DLL vs ProbNN PID variables

A.Hicheur, CBPF Rio de Janeiro

28/01/2014

54th LHCb Analysis and Software week

PID Workshop

Outline

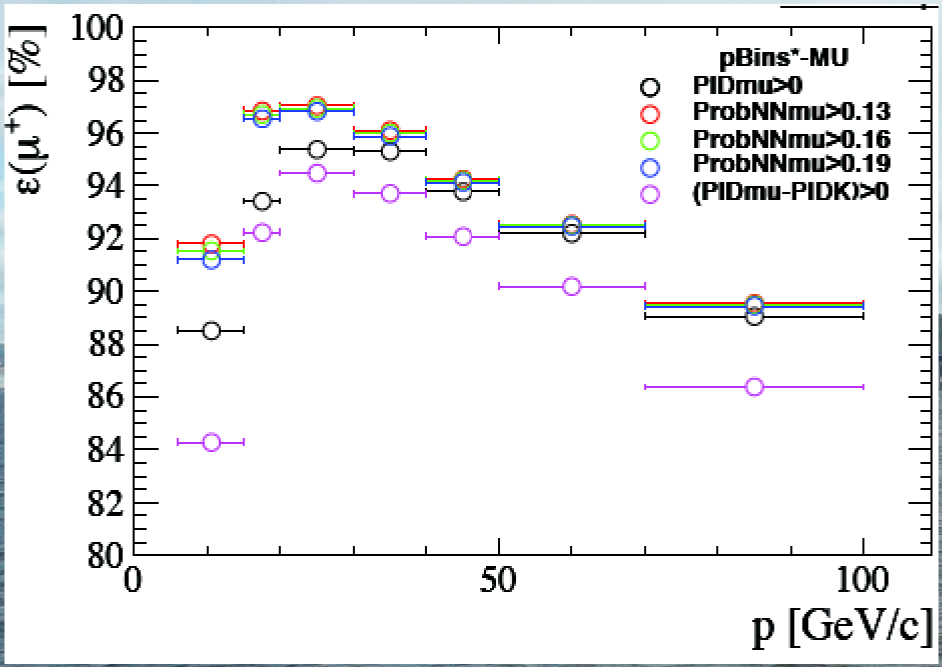
- Presentation of the variables
- Muons
- K and π ID
- Protons
- Conclusion

Thanks to all the people who provided feedback

Presentation of variables

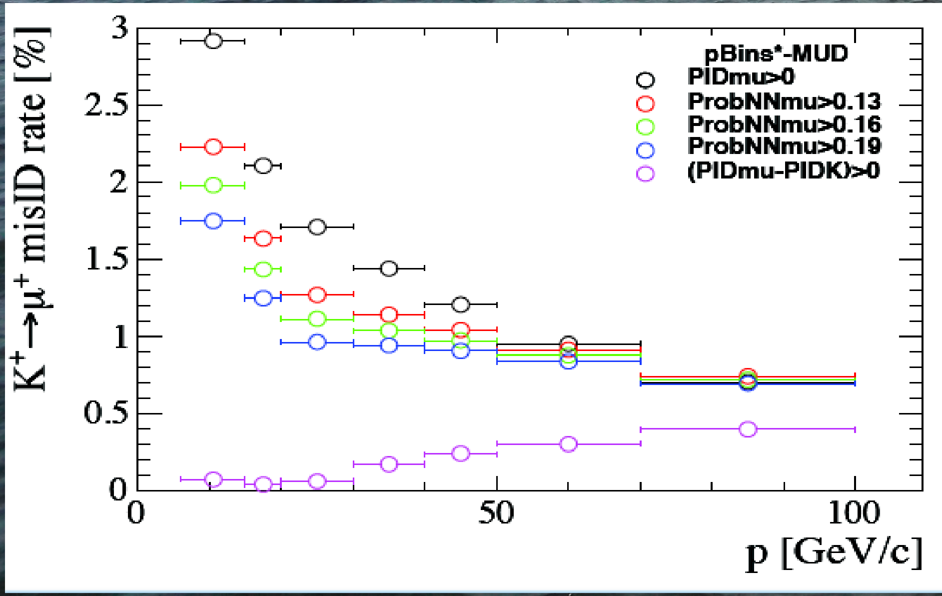
- (Combined) DLL : use information provided by each subsystem
  - Some caveats : definition of DLLs, different range of values for the various DLLs, peculiar structures (spikes), all information not used optimally etc...
- ProbNN : use more information
  - Inputs include basic PID variables from the CALO, MUON and RICH sub-systems + some tracking variables. Combined in TMVA MLP-CE network.
  - Provides consistent set of probability values for each particle hypothesis
- For more details, see e.g :
  - https://indico.cern.ch/getFile.py/access?contribId=1&resId=0&materialId=slides&confId=226062

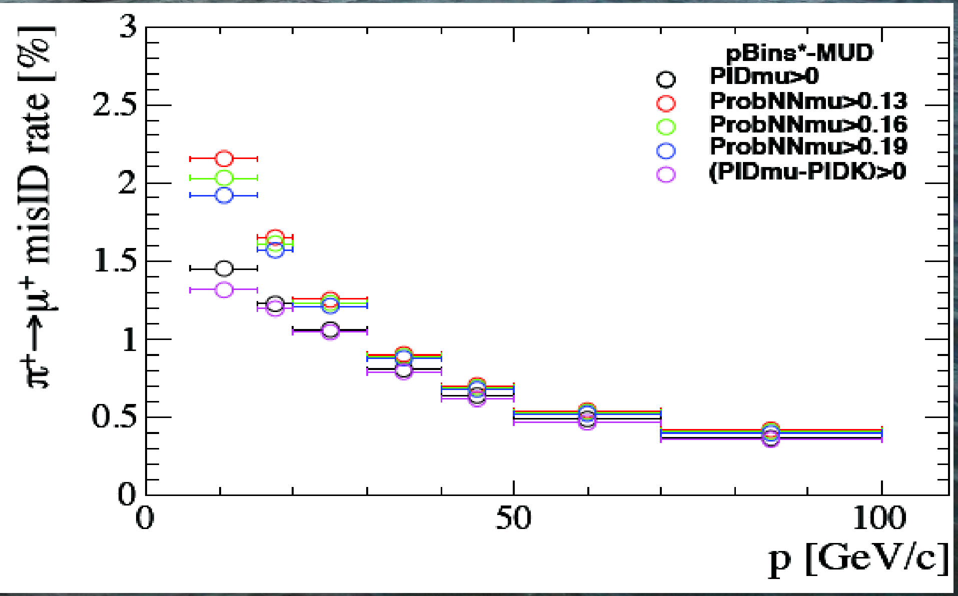
Muon ID – SL WG



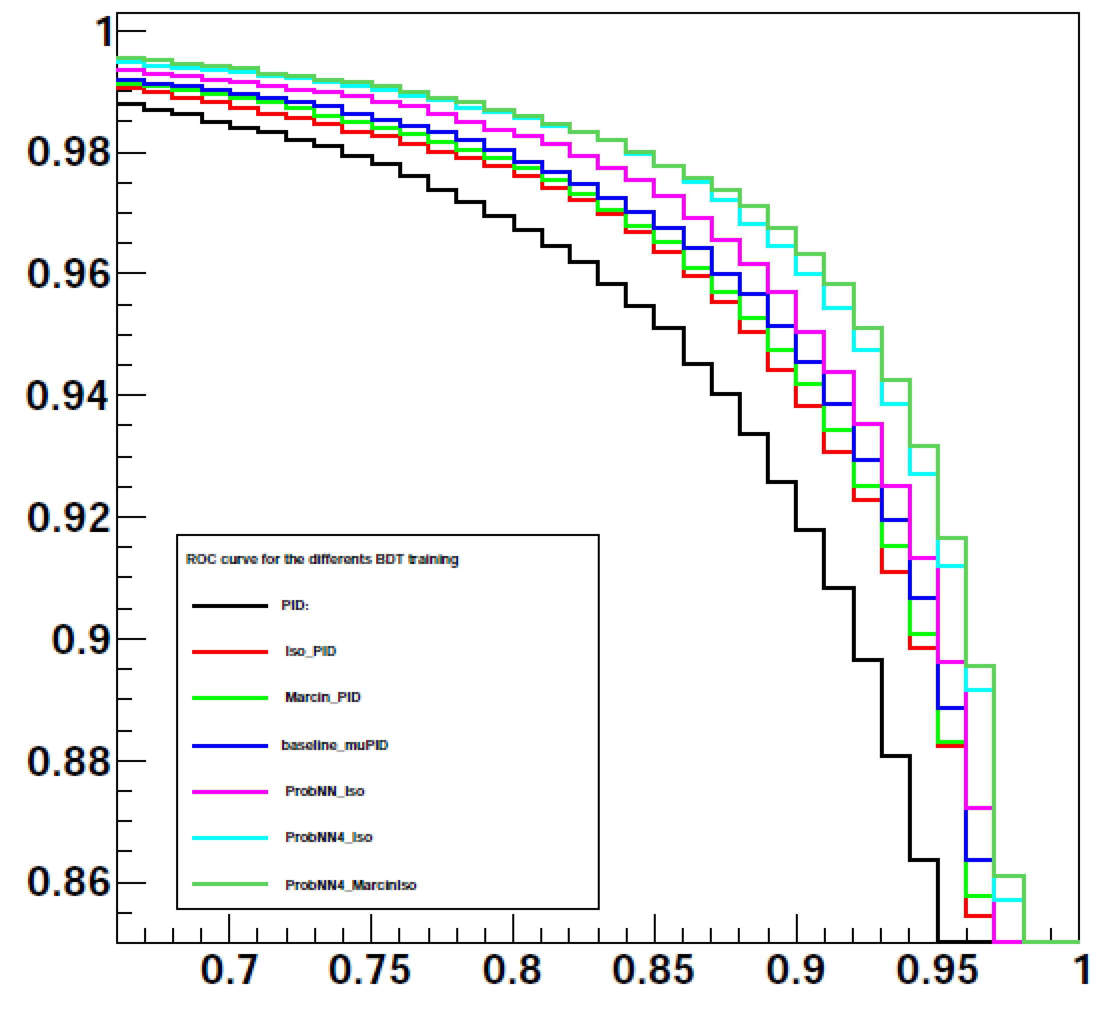
- The probNN variables do significantly better for K-mu separation at a given muon efficiency.
- The improvement is smaller for pi-mu separation.
- They are not yet used in analysis (yet)

