ARTIFICIAL INTELLIGENCE IN CLAS12 ANALYSES

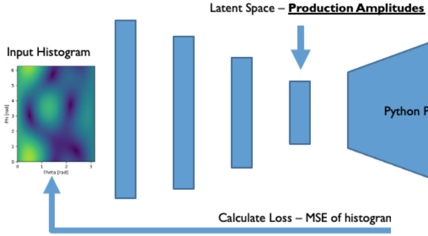
William Phelps

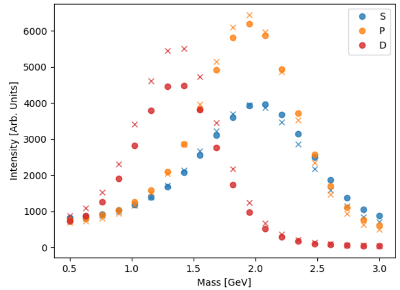
Christopher Newport University/Jefferson Lab

Roadmap

- Deep Learning - Partial Wave Analysis (PyPWA)
  - Uncertainty Quantification
  - Wave Selection
- AI Analysis Tools in CLAS12
  - 9 Projects with many institutions working on them
- **Disclaimer:** this talk will assume some level of familiarity with ML/AI concepts as it is proliferating very quickly throughout physics and the rest of society. For example, ChatGPT:
  - Plot beta vs momentum for pions, kaons, and protons
  - Write a letter of recommendation letter *as an example*
  - Create a LaTeX table for the specifications of a dual cascade lake server with 3 Nvidia V100's
  - 4-5 Sentences about a Cherenkov counter PMT characterization project

PyPWA





AI Analysis Tools in CLAS12

- Reaction ID
- Data Preservation
- Fast MC
- High Level Analysis Tools
- Particle ID

t time

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 4.0 with PyTorch library has been released in October 2022
- Efficient amplitude analysis framework including multithreading, CUDA support, and PyTorch libraries
- Optimizers include Minuit, Nestle, MCMC (or add your own!)
- NIM Paper almost ready to be submitted (Maybe this month!)

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

Group Members

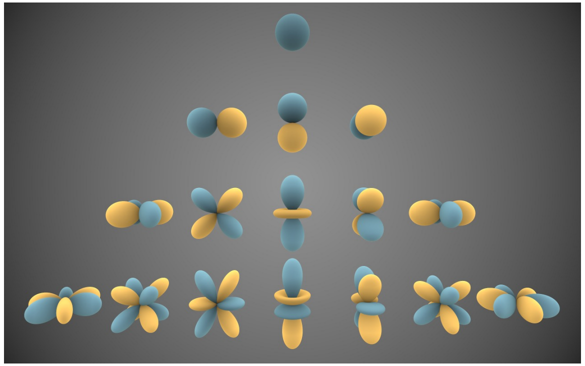
Carlos Salgado (NSU/Jlab)
Mark Jones (NSU)
Peter Hurck (Glasgow)
William Phelps (CNU/Jlab)
Andru Quiroga (CNU)
Nathan Kolling (CNU)
Ryan Brunk (CNU)
Rafael Diaz-Cruz (CNU)
Brian Rotich (NSU)

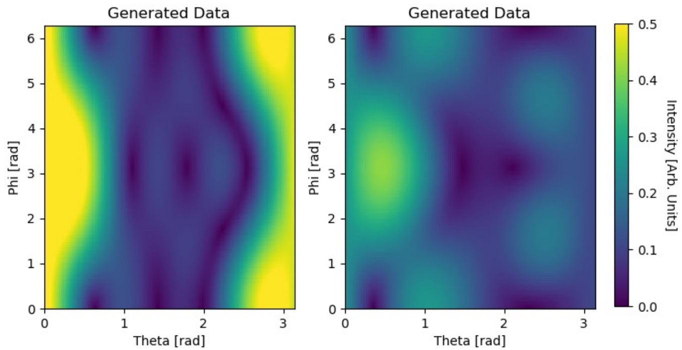
Former Group Members

Josh Pond
Stephanie Bramlett
Brandon DeMello
Michael Harris (NSU)
Bruna Goncalves (NSU)

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")





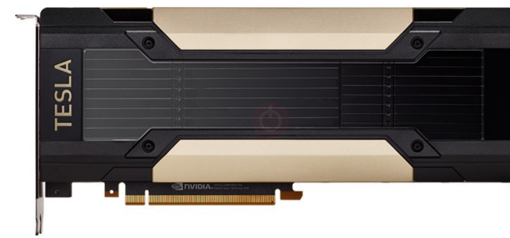
$$I(\Omega) = \sum_k \sum_{\epsilon_R} \sum_{l,|m|,l',|m'|} {}^{\epsilon_R}Y_l^{|m|}(\Omega)\ {}^{\epsilon_R}V_{l,|m|}^{k}\ {}^{\epsilon_R}V_{l',|m'|}^{k*}\ {}^{\epsilon_R}Y_{l'}^{|m'|*}(\Omega)$$

Production Amplitudes

Decay Amplitudes

Tools of the Trade

- Python 3.9 – Anaconda
  - Keras/TensorFlow – NN Libraries
  - Pandas/Numpy – Data Handling
  - Matplotlib – Visualization
  - Uproot – Native Python ROOT Library (J. Pivarski)
  - Optuna – Hyperparameter optimization library
- Institutional GPU nodes or those through Jefferson Lab
  - Either through Jupyterhub or interactively using slurm to request a node
  - Several institutions with Nvidia V100 and A100 Cards (NSU/JLAB)
  - Several machines with 4 NvidiaTitan RTX GPUs and some with 14 Nvidia T4 GPUs



Example Training Script

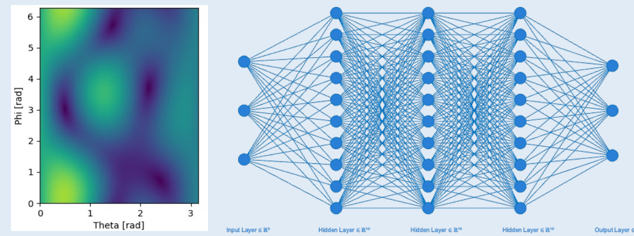
```
test = pd.read_csv("TRAIN/TRAIN.csv")
labels = pd.read_csv("TRAIN/TRAIN_labels.csv")
activation = 'relu'

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)
```

