A Deep Learning tool for fast simulation

Sofia Vallecorsa[1] for the GeantV Project

[1]Gangneung Wonju U. & CERN, Sofia.Vallecorsa@cern.ch

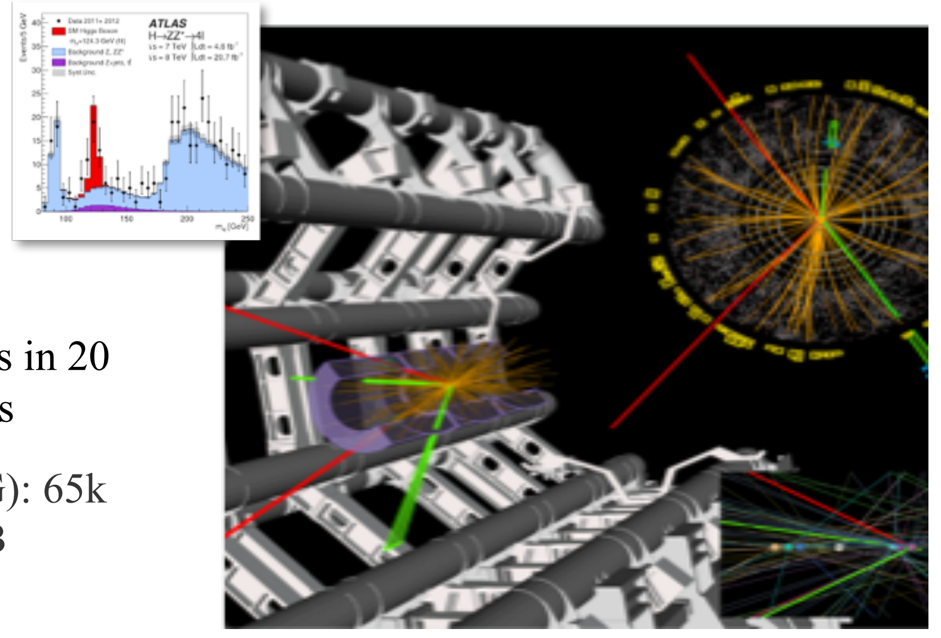
The project

Introduction and Motivation

Simulation of particle transport through matter is fundamental for understanding the physics of **High Energy Physics** (HEP) experiments, as the ones at the Large Hadron Collider (**LHC**) at CERN. Such experiments have dedicated so far most of their worldwide distributed CPU budget – in the range of half a million CPU-years equivalent – to simulation. In particular, the most computing-intensive components are **geometry** modeling, **navigation** through millions of objects and **physics** models.

200 Computing centers in 20 countries: > 600k cores

@CERN (20% WLCG): 65k processor cores ; 30PB

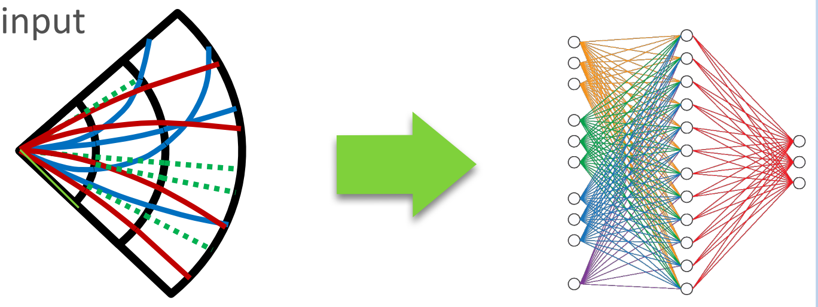


Deep Learning approach

A faster approach is to treat traditional simulation as a black-box and replace it by a **deep learning** algorithm trained on different particle quantities. We are testing several techniques such as generative adversarial networks (GANs) to replace the Monte Carlo approach. We expect to achieve a **significant speedup** (x25) with respect to GeantV full simulation approach. Development of such tool can further benefit other fields, such as radiation protection, environmental modeling and medicine.

