Generative models for fast simulation

Sofia Vallecorsa

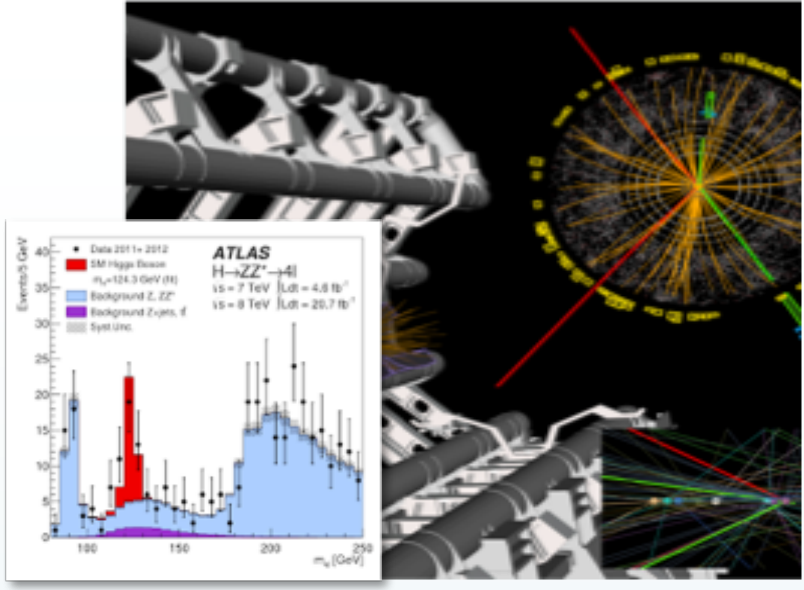
Gangneung-Wonju U. & CERN

OMA School on MC simulation
6-10 November 2017
Ludwig Maximilian University of Munich

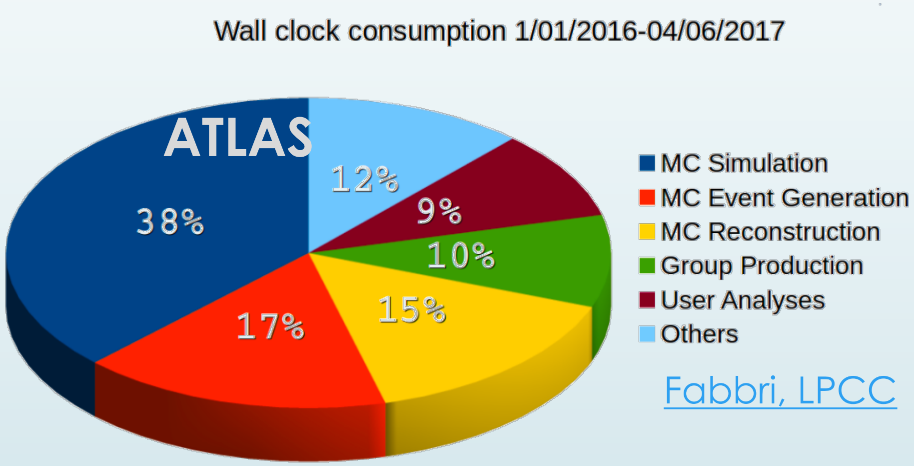
Outline

- 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

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



Fabbri, LPCC

WLCG
Worldwide LHC Computing Grid

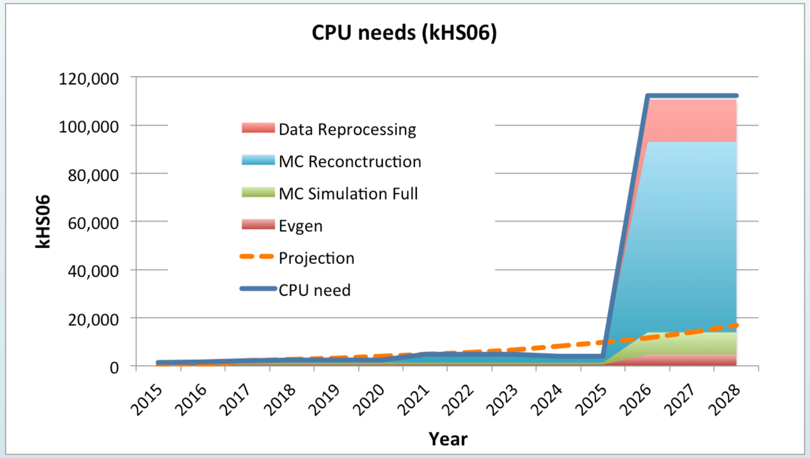
200 Computing centers in 20 countries: > 600k cores

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

The problem

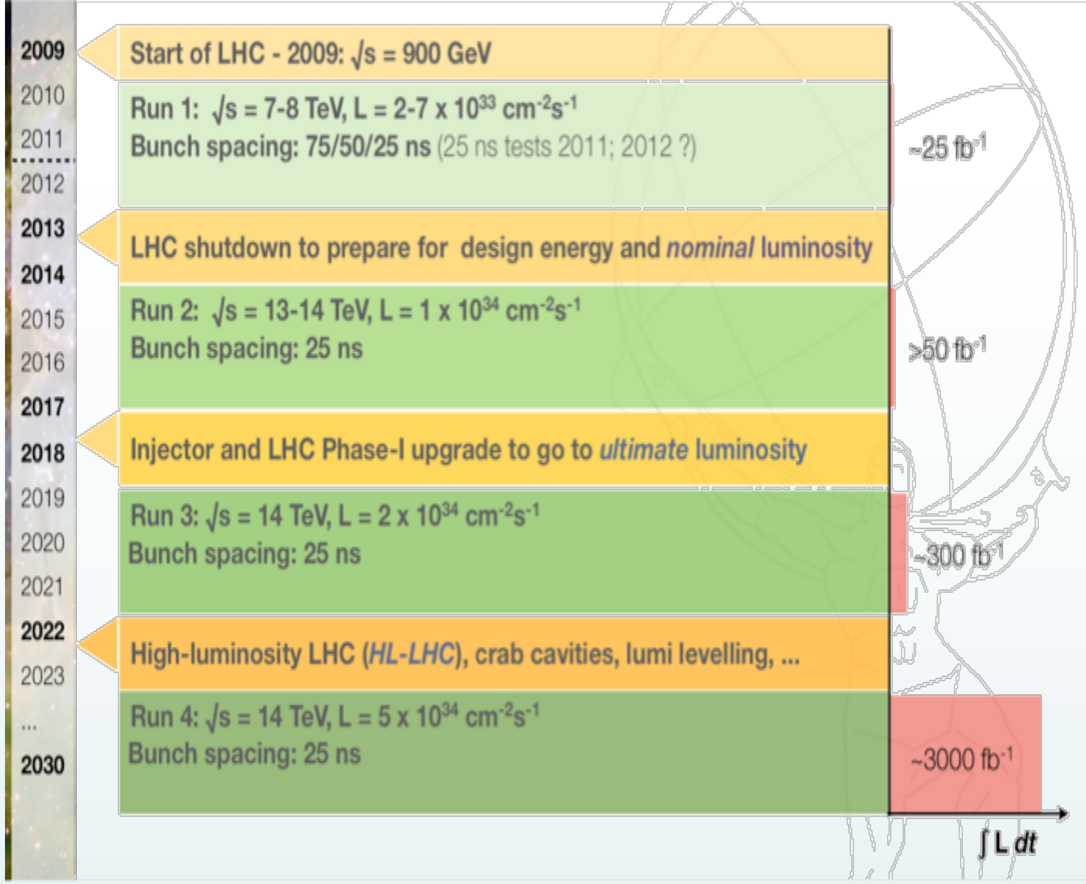
High Luminosity LHC

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



ATLAS computing needs

[Campana, CHEP 2016]



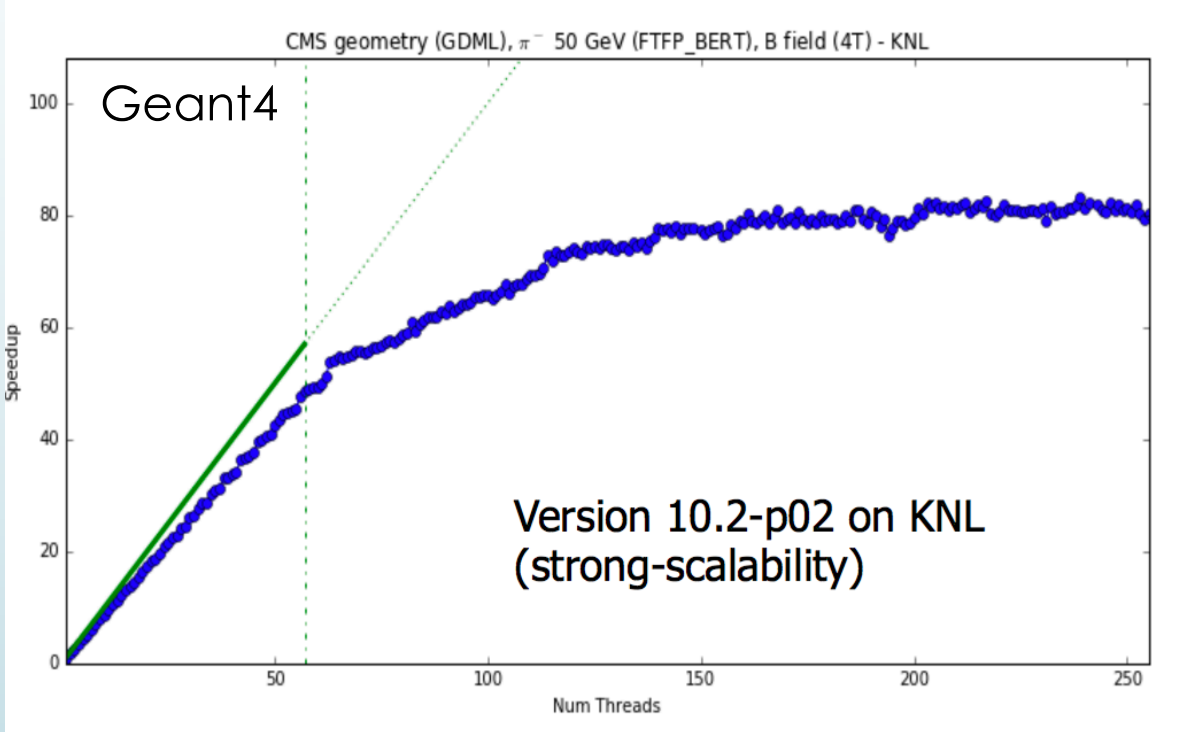
Other communities share similar needs:

- 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)

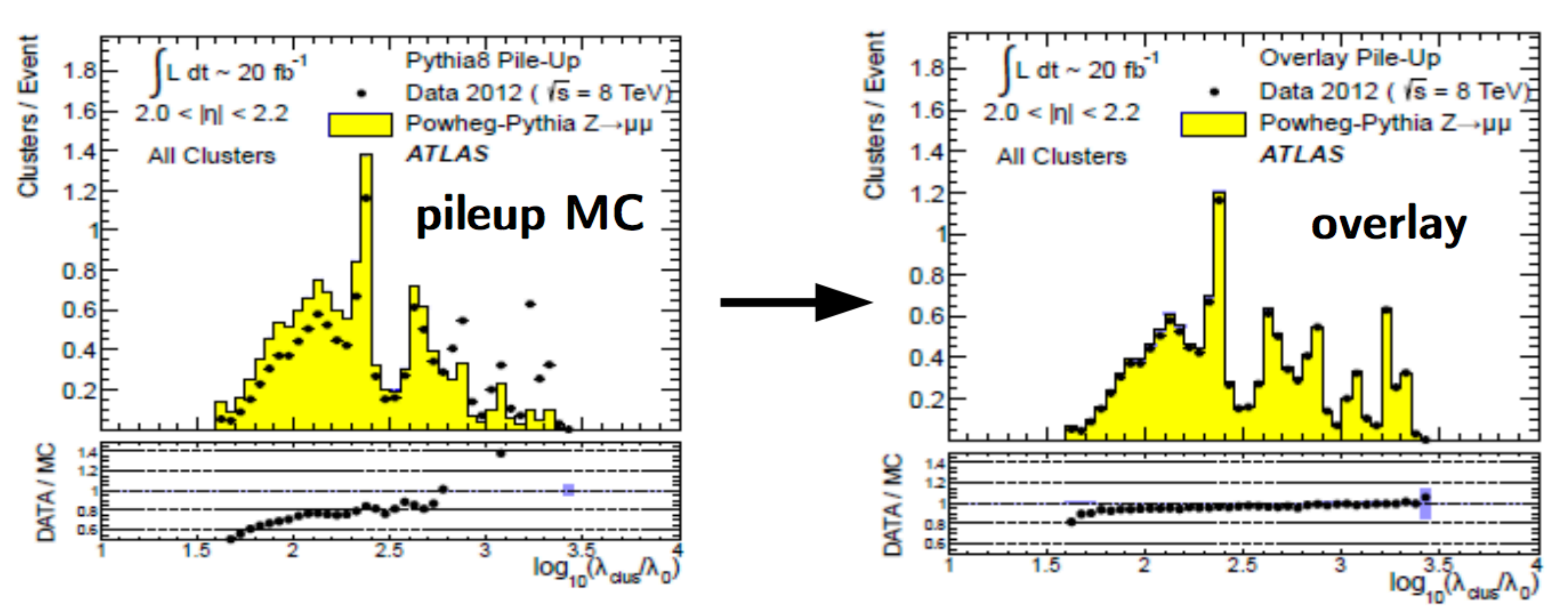


Event-level parallelism

Asai, LPCC

Speeding up simulation

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



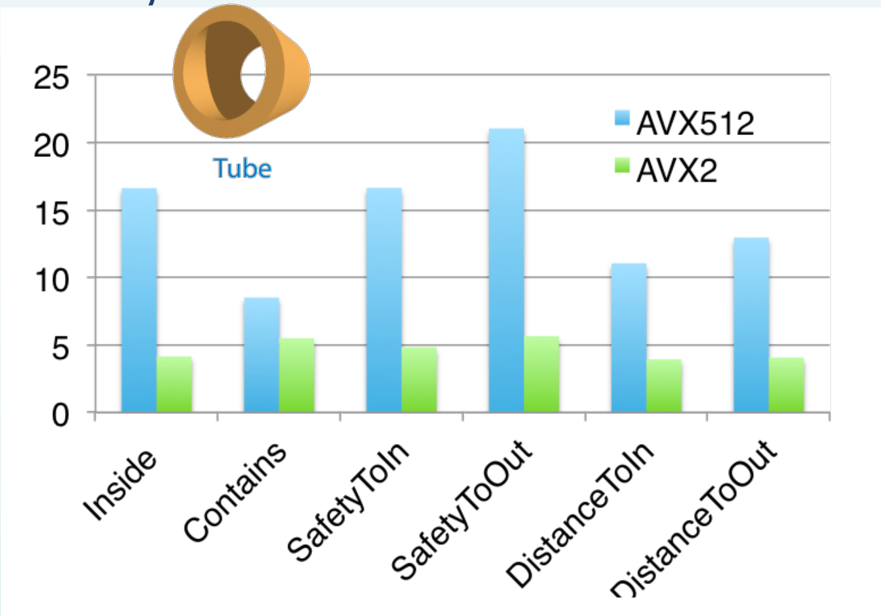
[Haas, CHEP 2016]()

see A: Gheata talk on GeantV

Speeding up simulation

- **Introduce fine grained parallelism**
- GEANTV aims at x5 total 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



