LHCb-PUB-2017-011
August 4, 2017

Fast neural-net based fake track rejection in the LHCb reconstruction

Michel De Cian[1], Stephen Farry[2], Paul Seyfert[3,*], Sascha Stahl[4].

[1] *Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany*
[2] *Oliver Lodge Laboratory, University of Liverpool, Liverpool, United Kingdom*
[3] *Sezione INFN di Milano Bicocca, Milano, Italy*
[4] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*
[*] *now at* [4]

Abstract

A neural-network based algorithm to identify fake tracks in the LHCb pattern recognition is presented. This algorithm, called ghost probability, retains more than 99 % of well reconstructed tracks while reducing the number of fake tracks by 60 %. It is fast enough to fit into the CPU time budget of the software trigger farm and thus reduces the combinatorics of the decay reconstructions, as well as the number of tracks that need to be processed by the particle identification algorithms. As a result, it strongly contributes to the achievement of having the same reconstruction online and offline in the LHCb experiment in Run II of the LHC.

© CERN on behalf of the LHCb collaboration, licence CC-BY-4.0.

1 Introduction

The LHCb detector consists of subsystems designed to perform high efficiency track reconstruction ($> 95\,\%$) [1] with an excellent momentum resolution ($0.5\,\%$ for $p < 20\,\mathrm{GeV}/c$) [2]. Two Ring Imaging Cherenkov detectors provide precise particle identification. In Run II of the LHC, a new scheme for the LHCb software trigger allows splitting the triggering of the event into two stages, giving room to perform the alignment and calibration in real time. In the novel detector alignment and calibration strategy for Run II, data collected at the start of the fill are processed in a few minutes and used to update the alignment, while the calibration constants are evaluated for each run [3]. This allows identical constants to be used in the online and offline reconstruction.

One of the challenges to run the full offline reconstruction in the software trigger is the limited CPU time budget of the computing farm. The reconstruction time of events depends strongly on the number of reconstructed charged particle tracks in an event in two ways. Firstly, the particle identification (PID) is evaluated for every reconstructed long track[1] in the second stage of the software trigger. Secondly, there are more possible track combinations to consider in the reconstruction of decay vertices with more reconstructed tracks.

A key ingredient to fit the offline reconstruction into the software trigger is the reduction of the fake track rate prior to the PID and decay reconstruction in the second software trigger stage. Fake tracks are defined as those reconstructed tracks which do not correspond to the trajectory of a true particle but instead are due to the mismatch of hits from separate particles or from detector noise. A neural network, described in this note, is deployed to identify these fake tracks, called the "ghost probability"[2].

Terminology

To avoid ambiguity, the bare term "performance" is avoided. Instead, when referring to how well good tracks are separated from fake tracks, the term "physics performance" is used since it is the figure of merit on which physics analyses depend. The term "CPU performance" is used for the amount of computing resources needed to execute the algorithm described in this note. As benchmark for the latter, the cycle count of callgrind [4] is used. Effects of instruction caching and data caching are assumed to be small, approximately confirmed by wall clock time measurements. The cycles spent in other algorithms which are *only* called to compute input quantities to the ghost probability are assigned to the ghost probability, most notably this includes the interpolation of tracks through active detector material to determine which channels should have a hit from the track – other algorithms which compute input quantities, which would be executed anyways, like the track fit, are not assigned to the ghost probability.

The term "ghost probability" is used for both the entire algorithm computing whether a track is considered a fake track or a real track, including the neural network, and for

[1] Full PID for downstream tracks was gradually introduced in the software trigger.

[2] The name is due to the fact that fake tracks are commonly called "ghosts" in LHCb.

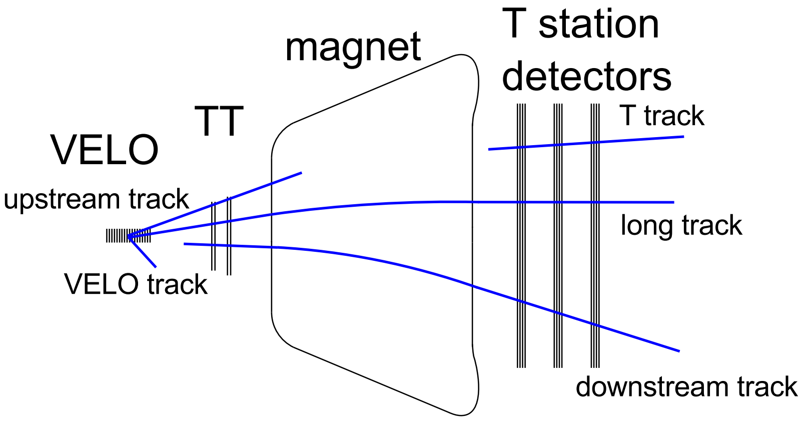

Figure 1: Illustration of the tracking system of LHCb, starting from the VELO around the collision point on the left, particles pass the TT, are deflected in the magnetic field of the dipole magnet and then detected in the T station detector (IT and OT). Different track types are reconstructed by different track finding algorithms. [1]

the numeric response of that algorithm. When the ghost probability is referred to as a selection requirement, the nominal working point corresponding to a fake track retention of 40 % is implied.

