Fast simulation of Time Projection Chamber response at MPD using GANs

MPD Physics Forum,

11 February 2021

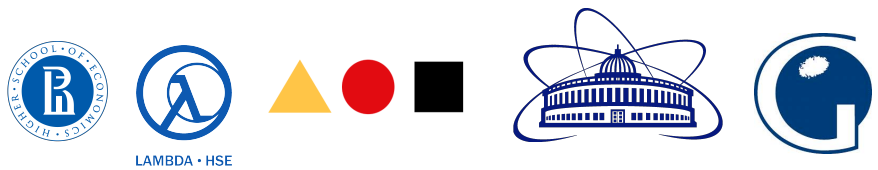
Artem Maevskiy[1], Fedor Ratnikov[1,2], Alexander Zinchenko[3], Victor Riabov[4]

[1]HSE University (National Research University Higher School of Economics)

[2]Yandex School of Data Analysis

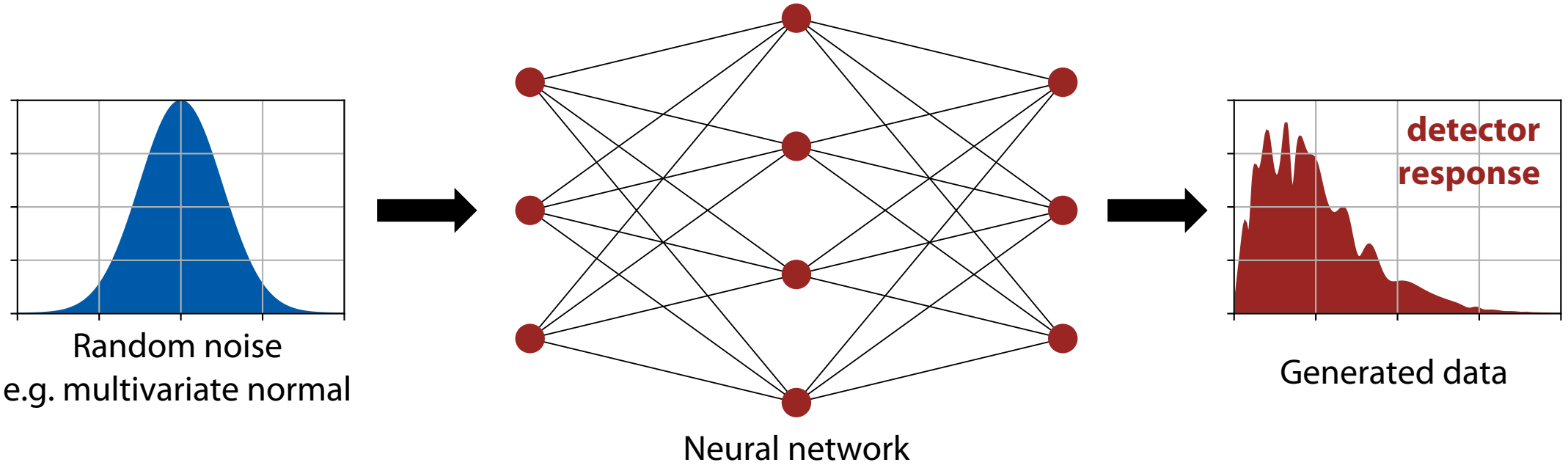
[3]Joint Institute for Nuclear Research

[4]Petersburg Nuclear Physics Institute



Generative Adversarial Networks and 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

GANs for fast simulation

- Quite a developing field!
- GANs interpolate the available training data

[8] A. Maevskiy *et al.* [LHCb Collaboration], "Fast Data-Driven Simulation of Cherenkov Detectors Using Generative Adversarial Networks," arXiv:1905.11825 [physics.ins-det].

[9] D. Belayneh *et al.*, "Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics," arXiv:1912.06794 [physics.ins-det].

[10] J. R. Vlimant, F. Pantaleo, M. Pierini, V. Loncar, S. Vallecorsa, D. Anderson, T. Nguyen and A. Zlokapa, "Large-Scale Distributed Training Applied to Generative Adversarial Networks for Calorimeter Simulation," EPJ Web Conf. **214**, 06025 (2019) doi:10.1051/epjconf/201921406025.

[11] D. Lancierini, P. Owen and N. Serra, "Simulating the LHCb hadron calorimeter with generative adversarial networks," Nuovo Cim. C **42**, no. 4, 197 (2019) doi:10.1393/ncc/i2019-19197-3.

[12] L. de Oliveira, M. Paganini and B. Nachman, "Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis," Comput. Softw. Big Sci. **1**, no. 1, 4 (2017) doi:10.1007/s41781-017-0004-6 [arXiv:1701.05927 [stat.ML]].

[13] S. Carrazza and F. A. Dreyer, "Lund jet images from generative and cycle-consistent adversarial networks," Eur. Phys. J. C **79**, no. 11, 979 (2019) doi:10.1140/epjc/s10052-019-7501-1 [arXiv:1909.01359 [hep-ph]].

[14] M. Paganini, L. de Oliveira and B. Nachman, "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters," Phys. Rev. Lett. **120**, no. 4, 042003 (2018) doi:10.1103/PhysRevLett.120.042003 [arXiv:1705.02355 [hep-ex]].

[15] L. de Oliveira, M. Paganini and B. Nachman, "Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters," J. Phys. Conf. Ser. **1085**, no. 4, 042017 (2018) doi:10.1088/1742-6596/1085/4/042017 [arXiv:1711.08813 [hep-ex]].

[16] M. Paganini, L. de Oliveira and B. Nachman, "CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks," Phys. Rev. D **97**, no. 1, 014021 (2018) doi:10.1103/PhysRevD.97.014021 [arXiv:1712.10321 [hep-ex]].

[17] F. Carminati, A. Gheata, G. Khattak, P. Mendez Lorenzo, S. Sharan and S. Vallecorsa, "Three dimensional Generative Adversarial Networks for fast simulation," J. Phys. Conf. Ser. **1085**, no. 3, 032016 (2018) doi:10.1088/1742-6596/1085/3/032016.

[18] M. Erdmann, L. Geiger, J. Glombitza and D. Schmidt, "Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks," Comput. Softw. Big Sci. **2**, no. 1, 4 (2018) doi:10.1007/s41781-018-0008-x [arXiv:1802.03325 [astro-ph.IM]].

[19] P. Musella and F. Pandolfi, "Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks," Comput. Softw. Big Sci. **2**, no. 1, 8 (2018) doi:10.1007/s41781-018-0015-y [arXiv:1805.00850 [hep-ex]].

[20] M. Erdmann, J. Glombitza and T. Quast, "Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network," Comput. Softw. Big Sci. **3**, no. 1, 4 (2019) doi:10.1007/s41781-018-0019-7 [arXiv:1807.01954 [physics.ins-det]].

[21] S. Vallecorsa, F. Carminati and G. Khattak, "3D convolutional GAN for fast simulation," EPJ Web Conf. **214**, 02010 (2019) doi:10.1051/epjconf/201921402010.

[22] S. Otten, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen, D. Podareanu, R. R. de Austri and R. Verheyen, "Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer," arXiv:1901.00875 [hep-ph].

[23] A. Butter, T. Plehn and R. Winterhalder, "How to GAN LHC Events," SciPost Phys. **7**, no. 6, 075 (2019) doi:10.21468/SciPostPhys.7.6.075 [arXiv:1907.03764 [hep-ph]].

[24] C. Ahdida *et al.* [SHiP Collaboration], "Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks," JINST **14**, P11028 (2019) doi:10.1088/1748-0221/14/11/P11028 [arXiv:1909.04451 [physics.ins-det]].

[25] S. Farrell, W. Bhimji, T. Kurth, M. Mustafa, D. Bard, Z. Lukic, B. Nachman and H. Patton, "Next Generation Generative Neural Networks for HEP," EPJ Web Conf. **214**, 09005 (2019) doi:10.1051/epjconf/201921409005.

[26] J. Arjona Martínez, T. Q. Nguyen, M. Pierini, M. Spiropulu and J. R. Vlimant, "Particle Generative Adversarial Networks for full-event simulation at the LHC and their application to pileup description," arXiv:1912.02748 [hep-ex].

[27] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini, "LHC analysis-specific datasets with Generative Adversarial Networks," arXiv:1901.05282 [hep-ex].

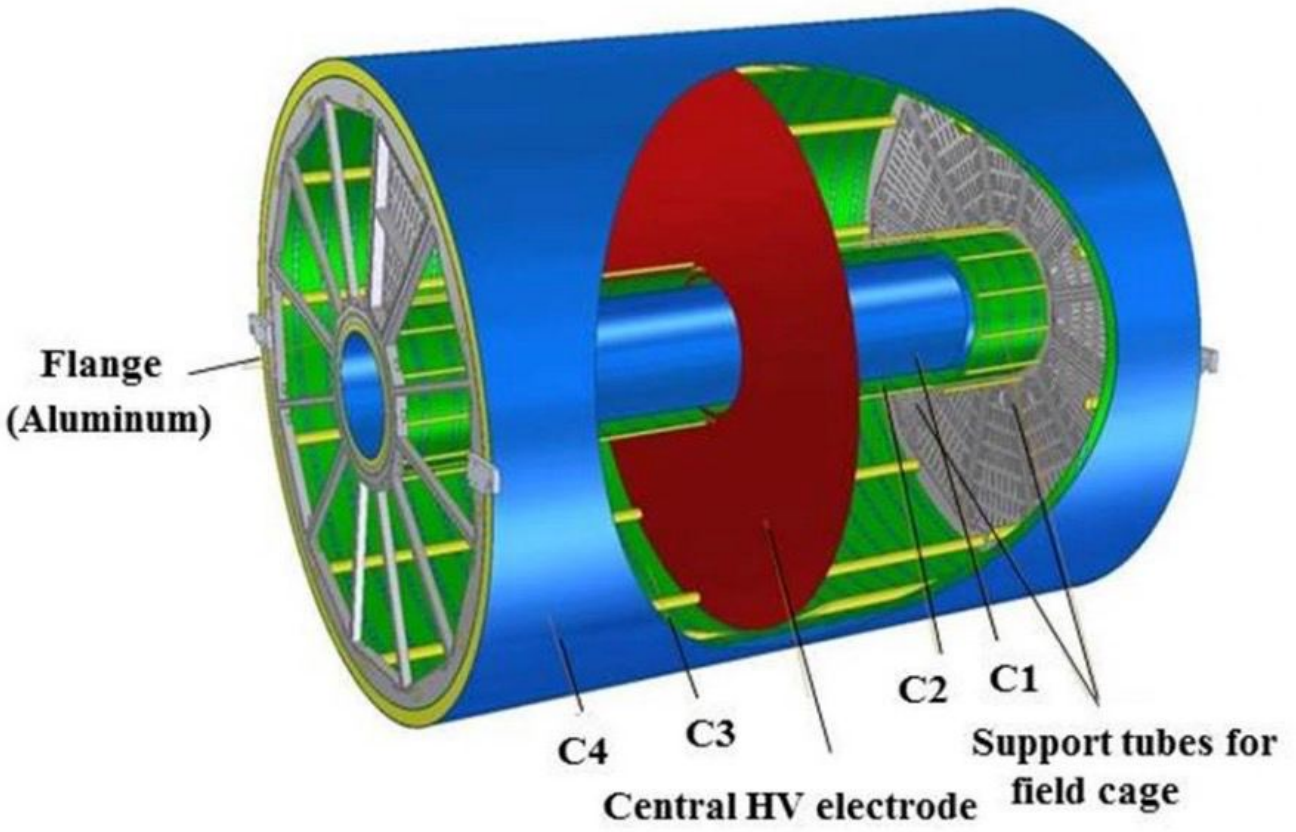
[28] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, "A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC," PoS LeptonPhoton **2019**, 050 (2019) doi:10.22323/1.367.0050.

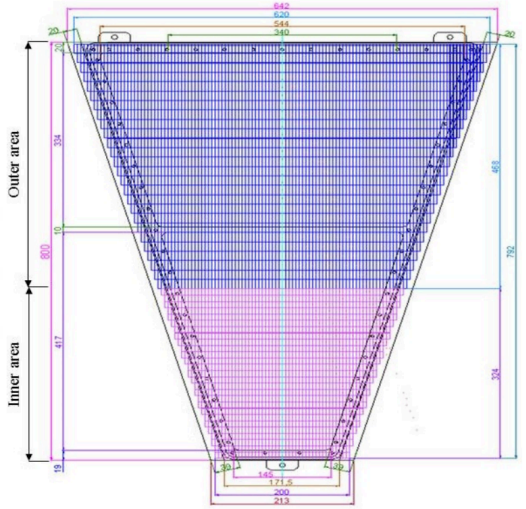
[29] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, "DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC," JHEP **1908**, 110 (2020) doi:10.1007/JHEP08(2019)110 [arXiv:1903.02433 [hep-ex]].

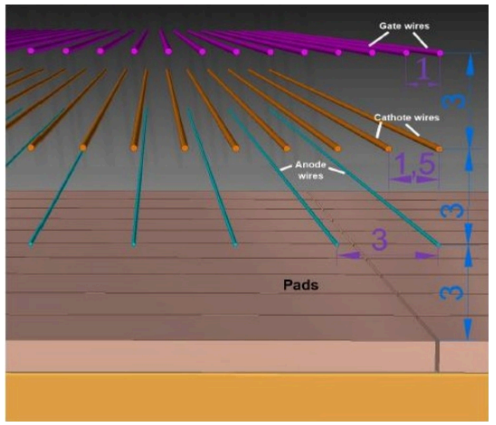
[30] Y. Alanazi, N. Sato, T. Liu, W. Melnitchouk, M. P. Kuchera, E. Pritchard, M. Robertson, R. Strauss, L. Velasco and Y. Li, "Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN)," arXiv:2001.11103 [hep-ph].

