Machine-Learning PID for the ePIC pfRICH

Charles Joseph Naim / Dongwi Dongwi / Lucas Rhode

10/2/2025

Motivation & Goals

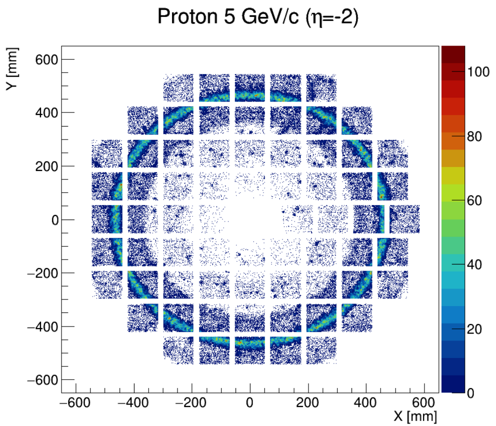
- pfRICH: particle identification (PID) via Cherenkov imaging in the ePIC detectir.
- Goal: train and evaluated ML-based classifier to separate $\pi^{+}/K^{+}/p$ across p = 1-10 GeV and η = -1.0…-3.5.
- Why ML now?
  - Integrate multi-feature patterns (hit positions, angles, timing) beyond traditional 1D radius cuts.
  - Provide a reproducible, modular pipeline for rapid iteration & fair comparisons to baseline methods.
- Deliverables today: dataset, pipeline, metrics, η–p performance maps, and plan

Dataset Summary

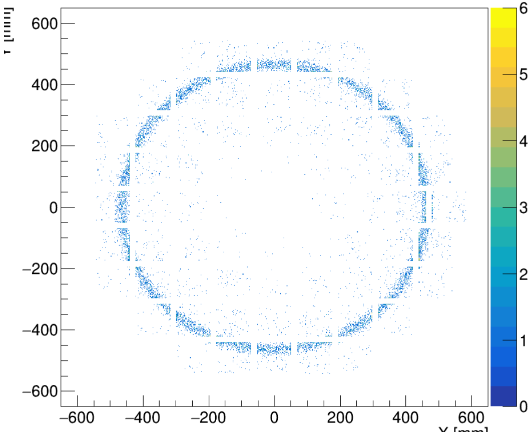
- Simulation inputs: 240 sanitized ROOT files (10k events each) generated from pfRICH simulations.
- Species: e^-, π^+, K^+, p (primary-only filter at merge stage).
- Phase space:
    - Momentum: p = 1-10 GeV (integer steps).
    - Pseudorapidity: $\eta \in \{-1.0, -1.5, -2.0, -2.5, -3.0, -3.5\}$.
    - ϕ randomized

