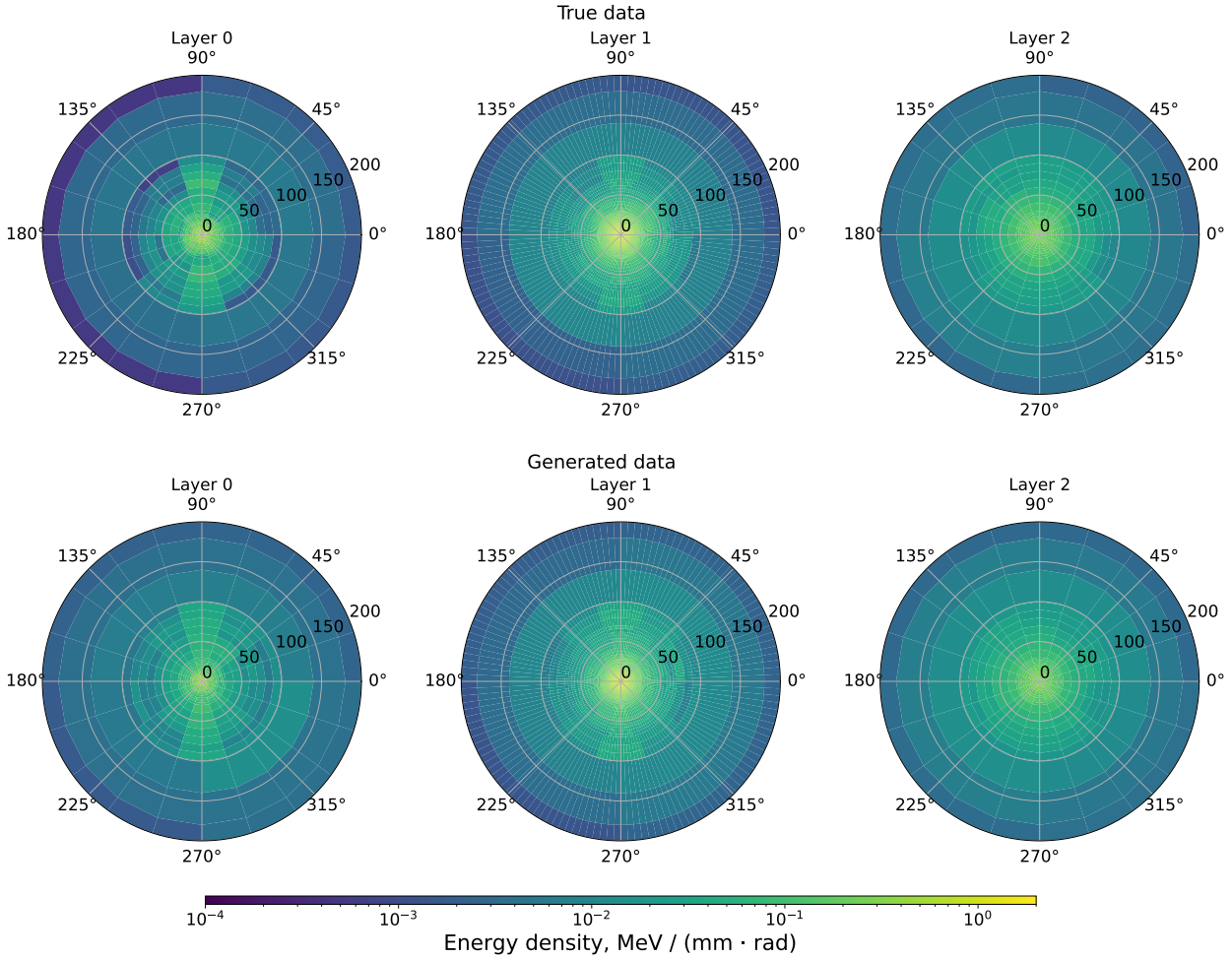

Figure 4: Comparison between average energy deposition of calorimeter showers simulated with Geant4 (top row) and simulated with CaloINN (bottom row). Energy depositions are shown in layers 0, 1 and 2 for 256 MeV photons in the region $0.20 < |\eta| < 0.25$. *The radius extends to 200 mm.*

3.2 Numerical metrics

The performance of the CaloINN model for various hyperparameter configurations was previously assessed by visually inspecting the loss curves from the training and testing sets and comparing the normalised GEANT4 and CaloINN observable distributions, shown in Figure 5. For each calorimeter layer used to train the network, the following observables are defined: the energy deposition in layer i, E_i, and the average pseudorapidity $\langle\eta\rangle_i$, with its standard deviation $\sigma_{\eta,i}$.