Plots from Christos Hadjivasiliou
PPTS meeting, October 7, 2013





Muon ID – EW penguin (RD WG)



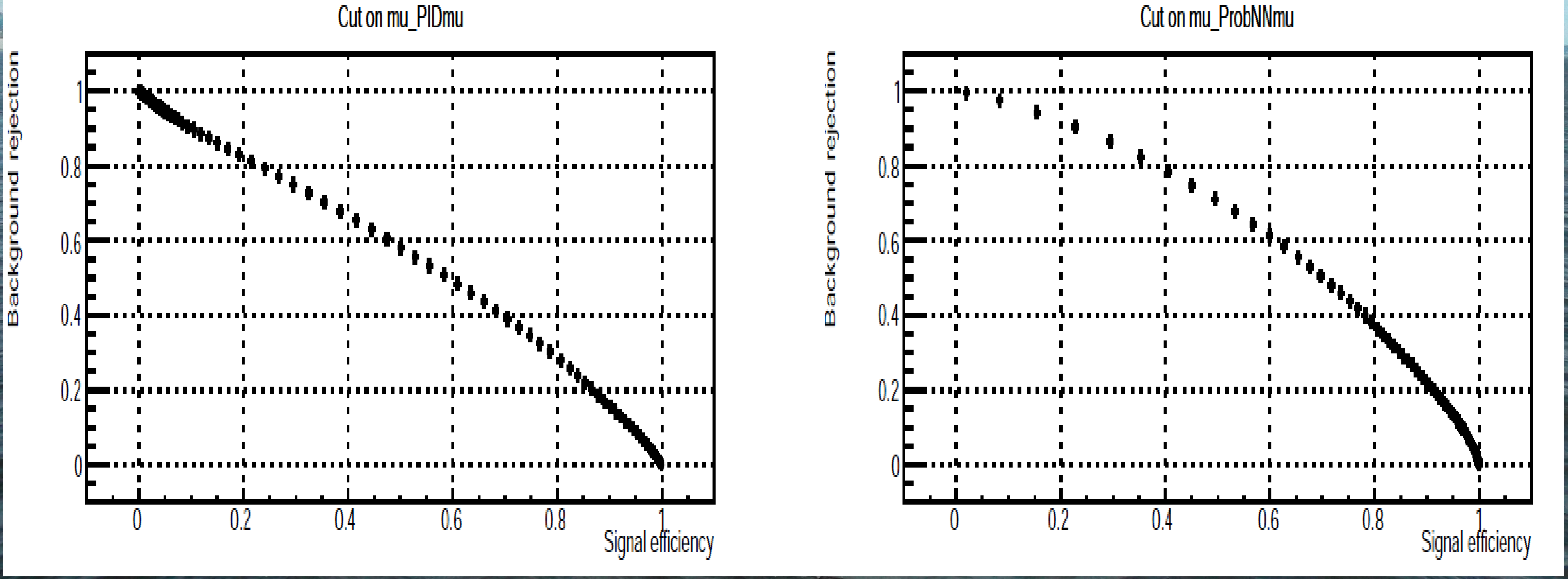
Gain using ProbNN instead of DLL variables in BDT
Need to account for Data-MC discrepancies, however

S.Coquereau et al., 27/11/2013, EWP
Bd → K* μ μ

Muon ID – Charm group

Test performed on the muon ID in B→D0(→KS (LL) ππ) μ ν

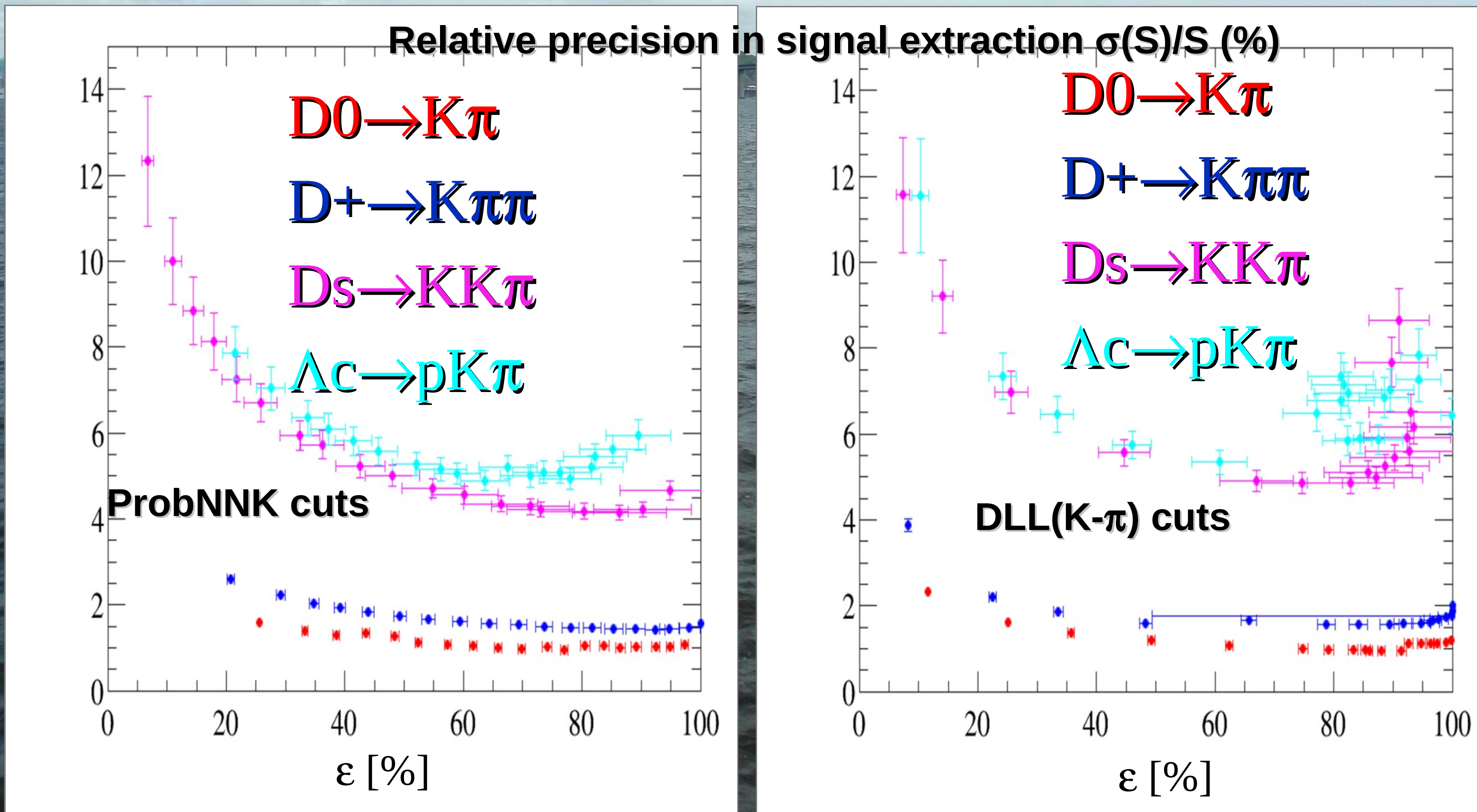
ROC curves (Stefanie Reichert)



ProbNNmu performs better

B hadrons and Quarkonia

- Ψ' polarisation (LHCb-PAPER-2013-043) and Υ @2.76 (LHCb-ANA-2013-048) studies
  - Use of ProbNNmu reduced the background by a factor of 2
- Prompt charm production (V.Belyaev, B&Q, 17/07/2013) :



B decays to open charm (1)

$$B^- \to D[K^+K^-\pi^+\pi^-]K^-$$

M. Coombes, J. Dalseno, J. Rademacker, N. Skidmore

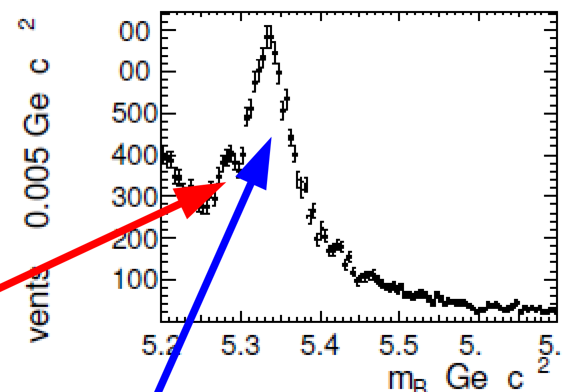
$B^- \to D[K^+K^-\pi^+\pi^-]K^-$ sensitive to γ

Most dangerous background with larger branching fraction, $B^- \to D[K^+K^-\pi^+\pi^-]\pi^-$

Bachelor π^- reconstructed with the K^- mass hypothesis

Peaking structure in the reconstructed B candidate mass shifted by $\approx m(K^-) - m(\pi^-)$

Apply all selection criteria except for PID



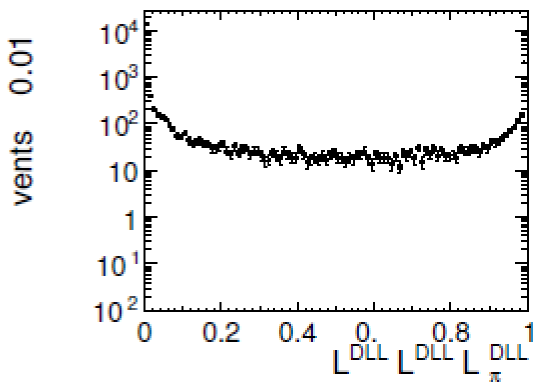
$B^- \to DK^-$ dominated by $B^- \to D\pi^-$ events