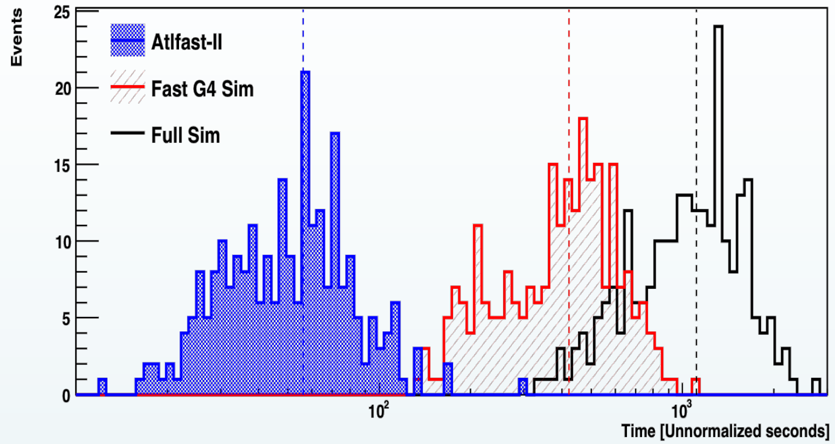
S.V., PASC17

Going beyond: Fast Simulation

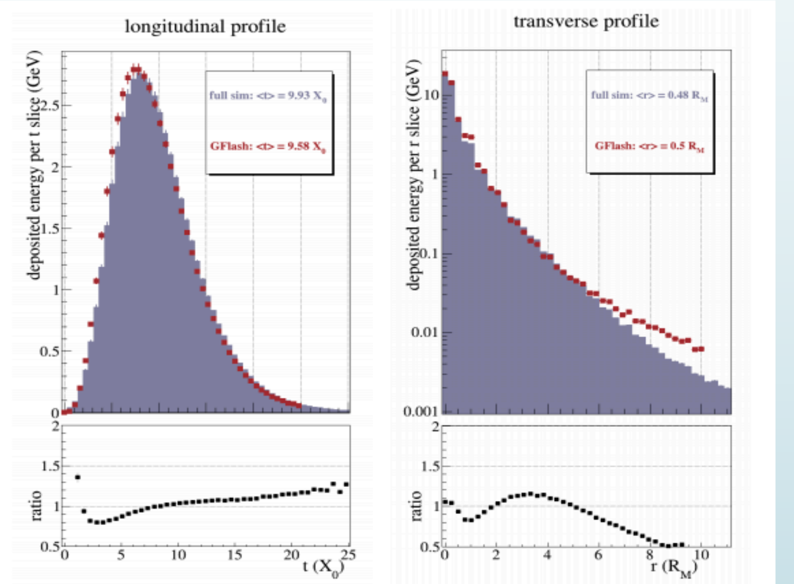
- Already used for searches, upgrade studies,…
- **Different techniques**
  - Shower libraries (pre-simulated EM showers, fwd calorimeters in ATLAS/CMS)
  - Shower shapes parametrizations (GFlash,..)
  - Fast trackers simulation (ATLAS FATRAS, ..)
  - Look-up tables
  - Fully parametrized simulation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

Choice is "experiment" dependent!

ATLAS



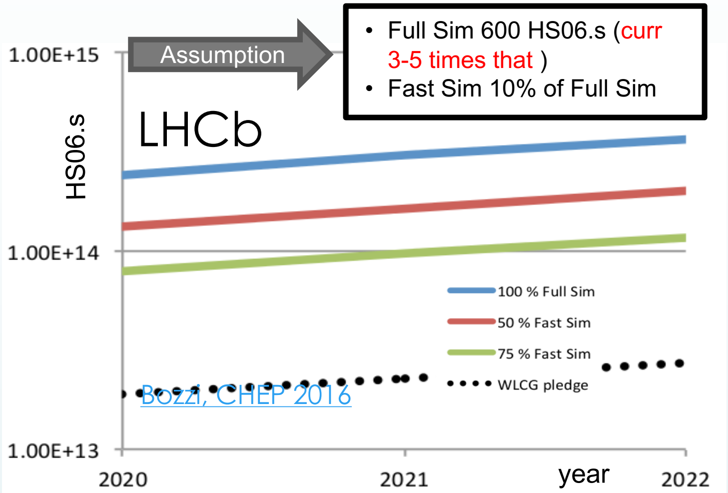
FCChh



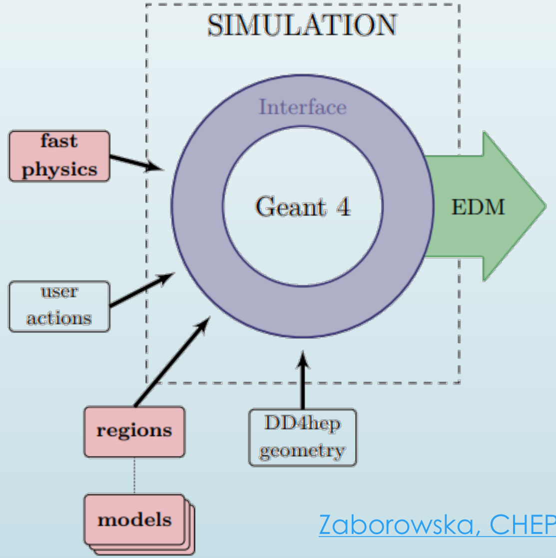
Zaborowska, CHEP2016

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 to develop a generic fully customizable fast sim framework
  - Deep Learning based



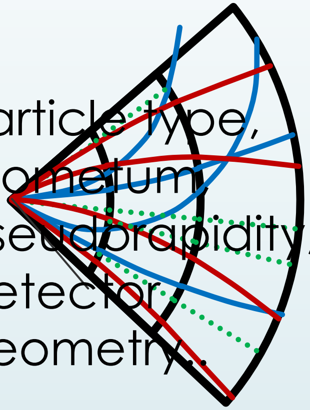
FCC Gaudi framework



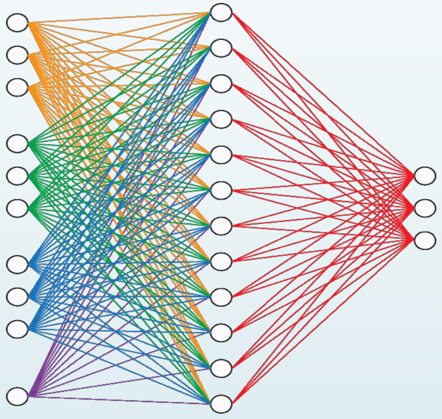
Zaborowska, CHEP2016

Deep Learning for fast sim

EX. SIMULATION OF A CALORIMETER

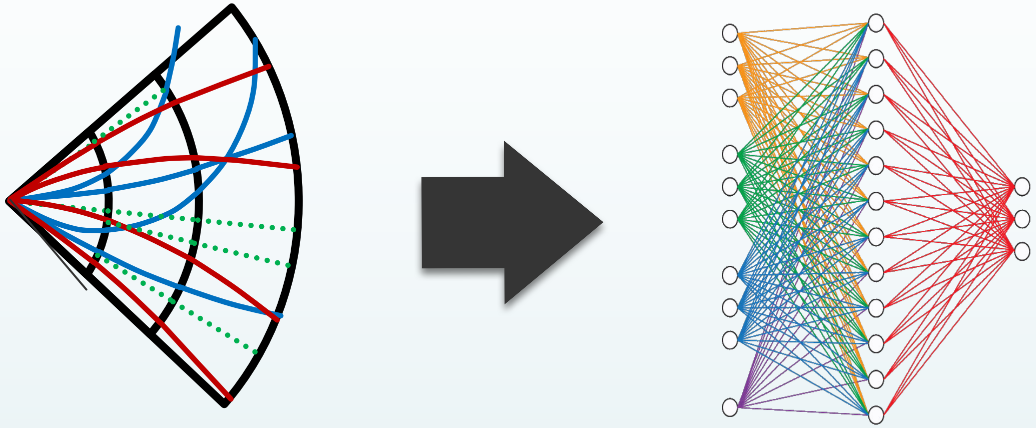


Particle type, momentum, pseudorapidity, detector geometry..



Energy depositions in cells

Deep Learning for fast sim



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

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

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

choose parameters that maximize the likelihood training data

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

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

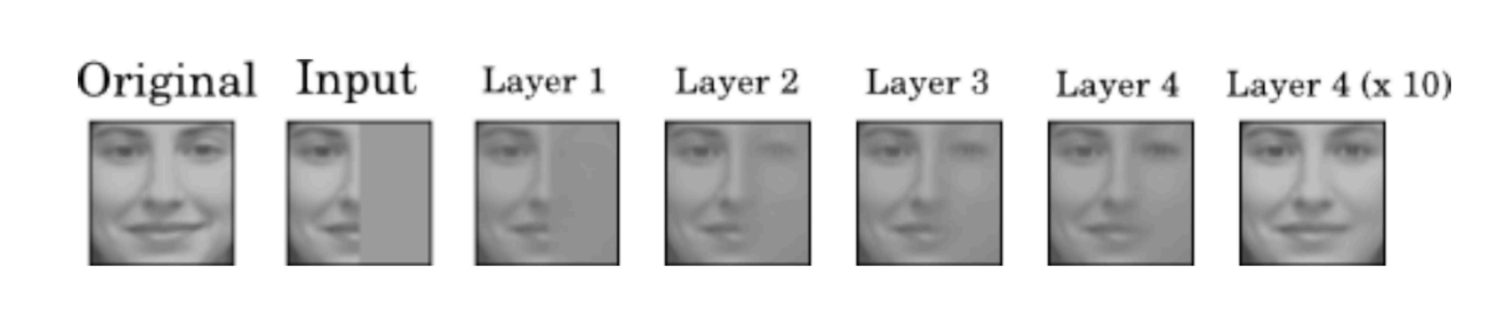
arXiv:1701.00160v3 9 Jan 201

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:

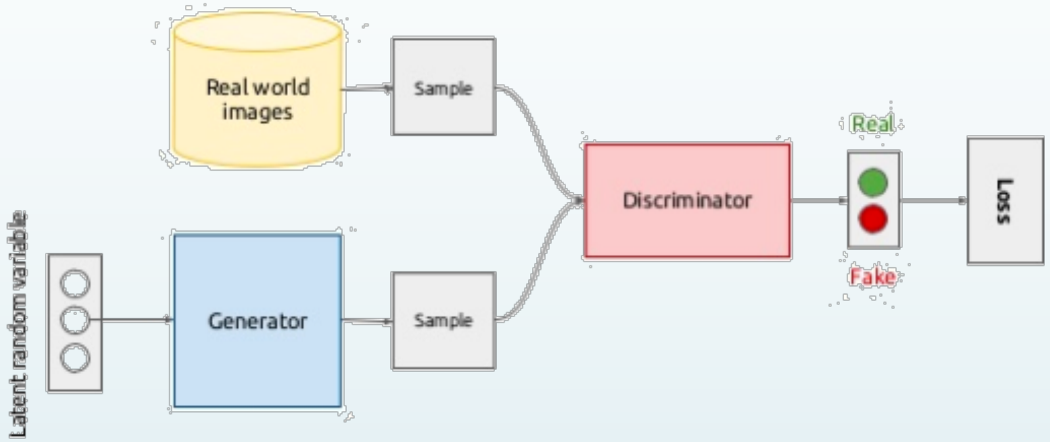
Can imaging approaches be useful?

- 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?

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

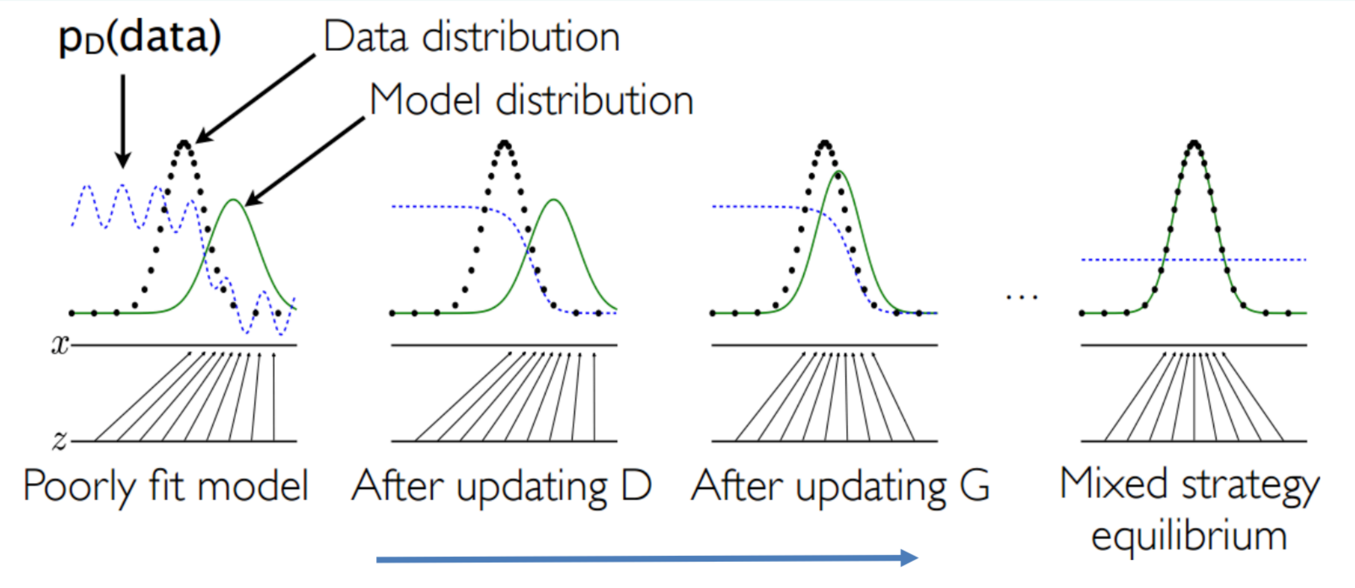
arXiv:1406.2661v1

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

GAN application examples

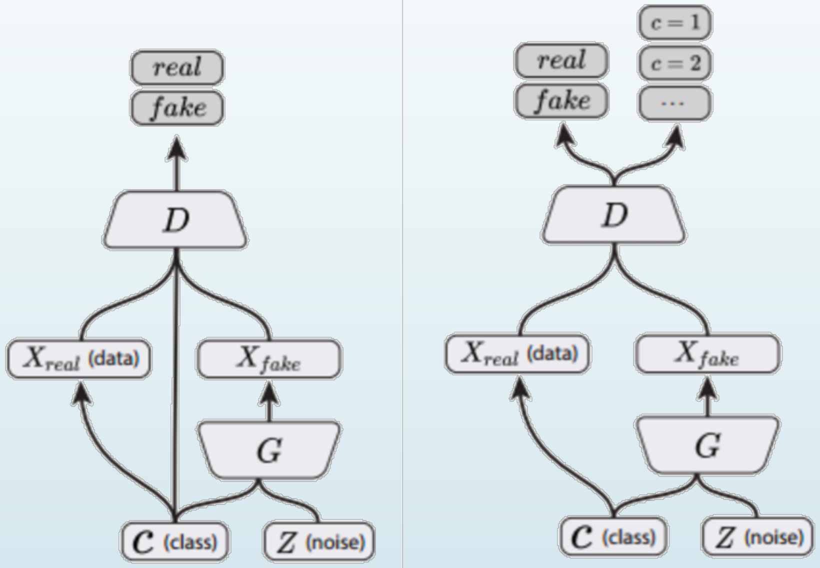
Samples of images of bedrooms generated by a DCGAN trained on the LSUN dataset.

https://arxiv.org/pdf/1701.00160v1.pdf

Samples drawn trained on the CIFAR-10 dataset

Many GAN flavors

- Original GAN was based on Multi Layer Perceptrons in 2014
- Deep Convolutional GAN in 2015
- Conditional GAN
  - Extended to learn a parameterized generator $p_{model}(x|\theta)$;
  - 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

Convolution layers

- Images can be considered as a matrix of pixel values
- **Convolutions** extract features from the input image using small squares of input data
  - preserve spatial relationship between pixels.

