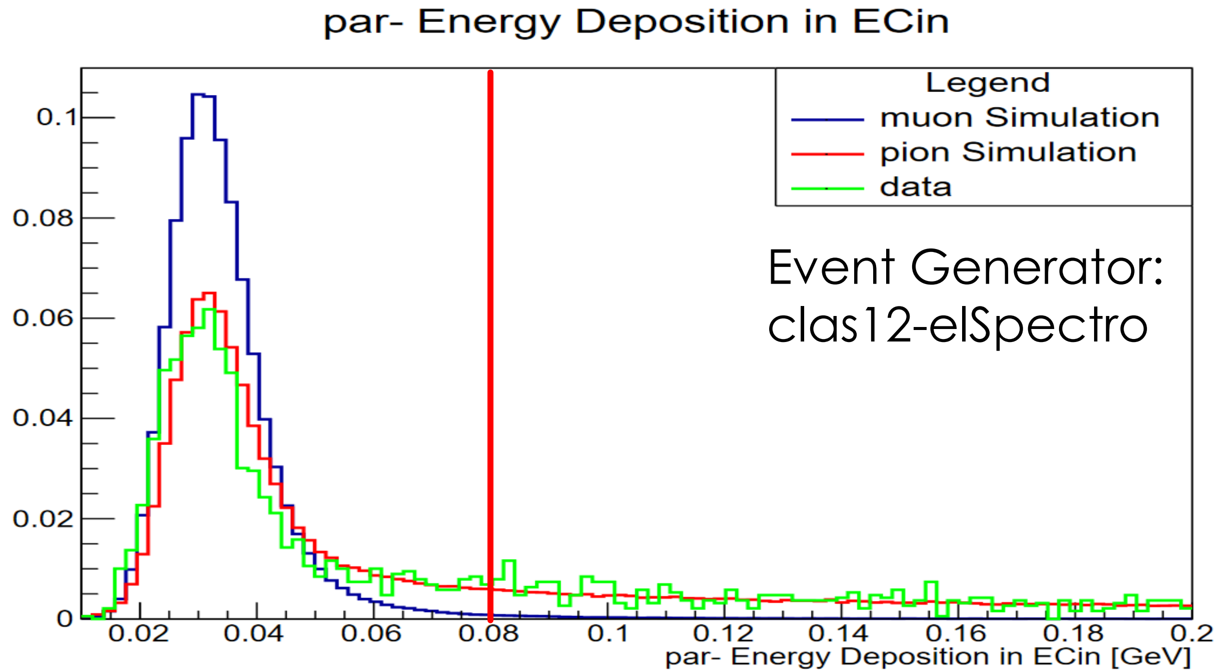
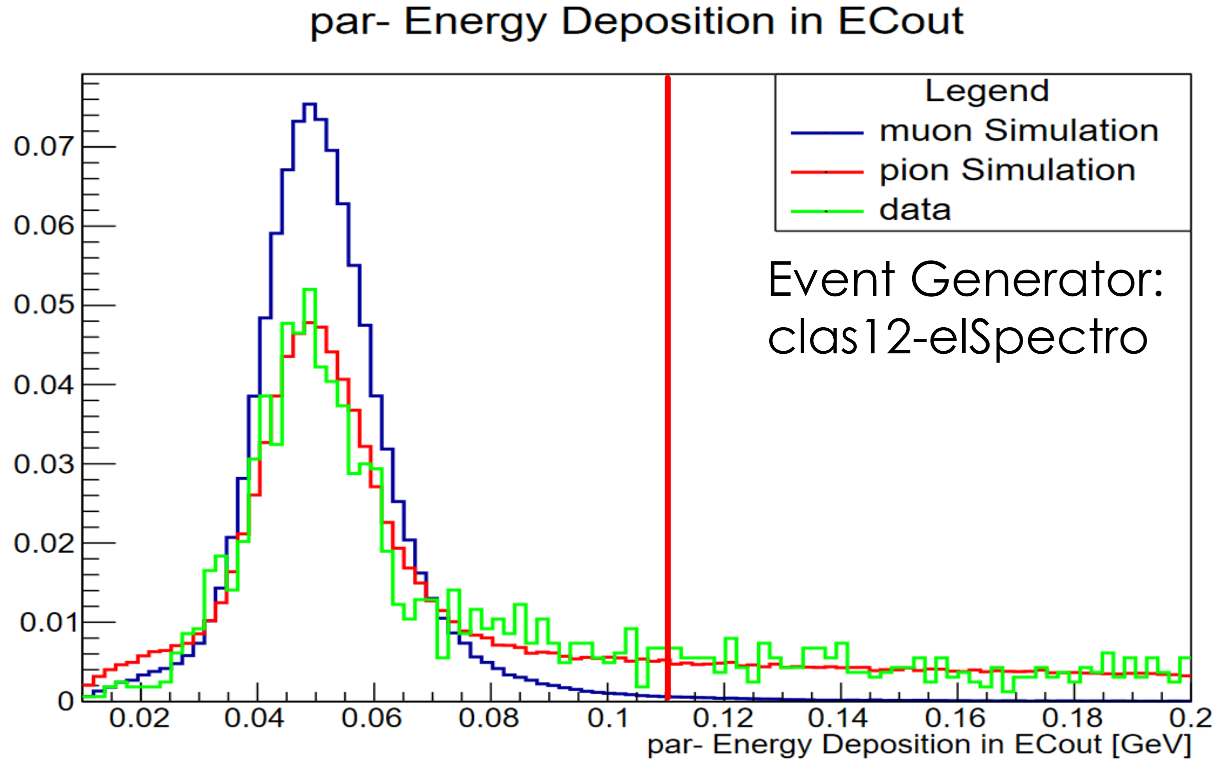
ML RID Projects

