Contents lists available at ScienceDirect

Physics Letters B

journal homepage: www.elsevier.com/locate/physletb

MetaHEP: Meta learning for fast shower simulation of high energy physics experiments

Dalila Salamani*, Anna Zaborowska, Witold Pokorski

CERN, Espl. des Particules 1, 1211 Meyrin, Switzerland

ARTICLE INFO

Article history:
Received 15 August 2022
Received in revised form 8 June 2023
Accepted 8 July 2023
Available online 18 July 2023
Editor: M. Pierini

Keywords:
Fast calorimeter simulation
Generative models
Meta learning

ABSTRACT

For High Energy Physics (HEP) experiments, such as the Large Hadron Collider (LHC) experiments, the calorimeter is a key detector to measure the energy of particles. Incident particles interact with the material of the calorimeter, creating cascades of secondary particles, so-called showers. A detailed description of the showering process relies on simulation methods that precisely describe all particle interactions with matter. Constrained by the need for precision, the simulation of calorimeters is inherently slow and constitutes a bottleneck for HEP analysis. Furthermore, with the upcoming high luminosity upgrade of the LHC and a much-increased data production rate, the amount of required simulated events will increase. Several research directions have recently investigated the use of Machine Learning based models to accelerate particular calorimeter response simulation. These models typically require a large amount of data and time for training, and the result is a simulation tuned specifically to this configuration. Meta-learning has emerged recently as a fast learning algorithm using small training datasets. In this paper, we use a meta-learning approach that "learns to learn" to generate showers from multiple calorimeter geometries, using a first-order gradient-based algorithm. We present MetaHEP, the first application of the meta-learning approach to accelerate shower simulation using very high granular data and using one of the calorimeters proposed for the Future Circular Collider (FCC), a next-generation of high-performance particle colliders.

1. Introduction

The nearest future is very exciting for High Energy Physics (HEP), especially for the Large Hadron Collider experiments and their high-luminosity upgrade (HL-LHC) foreseen for 2029 [1]. The increase of luminosity means more data will be gathered, enabling new and higher precision measurements. This will require a corresponding increase in the amount of Monte Carlo (MC) simulated data, which already is a major consumer of the computing time for most experiments [2,3]. In order to fit within the available resources, HEP experiments are carrying out multiple R&D activities, including efforts to accelerate the simulation using so-called fast simulation techniques [3–6]. Some of those methods are already successfully applied in ATLAS [7], CMS [8], and LHCb [9].

Detailed simulation of particle transport through a model of a detector is performed using the Geant4 toolkit [10]. It allows to emulate a particle's interactions with matter by transporting it step by step by a small distance. This process evaluates the probabilities of interactions with the material of the detector. The energy of excitations and particles that are not simulated such as low-energy electrons, positions and photons are registered as energy depositions.

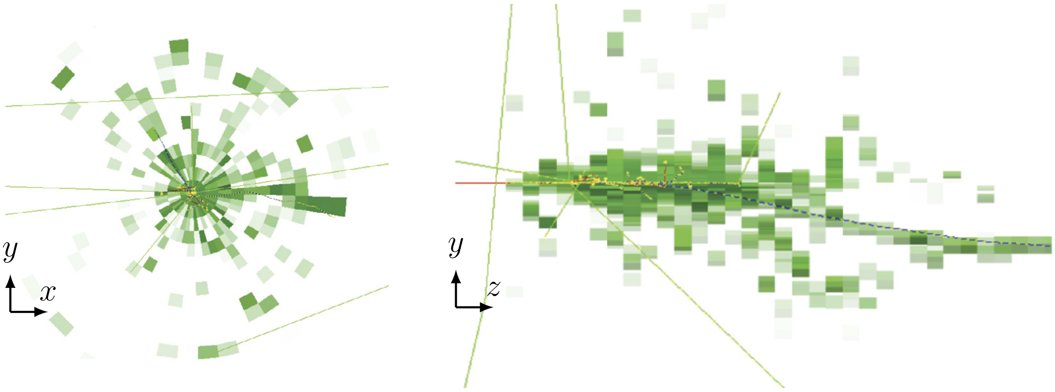
Interactions may produce secondary particles, which are added to the queue of particles, and simulated once the step-by-step propagation of the previous particle is completed. This detailed simulation is often referred to as a full simulation.

In a typical HEP collider experiment, a detector is comprised of light tracking devices to measure the momenta of the charged particles, one or more calorimeters to measure the energy of electrons, positrons, photons, and hadrons (both charged and neutral) and finally a muon tracking system to measure the momenta of muons. The basic principle of calorimetry depends on the creation of cascades of the secondary particles and their absorption in the dense material of the calorimeter. Those showers are composed of many particles and thus their simulation is computationally intensive. Calorimeters are typically the slowest detectors to simulate [11]. The acceleration of the simulation of showers in calorimeters is vital to address the computing challenges ahead.

Techniques for the fast simulation of showers rely on replacing the detailed step-by-step simulation of particle interactions

* Corresponding author.
E-mail addresses: dalila.salamani@cern.ch (D. Salamani), anna.zaborowska@cern.ch (A. Zaborowska), witold.pokorski@cern.ch (W. Pokorski).

https://doi.org/10.1016/j.physletb.2023.138079



(a) Energy deposition in the (x,y) plane. (b) Energy deposition in the (z,y) plane.

Fig. 1. Visualization of a single shower initiated by a 10 GeV electron and simulated with Par04 example [32]. Energy is deposited in a cylindrical mesh around the incident particle momentum. Projections in the lateral plane (a) and the longitudinal plane (b) are shown.

with matter with instantaneous creation of energy deposits in the detector. An example visualization of the deposited energy in a simulation of a 10 GeV electron is shown in Fig. 1. Such response of the detector to an incoming particle can be parametrized, with the goal of maintaining sufficient precision while decreasing the simulation time. Several approaches have been studied and applied in the experiments. One of the possibilities is to reuse several times the same detailed simulation [12] or to approximate with mathematical formulas the shape of the shower [13]. Given the complexity of the showers and the difficulty to model the statistical fluctuations, it is tough to address the fast shower simulation with high precision using current models, especially for granular calorimeters. Those solutions are therefore used only in the regions of the detector in which performance is not of utmost importance, or for the physics analysis that require very high statistics of simulated data, with fewer constraints on their precision.

In recent years, deep learning techniques have achieved great success in various science areas. In HEP, deep learning techniques have been used for fast simulation of detector responses such as in [14–19]. These studies focused on a single detector geometry using a large amount of data for training, which was found necessary to achieve good performance.

Meta-learning [21] is a learning-to-learn approach. Historically, this idea first appeared in 1987 [22]. In this family of methods neural networks are trained to get as inputs their own weights and predict weight updates. In Ref. [23] meta-learning with gradient descent and backpropagation was first proposed. This approach was extended for use in reinforcement learning [24]. A meta-learning approach takes a distribution of tasks, where each task corresponds to a learning problem. The goal is then to produce a quick learner which can generalize from small amounts of data examples. Model-agnostic meta-learning (MAML) [25] is a meta-learning algorithm whose optimization problem is learning the initialization parameters (weights) of a neural network. These parameters constitute the meta-knowledge and they are learned during a meta-training step. They can be used as initialization weights of the model and subsequently tuned for a new task. We refer to task-specific tuning as adaptation.

Reptile [26] is a first-order gradient-based meta-learning algorithm, i.e., it performs stochastic gradient descent (SGD) on each task in a standard way as opposed to MAML which computes the second derivatives. This makes Reptile more computationally and memory efficient, while the optimization problem remains the same as for MAML.

In this paper, we present MetaHEP, a generalizable and reusable solution for fast shower simulation. Our model is trained on multiple detector geometries and adapts quickly to a new one. To the best of our knowledge, MetaHEP is the first application of a generative meta-learning approach for fast shower simulation in HEP. It uses highly granular data, making it suitable for detectors of current and future experiments, such as the Future Circular Collider (FCC) [27].

2. Related work

Generative models such as generative adversarial networks (GANs) [39] and variational autoencoders (VAEs) [40] have been explored as a promising approach for fast shower simulation in HEP. Early applications using GANs [14] were implemented for electromagnetic shower simulation. The authors showed that GANs were able to generate highly realistic showers and reduce the computational cost of simulation by up to two orders of magnitude compared to full Geant4 simulation. Ref. [15], presents the first application of GANs and VAEs for calorimeter shower simulation of photons in the ATLAS detector. The two models use the readout of the energy deposits recorded by the cells of the ATLAS calorimeter. The computational time to generate new showers with these models was also reduced up to two orders of magnitude compared to the full Geant4 simulation, with a very small memory footprint of the order of 5 MB [20]. Because the ATLAS cell structure is not homogeneous, FastCaloGAN [19], another GAN-based model for fast simulation in the ATLAS detector, uses a voxelisation procedure to group the spatial energy deposits in each calorimeter layer from the Geant4 full simulation data into volumes called voxels. In fact, FastCaloGAN parameterizes the interactions of particles in the ATLAS calorimeter using 300 GANs, one for each particle type and η slice in which the reference samples are produced. FastCaloGAN uses the Wasserstein GAN with a gradient penalty (WGAN-GP) term in the loss function of the discriminator.

