Advanced Machine-Learning Solutions in LHCb Operations and Data Analysis

Fedor Ratnikov

National Research University High School of Economics,
Yandex School of Data Analysis,

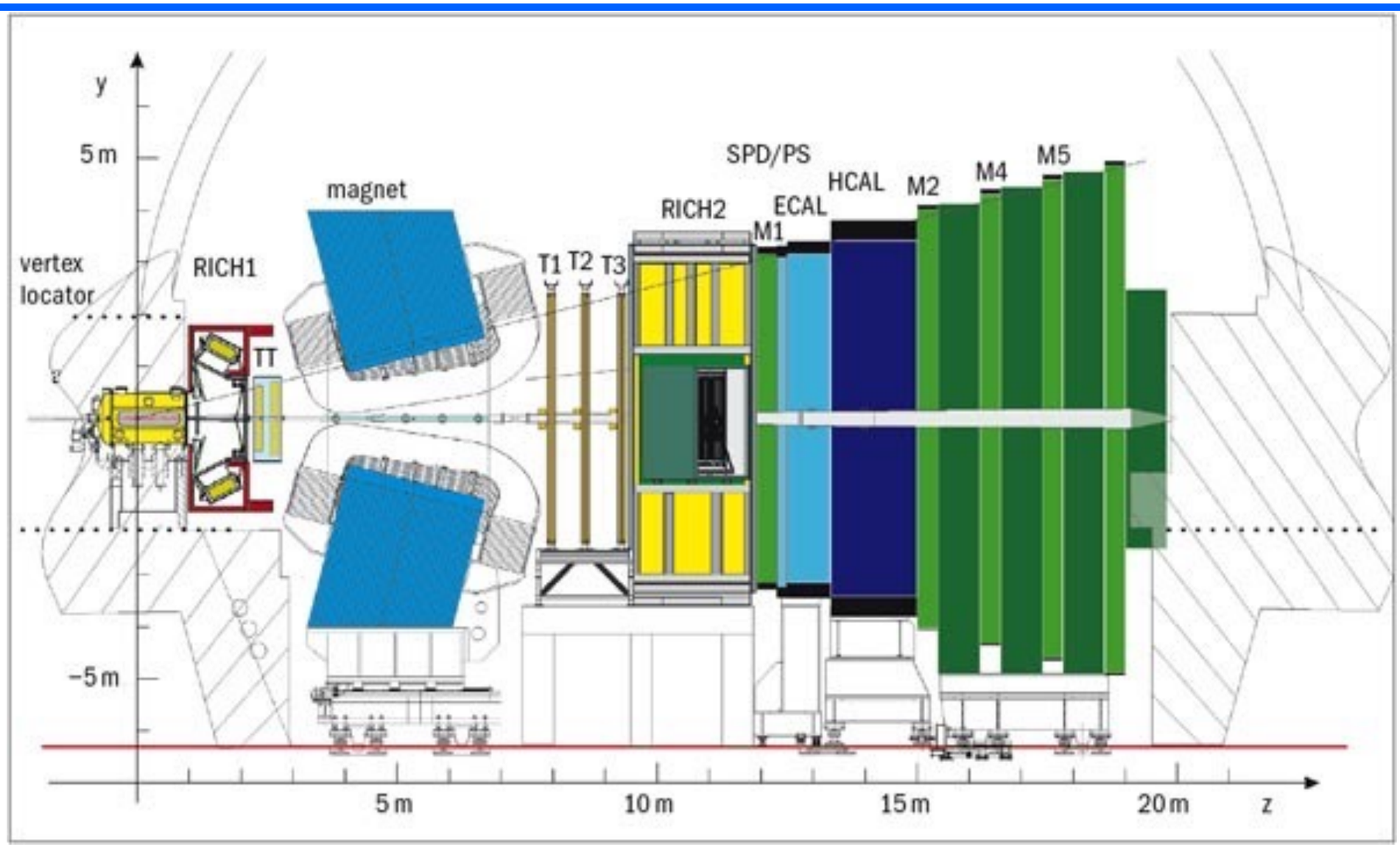
on behalf of LHCb Collaboration

Fermilab, July 31 - August 4, 2017

Content

- Introduction
- ML in Trigger
- ML in PID
- ML in Monitoring

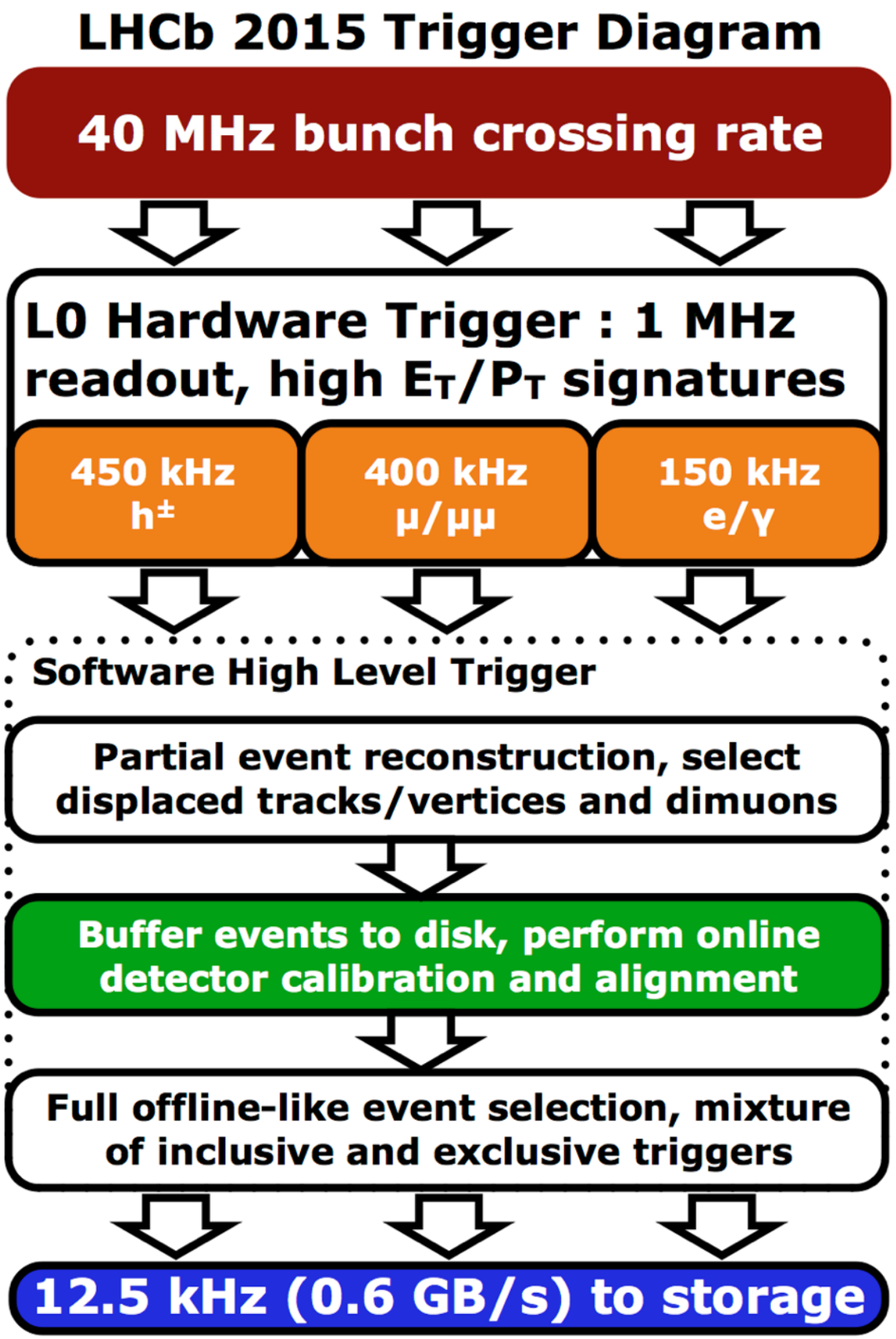
LHCb Detector @LHC



- One of four big LHC detectors — Int. J. Mod. Phys. A 30 (2015) 1530022
- Primary physics program: charm and beauty physics
  - need excellent track reconstruction and particle ID

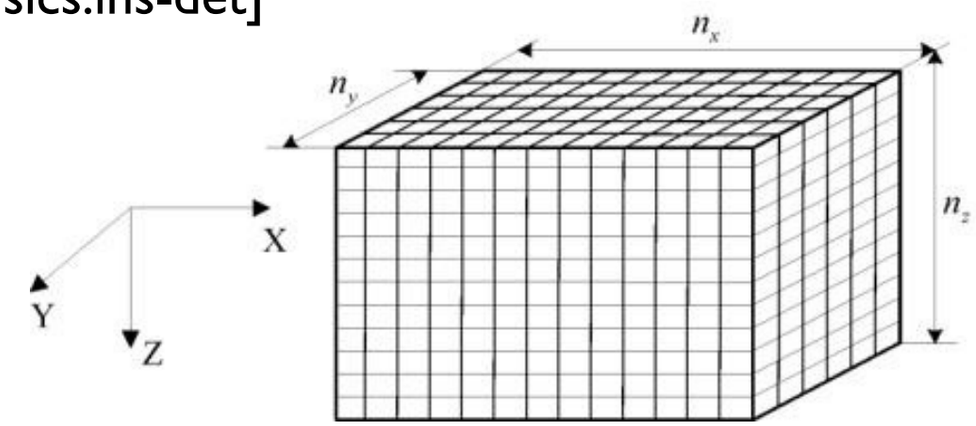
ML in LHCb Trigger

- Use real time alignment, calibration and reconstruction in Run II
  - need RECO online ≡ RECO offline
- Use BDT and/or NN in
  - track reconstruction
  - fake tracks suppression
  - particle ID
  - High Level Trigger decisions



