Meta-learning for fast simulation of multiple calorimeter responses

Dalila Salamani, Anna Zaborowska & Witold Pokorski
CERN

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (CERN-OPEN-2018-006)

EP-R&D
Programme on Technologies for Future Experiments

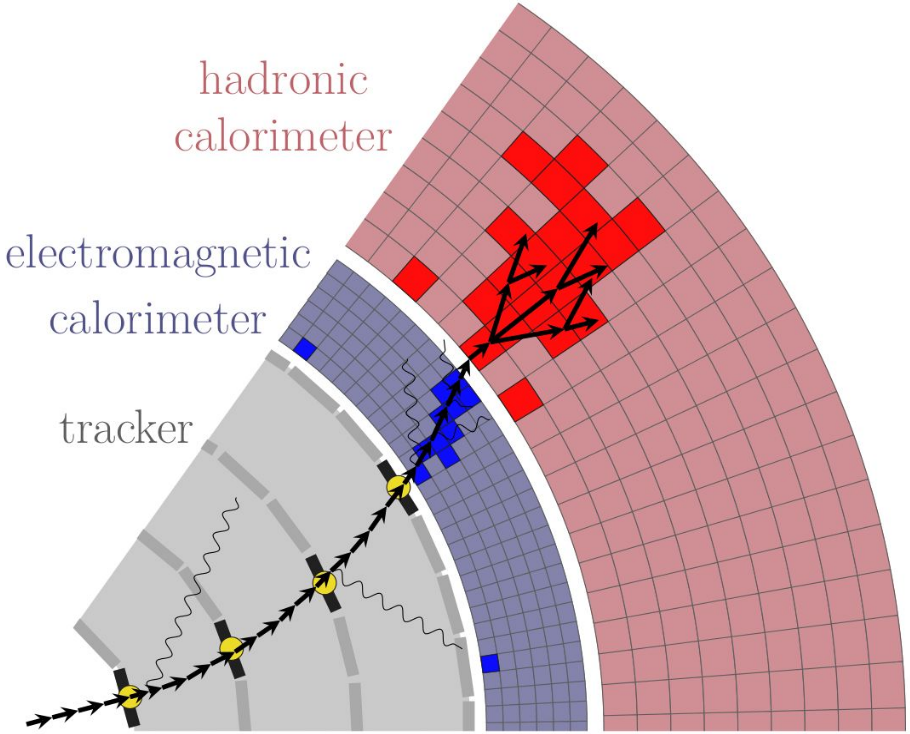
AMLD, Lausanne, 28/03/2022

Geant4 : a simulation toolkit

Website

- Toolkit born in the 1990s, providing a highly flexible simulation framework in C++
- Geant4 mission
  - Provide production-quality simulation toolkit and support to various experiments
  - Improve the physics models with better precision and energy range extensions
  - Improve the overall computational performance of simulation
  - Provide long-term maintenance & sustainability

Simulating particle interactions with Geant4 (Full Simulation)



[Geant4]

The need for fast simulation methods at Large Hadron Collider

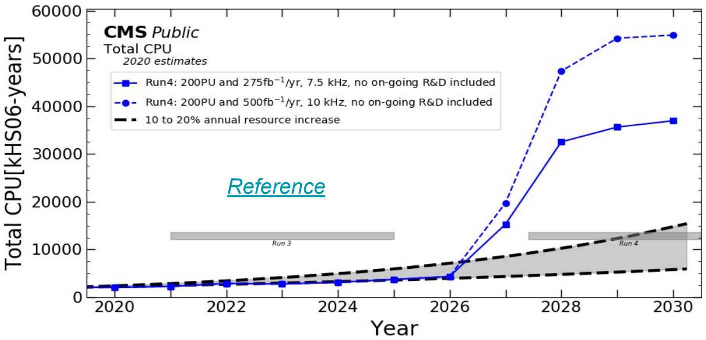
- Speed-up simulation to generate more data within the same CPU time

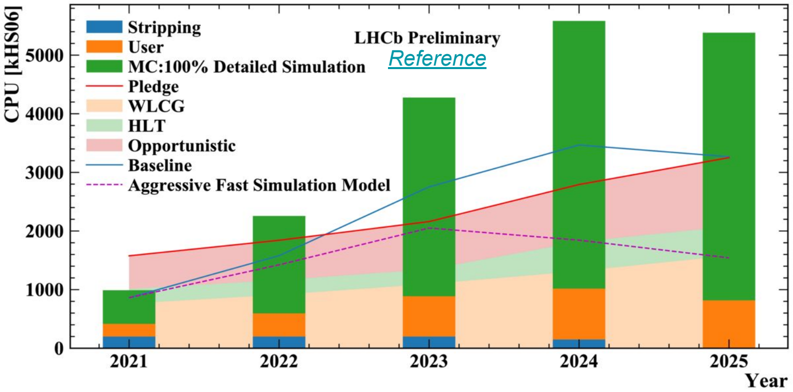
ALICE

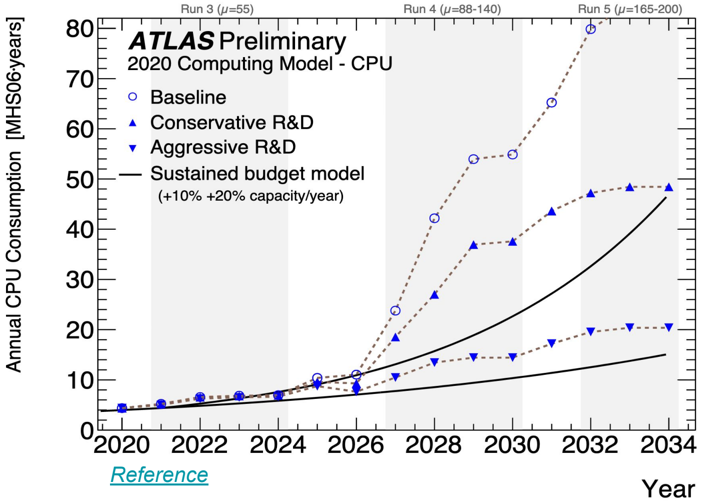
Resource	Sim	Reco	Data Analysis
CPU	56%	7%	37%(*)
Disk	54%	39%	7%

Percentage of the CPU and disk storage from April 2019 to January 2020

Reference

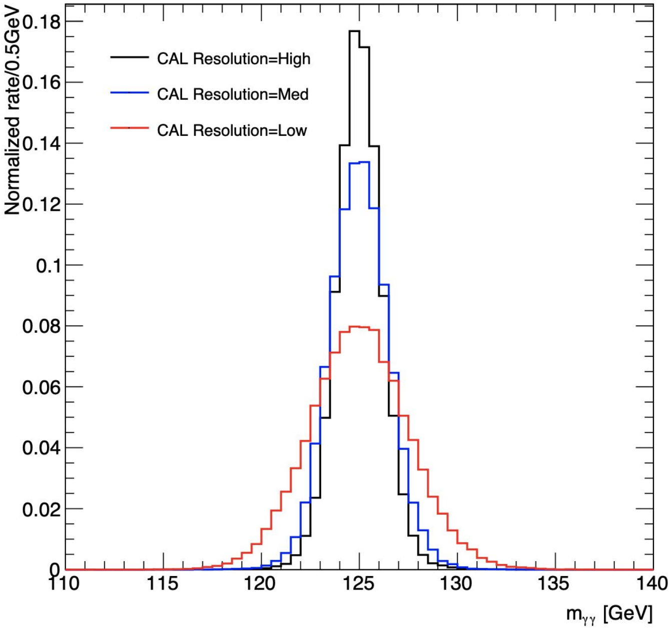






The need for fast simulation methods at future experiments

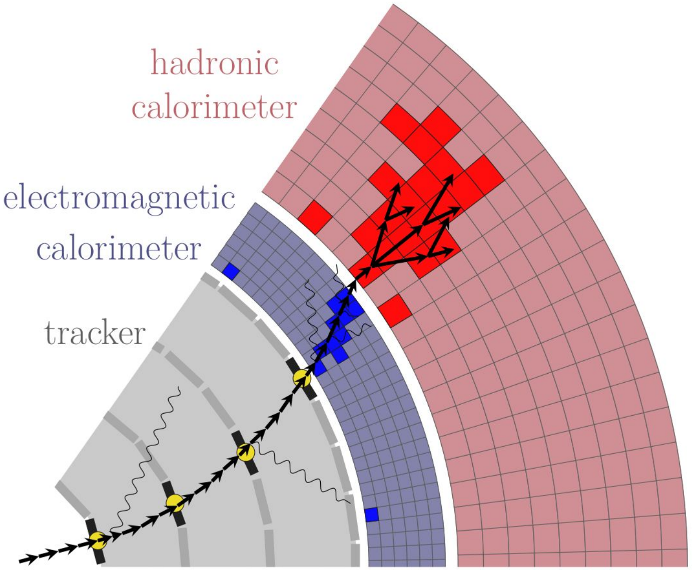
- Physics studies & detector performance benchmarks
  - Eg. to study impact of detector performance on physics observables (in the plot: calorimeter resolution on Higgs mass)



