

A neural network based dedicated boosted Higgs b-tagging algorithm in CMS

Master Thesis

Mathematisch-Naturwissenschaftliche Fakultät
der
Universität Zürich

Thea Klaeboe Arrestad



Prof. Dr. Benjamin Kilminster
Dr. Clemens Lange
Dr. Andreas Hinzmann

Zürich
October 2014

Abstract

In the new energy regime explored by the Large Hadron Collider new heavy resonances suggested by extensions of the Standard model become available for production and can be detected with the CMS detector. Of recent interest are resonances that decay into Higgs bosons, leading to final states with highly energetic (“boosted”) Higgs bosons that decay dominantly to b quarks. Due to the high momentum of the Higgs boson, the b quarks from its decay are highly collimated, such that they cannot be resolved by the jet algorithm and are merged into a single jet. Current methods to identify jets originating from a b quark decay (“b-tagging”) fail in the case when the two b quarks are too collimated, because they are not treating the overlap of the tracks initiated by the two b quarks correctly. A dedicated b-tagging algorithm for jets containing two b quarks is therefore needed. This thesis presents the first b-tagging algorithm in CMS capable of tagging jets where two b quarks are merged into one jet. It is capable of discriminating between jets containing one b quark and jets containing two, as well as discriminating two-b jets from light-flavoured jets. The algorithm also provides discrimination between two-b jets coming from a boosted Higgs boson and two-b jets stemming from the QCD background. The presented algorithm achieves a 30 and 10% higher identification efficiency for highly boosted Higgs bosons than the current methods at same mistagging probability for two-b versus one b jet and two-b versus light-flavoured jet discrimination respectively.

Contents

1	Introduction	9
2	Bottom quark physics	11
2.1	The Standard Model bottom quark	11
2.1.1	Production and decay modes	11
2.1.2	Properties	12
2.1.3	Coupling to the Higgs boson	13
2.1.4	New heavy resonances decaying to Higgs Bosons	13
3	The LHC and the CMS Detector	17
3.1	The Large Hadron Collider	17
3.2	The CMS detector	17
3.2.1	Conventional coordinate system	17
3.2.2	Inner Tracking System	18
3.2.3	The Calorimetric System	19
3.2.4	The Muon System	20
4	Event reconstruction	23
4.1	Track reconstruction	23
4.2	Primary vertex reconstruction	23
4.3	Secondary vertex reconstruction	24
4.3.1	The Inclusive Vertex Finder	24
4.4	Particle Flow algorithm	26
4.5	Jets	26
4.5.1	The Cambridge-Aachen Algorithm	26
4.5.2	The anti- k_T algorithm	27
5	Samples and event selection	29
5.1	Monte Carlo samples	29

5.2 Flavour definition in CMSSW: A new bb flavour	29
5.3 Event selection	30
6 b jet identification in CMS	33
6.0.1 Algorithms	33
6.0.2 Backgrounds	37
6.0.3 b-tagging in boosted topologies	37
7 Dedicated bb vs. b training	41
7.1 Artificial Neural Network	42
7.1.1 Multilayer Perceptron	42
7.2 The double b-tagger training	43
7.2.1 Framework	43
7.2.2 Training preparation	43
7.2.3 p_T - η reweighting	47
7.2.4 Training bins	48
7.2.5 Biasing	48
7.2.6 Variable preprocessing	50
7.2.7 Classification	51
8 Results	55
8.1 Validation of training framework	55
8.2 Performance of the double b-tagging algorithm	56
8.2.1 bb versus b-flavoured jets	57
8.2.2 Training with different mass points	57
8.2.3 Varying p_T - η training bins	58
8.2.4 p_T -dependence	59
8.2.5 bb versus light-flavoured jets	60
8.2.6 Performance on larger jets	61
8.2.7 bb versus b for gluon splitting	62
8.2.8 $g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$	62
8.2.9 $H \rightarrow b\bar{b}$ versus b jets from QCD	64
8.2.10 Adding additional vertex training categories	65
9 Conclusion	69
A The MVA Trainer and MVAComputer	71

A.1	Implementation of additional vertex categories	71
A.2	Available processors	73
A.3	Biases	74
A.4	$g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$ additional plots	74
A.5	Technical training description	75
A.6	List of samples	77
	Bibliography	79

Chapter 1

Introduction

In analyses of highly energetic (“boosted”) Higgs bosons or searches for new heavy resonances with b quarks in the final state, the final decay products will be very collinear and can be hard to separate. If the two b jets are too close to one another for the jet algorithm to resolve, they will be merged into one single jet. This can for instance be the case for Higgs pair production mediated by a Radion (Figure 1.1) predicted in warped extra dimension theories (see Section 2.1.4) [1]. It is therefore important to be able to identify these jets and separate them from

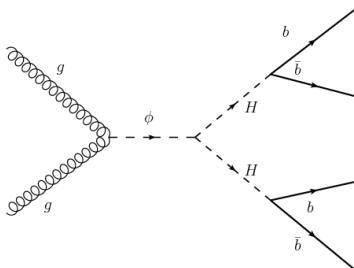


Figure 1.1: Higgs pair production mediated by Radion (ϕ) in warped extra dimension theories.

jets containing a single b quark. The b-jet identification itself, called b-tagging, is based on the production and subsequent decay of b hadrons. The long lifetime of b hadrons (corresponding to $c\tau \approx 450 \mu m$) leads to secondary vertices displaced from the primary vertex. These are detectable with the CMS inner tracker. The high mass of the b hadrons and large multiplicity of charged particles in the final state also separates b jets from light-flavoured quark- and gluon-initiated jets. For boosted topology events, jet substructure techniques where the jet is reclustered into its constituent subjets, are often used [2][3]. However, as the boost of the mother particle increases, the distance between the jet axes of the two subjets decreases ($\Delta R \propto 2m/p_T$, where m is the mass of the mother particle and p_T is its transverse momentum) and separating them becomes non trivial due to shared tracks and correlations that are difficult to model. To avoid the limitations imposed by track sharing complications, tagging the jet as a whole without splitting it into subjets becomes feasible to reach a higher b-tagging efficiency in the high- p_T region. Tagging the jet in this way however, disregards information of whether the jet contains one b quark or two. A dedicated algorithm designed to separate these two cases is therefore needed. This thesis introduces the first dedicated boosted b-tagging algorithm in CMS capable of discriminating jets containing one b quark from jets containing two b quarks. It is based on multivariate analysis (MVA) techniques combining variables from displaced tracks and secondary vertices using a multilayer perceptron neural network.

The thesis is organized as follows: A theoretical introduction to b quark physics and heavy resonance searches with b quarks in the final state is given in Chapter 2. Then a description of the CMS detector, with focus on the inner tracking system is presented in Chapter 3, followed by the event reconstruction in Chapter 4. The samples and selection criterias are presented in Chapter 5 and a brief overview of current b-jet algorithms used in CMS is given in Chapter 6. The dedicated double b-tagger training is explained in Chapter 7, with the achieved results given in Chapter 8 and conclusions in Chapter 9.

Chapter 2

Bottom quark physics

2.1 The Standard Model bottom quark

In the six-quark Standard Model of the strong, weak and electromagnetic interactions, quarks are predicted to exist in three left-handed doublets

$$\begin{pmatrix} u \\ d \end{pmatrix}_L, \begin{pmatrix} c \\ s \end{pmatrix}_L, \begin{pmatrix} t \\ b \end{pmatrix}_L$$

and right-handed singlets of the weak SU(2) gauge group. The bottom (b) quark with its mass of about 4.2 GeV [4], is the heaviest of the down type quarks. It belongs to the third generation of quarks as the weak isospin partner of the top. The existence of the b quark was predicted when a third quark doublet was proposed by Kobayashi and Maskawa in their model of quark flavour mixing [5, 6], and was discovered in 1977 with the observation of the 9.5 GeV bound $b\bar{b}$ state, Υ , at Fermilab [7].

The weak eigenstates and mass eigenstates of the quarks are related to one another via the 3×3 CKM matrix through 3 different mixing angles and a CP violating phase:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix}_L = \begin{pmatrix} |V_{ud}| & |V_{us}| & |V_{ub}| \\ |V_{cd}| & |V_{cs}| & |V_{cb}| \\ |V_{td}| & |V_{ts}| & |V_{tb}| \end{pmatrix} = \begin{pmatrix} c_1 & -s_1c_3 & -s_1s_3 \\ s_1c_2 & c_1c_2c_3 - s_2s_3e^{i\delta} & c_1c_2s_3 + s_2c_3e^{i\delta} \\ s_1s_2 & c_1s_2c_3 + s_2s_3e^{i\delta} & c_1s_2s_3 - c_2c_3e^{i\delta} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}_L$$

Here $c_i = \cos \theta_i$ and $s_i = \sin \theta_i$, θ_i are the different mixing angles and $e^{i\delta}$ is the CP violating phase (same for up-type quarks). Due to the off-diagonal terms in this matrix, there exists a mixing between the quark generations, with the probability of transition from one quark to another given by the square of the corresponding matrix element. As the top quark is significantly heavier than its doublet partner, the b has to decay to quarks from the other generations, namely the second generation c or the first generation u quark. Due to rather small values of θ_2 and θ_3 , the probability for b quark transition across doublets is small. This leads to a significantly longer lifetime for the b quark than the other quarks.

2.1.1 Production and decay modes

In Quantum Chromodynamics (QCD), the quantum field theory describing the strong interaction between quarks and gluons, quarks can never be observed as free particles due to a phenomenon called *colour confinement* [8]. Colour confinement is due to the fact that the force between

quarks increases as they get pulled further apart from one another. As a consequence no coloured objects, like quarks and gluons, can exist as free particles and be observed in Nature. The quarks can, however, form colour singlet objects, such as hadrons and mesons, which we do observe. In most hadronic bound states containing a b quark, the b is accompanied by a light flavour (u, d, s) quark. All ground state b mesons decay via the weak interactions. As the b quark is quite heavy compared to the scale of the strong interactions (~ 1 GeV), the b meson decay can be approximated by the decay of a free b quark (spectator model) [9][10]. The dominant decay mode of the b quark is $b \rightarrow cW^{*-}$ where the off-shell W^* can decay hadronically ($q\bar{q}$) or leptonically ($\ell\nu$) in a so called semileptonic b decay. This can be seen in Figure 2.1. The main

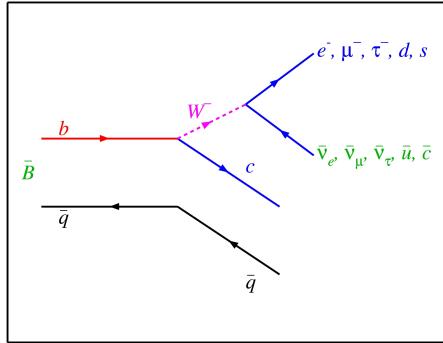


Figure 2.1: The b quark decays most of the time as $b \rightarrow cW^{*-}$, where the W can decay hadronically or leptonically. Illustrated here is a decaying B hadron where the accompanying quark is a spectator in the decay.

production channels of $b\bar{b}$ pairs in proton-proton collisions are shown in Figure 2.2. There are three processes at leading order: direct production (corresponding to the top four diagrams), flavour creation (bottom left diagram) and gluon splitting (bottom right) [12].

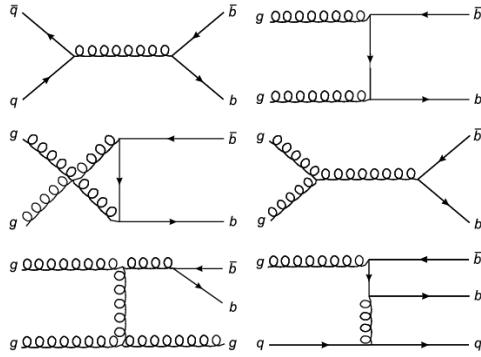


Figure 2.2: The main b quark production channel in a proton collider are direct production (top four diagrams), flavour creation (bottom left) and gluon splitting (bottom right) [12].

2.1.2 Properties

The lifetime of the b hadron is around $\tau = 1.5$ ps, causing it to travel some distance before it decays. The momentum spectrum of the produced b quarks typically peaks near the b mass with a tail to high momenta leading to typical decay lengths of around a millimeter. This is

increased if the decay products are boosted. Mass, lifetime and quantum numbers of the b quark are listed in Table 2.1.2.

b quark	
<i>Mass[GeV/c]</i>	4.2
<i>Lifetime[s]</i>	10^{-12}
<i>Q</i>	$-\frac{1}{3}$
<i>B</i>	$\frac{1}{3}$
<i>Y</i>	$-\frac{2}{3}$
<i>I</i> ₃	0
<i>I</i>	0

Table 2.1: Mass, lifetime and different quantum numbers of the b quark. Q is the electric charge, B is the baryon number, I_3 is the third component of the isospin and I the isospin [4].

2.1.3 Coupling to the Higgs boson

Through Yukawa interaction between the Higgs field and the massless fermion fields, fermions acquire a mass proportional to the Higgs vacuum expectation value through spontaneous symmetry breaking [14]. The Higgs coupling is proportional to the mass of the fermions and has a branching ratio given as

$$\Gamma(H \rightarrow f\bar{f}) = \frac{M_H}{8\pi} \left(\frac{M_f}{v} \right)^2 N_c \left(1 - \frac{4M_f^2}{M_H^2} \right)^{\frac{3}{2}}$$

where M_H is the Higgs mass, M_f is the fermion mass, v is the vacuum expectation value of the Higgs field and $N_c=3$ is the QCD colour factor. The interaction vertex of the Higgs coupling to fermions and the different branching fractions for a Standard Model Higgs boson with a mass of $M_H=125$ GeV, is illustrated in Figure 2.3. With the b quark being the heaviest available fermion, $H \rightarrow b\bar{b}$ is the dominant decay process with a branching ratio of almost 60%.

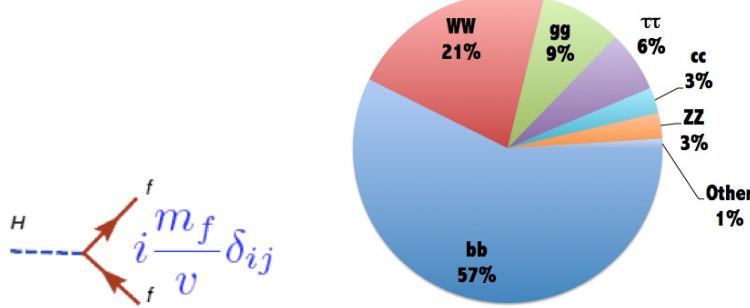


Figure 2.3: Feynman diagram of the Higgs coupling to fermions (left) and the branching ratios of the Standard Model Higgs boson with $M_H=125$ GeV (right). The coupling is proportional to the fermion mass and with the b quark being the heaviest available fermion, the Higgs boson preferably decays into b quarks.

2.1.4 New heavy resonances decaying to Higgs Bosons

With the discovery of a Higgs boson with a mass of 125 GeV [13], the question of its mass stability has become pressing. To avoid delicate levels of fine-tuning in the Standard Model,

new heavy particles must exist in order to stabilize its mass. Many interesting Beyond Standard Model (BSM) theories offer solutions to this problem and predict new heavy resonances coupling to Higgs bosons. Among these are so called composite Higgs models [18]. In the Standard Model, the Higgs boson gains mass through quantum corrections from fermion loops. The Higgs mass is a sum of the bare mass, M_0 and higher order mass correction terms

$$M_H^2(125 \text{ GeV}) = M_0^2 + \delta M_H^2, \quad \delta M_H^2 = -2 \frac{|\gamma_f|^2}{16\pi^2} \Lambda^2 + \dots \quad (2.1)$$

where Λ is finetuned to be at the order of the gravitational scale M_{Pl} . In composite Higgs models, the Higgs is considered a pseudo Nambu-Goldstone boson that couples to the SM particles and to new heavier gauge bosons, such as Z' and W' , with masses in the TeV region. It is a composite state of a new strong interaction and the cut off Λ is set by the scale of the new dynamics. This solves the hierarchy problem, as corrections to the Higgs mass are screened above $1/l_H$, where l_H is the diameter of the composite state. Figure 2.4 shows an example of a heavy W' decaying to a W and a Higgs boson.

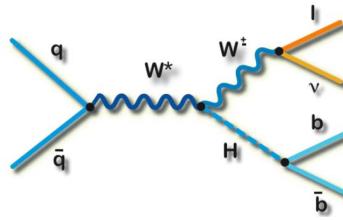


Figure 2.4: Feynman diagram of the production of a new heavy W' boson predicted by composite Higgs models. Here decaying to a W and a Higgs boson.

Other BSM theories predicting new heavy resonances decaying to Higgs bosons are theories of warped extra dimensions (WED) [1]. WED models attempt to explain the large difference between the weak scale $M_W \sim 100$ GeV and the Planck mass (mass scale that weakens the gravitational interaction) $M_P \sim 2.4 \cdot 10^{18}$ GeV. In the original WED proposal [15, 16] this is done by introducing one extra warped finite spatial dimension with a non-trivial metric. Here the gravitational interaction can propagate and effectively modify gravitational behaviour at short distances. The 5D space between the two branes is called the bulk and is illustrated in Figure 2.5. Fluctuations around the infinite part of the metric correspond to a 5D spin 2 Graviton

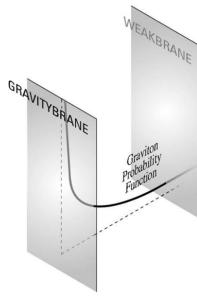


Figure 2.5: Dimensions in RS theory. The Weak (TeV) brane and Gravity (Planck) brane are 4D boundaries of the extra dimension which is a line segment confined on an interval $0-\pi$. The Graviton probability function illustrate the behaviour of the metric along the extra dimension [1].

field, and fluctuations around the finite part give rise to a spin-0 Radion field. Excitation modes on our 4D brane manifest themselves as heavy resonances, called Kaluza-Klein modes, with calculable masses. In order to solve the mass hierarchy problem between the weak and the Planck scale, the first excitation modes are expected to be at the TeV scale. The presence of trilinear terms in the Higgs Radion (ϕ) sector opens up the possibility of $\phi \rightarrow HH$. With, for example, a Higgs mass $M_H=120$ GeV, Radion field expectation value $\Lambda_\Phi=5$ TeV and Radion mass between $M_R=250\text{-}350$ GeV, the branching ratio, $\text{BR}(\phi \rightarrow HH)$, ranges between 20 and 30% [17].

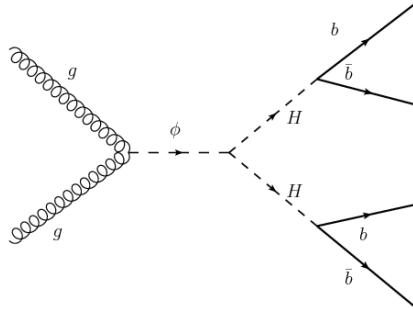


Figure 2.6: Higgs pair production mediated by a Radion(ϕ) in warped extra dimension theories.

Chapter 3

The LHC and the CMS Detector

3.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is a 27 km long particle collider based at the European Organization for Nuclear Research (CERN) laboratory on the border of Switzerland and France [19]. It is situated underground and is designed to accelerate two proton beams up to an energy of 7 TeV each before bringing them into collision, leading to a total center-of-mass collision energy of $\sqrt{s}=14$ TeV. The LHC was built by a joint collaboration of scientists and engineers from over a hundred different countries, with the main goal of proving the existence of the long sought after Higgs boson [14] and to test a variety of physics theories including the Standard Model. The proton beams are collided at four points around the ring, corresponding to the location of the four large particle detectors ATLAS [20], CMS, ALICE [21] and LHCb [22].

3.2 The CMS detector

The Compact Muon Solenoid (CMS) detector is a general purpose detector located near Cessy in France and is designed to investigate a wide range of physics. It is constructed around a superconducting solenoid coil, providing a homogeneous solenoidal magnetic field of 3.8 T. The detector consists of a tracker, electromagnetic and hadronic calorimeters and a muon system. The tracker and calorimeters are located inside the coil where the high magnetic field allows precise measurements of the particles momenta, while the muon system is placed on the outside of the magnet. In the following, the CMS inner tracking system and calorimeters are briefly described as they comprise the most crucial part of the CMS detector for b quark jet identification. A full description of the CMS detector can be found elsewhere [23].

3.2.1 Conventional coordinate system

The coordinate system used to describe the CMS detector in this thesis is adopted from the conventional global CMS coordinate system [23]. It is a right-handed system with the origin located at the beam interaction point in the center of the detector and the z axis pointing tangentially along the beamline. The x axis points directly towards the center of the LHC ring and the y axis vertically upwards. In the transverse plane, coordinates are described using the azimuthal angle ϕ , while the polar angle θ measures the inclination with respect to the z axis. The polar angle is usually replaced by the pseudorapidity $\eta = -\ln \tan \theta/2$ which is a Lorentz invariant quantity (Figure 3.1). To describe distance between objects in the detector the variable $\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2}$, with ϕ and η defined as before, is often used.

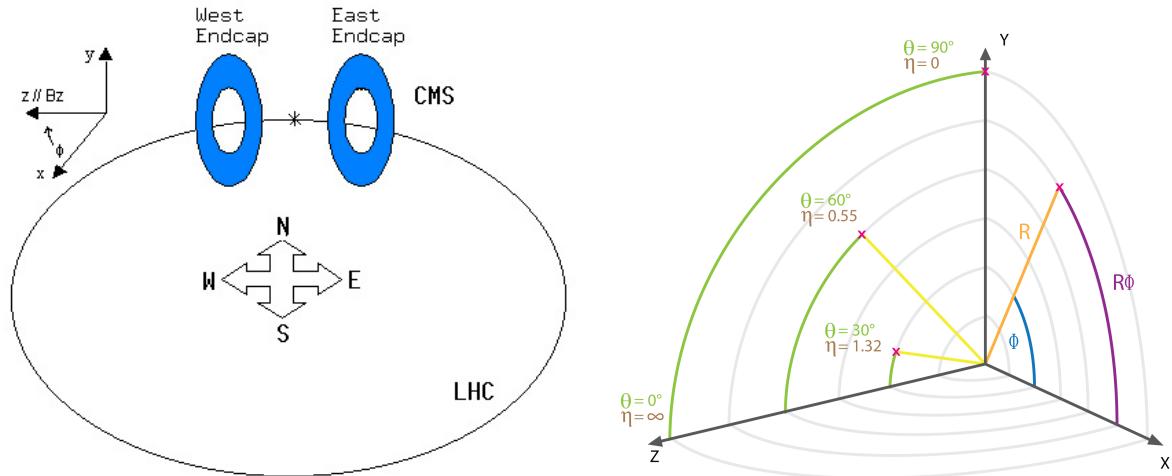


Figure 3.1: The CMS conventional coordinate system.

3.2.2 Inner Tracking System

The long flight path of the b hadron leads to displaced tracks and secondary vertices distinguishing it from other quark and gluon initiated jets. Good reconstruction of charged particle tracks close to the interaction point is therefore crucial. This is provided by the inner tracking system. The CMS inner tracker is a silicon detector fully embedded in the 3.8 T magnetic field. It consists of an inner pixel detector for precise spatial measurements close to the beam pipe and an outer silicon strip detector providing multiple measurement points for momentum calculation. The structure of the tracking system is shown in Figure 3.2 and the different subsystems are described in the following.

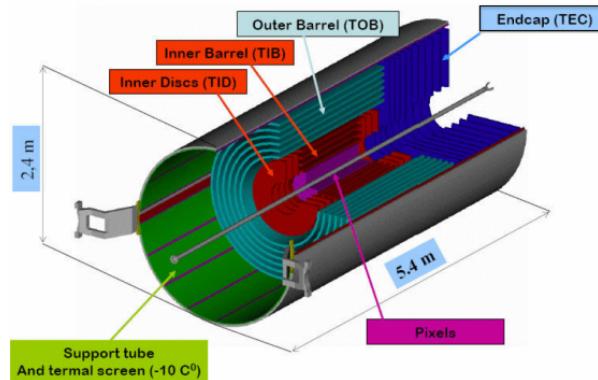


Figure 3.2: The CMS inner tracker consists of an inner pixel detector and an outer silicon strip detector surrounding the interaction point with a coverage up to $|\eta| = 2.5$. The thirteen barrel layers provide few, but precise measurement points [23].

