Particle identification at LHCb

New calibration techniques and machine learning classification algorithms

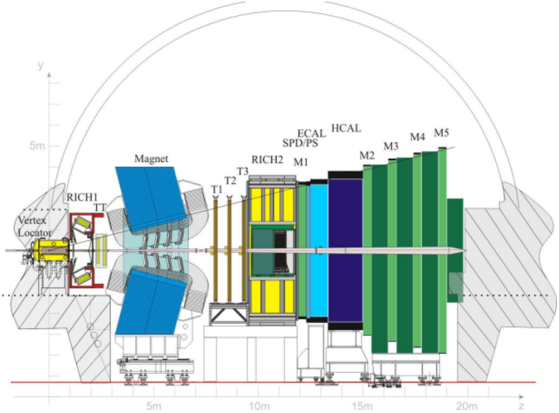
Anton Poluektov

University of Warwick, UK

5 July 2018

On behalf of LHCb collaboration

One-arm spectrometer optimised for studies of beauty and charm decays at LHC

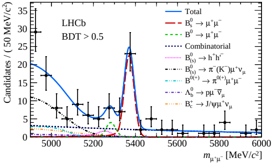


- Good vertexing: measure B^0 and B_s^0 oscillations, reject prompt background
- Particle identification: flavour tagging, misID background
- High-resolution tracking
- Calorimetry: reconstruct neutrals (π^0, γ) in the final state
- Efficient trigger, including fully hadronic modes

Particle ID in LHCb

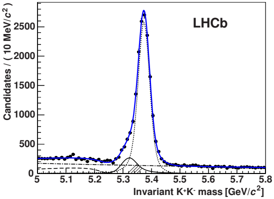
Excellent Particle identification performance is vital for LHCb physics

$B_s^0 \to \mu^+\mu^-$



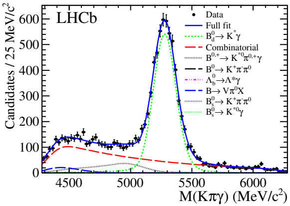
[PRL 118, 191801 (2017)]

$B_s^0 \to K^+K^-$



[JHEP 10 (2013) 183]

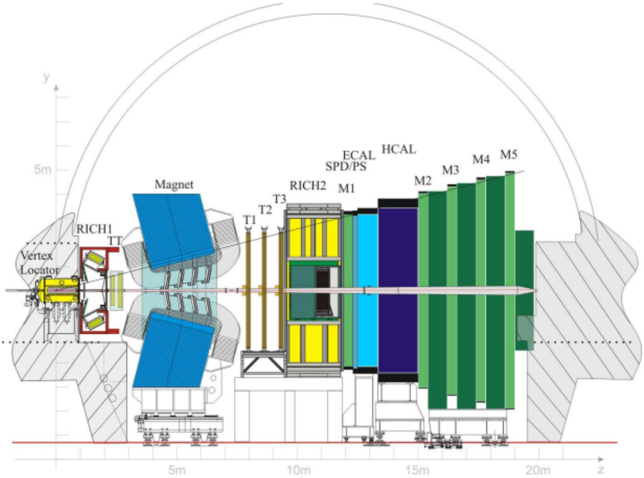
$B \to K^*\gamma$



[Nucl. Phys. B867 (2013), 1]

- Background rejection for rare decays
- Classification of final states with the same topology
- Reduction of bandwidth in the trigger

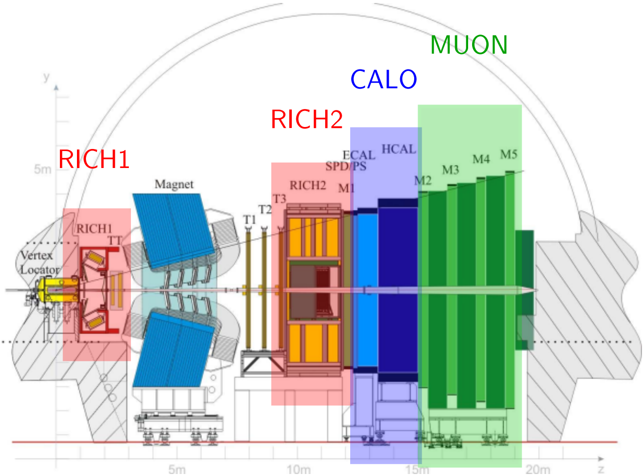
PID subsystems in LHCb



Identify long-lived final state particles based on information from subdetectors:

- Charged: π, K, p, e, μ
- Neutral: π^0, γ

PID subsystems in LHCb



Identify long-lived final state particles based on information from subdetectors:

- Charged: π, K, p, e, μ
- Neutral: π^0, γ

PID and machine learning

Areas for machine learning in PID:

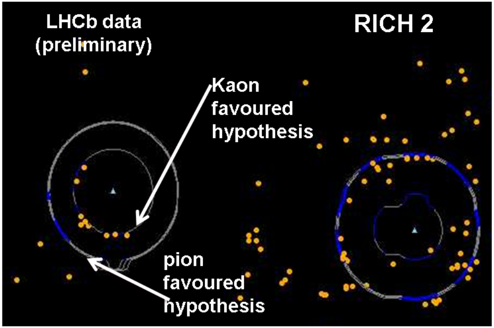
- Identification of final state particles: supervised learning, multiclass classification
- Evaluation of PID efficiency from calibration data samples: unsupervised learning, density estimation
- Simulation of PID response: generative models.

See [next talk by Fedor Ratnikov]
"Fast calorimeter simulation in LHCb"

PID strategy and performance in Run2: see [talk by Carla Marin Benito]

PID variables

- Low-level PID information: likelihoods obtained from info of individual detectors
  - Rings in RICH detectors
  - Clusters in calorimeter
  - Hits in muon system

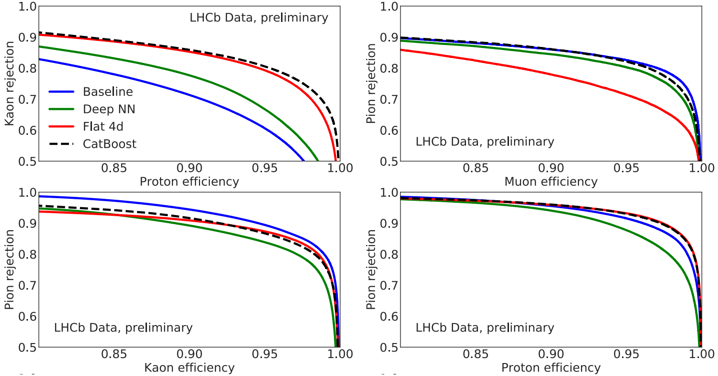


- Higher-level variables (ProbNN):
  - ANN output combining the above (+auxiliary info from tracking etc.)
  - 6 models for each of charged PID hypotheses + "ghost" (tracks not representing real particles)
  - Trained on MC
  - Baseline approach: MLP implemented in TMVA, 1 hidden layer

Advanced classification techniques for charged PID

Trying new classification techniques

- XGBoost [arXiv:1603.02754]
- CatBoost [arXiv:1706.09516]
- Boosting to flatness [JINST 10 (2015) T03002]
- Deep Neural Networks (`keras` library)



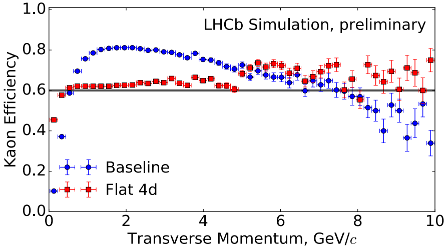
Improvements are possible with advanced classifiers, but careful choice of training samples is needed (more sensitive to kinematic properties than baseline).

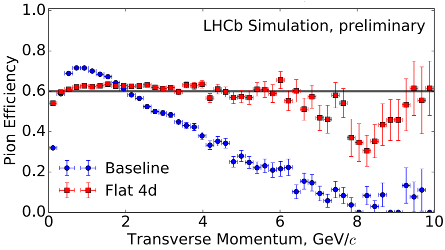
PID classifier with flat efficiency

- Typically, PID performance depends on track kinematics (p, η) and event multiplicity
- Systematics-limited measurements: having a classifier with efficiency independent of kinematics/multiplicity is an advantage
- Flat4d: classifier trained with flatness term in loss function

[JINST 10 (2015) T03002]

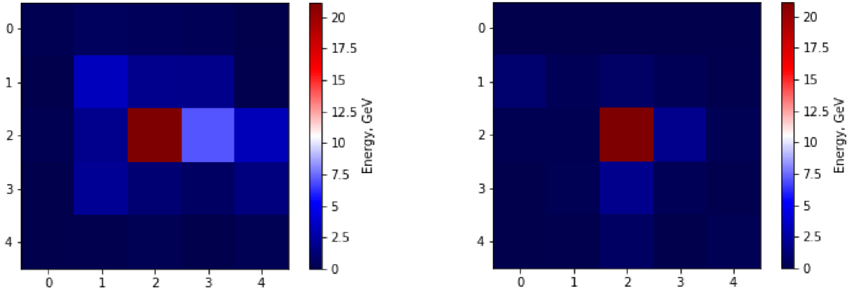
$$\mathcal{L} = \mathcal{L}_{\mathrm{exp}} + \alpha \mathcal{L}_{\mathrm{FL}}, \qquad \mathcal{L}_{\mathrm{FL}} = \sum_b \int |F_b(s) - F(s)|^2 ds$$