In addition to the above-mentioned models, other approaches have also been explored. In Ref. [18], a combination of VAEs and GANs known as the bounded information bottleneck autoencoder (BIB-AE) [41], is used to simulate electromagnetic showers in the central region of the Silicon-Tungsten calorimeter of the proposed ILD [28]. The authors show that using a modified BIB-AE for generation can accurately model all tested relevant physics distributions to a higher degree than achieved by traditional GANs. Normalizing flows have also been explored. The authors of CaloFlow [17] introduce a fast detector simulation framework based on normalizing flows. They also introduce a new validation metric of fast simulation approaches based on a classifier to test whether $p_{FastSim}(x) = p_{FullSim}(x)$. This is achieved by training a classifier on FullSim data and if it performs similarly to a classifier trained on

Table 1
Dimensions of the physical layout of the studied detectors.

Detector	1^{st} material thickness (mm)	2nd material thickness (mm)	Number of layers
SiW cylinders	0.3 mm	1.4 mm	90
SciPb cylinders	1.2 mm	4.4 mm	45
$PbWO_4$ cylinders	200.25 mm	-	1
SiW FCC-ee	0.5 mm	1.9 mm	40

Table 2
Dimensions of the dynamic mesh readout (presented in Fig. 3b) of the studied detectors.

Detector	R	P	N	Δr (mm)	Δz (mm)
SiW cylinders	18	50	45	2.3	3.4
SciPb cylinders	18	50	45	4	5.6
$PbWO_4$ cylinders	18	50	45	4.9	4.5
SiW FCC-ee	18	50	45	4.9	5.05

FastSim data, then this means that the FastSim data are well approximating the FullSim data.

3. Calorimetry and datasets

A particle entering the calorimeter initiates a cascade of secondary particles. Those particles deposit energy and produce further particles until the full energy of the primary particle is absorbed (or until the particle escapes the calorimeter).

The description of the three-dimensional shape of a shower is usually done by splitting the representation along the shower axis, i.e., the direction of the incident particle, to create the longitudinal profile, and in the transverse plane, named the lateral profile. The development of showers depends on the energy, the type and the direction of the incident particle, as well as on the characteristics of the detector or its detailed geometry: material, size, and thickness. The denser the detector, the smaller the size of the produced shower. For convenience, one can express the longitudinal and lateral distance from the shower axis in the material-specific units: the radiation length[1] z/X_0 and the Moliére radius[2] r/R_M. The values of z/X_0 and r/R_M units are used commonly to describe and compare different calorimeters, and they are also the basis of the material-independent shower parametrisation GFlash [13]. The importance of these units for MetaHEP is described further in Sec. 3.2.

3.1. Calorimeter geometry description

In this paper, we consider four different calorimeter geometries. Three geometries are idealised versions of calorimeters serving as demonstrators. They are setups of concentric cylinders with interchanging layers of materials. The first geometry uses layers of silicon and tungsten (SiW), the second one uses scintillator and lead (SciPb) and the third one consists of a single material, lead tungstate ($PbWO_4$). The fourth geometry used in this paper is a more realistic geometry, studied in the context of the electron-positron FCC-ee experiment [29] and based on the CLIC detector [30]. This geometry comprises flat silicon sensors, placed in an octagonal pattern in the transverse cross-section of the detector. Silicon sensors are accompanied by a number of materials, most notably tungsten, but also readout and electrode plates, and include spacing in between layers. The important physical characteristics of those detectors are summarised in Table 1. For the FCC-ee calorimeter, only the thickness of silicon and tungsten are mentioned as the first and second materials, respectively. The source file for the FCC-ee calorimeter can be found in [31].

Fig. 2 presents a comparison of basic shower characteristics for the studied detectors. It is shown using the example of 64 GeV electrons entering the detectors perpendicularly to the cylinder axis. The distribution of energy deposits in a single readout cell is presented in Fig. 2a. The readout structure of the detectors is described in detail in Sec. 3.2. The largest energy deposits are made in the cells of the lead tungstate calorimeter ($PbWO_4$) as this is the homogeneous detector, where the absorber is also the material in which the deposited energy is recorded. This is also visible in the distribution of the total energy presented in Fig. 2b. Almost all of the energy of the incident particle is deposited in the homogeneous calorimeter and only a few per cent for the other sampling detectors.

The longitudinal shower profile is shown in Fig. 2c in metric units, and in Fig. 2d in units of radiation length. The latter distribution demonstrates how expressing the longitudinal distance in terms of material-specific units can help to generalize the shower profile despite the difference in materials. The same conclusion can be drawn for the lateral shower profile, shown in Fig. 2e in metric units, and in Fig. 2f in units of Moliére radius.

3.2. Detector readout

The basic idea behind MetaHEP is connected to the construction of the detector readout. In order to obtain similar information on energy deposits from different detectors, an innovative scoring mesh is implemented, independent of the physical readout of the detector. The advantage of this definition is that it allows a high granularity segmentation of energy deposited in the simulation, independently of the angle at which a particle enters the detector. Moreover, such a representation of a shower with mesh cell size expressed in material-specific units facilitates generalization to different geometries.

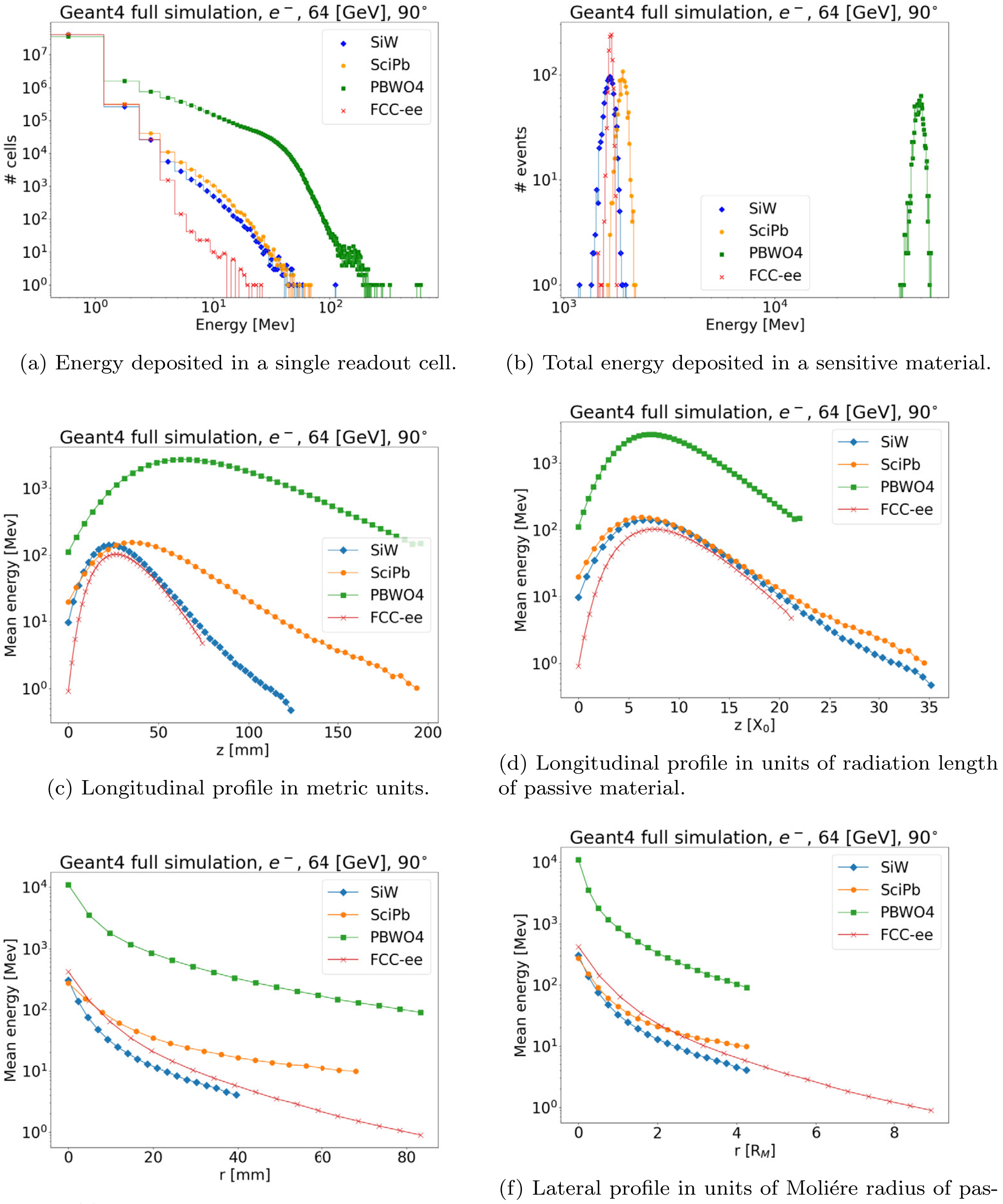
The principle of energy scoring in this mesh is the following: whenever a particle enters the volume of the calorimeter, its momentum direction is used to define the position and the orientation of the cylindrical readout structure, which is centred around the particle momentum, as shown in Fig. 3a. The metric size of the mesh determines the granularity of the produced showers and can differ for each of the studied geometries. It should remain similar when expressed in material-specific units. Moreover, the number of mesh cells should remain identical, as it is directly linked to the design of the ML parameterisation model. The cylindrical readout is presented in Fig. 3b. Dimensions of the mesh readout for the studied geometries are summarised in Table 2. The number of mesh cells has been optimized to contain on average 95% of energy of 1 TeV electrons.

