DATA SCIENCE IN EXPERIMENT

William Phelps

Christopher Newport University/Jefferson Lab

Data Science

- Heavily involves
  - Data
  - Statistics
  - Machine Learning
    - **In particular this talk will be *heavily* biased towards ML**
- This talk will focus on projects carried out by users and staff

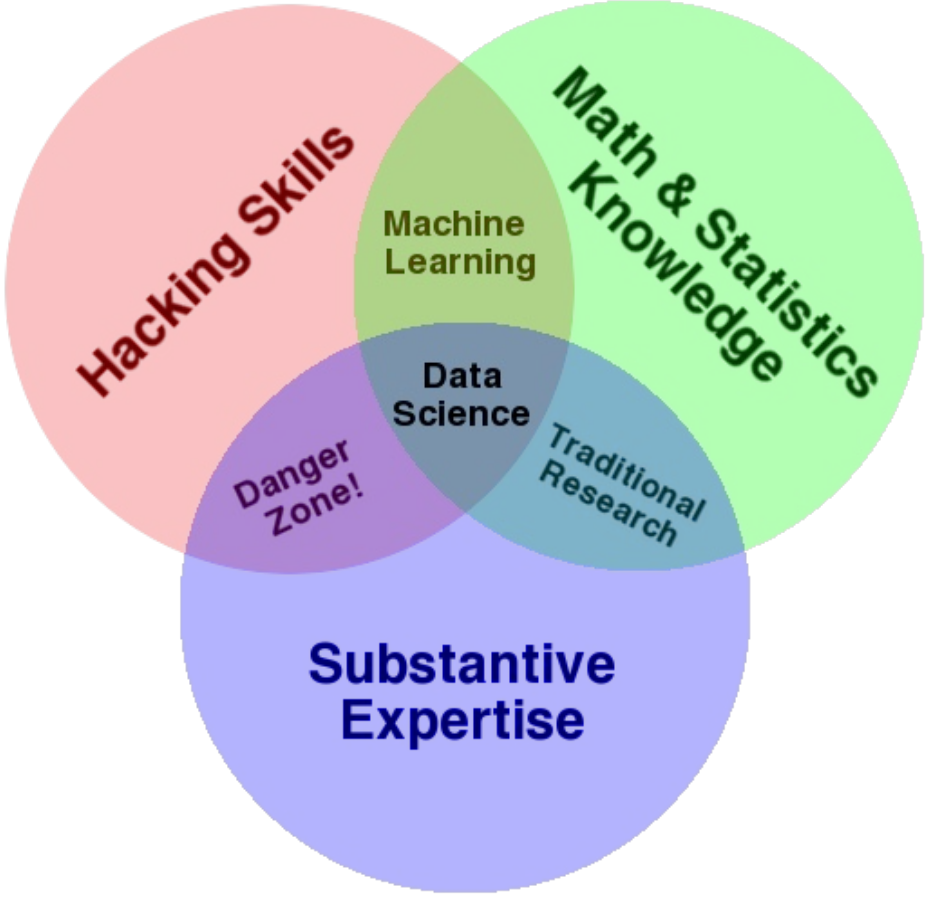
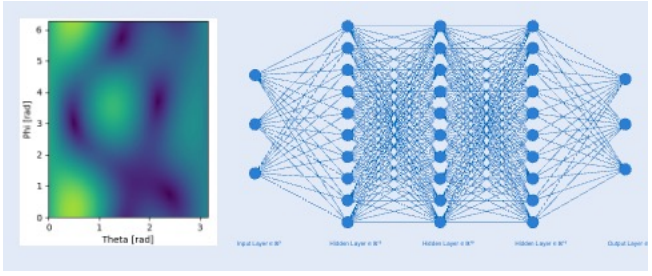
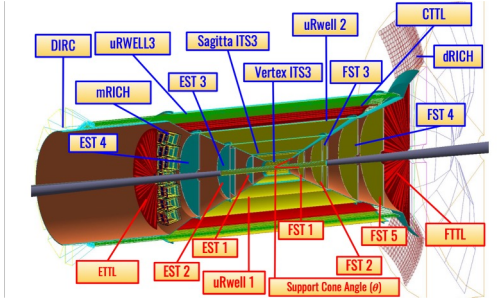


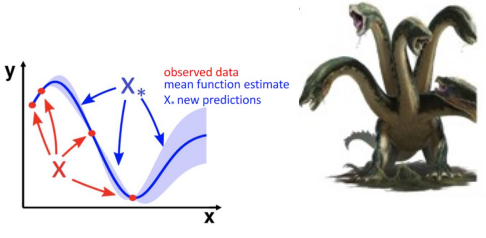
Figure source: Drew Conway

Roadmap

- Deep Learning - Partial Wave Analysis
- Machine Learning - Detector Optimization
- Data Science Group Projects
  - Hydra
  - AIEC







t time

Partial Wave Analysis

- 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!

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)

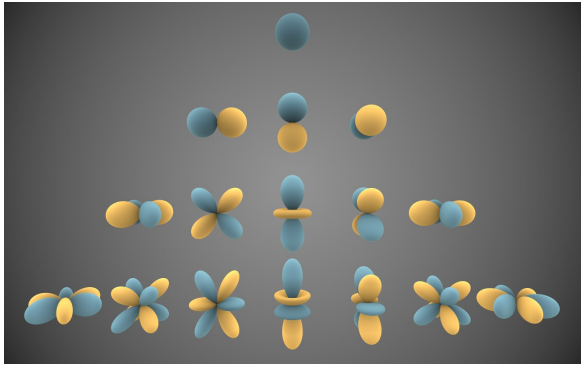
Former Group Members

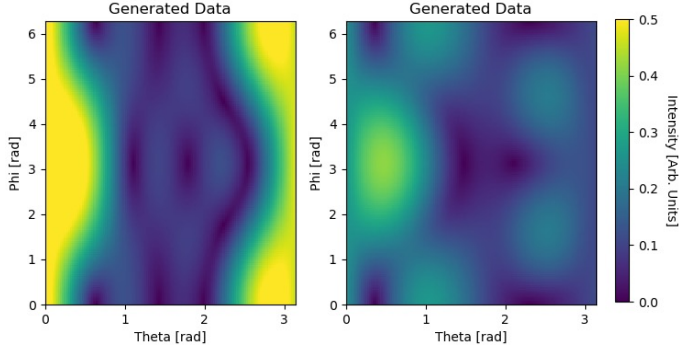
Josh Pond
Stephanie Bramlett
Brandon DeMello

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

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

Tools of the Trade

- 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

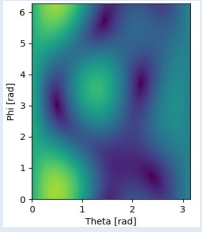
Sample Training Script

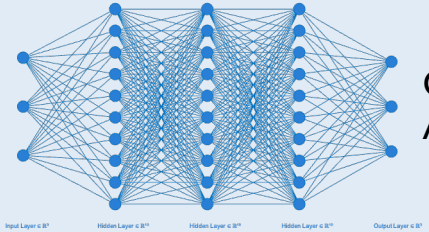
```
test = pd.read_csv("TRAIN/TRAIN.csv")
labels = pd.read_csv("TRAIN/TRAIN_labels.csv")
activation = 'relu'

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)
```

