LUDWIG-
MAXIMILIANS
UNIVERSITY
MUNICH

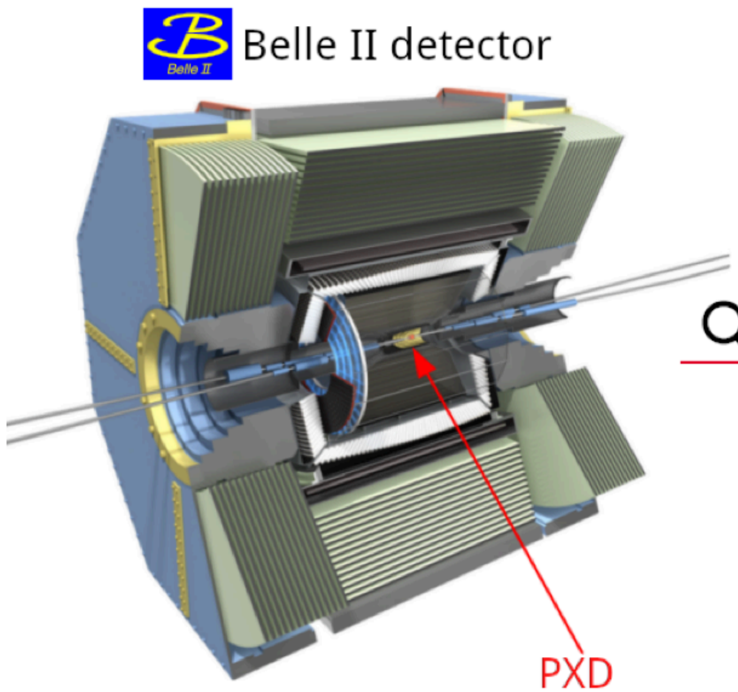
IEA-GAN: Intra-Event Aware GAN with Relational Reasoning for Efficient High-Resolution Detector Simulation

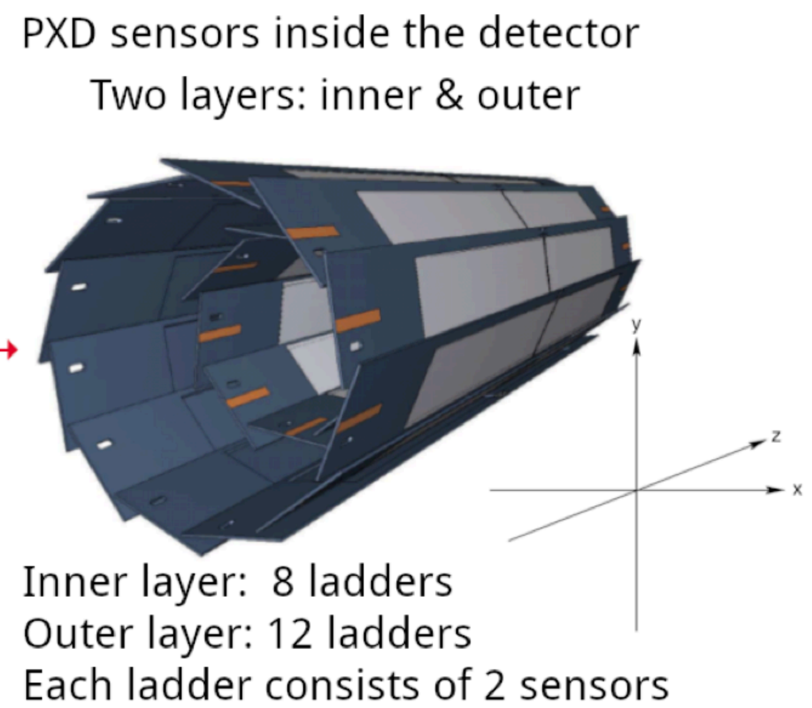
Hosein Hashemi [1], Nikolai Hartmann [1], Sahand Sharifzadeh [2], James Kahn [3], Thomas Kuhr [1]

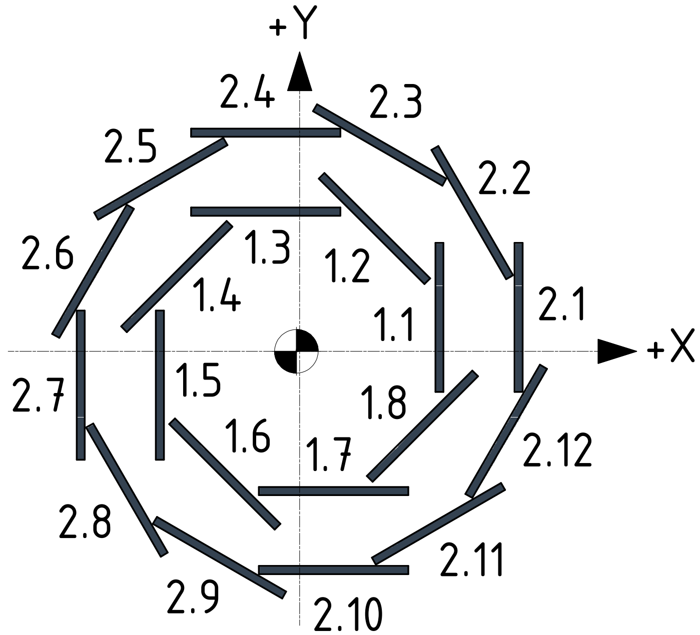
Ludwig-Maximilians-Universität München [1,2]
The ORIGINS Excellence Cluster [1]
Helmholtz AI [3]
DeepMind [2]

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





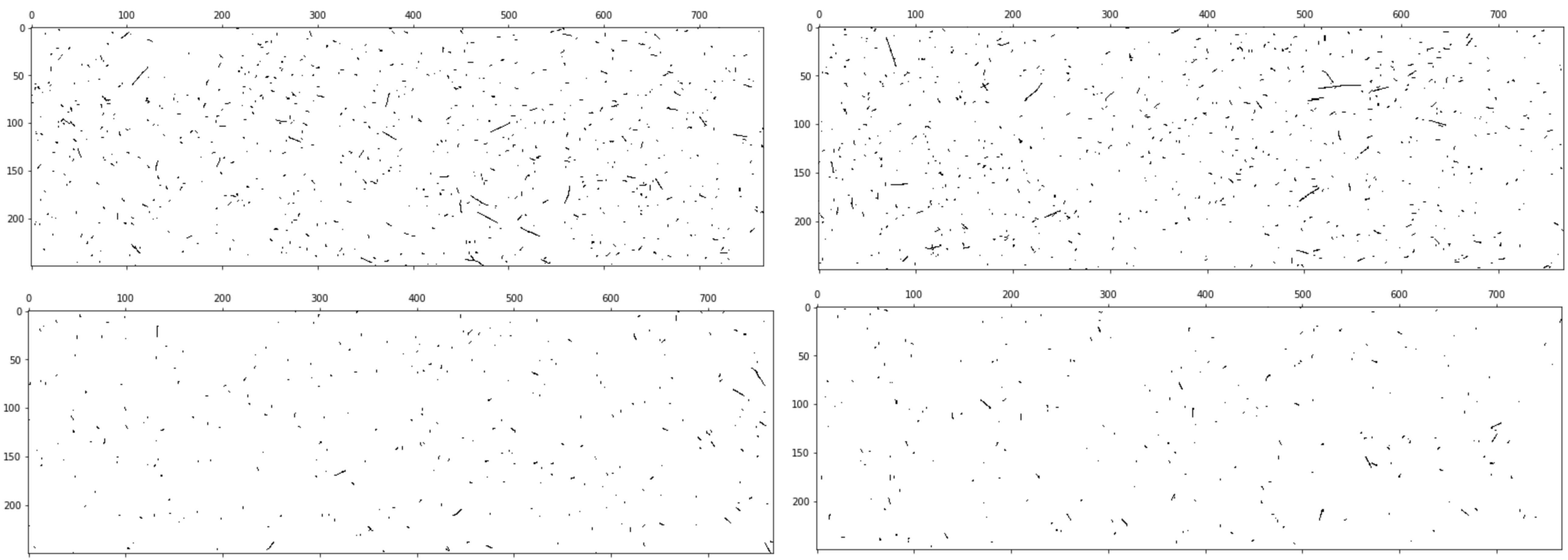


Backgrounds

- **The PXD hits are coming from two sources:**
  - **Signal Decays:** Involve on average less than 1 percent hits per sensor
  - **Backgrounds:** Majority of hits