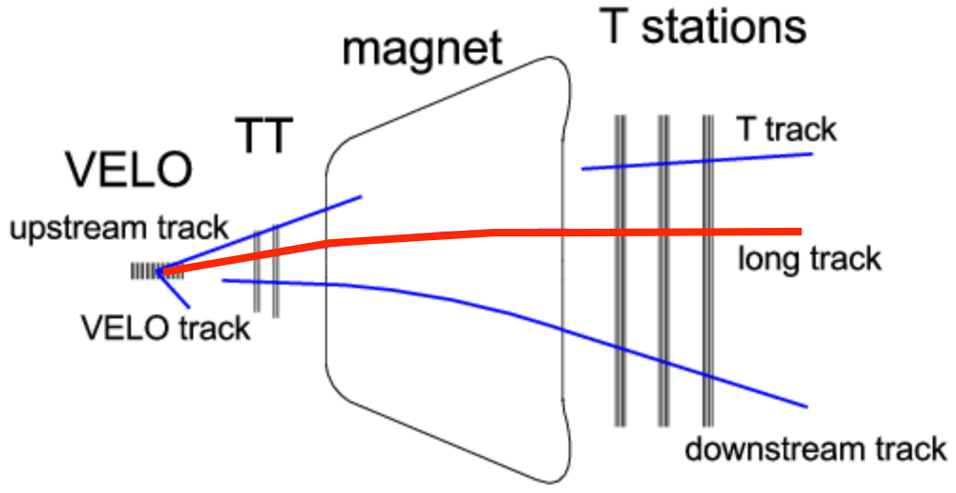
BDT Speedup

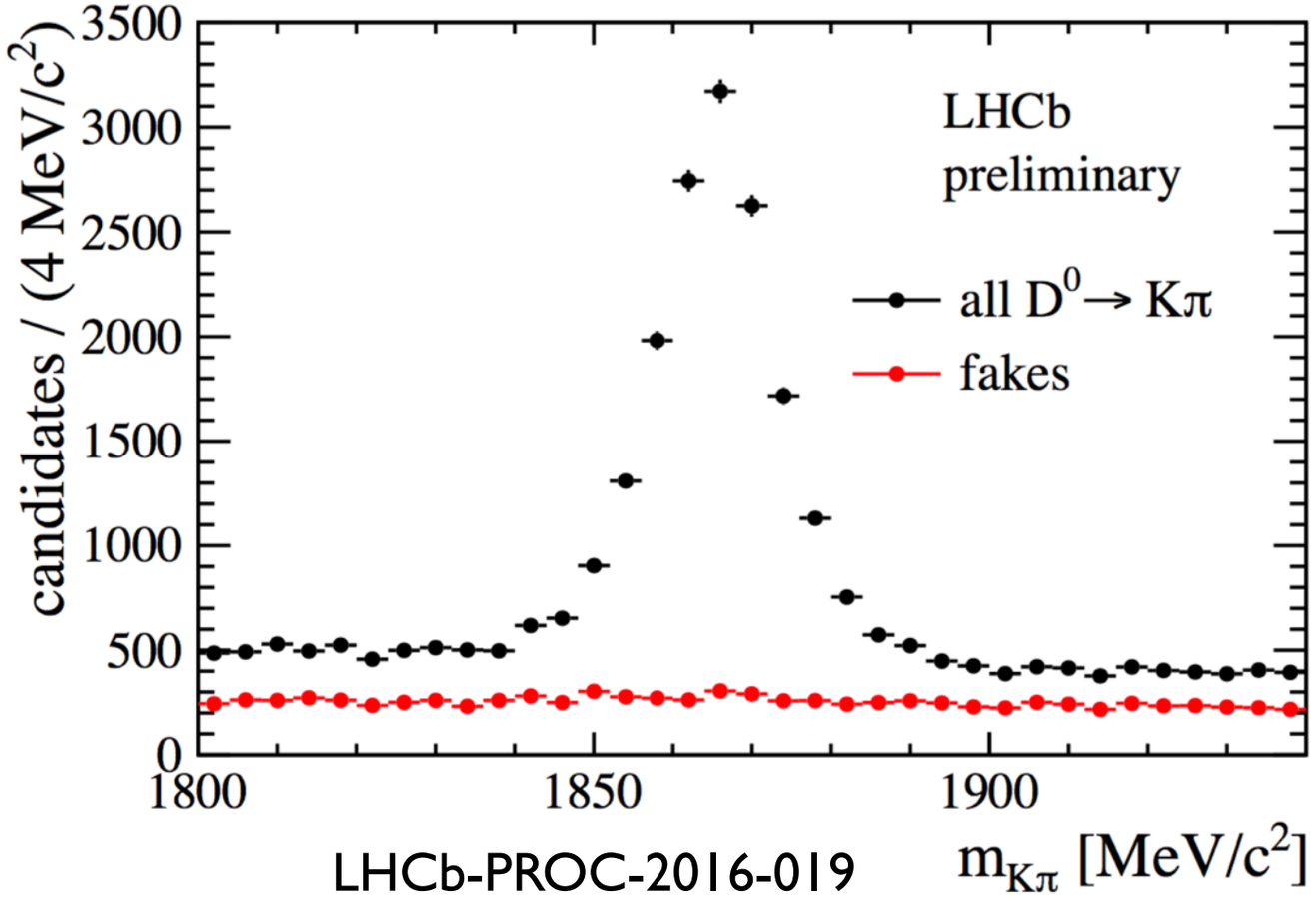
- Strong requirements to the speed of the code which runs in real time
- Out of the box ML solution usually produce heavy classifiers
  - need speed up / simplify
- Two main approaches to speed up BDT
  - post-pruning
    - remove less important branches/trees
    - reduce number from thousands to hundred
    - trading speed for precision
  - Bonsai BDT (BBDT) arXiv:1210.6861 [physics.ins-det]
  - discretisation of input features
    - converts trees into Look Up Table
      - trading speed for memory
        - limited by ~1Gb



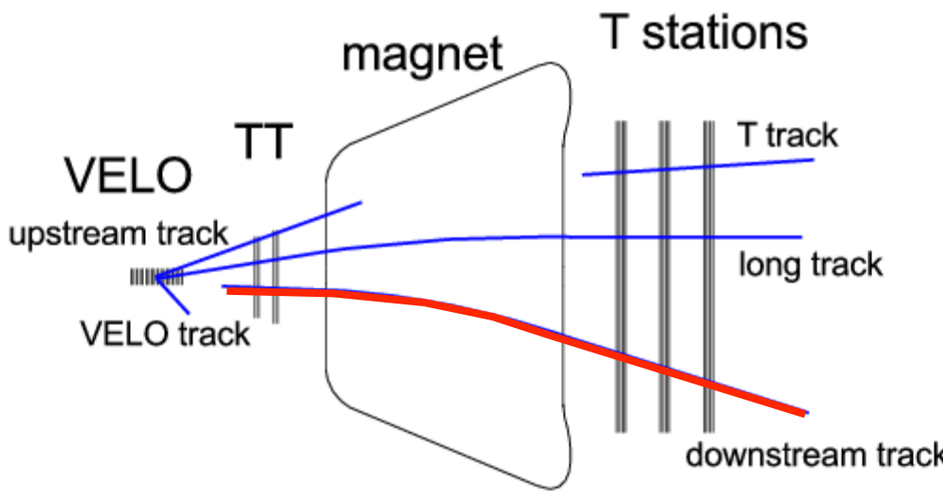
Long Tracks Reconstruction

- In Run II apply NN-based clones and fakes rejection online
- Use two deep NN in forward tracking
  - for track candidate selection after stereo fit
    - 16 Input nodes, 17,9,5 HL nodes
  - for rejection of bad 4-layer clusters
    - 9 Input nodes, 16, 10 HL nodes





