3D convolutional GAN for fast simulation

HSF Fast Simulation topical meeting

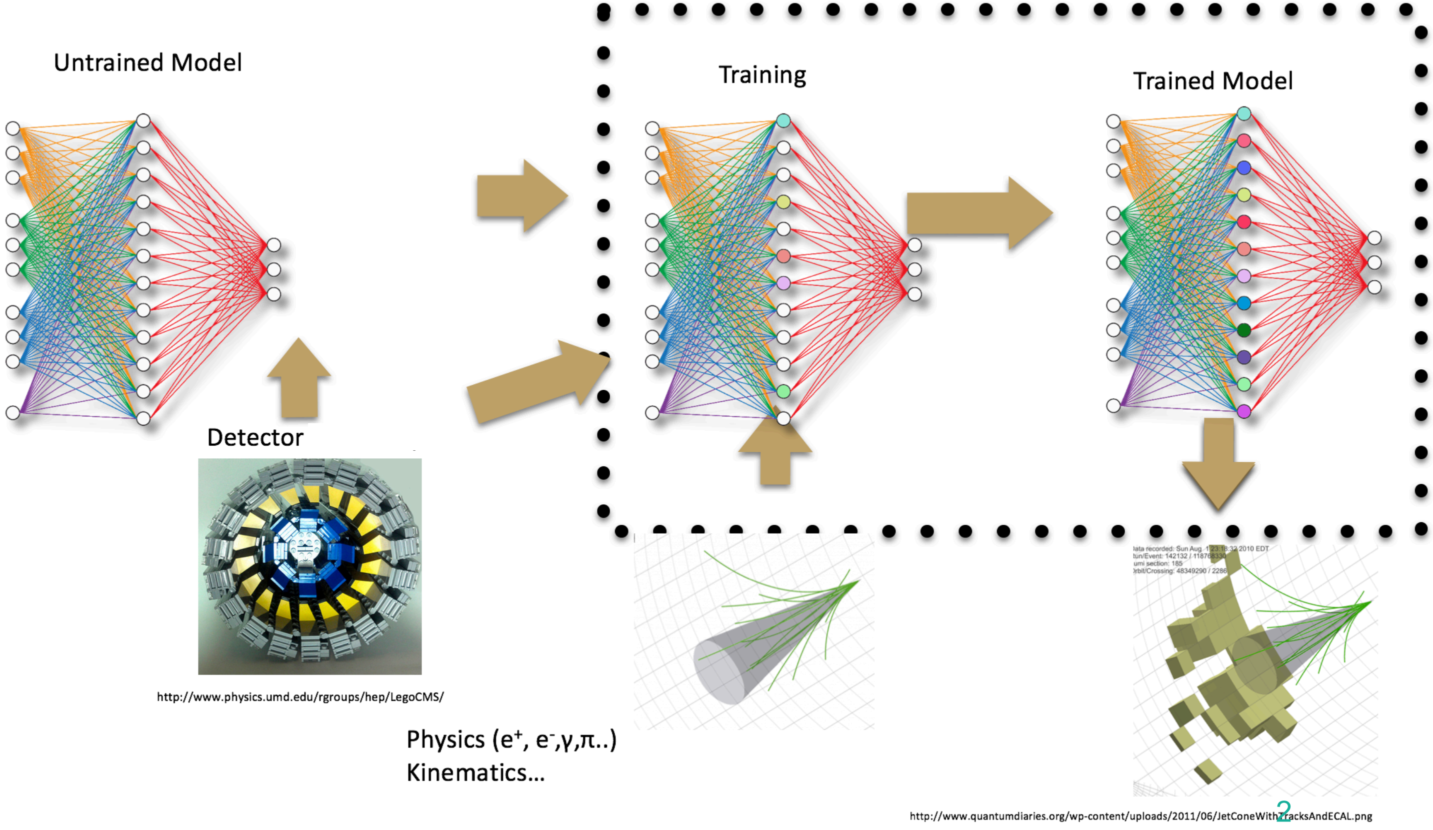
F. Carminati, G. Khattak, S. Vallecorsa

March 2019

A DL engine for fast simulation

Provide a tool that can be configured and trained for different detectors

- Start with time consuming detectors
  - Next generation highly granular calorimeters
- Train on Monte Carlo data
- Optimise training time



CERN openlab

Our plan

- Are generative models accurate enough?
- Can we sustain the increase in detector complexity?

→

- A first proof of concept
- Understand performance and validate accuracy

- How generic is this approach?
- Can we “adjust” architecture to fit a larger class of detectors?
- What resources are needed?

→

- Prove generalisation is possible
- Understand and optimise computing resources

- Can it be integrated in a “standard” simulation workflow?

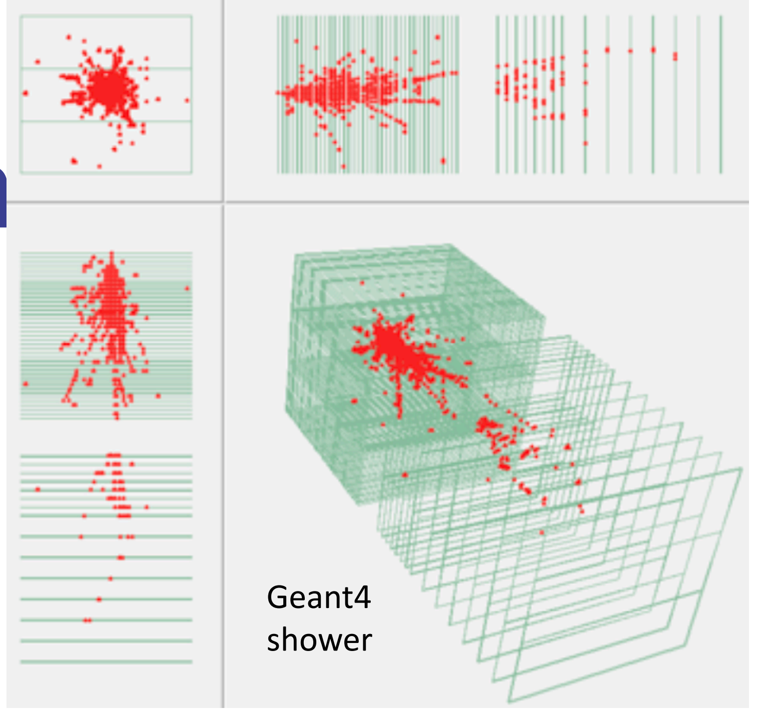
→

- Pre-processing
- Interfaces to simulation engine

Collaboration with CERN SFT group

Proof of concept, benchmarking and validation

CLIC calorimeter simulation



Data is essentially a 3D image

Electromagnetic calorimeter detector design(*)

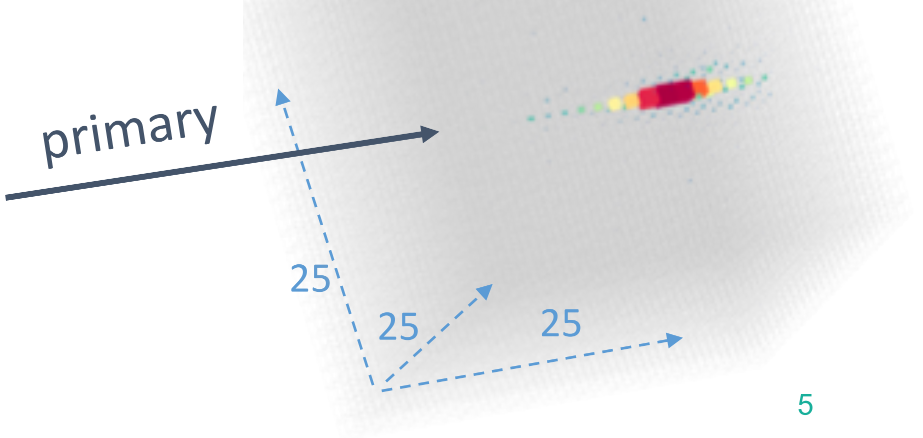
(Linear Collider Detector studies)

- 1.5 m inner radius, 5 mm×5 mm segmentation: 25 tungsten absorber layers + silicon sensors

1M single particle samples (e,γ,π) with flat energy spectrum (10-500) GeV

1. Orthogonal to detector surface (25x25x25 pixels)
2. +/- 30° random incident angle (51x51x25 pixels)

Highly segmented
Sparse.



(*) http://cds.cern.ch/record/2254048#

3D convolutional GAN

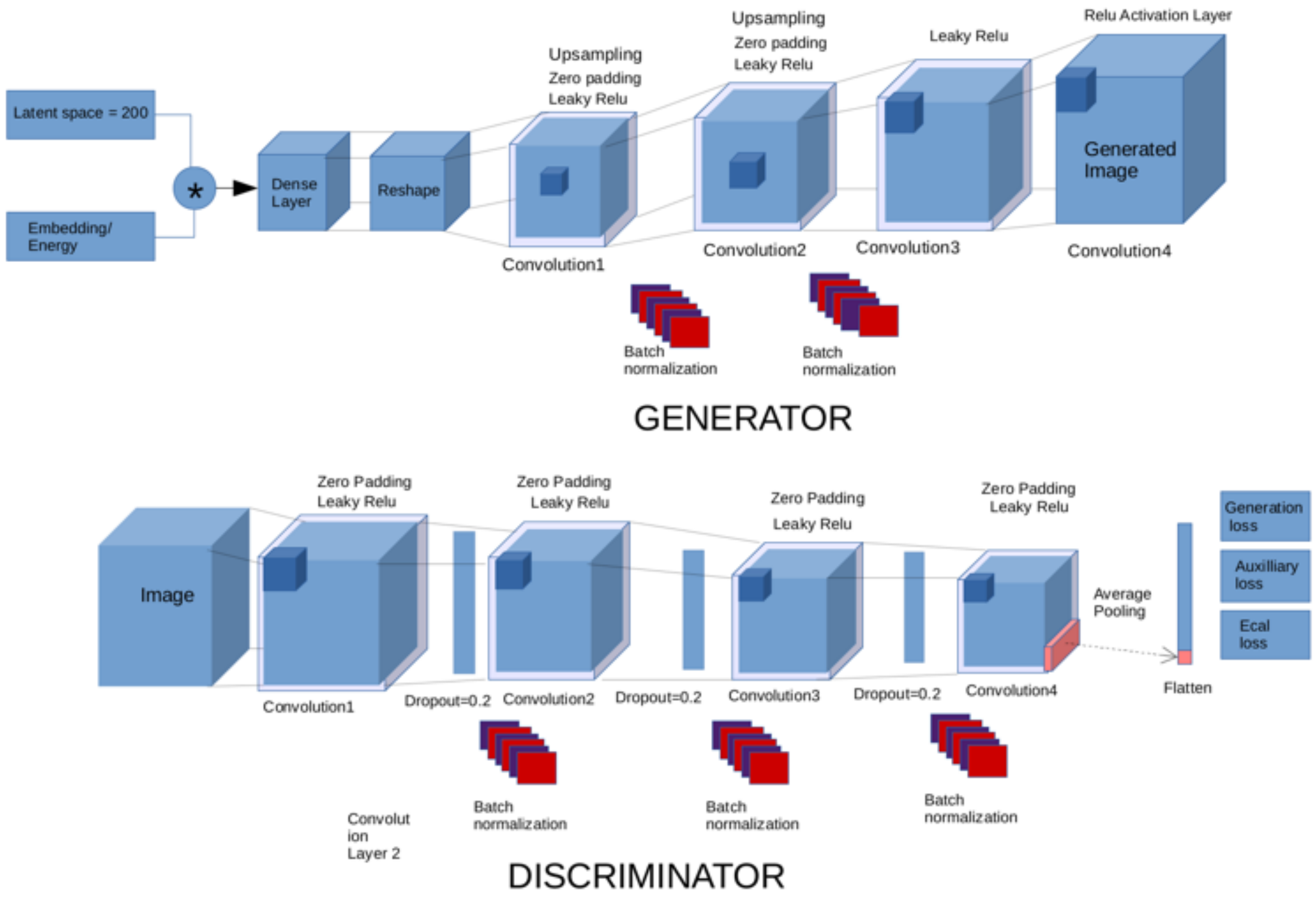
Similar discriminator and generator models

