technische universität
dortmund

Master Defense on
Optimization of Opposite-Side Flavor-Tagging Algorithms for the LHCb Upgrade

Thomas-Christopher Ogasa

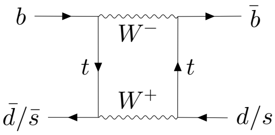
Supervisors: Dr. Quentin Führing and Dr. Vukan Jevtic

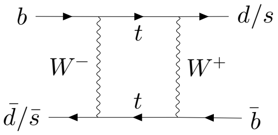
TU Dortmund University
Working group Albrecht

29.09.2025

Flavor Tagging: Goal and Motivation

- LHCb physics program includes time-dependent studies of B^0 and B_s^0
  - Including meson oscillation and some $C\!P$-violation studies
  - Requires production flavor
  - Not ascertainable from decay products
- Goal of Flavor Tagging: Reconstruct the production flavor of B^0 and B_s^0
  - performance directly impacts statistical uncertainty of the measurements
  - Algorithms from Run 2 available, for Run 3 in development [1]





Flavor Tagging Principle

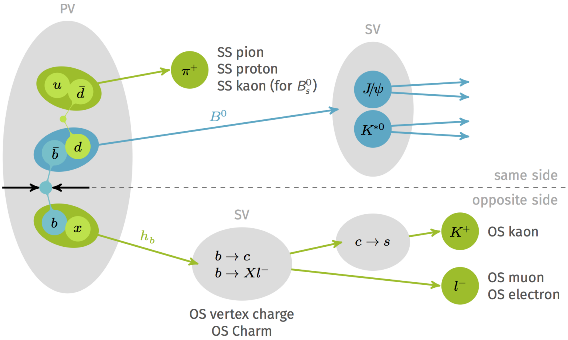


Figure: Schematic representation of the strategies used for the Flavor-Tagging algorithms available at LHCb [2].

Algorithm performance

- Performance of FT algorithms depend on two values
  - Tagging efficiency : $\epsilon_{\mathrm{tag}} = \frac{N_{\mathrm{W}}+N_{\mathrm{R}}}{N_{\mathrm{W}}+N_{\mathrm{R}}+N_{\mathrm{U}}}$
  - Mistag probability : $\omega = \frac{N_{\mathrm{W}}}{N_{\mathrm{W}}+N_{\mathrm{R}}}$
- Combine into tagging power: $\epsilon_{\mathrm{tag,eff}} = \epsilon_{\mathrm{tag}}(1-2\omega^2)$
  - Fraction of events with accurate tagging decision
  - $\sigma_{\mathrm{stat}} \propto \frac{1}{\sqrt{N\epsilon_{\mathrm{eff}}}}$
- Mistag probabilities of each FT algorithm for an event can be combined

  → One prediction per event, increased combined tagging power

Data

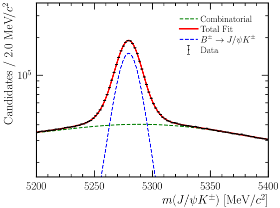
Simulation:

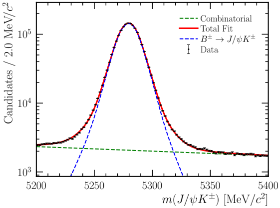
- Simulated 2024 data with UT
- $B^{\pm} \to J/\psi K^{\pm}$
- Yield: $\sim 3.41 \cdot 10^6$ Events

Data:

- 2024 Data
- BDTs for background rejection
  - $\to$ BDT features from Run 2 $\sin(2\beta)$ analysis [3, 4]
- $B^{\pm} \to J/\psi K^{\pm}$

	SIG Yield $[10^6]$	BKG Yield $[10^6]$
Pre BDT	1.95 ± 0.02	3.71 ± 0.02
Post BDT	1.855 ± 0.002	0.201 ± 0.002





LHCb Upgrade I Detector

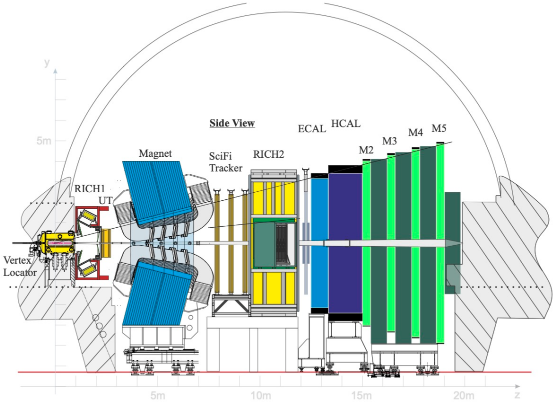


Figure: Schematic representation of the LHCb Upgrade I detector [5].

