Fast simulation with Machine Learning

Marilena Bandieramonte
marilena.bandieramonte@cern.ch

23rd Geant4 Collaboration Meeting
Lund, Sweden

Outline

Introduction

Why experiments need more and more fast simulations approaches & status of the Art

AI, ML and Deep Learning

The Machine Learning Hype

Fast Simulation with ML in the experiments

Different R&D on fast simulation with ML in LHC experiments and work in progress results

A Generic Fast simulation approach

To what extent these approaches can be generalized? What can we provide as a community?

Thanks to..

ATLAS — John Chapman, Graeme A Stewart, Jana Schaarschmidt

CMS — Sezen Sekmen, V. Daniel Elvira

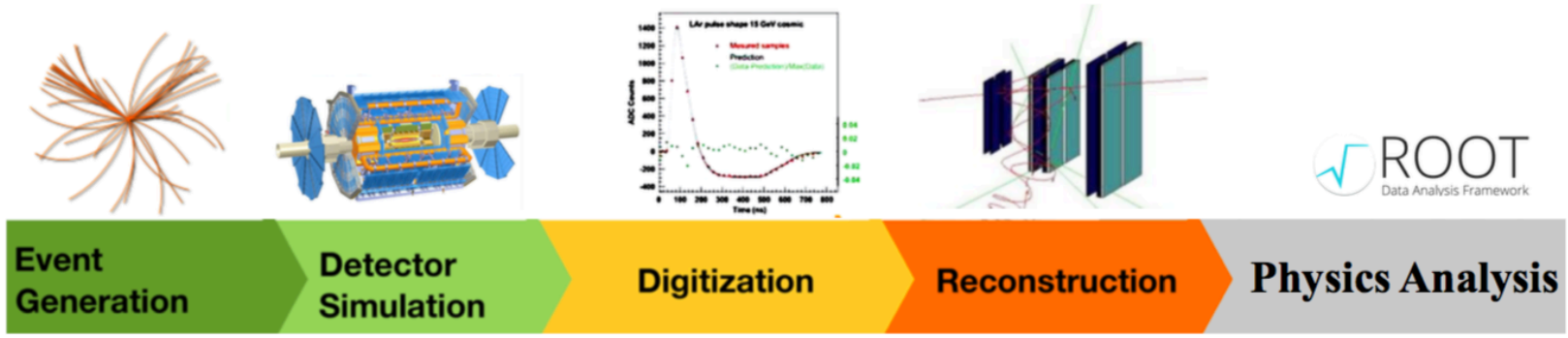
LHCb — Gloria Corti, Mark Whitehead

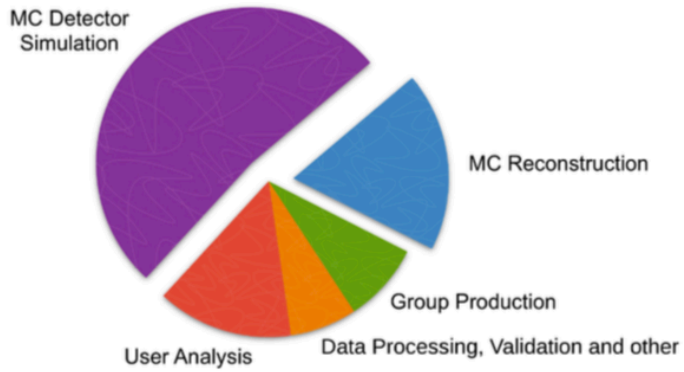
ALICE — Sandro C. Wenzel, Tomasz Trzcinski

…and a disclaimer

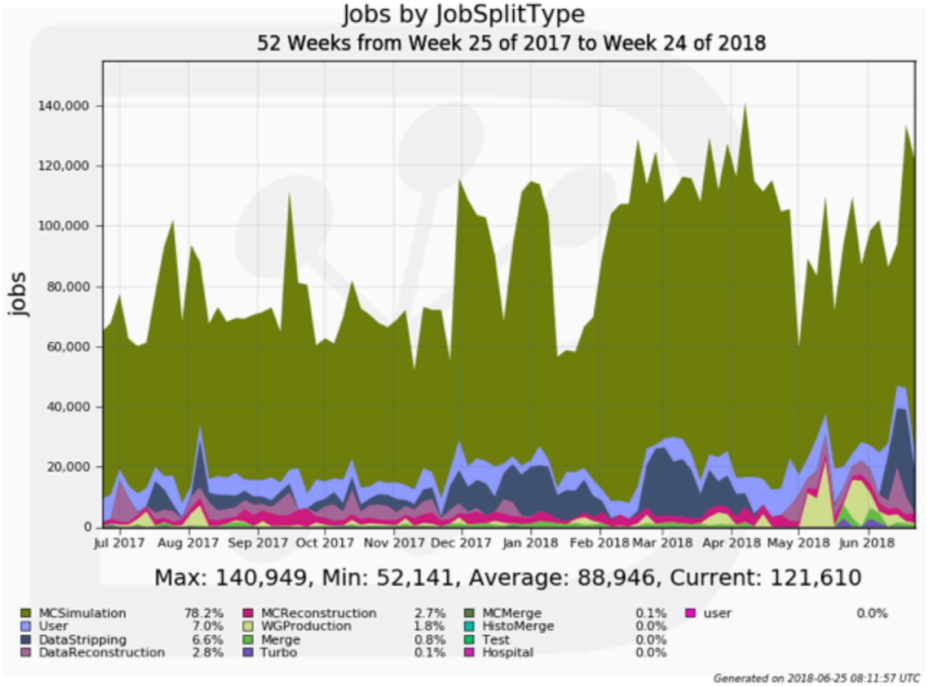
I am not an expert in Machine Learning … but I have watched a number of YouTube videos :)

Need for Fast Sim

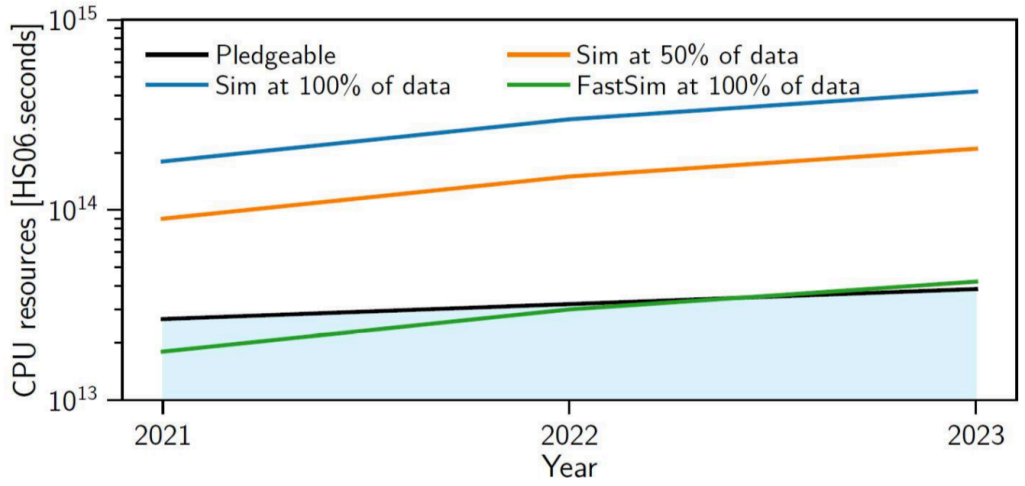




ICHEP2018, Hasib Ahmed (Atlas)



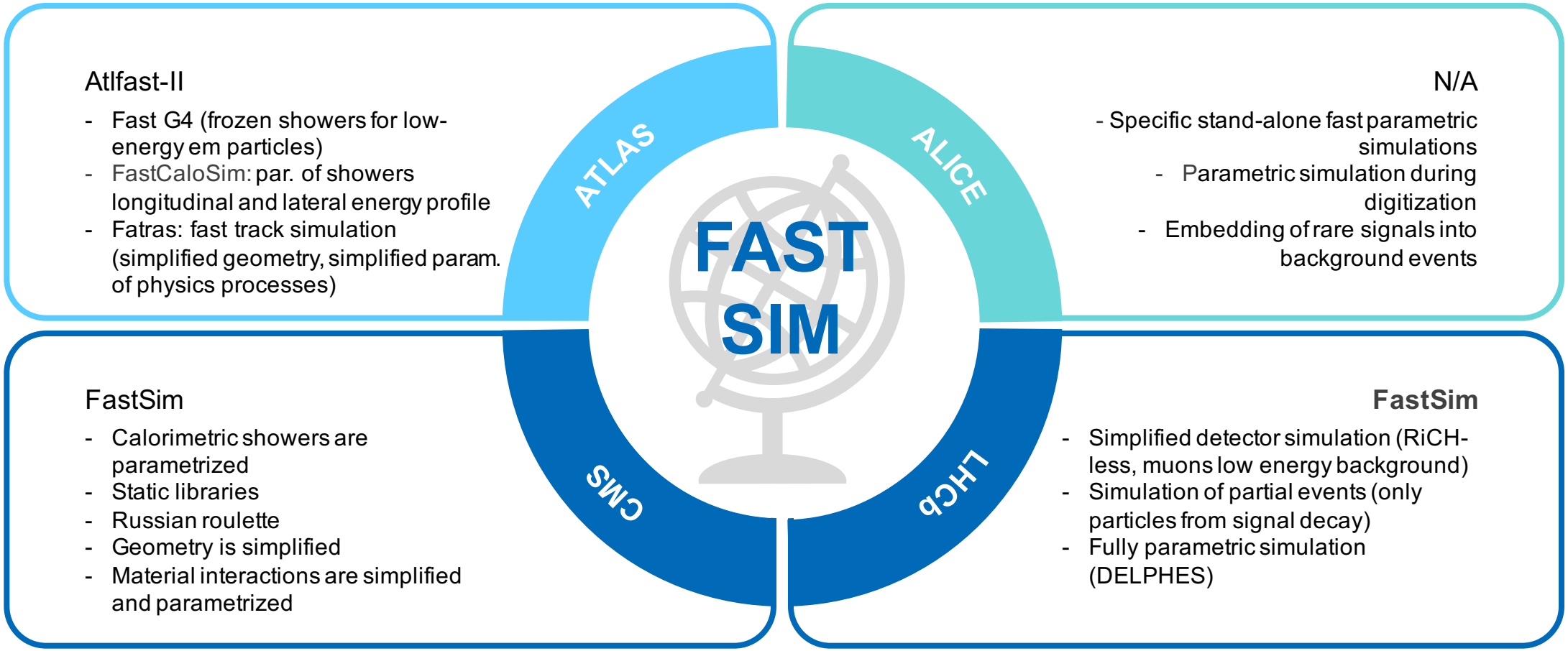
Legend: “Sim at 50% of data” = FullSim sample is 50% the datasize, FastSim sample is 50% the datasize



FastSim assumed to be 1/10 faster than FullSim

HSF workshop 2018, G. Corti (LHCb)

”Traditional” FastSim



Atlfast-II

- Fast G4 (frozen showers for low-energy em particles)
- FastCaloSim: par. of showers longitudinal and lateral energy profile
- Fatras: fast track simulation (simplified geometry, simplified param. of physics processes)

N/A

- Specific stand-alone fast parametric simulations
- Parametric simulation during digitization
- Embedding of rare signals into background events

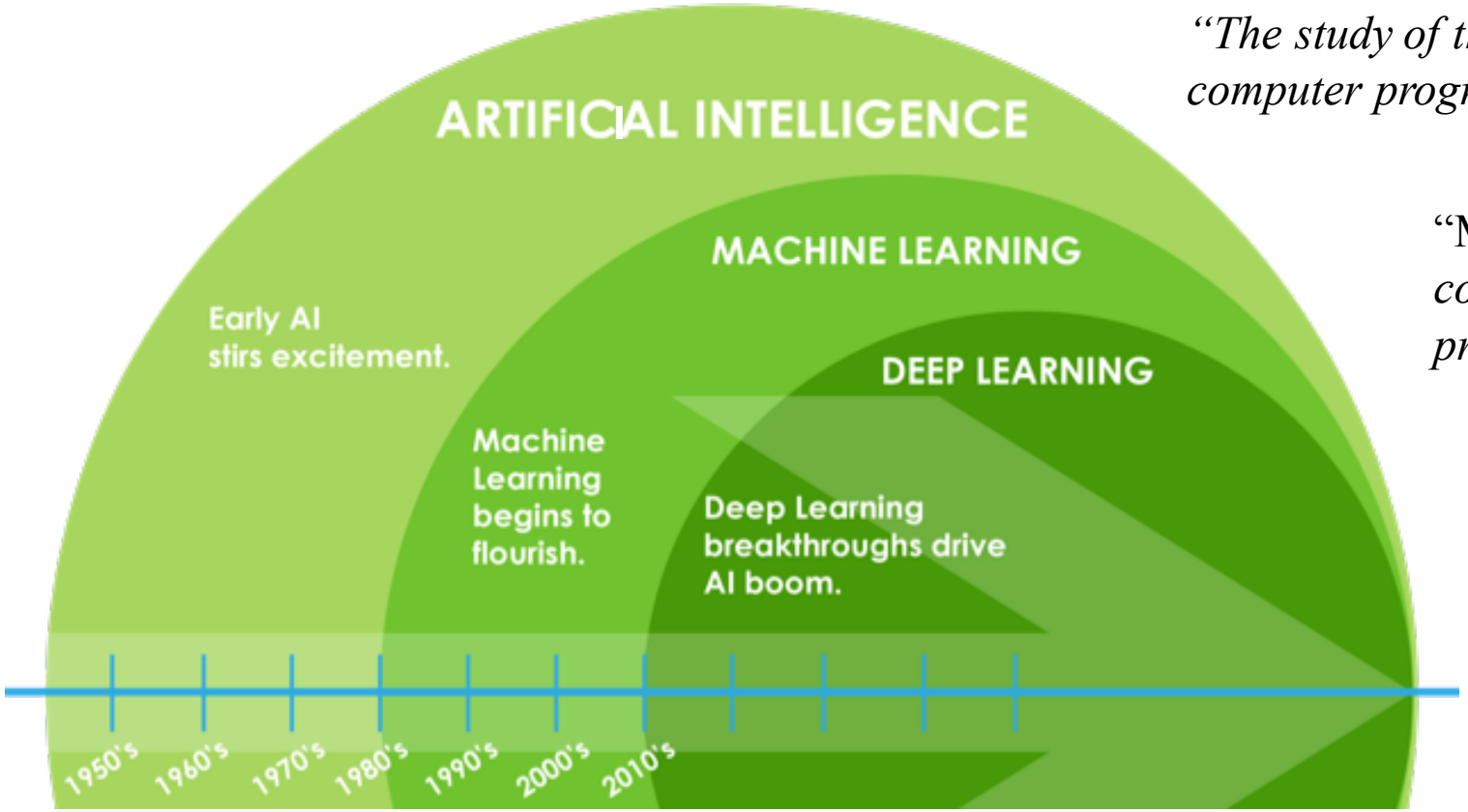
FastSim

- Calorimetric showers are parametrized
- Static libraries
- Russian roulette
- Geometry is simplified
- Material interactions are simplified and parametrized

FastSim

- Simplified detector simulation (RiCH-less, muons low energy background)
- Simulation of partial events (only particles from signal decay)
- Fully parametric simulation (DELPHES)

AI, ML and DL

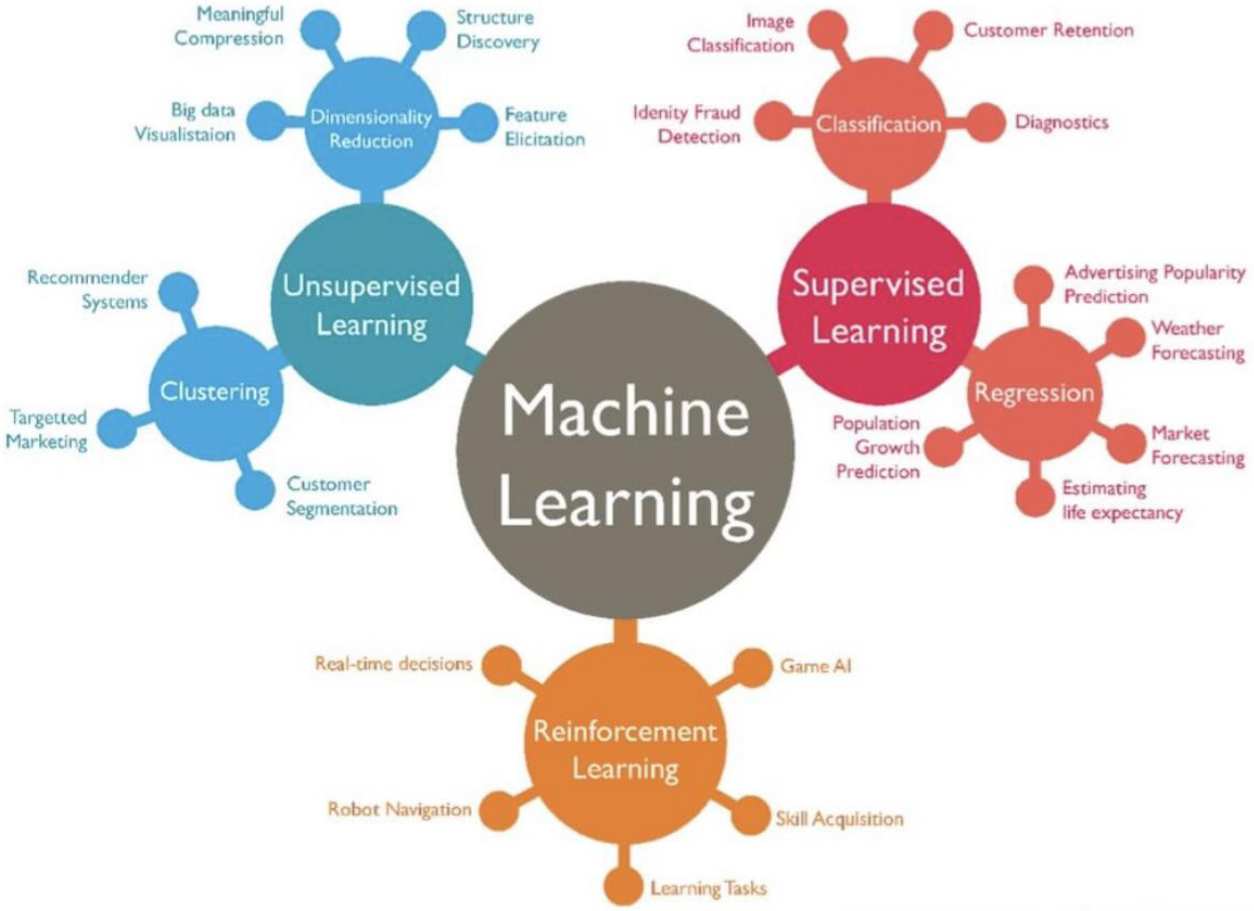


"The study of the modelling of human mental functions by computer programs."— Collins Dictionary

