Meta-learning for multiple detector geometry modeling

Dalila Salamani, Anna Zaborowska, Witold Pokorski
CERN, Switzerland

E-mail: dalila.salamani@cern.ch, anna.zaborowska@cern.ch, witold.pokorski@cern.ch

Abstract.
The simulation of the passage of particles through the detectors of High Energy Physics (HEP) experiments is a core component of any physics analysis. A detailed and accurate simulation of the detector response using the Geant4 toolkit is a time and CPU consuming process. With the upcoming high luminosity LHC upgrade, with more complex events and a much increased trigger rate, the amount of required simulated events will increase. Several research directions investigated the use of Machine Learning based models to accelerate particular detector response simulation. This results in a specifically tuned simulation and generally these models require a large amount of data for training. Meta learning has emerged recently as fast learning algorithm using small training datasets. In this paper, we propose a meta-learning model that "learns to learn" to generate electromagnetic showers using a first-order gradient based algorithm. This model is trained on multiple detector geometries and can rapidly adapt to a new geometry using few training samples.

1. Introduction

High energy physics (HEP) experiments relies on Monte Carlo (MC) simulation as a basis for data analysis and detector design. This simulation, called full or detailed simulation, allows us to understand how detector design can affect measurements and physics processes, to validate the theory models, and also to compare against collision data. Geant4 [1] toolkit is used commonly in HEP to simulate particle passage through the detector. It models the interactions of particles with matter at the microscopic level. This simulation process is inherently slow, especially for dense calorimeters and can present a bottleneck for HEP experiments. Fast simulation techniques address this challange by parameterizing in a single stage the showering process instead of simulating the step by step interactions with matter.

In recent years deep learning techniques have achieved remarkable successes in various domain applications. In HEP, deep learning techniques have been used for fast simulation of detector responses such as in [2], [3] and [4]. These studies focused on a single detector geometry using a large amount of data for training, needed to achieve good performance. In this paper, a work on a generalizable and reusable solution is presented. The model is trained on multiple detector geometries and adapts quickly to a new geometry.

Meta-learning [5] is a learning to learn technique which takes a distribution of tasks, where each task corresponds to a learning problem with its set of examples, and it produces a quick learner which can generalize from small amounts of data examples. Model-agnostic meta-learning (MAML) [6] is a meta-learning algorithm which optimization problem is learning

the initialization parameters (weights) of a neural network. These parameters constitute the meta-knowledge and their learning process is the meta-training step. They can be used as an initialization weights of the model and subsequently tuned on a new task. Reptile [7] 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 take less computations and memory, while the optimization problem remains the same as for MAML.

2. Multiple detector geometry modeling workflow

2.1. Workflow

Fig. 1 represents the different modules of the multiple detector geometry modeling workflow. The preprocessing module allows us to prepare (data cleaning, scaling,...) the full simulated data generated using Geant4 applications, creating a universal shower representation. It also encodes the geometry, energy of the particle initiating the shower, and the angle at which particle enters the detector. The preprocessed data is then used by the generative model for training. In order to search for the best set of hyperparameters of the model, the AutoML (Automatic Machine Learning) module is used. After training, optimization and validation the model is converted into a format such as JSON or ONNX that can be used for inference in C++ (Geant4 inference). Par04 [8] is a Geant4 example which demonstrates how to use a generative model incorporating inference libraries such as LWTNN [9] and ONNX runtime [10]. Optimization techniques such quantization and graph optimizations are used to further reduce the memory footprint of the model at the inference time. Moreover, optimization techniques such as pruning [11] and knowledge distillation [12] are used at training time to reduce the number of trainable parameters. In this paper, we will only focus on the generative model component.

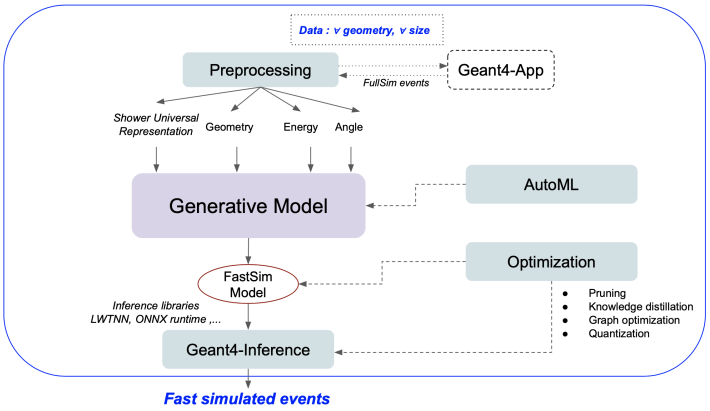

Figure 1. Multiple detector geometry modeling workflow.

