A new method for the data driven estimation of background using GAN :

Case study on γ + Jets background in $H \to \gamma\gamma$ analysis

IN2P3/IRFU ML workshop - September 28th 2022

Victor Lohezic (victor.lohezic@cea.fr)
Fabrice Couderc, Julie Malclès, Özgür Şahin
IRFU - CEA Saclay

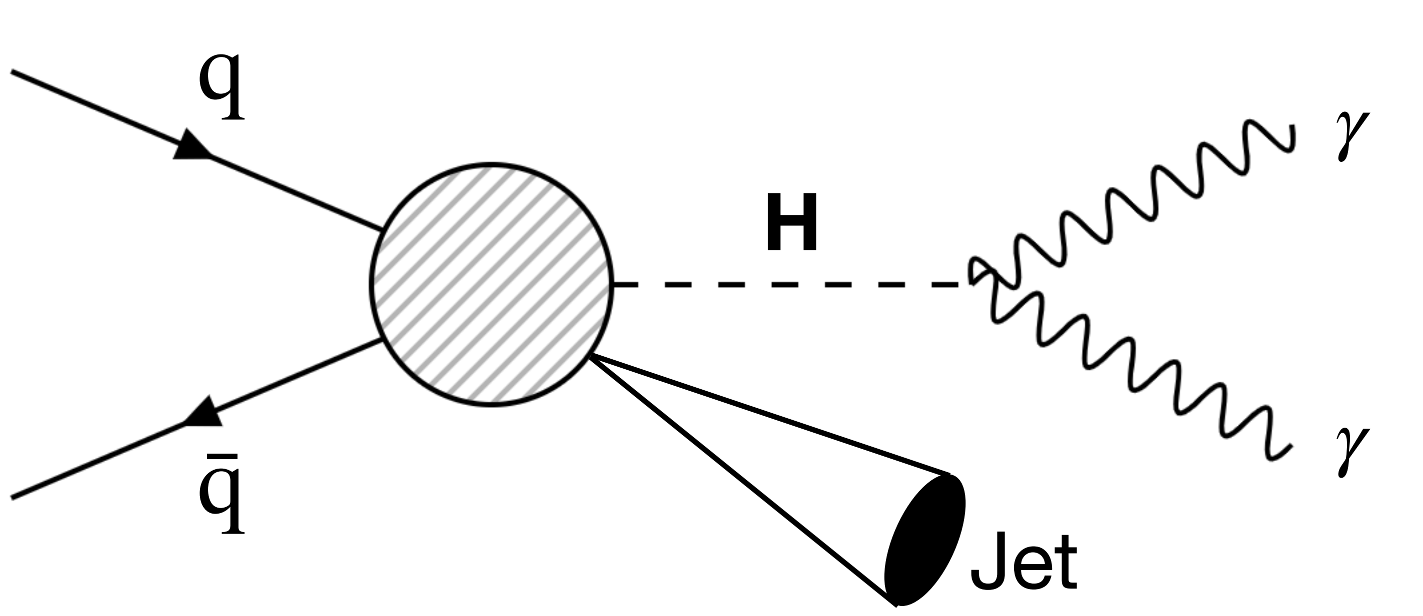
Introduction

GAN based data-driven technique to estimate background processes with a misidentified object in collider events. We will showcase this technique for the γ + Jets background process of the H → $\gamma\gamma$ analysis.

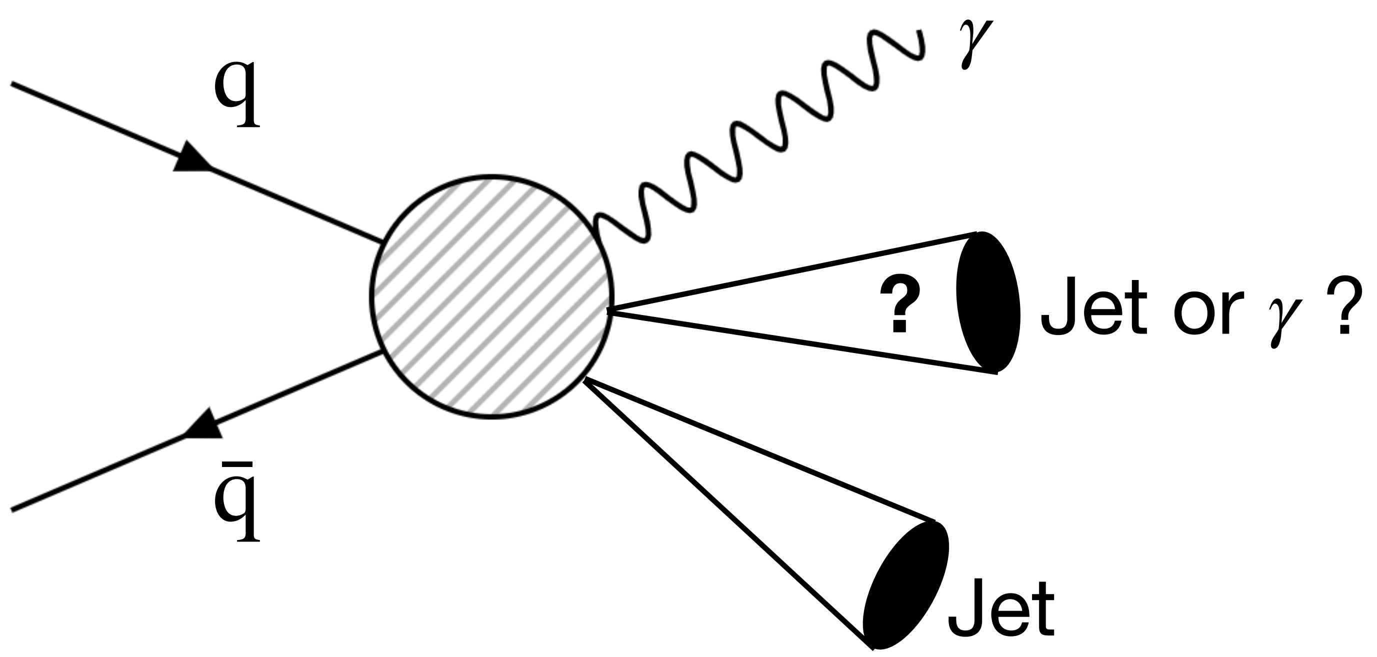
In the H → $\gamma\gamma$ analysis, dominant backgrounds are : $\gamma\gamma$ + Jets, γ + Jets, Multi Jets (MJ)

- The agreement between Data and Monte Carlo (MC) simulated samples for γ + Jets and MJ is not satisfying and the statistics is too low for the training of subsequent discriminants.

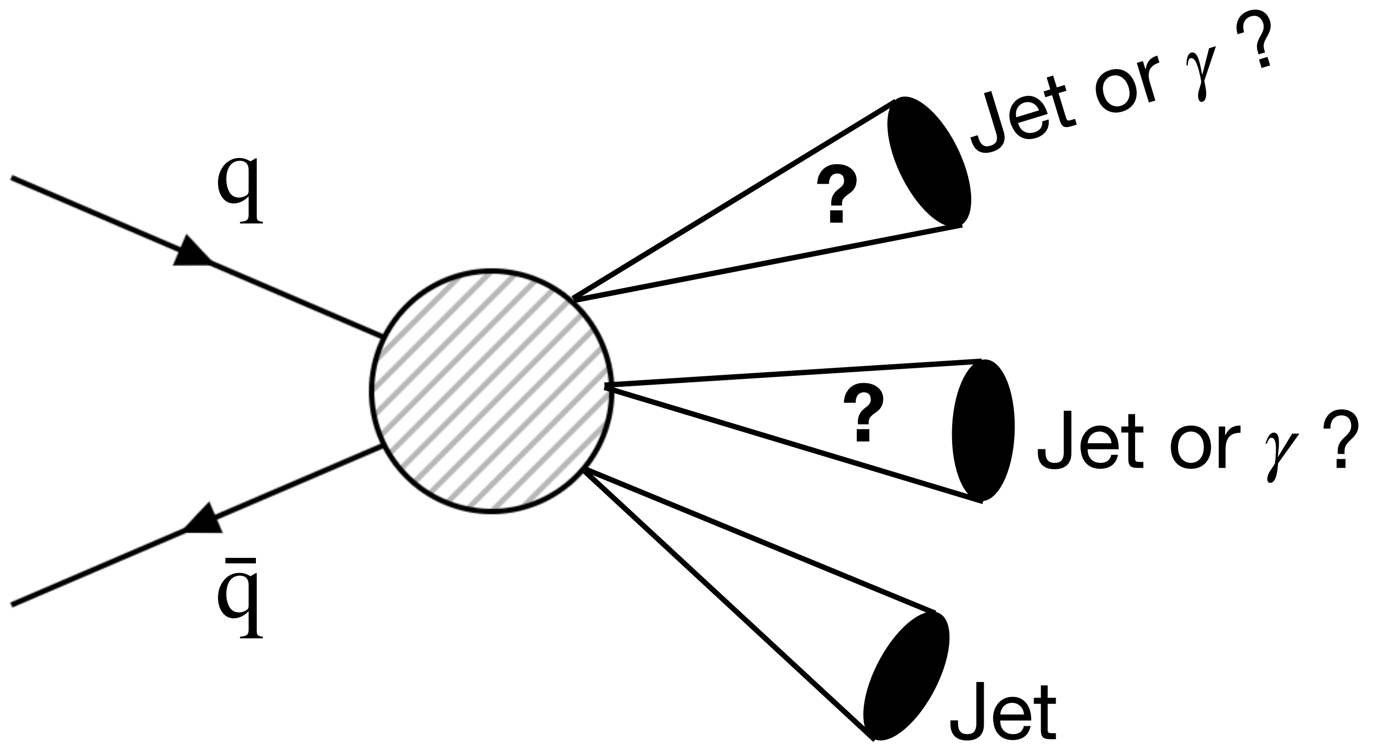
Example of H → $\gamma\gamma$ event (Signal)



Example of γ + Jets event (Bkg)



Example of MJ event (Bkg)



➡ What if we use data directly to describe those samples ?

- We would like to improve the data driven approach used in the previously published analyses using this technique.

Overview

I. A data driven estimation of the background

II. Training a GAN

a. Generative Adversarial Network (GAN)
b. Evaluation procedure

III. Generating a full object (misidentified photon)

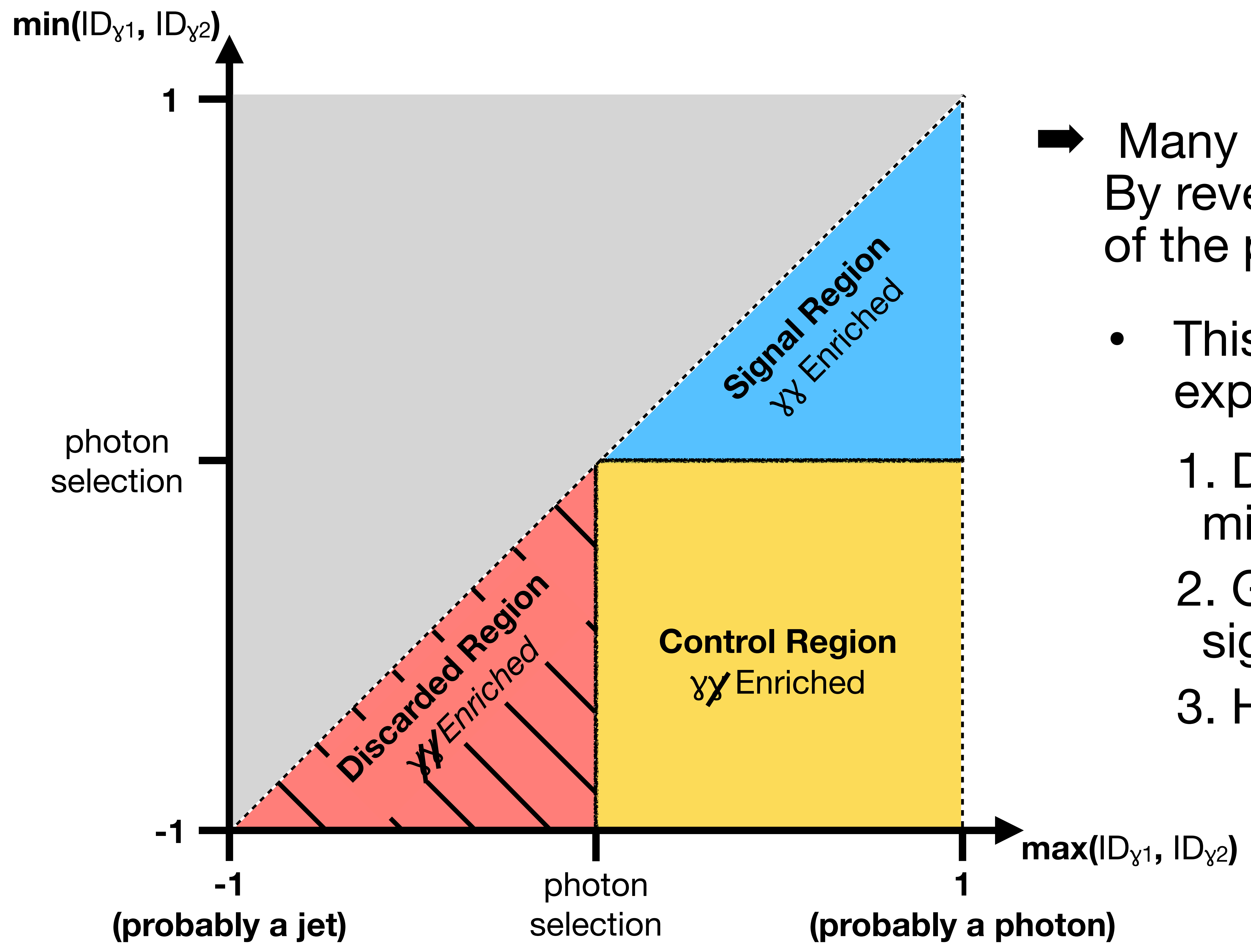
a. Optimization of training
b. Applying GAN to MC control region

IV. Conclusions and outlooks

I. A data driven estimation of the background

In an event each photon is given a score (photon ID) representing its likelihood to be a photon. Control region in data based on photon ID is used to replace MC γ + Jets / MJ samples (better agreement, more statistics).

➡ Need one photon with very low photon ID : probably a misidentified photon $\not\gamma$ (as opposed to a prompt photon γ)



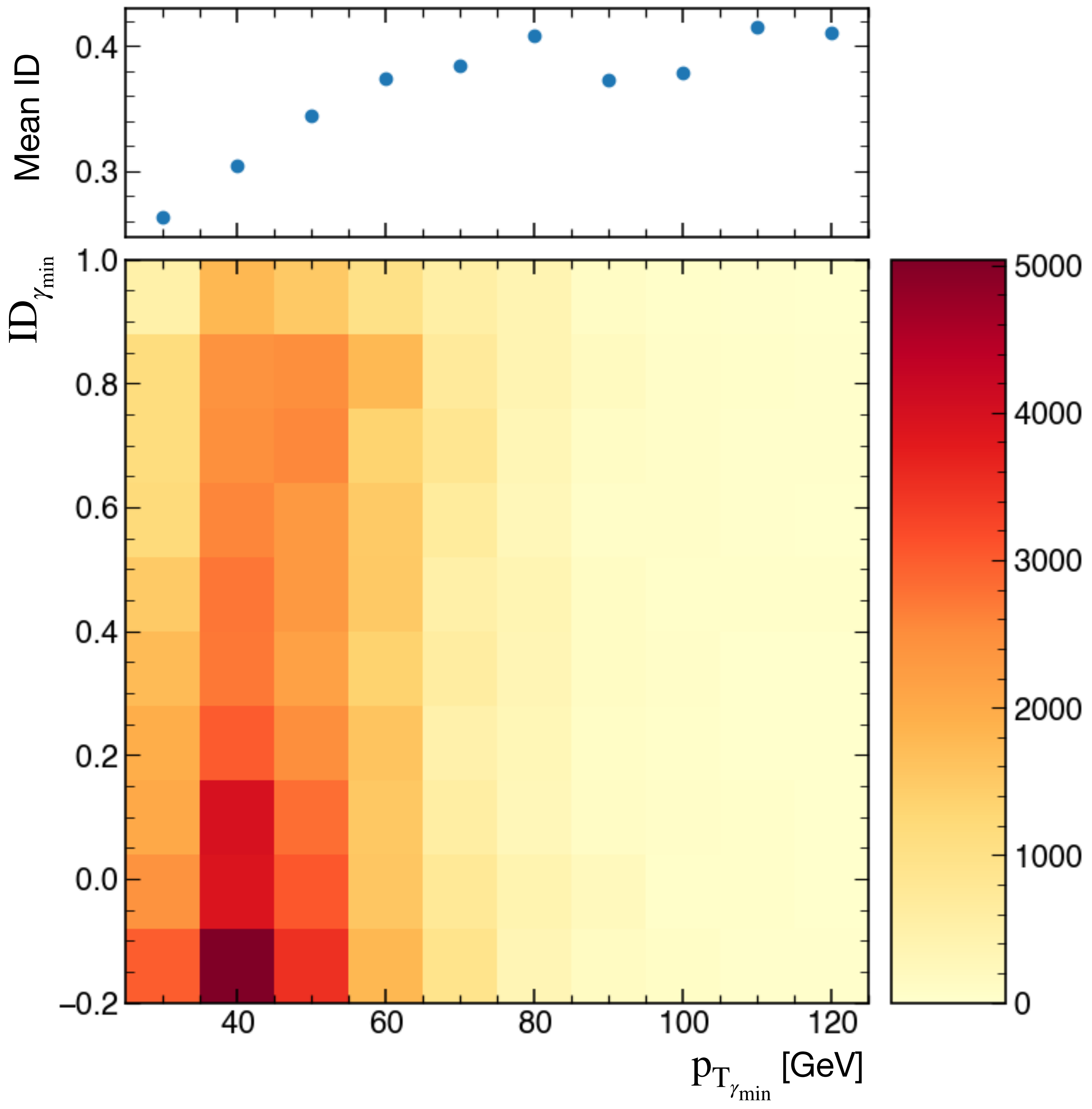
➡ Many analysis already use data driven background estimation. By reverting the cut on the min photon ID, one needs either to get rid of the photon ID variable or to generate a new min photon ID !

