Warsaw University of Technology

Generative Models for Fast Cluster Simulations in the TPC for the ALICE Experiment

Kamil Deja and Tomasz Trzciński for the ALICE Collaboration

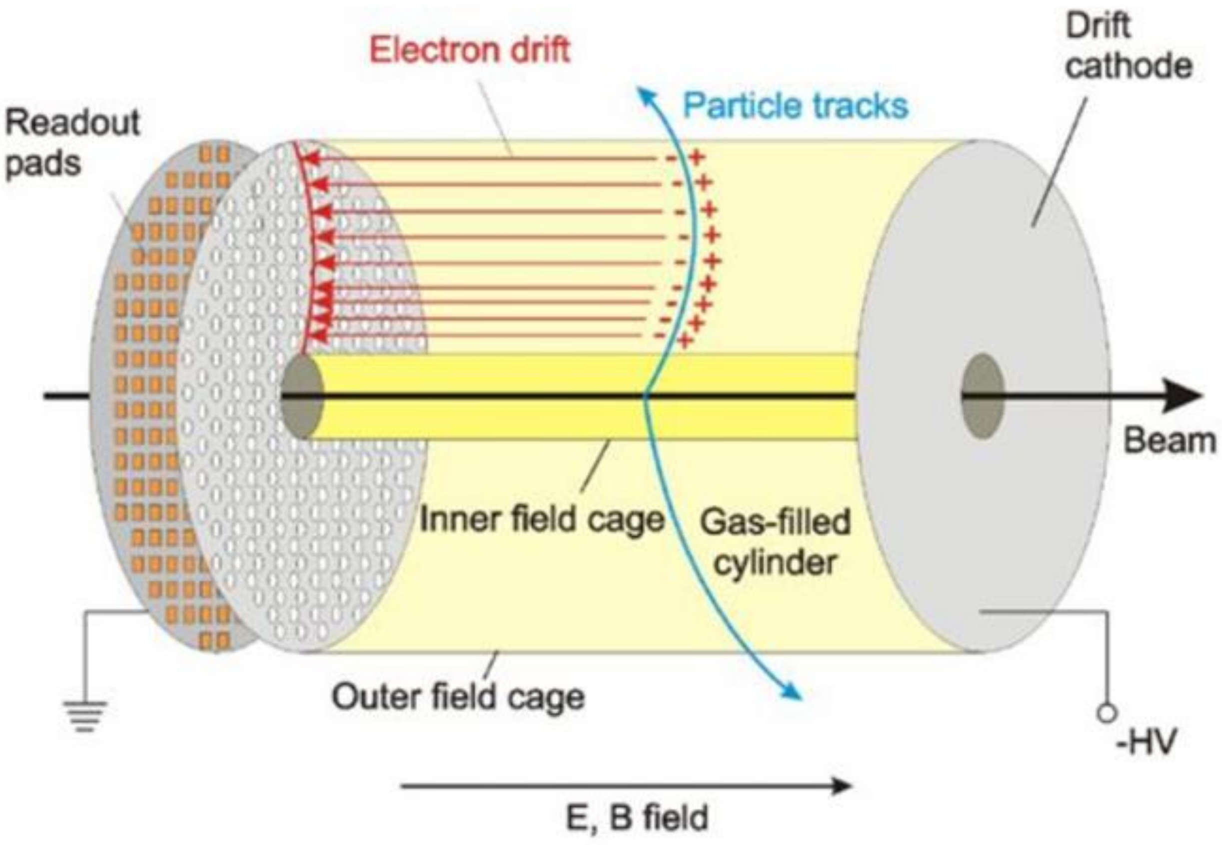
2nd IML Machine Learning Workshop 10/04/2018

Outline

1. Particle clusters simulation - problem description
2. Generative Models
    a. Variational Autoencoder (VAE)
    b. Generative Adversarial Networks (GAN)
3. Clusters simulation with Generative Models
    a. VAE
    b. DCGAN
    c. Progressive GAN
4. Results
5. Future work

Particle clusters

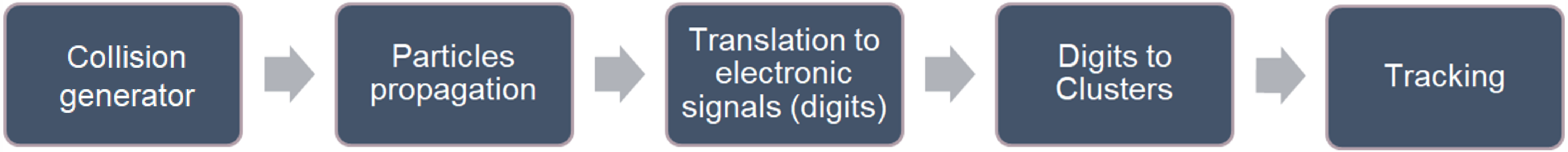
- Points in 3 dimensional space, together with the energy, which were presumably generated by a particle crossing by.
- Base for particle tracks generation
- Up to 159 points per particle
- Possible values restricted by the detector size ~ 5m x 5m x 5m
- No clusters in the inner field cage



