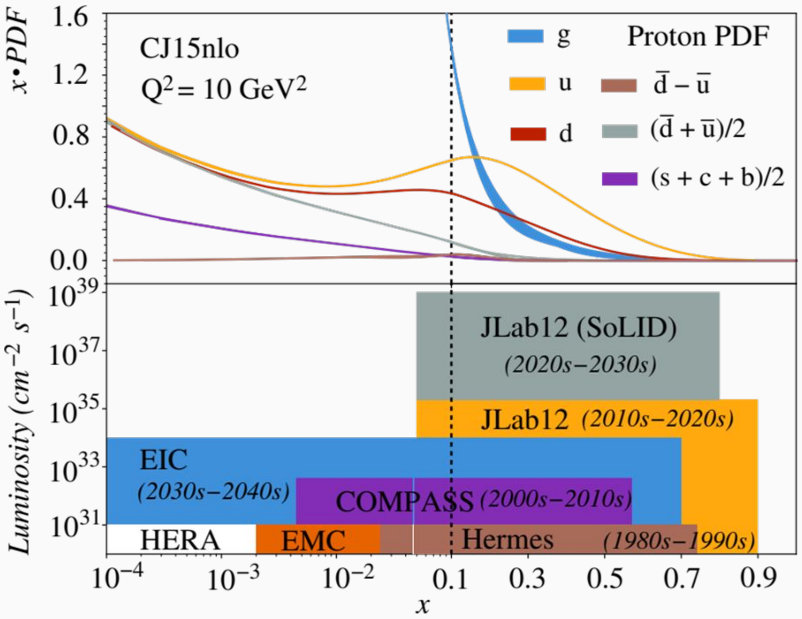
Machine Learning PID
SoLID ECal Beam Test

Darren W Upton

OLD DOMINION UNIVERSITY

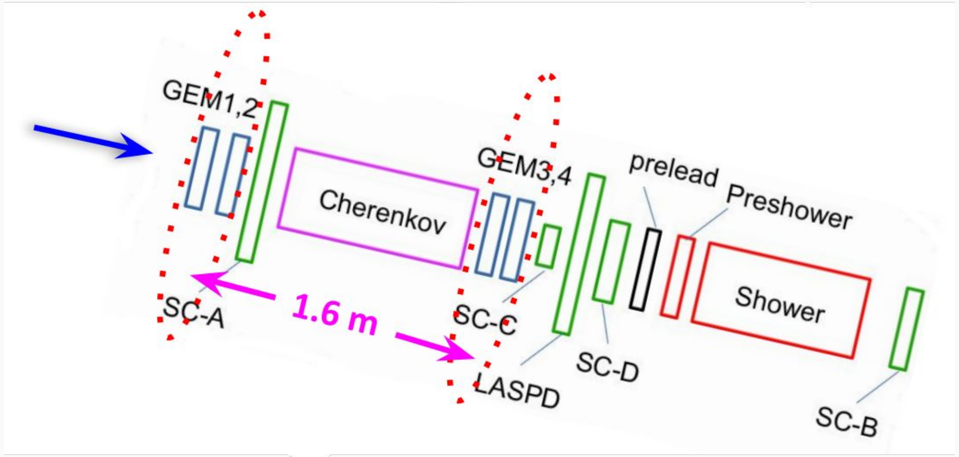
Getting the most out of SoLID

- SoLID is about getting most out of JLab
- How do we get the most out of SoLID?



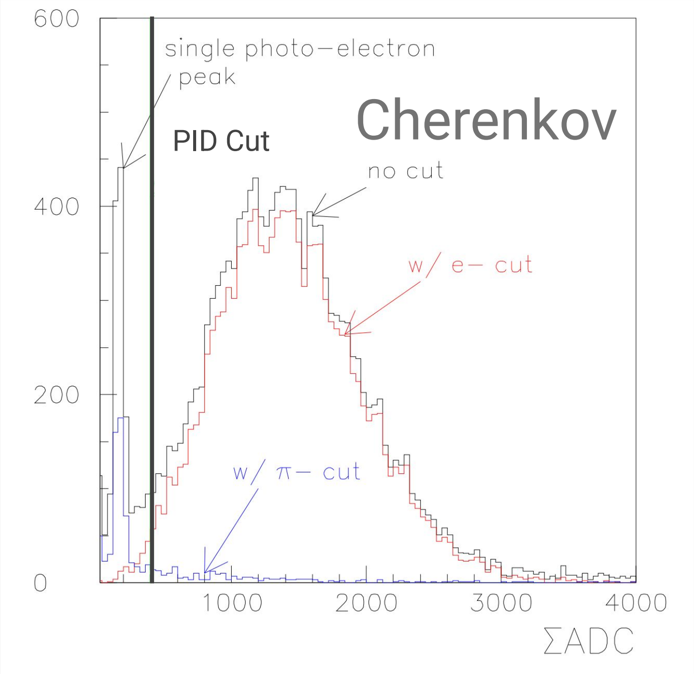
SoLID ECal Beam Test

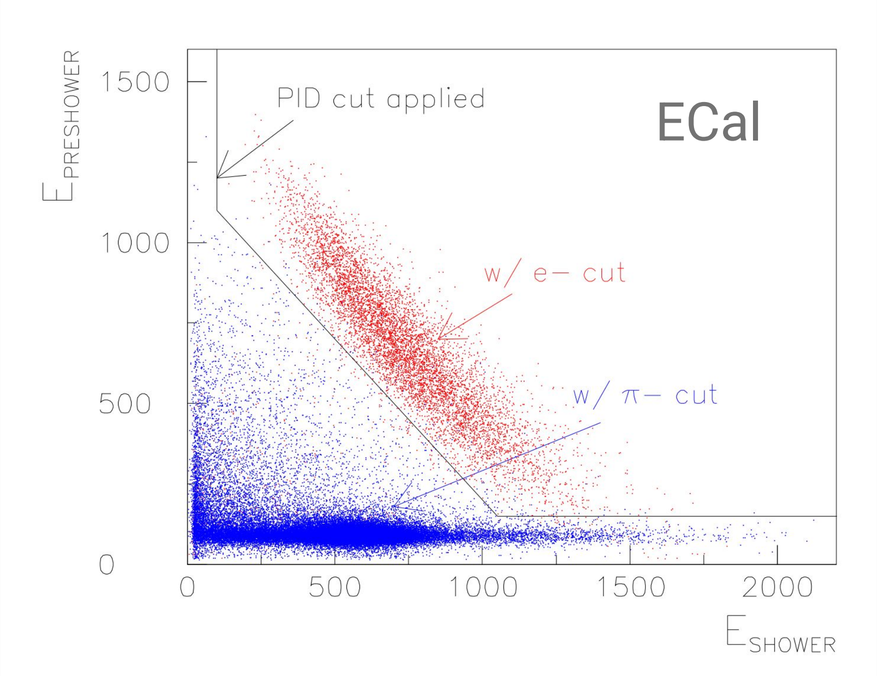
- ❖ Focus on characterizing ECal
- ❖ Main Detectors
  - ➢ 3 PreShower-Shower modules
  - ➢ 4 scintillators
  - ➢ Light-gas Cherenkov



