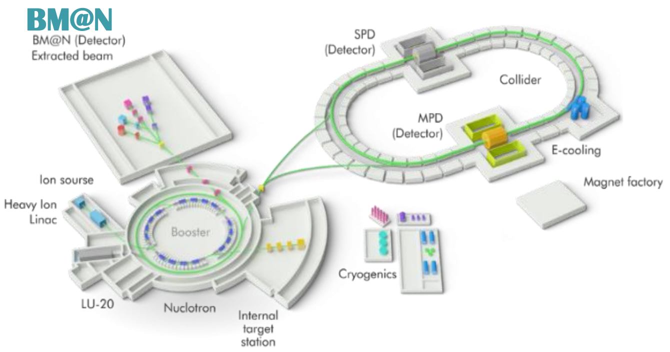
GAN-based simulation of microstrip triple GEM detector in the BM@N experiment

Baranov Dmitry

BM@N Experiment

BM@N (Baryonic Matter at Nuclotron) is the first stage experiment of mega-science project NICA

This is a fixed target experiment aimed to study dense baryonic matter on heavy-ion collisions at the NICA accelerator complex located at JINR in Dubna



At this moment **eight experimental RUNs** were performed since 2015:

BM@N EXPERIMENTAL RUNS

- **2015** — **RUN 1-2-3-4 (2015)** Beam: deuteron; Target: C, Cu
- **2016** — **RUN 5 (2016)** Beam: deuteron; Target: CH2, C, Cu, Pb
- **2017** — **RUN 6 (2017)** Beam: deuteron, C; Target: CH2, C, Al, Cu, Pb
- **2018** — **RUN 7 (2018)** Beam: Ar, Kr; Target: C, Cu
- **2023** — **RUN 8 (2023-2024)** Beam: Ar, Xe; Target: CsI
- **2025** — **RUN 9 (2025)** Beam: Xe; Target: CsI

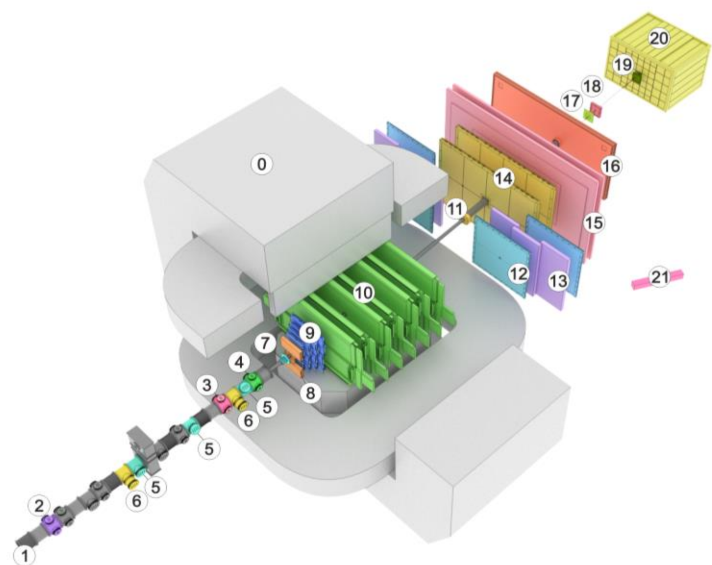
BM@N setup

TRACKING SYSTEM
- **SiBT** (Silicon Beam Tracker)
- **VSP** (Vertex Silicon Plane)
- **FSD** (Forward Silicon Detector)
- **GEM** (Gas Electron Multiplier)
- **CSC** (Cathode Strip Chamber)

PARTICLE INDENTIFICATION SYSTEM
- **TOF400** (Time Of Flight detector)
- **TOF700** (Time Of Flight detector)

OTHER SYSTEMS
- **Trigger system**
- **FQH** (Forward Quartz Hodoscope)
- **ScWall** (Scintillator Wall)
- **FHCal** (Fwd. Hadron Calorimeter)
- **HGN** (High Granularity Neutron det.)

RUN 9

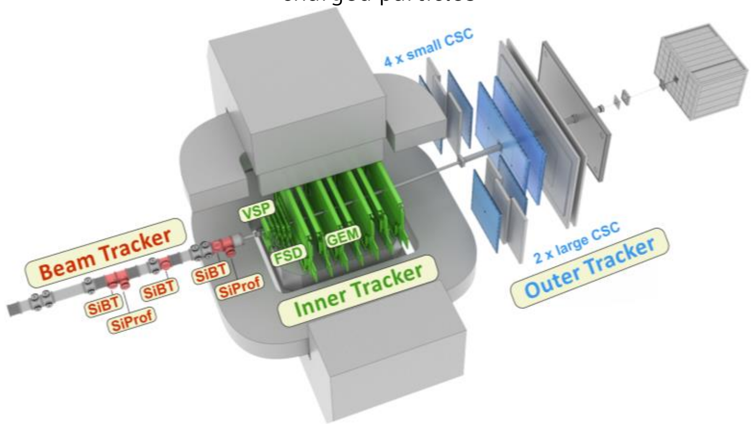


- Magnet SP-41 (0)
- Vacuum Beam Pipe (1)
- BC1, VC, BC2 (2-4)
- SiBT, SiProf (5, 6)
- BD (7)
- VSP, FSD, GEM (8, 9, 10)
- FD (11)
- 4 x CSC 1x1 m^2 (12)
- TOF 400 (13)
- 2 x CSC 2x1.5 m^2 (14)
- TOF 700 (15)
- ScWall (16)
- Small GEM (17)
- Beam Profilometer (18)
- FQH (19)
- FHCal (20)
- HGN (21)

Hybrid Tracking System: GEM Detector

Hybrid Tracking System

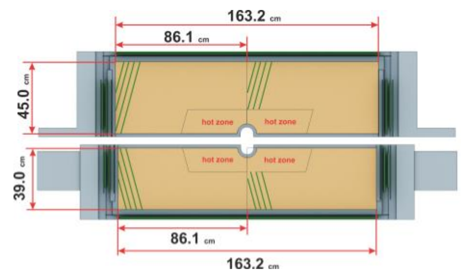
The **hybrid tracking system** of the BM@N experiment consists of the different types of coordinate detectors to register trajectories of charged particles

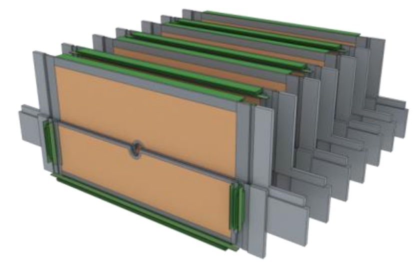


Configuration of hybrid tracker for the RUN-9

Gas Electron Multipliers (GEM)

GEM (Gas Electron Multipliers) is a microstrip coordinate detector of the inner tracker. It consists of gaseous chambers with electron multiplier system inside





The configuration of this detector for RUN-9 comprises seven stations located inside the magnet along the beam axis

