Identification of Ghost Tracks using Neural Networks

LHCb Note

Issue:	1
Revision:	0
Reference:	LHCb 2007-158
Created:	October 31, 2007
Last modified:	December 21, 2007

Prepared by: Adrian Perieanu
Physikalisches Institut Heidelberg

Abstract

The tracks samples found by the pattern reconstruction algorithms have a high efficiency but also a large fraction (18.5%) of misidentified tracks, named ghost tracks.

To offer the possibility of having a tracks sample with a high track efficiency (as high as from the pattern reconstruction algorithms) and in the same time with a very low ghost rate, a neural network has been developed.

Variables as χ^2 probability or number of hits in Vertex Locator are combined with internal pattern reconstruction algorithm variables as matching χ^2 or quality of PatForward tracks in a neural network. The packages TMultiLayerPerceptron and TMVA from ROOT[1] software are used to setup the neural network.

Results of various tests performed with the neural network on best long tracks are presented.

Guide lines of usage of the first neural network tool implemented in the reconstruction software, Brunel[2], are given.

[1]) http://www.root.cern.ch

[2]) http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/brunel/

Document Status Sheet

1. Document Title: Identification of Ghost Tracks using Neural Networks			
2. Document Reference Number: LHCb 2007-158			
3. Issue	**4. Revision**	**5. Date**	**6. Reason for change**
Draft	1	October 22, 2007	First version.
Final	1	November 19, 2007	Final version.

Contents

List of Figures

1 Introduction

The LHCb detector is a forward spectrometer which covers a pseudo-rapidity range up to 5.2. Tracks of the particles produced in the pp collision are reconstructed in the LHCb tracking system. This is composed from Vertex Locator (VeLo), Trigger Tracker (TT) and Tracking Stations (TS). A detailed LHCb detector description can be found at [1].

There are several tracks categories that can be reconstructed in LHCb: upstream tracks with VeLo and TT hits, long tracks which use hits from all tracking system, Ttracks with hits from TS and downstream tracks which combines hist from TT and TS.

The pattern reconstruction algorithms in LHCb software for long tracks are TrackMatching [2] and PatForward [3]. These algorithms store the found tracks in containers.

General informations about tracks as χ^2 probability ($P(\chi^2)$) or number of degrees of freedom (n.d.f.) of the Kalman filter can be combined with specific variable of the pattern reconstruction algorithms as matching χ^2 (χ^2_{Match}) or quality of patforward tracks ($Q_{PatForward}$) in a neural network (NN). The complete set of variables used in NN is presented in Sec. 2.

Misidentified tracks can be found with a certain probability using this NN. These tracks are classified as ghost tracks as long as they have less than 70% of hits matching the hits of any final state Monte Carlo particle. A large amount of ghost tracks can deteriorate signals that are looked for in $b\bar{b}$ analysis thus two packages have been used to setup the NN: TMultilayerPerceptron [4] and TMVA [5] from ROOT software. The NN performance can be seen in Sec. 3.

Results of various NN studies are shown for best long tracks in Sec. 4. The best tracks are stored in the track container named Default. They are tracks, from all track containers, that pass the Clone Killer [6] selection. Out of them, the tracks which have been found with TrackMatching and PatForward algorithms are selected. These are the best long tracks.

A Monte Carlo (MC) sample generated and simulated for $B^0 \to J/\Psi K^0_S$ with $J/\Psi \to \mu^+\mu^-$ and $K^0_S \to \pi^+\pi^-$ was used to select the tracks in the reconstruction software Brunel v31r7.

In Sec. 7 are presented the types of ghosts stored in the track container and their rate evolution with the NN decision.

A study of the NN behaviour at high luminosity (5×10^{32}cm^{2}s^{-1}) is shown in Sec. 5.

NN have been tested also on different MC samples based on other B mesons decay channels. The results on this studies are illustrated in Sec. 6.

The first version of the NN is implemented in Brunel v31r10 under Tr/NNTools/v1r0 package. Performances of the NN in Brunel v31r9 and details of NN usage in Brunel and DaVinci frameworks are presented in Sec. 8. The results presented here are compatible with the tests done in Brunel v31r12.

2 Neural Network Variables

Combining variables that can discriminate between ghost and good tracks in a NN has the advantage of being more efficient than using cuts on each variable alone. The NN has the capability of learning how to distinguish between ghost and good tracks in the training phase. The testing phase gives a feed-back of the learning process, here for each new track a ghost probability is returned.

In this section the variables used to train the NN are shown. These tracks variables can be classified in *general variables* and *specific variables* to pattern reconstruction algorithms.

The general variables are easily accessible from the track object. The specific variables are built inside the pattern reconstruction algorithms and then written as additional information to the track.

2.1 General Variables

From the track Kalman fit the $P(\chi^2)$ and n.d.f. variables are used. The separation power on best long tracks can be seen in Fig. 1 and in Fig. 2, respectively.

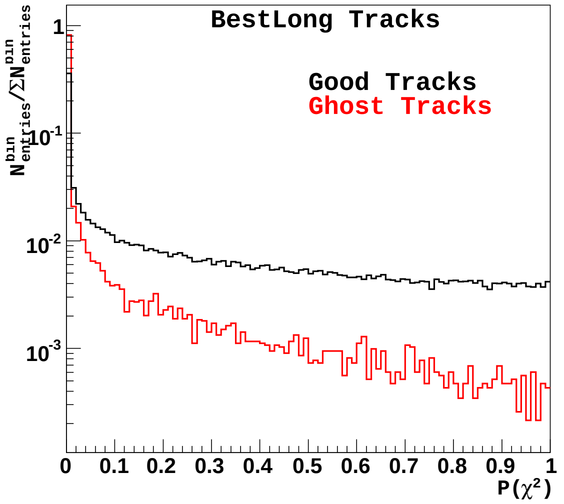

Figure 1: *The $P(\chi^2)$ distribution for best long tracks. The histograms are normalized to 1.*