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

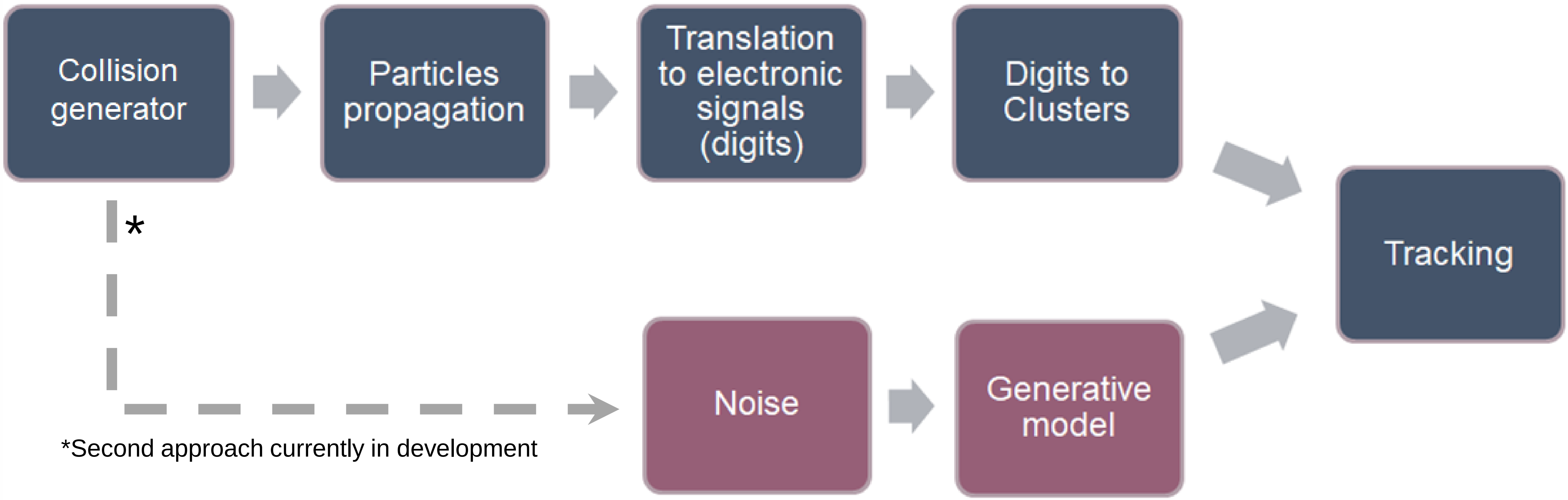
Simulation and Reconstruction

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



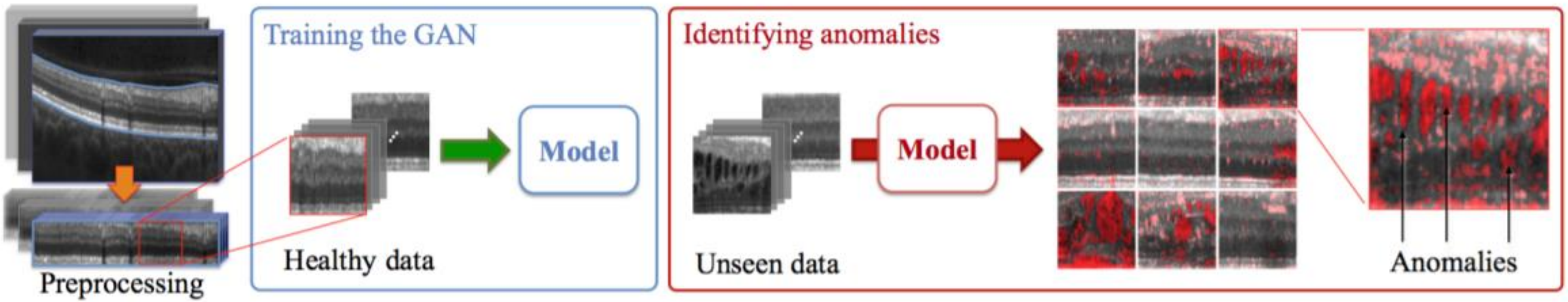
Simulation and Reconstruction

Generative solution for clusters simulation



Motivation

- Fast particle clusters simulation
- Semi-real time anomaly detection tool for Quality Assurance
- Generating possible clusters distribution to compare them with the real detector's output