K. Matchev, P. Shyamsundar, Uncertainties associated with GAN-generated datasets in high energy physics, arXiv:2002.06307 [hep-ph]

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

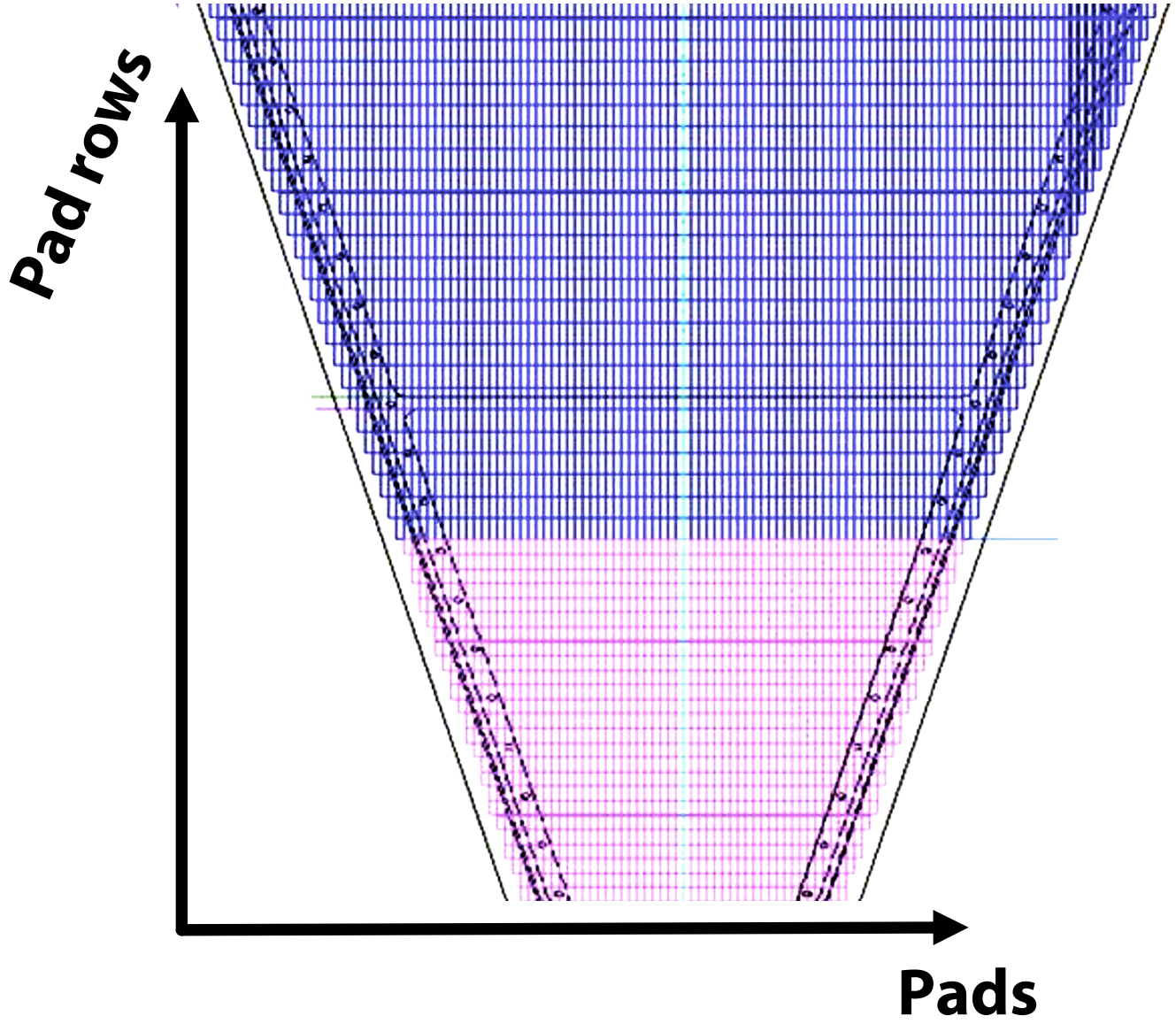
http://mpd.jinr.ru/wp-content/uploads/2019/01/TpcTdr-v07.pdf

Our approach to fast simulating TPC

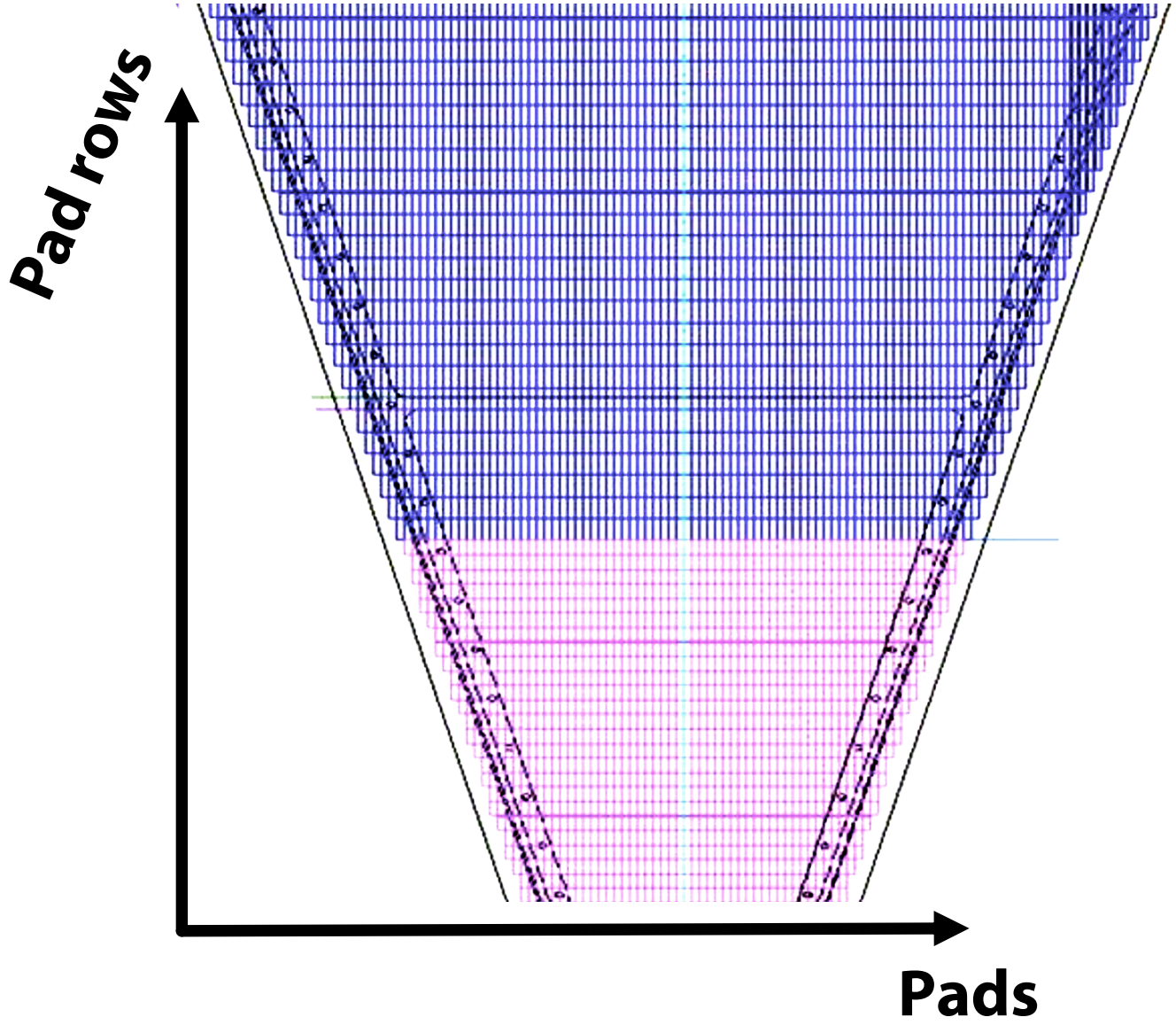
Objective

- For each event need to generate the signal for:
  - $95\,232 \cdot 310$ elements (pads x time buckets)
  - Conditioned on the track parameters for the whole event
- Very large output space
- Input of varying dimensionality

- Need to simplify somehow!

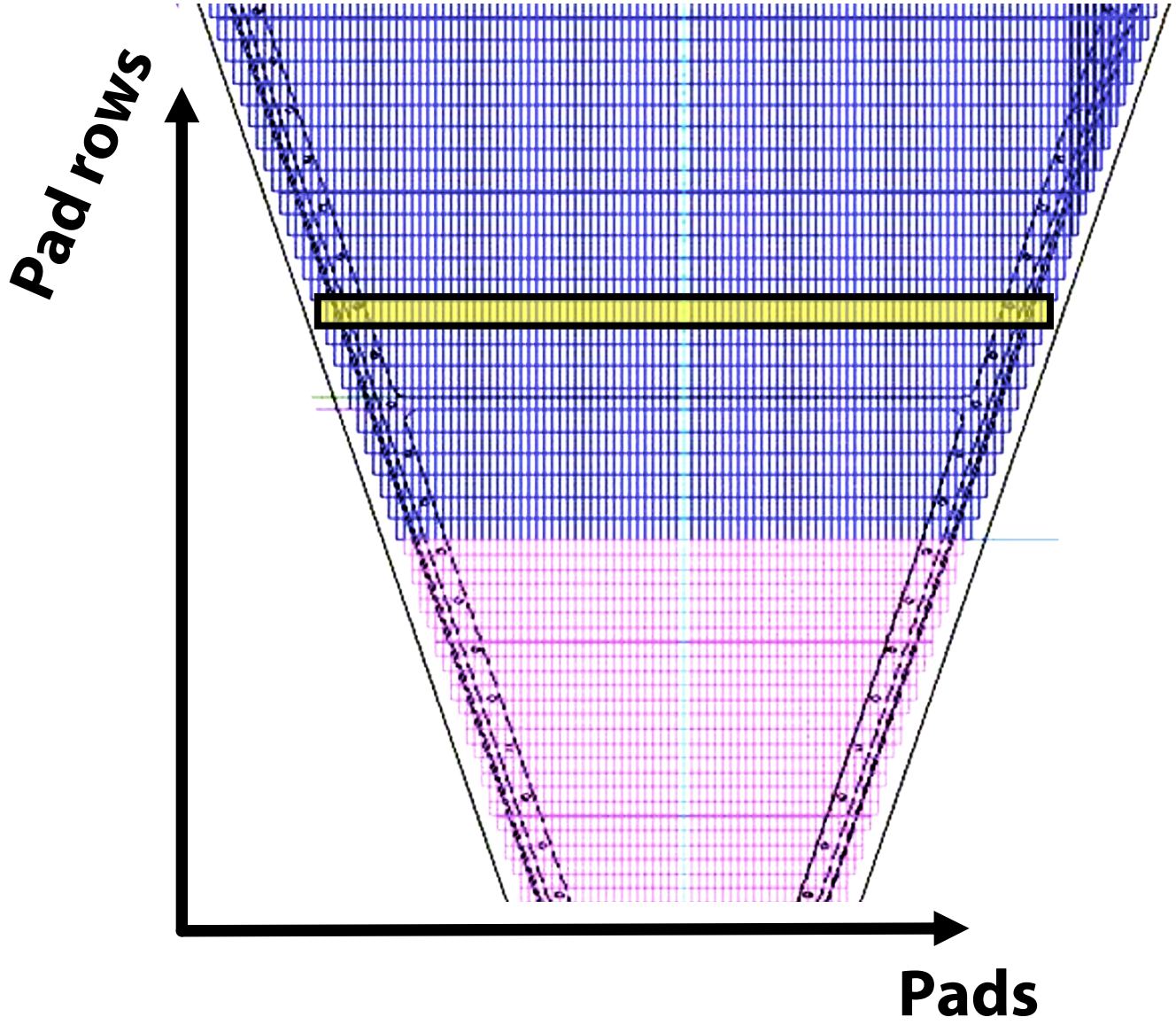


Assumptions for fast simulation



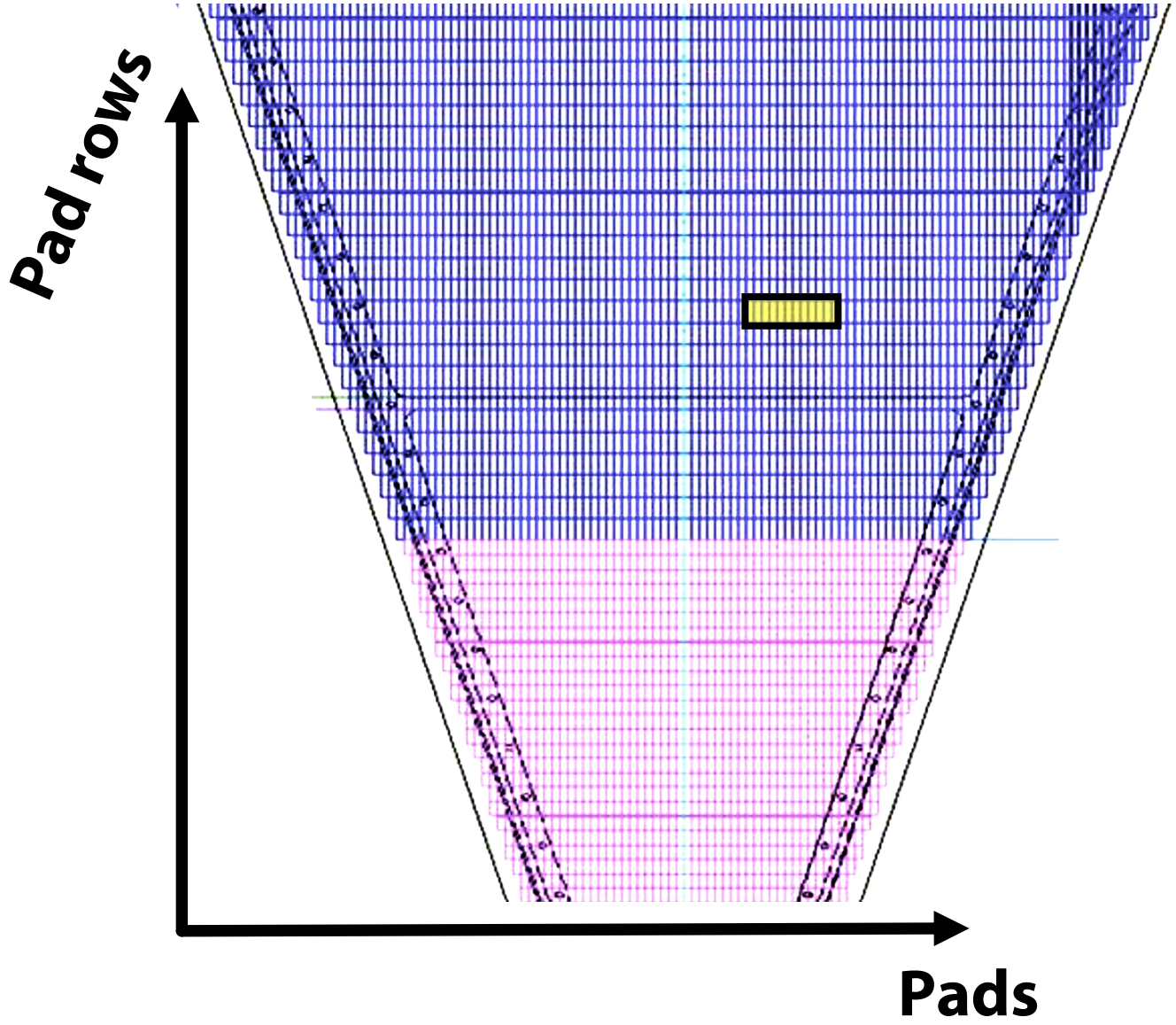
Assumptions for fast simulation

- Factorizing the pad rows
  - dividing tracks to segments, each contributing to a particular pad row
  - can model such contributions **independently**!



