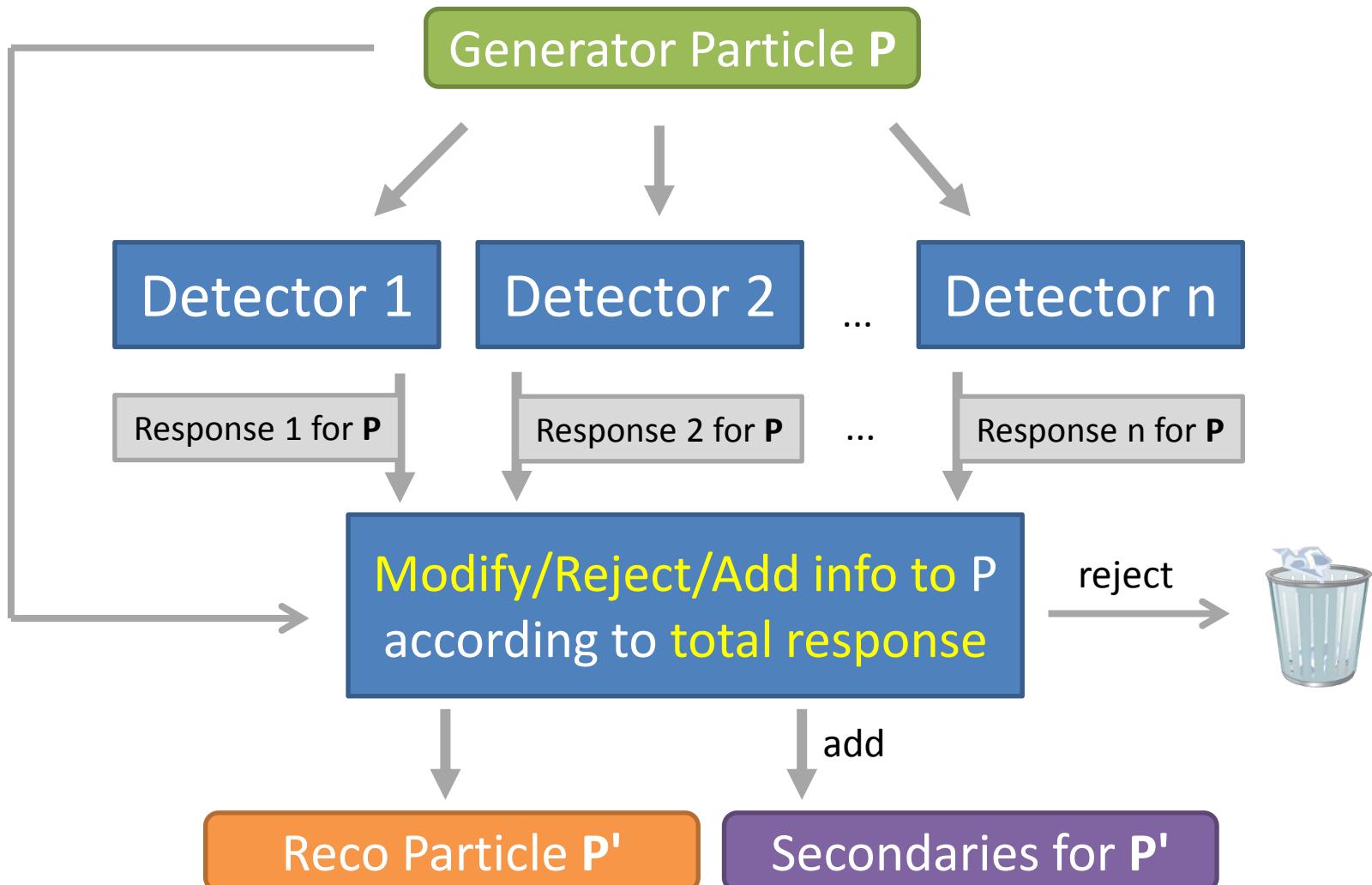


# PID Parametrizations in PANDA Fast Simulation

*PID Workshop GSI  
May 2017*

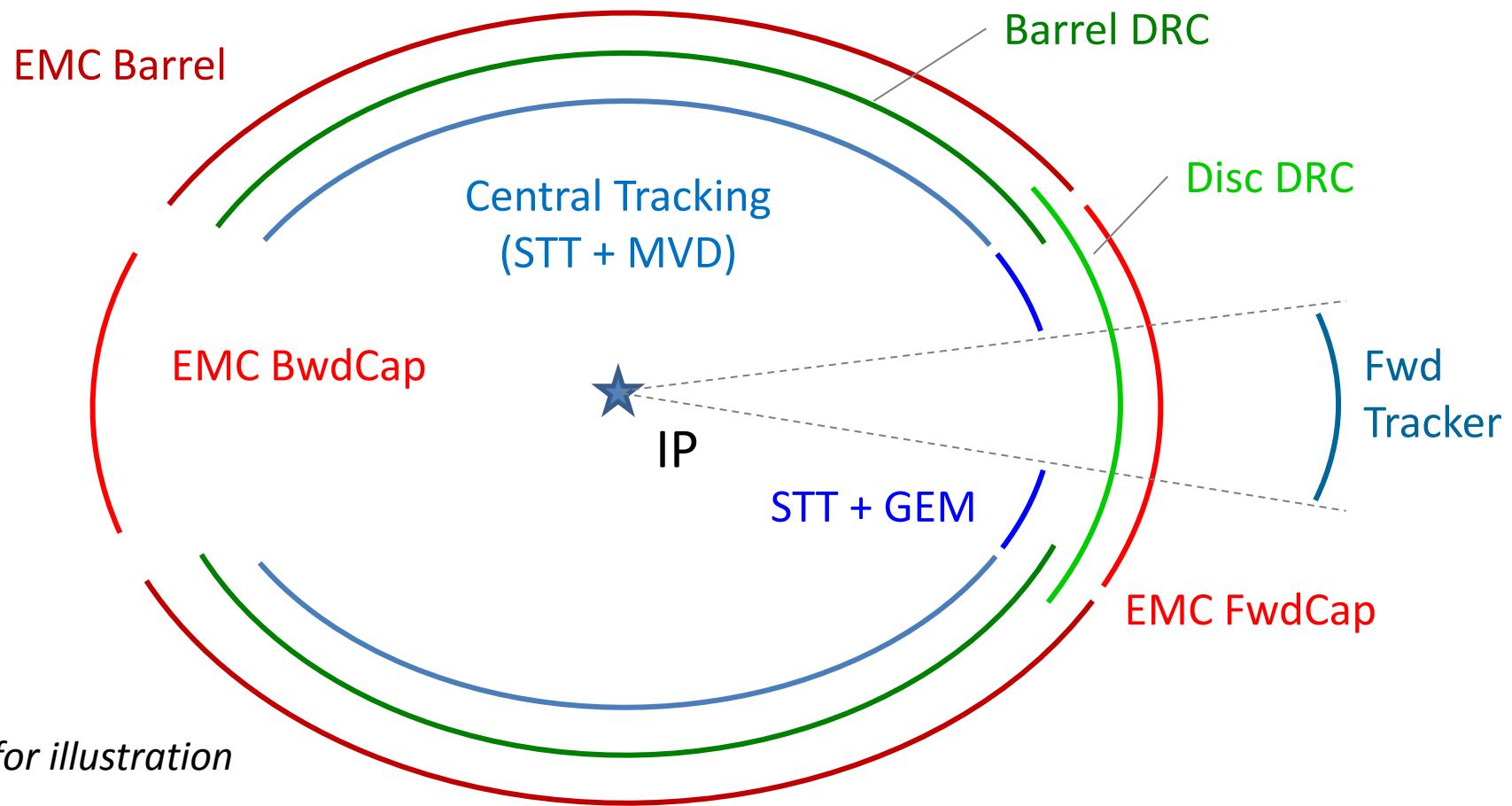
K. Götzen  
GSI Darmstadt

# Concept



# Fast Sim Detector Concept

- Spatial acceptance typically defined by polar  $\theta$  range from IP
- Different trackers or EMCs should not overlap



# Detector Modules

## Target Spectrometer

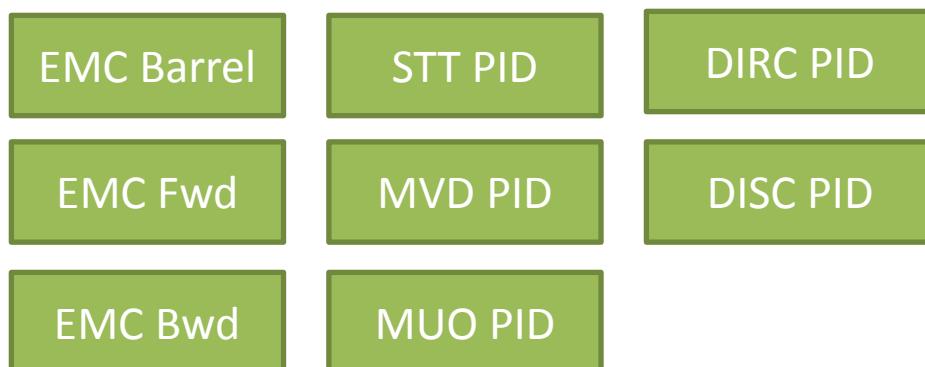
TRK



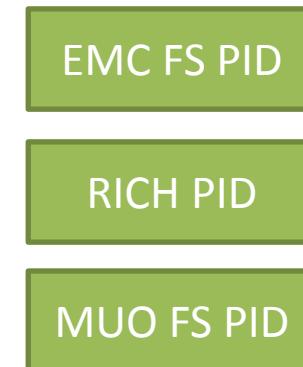
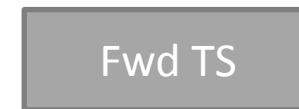
EMC



PID



## Forward Spectrometer



# PID Simulation in general

General strategy for PID simulation:

1. Determine PDF distributions for all particle species with current particle properties ( $p$ ,  $\theta$ , ...)  
→ parametrized or sampled from Full Sim
2. Take random value from true distribution  
(e.g.  $dE/dx$  value from pion distribution)
3. Determine probabilities (value of PDF for that specific  $dE/dx$  value) for all five particle species
4. Return this set of five  $P_i$  as LH values for the particle

