Warsaw University of Technology

ALICE

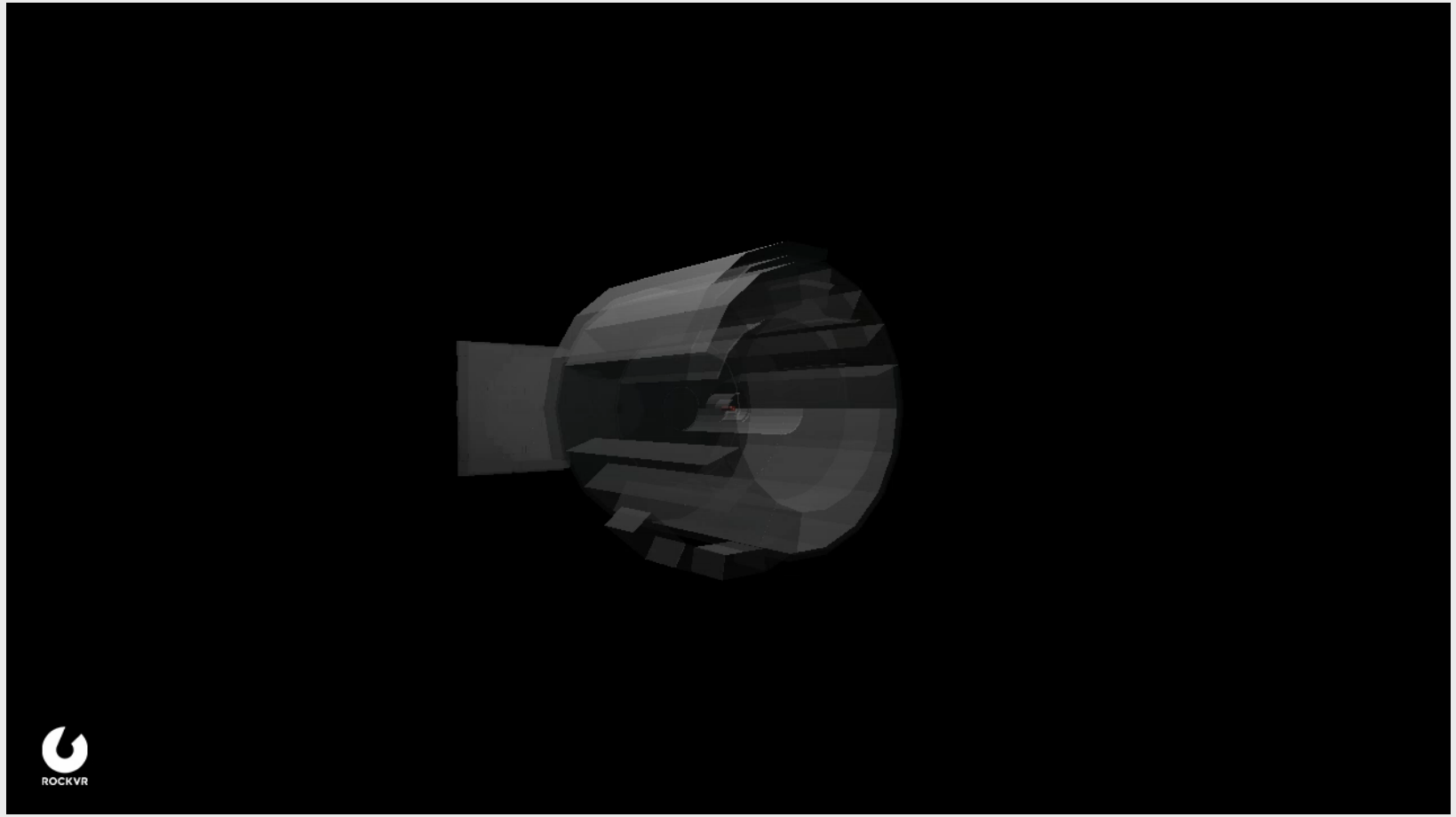
# Using Generative Adversarial Networks for Fast Simulation in ALICE Experiment

