3D convolutional GAN for fast simulation

GeantV meeting

F. Carminati, G. Khattak, S. Vallecorsa

June 2018

Outline

- Introduction
- Status
  - Physics performance validation
- Plan for 2018
  - Generalisation
  - Optimisation of computing resources
- Summary

A plan in two steps

Is generative models output 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?

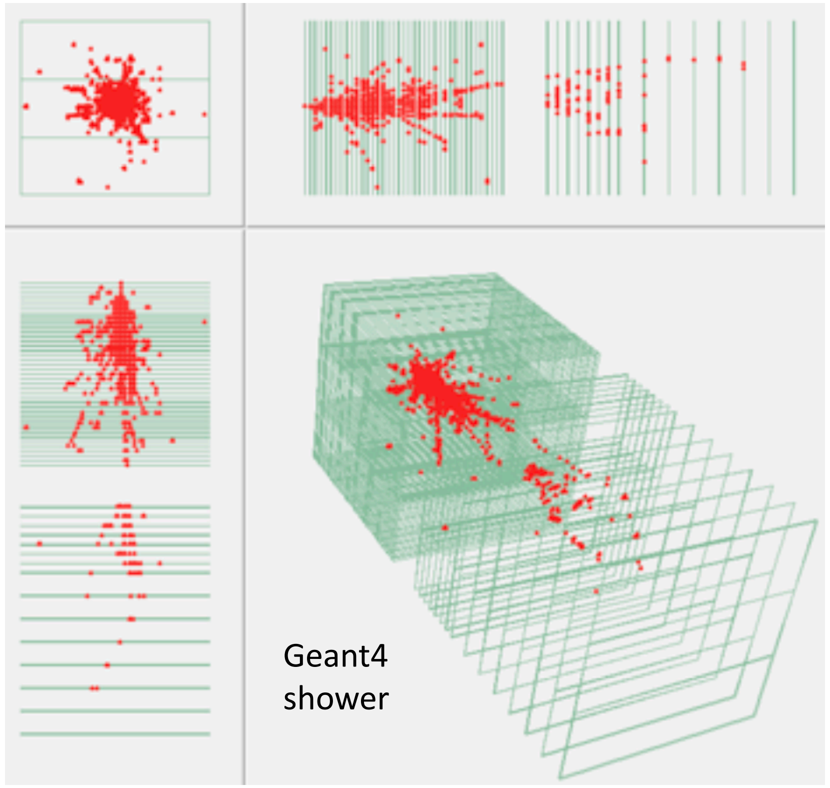
Can we “adjust” architecture to fit a larger class of detectors?

What resources are needed?

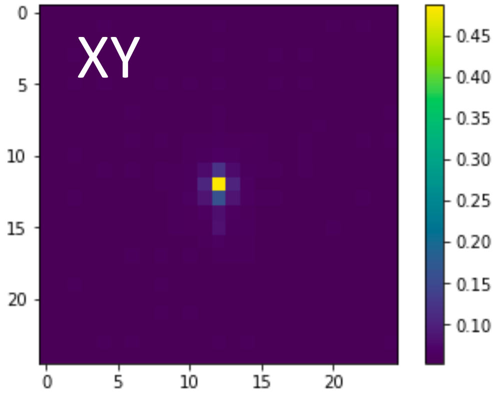
- Prove generalisation is possible
- Understand and optimise computing resources

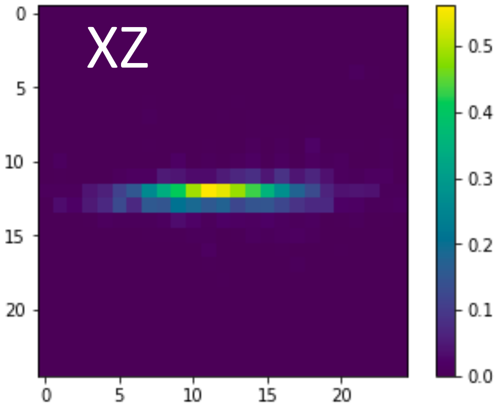
Proof of concept, benchmarking and validation

CLIC calorimeter simulation



- 1M single particle samples (e,γ,π)
  - Flat spectrum (10-500) GeV
  - Orthogonal to detector surface
  - 60°-120° random incident angle (NEW!)
- Stored as a 25x25x25 HDF5 dataset



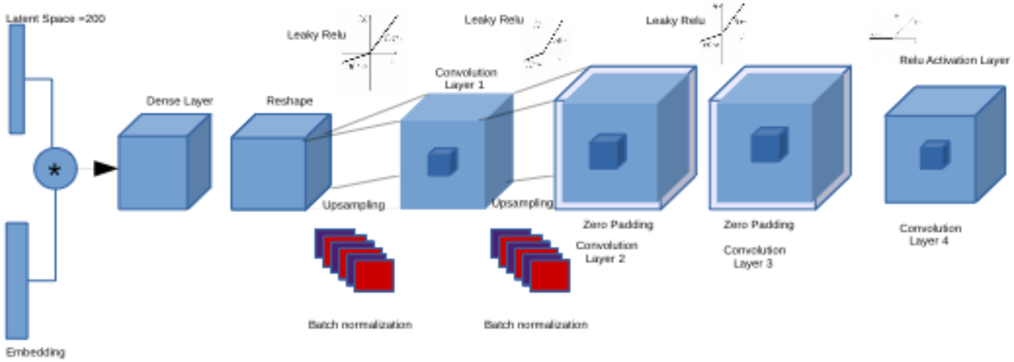


- Data is essentially a 3D image
  - Highly segmented
  - Sparse.

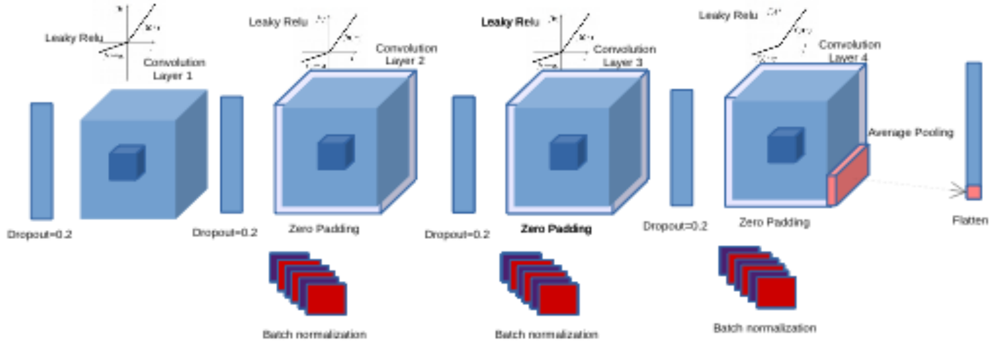
5

3D convolutional GAN

- Similar discriminator and generator models
  - 3d convolutions (keep X,Y symmetry)
  - Tested several tips&tricks from literature*
  - Some helpful (no batch normalisation in the last step, LeakyRelu, no hidden dense layers, no pooling layers)
  - RMSProp optmiser for both networks
- Batch training
- Implementation in keras (TF backend)



GENERATOR



DISCRIMINATOR

*https://github.com/soumith/ganhacks

Conditioning and auxiliary tasks

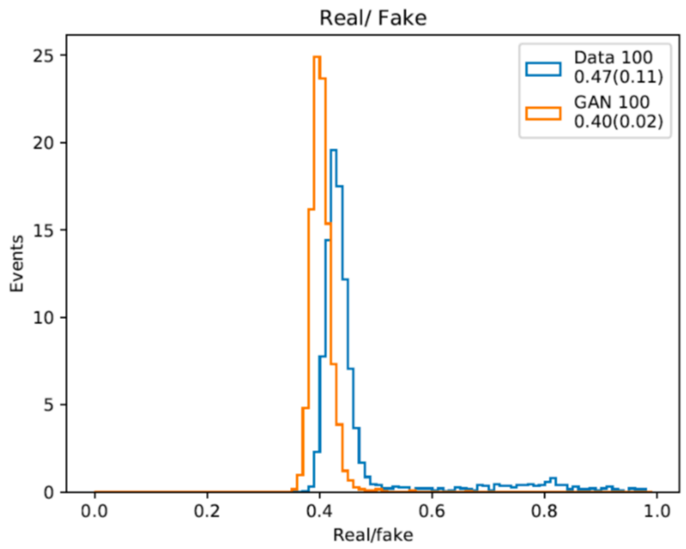
Condition training on several input variables (particle type, energy)

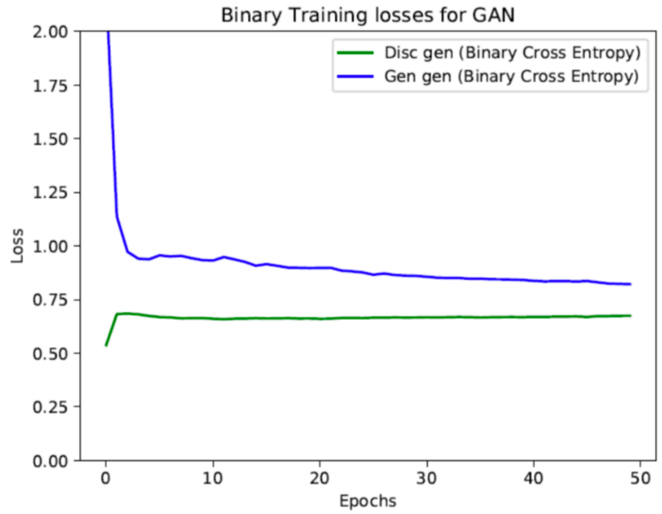
Auxiliary regression tasks assigned to the discriminator: primary particle energy and deposited energy

Loss is linear combination of 3 terms:

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

Generalise to multi-class approach (or multi-discriminator approach): angle..