RICHARD TYSON

Muon ID Refinement

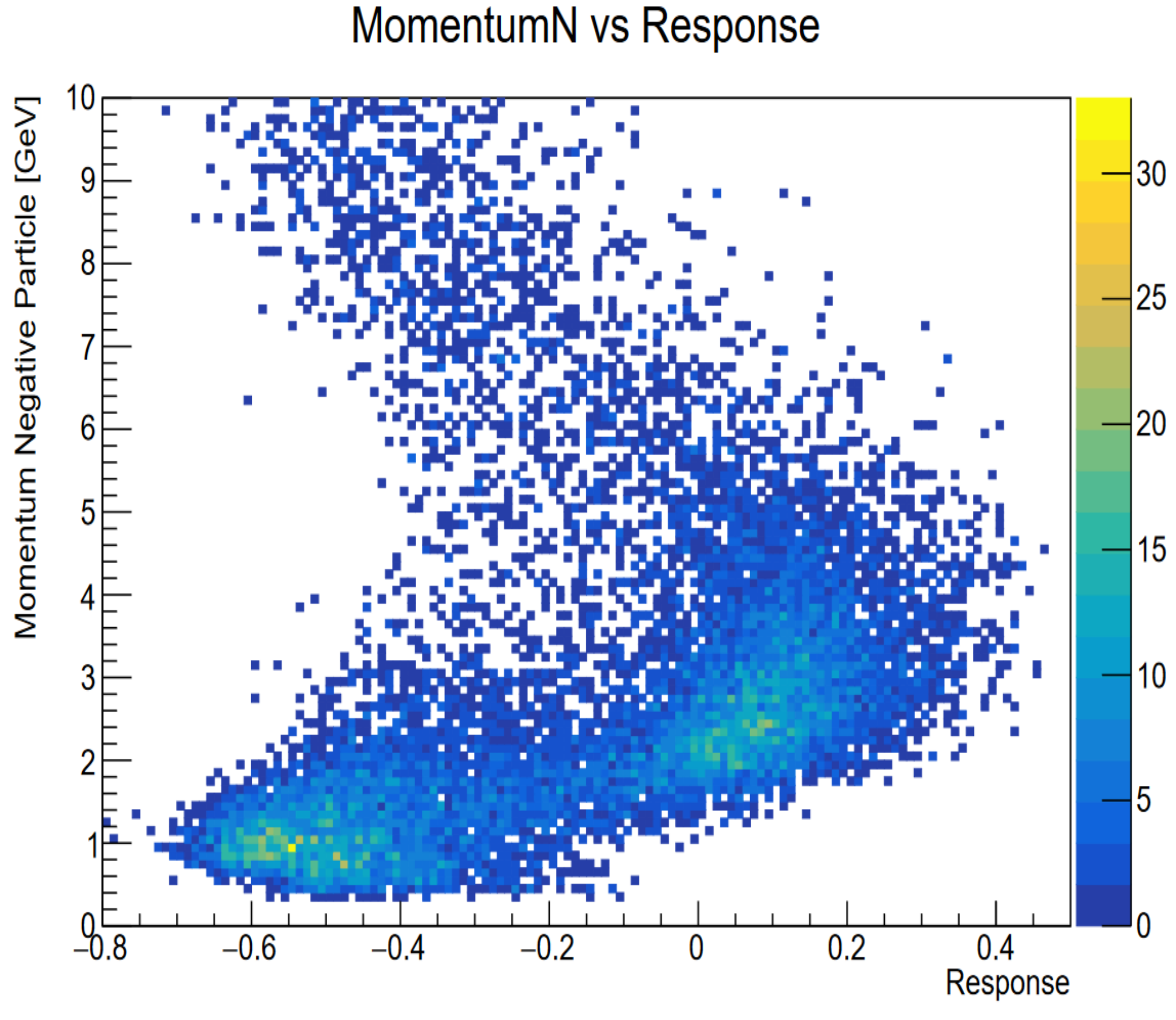
- Muon candidates are minimum ionizing particles and are therefore selected based on their energy deposition in the calorimeters.
- These cuts are susceptible to a high pion contamination which we can improve with a classifier.
- Training/prediction is done with chanser interface for the ROOT TMVA software package.
- Our positive and negative training samples are then:
  - MC $\mu^+\mu^-$ which pass energy deposition cuts.
  - MC $\pi^+\pi^-$ which pass energy deposition cuts.





