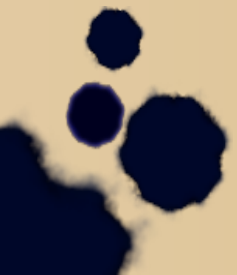
Machine Learning Algorithms for Fast Detector Simulation

Gulrukh Khattak
University of Engineering and Technology Peshawar
Pakistan

Supervisor: Sofia Vallecorsa
&
Andrei Ghaeta

Overview

- Personal Academic Background
- Machine Learning
- Current Status
- Future Plans

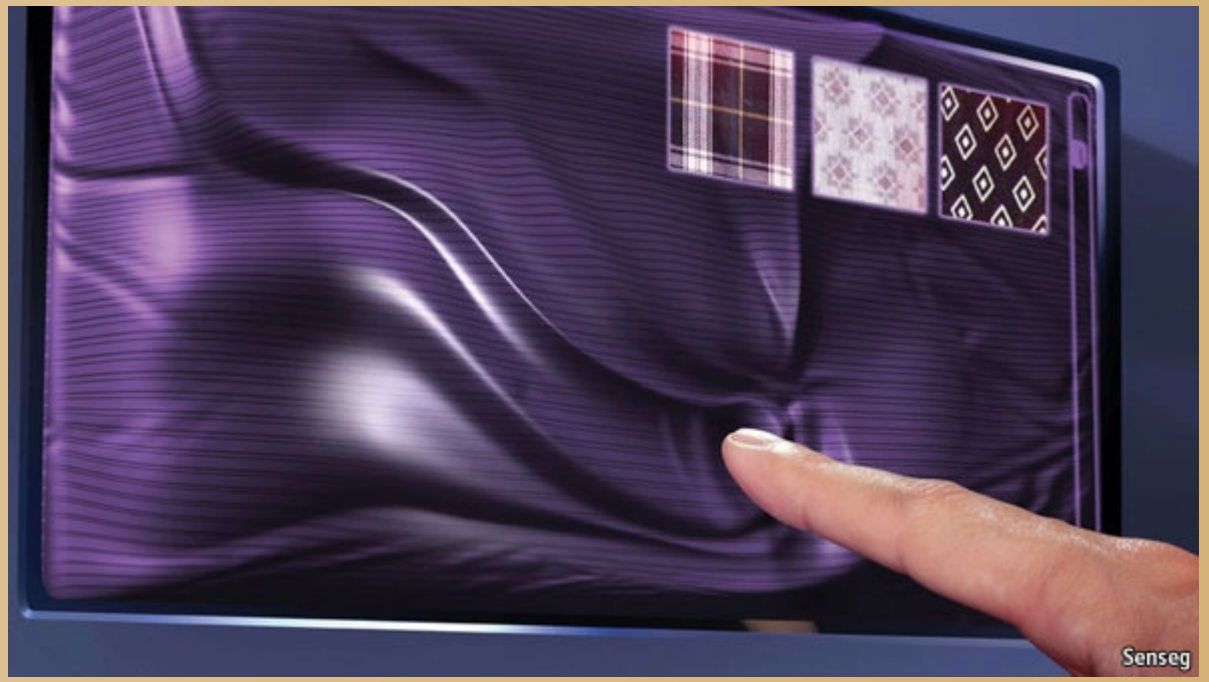


Academic

- Doctoral Student CERN(April 2017)
- MS. Microelectronics
- Bsc. Electronic Engineering