Early Results

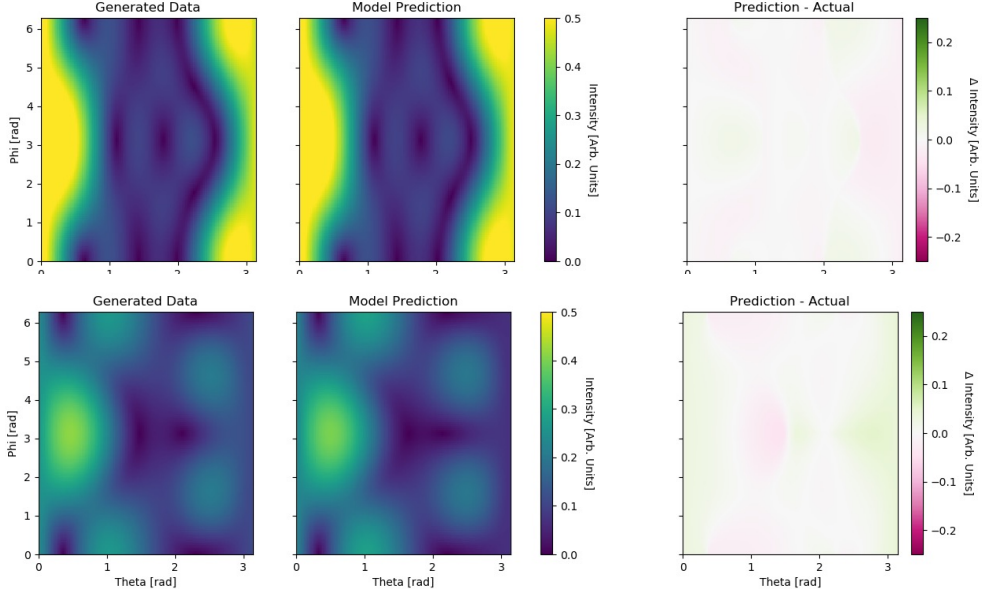
- We compare the intensity function and compare it to the model prediction
- Model Architecture:
  - 128x128 2D histogram as input
  - 9x128 Dense Layers – Relu activation
  - 9 production amplitudes as output
- In order to deal with the vast amounts of data we used generators to generate data for each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

Input Histogram





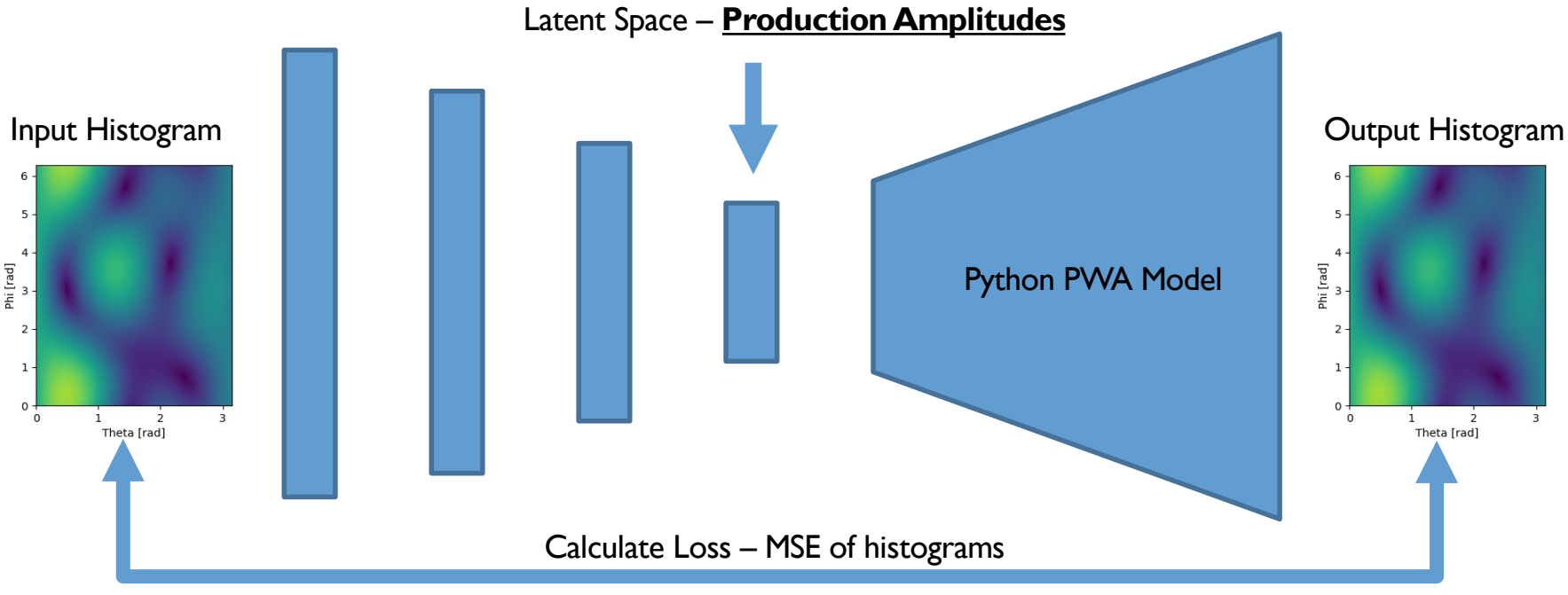
Output - Production Amplitudes



Useful Tools: Generators, Complex Valued Deep Learning

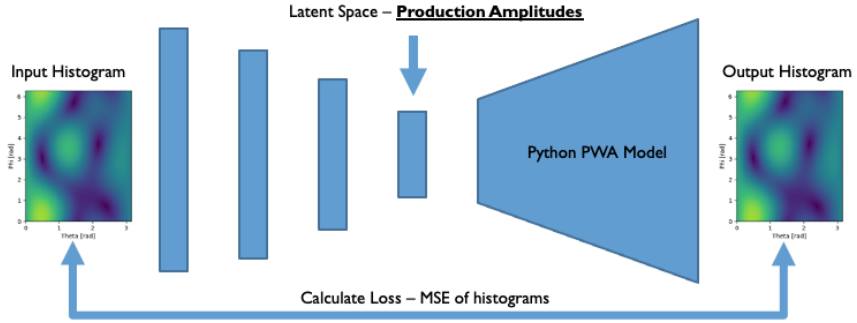
Autoencoder for PWA

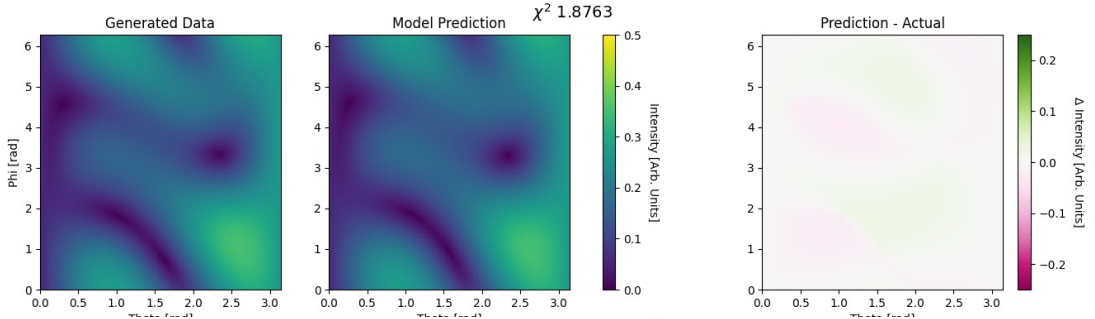
Unsupervised learning!