Neutral PID

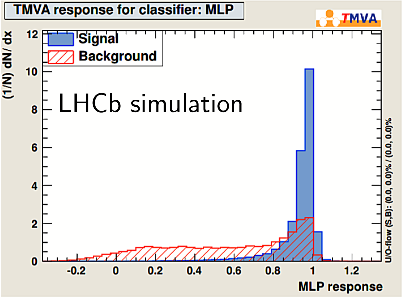
- Radiative decays (e.g. $B \to K^* \gamma$): sensitive to New Physics, energetic photons in the final state
- Large backgrounds from π^0: high-momentum π^0 do not form separated clusters in ECAL.

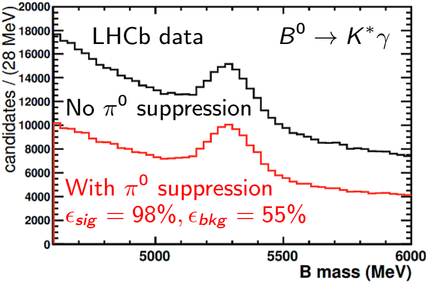


- Pattern recognition to separate γ from π^0

γ/π^0 separation: baseline classifier

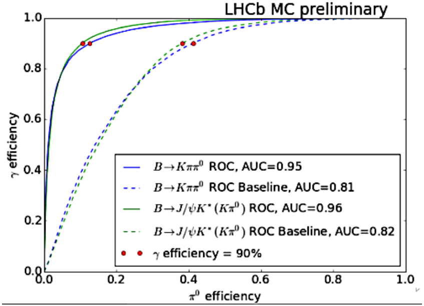
- Input features based on 3×3 "image" around a center of the cluster:
  - Shape of the cluster (width, tails, eccentricity, orientation)
  - Energies of the most and 2nd-most energetic cells
  - Hit multiplicity and shape in the preshower cells
- Output: MLP with 2 hidden layers in TMVA

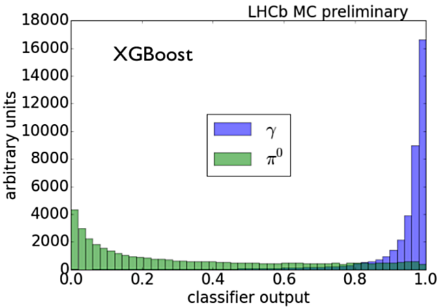




γ/π^0 separation: new classifier

- Input features: energy deposition in 5×5 ECAL and PS cells ("raw images")
- Training samples: $B \to K\pi\gamma$ (signal) and $B \to K\pi\pi^0$ (background)
- Several classifiers tried:
  - ANNs with 1–2 hidden layers, different optimisers (Adamax, Adagrad, SGD)
  - BDTs (XGBoost, CatBoost, LightGBM)





BDT with XGBoost shows the best performance (AUC=0.95)

PID resampling and correction

PID response is widely used in physics selections $\Rightarrow$ need to reproduce it precisely in simulation to evaluate selection efficiency, background contamination.

PID performance is a complicated function of track kinematics and event multiplicity $\Rightarrow$ multivariate problem.

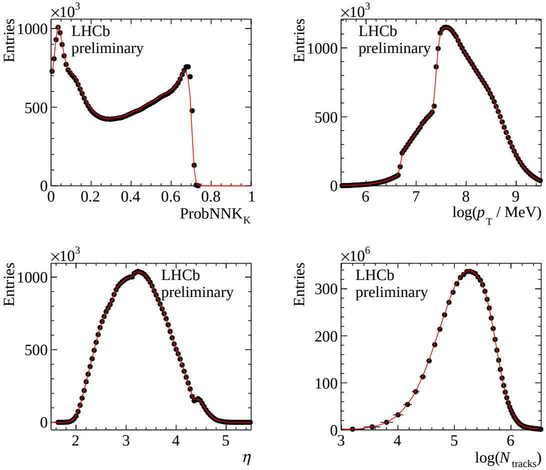
Two procedures developed at LHCb:

- **Resampling (PIDGen)**: Using the known 4D distribution of calibration sample in *PID* variable, track kinematics (p_T and η) and event multiplicity (N_{tracks}), *generate* PID variable that looks like in data for any given track kinematics and multiplicity.
- **Variable transformation (PIDCorr):** Using the above 4D distributions for data and MC, construct a function that transforms simulated PID response such that it matches data.
  This approach preserves correlations between different PID responses for the same track (e.g. π and K probabilities).