Each station is assembled by two chambers: upper and lower which are joined together to form a plane

Gas volume thickness: **9 mm**
strip pitch: **800 µm**
stereo angle between strips: **15º**

Hybrid Tracker Composition

SiBT (Silicon Beam Tracker)
(3 planes of 63x63 mm)

SiProf (Silicon Beam Profilometer)
(2 planes of 63x63 mm)

It includes detectors located inside the vacuum pipe in front of magnet SP-41
BEAM TRACKER

It comprises coordinate detectors located inside the magnet
INNER TRACKER

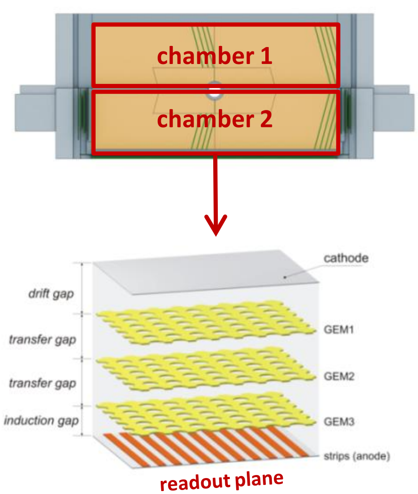
Hybrid Tracking System

It consists of detectors located behind the magnet
OUTER TRACKER

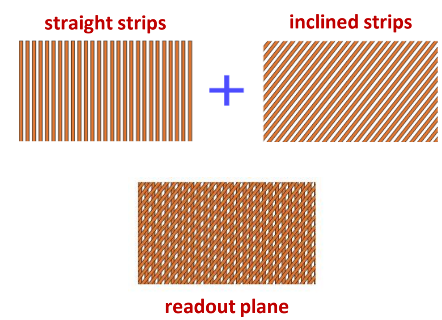
- **VSP** (Vertex Silicon Plane based on STS) *(one plane of 6 modules)*
- **FSD** (Forward Silicon Detector) *(8 half planes)*
- **GEM** (Gas Electron Multipliers) *(14 half planes)*

- **small CSC** (Cathode Strip Chambers) *(4 planes of 1x1 m)*
- **large CSC** (Cathode Srtrip Chambers) *(2 planes of 2x1.5 m)*

Triple GEM Chamber

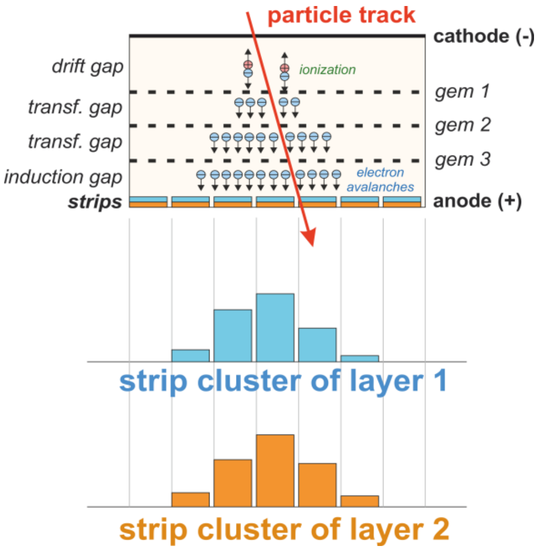


Readout plane of GEM chamber assembled by two strip layers (straight and inclined strips)



Signal Formation in Triple GEM Chamber

Operation Principles of GEM



<u>Stages of signal formation in a GEM chamber:</u>

Gas ionization

A particle passes through the detector and ionizes gas molecules, producing electron-ion pairs. Positive ions and electrons drift to the cathode and to the anode, respectively.

Electron multiplication

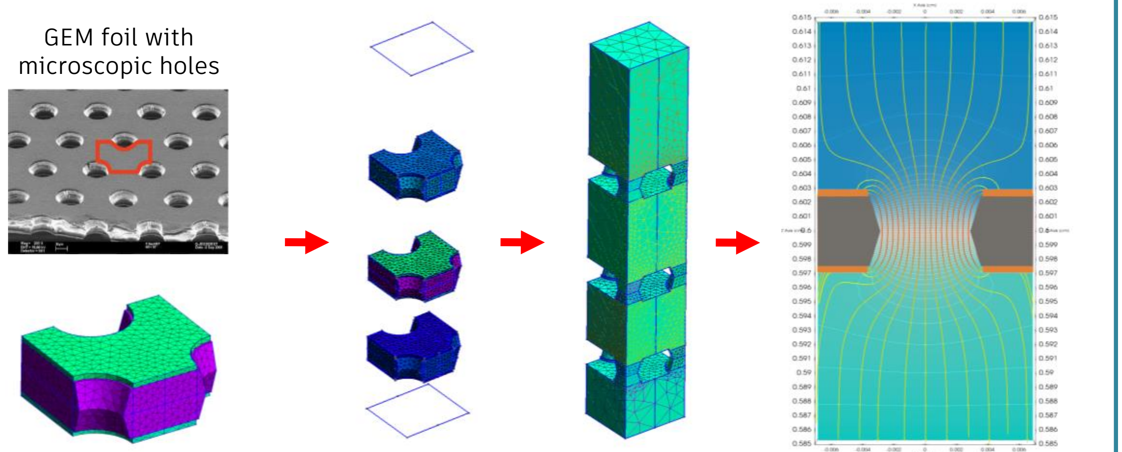
Primary electrons, passing through amplifying GEM cascades, gain their kinetic energy and enable secondary ionization. As a result of it is a lot of secondary electrons (electron avalanches). Amplification is about $10^4 - 10^5$.

Charge collection

Being collected on the anode, electrons form clusters on each strip layer.

Detailed Description of GEM

GEM foil with microscopic holes

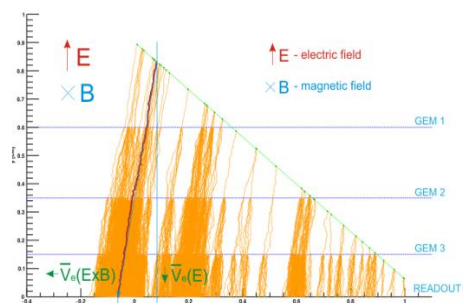


GEM cell

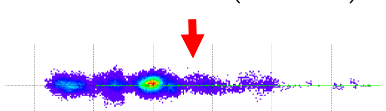
Assembling of GEM cells (with **GMSH**)

Calculation of electric field map of GEM (with **ELMER**)

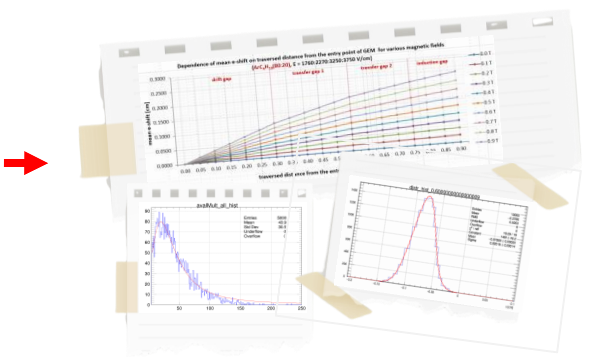
Detailed Simulation of GEM with Garfield++



