3d GAN for fast simulation

Sofia Vallecorsa

IML meeting – November 2017

Outline

- Introduction
- 3d GAN model
- Results
- A generic framework for fast simulation
- Summary & Plans

Introduction

- Detailed simulation has heavy computation requirements
- Calorimeters are among the most time consuming part
- To improve computing performance
  - Geant4 provides event level parallelism
  - GeantV introduces fine grained parallelism and smart memory management
  - GeantV targets x5 speedup → Not enough for HL-LHC

Improved, efficient and accurate fast simulation

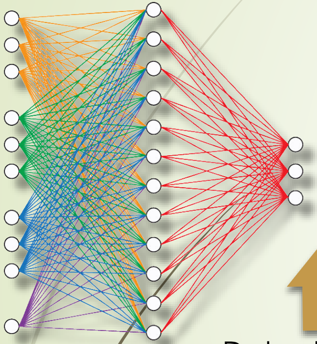
- Currently available solutions are detector dependent

Deep Learning for fast sim

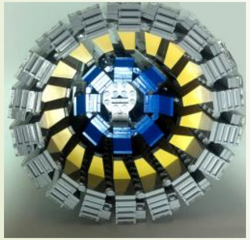
- Initially focus on time consuming detectors
  - Reproduce particle showers in high granularity calorimeters
  - CLIC EM calorimeter
- Train networks on full simulation
  - Next test possibility of training on real data
- Test different models
  - **Generative Adversarial Networks**, Recurrent networks,…
- Embed inference (and training) step in GEANTV
  - Provide a configurable interface integrated new basket flow

Deep Learning engine

Untrained Model



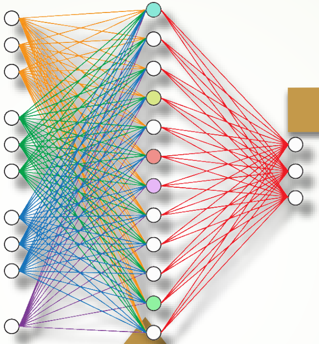
Detector Geometry

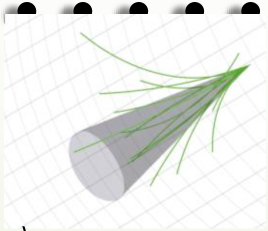


http://www.physics.umd.edu/rgroups/hep/LegoCMS/

GeantV

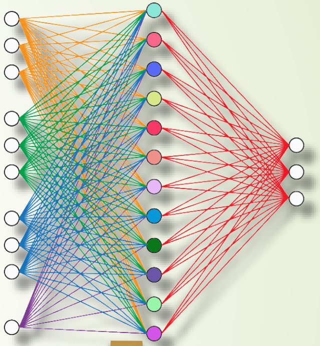
Training

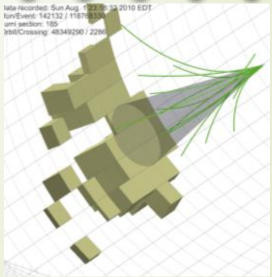




Physics (e^+, e^-, γ, π..)
Kinematics…

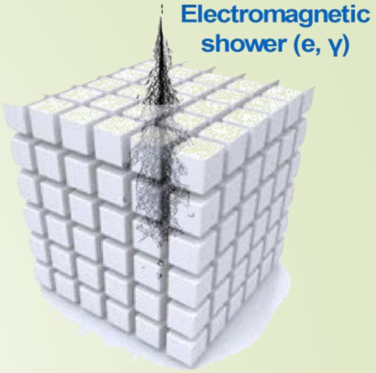
Trained Model



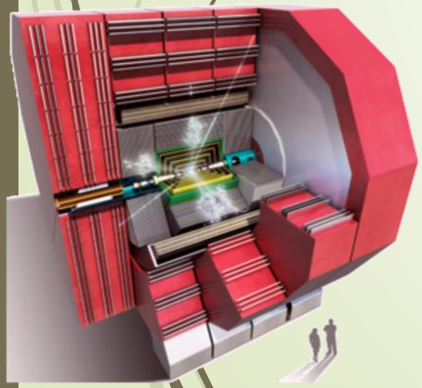


http://www.quantumdiaries.org/wp-content/uploads/2011/06/JetConeWithTracksAndECAL.png