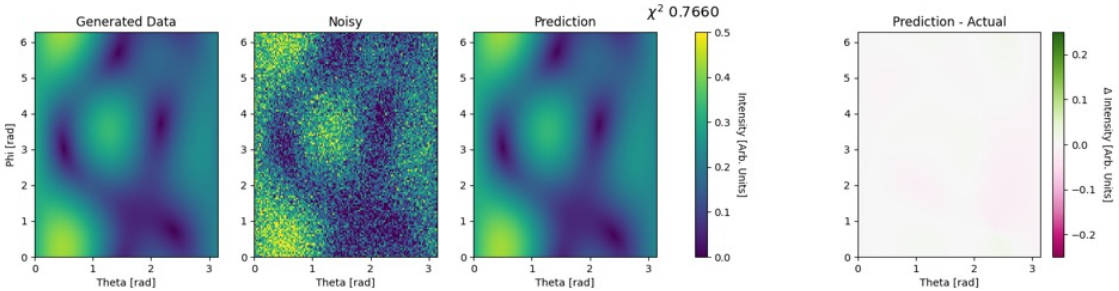


Autoencoders for PyPWA

- 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

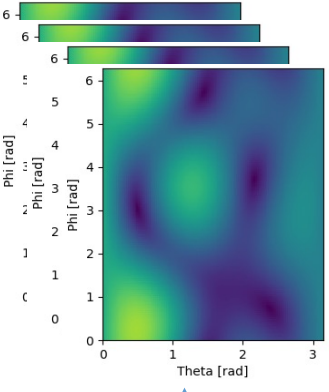






Mass Dependent Autoencoder work for PWA

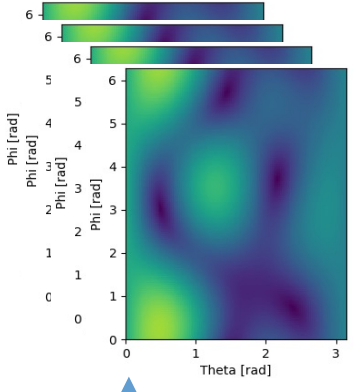
Input ~~Histogram~~
(20x) Histograms



Latent Space – **Production Amplitudes**

Python PWA Model

Output ~~Histogram~~
(20x) Histograms

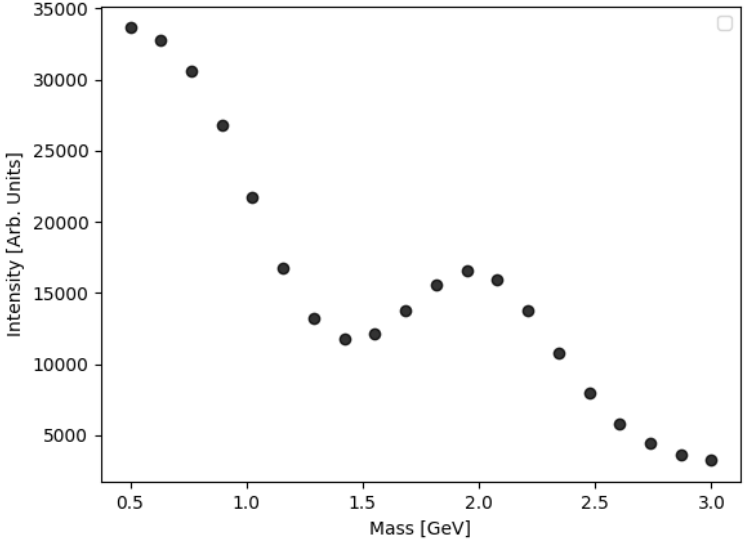


Latent Space p.2 – Resonance Mass and Width

Calculate Loss – MSE of 20 histograms

The Mass-Dependent Generator

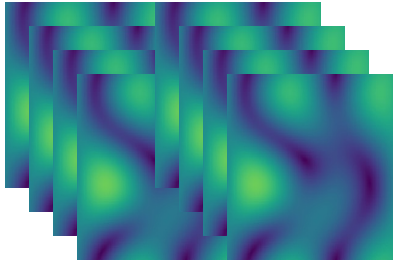
Randomly Generated Event
(Currently One Resonance per Wave)



Production Amplitudes
(Complex)

Breit-Wigner Coefficients
(Mean, Width)

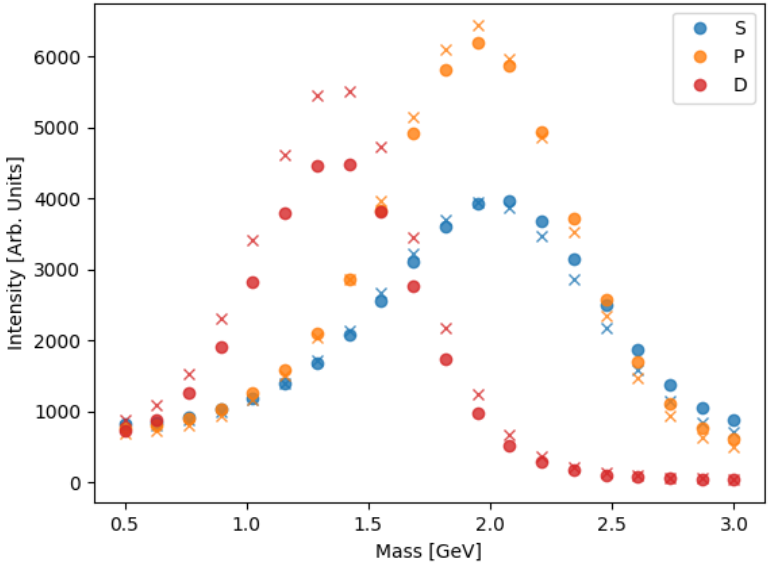
Histogram
Generator



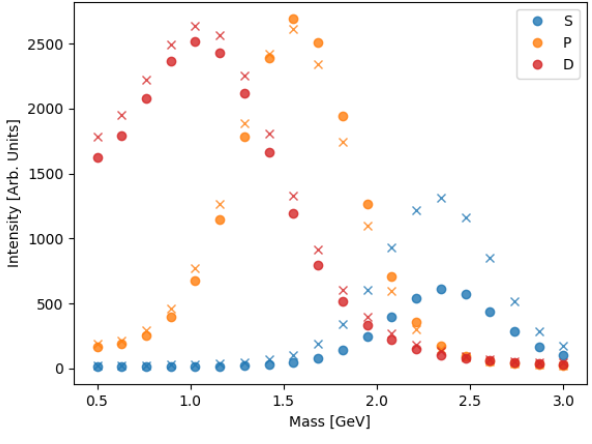
Set of
Histograms
Binned in Mass
(3D-Histogram)

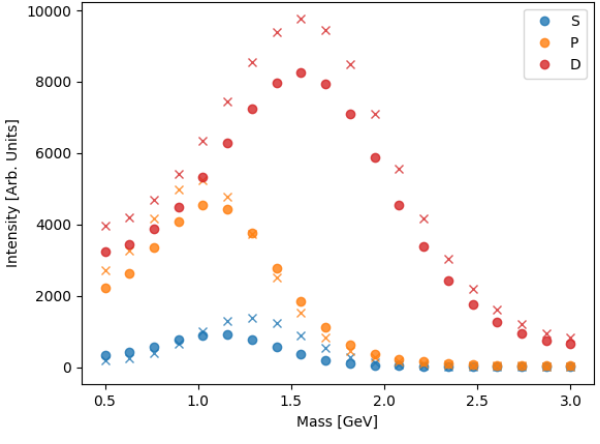
Results

- 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

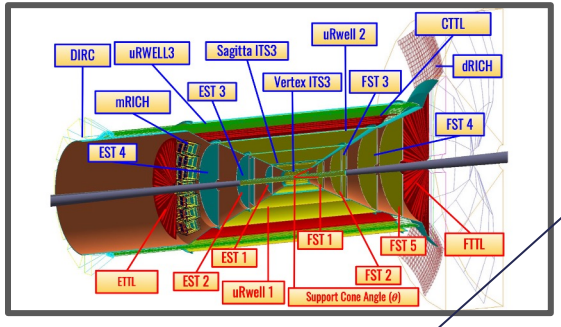
Many thanks to the EPSCI and Data Science group at JLab!
David Lawrence, Thomas Britton, Malachi Schram, Kishansingh Rajput

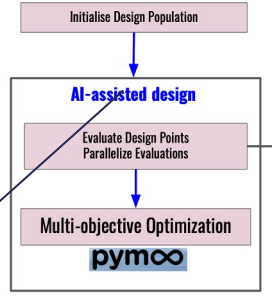
AI-optimized design of the ECCE tracker

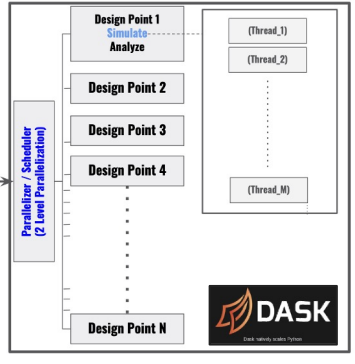
C. Fanelli, Z. Papandreou, K. Suresh

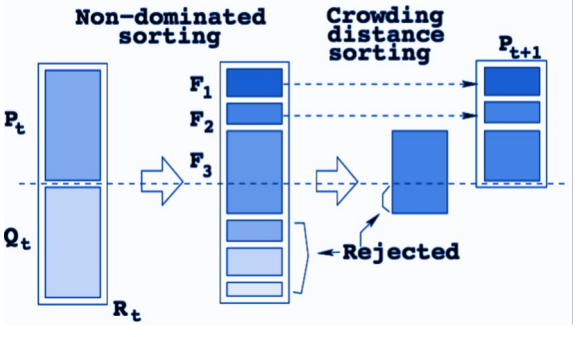
NEW

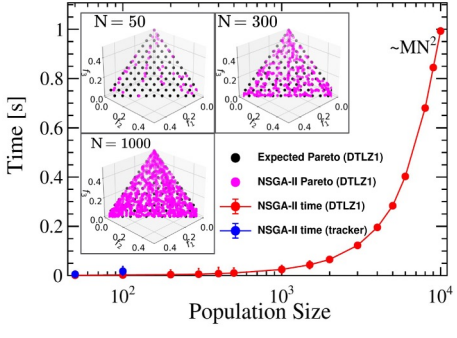
C. Fanelli, Z. Papandreou, K. Suresh et al (ECCE),
AI-assisted Optimization of the ECCE Tracking System at the Electron Ion Collider
https://arxiv.org/pdf/2205.09185.pdf (submitted to NIM-A)