Assumptions for fast simulation

- Factorizing the pad rows
  - dividing tracks to segments, each contributing to a particular pad row
  - can model such contributions **independently**!
- Signal localization (both position & time)
  - model only a **small area** instead of the full row
  - model only a **few time buckets**

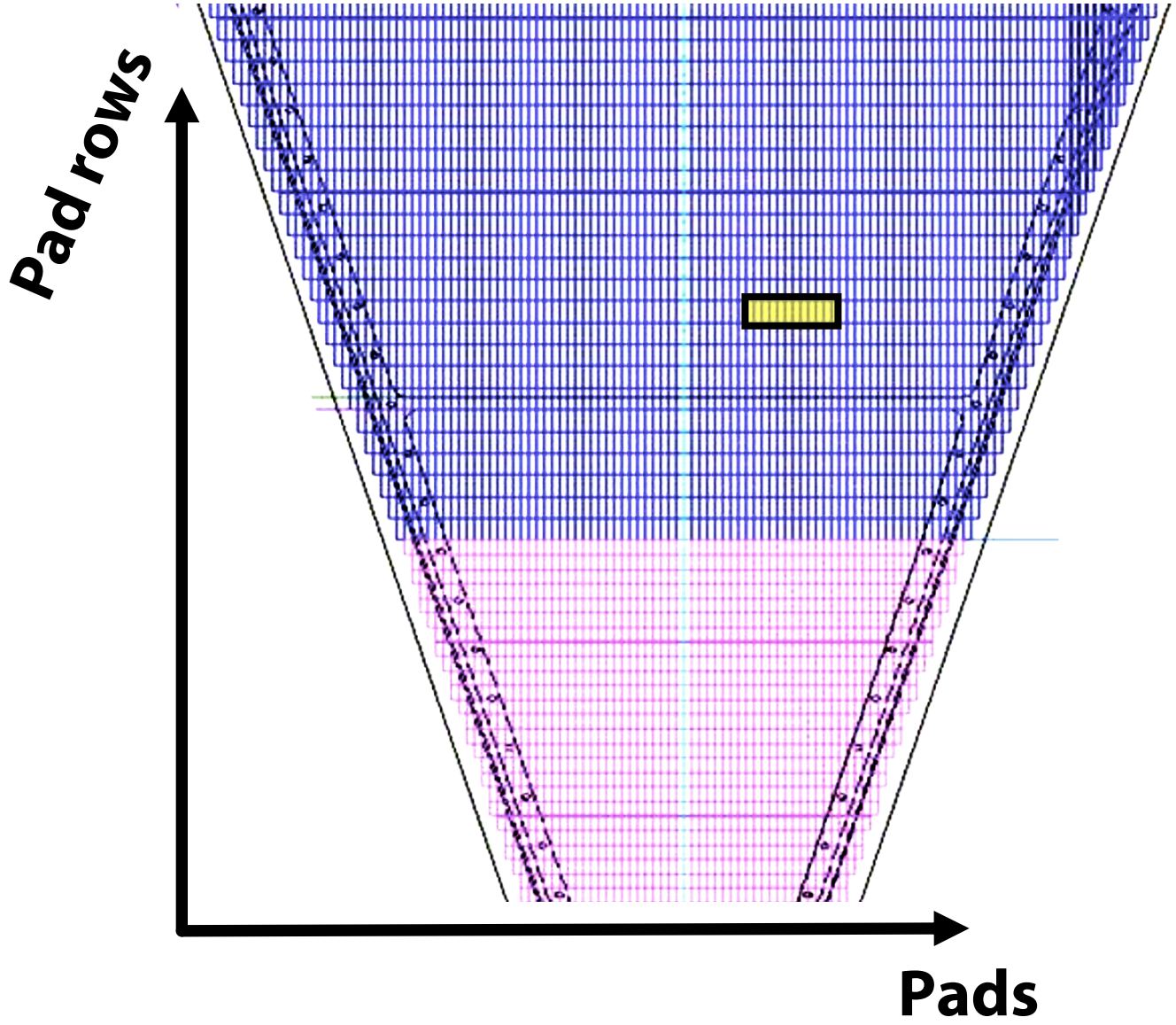


Assumptions for fast simulation

- Factorizing the pad rows
  - dividing tracks to segments, each contributing to a particular pad row
  - can model such contributions **independently**!
- Signal localization (both position & time)
  - model only a **small area** instead of the full row
  - model only a **few time buckets**
- Target dimensionality:

  8 pads x 16 time buckets

  (instead of original 95 232·310)

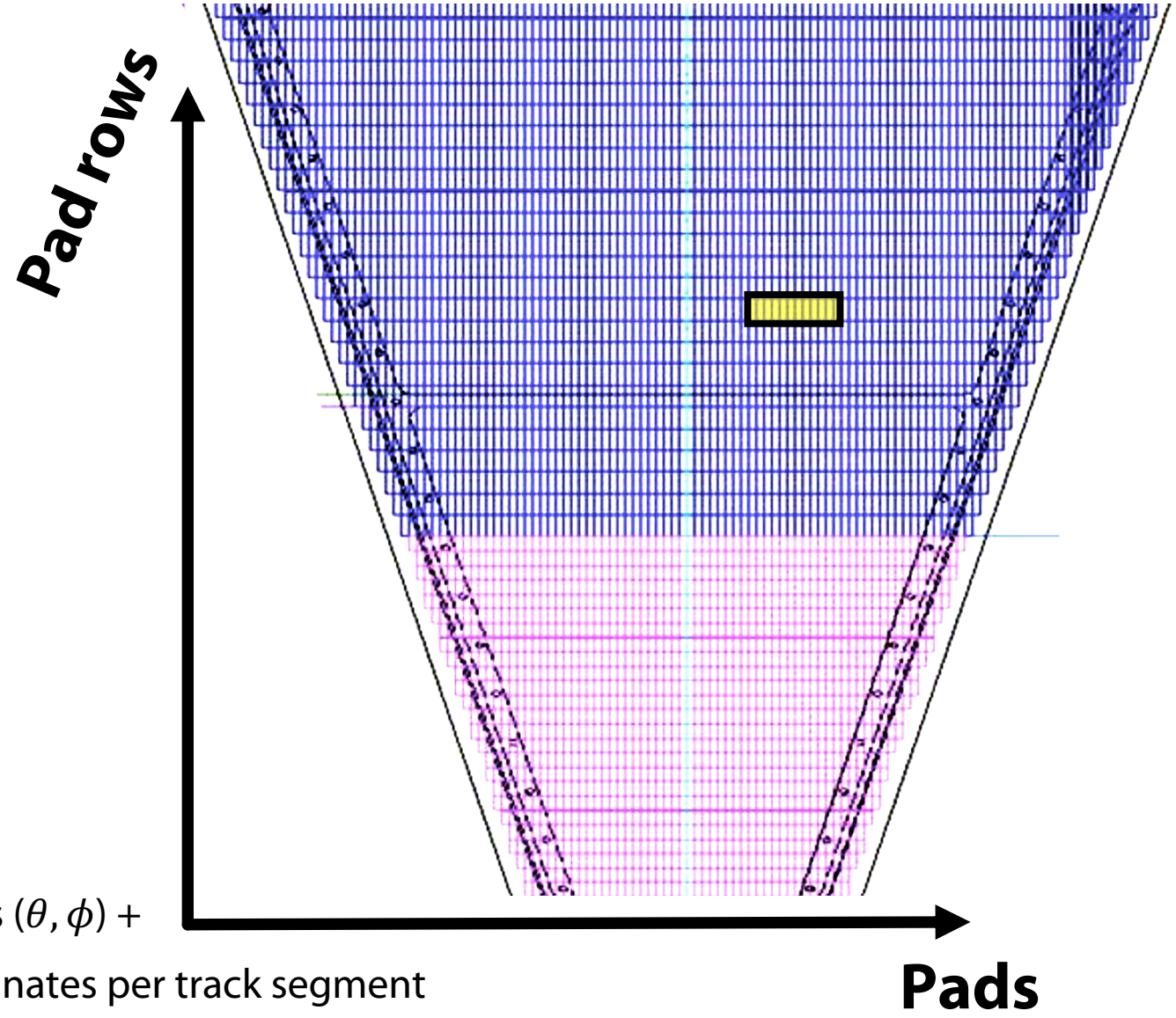


Assumptions for fast simulation

- Factorizing the pad rows
  - dividing tracks to segments, each contributing to a particular pad row
  - can model such contributions **independently**!
- Signal localization (both position & time)
  - model only a **small area** instead of the full row
  - model only a **few time buckets**
- Target dimensionality:

  8 pads x 16 time buckets

  (instead of original 95 232·310)



Input:

