Faculty of Physics

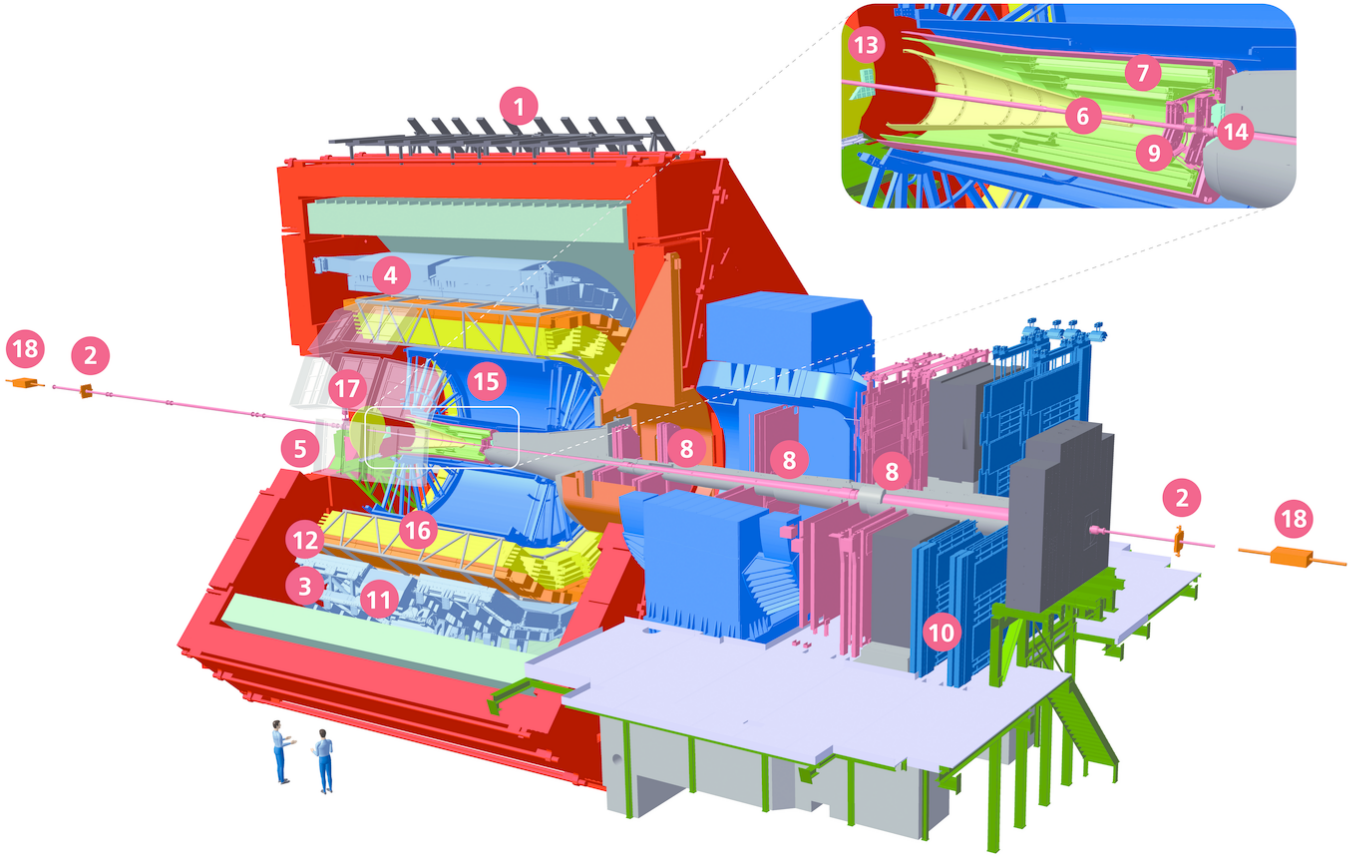
WARSAW UNIVERSITY OF TECHNOLOGY

Using Machine Learning methods for improving data quality in ALICE

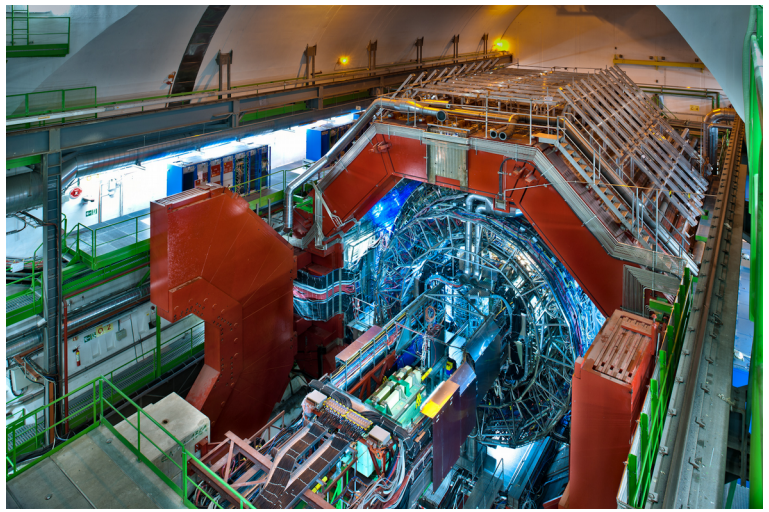
Łukasz Graczykowski
Tomasz Trzciński, Kamil Deja, Michał Glinka
for the ALICE Collaboration

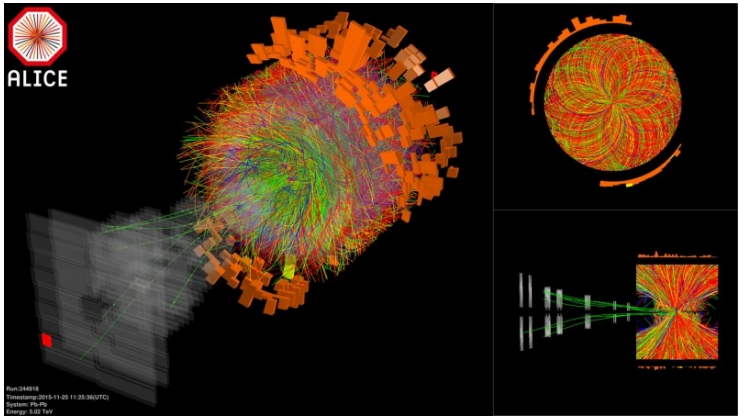
XIII Quark Confinement and the Hadron Spectrum
Maynooth, Ireland
2 August 2018

The ALICE experiment



1. **ACORDE** | ALICE Cosmic Rays Detector
2. **AD** | ALICE Diffractive Detector
3. **DCal** | Di-jet Calorimeter
4. **EMCal** | Electromagnetic Calorimeter
5. **HMPID** | High Momentum Particle Identification Detector
6. **ITS-IB** | Inner Tracking System - Inner Barrel
7. **ITS-OB** | Inner Tracking System - Outer Barrel
8. **MCH** | Muon Tracking Chambers
9. **MFT** | Muon Forward Tracker
10. **MID** | Muon Identifier
11. **PHOS / CPV** | Photon Spectrometer
12. **TOF** | Time Of Flight
13. **T0+A** | Tzero + A
14. **T0+C** | Tzero + C
15. **TPC** | Time Projection Chamber
16. **TRD** | Transition Radiation Detector
17. **V0+** | Vzero + Detector
18. **ZDC** | Zero Degree Calorimeter





Goals

- Use ALICE and its data as a unique environment to advance the Machine Learning field of science
- Identify areas where both ALICE (or HEP in general) and ML communities can mutually benefit
- Focus on Machine Learning research rather than using standard ML tools for ALICE use cases
- Disclaimer:
  - I'm a physicist without a ML expertise – just started my (human) learning of machine learning :)
  - My task is to guide and coordinate the work of WUT ML computer scientists within ALICE

