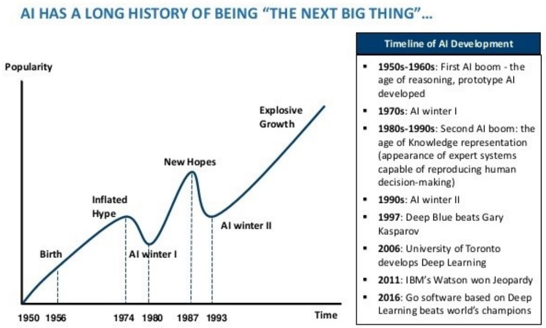
MACHINE LEARNING

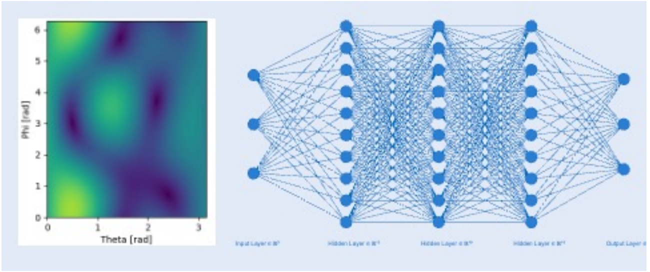
William Phelps

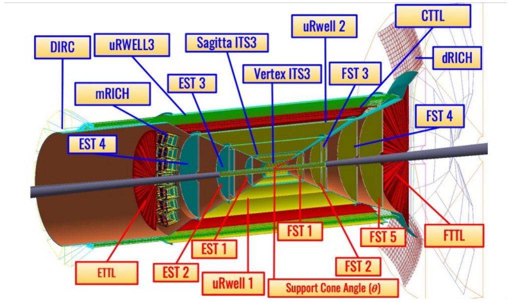
Christopher Newport University/Jefferson Lab

Roadmap

- Machine Learning and Neural Networks (Historical Perspective)
- Deep Learning - Partial Wave Analysis
- Machine Learning - Detector Optimization



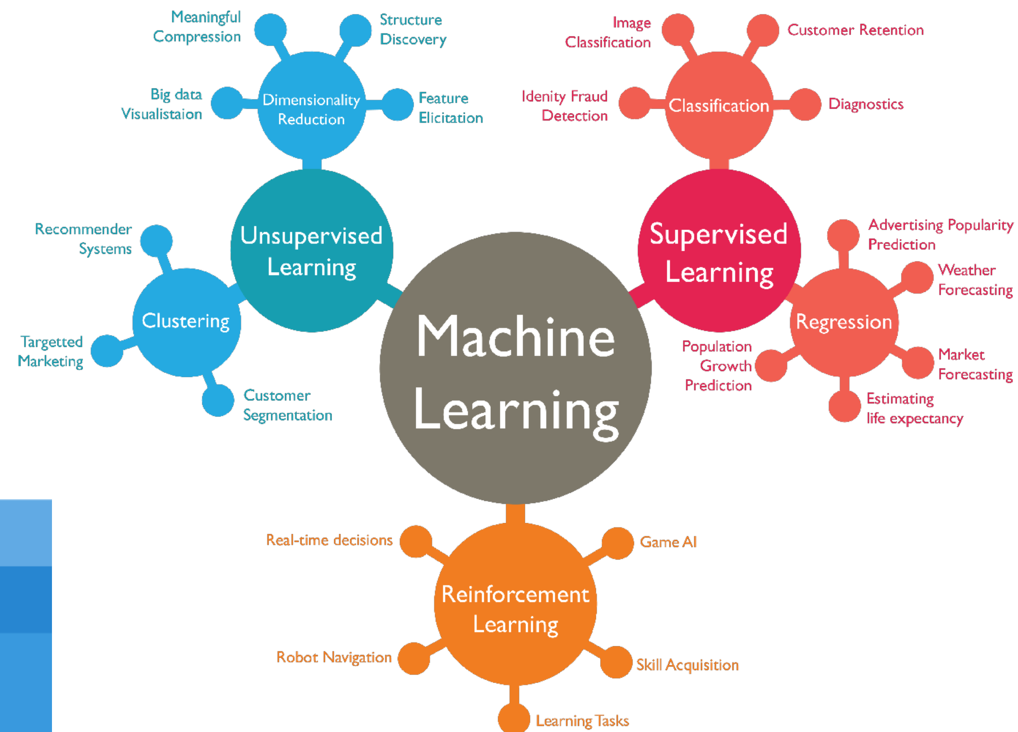


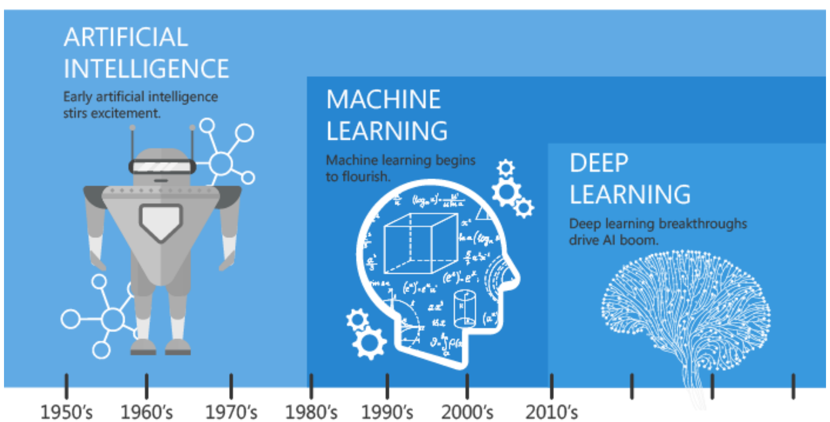


t time

Machine Learning

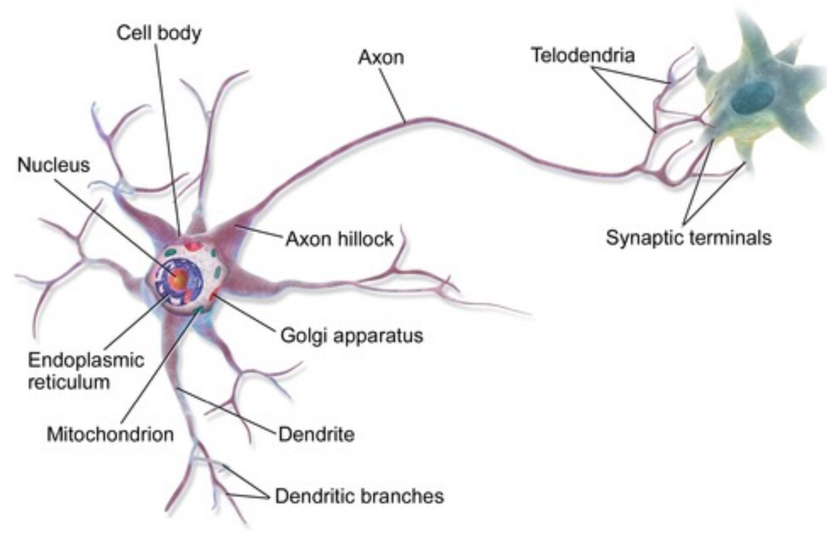
- Machine learning is a broad field with heavy application in nuclear physics
- Deep learning has become a very active field in the last decade now that we have computational resources and software libraries that are easy to use





Early History

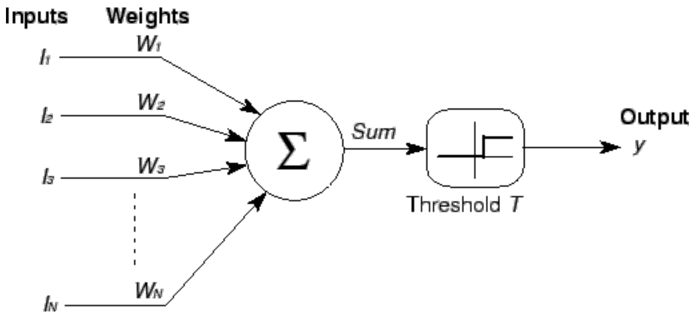
- Artificial Neural Networks were proposed in 1943
  - Warren McCulloch a neurophysiologist
  - Walter Pitts, mathematician
- Donald Hebb furthered ANNs
  - ‘Hebbian Learning’ — a model of learning based on neural plasticity, proposed by Donald Hebb in his book “The Organization of Behaviour” often summarized by the phrase: “Cells that fire together, wire together.”



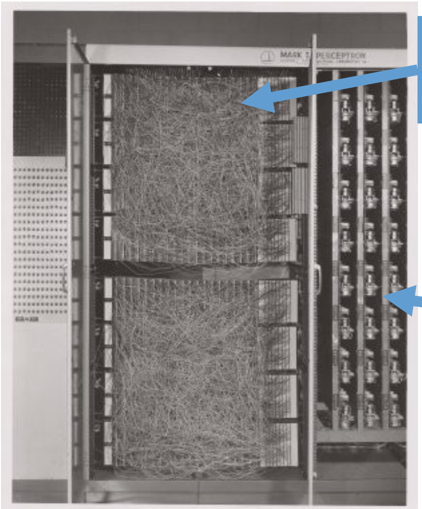
- The human brain has approximately 86 billion neurons
  - ~100 trillion connections
  - ~300k mi of connections
  - Axons can be very short or up to a meter in length

Early History

- Frank Rosenblatt created the first machine called the Mark I Perceptron in 1958
- It was based on the McCulloch-Pitts neuron
  - Warren McCulloch
  - Walter Pitts
- The Mark I Perceptron was connected to a 20x20 pixel "camera" made of cadmium sulfide photocells



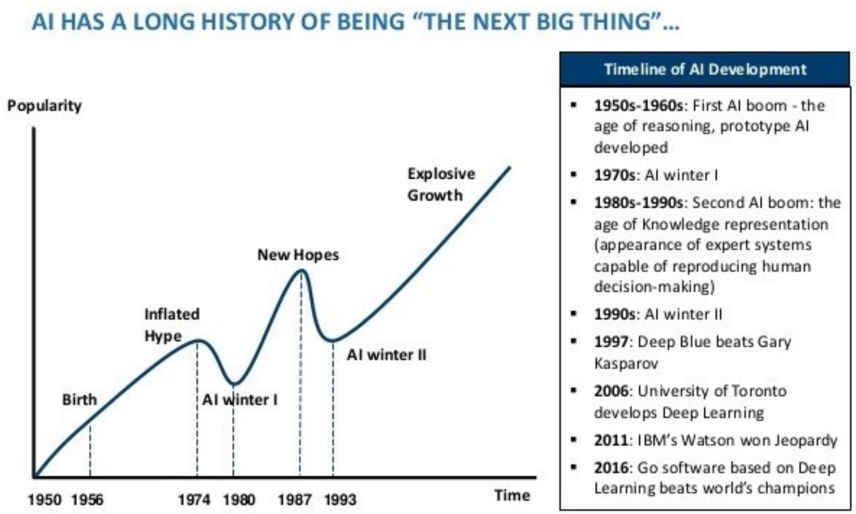
McCulloch-Pitts Neuron



Mark I Perception - 1958

Early History and Hype!