1.1 Track reconstruction

Owing to the design of the LHCb detector, which consists of tracking detectors mainly outside the magnetic field, charged particle tracks are approximately straight line segments in the upstream part (VELO and TT) and in the downstream part (T stations). Figure 1 shows an overview of the different track types defined in the LHCb reconstruction: VELO tracks, which have hits in the VELO only; upstream tracks, which have hits in the two upstream trackers; T tracks, which have hits in the T stations only; downstream tracks, which have hits in TT and the T stations; and long tracks, which have hits in the VELO and the T stations. The latter tracks can additionally have hits in TT.

If a particle is reconstructed more than once, as different track types, only the track best suited for analysis purposes is kept. Hereby, long tracks are preferred over any other track type, upstream tracks are preferred over VELO tracks, and downstream tracks are preferred over T tracks.

Most analyses use long tracks because they provide the best momentum and spatial resolution among all track types. Unless otherwise stated, track reconstruction at LHCb refers to the reconstruction of long tracks. In a typical signal triggered event, around 60 long tracks are reconstructed. Other track types, such as downstream tracks, are used for the reconstruction of decay products of long-lived particles such as K^0_S mesons, or for internal alignment of the tracking detectors.

Tracks are fit with a Kalman filter. In addition to a global fit χ^2, separate contributions to the χ^2 from the downstream detectors (IT and OT), χ^2_D, and from the upstream detectors (VELO and TT), χ^2_U are computed. A large number of fake tracks result from wrong combinations of well reconstructed track segments in the upstream and downstream regions. These usually have good χ^2_D and χ^2_U but the additional contribution from matching the two segments, $\chi^2_M = \chi^2 - \chi^2_D - \chi^2_U$, is large for these "matching" fakes.

The Kalman fit performs an outlier removal to account for individual detector hits which are not due to the reconstructed particle. Beyond that, a special treatment for Outer Tracker hits is in place. The readout electronics is designed to select only a single hit in each channel per bunch crossing; if two charged particles pass the same straw, a drift-time measurement will only be provided for one of them. To describe tracks in high occupancy OT modules, the drift-time measurement can be ignored and only the information that a track went somewhere through the straw is used. This is decided for each straw–track combination individually if the hit residual is too large, similar to a standard outlier removal. This drift-time suppression ensures that the track fit χ^2 is not biased to larger values for tracks in high multiplicity events, for tracks in the OT with respect to tracks in the IT, or for tracks in high occupancy modules, which are those closer to the beam axis.

1.2 Previous works

An earlier version of the work presented here [5], referred to as the "old ghost probability", was already used in analyses of Run I data. It was evaluated in the offline reconstruction to distinguish fake tracks from real particles' *e.g.* in Ref. [6]. The network was trained on all reconstructed tracks in simulated events with at least one $b\bar{b}$ pair produced in the pp collision.

The 22 input variables to the old ghost probability are the overall track fit χ^2 along with the individual contributions $\chi^2_D, \chi^2_U, \chi^2_M$ and the corresponding degrees of freedom for each fit; the numbers of hits on the track in each tracking detector; the reconstructed track p_{T} and pseudorapidity; the difference in the number of observed hits on a track and the "expected hits", calculated by interpolating the track through the detector and counting how many active strips/straws the track passes through; and finally the occupancies of all tracking detectors.

There are separate networks for each track type, where input variables are removed if they are not defined for that track type (*e.g.* VELO hits for downstream tracks).

1.3 Network architecture tuning

As framework for the neural network, the TMVA package [7] is chosen since it is

- equipped with a root file interface for the training, which is the common data file format in LHCb software,
- commonly known in LHCb (ensuring future maintainability),
- able to provide code generation for the trained network such that the network can be integrated into any C++ code without creating dependency on external libraries.

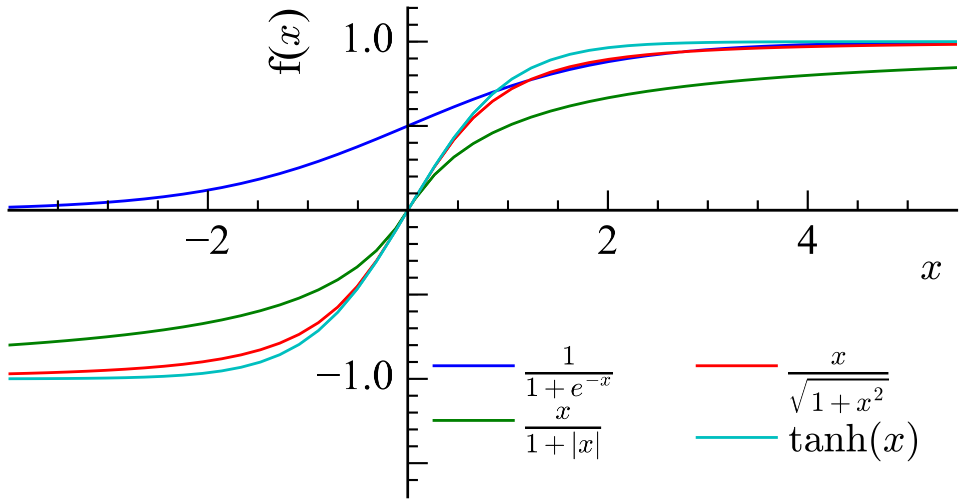

Figure 2: Functional shape of sigmoid functions.

