Flavour Tagging at LHCb

"Computing perspective"

Claire Prouve – on behalf of the LHCb Flavour Tagging group
Universidade da Coruña

MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES

UNIÓN EUROPEA
FONDO EUROPEO DE
DESARROLLO REGIONAL
"Una manera de hacer Europa"

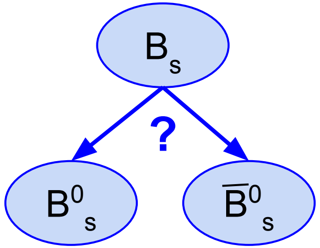
AGENCIA
ESTATAL DE
INVESTIGACIÓN

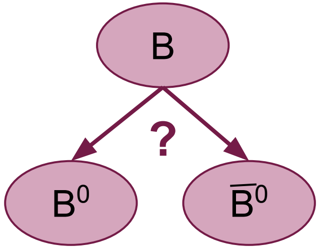
XUNTA
DE GALICIA

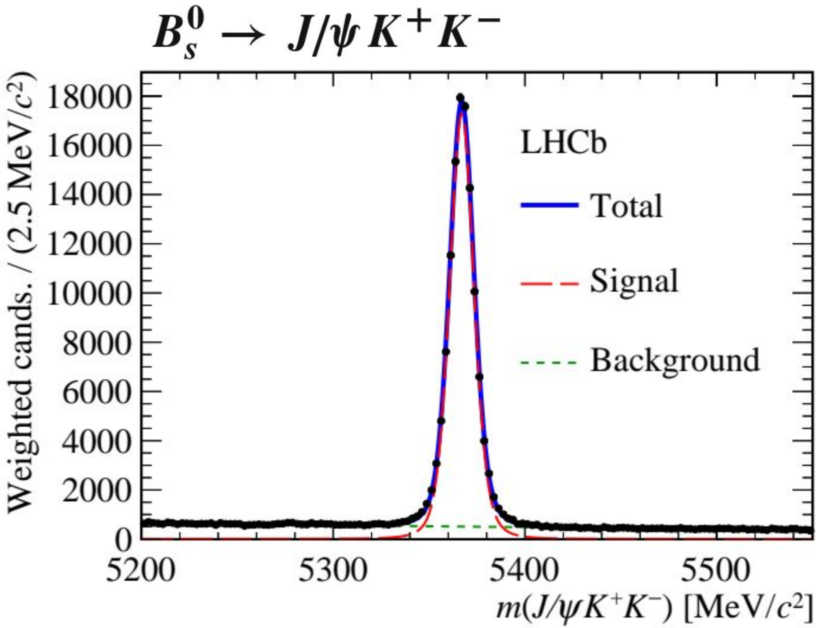
INTALENT
UDC - INDITEX

Flavour Tagging at LHCb

Determining the **flavour at production** of neutral B mesons





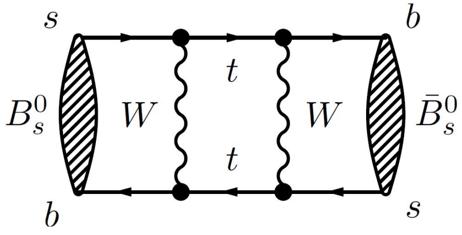


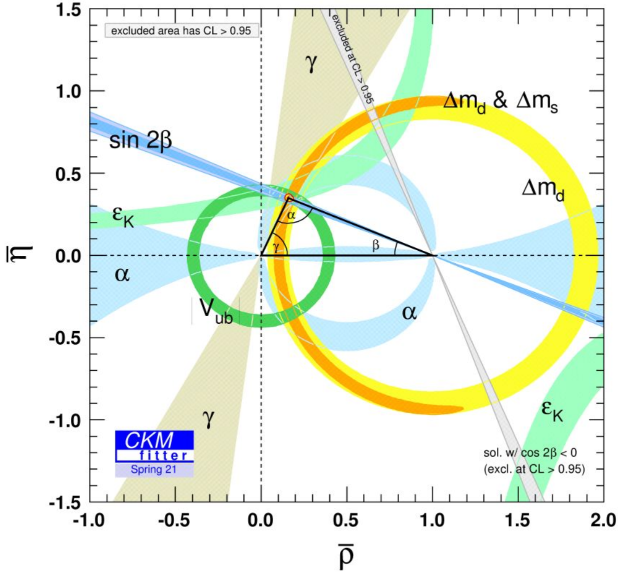
Eur. Phys. J. C 79 (2019) 706

Example: B_s meson oscillation

- Neutral *s,c,b* mesons can mix into their anti-particles
- Flavour eigenstates B^0_s and $\bar{B}^0_s$ are linear combinations of the mass eigenstates B_H and B_L
- Oscillation frequency $\mathbf{\Delta} m_s = m_H - m_L$ provides powerful constraint on CKM triangle

$$P(t) \sim e^{-\Gamma_s t}\left[\cosh\left(\frac{\Delta\Gamma_s t}{2}\right) + C \cdot \cos(\boxed{\Delta m_s} t)\right]$$