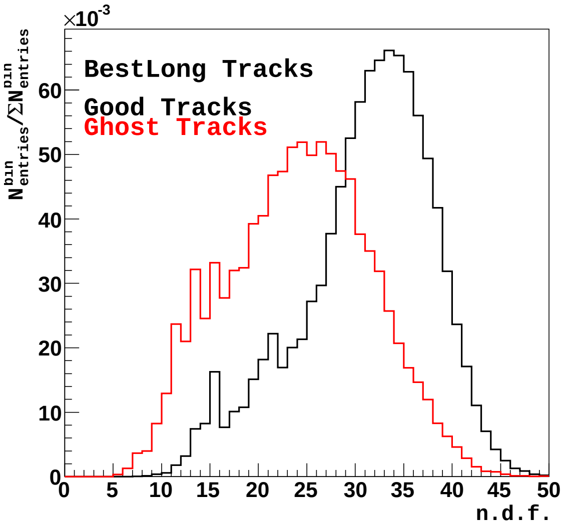

Figure 2: *The n.d.f. distribution for best long tracks. The histograms are normalized to 1.*

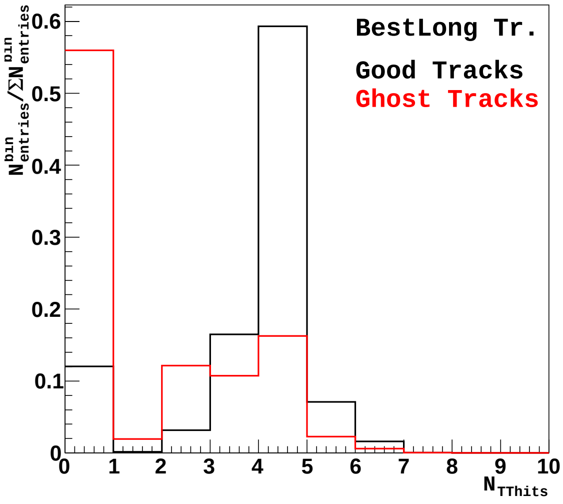

Figure 3: *The N_{TThits} distribution for best long tracks. The histograms are normalized to 1.*

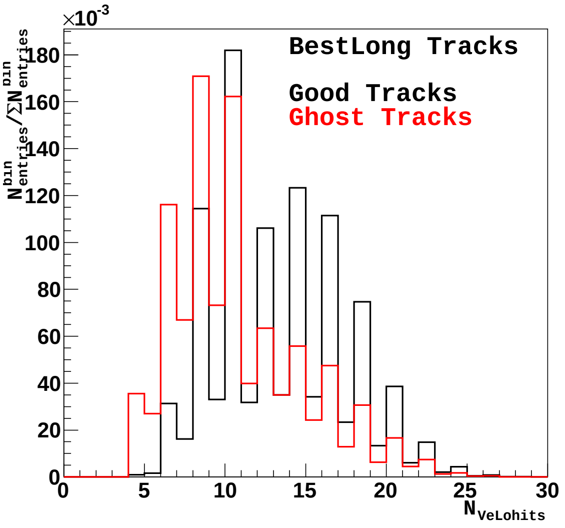

Figure 4: *The $N_{VeLohits}$ distribution for best long tracks. The histograms are normalized to 1.*

The number of hits in TT (N_{TThits}) and the one in VeLo ($N_{VeLohits}$) are also used. These are presented in Fig. 3 and in Fig. 4, respectively.

In the studies presented here the number of tracks in an event ($N_{tracks/ev.}$) and the track pseudorapidity (η) complete the general variables list. Their spectra can be seen in Fig. 5 and Fig. 6, respectively. The variable $N_{tracks/ev.}$ is removed in version v1r0, see Sec. 8.

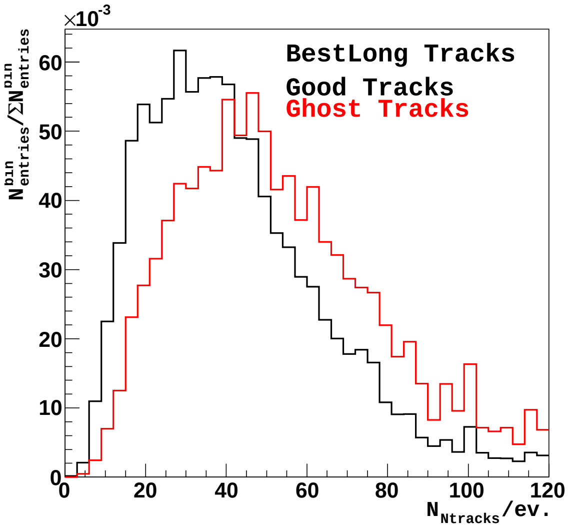

Figure 5: *The $N_{tracks/ev.}$ distribution for best long tracks. The histograms are normalized to 1.*

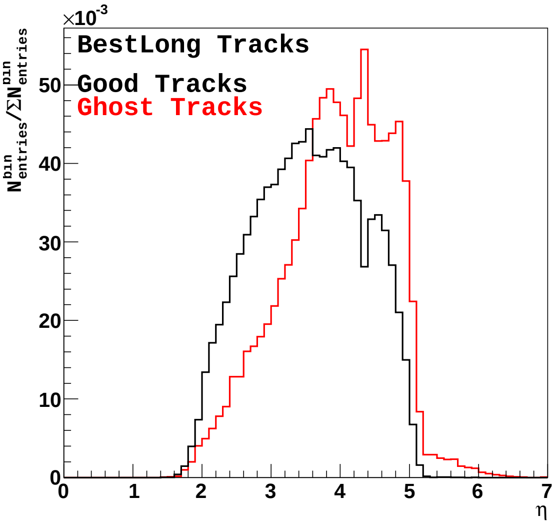

Figure 6: *The $N_{VeLohits}$ distribution for best long tracks. The histograms are normalized to 1.*

2.2 Specific Variables

Long tracks are built having as seed the VeLo track segment. To the seed are added TS hits in PatForward algorithm and matched TS track segments in TrackMatching algorithm.

