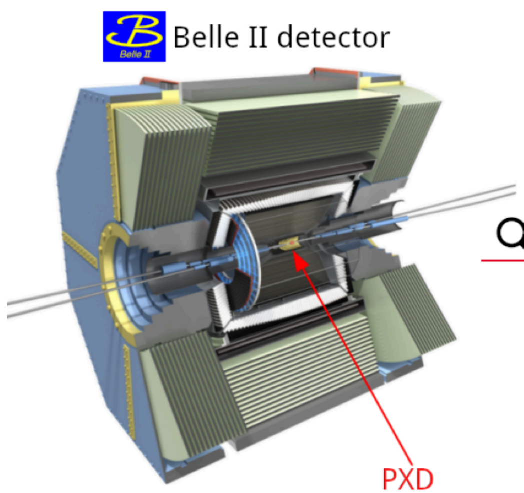
Generating PXD Background Hitmaps with Generative Adversarial Networks at Belle II

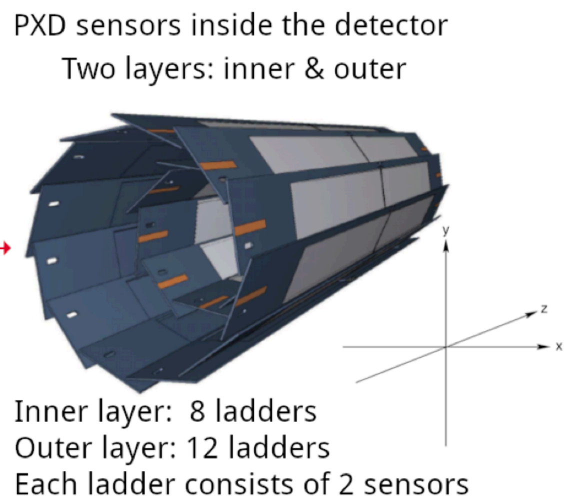
Hosein Hashemi, Nikolai Hartmann,
Thomas Kuhr, Martin Ritter, Matej Srebre

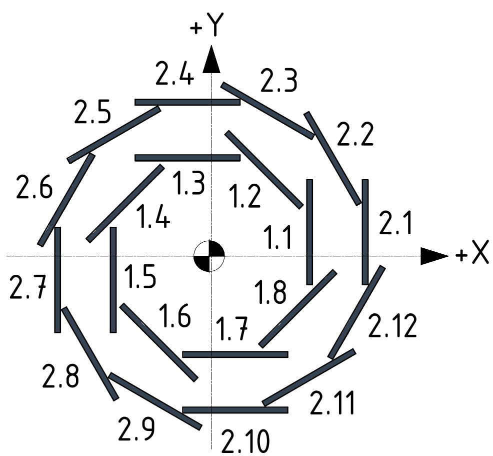
Ludwig-Maximilians-Universität München
The ORIGINS Excellence Cluster

Introduction

- ***The Pixel Vertex Detector (PXD)*** is the innermost semi-conductor sub-detector at Belle II.
- The sensitive area of the PXD is assembled from ***40 modules***, where each module consists of a ***250*** $\times$ ***768*** pixel matrix of the pixel sensors.
- **The inner layer:** 16 modules implemented into 8 ladders
- **The outer layer:** 24 modules implemented into 12 ladders