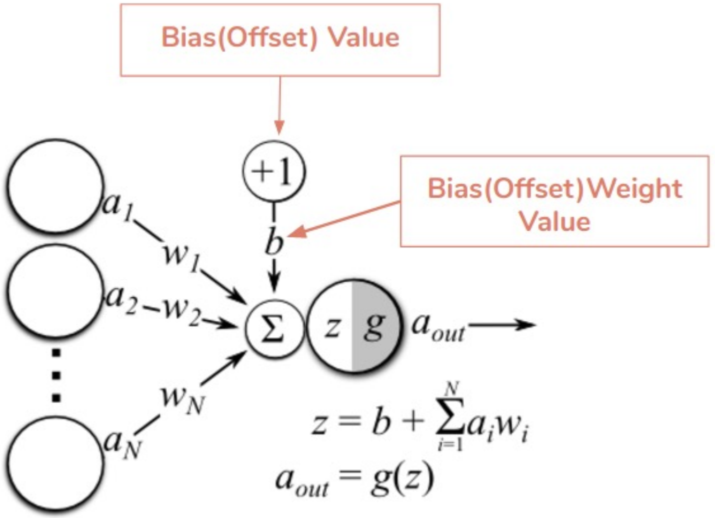
- Research continued into the 60's and was receiving much attention and funding
  - Machines were only able to separate linearly separable data
- In 1969, "Perceptrons" was published by Marvin Minsky, founder of MIT AI Lab and Seymour Papert the director of the AI Lab
  - Caused much doubt in MLPs and contributed to AI Winter



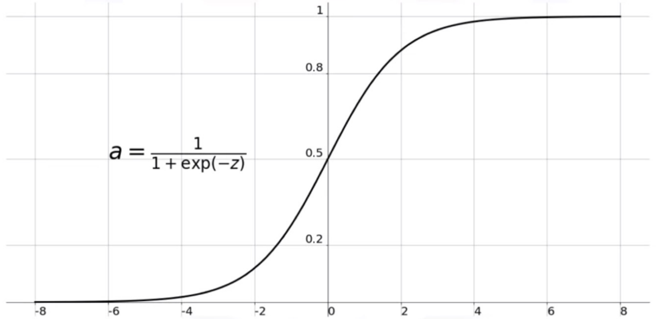
https://www.actuaries.digital/2018/09/05/history-of-ai-winters/

Neuron and Activation Functions

- Neural Networks are composed of individual Neurons
- Neurons have inputs and those inputs have trainable parameters called weights
- The weights are applied to the respective inputs and summed together
- After summing the inputs/weights and bias the output is determined by the activation function



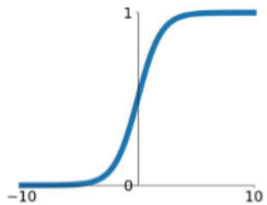
Sigmoid Function



Activation Functions

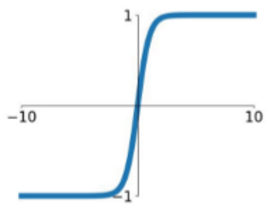
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



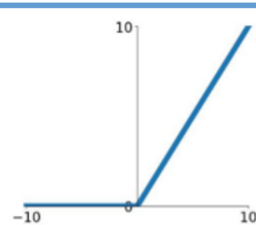
tanh

$\tanh(x)$



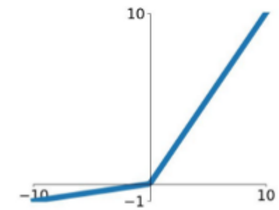
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

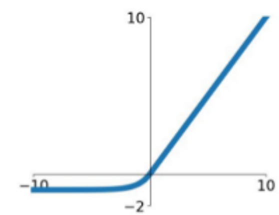


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

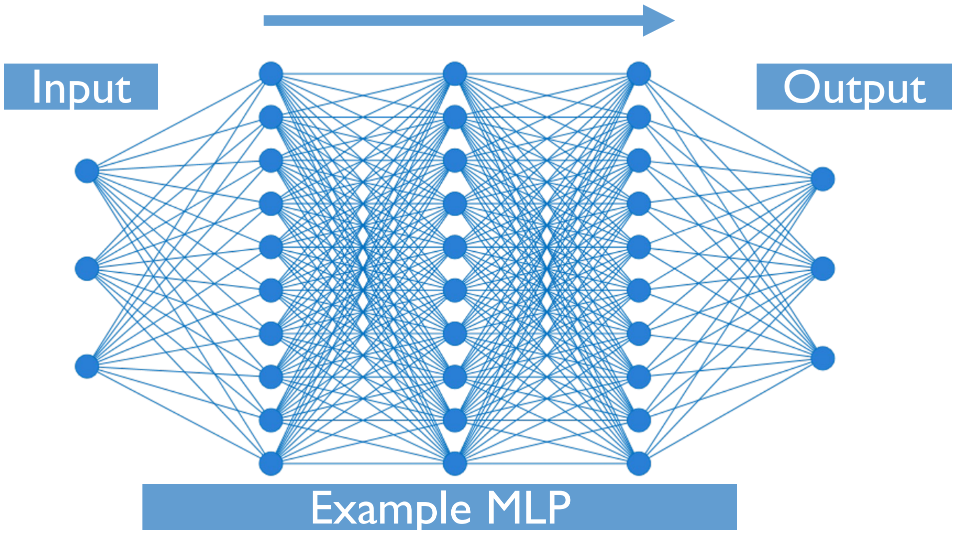
ELU

$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

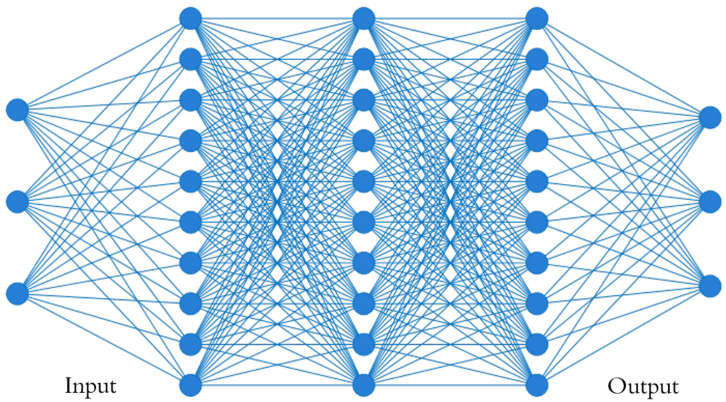


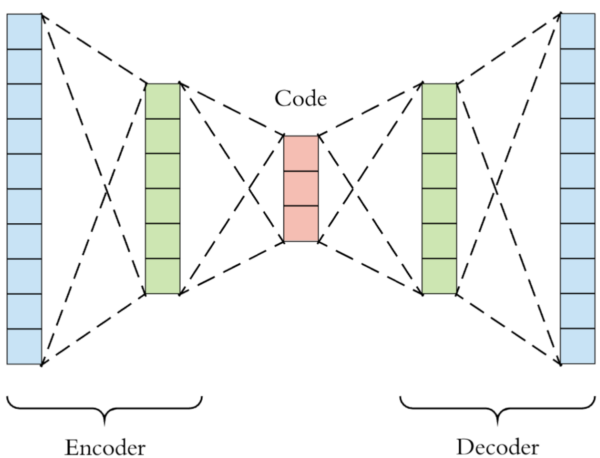
Multilayered perceptron MLP

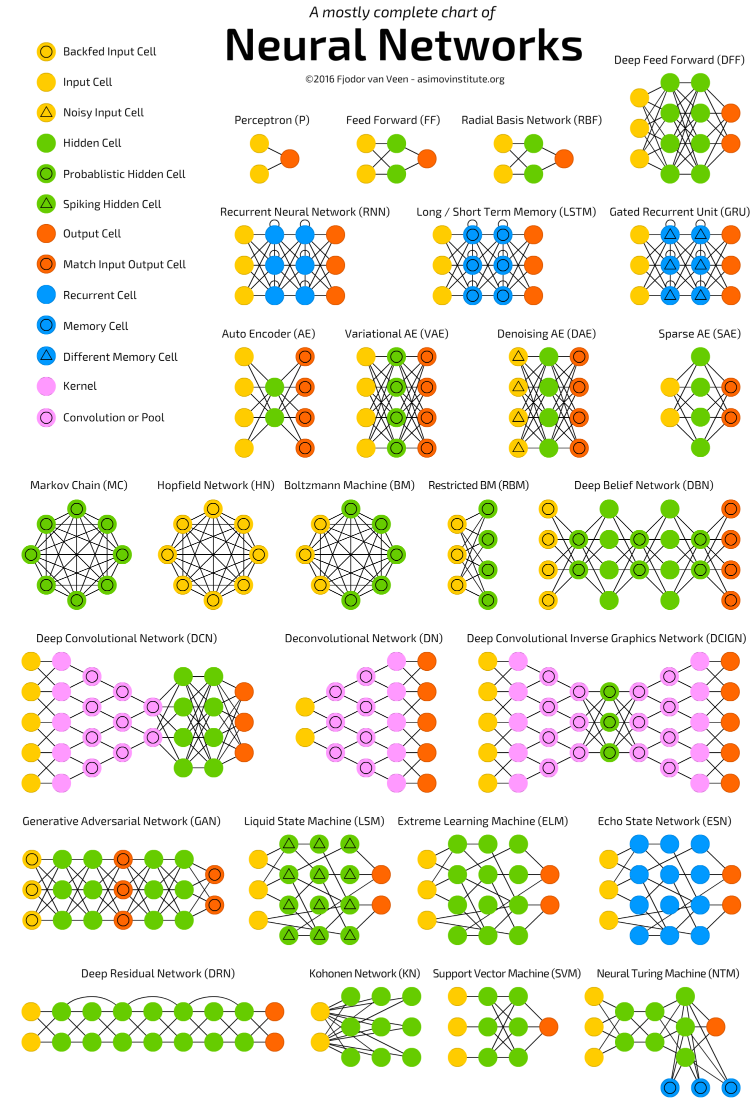
- The arrangement of neurons within a neural network is called a network architecture
- In the 1980's creating layers of perceptrons became a popular network architecture
  - It is still widely used today
- This architecture is powerful but posed significant challenges when it came to setting weights and biases
  - Future: Backpropagation/Gradient Descent



