AIDA++ Fast Simulation

November 22, 2019

Anna Zaborowska

Proposed deliverables (by Graeme & Frank)

CLASSICAL FASTSIM Deliverable 1: 24 PM (8 from EU)	● Deliver generic fast calorimeter plugin, with generic reusable tool for experiments.
ML-AIDED FASTSIM Deliverable 2 24 PM (8 from EU)	● Implement and test different DNN architectures for at least 2 different calorimeter detector types. ● Draw conclusions on the general suitability of architectures to support different calorimeter models.
ML-AIDED FASTSIM Deliverable 3 12 PM (4 from EU)	● Support calo-ML inference within the Geant4 toolkit, for at least one type of inference model

Classical fast simulation for calorimetry

Approach

- Parametrisation of EM showers
- Currently applied mostly to forward calorimeters (acceptable bargain between sim time and accuracy)
- Used in ATLAS, CMS

Issues

- Highly dependent on the detector (and conditions)
- Embedded in experiments' frameworks
- Problemating and costly re-tuning of parameters

Goal

Generic tools:

- Easy (quick) to implement for any calorimeter
- Flexible re-tuning procedures
- Available from within Geant4 to mix with full simulation

DELIVERABLE 1

Classical fast simulation for calorimetry

Currently:

- Geant4 offers hooks for fast simulation that allow to **seamlessly mix** fast and full simulation.
- Implementation of 'how' to parametrise (models) up to the user.
- One existing model (**GFlash**) of EM shower parameterization offers poor accuracy with hardcoded values of energy/material independent parameters (from arXiv:hep-ex/0001020)

Tasks:

- Develop tools for re-tuning of parameters based on the full simulation and users' geometry. IMPROVED ACCURACY EASIER / QUICKER TO USE
- Benefit from existing & used in production approaches (e.g. CMS' which originated from GFlash). TESTED & VALIDATED MODELS
- Validate tools on different geometries. GENERALITY
- Implement into Geant4 ACCESSIBLE / READY-TO-USE

Deliverable 2

ML-aided fast simulation for calorimetry

Approach

- Showers generated by DNNs
- GANs, VAEs, autoregressive, ...
- Potential for higher accuracy and speed-up wrt classical approaches
- Prototyped in many experiments

Issues

- Highly dependent on the detector (trained, possibly architecture-wise reusable)
- Still in prototyping stage
- Costly NN training, if started from scratch many lessons to be (re)learnt

Goal

- Test different DNN architectures on different types of calorimeters
- Draw conclusions on generalisation of architectures and application to types of calorimeters

Deliverable 2

ML-aided fast simulation for calorimetry

Currently:

- Many prototypes existing in many experiments (with different calorimeter types).
- Different architectures tested: GANs, VAEs, autoregressive models.

Tasks:

- Implement and test different DNN architectures on at least 2 different calorimeter types. GENERALITY
- Learn from existing approaches (both the ones that did work & those that failed).
  TESTED & VALIDATED MODELS COMMON FEATURES
- Formulate conclusions: are there commonalities? Similar problems? Differences? Hints on when (with which detector) to use which DNNs?
  TIPS WHAT TO USE / EXAMPLES ?

Deliverable 3

Common tools for ML-aided fast simulation

Approach

- DNN training and validation done outside of experiments' framework (and usually in Python)
- Inference needs to be integrated in the framework (C++) and used as any other simulation

Issues

- Most ML tools Python-based
- Integration within C++ frameworks of experiments a potentially reinventing-the-wheel task

Goal

- Integrate inference within C++ framework: Geant4 (to allow to easy mix full and fast = ML simulation)

DELIVERABLE 3

Common tools for ML-aided fast simulation

Currently:

- DNN training done in Python-based ML toolkits (Keras, TensorFlow, ...).
- Models can be saved and stored in e.g. HDF5 files (already supported by Geant4).
- Inference from experiment framework requires C++ library:
  - TensorFlow - heavy dependency, not easy to build, but C libraries available in LCG releases
  - Light inference libraries (lwtnn, frugally-deep) - can limit architectures (e.g. lack of Conv3D layer)

Tasks:

- Investigate existing C++ inference libraries.
- Integrate inference tools within Geant4 toolkit to allow to use fast simulation

EASY / QUICK TO USE

- Provide examples based on DNN architectures explored in Deliverable 2

TESTED & VALIDATED MODELS

CERN

- ❏ No problem with matching effort: 2-3 P already working on the topic.
- ❏ Deliverables 1, 3 - Geant4 related, work in that direction has started.
- ❏ Deliverable 2 on DNN architectures: study began on application of autoregressive models on generic calorimeters.
- ❏ Infrastructure for ML training (GPUs) available.

DESY

- No problem with matching effort: 2-3 P already working on the topic.
- Would focus on D 2.2 implementation and testing of different DNN architectures for Calorimeter simulation
- Infrastructure for ML training (GPUs) available.

LAL

- No problem with matching effort: 1 P (permanent) already working on the topic. 1 more permanent will start. 1 PhD student will be replaced.
- Focus on D 2.2 implementation and testing of different DNN architectures for Calorimeter simulation
- Infrastructure for ML training (GPUs) available. (Jean Zay new AI-HPC machine at Idris)

Manchester

- No problem with matching effort: 1-2 P already working on the topic.
- Would focus on D 1 and can contribute both to development of tuning tools and to implementation in Geant4
- Have access to in-house support from Geant4 developer