2 angles (θ, ϕ) +

3 coordinates per track segment

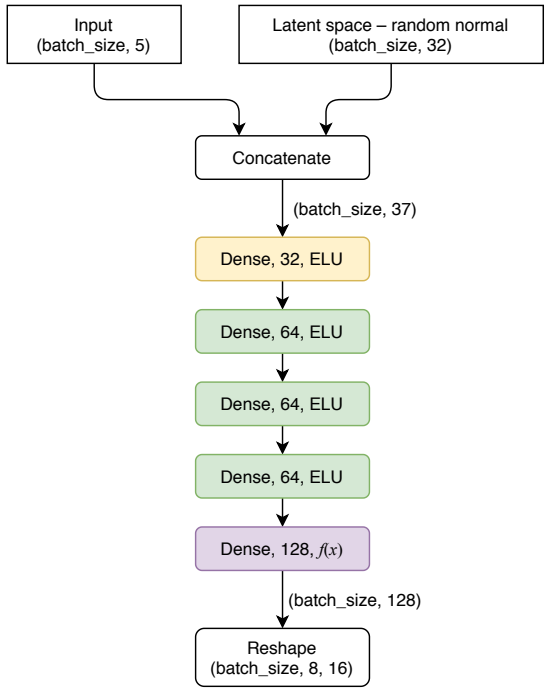
Training data

- 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

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

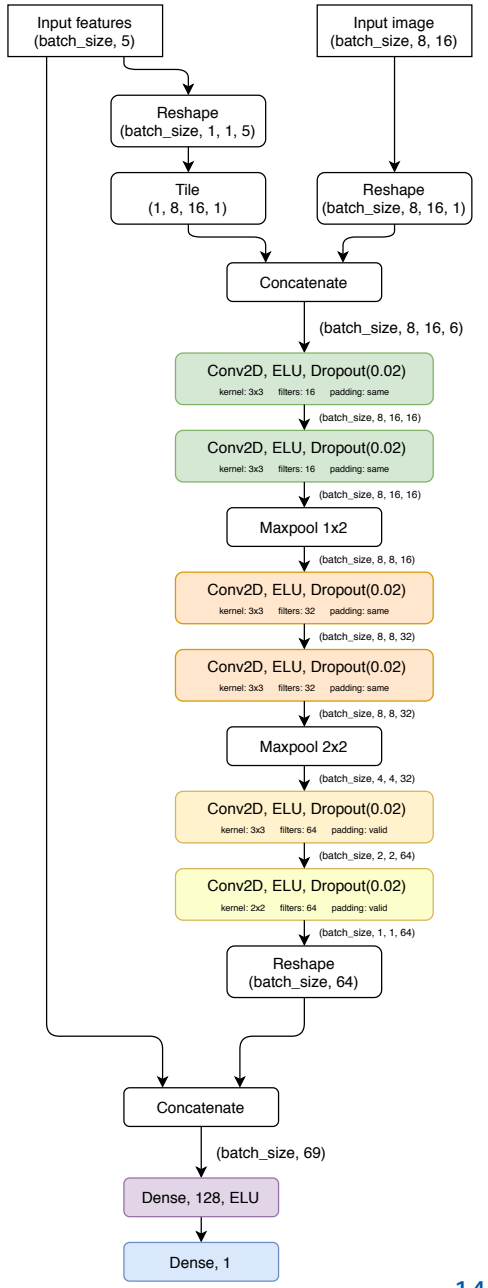
Generator



Custom activation

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

Discriminator

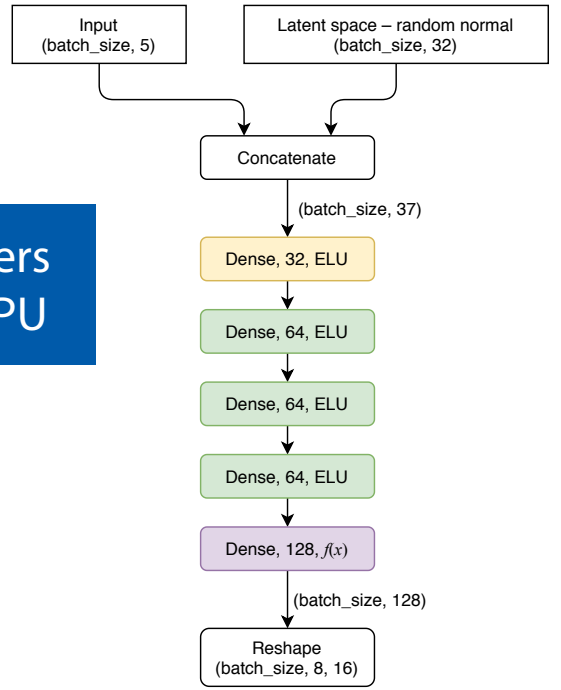


Model details

- 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

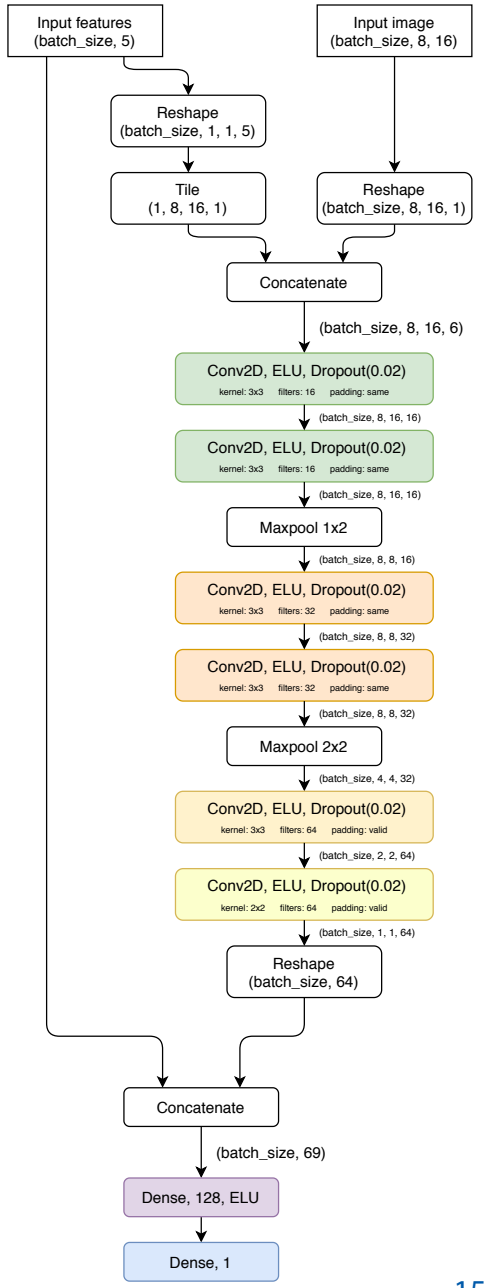
Generator



Custom activation

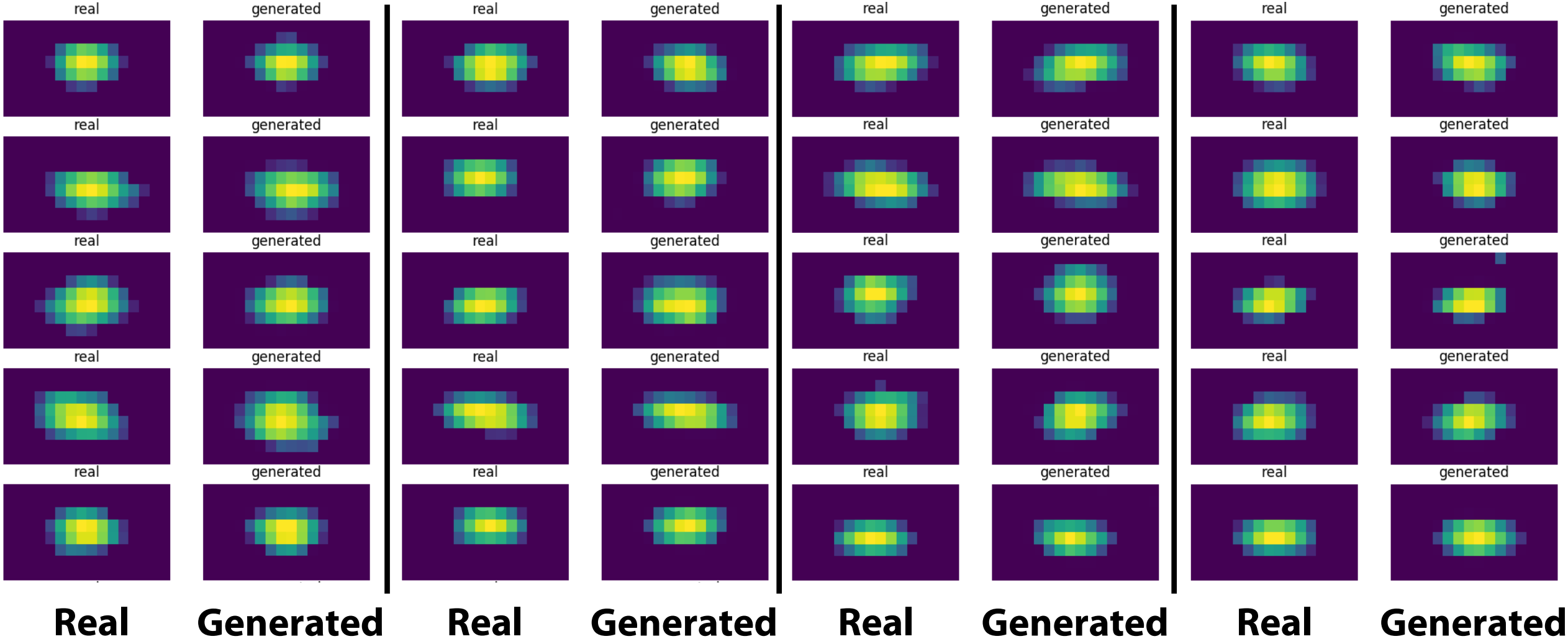
$$f(x) = \begin{cases} \alpha T e^{x} & x \leq 0 \\ T\left(\alpha + (1-\alpha)\frac{x}{\delta}\right) & 0 \leq x \leq \delta \\ T - \delta + x & \delta \leq x, \end{cases}$$

Discriminator

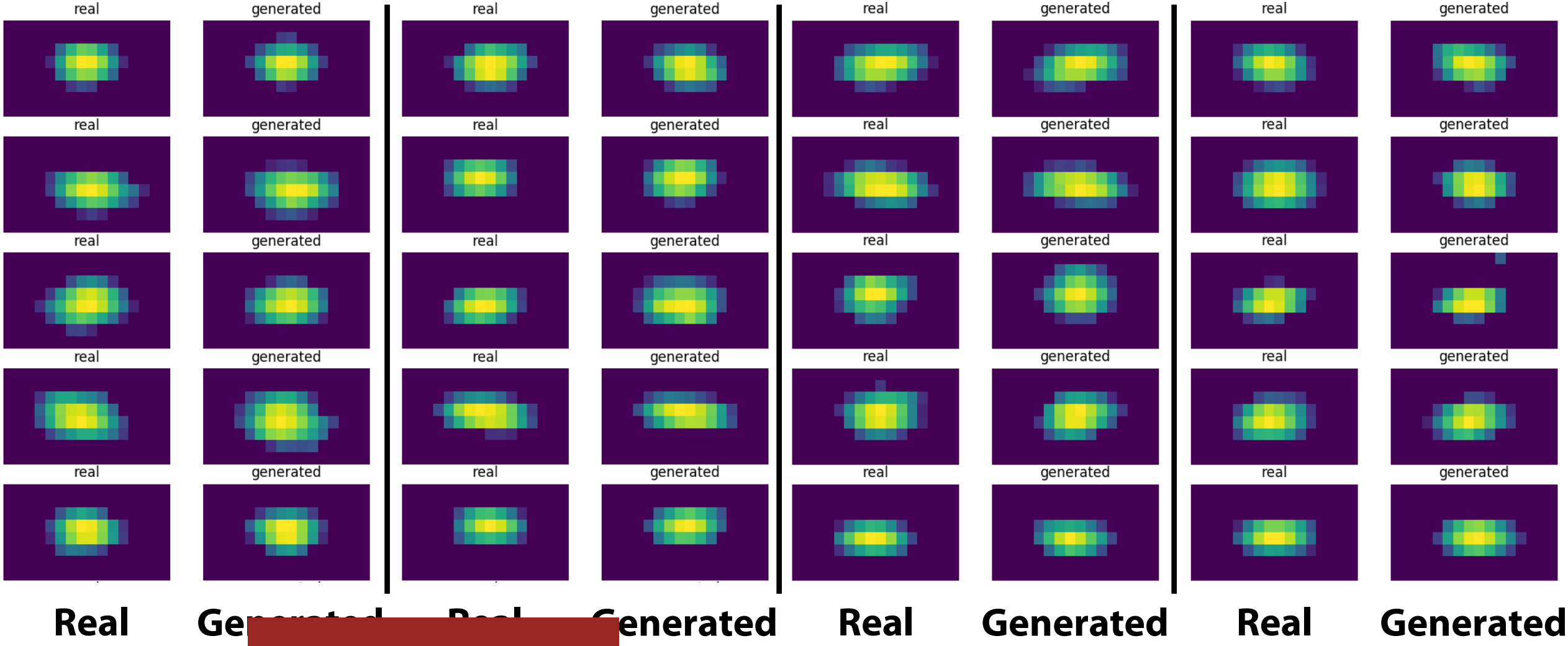


Results

Results

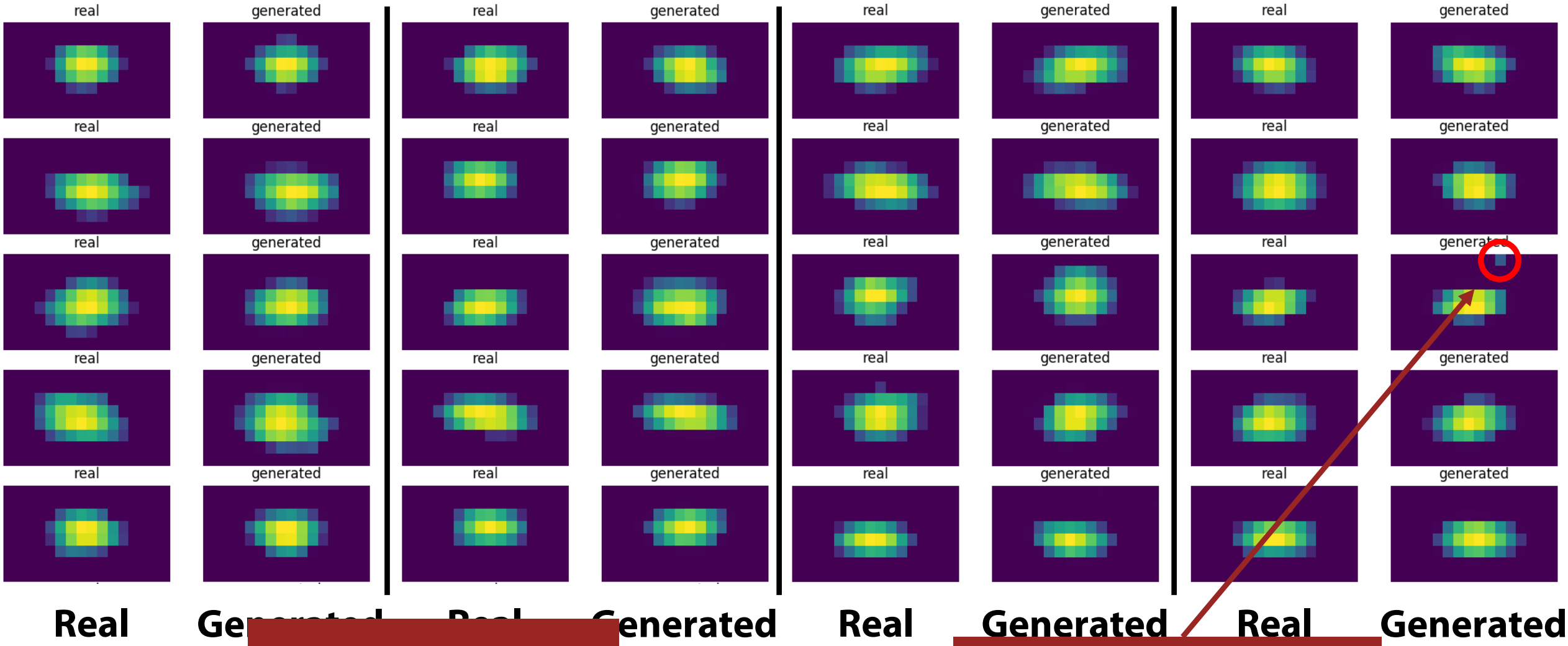

real
generated
Real
Generated
Real
Generated
Real
Generated
Real
Generated

Results



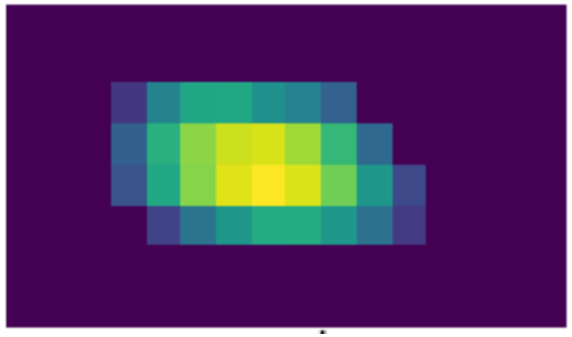
Visually similar!

Results


real
generated
Real
Generated
Real
Generated
Real
Generated
Visually similar!
With minor artifacts

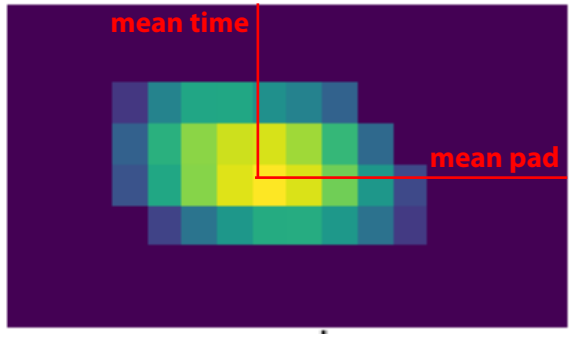
Metrics

- Start with a simple preliminary metric: we compare the 1st & 2nd order moments of the signal images, i.e.:



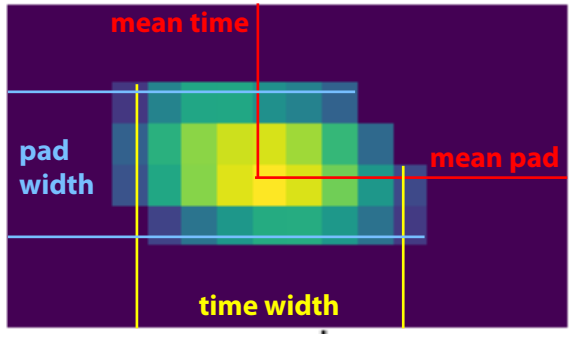
Metrics

- 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



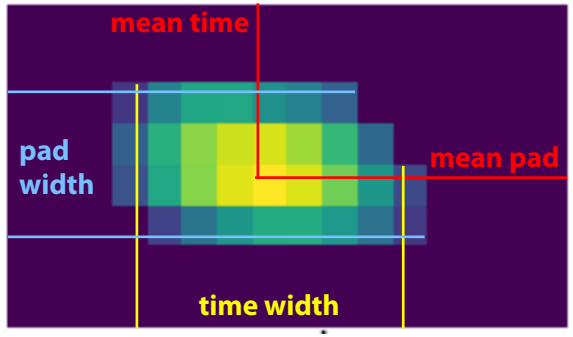
Metrics

- 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



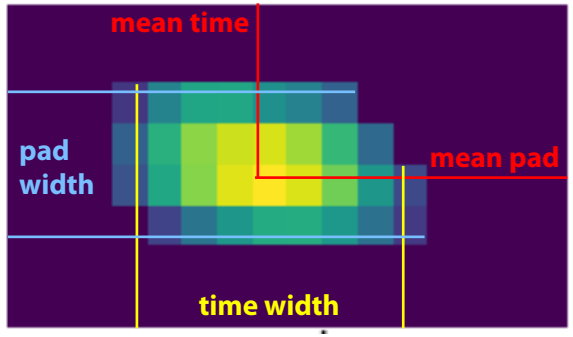
Metrics

- 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



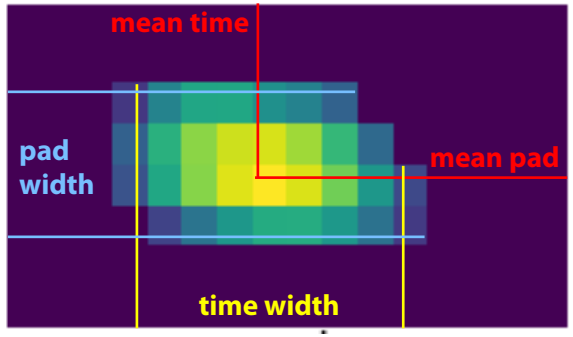
Metrics

- 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



Metrics

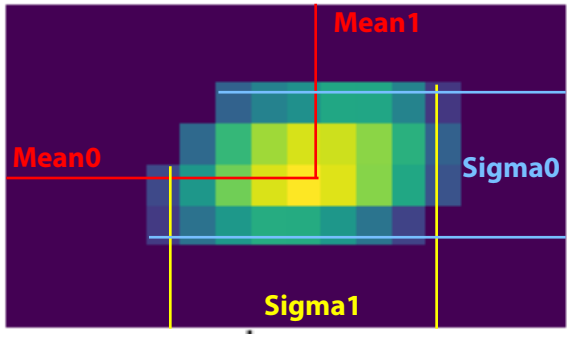
- Start with a simple preliminary metric: we compare the 1st & 2nd order moments of the signal images, i.e.:
  - the location of the signal in pads and time bins
  - the widths of the signal in pads and time bins
  - the tilt of the signal in the pad-time matrix

