This Dissertation

entitled

SEARCH FOR LEPTON FLAVOR VIOLATING DECAYS

OF HIGGS BOSONS

WITH THE CMS EXPERIMENT

typeset with NDdiss2$_\varepsilon$ v3.2017.2 (2017/05/09) on May 30, 2019 for

Nabarun Dev

This LaTeX 2$_\varepsilon$ classfile conforms to the University of Notre Dame style guidelines as of Fall 2012. However it is still possible to generate a non-conformant document if the instructions in the class file documentation are not followed!

> Be sure to refer to the published Graduate School guidelines at http://graduateschool.nd.edu as well. Those guidelines override everything mentioned about formatting in the documentation for this NDdiss2$_\varepsilon$ class file.

This page can be disabled by specifying the "noinfo" option to the class invocation. (i.e.,\documentclass[...,noinfo]{nddiss2e})

This page is *NOT* part of the dissertation/thesis. It should be disabled before making final, formal submission, but should be included in the version submitted for format check.

NDdiss2$_\varepsilon$ documentation can be found at these locations:

http://graduateschool.nd.edu
https://ctan.org/pkg/nddiss

SEARCH FOR LEPTON FLAVOR VIOLATING DECAYS

OF HIGGS BOSONS

WITH THE CMS EXPERIMENT

A Dissertation

Submitted to the Graduate School

of the University of Notre Dame

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Physics

by

Nabarun Dev

_____

Colin Philip Jessop, Director

Graduate Program in Physics

Notre Dame, Indiana

May 2019

SEARCH FOR LEPTON FLAVOR VIOLATING DECAYS

OF HIGGS BOSONS

WITH THE CMS EXPERIMENT

Abstract

by

Nabarun Dev

DEDICATED TO

To my family

CONTENTS

FIGURES

TABLES

PREFACE

Long time ago in a galaxy far far away....(preface is optional)

ACKNOWLEDGMENTS

I would like to acknowledge the light side of the force, Master Kenobi and Grand Master Yoda.

SYMBOLS

c speed of light

m mass

e elementary charge

E energy

CHAPTER 1

OBJECT RECONSTRUCTION AND EVENT GENERATION

1.1   Introduction

This chapter is divided into two parts. In the first part, the procedure for the generation of simulated events is described. This is done in several distinct stages with the output of one stage serving as an input for the next. A suite of software packages, developed mostly by the particle and nuclear physics communities, is used to achieve this. This part concludes by detailing the simulated datasets used in the analyses described in this thesis. In the second part of this chapter, the reconstruction of physics objects is described in detail. It starts with a description of the particle-flow algorithm which is kind of a global event reconstruction scheme for the entire event. This is followed by descriptions of track , muon and electron reconstructions. Reconstruction of jets is described next followed by description of composite objects used in the analysis such as collinear mass and transverse mass. Brief desciptions of tau lepton reconstruction and b-tagging of jets is also included.

1.2   Event Simulation

A pp collision at the LHC, like any hadronic collision, is more complex than the hard interaction of two participating partons. The proton being a composite object, the colliding partons from the hard interaction are accompanied by other quarks and gluons that interact and rearrange themseleves into colorless objects. A pp collision thus consists of: the Hard Scattering which represents the part of the collision where

two partons in the initial state interact by exchanging a high transverse momentum, and the Underlying Event that represent the interaction of the everything else in the collision except the partons in hard scattering. In addition to the implementing the above, i.e. physics of a *pp* collision that produces a bunch of final state particles, the event simulation also has to include interactions of these particles with the CMS detector. Monte Carlo methods, that use generation of random numbers to simulate sampling from a given probability distribution , are used to model the above event simulations [5].

1.2.1   Monte Carlo method

Monte Carlo (MC) methods (named after a famous casino in the city state of Monaco) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results [6]. In particle physics, these methods play a key role in generation of events and are used primary for : generation of samples from specified probabilty distributions, and the calculation of integrals. Programs which implement the above method, called MC event generators, use generation of random numbers to make decisions about physics processes. These can range from selection of processes are generated in the collision, to which decay channel a particle decays in, to making decisions on how the particle interacts with detector material. Usually, each such decision is the result of a draw from a distribution which depends only on the current state the process is in, and not on previous states. The MC generator is provided as input the distributions that represent the physics of the generated particles, their production, their decay modes and their couplings. A MC generator starts by using a pseudo-random number generator that usually outputs a random number between 0 and 1 with. Although, true random number generation can only be done by physical processess, modern pseudo-random number generators are known to generate numbers with a high degree of randommness. Starting from

this distribution, the MC event generator uses one of the various methods such as the inverse-transform method, or the rejection sampling method to convert this uniform distribution into a desired probablity distribution, $p(x)$. It is then possible to generate random numbers according to this distribution to simulate physical processes.