**Tomasz Trzciński**, Kamil Deja, Łukasz Graczykowski
for the ALICE Collaboration

# Agenda

1. ALICE Experiment
2. Particle clusters simulation - problem description
3. Generative Models
   - Variational Autoencoder (VAE)
   - Generative Adversarial Networks (GAN)
   - Conditional Generative Adversarial Networks
4. Clusters simulation with Generative Models
5. Results

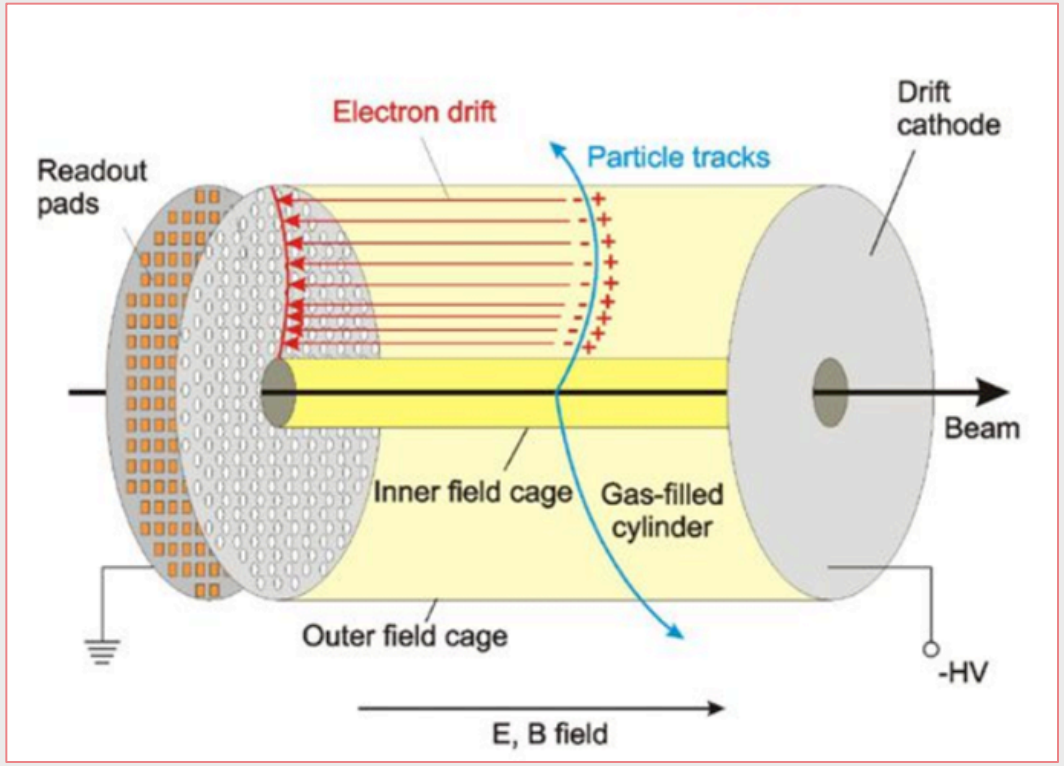
# ALICE Experiment




J.Ramatowski, Visualization of data from ALICE experiment in virtual reality

# Particle clusters

- Points in **3-dimensional space**, together with the energy, which were presumably generated by a particle crossing by.
- Input for particle tracks generation
- Up to **159 points per particle**
- Possible values **restricted** by the detector size ~ 5m x 5m x 5m
- **No clusters** in the inner field cage