- **3D convolutions** (keep X,Y symmetry)
- ~1M parameters total

Condition training on input variables (energy, angle)

Auxiliary regression tasks assigned to the discriminator: cross check

- **Custom losses**



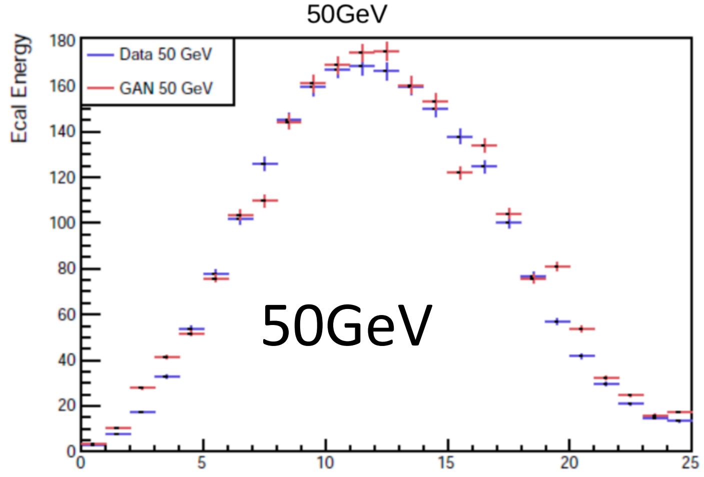
CERN openlab

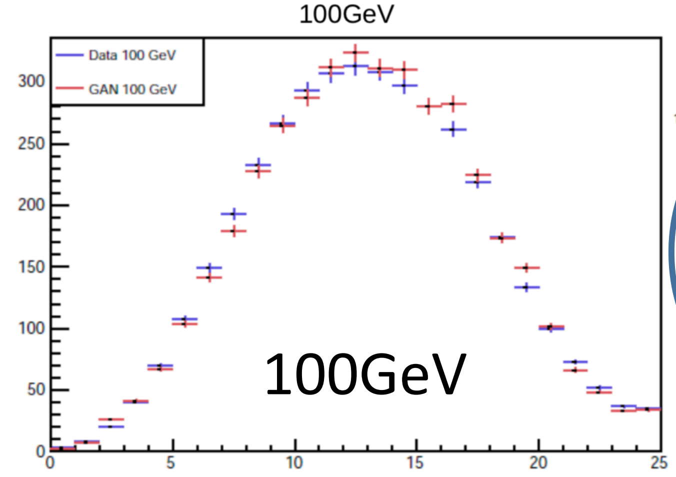
Validation and optimisation

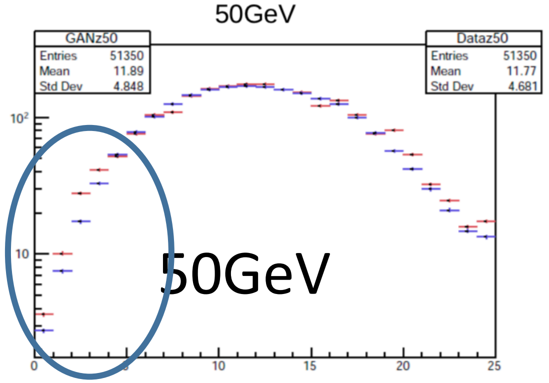
- Detailed GAN vs GEANT4 comparison (More than 200 Plots!)
  - High level quantities (shower shapes)
  - Calorimeter response (single cell response)
  - Particle properties (primary particle energy)
- Optimisation on
  - Network Architecture (Layers, filters, kernels, initialisation)
  - Losses definition
  - Data pre-processing
- Results agree within a few % to Geant4 (sometimes labelled “DATA” in next slides ☺)
- Run TriForce classification and regression engine on GAN generated data
  - (Matt Zhang, https://github.com/BucketOfFish/Triforce_CaloML)

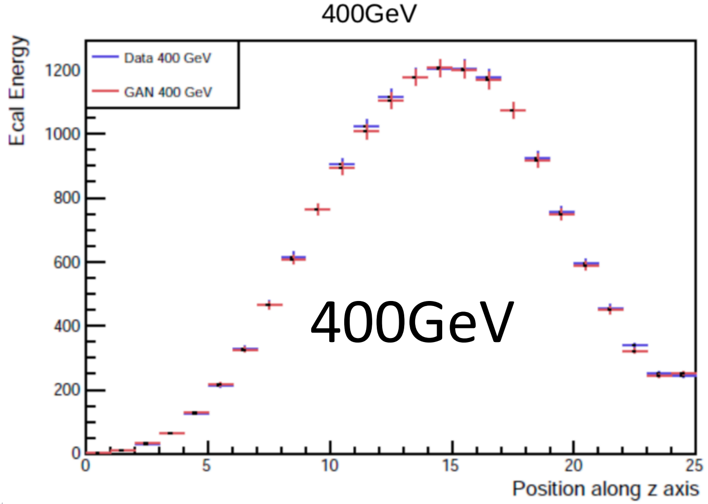
We run on Caltech ibanks GPU cluster thanks to Prof M. Spiropulu