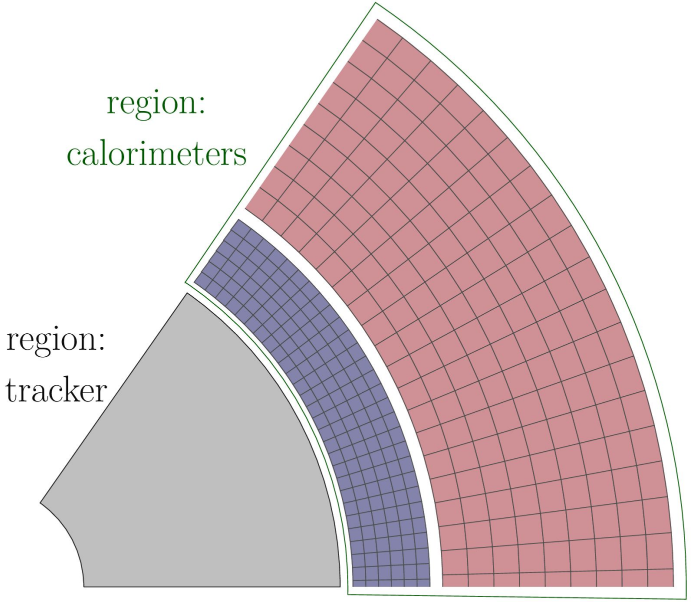
Reference

How to fast simulate particles? Machine Learning

FullSim



FastSim



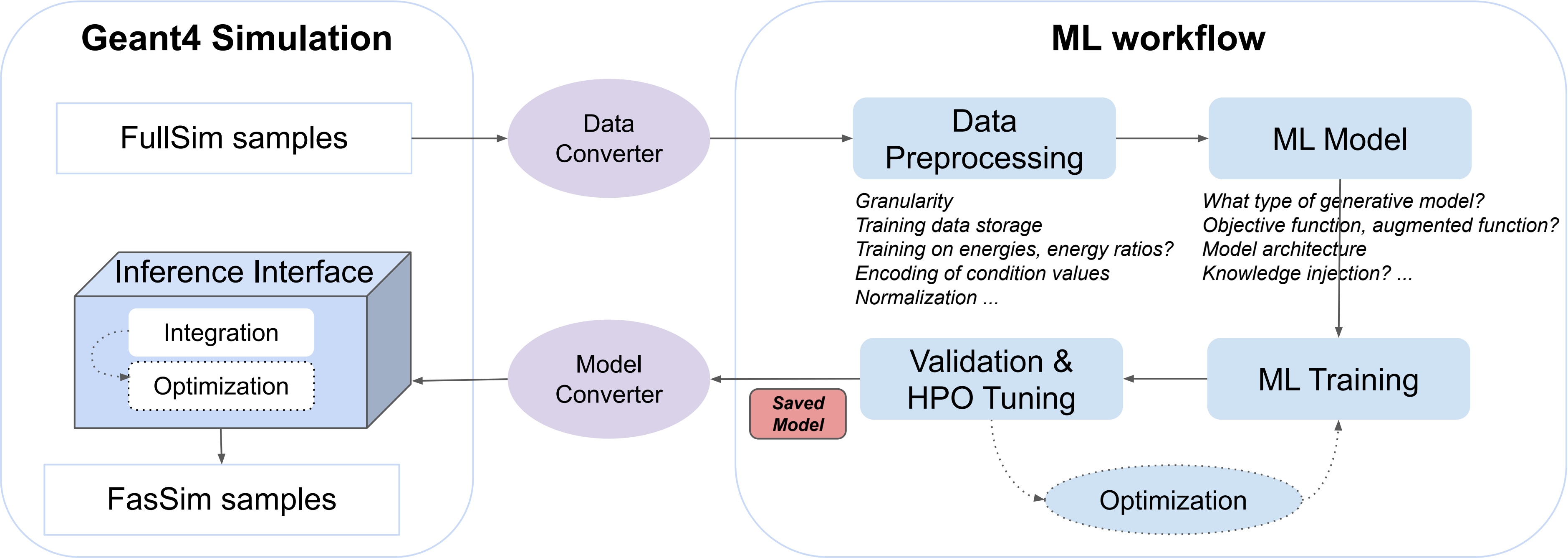
Shortcut standard tracking & detailed simulation *[Geant4]*

```python
# MLFastCaloSim
 def MLFastCaloSim(geometry,type,energy,angle):
     return P(shower|geometry,type,energy,angle)
```

Generalizable & reusable solution

- *Trained on multiple detector geometries*
- *Adapt quickly to a new geometry*

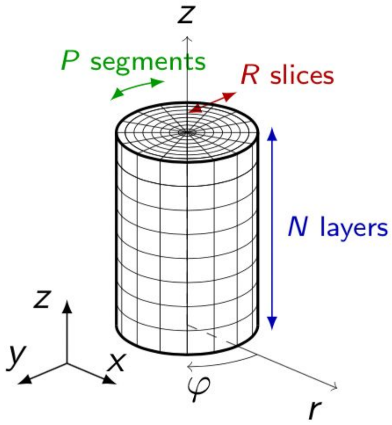
From ML training to Geant4 fast simulation

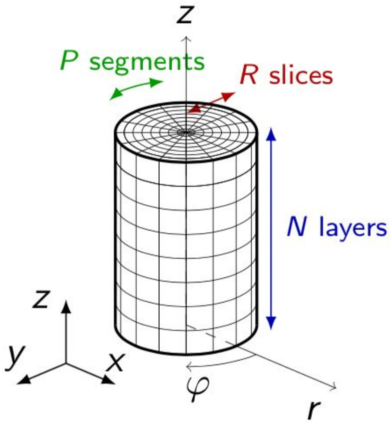


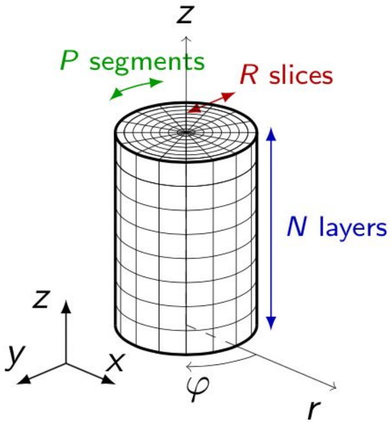
C++ environment
Inference : CPU, GPU ...

Python environment
CPU, GPU, ...

Calorimeter geometries (1/2)