# PID Concept in PandaROOT

- Each PID subdetector delivers individual probability/likelihood information
- **PID Combiner**, which combines the likelihood values from different detectors

$$\tilde{P}_{tot,i} = \prod_{Det} P_{Det,i} \quad i \in \{e, \mu, \pi, K, p\}$$

- Normalization via

$$P_{tot,j} = \frac{\tilde{P}_{tot,j}}{\sum_i \tilde{P}_{tot,i}}$$

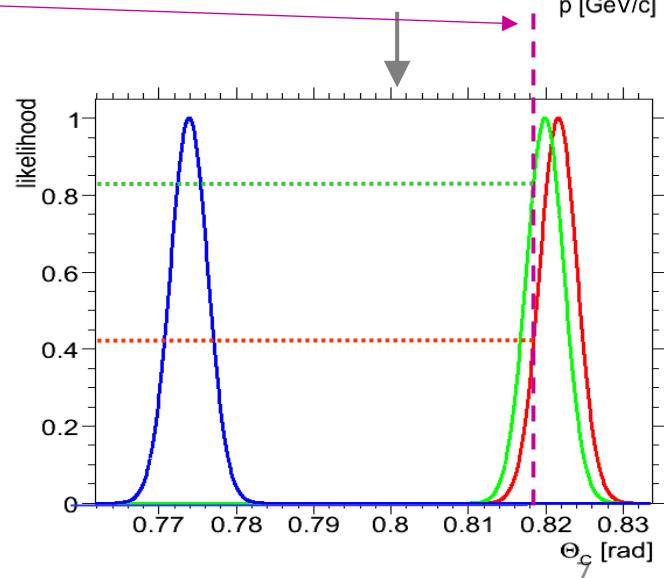
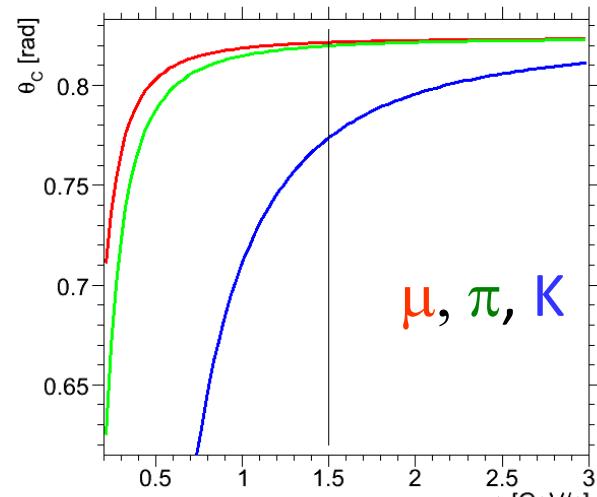
- **PID Selector**, which requires certain  $P_{tot}$  for positive identification

# PID Simulation: Cherenkov Detectors

- Compute number  $N$  of produced cherenkov photons for particular  $(p, \theta, \varphi)$  in radiator
- Resolution  $\sigma = \sigma_0 / \sqrt{N \cdot \varepsilon}$   
(single phot. res.  $\sigma_0 \approx 10$  mrad, efficiency  $\varepsilon \approx 0.1$ )
- Compute PDFs for all hypothesis with expected cherenkov angles for given momentum
- Take random value from correct PDF distribution
- Compute value for all PDFs at this position