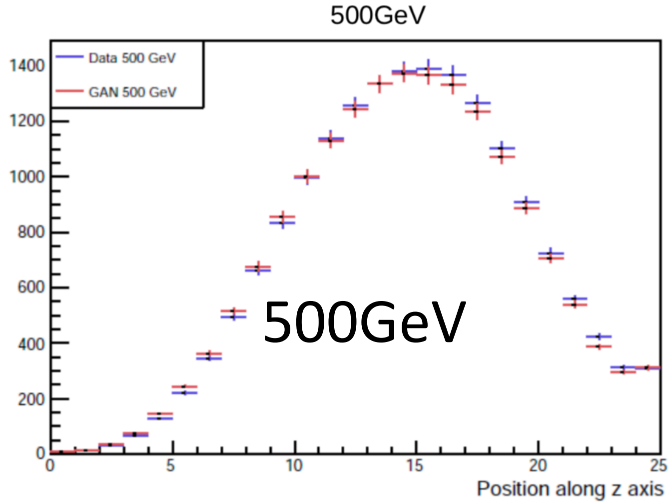
Electrons shower shapes

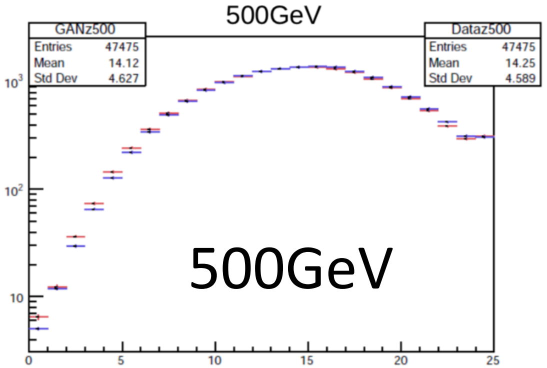






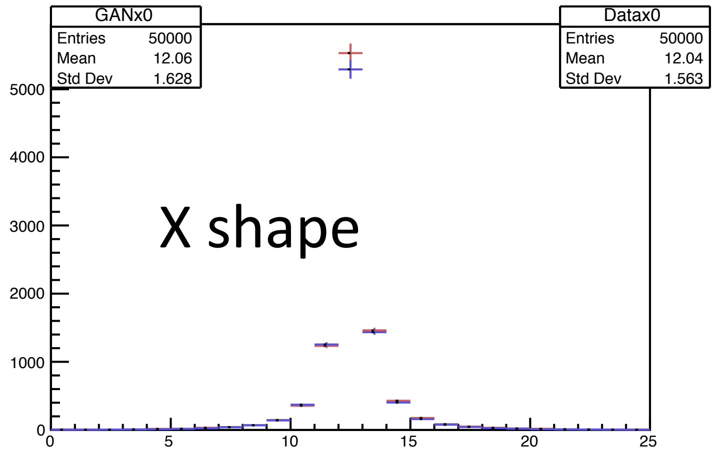


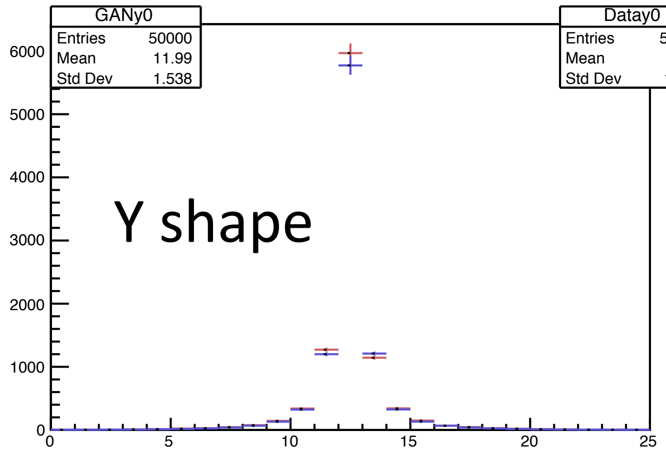


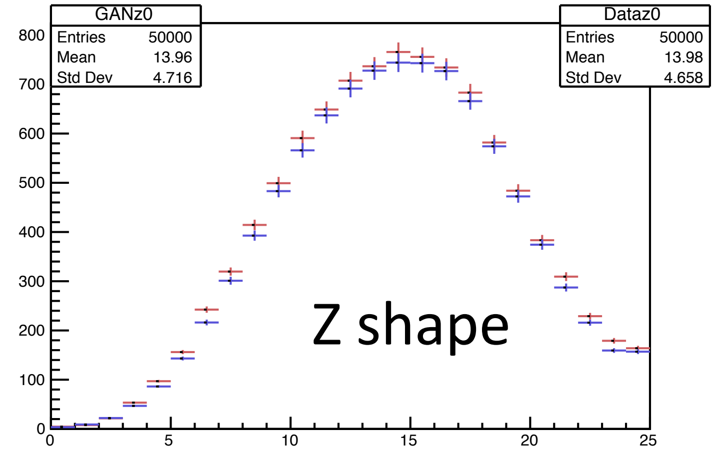


Neutral Pions

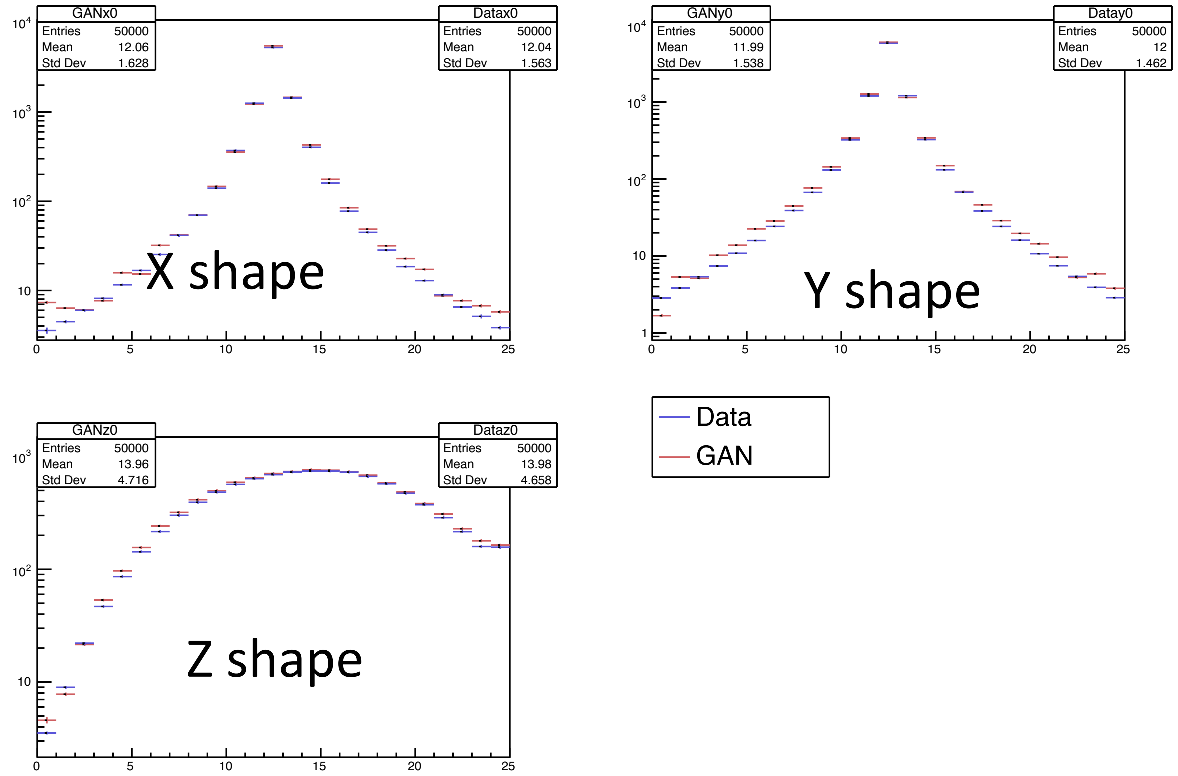
10-500 GeV





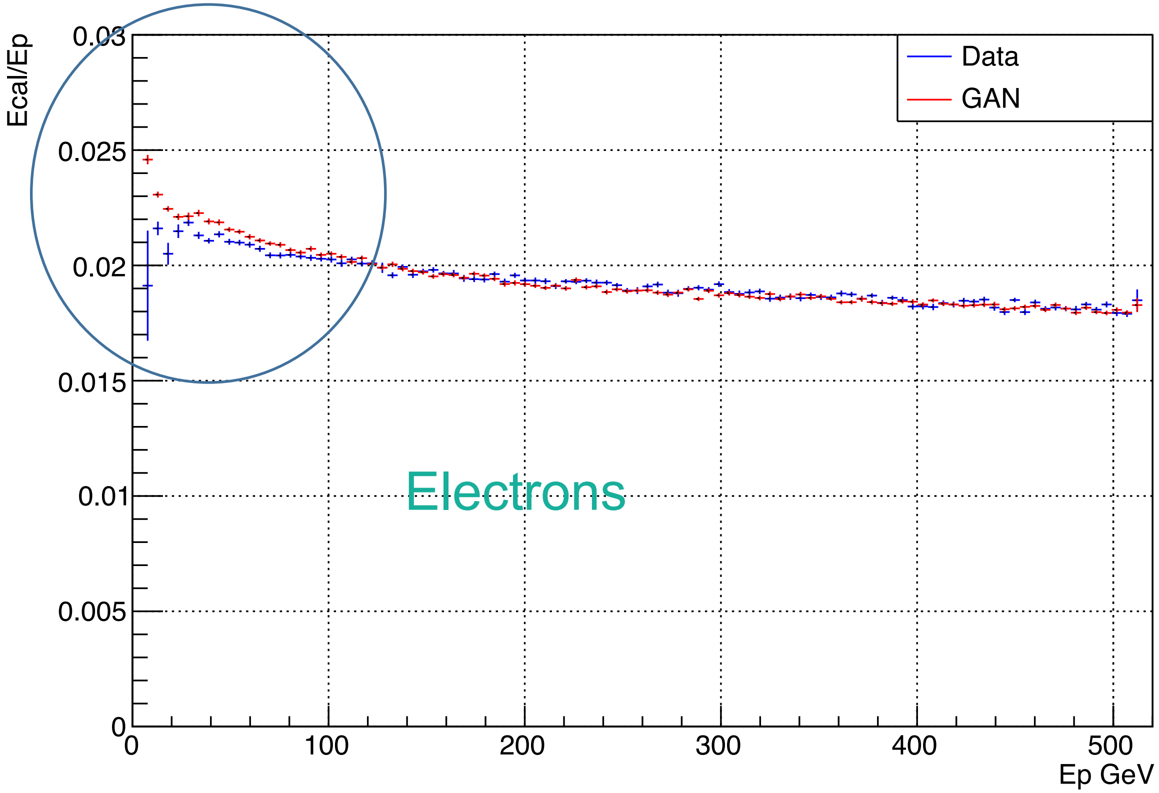


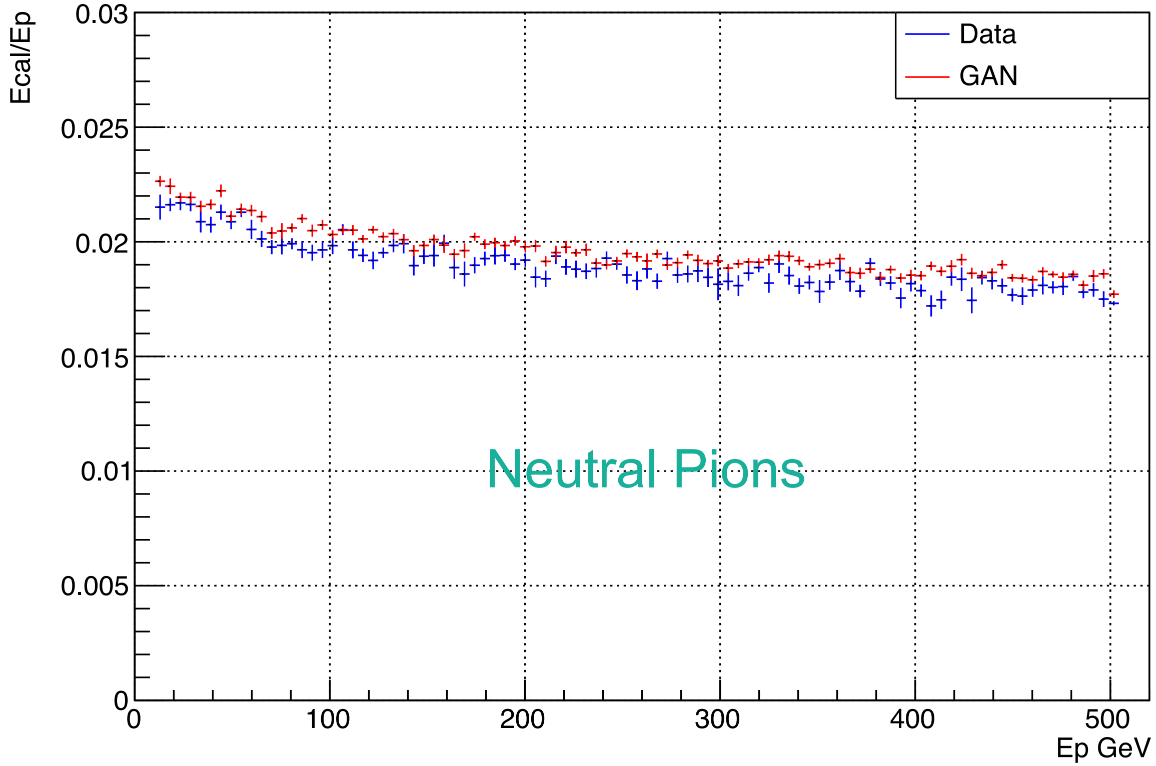
Log scale



CERN openlab

Deposited energy and sampling fraction





GAN seems to slightly overestimate slightly neutral pions energy deposits

Computing resources & Generalisation

Distributed training

Hyper-parameter scans

Computing performance

Distributed training is needed

Inference:

Geant4: 17 s/particle **vs** 3DGAN: 7 ms/particle

➔ speedup factor > 10000!!

➔ Testing inference on accelerators (GPUs, FPGAs) and dedicated hardware

Training:

45 min /epoch on Tesla P100

Introduce **data parallel** training based on MPI

Test several libraries

Run on HPC clusters and Cloud (HNSciCloud providers)

Time to create an electron shower		
Method	Machine	Time/Shower (msec)
Full Simulation (geant4)	Intel Xeon Platinum 8180	17000
3d GAN (batch size 128)	**Intel Xeon Platinum 8180 (TF 1.4)**	**7**

