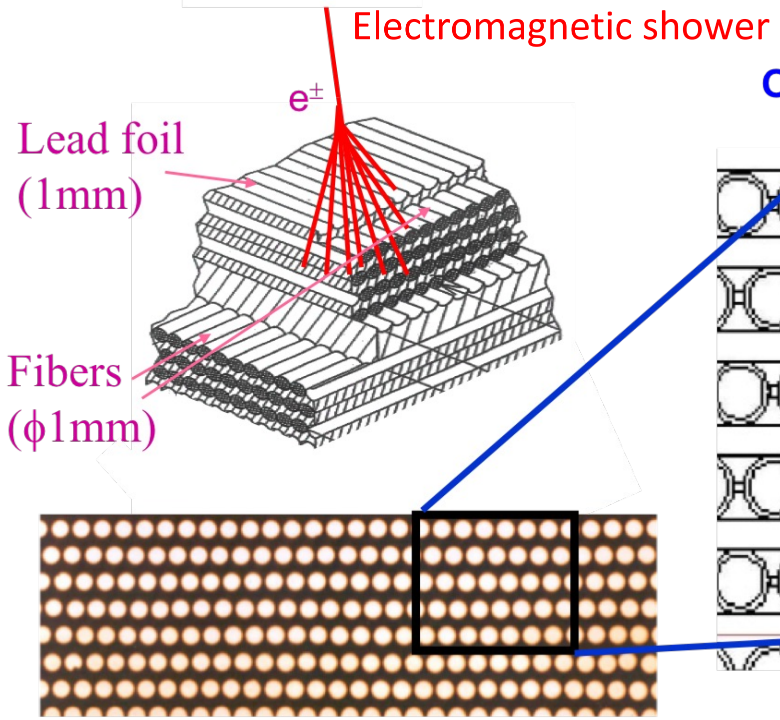
ZDC - EIC

Hsin-Yi Chou

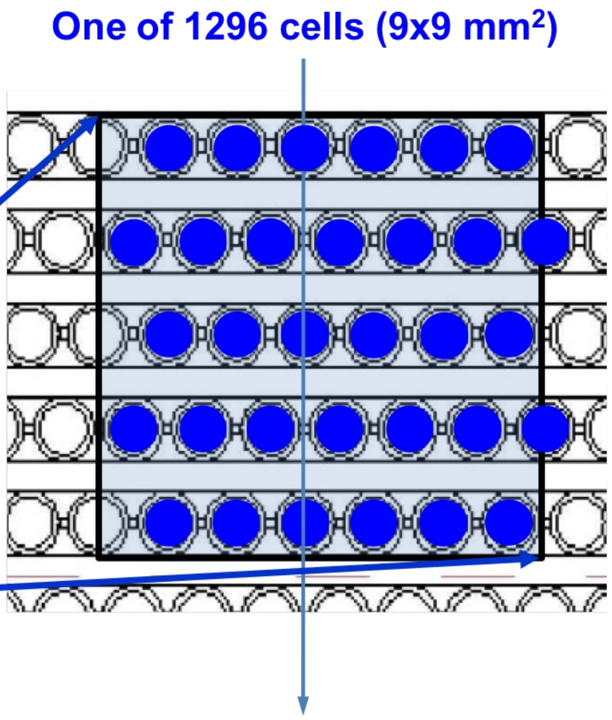
Institute of Physics, Academia Sinica

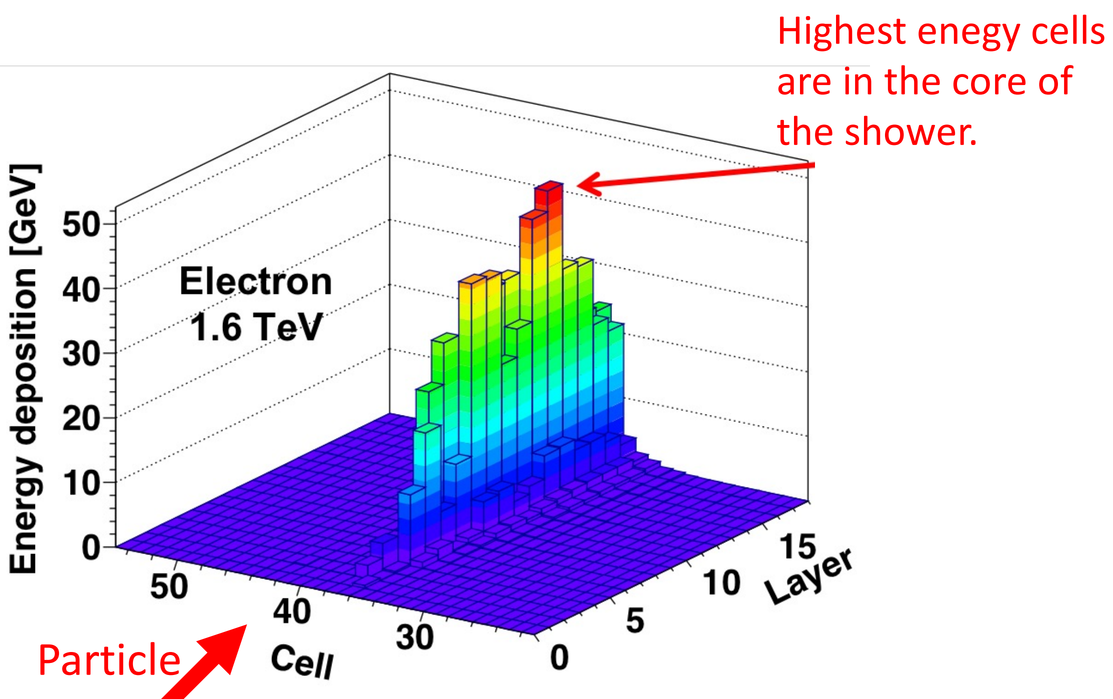
Nov 06, 2024

AMS Electromagnetic Calorimeter (ECAL)



50,000 fibers, ϕ = 1 mm
distributed uniformly inside 1,200 lb of lead





1296 cells = 72 cells/layer x 18 layers

Direction Projection of Layers (10 in Z-Y, 8 in Z-X)