Generative models (Generative Stochastic Networks, Variational Auto-Enconders, Generative Adversarial Networks, ..) can be used for simulation

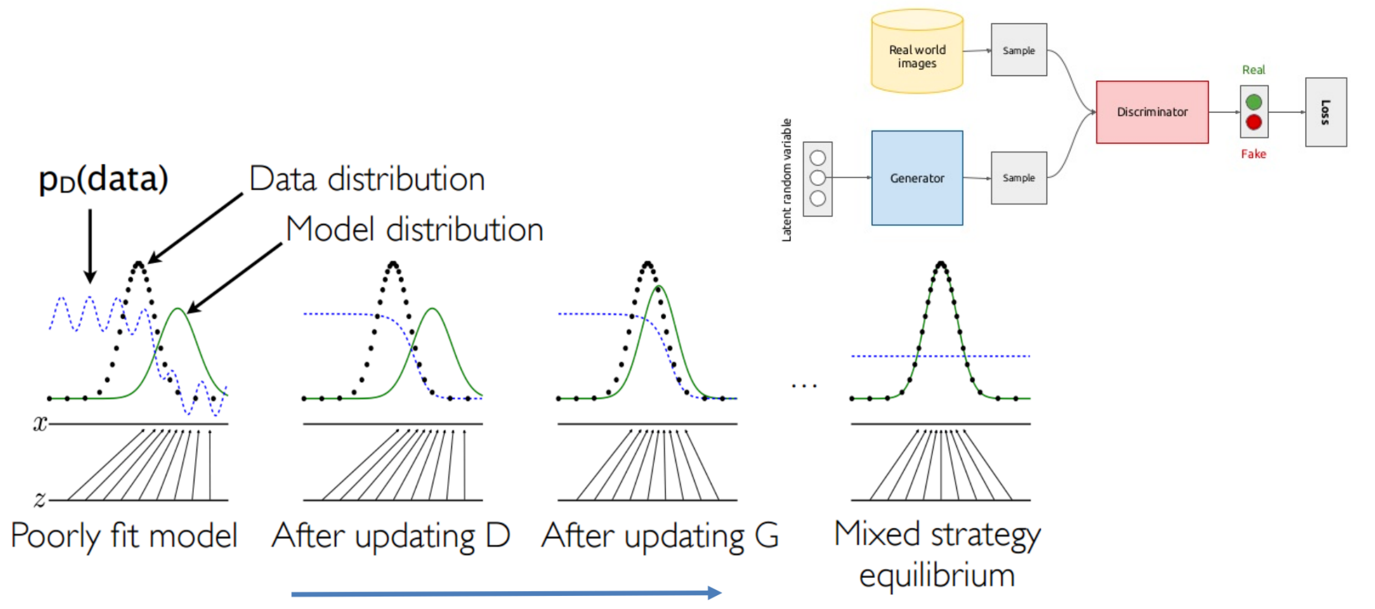
- Realistic generation of samples
- Use complicated probability distributions
- Optimize multiple output for a single input
- Can do interpolation
- Work well with missing data



Generative adversarial networks

Simultaneously train two models:

- G(z) captures the data distribution
- D(x) estimates the probability that a sample came from the training data rather than G
- Training procedure for G(z) is to maximize the probability of D(x) making a mistake