“Classical” Particle IDentification (PID)

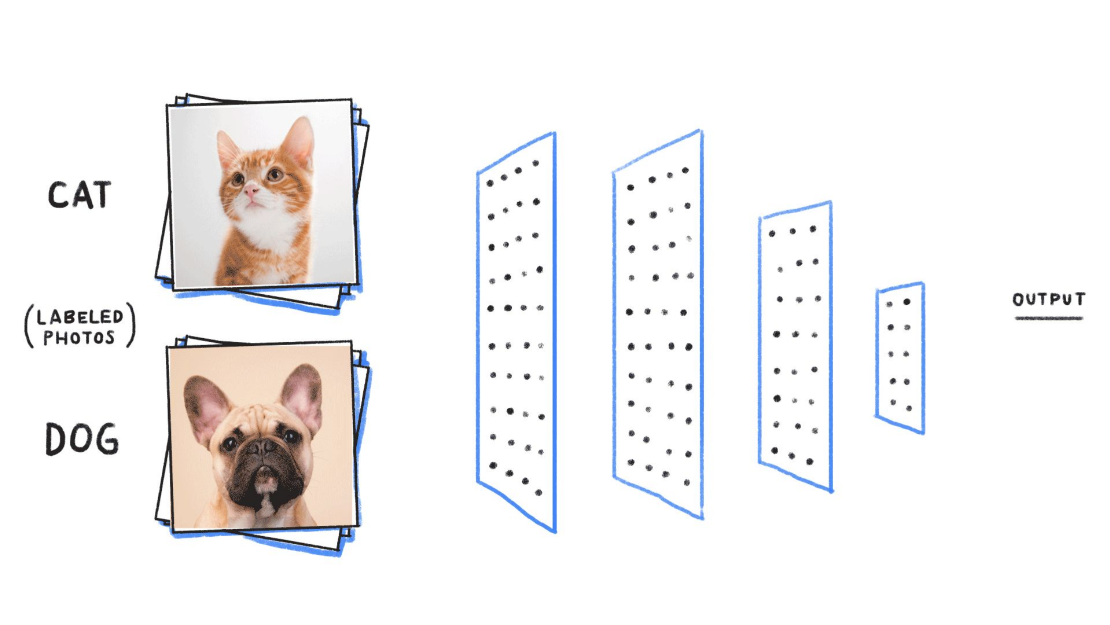
- Selecting electrons vs charged pions, start with Cherenkov & ECal cuts
- Low dimensional cuts remove “good” events





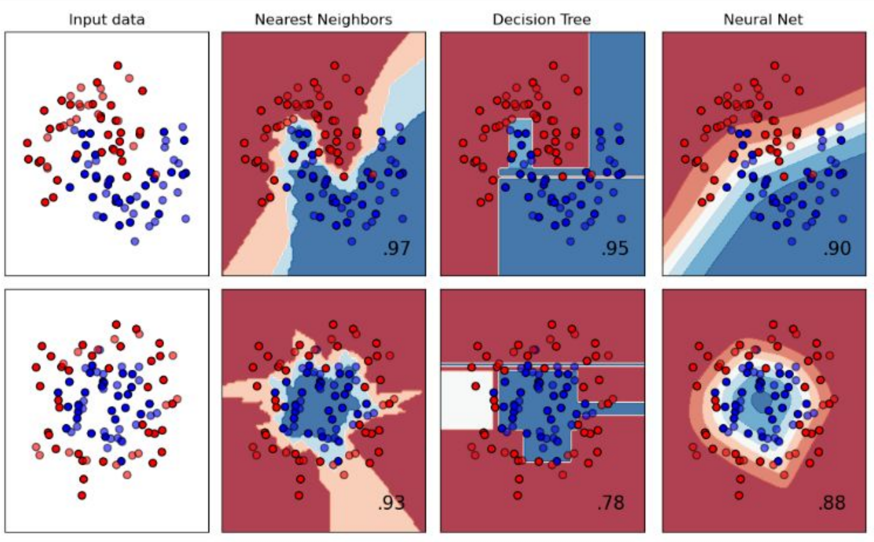
Machine Learning for Classification

- Train on labelled images/data
- Determine label for given set of input
  - Label Cats vs Dogs, etc



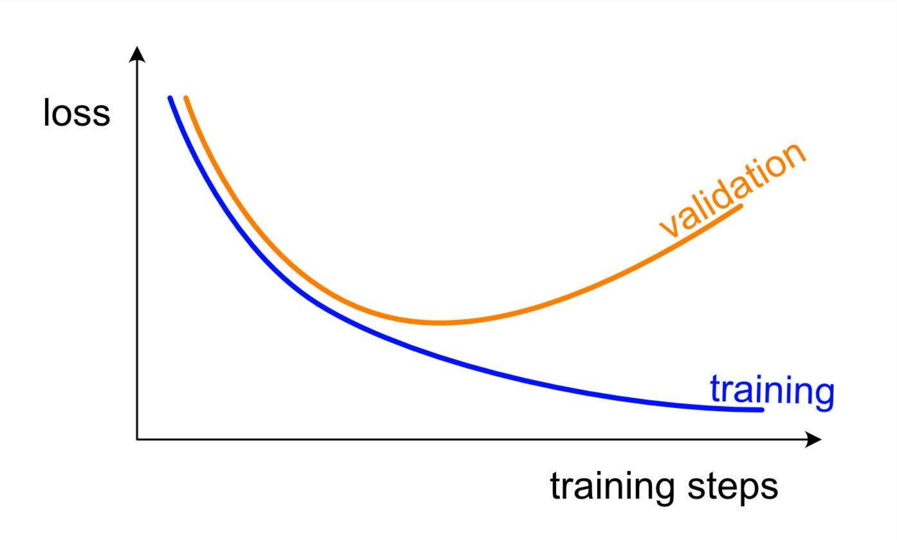
Machine Learning for Classification

- ❖ Given some labels for values in input space
- ❖ Optimize separation of classes
- ❖ Multiple approaches for supervised & unsupervised
  - ➢ Clustering, decision tree, NN, etc



