Particle Identification at LHCb

Miriam Lucio, on behalf of the LHCb Collaboration

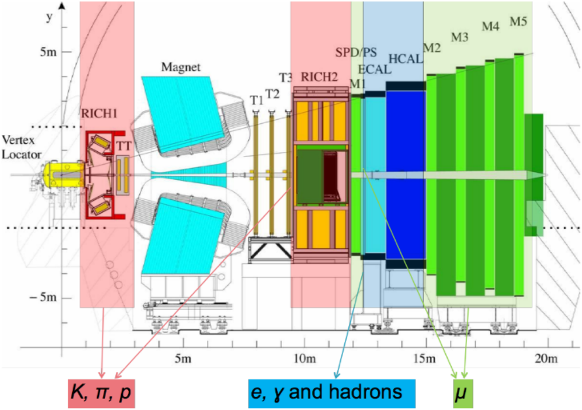
IML Workshop

April 10, 2018

Outline

LHCb [Int J Mod Phys A30 (2015) 1530022]

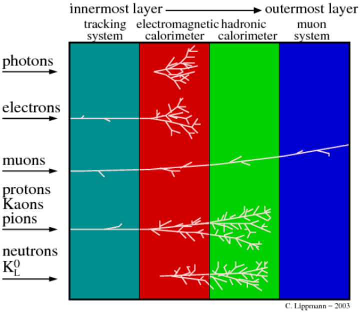
- Single-arm forward spectrometer, $2 < \eta < 5$
- **General purpose experiment**, initially designed to study of particles containing b or c quarks



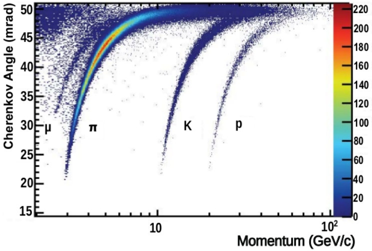
- **Detected particles:** e, μ, π, K, p, γ, ghosts (tracks that do not correspond to any real particle)

PID at LHCb

- **Problem:** identify particle type associated with a track/energy deposited in the subdetectors
  - Charged: π, e, μ, K, p
  - Neutral: π^0, γ, n
- Better PID performance $\rightarrow$ better bkg rejection $\rightarrow$ more precise results
- PID also used for **trigger** (in particular for upgrade): less background $\rightarrow$ less resources (less bandwidth)
- High-level info from subdetectors + track quality info $\rightarrow$ multi-class classification in **machine learning**



[Int J Mod Phys A30 (2015) 1530022]



Outline

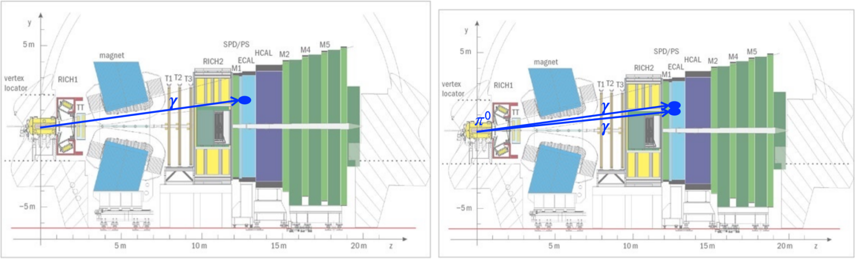
Introduction

- **Radiative decays**: interesting area of study at LHCb (e.g. photon polarisation measurement [PRL 112, 161801 (2014)])

Problem

- π^0 copiously produced at LHCb , inmediate decay to $\gamma\gamma$
- high momentum $\pi^0 \rightarrow$ merge of ECAL clusters $\rightarrow$ huge background for radiative decays

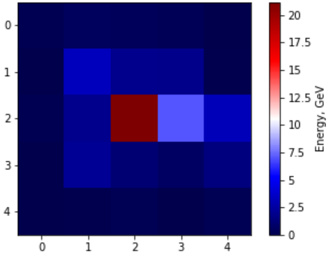
Need for a powerful tool to discriminate signal (γ) from background ($\pi^0 \rightarrow \gamma\gamma$)



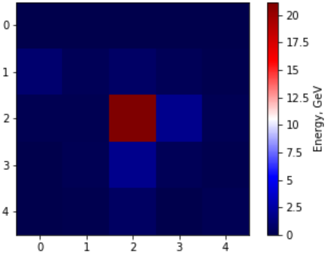
ECAL Signatures

Different signatures (MC events):