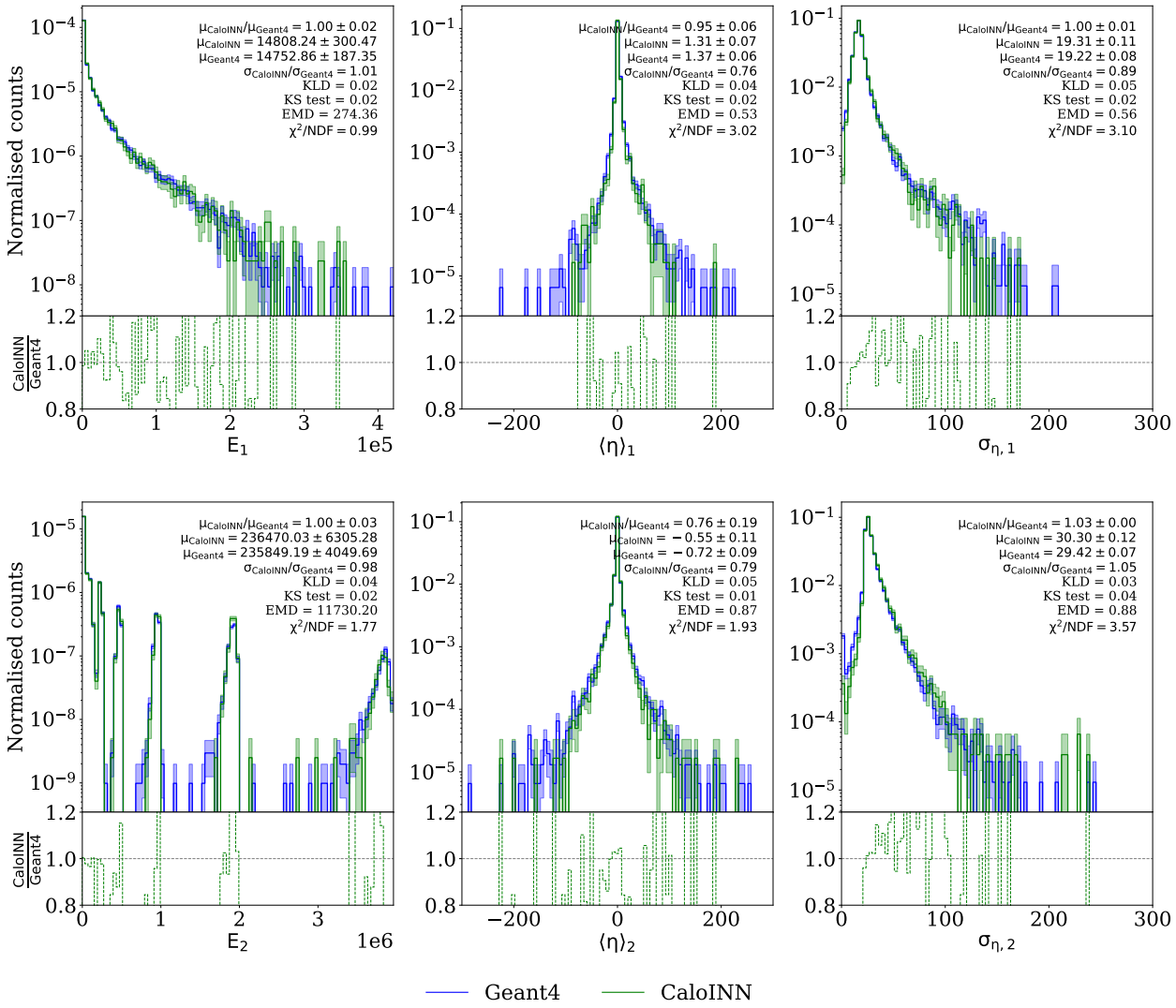

Figure 5: Comparison of various shower shape properties for calorimeter showers simulated with Geant4 and CaloINN. The energy deposition in layer i, E_i, and the average pseudorapidity $\langle\eta\rangle_i$, with its standard deviation $\sigma_{\eta,i}$, are shown for layers 1 and 2. Performance metrics for each graph are shown at the top right-hand corner. Plots are shown for all particle incident energies averaged.

To quantify the performance beyond visual inspection, a range of numerical performance metrics were implemented. Given that some observable histograms span several orders of magnitude, and have different unnormalised histogram counts or distinct features, no single benchmark can perfectly capture all aspects of the model's performance. Therefore, the following standard metrics were implemented for the comparison between CaloINN and GEANT4:

- Mean ratio
- Standard deviation ratio
- χ^2 per degrees of freedom (χ^2/NDF)
- Kullback–Leibler divergence (KLD) [27]
- Two-sample Kolmogorov–Smirnov test (KS test) [28]
- Wasserstein distance, or earth mover's distance (EMD) [29]

As illustrated in Figure 5, the feature to display metric values for each observable plot has been implemented. Additionally, the capability to average these metrics across all observables and plot their variation over different epochs has been added, as depicted in Figure 6. As expected, curves initially decrease and then plateau towards the end of the training. The mean and standard deviation ratios are converging towards unity.

