Generative Models for Calorimeter Showers

Engin Eren, Sascha Diefenbacher
Gregor Kasieczka, Frank Gaede
Ties Behnke, Anatoli Korol, Erik Buhmann

AMALEA Project Meeting
03.12.2019

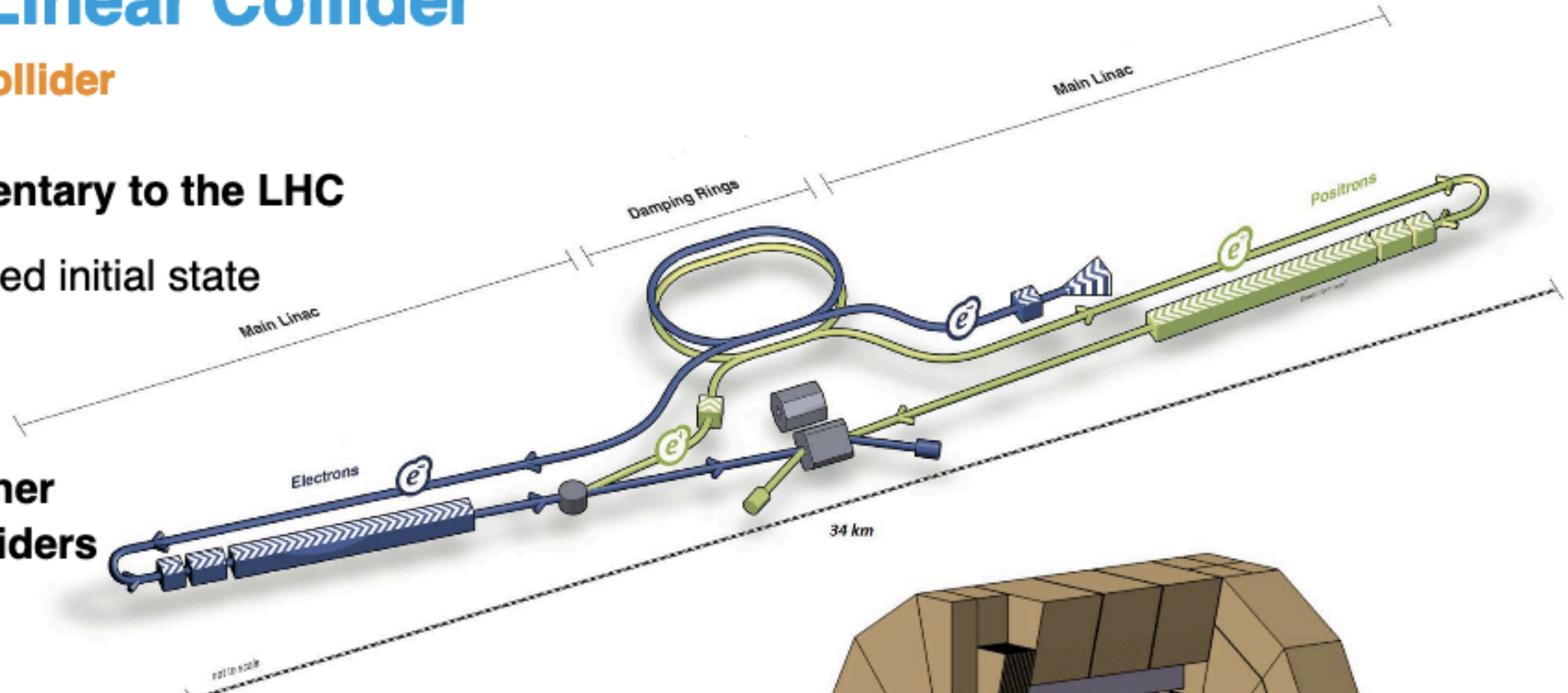
A bit of context..

The International Linear Collider

A planned electron-positron collider

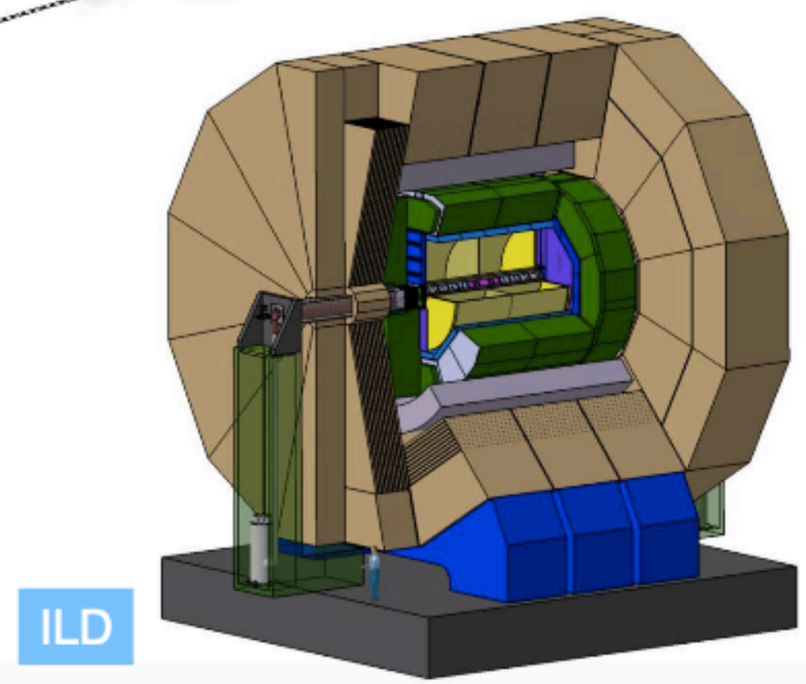
Lepton colliders are complementary to the LHC

- cleaner environment, controlled initial state
- coupling to leptons is tested



Advantages of the ILC over other planned electron-positron colliders

- mature technology
- centre-of-mass energy can be tuned and increased: 250 GeV in initial stage, upgrades to 500 GeV and 1 TeV
- polarisation of both beams: *P(e-)*=±80%, *P(e+)*=∓30%
- triggerless operation
- hermeticity of detector down to lowest angles