MLP Results

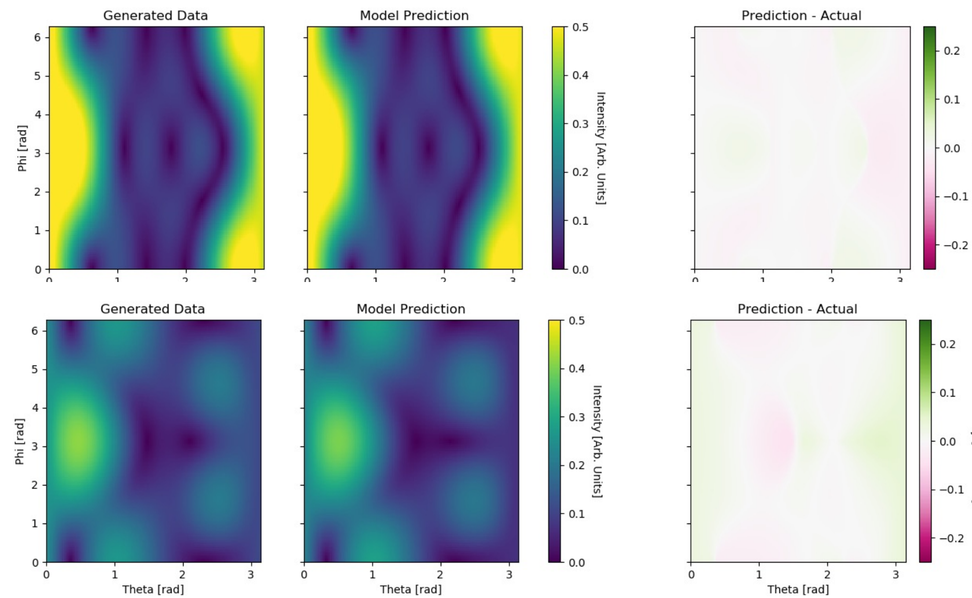
TensorFlow Keras python

- 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



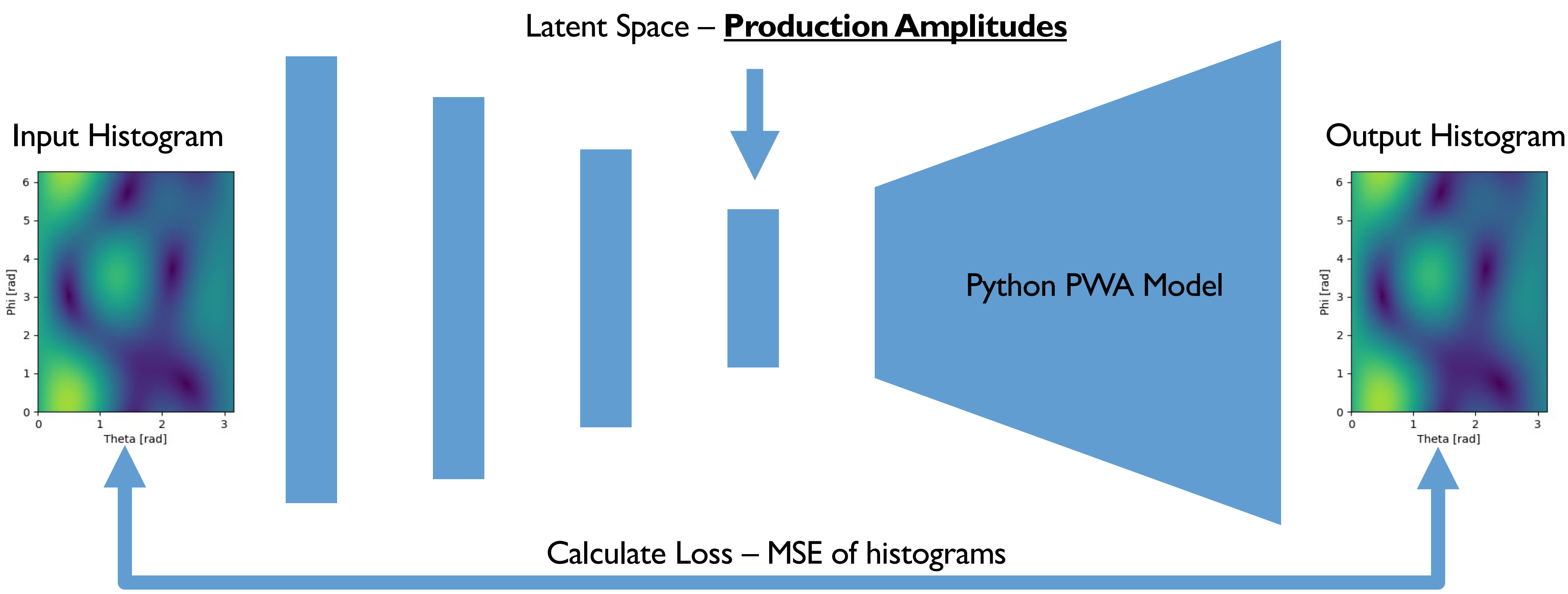
Output - Production Amplitudes



Useful Tools: Generators, Complex Valued Deep Learning

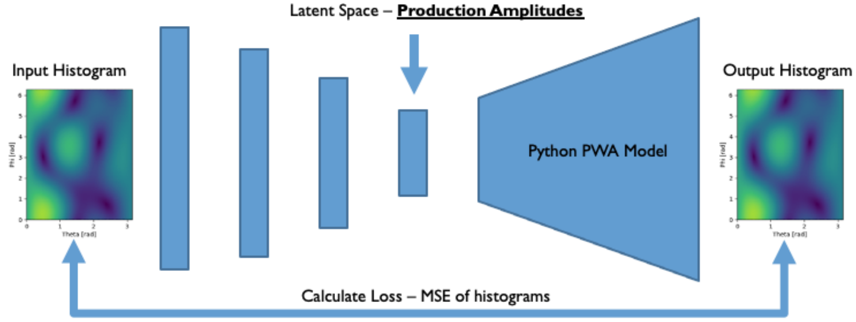
Autoencoder for PWA

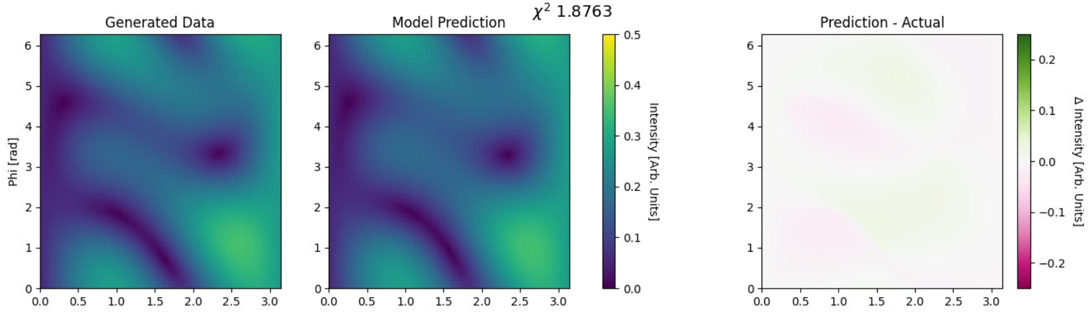
Unsupervised learning!