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

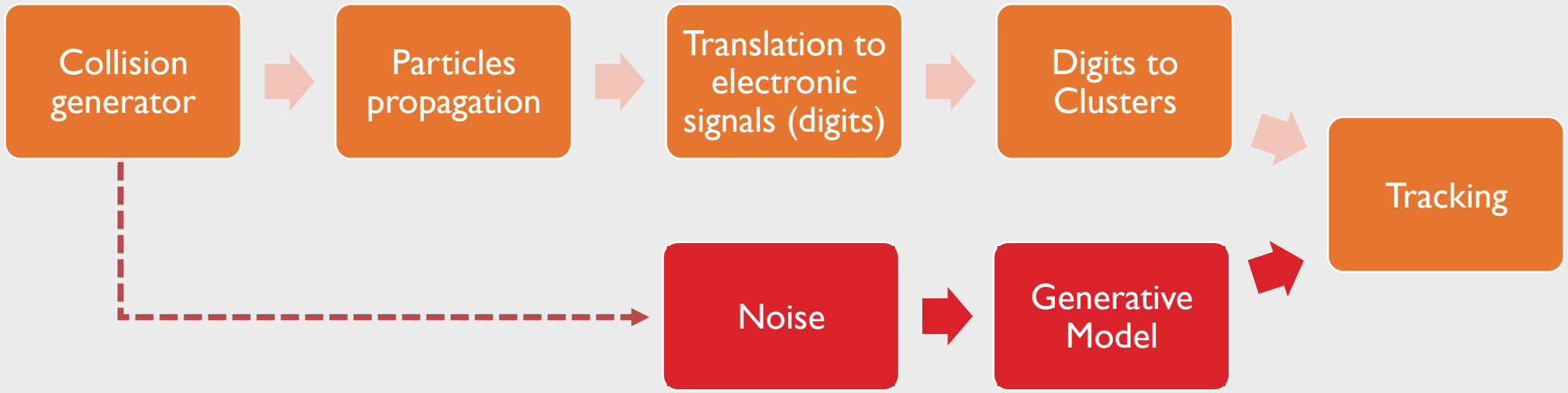
# Simulation and reconstruction

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

Collision generator → Particles propagation → Translation to electronic signals (digits) → Digits to Clusters → Tracking

# Simulation and reconstruction

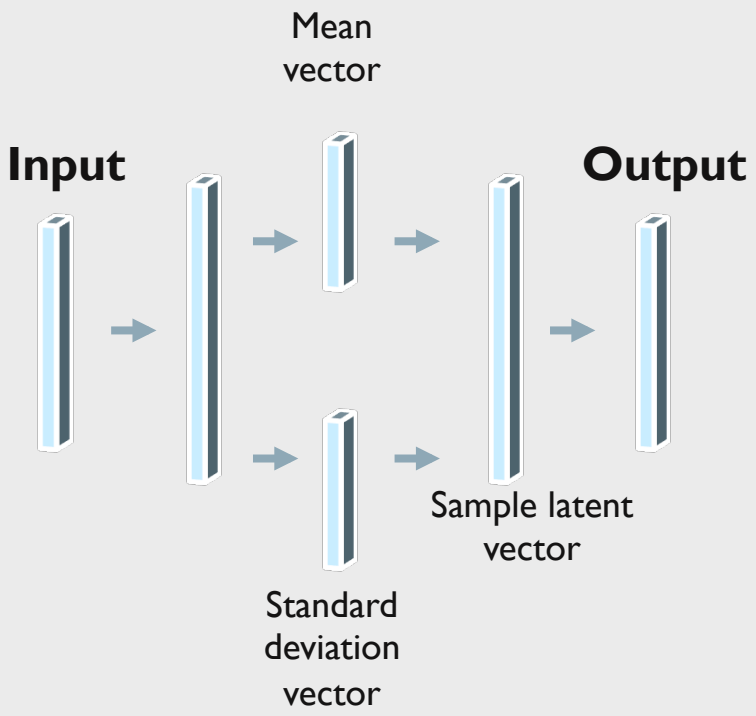
## Generative solution for clusters simulation



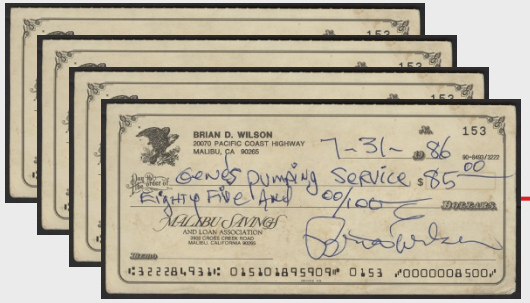
# Generative Models

# Variational Autoencoder

- Extension of the Autoencoder which re-generates the input at the system output
- **Normalisation** on the first hidden layer which forces it's output to have a normal distribution
- Generation by providing significant noise on the **Latent Space**

