Generative models for fast simulation

Sofia Vallecorsa*

for the GeantV project

ACAT 2017
21-25 August 2017
University of Washington, Seattle

* Gangneung-Wonju U. & CERN

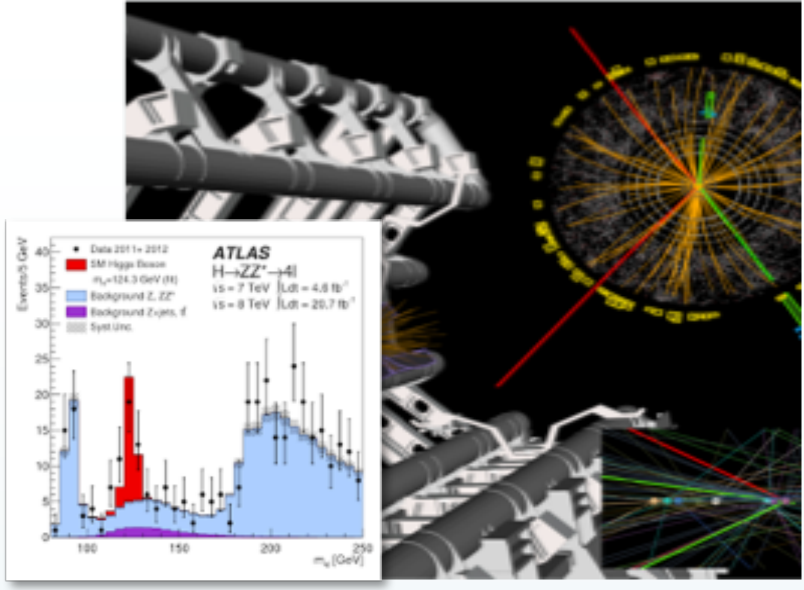
Outline

- Introduction
  - Detector Simulation and fast simulation
  - A general framework: Deep Learning tool for fast simulation
- Simulation as an image reconstruction problem
  - Generative Adversarial Networks (GAN)
  - Some examples
- Summary & Outlook

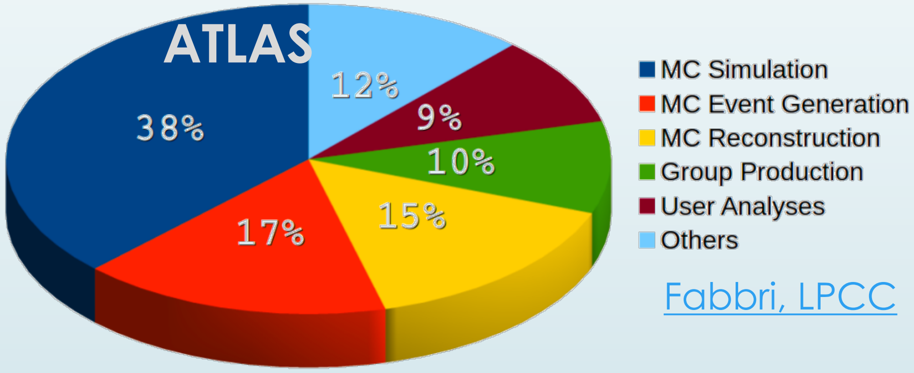
3

Simulation in HEP

- Detailed simulation is essential from detector R&D to data analysis
- Large statistics are generally needed to reduce systematic errors or study rare signals
  - Complex physics and geometry modeling
  - Heavy computation requirements, strongly CPU-bound
- More than 50% of WLCG power is used for simulations



Wall clock consumption 1/01/2016-04/06/2017



[Fabbri, LPCC]

WLCG Worldwide LHC Computing Grid

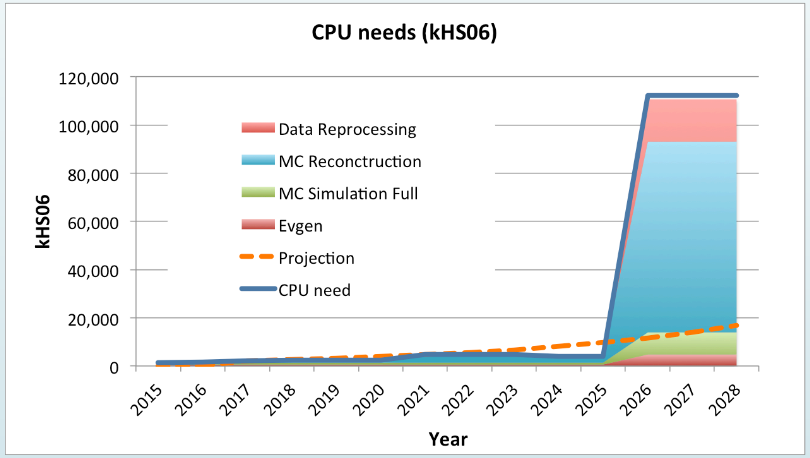
200 Computing centers in 20 countries: > 600k cores

@CERN (20% WLCG): 65k processor cores ; 30PB disk + >35PB tape storage

The problem

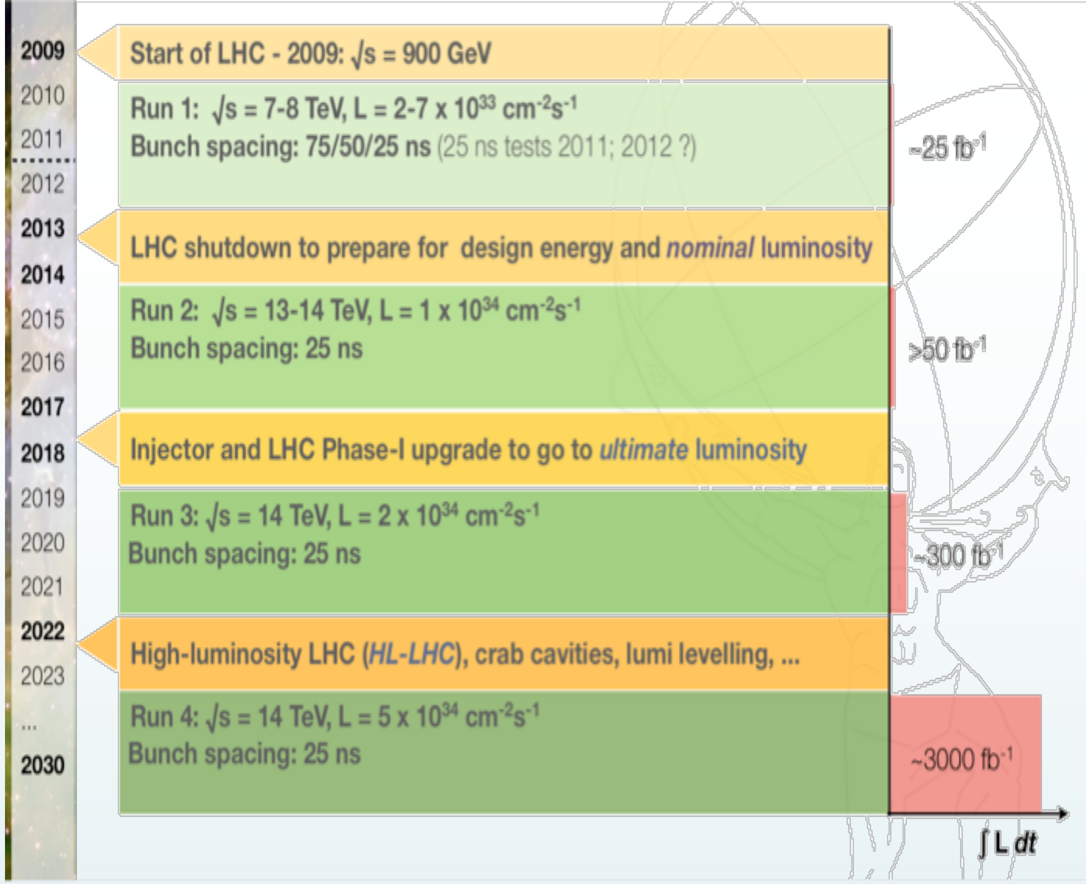
High Luminosity LHC

- Higher Luminosity → higher statistics → smaller simulation errors → larger MC statistics (.. and precise physics modelling)



ATLAS computing needs

[Campana, CHEP 2016]



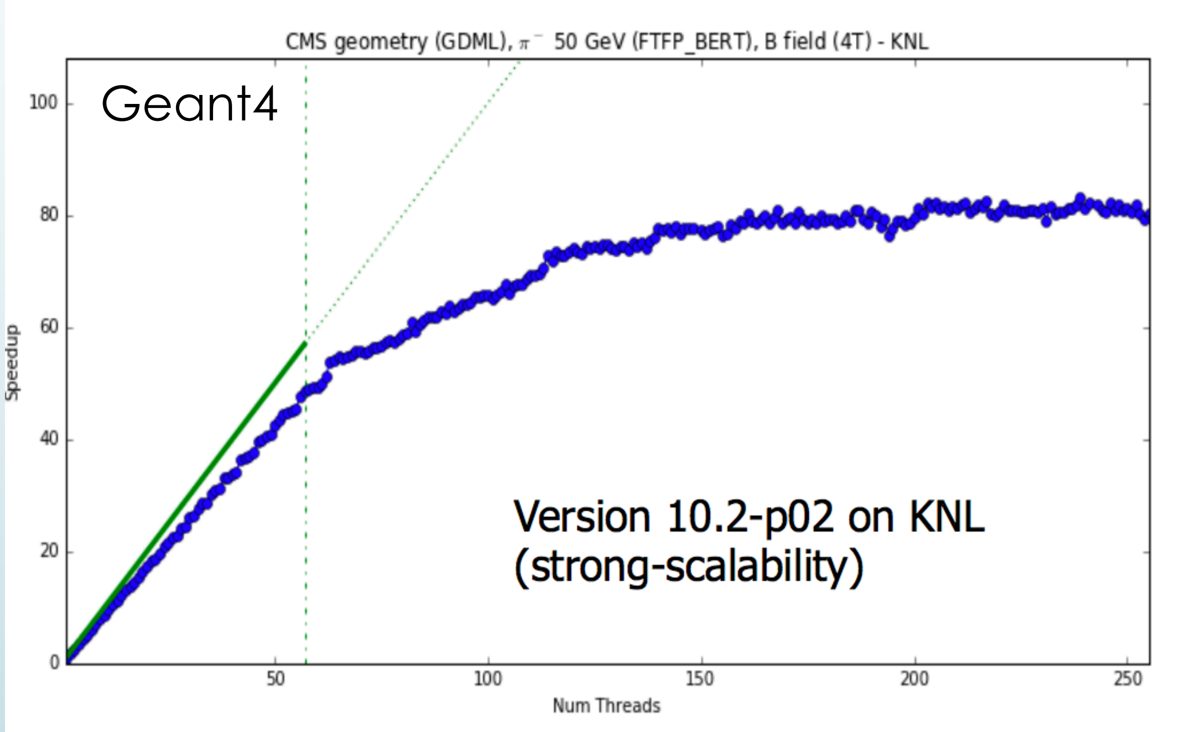
Other communities share similar needs:

- Intensity frontier experiments need to have detailed description of larger phase spaces