Y	Y	X	X	Y	Y	X	X	Y	Y	X	X	Y	Y	X	X	Y	Y

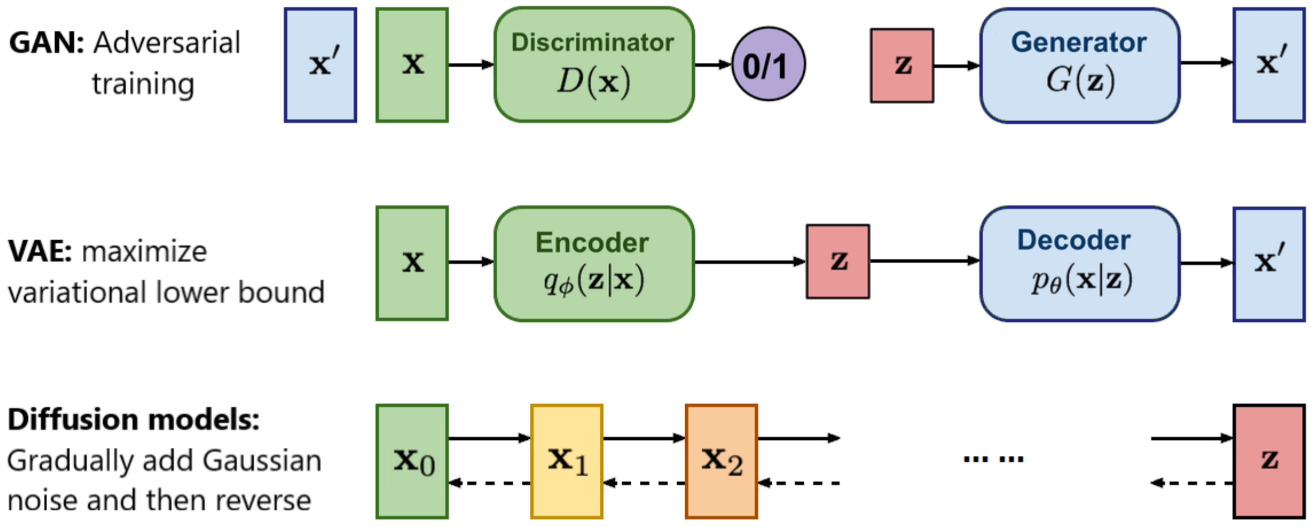
Generator

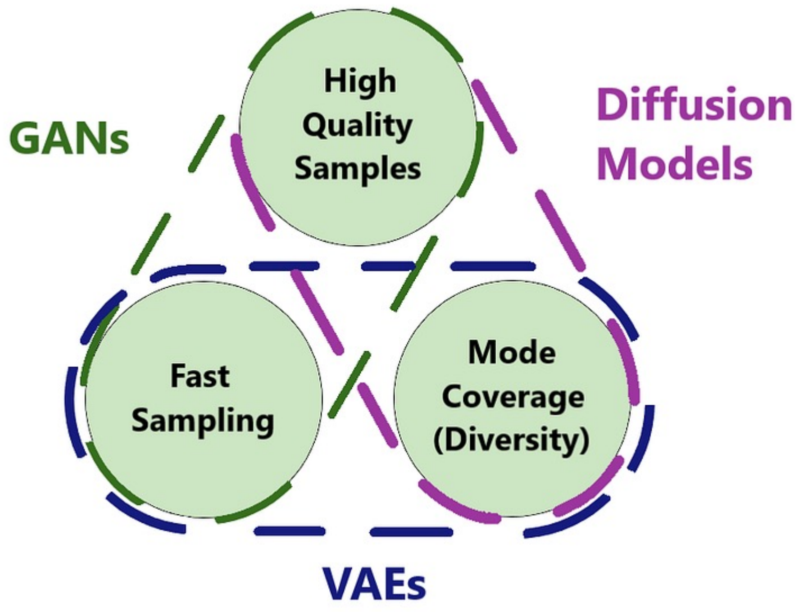
https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9ae

GANs [1, 2] learn to generate new data similar to a training dataset. It consists of two neural networks, a generator, and a discriminator, that play a two-player game. The generator takes in random values sampled from a normal distribution and produces a synthetic sample, while the discriminator tries to distinguish between the real and generated sample. The generator is trained to produce realistic output that can fool the discriminator, while the discriminator