3.3. Datasets

Simulation of single electrons in the studied detectors has been performed with GEANT4 producing datasets used in this paper. The energy of incident electron ranges from 1 GeV to 1 TeV (in powers of 2, in total 11 energy values: 1 GeV, 2 GeV, 4 GeV, . . . , 1024 GeV) and incident angle θ from 50° to 90° (in a step of 10°, in total 5 angles). Particles with an incident angle θ of 90° are perpendicular to the detector axis. Incident angle ϕ is sampled uniformly from 0 to 360° for a cylindrical detector and is set to 90° for a realistic

[1] The radiation length is the average distance over which an energetic electron loses $1-e^{-1}$ of its energy by bremsstrahlung. It is also on average 7/9 of the mean free path an energetic photon travels before creating e^-e^+ pair.

[2] The Moliére radius is defined as the radius of a cylinder in which particle deposits on average 90% of its initial energy.



(a) Energy deposited in a single readout cell.

(b) Total energy deposited in a sensitive material.

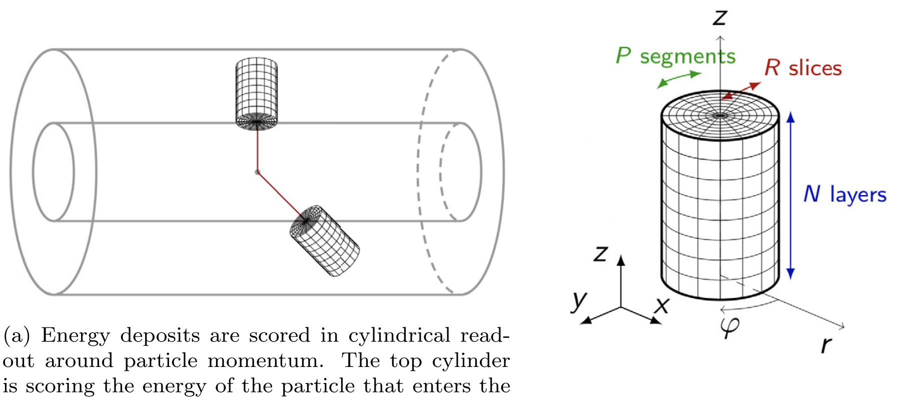
(c) Longitudinal profile in metric units.

(d) Longitudinal profile in units of radiation length of passive material.

(e) Lateral profile in metric units.

(f) Lateral profile in units of Moliére radius of passive material.

Fig. 2. Comparison of basic shower characteristics for the studied geometries.



(a) Energy deposits are scored in cylindrical readout around particle momentum. The top cylinder is scoring the energy of the particle that enters the calorimeter at the incident angle of 90°.

(b) Cylindrical (r, ϕ, z) segmentation.

Fig. 3. Energy scoring mesh.

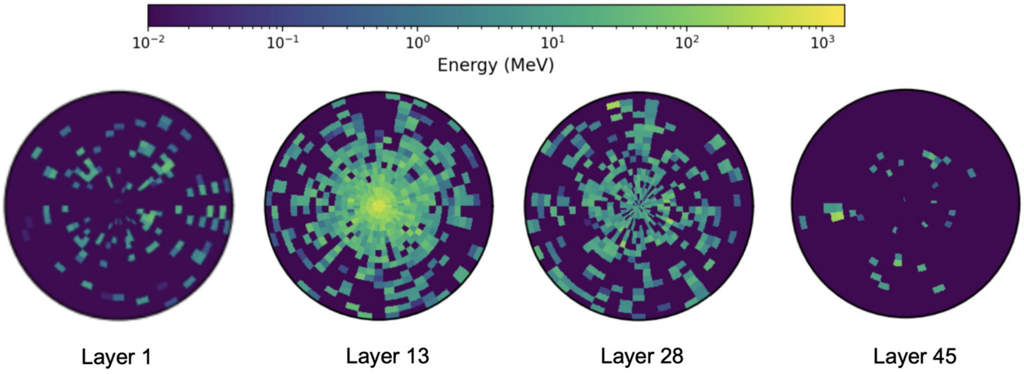


Fig. 4. Shower energy deposition in 4 selected layers of a 1 TeV electron entering the SiW calorimeter at an incident angle of 90°.

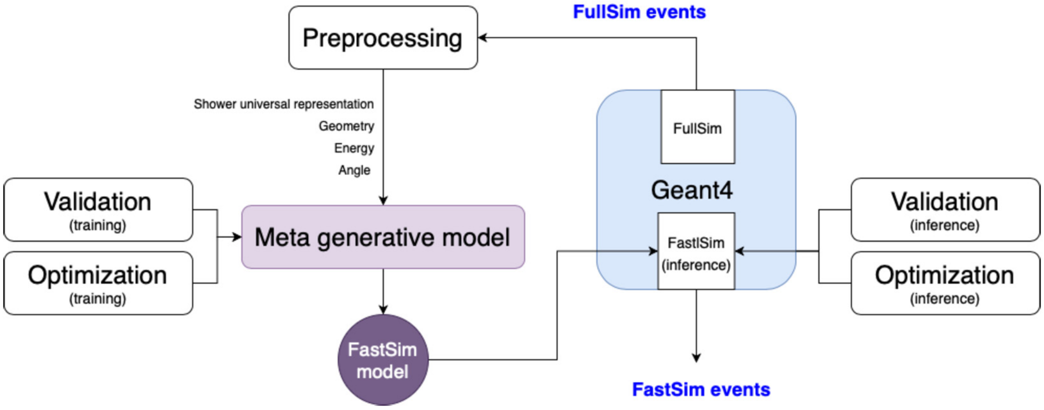


Fig. 5. MetaHEP workflow.

detector. In the latter case, due to the lack of ϕ symmetry one needs to consider both ϕ and θ as input. 10'000 particle showers are simulated for each energy and angle. Separate samples were generated for the sake of validation studies.

Idealised, cylindrical detectors used in this paper are simulated with the ParO4 example of GEANT4 toolkit [32] with the dimensions of the materials and the readout mesh described in Table 1 and Table 2, respectively. Full simulation of the model of an FCC-ee detector is performed with the FCCSW [33], a common software for all FCC experiments using the turnkey software stack Key4HEP [34]. Datasets that were used in the initial training of the model (SiW and SciPb) are available on Zenodo [35]. Datasets simulated with the $PbWO_4$ cylindrical geometry and the FCC-ee detector are used to demonstrate the power of MetaHEP, the adaptation step. In order to demonstrate that the model can learn from a small subset of a dataset, only 30% of the available statistics (for each energy and angle) is used for the meta-training.

Fig. 4 shows a shower energy deposition in (x,y) coordinates for four selected layers of the SiW cylindrical detector, for a primary electron with an energy of 1 TeV. It shows the evolution of the shower, with small deposits in the first and last layers, and the bulk of the energy deposited in the middle.

4. Meta learning with generative modelling

We now describe our meta-generative model. First, we present the experimental setup and then describe how we can approximate a generation function for fast shower simulation using a meta-learning model.

4.1. Proposed workflow

Fig. 5 represents the different modules of the MetaHEP solution workflow. The preprocessing module allows us to prepare the simulated data, using the representation of showers in the readout mesh described in Sec. 3.2. This module is also used to encode condition information such as detector identifier, the energy of the particle initiating the shower, and the angle at which the particle enters the detector. The detector identifier is encoded as a one-hot encoding vector of size 3, i.e., the first detector gets [0,0,1] and the second [0,1,0] and the third gets [1,0,0].

The preprocessed data is then used by the meta-generative model for training and adaptation. After this step, the model is converted into a format such as JSON or ONNX that can be used for inference in C++ (GEANT4 inference). ParO4 [32] is a GEANT4 example which demonstrates how to use a generative model incorporating inference libraries such as LWTNN [36], ONNX runtime [37] and LibTorch [38]. Running inference within the same framework that performs full simulation enables direct comparison of full and fast simulation, and, more importantly, its application in a HEP experiment data production cycle. Two stages of optimization are used, one at training and one at the inference phase. The former is used to search for the best set of hyperparameters of the model with an optimized number of trainable parameters. The latter is used for inference, enabling further optimization to reduce the memory footprint of the model.