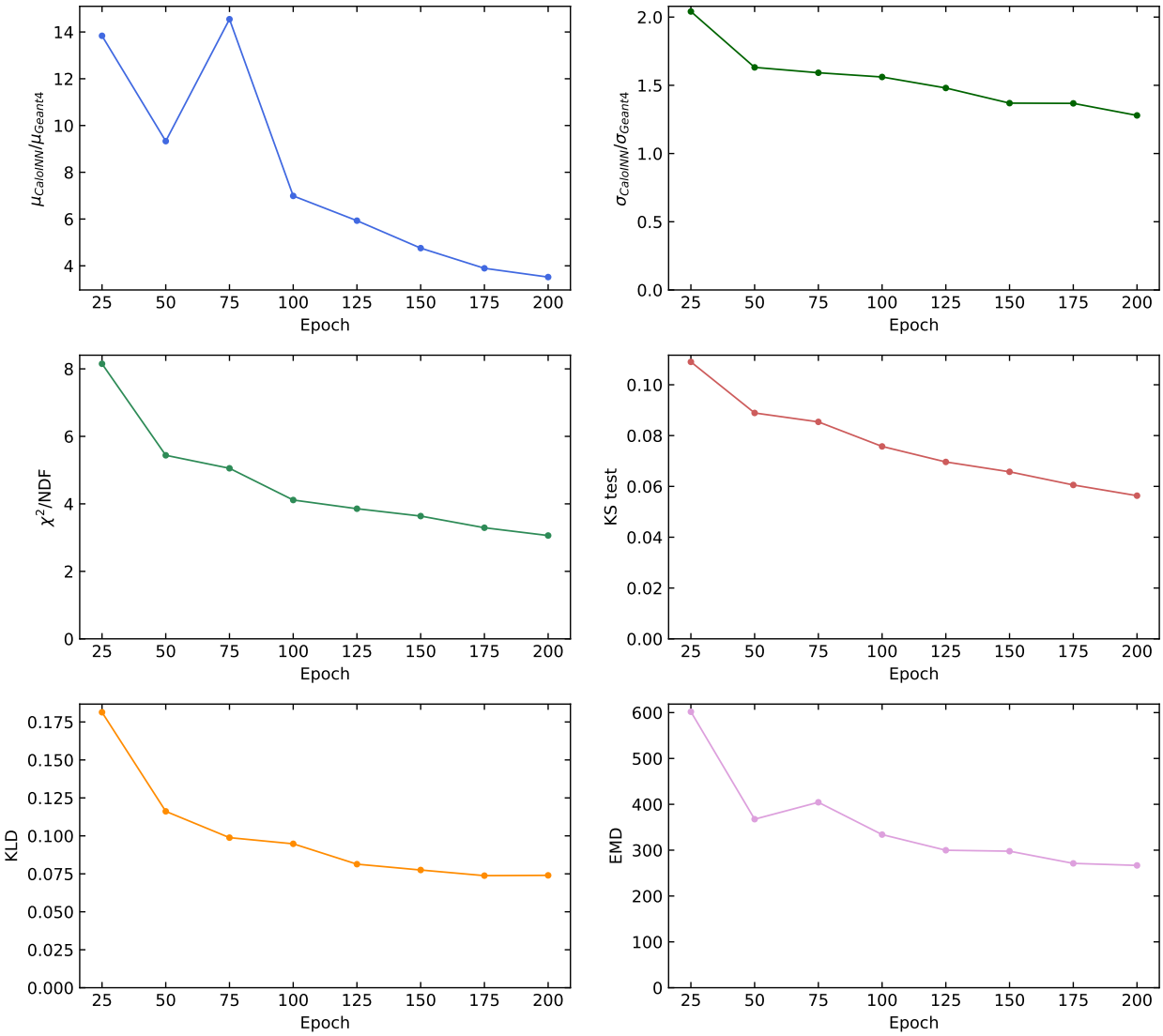

Figure 6: Performance metrics averaged over different observables against the training epoch.

4 Experiment tracking

CaloINN runs are self-documenting, automatically saving runtime logs, parameter files, the PyTorch model, and various plots for each run. However, during development, as multiple hyperparameter configurations and code versions are explored, comparing results across these runs becomes cumbersome.

After introducing the performance metrics, an implementation of experiment tracking software was explored to save the relevant files for each run of the model training. Based on the new metrics introduced, it should allow for the comparison between different hyperparameter configurations of the CaloINN, as well as cross-model comparisons (e.g.: INN vs GAN vs diffusion model). MLflow

was selected as an experiment-tracking tool for its open-source nature, active community support, simplicity, and flexibility [30].

4.1 MLflow

MLflow effectively addresses key challenges in the machine learning model lifecycle, providing a convenient platform for collaboration and experimentation. Its API is designed to be highly flexible, supporting any programming language, algorithm, or ML framework. This makes it easy to start and end runs, log parameters, track metrics (throughout training epochs), and store relevant files as artifacts. These API calls are straightforward to integrate into existing codebases and non-destructive.

A centralised MLflow tracking setup that enables multiple developers to log and compare their runs in a shared environment was implemented on CERN's Andrew File System (AFS). Logged runs can be accessed and queried through either the API or a web-based UI. The UI offers tools for grouping, sorting, and comparing runs. For example, it allows users to generate scatter plots to compare different hyperparameters against selected metrics or use parallel coordinates plots to visualise the relationships of different hyperparameter configurations with various comparison metrics, an example of which is shown in Figure 7. Additionally, it provides real-time updates and visualisation options for ongoing runs, enhancing the overall analysis and debugging process [30].

