Attempts to generate fast electromagnetic shower simulation with Generative Adversarial Networks (GANs)

Engin Eren

05.08.2019

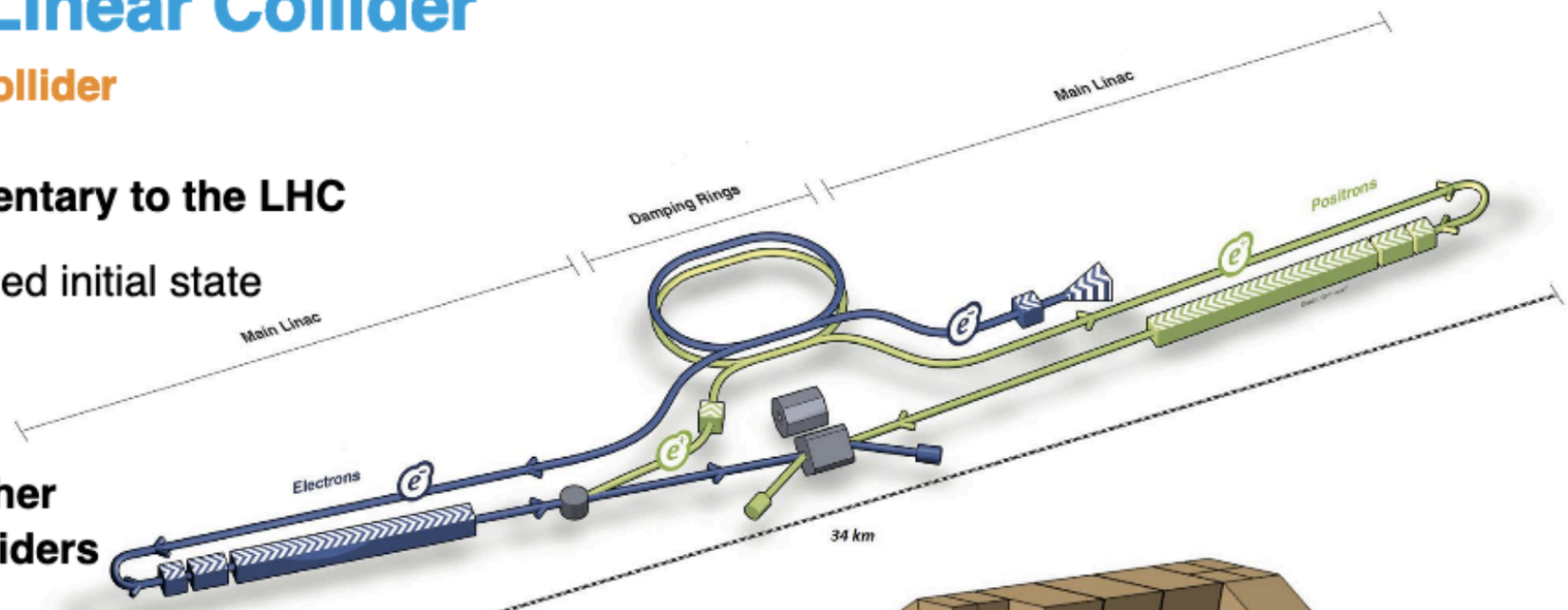
Introduction

The International Linear Collider

A planned electron-positron collider

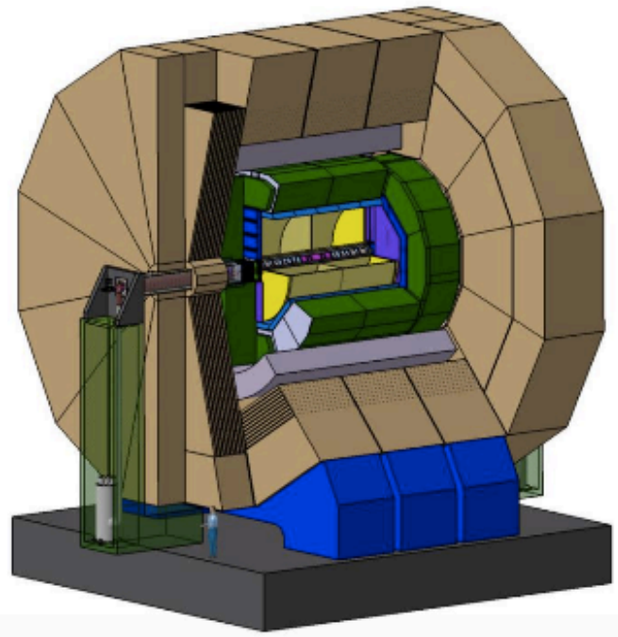
Lepton colliders are complementary to the LHC