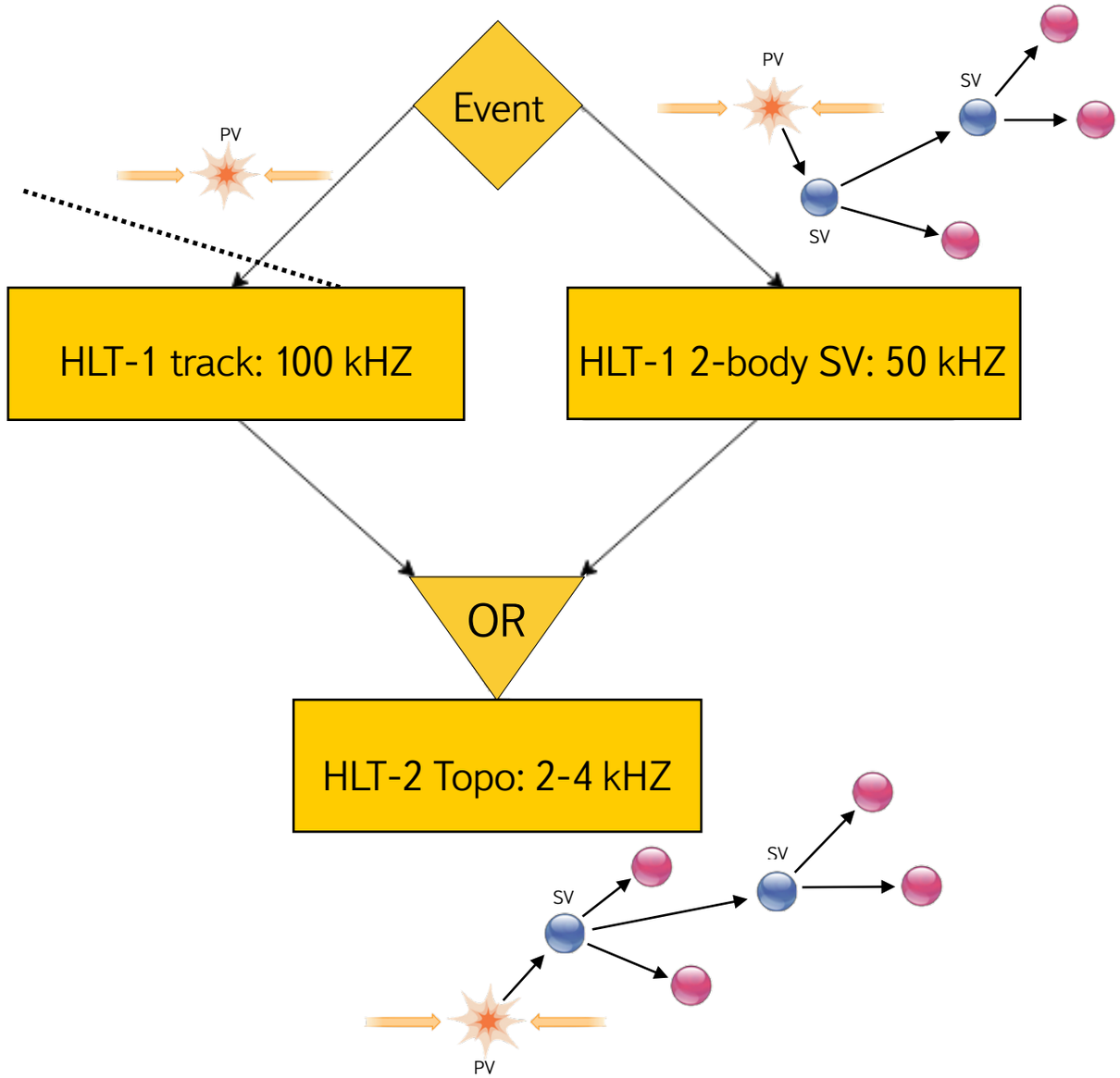
Downstream Tracking



- Downstream tracking contains two Multivariate classifiers
  - reject as much fake T-seeds as possible: avoiding unnecessary reconstruction
    - BBDT
  - final accepting tracks: further reducing fake tracks
    - MLP
- Improved both fake tracks reduction and signal efficiency gain by 3-5%
  - implemented for 2017 operation

Topological Trigger

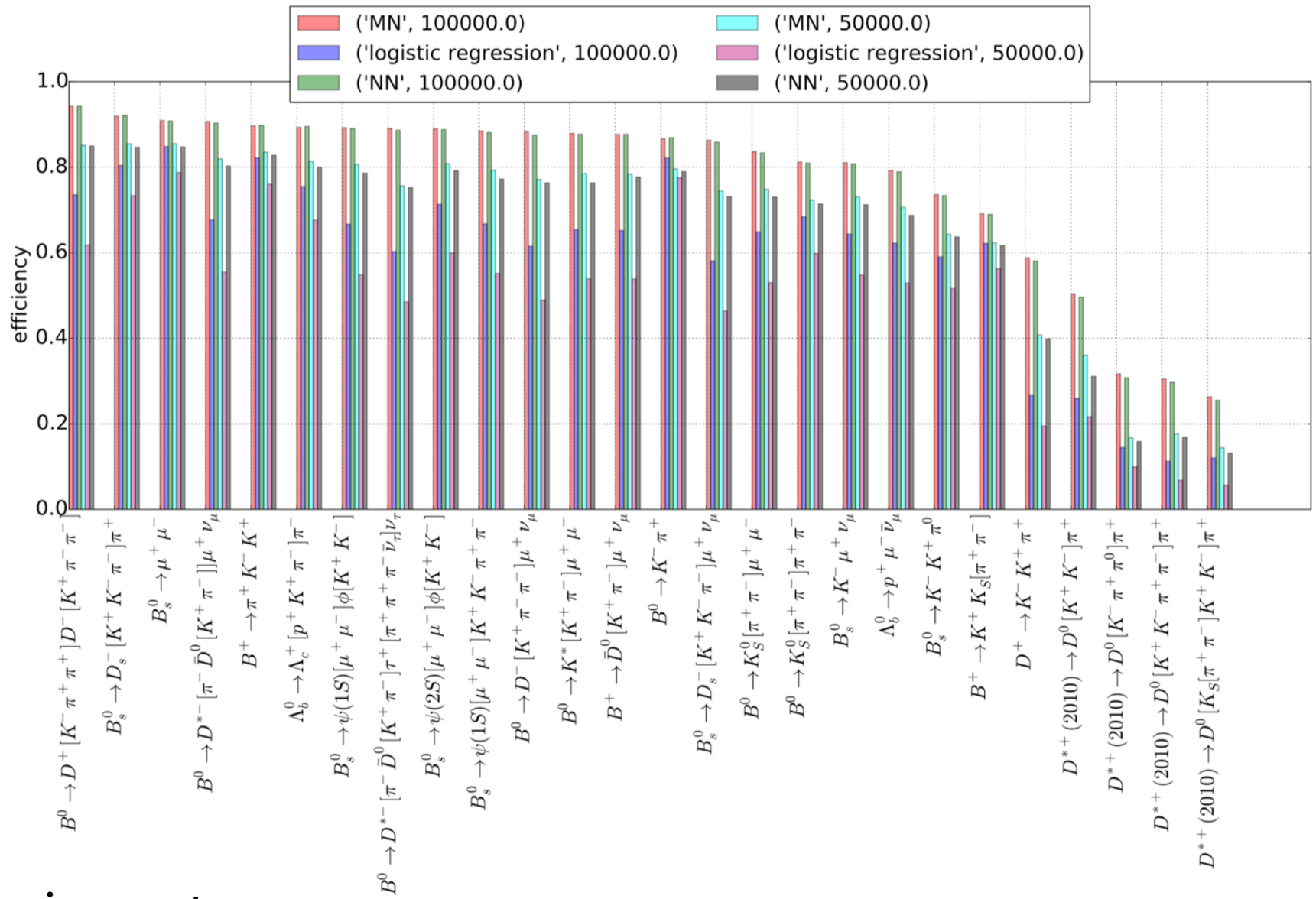
- HLT-1 track is looking for either one super high p_T or high displaced track
- HLT-1 2-body SV classifier is looking for two tracks making a vertex
- HLT-2 improved topological classifier uses full reconstructed event to look for 2, 3, 4 and more tracks making a vertex
  - it's all about finding Secondary Vertices
  - most of LHCb Run I papers were made using this trigger
- We aggressively re-optimised this trigger for Run II operation



Compare Different Classifiers

- NN: neural net
- MN: Yandex's MatrixNet (a BDT)
- Logistic Regression

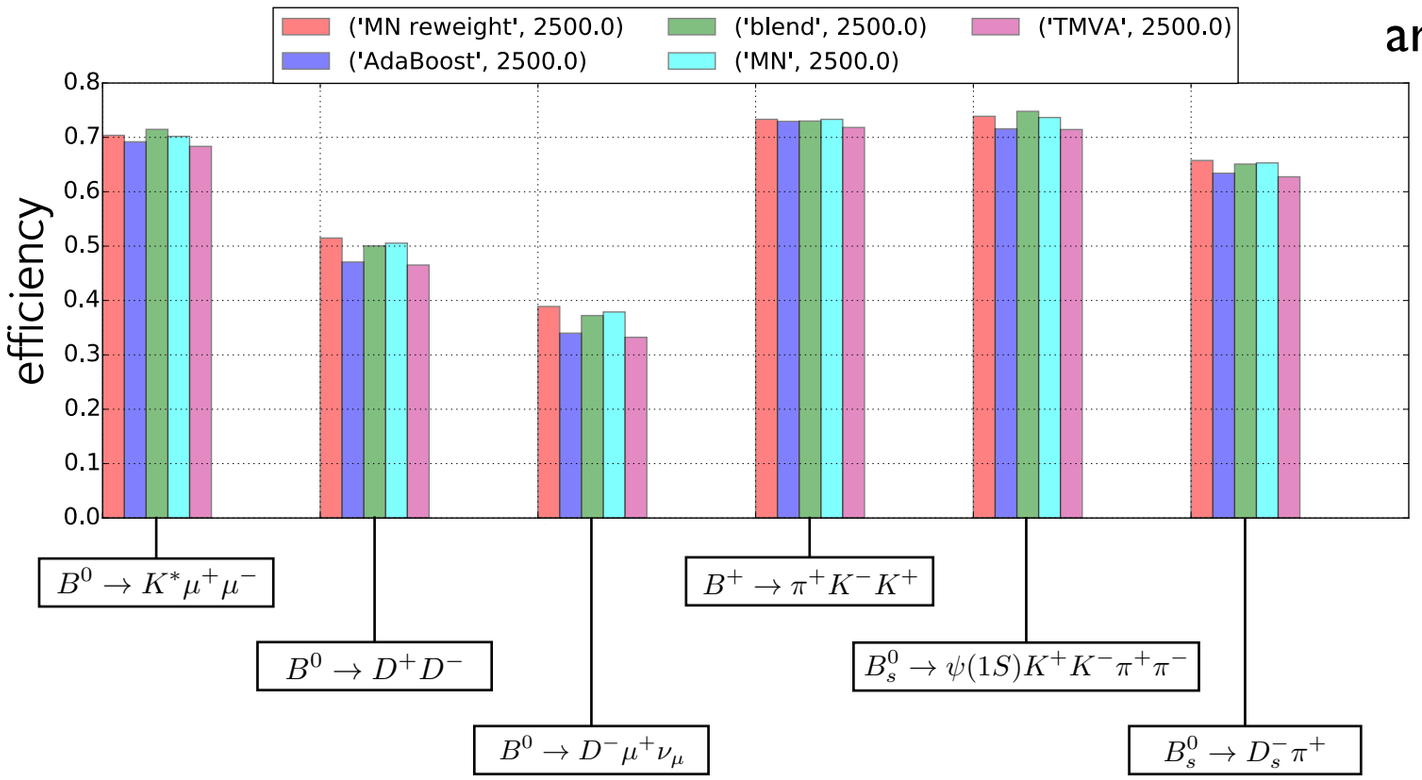
arXiv:1510.00572 [physics.ins-det]