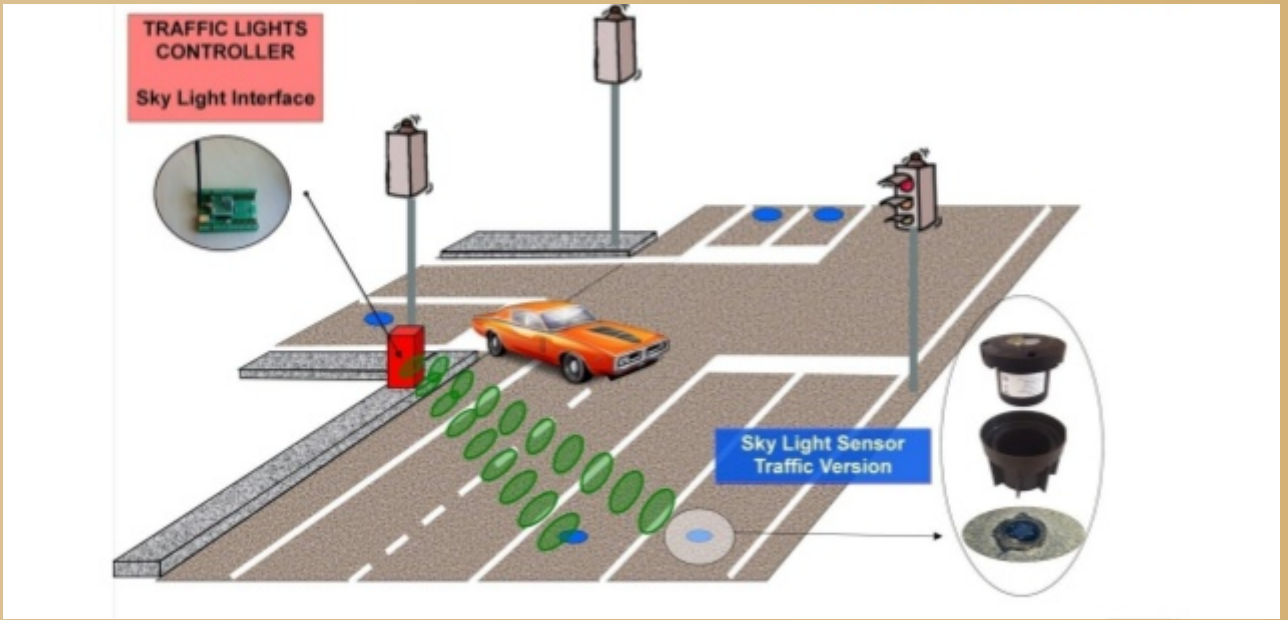
Past Research

- Haptic Texture Simulation (2010)
  - Research student Iwate University Japan
  - OpenGL, C++ Haptic Libraries.
    http://art-science.org/dl/nicograph/program2010a.html



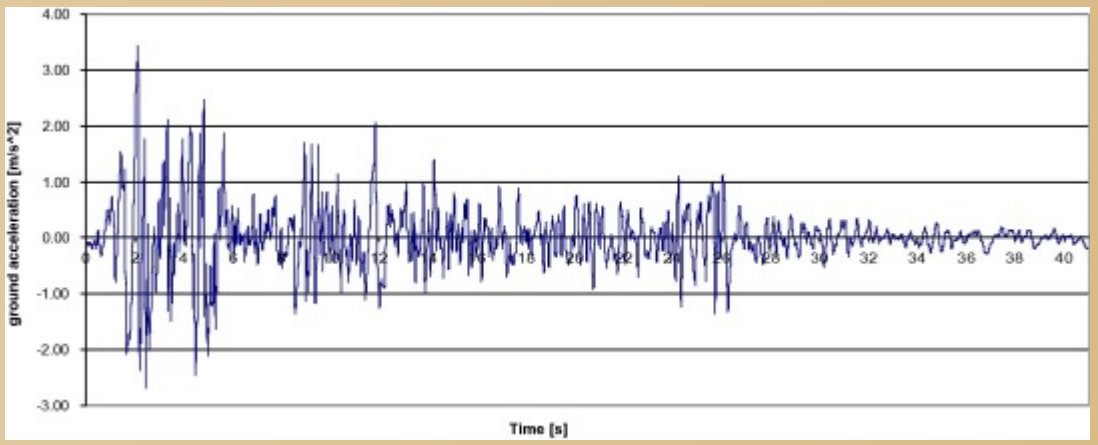
Past Research

- Adaptive Traffic Signal Controller(2013)
  - Research Associate UET Peshawar(Pakistan)
  - Smart Grid and Intelligent Traffic Systems
  - C, Verilog, VHDL, micro controller programming, ANN
  - http://ieeexplore.ieee.org/document/6799804/



Past Research

- Seismic Event Detection (2017)
  - UET Peshawar
  - C → Artificial Neural Networks (CGPANN)



http://digital-library.theiet.org/content/conferences/10.1049/cp.2017.0165

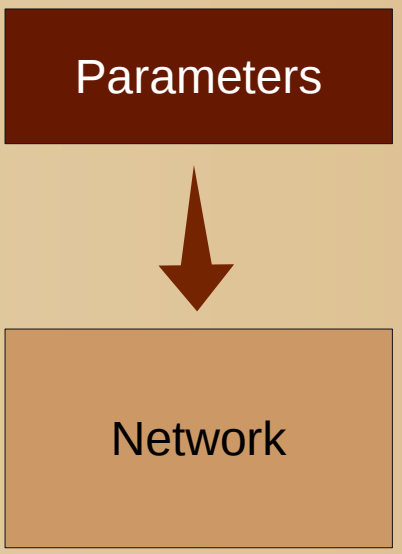
http://velastin.dynu.com/icprs17/USBICPRS2017/papers/ProceedingsICPRS2017.pdf