Mathematically, the shallow neural network is implemented as composed function

$$\mathbb{R}^n \xrightarrow{\text{linear + const}} \mathbb{R}^m \xrightarrow{\text{element wise non-linear}} \mathbb{R}^m \xrightarrow{\text{linear + const}} \mathbb{R} \xrightarrow{\text{non-linear}} \mathbb{R}.$$

With operations in each step:

$$\begin{aligned}
(x_i) &\mapsto \left(\sum_i M_{j,i} x_i + b_j\right) \\
(x'_j) &\mapsto \left(f(x'_j)\right) \\
(x''_j) &\mapsto \sum_j M'_{1,j} x''_j + b' \\
x''' &\mapsto f'(x''')
\end{aligned}$$

The parameters of the linear mappings $(M_{j,i}, b_j, M'_{1,j}, b')$ are subject to optimisation, where $1 \leq i \leq n$, $1 \leq j \leq m$, and n is the number of input variables and m is chosen as $n + 5$. The non-linearities (f and f'), so-called "activation functions", are fixed real valued functions.

At the time of development, the $\tanh(x)$ function was a commonly used activation function in TMVA, while known as a computationally expensive function to be optimised for the LHCb pattern recognition [8]. Yet it is not the only possible sigmoid function [9] and consequently custom activation functions have been added to TMVA [10].

Of the tested functions $\frac{x}{\sqrt{1+x^2}}$ is the fastest to compute, while no significant physics performance difference is expected given the similar functional shape, see Fig. 2. In Fig. 3

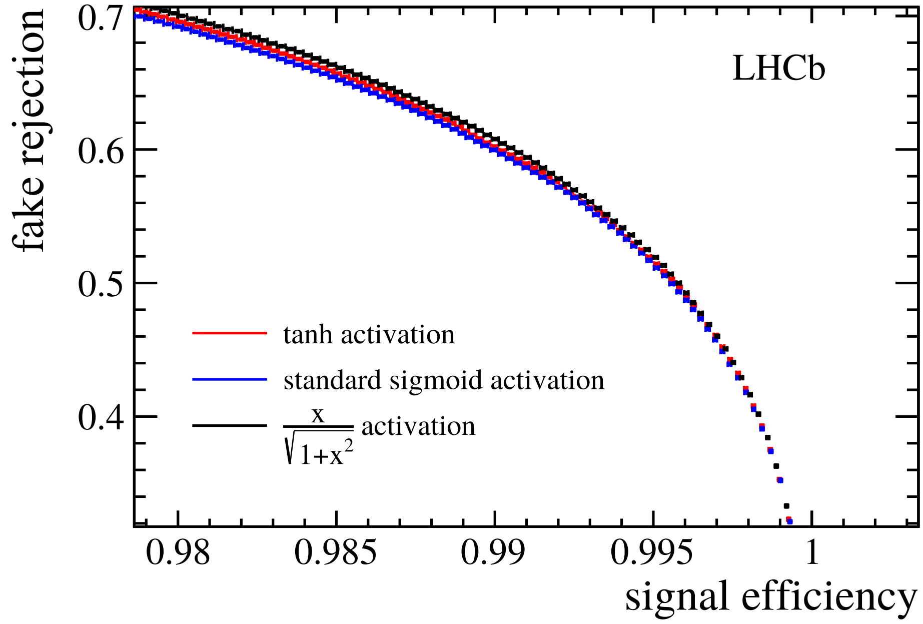

Figure 3: ROC curves for fake track discriminating neural networks, using different activation functions. A very small physics performance improvement is observed when changing from the TMVA standard functions to $\frac{x}{\sqrt{1+x^2}}$.

even a small physics performance improvement is visible, possibly due to its larger gradient with respect to $\tanh(x)$ [11]. Therefore, it is chosen as activation function.

The ghost probability is a classification problem, and thus cross entropy [12] is chosen as the loss function in the network training. With respect to the Run I implementation, where a mean-square-error loss was chosen, this contributes to the physics performance improvement. The activation function of the output layer is $\frac{1}{1-e^{-x}}$ in the training. In the application, a custom output calibration is applied instead, as described in Sect. 1.5.

1.4 Variable selection

To allow for enough development time for testing and evaluation, the selection of input variables is mostly unchanged from Run I with two exceptions. The track interpolation to determine the number of expected hits is removed to reduce the CPU usage of the ghost probability by a factor 10. The number of track candidates competing for shared hits in the pattern recognition is added as input variable.

1.5 Output transformation

To ease the usage of the ghost probability, a transformation of the network response is applied. A probability integral transform – commonly referred to as “flattening” or "rarity transformation" – is implemented as a linear spline fit to the cumulative network response for fake tracks in simulated events. The discriminating behaviour of any classifier is invariant under monotonous transformations and so is the physics performance under the probability integral transform. Motivations for this transformation are primarily to give a physical interpretation to the response: rejecting tracks with a ghost probability of larger than $x\,\%$ will retain $x\,\%$ of all fake tracks.