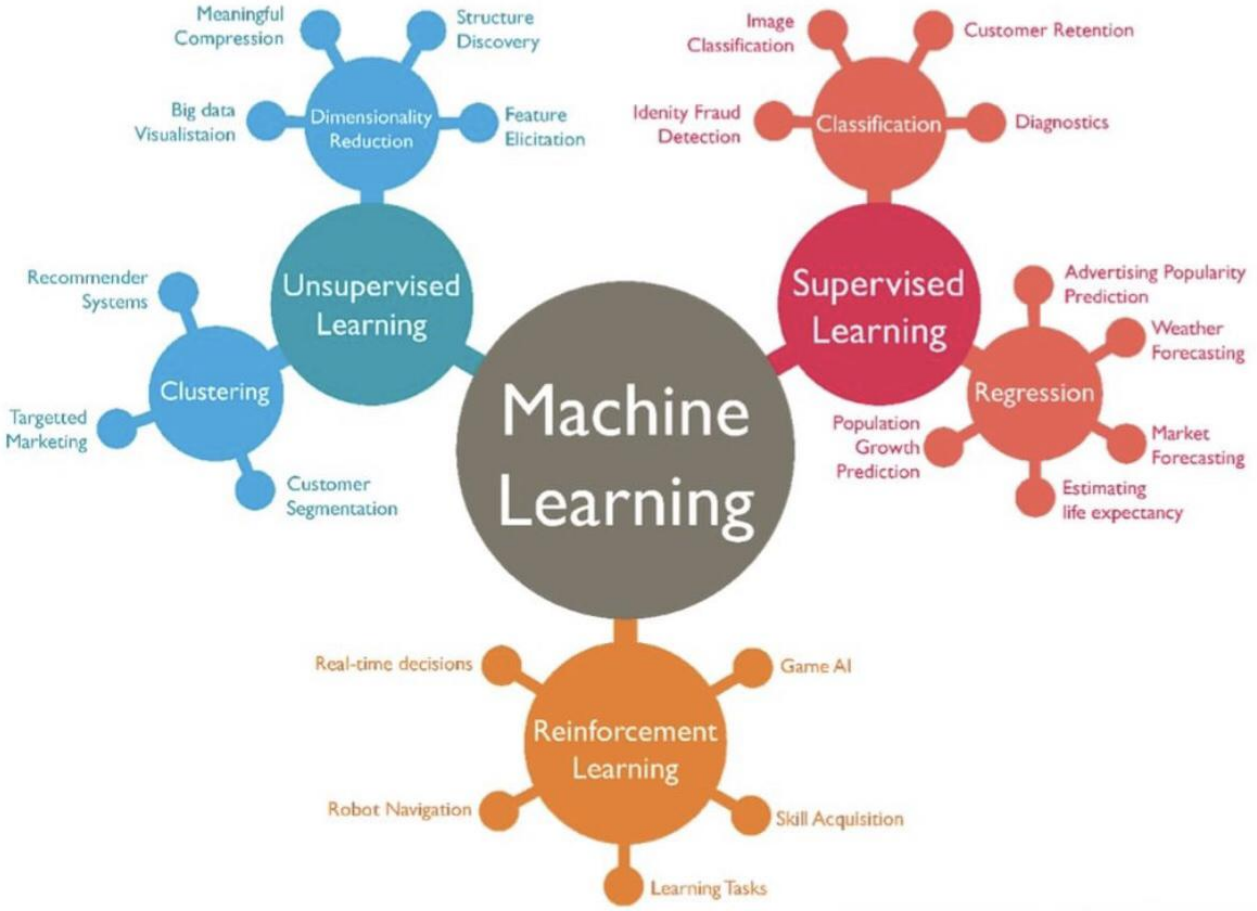
"Machine learning is the sc*ience of getting computers to act without being explicitly programmed."—Stanford University*

"Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks".— Machine Learning Mastery

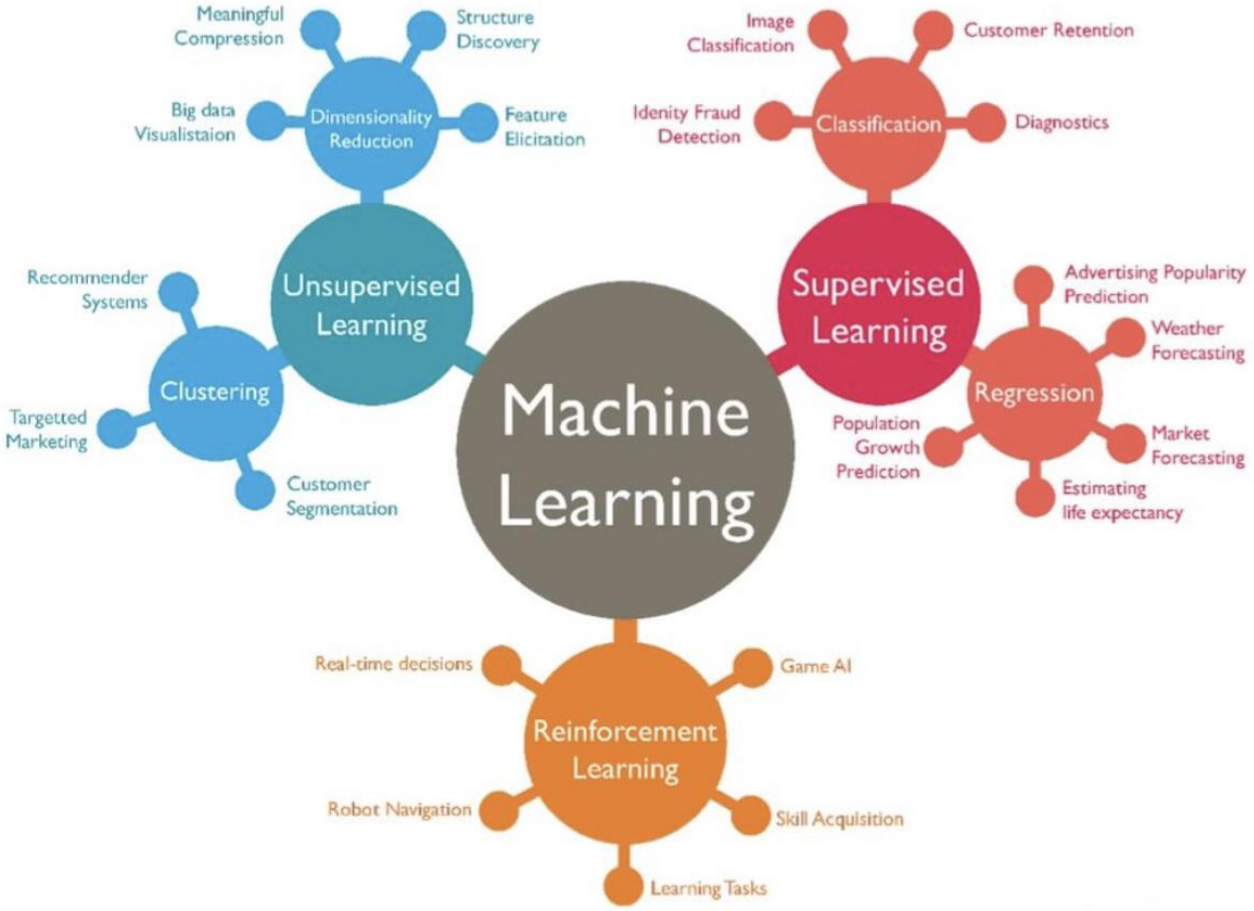
Variety of ML/DL algorithms



Variety of ML/DL algorithms



Variety of ML/DL algorithms



Variety of ML/DL algorithms

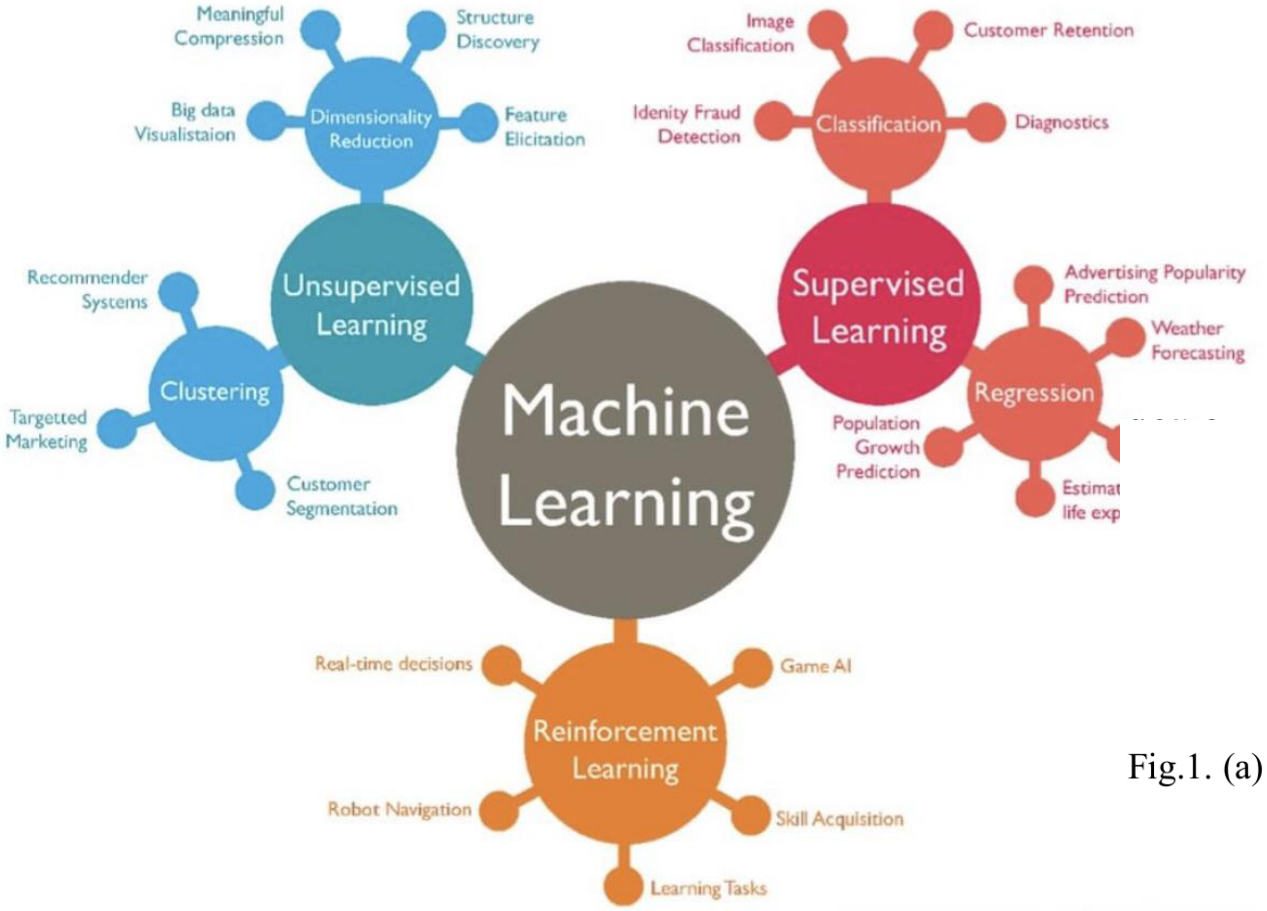
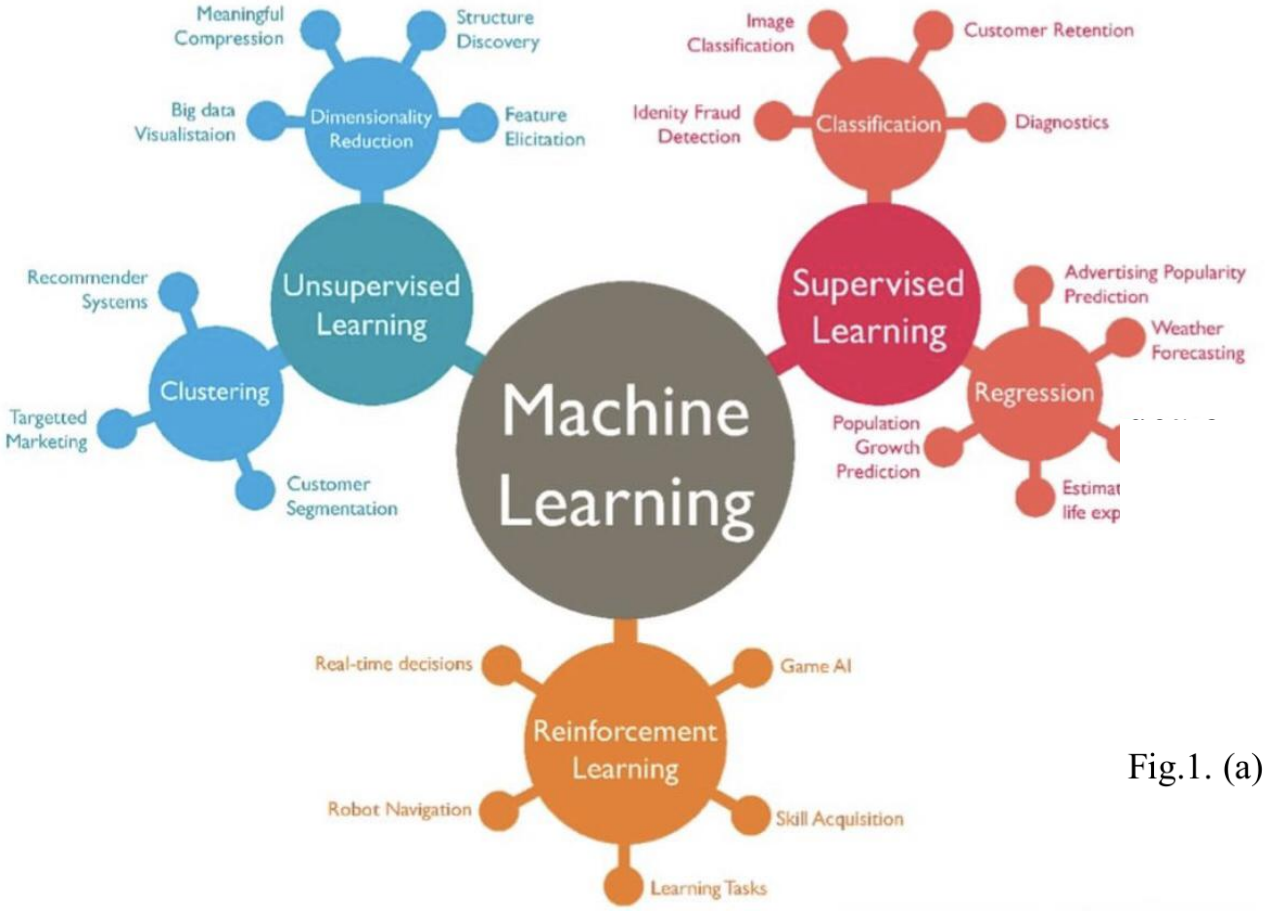


Fig.1. (a) Low Resolution Input Face; (b) Resultant Hallucinated Face; (c) Original Face.