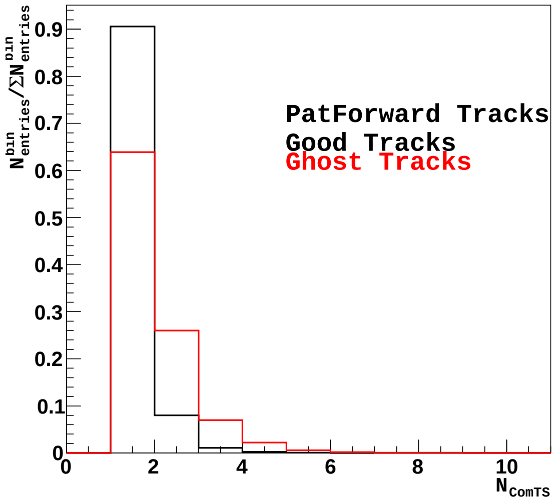

Figure 7: *The N_{ComTS} distribution for PatForward tracks. The histograms are normalized to 1.*

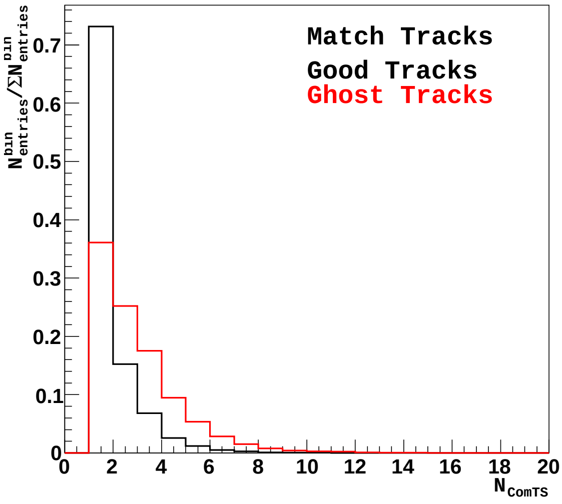

Figure 8: *The N_{ComTS} distribution for Match tracks. The histograms are normalized to 1.*

One specific variable is the number of candidates found for one track which have 70% of the TS hits in common, as in PatForward, or which have the same TS track segment, as in TrackMatching (N_{ComTS}). Fig. 7 and Fig. 8 illustrate the N_{ComTS} variable for PatForward and TrackMatching sample, respectively.

A key variable in the TrackMatching algorithm is χ^2_{Match}. Thus for these tracks χ^2_{Match} and the difference in χ^2 between the first and the second candidate of a track ($\Delta\chi^2$) are used in NN. The sorting of the candidates is done from lowest χ^2 to highest χ^2, for more details see [7]. The χ^2_{Match} and $\Delta\chi^2$ distributions are presented in Fig. 9 and Fig. 10 respectively.

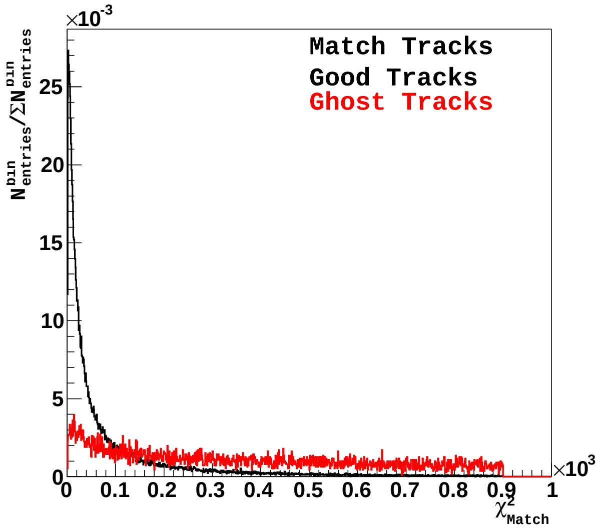

Figure 9: *The χ^2_{Match} distribution for Match tracks. The histograms are normalized to 1.*

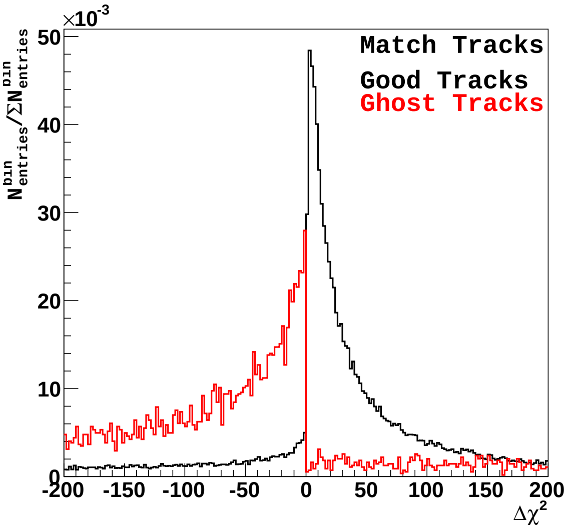

Figure 10: *The $\Delta\chi^2$ distribution for Match tracks. The histograms are normalized to 1.*

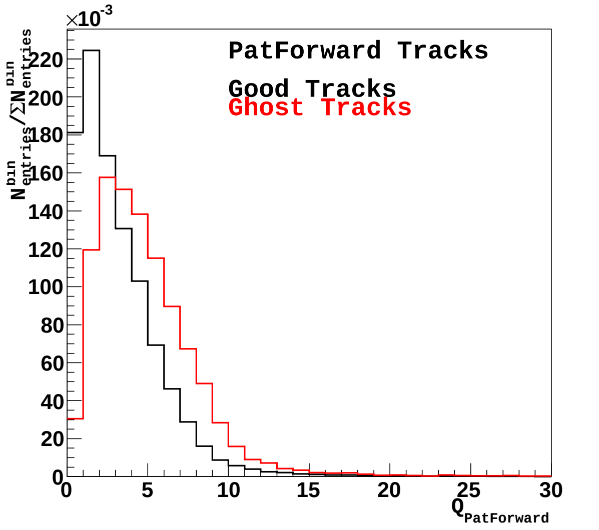

Figure 11: *The $Q_{PatForward}$ distribution for PatForward tracks. The histograms are normalized to 1.*