In addition any update of the ghost probability training will have the same behaviour and thus the optimal working points of algorithms using the ghost probability algorithm will remain unchanged to a good approximation.

1.6 Category classifiers

Fake tracks produced by different pattern recognition algorithms might have different track properties. It might therefore be beneficial to train separate neural networks for the two long track reconstruction algorithms at LHCb. On simulated events, the physics performance of two separate networks does not differ from the physics performance of a single network. Similarly, different networks for different T station regions have been tested (one for tracks in the OT, IT, and the overlap region), without significant performance gain. The possible reasons are twofold. Firstly, the network already knows which T station tracker a track went through due to the hit counts in the individual subdetectors. Secondly, for the different algorithms, it is possible the common track fit is more indicative for whether a track is a fake or not, than the pattern recognition strategy.

Consequently, a single network for all pattern recognition algorithms in the entire detector is deployed.

1.7 Training sample

The LHCb track reconstruction needs to be able to handle a wide range of LHC running conditions. At the time of preparing for data taking in 2015 it was not clear whether the LHC would operate at 25 ns bunch spacing or 50 ns bunch spacing. Four scenarios were simulated:

- 25 ns bunch spacing, $\nu = 1.6$
- 25 ns bunch spacing, $\nu = 1.9$
- 50 ns bunch spacing, $\nu = 1.6$
- 50 ns bunch spacing, $\nu = 2.7$

where ν is the average number of pp interactions per bunch crossing.

The scenarios differ, for what concerns the track reconstruction, significantly in detector occupancy and spillover in the OT. This may lead to different behaviours of fake track reconstruction and require different network trainings for different running conditions. The necessity for having different network trainings is assessed by training networks for each

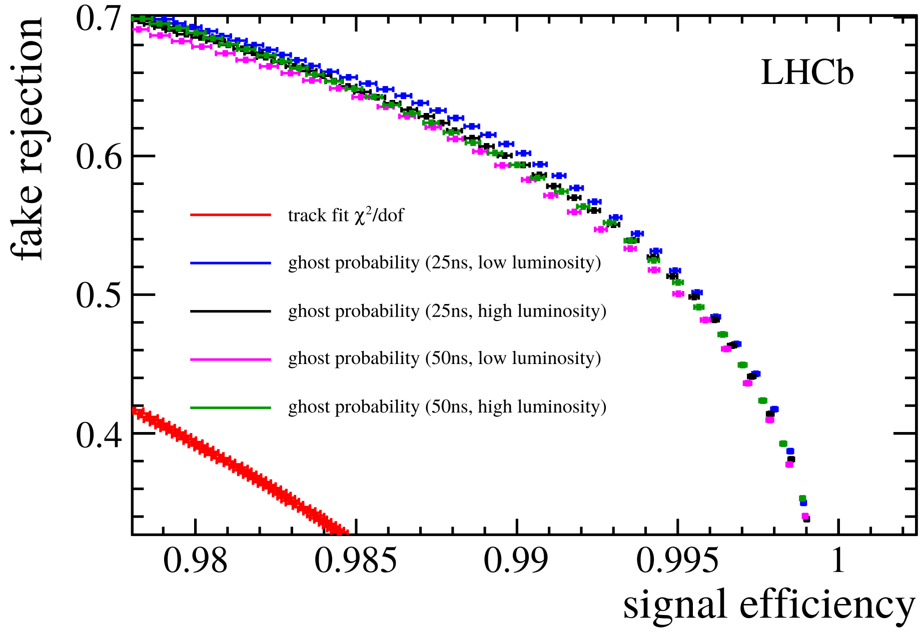

Figure 4: ROC curves for fake track discriminating neural networks (blue, black, magenta, and green), trained for different LHC running conditions, evaluated for 25 ns bunch spacing, $\nu = 1.6$. The red points are the ROC curve for the track fit reduced χ^2, which performs significantly worse than any of the neural networks.

of the running conditions, with all other training parameters fixed, and evaluating the networks on one of the samples and their discriminating powers are compared. Figure 4 shows the receiver operating characteristics (ROC) curves for the 25 ns sample at low pile-up. The discriminating powers of the four networks do not largely differ and thus for simplicity only a single network (trained at the favoured scenario of 25 ns bunch spacing at low pile-up) is deployed.

2 Physics Validation

The data taking strategy of LHCb in Run II involves the application of the same track reconstruction in the software trigger as in the offline data processing. This goal can only be achieved within the time constraints of the software trigger by applying the ghost probability in the trigger. This ghost probability reduces the rate of fake tracks entering the particle identification and combinatorics of decay reconstructions and thereby saves more time than the computation of the ghost probability.

It must therefore be ensured that the full physics program of LHCb can be done with tracks passing the ghost probability, and that there is no corner of phase space or particle species, which is rejected by the ghost probability.