1.2.2 CMS simulation pipeline

The MC simulation of events in CMS consists of the following sequential steps. The first step is simulation of the Hard Scattering.As mentioned earlier, this represents the primary hard interaction in a collision where two partons in the initial state interact by exchanging high transverse momentum resulting in a final state with two or more partons. The parton density function (pdf) which parametrizes the distributions of the partons inside each hadron are used to model the momenta of incoming partons. It represents the probability of finding a parton of a certain flavour at a certain longitudinal momentum fraction, when the hadron, that contains it, is probed at a certain scale. The PDF are extracted from fits to the data, mainly from ep collisions, and various PDF sets are available for each parton flavour. Commonly used pdf sets include ones provided by the CTEQ, HERA (H1 and ZEUS) and NNPDF collaborations. The LHAPDF library provides a unified C++ interface to all major PDF sets. The matrix element formulation is used to model the hard scattering process to leading order in perturbative QCD, or to higher orders depending on the generator. The next step is simulation of the parton shower. The hadronization and radiation of quarks and gluons in the initial and final states cannot be feasibly encapsulated in the matrix element computation. Parton shower describes these missing parts. The matrix element calculations are combined with the parton shower by one of the different matching schemes which ensure that there is no double counting of terms present in both the matrix element and the partion shower expansion. The matching schemes that are most often used are MLM [7], CKKW [8] and FxFx [9]. The simu-

lation of the Underlying Event comes next. Underlying event includes everything in the collision that is not associated with the primary hard scattering process. They consist mostly of soft QCD interactions, and implemented using the MC event generators and interfaced with the matrix element simulation. The hadronization of the quarks and gluons is simulated next and it conisists of recombination of individual partons into colorless hadrons. Lastly the decay of short-lived particles is simulated.

An important part of the event generation chain is the simulation of pileup. The protons circulate inside the LHC not as a continuous beam but in discrete closely packed bunches. This leads to more than one proton-proton collision per bunch crossing, i.e. pileup both in-time and out-of-time (see chapter **??**). Event generators add pile-up events to the hard scattering samples by randomly simulating soft inelastic collisions and overlapping them. The distribution of the number of pileup interactions in data is hard to predict. MC event generators usually produce events for a scenario with a higher number of pileup vertices, and with a flat disctribution of number of vertices . This is afterwards reweighted to match the observed distribution of pileup interactions in data.

Several MC generators have been developed. Some of these can produce all components of the above simulation pipeline while some calculate only the matrix element and need to be interfaced with other generators for the simulation of remaining parts. Pythia [10] and Herwig [11] can produce the entire chain while Powheg [12–17], aMC@NLO [18] and Madgraph [19] produce up to matrix element stage. Powheg and aMC@NLO can perform next-to-leading order calculations.

Finally, the Geant4 (GEometry ANd Tracking) [20] package is used to simulate the interaction of physical particles after the collision, produced by pipeline described above, with a sophisticated and complex simulation of the detector itself. This simulated detector response is used as input for the same physics reconstruction algorithms (desribed in the next section), that are used to reconstruct the data, thus enabling

a direct comparison of the two. If differences are observed in the behavior of these reconstruction algorithms for MC events in comparsion to observed data, the MC events are tuned to the behavior observed in data.

1.3 MC samples used for the analyses

The ggH and VBF Higgs boson samples are generated with POWHEG 2.0 while an extension of POWHEG 2.0 [21] is used for the WH and ZH simulated samples. For the H $\rightarrow \mu\tau_e$ analysis, only the gluon fusion (ggH) production mode has been considered. Samples are generated for a range of H masses from 200 to 900 GeV.

The Z + jets and W + jets processes are simulated using the MG5_aMC@NLO generator at leading order (LO) with the MLM jet matching and merging scheme. The same generator is also used for diboson production which is simulated at next-to-LO (NLO) with the FxFx jet matching and merging scheme. POWHEG 2.0 and 1.0 are used for top quark-antiquark ($t\bar{t}$) and single top quark production, respectively. The POWHEG and MADGRAPH generators are interfaced with PYTHIA 8 for parton showering, fragmentation, and decays.

As mentioned earlier in this chapter, additional pileup interactions are also a part of the MC generation pipeline. All simulated samples are reweighted to the pileup distribution observed in data. An event weight is applied based on the number of simulated pileup events and the instantaneous luminosity per bunch-crossing, averaged over the run period. Several other scale factors are used to reweight the events in order to get the MC simulation to match the data closely. These include scale factors based on trigger, lepton identification, lepton isolaton and b-jet tagging efficiencies.