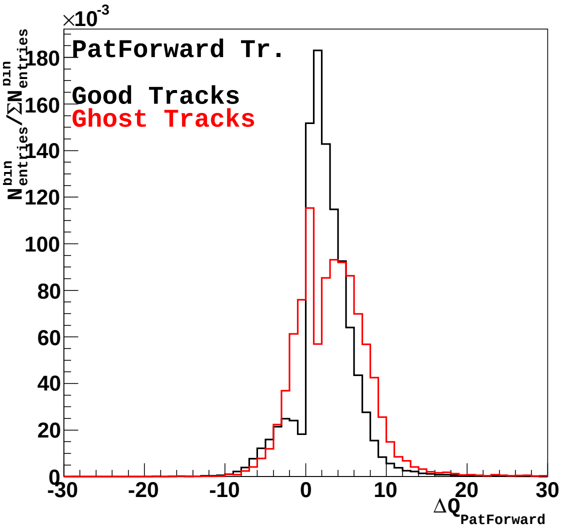

Figure 12: *The $\Delta Q_{PatForward}$ distribution for PatForward tracks. The histograms are normalized to 1.*

With a similar key role the $Q_{PatForward}$ is built in the PatForward algorithm. This variable together with the difference in $Q_{PatForward}$ ($\Delta Q_{PatForward}$) between first and second candidate are completing the list of specific variables. Here the ordering is done with respect to $Q_{PatForward}$, see [7]. In Fig. 11 one can observe the $Q_{PatForward}$ distribution and in Fig. 12 the $\Delta Q_{PatForward}$ one.

The different structures seen in the differential variables of the pattern recognition algorithms are due to the fact that in TrackMatching a cut is applied on χ^2_{Match} and all candidates with smaller χ^2_{Match} are taken as tracks, while in PatForward the number of accepted candidates is limited due to the selection in steps of the best candidates, see [3].

3 Neural Network Performance

In order to check the training process of the NN one has to investigate the separation power of the NN, see Fig. 13, and its performances.

The performance of the NN presented in this paper is illustrated using two distributions. Each NN can be described by the *Background Rejection* vs. *Signal Efficiency* distribution as shown in Fig. 14.

The red curve shows the NN performance for training and testing on TrackMatching (Match) container, the blue curve on PatForward container and the black one on best long tracks container.

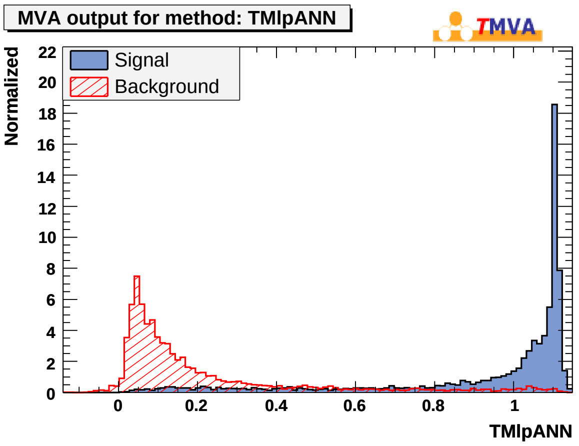

Figure 13: *NN separation power: good (signal) vs. ghost (background) tracks.*

The *Background Rejection* refers to the fraction of ghost tracks rejection when a cut on the NN decision is applied and it is calculated with respect to the initial value of the ghost rate in the sample. If all ghost tracks are removed, *Background Rejection* is 1 (100%), but almost no track is left and the *Track Efficiency* will go towards 0.

The *Signal Efficiency* gives a measure of how many of the good tracks from the initial fraction remains when a cut on the NN decision is used. When all tracks are kept the *Signal Efficiency* is 1 (100%), but no ghost track is rejected and the *Background Rejection* is 0.

An interpretation of the NN performance in terms of *Track Efficiency* vs. *Ghost Rate* can be seen in Fig. 15.

The *Track Efficiency* is calculated as the ratio between the total number of found tracks that were associated to a final state MC particle and the total number of MC reconstructible particles. Under MC reconstructible particle it is understood the MC particle which can be reconstructed using the VeLo and the TS detector information, see [8]. The association between a track and a MC particle is done when at least 70% of the track hits belong to the MC particle.

The *Ghost Rate* represents the relative amount of tracks that could not be associated to a final state MC particle from the total number of tracks which were found.