- **Input**: 5 x 5 image
- **Filter**: 3 x 3 matrix
- Slide the filter ouput matrix element
  1. element wise multiplication
  2. Sum of the multiplication outputs

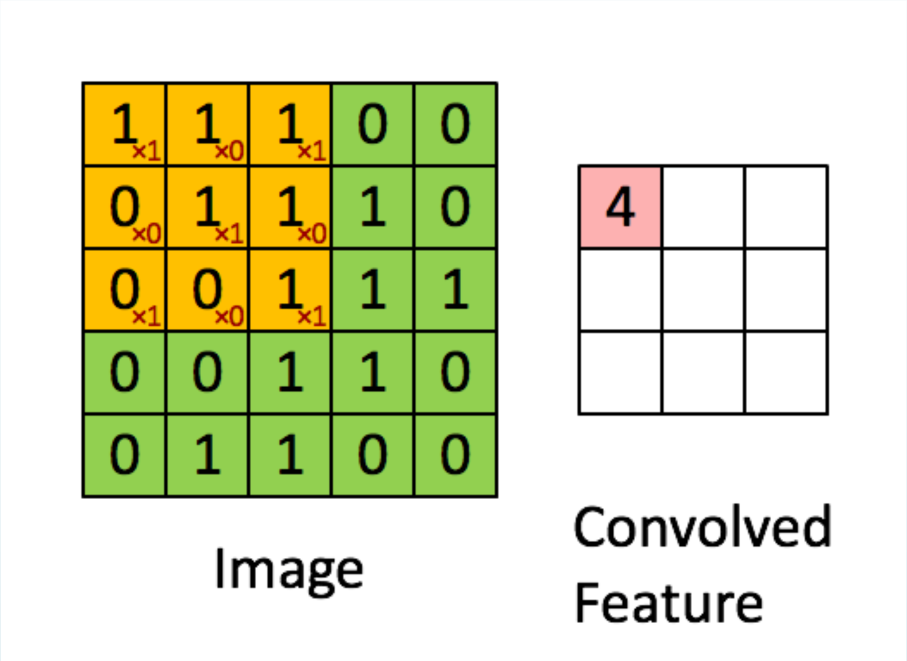
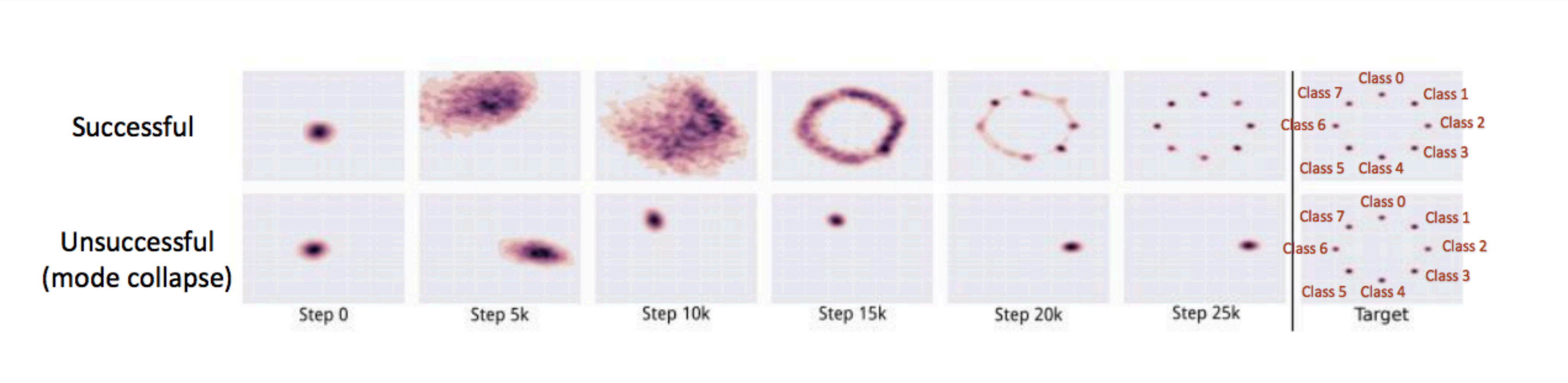


Image source

Common GAN problems

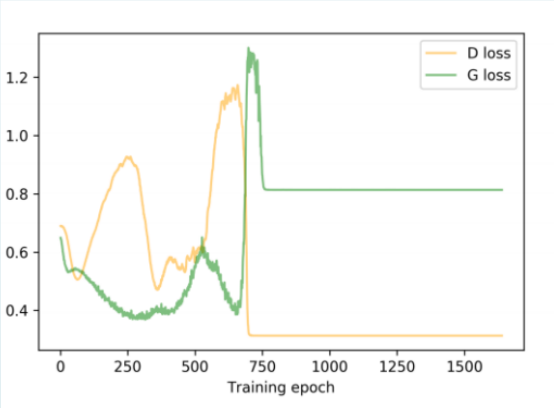
Collapse Mode:

- Goal of GAN: To generate fake examples imitating real samples
- Easy way of achieving goal: Just generate easy modes (classes).



Vanishing/Exploding gradients

- The representational power (or capacity) between discriminator and generator is not balanced



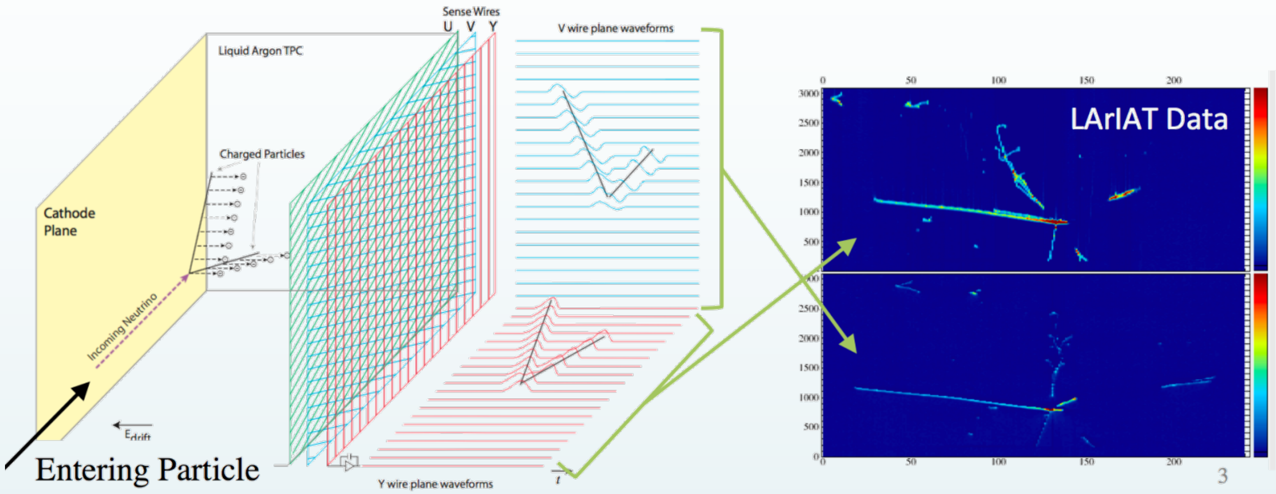
Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks (2016).

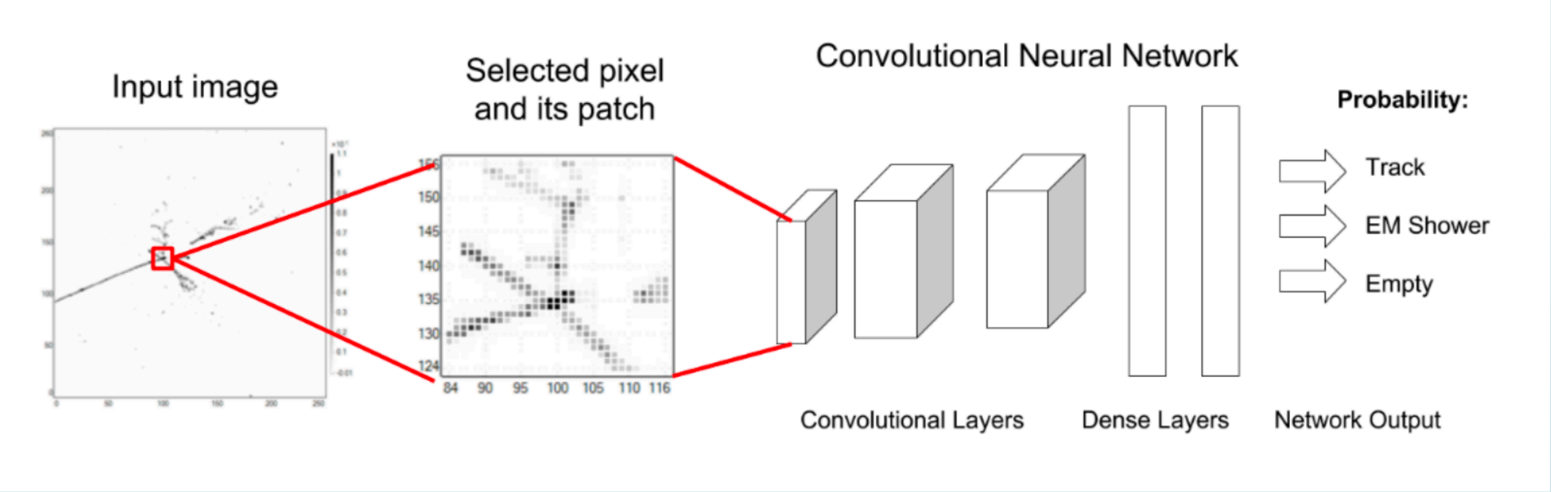
Applications

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

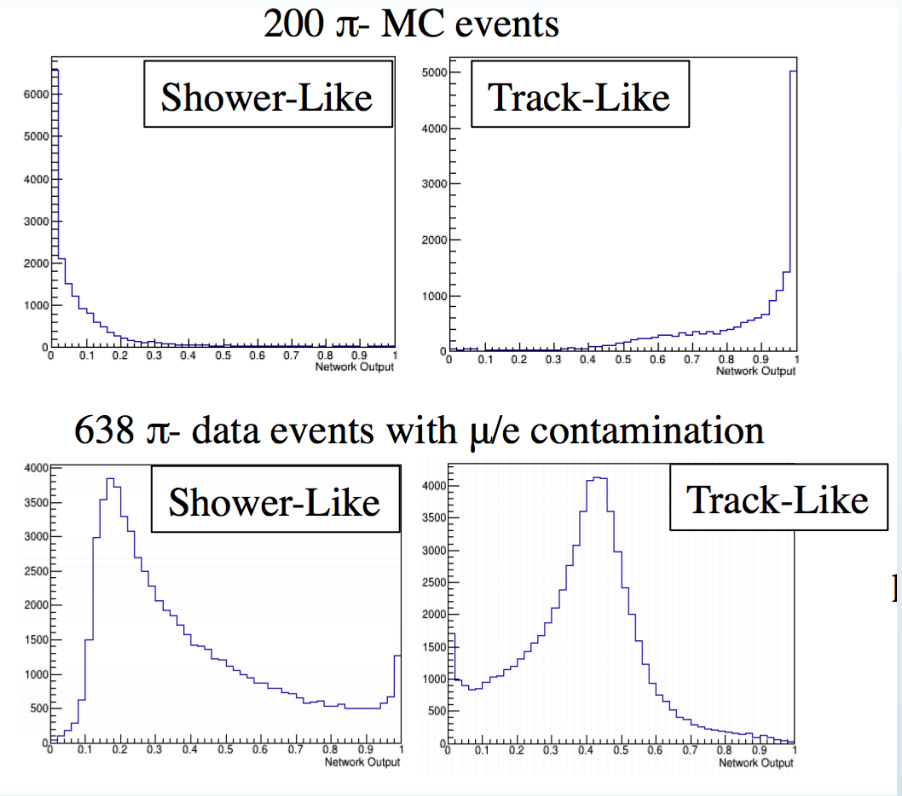




Enhancing MC simulation with GAN

- MC shows good separation as expected however,
- But the method does not work on data!