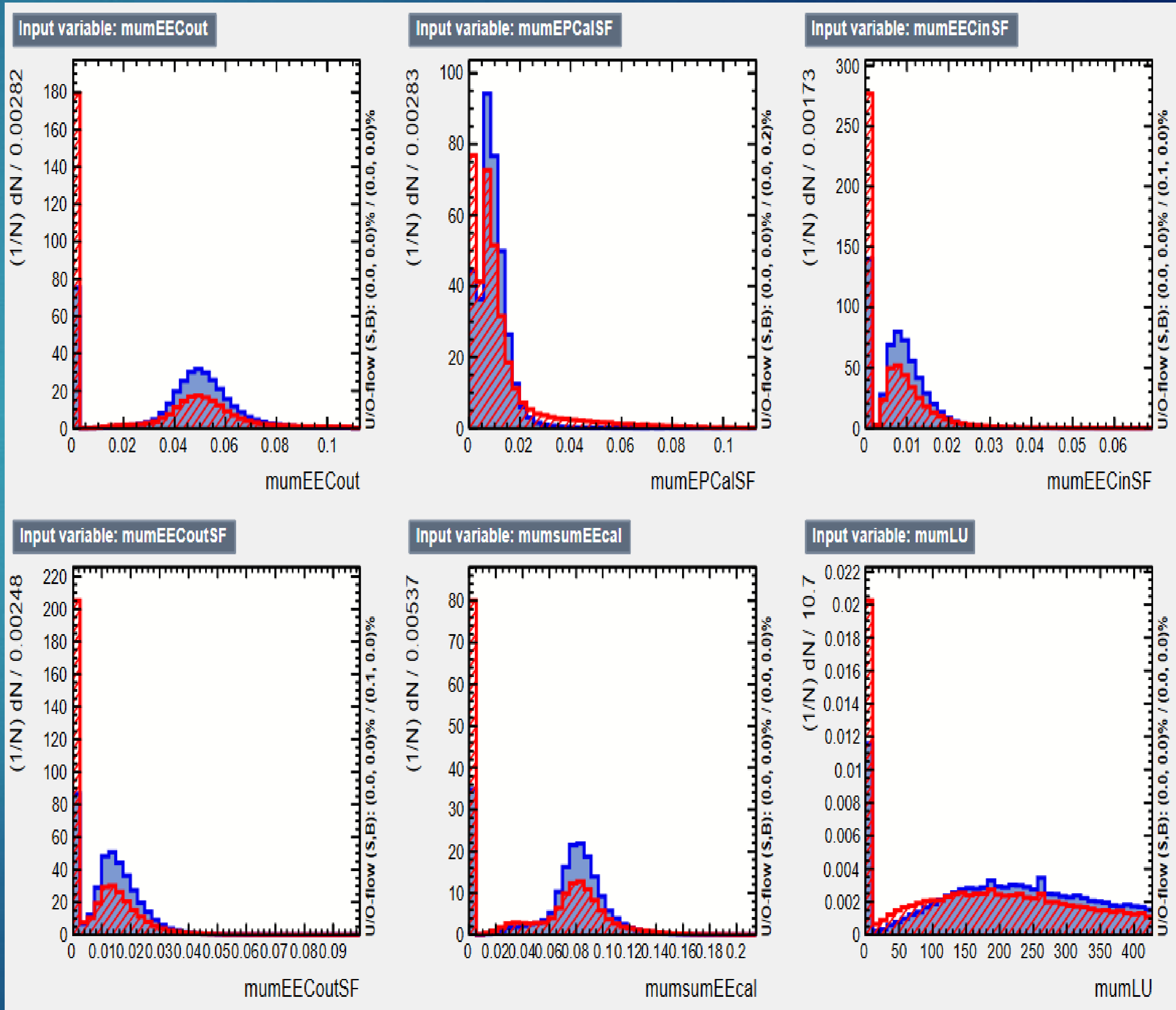
Importance of the Training Sample

- Initially I trained with a simulation that had a J/psi resonance for the muons but no resonances for the pions.
- The issue there is that the classifiers then learned the kinematics of the individual particles instead of distinguishing between particles.
- To avoid this, I created simulations without any resonances for both muons and pions.