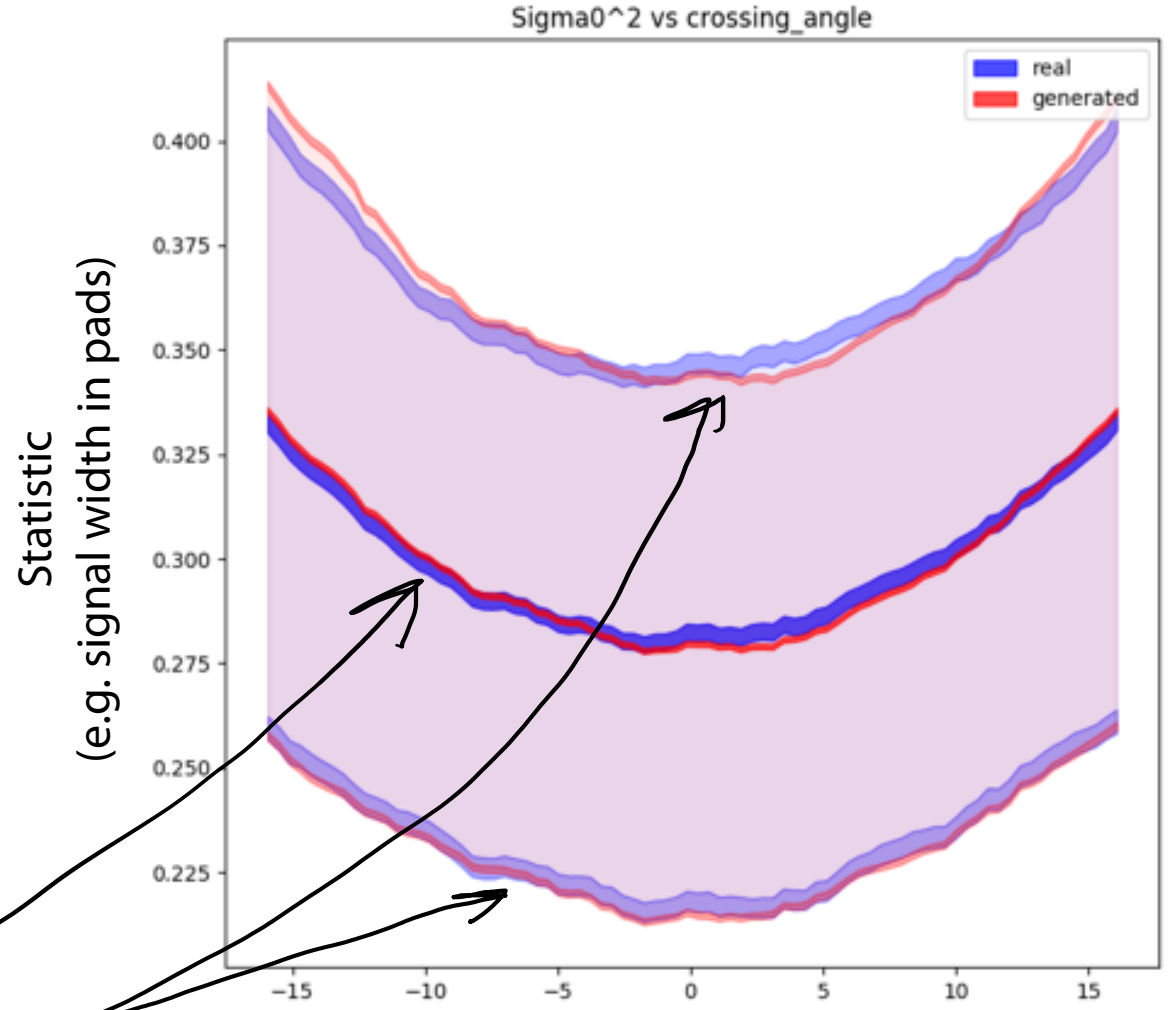


- Also looking at the integrated amplitudes
- All this as a function of track segment parameters (2 angles + 3 coordinates)

Explaining the profiles

Widths of the shaded lines correspond to the statistical uncertainties

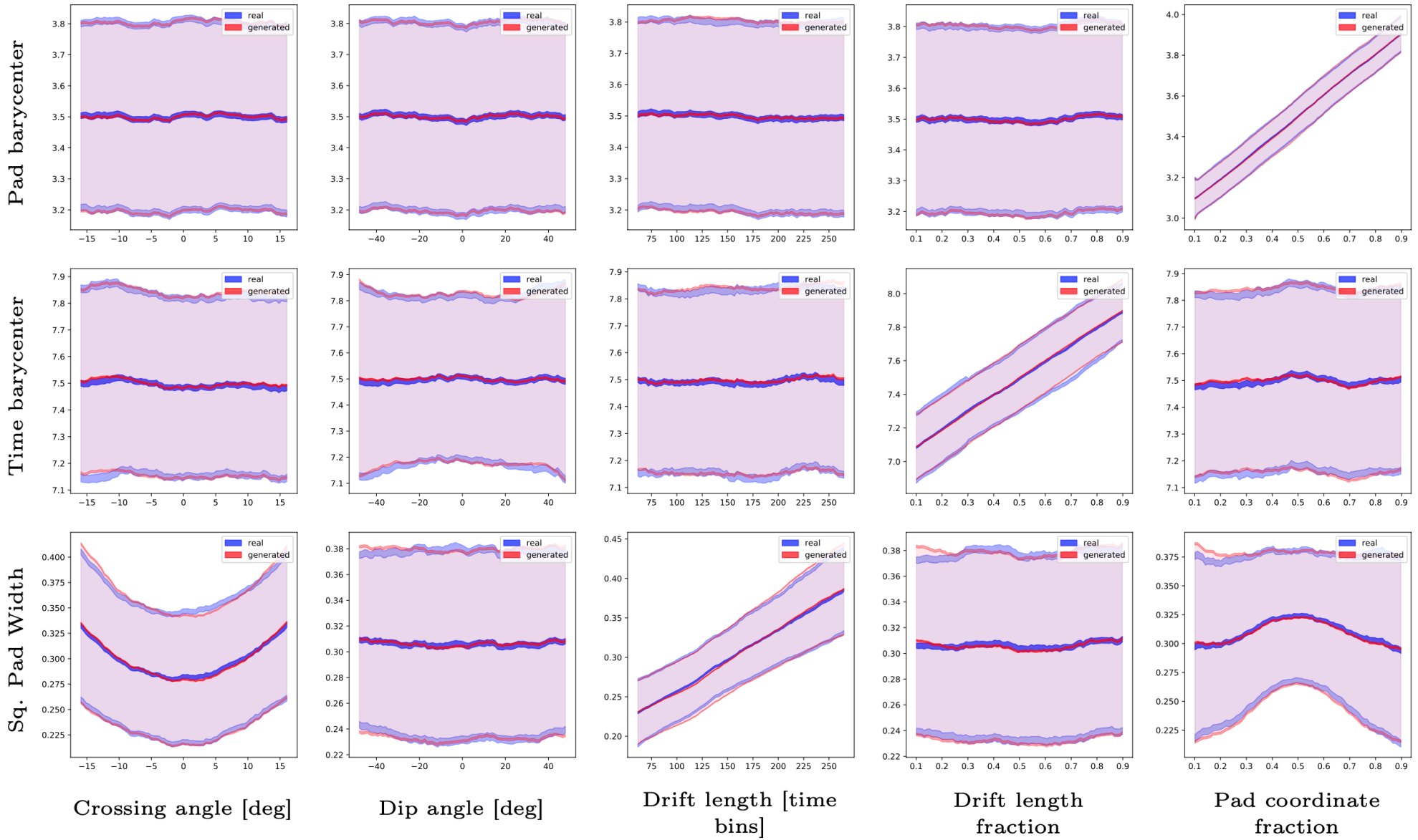




Mean of the statistic

Mean ± 1 standard deviation

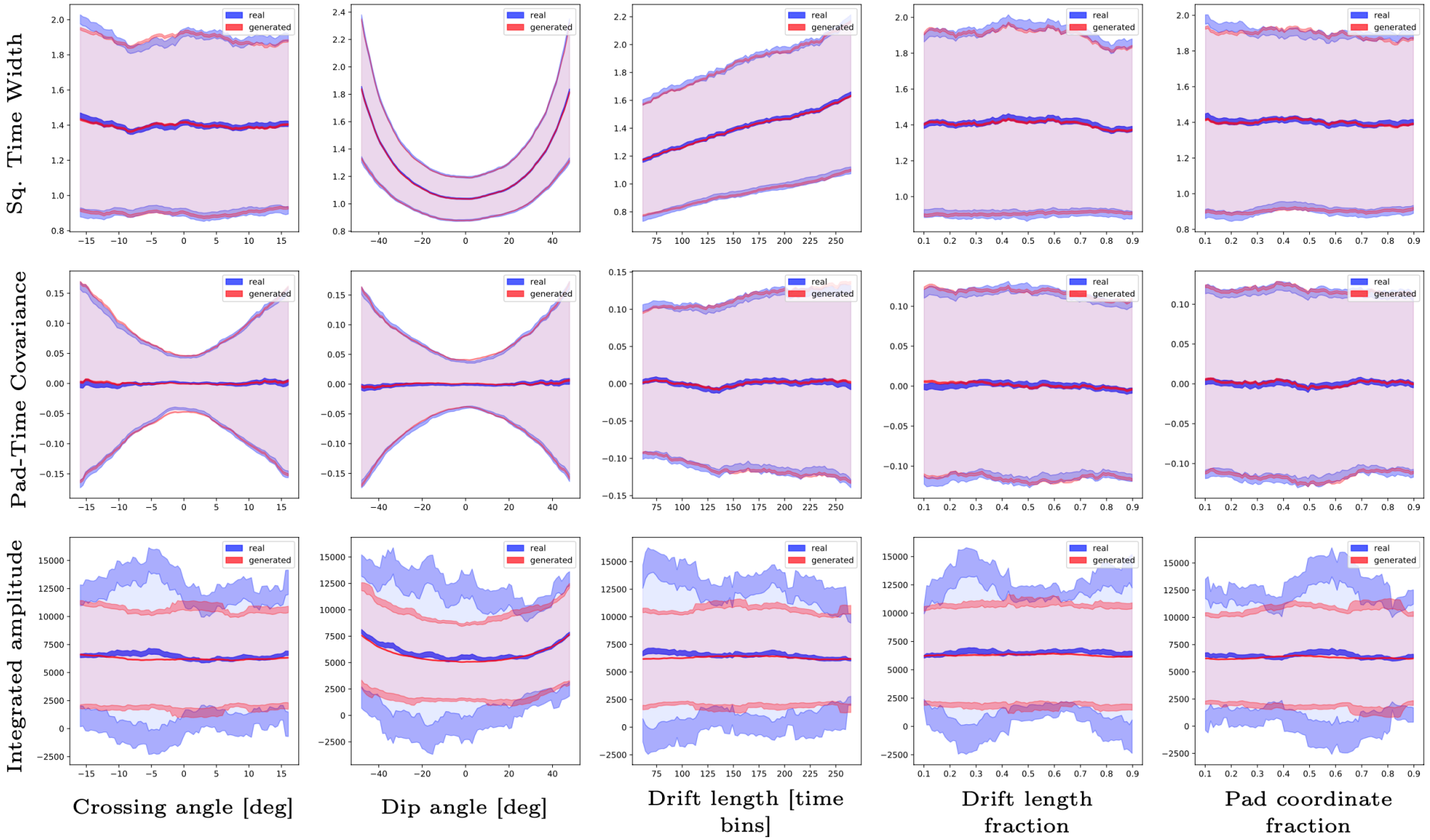
Results (profiles)



Mostly good agreement

"Real"
Generated

Results (profiles)