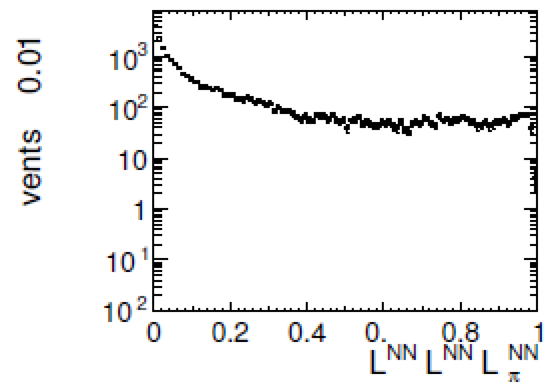
Need clean $B^- \to DK^-$ sample for Dalitz analysis of $D \to K^+K^-\pi^+\pi^-$

1) Compare built Likelihood ratio

$$\mathcal{L}_{K/\pi}^{\mathrm{DLL}} = \frac{\exp(\log(\mathcal{L}_K^{\mathrm{DLL}}/\mathcal{L}_\pi^{\mathrm{DLL}}))}{\exp(\log(\mathcal{L}_K^{\mathrm{DLL}}/\mathcal{L}_\pi^{\mathrm{DLL}}))+1}$$

$$\mathcal{L}_{K/\pi}^{\mathrm{NN}} \equiv \frac{\mathcal{L}_K^{\mathrm{NN}}}{\mathcal{L}_K^{\mathrm{NN}}+\mathcal{L}_\pi^{\mathrm{NN}}}$$

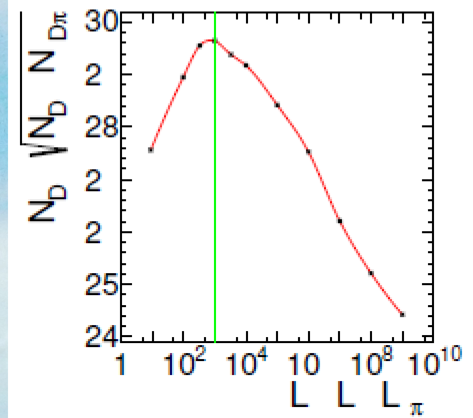




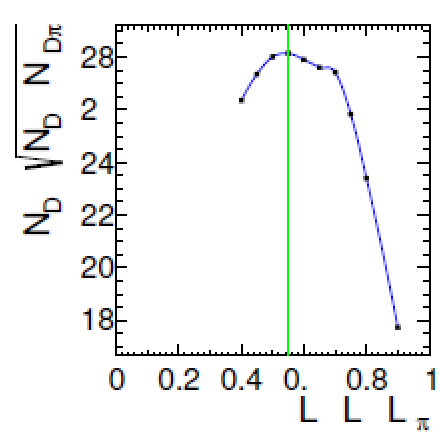
DLL peaks more strongly at 1 for tracks consistent with the kaon hypothesis

2) Optimize cuts vs Dπ bkg

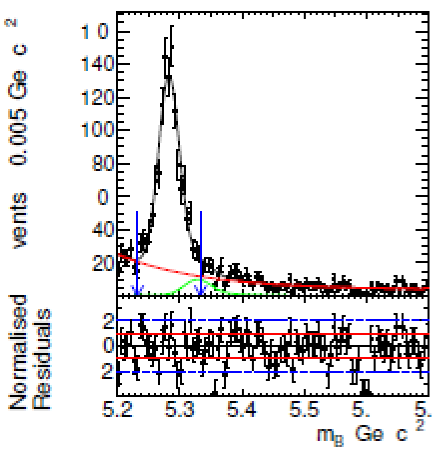
DLL



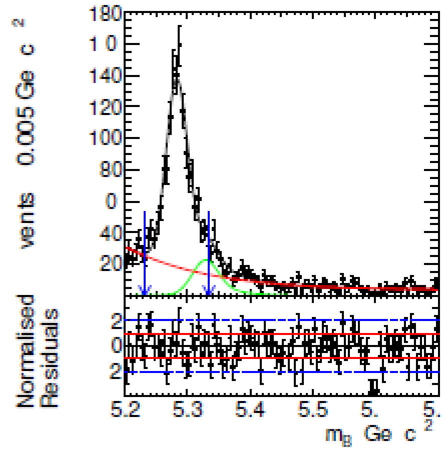
NN



DLL



NN



$N_{DK} = 985 \pm 55$

$N_{DK} = 1003 \pm 49$

$N_{D\pi} = 119 \pm 30$

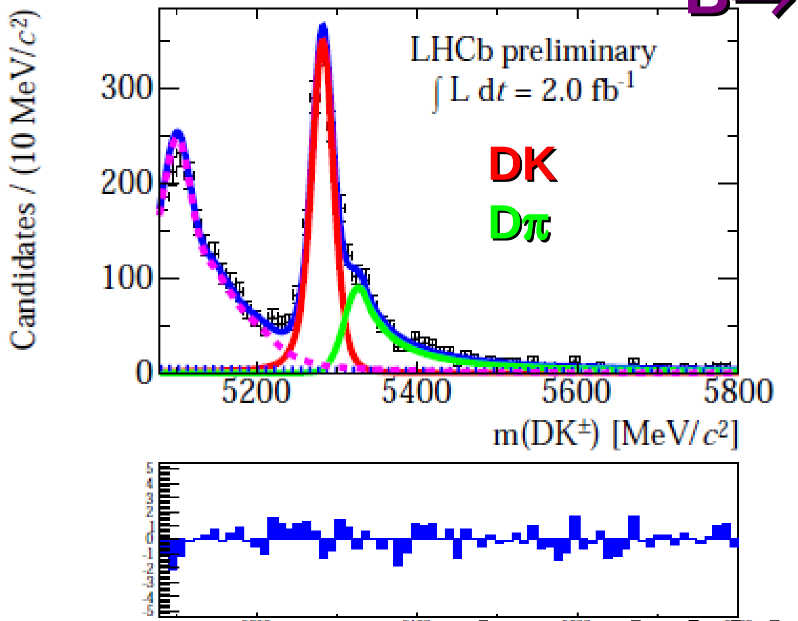
$N_{D\pi} = 268 \pm 36$

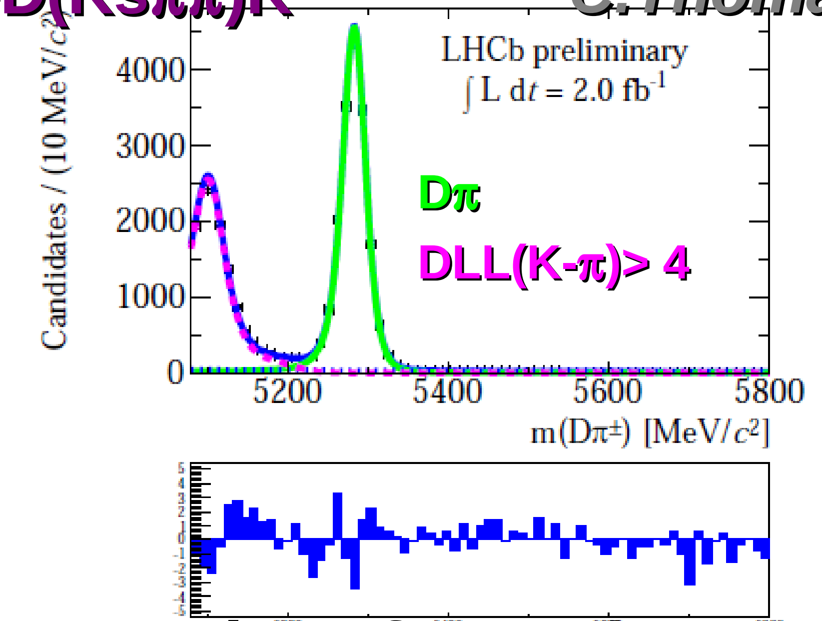
3) Results :
Similar signal yield but
background x 2

B decays to open charm (2)

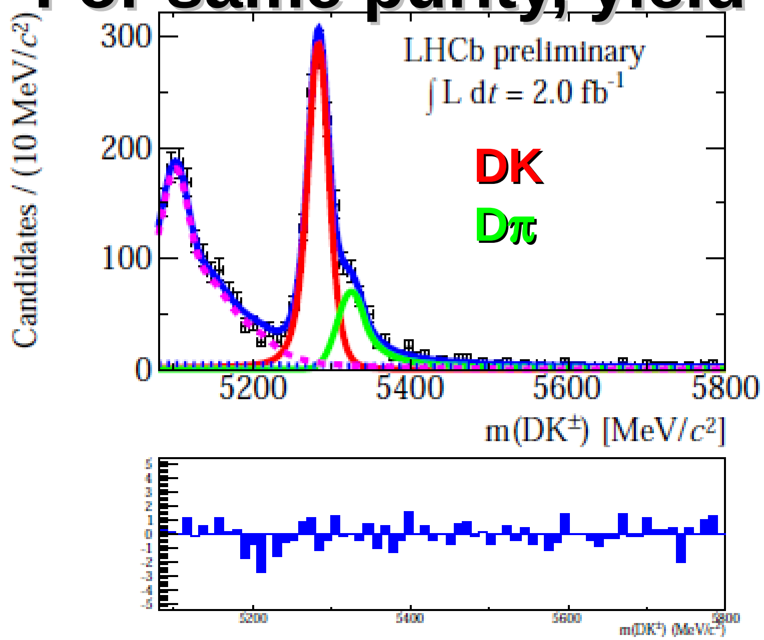
B→D(Ksππ)K

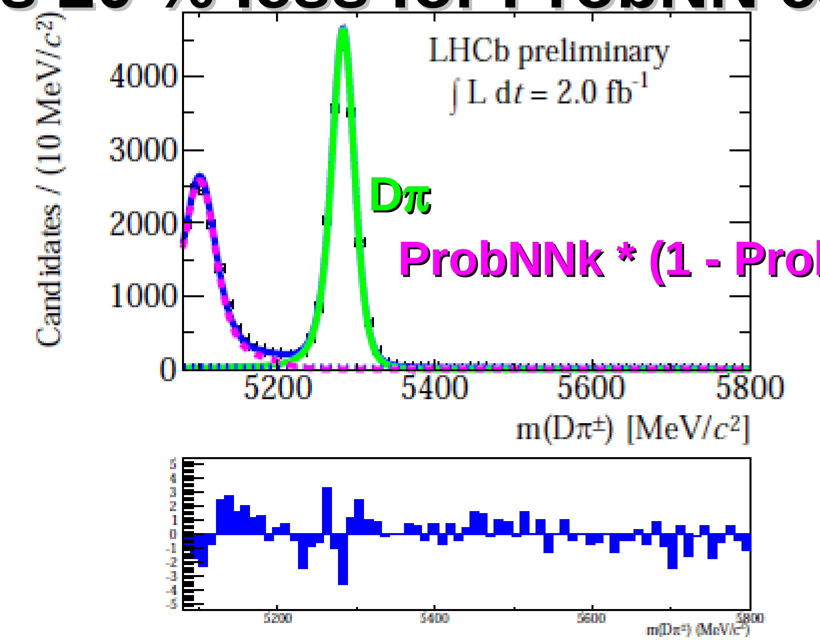
C.Thomas (Oxford)