Variety of ML/DL algorithms



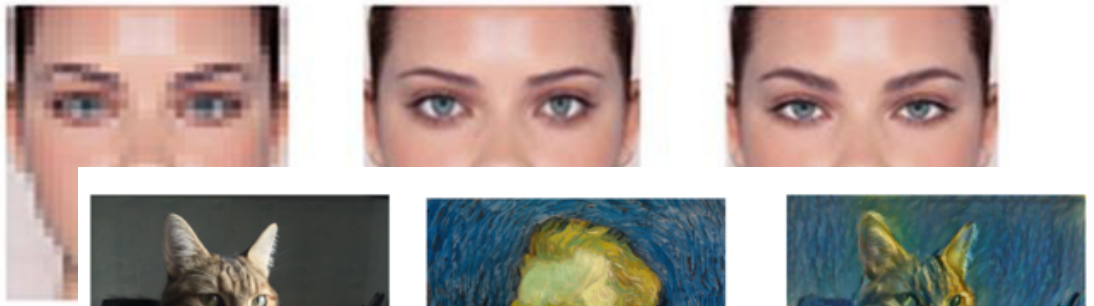
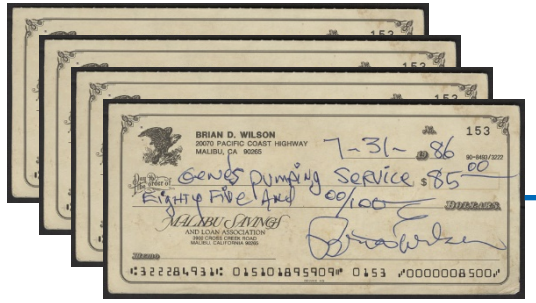


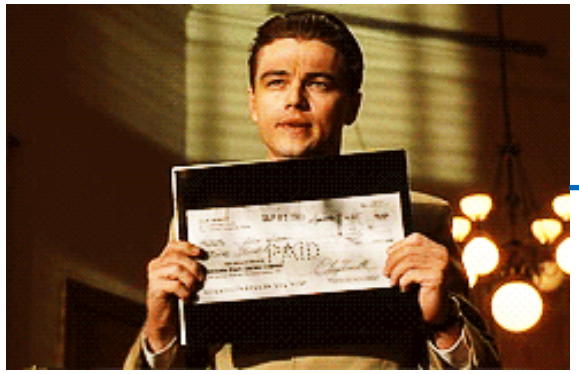
Fig.1. (a) Low I ace.

Generative Adversarial Netwoks

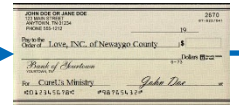


https://33milesinnewaygocounty.files.wordpress.com

Generator



https://giphy.com/gifs/leonardo-dicaprio-catch-me-if-you-can-5leocharacters-t1h4nnWEWKfn2



Discriminator

https://thechive.files.wordpress.com

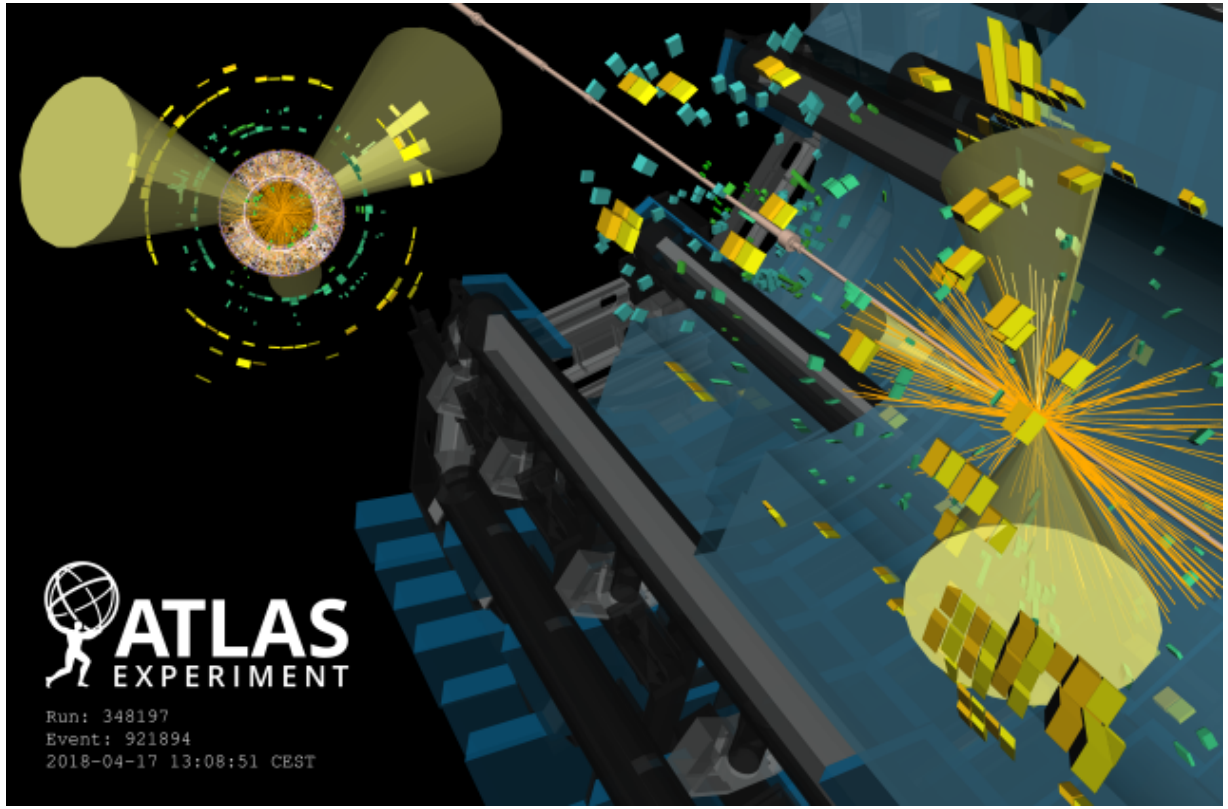
*First example of simplified calorimeter simulation, CaloGan: M. Paganini et al. arXiv:1712.10321

FastSim Atlas

During Run 1 and 2 of the LHC, a fast calorimeter simulation (FastCaloSim) was successfully used in ATLAS.

An improved version of FastCaloSimv2 that incorporates the experience gained with the Run 1 version is currently under development.

The new FastCaloSim makes use of machine learning techniques, such as **principal component analysis** and **neural networks**, to optimise the amount of information stored in the ATLAS simulation infrastructure.



PCA

The longitudinal energy parametrization is based on a principal component analysis (PCA), to decorrelate deposited energies in the various calorimeter layers

GAN

Generative Adversarial Network. The main concept behind this unsupervised generative model is to train two neural networks to play a min-max game between each other.

Longitudinal and lateral energy parameterization

CHEP 2018, Jana Schaarschmidt *(ATLAS)*

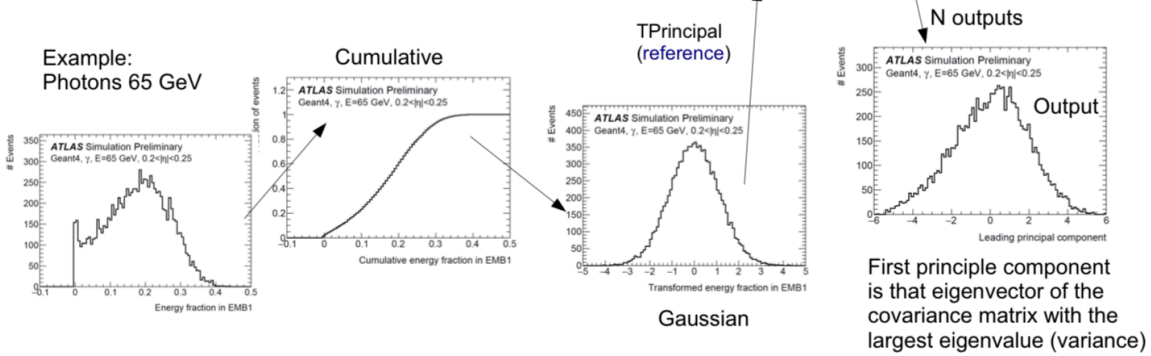
The Longitudinal Energy Parametrisation

Energy deposit in each calorimeter layer along the shower axis and total energy

Problem: The energy deposits in the various layers are correlated with each other

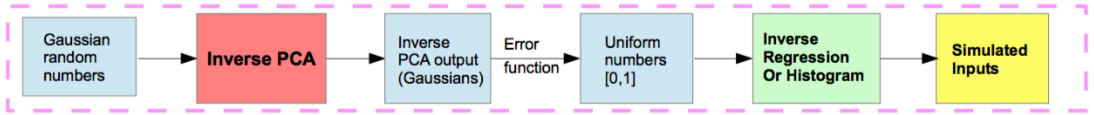
Transformation to uncorrelated set of variables with principal component analysis, to reduce complexity

1st PCA chain:

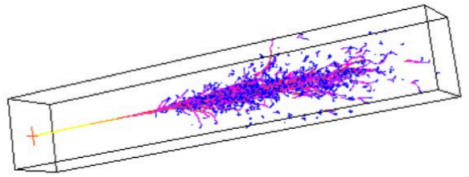


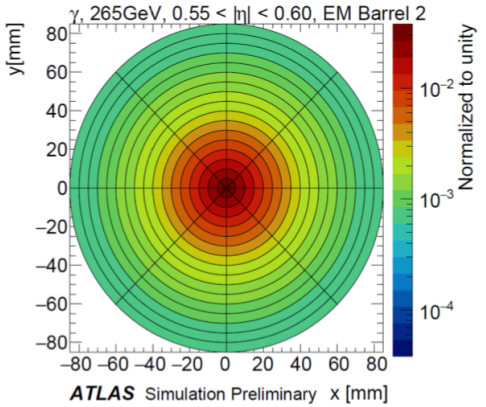
First principle component is that eigenvector of the covariance matrix with the largest eigenvalue (variance)

During simulation, this chain is performed back-wards:



The Lateral Energy Parametrisation („Shape“)



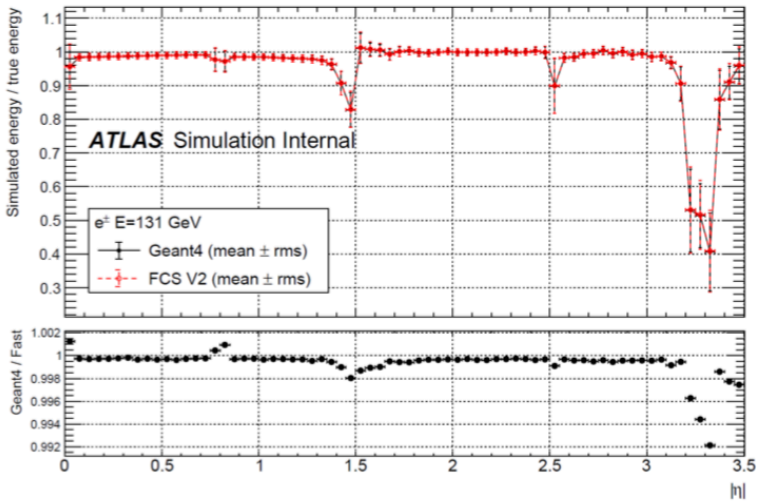


- Shower shape:
  - Most energies in the center (close to the shower axis)
  - Energy tails extending perpendicular to the axis
- The shape parametrisation is based on Geant4 HITs.
  - Close-by hits merged to reduce computation time
  - Hits saved in ntuple format to be used to derive histograms
- These 2D histograms act as **probability density functions** during the fast simulation: Fast sim hits are randomly sampled from it
- 2D histogram stored per layer and per PCA bin
- Spline and regression techniques can be used to reduce memory

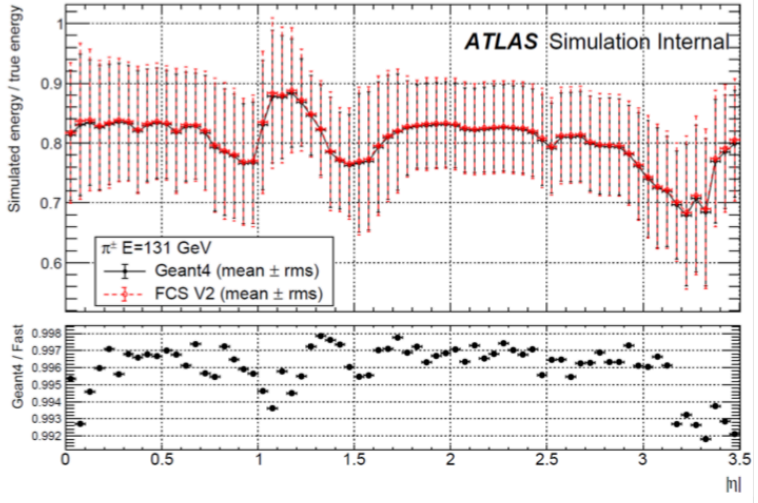
Validation of the energy response

Validation of the energy response

Electrons:



Pions:



- Egamma showers are more narrow, so more sensitive to the detector geometry changes
- Total energy response agrees remarkably well between G4 and new FastCaloSim
- Even if correlations between layers are not well modelled for difficult eta regions, the total energy is still well reproduced

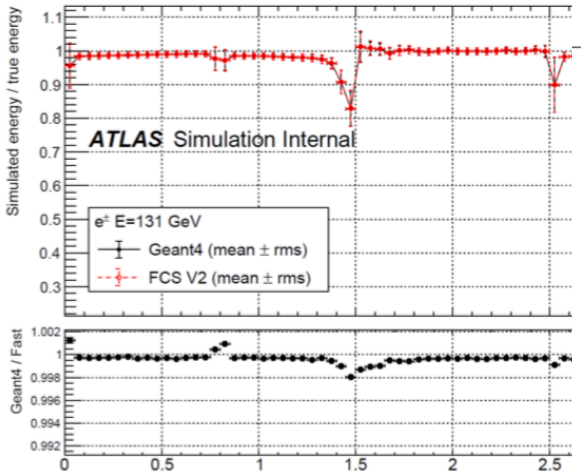
CHEP 2018, Jana Schaarschmidt *(ATLAS)*

Validation of the energy response

CHEP 2018, Jana Schaarschmidt *(ATLAS)*

Validation of the energy response

Electrons:

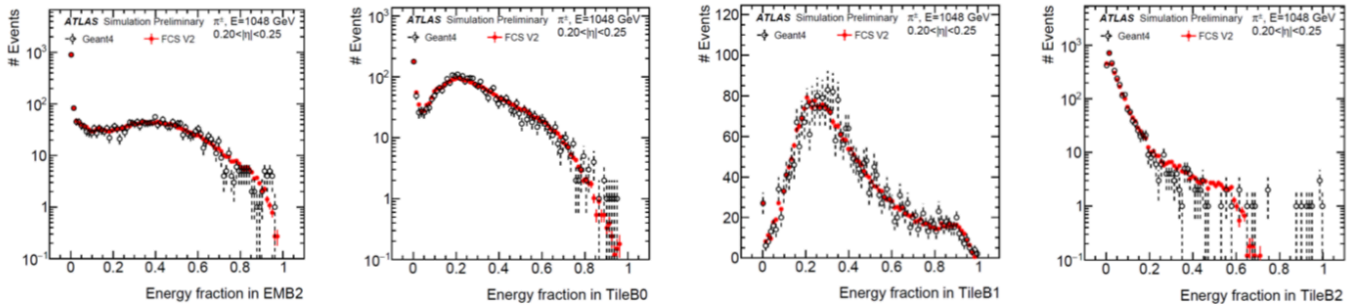


- Egamma showers are more narrow,
- Total energy response agrees remai
- Even if correlations between layers still well reproduced

Pions:

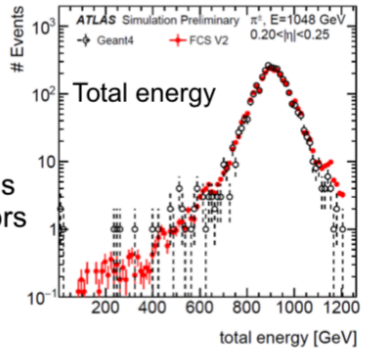
Validation of the energy response

Energy deposited in each layer, 1 TeV central pions:

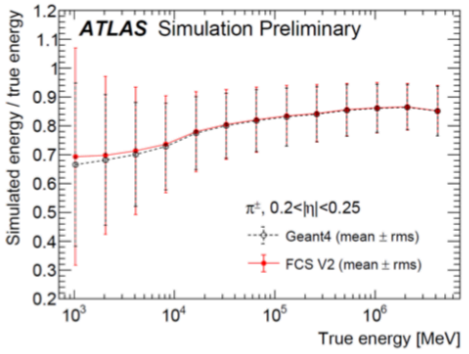


Energy in all layers well modelled.

For transition regions between subdetectors still some problems (not shown here).



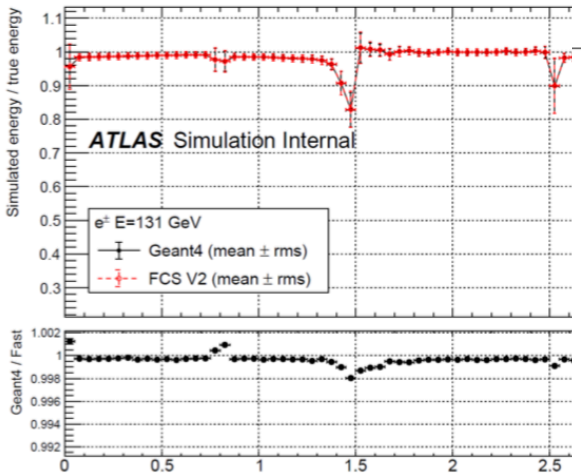
Energy response, central pions, scan over energy →



Validation of the energy response

Validation of the energy response

Electrons:

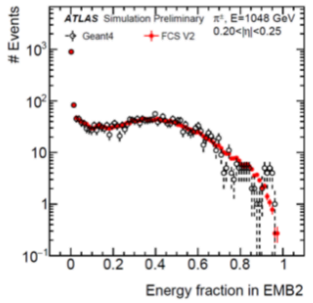


- Egamma showers are more narrow,
- Total energy response agrees remai
- Even if correlations between layers still well reproduced

Pions:

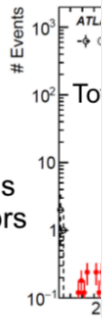
Validation of the energy response

Energy deposited in each



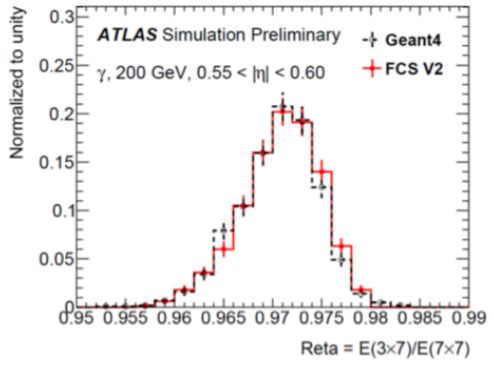
Energy in all layers well modelled.