Strategy

1. Each tagger selects a track matching expectations of its specific process
   $\rightarrow$ Done by decision tree

---

2. MVA classifiers, gauging the probability of the tagging decision being wrong
   - Neural network trained on MC or Data
   - Requires a calibration for accurate mistag probabilities
3. Combine OS algorithms
4. Evaluate all Tagging performances on Data and compare

Track Selection

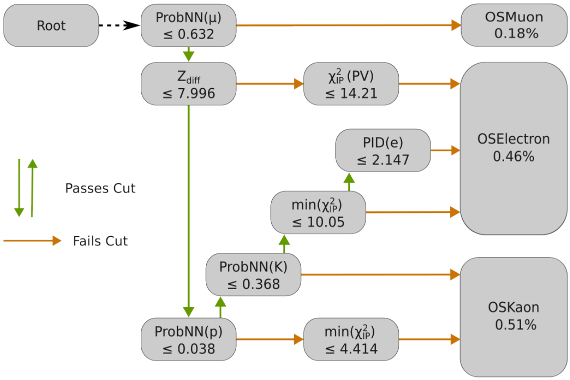


Figure: Relevant parts of the decision tree, trained previously [1], to categorize tracks by tagger assignment or exclusion.

MVA Classifier Training

- Neural networks
  - Previously trained on simulation
  - In this study trained simulation, data, and mixed approach
- 17 features
  - Particle ID, kinematic, etc.
- Hyperparameter space broader and more flexible than in previous studies
  - Optimized in grid search
- For data: Each sample weighted to further reduce background contribution

Performance on Data

Trained on	Tagging power [%] on $B^{\pm} \to J/\psi K^{\pm}$ data OSKaon	OSMuon	OSElectron	OSCombined
Simulation [6] (reference)	1.09 ± 0.02	0.88 ± 0.02	0.37 ± 0.01	2.12 ± 0.03
Simulation	1.36 ± 0.03	0.76 ± 0.02	0.37 ± 0.01	2.29 ± 0.03
Data	1.80 ± 0.03	0.81 ± 0.02	0.43 ± 0.02	2.66 ± 0.03

- Improvement in OSKaon from architecture changes
- Training on data improves OSKaon and OSElectron
- Performance decreases in OSMuon
- OSCombined increased $(25 \pm 2)\,\%$

Performance on Simulation

Trained on	Tagging power [%] on $B^{\pm} \to J/\psi K^{\pm}$ simulation		
	OSKaon	OSMuon	OSElectron
Data	2.48 ± 0.06	0.87 ± 0.04	0.57 ± 0.03
Simulation	2.55 ± 0.06	0.85 ± 0.04	0.57 ± 0.03

- Performance differences are smaller or vanish
- Simulation-trained models seem to generalize worse
  - $\to$ minimize impact of simulation-data-differences

Domain Adaptation by Back-Propagation

- Unsupervised learning method
- Split the model into feature extractor and label predictor
- Add domain classifier with gradient reversal layer

→ Feature extractor extracts domain-agnostic but predictive features

→ Adds new hyperparameter λ balancing both objectives

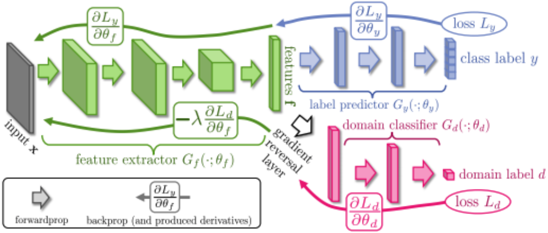


Figure: Scheme of Domain Adaptation with a gradient reversal layer [7]

Domain Adaptated Models

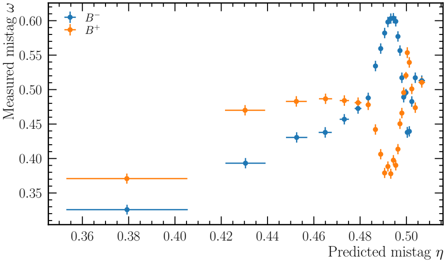
	Metrics [%] of Domain Adapted models			
	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$
ϵ_{eff} Data	1.32 ± 0.03	1.38 ± 0.03	1.27 ± 0.03	1.22 ± 0.03
ϵ_{eff} Simulation	2.73 ± 0.07	2.75 ± 0.07	2.71 ± 0.07	2.63 ± 0.07
Domain accuracy	66.5	50.5	50.6	50.4