Machine Learning for PID

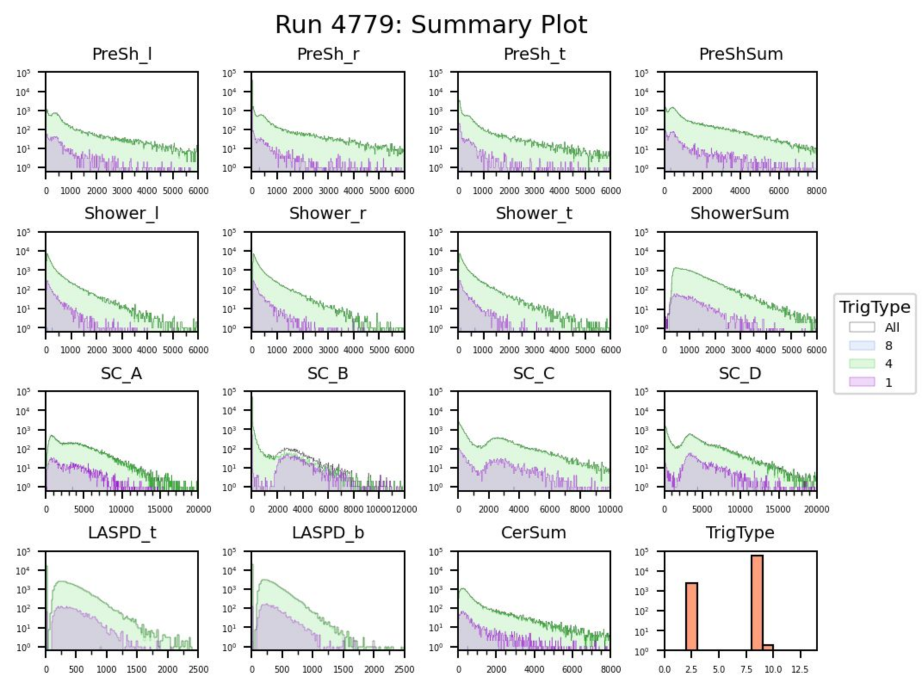
- Train NN given some labels with embedding in input space
- Study training metrics & model performance



		Predicted	
		Spam	Non-spam
Actual	Spam	**600** (True positive)	**300** (False negative)
	Non-spam	**100** (False positive)	**9000** (True negative)

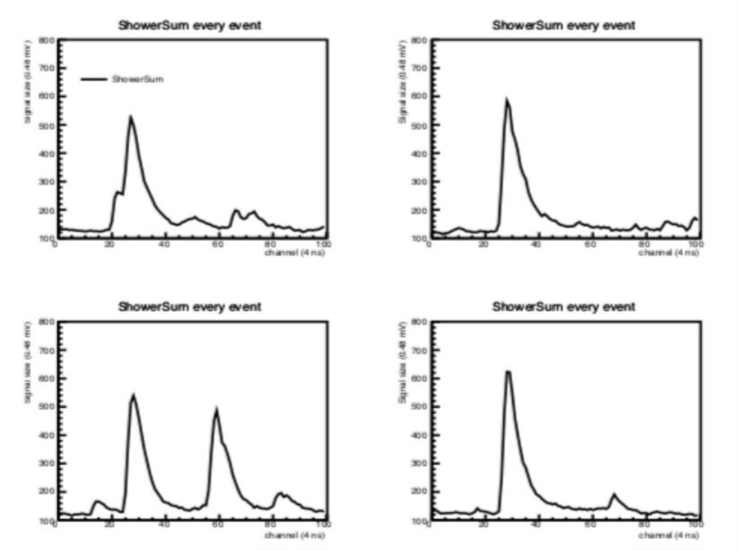
Data Distributions

- ADC values for different detectors from different triggers
- Determine Minimum Ionizing Particle (MIP) peaks for sim-data scaling factor

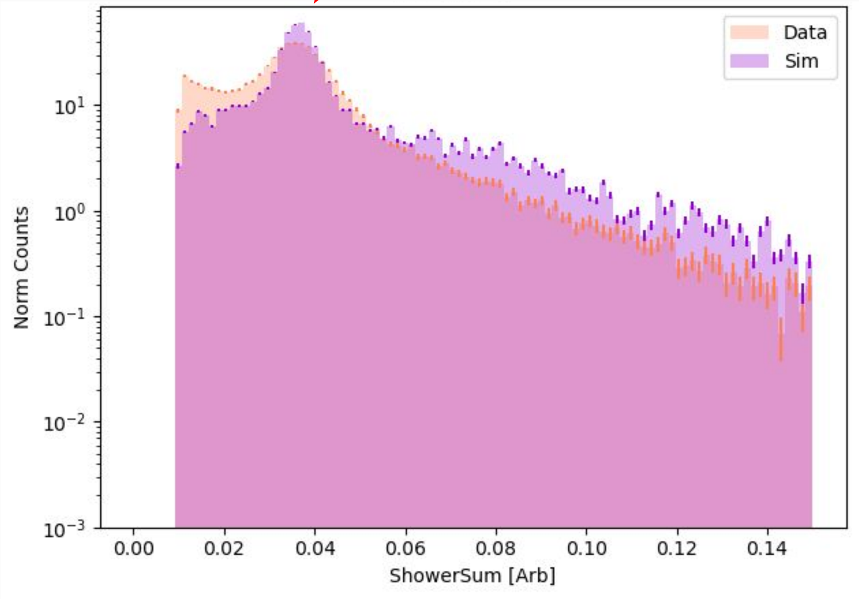


Background Mixing & Smearing

- ❖ Match sim-data overall distributions
- ❖ Merge concurrent EM background into sim events
  - ➢ Rate 3:1 bkg:sim for 10 uA
- ❖ Smear sim MIP peaks to match data