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



Advantages of the ILC over other planned electron-positron colliders

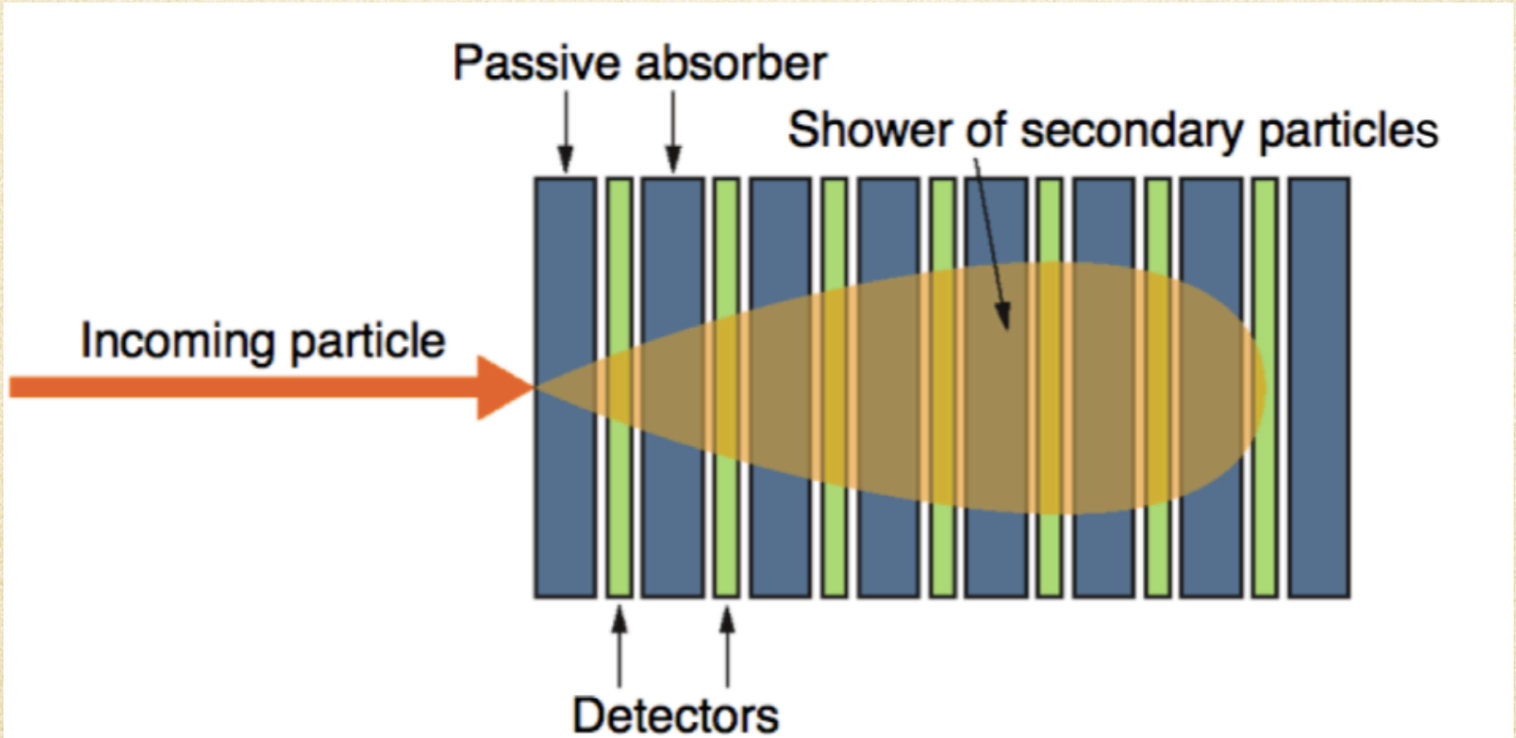
- mature technology
- centre-of-mass energy can be tuned and increased:
  250 GeV in initial stage, upgrades to 500 GeV and 1 TeV
- polarisation of both beams: $P(e^-)=\pm 80\%$, $P(e^+)=\mp 30\%$
- triggerless operation
- hermeticity of detector down to lowest angles



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**

➡ A calorimeter consists of alternating layers of passive absorbers and active detectors

➡ ILD prefers to use this type of EM calorimeter

GANs for EM Shower Simulation

Promising results by CaloGAN in 2018

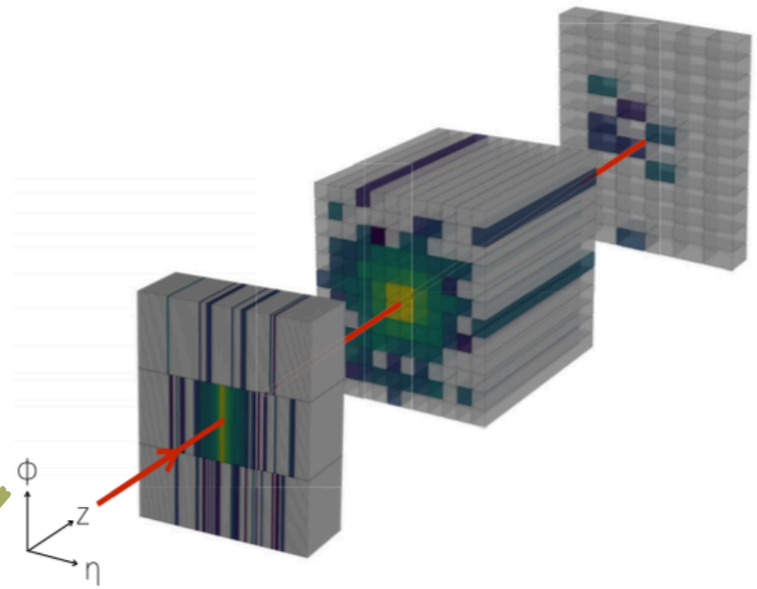
- why not try for ILD ?
  - CaloGAN has a simple setup : Only 3 layers with LAr (+ lead absorbers) ✓
  - ILD has 30 layers of Silicon - Tungsten ECAL ☠

CALOGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks

Michela Paganini,[1,2,*] Luke de Oliveira,[2,†] and Benjamin Nachman[2,‡]
[1]*Yale University, New Haven, Connecticut 06520, USA*
[2]*Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

(Received 18 July 2017; published 30 January 2018)

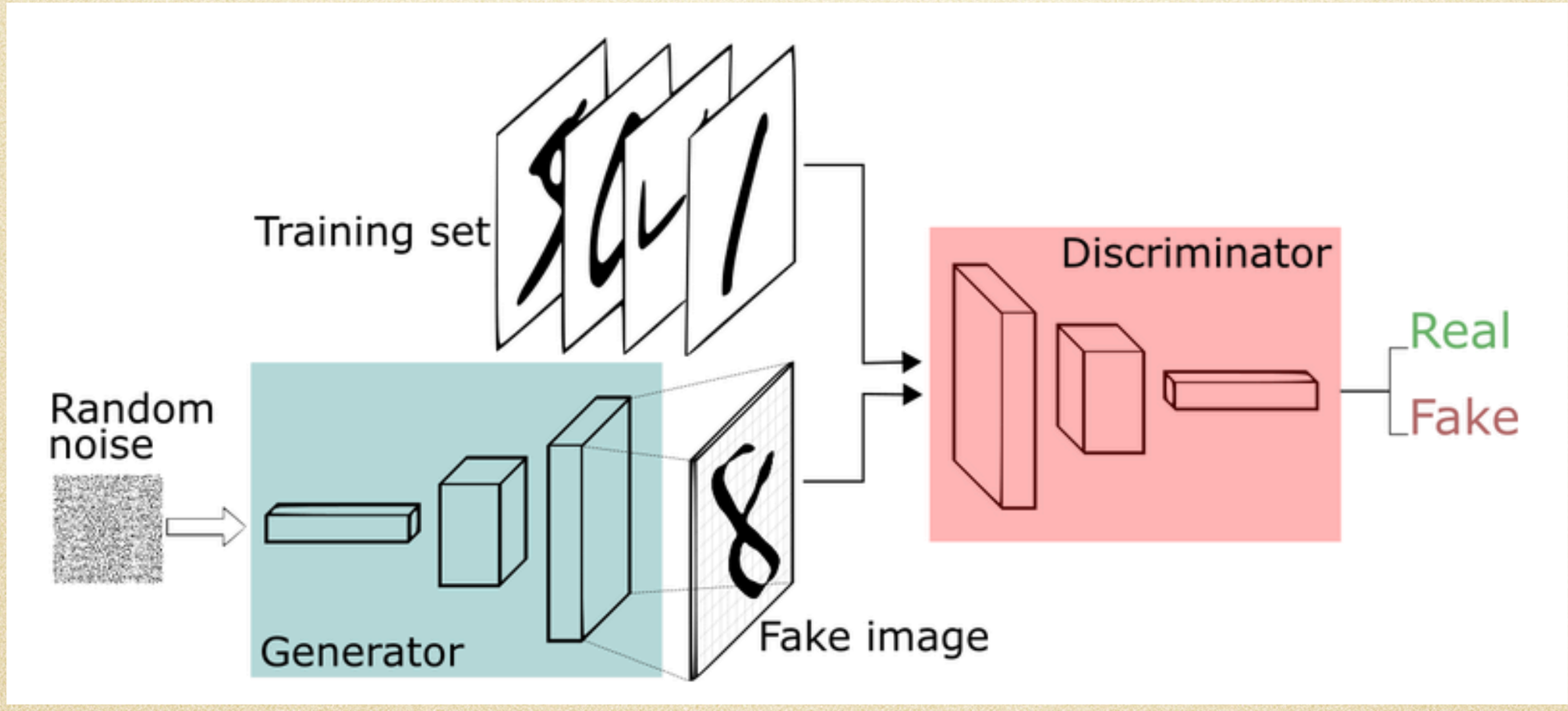
Simulator	Hardware	Batch size	ms/shower
GEANT4	CPU	N/A	1772 ☠
CALOGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012 ✓



Generative Adversarial Networks (GAN)

Consists of two networks playing min-max game :