In order to model a physics phenomenon, such as a detector response to a traversing particle, a certain number of occurrences of this phenomenon are simulated. Analysis of the distributions describing measured properties of these events on a statistical basis provides insight into the quality of the modelling. The validation

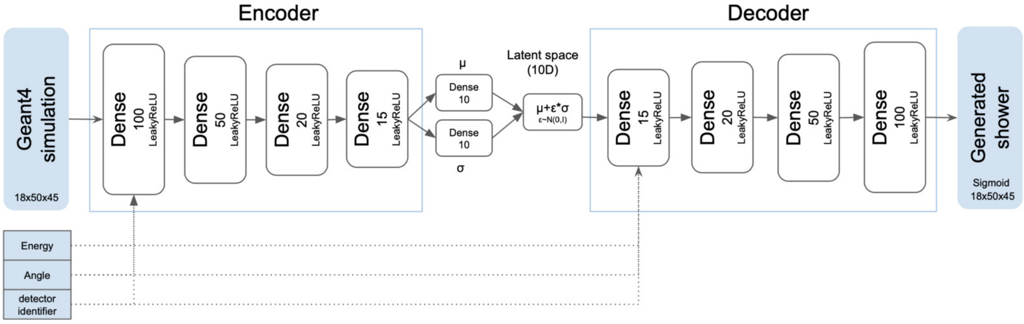


Fig. 6. VAE model architecture.

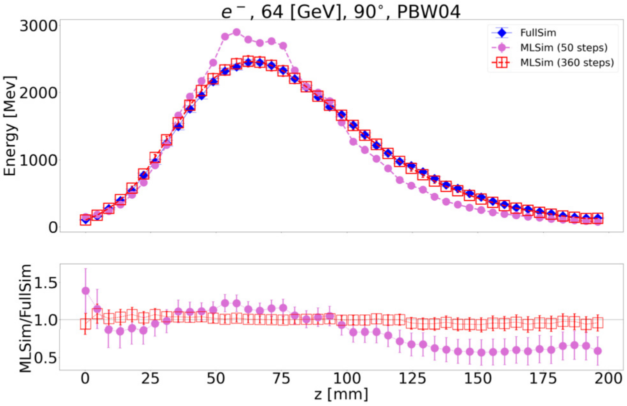


Fig. 7. Average longitudinal profile after 50 and 360 adaptation steps using a mini-batch of 100 samples of 64 GeV electrons entering the $PBWO_4$ calorimeter at an incident angle of 90°. The error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

is a key module of the workflow used to assess the performance of the model. In our study, validation is based on comparing distributions of shower observables between full and fast simulation. These observables characterise shower properties such as longitudinal and lateral profiles. The longitudinal profile describes how the energy is deposited by a shower as a function of the depth of the calorimeter i.e., the z-axis, shown in Fig. 3b. The lateral profile represents the energy density distribution as a function of the radial coordinate i.e., the r-axis shown in Fig. 3b. The first and second moments of those distributions are also checked in the validation phase, together with the basic distributions of energy: single cell distributions and the total deposited energy i.e., the sum of the energies of all cells. To evaluate the generative capability of the model, a standalone validation procedure is performed (in Python) during or after the training/adaptation phase. On the other hand, at the inference stage, when the model is being run within GEANT4, is used to ensure that the fast simulation is accurate after mapping the inferred energies to positions in the calorimeter. The result of the two validation procedures should be identical.

In this paper, we will only focus on the meta-generative model component.

4.2. Generative fast shower simulation model

Variational Autoencoders (VAEs) combine ideas from representation learning and probabilistic latent variable modelling to derive a class of deep learning models. Representation learning approaches have been widely used for supervised and unsupervised tasks, and in particular with deep learning architectures, with the multiple non-linear transformations yielding more abstraction and potentially more useful representations [42].

The model used in this paper is a VAE. A VAE is composed of two stacked deep neural networks acting as encoder and decoder. The encoder learns a meaningful representation of the input data with a variational principle. This representation in a latent space has a lower dimensionality than the input. The decoder, on the other hand, learns the inverse mapping, thus reconstructing the original input from this latent representation. The VAE is designed with a prior on the representation space therefore to constrain the encoded distributions to be Gaussian distributions. The encoder network is tasked to return the mean and the covariance matrix describing those distributions. The loss function that is optimized during the training of the VAE is composed of two parts. The first is a regularization loss to minimize the Kulback-Leibler divergence [43] between encoded distributions and the prior' Gaussian distributions. The second is a reconstruction loss to minimize the error by computing the binary cross-entropy between the input and the version reconstructed using the latent representation. In domain applications, we are often interested in conditional learning. In our work, the VAE model is tasked to learn a conditional probability density based on information about the detector identifier, the particle's energy, and the particle's incident angle. The VAE architecture used in this work is shown in Fig. 6. It comprises 4 hidden layers with widths of 100,50,20,15 and 15,20,50,100 for the encoder and decoder respectively.

4.3. Meta learning for fast shower simulation

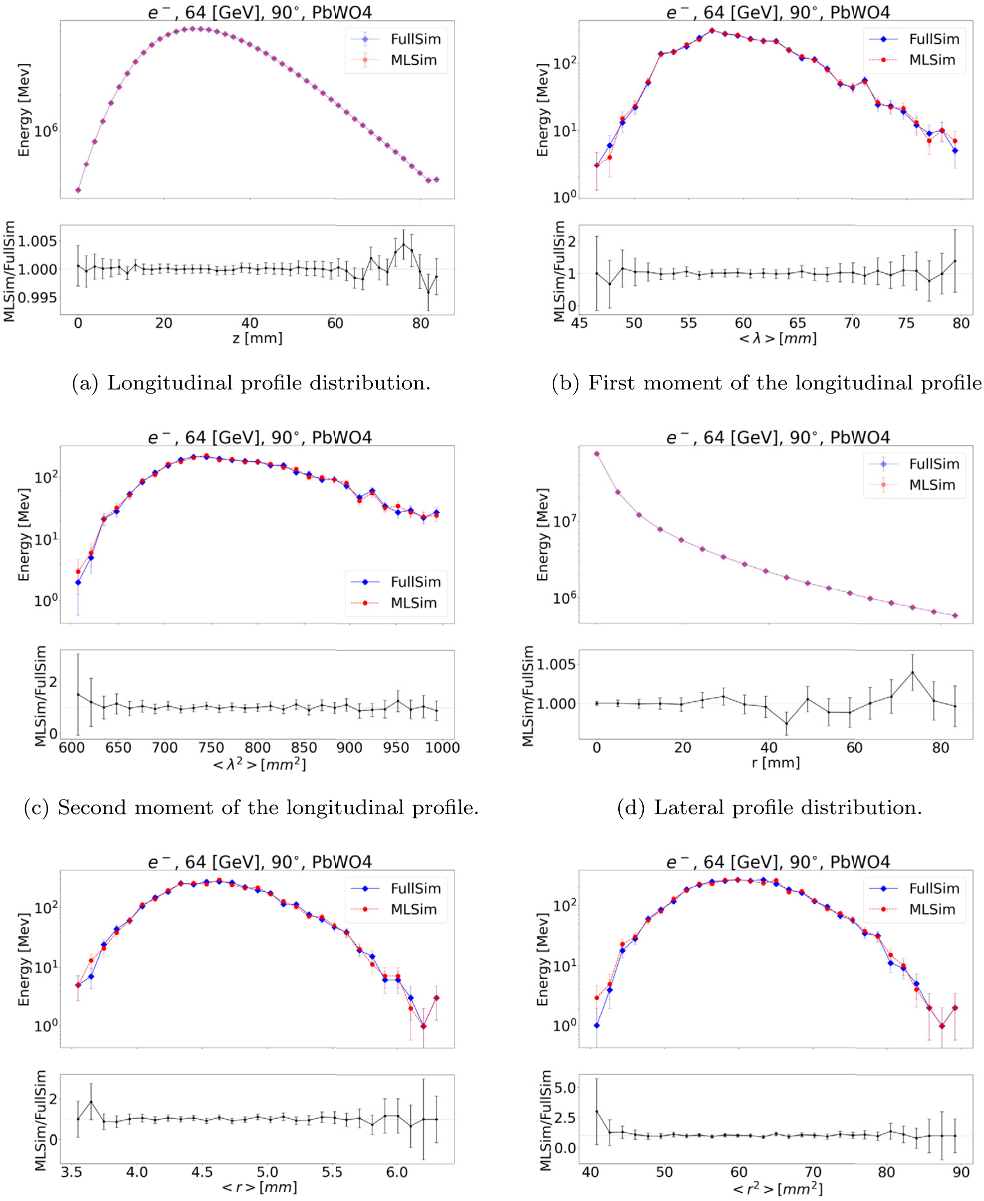
In this paper, we use Reptile [26] as a meta-learning approach. Let θ denote the initial model parameters and t a task from the set of tasks. In our case, a task corresponds to learning to simulate showers from one detector geometry. For a randomly sampled task t, the optimization problem is to minimize $E_t[L_t(U_t^k(\theta))]$, where L_t represents the loss of the task t, k is the number of steps or updates and U denotes the updating operator such as gradient descent. The loss function is the VAE loss constructed as a binary cross entropy loss and a Kulback-Leibler divergence.

Reptile, described in Algorithm 1, is an iterative algorithm. It first starts by sampling a task from the distribution of tasks, trains on the task, and then moves the initial weights of the model towards the trained weights [26].