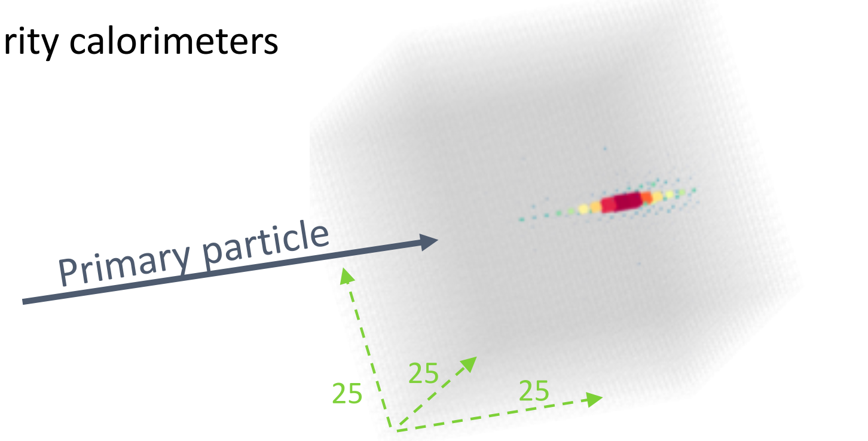
Achievements

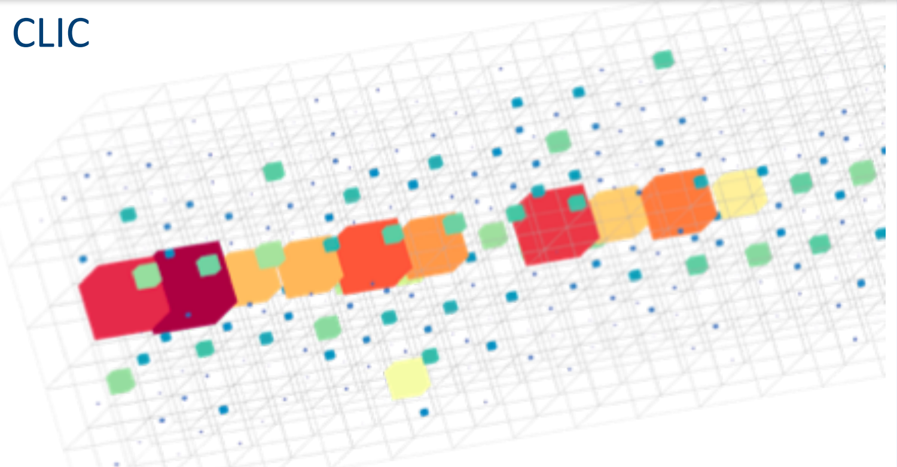
VGAN for particle detector

Single particle energy deposits in the CLIC calorimeters (http://cds.cern.ch/record/2254048#)