2.2. Generative model

This generative model is based on the Reptile meta-learning algorithm. 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 a single detector geometry. Different geometries correspond to different tasks. For a randomly sampled task t, the learning minimizes $E_t[L_t(U_t^k(\theta))]$, 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. Reptile, described in Alg. 1, is an iterative algorithm which starts by sampling a task from the distribution of tasks, trains on the task, and then moves weights of the model towards the trained weights [7]. The algorithm is as follows

Algorithm 1 Reptile algorithm [7].

Initialize θ, the vector of initial parameters

for iteration=1,2,... **do**

Sample task t, corresponding to loss L_t on weight vectors η

Compute $\eta = U_t^k(\theta)$, denoting k steps of SGD or Adam

Update $\theta \leftarrow \theta + \epsilon(\eta - \theta)$.

end for

3. Calorimeter geometry and training dataset

3.1. Calorimeter geometry

The calorimeter under consideration, is a setup of concentric cylinders of layers. Each layer consists of active and passive material (or just active material for homogeneous calorimeters). Energy deposits are scored in the detector using the cylindrical readout structure, centred around the particle momentum, as shown in Fig. 2. This readout of energy deposits is not connected to the physical detector segmentation and allows to obtain the high granularity independent of the angle at which a particle enters the detector.

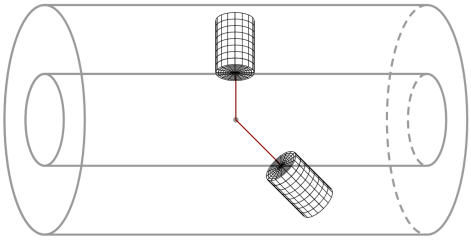

Figure 2. Energy deposits are scored in cylindrical readout around particle momentum.

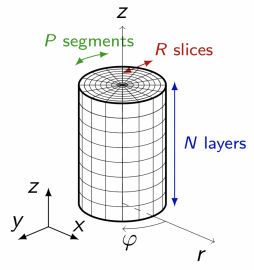

Figure 3. Cylindrical (r, ϕ, z) segmentation.

For SiW it is 0.3mm silicon and 1.4mm tungsten as passive absorbe

In this paper we use, for meta-training, two detector geometries. The first consists of 90 layers, with 0.3 mm silicon as active material, and 1.4 mm tungsten as passive absorber, called here SiW geometry. The second geometry is built of 45 layers, with 4.4 mm of lead and 1.2 mm of scintillator, called here a SciPb geometry. The size of the cylindrical readout has been optimised to contain (on average) 95 % of energy of 1 TeV electrons. The number of readout cells is $R \times P \times N = 18 \times 50 \times 45 = 40500$ representing r, ϕ, z cylindrical segmentation as shown in Fig. 3. The size of a single cell has been chosen to correspond to (approximately) 0.25 Moliere radius along the r axis and 0.5 radiation length along the z axis. Such representation of shower in different geometries can facilitate its generalization. Particle momentum that is used to define the orientation and the placement of the cylindrical readout (which will differ from particle to particle) is measured at the entrance to the calorimeter.

3.2. Training dataset

The full simulation samples for the two detector geometries (SiW and SciPb) are showers of electrons generated with an energy range from 1 GeV to 1 TeV (in powers of 2) and angles from 50° to 90° (in a step of 10°). Entrance angle of 90° means perpendicular to the z-axis. Ten thousand particle showers are simulated for each primary particle energy and angle. The dataset is available on Zenodo [13].

In order to demonstrate that the model can learn from a small amount of datasets, only 30% of the available statistics (for each energy and angle) are used for the training.

4. ML model and training

The model used in this paper is a Variational Autoencoder (VAE) [14], a deep learning generative model. The VAE is composed of two stacked deep neural networks acting as encoder and decoder. The encoder learns a mapping from the input space to an unobserved or latent space in which a lower dimensional representation of the full simulation is learned. The decoder learns the inverse mapping, thus reconstructing the original input from this latent representation. The encoded distributions are constrained to be Gaussian distributions and the encoder is tasked to return the mean and the covariance matrix that describe those distributions. The loss function that is optimized during the training of the VAE is composed of a regularization loss to minimize the Kulback-Leibler divergence between encoded distributions and prior Gaussian distributions, and a reconstruction loss to minimize the error by computing the binary cross-entropy between the input and its reconstruction version using the latent representation. The VAE architecture, shown in Fig. 4, comprises 4 hidden layers with width of 100,50,20,15 and 15,20,50,100 for the encoder and decoder respectively.