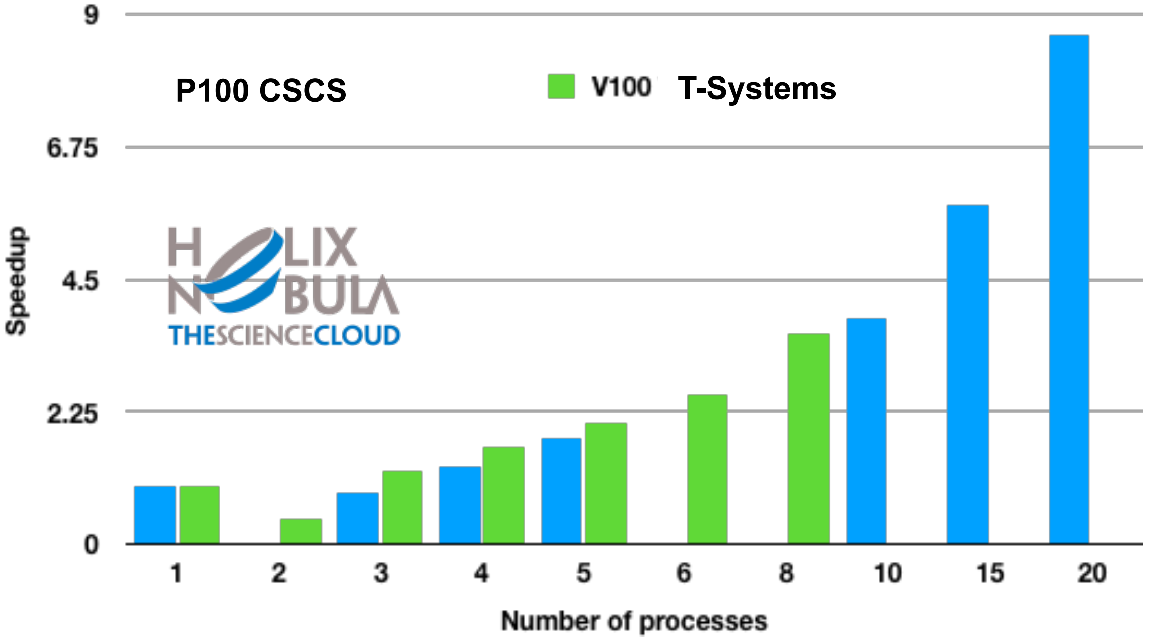
Optimisations in TF (1.2) + MKL-DNN

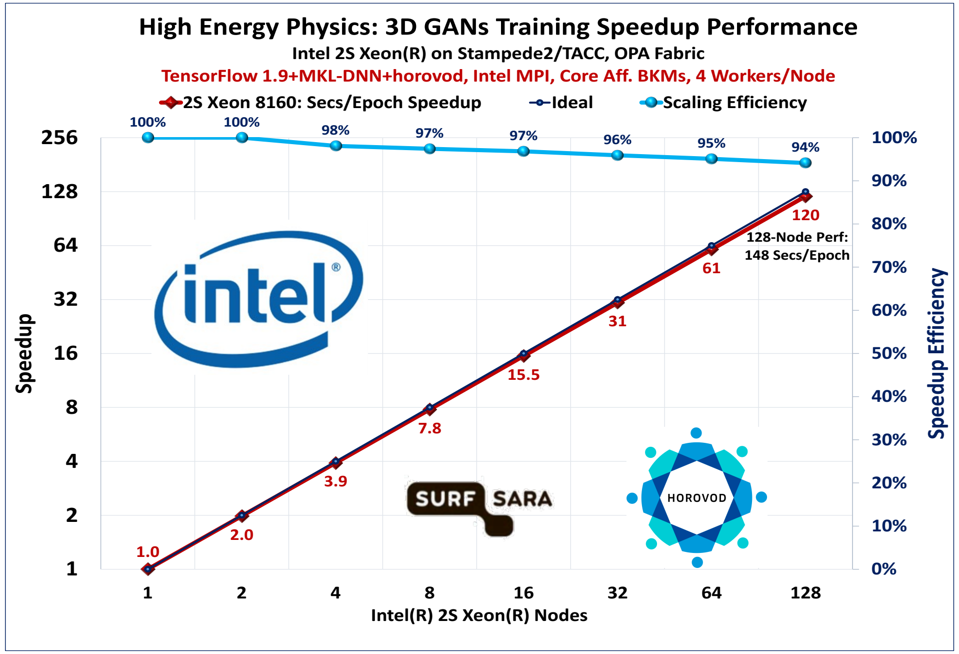
1 msec/shower

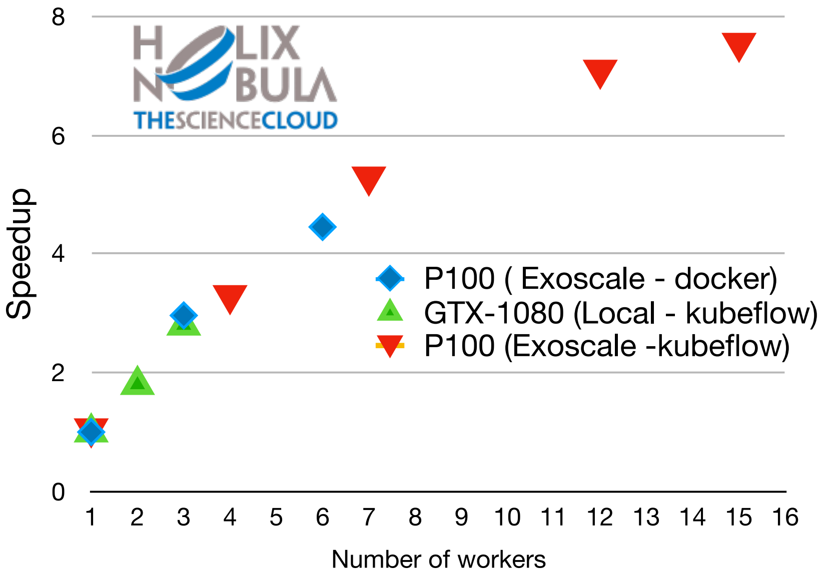
→ Speedup factor >10000

Distributed training

- Cloud deployment via docker + Kubernetes/Kubeflow (R. Rocha CERN IT-CM)
- Frameworks
  - Horovod
  - mpi_learn
- Hardware resources
  - Cloud (HNSciCloud)
  - HPC centers (Oakridge – TACC)







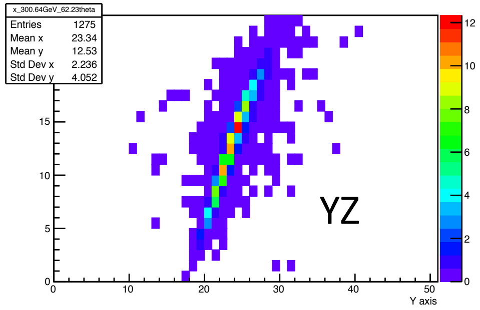
Generalisation

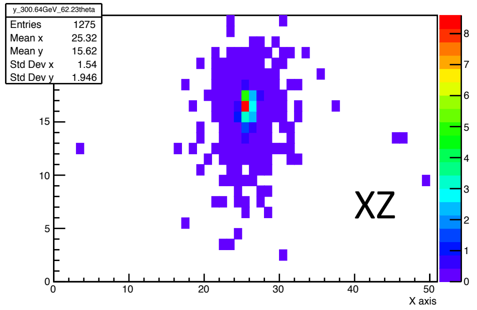
Electrons enter the calorimeter within a 60°-120° angle range

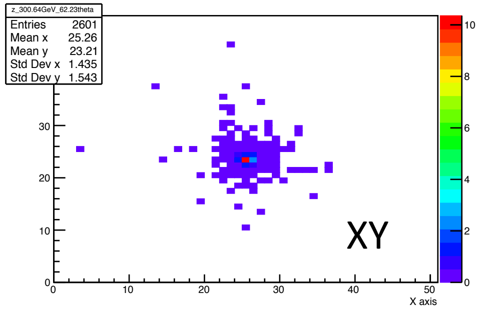
Wider/asymmetric image size (51x51x25)

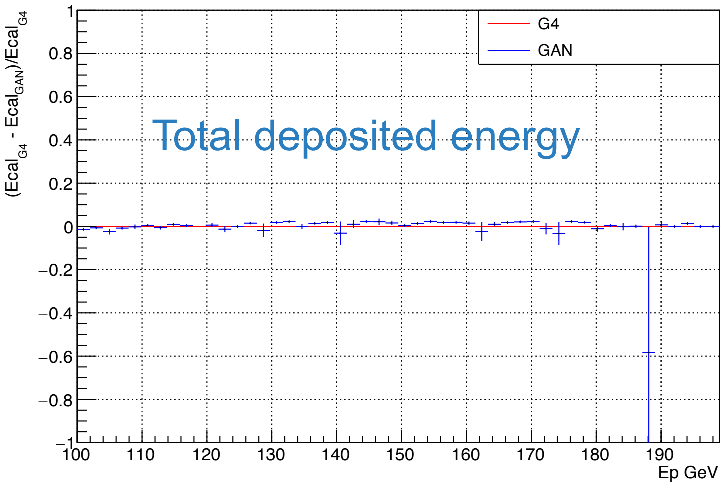
Minimal architecture changes “by hand”

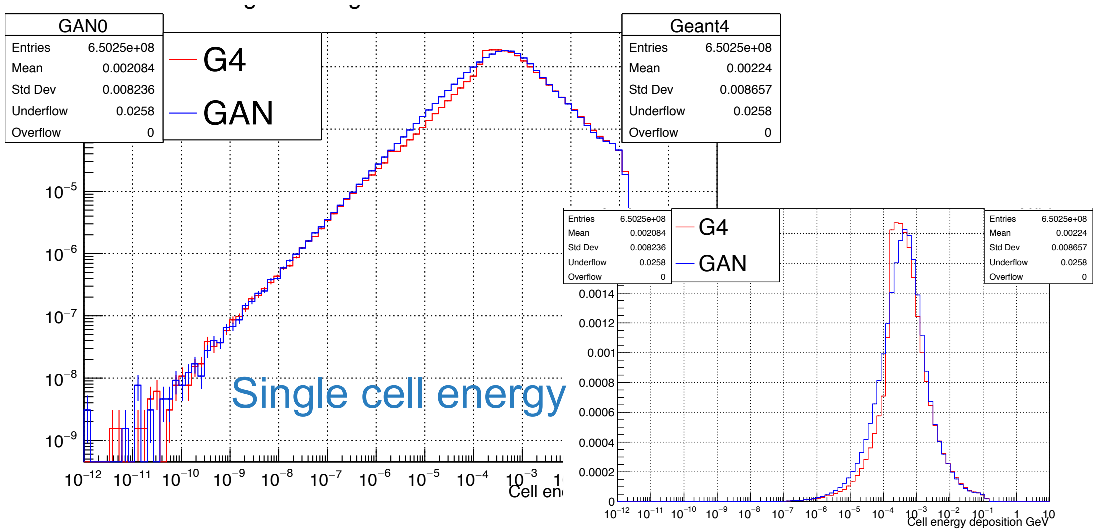
- Adjust convolution parameters
- Additional terms in the loss function:
  - Angle-related term
  - Constrain energy spectrum











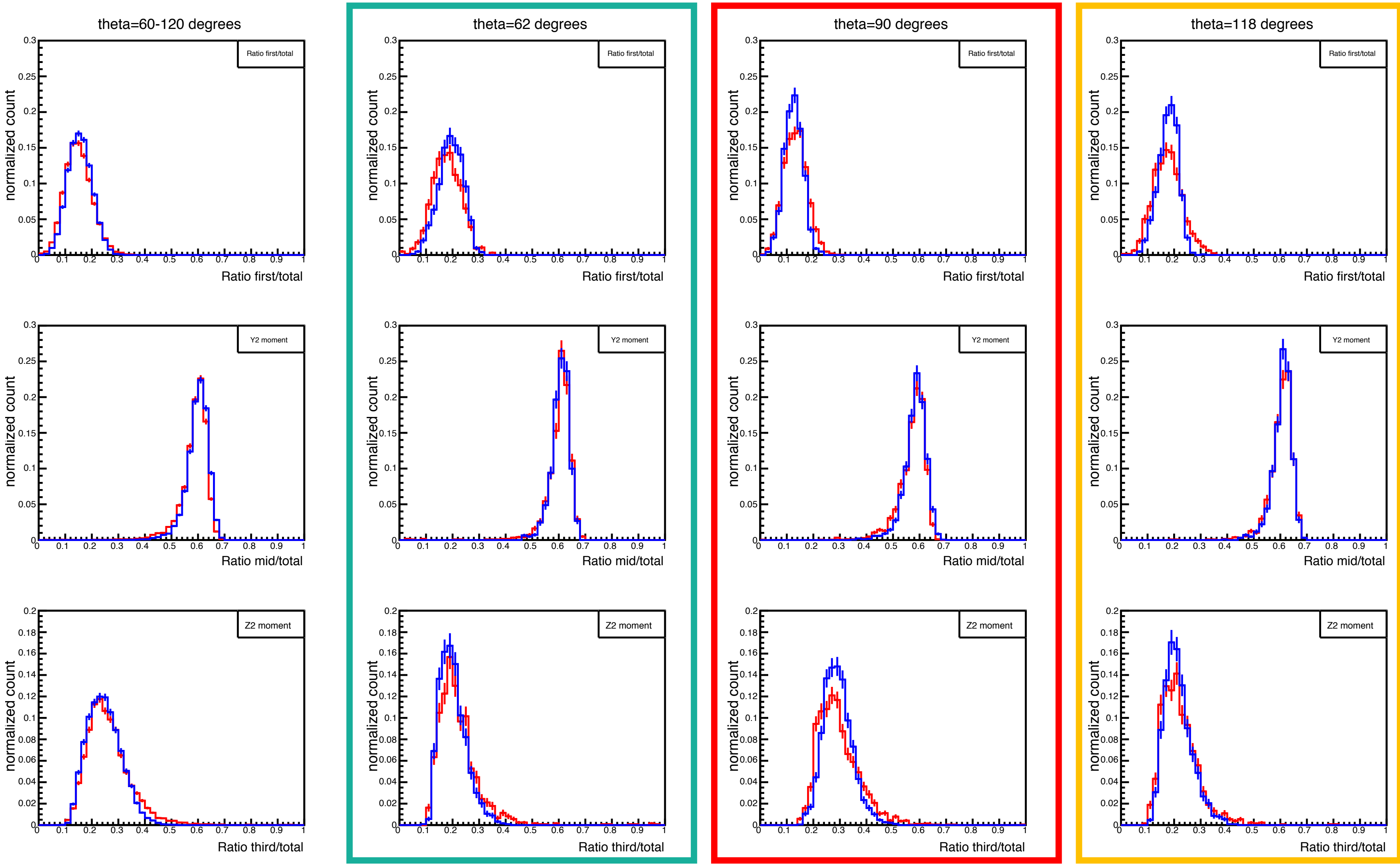
CERN openlab

Angle dependence (I)

Spit calorimeter in 3 layers along its depth (cells: 1-8, 8-16, 24-25)