T. Schlegl Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery

Warsaw University of Technology

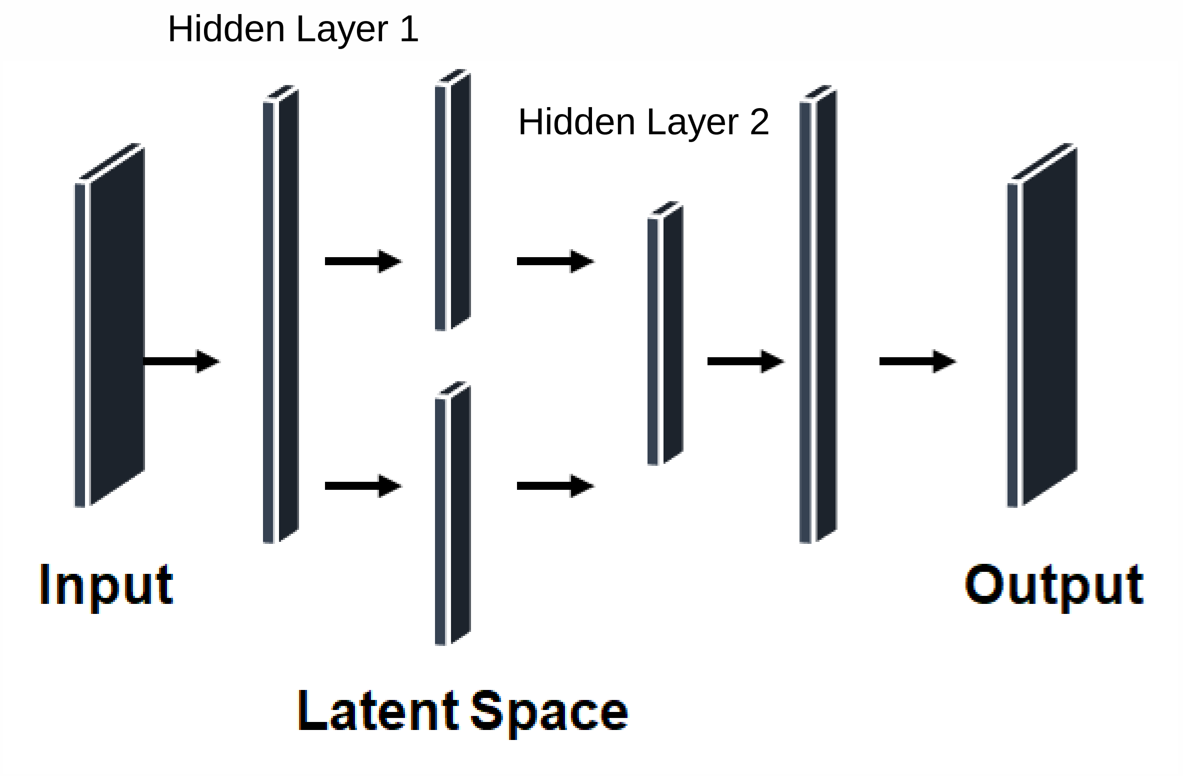
ALICE

CERN

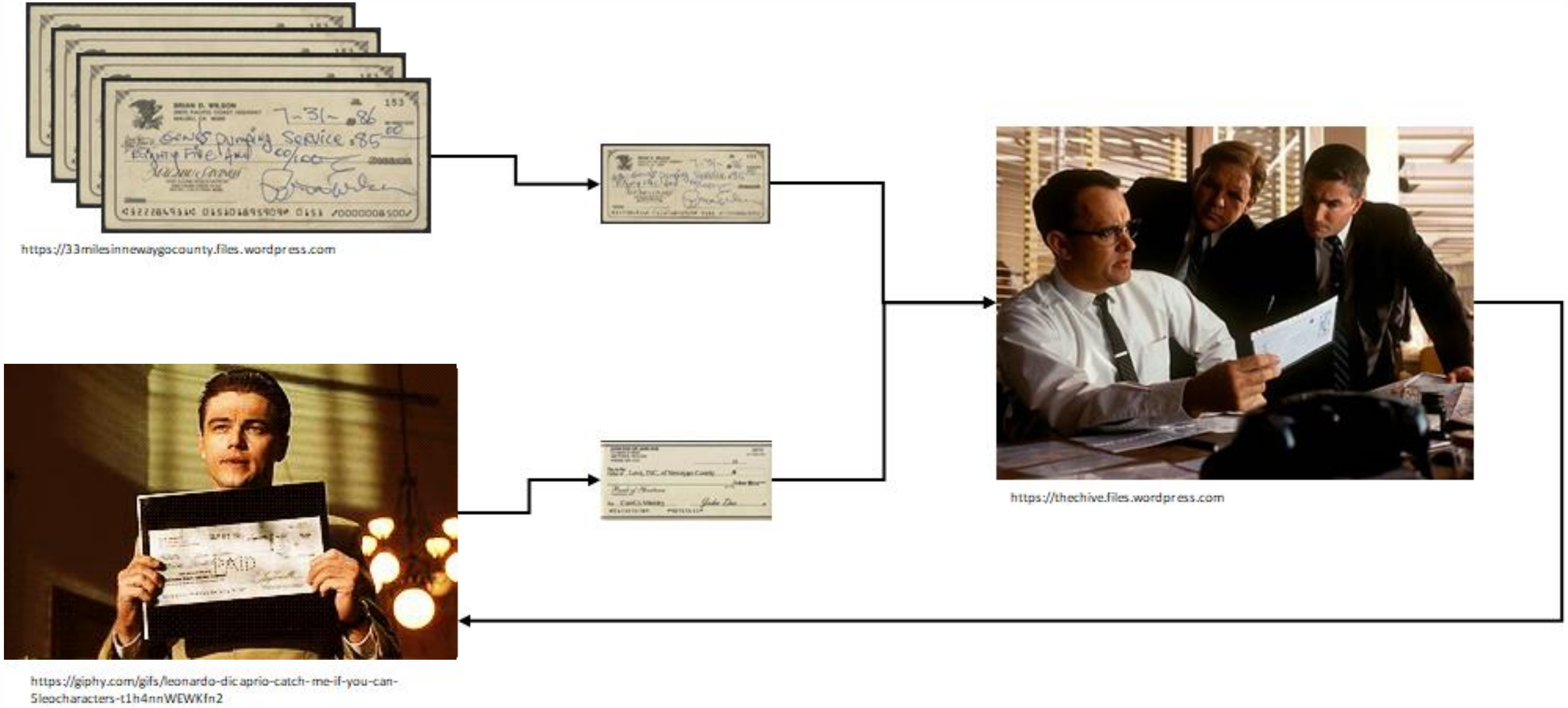
Generative Models

Variational Autoencoder

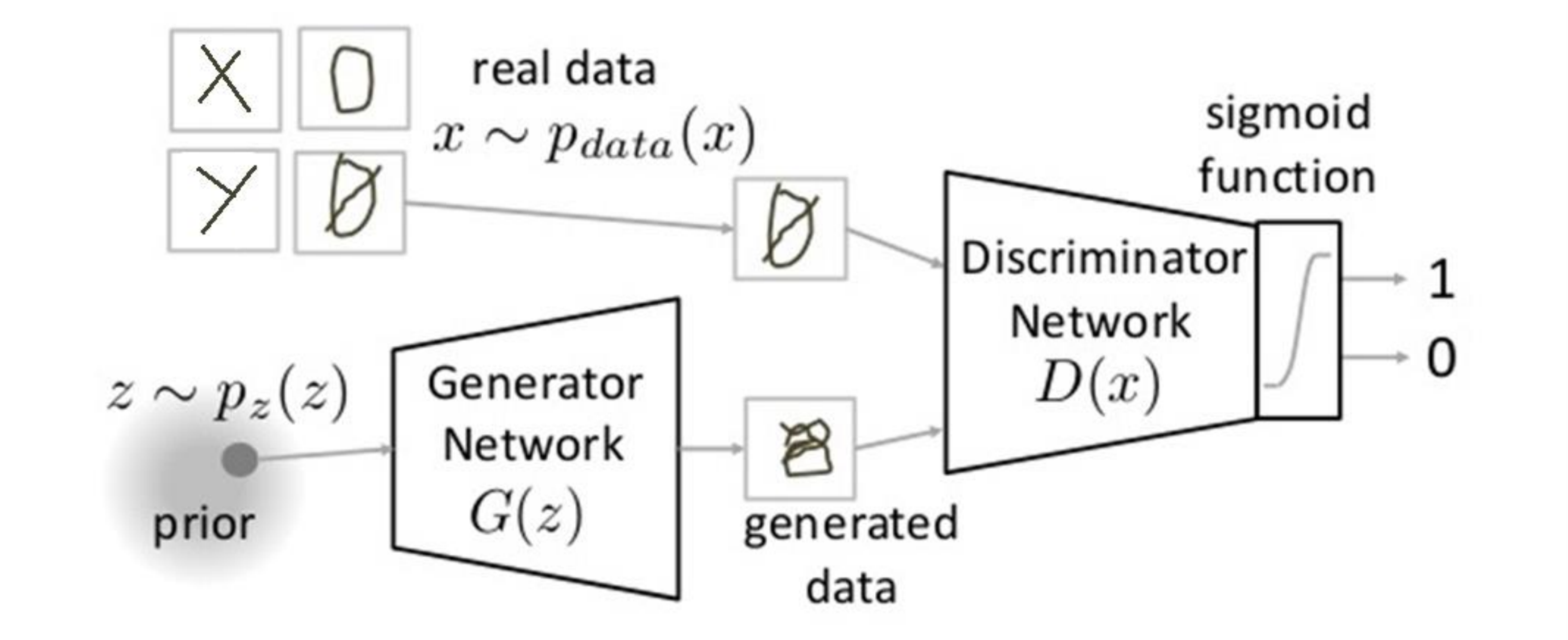
- Deriving from Autoencoder - re-generates same Output as Input