- **Objective:** Faithful PXD background simulation based on different sensors

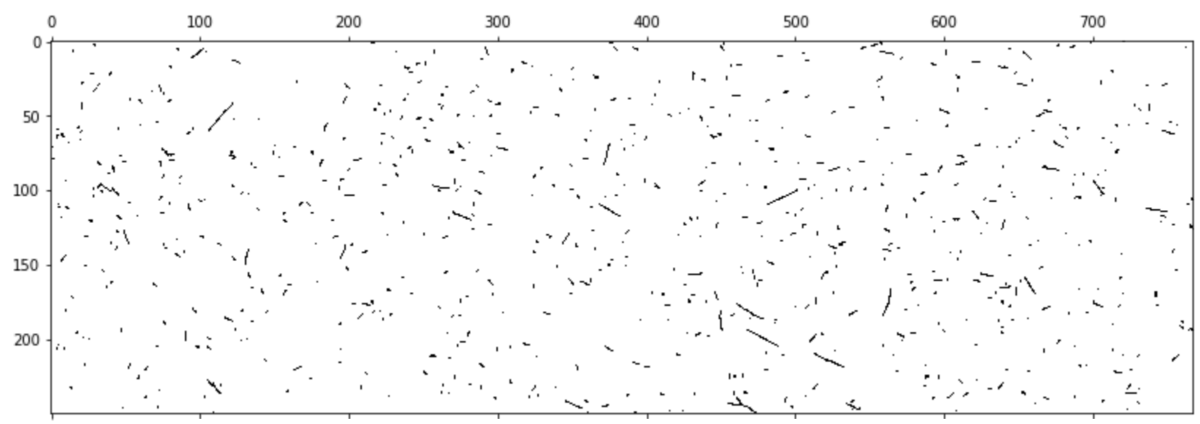
Colour-reversed real (simulated) image

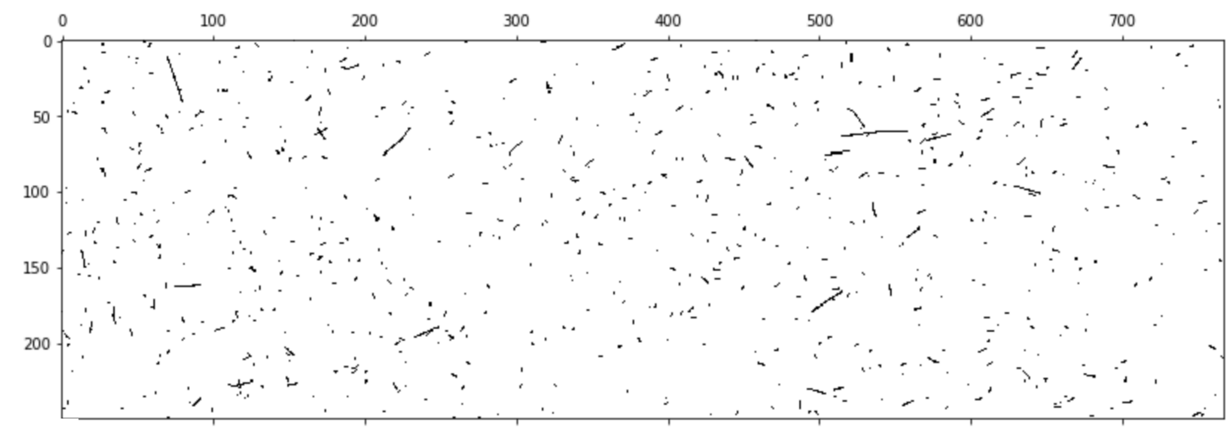


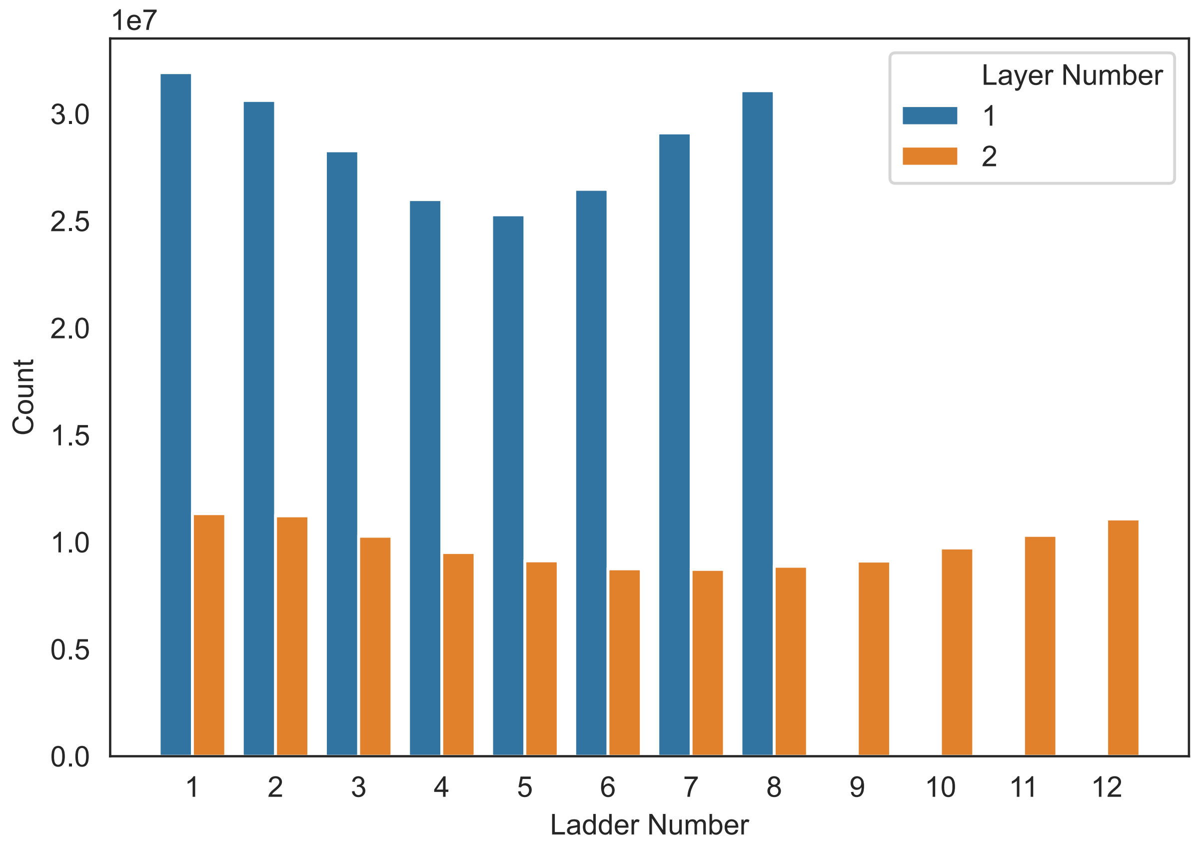
Beam direction

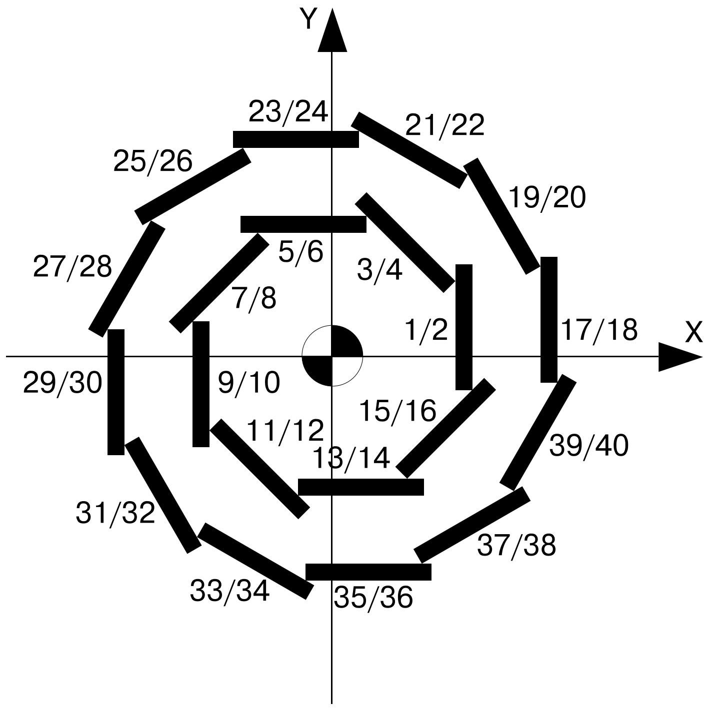
Motivation

- **Using spatial class-conditions based on the sensor number 1-40:**
  - ☑ Training Data: GEANT4 Simulated beam background events
  - ☑ Objective: **Generation of sensor-dependent, High Resolution PXD images**



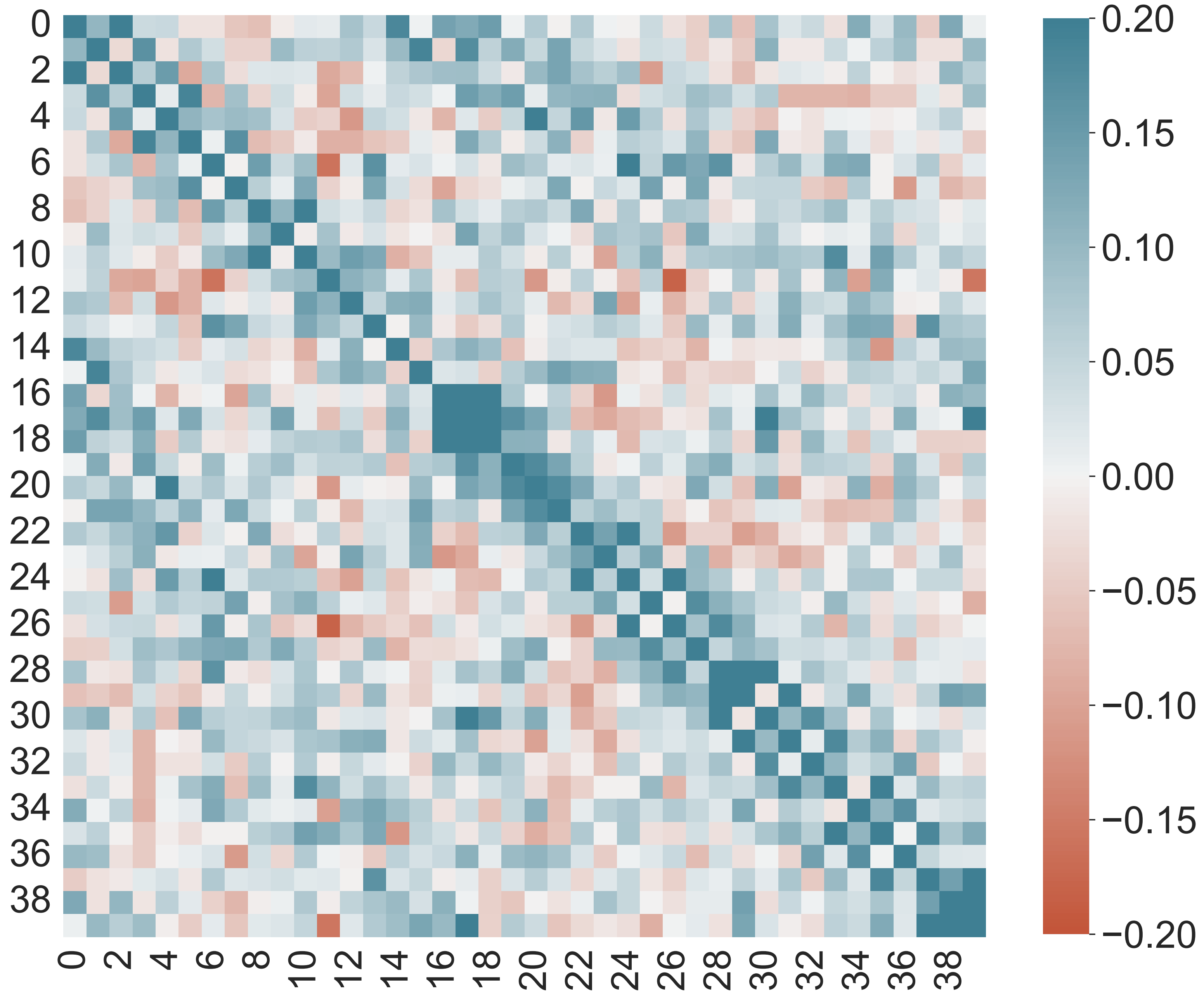


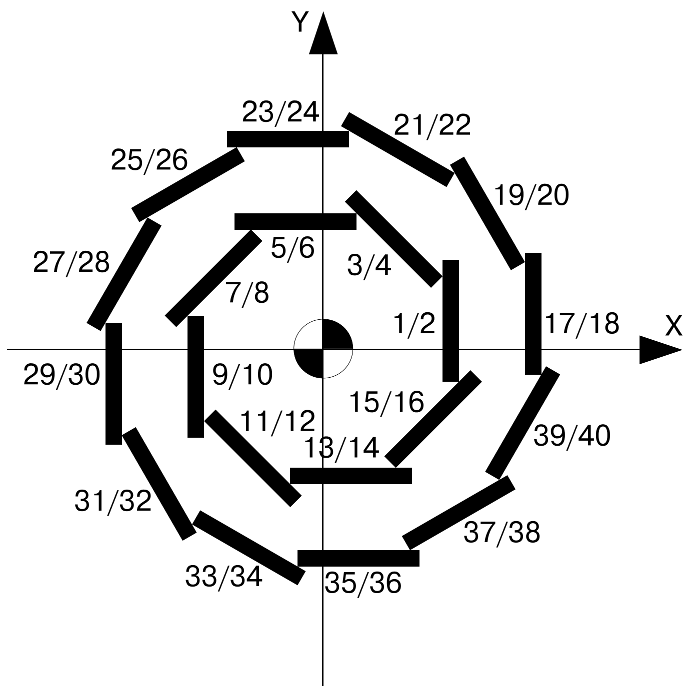




Motivation

- **Using spatial class-conditions based on the sensor number 1-40:**
  - ☑ Training Data: GEANT4 Simulated beam background events
  - ☑ Objective: **Generation of Fine-grained sensor-dependent, High Resolution, Correlated PXD images**