The computation of the track fit χ^2 was last revised in 2015 between data taking

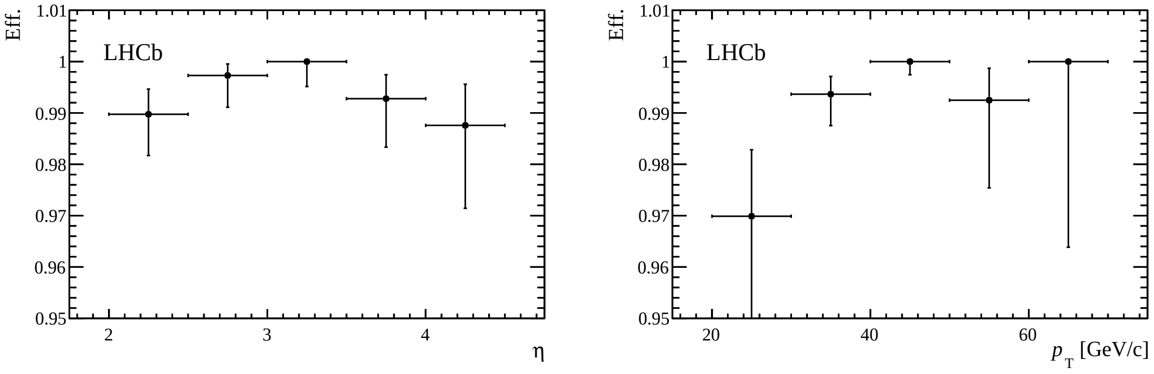

Figure 5: Efficiency for $Z \to \mu\mu$ tracks to pass the ghost probability, in data from 2015 at a bunch spacing of 50 ns.

at 50 ns and 25 ns bunch spacing, [13]. For conclusive validations, tracks in Run I data and from 2015 data with 50 ns bunch spacing are refitted before computing the ghost probability.

2.1 High momentum tracks

Due to their low cross section, stable charged particles at high momenta, in the momentum range of $Z \to \mu\mu$ decays, are absent in the training data. This could lead to a low selection efficiency for very high momentum tracks.

In the early measurement period (corresponding to the first 6 pb^{-1}) in 2015 at a bunch spacing of 50 ns, the nominal 2015 pattern recognition was used without application of the ghost probability. Refitting the candidate tracks from $Z \to \mu\mu$ decays in that period allows to assess the performance of the ghost probability for very high momentum tracks. The measured efficiency in Fig. 5 shows that the absence of very high momentum tracks in the training data does not lead to a low efficiency.

2.2 Electron reconstruction

Electrons are more challenging to reconstruct than the standard candles ($Z \to \mu\mu$ or $D \to K\pi$). At the same time, it can be expected that the response of the ghost probability for electrons differs from that for other particles as electrons undergo more multiple scattering.

It is assumed that the reconstruction of converted photons as electron pair is the most vulnerable part for the following reasons. The photon conversion can happen “late” in the VELO leaving only few hits. In addition, the e^+e^- pair has a small opening angle which could lead to hit ambiguities in the VELO pattern reconstruction. It should be noted that the branching fraction of $B_s^0 \to K^*\gamma$ is anyhow low and the statistical sensitivity of an analysis of this decay would immediately suffer from further efficiency loss.

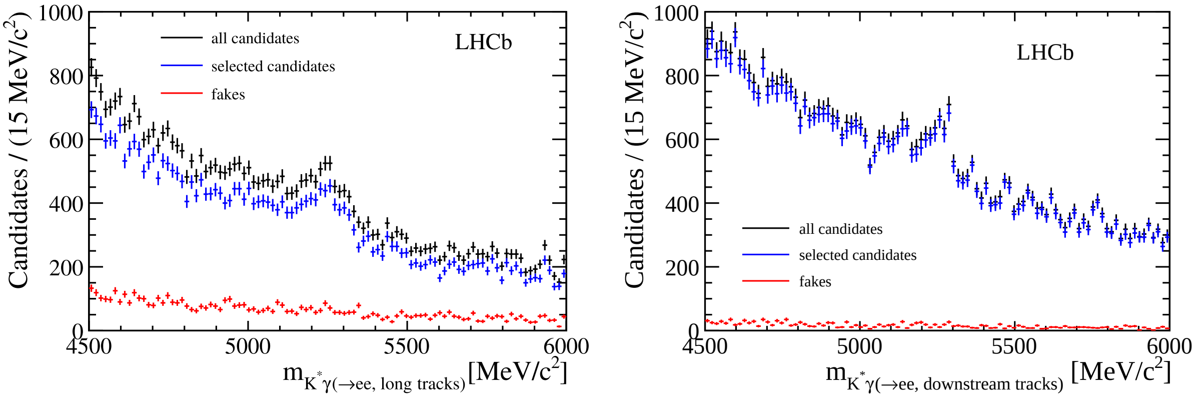

Figure 6: $B_s^0 \rightarrow K^*\gamma$ candidates, where the photon is reconstructed as pair of electron tracks, using long tracks (left) and using downstream tracks (right). The candidates prior to a cut on the ghost probability of the electron tracks are shown in black, those passing the ghost probability in blue, and those candidates rejected by the ghost probability in red. No signal loss is visible in the rejected candidates.