CLIC calorimeter



- CLIC is a CERN project for a linear accelerator of electrons and positrons to TeV energies
- Associated calorimeter detector design(*)
- An array of absorber material and silicon sensors
  - ECAL (1.5 m inner radius, 5 mm×5 mm segmentation): 25 tungsten absorber layers + silicon sensors
  - HCAL (3.0 cm×3.0 cm segmentation): 60 steel absorber layers + polystyrene scintillators

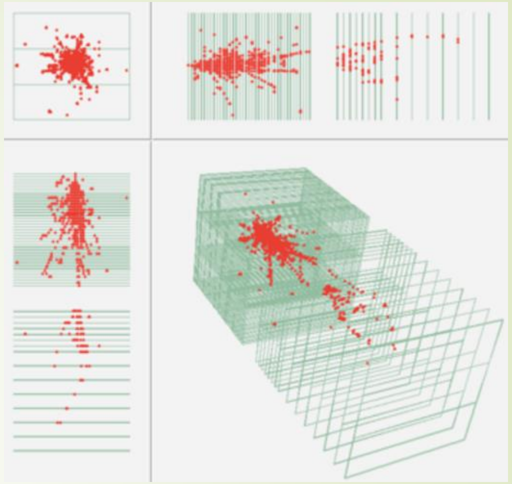


Data released within CERN OpenData initiative

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

CLIC calorimeter data

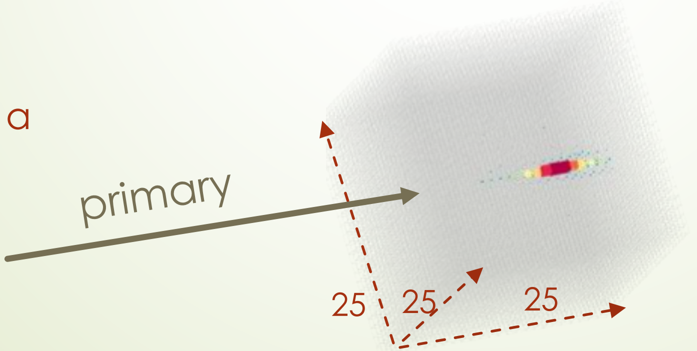
Geant4 shower

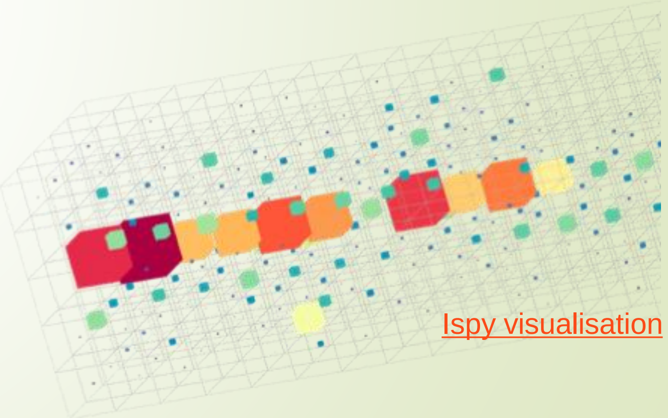


Geant4 single-particle benchmark datasets (e^+, e^-, γ, π) (DD4hep/DDG4/ddsim)

- Uniform energy distribution
- Fixed energy points (10,50,100,200 GeV)
- Simplified: no clustering/clustering id algorithms applied