$\pi^0 \rightarrow \gamma\gamma$



γ

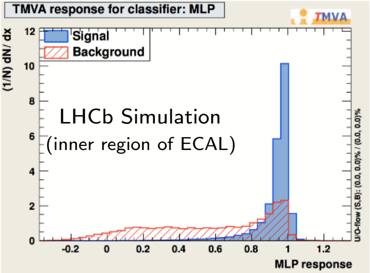


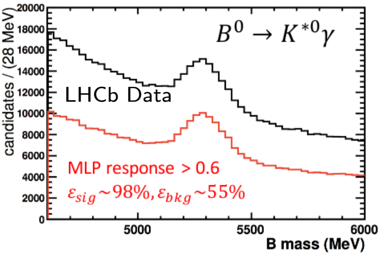
ECAL clusters (3x3 cells)

Coarse granularity $\rightarrow$ separation is not straightforward

Baseline approach [LHCb-PUB-2015-016]

- Neural Network with 2 hidden layers (TMVA MLP)
  - Train with $B^0 \to K^{*0}\gamma$ as signal and $B^0 \to \pi^0 X$ as background
- 14 ECAL and Pre-Shower cluster parameters (grouped under **shape** and **symmetry**)
  - 4 variables that account for the size & tails, semiaxes and orientation of the ellipse in the ECAL
  - 2 variables related to the energy of the most (*seed*) and the second most energetic cells of the cluster
  - 4 variables for multiplicities of hits in the PS cells matrix in front of the seed of the electromagnetic cluster
  - 4 shape and asymmetry variables in the 3x3 PS cells





New approach

- **2x25 input features:** responses in 5x5 cells cluster from ECAL and pre-shower detectors
- **Train:** $B \to K\pi\gamma$ and $B \to K\pi\pi^0$ MC samples (kinematically similar)
- **Test:** $B \to K\pi\gamma$ and $B \to K\pi\pi^0$, $B \to J/\psi K^*\pi^0$ MC samples
- Potentially misleading π^0 candidates with 2 outgoing γ sharing the same cluster are studied
- 8 TeV MC data, 120k photons, 220k π^0
- **Neural Networks** and **BDT** considered

New approach

Neural Network

- 1 or **2** hidden layers
- **Width:** 100, **250**, 500, 800 units for ECAL + **10**, 50 Preshower
- **Optimizer:** **Adamax**, Adagrad, SGD

ROC AUC = 0.89

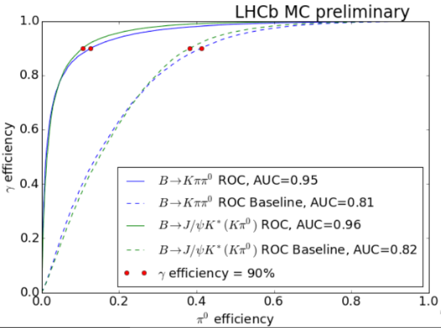
BDT

- **XGBoost**, CatBoost, LightGBM [a]

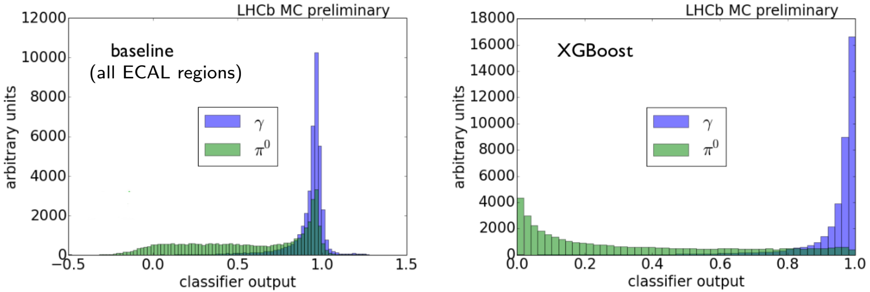
ROC AUC = 0.95

[a] https://github.com/yandexdataschool/modelgym/

Big improvement, specially for moderate efficiencies



Comparison



- Aggressive bacgkround suppression
- Good prospects for π^0 suppression & photon suppression for π^0 recovery