Start with the most time consuming detectors: high granularity calorimeters

Data is essentially a 3D image



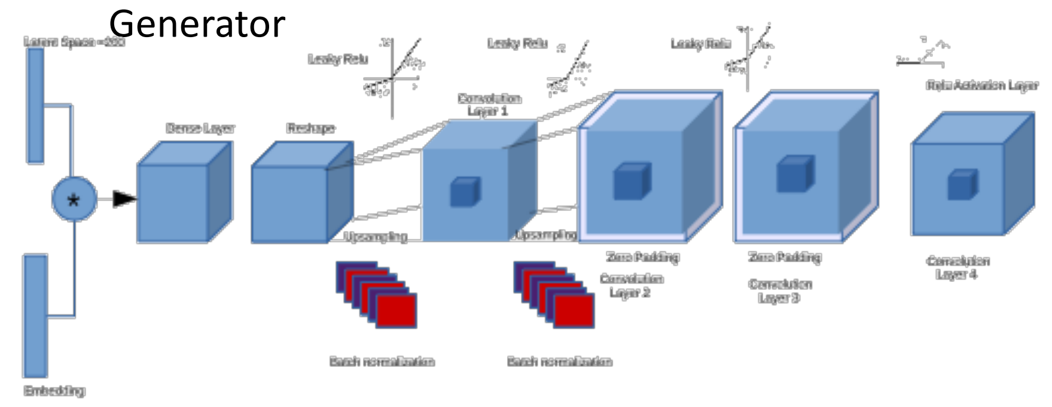


Geant4 π shower in LCD calorimeter

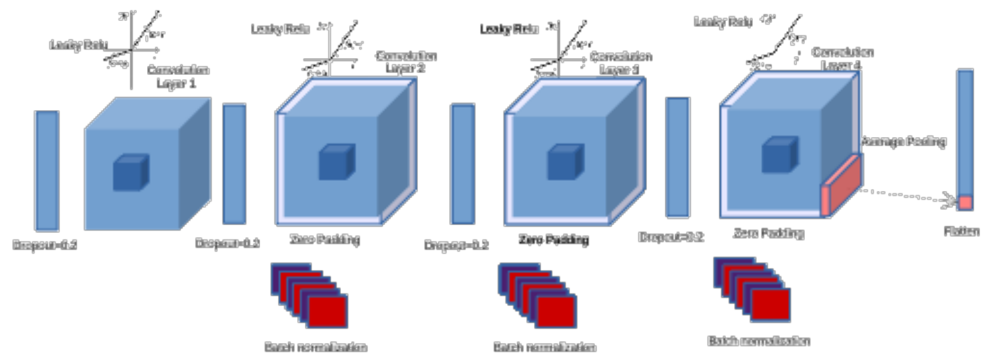
Generator and Discriminator based on 3D convolutions

Explored several “tips&tricks” : No batch normalisation in the last step, LeakyRelu, no hidden dense layers 😀, Adam optimiser ☹

Generator



Discriminator



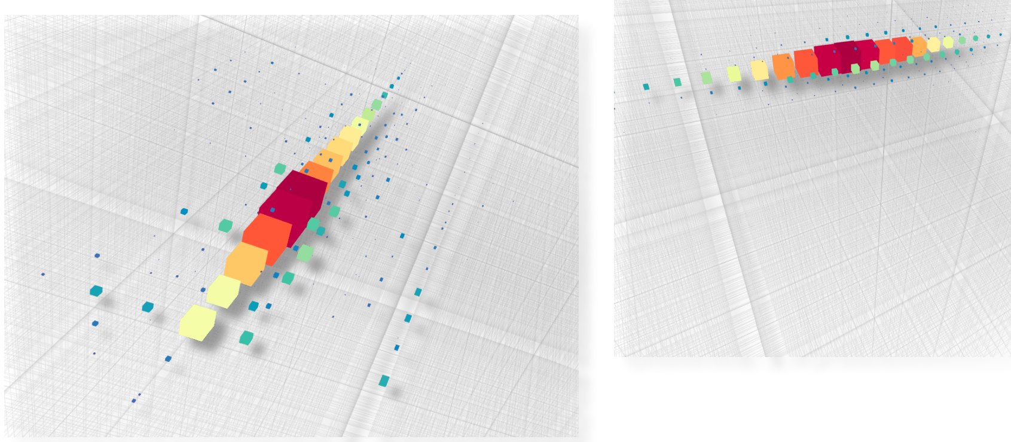
Generated electromagnetic showers

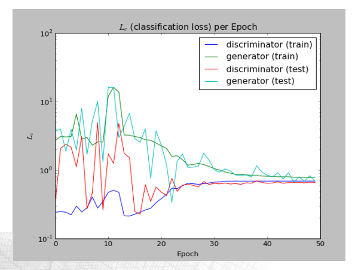
GAN generated electron (energy: 100 GeV)

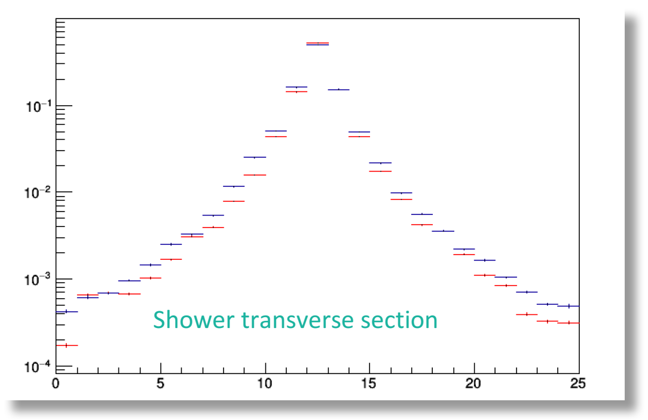
One of the first VGAN implementations (Keras + Tensorflow backend)