Data is essentially a 3D image

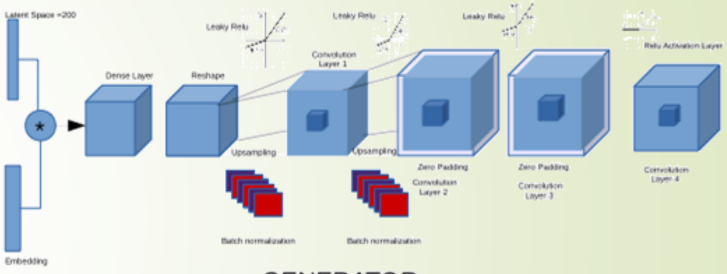




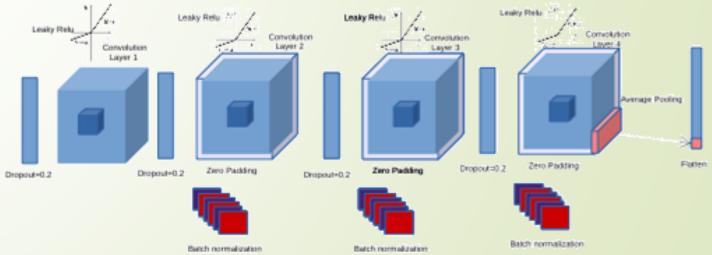
Ispy visualisation

3d GAN for calorimeter images

- Based on convolution/deconvolutions
  - 3D (de)convolutions to describe full shower development
  - Particle tag as auxiliary classifier
- Implemented tips&tricks found in literature
  - Some helpful (no batch normalisation in the last step, LeakyRelu, no hidden dense layers, no pooling layers)
  - Some not (Adam optimiser)
- Batch training
- Loss is combined cross entropy