For same purity, yield is 20 % less for ProbNN case

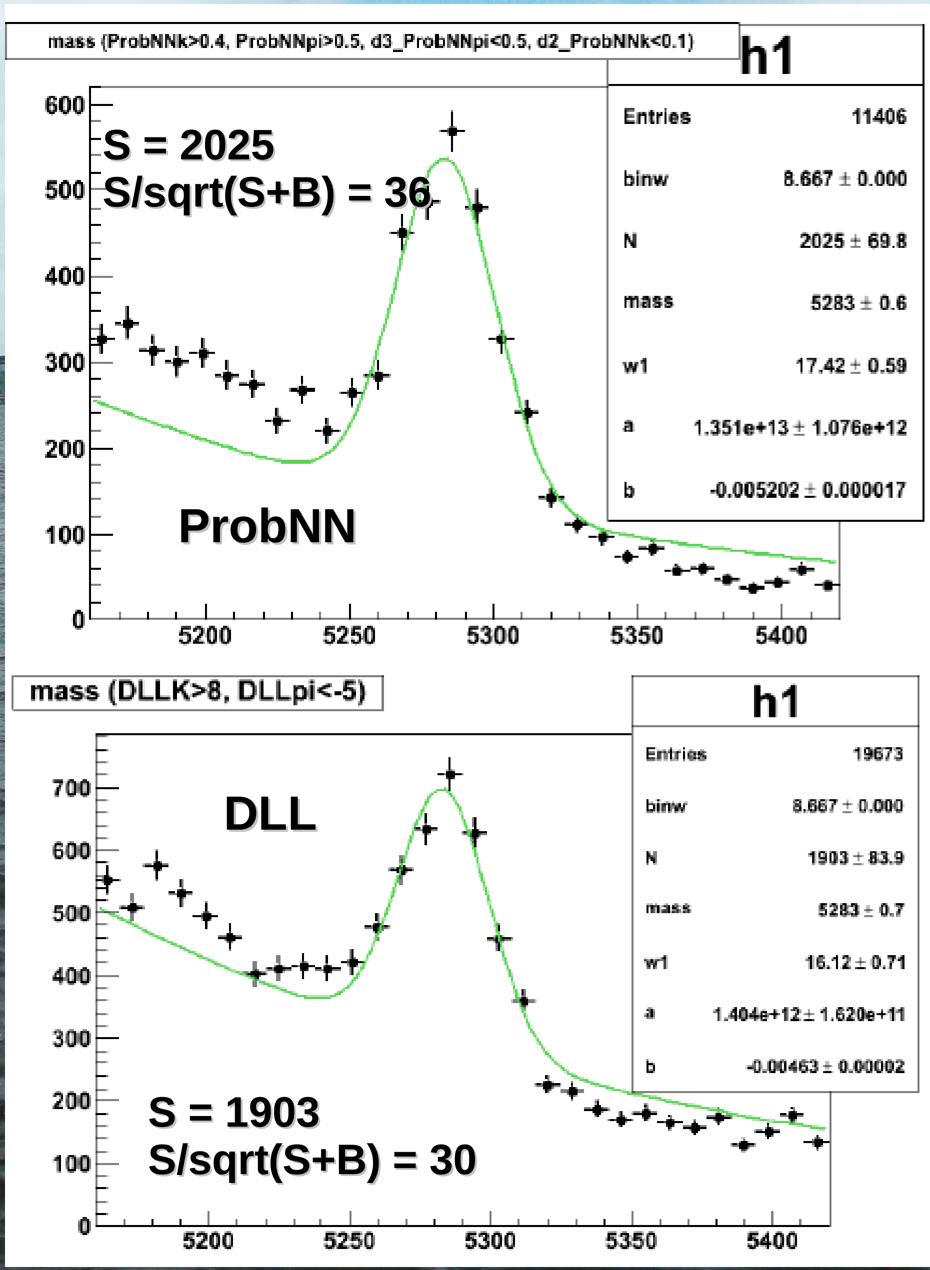




hhh analyses

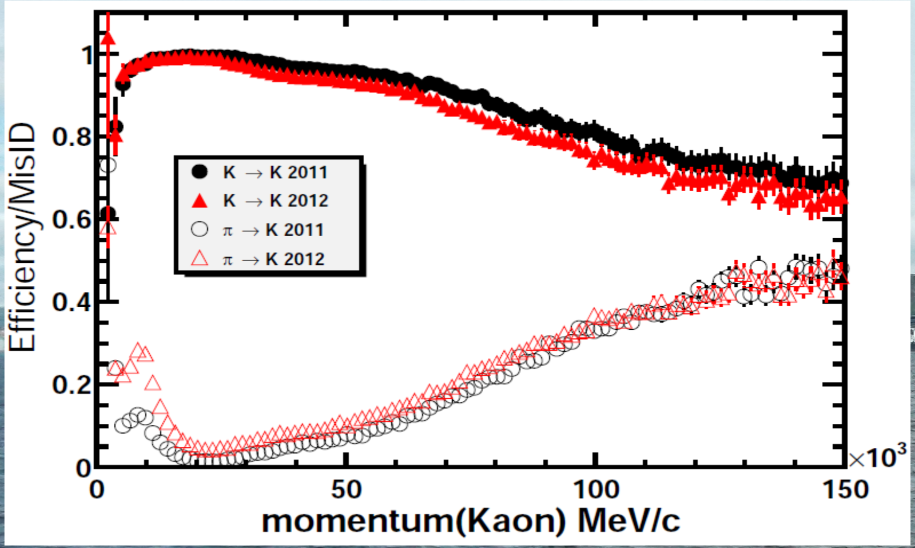
Illustration for KKπ 2012 (plots from Irina Nasteva)

- Many studies performed, testing PID against background and cross-feed
- The conclusions are that :
  - KKπ : ProbNN better
  - KKK and K$\pi\pi$: DLL and ProbNN ~ equivalent
  - $\pi\pi\pi$: DLL better
- For Kaon ID, ProbNNk seems enough but for Pion ID, we need both ProbNNpi and ProbNNk
- Using combinations of ProbNNs does not bring improvement
- Finally : chose ProbNN to follow the trend of changes, pending possible further improvements

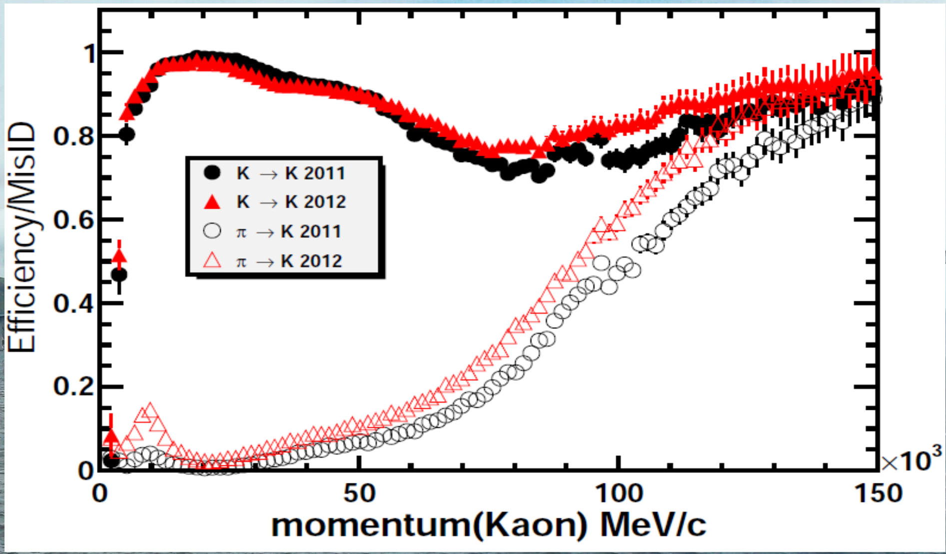


K for p$\bar{p}$K (1)