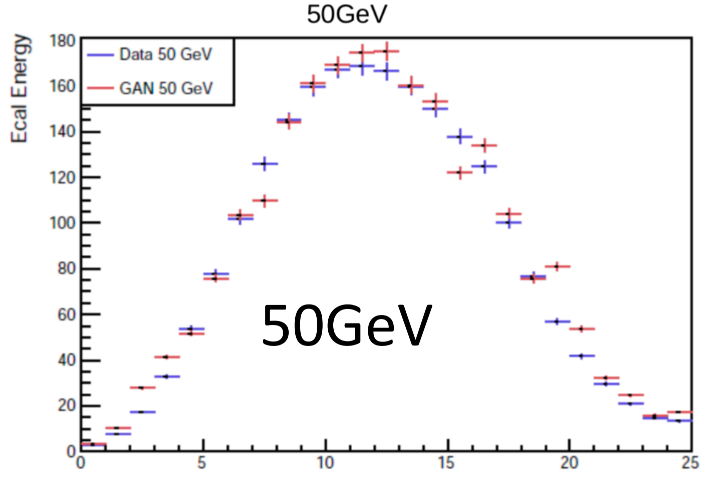


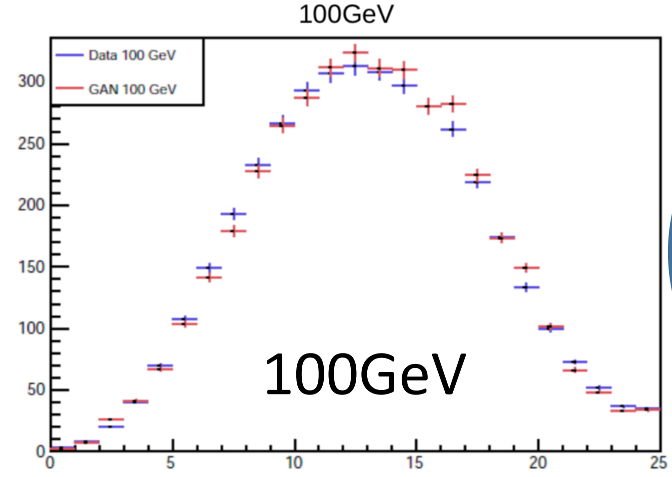
Validation and optimisation

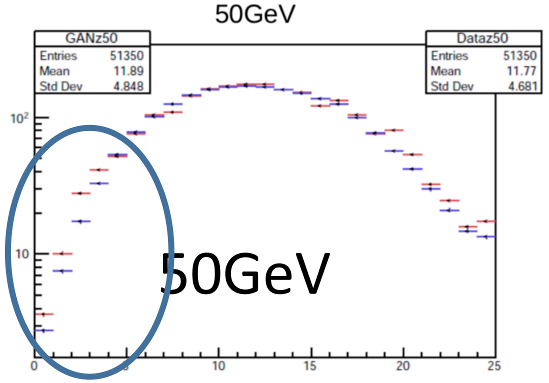
- Detailed GAN vs GEANT4 comparison (More than 200 Plots!)
  - High level quantities (shower shapes)
  - Detailed calorimeter response (single cell response)
  - Particle properties (primary particle energy)
- Optimisation on
  - Network Architecture (Layers, filters, kernels, initialisation)
  - Losses definition
  - Data pre-processing
  - Rely on GAN losses only !! No physics variable explicitly constrained!
- Results agree within a few % to Geant4 (labelled “DATA” in next slides ☺)

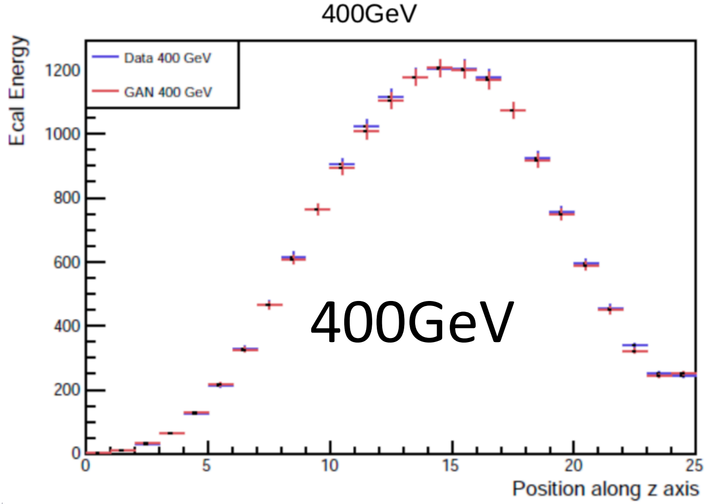
We run on Caltech ibanks GPU cluster thanks to Prof M. Spiropulu