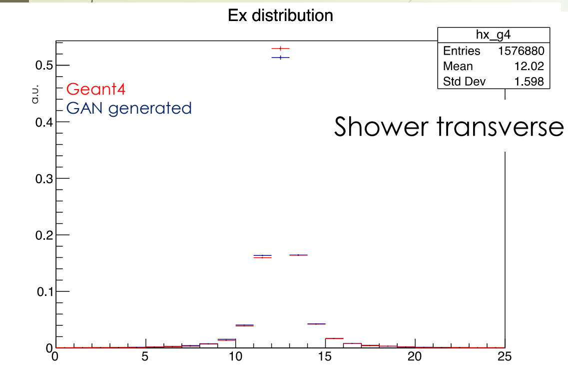


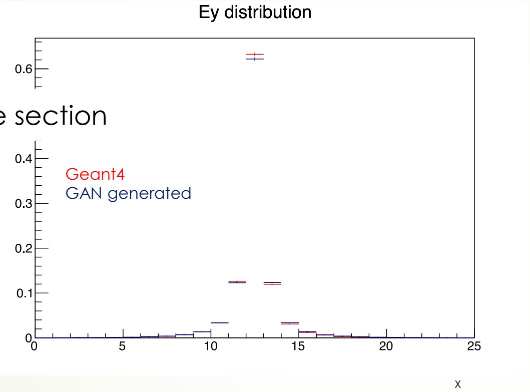
GENERATOR



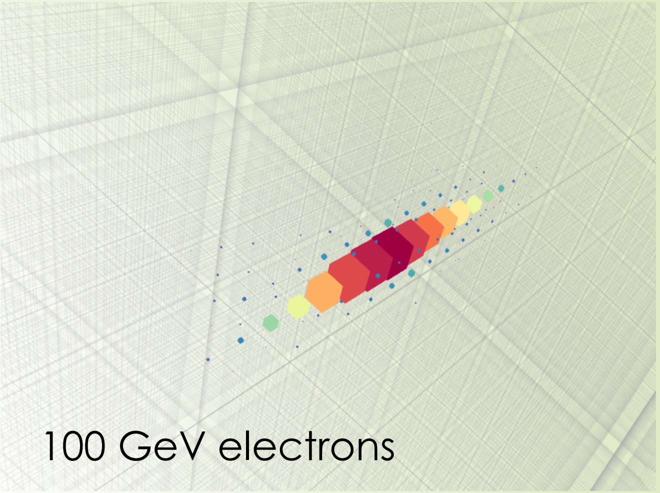
DISCRIMINATOR

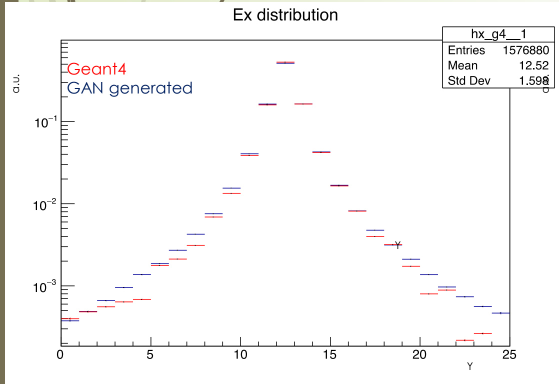
GAN generated electrons

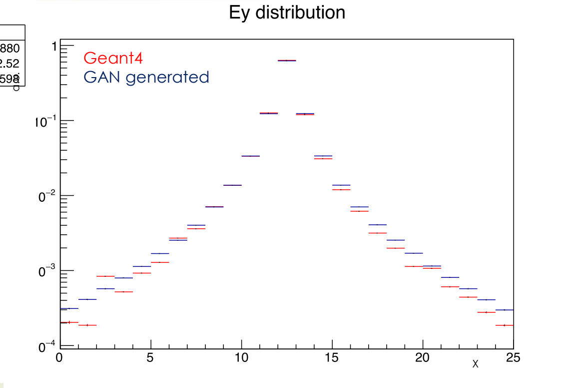


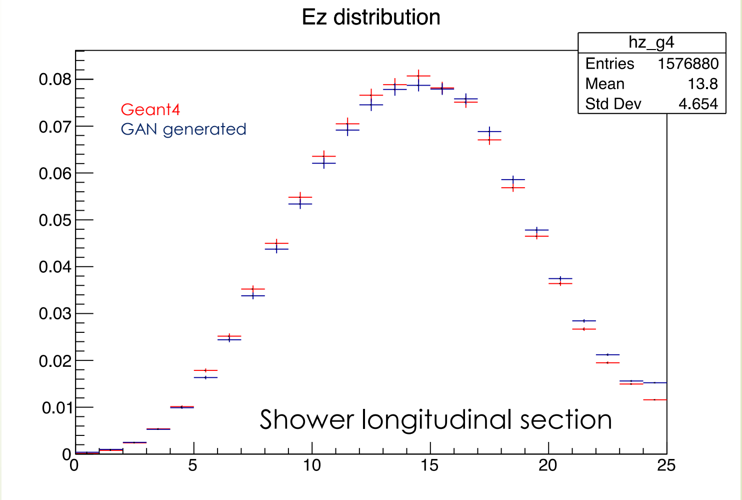


Shower transverse section





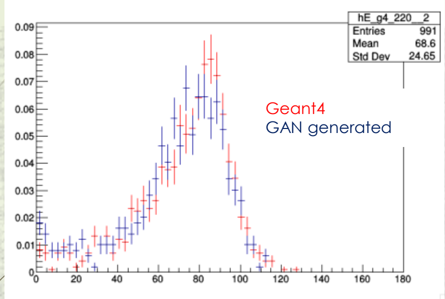


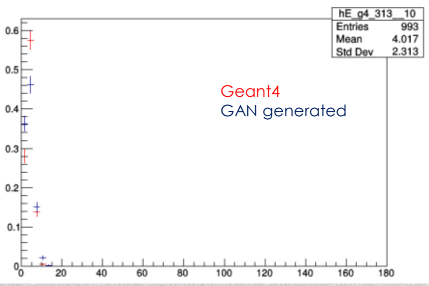


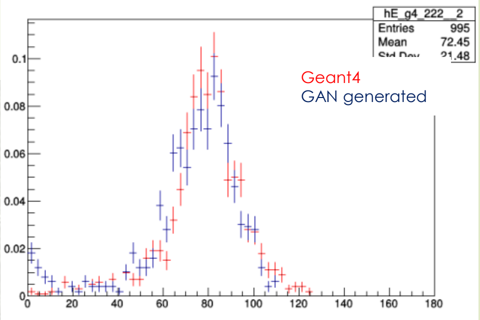
Shower longitudinal section

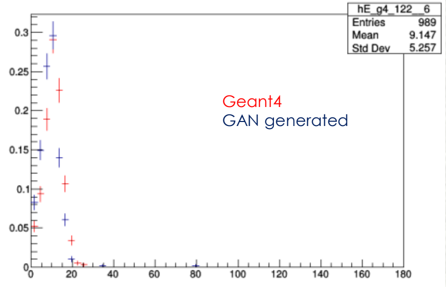
Preliminary

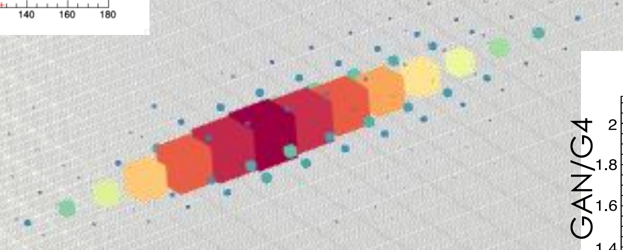
Single cell response