[arXiv:1803.00824]

PIDGen and PIDCorr: input variables

sPlot technique applied to calibration samples to statistically subtract background [NIM A555 (2005) 356]



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

Describe PDFs of the sWeighted calibration sample in 4 variables:

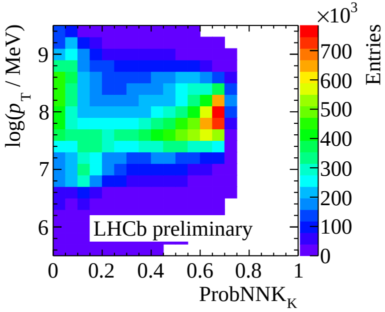
- PID variable (transformed to avoid sharp peaks)
- $\log p_T$
- Pseudorapidity η
- Track multiplicity $\log N_{\text{tracks}}$

PIDGen and PIDCorr: kernel density estimation

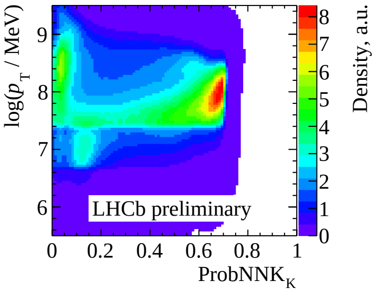
Four-dimensional kernel density estimation of calibration data performed using `Meerkat` library [JINST 10 (2015) P02011] [HepForge]

- Provides kernel-based correction to the approximated density
- Efficient with multidimensional PDFs

Example: two-dimensional projections onto $PID - \log p_T$:



Binned sWeighted data



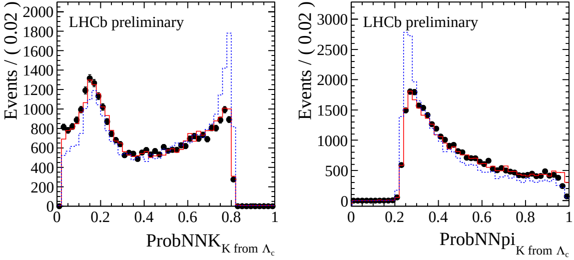
Kernel density estimation

PIDGen: validation of resampled variables

PIDGen: discard simulated PID response, resample from calibration density for a given track p_T, η and track multiplicity

$$PID_{\text{corr}} = P_{\text{exp}}^{-1}(\xi | p_T, \eta, N_{\text{tracks}})$$

Performance is validated on independent clean high-statistics data samples.



sWeighted data

Uncorrected simulation

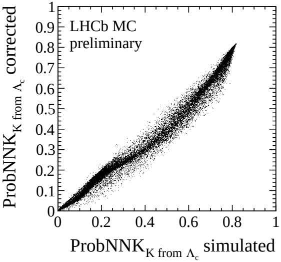
Corrected simulation (PIDGen)

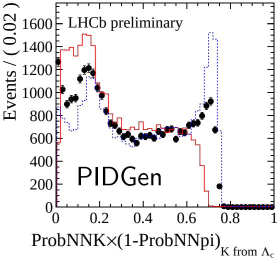
PIDGen resampled variables (ProbNNK and ProbNNpi) for a kaon track from sWeighted $\Lambda_b^0 \to \Lambda_c^+ \pi^-$, $\Lambda_c^+ \to pK^-\pi^+$ sample.

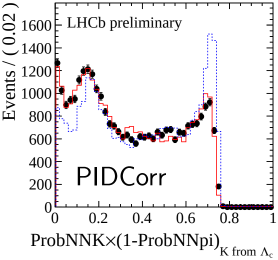
PIDCorr: preserve correlations between different PID responses for the same track. Transformation of simulated PID instead of complete resampling.

$$PID_{\mathrm{corr}} = P^{-1}_{\mathrm{exp}} \left(P_{\mathrm{MC}}(PID_{\mathrm{MC}} | p_T, \eta, N_{\mathrm{tracks}}) \, | p_T, \eta, N_{\mathrm{tracks}} \right)$$

Reproduce not only individual PID responses (ProbNNpi, ProbNNK, etc.), but also their combinations



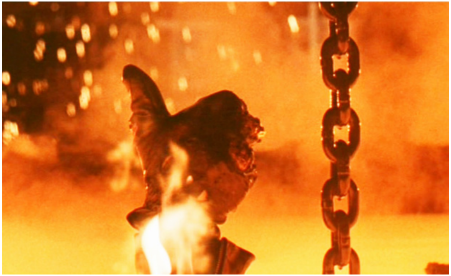




sWeighted $\Lambda^0_b \to \Lambda^+_c \pi^-$, $\Lambda^+_c \to pK^-\pi^+$ data,
uncorrected simulation,
corrected simulation (PIDGen or PIDCorr).