Training Variables

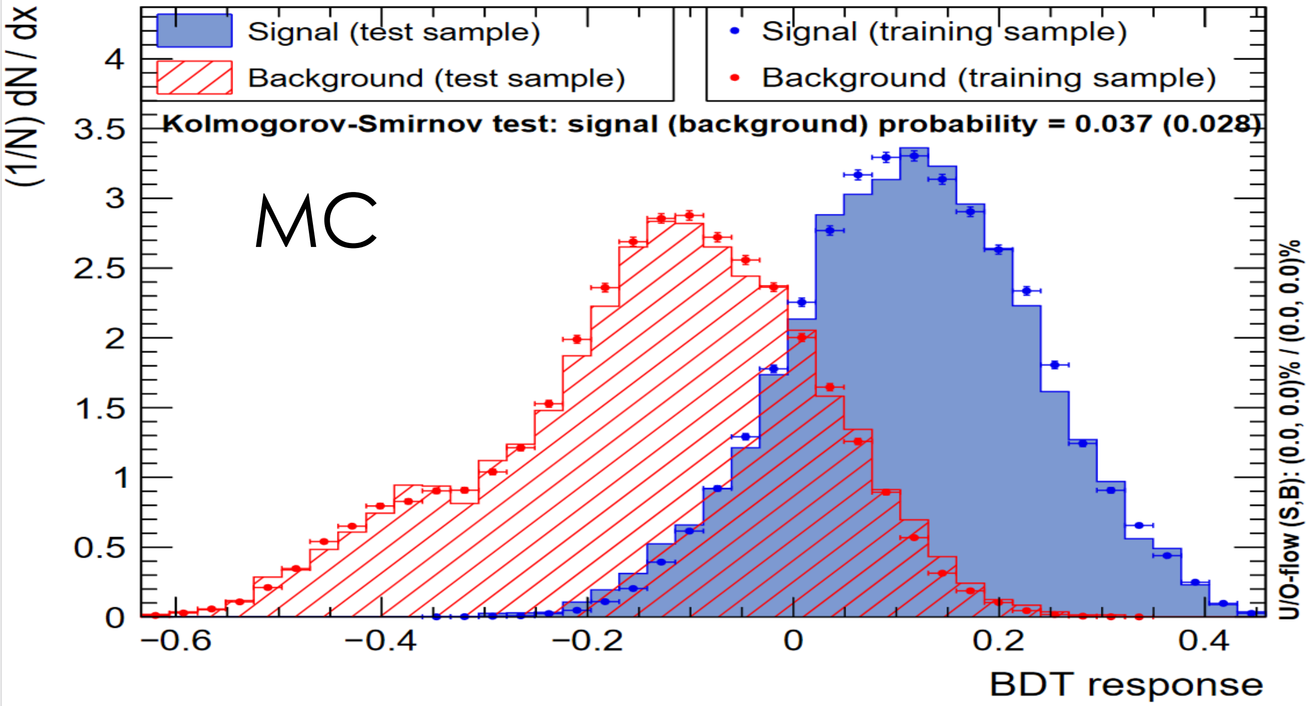
- We train with variables from both positive and negative particles at the same time.
- These are mainly P/Theta/Phi then info from the calorimeters (ie sampling fractions, LU/LV/LW, DU/DV/DW, M2U/M2V/M2W).