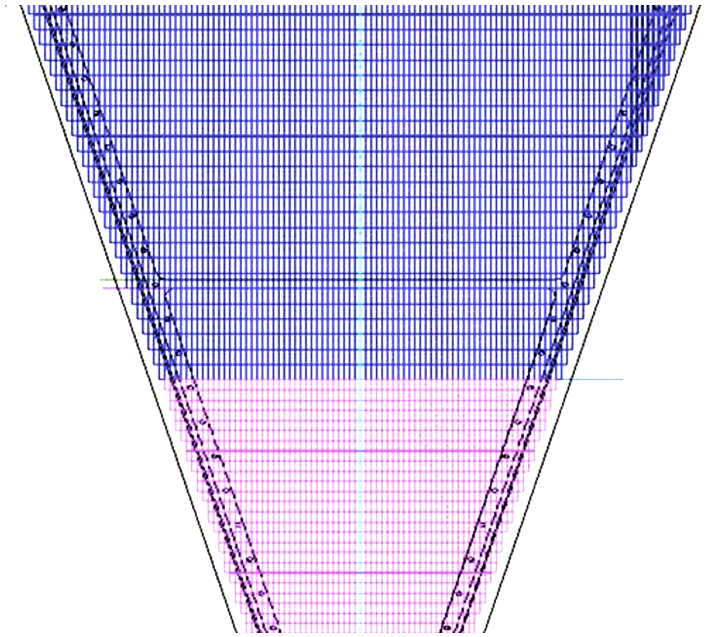
Mostly good agreement

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

"Real"
Generated

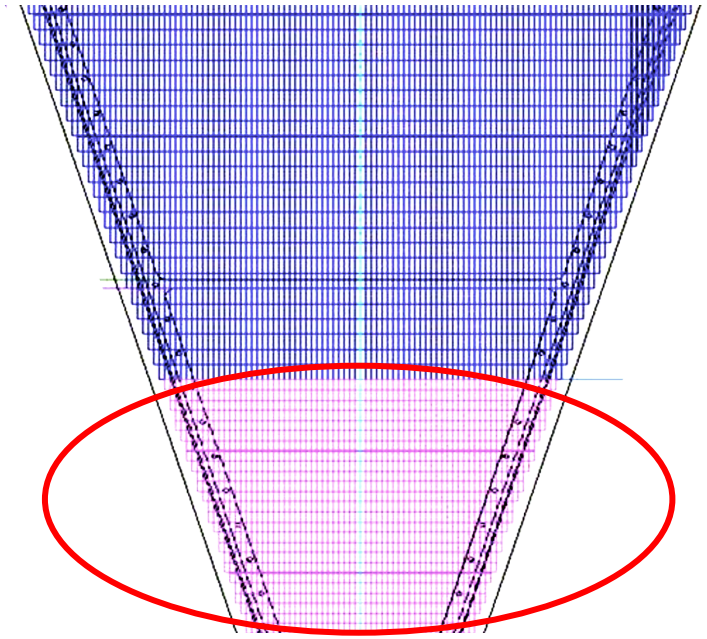
Reconstructed characteristics

- 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



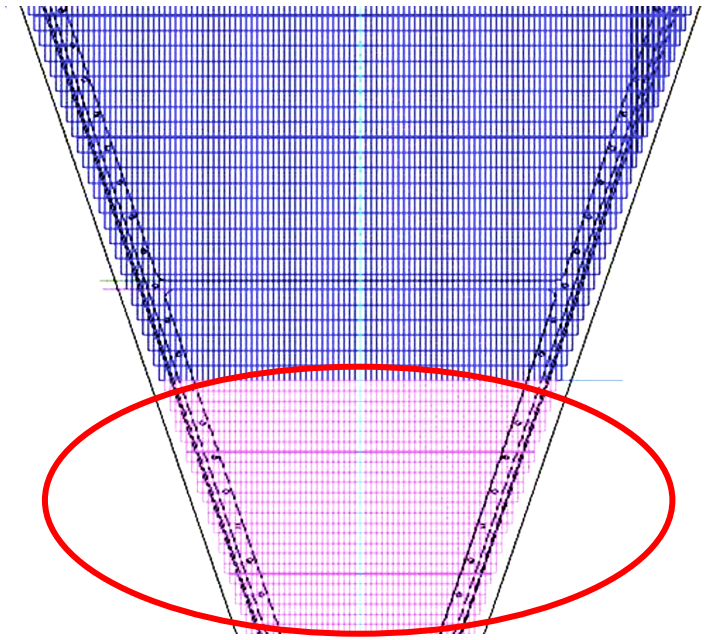
Reconstructed characteristics

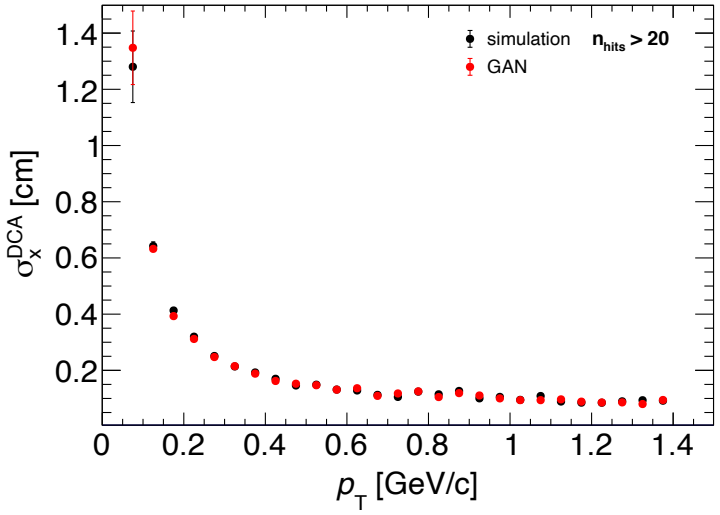
- The model was integrated into the MPD software which allows to validate the reconstruction-level characteristics as well
- Estimated the speed-up to be of **x12**
  - Measured on a single core of an Intel Core i7-3770K (3.50GHz) CPU
- Note: the model was only trained on the responses from the **short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



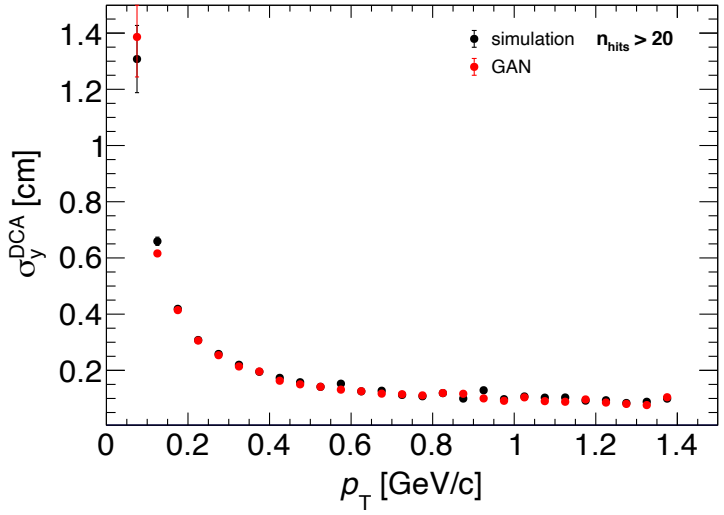
Reconstructed characteristics

- 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 $|y| < 0.5, n_{hits} \geq 20$

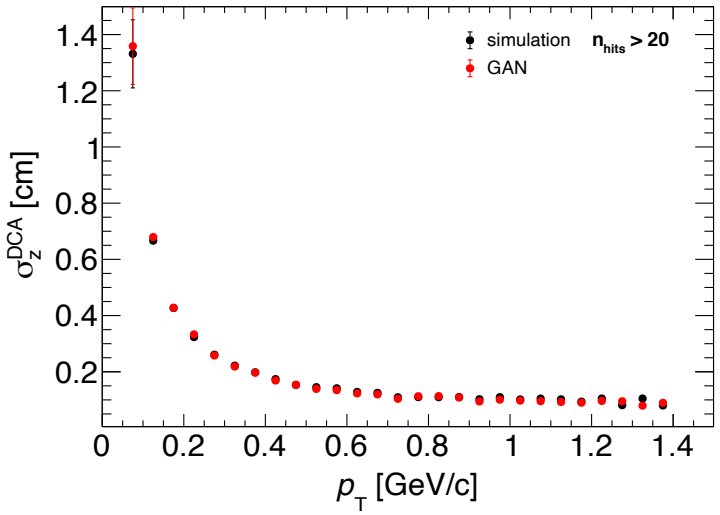




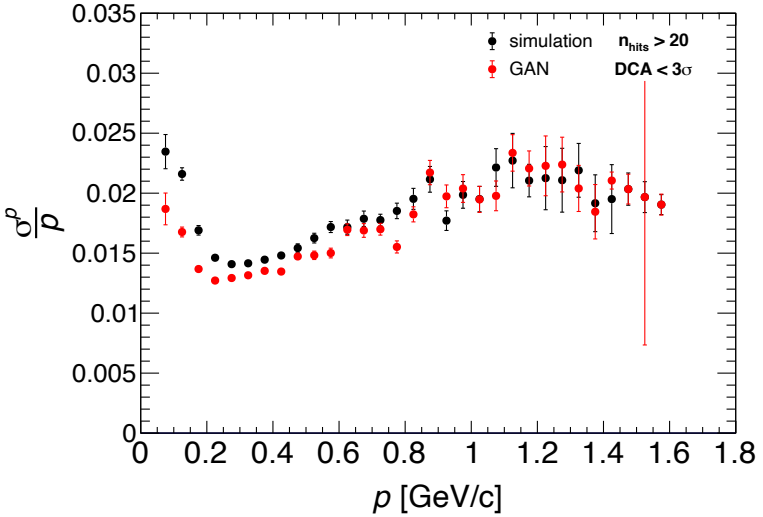
(a) Distance of closest approach resolution along x



(b) Distance of closest approach resolution along y

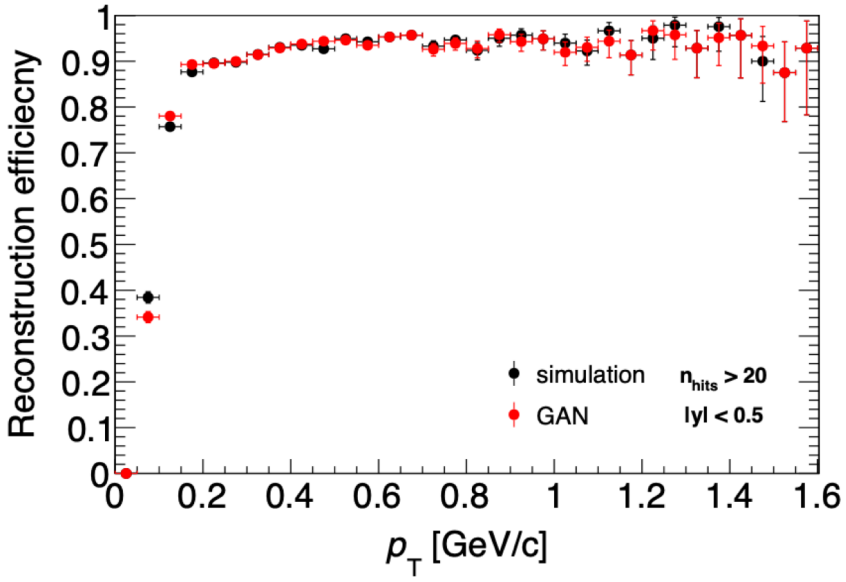


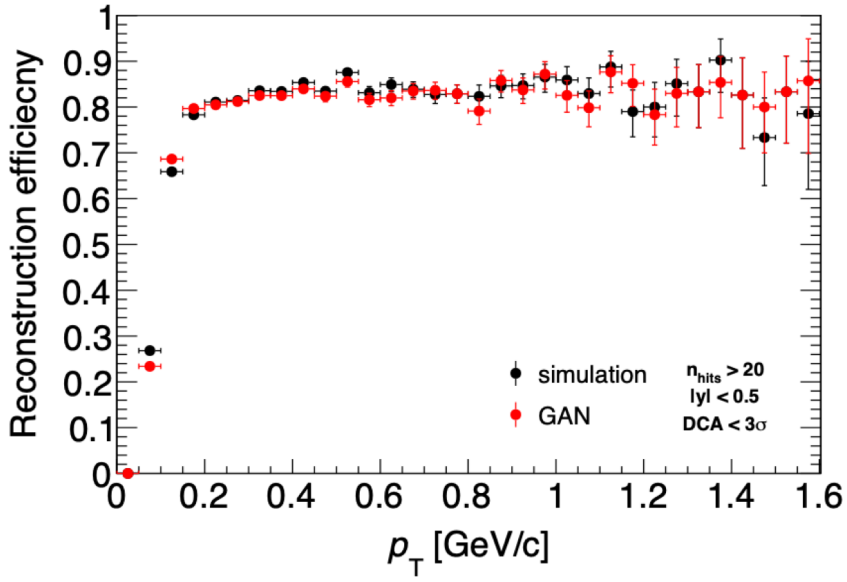
(c) Distance of closest approach resolution along z