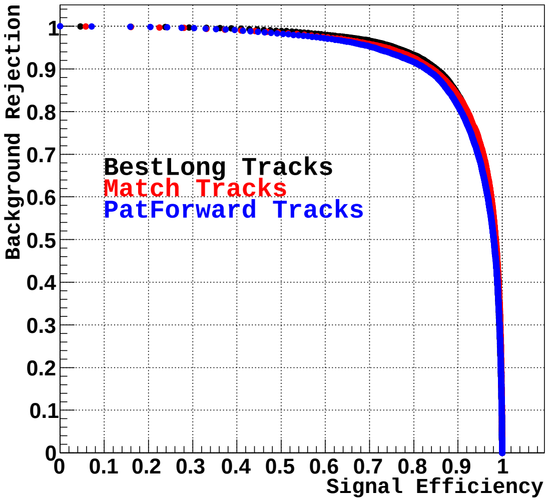

Figure 14: *NN performance: Background Rejection vs. Signal Efficiency.*

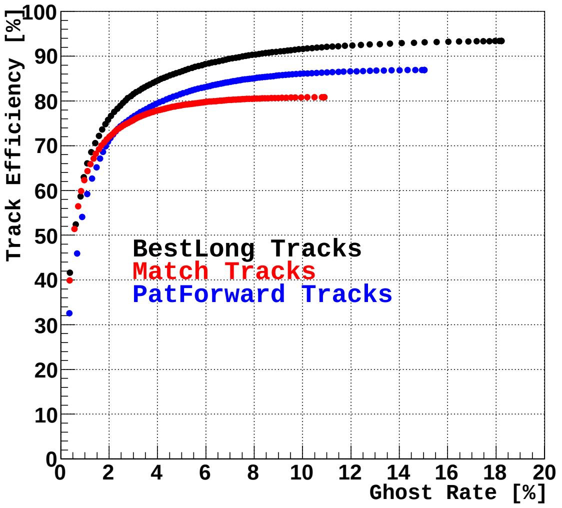

Figure 15: *NN performance: Tracks Efficiency vs. Ghost Rate.*

4 Studies on Different Types of Tracks in an Event

4.1 Physics Signal Tracks

In order to validate the NNTools, a series of studies have been performed. One of them involves to the tracks that belong to the decay products of the B meson decay which is simulated. These tracks are named physics signal tracks.

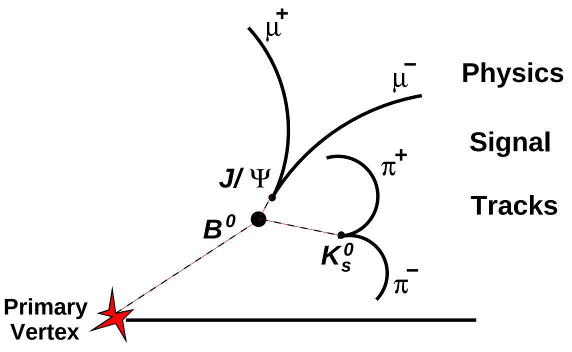

Figure 16: *Schematic view of a* $B^0 \to J/\Psi K^0_S$ *decay.*

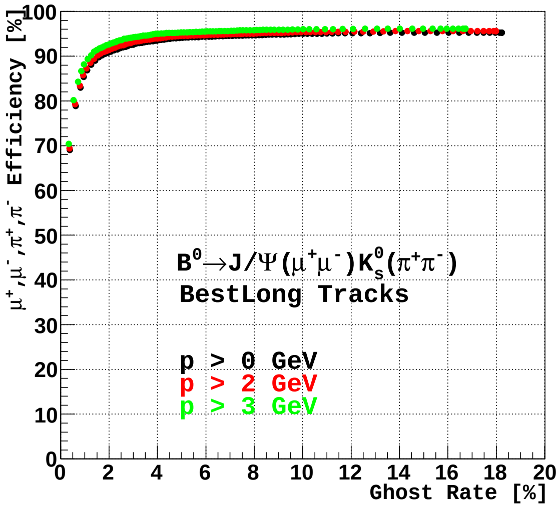

Figure 17: *Physics Signal Tracks Efficiency vs. Ghost Rate.*

In Fig. 16 a schematic view is presented for an event where a $B^0 \rightarrow J/\Psi K^0_S$ is produced. The μ^+, μ^-, π^+ and π^- coming from the grand-mother B^0 are the physics signal tracks for this type of events.

The efficiency for the physics signal tracks is defined as ratio between the tracks that can be associated to the B^0 meson final state decay particles and the number of reconstructible decay MC particles, see Fig. 17.

The four tracks are counted individually, e.g. if only three decay MC particles are reconstructible and the three corresponding tracks are found, the physics signal tracks efficiency will be 100%.

This distribution is plotted also for different momentum cuts: for the black curve no cut is applied, the red one has a cut of $p > 2$ GeV and the green curve shows the case when tracks have $p > 3$ GeV.

4.2 Opposite Side Tagger Tracks

The event where a $b\bar{b}$ pair of quarks is produced is one of the most interesting ones for the LHCb.

Such an event can be split in two parts: one of the *physics signal* where all decay particles of a B^0 meson are reconstructed, see Sec. 4.1, and one with tracks originating from the same vertex as second quark of the produced pair or decay particles of the $\overline{B^0}$ meson, *opposite side tagger*.

In Fig. 18 one can see a schematic view of such an event.

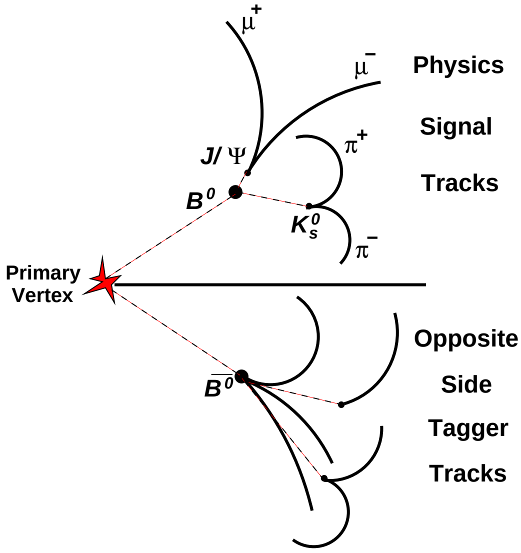

Figure 18: *Schematic view of a $b\bar{b}$ event.*

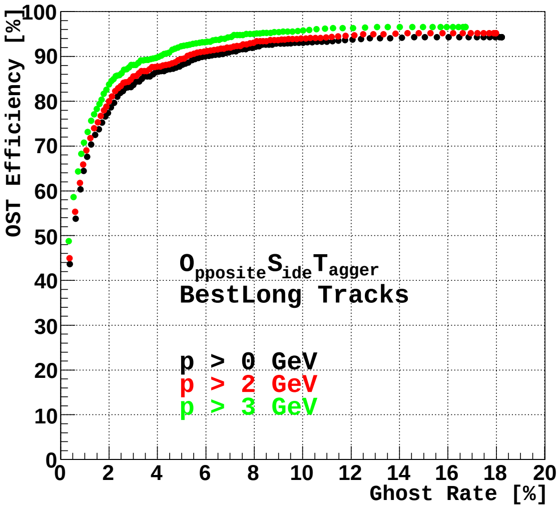

Figure 19: *Opposite Side Tagger Tracks Efficiency vs. Ghost Rate.*

The opposite side tagger tracks efficiency, calculated as physics tracks efficiency in Sec. 4.1, is plotted with respect to the ghost rate in Fig. 19.