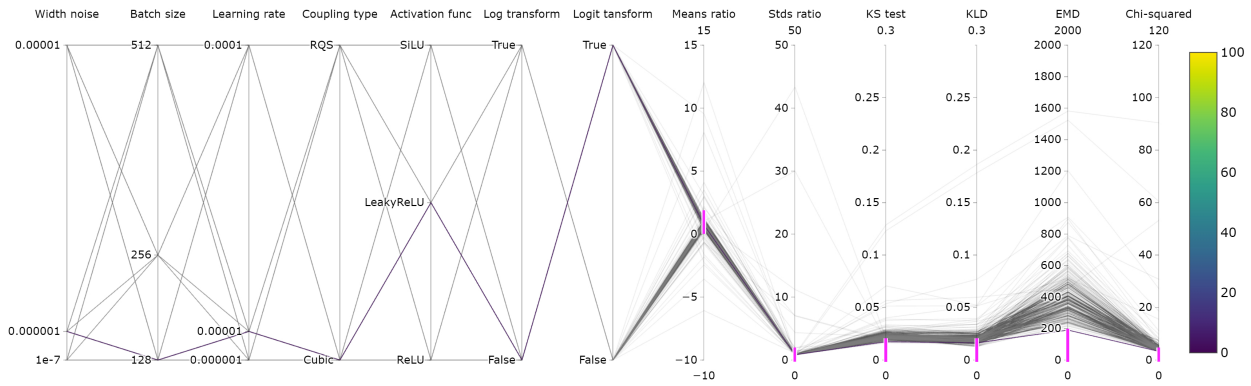

Figure 7: A parallel coordinates plot illustrating the relationship between various hyperparameter configurations and different performance metrics, with colouring based on χ^2/NDF. Pink sliders allow filtering, with greyed-out lines representing excluded configurations. This graph was used to identify the optimal hyperparameter settings in Section 5.2.

5 Hyperparameter tuning

The optimal hyperparameter configuration for CaloINN was initially selected under the assumption that different hyperparameters are uncorrelated and can be optimised independently. To refine this approach, the model's performance under different hyperparameter configurations was further explored by leveraging the newly added performance metrics and MLflow platform.

5.1 Optimal hyperparameters and the model size

Performing grid or random searches over a large hyperparameter space is computationally expensive, often requiring a full day to train a single model, even on a server GPU. Therefore, before an extensive search was committed to, it was first tested whether the optimal hyperparameters depend on model size.

The size of CaloINN is controlled by three hyperparameters: the number of coupling blocks N_{blocks}, the number of knots N_{knots}, and the internal size N_{int}. The hyperparameter space outlined in Table 1 was created as an ansatz, motivated by the CaloINN parameters from the *CaloChallenge*. 10 different hyperparameter configurations were randomly sampled from the hyperparameter space. Each was tested for both small and large model sizes by varying N_{blocks} and N_{int} parameters. To minimise training time, only the metrics averaged over all incident energies were considered. Additionally, training was limited to layers 1 and 2 of the calorimeter, as they collect most of the deposited energy and are the basis of photon identification in ATLAS [31]. Two experiments were done:

1. Compared a large model ($N_{\text{blocks}} = 14$, $N_{\text{int}} = 90$) with a small model ($N_{\text{blocks}} = 3$, $N_{\text{int}} = 32$), keeping other hyperparameters constant.
2. Compared a large model ($N_{\text{blocks}} = 14$, $N_{\text{int}} = 90$) with a medium-sized model ($N_{\text{blocks}} = 6$, $N_{\text{int}} = 64$), again with other hyperparameters unchanged.

In the first experiment, some correlation between the optimal hyperparameters of the small and large models was visually apparent, but insufficient to justify tuning the small model. Therefore, N_{blocks} and N_{int} were increased to test if convergence with the medium-sized model's results would occur. In the second experiment, the results resembled random noise, leading to the conclusion that, at least for the model sizes tested, optimal hyperparameters are correlated with model size.

5.2 Tuning large size model

Given that the optimal hyperparameters were found to be correlated with model size, a grid search across the entire hyperparameter space defined in Table 1 was conducted, encompassing 270 different configurations. The optimal hyperparameters, determined based on the performance metrics, are listed. The selection procedure is illustrated by a parallel coordinates plot obtained with MLflow and shown in Figure 7. A large model with $N_{\text{blocks}} = 14$ and $N_{\text{int}} = 90$ was used for this search. The jobs were launched in parallel on a batch computing system.

Table 1: Ranges of values used in a large model with $N_{blocks} = 14$ and $N_{int} = 90$ hyperparameter tuning. A width noise value of 1×10^5 was excluded when using logit as a preprocessing function, resulting in the total size of the hyperparameter space of 270.

Hyperparameter	**Selected value**	**Optimisation range**
Width noise	1×10^{-6}	$[1 \times 10^{-5}, 1 \times 10^{-6}, 1 \times 10^{-7}]$
Batch size	128	[128, 256, 512]
Learning rate	1×10^{-5}	$[1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}]$
Coupling type	cubic	[cubic, rational quadratic]
Activation function	Leaky ReLU	[SiLU, ReLU, LeakyReLU]
Preprocessing function	logit	[log, logit]

5.3 Reducing the model size

The size of CaloINN scales linearly with the number of knots in the splines used in the coupling blocks N_{knots}, and the number of coupling blocks, N_{blocks}. Increasing the model size impacts both training and inference time. While the timing does not necessarily scale linearly with N_{knots}, there is an expected linear relationship between time and N_{blocks}. To explore potential reductions in model size without significant performance loss, heatmaps of N_{knots} against N_{blocks} were generated, with various performance metrics as heat values. Examples with χ^2/NDF and KS test are shown in Figure 8.

