ANN PID Retuning for Reco14 Data

C. Jones

Outline

- Introduction
- Training Parameters
- TMVA Performance
- TMVA / NeuroBayes Comparison
- "MC12TuneV2" Performance
- Conclusions

Combined DLLs

- Combined DLLs introduced some time ago.
  - Designed as a way of combining the information from multiple sub-systems.
  - Each sub-system provides log-likelihood information (differences) on the particle species it is able to distinguish.
- These delta log likelihoods are then added together to form the "Combined DLLs".
- Simple and easy to understand. But do have some limitations.

Combined DLLs - Drawbacks

- Assumes all PID information can be presented as a likelihood.
  - Excludes information like Muon # shared hits, sub-detector acceptance flags etc.

- Summing the sub-detector DLLs relies on them being mathematically well defined (i.e. log of product of probability values).
  - This of course should be the case, but in practice does not always work out. Different sub-detectors have different 'dynamic ranges'.
    - RICH DLLs have to be rescaled to match the CALO and MUON values, when forming the electron and muon CombDLLs.
  - Different range of values for the various CombDLLs

- DLLs can have 'funny' features (such as the spike at DLL(K-π)=0 when both K and π are below the Cherenkov threshold).

- Not the most powerful way of combining information.

Solution - Multivariate PID

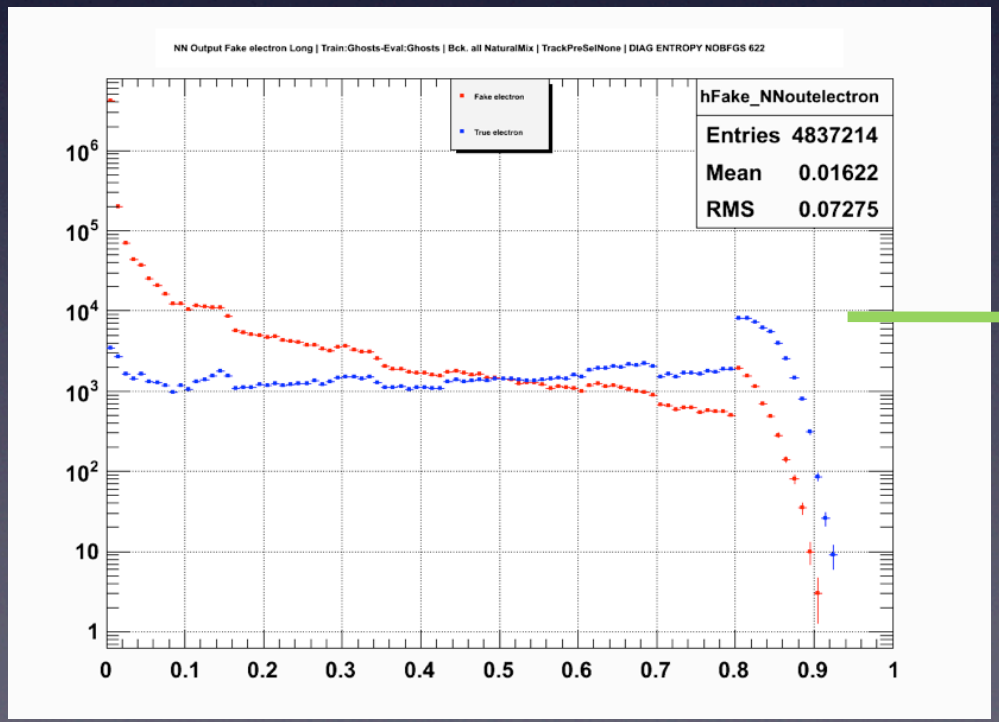
- Can use any (and all) separating variables, integer or floating point.
- Can utilize non-PID information (track variables) through correlations.
- For each track provide a consistent set of probability values for each particle species.

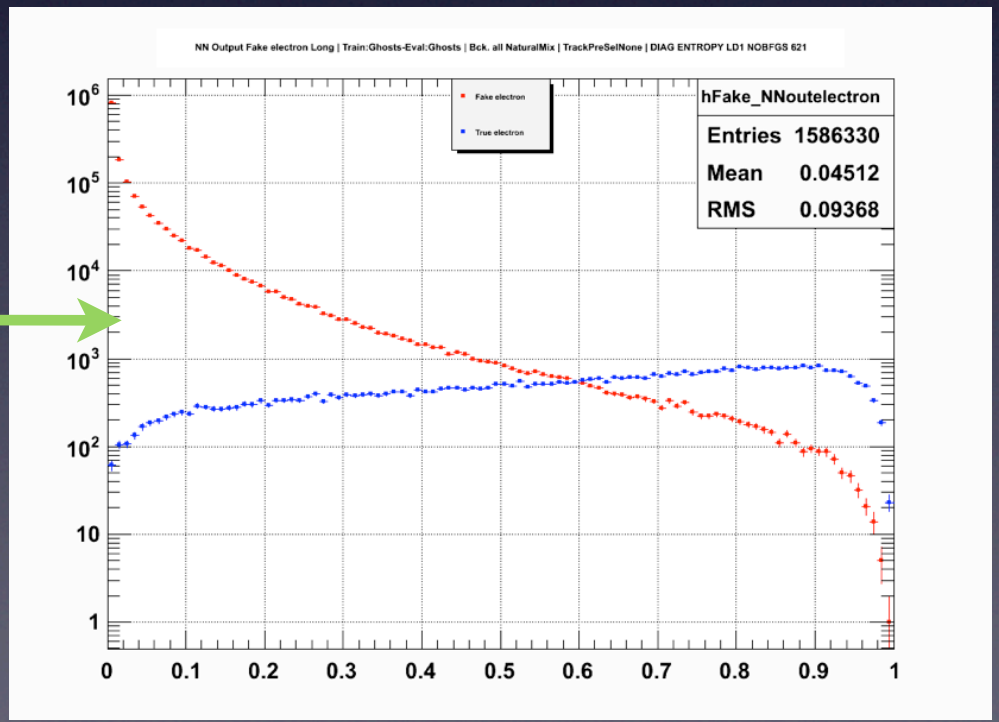
- Finally, should give ***better PID performance***

ANN PID History

- First introduced in Reco12/Stripping17
  - Training based on the NeuroBayes library

- Retuned for Reco14 (“MC12TuneV1”) using some preliminary Reco13<x> data.
  - Changed NeuroBayes parameters, activating boosting, as this had the effect of ‘smoothing’ the output distribution.





NeuroBayes Issues

- Problems observed with "MC12TuneV1" during Reco14 processing.
  - Jobs would terminate unexpectedly. Eventually traced to NeuroBayes library calling "exit(1)".
  - NB 'boosted' mode suffers from numerical precision issues.
    - Uninitialized variables, output values unstable.
    - Problem confirmed by 'BlueYonder'. Fix promised, but still waiting...
- Always intended to consider a new tuning Reco14 ("MC12TuneV2") once official MC samples where available.
- Decided to extend training to use TMVA as well as NeuroBayes as the backend training engine, to perform a direct comparison.

Training

ANN Training

- Training done using MC inclusive B events.
  - All tracks in the events considered, including ghosts (*).
  - Events resampled based on event occupancy, to match data. ***New to MC12TuneV2.***
  - Independent training and test samples.
  - Separate networks trained for Long, Downstream and Upstream tracks.
  - Separate electron, muon, pion, kaon, proton and ghost networks
- Many MVA alternatives considered.
- Many different ways ways of looking at the performance
  - Eff. v {purity, <type>msID, overall misID}
  - Eff. and misID v. $\{P,P_t\}$ for various cut values
  - Too many plots to show here. Representative selection shown.
  - http://www.hep.phy.cam.ac.uk/~jonesc/lhcb/PID/ANN/MC12TuneV2/

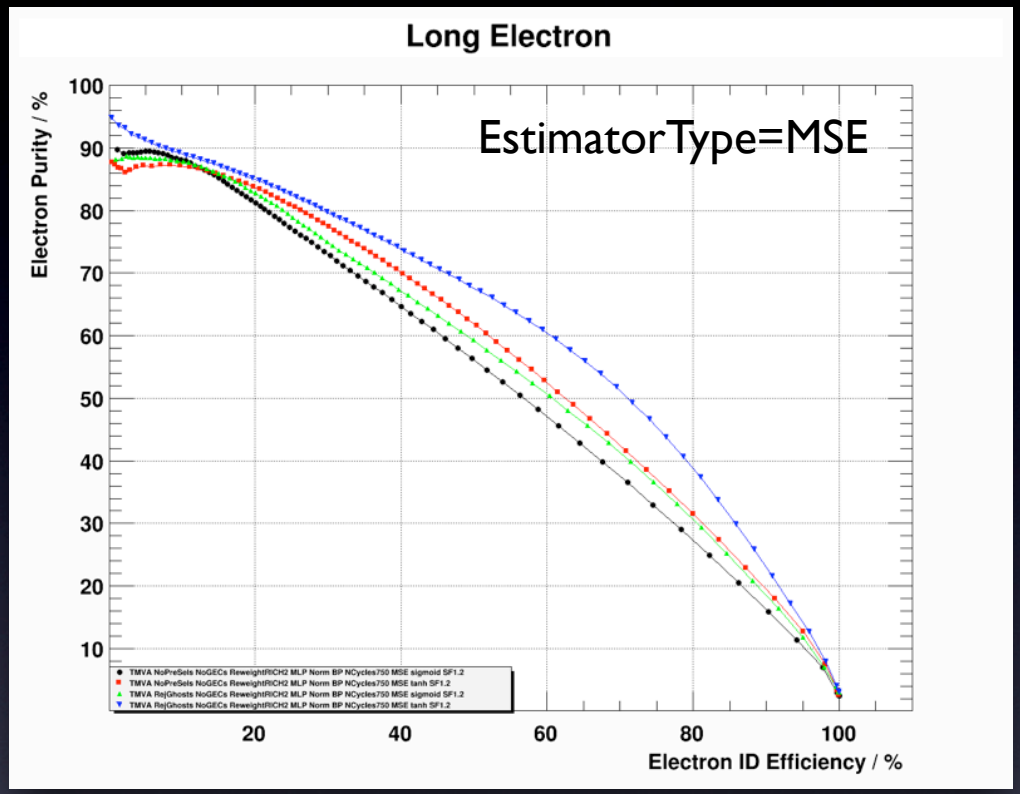
* Muon networks trained with IsMuon=True preselection

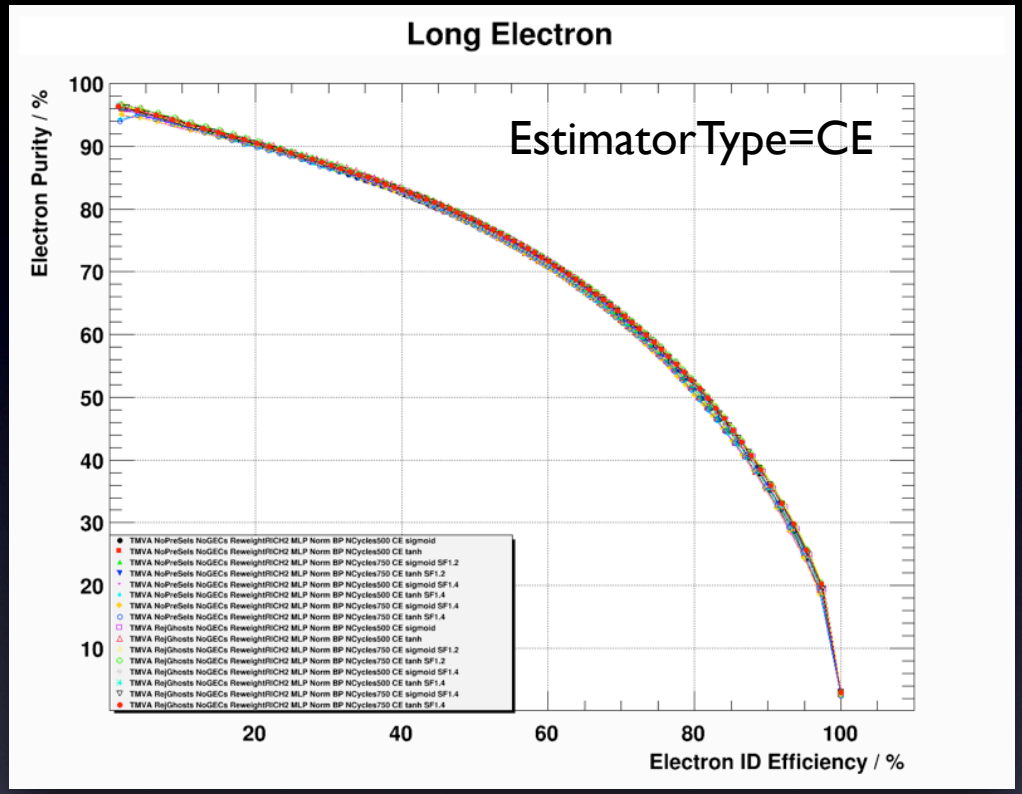
Input Variables (Long Tracks)

Tracking	TrackP, TrackPt, TrackChi2PerDof, TrackNumDof, TrackLikelihood, TrackGhostProbability, TrackFitMatchChi2, TrackCloneDist, TrackFitVeloChi2, TrackFitVeloNDoF, TrackFitTChi2, TrackFitTNDoF
RICH	RichUsedAero, RichUsedR1Gas, RichUsedR2Gas RichAboveMuThres, RichAboveKaThres, RichDLLe, RichDLLmu, RichDLLk, RichDLLp, RichDLLbt
Muon	MuonBkgLL, MuonMuLL, MuonIsMuon, MuonNShared, InAccMuon, MuonIsLooseMuon
CALO	EcalPIDe, EcalPIDmu, HcalPIDe, HcalPIDmu, PrsPIDe, InAccBrem, BremPIDe
VELO	VeloCharge

TMVA Performance

TMVA MLP - Electrons

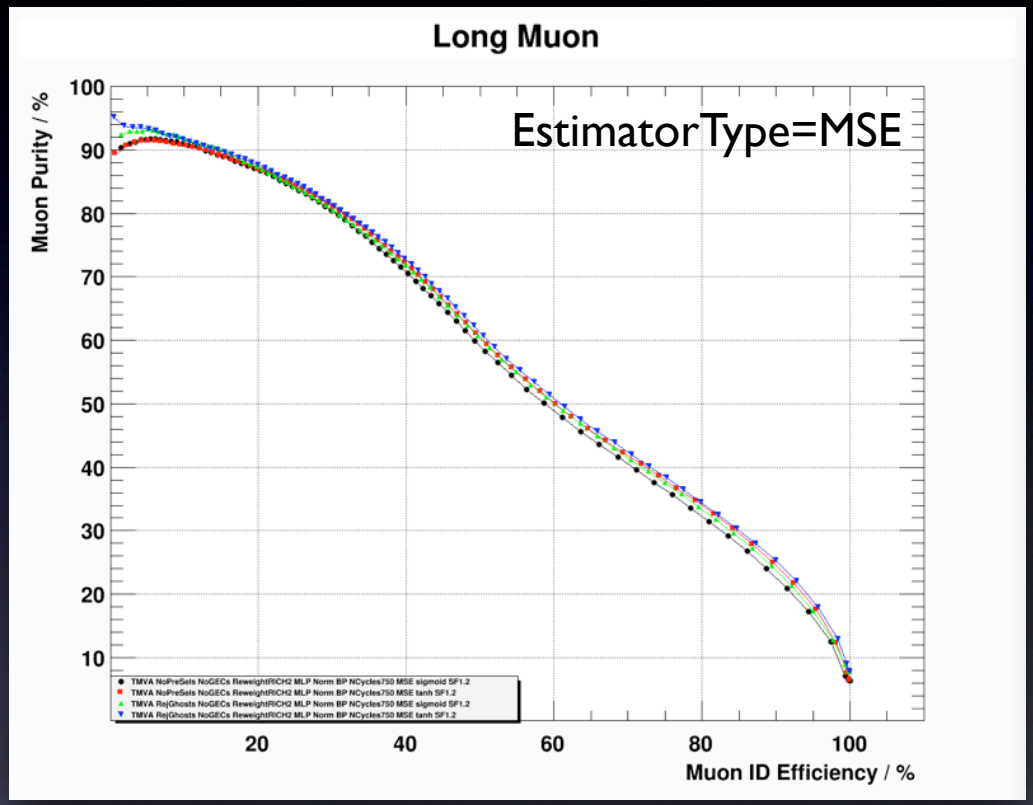


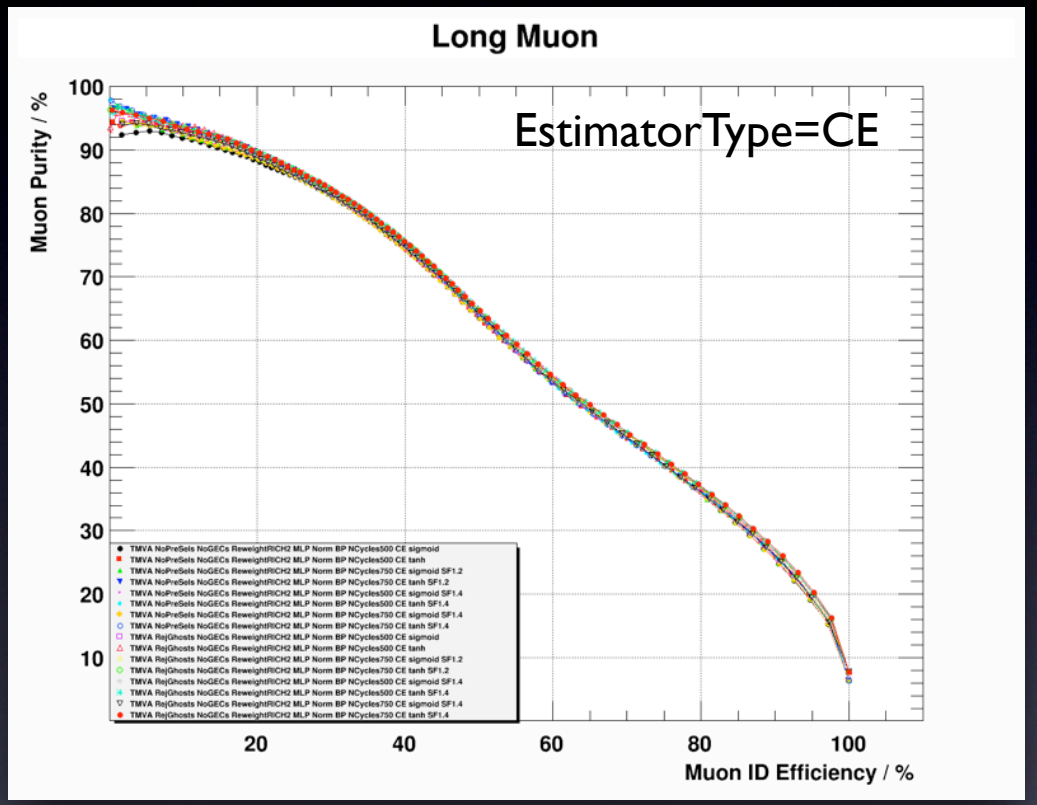


- CE (Bayesian) Estimator gives better performance overall
- More consistent performance between other training options
- MVA output using CE is a probability value.