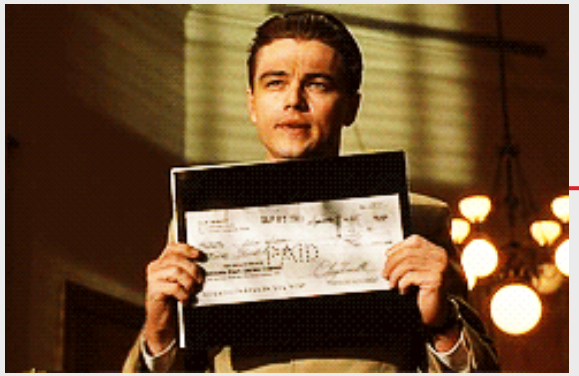


# Generative Adversarial Networks

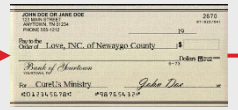



https://33milesinnewaygocounty.files.wordpress.com

Generator



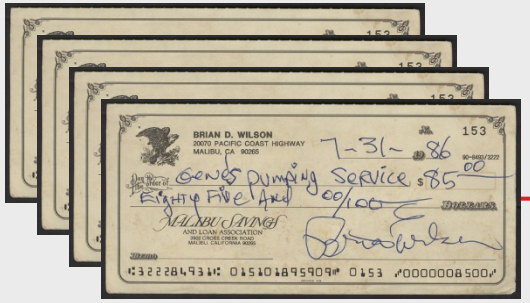
https://giphy.com/gifs/leonardo-dicaprio-catch-me-if-you-can-5leocharacters-t1h4nnWEWKfn2



Discriminator

https://thechive.files.wordpress.com

# Conditional Generative Adversarial Networks



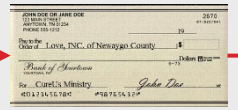

https://33milesinnewaygocounty.files.wordpress.com

Discriminator

https://thechive.files.wordpress.com

Generator

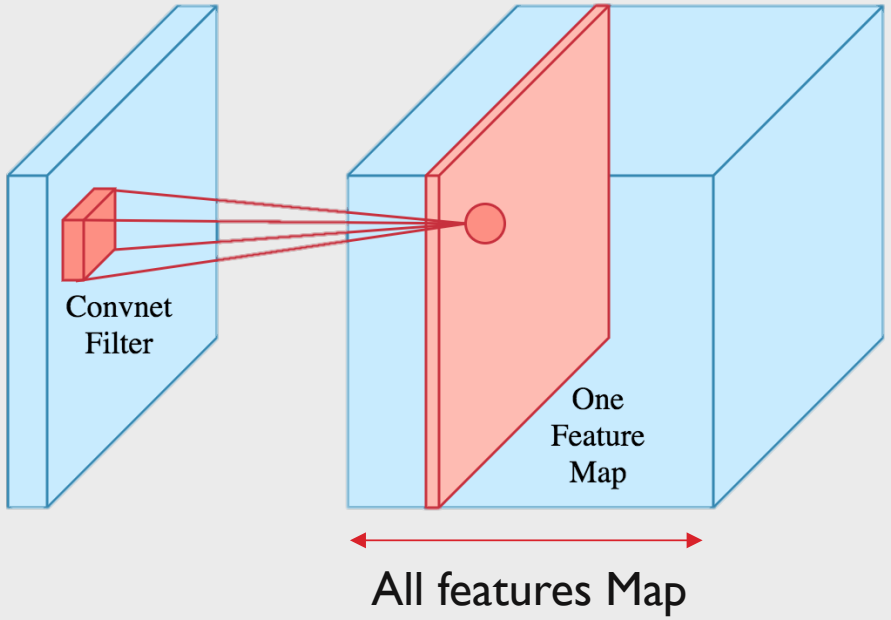
https://giphy.com/gifs/leonardo-dicaprio-catch-me-if-you-can-5leocharacters-t1h4nnWEWKfn2