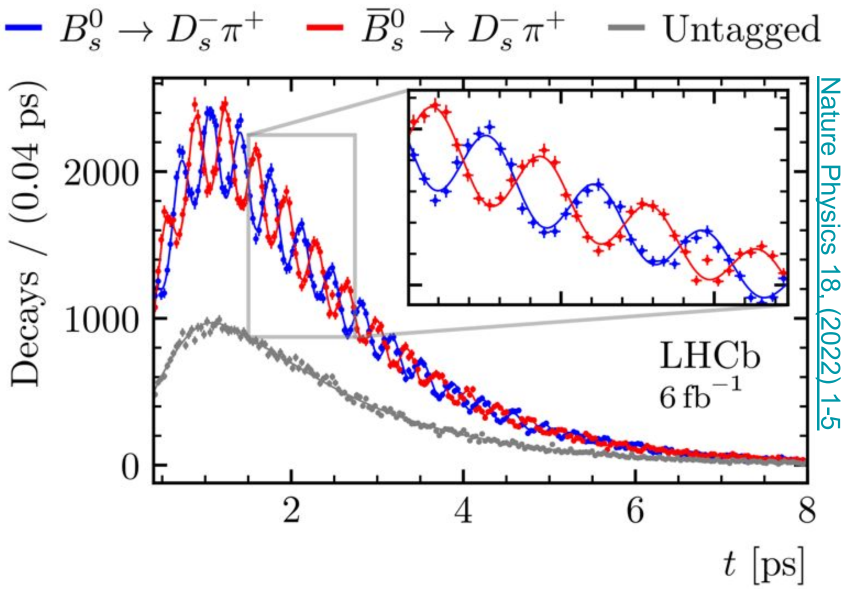




Example: B_s meson oscillation

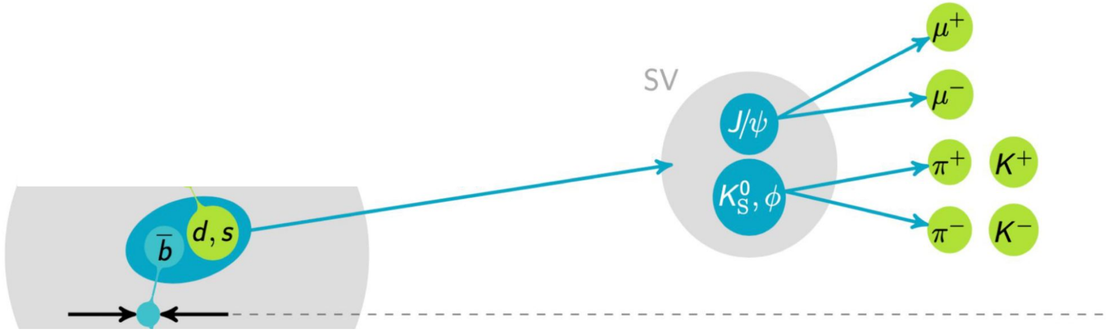
- Reconstruct the B_s in a flavour-specific final state
- Flavour-tag the B_s at production
- Measure and fit the decay time distributions

$$P(t) \sim e^{-\Gamma_s t}\left[\cosh\left(\frac{\Delta\Gamma_s t}{2}\right) + C \cdot \cos(\boxed{\Delta m_s} t)\right]$$



Nature Physics 18, (2022) 1-5

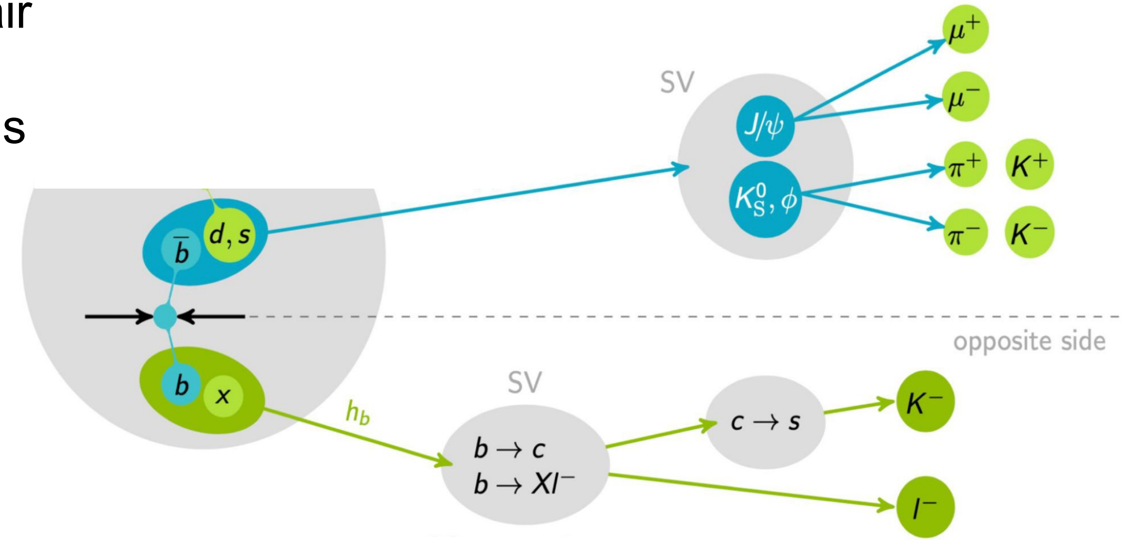
Flavour Tagging



Flavour Tagging

<u>Opposite side B decay:</u>

- 24% of events have that $b\bar{b}$ pair in the LHCb acceptance
- look for decay product that tags the other (opposite side) B meson