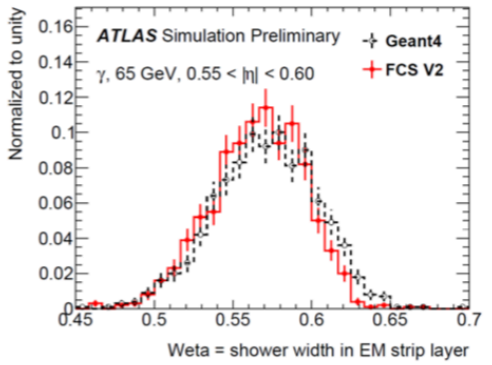
For transition regions between subdetectors still some problems (not shown here).

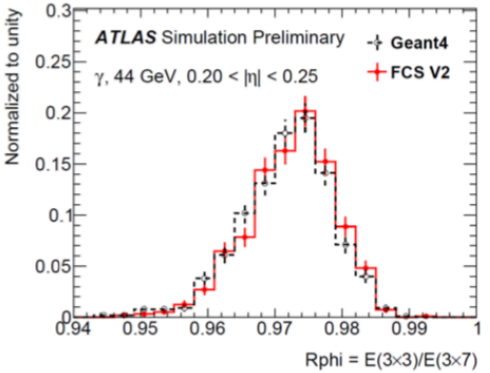


Validation of single particles (after reconstruction)

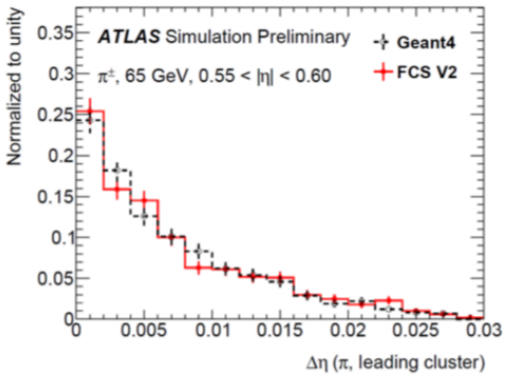
Photons, 0.2<|η|<0.25:

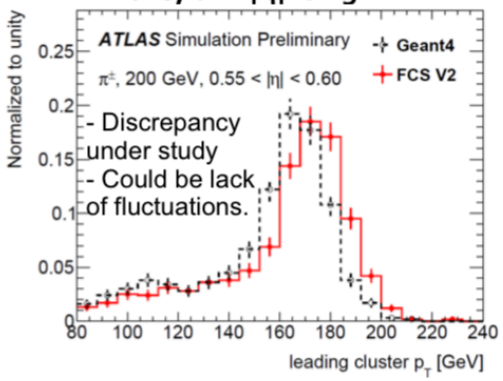


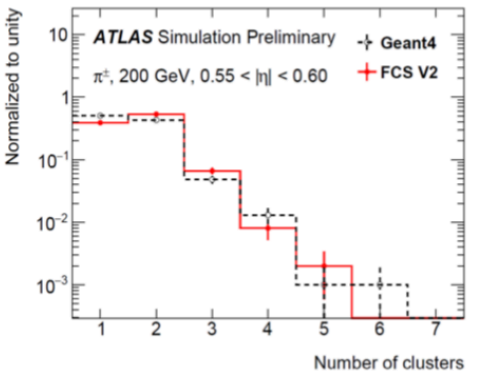




Pions, 0.2<|η|<0.25:







CHEP 2018, Jana Schaarschmidt *(ATLAS)*

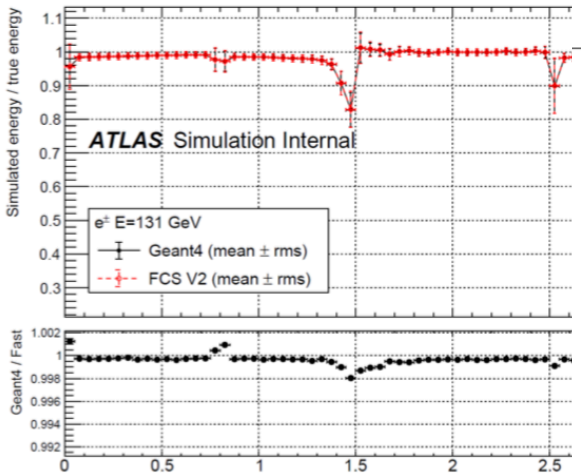
Validation of the energy response

CHEP 2018, Jana Schaarschmidt *(ATLAS)*

Validation of the energy response

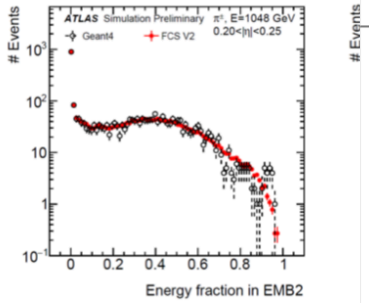
12 / 16

Electrons:



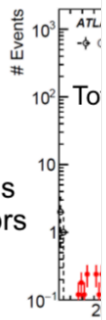
- Egamma showers are more narrow,
- Total energy response agrees remai
- Even if correlations between layers still well reproduced

Pions:



Energy in all layers well modelled.

For transition regions between subdetectors still some problems (not shown here).



Validation of the energy response

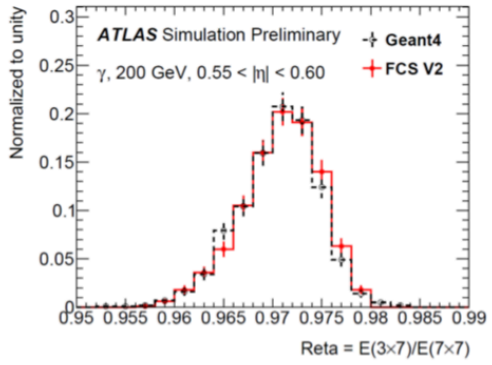
13 / 16

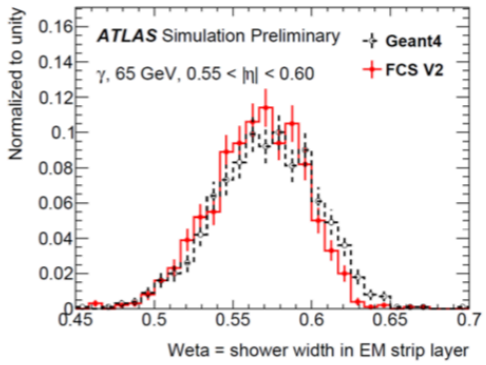
Energy deposited in each

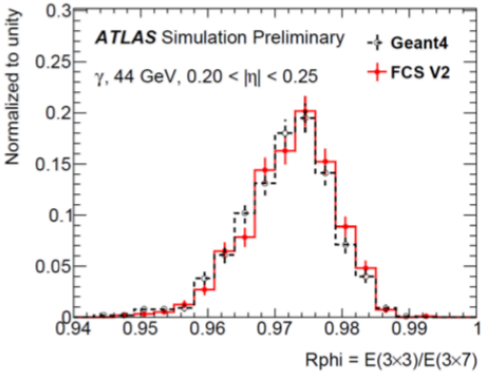
Validation of single particles (after reconstruction)

14 / 16

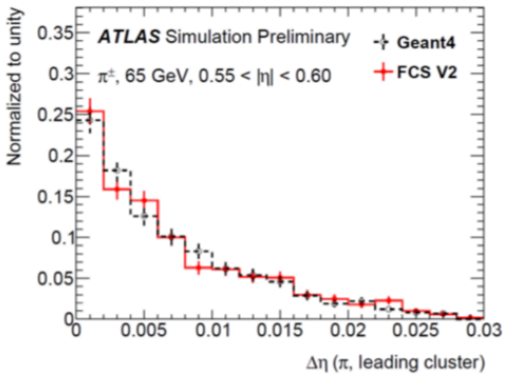
Photons, 0.2<|η|<0.25:

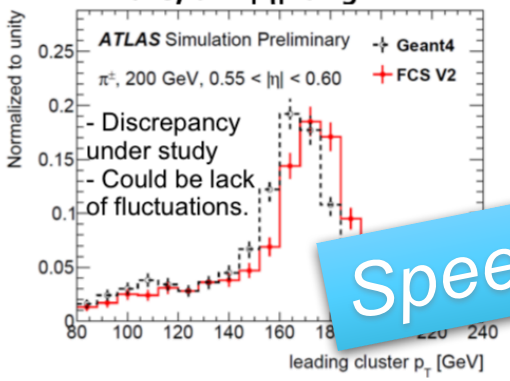


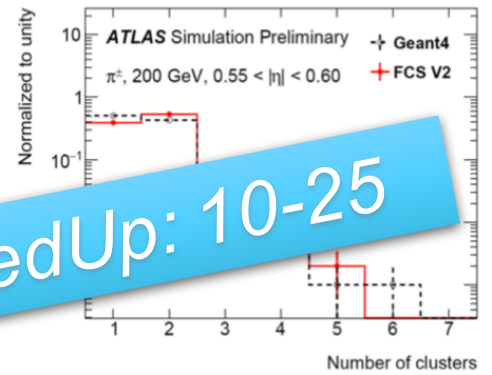




Pions, 0.2<|η|<0.25:







SpeedUp: 10-25

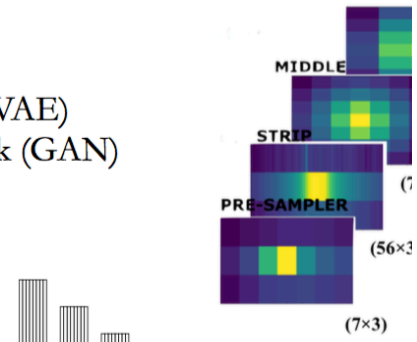
DNNCaloSim*

ICHEP 2018, Hasib Ahmed *(ATLAS)*

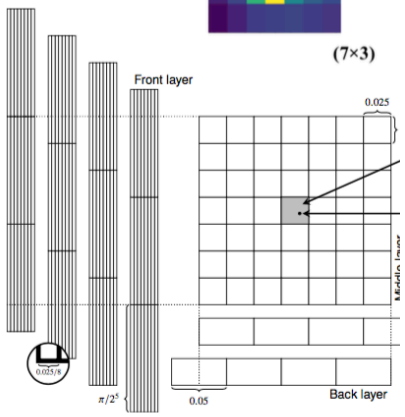
New approaches of fast simulation: *DNNCaloSim*

Deep generative networks to generate EM showers

Networks investigated:
Variational Auto Encoder (VAE)
Generative Adversarial Network (GAN)



- Only photons in EM calorimeter (< 1% leakage to hadronic calorimeter)
- Energies [1, 260] logarithmically spaced
- Pseudo rapidity $0.20 < |\eta| < 0.25$
- The energy deposits are voxalized into rectangular shapes
- A total of 266 cells are considered for energy deposits
- The networks are trained with energies normalized to the energy of the incident particle

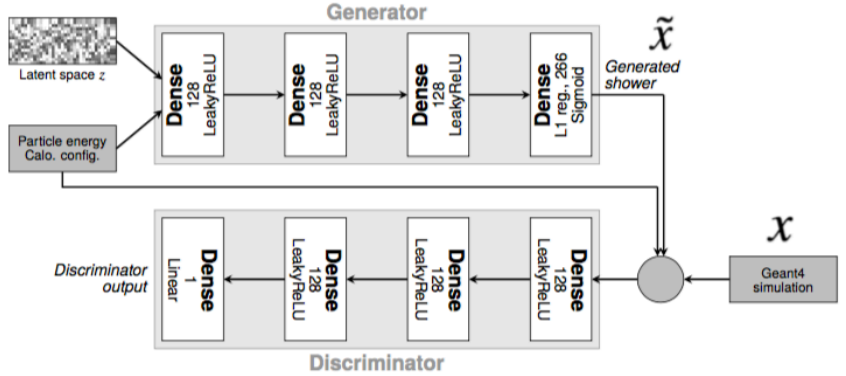


Hasib Ahmed(U Edinburgh)

Generative Adversarial Network

DNNCaloSim

Generative network with a feedback from a Discriminator network



Improve the robustness of training by calculating Wasserstein loss with a two sided gradient penalty

$$L_{\text{GAN}} = \underbrace{E_{\tilde{x}\sim p_{\text{gen}}}[D(\tilde{x})]}_{\text{ability to identify generated shower correctly}} - \underbrace{E_{x\sim p_{\text{Geant4}}}[D(x)]}_{\text{ability to identify Geant4 shower correctly}} + \lambda \underbrace{E_{\hat{x}\sim p_{\hat{x}}}[(||\Delta_{\hat{x}} D(\hat{x})||_2 - 1)^2]}_{\text{penalizes by calculating Wasserstein loss}}.$$

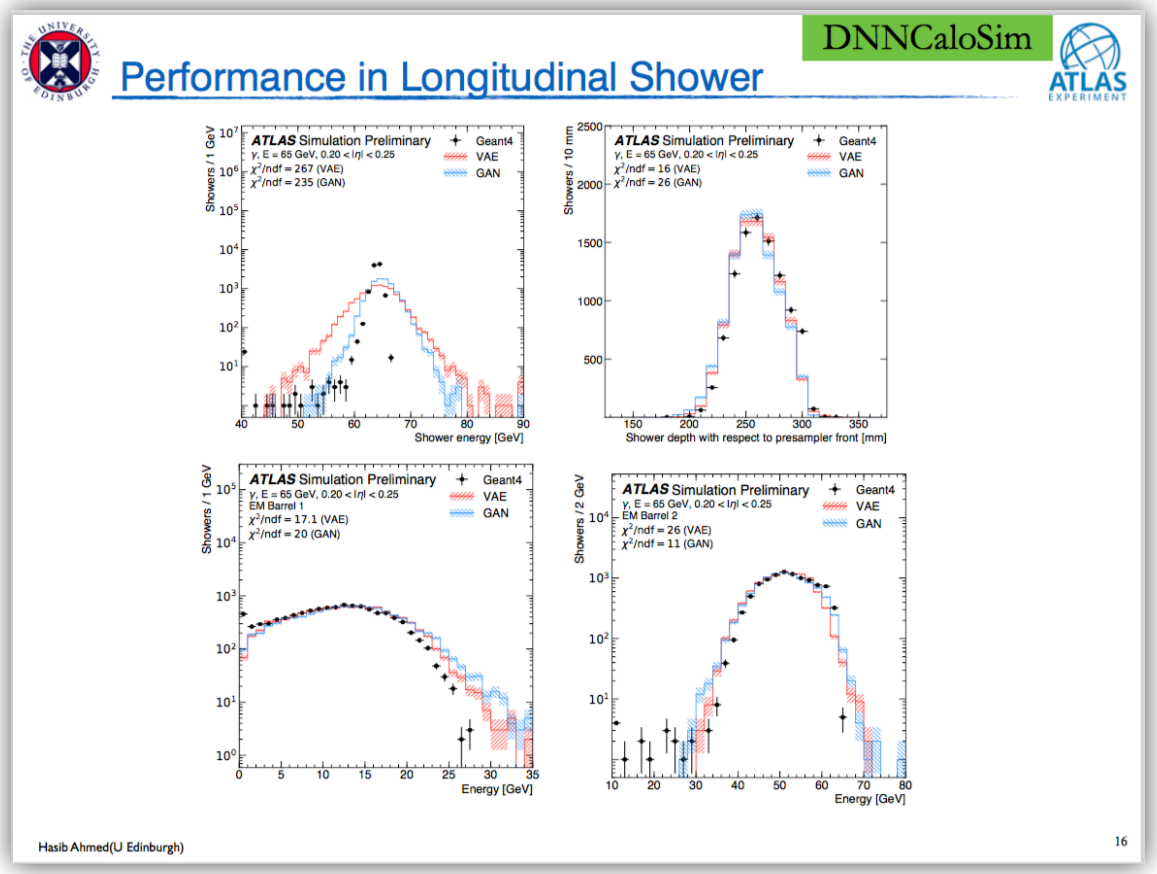
Hasib Ahmed(U Edinburgh)