- Normalisation on the first hidden layer which forces it’s output to have a normal distribution
- Generation by providing significant noise on the Latent Space



Generative Adversarial Networks - introduction



Generative Adversarial Networks



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

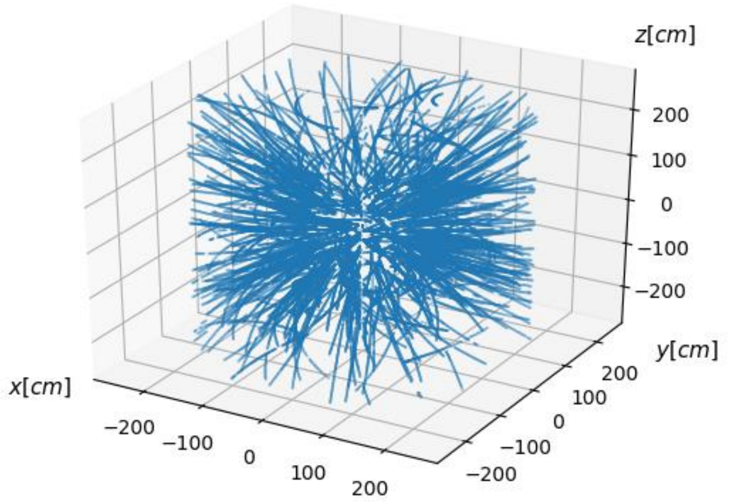
https://www.analyticsvidhya.com/blog/2017/06/introductory-generative-adversarial-networks-gans/