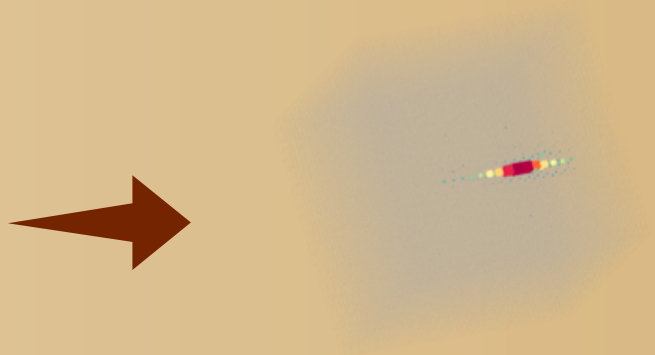
Machine Learning

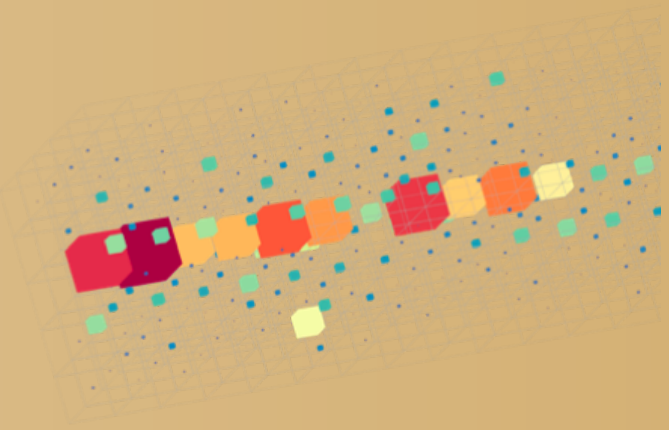
- Black Box Approach
  - Fast but Approximate
  - No Deterministic Solution exists
  - Time consuming and complex computations
  - Complex Probability Distributions
  - Variables are too many, hard to measure, missing values.
- Classification and Regression
  - Online Data Selection (Trigger)
  - Replace Complex computations (Simulation, Tracking)
  - Replace Expert Knowledge (Data Monitoring)
  - Into the Unknown …… (Anomaly Detection)
- Artificial Neural Networks
  - 100 billion neurons → Average Human Brain
  - Neuron 1943
  - Deep Learning 1990
  - GAN 2014

Problem Specification

- Calorimeter Simulation
  - Calorimeter → Three Dimensional Image
- Input vector mapped to 15,625 outputs
- The Images should be:
  - Parametric
  - Sampled from Probability Distribution
  - Physics Consistency