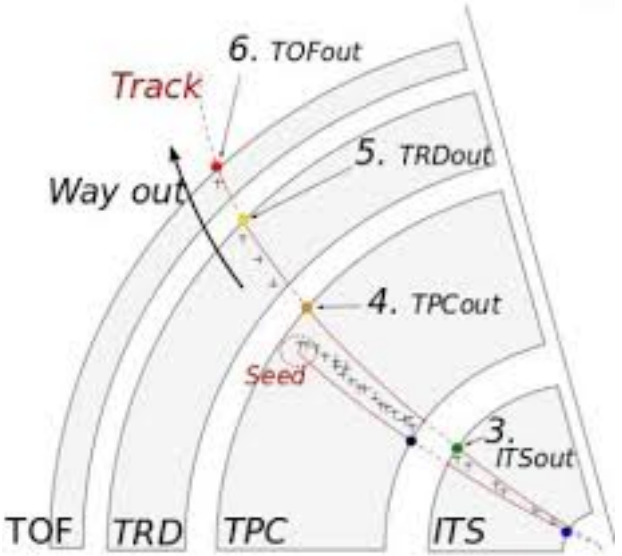
Three areas of research

- Data Quality Assurance – prediction of detector quality label assignment
  - not covered in this talk
- Simulation of TPC clusters in Monte Carlo data using generative networks
- Development of more precise particle identification (PID)

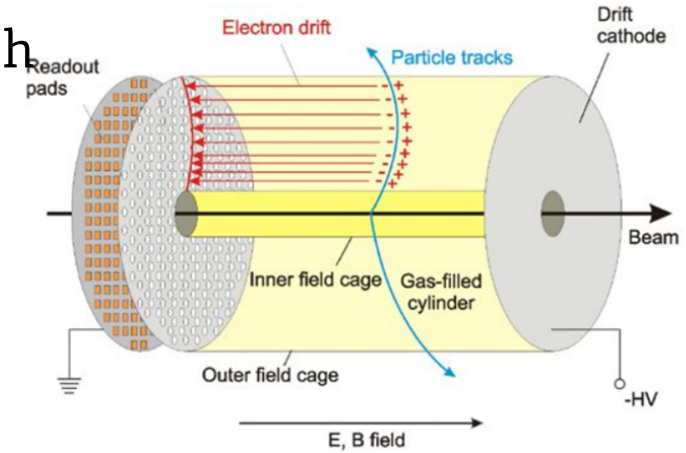
Simulation of TPC clusters in Monte Carlo data using generative networks

Time Projection Chamber

- Tracking in ALICE is performed by ITS, TPC, TRD and TOF
- First attempts – focus on the TPC only:
  - main tracking device
  - located from 0.8 m (inner radius) to 2.5 m (outer radius) from the beam and extending ~2.5 m in each direction along the beam axis
  - volume of 95 m^3
  - filled with Ne-CO_2 gas mixture (90%-10%)
  - **clusters** - points in 3D space, together with the energy loss, which were presumably generated by a particle traveling through
  - provides up to 159 clusters per track



ALICE Data Preparation Group

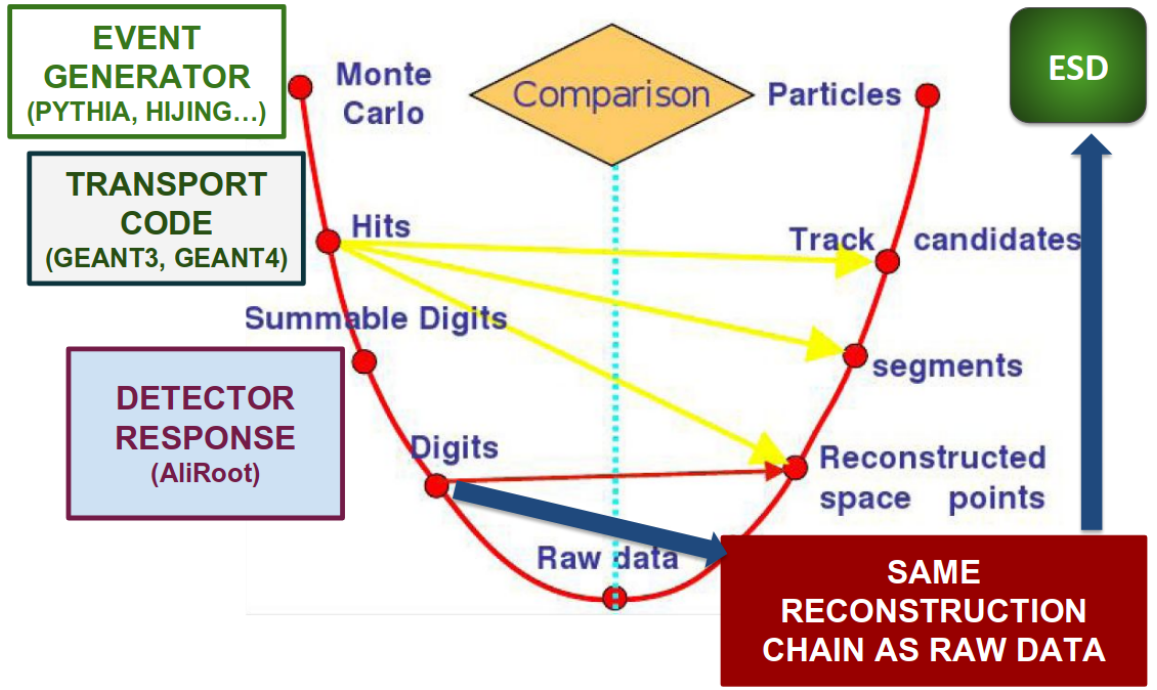


I.Konorov, Front-end electronics for Time Projection chamber

Simulation and reconstruction

- Current process relies on 5 independent modules
- The computationally most expensive module is particle propagation through the detector's matter

Collision generator → Particle propagation → Electronic signals (digits) → Digits to Clusters → Tracking

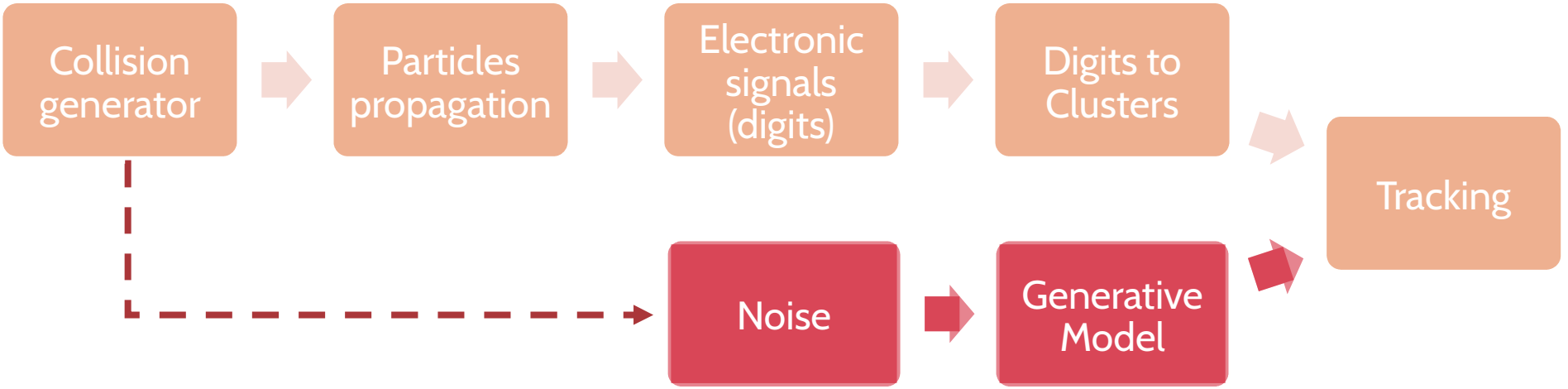


Monte Carlo simulation

Real data

ALICE Data Preparation Group

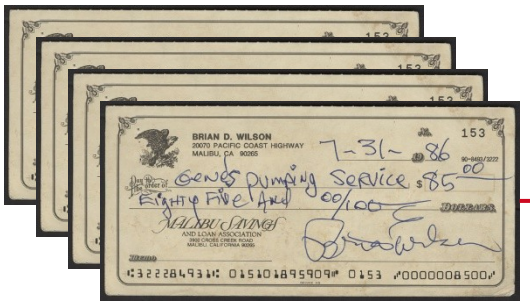
Simulation and reconstruction



- **Generative solution for cluster simulation:**
  - substitute the detector simulation and check for the speed-up
  - full simulation **still needed** to generate training samples
  - **immediate drawback:** quality of such MC data can be either comparable or lower than the full detector simulation – limits potential applications