Neural Network Architectures

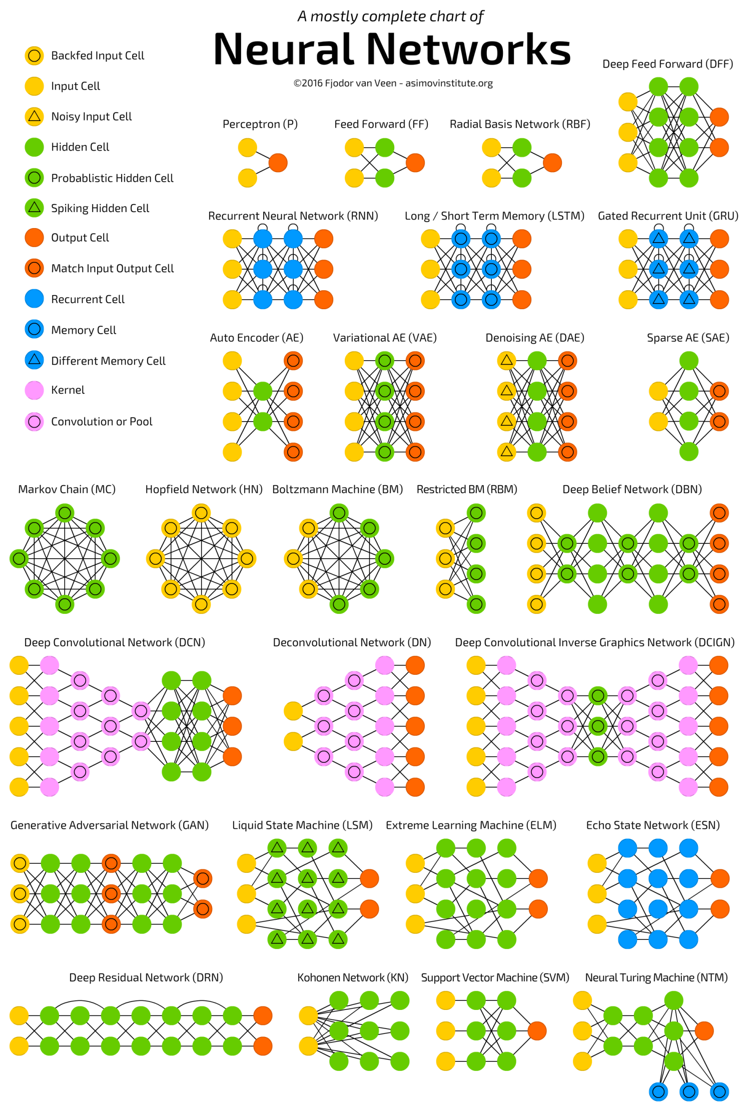






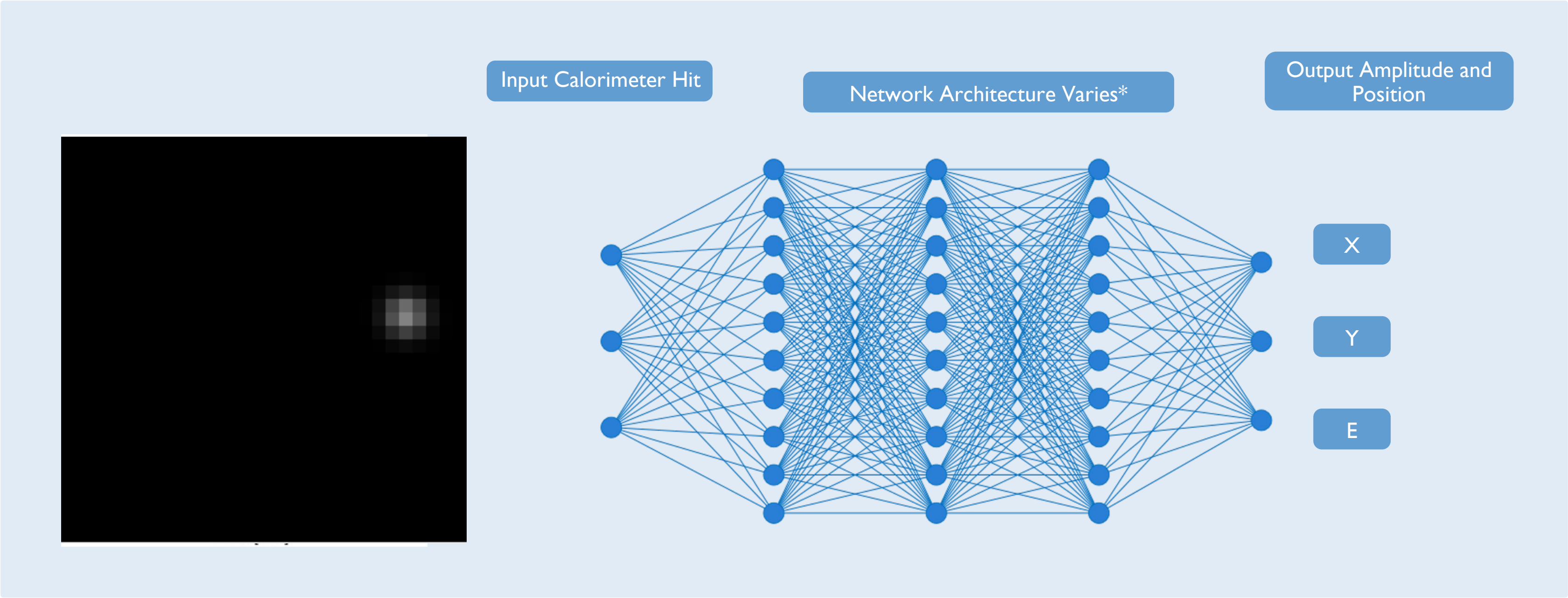
Neural Network Architectures

- MLPs – General all around neural network, what everyone thinks of when you mention NNs
- Recurrent Neural Networks/LSTMs – Used for time series data
- Convolutional NNs – Good for images or other data where the relative position of features is important
- Autoencoders – Unsupervised learning, good for denoising
- Generative models – GANs, VAEs, etc. Good for generating new data based on prior knowledge
- …



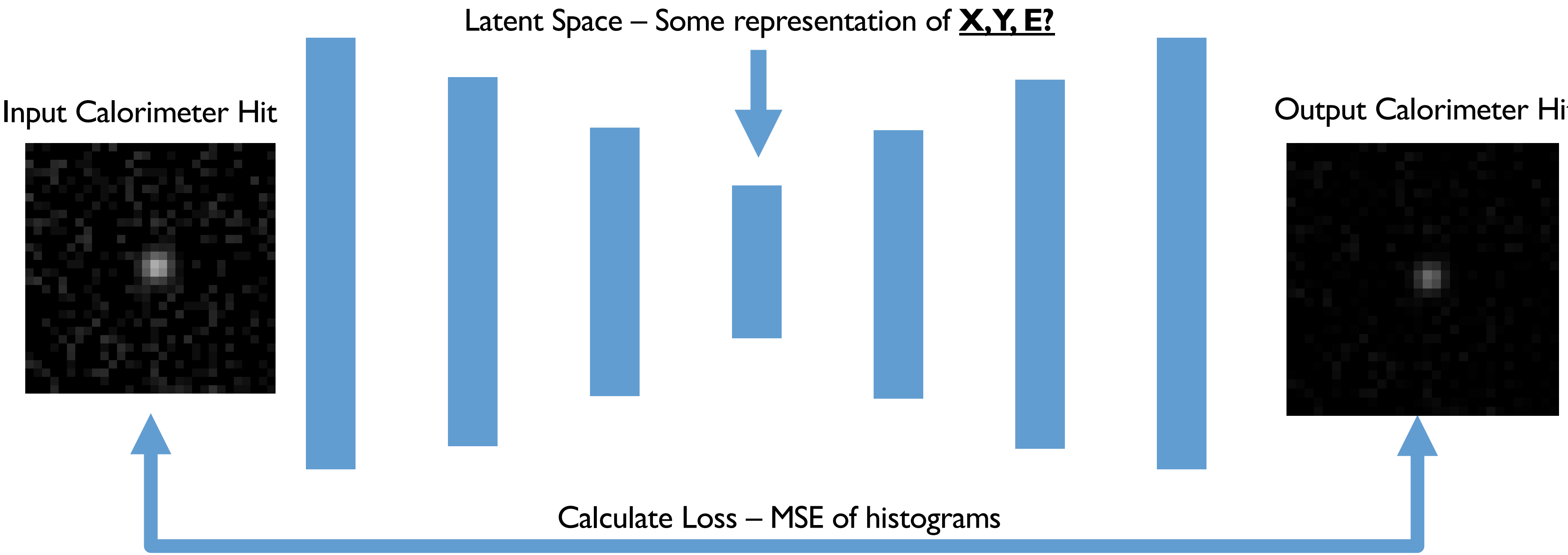
MLP Example – Hackathon

- Read in 30x30 pixel hits from CSV files, provide calorimeter hit position (x, y) and amplitude of the hit.
- Model Architecture: MLP w/3 layers of 2000 neurons, relu activation function



Autoencoder Example – Hackathon

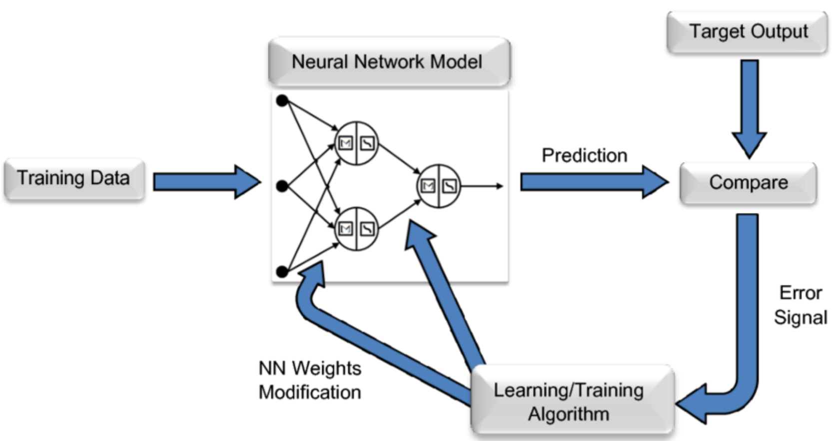
Unsupervised learning!