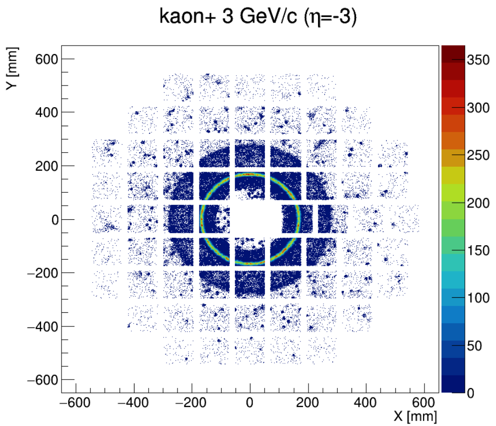
Raw ROOT files

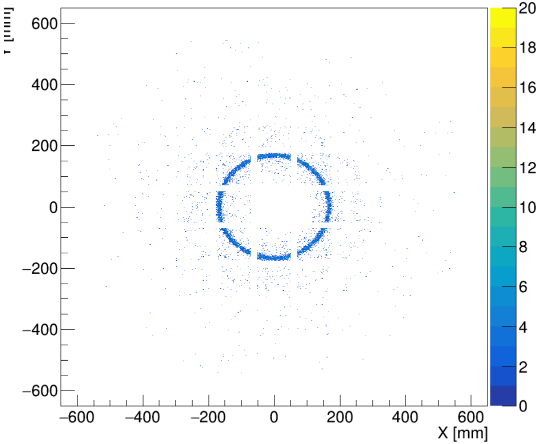
Sanitized ROOT files

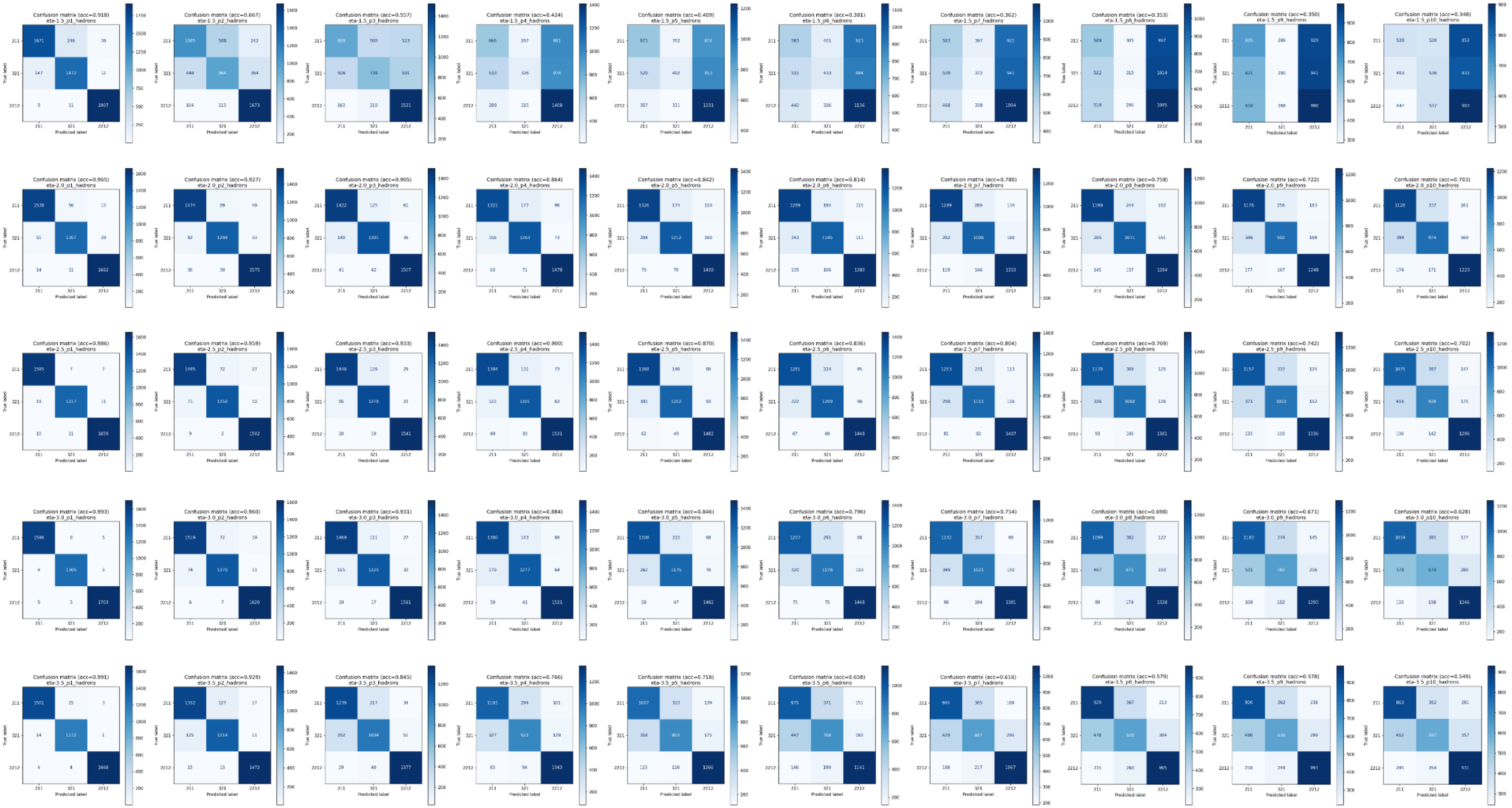


readTree.C

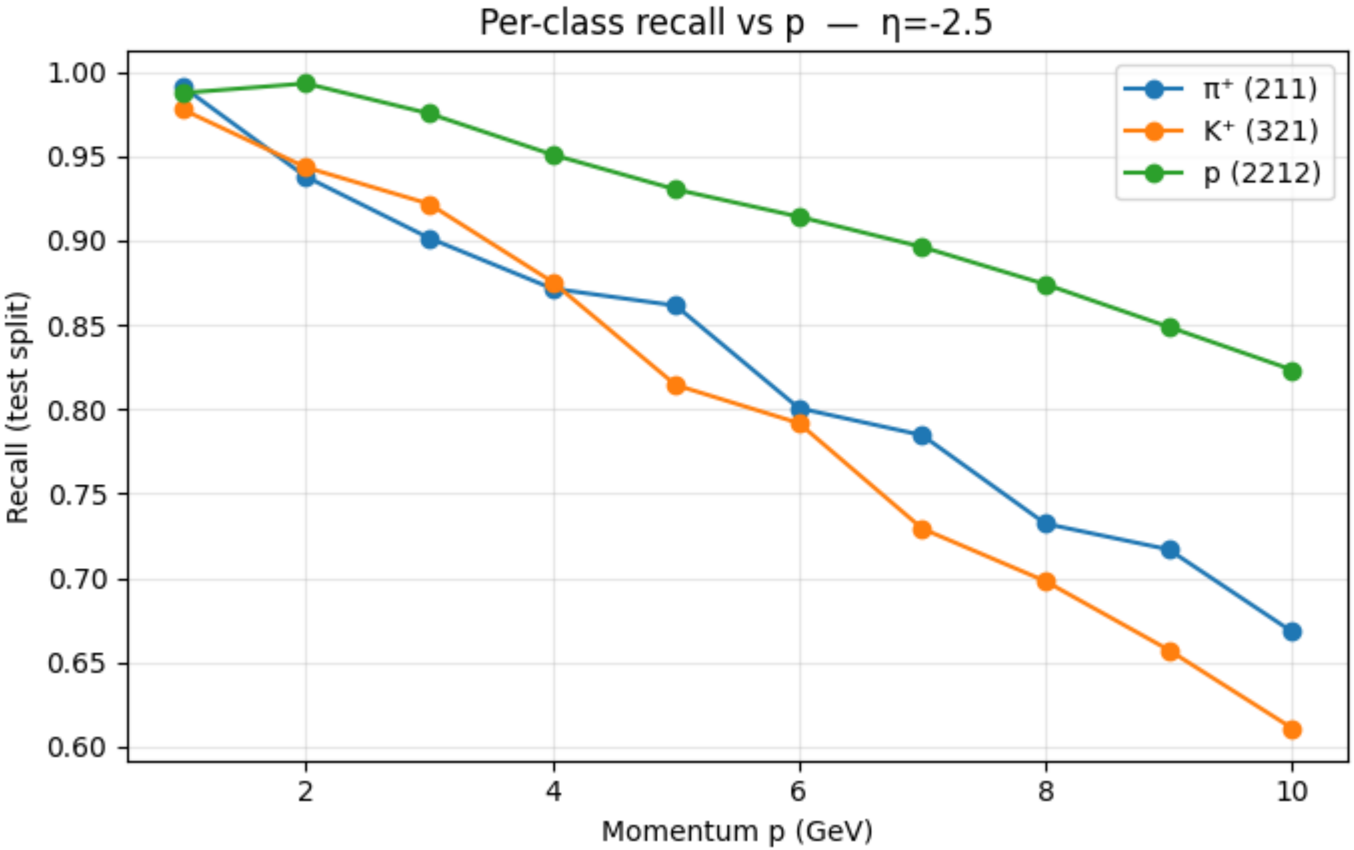








Confusion matrix (acc=0.918) eta-1.5_p1_hadrons
Confusion matrix (acc=0.667) eta-1.5_p2_hadrons
Confusion matrix (acc=0.557) eta-1.5_p3_hadrons
Confusion matrix (acc=0.424) eta-1.5_p4_hadrons
Confusion matrix (acc=0.409) eta-1.5_p5_hadrons
Confusion matrix (acc=0.381) eta-1.5_p6_hadrons
Confusion matrix (acc=0.362) eta-1.5_p7_hadrons
Confusion matrix (acc=0.353) eta-1.5_p8_hadrons
Confusion matrix (acc=0.350) eta-1.5_p9_hadrons
Confusion matrix (acc=0.348) eta-1.5_p10_hadrons
Confusion matrix (acc=0.965) eta-2.0_p1_hadrons
Confusion matrix (acc=0.927) eta-2.0_p2_hadrons
Confusion matrix (acc=0.905) eta-2.0_p3_hadrons
Confusion matrix (acc=0.864) eta-2.0_p4_hadrons
Confusion matrix (acc=0.842) eta-2.0_p5_hadrons
Confusion matrix (acc=0.814) eta-2.0_p6_hadrons
Confusion matrix (acc=0.780) eta-2.0_p7_hadrons
Confusion matrix (acc=0.758) eta-2.0_p8_hadrons
Confusion matrix (acc=0.722) eta-2.0_p9_hadrons
Confusion matrix (acc=0.703) eta-2.0_p10_hadrons
Confusion matrix (acc=0.986) eta-2.5_p1_hadrons
Confusion matrix (acc=0.959) eta-2.5_p2_hadrons