The 50 ns early data of 2015 does not correspond to enough integrated luminosity to obtain a satisfying estimation of the consequences of a cut on the ghost probability on converted photons. For this reason, the tracks of $B_s^0 \rightarrow K^*\gamma$ candidates from Run I – using a simplified version of the selection presented in Ref. [14] without Bremsstrahlungs correction – are refitted using the track fit configuration as used in 2015 and the ghost probability is evaluated. The invariant mass spectrum shown in Fig. 6 shows candidates without the application of a cut on the ghost probability, those passing, and those failing; both for converted photons reconstructed as pair of downstream tracks and as pair of long tracks. To the statistical precision of this test, no signal loss is visible.

2.3 Validation with 25 ns data

A cut on the ghost probability is included in the standard track reconstruction since data taking at 25 ns bunch spacing started. To investigate the behaviour of the ghost probability in real data with 25 ns bunch spacing, events are re-reconstructed without a cut on the ghost probability. Under the assumption, that most K_S^0 are part of the underlying event and most triggered events containing K_S^0 would have been triggered without those K_S^0, K_S^0 are used as probe of the ghost probability.

The invariant mass spectrum of K_S^0 candidates after re-reconstruction is shown in Fig. 7 for both long tracks and downstream tracks. In both cases, K_S^0 candidates are reconstructed from two opposite charged pions which are compatible with originating from a common vertex, which satisfy fiducial momentum requirements, and which are significantly displaced from any primary collision vertex. In either case, the background contribution is largely reduced when rejecting events where at least one of the tracks has a ghost probability of larger than 0.4. There is no signal visible in the events rejected. It

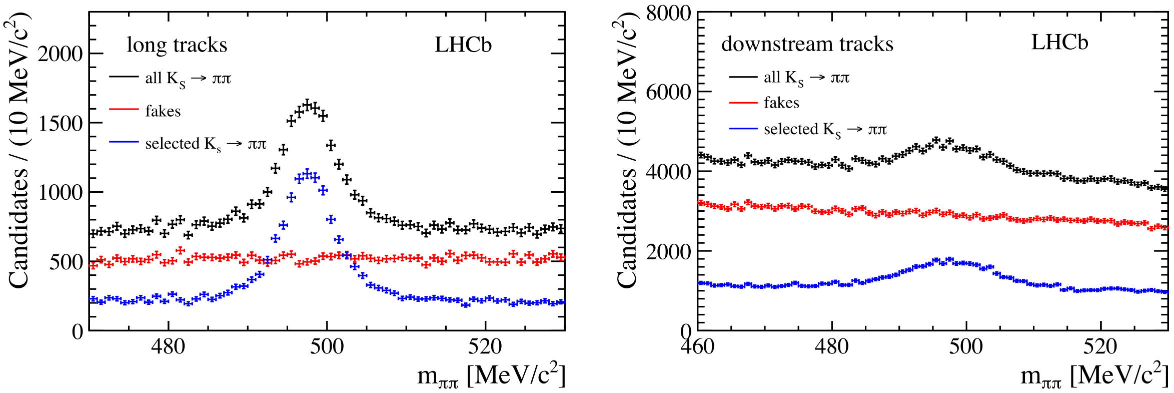

Figure 7: $K^0_\mathrm{S} \to \pi\pi$ invariant mass spectrum for events reconstructed without using the ghost probability in the track selection; using long tracks (left) and downstream tracks (right). Candidates for which at least one track fails the default ghost probability requirement are shown in red and do not exhibit a signal contamination. The remaining candidates are shown in blue.

is concluded that no physics signal is lost due to the application of the ghost probability within the statistical sensitivity of the test in the kinematic spectrum of the selected K^0_S.

2.4 Decay time acceptance

The study of long lived particles (b and c hadrons) is the major part of the LHCb physics program. It must therefore be ensured that the ghost probability does not reject particles from displaced vertices at a higher probability than particles from primary collisions (which have a higher prevalence in the training).

To evaluate a possible decay time bias of the ghost probability, for each reconstructed particle in simulated events with a $b\bar{b}$ production, the average of the true decay time of the ancestor particles is determined. When rejecting tracks which fail the ghost probability criterion, the average decay time of the parent particle changes by (1.5 ± 2.0) fs. This is smaller than the statistical sensitivity of this test, and smaller than the systematic uncertainty to which the lifetime bias of the LHCb reconstruction is known [15].

3 Impact on trigger CPU consumption

Applying the ghost probability in the first software trigger stage, reduces the rate of events which pass the first stage, and therefore the HLT2 reconstruction needs to reconstruct 4 % less events. In combination with the ghost probability requirement in HLT2, the RICH PID is executed for 22 % fewer tracks and the CPU used for it is reduced by 23 %. The number of combinations, in the 2-body topological trigger, is reduced by 69 %, resulting in 53 % fewer trigger candidates. The CPU consumption for combinatorics for all trigger lines is reduced by 58 %.

The evaluation of the ghost probability itself "costs" 0.2 % of the HLT CPU budget (in roughly equal parts for HLT1 and HLT2), which underlines the overall benefit of its application.

For the entire software trigger this results in a reduced CPU consumption of 16 % – assuming that roughly half of the current HLT farm costs $\mathcal{O}(2\,\mathrm{M\,CHF})$, this is equivalent to 640 k CHF.