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

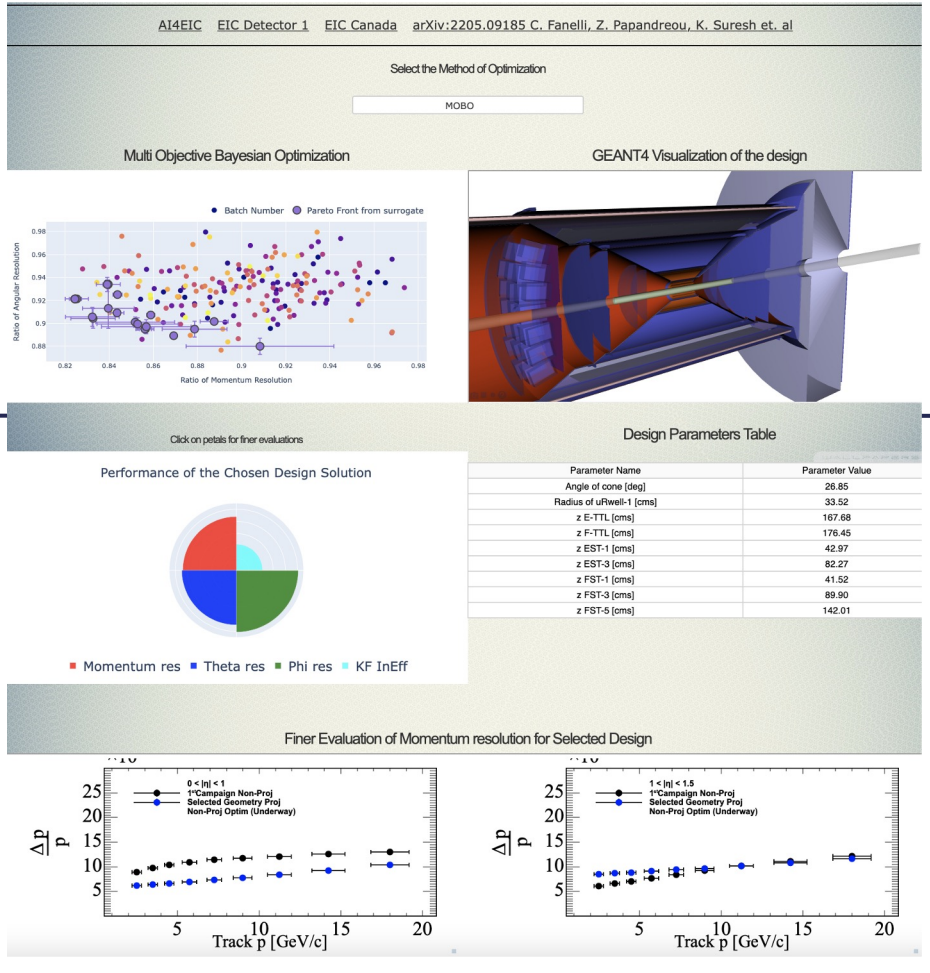
Multiple objectives: different physics analysis

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



Parameter Name	Parameter Value
Angle of cone [deg]	26.85
Radius of uRwell-1 [cms]	33.52
z E-TTL [cms]	167.68
z F-TTL [cms]	176.45
z EST-1 [cms]	42.97
z EST-3 [cms]	82.27
z FST-1 [cms]	41.52
z FST-3 [cms]	89.90
z FST-5 [cms]	142.01

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

https://ai4eicdetopt.pythonanywhere.com

(credits K. Suresh)

https://eic.ai

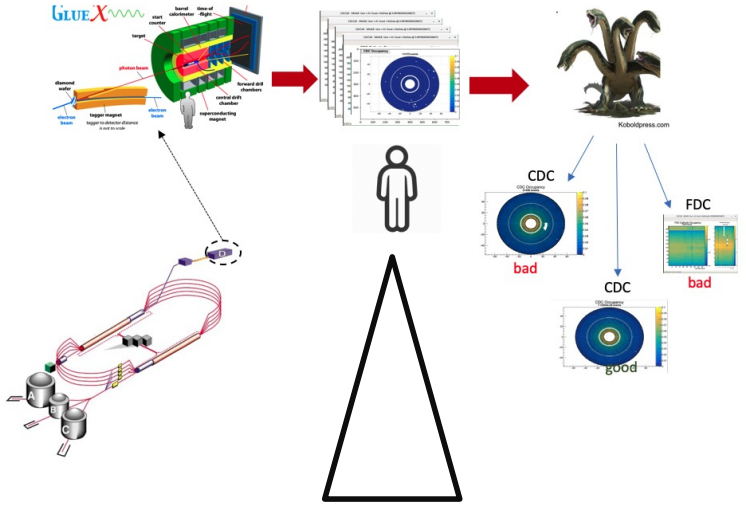
Ongoing Projects – Data Science Group

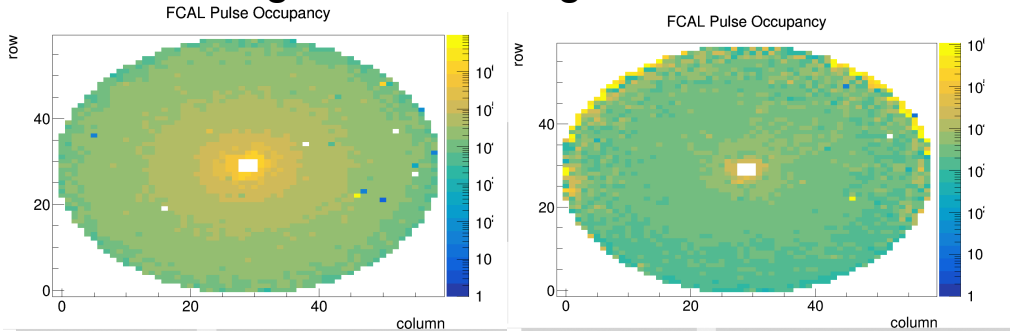
- Hall A SoLID: Kishansingh Rajput
  - Particle ID
- Hall D CCP: Nikhil Kalra
  - Particle ID
- Hall D AIEC calibration: Diana McSpadden
  - Online calibration using Gaussian Processes
- Hydra: Thomas Britton
  - Online Monitoring

Thomas Britton

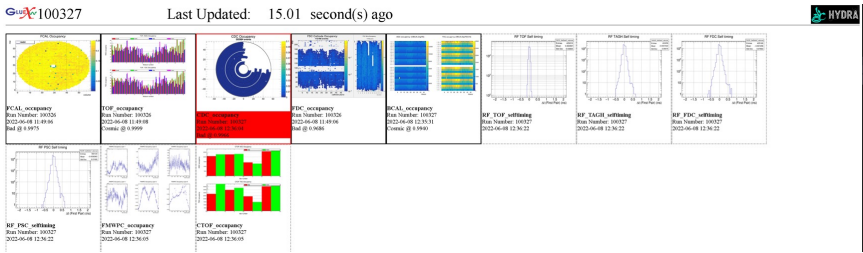
Hydra – Detector Monitoring

- 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

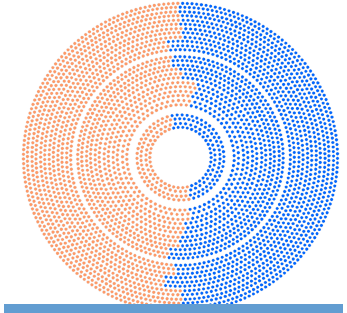


HydraRun also saw the FDC problem, which I probably would have missed inspecting it by eye.