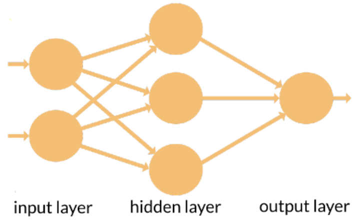
Outline

Introduction

- **Goal:** improve PID variables for charged particles → better background rejection
- **Particles:** electron, muon, pion, kaon, proton, ghosts
- Only tracks using information from the full detector considered

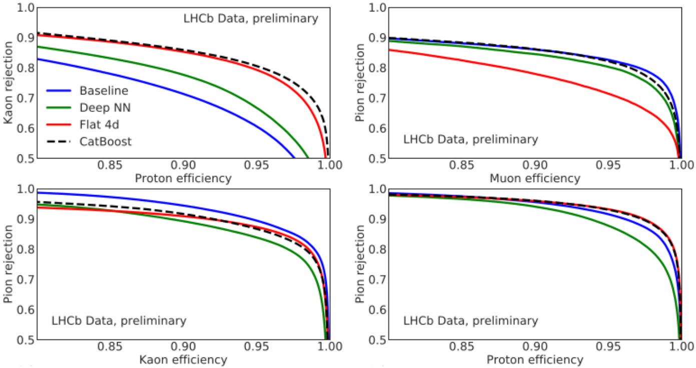
Baseline solution (ProbNN)

- Standard MVA used for PID LHCb
- Artificial neural networks with 6 binary classification models (One-versus-rest approach: separate one type from the others)
- 1 hidden layer, TMVA MLP [arXiv:0703039]
- **Activation function:** tanh, sigmoid
- **Training method:** Back-Propagation (BP), BFGS Algorithm (BFGS), or Genetic Algorithm (GA - slower and worse)
- **Estimator:** MSE (Mean Square Estimator) for Gaussian Likelihood or CE(Cross-Entropy) for Bernoulli Likelihood
- Trained on 2015 MC



Non-flat efficiency approach

- Gradient boosting:
  - XGBoost [arXiv:1603.02754]
  - Decision train (DT) [arogozhnikov.github.io/2015/05/22/decision-train-classifier.html]
  - CatBoost [arXiv:1706.09516]
- Artificial neural networks (NN)
  - One hidden layer
  - Deep neural networks
  - Linear combinations of features from subdetectors



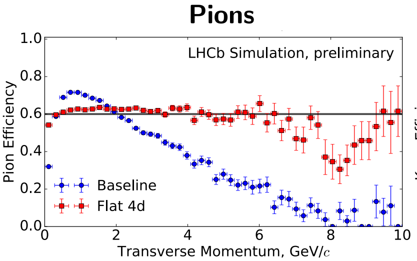
- **CatBoost** and **DNN** give best results

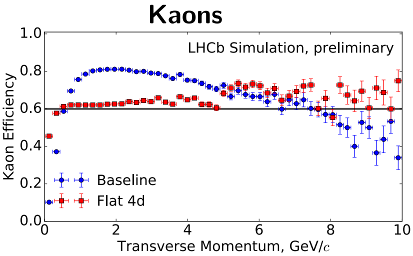
Flat efficiency approach

- PID performace depends on **particle kinematics** (p,p_T,η) and **N$_{\mathbf{tracks}}$**
- Flat PID efficiencies:
  - ★ Good discrimination for different analyses
  - ★ Unbiased background discrimination
  - ★ Reduced systematic uncertainties

Introduce flatness term in loss function: $\mathcal{L} = \mathcal{L}_{AdaLoss} + \alpha\mathcal{L}_{Flat}$

- **Flat4d:** $\mathcal{L}_{Flat_{4d}} = \mathcal{L}_{Flat\_P} + \mathcal{L}_{Flat\_PT} + \mathcal{L}_{Flat\_nTracks} + \mathcal{L}_{Flat\_\eta}$





Flat4d, ProbNN

$\rightarrow$ Better PID efficiency flatness in p,p_T,η,N_{tracks} than baseline

Outline

Introduction

- PID information used in both trigger selection and offline data analysis
- Obtain efficiency & systematic effects for the PID requirements applied

`PIDCalib` package [CERN-LHCb-PUB-2016-021]

- Data-driven technique
- Efficiency obtained using per-event weights from simulated calibration sample
  - × PID variables cannot be used to train multivariate classifiers
- `MCResampling`: PID response replaced by the one generated from calibration PDFs
  - × Problematic for systematics computation
  - × Ignores correlation