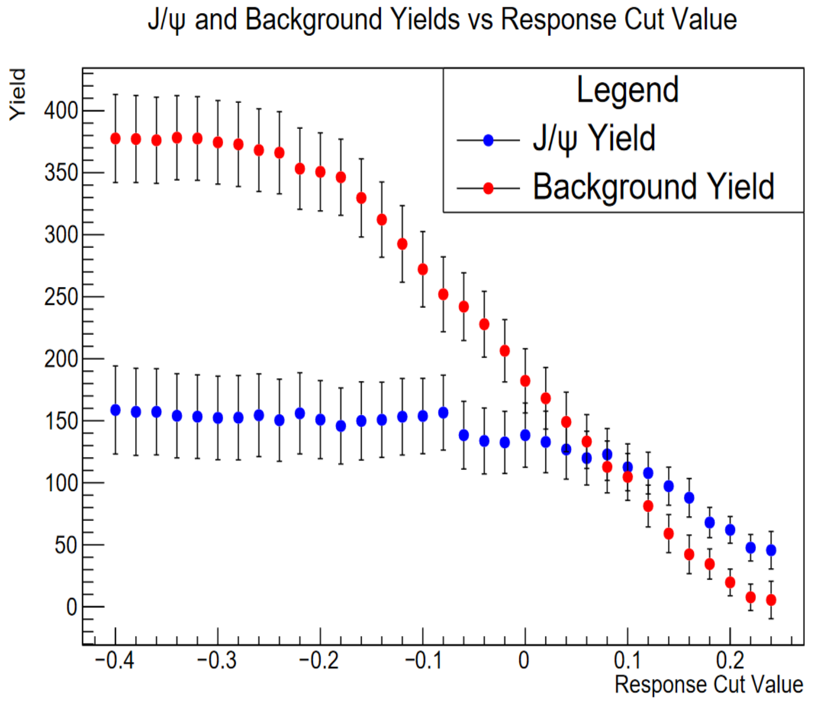


Response

- The classifier output is given as a probability of being a signal event. We call this probability the response.

- The classifier effectively reduces the PID process down to a cut on the response.