Lastly, if the ghost probability was removed from the HLT and the remaining setup not adapted, the output rate would increase by 36 %.

4 Outlook

The current networks are trained for the track reconstruction for data taking in 2015 at 25 ns bunch spacing, using the latest simulations available at the time. Retrainings are advisable for significant updates in the track reconstruction before the ghost probability (*i.e.* the pattern recognition and the track fit) and to account for expected changes in the detector performance due to radiation damage. Physics performance gains can also be expected with improved machine learning techniques or event simulations.

Additional separation between "good" tracks and fake tracks could be gained by testing more detailed information about the hit expectations in active layers. The old ghost probability used only the number of expected hits per subdetector, instead of their locations.

The current training is purely based on simulated events, the domain adaptation approach from Ref. [16] is not applied as it currently does not lead to an improved fake track rejection. The ghost probability network is retrained using good tracks and fake tracks from simulated events and unlabelled tracks from real events with the Caffe software framework [17][3]. In addition to the network with domain adaptation, a conventional network is trained to disentangle effects from the training algorithm (replacing TMVA by Caffe) and network architecture (adding a gradient reversal layer and domain classifier). This working point of the network responses is chosen to retain the same number of candidates in a $K^0_\mathrm{S} \to \pi\pi$ selection as the application of the nominal ghost probability. From the invariant mass distribution in Fig. 8, it can be seen that the TMVA network and the two networks trained in Caffe yield close to identical physics performance; the data points of the network with domain adaptation are almost entirely covered by the data points of the Caffe network without domain adaptation. This does not rule out that domain adaptation can improve the physics performance of the ghost probability in the future.

The current network relies on auto-vectorisation. The methods suggested by Ref. [18] lead to tenfold improvement of the CPU cycle count of the neural network implementation

[3] An implementation of the method suggested in Ref. [16] has been provided by its authors in the Caffe framework.

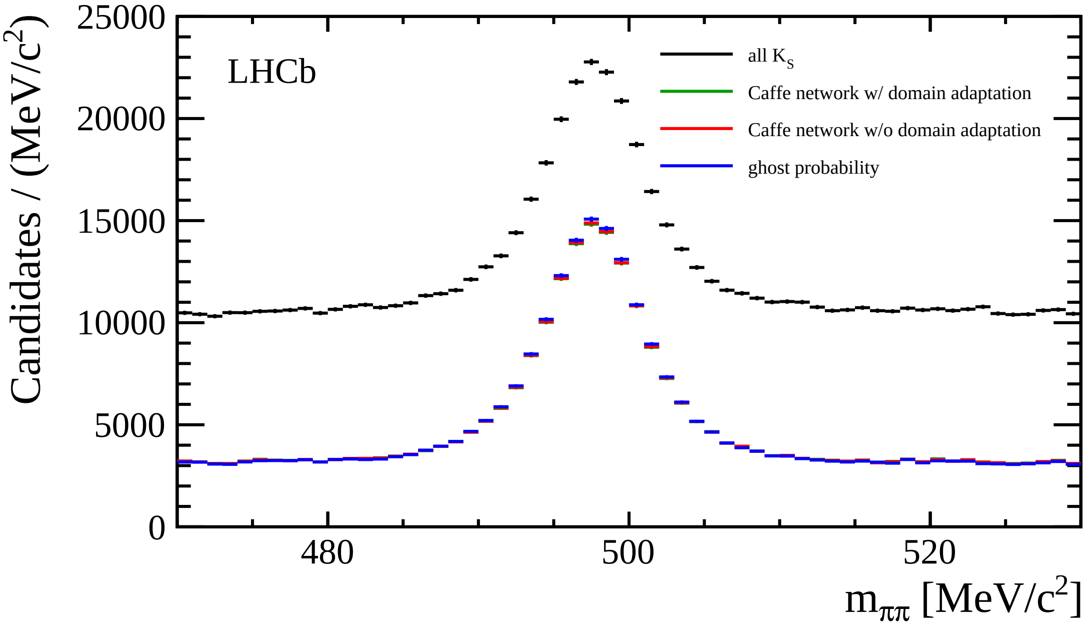

Figure 8: Comparison of the K^0_S selection with the nominal ghost probability (blue), a network trained with the Caffe package (red), and a network trained with domain adaptation (green, covered by red). The working points of the Caffe networks are chosen to retain the same numbers of candidates. Hardly any performance difference is visible.

Table 1: Callgrind benchmark comparisons of different activation functions. Fields with n/a have not been evaluated or are not available with AVX intrinsics. The activation function used by the ghost probability is marked with (*).

function	default compiler options	AVX vectorisation by hand
tanh	19,355,124,355	n/a
$\frac{1}{1+e^{-x}}$	21,140,125,632	n/a
$\frac{x}{\sqrt{1+x^2}}$(*)	415,121,741	195,121,939
$\frac{x}{1+\|x\|}$	395,121,798	195,104,759
$\max(0,x)$	155,095,875	115,095,891

[19] using AVX [20] intrinsic commands. This approach has not been followed up to ensure platform independence of the ghost probability.