Fine-Grained Image Generation

- **class-conditional GAN :** The type of animal is the condition (class)

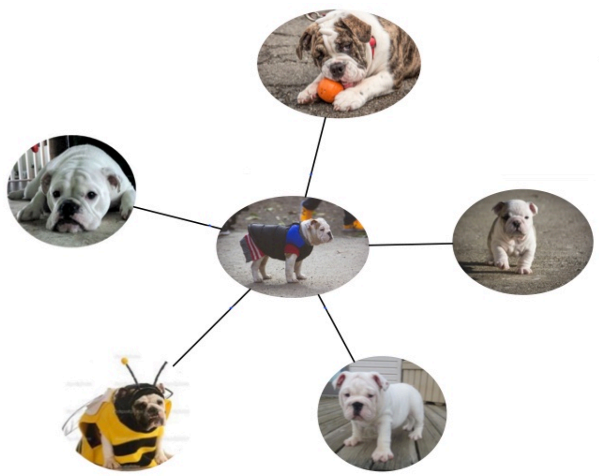
- **Fine-grained class-conditional image generation:**

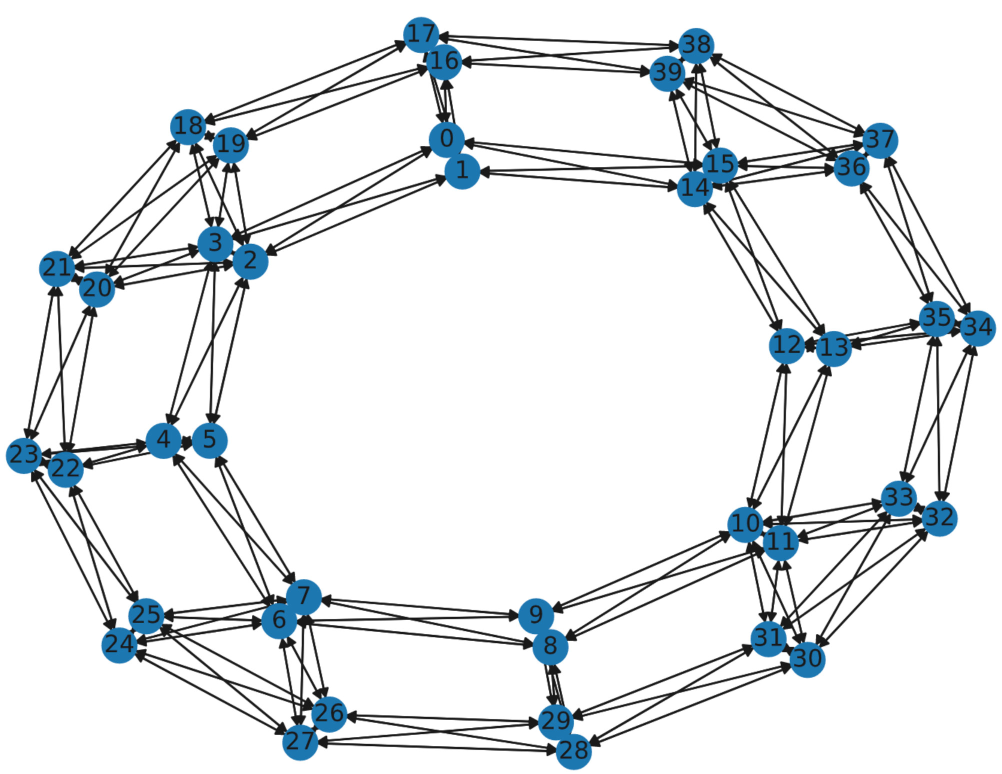
  A. The classes show both statistical and semantic similarity

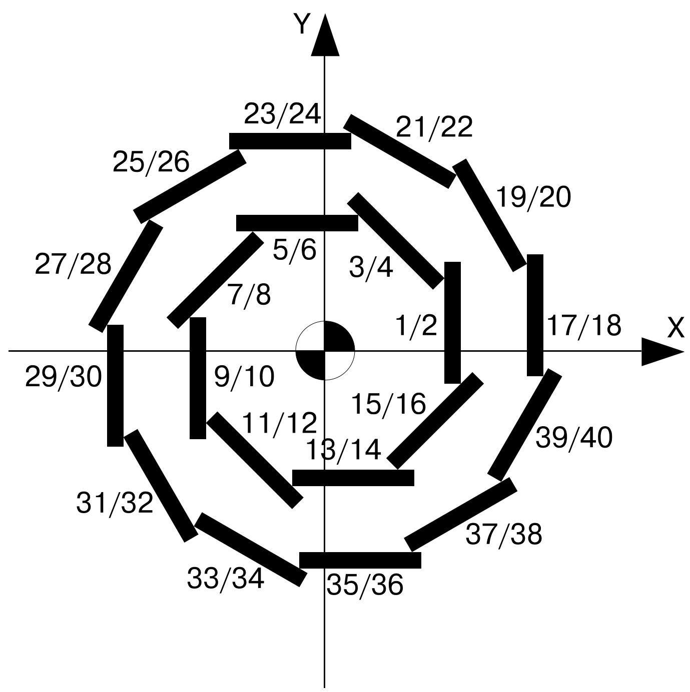
  B. Similar datasets: The Stanford Cars, iNaturalist

  C. The objective is to create objects from subordinate categories such as breeds of dogs or models of cars.

  D. The small inter-class and large intra-class variation inherent to fine-grained image analysis makes it a challenging problem.







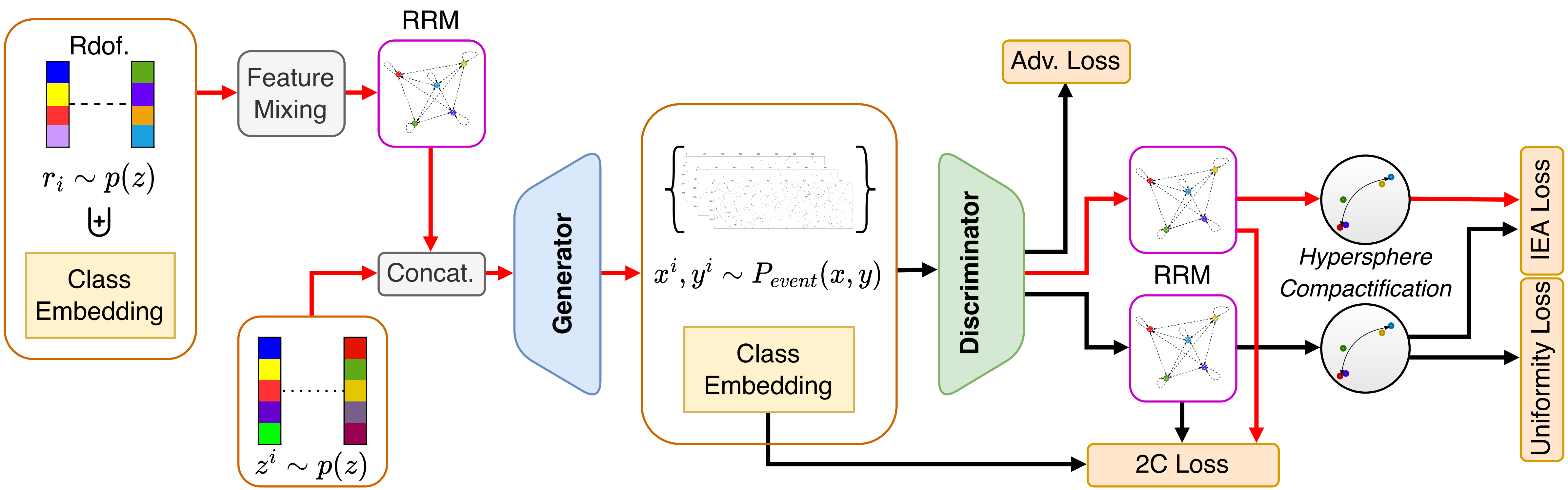
IEA-GAN Model (prologue)

- **How to approximate the concept of an "Event" while we simulate the detector response?**
  - **Proper Sampling:** Defining an image per class sampler (**generating event by event**) and shuffling within each batch (event).
  - **Intra-event relational reasoning:** Using a Relational Reasoning Module over an event to weight the importance of each sample with respect to each other. Thus, the model will understand the class-to-class relations in a single event.
  - **Maintaining the Discriminator's Information Entropy**
  - **Transferring Discriminator's Intra-event Knowledge to the Generator**

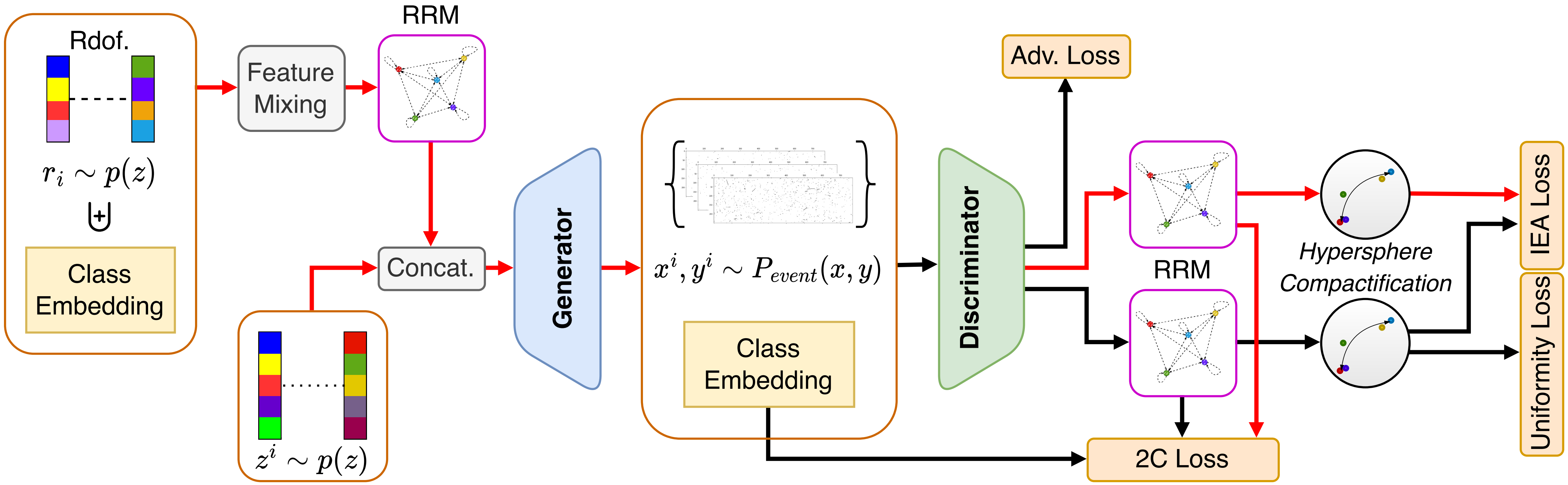
IEA-GAN Model (prologue)