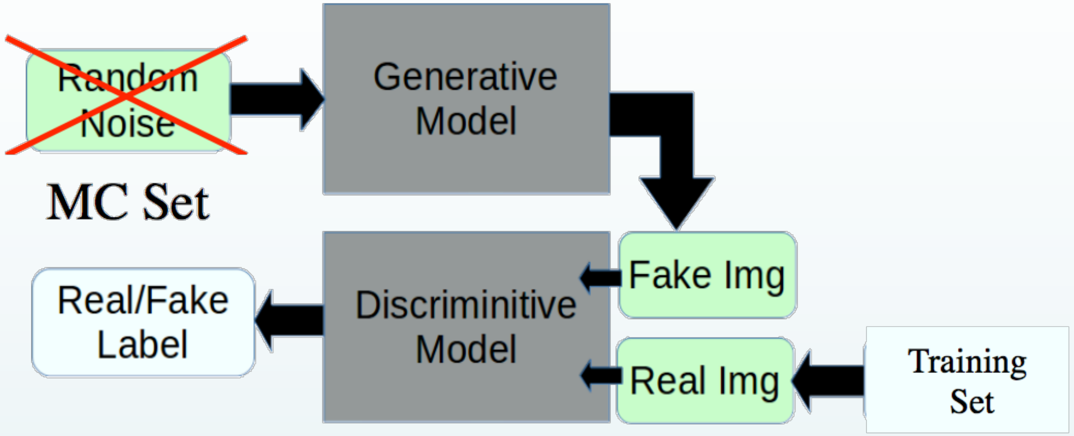
- Difference between MC and data affects the network in unpredictable ways
  - wire-to-wire cross-talk
  - wire-to-wire inductance,
  - physics of wire charge deposition range, electronic noise,
  - wire-to-wire variance are all not simulated in MC

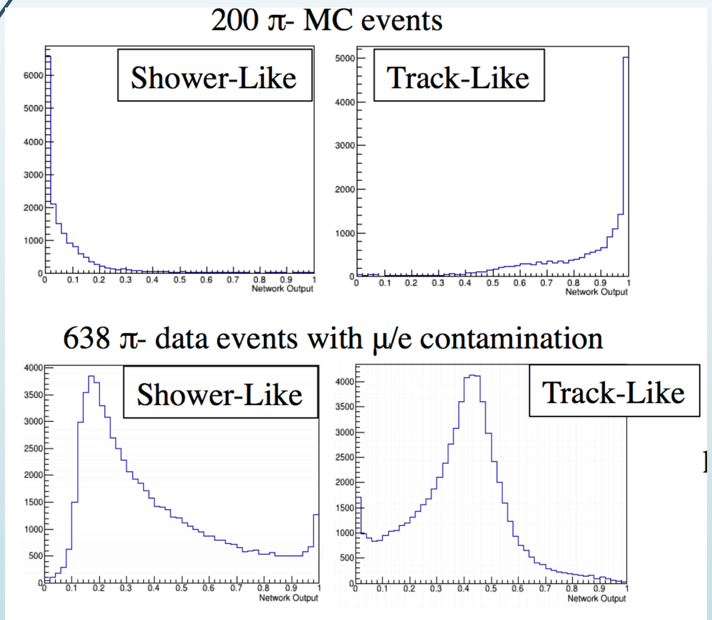


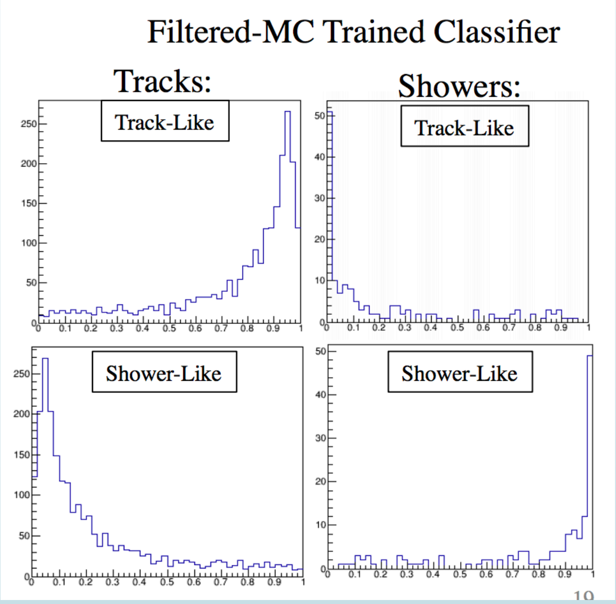
Enhancing MC simulation with GAN

[Smith, IML workshop](#)

- Pass a MC sample to a GAN generator
- Training against data will create a data-driven filter for MC
- A filtered MC sample that is very similar to data



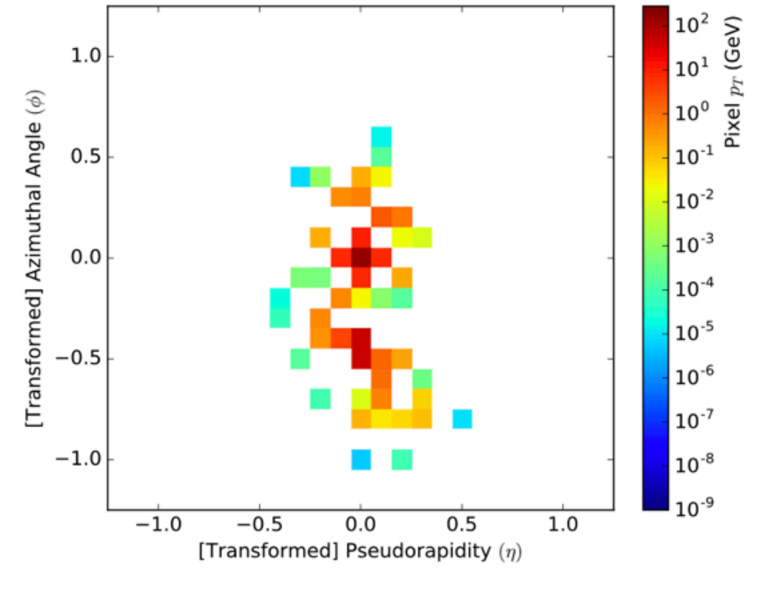


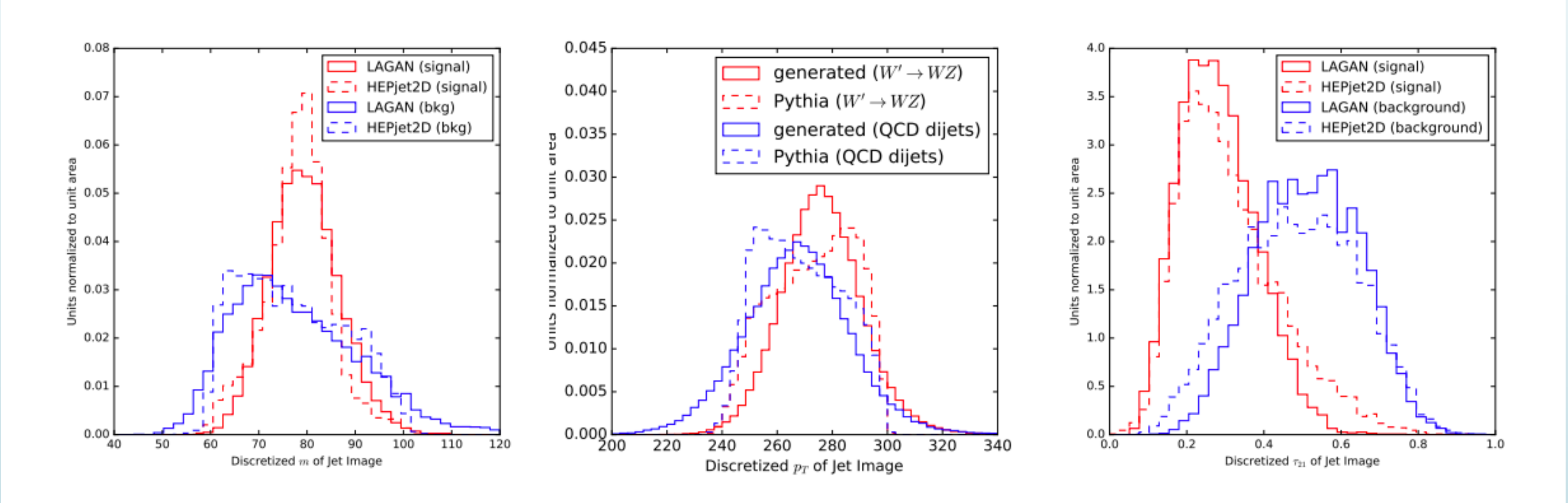


[arxiv:1701.05927](https://arxiv.org/abs/1701.05927)

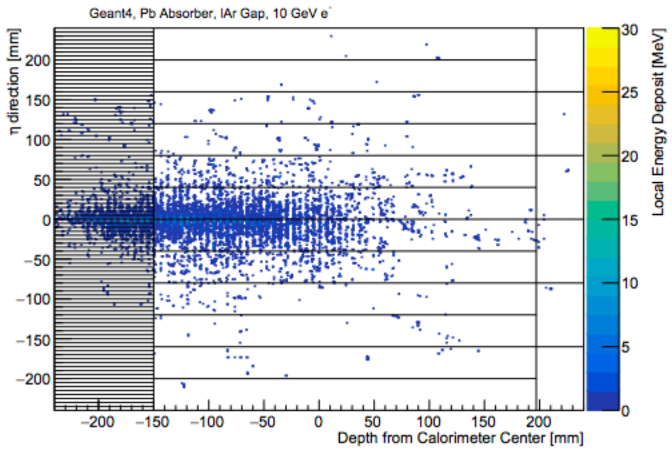
Location Aware GAN

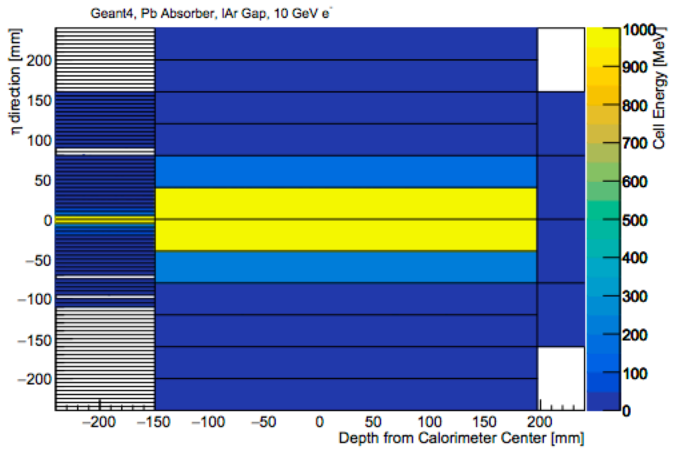
- Reproduce 2D generator level anti-kT jet images (generator-level study)
- Modification of DCGAN (convolutions) and ACGAN (uses particle type information)
- Image sparsity
- Location dependent features
- Large dynamic range



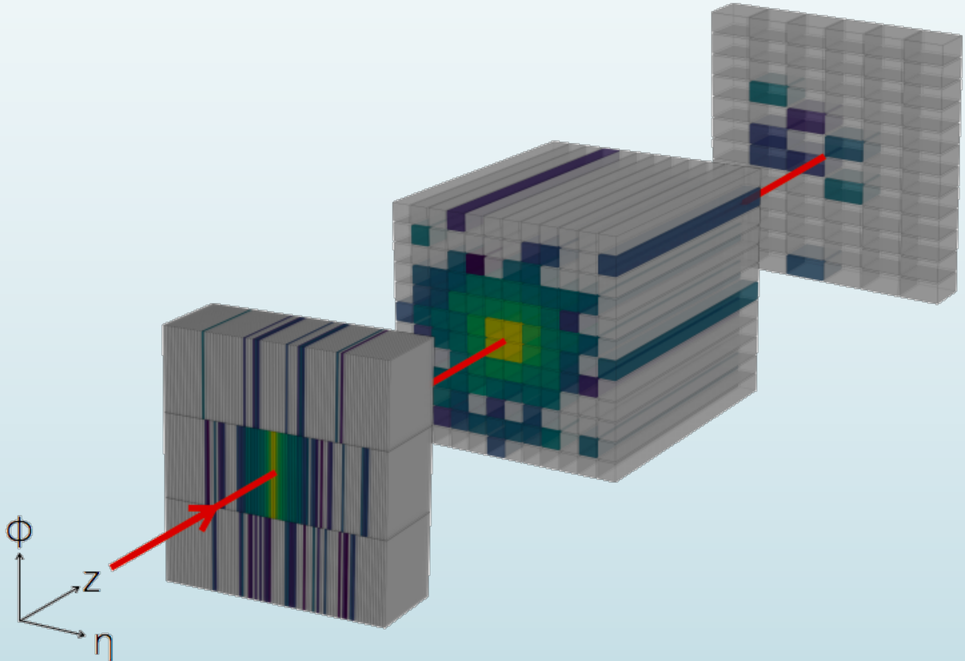


CaloGAN



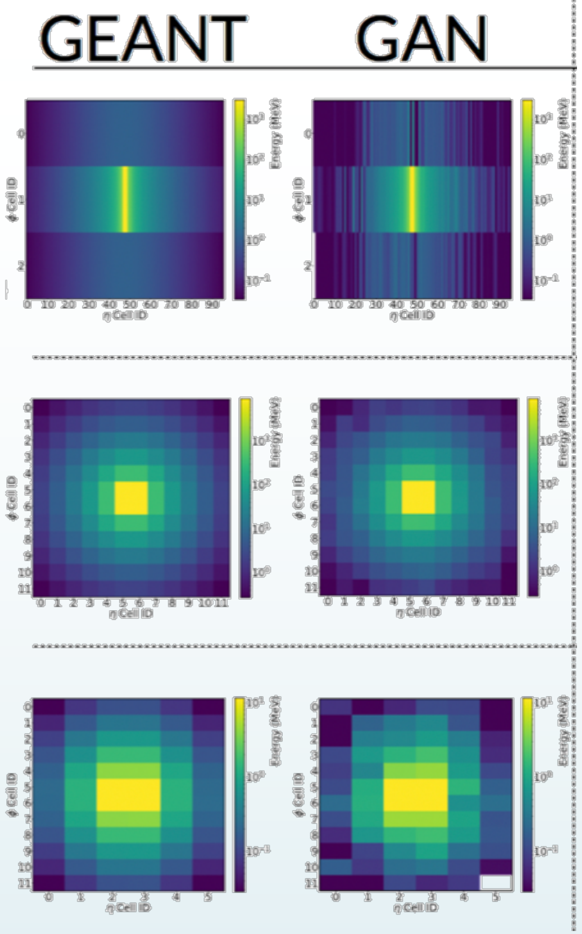


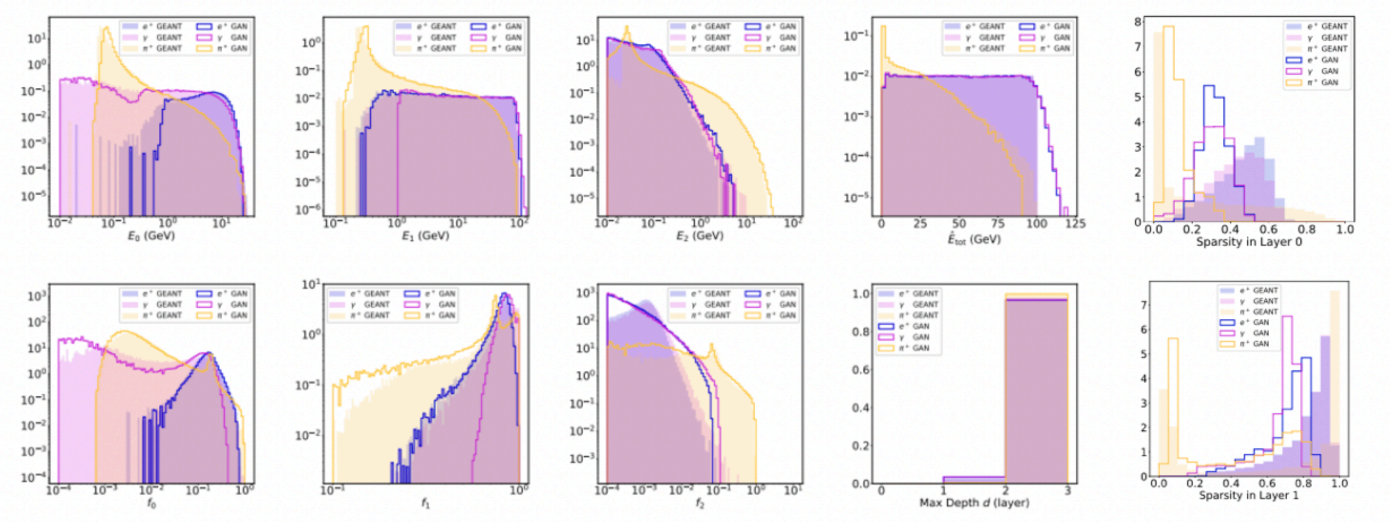
- ATLAS LAr calorimeter
  - Heterogeneous longitudinal segmentation into 3 layers
  - Irregular granularity in eta and phi
- Energy deposition in each layer as a 2D image
- Build one LAGAN per layer
- Trainable transfer unit to preserve layer correlations
- Result is a concatenation of 2D images that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