1.4 Physics Object Reconstruction

This section begins with the description of the particle-flow algorithm followed by reconstruction of tracks and vertices, electrons, muons, jets and other physics objects.

1.4.1 Particle Flow

1.4.2 Track and primary vertex reconstruction

Tracks of charged particles, that traverse the CMS tracker (described in section **??**), are reconstructed [22] using hits from the pixel and strip detectors in the tracker. Hits are reconstructed by clustering signals above specified thesholds in the pixel and strip channels, and then estimating the cluster positions and uncertainties in a local orthogonal system plane of each sensor. During track reconstruction, a translation is made between the local coordinate system of these hits to the global coordinate system of the tracks. The software used to reconstruct tracks by CMS is called the Combinatorial Track Finder (CTF) and is adaptation of the Kalman filter [23]. Tracks are reconstructed using a iterative procedure with the basic idea being, that tracks that are easiest to find (e.g., high p_T tracks, and produced near the interaction region) are searched in the initial iterations with subsequesnt iterations looking for more difficult sets of tracks (e.g., low p_T tracks , or tracks produced far from the interaction region). Hits unambiguously assigned to the track in the previous iterations are removed for the subsequent ones, thus reducing the combinatorial complexity. Each iteration can be divided into four sequential steps.

The first step is seed generation which provides initial track candidates that define the starting trajectory parameters and associated uncertainties of potential tracks. Charged particles follow a helical paths in the quasi-uniform magnetic field of the tracking requiring a total of five parameters to determine the trajectory. These five parameters are extracted using two or three hits in the inner region of the tracker. The seeds are constructed in the inner part (and then tracks constructed outwards, and not in the opposite manner) because the high granularity of pixel detectors (in contrast to outer strip layers) ensure low fraction of channels that are hit. Also, particles like pions and electrons interact inelastically with tracker material or lose

energy due to bremsstrahlung radiation as they traverse through the tracker to its outer regions making the idea of construcing seeds in the inner region a better choice.

The second step in track generation is track finding which is closely based on the Kalman filter. It extrapolates the seed trajectories along the expected path of a charged particle, beginning with an estimate of the track parameters provided by the trajectory seeds generated in the last step. It then uses the location and uncertainty of detected hits, and estimations of effects such as Coulomb scattering, at successive detector layers, to build track candidates, updating the parameters at each layer. First, using the parameters of the track candidate, evaluated at the current layer, an analytical extrapolation is done that determines which adjacent layers of the detector the trajectory can intersect. This takes into account the current uncertainty in that trajectory just like a Kalman filter. Secondly, a search is perfomed for silicon modules in these layers that are compatible with the extrapolated trajectory. All compatible modules in each layer are then grouped into mutually exclusive groups, such that no two modules in each group overlap. The collection of all hits from from one such module group forms a group of hits. Finally, new track candidates are formed by adding exactly one of the compatible hits from each group, to each original track candidate. The modules in a given group are mutually exclusive and a contribution of more than one hit from each group is not expected. The trajectory parameters of the new candidates are then updated by combining the information from the added hits with the extrapolated trajectory of the original track candidates. Fig 1.1 illustrates the reconstruction efficiency of tracks in case of isolated muons.

The third step in track generation track fitting. In this step the collection of hits from the last step are refitted using a Kalman filter and smoother, to provide a best possible estimate of parameters for each track trajectory. The procedure described above, in conditions as challenging as the LHC, can yields several fake tracks that are not associated with any charged particle passing through the tracker. The fourth
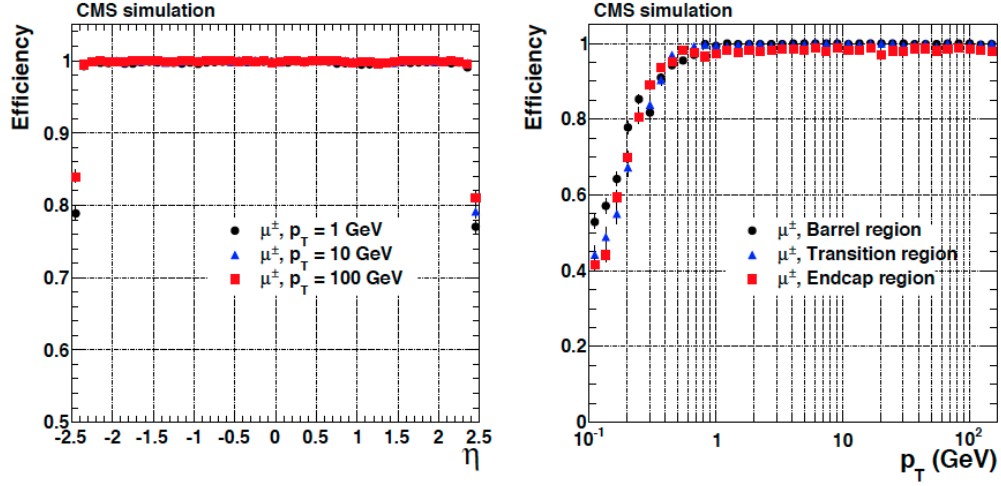
Figure 1.1: Track reconstruction efficencies for single isolated muons as a function of η and p_T [22].