$\rightarrow$ Domain accuracy decreases with introduction of $\lambda \neq 0$

$\rightarrow$ Domain adaptation can increase performance

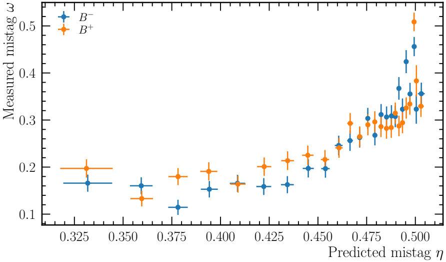
Flavor Tagging Asymmetry

- Asymmetry seen across architectures, in simulation and data, mainly in OSElectron and OSMuon
- Deviations seem to counteract each other
- First seen, currently cause not definitively known
- Reason to believe that many Run 3 FT algorithms are affected
- Current hypothesis: different OS tracks are not filtered correctly, example: leptons from $c \to s + l^+$



Flavor Tagging Asymmetry

- Asymmetry seen across architectures, in simulation and data, mainly in OSElectron and OSMuon
- Deviations seem to counteract each other
- First seen, currently cause not definitively known
- Reason to believe that many Run 3 FT algorithms are affected
- Current hypothesis: different OS tracks are not filtered correctly, example: leptons from $c \to s + l^+$



Summary

- Improvement of combined OS performance by $(25 \pm 2)\,\%$
  - Architecture improvements
  - Training directly on Data
  - Mainly OSKaon improved
- Best OSKaon algorithm trained without labeled data, achieved using domain adaptation
  - May allow improvements in SS algorithms
- Asymmetry-structures found in several algorithms
  - May hold back OSElectron and OSMuon performance
  - Hypothesis about the cause of the asymmetry was postulated

Outlook

- Migrate taggers to new Decision Tree
- Train SS taggers
  - Possibly using domain adaptation with hyperparameter optimization
- Combine taggers
- Investigate observed asymmetries
  - May be addressed by new Decision Tree
  - Otherwise more sophisticated/additional selection
- Possibly: Implement similar tagging algorithms into LHCb software

Bibliography I

[1] S. Celani, Q. Führing, and M. Olocco. "Single-track flavour-tagging algorithms for Run3". Work in Progress LHCb internal Analysis note. 2025.

[2] J. Wishahi. *Flavour Tagging Plots for Conference*. 2016. URL: https://twiki.cern.ch/twiki/bin/view/LHCb/FlavourTaggingConferencePlots.

[3] V. Jevtić. "Measurements of the CKM parameter $\sin(2\beta)$ in $B^0 \to J/\psi K^0_S$ decays with the LHCb experiment". PhD thesis. Technische Universität Dortmund, 2024. DOI: http://dx.doi.org/10.17877/DE290R-24785.

[4] V. Jevtić et al. "Measurement of CP violation in $B^0 \to J/\psi K^0_S$ decays". LHCb internal Analysis note. 2023.

Bibliography II

[5] LHCb Collaboration. "The LHCb Detector at the LHC". In: *JINST* 3.08 (Aug. 2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[6] M. Olocco. Private communication with the main developer of the Run 3 Flavor-Tagging algorithms. 2025.

[7] Y. Ganin, and V. Lempitsky. "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 1180–1189.

Backup

BDT Features

Target	Observable	Description
B^+	$\chi^2_{\mathrm{Vtx}}/n_{\mathrm{dof}}$	Vertex reconstruction quality
	$\chi^2_{\mathrm{IP,PV}}$	Reconstruction quality of impact parameter, with respect to the primary vertex
	η	Pseudorapidity of B^+
	χ^2_{DTF}	Quality of the decay tree fit of B^+ with constrained J/ψ mass and primary vertex
J/ψ	$\mathrm{IP}_{\mathrm{PV}}$	Impact parameter of J/ψ with respect to the primary vertex of B^+
$\mu^\pm$	$\mathrm{IP}_{\mathrm{PV}}$	Impact parameter of $\mu^\pm$ with respect to the primary vertex of B^+, where $\mu^\pm$ are the reconstructed decay products of the J/ψ
K^+	$\mathrm{IP}_{\mathrm{PV}}$	Impact parameter of K^+ with respect to the primary vertex of B^+
	η	Pseudorapidity of K^+
	$\min(\mathrm{IP})$	minimum reconstructed impact parameter of K^+