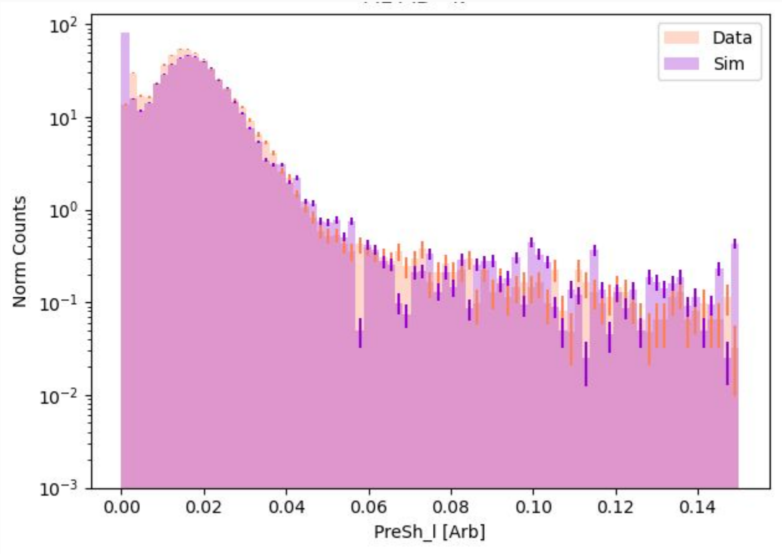


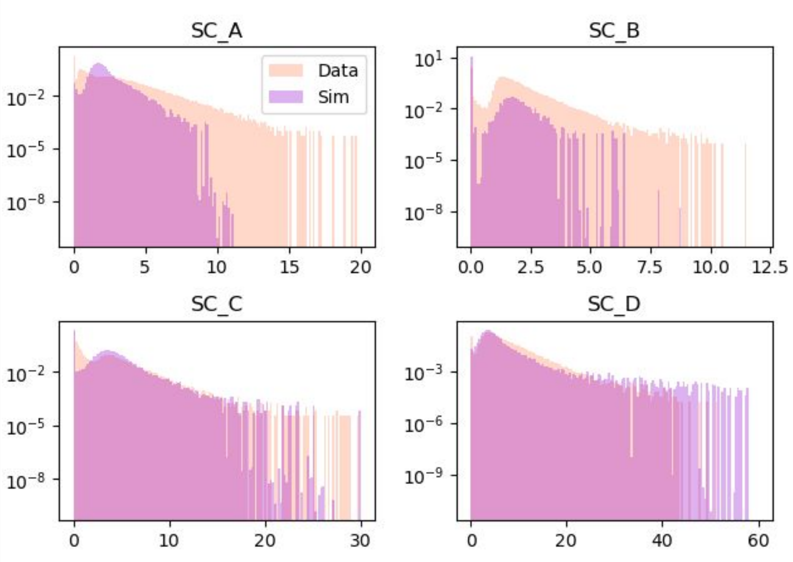
Narrow MIP Peak



Data-Sim Comparison

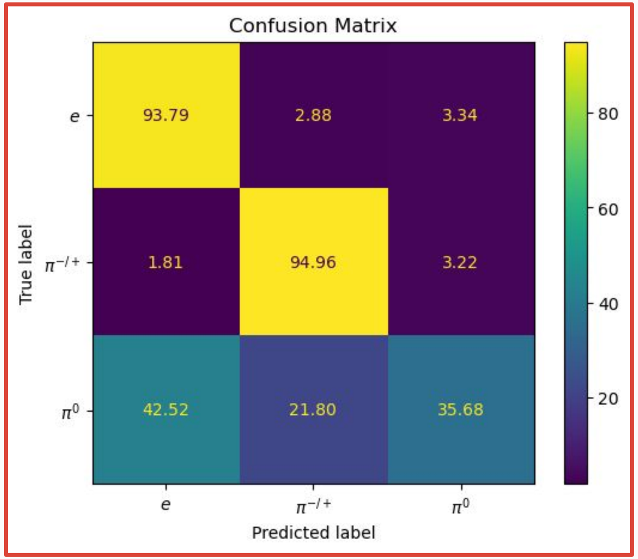
- MIP peak aligned & scaled for ECal modules
- More work needed for some scintillators

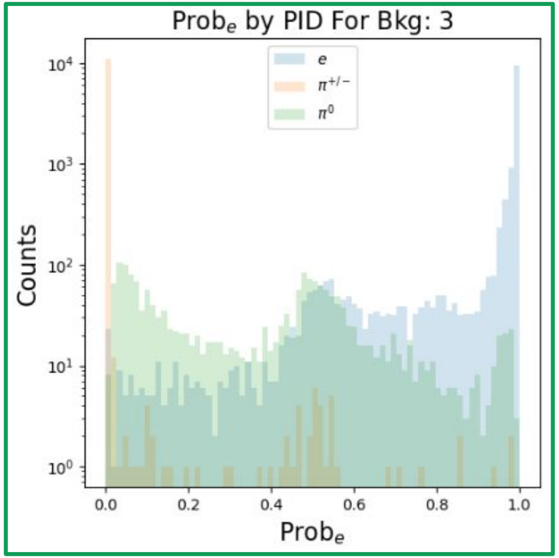




ML Model - Output

- Train model for 10 uA data & check performance for e^-, $\pi^{+/-}$, π^0
- Confusion matrix shows good e^- vs $\pi^{+/-}$ but poor e^- vs π^0
- Reduce high-dimensional input into 1D probability distribution





Figures By: Mohhamed Rafi

$\pi^{+/-}$ PID - Shower Modules

- Charged pion classification
- Reasonable match between data & sim in Shower
- More tuning needed for shoulder before MIP

