Comprehensive PID Processor

Beta version now online

Uli Einhaus
ILD SW & Ana Meeting
05.07.2023

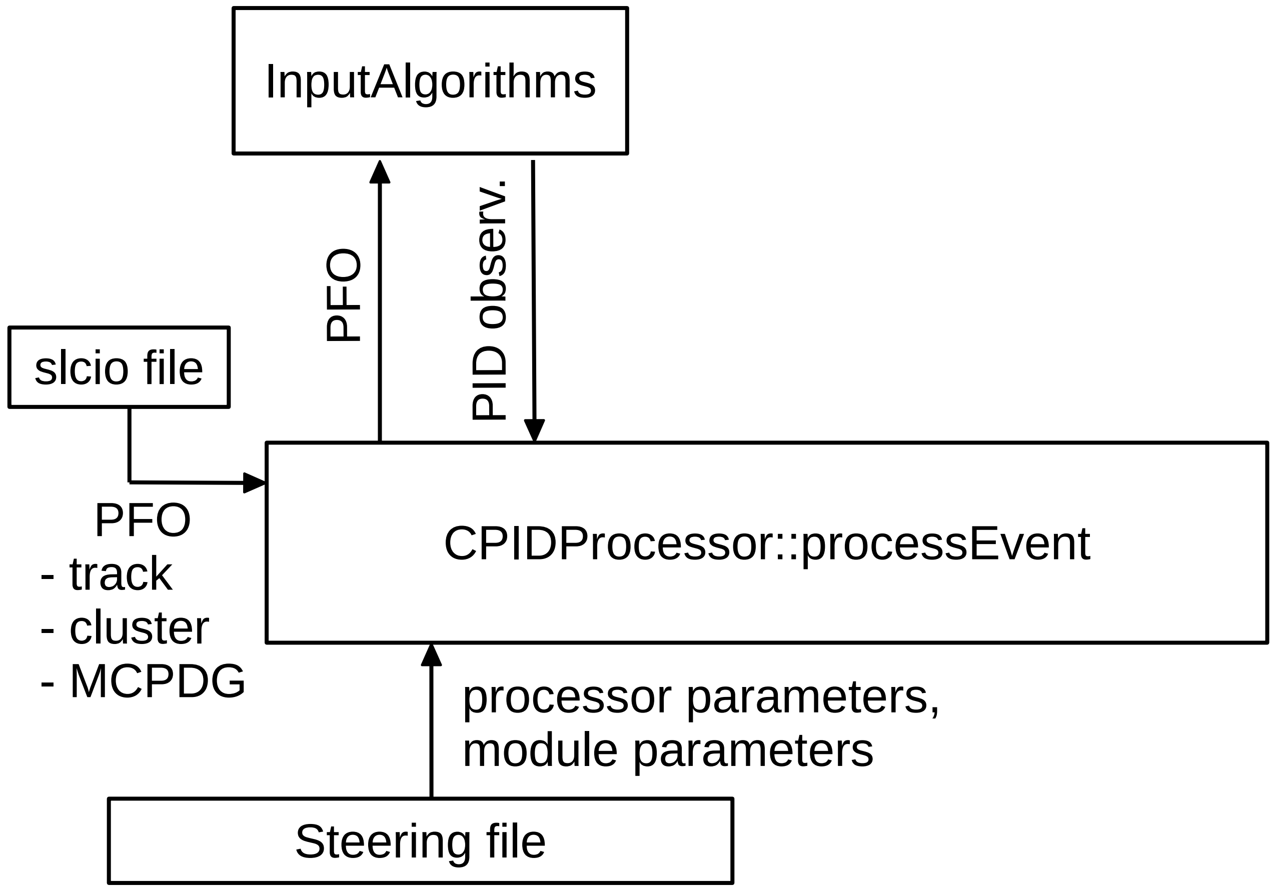
Overview

- Comprehensive Particle Identification (CPID) Processor
- Target: provide platform for future collider detectors to evaluate PID
- Approach: central book-keeping, modules for PID observables as well as training & inference
- Use Particle Flow Objects (PFOs),
- Currently Marlin processor using LCIO, usable in Gaudi via MarlinWrapper, goal is to have native implementation
- CPID (beta version) is part of the current iLCSoft release

