LUDWIG-MAXIMILIANS-UNIVERSITY MUNICH

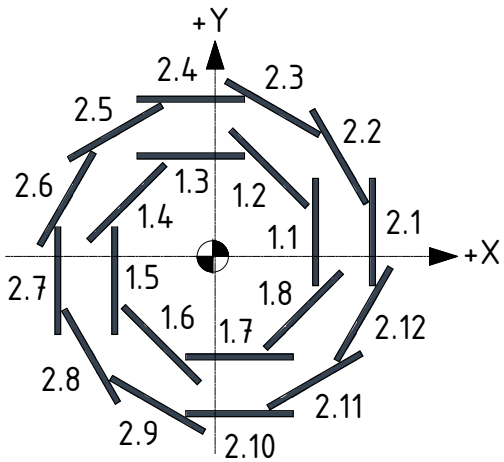
# Generating PXD Background Hit Maps with Generative Adversarial Networks at Belle II

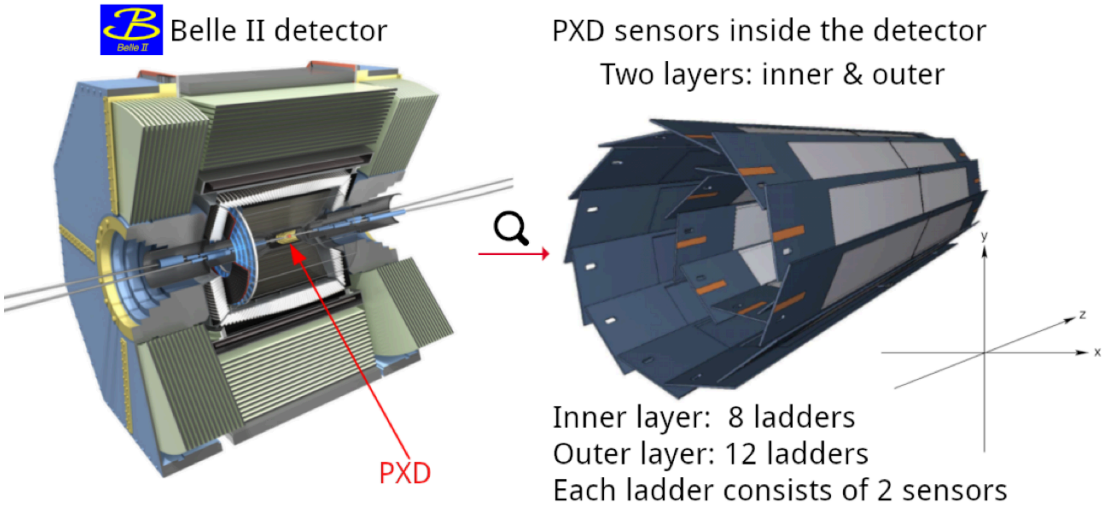
*Matej Srebre, Thomas Kuhr, Martin Ritter, Hosein Hashemi*