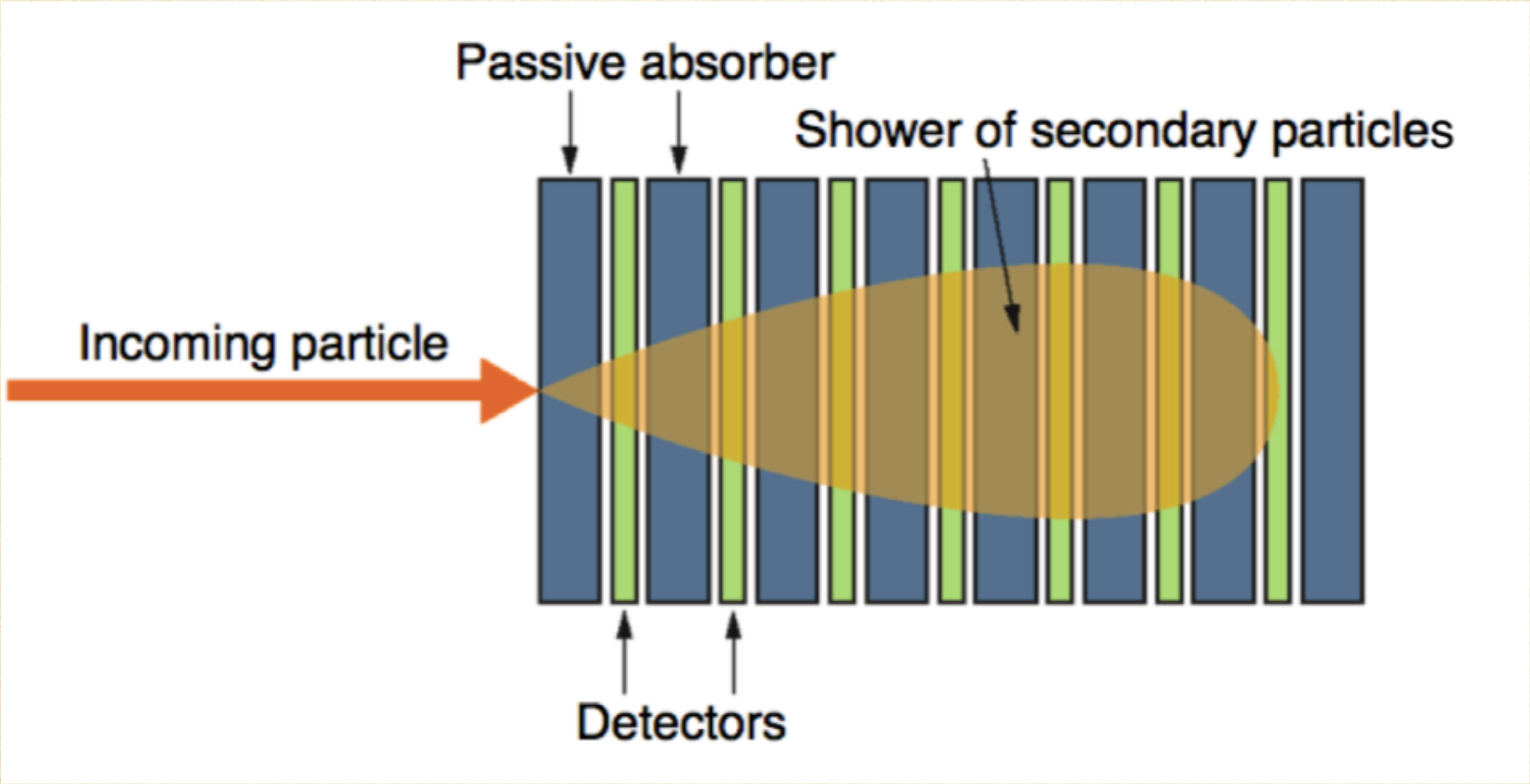


DESY. I Particle Discovery Opportunities at the ILC I M. Habermehl, 12 July 2019 Page 2

The Slide taken from EPS-HEP 2019 Conference, presentation by M.Habermehl

Electromagnetic Showers in a HEP Experiment

- Incoming particle initiates the showers and **secondary particles** are produced
- These secondary particles further produce other particles until the full energy is absorbed



Picture : https://www.hephy.at/fileadmin/user_upload/VO-6-Calorimeters.pdf

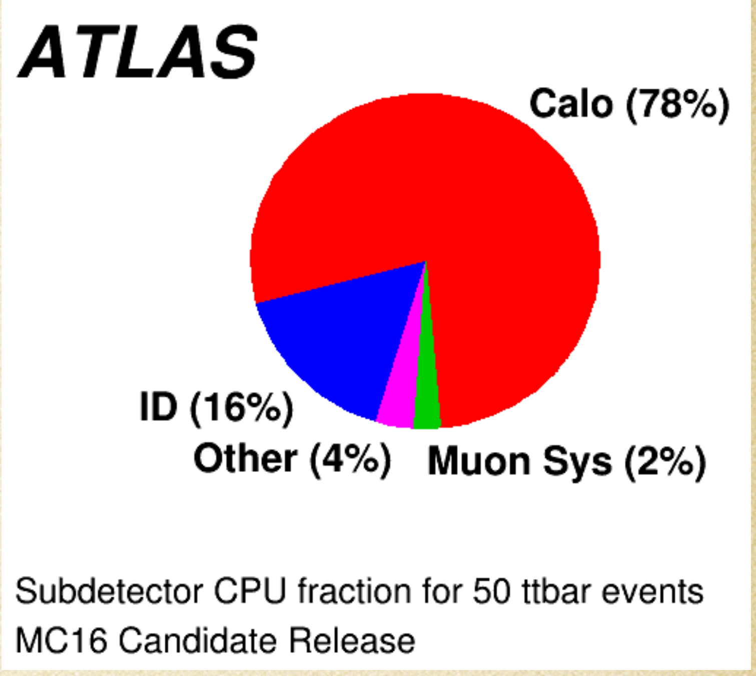
This is one type of EM calorimeter : so-called **sampling calorimeter**

➡ Consists of alternating layers of passive absorbers and active detectors

➡ Only **fraction** of particle energy deposited

Shower Simulation

- Particle showers in the calorimeter are simulated by Geant4
  - State of the art : First principle **physics** based simulation
- CPU intensive; due to large number of interacting particles