- **How to approximate the concept of an "Event" while we simulate the detector response?**
  - ☑ **Proper Sampling:** Defining an image per class sampler (**generating event by event**) and shuffling within each batch (event).
  - ☑ **Intra-event relational reasoning:** Using a Relational Reasoning Module over an event to weight the importance of each sample with respect to each other. Thus, the model will understand the class-to-class relations in a single event.
  - ☑ **Maintaining the Discriminator's Information Entropy**
  - ☑ **Transferring Discriminator's Intra-event Knowledge to the Generator**

Deep Metric Learning + Knowledge Distillation + Permutation Equivariant Relational Inductive Bias = IEA-GAN



IEA-GAN Model (Generator)



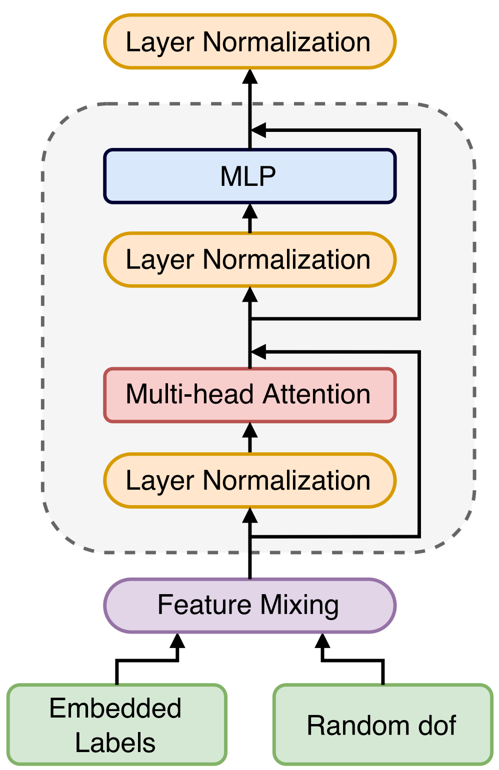
$$L_{gen} = L_{Adv} + \lambda_{2C} L_{2C} + \lambda_{IEA} L_{IEA}$$

Contrastive Loss: $$L_{2C}(x_i, y_i; t) = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{\exp(h(x_i)^T e(y_i)/t)}{\exp(h(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} \mathbf{1}_{k \neq i} \,.\, \exp(h(x_i)^T h(x_k)/t)} \right)$$

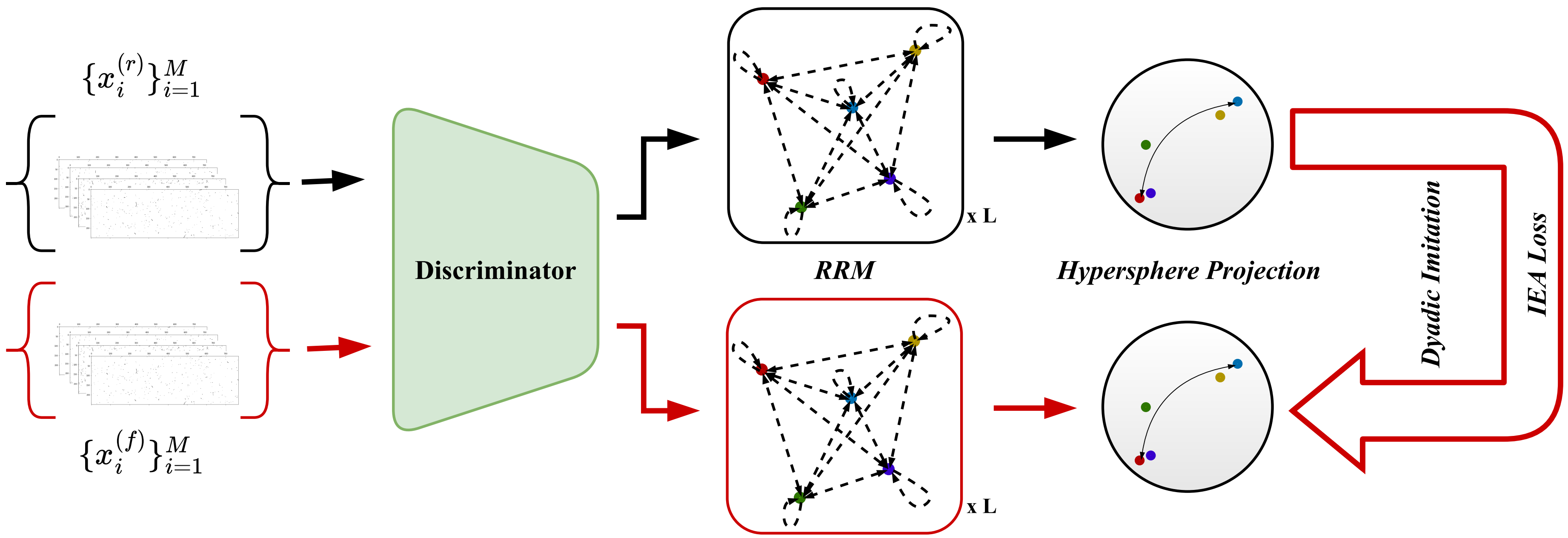
IEA Loss: $$L_{IEA}(x_f, x_r) = D_{KL}\Big(\sum_{i,j} \sigma(h(x_i^{(r)}) h(x_j^{(r)})^\top) \,|\, \sigma(h(x_i^{(f)}) h(x_j^{(f)})^\top) \Big)$$

$h(.)$: *Relational embedding*
$e(.)$: *proxy (class embedding)*
$\sigma(.)$: *Softmax function*
$x^{(f)}$: *generated images*
$x^{(r)}$: *real images*

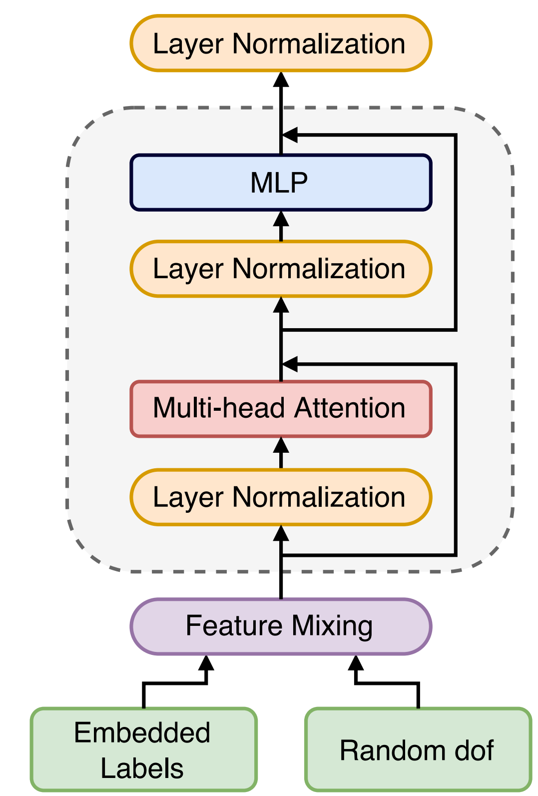
Relational Reasoning Module



IEA-GAN Model (Generator)



Relational Reasoning Module



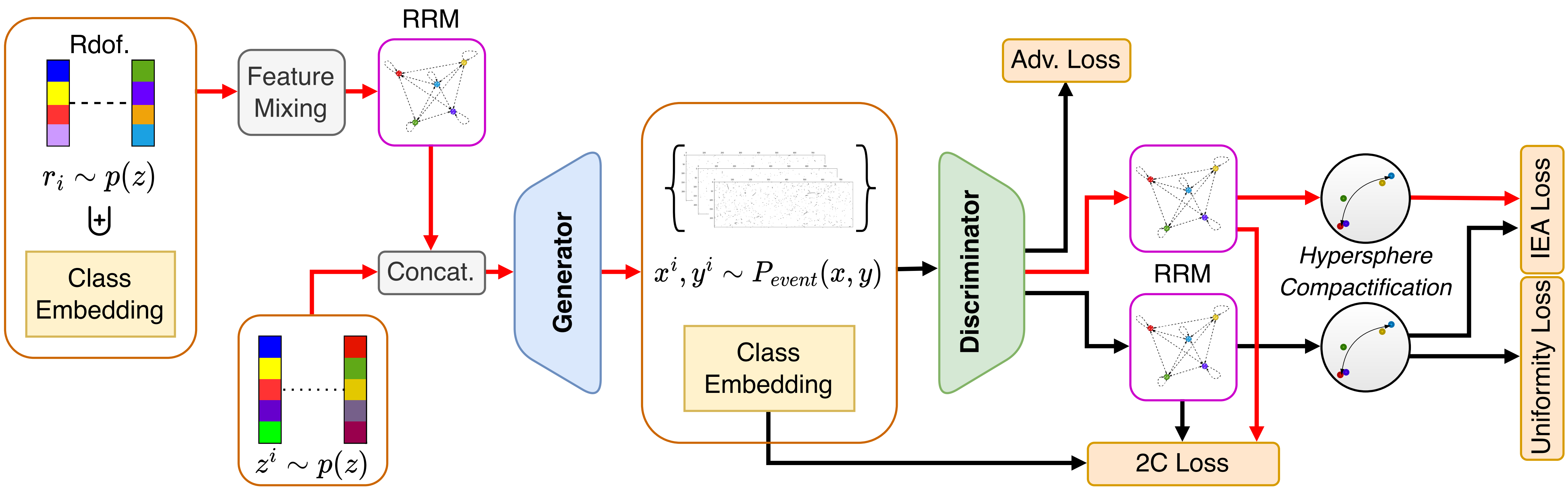
Hypersphere dimension: 128
MLP dimension: 128
Number of Heads: 2
Number of Layers: 1

IEA Loss: $$L_{IEA}(x_f, x_r) = D_{KL}\left(\sum_{i,j} \sigma(h(x_i^{(r)})h(x_j^{(r)})^\top) \,|\, \sigma(h(x_i^{(f)})h(x_j^{(f)})^\top)\right)$$

$h(.)$: *Relational embedding*
$e(.)$: *proxy (class embedding)*
$\sigma(.)$: *Softmax function*
$x^{(f)}$: *generated images*
$x^{(r)}$: *real images*