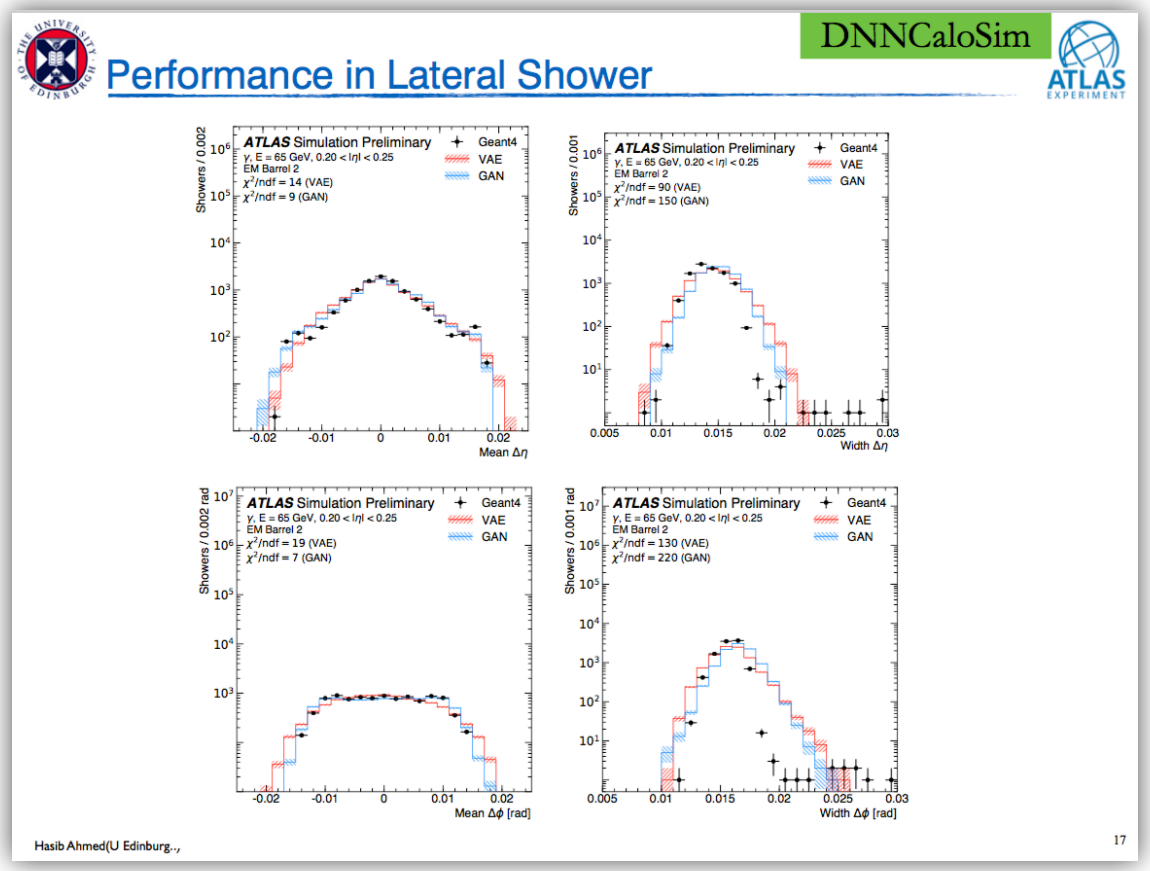
15

*Based on CaloGan: M. Paganini et al. arXiv:1712.10321

DNNCaloSim

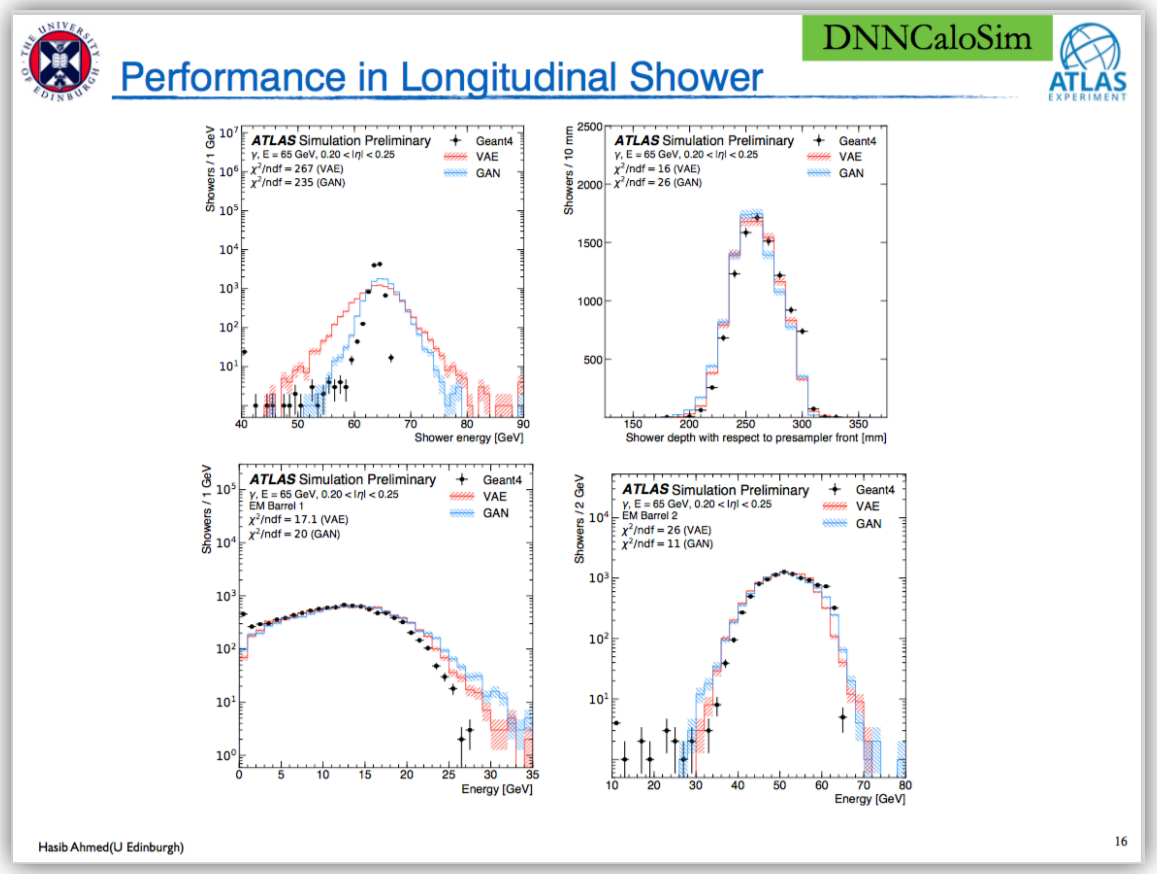
ICHEP 2018, Hasib Ahmed *(ATLAS)*

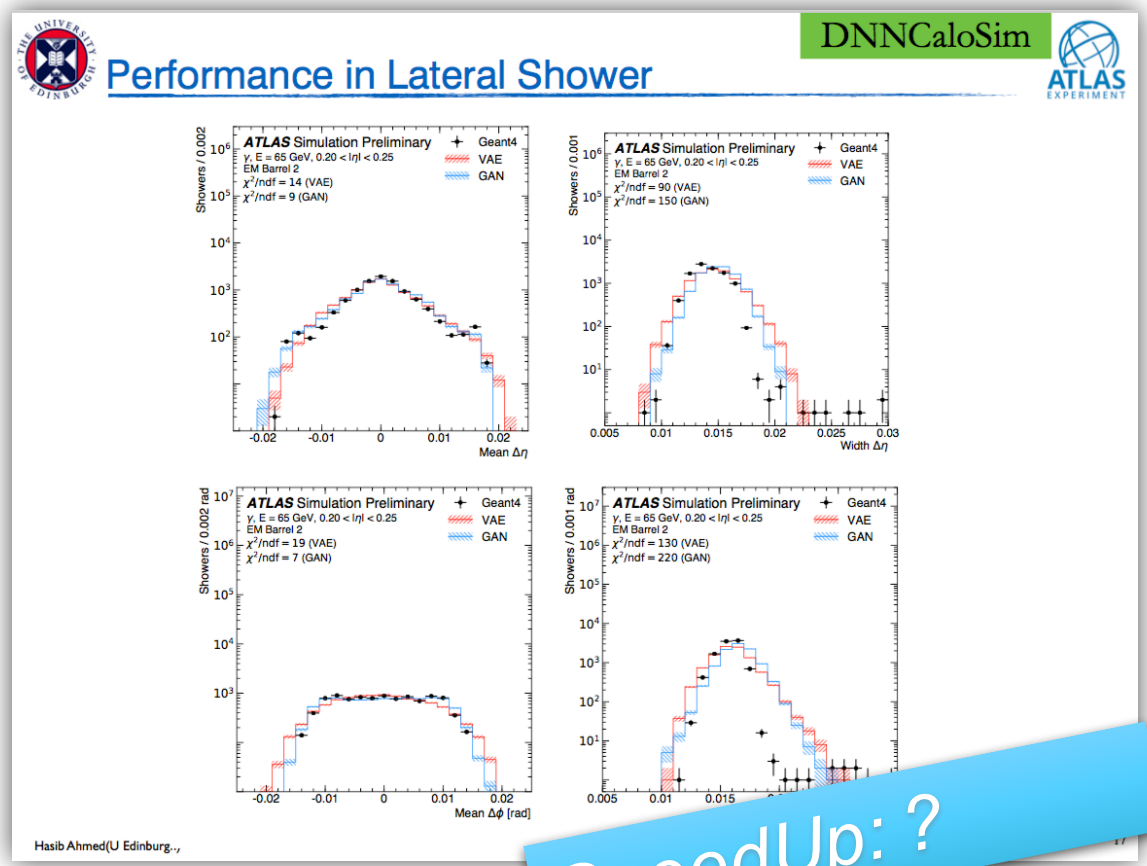




DNNCaloSim

ICHEP 2018, Hasib Ahmed *(ATLAS)*





FastSim Alice

Using generative models for fast simulations in the TPC (Time Projection Chamber) detector for the ALICE Experiment

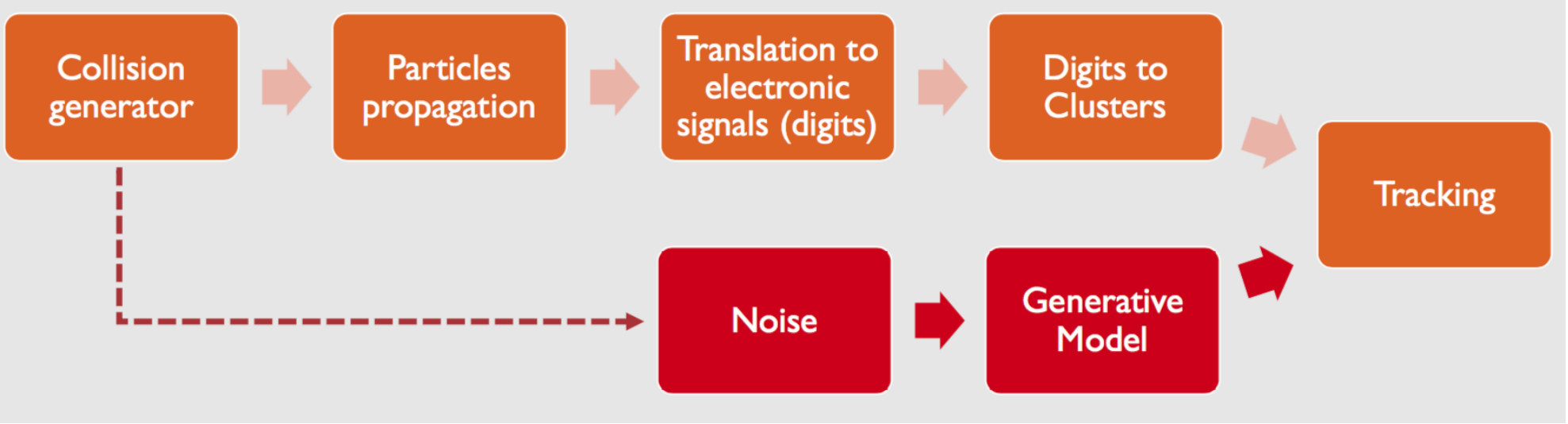
Substitute part of the simulation pipeline, namely particle propagation and translations to digits and clusters, with a generative model, initialized with noise.

Cluster Simulation

The dataset consists of 3D trajectories of particles after collision generated using Monte Carlo simulation

DCGAN

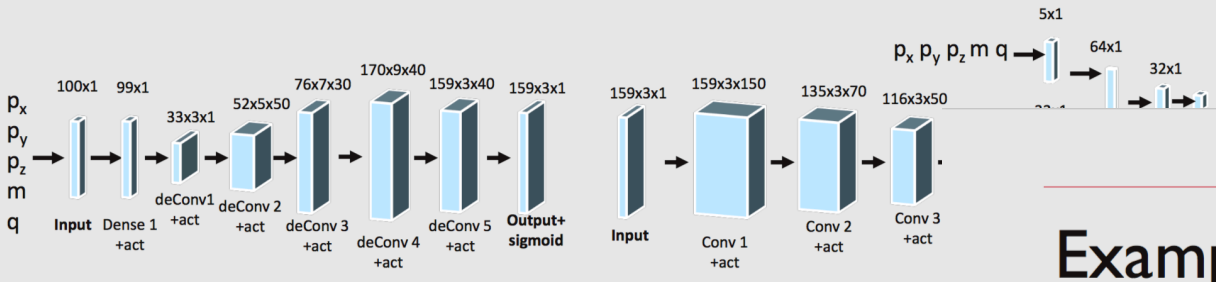
Class of networks that use convolutional and de-convolutional layers to seek for and produce meaningful patterns



Deep Conditional Convolutional GAN

CHEP 2018, **Tomasz Trzcinski** *(ALICE)*

condDCGAN: Conditional DCGAN

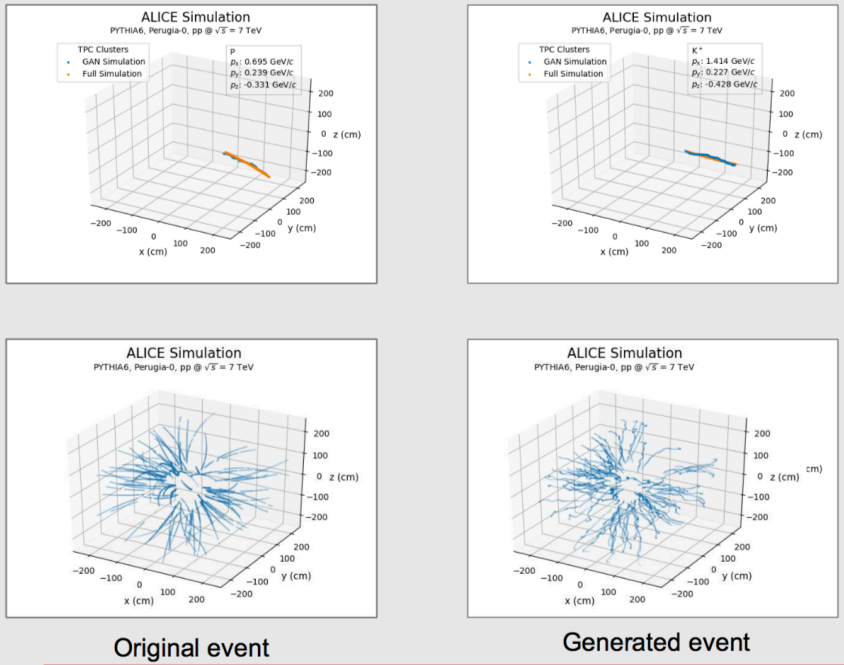


- Deep Conditional Convolutional GAN
- 2D Convolutional/ Deconvolutional Layers
- Leaky ReLU Activation

- Dropout
- Batch Normalization
- Sigmoid activation on outpu

Examples

for the conditional cluster simulation:



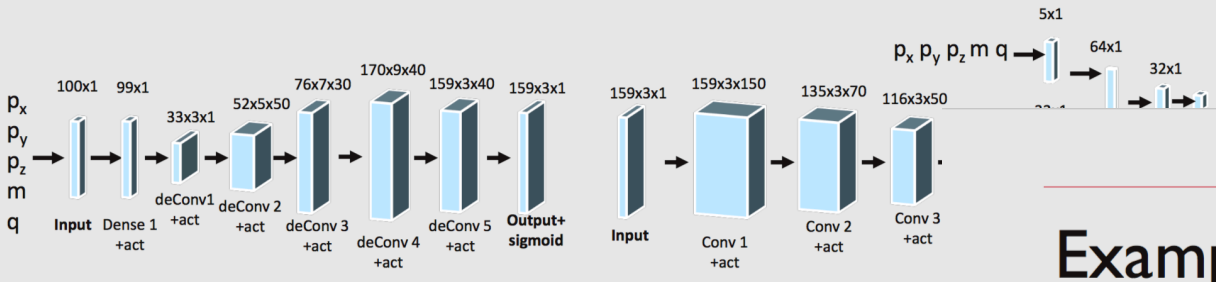
Original event

Generated event

Deep Conditional Convolutional GAN

CHEP 2018, **Tomasz Trzcinski** *(ALICE)*

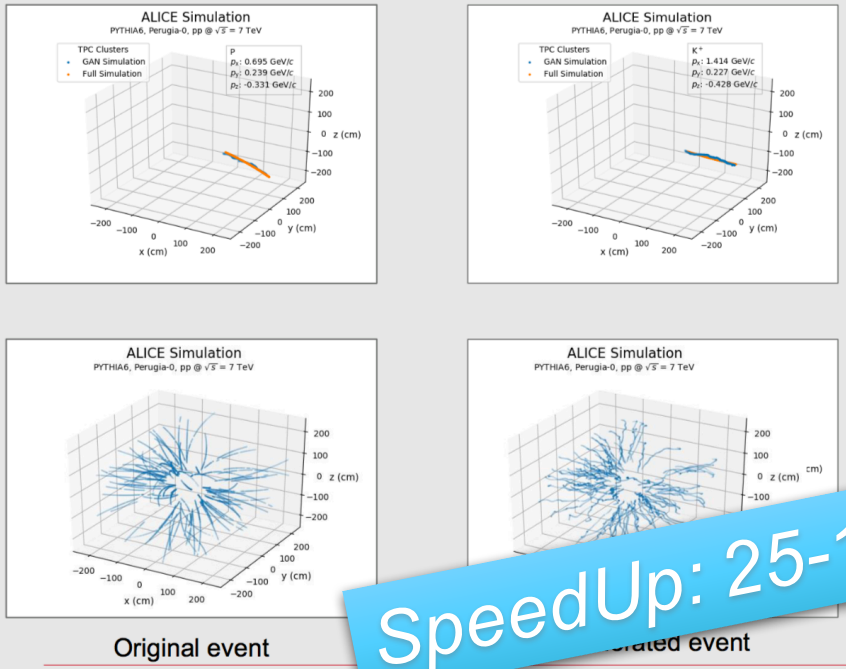
condDCGAN: Conditional DCGAN



- Deep Conditional Convolutional GAN
- 2D Convolutional/ Deconvolutional Layers
- Leaky ReLU Activation
- Dropout
- Batch Normalization
- Sigmoid activation on outpu