The Silicon Pixel Detector

The detector part closest to the LHC beampipe is pixelated and has three barrel layers and two disk caps, the so called pixel detector. It provides three spatial measurements used for the primary vertex reconstruction and the trajectory seeding of charged particles. High resolution

is therefore crucial. The pixel detector consists of two parts; the barrel pixel detector (BPIX) and the forward pixel detector (FPIX), as can be seen in Figure 3.3. The BPIX consist of three

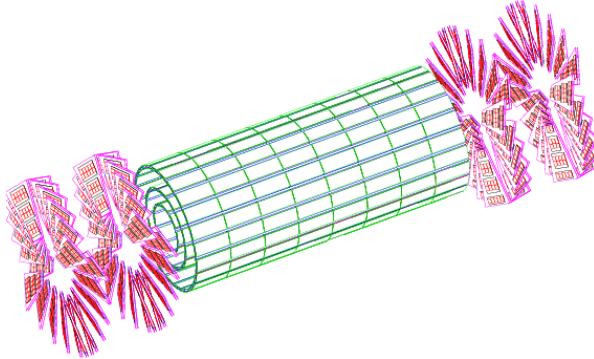


Figure 3.3: The CMS pixel detector. The forward (FPIX) detector is shown in red and the barrel (BPIX) in green.

half meter long cylinders, the first layer being 4.4 cm and the last 10.2 cm from the beam axis. Each cylinder contains several detector units, “modules”, which hold the silicon sensor, readout chips and electronic components. The silicon sensors are 285 μm thick and segmented, where each pixel cell has a size of 100 μm in the $R\text{-}\phi$ plane and 150 μm along the z axis. The FPIX is made of two disks at both ends of the BPIX detector. The discs have a radii of 6 and 15 cm and are mounted at $z=\pm 34.5$ cm and $z=\pm 46.5$ cm. The hit resolutions of the pixels are between 9 and a few tens of μm depending on the pixel location [24].

The Silicon Strip Tracker

The ten outer barrel layers of the tracker consist of silicon micro-strip sensors partially mounted in stereo geometry [25]. The main purpose of the silicon strip tracker is to estimate the momentum of charged particles from the track curvature in the magnetic field, and hence requires multiple measurements spread over a large distance. It covers the region between 20-116 cm and consists of three parts: the Tracker Inner Barrel (TIB) + Disks(TID), the Tracker Outer Barrel (TOB) and the Tracker Endcaps (TEC \pm). These are illustrated in Figure 3.4. The TIB extends out to 55 cm and consists of 4 layers with 3 TIDs at each end. Following the TIB is the 6-layer TOB with a coverage up to $z < \pm 118$ cm. The additional longitudinal coverage is provided by the TECs consisting of 9 rings and a coverage of $124 \text{ cm} < |z| < 282$ cm. The endcaps have a radius between 22.5 and 113.5 cm. Different silicon micro-strips are used for different parts of the detector to account for flux and resolution requirements. The resolution ranges between 15 and 40 μm for the different subdetectors [34].

3.2.3 The Calorimetric System

Electromagnetic calorimeter (ECAL)

The CMS electromagnetic calorimeter is a lead-tungstate (PbWO_4), segmented scintillating detector [26]. It consist of a barrel and two endcaps and is used to measure the energy of photon and electron showers. The energy resolution of the ECAL goes roughly as $1/E^2$ where the relative resolution can be parametrized as

$$\left(\frac{\sigma^2}{E}\right)^2 = \left(\frac{S^2}{\sqrt{E}}\right)^2 + \left(\frac{N^2}{E}\right)^2 + C^2 \quad (3.1)$$

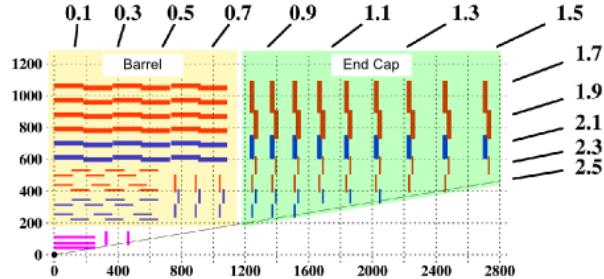


Figure 3.4: The position of the detector modules in the r-z plane. The barrel layers are mounted in stereo geometry with an angle of 5.7° .

where S is the stochastic, N the noise and C the constant term which are measured in electron test beams. The energy resolution as measured in 2013 for electrons from Z-boson decays is better than 2% in the central region of the ECAL barrel (for $\eta < 0.8$) and is 2–5% elsewhere [27].

Hadron calorimeter (HCAL)

Located between the ECAL and the magnet, is the CMS hadronic calorimeter [28]. The HCAL is used to measure the energy of hadronic showers. It is a sampling calorimeter made of brass absorber material and layers of plastic scintillator as detector material. The resolution of the hadronic calorimeter goes roughly as $1/\sqrt{E}$ [29].

3.2.4 The Muon System

The muon system [30] is the outermost part of the CMS detector and consists of a barrel and two endcaps. It uses three different types of gas detectors: regular drift tubes (DT), cathode strip chambers (CSC) and resistive plate chambers (RPC). Drift tubes are used for the barrel part as the magnetic field is weaker and the trigger rate is lower here than for the endcaps, where more tolerant cathode strip chambers are implemented. Both barrel and endcaps use resistive plate chambers to provide accurate timing information. The spatial resolution is around 200 μm for both the drift tubes and the cathode strip chambers. An overview of the muon system is shown in Figure 3.5.

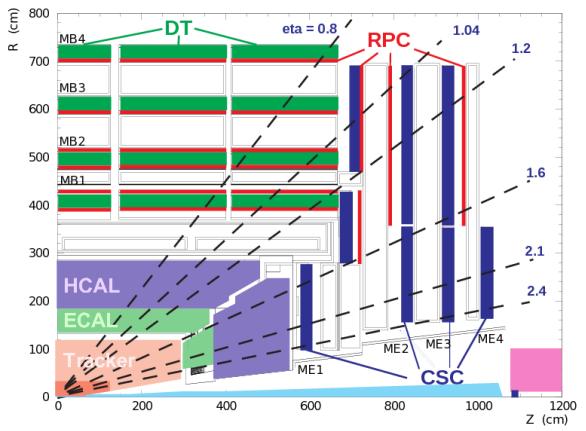


Figure 3.5: The CMS muon system. The barrel part implements drift tubes (DT) for spatial measurements and resistive plate chambers (RPC) for timing information. The two endcaps uses cathode strip chambers (CSC) and RPCs [30].

Chapter 4

Event reconstruction

Collision data from the most interesting CMS events is stored as a set of signals containing spatial information, timing, pulse heights etc. In order to turn this information into actual physics object, complex algorithms are used. In the case of b-tagging, the most important objects are tracks and jets. The reconstruction of these are described in the following.

4.1 Track reconstruction

Track reconstruction is performed by the Combinatorial Track Finder (CTF) algorithm [31]. It starts by finding a hit in the pixel detector that has a high signal-to-noise ratio and uses that as a seed to search for other hits to cluster into it, building the first track candidates. If the charge deposit of the cluster (2-3 hits) is above some threshold, the hit position is determined by a sophisticated algorithm, and a pattern recognition algorithm extrapolates the seed trajectories outwards searching for other hits using the Kalman Filter Formalism [32]. A track fitting scheme then refits the trajectories to eliminate biases introduced by the seeding and then smoothens the trajectory by a second fit in reverse order (inwards, towards the beam). The track reconstruction efficiency depends on the track p_T and η , as well as the precision of the hit position measurements. The average track reconstruction efficiency measured using $t\bar{t}$ events with normal pile-up conditions for charged particles with transverse momenta $p_T > 0.9$ GeV is 94% for pseudorapidities of $|\eta| < 0.9$ and 85% for $|\eta|$ between 0.9 and 2.5 [24]. For higher values of $|\eta|$ the lever arm of the measurement is reduced and the measurement less precise.

4.2 Primary vertex reconstruction

The position of the interaction point is determined from the reconstructed tracks. To reconstruct the primary vertex (PV) the tracks to be considered are selected from requirements on their point of closest approach with respect to the beam line and the total number of tracker hits. The tracks are then clustered into vertex prototypes by a vertex finding algorithm using a χ^2 minimization procedure. Finally the vertex prototypes are fitted to determine the position and uncertainty of the vertex collection using the Adaptive Vertex Fitter (AVF) [33]. The PV with the highest sum of track p_T is selected. The primary vertex position resolution improves roughly with the number of tracks as $1/N_{tr}$ and is around 10-12 μm in each of the three spatial dimensions for high track multiplicities (at least 50) [24].

4.3 Secondary vertex reconstruction

4.3.1 The Inclusive Vertex Finder

Good reconstruction of secondary vertices is essential for b-tagging. As previously mentioned, in cases where parton pairs are merged into a single jet due to their small angular separation (eg. boosted Higgs $\rightarrow b\bar{b}$) the two partons can be difficult to disentangle. Figure 4.1 shows the true angular separation between all generated b quark pairs in a QCD Monte Carlo sample and the same distribution for events where the b hadron pair produces reconstructed jets. A large inefficiency in the collinear region (small $\Delta\Phi$) is observed [34]. The Inclusive Vertex finding

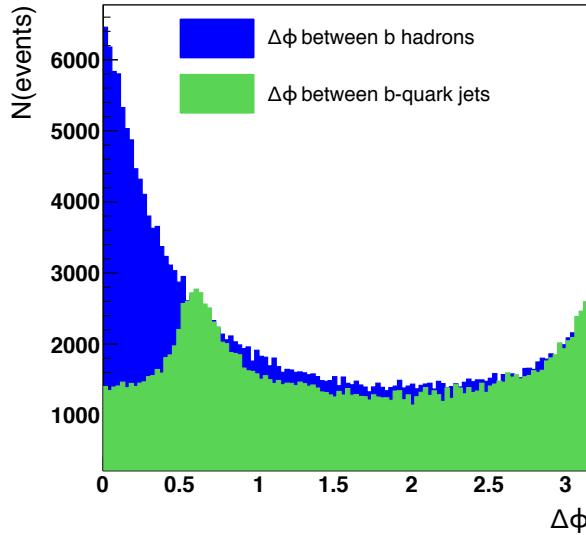


Figure 4.1: True transverse separation between all generated B hadron pairs (blue) and distance between the corresponding B hadrons that have reconstructed jets in the event (green). At small angular separations, there is a significant drop in reconstruction efficiency [34].

algorithm (IVF) is an algorithm designed to find secondary vertices completely independent of jets [34]. It uses all tracks in the event and looks for those that are highly displaced. Only tracks with > 8 valid hits in the tracker and $p_T > 0.8$ GeV are used. The tracks with 3D impact parameter (point of closest approach with respect to the jet axis) $> 50\mu\text{m}$ and impact parameter significance (impact parameter divided by its error) > 1.5 are used as seeds for the reconstruction. It then uses each seed and searches for additional tracks to cluster under the hypothesis of a common origin. The compatibility is computed using distance and angular separation variables. Clusters of more than 30 tracks are disregarded as they most likely correspond to the Primary Vertex. Each set of tracks are then fitted using the Adaptive Vertex Finder. To avoid track sharing between vertices, they are required to be separated with IP significance > 2 and have less than 70% of the tracks in common. The angular resolution of the IVF algorithm can be seen in Figure 4.2, where the difference in true distance between the simulated b hadrons and the reconstructed potential b candidates is plotted. It is observed to be around 0.02 for both ΔR and $\Delta\Phi$. This is a large improvement compared to the reconstruction resolution seen in Figure 4.1, where the efficiency falls dramatically for $\Delta\Phi$ values below 0.7. For the SV reconstruction additional track requirements are applied. Only high purity tracks fulfilling these requirements are selected:

- $\Delta R(p_{jet}, p_{track}) < 0.3$, where p_{jet} and p_{track} is the jet direction and track momentum respectively

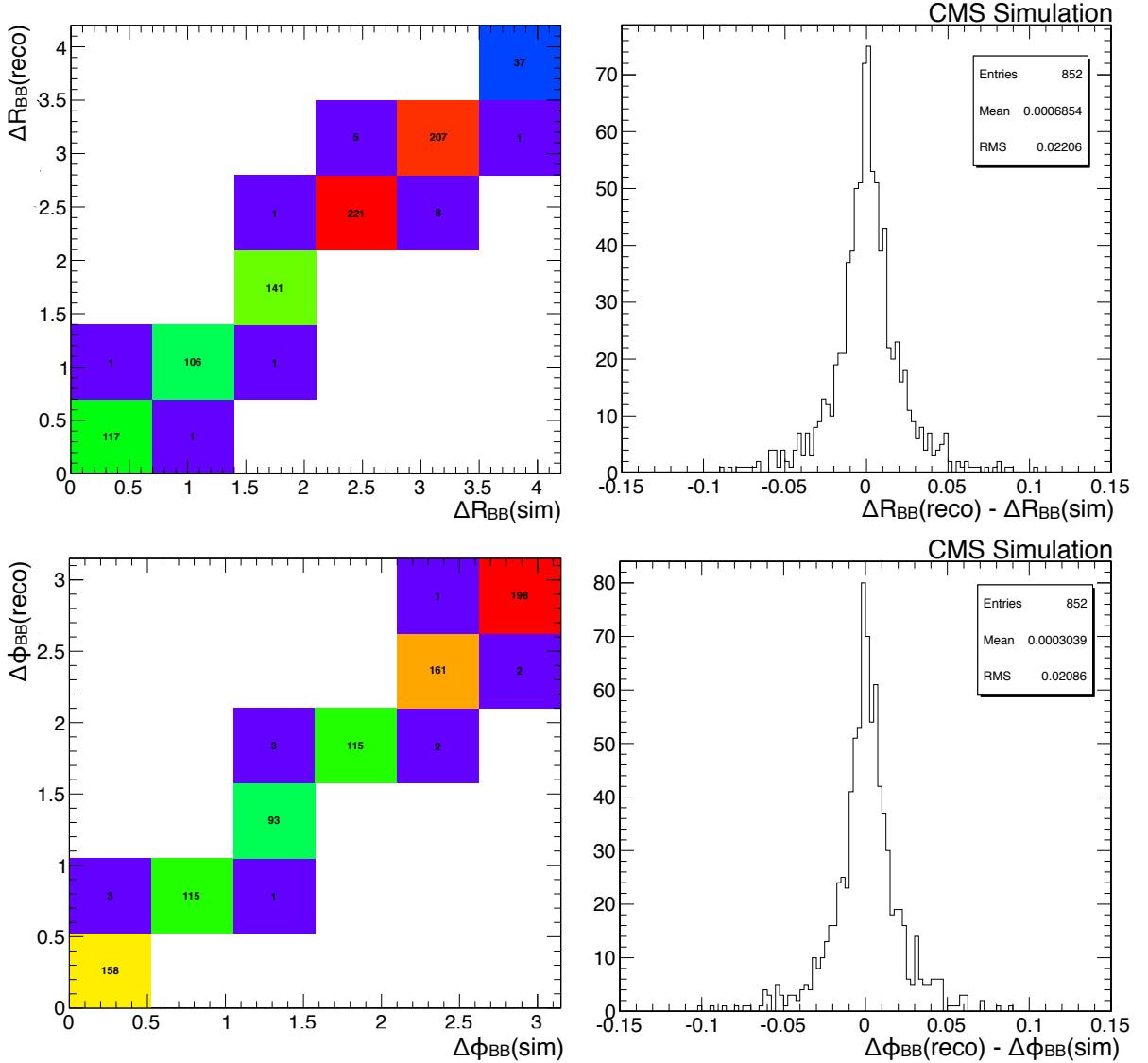


Figure 4.2: Angular resolution of the IVF. The top plots show the correlation and projection between true ΔR of the b hadrons and the reconstructed. The bottom plots show the same for $\Delta\phi$. The 2D/3D resolution is around 0.02 [34]

- Distance between the track and the jet axis is less than 0.2

The reconstructed Secondary Vertices are then filtered through the requirements:

- $|M_{vertex} - M_{K_0}| > 0.05$ GeV, where M_{K_0} is the Kaon mass.
- Number of tracks ≥ 2
- $M_{vertex} < 6.5$ GeV
- $\Delta R(\text{vertex, jet axis}) < 0.3$
- Fraction of tracks shared with PV < 0.65

- $0.01 \text{ mm} < 2\text{D vertex flight distance} < 2.5 \text{ mm}$, where 2D flight distance is the transverse distance between the primary and secondary vertex
- 2D vertex flight significance $> 3.0 \text{ mm}$

4.4 Particle Flow algorithm

To further reconstruct and identify long lived particles in CMS events, information from all the subdetectors is combined using the Particle Flow algorithm [35]. Particle Flow reconstructs particle objects by linking the tracks reconstructed in the tracker to the calorimetric depositions and muon chambers, providing excellent measurements of the particles energy and momentum. A link algorithm connects information from the different subdetectors by using the geometrical distance between tracks in the tracker and calorimeter deposition or a χ^2 minimization of the global fit in the case of tracker tracks and muon chamber depositions. Charged hadrons are built by connecting tracks to calorimeter deposits and applying a matching requirement between the track momenta and energy deposit in the calorimeter.

4.5 Jets

Gluons and quarks are never visible as final state particles, but hadronise into a collection of collimated hadrons called jets (see Section 2.1.1). Measuring the jets direction and momentum gives information on the original parton. There exist different schemes for how to cluster a group of particles into a jet, where all of them include some parameter defining the minimum distance between two particles in order to be in the same jet and a procedure of how to combine the momentum of the particles. The most used jet algorithms in CMS are the Cambridge-Aachen algorithm and the anti- k_T algorithm [36]. They are both based on sequential recombination of particle pairs using longitudinally invariant variables to cluster Particle Flow particles into jets. The benefit of using sequential recombination schemes over cone algorithms is that soft/collinear particles have a smaller impact on the reclustering procedure, making the algorithms infrared and collinear safe. Both algorithms use distance measures between particle i and particle j that can be generalized as

$$d_{ij} = \min(p_{ti}^{2k}, p_{tj}^{2k}) \frac{\Delta R_{ij}^2}{R^2}, \quad \Delta R_{ij}^2 = (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2, \quad (4.1)$$

$$d_{iB} = p_{ti}^{2k} \quad (4.2)$$

where the parameter k is 0 for the Cambridge-Aachen algorithm and -1 for the anti- k_T algorithm, p_{ti} , η_i and ϕ_i are respectively the transverse momentum, pseudorapidity and ϕ coordinate of particle i and R^2 is the cut off parameter of the jet algorithm. d_{iB} is the distance between the particle and the beam, so if d_{iB} is smaller than d_{ij} , the particle is recombined with the beam. In order to remove particles from additional vertices, the charged tracks not associated with the primary vertex are removed (Charged Hadron Subtraction).

4.5.1 The Cambridge-Aachen Algorithm

With $p = 0$, $d_{iB}=1$ and $d_{ij} = \Delta R_{ij}^2/R^2$ making the Cambridge-Aachen clustering scheme energy independent and it only considers distances in the ϕ - η plane. It proceeds by computing d_{ij} for each particle pair, recombining the pair with the smallest ΔR_{ij}^2 and repeating the process until all objects in the event have $\Delta R_{ij}^2 > R$. The R parameter sets the size of the jet cone. Cambridge-Aachen jets are mainly used for larger jets.

4.5.2 The anti- k_T algorithm

The anti- k_T algorithm ($p = -1$) clusters hard particles first. The jet grows outwards starting from an initial hard seed and continues until no particles remain within the jet cut-off parameter R. It computes d_{ij} and d_{iB} for all particle pairs and finds the smallest. If it is d_{iB} , i is declared a final state jet and removed from the list of particles. If d_{ij} is smaller, the algorithm proceeds as above until no particles remain. The most used anti- k_T jets in CMS are AK5 (R=0.5) and AK8 (R=0.8).

Chapter 5

Samples and event selection

5.1 Monte Carlo samples

To be able to predict the response of the CMS detector to different physics scenarios, an accurate description of the event topology is needed. To acquire this one has to rely on computer simulated events called Monte Carlo samples. These are generated by first modelling the initial interaction between the quarks, the hard process, based on some theoretical model. Then the branching of the partons into other parton pairs, called showering and hadronisation, is simulated before the final state particles are sent through a full detector simulation to model the detector response.

The results presented in this thesis are obtained from Monte Carlo simulated samples passed through a full simulation of the CMS detector and reconstructed using the standard CMS reconstruction algorithms. The b and light-flavoured (cudsg) jets are taken from QCD multijet events generated with Pythia6 [37] at a center-of-mass energy of 8 TeV. This sample covers a wide range of jet transverse momentum from 20 GeV to 1800 GeV. For the bb jets (jets containing two b quarks), different samples are used; two Radion samples with resonance masses of $M_R = 1000$ GeV and $M_R = 1100$ GeV respectively. These are generated with Madgraph [38] interfaced with Pythia6 for showering and hadronisation. In these samples the Radion decays into a Higgs pair, which subsequently decays into b quarks. Performance studies have also been done for three Graviton samples of different mass points where $G(X \text{ TeV}) \rightarrow HH \rightarrow 4b$ and with Graviton masses of 1.0, 1.5 and 2.0 TeV. The samples used for the neural network training and validation are listed in Table 5.1.

Process	Mass point/ p_T range [GeV]	Generator
$R \rightarrow HH \rightarrow 4b$	1100	MadGraph/Pythia 6
$R \rightarrow HH \rightarrow 4b$	1100	MadGraph/Pythia 6
$G \rightarrow HH \rightarrow 4b$	1000	MadGraph/Pythia 6
$G \rightarrow HH \rightarrow 4b$	1500	MadGraph/Pythia 6
$G \rightarrow HH \rightarrow 4b$	2000	MadGraph/Pythia 6
QCD	$20 < p_T < 1800$	Pythia 6

Table 5.1: Samples used for the training and validation of the double b-tagging algorithm.

5.2 Flavour definition in CMSSW: A new bb flavour

In order to assign a flavour to reconstructed jets in the Monte Carlo samples, a scheme matching the generated quarks to the reconstructed jets is needed. For the current jet flavour matching in the CMS Offline Software (CMSSW), c and b hadrons are used for c and b jets instead of

quarks. This is to avoid having to define generator specific jet flavour algorithms due to different hadronisation schemes in the different generators and because the B hadrons are closer to the actual jet than the b quarks. The jet flavour is determined by reclustering the jet constituents with its selected hadrons. The hadron four-momenta is then rescaled by a very small number (the default rescaling factor is 10^{-18}), turning them into “ghosts”. These ghost hadrons are then clustered together with all of the jet constituents, leaving the jet collection practically identical to the original one but with ghost hadrons inside. This is illustrated in Figure 5.1. The flavour is

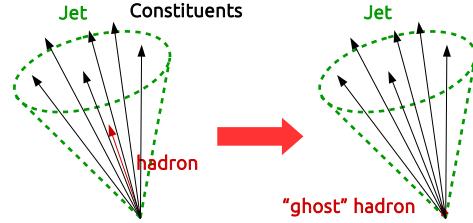


Figure 5.1: For the jet flavour matching the clustered hadrons inside a jet are made soft, the jet reclustered and the flavour determined by which jet the hadron ends up in.

then determined based on the ghost hadron content of the jet [39]. Previously all jets containing one or more b hadrons was defined as a b jet. In order to distinguish between jets containing one b and jets containing two b quarks, a new flavour definition is introduced:

- The jet is considered a b jet if there is one b hadron clustered inside it
- The jet is considered a bb jet if there are more than one b hadron clustered inside it

5.3 Event selection

The jets used in this study are required to have $p_T > 20$ GeV, pseudorapidity $|\eta| < 2.5$ and are reconstructed with the Particle Flow algorithm. Certain quality requirements are applied to ensure these jets originate from the hadronisation of quarks. These are:

- Jet neutral hadron energy fraction < 0.99
- Jet neutral electromagnetic energy fraction < 0.99
- Number of jet constituents > 1
- Charged hadron energy fraction > 0
- Charged track multiplicity > 0
- Charged electromagnetic energy fraction < 0.99

To remove contributions coming from other interactions in the event not related to the primary vertex (“pile up”), an additional step in the PF reconstruction is used where the PF particle is matched to the primary vertex. This is done by extrapolating the particle direction towards the beam line and checking that it is compatible with the PV position. For the jet reconstruction, two different algorithms are used: The anti- k_T algorithm with a cut-off parameter of $R=0.5$ (AK5), and the Cambridge-Aachen algorithm with $R=0.8$ (CA8), 1.2 (CA12) and 1.5 (CA15). The jet energies are corrected for pile up dependence and electronic noise, η and p_T . For the tracks used in b-tagging, only those fulfilling the following track quality requirements are used:

- $p_T > 1 \text{ GeV}$
- ≥ 2 valid hits in the pixel detector
- ≥ 8 valid hits in the tracker
- The track normalized $\chi^2 < 5$
- The distance between the track and the primary vertex in the transverse plane is less than 0.2 mm
- The distance between the primary vertex and the track is less than 17 mm

Only jets with at least three tracks and a valid value of the impact parameter are stored. After this selection, further track quality requirements are also applied which might reduce the amount of valid tracks in each jet. These cuts are:

- $\Delta R(p_{jet}, p_{track}) < 0.3$
- Distance between track and jet axis $< 0.07 \text{ mm}$
- Distance between primary vertex and track point of closest approach $< 5 \text{ mm}$
- $|M_{\sum \text{tracks}} - M_{K^0}| > 0.03 \text{ GeV}$

Quality cuts regarding the secondary vertex reconstruction are explained in Section 4.3.