4.3 Electron and Positron Tracks

The electron and positron tracks were also studied. In Fig. 20 the track efficiency in case of electrons and positrons which have as mother or grand-mother the B mesons is presented as a function of the ghost rate. The observed steps structure is due to small amount of electrons and positrons in the MC sample.

The distribution is plotted also for electron and positron tracks with $p > 2$ GeV, see the red curve, and $p > 3$ GeV, the green curve.

The efficiency drops when the momentum cut is set to 3 GeV with $\Delta_{Efficiency} = -6\%$ compared with no momentum cut distribution.

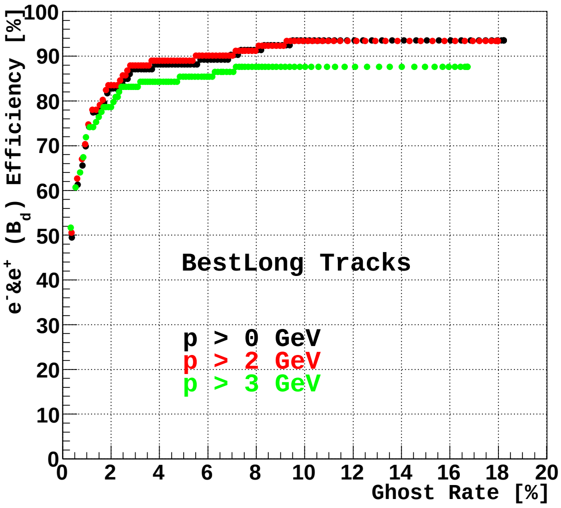

Figure 20: *Electrons Tracks Efficiency vs. Ghost Rate.*

5 Neural Networks at High Luminosity

The performance of the NN at high luminosity, $5 \times 10^{32}\text{cm}^2\text{s}^{-1}$, can be seen for track efficiency in Fig. 21 and for physics signal tracks efficiency in Fig. 22. The red curve shows the NN performance at nominal luminosity, $2 \times 10^{32}\text{cm}^2\text{s}^{-1}$. The NN behaviour shows that one can reduce the ghost rate for the high luminosity events to 10% with minimal losses in efficiency.

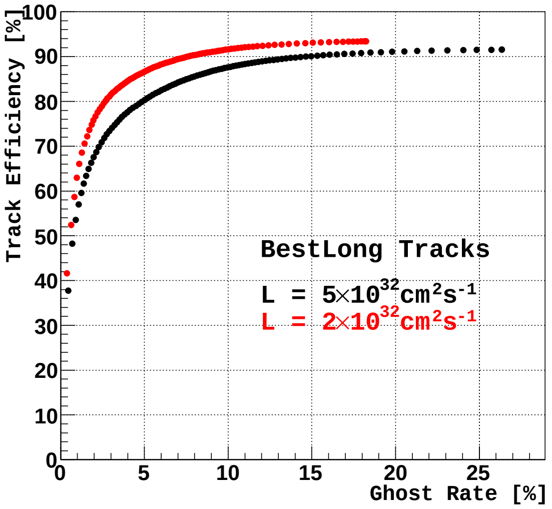

Figure 21: *Track Efficiency vs. Ghost Rate at nominal and high luminosity.*

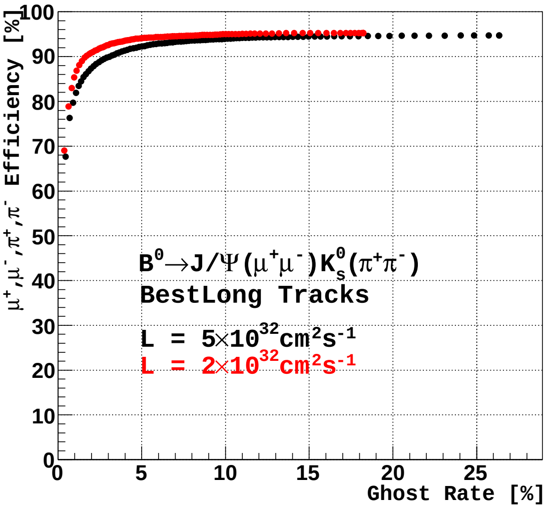

Figure 22: *Physics Signal Tracks Efficiency vs. Ghost Rate at nominal and high luminosity.*

6 Test on a Different B Meson MC Sample

Another useful test is to check the NN trained on one B meson MC sample on a sample with a different B meson.

To perform this test the $B^+ \rightarrow \overline{D^0}K^+$ has been chosen. The decay of the $\overline{D^0} \rightarrow K^0_S\pi^+\pi^-$ and $K^0_S \rightarrow \pi^+\pi^-$ makes this MC sample a lot different from the tracks topology of the $B^0 \rightarrow J/\Psi K^0_S$ decay with $J/\Psi \rightarrow \mu^+\mu^-$ and $K^0_S \rightarrow \pi^+\pi^-$.