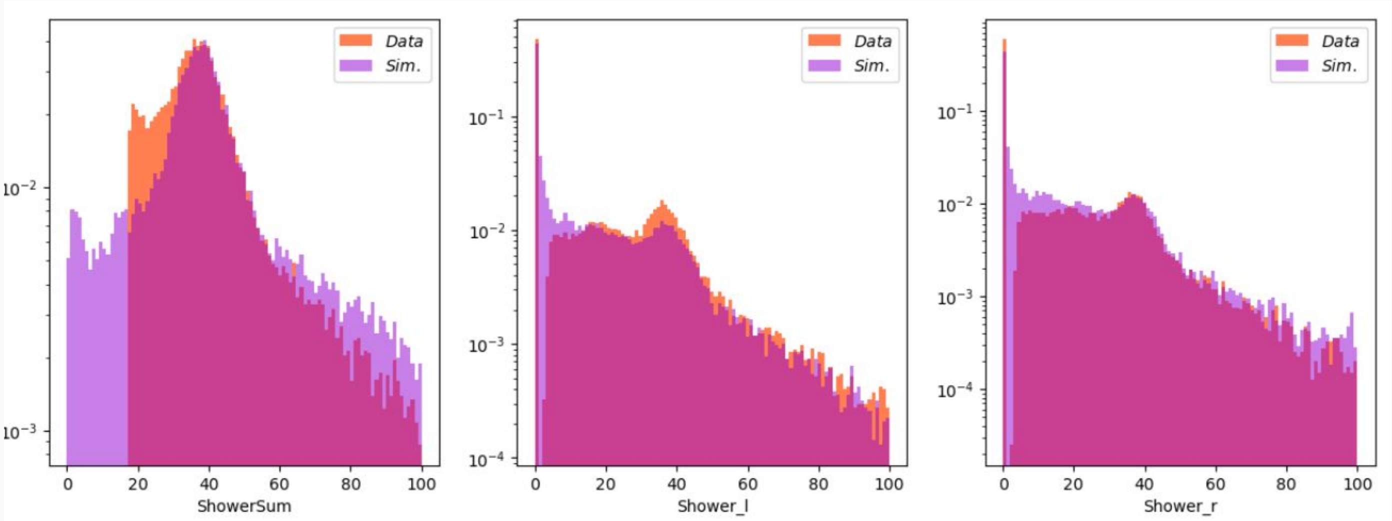
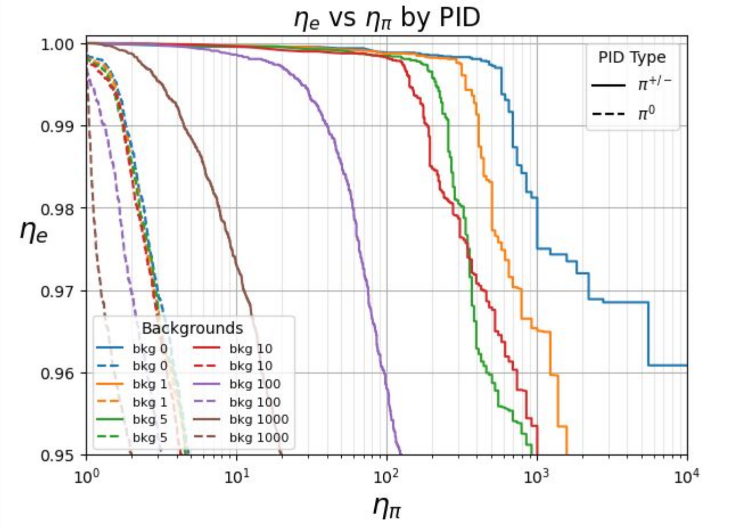


Figure By: Mohhamed Rafi

Impact of Background Merge Factor

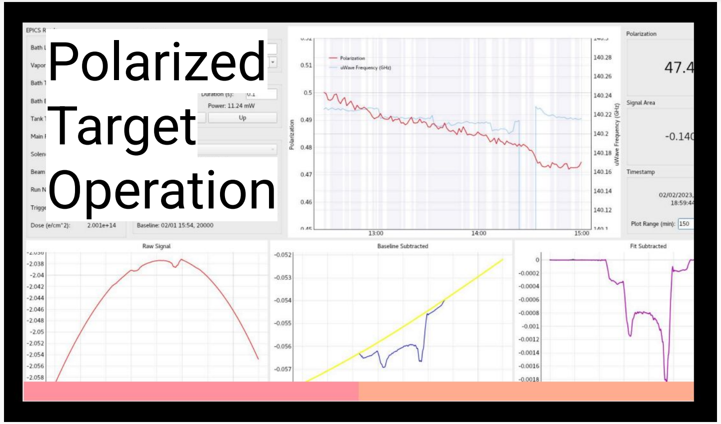
- Increasing background:signal decreases performance
- Map ratio onto beam current then compare with classical PID

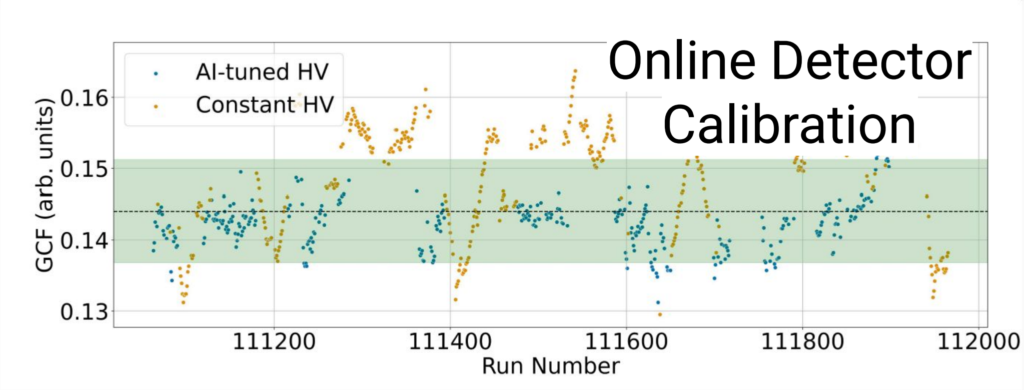
Bkg Sampling Ratio	Electron Efficiency	$\pi^{\pm}$ Rejection
0	0.9674	473.2436
1	0.9499	439.7209
3	0.9675	370.5415
5	0.9623	291.6676
10	0.9355	309.3121
13	0.9222	300.9621
100	0.9331	102.3948