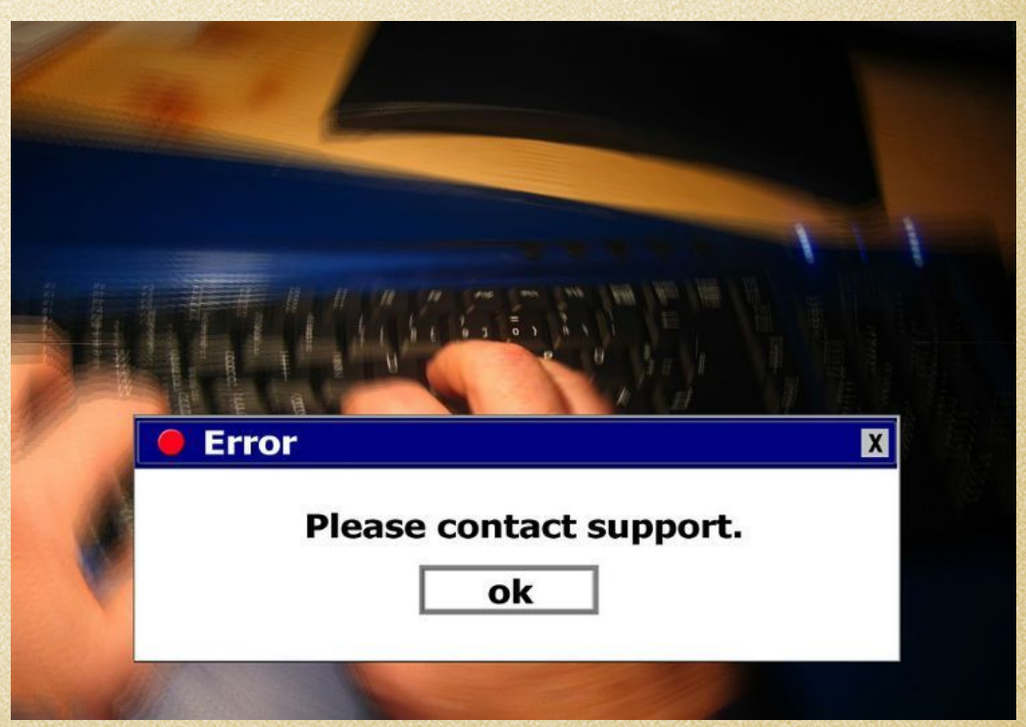
- Generator learns to **fool** the discriminator
- Discriminator learns to **distinguish** fake or real images
- Continuous feedback between them. Both tries to get better



source : https://sthalles.github.io/intro-to-gans/

Steps towards CaloGAN usage

1. Reproduce the original result
   - Manage to run the code, which is 2 years old
   - Ask authors for clarification (if they answer)
   - Dockerize the code and adapt it for CaloLayers > 3
2. Prepare your computing environment
   - Ask help from IT & learn how to run the code in GPUs (read confluence)
   - Explore maxwell HPC cluster
3. Prepare the training data
   - Convert *slcio* files to *hdf5* files



CaloGAN-docker

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

21 lines (15 sloc) | 422 Bytes

```
1  FROM tensorflow/tensorflow:latest-gpu
2
3  #Install dependencies
4
5  RUN pip install --upgrade pip
6  RUN pip install pandas
7  RUN pip install matplotlib
8  RUN pip install numpy scipy scikit-learn
9  RUN pip install pyyaml
10 RUN pip install keras
11 RUN pip install uproot
12
13 RUN apt-get update
14 RUN apt-get upgrade -y
15 RUN apt-get install -y git
16 RUN apt-get install -y vim
17
18 WORKDIR /home
19 RUN git clone https://github.com/EnginEren/CaloGAN.git
```

EnginEren / CaloGAN-docker

Watch 0 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights | Settings

This is an attempt to run CaloGAN in docker env. Edit

Manage topics

17 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

EnginEren uproot added — Latest commit 4416a6f 22 hours ago

Dockerfile	uproot added	22 hours ago
README.md	"--nv" option was added	17 days ago

README.md

CaloGAN-docker

Build and push image :

```
$ docker build -t engineren/calogan-docker .
```

```
$ docker image push engineren/calogan-docker
```

Running in my local :

```
$ docker run -it --rm -v $PWD/CaloGAN/data:/home/CaloGAN/data calogan-docker python -m models.train models/particles.yaml
```