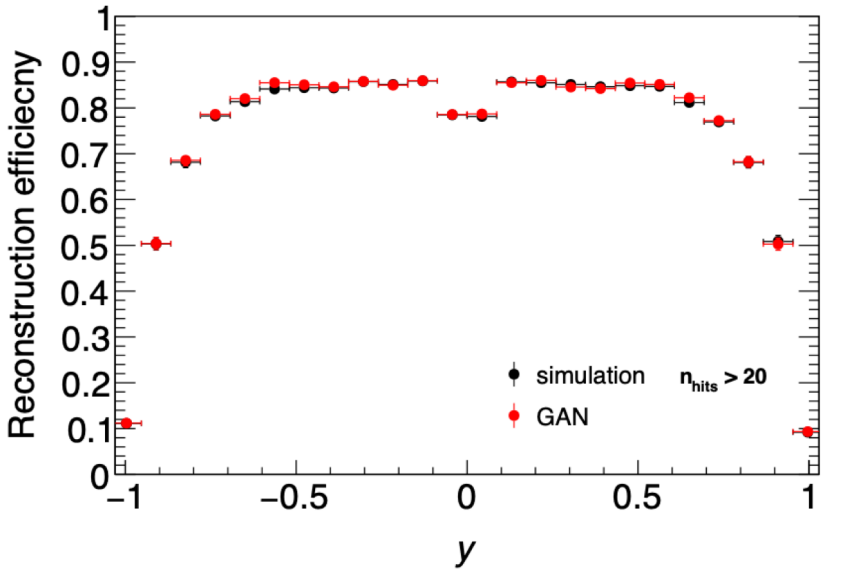


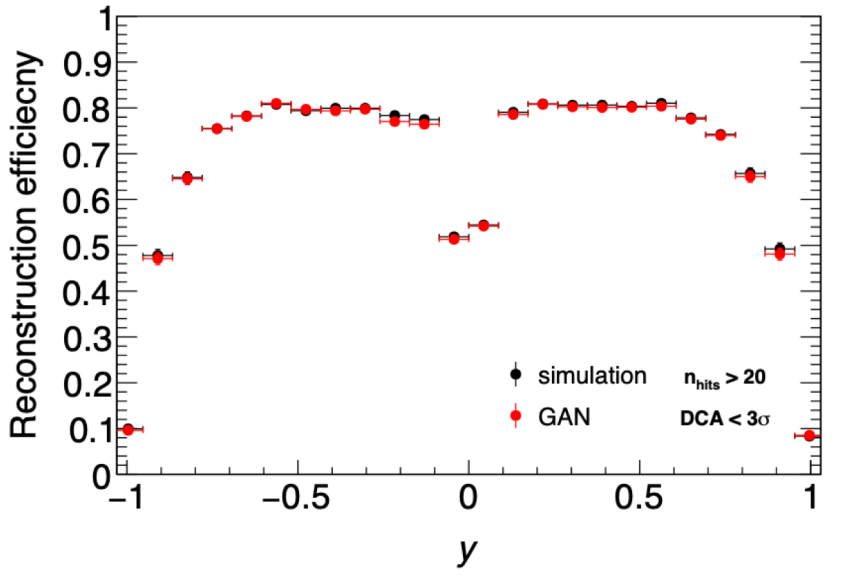
(d) Momentum resolution

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

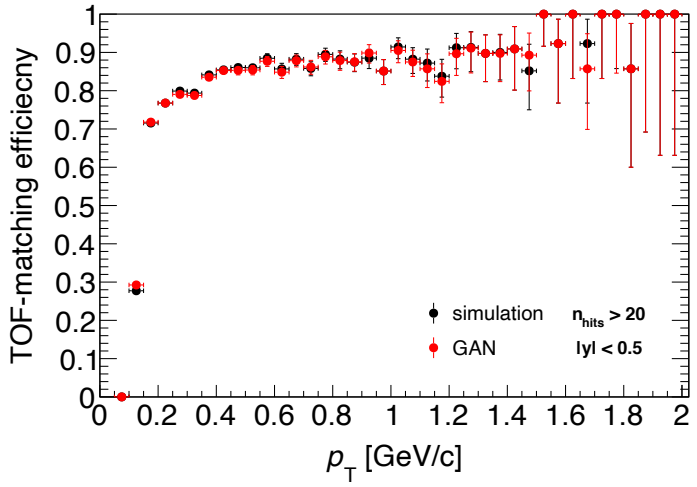




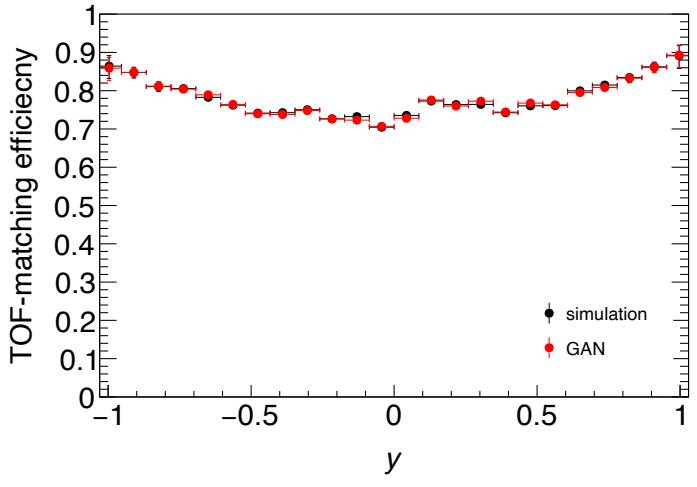




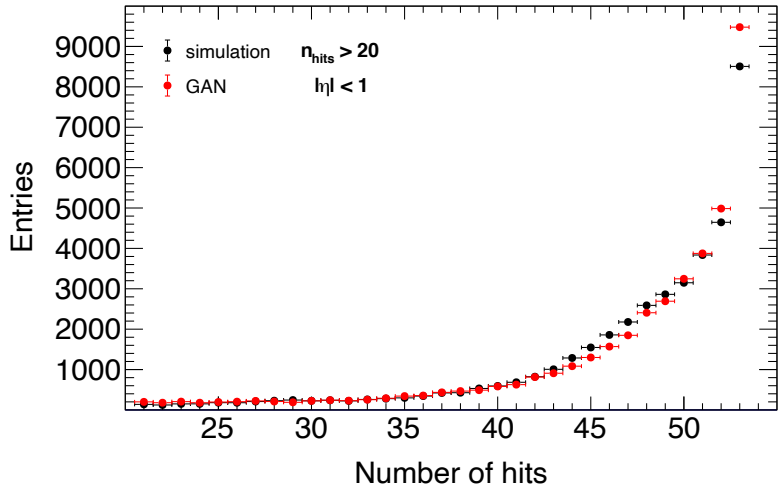
- Reasonable agreement for the reconstruction efficiencies



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



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



(c) Distribution of the number of hits per track

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

Δx for short vs long pads

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

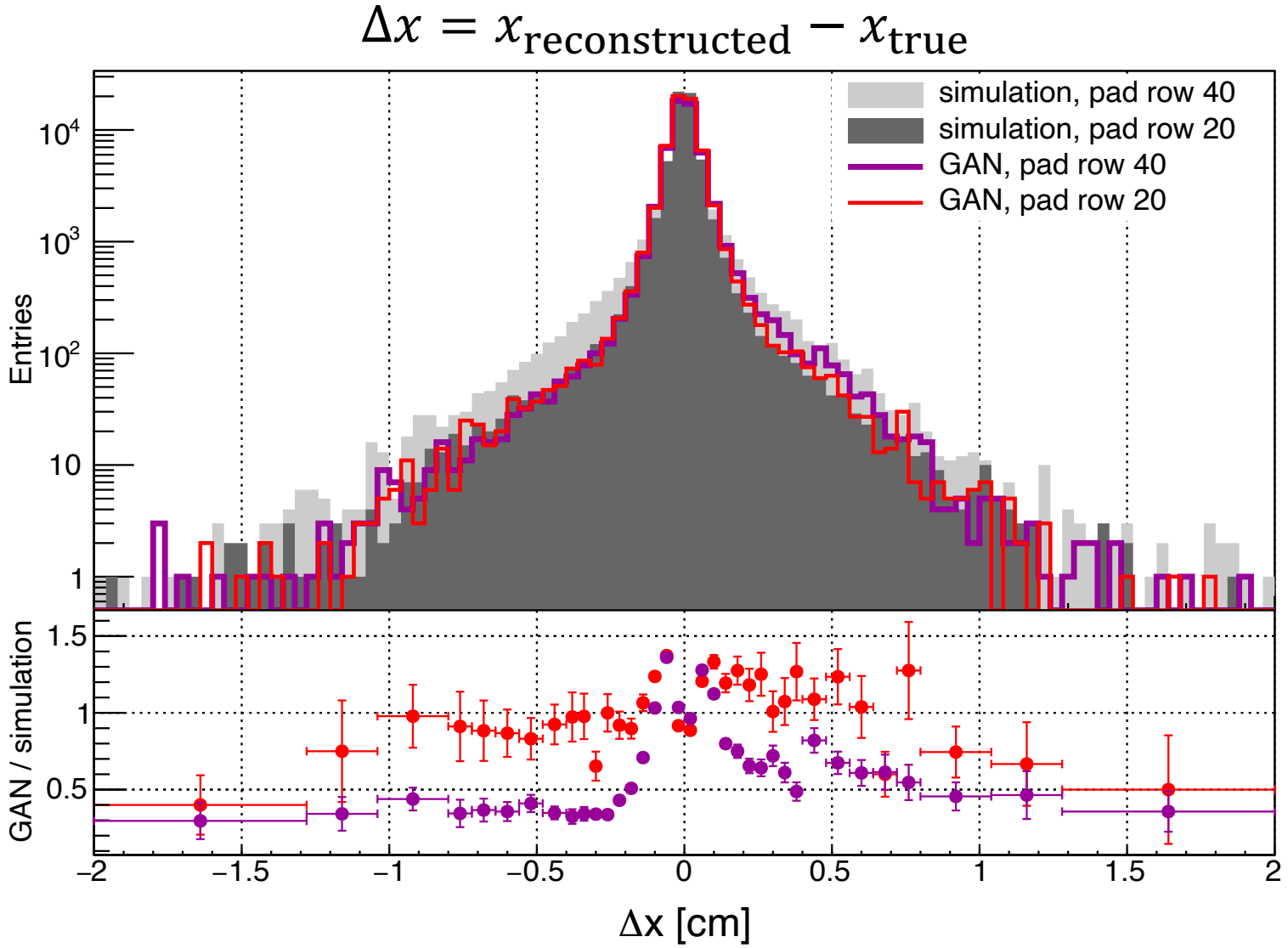


Fig. 7. Distributions of differences $\Delta x = x_{\text{reconstructed}} - x_{\text{true}}$ between the reconstructed and true cluster coordinates along the pad row direction. For the short (long) pads from the pad row 20 (40), the detailed simulation results are shown in the dark (light) gray shaded 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.

Summary

- Promising results
  - Reasonable quality according to the simple metrics and reconstruction-level characteristics
- 12x improvement in speed wrt detailed TPC digitization
- These results are submitted to EPJC (https://arxiv.org/abs/2012.04595)

- Major TODOs:
  - Include training on responses from the long pads as well
  - Introduce more input parameters: various particle types and p_{T}

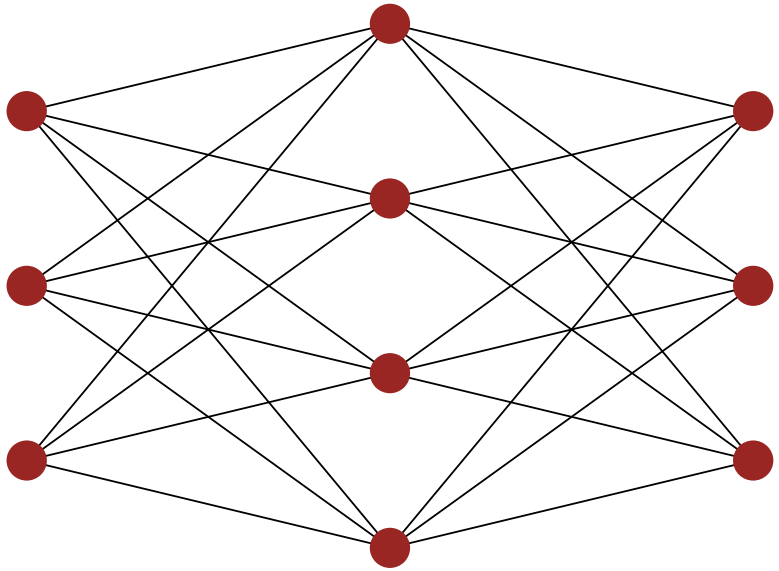
We aim to strengthen the collaboration between HSE and JINR and further improve MPD fast simulation!

Thank you!

Backup

Generative Adversarial Networks

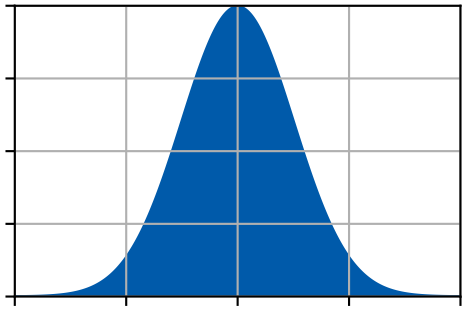
How can a neural network generate data?



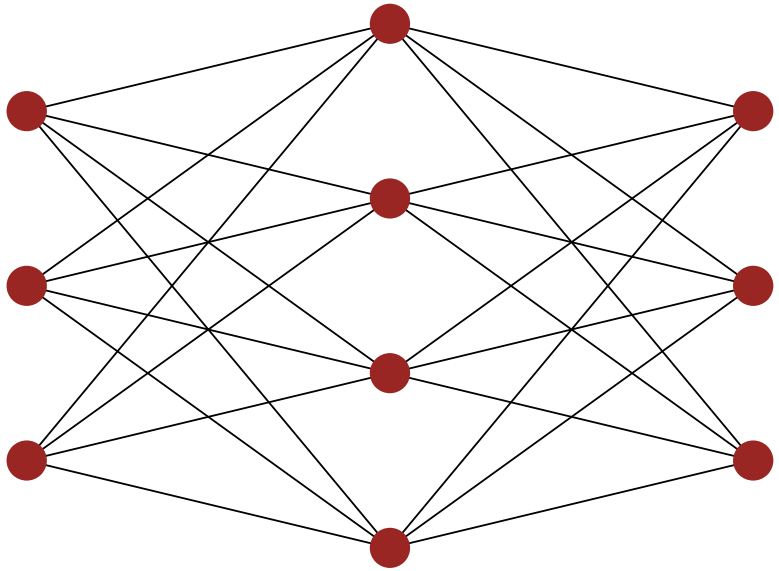
Neural network

Cat image attribution: https://pixabay.com/users/chiemsee2016-1892688/

How can a neural network generate data?



Random noize
e.g. multivariate normal

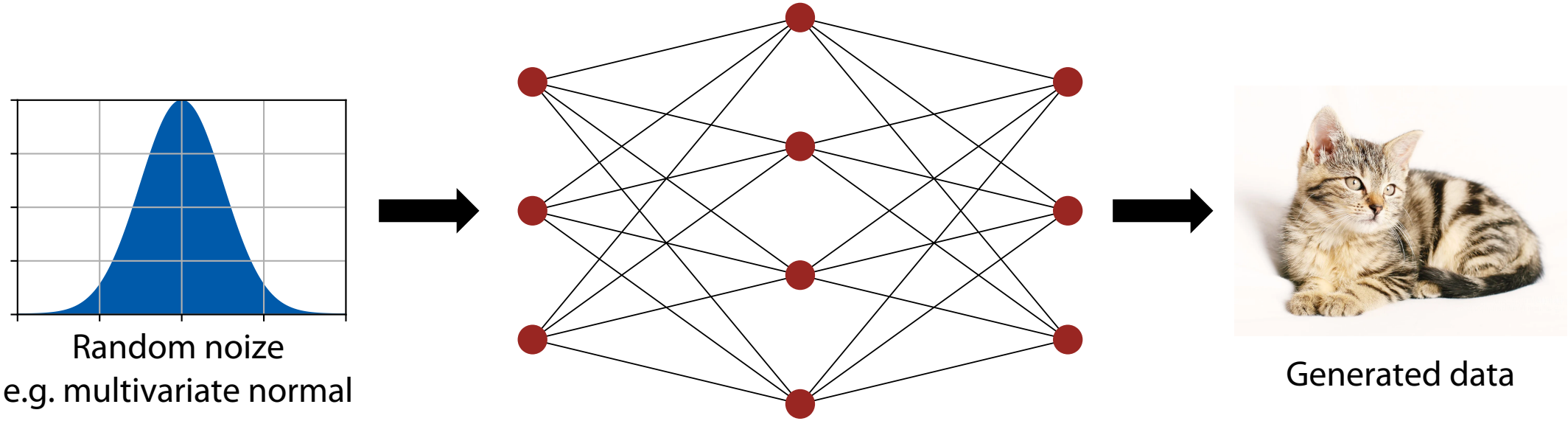


Neural network

Generated data

Cat image attribution: https://pixabay.com/users/chiemsee2016-1892688/

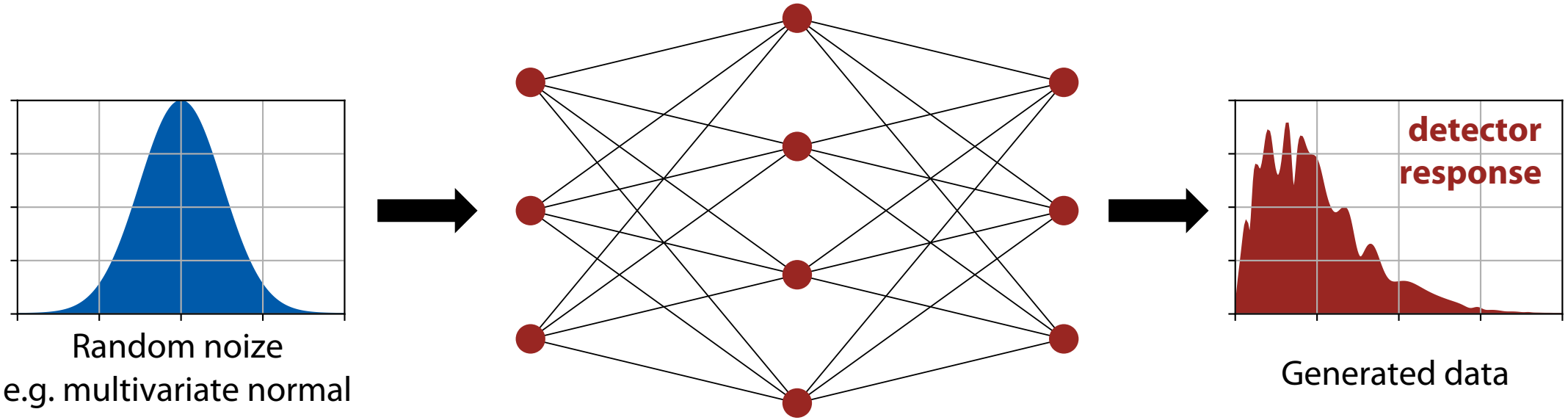
How can a neural network generate data?