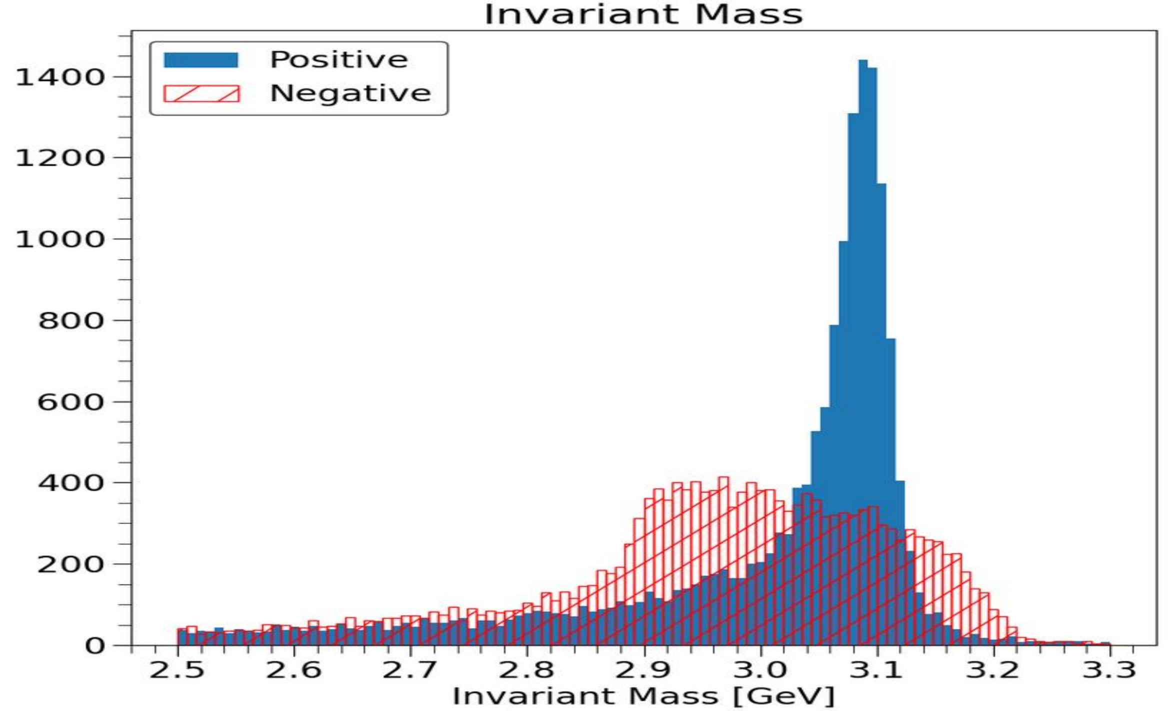


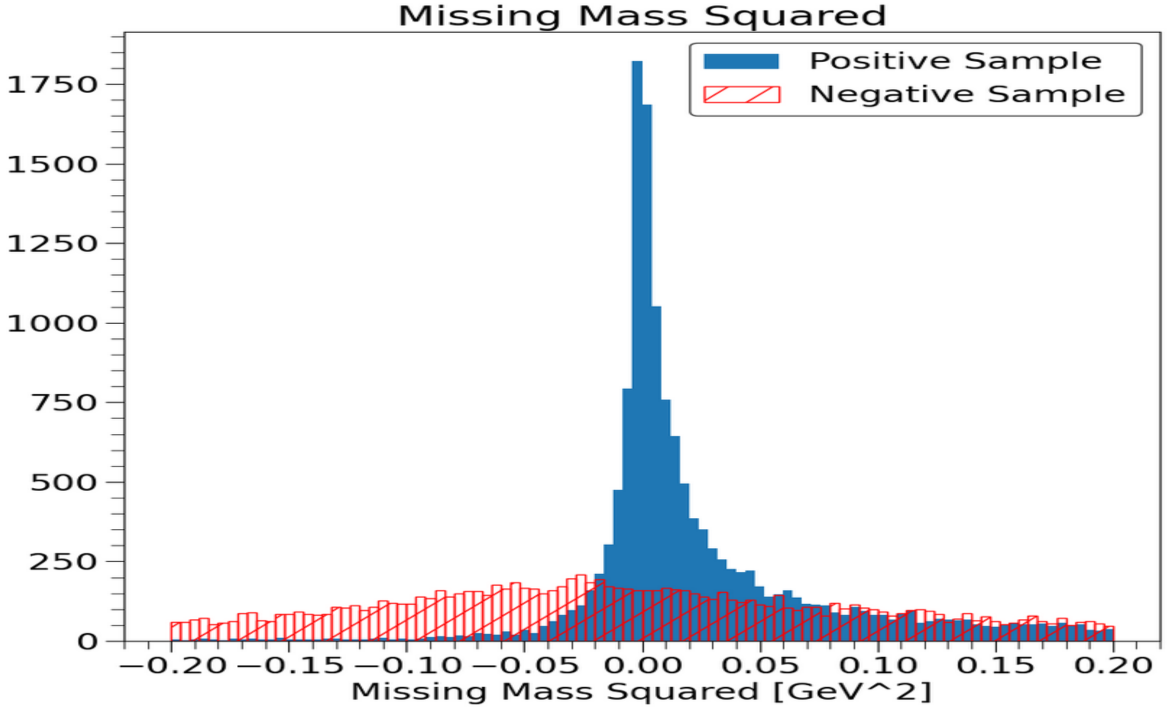


RG-A fall 2018

AI Reaction ID

- Here we do something different from muon ID where we train the classifier to recognize specific reactions, therefore with resonances in the simulations.
- However, the same principle still applies in that don't want the classifier to learn the P/Theta/Phi of individual particles.
- We can enforce this by creating a negative sample by mismatching particles from different events of the positive sample