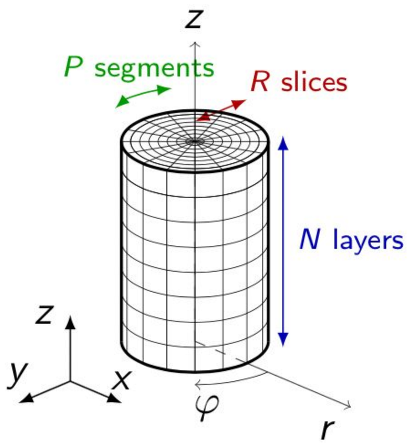




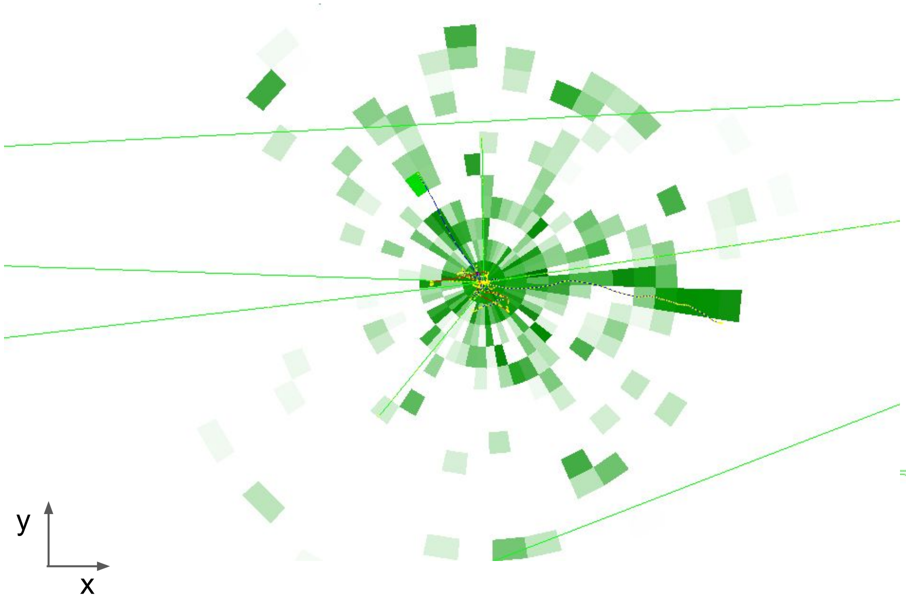


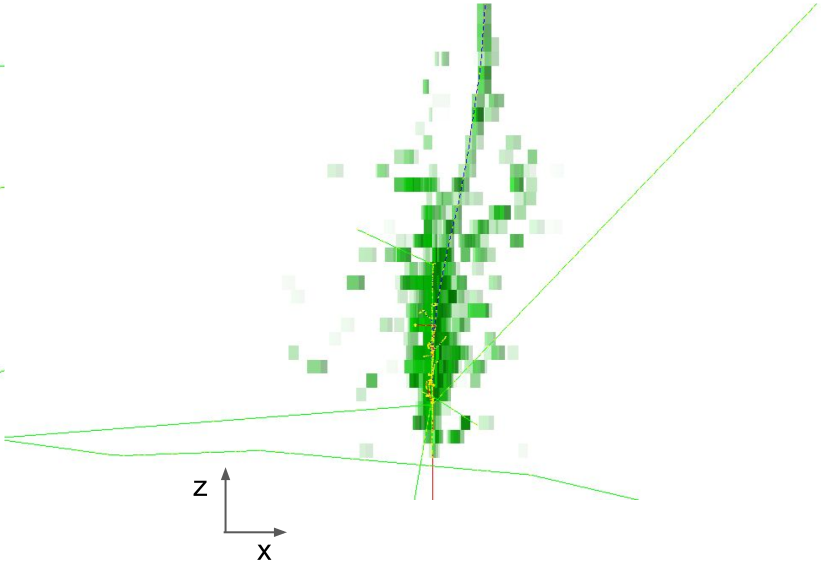
Material (s)	Silicon and tungsten	Scintillator and lead	Lead tungstate
Geometry name	**SiW**	**SciPb**	**PBWO4**
Number of layers	90	45	1
Layer thickness	1.7mm	1.7mm	200.25 mm

Calorimeter geometries (2/2)



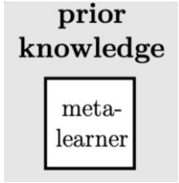
Number of readout cells is **R x P x N = 18 x 50 x 45**





Meta Learning: learn to learn “fast”

Step 2 : Adaptation



θ_0

Meta parameters

Step 1 : Meta training

Reference slide

Meta Learning : Reptile

Reptile

On First-Order Meta-Learning Algorithms

Alex Nichol and Joshua Achiam and John Schulman
OpenAI
{alex, jachiam, joschu}@openai.com

arXiv:1803.02999

Abstract

This paper considers meta-learning problems, where there is a distribution of tasks, and we would like to obtain an agent that performs well (i.e., learns quickly) when presented with a previously unseen task sampled from this distribution. We analyze a family of algorithms for learning a parameter initialization that can be fine-tuned quickly on a new task, using only first-order derivatives for the meta-learning updates. This family includes and generalizes first-order MAML, an approximation to MAML obtained by ignoring second-order derivatives. It also includes Reptile, a new algorithm that we introduce here, which works by repeatedly sampling a task, training on it, and moving the initialization towards the trained weights on that task. We expand on the results from Finn et al. showing that first-order meta-learning algorithms perform well on some well-established benchmarks for few-shot classification, and we provide theoretical analysis aimed at understanding why these algorithms work.

MAML

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Chelsea Finn, Pieter Abbeel, Sergey Levine

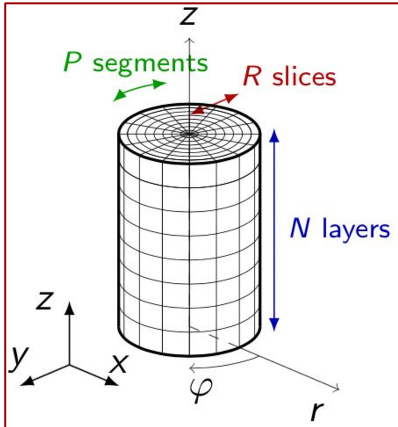
arXiv:1703.03400

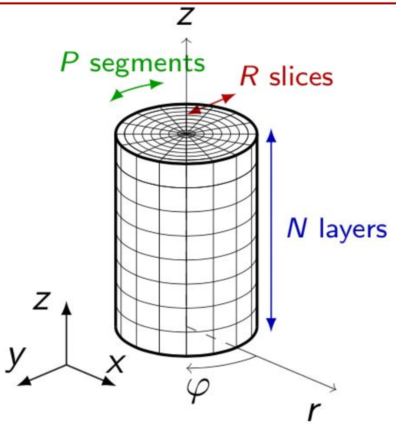
We propose an algorithm for meta-learning that is model-agnostic, in the sense that it is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. The goal of meta-learning is to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples. In our approach, the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task. In effect, our method trains the model to be easy to fine-tune. We demonstrate that this approach leads to state-of-the-art performance on two few-shot image classification benchmarks, produces good results on few-shot regression, and accelerates fine-tuning for policy gradient reinforcement learning with neural network policies.

Algorithm 1 Reptile (serial version)

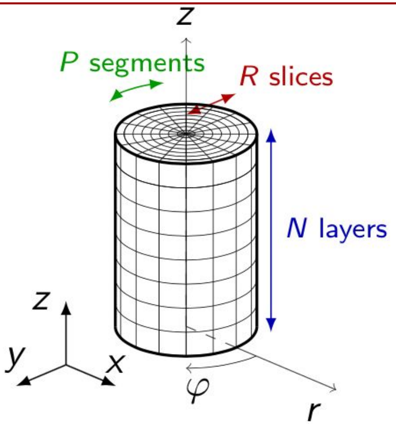
Initialize ϕ, the vector of initial parameters
for iteration $= 1, 2, \ldots$ **do**
  Sample task τ, corresponding to loss L_τ on weight vectors $\widetilde{\phi}$
  Compute $\widetilde{\phi} = U_\tau^k(\phi)$, denoting k steps of SGD or Adam
  Update $\phi \leftarrow \phi + \epsilon(\widetilde{\phi} - \phi)$
end for

Meta learning for fast shower simulation





Meta training



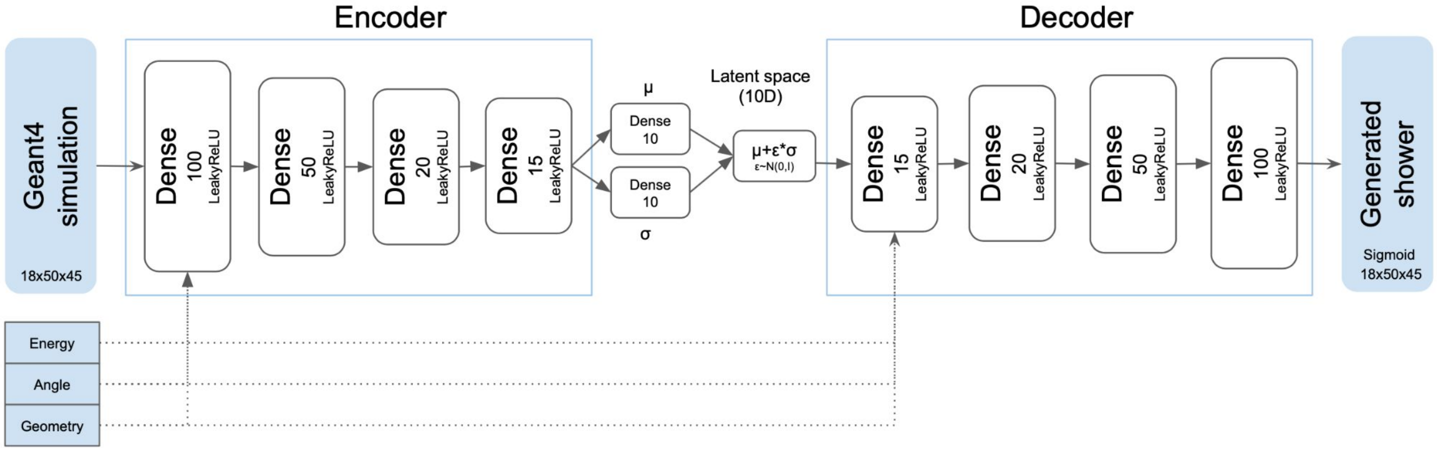
Adaptation

Material (s)	Silicon and tungsten	Scintillator and lead	Lead tungstate
Geometry name	**SiW**	**SciPb**	**PBWO4**
Number of layers	90	45	1
Layer thickness	1.7mm	1.7mm	200.25 mm