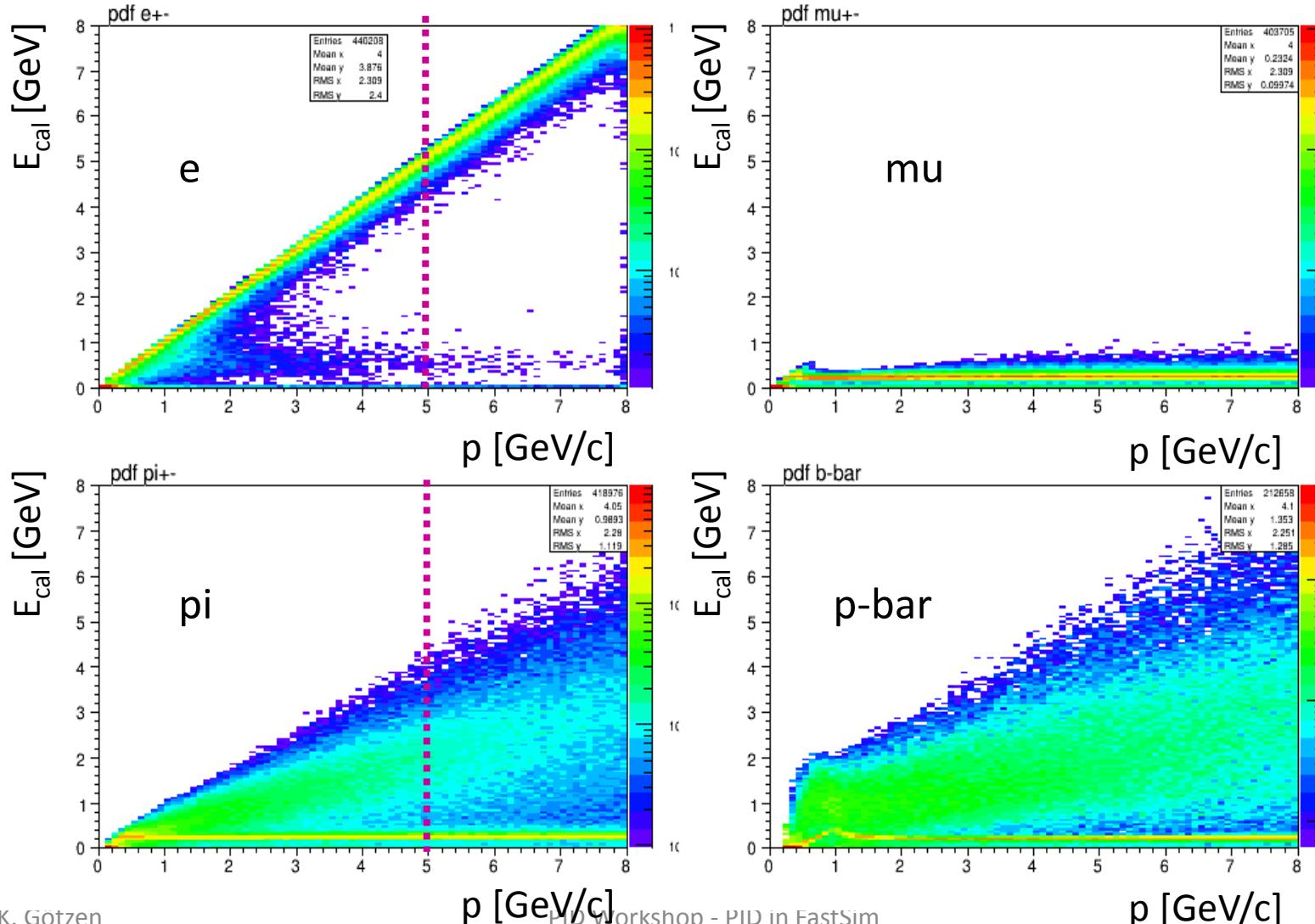
In this example: true particle type  $\mu$