Initial Parameters

# Deep convolutional GAN

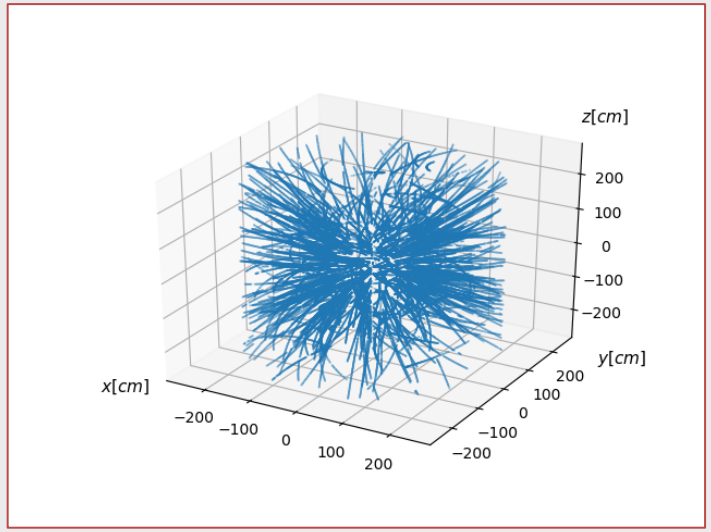
- Class of architectures which use the convolutional and deconvolutional layers – mostly used with images

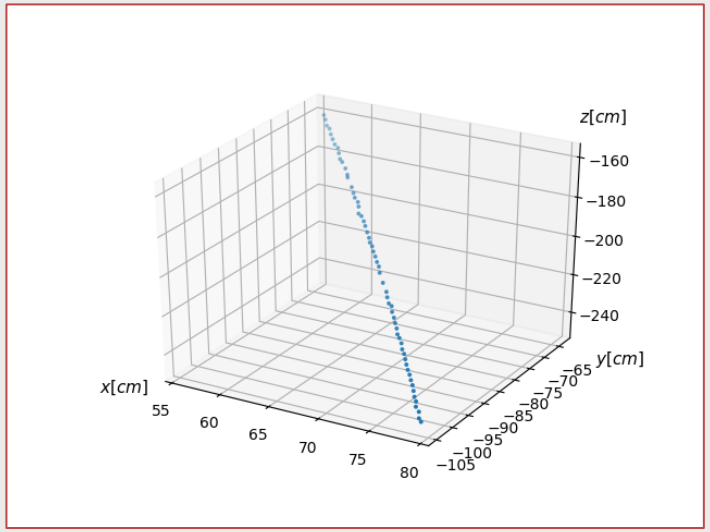


# Clusters Simulation with Generative Models

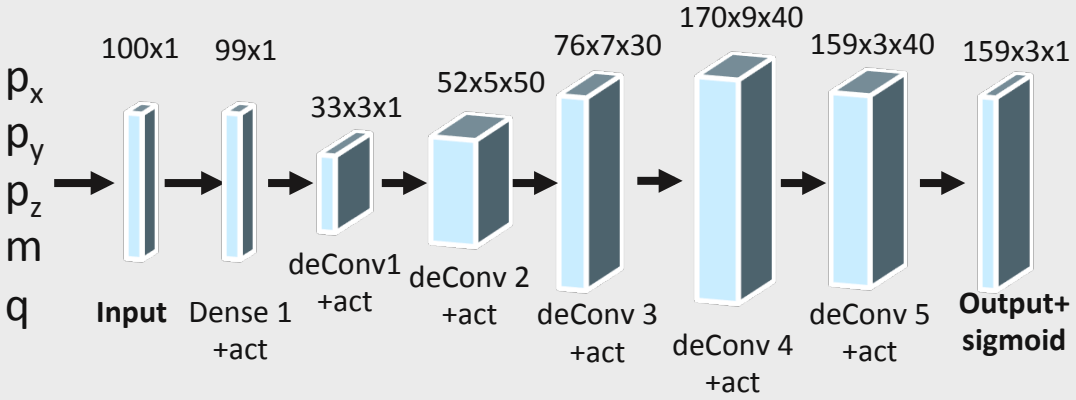
# Dataset

- It is not possible (yet) to generate the full 3D image of the event at once
- (5000 x 5000 x 5000 resolution)
- Our solution is to:
  - Generate clusters for single particle (as 2D table with x, y ,z ,q, $q_{max}$ values)
  - Two separate flows for x, y ,z and q, $q_{max}$
  - Merge generated samples
- Training on the original reconstructions