Speeding up simulation

Several initiatives are on-going

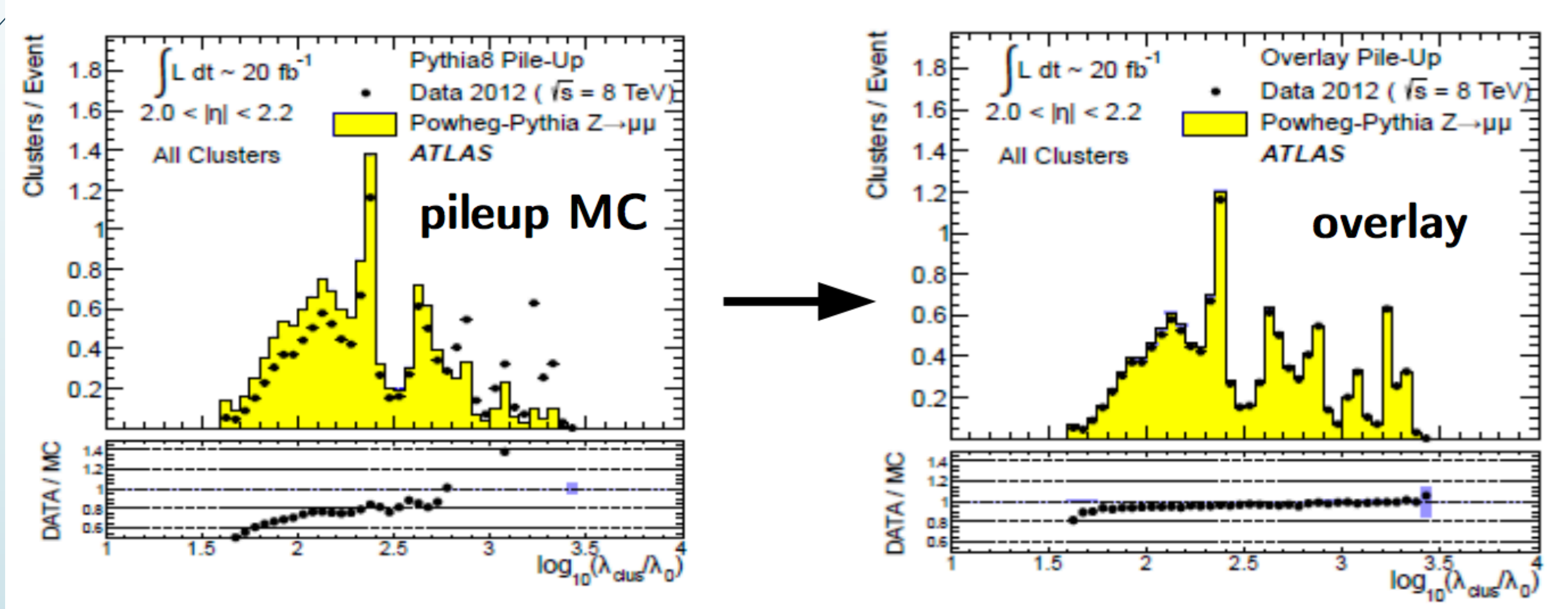
- **Introduce multi-threading and/or task** based approach (GaudiHive, GaudiMP, Geant4 Multi-threading)



Asai, LPCC

Speeding up simulation

- **Mix data to simulation** (pile-up overlay techniques) to reduce CPU time and memory



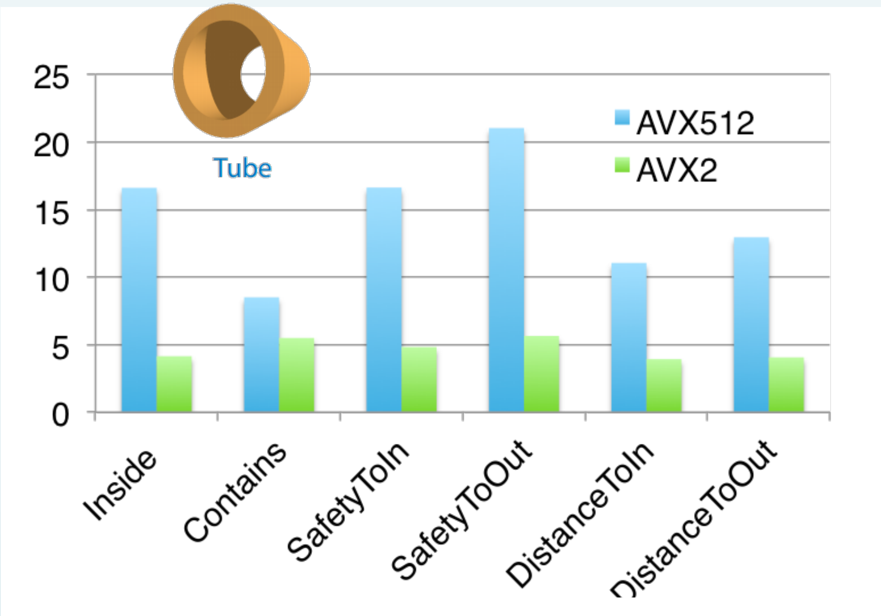
[Haas, CHEP 2016]

see A: Gheata talk on GeantV
and G. Amadio poster on VecCore

Speeding up simulation

- **Introduce fine grained parallelism**
  - GEANTV aims at x5 speedup through vectorisation, concurrency, locality
  - Improved geometry algorithms: VecGeom library developed for GEANTV (also available to GEANT4 and ROOT)
  - New SIMD library (VecCore)

VecGeom vectorisation speedup measured on Intel Xeon Phi



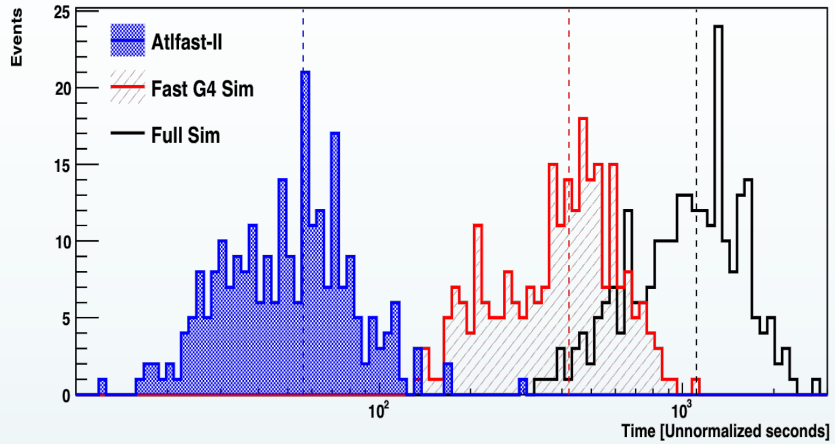
S.V., PASC17

Going beyond: Fast Simulation

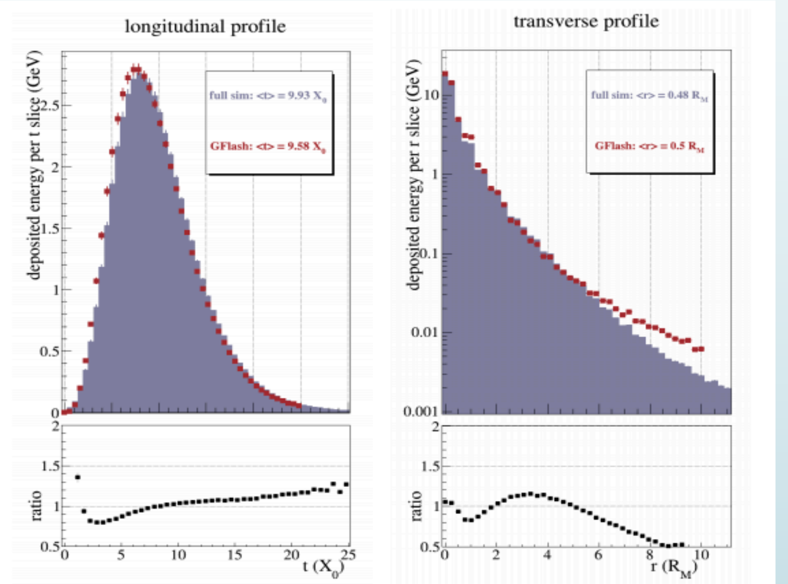
- Already used for searches, upgrade studies,…
- **Different techniques**
  - Shower libraries (pre-simulated EM showers, fwd calorimeters in ATLAS/CMS)
  - Shower shapes parametrizations (GFlash,..)
  - Fast trackers simulation (ATLAS FATRAS, ..)
  - Look-up tables
  - Fully parametrized simulation (DELPHES)
- **Different performance**
  - Different speed improvements (x10 - x1000)
  - Different levels of accuracy (~10% wrt full sim)

Choice is "experiment" dependent!

ATLAS



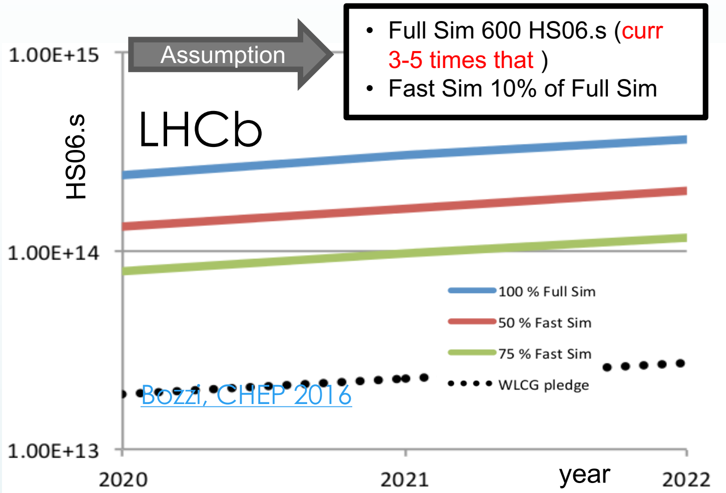
FCChh



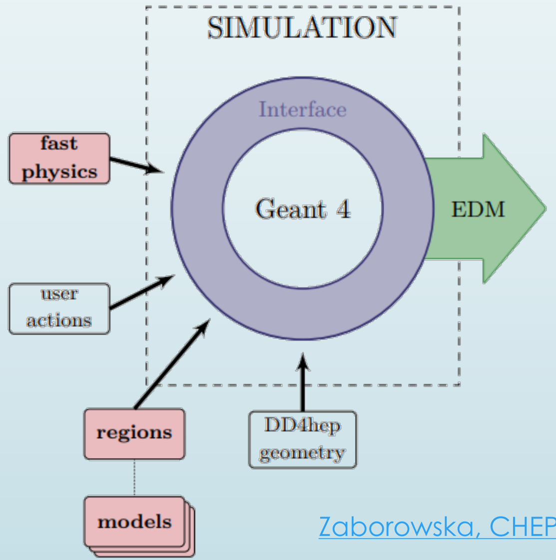
Zaborowska, CHEP2016

A generic framework for fast simulation

- MC need to integrate fast simulation
  - GEANT4 has mechanism to mix fast and full simulation: user-defined models within "envelopes" → few use it
- Towards a common framework providing
  - Algorithms and tools
  - Mechanism to mix fast and full simulation according to particle type and detector
- R&D within GeantV to develop a generic fully customizable fast sim framework
  - Deep Learning based



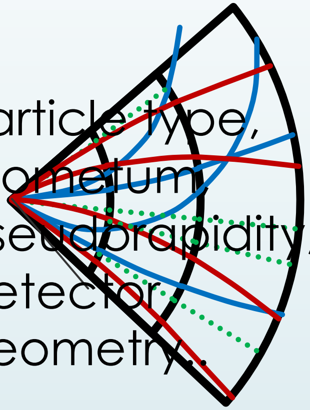
FCC Gaudi framework



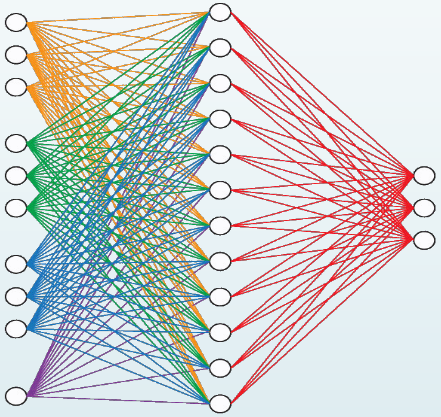
Zaborowska, CHEP2016

Deep Learning for fast sim

EX. SIMULATION OF A CALORIMETER

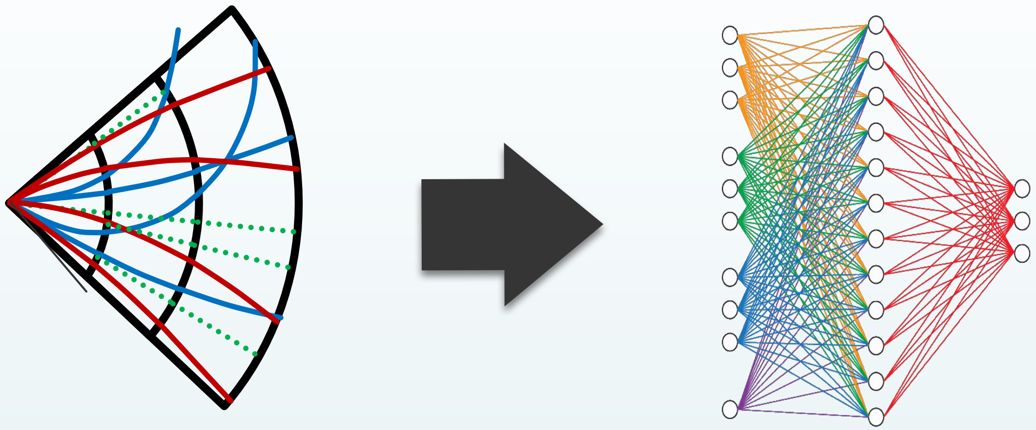


Particle type, momentum, pseudorapidity, detector geometry..