$\Rightarrow LH(\mu) = 0.42$ ,  $LH(\pi) = 0.84$ ,  $LH(K) = 0.0$



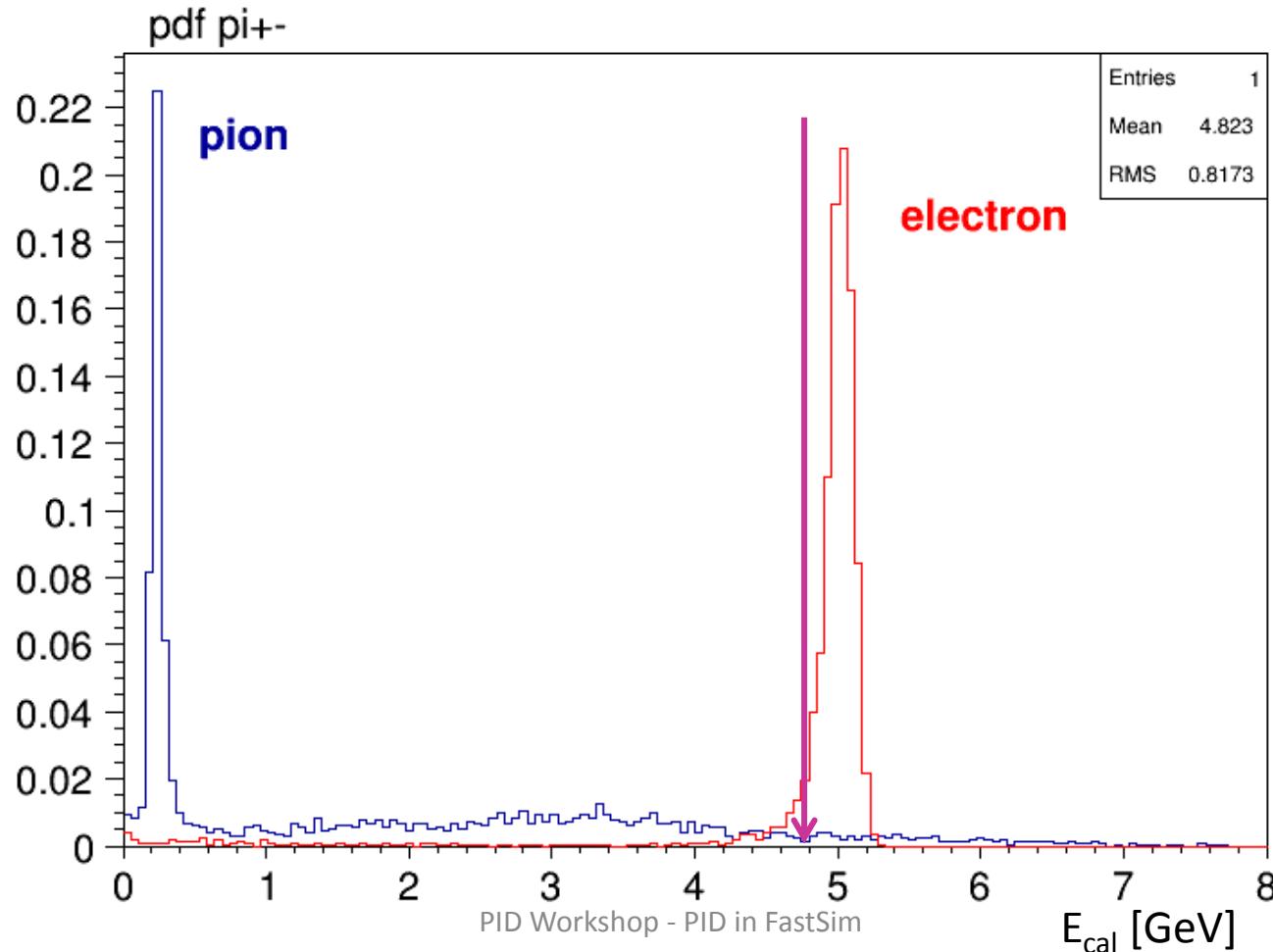
# PID Simulation: EMC

- EMC PID based on  $E_{\text{calib}}$  vs  $p$  (*from full sim*); *slice = pdf( $p$ )*



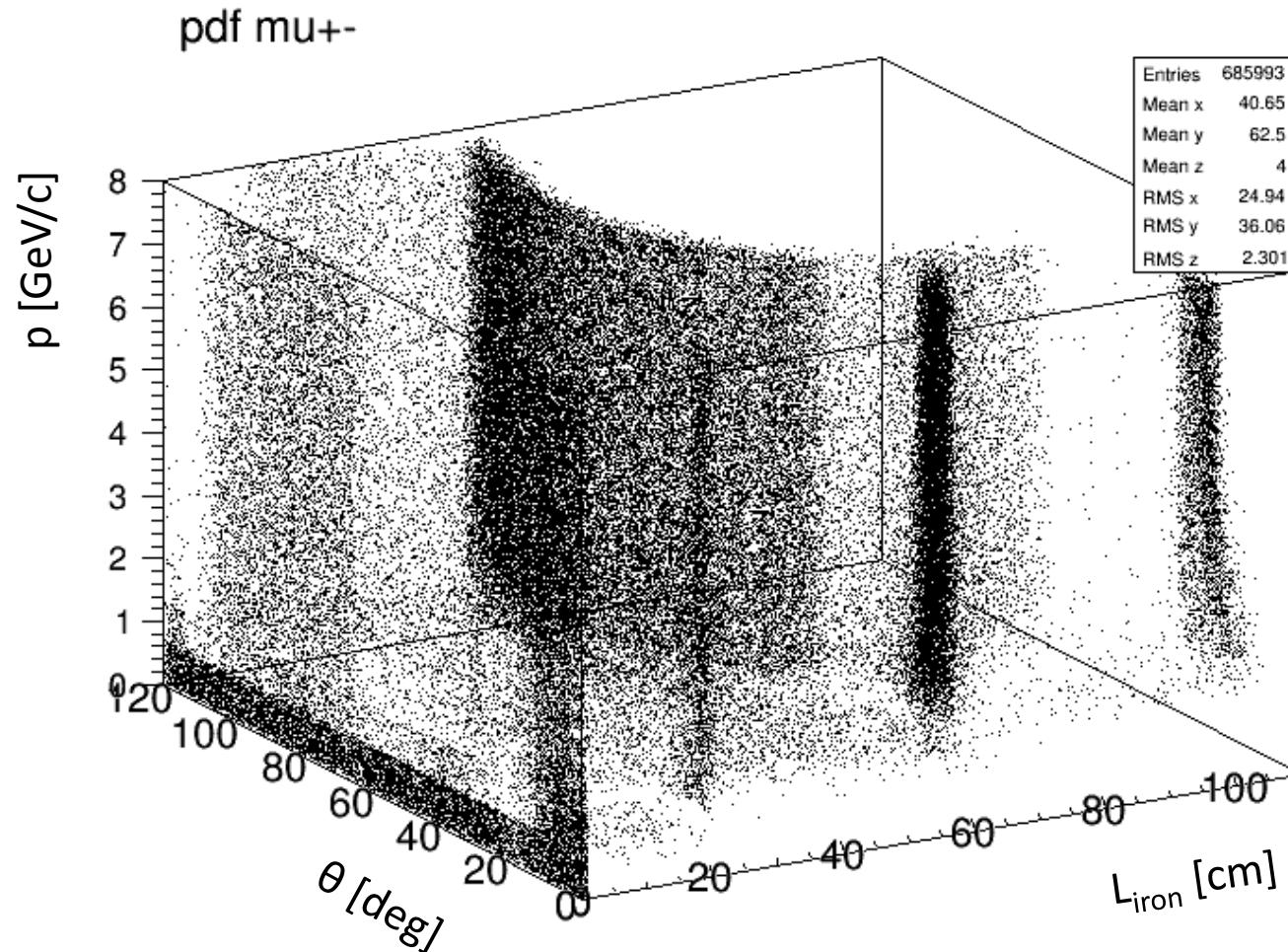
# PID Simulation: EMC

- Select slice according to track  $p$  (e.g. 5 GeV/c)
- Choose **random value** from true distribution (e.g. **electron**)
- Return for each species pdf value at that position