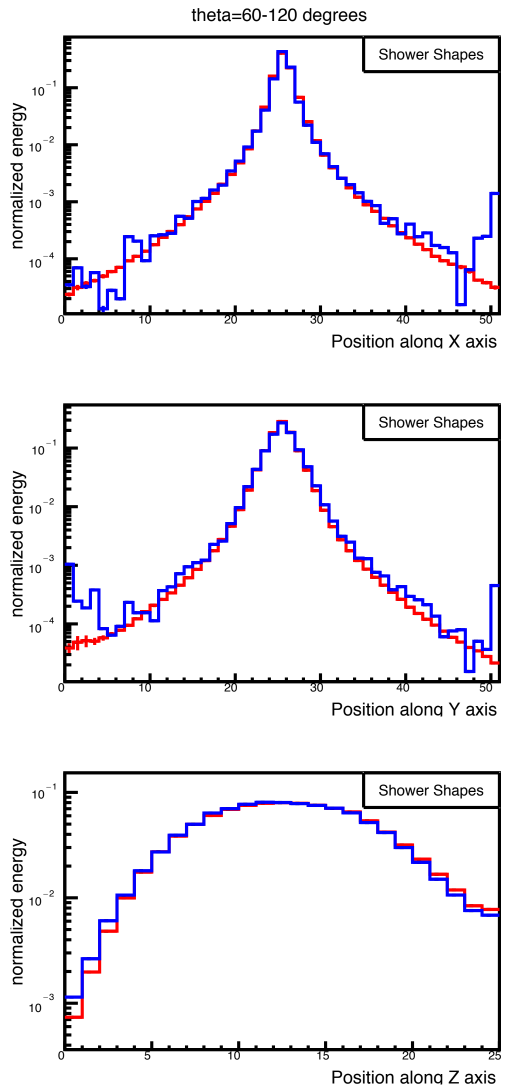
Measure energy deposited in layers wrt total

inclusive | 60° incident angle | Ortogonal | 120° incident angle

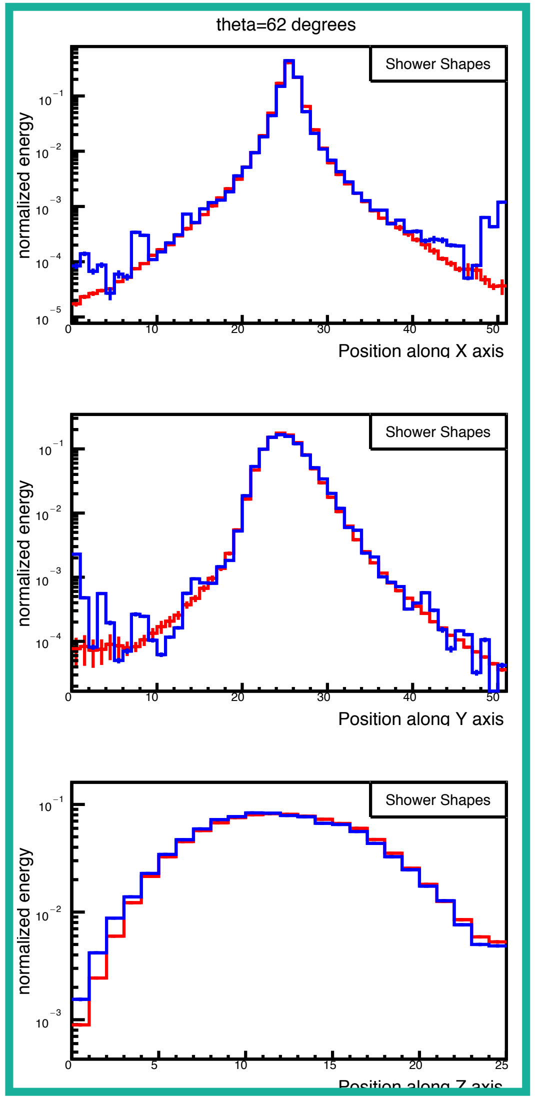


Angle dependence (II): shower shapes

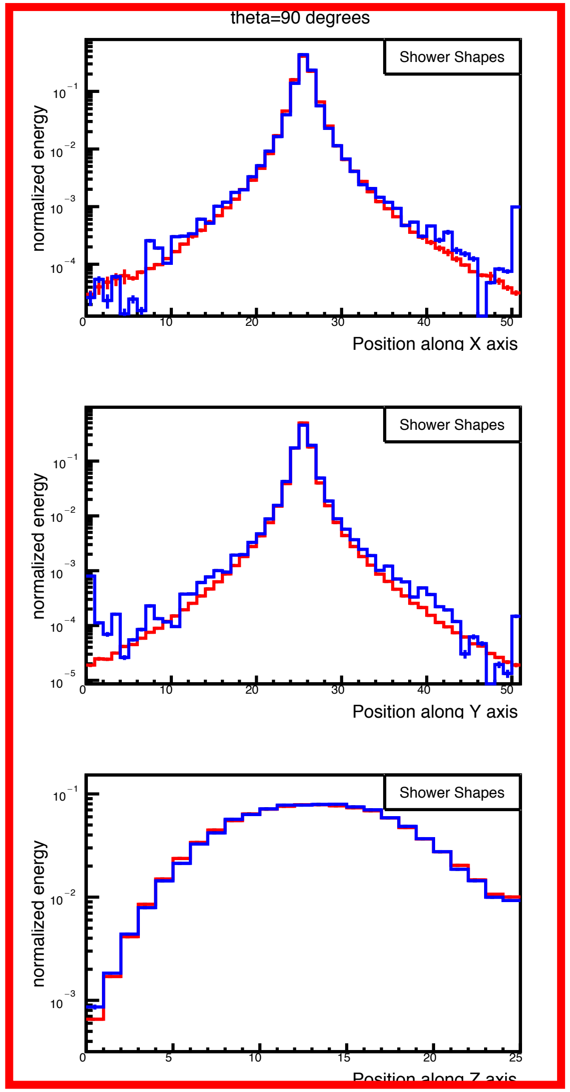
inclusive



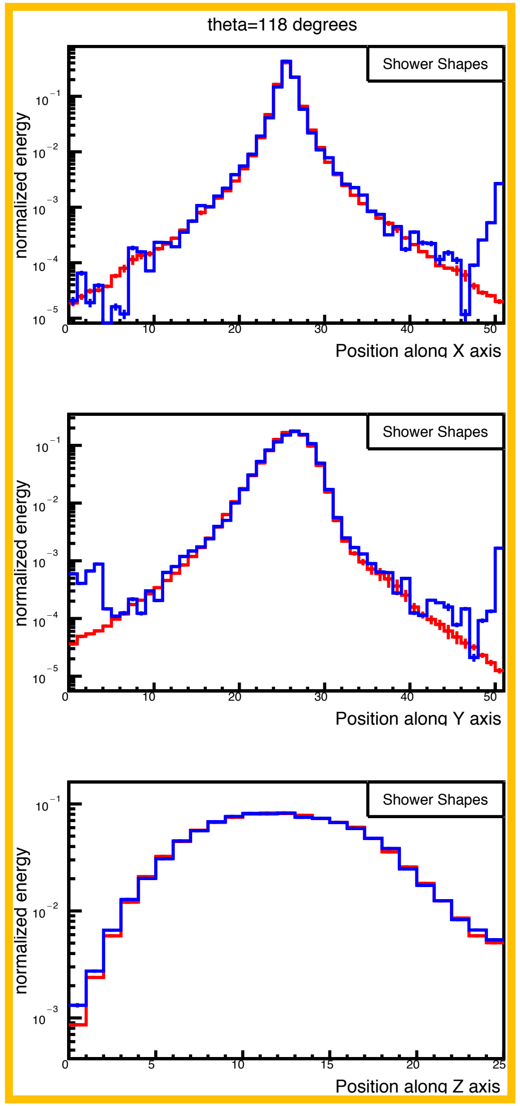
60° incident angle



Ortogonal



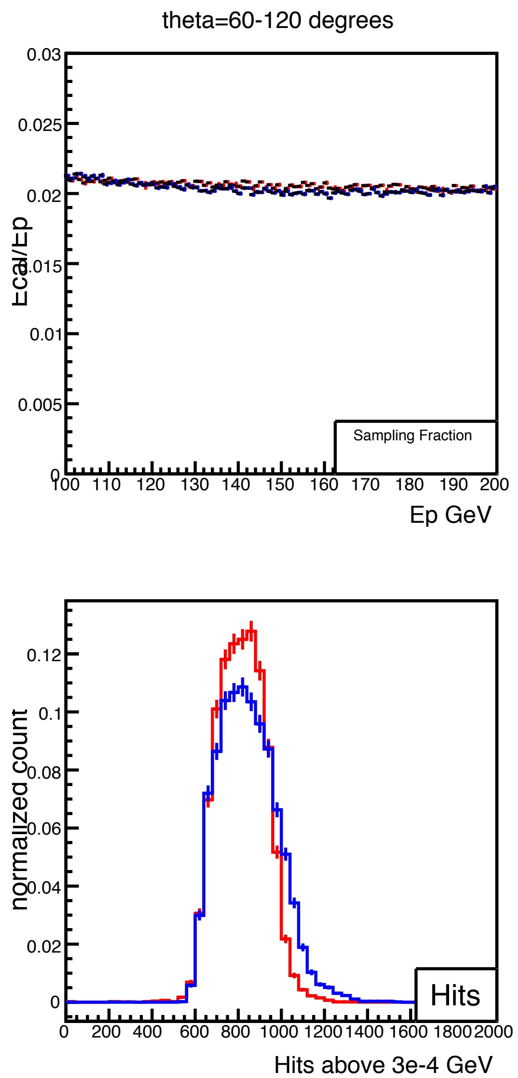
120° incident angle



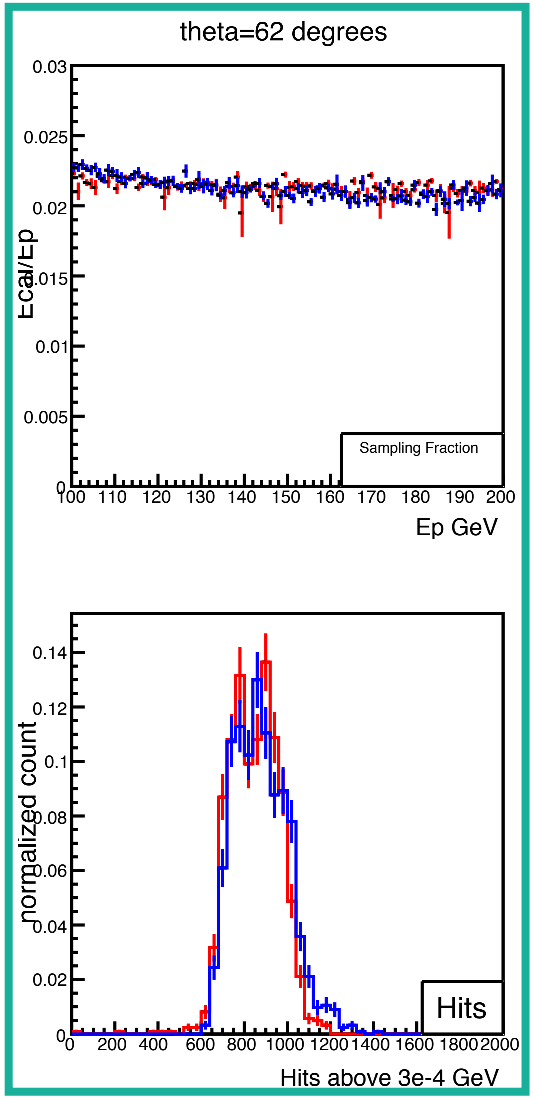
Angle dependence (III)