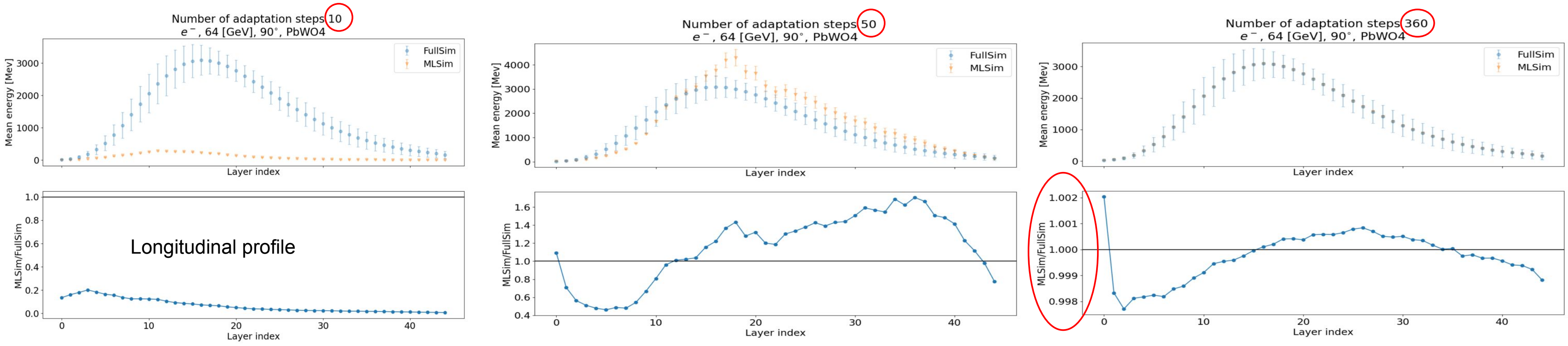
Available on Zenoodo : [High Granularity Electromagnetic Calorimeter Shower Images]

Meta-training step

- Energy range: 1GeV-1TeV (discrete values in powers of 2)
- Incident angle : 50-90º (step of 10º)
- Generative model : Variational Autoencoder (VAE)

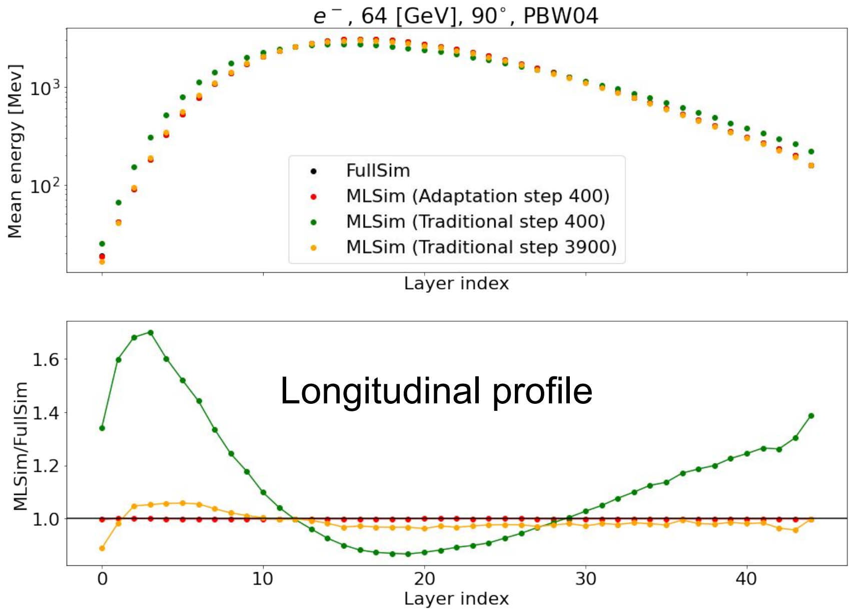


Adaptation step on a new geometry



- Meta training step: model trained on two detector geometries (SiW & SciPb)
- Fast adaptation step: the pretrained model is adapted to the PBWO4 geometry (360 steps takes 18s on a CPU machine)

Adaptation vs traditional training



Meta learning - Adaptation

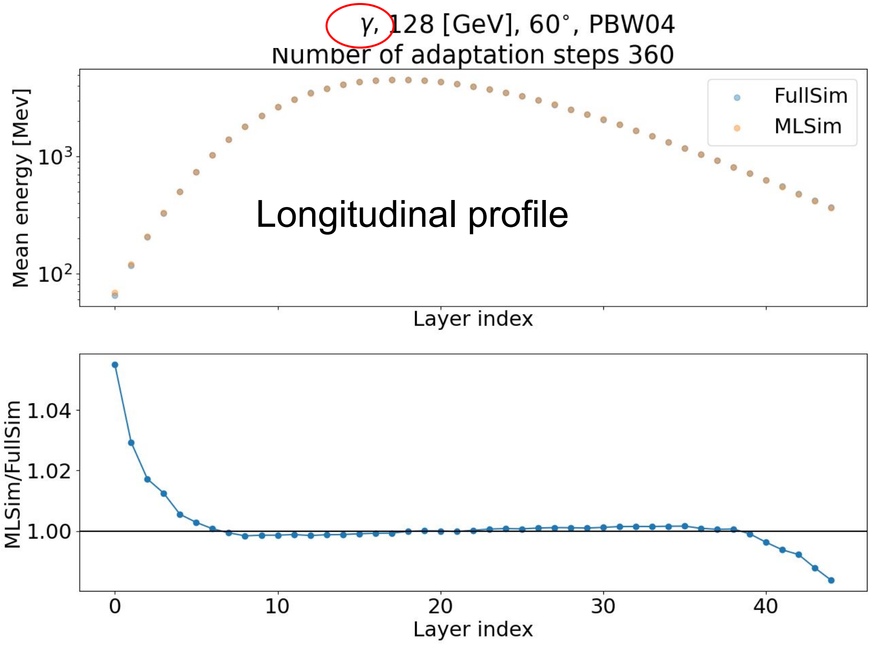
- Meta training using geometries & adaptation on a new geometry
- 400 steps of adaptation : 20.48 s

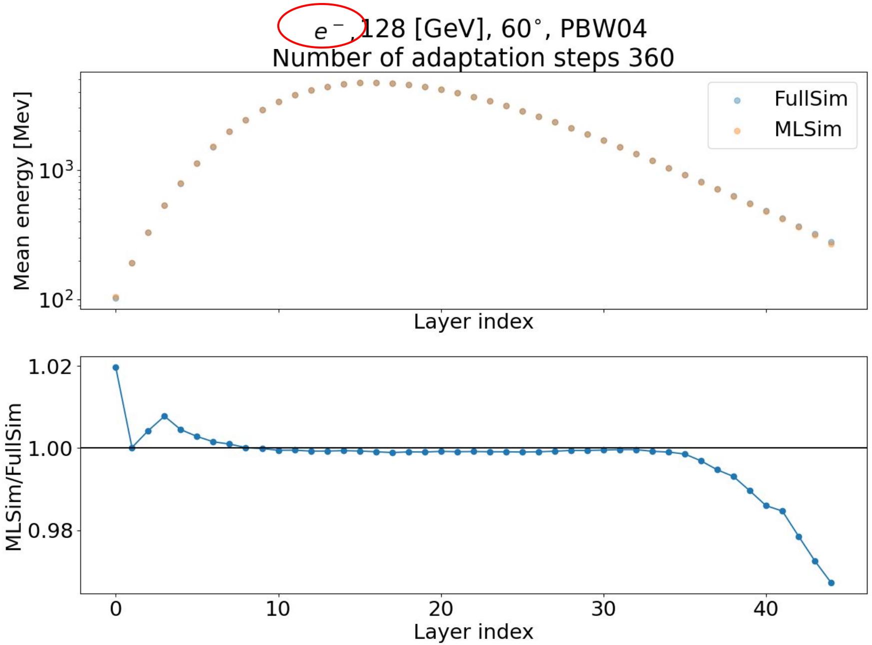
Traditional training

- Training on a single geometry with checkpoint saved every 100 epochs
- 400 steps of training : 1200 s (around 3h for 3900 steps)