Bundesministerium für Bildung und Forschung

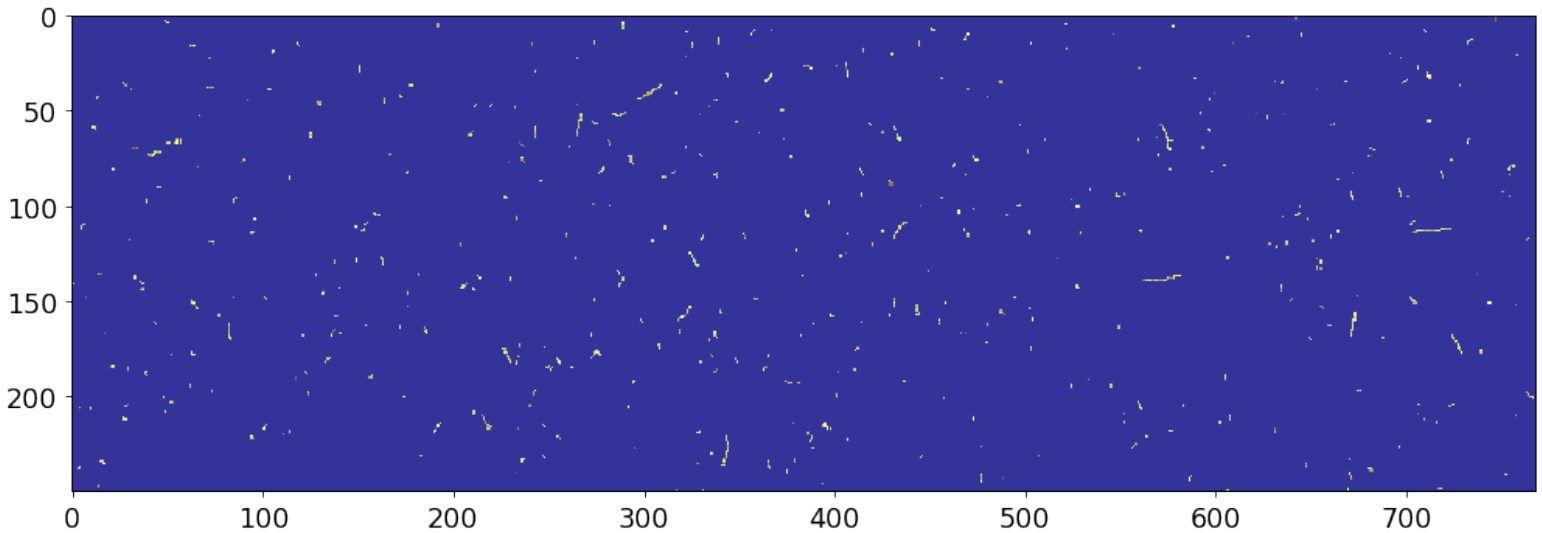
# Introduction

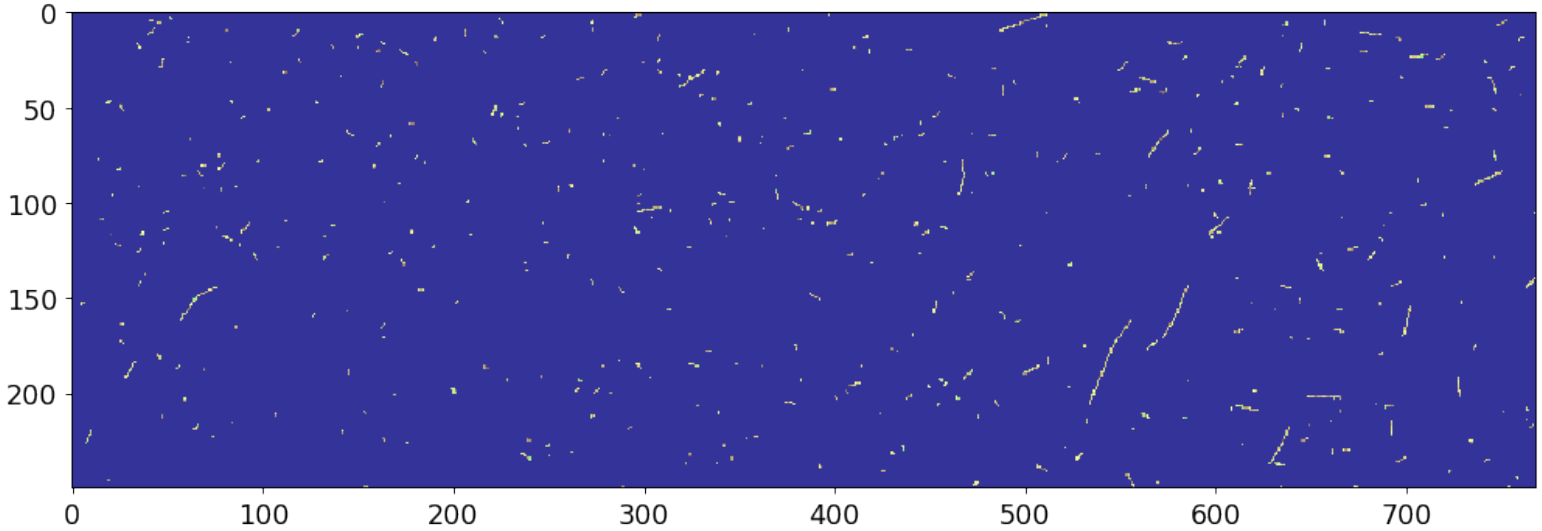
- ***The Pixel Vertex Detector (PXD)*** is the innermost semi-conductor sub-detector at Belle II.
- The sensitive area of the PXD is assembled from ***40 modules***, where each module consists of a ***250*** $\times$ ***768*** pixel matrix of the pixel sensors.
- *The PXD consists of two layers of pixel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



# Simulated PXD images



# Problem

- ***These tracks are coming from two sources:***
  - Decays that we want to analyse
  - Unwanted processes with respect to the analysis
- **Realistic detector simulation has to take into account effects from background processes**
  - *Simulation requires many PXD images with statistically independent background*
  - *Highest storage consumption, almost 100 KB per event cost*
  - *Requires distributing over all sites where MC is produced*