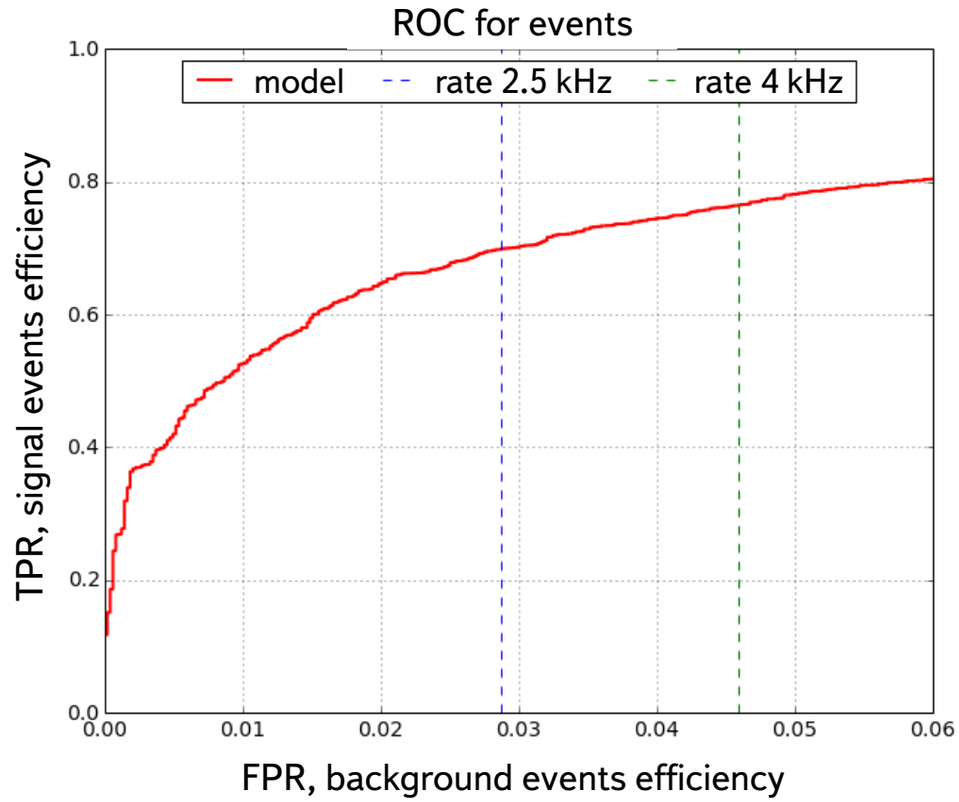
- Compare efficiency at given rate
- NN and MN are close, but MN usually wins

Aggregated Metric

arXiv:1510.00572 [physics.ins-det]



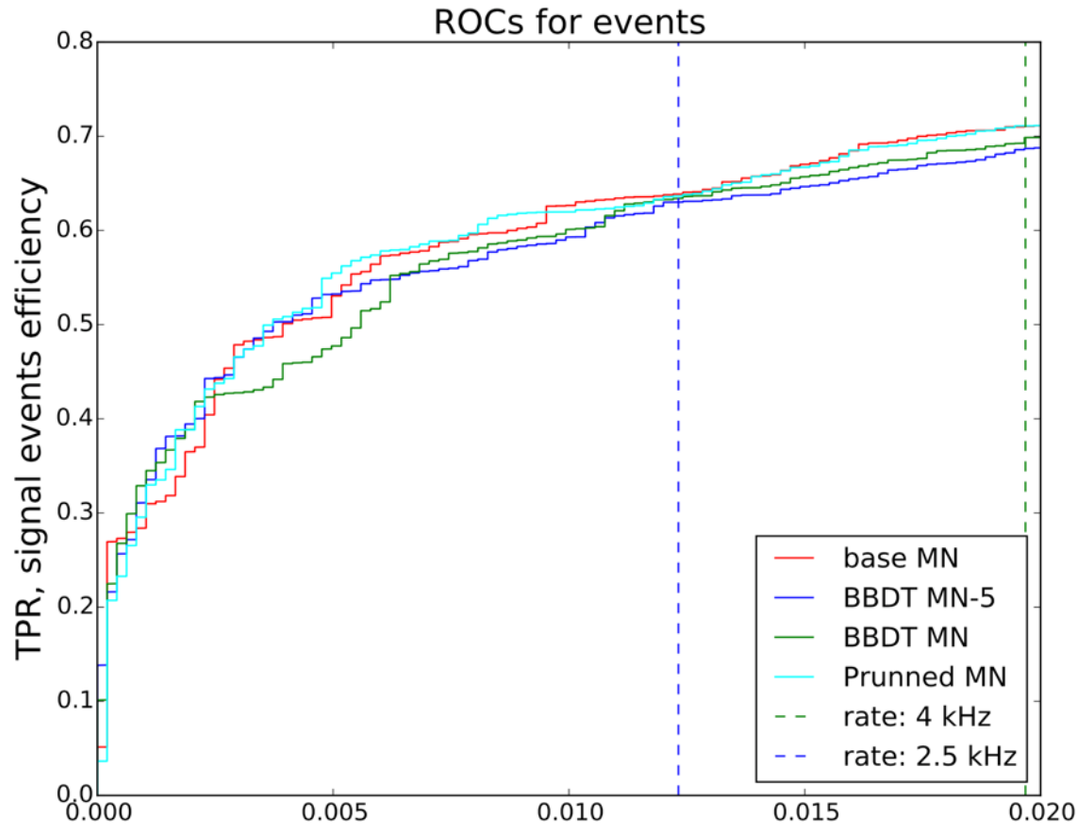
- Need to have aggregative metric to measure quality
- Weight signal events in such a way that decays have the same sum of weights
- Optimise ROC curve in a region with small FPR



Speed up and Improvements

arXiv:1510.00572 [physics.ins-det]

- Need to speed up trees
  - compare BBDT with post-pruning
    - post-pruning is slightly better



- Gain 10%..70% efficiency for different channels

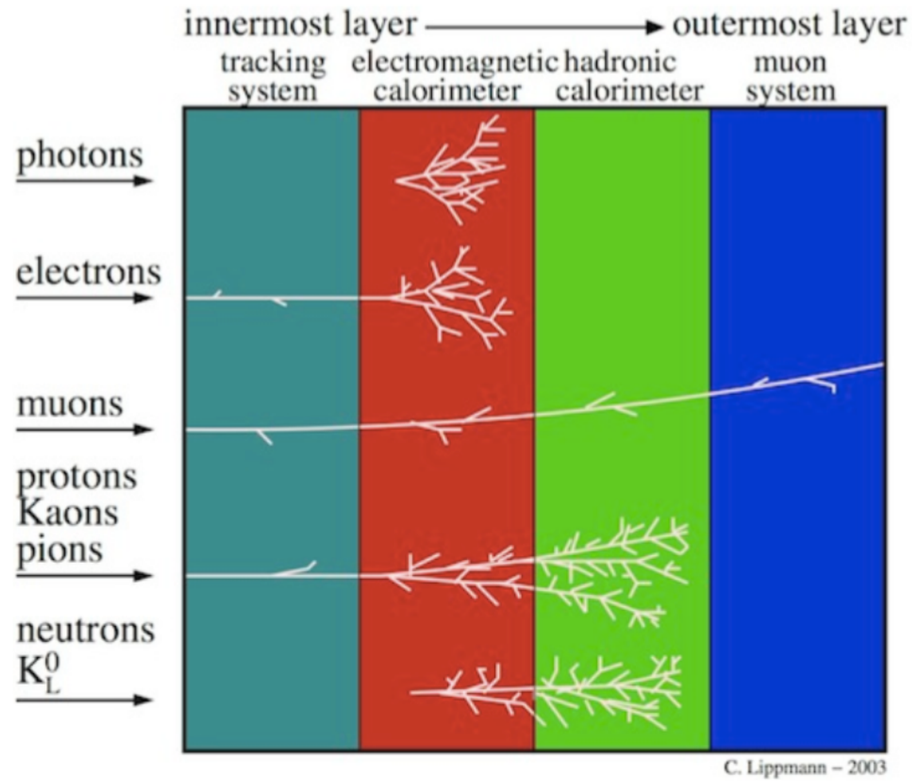
arXiv:1510.00572 [physics.ins-det]