- Today: structure, how to use, module overview, PID performance

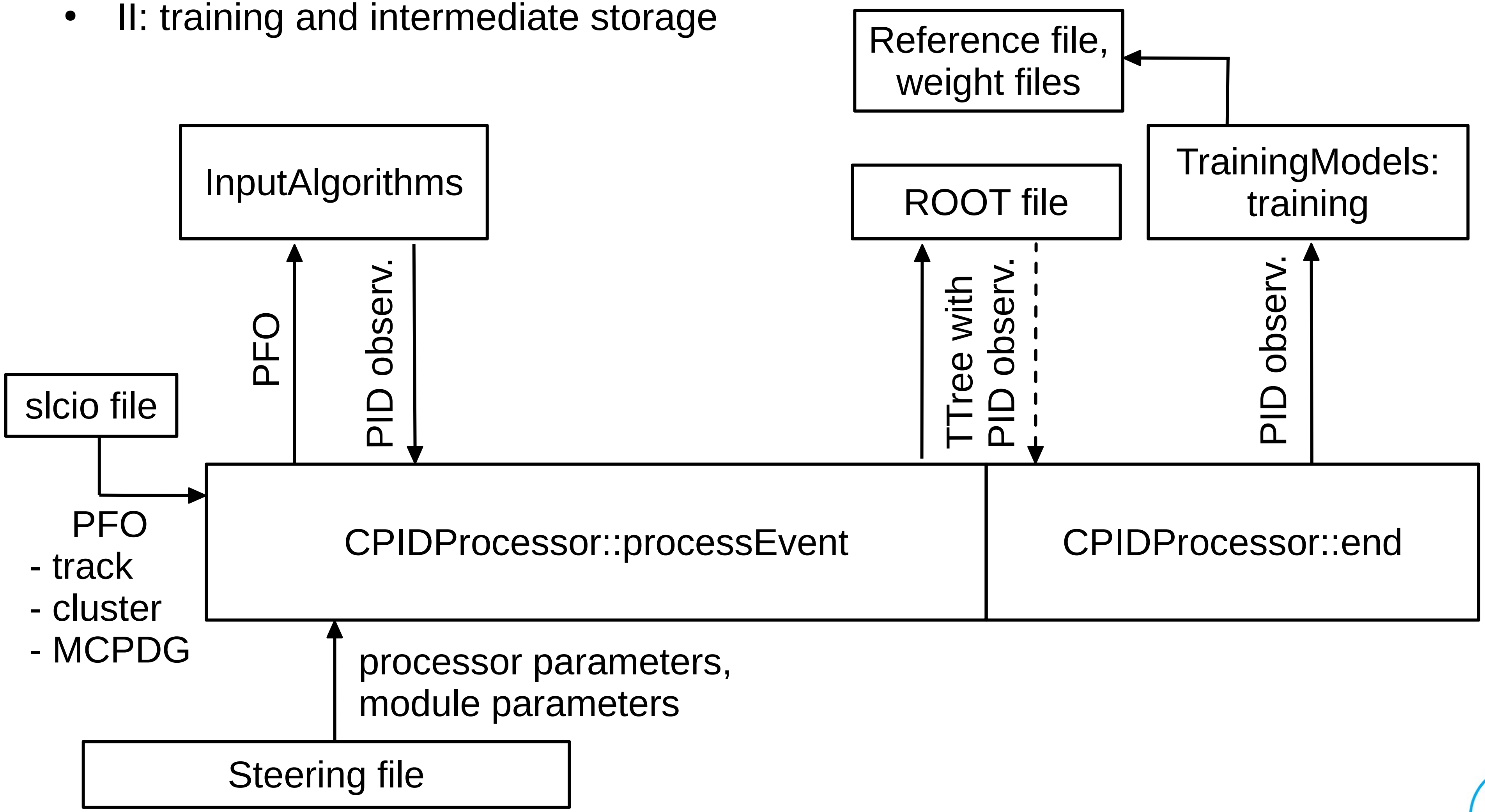
Structure of the CPID workflow

- I: set up and observable extraction



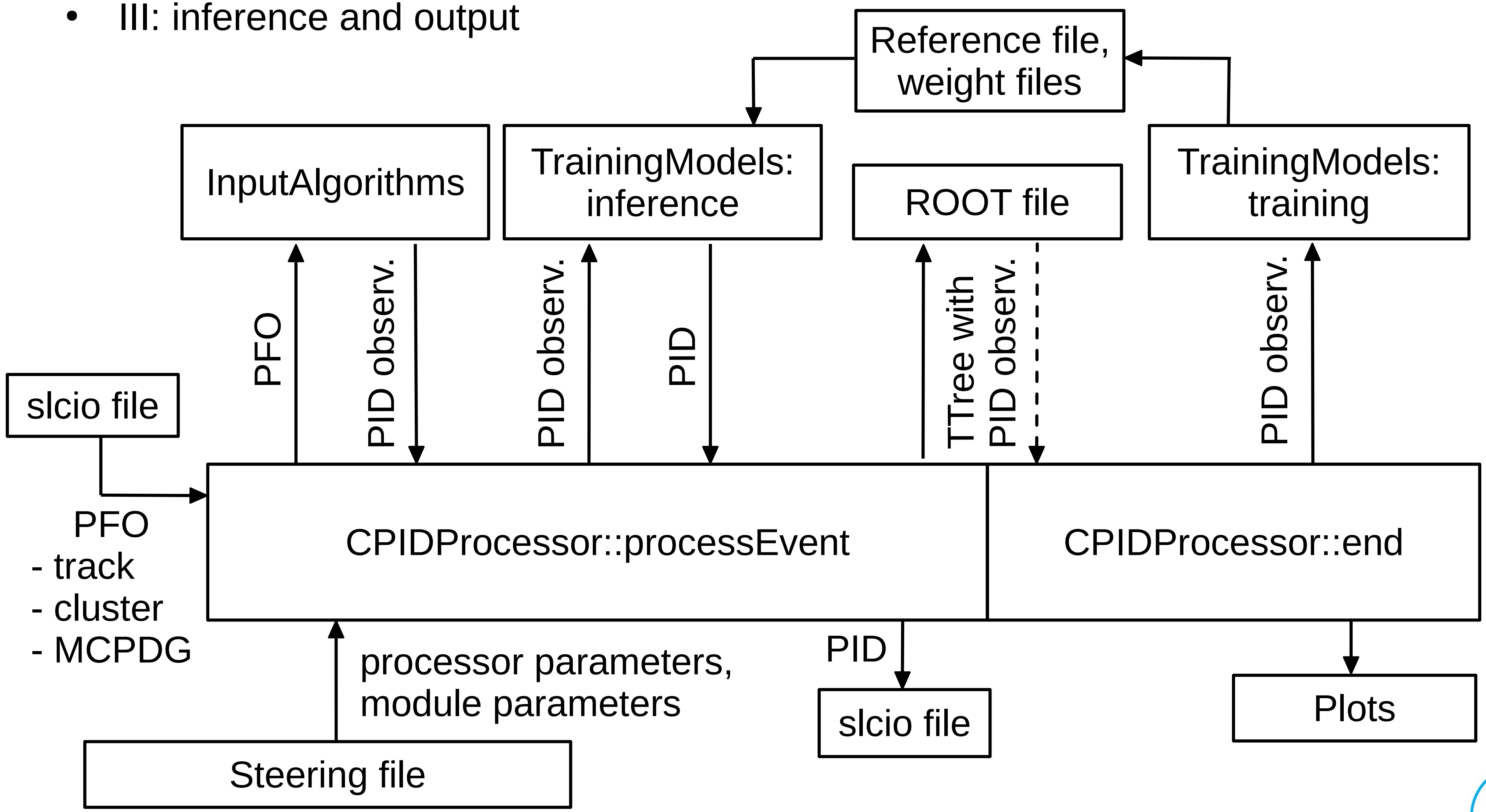
Structure of the CPID workflow