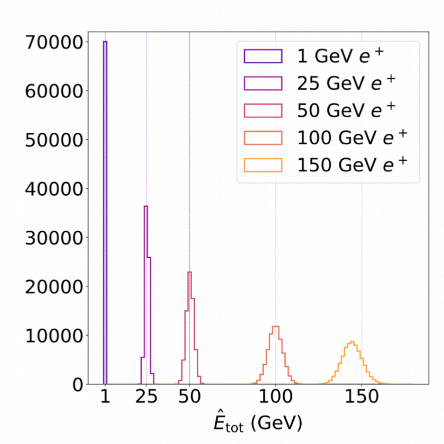


CaloGAN performance

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

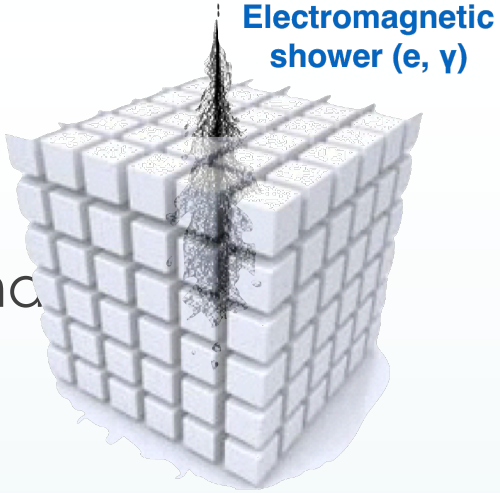






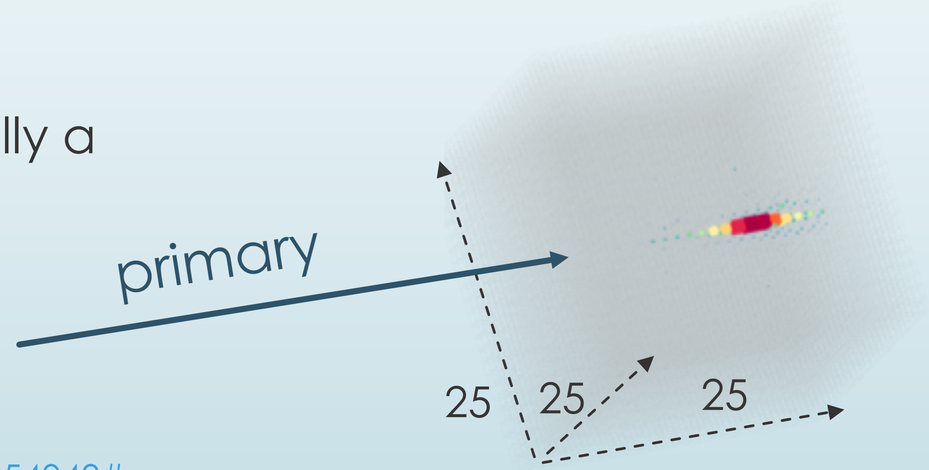
GeantV GAN

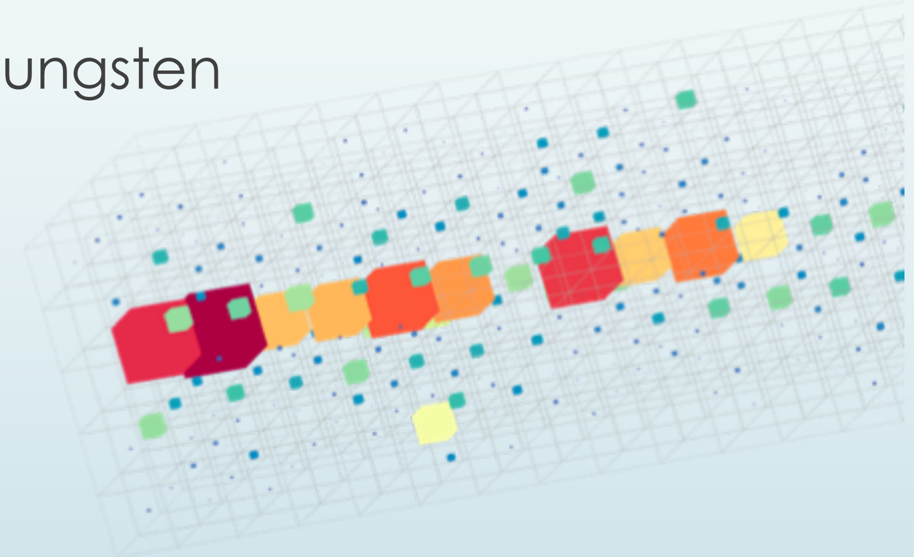
CLIC calorimeter data



- CLIC is a CERN project for a linear accelerator of electrons and positrons to TeV energies
- Associated electromagnetic calorimeter detector design(*)
- A highly segmented array of absorber material and silicon sensors
  - 1.5 m inner radius, 5 mm×5 mm segmentation: 25 tungsten absorber layers + silicon sensors

Data is essentially a 3D image

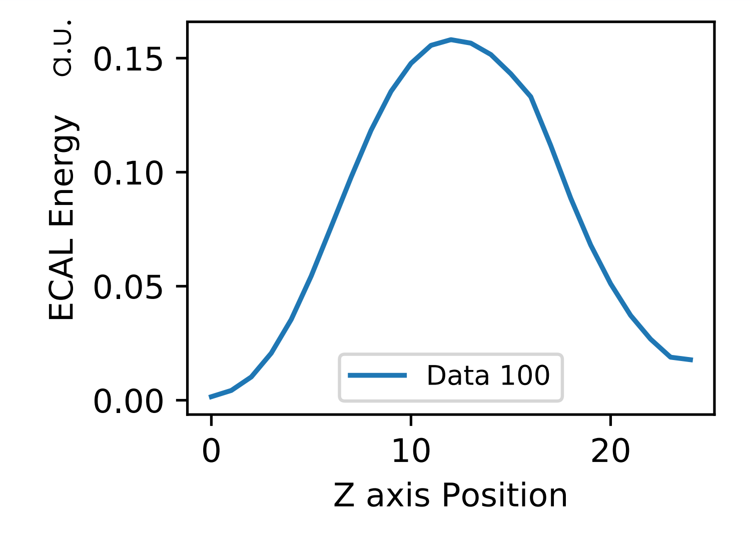


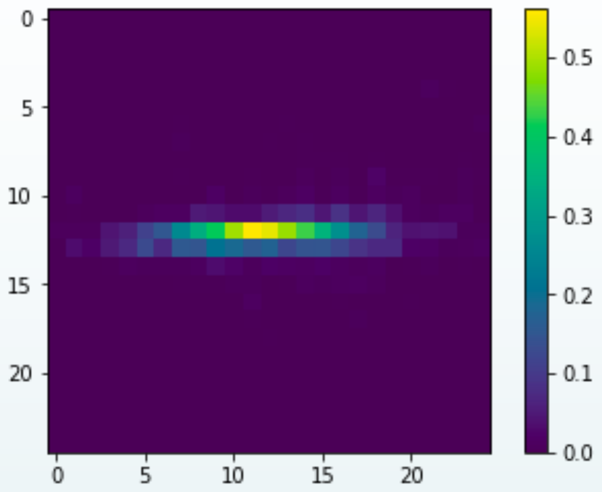


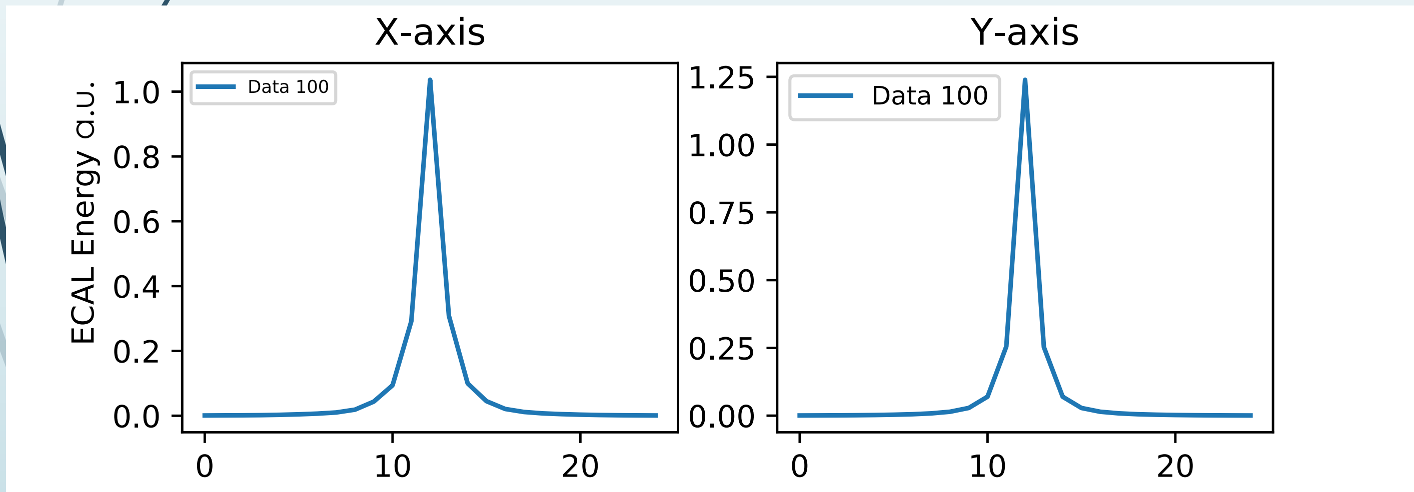
(*) http://cds.cern.ch/record/2254048#

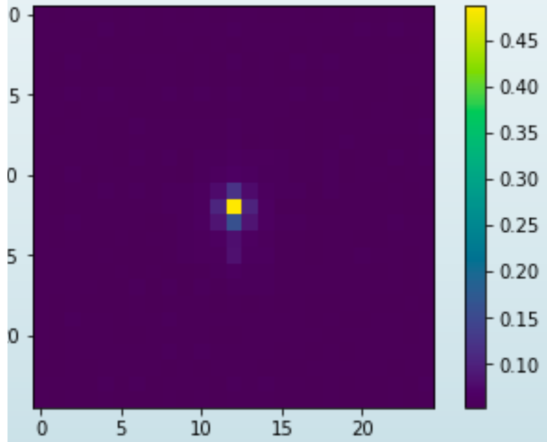
CLIC calorimeter data

- Highly segmented (pixelized)
  - Segmentation is critical for particle identification and energy calibration.
- Sparse.
- Non-linear location-dependency