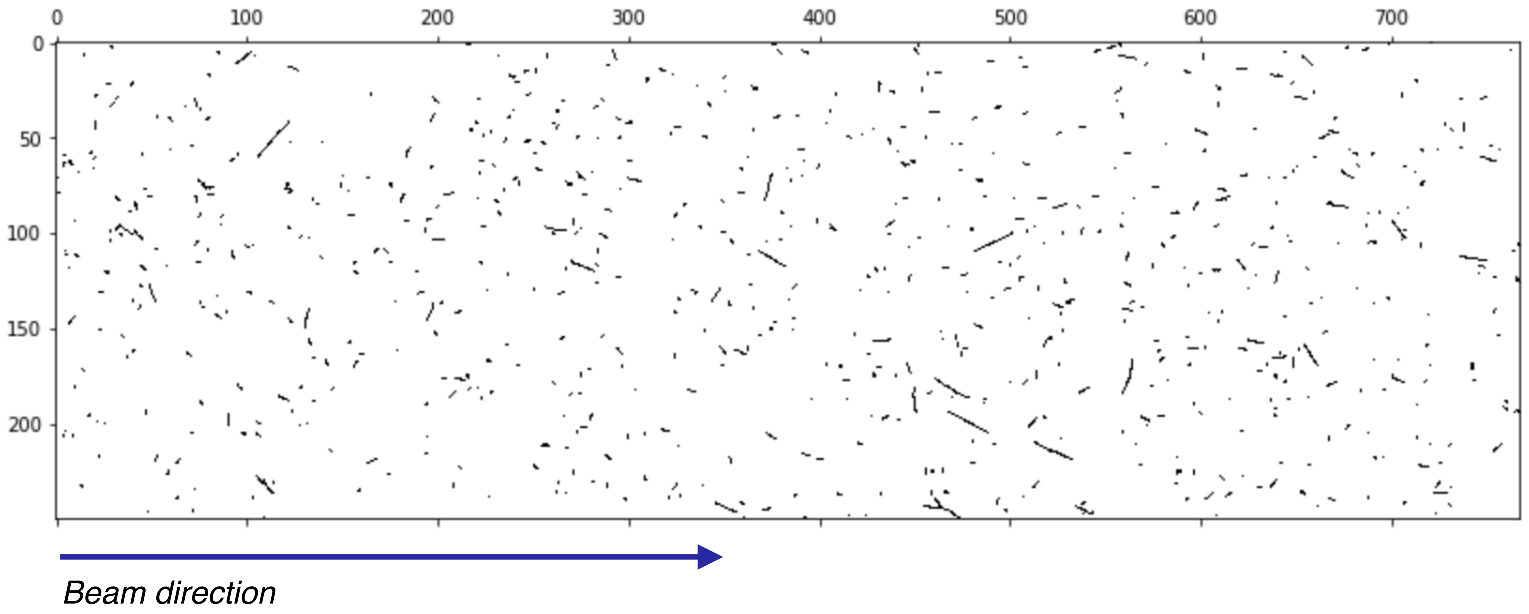




Backgrounds

- ***The PXD hits are coming from two sources:***
  - **Signal Decays:**
    A. *particles which originate from the physics processes of interest.*
  - **Backgrounds:**
    A. ***Beam-induced****: intra-beam scattering, Beam-Gas scattering, synchrotron radiation*
    B. ***Luminosity dependent:*** *Radiative Bhabha scattering, two-photon process*

Colour-reversed real (simulated) image



Backgrounds

- **Realistic detector simulation has to take into account effects from background processes**
  - *Simulation requires many PXD hitmaps with statistically independent background.*
  - *Overlay hits from simulated background or random trigger data to hits from signal MC.*
  - *PXD hits have the highest storage consumption, almost 200 kB per event cost.*
  - *Requires distributing over all sites where MC is produced.*

Backgrounds

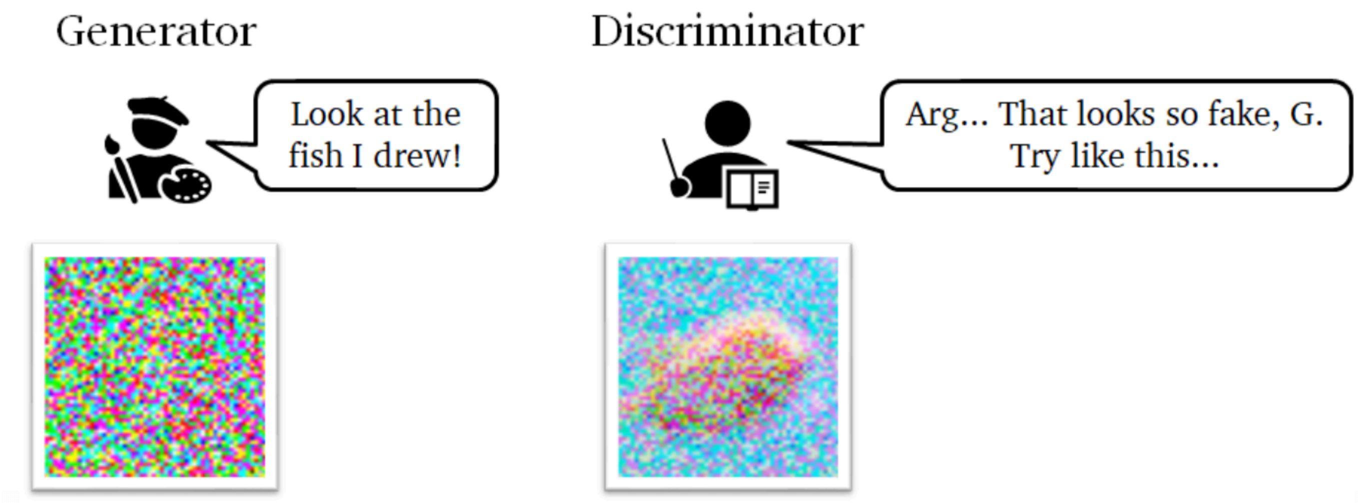
- **Realistic detector simulation has to take into account effects from background processes**
  - *Simulation requires many PXD hitmaps with statistically independent background.*
  - *Overlay hits from simulated background or random trigger data to hits from signal MC.*
  - *PXD hits ... ge consumption, almost 200 kB per event cost.*
  - *Re ... re MC is produced.*

Solution:

Let's generate ze bkg on ze way of analysis with Generative Adversarial Networks, instead of storing them.

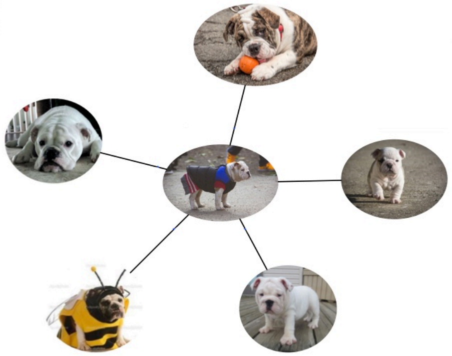
GAN is all you need!