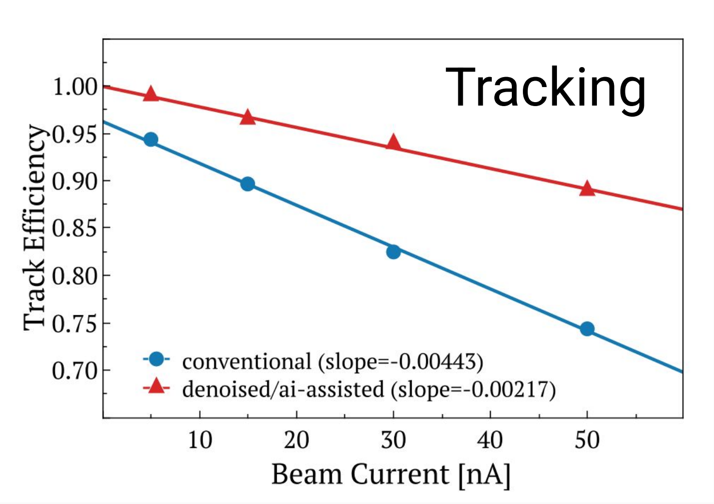
Figures By: Mohhamed Rafi

Other Applications for ML





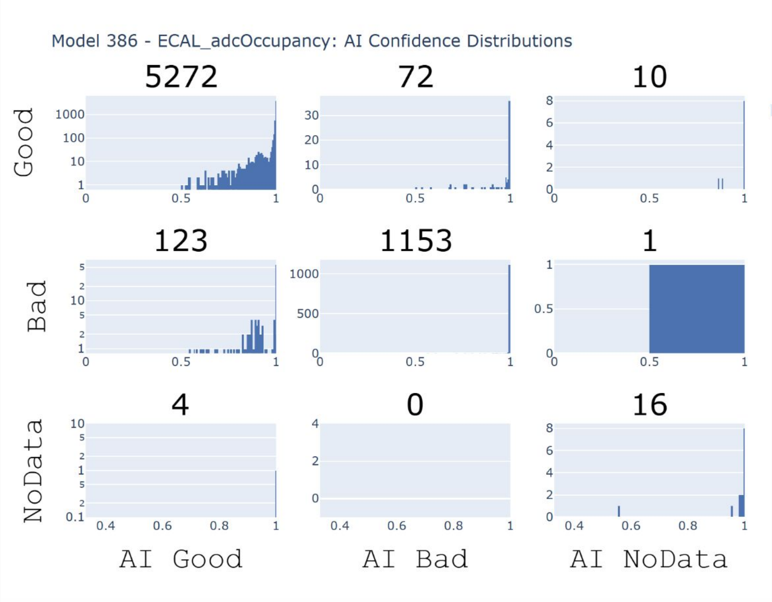
HYDRA



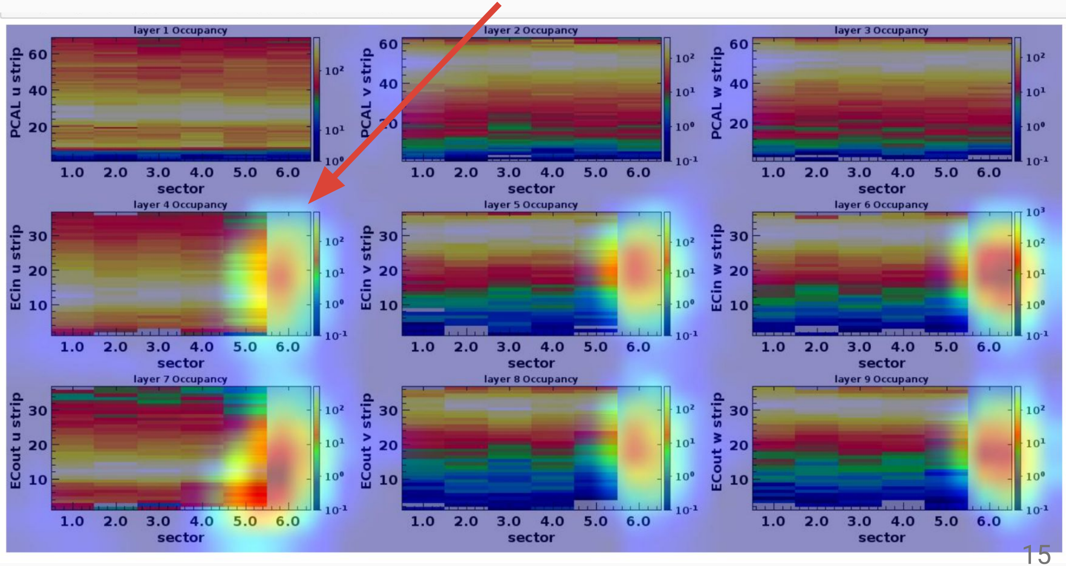
Supervised ML Classification via Hydra

- Train image-classification NN on monitoring plots
- Augment failure examples with pseudo-data

