Fast TPC simulation

IX Collaboration Meeting of the MPD Experiment at the NICA Facility
April 25-27, 2022

Artem Maevskiy[1], Fedor Ratnikov[1,2], Alexander Zinchenko[3], Victor Riabov[4], Dmitry Evdokimov[1], Aleksey Sukhorosov[1]

[1]HSE University (National Research University Higher School of Economics)
[2]Yandex School of Data Analysis
[3]Joint Institute for Nuclear Research
[4]Petersburg Nuclear Physics Institute

LAMBDA • HSE

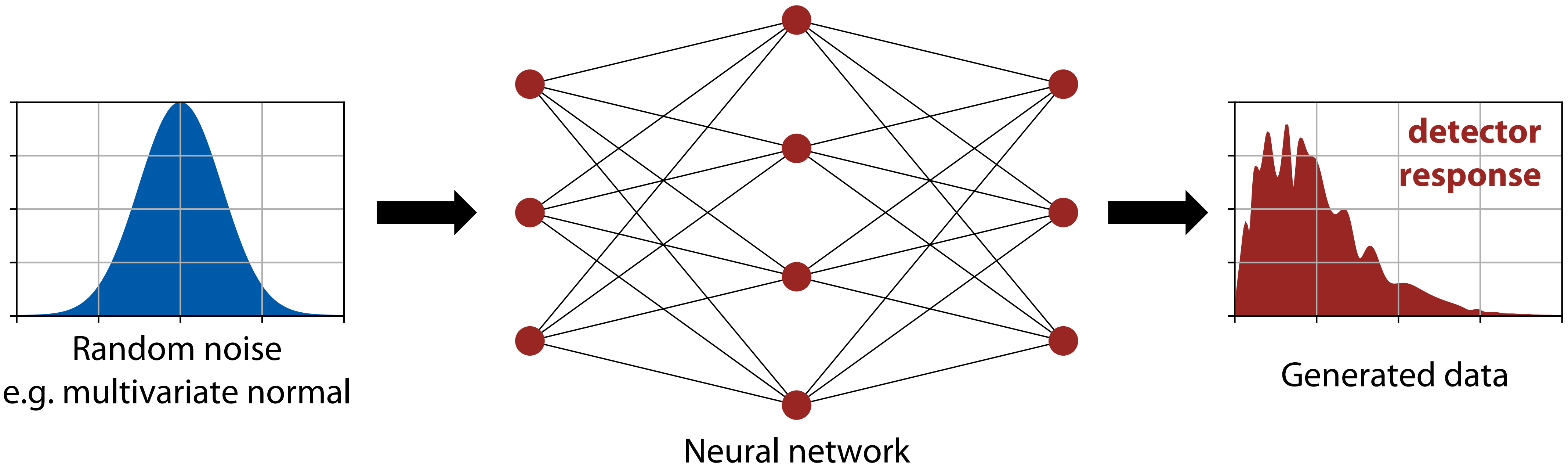
April 26, 2022

Outline

- GANs for fast simulation
- Our TPC fast simulation model & results
- Fast simulation pipeline developments
- Possible improvements

Generative Adversarial Networks for Fast Simulation

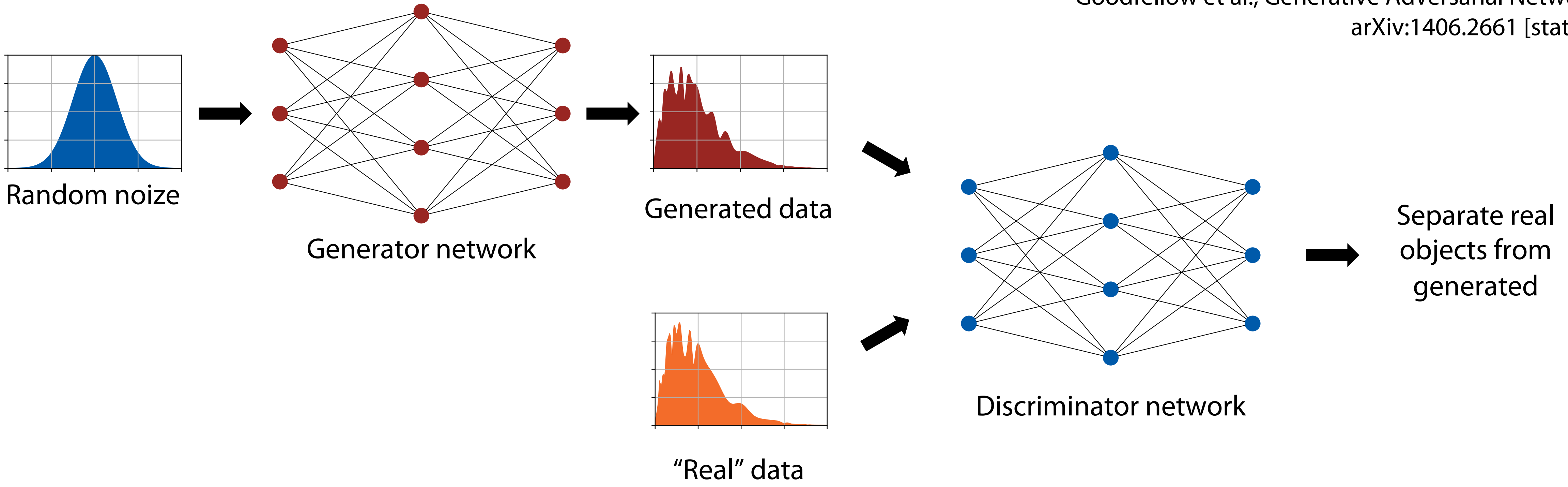
How can a neural network generate data?



- This makes the generated object being a **differentiable function** of the network parameters
  - The parameters of the network can be optimized with gradient-based methods
- Generating a sample is as fast as a **single forward pass** through the net

Adversarial approach

Goodfellow et al., Generative Adversarial Networks,
arXiv:1406.2661 [stat.ML]



- Measure of similarity: how well can another neural network (discriminator) tell the generated objects apart from the real ones

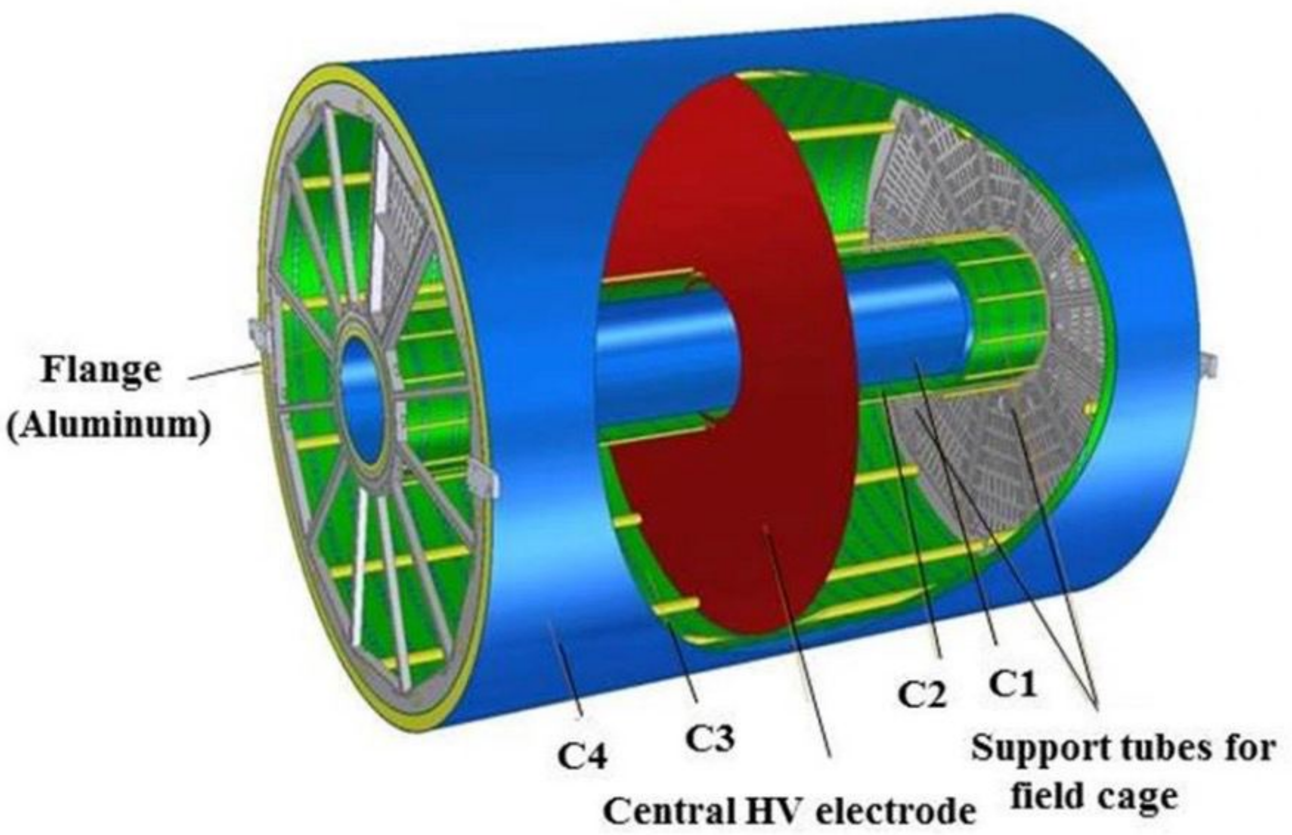
GANs for fast simulation

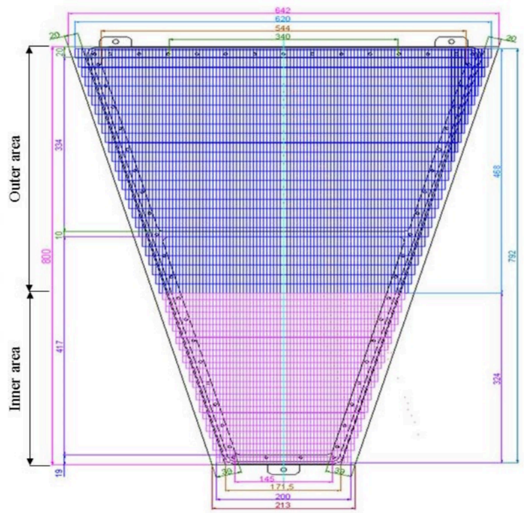
- Quite a developing field!

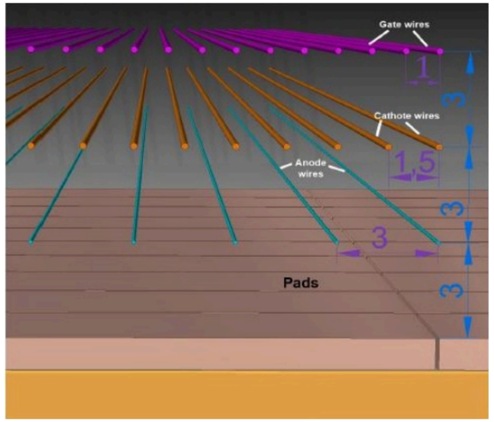
- GANs:
  - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis [DOI]
  - Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters [DOI]
  - CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks [DOI]
  - Image-based model parameter optimization using Model-Assisted Generative Adversarial Networks [DOI]
  - How to GAN Event Subtraction [DOI]
  - Particle Generative Adversarial Networks for full-event simulation at the LHC and their application to pileup description [DOI]
  - How to GAN away Detector Effects [DOI]
  - 3D convolutional GAN for fast simulation
  - Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks [DOI]
  - Lund jet images from generative and cycle-consistent adversarial networks [DOI]
  - How to GAN LHC Events [DOI]
  - Machine Learning Templates for QCD Factorization in the Search for Physics Beyond the Standard Model [DOI]
  - DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC [DOI]
  - LHC analysis-specific datasets with Generative Adversarial Networks
  - Generative Models for Fast Calorimeter Simulation.LHCb case [DOI]
  - Deep generative models for fast shower simulation in ATLAS
  - Regressive and generative neural networks for scalar field theory [DOI]
  - Three dimensional Generative Adversarial Networks for fast simulation
  - Generative models for fast simulation
  - Unfolding with Generative Adversarial Networks
  - Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks [DOI]
  - Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks [DOI]
  - Generative models for fast cluster simulations in the TPC for the ALICE experiment
  - RICH 2018 [DOI]
  - GANs for generating EFT models [DOI]
  - Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network [DOI]
  - Reducing Autocorrelation Times in Lattice Simulations with Generative Adversarial Networks [DOI]
  - Tips and Tricks for Training GANs with Physics Constraints
  - Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters [DOI]
  - Next Generation Generative Neural Networks for HEP
  - Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics
  - Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics [DOI]
  - Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed
  - AI-based Monte Carlo event generator for electron-proton scattering
  - DCTRGAN: Improving the Precision of Generative Models with Reweighting [DOI]
  - GANplifying Event Samples
  - Graph Generative Adversarial Networks for Sparse Data Generation in High Energy Physics
  - Simulating the Time Projection Chamber responses at the MPD detector using Generative Adversarial Networks
  - Explainable machine learning of the underlying physics of high-energy particle collisions
  - A Data-driven Event Generator for Hadron Colliders using Wasserstein Generative Adversarial Network [DOI]
  - Reduced Precision Strategies for Deep Learning: A High Energy Physics Generative Adversarial Network Use Case [DOI]
  - Validation of Deep Convolutional Generative Adversarial Networks for High Energy Physics Calorimeter Simulations
  - Compressing PDF sets using generative adversarial networks
  - Physics Validation of Novel Convolutional 2D Architectures for Speeding Up High Energy Physics Simulations
  - The use of Generative Adversarial Networks to characterise new physics in multi-lepton final states at the LHC
  - Latent Space Refinement for Deep Generative Models
  - Particle Cloud Generation with Message Passing Generative Adversarial Networks
  - Black-Box Optimization with Local Generative Surrogates [url]
  - Fast Simulation of a High Granularity Calorimeter by Generative Adversarial Networks
  - Photon detection probability prediction using one-dimensional generative neural network
  - Polarization measurement for the dileptonic channel of $W^+ W^-$ scattering using generative adversarial network