- Generate PXD background events with <u>Generative Adversarial Network</u> **(GAN)**
- **Whats is GAN?**



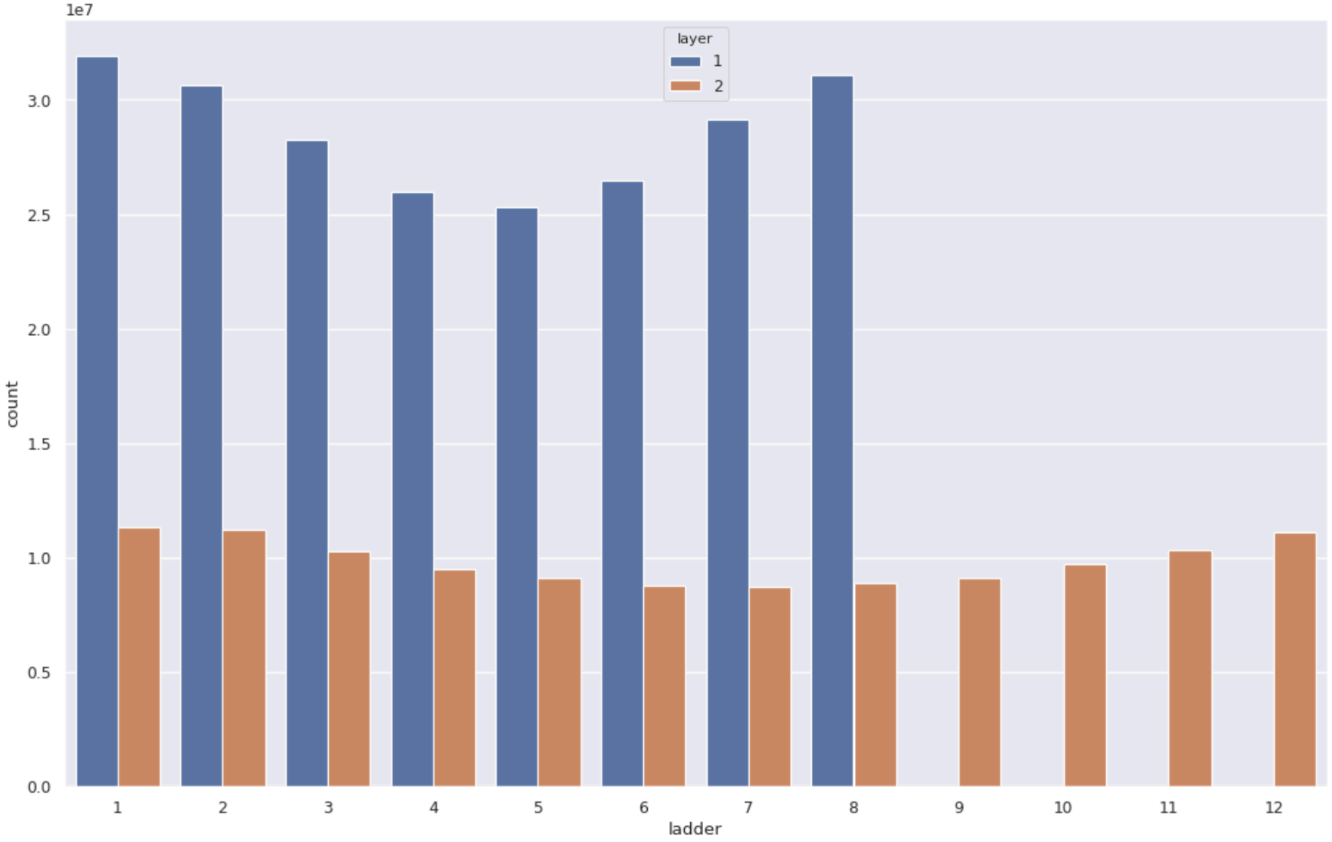
The GAN game

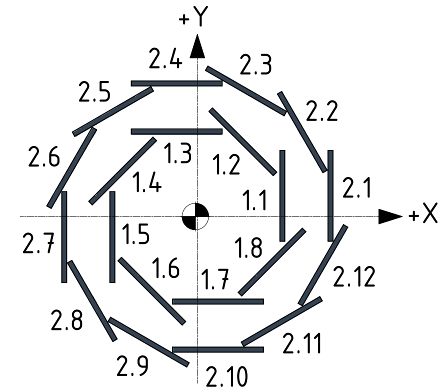
- **Conditional GAN :** *The type of animal is the condition*
- ***Close-Conditional (relational)***



Conditional GAN

- **Using spatial class-conditions based on the sensor number 1-40:**
  - ☑ *Increase the image fidelity*
  - ☑ *Generation of sensor-dependent images*