Example of electron avalanche production in GEM chamber (Garfield++)



Cluster of electrons collected by the readout plane (before they are distributed on strips)



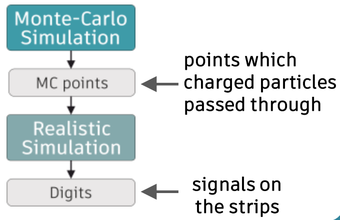
- ❑ Dependence of mean electron shift on magnetic field
- ❑ Dependence of mean electron diffusion on drift path
- ❑ Distribution of mean free path for particles
- ❑ Distribution of electron multiplication in gem-holes
- ❑ etc

Approaches to GEM Simulation

Simulation Procedure

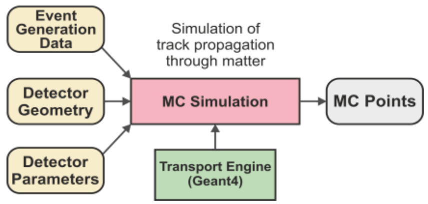
Full simulation consists of **two stages**:

1. Monte-Carlo simulation
2. Realistic simulation

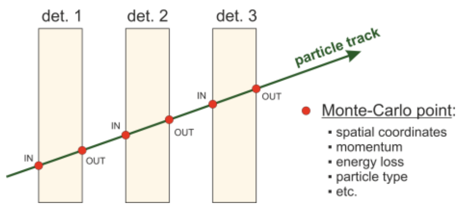


Stage 1: Monte-Carlo simulation

Monte-Carlo simulation is used for simulation of charged particle passage through matter. In order to do this Geant4 transport engine is used

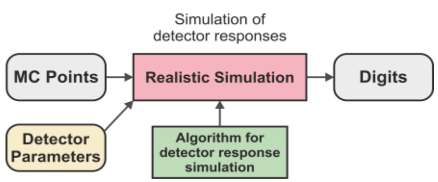


<u>Result:</u> A set of **MC points** which charged particles left in detectors

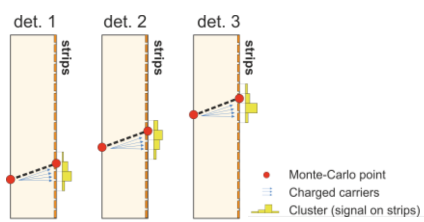


Stage 2: Realistic simulation

Realistic simulation is used to create signals on the strips (clusters) taking into account the features of signal formation



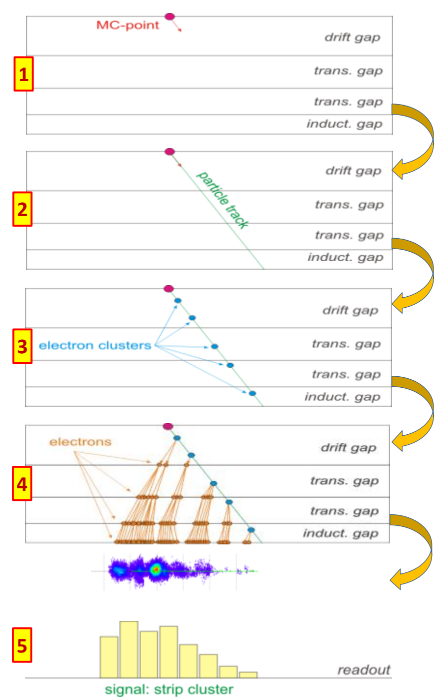
<u>Result:</u> A set of **digits** (lighted strips) as real responses of detectors



Methods of Realistic Simulation of GEM

Very slow: hours to generate a single cluster	Currently used for GEM-simulation in BMNROOT	**New approach** to simulation of clusters
GARFIELD++ simulation →	**Standard simulation** →	**GAN-based simulation**
Used to calculate dependencies and distributions required for standard algorithm	Used to prepare dataset for GAN training	Replaces the steps [2-4] of the standard algorithm

Standard Realistic Simulation Algorithm



1. Information extracted from a MC-point is used to initialize inputs parameters
2. Reconstruction of a particle trajectory in the chamber (line from the entry point to the exit point of the particle)
3. Calculation of points along the track based on the ionization characteristics from Garfield++
4. Multiplication of electrons and calculation of their positions on the readout plane based on the distributions and dependencies from Garfield++
5. Result: **clusters on the strips**

Generative Networks for Cluster Generation

GAN

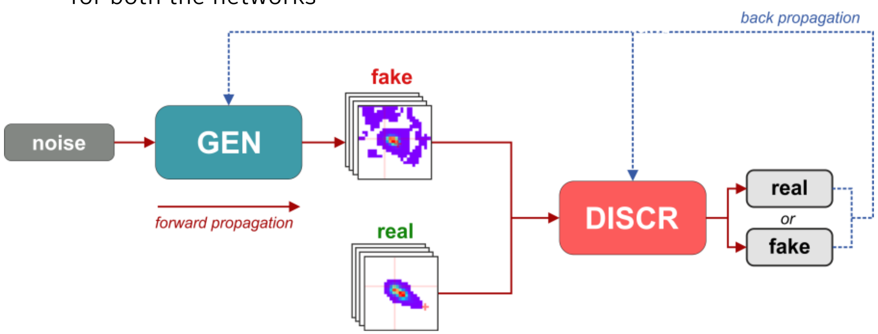
GAN (**G**enerative **A**dversarial **N**etwork) consists of two neural networks – generator and discriminator. The generator is used to create new data samples, the discriminator is used to distinguish between real samples and generated ones

GAN Modes

Training mode

Goal: Improvement of the model

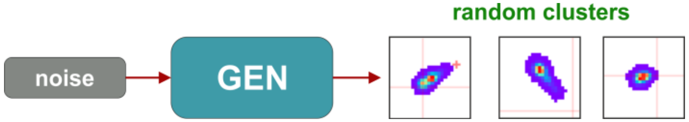
- Both the generator and discriminator are used
- The generator tries to fool the discriminator, while the discriminator tries to distinguish real from fake data
- **Forward propagation**: input flows through the network to produce output (getting prediction)
- **Back propagation**: gradients are computed and weights are updated for both the networks