Diana McSpadden

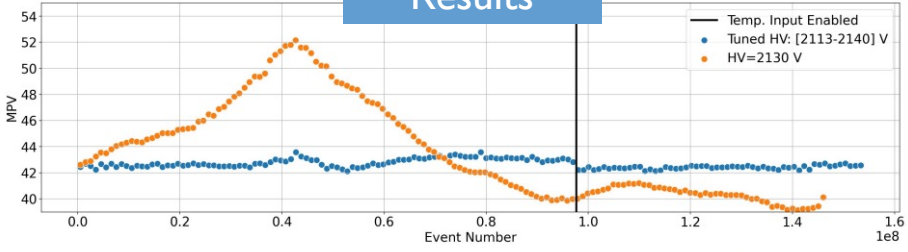
AIEC: Stabilizing Gain in the Central Drift Chamber

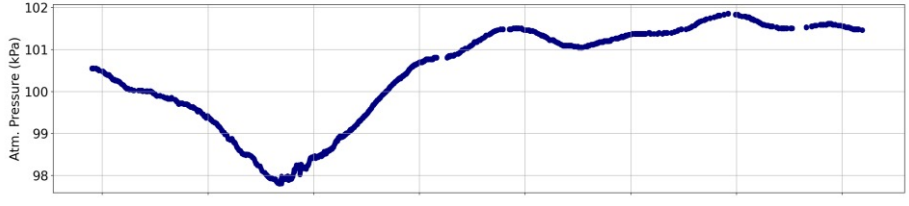
- Accelerate the calibration from months to minutes.
  1. Gain Correction Factor: CDC Voltage Gain calibration
  2. Time to Distance: track fitting calibration
- Calibration is required to provide reliable PID for physics analysis
- Considerations:
  - External environmental conditions (temperature, 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





Summary

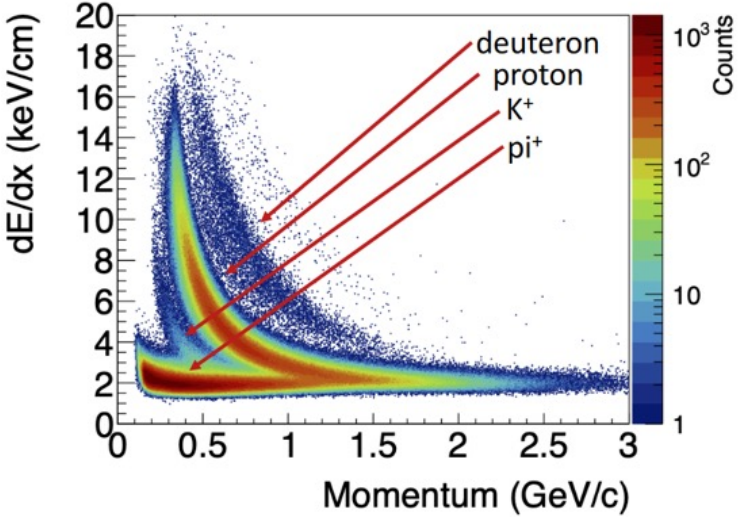
- There are many projects going on at the lab but I was only able to mention a few
  - AI Townhalls have been held in 2020 and 2021 and are a great way to see what the community is doing
- If you are interested in participating keep an eye out for Jlab/EIC hackathons and workshops (https://eic.ai)

Backup

AIEC: Stabilizing Gain in the Central Drift Chamber

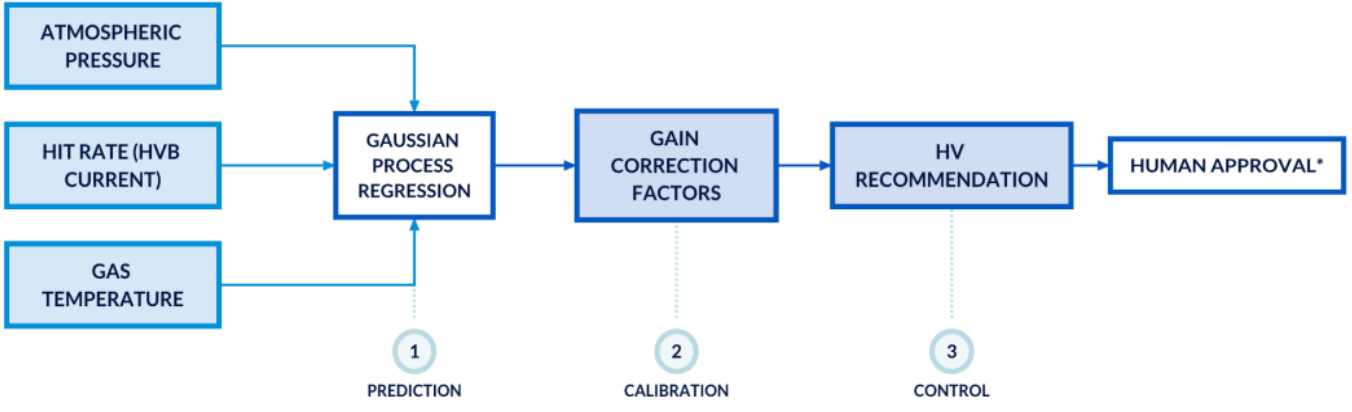
Why?

- Accelerate the calibration from month(s) to minute(s).
  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)



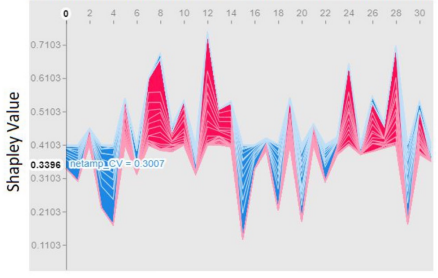
Goals: Near real-time control and calibration

- Traditional Calibration constants are generated per run (~2 hours) due to changing environmental conditions
  - **Requires significant computing time, attention from experts**
- **Goal: Using environmental and experimental data:**
  - **predict** calibration constants during data taking, and
  - **c**

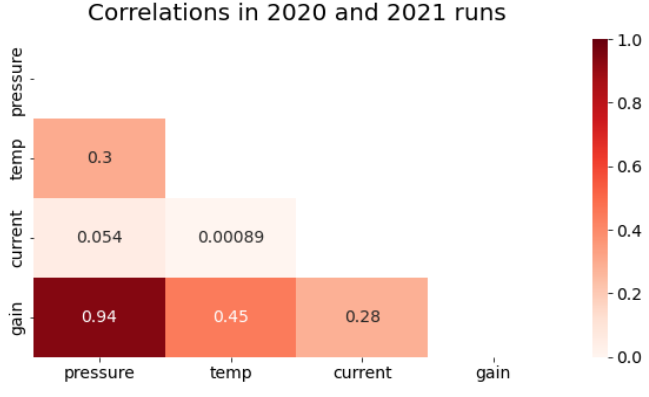


Environmental and Experimental Data

- Began project with 122 available features
- Feature evaluation reduced to 3.
- Selected features are readily available from EPICS system in near real time.
- **Training and Test Data**:
  - 536 runs from 2020 run period
  - 65 runs from 2021 run period
  - "**Pressure balanced**" train and test sets:
    - **Equal** proportions of low, medium, and high atmospheric pressures
    - Training: 480 runs
    - Test: 121 runs

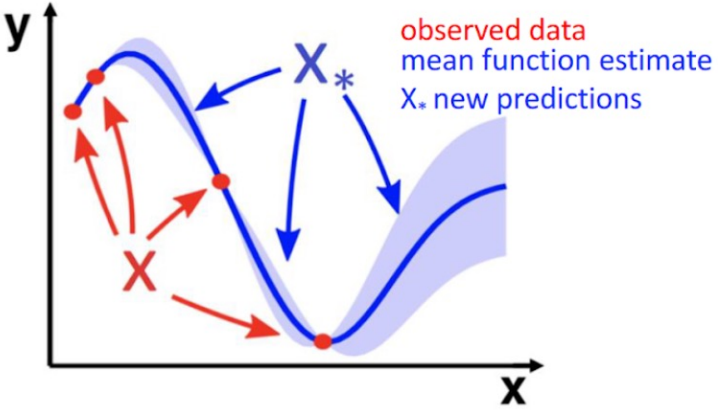


Example Shapely value analysis for feature important evaluation.