Sampling fraction annd number of hits

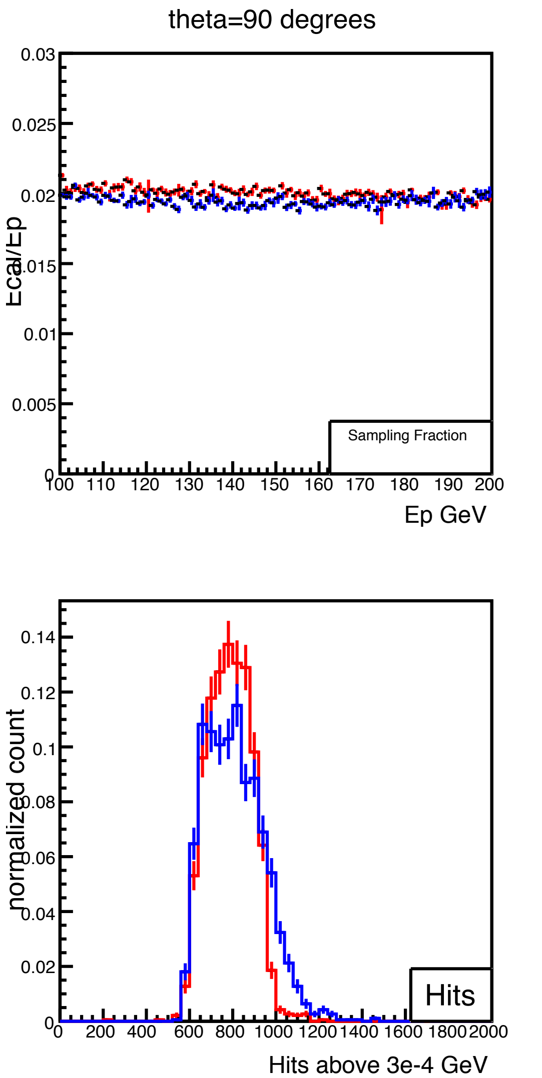
inclusive



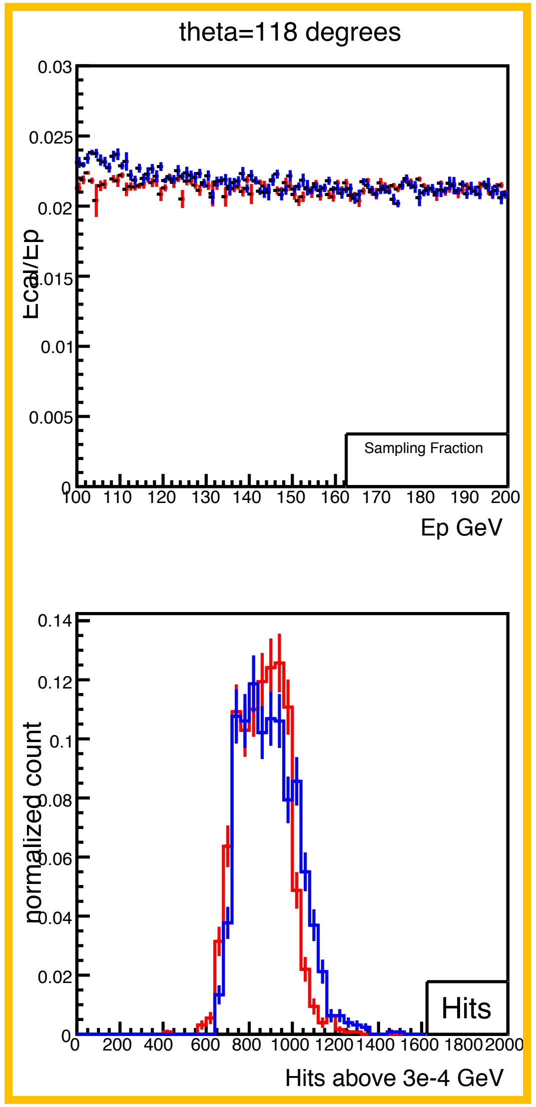
60° incident angle



Ortogonal



120° incident angle

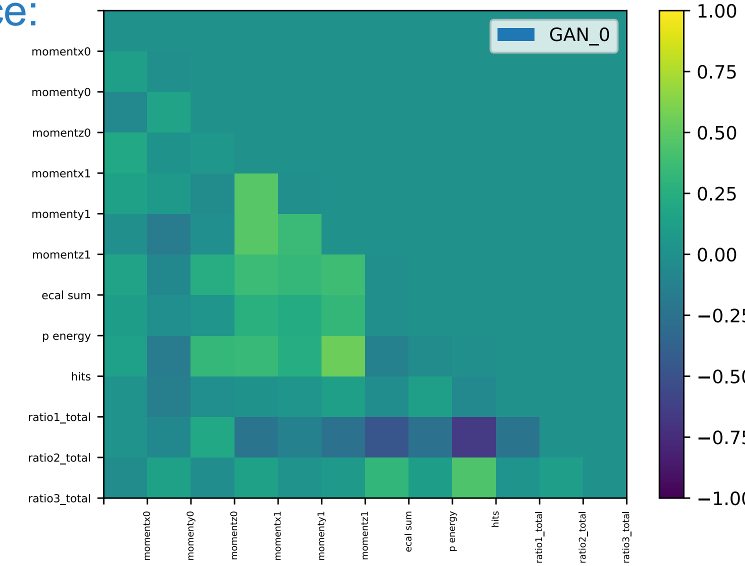


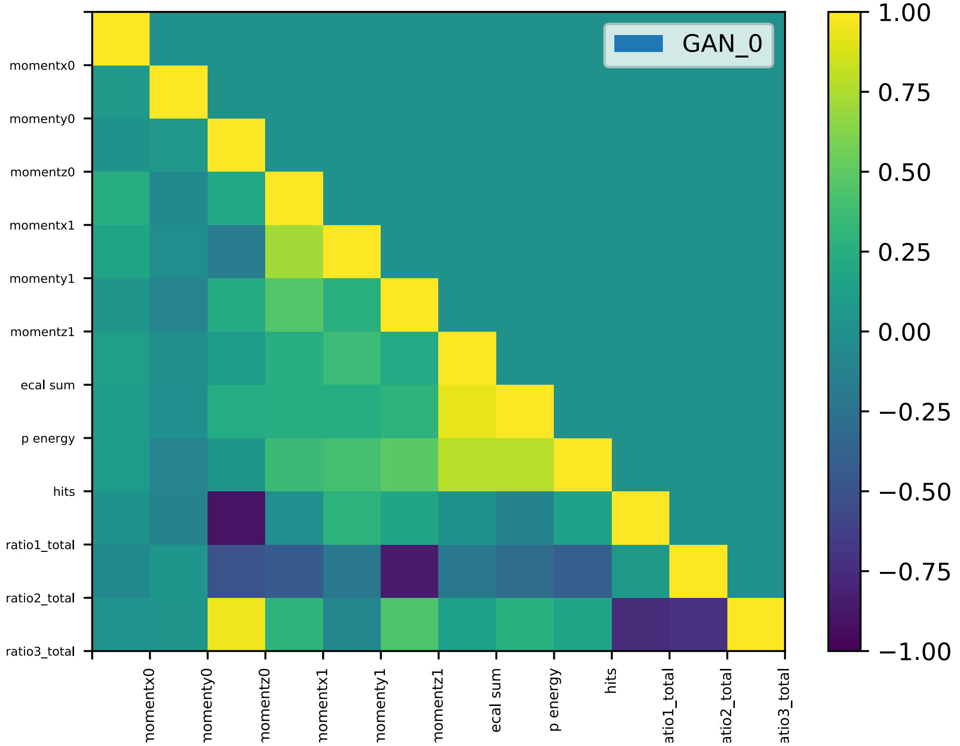
CERN openlab

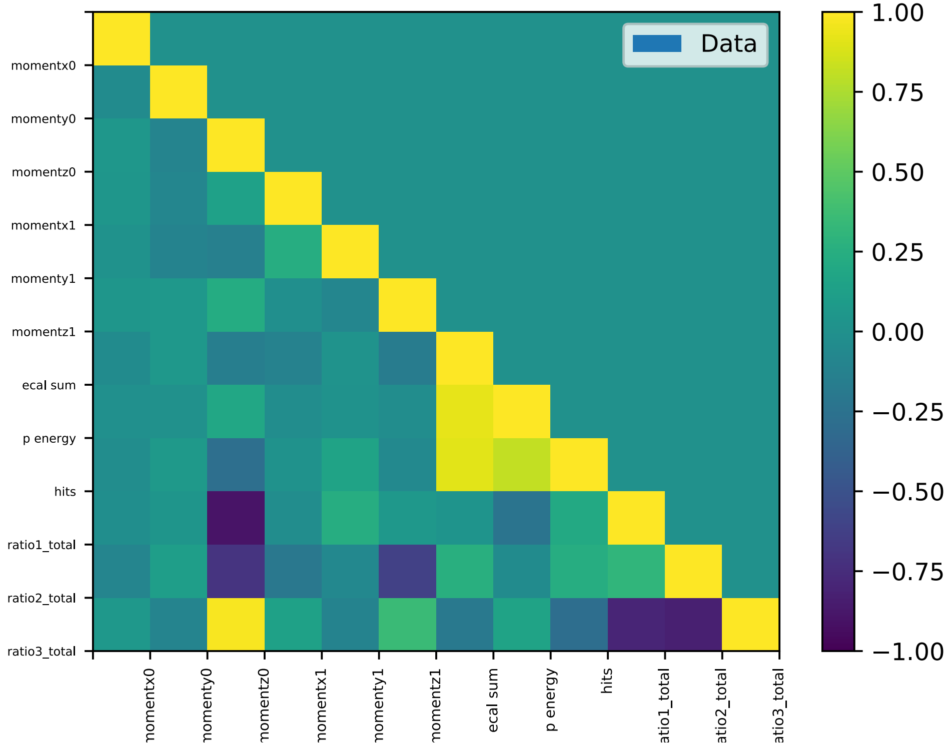
Internal correlations

3DGAN can correctly reproduce most G4 internal correlations

Difference:







Generalisation

Training and architecture hyper-parameters optimisation

How much can we generalise our network to other calorimeters?

- Different geometries, read-out patterns, energy scales

Tuning the right architecture cannot be done by hand

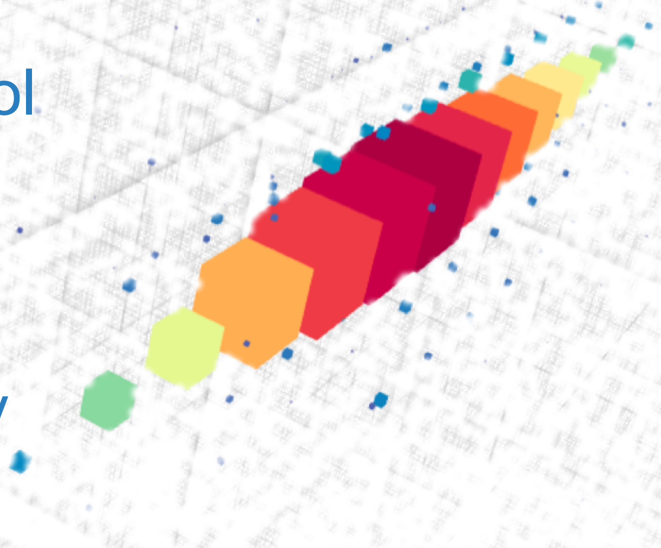
- Full parameter scan is resource/time consuming.