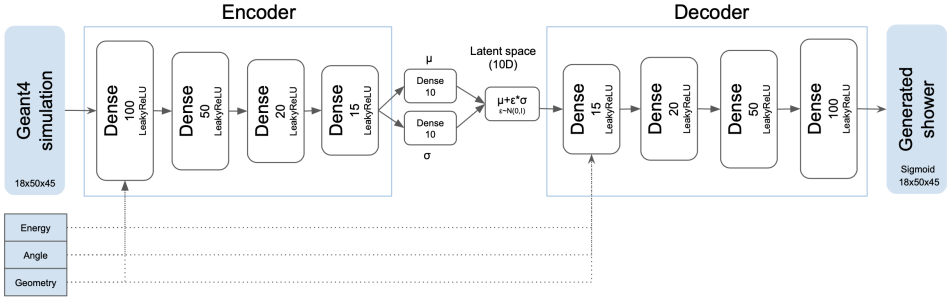

Figure 4. VAE model architecture.

5. Inference and validation

After training, the decoder can be used as a generator to perform the inference. The input inference vector is constructed by sampling from a 10D Gaussian distribution. The condition vector comprises condition values of particle energy, entrance angle, and encoding of the calorimeter geometry.

5.1. Validation on a meta-training geometry

Reptile is demonstrated to converge towards a solution θ that is close (in Euclidean distance) to each task manifold of optimal solutions [7]. This section demonstrates how the meta-knowledge can be applied on a meta-training task (geometry) and which can be fine tuned using few adaptation steps for a better performance. Fig. 5 (Fig. 6) show the longitudinal profile for 64 (512) GeV particle with 90° entrance angle for the SiW geometry. Because this is one of the geometries that was used during the meta-learning step, one can see in the case where only the meta-knowledge is used (Step 0), there is fair agreement. After performing the adaption step with 100 steps, a better agreement is observed.

5.2. Fast adaptation to a new geometry

The key idea behind using a meta-learning approach is that instead of starting the training process from scratch on each new geometry, one can use the meta-knowledge for a faster and

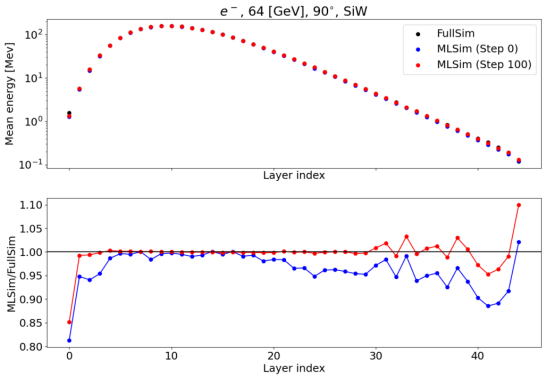

Figure 5. Longitudinal profile for 64 GeV particles.

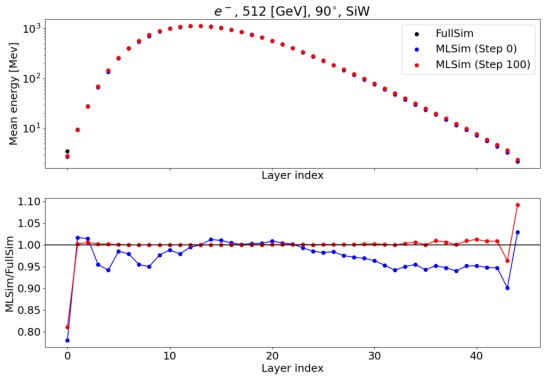

Figure 6. Longitudinal profile for 512 GeV particles.

more data-efficient adaptation. To test on a new geometry, full simulation samples are generated using a lead tungstate ($PBWO_4$) geometry with the same segmentation of $R \times P \times N = 18 \times 50 \times 45$ as shown in Fig. 3. The size of cells is chosen to correspond to the same granularity in units of Moliere radius and radiation length. The weights of the model are first initialized with the meta-knowledge and the adaptation step is tested for a 1000 steps with checkpoints every 10 steps. The adaptation progress is illustrated in Fig. 7 and Fig. 8, using the longitudinal profile distribution with 50 and 360 steps respectively. After 50 steps the generated longitudinal profile is in poor agreement with full simulation, but with 360 steps there is a very good agreement to the full simulation.

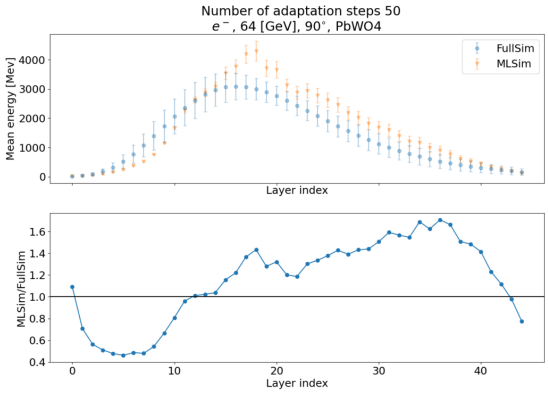

Figure 7. Longitudinal profile for 64 GeV particles with 50 adaptation steps.