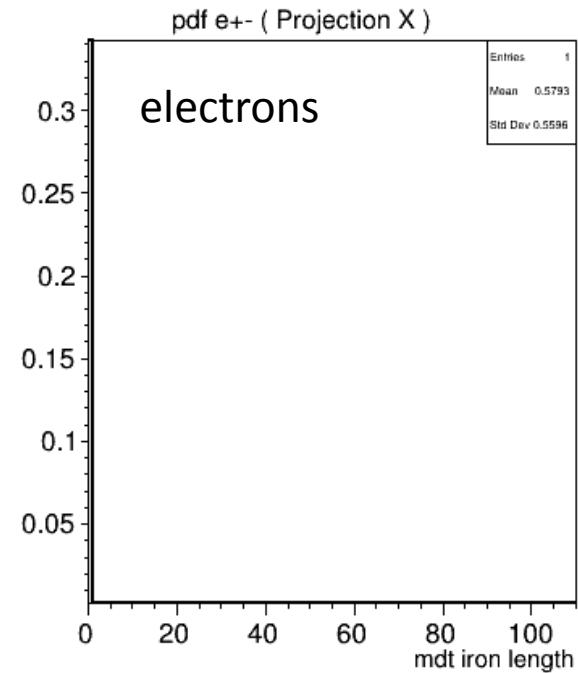
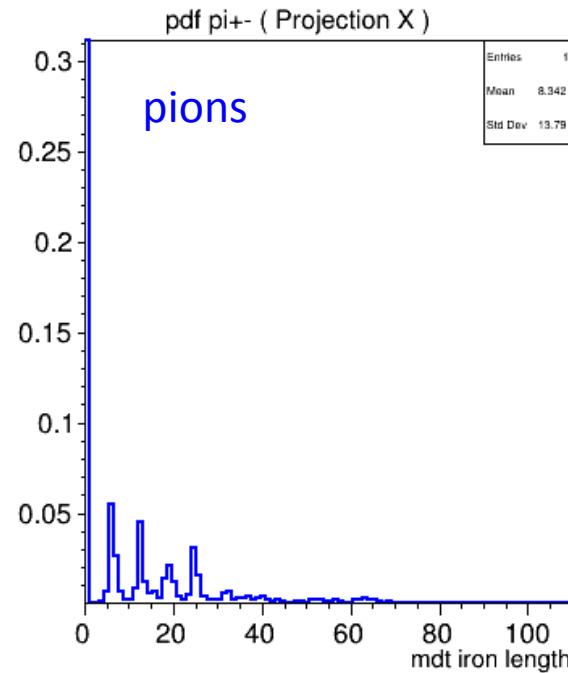
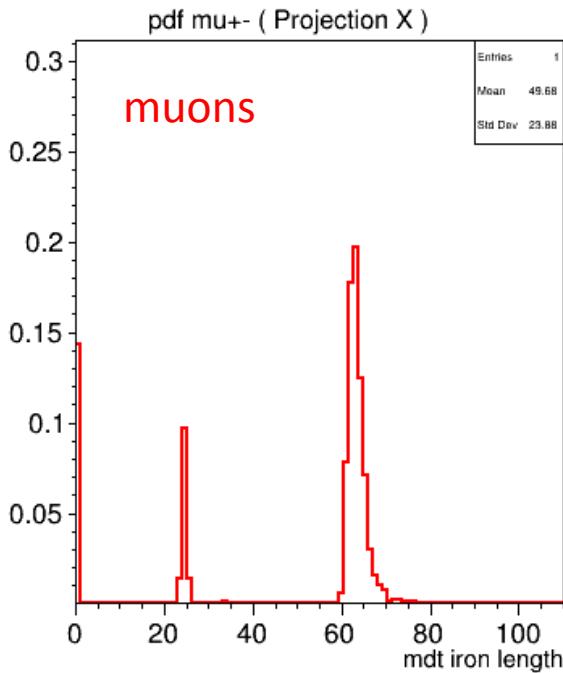
# PID Simulation: MDT

- MDT PID based on  $L_{\text{iron}}$  vs  $\theta$  vs  $p$  (*from full sim*)



# PID Simulation: MDT

- MDT PID based on  $L_{\text{iron}}$  vs  $\theta$  vs  $p$  (*from full sim*)
- PDFs for e, mu, pi, K, p, b-bar
- Take slices from 3D plots