Test different optimisation approaches:

- Sequential Model-Based Optimization
  - Optimize intial architecture candidate, defining a finite set of states to explore
- Reinforcement Learning
  - Network accuracy is the **reward function**. Architecture or hyper-parameter modification are **actions**
- Evolutionary Algorithms
  - Can allow simulataneous weights training and architecture optimisation

mpi-learn integrates a optimisation engine (mpi-opt)

Summary & Plans



3D GAN: first step towards customizable simulation tool

- Agreement to Monte Carlo within few percent
- Work in SFT to test integration in simulation framework

Meta-optimization and hyper-parameters scans are key

- Test of several distributed training approaches
- Understand / optimize performance at scale

We are working heavily on “technological/computing” aspects in collaboration with industry (Intel, IBM, Google)

Test different platforms for inference and training: CPUs, accelerators (GPUs, FPGAs)

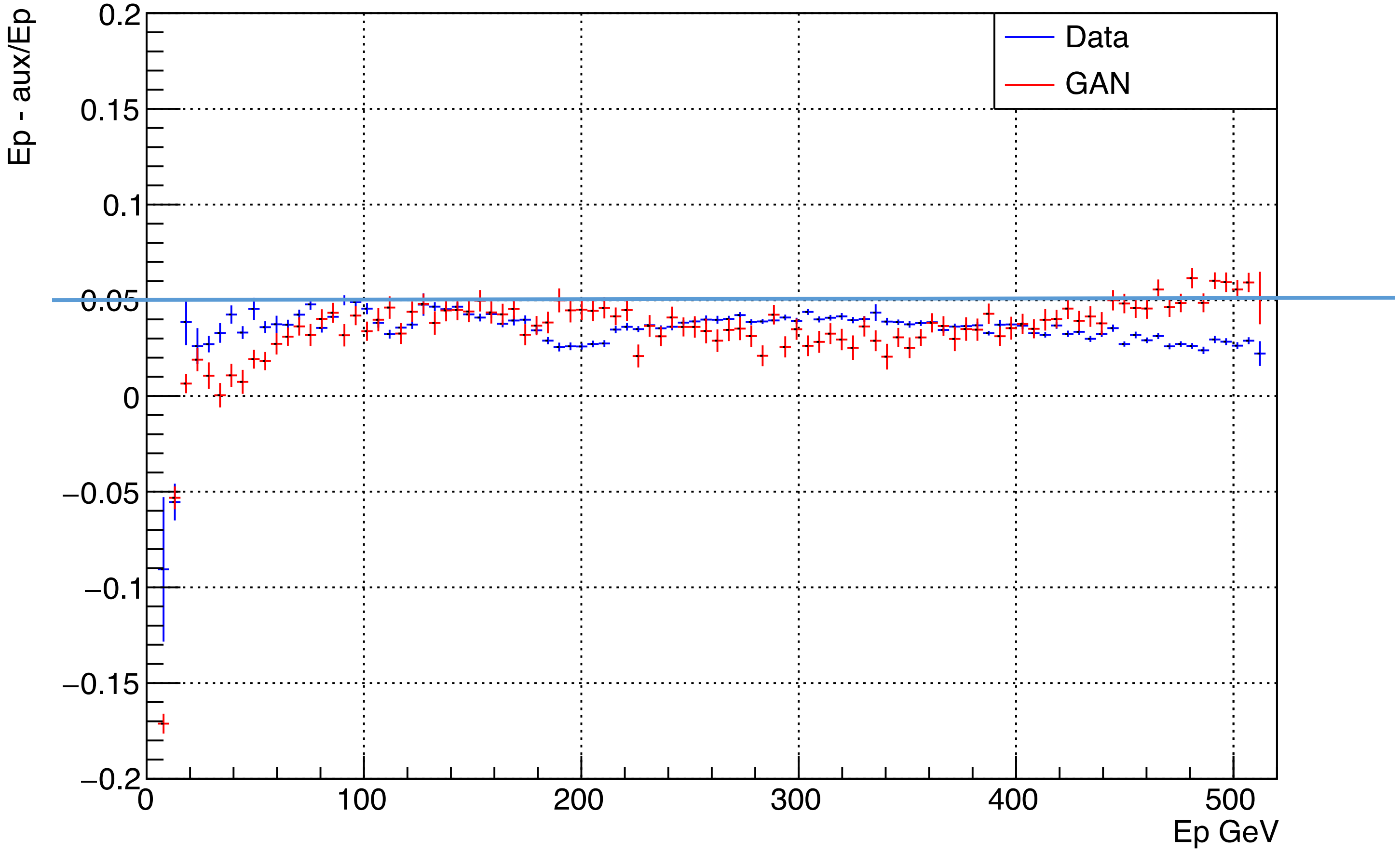
Questions?

Sofia.Vallecorsa@cern.ch

Thanks!

CERN openlab

Discriminator regression on input energy

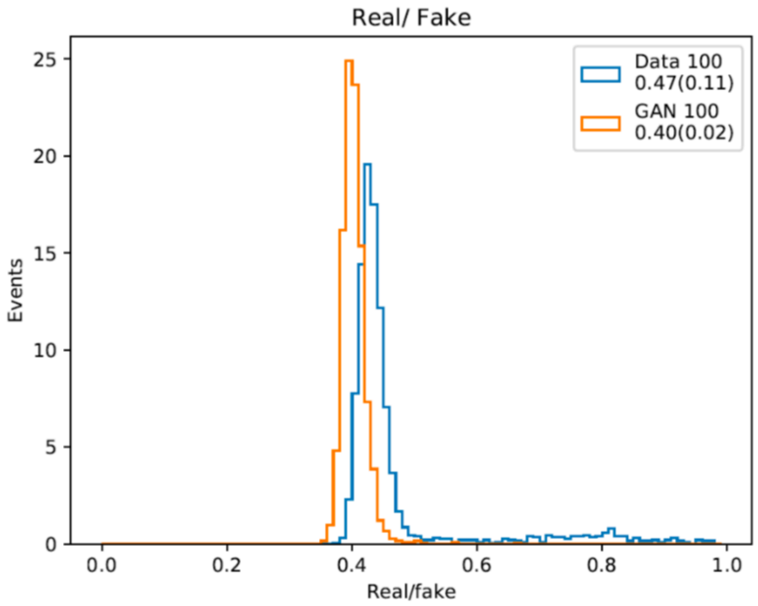


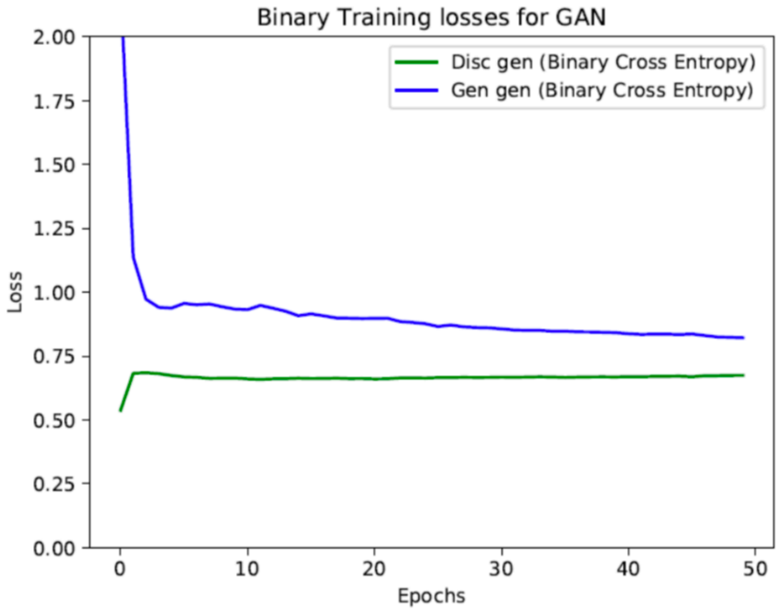
5% error on auxiliary energy regression

Conditioning and auxiliary tasks

Loss is linear combination of 3 terms:

- Combined cross entropy (real/fake)
- Mean absolute percentage error for regression tasks