https://github.com/iml-wg/HEPML-LivingReview

Time projection chamber







Pad and wire planes

95 232 pads

310 time buckets

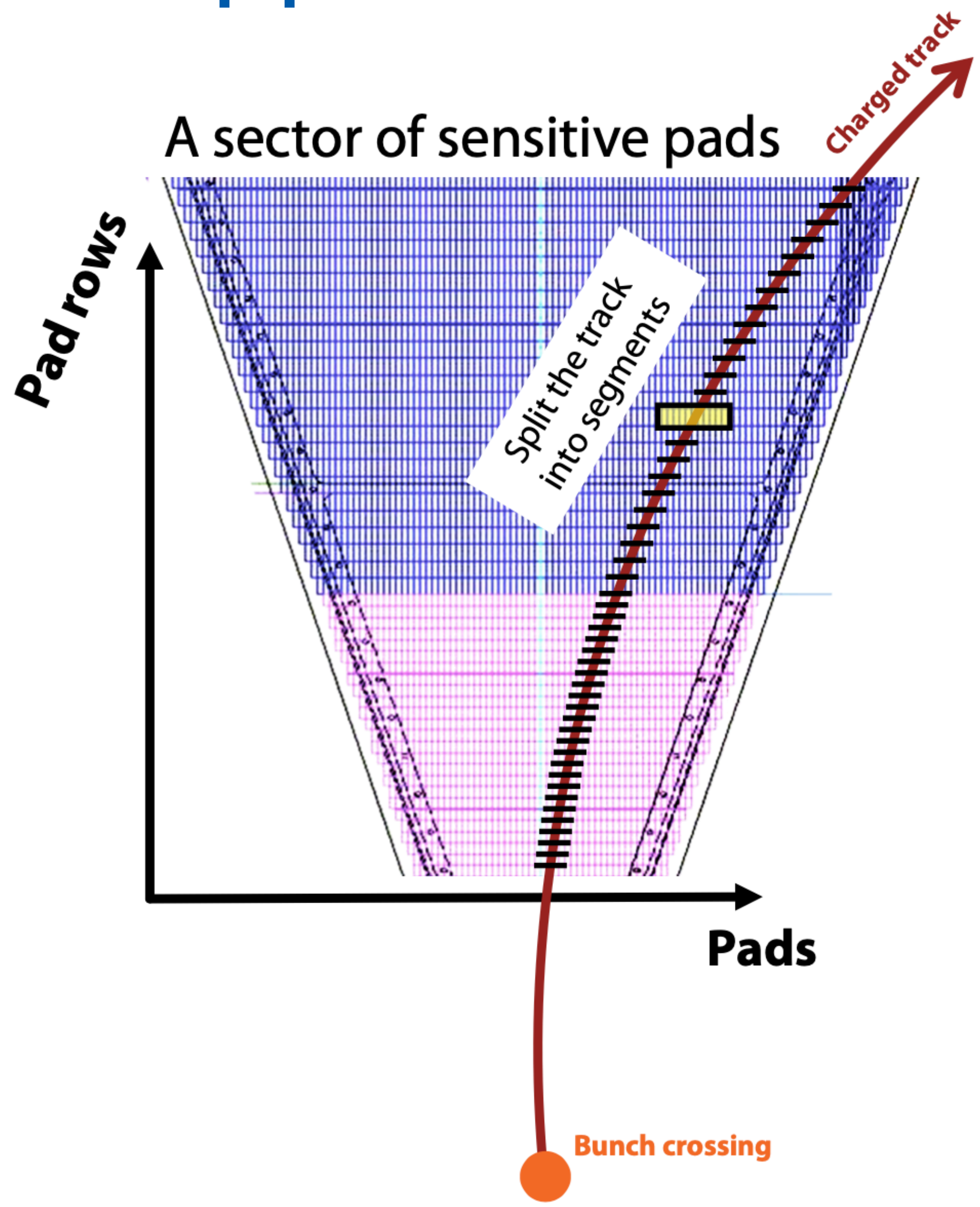
Digitization time: ~25 sec/event

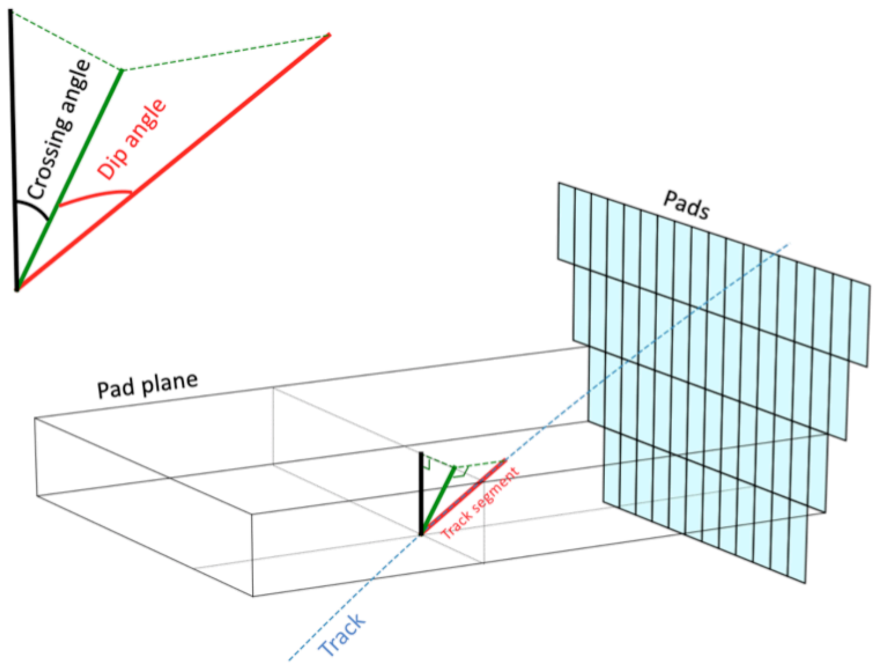
Goal:
develop a deep learning model for faster digitization of TPC

http://mpd.jinr.ru/wp-content/uploads/2019/01/TpcTdr-v07.pdf

Our approach to fast simulating TPC

Our approach





- Factorizing the pad rows
- Signal localization (both position & time)
  - model only a **small area** in a **few time buckets**
- Target dimensionality:

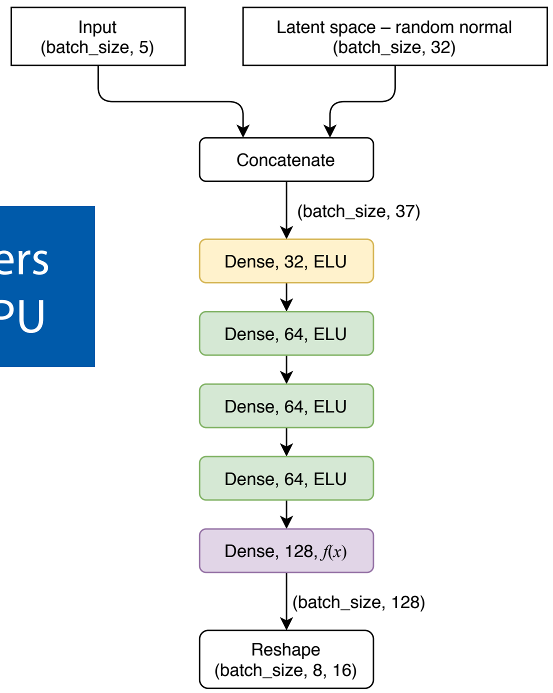
  8 pads x 16 time buckets

Model details

- Model: WGAN-GP (arXiv:1704.00028 [cs.LG])
- Generator:
  - Fully connected
  - ELU activations, custom output layer activation
  - 5 layers
- Discriminator:
  - Deep convolutional NN
  - ELU activations
  - Dropout layers
- Optimization: RMSprop, learning rate exponential decay

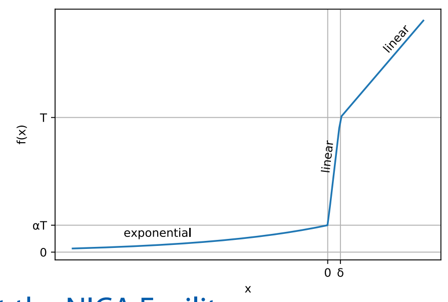
Convolutional layers are too slow on CPU

Generator

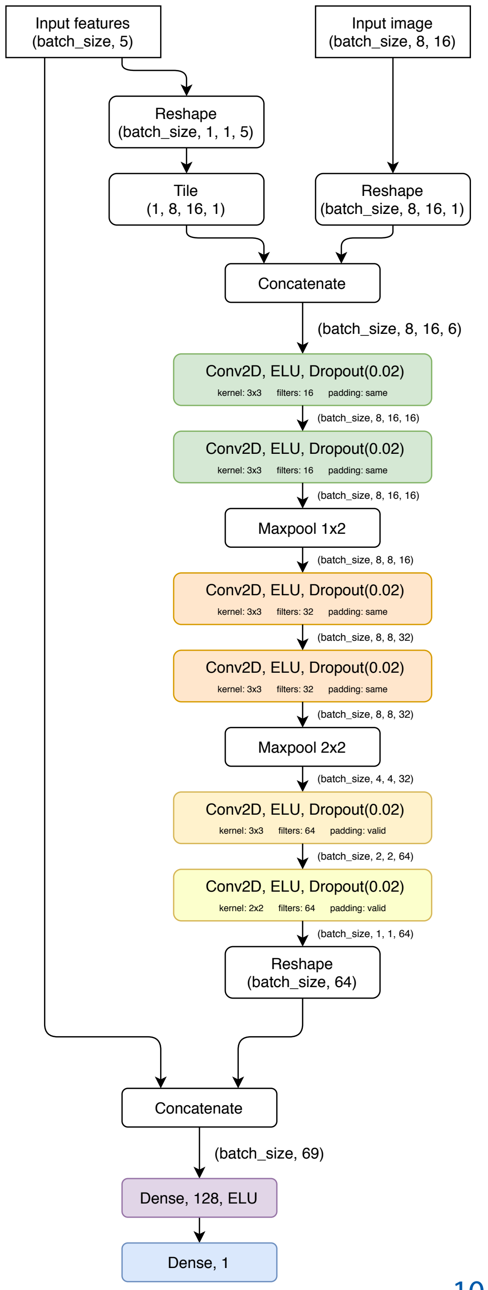


Custom activation

$$f(x) = \begin{cases} \alpha T e^{x} & x \le 0 \\ T\left(\alpha + (1-\alpha)\frac{x}{\delta}\right) & 0 \le x \le \delta \\ T - \delta + x & \delta \le x, \end{cases}$$