is trained to correctly distinguish between the real and generated data. The top row of Figure 1 shows the scheme of its work.

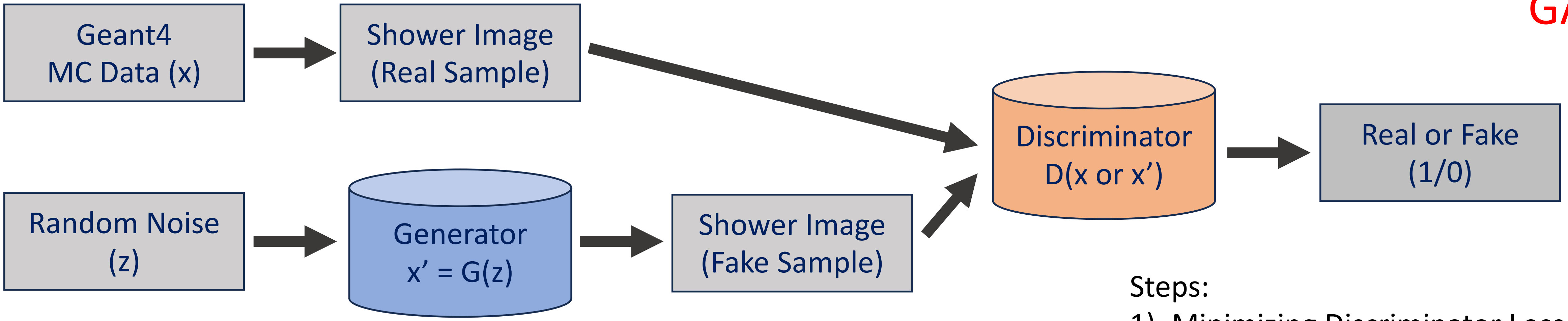
VAEs [3, 4] consist of an encoder and a decoder. The encoder maps high-dimensional input data into a low-dimensional representation, while the decoder attempts to reconstruct the original high-dimensional input data by mapping this representation back to its original form. The encoder outputs the normal distribution of the latent code as a low-dimensional representation by predicting the mean and standard deviation vectors. The middle row of Figure 1 demonstrates its work.