and final step applies several quality requirements to set of reconstructed tracks and substantially reduces the fake contribution. The requirements are based on criteria such as the minimum number of layer the track has hits in, how compatible its origin is with a primary vertex, how good a fit they yield etc.

Proton-proton interaction vertices are reconstructed by selecting tracks that are produced promptly in the primary interaction region. The selected tracks are then clustered on the basis of their z-coordinatesat their point of closest approach to the centre of the beam spot, which represents a 3-D profile of the region where the LHC beams collide inside the CMS detector. The exact positions of the vertices are then obtained from these clustered candidates, by using a fitting procedure, called the adaptive vertex fitter [24]. The vertex which has the largest sum of squared transverse momenta of tracks originating from it is considered the primary interaction vertex.

1.4.3 Muon Reconstruction

Hits in the muon system (described in section **??**) and tracks (muons being charged particles leave tracks in the tracker) from the tracker are used to reconstruct

muons [25]. When muons traverse a muon subdetector (such as RPC, CSC or DT) in the muon system, they ionize the gas in the chambers. The electrical signals produced on the wires and strips as a consequence of the ionization are read out by electronics systems that associate these "hits" with well-defined locations in the detector. Various algorithms depending on the subdetector technology are used to reconstruct these hits. Reconstruction of muon tracks using these hits first proceeds independently of track reconstruction in the tracker. These tracks, called *standalone-muon tracks*, are built using these reconstructed hits from the muon system using a Kalman filter. Muon tracks are also built inside-out by propagating tracker tracks (described in previous section) with transverse momentum above 0.5 GeV to the muon system and matching them to (straight-line) segments of hits in DT or CSC. If a match is found, the tracker track qualifies as a *tracker muon track*. Muon tracks are also built outside-in by matching standalone-muon tracks with tracker tracks, and combining information from both using a Kalman filter fit. These are called *global muon tracks*. While the global muon reconstruction is especially efficient for muons leaving hits in several muon stations. The *tracker muon* reconstruction is more efficient for low p_T muon candidates but it can also cause fake muon tracks due to hadronic particles which *punch-through* to the innermost muon stations. The *global muon* reconstruction has high efficiency for muons penetrating through more than one muon station, and reduces the muon misidentification rate compared to tracker muons. Combining both *tracker muon tracks* and *global muon tracks*, the efficiency for reconstructing a muon is as high as 99%. The particle-flow algorithm applies a set of requirements, based on various quality parameters from muon reconstruction as well as information from other sub-detectors, to reconstructed candidates. The PF muon candidates used in the analyses described in this thesis were required to satisfy the following set of criterion to be identified as a muon:

- Must be a global muon or a tracker muon.

- Must have at least one hit in the pixel subdetector of the tracker

- χ^2 of the compatibility between the position of the standalone and trackers tracks < 12

- Transverse impact parameter of the associated tracker track with respect to the primary vertex $d_{xy} < 2mm$

- Longitudinal distance of the (origin of)associated tracker track with respect to the primary vertex $d_z < 5mm$

- constraints on muon segment matching compatibility between tracker and muon system dependent on if it is a global muon

The efficiency of the above selection for muon identification is illustrated using a plot from a study performed by the CMS Muon Physics Object group in Fig. 1.2. As can be seen from the plots, there is a difference in the efficiencies in data and MC simulation. This is corrected using a set a of scaled factors applied as a function η and p_T to adjust the efficiency in simulation to get it to match the efficiency in data.

The momentum of muons is measured by CMS using one among different possible ways involving the tracker and muon system [26] and then using the PF algorithm to refine this informmation that exploits information from the full event.