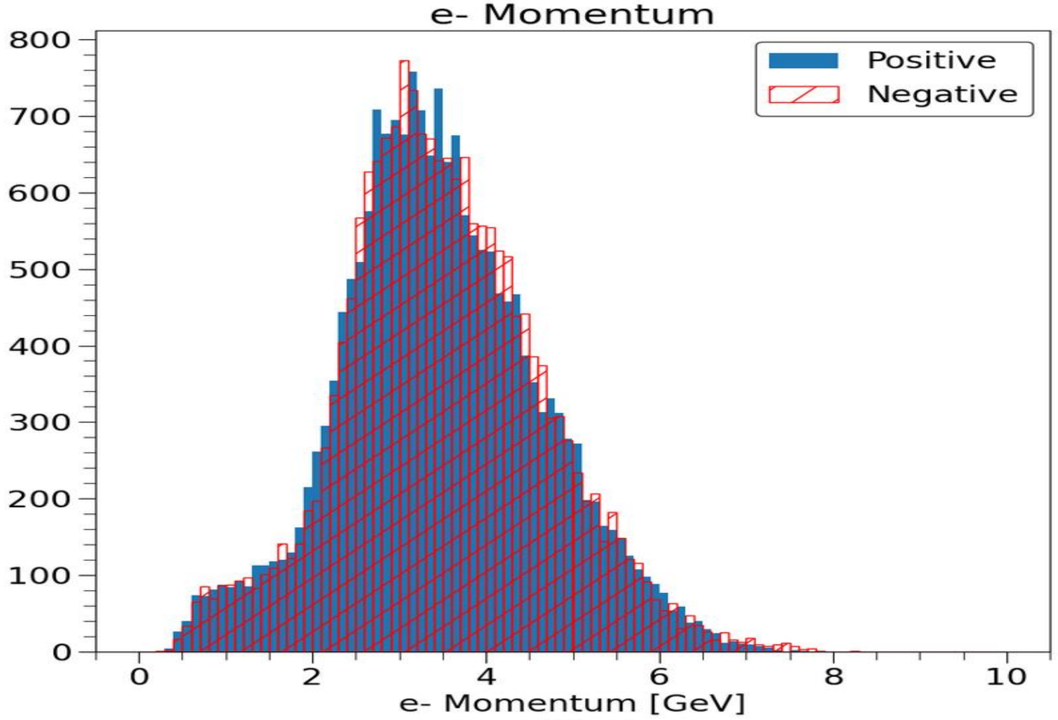


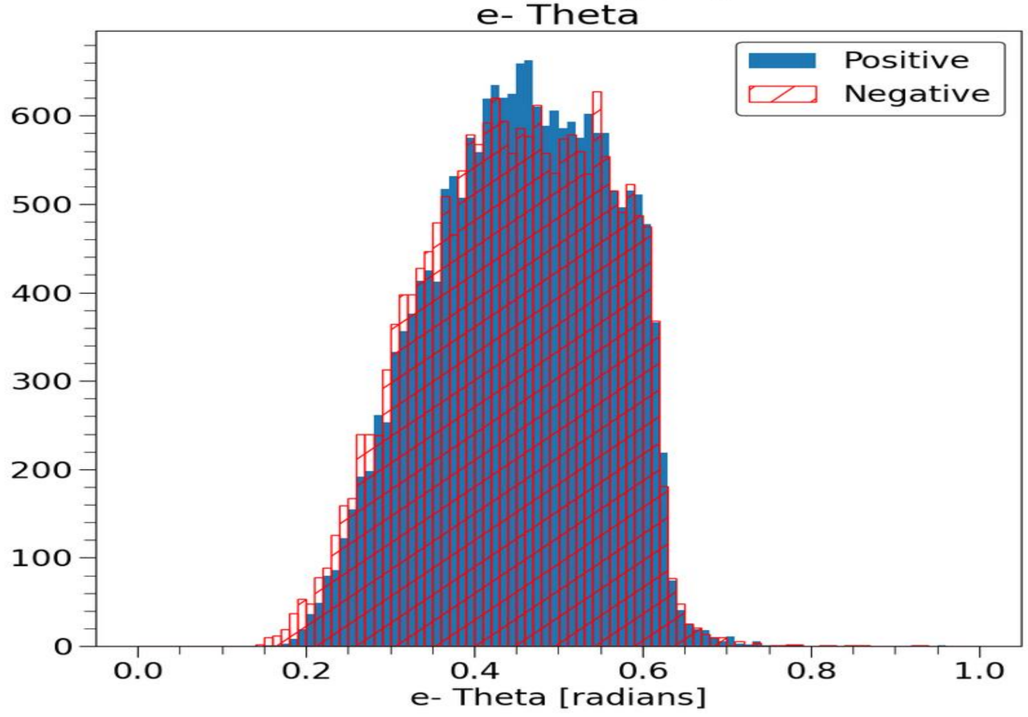
AI Reaction ID

- Here we do something different from muon ID where we train the classifier to recognize specific reactions, therefore with resonances in the simulations.

- However, the same principle still applies in that don't want the classifier to learn the P/Theta/Phi of individual particles.

- We can enforce this by creating a negative sample by mismatching particles from different events of the positive sample





Kaon/pion ID

- Ideally, we'd want to ID both reactions and particle type at the same time. For example, to distinguish between:
  - e' K+ K- p
  - e' pi+ pi- p
- We simulated training samples with ("fake") resonances in the K+ K- or pi+ pi- invariant mass.
- Gave as many variables as possible, containing P/Theta/Phi, vertex info, and info from various subsystems such as time of flight counters, calorimeters, HTCC/LTCC...
- The work here was done by undergraduate students for their final year projects.