Energy depositions in cells

Deep Learning for fast sim



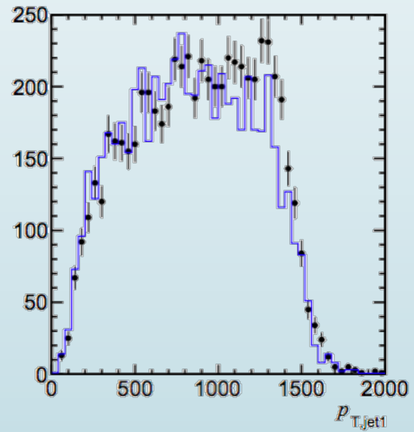
- Generic approach
- Can encapsulate expensive computations
- DNN inference step is generally faster than algorithmic approach
- Already parallelized and optimized for GPUs/HPCs.
- Industry building highly optimized software, hardware, and cloud services.

A precursor - Falcon

Ultra-fast, self-tuning, non-parametric simulation based on lookup tables that directly map generated events into simulation events

- Turbosim (B. Knuteson) developed at the Tevatron
- Falcon: Modern version (Gleyzer at al., 1605.02684)
- Consists of two parts:
  - Builder: Non-parametric representation of the detector response function obtained from FullSim events.
  - Uses a k-d tree to bin the generated objects in the lookup table.
  - Simulator: Uses events in the parton level to simulate reconstruction level events.

Leading jet p_T from $p + p \rightarrow H \rightarrow f\bar{f}$ events



Generative Models

Generative models

The problem:

- Assume data sample follows p_{data} distribution
- Can we draw samples x from distribution p_{model} such that $p_{model} \approx p_{data}$?

A well known solution:

- Assume some form for p_{model}, using prior knowledge and parameterized by θ
- Find the maximum likelihood estimator

$$\theta^* = \arg\max_{\theta} \sum_{\mathbf{x} \in \mathcal{D}} \log(p_{\text{model}}(\mathbf{x}; \theta))$$

- Draw samples from $p_{\theta*}$

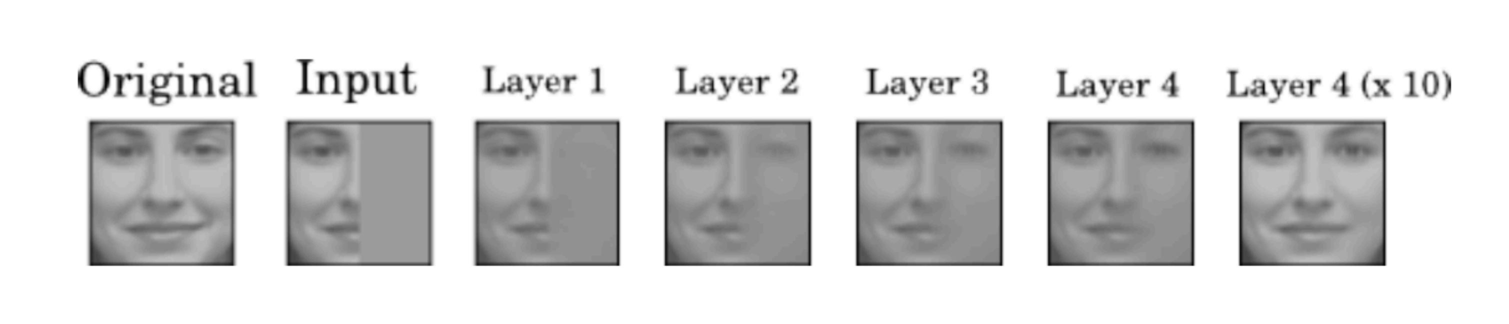
- Generative models don't assume any prior form for p_{models}
- Use Neural Networks instead

Generative models for simulation

Many models: Generative Stochastic Networks, Variational Auto-Econders, Generative Adversarial Networks ..

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

https://arxiv.org/pdf/1605.05396.pdf



Ranzato, Susskind, Mnih, Hinton, IEEE CVPR 201

Questions:

Can imaging approaches be useful?

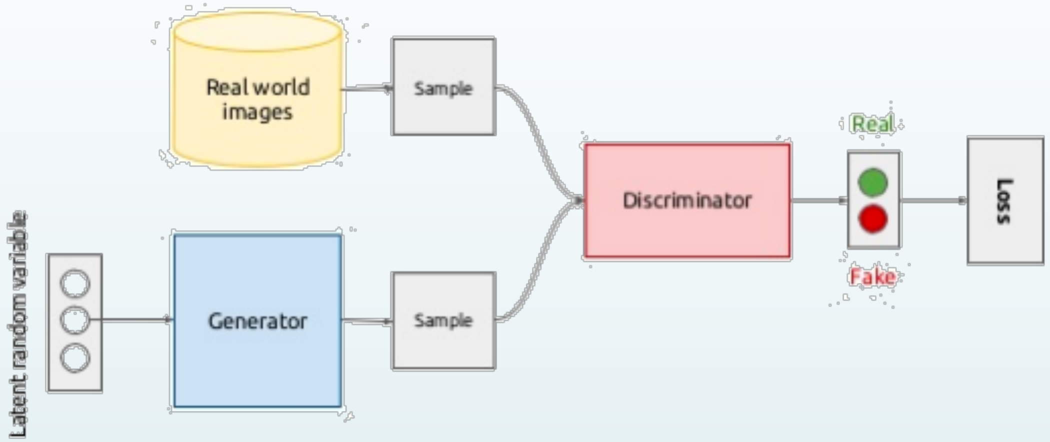
- Can we keep accuracy while doing things faster?
- Can we sustain the increase in detector complexity (future highly-granular calorimeters are more demanding)?
- What resources are needed?

arXiv:1406.2661v1

Generative adversarial networks

Simultaneously train two networks that compete and cooperate with each other:

- Generator learns to generate data starting from random noise
- Discriminator learns how to distinguish real data from generated data



The counterfeiter/police case

- Counterfeiter shows police the fake money
- Police says it is fake and gives feedback
- Counterfeiter makes new money based on feedback
- Iterate until police is fooled

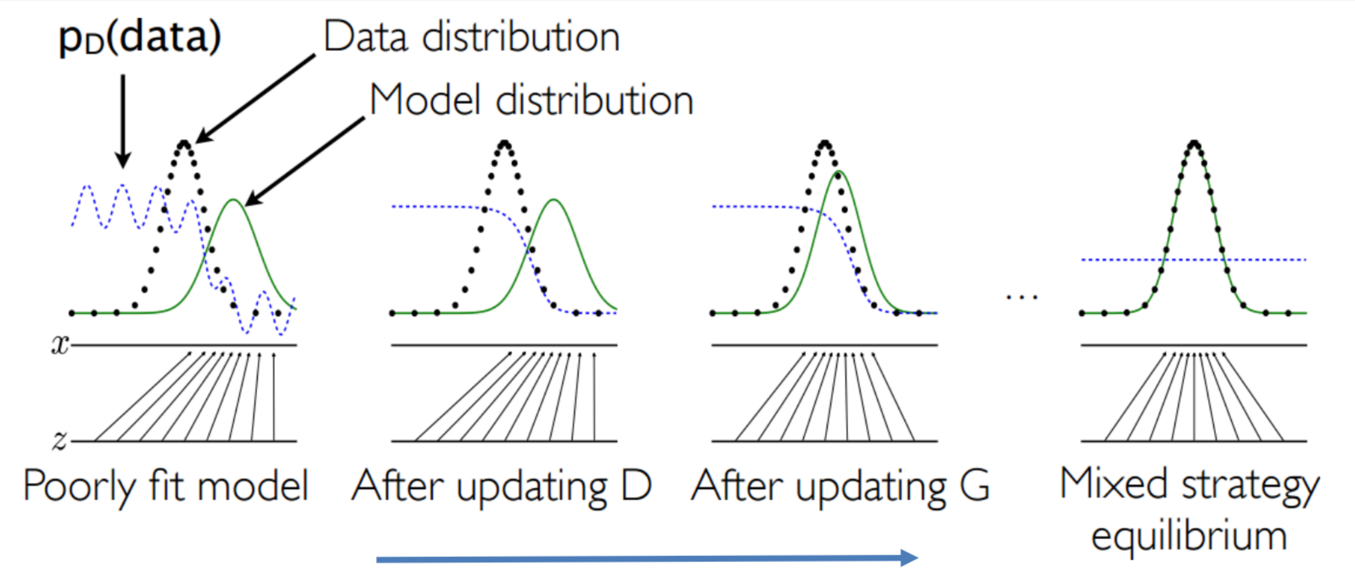
arXiv:1406.2661v1

Generative adversarial training

Generator is trained to maximize the probability of Discriminator making a mistake

D gradient guides G to regions more likely to be classified as data

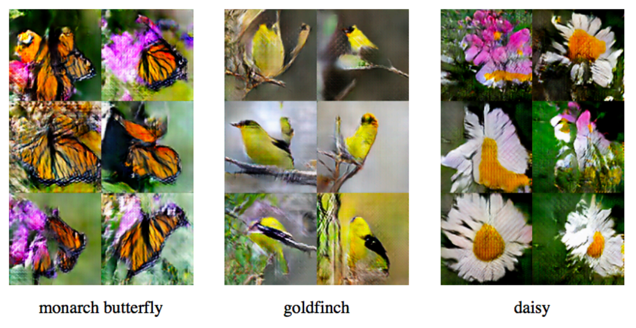
D is not an accurate classifier



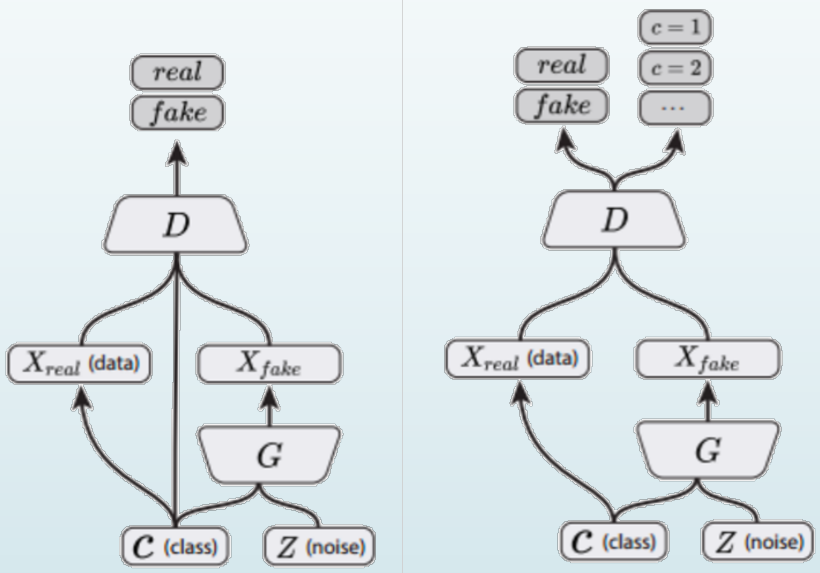
G and D don't improve anymore. D is unable to differentiate

D is trained to discriminate samples from data

Many GAN flavors



- Original GAN was based on MLP in 2014
- Deep Convolutional GAN in 2015
- Conditional GAN
  - Extended to learn a parameterized generator $p_{model}(x \mid \theta)$;
  - Useful to obtain a single generator object for all θ configurations
  - Interpolate between distribution
- Auxiliary Classifer GAN
  - D can assign a class to the image



Conditional GAN (Mirza & Osindero, 2014)