Examples

for the conditional cluster simulation:



Original event

SpeedUp: 25-100

FastSim LHCb

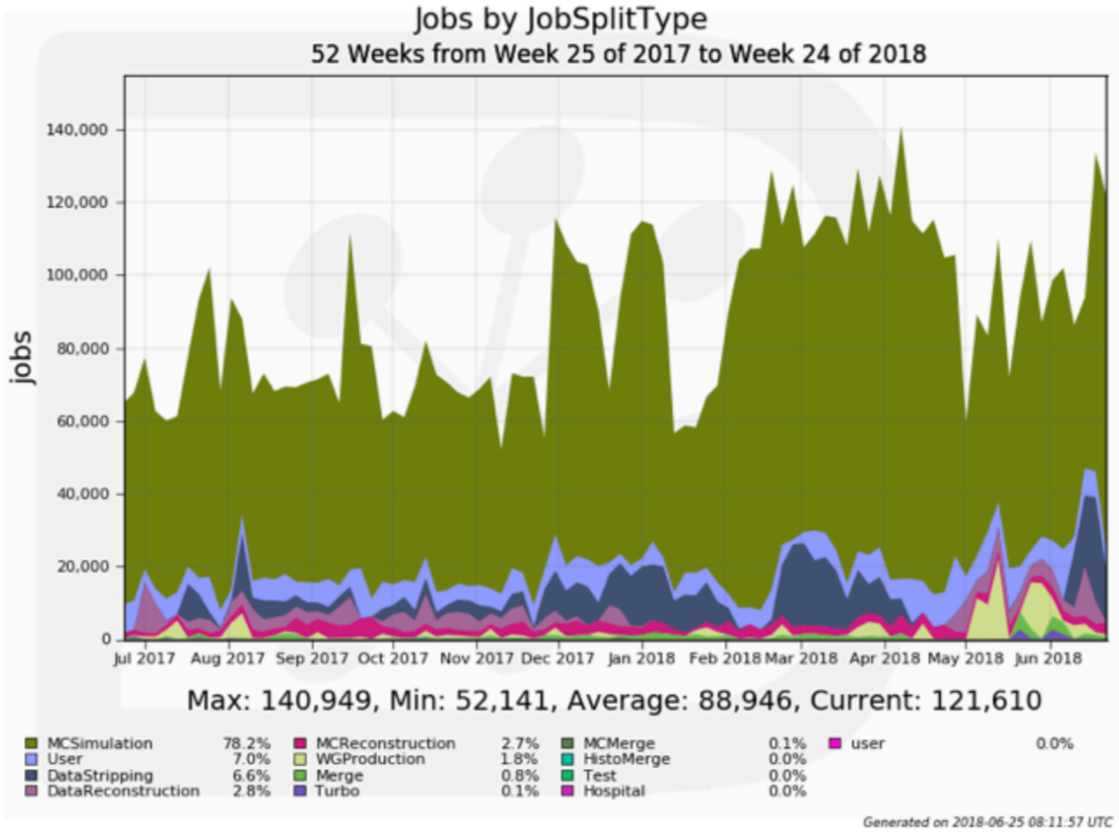
- The simulation application for the LHCb experiment is *Gauss*
  - Particle generation and transport in the detector Based on the Gaudi framework
  - Depends on a number of external libraries, including Geant4 for particle transport
  - A separate application, Boole, takes care of the digitized detector's response
- Simulation takes most of the LHCb CPU resources

Run III

Collecting more interesting events in Run III – and further – will require more events to be simulated

FastSim need

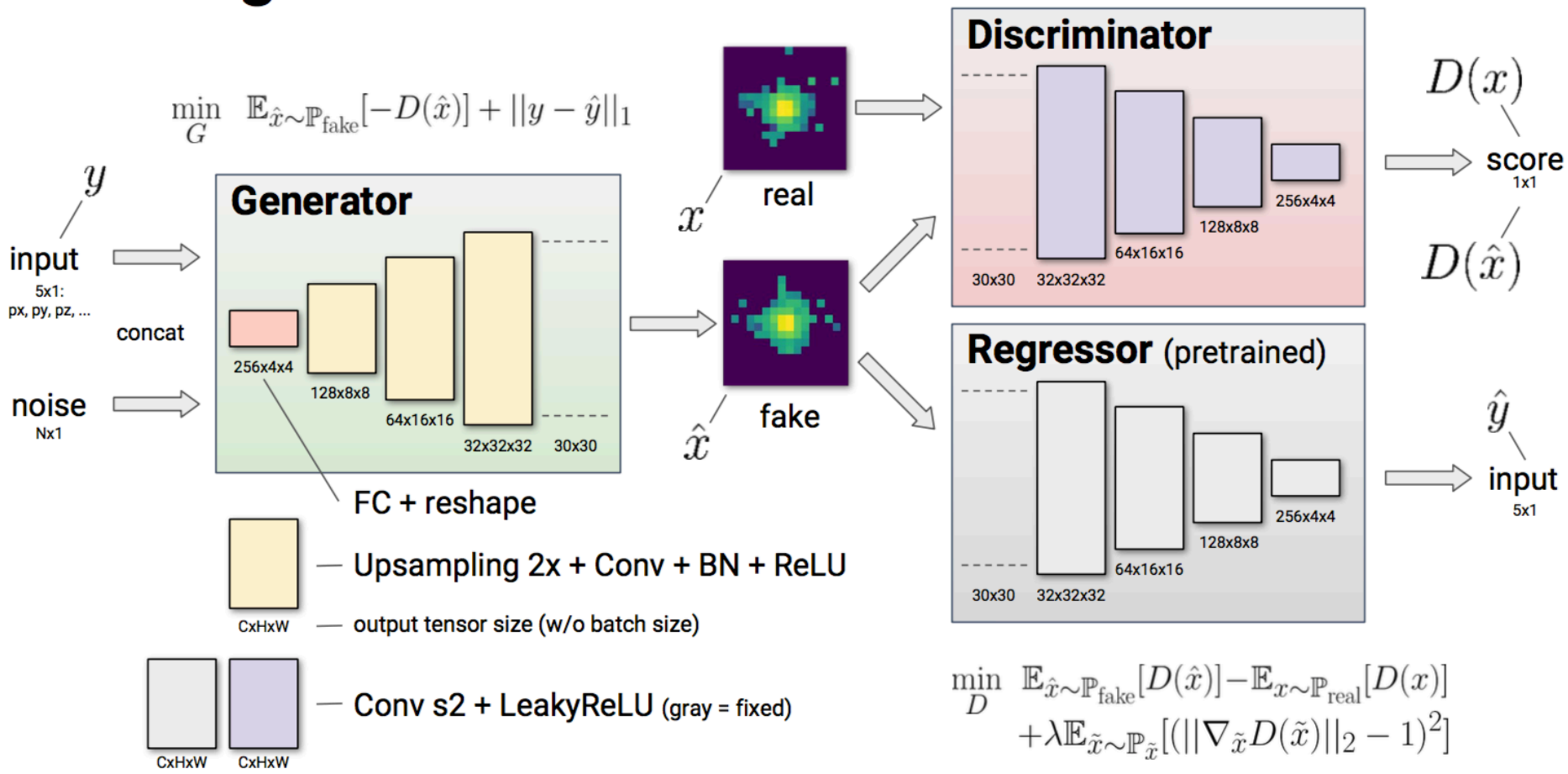
Need to shift towards a scenario where a significant fraction of LHCb MC events is fast-simulated



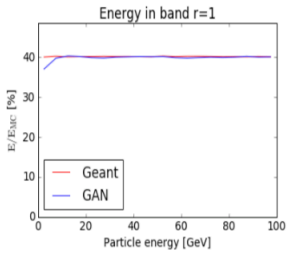
ICHEP2018, **Mark Whitehead (LHCb)**

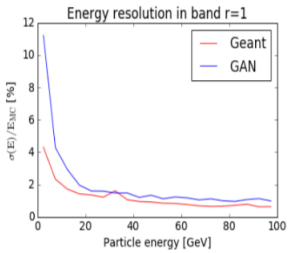
Wasserstein Conditional GAN

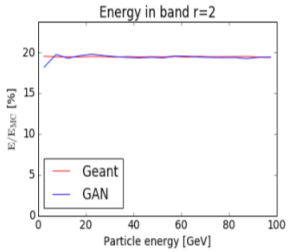
Training scheme

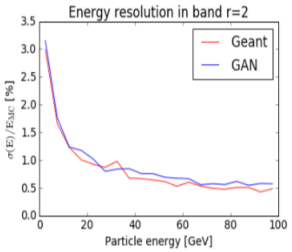


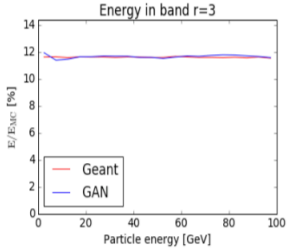
Performance

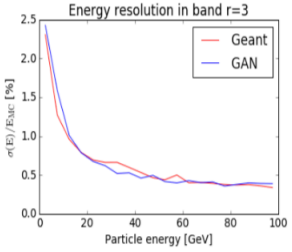


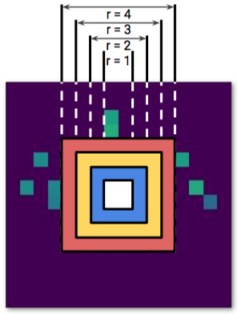


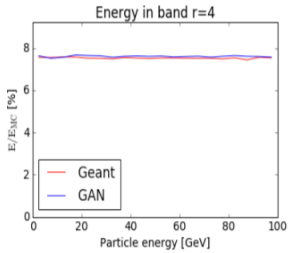


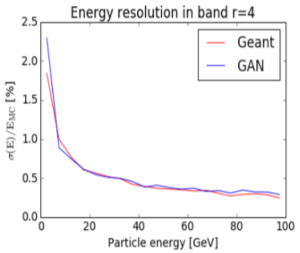












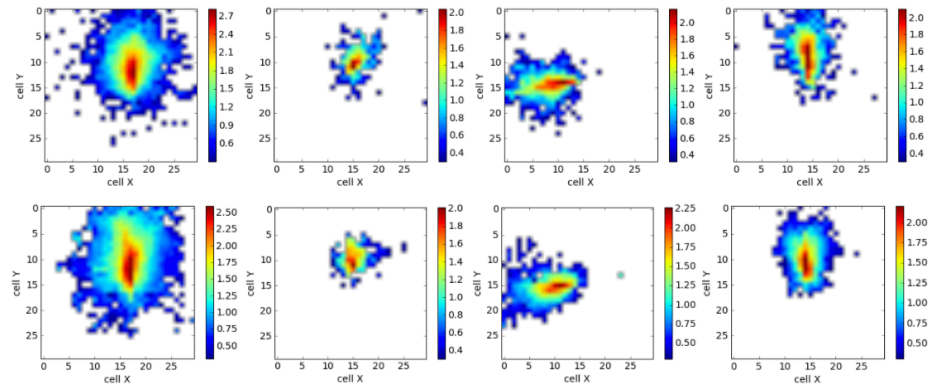
- Good reproduction of first and second moments for cluster shape

ICHEP2018, **Mark Whitehead (LHCb)**

GEANT Simulated

$\log_{10}$(cell energy)

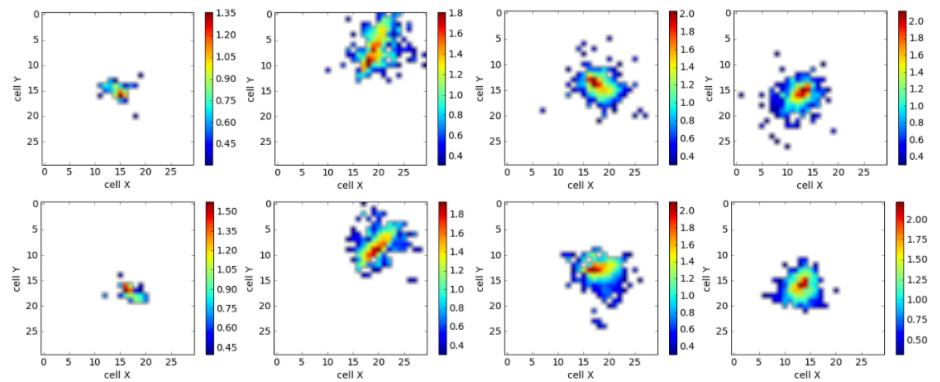
GAN Generated



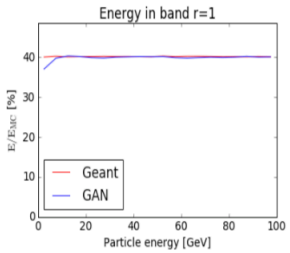
GEANT Simulated

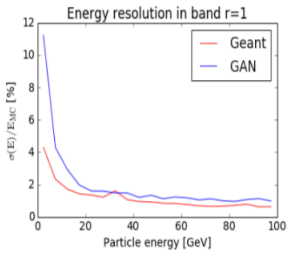
$\log_{10}$(cell energy)

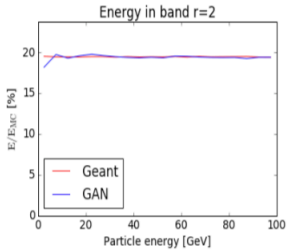
GAN Generated

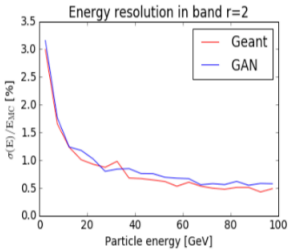


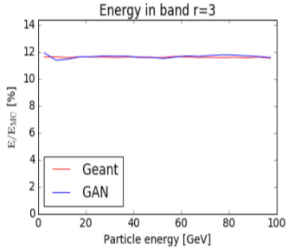
Performance

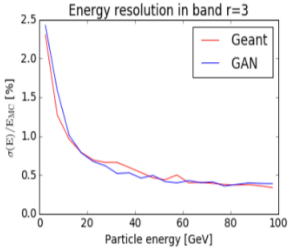


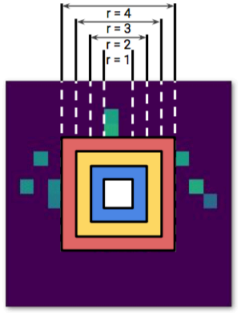


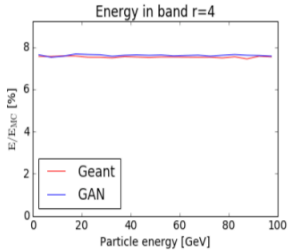


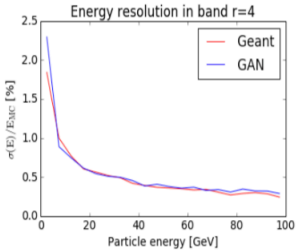












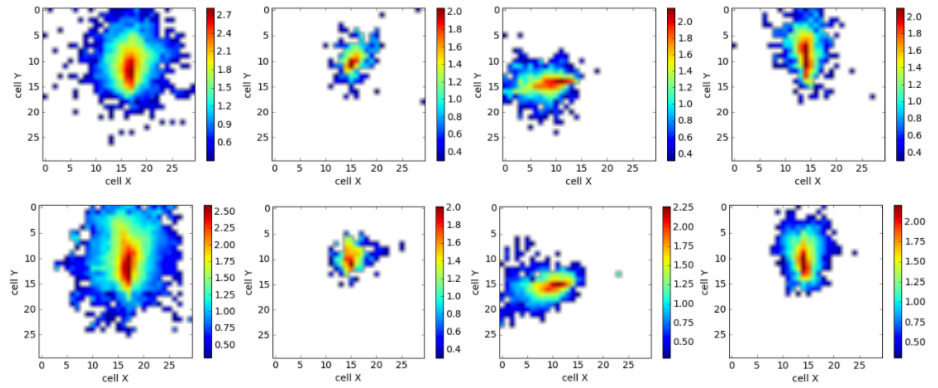
- Good reproduction of first and second moments for cluster shape

ICHEP2018, **Mark Whitehead (LHCb)**

GEANT Simulated

$\log_{10}$(cell energy)

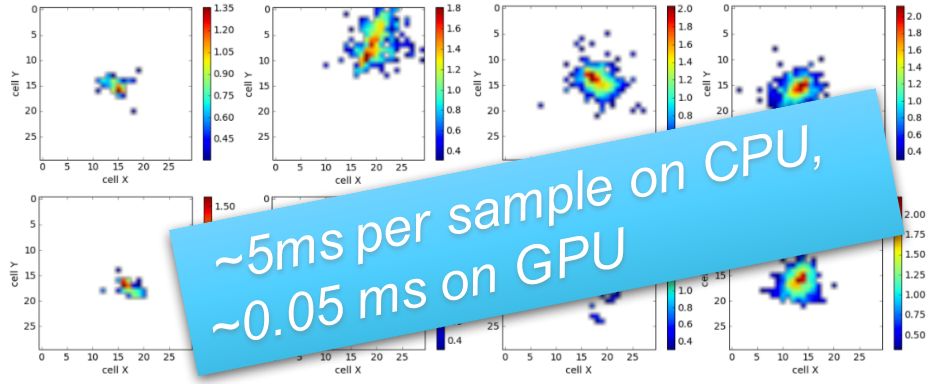
GAN Generated



GEANT Simulated

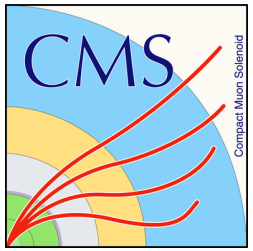
$\log_{10}$(cell energy)

GAN Generated



~5ms per sample on CPU,
~0.05 ms on GPU

FastSim CMS



CMS offline & computing week , **Sezen Sekmen** *(CMS) – Apr 2018*

Fast simulation (FastSim) is an integral part of CMS physics studies and the CMS software framework.