Data Science Technique

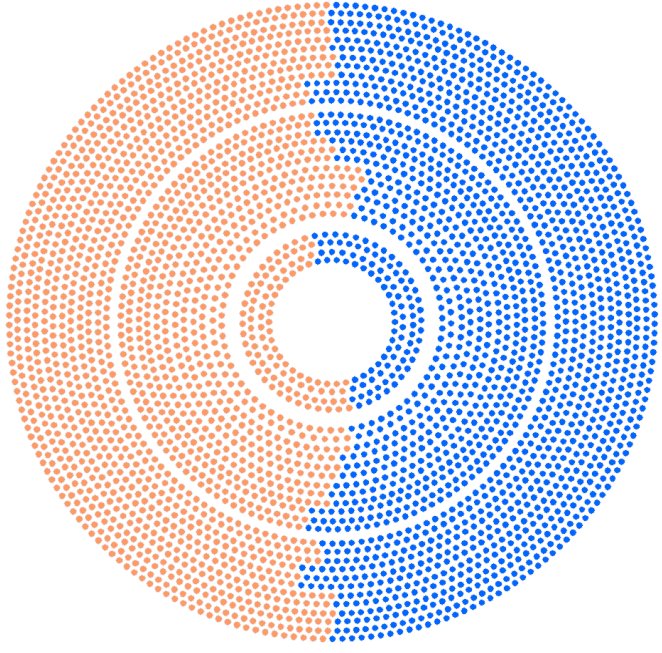
- **Gaussian Process**
  - **3 features** (after feature importance evaluation of over 100 features):
    - Atmospheric pressure
    - Temperature
    - Current drawn from CDC high voltage board.
  - **GP** calculates the PDF over **all admissible functions** that fit the data
  - GP provides standard deviation: used for uncertainty quantification



Cosmic Test: Can we stabilize gain in near real-time?

- Split the CDC into 2 halves
  - One side with **fixed HV** (traditional setup)
  - Let a **Gaussian Process control the other side**

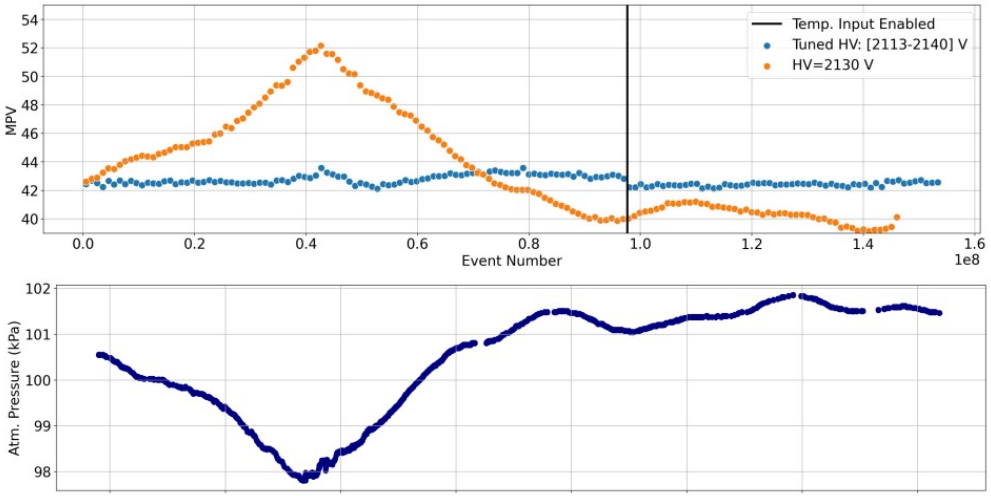
- **Cosmic Experiment:**
  - Update the HV every 5 minutes using GP
  - **Completely autonomous**
  - All information logged in database
  - Remote monitoring of HV settings



Schematic of downstream view of CDC, with straws HV control status indicated.

Results: Yes, we stabilized gain

- Compared MPV of the peak height, ADC units (no magnetic field = no dE/dx) on cosmic muon values during each cosmic run for both sides of the CDC.



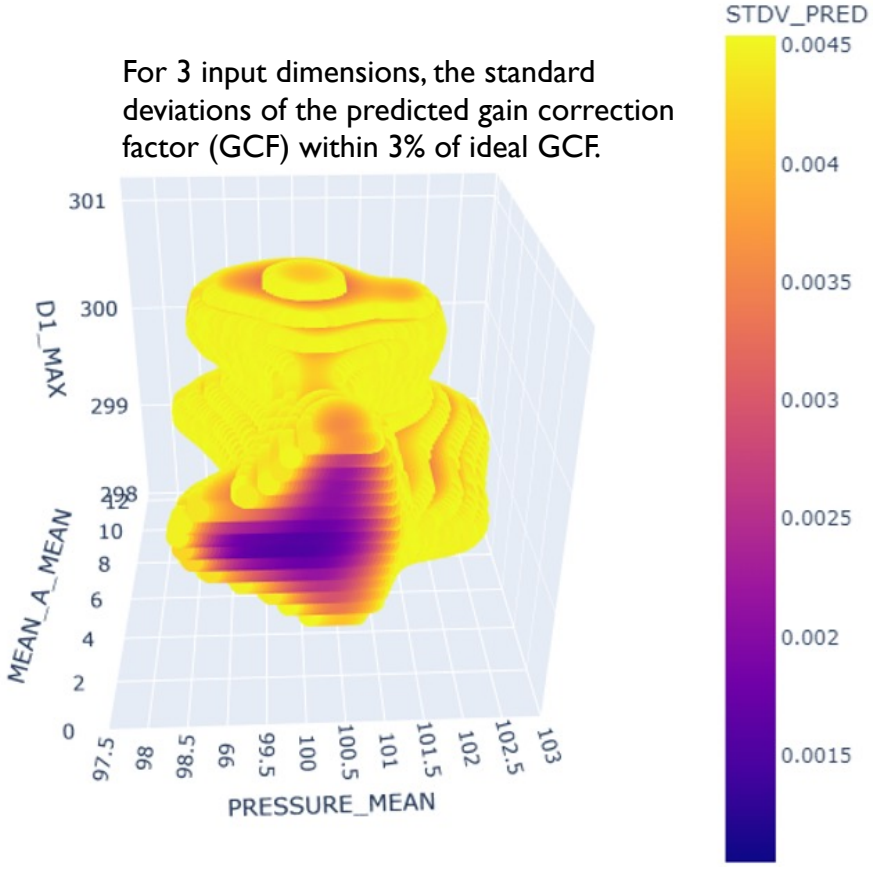
- **Peak heights from Gaussian Process side** of the CDC show dramatic reduction in pressure dependence

Ongoing Work: Using UQ

- Gaussian Process provides uncertainty quantification.
- Do not want to adjust high voltage to an "uncertain" value.
- We only apply a new calibration if the uncertainty is within the 3% of ideal gain correction factor otherwise, we extract to the closest prediction within tolerance

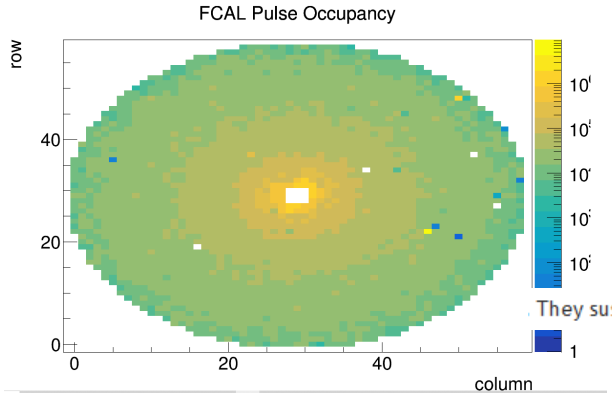
This method will be implemented in the upcoming GlueX CPP run.