AC-GAN (Present Work)

arXiv: 1411.1784

arXiv:1610.0958

Applications

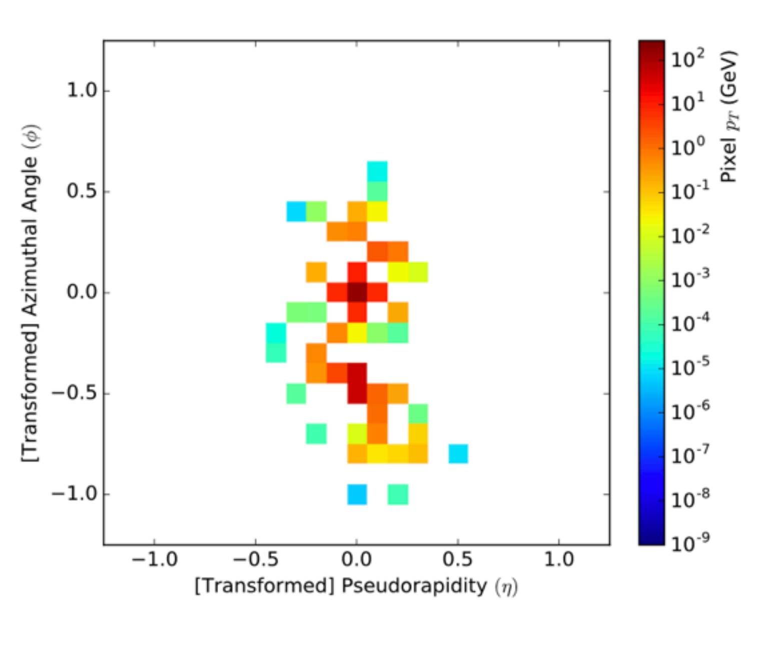
LAGAN & CaloGAN

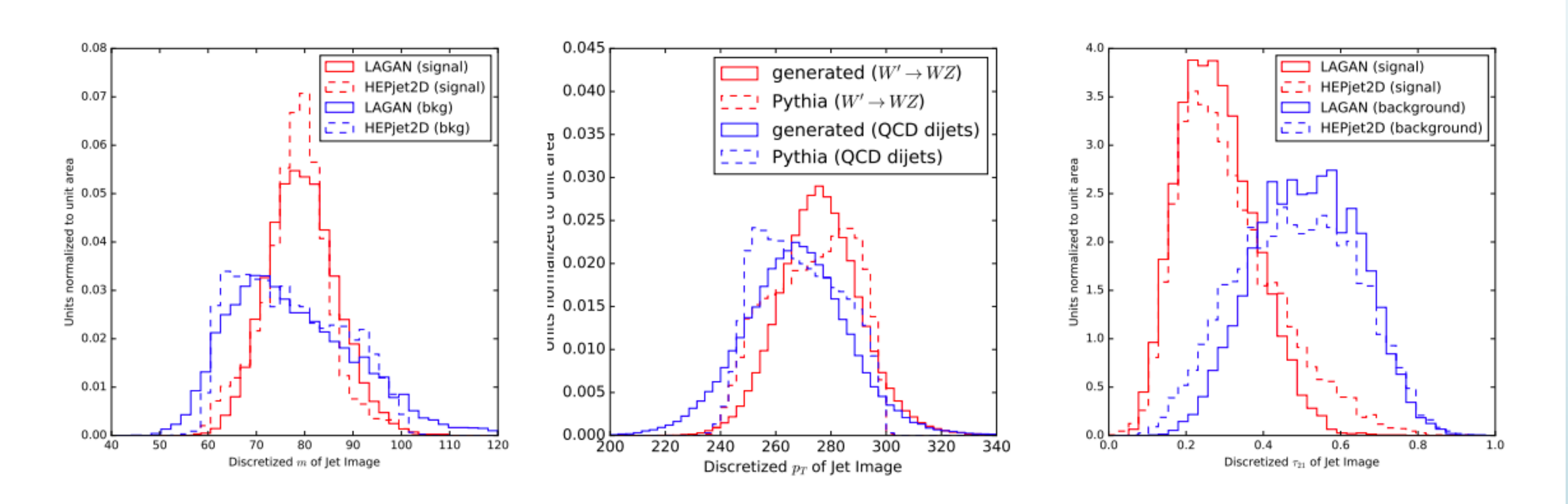
See Paganini
and de Oliveira
talks in parallel
sessions

arxiv:1701.05927

Location Aware GAN

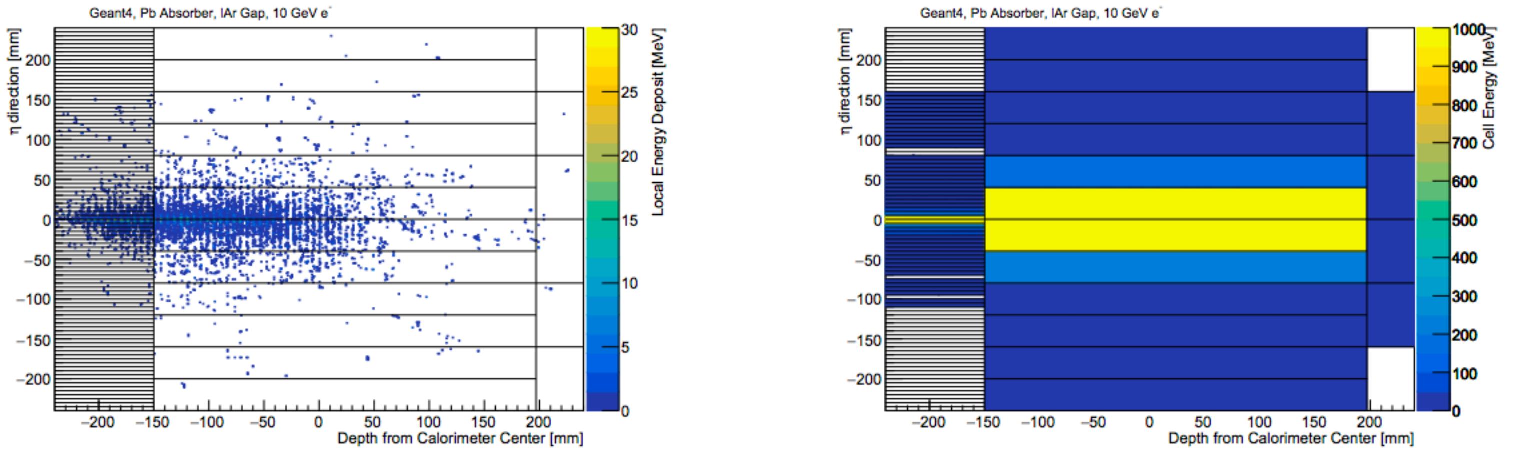
- Reproduce 2D generator level anti-kT jet images (generator-level study)
- Modification of DCGAN (convolutions) and ACGAN (uses particle type information)
- Image sparsity
- Location dependent features
- Large dynamic range



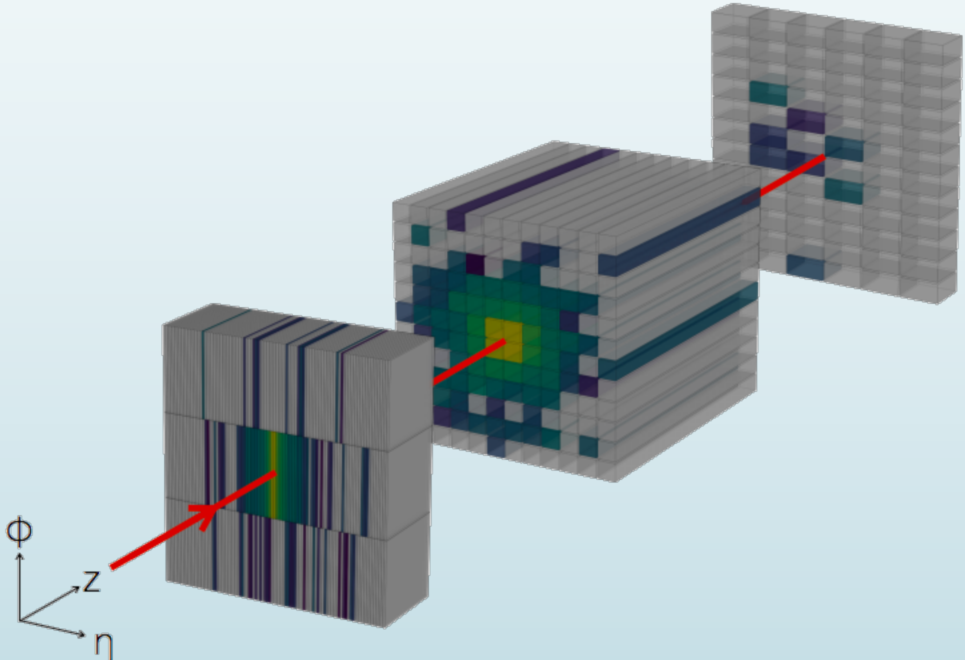


arxiv:1705.02355

CaloGAN

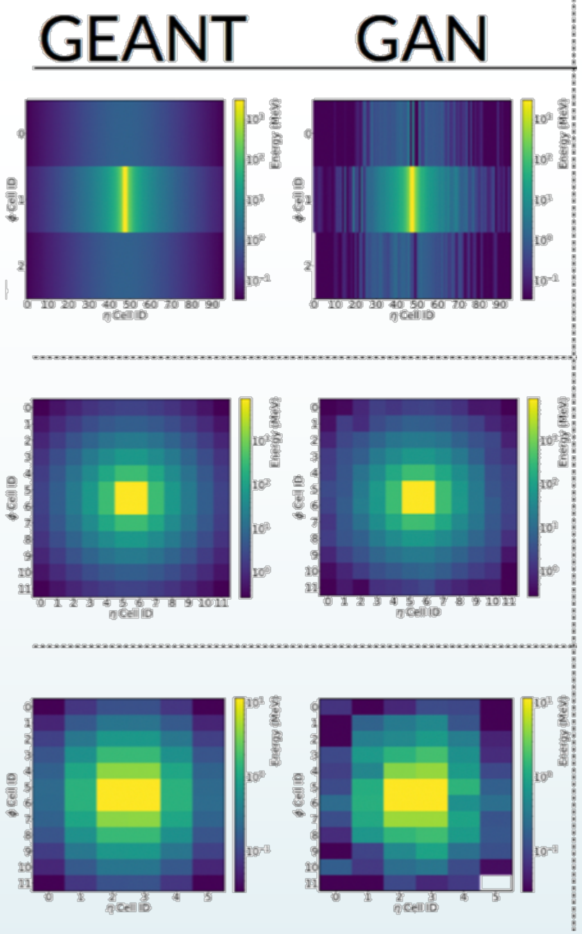


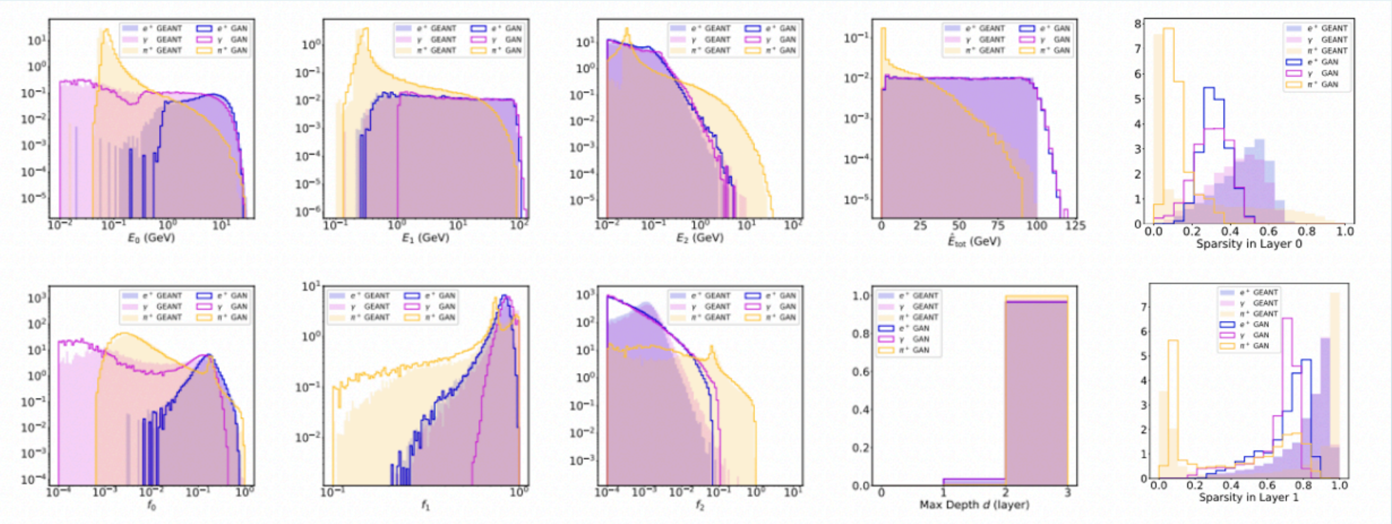
- ATLAS LAr calorimeter
  - Heterogeneous longitudinal segmentation into 3 layers
  - Irregular granularity in eta and phi
- Energy deposition in each layer as a 2D image
- Build one LAGAN per layer
- Trainable transfer unit to preserve layer correlations
- Result is a concatenation of 2D images that reproduce full 3D picture

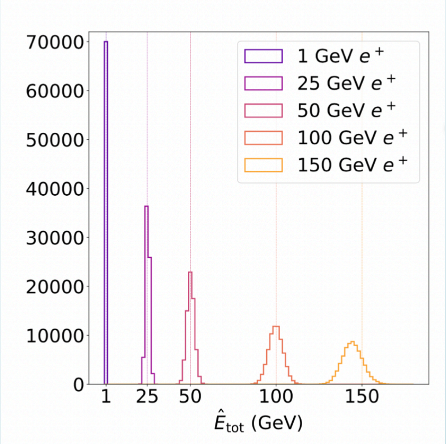


CaloGAN performance

- Comparison to full simulation:
  - Average showers
  - Shape variables (depth, width, layer energy..) and event variables (sparsity level per layer)
- Energy reconstruction