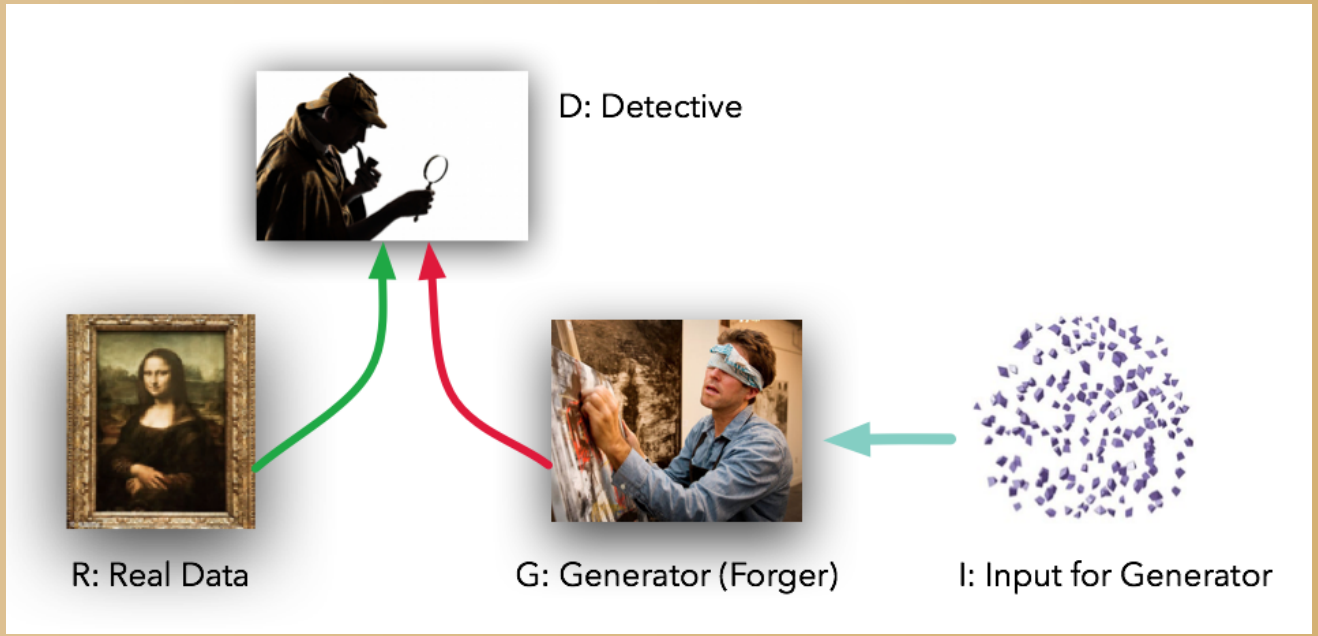




Generative Adversarial Networks

- Deep learning

- GAN



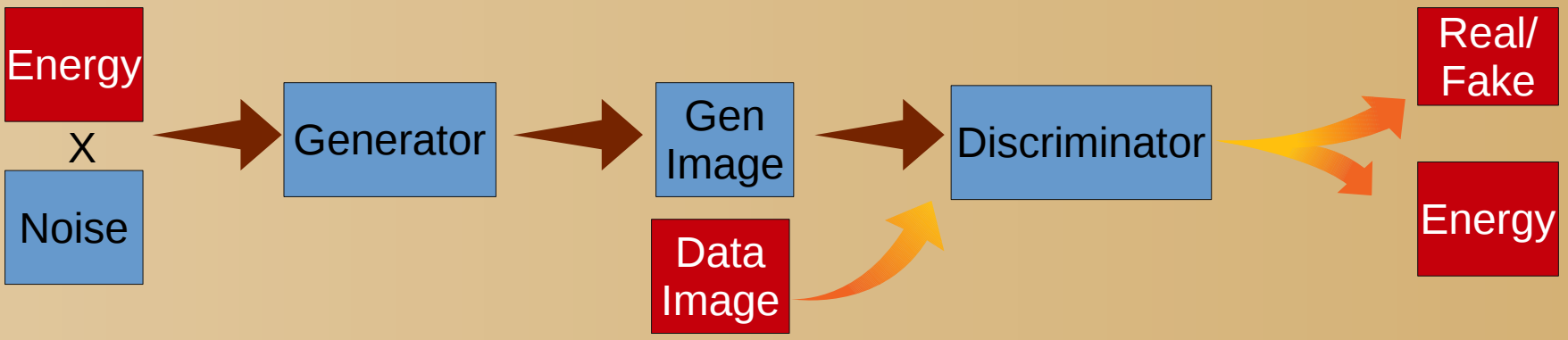
Current Status

- Misc. Tasks:
  - Tuning of GAN conditioned on particle type.
- GAN conditioned on Energy
- Validation and optimization
- Hyper parameter scan

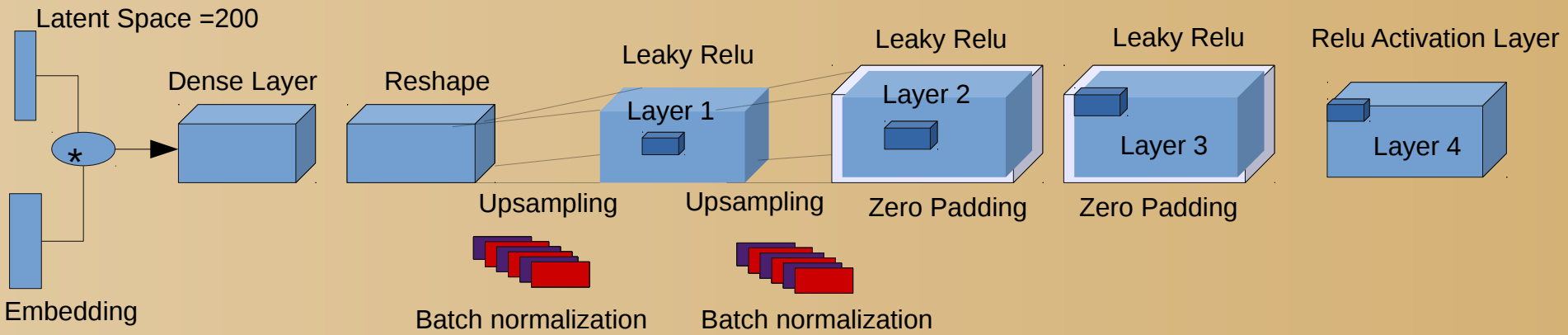
https://github.com/svalleco/3Dgan/tree/master/keras

Parametrised GAN

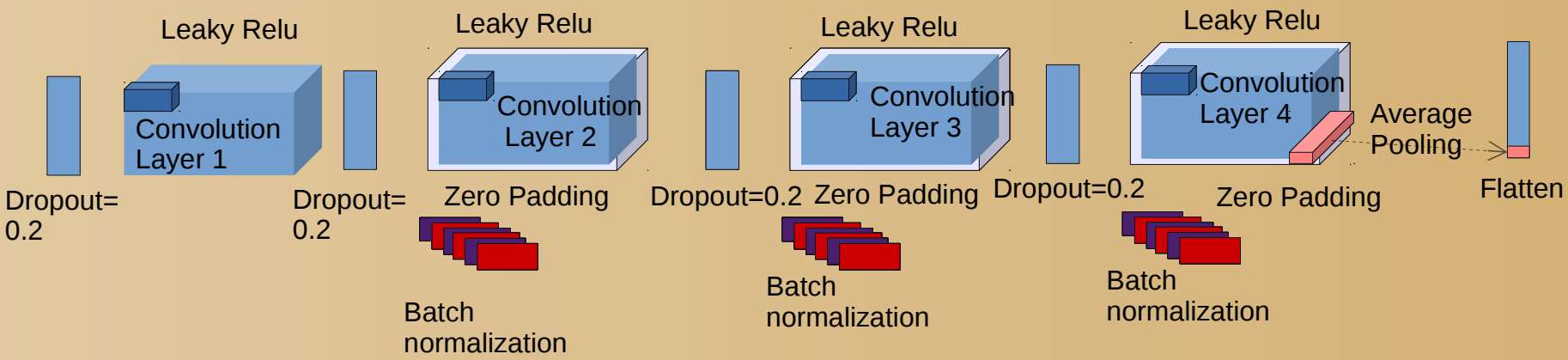
- Conditioned on Energy as:
  - Super-imposed on latent noise vector:
    - Embedding(Shruti)
    - Latent Space x Energy
  - Additional Input to:
    - Dense layer ?
    - Convolutional layers ?
- Conditioned on angle:
  - Waiting for samples with variable input trajectory