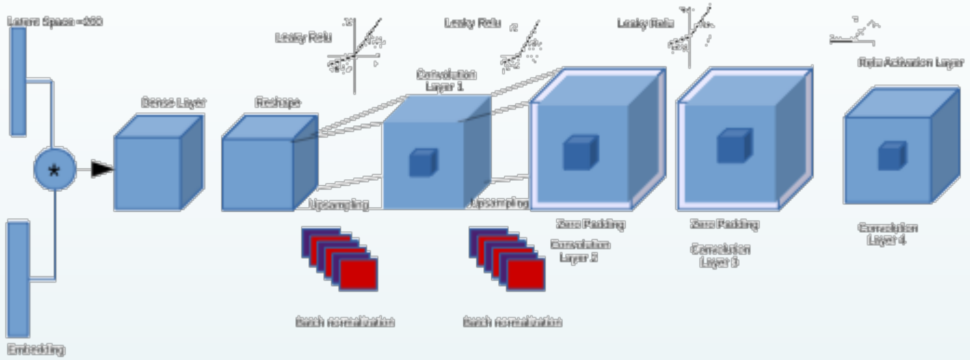




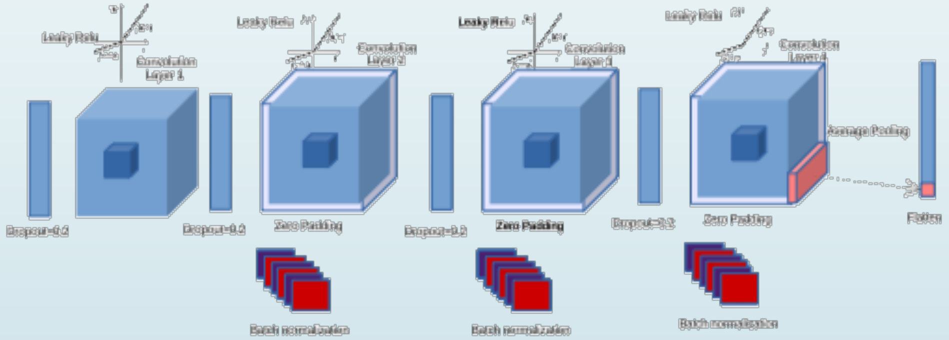


GeantV GAN for calorimeter images

- 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

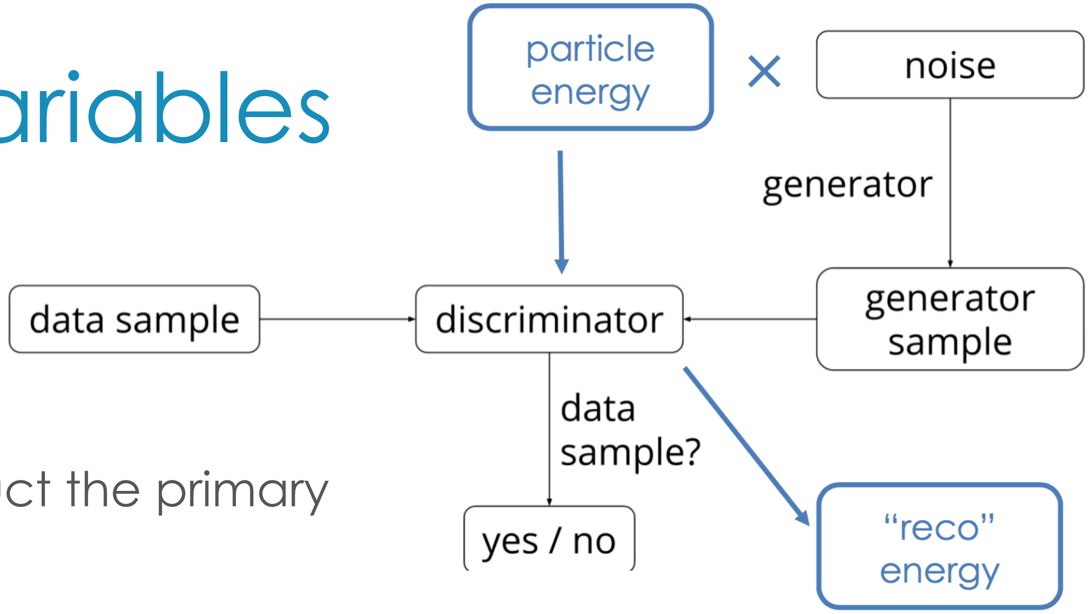


GENERATOR



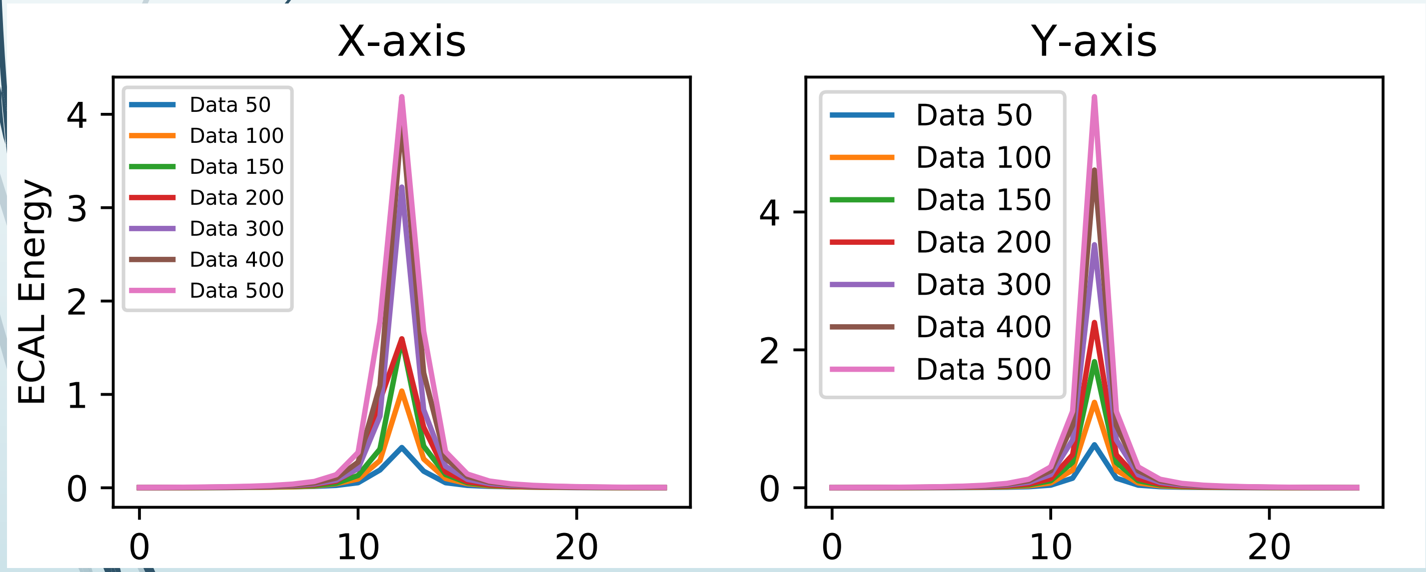
DISCRIMINATOR

Conditioning on additional variables



Training the generator and the discriminator using initial particle energy

- Add a regression task to the discriminator to reconstruct the primary particle energy
- Train the generator to reproduce correct shapes



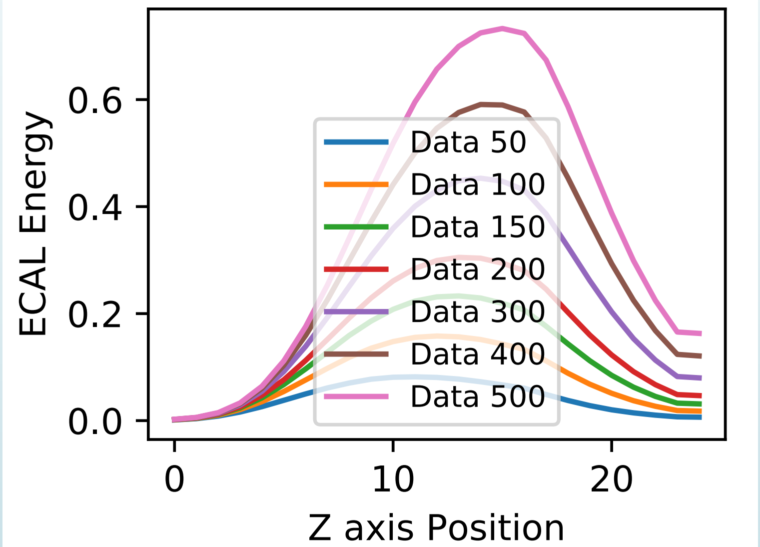


Image quality assessment and validation

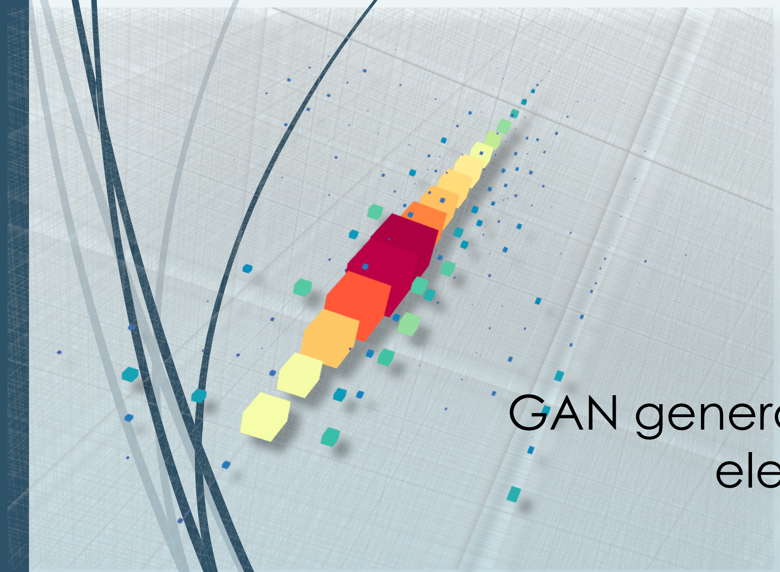
- Detailed study of calorimeter response
  - Energy distribution in single cells
- Average shower shapes
- Primary particle energy estimation from discriminator
- High level variables (e.g. jet features)
- Does analysis tools performance change if we replace detailed simulation with GAN generated data? (e.g. particle identification algorithms)

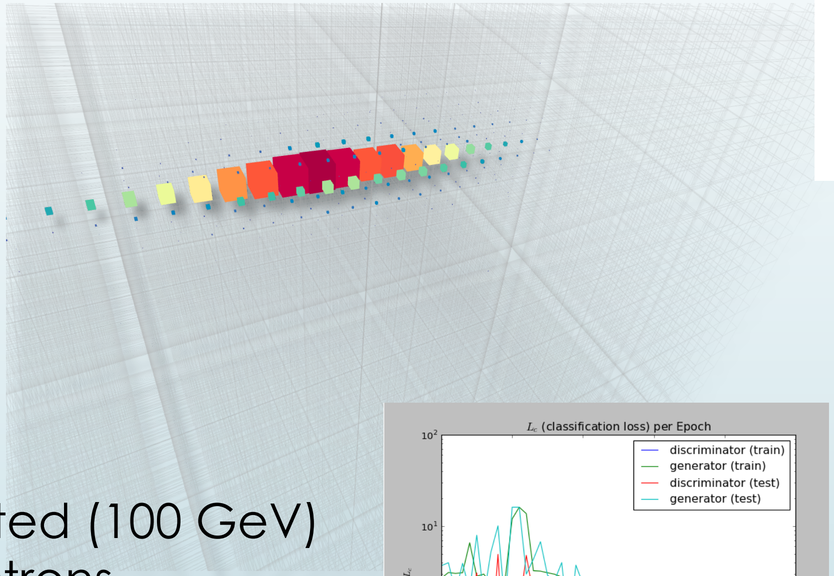
WORK iN PROGRESS

Comparison to full sim and different fast sim tools

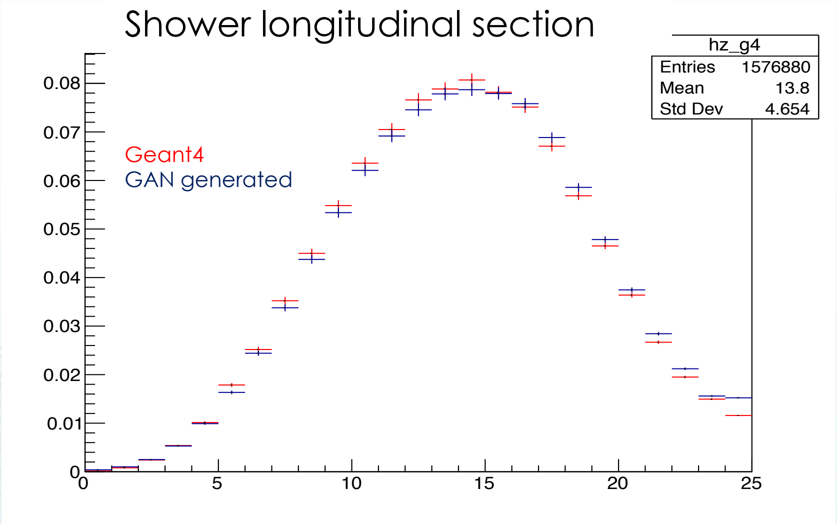
First 3D images

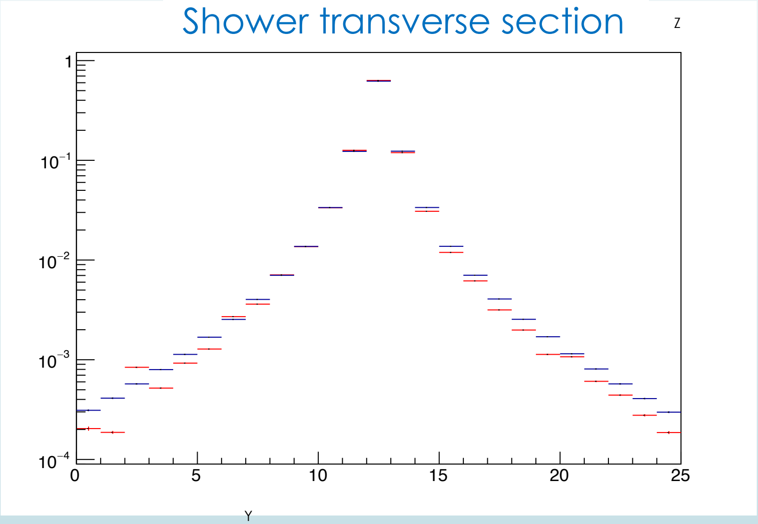
- First generated results look promising!
- Qualitative results show no collapse problem