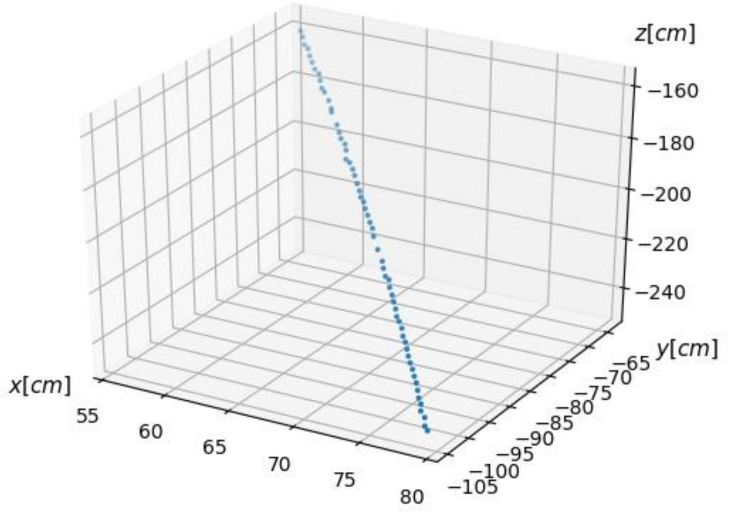
Warsaw University of Technology

Clusters simulation with Generative Models

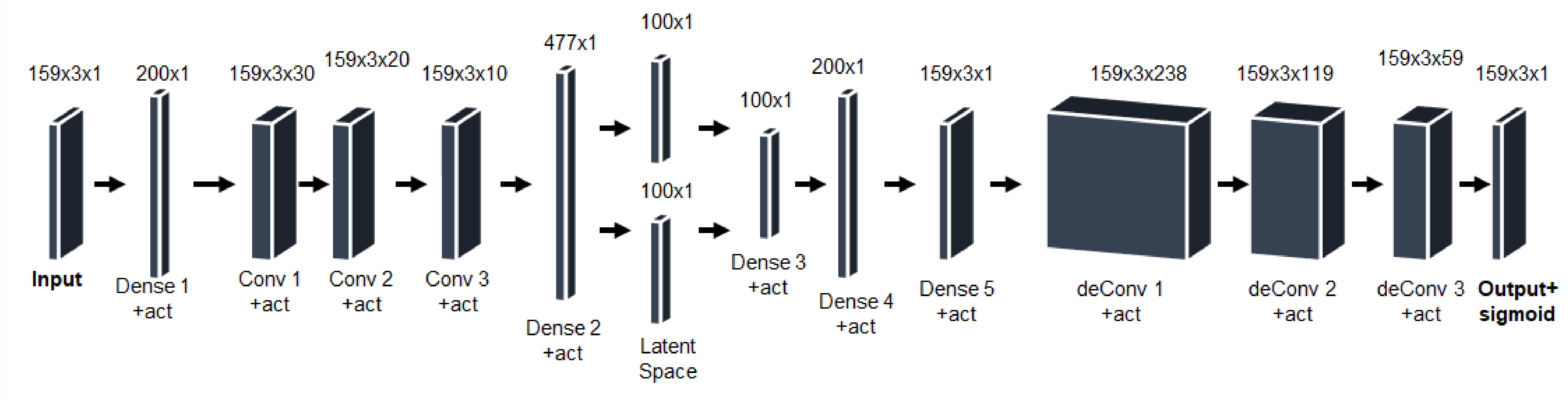
Real dataset

- 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





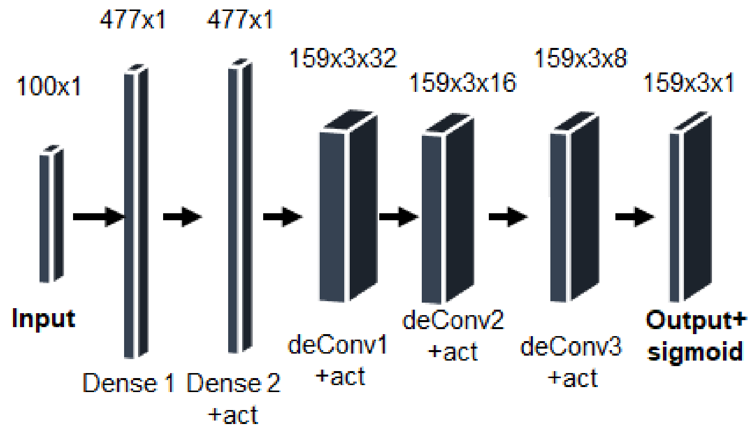
Convolutional Variational Autoencoder



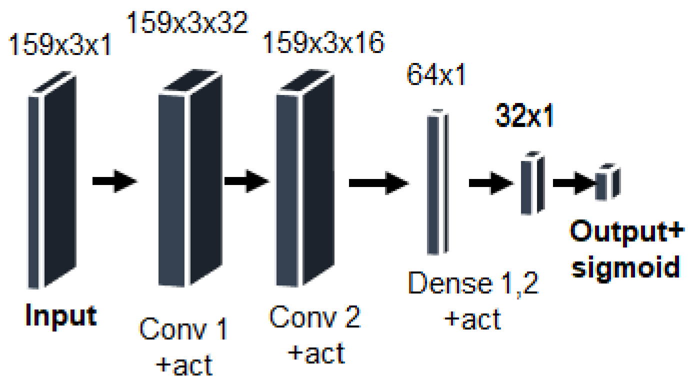
- Deep Convolutional Variational Autoencoder
- 2D Convolutional/ Deconvolutional Layers
- Leaky ReLU Activation
- Dropout
- Batch Normalisation
- Sigmoid activation on output
- VAE’s loss function