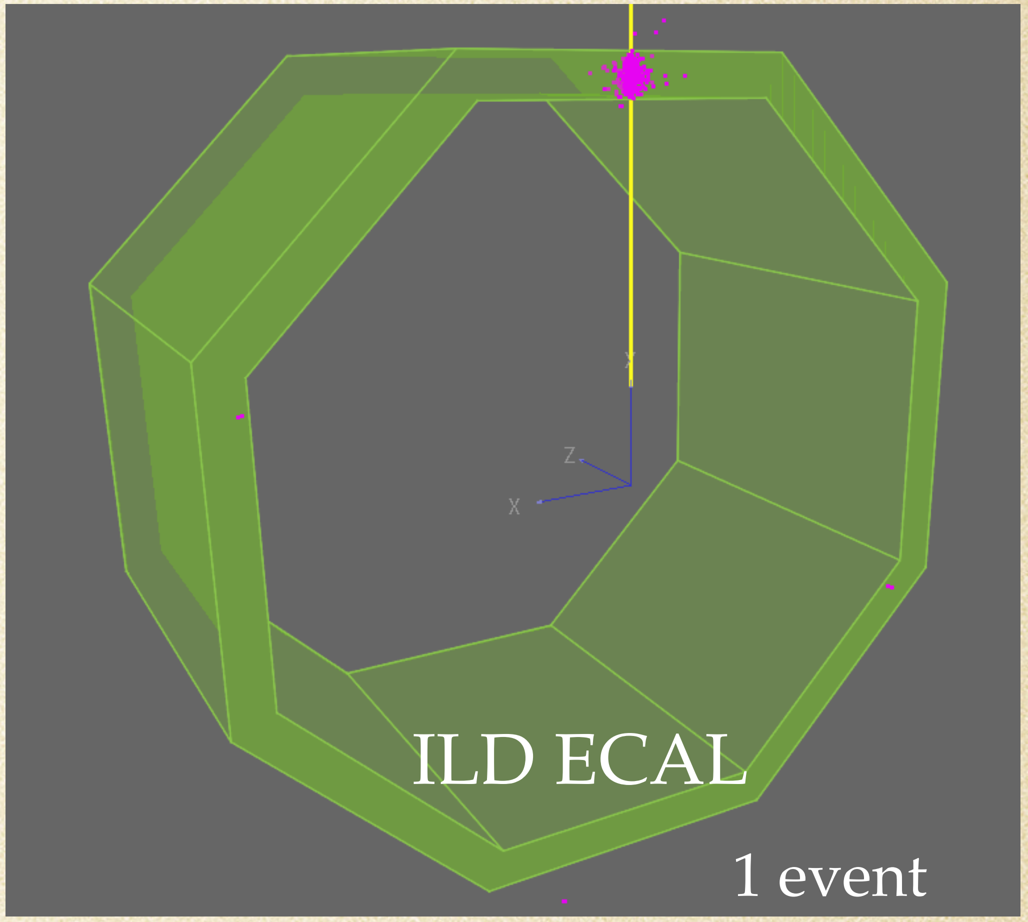


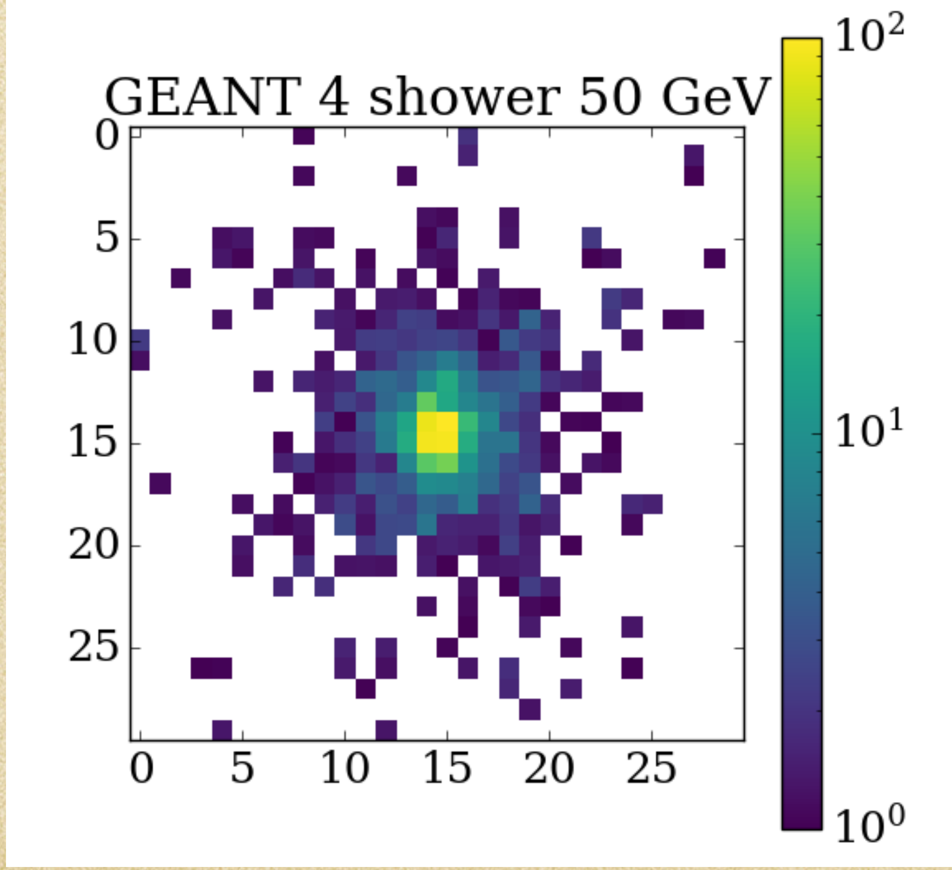
Goal : Reproduce shower simulations with a faster, powerful **generator**; based on state-of-the-art machine learning techniques

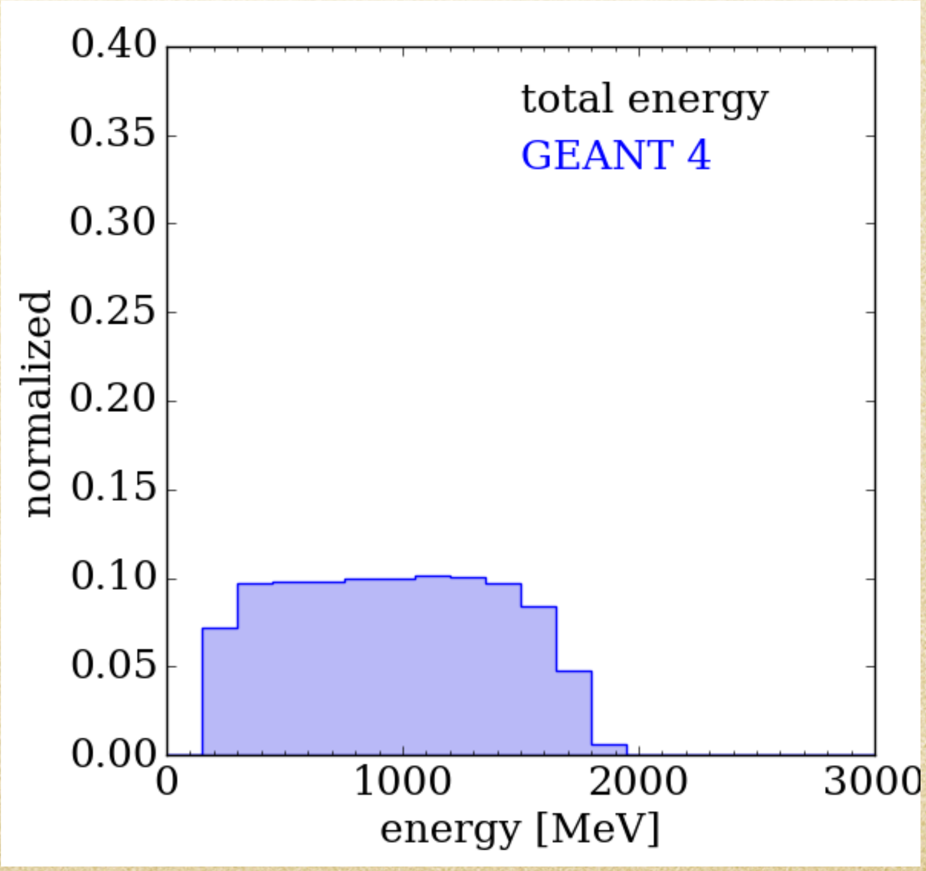
Enormous amounts of CPU time could be potentially saved!