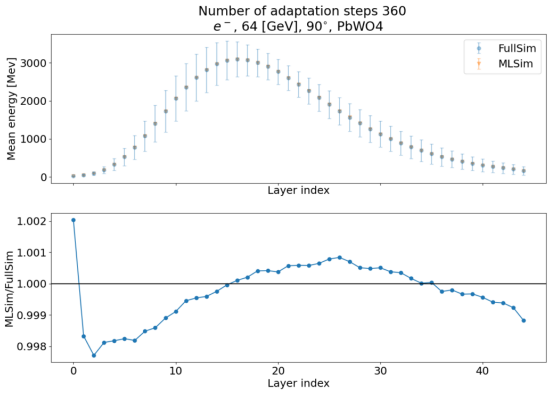

Figure 8. Longitudinal profile for 64 GeV particles with 360 adaptation steps.

5.3. Adaptation and traditional training

The strength of the presented approach is best visible in comparison to a "traditional" training where a model starts the learning process from scratch. Fig. 9 shows the longitudinal profile for the compared models using the same architecture, illustrated in Fig. 4, and the same number of training/adaptation steps. This shows that the adaptation step, after meta-training, can provide

a faster solution to converge. A traditional training requires many more steps to converge. On top of that, the time to run 400 steps of adaptation is about 20.5 s compared to 1200 s for the same number of 400 steps of the traditional training on the same machine.

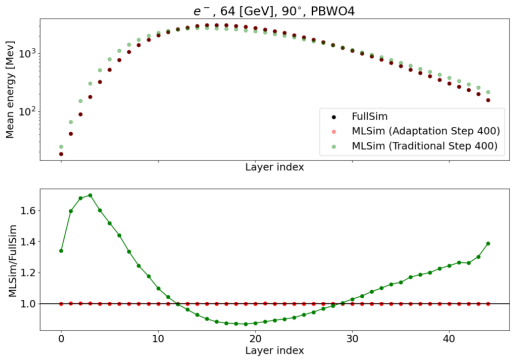

Figure 9. Longitudinal profile for 64 GeV particles.

6. Conclusion

This paper presents the first application of meta-learning VAE for simulating particle showers in different detectors. Two detectors have been used for the meta-training step in order to learn parameters initialization. These parameters are then used as a starting point for the fast adaptation step where a new geometry has been tested. The results show good agreement with showers from the full simulation. This paper demonstrates the feasibility of using a meta-learning approach as a generalizable and reusable solution for fast calorimeter simulation. It opens the possibility to quickly employ ML-based fast calorimeter simulation, which can be of special interest for detectors that are in design phase, like those on the future accelerators.

References

[1] S. Agostinelli et al. GEANT4: A Simulation toolkit. Nucl. Instrum. Meth., A506:250–303, 2003. doi:10.1016/S0168-9002(03)01368-8.
[2] ATLAS Collaboration. Deep generative models for fast shower simulation in ATLAS. ATL-SOFT-PUB-2018-001, 2018. URL:https://cds.cern.ch/record/2630433.
[3] M. Erdmann et al. Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network. February 2020. arXiv:1807.01954
[4] C. Krause et al. CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows. August 2021. arXiv:1807.01954
[5] DK. Naik et al. Meta-neural networks that learn by learning. In: International Joint Conference on Neural Networks, IEEE, IJCNN'1992, vol 1, pp 437–442.
[6] C. Finn et al. Model-agnostic Meta-learning for Fast Adaptation of Deep Networks. In: Proceedings of the 34th International Conference on Machine Learning, JMLR.org, ICML'2017, pp 1126–1135.
[7] A. Nichol et al. On First-Order Meta-Learning Algorithms. 2018. arXiv preprint arXiv:180302999.
[8] Geant4 collaboration. Par04 example. Machine learning inference for fast simulation in Geant4. URL:https://gitlab.cern.ch/geant4/geant4/-/tree/master/examples/extended/parameterisations/Par04.
[9] Lightweight Trained Neural Network library. URL:https://github.com/lwtnn/lwtnn.
[10] Open Neural Network Exchange runtime. URL:https://github.com/microsoft/onnxruntime.
[11] MC. Mozer et al. Skeletonization: A Technique for Trimming the Fat from a Network via RelevanceAssessment. 1988. Advances in Neural Information Processing Systems (NIPS) pp 107–115.
[12] H. Geoffrey et al. Distilling the knowledge in a neural network. 2015. arXiv:1503.02531.
[13] D.Salamani, A.Zaborowska. (2022). High Granularity Electromagnetic Calorimeter Shower Images [Data set]. Zenodo. URL:https://doi.org/10.5281/zenodo.6082201.
[14] D. P Kingma, M. Welling. Auto-Encoding Variational Bayes. December 2013. arXiv:1312.6114.