HYDRA

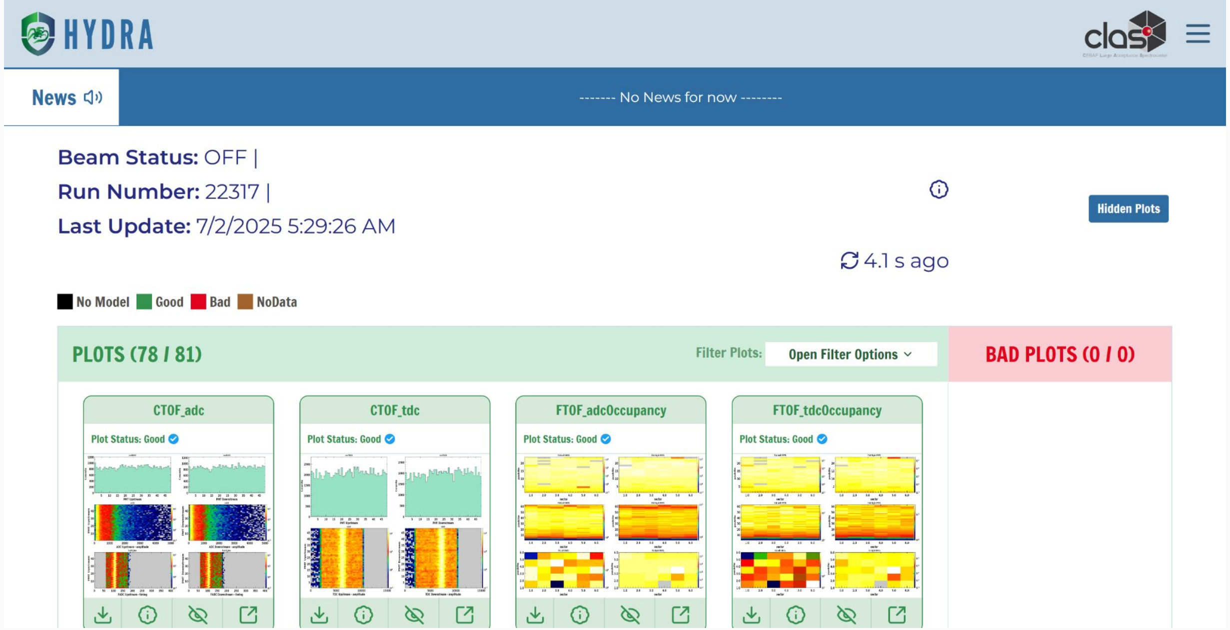


Model looks at images and finds problems



Supervised ML Classification via Hydra

- Same general principle as ML for PID
- Interface operational for all four experimental halls
- Online models for CLAS12 and GlueX



What's the Point of ML?

- Leveraging correlations in high-dimensional data
  - ML PID boils complicated cuts into 1D probability cuts
- Data Science / ML methods forces careful understanding of data
  - Careful matching of sim-data needed for ML PID
- Developing ML-based tools provides training ground for students
  - Taught detector physics, analysis methods, etc to me + 4 students

Questions/Comments

ML Model - Basics

Small NN

```
Dense(128, activation="relu"), BatchNormalization(), Dropout(0.15),
Dense(64, activation="relu"), BatchNormalization(), Dropout(0.15),
Dense(32, activation="relu"), BatchNormalization(), Dropout(0.15),
Dense(16, activation="relu"), BatchNormalization(), Dropout(0.15),
Dense(8, activation="relu"), BatchNormalization(), Dropout(0.15),
Dense(len(np.unique(y)), activation="softmax")
```

Trigger Cuts

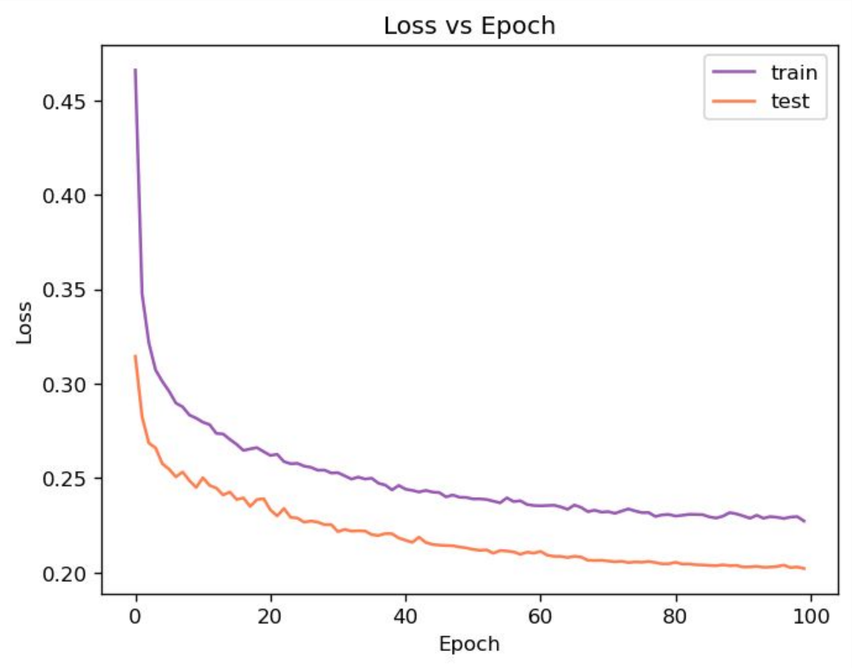
Outlier Cuts

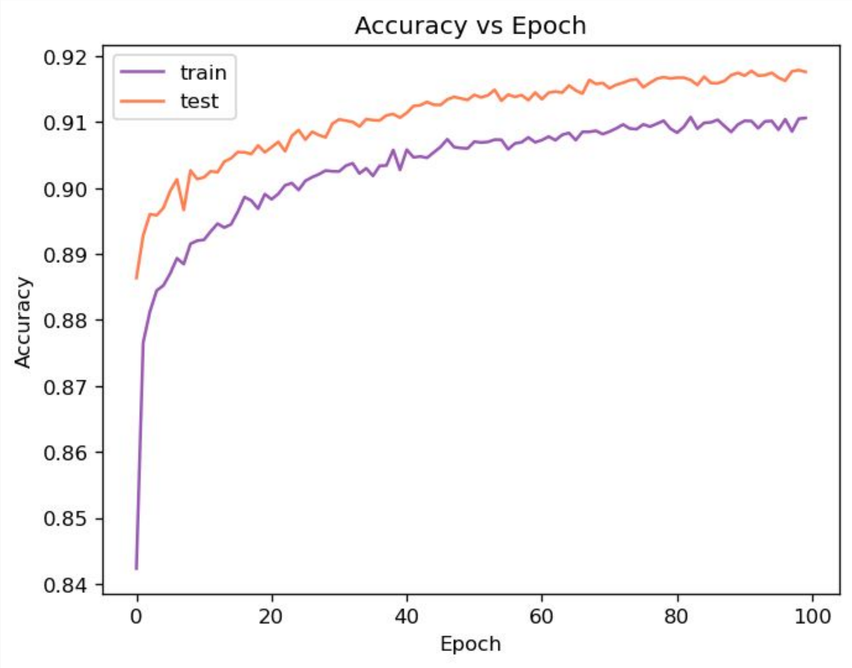
Train:Test - 60865:26085

```
#Number of background events per data event
n_bkg = 0#1#10

Scint_MIPs = [1.65, 3.5, 3.5, 1.65] # A, D, C, B

keeps = (sim_df["pid"]!=0) #((sim_df["pid"]==11) | ((sim_df["pid"]==211)) |
trig_keeps = ((sim_df["SC_A_Eendsum"]>Scint_MIPs[0]/2)
              & (sim_df["SC_D_Eendsum"]>Scint_MIPs[1]/2)
              & (sim_df["ShowerSum"]>.5)
             )
outlier_cuts = ((sim_df["SC_A_Eendsum"]<10) & (sim_df["SC_B_Eendsum"]<4) &
                (sim_df["SC_C_Eendsum"]<20) & (sim_df["SC_D_Eendsum"]<50))

#data_np = sim_cher[keeps].to_numpy() #Cher Channels
data_np = (sim_df[(keeps & trig_keeps & outlier_cuts)]).to_numpy()
#bkg_np = ((raw_bkg_df.sample(n=n_bkg*len(data_np), random_state=42, replac
#cher_np = Cher_df.to_numpy()

X = data_np[:, [16,17,18,19, 20,21,22,23, 31,28]]#, 25,34,31,28]]#[0, 16,17
```