- Speeds up CMS event simulation ~100 times and CMS event simulation+reconstruction ~20 times.
- Regularly validated within the official CMS software release validation framework.
- Mainly validated against FullSim. Reproduces FullSim mostly by about 10%.

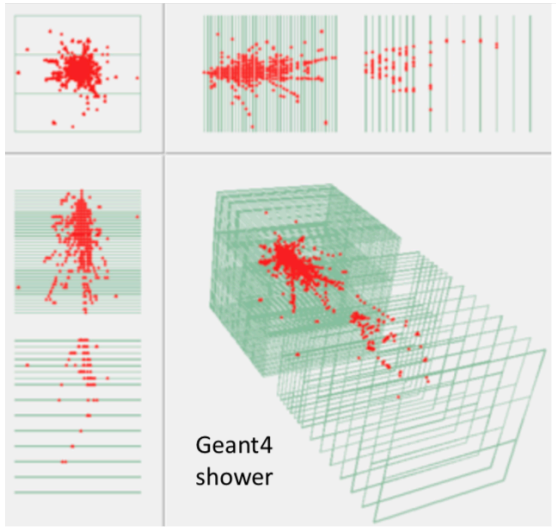
- Actively maintained by ~15 developers working part time on different aspects of the framework.

- Mainly validated against FullSim. Reproduces FullSim mostly by about 10%.

Areas of application

- Priority: Calorimetry showers:
  - Currently done with a GFLASH like parametrization.
  - Use GEANT-simulated showers for tuning.
  - Parametrization is fast. Study ML methods to improve shower simulation accuracy, which is difficult to achieve with parametrization. Expect improvements in boosted object simulation.
- Tracking simulation and reconstruction
  - Improving accurate modelling of material interactions

A generic FastSim approach



- CLIC calorimeter simulation for the proof of concept
  - Data is essentially a 3D image

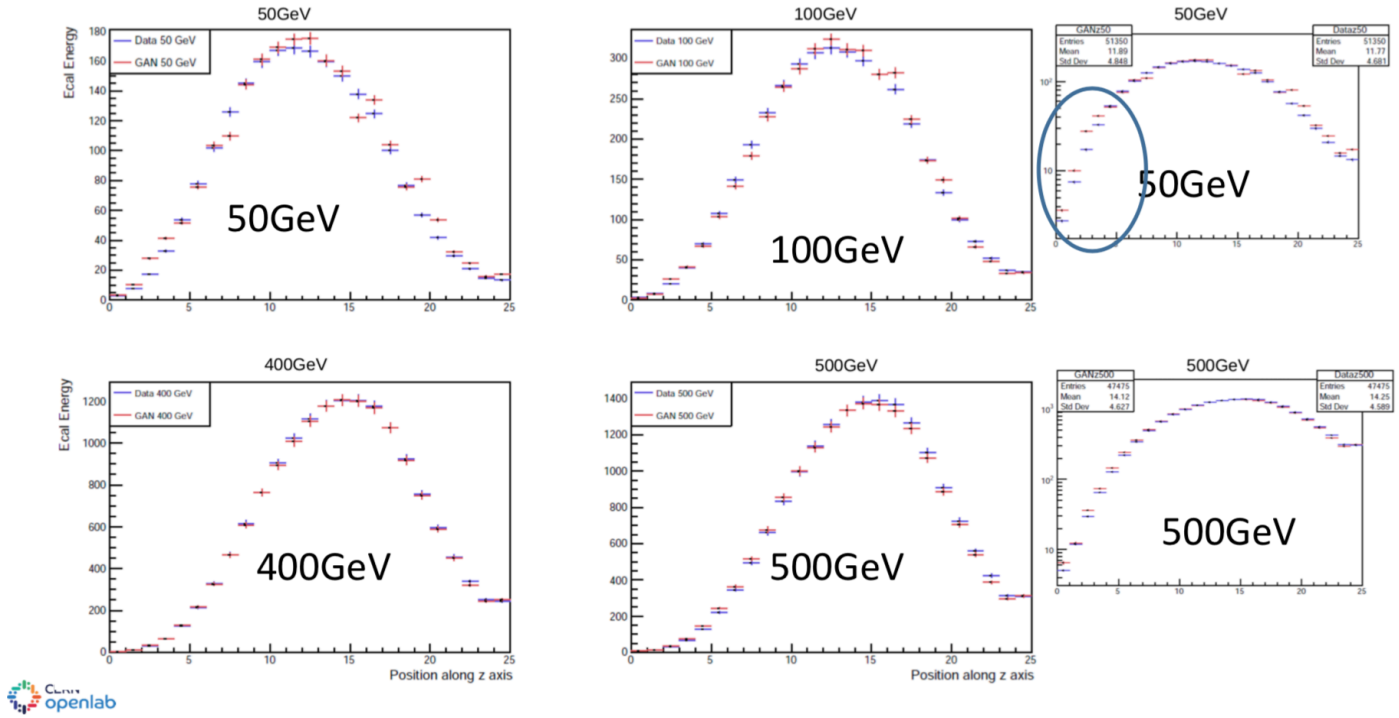
Electromagnetic calorimeter detector design(*)

(Linear Collider Detector studies)

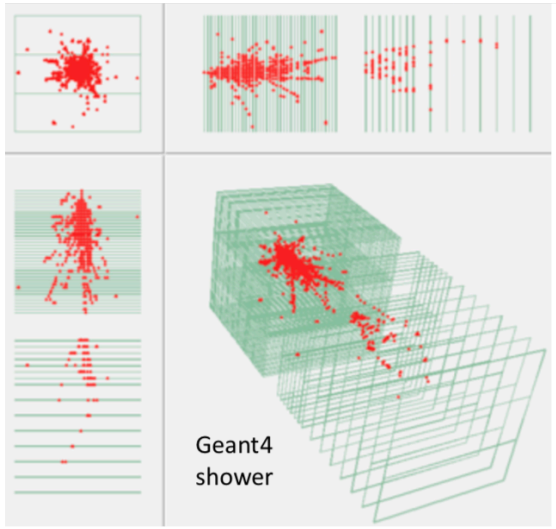
- 1.5 m inner radius, 5 mm×5 mm segmentation: 25 tungsten absorber layers + silicon sensors

1M single particle samples (e,γ,π)

- Flat energy spectrum (10-500) GeV
- Orthogonal to detector surface
- +/- 10° random incident angle



A generic FastSim approach



- CLIC calorimeter simulation for the proof of concept
  - Data is essentially a 3D image

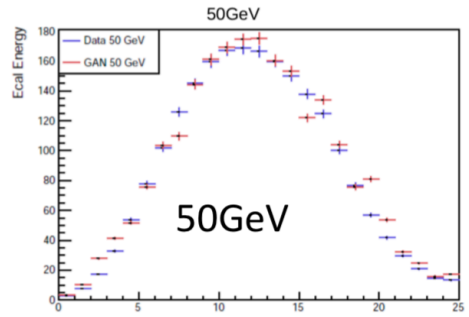
Electromagnetic calorimeter detector design(*)

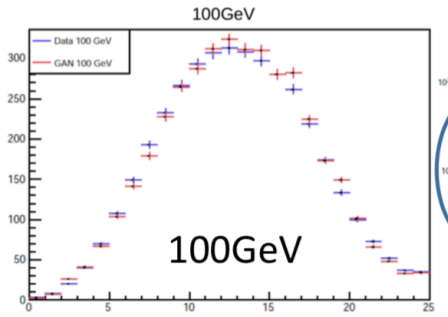
(Linear Collider Detector studies)

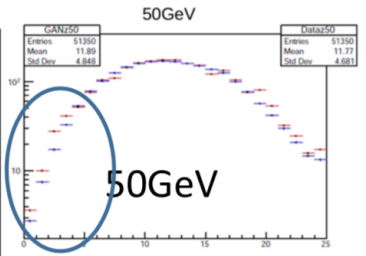
- 1.5 m inner radius, 5 mm×5 mm segmentation: 25 tungsten absorber layers + silicon sensors

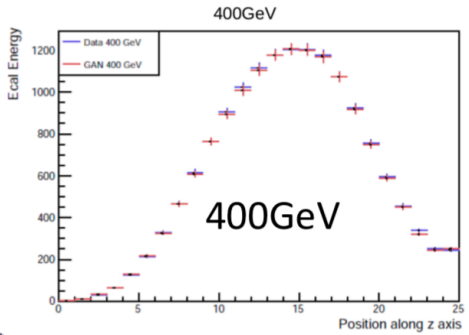
1M single particle samples (e,γ,π)

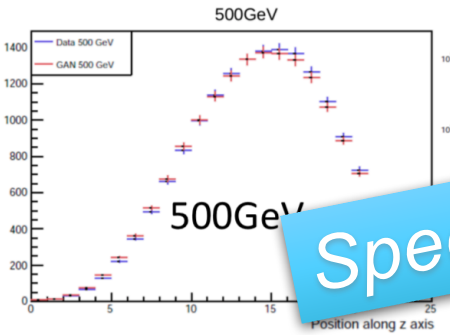
- Flat energy spectrum (10-500) GeV
- Orthogonal to detector surface
- +/- 10° random incident angle

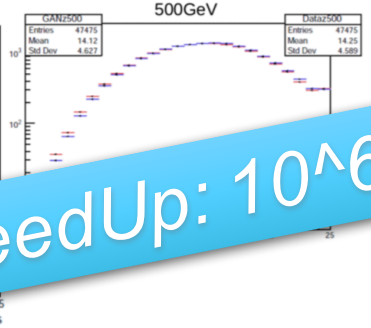










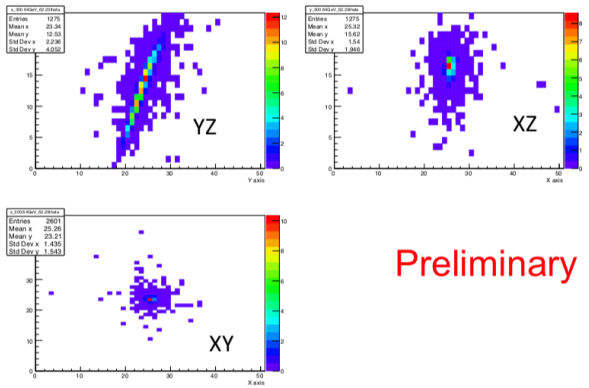


SpeedUp: 10^6

A generic FastSim approach

Generalisation

Variable angle sample



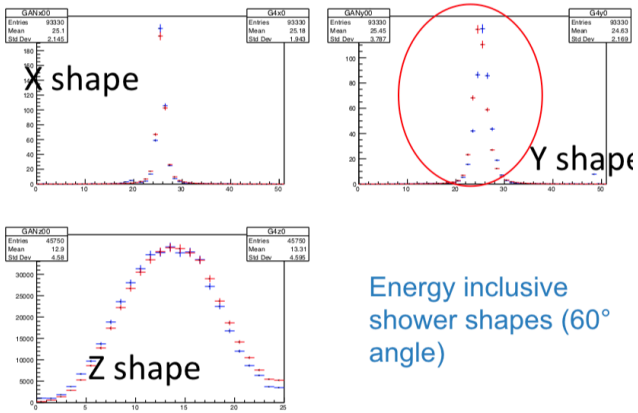
Preliminary

Adjust convolution parameters to improve energy description vs angle

Minimal architecture changes

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

Wider/asymmetric image size (51x51x25):

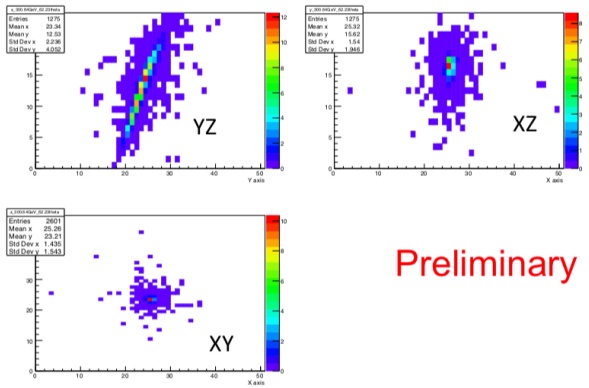


Energy inclusive shower shapes (60° angle)

A generic FastSim approach

Generalisation

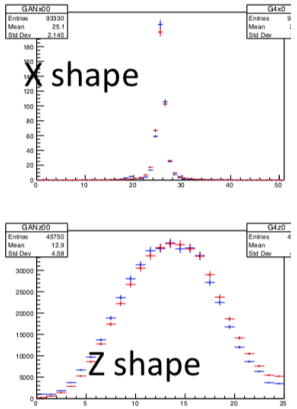
Variable angle sample



Preliminary

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

Wider/asymmetric ima

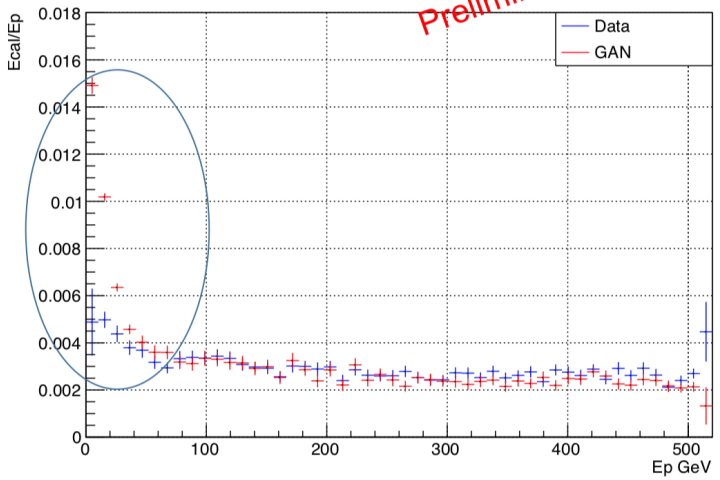


Adjust convolution parameters to improve energy description vs angle

Minimal architecture changes

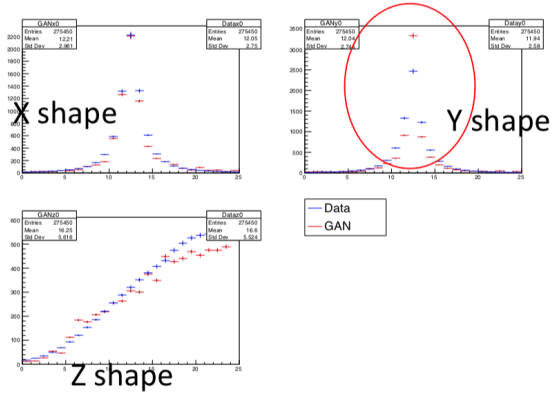
Generalisation

Charged Pions



Charged pions have small energy deposits

Energy showers are delayed along Z



Summary

- Machine Learning techniques have been already used in different applications by the HEP community
- The interest in applying ML to simulation is growing and all the experiments are investigating DL approaches
- Geant4 provides fast simulations ”hooks” and fast simulation techniques (i.e. biasing techniques)
- To what extent fast simulation/ML approaches can be generalized?
- What can we provide as a community?

The floor is open for discussion!

Thanks for your attention.