Chapter 6

b jet identification in CMS

As described in Chapter 2, the long lifetime of the b hadron allows it to travel a considerable distance before it decays leading to displaced tracks originating from a common secondary vertex. This provides a good discriminant between b jets and light flavoured jets. Figure 6.1 illustrates a typical event where a b quark is produced. To define whether a track originates from the PV

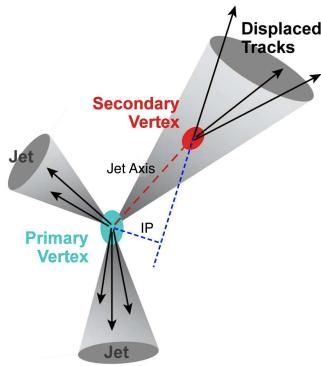


Figure 6.1: A b hadron has a long lifetime and can travel a considerable distance from the primary vertex before it decays. This leads to a secondary vertex consisting of displaced tracks incompatible with the primary vertex. The magnitude of the displacement is quantified by the track impact parameter (see Figure 6.2).

or a potential SV, the track impact parameter (IP) is used. The IP is defined as the distance between the primary vertex and the track at the point of closest approach (Figure 6.2). It can take positive (downstream) and negative (upstream) values depending on the sign of the scalar product of the IP-vector and the jet axis. The IP is expected to be symmetric around 0 for decays with a short flight path and mostly positive for a particle with longer lifetime [40]. The impact parameter significance (IP-Sig) defined as IP/σ_{IP} is normally used, where σ_{IP} is the uncertainty on the IP.

In the $b \rightarrow cW^{*-}$ decay (see Section 2.1.1) the c-flavoured quark can also travel a significant distance before decaying [11], leading to a third vertex in the event and an even higher multiplicity of displaced tracks. This is illustrated in Figure 6.3.

6.0.1 Algorithms

The main feature of any b-tagging algorithm used in CMS is to provide a simple discriminating variable separating b jets from other jets. In this way, applying cuts on the variable allows one

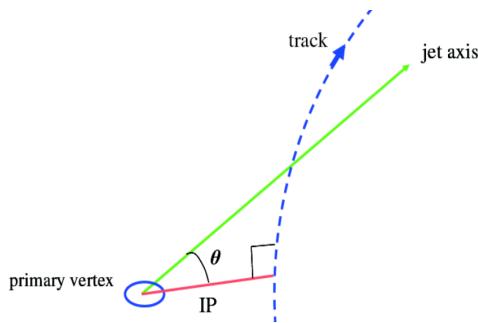


Figure 6.2: The track impact parameter (IP) is the point of closest approach between the track and the jet axis. For B-hadrons, the IP is set by its flight path from the primary vertex due to its long lifetime

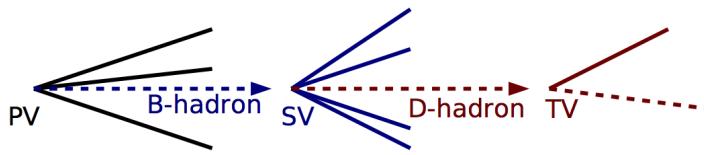


Figure 6.3: When a b-flavoured quark decays as $b \rightarrow cW$, the c quark also travels some distance before it decays. This can lead to a third vertex in the event. A D hadron is any hadron containing a c quark.

to select the value of efficiency and purity required for the analysis. The b-tagging algorithms can be secondary vertex based, impact parameter based or a combination of many variables and are described in the following.

The Track Counting Algorithm

The Track Counting Algorithms are b-tagging algorithms based on the track impact parameters only. Here a jet is defined as a b jet by having a certain number of tracks with an IP-significance above some chosen value. By sorting the tracks by their IP-significance in decreasing order, one defines the discriminator as the IP-significance of the N-th track. The Track Counting High Efficiency (TCHE) uses the IP-significance of the 2nd track, ensuring a high tagging efficiency, whilst the Track Counting High Purity (TCHP) algorithm uses the IP-significance of the 3rd track, providing a higher purity sample. The Track Counting discriminants are shown in Figure 6.4.

Jet Probability Algorithms

With the IP-significance one can obtain a probability density function for non b jet tracks. This is done by taking the negative IP-Sig distribution as calibration, as it consists mainly of tracks from the primary vertex. A track by track probability, P_{tr} , of having originated from a non b quark can then be defined. This forms the basis of the Jet Probability algorithms (JP). Here the compatibility of a collection of tracks having their origin at the PV is computed. If this probability is small, it is more likely that the tracks come from a b hadron. The JP has two discriminators; "Jet Probability" and "Jet b Probability". Jet Probability is the total probability that all the tracks in a jet originate from the PV and it is defined in the following

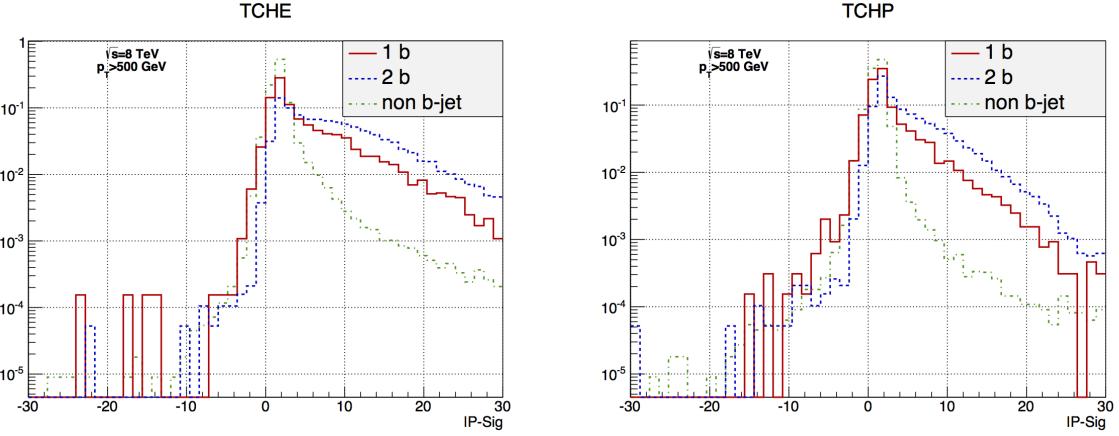


Figure 6.4: Discriminant distributions for the Track Counting algorithms using the IP significance of the 2nd (TCHE) and 3rd (TCHP) track. The discriminants are shown for light-flavoured jets, jets containing 1 b and jets containing 2 b quarks for jets with $p_T > 500$ GeV. Here the non b jet efficiency is based on QCD multijet events covering a wide range of jet transverse momentum from 20 GeV to 3000 GeV. The b/bb-jet efficiency is taken from a Graviton sample with $M_G=1500$ GeV and $G \rightarrow HH \rightarrow 4b$.

way:

$$P_{jet} = \prod \sum_{j=0}^{N-1} \frac{(-\ln \Pi)^j}{j!} \quad (6.1)$$

where

$$\Pi = \prod_{i=0}^N \tilde{P}_{tr}(i) \quad (6.2)$$

Here \tilde{P}_{tr} is the redefined track probability and N is the number of tracks in the jet. This is defined to ensure that the track probability is always positive. For $P_{tr} > 0$, $\tilde{P}_{tr} = P_{tr}/2$ and for $P_{tr} < 0$, $\tilde{P}_{tr} = 1 + P_{tr}/2$. The discriminating variable used for this algorithm is $-\ln P_{jet}$ and is expected to be low for light flavour jets and high for b-jets. To prevent single tracks in the jet with a very small probability of coming from the primary vertex from dragging down the distribution, tracks with a probability $P_{jet} < 5 \times 10^{-3}$ only contributes to the jet probability with 5×10^{-3} . Jet b Probability gives the probability that the four most displaced tracks originate from the PV. Four is chosen as the average number of charged particles in a weak B decay is 5 and there is an 80% track reconstruction efficiency for tracks in jets. A more detailed description of the JP algorithms can be found in [41].

The Combined Secondary Vertex Algorithm

The Combined Secondary Vertex (CSV) algorithm uses several discriminating variables in combination with the track IP-significance. To select secondary vertex candidates, the algorithm applies requirements to the transverse distance between the primary and secondary vertex and the invariant mass of the charged particles associated to the vertex. This selection defines three different vertex categories:

1. **RecoVertex:** At least one reconstructed SV.
2. **PseudoVertex:** No reconstructed SV, but at least two charged particle tracks with an IP-significance greater than 2.

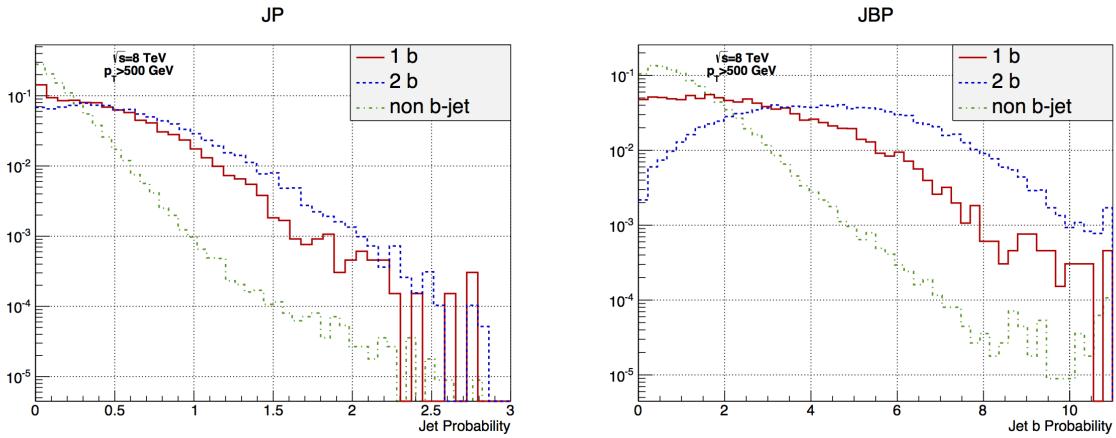


Figure 6.5: Distributions of the Jet Probability (JP) and Jet b Probability (JBP) discriminants for light-flavoured jets, jets containing 1 b and jets containing 2 b quarks.

3. NoVertex: None of the above satisfied.

Depending on the vertex category above, different variables are considered. A full list and description of the variables is given in Section 7.2.2 and more detailed information about the CSV algorithm can be found in [42]. Figure 6.6 shows the distribution of the CSV discriminant.

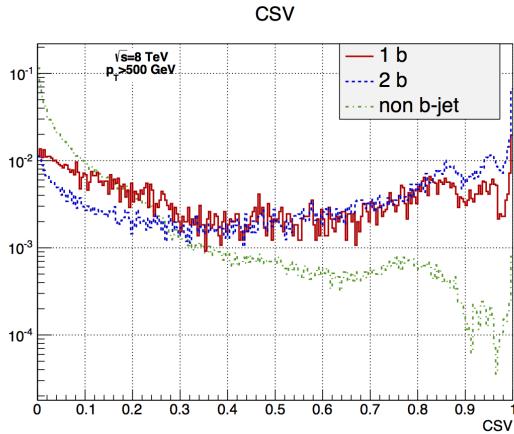


Figure 6.6: Distribution of the CSV discriminating variable for light-flavoured jets, jets containing 1 b and jets containing 2 b quarks.

As can be seen from the distributions of the different b-tagging discriminants, all variables have a higher value for b jets than for lighter jets and a cut on the discriminator providing the requested efficiency and purity can easily be applied. The CMS b-tagging group has defined official working points for the different b-tagging algorithms derived from the mistagging rate as follows:

- Loose: 10%
- Medium: 1.0%
- Tight: 0.1%

The cut values for different datasets are continuously updated and can be found in [43].

6.0.2 Backgrounds

The dominant backgrounds when doing b-tagging comes from the decay of c quarks and τ leptons. Both particles travel a significant distance before decaying, leading to secondary vertices and displaced tracks similar to a b quark decay topology. The lifetime of the different c baryons/mesons differs, but some, like the D meson, has a lifetime of the same order as the b quark around 1×10^{-12} [44]. As the c quark is lighter than the b with a mass around $M_c=1.3$ GeV [11], cuts on the vertex mass can be applied in order to reduce this background. The τ lepton has a mass of $M_\tau=1.8$ GeV and a lifetime of around 0.3 ps [45]. The flight length of the τ is significantly shorter than the b quark flight length and vertex mass cuts as well as flight length requirements can be applied.

6.0.3 b-tagging in boosted topologies

As illustrated in Figure 6.7, b quarks coming from a boosted parent will be emitted in a collinear topology and can get merged into a single jet by the jet algorithm. Previous studies have shown

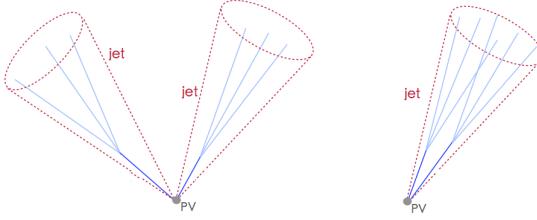


Figure 6.7: If the decaying particle is boosted enough, the separating angle between its decay products will be too small for the jet algorithm to resolve. Instead of having two well separated jets, the b quark pair is merged into one single jet.

that for a boosted Higgs with a generator level $p_T \geq 250$ GeV, the two b quarks are fully contained within a jet of $R=0.8$ [47]. Figure 6.8 shows the ΔR between the b quarks stemming from a Higgs boson decay versus the true transverse momentum of the Higgs. As the transverse momentum of the Higgs boson increases, the distance between the decay product decreases. For these topologies, jet substructure methods [46] have proven to be powerful. This technique consists of reclustering the jet into its smaller constituents and running the b-tagging algorithms separately on the subjets. One can alternatively use “fat jet b-tagging”, where the jet is tagged as a whole without splitting it into subjets. The two different methods are illustrated in Figure 6.9. A Cambridge-Aachen jet with an R parameter of minimum 0.8 is typically used for the fat jet and smaller cones of $R=0.5$ are used for the subjets. Subjet b-tagging has shown good performance in boosted Higgs scenarios [47], but the strategy fails once the subjets get too close to one another and their tracks start to overlap. This already happens for standard AK5 jets at a Higgs $p_T > 400$ GeV, as can be seen from the plot in Figure 6.8. The tracks included by the b-tagging algorithm in the discriminant computations, are set by the Jet Track Association Cone (JTA) and has a default value of $R = 0.3$ for AK5 jets. When the subjet JTA cones get closer than $R = 0.3$, there is track sharing between the cones and these correlations can be hard to model in the b-tagging algorithm (see Figure 6.10). This can cause a loss of one or both b jets in the high- p_T region. In situations like these, fat jet b-tagging has shown to perform better than subjet b-tagging in the high-efficiency region [47]. Figure 6.11 shows the mistagging rate as a function of b jet tagging efficiency for $H \rightarrow b\bar{b}$. In the high-efficiency region, fat jet

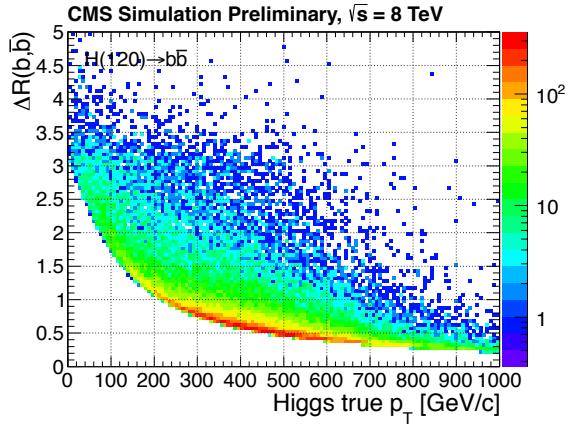


Figure 6.8: $\Delta R(b\bar{b})$ as a function of generator level p_T of the Higgs. For a Higgs $p_T \geq 250$ GeV the two b quarks are fully contained within a jet cone of $R=0.8$ [47].

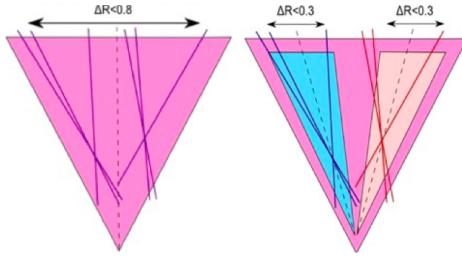


Figure 6.9: Subjet (left) versus fat jet (right) b-tagging. Both start with a large jet (here $R=0.8$). For subjet b-tagging, the two subjets are tagged separately using a JTA cone of $\Delta R < 0.3$, while for fat jet b tagging the JTA cone is increased to $\Delta R < 0.8$ [46].

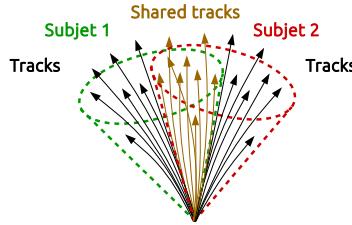


Figure 6.10: When the Jet-Track Association (JTA) cone of the subjets start to overlap, there is track sharing between the jets, leading to correlations that are hard to model when doing b-tagging.

b-tagging achieves a 10% higher tagging efficiency than subjet b-tagging. One can attempt to do fat jet b-tagging using the standard b-tagging algorithms on jets containing two b quarks. This is shown in Figure 6.12 where the tagging efficiency versus misidentification probability is plotted for jets containing one b quark (left) and jets containing two b quarks (right). The CSV algorithm has the highest tagging efficiency for a given mistagging rate for 1b jets as expected. For jets containing two b quarks however, the simpler JBP algorithm performs almost 10% better than the CSV at a loose (10% mistagging rate) working point. This illustrates the need for a dedicated double b-tagger designed to tag jets containing two b quarks.

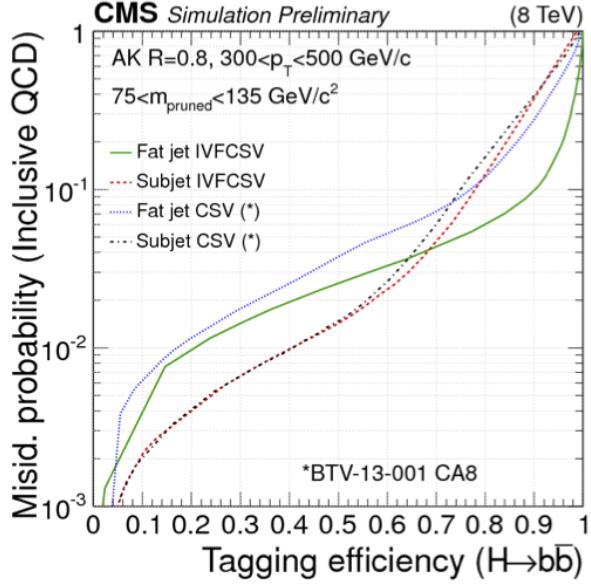


Figure 6.11: Misidentification probability as a function of the tagging efficiency for boosted $H \rightarrow b\bar{b}$ and inclusive QCD jets for the CSV algorithm applied to fat jets and subjets of fat jets with $300 \text{ GeV} < p_T < 500 \text{ GeV}/c$. For the subjets, both subjets are required to pass the same selection requirement on the b-tag discriminator. Here “IVFCSV” is the current CSV algorithm using IVF vertices, while “CSV” is an older generation CSV algorithm using another secondary vertex reconstruction scheme [47].

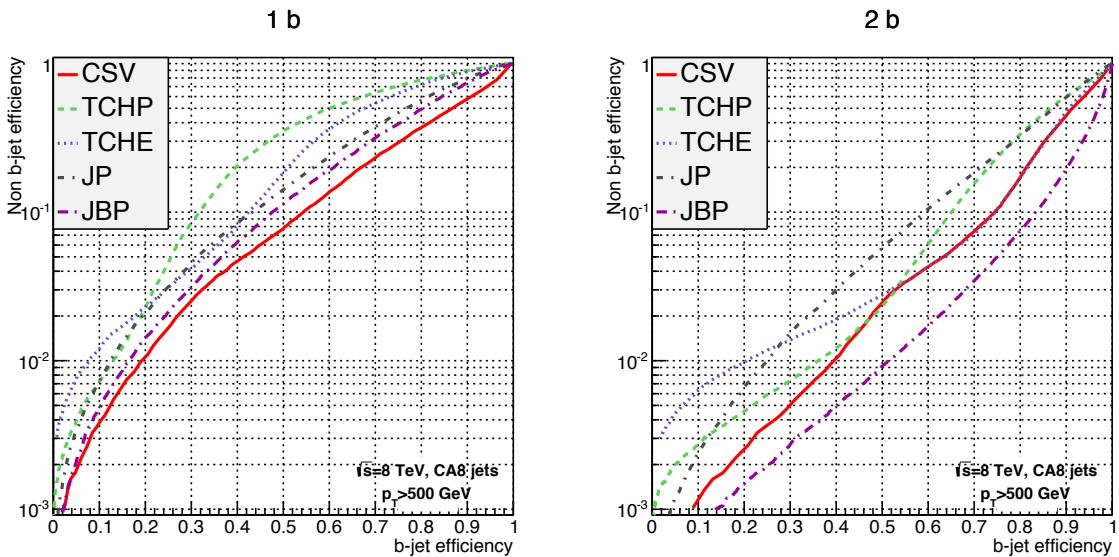


Figure 6.12: b/bb tagging efficiency versus mistagging rate for jets containing one b quark (left) and jets containing two b quarks (right). The CSV algorithm has the highest efficiency for a given mistagging rate for 1b jets, whilst the simpler JBP algorithm performs significantly better for jets containing two b quarks.