Generative Adversarial Networks

- Generative Adversarial Network (GAN) is a neural network architecture of two networks competing with each other (playing a min-max game)
  - **"Generator"** is trained to produce fake data resembling the real data
  - **"Discriminator"** predicts whether an example data is real or fake

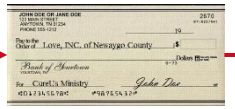


https://33milesinnewaygocounty.files.wordpress.com

Discriminator

https://thechive.files.wordpress.com

Generator

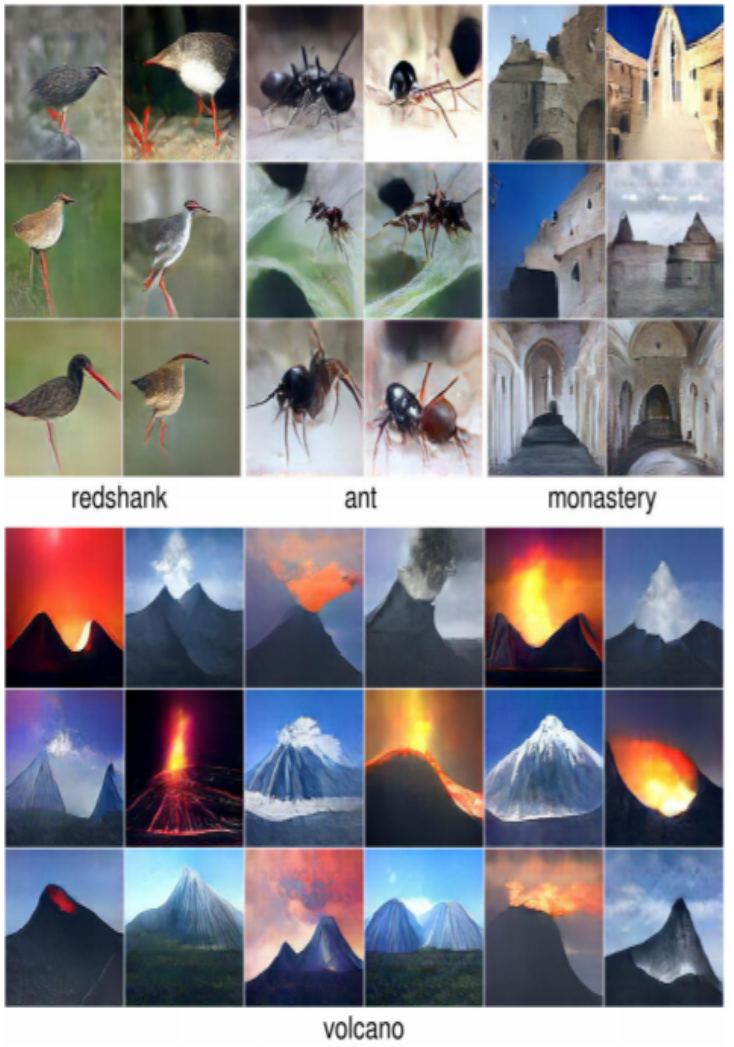


https://giphy.com/gifs/leonardo-dicaprio-catch-me-if-you-can-5leocharacters-t1h4nnWEWKfn2

Generative Adversarial Networks

- Typical use cases:
  - mainly generation of photo quality fake images (i.e. of celebrities)

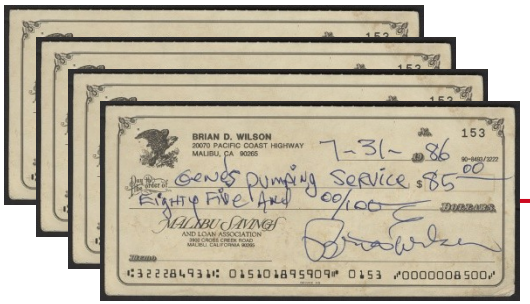
https://arxiv.org/abs/1710.10196



https://arxiv.org/pdf/1612.00005v1.pdf

Generative Adversarial Networks

- **Extending the GAN architecture – provide a set of initial parameters for the generator and discriminator:**
  - generator would not generate a random output, but a customized one
  - in our case: initial momenta of Monte Carlo particles



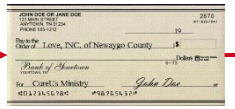
https://33milesinnewaygocounty.files.wordpress.com

Discriminator

https://thechive.files.wordpress.com

Generator

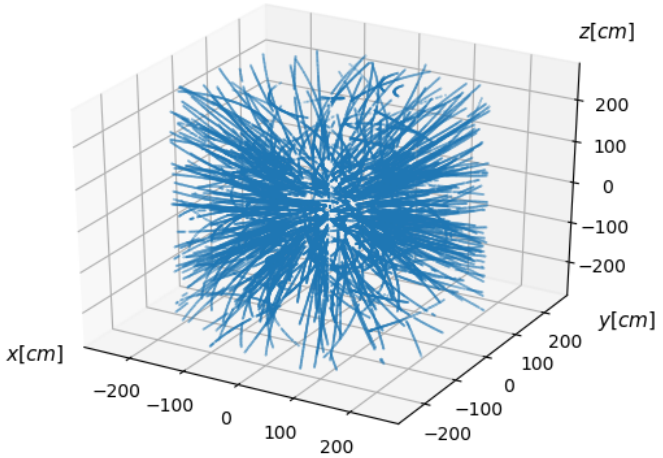
https://giphy.com/gifs/leonardo-dicaprio-catch-me-if-you-can-5leocharacters-t1h4nnWEWKfn2

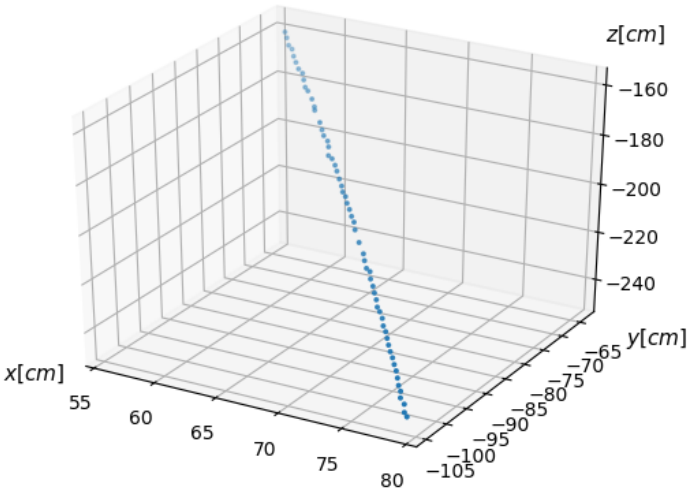


Initial Parameters

TPC clusters with GANs

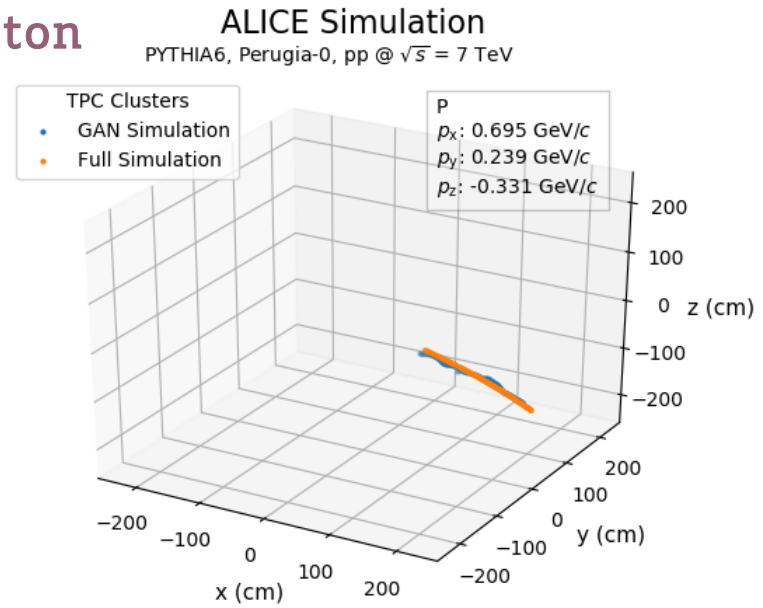
- It is not (yet!) possible to generate the full 3D image of the event at once (especially in the Pb-Pb event)
- Our solution is to:
  - generate clusters for single particles
  - two separate flows for spatial coordinates (x,y,z) and the energy
  - in the beginning focus only on 3D coordinates
  - merge generated samples (individual particles) into full images
  - training of the GAN on original full simulations