Inference mode

Goal: Generation of new realistic samples

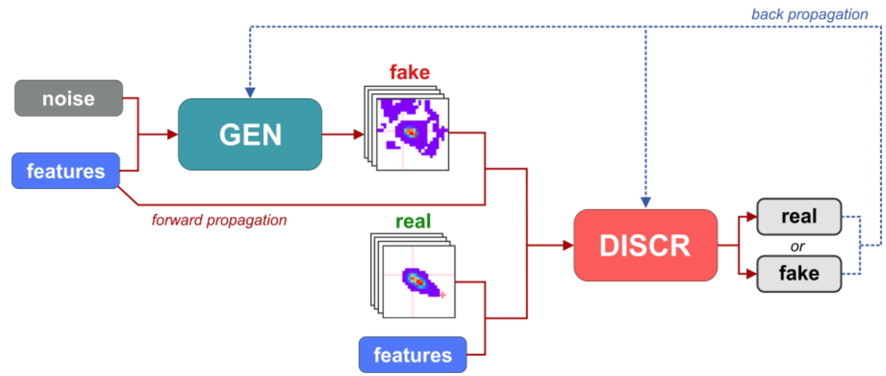
- Only generator is used
- Forward propagation only is used to create samples



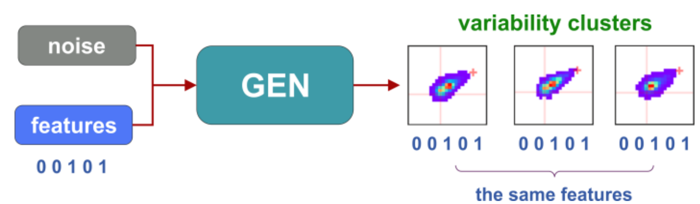
GAN is not allows to generate clusters with predefined parameters!

Conditional GAN

C-GAN (**C**onditional **G**enerative **A**dversarial **N**etwork) is an extension of a standard GAN, where both the generator and discriminator are conditioned on additional information (features) for controlled data generation



To simulate realistic responses from GEM chambers, only the **generator** network (trained on prepared samples) is used



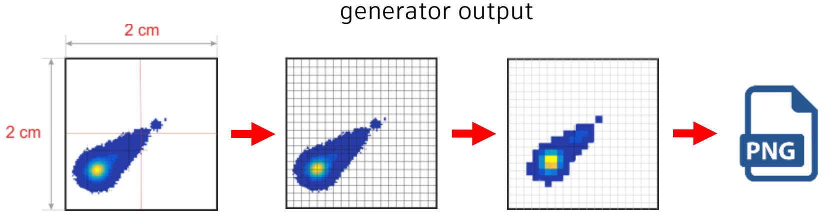
By feeding the network with desired features, the corresponding clusters can be obtained

C-GAN allows us to control the generation of clusters based on MC information

Dataset Preparation

Training Samples

One training sample represents a cluster as a 2D tensor (matrix) of size **64×64**. The form of a 2D tensor defines the generator output

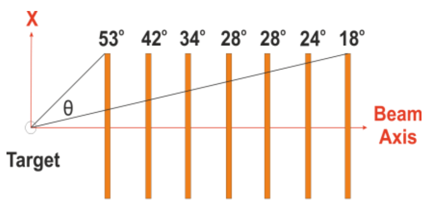


area size 2x2 cm | binning 64x64 cells | matrix 64x64 | PNG picture 64x64 pixels

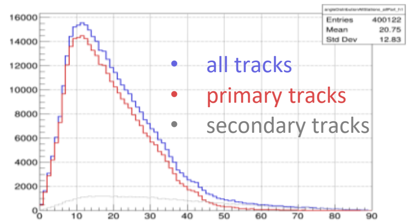
Steps of sample preparation:

1. Choosing the size of the description area for a cluster (2x2 cm)

- The area should include clusters from tracks with an angle of inclination to the beam axis from 0 to 50 degrees



Acceptance angle for each station in GEM (RUN-9)



Distribution of angles of tracks in GEM (RUN-9)

2. Binning the description area into 64×64 cells

- The size of a bin is 2/64 = 0.03125 cm
- Cell RMS = $\frac{bin\_size}{\sqrt{12}} = 0.009\ cm$ (90 μm)

3. Converting the 64×64 matrix into a PNG image, where each cell value is encoded as a 24-bit (into 3 RGB channels)

- The size of one uncompressed PNG file is about 700 B
- The size of one uncompressed TXT file is about 12 KB

Result: training dataset of **800 000 samples** was prepared

Input Parameters

1. Components of the particle's momentum **px**, **py**, **pz** (from a MC point)

- px, py, pz are normalized to the range [-1, +1] under the condition:

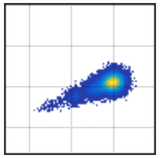
$$R = \sqrt{px^2 + py^2 + pz^2} = 1$$

- px, py, pz can be transformed into a spherical coordinate system for better interpretability

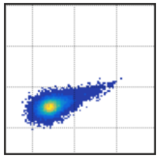
$$R = \sqrt{px^2 + py^2 + pz^2}, \quad \theta = \arccos\left(\frac{pz}{R}\right), \quad \varphi = \arctan\left(\frac{py}{px}\right)$$

2. Electron drift direction $\mathbf{E_{drift}}$ (forward or backward)

- E_{drift} is a binary parameter that can be either 0 or 1, depending on the GEM chamber orientation. It influences the resulting cluster shape

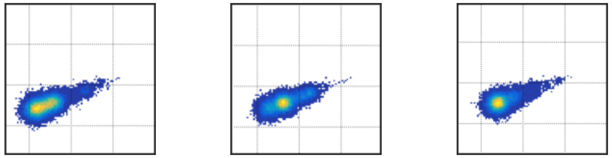


forward orientation



backward orientation

3. Random noise vector **noise_vec** (64 dimensions) used to introduce variability in the generation process

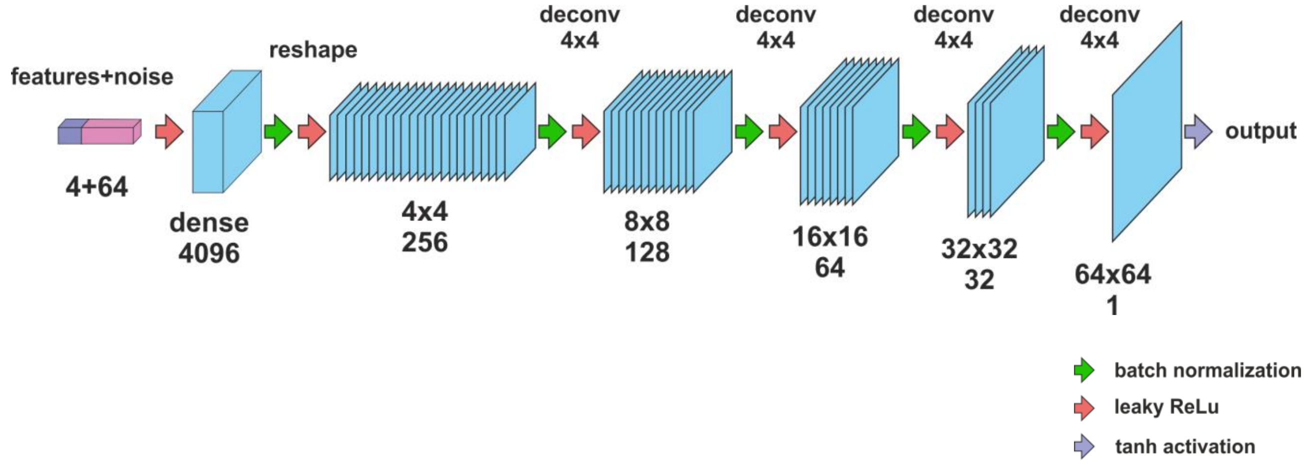


Example of three clusters with the same px, py, pz, and E_{drift} values, but different random noise vectors