Data

- Simulation is done by Geant4
- Shooting photon perp. to ECAL
  - 85k photon showers
  - Photon energy (10 - 100 GeV)
  - 30x30x30 pixels
  - For now: Sum along the beam direction
    - ➡ 30 x 30 images



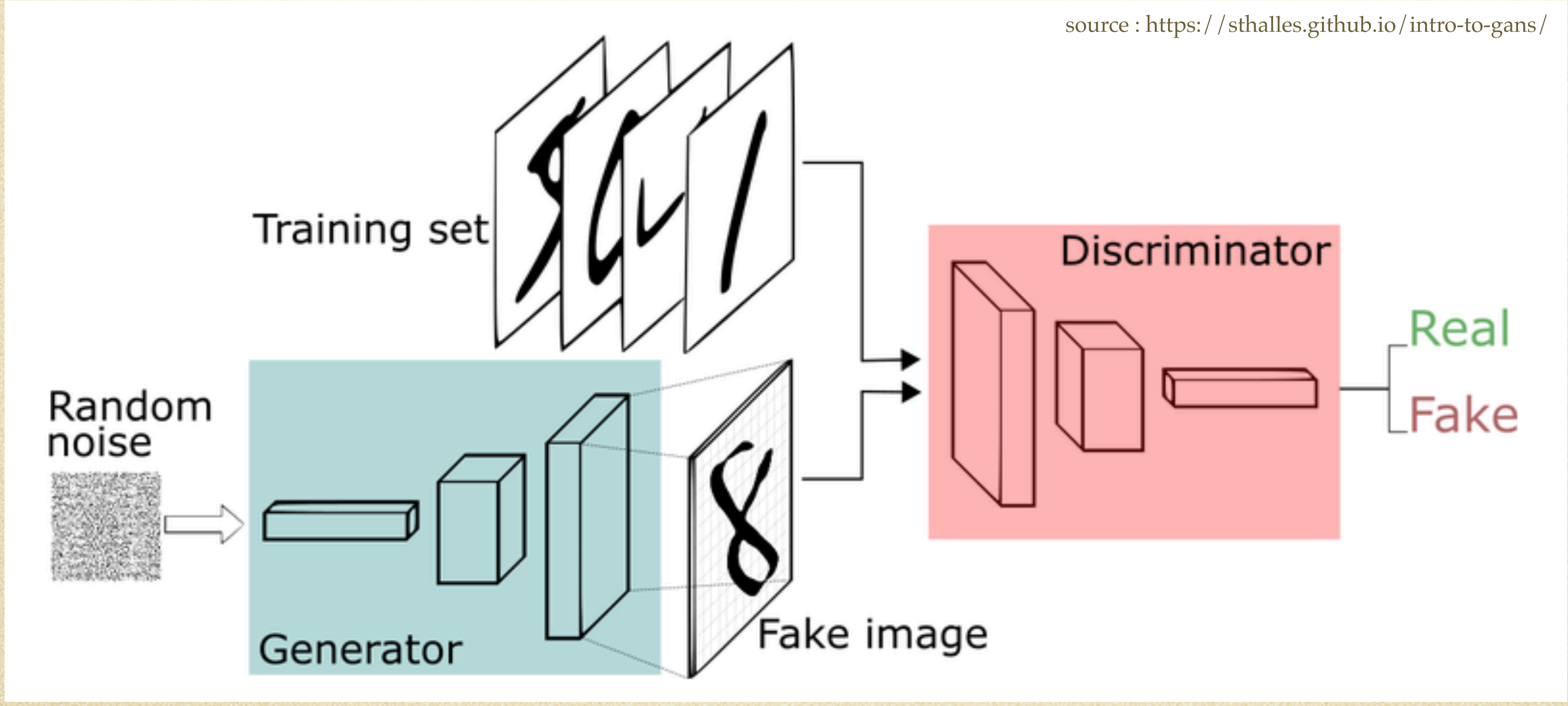




Generative Adversarial Networks (GAN)

Consists of two networks playing min-max game :

- Generator learns to **fool** the discriminator
- Discriminator learns to **distinguish** fake or real images
- Continuous feedback between them. Both tries to get better



$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}\Big(\log D(x) \Big) + \mathbb{E}_{z \sim z_{data}(z)}\Big(\log(1 - D(G(z))) \Big)$$

Generative Adversarial Networks (GAN)

Consists of two networks playing min-max game :