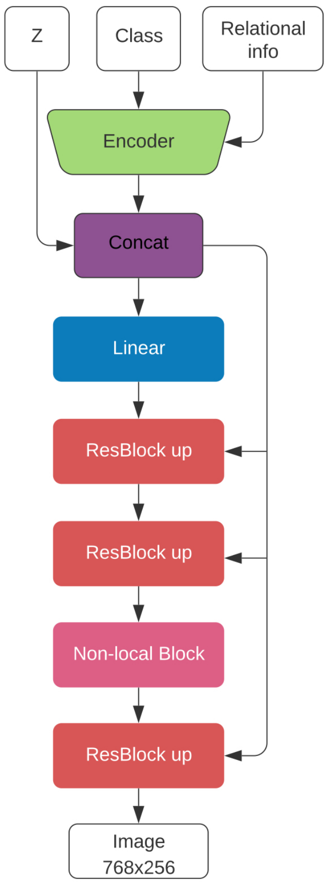




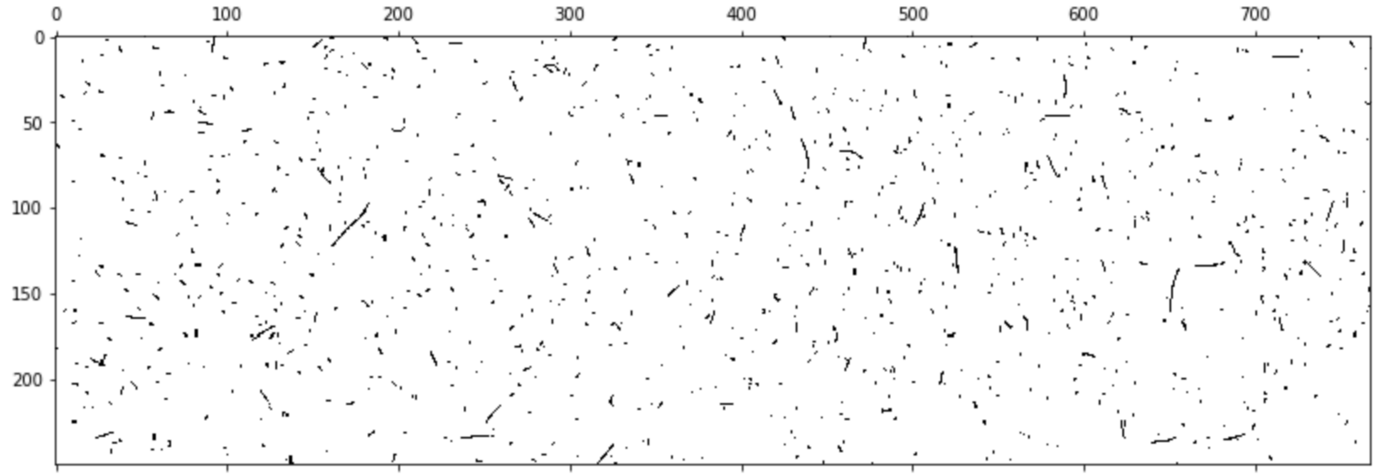
Generated vs Real PXD Images

Model: (updated)