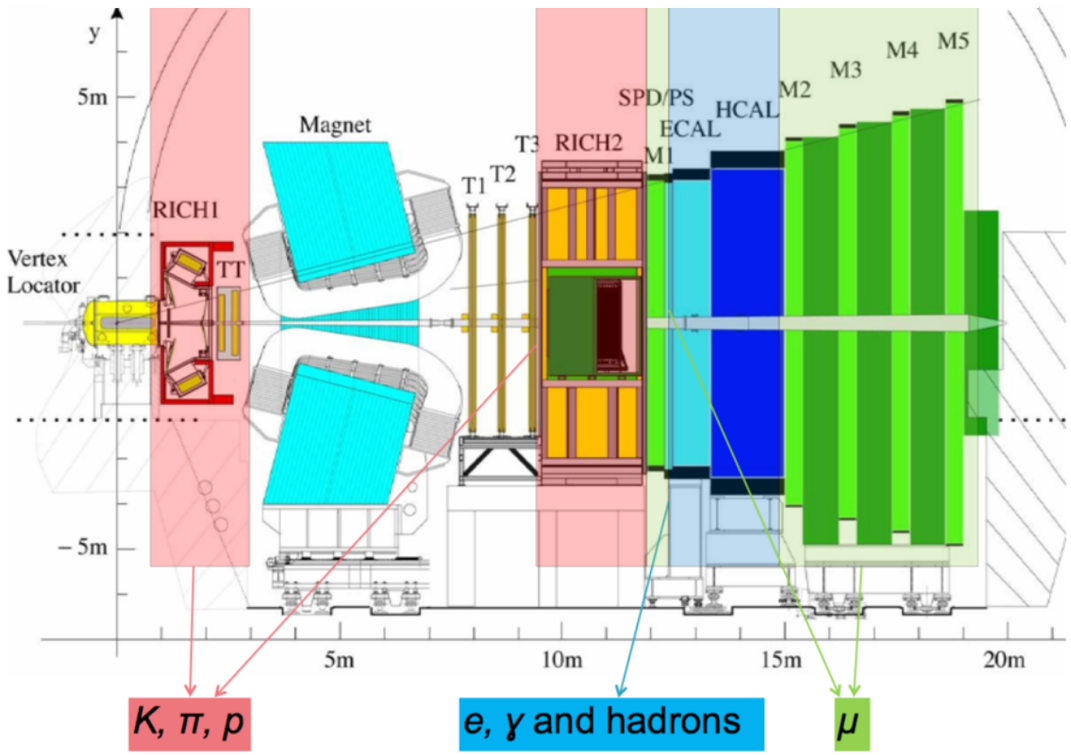
ε_{HLT}(Run II) / ε_{HLT}(Run I)

mode	2.5 kHz	4. kHz
$B^0 \to K^*[K^+\pi^-]\mu^+\mu^-$	1.64	1.72
$B^+ \to \pi^+K^-K^+$	1.59	1.65
$B_s^0 \to D_s^-[K^+K^-\pi^-]\mu^+\nu_\mu$	1.14	1.47
$B_s^0 \to \psi(1S)[\mu^+\mu^-]K^+K^-\pi^+\pi^-$	1.62	1.71
$B_s^0 \to D_s^-[K^+K^-\pi^-]\pi^+$	1.46	1.52
$B^0 \to D^+[K^-\pi^+\pi^+]D^-[K^+\pi^-\pi^-]$	1.40	1.86

Multi-Class Charge Particles ID

- Problem: identify charged particle associated with a track (multiclass classification problem)
- particle types: Ghost, Electron, Muon, Pion, Kaon, Proton.
- LHCb detector provides diverse information, collected by subdetectors: CALO, RICH, Muon and Track observables
- this information can be efficiently combined using ML
- Monte Carlo simulated samples for various decays are available (6 millions tracks in training and 6 millions in test)

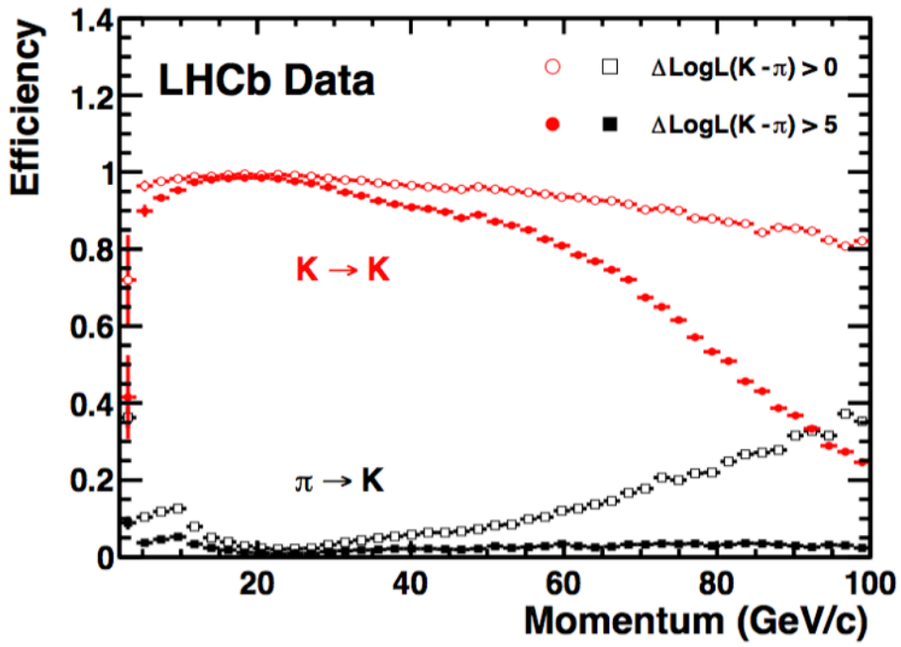


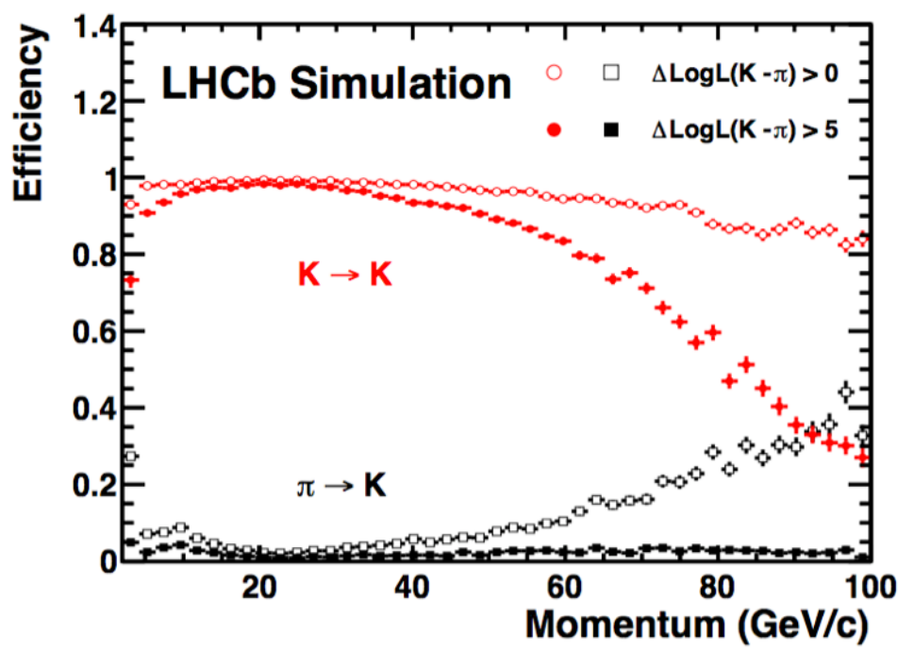


Particle ID Approaches: Δ log $\mathscr{L}$

- Separate classification by subsystems (no MVA)
  - convert particle response in the subsystem for likelihood of particular ID hypothesis
    - two RICH detectors
    - calorimeters: pre-shower, ECAL, HCAL
    - muon systems
  - combine likelihood of different subsystems into global likelihoods of different ID hypothesis
    - $\mathscr{L}(\text{id}) = \mathscr{L}_{\text{RICH}}(\text{id}) \times \mathscr{L}_{\text{Calo}}(\text{id}) \times \mathscr{L}_{\text{Muon}}(\text{id})$