- II: training and intermediate storage



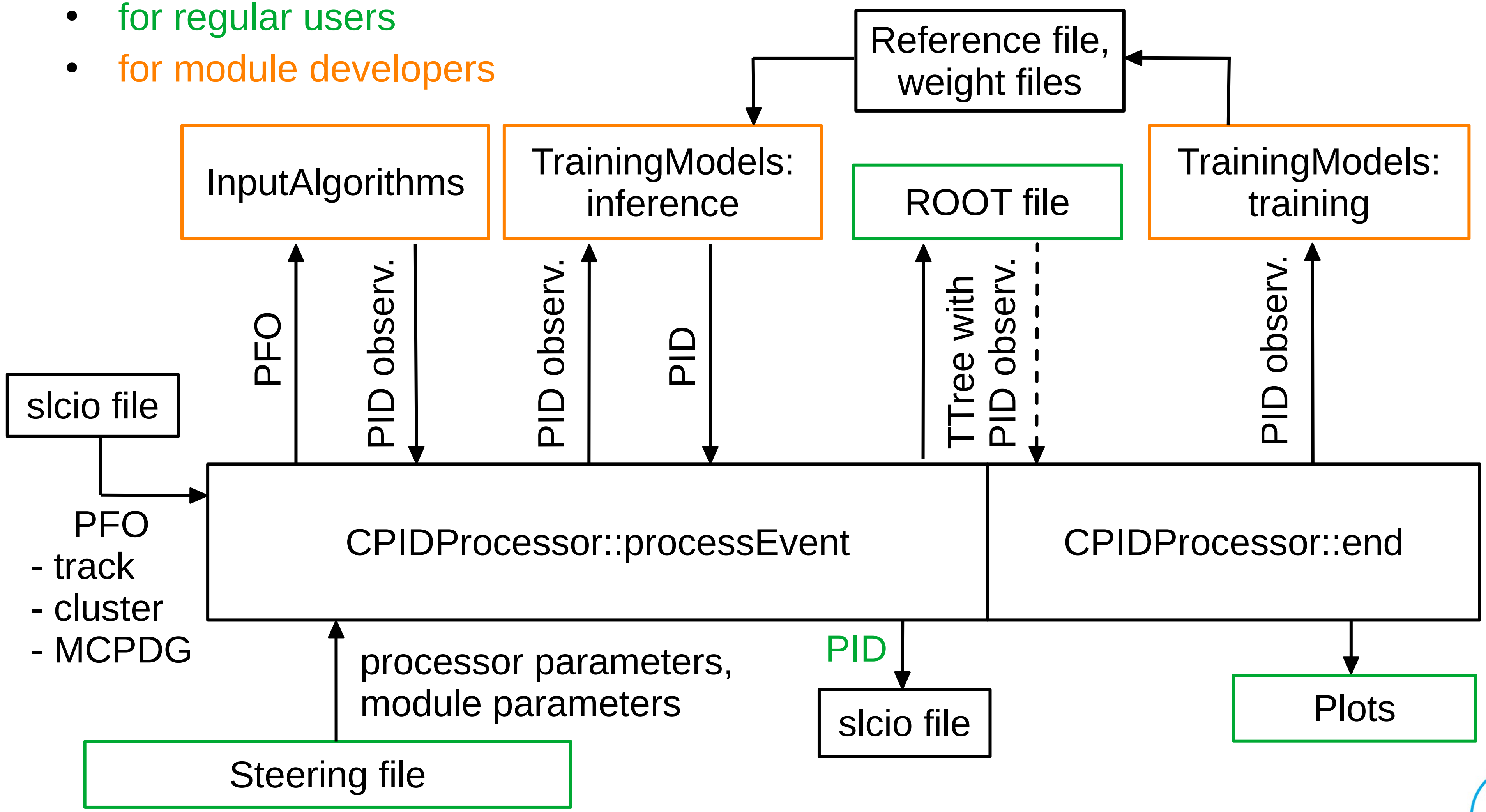
Structure of the CPID workflow

- III: inference and output



Structure of the CPID workflow

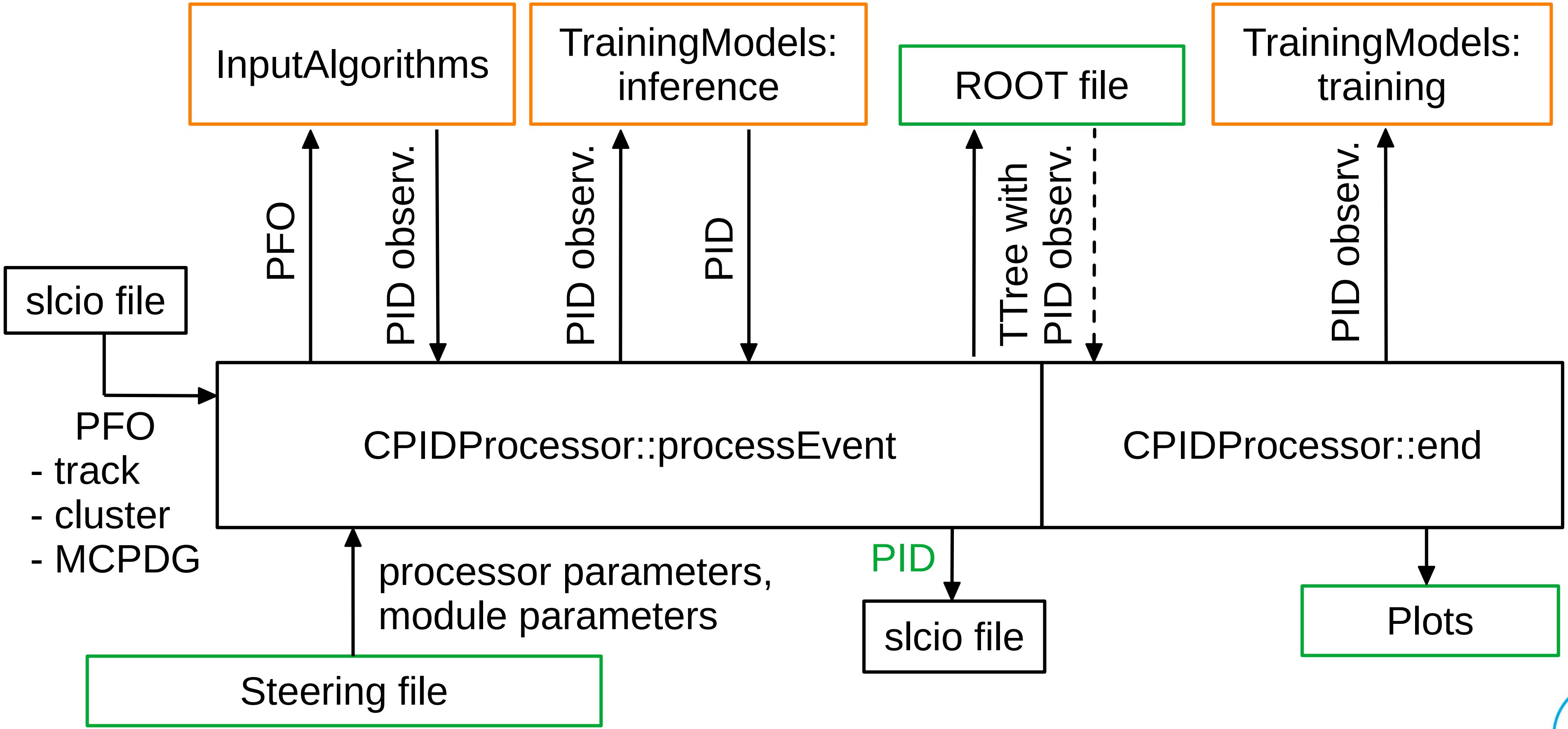
- for regular users
- for module developers



Structure of the CPID workflow

- for regular users
- for module developers

Dynamic loading of modules means module developers don't need to touch the actual processor (analogous to Marlin processors and actual Marlin)