Algorithm 1 Reptile algorithm [26].

```
Initialize $\theta$, the vector of initial parameters
for $iteration = 1, 2, \ldots$ do
    Sample task $t$, corresponding to loss $L_t$ on weight vectors $\eta$
    Compute $\eta = U_t^k(\theta)$, denoting $k$ steps of SGD or Adam
    Update $\theta \leftarrow \theta + \epsilon(\eta - \theta)$.
end for
```

As an alternative to the last step of the algorithm, $\theta - \eta$ is considered as a gradient and plugged into an Adam optimizer. In our



(a) Longitudinal profile distribution.

(b) First moment of the longitudinal profile.

(c) Second moment of the longitudinal profile.

(d) Lateral profile distribution.

(e) First moment of the lateral profile.

(f) Second moment of the lateral profile.

Fig. 8. Shower shape observables for a sample of 3'000 electrons of 64 GeV entering the $PBWO_4$ calorimeter at incident angle of 90°. The error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

Reptile implementation, we use k=10 as the number of steps of Adam optimization. We use 100'000 iterations for the Reptile training loop, and periodically every 1'000 iterations the performance of the model is checked on a particular geometry using a minibatch (100) of shower inputs. The meta-training was completed in approximately 10 hours on an NVIDIA Quadro RTX 8000 GPU[3] with a processing power of 4608 cores and clock speeds of 1395 MHz and 7001 MHz for graphics and memory respectively.

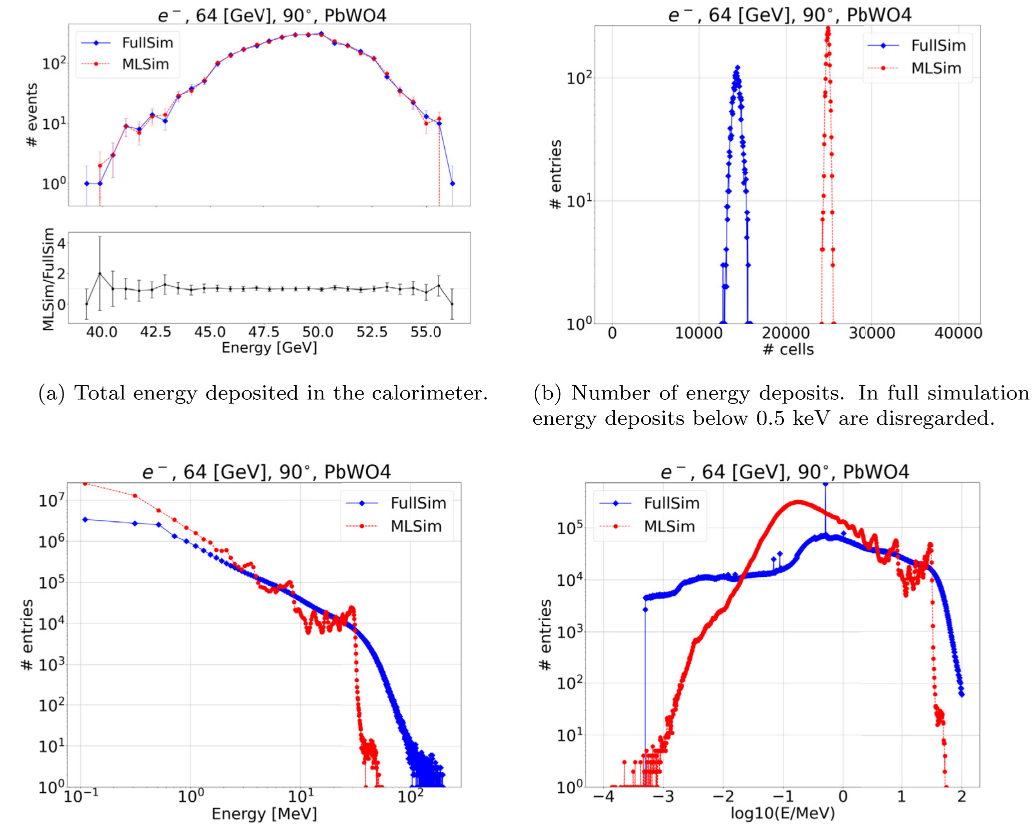
5. Experimental validation

The key idea behind the meta-learning approach is that instead of starting the training process from scratch on every new geometry, we can use the meta-knowledge, accumulated during the meta-training step, for a faster and more data-efficient adaptation. This refers to learning from prior experience and using this meta-knowledge to guide the search for optimal model parameters for a new geometry. In our Reptile training loop or meta-training, only two detector geometries of cylindrical SiW and SciPb are used to build the meta-knowledge. The remaining two geometries of cylindrical $PBWO_4$ and SiW FCC-ee are used to demonstrate the adaptation capabilities of the model. Once the adaptation is done, only the decoder network is used as a generator to perform inference. The input inference vector is constructed by sampling from a 10D Gaussian distribution. The condition vector is comprised of the values of the particle energy, the incident angle and the detector identifier. The results of inference, the fast simulation (MLSim) are compared to the full simulation (FullSim).

5.1. Fast adaptation with a cylindrical calorimeter

To test the adaptation on a new detector a full simulation sample of $PBWO_4$ cylindrical geometry is used. The weights of

[3] The meta-training used 20 GB of the total GPU memory (46 GB).



(a) Total energy deposited in the calorimeter.

(b) Number of energy deposits. In full simulation energy deposits below 0.5 keV are disregarded.

(c) Distribution of single mesh cell energy in linear scale, plotted on a logarithmic axis.

(d) Distribution of single mesh cell energy in logarithmic scale.

Fig. 9. Shower energy observables for a sample of 3'000 electrons of 64 GeV entering the PBWO$_4$ calorimeter at incident angle of 90°.

the model are first initialized with the meta-knowledge and the adaptation process is checked every 10 steps and stopped after 1000 adaptation steps. This allows identifying at which point further adaptation brings no further improvement to the validation distributions. The adaptation progress on the longitudinal profile distribution is illustrated in Fig. 7. After 50 steps the generated longitudinal profile is in poor agreement with full simulation, but with 360 steps there is a substantial agreement with a maximum relative difference at or below 1% that can be seen in the ratio plot (MLSim/FullSim)

The results of shower observables are presented in Fig. 8 for shower shapes and in Fig. 9 and Fig. 10 for energy distributions. Fig. 8a, Fig. 8b and Fig. 8c compare the longitudinal shower shape: profile distribution, its first moment, and its second moment, respectively. Fig. 8d, Fig. 8e and Fig. 8f compare the lateral shower shape: profile distribution, its first moment, and its second moment, respectively. All of those distributions show good agreement between the full and fast simulation. The longitudinal distribution of energy can also be compared by looking at the distribution of accumulated energy within one layer. Fig. 10 shows the energy deposited per layer for each of the 45 layers in the detector. Plotted with a logarithmic scale on the vertical axis, they demonstrate good agreement between full and fast simulation. It is also visible in Fig. 9a, of the distribution of the total energy deposited in the calorimeter. Except for the tails of the distribution, the total deposited energy is modelled within a few per cent accuracies of the full simulation.

Having a further look into energy validation, Fig. 9c and Fig. 9d show a distribution of energy deposited in a single cell of the mesh readout. Both of those histograms represent the same data, with a difference in the width of the bin: the first one shows a constant bin width in energy (drawn on a logarithmic axis) and the second one shows a constant bin width in the logarithm of energy. This distribution shows the biggest challenge of our ML model and the mismatch between full and fast simulation. There is an overabundance of low-energetic energy deposits in the calorimeter in fast simulation, and thus an underestimation of high-energetic deposits. This is also visible in the distribution of the number of created deposits shown in Fig. 9b. Fast simulation creates around 1.6 times more deposits than full simulation. This can be linked to the typical VAE issue of blurriness in the generated images [44,45]. The main objective of further studies is to address this issue by re-designing the underlying ML model. A second direction of studies will investigate what is the actual meaning of this cell energy mismodelling for the physics analysis. The readout mesh that we employ is highly granular and after mapping this mesh readout to the physical readout of the detector, cell energy distribution from fast simulation may get closer to full simulation. However, we do not expect it to improve considerably for high granular detectors, which are of particular importance for future experiments.