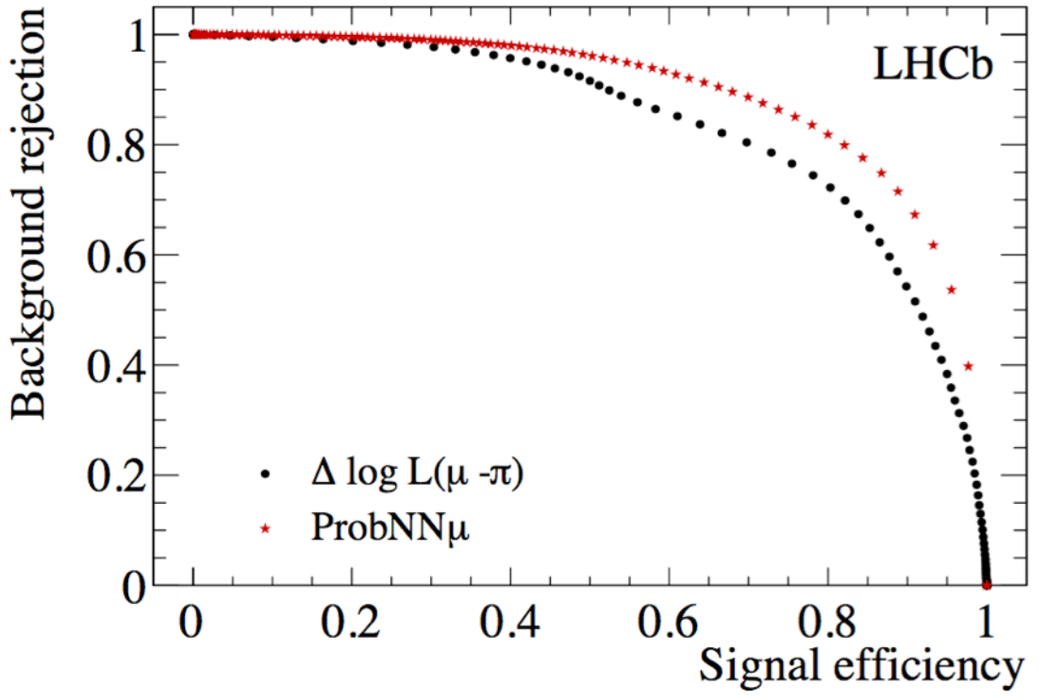
Int. J. Mod. Phys. A 30 (2015) 1530022

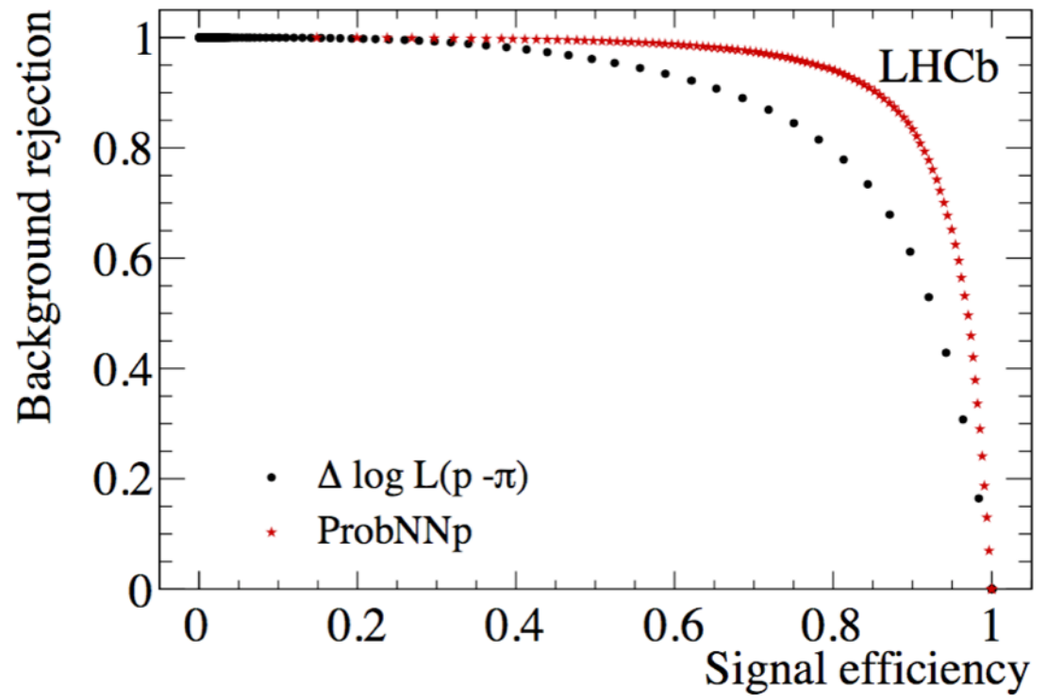




Combined Particle ID, ProbNN

- Combined one-class classification
  - information from all subsystems is analysed together using neural network with one hidden layer (TMVA MLP)
  - classifiers “this ID” vs “any other ID” are optimised for every hypothesis
  - best established approach, mostly used by current physics analyses



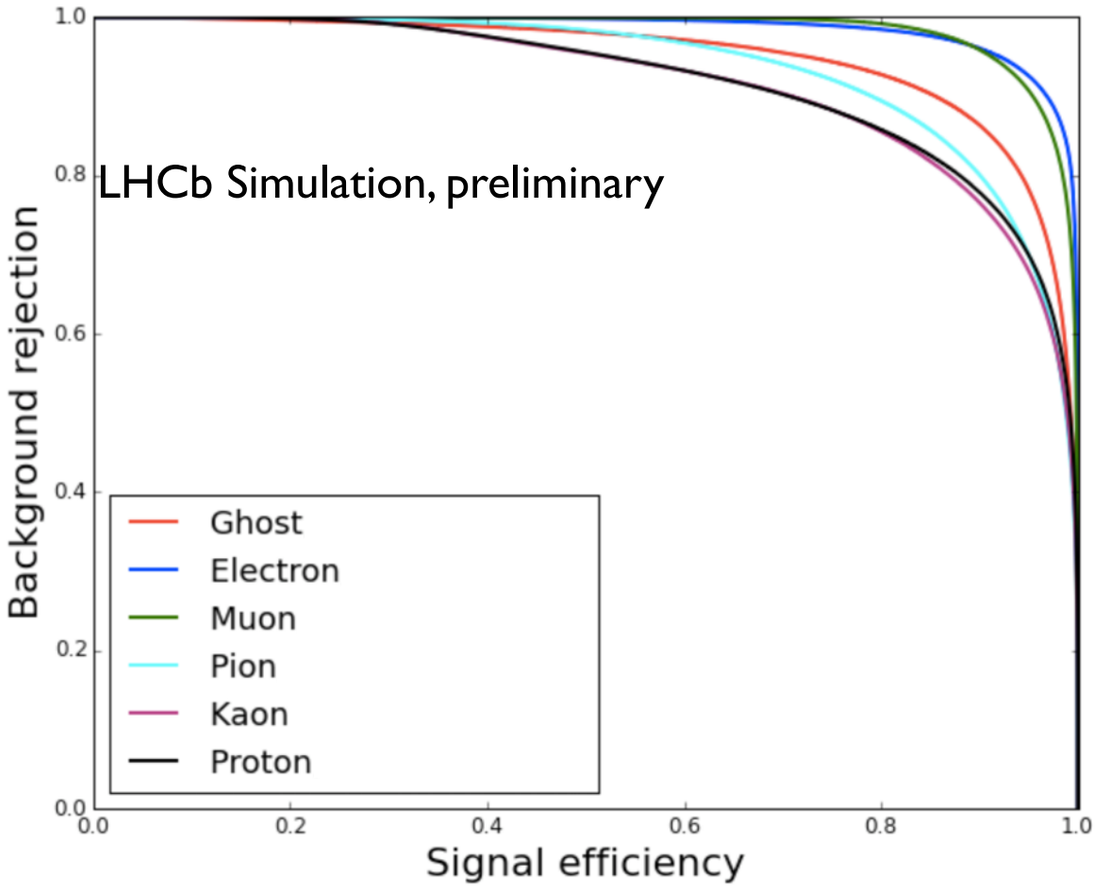


Int. J. Mod. Phys. A 30 (2015) 1530022

$\Sigma^+ \to p\mu^+\mu^-$

Further Approaches: Baseline

- Work on further improvement of the particle identification quality
- Consider One-class approach (ProbNN) as a baseline
  - consists of 6 binary classification models: One-vs-Rest
  - good separation quality
    - ROC AUC ~0.91 .. 0.99 for different hypothesis
    - statistical AUC uncertainties for used simulated samples are ~0.0002
- Consider (1. - ROC AUC) for the same ID as a metric for quality comparison



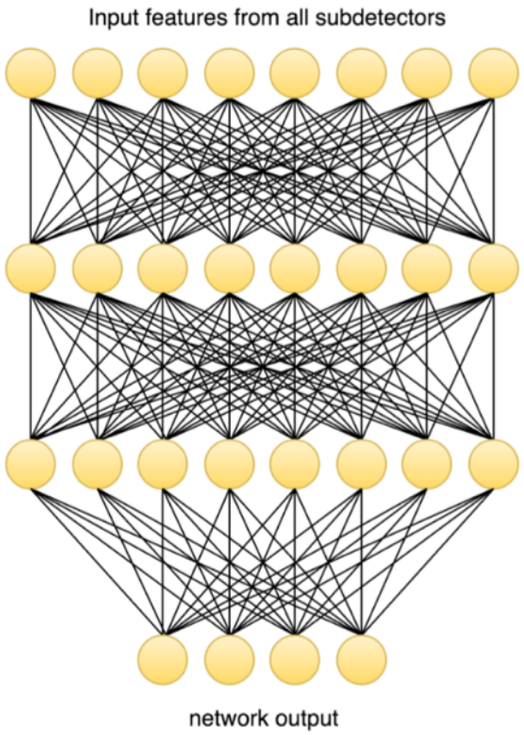
(1-AUC)/(1-AUC$_{baseline}$)
<1 : AUC closer to 1.0, better than baseline
>1 : AUC closer to 0.5, worser than baseline