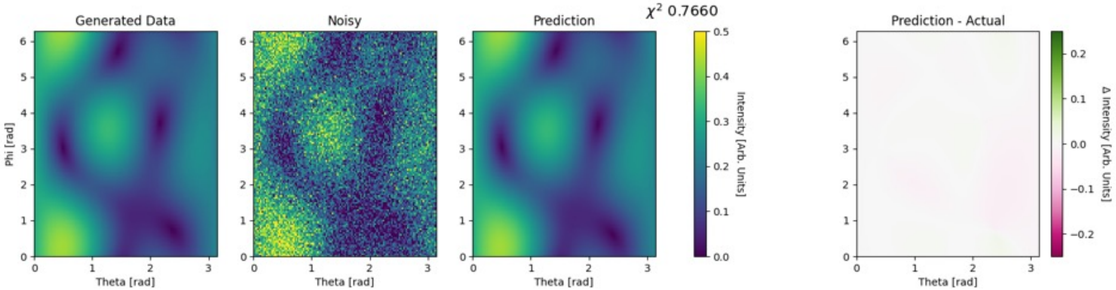


Autoencoders for PyPWA

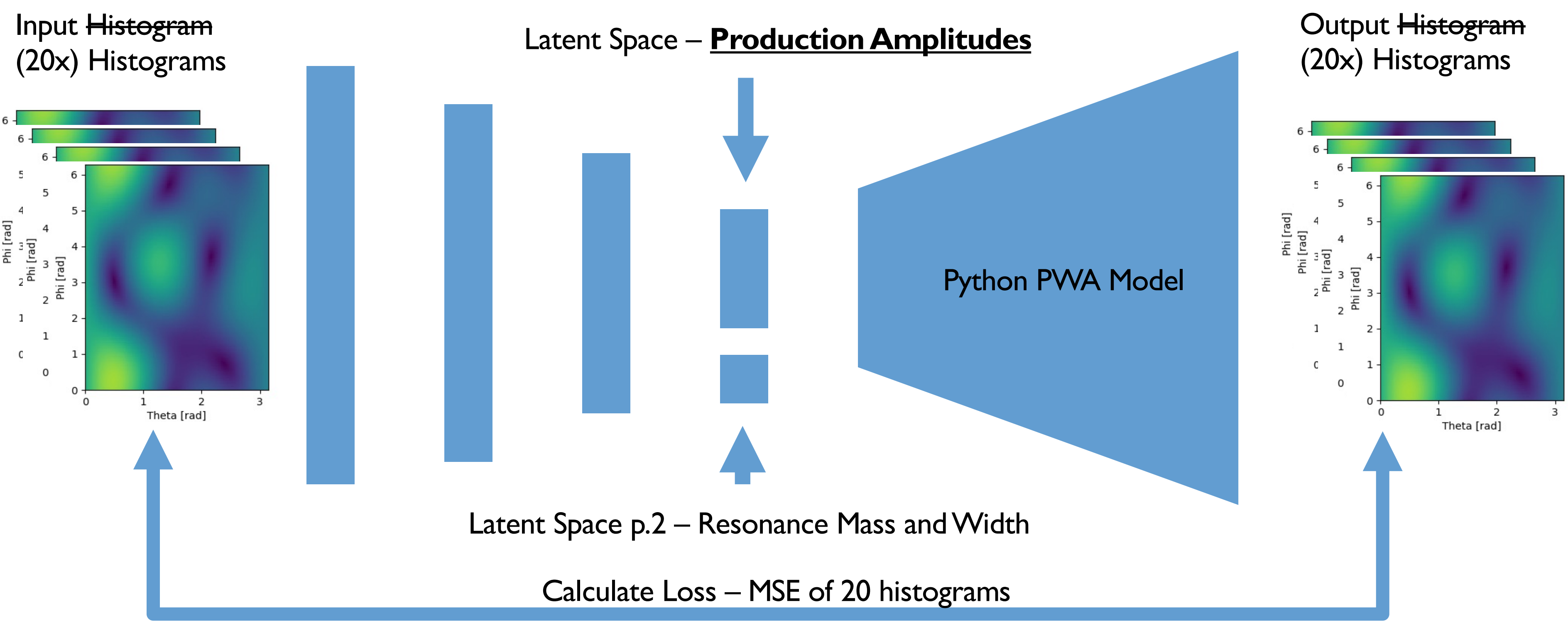
- Encoder portion is a standard MLP, but without labels!
- Decoder is a PyPWA model that takes in production amplitudes and produces a histogram
- Autoencoders *dramatically* improved the accuracy!
- Even works well for noisy data