1.4.4 Electron Reconstruction

Besides muons, electron form the other primary part of the final state of the decay we are searching for in this thesis. Electrons, in the CMS, are reconsrtucted using clusters of energy formed in the ECAL (described in section **??**) and associating them with tracks from the tracker. The reconstruction of electrons is made complicated by the fact that they can radiate a significant amount of energy before reaching the ECAL. This happens due to the radiation of bremsstrahlung photons caused by the
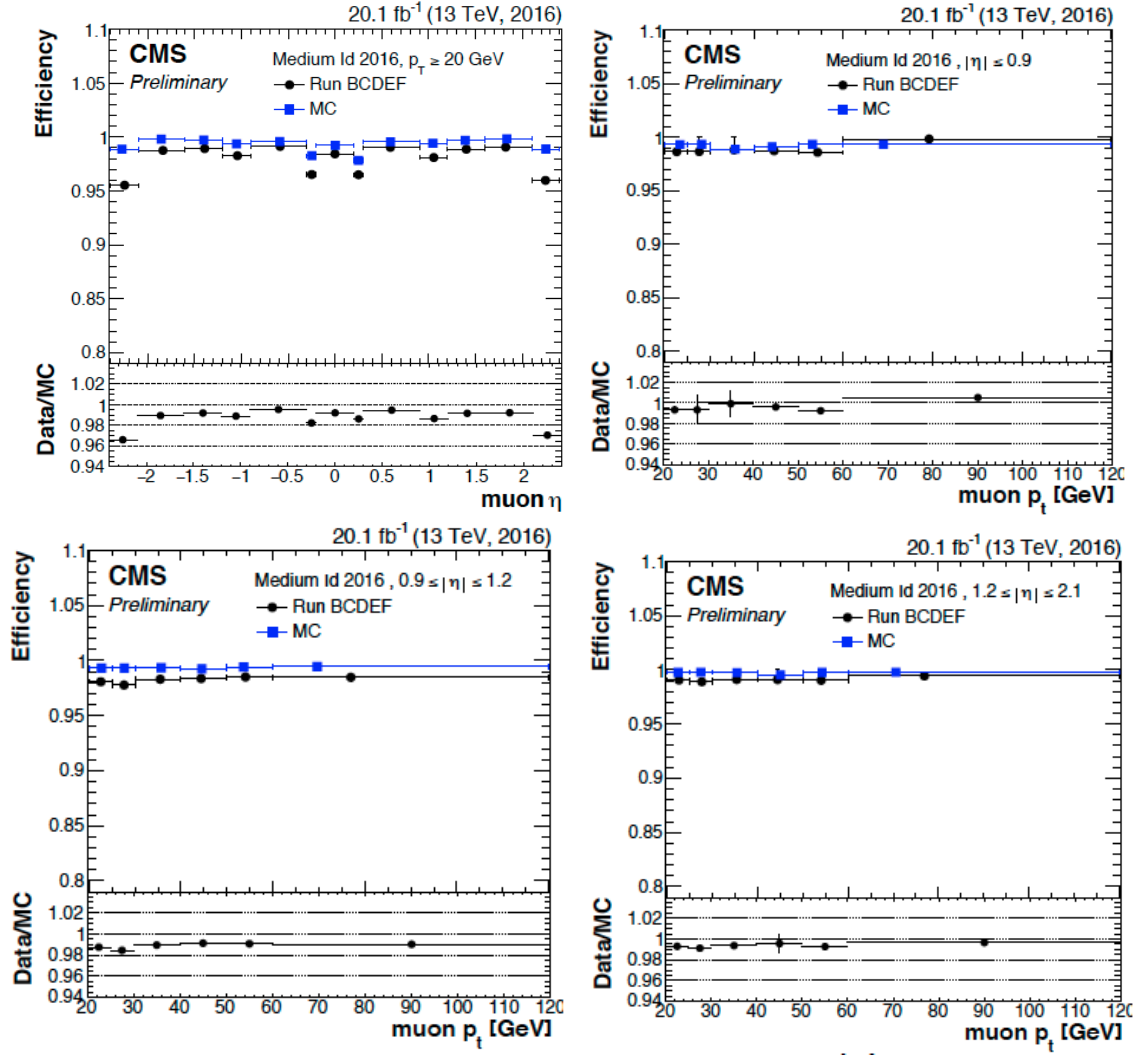
Figure 1.2: Efficiency of muon identification as a function of η and p_T, for data (black) and simulation (blue)

ineraction of electrons with atoms as they pass through the tracker. This loss can range from 33% to as high as 86% depending on η (as a consequence of the fact that the amount of detector material the electron has to cross is *eta* dependent). In order to measure the an electron's energy, clustering algorithms thus need to take into the energy from these bremsstrahlung photon showers together with the deposit made in the ECAL by the electron. The energy from these radiated photons spreads primarily in the ϕ owing to bend in electron trajectory in the magnetic field of CMS. The spread in the η direction is relatively small. These facts are used by the clustering algorithms.