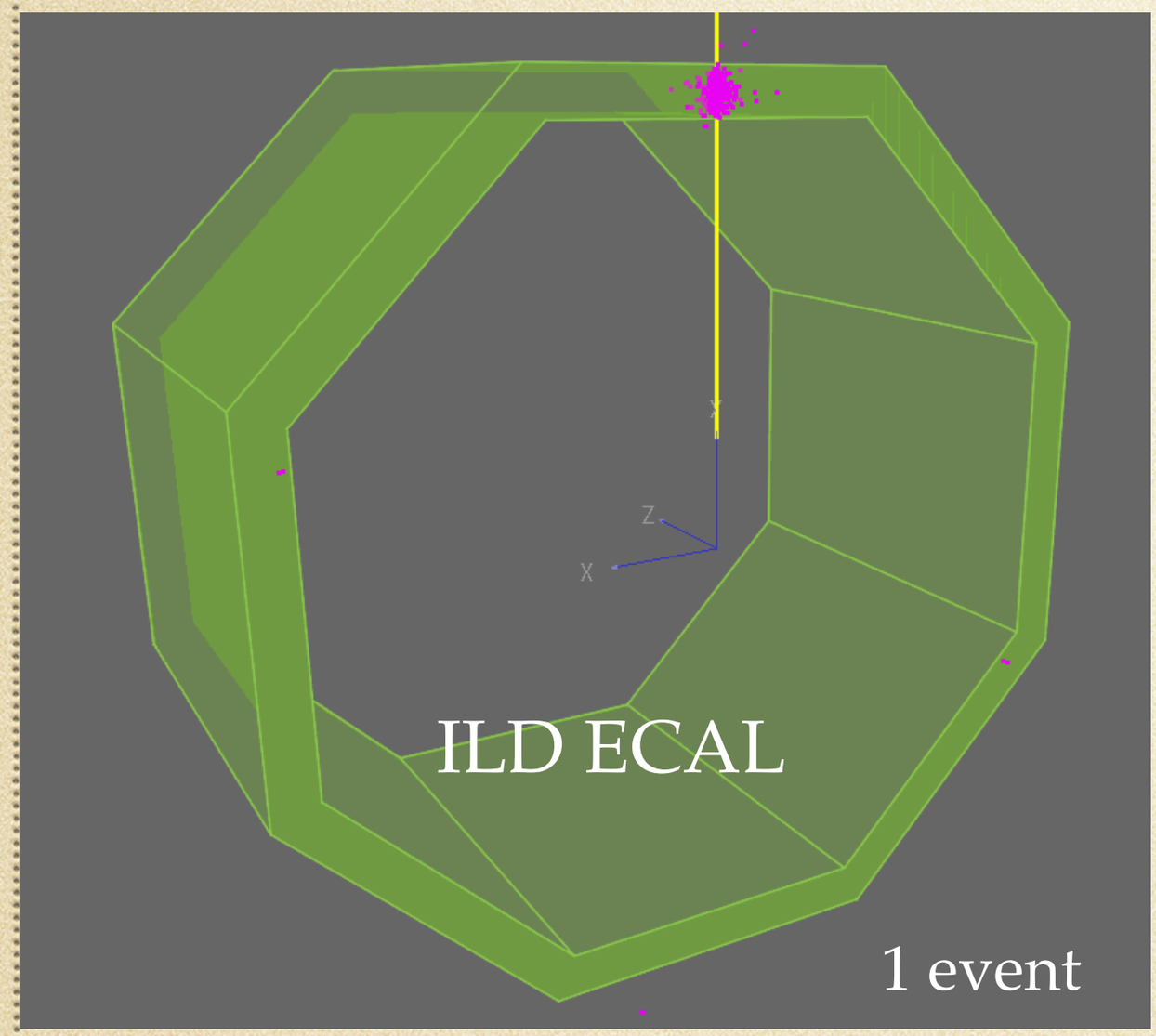
Running naf-ilc-gpu :

```
$ singularity pull docker://engineren/calogan-docker:latest
```

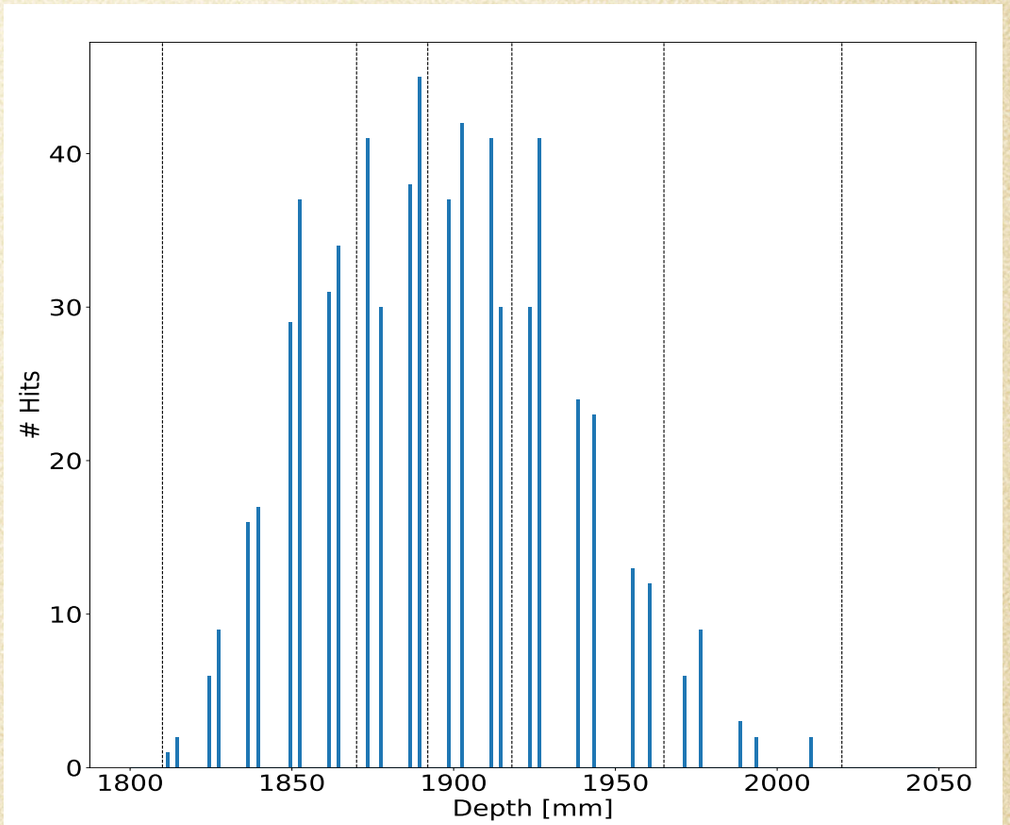
```
$ singularity instance start --bind data:/home/CaloGAN/data --nv calogan-docker_latest.sif caloGAN
```