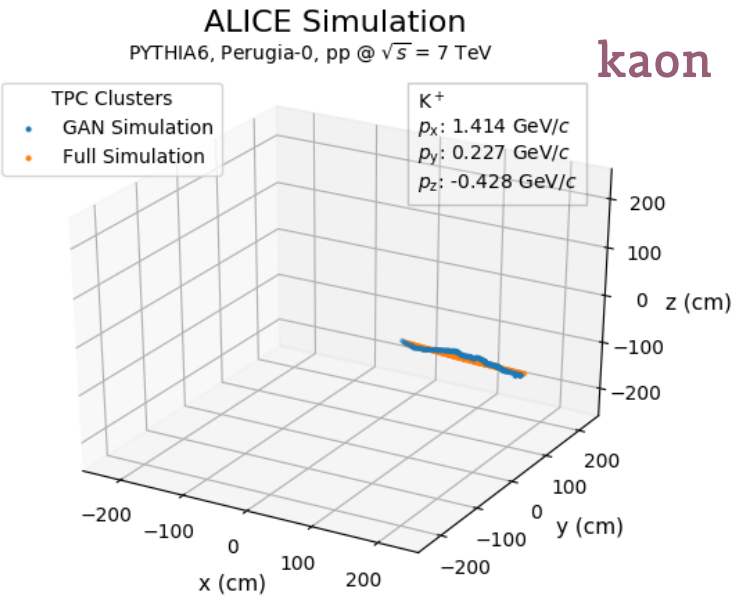


Example results

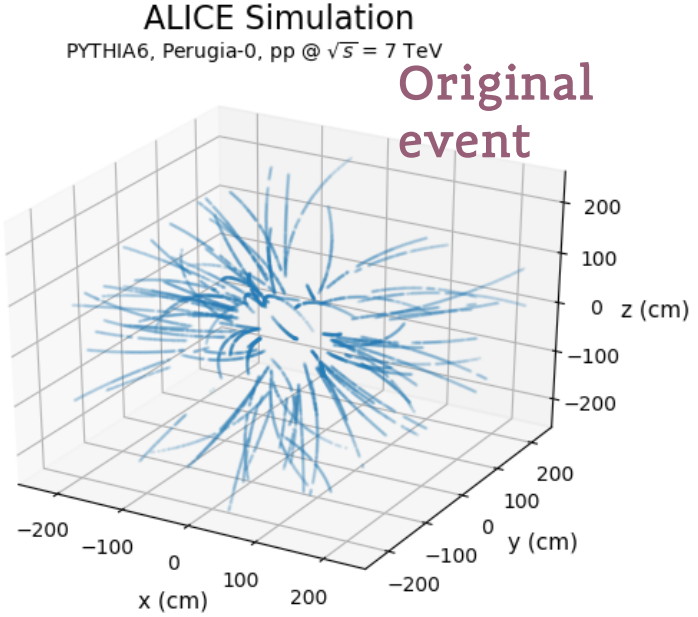
proton



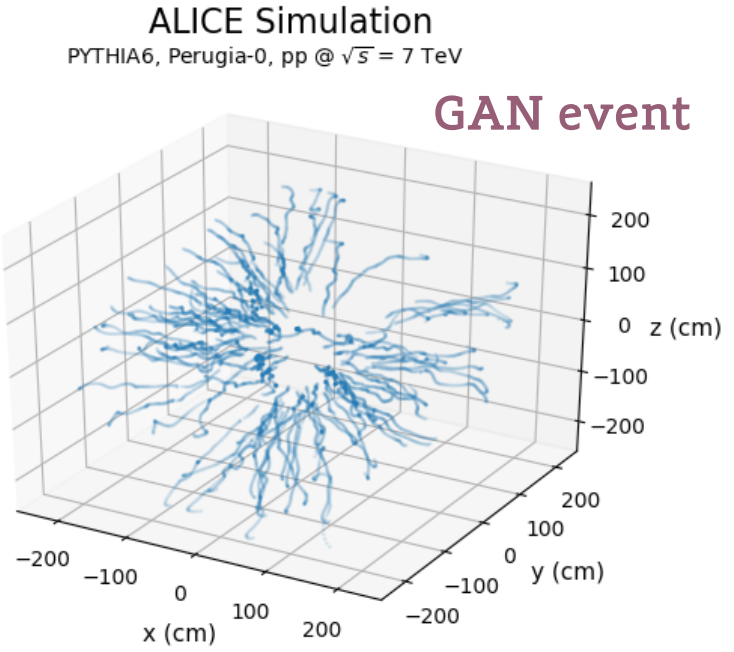
kaon



Original event



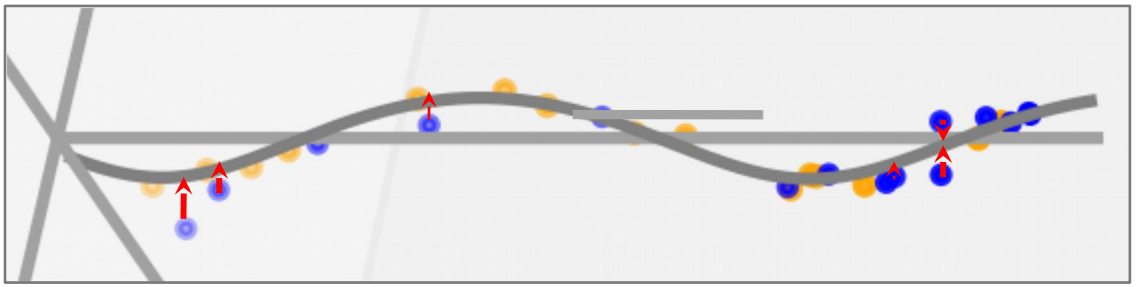
GAN event



Results

- Mean Squared Error (MSE) from the original helix as a quality measure
- Evaluation conducted on the separate test-set with ~15000 tracks

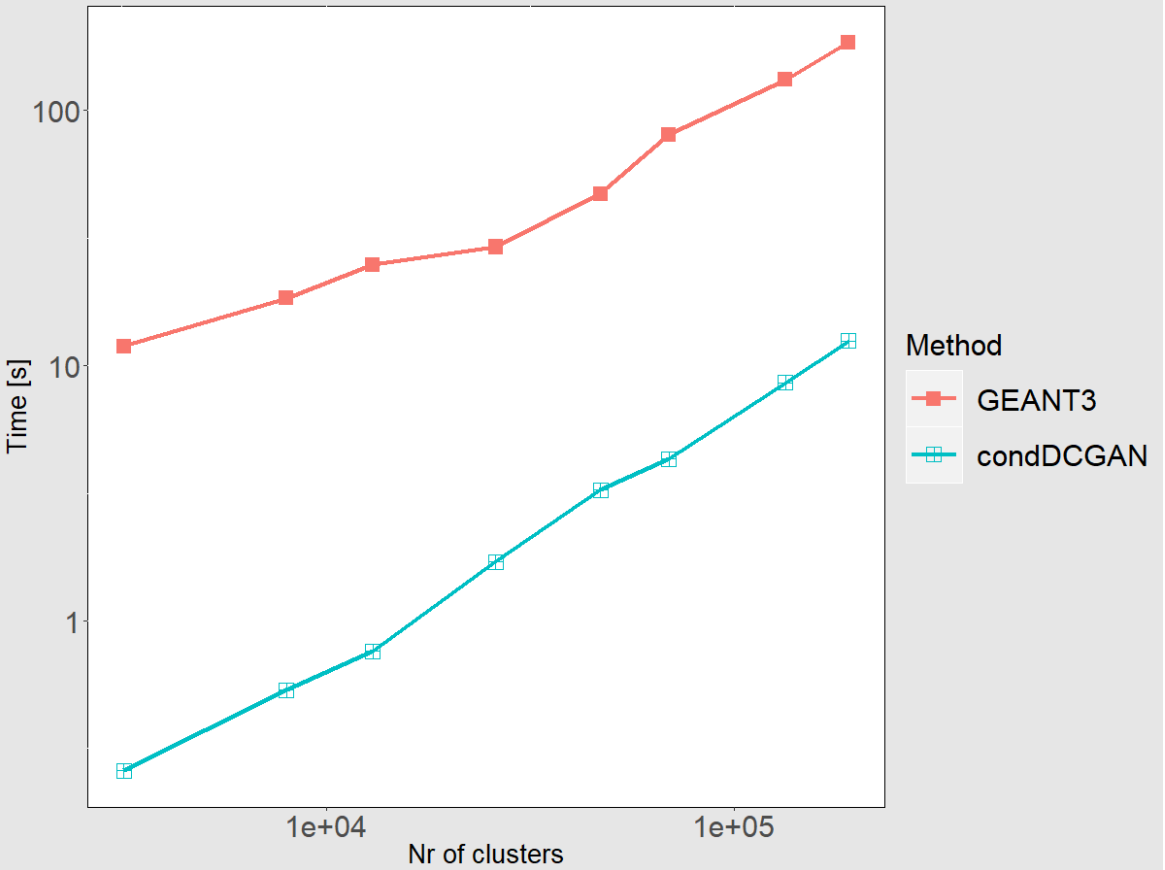
MSE visualisation:
Red - error
Grey- ideal helix
Orange – original clusters
Blue – generated clusters