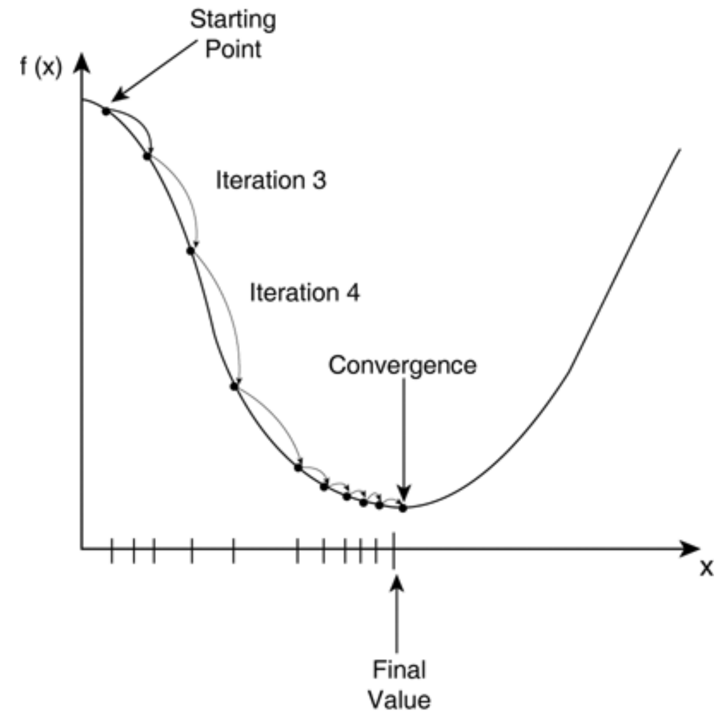
2000⟶1000⟶100⟶3 …

Backpropagation and Gradient Descent

- Up until 1986 backpropagation was not used in neural networks
- Without backpropagation you cannot calculate the gradient for each weight
- Loss function defined to compare prediction w/target output



Training a Neural Network



Gradient Descent

Tools of the Trade

- Long gone are the days where you would write your own NN Library!!!
- Today there are two main packages competing for dominance: PyTorch (Facebook/Meta) and Tensorflow/Keras (Google)
- Python 3.9 – Anaconda
  - Keras/TensorFlow - NN Libraries
  - Pandas/Numpy - Data Handling
  - Matplotlib - Visualization
- Many GPU nodes that Scientific Computing division has available
  - Either through Jupyterhub or interactively using slurm to request a node
  - Several machines with 4 NvidiaTitan RTX GPUs and some with 14 Nvidia T4 GPUs

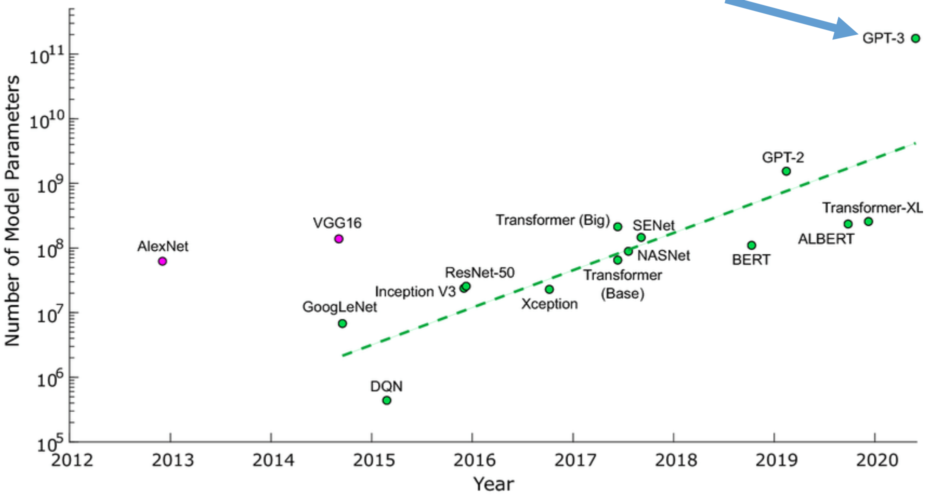
Sample Training Script

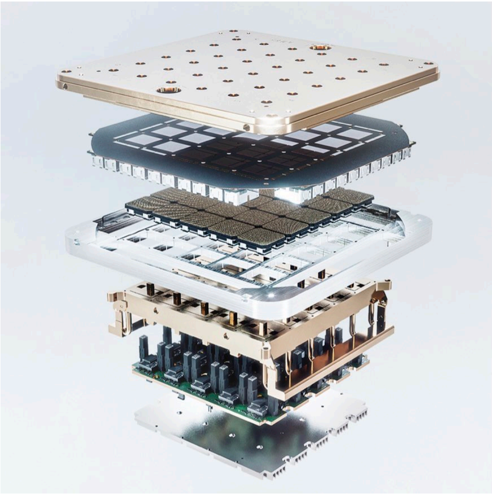
```
test = pd.read_csv("TRAIN/TRAIN.csv")
labels = pd.read_csv("TRAIN/TRAIN_labels.csv")
activation = 'relu'

model = Sequential()
model.add(Dense(units=1000, activation=activation, input_shape=(3600, )))
model.add(Dense(units=1000, activation=activation))
model.add(Dense(units=1000, activation=activation))
model.add(Dense(units=2))
model.compile(optimizer=adam(lr=.001), loss='mean_squared_error', metrics=['accuracy'])

model.fit(test, labels[labels.columns[1:]], epochs=300, batch_size=256, validation_split=0.2)
```

Computing Resources

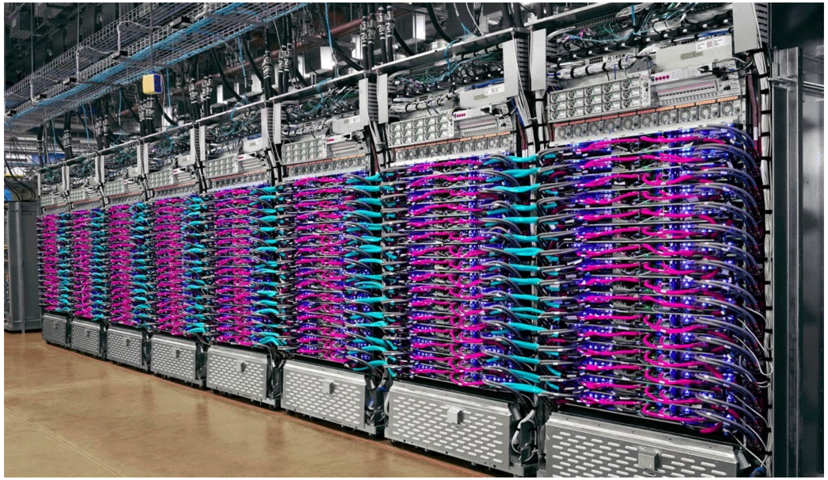
- First trained on GPU in 2009
- Models are getting larger and larger and require specialized hardware to train

350 GB in fp16 precision (!!!)





Tesla D1 Dojo Chip



Google TPU

Partial Wave Analysis

PyPWA

- A python-based software framework designed to perform Partial Wave and Amplitude Analysis with the goal of extracting resonance information from multi-particle final states.
- In development since 2014 and has been significantly improved with each revision. Version 3.4 out now.
- Efficient amplitude analysis framework including multithreading and CUDA support
- Optimizers include: Minuit, Nestle (or add your own!)
- NIM Paper in development!

Website: https://pypwa.jlab.org
GitHub: https://github.com/JeffersonLab/PyPWA

Group Members

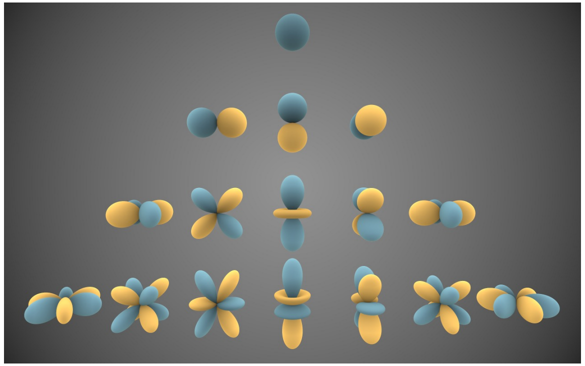
Carlos Salgado (NSU/Jlab)
Mark Jones (NSU)
William Phelps (CNU/Jlab)
Michael Harris (NSU)
Andru Quiroga (CNU)
Bruna Goncalves (NSU)
Nathan Kolling (CNU)

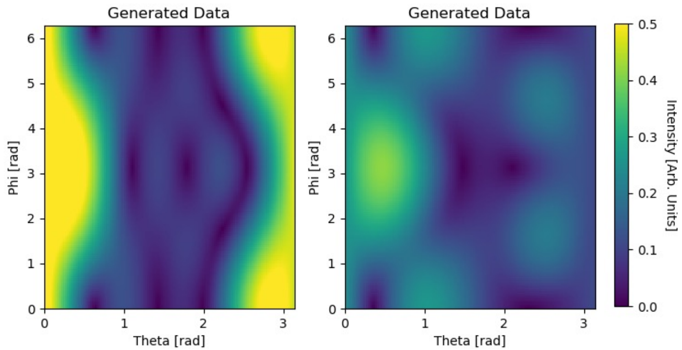
Former Group Members

Josh Pond
Stephanie Bramlett
Brandon DeMello

PWA using Neural Networks

- Generate datasets using decay amplitudes (linear combination of spherical harmonics) with the following quantum numbers
  - L = 1,2,3
  - m = 0,1
  - ϵ_R= -1,+1
  - 9 total waves ("fit parameters")





$$I(\Omega) = \sum_k \sum_{\epsilon_R} \sum_{l,|m|,l',|m'|} {}^{\epsilon_R}Y_l^{|m|}(\Omega)\ {}^{\epsilon_R}V^k_{l,|m|}\ {}^{\epsilon_R}V^{k*}_{l',|m'|}\ {}^{\epsilon_R}Y_{l'}^{|m'|*}(\Omega)$$