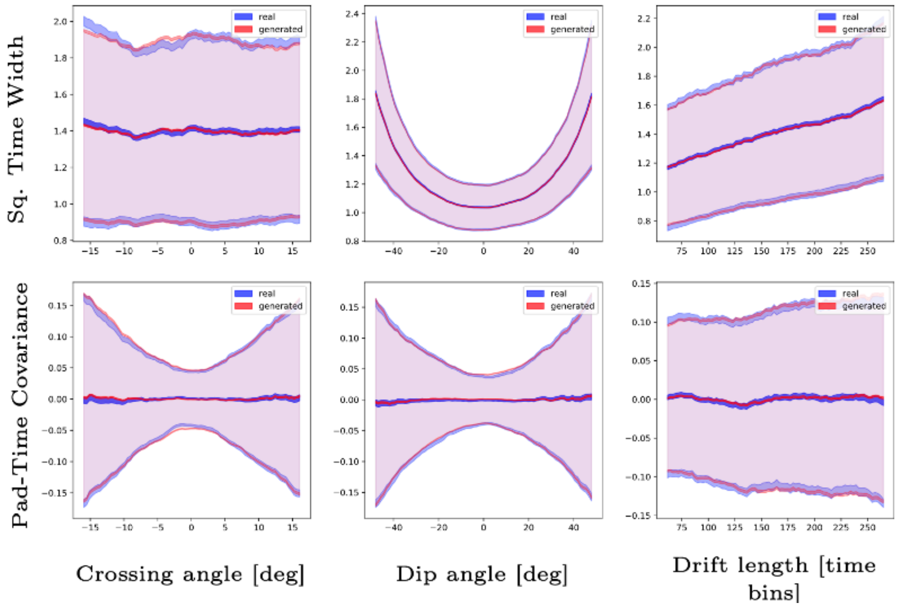


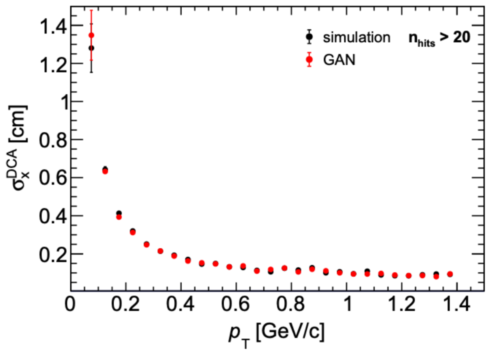
Discriminator



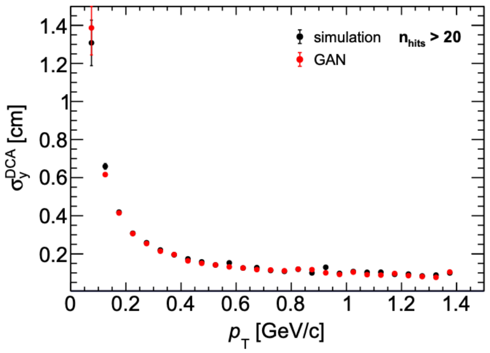
Low- and high-level validation

- Low-level validation:
  - Signal image → 1st & 2nd order moments (6 numbers)
  - Compare resulting distributions between GAN and detailed sim.
- High-level validation:
  - Model integrated into MPD software
  - Compare the reconstructed characteristics
- **x12 speedup!**
- More info in backup and in publication: Eur. Phys. J. C 81, 599 (2021) [DOI]

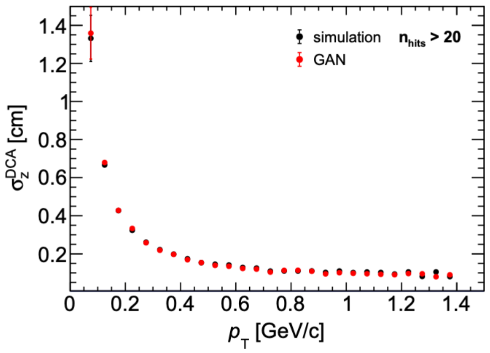




(a) Distance of closest approach resolution along *x*



(b) Distance of closest approach resolution along *y*

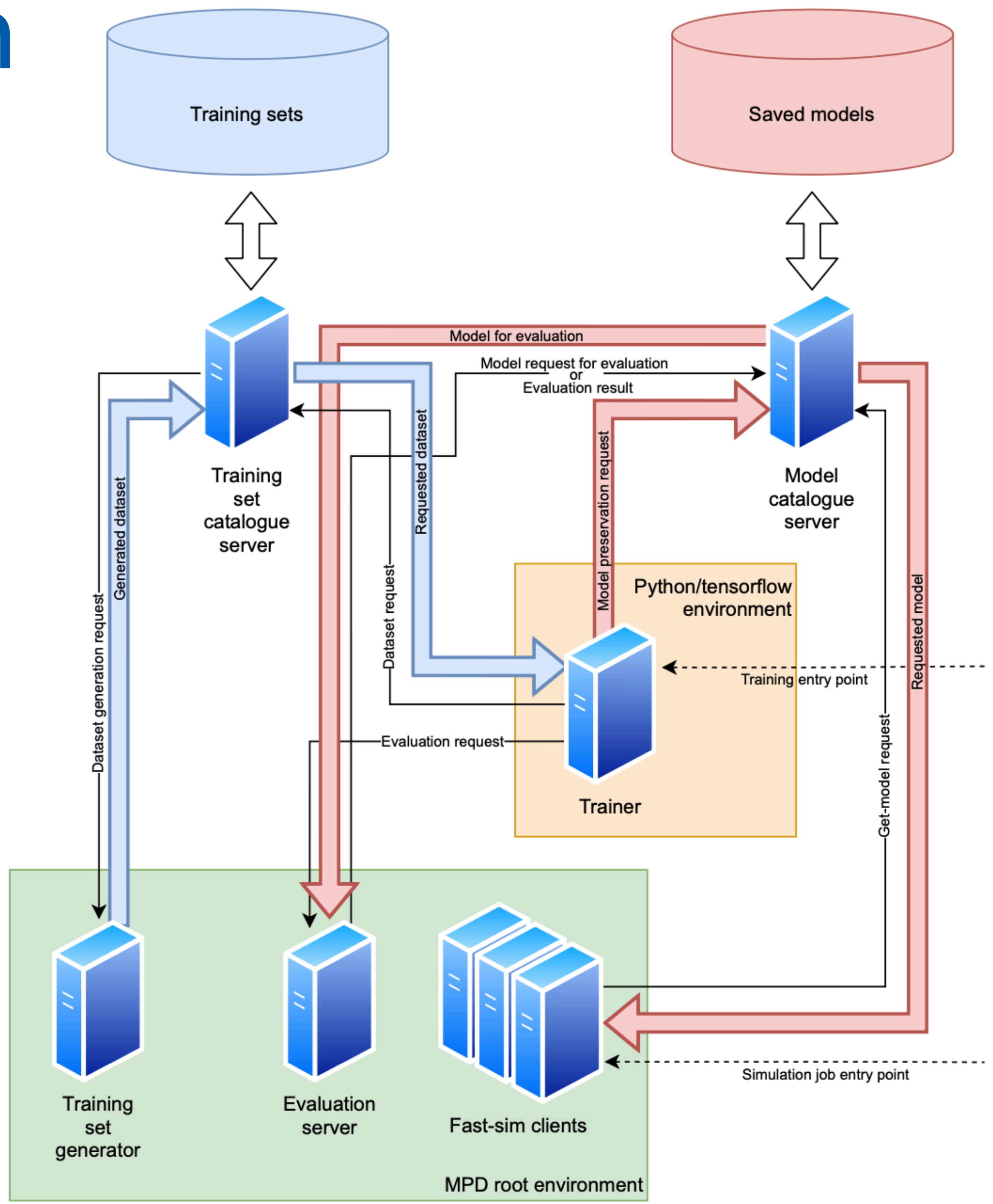


(c) Distance of closest approach resolution along *z*

Fast-sim pipeline developments

MPD production integration

- Goal: make an automated and well-documented solution
- Various simulation configurations may arise
- Thus, we need:
  - Automated training
    - Training data generation
    - Model training, evaluation and selection
  - Model library
    - Database for storage and prompt model retrieval



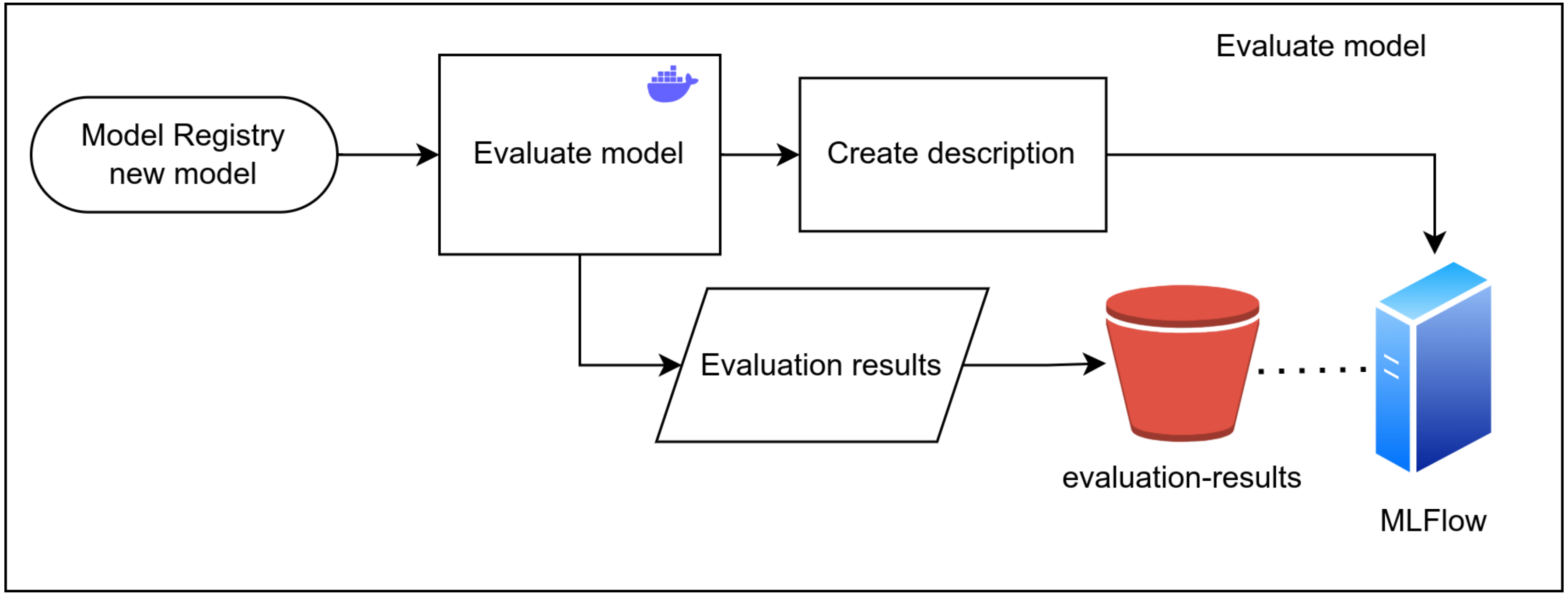
Airflow to manage the workflow

- Describe calculation pipelines as Directed Acyclic Graphs (DAGs)
- Tasks and dependencies are defined in Python
- Then scheduled and executed by Airflow

Apache Airflow

Airflow is a platform created by the community to programmatically author, schedule and monitor workflows.