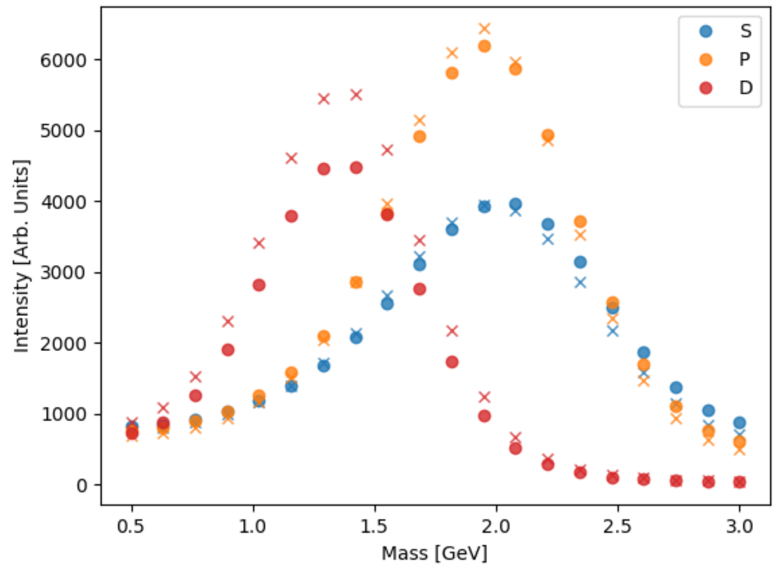


Mass Dependent Autoencoder work for PWA

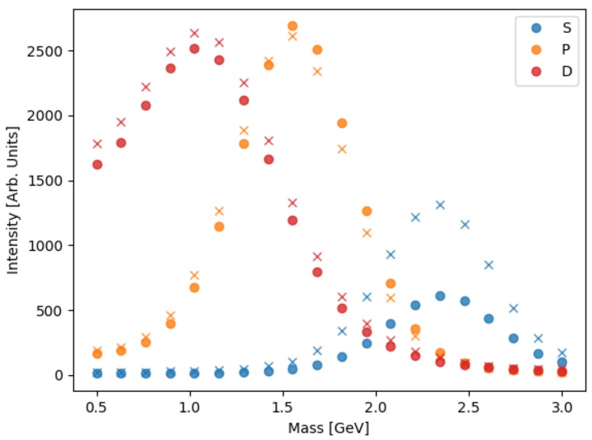


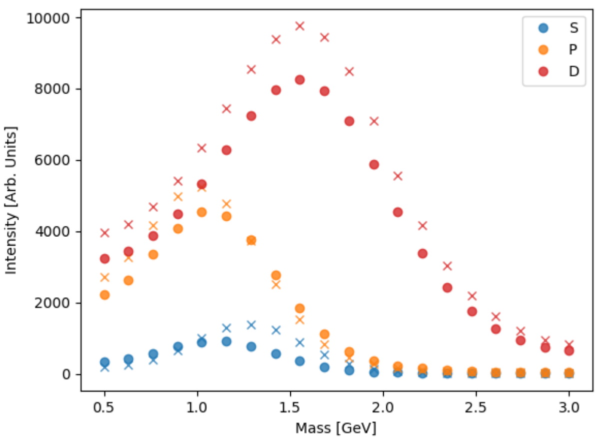
Results

- 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



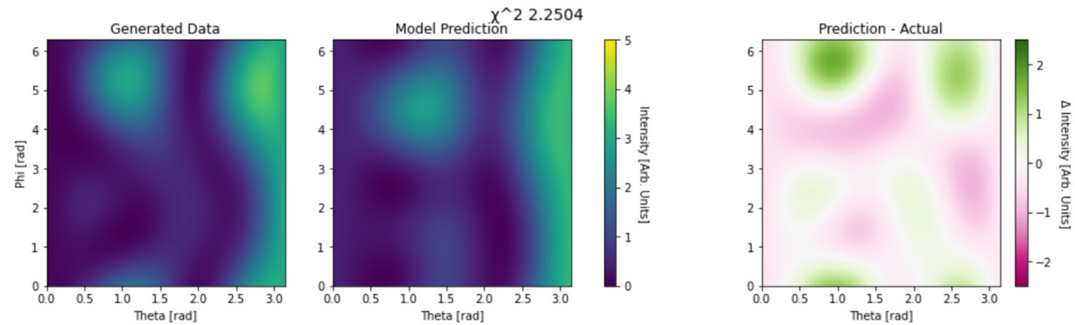
Crosses – Generated
Circles – Inference

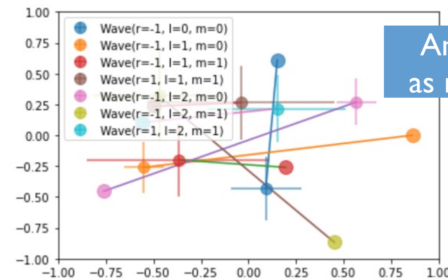




Uncertainty Quantification - VAE

- For uncertainty quantification we are using Variational Autoencoders (VAE) with some success
- Traditional (hybrid) autoencoder performs better for now
- Future work could involve some constraints to resolve ambiguities and allow better fits



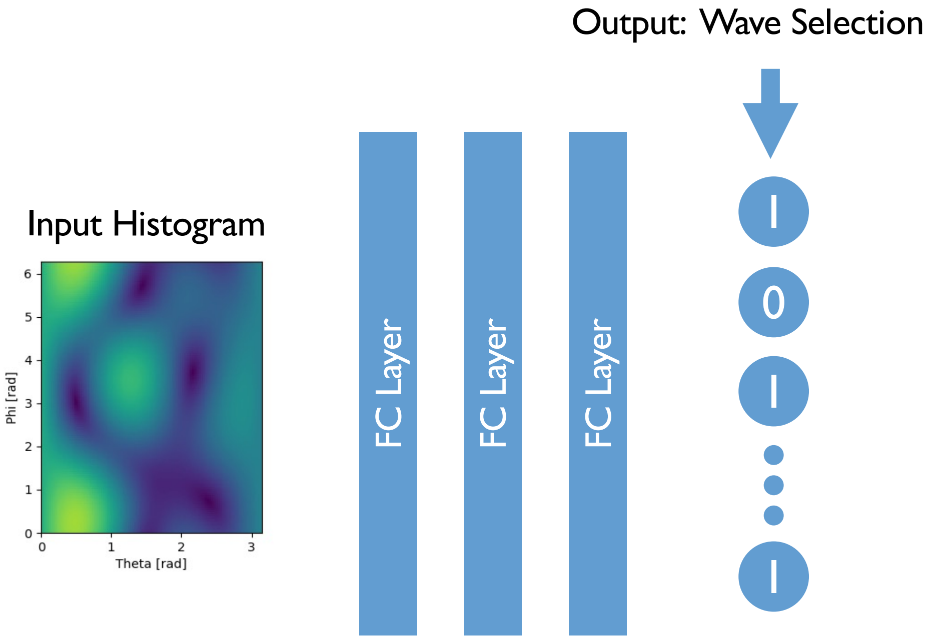


Ambiguities appear as rotations/sign flips

	X_0	X_1	X_2	X_3	X_4	X_5	X_6	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
True	0.15	0.86	0.19	-0.50	-0.76	0.45	-0.56	0.61	0.00	-0.26	0.24	-0.45	-0.87	0.10
Pred	0.09	-0.55	-0.37	-0.04	0.57	-0.47	0.15	-0.43	-0.26	-0.20	0.26	0.27	0.32	0.22

Wave Selection DNN

- One of the problems that is regularly seen in PWA is choosing the right waves to use in your fit
- We simplified the regression problem we have posed in earlier slides to create a tool that could be used to select which waves are present
- Multi-label classification
- May be used as a part of an ensemble



Preliminary results:
79% accuracy in selecting the right set of waves (Lmax=2)
96.3% wave/"digit"-wise accuracy

CLAS12 AI Projects in Analysis

- In this talk I will *briefly* summarize several AI Analysis Tools actively being used in the CLAS Collaboration
- These tools/analyses are at various stages of maturity. Some have been used in published PRLs and some are just starting.
- Also, this is not a complete list, by far! I apologize if I have missed anyone.

AI Analysis Tools in CLAS12

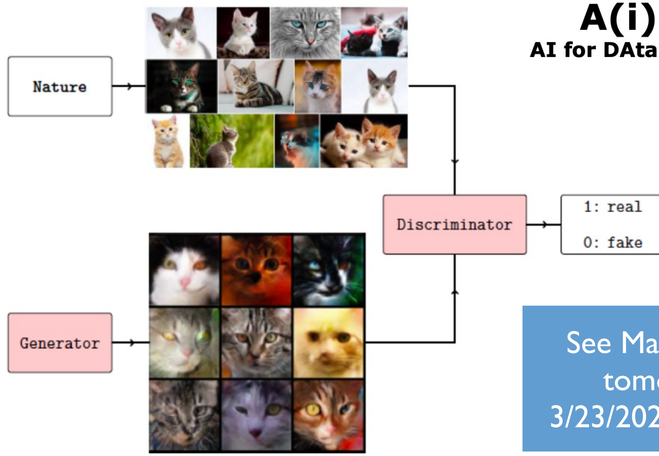
- Reaction ID
- Data Preservation
- Fast MC
- High Level Analysis Tools
- Particle ID

AI for Data Preservation

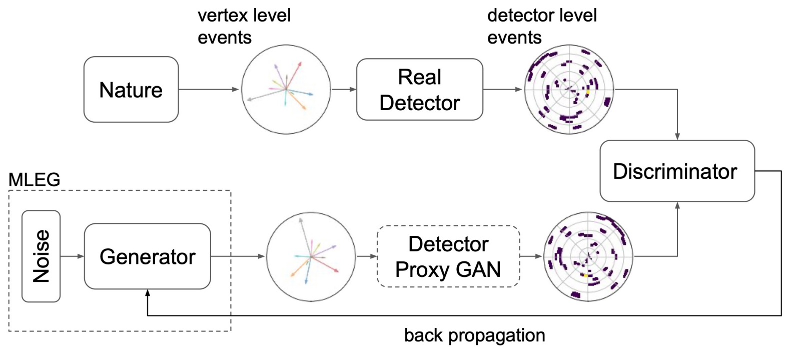
M. Battaglieri and A(i)DAPT

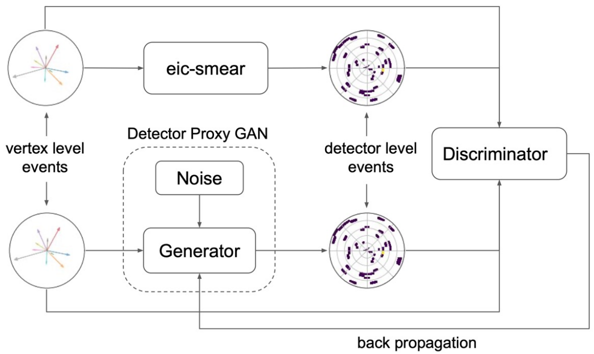
A(i)DAPT
AI for DAta PreservaTion

- Collaboration between theory and experimentalists
- One of the goals is to preserve physics using an event generator using GANs instead of the more traditional cross section measurements
- This is only one of their projects so please visit the wiki link below for more information or prior talks