arXiv:1701.07875v3

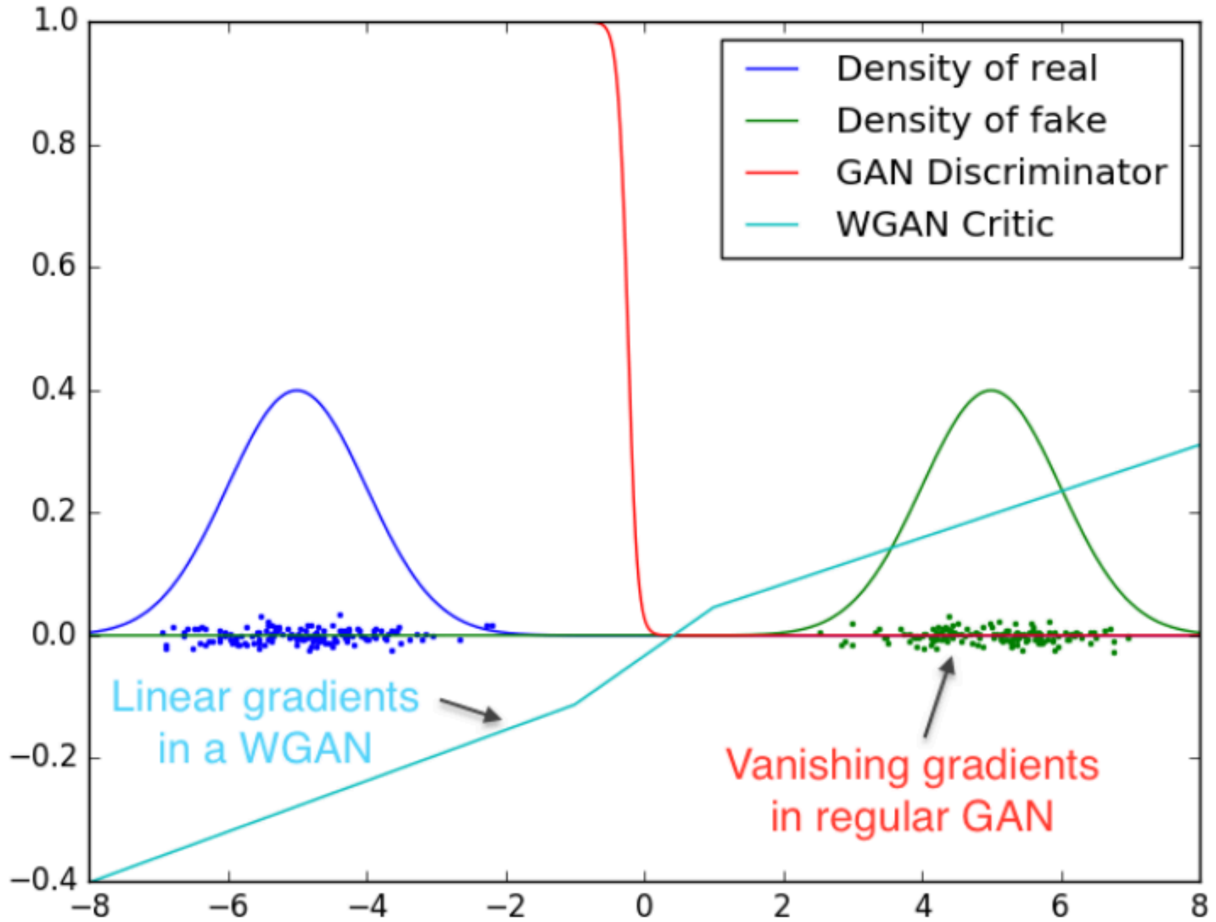


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the discriminator of a minimax GAN saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\Big(\log D(x)\Big) + \mathbb{E}_{z\sim z_{data}(z)}\Big(\log(1-D(G(z)))\Big)$$

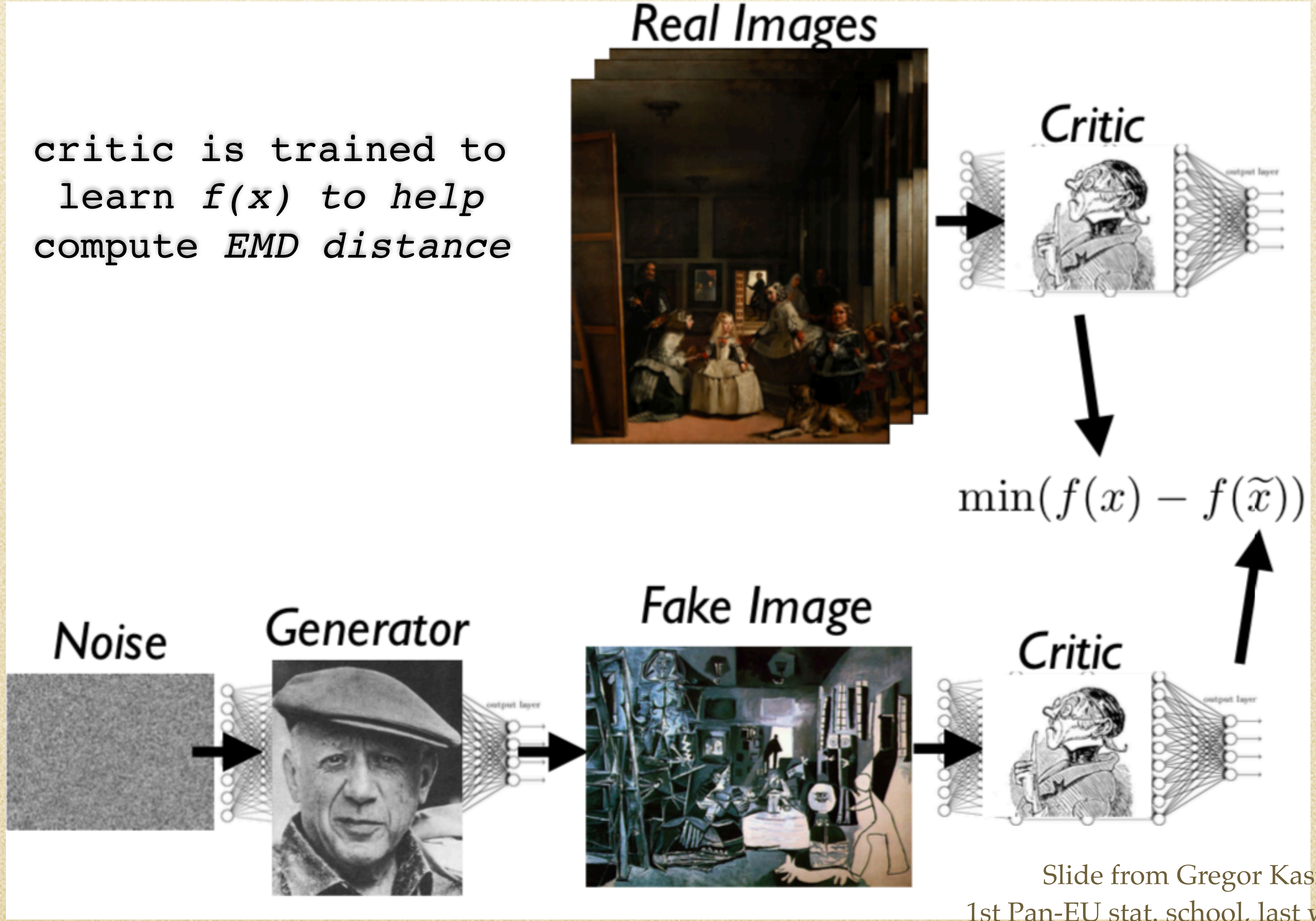
Wasserstein GAN (WGAN)

a GAN trying to minimise EMD between the real and generated distributions:

$$\mathrm{EMD}(P,Q) = \sup_{||f||_L \leq 1} \left[\mathbb{E}_{x\sim P} f(x) - \mathbb{E}_{x\sim Q} f(x)\right]$$