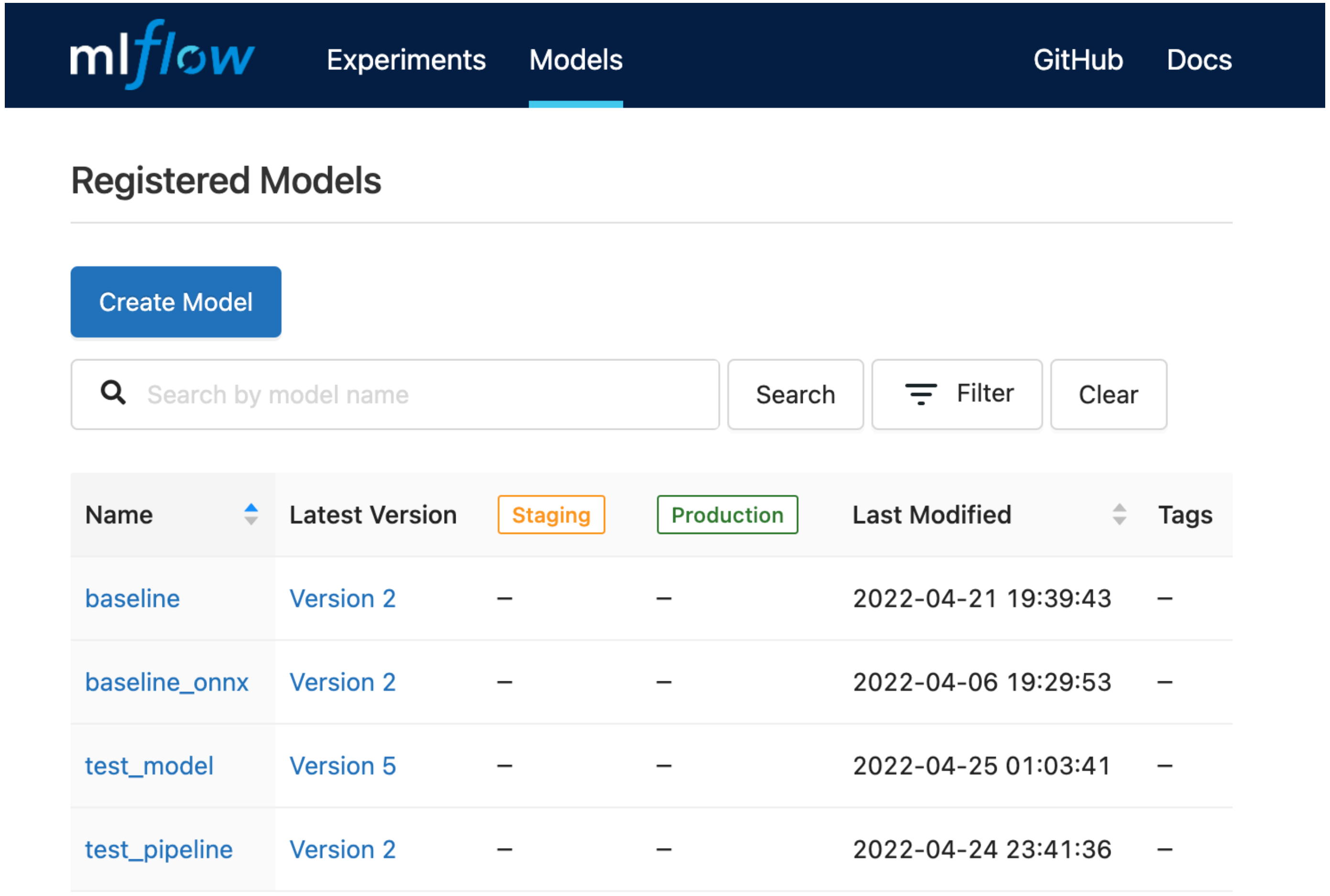
Example: model validation pipeline



- Other pipelines:
  - Dataset creation
  - Model training
  - Model conversion and upload to library

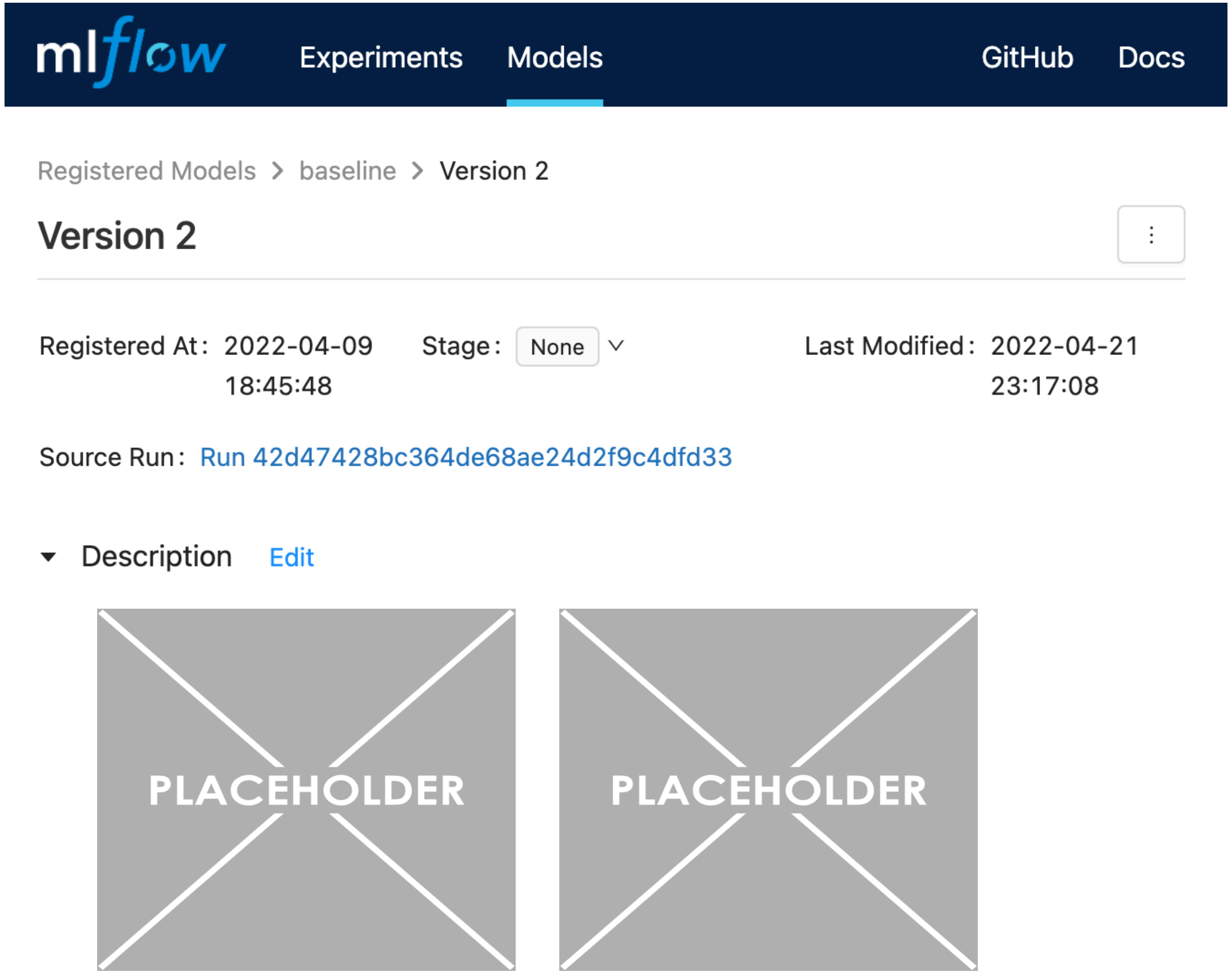
MLflow for model library

- ”An open source platform for the machine learning lifecycle”
- We use the “Model Registry” component
  - Model versioning and tagging
  - Web interface
  - REST API to download a model for inference

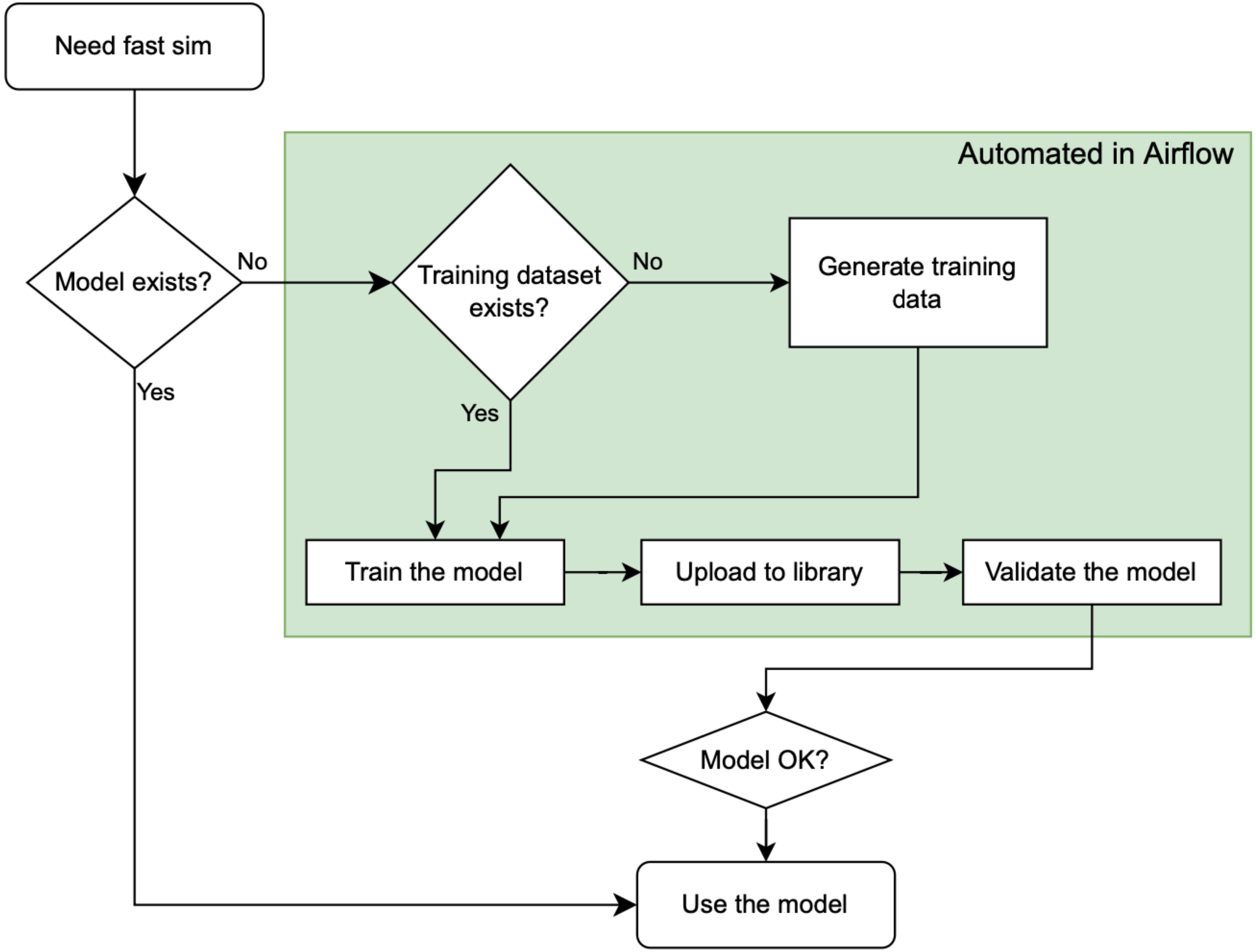


MLflow for model library

- Customizable model description
- Can be modified through API
  - e.g.: autoupload validation results from a full physics validation pipeline



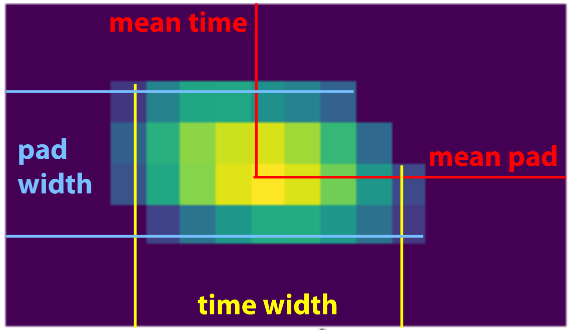
Full workflow (as we see it)