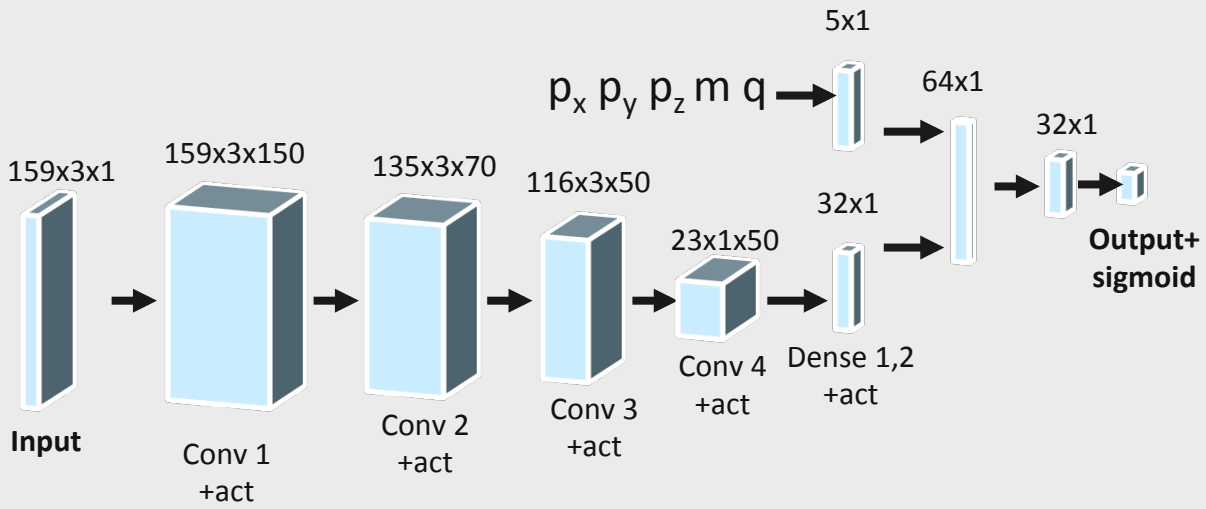


# condDCGAN: Conditional DCGAN




Generator




Discriminator

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

- Dropout
- Batch Normalization
- Sigmoid activation on output

# condGAN+: combined loss

- Training only on the real examples from dataset
- Preparing the noise from initial parameters of real examples
- Comparing the generated samples with original ones
- Combining original 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

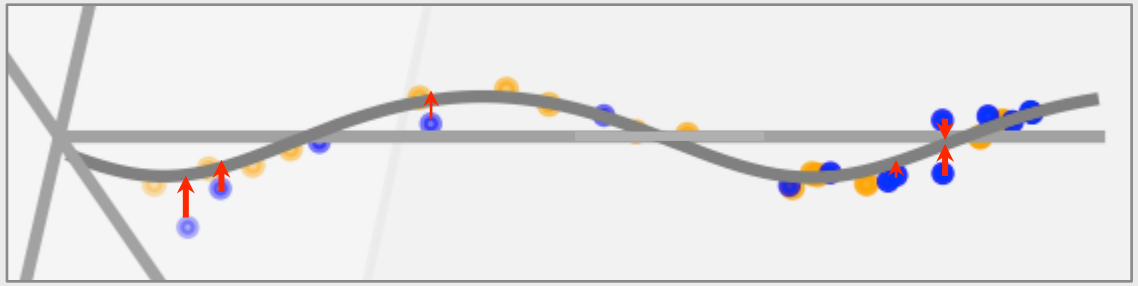
Additional parameters $\alpha$ and $\beta$ are used to weight the share of individual losses.
Best performing values are $\alpha$ = 0.6 and $\beta$ = 0.8