# Problem

- ***These tracks are coming from two sources:***
  - Decays that we want to analyse
  - Unwanted processes with respect to the analysis
- **Realistic detector simulation has to take into account effects from background processes**
  - *Simulation requires* [illegible] *atistically independent background*
  - *Highest storage co* [illegible] *per event cost*
  - *Requires distributin* [illegible] *s produced*

**Solution:**

Generating the bkg on the way of analysis instead of storing them

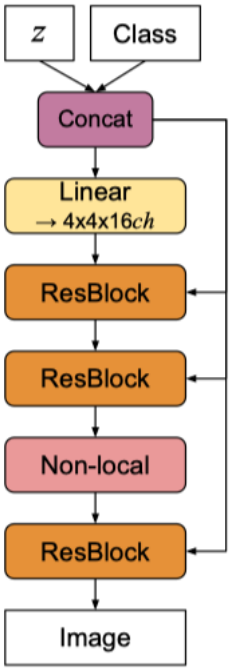
# Solution

- Generate PXD background events with Generative Adversarial Network **(GAN)**
- **Used Models:**

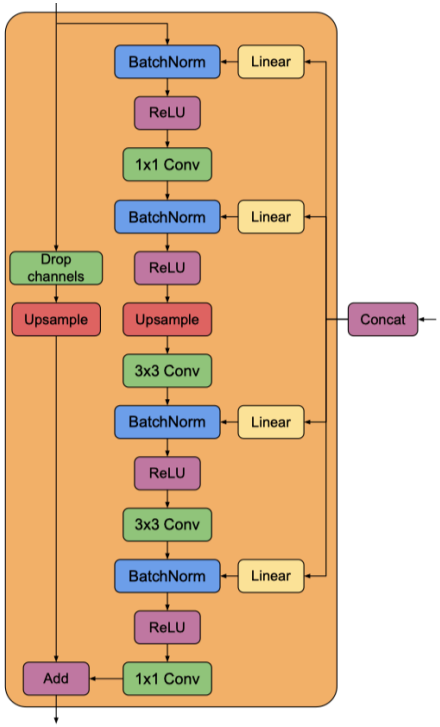
**DCGAN → WGAN-gp → LOGAN (LO-cWGAN) → BigGAN-AC → BigGAN-deep**

- **Technologies:**
  - *Self-Attention Module*
  - *Hinge Loss*
  - *Spectral Norm*
  - *Orthogonal regularisation*
  - *"Skip Connections" from noise*
  - *Orthogonal Weight init.*
  - *Latent Optimisation*
  - *Truncation trick (for BigGAN-AC)*