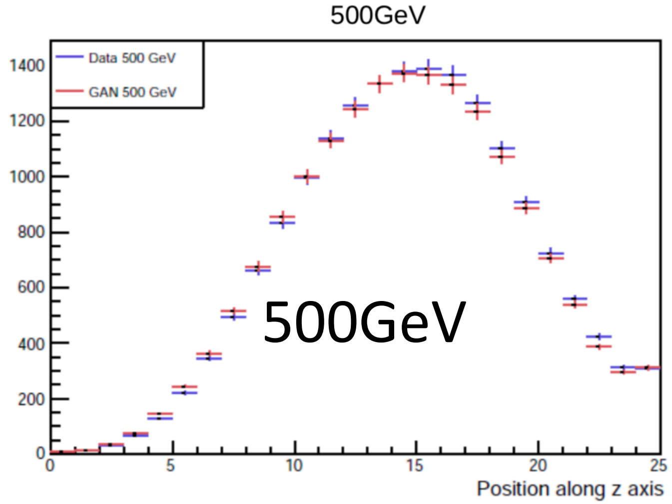
Shower shapes vs primary energy

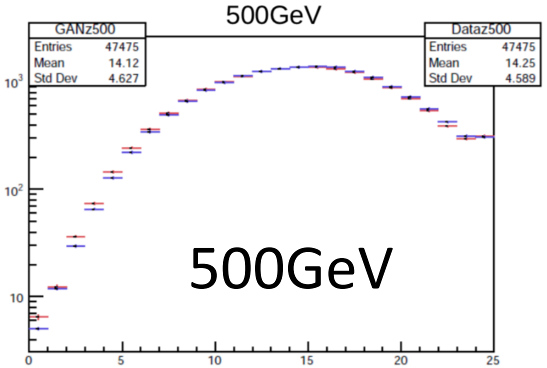




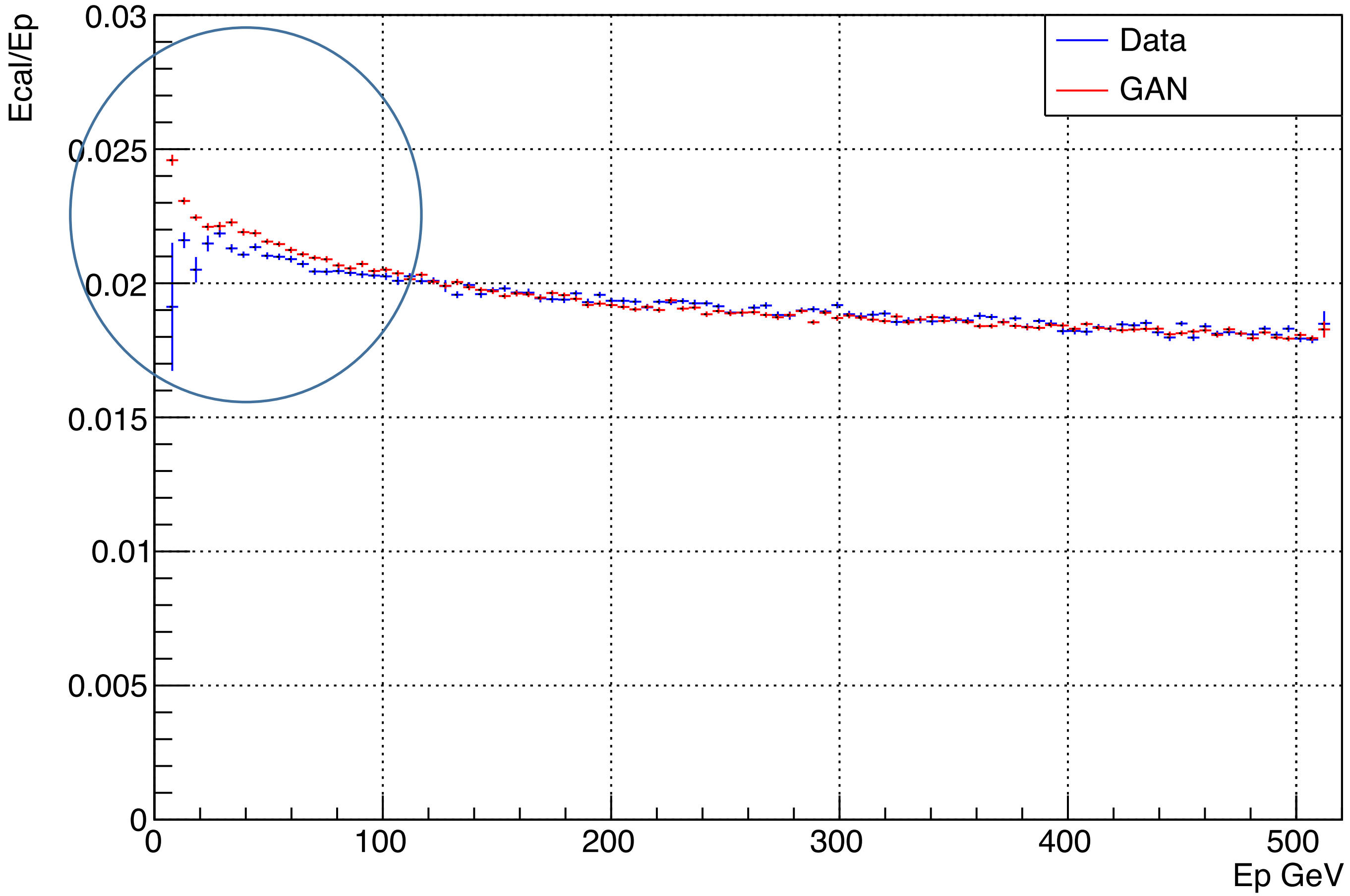




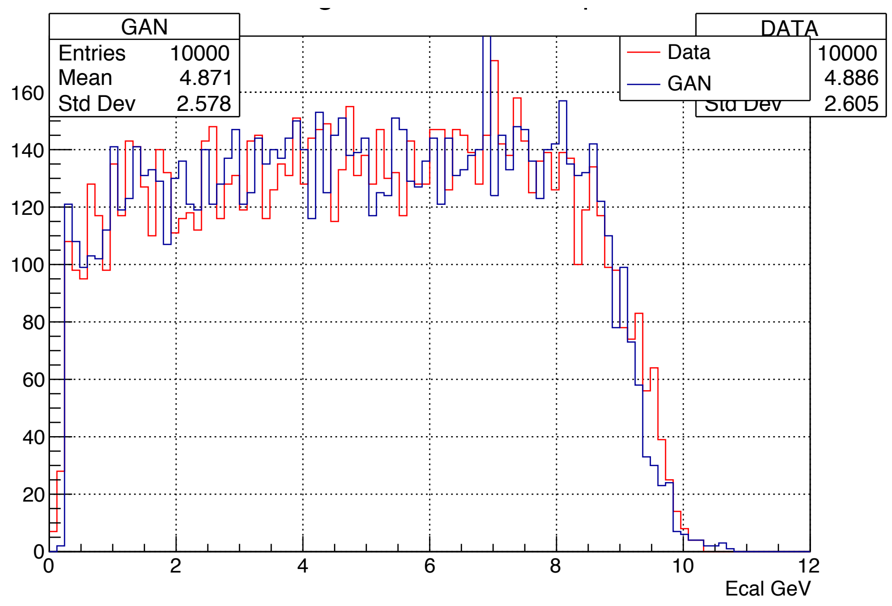




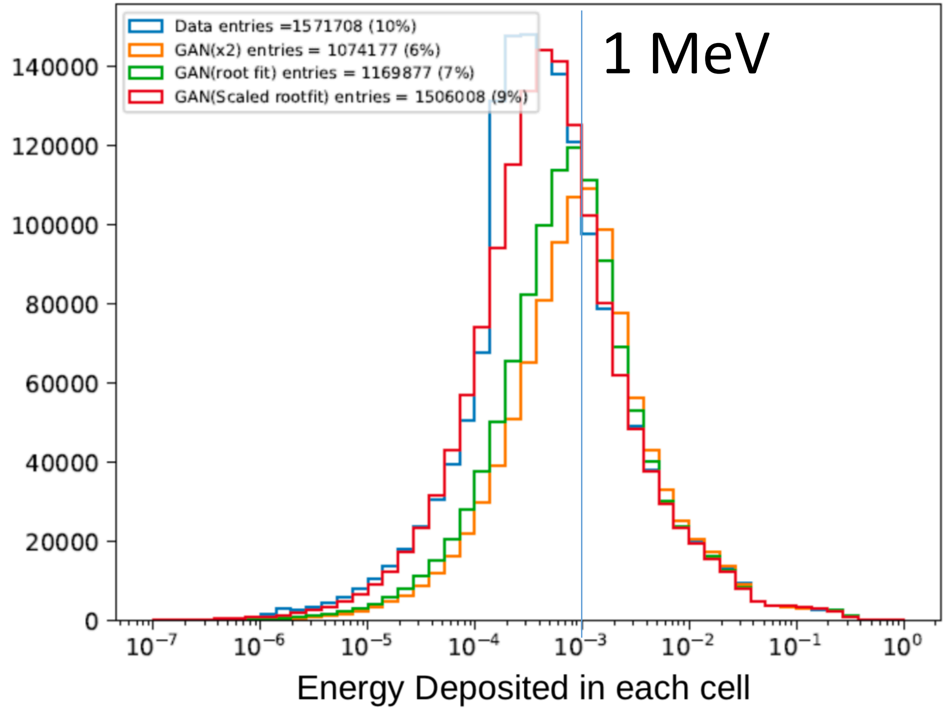
Calorimeter sampling fraction



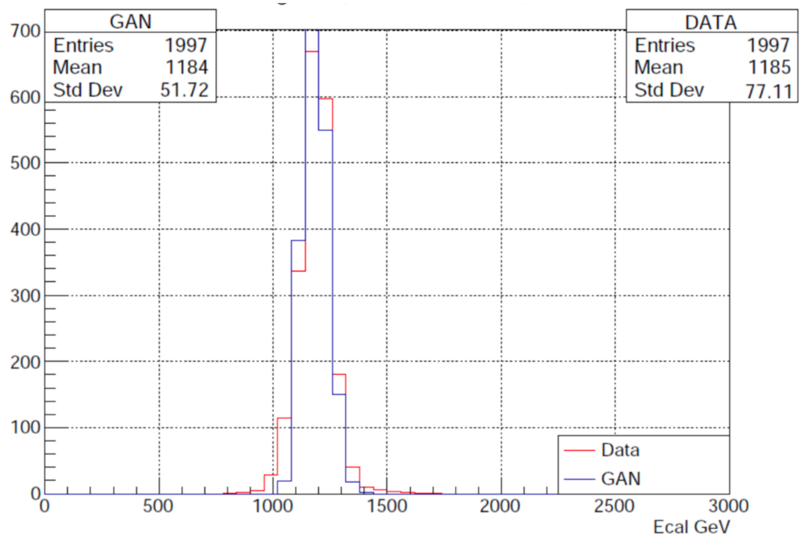