- **Upon minimising it, we are putting a self-supervised penalising system over the intra-event awareness of the the generator by encouraging it to look for more detailed connections among the images.**
- **In the end we want to maximise the agreement of data points on two unit hyperspheres of real image and generated image embeddings.**

IEA-GAN Model (Discriminator)



$$L_{dis} = L_{Adv} + \lambda_{2C} L_{2C} + \lambda_{uniform} L_{uniform}$$

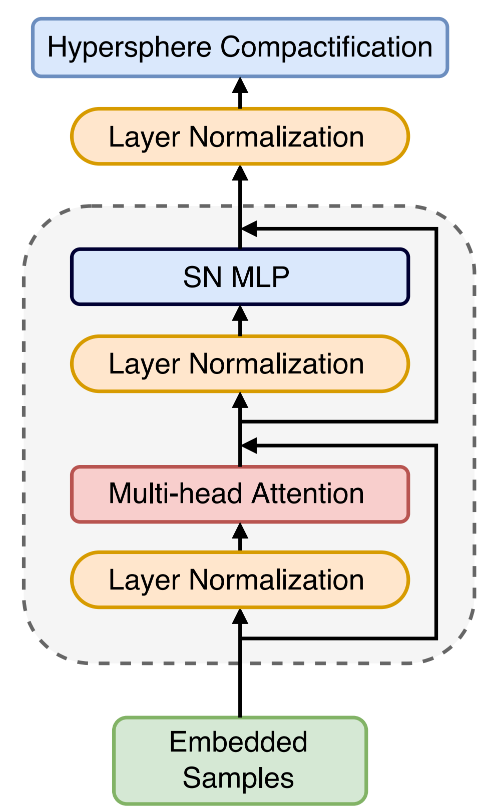
Contrastive Loss: $$L_{2C}(x_i, y_i; t) = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{\exp(h(x_i)^T e(y_i)/t)}{\exp(h(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} \mathbf{1}_{k \neq i} \,.\, \exp(h(x_i)^T h(x_k)/t)} \right)$$

Uniformity Loss: $$L_{uniform}(h; t) = \log \mathbb{E}_{x_i, x_j \sim p_{data}} [\exp(-t \| h(x_i) - h(x_j) \|_2^2)]$$

- **By imposing uniformity condition over the feature vectors on the unit hypersphere, they preserve as much information as possible since the uniform distribution carry high entropy.**

$h(.)$: *Relational embedding*
$e(.)$: *proxy (class embedding)*

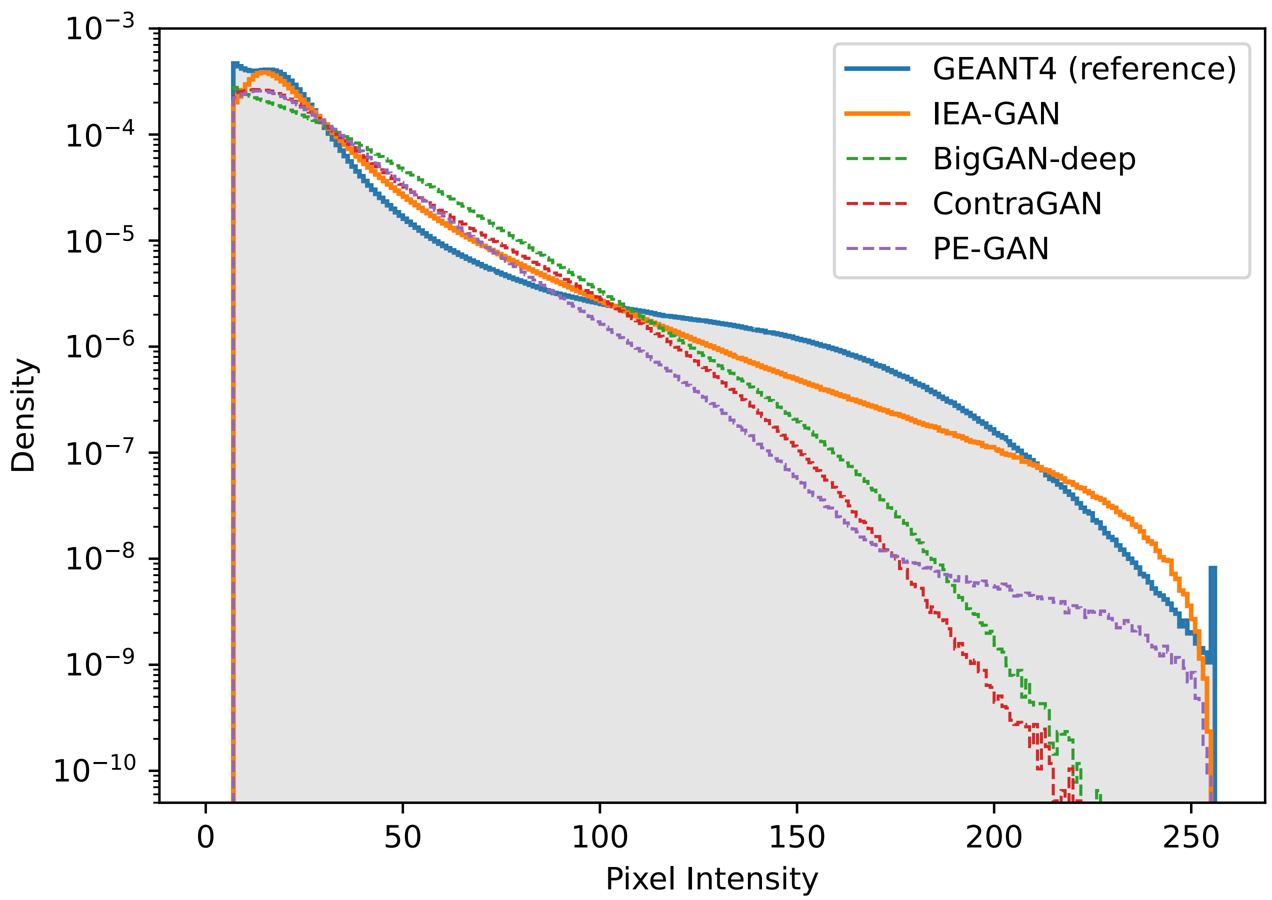
Relational Reasoning Module

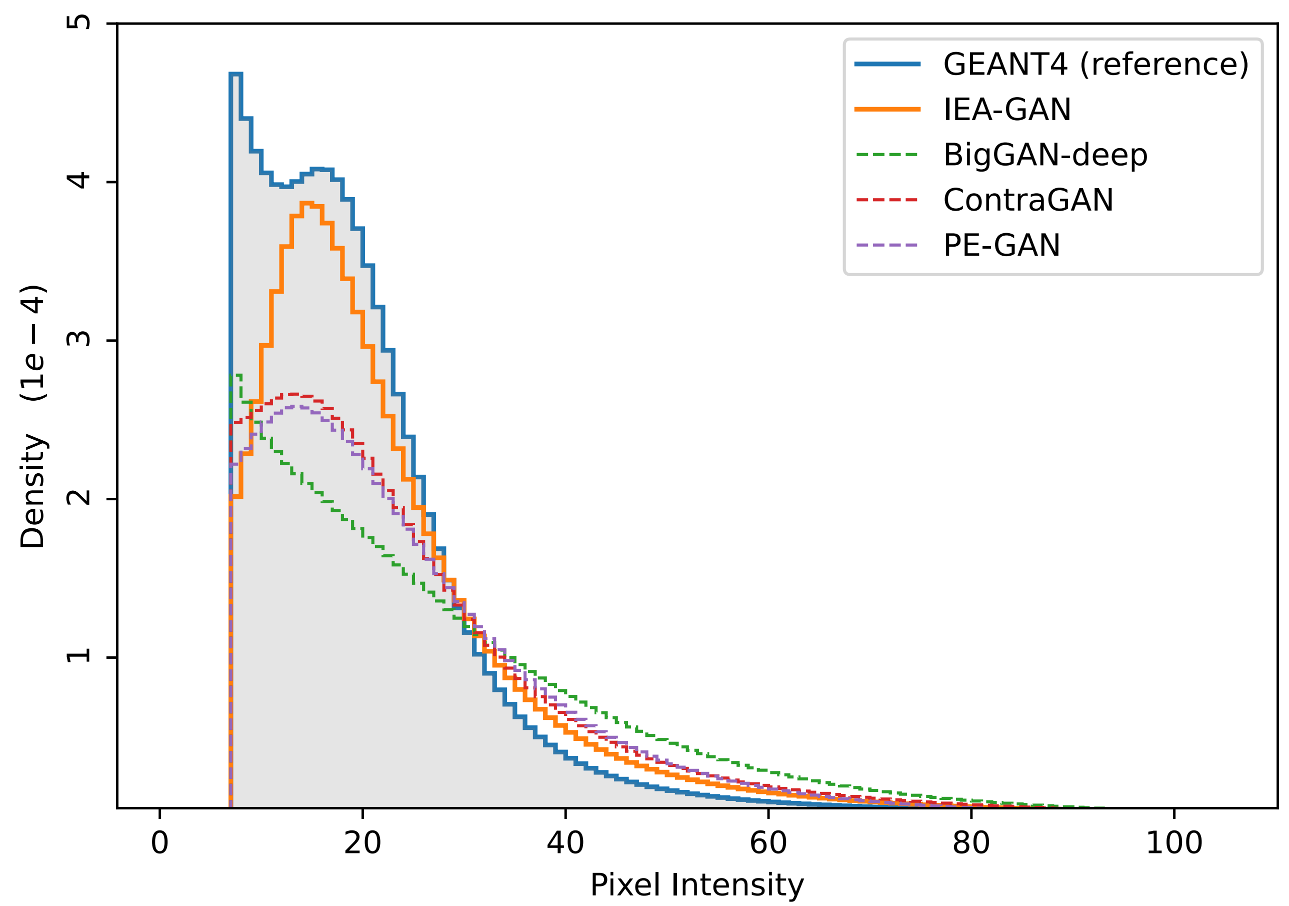


Hypersphere dimension: 1024
SN-MLP dimension: 512
Number of Heads: 4
Number of Layers: 1

Validation of generated PXD images

- **Validation Metrics over the test set in comparison to SOTA in High Resolution Image Generation:**
  - ☑ **Pixel Energy above the threshold:**