Production Amplitudes

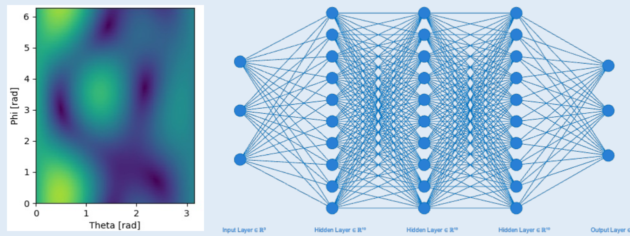
Decay Amplitudes

Early Results

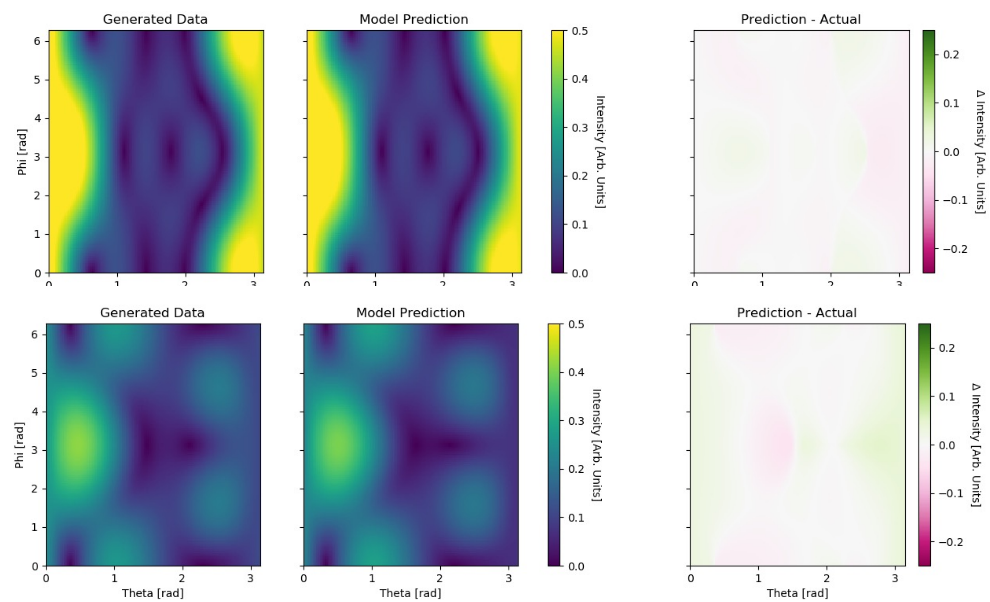
TensorFlow Keras python

- We compare the intensity function and compare it to the model prediction
- Model Architecture:
  - 128x128 2D histogram as input
  - 9x128 Dense Layers – Relu activation
  - 9 production amplitudes as output
- In order to deal with the vast amounts of data we used generators to generate data for each epoch on the fly

Input Histogram



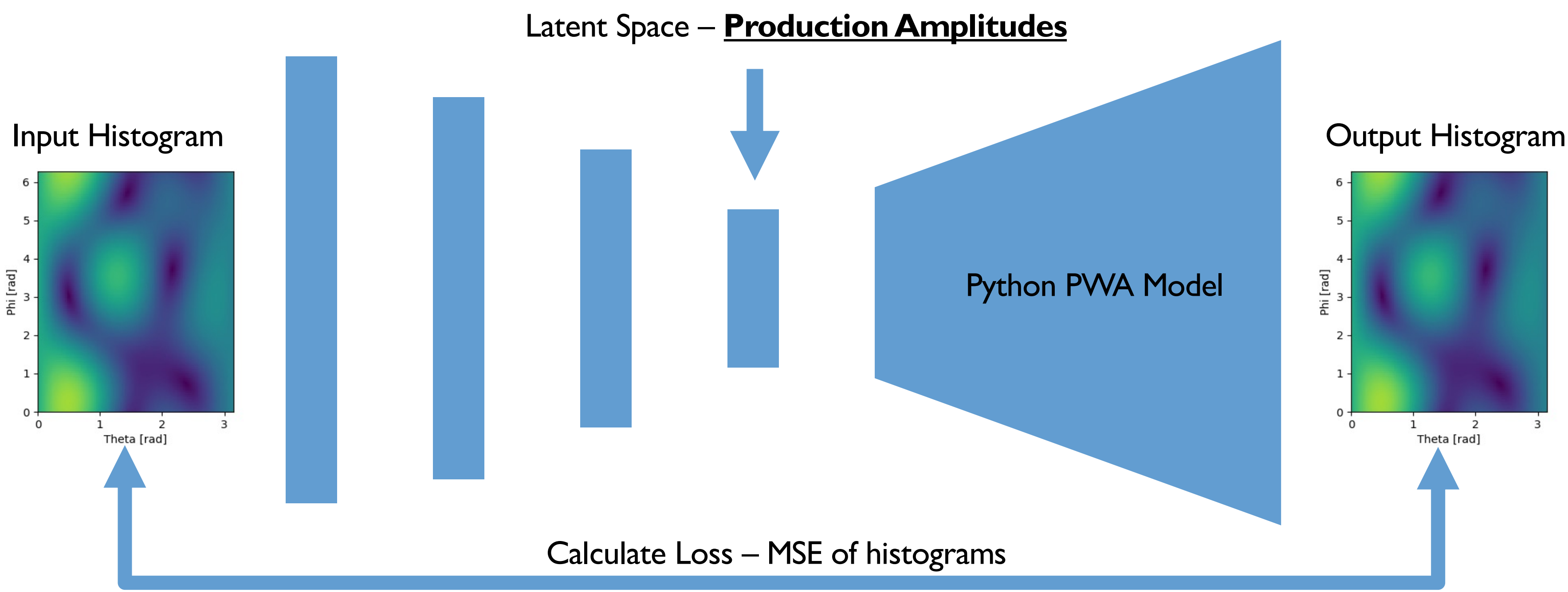
Output - Production Amplitudes



Useful Tools: Generators, Complex Valued Deep Learning

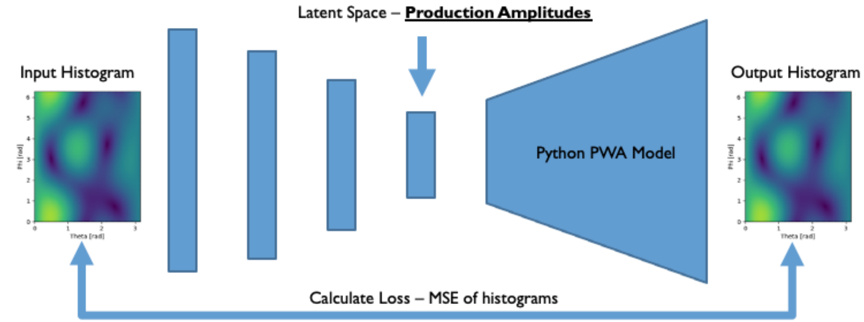
Autoencoder for PWA

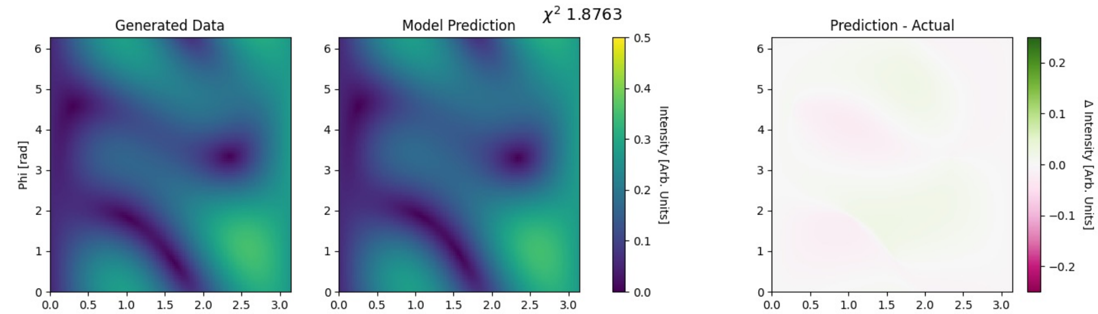
Unsupervised learning!