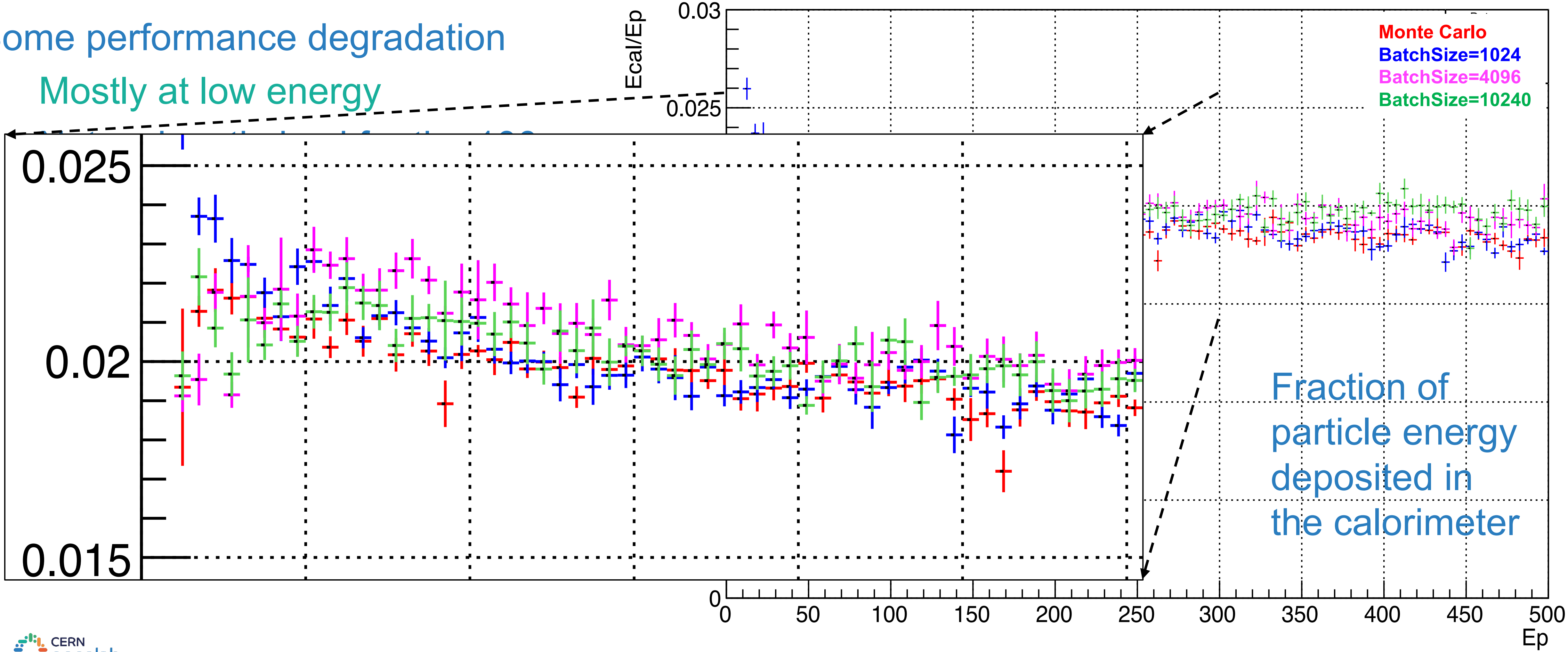




Physics performance

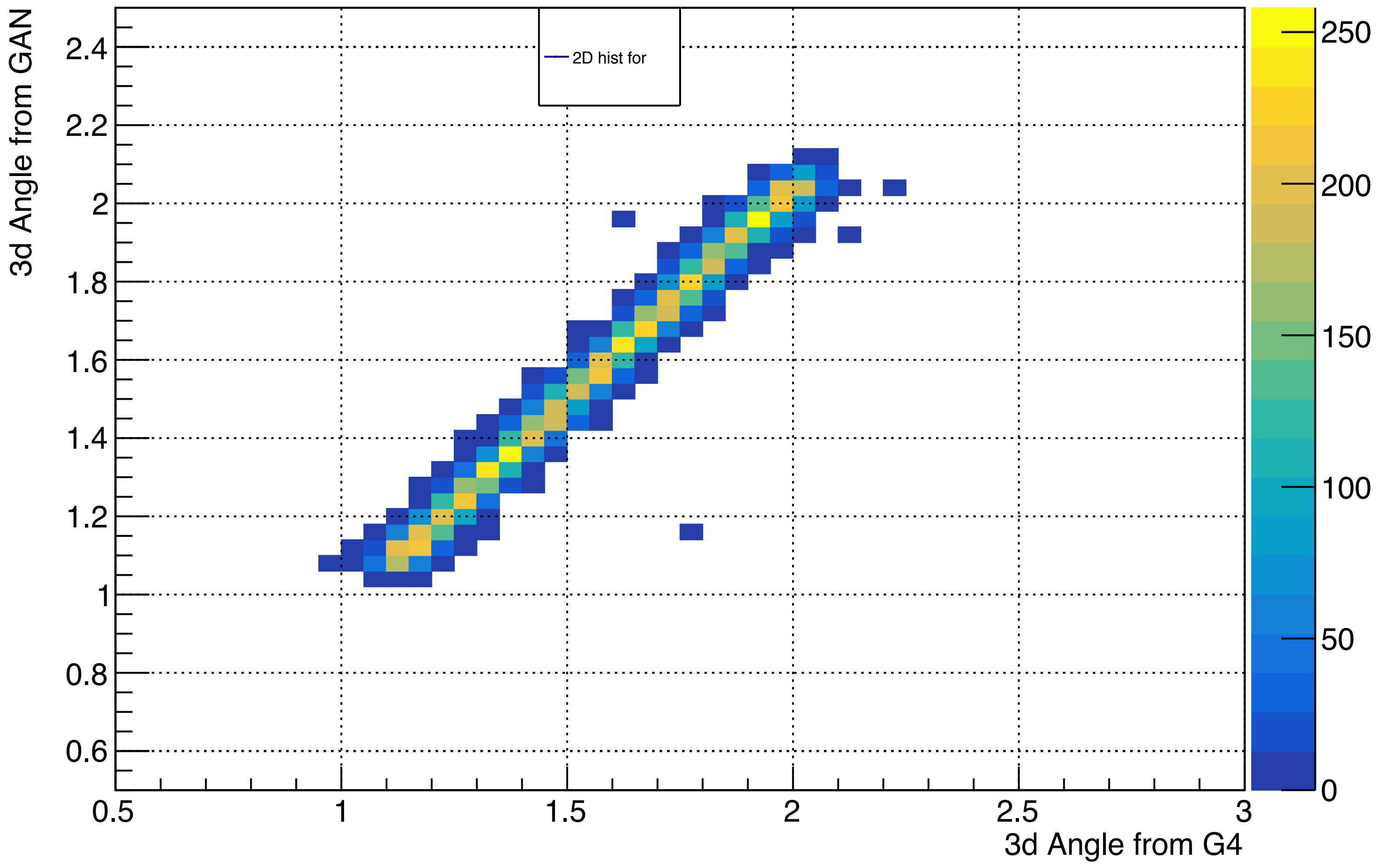
Some performance degradation

Mostly at low energy



CERN openlab

Angles


2D hist for
3d Angle from GAN
3d Angle from G4
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.5
1
1.5
2
2.5
3
250
200
150
100
50
0

https://github.com/vlimant/mpi_opt

https://github.com/vlimant/mpi_learn

mpi-opt

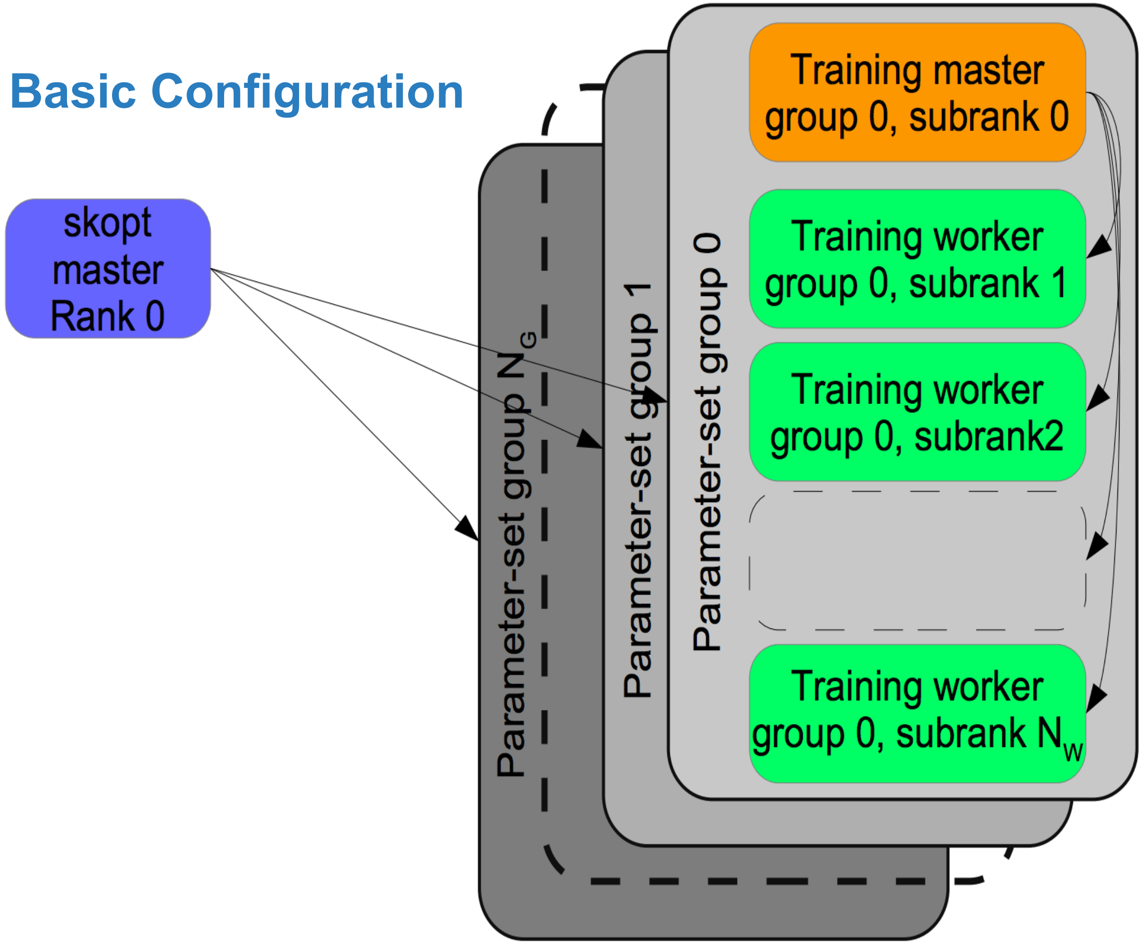
Parallelise optimisation with mpi-learn

Bayesian optimiser

Evolutionary approach

- One master runs optimisation (skopt)
- N_G groups of nodes train on a parameter-set
  - One training master
  - N_W training workers
- Can also run in “sub-masters mode¨
  - Sub-masters perform intermediate averaging

Basic Configuration



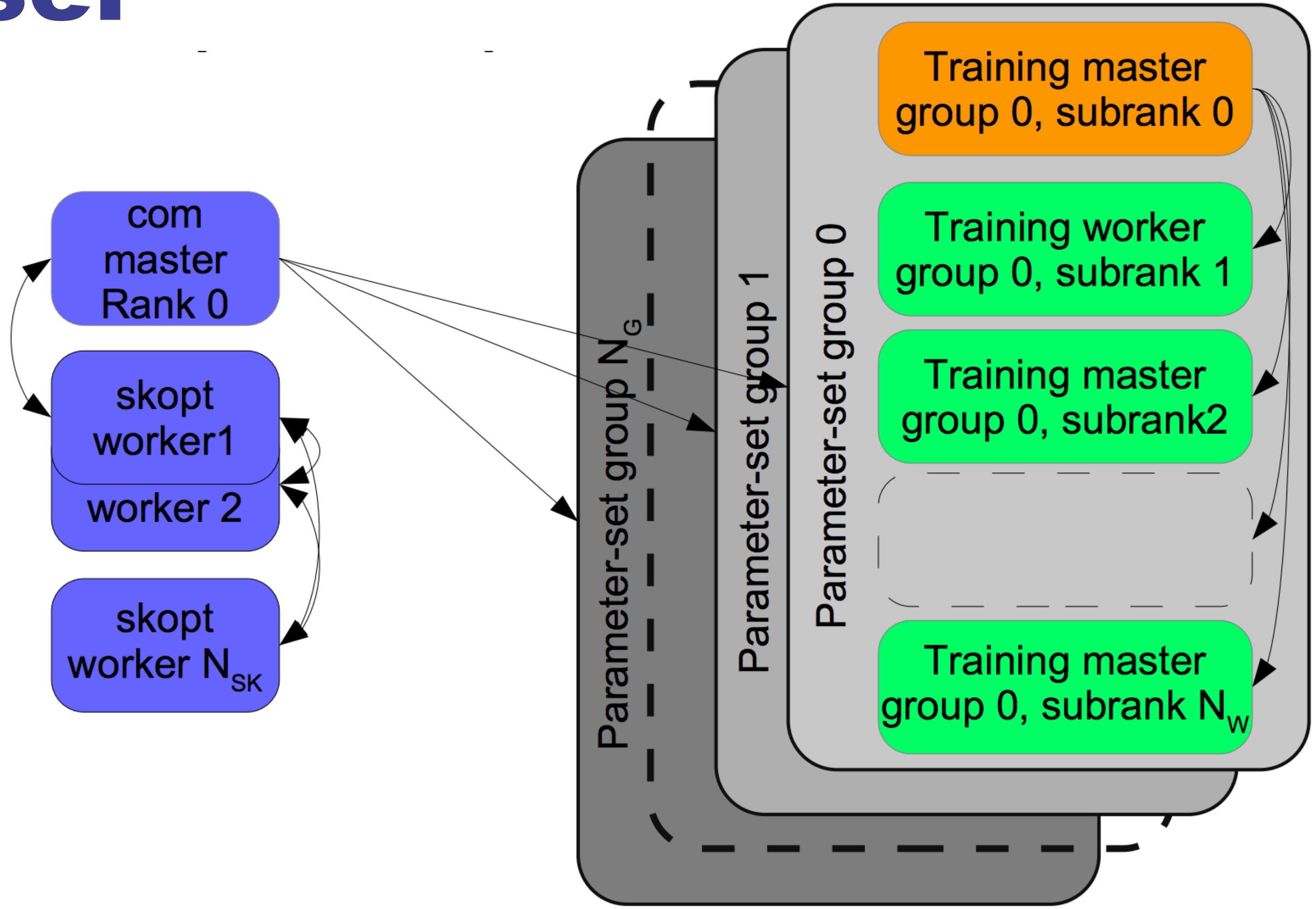
Parallel optimiser

One master running communication of parameter set

N_{SK} workers running the bayesian optimization

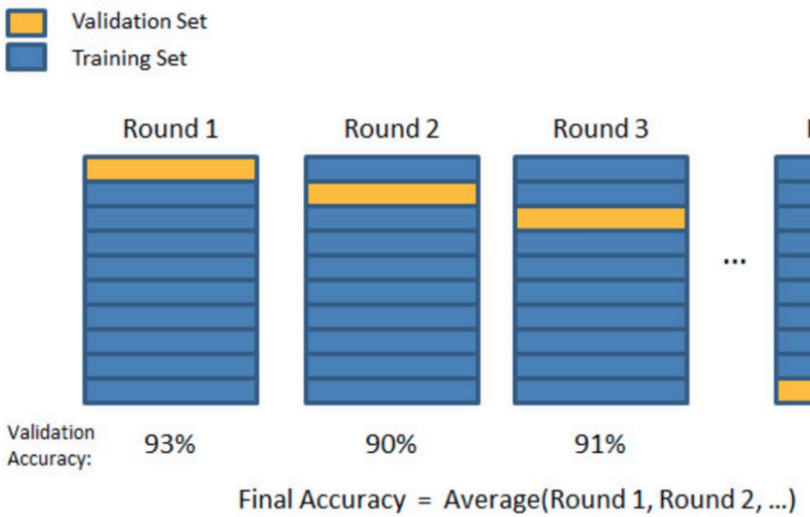
N_G groups of nodes training on a parameter-set on simultaneously

- One training master
- N_W training workers



prevent working groups to be idle while the optimisation fit is performed

K-folding cross validation



Estimate performance over different validation parts of the training dataset

Account for variance from multiple sources

- One master runs the bayesian optimization.
- Receives the average fom over N_F folds of the data
- N_G groups of nodes training on a parameter-set simultaneously
- N_F groups of nodes running one fold each
  - One training master
  - N_W training workers