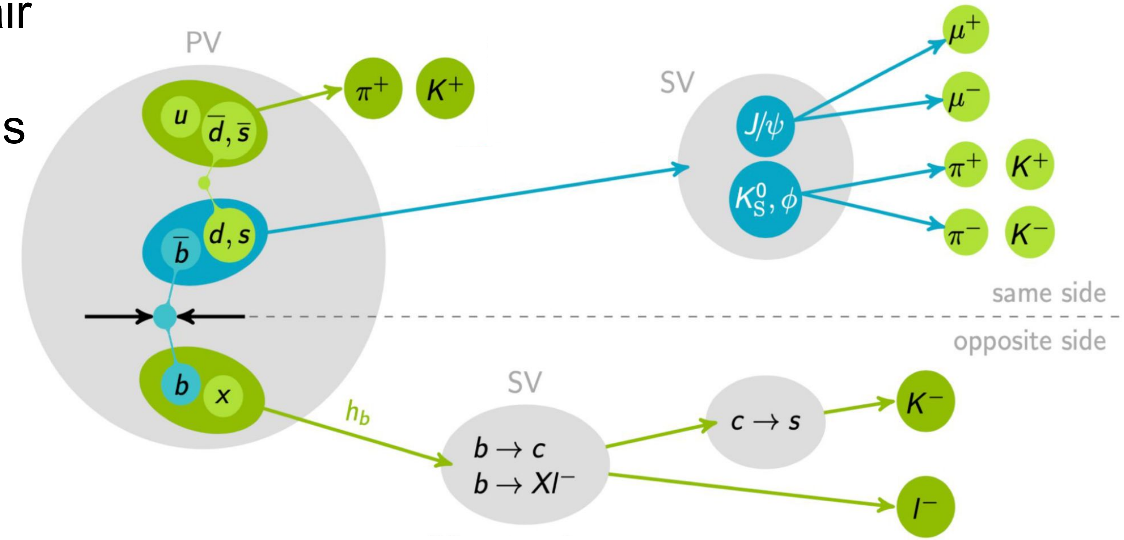
Flavour Tagging

Opposite side B decay:

- 24% of events have that $b\bar{b}$ pair in the LHCb acceptance
- look for decay product that tags the other (opposite side) B meson

Same side information:

- additional particles are created during the fragmentation process
- 50% of B are accompanied by a charged pion, 50% of B_s by a charged kaon

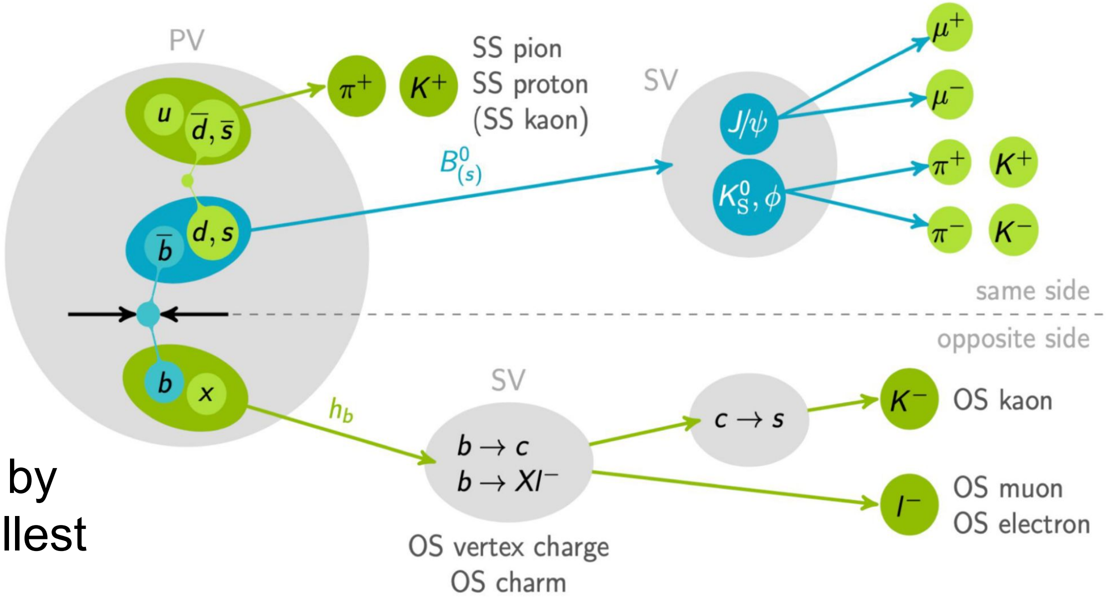


Flavour Taggers

Classical Taggers

- <u>one tagger per final state species</u>

  Same-side Pion
  Same-side Proton
  Same-side Kaon
  Opposite-side Kaon
  Opposite-side Muon
  Opposite-side Electron
  (Opposite-side Vertex Charge)