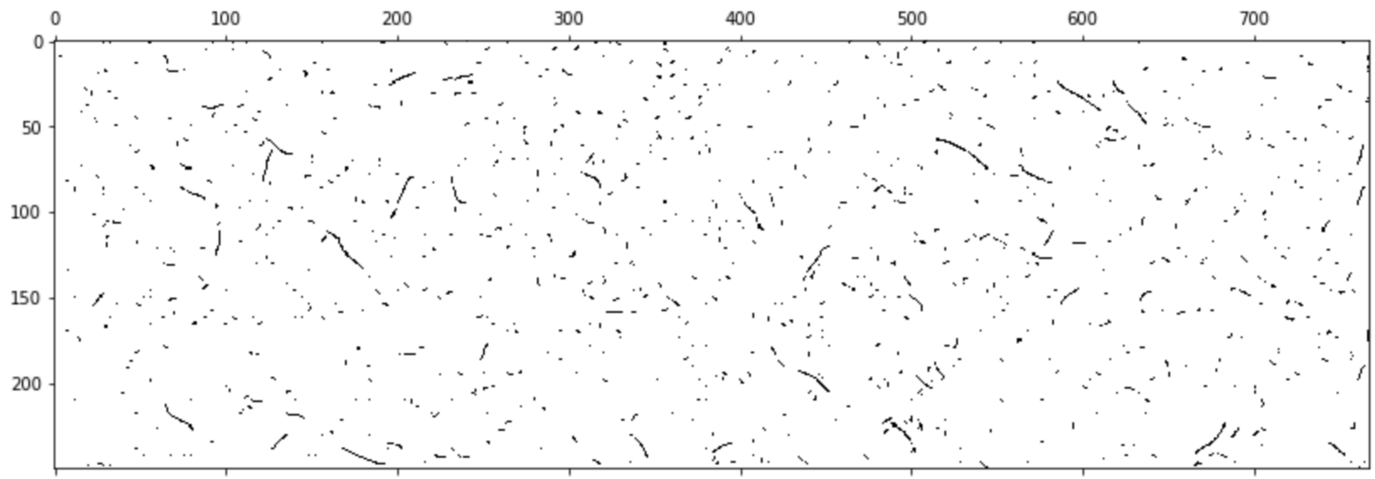
Modified BigGAN-deep



Colour-reversed real (simulated) image



Colour-reversed generated image



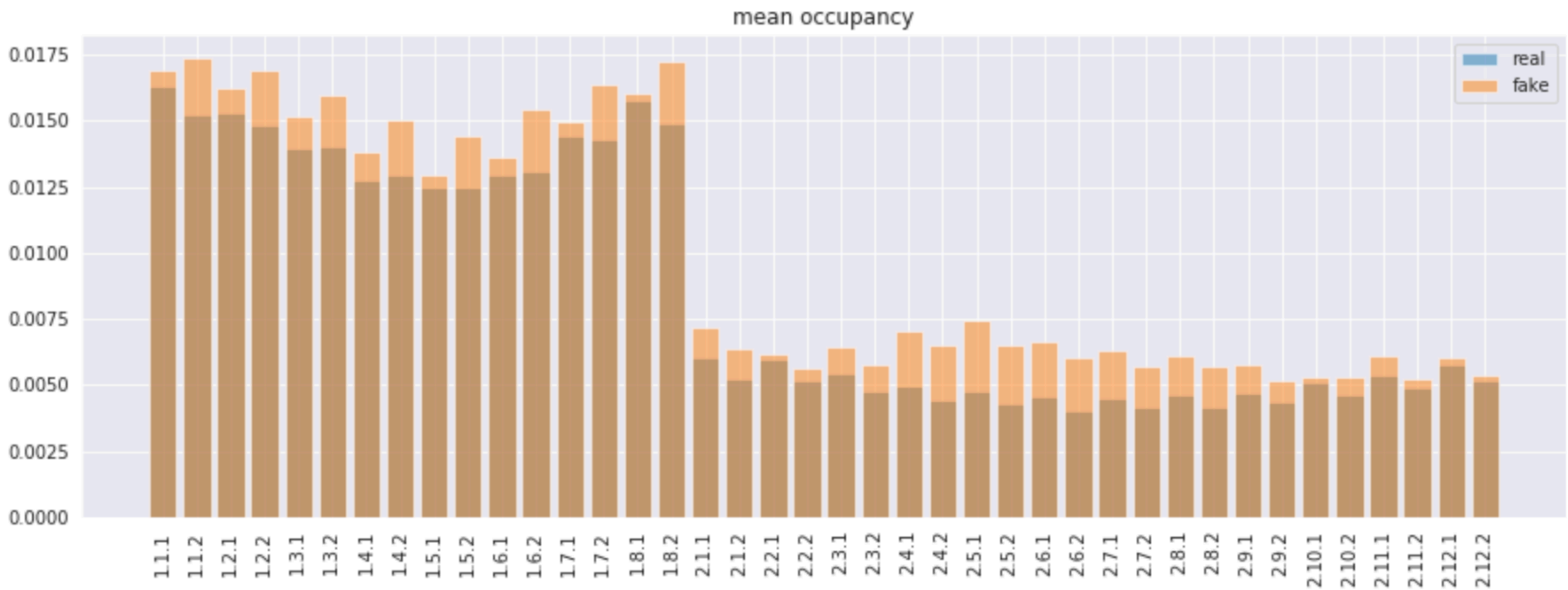
Validation of generated PXD images

Problem:

Finding a <u>metric</u> to say how good the generated images are.

Solutions:

- [x] Compare the occupancy information for PXD (updated)



Summary and Outlook

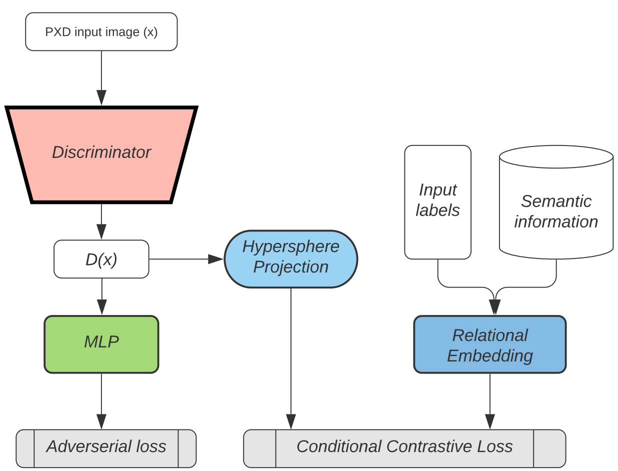
- Successful proof of principle that conditional GANs can be used to generate sensor-dependent PXD background
- Refine the GAN setup in order to capture ***correlation*** between two layers or each sensor relative to each other for PXD detector.
- Adding bkg types as colour channels to the images.
- Create a custom ***Inception Score (IS)***, based on simulated events in order to have a fully automated evaluation metric.
- Doing a comprehensive validation of generated hitmaps by estimating the systematic uncertainty on the tracking efficiency, fake rate and resolution.
- Simulation Software implementation.

