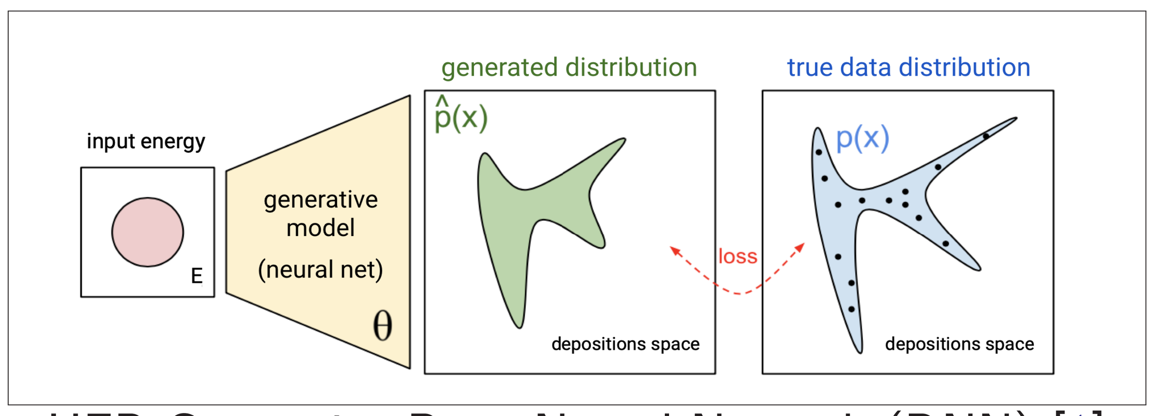
End–To–End Deep Learning Fast Simulation Framework

Ioana Ifrim • Witold Pokorski • Anna Zaborowska

EP-SFT, CERN, Geneva, Switzerland; ioana.ifrim@cern.ch

Deep Learning aided FastSim



HEP Generative Deep Neural Network (DNN) [1]

Inference Module Integration

– The inference module while integrated within the (full and fast) simulation toolkit (Geant4 [2]) is agnostic to generative model architecture.

General Validation Benchmark

– Generalisation of validation tools allows different DL models to be assessed against a common benchmark procedure (also used for parametrisation).

Streamlined DNN Fast Simulation Workflow

In the context of Fast Simulation (FastSim) we present the ongoing work on the implementation of an end–to–end framework which integrates Deep Learning (DL) simulation methods with an existing simulation toolkit (Geant4 [2]).

Step 1: Data Production
– Creating matrices of energy deposits using Geant4.

Step 2: DNN Training
– Generative models HEP customised and trained.

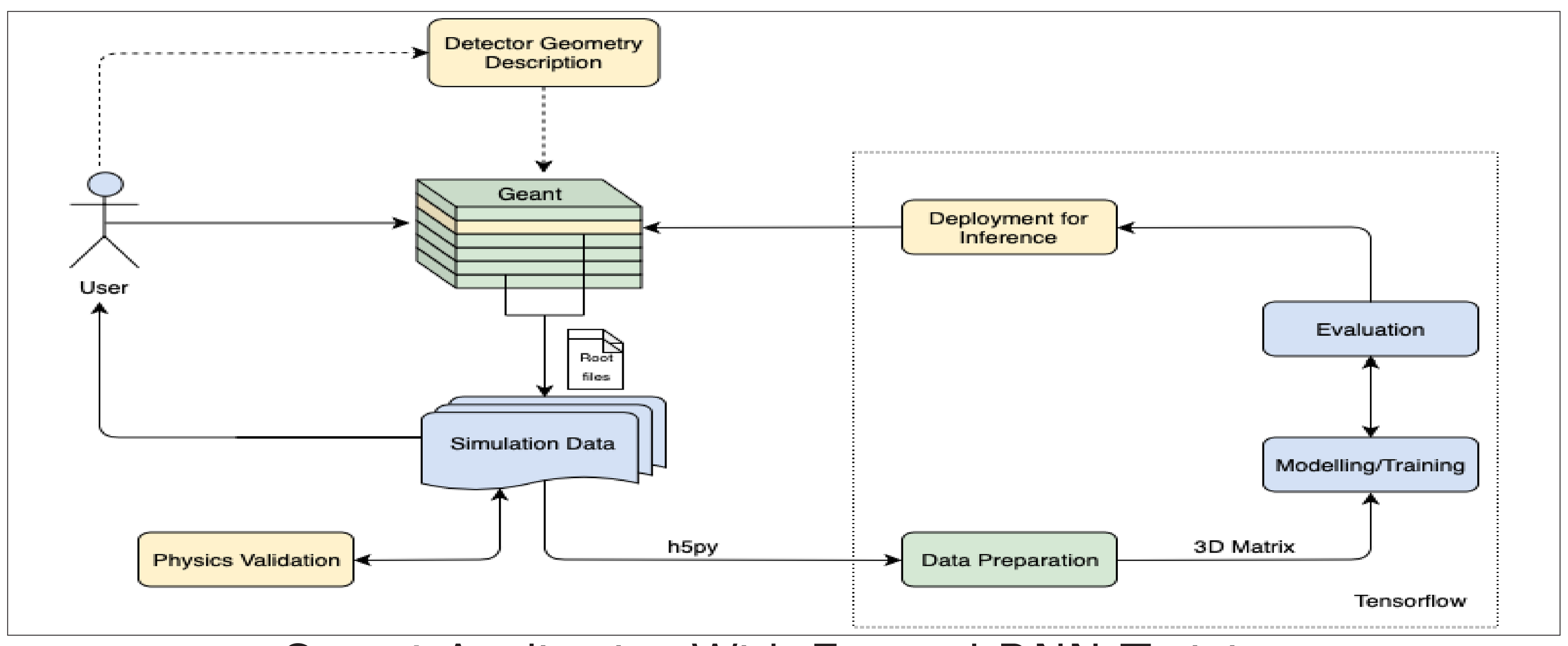
Step 3: Physics Validation
– Sequence chain of HEP performance measurements.