Full Decision Tree

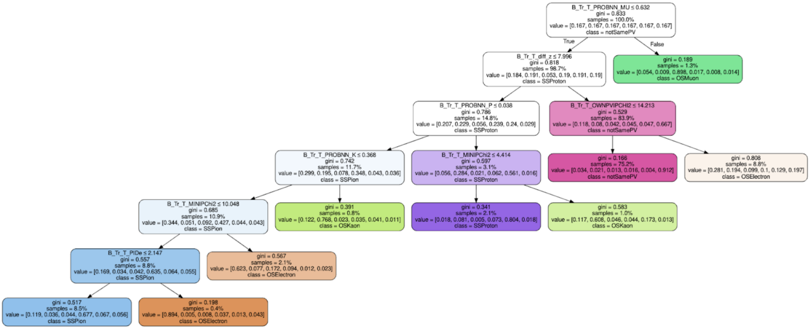


Figure: Decision tree, trained previously, to categorize tracks by tagger assignment or exclusion [1].

NN Features

Features	Description
n_{tracks}	Number of tracks in the event
n_{PVs}	Number of primary vertices in the event
$p_T(B)$	Transverse momentum of the b-meson
p	Momentum of the track particle
p_T	Transverse momentum of the track particle
χ^2/n_{dof}	Quality of track reconstruction
$P_{\text{NN}}(K)$	Predicted probability of the track to be of a K
$P_{\text{NN}}(\pi)$	Predicted probability of the track to be of a π
$P_{\text{NN}}(p)$	Predicted probability of the track to be of a p
$P_{\text{NN}}(\mu)$	Predicted probability of the track to be of a μ
$P_{\text{NN}}(e)$	Predicted probability of the track to be of a e
GhostProb	Predicted probability of the track to be a ghost track
IP	Impact parameter of track with respect to primary vertex of B
$\text{IP}/\sigma_{\text{IP}}$	Significance of impact parameter
E/p	Energy divided by momentum of the track particle
ΔR	Squared sum of $\Delta\phi^2$ and $\Delta\eta^2$
ΔQ_X	Amount of change the invariant Mass experiences if the track was added. Defined for $X \in \{K, \mu, e\}$ as $\Delta Q_X = \sqrt{(E_X + E_B)^2 - \lvert\vec{p}_X + \vec{p}_B\rvert^2} - M_B - M_X$

Previous Tagging Algorithms

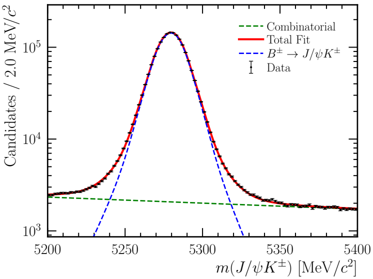
- Neural Networks Trained on MC
- Hyperparameters:
  - learning rate
    $\{0.001, 0.01, 0.1\}$
  - Batch size
    $\{32, 128, 1024, 2048\}$
  - Architecture
    {'Simple', 'Complex'}
  - minimum improvement $\Delta_{\min}$
    $\{0.0, 0.0001, 0.001, 0.01\}$

Simple	
Hiddenlayers	$3, 3$
Dropout	0
Activation	Elu, Sigmoid

Complex	
Hiddenlayers	$32, 64, 32$
Dropout	0.5
Activation	Elu, Sigmoid

New Tagging Algorithms

- Same features
- Hyperparameters:
  - learning rate $\{10^{-4}, 10^{-3}, 10^{-2}\}$
  - Batch size $\{2048, 4096, 8192\}$
  - Number of Layers $\{2, 4, 6, 8, 16\}$
  - Number of Neurons $\{8, 16, 32, 64, 128, 256\}$
- For Data: Each sample weighted by Ratio of PDFs



FT Combination

$$p_b = \prod_i \left(\frac{1 + d_i}{2} - d_i(1 - \eta_i) \right)$$
$$p_{\overline{b}} = \prod_i \left(\frac{1 - d_i}{2} - d_i(1 - \eta_i) \right) . \quad (1)$$

$$P_b(p_b, p_{\overline{b}}) = \frac{p_b}{p_b + p_{\overline{b}}}, \quad (2)$$

$$P_{\overline{b}} = 1 - P_b. \quad (3)$$

$$d_{\mathrm{comb}} = \mathrm{sign}(P_b - P_{\overline{b}})$$
$$\eta_{\mathrm{comb}} = 1 - \max(P_b, P_{\overline{b}}). \quad (4)$$