Thank You

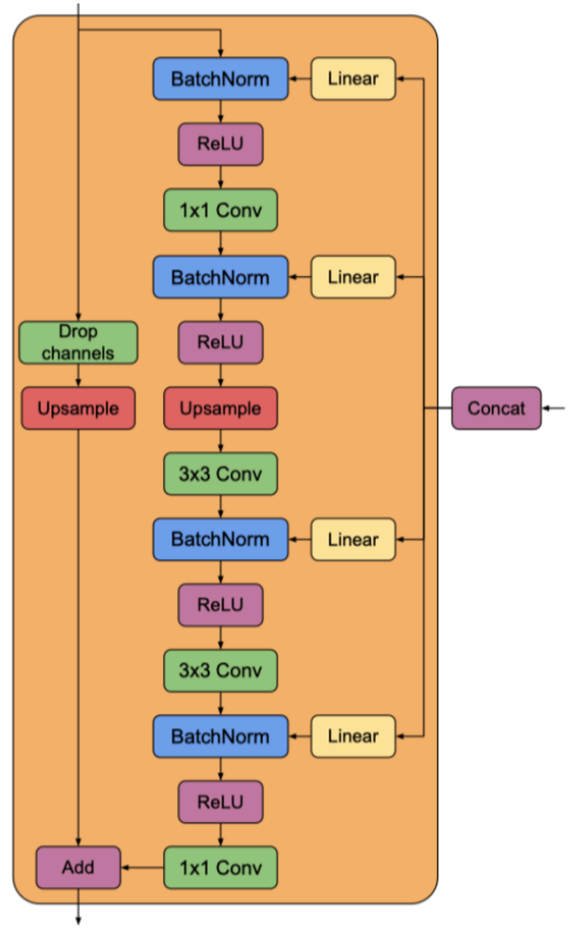
Back up Slides

☑ The Base Model: *[BigGAN-deep (updated)]*

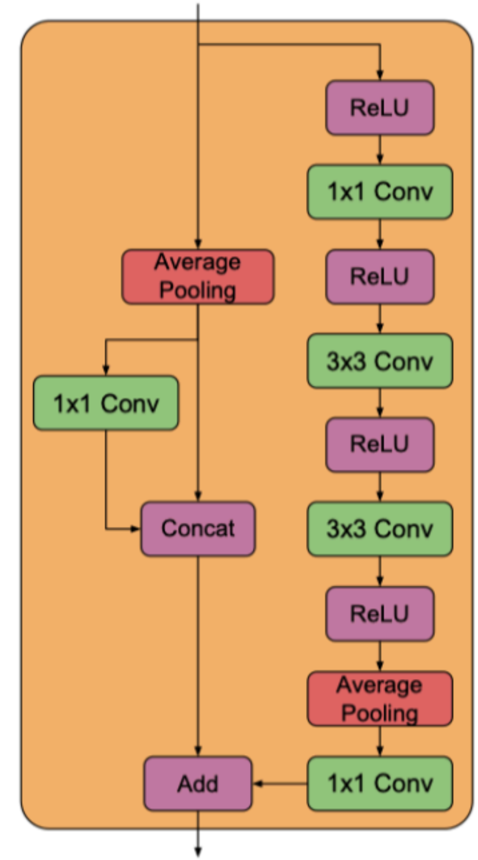
- **Technologies:**
  - *Self-Attention Block*
  - *Residual blocks*
  - *Spectral Normalisation*
  - *Orthogonal Weight init.*
  - *Orthogonal regularisation*
  - *Contrastive Learning*
  - *Hinge Loss*
  - *Consistency Regularisation*



ResBlock up in G



ResBlock down in D



Back up Slides

- The fired pixels are only read out if their value exceeds a threshold, 7 ADU.
- How to capture this prior information about the image?
- Solution: To add this information to the training

Algorithm 1 Pixel-Aware regularization.

Input: generator and discriminator parameters θ_G, θ_D, pixel-aware regularization coefcient λ, Adam hyperparameters α, β_1 , β_2, batch size M, number of discriminator iterations per generator iteration N_D

for number of training iterations **do**
    for $t = 1, ..., N_D$ **do**
        for $i = 1, ..., M$ **do**
            sample $z \sim p(z),\ x = p_{data}(x)$
            $L_D^{(i)} \leftarrow D[G(z)] - D(x)$
        end for
        $\theta_D \leftarrow Adam(\frac{1}{M}\sum_{i=1}^{M}(L_D^{(i)}), \alpha, \beta_1, \beta_2)$
    end for
    sample $\{z^{(i)}\}_{i=1}^{M} \sim p(z)$
    $x_{fake} = G(z)$
    $F[G(z)] : x_{fake} \mapsto x_{fake}^{cutoff}$ ▷ Threshold wrt. the pixel constraints.
    $L_{pr}^{(i)} \leftarrow ||G(z) - F[G(z)]||^2$
    $L_G^{(i)} \leftarrow -D[G(z)]$
    $\theta_G \leftarrow Adam(\frac{1}{M}\sum_{i=1}^{M}(L_G^{(i)} + L_{pr}), \alpha, \beta_1, \beta_2)$
end for