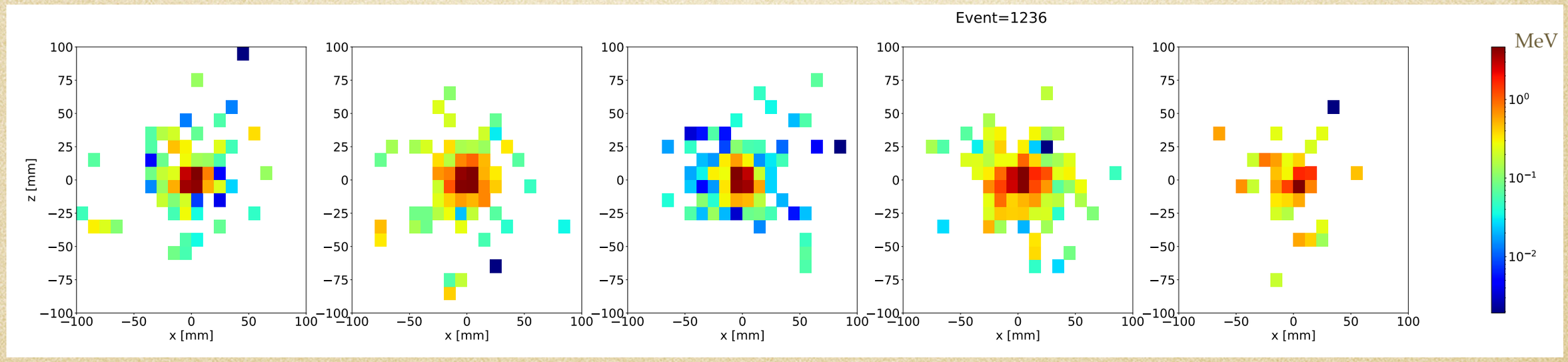
```
$ singularity run instance://caloGAN python -m models.train models/particles.yaml
```

Particle Gun via Geant4



- ✓ Shoot photon perpendicular to the ECAL
- ✓ Project depth information into the 5 layers
- ✓ LCIO to hdf5 conversion





Training

Training is being performed in GPU nodes in BIRD as well as in Maxwell cluster

```
[eren@max-display001 ~]$ cat train-max-cmsg003-2862127.out
2019-08-01 20:18:44,200 - models.train[INFO]: 1 particle types found.
2019-08-01 20:18:44,201 - models.train[INFO]: loading data from 1 files
2019-08-01 20:18:45,562 - models.train[INFO]: Building discriminator
2019-08-01 20:19:05,660 - models.train[INFO]: Building generator
2019-08-01 20:19:19,089 - models.train[INFO]: using attentional mechanism
2019-08-01 20:19:52,577 - models.train[INFO]: commencing training
2019-08-01 20:19:52,583 - models.train[INFO]: Epoch 1 of 201
2019-08-01 20:19:52,583 - models.train[INFO]: processed 1/119 batches
[eren@max-display001 ~]$
```

Training data :

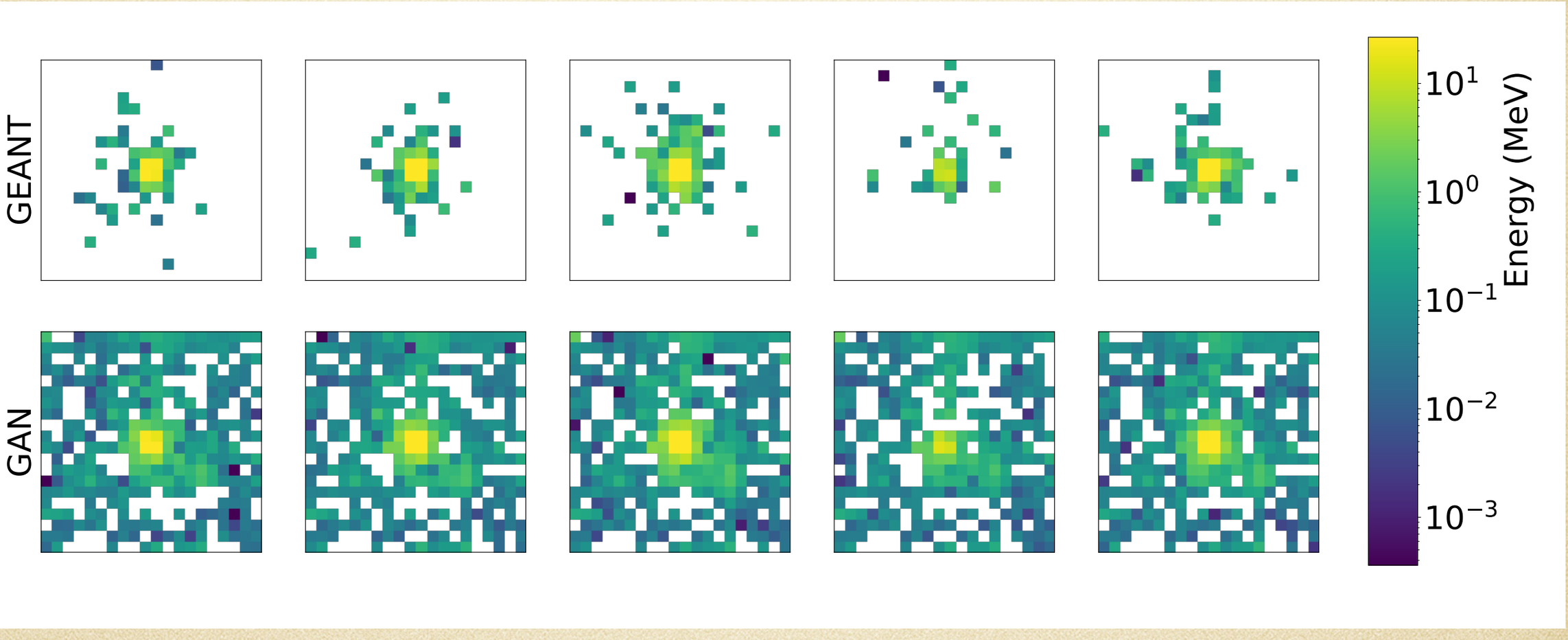
- 30k Geant4 shower events
- Each calorimeter layer is represented by a 20x20 size image.
- Incoming photon's energy is uniformly distributed between 10-50 GeV