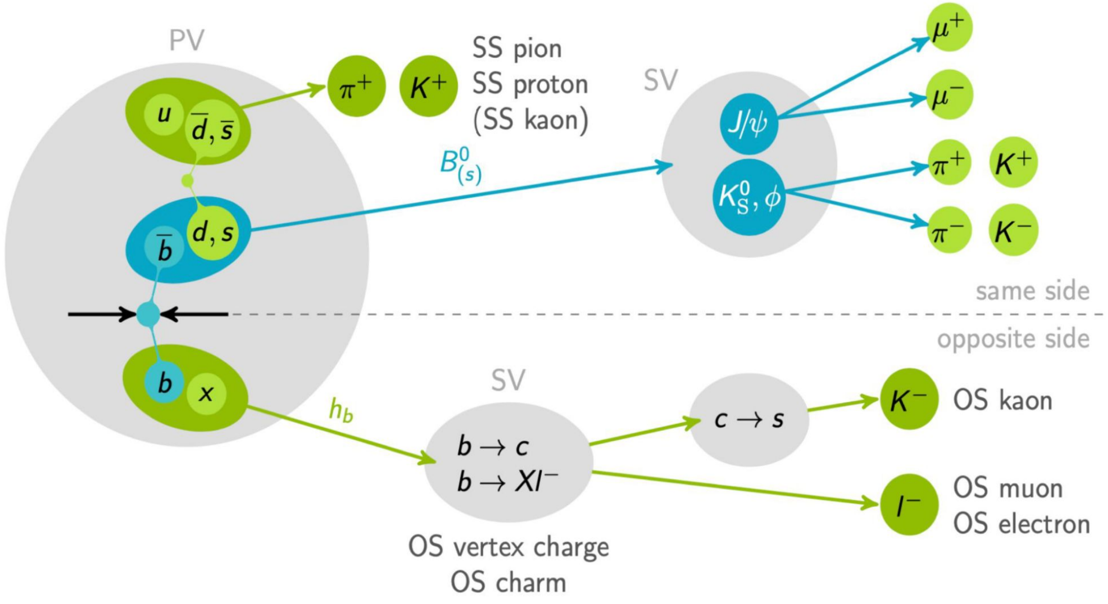
- combination of several taggers by picking the tagger with the smallest predicted mistag

Classical Taggers

- <u>One tagger per final state species:</u>

1. Selection to find the "tagging particle" (cut and/or MVA based)
2. MVA to evaluate mistag rate
3. Calibration

4. Combination of several taggers by picking the tagger with the smallest predicted mistag



Classical Taggers

1. **Selection to find the "tagging particle"**
   – start with all charged particles in the event
   – use rectangular cuts and/or MVA (BDT)

2. **MVA to evaluate "mistag probability"**
   – all tagging particles surviving the selection
   – B flavour predicted by tagging particle charge
   – MVA (BDT or simple NN) predict if the B flavour is right/wrong
   – tagging decision decided by tagging particle that has the "best" MVA response

3. **Calibration**
   – calibrate the output of the MVA to represent the "mistag probability"

(4. Combination decisions of individual taggers)

Performance metrics

Tagging efficiency ε: fraction of events that have a tagging decision (not necessarily that the tagging decision is correct)

Mistag rate/probability ω: fraction of events that have a wrong tagging decision / probability that a tagging decision is wrong

Tagging power $\varepsilon_{eff} = \varepsilon\,(1-2\cdot\omega)^2$: statistical precision of the sample

Run 2 examples of tagging power:

$B^0 \to J/\psi\, K^{*0}$: $\varepsilon_{eff} \sim 4.5\ \%$

$B_s \to D_s \pi^-$: $\varepsilon_{eff} \sim 6.5\ \%$

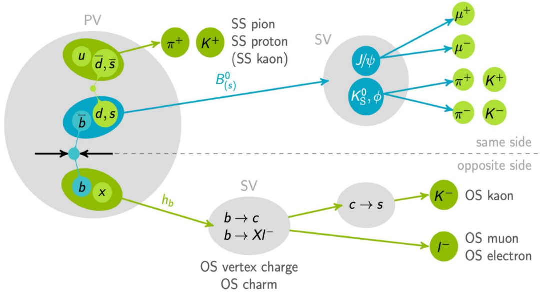
Training for Run 3 – Selection

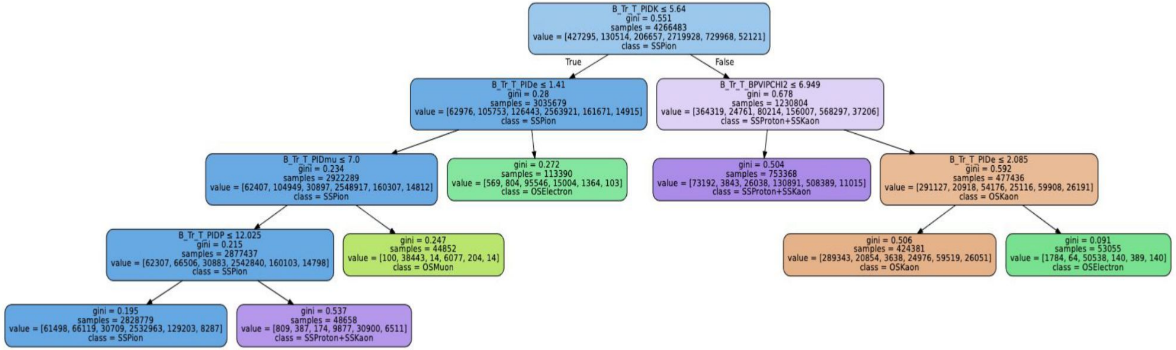
- lose selection (and let tagging-MVA do most of the job)
- simple, “reasonable”, comprehensible selection

- purity vs efficiency
- balanced vs unbalanced training (reconstruction efficiencies play a role)

Composition	Percentage
SSPion	64%
OSKaon	10%
SSKaon	10%
SSProton	7%
OSElectron	5%
OSMuon	3%
OSProton	1%