How To Use

- Example steering file

```xml
<processor name="MyComprehensivePIDProcessor" type="ComprehensivePIDProcessor">

  <parameter name="PFOCollection" type="string" value="PandoraPFOs"/>
  <parameter name="RecoMCTruthLink" type="string" value="RecoMCTruthLink"/>
```

collections

```xml
  <parameter name="modeExtract" type="bool" value="true" />
  <parameter name="modeTrain"   type="bool" value="true"/>
  <parameter name="modeInfer"   type="bool" value="false"/>
```

mode selection

```xml
  <parameter name="TTreeFileName" type="string" value="TTreeFile.root"/>
  <parameter name="reffile" type="string" value="Ref.12bins.txt"/>
  <parameter name="signalPDGs" type="FloatVec"  value="11 13 211 321 2212"/>
  <parameter name="backgroundPDGs" type="FloatVec"  value=""/>
  <parameter name="plotFolder" type="string" value="."/>
  <parameter name="fileFormat" type="string" value=".png"/>
```

miscellaneous

```xml
  <parameter name="momMin" type="float" value="1"/>
  <parameter name="momMax" type="float" value="100"/>
  <parameter name="momLog" type="bool" value="true"/>
  <parameter name="momNBins" type="float" value="12"/>
```

momentum bins → separate model run per bin

```xml
  <parameter name="cutD0" type="float" value="0"/>
  <parameter name="cutZ0" type="float" value="0"/>
  <parameter name="cutLamMin" type="float" value="0"/>
  <parameter name="cutLamMax" type="float" value="0"/>
  <parameter name="cutNTracksMin" type="int" value="1"/>
  <parameter name="cutNTracksMax" type="int" value="-1"/>
```

PFO cuts, not used for training

How To Use

- Example steering file

```
<parameter name="inputAlgoSpecs" type="StringVec">
  dEdx_RCD:dEdx_RCD
  TOF:TOF50
  Pandora:Pandora
</parameter>

<parameter name= "TOF0.S" type="StringVec" value="TOFEstimators0ps" />
<parameter name="TOF10.S" type="StringVec" value="TOFEstimators10ps"/>
<parameter name="TOF50.S" type="StringVec" value="TOFEstimators50ps"/>

<parameter name="dEdx_RCD.F" type="FloatVec">
  -1.28883368e-02   2.72959919e+01   1.10560871e+01 -1.74534200e+00  -9.84887586e-07
   6.49143971e-02   1.55775592e+03   9.31848047e+08  2.32201725e-01   2.50492066e-04
   6.54955215e-02   8.26239081e+04   1.92933904e+07  2.52743206e-01   2.26657525e-04
   7.52235689e-02   1.59710415e+04   1.79625604e+06  3.15315795e-01   2.30414997e-04
   7.92251260e-02   6.38129720e+04   3.82995071e+04  2.80793601e-01   7.14371743e-04
   1
</parameter>

<parameter name="trainModelSpecs" type="StringVec">
  TMVA_BDT_MC:TMVA_BDT_MC_12bins
</parameter>
<parameter name="trainingObservables" type="StringVec"> </parameter>

<parameter name="TMVA_BDT_MC_12bins.S" type="StringVec">
  !V:!Silent:Color:DrawProgressBar:Transformations=I;D;P;G,D:AnalysisType=multiclass
  SplitMode=Random:NormMode=NumEvents:!V
  !H:!V:NTrees=100:BoostType=Grad:Shrinkage=0.10:UseBaggedBoost:BaggedSampleFraction=0.50
  dEdx_RCD_piDis>-900&&dEdx_RCD_kaDis>-900
</parameter>
```

specify InputAlgorithms
[type]:[name]

give module parameters
[name].S, [name].F
-
these depend on the
individual modules

specify TrainingModel(s)
[type]:[name]

give module parameters
[name].S, [name].F

TrainingModels: TMVA_BDT & TMVA_BDT_MC

- Direct connection with observables TTree, run ‘standard’ ROOT TMVA BDT
- Take 4 string inputs, corresponding to TMVA loader and factory options

- TMVA_BDT uses simple sig/bkg BDT, trains _signalPDGs vs. _backgroundPDGs
- TMVA_BDT_MC uses multiclass BDT, trains all _signalPDGs against each other
  → used for performance plots on following slides