For 3 input dimensions, the standard deviations of the predicted gain correction factor (GCF) within 3% of ideal GCF.

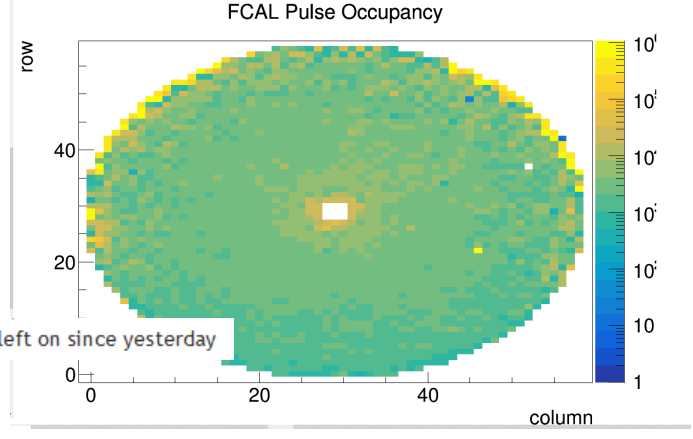


Hydra

- Plenty of detector issues that are not alarmable in the traditional sense but still detectable

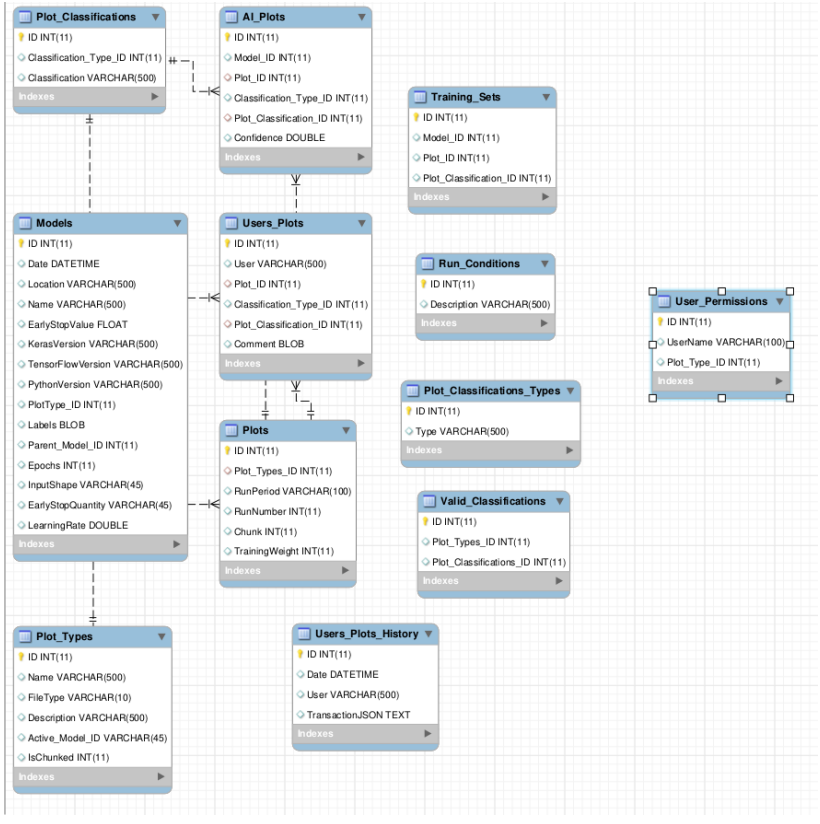


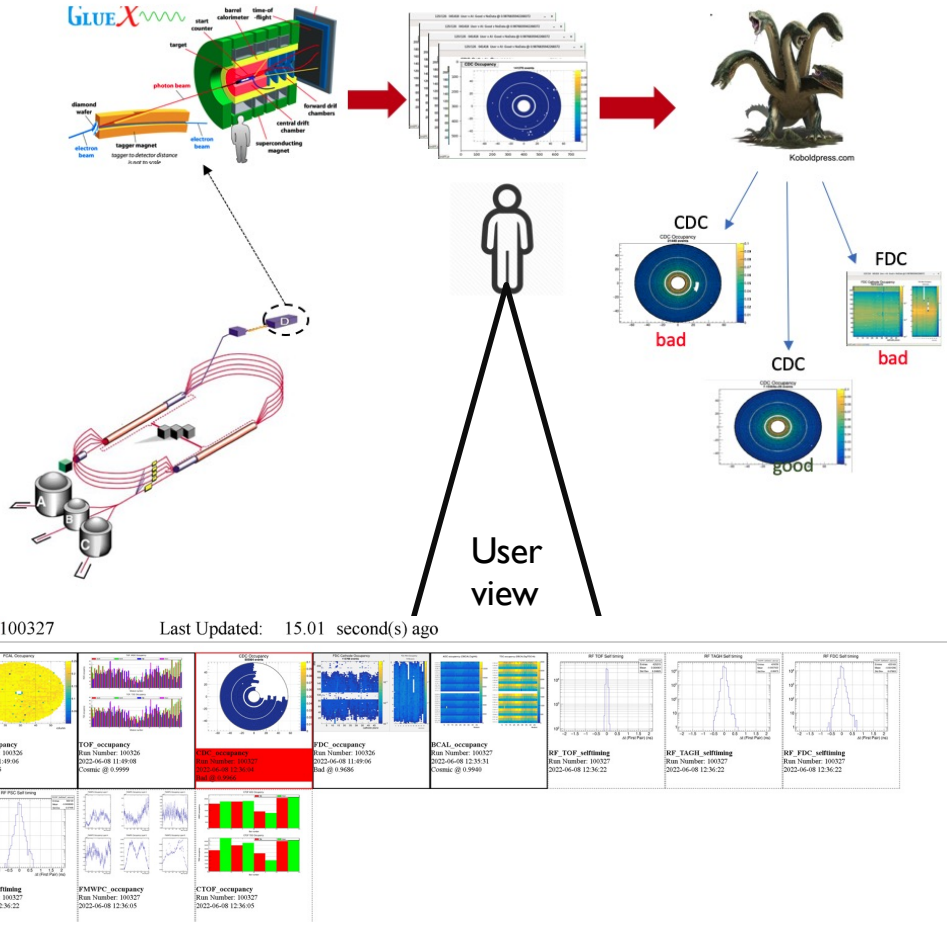
They suspect that the red light in the FCAL is left on since yesterday



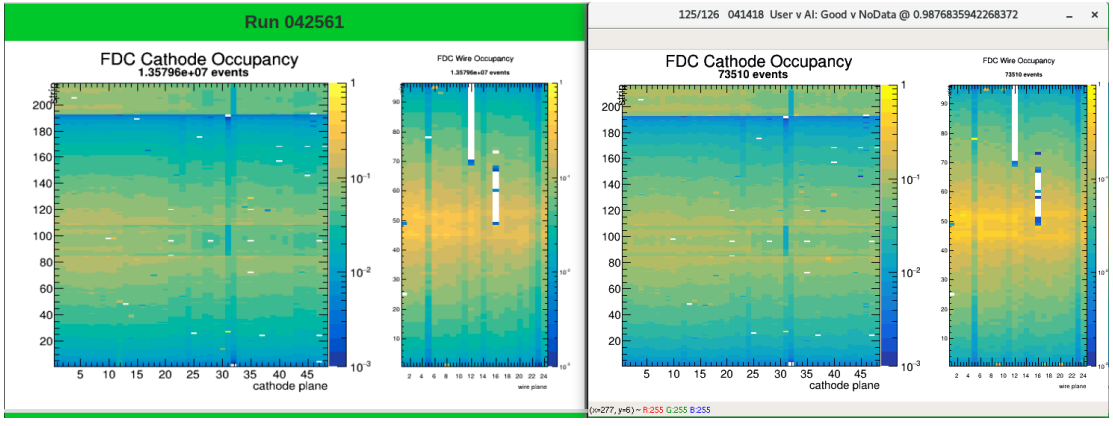
Every run produces an initial >**22 plots**. More thorough monitoring is performed offline and produces >**109 plots**. With a run lasting **~2-3 hours** every day there are **between ~175 and 875** plots to look at every day. Desire AI to augment monitoring.