Deep Convolutional Generative Adversarial Network (DCGAN)

- 2D Convolutional/ Deconvolutional Layers
- Dense Layers for input, and output
- Leaky ReLU Activation
- Dropout
- Sigmoid activation on output



Generator

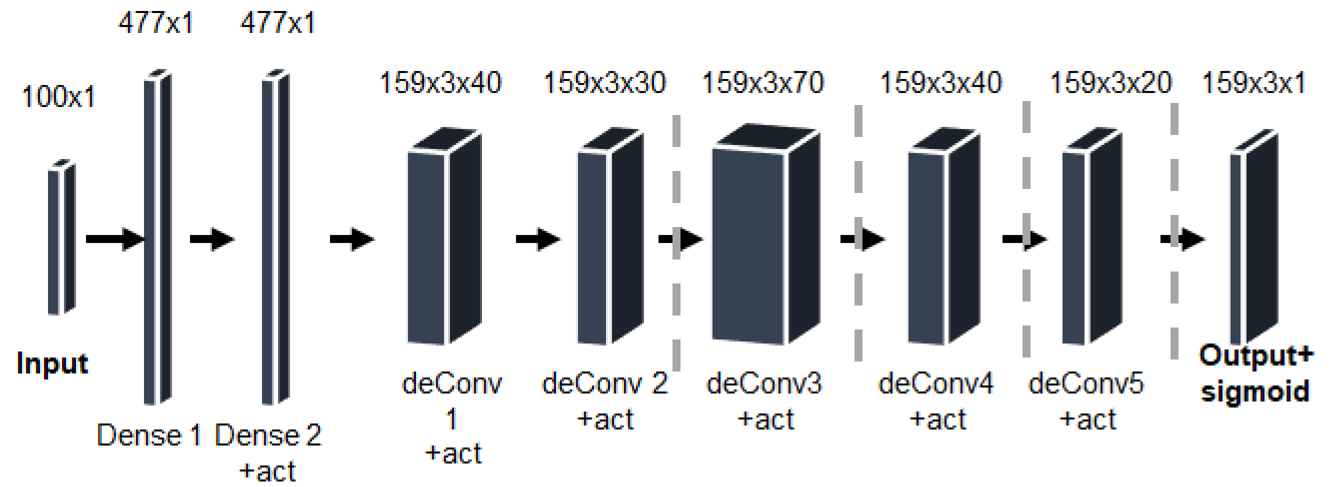


Discriminator

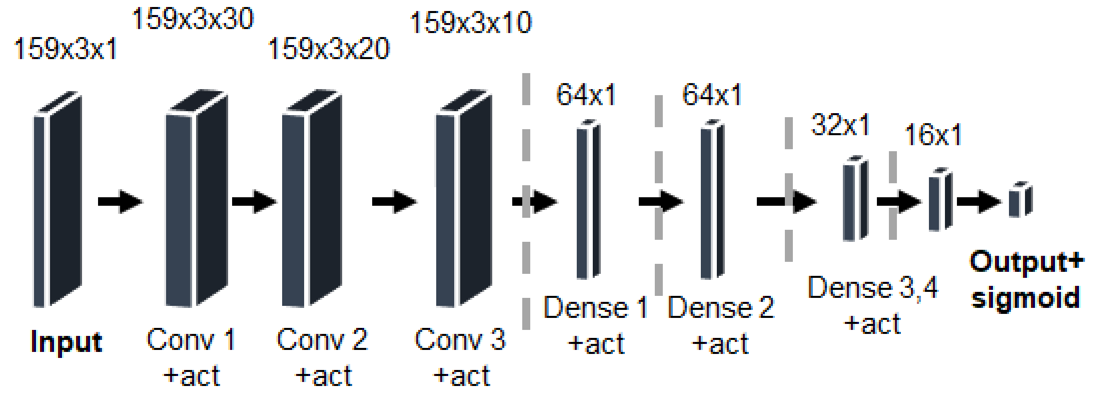
Progressive DCGAN

Progressive training for standard DCGAN

- Gradually increased number of layers
- Training on data samples with steadily growing precision
- Constant enhancement of generated samples resolution