Conditional GAN Architecture

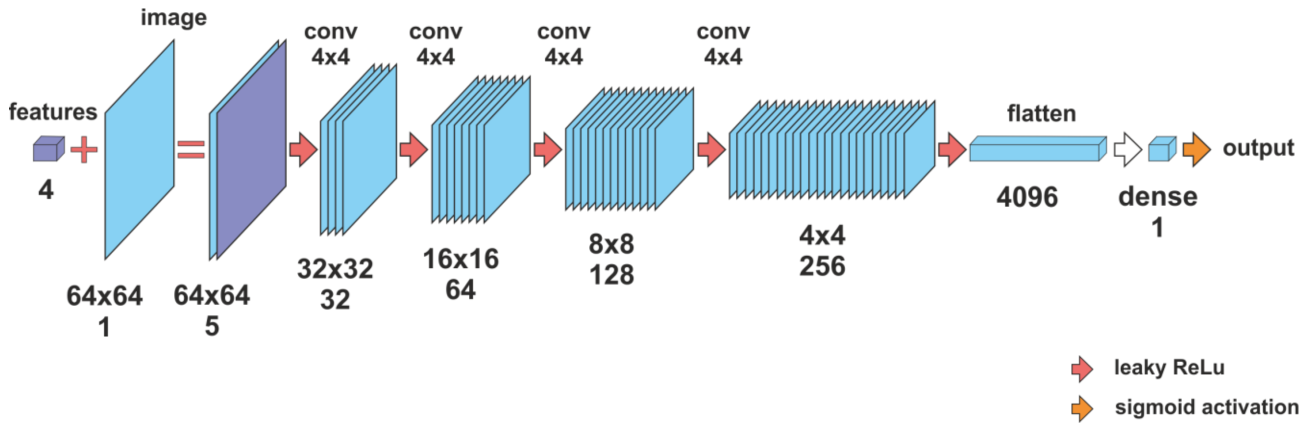
Generator Network

- **INPUT**: features (4) + noise (64)
- Dense layer (4096)
- Batch normalization
- Activation: LeakyRelu
- Reshape layer (4x4x256)
- Conv2DTranspose layer (256, 4x4)
- Batch normalization
- Activation: LeakyRelu
- Conv2DTranspose layer (128, 4x4)
- Batch normalization
- Activation: LeakyRelu
- Conv2DTranspose layer (64, 4x4)
- Batch normalization
- Activation: LeakyRelu
- Conv2DTranspose layer (32, 4x4)
- Batch normalization
- Activation: LeakyRelu
- Conv2DTranspose layer (1, 4x4)
- **OUTPUT**: Activation: tanh [-1, +1]



Discriminator Network

- **INPUT**: features (4) + image (64x64x1)
- Concatenate layer (64x64x5)
- Conv2D layer (32, 4x4)
- Activation: LeakyRelu
- Conv2D layer (64, 4x4)
- Activation: LeakyRelu
- Conv2Dlayer (128, 4x4)
- Activation: LeakyRelu
- Conv2D layer (256, 4x4)
- Activation: LeakyRelu
- Flatten layer (4096)
- **OUTPUT**: Activation: sigmoid [0, +1]



Conditional GAN Training

Training Setup

Framework: Tensorflow with Keras API (Python)

Optimizer: ADAM (Adaptive Moment Estimation)

- α = 0.0002, β_1 = 0.5, β_2 = 0.999

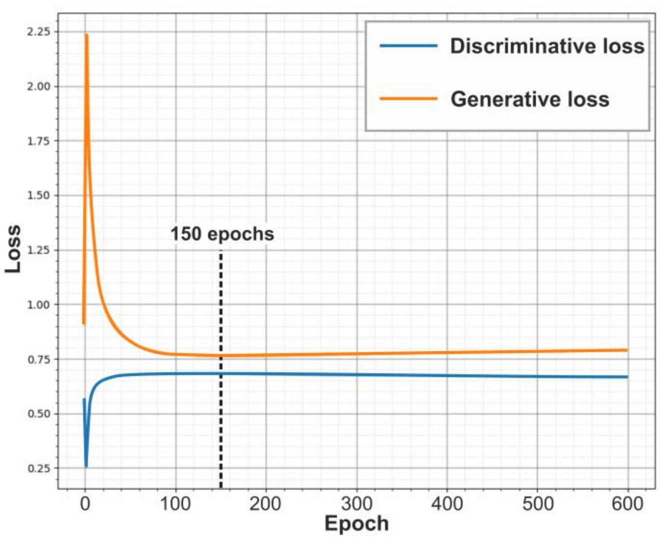
Loss function: binary cross-entropy

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_{true}\log(y_{pred}) + (1 - y_{true})\log(1 - y_{pred})\right]$$