ML Model - Training

- ❖ Provides information on training performance
- ❖ Offset is just normalization effect



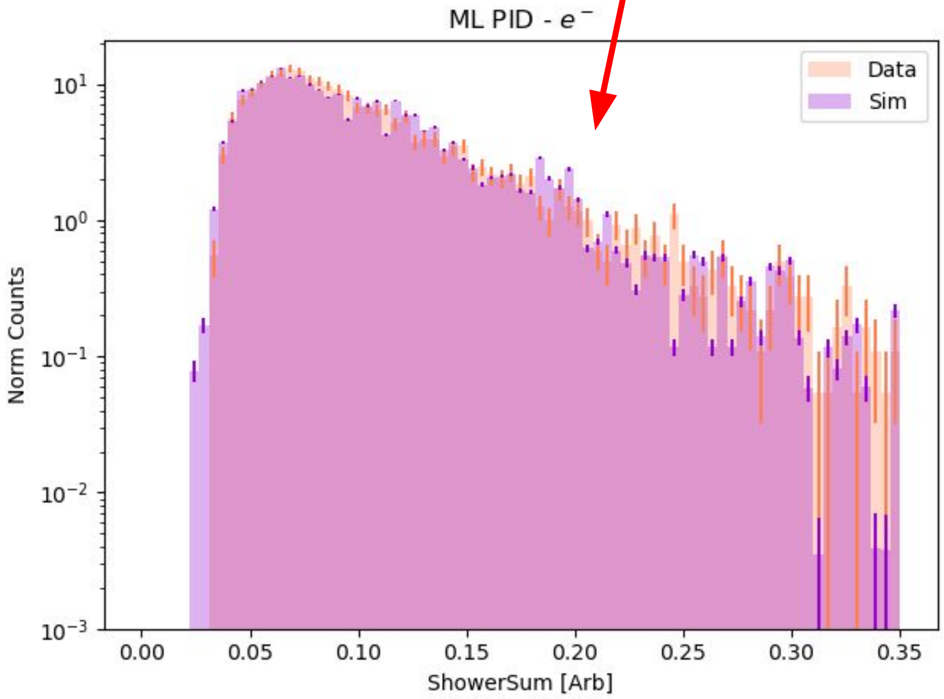


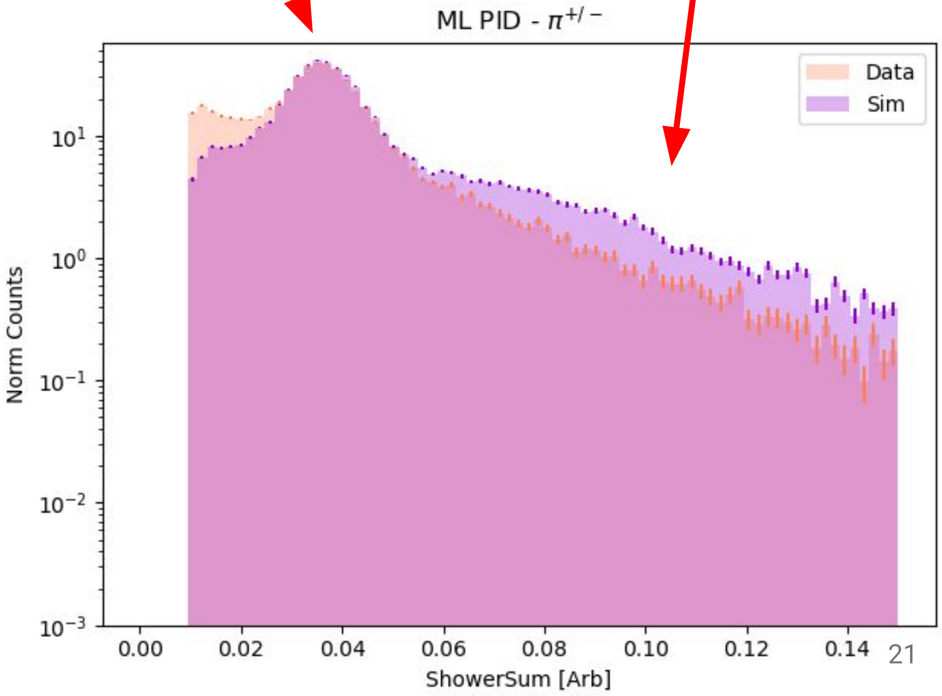
ML PID - Shower Sum

Good Agreement

MIP Match

Tail Disagrees





Ongoing Questions

1. **Data-Sim Scaling**: Aligning data-sim distributions
2. **ECal Resolution:** What resolution/smearing effects should be considered? We use 35% for the PreShower and 10% for the Shower.
3. **Sim Rate**: Are the "# rate" values accurate? This is critical for realistic comparison where we weight the histograms by rate.
4. **Data Runs**: Which beam currents can we use