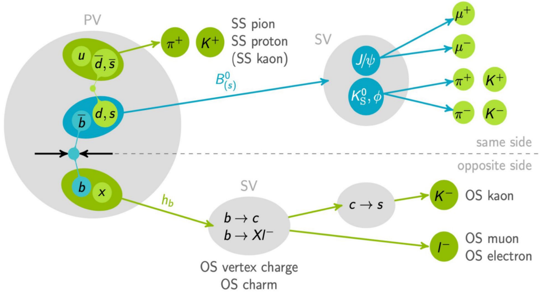
⇒ exclusive, shallow DT based classification of particles
⇒ among 41 input features → PIDs + IPχ^2 identified as the most powerful cutting features





Training for Run 3 – MVA

- all particles that get classified/selected for a given tagger are used as input to the MVA

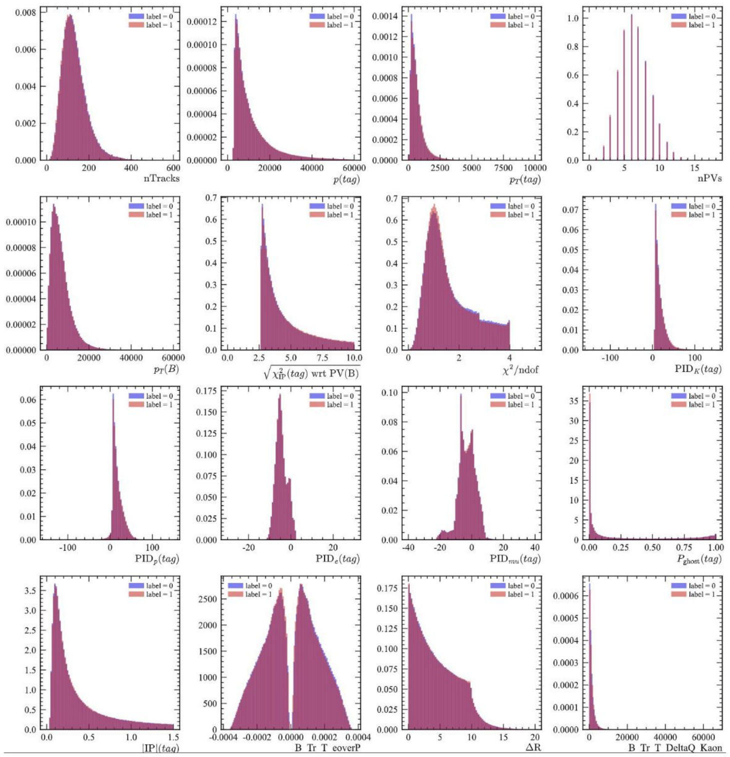


- charge correctly predicts B-flavour = signal (label=1)
- charge incorrectly predicts B-flavour = bkg (label=0)

- distributions for “signal” and “background” very similar
- AUC-ROC ~ 0.52

⇒ Flavour tagging is a difficult task!

An example: OSKaon

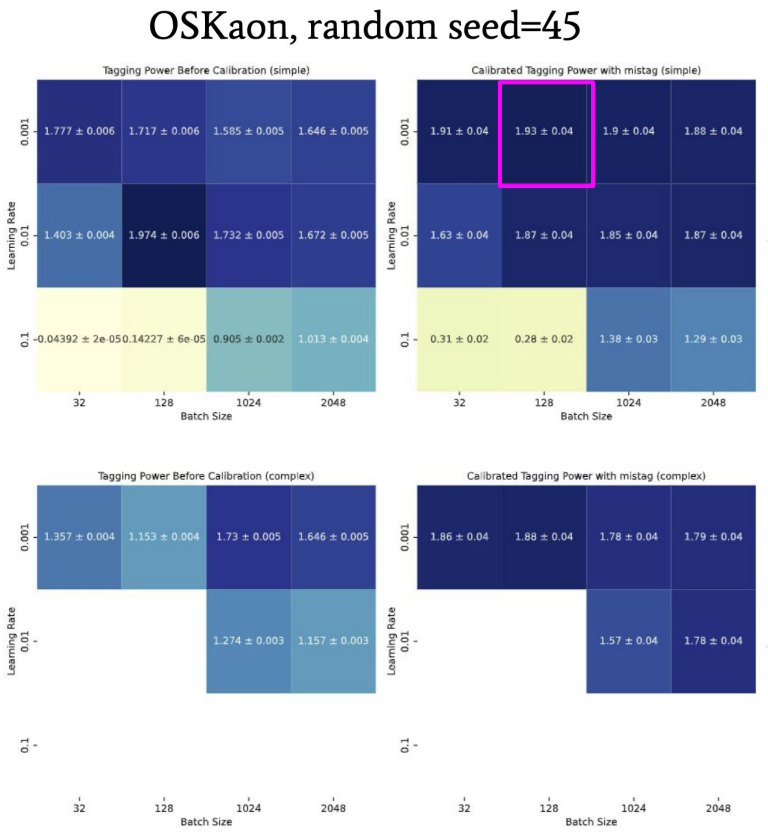


Training for Run 3 – MVA

- maximise calibrated tagging power
- fast in training and application

<u>NN architectures:</u>
- simple: one hidden layer with 3 nodes
- complex: two hidden layers with 32 and 64 nodes
- learning rate: 0.001, 0.01, 0.1
- training batch size: 2024, 1024, 128, 32

⇒ Optimisation still ongoing!