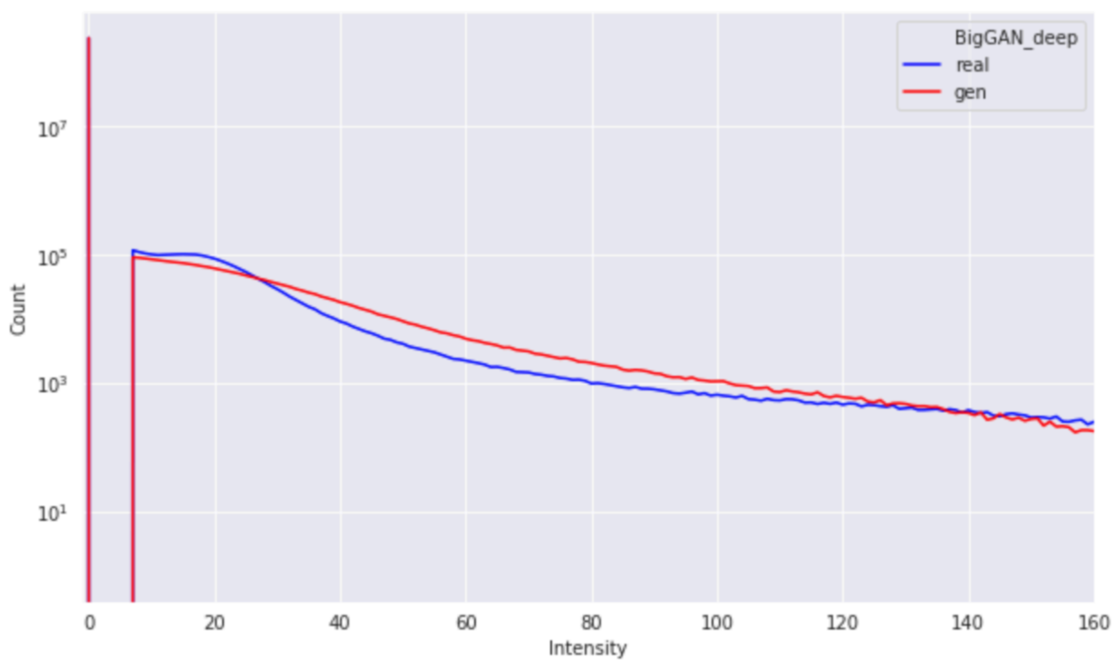
Validation of generated PXD images

Problem:

Finding a metric to say how good the generated images are.

Solutions:

- Image pixel intensity analysis:
  - Pixel value 0 means complete blackness
  - The fired pixels are only read out if their value exceeds a threshold, 7 ADU.
  - How to capture this prior information about the image?



Back up Slides

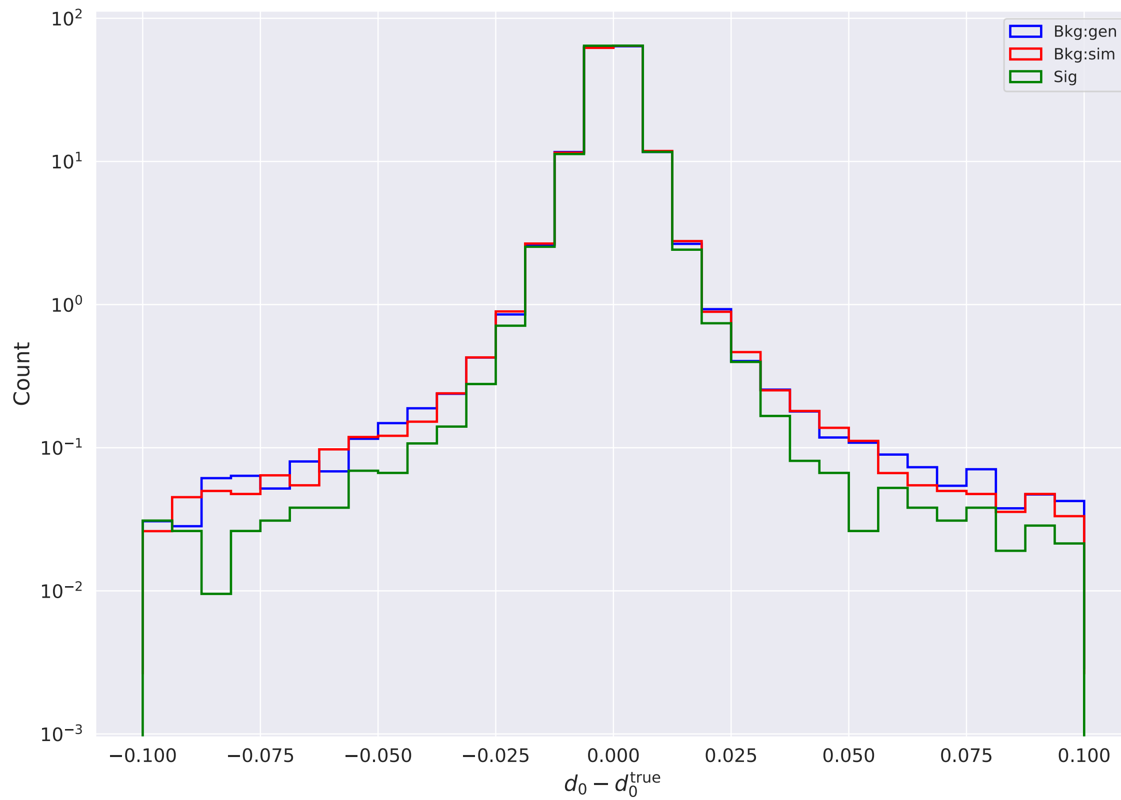
- Evaluate tracking performance for
  - Signal + no bkg.
  - Signal + nominal bkg.
  - Signal + generated bkg.