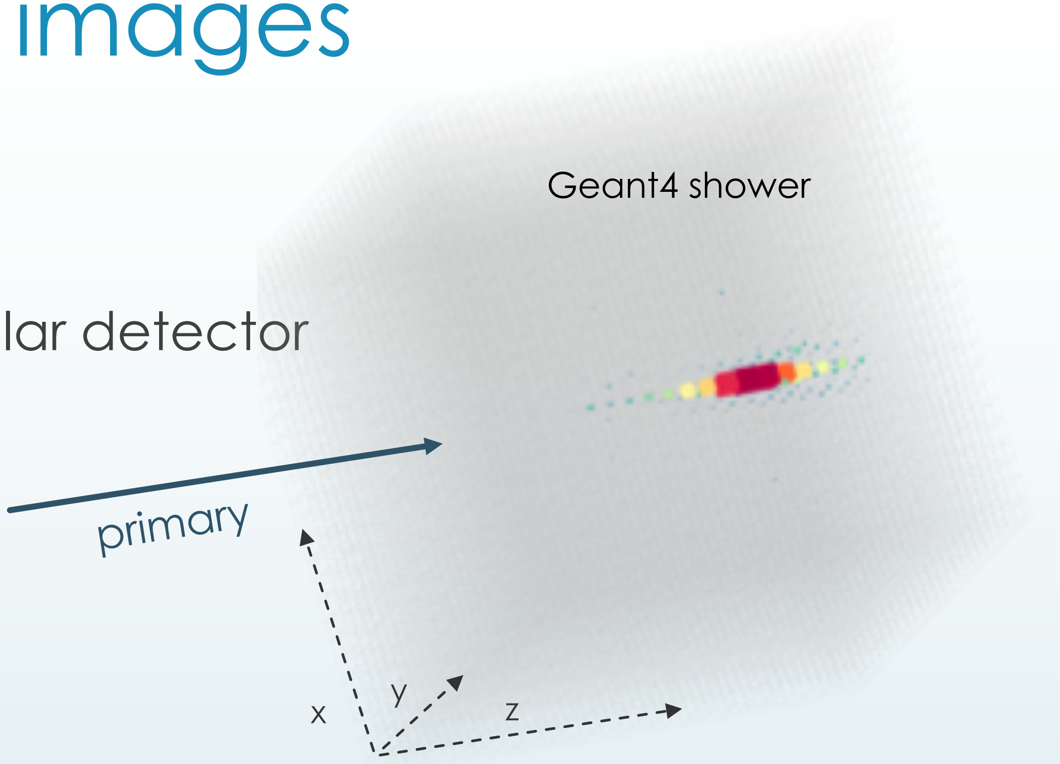


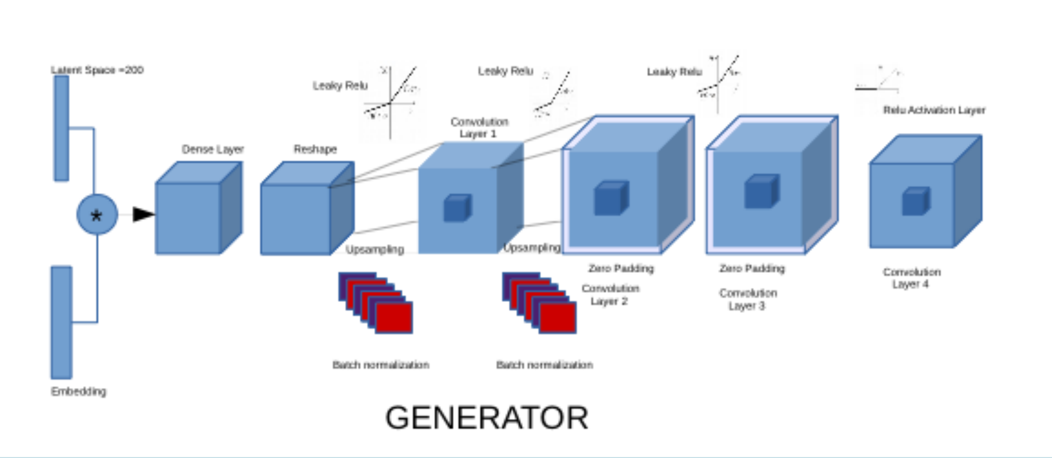


3d GAN

3d GAN for calorimeter images

- CLIC electromagnetic calorimeter (*)
  - Example of next-generation highly granular detector
  - Data is essentially a 3D image
- Based on convolution/deconvolutions
  - 3D (de)convolutions to describe full shower development
- Particle tag as auxiliary classifier
- Implementation/Training details in backup

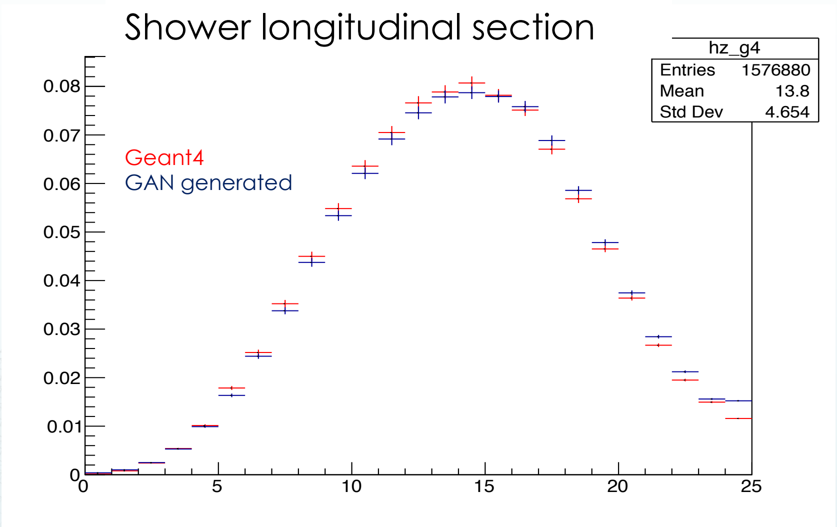


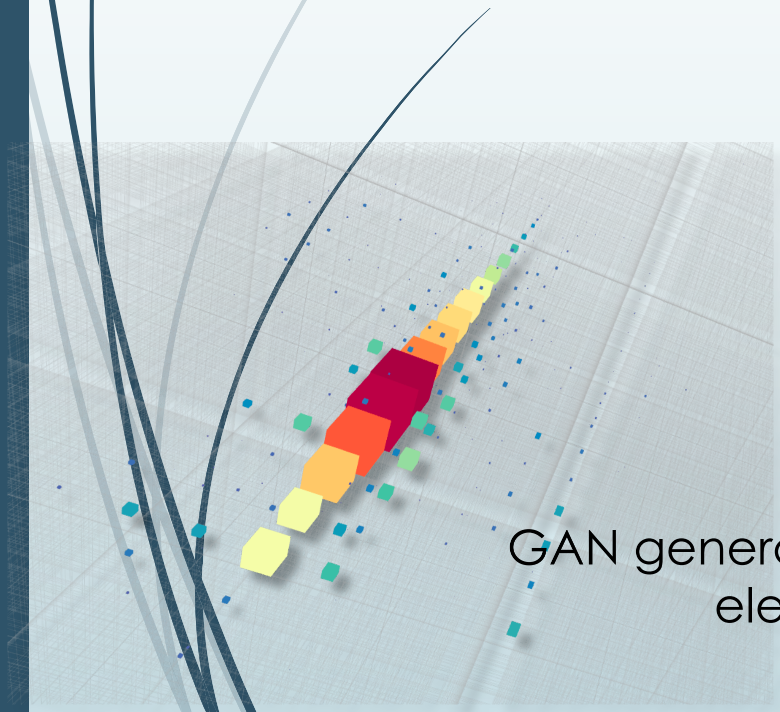


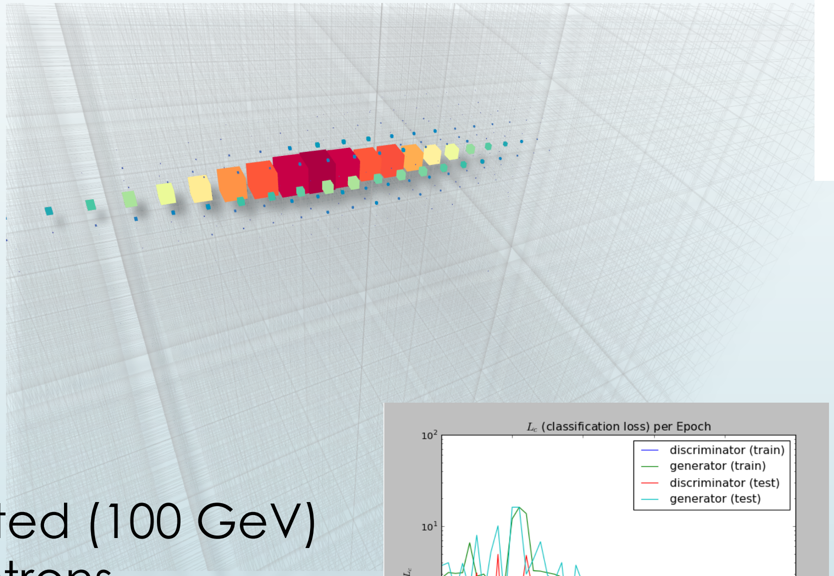
(*) http://cds.cern.ch/record/2254048#

First 3D images

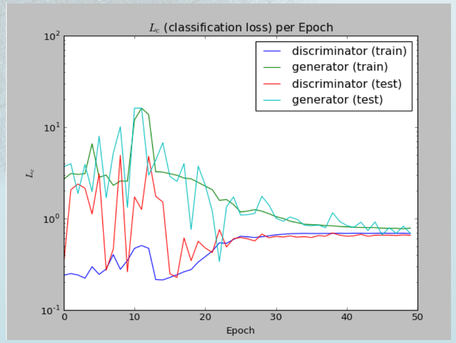
- First generated results look promising!
- Qualitative results show no collapse problem







GAN generated (100 GeV) electrons



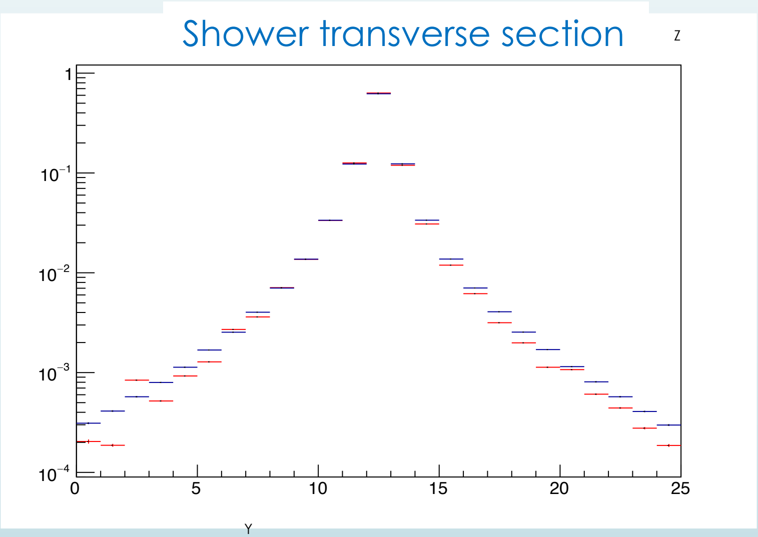
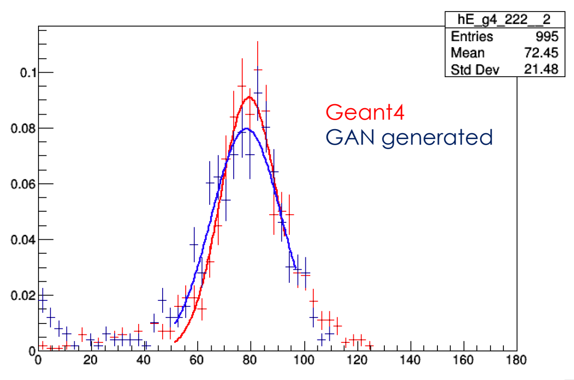
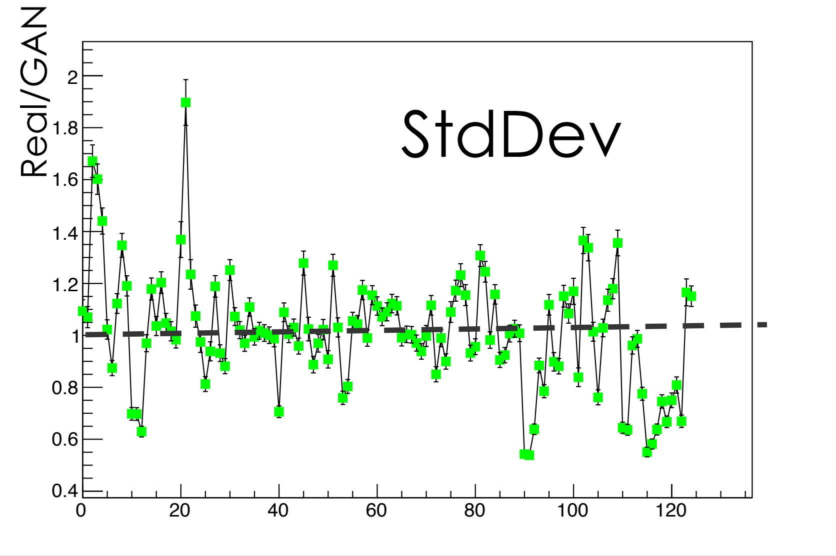