- This procedure was used in published analysis from CMS experiment [1], new ID was generated by :

  1. Deriving a 1D probability density function (PDF) from the misidentified photon ID distribution
  2. Generating a random min photonID following this PDF, in the signal region but below the max photonID
  3. However correlations are not preserved

[1] Measurements of ttH production and the CP structure of the Yukawa interaction between the Higgs boson and the top quark in the diphoton decay channel, CMS collaboration

- All this procedure is relying on key assumptions :
  - ☑ Features independent from the photons of the events are behaving identically in signal and control regions
  - ☑ Events in the control region are γ + Jets or MJ events (true for 96% of the events in MC)
  - ☑ Photon with min photon ID is misidentified (always true for MJ, true 96.2% of the time for γ + Jets)
  - ☐ photon ID is not correlated with other features of the photon (p_T, η, ϕ)



- Additional drawback : need to reweight the events. Differences in kinematic features between control and signal region. New weights computed using MC but always some subjectiveness in the choice of features.

➡ We propose a new method to generate a suitable photon (not only ID) taking into account these correlations thanks to ML and more specifically GAN (Generative Adversarial Networks)

Overview

I. A data driven estimation of the background

II. Training a GAN

a. Generative Adversarial Network (GAN)
b. Evaluation procedure

III. Generating a full object (misidentified photon)

a. Optimization of training
b. Applying GAN to MC control region

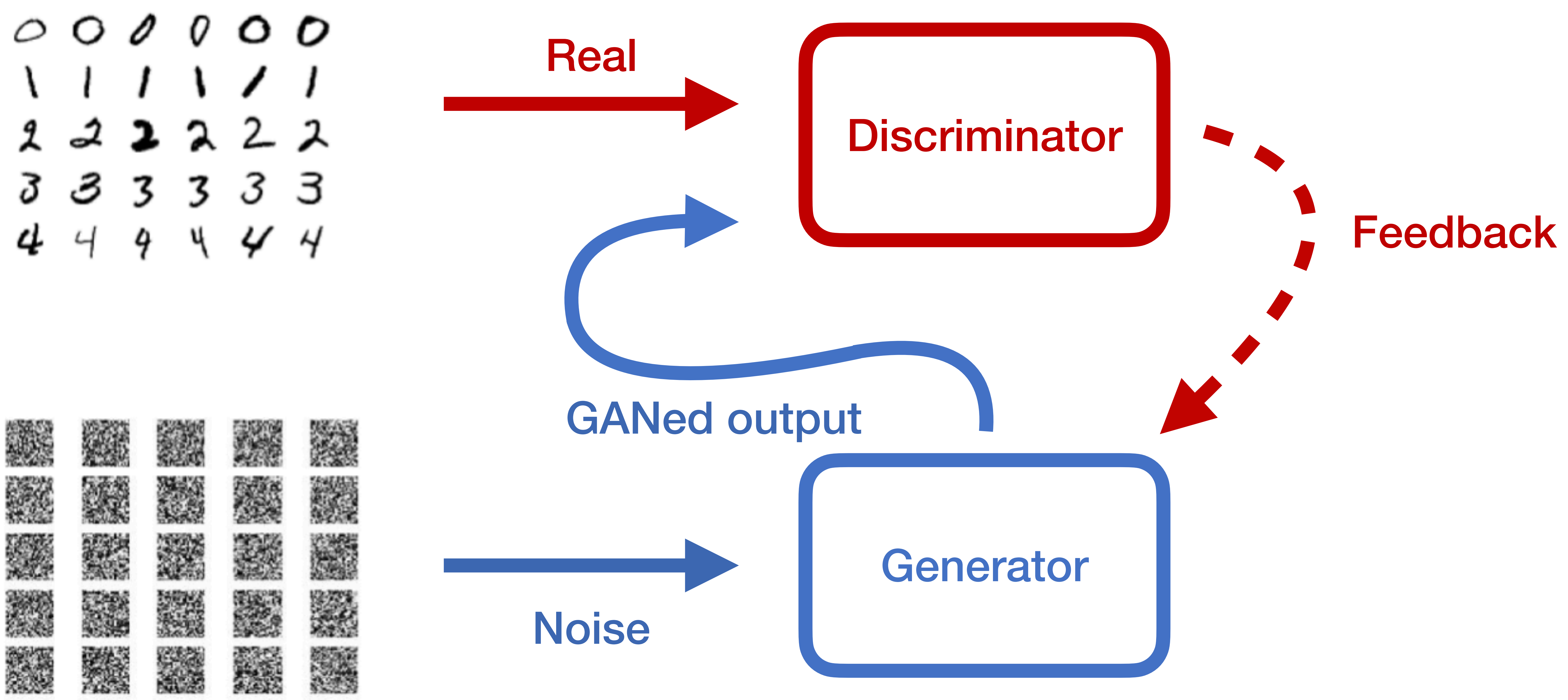
IV. Conclusions and outlooks

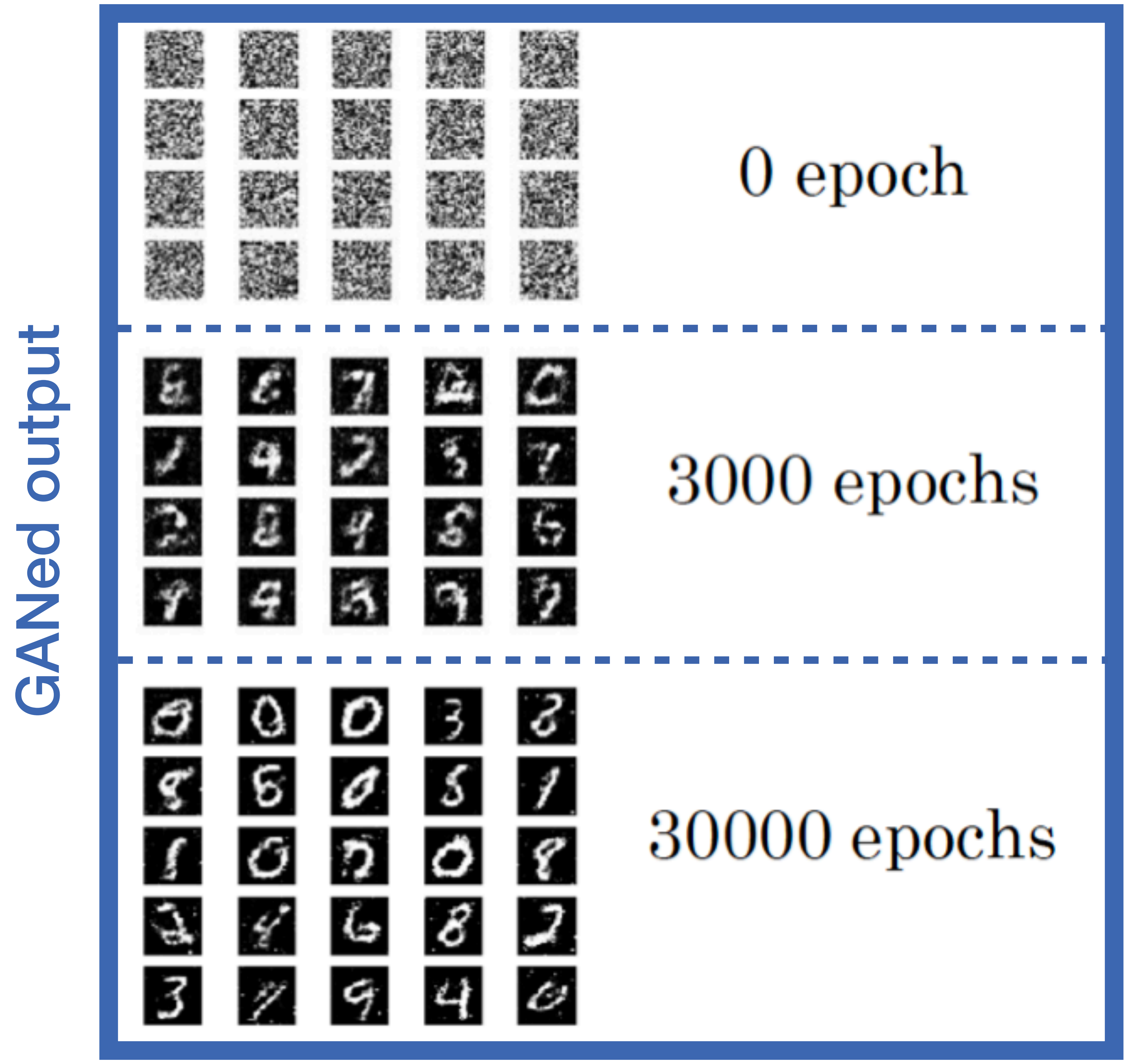
II. Training a GAN

II.a - Generative Adversarial Networks (GANs)

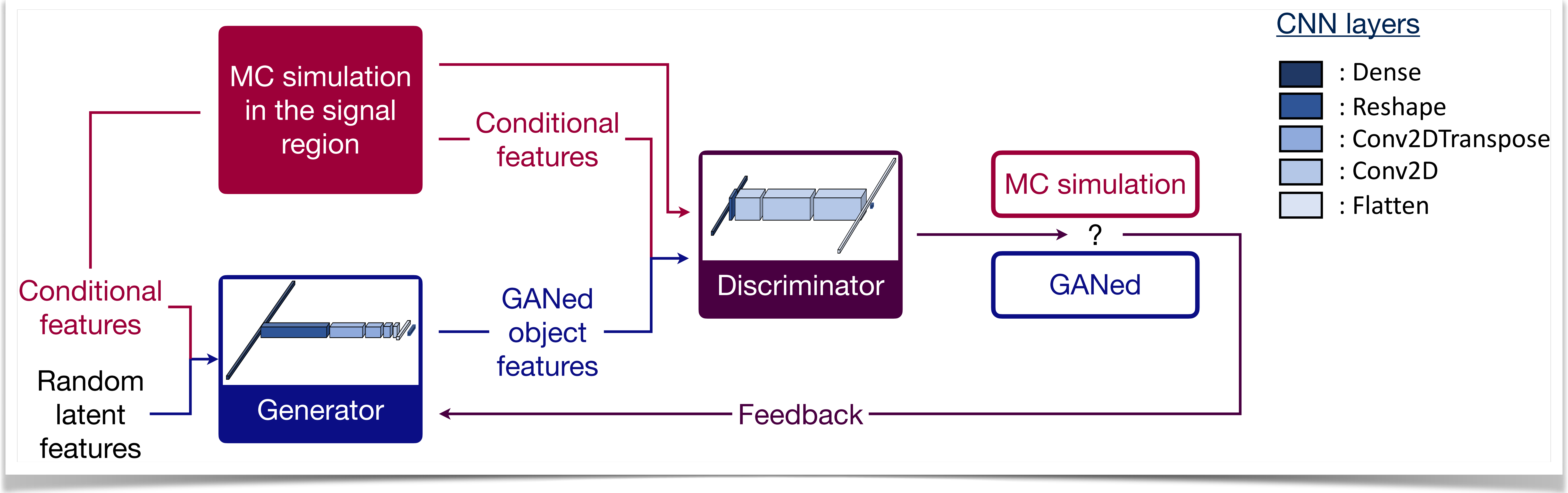
Would it be possible to create an algorithm capable of learning underlying correlations and capable of generating a sample statistically independent from the training sample ?

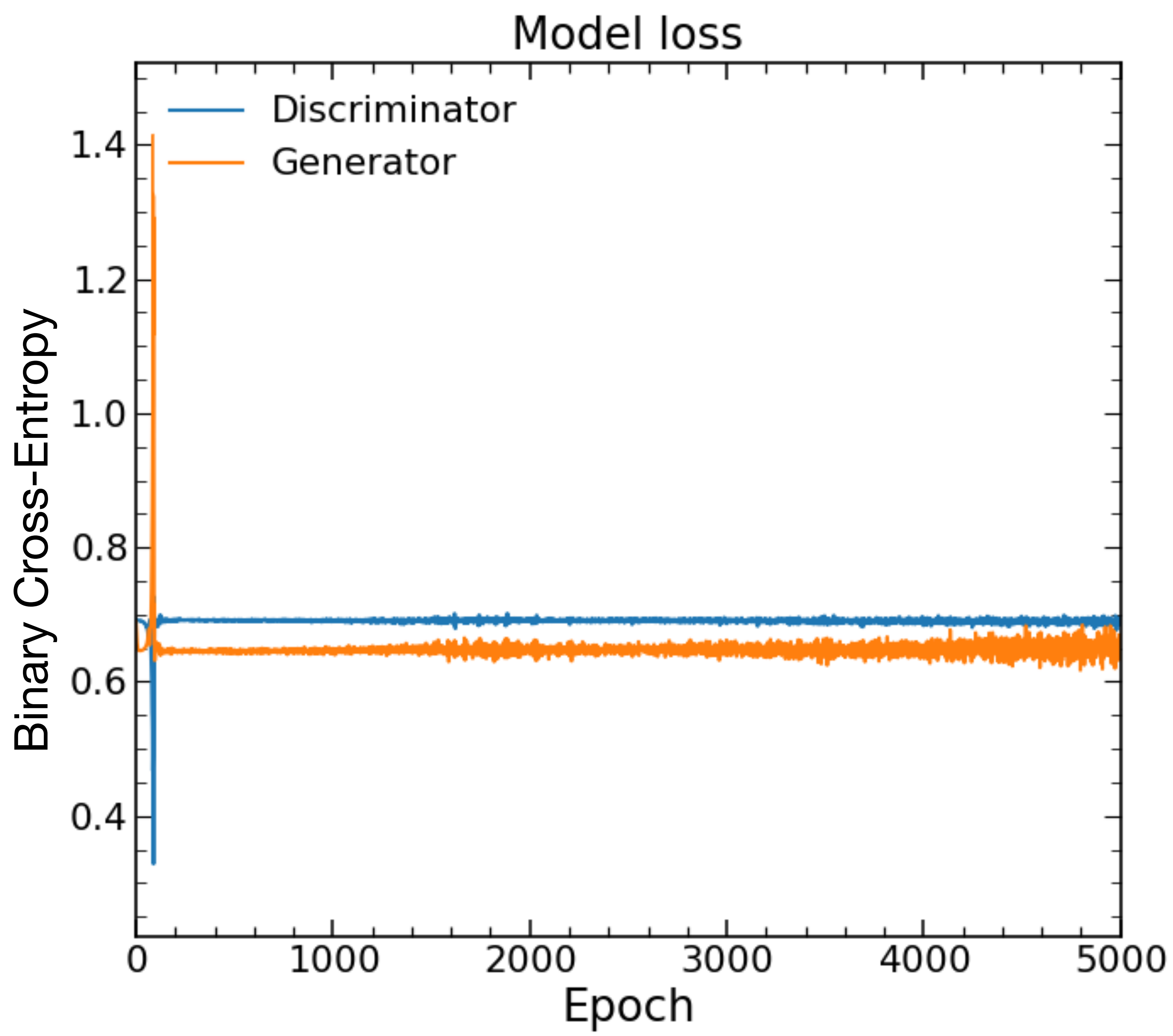
- *Goodfellow et al.* suggested a model consisting of two neural networks competing against each other :
  - the "discriminator" sorts samples between real and generated ones - *i.e.* discriminates fakes
  - the "generator" tries to produce samples which will fool the discriminator





In our case we use a conditional version of a GAN and we train on the misidentified photon (ID, p_T, η, ϕ) :





- Usually, monitoring the loss of a neural network is enough to evaluate its performance. It is not the case for a GAN where both networks need to perform well against the other so their loss stays flat.