Chapter 7

Dedicated bb vs. b training

Separating jets containing two b quarks from jets containing one b quark is more difficult than separating b jets from light-flavoured jets. In this case both jets contain several displaced tracks and share many other b jet characteristics. Figure 7.1 shows the distribution of the track impact parameter significance that raises the mass of the jet above the charm mass for b and cdusg jets (left) and for bb and b jets (right). This is one of the variables used by the CSV algorithm offering the highest discriminating power between b jets and light-flavoured jets. For the bb and b jets, however, the distributions are very similar. This goes for many of the variables typically used in normal b-tagging. The best way to combine several such “weakly” discriminating variables

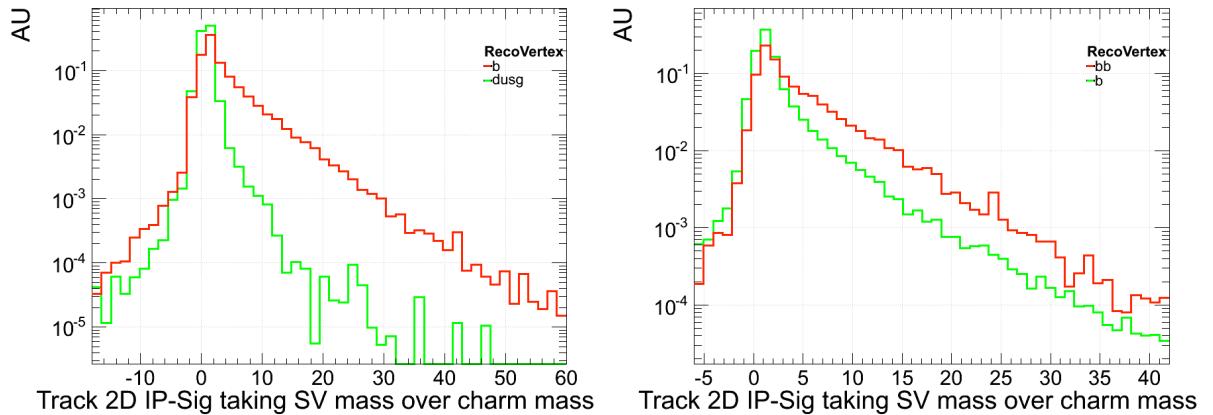


Figure 7.1

into one powerful one is by using Multivariate Analysis (MVA) Techniques. MVA techniques have been growing increasingly more popular in situations requiring classification and analysis of data described in multiple dimensions by a large set of variables. They base themselves on the principle of multivariate statistics and are able to treat the variables independently as well as treating the correlations between them. The dedicated double b-tagging algorithm presented in this thesis uses an MVA technique called Artificial Neural Network (ANN) [48] and will be further explained in Section 7.1. It is trained as a simple signal (bb) versus background (b/light-flavoured) discriminating algorithm. The neural network is fed two different distributions, one for signal and one for background, containing the same set of variables. The network is then trained to tell the difference between the two types of events based on the information it has gathered during the training. The algorithm is a retraining of the standard CSV algorithm and is trained separately for bb vs. b and bb vs. light-flavoured jets. Like the CSV algorithm, the

double b-tagger is also trained in three different vertex categories: RecoVertex, PseudoVertex and NoVertex (as described in Section 6.0.1). Different sets of variables are used for the different categories (see Table 7.1).

7.1 Artificial Neural Network

A neural network works by using a large number of highly interconnected processors called neurons that work together to classify data. It consists of at least three layers of neurons, where each layer is a non-linear combination of non-linear functions from the layer before. The first layer receives the input data and the last layer provides the output after the data has been processed. The layers in between, the so called “hidden layers”, do all the data processing. The network layout is shown in Figure 7.2.

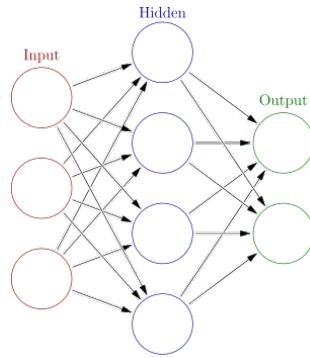


Figure 7.2: Layout of an Artificial Neural Net. The input layer contains the training input variables, then there is a layer of neurons computing the training weights followed by the output neurons using the given weights and providing the network output.

7.1.1 Multilayer Perceptron

There exists several types of neural networks and here we use a multilayer perceptron (MLP) [49]. This is a “feedforward” neural net, meaning that the connection between the neurons never form a cycle and the information moves only in one direction; from the input node to the output nodes. Each node in one layer connects with a certain weight to every node in the subsequent layer. All nodes are neurons with a non-linear “activation function”. An activation function is a function that maps the weighted inputs to the output of each neuron. The non-linearity of this function is part of what makes an MLP different from other neural networks. The learning itself occurs through “backpropagation”. Backpropagation is done by changing the connection weights after each data point is processed based on the amount of error in the output compared to the expected result. To quantify the error one uses “cost functions” that calculate how far away from the optimal solution the computed training function is. In our case we use a simple χ^2 over 100 training iterations (see Section 7.2.7) with the goal of minimizing the function. In short, the neural network is given a set of input variables to learn from in order to produce a single output variable, the network discriminant.

7.2 The double b-tagger training

7.2.1 Framework

The construction of the MLP neural network is done through the MVATrainer and MVAComputer framework implemented in the CMS offline software (CMSSW) [50]. The benefit of using this framework is that it contains a full chain of machine learning processors easily configurable through steering files. Here users are allowed to define their own networks by combining different types of variable processors: preprocessors, processors dealing with multi-appearance variables and classifiers. An example of a preprocessor can be the normalization of a variable to avoid sharply peaked distributions and a classifier is for instance an MLP neural network. This allows full control over the preparation of variables and the training through one single configuration file. A full list of available processors can be found in the Appendix. The MVATrainer is responsible for the preprocessing and training of the network. The information it has gained during the training (called “calibration”) is then stored in order to be evaluated by the MVAComputer. The MVAComputer uses exactly the same input variables and processors as the MVATrainer and produces the final discriminant. The schematic overview over the framework can be seen in Figure 7.3.

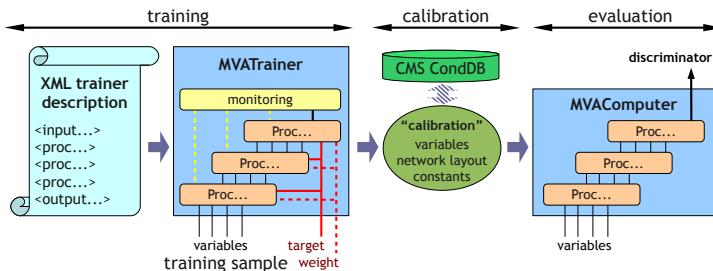


Figure 7.3: Overview over the MVATrainer and MVAComputer framework [50]. The training is fully configurable through XML steering files where the training input is defined. The training samples are then sent to the MVATrainer for the learning process, and the final network layout (called “calibration”) is stored. The network is then evaluated by the MVAComputer providing the final discriminant.

7.2.2 Training preparation

For sufficient statistics, different Monte Carlo samples for the bb jets and b/light- jets are used in the training of the double b-tagging algorithm: a Radion sample where $R \rightarrow HH \rightarrow 4b$ for bb jets and QCD for the b/light flavoured jets. Three training trees for each flavour are created according to what vertex category the jet falls in (Reco, Pseudo or No). This results in a total of 9 different training samples (3 for bb, 3 for b and 3 for light flavoured jets). The total list of variables for each vertex category can be found in Table 7.1. The variables marked as n/u are not used for that vertex category.

Variable name	RecoVertex	PseudoVertex	NoVertex
jetPt	used	used	used
jetEta	used	used	used
trackSip3dSig	used	used	used
trackSip2dSigAboveCharm	used	used	used
trackPtRel	used	used	used
trackDeltaR	used	used	used
trackPtRatio	used	used	used
trackJetDist	used	used	used
trackDecayLenVal	used	used	used
trackSumJetEtRatio	used	used	used
trackSumJetDeltaR	used	used	used
trackEtaRel	used	used	n/u
vertexMass	used	used	n/u
vertexNTracks	used	used	n/u
vertexEnergyRatio	used	used	n/u
vertexJetDeltaR	used	used	n/u
flightDistance2dSig	used	n/u	n/u
jetNSecondaryVertices	used	n/u	n/u

Table 7.1: Variables used for the training in the Reco, Pseudo and NoVertex category. The variables marked as n/u are not used for the training in that vertex category.

A short description of the variables is given below and taken from [51]:

- jetPt: the transverse momentum of the jet.
- jetEta: the pseudorapidity of the jet.
- trackSip3dSig: the signed impact parameter significance of each selected track.
- trackSip3dSigAboveCharm: the signed impact parameter significance of the track that raises the mass obtained from the summed four-momenta of the current track with the previous tracks (where the tracks are sorted by decreasing value of Sip3dSig) above the mass of the charm quark (1.5 GeV).
- trackPtRel: the track transverse momentum, relative to the jet axis.
- trackDeltaR: ΔR between the jet direction and the track momentum.
- trackPtRatio: the track transverse momentum, relative to the jet axis, normalized to the magnitude of its momentum.
- trackJetDistVal: the distance between the track and the jet axis.
- trackDecayLenVal: the decay length of the track calculated as the distance between the primary vertex and the Point of Closest Approach of the track with respect to the jet axis.
- trackSumJetEtRatio: the ratio of the transverse energy of the summed four-momenta of all selected tracks and the transverse energy of the jet.
- trackSumJetDeltaR: ΔR between the summed four-momenta of all selected tracks and the jet direction.

- trackEtaRel: $\eta_r = 0.5 \ln \left(\frac{E+trackPPar}{E-trackPPar} \right)$, with $E = \sqrt{|\mathbf{P}_{\text{track}}^2 + m_\pi^2|}$, and $trackPPar$ is the track momentum parallel to the jet direction (scalar product of the jet direction and the track momentum). Track pseudorapidity with respect to the jet axis
- vertexMass: mass of track sum at secondary vertex.
- vertexNTracks: the number of tracks associated with the secondary vertex.
- vertexEnergyRatio: the ratio of the energy of the summed four-momenta of all secondary-vertex tracks and the energy of the summed four-momenta of all tracks associated with the jet.
- vertexJetDeltaR: ΔR between the summed four-momenta of all secondary-vertex tracks and the jet direction.
- flightDistance2dSig: the transverse distance between the primary and the secondary vertex divided by its error.
- jetNSecondaryVertices: the number of reconstructed secondary vertices (of the type ‘RecoVertex’).

The distribution of the variables listed above for bb and b jets in the RecoVertex category are shown in Figure 7.4. The distributions for bb and b jets are similar, but some have a higher discriminating power than others. This goes especially for the ΔR between the four-momenta of all SV tracks and the jet direction (5th row, right), track pseudorapidity with respect to the jet axis (3rd row, left) and ΔR between the jet direction and the track momentum (2nd row, right). For the ΔR between the secondary vertex and jet direction, the secondary vertex direction seems to be further away from the jet direction in ΔR space for bb jets than for b jets. The distribution is also broader. This is most likely because there are two b quarks going in two slightly different directions with the jet axis in the middle whereas for b quarks, the jet originates from only one b quark decay and should be aligned with the secondary vertex direction. The same argument goes for the ΔR between the jet direction and the track momentum. The tracks in the bb jets are closer to the jet axis in η space. This can be expected as bb jets are formed due to their collimated tracks.

Figure 7.5 shows the number of jets in each vertex category as a function of p_T (left) and η (right). The bb jet compared to the b jet statistics are low and concentrated in the $250 \text{ GeV} < p_T < 600 \text{ GeV}$ region. This is due to the fact that we are working with Higgs bosons coming from a Radion with a certain mass. The momenta of the decay products are then limited by conservation laws. The training is performed in flat p_T - η space in order to account for this (see Section 7.2.3), but the low/high- p_T bins are scarcely populated. A solution to this would be to add Radion samples of lower/higher mass points to the bb training samples. For a lower mass point sample however, the boost of the Higgs boson decreases and the b quarks are not as collimated. Subjet b-tagging methods can then be used as long as the subjet JTA cones do not overlap. Adding higher mass points to the training might be useful in order to increase the statistics in the high- p_T region and in order to improve the double b-tagger for Higgs bosons with a higher boost. The difference in performance using higher mass points is presented in Chapter 8. The statistics in the Pseudo and NoVertex categories for bb jets are also low. This could lead to large random fluctuations in the variable distributions and have a large impact on the training. Nevertheless, we do not expect this to have a big influence on the final efficiency as bb jets without a reconstructed vertex are so rare.

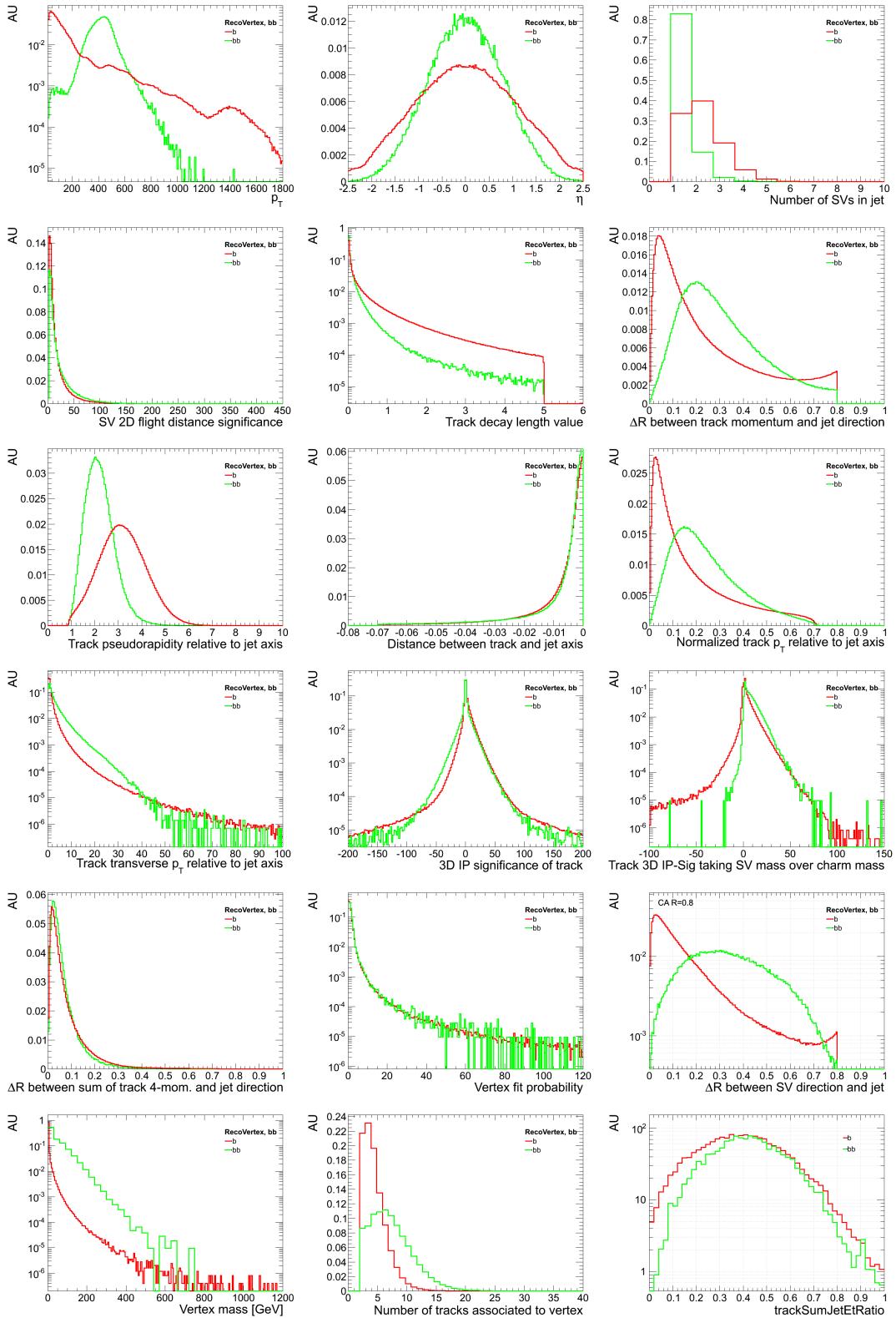


Figure 7.4: The distribution of variables entering the training for bb (green) and b jets (red) normalized to one.

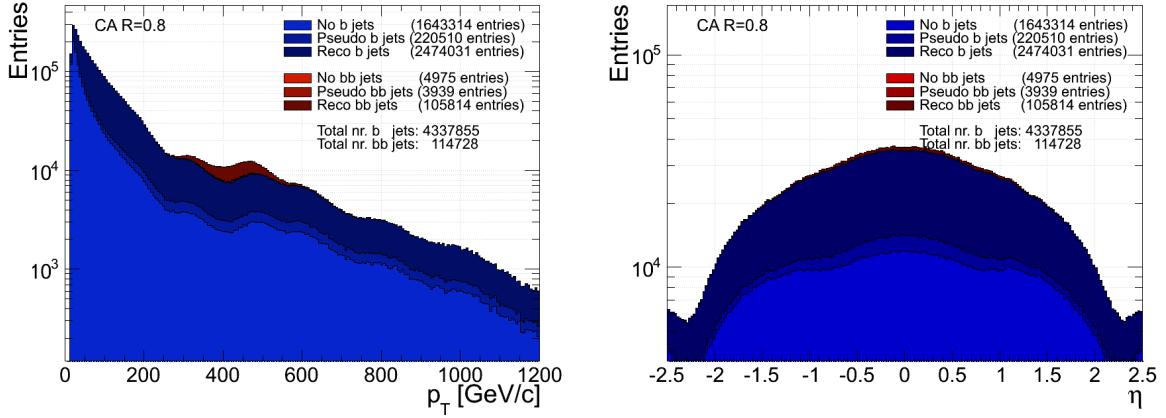


Figure 7.5: Number of bb and b jets per vertex category as a function of p_T (left) and η (right). Most of the bb jets fall in the RecoVertex category as expected, and the statistics in the remaining categories are low. The number of b jets is in general a lot higher than bb jets, but this is later reduced through a skimming procedure (see Figure 7.8).

7.2.3 p_T - η reweighting

The p_T spectrums of the bb and b jets are very different as can be seen in the left plot in Figure 7.5. As we do not want the training to learn from the p_T and η of the jet such that it for instance believes that all high- p_T jets are bb jets, we apply a weight to each jet in order to train in flat p_T - η space. This is done by creating 2D p_T versus η histograms for each flavour and vertex category, a total of 9 histograms, with 50 p_T bins ranging from 20 GeV to 1000 GeV and 40 η bins with $|\eta| \leq 2.5$. The 2D η versus p_T histogram for bb jets in the RecoVertex category is shown in Figure 7.6. The jet weight is then computed as the inverse of the number of jets in

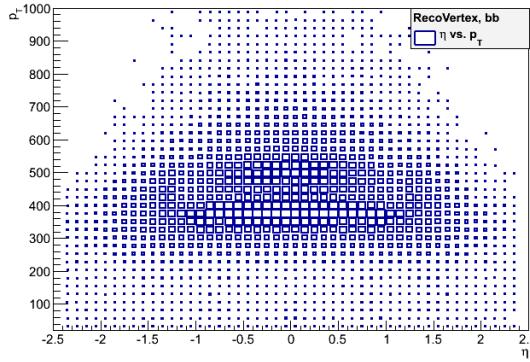


Figure 7.6: p_T versus η for bb jets in the RecoVertex category. Every jet is reweighted with a weight corresponding to the inverse of the bin content of the p_T – η bin the jet falls in.

the bin the jet falls in. The 2D projections of the reweighted p_T and η distribution are shown in Figure 7.7. The distributions are not completely flat at high p_T/η values as for high p_T not all η bins are filled and vice versa.

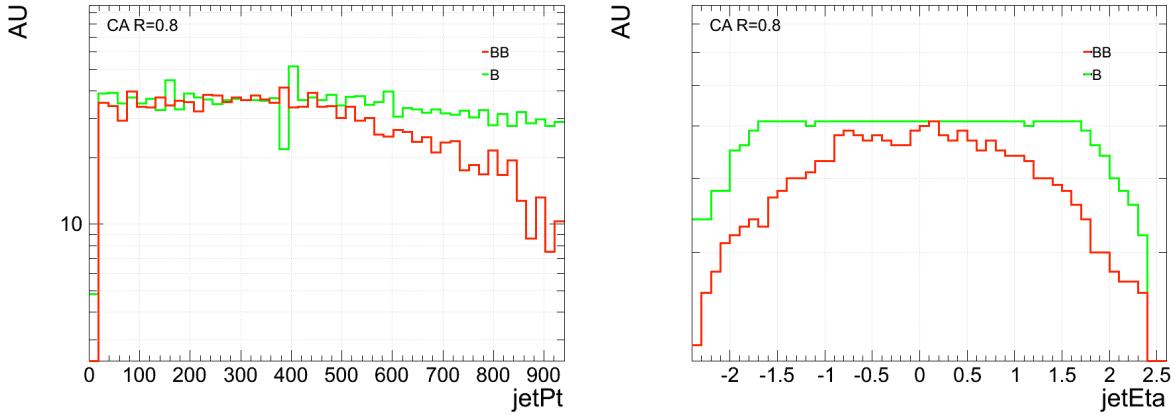


Figure 7.7: The 2D projections of the p_T and η spectrum after reweighting for bb jets (red) and b jets (green).

7.2.4 Training bins

As the input variables themselves might depend on the jet p_T and η , the training is performed separately in bins of p_T and η in order to increase the sensitivity. The binning is chosen such that each bin contains sufficient training statistics. The standard CSV training uses a p_T - η binning with multiple low- p_T bins and fewer at high- p_T as the statistics are higher of both b and light flavour jets in the low- p_T region. For bb jets, however, the distribution peaks in a narrow p_T window between $150 \text{ GeV} < p_T < 600 \text{ GeV}$. The binning has therefore been changed to better suit our signal, resulting in the 13 different training bins listed on the right side of Table 7.2. The binning used by the standard CSV training is shown on the left. In the new binning scheme the low- p_T bins are merged into one single $20 \leq p_T \leq 150 \text{ GeV}$ bin, while the region between 150 and 400 GeV has been split in two. What binning provides the best results requires optimization depending on what training samples are used and results using both default and new p_T - η bins are presented in Chapter 8.

In order to avoid unnecessarily large computational times the samples are skimmed, keeping only 50,000 jets in each p_T - η training bin. This leads to a falling p_T distribution in each bin, which is taken into account by the p_T - η reweighting. The number of jets as a function of p_T after the skimming can be seen in Figure 7.8 for the default p_T - η training bins on the left and the new bins on the right. Another reason for changing the p_T - η training bins is to reduce the sharp increase of bb and decrease of b jets in the range $200 \text{ GeV} < p_T < 400 \text{ GeV}$ for the old binning.

7.2.5 Biasing

As we have disregarded all prior information of the probability of a jet with a certain flavour to fall in a certain vertex category, we want to reinsert this information using a physically realistic model. We therefore bias the training by inserting the fraction of bb over b/light-flavoured jets that fall in each vertex category. The bias is computed for each p_T - η bin, vertex category and flavour. For this we use a $\text{Radion}(M_R = 1.0 \text{ TeV}) \rightarrow \text{HH} \rightarrow 4\text{b}$ sample as it simulates our b-tagging scenarios well. This sample contains all jets (without skimming) in the Reco, Pseudo and NoVertex categories for all three flavours (bb, b and cdsg). The bias is then computed as

$$b_i = \frac{N_{\text{sig},i,\text{vtx}}/N_{\text{sig},i,\text{all}}}{N_{\text{bkg},i,\text{vtx}}/N_{\text{bkg},i,\text{all}}} \quad (7.1)$$

Bin	p_T range [GeV]	$ \eta $ range	Bin	p_T range [GeV]	$ \eta $ range
0	15 - 40	0 - 1.2	0	20 - 150	0 - 1.2
1	15 - 40	1.2 - 2.1	1	20 - 150	1.2 - 2.1
2	15 - 40	2.1 - 2.4	2	20 - 150	2.1 - 2.4
3	40 - 60	0 - 1.2	3	150 - 250	0 - 1.2
4	40 - 60	1.2 - 2.1	4	150 - 250	1.2 - 2.1
5	40 - 60	2.1 - 2.4	5	150 - 250	2.1 - 2.4
6	60 - 90	0 - 1.2	6	250 - 400	0 - 1.2
7	60 - 90	1.2 - 2.1	7	250 - 400	1.2 - 2.1
8	60 - 90	2.1 - 2.4	8	250 - 400	2.1 - 2.4
9	90 - 150	0 - 1.2	9	400 - 600	0 - 1.2
10	90 - 150	1.2 - 2.1	10	400 - 600	1.2 - 2.4
11	90 - 150	2.1 - 2.4	11	600 - inf.	0 - 1.2
12	150 - 400	0 - 1.2	12	600 - inf.	1.2 - 2.4
13	150 - 400	1.2 - 2.1			
14	150 - 400	2.1 - 2.4			
15	400 - 600	0 - 1.2			
16	400 - 600	1.2 - 2.4			
17	600 - inf.	0 - 1.2			
18	600 - inf.	1.2 - 2.4			

Table 7.2: p_T - η training bins used for the default CSV training (left) and the new bins created for the double b-tagger (right).