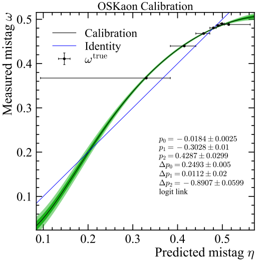
Calibration

- Second order Polynomial
- B_{ki} matrix of parameters defining the basis with minimum correlation
- Δp_k allow for asymmetry

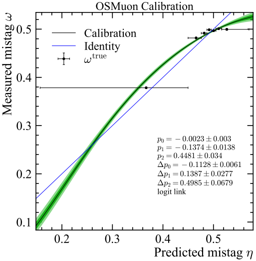
$$P_k(\eta) = \sum_{i=0}^{2} B_{ki} g^{-1}(\eta)^k. \tag{5}$$

$$\begin{aligned} \omega^{B}(\eta) &= g\left(g^{-1}(\eta) + \sum_{k=0}^{2}\left(p_k + \frac{\Delta p_k}{2}\right)P_k(\eta)\right) \\ \omega^{\overline{B}}(\eta) &= g\left(g^{-1}(\eta) + \sum_{k=0}^{2}\left(p_k - \frac{\Delta p_k}{2}\right)P_k(\eta)\right). \end{aligned} \tag{6}$$

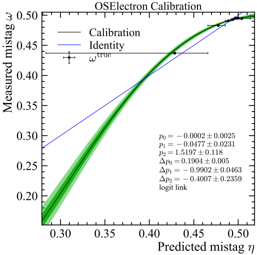
Simulation trained - Data calibrated



(a): OSKaon trained on simulation.

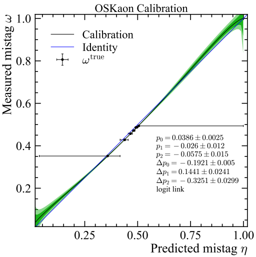


(b): OSMuon trained on simulation.

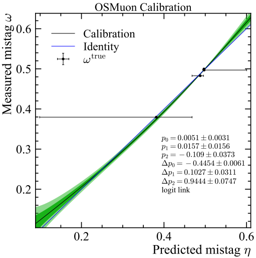


(c): OSElectron trained on simulation.

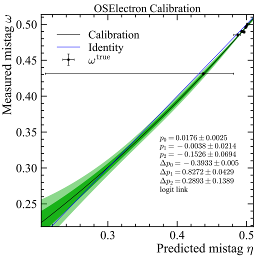
Data trained - Data calibrated



(a): OSKaon trained on simulation.

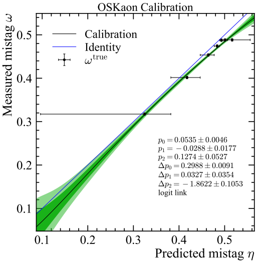


(b): OSMuon trained on simulation.

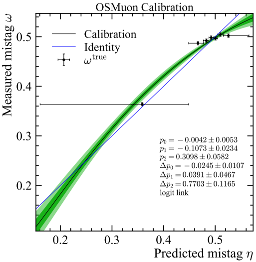


(c): OSElectron trained on simulation.

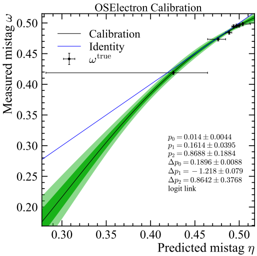
Simulation trained - Simulation calibrated



(a): OSKaon trained on simulation.

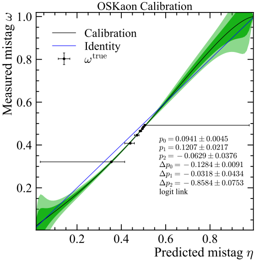


(b): OSMuon trained on simulation.

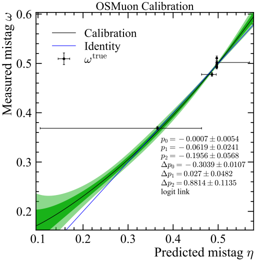


(c): OSElectron trained on simulation.

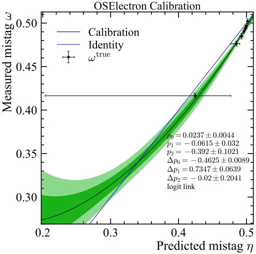
Data trained - Simulation calibrated



(a): OSKaon trained on simulation.



(b): OSMuon trained on simulation.



(c): OSElectron trained on simulation.

Domain Adaptation

- Field of study in machine learning dealing with distinct data domains
  - Source domain: Labeled
  - Target domain: Unlabeled, inference of this data is the goal
- Goal: Train on source data, such that it generalizes to target data