➡ We need to set up a more elaborate evaluation procedure

II.b - Evaluation procedure

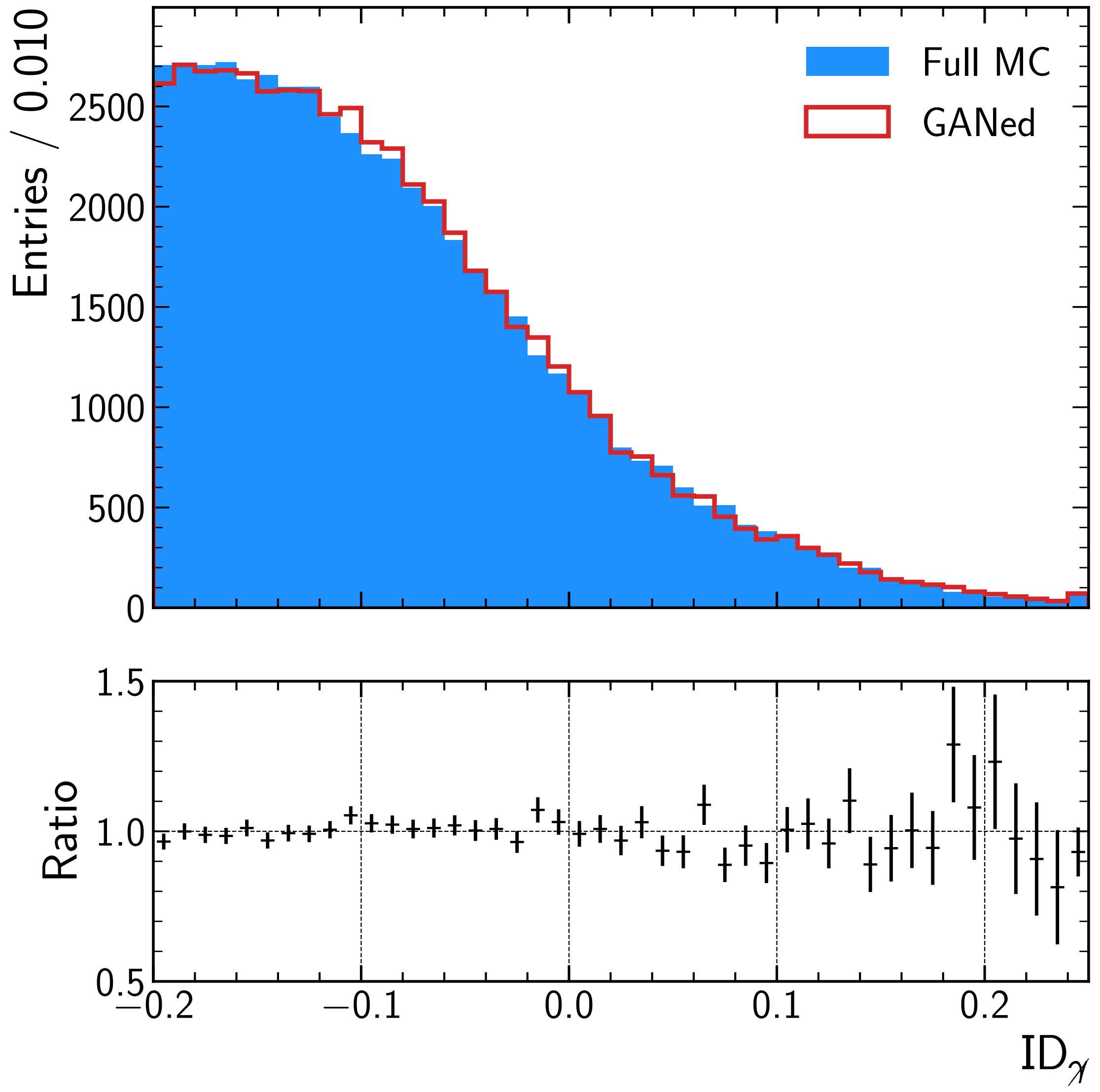
- To evaluate the performance of a given model, we rely on different metrics computed for each training epoch on the training sample and on a validation sample :

χ^2 metric :

$$\chi^2 = \sum_{k=1}^{\#\text{bins}} \frac{(n_k - N_k)^2}{N_k^2}$$

n_k : sum of the weights of **GANed events** in bin k

N_k : sum of the weights of **original events** in bin k



Log Likelihood metric :

$$-2\ln(\Lambda) = -2\sum_{k=1}^{\#\text{bins}} N_k \cdot \log(p_k), \quad p_k = \frac{n_k}{\sum n}$$

n_k : sum of the weights of **GANed events** in bin k

N_k : sum of the weights of **original events** in bin k

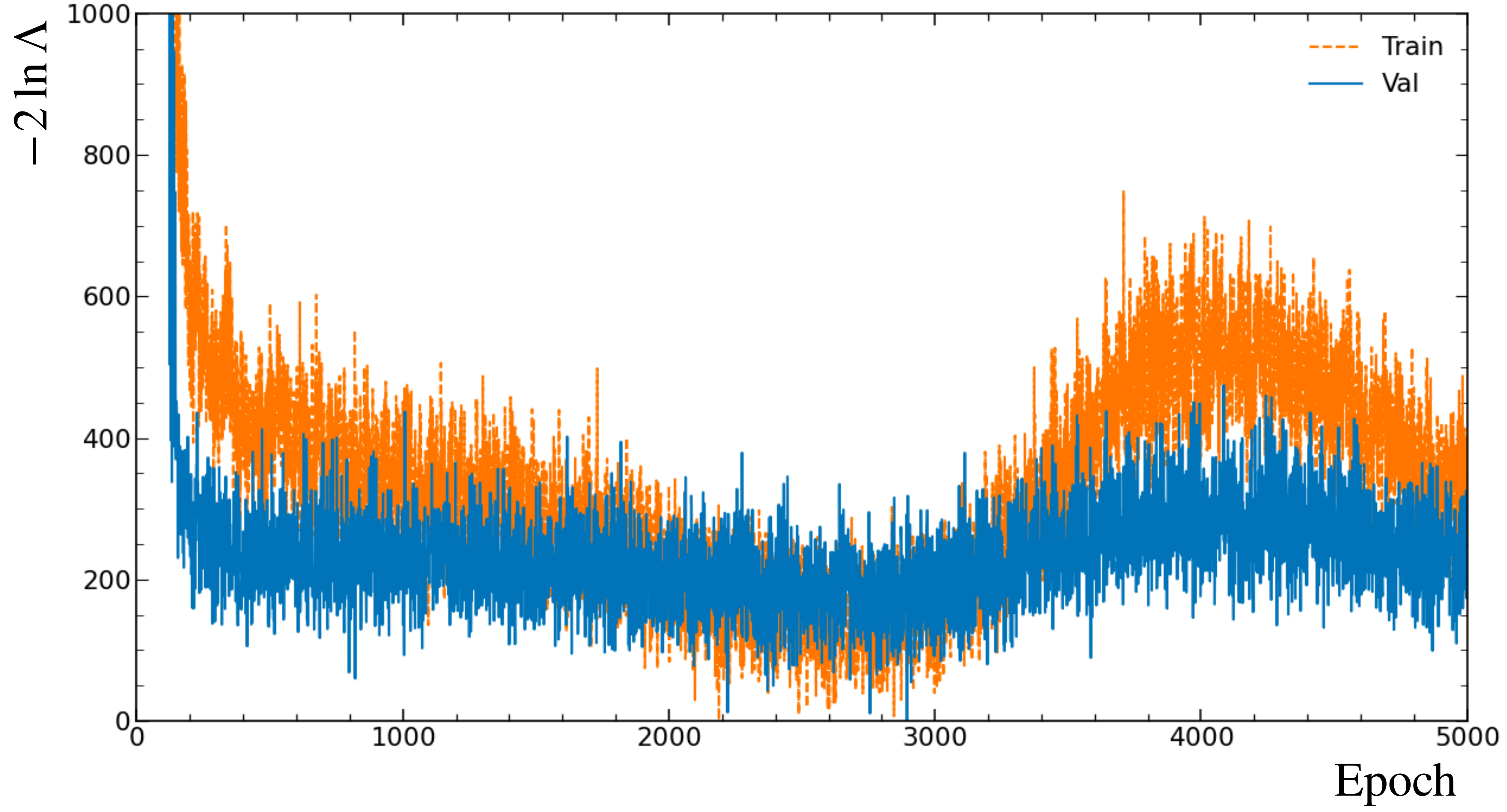
For the NLL we histogram our events in 4D :

- transverse momentum of misidentified photon p_{T_γ}
- pseudorapidity of misidentified photon η_γ
- p_T of diphoton pair over its mass $\frac{p_{T_{\gamma\gamma}}}{m_{\gamma\gamma}}$
- ID of misidentified photon ID_γ

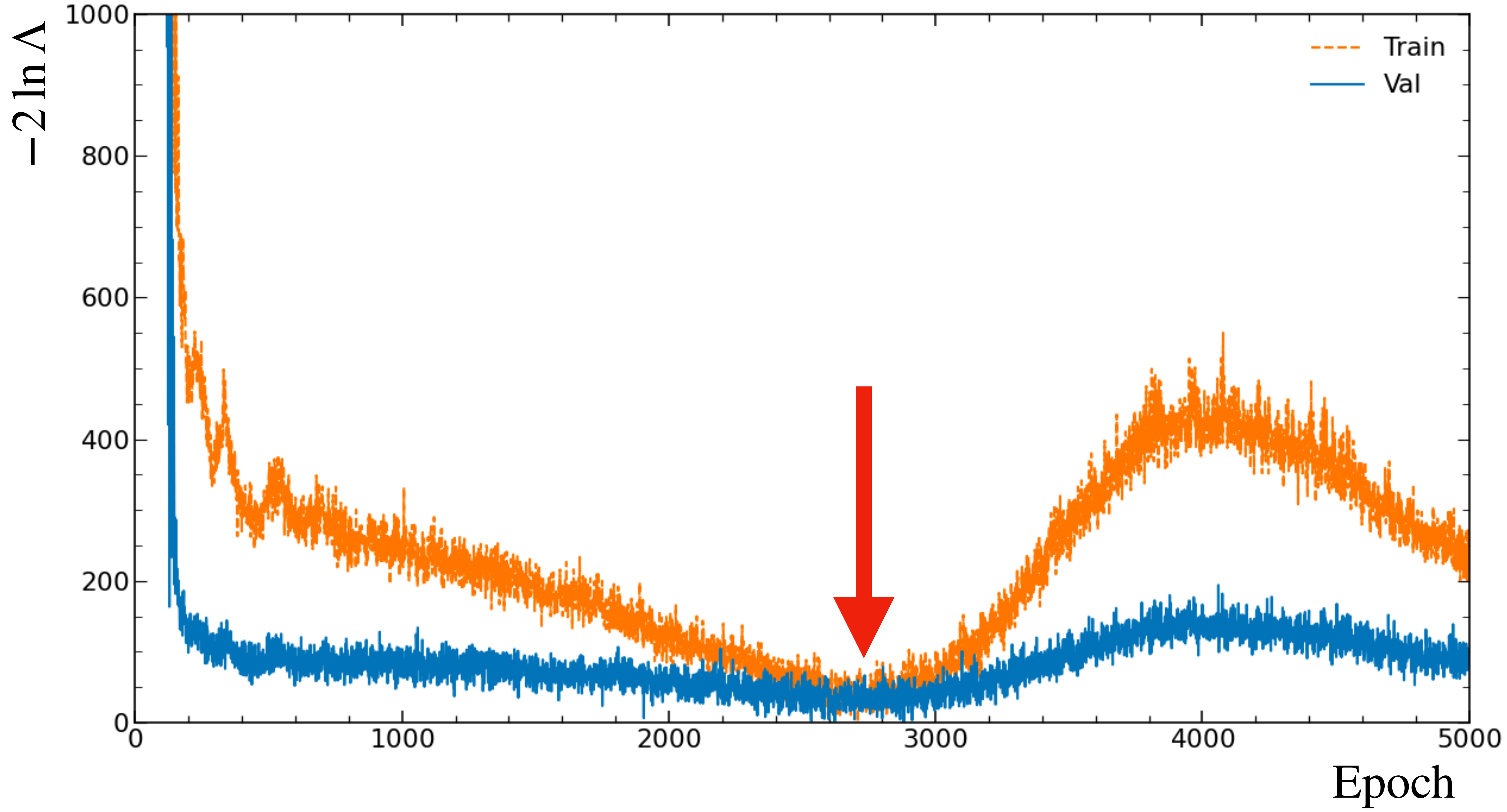
Takes into account correlations by construction

- p_k (see slide 9) estimation is statistically limited creating fluctuations in the NLL. These fluctuations can be reduced by increasing the number of generation per event :

1 generation per event



10 generations per event



- Seeing how the fluctuations decrease, we decide to go to 100 generation per event

- Then we can find epochs where the model is reaching minima for these metrics and take a closer look at its performance

Overview

I. **A data driven estimation of the background**

II. **Training a GAN**
   a. Generative Adversarial Network (GAN)
   b. Evaluation procedure

III. **Generating a full object (misidentified photon)**
   a. Optimization of training
   b. Applying GAN to MC control region

IV. **Conclusions and outlooks**

III. Generating a full object (misidentified photon)

III.a - Optimization of training

Performance of a GAN (and neural network in general) is affected by the way it is trained and by what are called hyperparameters. Here are some of the hyperparameters we optimize :

- ☑ **Batch size** : Each epoch is a training over the full training sample and this sample can be divided in batch. Help to reduce the fluctuations of the training but could stop on local minima.

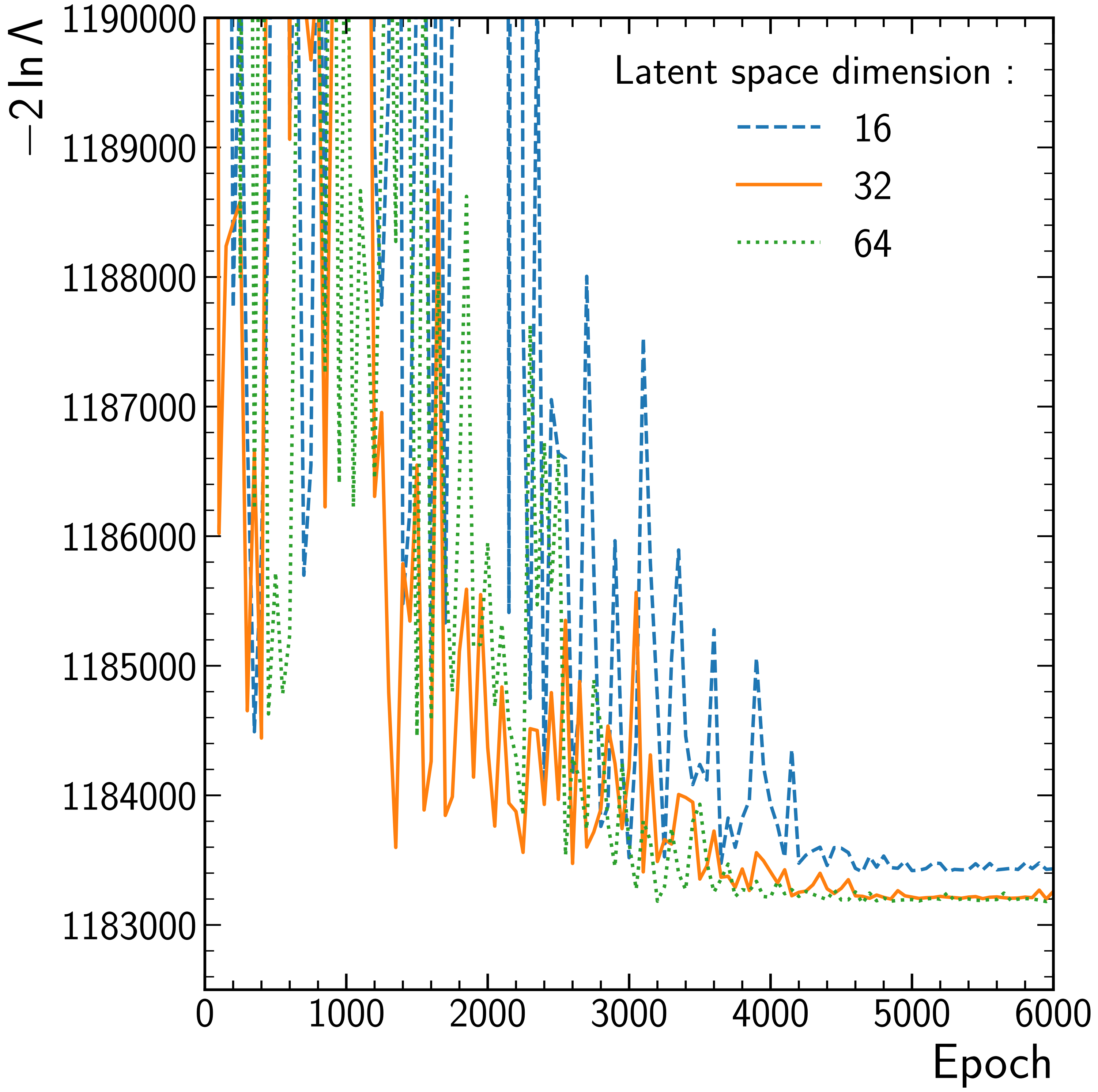
- ☑ **Learning rate** : Coefficient applied when updating the weights of the networks. With a high LR we make bigger steps toward the optimal weight distribution but with a risk to go over it.

- ☑ **Gradient descent optimiser** : Algorithm to update the weights toward their optimal distribution. Some allows to converge quickly but can switch between multiple distributions and others can focus on converging toward one optimal distribution only.