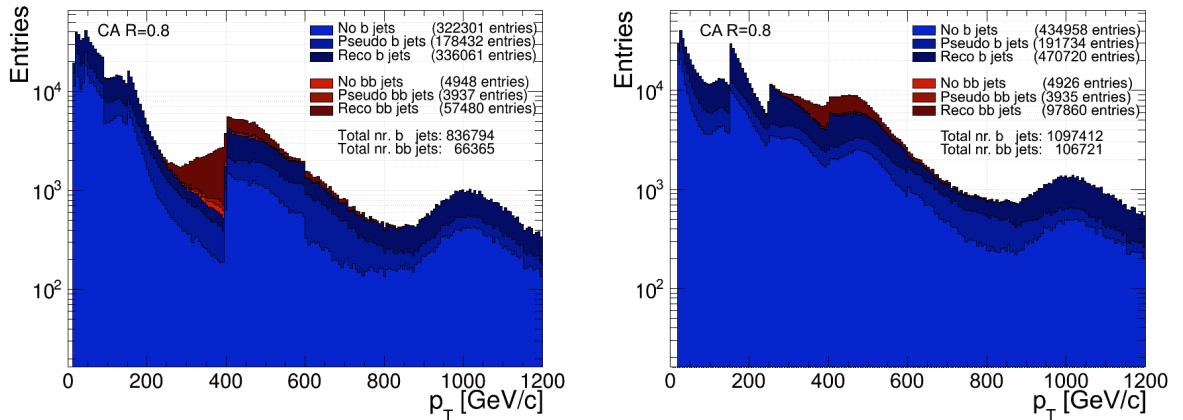


Figure 7.8: Number of bb and b jets per vertex category as a function of p_T after skimming the training samples, keeping only 50k events per p_T - η bin. The distribution for the default CSV binning is shown in the left plot and the new binning is shown to the right.

where $N_{sig,i,vtx}$ is the number of bb jets in a certain vertex category vtx and p_T - η bin i , $N_{sig,i,all}$ is the total number of bb jets in bin i for all vertex categories and $N_{bkg,i,vtx}$ is the corresponding number for b/cdusg jets. This is then applied to the training by means of a second likelihood ratio (explained further in Section 7.2.6):

$$f_i = \frac{b_i \times LR_{sig_i}}{b_i \times LR_{sig_i} + LR_{bkg_i}} \quad (7.2)$$

Here LR_{sig_i}/LR_{bkg_i} is the likelihood ratio distribution in training bin i for signal and background respectively. The bias used in this study is chosen as it models the boosted topology we are interested in and contains sufficient bb/b jet statistics. Of course; there is a “bias using a bias” in the sense that we are inserting p_T information that depends on the Radion mass point of our choice. A b-tagger biased with a 1 TeV resonance mass sample is better at tagging bb jets stemming from a similar mass resonance. An option would be to have different double b-taggers for different mass points. Then the user can choose the most suitable tagger for his or her analysis.

7.2.6 Variable preprocessing

Before performing the neural network training, some preprocessing of the variables is needed. First the jets are split into separate p_T - η training categories. The 13 different p_T - η training bins (see Table 7.2) are defined and the input jets are assigned to one, and only one, of the categories.

Variable normalization

After the p_T - η binning, the variables are all normalized to a value between 0 and 1 in order to be easily combined in a likelihood ratio. This is done by transforming the value x to x' through the transformation

$$x' = \frac{\int_{x_{min}}^x pdf(x)dx}{\int_{x_{min}}^{x_{max}} pdf(x)dx} \quad (7.3)$$

Here $pdf(x)$ is the probability density function of the variable and x_{min}/x_{max} is the minimum/maximum value of variable x . This procedure is completely lossless with respect to the training sample and has the benefit of smoothing the variable distributions such that sharp peaks are avoided. The pdfs as well as the transformed variables of the track η relative to the jet axis (“trackEtaRel”) and the track 3D impact parameter significance (“trackSip3dSig”) are shown in Figure 7.9 for bb and b jets in the RecoVertex category. The sharp peaks visible in the variable distributions in Figure 7.4 are flattened and spread over several bins.

Variable splitting

The final step of the preprocessing is to take variables that are stored as vectors and occur multiple times, like track-by-track related variables, and split them into separate entities. For variables like the 3D signed impact parameter significance, the value of the IP significance is stored for all tracks in the jet ordered by decreasing IP significance values. For b jets, the probability of having more than two tracks with a high impact parameter significance is a lot higher than the corresponding probability for a light flavoured jet. This means that the IP significance of the third track should be significantly higher for b jets than for light flavoured jets. The equivalent is true for bb jets. With an average of 5 charged tracks coming from a weak b hadron decay, it is more likely to have a high IP significance value of the third track for bb jets than for b jets simply because there are more of them. The “trackSip3dSig” is therefore split into separate entities for the first, second, third and remaining tracks in the jet. For the case of bb versus b it may be useful in the future to split the distribution into more separate entities and use for instance the IP significance of the 8th track (five charged tracks from each b hadron decay with an 80% reconstruction efficiency). Figure 7.10 shows the variable distribution of the 3D signed impact parameter significance for the first, second, third and remaining tracks for bb jets. It is clear that there is a higher discriminating power for the third track than for the first.

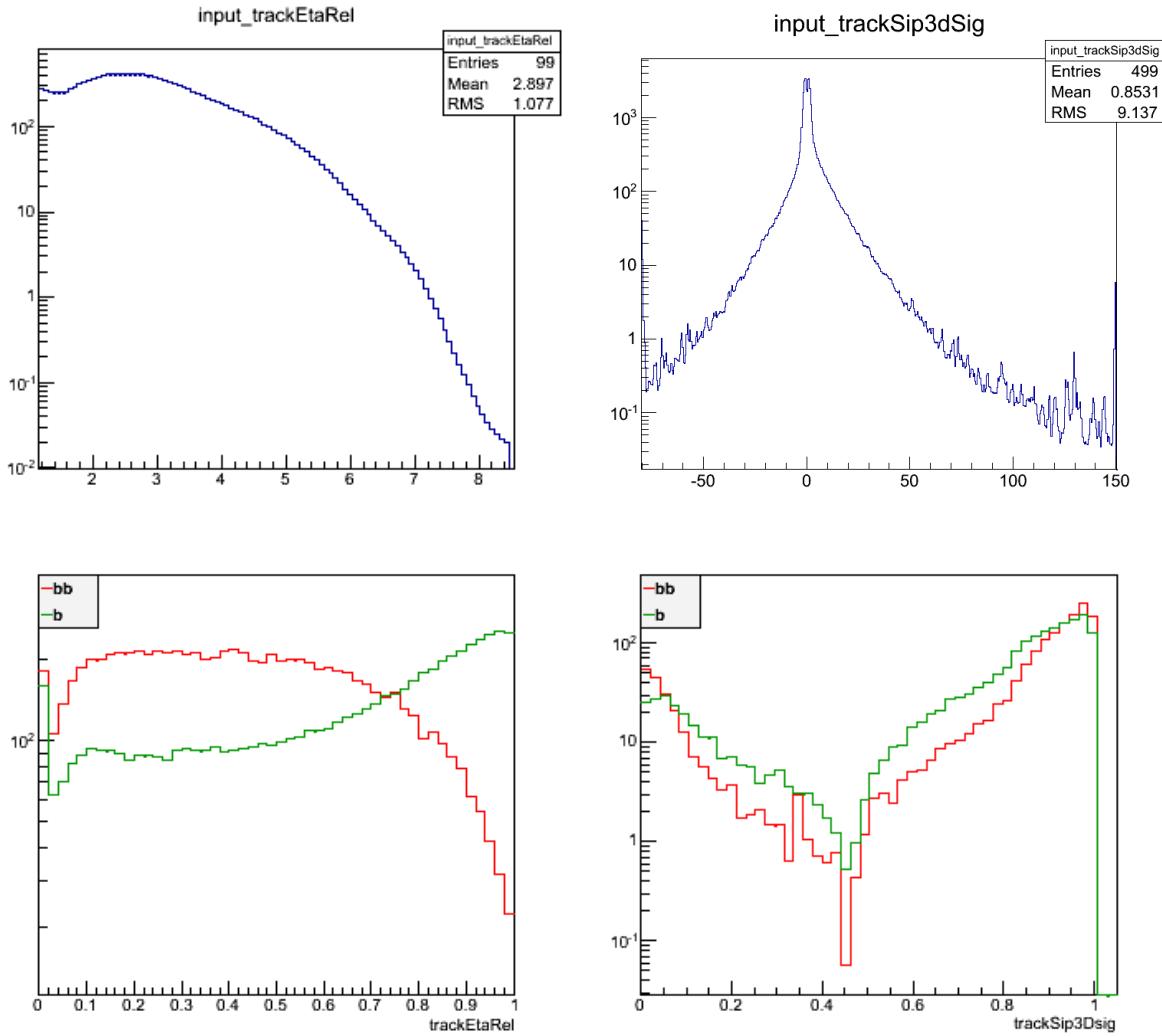


Figure 7.9: The top row shows the probability density functions (pdf) of the track η relative to the jet axis (top left) and the track 3D impact parameter significance (top right). The pdfs are computed with signal and background combined. The bottom row shows the variables after the transformation. The sharp peaks in the distributions (especially at the tails) have been flattened out resulting in a smooth distribution between 0 and 1.

7.2.7 Classification

After the preprocessing steps, the training itself is performed. This is done with a simple neural network MLP with 44 neurons in one hidden layer. The cost function is a χ^2 over 100 training iterations. A χ^2 minimization involves minimizing the expression

$$\sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}} \quad (7.4)$$

For each tenth iteration, the average χ^2 is printed out giving the possibility to monitor the progression of the training. An example can be seen in Figure 7.3 for the bb versus b training in the RecoVertex category. Here, “relative χ^2 ” is the χ^2 fit index divided by the degrees of freedom in the fit. The improvement of the training per training iterations can be seen from the decreasing value of the χ^2 .

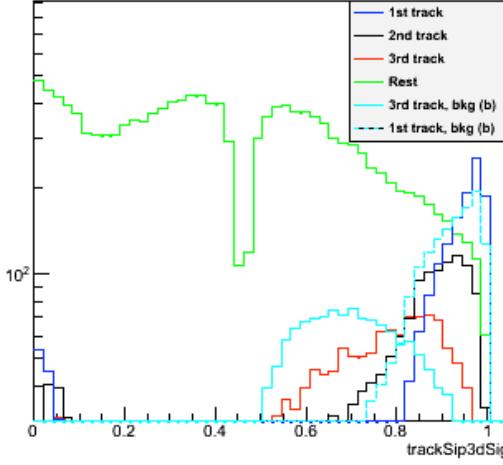


Figure 7.10: The normalized 3D impact parameter significance of the first (blue), second (black), third (red) and remaining (green) tracks for bb jets. Also shown in cyan is the 3D IP-Sig of the first (dotted) and third (solid) track for the b jet background. The IP significance of the 3rd track has better separation power than the first.

Training MLP epoch	Relative χ^2
1	0.0719362
11	0.0687354
21	0.0669674
31	0.0662043
41	0.0655118
51	0.0649827
61	0.0645771
71	0.0642629
81	0.0640213
91	0.0637985

Table 7.3: The relative χ^2 indicating how far away from the optimal result the training is per tenth training iteration. Here for the bb versus b training in the RecoVertex category

Discriminant normalization

After the training has been performed and the output discriminant computed, the discriminant is normalized using Equation 7.3 to reduce sharp peaks in the distribution. Figure 7.11 shows the unnormalized (left) and normalized (right) discriminant distribution for the bb versus b training in the RecoVertex category. The normalized discriminant has a smoother shape and takes only values between 0 and 1.

Biasing the discriminant

At this stage the prior probability for a jet of a certain flavour to end up in a certain training category is unaccounted for. The bias mentioned in Section 7.2.5 therefore needs to be applied to the discriminant. In order to do so, the likelihood ratio per $p_T\text{-}\eta$ bin is computed from

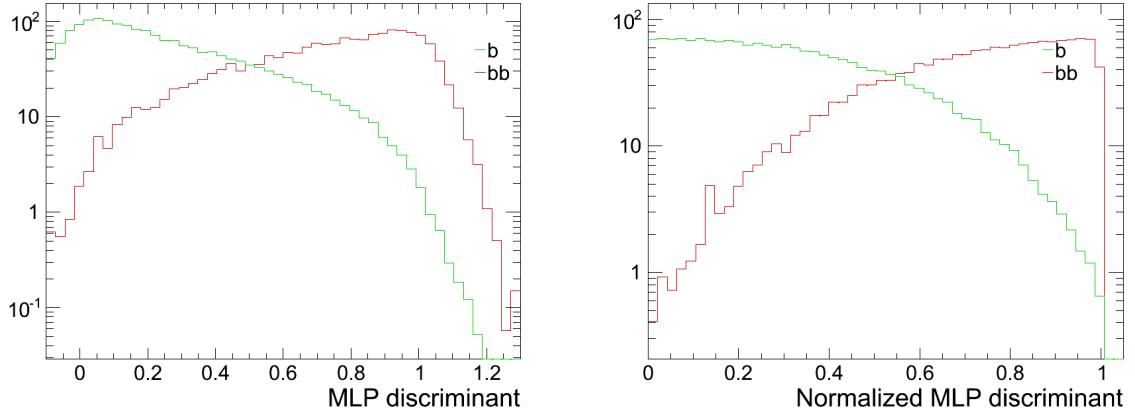


Figure 7.11: The discriminant distribution after the neural network training before (left) and after (right) normalizing the distribution for bb and b jets in the RecoVertex category. Here the normalization serves to smooth out the distribution.

the normalized discriminant and the bias is inserted via a second likelihood ratio as shown in Equation 7.2. The final discriminant for the bb versus b training in the RecoVertex category after the bias has been applied is shown in Figure 7.12. Applying the bias introduces spikes in the distribution as can be expected due to different turn on values in each p_T - η bin and finer binning.

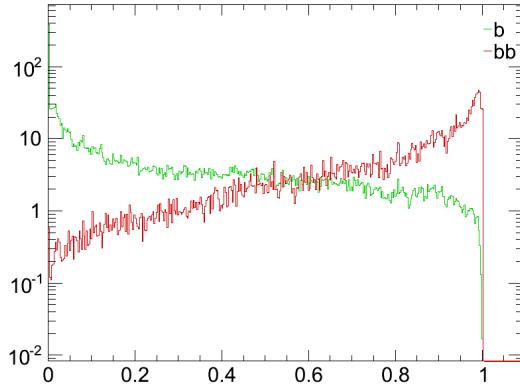


Figure 7.12: The final discriminant after the biases have been applied. Compared to the distributions shown in Figure 7.11 there are more spikes present here due to the different biases in each p_T - η bin and finer binning.

Final discriminant

There is one network discriminant output for each training. In the training of the default CSV, the b vs. c and b vs. dsg training were combined into one single discriminant at the end. This was done by adding the different discriminator contributions linearly with a weight based on the flavour fraction of c and d, u, s, g jets in the decay of W bosons into quarks. The resulting fraction is 0.25 for b versus c jets and 0.75 for b versus dsg jets. In our case, the training is performed separately for bb versus b and bb versus cdsg jets. So far these trainings are not

combined. The question is ultimately whether we want a dedicated bb versus “all” or a bb versus b tagger using standard b-tagging algorithms for b versus light-flavour discrimination. One solution would be to use a 2D discriminant as illustrated in Figure 7.13 where the user applies a cut in “bb vs. b”-“b vs. cdusg” space for desired bb/b purity simultaneously. If

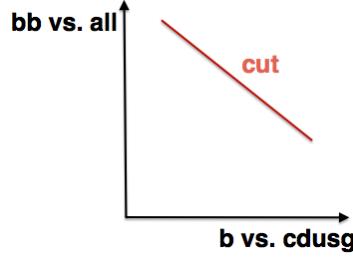


Figure 7.13: A possible double b-tagging discriminant where the user applies a cut in 2D space for desired b and bb jet purity simultaneously.

one wishes to combine the bb vs. b and bb vs. cdusg trainings, one could possibly take the combination coefficient from a $t\bar{t}$ -sample. This yields 0.30 for the bb versus b and 0.70 for the bb versus cdusg contributions. So far this has not been done. The final discriminants after the training for all vertex categories can be seen in Figure 7.14. The discriminating power is higher in the RecoVertex category than in the NoVertex and PseudoVertex categories.

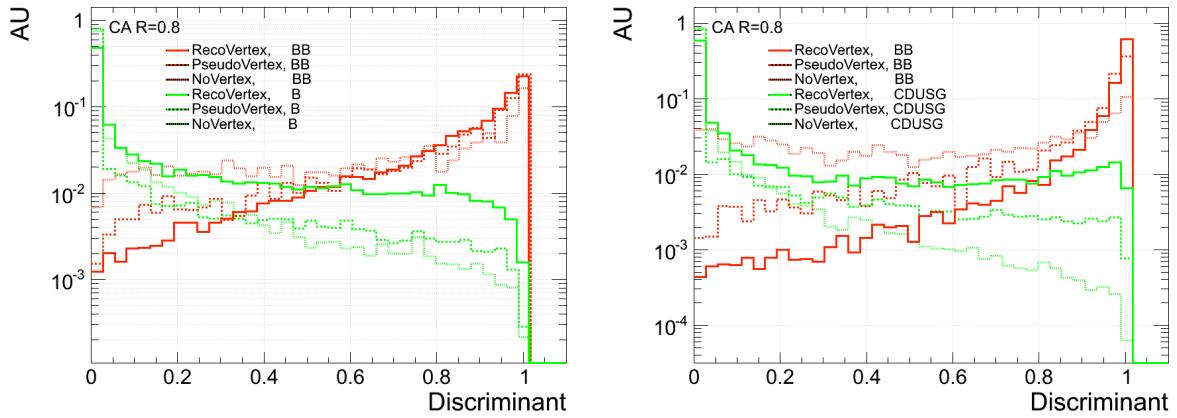


Figure 7.14: The discriminant after the training has been performed for the bb versus b training (left) and the bb versus cdusg training (right). Here shown for the three different vertex category trainings separately.

Chapter 8

Results

After the training is performed and the output network layout is stored, the double b-tagging algorithm is ready to be validated. This involves looking at the performance of the b-tagging algorithm applied to an independent Monte Carlo sample where the flavour generator information is available. There is full flexibility in the validation in terms of defining what jet algorithm, jet cone, momentum range etc. to be used for the jets to be validated, completely independent of what sample the training is performed on. To study the performance of the double b-tagging algorithm, the fraction of bb jets tagged by the algorithm over the total number of bb jets in the sample, as a function of cut on the discriminant, is calculated. The corresponding efficiencies, or “mistagging” efficiencies, for b and light flavoured jets are also computed in order to quantify how many non bb-flavoured jets would be tagged as bb jets. The purity of the final sample would then be one minus the fraction of fake jets tagged. The best way to compare different tagging performances is in the form of a ROC (Reciever Operating Characteristics) curve. This plots the signal efficiency versus the mistagging efficiency in one graph. That makes it easy to see what algorithm yields the highest signal efficiency for a given purity. The final results after validation of the double b-tagging algorithm are presented in the following

8.1 Validation of training framework

Before beginning the retraining of the CSV for the double b-tagging algorithm, the standard CSV training framework has been tested and validated. This involves repeating the training of the default CSV with the goal of achieving the same results. The details of the CSV training will not be outlined here and can be found in [51]. For this study all jets (b, c and dusg) in the training samples are taken from the same QCD simulated samples as were used for the default CSV training. The training is performed separately for b versus c and b versus light flavoured jets and linearly combined to one final discriminant in the end with weights of 0.25 and 0.75, respectively. The efficiency of the newly trained algorithm and the standard CSV algorithm is then tested simultaneously on $t\bar{t}$ events and their different performances compared. We would expect them to be identical. The b jet tagging efficiency versus mistagging efficiency for c- and light-flavoured jets respectively are shown in Figure 8.1. The agreement between the newly trained and the default CSV trainings are good and we consider the framework validated. The training and validation of the standard CSV algorithm is performed on AK5 jets using a default JTA cone of R=0.3. The results presented in the following are obtained from the dedicated double b-tagging training only.

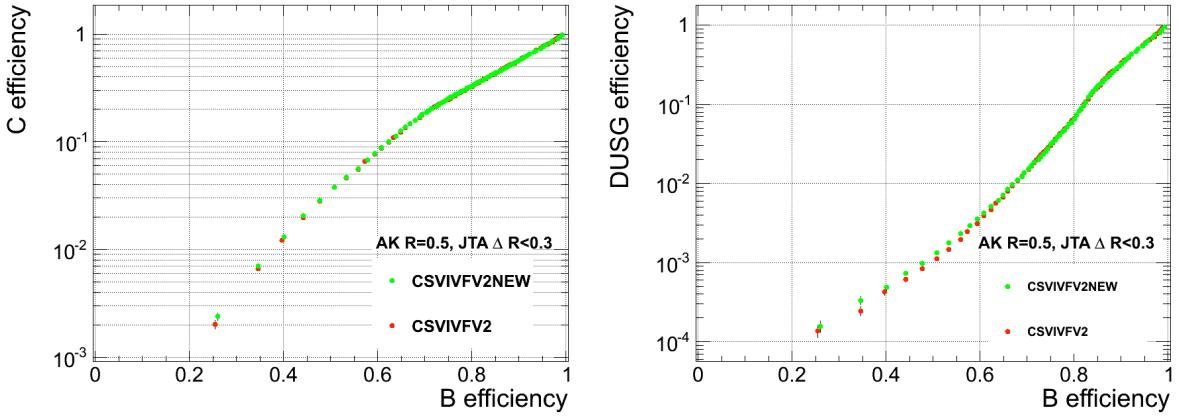


Figure 8.1: Validation of the CSV framework. The green curve is the new training to be validated and the red is the standard CSV algorithm. The different trainings should be the same and give equal performance curves.

8.2 Performance of the double b-tagging algorithm

The discriminant distributions after evaluation of the double b-tagging trainings are shown in Figure 8.2. The trainings have been performed using bb jets from the $R(M_R=1.0 \text{ TeV})$ sample, and QCD for the b/light flavoured jets (see samples listed in Section 5.1). The different trainings are then validated on $R(M_R=1.1 \text{ TeV}) \rightarrow \text{HH} \rightarrow 4\text{b}$. The training as well as the validation is using CA8 jets with a JTA cone of $R=0.8$. There is good separation power between bb and b/light-flavoured jets for both trainings. The algorithm trained to discriminate bb from b jets has good separation power for bb vs. cdusg jets as well without being trained for this purpose, while the algorithm trained for bb versus cdusg jets is less sensitive to bb versus b differences. The b and

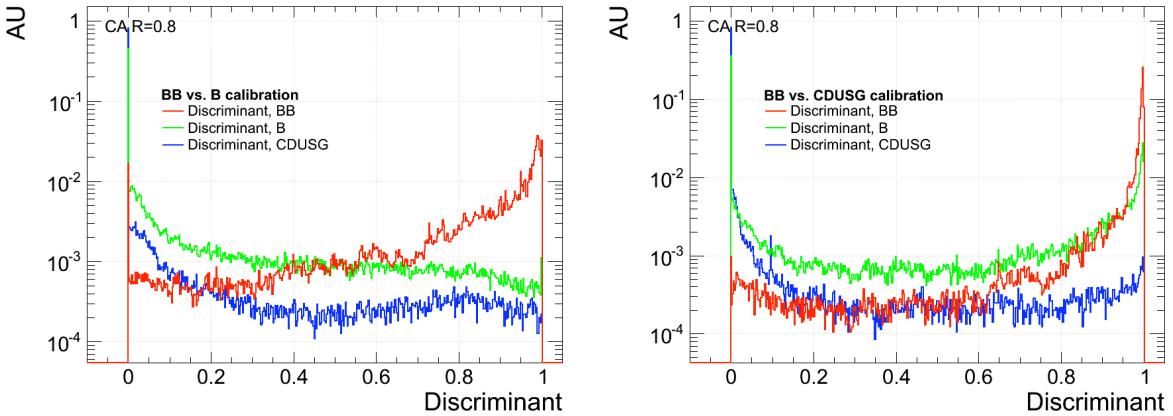


Figure 8.2: Discriminant output after evaluation of the bb vs. b training (left) and the bb versus c/light-flavoured training (right). The discriminant for all jet flavours is shown. The bb vs. b training has good discriminating power both for bb vs. b and bb vs. c/light flavoured jets. The dedicated bb vs. cdusg training performs well for light flavoured jets but fails to separate bb and b jets.

cdusg distributions for both cases peak rather sharply at zero. This means that there is a lot of

sensitivity in one single bin and that a significant amount of background events are removed by placing a cut after this bin.