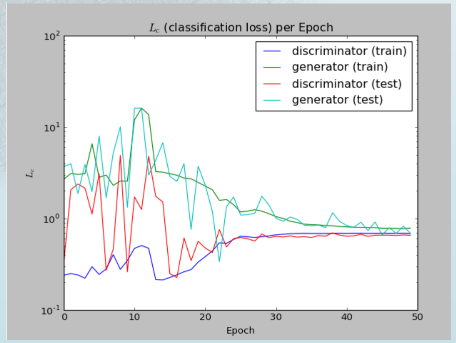




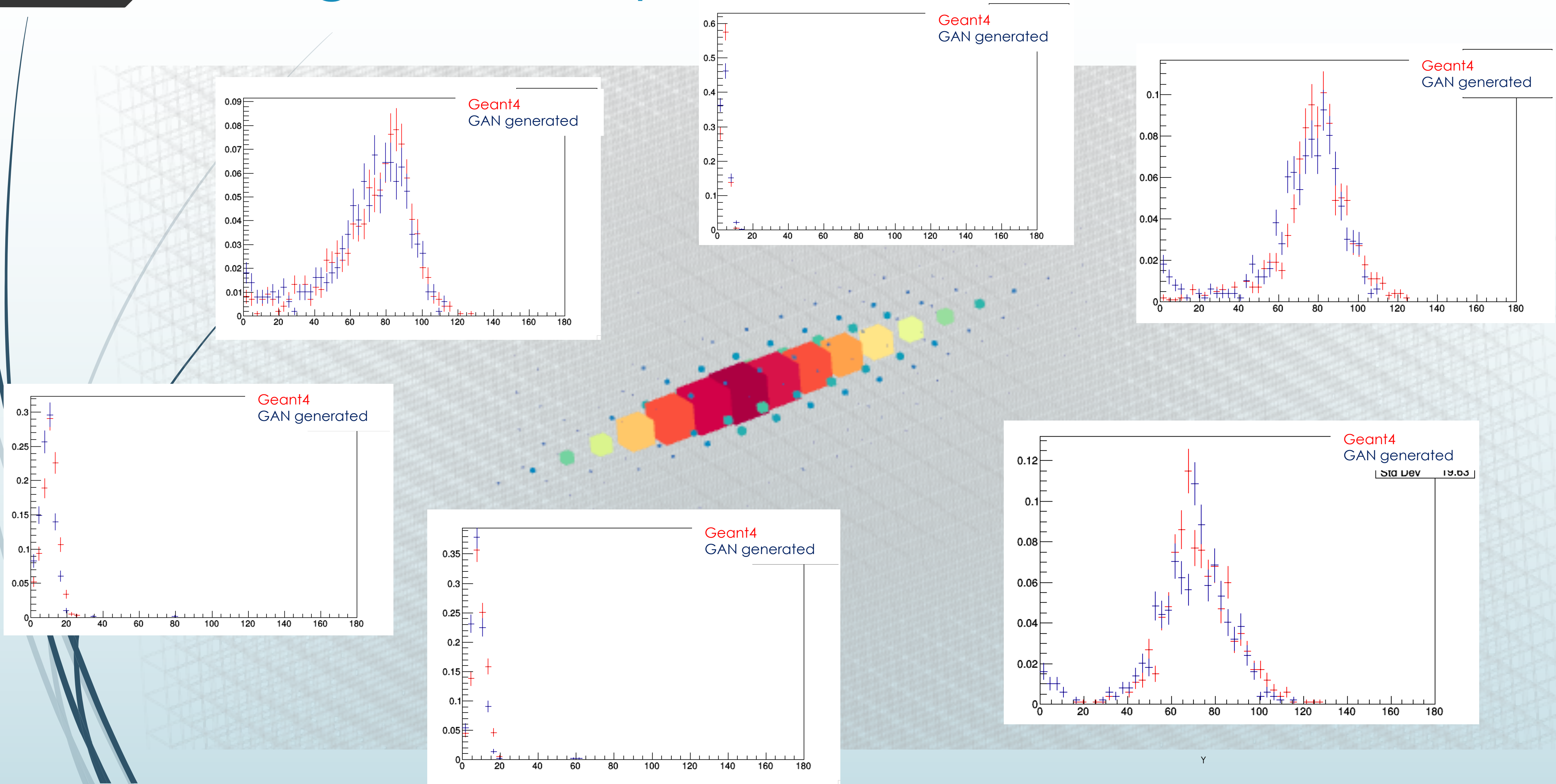
GAN generated (100 GeV) electrons



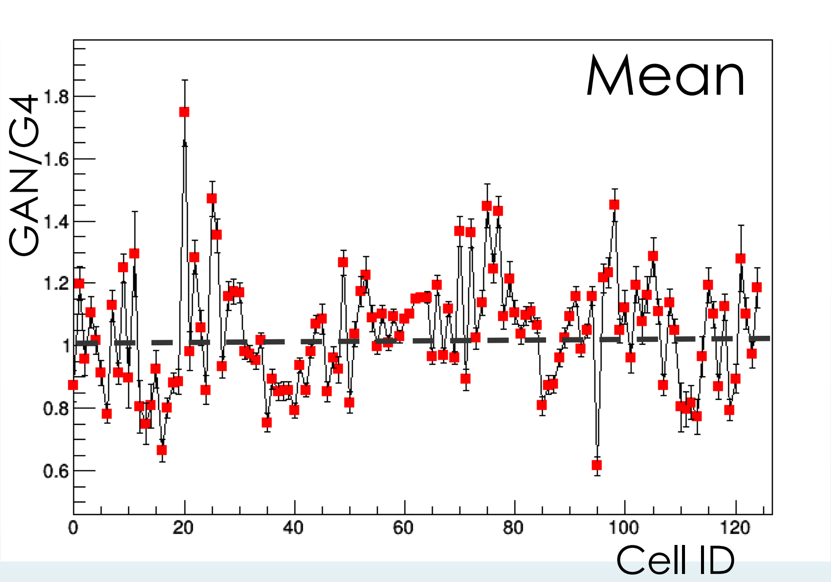


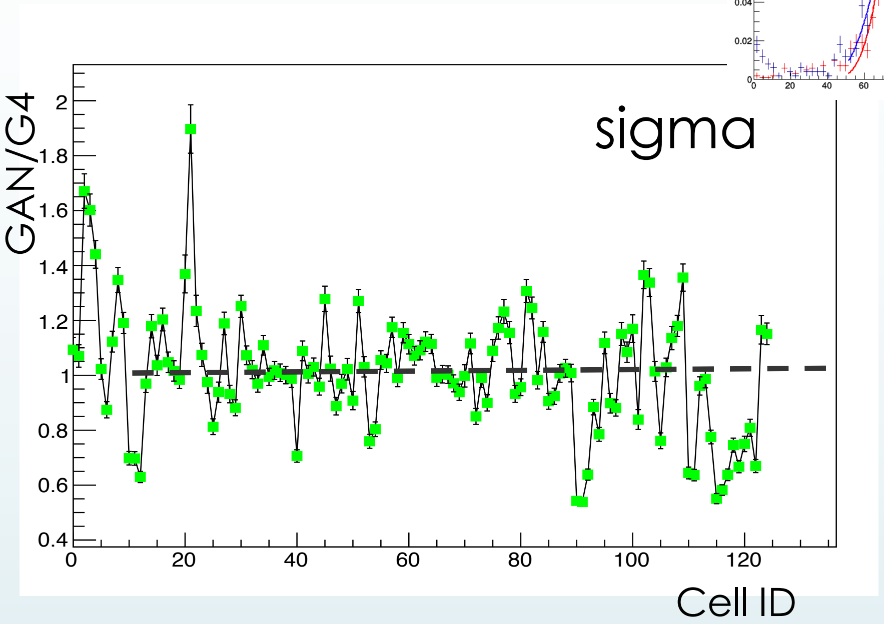


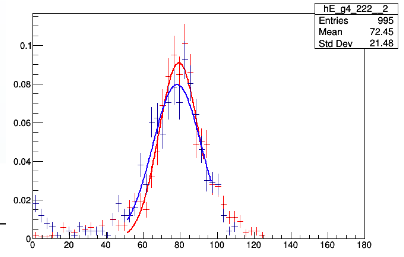
Single cell response



Single cell response







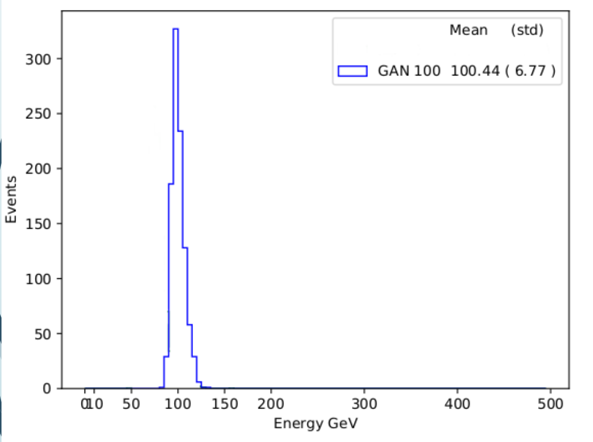
Single cell response is not perfect

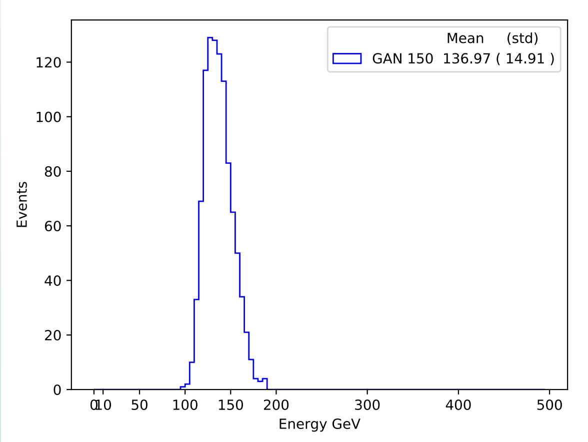
- Set up higher level criteria for image validation (reconstructed variables)

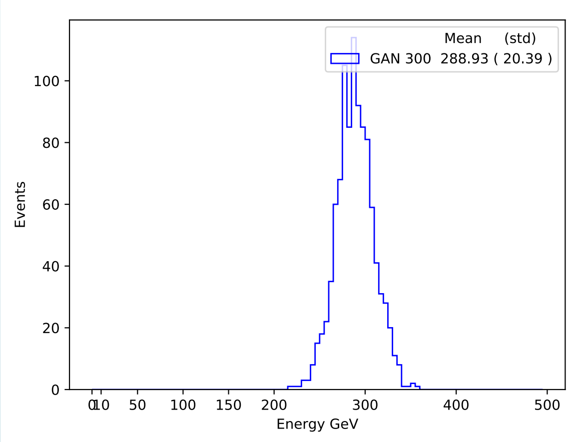
Energy regression test

- Train the network on a uniform energy spectrum (100-500) GeV
- Test the capability of the discriminator to correctly predict the primary particle energy.
- This is an additional regression task.
- Not the typical simulation use case

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

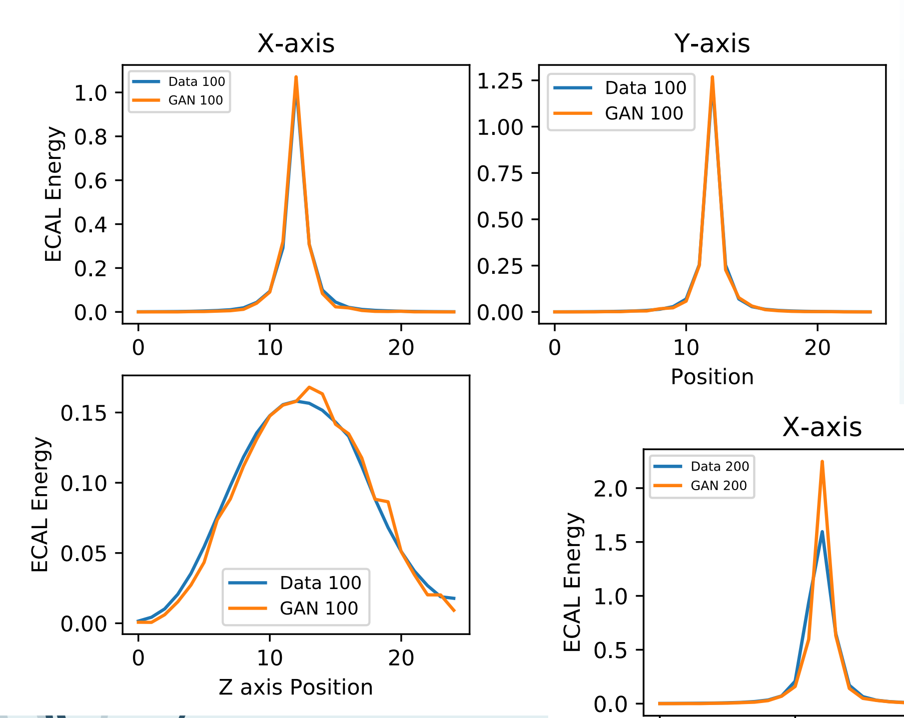


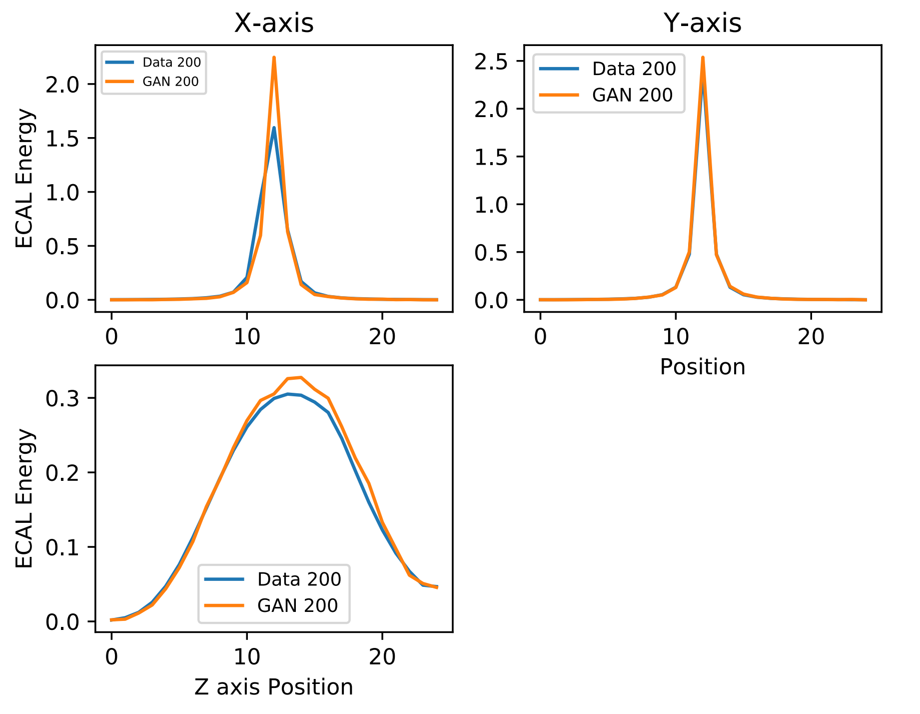


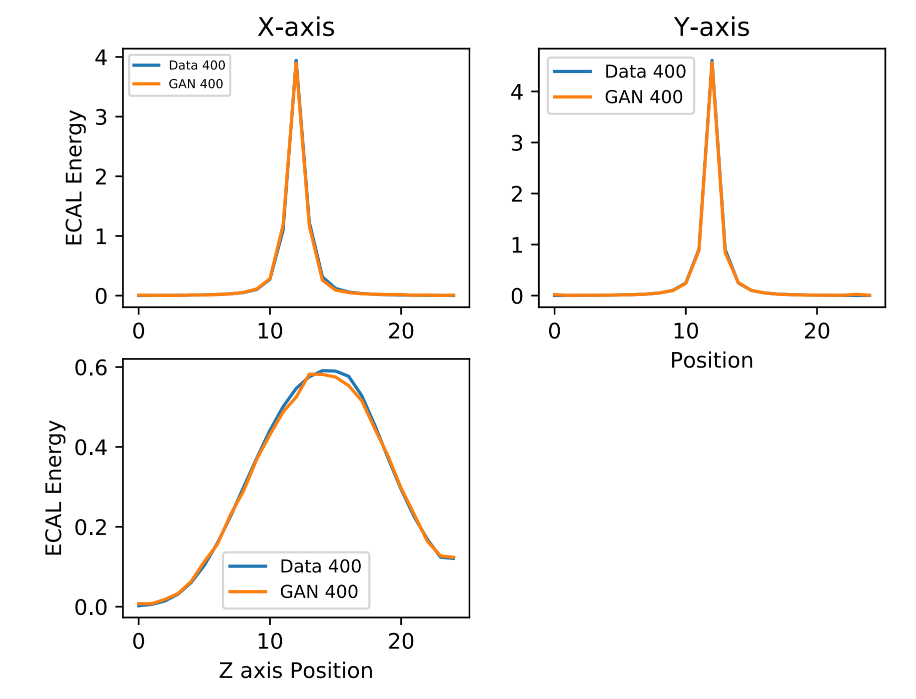


Energy shower shapes

- Check that the networks correctly describes the energy shapes for different input energies (generator output)







From the computing resources perspective…

Inference

- Using a trained model is very fast
  - Orders of magnitude faster than detailed simulation
  - Even on a simple laptop!

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

We are testing performance on FPGA and new integrated accelerator technologies

Training

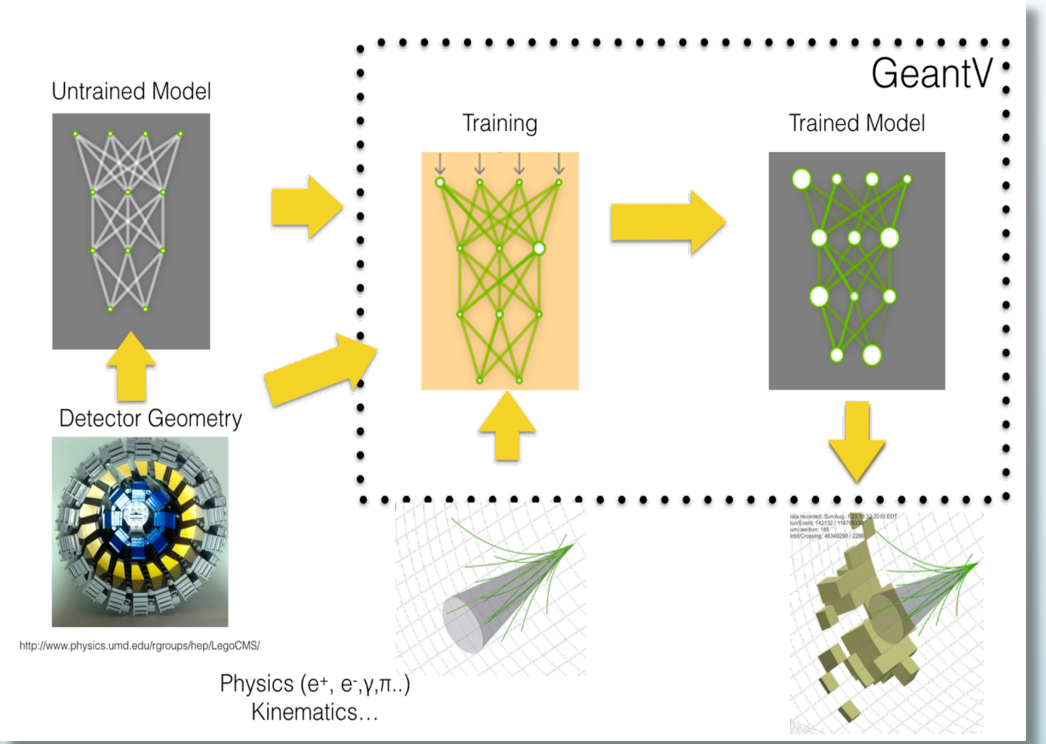
- Training on NVIDIA GTX-1080 for 30 epochs on 200k particles takes ~1 day
- Test different hardware
  - Testing on single node with an Intel® Xeon Phi™ processor (formerly code named Knights Landing)
    - Performance of underlying mathematical library (MKL) not as good (probably due to the size of our matrix operations)
  - Cloud environment
- Testing different frameworks
  - Intel Nervana Neon implementation is about 20% faster than Tensorflow on GPU

Multi-node scaling

- We want to provide a generic, fully configurable tool for fast simulation
- Optimal network design depends on the problem to solve
  - Hyper-parameters tuning and meta-optimization
- Parallelization on distributed systems
  - Evaluate existing libraries
  - Optimize training strategy and reduce communication overhead

DL engine for fast simulation

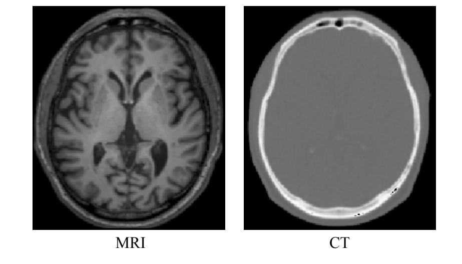
- GeantV GAN represent first proof of concept, developed within the GeantV prototype
  - We aim at a generic fully configurable tool
- Embed the tool in the GeantV prototype for testing
  - Inference step
  - Automated training
- Make it available as soon as possible in Geant4
  - or any future GeantX!