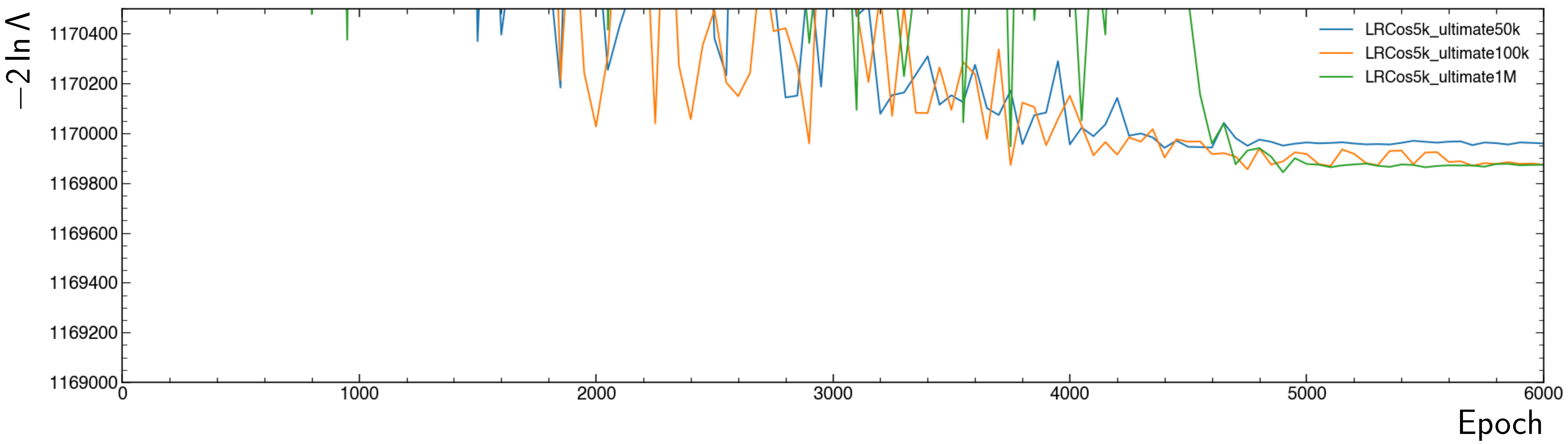
- ☑ **Noise on labels during training** : Instead of identifying a MC photon with label 1 and a GANed photon with label 0, we add X% noise on this value. Help the GAN converge and stabilise.

- ☑ **Latent space dimension** : The generator needs random vector of given size as input to generate values.

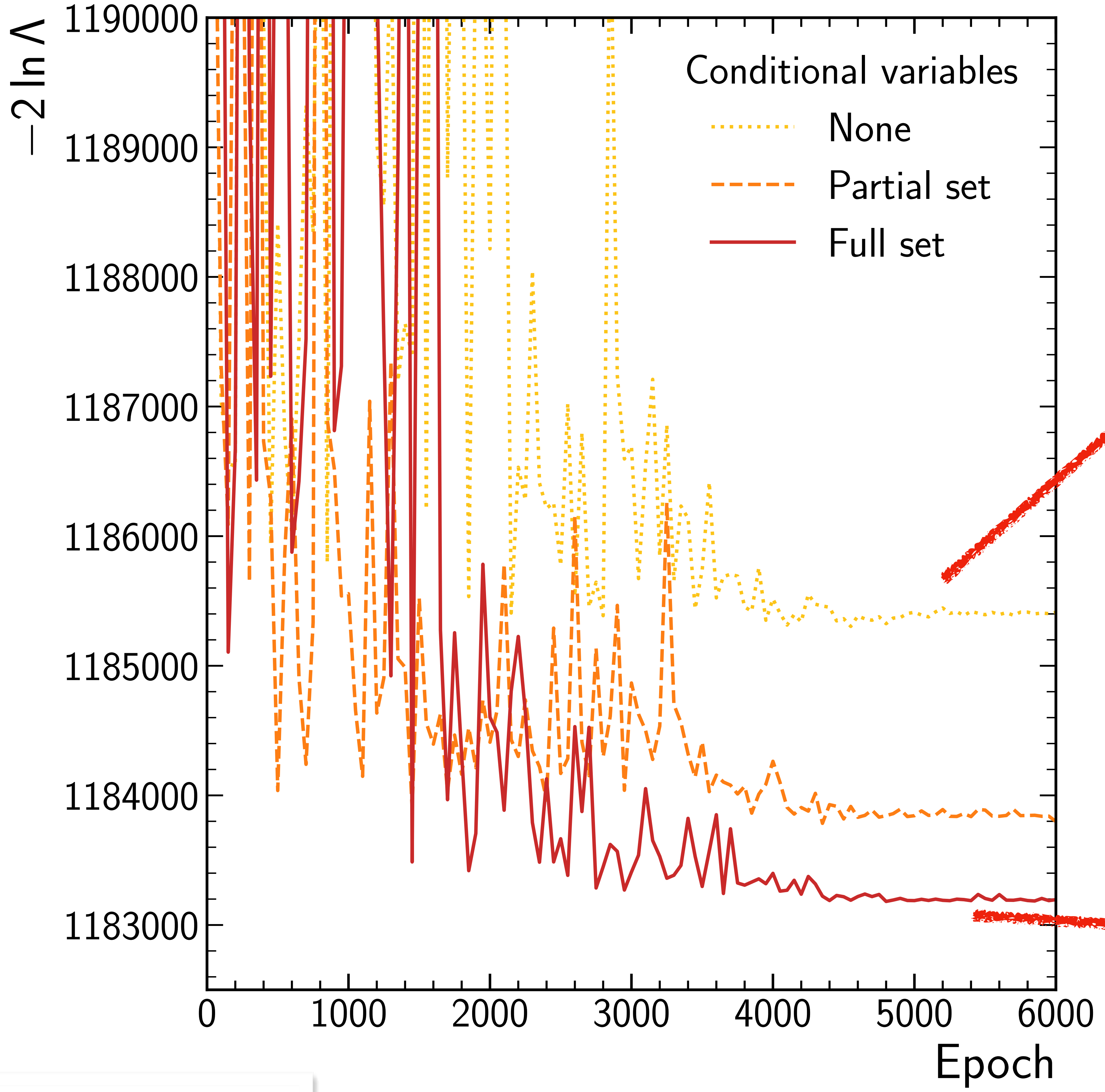
- Example of hyperparameter selection

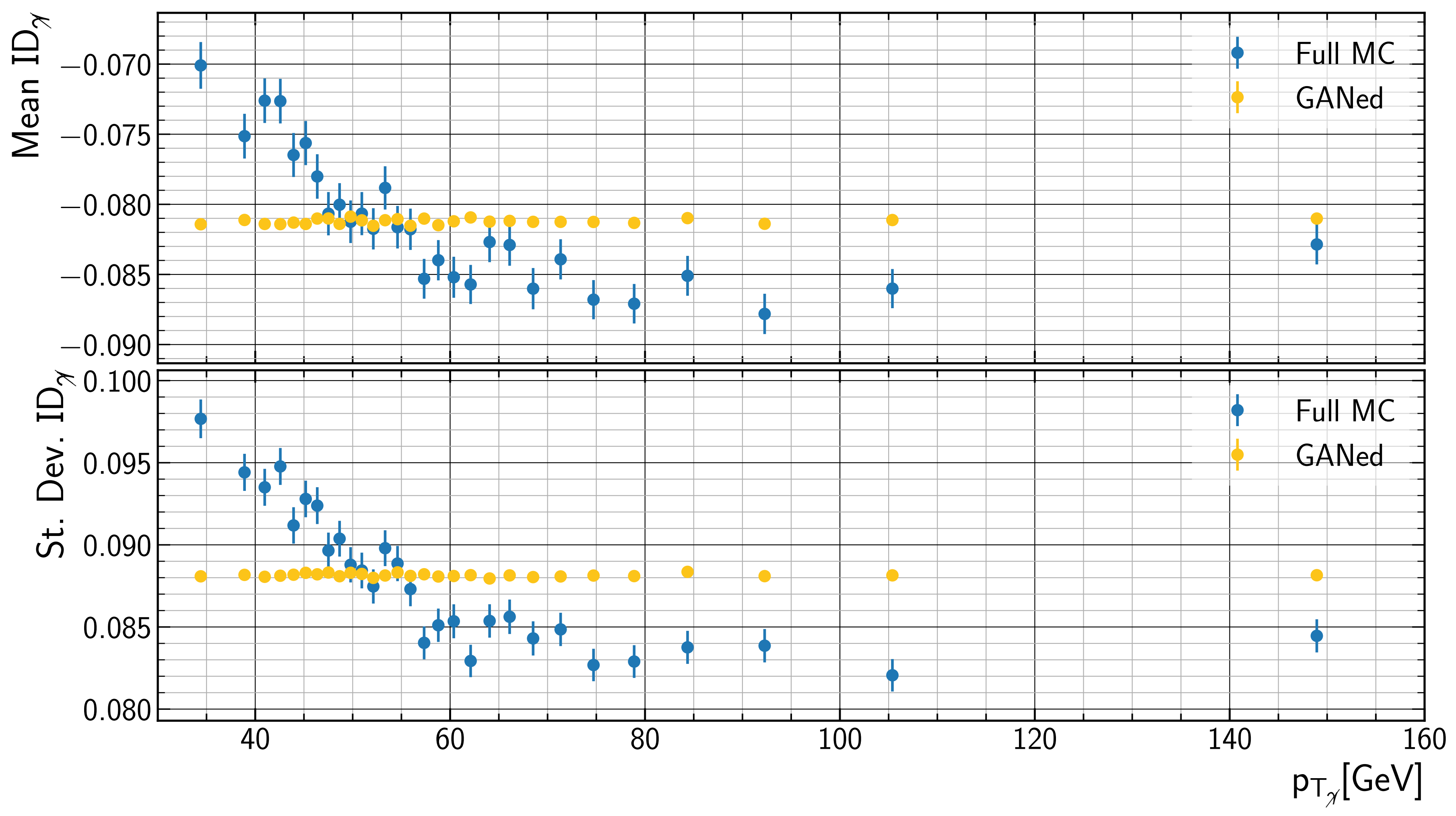


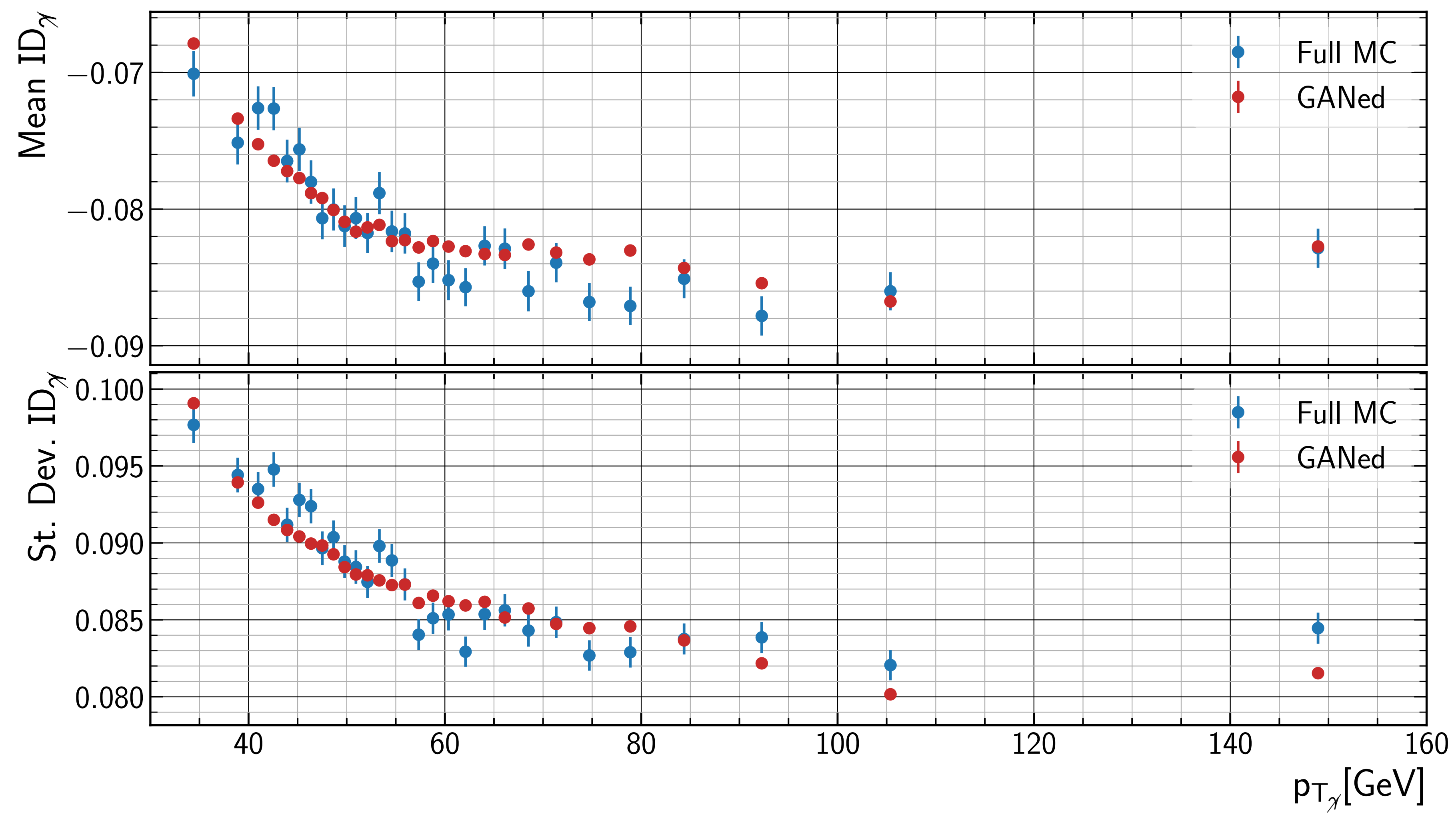
- Test to which extend the training benefits from a larger training sample

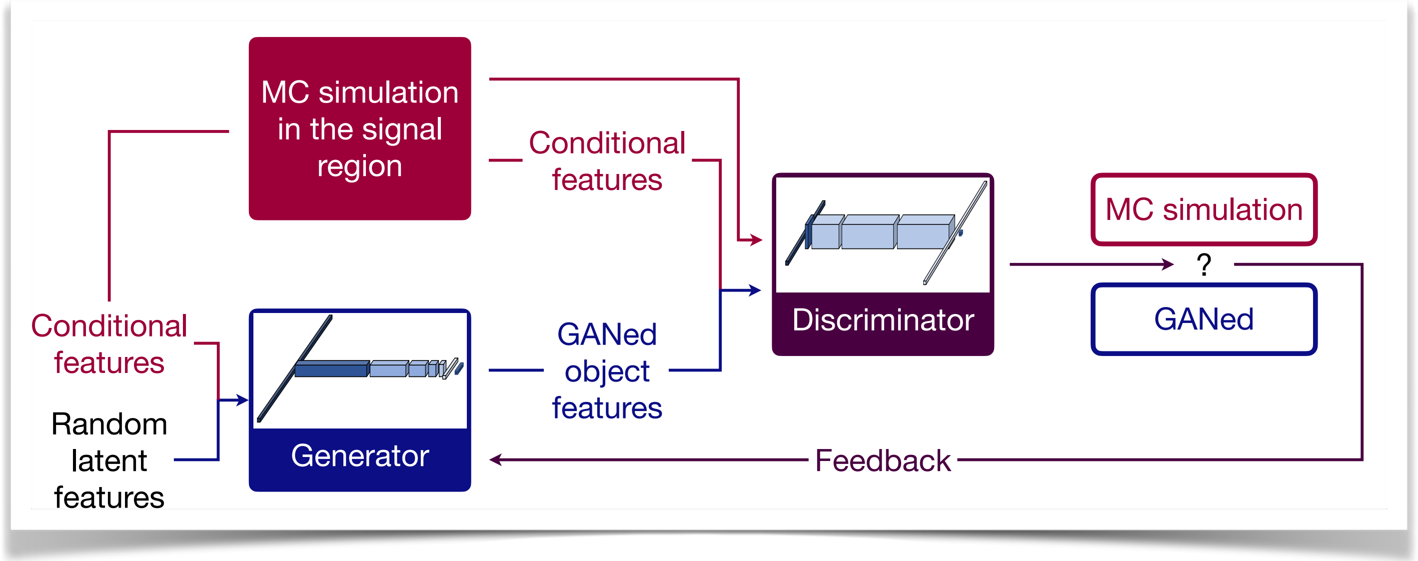


- Trying to find the correct set of observables to train our GAN, we can clearly see how hiding information from the network affects its ability to learn correlations