The algorithm used to cluster the electron energy deposit in the ECAL barrel is called the *hybrid* algorithm. It exploits the above property of the electron shower shape, and uses the geometry of the ECAL to form clusters thare are narrow in η direction but wide in ϕ direction. Starting with a (seed) crystal containing the largest amount of energy deposited in a considered region above a certain threshold (1 GeV), it adds 5x1 arrays of crystals in $\eta \times \phi$ around the seed crystals in both directions of ϕ if the energy contained in the arrays is above another predefined threshold (0.1 GeV). Contiguous arrays are merged into clusters, and finally a electron supercluster is formed from all such strip clusters which have at least one seed strip with energy above another predefined threshold (0.35 GeV). The position of the supercluster is computed as the energy-weighted mean of the cluster positions, whereas its energy is simply taken as the sum of the energy of all its constituent clusters. In the ECAL endcap a different clustering algorithm is used owing to different geometrical arrangement of the crystals. This algorithm called the *5x5* algorithm starts similarly with a seed crystal with maximum energy in a local region, and satisfying the minimum energy requirement of 0.18 GeV. Clusters of 5x5 crystals are progressively grouped around the seed crystal, making a supercluster, if the total cluster energy exceeds 1 GeV and the are withing ± 0.7 and ± 0.3 respectively in η and ϕ around the seed crystal. The

12

positon and energy of the supercluster is calculates in the same manner as the barrel. The energy from the preshower is also added into the supercluster, using it's most energetic cluster and it's maximum distance in ϕ to other clusters and extrapolating it to the preshower plane to define the spread in the preshower. The thresholds used in the above algorithms were optimized using simulation and adjusted durting data taking periods.

The standard track reconstuction (section 1.4.2) is not efficient for electrons. This is because the standard approach is compromised by the large radiative losses in the tracker leading to a poor estimation of track parameters. Therefore a dedicated tracking procedure is used for electron candidated that used infromation not only from the tracker but also the ECAL. Just like the standard track reconstruction procedure the first step in electron track reconstruction is seeding. This is done in two ways and the results are then combined. In the first method, superclusters from ECAL are used. As mentioned earlier, owing to strong magnetic field, the bremsstrahlung photons emitted by the electrons deposit energy in the ECAL at η values similar to that of the electron, but at diffrent ϕ leading to a spread. The ECAL supercluster algorithms described above recover this energy. The position and energy of these reconstructed superclusters along with the assumption that the electrons originated close to the center of the beam spot can be used to constrain the trajectory of the electron through the tracker. Hits in the first layers of the trackers compatible with these trajectories are deemed electron seeds. In the second method of seeding, the "opposite" is done. Tracks constructed by the regular tracking algorithm are extrapolated to the ECAL and matched with a supercluster. The seeds corresponding to such matching tracks are retained as electron seeds. The seed collections from these two methods are merged leading to a increase in overall efficiency of the seeding procedure. These seeds are then used to intiate electron track finding phase and fitting phases. This track finding procedure is similar to that used in standard track-

ing except for small adjustments. The χ^2 fit thresholds used by the Kalman filter to decide whether a hit is compatible with a trajectory (see section 1.4.2 is weakened to accommodate tracks that deviate from their expected trajectory because of bremsstrahlung. Similar adjustments are made to the penalties assigned to track candidates for passing through a tracker layer without being assigned a hit. The final track fit uses a modified version of the Kalman filter, called the Gaussian Sum Filter (GSF), to account for the fact that the energy loss of an electron traversing the tracker material is non-Gaussian. This makes it unsuitable to use a conventional Kalman filter algorithm which assumes gaussian distribution. The GSF technique deals with this by approximating this non-Gaussian energy-loss distribution as the sum of several Gaussian functions, and is found to perform much better than the regular fitting procedure.

Finally, electron candidates are constructed by associating a electron track (called GSF track) produced by the above procedure with a supercluster in the ECAL. For ECAL-seeded candidates this association is made by a geometrical matching in $\eta - \phi$, while for tracker-seeded candidates a multivariate (MVA) technique that combines information from supercluster and GSF track is used. The electron charge is estimated using a combination of three procedures involving the use the GSF track curvature, use of ECAL supercluster position and its relative postion in ϕ to that of the first hit in the GSF track, and also by using KF tracks that have common hits with the GSF tracks. The combination of a best vote of three methods reduces the charge misidentification probability to 1.5% compared to 10% when using just the GSF track curvature method. Like other variables, the momentum of electrons is also estimated using a combination of tracker and ECAL measurements.