Confusion matrix (acc=0.933) eta-2.5_p3_hadrons
Confusion matrix (acc=0.900) eta-2.5_p4_hadrons
Confusion matrix (acc=0.870) eta-2.5_p5_hadrons
Confusion matrix (acc=0.836) eta-2.5_p6_hadrons
Confusion matrix (acc=0.804) eta-2.5_p7_hadrons
Confusion matrix (acc=0.769) eta-2.5_p8_hadrons
Confusion matrix (acc=0.742) eta-2.5_p9_hadrons
Confusion matrix (acc=0.702) eta-2.5_p10_hadrons
Confusion matrix (acc=0.993) eta-3.0_p1_hadrons
Confusion matrix (acc=0.960) eta-3.0_p2_hadrons
Confusion matrix (acc=0.931) eta-3.0_p3_hadrons
Confusion matrix (acc=0.884) eta-3.0_p4_hadrons
Confusion matrix (acc=0.846) eta-3.0_p5_hadrons
Confusion matrix (acc=0.796) eta-3.0_p6_hadrons
Confusion matrix (acc=0.754) eta-3.0_p7_hadrons
Confusion matrix (acc=0.698) eta-3.0_p8_hadrons
Confusion matrix (acc=0.671) eta-3.0_p9_hadrons
Confusion matrix (acc=0.628) eta-3.0_p10_hadrons
Confusion matrix (acc=0.991) eta-3.5_p1_hadrons
Confusion matrix (acc=0.929) eta-3.5_p2_hadrons
Confusion matrix (acc=0.845) eta-3.5_p3_hadrons
Confusion matrix (acc=0.766) eta-3.5_p4_hadrons
Confusion matrix (acc=0.718) eta-3.5_p5_hadrons
Confusion matrix (acc=0.658) eta-3.5_p6_hadrons
Confusion matrix (acc=0.616) eta-3.5_p7_hadrons
Confusion matrix (acc=0.579) eta-3.5_p8_hadrons
Confusion matrix (acc=0.578) eta-3.5_p9_hadrons