Summary

- Particle identification at LHCb: a broad area to apply advanced machine learning techniques
- Several new approaches tested on Run1/Run2 data:
  - Multivariate classifiers for charged and neutral particle classification
  - Density estimation of calibration data: resampling and correction of MC PID response
- PID will be even more important after LHCb upgrade: software trigger including PID information

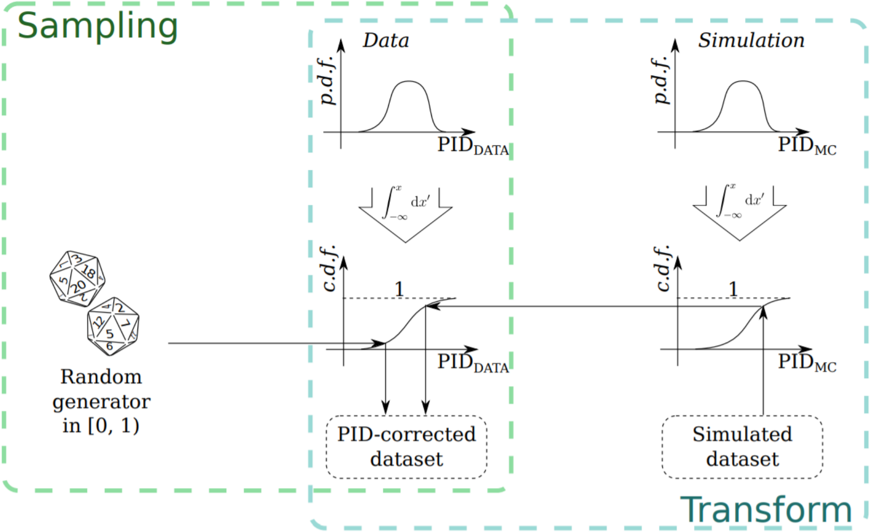


Backup

Input variables for charged ANN classifiers

Tracking	
	Total momentum
	Transverse momentum
	Quality of the track fit
	Number of clusters associated to the track
	ANN response trained to reject ghost tracks
	Quality of the fit matching track segments upstream and downstream of the magnet
RICH detectors	
	Geometrical acceptance of the three radiators, depending on the direction of the track
	Kinematical acceptance due to Cherenkov threshold for muons and kaons
	Likelihood of the electron, muon, kaon, and proton hypotheses relative to the pion
	Likelihood ratio of the below-threshold and pion hypotheses
Electromagnetic calorimeter	
	Likelihood ratio of the electron and hadron hypotheses
	Likelihood ratio of the muon and hadron hypotheses
	Matching of the track with the clusters in the *preshower* detector
	Likelihood ratio of the electron and pion hypotheses, after recovery of the Bremsstrahlung photons
Hadronic calorimeter	
	Likelihood ratio of the electron and hadron hypotheses
	Likelihood ratio of the muon and hadron hypotheses
Muon system	
	Geometrical acceptance
	Loose binary requirement already available in the hardware trigger
	Likelihood of the muon hypothesis
	Likelihood of the non-muon hypothesis
	Number of clusters associated to at least another tracks

PIDGen and PIDCorr approaches



Meerkat approach to density estimation

Traditional kernel density: data points x_i, kernel $K(x)$

$$P_{\mathrm{KDE}}(x) = \sum_i K(x - x_i)$$

`Meerkat` technique (relative kernel density estimation): [JINST 10 (2015) P02011]

$$P_{\mathrm{corr}}(x) = \frac{\sum\limits_{i=1}^{N} K(x - x_i)}{(P_{\mathrm{appr}} \otimes K)(x)} \times P_{\mathrm{appr}}(x).$$

In other words, we represent the PDF as a product of approximation PDF and kernel correction:

$$P_{\mathrm{corr}}(x) = f(x) P_{\mathrm{appr}}(x)$$

$P_{\mathrm{appr}}(x)$ takes care of boundary effects and narrow structures.
In the practical implementation, use binning with multilinear interpolation:

$$P_{\mathrm{interp}}(x) = \frac{Bin\left[\sum\limits_{i=1}^{N} K(x - x_i)\right]}{Bin\left[(P_{\mathrm{appr}} \otimes K)(x)\right]} \times P_{\mathrm{appr}}(x).$$