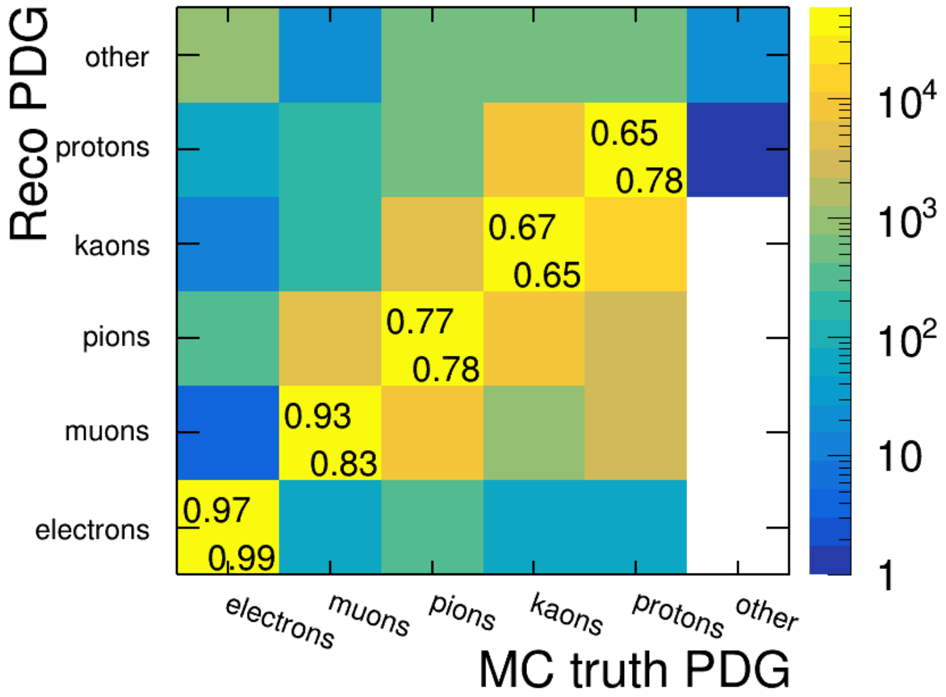
InputAlgorithm: dE/dx_RCD

- RCD = reference curve distance, i.e. distance to Bethe-Bloch curves, removes momentum dependence compared to using dE/dx value directly
- Takes (fully reconstructed) dE/dx value from Compute_dEdxProcessor, and reference curves from dEdxAnalyser as float parameter input
- Optional: adjustment of dE/dx value by scaling of distance to true curve, emulates better/worse dE/dx resolution
- Gives 5 observables: el_dis, mu_dis, pi_dis, ka_dis, pr_dis

InputAlgorithms: Pandora

- Takes the PID assigned by PandoraPFA (based on cluster shapes) to the PFO
- Either e, μ, π or photon, returns PDG (11, 13, 211 or 22)

- Plot: combination of dE/dx_RCD and Pandora

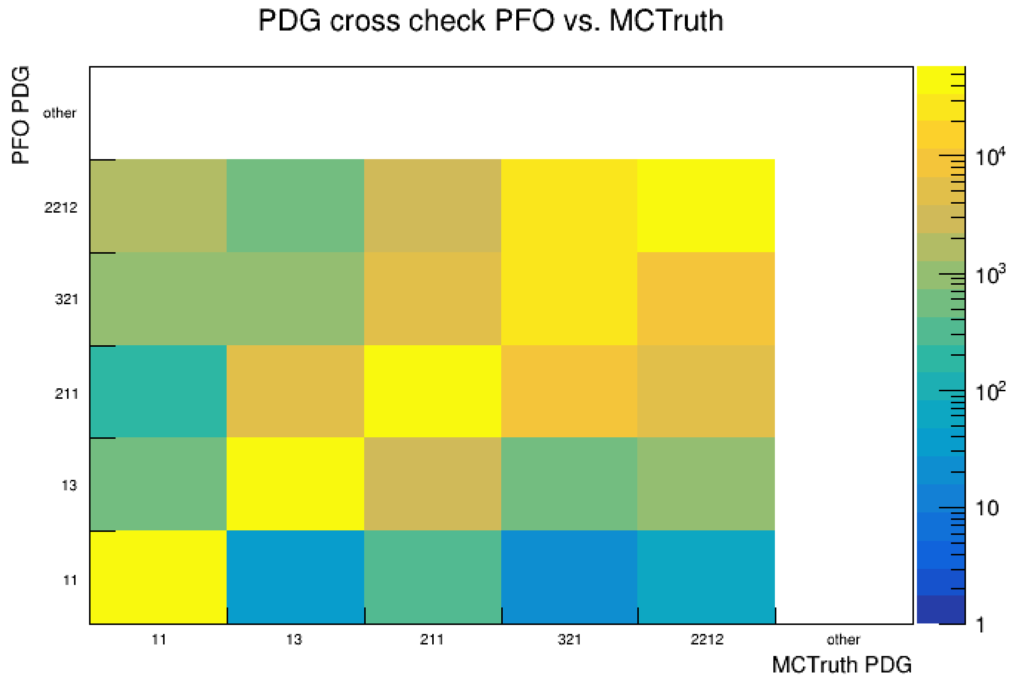


InputAlgorithms: TOF and TOF223

- TOF:
  - time of flight, using initial implementation
  - various issues leasding to limited purity
  - available in last large MC production
  - returns track beta (v/c)
- TOF223:
  - time of flight (in iLCSoft v02-02-03), using B. Dudar’s new TOF implementation including track length estimation
  - only available in newest simulation
  - runs on REC files, i.e. can’t be done retroactively
  - returns reconstructed PFO mass
- Both use output of TOFEstimator, only need corresponding estimator name to find in PFO PIDHandler