Total deposited energy



Low energy performance & single cells



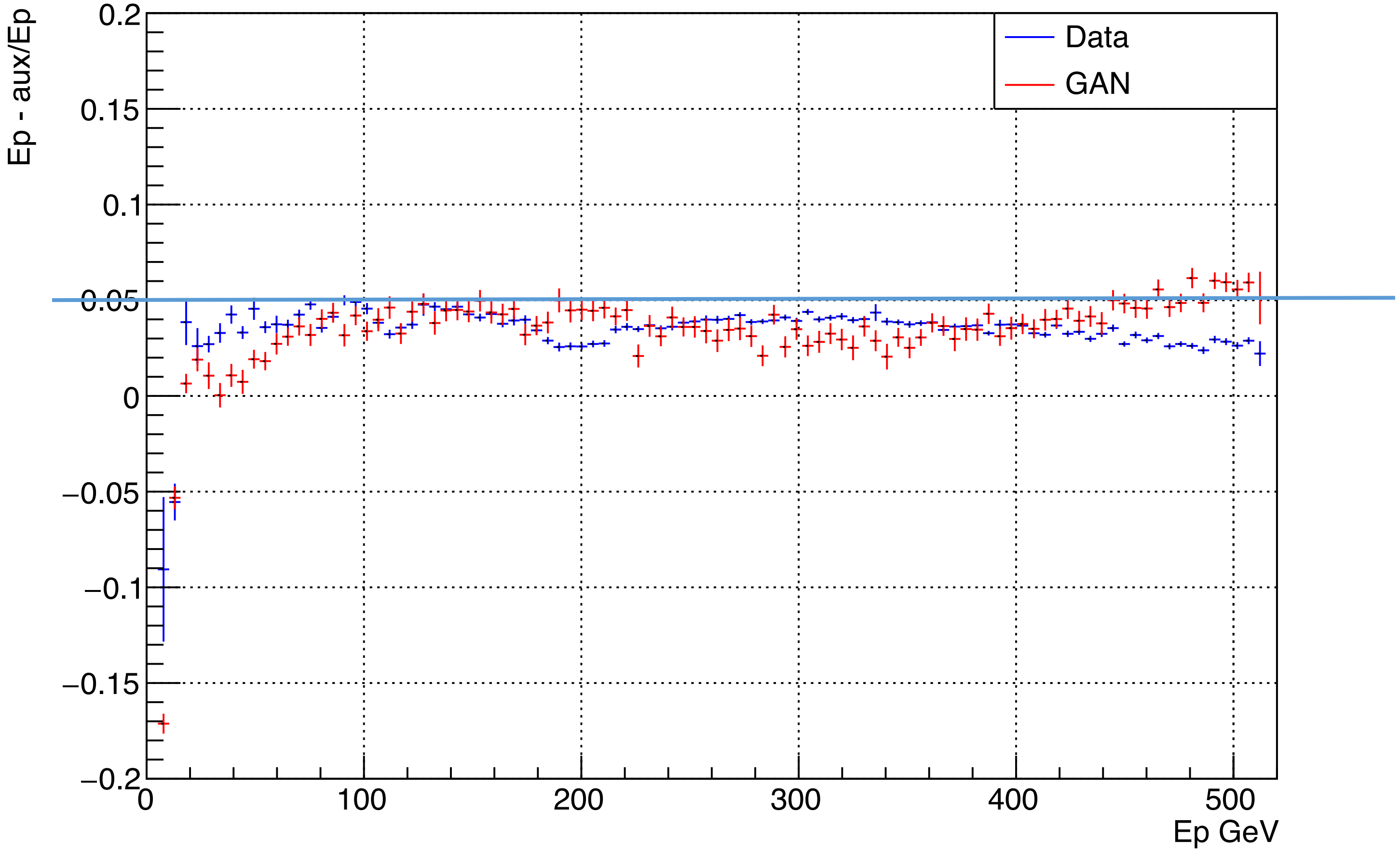
Number of hits (above 200 keV)



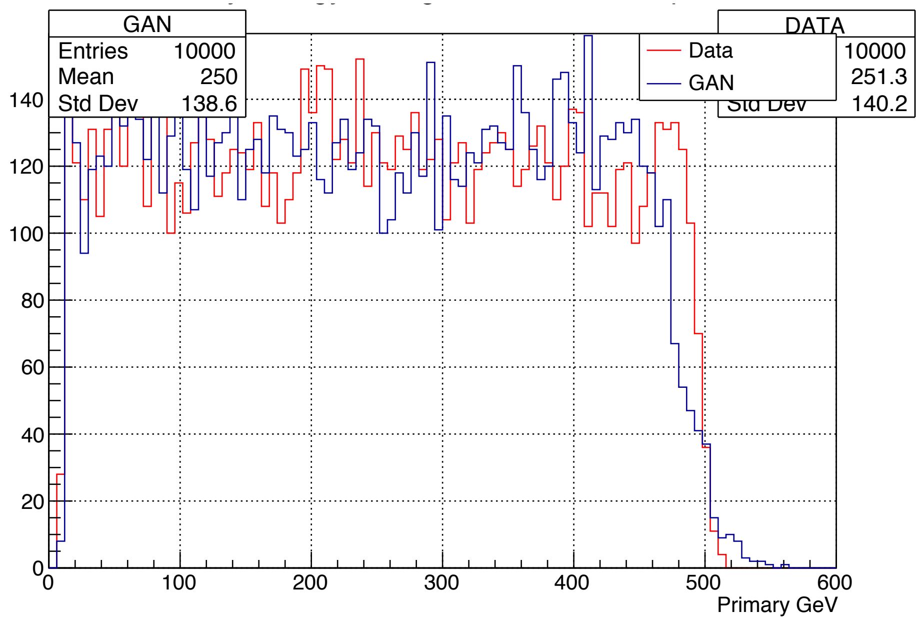
Several pre-processing optimisation steps improved performance at low energy

More details in [G. Khattak talk at IML workshop]