Method	Mean MSE (mm)	Median MSE (mm)	Speed-up
GEANT3	1.20	1.12	1
Random (estimated)	2500	2500	N/A
condLSTM GAN	2093.69	2070.32	100
condLSTM GAN+	221.78	190.17	
condDCGAN	795.08	738.71	25
condDCGAN+	**136.84**	**82.72**	

Computational cost

- Performance test conducted on the standalone machine with Intel Core i7-6850K (3.60 GHz) CPU using single core and no GPU
- Additional order of magnitude speed-up for GAN models with nVidia Titan Xp GPU

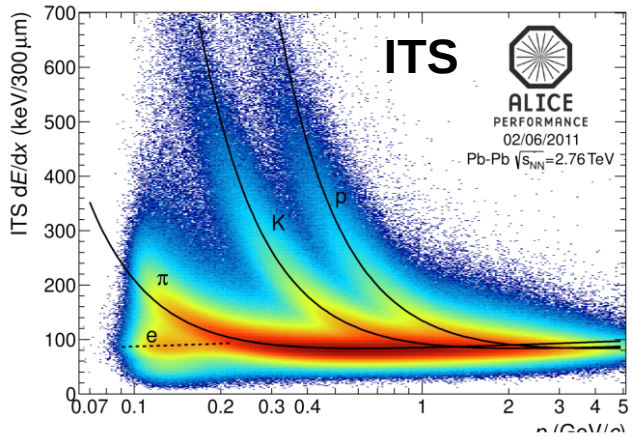


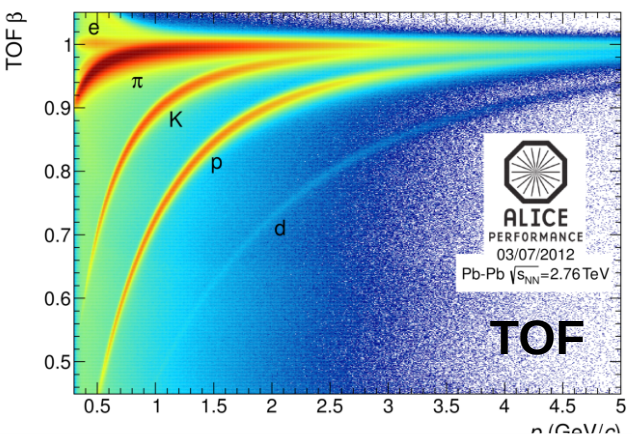
PID with Machine Learning

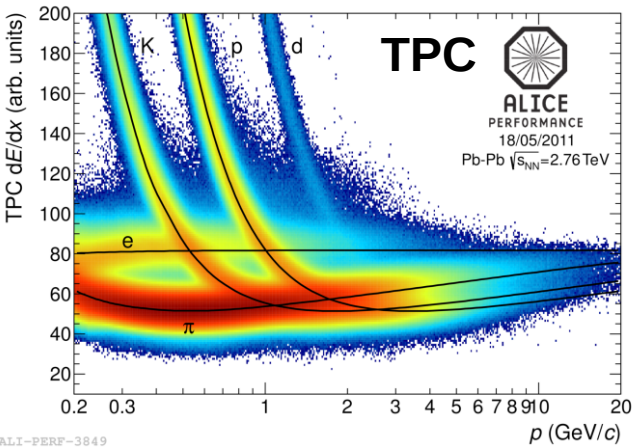
Particle identification

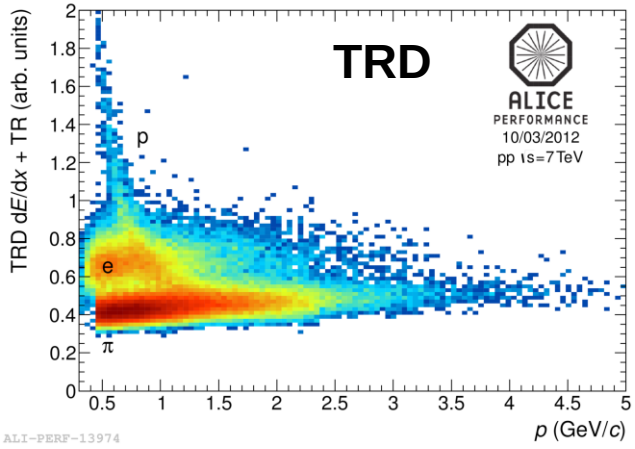
- Particle identification (PID) is one of the most important steps in many physics analyses
- Crucial for Quark-Gluon Plasma measurements
- PID is one of the strongest advantages of ALICE:
  - practically all known techniques used (dE/dx energy loss, time-of-flight, Cherenkov radiation for hadrons and transition radiation for electrons)
  - possibility to identify (anti-)nuclei
  - very good separation of pions, kaons, protons, electrons over a wide momentum range
  - separation of signals of charged hadrons and electrons for very low momenta (down to 0.1 GeV/c)