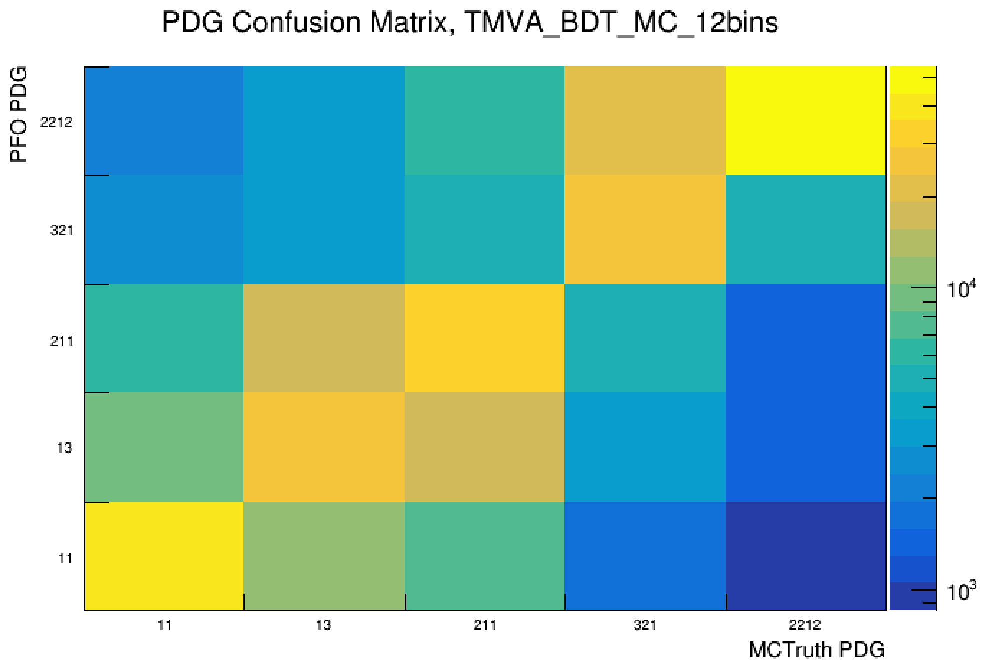
InputAlgorithm: LeptonID

- L. Reichenbach’s new LeptonID (e vs. µ vs. hadrons, target: semileptonic b/c decays)
- Use LeptonID’s BDT scores (based on ~20 inputs) as input for CPID



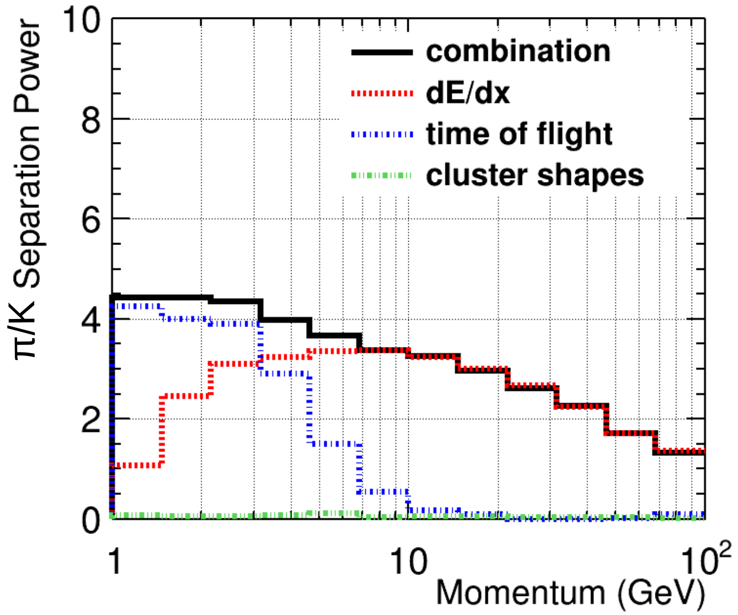
InputAlgorithm: dN/dx

- Request by IDEA DC people
- Using parametrisation of cluster counting dN/dx(βγ) from Delphes
- Take first track in PFO, calculate track length in TPC or parametrised cylinder, get Poisson(average #clusters), get distance to reference curves



π/K Separation with Combined Observables

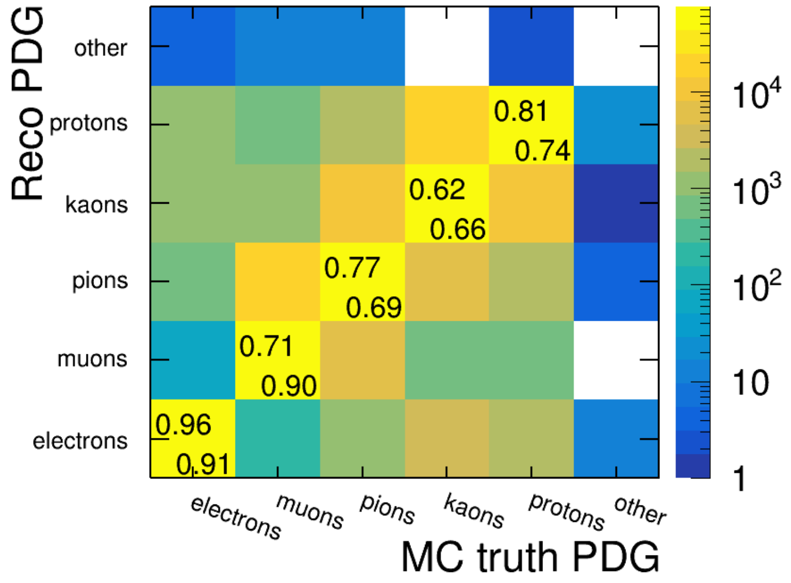
- dE/dx, TOF10 and Pandora
- TMVA_BDT with sig = K, bkg = π
  train & eval per 12 mom bins and per used observable(s)
- TOF works at low momenta, dE/dx at moderate ones, Pandora not at all (as expected)
- Combination corresponds to independent observables