# Performance: Raw PID

DIRC

STT

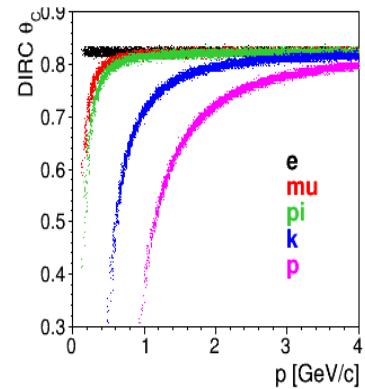
EMC

MDT

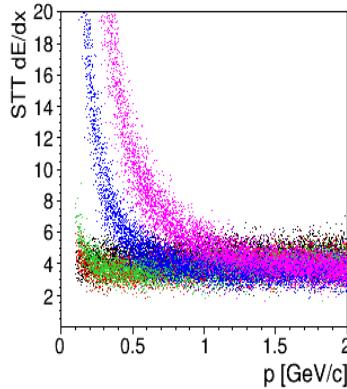
MVD

**FAST**

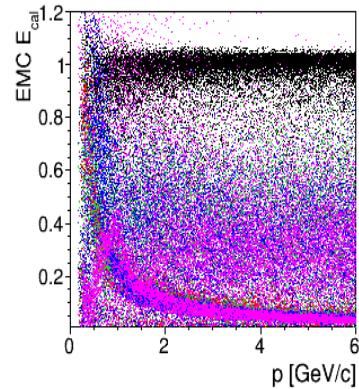
DIRC  $\theta_C$



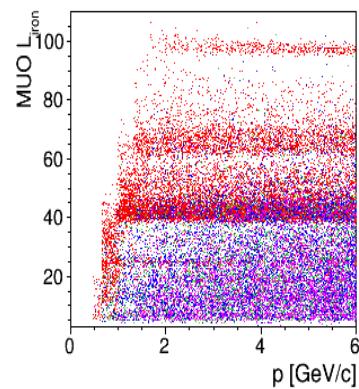
FAST: STT dE/dx



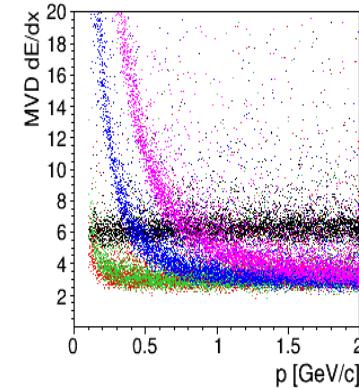
FAST: EMC  $E_{cal}$



FAST: MUO  $L_{iron}$

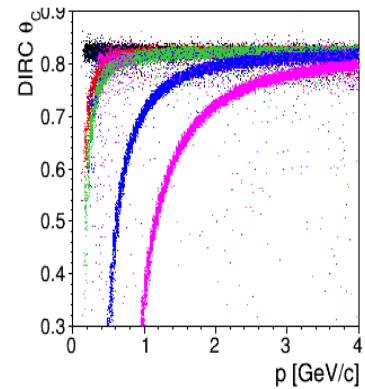


FAST: MVD dE/dx

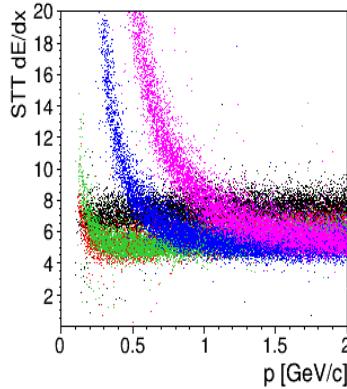


**FULL**

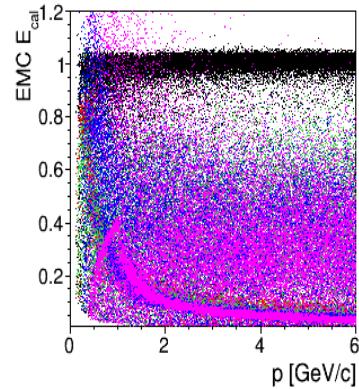
DIRC  $\theta_C$



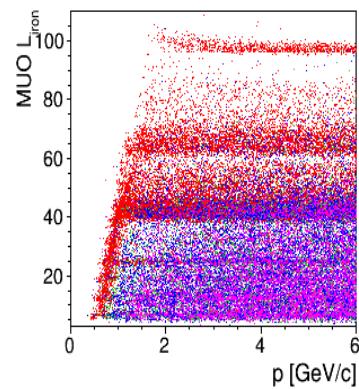
FULL: STT dE/dx



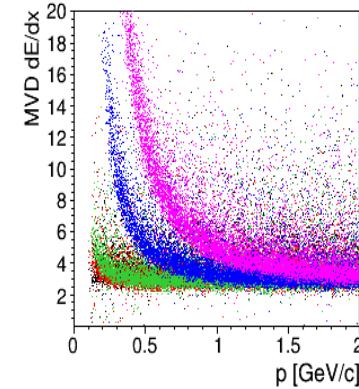
FULL: EMC  $E_{cal}$



FULL: MUO  $L_{iron}$



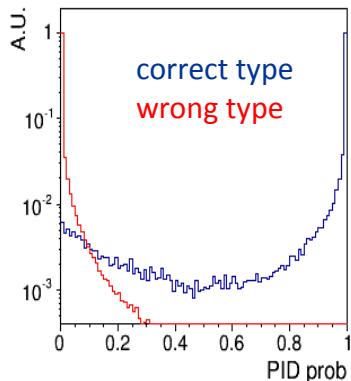
FULL: MVD dE/dx



# Performance: PID Probabilities

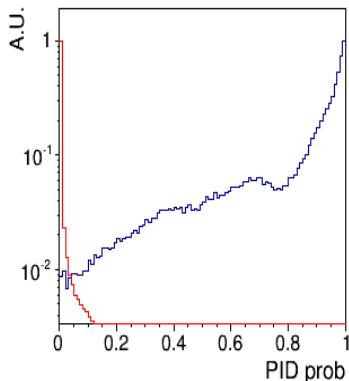
$P_{\text{electron}}$

**FAST** PID prob. e



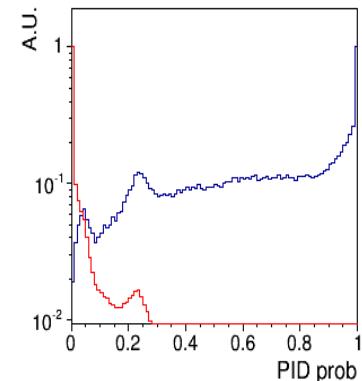
$P_{\mu\text{on}}$

FAST: PID prob. mu



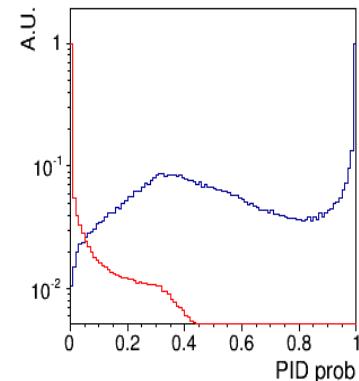
$P_{\pi\text{on}}$

FAST: PID prob. pi



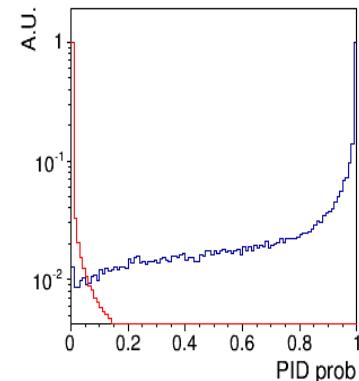
$P_{\text{kaon}}$

FAST: PID prob. k



$P_{\text{proton}}$

FAST: PID prob. p



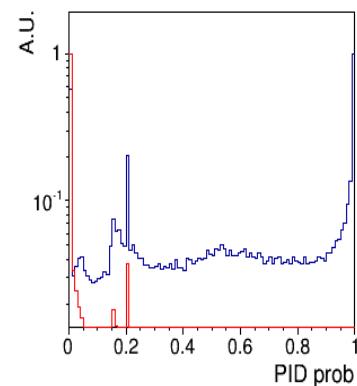
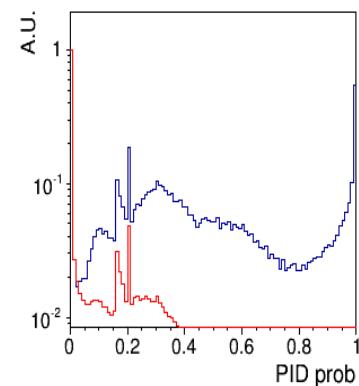
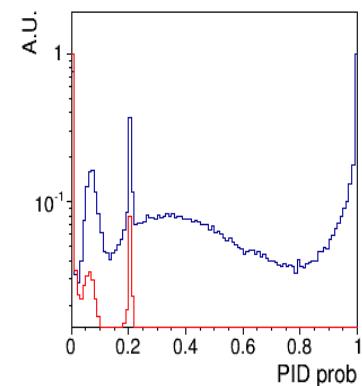
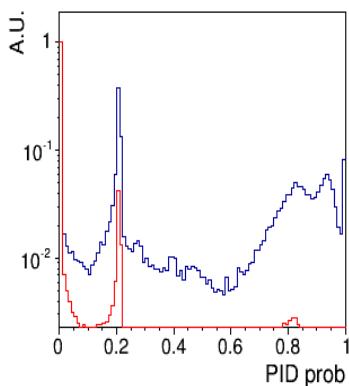
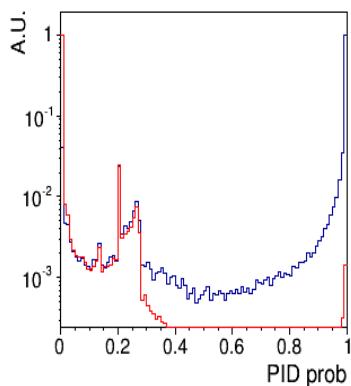
**FULL** PID prob. e