# Results

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

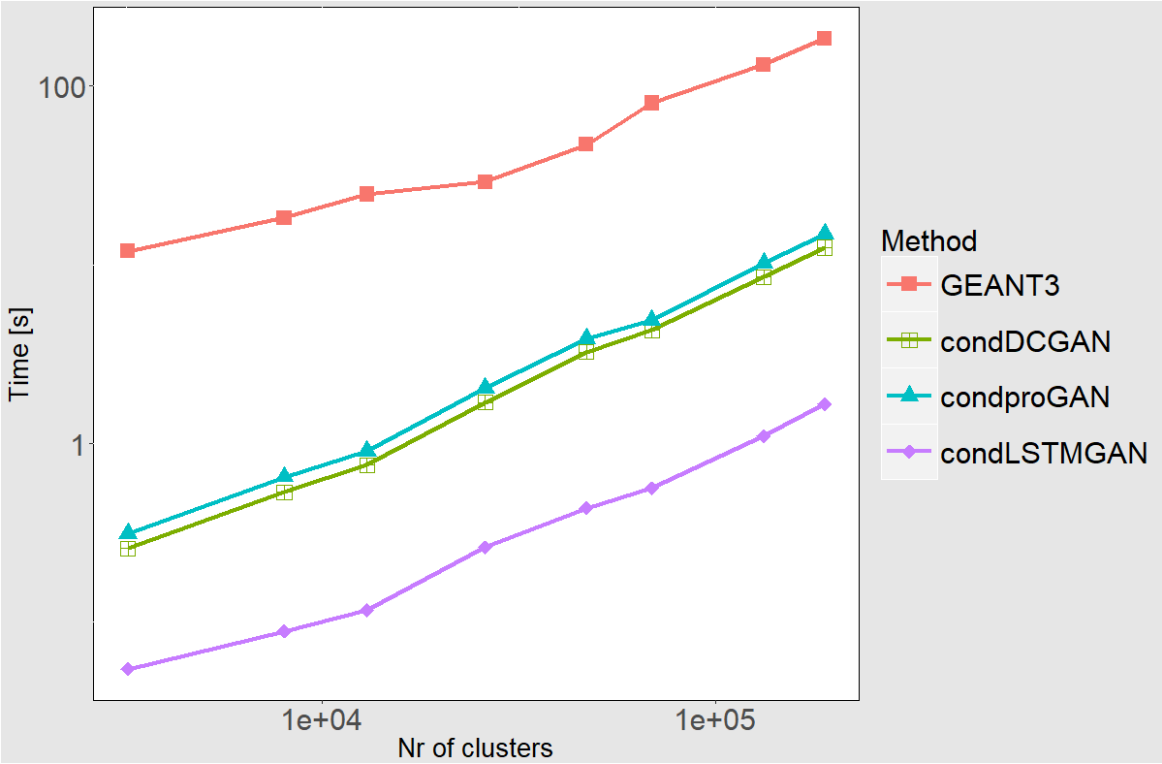
| Method | Mean MSE (mm) | Median MSE (mm) | Speed-up |
|---|---|---|---|
| GEANT3 | 1.20 | 1.12 | 1 |
| Random (estimated) | 2500 | 2500 | N/A |
| condLSTM GAN | 2093.69 | 2070.32 | 100 |
| condLSTM GAN+ | 221.78 | 190.17 | |
| condDCGAN | 795.08 | 738.71 | **25** |
| **condDCGAN+** | **136.84** | **82.72**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

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



# Computational costs

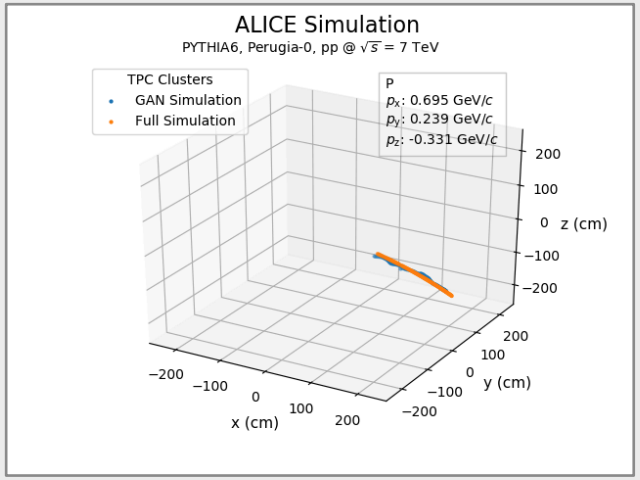
- Performance test conducted on the standalone machine with Intel Core i7-6850K (3.60GHz) CPU (using single core, no GPU acceleration)
- Additional order of magnitude speedup for Generative models with Nvidia Titan Xp