Diffusion models [5, 6] consist of forward diffusion and reverse diffusion processes. Forward diffusion is a Markov chain that gradually adds noise to input data until white noise is obtained. It is not a learnable process and typically takes 1000 steps. The reverse diffusion process aims to reverse the forward process step by step removing the noise to recover the original data. The reverse diffusion process is implemented using a trainable neural network. The bottom row of Figure 1 shows that.





GAN



Steps:

1). Minimizing Discriminator Loss

2). Minimizing Generator Loss

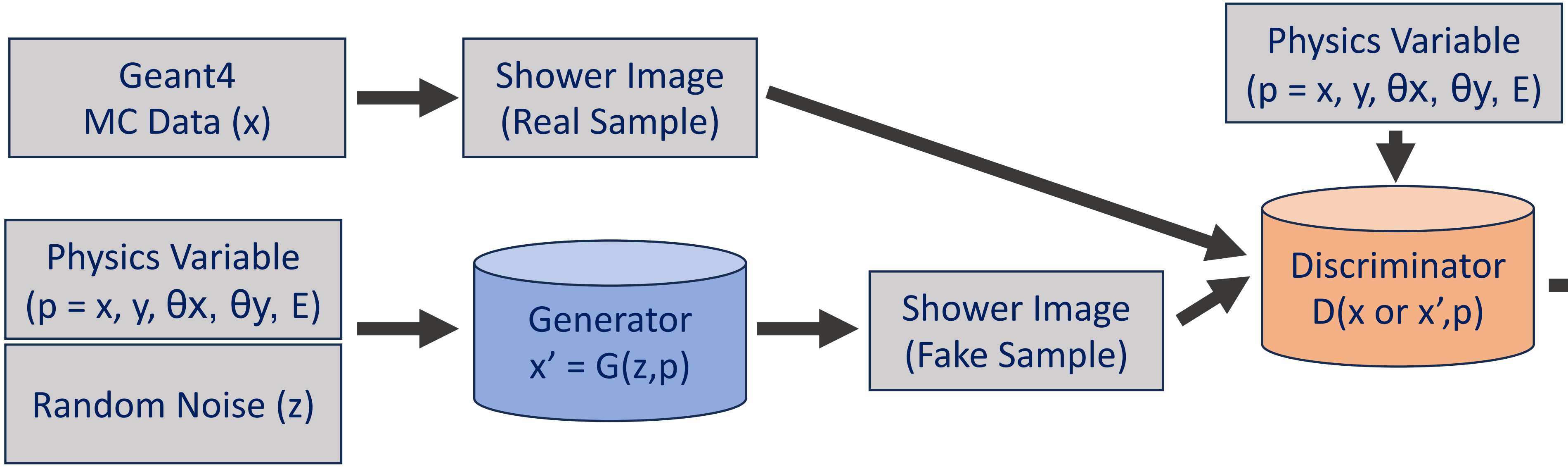
3). Repeat 1) and 2)

Discriminator Loss = -mean{ log[D(x)] + log[1 - D(x’)] } = -{ log[D(x)] + log[1 - D(G(z))] }