$P(shower|geometry, energy, angle, type)$

Condition on the particle type

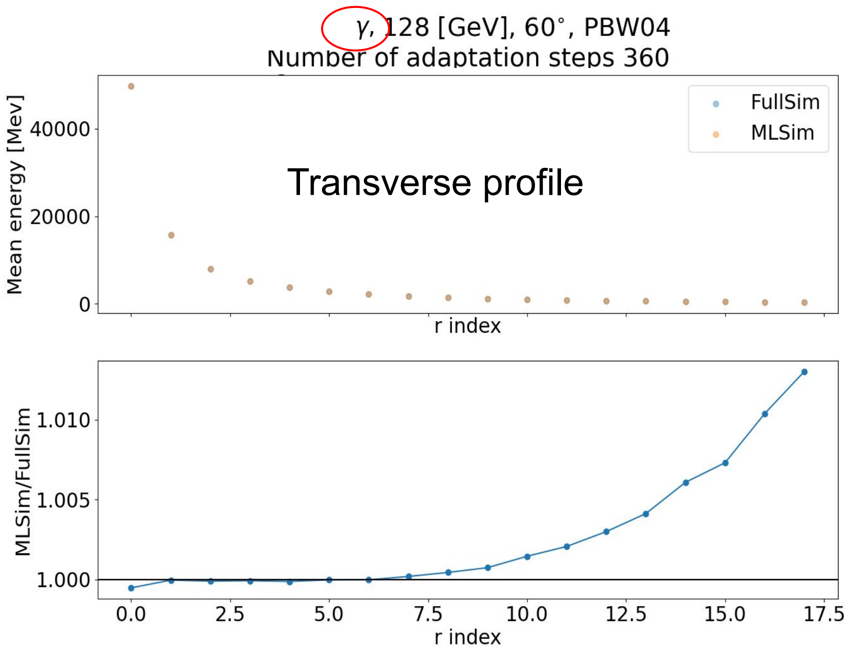


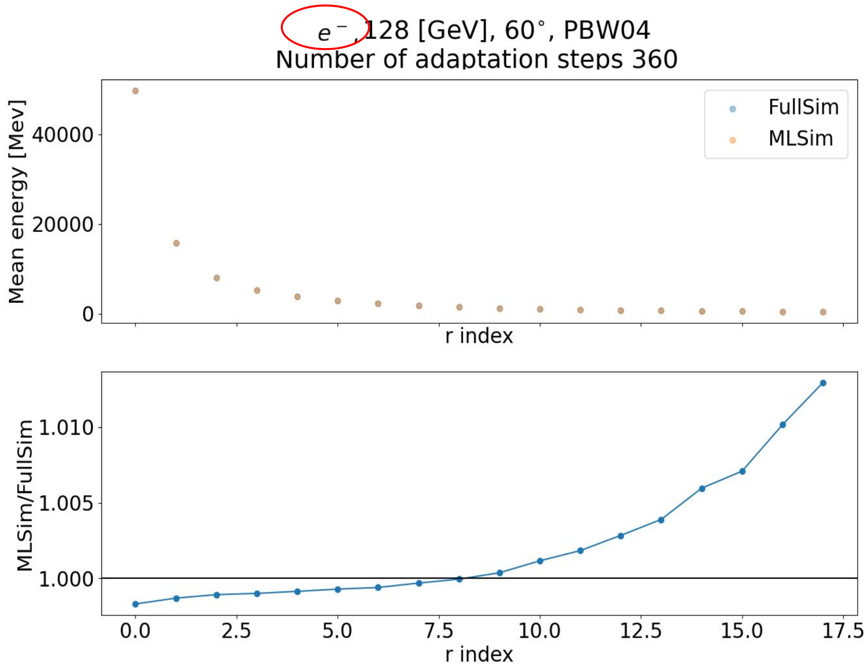


γ,

$P(shower|geometry, energy, angle, type)$

Condition on the particle type





Summary & conclusion

- Fast simulation techniques are needed to cope with the new challenges for detector simulation
- ML approaches are largely investigated in high energy physics for fast calorimeter simulation
    - Experiment specific models
    - **Generalizable simulator**
        - **Meta learning approach** for multiple detector geometry modeling
        - Ongoing work on validation for use on realistic geometries (Future Circular Colliders)
    - Very promising results and some models are now already in production!

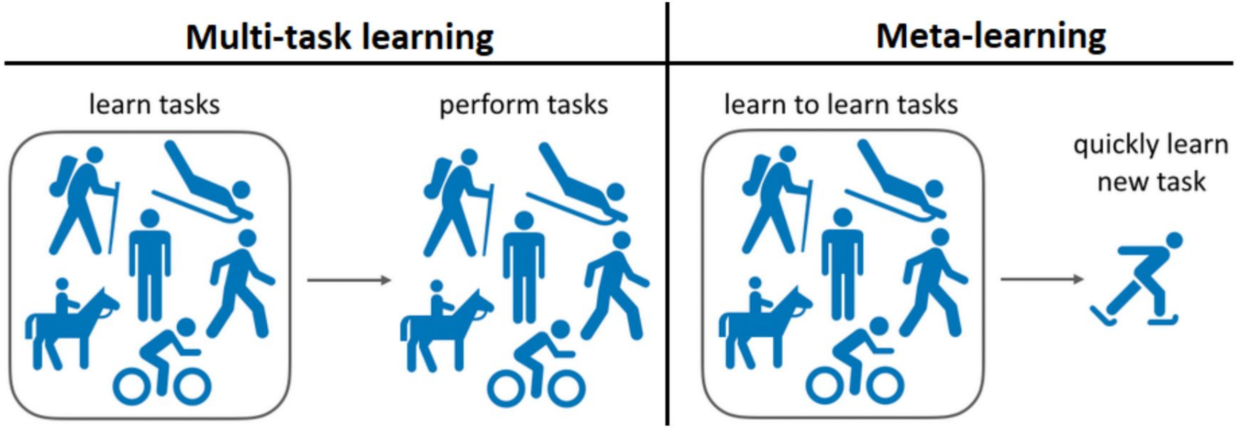
CalloChallenge : first-ever Fast Calorimeter Simulation Challenge!

- **Dataset 1** can be downloaded from Zenodo with DOI 10.5281/zenodo.6234054. It is based on the ATLAS GEANT4 open datasets that were published here. There are three files, two for photons and one for charged pions. Each dataset contains the voxelised shower information obtained from single particles produced at the calorimeter surface in the η range (0.2-0.25) and simulated in the ATLAS detector. There are 15 incident energies from 256 MeV up to 4 TeV produced in powers of two. 10k events are available in each sample with the exception of those at higher energies that have a lower statistics. These samples were used to train the corresponding two GANs presented in the AtlFast3 paper SIMU-2018-04 and in the FastCaloGAN note ATL-SOFT-PUB-2020-006. The number of radial and angular bins varies from layer to layer and is also different for photons and pions, resulting in 368 voxels for photons and 533 for pions.
- **Dataset 2** can be downloaded from Zenodo with DOI 10.5281/zenodo.6366270. It consists of two files with 100k GEANT4-simulated showers of electrons each with energies sampled from a log-uniform distribution ranging from 1 GeV to 1 TeV. The detector has a concentric cylinder geometry with 45 layers, where each layer consists of active (silicon) and passive (tungesten) material. Each layer has 144 readout cells, 9 in radial and 16 in angular direction, yielding a total of 9x16x45 = 6480 voxels. One of file should be used for training the generative model, the other one serves as reference file in evaluation.
- **Dataset 3** can be downloaded from Zenodo with DOI 10.5281/zenodo.6366323. It consists of 4 files, each one contains 50k GEANT4-simulated eletron showers with energies sampled from a log-uniform distribution ranging from 1 GeV to 1 TeV. The detector geometry is similar to dataset 2, but has a much higher granularity. Each of the 45 layer has now 18 radial and 50 angular bins, totalling 18x50x45=40500 voxels. This dataset was produced using the Par04 Geant4 example. Two of the files should be used for training the generative model, the other two serve as reference files in evaluation.