Confusion matrix (acc=0.549) eta-3.5_p10_hadrons


Per-class recall vs p — η=-2.5
Recall (test split)
Momentum p (GeV)
π⁺ (211)
K⁺ (321)
p (2212)
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
2
4
6
8
10

Features & Labels

- Base features (6): posX, posY, theta, phi, time, pT
- Engineered (2): radius = $\sqrt{(posX^2+posY^2)}$, bin_index (stable 2D bin over sensor plane).
- Target: pdgID $\in \{211, 321, 2212\}$
- Rationale:
  - posX/posY/radius capture Cherenkov ring geometry
  - theta/phi and pT provide kinematic context
  - time can encode photon path differences / detector timing

Model & Training Protocol

- Model: Gradient-boosted decision trees (XGBoost) via GradientBoostHybrid interface.
- Per-η, per-p models: train on each (η, p) slice; all-p@fixed η model for robustness
- Split: 80/20 train/test (falls back to non-stratified split if a class is underpopulated).
- Artifacts saved per run: booster JSON, label encoder, feature list, confusion matrix PNG, raw CM NPY, per-class CSV, merged ROOT used for training

What’s Next:

1. Hyper-parameter tuning
2. Physics-aware features: ring-fit parameters, photon yield, expected θ_C from refractive index model
3. Baseline parity: reproduce CDR-style $N\sigma$ separations, then show ML delta over the baseline