0.1.3 Look at created hdf5 file

```
In [30]: f = h5py.File('/home/eren/naf-mount/gamma.hdf5', 'r')
         list(f.keys())

Out[30]: ['energy', 'layer_0', 'layer_1', 'layer_2', 'layer_3', 'layer_4']

In [33]: b = f['layer_2']
         b.shape

Out[33]: (30711, 20, 20)
```

Very first results..

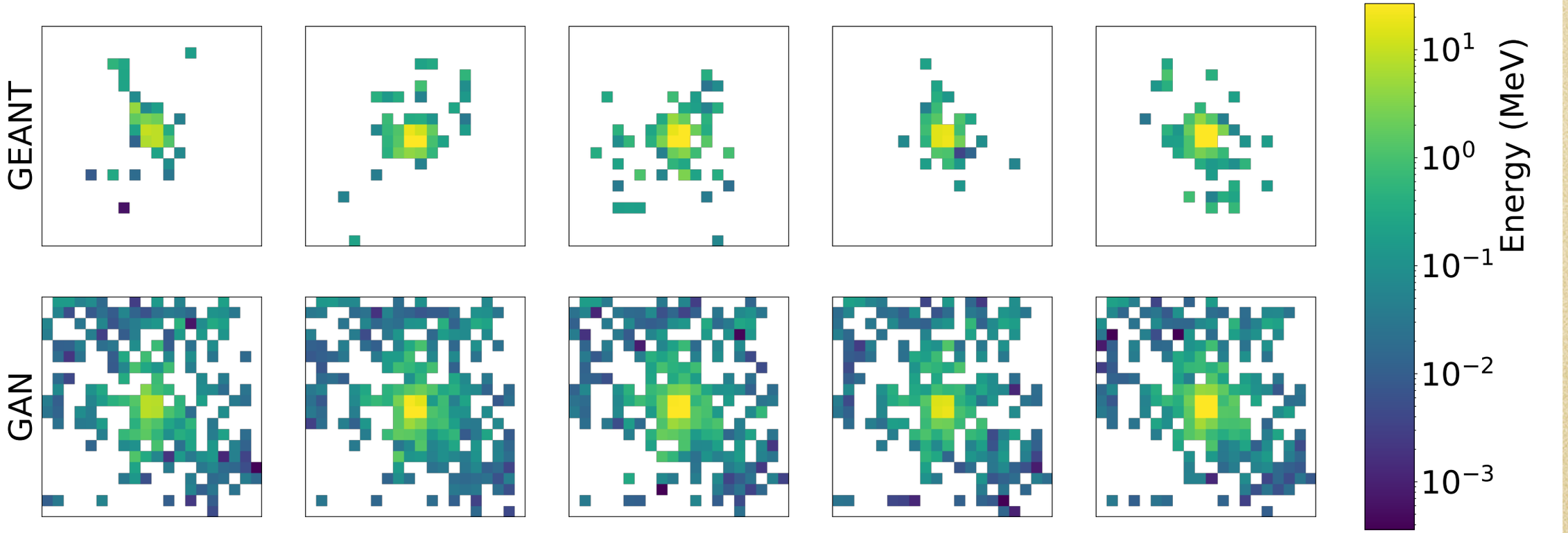
Let's pick 5 random images out of the training set and find their nearest neighbor in the synthesised (i.e generated) datasets.



Epoch 10, Layer 1

Very first results..