See Marco's talk tomorrow 3/23/2023 at 09:00





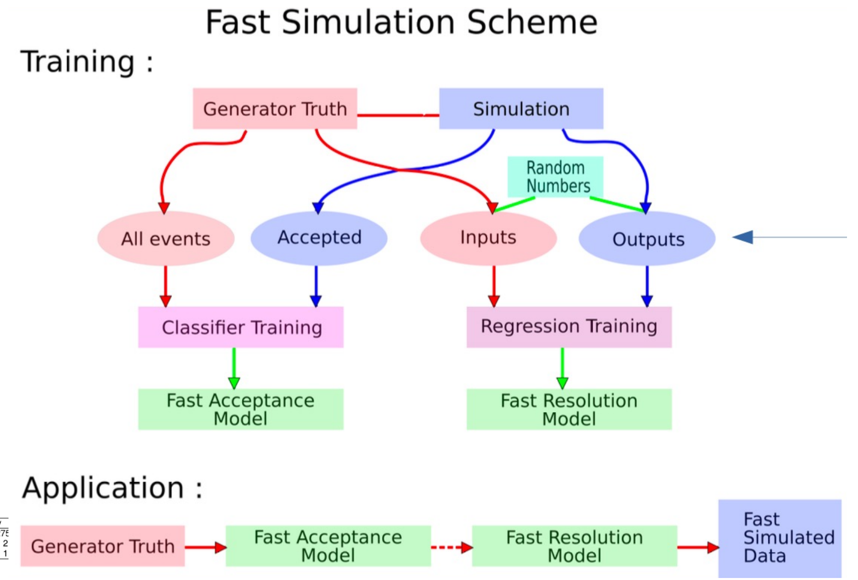
https://clasweb.jlab.org/wiki/index.php/A(I)DAPT_-_AI_for_Data_Analysis_and_PresevaTion

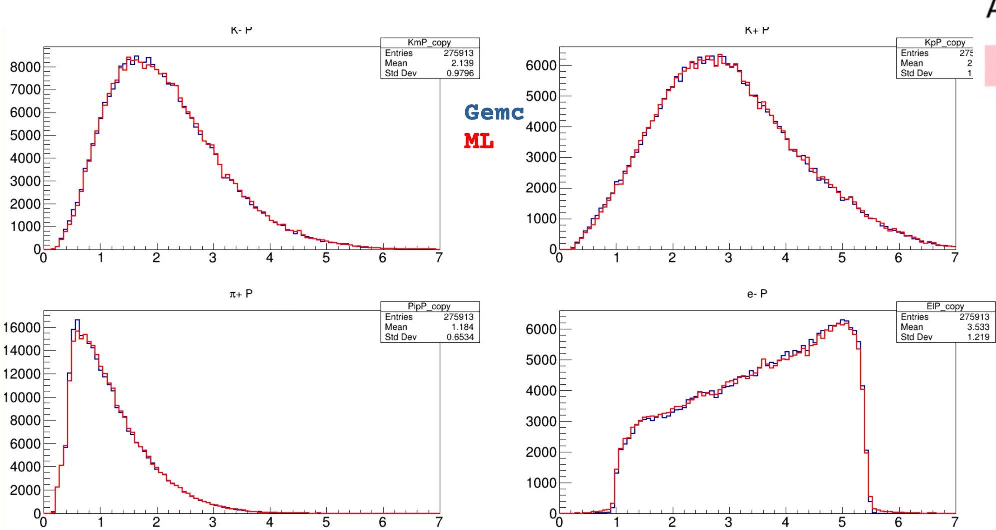
Detector Proxy GAN

D. Glazier (Glasgow)

AI Fast Monte Carlo

- Fast simulation using AI methods
- Used two models, one for detector acceptance one for momentum smearing



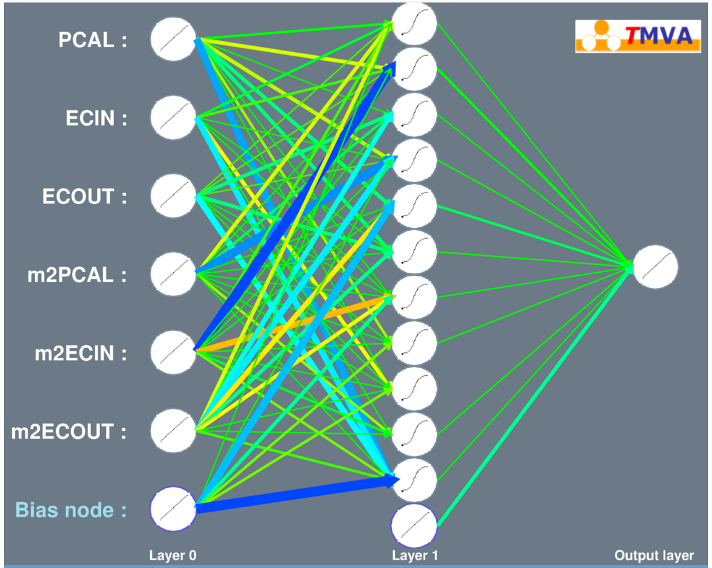


- MLPs and decision trees were used
- Momentum distributions shown for one reaction but all distributions match well after reweighting!

Lepton/Pion Separation for TCS

P. Chatagnon (Jlab/IJCLab)

- Accurate pion/lepton separation needed for TCS reaction
- HTCC cannot distinguish e^+ and π^+ above 4.5 GeV/c so
- MLP model used after comparing with other ML methods
- **Resulting in 50% S/B ratio dropping to 5% S/B when $P_{positron}$ > 4.5 GeV**



MLP Classifier Model

P. Chatagnon et al. Phys. Rev. Lett. 127, 262501 (2021)

See talk 3/24/2023 at 11:24

IJCLab Irène Joliot-Curie Laboratoire de Physique des 2 Infinis

Positron: electromagnetic shower   Pion: Minimum Ionizing Particle (MIP)

$$SF_{\mathrm{EC\ Layer}} = \frac{E_{dep}(\mathrm{EC\ Layer})}{P} \qquad M_2 = \frac{1}{3}\sum_{U,V,W} \frac{\sum_{\mathrm{strip}} (x-D)^2 \cdot \ln(E)}{\sum_{\mathrm{strip}} \ln(E)} \rightarrow \textbf{6 variables}$$