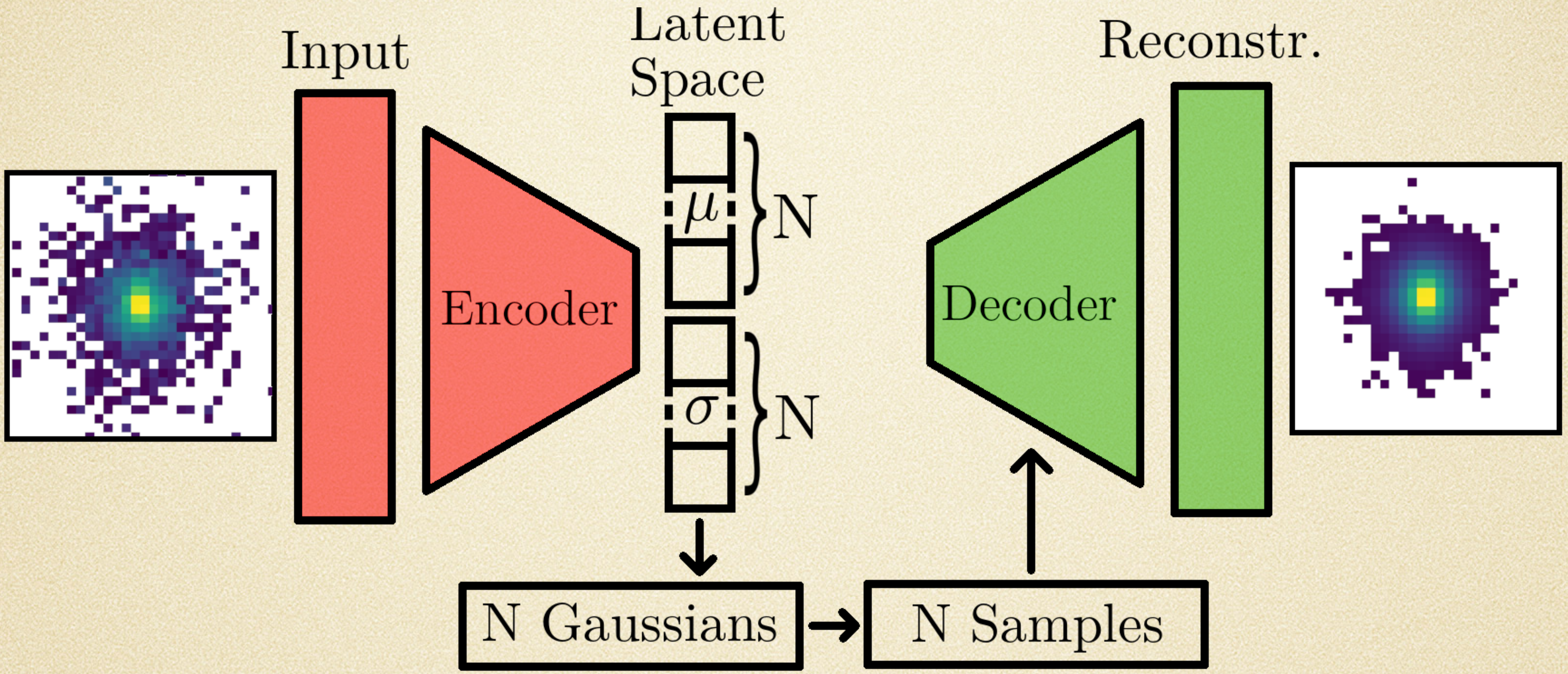
Kantorovich-Rubinstein duality!

<u>during the training:</u>



Slide from Gregor Kasieczka
1st Pan-EU stat. school, last week @DESY

Variational AutoEncoders (VAE)

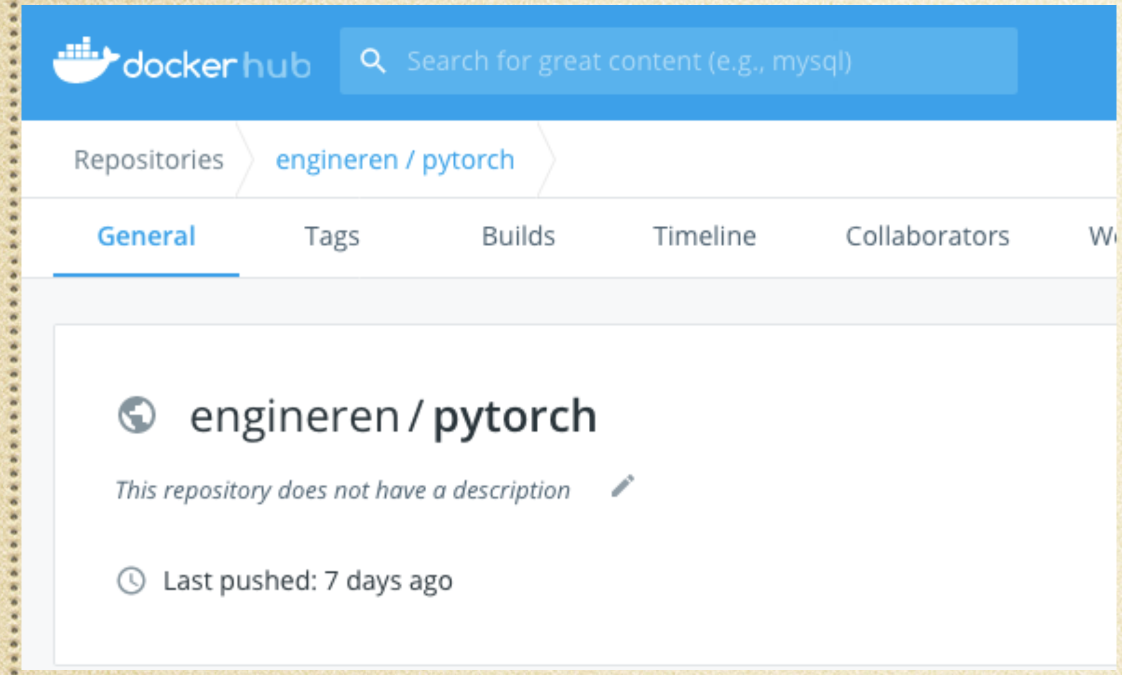


- It learns features by compression and reconstruction
- Introduces latent space consisting of Gaussian distributions
- Regularised latent space allows **image generation**

$$L = L_{MSE} + L_{Pool_{10\times10,s=5}} + K_{KLD} + L_{MMD_{radial}} \\ + L_{\Delta(E_{incoming},E_{predict})} + L_{\Delta(E_{sum},E_{sum-recon})}$$

Training on maxwell cluster with Docker + Singularity

Purpose : To package up an application with all of the parts it needs, such as **libraries and other dependencies**, and ship it all out as **one** package



10 lines (6 sloc) | 208 Bytes

```
FROM engineren/pytorch:test

RUN pip install --upgrade pip

COPY requirements.txt requirements.txt
RUN pip install --upgrade --no-cache-dir
        rm requirements.txt



WORKDIR /home
```