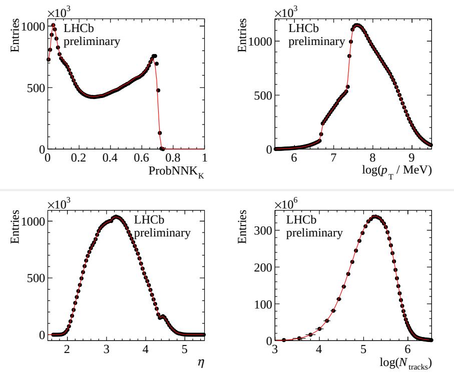
NEW:

- Resampling of PID variables: **`PIDGen`**
- Transformation of PID variables: **`PIDCorr`**

PIDGen & PIDCorr

Input Variables for PDF

- PID variable, $\log p_T$, η, $\log N_{tracks}$
  - Transformed to remove narrow peaks

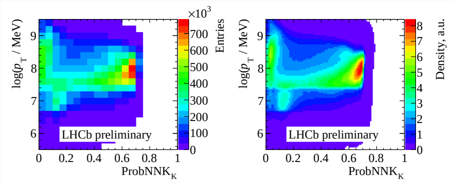


sweighted $D^{*\pm} \to D^0\pi^\pm$ calibration sample

PDF computation

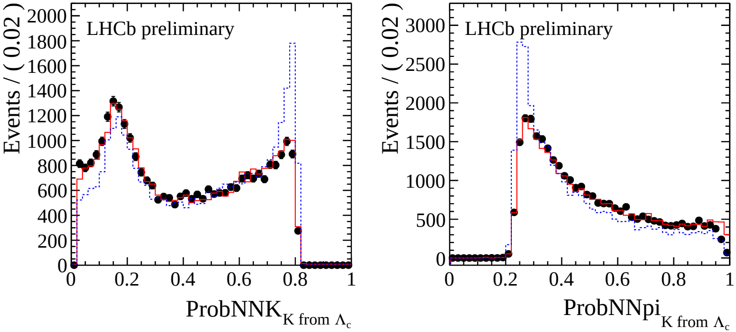
- Four-dimensional kernel density estimation
- `Meerkat` library [2015 JINST 10 P02011]

binned data | estimation



PIDGen validation

- For a given set of ($\log p_T$, η, $\log N_{tracks}$), *generate* PID variable that looks like data using the known 4D distribution of the calibration sample in the PID variable, $\log p_T$, η and $\log N_{tracks}$
- Clean, high-statistics data sample: $\Lambda_b^0 \to \Lambda_c^+ \pi^-$, $\Lambda_c^+ \to p K^- \pi^+$

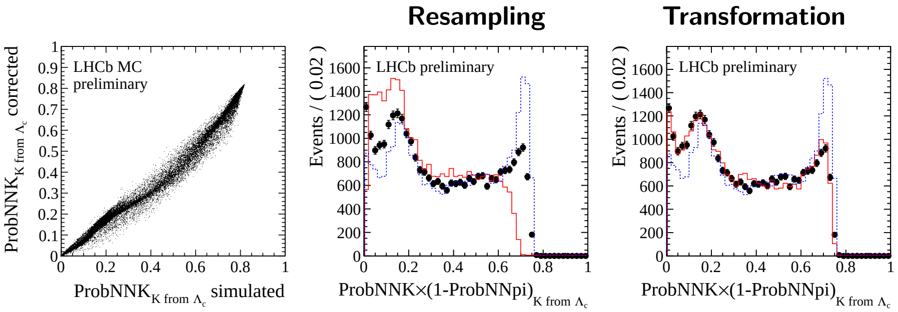


sweighted data, uncorrected simulation, PIDGen-corrected

- Good agreement between corrected MC and data

PIDCorr validation

- Using the obtained 4D PDF for data and MC, construct a function that transforms simulated PID response such that it matches data
- Preserves correlations between different PID responses for the same track



sweighted data, uncorrected simulation, PIDGen-corrected (center), PIDCorr-corrected (right)

Combinations of PID variables:

- Resampling procedure fails (correlations are ignored)
- Transformation of variables: better agreement

Outline

Conclusions & Outlook

- Big improvement in π^0 - γ separation
- Implemented PID transformation tools inside `PIDCalib` that preserves correlations $\rightarrow$ betteer agreement with data
- Baseline ProbNN extended with deep neural networks and gradient boosting
- PID algorithms with better PID efficiency flatness studied

Stay tuned!

Thanks for your attention!