Generator



Discriminator

Results

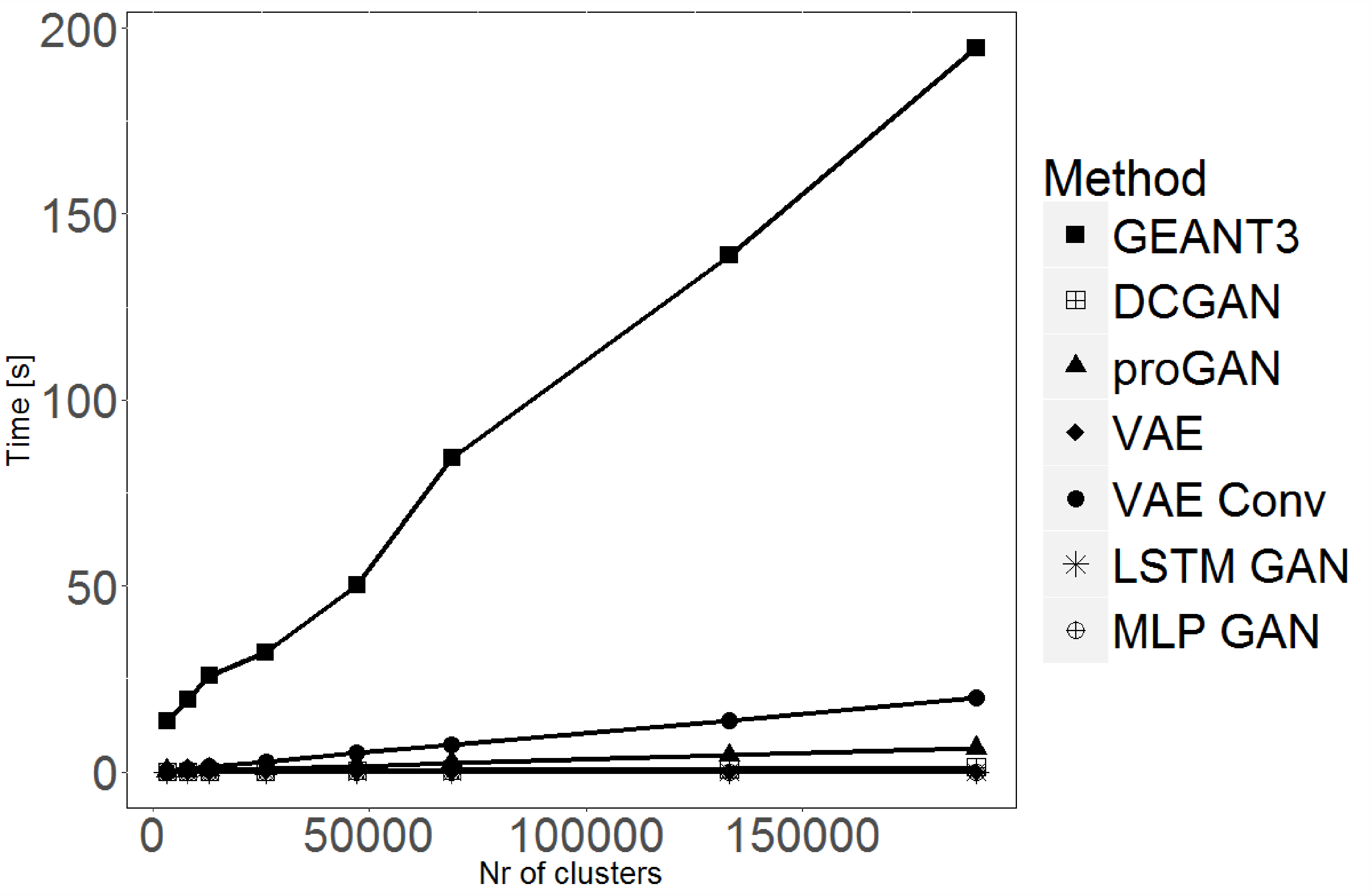
Preliminary qualitative and performance results

- Mean Squared Error (MSE) from the ideal helix as a quality measure
- 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 GTX 1080 GPU

Method	MSE(mm)	speedup
GEANT3	0.085	1
Random (estimated)	166.155	N/A
GAN-MLP	55.385	10^4
GAN-LSTM	54.395	10^4
VAE	37.415	10^4
DCGAN	26.18	10^2
cVAE	13.33	10
proGAN	0.88	30

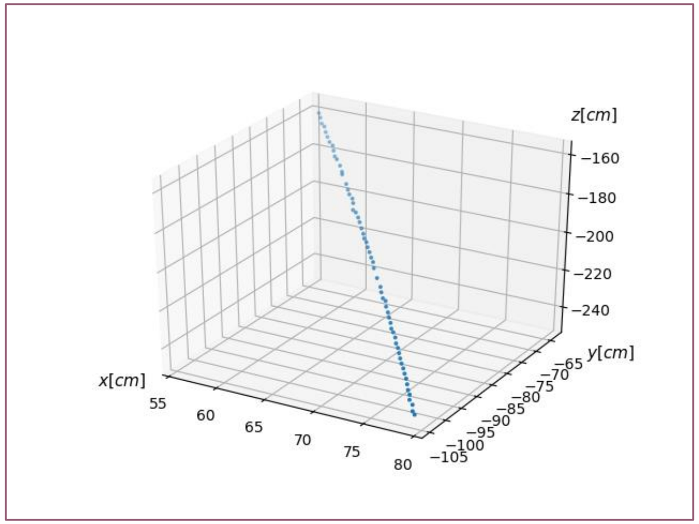
Quality of the Generative models, and their run-time comparing to the GEANT3 based simulation solution.