- This makes the generated object being a **differentiable function** of the network parameters

Cat image attribution: https://pixabay.com/users/chiemsee2016-1892688/

How can a neural network generate data?



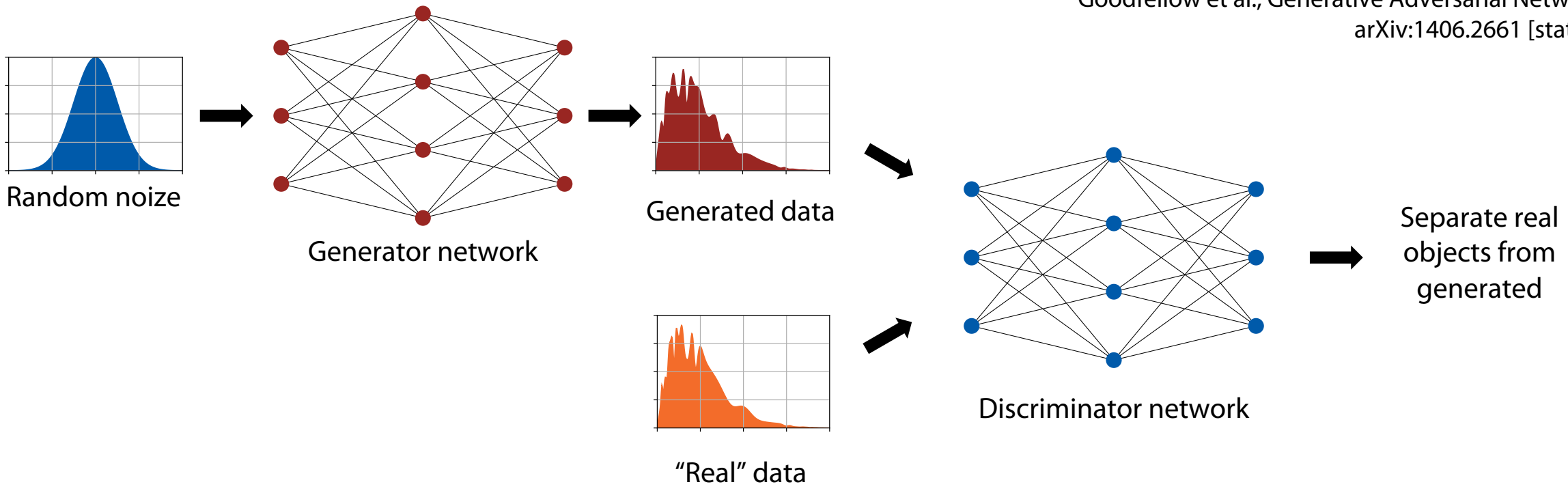
- This makes the generated object being a **differentiable function** of the network parameters

How to train such a generator?

- Generated object is a differentiable function of the network parameters
- Need a differentiable **measure of similarity** between the generated objects and real ones
  - Can optimize with gradient descent
- How to find such a measure?

Adversarial approach

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

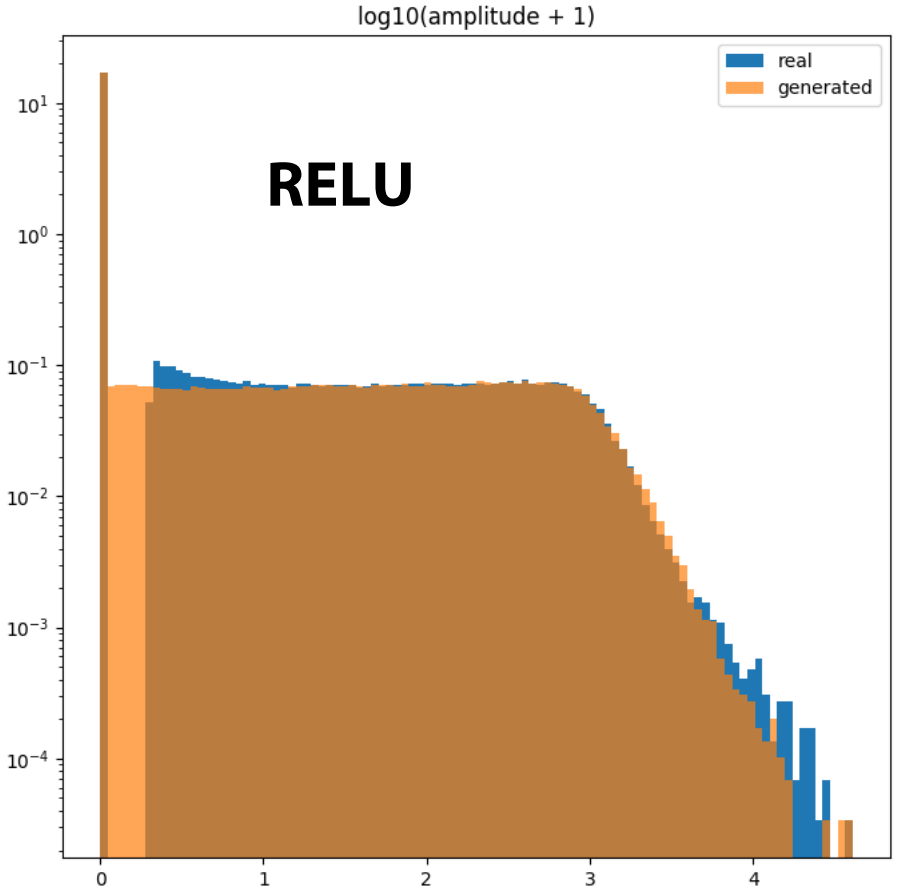


- Measure of similarity: how well can another neural network (discriminator) tell the generated objects apart from the real ones

Output layer activation functions

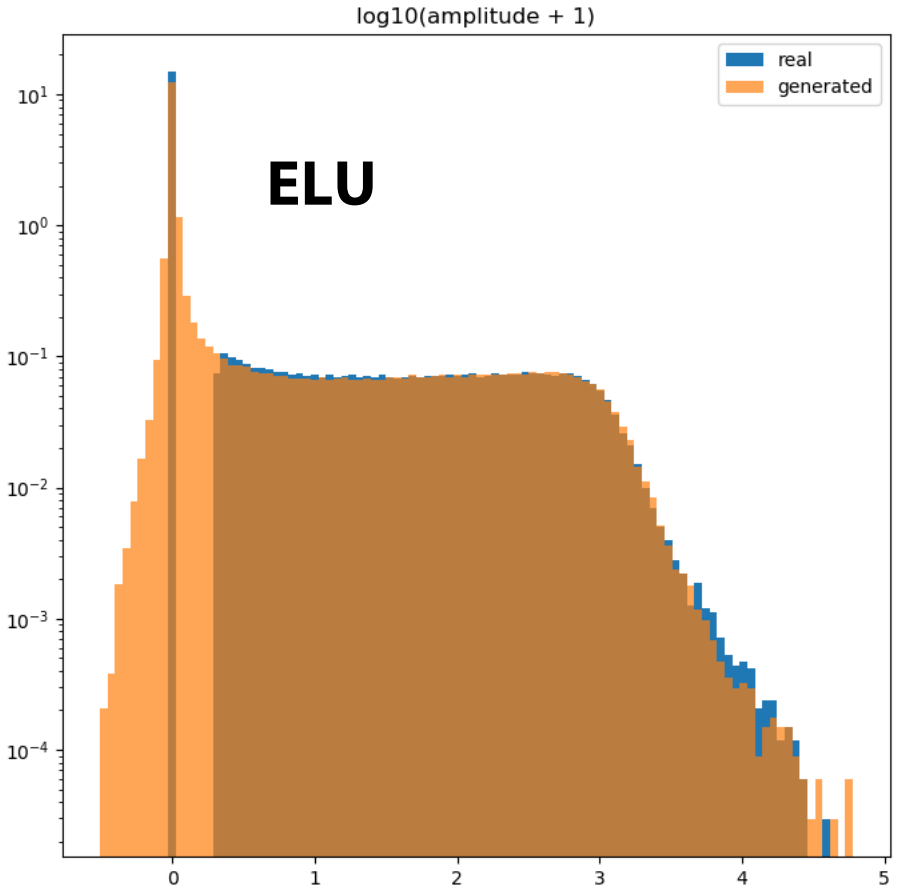
Generator output layer activation

- The signal we're training on is set to 0 for signals below 1
- Problematic for a GAN to learn
- May be able to handle with a proper output layer activation



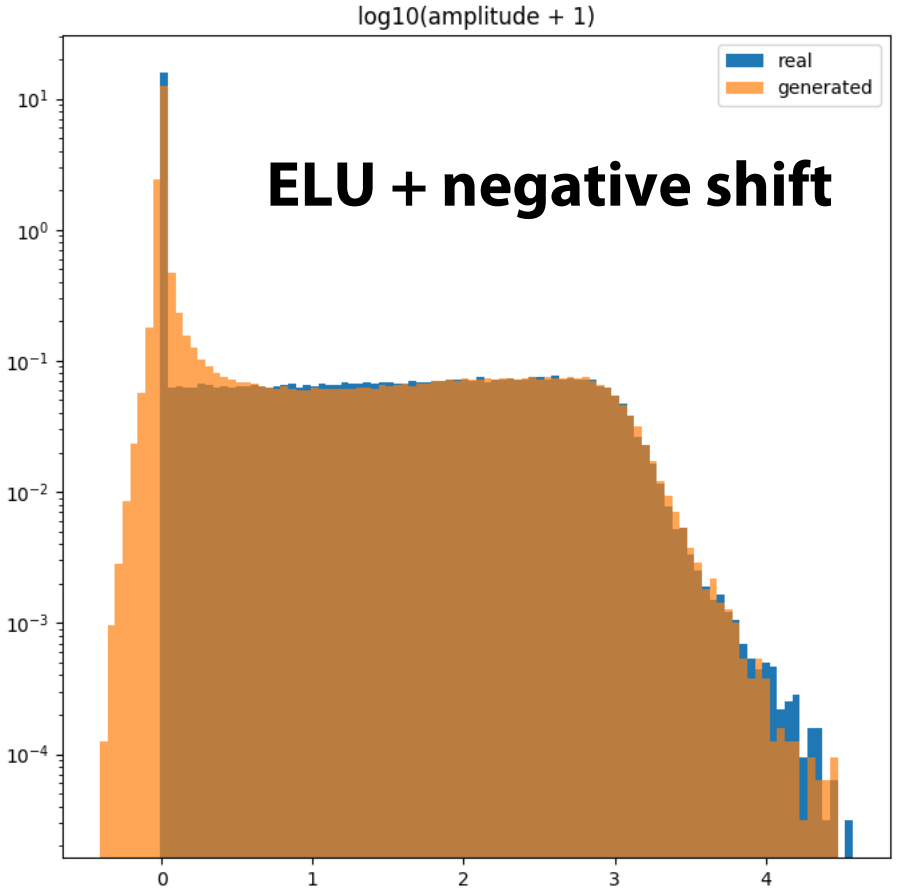
Generator output layer activation

- The signal we're training on is set to 0 for signals below 1
- Problematic for a GAN to learn
- May be able to handle with a proper output layer activation



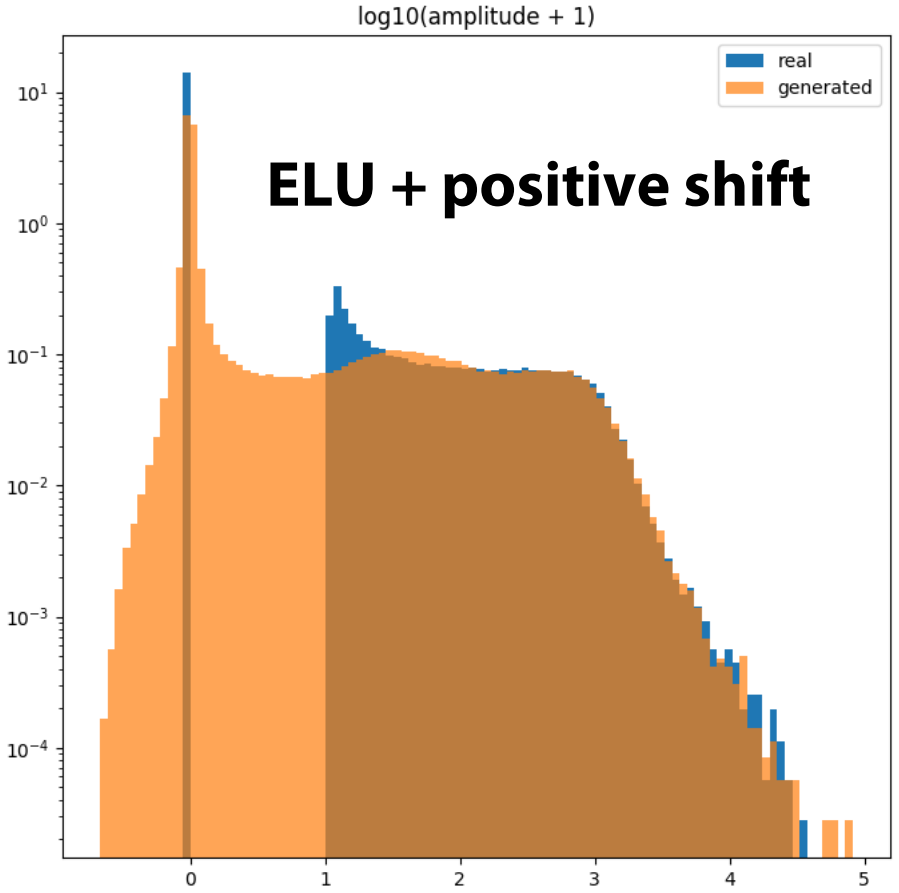
Generator output layer activation

- The signal we're training on is set to 0 for signals below 1
- Problematic for a GAN to learn
- May be able to handle with a proper output layer activation



Generator output layer activation

- The signal we're training on is set to 0 for signals below 1
- Problematic for a GAN to learn
- May be able to handle with a proper output layer activation



Generator output layer activation

- The signal we're training on is set to 0 for signals below 1
- Problematic for a GAN to learn
- May be able to handle with a proper output layer activation