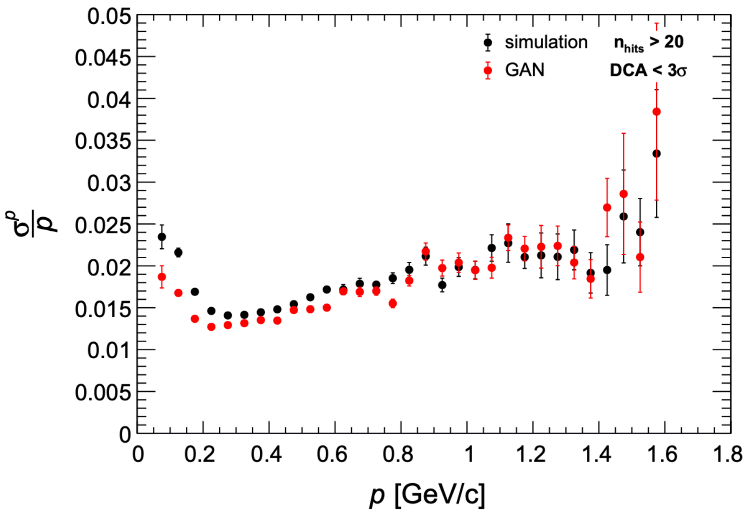


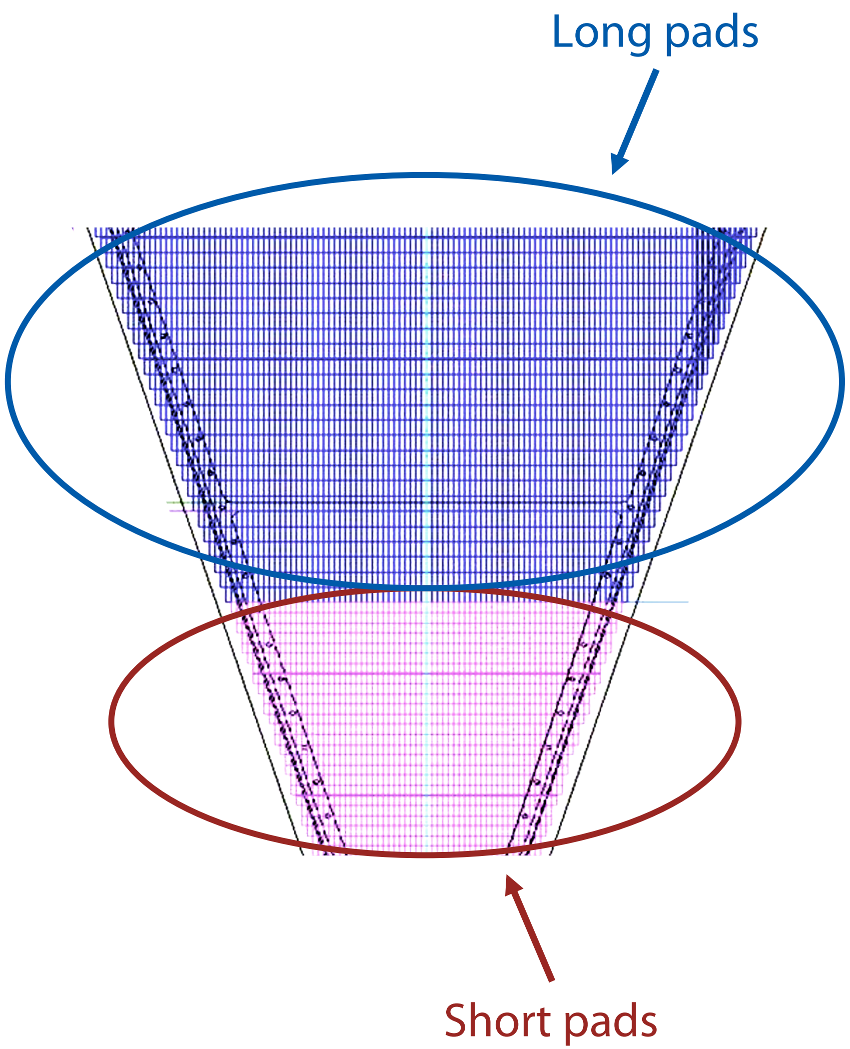
TODO: model improvements

Model improvements. #1: pad type

- Currently, we generate the responses as if they were produced by the short pads everywhere in the detector
- Pad length affects the signal shape
  - Response width ⇒ hit coordinates resolution ⇒ momentum resolution

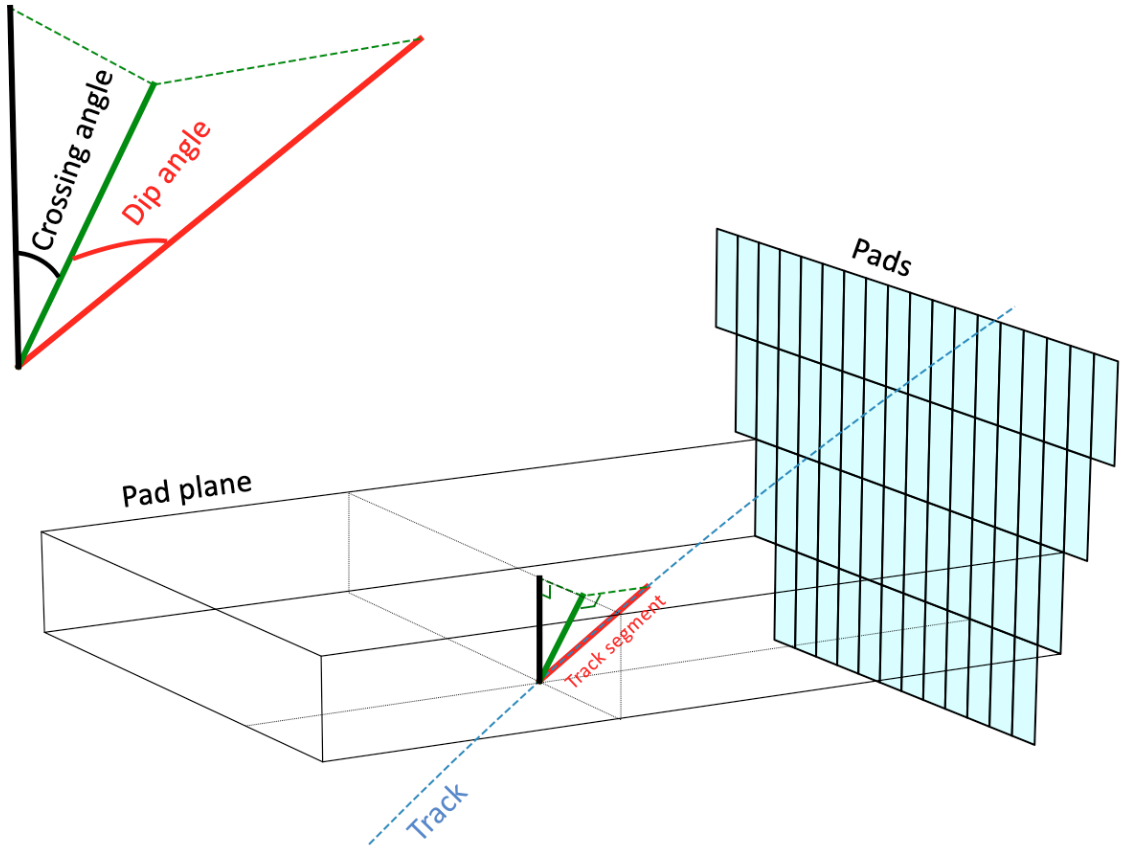






Model improvements. #2: more parameters

- Currently, we approximate the track segments as straight lines and only use their geometric characteristics
  - (coordinates of a point on the segment + 2 angles)
- By introducing the absolute momentum, we can take the segment curvature into account
  - This should be more important for long pads and for lower momenta
- Additional characteristics (e.g. particle type) may also affect the detector response

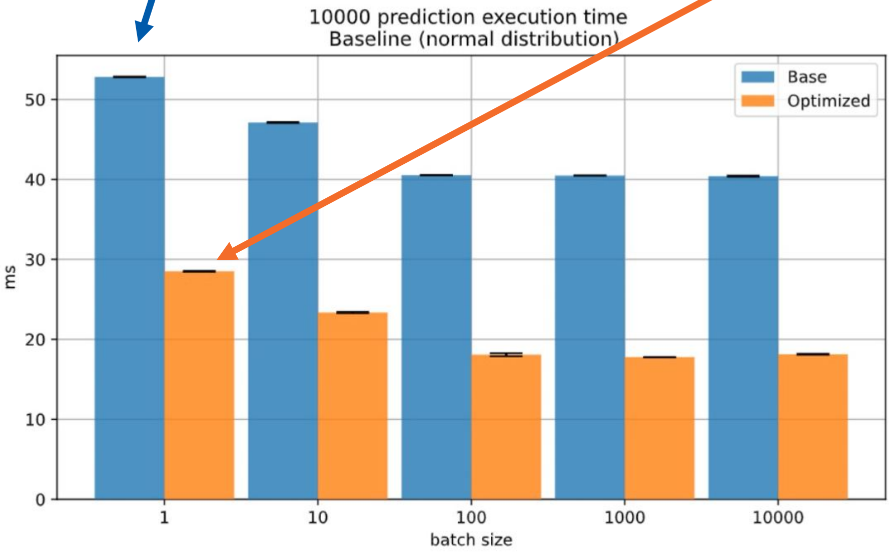


Model improvements. #3: speed/quality tradeoff

Model version from the EPJC paper

Technical improvements not affecting the quality

Model timing

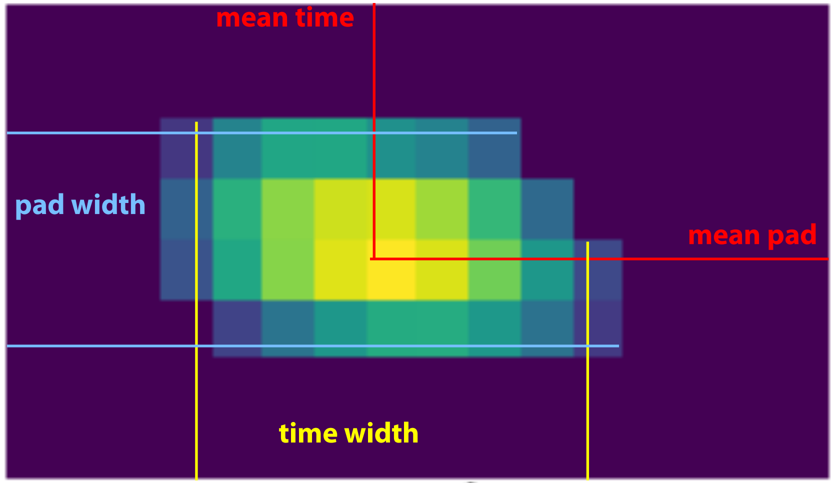


- Plenty of room for speedup
- Not tested within the MPD framework (yet)

Model quality (lower better)

Model improvements. #4: simpler model

- Our main (low-level) metrics are the 5 moments + 1 amplitude
- Why don't we just generate these 6 numbers, rather than the full image?
  - Easier to train
  - Low-level metrics optimized directly by the GAN
- Challenges:
  - Still need to build the image from these 6 numbers
  - Hard to define analytically
- Work in progress by Dmitry Evdokimov
  - Preliminary results included in our ACAT-2022 proceedings arXiv:2203.16355



Summary

- A GAN-based fast simulation model for TPC presented
  - Runs 12x faster compared to detailed simulation
- Developing an automated pipeline for integrating our model into mpdroot
  - A set of tools combined into a working prototype
- Planning further model improvements

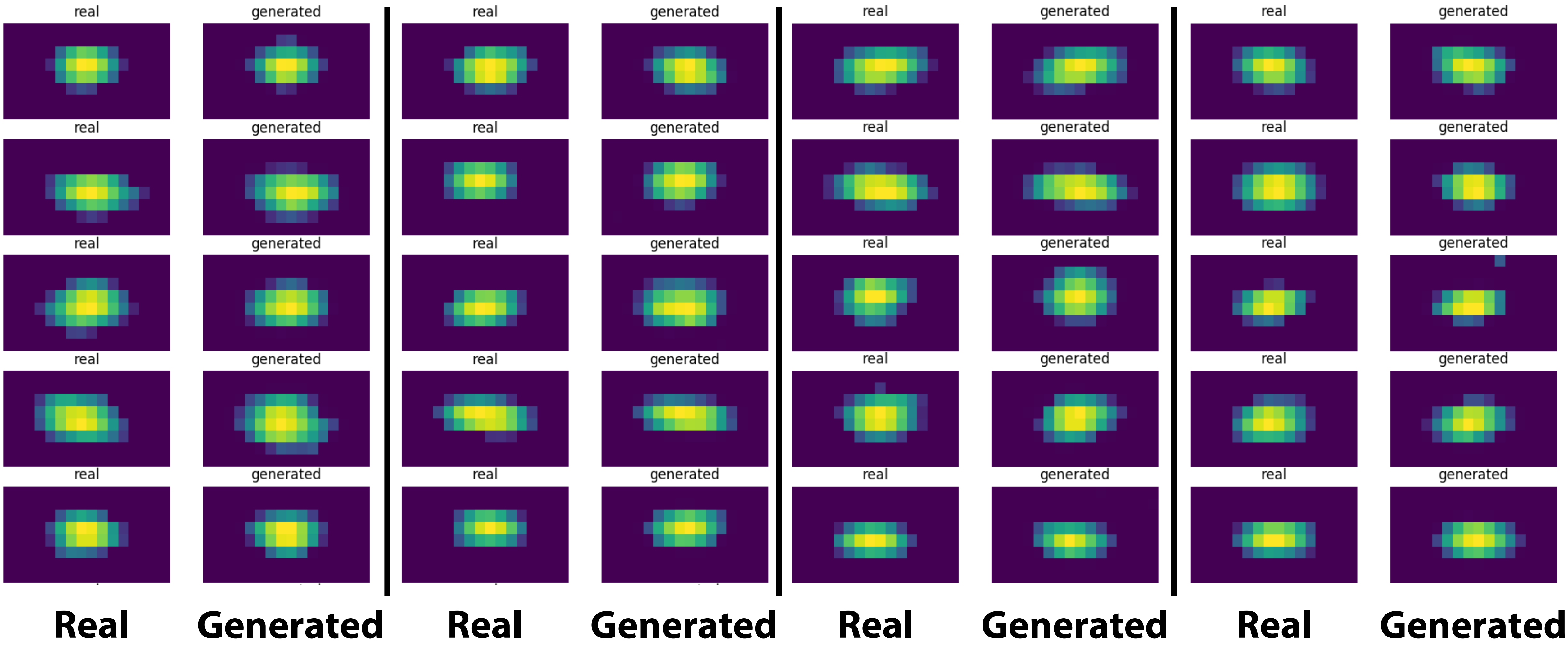
Thank you!