TMVA MLP - Muons

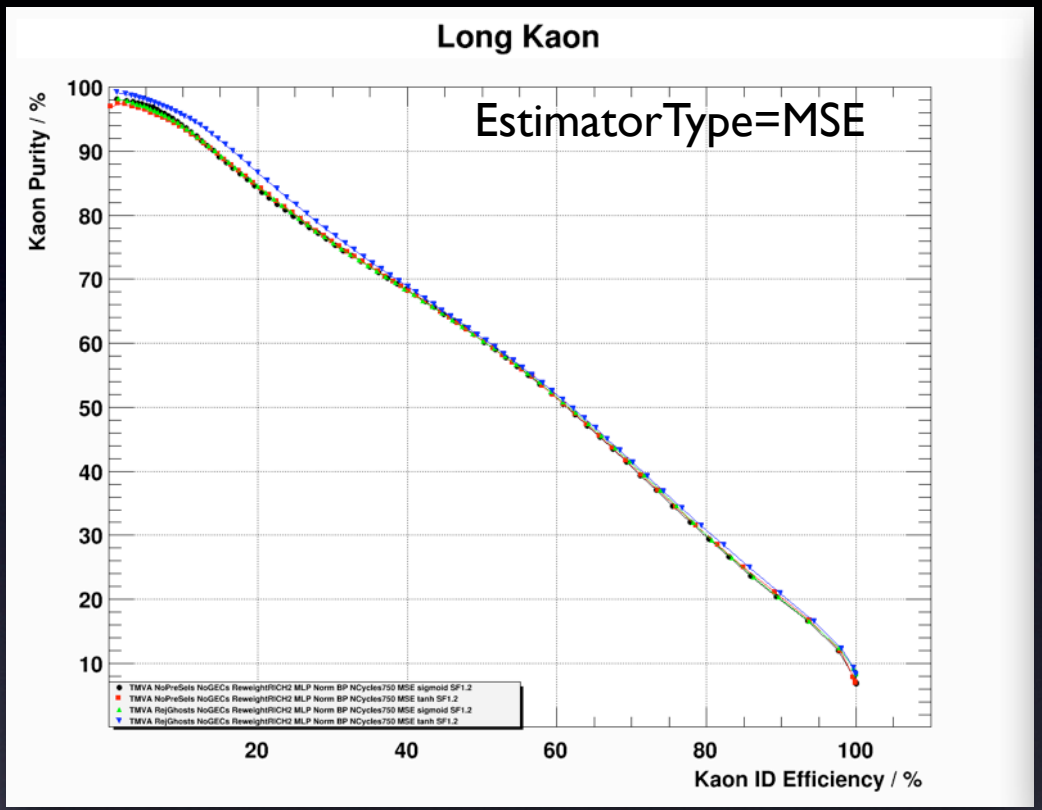
* Muon networks trained with IsMuon=True preselection

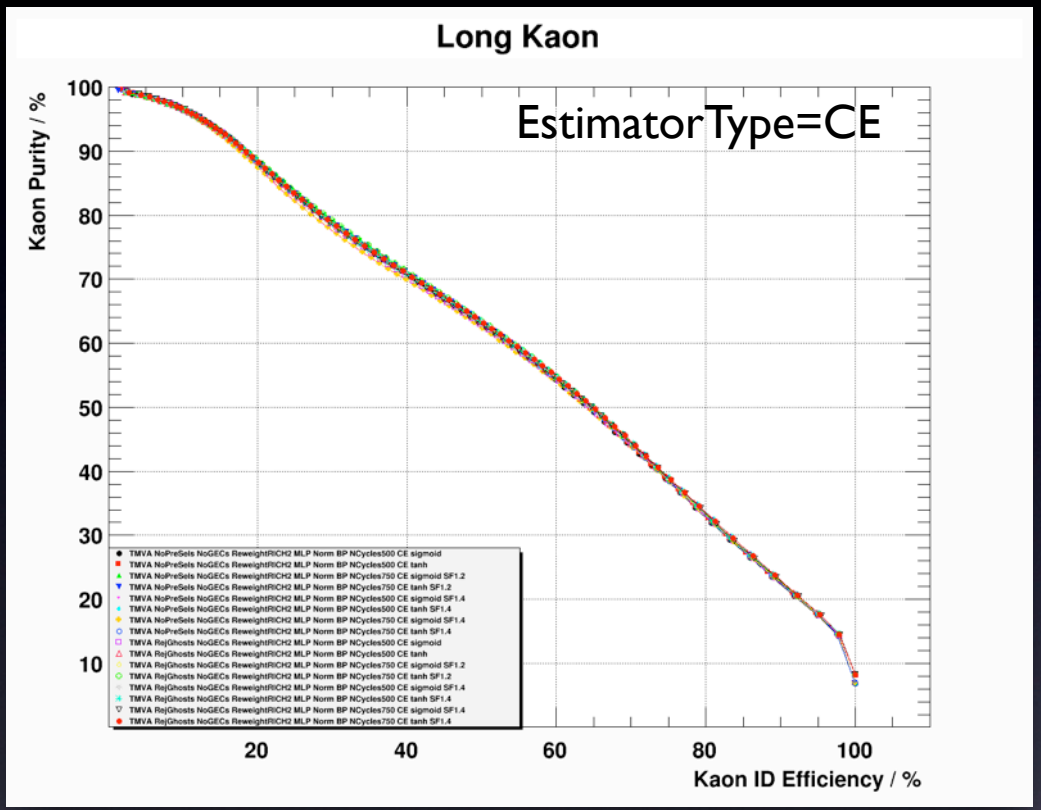




- CE still marginally better

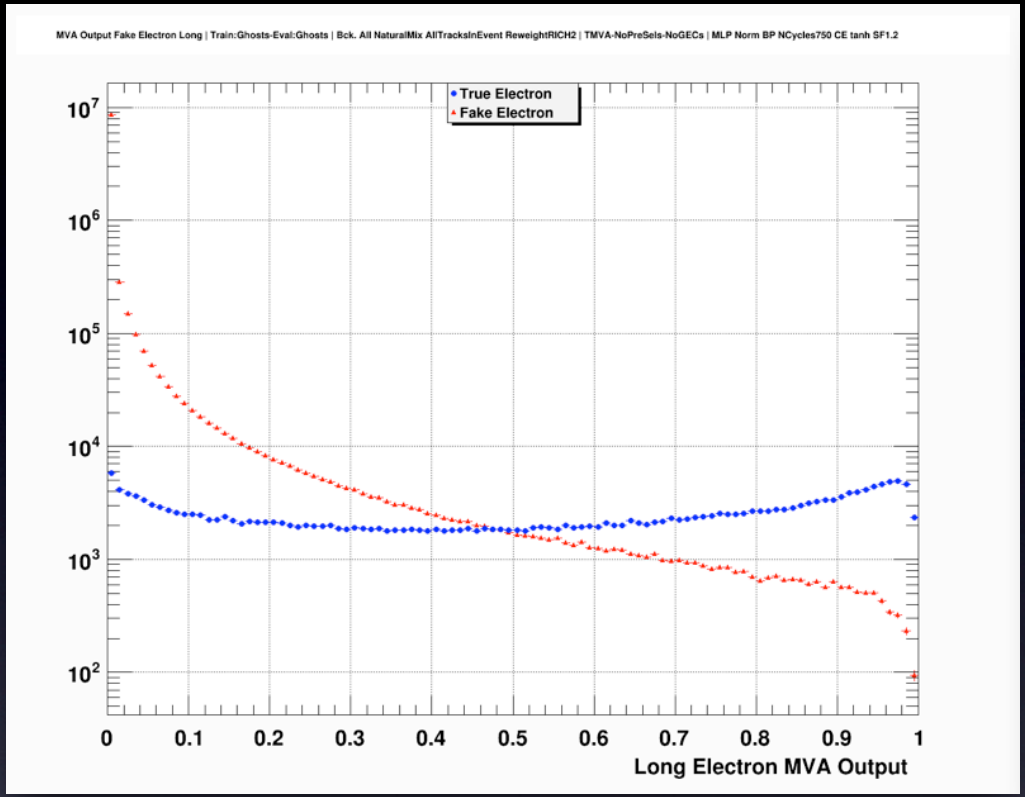
TMVA MLP - Kaons

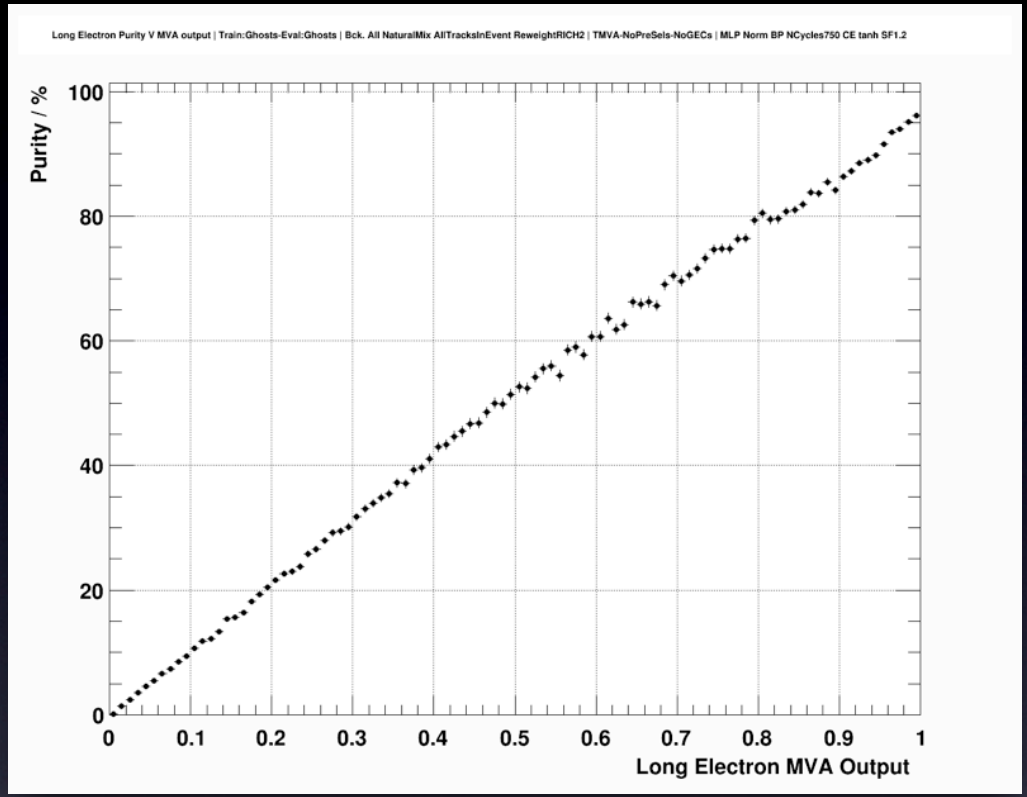




- CE still marginally better

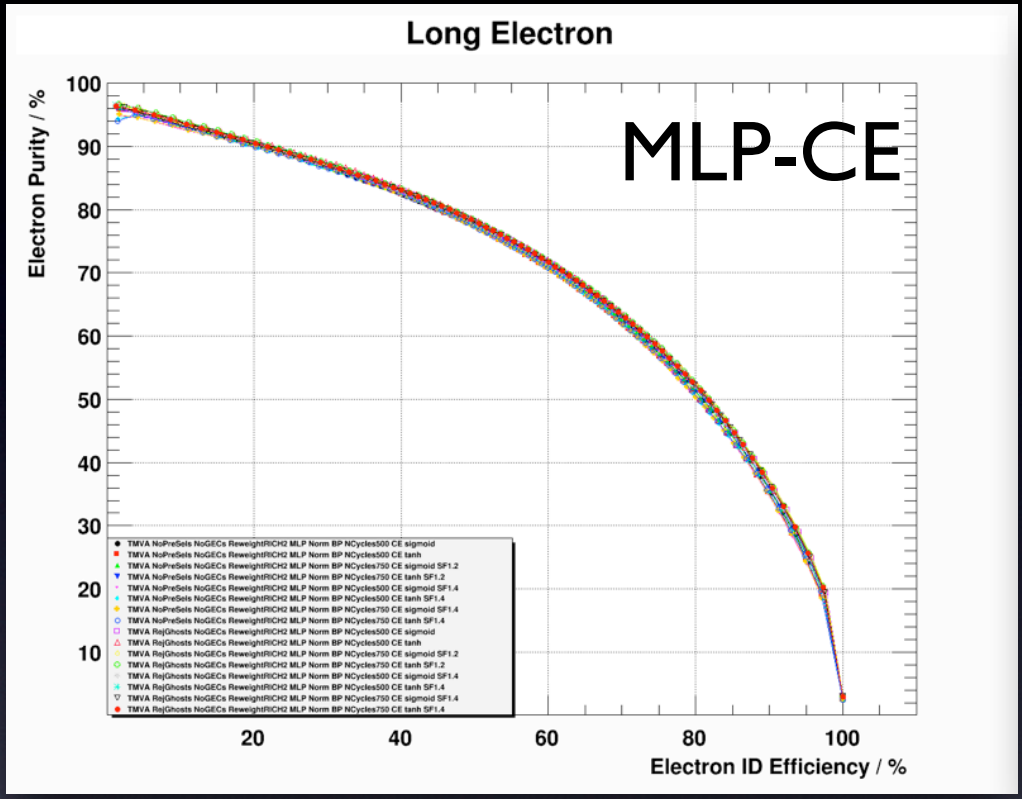
EstimatorType=CE - MVA Output

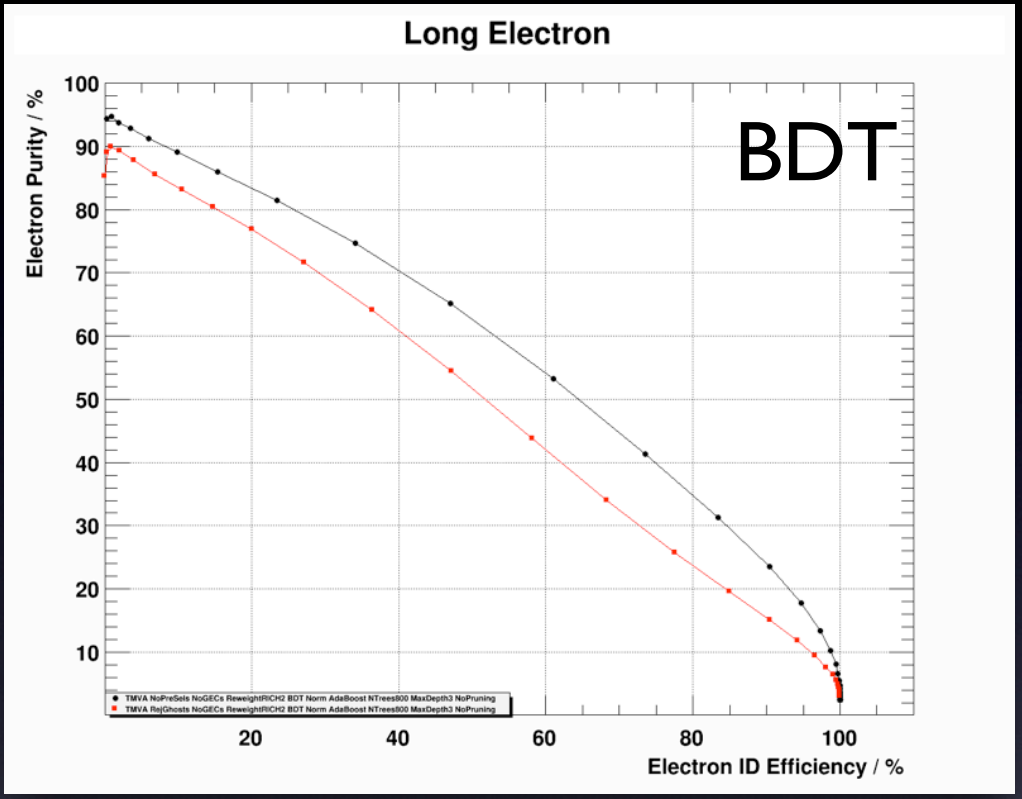




- Output strictly constrained to range {0,1} (not the case for MSE)
- Purity versus output value has form y=x

MLP versus BDT





- Similar story for other networks. MLP beats BDT.

TMVA Conclusions

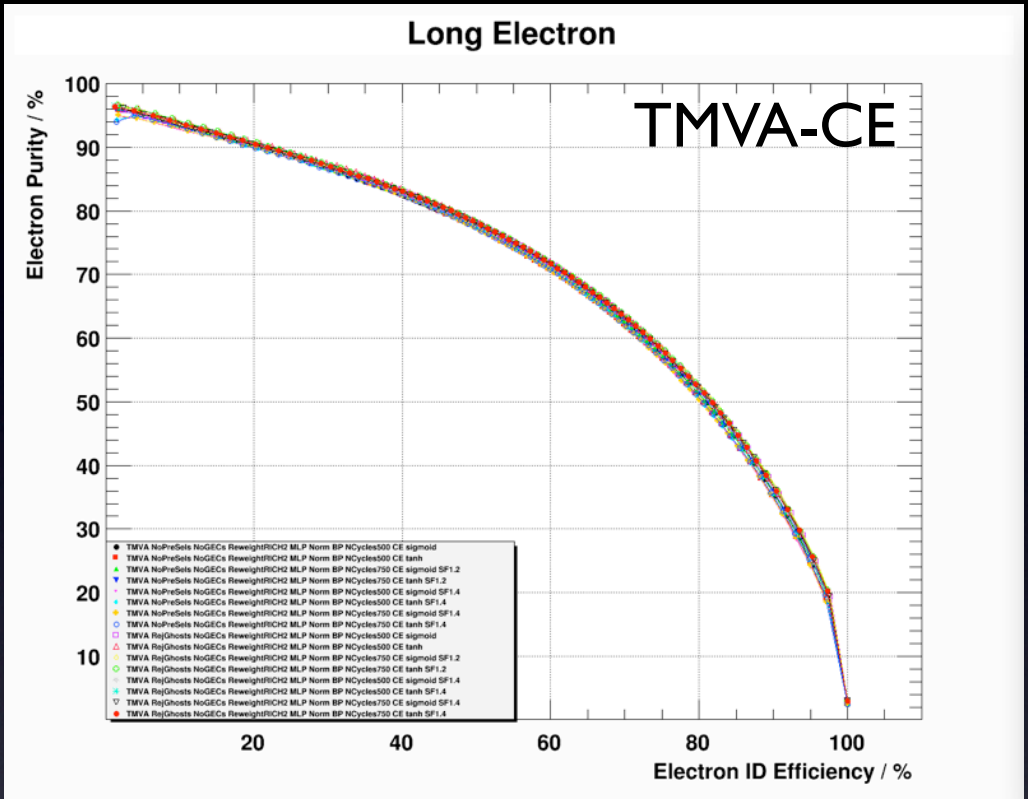
- TMVA's MLP with the CE Estimator the clear winner in this use case
  - Better performance, well defined output value.

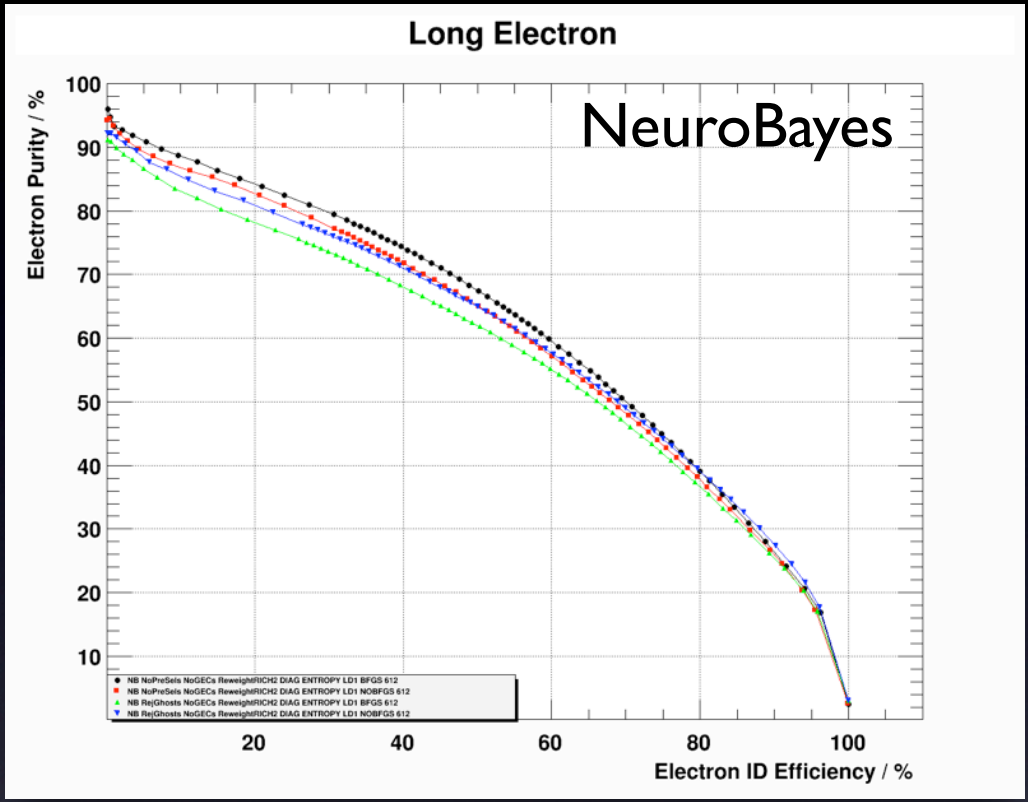
- CE Estimator is perhaps not as well known as it should be.
  - Not mentioned in the current manual (based on 4.0, added in 4.1)
  - Not mentioned in the example scripts
  - Implements a Baysian Network.
  - See http://arxiv.org/abs/1103.2854 for details.