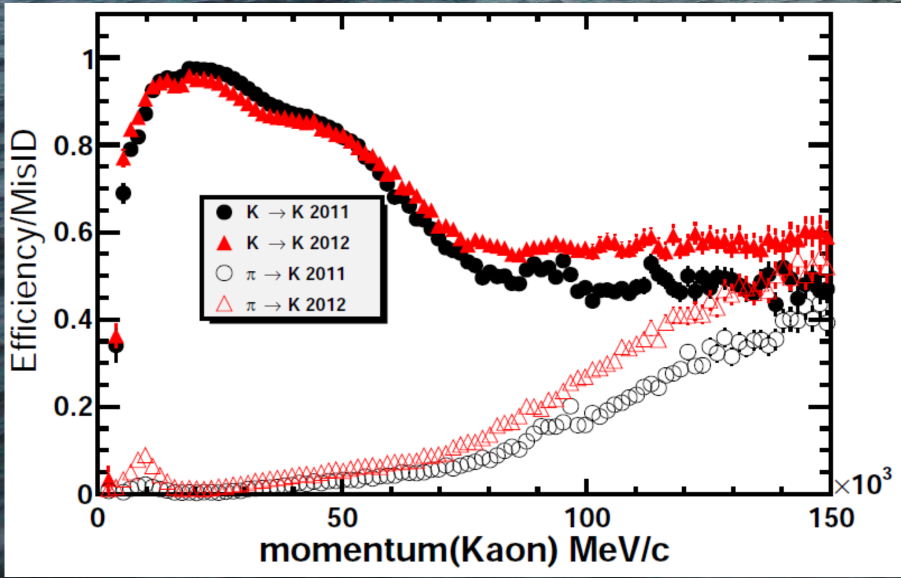
DLL cut (K-π>0)



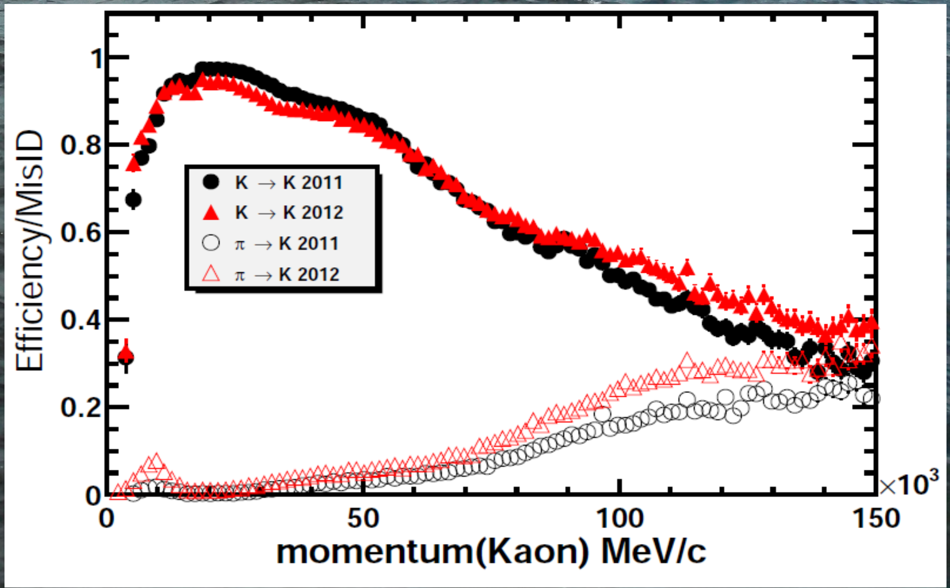
ProbNNk>0.1 & ProbNNpi<0.8



ProbNNk>0.2 & ProbNNpi<0.7

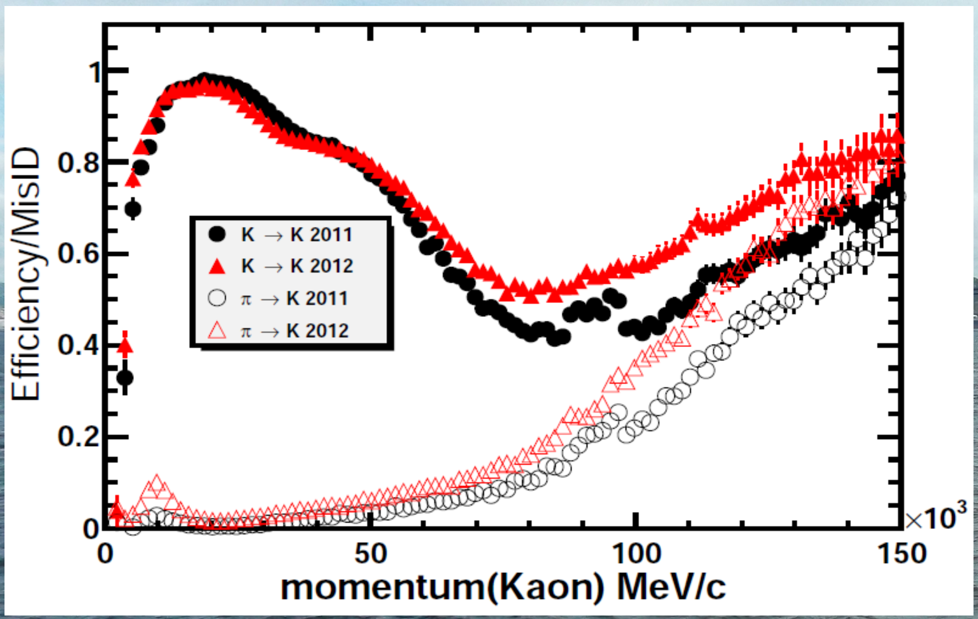


ProbNNk>0.25 & ProbNNpi<0.8

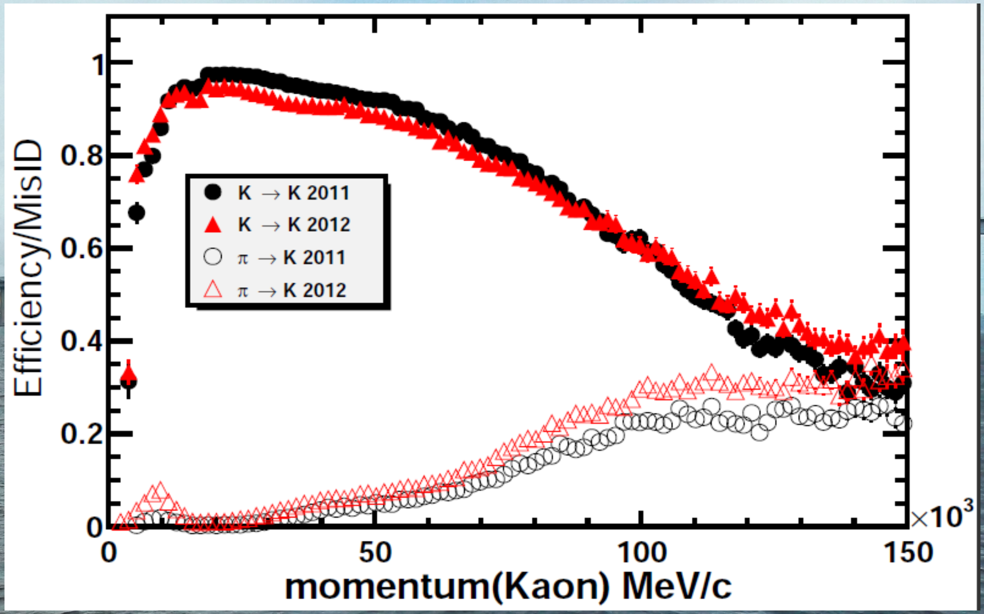


K for $p\bar{p}K$ (2)

ProbNNk>0.1 & ProbNNpi<0.6

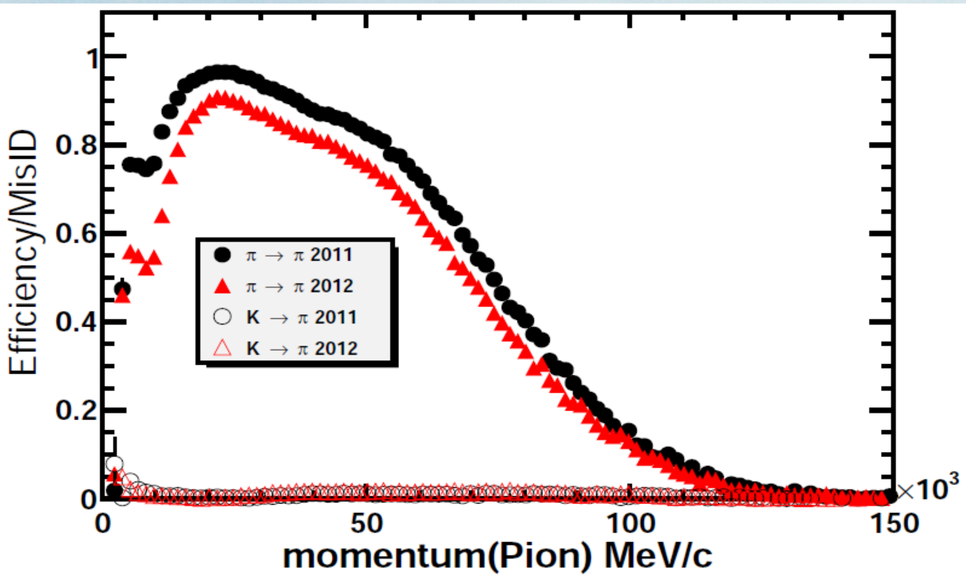


ProbNNk>0.25

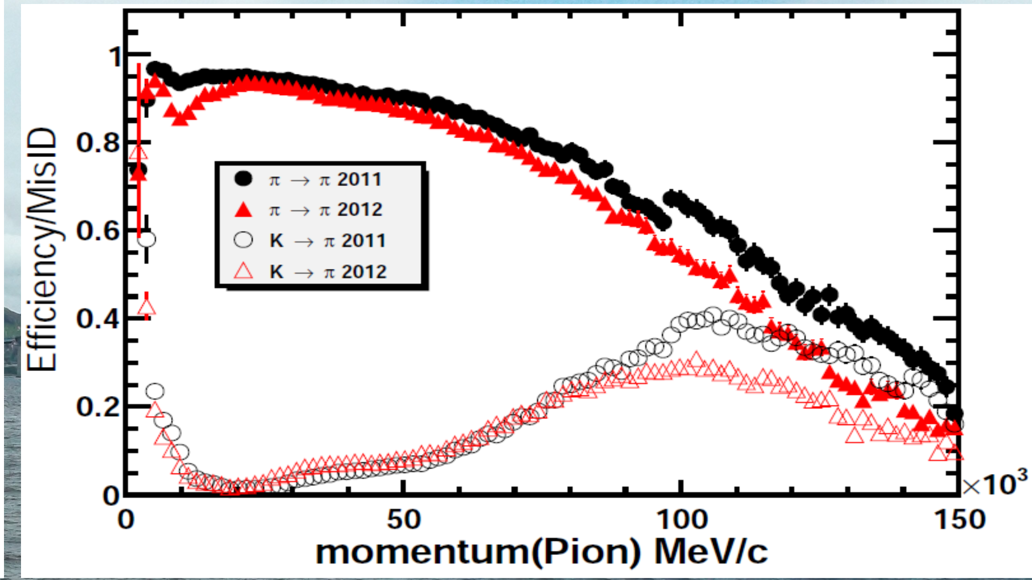


π for $p\bar{p}\pi$ (1)