Backup

Training data

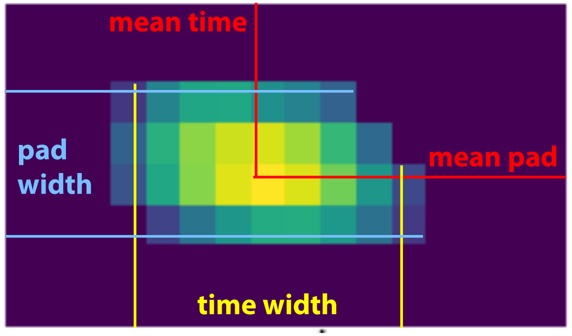
- Pion particle gun
- 20 000 pions with fixed $p_{\mathrm{T}} = 478.3$ MeV/c
- Origin point uniformly distributed along the drift path and the pad row direction
- Uniformly distributed azimuthal and polar angles

Results



Metrics

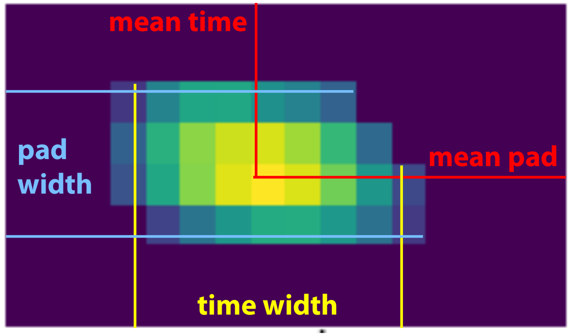
- Start with a simple preliminary metric: we compare the 1st & 2nd order moments of the signal images, i.e.:
  - the location of the signal in pads and time bins
  - the widths of the signal in pads and time bins
  - the tilt of the signal in the pad-time matrix

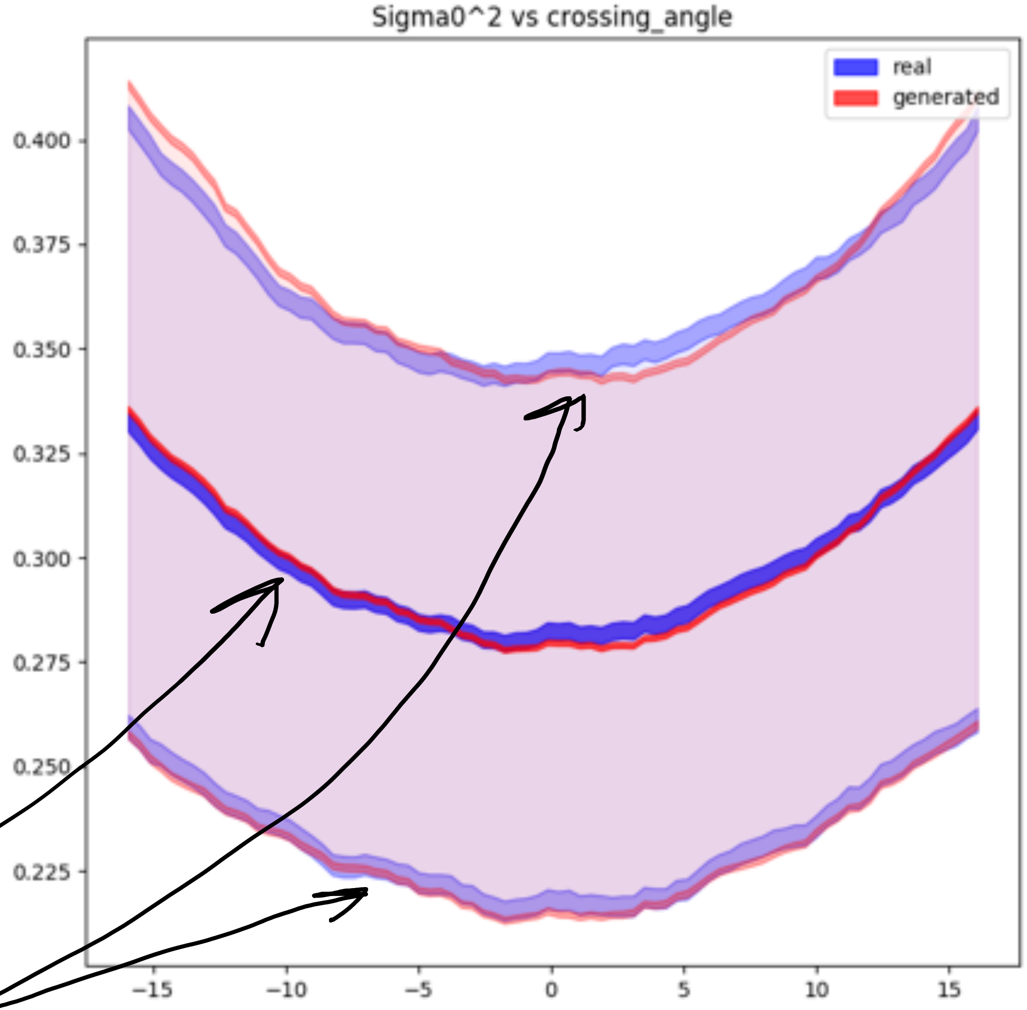


- Also looking at the integrated amplitudes
- All this as a function of track segment parameters (2 angles + 3 coordinates)

Explaining the profiles

Widths of the shaded lines correspond to the statistical uncertainties





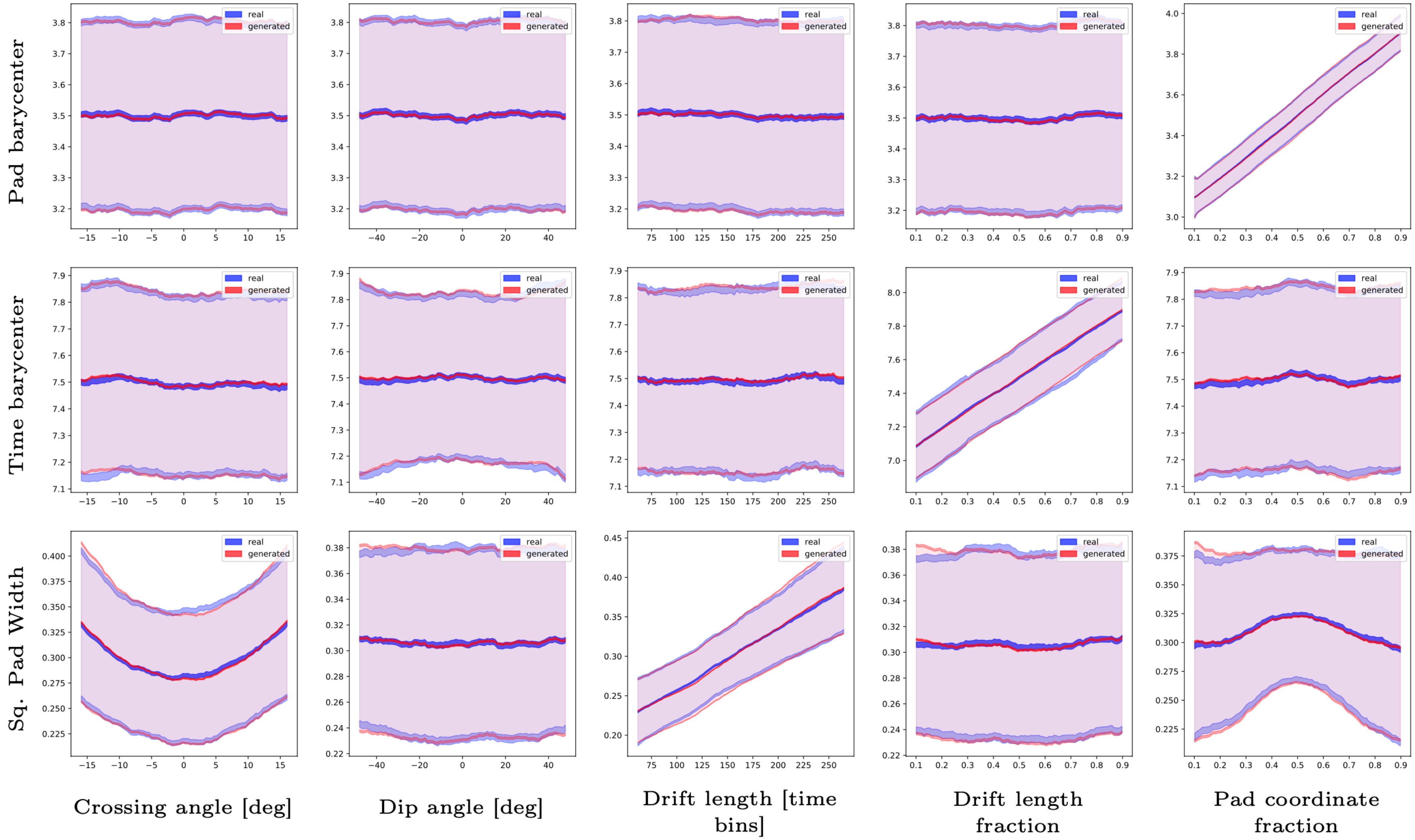
Statistic
(e.g. signal width in pads)

Input variable
(e.g. crossing angle)