Further, several quality requirements are used on reconstructed electron candidates to identify (real/signal) electrons to supress fake sources such as photon conversions, jets misidentified as electrons etc. These requirements are based on variables

that fall into three broad categories: variables that compare measurements from ECAL and the tracker, variables that come only from ECAL (such as transverese shape of electromagnetic showers, ratio of energy fractions deposited in the HCAL to the ECAL) and purely tracking based variables (such as information from GSF track, difference between the information from GSF and KF-fitted tracks). These variables can be used in two ways: a cut-based method that uses the variables above directly to apply threshold requirements, or a multivariate (MVA) technique that uses all these variables as an input to a Boosted Decision Trees classifier to obtain a combined discriminator variable on which a threshold is applied. The BDT based method has much better performance as is illustrated in Fig **??**Two separate BDTs are trained depending on whether electron is required to pass a HLT triggering requirement or it is not. The trigger selection used in the analyses described in this thesis uses trigger based on muons. The BDT based identification criterion for non-triggering electrons is thus used in this analysis. The threshold corresponding to the working point used in this analysis has an efficiency of approximately 80%.

1.4.5 Jet Reconstruction

1.4.6 MET, MT and Collinear Mass

1.4.7 Tau Lepton and others

1.5 Datasets

The data analysed in this search was gathered by the CMS detector in 2016 during proton-proton collisions at the LHC, corresponding to an integrated luminosity of $35.9 fb^{-1}$. This data corresponds to a center-of-mass energy of 13 TeV and a spacing of 25ns between bunch crossings in the LHC with an average of about 30 collisions per bunch crossing. The subset of samples used among all collected by CMS are the ones having at least one isolated muon having transverse energy over 24 GeV, as triggered
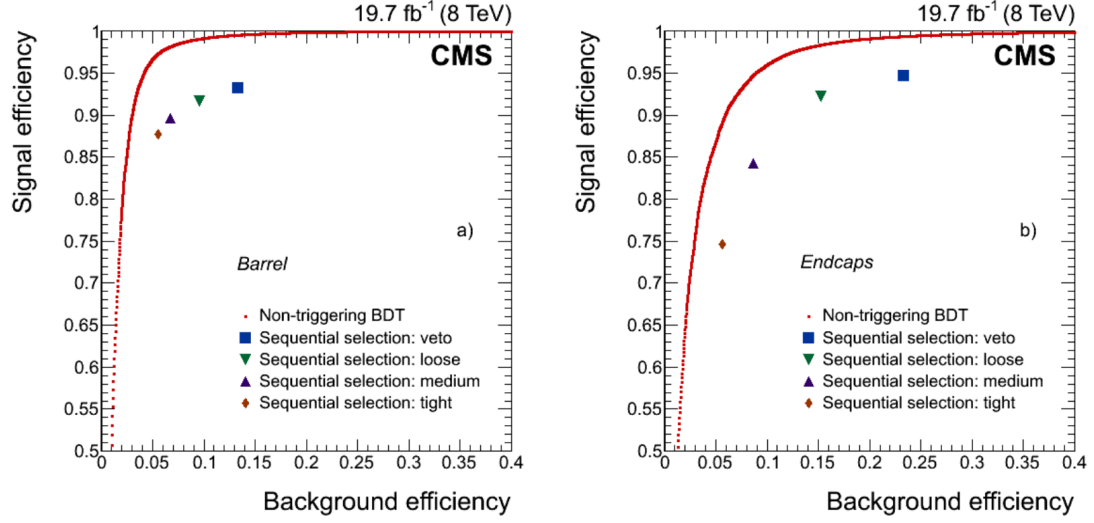
Figure 1.3: Performance of the BDT-based electron identification algorithm (red dots) compared with results from several working points of cut-based selection for electron candidates in the ECAL barrel (left), and endcaps (right).

by the CMS high level isolated muon trigger (HLT_IsoMu24 in CMS parlance).

APPENDIX A

BOOSTED DECISION TREES

A.1 Introduction

BIBLIOGRAPHY

1. L. Evans and P. Bryant. LHC machine. In *Journal of Instrumentation*, volume 3, August 2008.

2. CMS Collaboration. CMS integrated luminosity - public results. Website. `https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults`.

3. Joel Butler et al. Phase II Upgrade Scope Documentt. *CERN-LHCC-2015-019*, 2015.

4. CMS Collaboration. The CMS experiment at the CERN LHC. *JINST*, 3, August 2008.

5. Andy Buckley et al. General-purpose event generators for lhc physics. *Physics Reports*, 504:243, July 2011. doi: 10.1016/j.physrep.2011.03.005.