Hydra


Plot_Classifications
ID INT(11)
Classification_Type_ID INT(11)
Classification VARCHAR(500)
Indexes
AI_Plots
ID INT(11)
Model_ID INT(11)
Plot_ID INT(11)
Classification_Type_ID INT(11)
Plot_Classification_ID INT(11)
Confidence DOUBLE
Indexes
Training_Sets
ID INT(11)
Model_ID INT(11)
Plot_ID INT(11)
Plot_Classification_ID INT(11)
Indexes
Models
ID INT(11)
Date DATETIME
Location VARCHAR(500)
Name VARCHAR(500)
EarlyStopValue FLOAT
KerasVersion VARCHAR(500)
TensorFlowVersion VARCHAR(500)
PythonVersion VARCHAR(500)
PlotType_ID INT(11)
Labels BLOB
Parent_Model_ID INT(11)
Epochs INT(11)
InputShape VARCHAR(45)
EarlyStopQuantity VARCHAR(45)
LearningRate DOUBLE
Indexes
Users_Plots
ID INT(11)
User VARCHAR(500)
Plot_ID INT(11)
Classification_Type_ID INT(11)
Plot_Classification_ID INT(11)
Comment BLOB
Indexes
Run_Conditions
ID INT(11)
Description VARCHAR(500)
Indexes
User_Permissions
ID INT(11)
UserName VARCHAR(100)
Plot_Type_ID INT(11)
Indexes
Plot_Classifications_Types
ID INT(11)
Type VARCHAR(500)
Indexes
Plots
ID INT(11)
Plot_Types_ID INT(11)
RunPeriod VARCHAR(100)
RunNumber INT(11)
Chunk INT(11)
TrainingWeight INT(11)
Indexes
Valid_Classifications
ID INT(11)
Plot_Types_ID INT(11)
Plot_Classifications_ID INT(11)
Indexes
Plot_Types
ID INT(11)
Name VARCHAR(500)
FileType VARCHAR(10)
Description VARCHAR(500)
Active_Model_ID VARCHAR(45)
IsChunked INT(11)
Indexes
Users_Plots_History
ID INT(11)
Date DATETIME
User VARCHAR(500)
TransactionJSON TEXT
Indexes


GlueX
target
photon beam
diamond wafer
tagger magnet
electron beam
forward drift chambers
central drift chamber
superconducting magnet
Koboldpress.com
CDC
bad
FDC
bad
CDC
good
User view
GlueX100327 Last Updated: 15.01 second(s) ago
HYDRA

Hydra (an anecdote)



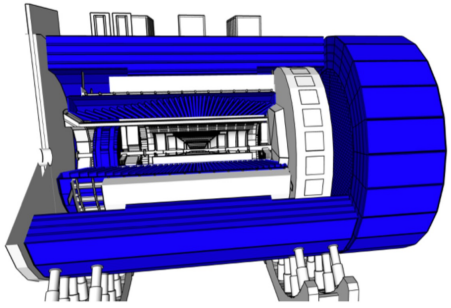
•The labeler was instructed by the detector expert to label any plot containing **fewer than 100k events as "NoData"**. This is one example of several in which the labeler labeled as "Good" and the **A.I. predicted "NoData"**...the true label given the number of events

Real world result: HydraRun also saw the FDC problem, which I probably would have missed inspecting it by eye. There were no alarms

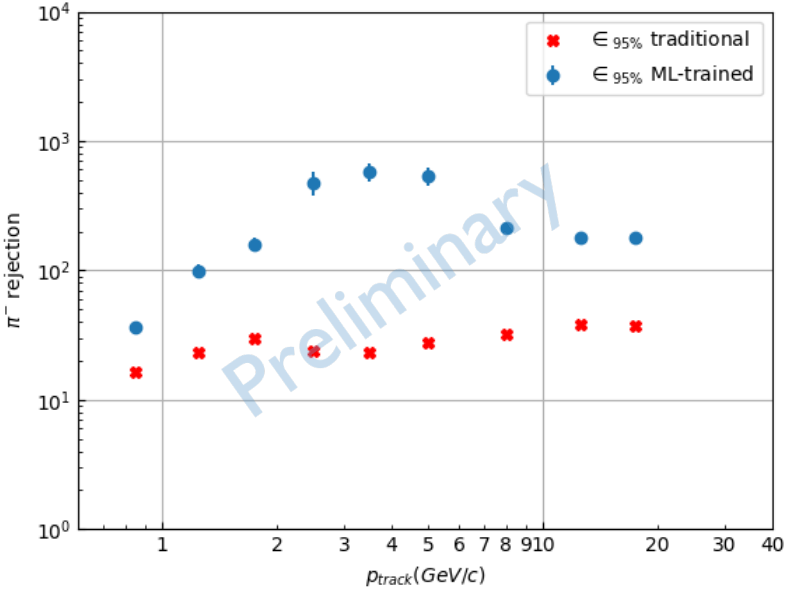
Detector expert

BECAL PID Study

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

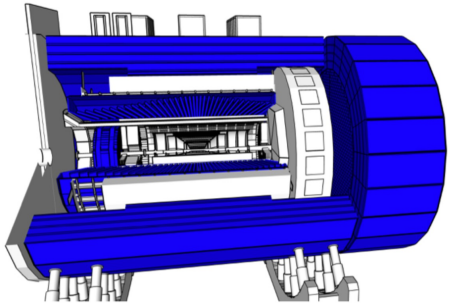


- Higher pion rejection compared to conventional methods when considering high electron efficiency (~95%)
- Work is in progress (started on thanksgiving)
- Interface with ECCE software: reco-track, track projection, 7x7 calorimeter towers near track (track-based clustering by AI) [Link to details]
-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

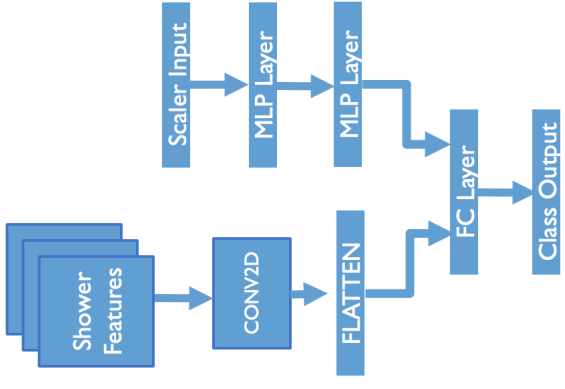


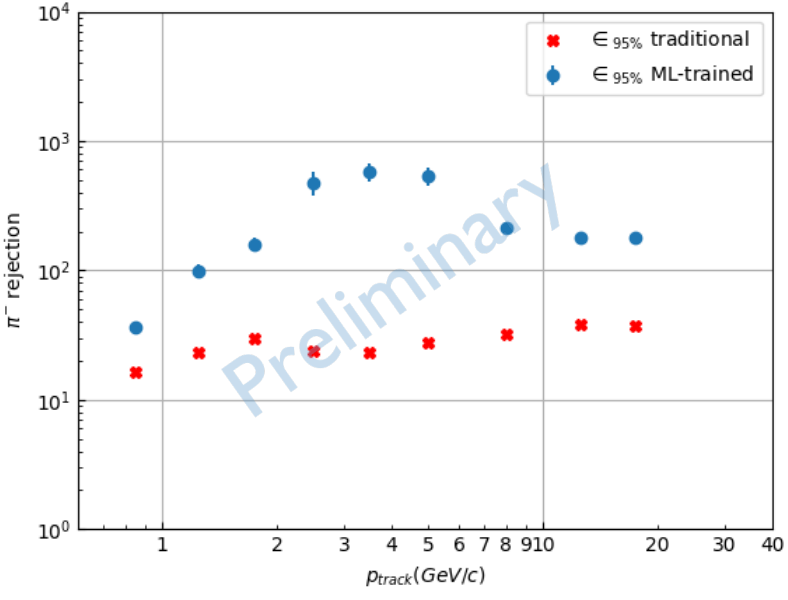
BECAL PID Study

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