8.2.1 bb versus b-flavoured jets

Figure 8.3 shows the tagging efficiency versus cut on the discriminant using the bb versus b training on bb and b jets. There is a sharp drop in the b jet mistagging efficiency from 100% tagging efficiency to around 65% after applying a cut on the first bin which reflects the sharp peak at 0 in the discriminant distribution. The plot on the right in Figure 8.3 shows the two curves plotted against each other illustrating the bb jet efficiency for a given b jet purity requirement. For a loose cut on the discriminant corresponding to a 10% mistagging rate, the bb tagging

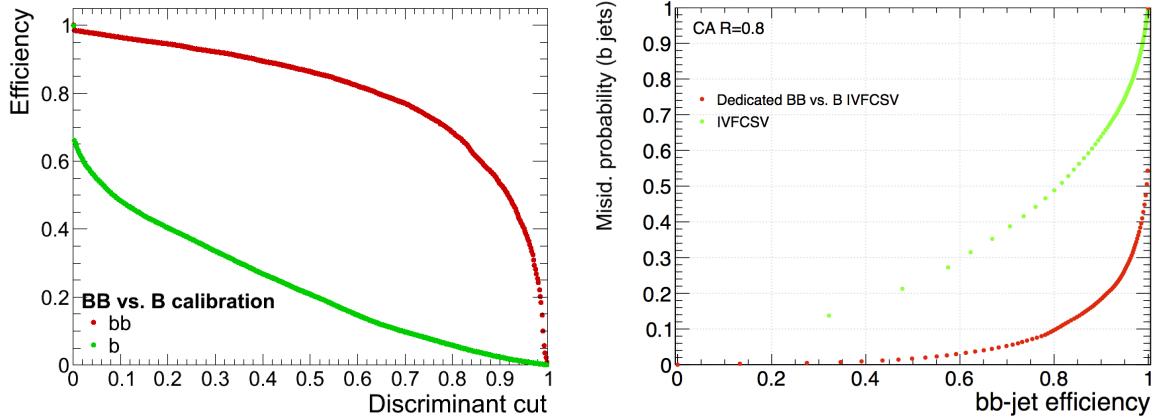


Figure 8.3: The efficiency versus cut on the discriminant for bb and b jets (left) and the same curves plotted against each other showing the bb tagging efficiency versus b jet mistagging rate (right). Here for bb and b jets from $R(M_R=1.1 \text{ TeV}) \rightarrow HH \rightarrow 4b$. The training is performed using bb jets from a $R(M_R=1.1 \text{ TeV})$ sample and b jets from the QCD background.

efficiency is almost 80%. For a medium working point (1% mistagging rate) the efficiency drops to 40%.

8.2.2 Training with different mass points

The double b-tagging algorithm evaluated above has been trained using one Radion sample with a mass point of $M_R=1.0 \text{ TeV}$. To check if there is lack of sensitivity caused by limited statistics, a $R(M_R=1.1 \text{ TeV}) \rightarrow HH \rightarrow 4b$ is added to the $R(M_R=1.0 \text{ TeV})$ training sample and the training is repeated. As the mass points are so close to one another, most of the added jets fall into the same training bins as the old ones. Already scarcely populated bins are doubled in statistics, but the very low- and high- p_T regions stay mostly as they were using only one sample. The training has also been performed using a combination of 5 different mass points: the two Radion samples as before, plus three Graviton ($M_R = 1.0 + 1.5 + 2.0 \text{ TeV}$) samples where $G(X \text{ TeV}) \rightarrow HH \rightarrow b\bar{b}b\bar{b}$. There are around 20k events in each Graviton sample, and 100k events in the Radion samples. The statistics in the Graviton samples are too low to compute the bias (see Section 7.2.5) as it leads to empty p_T - η bins and an undefined bias. The bias is therefore still taken from the $R(M_R = 1 \text{ TeV})$ sample. The performance of the b-tagging algorithm has been looked at for two different samples: the $R(M_R = 1.1 \text{ TeV})$ sample as before and on a $G(M_R = 2.0 \text{ TeV}) \rightarrow HH \rightarrow b\bar{b}b\bar{b}$ sample. The performance of the 3 different trainings using 3 different mass point combinations can be seen in Figure 8.4, the left plot for $R(M_R = 1.1 \text{ TeV})$ and the right for $G(M_R = 2 \text{ TeV})$.

When tagging bb jets from the $R(M_R = 1.1 \text{ TeV})$ sample there is a small gain in performance

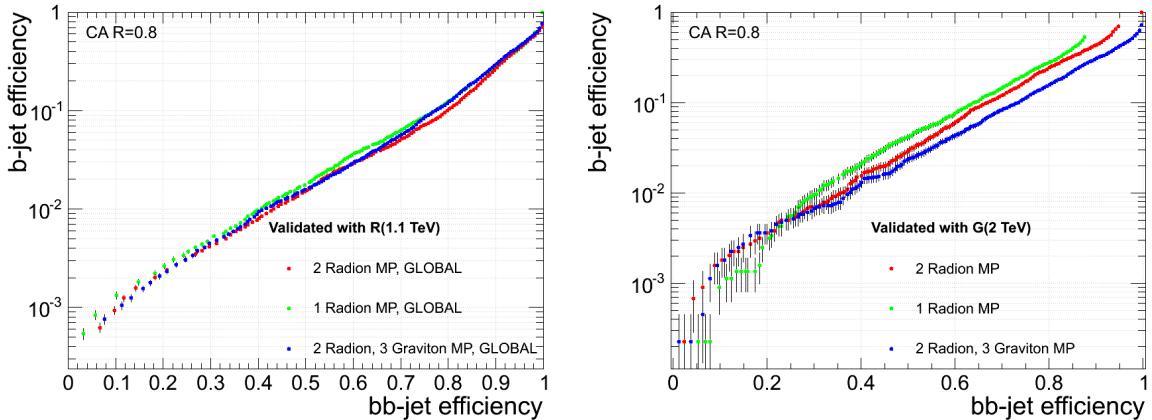


Figure 8.4: Performance using three different combinations of mass points for the training: Radion ($M_R = 1.0 \text{ TeV}$), Radion ($M_R = 1.0 + 1.1 \text{ TeV}$) and Radion ($M_R = 1.0 + 1.1 \text{ TeV}$) +3 Graviton ($M_R = 1.0 + 1.5 + 2.0 \text{ TeV}$). The evaluation on jets from a $R(M_R = 1.1 \text{ TeV})$ sample is shown to the left and on jets from $G(M_R = 2.0 \text{ TeV})$ to the right. For bb jets from a $R(1 \text{ TeV}) \rightarrow HH \rightarrow 4b$ decay the performance stays the same and there is no significant gain in performance by increasing the statistics. For bb/b jets coming from the 2 TeV Graviton however, the double b-tagging algorithm trained using higher mass points performs better.

by training with 2 Radion mass points compared to using only one. This is to be expected when training with the same sample as the one used for validation. The gain is around 5% for a CSV loose working point. Adding three higher mass points to the training does not improve the performance when validating on a $R(M_R = 1.1 \text{ TeV})$ sample, but does not degrade it significantly either. For validation on bb jets from the 2 TeV Graviton sample the performance improves when other samples of higher mass are added to the training. For a 10% mistagging rate there is a gain of almost 10% when using a double b-tagger trained with multiple mass points, compared to using only one 1 TeV training sample. The error bars at high purity are large due to lack of statistics in the $G(M_R = 2.0 \text{ TeV})$ sample. The question is whether one wished to have one double b-tagger that can be used for a large range of searches, or dedicated algorithms for specific p_T ranges. The double b-tagging algorithm trained using two Radion mass points is set as the default in this thesis and used in the following as it has the overall best performance for both cases.

8.2.3 Varying $p_T\text{-}\eta$ training bins

As explained in Section 7.2.4, the training is performed in different $p_T\text{-}\eta$ bins in order to increase the sensitivity. The performance using the default CSV training bins and the new training bins optimized for the bb jet p_T and η distribution is shown in Figure 8.5. For a loose working point corresponding to a 10% mistagging rate, the training performed with the new $p_T\text{-}\eta$ binning has a 5% higher bb jet tagging efficiency than when using the default bins. For a medium working point (1.0% mistagging rate) there is a 10% efficiency gain using the new training bins. In the high-efficiency region the training performed with the default $p_T\text{-}\eta$ binning performs slightly better than the training using the new bins, indicating that the binning can be optimized further.

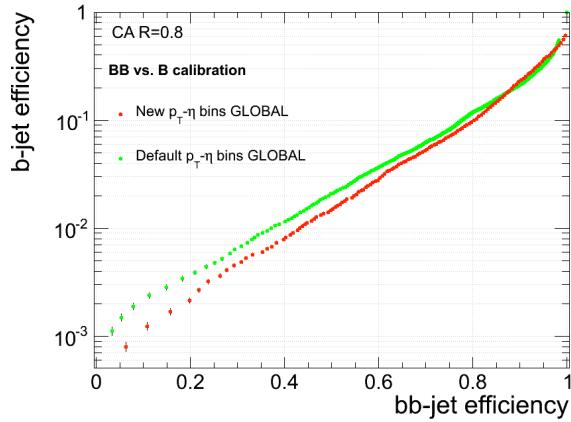


Figure 8.5: bb tagging efficiency versus b jet mistagging rate for the double b-tagger training using the default (green) and new (red) p_T - η binning described in Section 7.2.4.

8.2.4 p_T -dependence

The p_T -distributions of the bb, b and light-flavoured jets that enter the validation differs significantly. As tagging performance is highly p_T -dependent, it is important to check the performance in different p_T -ranges to ensure a fair comparison. The p_T -distribution of the jets coming from the R($M_R = 1.1$ TeV) sample used in the validation can be seen in Figure 8.6. The bb tagging

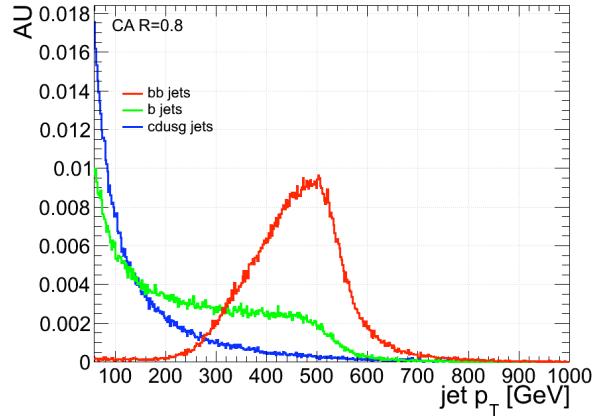


Figure 8.6: The p_T spectrum for bb (red), b (green) and c/light flavoured (blue) jets. The p_T -distribution for signal (bb) and background (b/cdusg) differs significantly and most of the bb jets have $200 \text{ GeV} < p_T < 700 \text{ GeV}$.

efficiency versus b mistagging rate for different p_T -ranges are shown in Figure 8.7. The three different curves show the efficiency using all jets , the efficiency for jets with $150 \text{ GeV} < p_T < 400 \text{ GeV}$ and for jets with $400 \text{ GeV} < p_T < 600 \text{ GeV}$. The performance is stable as a function of p_T with the three performance curves almost overlapping.

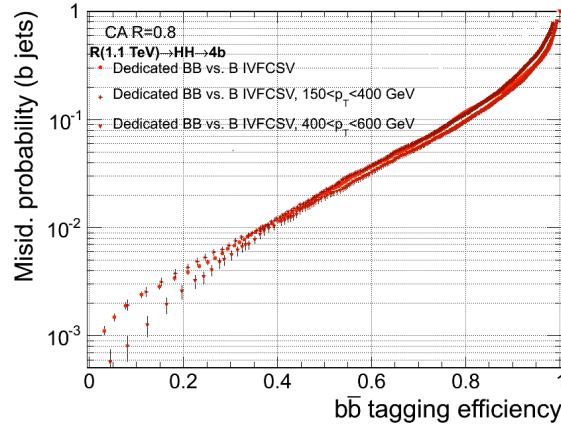


Figure 8.7: bb jet efficiency versus b jet mistagging rate. The plot to the left shows the performance of the dedicated double b-tagger for all jets, jets with $150 \text{ GeV} < p_T < 400 \text{ GeV}$ and for jets with $400 \text{ GeV} < p_T < 600 \text{ GeV}$. The performance is stable as function of p_T , with the three curves almost overlapping.

8.2.5 bb versus light-flavoured jets

The bb efficiency versus light-flavoured jet mistagging rate is shown in Figure 8.8. Here a combination of $R(M_R = 1.1 \text{ TeV}) \rightarrow HH \rightarrow 4b$ and a $t\bar{t}$ sample is used for the validation in order to increase the light-flavoured jet statistics. For a medium working point, the bb efficiency is almost 80%, around 10% better than what is achieved with the current CSV. However, as mentioned previously, the current CSV is trained using AK5 jets making a direct comparison difficult. For completeness, applying the bb vs. b training to cdusg jets has also been studied and

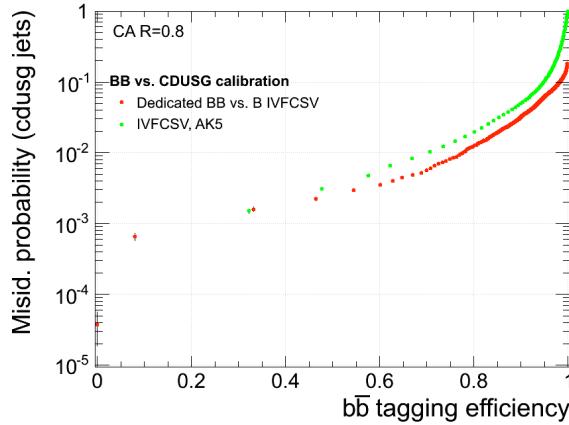


Figure 8.8: Discriminant output after validation using the bb vs. c/light training.

the corresponding ROC curve is shown in Figure 8.9. As was already seen in the discriminator distributions in Figure 8.2, the bb vs. b training does have some discriminating power for bb versus light-flavoured jets even though the algorithm is not trained for this. For a medium working point the bb jet efficiency is around 55%, which is slight worse than when using the default CSV tagging algorithm.

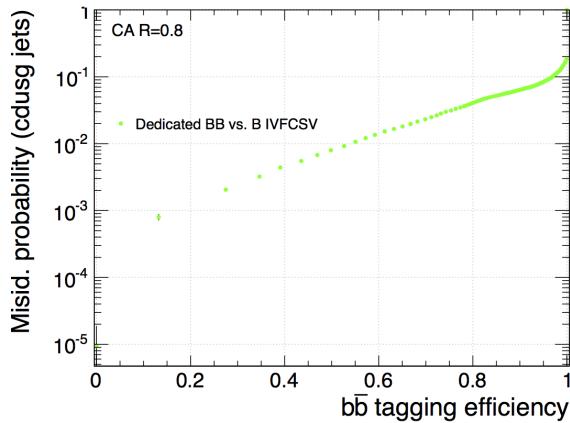


Figure 8.9: bb jet tagging efficiency versus light-flavoured jet mistagging rate using the bb vs. b training. There is good discrimination power despite not being specifically trained for this purpose.

8.2.6 Performance on larger jets

The performance of the double b-tagger on jets with larger cone sizes has also been studied. Here the same training as before, using CA8 jets, is validated on CA1.2 and CA1.5 jets. The results are shown in Figure 8.10. There is a strong degradation in performance as the size of the

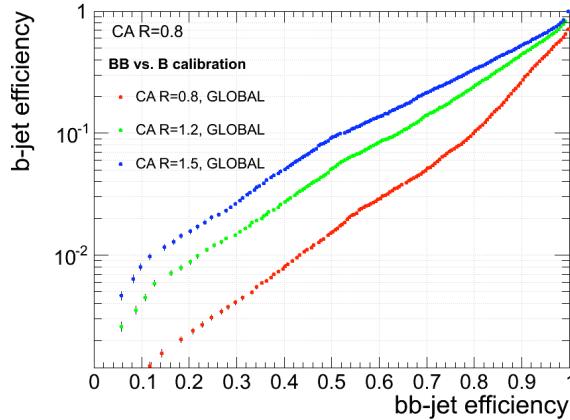


Figure 8.10: The performance of the double b-tagger on CA8 (red), CA1.2 (green) and CA1.5 (blue) jets. There is a significant decrease in efficiency for a given mistagging rate when increasing the jet R parameter.

jet cone increases. This could be either due to a decrease in the bb jet efficiency or an increase in the b jet efficiency. It is therefore useful to look at the efficiency versus discriminant cut for the bb and b jets separately. This is shown in Figure 8.11. The top three curves are the bb jet efficiency for the three different jet cone sizes, and the three bottom curves are the b jet efficiency. The bb jet efficiency stays almost the same when increasing the jet cones, but there is a large increase in b jet mistagging rate. The reason for this might be due to a larger pollution of misplaced tracks coming from other interactions in the event, leading the algorithm to believe that the jet contains several vertices, displaced tracks etc. For the case of highly boosted Higgs bosons decaying to b quarks, choosing a larger jet cone is not strictly necessary as the double-b

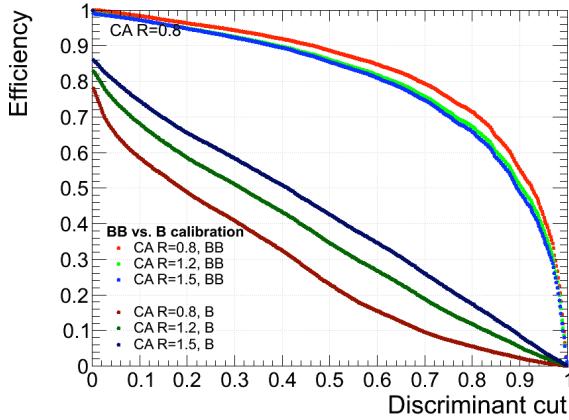


Figure 8.11: The efficiency versus discriminant cut for bb jets (three top curves) and b jets (three bottom curves). The bb tagging efficiency stays the same for all three jet sizes, but there is an increase in the b mistagging rate when increasing the jet cone.

tagging algorithm is intended for cases where the particles are so collimated that their jet cones are inseparable and subjet b-tagging becomes difficult. This implies that the tracks will be collimated in space already and a cone size of $R=0.8$ should be enough.

8.2.7 bb versus b for gluon splitting

Despite being designed for tagging bb jets in boosted topologies, the performance of the double b-tagging algorithm has also been studied on bb jets coming from gluon splitting (see Section 2.1.1). The reason for this is that b/bb jets stemming from QCD is one of the the main backgrounds for $H \rightarrow b\bar{b}$ searches and we need to understand the performance of the double b-tagger for these types of events. b quarks coming gluon splitting are usually softer than b quarks stemming from a Higgs boson decay and the b jets might end up in the low- p_T region where the training statistics are low. The discriminant after applying the bb vs. b training to jets from a QCD sample can be seen in Figure 8.12 (here the bb jets are assumed to come from gluon splitting events only). The left plot shows the discriminant distribution for all jets, while the plot on the right is for jets with $250 \text{ GeV} < p_T < 400 \text{ GeV}$. Here the separation power between bb and b jets is significantly reduced. There is a sharp peak at zero for both signal and background, implying that the algorithm is unable to assign a value to the bb jets. As the effect is seen also in the higher p_T window, this cannot be due to lack of training on low- p_T jets. It is known from previous studies that b quarks from gluon splitting have different variable distributions than b quarks originating from a Higgs decay [39]. This is also the case for this study as can be seen in Figure 8.13 where some of the variable distributions entering the training are shown for bb jets coming from a boosted Higgs boson and for bb jets from gluon splitting. The variables have a significantly different shape. The final ROC curves for bb jets from gluon splitting versus b jets is shown in Figure 8.14. For a loose working point the bb efficiency is only around 37% for gluons splitting jets compared to 80% for bb jets from the Higgs decay.

8.2.8 $g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$

The degradation in performance for bb jets from gluon splitting, is, as previously mentioned, an advantage in the sense that this would be the largest background in analyses looking for boosted Higgs bosons decaying to b quarks. It is therefore useful to look at the $H \rightarrow b\bar{b}$ versus

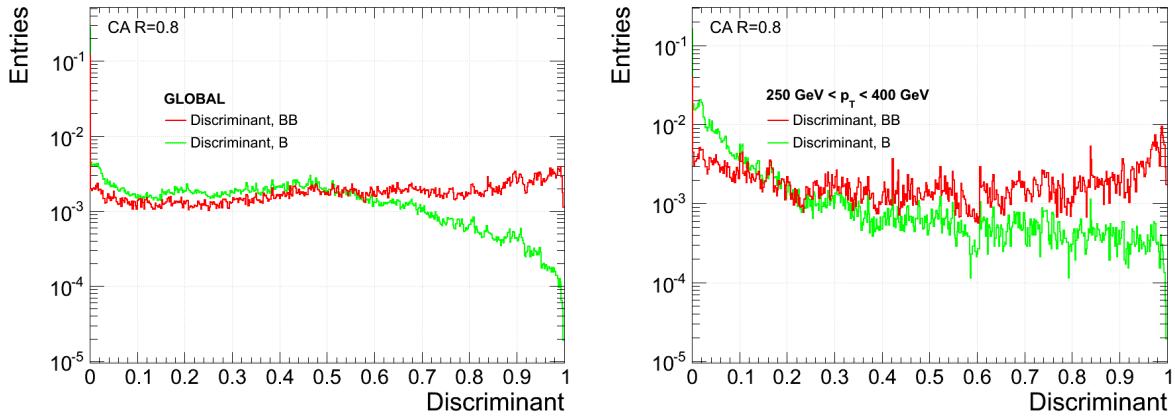


Figure 8.12: The discriminant distribution using the double b-tagger on bb-jets from gluon splitting. The left plot shows the discriminant for all jets and the right plot for jets with $250 \text{ GeV} < p_T < 400 \text{ GeV}$. The discriminating power is significantly lower for bb jets from $g \rightarrow b\bar{b}$ than for bb jets coming from a boosted Higgs decay in both cases (see Figure 8.2).