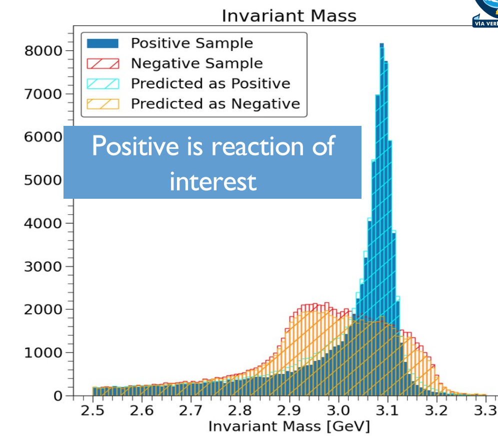
Reaction ID for J/Psi

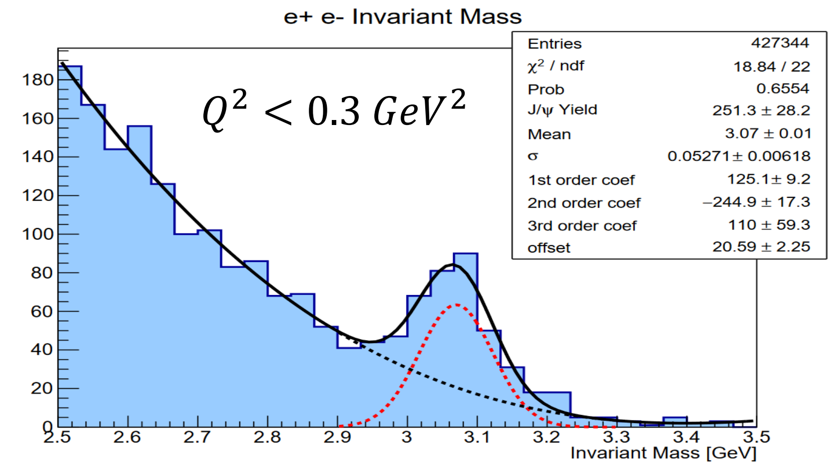
R. Tyson(Glasgow), B. McKinnon(Glasgow), and G. Gavalian (JLab)

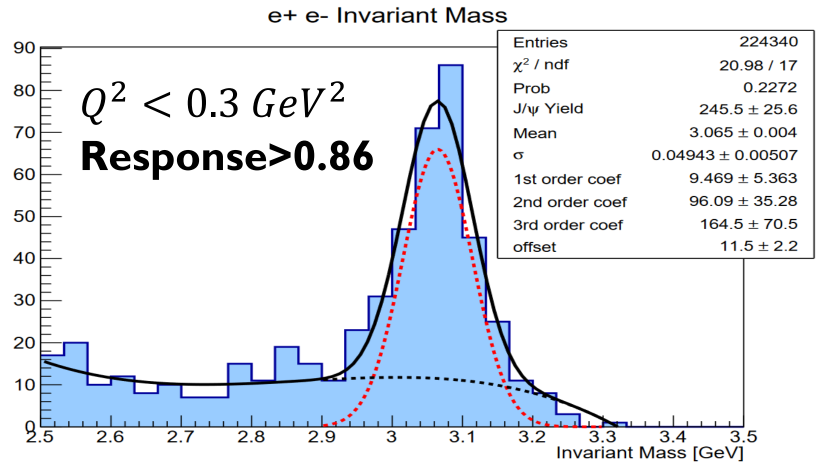
University of Glasgow

See talk at 3/24/2023 at 11:50

- Using reconstructed 4-Vectors inclusive and exclusive reactions were improved by using MLPs to filter out background events
- Preliminary work not used in cross section calculation



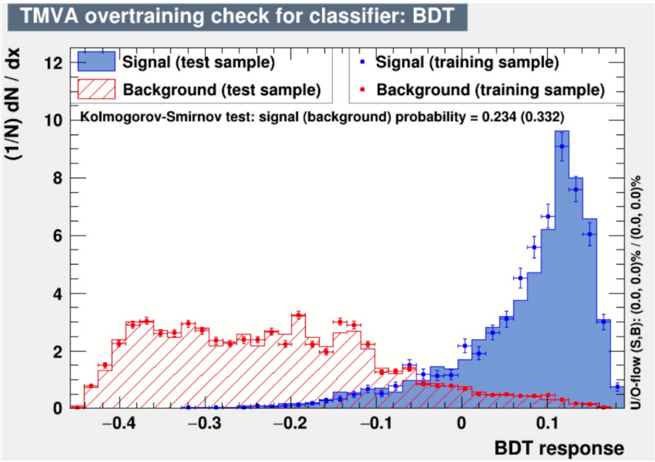


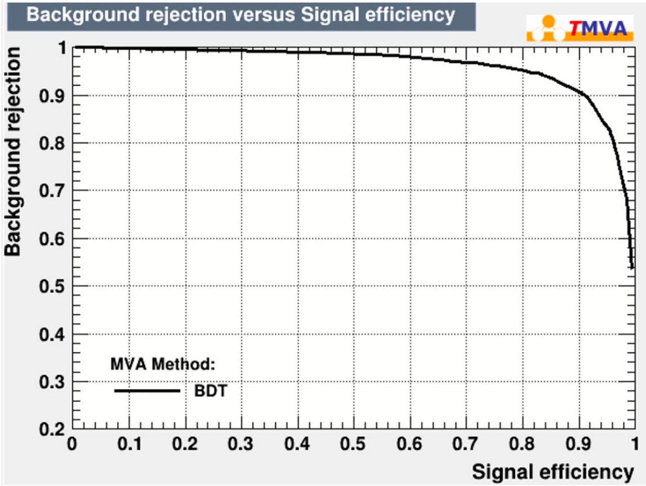


Proton/Neutron ID for nDVCS

A. Hobart (IJCLab)

- Protons account for >40% of contamination for nDVCS events for this analysis on RG-A data
  - Lack of tracks in some areas in the central detector leads to an excess of misidentified neutrons.
- Used Boosted Decision Trees (BDTs) for classification
  - Used detector variables from CTOF and CND in addition to a delta phi variable





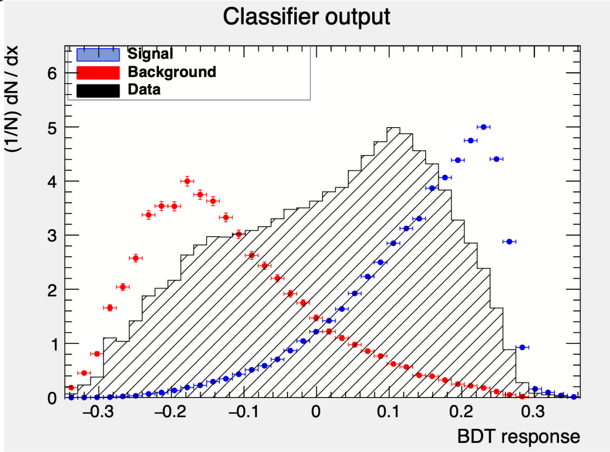
Irène Joliot-Curie
Laboratoire de Physique des 2 Infinis