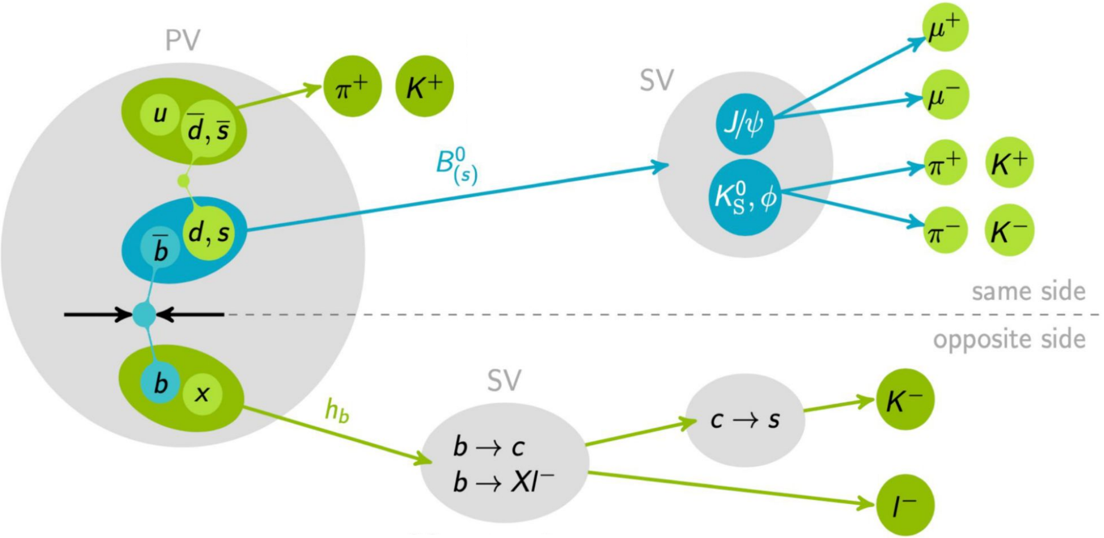
Inclusive Tagger

[see John Wendel's talk tomorrow]

- can consider "all" tracks in the event

Requirements:

- variable number of inputs (number of tracks varies by event)
- permutation invariant
- fast to train and apply

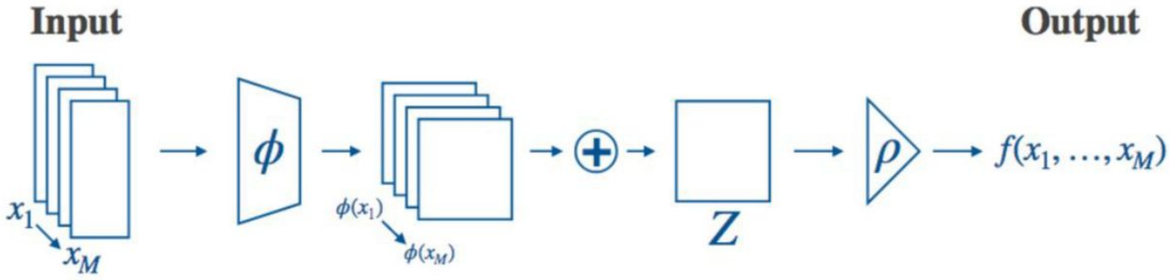


DeepSet Neural Network

see John Wendel's talk tomorrow

- "every instance (particle) is transformed (possibly by several layers) into some representation ϕ"
- "the representations ϕ are added up and the output is processed using the ρ network"

→ permutation invariant, component per particle, component for entire event
→ ~ hour to train (previous inclusive tagger took ~days)
→ ~ 7μs per event to evaluate (previous inclusive tagger took ~5h for 3M events)
~ 20 - 35% increase in tagging power wrt classical taggers

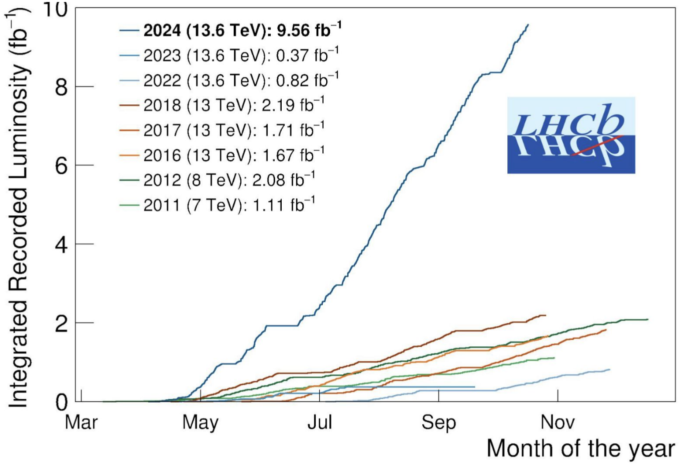


arXiv:1703.06114 [cs.LG]

Computing resources

LHCb Run3 data taking

- increased instantaneous luminosity μ ~ 1.4 → μ ~ 5.3
- removal of hardware trigger



Real Data in bookkeeping as of 05/11	**Run1 + Run2**	**Run3** *so far…*
Tape	72 PB	~ 118 PB
Disk	16 PB	~ 41 PB

HLT2 was producing 0.6 GB/s in Run2, we are now at ~10 GB/s in Run3

In a 10 hour fill of p-p, the HLT2 produces 360 TB that are saved to tape (twice). After sprucing, 126 TB go to disk (with 2 instances, that's ¼ of a PB)

" Our upgrade brings us data volumes similar to what ATLAS and CMS currently record…"
⇒ Disk and tape space are becoming a limiting factor.