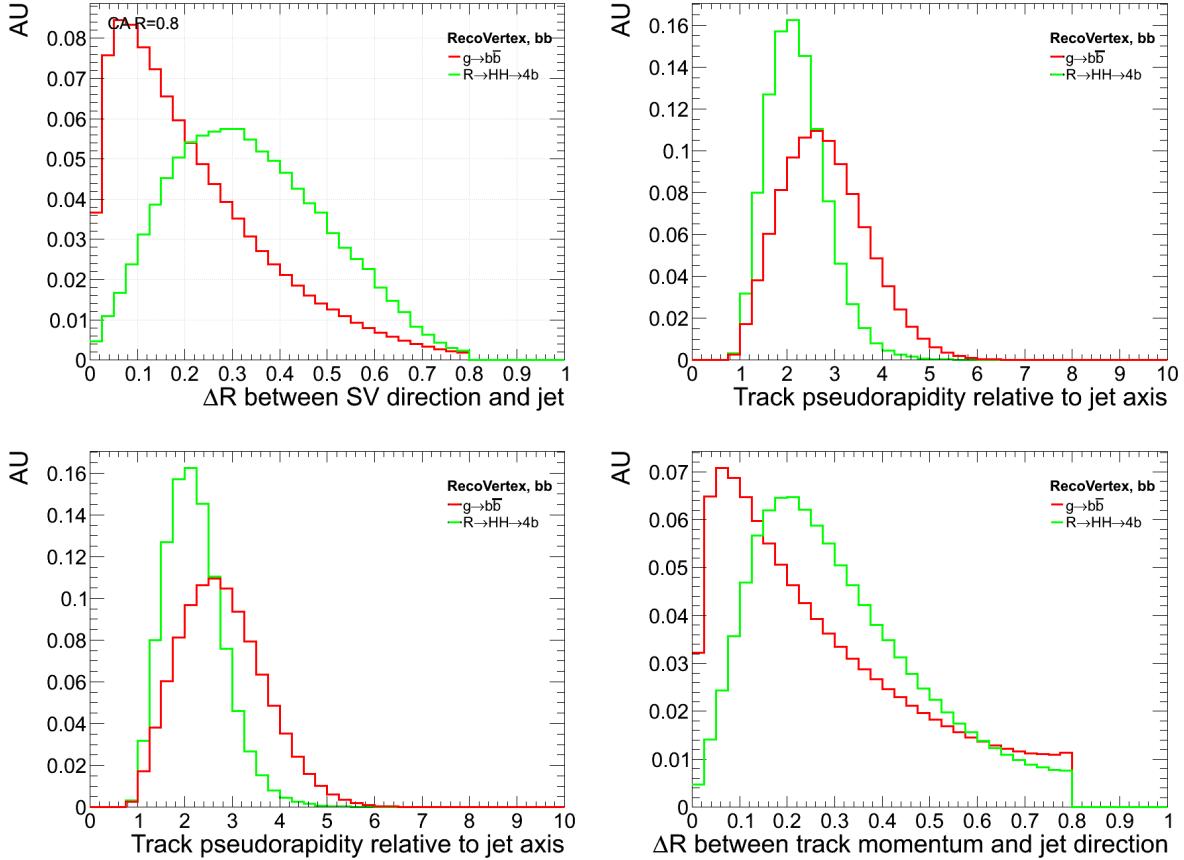


Figure 8.13: Comparison of variable distributions for $g \rightarrow b\bar{b}$ and $H \rightarrow b\bar{b}$. bb jets coming from gluon splitting has a significantly different distribution than bb jets from a boosted Higgs boson.

$g \rightarrow b\bar{b}$ efficiency. This is plotted in Figure A.3 for jets with $200 \text{ GeV} < p_T < 700 \text{ GeV}$. For a 10%

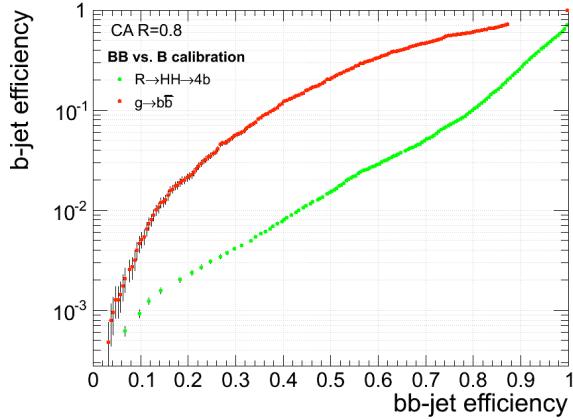


Figure 8.14: Tagging efficiency for bb jets coming from gluon splitting versus b jet mistagging probability (red). The b jet tagging efficiency is from the same QCD sample. Shown for comparison is the bb jet tagging efficiency using bb and b jets from $H \rightarrow b\bar{b}$.

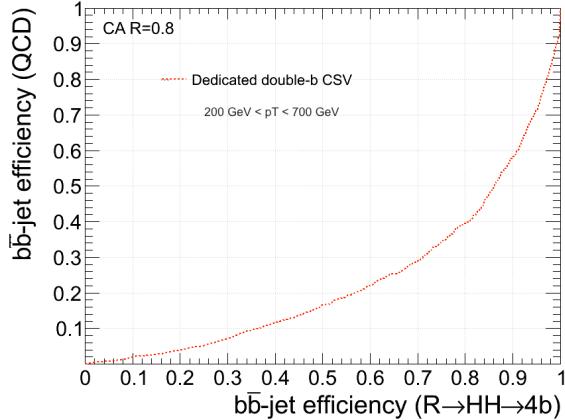


Figure 8.15: Tagging efficiency for bb jets from gluon splitting versus bb jets from a Higgs decay for all jets for jets with $200 \text{ GeV} < p_T < 700 \text{ GeV}$.

$g \rightarrow b\bar{b}$ mistagging rate, the double b-tagging algorithm achieves a 37% $H \rightarrow b\bar{b}$ efficiency despite not being training especially for this purpose.

8.2.9 $H \rightarrow b\bar{b}$ versus b jets from QCD

In order to check whether we introduce any bias when using b jets from the Radion sample as background, we have also looked at the performance using bb jets from $H \rightarrow b\bar{b}$ and b jets from a QCD sample. This is shown in Figure 8.16. In the high-purity region, the bb jet tagging efficiency for a given mistagging rate is higher when using QCD b jets than when using b jets from $H \rightarrow b\bar{b}$. For a medium working point (1% mistagging rate) the efficiency gain is almost 10%. In some ways; this is a more realistic scenario, as the QCD and $t\bar{t}$ background will be the main backgrounds when using the double b-tagging algorithm in analyses.

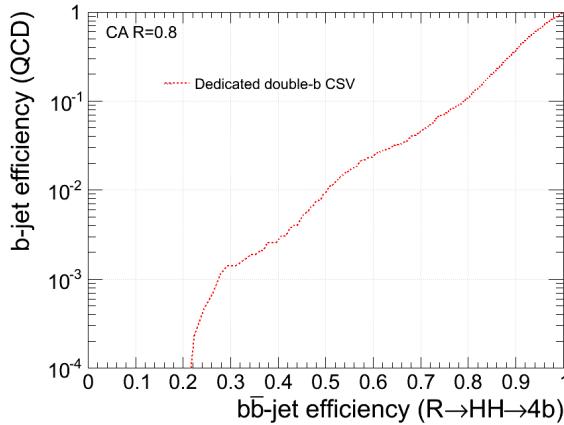


Figure 8.16: Tagging efficiency for $b\bar{b}$ jets from a boosted Higgs decay versus b jet mistagging probability where the b jets are from the QCD background. The performance is slightly better using QCD b jets as background than when using b jets from the $R \rightarrow HH \rightarrow b\bar{b}$ sample.

8.2.10 Adding additional vertex training categories

The algorithm presented so far has been trained with variables not optimized for discriminating jets containing two b quarks from jets containing one. For instance, variables such as the vertex mass use the sum of all tracks coming from all secondary vertices in the jet without splitting them into separate entities. Other vertex related variables, like the distance between the secondary vertex and the jet axis, stores information for the best reconstructed secondary vertex only and disregards all information regarding multiple vertices.

In an attempt of further enhancing the $b\bar{b}$ versus b separation, a fourth vertex training category has been implemented; the RecoRecoVertex category. The old RecoVertex category is split into two fully disjoint subsets: the RecoVertex category which contains jets with one reconstructed secondary vertex($SV==1$), and the RecoRecoVertex category containing only jets with more than one reconstructed SV ($SV>1$). In this way, variables related to additional secondary vertices can be used in the training in order to increase the sensitivity for jets containing two b quarks. The implementation and training using the new vertex categories is first tested using the standard CSV algorithm trained to discriminate b from light flavoured jets. The results can be seen in Figure 8.17. There is not much improvement in performance by adding an additional training category for this case as can be expected (there are no $b\bar{b}$ jets). For the b versus light-flavoured discrimination, the performance stays the same. However, there is a slight gain in performance using the new vertex categories for b versus c discrimination. The reason for this might be due to the fact that the b quark mainly decays to a c quark and a W boson. The c quark, like the b , also travels some distance before it decays leading to a second SV inside the jet cone as described in Section 2.1.1. Hence a b -flavoured jet is more likely to contain two secondary vertices than a c -flavoured jet. Having two b quarks in one jet might therefore lead to as many as four reconstructed vertices actually stemming from the b decay itself. Introducing the RecoRecoVertex category and storing vertex related variables for each vertex separately can therefore enhance the separation power. If one for instance compares the vertex mass of the second best vertex in the jet for a $b\bar{b}$ jet versus a b -flavoured jet, the mass is more likely to be higher and close to the b quark mass for a two- b jet.

The $b\bar{b}$ versus b training is then repeated using the new vertex training categories and its performance evaluated on jets from $R(M_R=1.1 \text{ TeV}) \rightarrow HH \rightarrow 4b$. This is shown in Figure 8.18. Using the default training categories seems to perform slightly better than when training with

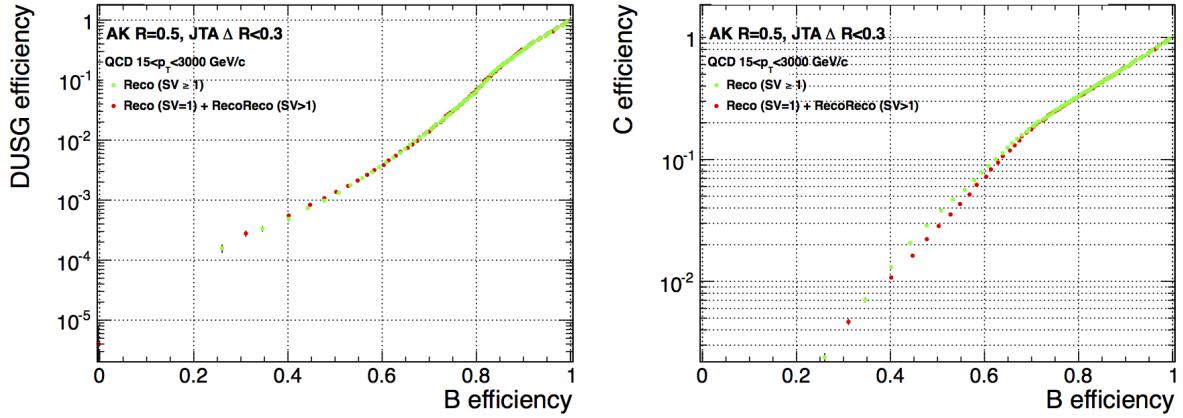


Figure 8.17: b jet tagging efficiency versus light-flavoured (left) and c-flavoured (right) jet mistagging rate, using default vertex training categories (green) and new vertex training categories (red).

the RecoVertex category split into two subsets. in the very high-efficiency region however, the curves cross again and there is a slight gain in efficiency using the new categories.

As most of the bb jets have a reconstructed secondary vertex (around 92%), we focus on the

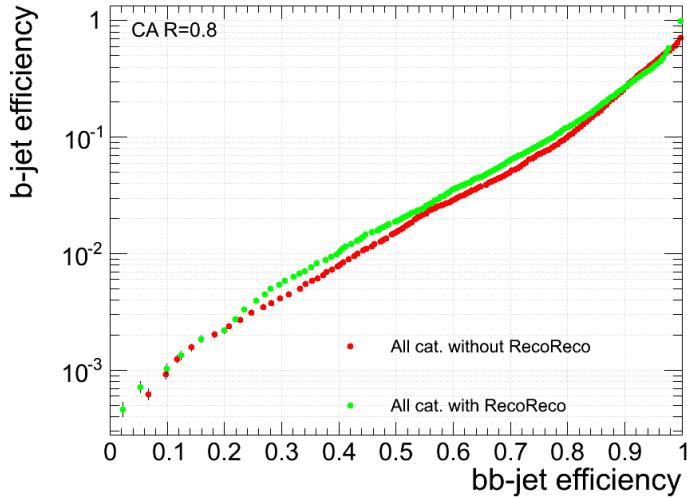


Figure 8.18: bb tagging efficiency versus b jet mistagging rate using the default training categories No, Pseudo and Reco (red) and the corresponding performance using the new No, Pseudo, Reco and RecoReco categories (green). The performance using the default training categories is slightly better than using the new training categories.

RecoVertex category in order to improve the training using the new vertex categories. We want to look at the performance using the default RecoVertex category and the new Reco+RecoReco training categories only, disregarding the No and PseudoVertex trainings as only 8% of the bb jets enter into these categories. Here the training has been performed using only jets with at least one secondary vertex, while the validation uses all jets. The maximum bb efficiency the algorithms can reach is limited by the number of jets that fall in the No, Pseudo and Reco/RecoReco category as the efficiency uses all bb jets in the denominator. Passing trainings for single vertex

categories will leave “unidentifiable” jets in a discriminant underflow bin (the algorithm does not know that to do with a PseudoVertex jet if only the RecoVertex training is passed). Figure 8.19 shows the performances using the default RecoVertex category and the new Reco+RecoReco training categories only. The performance using the default RecoVertex category (all jets with at least one secondary vertex) only is a lot better than when using all training categories as this category contains most of the bb jets and hence has higher sensitivity. Here the maximum b jet efficiency only reaches around 70% as 30% of the jets fall in the NoVertex and PseudoVertex categories and therefore are unclassified by the double b-tagging algorithm. Around 8% of the bb jets fall in the No and PseudoVertex categories and the bb jet tagging efficiency therefore has a cut off at 92%. The bb tagging efficiency corresponding to a 10% mistagging rate is around 90%. When passing the Reco+RecoReco trainings only, however, something seems to fail within the framework. Here the maximum bb efficiency drops down to around 35% despite the fact that 60% percent of the bb jets fall in the RecoRecoVertex category and 32% in the Reco category. Now almost 60% of the bb jets in addition to the 8% in the Pseudo and NoVertex categories are unclassifiable for the algorithm end up in a discriminant underflow bin. This keeps the Reco+RecoReco combined trainings from reaching the same 92% bb efficiency point as the default RecoVertex category with no cut on the discriminant. Also the maximum b jet efficiency drops by around 10%. The validation uses the same code to place jets in vertex categories as the

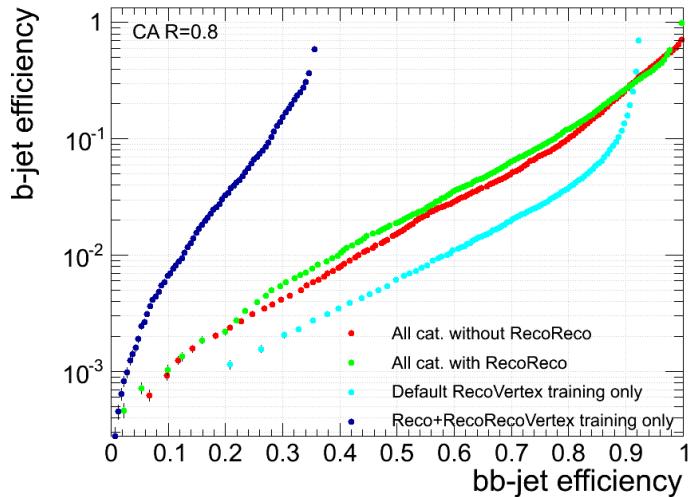


Figure 8.19: The bb tagging efficiency versus b jet mistagging rate using the default RecoVertex (cyan) and the new Reco+RecoReco (blue) training categories . The performances using all categories (red and green) are the same as shown in Figure 8.18 and are there for comparison. Something fail when passing only the Reco+RecoReco trainings.

training and generation of training trees uses. The implementation of the new vertex categories has been checked in order to make sure it is implemented correctly. The number of secondary vertices contained in each jet for the new RecoVertex and RecoRecoVertex categories are shown in Figure 8.20. As can be seen from the plots, all Reco jets (left) have one SV only, while jets in the RecoRecoVertex category (right) has two or more secondary vertices.

Further checks have been performed in order to explain what fails with the new vertex training categories and this is explained in detail in the Appendix. The conclusion after the studies is that both trainings are used to tag jets with one secondary vertex and jets with multiple vertices. However, they do not always tag the same jets so combining them do provide a higher efficiency than when passing only one training. After discussing with other CSV experts there

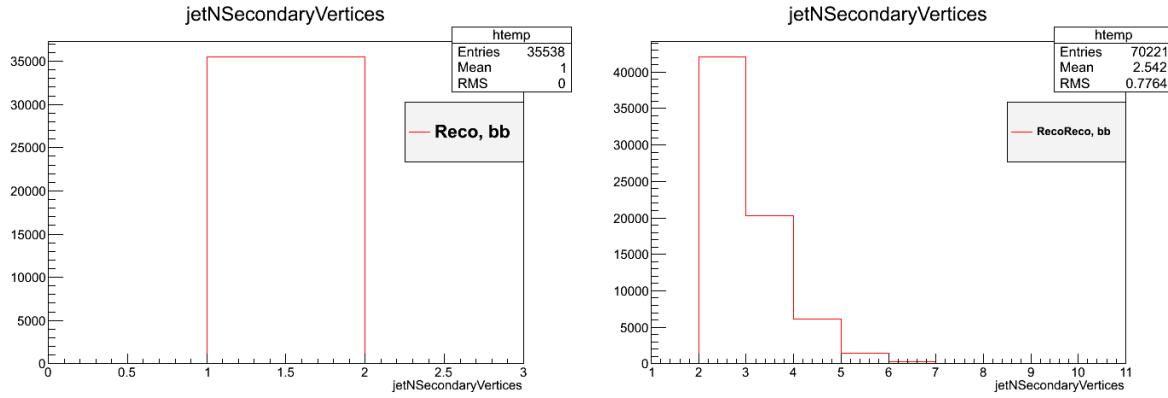


Figure 8.20: Number of secondary vertices in each jet for the new RecoVertex (left) and RecoRecoVertex (right) categories.

is general agreement that validating training categories separately is not trivial to do in the current training framework. With the impressive performance achieved using the default vertex categories only, implementation of an additional vertex category have not be pursued further.

Table 8.1 summarizes the bb jet tagging efficiency for three different mistagging rates for bb vs. b, bb vs. light and $g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$ discrimination. For bb versus b jet separation, the dedicated double b-tagger achieves a 30% higher tagging efficiency than the standard CSV algorithm for a 10% mistagging rate. For bb versus light-flavoured jets, the tagging efficiency is 10% higher than for the default CSV. The final algorithm is also capable of discriminating between two-b jets coming from gluon splitting and two-b jets coming from $H \rightarrow b\bar{b}$ with a 37% $H \rightarrow b\bar{b}$ tagging efficiency for a 10% $g \rightarrow b\bar{b}$ mistagging rate.

Mistagging rate	$H \rightarrow b\bar{b}$ tagging efficiency		
	bb vs. b	bb vs. light	$g \rightarrow b\bar{b}$ vs. $H \rightarrow b\bar{b}$
10%	80/79%	95%	37%
1.0%	44/50%	78%	5%
0.1%	11/26%	15%	-

Table 8.1: The bb jet efficiency for three given mistagging rates corresponding to a CSV loose (10%), medium (1.0%) and tight (0.1%) working point. Here for bb versus light, bb versus b and $g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$ discrimination. The two numbers for bb vs. b are the different efficiencies when evaluating using b jets from $H \rightarrow b\bar{b}$ and from QCD respectively.

Chapter 9

Conclusion

The training and validation of a neural network based b-tagging algorithm capable of tagging jets containing two b quarks has been performed. The algorithm is especially designed for searches in boosted topologies where the b quarks are highly collinear and merged into a single jet. The presented algorithm achieves an 80% tagging efficiency for a 10% mistagging probability for two-b versus one b jet discrimination. This is 30% higher than what is achieved using current methods. The tagging efficiencies for a 1.0% and 0.1% mistagging rate reaches 50% and 26% respectively. For two-b versus light-flavoured jet discrimination, the algorithm has a 95% two-b jet tagging efficiency for a 10% mistagging rate, an improvement of 10% compared to what is achieved with standard b-tagging methods. At higher purity working points, the efficiencies for two-b versus light-flavoured jet discrimination are 78% and 15% for a 1.0% and 0.1% mistagging probability. The algorithm also allows to discriminate between two-b jets coming from a boosted Higgs decay and two-b jets stemming from the QCD background, reaching a 35% $H \rightarrow b\bar{b}$ tagging efficiency for a 10% mistagging probability.

Appendix A

The MVA Trainer and MVAComputer

A.1 Implementation of additional vertex categories

In order to further investigate the observed effect when passing the new Reco and RecoReco vertex trainings, the trainings are passed separately for validation. This would make it clear if one of the trainings has failed. Figure A.1 shows the performance when passing the new RecoVertex training (Nr. SV==1) only and the same when passing only the RecoRecoVertex training. Here they both reach the exact same maximum bb efficiency of around 32%. As 32% of the bb jets fall in the RecoVertex (SV==1) category, it appears as both trainings are tagging the same Reco jets, not knowing what do do with No, Pseudo and RecoReco jets. However, when combining the two trainings (red) there is a few percent gain in bb/b efficiency, indicating that they are not tagging the exact same jets in all cases. The performance using the default RecoVertex category it there for comparison. To check what jets the Reco and RecoRecoVertex

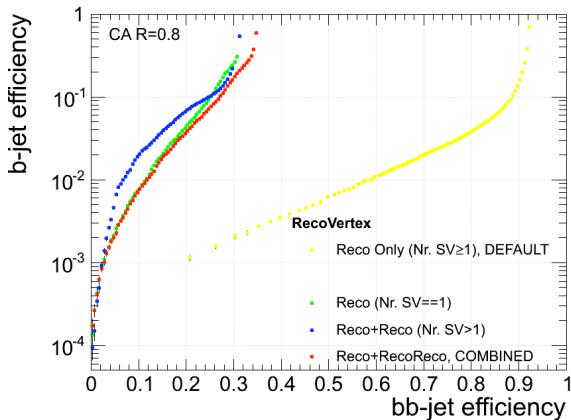


Figure A.1: Performance using the default RecoVertex (yellow) category versus using the new Reco (green) training only, RecoReco (blue) or a combination of them both (red). The combined Reco+RecoReco curve (red) should be equal or better than using the default RecoVertex category only, so something is clearly going wrong when combining the trainings.

training are actually tagging, the performance of the Reco and RecoReco trainings separately on jets containing only one secondary vertex, and performance on jets containing only more than one secondary vertex have been studied. The results are shown in Figure A.2. The denominator

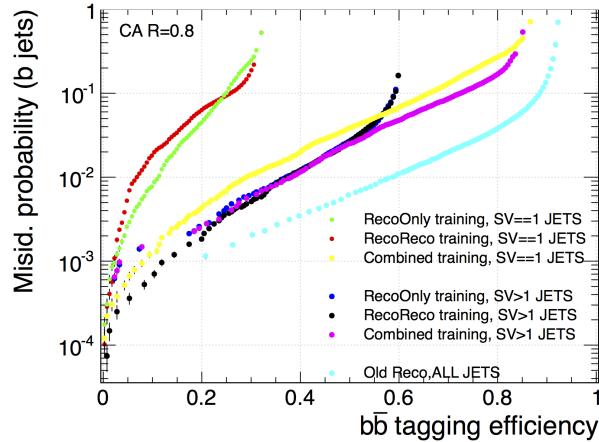


Figure A.2: Performances for the Reco (green), RecoReco (red) and the combined Reco+RecoReco (yellow) trainings validated on jets with no, a pseudo or *only one* secondary vertex reconstructed. The blue, black and purple curves are the corresponding trainings validated on jets with no, pseudo or *multiple* secondary vertices reconstructed. The dedicated Reco training performs best for jets containing only one secondary vertex and the dedicated RecoReco training performs best on jets containing multiple secondary vertices. However, both trainings seem to be tagging both jet types. The fact that they both reach the exact same bb tagging efficiency indicates they are even tagging the same jets. The yellow and purple curves are the efficiency combining the Reco and RecoReco trainings, which ideally should correspond or improve the training using the default RecoVertex category (cyan).

is using all bb jets in the sample. Using only the Reco or only the RecoReco training in the validation reaches a maximum bb efficiency of around 32% for both taggers, which is expected as 68% of the jets fall in the No, Pseudo and RecoReco category. However, for jets containing only one secondary vertex, we do not expect the RecoReco training to be used at all. The Reco training seems to perform a bit better than the RecoReco training in this case. The blue, black and purple curves are the corresponding trainings validated on jets with no, pseudo or multiple secondary vertices reconstructed. Here the dedicated RecoReco training performs the same as the training on Reco jets only. Both trainings are again applied to tag jets with multiple secondary vertices. The yellow and purple curves are the efficiency combining the Reco and RecoReco trainings for jets containing one and two SVs respectively, which ideally should correspond or improve the training using only the default RecoVertex category (cyan). Here it is surprising to see that there really is a gain in efficiency by using both trainings, so in some way they do seem to complement each other.