Step 4: DNN Inference
– Geant4 hooks for FastSim with DNN dependencies.

Steps 1, 3 and 4 are within the Geant4 application, while step 2 is performed independently in custom designed tools.

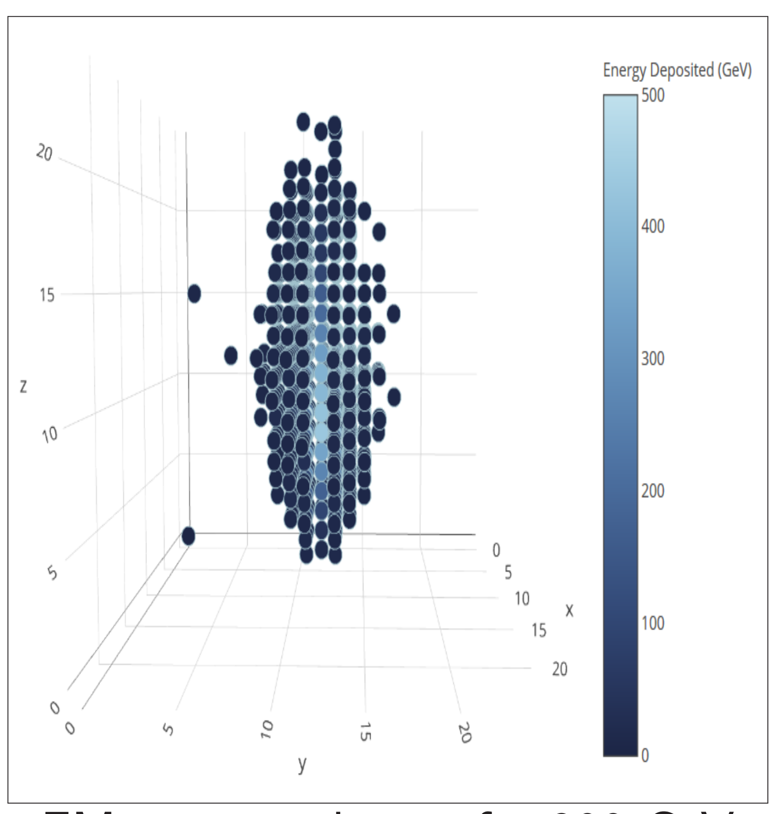
General Components Integration Overview

The overall goal is to facilitate the usage of generative DNNs by integrating the inference module with Geant4 and offering a general validation benchmark environment. An example Geant4 application can be employed to produce simulation data to be used for DNN training.



Geant4 Application With External DNN Training

Data Production



EM average shower for 300 GeV electrons

Dataset production of calorimeter showers (Geant4) in a simple but configurable detector setup:

– adjustable granularity and size of detector
– $PbWO_4$,Pb/LAr,W/Si,...
– data is stored as a matrix of energy deposits
– initial particle properties stored as labels for training

DNN Training

Customised generative models are trained with the produced datasets resulting in a serialised graph used later for inference:

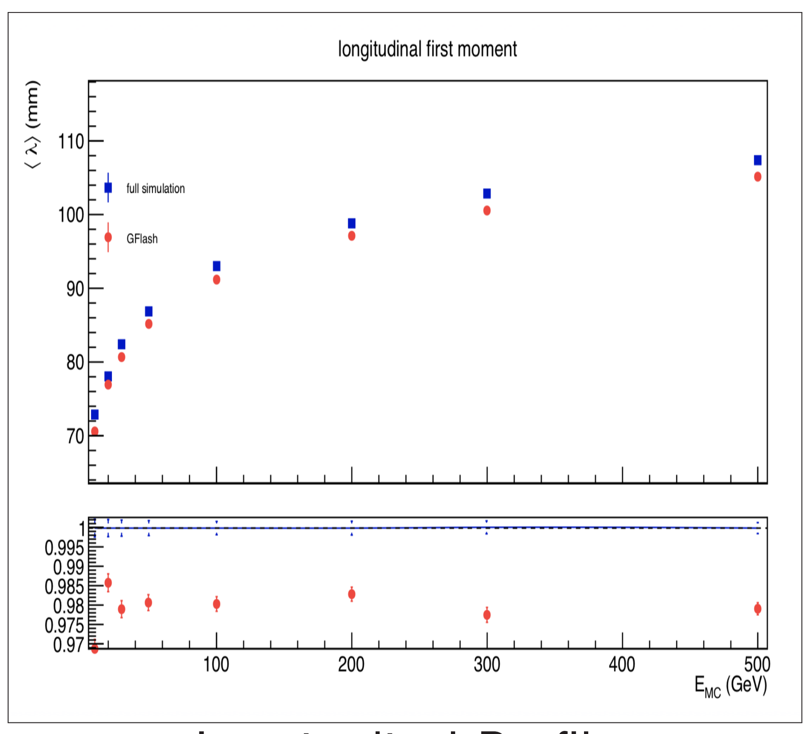
– values of the variables and the graph to be kept in a single file
– training only operations to be removed (checkpoint saving)
– parts of the graph that are never reached to be stripped out
– debug operations like CheckNumerics to be removed

Physics Validation

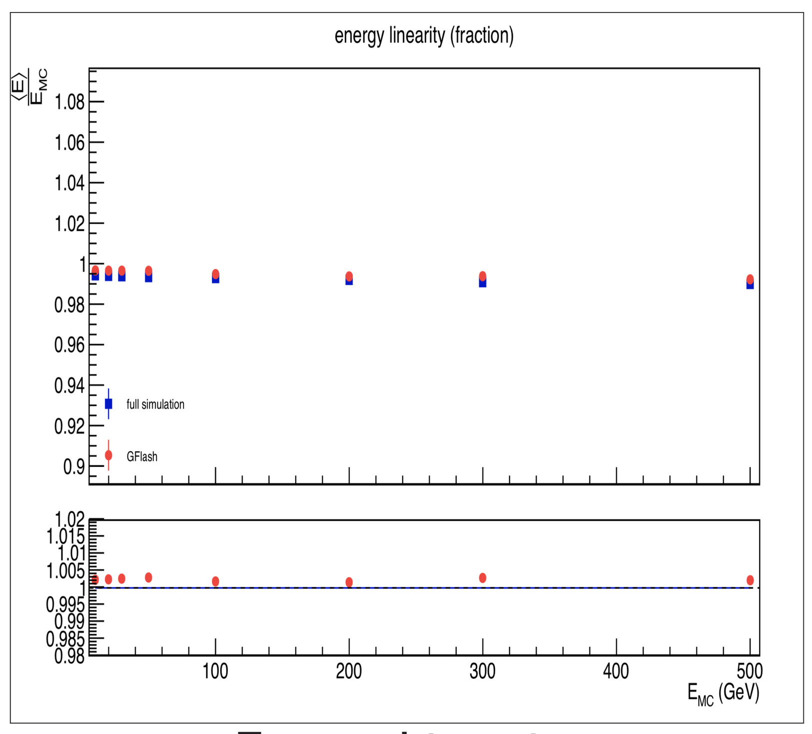
A standardised set of validation procedures were developed, amongst which:

– total deposited energy
– energy distribution layer-wise
– longitudinal and transverse profiles (and first/second moments)

The depicted validation is used in the context of GFlash [3] parametrisation to underline the general use case of the benchmarks which are applicable for any DNN Fast Simulation generative model.



Longitudinal Profile



Energy Linearity

DNN Inference

Internal hooks of Geant4 (`G4VFastSimulationModel`) are used to call the (under development) inference module (with an external dependency on the TensorFlow C++ API [4] at the moment).

We use the restored model for prediction. Where the input node is fed the incoming particle's parameters (energy, direction, ...) and the end result is a matrix of energy depositions.

References

[1] OpenAI - Generative Models, June 2016. Retrieved from Generative Models

[2] Agostinelli, Sea, et al. "GEANT4-a simulation toolkit." Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506.3 (2003): 250-303

[3] Grindhammer, Guenter, and S. Peters. "The parameterized simulation of electromagnetic showers in homogeneous and sampling calorimeters." arXiv preprint hep-ex/0001020 (2000)

[4] TensorFlow C++ Reference Retrieved from TensorFlow C++