5.2. *Fast adaptation with a realistic calorimeter*

The second test of MetaHEP's capabilities is done on the SiW FCC-ee detector, which is very different from the other three detectors considered so far. As it is a detector with a realistic layout, it is much more complicated and differs considerably from the detectors used in the initial training. For the adaptation to this detector, the weights of the model are first initialized with the same meta-knowledge as in the previous test presented in Sec. 5.1, i.e., on the pre-trained model to cylindrical SiW and SciPB detectors. The adaptation process is checked every 10 steps and is stopped

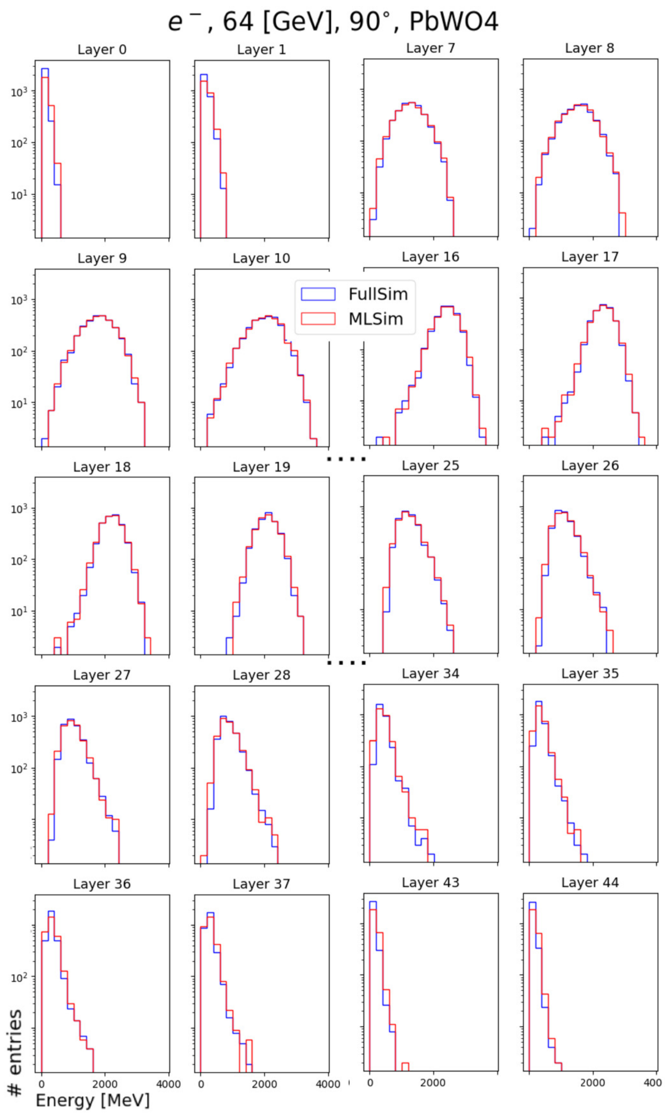


Fig. 10. Accumulated energy deposited in each layer of the $PBWO_4$ calorimeter for a sample of 3'000 electrons of 64 GeV entering at an incident angle of 90°. To enhance the clarity of the illustration, only a subset of the rows (four out of the total of nine that represent the 45 layers) are depicted.

after 2000 adaptation steps. Compared to the cylindrical $PbWO_4$ calorimeter, the number of adaptation steps necessary to get a very good agreement with the full simulation is almost 3 times higher. We attribute the extra effort required to more complex geometry and the significant differences in the layout from the detectors used for the initial training.

The results of shower observables after 1000 adaptation steps are presented in Fig. 11 for shower shapes and in Fig. 12 for energy distributions. Fig. 11a, Fig. 11b and Fig. 11c compare the longitudinal shower shape: profile distribution, its first moment, and its second moment, respectively. Fig. 11d, Fig. 11e, and Fig. 11f compare the lateral shower shape: profile distribution, its first moment, and its second moment, respectively. All of those distributions show a good agreement between the full and fast simulation. The same applies to the total energy distribution in the calorimeter, shown in Fig. 9a. Except for the tails of the distribution, the total deposited energy is modelled with a few per cent accuracies by the fast simulation. The single-cell energy distribution remains the least-well-modelled distribution. It is shown in Fig. 12c for constant energy bin width and in Fig. 12d. For the FCC-ee SiW detector, there is a slightly better agreement of the distributions than for the previously studied detector ($PbWO_4$), but there is still an overabundance of low-energetic cells. It is very prominent also in the distribution of the number of created deposits shown in Fig. 12b. Fast simulation creates around 5 times more deposits than full simulation. All conclusions about the model accuracy drawn in the previous section, Sec. 5.1, remain. It will be especially interesting to complete the study looking at the mapping to the physical readout of the detector, which is on an immediate list of plans.

Table 3
Comparison between traditional training and adaptation using the simplified calorimeter geometry $PBWO_4$. The value of speed-up is calculated w.r.t. 3900 steps of traditional training, comparing the results of the same accuracy. The experiments were run on a workstation equipped with an AMD EPYC 7282 16-core processor with a clock speed of 2800 MHz. The architecture is x86-64.

Approach	Number of steps	Time	Speedup
Traditional training	400	20 min	-
Traditional training	3900	3 h 15 min	-
Adaptation	400	20.5 s	×527

Table 4
Comparison between the adaptation phase of the cylindrical $PBWO_4$ and the realistic FCC-ee calorimeter. The experiments were run on a workstation equipped with an AMD EPYC 7282 16-core processor with a clock speed of 2800 MHz. The architecture is x86-64.

Detector	Number of adaptation steps	Time (s)
$PBWO_4$	360	19
FCC-ee	1000	51

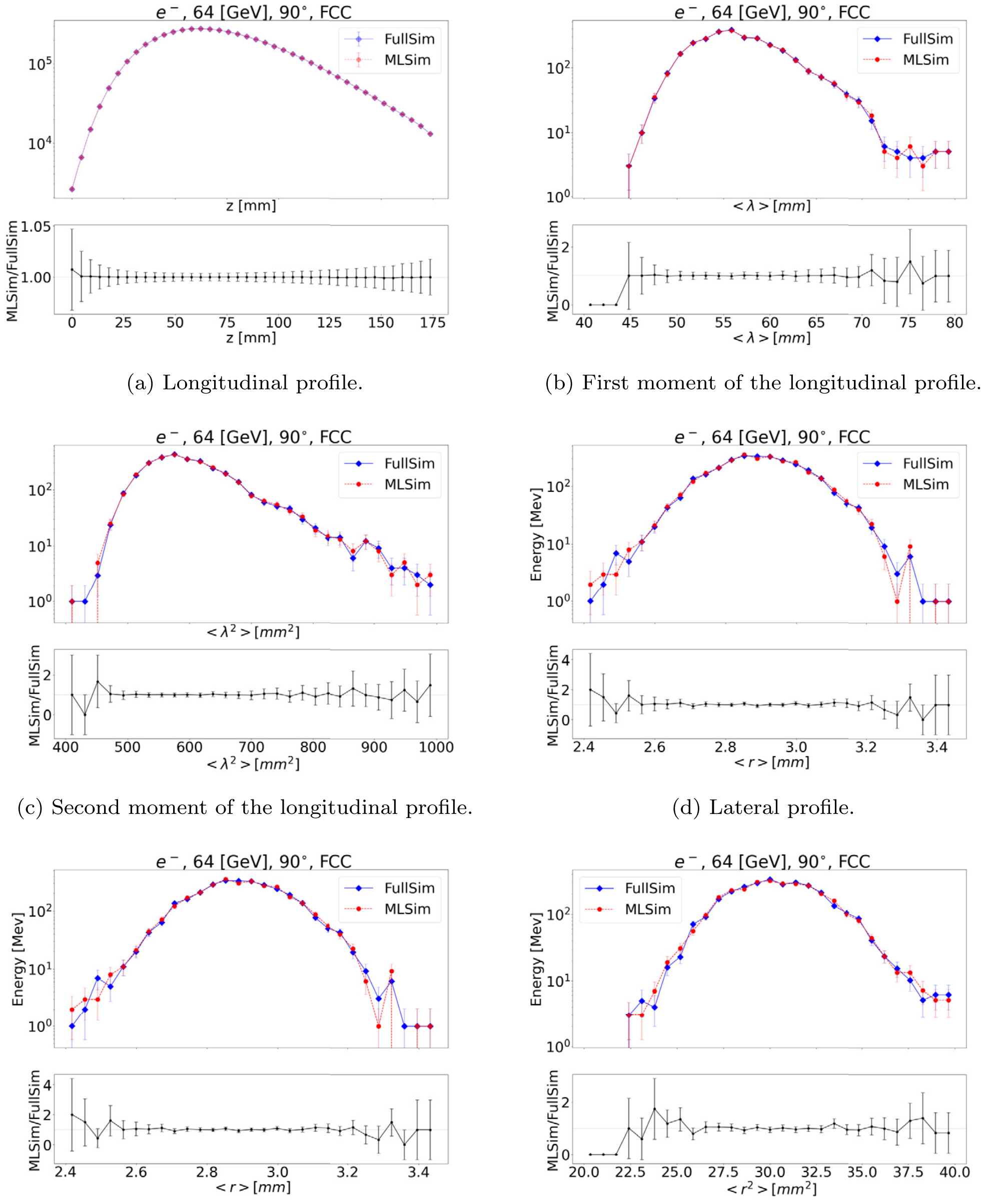
5.3. *Adaptation and traditional training*

Typically ML models are trained from scratch for a specific task (for a specific calorimeter in our case) using a fixed learning algorithm, which we will call "traditional" training. In order to show the strength of the MetaHEP approach, we first compare this meta-learning approach to a "traditional" training and then to a pre-trained model on a single geometry (SiW) using the simplified calorimeter geometry $PBWO_4$and the same model architecture, illustrated in Fig. 6.

Fig. 13 shows the longitudinal profile for the same number of training/adaptation steps and for a higher number of steps for the traditional case. One can conclude that adaptation, after the initial meta-training, provides a faster solution to converge. Traditional training requires many more steps to converge (here is shown to be almost 4000 steps).