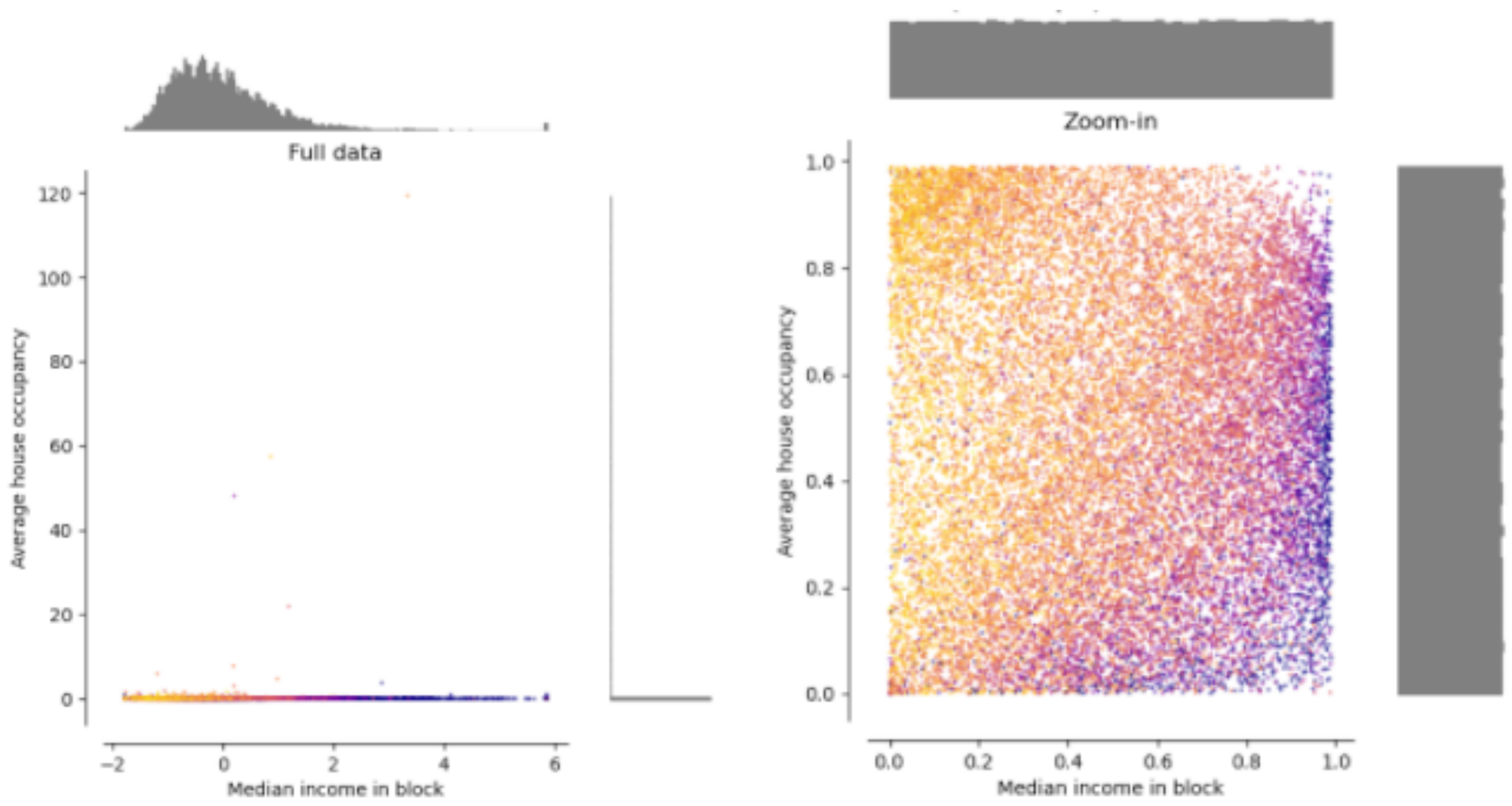


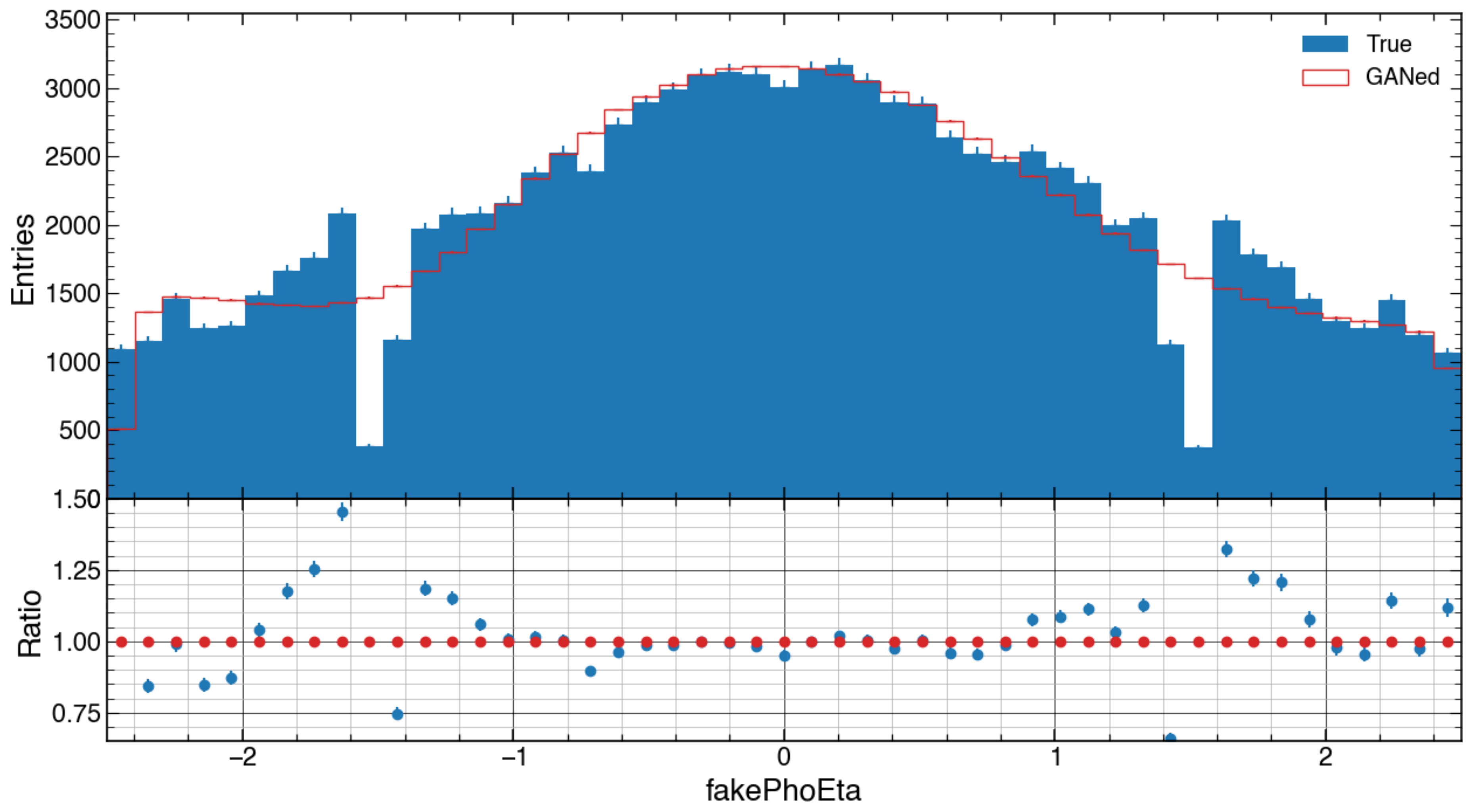


- Discrepancies for low p_T, low photonID and also in the η gaps, a better preprocessing could help

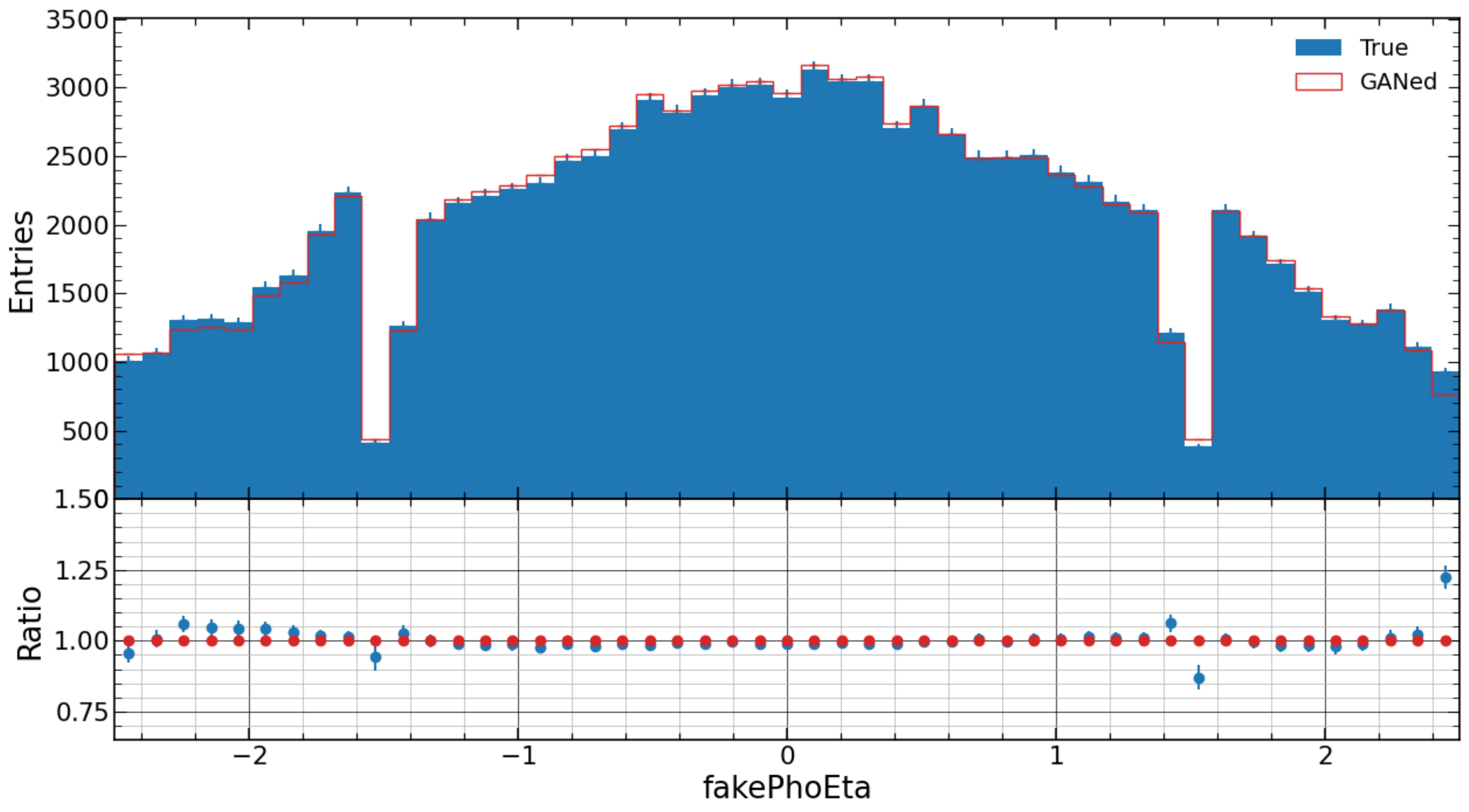
- Preprocessing the input data using a quantile transformation

Example from scikit-learn's documentation



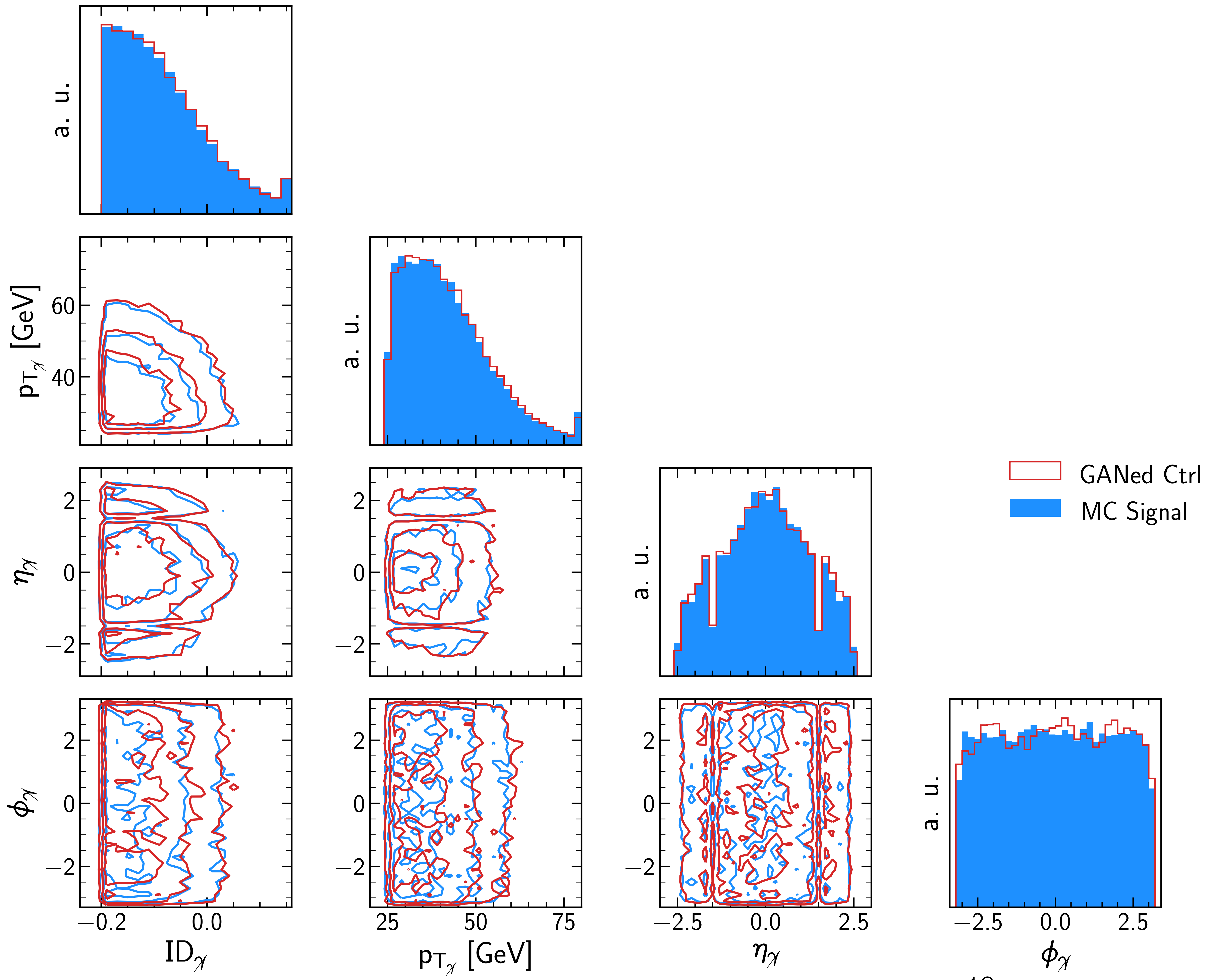


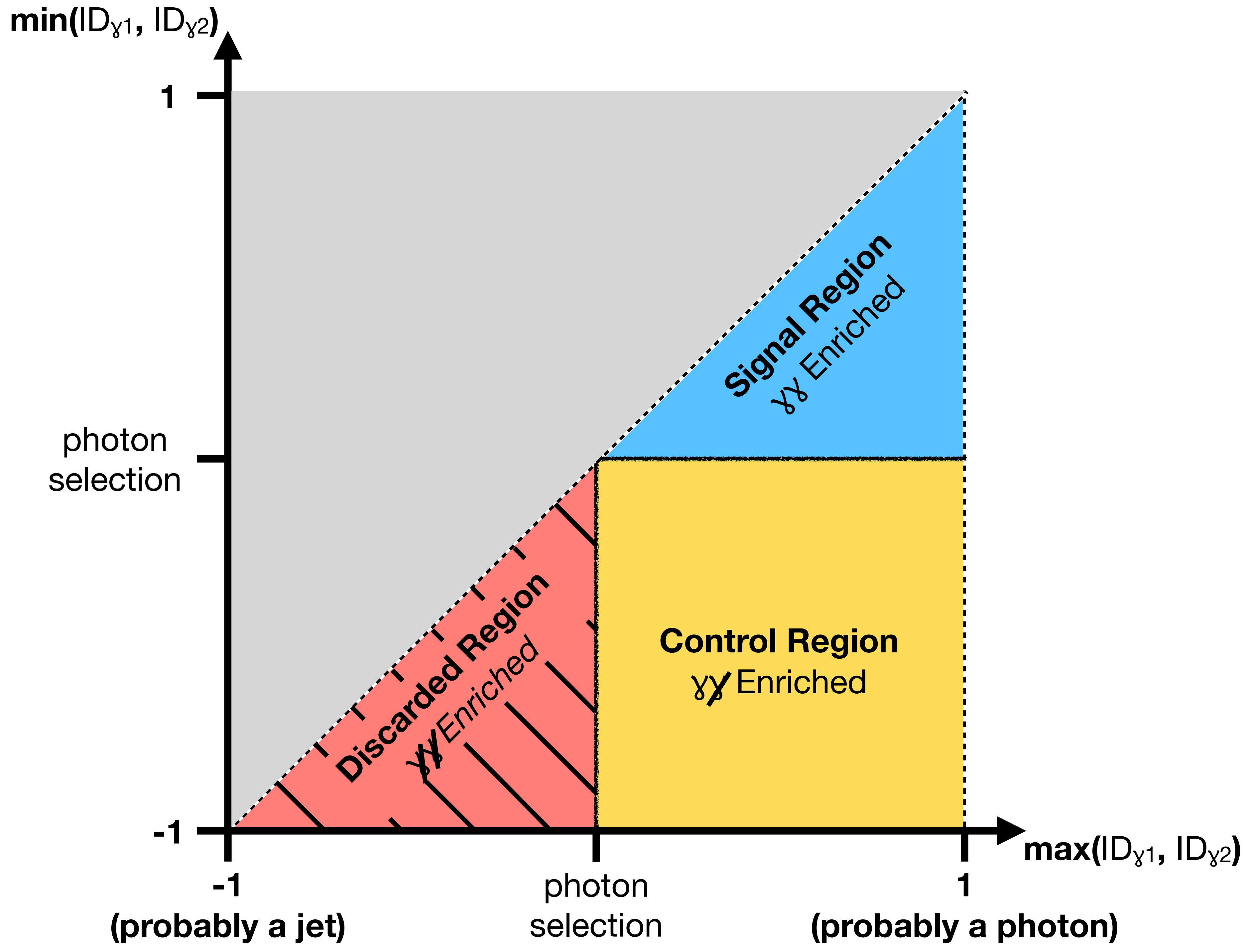
Quantile Transformation used in preprocessing