- Recommend anyone using TMVA, who hasn't tried it to give it a go.

TMVA-CE Versus NeuroBayes

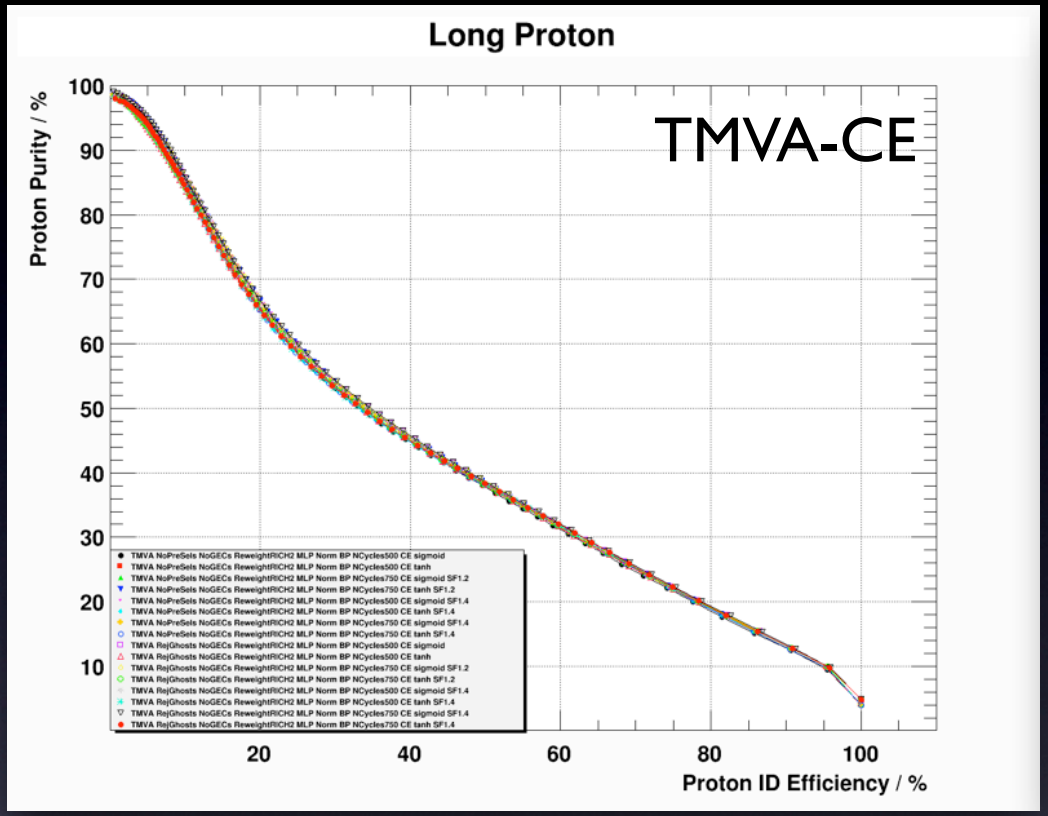
Electron

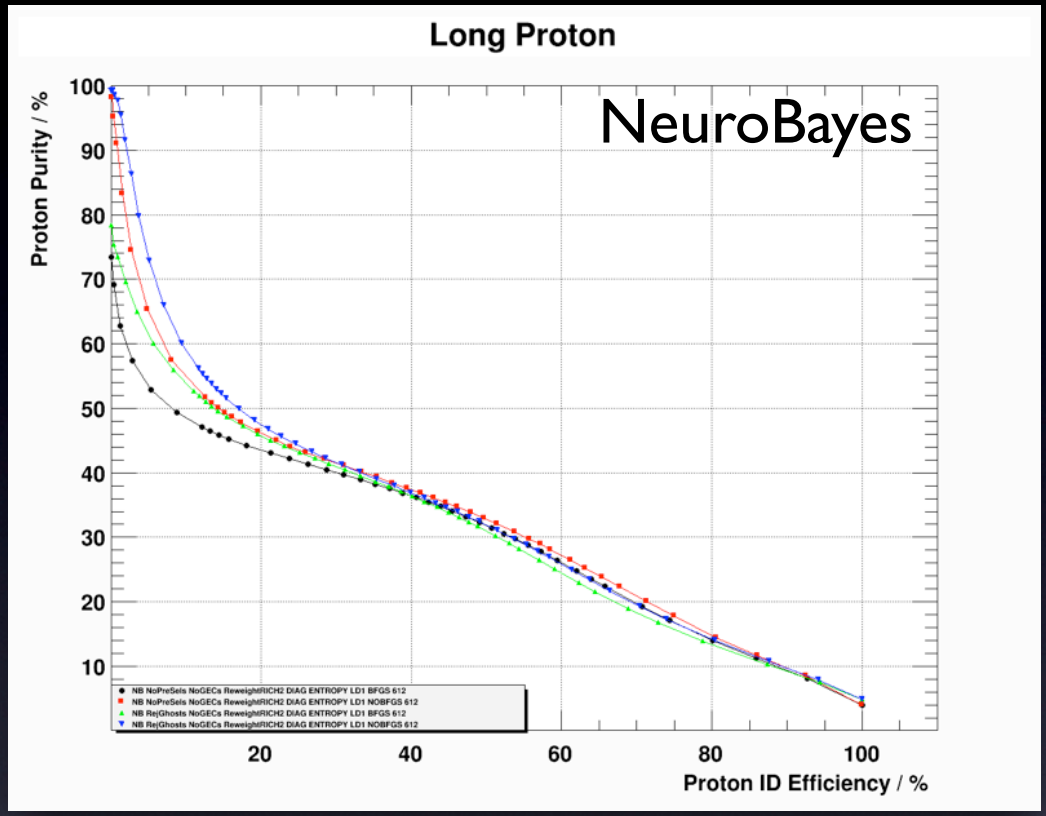




- TMVA-CE out performs NB

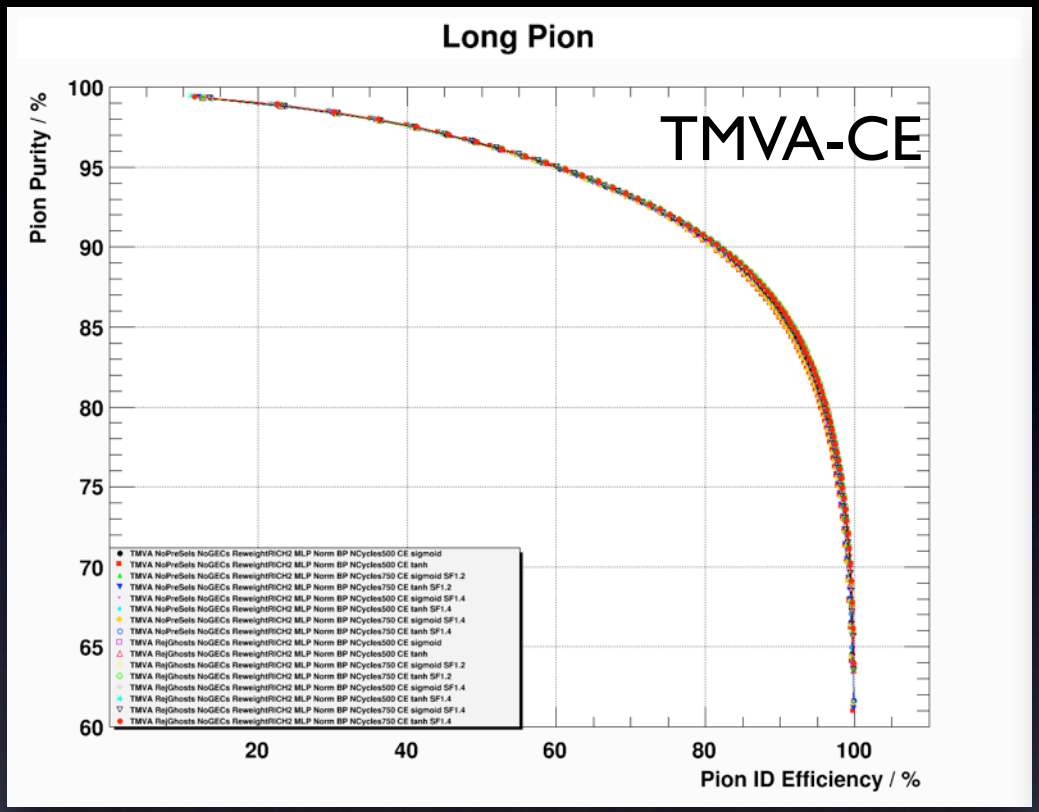
Proton

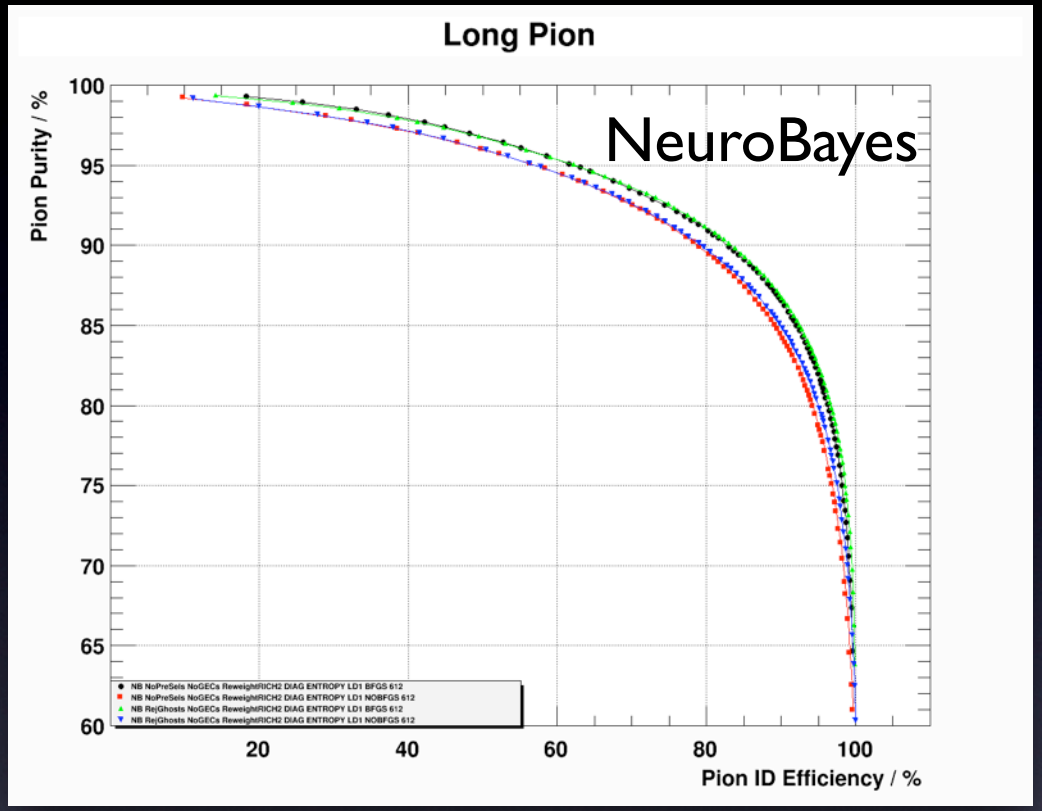




- Again, TMVA gives better performance

Pion





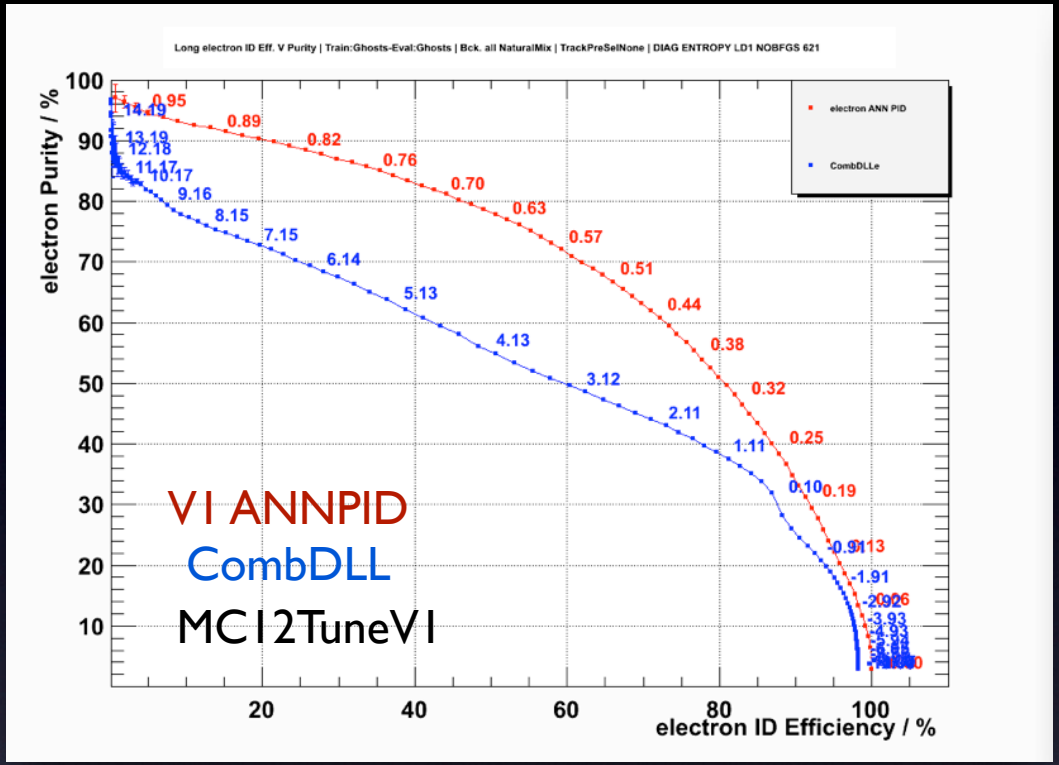
- Very similar performance in this case

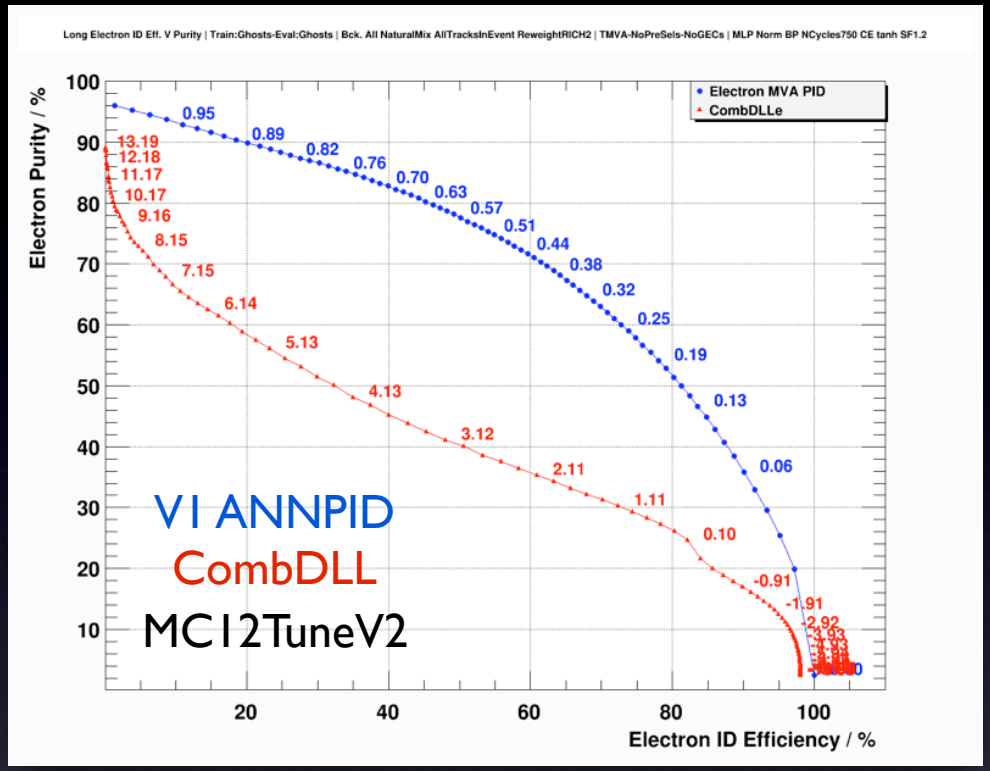
TMVA/NB Conclusions

- TMVA's MLP using the CE Estimator gives the best overall performance.
- Several practical reasons to prefer TMVA over NB
  - Easier debugging in a production environment
  - Much faster implementation :-
    - NB O(50ms/event)
    - TMVA Reader O(15ms/event)
    - TMVA Compiled C++ O(4ms/event)
  - Compiled C++ also removes any dependency on external libraries.
- ***Switch to TMVA MLP-CE networks for "MC12TuneV2" retuning***

“MC12TuneV2” Performance

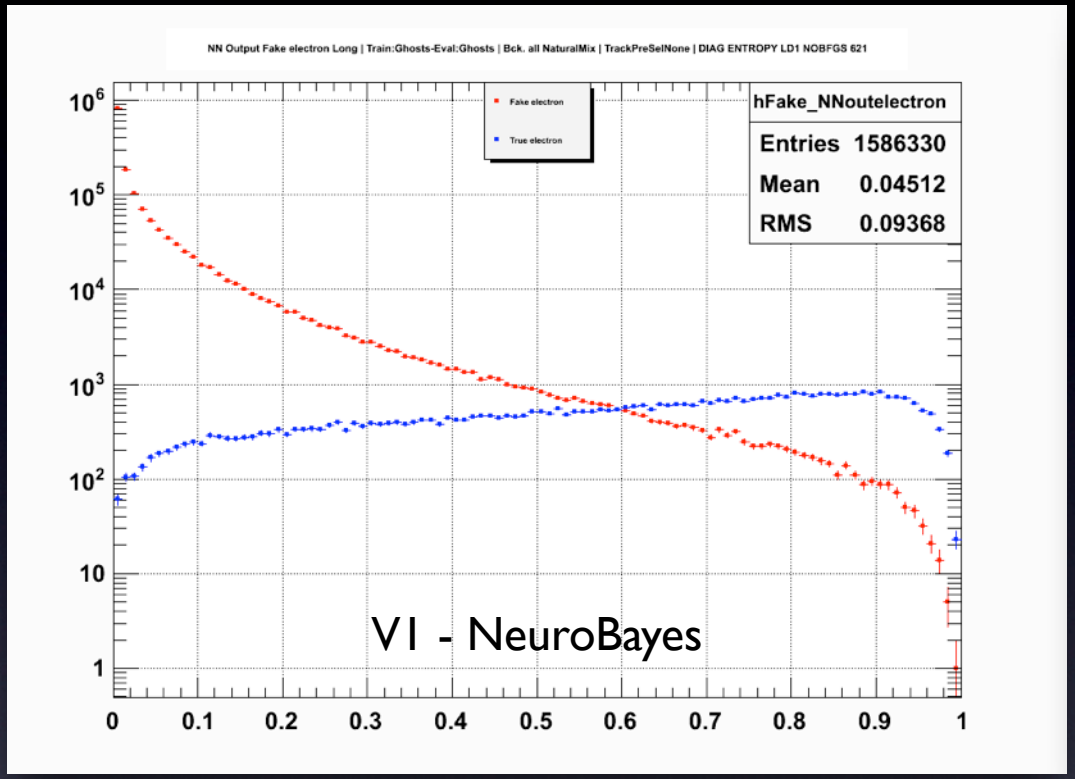
Electrons :V1 versus V2 Tuning

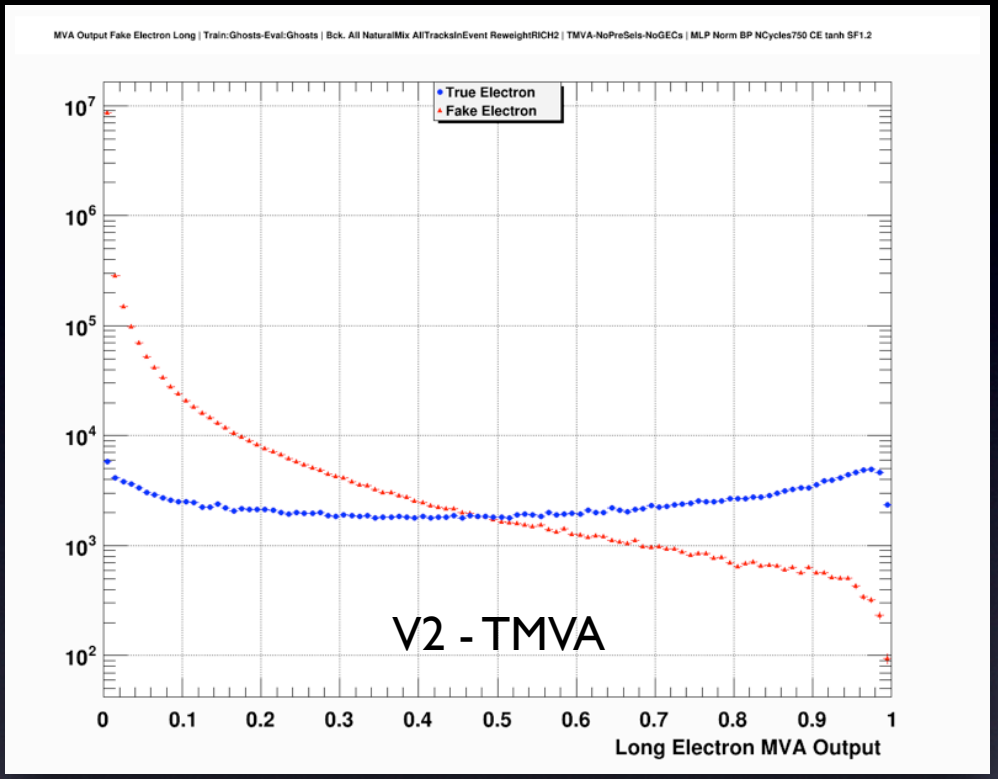




- ***Remember*** :V2 Tuning uses a resampled MC sample, to increase the event occupancies. Performance on average will be worse (as can be seen by comparing the DLL curves)
- **ANNPID performance better, compared to the DLLs in V2 to V1**
- ***NOTE*** : Due to differences in the output distributions. Different cut values required for similar eff. working points.