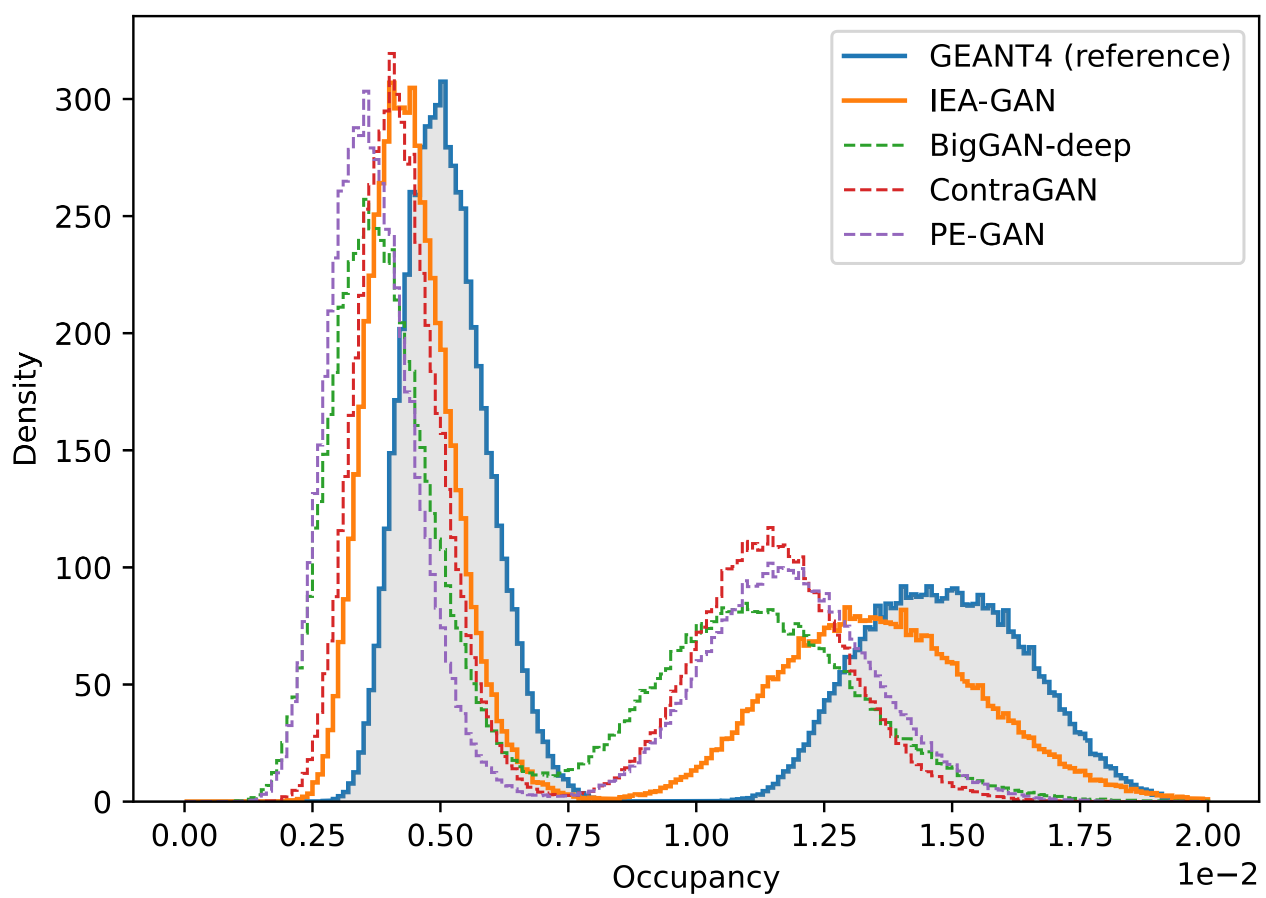


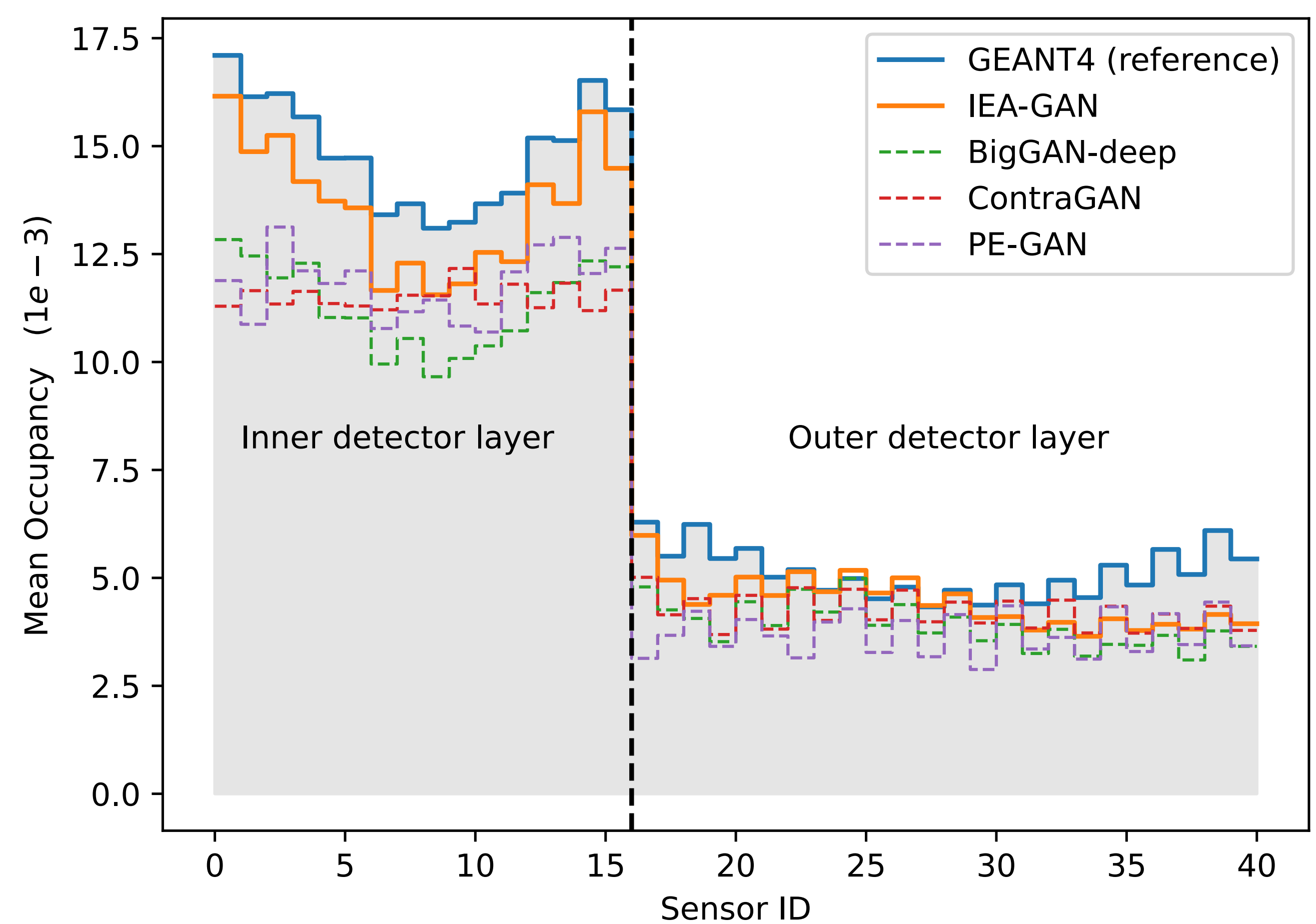


BigGAN-deep, Brock, A., Donahue, J., Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis, arXiv (2019). https://doi.org/10. 48550/arXiv.1809.11096 .
ContraGAN: Contrastive Learning for Conditional Image Generation. In: Advances in Neural Information Processing Sys- tems, vol. 33, pp. 21357–21369.
PE-GAN: Hashemi et al.: Pixel Detector Background Generation using Generative Adversarial Networks at Belle II. EPJ Web of Conferences 251, 03031 (2021). https://doi.org/10. 1051/ep jconf/202125103031

Validation of generated PXD images

- **Validation Metrics over the test set in comparison to SOTA in High Resolution Image Generation:**
  - **Occupancy Density and Mean Occupancy :**





BigGAN-deep, Brock, A., Donahue, J., Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis, arXiv (2019). https://doi.org/10. 48550/arXiv.1809.11096 .
ContraGAN: Contrastive Learning for Conditional Image Generation. In: Advances in Neural Information Processing Sys- tems, vol. 33, pp. 21357–21369.
PE-GAN: Hashemi et al.: Pixel Detector Background Generation using Generative Adversarial Networks at Belle II. EPJ Web of Conferences 251, 03031 (2021). https://doi.org/10. 1051/ep jconf/202125103031

Validation of generated PXD images

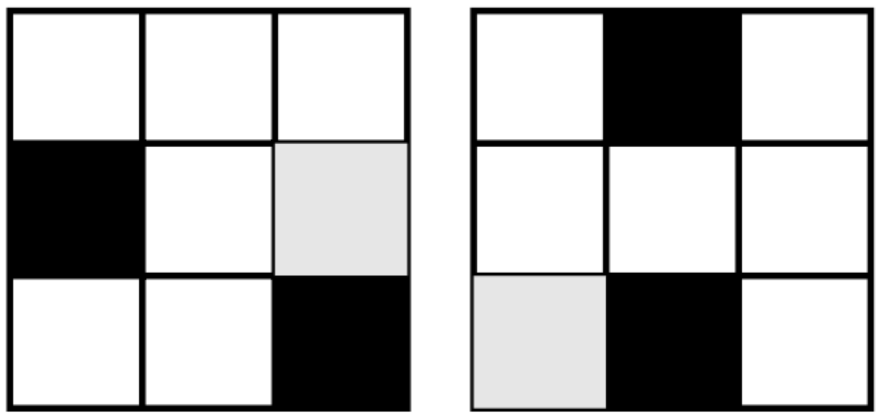
- **Validation Metrics over the test set:**
  - **FID:**
    - FID is one of the most popular metrics for measuring the feature distance between the real and the generated images. Frechet Distance is used to compute the distance between two "multivariate" normal distribution. For a "univariate" normal distribution Frechet Distance is given as

$$d^2(x_r, x_f) = (\mu_r - \mu_f)^2 + (\sigma_r - \sigma_f)^2.$$

    - The use of activations of the last layer from the **Inception-V3** model trained on the PXD images to summarise each image, gives the score. **The lower the FID the better the image diversity and Fidelity.**