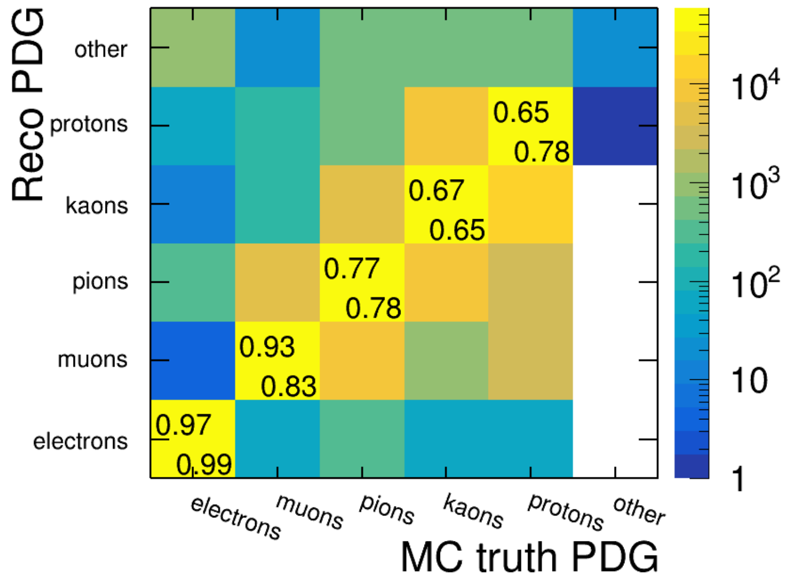
Performance Comparison

- Here: multiclass BDT; confusion matrix with $^{\text{eff}}/_{\text{pur}}$ on diagonal
- Simple BDT already generates similar performance to current LikelihoodPID
- Using dE/dx and Pandora (based on cluster shapes)

Current LikelihoodPID



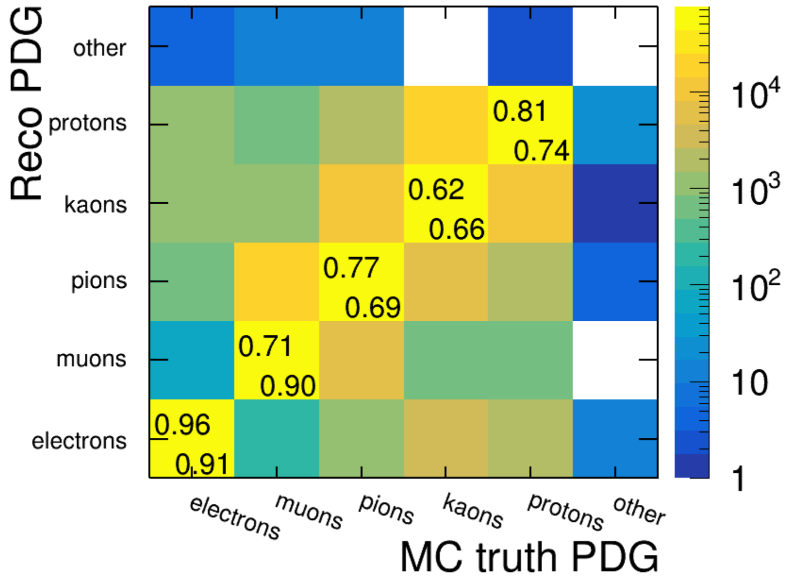
CPID framework



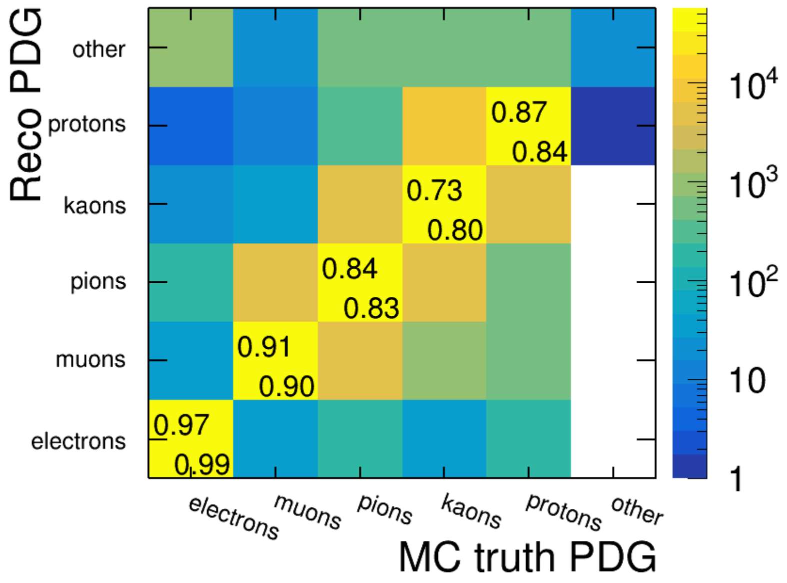
Performance Comparison

- Here: multiclass BDT; confusion matrix with $^{\text{eff}}/\text{pur}$ on diagonal
- Simple BDT already generates similar performance to current LikelihoodPID
- Addition of TOF gives immediately better result – previously hard, easy in CPID

Current LikelihoodPID



CPID framework; + TOF



Conclusion and Outlook

- First CPID version online – beta version, no warranty!
- Please test and send feedback & feature requests!

- More features
  - (abstract) RICH algorithm?
  - neural network training, possible interface with pyTorch
  - more performance assessment output
  - more documentation…
- Intention: add CPID to standard high level reco and add PID estimator to the PFO, possibly a conservative and an ambitious one
- Make this available in native Key4HEP / Gaudi