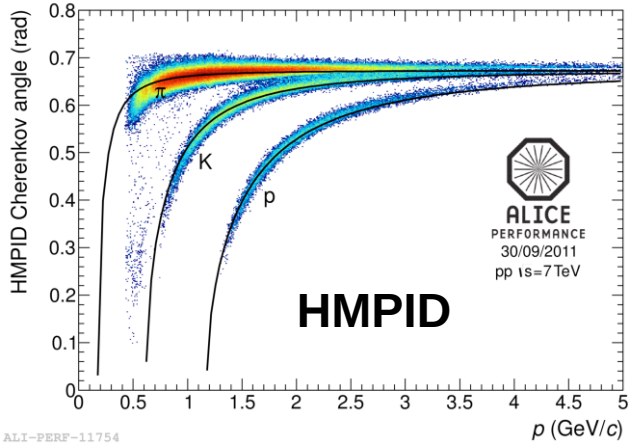
Particle identification

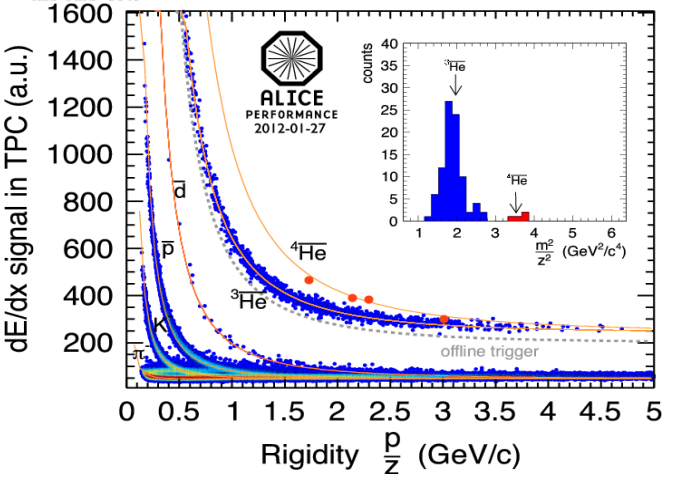


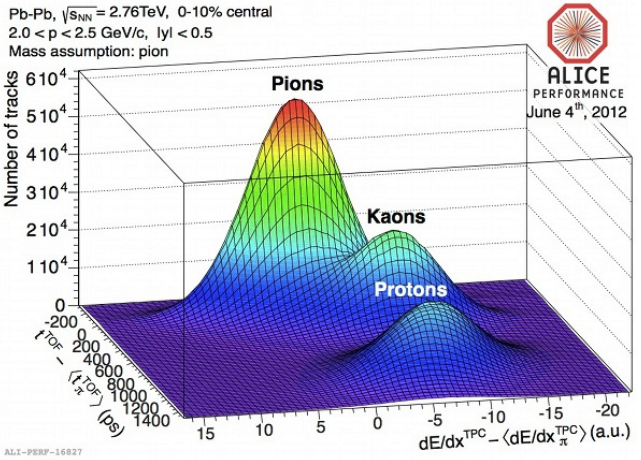






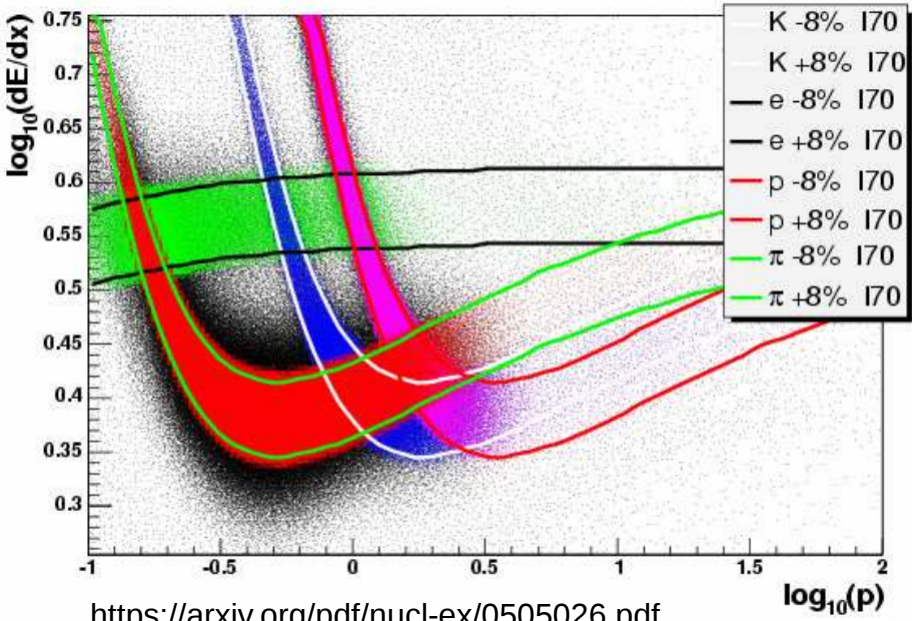






Traditional vs ML PID

- **Traditional PID:**
  - a typical analyzer selects particles "manually" by cutting on certain quantities, like the number of standard deviations of a signal from the expected value
  - most limitations come in the regions where signals from different particle species cross
  - "cut" optimization is a time-consuming task

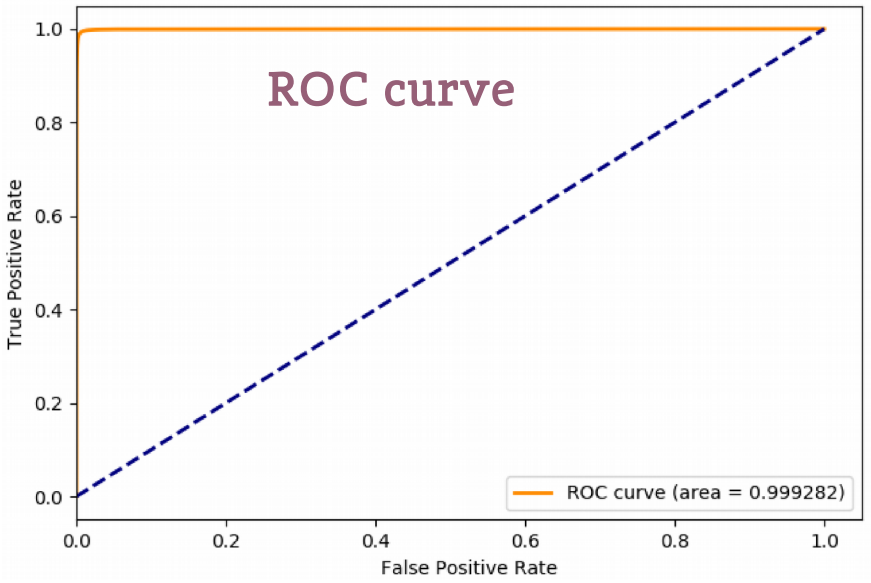


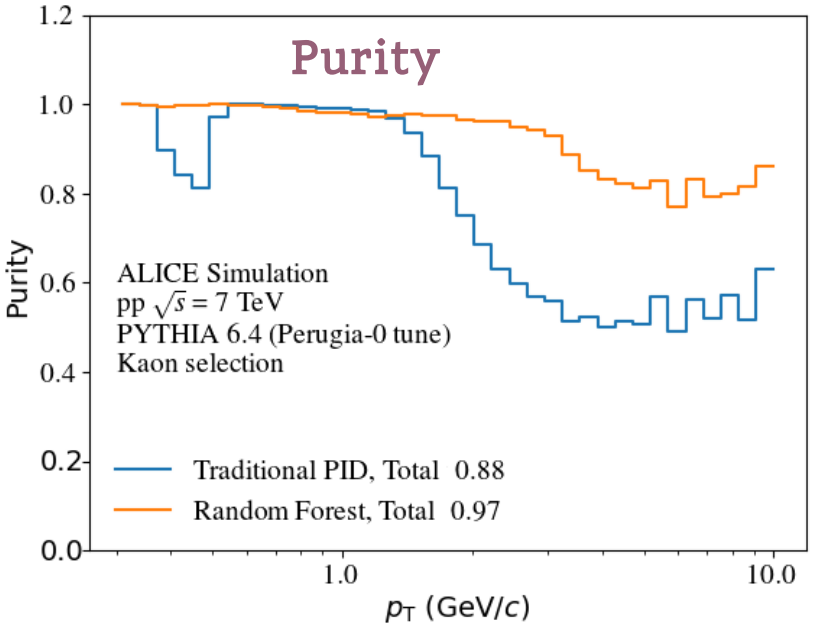
https://arxiv.org/pdf/nucl-ex/0505026.pdf

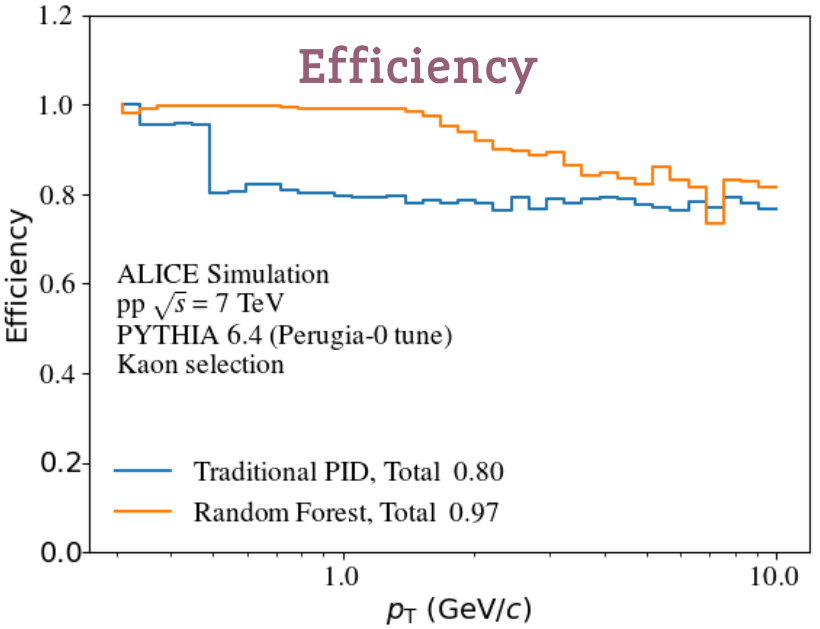
- **Machine learning PID:**
  - perfect task for machine learning
  - can learn non-trivial relations between different track parameters and PID
  - no "trial and error" approach