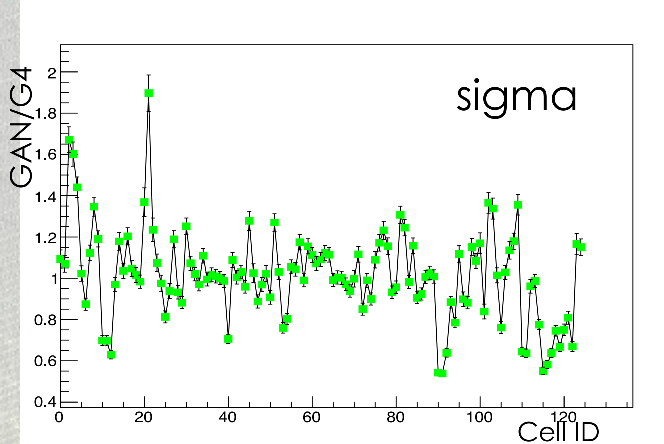








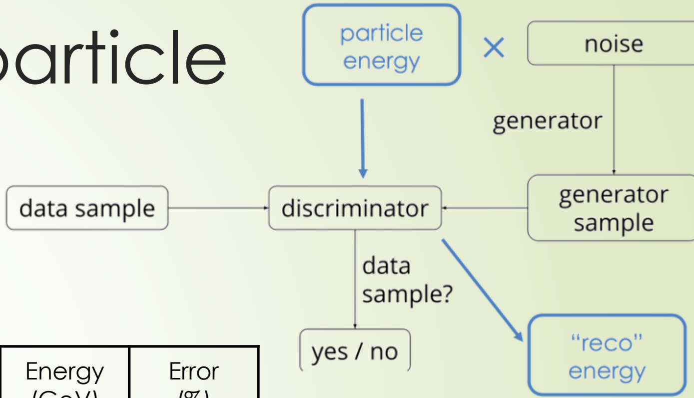




Single cell response is not perfect

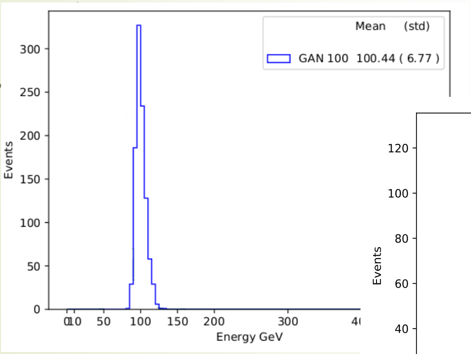
Set up higher level criteria for image validation (reconstructed variables)

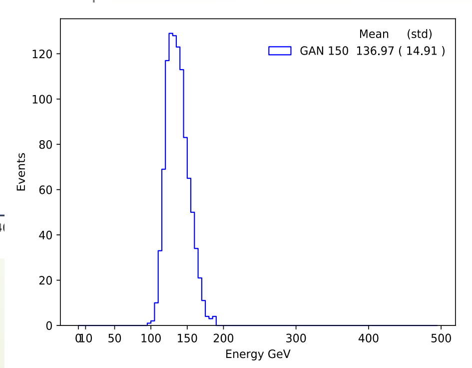
Conditioning on primary particle energy



similar to Auxiliary Classifer GAN

- Add energy information to latent space
- Add a regression task to the discriminator





Energy (GeV)	Error (%)
100	5
150	13
200	10
300	6
400	10
500	15

The idea is to generalize to multi-class approach (multi-discriminator approach): primary particle entry point, angle, etc..)