https://calochallenge.github.io/homepage/

Backup

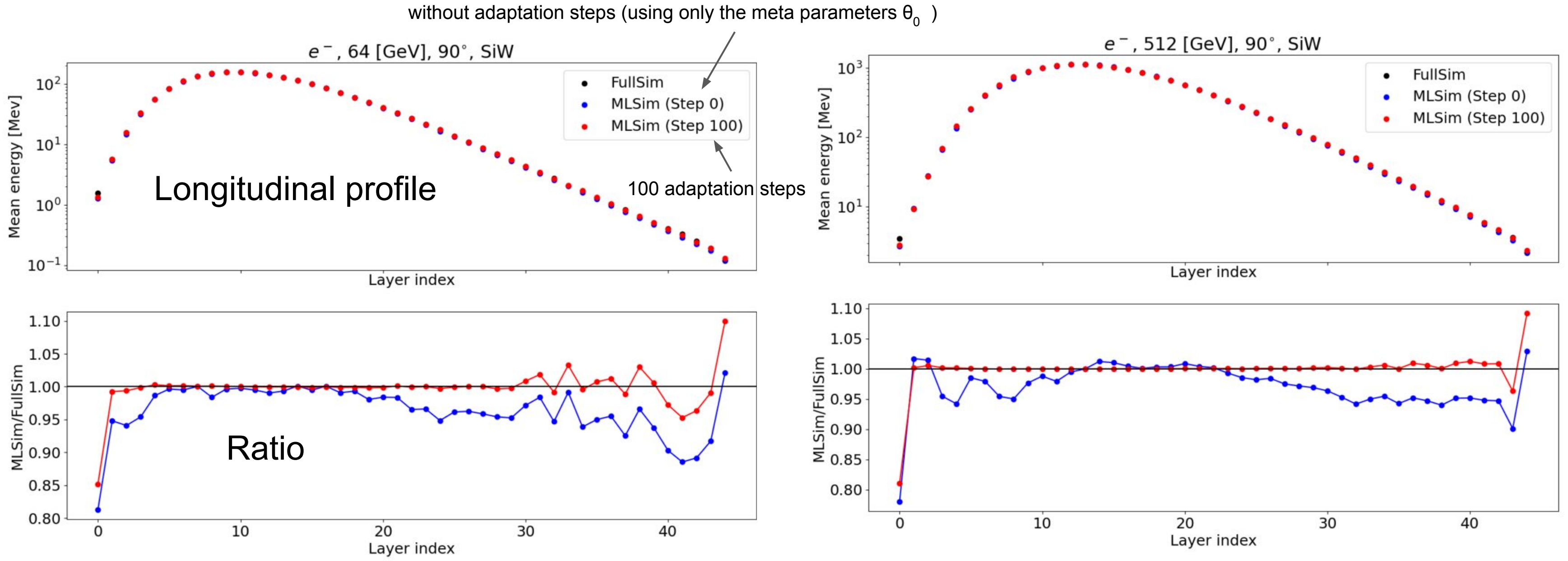
Multi-task learning vs Meta-learning



Jointly trained to perform well on multiple **fixed** tasks

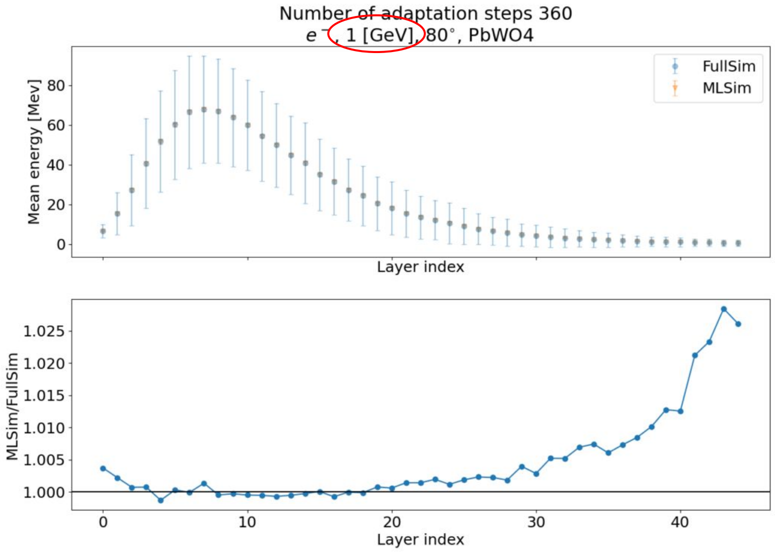
Leverage prior learning experience in order to learn **new** tasks more quickly

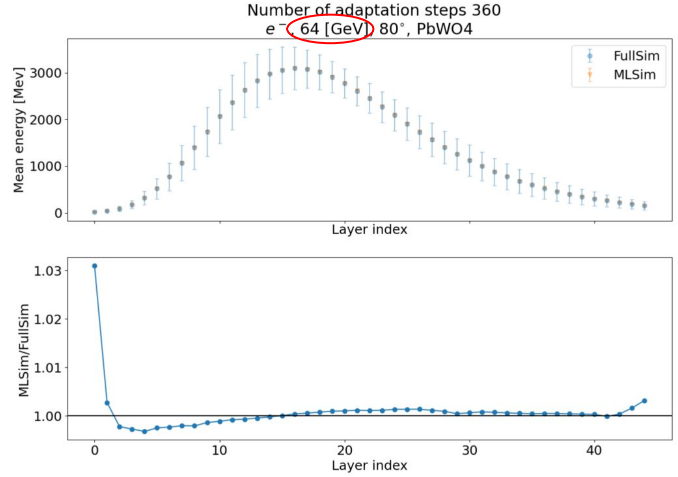
Validation on a meta-training geometry

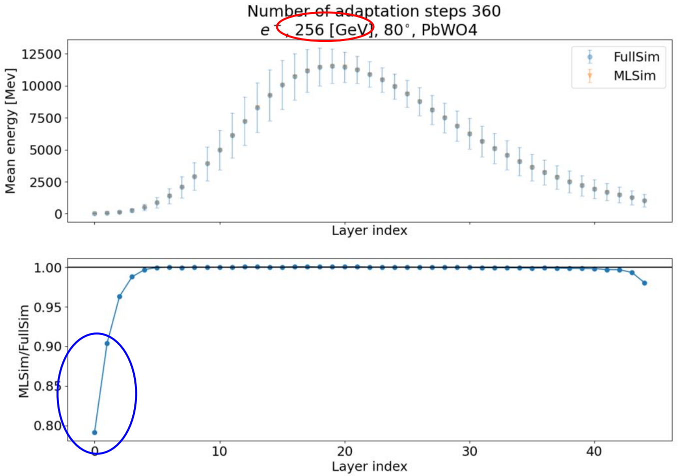


Meta training step: model trained on two detector geometries (SiW & SciPb)