Network Architecture



GENERATOR



DISCRIMINATOR

https://github.com/svalleco/3Dgan/tree/Energy-gan

Validation with Python ???

- Why Python(Matplotlib)
  - Keras uses Python
  - Hdf5 Format
  - Python → hdf5 → root files → Analysis (Save on disk)
  - Python → Numpy Arrays → Analysis (No need to save)
  - Memory wasted

https://github.com/svalleco/3Dgan/tree/Energy-gan/keras/analysis

Validation

- GAN vs GEANT Comparison
- 224 plots
  - Maximum Energy position
  - Energy deposited along different axis
  - Discriminator outputs
  - Ecal Energy Deposition
  - First two moments
  - Ratio of first half to second and first to total

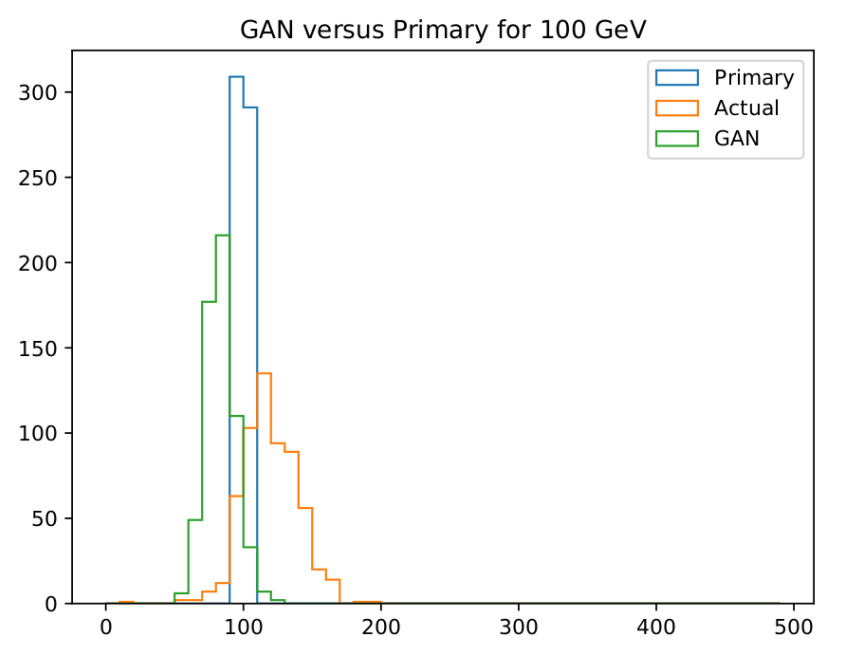
Loss function

- Mean Absolute Error
  - Low energies → Over estimated
  - High energies → Under estimated
  - Discriminator → Biased
  - ECAL sum → Flat
- ECAL sum loss
- Mean Absolute Percentage Error
- Loss weights

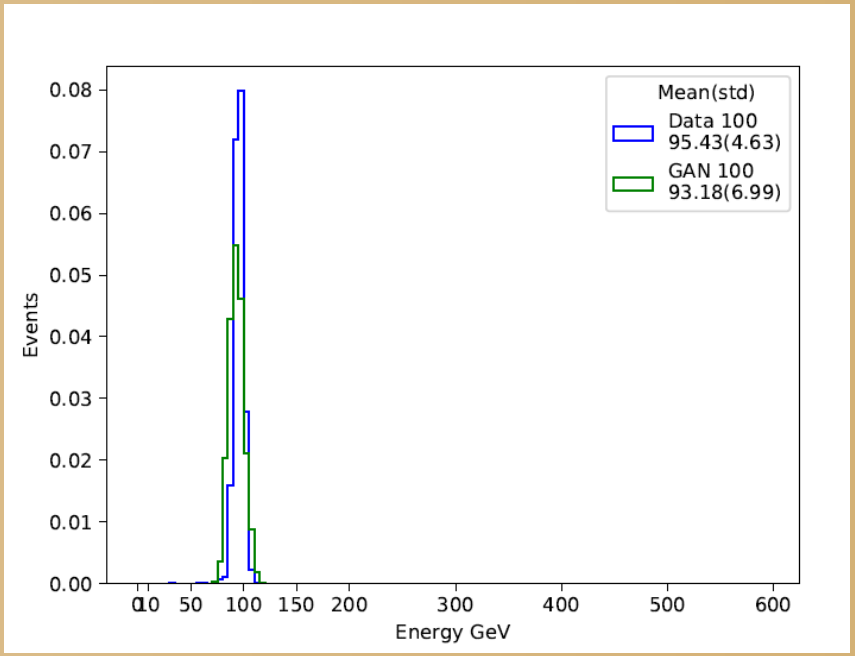
Other Improvements

- Architecture:
  - Layers, filters, kernels
- Improved Training
  - Generating images at same primary energy as data images