Before concluding…

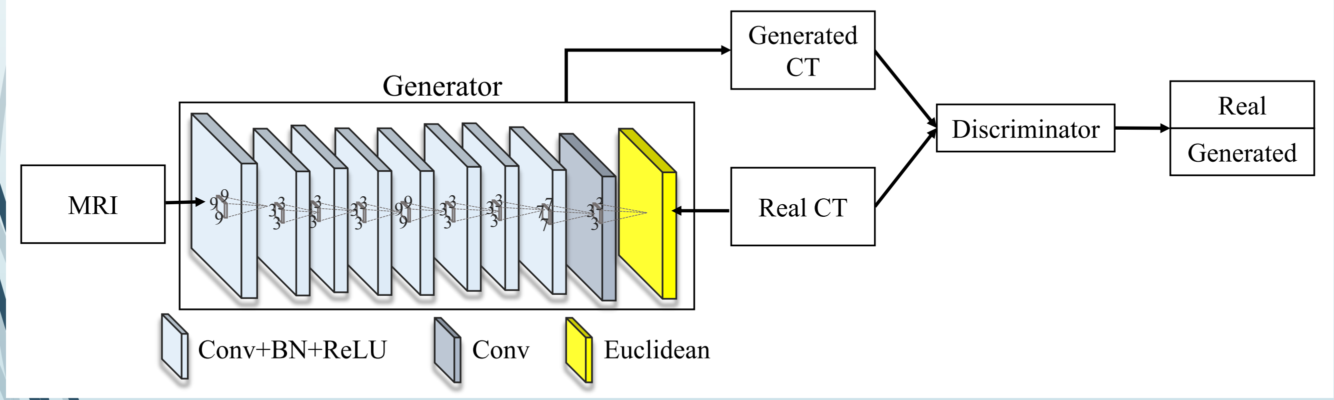
.. another 3d convolutional GAN application

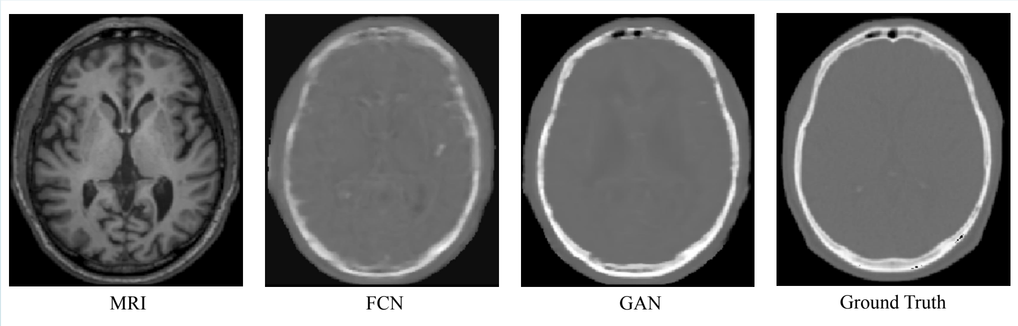
Medical Image Synthesis



- Patients are exposed to radiation during CT imaging
  - further increase potential risks of cancer
- MRI images contain much richer texture information than CT images
- It is challenging to directly estimate a mapping from MRI to CT.

- Use 3D convolutional GANs to “simulate” CT images from MR





Summary I

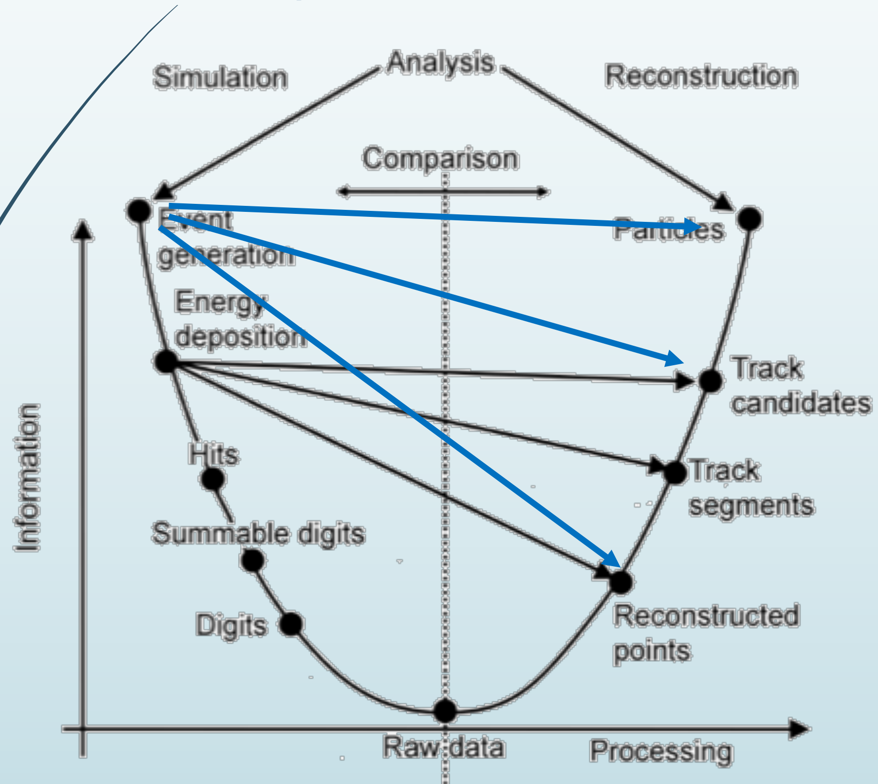
- MC production has been so far a major fraction of WLCG workload
  - Experiments are implementing a large range of fast simulation solutions
- HL-LHC runs will scale up MC needs by orders of magnitude
- A generic framework with common fast sim algorithm and strategies for mixing full and fast sim
 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

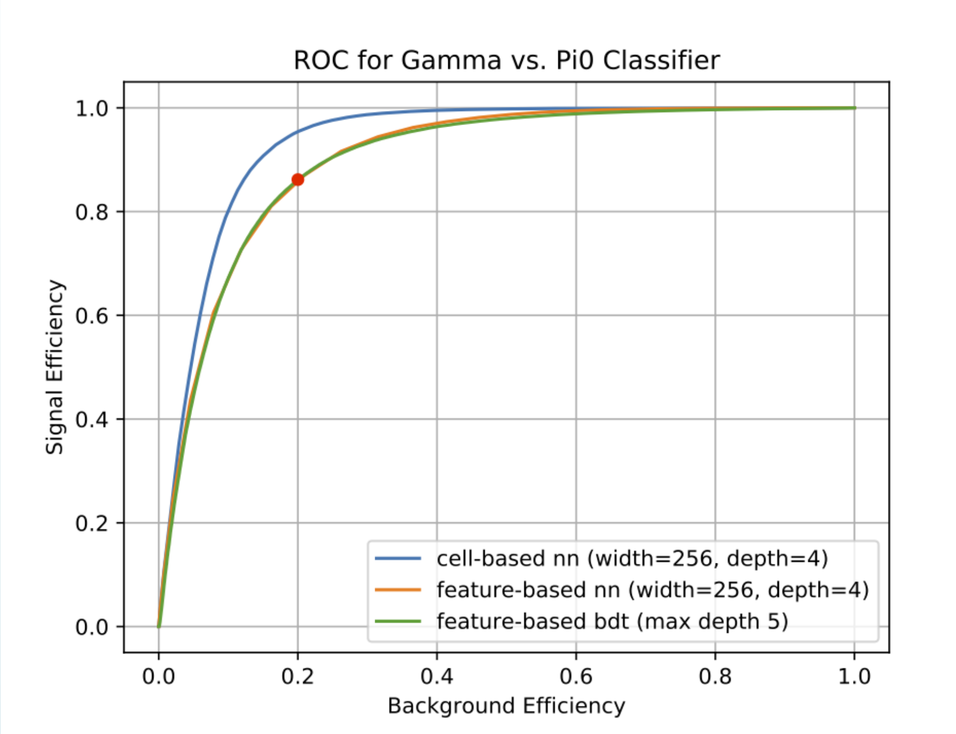
Summary II

- Generative Models seem good 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 towards a generic fully configurable tool
- Initially integrated in GeantV , and then integrate in Geant4 and other frameworks

Outlook

- Even larger speedup gained by replacing digitization and reconstruction steps
- As improved analysis techniques arise .. Could this not even be an issue any more??





B.Hooberman, S. V. et al. Submitted to NIPS2017

The end..

Deep Learning represent an impressive inter-disciplinary example!

HEP community can certainly profit from opening up and collaborating to different fields!

Thank you!

Questions?

Some references

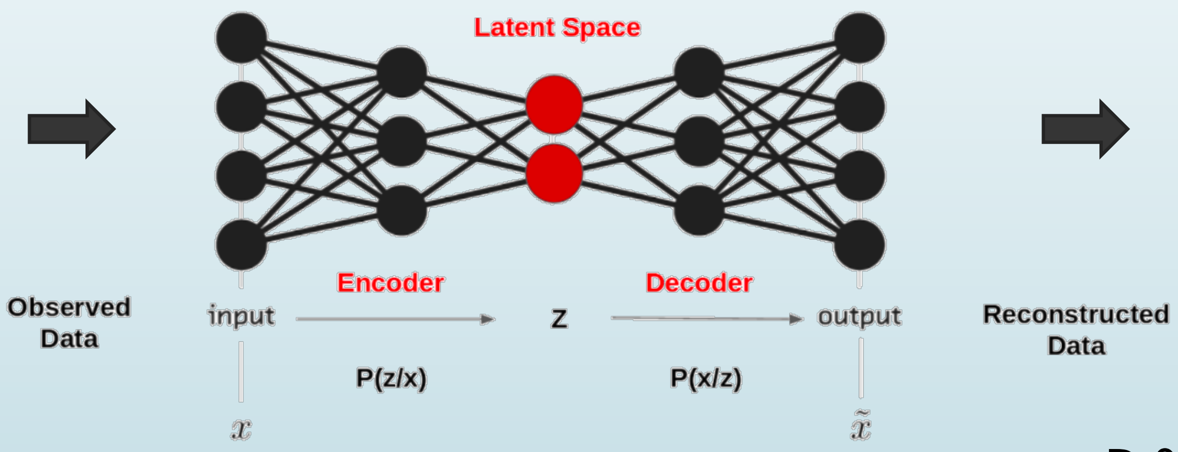
- GANs:
  - Just google “Generative Adversarial Networks”!
  - I. Goodfellow recent seminar: https://indico.cern.ch/event/673989/
  - A. Radford, L. Metz and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. 2015.
  - Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. 2014.
  - Augustus Odena, Christopher Olah, Jonathon Shlens, Conditional Image Synthesis with Auxiliary Classifier GANs. ICML, 2017.
  - Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. 2016.
  - Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. NIPS, 2016.
- Advanced GANs:
  - https://indico.cern.ch/event/655447/contributions/2742180/attachments/1552018/2438676/advanced_gans_iml.pdf (see refs on page 16)
- Physics and ML:
  - DS@HEP : (2017 workshop) https://indico.fnal.gov/event/13497/timetable/#20170508
  - Connecting the dots:
  - https://indico.hephy.oeaw.ac.at/event/86/timetable/#20160222 (2016 workshop)
  - IML workshops: https://indico.cern.ch/event/595059/ and https://indico.cern.ch/event/655447/

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)

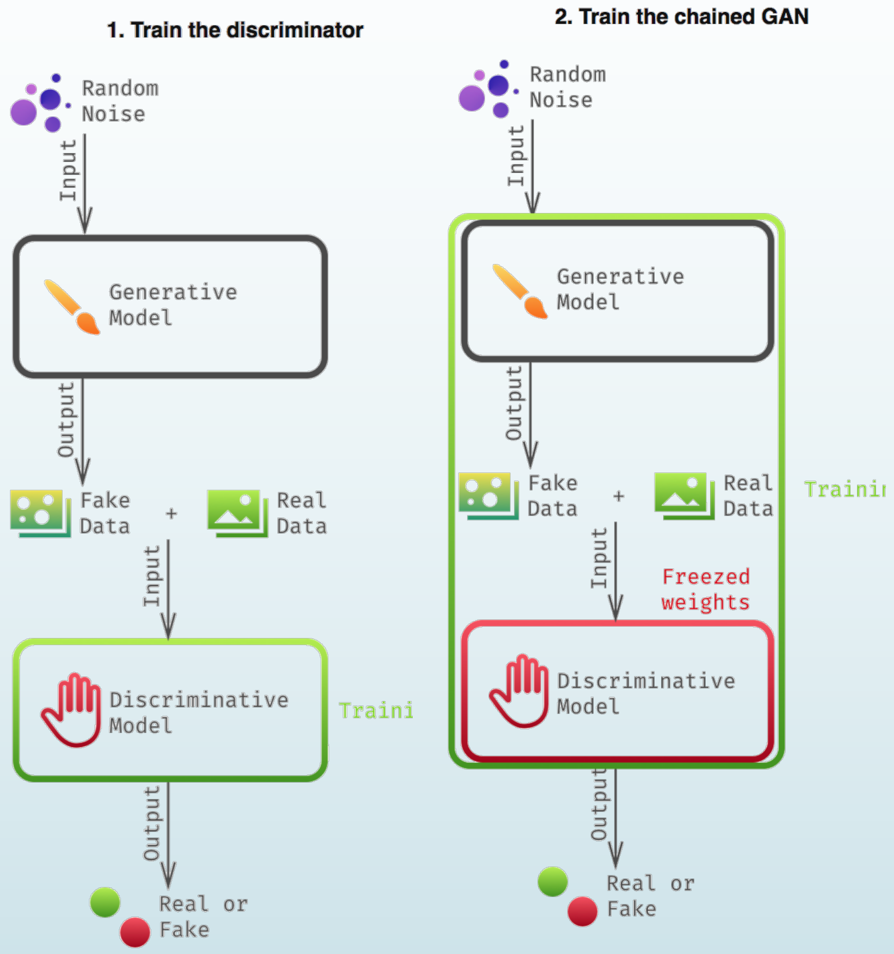


Generated Calorimeter Shower Hit (χ,y,z,E,σ)

D.Salamani, U. of Geneva

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

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 $p + p \to H \to f\bar{f}$ events