The analysis shows that metric values exhibit slight random fluctuations with N_{knots} between 7 and 10, suggesting that reducing N_{knots} to 7 can be done with minimal performance impact.

As expected, metrics improve with increasing N_{blocks}, but appear to begin to level off around 10 and reach a plateau between 12 and 14. However, minor artifacts are still present in the observable graphs at $N_{\text{blocks}} = 10$, indicating a tradeoff between model speed and performance. The final choice of parameters will therefore depend on the specific production requirements.

The full-scale hyperparameter tuning was conducted on a model with $N_{\text{blocks}} = 14$ and $N_{\text{int}} = 90$, as detailed in Section 5.2. Therefore, rerunning the hyperparameter search over the entire space in Table 1 could benefit a smaller model.

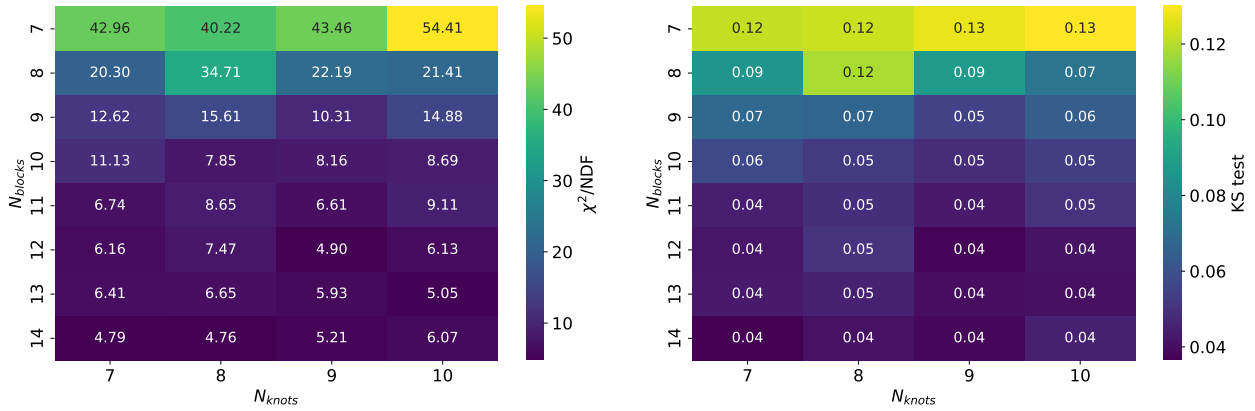

Figure 8: Heatmaps of the number of knots in the splines used in the coupling blocks N_{knots} against the number of coupling blocks, N_{blocks}. χ^2/NDF and KS test are shown as heat values.

6 Conclusion

Implementing the performance metrics and conducting the hyperparameter tuning experiments are important steps towards deploying CaloINN in a production environment. While in development, the joint experiment tracking server is expected to facilitate benchmarking and comparison with other models.

The next phase involves training the model on additional η regions and investigating the case of pions. Given the different nature of hadronic and electromagnetic physics, alternative ML model architectures may be considered.

AtlFast3 has shown excellent performance in simulating a wide range of physics processes. Alternative ML models such as INNs show great potential for further improving the quality of simulations in ATLAS and are likely to complement or fully replace the GAN-based shower generation of AtlFast3 in the near future.

Acknowledgments

I want to thank my supervisors, Joshua Falco Beirer and Florian Ernst, for warmly welcoming me into the FastCaloSim group, engaging in daily discussions, and encouraging my progress. I will always cherish this summer.