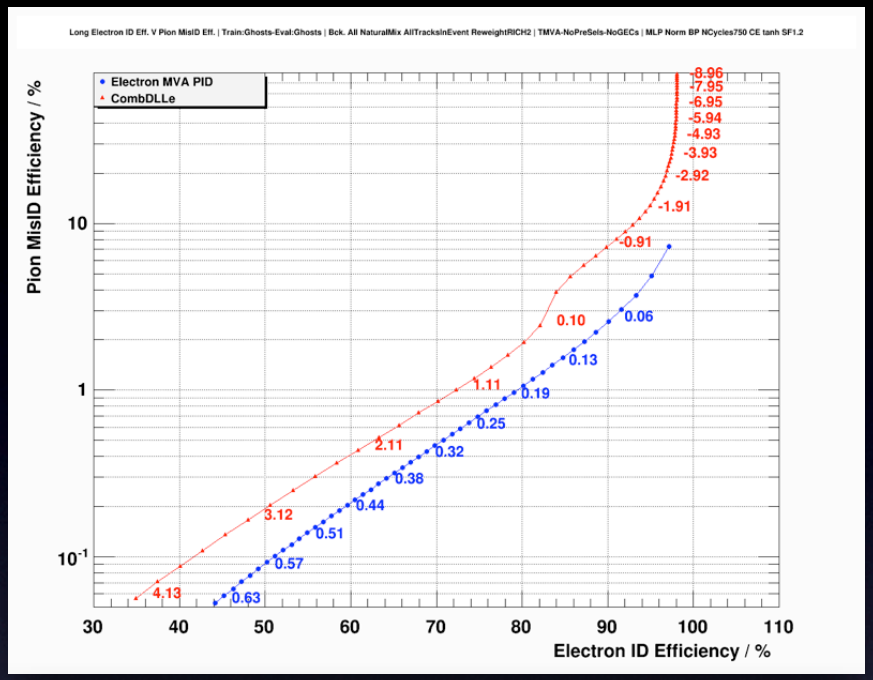
Electron MVA Output

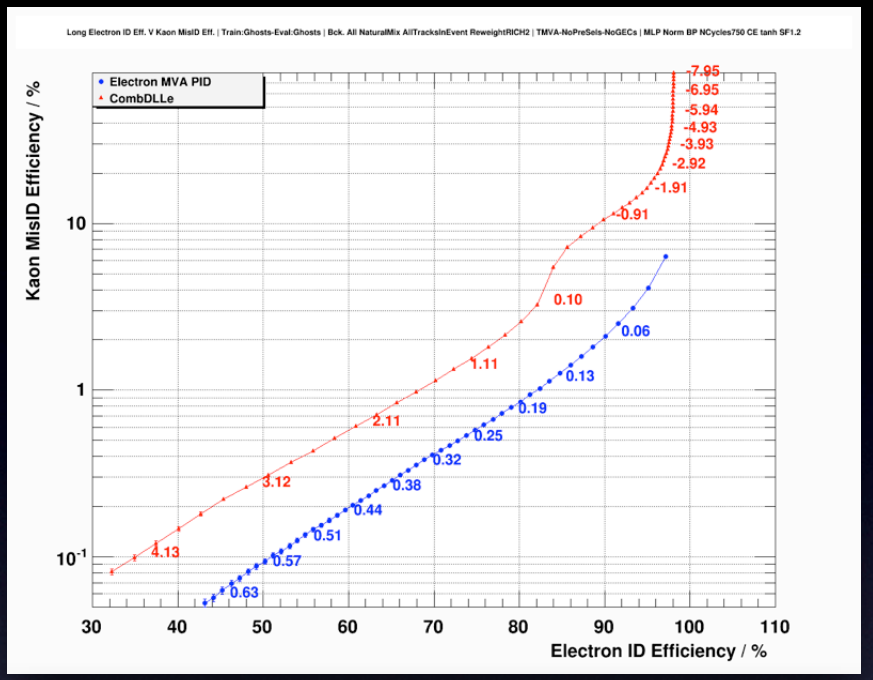


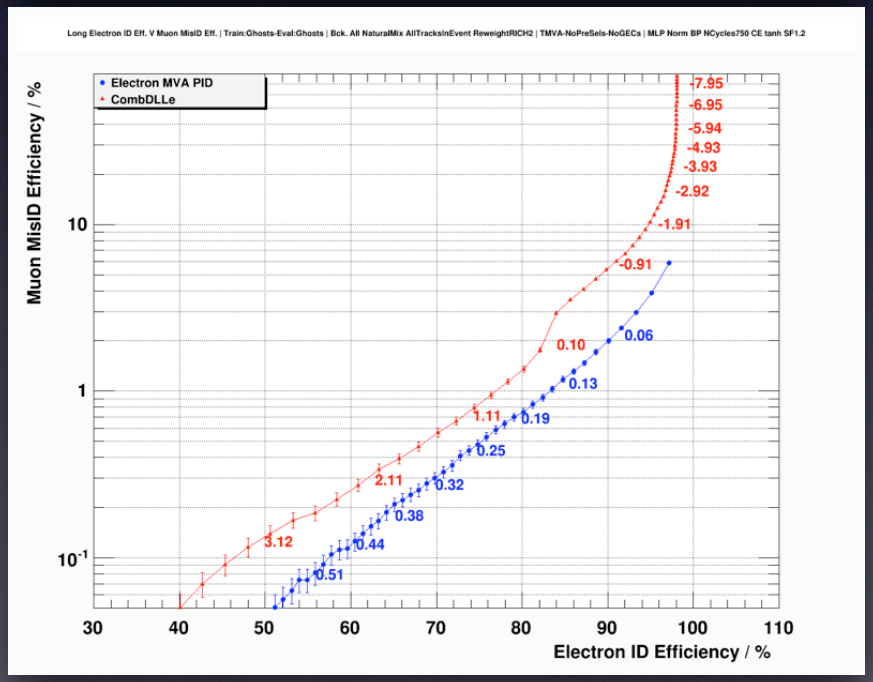


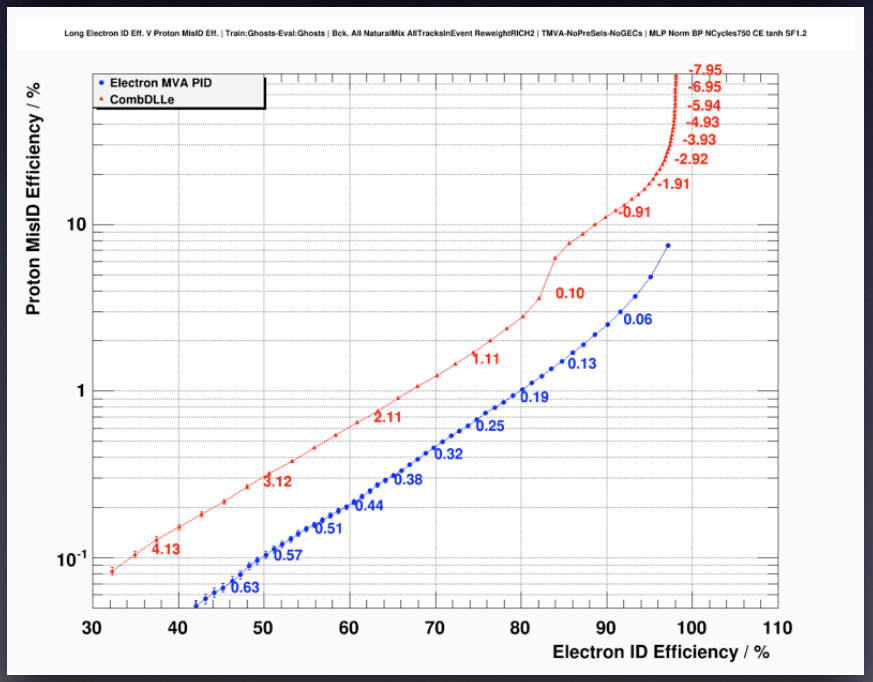
In general, looser cuts are required with TMVA

Electrons - Efficiency V MisID

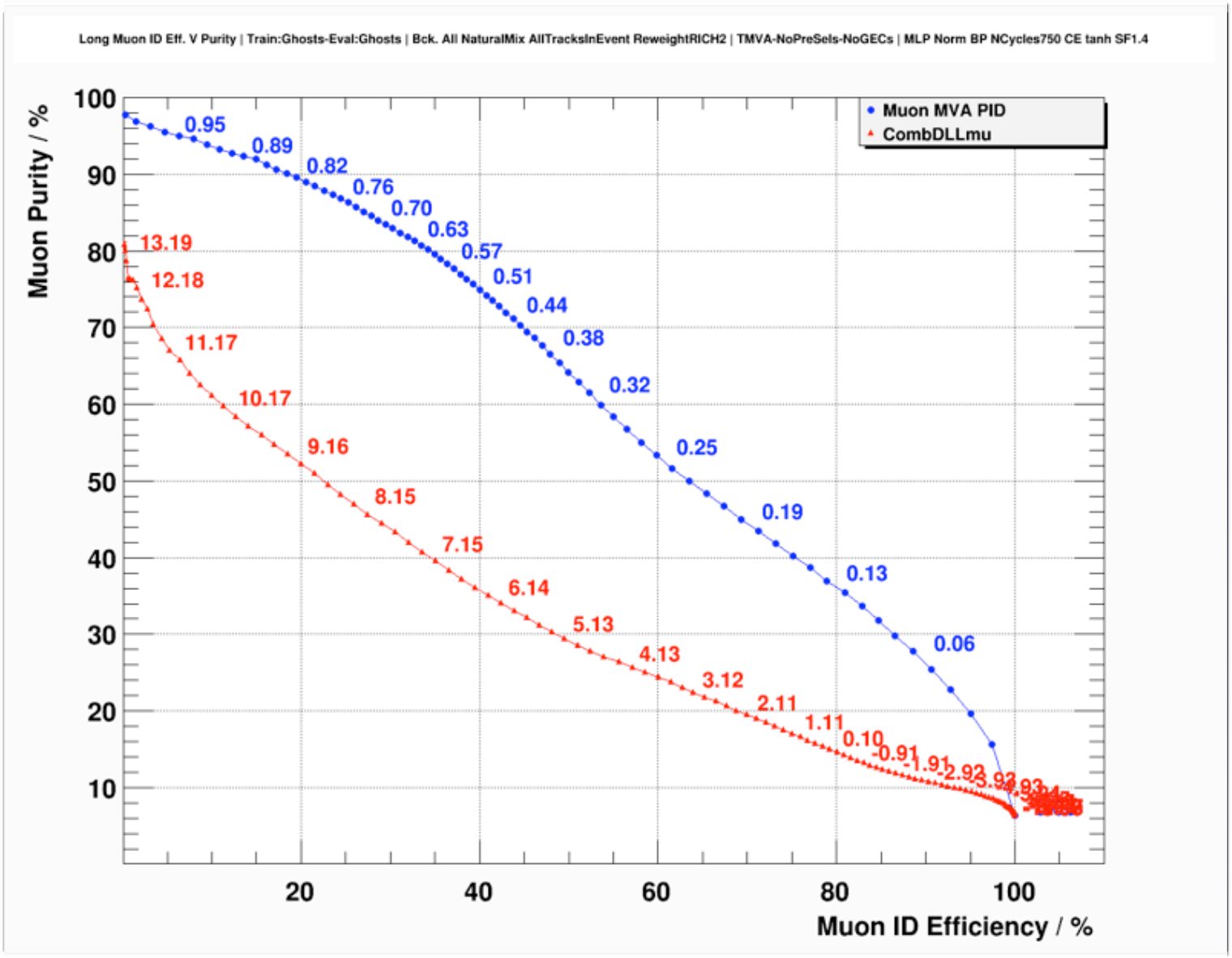






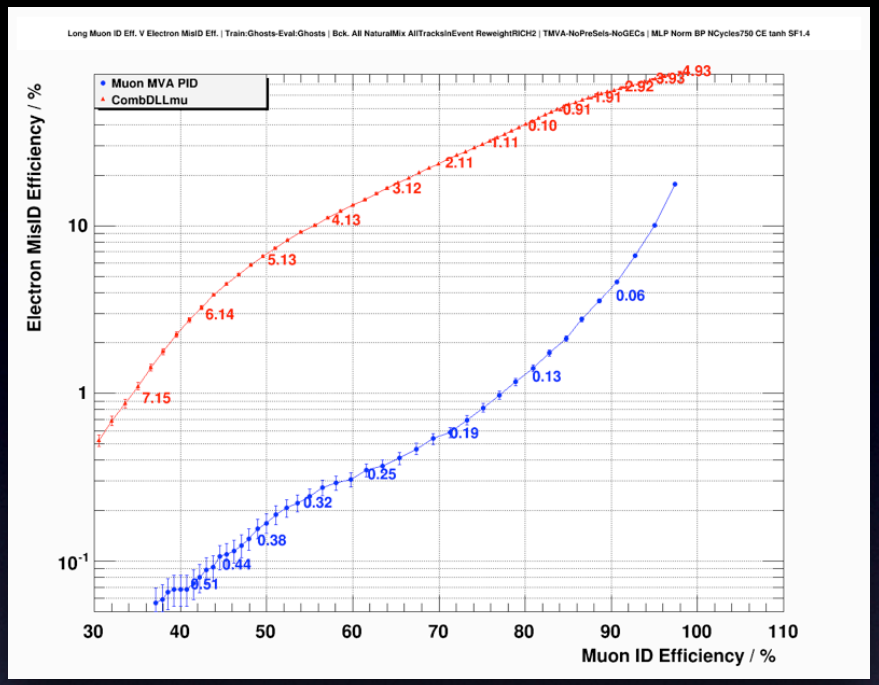


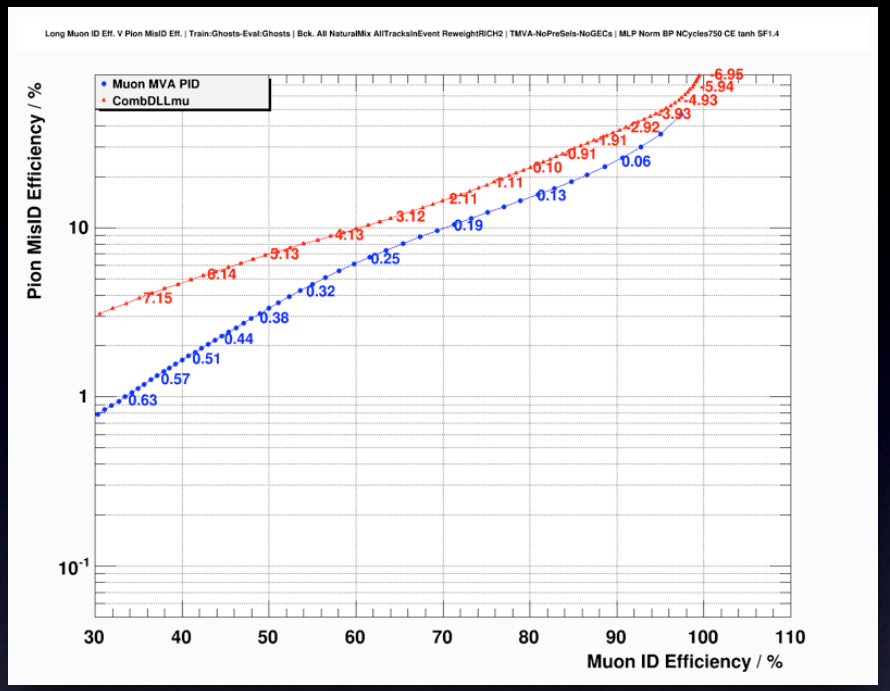
Muons - Efficiency V Purity

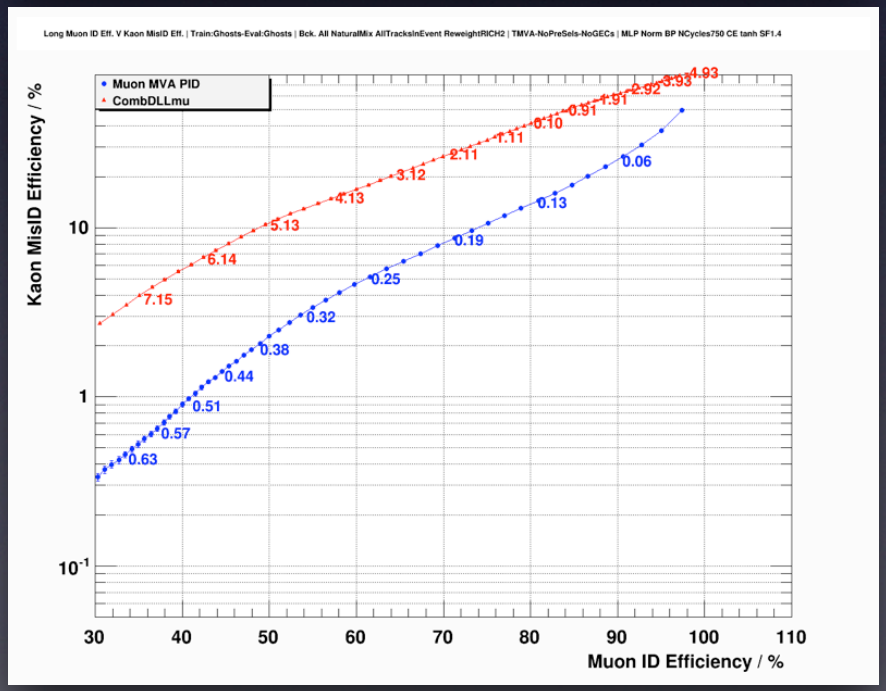


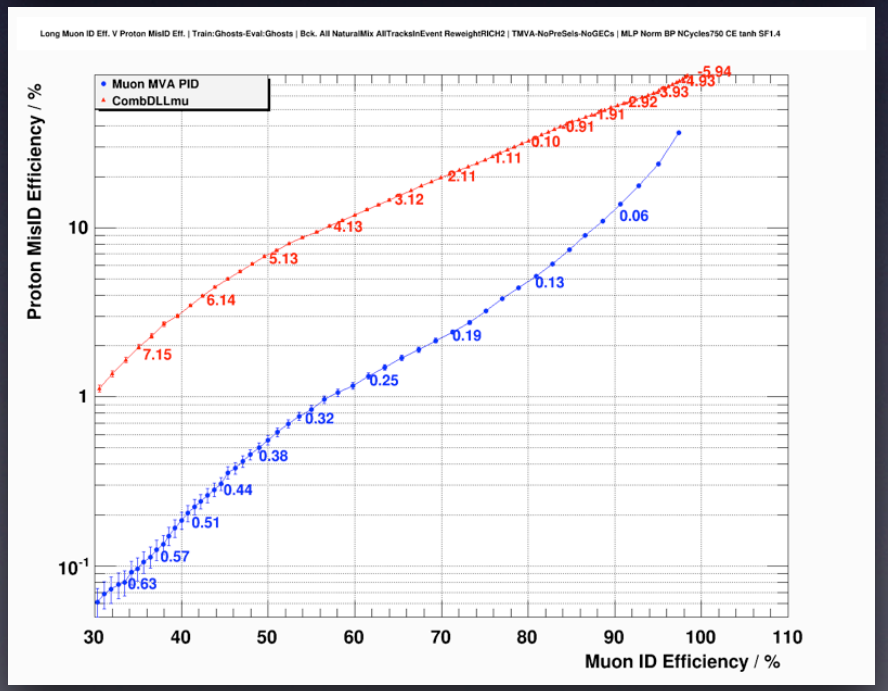
* Muon networks trained with IsMuon=True preselection