Computing resources

		Time/Shower (msec)
Full Simulation (G4)	Intel Xeon E5	56000
3d GAN (batchsize 128)	Intel i7 (laptop)	66
	GeForce GTX 1080	0.04

- Generating a shower with a trained model is orders of magnitude faster than using Geant4
- Using DL techniques for fast simulation is profitable if training time is not a bottleneck (~1 day on NVIDIA GTX 1080)
  - Depending on the final use case retraining the networks might be necessary
- Test different hardware & multi-node scaling
- We want to provide a generic, fully configurable tool
  - Optimal network design depends on the problem to solve
  - Need embedded algorithms to perform hyper-parameters tuning and meta-optimization

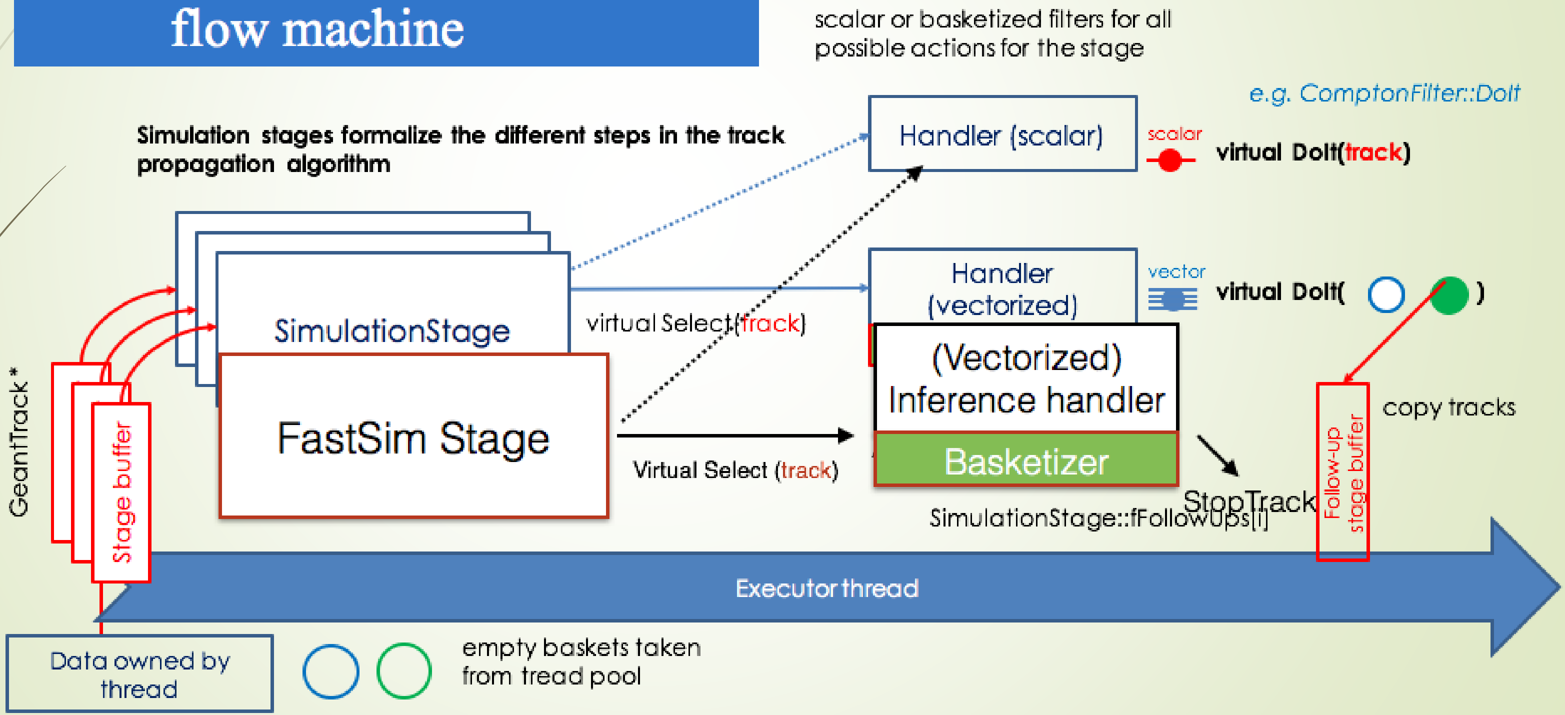
Summary

- A fully configurable framework for fast simulation
  - Easily embed user code
  - A reasonable library of algorithms
- A DL based algorithm
  - Evaluation of several models and optimisations
  - Standalone training tool
- Integration in GEANTV: inference step first, then training tool
- Prototype interface and ML proof of concept in GEANTV beta
  - ML model design and meta-optimisation later on
  - It could also be back-ported to Geant4

Thanks

GeantV flow

V3: A generic vector flow machine



Machine Learning engine