FULL: PID prob. mu

FULL: PID prob. pi

FULL: PID prob. k

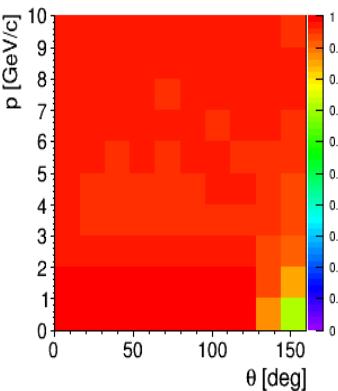
FULL: PID prob. p



# Performance: Efficiency (Best)

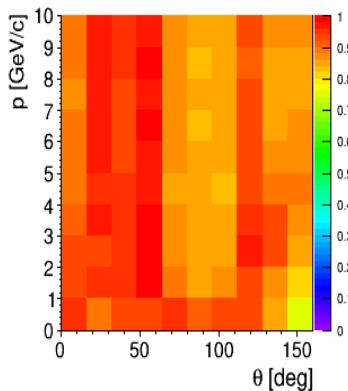
$\varepsilon_{\text{electron}}$

**FAST** PID eff e



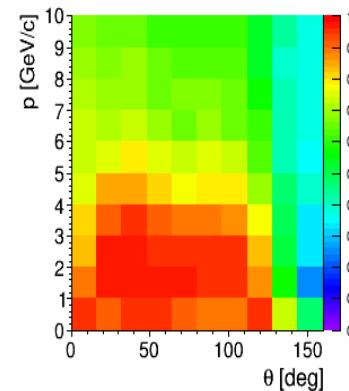
$\varepsilon_{\text{muon}}$

FAST: PID eff mu



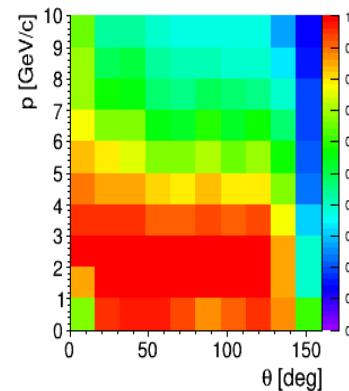
$\varepsilon_{\text{pion}}$

FAST: PID eff pi



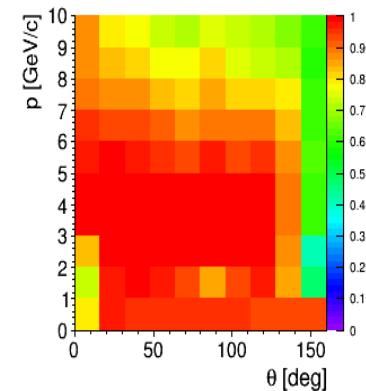
$\varepsilon_{\text{kaon}}$

FAST: PID eff k

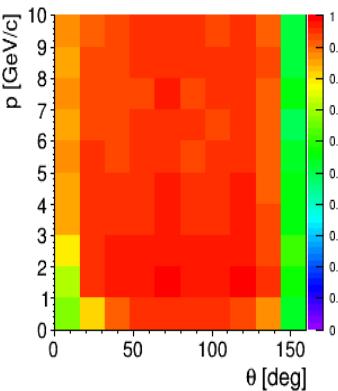


$\varepsilon_{\text{proton}}$

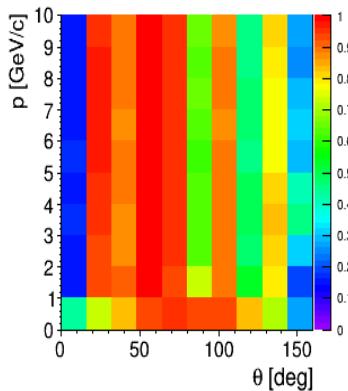
FAST: PID eff p



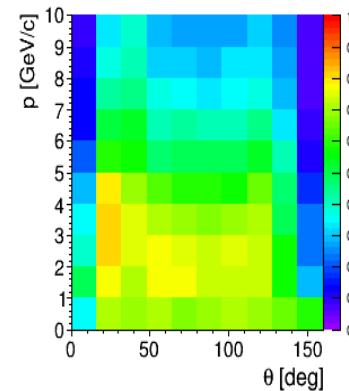
**FULL** PID eff e



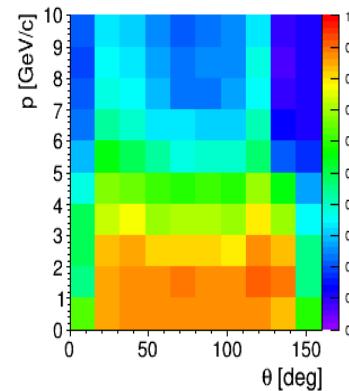
FULL: PID eff mu



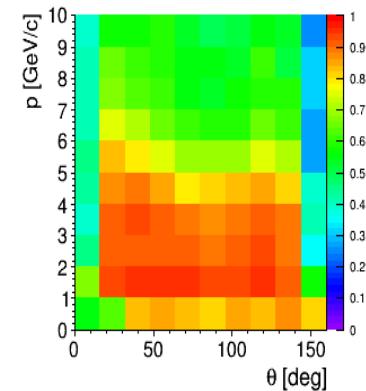
FULL: PID eff pi



FULL: PID eff k



FULL: PID eff p



# Performance: Mis-ID (Best)

mis<sub>electron</sub>

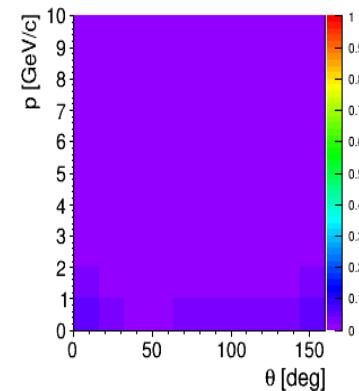
mis<sub>muon</sub>

mis<sub>pion</sub>

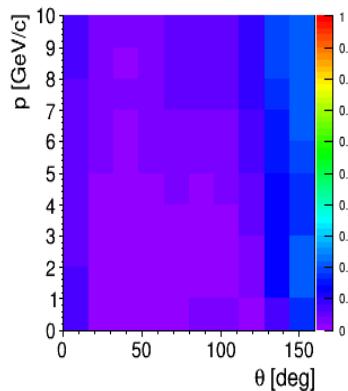
mis<sub>kaon</sub>

mis<sub>proton</sub>

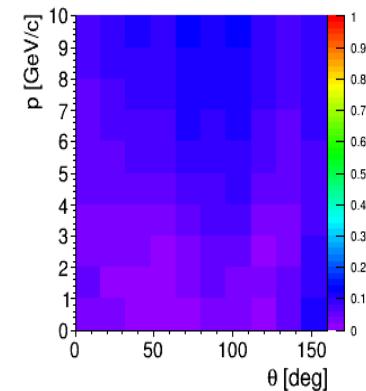
**FAST** : PID mis-id e



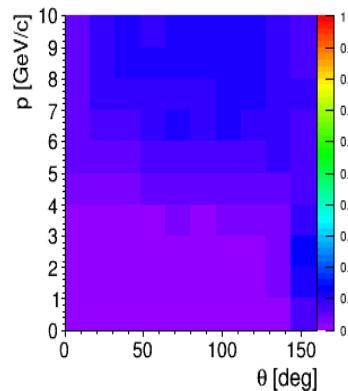
FAST: PID mis-id mu



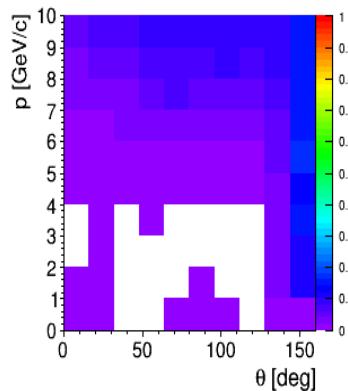
FAST: PID mis-id pi



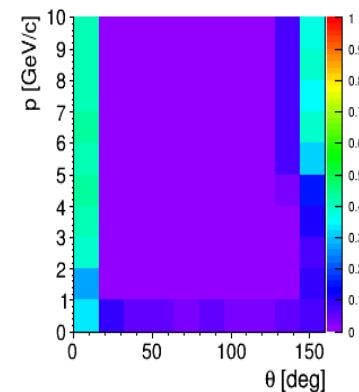
FAST: PID mis-id k



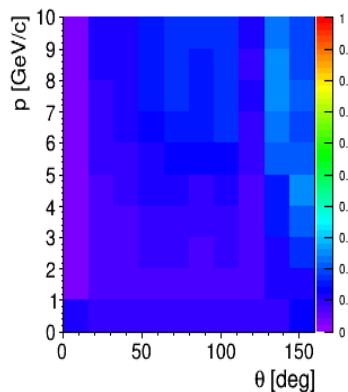
FAST: PID mis-id p



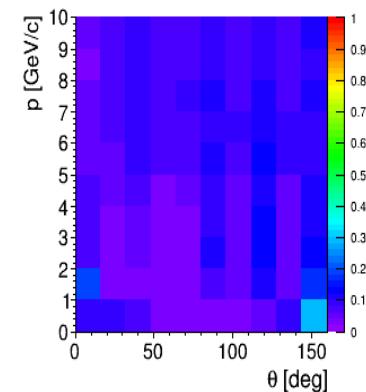
**FULL** PID mis-id e



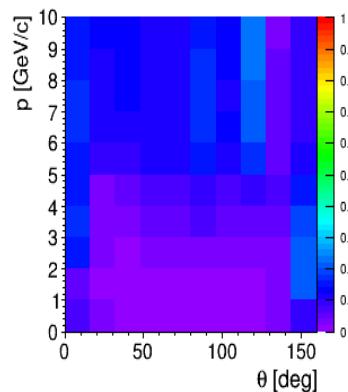
FULL: PID mis-id mu



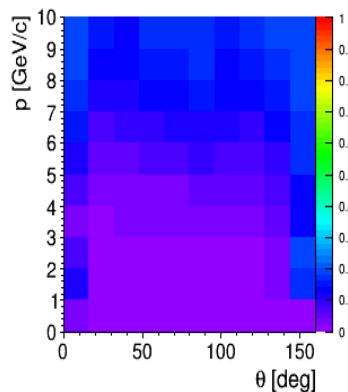
FULL: PID mis-id pi



FULL: PID mis-id k



FULL: PID mis-id p



# PID Concept in PandaROOT

```
FairRunAna *fRun = new FairRunAna();
fRun->SetInputFile("pid_complete.root");

PndAnalysis *ana = new PndAnalysis();
RhoCandList eplus, eminus, muplus, muminus;
TString myPidAlgosElectron = "PidAlgoEmcBayes;PidAlgoDrc"
TString myPidAlgosMuon      = "PidAlgoMdtHardCuts"
...
while ( ana->GetEvent() )
{
    ana->FillList( eplus,   "ElectronTightPlus",   myPidAlgosElectron );