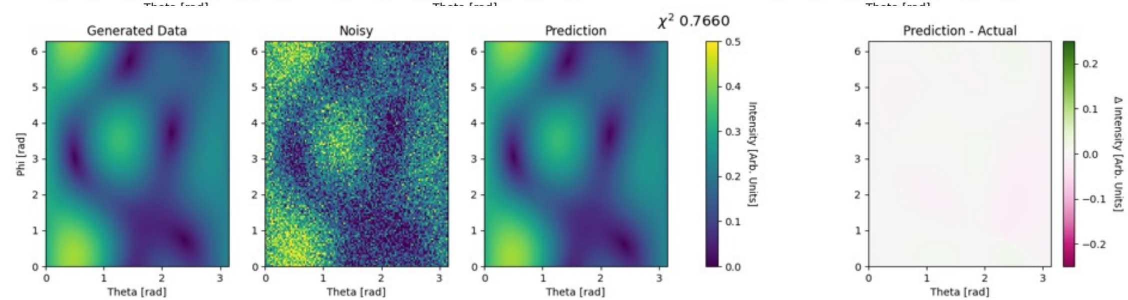


Autoencoders for PyPWA

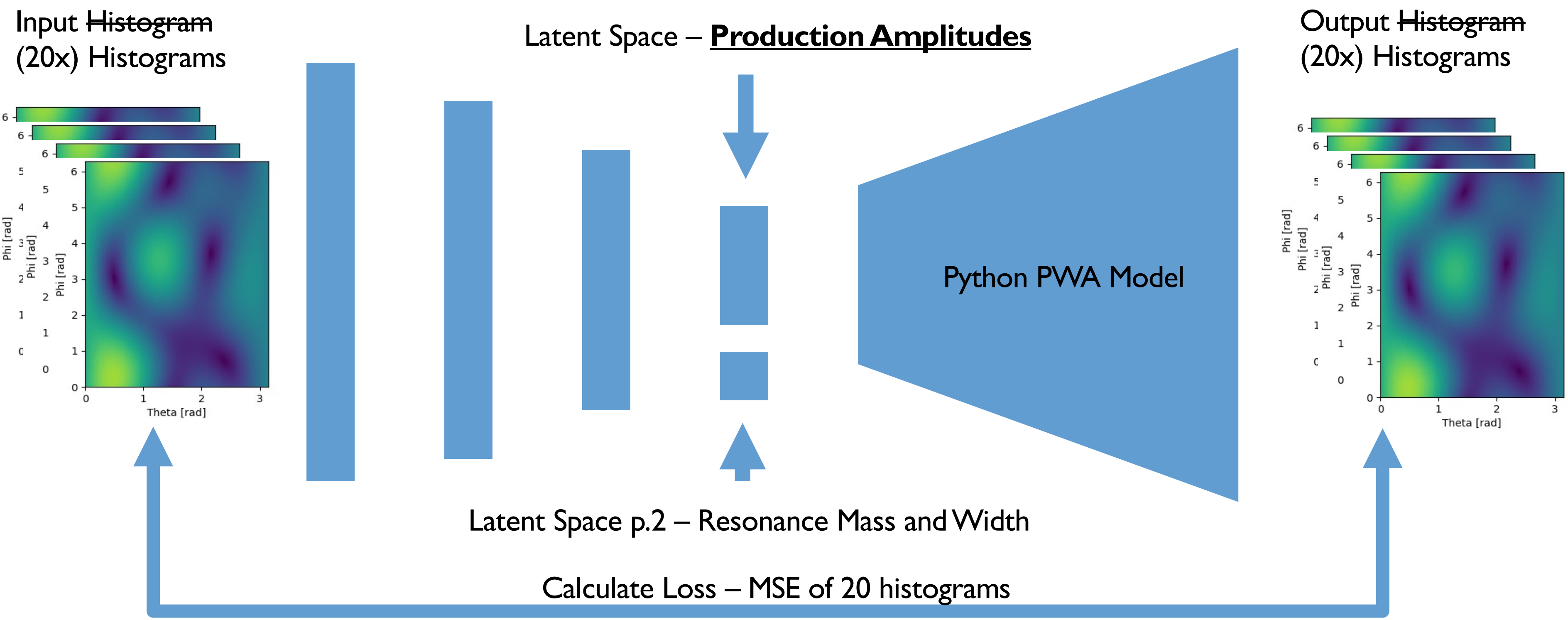
- Encoder portion is a standard MLP, but without labels!
- Decoder is a PyPWA model that takes in production amplitudes and produces a histogram
- Autoencoders *dramatically* improved the accuracy!
- Even works well for noisy data





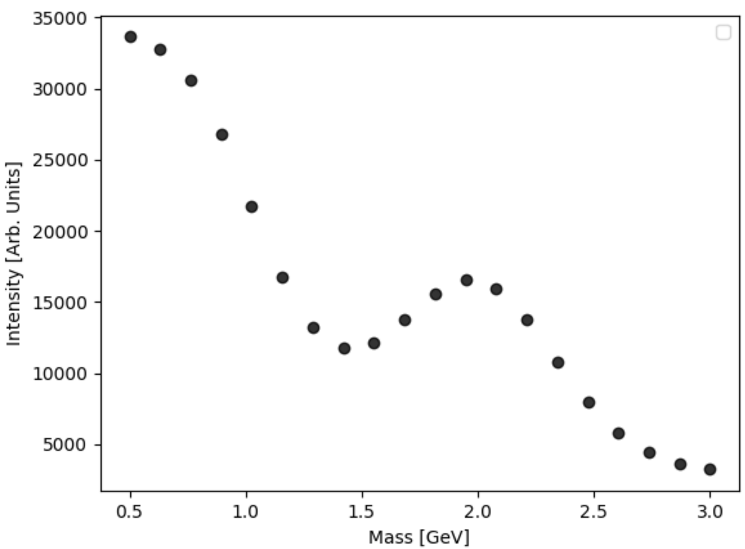


Mass Dependent Autoencoder work for PWA



The Mass-Dependent Generator

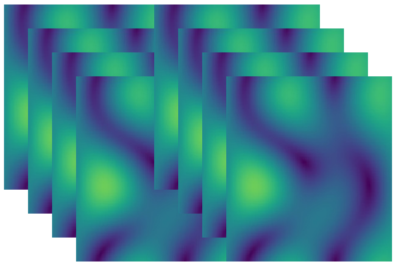
Randomly Generated Event
(Currently One Resonance per Wave)



Production Amplitudes
(Complex)

Breit-Wigner Coefficients
(Mean, Width)

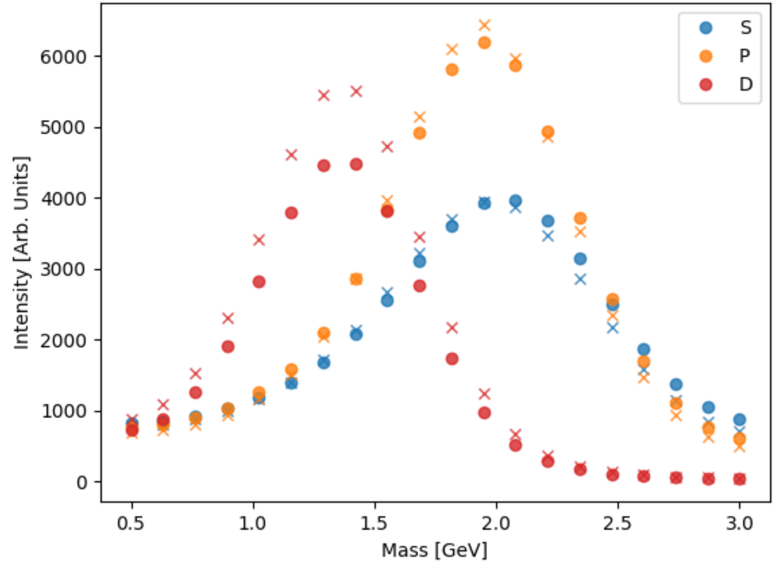
Histogram
Generator



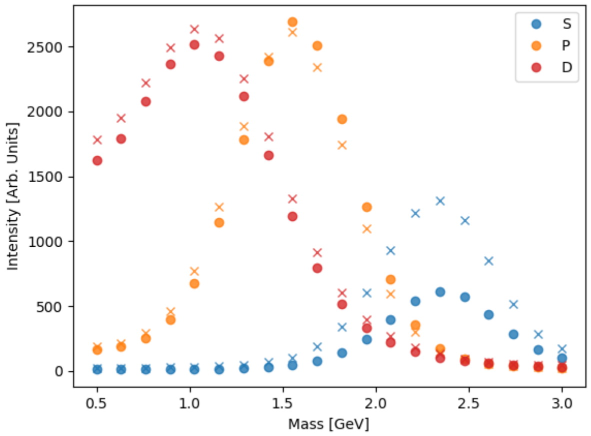
Set of
Histograms
Binned in Mass
(3D-Histogram)

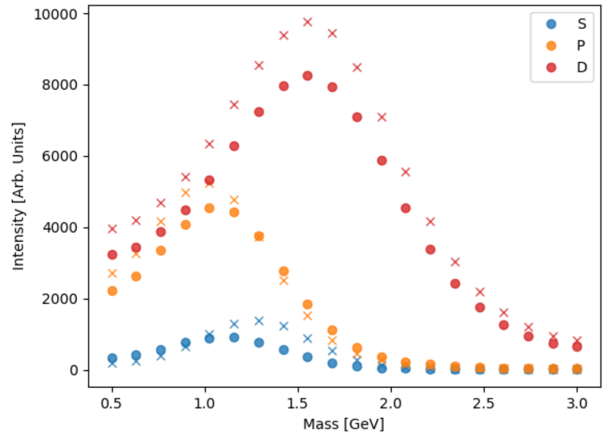
Results

- With a CONV3D input to our autoencoder we see a good agreement with the generated data and inference from our neural networks
- Shown on the right are three different tests with randomly generated data/resonances



Crosses – Generated
Circles – Inference





PWA Summary

- We have been able perform PWA "fits" with neural networks
- Autoencoders dramatically improved the performance
- Future work includes uncertainty quantification, and we are currently investigating Bayesian Neural Networks, dropout during inference, and Variational Autoencoders

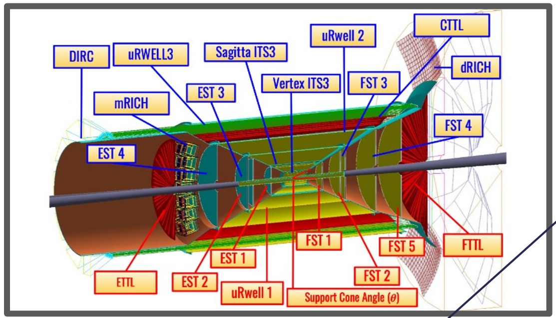
Many thanks to the EPSCI and Data Science group at JLab!
David Lawrence, Thomas Britton, Malachi Schram, Kishansingh Rajput

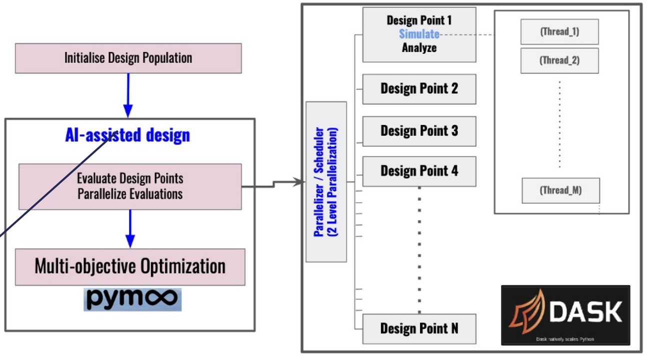
AI-optimized design of the ECCE tracker

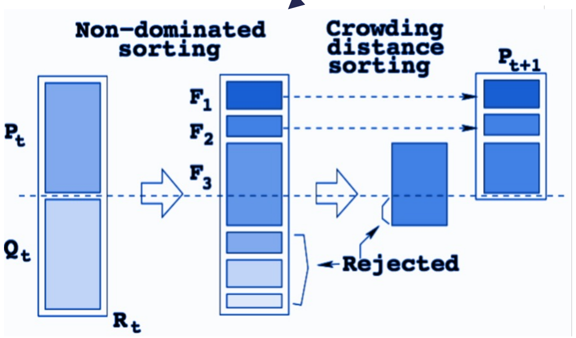
C. Fanelli, Z. Papandreou, K. Suresh

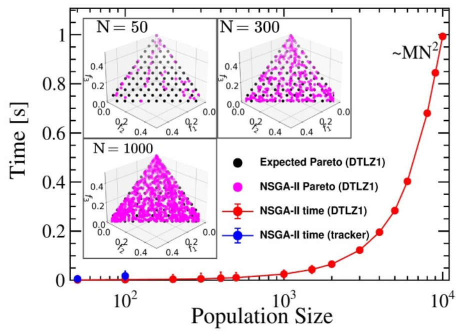
NEW

C. Fanelli, Z. Papandreou, K. Suresh et al (ECCE),
AI-assisted Optimization of the ECCE Tracking System at the Electron Ion Collider
https://arxiv.org/pdf/2205.09185.pdf (submitted to NIM-A)