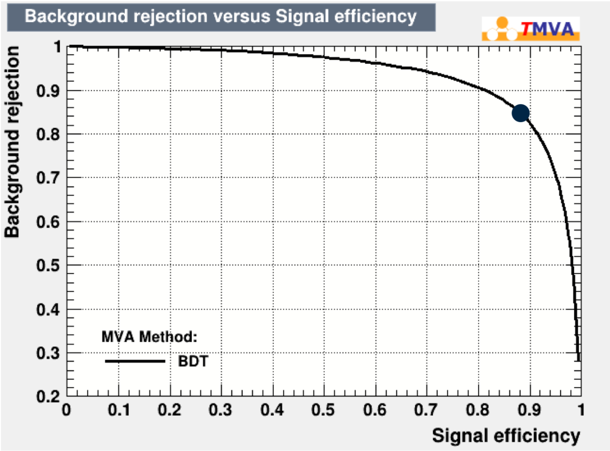
See Adam's earlier talk 3/22/2023 at 10:00

AI Reaction ID for DVCS

J.S. Alvarado (IJCLab)

- Used Boosted Decision Trees (BDTs) for reaction classification to separate $ep\gamma$ events from $ep\pi^0$
- BDT looks promising as a replacement for other event selection





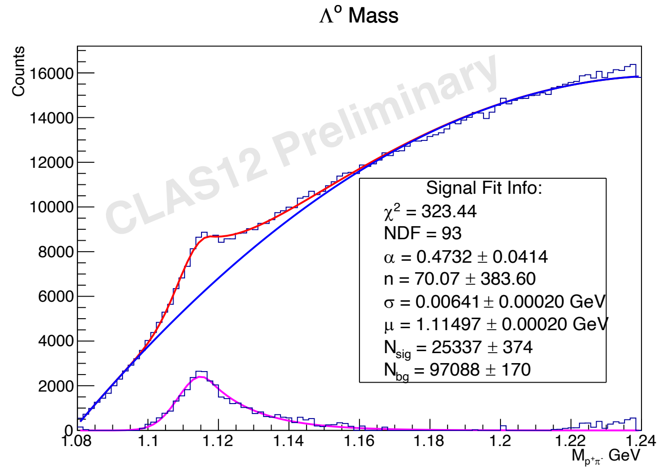
See Juan Sebastian's earlier talk 3/22/2023 at 12:10

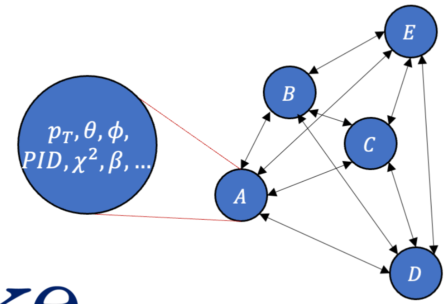
Irène Joliot-Curie
Laboratoire de Physique des 2 Infinis

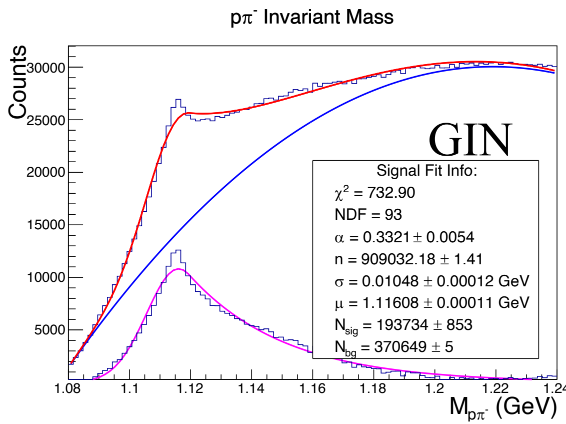
Lambda GNN PID

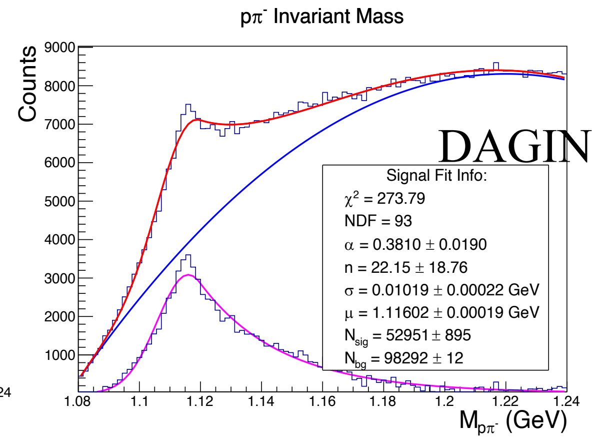
M. McEneaney (Duke)

- Particle ID using graph neural networks
- Each node consists of a particle's features as shown below







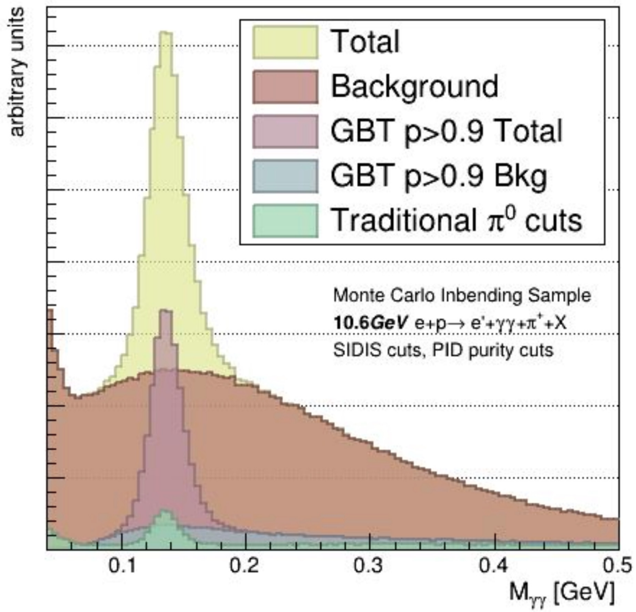


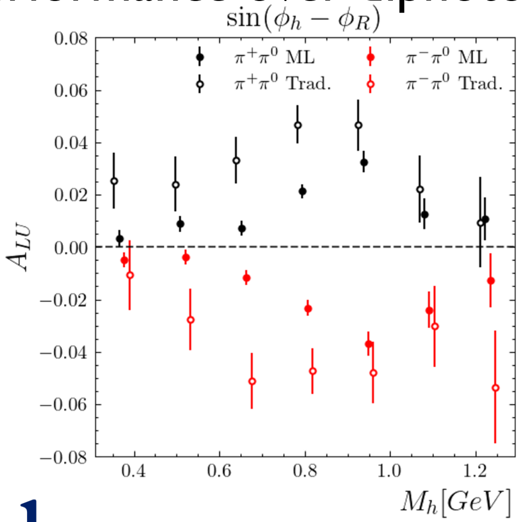
Duke

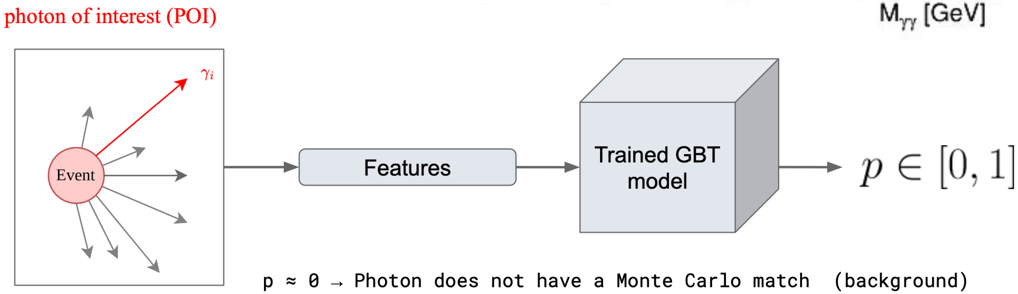
Reaction ID for Photons

- Used Gradient Boosted Trees to identify photons from π^0 decays
- Single photon model had better performance over diphoton model