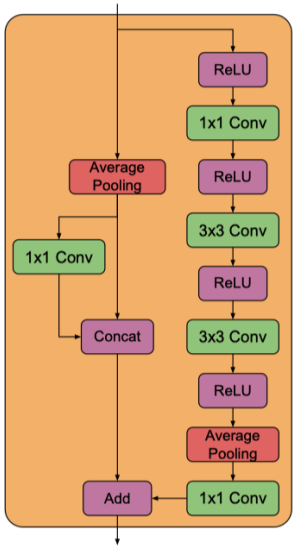
BigGAN-deep




ResBlock up in G

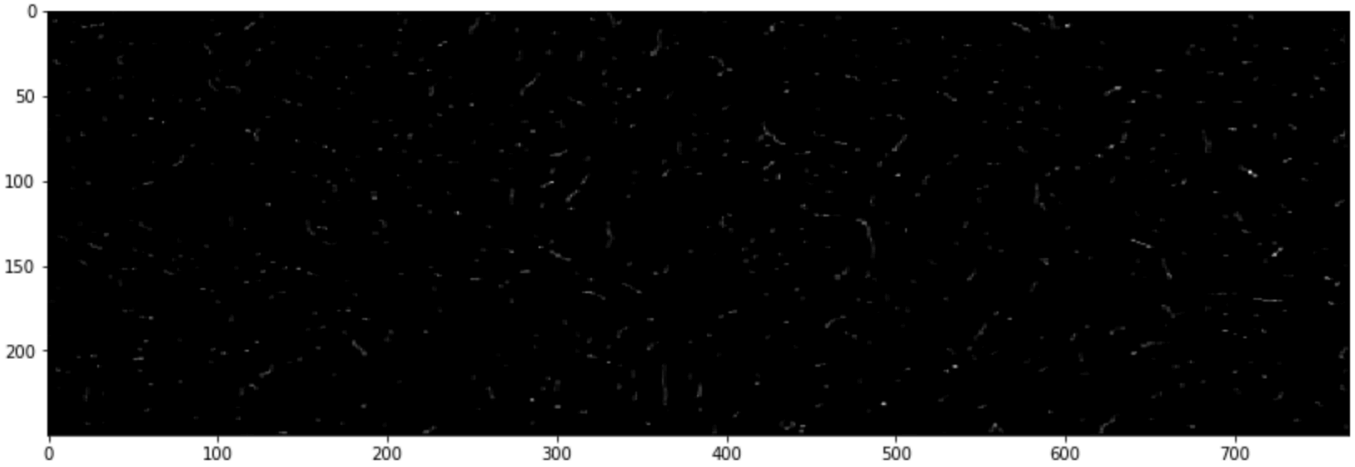



ResBlock down in D

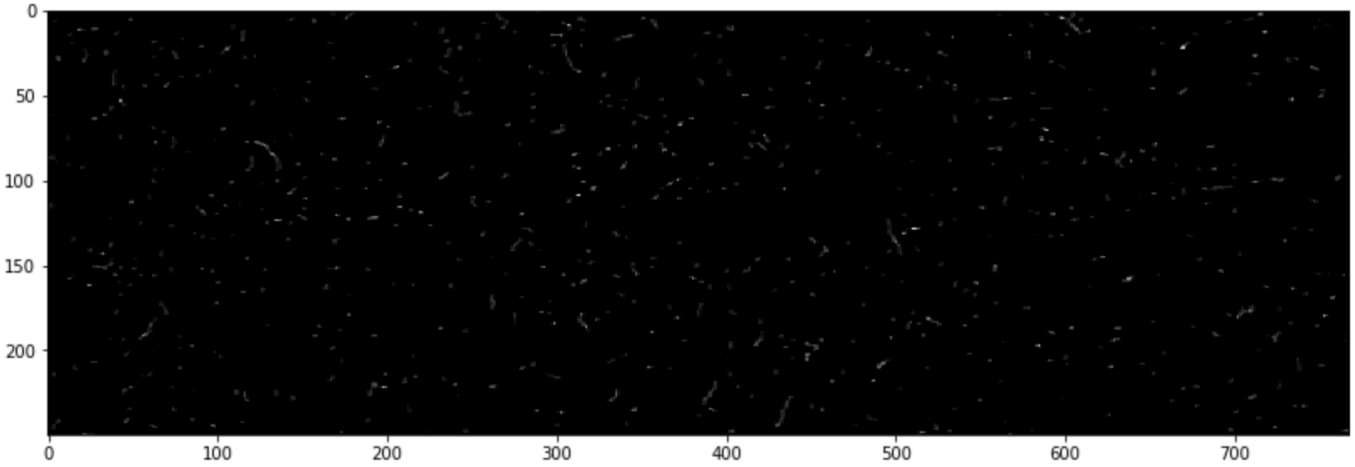


# Simulated vs Generated

Simulated Image




Generated Image



# Validation of generated PXD images

## *Problem:*

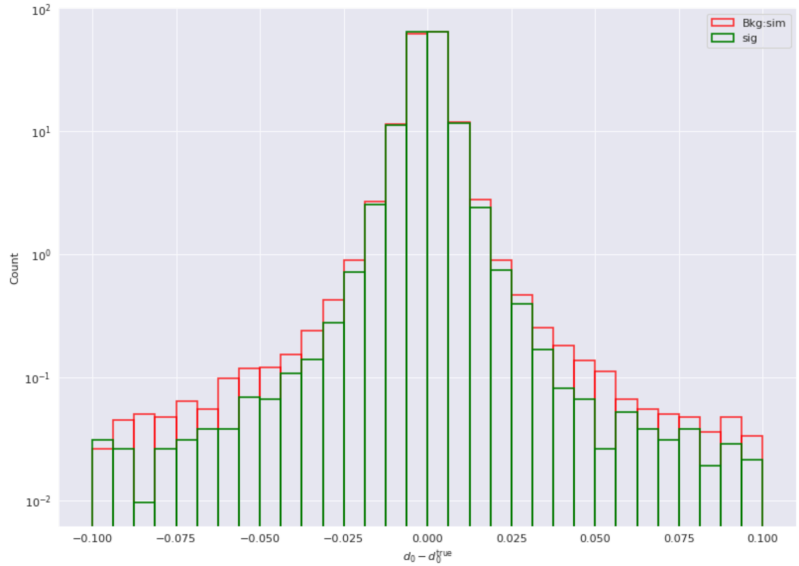
Missing a metric to say how good the generated images are.