Discriminator regression on input energy

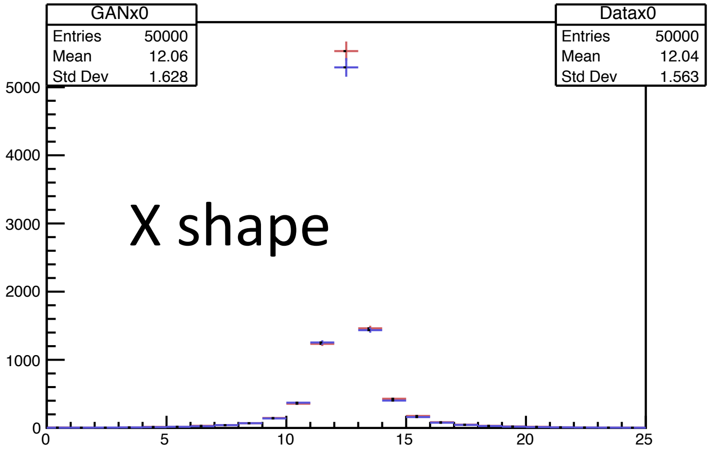


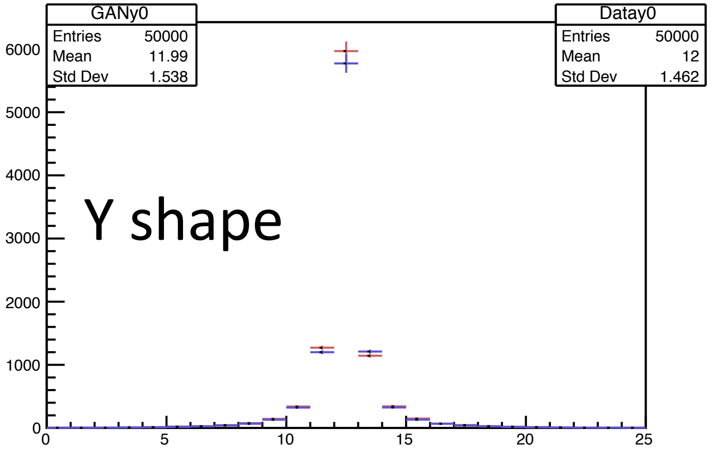
5% error on auxiliary energy regression

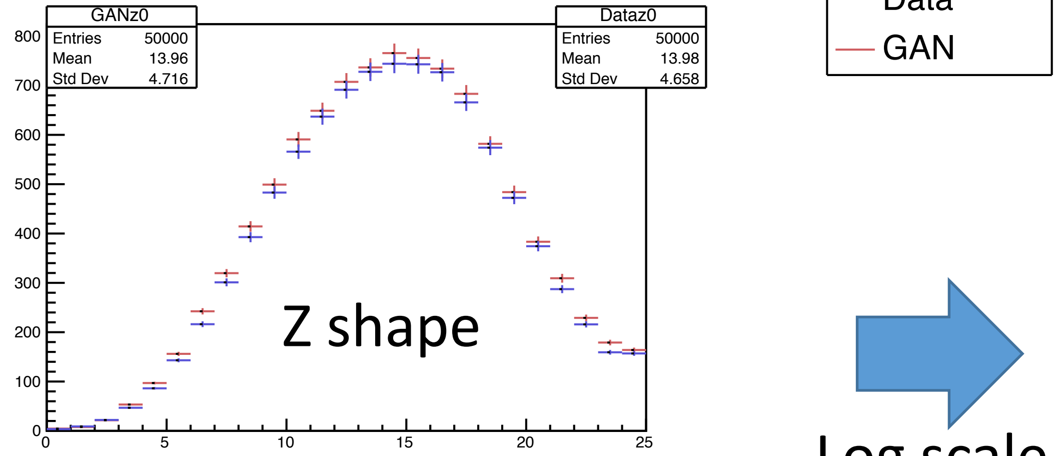


Pions

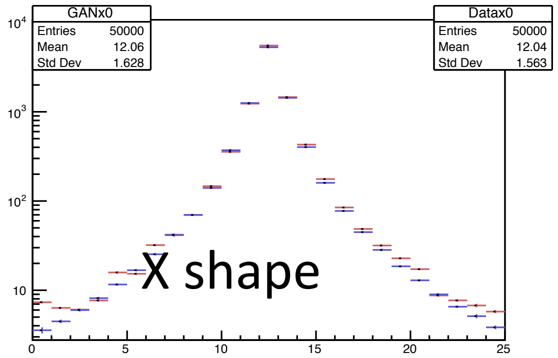
10-500 GeV

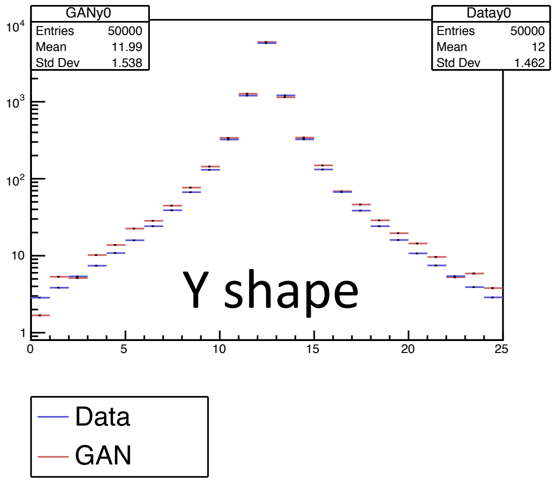


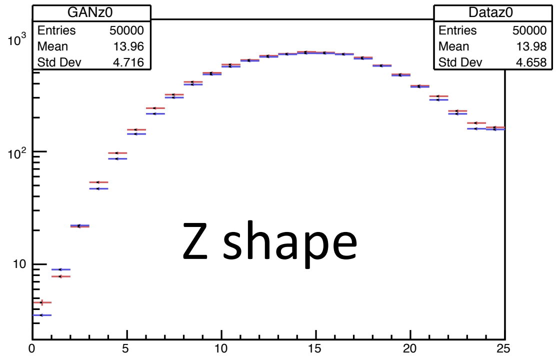




Log scale

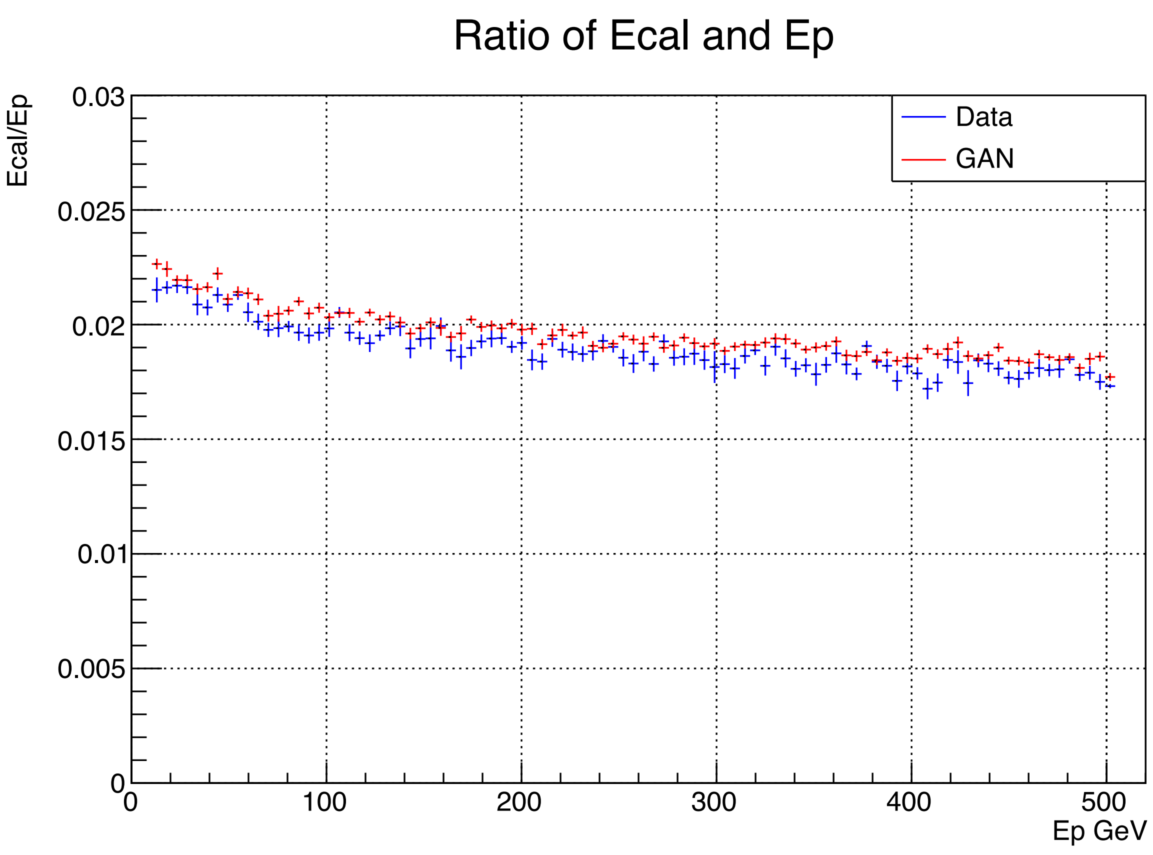


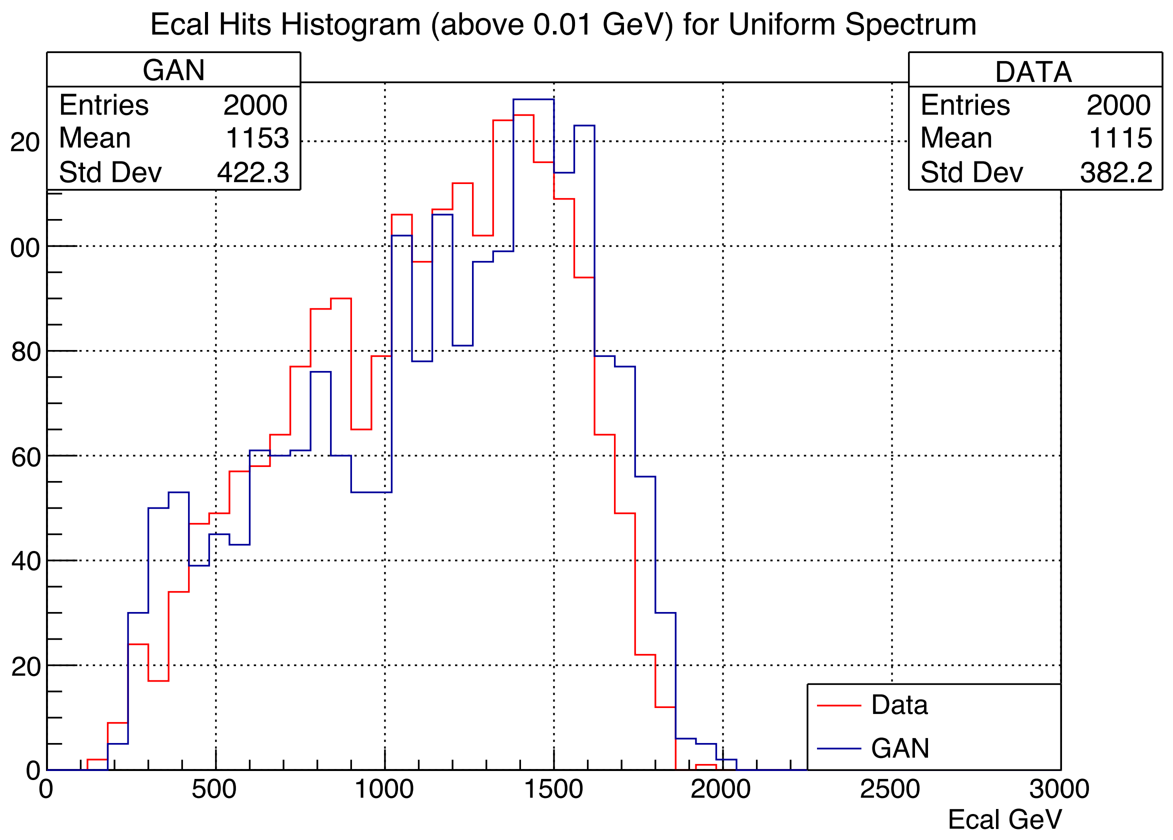




Deposited energy

10-500 GeV -Pions





GAN seems to overestimate slightly energy deposits

CERN openlab

Generalisation & Computing resources

Hyper-parameter scans
Distributed training

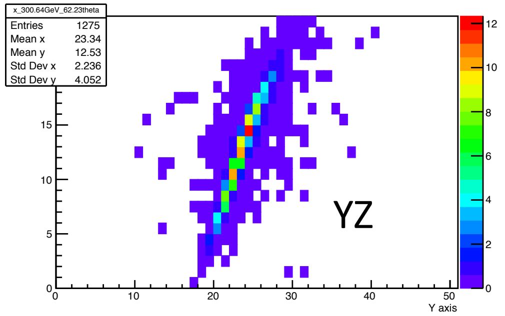
Generalisation

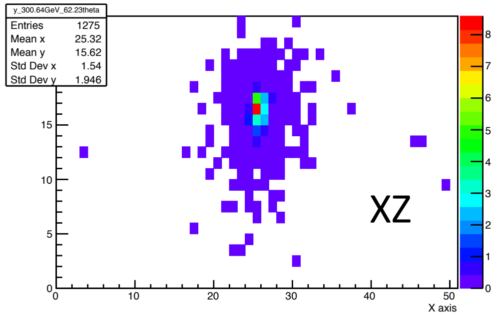
Variable angle sample