Mean of the statistic

Mean ± 1 standard deviation

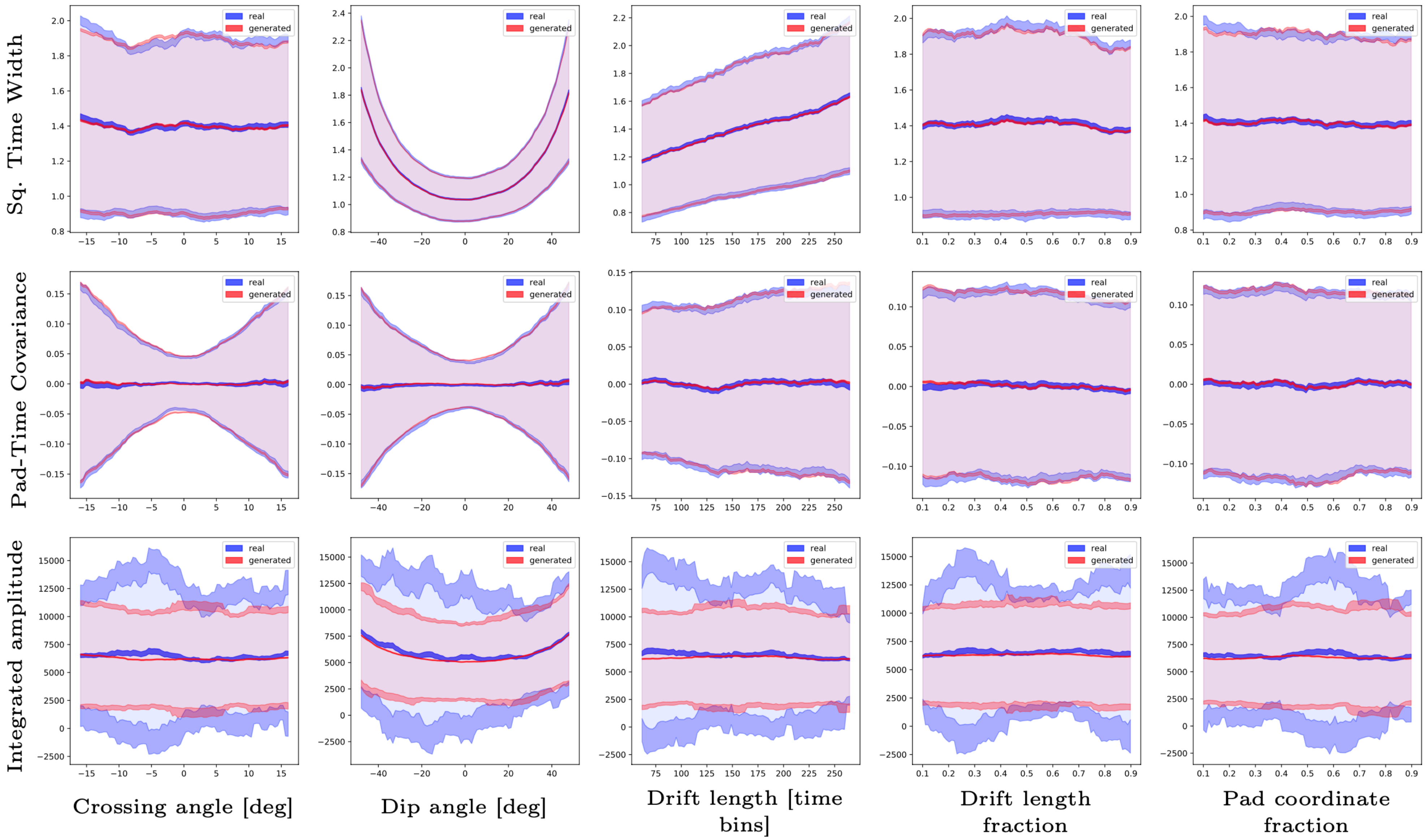
Results (profiles)



Mostly good agreement

"Real"
Generated

Results (profiles)



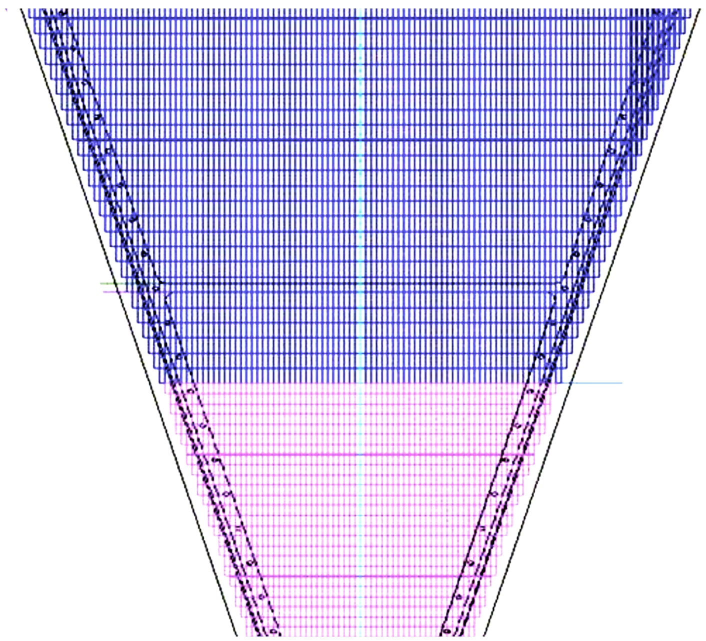
Mostly good agreement

Integrated amplitude can be factorized out and simulated separately from 1st principles

"Real"
Generated

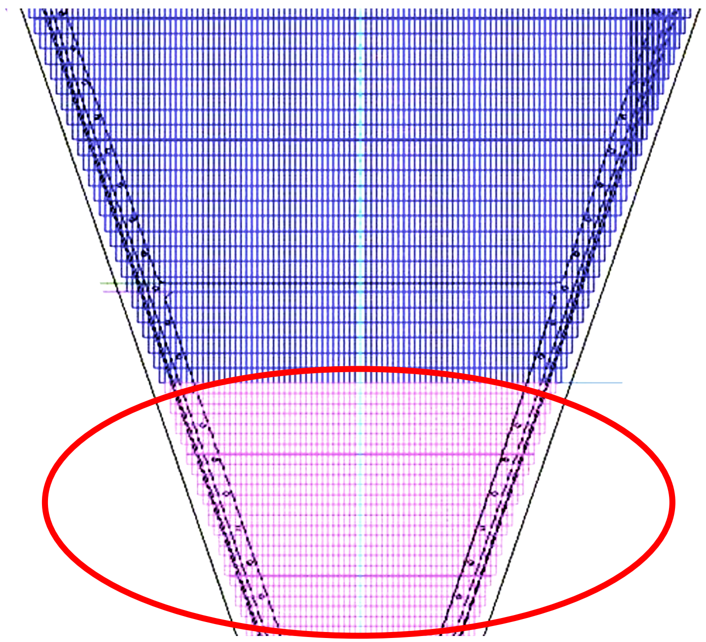
Reconstructed characteristics

- The model was integrated into the MPD software which allows to validate the reconstruction-level characteristics as well
- Estimated the speed-up to be of **x12**
  - Measured on a single core of an Intel Core i7-3770K (3.50GHz) CPU



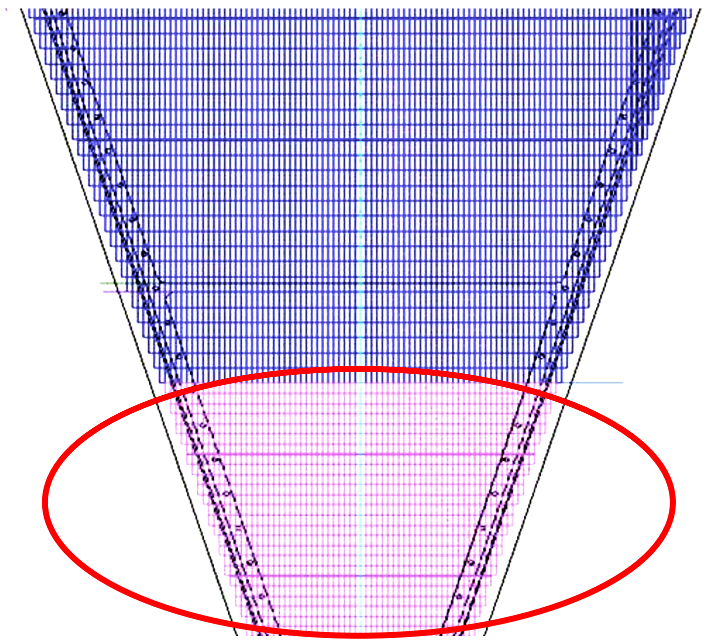
Reconstructed characteristics

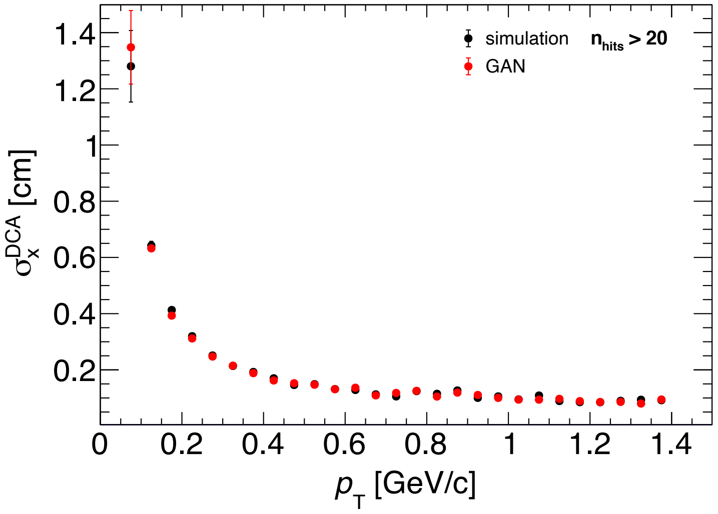
- The model was integrated into the MPD software which allows to validate the reconstruction-level characteristics as well
- Estimated the speed-up to be of **x12**
  - Measured on a single core of an Intel Core i7-3770K (3.50GHz) CPU
- Note: the model was only trained on the responses from the **short pads**, while applied for the whole TPC



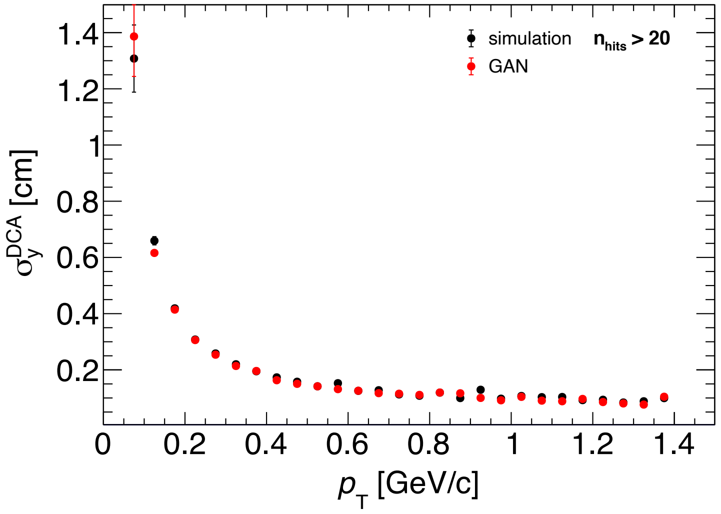
Reconstructed characteristics

- The model was integrated into the MPD software which allows to validate the reconstruction-level characteristics as well
- Estimated the speed-up to be of **x12**
  - Measured on a single core of an Intel Core i7-3770K (3.50GHz) CPU
- Note: the model was only trained on the responses from the **short pads**, while applied for the whole TPC
- Simulated central Au+Au collisions at $\sqrt{s_{NN}} = 9$ GeV
- Comparison made on pions with $|y| < 0.5, n_{hits} \geq 20$

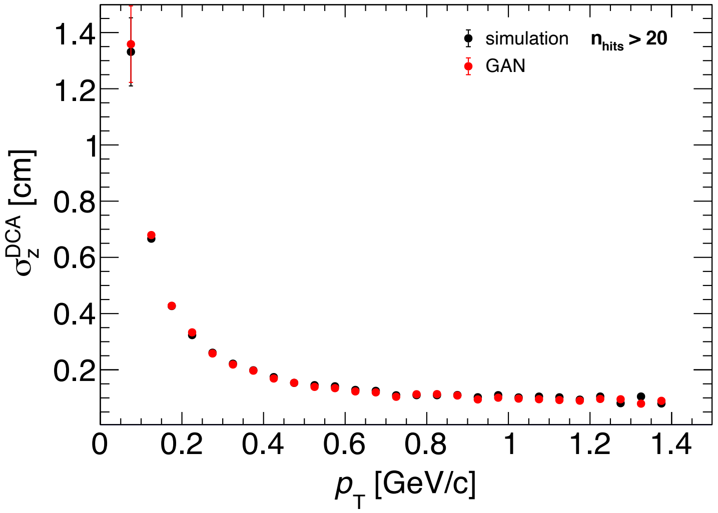




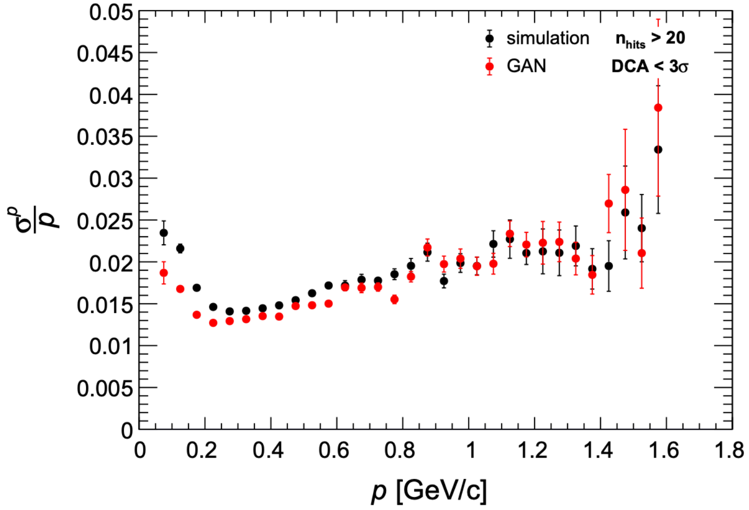
(a) Distance of closest approach resolution along x