- **Scoring**: *Using Frechet Distance (2-Wasserstein distance):*

$$W_2(\mu_1, \mu_2)^2 = \|m_1 - m_2\|_2^2 + \mathrm{trace}\left(C_1 + C_2 - 2\left(C_2^{1/2} C_1 C_2^{1/2}\right)^{1/2}\right).$$

for $\mu_1 = N(m_1, C_1)$ and $\mu_2 = N(m_2, C_2)$.

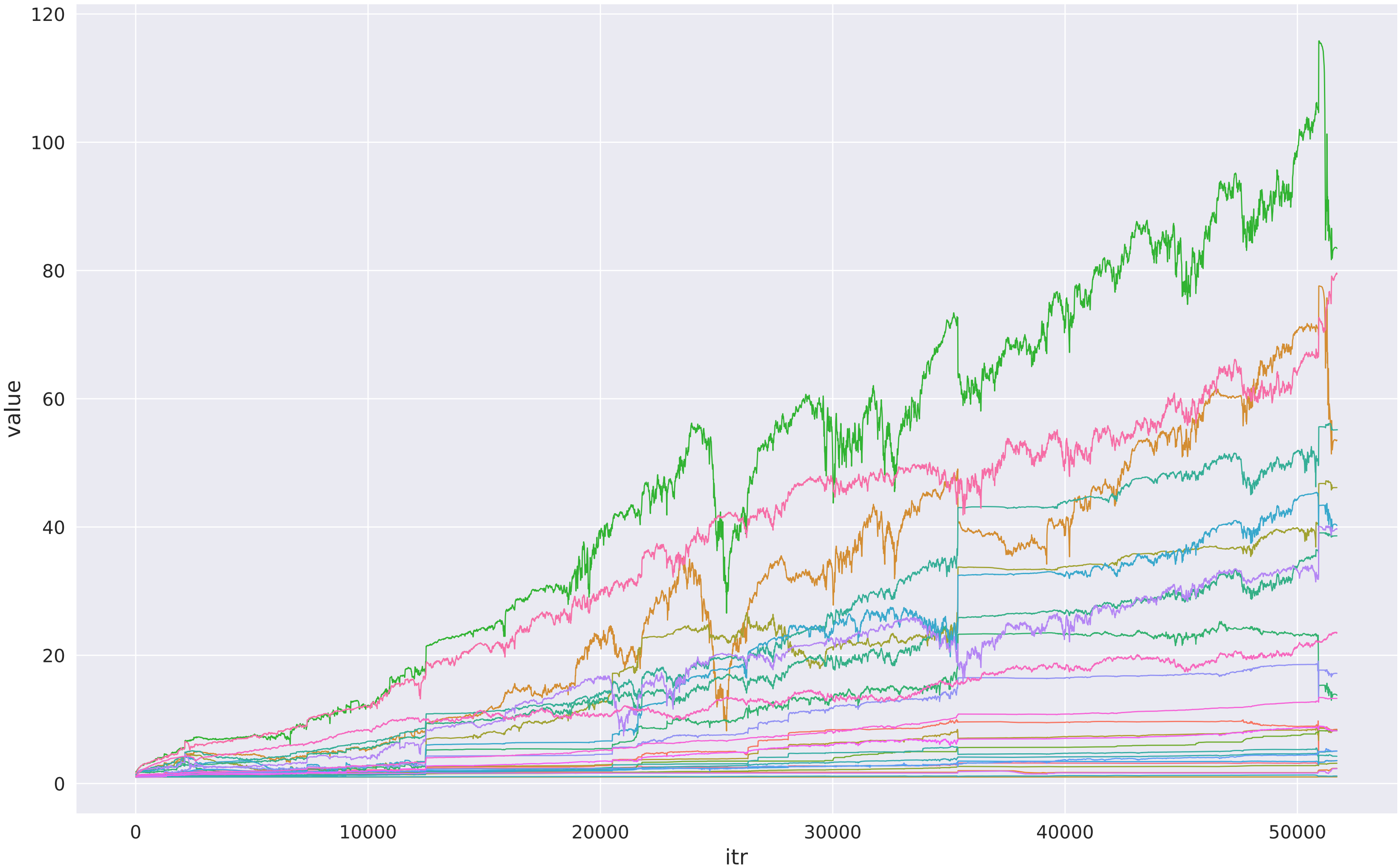
The Lower the FD score, the better the image quality and diversity from the physics point of view.



FD scores of between:

A. sim-sig : 5.42e-4
B. gen-sig : 7.41e-4
C. gen-sig : 1.64e-5

An Example of Collapse



A Healthy Model