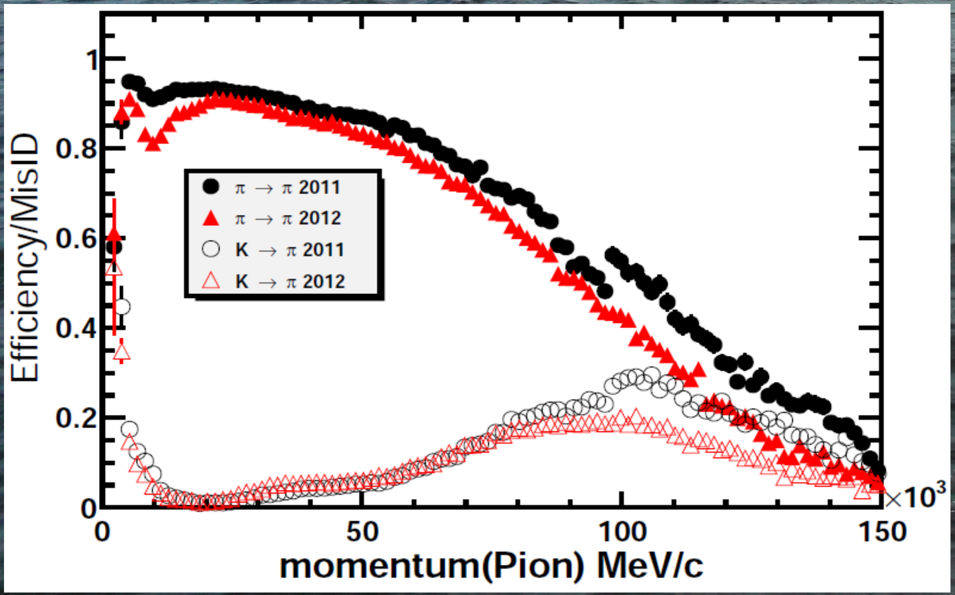
DLL cut (K-π<–5)



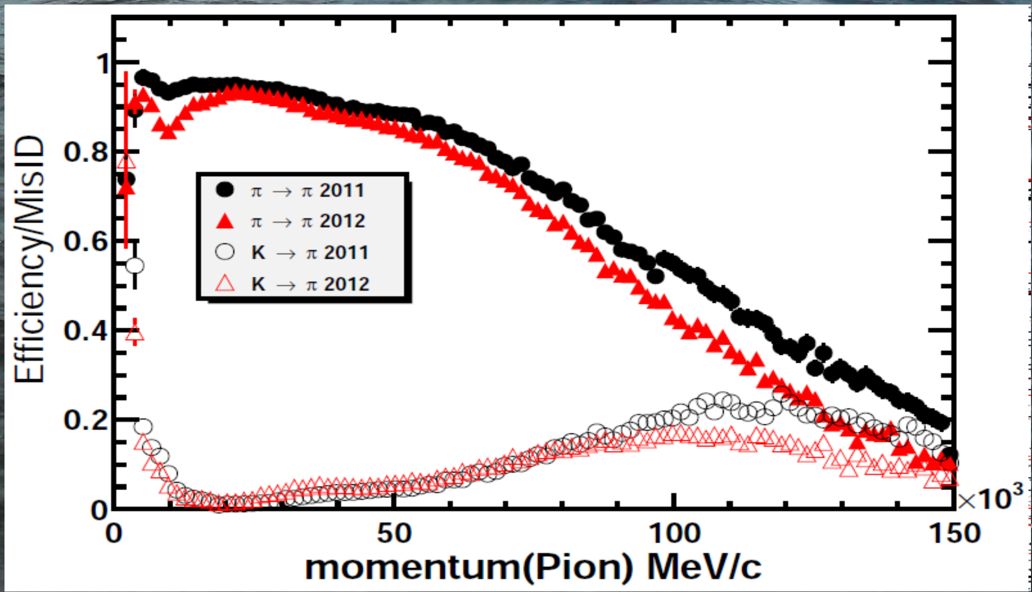
ProbNNk<0.4 & ProbNNpi>0.6



ProbNNk<0.4 & ProbNNpi>0.7



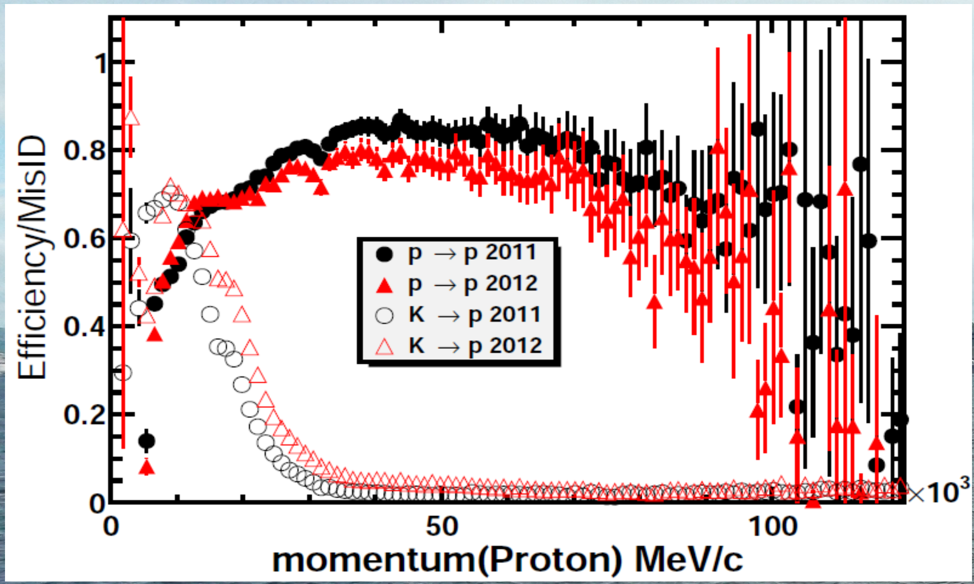
ProbNNk<0.25 & ProbNNpi>0.6

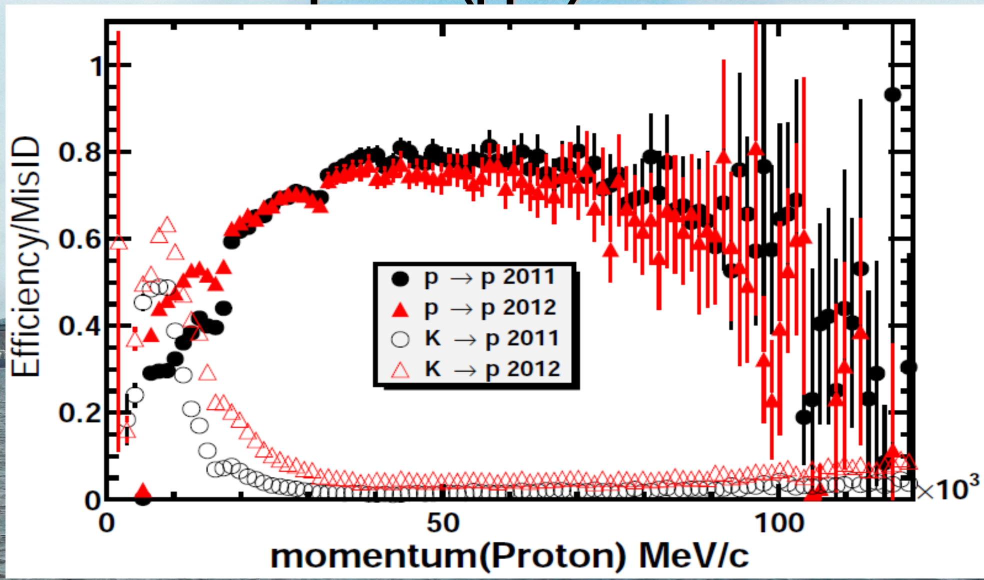


Proton ($p\bar{p}h$)

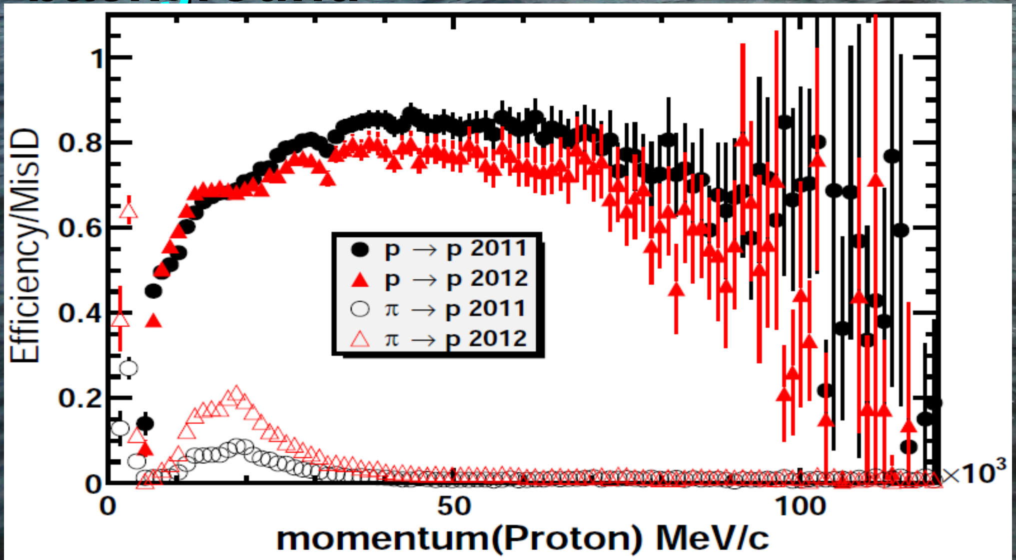
DLL cut $1/\sqrt{2}(p\text{-}K+p\text{-}\pi)>10$