Detector technology being finalized: more realistic material budget, simulation details (e.g. background), and constraints to be included

Plan to encode all this in the existing framework

Opportunity to explore other optimization strategies and scalability; more advanced distributed pipelines/workflow

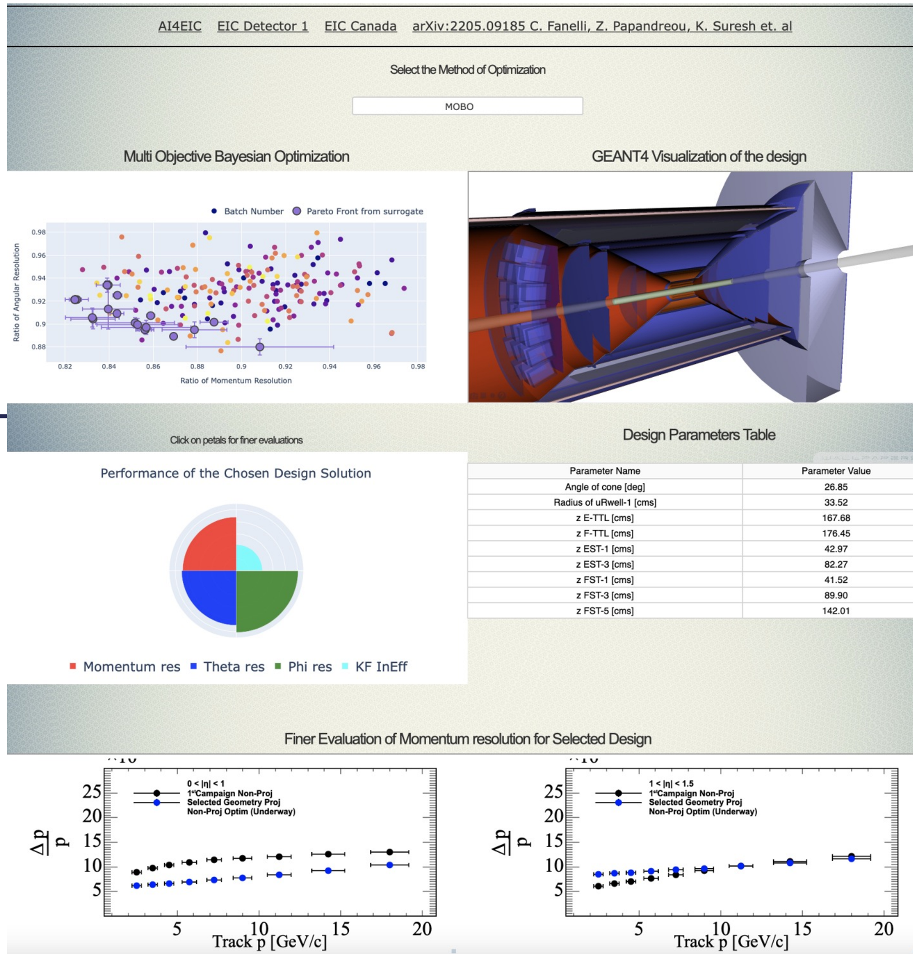
Multiple objectives: different physics analysis

In few months we will make a decision on the SW framework (solution adopted by ECCE so far VS ATHENA one). With Regina I am planning to explore both coupling to Fun4All and to DD4Hep

AI-optimized design of the ECCE tracker

C. Fanelli, Z. Papandreou, K. Suresh

NEW



Navigate the Pareto front solutions (different detector design points) interactively

https://ai4eicdetopt.pythonanywhere.com

(credits K. Suresh)

https://eic.ai

Flux+Mutability

NEW

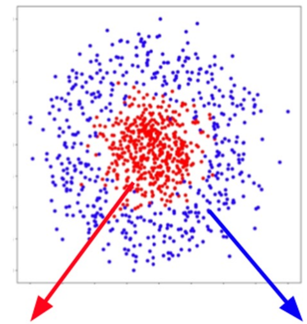
Flux+Mutability
https://arxiv.org/pdf/2204.08609.pdf (submitted to IOP MLST)

"Flux+Mutability": A Conditional Generative Approach to One-Class Classification and Anomaly Detection

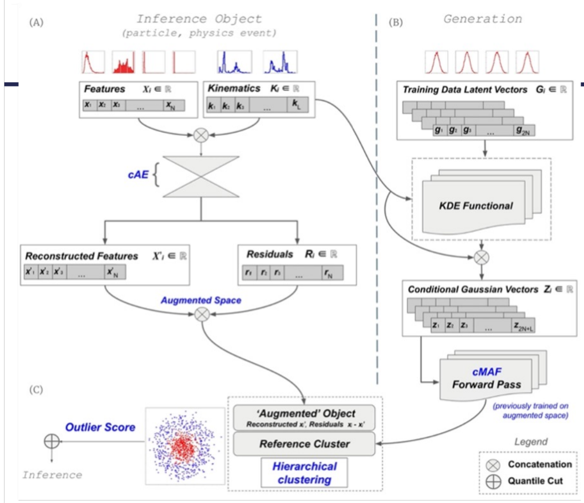
C. Fanelli[1,2,*], J. Giroux[3,*], Z. Papandreou[3,‡]
[1] Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
[2] The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts 02139, USA,
[3] University of Regina, Regina, SK S4S 0A2, Canada
E-mail: *cfanelli@mit.edu, *jng887@uregina.ca, ‡zisis@uregina.ca

20 April 2022

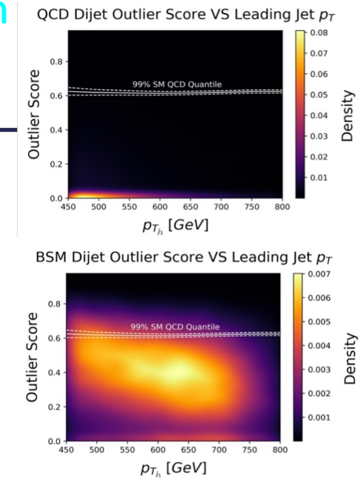
Abstract.
Anomaly Detection is becoming increasingly popular within the experimental physics community. At experiments such as the Large Hadron Collider, anomaly detection is at the forefront of finding new physics beyond the Standard Model. This paper details the implementation of a novel Machine Learning architecture, called Flux+Mutability, which combines cutting-edge conditional generative models with clustering algorithms. In the 'flux' stage we learn the distribution of a reference class. The 'mutability' stage at inference addresses if data significantly deviates from the reference class. We demonstrate the validity of our approach and its connection to multiple problems spanning from one-class classification to anomaly detection. In particular, we apply our method to the isolation of neutral showers in an electromagnetic calorimeter and show its performance in detecting anomalous dijets events from standard QCD background. This approach limits assumptions on the reference sample and remains agnostic to the complementary class of objects of a given problem. We describe the possibility of dynamically generating a reference population and defining selection criteria via quantile cuts. Remarkably this flexible architecture can be deployed for a wide range of problems, and applications like multi-class classification or data quality control are left for further exploration.



known unknown



- OCC - GlueX
- AD - LHC Dijet
- Agnostic cuts (fully unsupervised)
- Lends itself to data monitoring



Quantile	TPR	TNR
1σ (68%)	68.18 ± 0.22 %	93.20 ± 0.06%
2σ (95%)	95.15 ± 0.10 %	42.40 ± 0.22%
3σ (99%)	99.03 ± 0.05 %	11.82 ± 0.14%
Fiducial cuts (99%)	98.92 ± 0.05 %	2.35 ± 0.06%

	Ours	Fraser et al.	Cheng et al.
AUC	**0.885 ± 0.003**	0.87	0.89

Summary

- There are many ML/AI projects going on at the lab but I was only able to mention a few
  - AI Townhalls have been held in 2020 and 2021 and are a great way to see what the community is doing
- If you are interested in participating keep an eye out for Jlab/EIC hackathons and workshops (https://eic.ai)
- Workshop in October 2022!

AI FOR THE ELECTRON ION COLLIDER - EVENTS

AI4EIC - October 10-14, 2022

2nd General Workshop on Artificial Intelligence for the Electron Ion Collider
Venue: William and Mary

Contacts:

support@eic.ai

What is a Hackathon?

- It can mean many things but, in this context, it is a competition and a learning experience with relevant problems usually generated w/toy models
- Specifically, we will talk about the AI hackathon that took place on July 27th, after the AI Town Hall
  - But you will see a few photos of the hackathon in 2020!
- Typically, there are teams with ~4-5 people
  - Not everyone has to be experienced
- Held over the course of one day with NP based problems/datasets

Hackathon in March 2020 – AI4NP

Hackathons

- Foster community engagement
- Great learning experience
- And most of all, fun!