(b) Distance of closest approach resolution along y

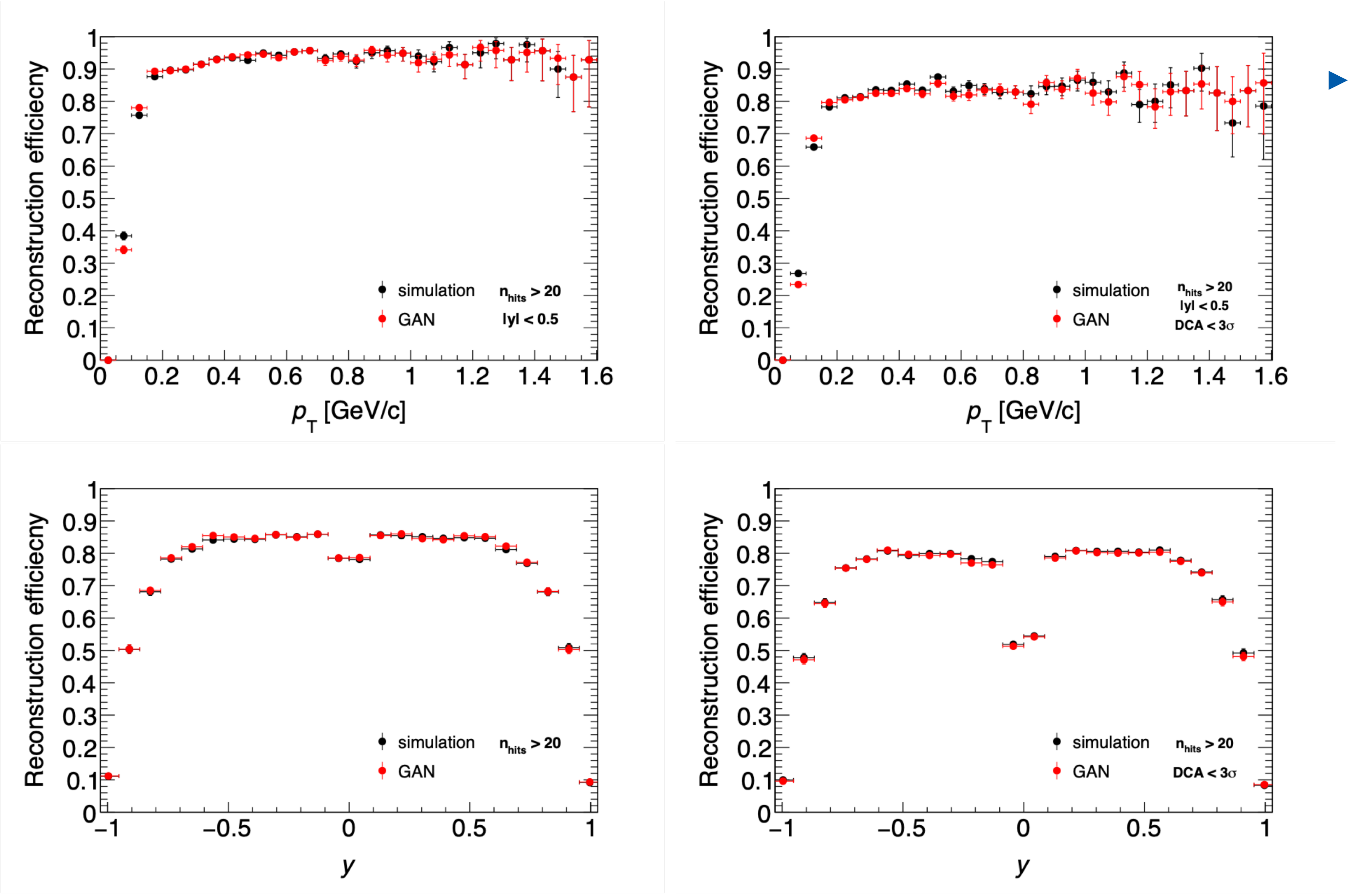


(c) Distance of closest approach resolution along z

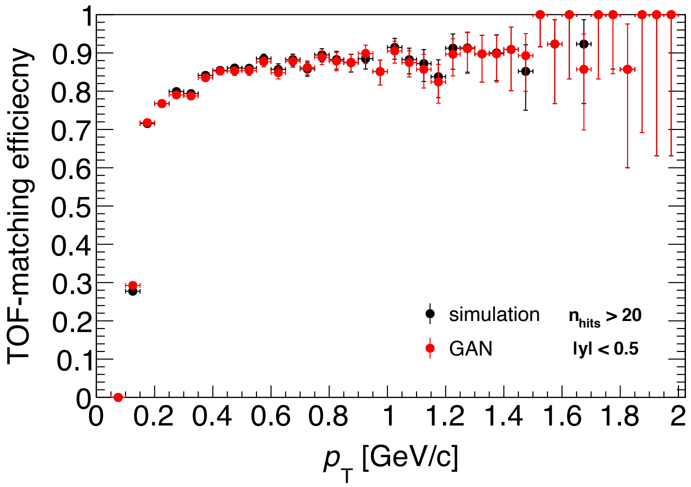


(d) Momentum resolution

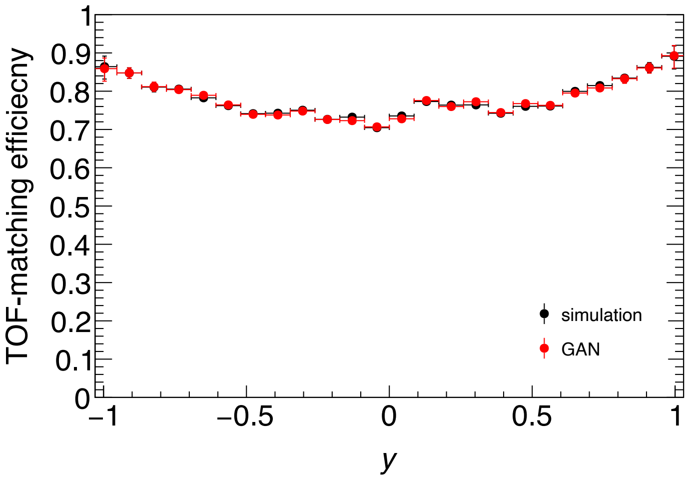
- DCA resolution well reproduced
- Momentum resolution overestimated
  - as one would expect with short pads everywhere



- Reasonable agreement for the reconstruction efficiencies

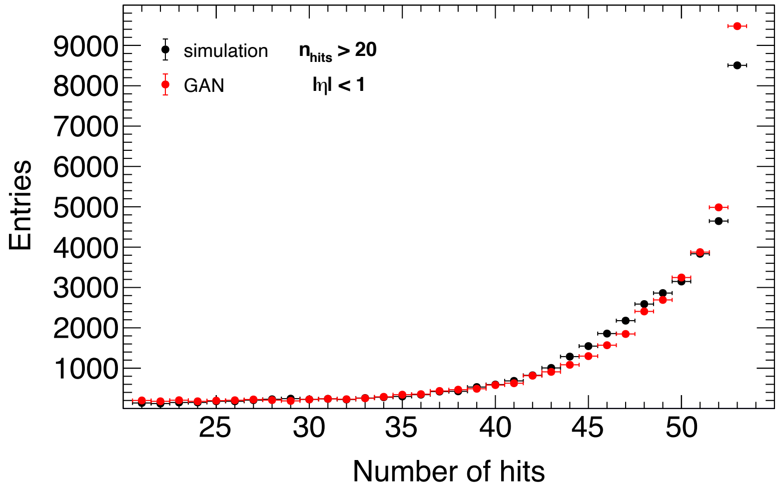


(a) TOF matching efficiency as a function of the transverse momentum



(b) TOF matching efficiency as a function of the rapidity

- Good agreement in the TOF matching efficiencies
- Overestimated number of hits per track
  - again, as one would expect with short pads everywhere



(c) Distribution of the number of hits per track

Δx for short vs long pads

- GAN predicts similar Δx for short and long pads
- Detailed simulation shows they should be different

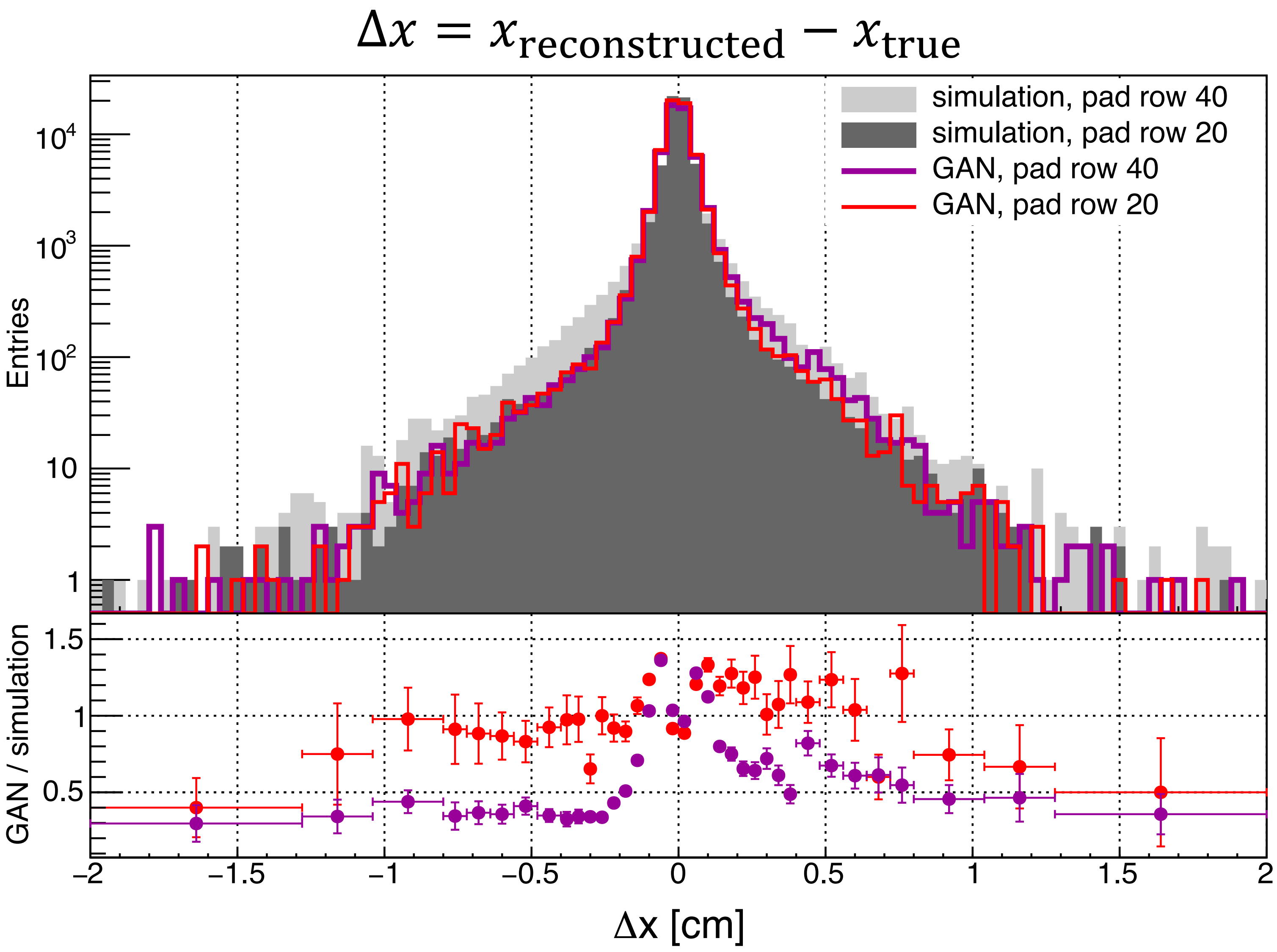


Fig. 7. Distributions of differences $\Delta x = x_{\text{reconstructed}} - x_{\text{true}}$ between the reconstructed and true cluster coordinates along the pad row direction. For the short (long) pads from the pad row 20 (40), the detailed simulation results are shown in the dark (light) gray shaded histogram, while the histogram for the GAN prediction is shown with the red (magenta) line. The ratios between the GAN and detailed simulation yields in the same pad rows are shown in the bottom part of the plot.