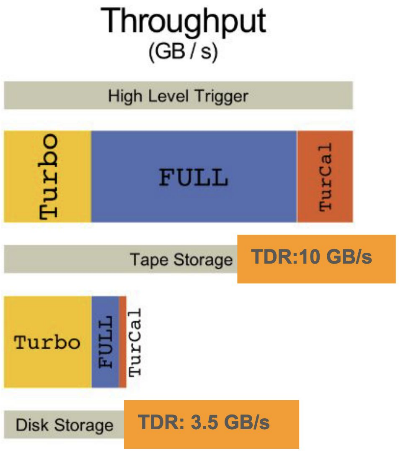
LHCb Run3 data taking

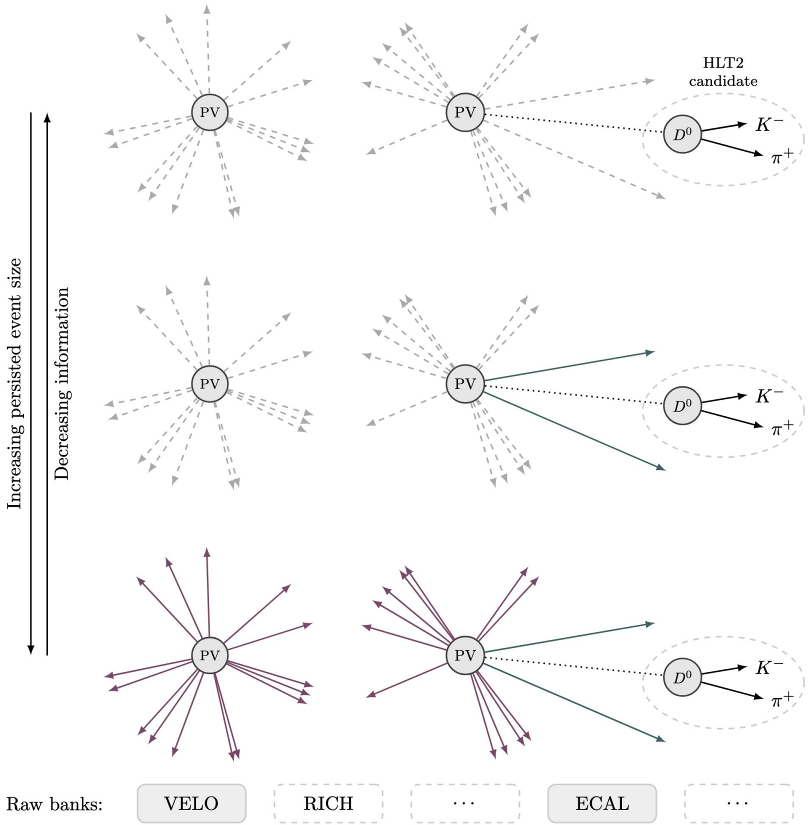
- increased instantaneous luminosity $\mu \sim 1.4 \rightarrow \mu \sim 5.3$
- removal of hardware trigger

⇒ full event reconstruction in the trigger

<u>Turbo line model:</u>
2500 turbo lines
400 full lines

Flavour Tagging requires a lot more information besides the signal candidate!
~150 ft lines in turbo



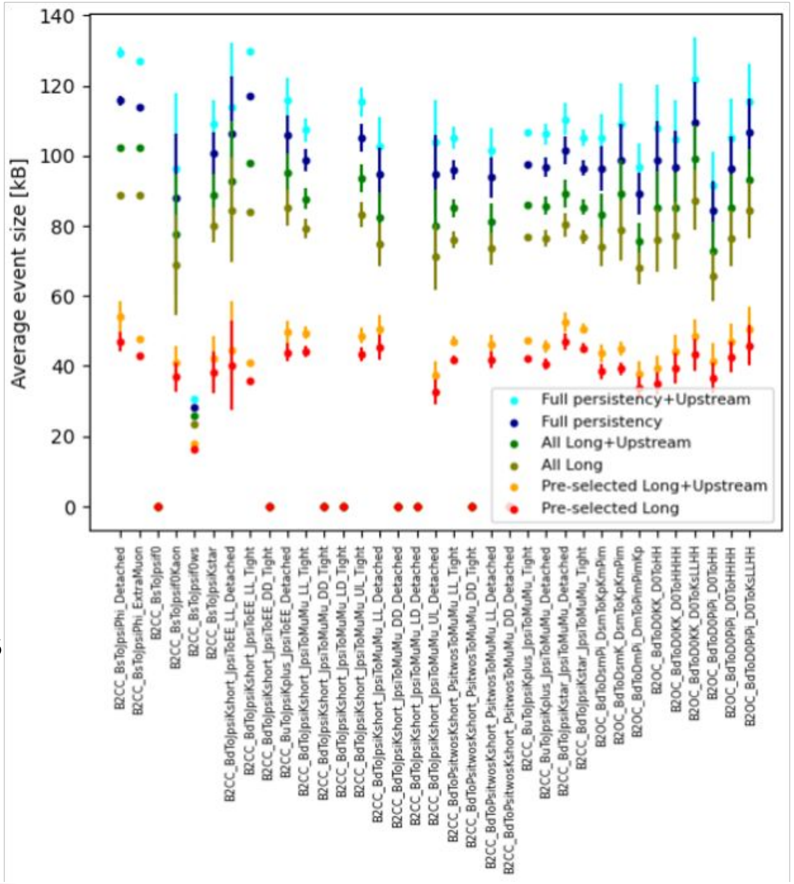


Flavour Tagging Bandwidth

How does saving different amount of information impact the event size? (Study on MC)