Muons - Efficiency V MisID

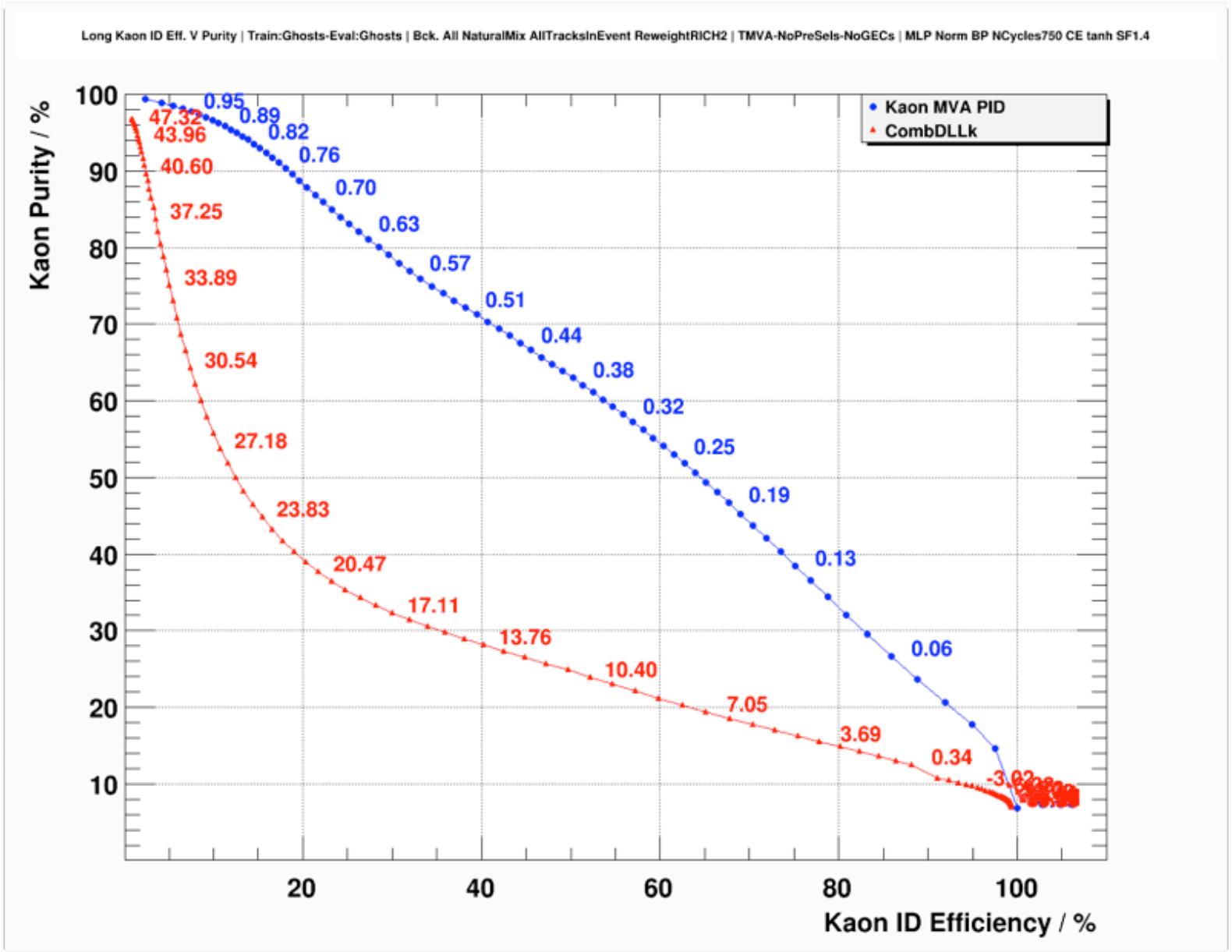




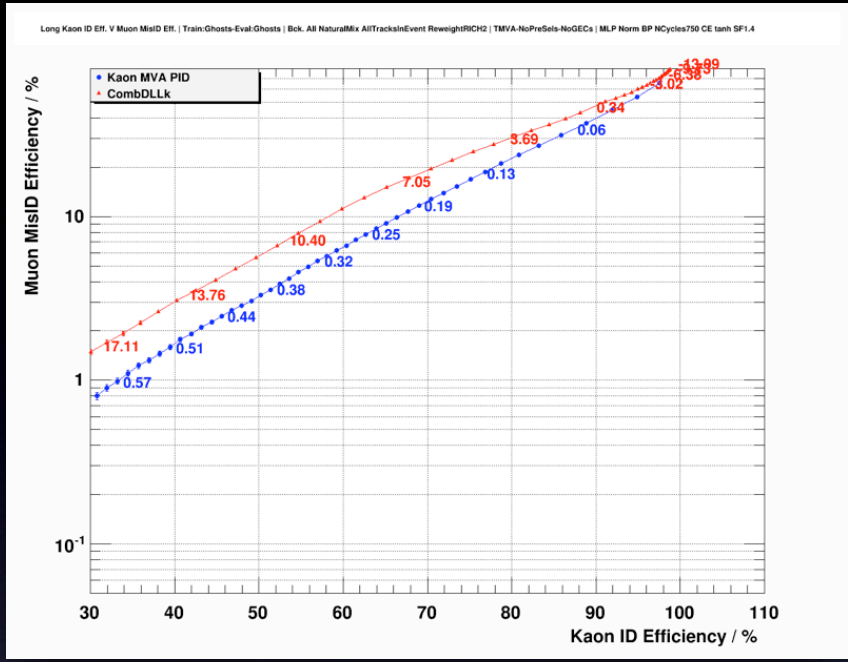


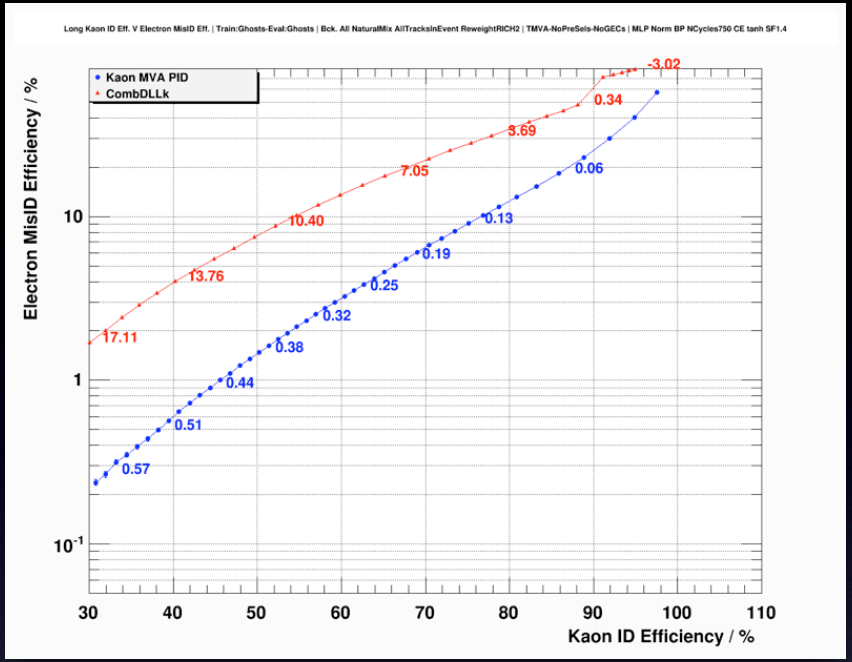


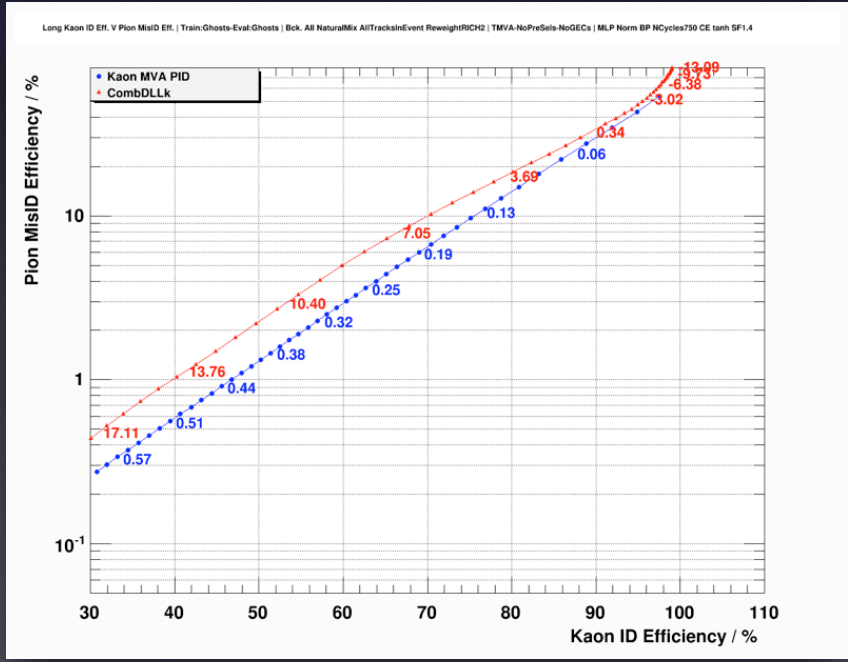
Kaon - Efficiency V Purity

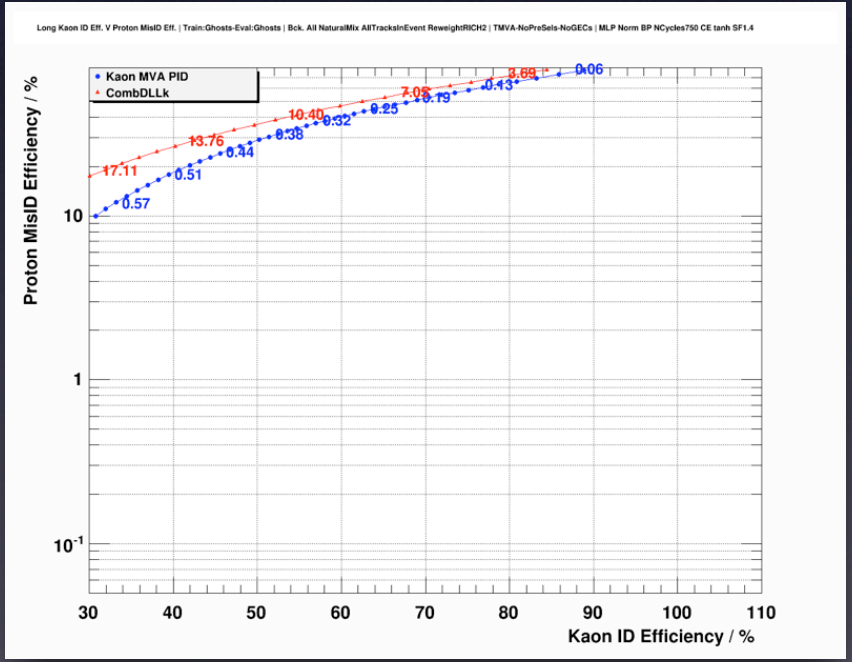


Kaons - Efficiency V MisID

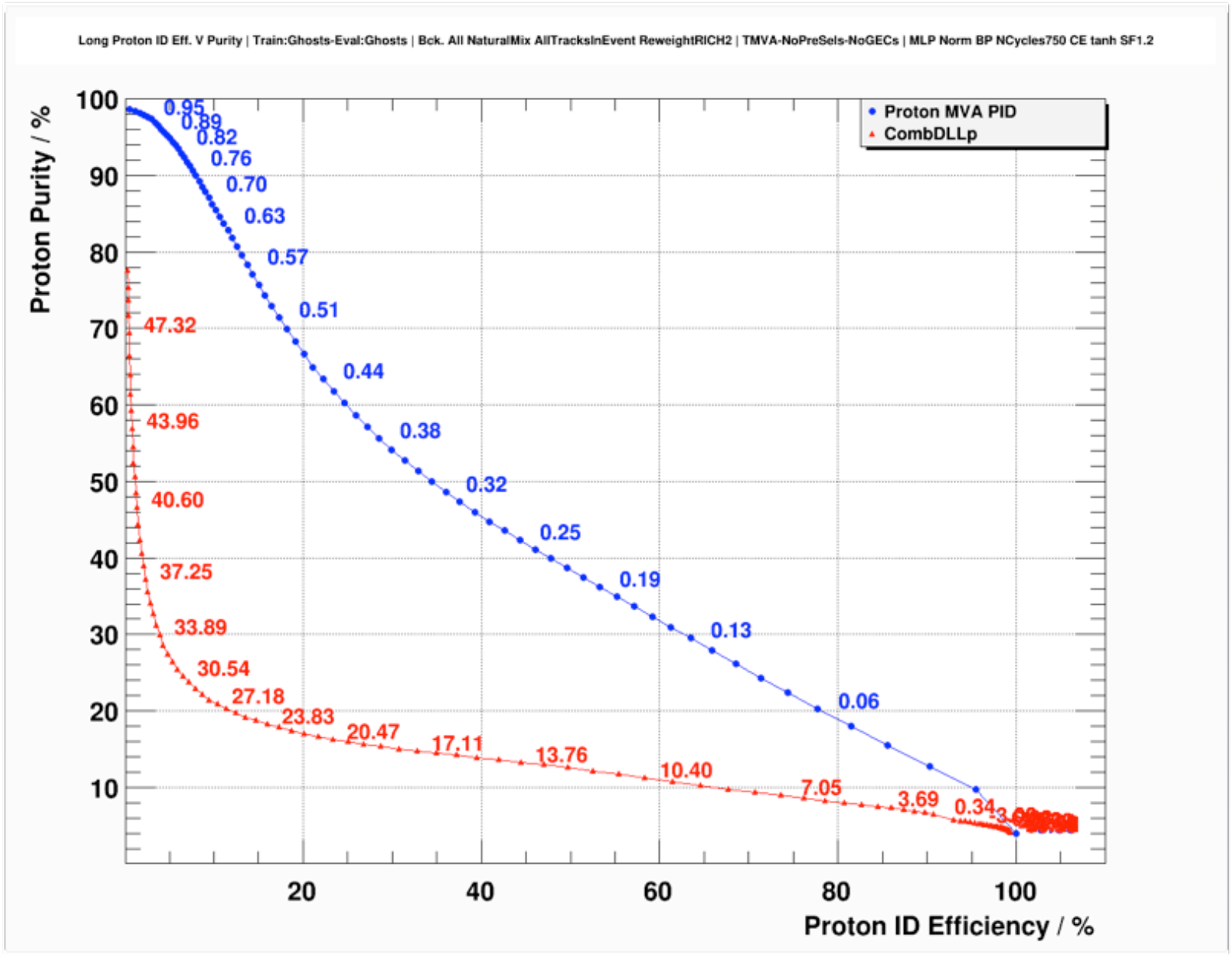




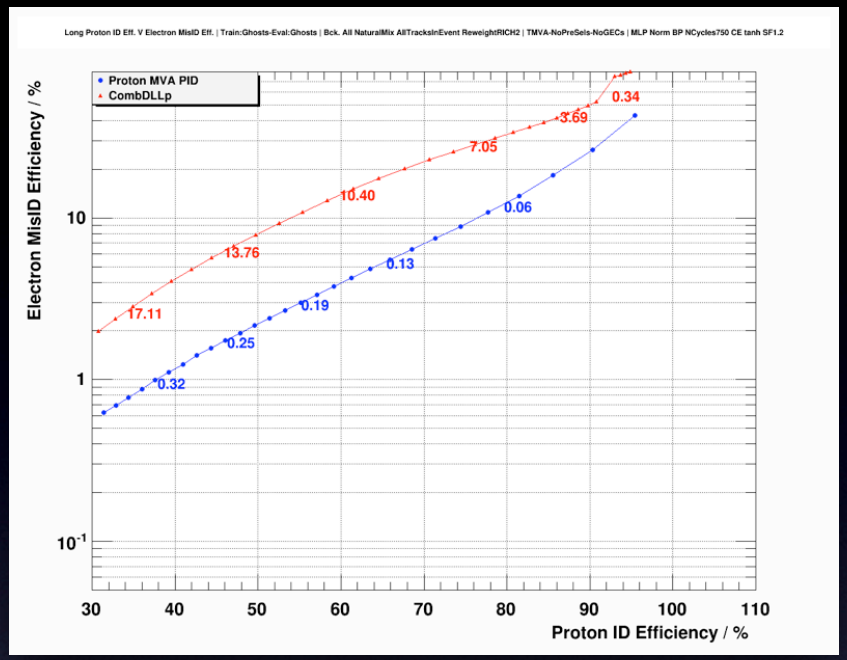


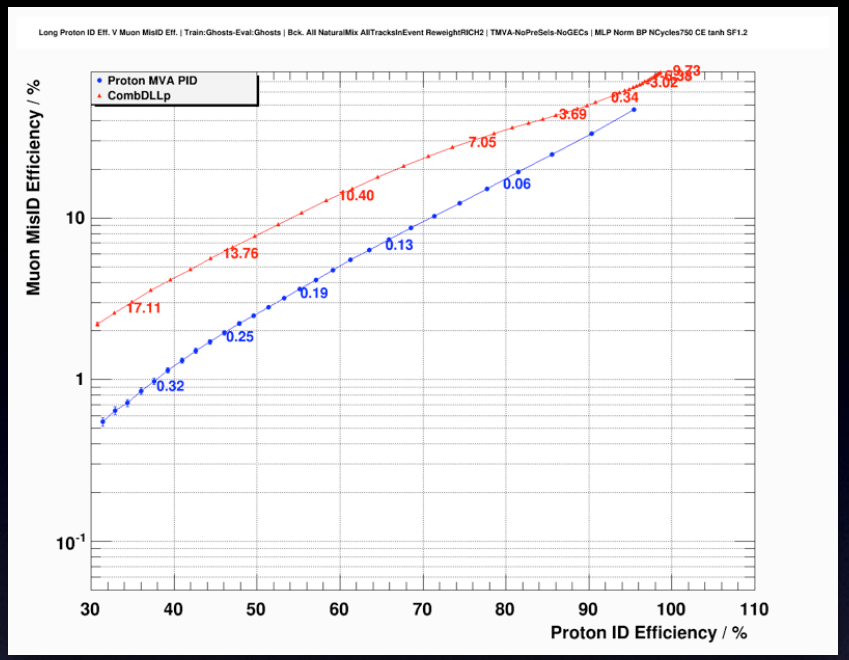


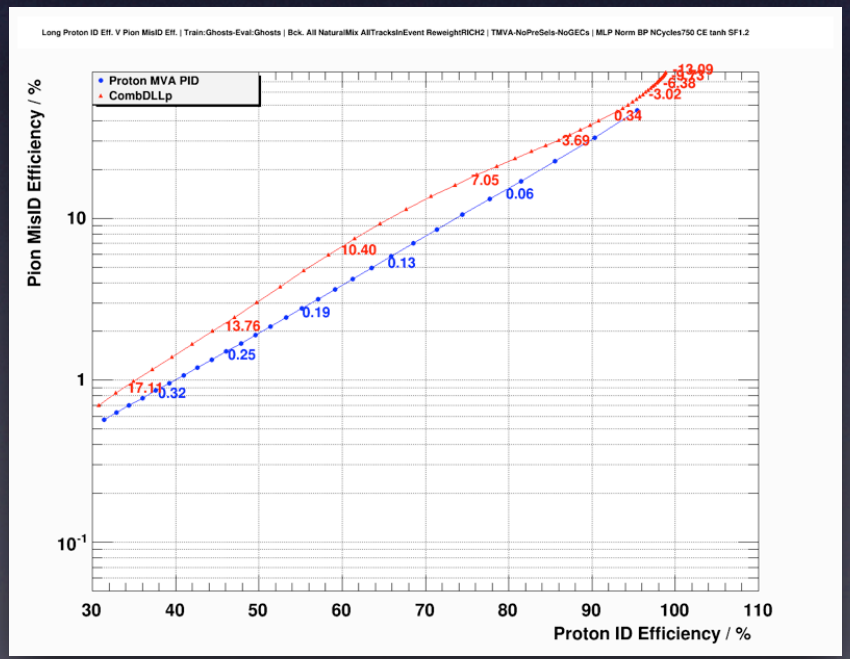
Protons - Efficiency V Purity

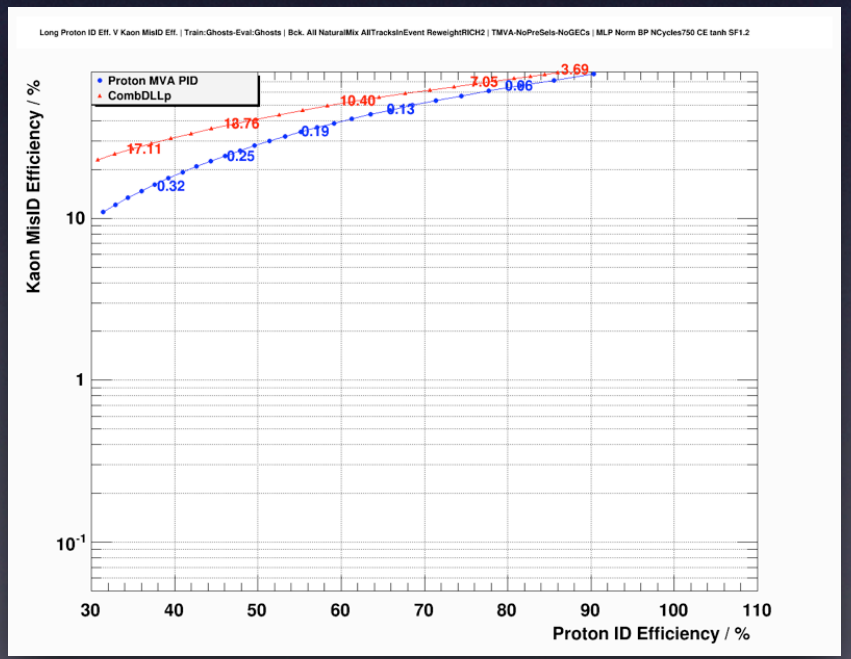