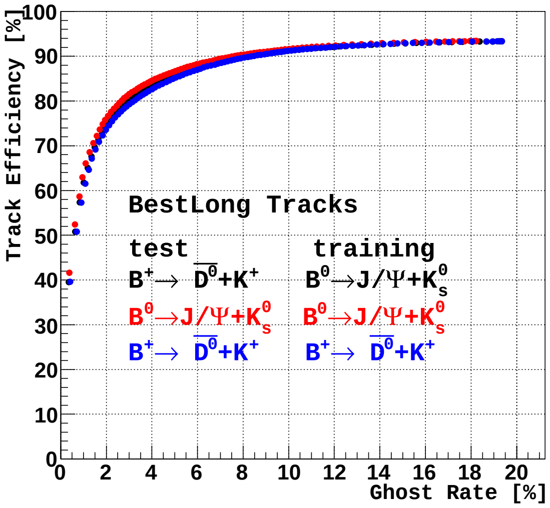

Figure 23: *Track Efficiency vs. Ghost Rate for different MC samples.*

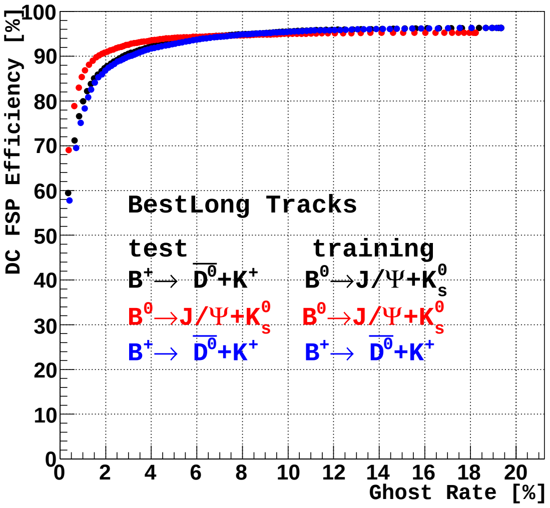

Figure 24: *Physics Signal Tracks Efficiency vs. Ghost Rate for different MC samples.*

In order to test the NN, three histograms are plotted in Fig. 23 for track efficiency vs. ghost rate and in Fig. 24 for physics signal track efficiency vs. ghost rate.

The red curves show the NN performance for testing procedure on $B^0 \rightarrow J/\Psi K^0_S$ when training was done on the same $B^0 \rightarrow J/\Psi K^0_S$ MC sample.

The blue curves indicate the NN performance for testing procedure on $B^+ \rightarrow \overline{D^0}K^+$ when training was done on the same $B^+ \rightarrow \overline{D^0}K^+$ MC sample.

The black curves depict the NN performance for testing procedure on $B^+ \rightarrow \overline{D^0}K^+$, but in this case the training was done on the $B^0 \rightarrow J/\Psi K^0_S$ MC sample. In order to observe these black curves one has to zoom in the blue curves. It can be seen that the black curves emulate the blue ones. This observation shows that one can use the NN trained on one B meson MC sample to other MC samples with different B mesons.

7 Ghost Types

One important question at this stage is *how can the NN performance be further improved?*

For this, one has to identify which ghost tracks are rejected and how evolves their rejection with the NN decision.

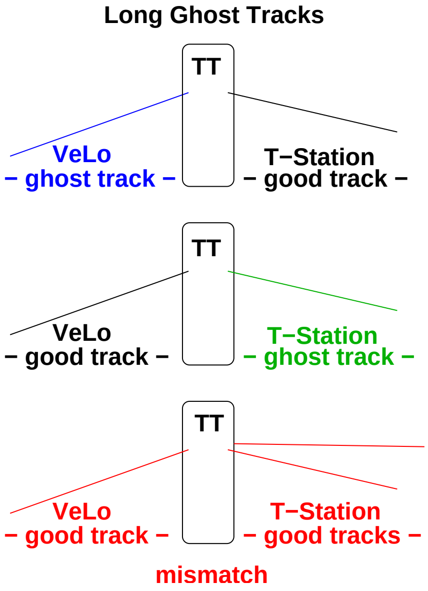

Figure 25: *Classification of ghost tracks.*

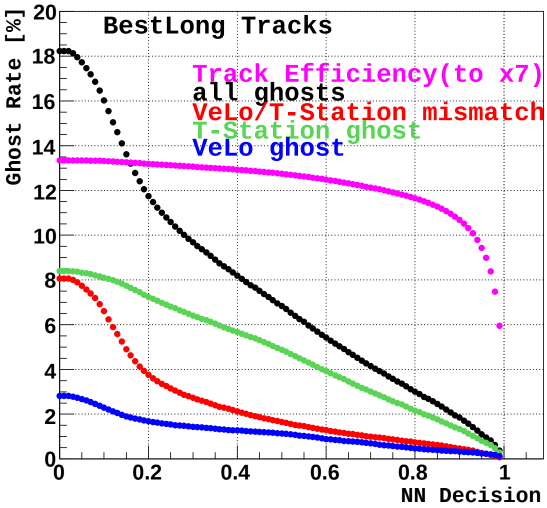

Figure 26: *Ghost types Rate evolution as a function of the NN decision. The corresponding Track Efficiency, down-scale factor 7, of the NN decision is illustrated by the magenta curve.*

A classification of the ghost tracks is illustrated in Fig. 25.

Three major types of ghosts can be seen:

– ghost which have the VeLo track segment not associated to a MC particle;

– ghost that are build with TS track segment identified as ghost;

– ghost made of good VeLo and TS track segments, but with a wrong matching between them.

The evolution of the ghost rate with the NN decision can be seen in Fig 26 for the three types of ghosts and the total rate.

The blue curve shows the ghost rate evolution for tracks with a ghost VeLo segment, the green one for tracks with a ghost TS segment and the red one for tracks that have the VeLo and the TS segments mismatched. The dominant contribution to the ghost rate belongs to tracks that have a ghost TS segment.

In order to have a better orientation in *Track Efficiency* vs. *Ghost Rate* space, in Fig. 26 is plotted also the track efficiency as a function of NN decision with a down-scaled factor of 7, the magenta curve.

8 Implementation in the LHCb Software

The tests presented until now were performed in Brunel version v31r7. In order to commission the NN as a tool to LHCb software an additional test is needed in the Brunel version v31r9.