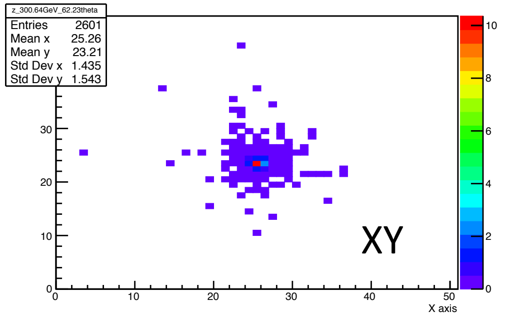
Electrons enter the calorimeter with a 60°-120° angle range

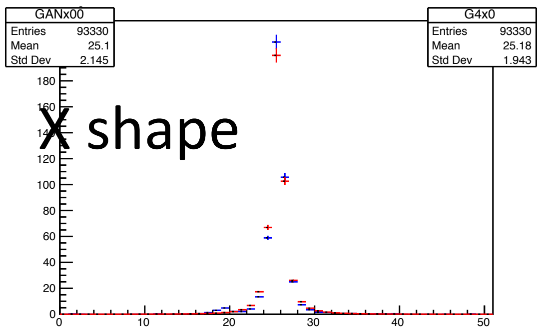
Wider/asymmetric image size (51x51x25):

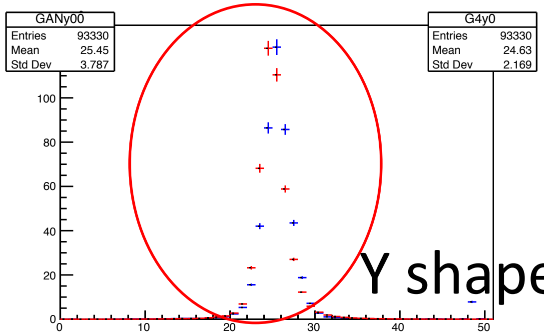
Minimal architecture changes

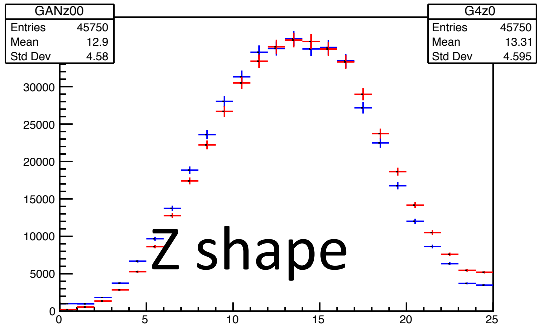








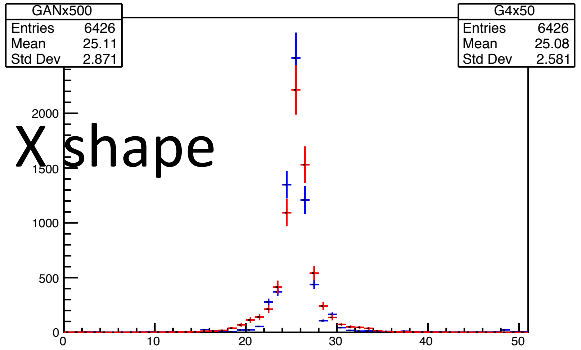


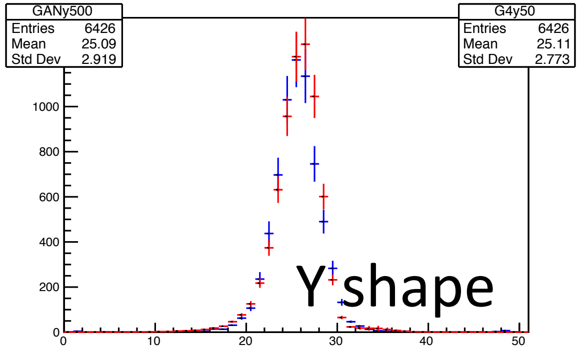


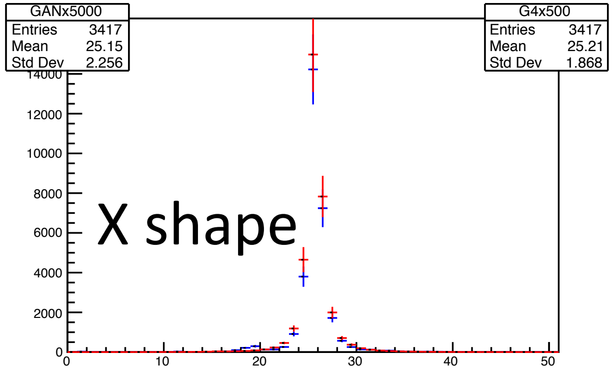
Energy inclusive shower shapes (60° angle) OK

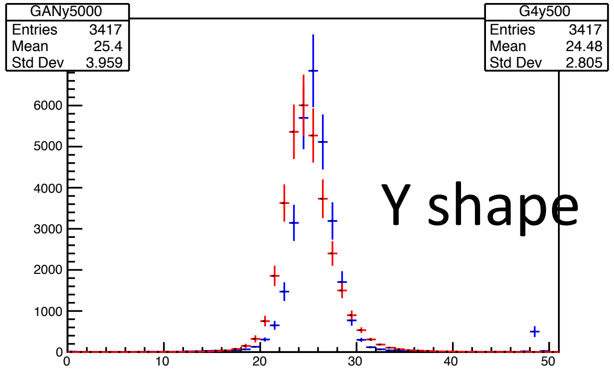
Preliminary

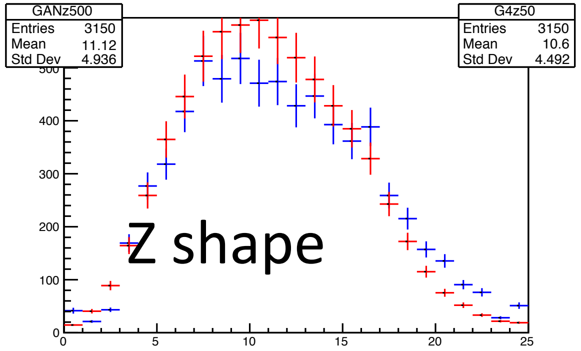
Variable angle

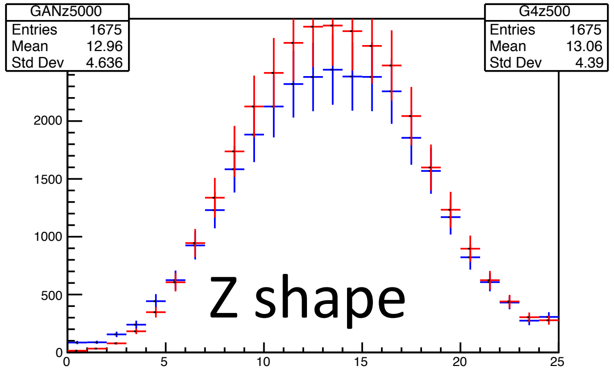












Energy dependency is wrong

Need to adjust loss

Need to adjust convolution parameters: hyperparameter scan

Distributed training Is needed!