A Appendix

A.1 Shower shape properties

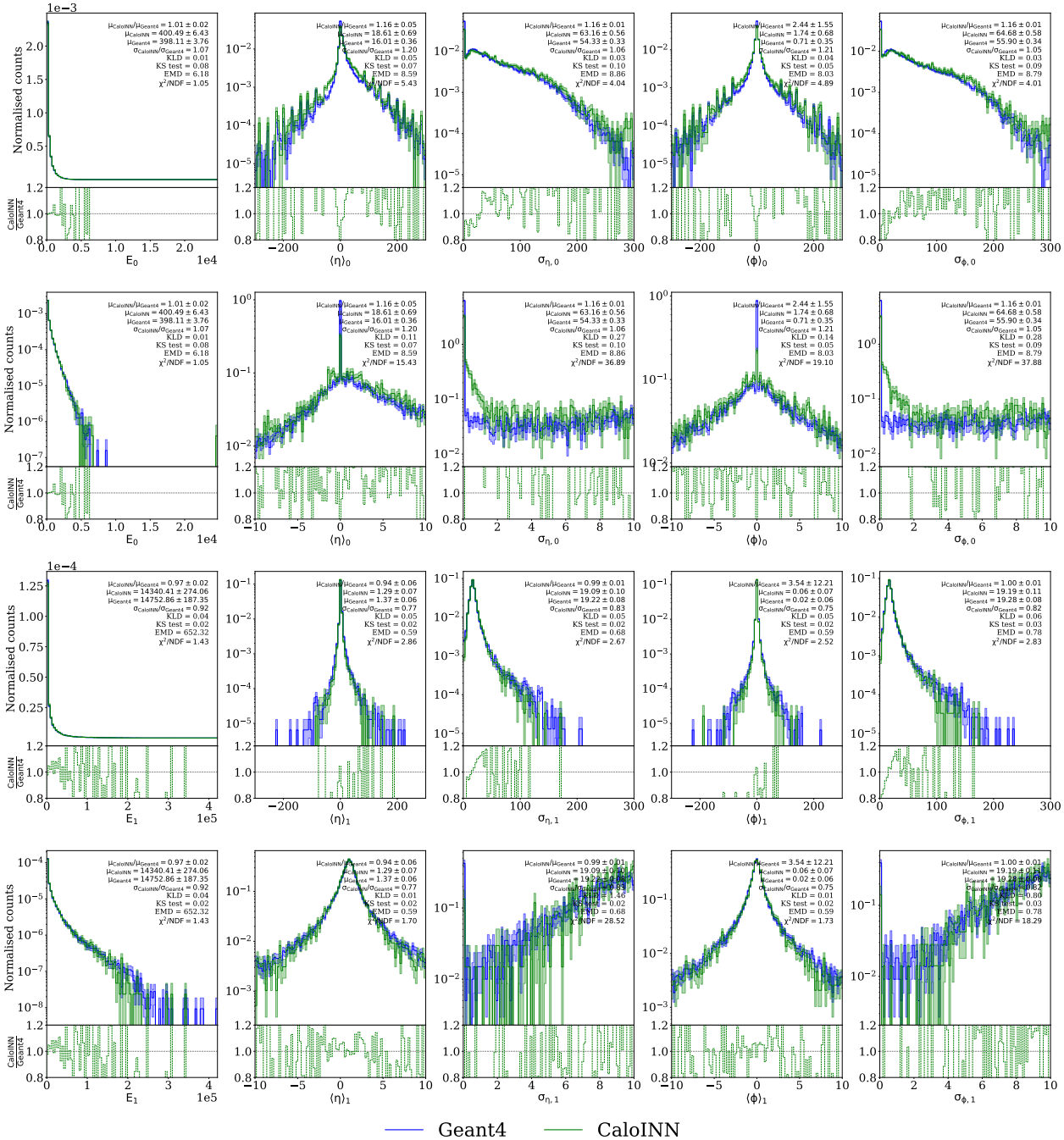

Figure 9: Comparison of various shower shape properties for calorimeter showers simulated with Geant4 and CaloINN. The energy deposition in layer i E_i, the average pseudorapidity $\langle\eta\rangle_i$, with its standard deviation $\sigma_{\eta,i}$, and the average azimuthal angle $\langle\phi\rangle_i$, with its standard deviation $\sigma_{\phi,i}$, are shown for layers 0 and 1. Performance metrics for each graph are shown at the top right-hand corner. Plots are shown for all particle incident energies averaged.