Oxpecker Reborn Team Roster

- Gagik Gavalian (JLab)
- Tyler Viducic (ODU)
- Andru Quiroga (CNU)
- Torri Jeske (JLab)
- William Phelps (CNU/JLab)

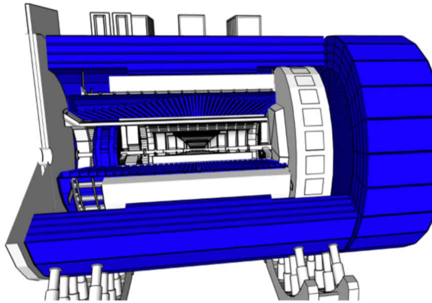
Hackathon in March 2020 – AI4NP

AI Townhall Hackathon - July 2021

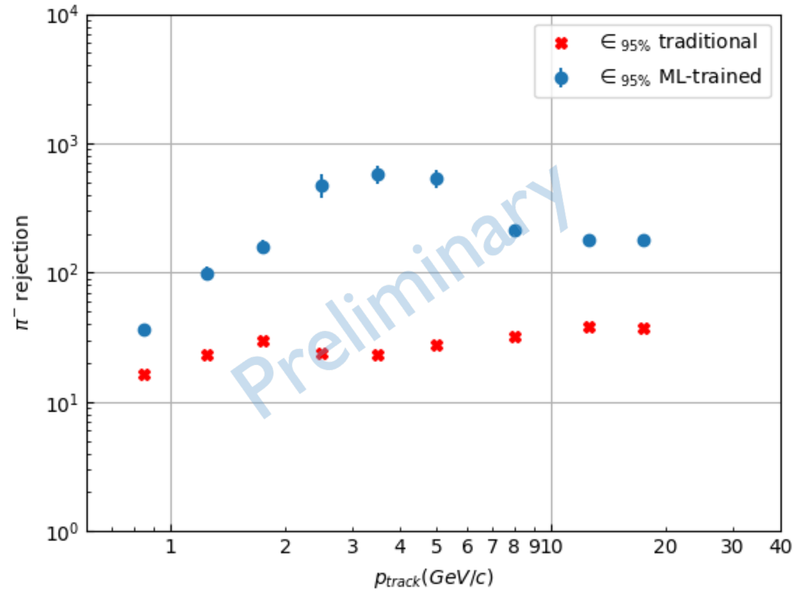
Backup

BECAL PID Study

A. Quiroga, W. Phelps, C. Fanelli, and J. Huang



- Higher pion rejection compared to conventional methods when considering high electron efficiency (~95%)
- Work is in progress (started on thanksgiving)
- Interface with ECCE software: reco-track, track projection, 7x7 calorimeter towers near track (track-based clustering by AI) [Link to details]
- Many models tried: MLPs, CNNs, Multi-Input models, Autoencoders, GANs.
- Ongoing hyperparameter tuning on 14 GPU nodes

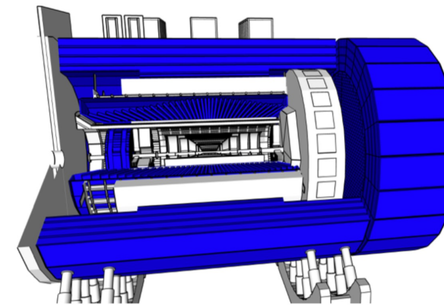


CHRISTOPHER NEWPORT UNIVERSITY

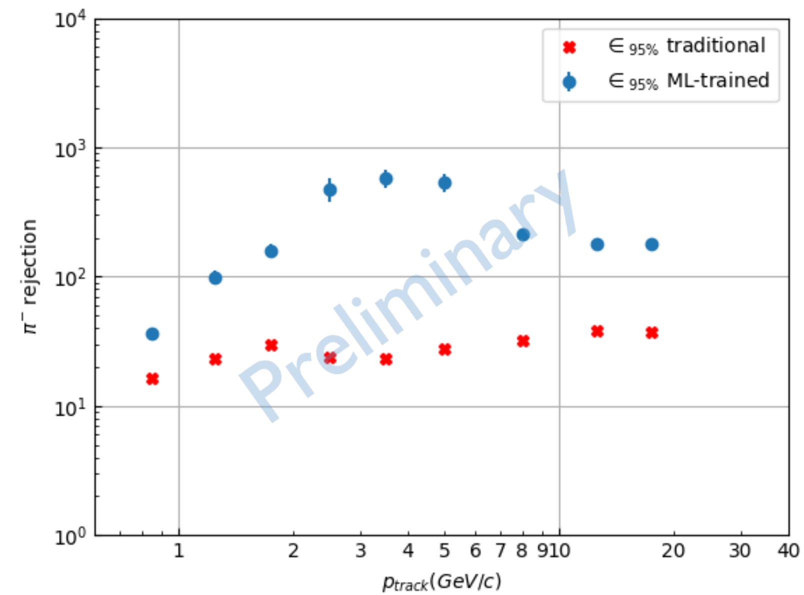
Jefferson Lab
Thomas Jefferson National Accelerator Facility

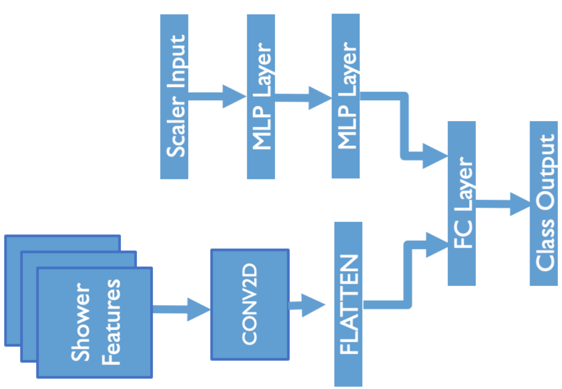
BECAL PID Study

A. Quiroga, W. Phelps, C. Fanelli, and J. Huang



- Higher pion rejection compared to conventional methods when considering high electron efficiency (~95%)
- Work is in progress (started on thanksgiving)
- Interface with ECCE software: reco-track, track projection, 7x7 calorimeter towers near track (track-based clustering by AI) [Link to details]
- Many models tried: MLPs, CNNs, Multi-Input models, Autoencoders, GANs.
- Ongoing hyperparameter tuning on 14 GPU nodes





CHRISTOPHER NEWPORT UNIVERSITY

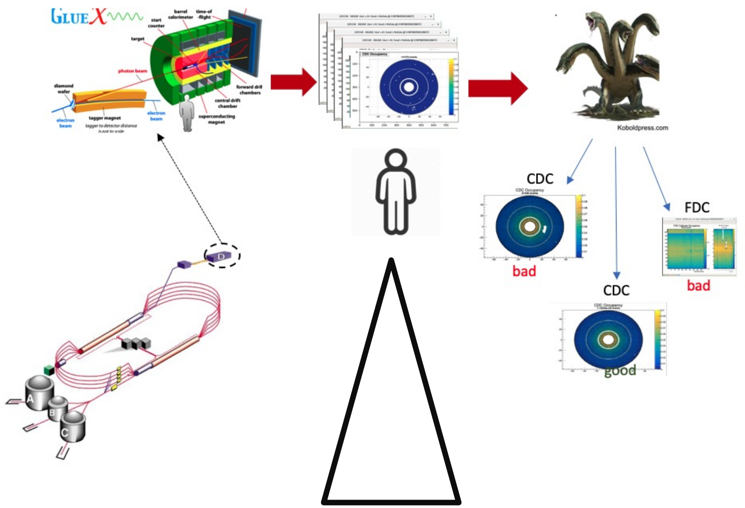
Ongoing Projects – Data Science Group

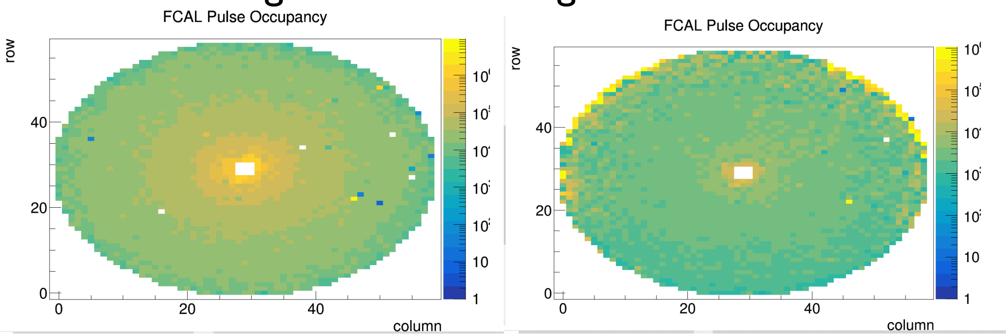
- Hall A SoLID: Kishansingh Rajput
  - Particle ID
- Hall D CCP: Nikhil Kalra
  - Particle ID
- Hall D AIEC calibration: Diana McSpadden
  - Online calibration using Gaussian Processes
- Hydra: Thomas Britton
  - Online Monitoring

Thomas Britton

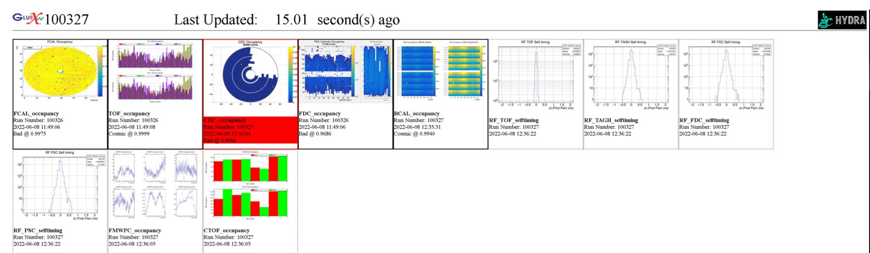
Hydra – Detector Monitoring

- Plenty of detector issues that are not alarmable in the traditional sense but still detectable
- Every run produces an initial >22 plots. More thorough monitoring is performed offline and produces >109 plots. With a run lasting ~2-3 hours every day there are between ~175 and 875 plots to look at every day. Desire AI to augment monitoring.





LED Light pulser left on

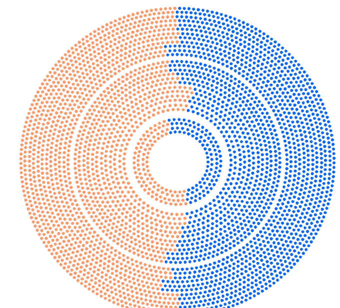


HydraRun also saw the FDC problem, which I probably would have missed inspecting it by eye.

Diana McSpadden

AIEC: Stabilizing Gain in the Central Drift Chamber

- Accelerate the calibration from months to minutes.
    1. Gain Correction Factor: CDC Voltage Gain calibration
    2. Time to Distance: track fitting calibration
- Calibration is required to provide reliable PID for physics analysis
- Considerations:
    - External environmental conditions (temperature, pressure)
    - Changing beam conditions (current)
- Features used:
    - Atmospheric pressure
    - Temperature
    - Current drawn from CDC high voltage board.



GlueX CDC (one half controlled by GP)

Results