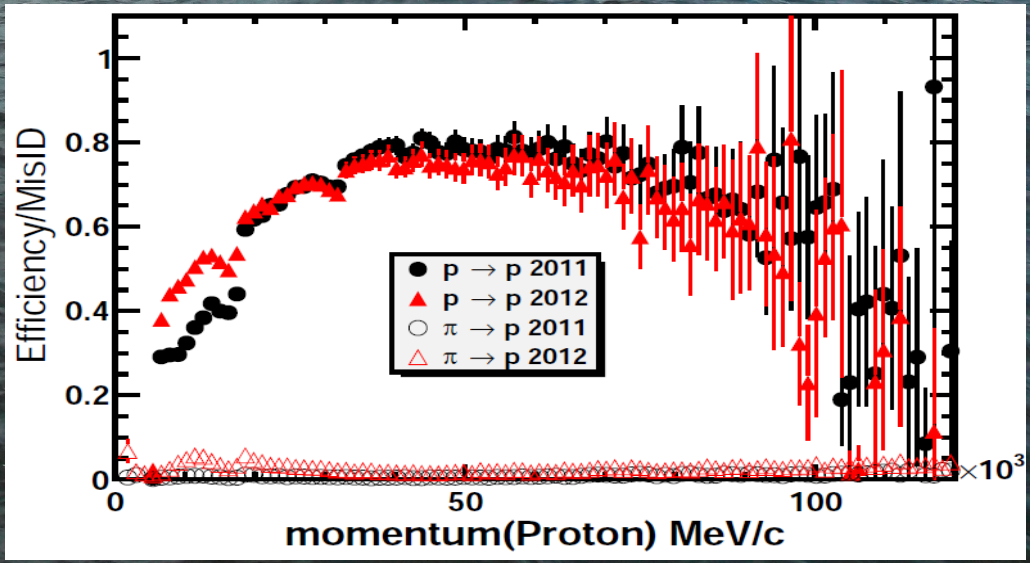
ProbNNp>0.2 & ProbNNk<0.8 & ProbNNpi<0.8 (ppK)





For similar efficiency, ProbNN has lower low-momentum misIDed background

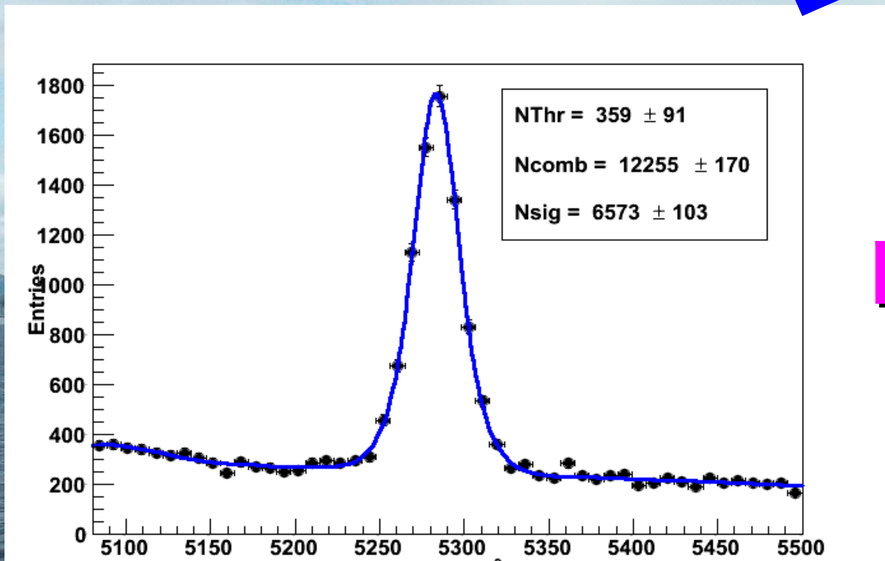


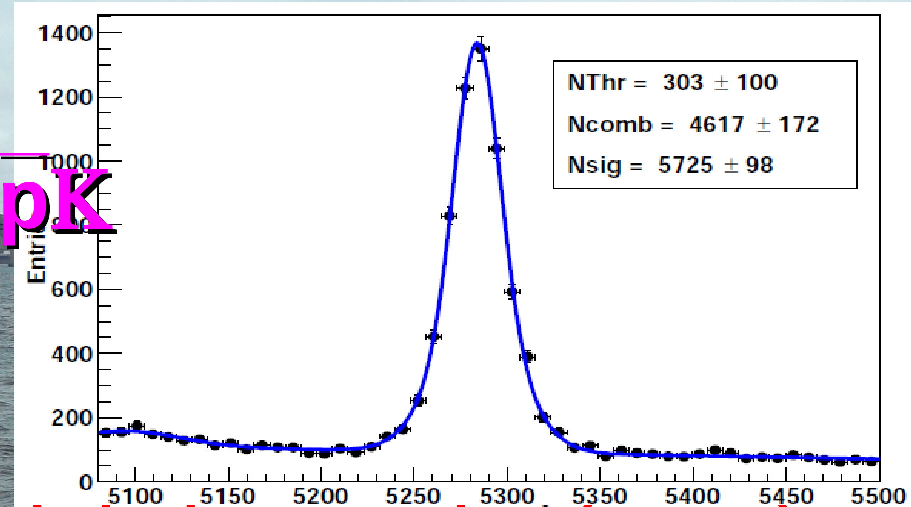


Consequence on $p\bar{p}h$ signals

(example with 2011 data, unique BDT cut)