11 lines (11 sloc) | 104 Bytes

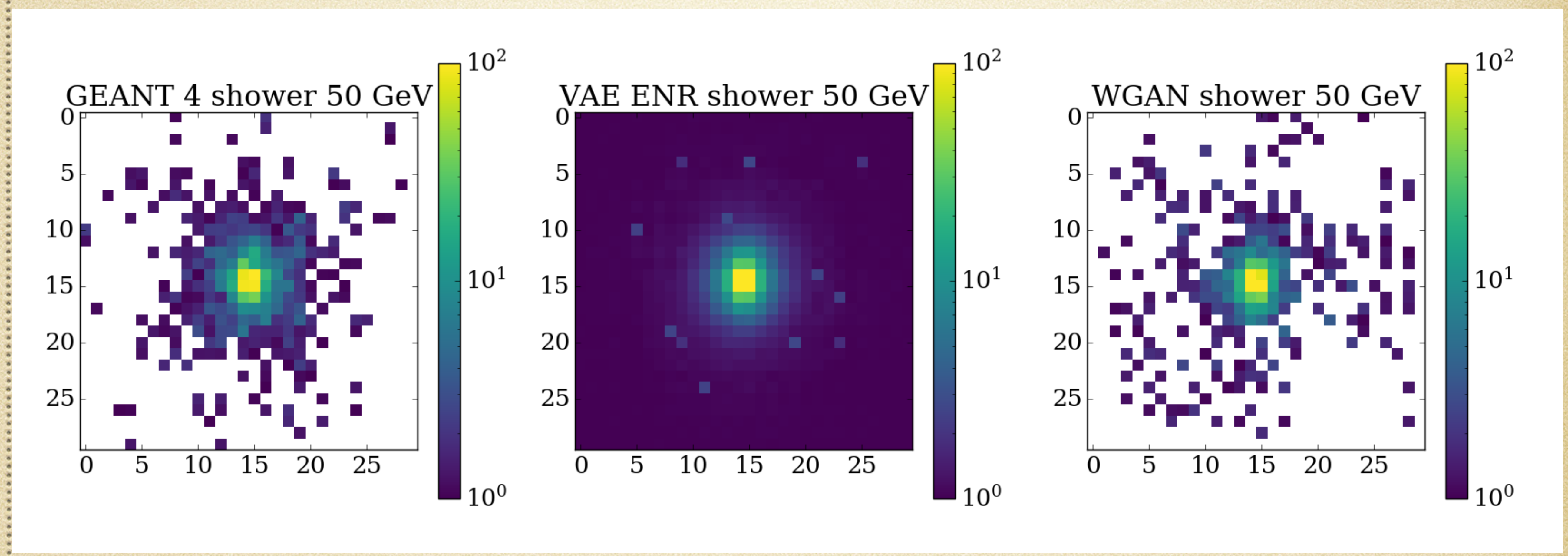
```
numpy
scikit-learn
uproot
matplotlib
pandas
scipy
h5py
jupyter
torchsummary
tensorflow
tensorboardX
```

create an instance
singularity instance start **--nv** **docker://engineren/pytorch:custom** train-GAN

start the container
singularity **run** instance://train-GAN **python train.py**

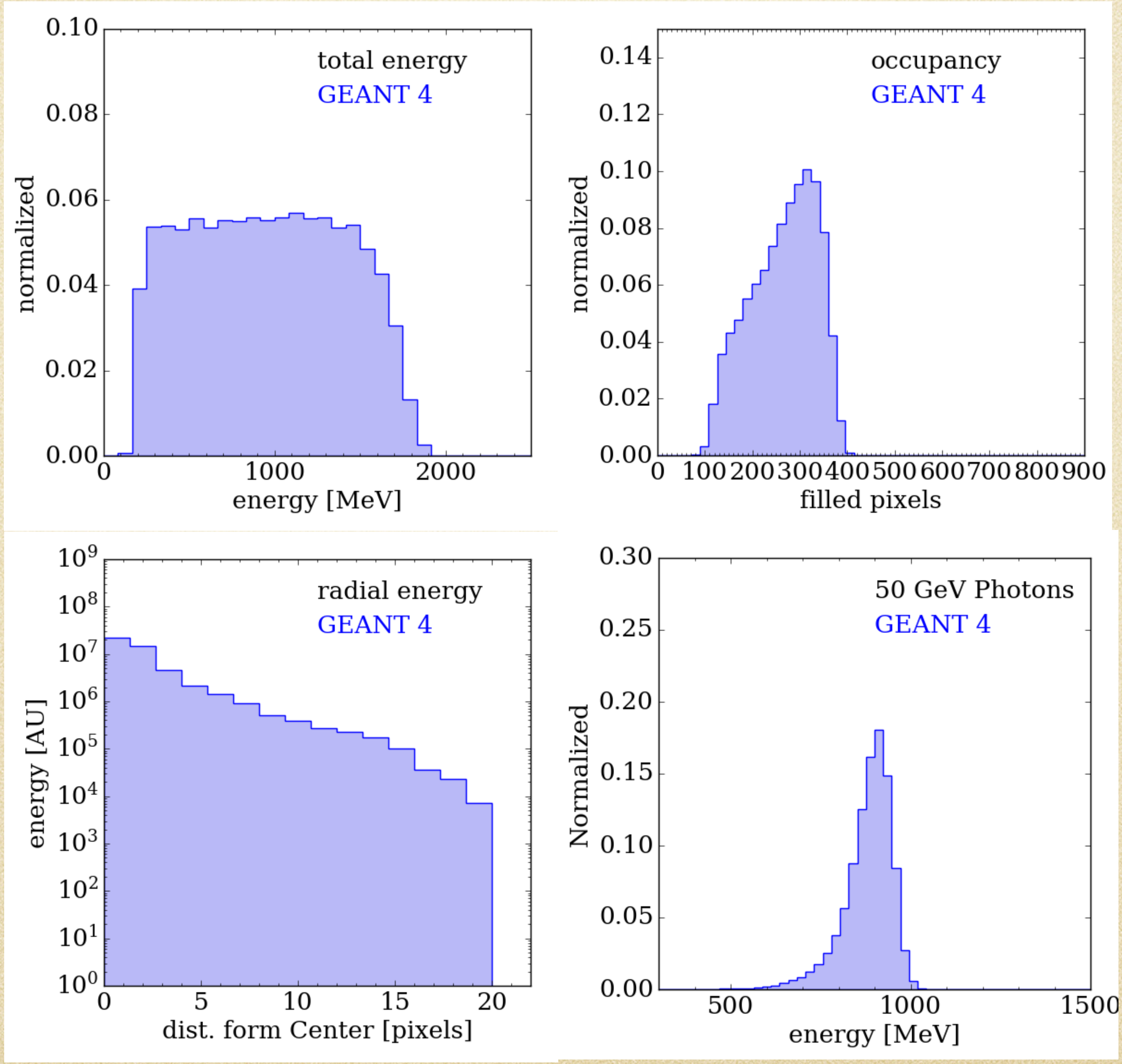
Results

Generated showers : **by-eye** comparison!