Results

- **Test data sample:**
  - pp @ 7 TeV, Pythia 6 Perugia-0
- **Traditional PID:**
  - $n^2_{\sigma,TPC} < 2$, for $p \leq 0.5$ GeV/c
  - $\sqrt{n^2_{\sigma,TPC} + n^2_{\sigma,TOF}} < 2$, for $p > 0.5$ GeV/c
- **Machine Learning PID:**
  - Random Forest classifier

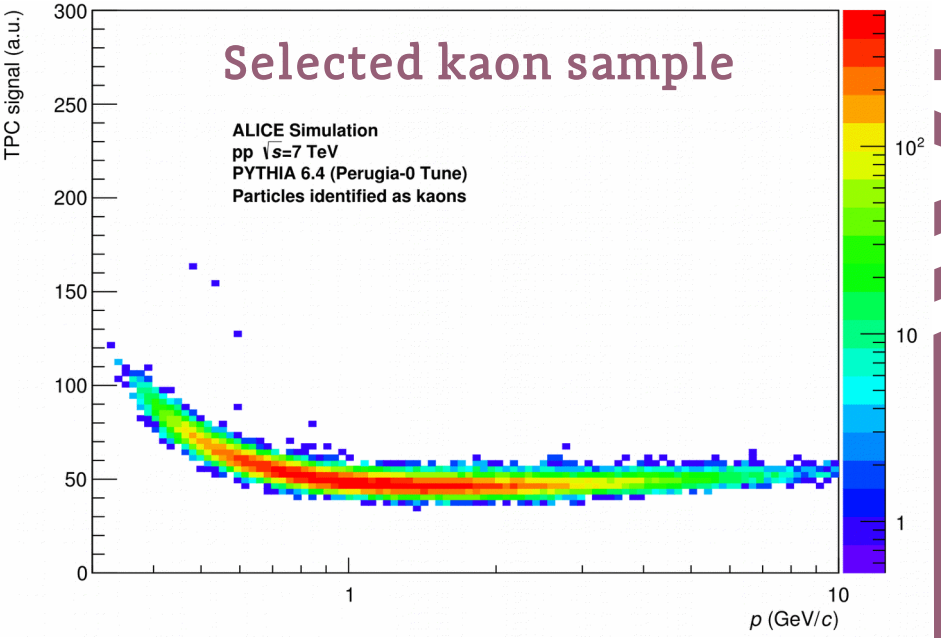


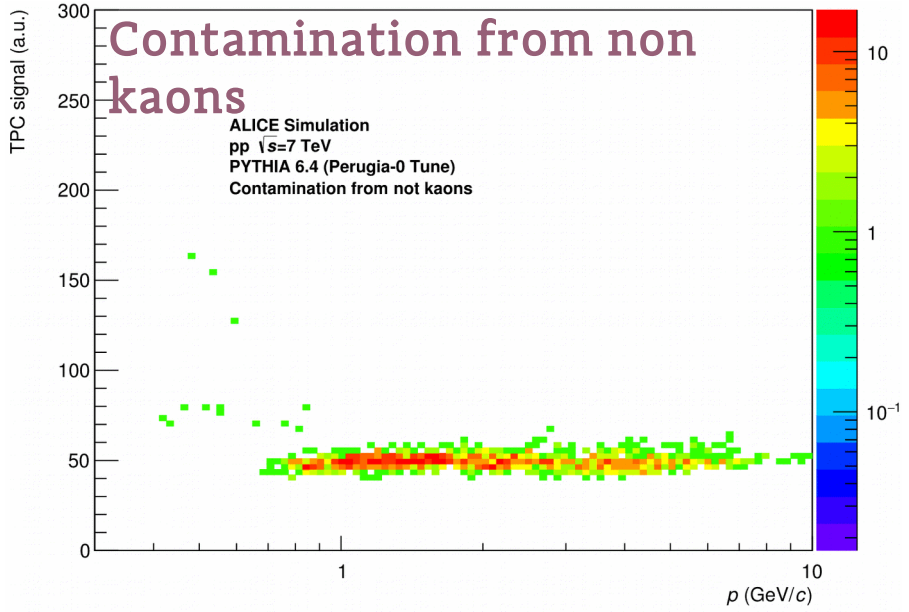


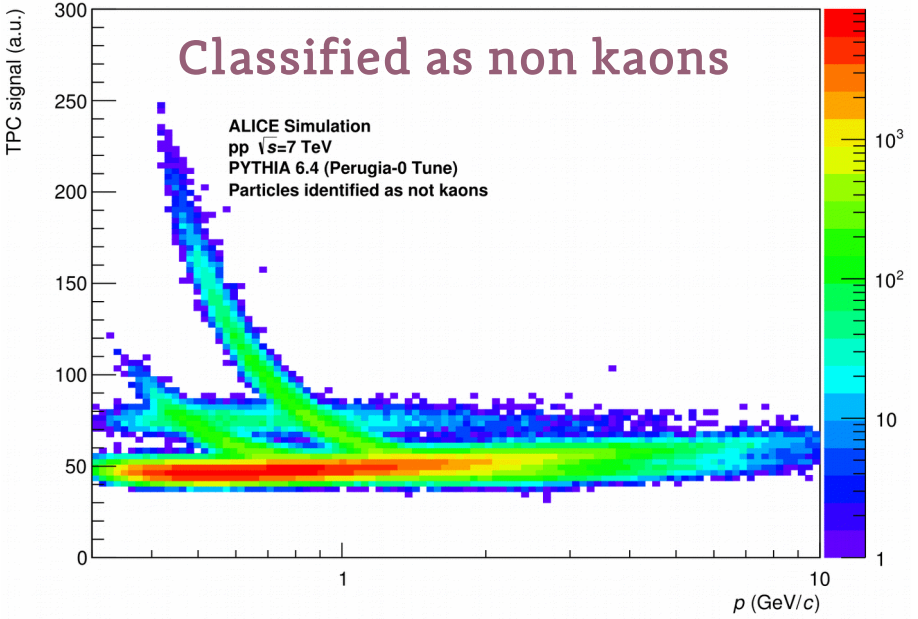


Results

- **Test data sample:**
  - pp @ 7 TeV, Pythia 6 Perugia-0
- **Traditional PID:**
  - $n^2_{\sigma,TPC} < 2$, for $p \leq 0.5$ GeV/c
  - $\sqrt{n^2_{\sigma,TPC} + n^2_{\sigma,TOF}} < 2$, for $p > 0.5$ GeV/c
- **Machine Learning PID:**
  - Random Forest classifier







Summary

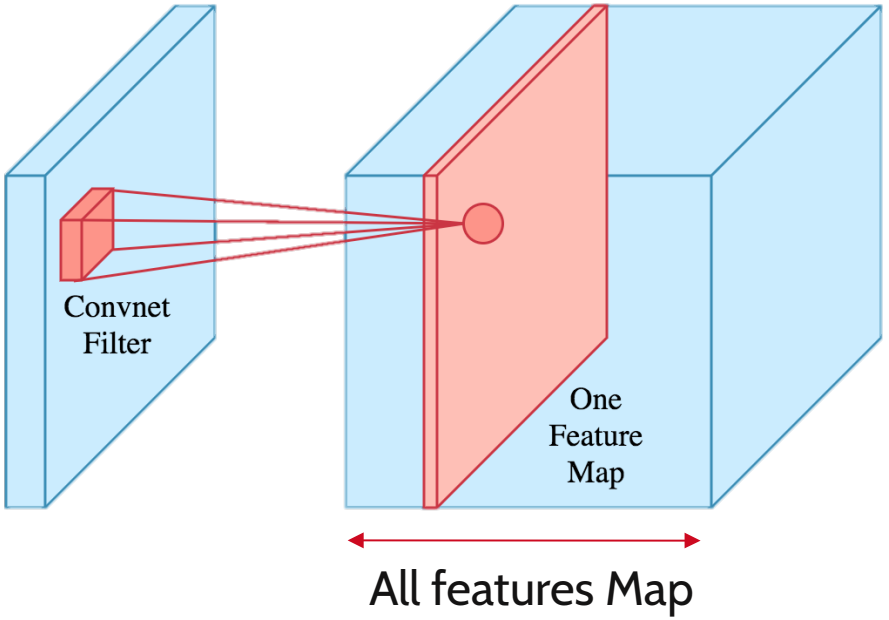
- GANs for TPC cluster simulation
  - Quality not yet equal to the full simulation
  - Massive speed-up of 25x (CPU) or 250x (GPU) wrt standard simulation
  - First step towards semi-real time anomaly detection tool
- PID
  - ML-based PID outperforms traditional PID, especially in the low momentum region
  - training needed only once for each data set – no need for manual cut optimizations
  - Quality of final classification more vulnerable to discrepancies between MC and real data

Thank you!