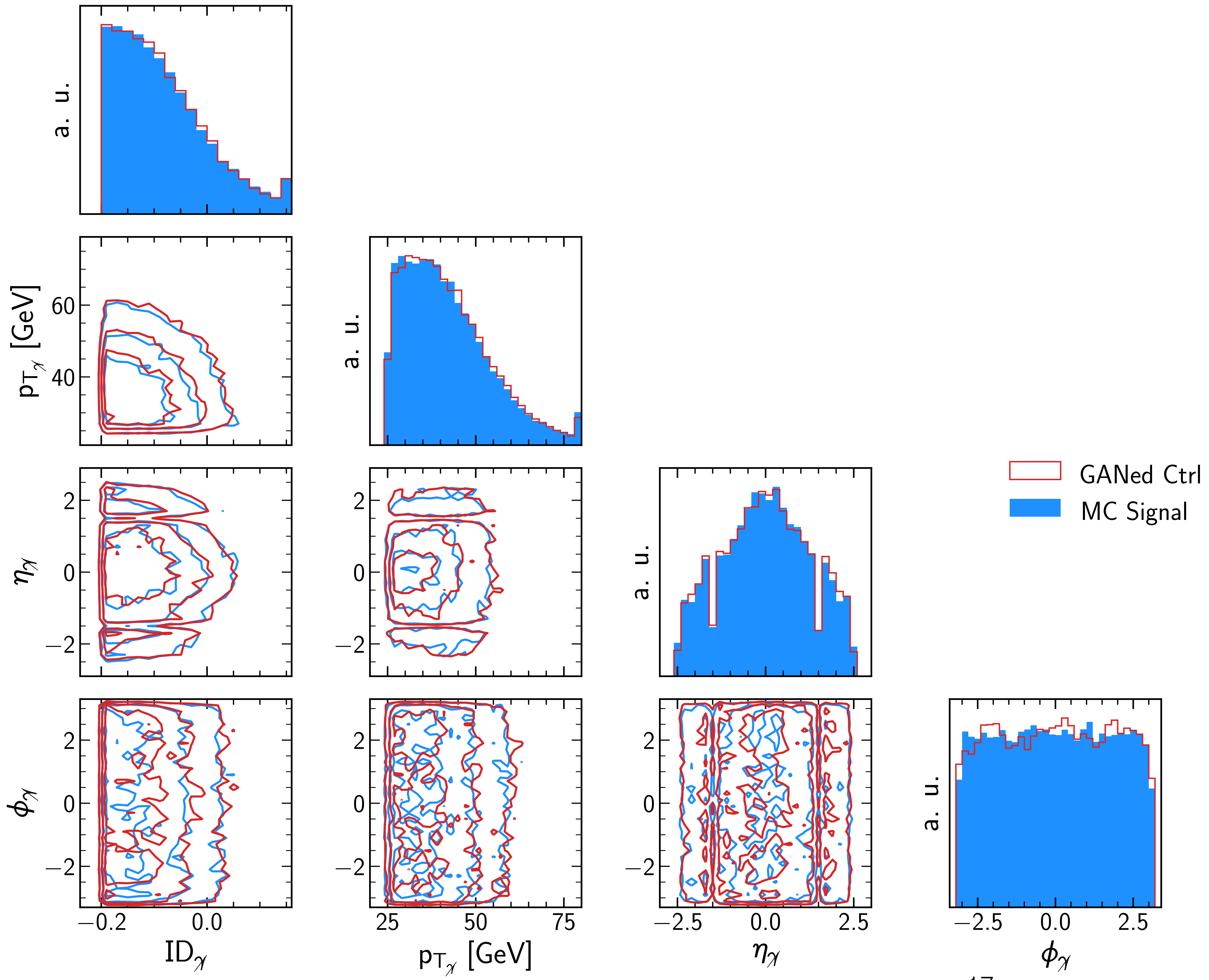
➡ Transformation helps the GAN recover the gaps in η and the core of the ID and p_T distributions

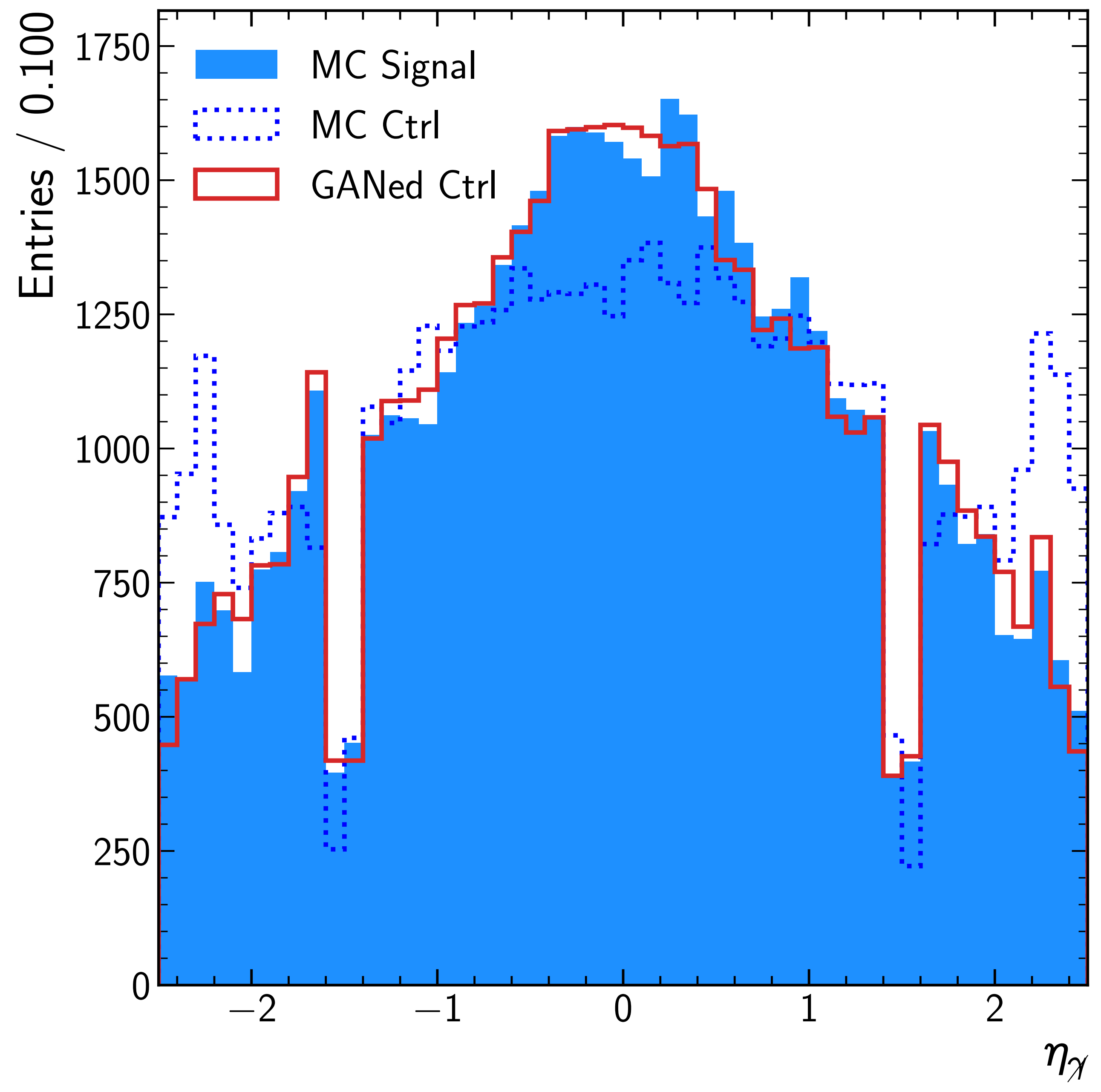
IV.b - Applying GAN to MC control region

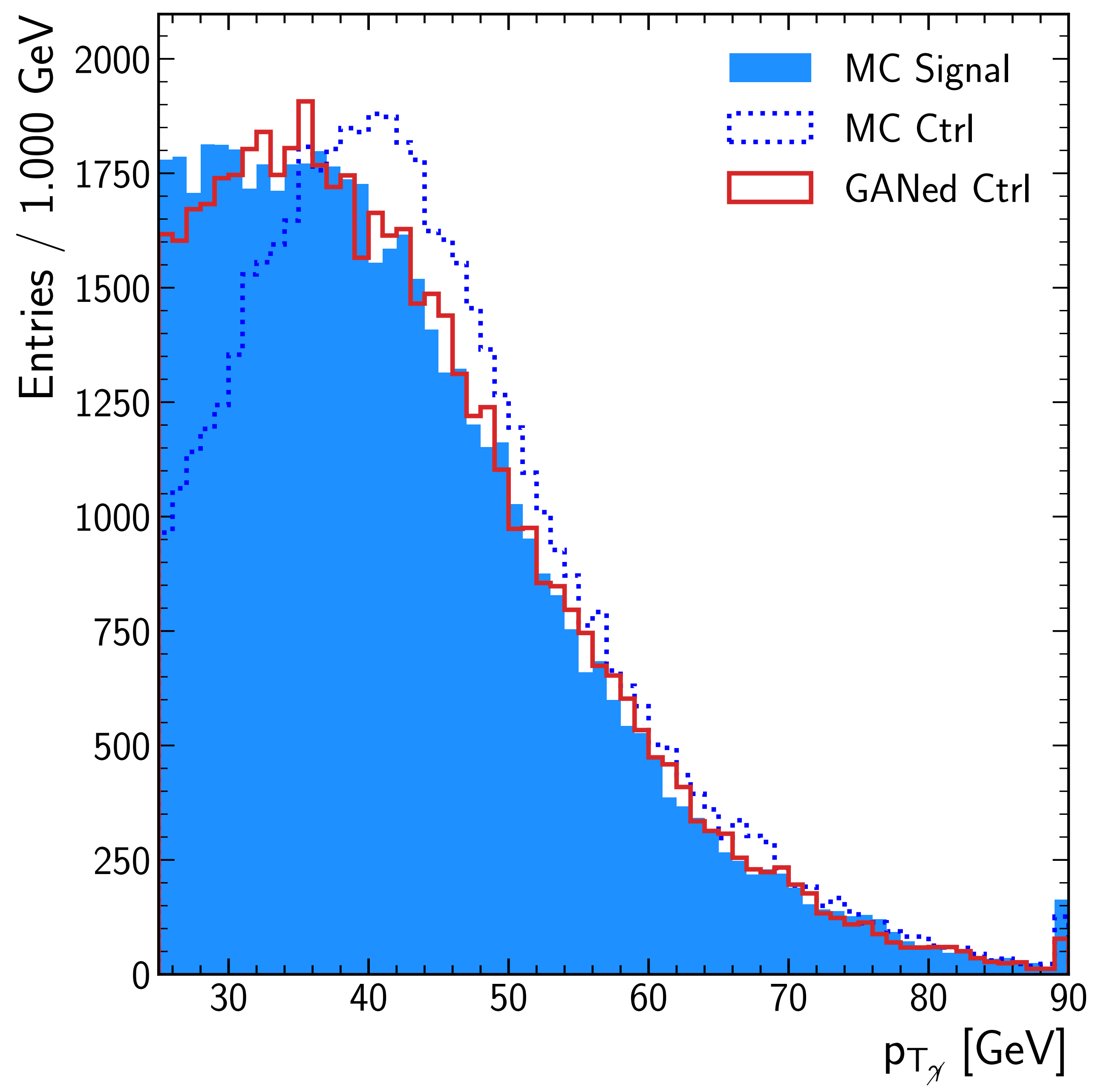




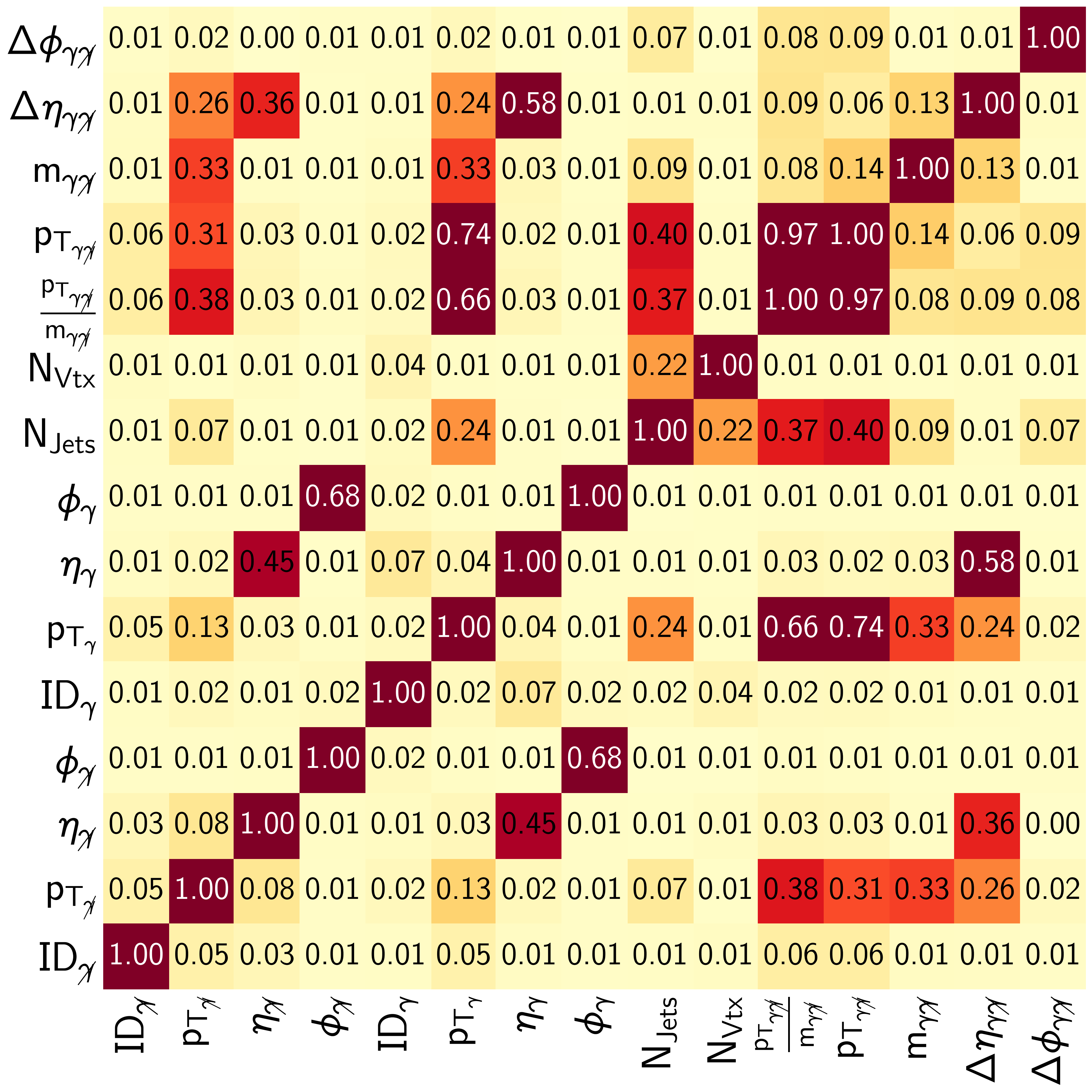
IV.b - Applying GAN to MC control region



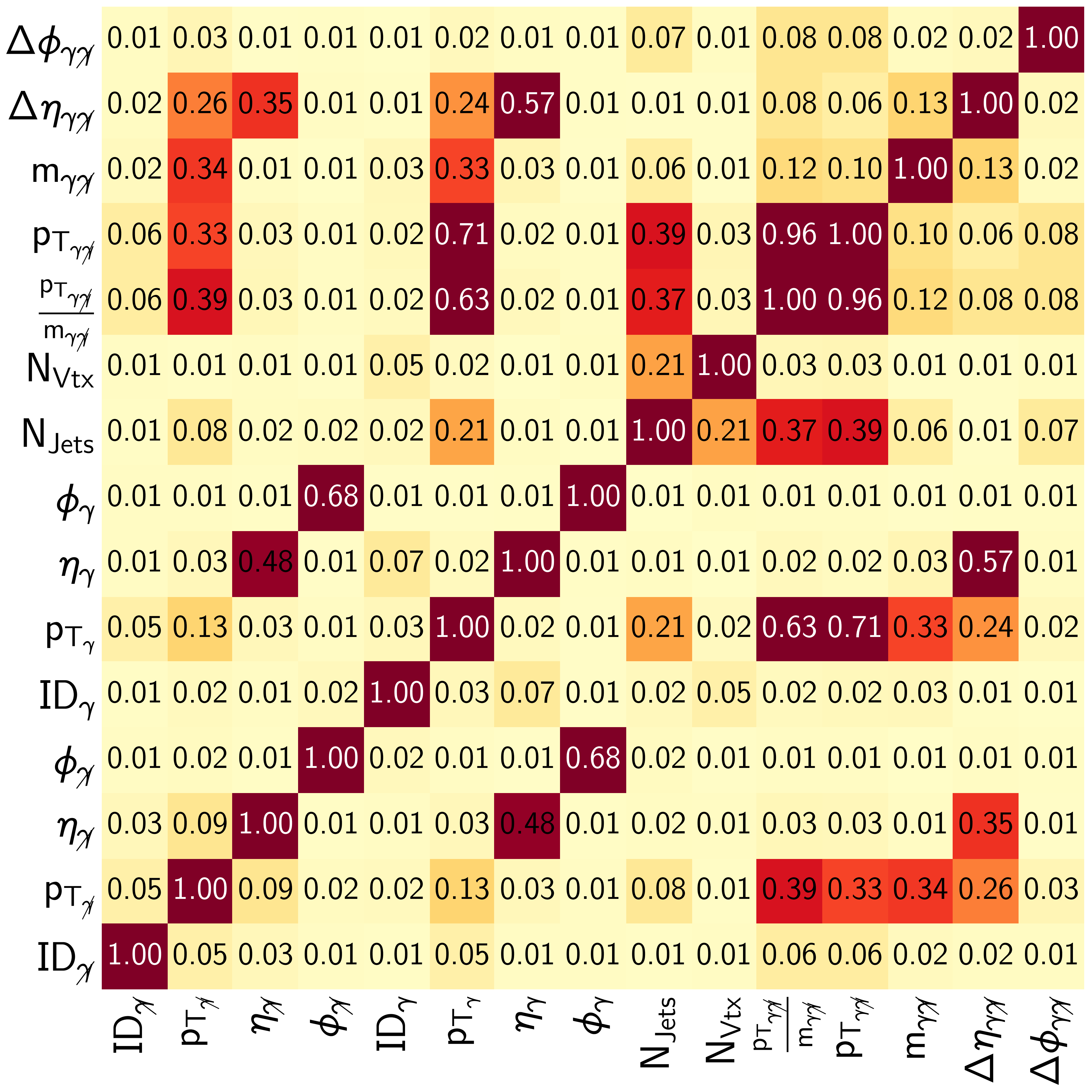




MC Signal Region



GANed Control Region



- Distance correlation coefficients computed to estimate any correlation between observables (not only linear correlations)