Reconstructed Primary Energy

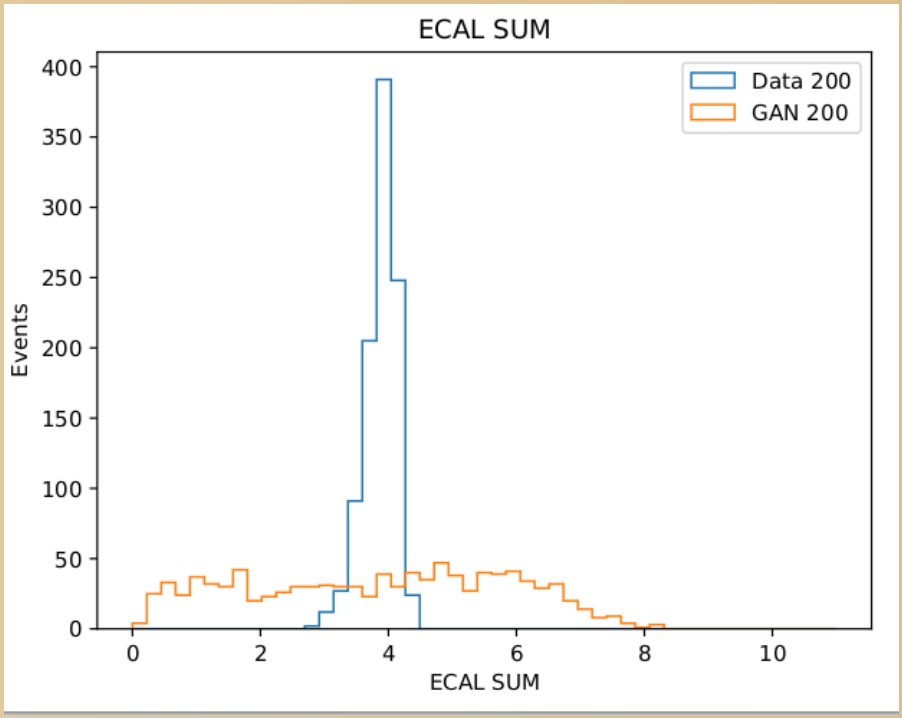


MAE

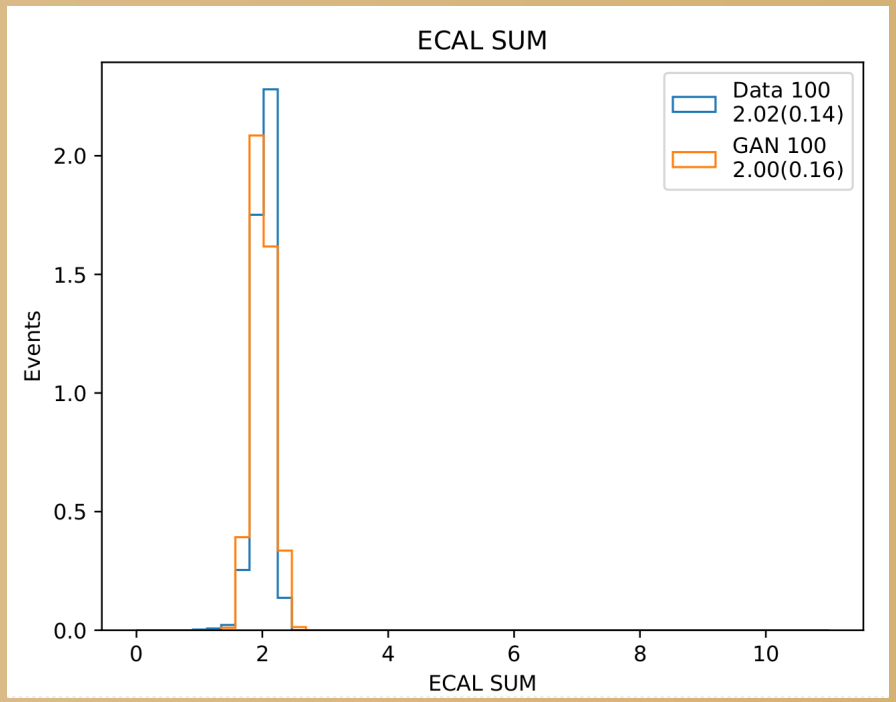


MAPE

MAE- ECAL Total Energy

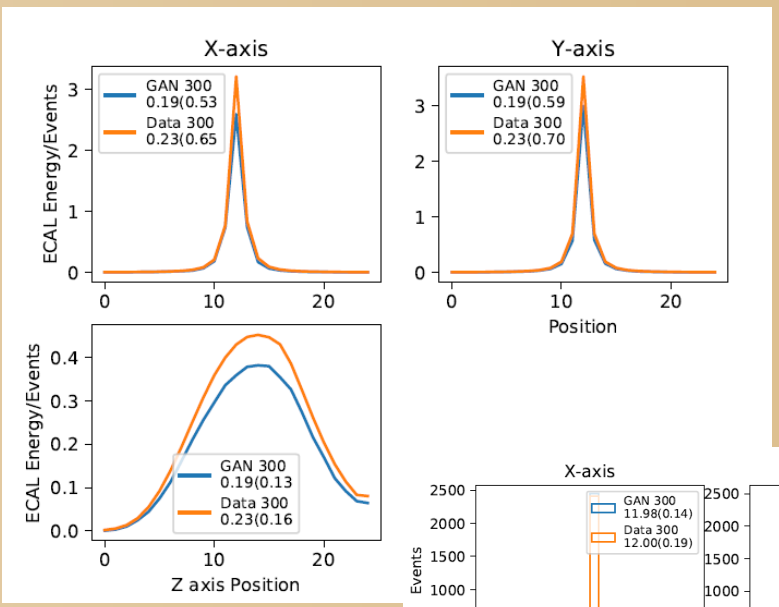


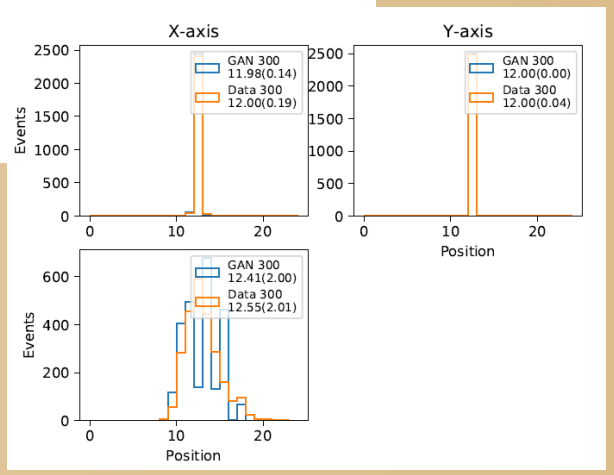
Before Ecal loss

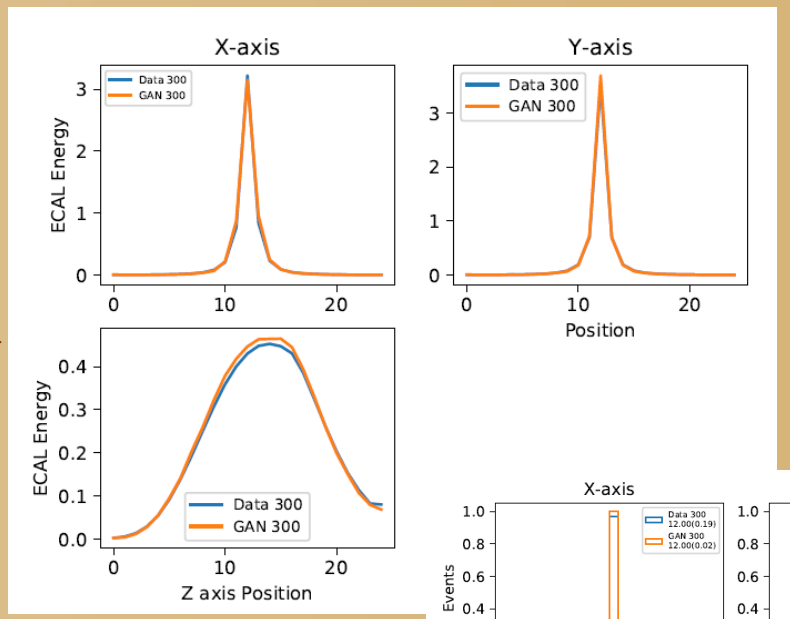


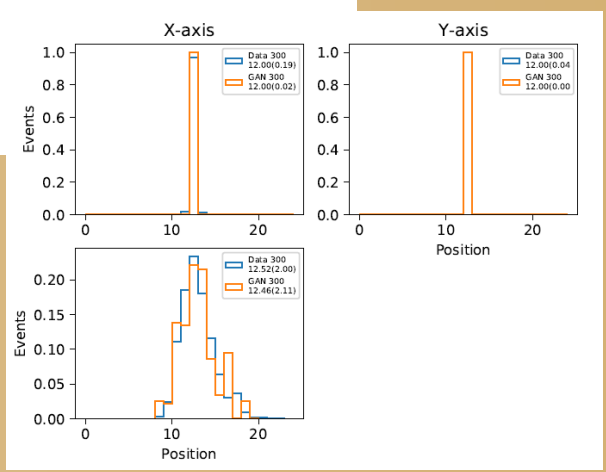
After Ecal loss

Loss Weights

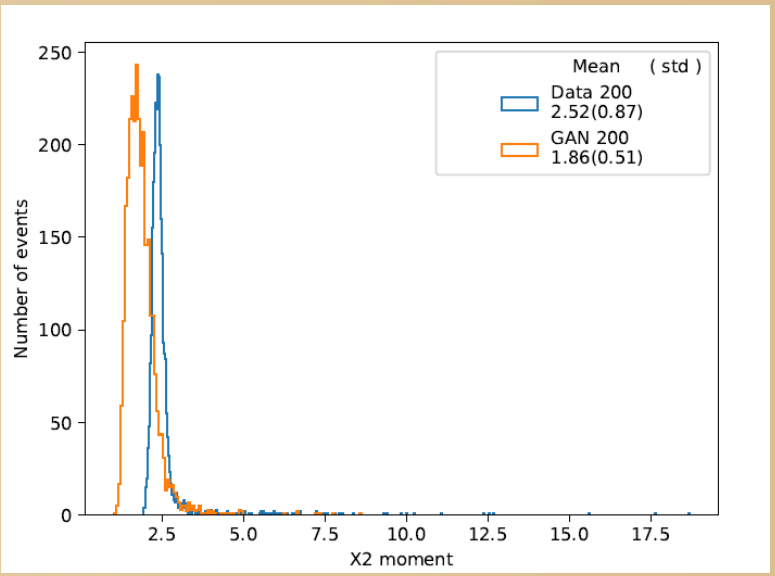


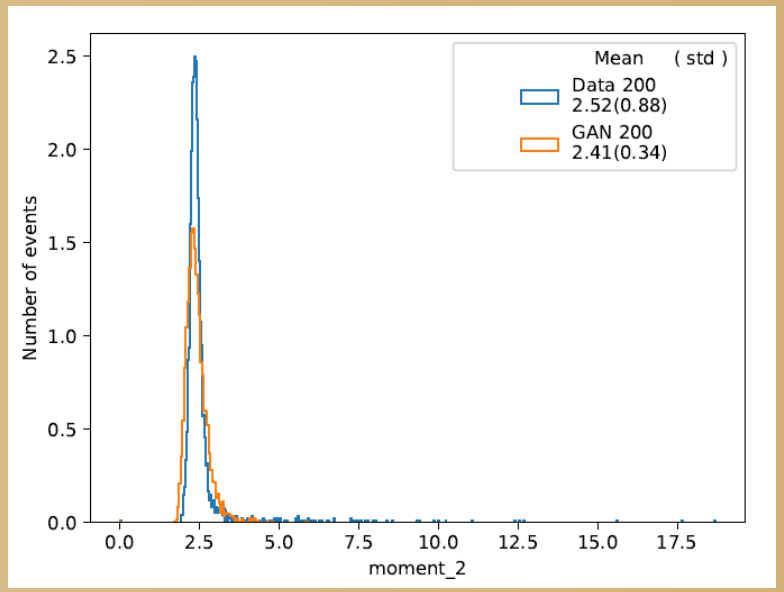






Improved Training





Some numbers

No.	Quantity	Discription	Size
1.	Memory	Data size	25 G bytes
2.		Discriminator Weights	300 k bytes
3.		Generator Weights	3.5 M bytes
4.		Architecture	2.5 k bytes
5.	Time	Simulate with Geant4(Intel Xeon E5-2683)	56000 ms/shower