Backup

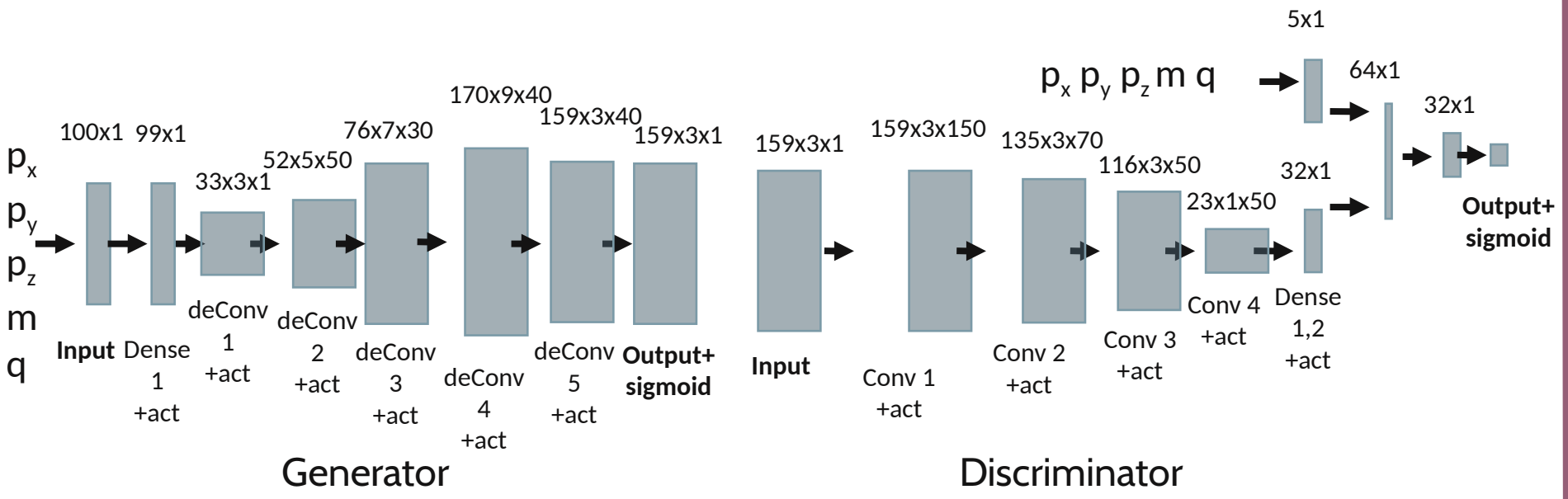
Deep Convolutional GAN

- Class of architectures which use the convolutional tools and deconvolutional layers – mostly used with images



condDCGAN: Conditional DCGAN

- Generator – deconvolutional layers
- Discriminator – convolutional layers
- Network conditioned on particle momenta, mass, and charge
- Output classification – sigmoid function

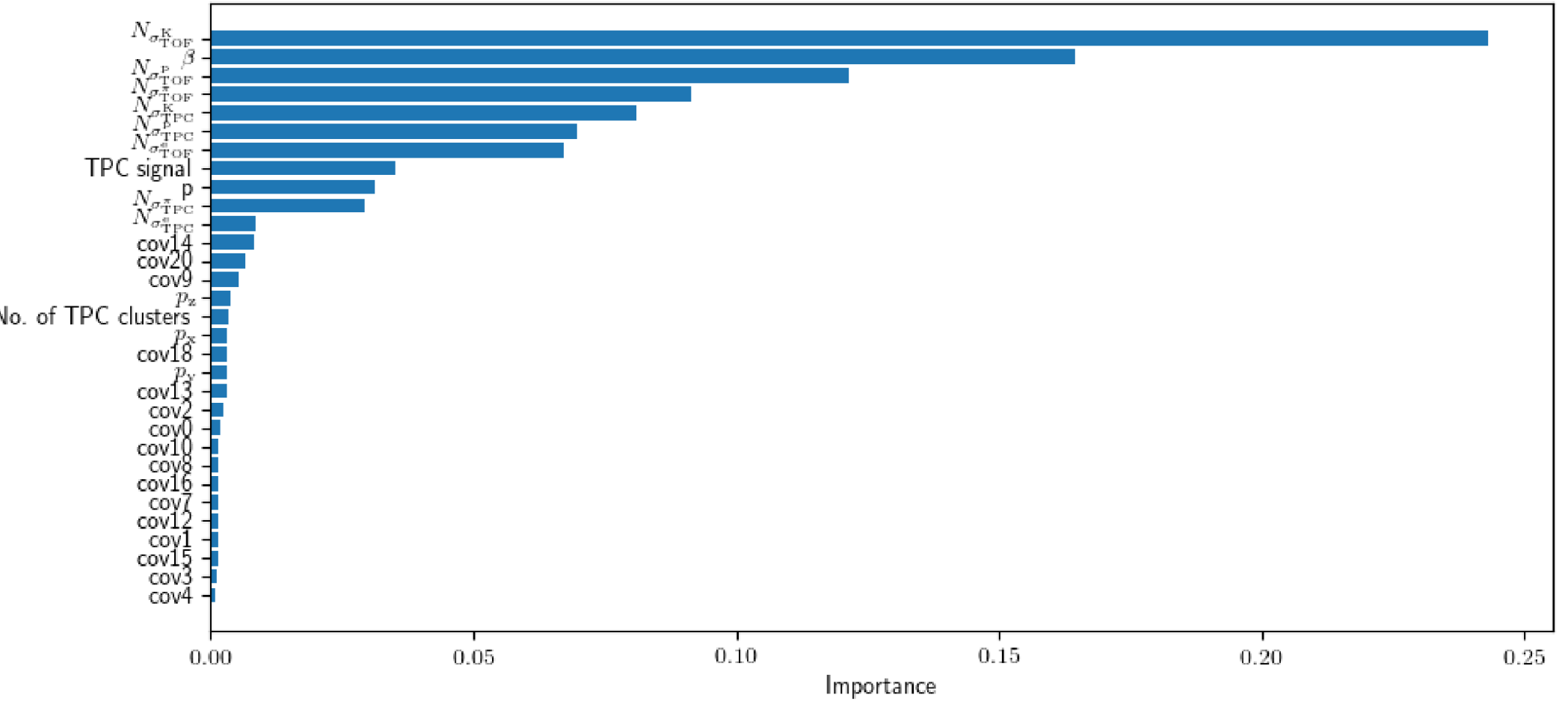


condDCGAN+: combined loss

- Training on on the full MC simulations
- Preparing the noise from initial parameters of MC simulations
- Comparing the generated samples with original ones
- Combining origininal conditional GAN loss with the results of comparison

$$\mathcal{L}_G(m, X) = \mathbb{E}_{z \sim p_z(z|m)}[\alpha \log(1 - D(G(z))) + \beta \frac{1}{n} \sum_{i=1}^{n} (X_i - G(\hat{z})_i)^2]$$

m - initial parameters (particle momenta),
X - original value corresponding to m ,
p(z|m) - distribution of a noise vector under initial parameters m
z - input into a generator
G and D - generator and discriminator
n - the number of produced clusters

Additional parameters α and β are used to weight the share of individual losses.
Best performing values are α = 0.6 and β = 0.8

PID parameter importance