Protons - Efficiency V MisID

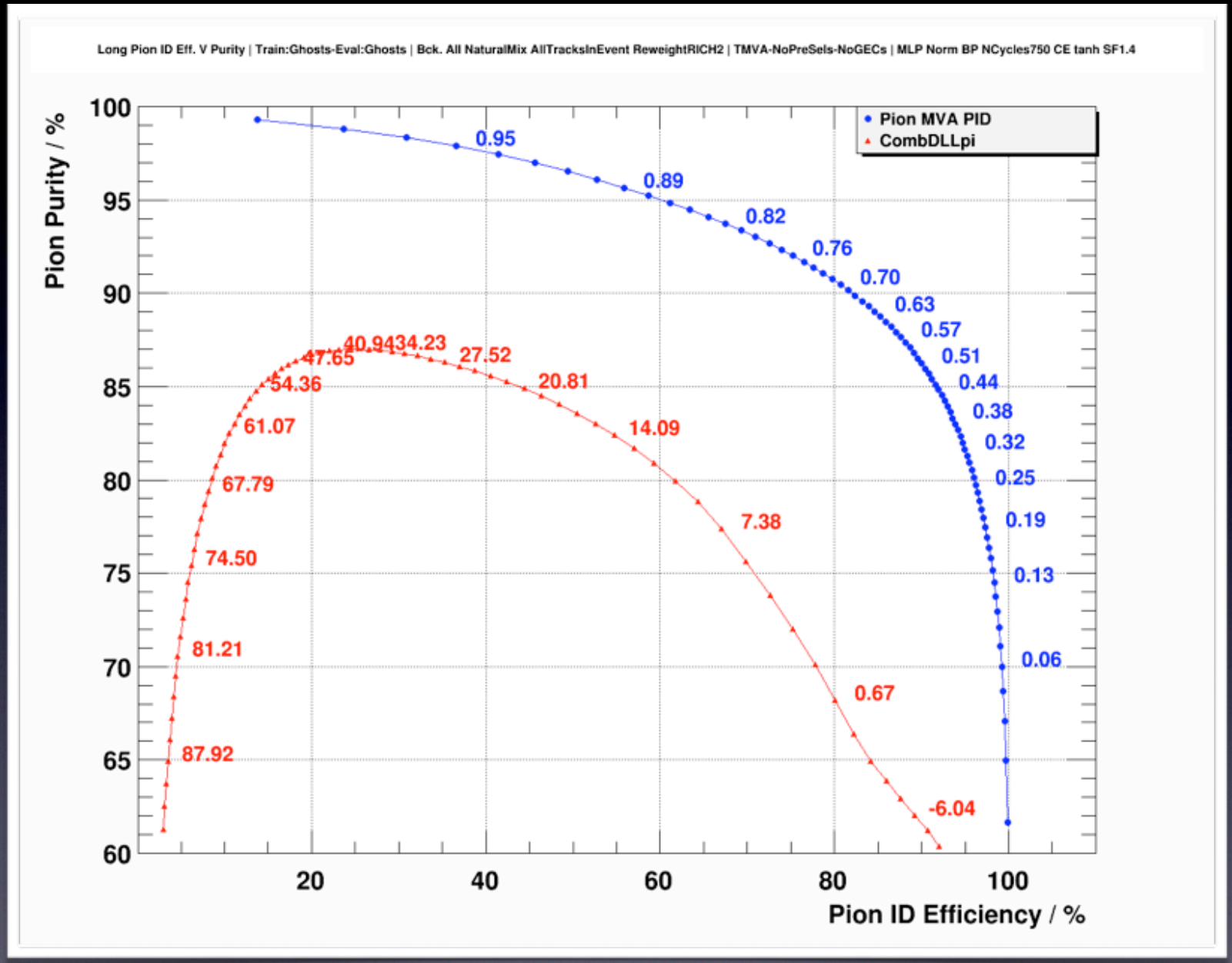






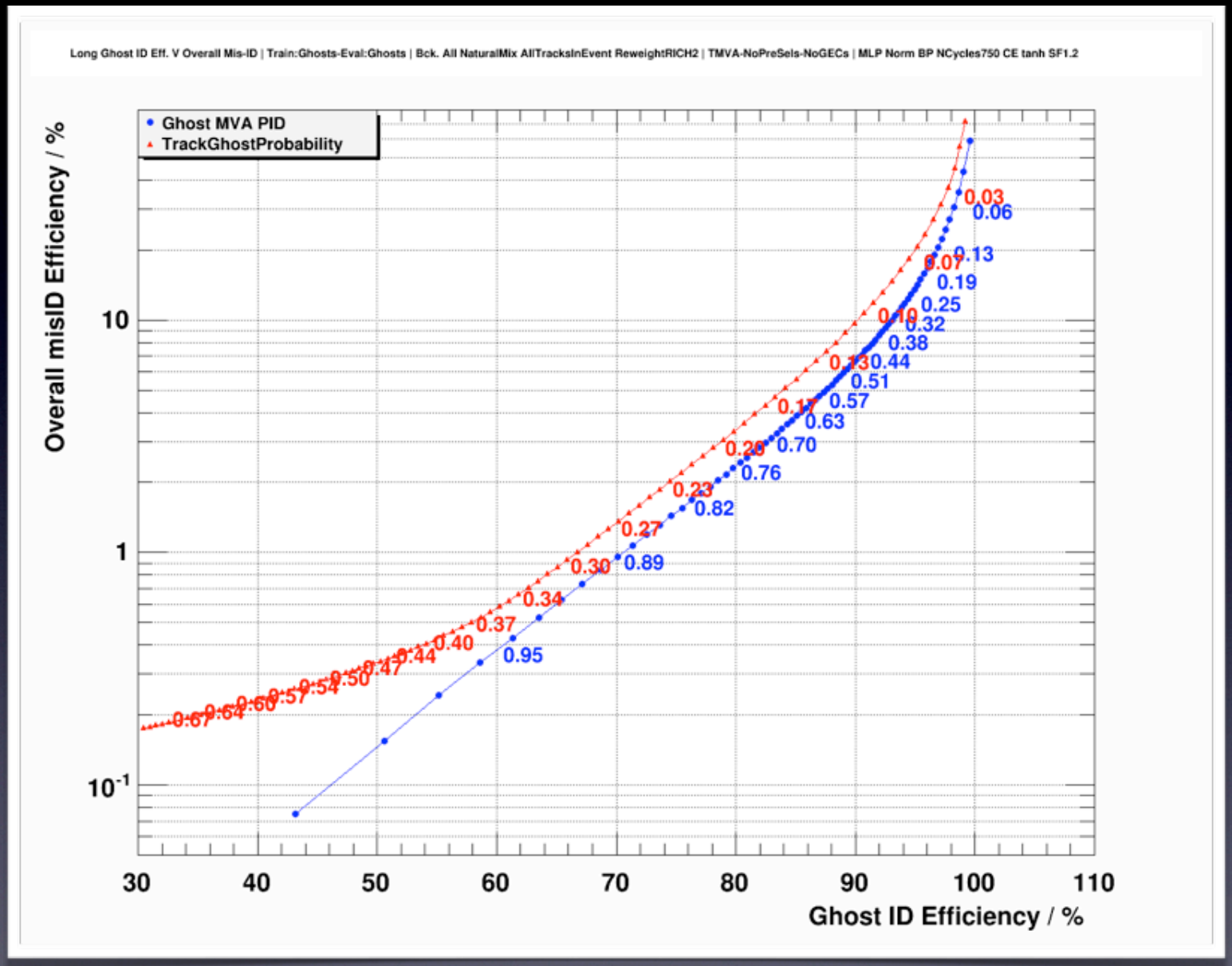


Pion ANNPID also available



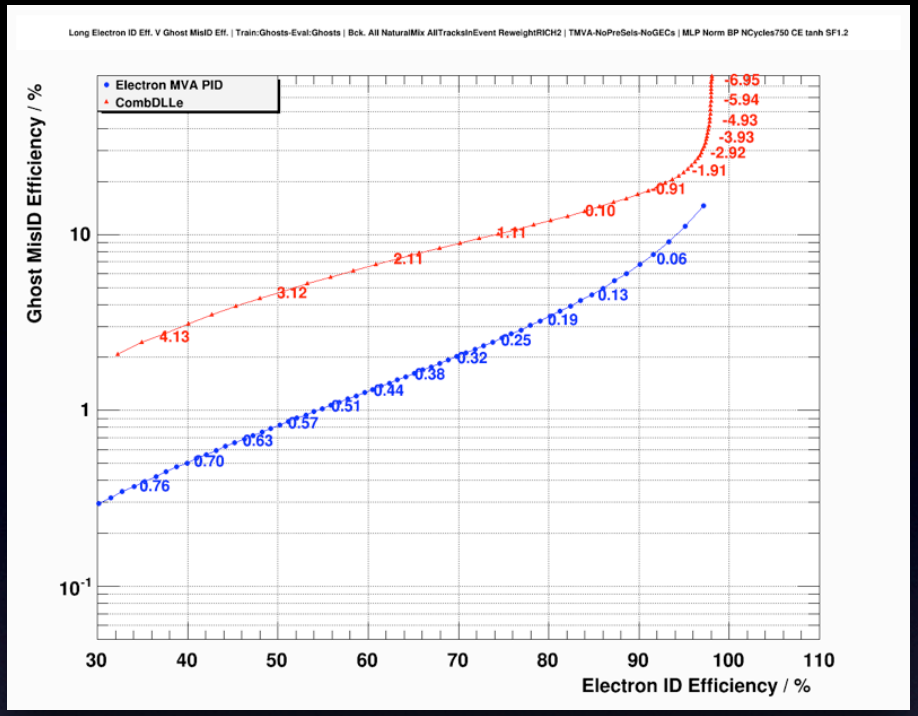
No direct CombDLL equivalent variable.
Used CombDLLpi = (-DLLel-DLLmu-DLLk-DLLp) just as an example

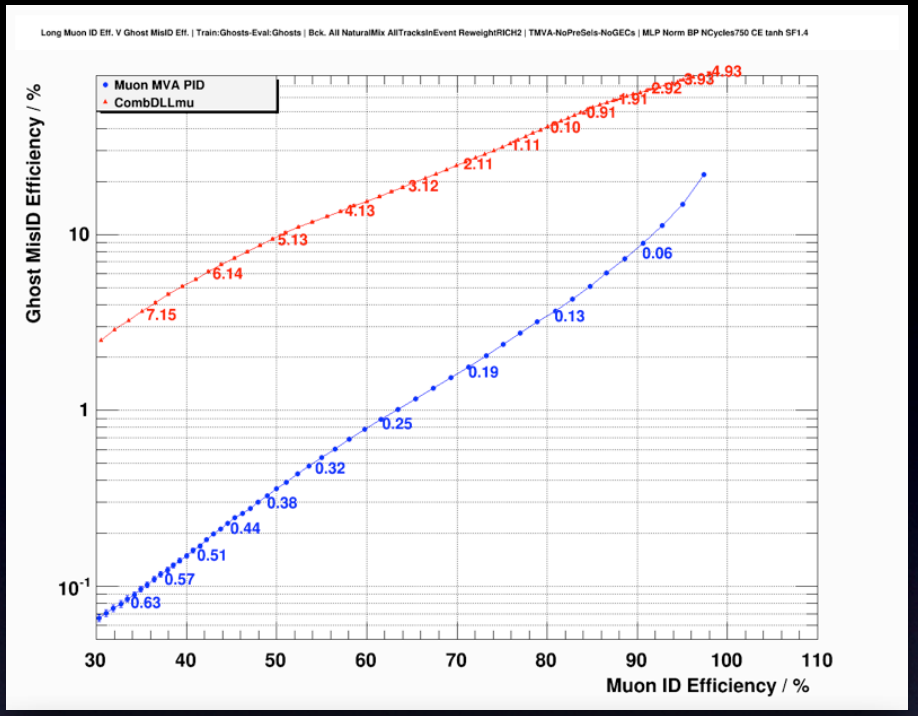
Ghost Selection (aka Rejection)

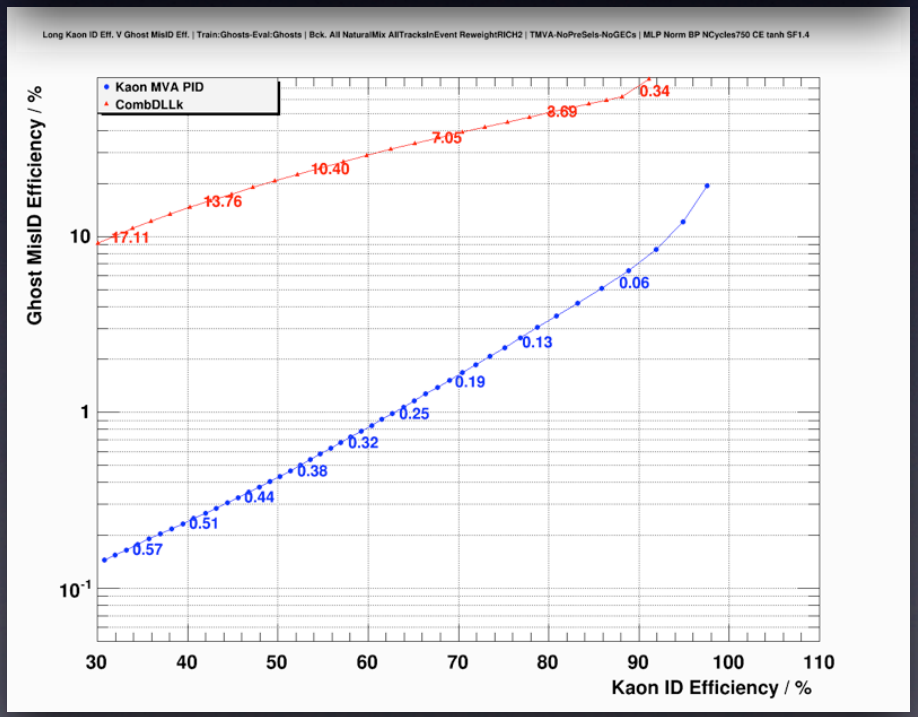


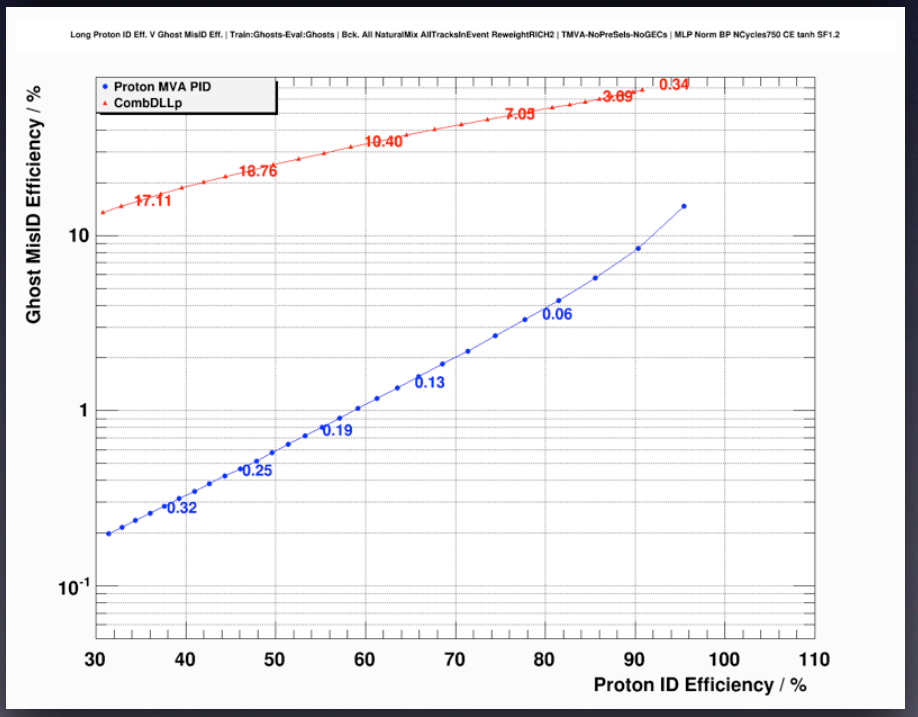
- Ghost Probability network. Combining tracking with PID