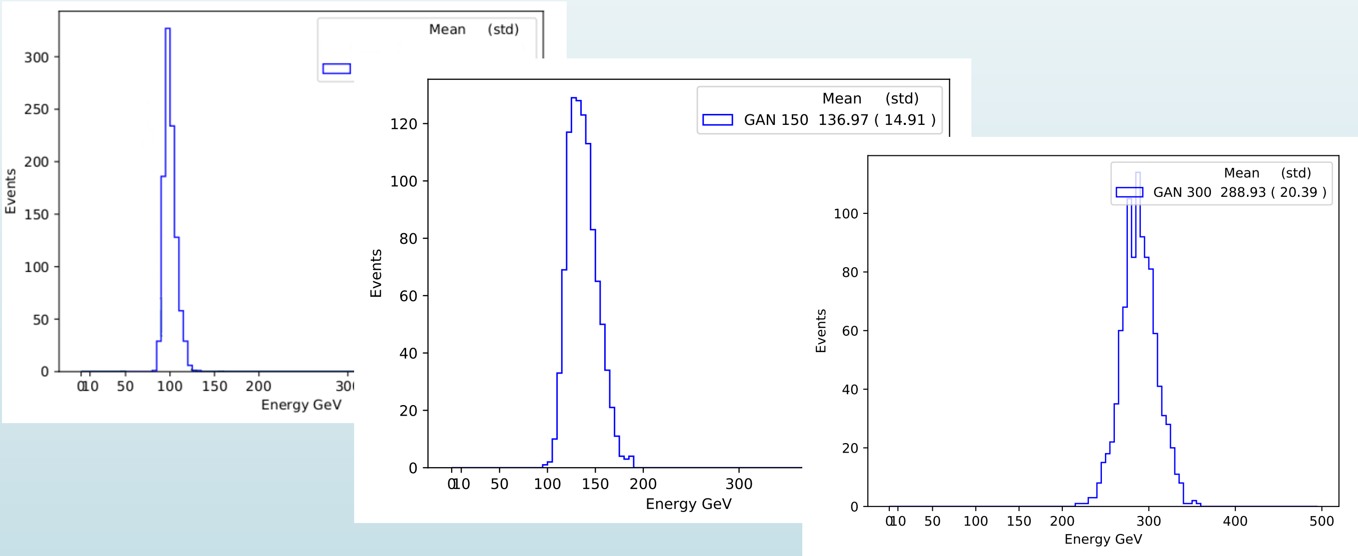
Image validation and energy response

- Detailed study of calorimeter response
  - Energy distribution in single cells





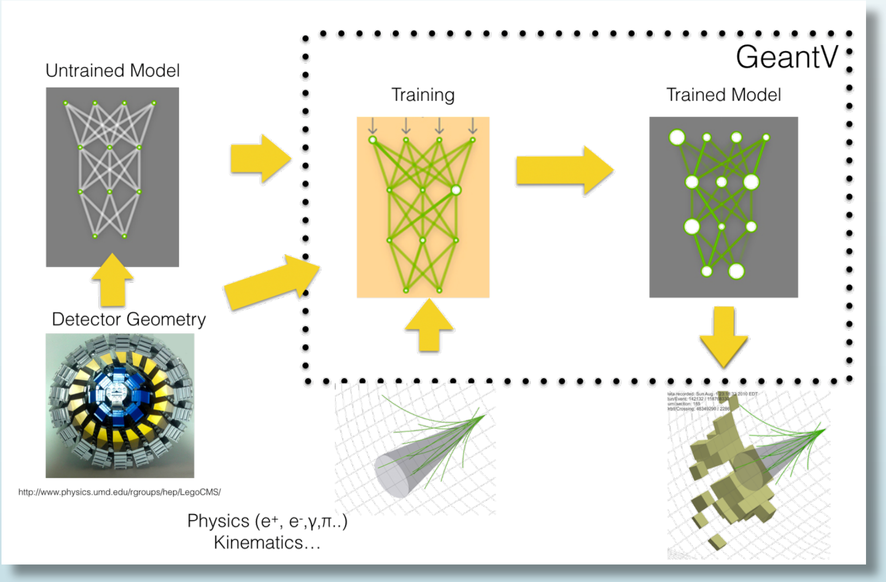
- Primary particle energy from discriminator



- Comparison to full sim and different fast sim tools is ongoing

DL engine for fast simulation in GeantV

- 3d GAN represent first proof of concept
  - We aim at a generic fully configurable tool
- Optimal network design depends on the problem to solve
  - Embedded algorithms for hyper-parameters tuning and meta-optimization
- Studying parallelization on clusters



Summary I

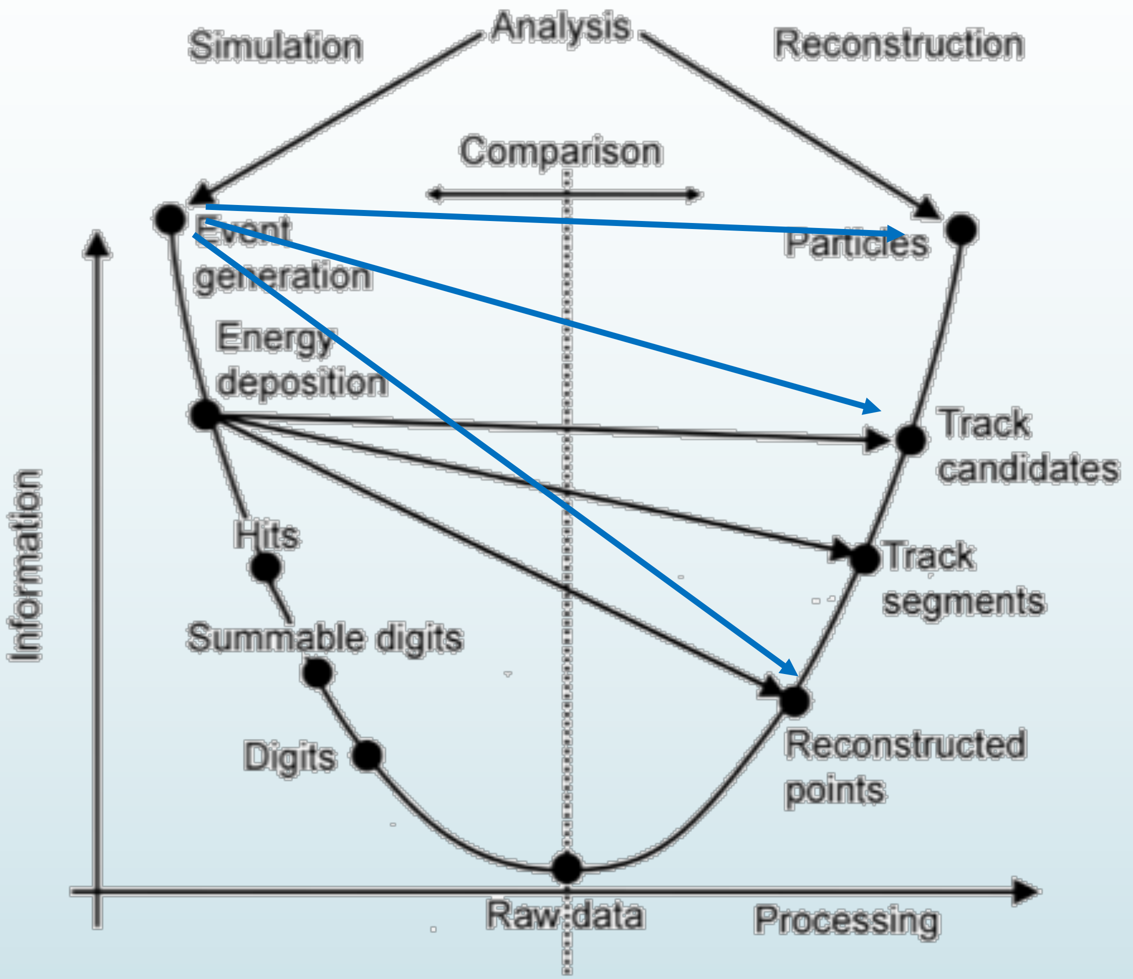
- MC production has been so far a major fraction of WLCG workload
  - Experiments are implementing a large range of fast simulation solutions
- HL-LHC runs will scale up MC needs by orders of magnitude
- A generic framework with common fast sim algorithm and strategies for mixing full and fast sim
  - Could bring great benefit to the HEP community
  - Serve small experiments/collaborations as well

Summary II

- Generative Models seem natural candidates to speedup simulation
  - Rely on the possibility to interpret “events” as “images”
  - First GANs applications to calorimeter simulations look very promising
  - Many studies ongoing in the different experiments
- 3d GAN is the initial step of a wider plan to do DL based fast simulation within the GeantV project

Outlook

- Even larger speedup gained by replacing digitization and reconstruction steps
- ML/DL tools are capable of "learning" extremely complicated feature spaces



Thank you!

Questions?

the very diverse GeantV team

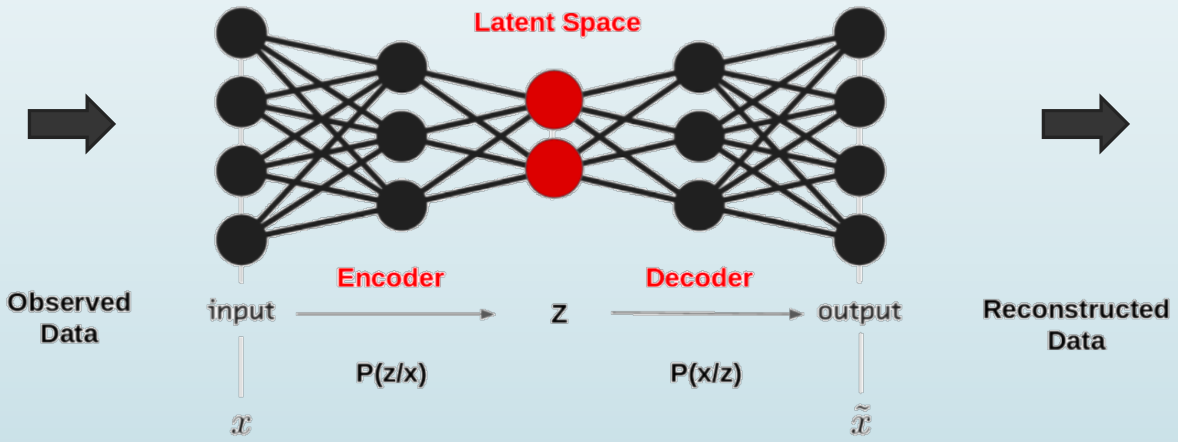
Proud of gender diversity in GeantV with 41% of female colleagues within the team!
(July 2017)

Variational Auto Encoders

- Typically used for un-labelled data and de-noising
- Two stacked NN (encoder – decoder)
- Sequentially de-construct input data into a latent representation
- Use this representation to reconstruct output that resembles the original

2D latent
space

Calorimeter
Shower Hit
(χ,y,z,E,σ)