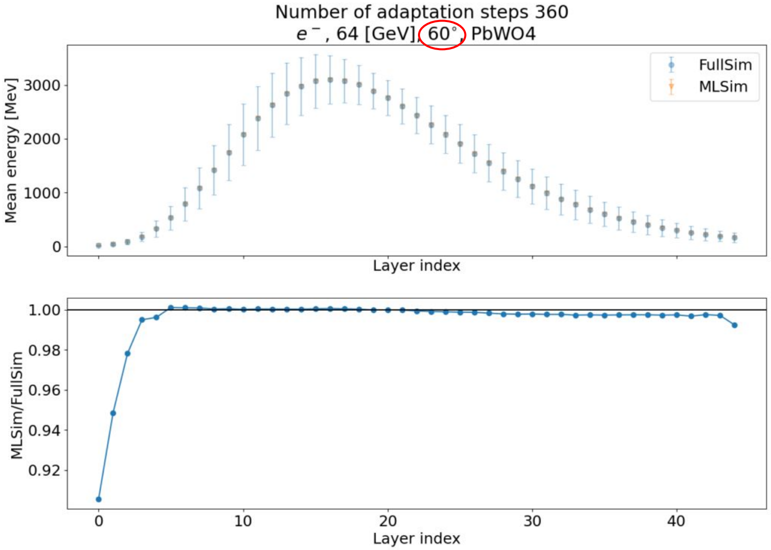
Adaptation across energies

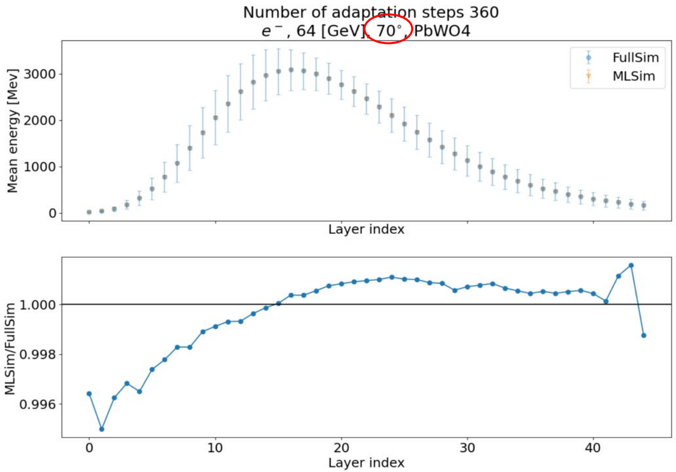


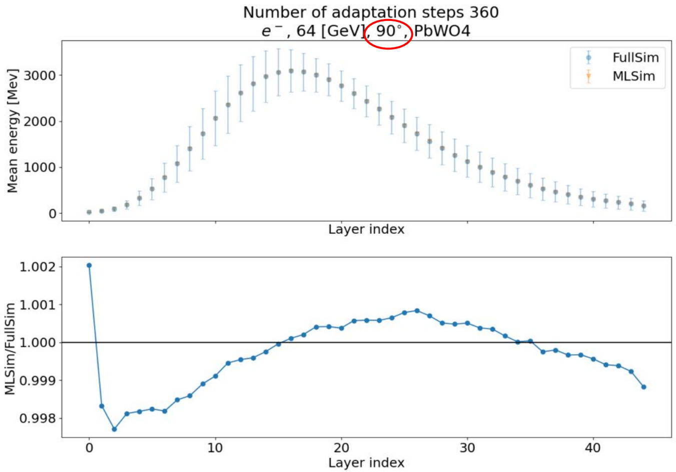




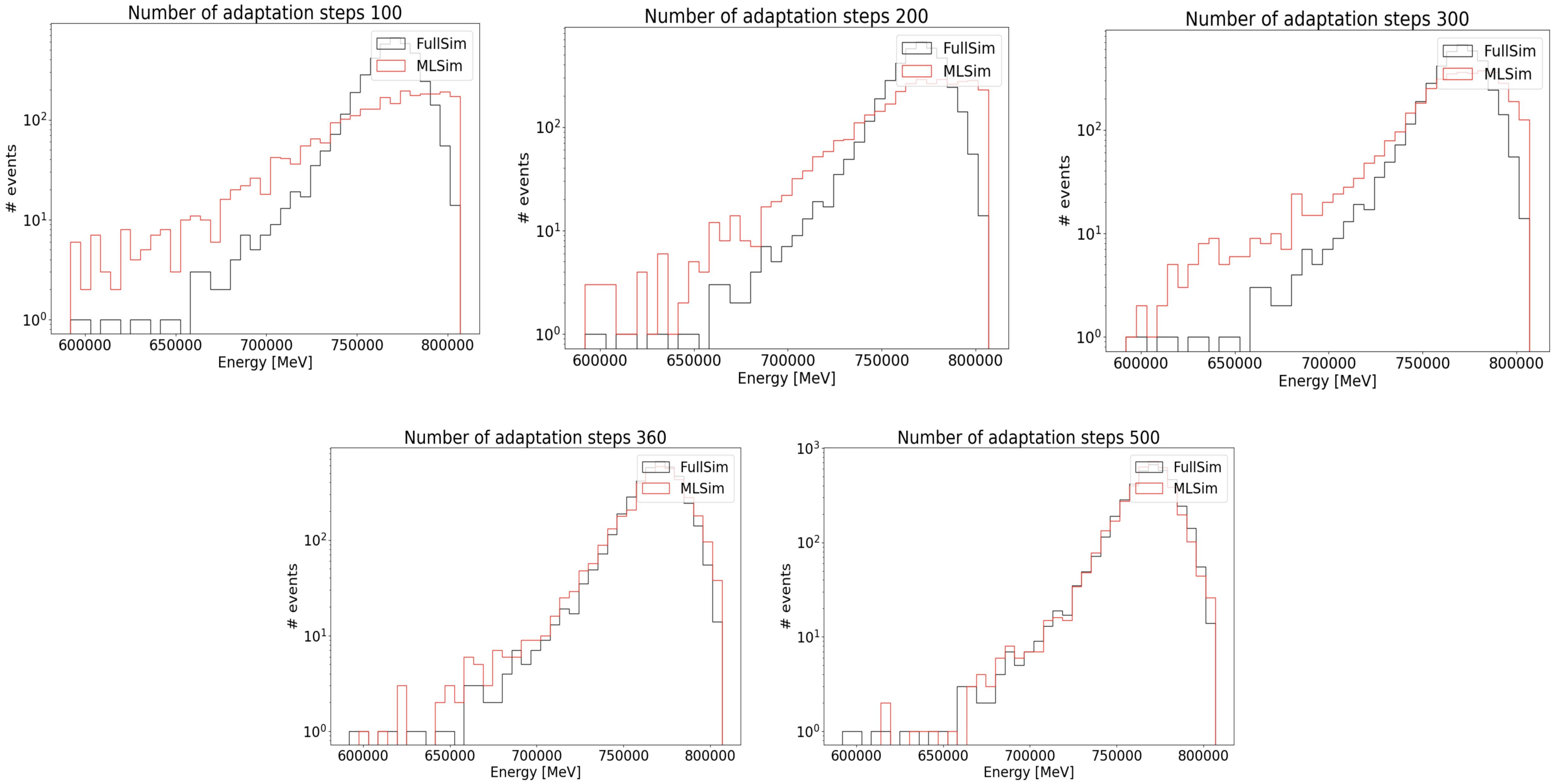
Adaptation across angles



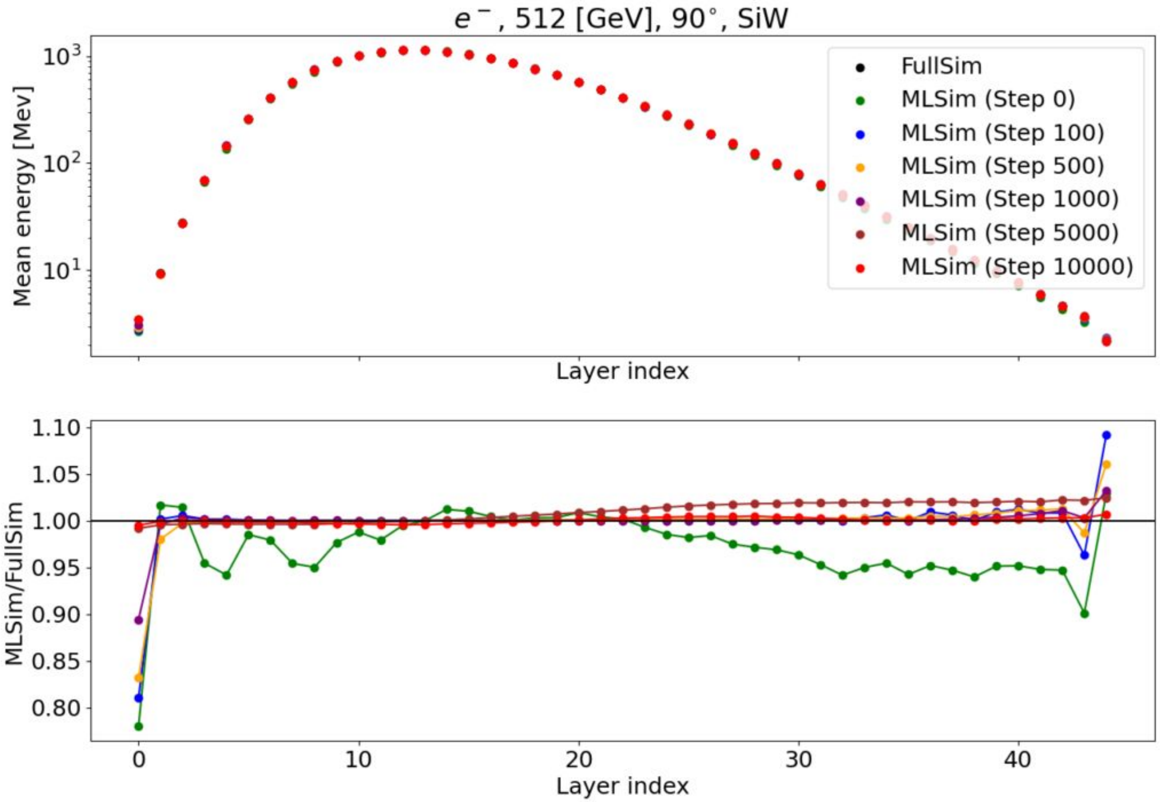




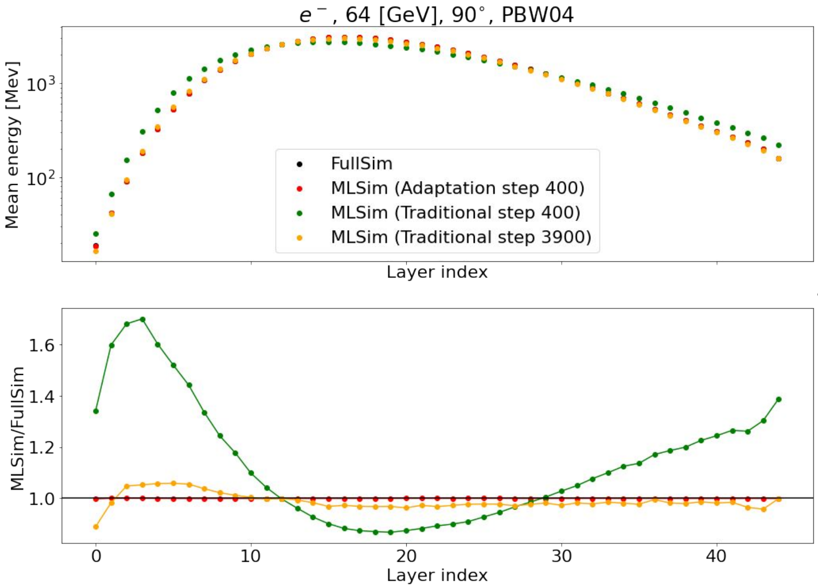
Total energy distribution, e^-, 1 TeV, 90^o