Training dataset size: 800 000 samples

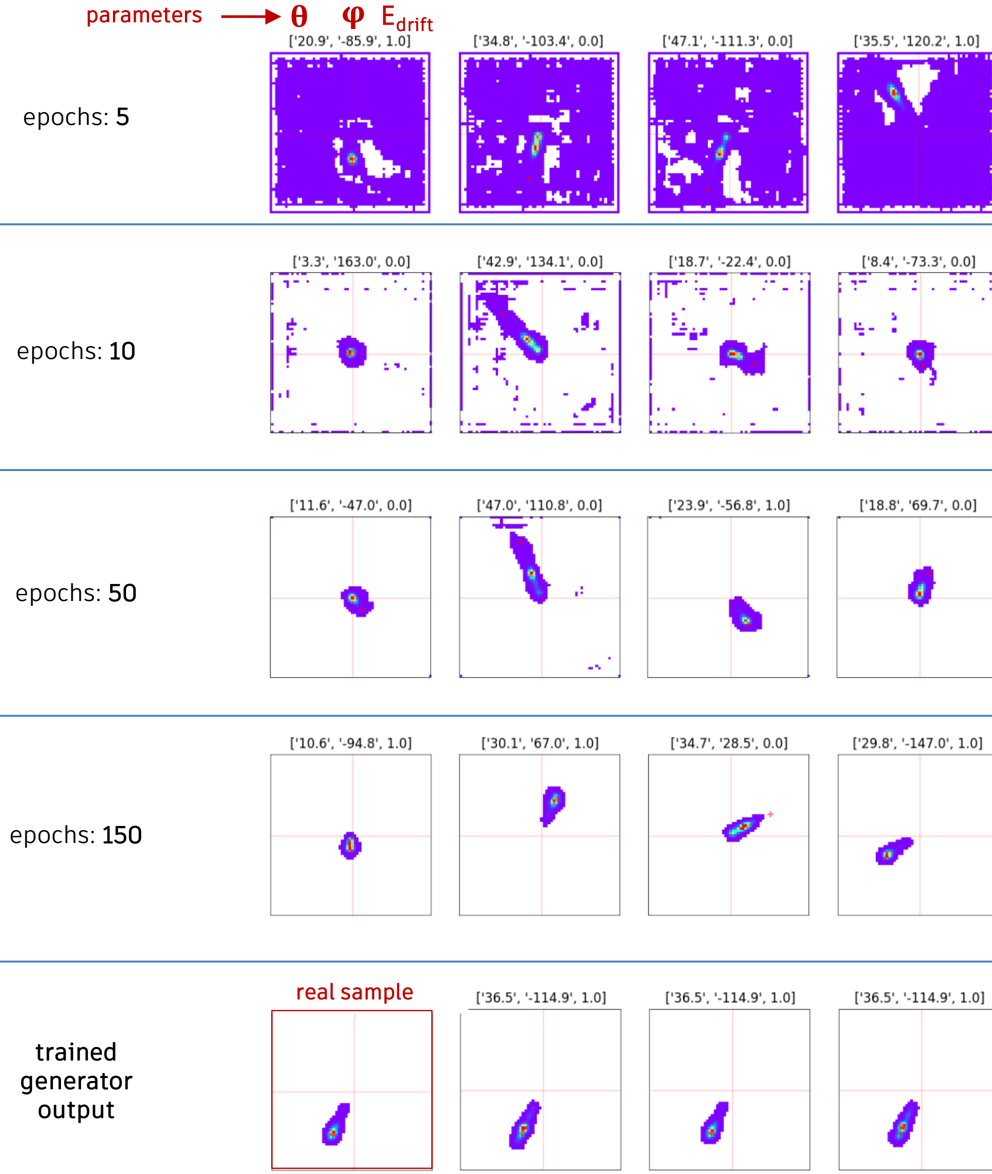
Number of epochs: 600 (150 is optimal)

Training Loss Curves



Training loss curves: discriminative and generative loss

Generated Sample Evolution

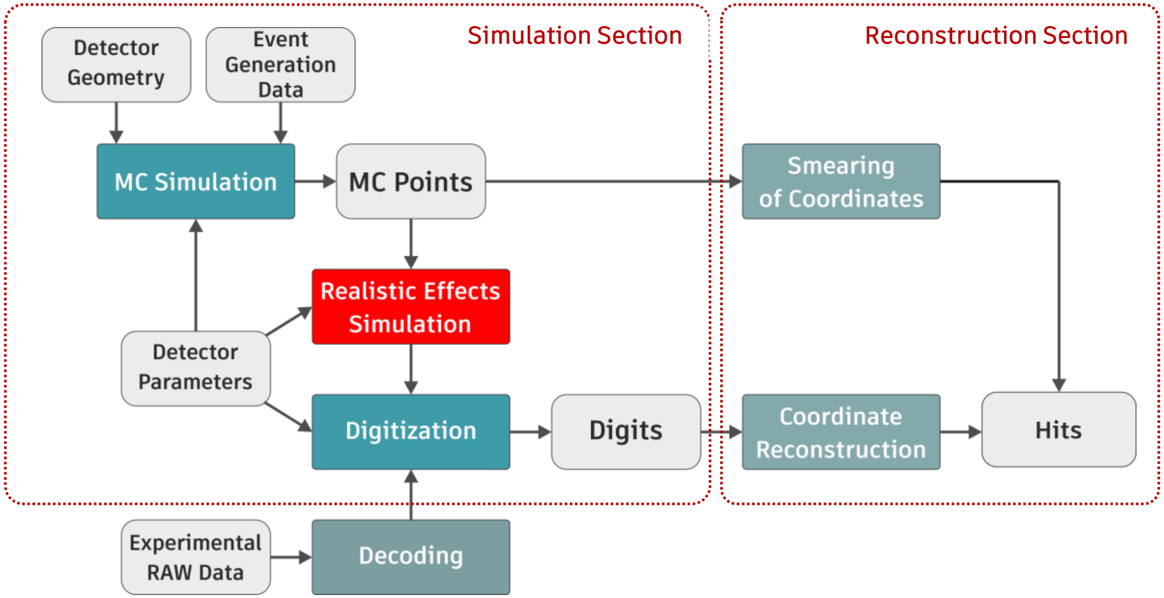


Generative Model: Integration

Data Processing in BMNROOT

Framework **BMNROOT** was developed for software support of the BM@N experiment. It provides powerful tools for simulation, reconstruction and data analysis

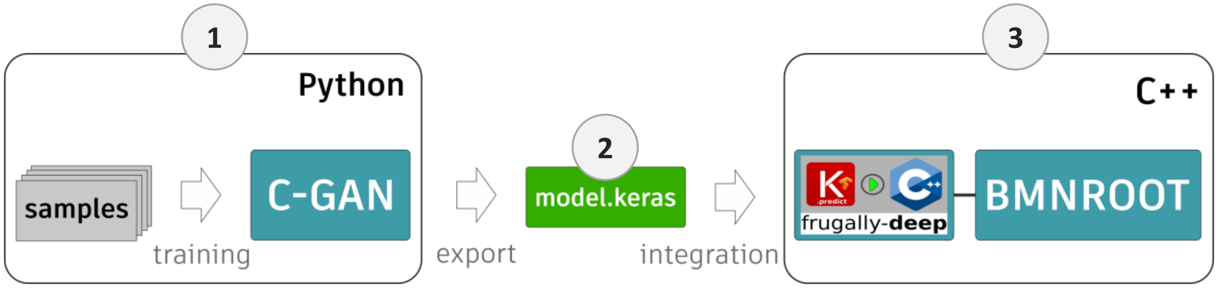
<u>Goal:</u> Replace the standard model of the realistic effects simulation with the generative model



Integration of Model into BMNROOT

<u>Steps:</u>

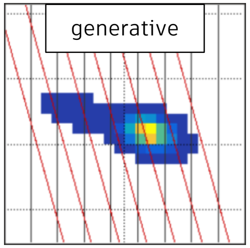
1. Training C-GAN on samples using **Tensorflow** framework (Python)
2. Exporting the model to “keras” format
3. Using **f-deep** library to integrate the model into a C++ application (BMNROOT)



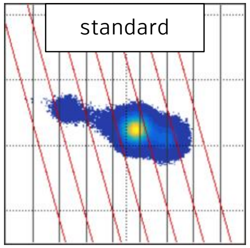
Generative Model: Evaluation (strip clusters)