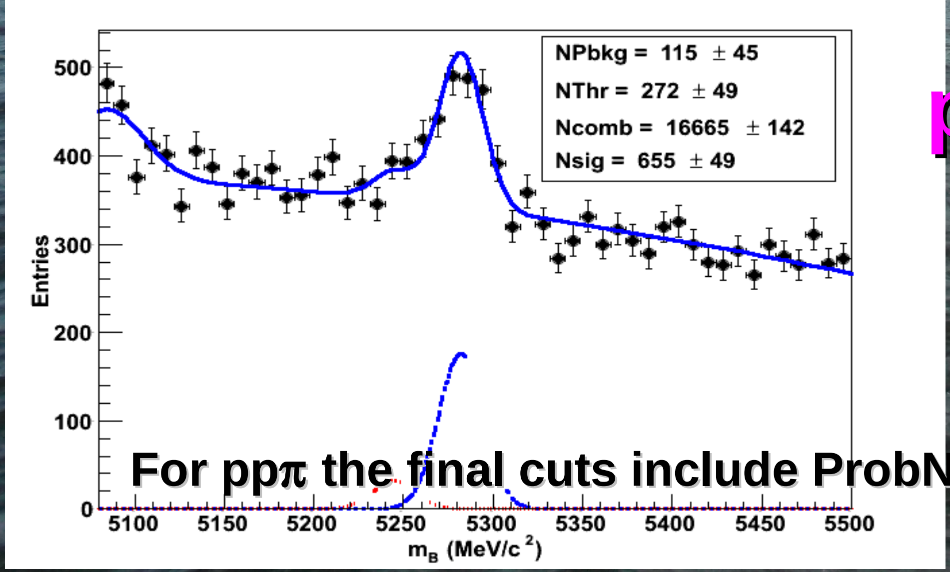
DLL cuts → **ProbNN cuts**

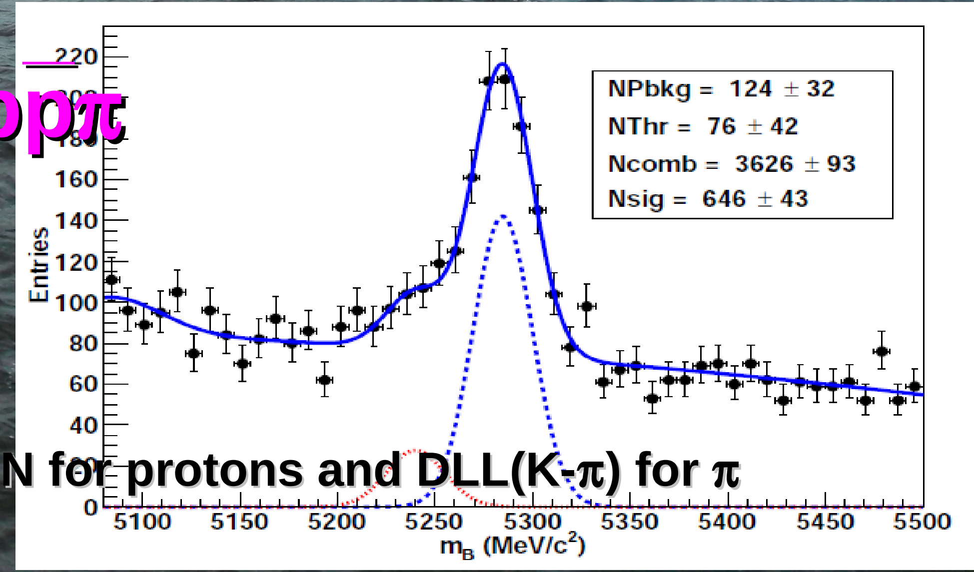




$p\bar{p}K$

Investigations have shown that the gain is due to ProbNN(proton) cuts





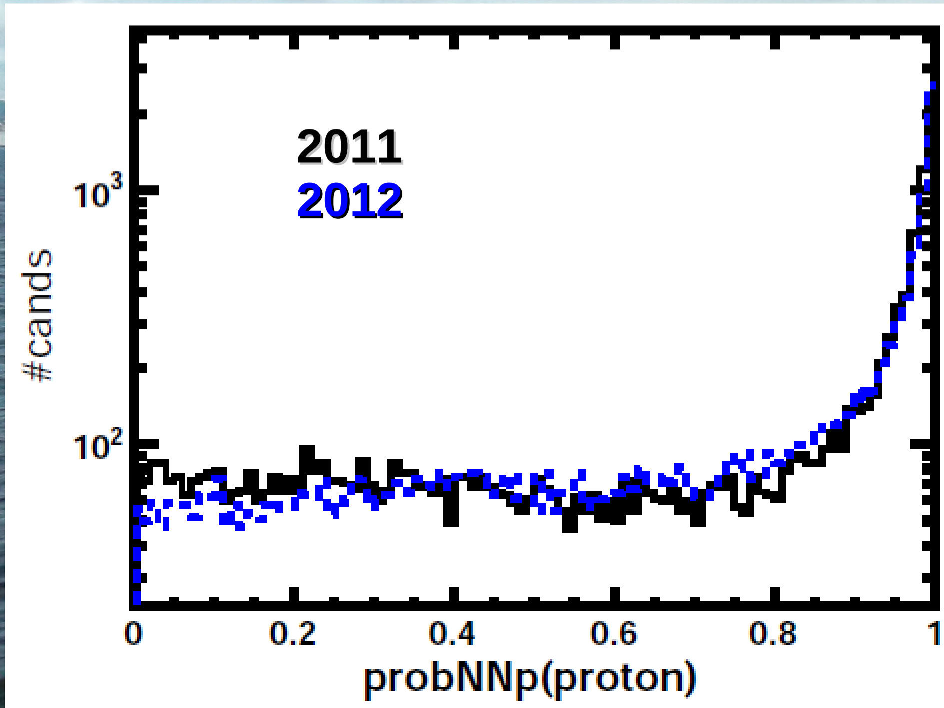
$p\bar{p}\pi$

For $pp\pi$ the final cuts include ProbNN for protons and DLL(K-π) for π

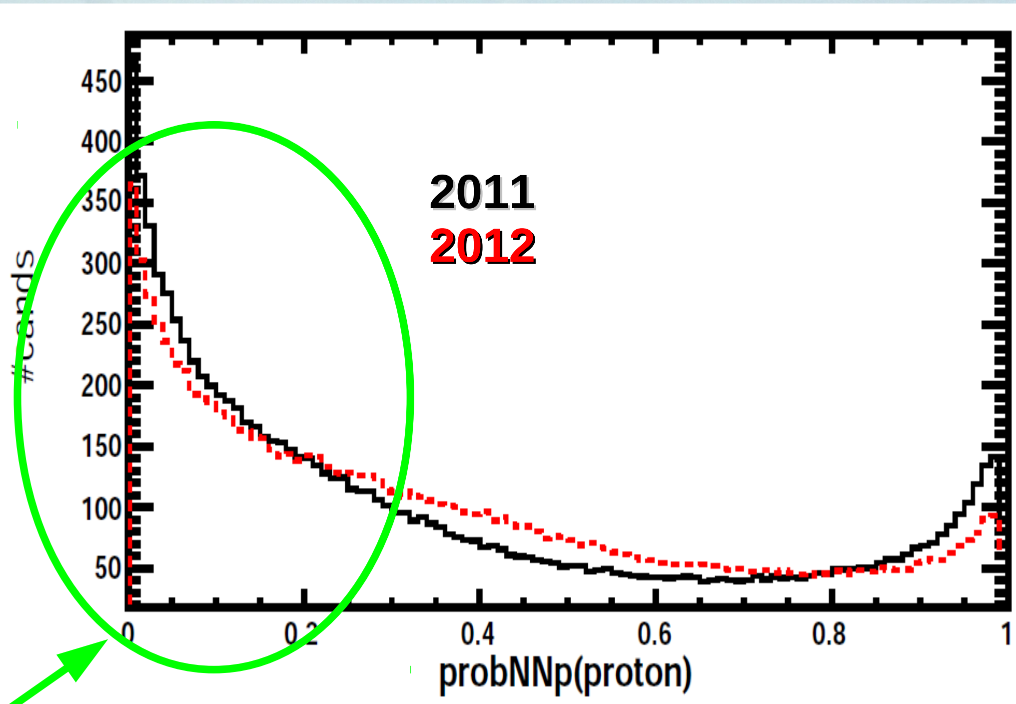
ProbNNp shape in data

Something that has « chocked » many users of calibration samples : more protons < 0.5 than >0.5, no peaking structure near 1

In ppK simulation



In calibration sample Λ^0



Due to protons of momentum < 20 GeV/c
« Poorly distinguishing» RICH information at low momentum
But still purity vs ProbNNp shows that this variable is sensible and interpretable as a probability (see backup)
n.b : output from MVA, used to be from NeuroBayes

Conclusion

- **Use of ProbNN spreads in the collaboration**
  - **All track types tested/studied except electrons**
- **The relevance of choosing ProbNN vs DLL depends on the analysis/channel but there are general emerging trends :**
  - **ProbNNK is « enough » for K ID (i.e ProbNNpi not much useful) , does a similar job as DLL(K-π)**
    - **For dedicated pion-driven cross-feed, DLL seems to be preferred**
  - **For π ID, one needs ProbNNpi and ProbNNK. DLL(K-π) seems more performant to get rid of kaon-generated cross-feed**
  - **ProbNN(p,K,π) helps a lot to remove (low-momentum) background in proton ID : ProbNN does better than DLL here, *hope for improvements for the efficiency***
  - **Consensus on ProbNNmu being more performant that DLLmu (in particular for Kaons rejection)**
- **People are encouraged to use new PID variables to consolidate our experience/understanding and to find the best ways to use them**
  - **Feedback to PID mailing list is very important both for experts and non-experts**

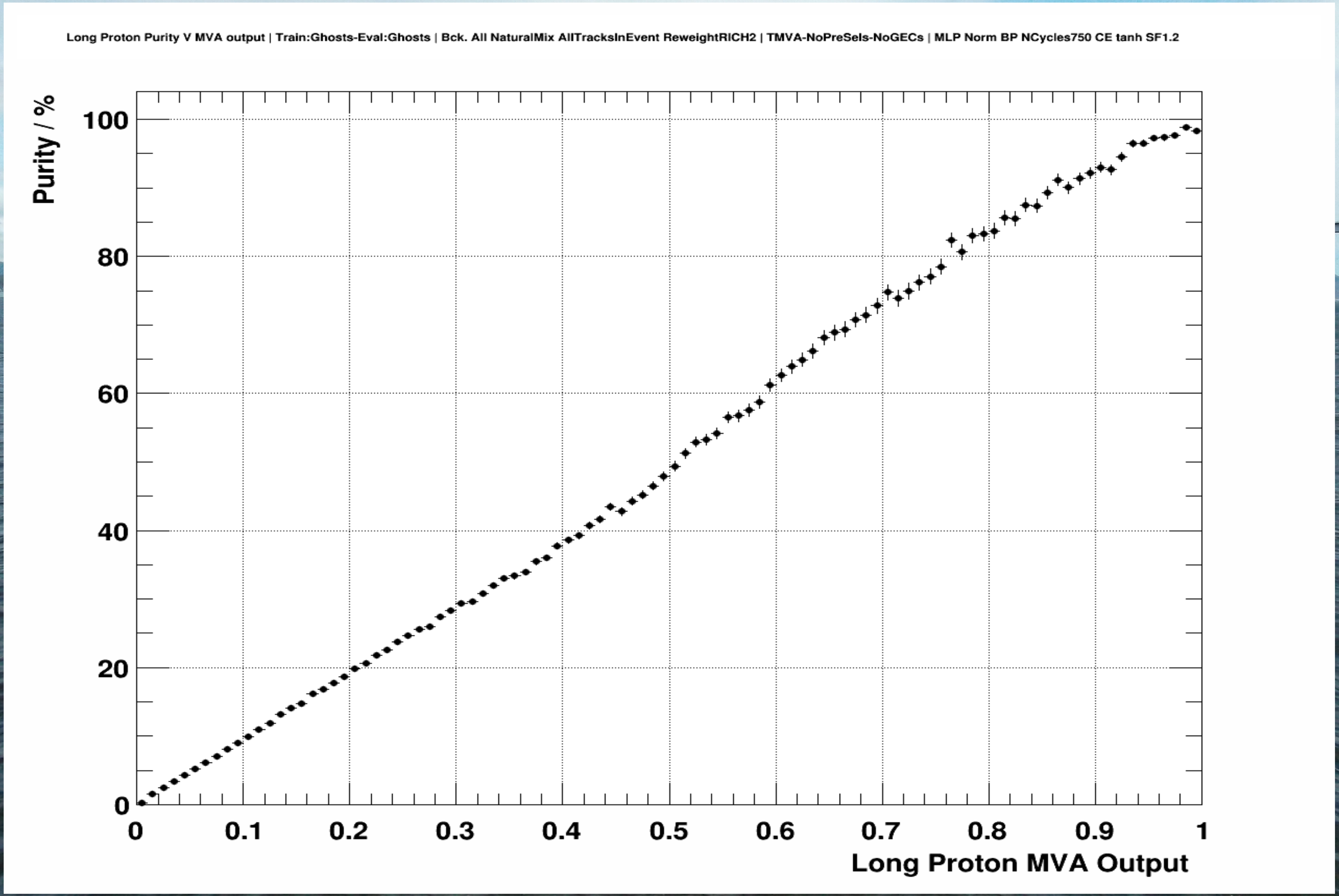
ProbNN training

(from C.Jones 05/2013 presentation)

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

Purity vs ProbNNp (plot from C.Jones)



ProbNN vs proton momentum in calib data

(taken from Till Moritz Karbach, discussion in PID mailing list)