On top of that, as shown in Table 3, the time to run 400 steps of adaptation is 20.5 s compared to 1200 s for the same number of 400 steps of the traditional training[4] on the same CPU. The ultimate speed-up, allowing to achieve the same accuracy, shows that adaptation is 500 times faster than traditional training. This number will depend on the detector, and Table 4 shows the adaptation time needed for cylindrical $PBWO_4$ detector described in Sec. 5.1 and the realistic FCC-ee calorimeter presented in Sec. 5.2. More time is needed for a more complicated detector, however, it is still a very fast adaptation (below 1 min). Adaptation naturally requires the meta-training step, which itself will take time, however, the ultimate goal is to provide meta-trained models for the community so that they can be reused by different users. Therefore this cost will be hidden from them and paid by the developers of those pre-trained models.

[4] For the meta-training approach, we save checkpoint models every 10 steps and for the traditional training every 100 steps. The same number of steps is considered in the comparison shown in Table 3.



(a) Longitudinal profile.

(b) First moment of the longitudinal profile.

(c) Second moment of the longitudinal profile.

(d) Lateral profile.

(e) First moment of the lateral profile.

(f) Second moment of the longitudinal profile.

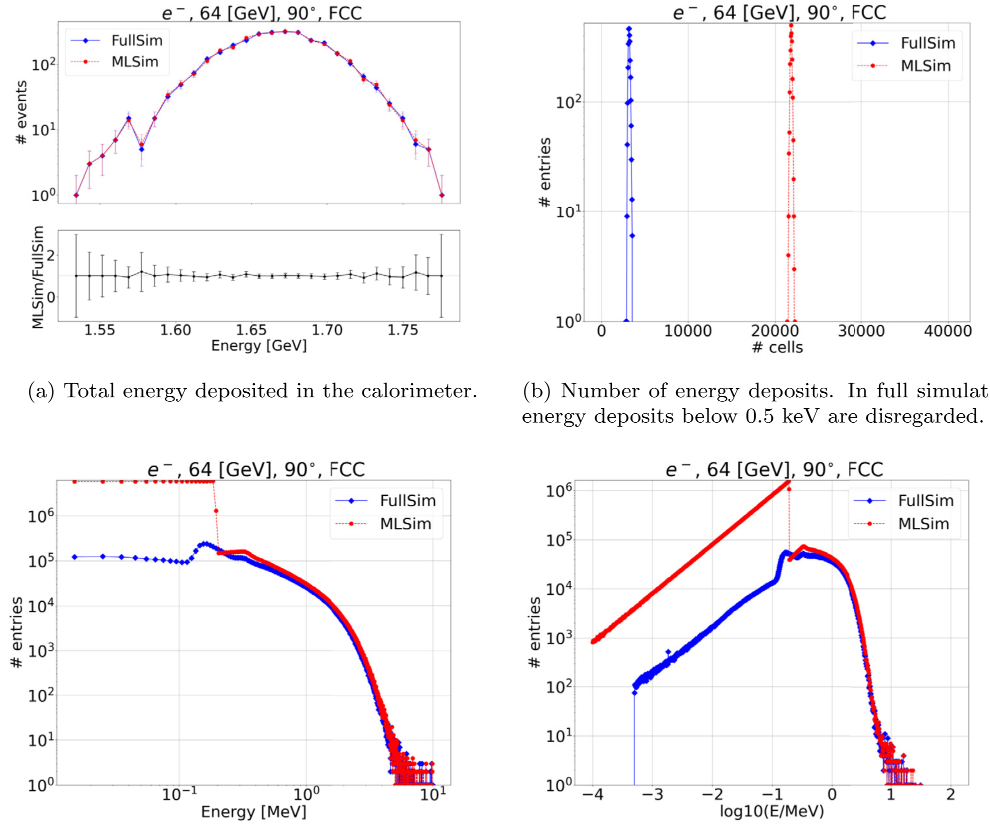
Fig. 11. Shower shape observables for a sample of 3'000 electrons of 64 GeV entering the FCC-ee SiW calorimeter at an incident angle of 90° and azimuthal angle of 90°. The error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

To show the strength of the MetaHEP, we compare the adaptation after the meta-training (using the two detector geometries SiW and SciPb) to a pre-trained model on a single detector geometry (SiW) and fine tuning it using the PBWO$_4$ geometry. Fig. 14 shows the longitudinal profile for this comparison. One can conclude that the adaptation after the meta-training phase is more efficient by comparing the same number of adaptation steps (400), the fine-tuning after a pre-training on a single geometry would need around four times steps (1600) to well model this distribution. This explains that the knowledge learned during the pre-training is very specific to the used detector and it would take more steps to converge (compared to a meta-training) in the fine-tuning process. This means that new detectors that are more different from the one used in the pre-training could be even more difficult to get the agreement right. Note that the time to run 400 steps for the adaptation after meta-training or the fine-tuning after pre-training on a single geometry remains the same.

For the FCC-ee, Fig. 15 shows the longitudinal profile comparing the adaptation after meta-training to traditional training to a fine-tuning after a pre-training on a single geometry. The number of steps for all three approaches for the FCC-ee is higher than the simplified calorimeter geometry PBWO$_4$ as this is a more complicated detector.

6. Conclusions

Deep learning has seen great successes in HEP for fast simulation of a specific calorimeter geometry. This approach needs to be trained on a large dataset and requires considerable resources. In this paper, we presented MetaHEP: a generalizable and reusable solution for fast shower simulation using a VAE meta-trained model with Reptile. It requires a custom simulation, to allow the creation of a virtual readout and scoring the energy in a highly granular mesh. The granularity of this mesh has a direct



(a) Total energy deposited in the calorimeter.

(b) Number of energy deposits. In full simulation energy deposits below 0.5 keV are disregarded.

(c) Distribution of single mesh cell energy in linear scale, plotted on a logarithmic axis.

(d) Distribution of single mesh cell energy in logarithmic scale.

Fig. 12. Shower energy observables for a sample of 3'000 electrons of 64 GeV entering the FCC-ee SiW calorimeter at an incident angle of 90° and azimuthal angle of 90°.

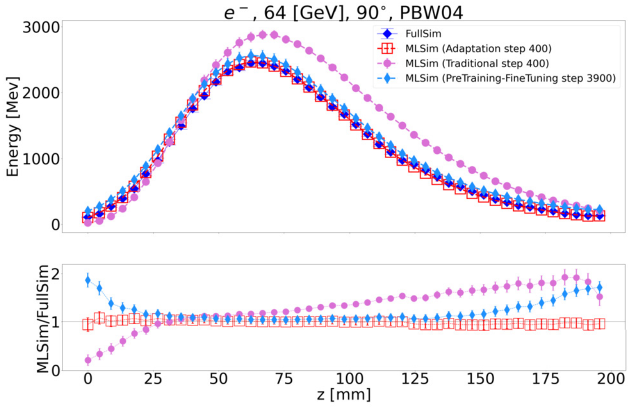


Fig. 13. Average longitudinal profile comparing the FullSim (blue) to the adaptation after meta training (red) to a traditional training with 400 (pink) and 3900 (light blue) steps, using a minibatch of 100 samples of 64 GeV electrons entering the $PBWO_4$ calorimeter at an incident angle of 90°. The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

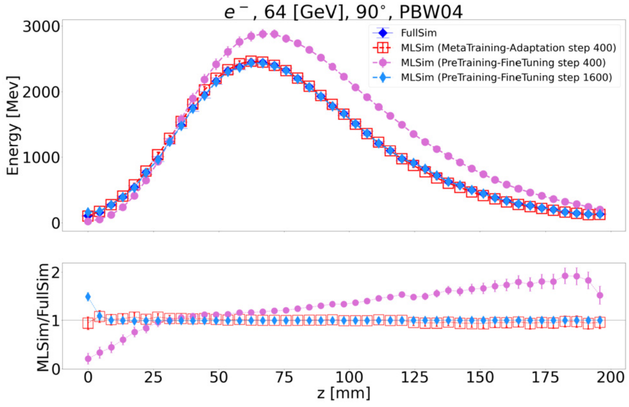


Fig. 14. Average longitudinal profile comparing the FullSim (blue) to the adaptation after meta training (red) to a fine tuning after a pre-training on a single detector geometry with 400 (pink) and 1600 (light blue) steps, using a minibatch of 100 samples of 64 GeV electrons entering the $PBWO_4$ calorimeter at an incident angle of 90°. The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

impact on the simulation time, as the deposits must be placed inside the detector. Fine-tuning of the granularity will be possible once MetaHEP is tested on more realistic geometries. The first results on the tested detectors show very promising results. There is a remaining challenge: the accurate modelling of single-cell distribution, but the study for its improvement is ongoing. Furthermore, it is linked to the choice of the underlying ML model and has no bearing on the application of meta-learning. The power of MetaHEP is its ability to generalize by learning parameter initialization that can be fine-tuned quickly on a new detector. We showed this is 500 times faster than training from scratch on datasets of this detector. In future work, we will investigate many input geometries in the meta-training, in order to achieve optimized weights for new test detectors, seeking to improve further the adaptation step. This is of great importance for detectors that are being designed for future experiments.