Comparison of Two Methods

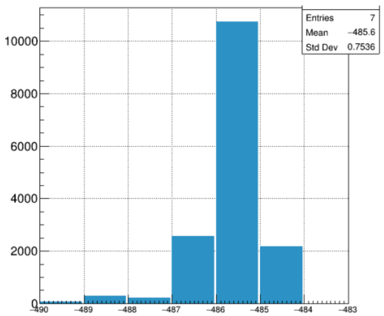
Goal of the evaluation: to compare the generative model and standard model



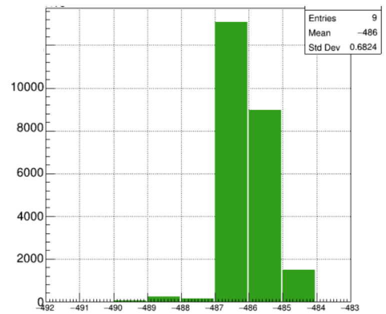
generated cluster on readout plane



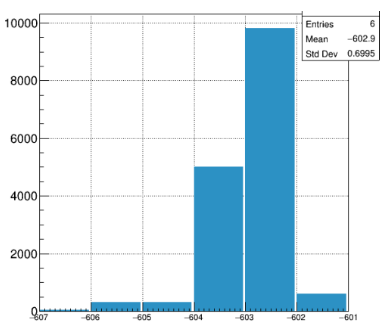
cluster on readout plane



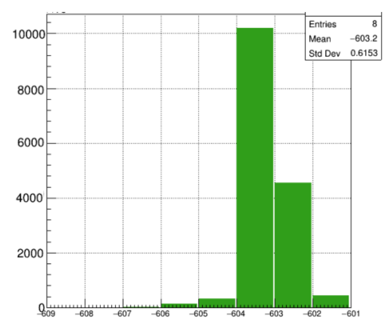
generated cluster on straight strips



cluster on straight strips

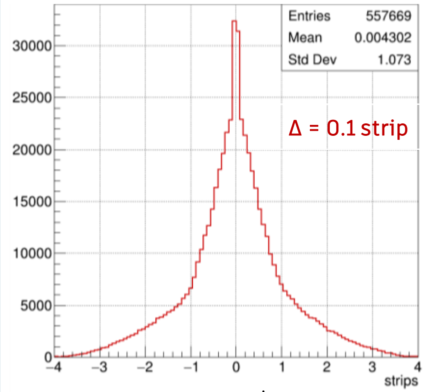


generated cluster on inclined strips

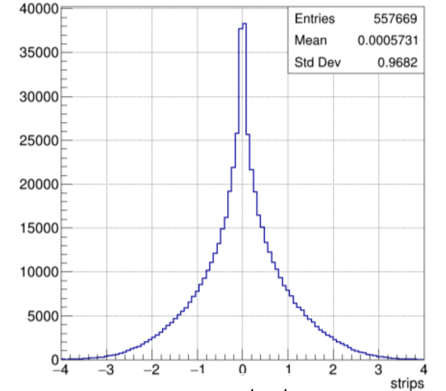


cluster on inclined strips

Residuals of **mean position of cluster** and **position of MC point** for generative and standard models (in strip units)

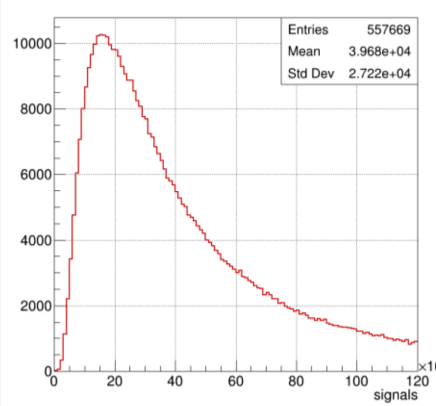


generative

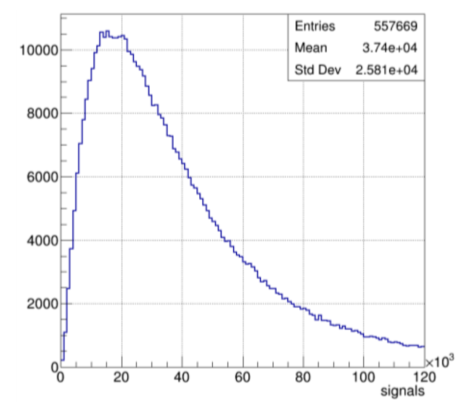


standard

Distribution of **cluster signals** for generative and standard models



generative

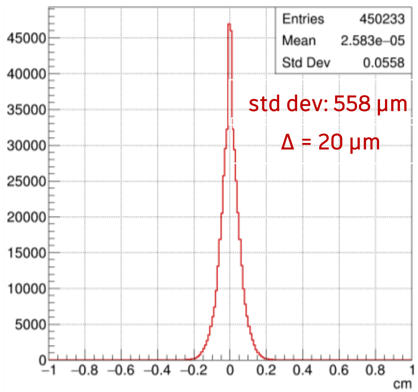


standard

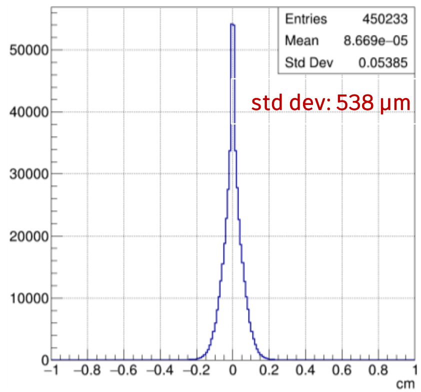
Generative Model: Evaluation (reconstructed coordinates)

Residuals for tracks 0-50 degrees

X-residuals of **MC point** and **reconstructed point** (cm)

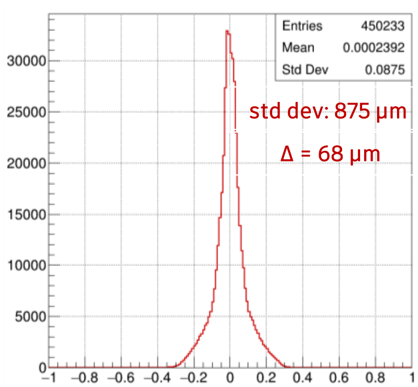


generative

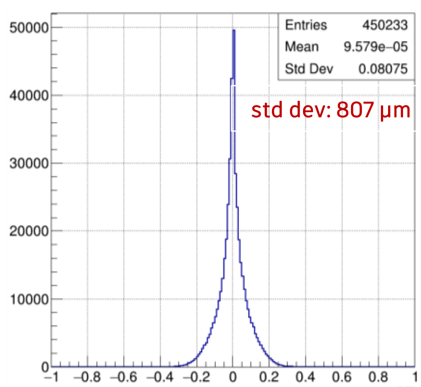


standard

Y-residuals of **MC point** and **reconstructed point** (cm)