Preliminary performance results

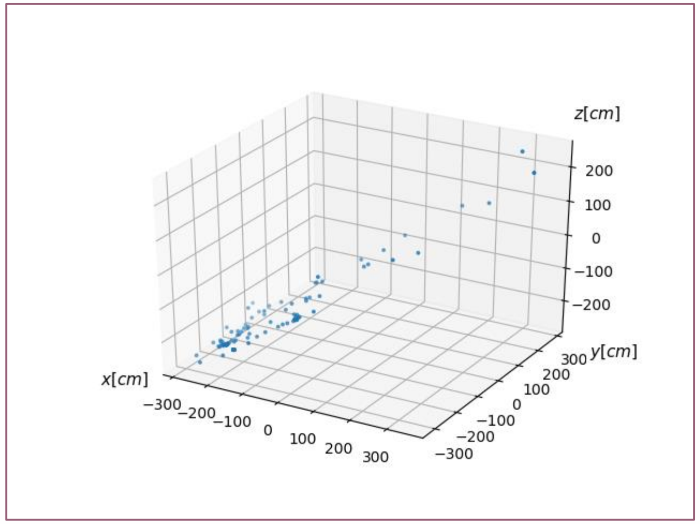


Warsaw University of Technology

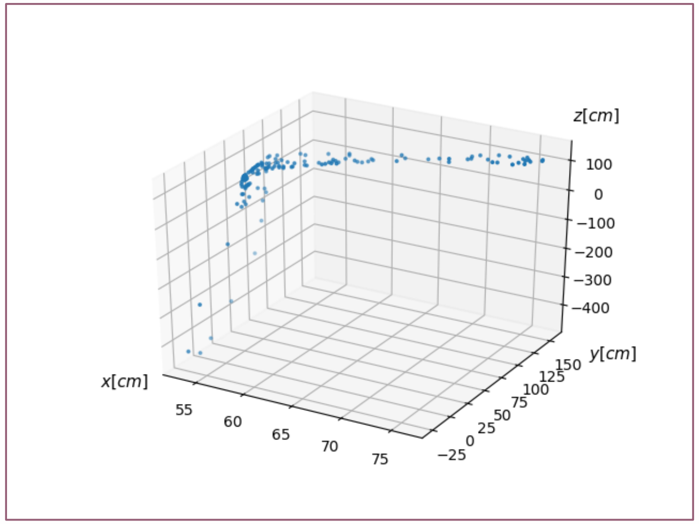
Example clusters generated by different models



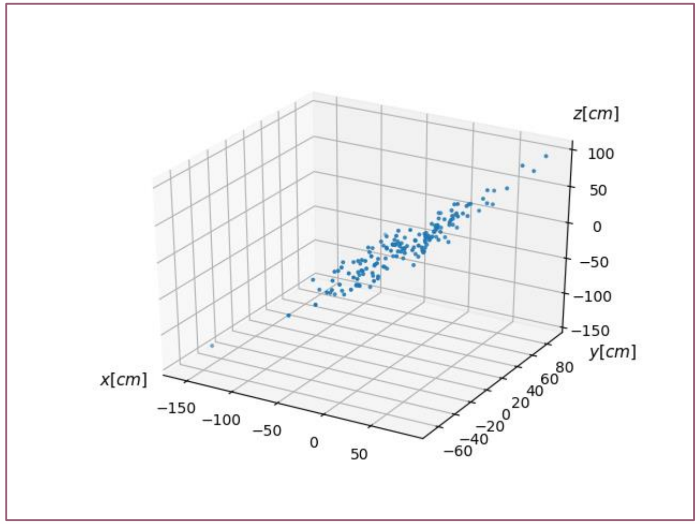
Original example



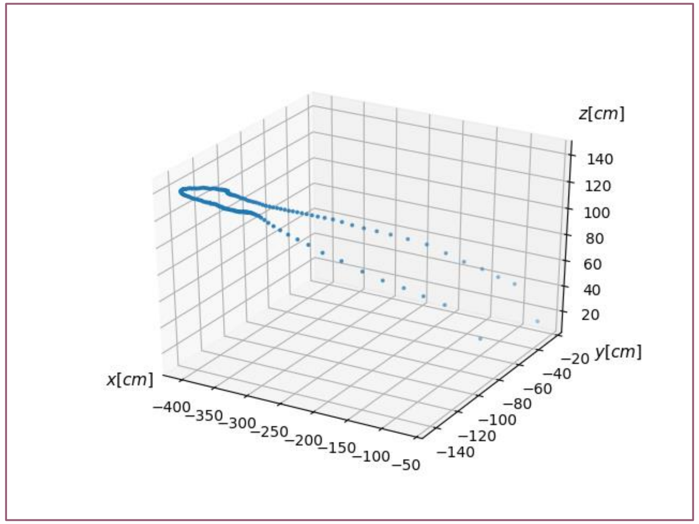
VAE



DCGAN



cVAE



proGAN

Future work

- Enhancing the quality of generated samples with additional cost applied to the loss function
- Conditional GAN for simulating particles propagation through detector based on the initial particles momenta
- Training with additional loss function straight from the original data samples
- Semi-real-time anomaly detection with GANs

References

1. R. Brun, F. Bruyant, F. Carminati, S. Giani, M. Maire, A. McPherson, G. Patrick, and L. Urban, “GEANT Detector Description and Simulation Tool,” CERNW5013, CERN-W-5013, W5013, W-5013, 1994.
2. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in NIPS, 2014.
3. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” CoRR, vol. abs/1312.6114, 2013.
4. A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” CoRR, vol. abs/1511.06434, 2015.
5. T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” CoRR, vol. abs/1710.10196, 2017.
6. F. Chollet et al., “Keras.” https://github.com/fchollet/keras, 2015.
7. T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in International Conference on Information Processing in Medical Imaging, pp. 146–157, Springer, 2017.