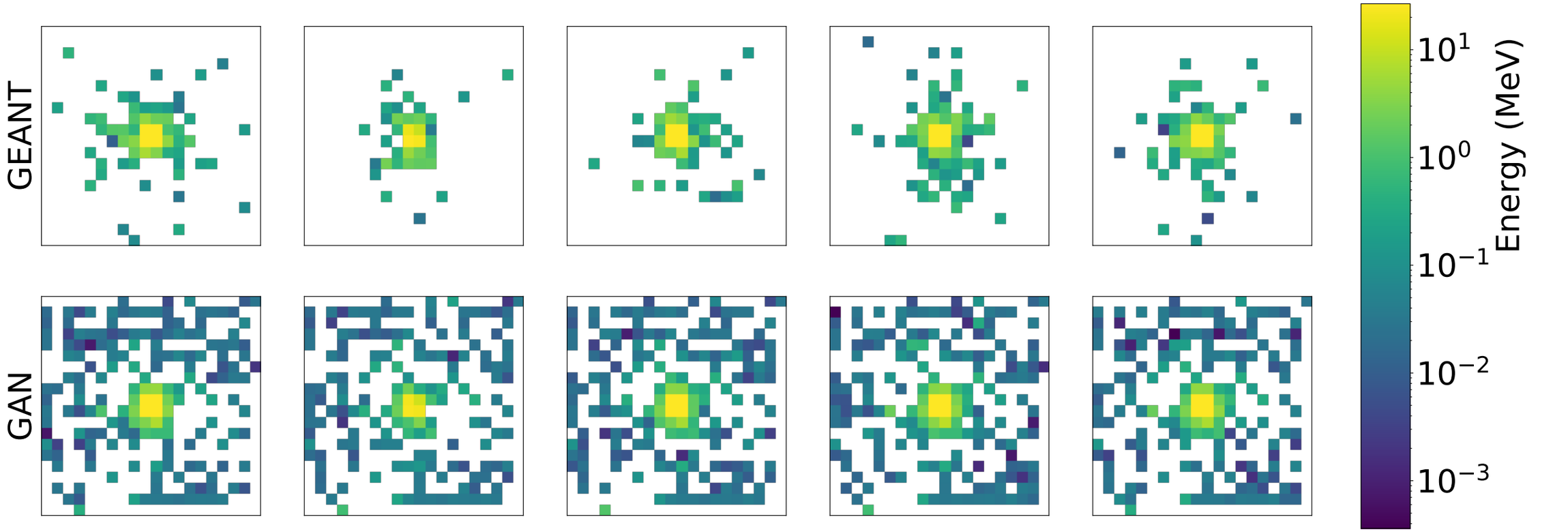
Let's pick 5 random images out of the training set and find their nearest neighbor in the synthesised (i.e generated) datasets.



Epoch 50, Layer 1

Very first results..

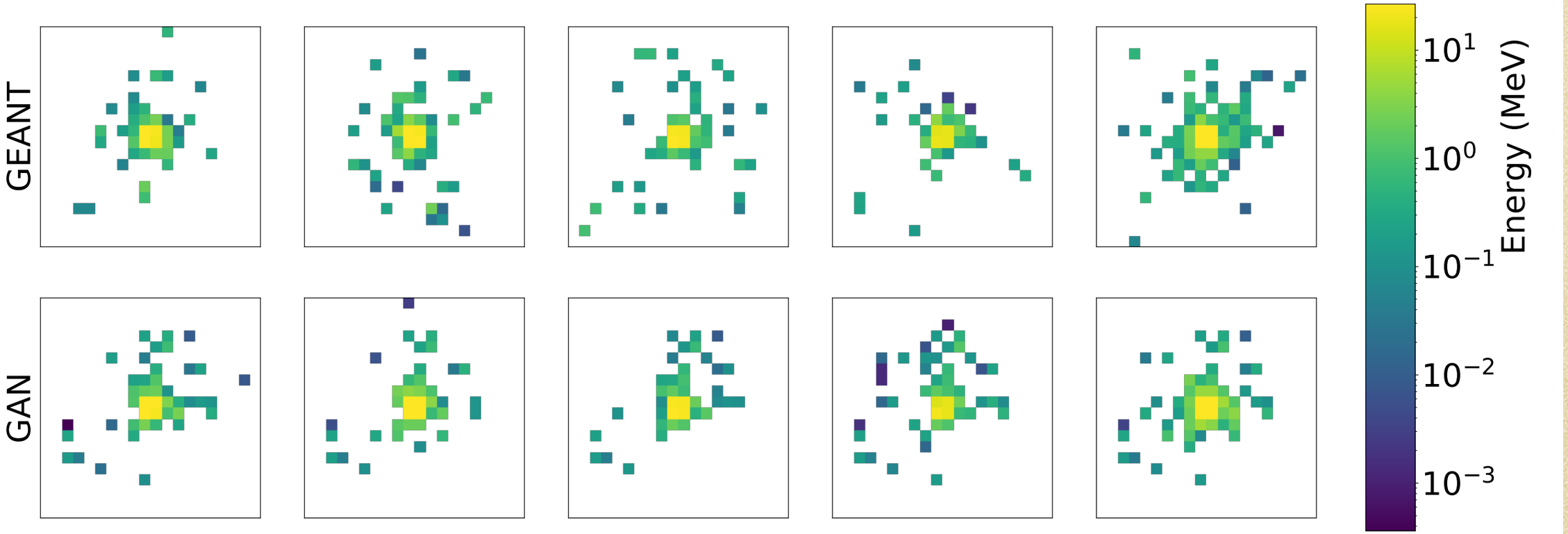
Let's pick 5 random images out of the training set and find their nearest neighbor in the synthesised (i.e generated) datasets.



Epoch 100, Layer 1

Very first results..

Let's pick 5 random images out of the training set and find their nearest neighbor in the synthesised (i.e generated) datasets.



Epoch 150, Layer 1

Summary

Application of GANs to the fast EM shower simulation in progress!

Outlook :

- Explore shower shape variables.
- Go for 30 layer ILD simulation, which is not easy due to input size!
- Explore other GANs (i.e Wasserstein GAN that CMS HGCAL is using)
- Collaborate with other people who uses ML/DL intensively