Generator Loss = -mean{ log[D(G(z))] }

Conditional GAN

E = 0.25 ~ 4000 GeV



Discriminator Loss = -{ log[D(x,p)] + log[1 - D(G(z,p),p)] }

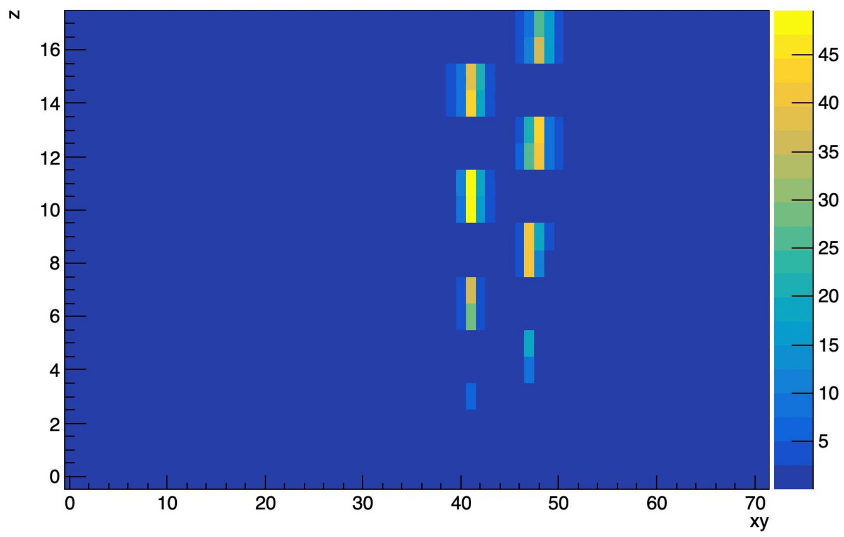
Generator Loss = -log[D(G(z,p))] + λ * L2(G(z,p), x)

L2(G(z,p), x) = mean {(G(z,p)-x)^2}

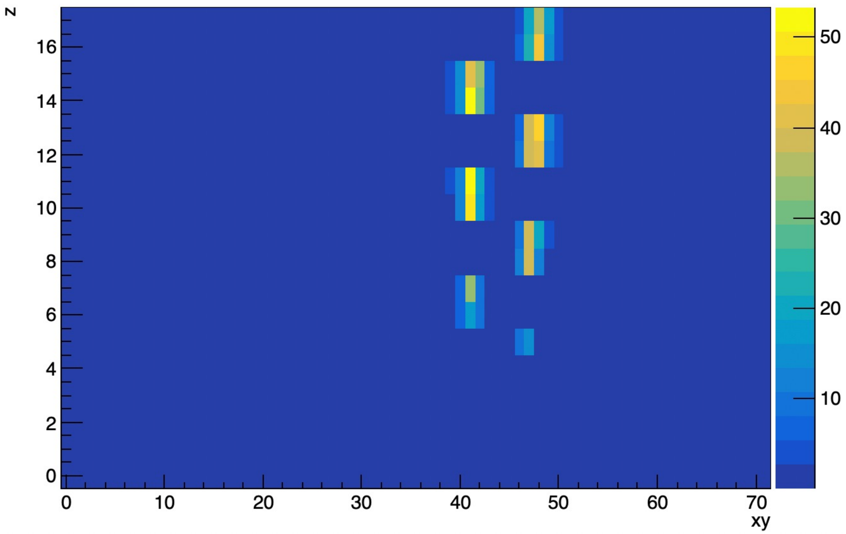
Step 1:

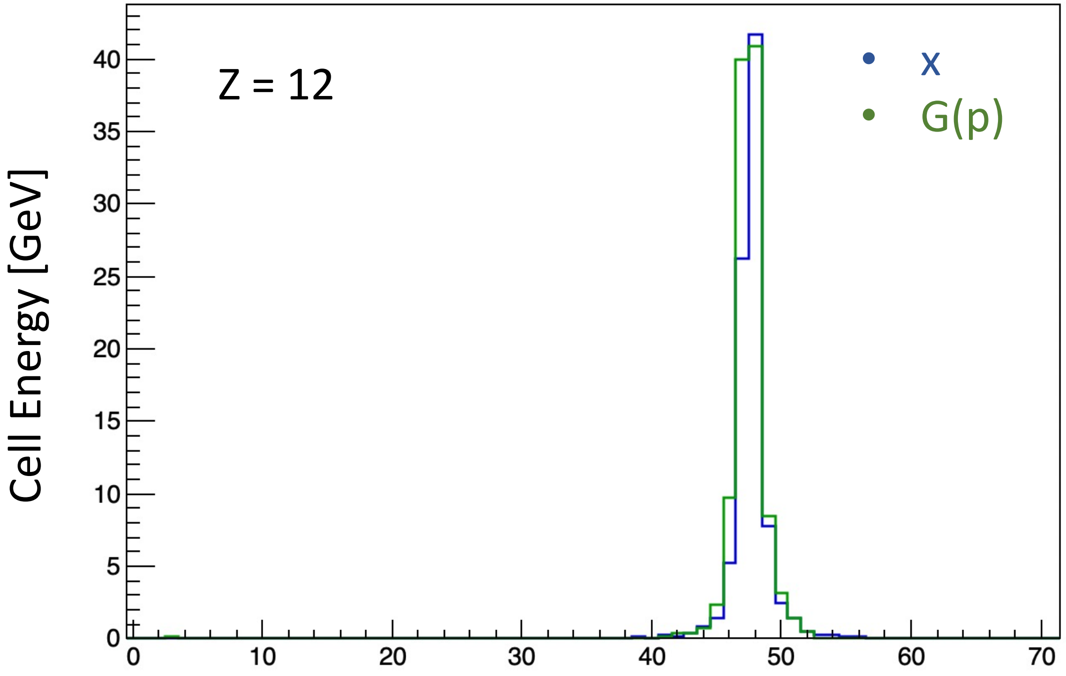
Generator Loss = L2(G(p), x)

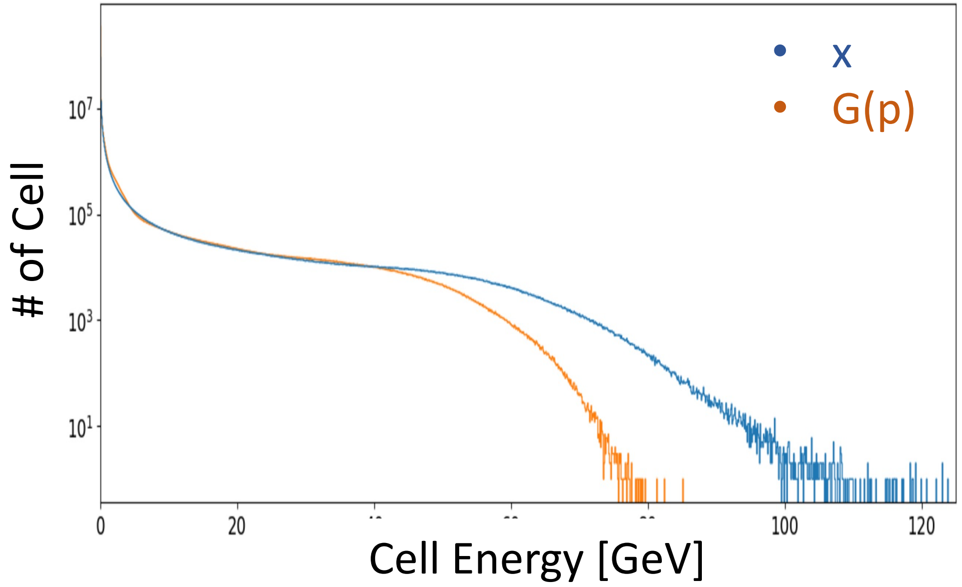
Real image (Geant4 MC) E = 1466 GeV



Reconstruction image G(p)