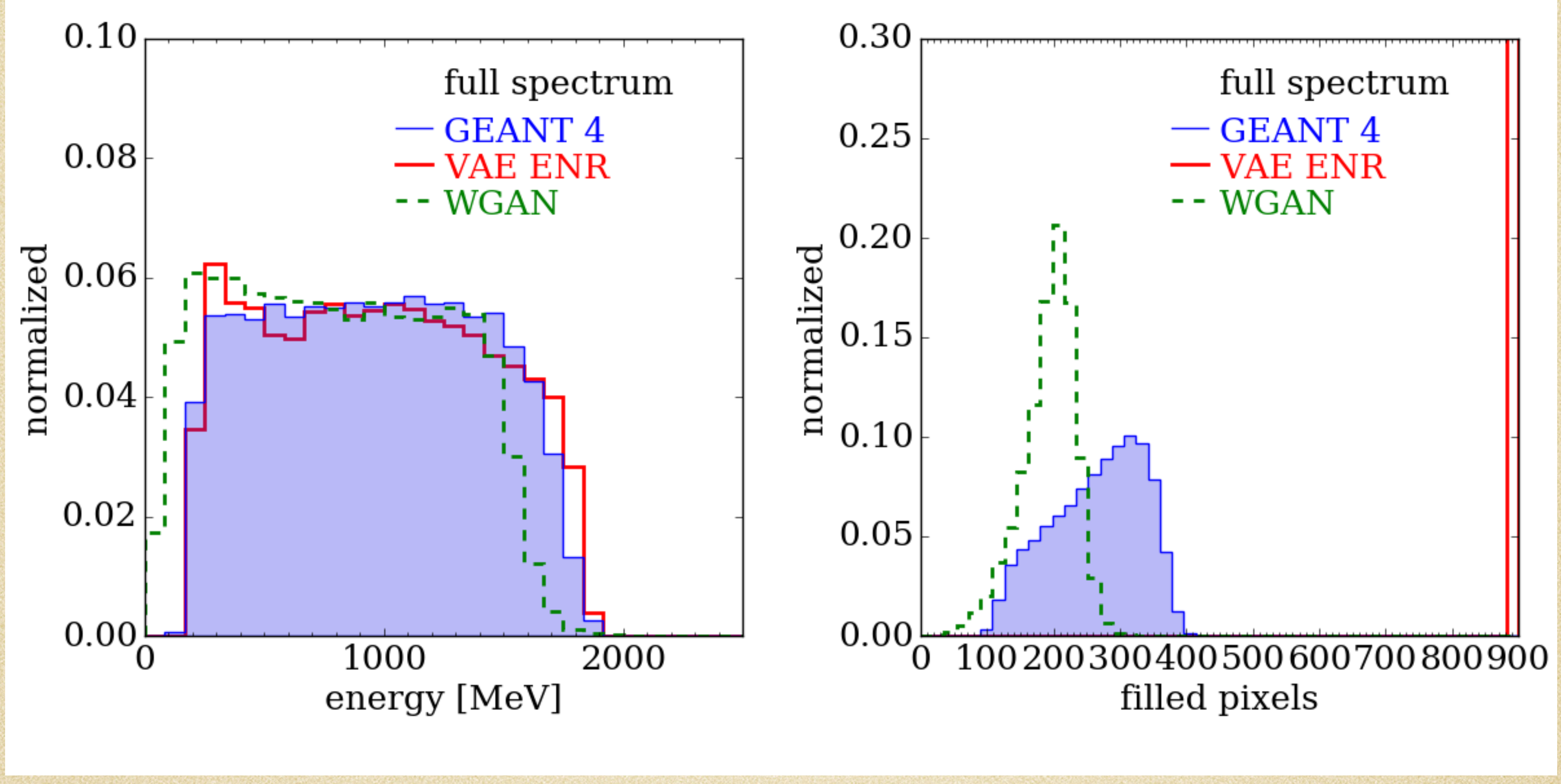
Quality Assurance

Need to ensure that our showers are **as good as** those from Geant4



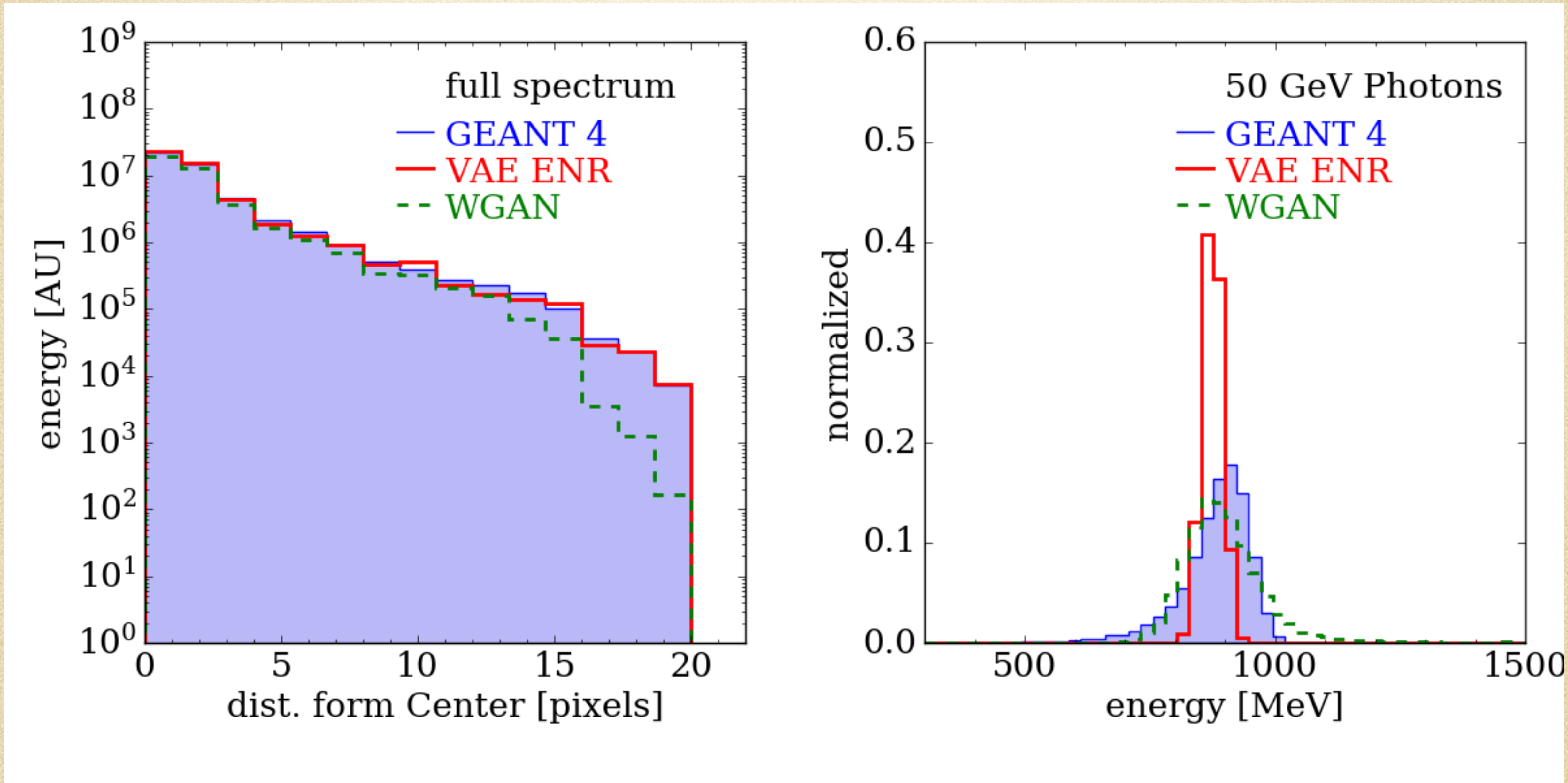
Quality Assurance

Need to ensure that our showers are **<u>as good as</u>** those from Geant4



Quality Assurance

Need to ensure that our showers are **as good as** those from Geant4



Conclusion and Outlook

Application of generative models to EM shower simulation in progress!

Outlook :

- Continue to explore shower shape variables.
- Go for 3D showers (i.e images 30x30x30)
- Adversarial VAE
- WGAN-GP and energy regressor
- Add standard (vanilla) GAN

Backup

CaloGAN : Energy distribution of generated showers