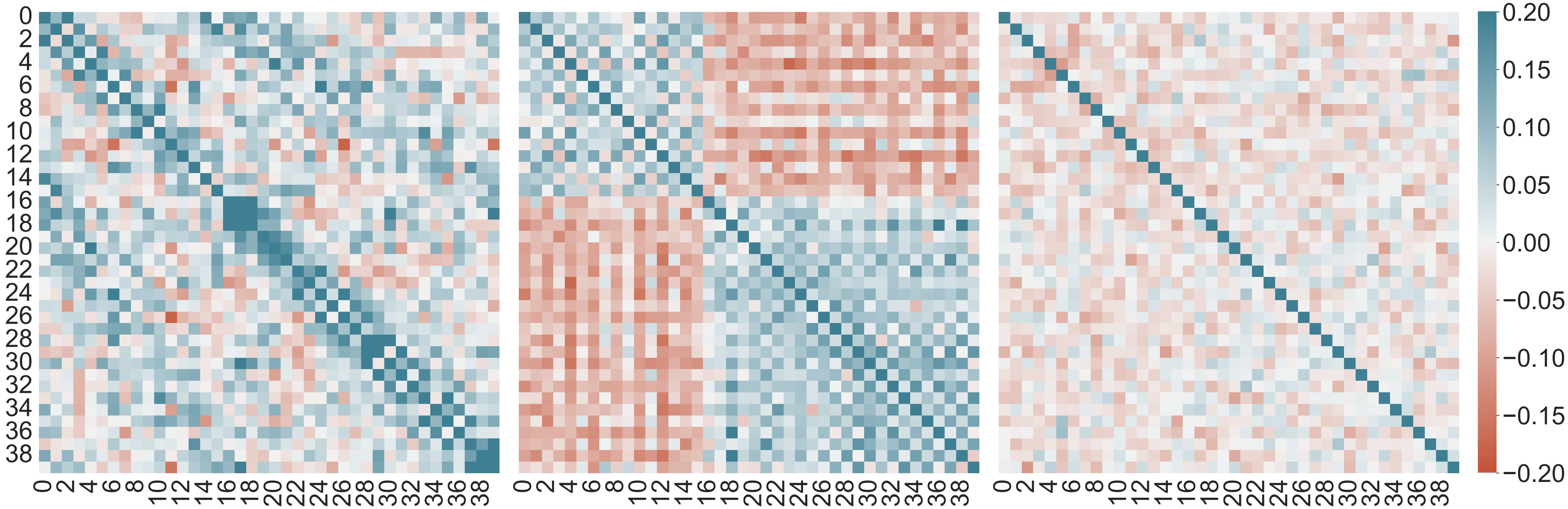
	WGAN-gp	BigGAN-deep	ContraGAN	PE-GAN	IEA-GAN
FID	12.09	4.40 ± 0.88	3.14 ± 0.74	2.61 ± 0.91	$\mathbf{1.50 \pm 0.16}$

    - Possible interpretation of FID in pixel level:

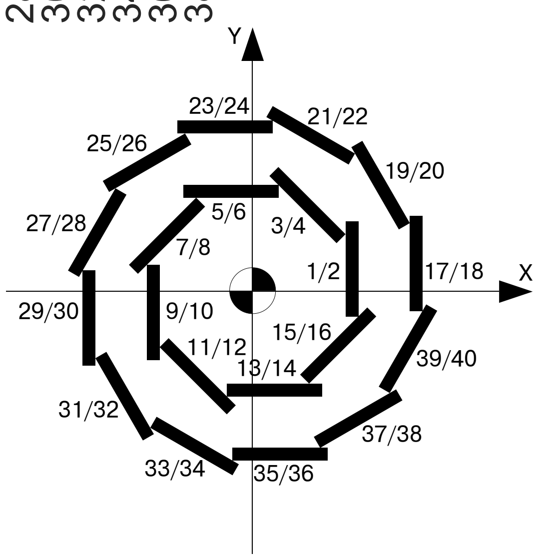


Validation of generated PXD images

- **Validation Metrics over the test set:**
  - **Spearman's correlation between the occupancy of Geant4 simulated images (left), and generated images from IEA-GAN (center), generated images from PE-GAN (right).**

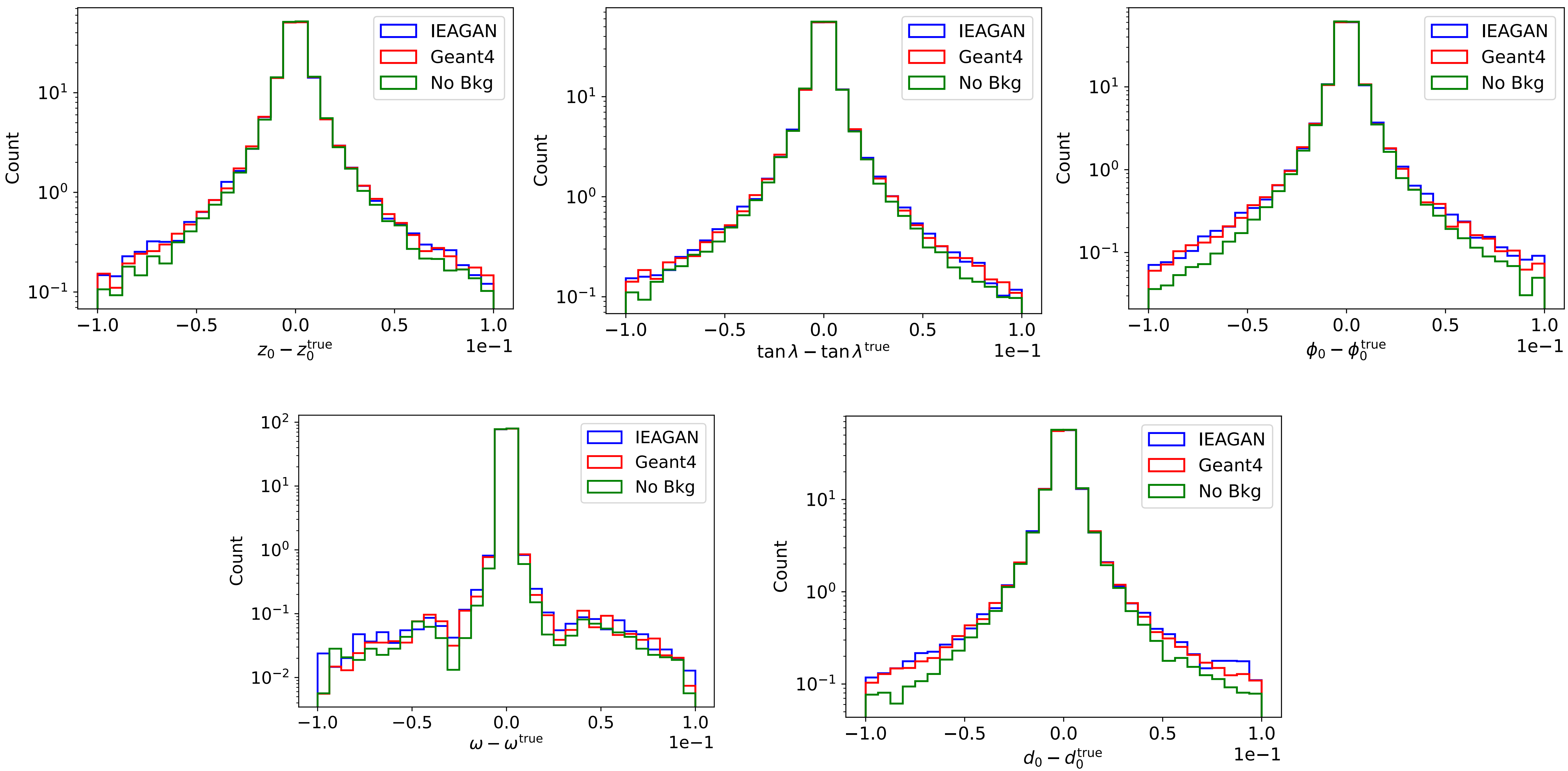


Geant4 IEA-GAN PE-GAN



Validation of generated PXD images

- **Validation Metrics over the test set:**
  - **Physics Analysis: Helix parameter resolutions**



Summary and Outlook

☑ IEA-GAN:

- Successful generation of fine-grained, High Resolution, Correlated PXD images based on the sensor number.
- Accepted at ML4PS workshop at NeurIPS 2022
- Open-source code: https://github.com/Hosein47/IEA-GAN, Pre-print coming soon.

☑ TODO/In Progress:

- Applying IEA-GAN to the calorimeter shower generation in Fast Calorimeter Simulation Challenge.
- Working on the real PXD detector data by transferring the same structure with minor modifications to generate them.
- Do Luminosity dependent Event generation (OOD simulation)
- Doing a comprehensive validation of generated hitmaps by estimating the systematic uncertainty on the tracking efficiency, fake rate and resolution.

Thank You

References

- Kang, Minguk, and Jaesik Park. "Contragan: Contrastive learning for conditional image generation." Advances in Neural Information Processing Systems 33 (2020): 21357-21369
- Hashemi, Hosein, et al. "Pixel Detector Background Generation using Generative Adversarial Networks at Belle II." EPJ Web of Conferences. Vol. 251. EDP Sciences, 2021.
- Srebre, Matej, et al. "Generation of Belle II Pixel Detector Background Data with a GAN." EPJ Web of Conferences. Vol. 245. EDP Sciences, 2020.
- Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." Advances in neural information processing systems 30 (2017).
- Brock, Andrew, Jeff Donahue, and Karen Simonyan. "Large scale GAN training for high fidelity natural image synthesis." arXiv preprint arXiv:1809.11096 (2018).

Back up Slides

☑ The Base Model:

- **Technologies:**
  - Residual blocks
  - Spectral Normalisation
  - Orthogonal Weight init.
  - Orthogonal regularisation
  - Contrastive Learning
  - Hinge Loss
  - Consistency Regularisation
  - Differentiable Augmentation
  - IEA Loss
  - 5x10^-5 lr for both G and D

Algorithm 1 Intra-Event Aware GAN

Require: generator and discriminator parameters θ_G, θ_D, Intra-Event-aware coefficient λ_{IEA}, Uniformity coefficient λ_{uniform} and hyperparameter s, Adam hyperparameters α, β_1, β_2, event size M, number of discriminator iteration steps per generator iteration N_D