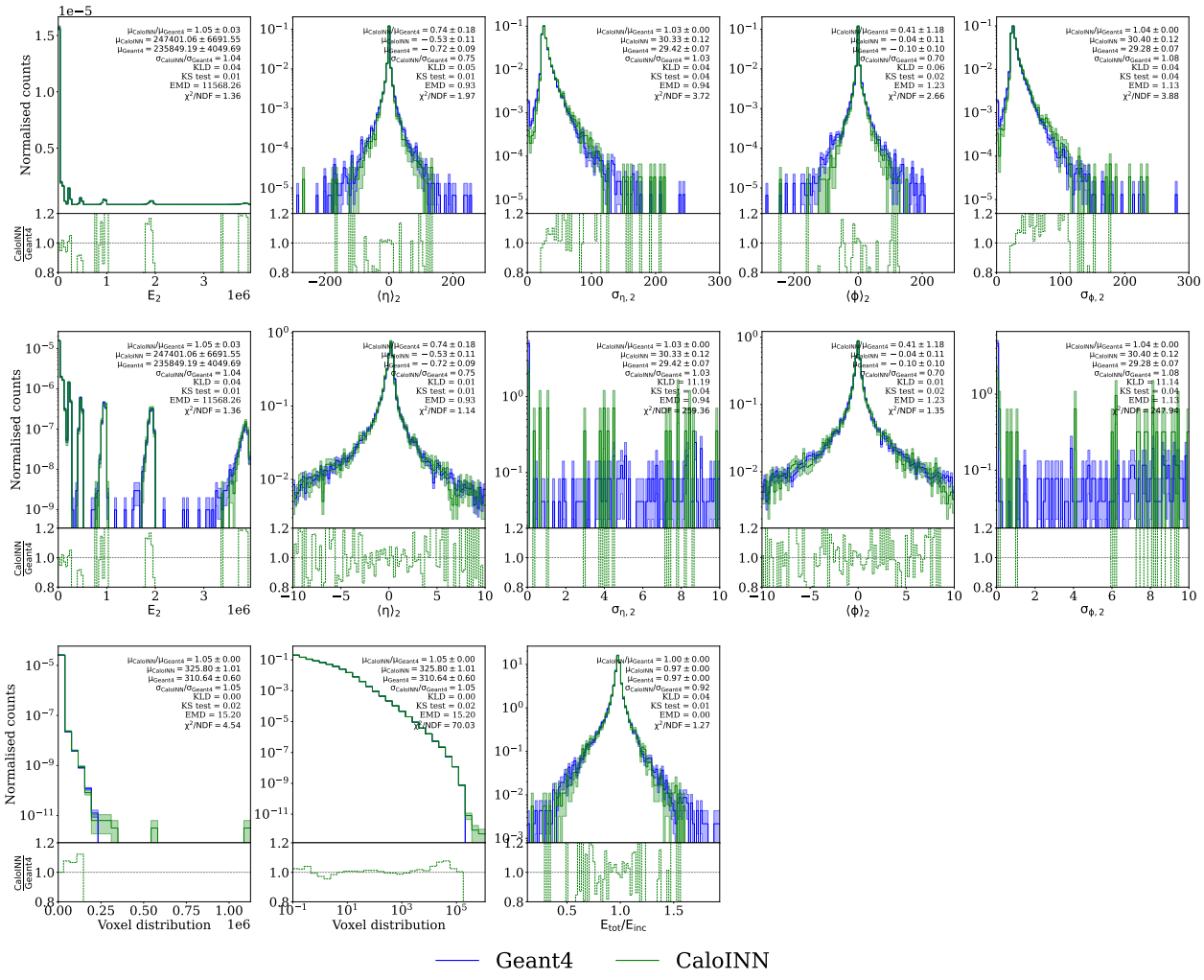

Figure 10: Comparison of various shower shape properties for calorimeter showers simulated with Geant4 and CaloINN. The energy deposition in layer i E_i, the average pseudorapidity $\langle\eta\rangle_i$, with its standard deviation $\sigma_{\eta,i}$, and the average azimuthal angle $\langle\phi\rangle_i$, with its standard deviation $\sigma_{\phi,i}$, are shown for layer 2. Additionally, two more observable distributions are analysed: voxel energies and the ratio of total energy to incident energy, E_{tot}/E_{inc}. Performance metrics for each graph are shown at the top right-hand corner. Plots are shown for all particle incident energies averaged.