G. Matousek (Duke)







p ≈ 0 → Photon does not have a Monte Carlo match (background)

...

p ≈ 1 → Photon likely has Monte Carlo match (signal)

Duke

Summary

- Many collaborators have been doing excellent work including AI as a part of their regular toolkits!
- Projects included: PID, Reaction ID, Fast MC, Data Preservation, and PWA
- For PWA we have been able perform PWA "fits" with autoencoders
- Future work includes continued work on hyperparameter optimization, uncertainty quantification, wave selection, and symbolic regression for PWA

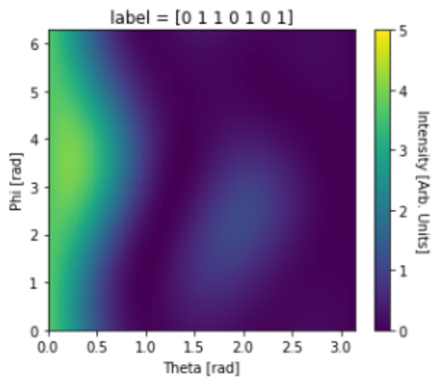
Thanks!

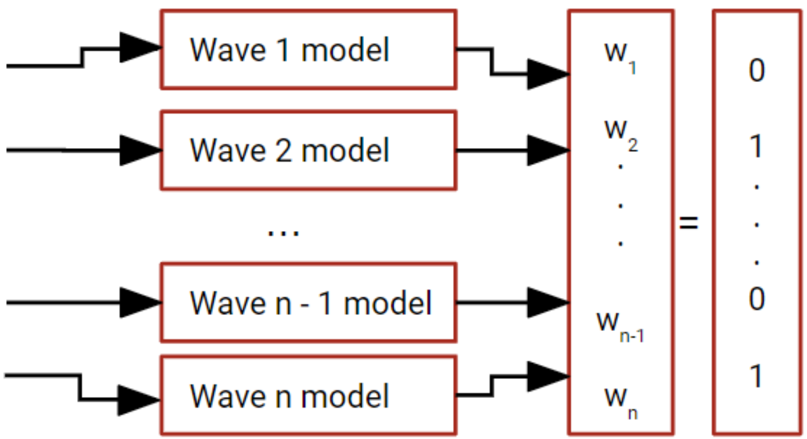
Backup

Wave Selection DNN

- Binary Classifier Ensemble/"Expert" Models
- Literature shows empirical evidence of increased performance
- https://doi.org/10.1016/j.patcog.2011.01.017

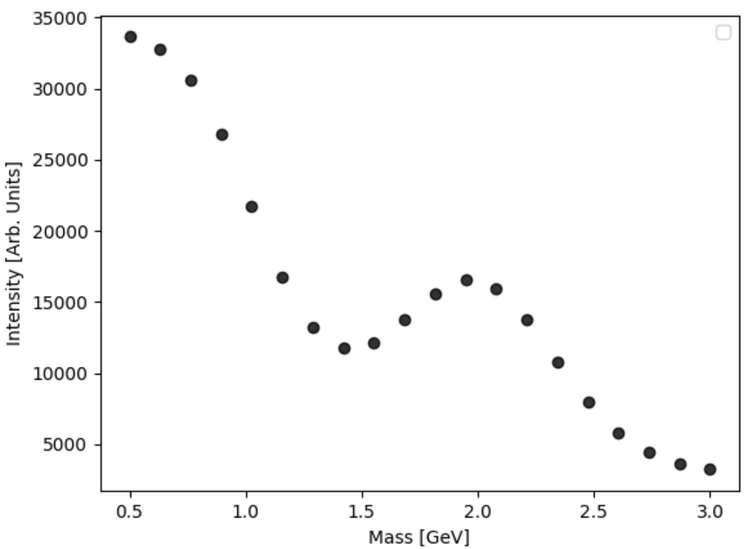
Preliminary results:
96% accuracy in selecting the right set of waves (Lmax=2)
99.4% wave/"digit"-wise accuracy

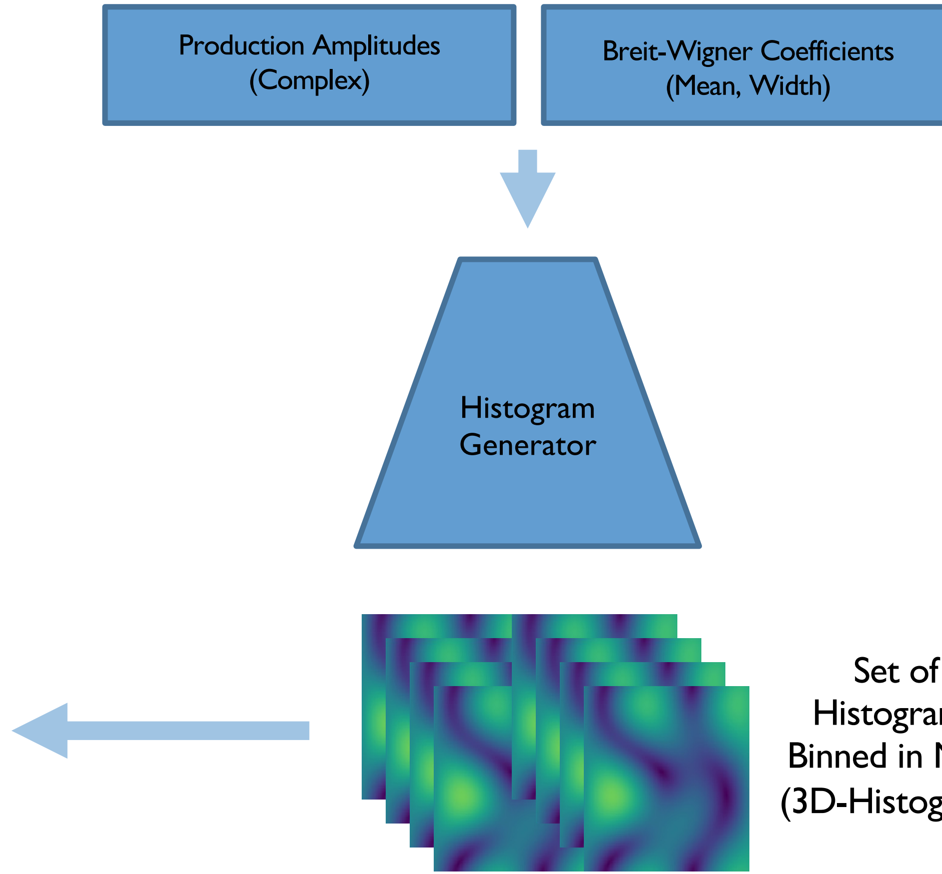




The Mass-Dependent Generator

Randomly Generated Event
(Currently One Resonance per Wave)





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