A comparison between the NN performances in the two Brunel versions can be seen in Fig. 27 for track efficiency and in Fig. 28 for physics track signal efficiency. Improvements of the track fit and

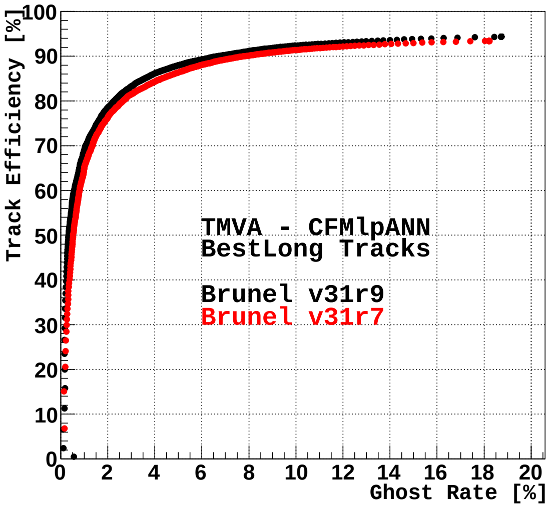

Figure 27: *Track Efficiency vs. Ghost Rate for different Brunel versions.*

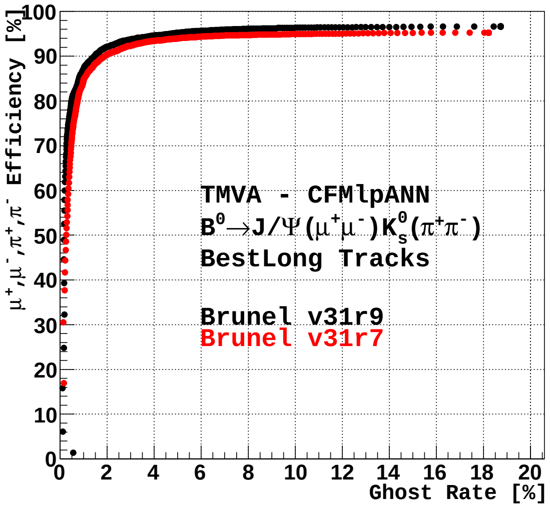

Figure 28: *Physics Signal Tracks Efficiency vs. Ghost Rate for different Brunel versions.*

algorithms make possible a better starting value for the track efficiency. Due to these improvements the NN performance is systematically better than in the previous Brunel version.

In Brunel v31r10 the algorithm which calculates the NN decision is implemented in the package Tr/NNTools/v1r0. The NN decision is calculate with the TVMA package from ROOT. The first version has only the CFMlpANN method from TMVA implemented. The NN which is implemented in Brunel v31r10 does not use the $N_{tracks/ev.}$ variable [9].

The NNTools returns the **Ghost Probability** of a track after NN evaluates the track with respect to the input variables. The evaluation is performed with the CFMlpANN method and the corresponding weights file.

The **Ghost Probability** is stored afterwards as additional information to the track and written to .dst files if these are produced.

The user can run the NNTools in Brunel editing in RecoTracking.opts file the following lines:

```
Trac.DetectorList={ "PatForward", ...,
                    "NNTools"
                  };

//NeuralNetTmva ghost probability
TrackNNToolsSeq.Members += {NeuralNetTmva};
```

In DaVinci framework the access to **Ghost Probability** is as follows:

```
float gp = protoparticle->track->info(LHCb::Track::GhostProbability,
                                      9999);
```

A second version Tr/NNTools/v1r1 is planned to have options files which should offer to the user a higher degree of freedom. As options are forseen different methods from TVMA package and own-user trained weights files. In addition, a tool for the TMultiLayerPerceptron from ROOT will be available and one for writing NTuples with various variables.

More details about the implementation in the LHCb software can be found at the address:
http://twiki.cern.ch/twiki/bin/view/LHCb/LHCbNeuralNet .

9 Conclusion

The purity of the tracks sample found by the pattern recognition algorithms is highly improved when a neural network which combines various track variables is used.

Variables as $P(\chi^2)$, n.d.f., N_{TThits}, $N_{VeLohits}$, η, $N_{tracks/ev.}$ are characteristics to every track and have a discrimination power between good tracks and ghost tracks.

To improve the neural network performance, additional variables as number of candidates of a track with common Outer Tracker segment, χ^2_{Match}, $\Delta\chi^2$ for Match tracks and $Q_{PatForward}$, $\Delta Q_{PatForward}$ for PatForward tracks, are added as input to the neural network. These additional variables are built inside the pattern recognition algorithms.

The performances of the network show that one can use various neural network decision values for tracks selection needed for analysing a certain subject. For the best long tracks one can choose to have for the track sample a *Track Efficiency* of 90% and a *Ghost Rate* of 8% instead of 94% and 18.5%, respectively, as given by the algorithm.

Tests on physics signal tracks, opposite side tagger tracks and electron tracks indicate a stabile neural network performance.The results of the neural network on events at high luminosity and tests performed on different B meson MC samples demonstrate that the neural network is robust. Once the neural network is trained on one B meson MC sample is could be used for evaluation also on MC samples with different B mesons.

First implementation of the neural network software for track evaluation is done in Brunel v31r10 under Tr/NNTools/v1r0 and has the CFMlANN method from TMVA package available. The NNTools return a **Ghost Probability** which is stored as additional information to the track and has an easy access in the DaVinci framework.

A second version of the NNTools is foreseen where more methods from the TMVA should be made accessible to users via option files.

References

[1] **LHCb** Collaboration, “Lhcb reoptimized detector design and performance technical design report”,. CERN-LHCC/2003-030. 6

[2] M. Needham, “Performance of the Track Matching.” LHCb-2007-129. 6

[3] O. Callot and S. Hansmann-Menzemer, “The Forward Tracking : Algorithm and Performance Studies.” LHCb-2007-015. 6, 10

[4] http://root.cern.ch/root/html310/TMultiLayerPerceptron.html. 6

[5] http://root.cern.ch/root/htmldoc/TMVA_Index.html. 6

[6] http://lhcb-release-area.web.cern.ch/.../class_track_event_clone_killer.html. 6

[7] https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbNeuralNet. 9, 10

[8] M. Needham, "Combined Long Tracking Performance." LHCb-2007-019. 10

[9] http://indico.cern.ch/.../access?contribId=2&resId=0&materialId=slides&confId=10174. 16