The current activation function in the neural network is $\frac{x}{\sqrt{1+x^2}}$. The rectified linear unit $\max(0,x)$ or $\frac{x}{1+|x|}$ are expected to be even faster, as listed in Table 1.

5 Conclusion

The ghost probability is introduced as a the default method of reducing the number of fake tracks in the LHCb reconstruction and is deployed for offline and online reconstruction. The reduction of fake tracks in the particle identification and combinatorics of decay reconstruction greatly reduces the demand of computing resources of the software trigger and enables LHCb to use identical reconstructions online and offline. Validations at different phase space points reveal no adverse side effects of applying the ghost probability centrally.

Acknowledgements

We express our gratitude to our colleagues in the LHCb collaboration who provided suggestions and helped in the implementation, especially Angelo Di Canto, Helge Voss, Manuel Schiller, Gerhard Raven, Chris Jones; from the Electronic Vision(s) group (Kirchhoff-Institute for Physics, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany) Eric Müller; and from the LHC Physics and New Particles group (Institute for Theoretical Physics, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany) Johann Brehmer.

References

[1] LHCb collaboration, R. Aaij *et al.*, *Measurement of the track reconstruction efficiency at LHCb*, JINST **10** (2015) P02007, `arXiv:1408.1251`.

[2] LHCb collaboration, R. Aaij *et al.*, *LHCb detector performance*, Int. J. Mod. Phys. **A30** (2015) 1530022, `arXiv:1412.6352`.

[3] S. Broghi *et al.*, *REAL TIME CALIBRATION public document*, , in preparation.

[4] J. Weidendorfer, M. Kowarschik, and C. Trinitis, *A Tool Suite for Simulation Based Analysis of Memory Access Behavior*, in *Computational Science - ICCS 2004, 4th International Conference, Kraków, Poland, June 6-9, 2004, Proceedings, Part III* (M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, eds.), vol. 3038 of *Lecture Notes in Computer Science*, pp. 440–447, Springer, 2004. http://www.valgrind.org/docs/pubs.html.

[5] J. Brehmer, J. Albrecht, and P. Seyfert, *Ghost probability: an efficient tool to remove background tracks*, `https://cds.cern.ch/record/1478372`. LHCb internal note LHCb-INT-2012-025.

[6] LHCb collaboration, R. Aaij *et al.*, *Observation of $B_c^+ \to J/\psi D_s^+$ and $B_c^+ \to J/\psi D_s^{*+}$ decays*, Phys. Rev. **D87** (2013) 112012, `arXiv:1304.4530`.

[7] A. Hoecker *et al.*, *TMVA: Toolkit for Multivariate Data Analysis*, PoS **ACAT** (2007) 040, arXiv:physics/0703039.

[8] M. Schiller, H. Voss, and L. Moneta, *code available on Jira ticket: fast tanh implementation*, https://sft.its.cern.ch/jira/browse/ROOT-7054.

[9] Wikipedia. Sigmoid function, 2015.

[10] P. Seyfert and H. Voss, *code available on Jira ticket: TActivation... implementations*, https://sft.its.cern.ch/jira/browse/ROOT-7062.

[11] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, PMLR, 13–15 May, 2010.

[12] C. M. Bishop, *Pattern recognition and machine learning*, Information science and statistics, Springer, New York [u.a.], 10. (corrected at 8th printing) ed., 2009; J.-H. Zhong *et al.*, *A program for the Bayesian Neural Network in the ROOT framework*, Comput. Phys. Commun. **182** (2011) 2655, arXiv:1103.2854.

[13] M. Heß, *Multiple scattering in track reconstruction*, https://cds.cern.ch/record/1957764. LHCb internal note LHCb-INT-2014-043.

[14] L. Beaucourt, E. Tournfier, M. N. Minard, and J. F. Marchand, *$B^0 \to K^*\gamma(e^+e^-)$ and $B^0_s \to \phi\gamma(e^+e^-)$ analysis status*, in *2nd Radiative decays @LHCb Workshop* (A. Oyanguren Campos *et al.*, eds.), 2015. https://indico.cern.ch/event/375424/.

[15] LHCb collaboration, R. Aaij *et al.*, *Measurements of the B^+, B^0, B^0_s meson and Λ^0_b baryon lifetimes*, JHEP **04** (2014) 114, arXiv:1402.2554.

[16] Y. Ganin and V. Lempitsky, *Unsupervised Domain Adaptation by Backpropagation*, ArXiv e-prints (2014) arXiv:1409.7495.

[17] Y. Jia *et al.*, *Caffe: Convolutional Architecture for Fast Feature Embedding*, arXiv:1408.5093.

[18] V. Vanhoucke, A. Senior, and M. Z. Mao, *Improving the speed of neural networks on CPUs*, https://research.google.com/pubs/archive/37631.pdf.

[19] P. Seyfert, *code available on github project TMVA-MLP*, https://github.com/pseyfert/tmva-mlp.

[20] N. Firasta *et al.*, *Intel AVX: New frontiers in performance improvements and energy efficiency*, Intel white paper **19** (2008) 20, https://software.intel.com/sites/default/files/m/9/e/1/8/1/16820-Intel_AVX_New_Frontiers_in_Performance_Improvements_and_Energy_Efficiency_WP.pdf.