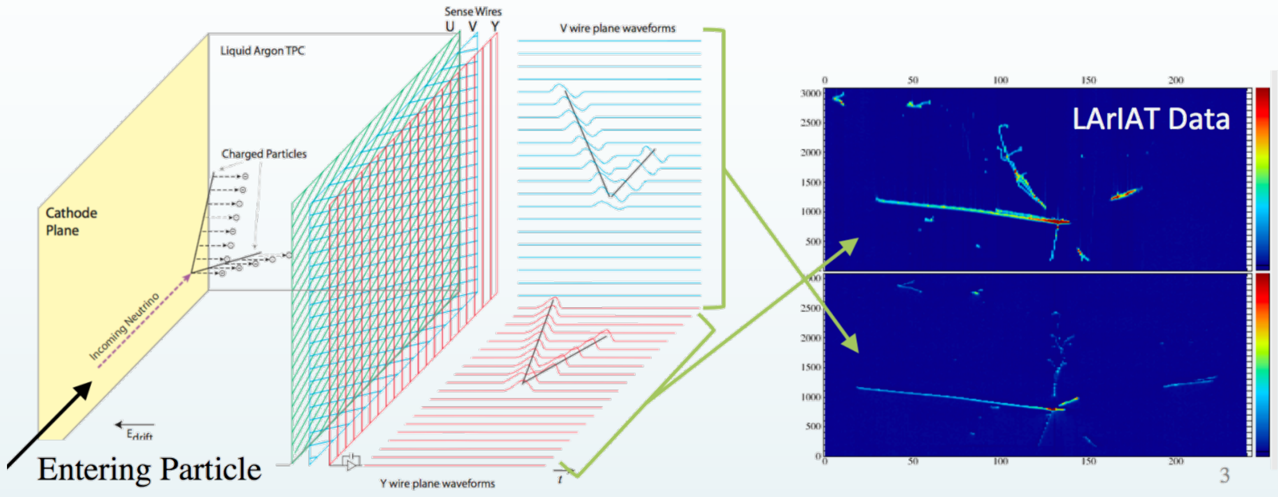
Generated
Calorimeter
Shower Hit
(χ,y,z,E,σ)

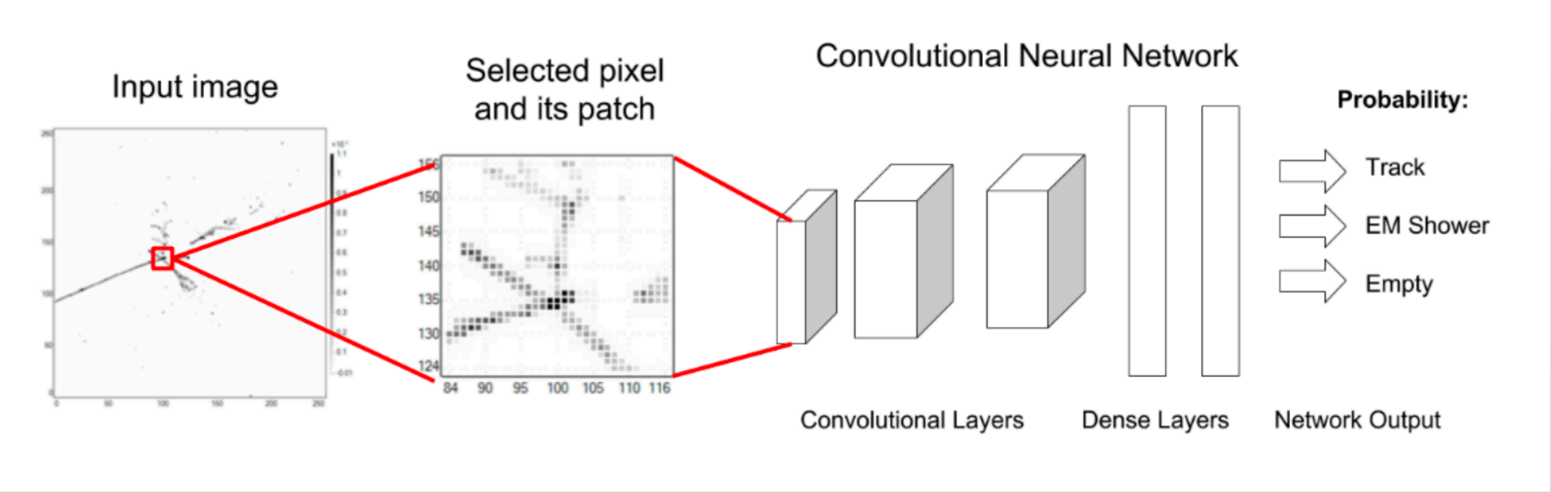
D.Salamani, U. of Geneva

Enhancing MC simulation with GAN

[Smith, IML workshop]

- An example from LAr TPC
- MC-Trained CNN to classify hits as shower-like or track-like
- Performed on noise-filtered ADC values after hit finding,
- one of the first reconstruction steps
- Greatly speeds up tracking
- Makes shower clustering possible

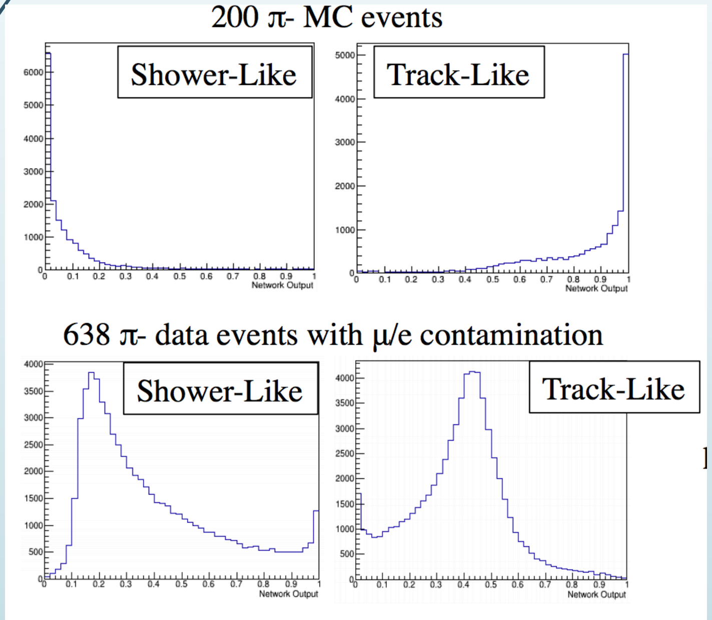


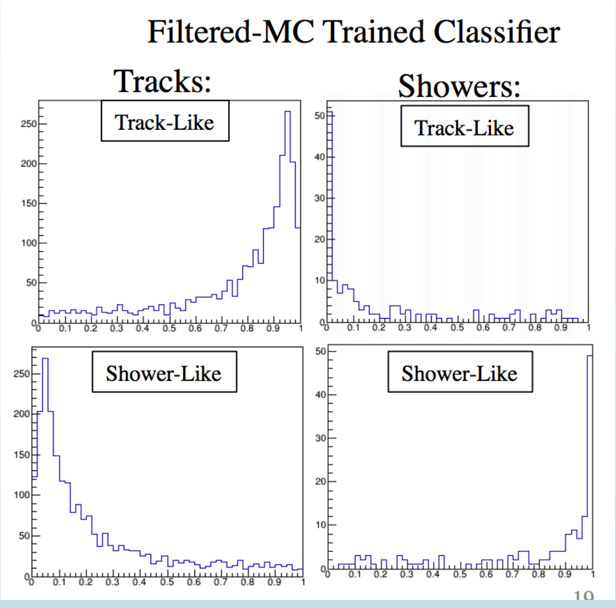


Enhancing MC simulation with GAN

Smith, IML workshop

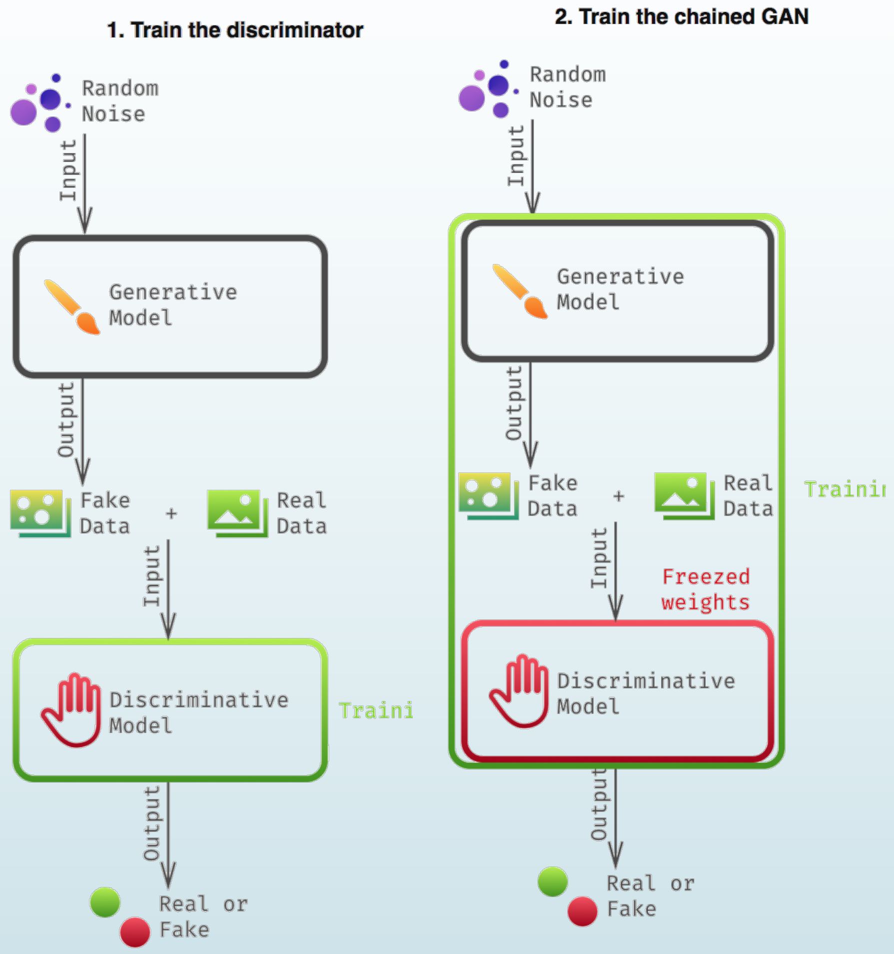
- Modify aGAN to pass in a MC sample into the generator, functionally turning it into a filter
- Training against data will create a data-driven filter for MC, allowing one to create a filtered MC sample that is very similar to data





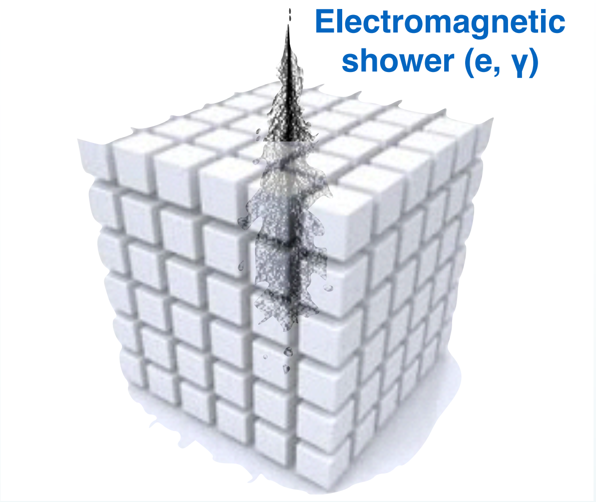
Training GANs is a many steps process:

1. Generate images with the Generator.
2. Train the Discriminator to recognize Generator data from Real data.
3. Push the combined model to tag it as Real data.
   I. Discriminator weights are frozen.
4. Back feed to Discriminator and repeat



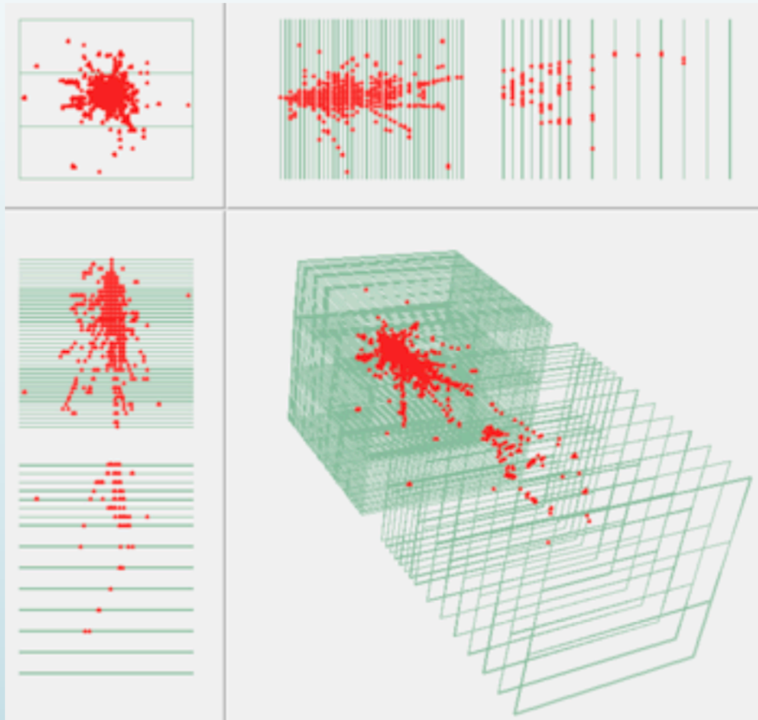
http://www.rricard.me/machine/learning/generative/adversarial/networks/2017/04/05/gans-part1.html