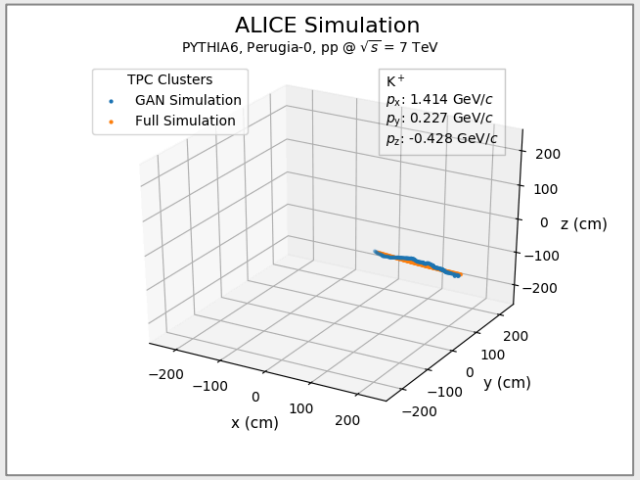


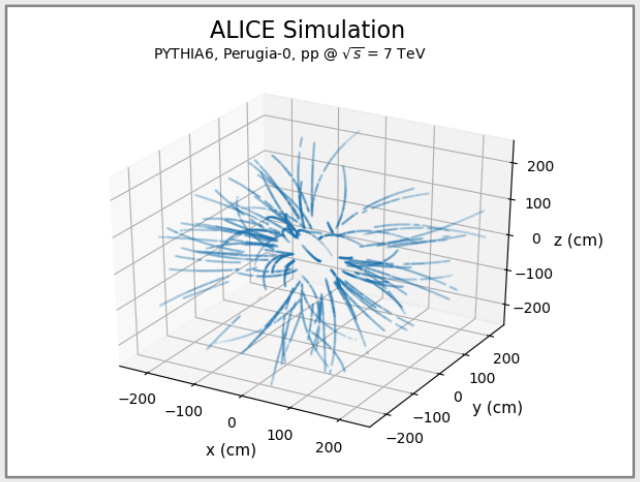

Conditional clusters simulation
(log-log scale)

# Examples

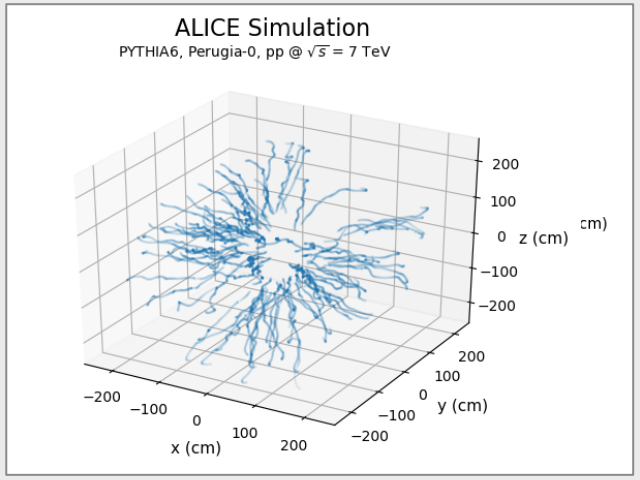
for the conditional clusters simulation:




Original event




Generated event

# Summary

- Quality not yet equal to this observed with full simulation
- Massive speed-up 25 (CPU) or 250 (GPU) comparing to standard simulation methods
- First step toward semi-real time anomaly detection tool

# Acknowledgments


- The authors acknowledge the support from the Polish National Science Centre grant no. UMO-2016/21/D/ST6/01946.
- The GPUs used in this work were funded by the grant of the Dean of the Faculty of Electronics and Information Technology at Warsaw University of Technology (project II/2017/GD/1).

National Science Centre Poland

Faculty of Electronics and Information Technology
Warsaw Universtiy of Technology