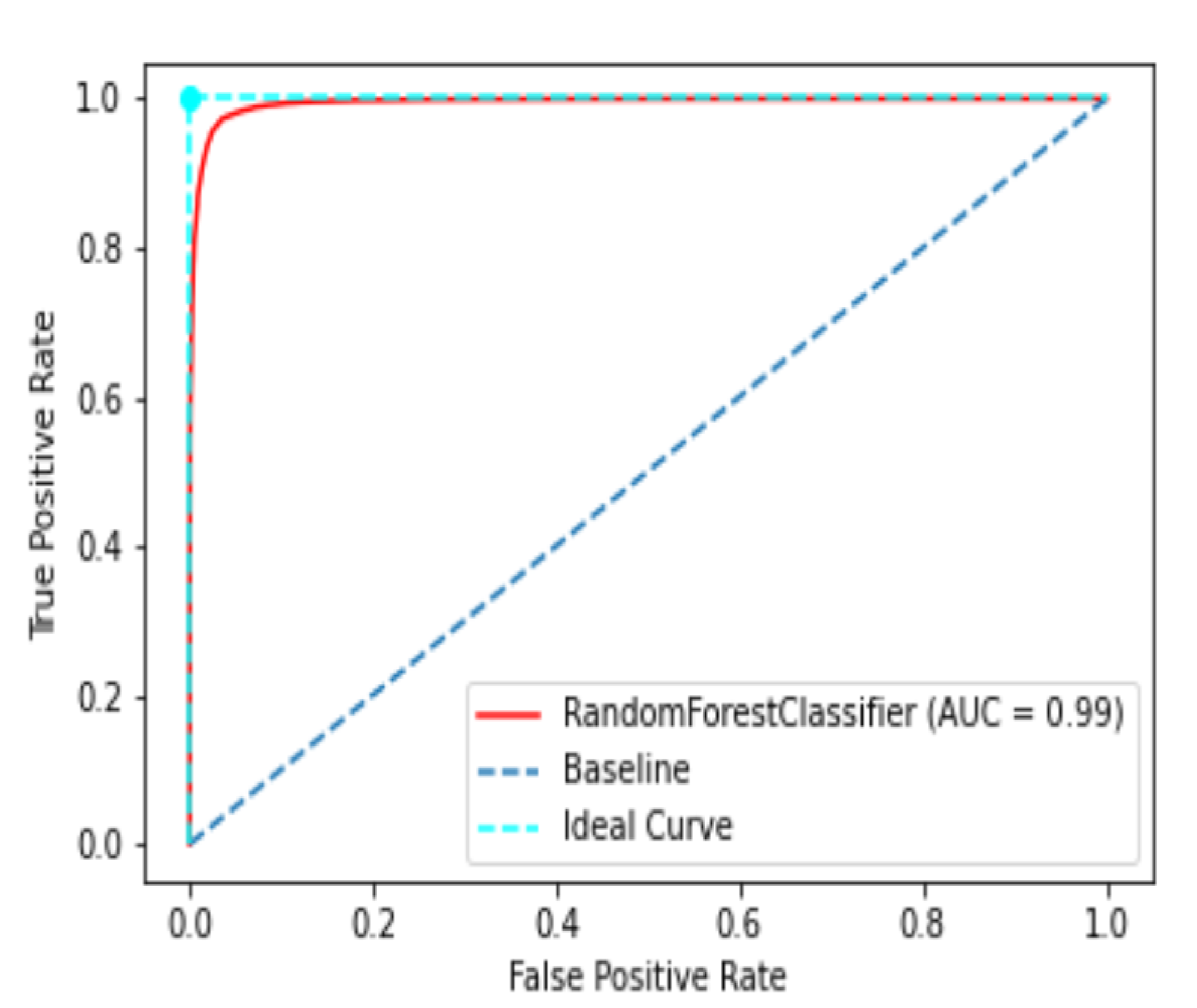
Feature Ranking:

K-/pi- P>3 | K+/pi+ P>3 | both P>3GeV | No Cuts

	Negative particles	Positive particles	Both Particles	No cuts > 3GeV
0	parpDeltaTimeVer	parpHTCC	parpHTCC	parmDeltaTime
1	parpDeltaTime	parmDeltaTimeVer	parpDeltaTimeVer	parmDeltaTimeVer
2	parmHTCC	parmDeltaTime	parmDeltaTimeVer	parpDeltaTimeVer
3	parpToF	parmToF	parmHTCC	parpToF
4	parpTheta	parpDeltaTimeVer	parmDeltaTime	parpDeltaTime
5	parmDeltaTimeVer	parmPath	parpDeltaTime	parmToF
6	parmDeltaTime	parmTheta	parpToF	parpHTCC
7	parpVt	parpDeltaTime	parmToF	parpTheta
8	parmToF	parpP	parpTheta	parmPath
9	parpP	parpToF	parpSF	parmHTCC
10	parmVt	parmLU	parpP	parpP
11	parpHypTime	parpTheta	parmP	parmTheta
12	parpPath	parmVz	parpLU	parmHypTime
13	parpLU	parmHypTime	parmHypTime	parmP
14	parpVz	parmP	parmPath	parmVz
15	parmHypTime	parpLU	parmLU	parpHypTime
16	parmP	parmLV	parpLV	parmLU
17	parmPath	parpSF	parmTheta	parmVt
18	parpHTCC	parmSector	parpLW	parpPath
19	parmLU	parmLW	parpEECin	parpVt

Kaon/pion ID

- Achieved ~96% accuracy compared to ~74% accuracy for the EB PID.
- Main conclusions were that gradient boosted decision trees and 1D convolutional networks worked best.
- Haven't tried this on Clas12 data yet.



Kaon/pion ID

- Achieved ~96% accuracy compared to ~74% accuracy for the EB PID.
- Main conclusions were that gradient boosted decision trees and 1D convolutional networks worked best.
- Haven't tried this on Clas12 data yet.