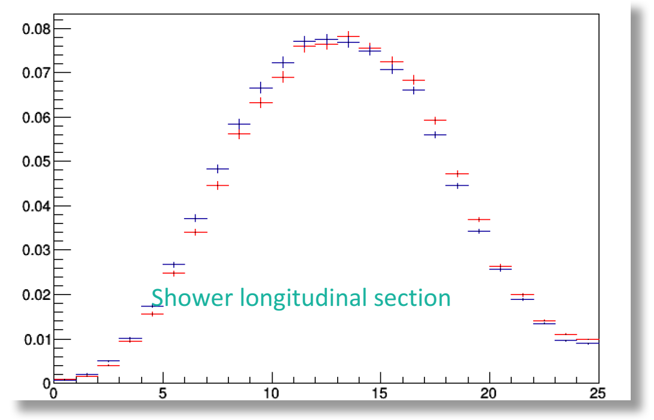
First results look very promising!

Qualitative analysis show no collapse problem







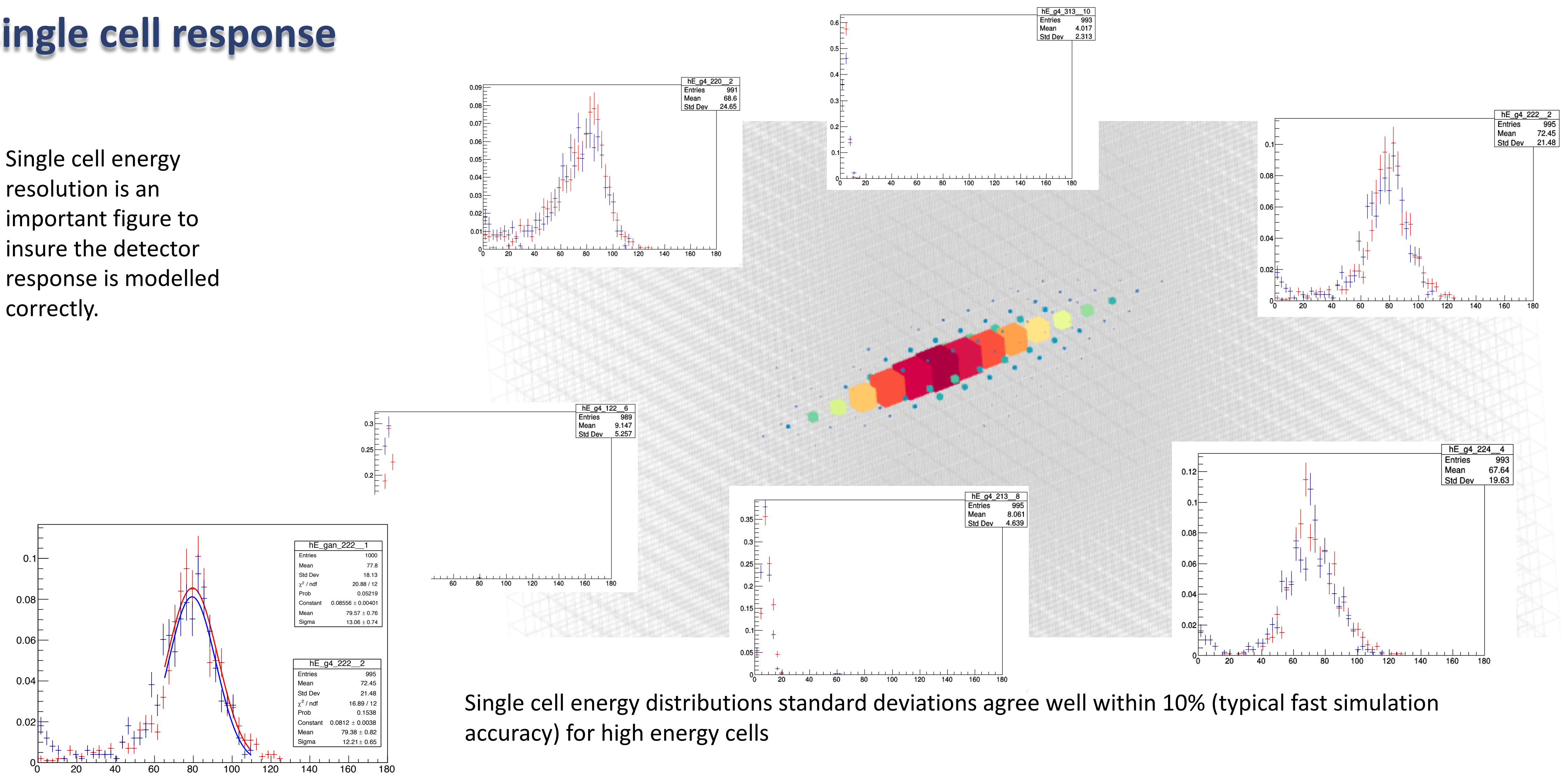


Using a trained GAN to generate images result sin several orders of magnitude speedup

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