6.		Simulate with GAN(i7 laptop)	66 ms/shower
7.		Simulate with GAN(GeForce GTX 1080)	0.04 ms/shower
8.	Parameters	Discriminator	73,450
9.		Generator	3,457,012

Hyper parameter scan

- Hyper Parameters
  - Grid search
  - Intelligent search
    - Spearmint
    - Hyperase
    - Skopt https://scikit-optimize.github.io/

Objective Function

- Quality of Images + Physics accuracy ≠ - loss function
- Design of a single metric
  - Relative Average Squared error on Energy deposition along x, y and z axis Histograms
  - Relative Average Squared error on 1st and 2nd moments along x, y and z axis

Data Generator

- Adlkit developed for LCD Calorimeter Dataset
  - https://github.com/anomalousdl/adlkit
- Efficient use of memory
- Loading batch wise
- Parallel threads for fetching data
- Pre processing + shuffling
- Use larger Training set
- Run on all platforms

Publications

- NIPS 2017

  https://dl4physicalsciences.github.io/files/nips_dlps_2017_15.pdf

- ACAT 2017

  https://indico.cern.ch/event/567550/contributions/2627179/

- Super Computing SC2017

  http://sc17.supercomputing.org/SC17%20Archive/tech_poster/tech_poster_pages/post159.html

Further Goals

- Improve Performance for low energies
- Hyper parameter scan
  - Implementation
- Distributed Training
- Testing TMVA Keras Interface for training and validation work
- Regression Problems in Simulation…….
- Other detectors…….