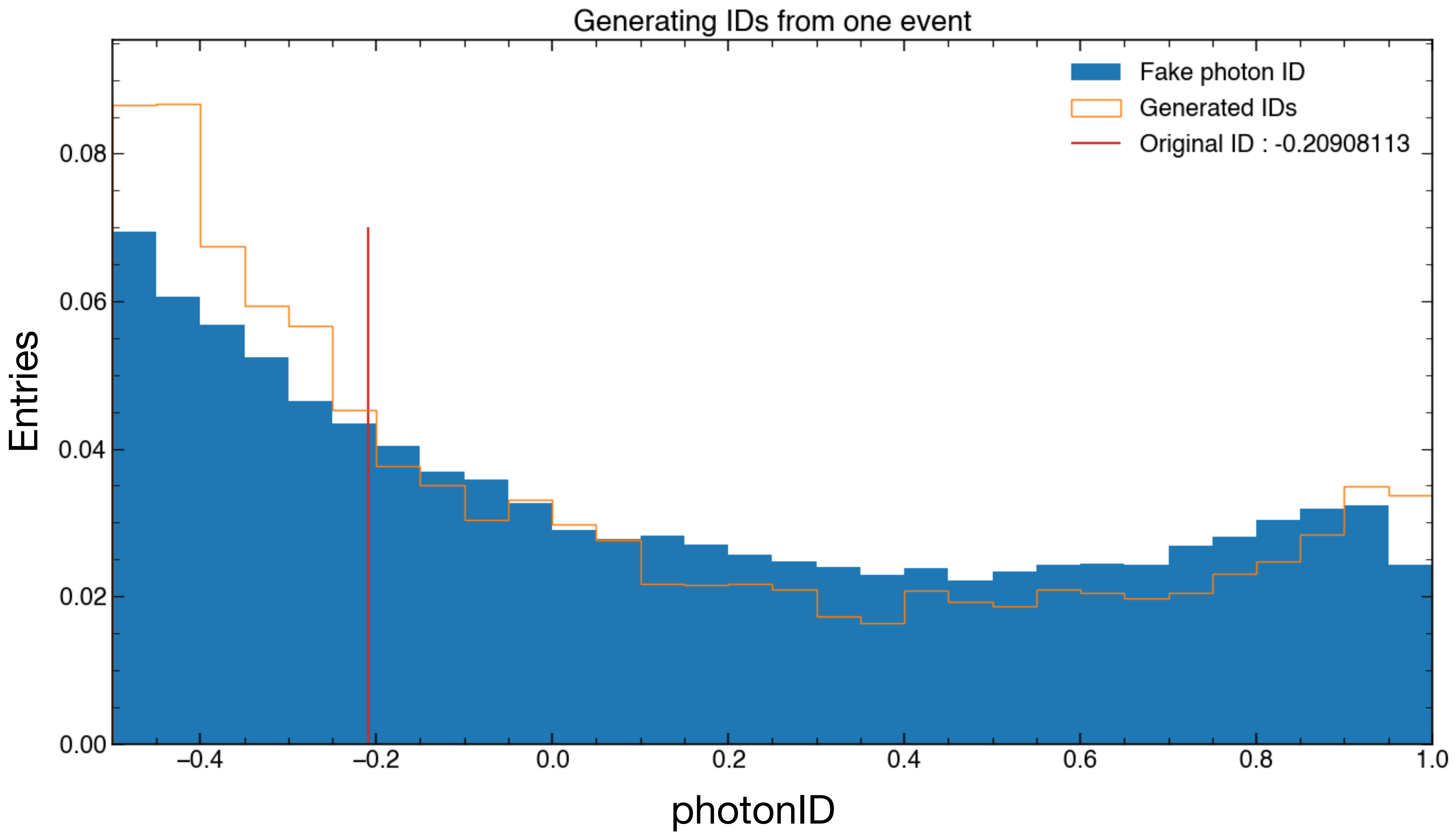
➡ Matrices look almost identical

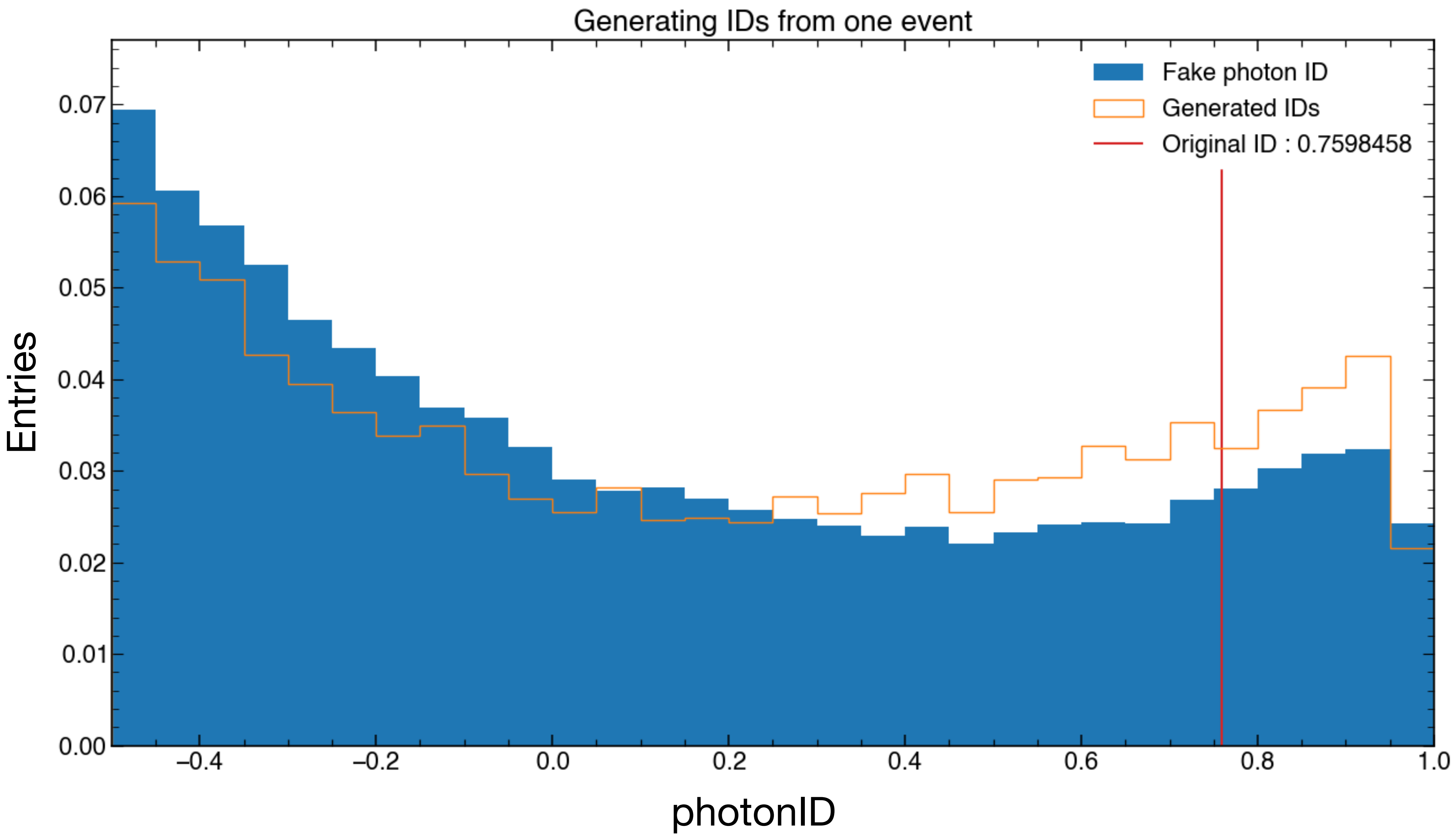
IV. Conclusions and outlooks

- We developed an evaluation procedure to test the GAN's performance and pick the best performing one
- Thanks to GAN we can generate a misidentified photon mimicking the behaviour of an object passing the photon selection criteria
- The produced sample can be used for any H → $\gamma\gamma$ analysis
- This method can be used as a general tool to generate other objects
- Next steps :
  - Publication of the general method
  - Apply the procedure on data to generate a new γ + Jets background sample for H → $\gamma\gamma$ analysis

Backup

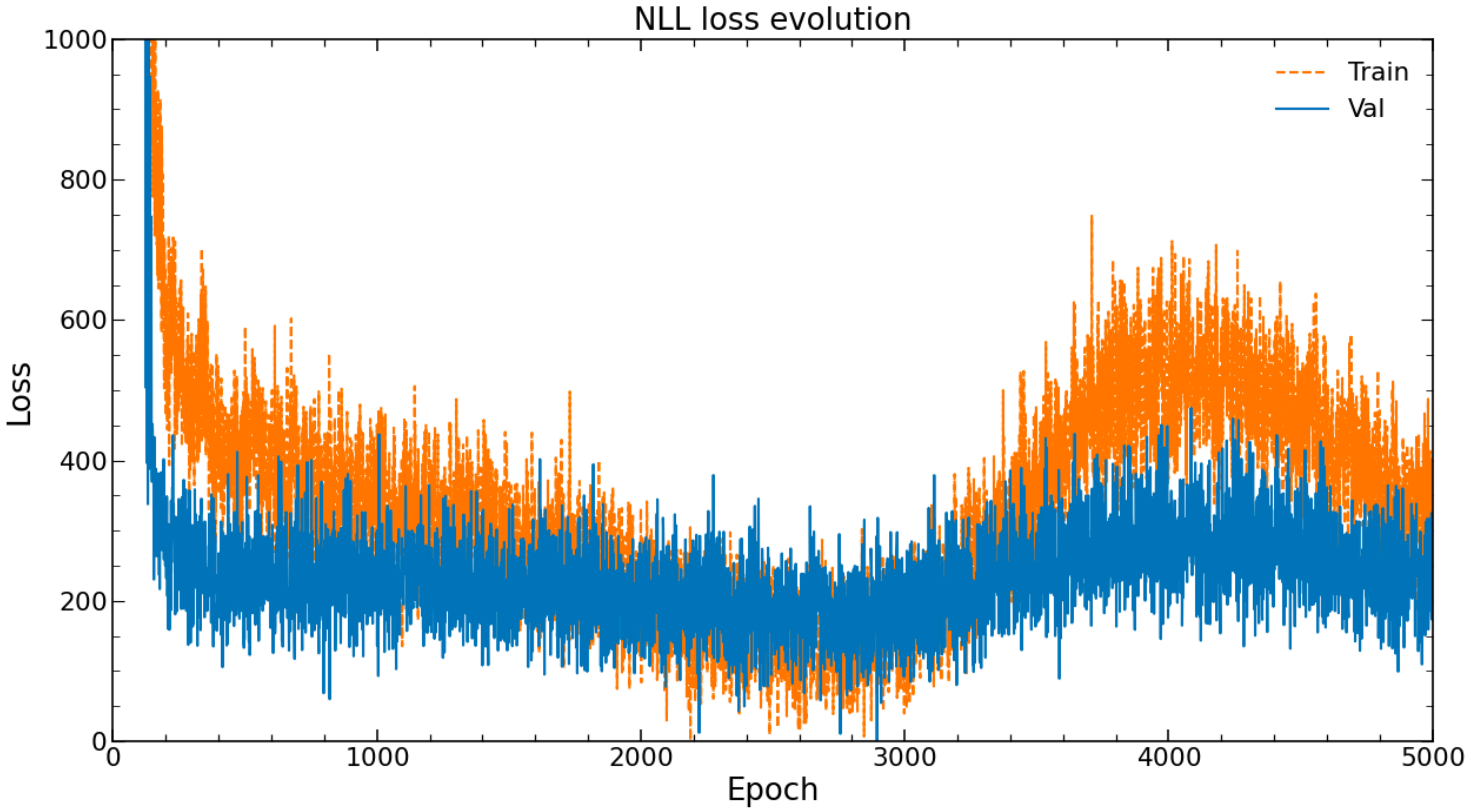
Generating several times per event



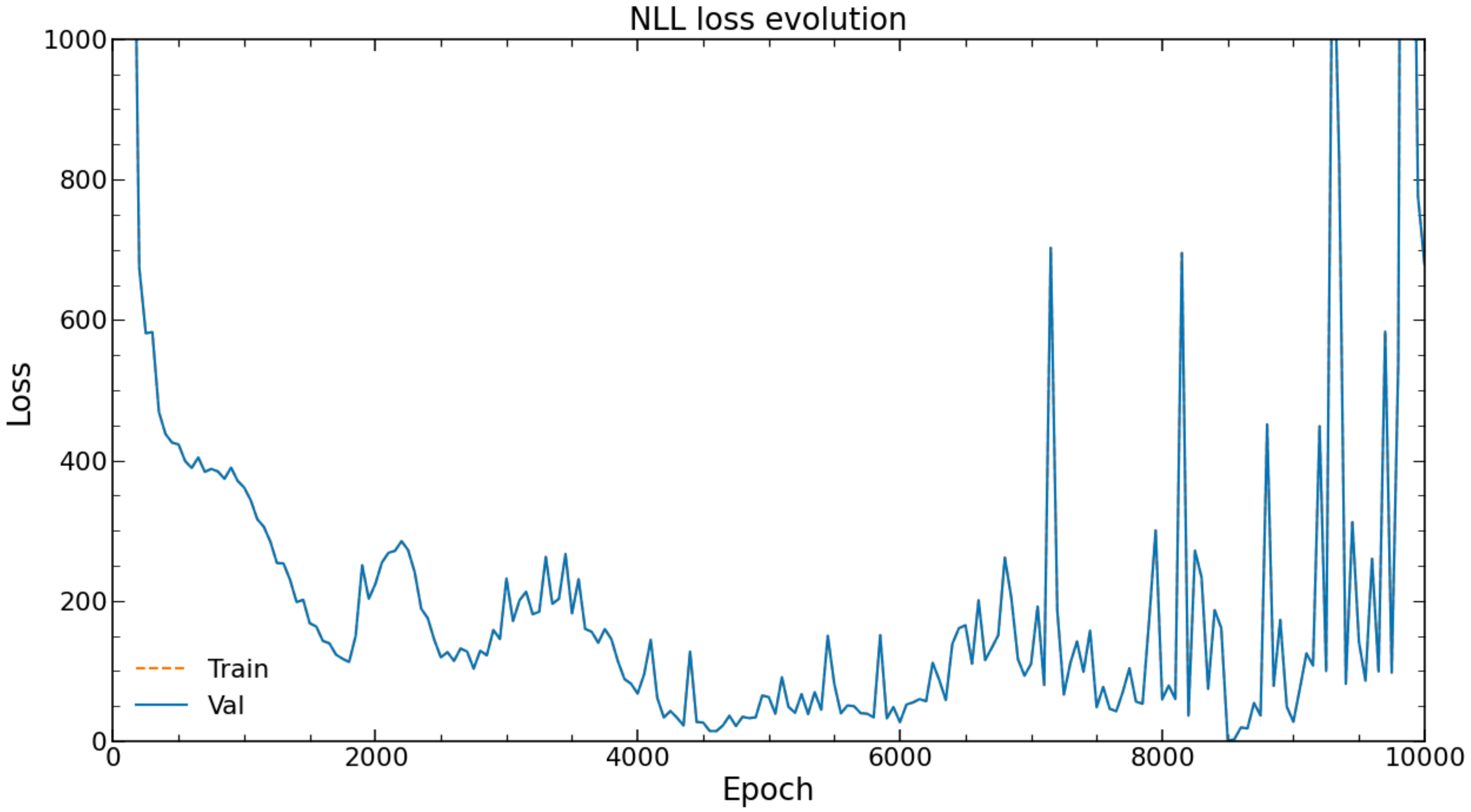


➡ By picking several random inputs and generating several times each output, we average over the random space