Computing performance

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**	**7**
3d GAN (batchsize 128)	GeForce GTX 1080	0.04

Inference:

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

➔ speedup factor > 2500!!

Training:

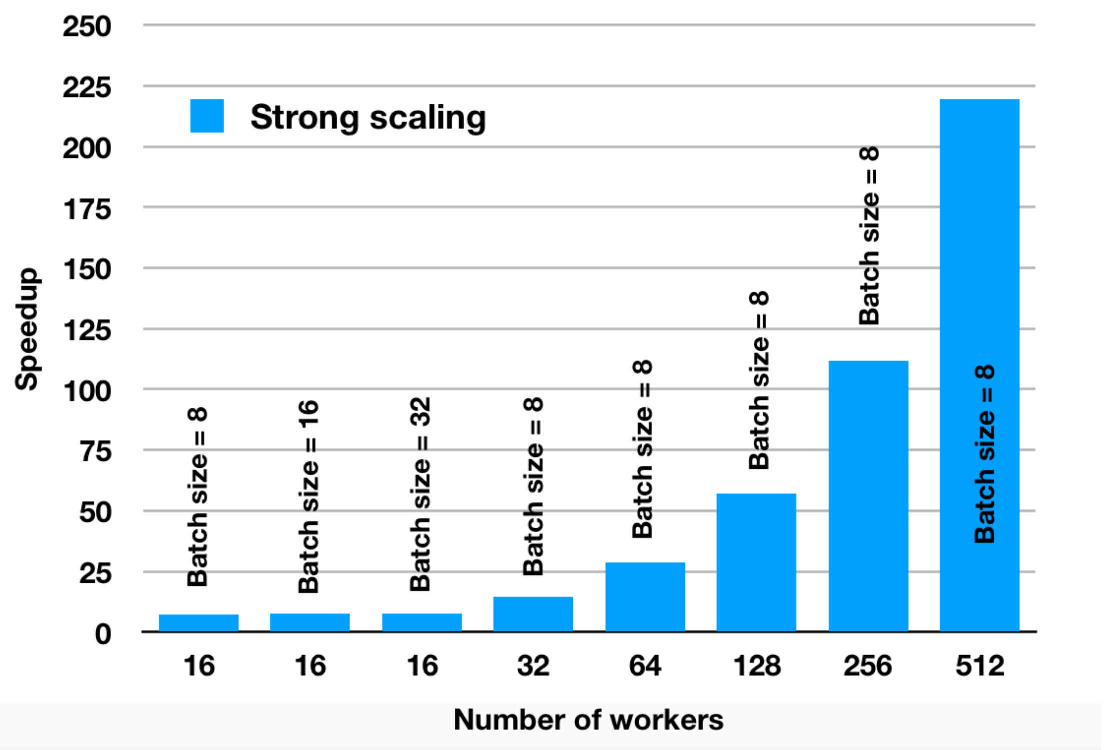
45 min /epoch on Tesla P100 (HNSciCloud test)

Introduce **data parallel** training based on MPI

Test several libraries

Horovod

Collaboration with SURFsara to test Horovod performance (based on Baidu-Ring-AllReduce approach) @TACC's Stampede



Xeon Scalable Processors
Platinum 8160
2 MPI processes per node

Collaboration with D. Moise , Cray inc.
Submitted to SuperComputing 2018

Cray ML plugin

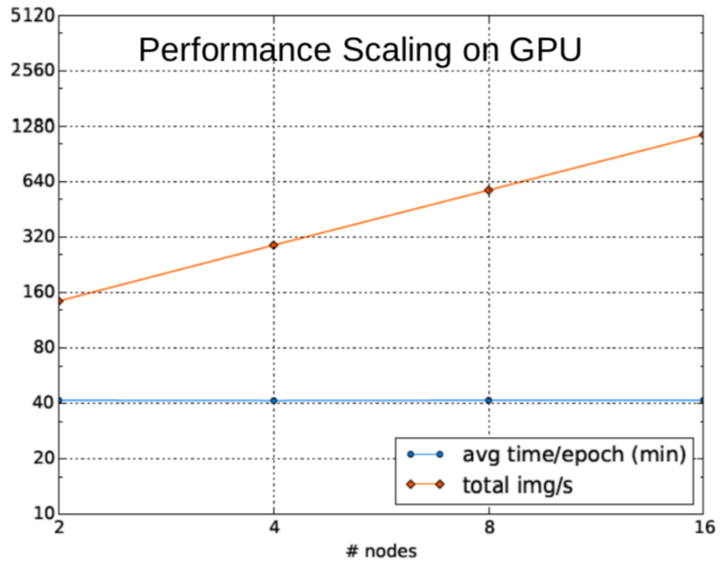
Optimal scaling through a large number of nodes

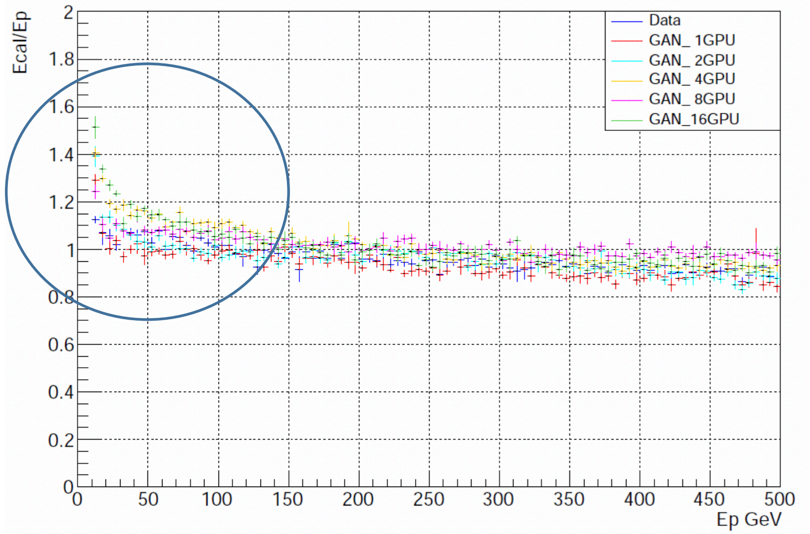
Observed performance degradation at low energy

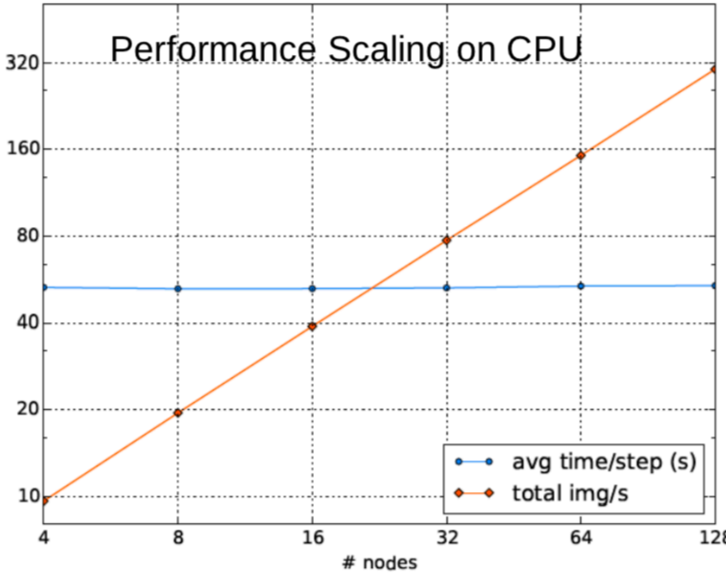
Increase in “effective” batch size?