Between the CSV training experts there is general agreement that when passing a training for jets in the RecoVertex category, this training is only used on RecoVertex jets, and the same for the No and Pseudo categories. However, this is something that has not been properly looked into. If one tries to pass the PseudoVertex training only for instance, this is not allowed in the system as the PseudoVertex category is missing some variables that are defined only in the RecoVertex category and it does not know what to do when being passed a Reco jet. It appears like the trainings for each vertex category is not explicitly used on jets falling in the category it is trained for. The way to move further with the implementation of a RecoReco category, would be to switch the training to the Root TMVA framework. Here adding an additional vertex category should be quite straight forward, and the only problem to overcome would be how to get the training back into the CMS Offline Software for use in analyses.

A.2 Available processors

The variable preprocessors available within the MVATrainer and MVAComputer framework are:

- ProcMatrix: Linear decorrelation using a matrix rotation defined by Principal Component Analysis (PCA). (lossless)
- ProcNormalize: Flattening of value range to [0, 1]. Uses a pdf of all data points to redistribute the values equally in the range. Values outside of the range seen in the training dataset are clamped (lossless with respect to training dataset).
- ProcLikelihood (in “individual transformation mode”): Same as for regular ProcLikelihood, except that $S/(S + B)$ is computed for each variable individually and not combined into a single likelihood ratio (lossy if $S/(S + B)$ distribution is not strictly increasing or decreasing).

And the following classifiers:

- ProcLikelihood: Combined Likelihood Ratio using cubic spline representation of the variables’ pdfs. Assumes uncorrelated variables and is a $S/(S + B)$ combination of individual variable probabilities.
- ProcLinear: A simple linear discriminant from a χ^2 regression fit (Fisher’s Discriminant).
- ProcMLP: Available via plugin: An Artificial Neural Network (ANN) from the MLP-fit package, a simple feed-forward network with configurable hidden layers, logistic activation function and several backpropagation learning methods.
- ProcTMVA: Access to all methods from ROOT’s Toolkit for Multivariate Analysis Techniques (TMVA) package.

A.3 Biases

The biases (see Section 7.2.5) applied to the different vertex categories are listed in Table A.1 for the different vertex training categories and for bb versus b and bb versus c/light.

NoVertex, bb vs. b		NoVertex, bb vs. cdusg		PseudoVertex, bb vs. b	
Bin	Bias	Bin	Bias	Bin	Bias
0	0.713432	0	0.206692	0	1.16637
1	0.809937	1	0.267246	1	0.86864
2	0.79426	2	0.33831	2	0.595255
3	0.35448	3	0.0834466	3	0.465573
4	0.355709	4	0.101435	4	0.714126
5	0.412422	5	0.162792	5	1.62803
6	0.177686	6	0.0525007	6	0.342521
7	0.213814	7	0.074651	7	0.494313
8	0.405193	8	0.177797	8	0.79881
9	0.153168	9	0.0540862	9	0.340009
10	0.202404	10	0.0808679	10	0.565412
11	0.21004	11	0.0819152	11	0.69634
12	1.03158	12	0.130702	12	0.859649

PseudoVertex, bb vs. cdusg		RecoVertex, bb vs. b		RecoVertex, bb vs. cdusg	
Bin	Bias	Bin	Bias	Bin	Bias
0	3.43547	0	1.09397	0	7.20115
1	3.93322	1	1.09548	1	9.23489
2	3.25316	2	1.17737	2	9.50078
3	0.844587	3	1.20229	3	4.60525
4	1.45975	4	1.22918	4	5.1248
5	3.11298	5	1.24992	5	4.78792
6	0.659249	6	1.33354	6	4.01089
7	1.01644	7	1.35355	7	3.92053
8	1.9881	8	1.35089	8	3.48562
9	0.512227	9	1.40886	9	3.6915
10	0.898033	10	1.41911	10	3.89914
11	0.693165	11	1.34141	11	3.4308
12	0.85773	12	1.00548	12	3.56707

Table A.1: The biases applied to the different trainings as described in Section 7.2.5

A.4 $g \rightarrow b\bar{b}$ versus $H \rightarrow b\bar{b}$ additional plots

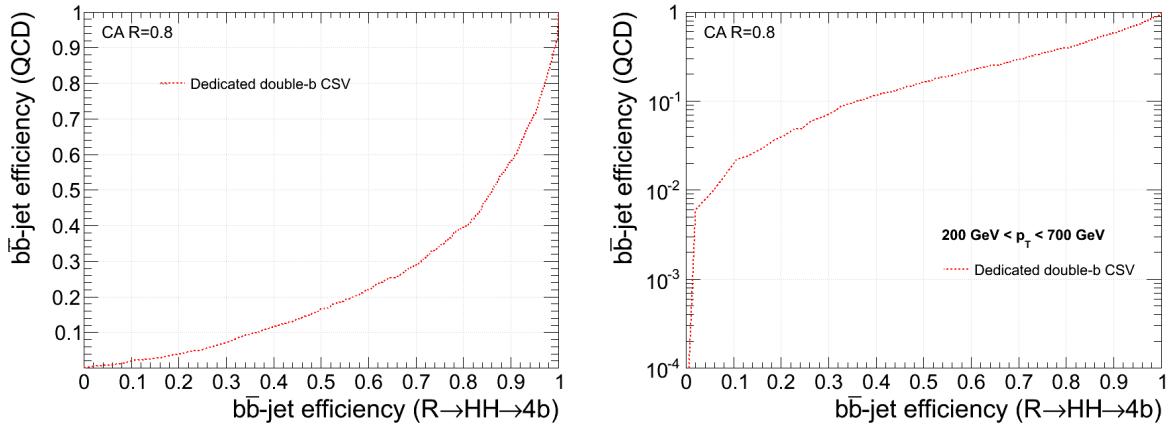


Figure A.3: Tagging efficiency for bb jets from gluon splitting versus bb jets from a Higgs decay for all jets (right) and for jets with $200 \text{ GeV} < p_T < 700 \text{ GeV}$ (log scale).

A.5 Technical training description

Some technical training information and implementation of the different processors is explained in the following. The Ntuplizer used to generate the flat training trees can be found in RecoBTau/JetTagMVALearning/test/BBtagCSVMLPIVF/VariableExtraction/VariableExtractor_IVF_cfg.py. The VariableExtractor is run locally on AOD samples with Crab at PSI Tier 3 and the output files stored directly at the T3 Storage Element. The files are then merged and copied to your local workspace by running mergeFromStorage.sh. This script also skims the cdusg NoVertex samples to keep only 1/20 of the events as this category is very high on statistics. After having merged the files, all samples are skimmed keeping only 50 000 jets in each p_T - η training category. In practice, the bias is computed by running RecoBTau/JetTagMVALearning/test/BBtag_CSVMPIVF/biasForXml.cpp. This takes as input flat trees (one for each vertex category and flavour) produced from the sample one wishes to calculate the bias from. It computes the bias and stores it in different text (.txt) files; one for each vertex category and for bb vs. b/bb vs. cdusg separately. Following is an example for the bb versus b training in the RecoVertex category:

```
< bias > 0.814042 < /bias >
< bias > 0.880306 < /bias >
< bias > 0.931248 < /bias >
< bias > 0.609174 < /bias >
< bias > 0.747541 < /bias >
< bias > 0.860736 < /bias >
< bias > 0.648791 < /bias >
< bias > 0.707409 < /bias >
< bias > 0.859501 < /bias >
< bias > 0.676458 < /bias >
< bias > 0.786535 < /bias >
< bias > 0.844379 < /bias >
< bias > 0.495463 < /bias >
```

The text files are stored in the same directory where the code is run from. This bias is then man-

ually added to the XML steering files, described further in Section ???. Storing target and $p_T - \eta$ weight is done by running RecoBTau/JetTagMVALearning/plugins/JetTagMVATreeTrainer.cc. This code takes the flat training trees and the $p_T - \eta$ histograms, computes the weight for each jet, creates a new tree and stores the weight and target. The TreeTrainer is configured through xml steering files, one for each vertex category and for bb vs. b and bb vs. dusg separately. The XML files defines the signal and background flavour and passes two input trees; signal and background. The final output is 6 different root files in total: RecoVertex_BB_B/CDUSG, PseudoVertex_BB_B/CDUSG and NoVertex_BB_B/CDUSG containing the necessary information to perform the training. The signal and background trees for the separate flavour trainings (bb vs. b and bb vs. dusg) are merged into a single tree with the jet target (signal (1) or background (0)) stored in a separate branch. The $p_T - \eta$ weight for each jet is also computed and stored. Then the trees are passed to the MVATrainer to perform the training. The MVATrainer is fully steered through an XML configuration file. The first step in the XML is the declaration of the training variables:

```
< inputid = "input" >
< varname = "jetPt" multiple = "false" optional = "false" / >
< /input >
```

Here the name of the variable is declared and two options are set: “multiple” means if the variable is allowed to appear multiple times and “optional” means if the variable can be omitted. The next step is the assignment to $p_T - \eta$ bins:

```
< processorid = "proc1" name = "ProcCategory" >
< input >
< varsource = "input" name = "jetPt" / >
< varsource = "input" name = "jetEta" / >
< /input >
< config >
< group >< box >< rangemax = "40" / >< rangemin = " - 1.2" max = "1.2" / >< /box >< /group >
< group >< box >< rangemax = "40" / >< rangemin = " - 2.1" max = "2.1" / >< /box >< /group >
< group >< box >< rangemax = "40" / >< range / >< /box >< /group >
< /config >
< output >
< varname = "var1" / >
< /output >
< /processor >
```

Where the first range sets the p_T maximum/minimum and the second sets the η range for a given bin. Next the variables are all normalized to a value between 0 and 1 using **ProcNormalize**:

```
< processorid = "proc3" name = "ProcNormalize" >
< input >
< varsource = "input" name = "trackSip3dSig" / >
< config >
< pdfsmooth = "3" size = "500" lower = " - 80" upper = "150" / >< ! --trackSip3dSig -- >
```

One can provide the range or let it be automatically calculated. The final step of the preprocessing is the **ProcSplitter** processor. The **ProcSplitter** takes variables that are stored as vectors and therefore occur multiple times, like track-by-track related variables, and split them into separate entities:

```
< processorid = "proc4" name = "ProcSplitter" >
< input >
< varsource = "proc3" name = "var3" / >
< /input >
< config >
< selectfirst = "3" / >
< /config >
< output >
< varname = "var1" / >
< varname = "var2" / >
< varname = "var3" / >
< varname = "var4" / >
< /output >
< /processor >
```

Here variable 3 is split into four entities. Then the Multilayer perceptron is defined and the training variables passed:

```
< processorid = "proc16" name = "ProcMLP" >
< input >
< varsource = "proc11" name = "var1" / >
< varsource = "proc11" name = "var2" / >
< varsource = "proc11" name = "var4" / >
< /input >
< config >
< configsteps = "100" boost = "-1" limiter = "0" > 44 < /config >
< /config >
< output >
< varname = "var1" / >
< /output >
< /processor >
```

Here “config steps” defines how many training iterations should be run, “limiter” can be used to reduce the amount of data points used for training (here set to 0. No reduction). “44” is the number of neurons to use, here for one layer. If one wishes to use more layers, they can be defined as “4:3:4”. That gives a network with three hidden layers consisting of 4, 3, and 4 neurons respectively.

A.6 List of samples

Sample	#events
RadionToHH_4b_M-1000_TuneZ2star_8TeV-Madgraph-pythia6/Summer12_DR53X_PU_S10_START53_V19-v1/AODSIM	~100k
RadionToHH_4b_M-1100_TuneZ2star_8TeV-Madgraph-pythia6/Summer12_DR53X_PU_S10_START53_V19-v1/AODSIM	~100k
GravitonToHH_4b_M-1000_TuneZ2star_8TeV-Madgraph-pythia6/Summer12_DR53X_PU_S10_START53_V7C-v1/AODSIM	~20k
GravitonToHH_4b_M-1500_TuneZ2star_8TeV-Madgraph-pythia6/Summer12_DR53X_PU_S10_START53_V7C-v1/AODSIM	~20k
GravitonToHH_4b_M-2000_TuneZ2star_8TeV-Madgraph-pythia6/Summer12_DR53X_PU_S10_START53_V7C-v1/AODSIM	~20k
QCD_Pt-15to30_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v1/AODSIM	~10M
QCD_Pt-30to50_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v2/AODSIM	~6M
QCD_Pt-50to80_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v2/AODSIM	~6M
QCD_Pt-80to120_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v3/AODSIM	~6M
QCD_Pt-120to170_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v3/AODSIM	~6M
QCD_Pt-170to300_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v2/AODSIM	~6M
QCD_Pt-300to470_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v1/AODSIM	~3M
QCD_Pt-470to600_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v2/AODSIM	~4M
QCD_Pt-600to800_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v2/AODSIM	~4M
QCD_Pt-800to1000_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v1/AODSIM	~4M
QCD_Pt-1000to14000_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v1/AODSIM	~2M
QCD_Pt-1400to1800_TuneZ2star_8TeV_pythia6/Summer12_DR53X_PU_S10_START53_V7A-v1/AODSIM	~2M

Table A.2: Samples used in the training and evaluation of the double b-tagging algorithm

Bibliography

- [1] Alexandra Oliveira, “Gravity particles from Warped Extra Dimensions, a review.”, arXiv:1404.0102v1 [hep-ph] 1 Apr 2014.
- [2] T. Plehn and M. Spannowsky, “Top Tagging”, J. Phys. G 39 (2012) 083001, doi:10.1088/0954-3899/39/8/083001, arXiv:1112.4441.
- [3] A. Altheimer et al., “Jet Substructure at the Tevatron and LHC: New results, new tools, new benchmarks”, J. Phys. G 39 (2012) 063001, doi:10.1088/0954-3899/39/6/063001, arXiv:1201.0008.
- [4] J. Beringer et al. (Particle Data Group), PR D86, 010001 (2012) and 2013 partial update for the 2014 edition (URL: <http://pdg.lbl.gov>).
- [5] N. Cabibbo,”Unitary Symmetry and Leptonic Decays”, Phys. Rev. Lett. 10, 531 (1963).
- [6] M. Kobayashi and T. Maskawa,”CP Violation in the Renormalizable Theory of Weak Interaction”, Prog. Theor. Phys. 49, 652 (1973).
- [7] S. W. Herb et al.,”Observation of a Dimuon Resonance at 9.5 GeV in 400-GeV Proton-Nucleus Collisions,” Phys. Rev. Lett. 39, 252 (1977).
- [8] Wilson, Kenneth G., “Confinement of quarks”, Phys. Rev. D Volume 10 Issue 8 (1974), doi:10.1103/PhysRevD.10.2445, <http://link.aps.org/doi/10.1103/PhysRevD.10.2445>.
- [9] M. Kreps et al.,”Production and decay of b-flavoured hadrons”, PDG Reviews 2012, <https://pdg.web.cern.ch/pdg/2012/reviews/rpp2012-rev-b-meson-prod-decay.pdf>.
- [10] S.Glashow and P. H. Ginsparg, “Top-Quark Mass and Bottom-Quark Decay”, Physical Review Letters Volume 50, Number 19 (1983).
- [11] K.A. Olive et al. (Particle Data Group), Chin. Phys. C38, 090001 (2014) (URL: <http://pdg.lbl.gov>).
- [12] The CDF Collaboration, “Measurements of $b\bar{b}$ azimuthal production correlations in $p\bar{p}$ collisions at $\sqrt{s} = 1.8$ TeV”, Phys. Rev. D **71** (2005) 092001 [hep-ex/0412006].
- [13] The CMS Collaboaration, “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”, Phys. Lett. B 716 (2012) 30, doi:10.1016/j.physletb.2012.08.021, CMS-HIG-12-028; CERN-PH-EP-2012-220, arXiv:1207.7235 [hep-ex].
- [14] P. Higgs, “Broken Symmetries and the Masses of Gauge Bosons”, Phys. Rev. Lett. 13, 508 – Published 19 October 1964.
- [15] Theodor Kaluza. “On the Problem of Unity in Physics”, Sitzungsber.Preuss.Akad.Wiss.Berlin (Math.Phys.) 1921 (1921), pp. 966–972.

- [16] O. Klein. “Quantum Theory and Five-Dimensional Theory of Relativity”, Z.Phys. 37 (1926), pp. 895–906, doi:10.1007/BF01397481.
- [17] D. Dominici et al., “Search for radion decays into Higgs boson pairs in the $\gamma\gamma bb, \tau^+ tau^- bb$ and $bbbb$ final state”, CERN-CMS-NOTE-2005-007, CMS-NOTE-2005-007.
- [18] M. Hoffmann et al. “Probing Compositeness with Higgs Boson Decays at the LHC,” arXiv:1407.8000 [hep-ex].
- [19] Bruening et al., “LHC design report. Vol. I: The LHC main ring,” CERN-2004-003-V-1.
- [20] ATLAS Collaboration (Aad, G. et al.), “The ATLAS Experiment at the CERN Large Hadron Collider”, JINST 3 (2008) S08003.
- [21] ALICE Collaboration, “ALICE: Physics Performance Report, Volume I”, 2004 J. Phys. G: Nucl. Part. Phys. 30 1517 doi:10.1088/0954-3899/30/11/001.
- [22] LHCb Collaboration, LHCb Technical Proposal, CERN/LHCC 98-4.
- [23] The CMS Collaboration, “CMS physics: Technical Design Report, Volume 1”, Technical Design Report CMS 2006.
- [24] The CMS Collaboration, ”Description and performance of track and primary-vertex reconstruction with the CMS tracker”, arXiv:1405.6569, CERN-PH-EP-2014-070, CMS-TRK-11-001 (2014).
- [25] Hahn, Kristan Allan et al., “Studies of the CMS Tracker at High Trigger Rate”, CMS-NOTE-2009-015, http://inspirehep.net/record/875957/files/NOTE2009_015.pdf.
- [26] The CMS Collaboration, “The CMS electromagnetic calorimeter project: Technical Design Report”, CERN-LHCC-97-033 (1997).
- [27] The CMS Collaboration, “Energy Calibration and Resolution of the CMS Electromagnetic Calorimeter in pp Collisions at $\sqrt{s} = 7$ TeV”, JINST 8 (2013) P09009 [arXiv:1306.2016 [hep-ex]].
- [28] The CMS Collaboration, “The CMS hadron calorimeter project: Technical Design Report”, CERN-LHCC-97-031 (1997).
- [29] J. Proudfoot, “Hadron Calorimetry at the LHC”, 34th SLAC Summer Institute On Particle Physics (SSI 2006), July 17-28, 2006, <http://www.slac.stanford.edu/econf/C060717/papers/L012.PDF>.
- [30] The CMS Collaboration, “The CMS muon project: Technical Design Report”, CERN-LHCC-97-032 (1997).
- [31] Adam et al., ”Track Reconstruction in the CMS tracker”, The CMS Collaboration, CMS-NOTE-2006-041” (2006).
- [32] The CMS Collaboration, ”Vertex fitting with the Kalman filter formalism in the ORCA reconstruction program”, CMS IN 2003/008, https://cmsdoc.cern.ch/documents/03/in/in03_008.pdf.
- [33] W. Waltenberger, R. Fruehwirth, and P. Vanlaer, “Adaptive vertex fitting”, J. Phys. G: Nuclear and Particle Physics 34 (2007), No. 12, N343. doi:10.1088/0954-3899/34/12/N01.

- [34] C. Favaro, "Study of the Associated Production of a Z Boson with b Hadrons in Proton-Proton Collisions with the CMS Experiment", Ph.D Thesis, Universität Zürich (2013).
- [35] The CMS Collaboration, "Particle-flow Event Reconstruction in CMS and performance for Jets, Taus and MET", CMS Physics Analysis Summary CMS-PAS-PFT-09-001, (2010).
- [36] G. Salam, "Towards Jetography", Eur.Phys.J. C67 (2010) 637-686 , doi:10.1140/epjc/s10052-010-1314-6, arXiv:0906.1833.
- [37] T. Sjostrand et al., "PYTHIA 6.4 physics and manual", JHEP 2006, doi:10.1088/1126-6708/2006/05/026, arxiv:0603175.
- [38] J Alwall et al., "MadGraph 5 : Going Beyond", JHEP 1106 (2011), doi:10.1007/JHEP06(2011)128, arxiv:1106.0522.
- [39] CMS TWiki, "Jet Flavour Identification (MC Truth)", <https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideBTagMCTools> (20. Sep 2014).
- [40] The CMS Collaboration, "Algorithms for b Jet Identification in CMS", CMS Physics Analysis Summary CMS-PAS-BTV-09-001.
- [41] The CMS Collaboration, "Track impact parameter based b-tagging with CMS", <https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideBTagTrackProba>, CMS Analysis Note CMS AN 2005/041(2005).
- [42] The CMS Collaboration, "A Combined Secondary Vertex Based B-Tagging Algorithm in CMS", <http://cds.cern.ch/record/927399?ln=en>, CMS Analysis Note CMS-NOTE-2006-014 (2006).
- [43] The CMS Collaboration, "Operating Points for b-tagging Algorithms", <https://twiki.cern.ch/twiki/bin/viewauth/CMS/BTagPerformanceOP> (28 Aug 2013).
- [44] C. Amsler et al. (Particle Data Group), PL B667, 1 (2008) and 2009 partial update for the 2010 edition (URL: <http://pdg.lbl.gov>).
- [45] C. Amsler et al. (Particle Data Group), PL B667, 1 (2008) and 2009 partial update for the 2010 edition (URL: <http://pdg.lbl.gov>).
- [46] The CMS Collaboration, "Jet Substructure Algorithms", CMS Physics Analysis Summary CMS-PAS-JME-10-013.
- [47] The CMS Collaboration, "Performance of b tagging at sqrt(s)=8 TeV in multijet, ttbar and boosted topology events", CMS-PAS-BTV-13-001 (2013).
- [48] C. Gershenson, "Artificial Neural Networks for Beginners", CoRR cs.NE/0308031 (2003), <http://arxiv.org/abs/cs.NE/0308031>.
- [49] R. Collobert et al., "Links between Perceptrons, MLPs and SVMs", Proceedings of the 21st International Conference on Machine Learning, Banff Canada (2004), http://machinelearning.wustl.edu/mlpapers/paper_files/icml2004_CollobertB04.pdf.
- [50] C. Saout, "A software framework for Multivariate Analysis Techniques for the CMS offline software", CMS Internal Note (2009).

- [51] P. V. Mulders et Al., "Implementation and training of the Combined Secondary Vertex MVA b-tagging algorithm in CMSSW", CMS Draft Analysis Note CMS AN AN-12-441, http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=AN2012_441_v3.pdf.
- [52] The CMS Collaboration, "The CMS magnet project: Technical Design Report", CERN-LHCC-97-010 (1997).
- [53] G. Salam et al., "FastJet User Manual", Eur. Phys. J. C 72 (2012) 1896, doi:10.1140/epjc/s10052-012-1896-2, arXiv:1111.6097.