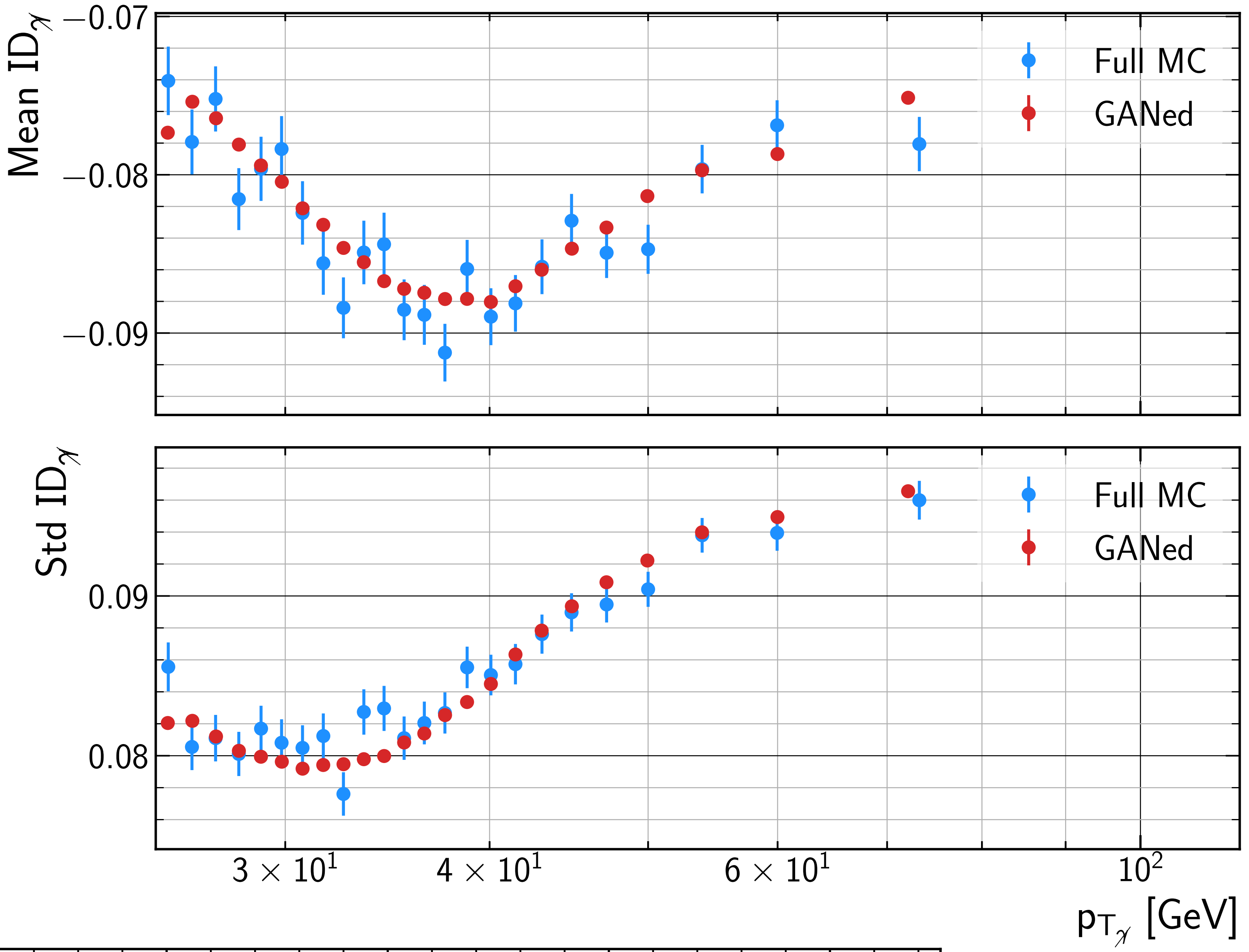
Generating 1 ID per event

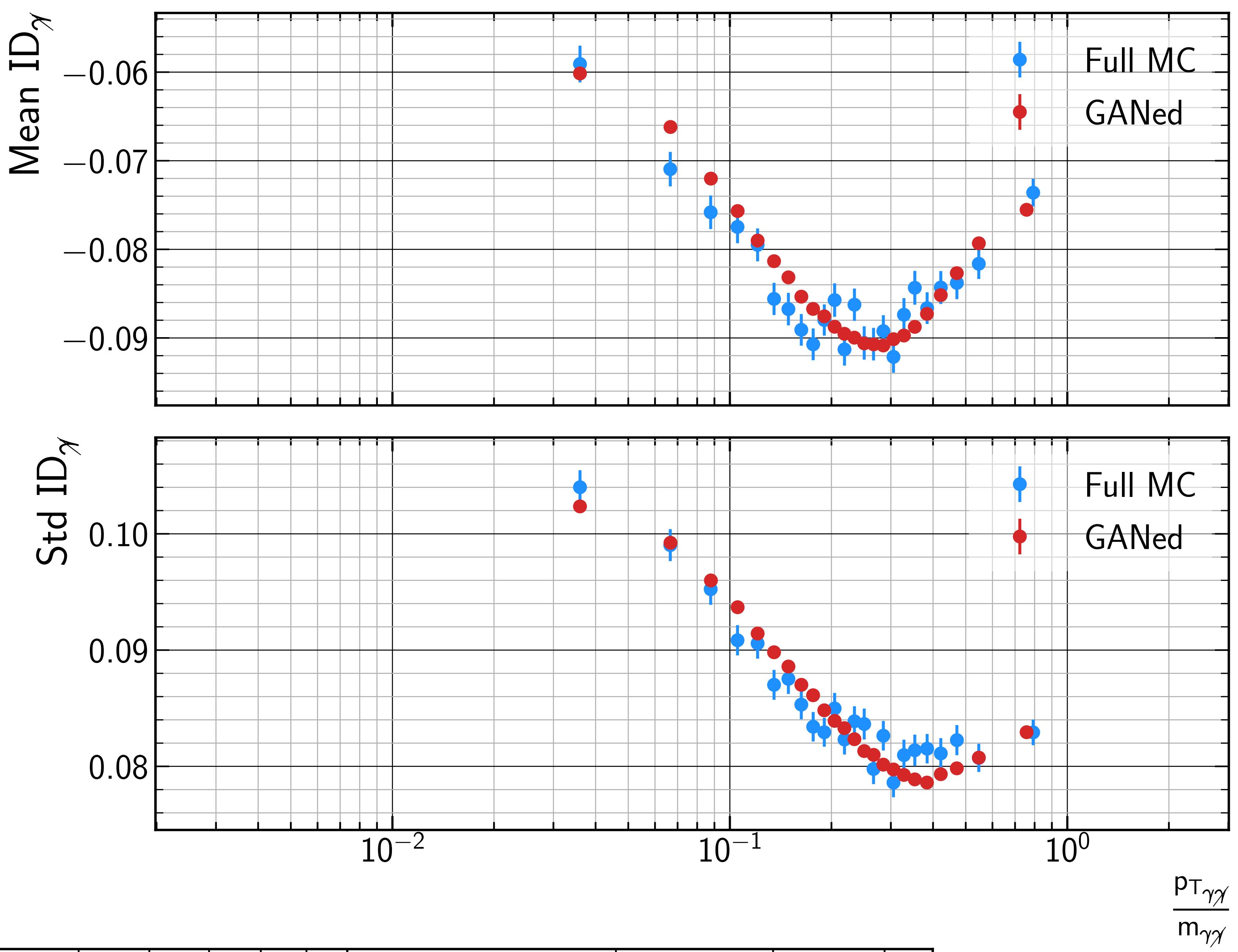


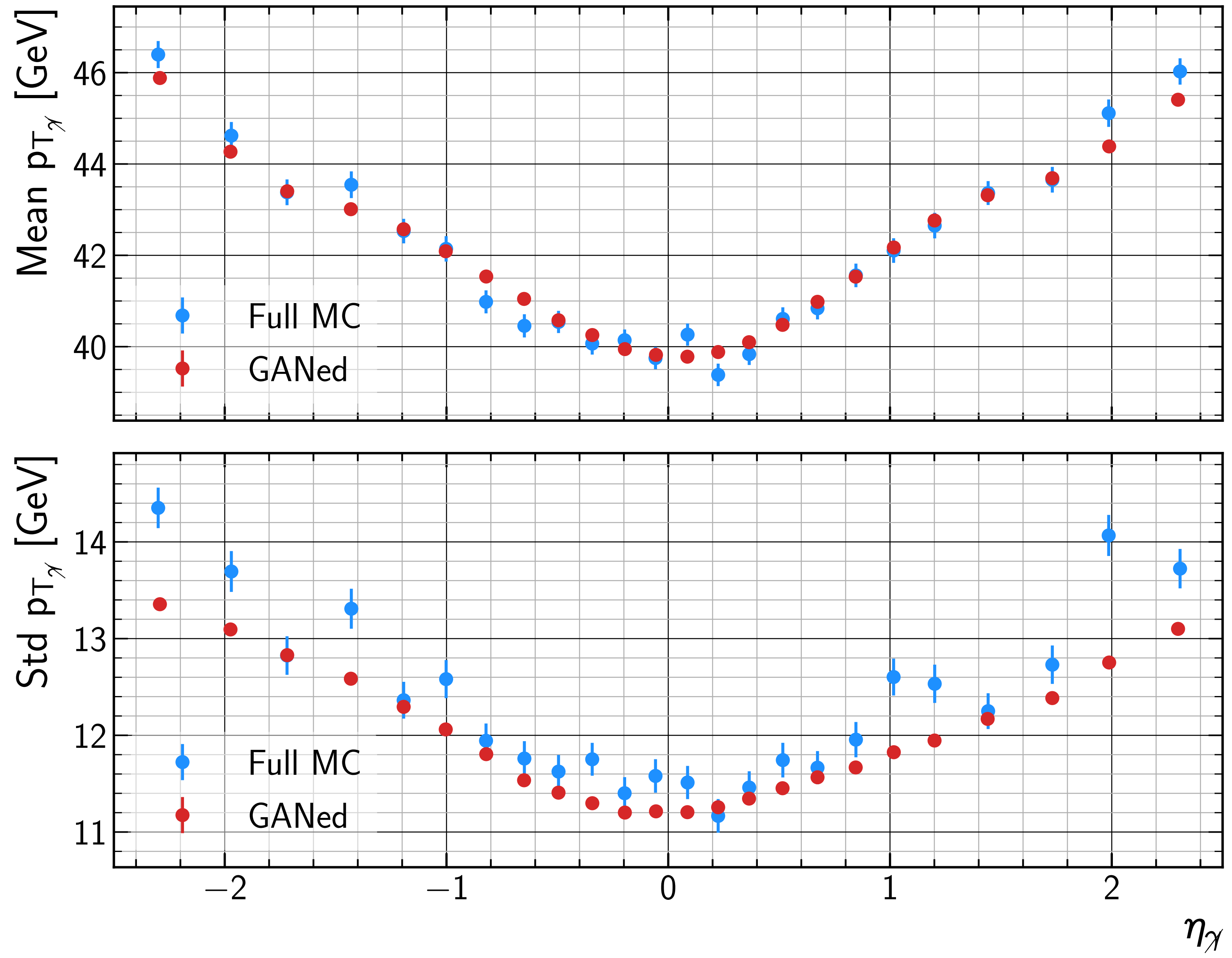
Generating 100 ID per event

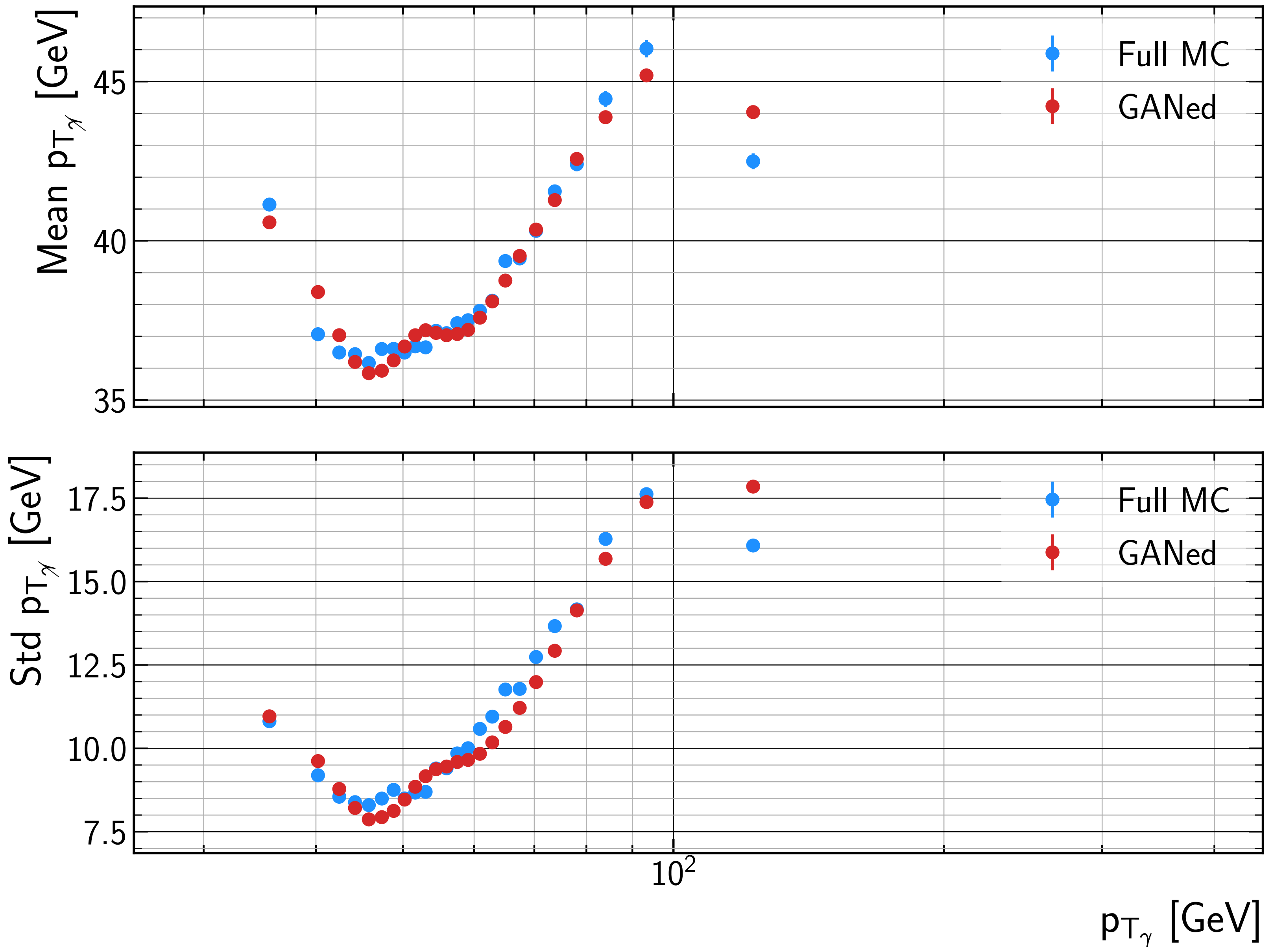


Other way to look at correlation between pair of variables. Plot events in 2D histograms and compute the average (and standard deviation) over the y values per x bin.



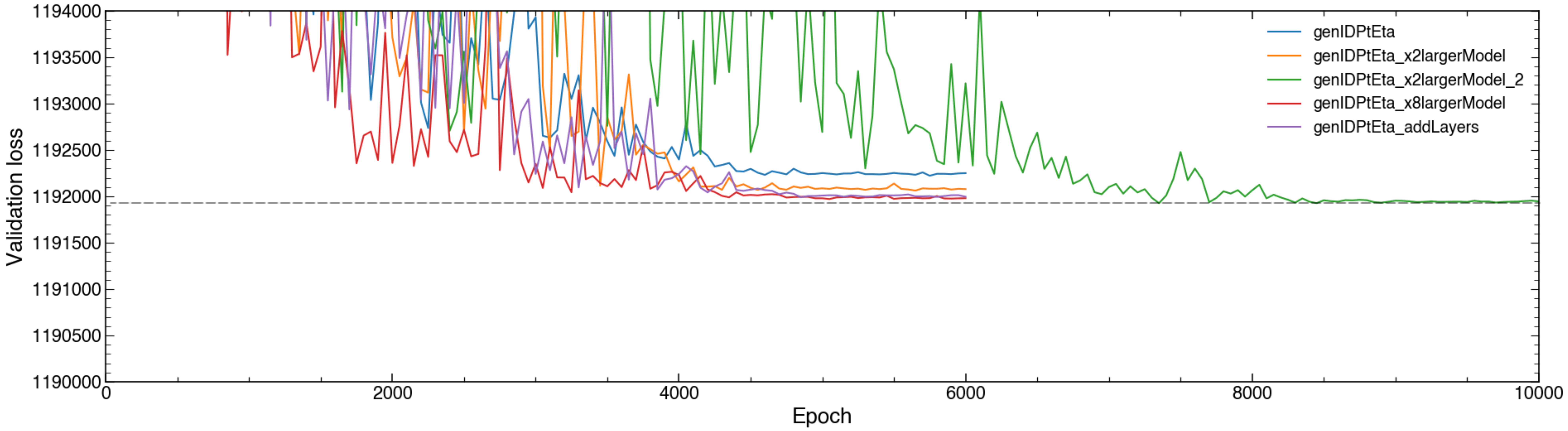






Improving the training for generation of the full object

We can upscale the training in different ways : using larger layers, adding layers, training for more epochs, …



➡ Each of these tests increase the performance of the GAN but the training time as well. Need to fix the limit where better performance is not worth the training time.