Single cell response

Single cell energy resolution is an important figure to insure the detector response is modelled correctly.



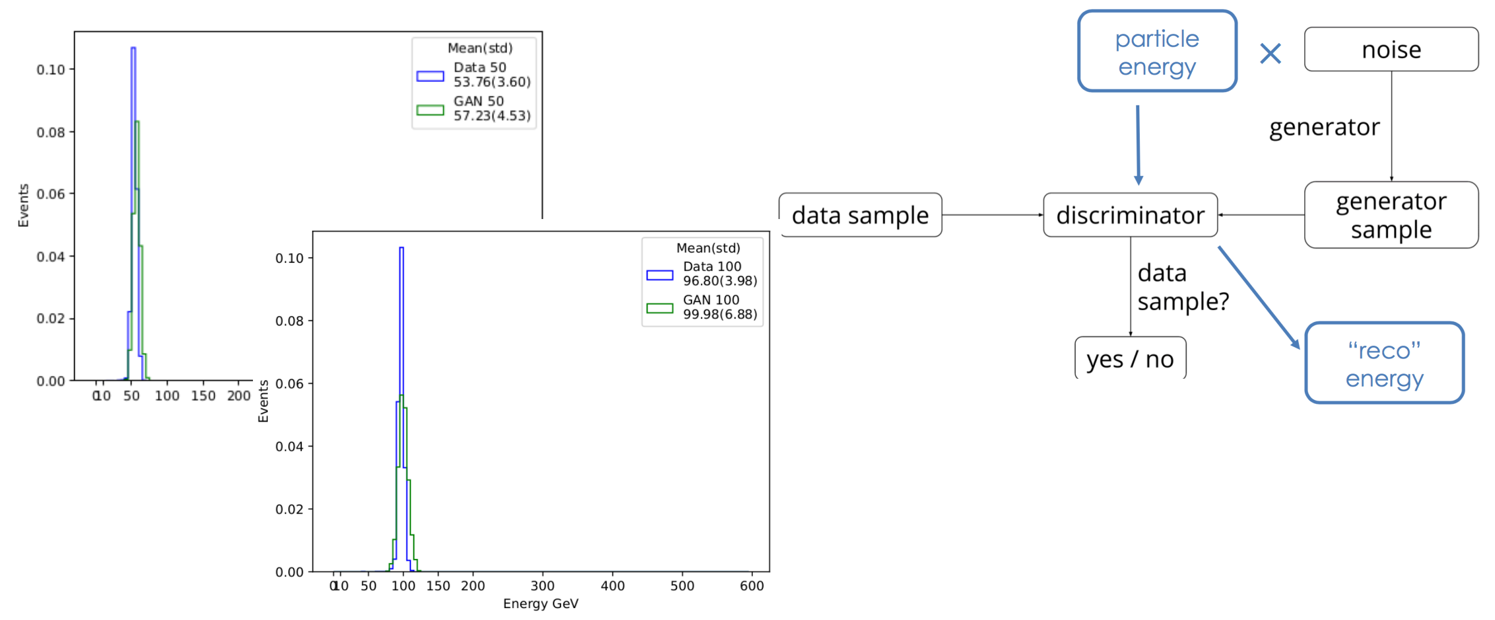
Single cell energy distributions standard deviations agree well within 10% (typical fast simulation accuracy) for high energy cells