Ghost rejection built into ANNPID variables



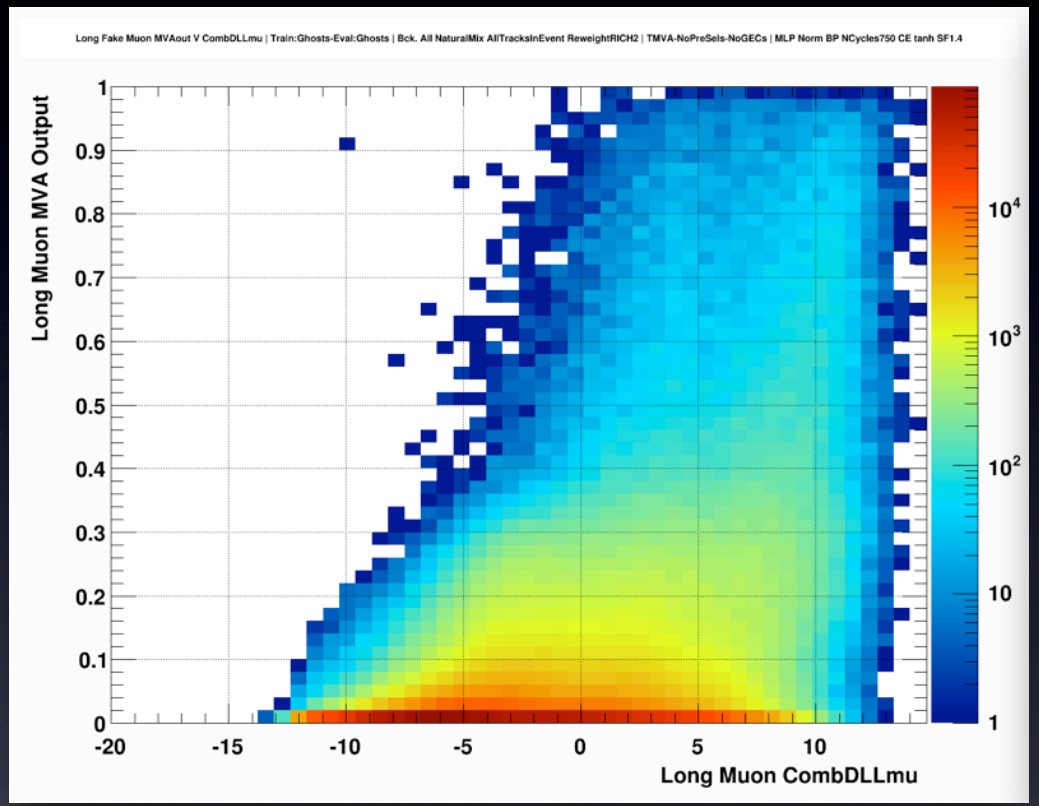




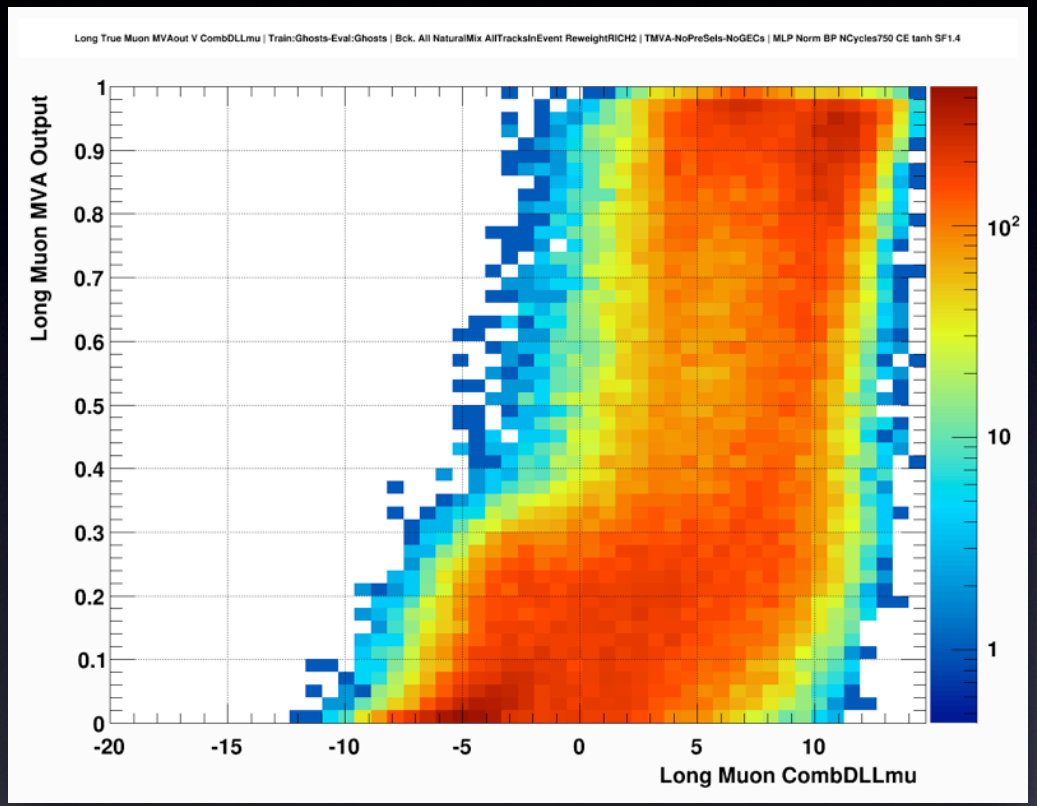


ANNPID V CombDLL Correlations

Muon ANNPID V CombDLLmu

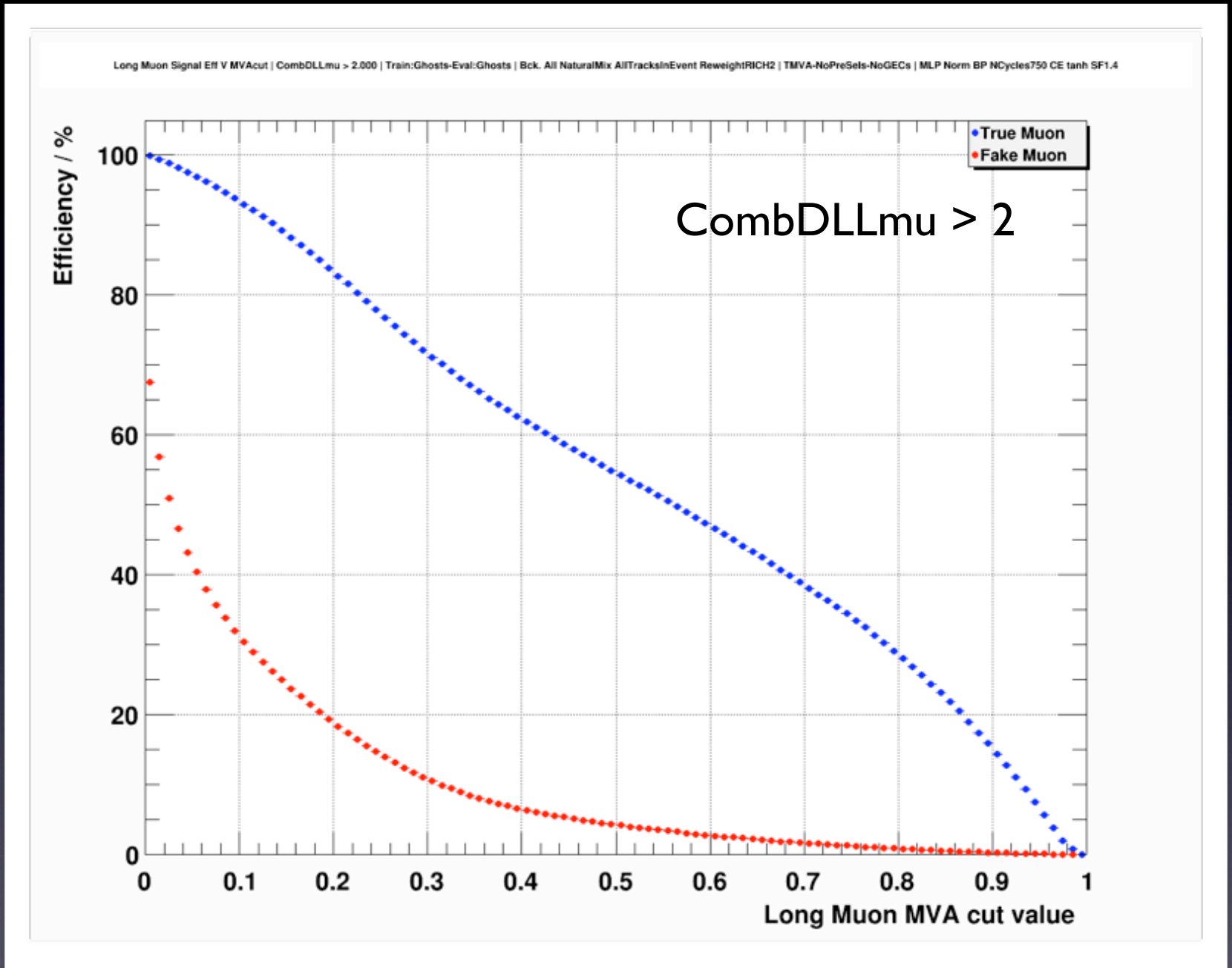


Background

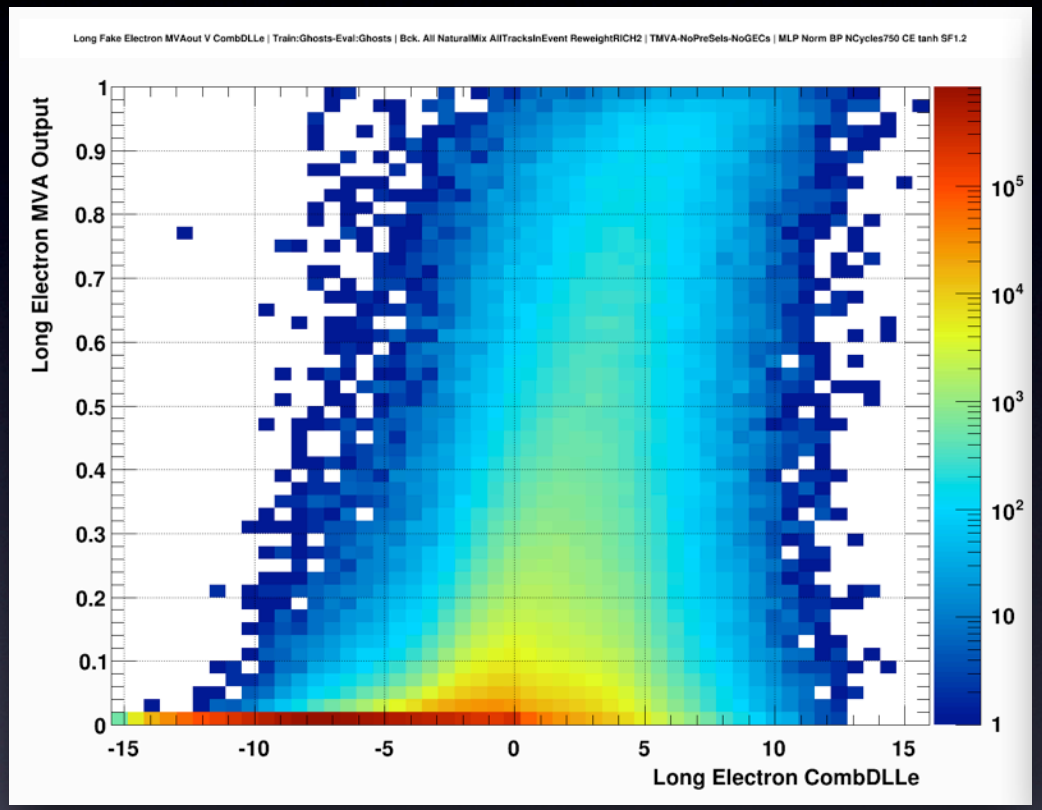


Signal

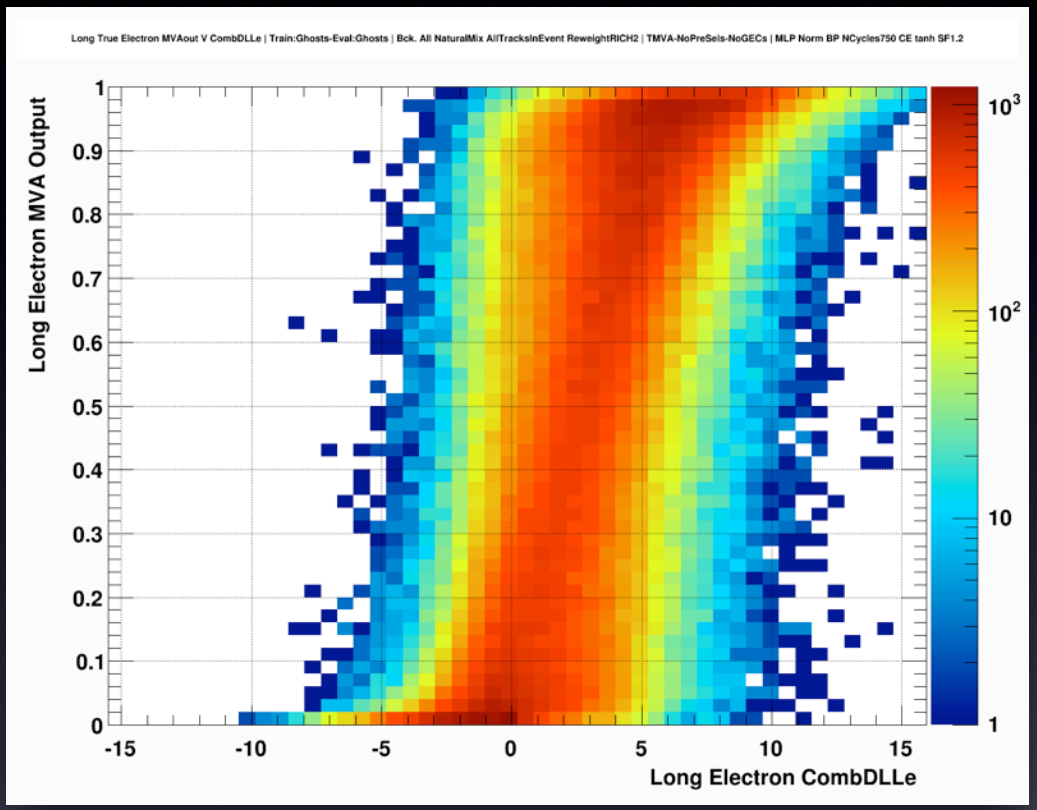
Muon ANNPID Efficiency for given CombDLLmu cut