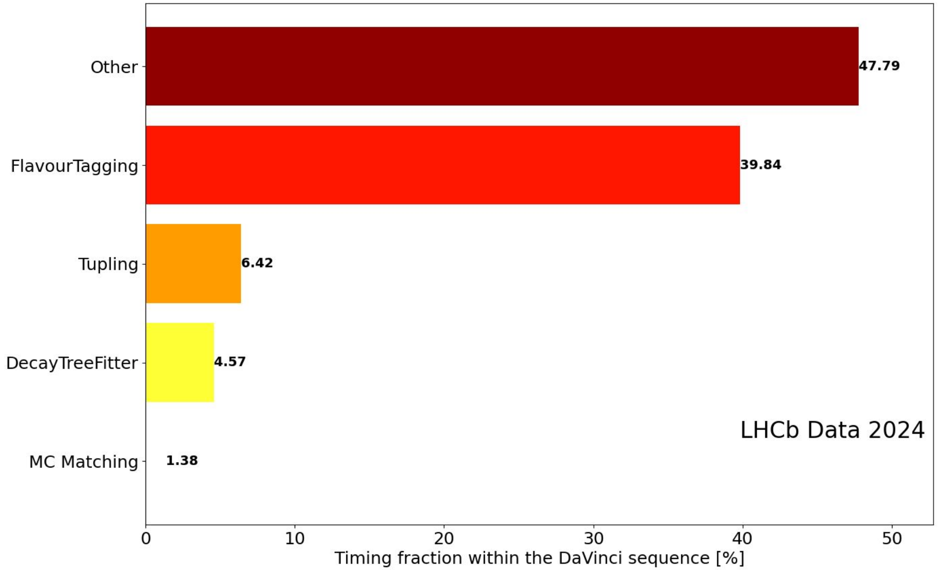
- pure turbo: ~3 kB
- turbo + all long tracks: ~80kB
- turbo + all long tracks + upstream tracks: ~90 kB
- “full” persistency: ~105 kB
- “full” persistency + all upstream tracks: ~115 kB

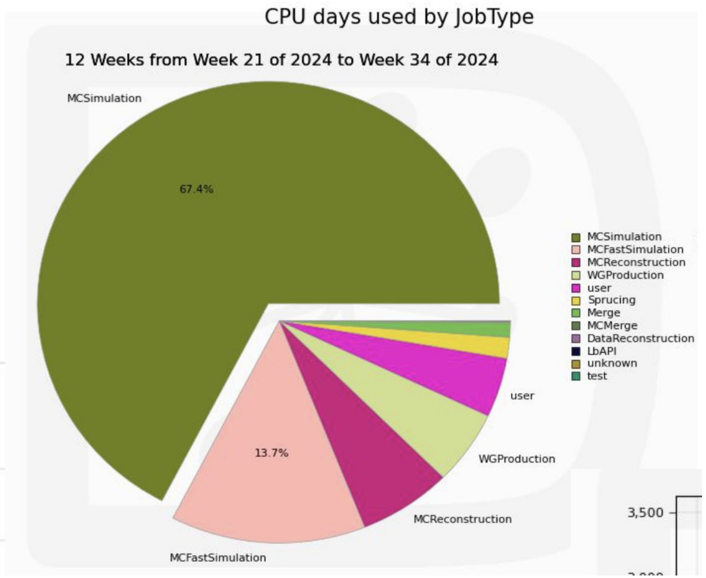
Due to the relatively small number of flavour tagging lines + and their small rates probably not a big problem, but worth studying and keeping an eye on!
⇒ What is the physics impact of different selections?



Offline CPU resources

All data taken needs to be processed offline to be available to analysts





~ 87% of CPU used for simulation ⇒ Flavour Tagging should not be a problem (but I found the timing test interesting anyway 😅)

Software development

Flavour Tagging in the software

All functionality we want to use needs to be implemented in the LHCb software!

- tagging algorithms: preselection, MVA, configuration, helper functionality such as PV unbiasing, distance calculation, tracking configuration etc
- persistency: configuration of what is saved in the trigger, functionality of saving and reading what we want to save
- tupling: tagging decisions + everything needed to train new taggers has to be written to usable format → ntuple
- tests: currently have tests in 3 different projects that need to be kept up to date

For the Run 3 software, we made ~50 MRs, between ~10 people and ~6000 lines of code. ⇒ It's a lot of ongoing work!

Summary & Conclusion

- Flavour tagging at LHCb is essential for meson mixing measurement and time-dependent CP violation measurements.

- Tagging Algorithms use information from particles created in correlation with the signal particles. Classical taggers and inclusive tagger are in development for Run 3, taking care that all parts are well understood and robust.

- Increased data volumes in Run 3 (and beyond) requires us to keep resources in mind.

- Many different parts of software are involved in the flavour tagging. Software development and maintenance always ongoing!