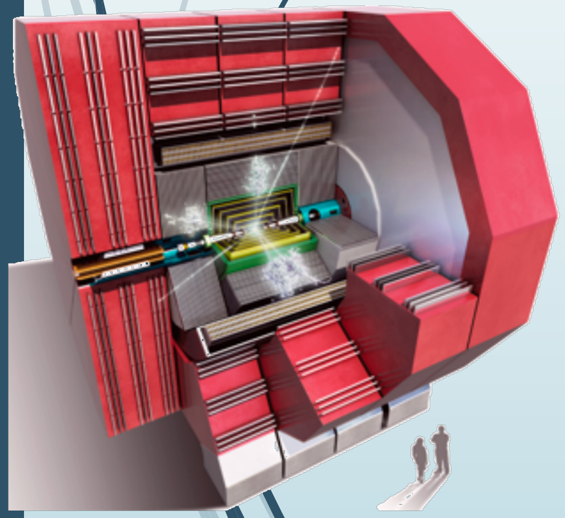
CLIC EM calorimeter data



- CLIC is a CERN project for a linear accelerator of electrons and positrons to TeV energies
- Calorimeter detector design associated to the project
- 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
- Geant4 single-particle datasets (e^+, e^-, γ, π)
- Simplified: no clustering/clustering id algorithms applied

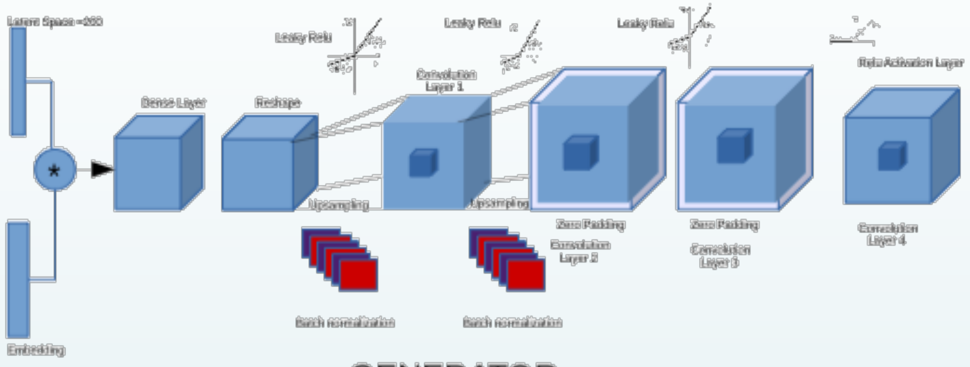
- Data released within CERN OpenData initiative



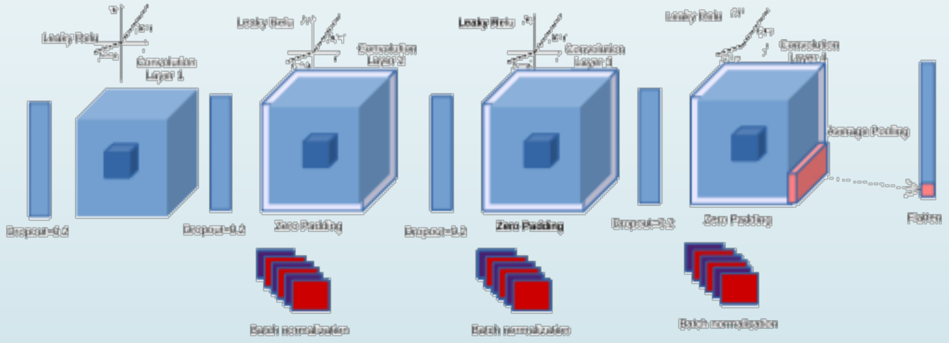


3dGAN for calorimeter images

- 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

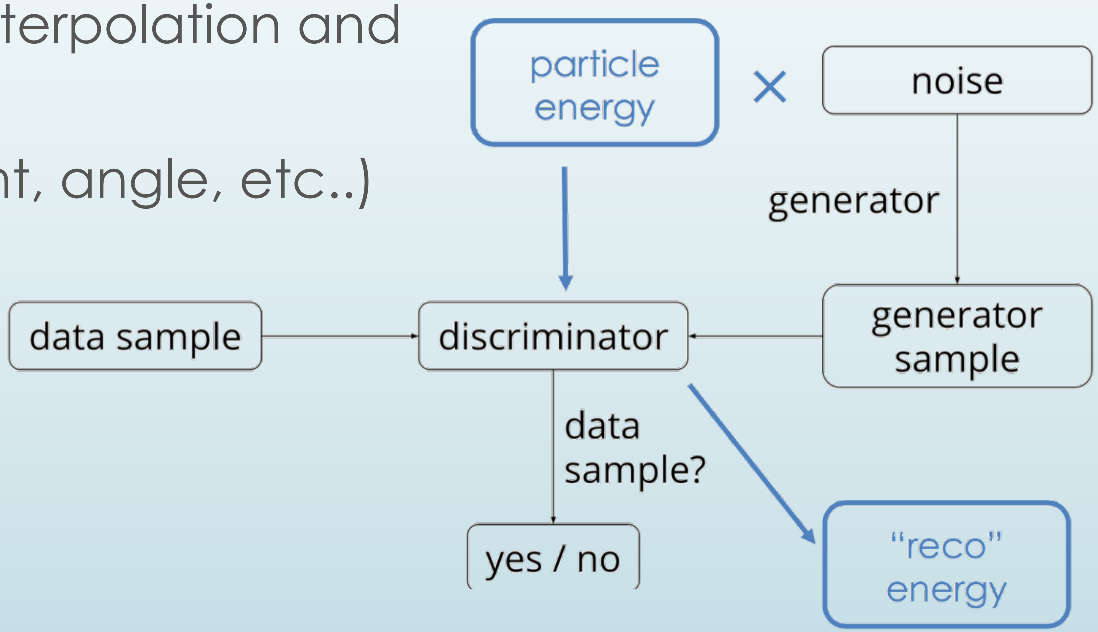


DISCRIMINATOR

Conditioning on energy

Training the generator and the discriminator using initial particle energy

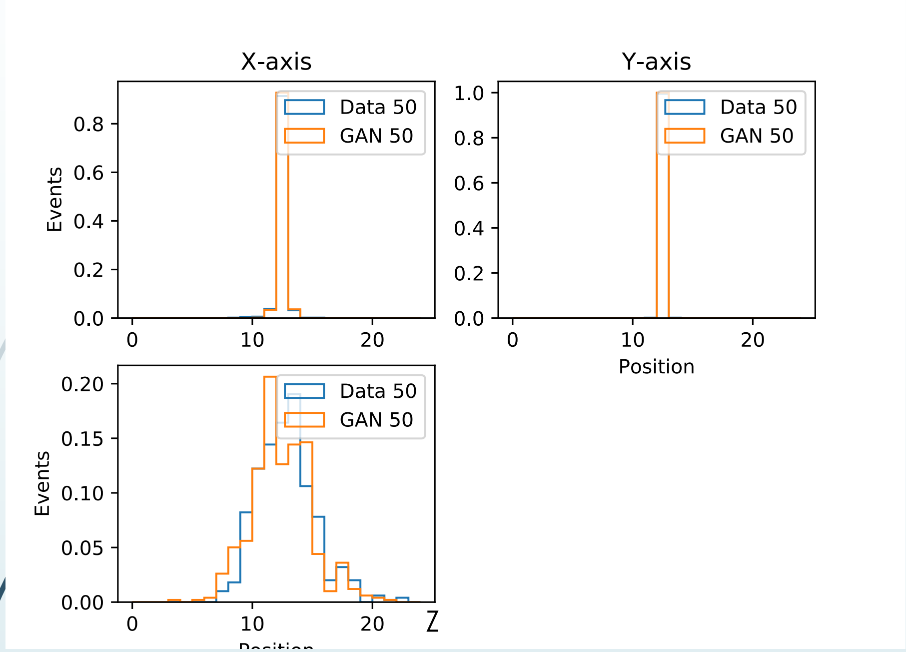
- The discriminator can be trained to do energy regression (including additional loss function)
- Test continuous spectrum and generate single energy points
- Train on fixed energy dataset to test interpolation and extrapolation
- Add other variables (primary entry point, angle, etc..)
- Energy loss is mean absolute error



Preliminary

3d GAN energy response

Energy regression test results



Energy regression test results

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

Z

Time costs?

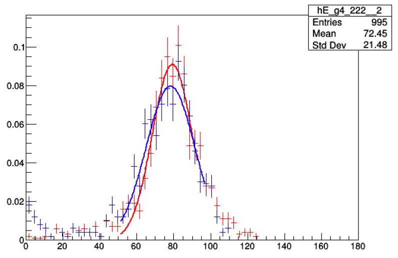
- Using DL techniques for fast simulation is profitable if training time is not a bottleneck
  - Depending on the final use case retraining the networks might be necessary
- Test different hardware & multi-node scaling
- Full Simulation generation time scales with energy

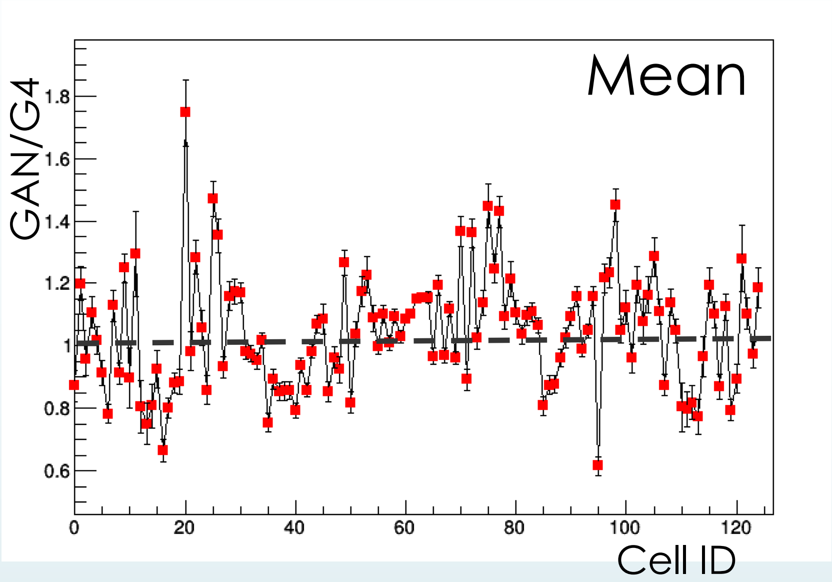
Generation Method	Hardware	Batch Size	milliseconds/shower
Geant4	CPU	N/A	1772
CaloGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012

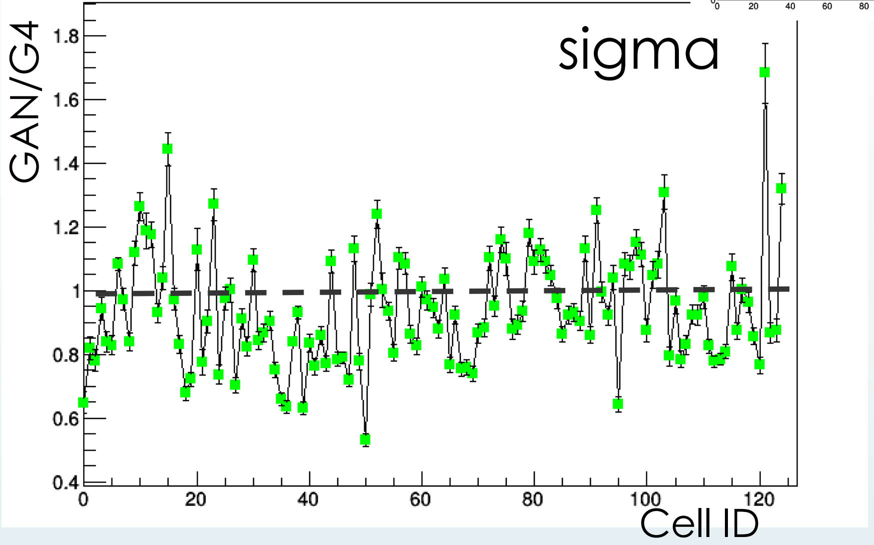
		Time/Shower (msec)
Full Simulation (G4)	Intel Xeon	
3d GAN (batchsize 128)	Intel i7 (laptop)	66
	GeForce GTX 1080	0.04

Single cell response

Preliminary







Cell energy sigma is underestimated by GAN

- Set up higher level criteria for image validation (reconstructed variables)
- Check uncertainty related to training sample statistics
- Compare to other fast sim approaches