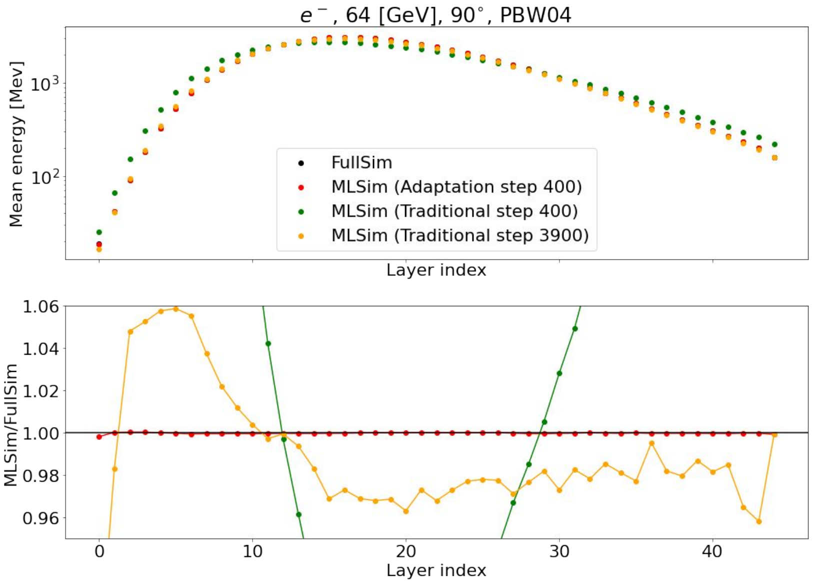
Improvement with longer steps



Adaptation vs traditional training



Zoom in the ratio plot



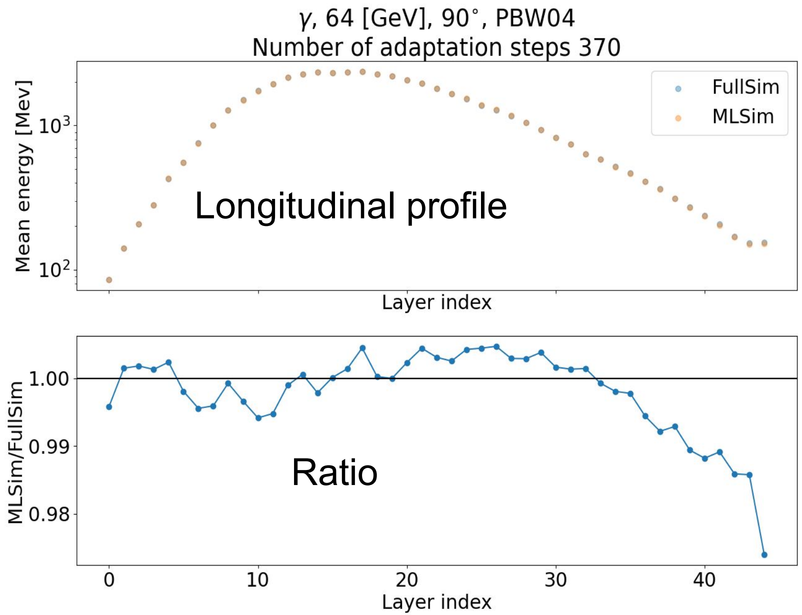
Meta learning - Adaptation

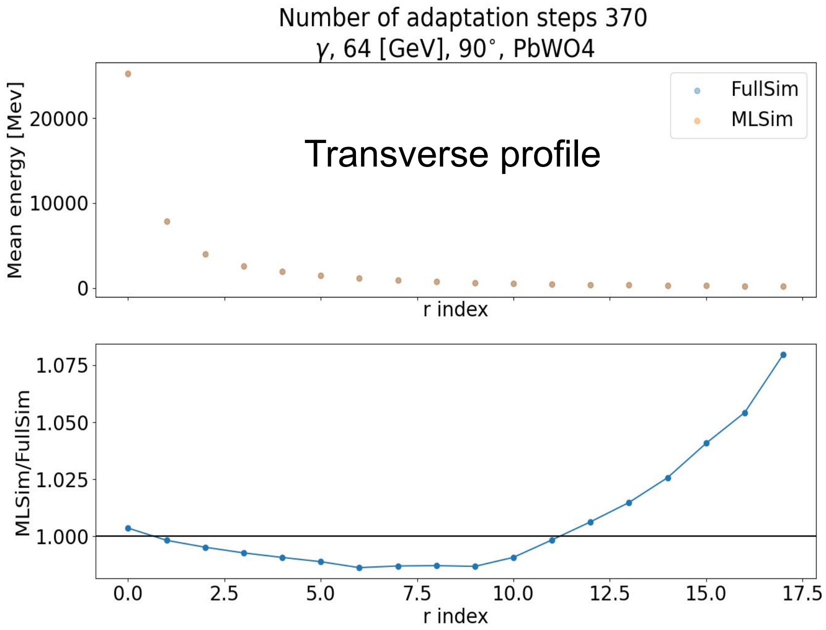
- Meta training using geometries & adaptation on a new geometry
- 400 steps of adaptation : 20.48 s

Traditional training

- Training on a single geometry with checkpoint saved every 100 epochs
- 400 steps of training : 1200 s

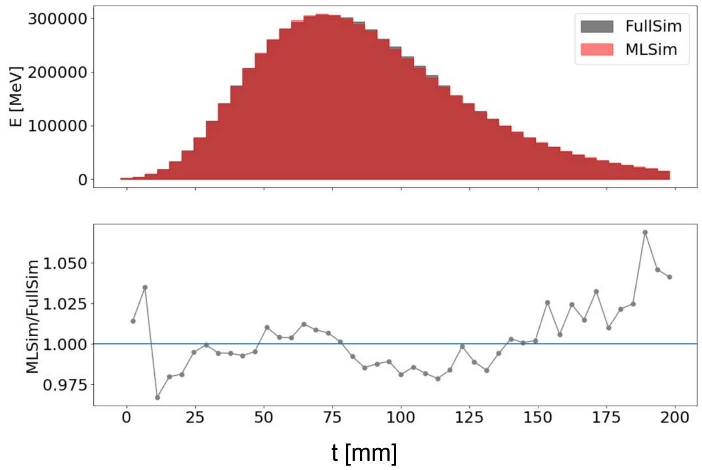
Adaptation step on a new geometry : new particle type



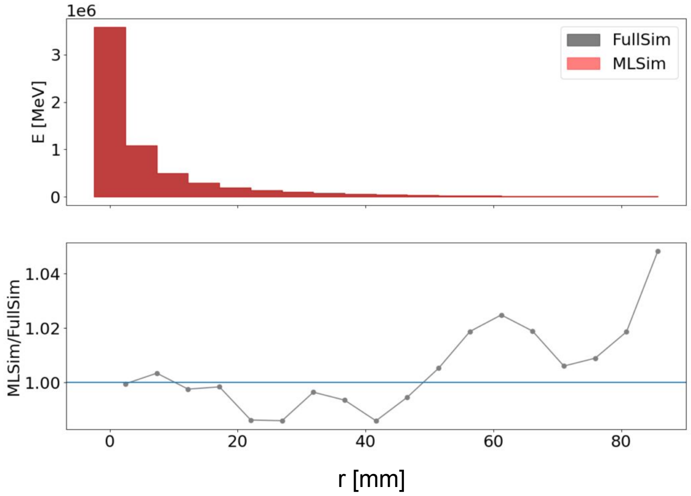


- Meta training step: model trained on two detector geometries (SiW & SciPb)
- Fast adaptation step: the pretrained model is adapted to the PBWO4 geometry

Inference in G4 after fast adaptation to a new geometry



Longitudinal profile



Transverse profile

Par04 example

- Fast simulation with ML within Geant4
- New Par04 extended example in Geant4 11.0

  [***examples/extended/parameterisation/Par04***]()

- Demonstrates how to incorporate inference libraries
  - ONNX Runtime
  - LWTNN
- The ML trained on 2 provided geometries, conditioned on the **energy** and **angle** of the particle
- Example can run full and fast simulation (if any of the inference libraries is available, e.g. via LCG)

Name
..
Par04ActionInitialisation.cc
Par04DefineMeshModel.cc
Par04DetectorConstruction.cc
Par04DetectorMessenger.cc
Par04EventAction.cc
Par04EventInformation.cc
Par04Hit.cc
Par04InferenceMessenger.cc
Par04InferenceSetup.cc
Par04LwtnnInference.cc
Par04MLFastSimModel.cc
Par04OnnxInference.cc
Par04PrimaryGeneratorAction.cc
Par04RunAction.cc
Par04SensitiveDetector.cc

Simulation time