Development

Conditioning on energy

Training the generator and the discriminator using initial particle energy

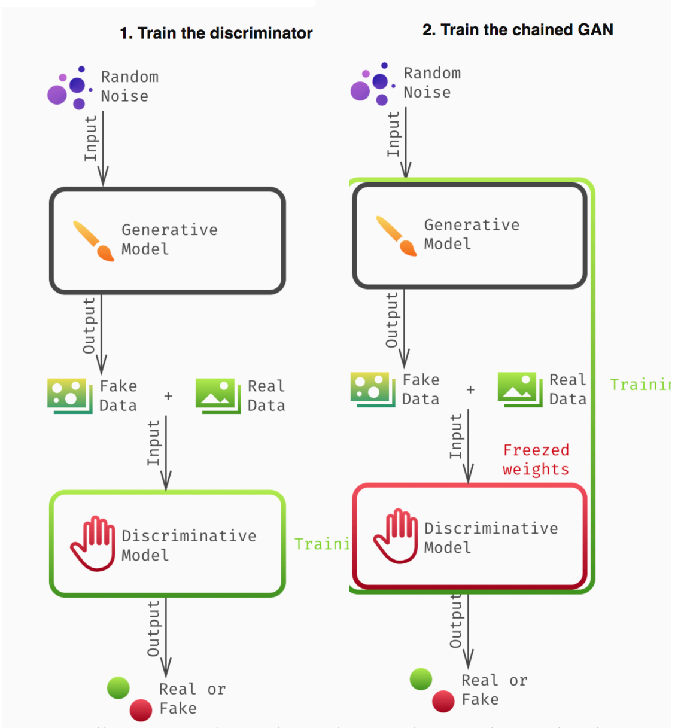
- Discrete energy slices to test interpolation and extrapolation
- Test continuous spectrum
- Add other variables (primary entry point, angle, etc..)



Primary Energy	Disc. of real images		Disc. of GAN images	
	Value	Error %	Value	Error %
50 GeV	53.7	7.4	57.23	14.5
100 GeV	96.8	3.2	99.98	2
200 GeV	180.4	9.8	189.76	10
300 GeV	269.3	12.5	289.99	5.12
400 GeV	355.7	6.1	375.7	6.1
500 GeV	444.7	11.6	463.7	7.3

Training time and multi-node scaling

- VGAN are not “out-of-the-box” networks
  - Complex training process
- Training time cannot be a bottleneck
  - Depending on the use case retraining might be necessary
  - Hyper-parameters scan and meta-optimization
  - Including additional variables will increase complexity
  - Training for 30 epochs on a single GPU GeForce GTX 1080 takes approximately 24h



http://www.rricard.me/machine/learning/generative/adversarial/networks/2017/04/05/gans-part1.html

- Thanks to a collaboration with CINECA, Italy and Intel, we will **test multi-node scaling** on a cluster of Xeon Phi interconnected with Intel Omni-Path

References

- Goodfellow et al. 2014
- Conditional GAN, arXiv: 1411.1744
- Deep Convolutional GAN, arXiv:1511.06434
- Auxiliary Classifier GAN, arXiv:1610.0958

GeantV is a collaboration among several research institutes. It is also partially funded by Intel Parallel Computing Center program (geant.cern.ch)