    ana->FillList( eminus,  "ElectronTightMinus",  myPidAlgosElectron );
    ana->FillList( muplus,  "MuonTightPlus",       myPidAlgosMuon );
    ana->FillList( muminus, "MuonTightMinus",     myPidAlgosMuon );
...
}
```

Predefined selection with keywords (probability based)

Electron / Muon / Pion / Kaon / Proton +  
All / VeryLoose / Loose / Tight / VeryTight / Best +  
Plus / Minus (*optional*)

*Simple keywords*

Charged / Plus / Minus / Neutral / All

# BACKUP

# PID Concept in PandaROOT

PandaROOT objects: **PndAnaPidCombiner**, **PndAnaPidSelector**

- **PndAnaPidCombiner**
  - combines *on demand* probabilities from **various algorithms** by computing product of all  $P_k$  ( $k=algorithms$ )
  - copies resulting probabilities to RhoCandidate/RhoCandList
- **PndAnaPidSelector**
  - selects particles based on these probabilities
- **PndAnalysis::FillList** is a short-cut to this functionality via

```
pndana.FillList(list, "ElectronLoose", "PidAlgoEmcBayes;PidAlgoDrc");
```

- Predefined selection with keywords (probability based)

Electron / Muon / Pion / Kaon / Proton +  
All / VeryLoose / Loose / Tight / VeryTight / Best +  
Plus / Minus (optional)

*Simple keywords*

Charged / Plus / Minus / Neutral / All

# PID Concept in PandaROOT

- PandaROOT objects:

**PndAnalysis, PndAnaPidCombiner, PndAnaPidSelector**

```
PndAnalysis *pndana= new PndAnalysis();
pndana->FillList(eplus, "ElectronLoosePlus", "PidAlgoEmcBayes;PidAlgoMvd");
pndana->FillList(eminus, "ElectronLooseMinus", "PidAlgoEmcBayes;PidAlgoMvd");
```

Or 'by hand':

```
RhoCandList charged, kaonLoose;

PndAnaPidSelector kaonSel("KaonSelector");
kaonSel.SetSelection("KaonLoose");           // set selection criterion

PndAnaPidCombiner pidComb("PidCombiner");
pidComb.SetTcaNames("PidAlgoDrc;PidAlgoMvd"); // set algo's

while (evr->GetEvent())
{
    pndana->FillList(charged, "Charged");      // start w/ charged candidates

    pidComb.Apply(charged);                     // copy probab. to candidates
    kaonLoose.Select(charged, kaonSel);          // select kaons from charged
}
```