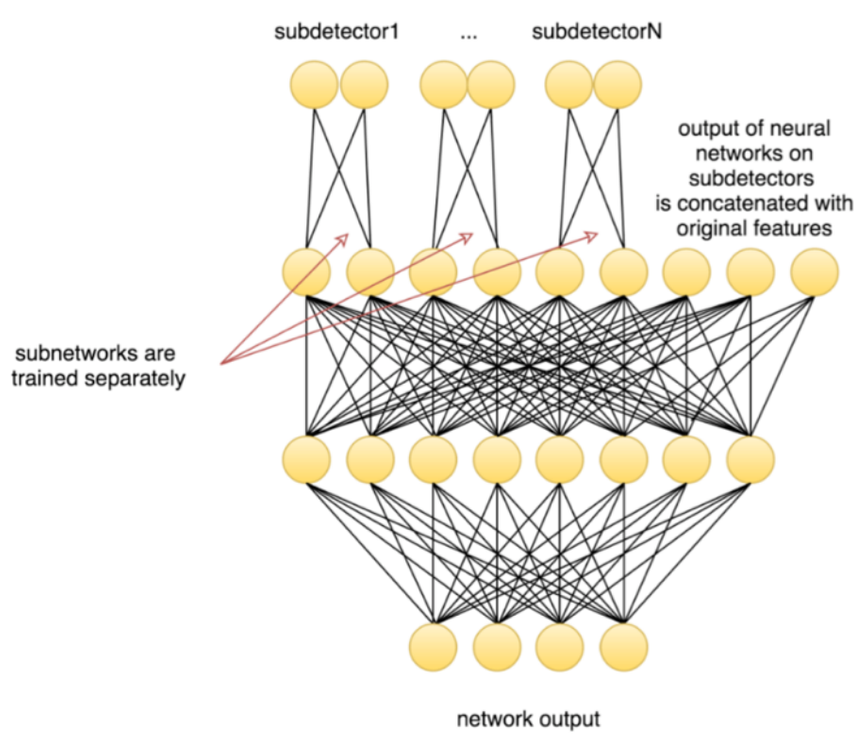
Combined Multi-class Classification

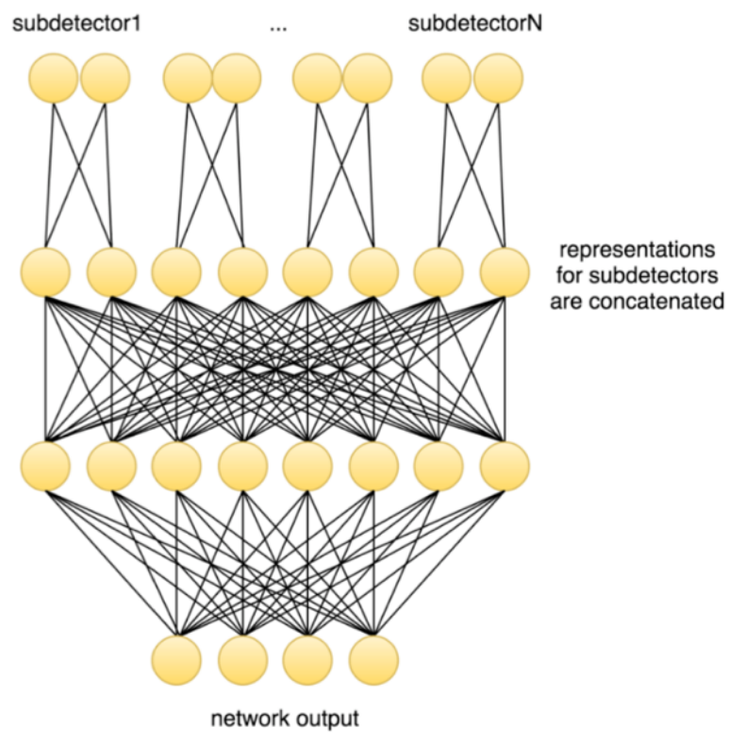
- Binary classification is widespread
  - major ML use-case in HEP
  - binary classification is very simple to implement
    - some algorithms are defined only for binary classification
- XGBoost, CatBoost, sklearn and most neural networks implementations support multi-class implementations
- Combined multi-class PID
  - information from all subsystems is analysed together using MVA
    - classifier gives weight to each hypothesis, $\sum w(id) = 1$
  - this is new approaches being tested on MC

NN

- The number of weights (parameters) in Neural Network is similar for binary classification and multi-class classification
- Computationally multi-class classification has (almost) the same complexity as binary classification (unless there are hundreds of classes and more)







(1-AUC)/(1-AUC$_{baseline}$) LHCb Simulation, preliminary

	Ghost	Electron	Muon	Pion	Kaon	Proton
baseline	1	1	1	1	1	1
deep NN	-29 %	-41 %	-52 %	-37 %	-20 %	-17 %

BDT

- Multi-class with XGBoost
- Multi-class with CatBoost

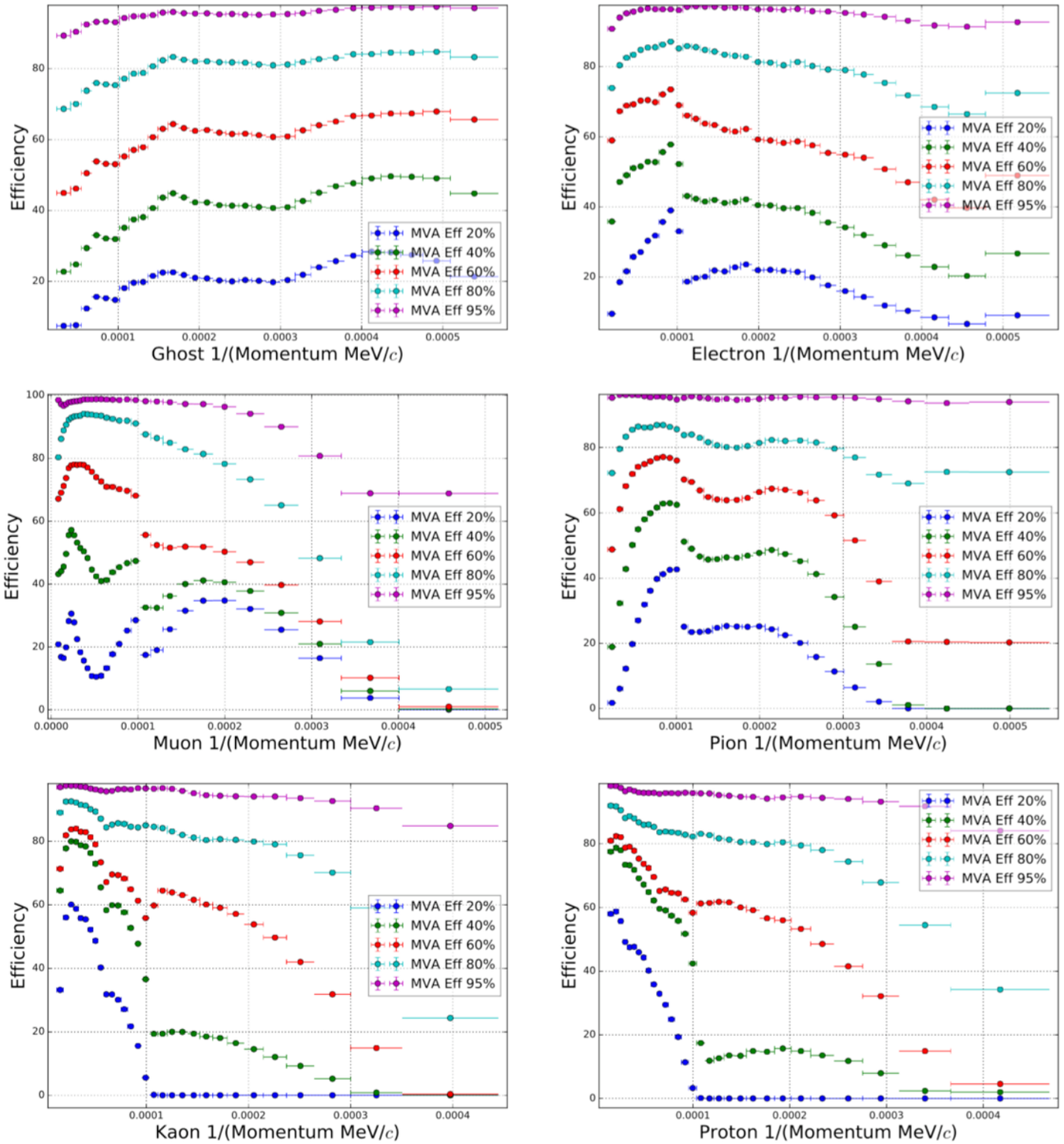
(1-AUC)/(1-AUC$_{baseline}$) LHCb Simulation, preliminary

	Ghost	Electron	Muon	Pion	Kaon	Proton
baseline	1	1	1	1	1	1
deep NN	-29 %	-41 %	-52 %	-37 %	-20 %	-17 %
XGBoost	-24 %	-37 %	-50 %	-34 %	-18 %	-15 %
CatBoost	-30 %	-43 %	-54 %	-37 %	-20 %	-18 %

Flatness

- The whole PID information strongly depends on particle momentum, that leads to strong dependency between PID efficiency and momentum
- In some analyses we need to have flat PID along signal particle momentum
- Requirement for flatness over a set of variables may be added into loss function for training
  - by price of some quality degradation
  - see arXiv:1305.7248 [nucl-ex] and arXiv:1410.4140 [hep-ex] for more details