Possibly compensate by increasing learning rate

Work in progress…





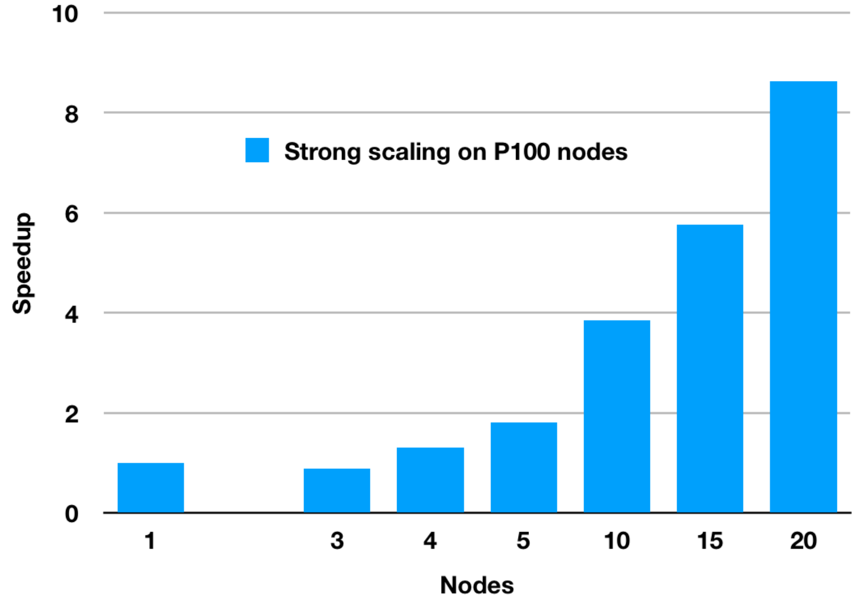


	GPU System	CPU System
Model	XC40/XC50	XC50
Computer nodes	Intel Xeon E5-2697 v4 @ 2.3GHz (18 cores, 64GB RAM) and NVIDIA Tesla P100 16GB	Two Intel Xeon Platinum 8160 @ 2.1GHz (2 x 24 cores, 192GB RAM)
Interconnect	Aries, Dragonfly network topology	Aries, Dragonfly network topology
Step	Epoch	Batch

Elastic Average SGD

mpi-learn

Collaboration with J.R, Vlimant, Caltech.
Submitted to International SuperComputing 2018



Modify mpi-learn library

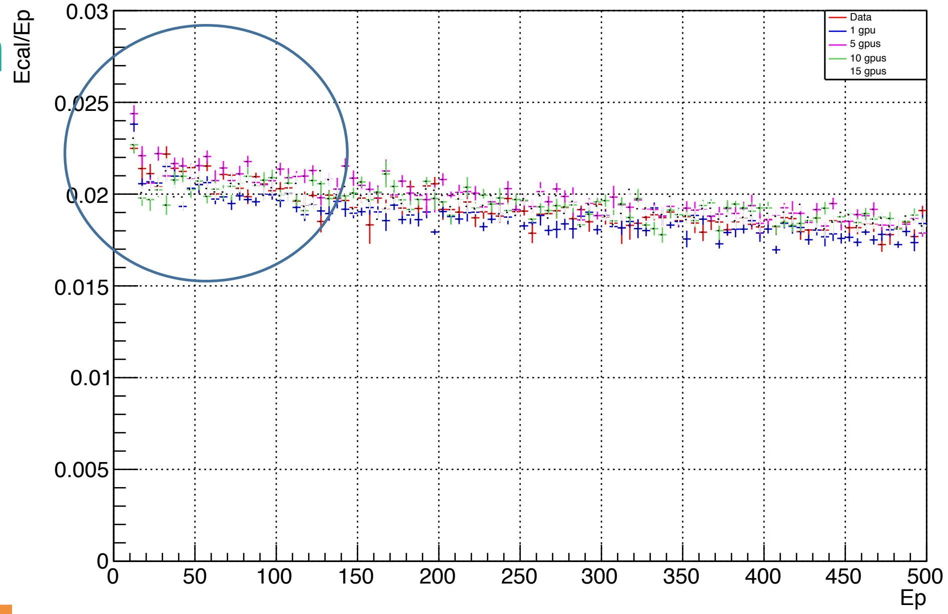
Test on 20 GPU (Nvidia P100) at CSCS

Good scaling

No performance degradation at low energy!

Skopt integration to run Hyper-Parameter scan in parallel (work by JR. Vlimant, T. Nguyen, V. Loncar)

Work in progress…



CERN openlab

https://github.com/duanders/mpi_learn

Plan

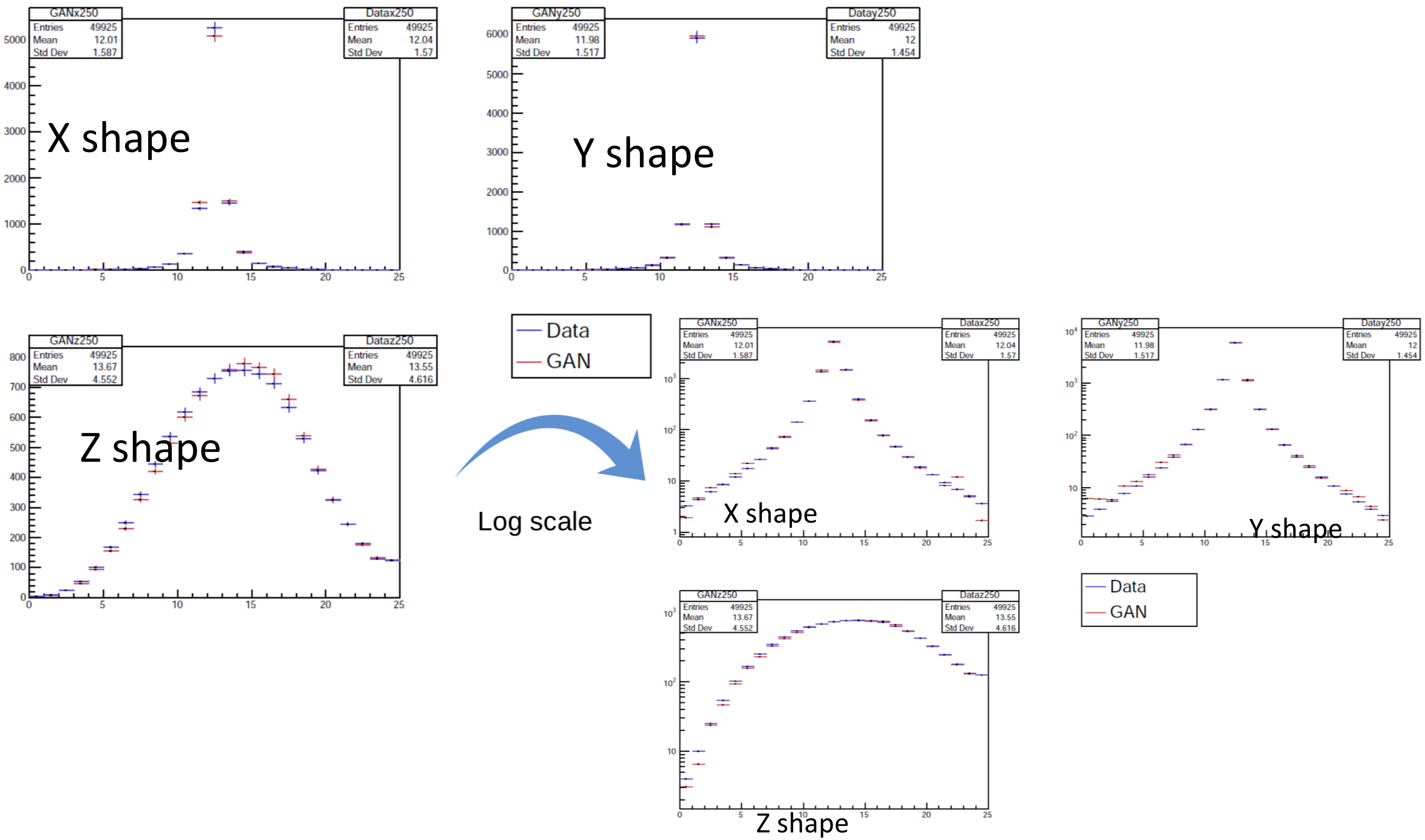
- Prove we can generalise this network to other calorimeters
    - Improve variable angle results
    - Test different geometries
- Run hyper-parameters scans
- Efficient training is a priority
- Computing performance optimisation
- Test different environments:
    - HPC (GPU, CPU?), cloud (CERN GPU benchmark on HNSciCloud project)
    - Big Data approach integration

Questions?

Thanks

Shower shapes

250 GeV electron



Shower shape moments: width

250 GeV electron

Central values are consistent

Stdev still slightly off