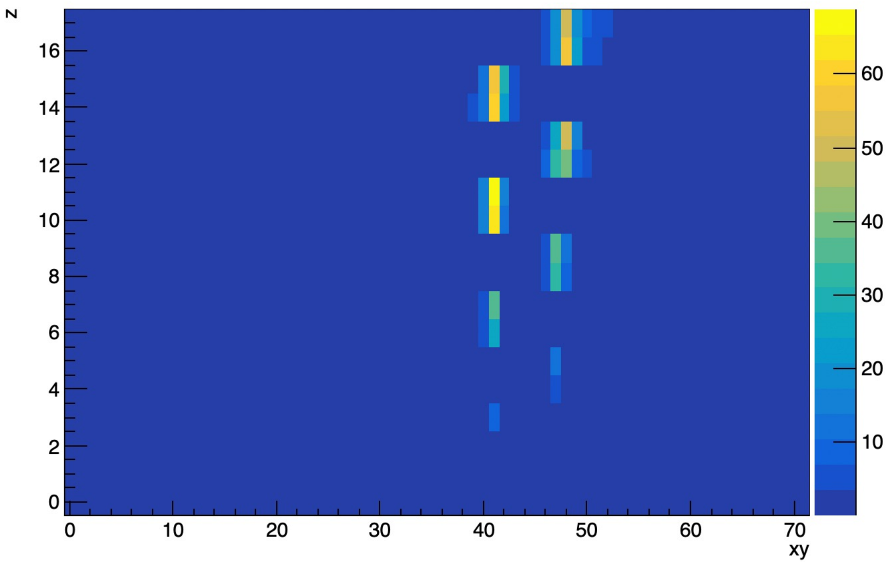


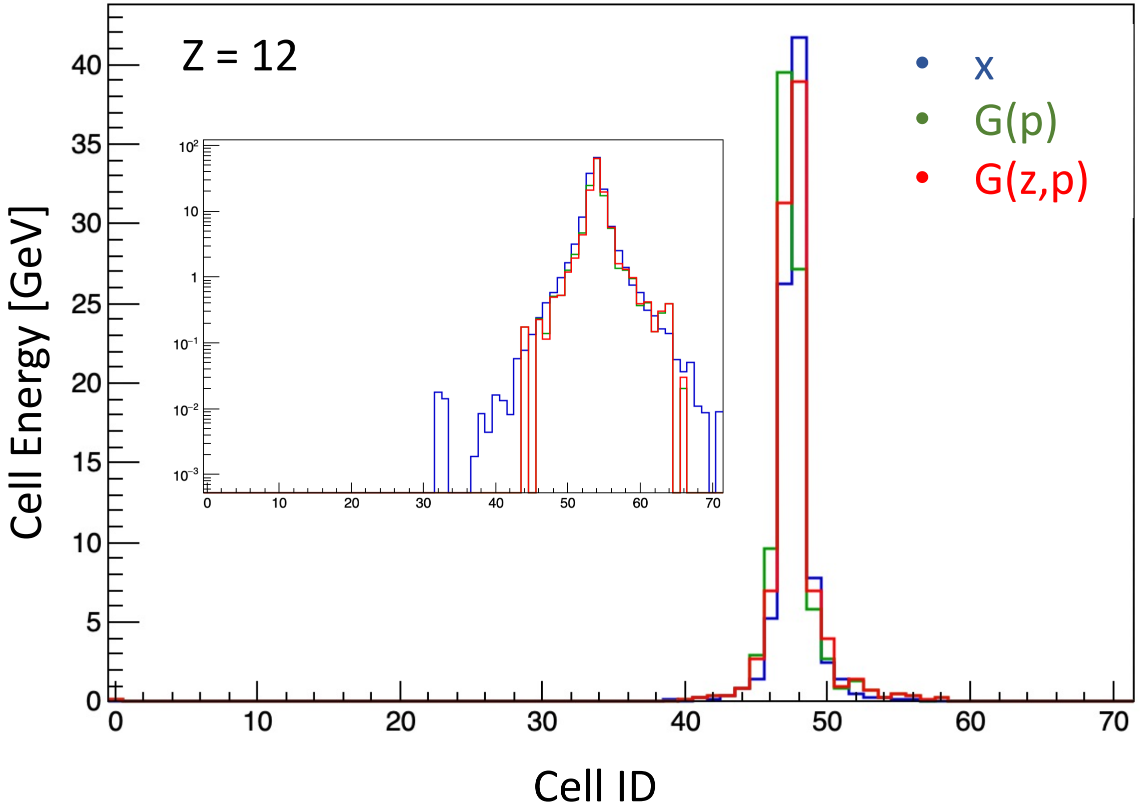
Step 2:

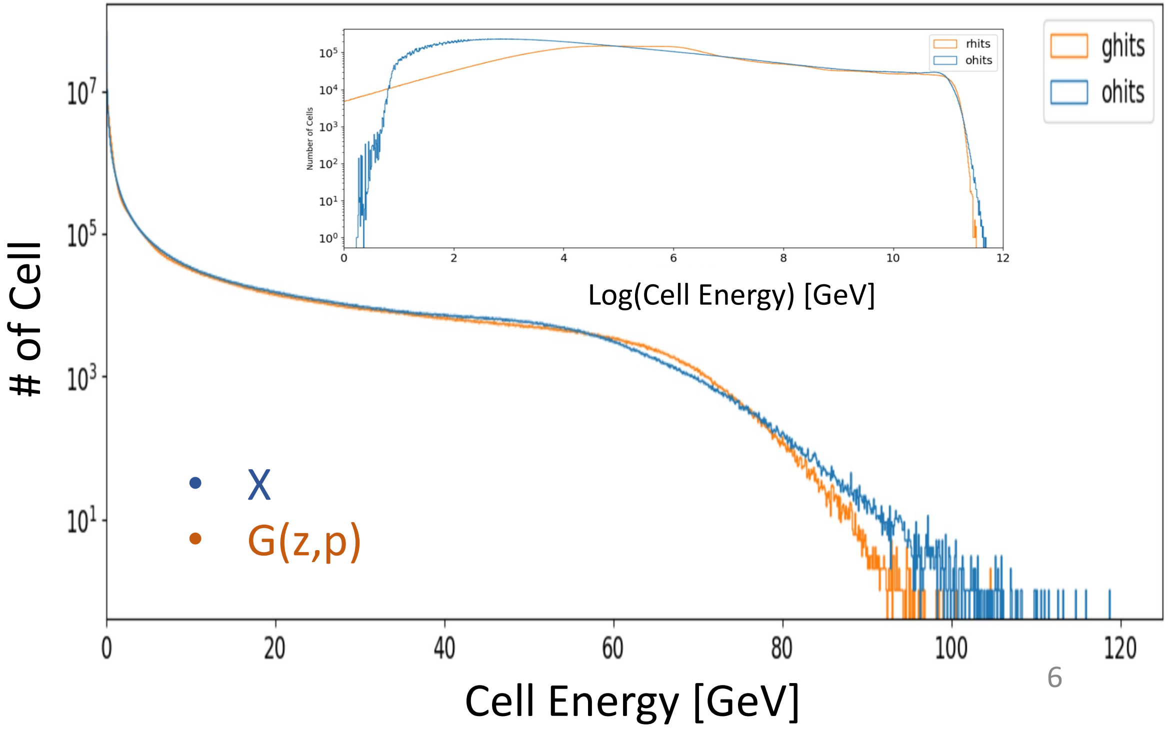
Discriminator Loss = -{ log[D(x,p)] + log[1 - D(G(z,p),p)] }

Generator Loss = -log[D(G(z,p))] + λ * L2(G(z,p), x)

Simulation image G(z,p) E = 1466 GeV







Key Issue

- Pre-processing:
  - Balanced samples
  - Data normalization (e.g. cell energy $10^{-3} \sim 10^{2}$)
- Model: VAE, GAN, Diffusion model, …
- Objective Function: Cross entropy loss, L2 loss, …
- Evaluation Methodology (Metric): cell energy distribution, …