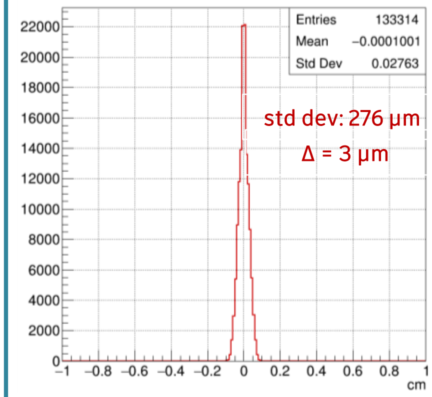
generative



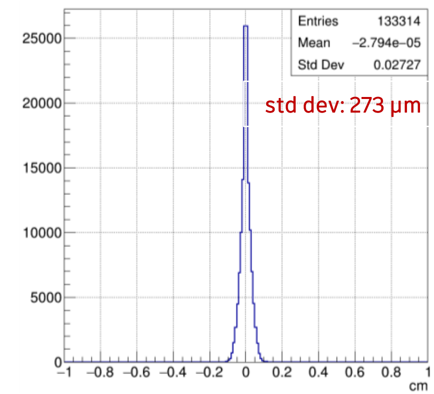
standard

Residuals for tracks 0-20 degrees

X-residuals of **MC point** and **reconstructed point** (cm)

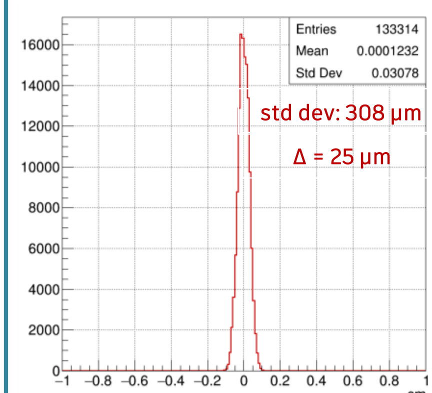


generative

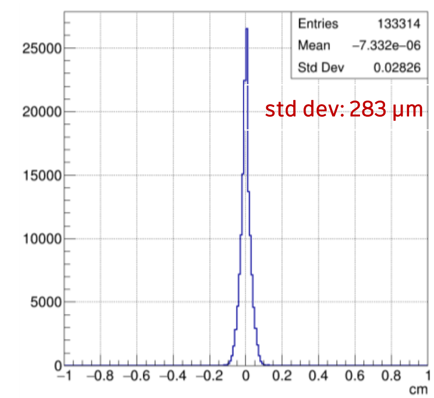


standard

Y-residuals of **MC point** and **reconstructed point** (cm)



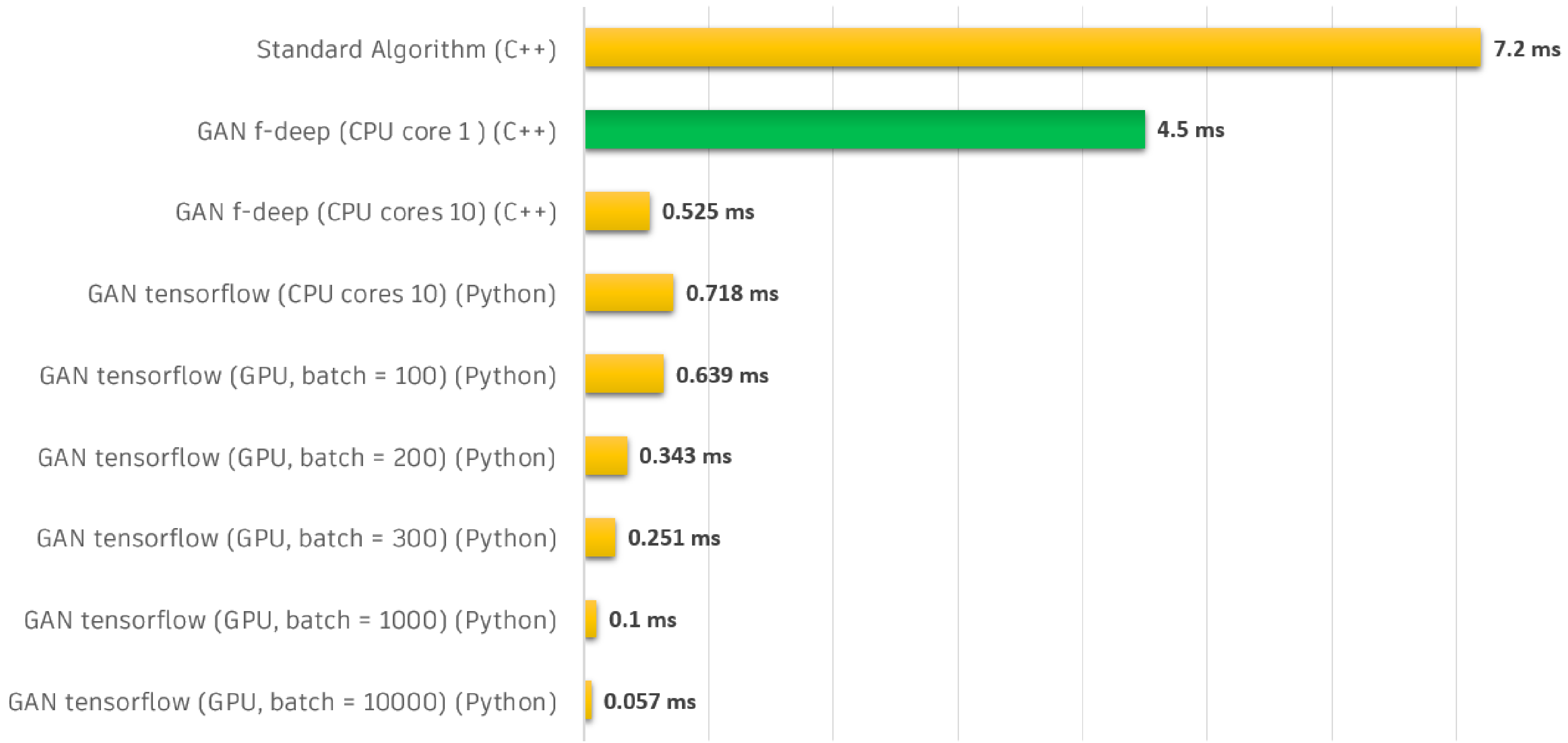
generative



standard

Generative Model: Inference Performance

Generation Time per Sample



- Using the generative model in multi-core mode requires some changes to event processing
- One event has about 200–300 Monte-Carlo points in the GEM detector (defines the maximum batch size)

Summary

What has been done:

- Generative model based on a Conditional GAN was developed for realistic response simulation in the GEM detector

What is planned next:

- Include magnetic field effects in the cluster generation model
- Implement multicore processing support to improve performance
- Other improvements and model optimization

Thank you for your attention …