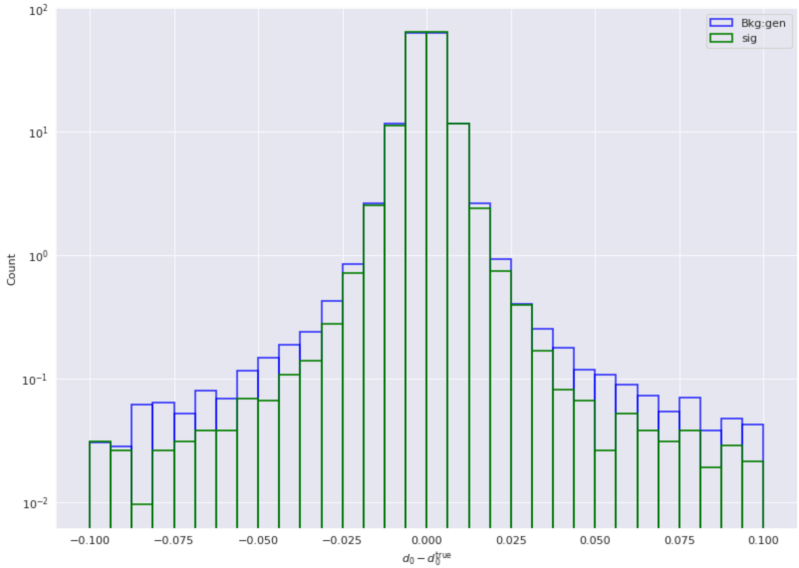
## *Solutions:*

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

  And compare:
    - Impact Parameter resolution: *Modified Frechet distance*
    - Track reconstruction efficiency

# Validation of generated PXD images for $d_0$ impact parameter



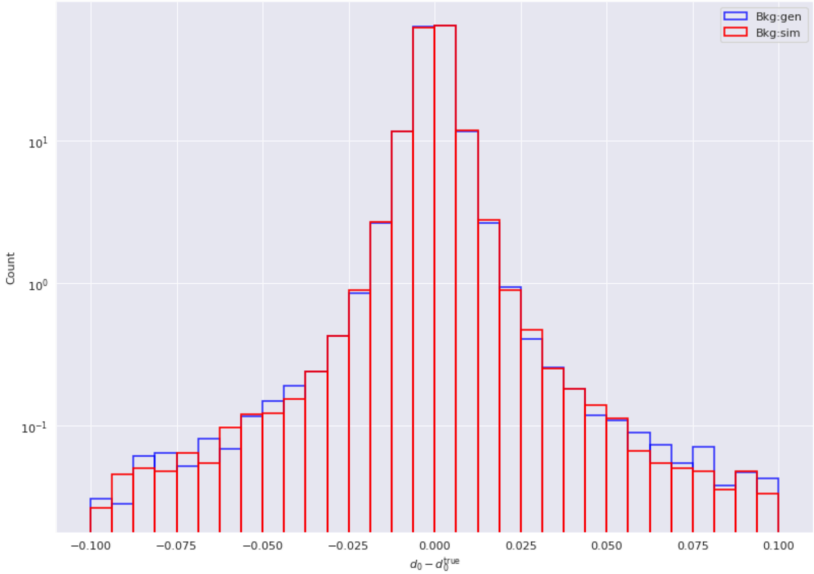




*FD scores of the new Model (**BigGAN-deep**) between:*

***A. sim-sig : 5.42e-4***

***B. gen-sig : 7.41e-4***

***C. gen-sim : 1.64e-5***




*FD scores of the old Model (WGAN-gp) between:*

*A. sim-sig : 5.42e-4*

*B. gen-sig : 8.01e-4*

*C. gen-sim : 1.79e-5*

# Outlook

- Refine the GAN setup in order to capture ***correlation*** between two layers of PXD detector like a *motion picture generation* using *sequence based GAN models.*
- Create a custom ***Inception Score (IS)***, based on simulated events in order to have a fully automated validation metric.
- Checking the performance of Variational Auto Encoders (the greatest enemy of GANs!)
- Simulation Software implementation.

# Thank You

# References


- *LOGAN (BigGAN-deep): "https://arxiv.org/abs/1912.00953"*
- *LOGAN implementation: "https://github.com/Hosein47/LOGAN"*
- *BIGGAN: "https://arxiv.org/abs/1809.11096"*
- *WGAN-gp: "https://arxiv.org/abs/1704.00028"*
- *DCGAN: "https://arxiv.org/abs/1511.06434"*