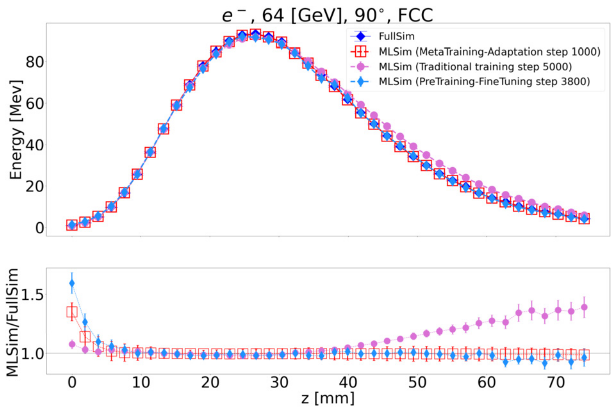


Fig. 15. Average longitudinal profile comparing the FullSim (blue) to the adaptation after meta training (red) to a traditional training (pink) and to a fine tuning after a pre-training on a single detector geometry (light blue), using a minibatch of 100 samples of 64 GeV electrons entering the FCC-ee calorimeter at an incident angle of 90°. The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dalila Salamani reports financial support was provided by CERN-EP-RD.

Data availability

A link to the data has been provided in the document.

Acknowledgements

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments [46] and has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 101004761.

References

[1] Large Hadron Collider, Longer term LHC schedule, http://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm, 2022.

[2] C. Bozzi, LHCb Computing Resource usage in 2021, LHCb-PUB-2022-011, CERN-LHCb-PUB-2022-011 CERN, Geneva, Feb 2022, http://cds.cern.ch/record/2802075.

[3] ATLAS Collaboration, ATLAS Software and Computing HL-LHC Roadmap (No. CERN-LHCC-2022-005), 2022.

[4] C.M.S. Offline, Software and Computing, CMS Phase-2 Computing Model: Update Document, CMS-NOTE-2022-008, CERN-CMS-NOTE-2022-008, CERN, Geneva, Jul 2022, https://cds.cern.ch/record/2815292.

[5] LHCb Collaboration, LHCb CPU Usage Forecast, https://cds.cern.ch/record/2696552, Oct 2019.

[6] A. Boehnlein, F. Simon, F. Hernandez, D. Britton, R. Bolton, C. Biscarat, et al., HL-LHC Software and Computing Review Panel, 1st Report (No. CERN-LHCC-2020-012), 2020.

[7] ATLAS Collaboration, AtlFast3: the next generation of fast simulation in ATLAS, arXiv preprint arXiv:2109.02551, 2021.

[8] S. Abdullin, P. Azzi, F. Beaudette, P. Janot, A. Perrotta, CMS Collaboration, The fast simulation of the CMS detector at LHC, J. Phys. Conf. Ser. 331 (3) (December 2011) 032049, IOP Publishing.

[9] V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin, E. Zakharov, Generative Models for Fast Calorimeter Simulation: the LHCb Case, EPJ Web of Conferences, vol. 214, EDP Sciences, 2019, p. 02034.

[10] Geant4 Collaboration, Geant4–a simulation toolkit, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 506 (3) (2003) 250–303.

[11] M. Rama, G. Vitali, Calorimeter fast simulation based on hit libraries LHCb Gauss framework, EPJ Web Conf. 214 (2019) 02040, https://doi.org/10.1051/epjconf/201921402040.

[12] E. Barberio, J. Boudreau, B. Butler, S.L. Cheung, A. Dell'Acqua, A. Di Simone, et al., Fast simulation of electromagnetic showers in the ATLAS calorimeter: frozen showers, J. Phys. Conf. Ser. 160 (1) (April 2009) 012082, IOP Publishing.

[13] G. Grindhammer, M. Rudowicz, S. Peters, The fast simulation of electromagnetic and hadronic showers, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 290 (2–3) (1990) 469–488.

[14] M. Paganini, L. De Oliveira, B. Nachman, CaloGAN: simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks, Phys. Rev. D 97 (1) (2018) 014021, https://doi.org/10.1103/PhysRevD.97.014021.

[15] ATLAS Collaboration, Deep generative models for fast shower simulation in ATLAS, ATL-SOFT-PUB-2018-001, https://cds.cern.ch/record/2630433, 2018.

[16] M. Erdmann, J. Glombitza, T. Quast, Precise simulation of electromagnetic calorimeter showers using a Wasserstein generative adversarial network, Comp. Softw. Big Sci. 3 (1) (2019) 1–13.

[17] C. Krause, D. Shih, Caloflow: fast and accurate generation of calorimeter showers with normalizing flows, arXiv preprint arXiv:2106.05285, 2021.

[18] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger, Getting high: high fidelity simulation of high granularity calorimeters with high speed, Comp. Softw. Big Sci. 5 (1) (2021) 1–17.

[19] The ATLAS Collaboration, AtlFast3: the next generation of fast simulation in ATLAS, Comp. Softw. Big Sci. 6 (7) (2022).

[20] ATLAS Collaboration, Deep generative models for fast shower simulation in ATLAS, ATLAS-SIMU-2020-04-002, https://cds.cern.ch/record/2836604, 2018.

[21] D.K. Naik, R.J. Mammone, Meta-neural networks that learn by learning, in: [Proceedings 1992] IJCNN International Joint Conference on Neural Networks, vol. 1, IEEE, June 1992, pp. 437–442.

[22] J. Schmidhuber, Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook, Doctoral dissertation, Technische Universität München, 1987.

[23] J. Schmidhuber, A possibility for implementing curiosity and boredom in model-building neural controllers, in: Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats, 1991, pp. 222–227.

[24] J. Storck, S. Hochreiter, J. Schmidhuber, Reinforcement driven information acquisition in non-deterministic environments, in: Proceedings of the International Conference on Artificial Neural Networks, vol. 2, Paris, October 1995, pp. 159–164.

[25] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, July 2017, pp. 1126–1135.

[26] A. Nichol, J. Achiam, J. Schulman, On first-order meta-learning algorithms, arXiv preprint arXiv:1803.02999, 2018.

[27] FCC Collaboration, FCC physics opportunities: future circular collider conceptual design report volume 1, Eur. Phys. J. C 79 (6) (2019) 474.

[28] The ILD Collaboration, The ILD detector at the ILC, arXiv:1912.04601, 2019.

[29] FCC Collaboration, FCC-ee: the lepton collider: future circular collider conceptual design report volume 2, Eur. Phys. J. Spec. Top. 228 (2) (2019) 261–623.

[30] D. Arominski, J.J. Blaising, E. Brondolin, D. Dannheim, K. Elsener, F. Gaede, et al., A detector for CLIC: main parameters and performance, arXiv preprint arXiv:1812.07337, 2018.

[31] FCC Collaboration, FCCDetectors, FCCeeCLD, https://github.com/HEP-FCC/FCCDetectors/blob/v0.1pre09/Detector/DetFCCeeCLD/compact/FCCee_o2_v02/ECalBarrel.xml.

[32] Geant4 Collaboration, Par04 example. Machine learning inference for fast simulation in Geant4, https://gitlab.cern.ch/geant4/geant4/-/tree/master/examples/extended/parameterisations/Par04.

[33] J. Cervantes, J. Faltova, G. Ganis, C. Helsens, J. Hrdinka, C. Neubüser, et al., A Software Framework for FCC Studies: Status and Plans, EPJ Web of Conferences, vol. 245, EDP Sciences, 2020, p. 05018.

[34] G. Ganis, C. Helsens, V. Völkl, Key4hep, a framework for future HEP experiments and its use in FCC, Eur. Phys. J. Plus 137 (1) (2022) 149.

[35] High Granularity Electromagnetic Calorimeter Shower Images [Data set], Zenodo https://doi.org/10.5281/zenodo.6082201.

[36] Lightweight Trained Neural Network Library, https://github.com/lwtnn/lwtnn.

[37] Open Neural Network Exchange runtime, https://github.com/microsoft/onnxruntime.

[38] PyTorch C++ frontend, https://pytorch.org/cppdocs/frontend.html.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[40] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114, 2013.

[41] S. Voloshynovskiy, M. Kondah, S. Rezaeifar, O. Taran, T. Holotyak, D.J. Rezende, Information bottleneck through variational glasses, arXiv preprint arXiv:1912.00830, 2019.

[42] L. Dinh, D. Krueger, Y. Bengio, Nice: non-linear independent components estimation, arXiv preprint arXiv:1410.8516, 2014.

[43] M.I. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, Introduction to variational methods for graphical models, Mach. Learn. 37 (1999) 183–233.

[44] Michele Sebag, V. Berger, Michèle Sebag, Variational Auto-Encoder: Not All Failures Are Equal, 2020.

[45] S. Zhao, J. Song, S. Ermon, Towards Deeper Understanding of Variational Autoencoding Models, 2017.

[46] M. Aleksa, C. Joram, P. Farthouat, A. Onnela, J. Blomer, C. Gargiulo, et al., Strategic R&D; Programme on Technologies for Future Experiments (No. CERN-OPEN-2018-006), 2018.