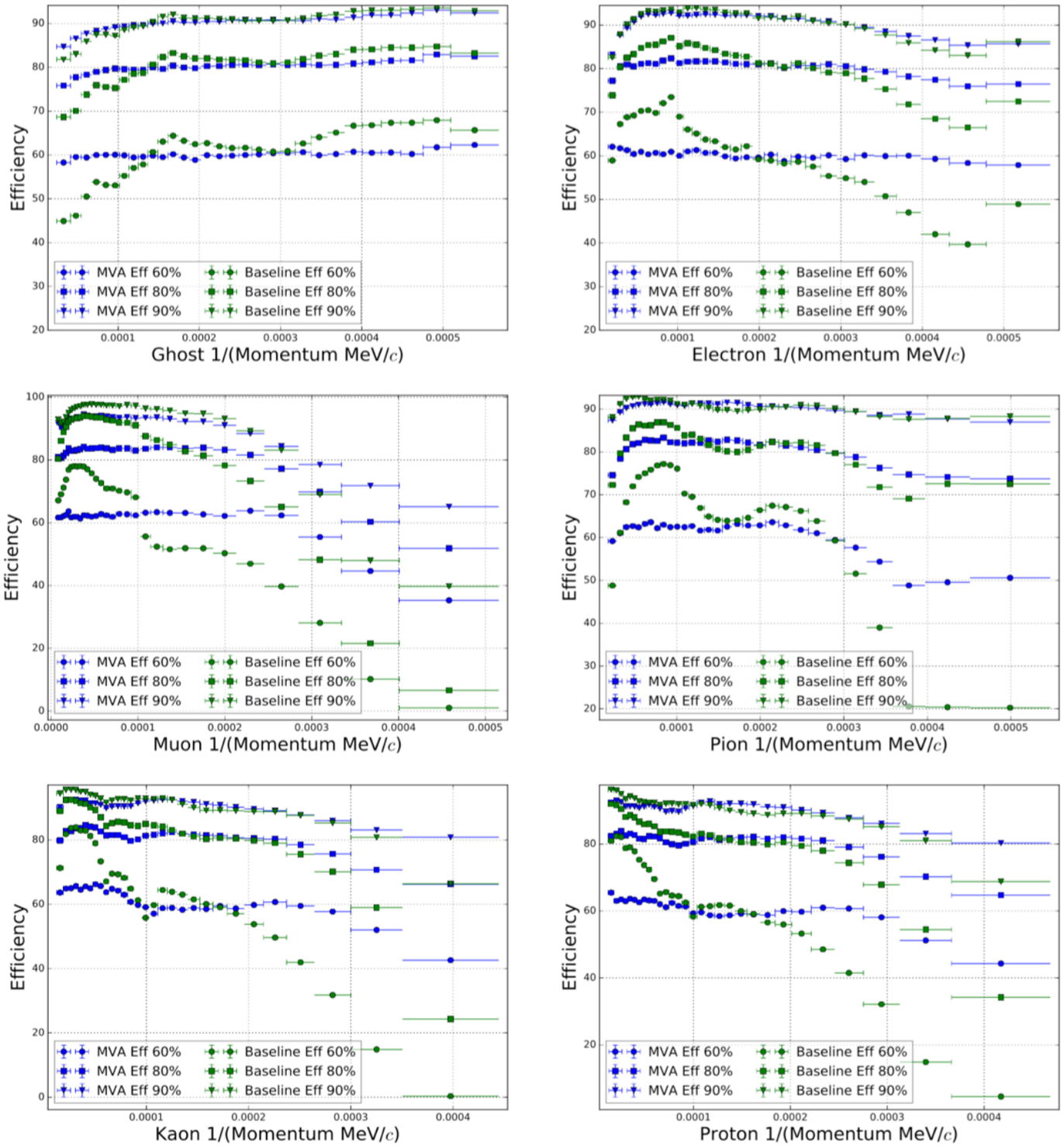
LHCb Simulation, preliminary



Flat Models

- Uniform boosting provides flatness along 4 variables at once
  - p
  - pT
  - η
  - nTracks

LHCb Simulation, preliminary



Flat Models Quality

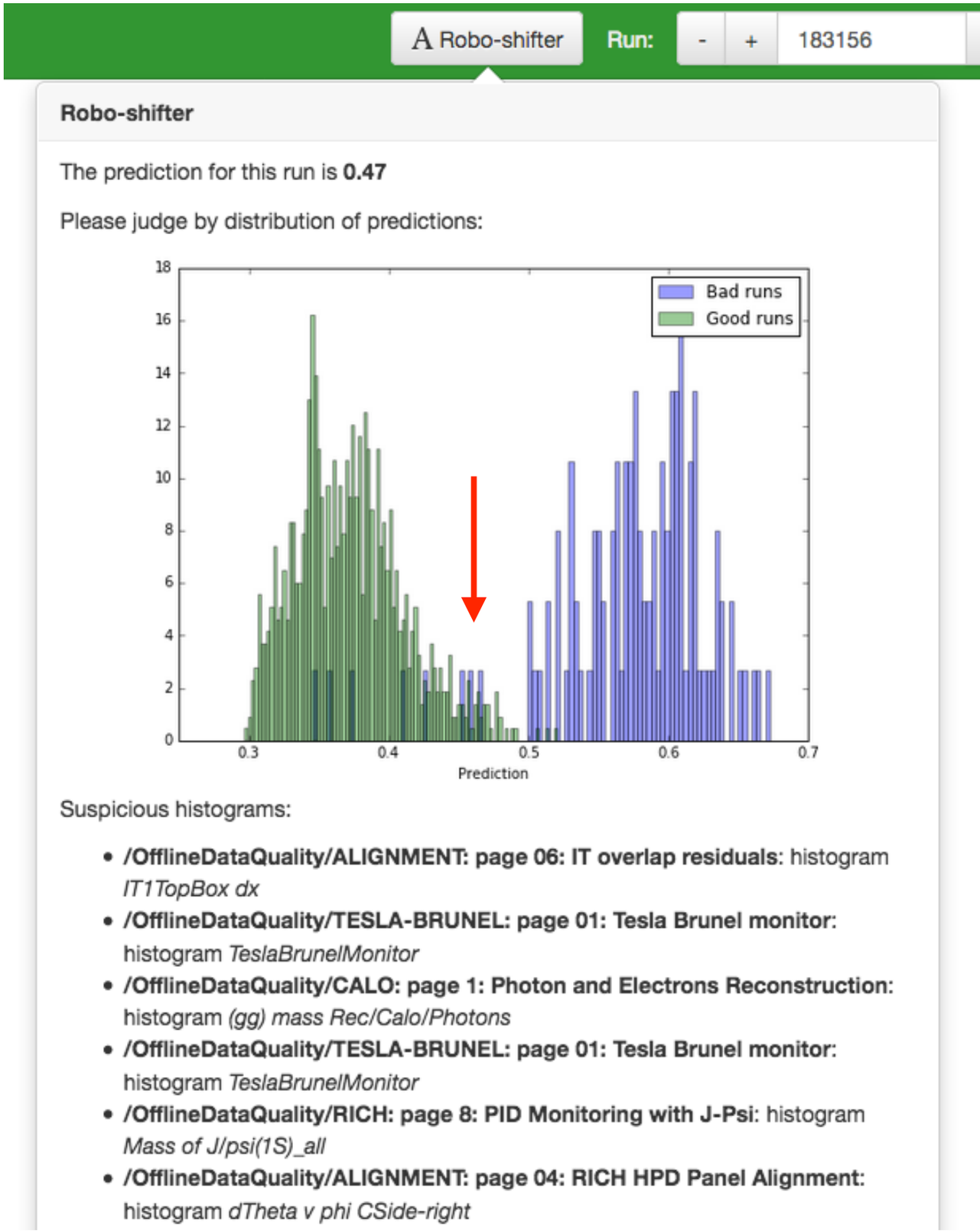
- Flatness is a very strong restriction, holding this restriction leads to quality decreasing.
- Still, flat model looks preferable comparing to baseline in ROC AUCs

(1-AUC)/(1-AUC$_{baseline}$) LHCb Simulation, preliminary

	Ghost	Electron	Muon	Pion	Kaon	Proton
baseline	1	1	1	1	1	1
p + p_T flatness	-23 %	-20 %	-27 %	-26 %	+2 %	+5 %
2d(p, p_T) flatness	-21 %	-9 %	-13 %	-23 %	+12 %	+23 %
p + p_T + η + nTracks flatness	-22 %	-13 %	-26 %	-24 %	+2 %	+6%
4d(p, p_T, η, nTracks) flatness	-21 %	-4 %	-13 %	-20 %	+10 %	+25 %

Monitoring Robo-shifter

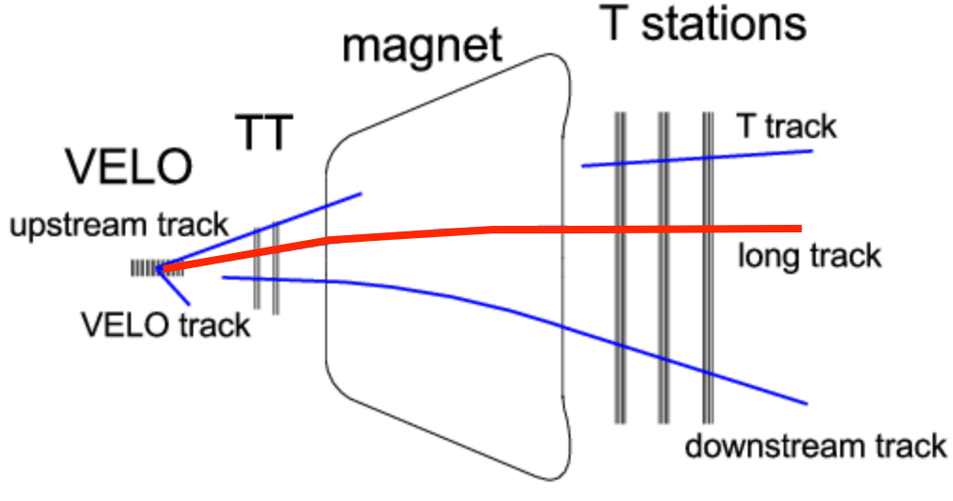
- Robo-shifter is machine-learning based system designed to assists the DQ shifter
- Given run data it can predict probability of run being good or bad
- Hint for potential problem sources is extracted from decision trees
- In process of comissioning for Data Quality Monitoring



Conclusions

- These are few highlights of successful or encouraging use of ML approaches in crucial points of the LHCb detector operation and data processing
- There are many other uses
  - boosting physics analyses
  - optimise data processing and data handling
  - neutral particles identification
  - …
- Different ML techniques are on the rise
  - LHCb follows this development and uses the best to ultimately improve our physics results

Backup: Forward Tracking



MC performance 2016 w.r.t. 2015	ν =1.6 w/ RL	w/o RL
timing HLT1	±0 %	
timing HLT2	+4 %	−38 %
fake rate	−27 %	−35 %
fake rate HLT1	−15 %	
ε long	+0.5 %	+0.1 %
ε long from B	+0.2 %	−0.2 %
$\varepsilon_{\text{HLT1}}$ long from B $p>3, p_T>0.5$ GeV	+0.1 %	