Electron ANNPID V CombDLLe

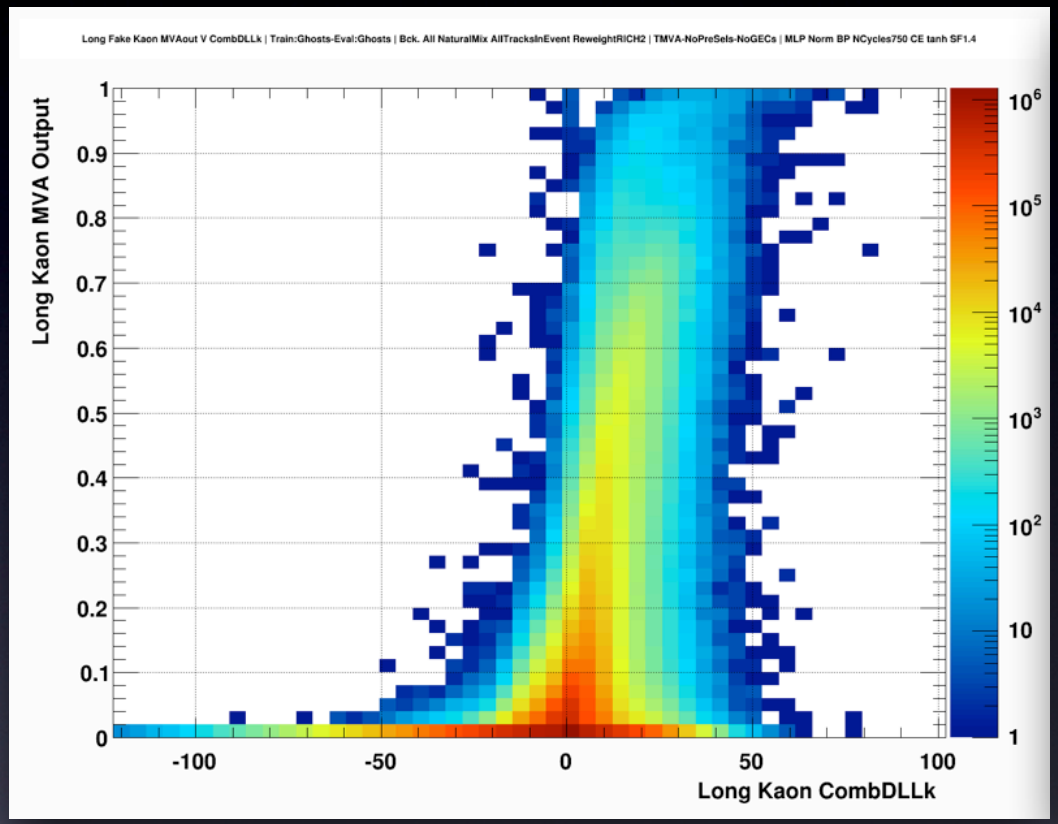


Background

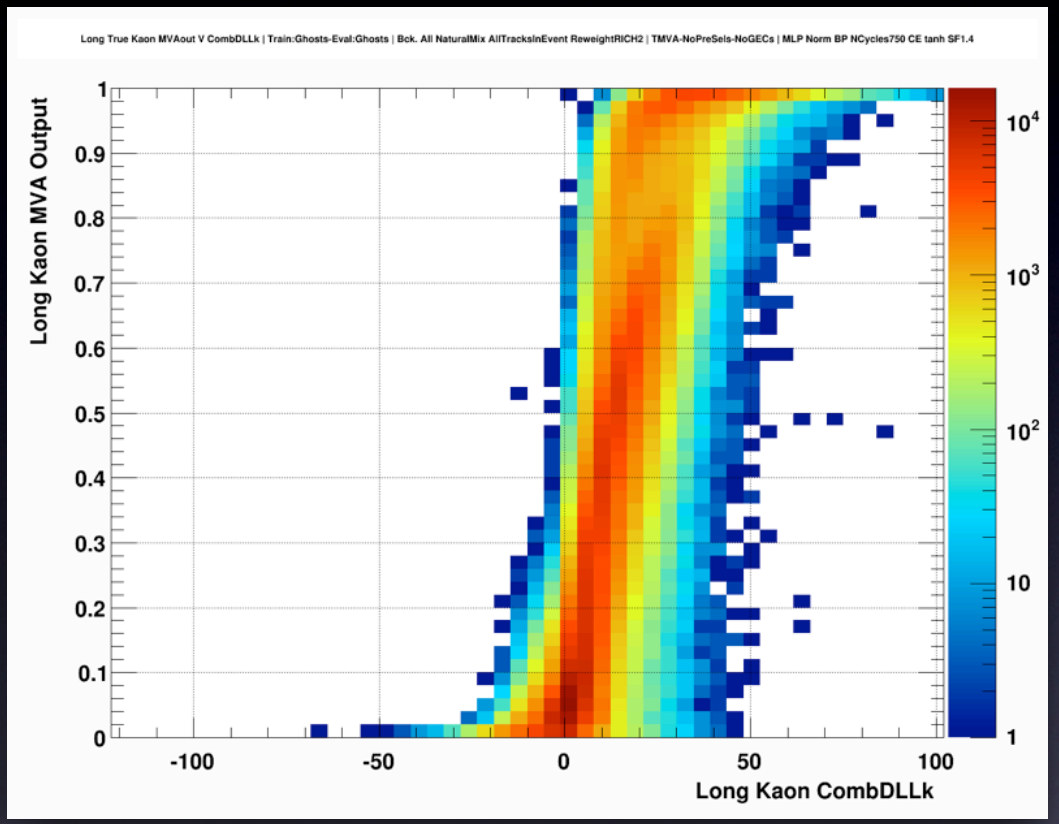


Signal

Kaon ANNPID V CombDLLk



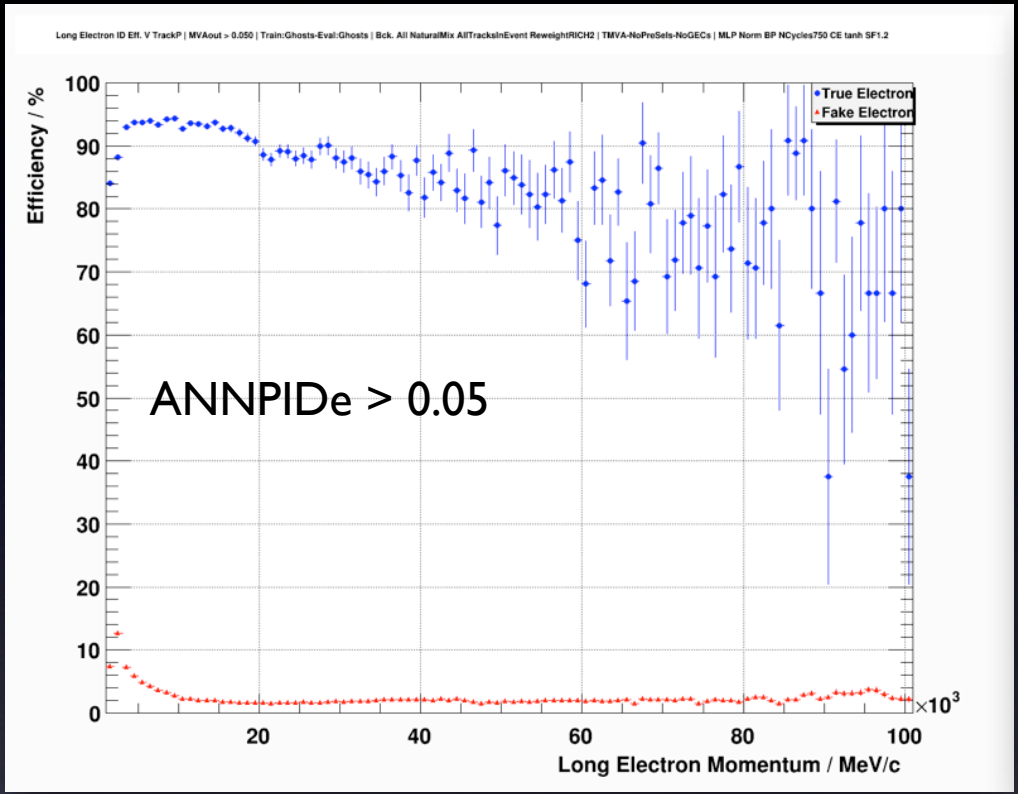
Background

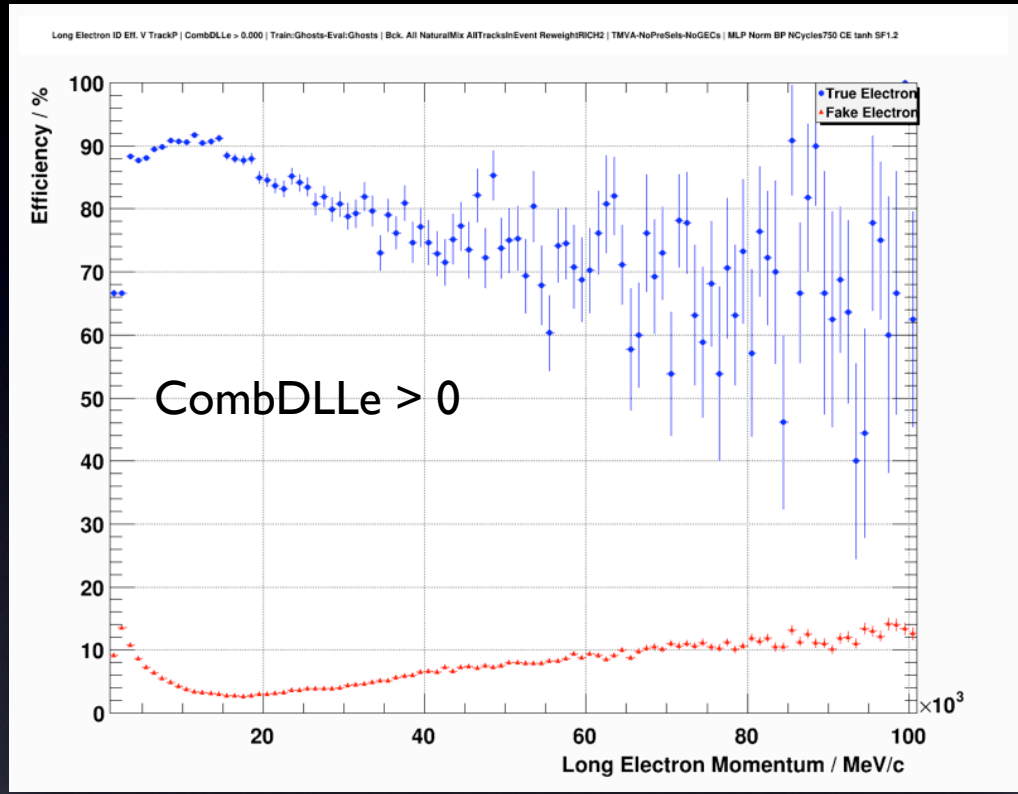


Signal

Efficiency V Momentum

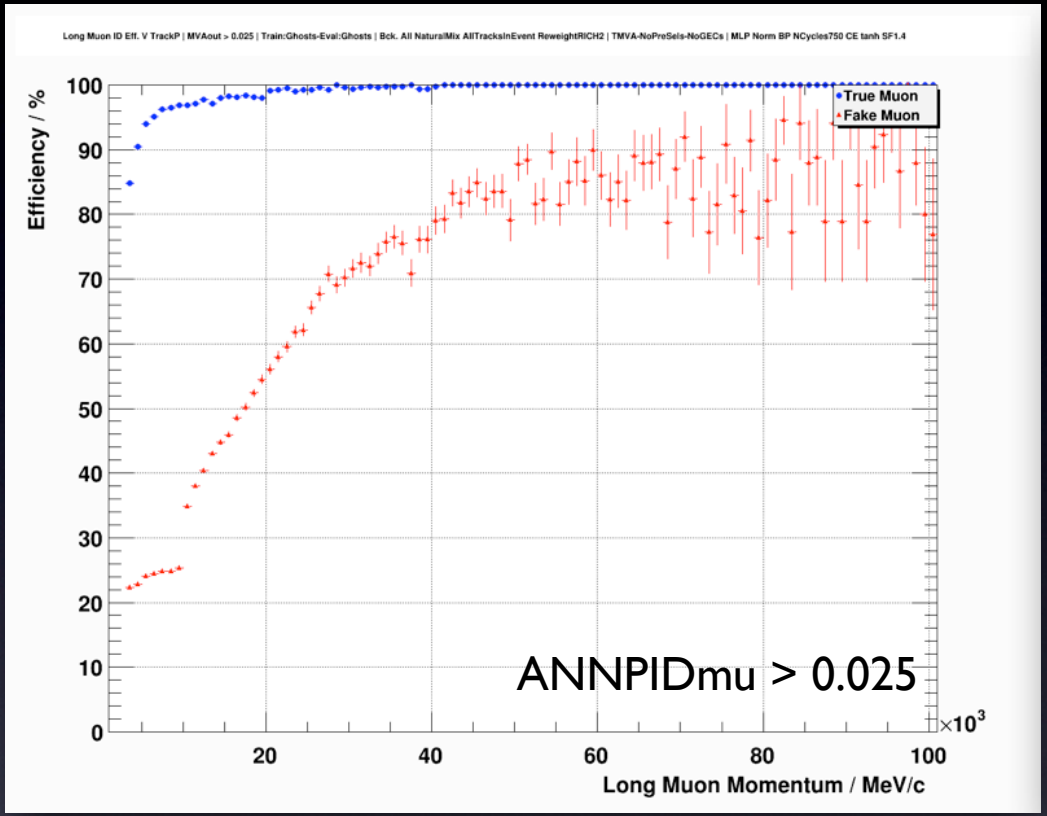
Electron : Eff.V Momentum

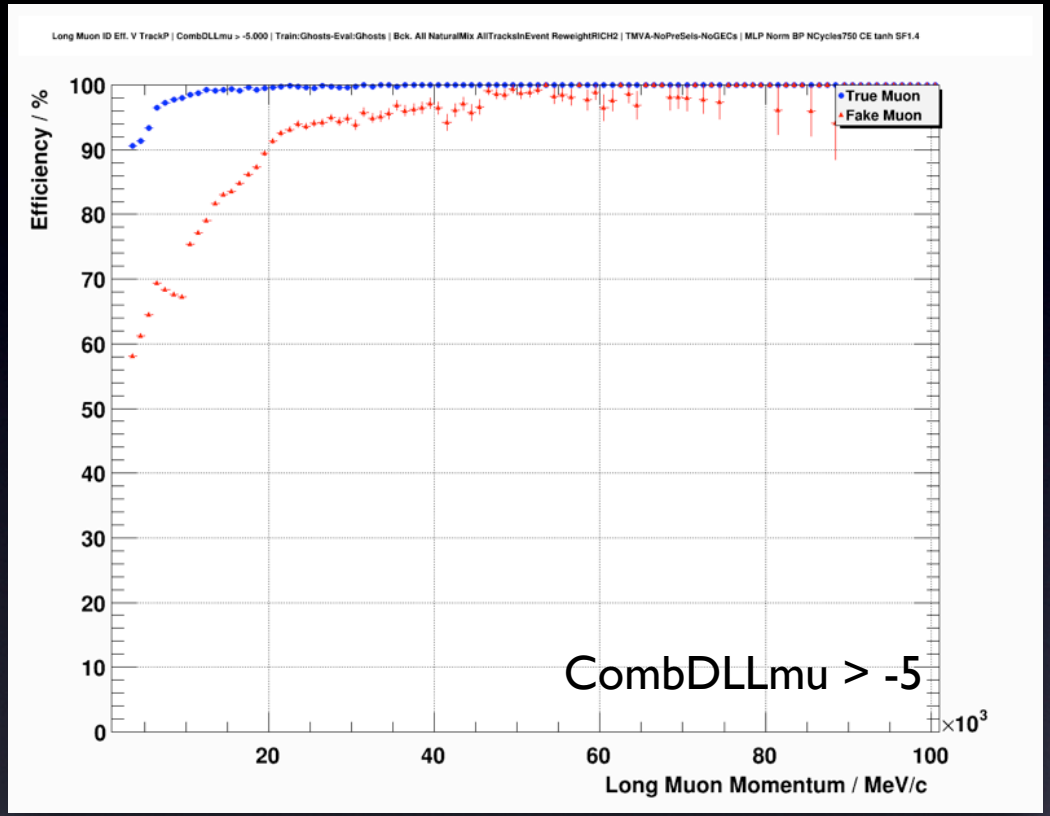




- Similar efficiency form, but flatter misID

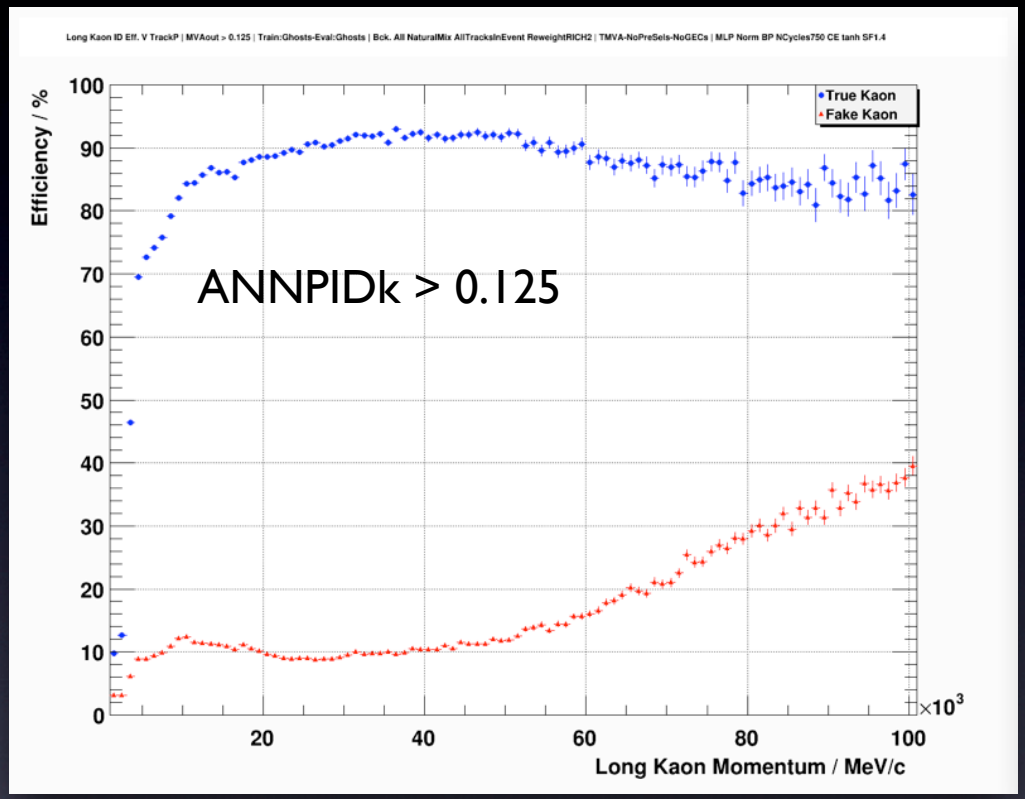
Muon : Eff.V Momentum

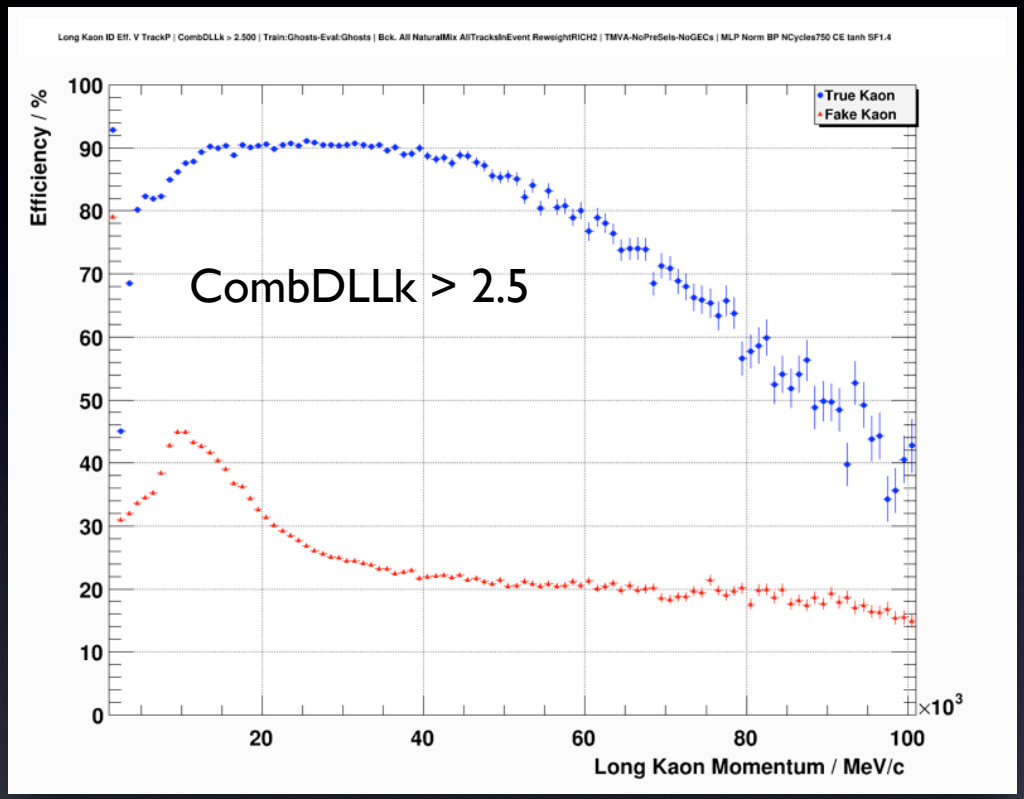




- Better background rejection

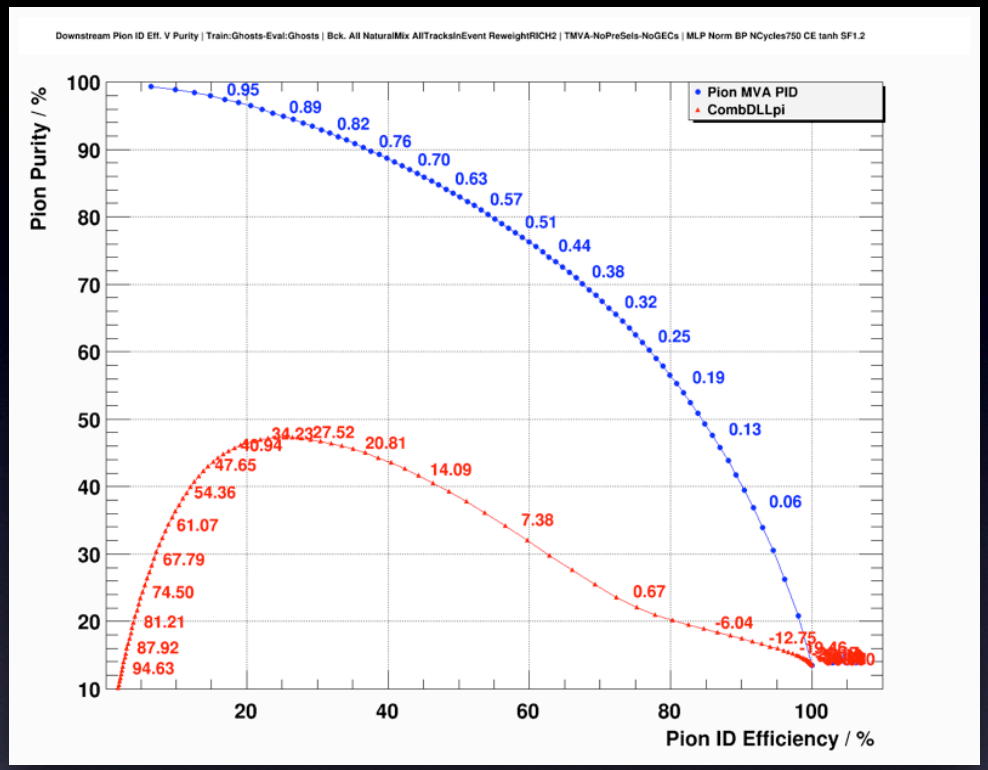
Kaon : Eff.V Momentum



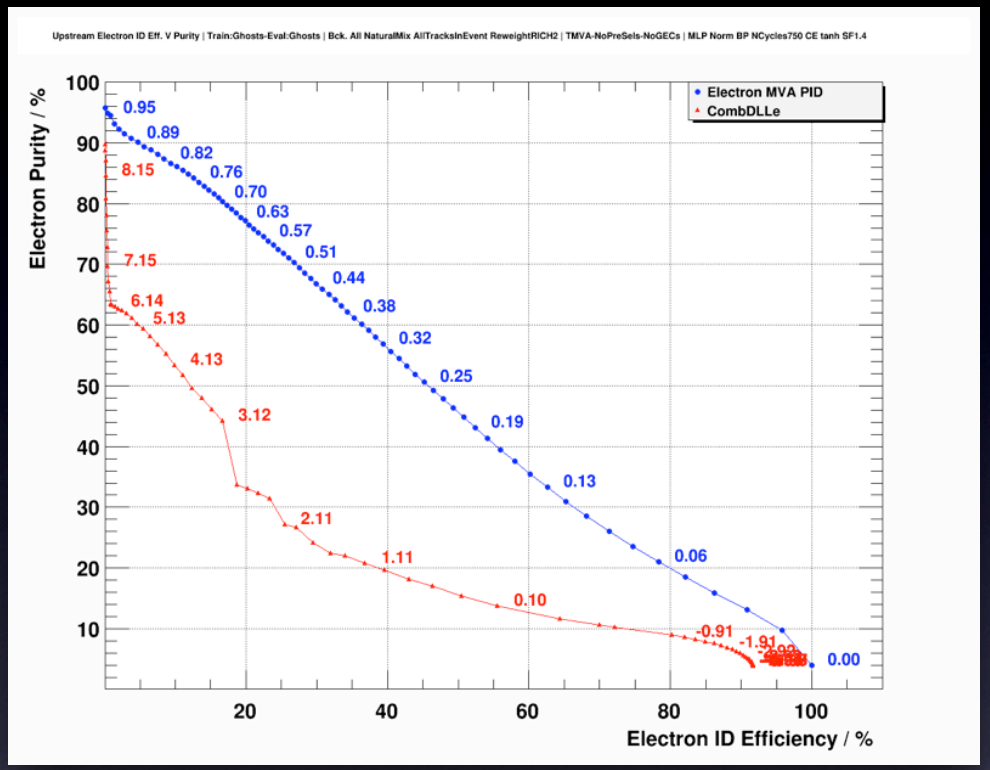


- Different behaviour as a function of P

Size isn't Everything



Downstream Pions



Upstream Electrons

- ANNPID also available for (Down/Up)Stream tracks

Software Details

- ***DaVinci v33r3p3***
  - Recalibration automatically applied when processing Reco14 data
  - Data driven check automatically skips the recalibration for any other processings
- ***Rec/ChargedProtoANNPID v2r1***
  - Algorithms to apply the networks
  - Algorithms/tools to produce the training ntuples
- ***Rec/ChargedProtoANNPIDTeacher v1r0***
  - Package to run the training

Conclusions (Finally...)

- Retuning of the ANNPID variables for Reco14 available.
  - Run by default from DaVinci v33r3p3 (and derived Bender releases)
  - ***Better performance over previous tuning.***
    - Should also apply better to data, due to occupancy sampling
  - Due to switch to TMVA, users will need to retune their cuts
  - Better performance (in MC) over Combined DLLs.

- ***Strongly urge*** users to try them in their data selections and provide feedback.

- Internal note under preparation ...
- Many more performance plots available from http://www.hep.phy.cam.ac.uk/~jonesc/lhcb/PID/ANN/MC12TuneV2/