A.2 Model size and performance heatmaps

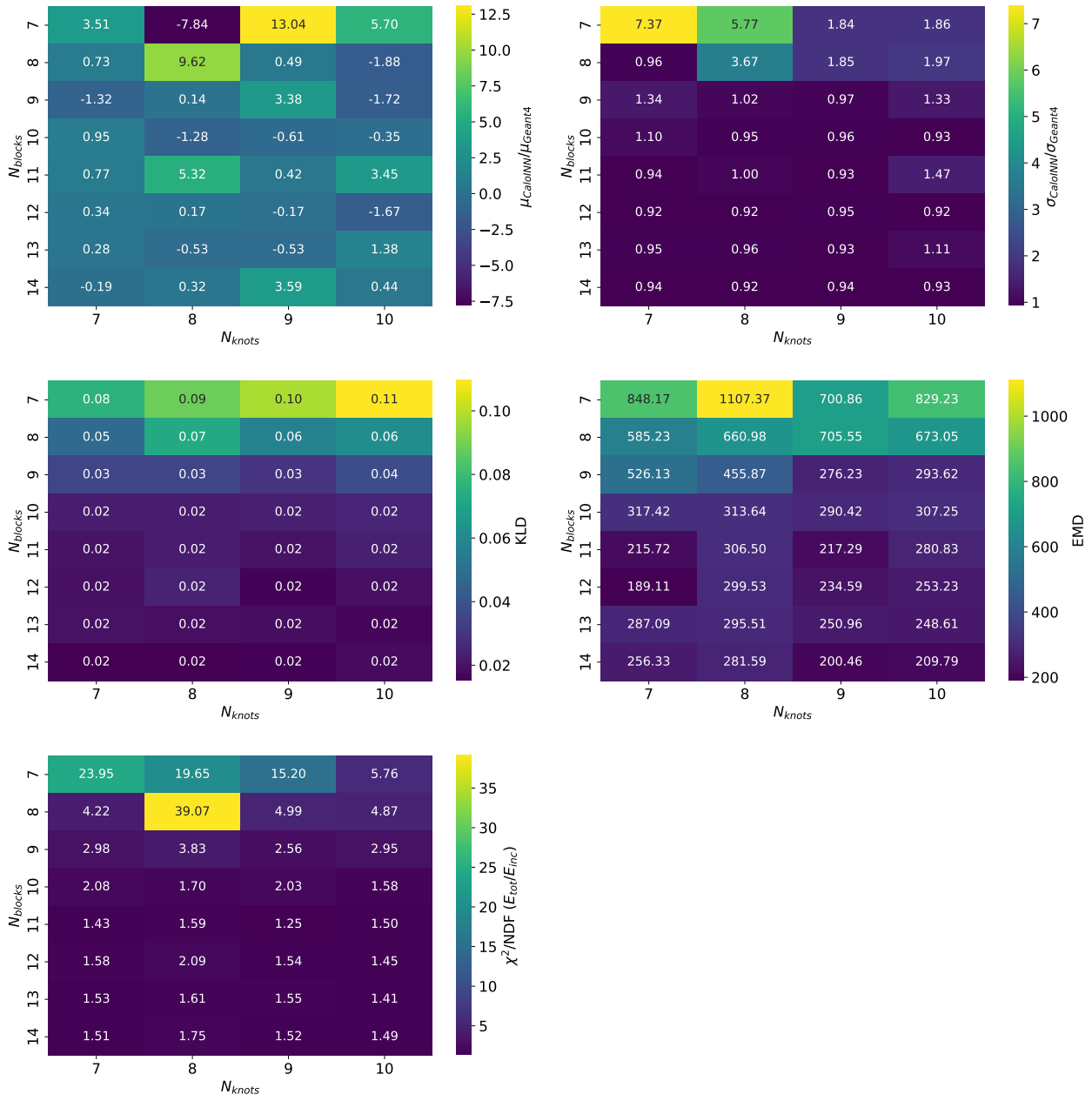

Figure 11: Heatmaps of the number of knots in the splines used in the coupling blocks N_{knots} against the number of coupling blocks, N_{blocks} and a performance metric as a heat value.

References

[1] Florian Ernst, Luigi Favaro, Claudius Krause, Tilman Plehn, and David Shih. Normalizing flows for high-dimensional detector simulations. *arXiv preprint arXiv:2312.09290*, 2023.

[2] Joshua Falco Beirer. *Novel Approaches to the Fast Simulation of the ATLAS Calorimeter and Performance Studies of Track-Assisted Reclustered Jets for Searches for Resonant $X \to SH \to b\bar{b}WW^*$ Production with the ATLAS Detector*. PhD thesis, Georg-August-Universität Göttingen, 2023.

[3] Anja Butter and Tilman Plehn. Generative Networks for LHC events. In *Artificial intelligence for high energy physics*, pages 191–240. World Scientific, 2022.

[4] Sea Agostinelli, John Allison, K al Amako, John Apostolakis, Henrique Araujo, Pedro Arce, Makoto Asai, D Axen, Swagato Banerjee, GJNI Barrand, et al. GEANT4—a simulation toolkit. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[5] John Allison, Katsuya Amako, John Apostolakis, Pedro Arce, Makoto Asai, Tsukasa Aso, Enrico Bagli, A Bagulya, S Banerjee, GJNI Barrand, et al. Recent developments in geant4. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835:186–225, 2016.

[6] Georges Aad, Brad Abbott, Dale C Abbott, A Abed Abud, Kira Abeling, Deshan Kavishka Abhayasinghe, Syed Haider Abidi, Asmaa Aboulhorma, Halina Abramowicz, Henso Abreu, et al. AtlFast3: the next generation of fast simulation in ATLAS. *Computing and software for big science*, 6(1):7, 2022.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[8] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[9] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[11] Michele Faucci Giannelli, Gregor Kasieczka, Claudius Krause, Ben Nachman, Salamani Dalila, Shih David, and Anna Zaborowska. Fast Calorimeter Simulation Challenge 2022. `https://calochallenge.github.io/homepage/`, 2022. [Accessed: 30-Aug-2024].

[12] Claudius Krause. The Fast Calorimeter Simulation Challenge 2022. In *ML4Jets2023*, Main Auditorium (DESY), November 2023. Presentation.

[13] Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

[14] Esteban Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8, 2010.

[15] Esteban Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[16] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji

Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[17] Claudius Krause, Ian Pang, and David Shih. CaloFlow for CaloChallenge dataset 1. *SciPost Physics*, 16(5):126, 2024.

[18] Computer Vision and Learning Lab. Analyzing inverse problems with invertible neural networks. `https://hci.iwr.uni-heidelberg.de/vislearn/inverse-problems-invertible-neural-networks/`, 2018. [Accessed: 30-Aug-2024].

[19] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[20] John A Gregory and Roger Delbourgo. Piecewise rational quadratic interpolation to monotonic data. *IMA Journal of Numerical Analysis*, 2(2):123–130, 1982.

[21] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Cubic-spline flows. *arXiv preprint arXiv:1906.02145*, 2019.

[22] Matthias Steffen. A simple method for monotonic interpolation in one dimension. *Astronomy and Astrophysics, Vol. 239, NO. NOV (II), P. 443, 1990*, 239:443, 1990.

[23] Lynton Ardizzone, Till Bungert, Felix Draxler, Ullrich Köthe, Jakob Kruse, Robert Schmier, and Peter Sorrenson. Framework for Easily Invertible Architectures (FrEIA). `https://github.com/vislearn/FrEIA`, 2022. [Accessed: 30-Aug-2024].

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[25] Florian Ernst. Exploring possibilities for higher dimensional calorimeter simulations. Master's thesis, University of Heidelberg, 2023.

[26] ATLAS Collaboration. The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3. *arXiv preprint arXiv:2305.16623*, 2023.

[27] Dmitry I Belov and Ronald D Armstrong. Distributions of the Kullback–Leibler divergence with applications. *British Journal of Mathematical and Statistical Psychology*, 64(2):291–309, 2011.

[28] Frank J Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[29] Victor M Panaretos and Yoav Zemel. Statistical aspects of Wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

[30] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.*, 41(4):39–45, 2018.

[31] ATLAS collaboration. Electron and photon efficiencies in LHC run 2 with the ATLAS experiment. *Journal of High Energy Physics*, 2024.