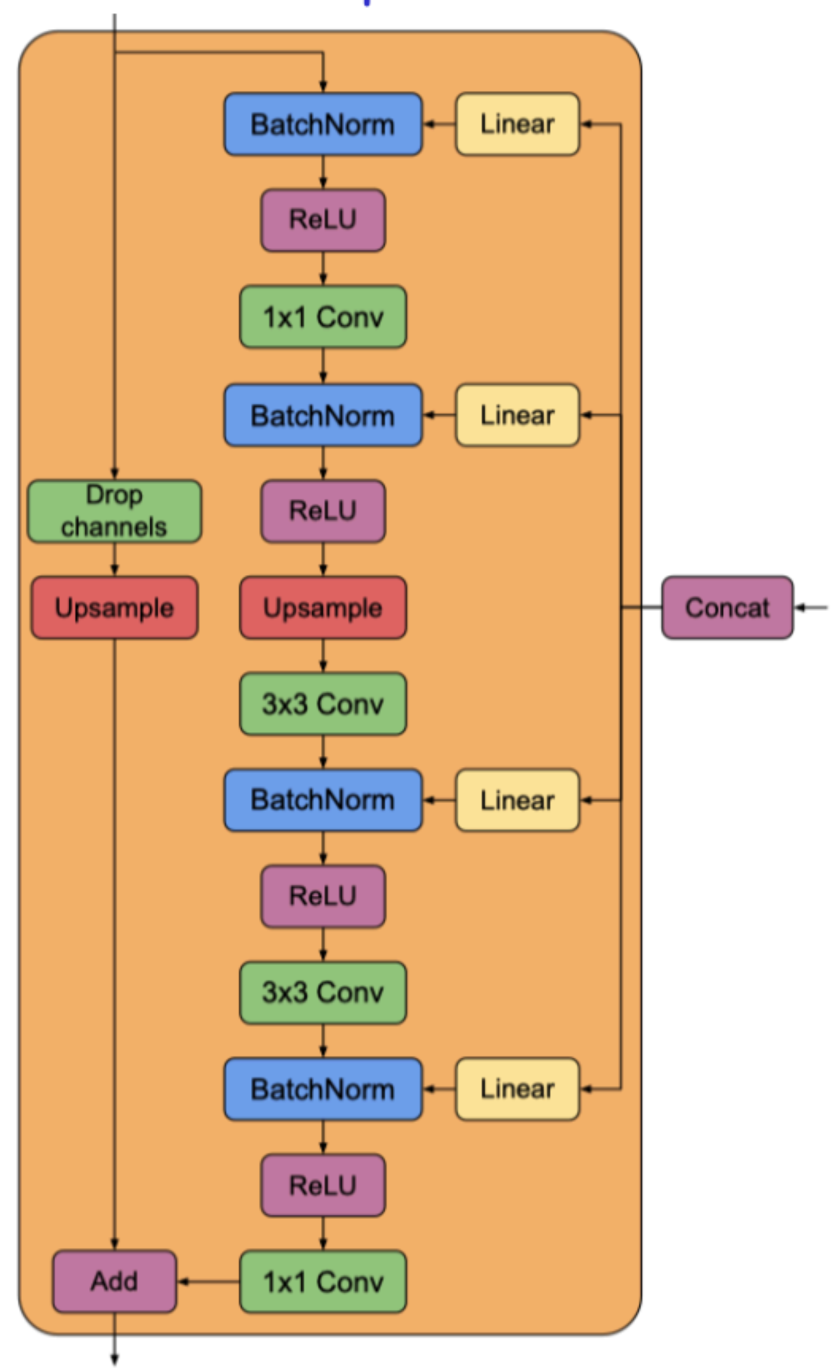
for number of training iterations **do**
  for $t = 1, \ldots, N_D$ **do**
    sample $\{z^i\}_{i=1}^{M} \sim p(z)$,
    $\{x^i, y^i\}_{i=1}^{M} \sim p_{\text{event}}(x, y)$, $\{r^i\}_{i=1}^{M} \sim p_{\text{Rdof}}(z)$ ▷ Event Sampling.
    for $i = 1, \ldots, M$ **do**
      $\ell_{D_{\text{hinge}}}^{(i)} \leftarrow \ell_{D_{\text{hinge}}}(x^{(i)};\ G(z^i, y^i, r^i))$
    end for
    $\mathcal{L}_{D_{\text{hinge}}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \ell_{D_{\text{hinge}}}^{(i)}$
    $\mathcal{L}_{\text{uniform}} \leftarrow \mathcal{L}_{\text{uniform}}(x;\ s)$ ▷ The Uniformity Loss.
    $\mathcal{L}_{2C}^{\text{real}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \ell_{2C}(x^i, y^i)$
    $\theta_D \leftarrow Adam(\mathcal{L}_{D_{\text{hinge}}} + \lambda_{2C}\mathcal{L}_{2C}^{\text{real}} + \lambda_{\text{uniform}}\mathcal{L}_{\text{uniform}}, \alpha, \beta_1, \beta_2)$
  end for
  sample $\{z^i\}_{i=1}^{M} \sim p(z)$,
  sample $\{r^i\}_{i=1}^{M} \sim p_{\text{Rdof}}(z)$ ▷ Event Sampling.
  for $i = 1, \ldots, M$ **do**
    $\ell_{G_{\text{hinge}}}^{(i)} \leftarrow \ell_{G_{\text{hinge}}}(G(z^i, y^i, r^i))$
  end for
  $\mathcal{L}_{G_{\text{hinge}}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \ell_{G_{\text{hinge}}}^{(i)}$
  $\mathcal{L}_{\text{IEA}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \ell_{\text{IEA}}(G(z^i, y^i, r^i), x^i)$ ▷ The Intra-Event Aware Loss.
  $\mathcal{L}_{2C}^{\text{fake}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \ell_{2C}(G(z^i, y^i, r^i), y^i)$
  $\theta_G \leftarrow \text{Adam}(\mathcal{L}_{G_{\text{hinge}}} + \lambda_{2C}\mathcal{L}_{2C}^{\text{fake}} + \lambda_{\text{IEA}}\mathcal{L}_{\text{IEA}}, \alpha, \beta_1, \beta_2)$
end for

Back up Slides

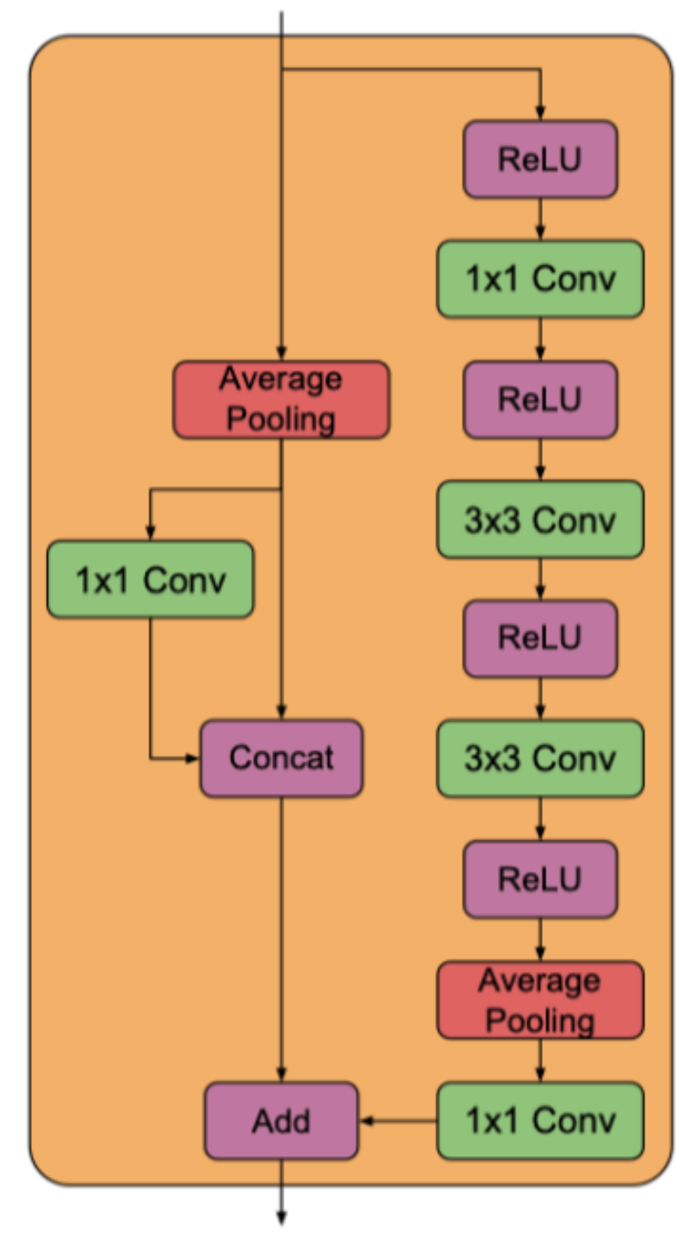
The Base Model:

- **Technologies:**
  - Residual blocks
  - Spectral Normalisation
  - Orthogonal Weight init.
  - Orthogonal regularisation
  - Contrastive Learning
  - Hinge Loss
  - Consistency Regularisation
  - Differentiable Augmentation
  - IEA Loss
  - 5x10^-5 lr for both G and D

ResBlock up in G



ResBlock down in D



Back up Slides

Table A1: FID comparison between IEA-GAN, IEA-GAN with RRM only, IEA-GAN with Uniformity loss only , and IEA-GAN with both IEA-loss, averaged across six random seeds.

	IEA-GAN	Only RRM	RRM with Uniformity	RRM with IEA-loss
FID	$\mathbf{1.50 \pm 0.16}$	2.74 ± 0.62	2.71 ± 0.14	3.42 ± 0.52

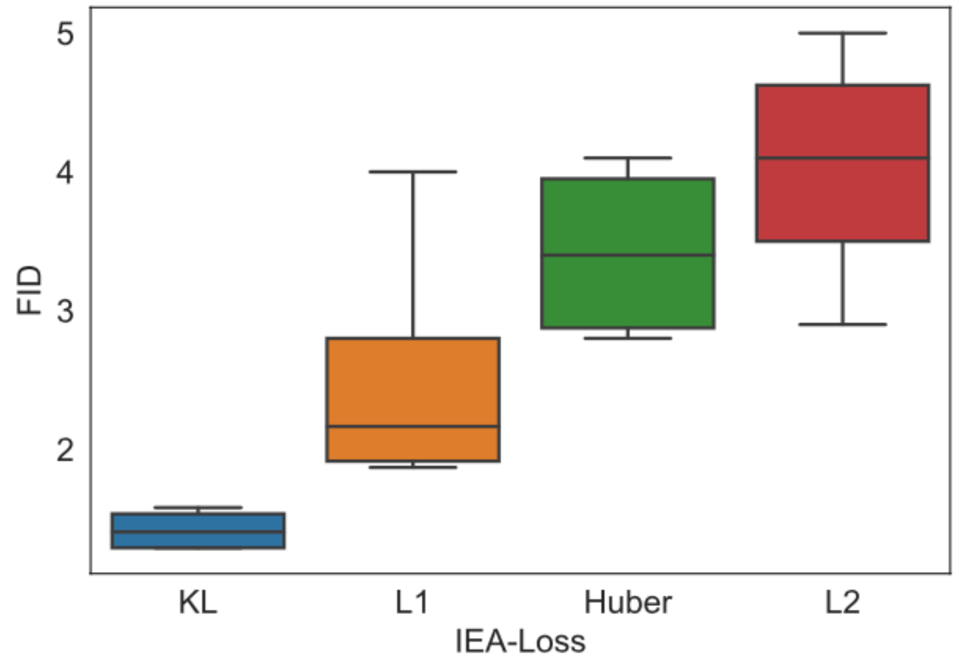


Fig. A1: Comparison of the FID between different IEA-losses

Back up Slides

Table B3: Computational performance of IEA-GAN and PE-GAN generators on a single core of an Intel Xeon Silver 4108 1.80GHz (CPU) and NVIDIA V100 with 32 GB of memory (GPU) compared to GEANT4. For the generative models, the mean and standard deviation obtained for sets of 1000 events. The time for GEANT4 refers to the theoretical time it would take to run the simulation of all background processes on-the-fly.

Hardware	Simulator	time/event[s]
CPU	Geant4	≈ 1500
	PE-GAN	11.781 ± 0.357
	IEA-GAN	10.159 ± 0.208
GPU	PE-GAN	0.090 ± 0.010
	IEA-GAN	0.070 ± 0.006

Overlay Problem

- **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.
  - Requires distributing over all sites where MC is produced.
- **Solution:** Generating the background data on the way of analysis with GANs instead of storing them.