Marilena Bandieramonte
marilena.bandieramonte@cern.ch

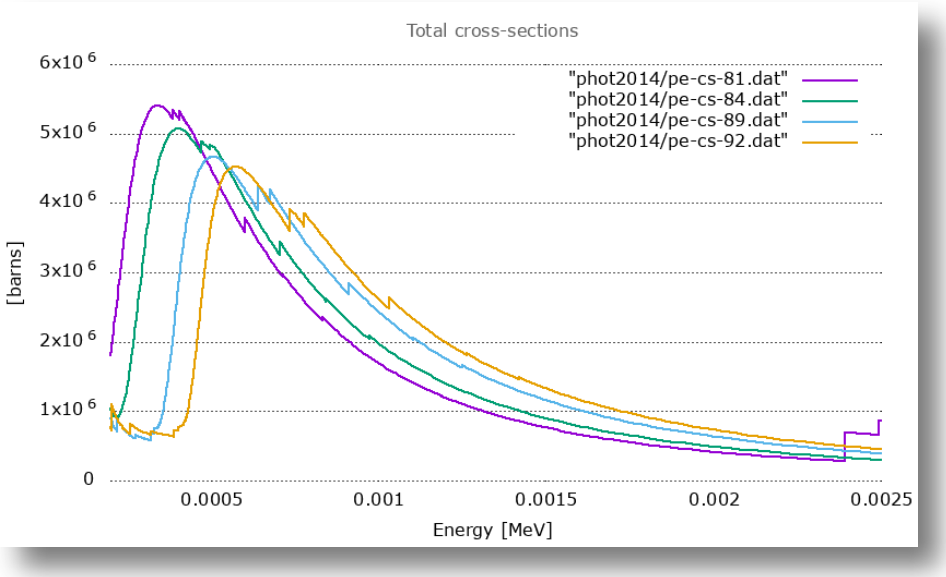
Machine learning to empower physics modeling

- Machine learning applied to FASTSIM looks very promising

- What if we go one level beyond and we replace computationally expensive physics models with ML blocks
  - Able to learn complex cross-sections shapes (total, differential)?
  - Able to directly generate the final-state?

→ From "physics-agnostic" to "**physics-aware**" neural networks

Training Physics-aware supervised neural networks[1][2]
- Embed physical-laws underlying the process
- To be used to infer physical quantities (momenta, directions, energies..)
- Both for continuous and discrete processes



[1] "QCD-Aware Recursive Neural Networks for Jet Physics", Kyle Cranmer et Al, https://arxiv.org/abs/1702.00748 - Feb 2017
[2] Physics Informed Deep Learning: Data-driven Solutions of Nonlinear Partial Differential Equations, Maziar Raissi et Al, https://arxiv.org/abs/1711.10561 - Nov 2017

A palette of fast simulations in LHCb

ICHEP2018, **Mark Whitehead (LHCb)**

01 Simplified detector simulation

- Reduced detector: RICH-less or tracker-only. *In production*
- Calorimeter showers fast simulation. *Under development*
- Muon low energy background, used with full muon detector simulation. *In production*

02 Simulation of partial events

- Simulate only particles from signal decay. *In production*
- ReDecay, e.g. use N-times the non-signal decay part of the event. *In production*

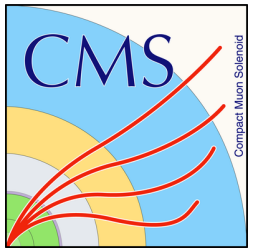
03 Fully parametric simulation

- Parametrized tracking, calorimeter and particleID objects with a DELPHES-based infrastructure. *Under development*

No single solution for all needs, but different options organized under the unique *Gauss* framework

Deploy solutions when mature for physics

FastSim CMS



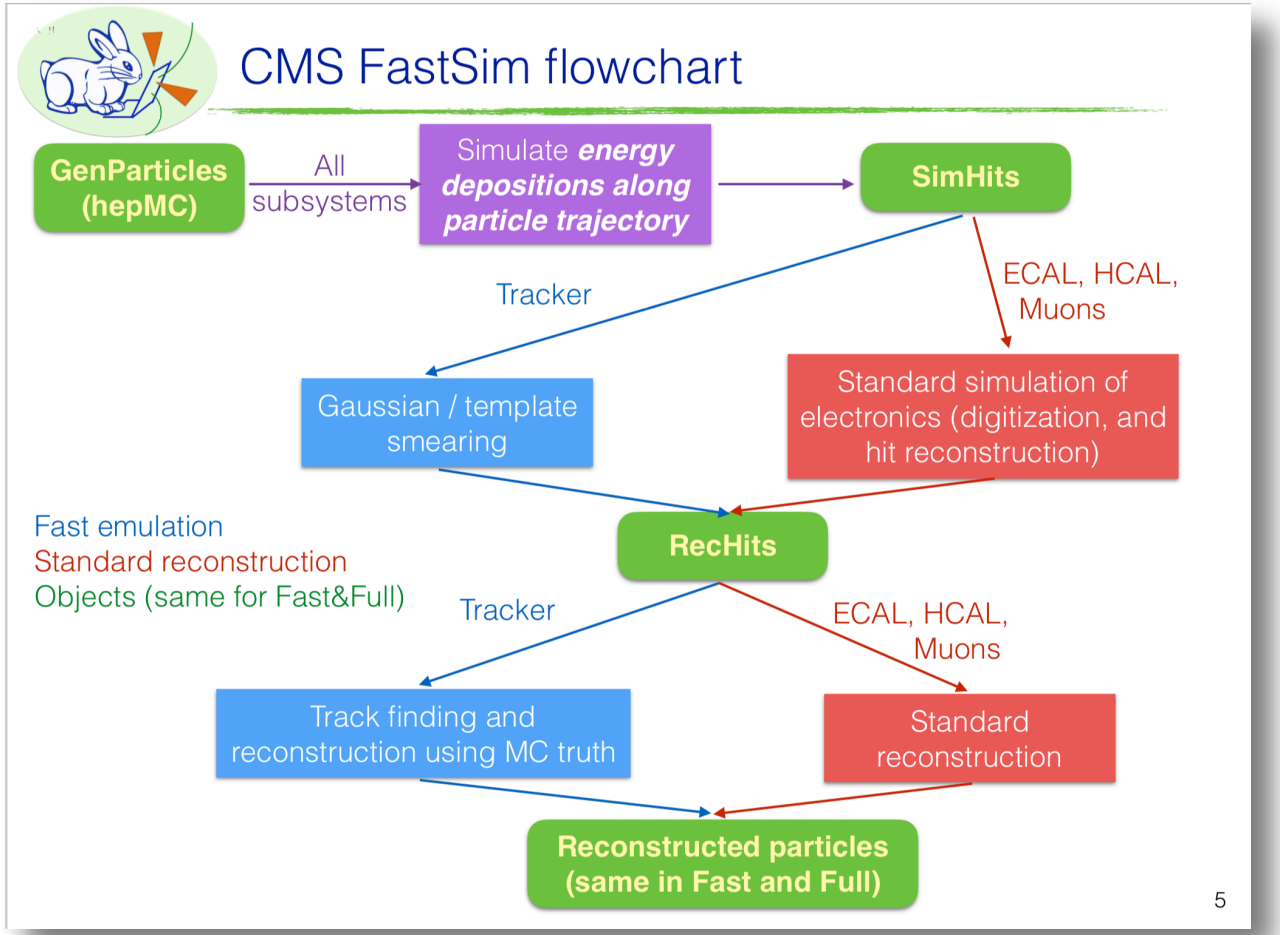
Fast simulation (FastSim) is an integral part of CMS physics studies and the CMS software framework.

- Speeds up CMS event simulation ~100 times and CMS event simulation+reconstruction ~20 times.
- Regularly validated within the official CMS software release validation framework.
- Mainly validated against FullSim. Reproduces FullSim mostly by about 10%.

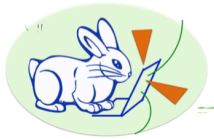
- Actively maintained by ~15 developers working part time on different aspects of the framework.

- Mainly validated against FullSim. Reproduces FullSim mostly by about 10%.

LPCC 2017, **Sezen Sekmen** *(CMS)*



CMS Fast Sim



Why is FastSim fast?

CMS FastSim concept: CMS FastSim is a single, uniquely-defined framework (as opposed to e.g. ATLAS, which consists of several different levels of simulation).

Main difference wrt FullSim is in the simulation step. Low level quantities are parametrized.

- Geometry is simplified.
- Material interactions are simplified and parametrized.
- Calorimetric showers are parametrized.

Hit reconstruction (RecHits) mostly follows standard reconstructi
(applied to FullSim and data). Exception:

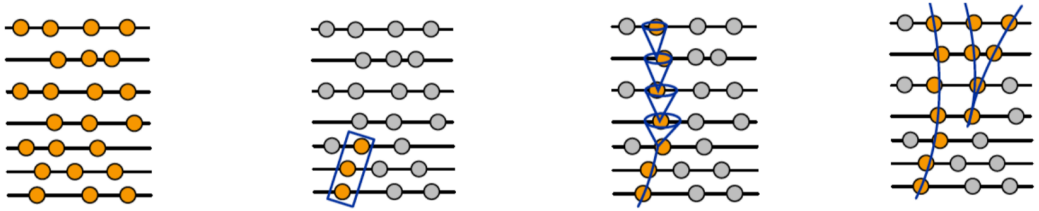
- Tracking RecHits: No digitization and local reco in tracker. RecHits emulated by smearing SimHits.

Object reconstruction mostly follows standard reconstruction
(applied to FullSim and data). Exception:

- Track reconstruction / finding emulated with help from MC truth

LPCC 2017, **Sezen Sekmen** *(CMS)*

Data/FullSim: Combinations of hits need to identified from a nearly infinite number of hit permutations created by charged particle trajectories, bent by the B field.



RecHits from charge deposits

seeding: find start of potential trajectories

trajectory finding: add hits to the seed that supports trajectory hypothesis

trajectory fit: global fit to estimate track parameters.

FastSim track reco: Restrict seeding and trajectory finding to only a local subset of hits using MC truth information. Large speed up by skipping permutations.

RecHits from charge deposits

Look up particle truth information

Create subsets of consistent RecHits

FastSim CMS

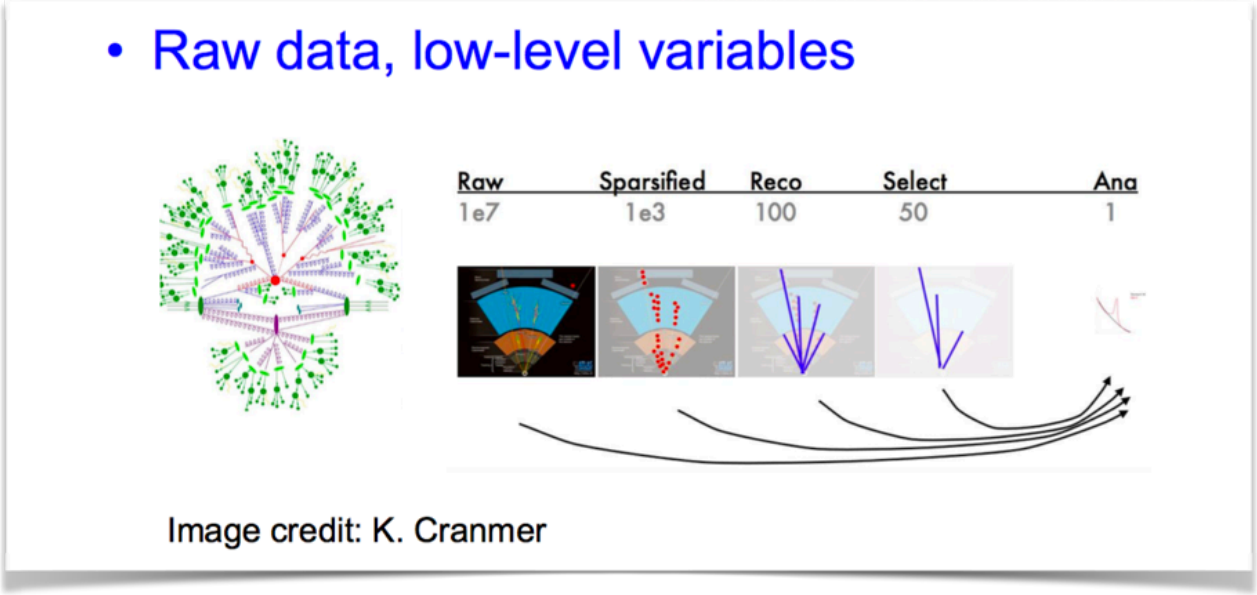
Existing Improvements as of end of 2017

- **Static library**: avoid calls to procedure linkage table (PLT) for dynamic loading of libraries
- **Production cuts**: 0.01mm (pixel), 0.1mm (strip tracker), 1 mm (ECAL/HCAL), 0.002 mm (muon systems), 1 cm (support structure)
- **Tracking cut**: 2 MeV (within beampipe) → avoid looping electrons
- **Time cut**: 500 ns
- **Shower library**: use pre-generated showers in forward region (HF, ZDC, Castor)
- **Russian roulette**: discard N-1 neutrons < 10 MeV or gammas < 5 MeV (in calorimeters), retain N^{th} particle and assign it a weight of N
- **FTFP_BERT_EMM**: modified physics list, simplified multiple scattering model for most regions (default used for HCAL, HGCal)
  - When all optimizations applied together, CMS achieves ~3–5× speedup!

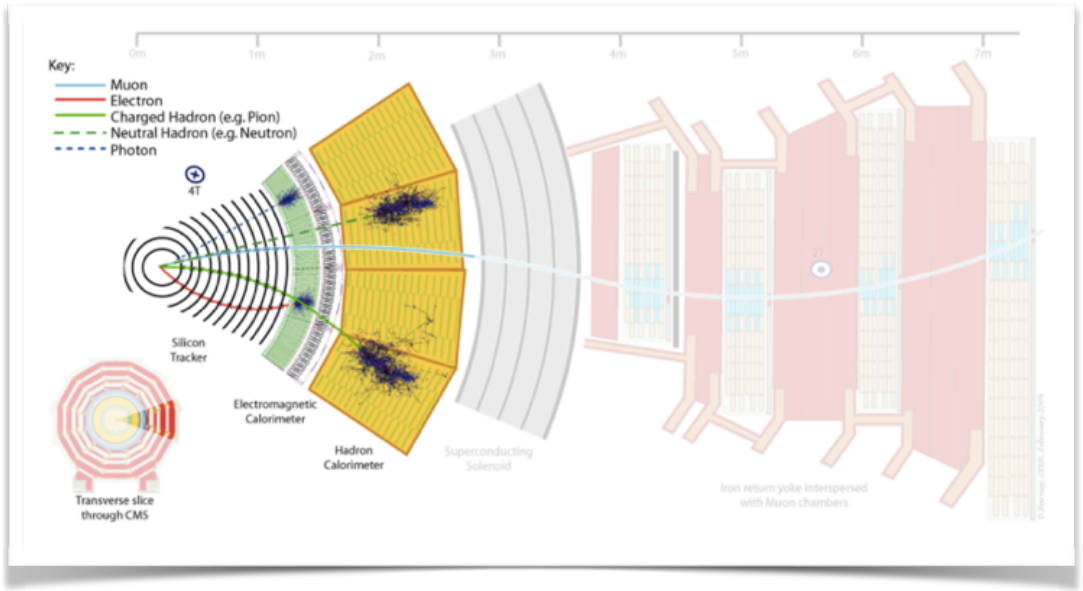
Fermilab

End-to-end learning

- All varieties of deep learning gaining traction
  - Convolutional, Recurrent, LSTM, GANs
  - Tree-based methods (XGBoost) still maintain some competitiveness
- Machine learning models increasingly used together with low-level information

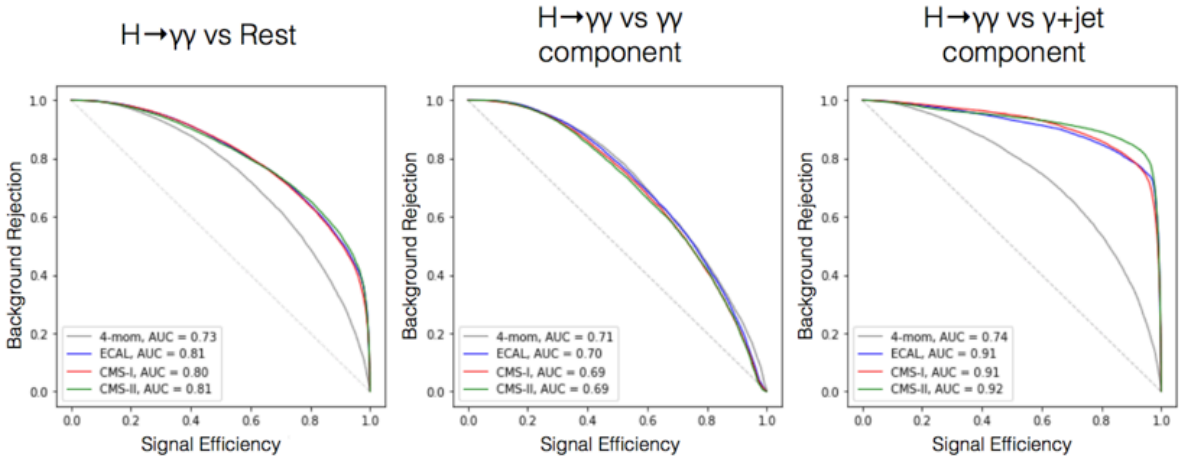


End-to-end learning



- PARTICLE AND EVENT ID CLASSIFIER WITH CNN
  - Able to learn particle kinematics and shower shapes
  - Classifier output can be de-correlated from mass of signal resonance
  - Well-suited to decays where particles can't be fully resolved/ reconstructed
  - Can tackle arbitrary decays: train on whole Standard Model on same network

Event ID: Results*, Barrel+Endcap



- Similar performance as before ⇒ scale well to multiple subdetector images
- Subdetectors other than ECAL mostly contain noise from PU or underlying event ⇒ little to no penalty in including additional noisy subdetector images
- Not very sensitive to choice of geometry segmentation (in this study)

*Using 2012 CMS Simulated Open Data