6. Wikipedia. Monte carlo method. Website, . `https://en.wikipedia.org/wiki/Monte_Carlo_method`.

7. F. Krauss et al. J. Alwall, S. Hche. Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions. *EPJC*, 53, 2008.

8. Leif Lonnblad. Correcting the color dipole cascade model with fixed order matrix elements. *JHEP*, 0205, 2002.

9. Rikkert Frederix and Stefano Frixione. Merging meets matching in MC@NLO. *JHEP*, 12:061, 2012. doi: 10.1007/JHEP12(2012)061.

10. Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A brief introduction to PYTHIA 8.1. *Comput. Phys. Commun.*, 178:852, 2007. doi: 10.1016/j.cpc.2008.01.036.

11. Johannes Bellm et al. Herwig 7.0/Herwig++ 3.0 release notes. *EPJC*, 76, Apr 2016. doi: 10.1140/epjc/s10052-016-4018-8.

12. Paolo Nason. A new method for combining NLO QCD with shower Monte Carlo algorithms. *JHEP*, 11:040, 2004. doi: 10.1088/1126-6708/2004/11/040.

13. Stefano Frixione, Paolo Nason, and Carlo Oleari. Matching NLO QCD computations with parton shower simulations: the POWHEG method. *JHEP*, 11:070, 2007. doi: 10.1088/1126-6708/2007/11/070.

14. Simone Alioli, Paolo Nason, Carlo Oleari, and Emanuele Re. A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX. *JHEP*, 06:043, 2010. doi: 10.1007/JHEP06(2010)043.

15. Simone Alioli, Keith Hamilton, Paolo Nason, Carlo Oleari, and Emanuele Re. Jet pair production in POWHEG. *JHEP*, 04:081, 2011. doi: 10.1007/JHEP04(2011) 081.

16. Simone Alioli, Paolo Nason, Carlo Oleari, and Emanuele Re. NLO Higgs boson production via gluon fusion matched with shower in POWHEG. *JHEP*, 04:002, 2009. doi: 10.1088/1126-6708/2009/04/002.

17. E. Bagnaschi, G. Degrassi, P. Slavich, and A. Vicini. Higgs production via gluon fusion in the POWHEG approach in the SM and in the MSSM. *JHEP*, 02:088, 2012. doi: 10.1007/JHEP02(2012)088.

18. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014. doi: 10.1007/JHEP07(2014)079.

19. Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. MadGraph 5: going beyond. *JHEP*, 06:128, 2011. doi: 10.1007/JHEP06(2011) 128.

20. S. Agostinelli et al. GEANT4 — a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250, 2003. doi: 10.1016/S0168-9002(03)01368-8.

21. Gionata Luisoni, Paolo Nason, Carlo Oleari, and Francesco Tramontano. $HW^\pm/HZ + 0$ and 1 jet at NLO with the POWHEG BOX interfaced to GoSam and their merging within MiNLO. *JHEP*, 10:083, 2013. doi: 10.1007/ JHEP10(2013)083.

22. Vardan Khachatryan et al. Description and performance of track and primary-vertex reconstruction with the CMS tracker. *JINST*, 9, 2014. doi: 10.1088/1748-0221/9/10/P10009.

23. Rudolf Frhwirth. Application of Kalman filtering to track and vertex fitting. *Nucl. Instrum. Meth. A*, 262:444, 1987. doi: 10.1016/0168-9002(87)90887-4.

24. Rudolf Frhwirth Wolfgang Waltenberger and Pascal Vanlaer. Adaptive vertex fitting. *J. Phys. G*, 34, 2007.

25. A. M. Sirunyan et al. Performance of the CMS muon detector and muon reconstruction with proton proton collisions at $\sqrt{s} = 13$ TeV. *JINST*, 13:P06015, 2018. doi: 10.1088/1748-0221/13/06/p06015.

26. Serguei Chatrchyan et al. Performance of CMS muon reconstruction in pp collision events at $\sqrt{s} = 7$ TeV. *JINST*, 7:P10002, 2012. doi: 10.1088/1748-0221/7/10/P10002.

27. Aaron Roodman. Blind Analysis in Particle Physics . 2003. doi: arXiv:physics/0312102v1.

28. CMS Collaboration. Observation of the Higgs boson decay to a pair of τ leptons. *Phys. Lett. B*, 779:283, 2018. doi: 10.1016/j.physletb.2018.02.004.

29. A. Hoecker et al. *TMVA - Toolkit for Multivariate Data Analysis*, 2017. `http://tmva.sourceforge.net/`.

30. CMS and ATLAS Collaborations. Procedure for the lhc higgs boson search combination in summer 2011. Technical report, August 2011.

31. Wikipedia. Poisson distribution. Website, . `https://en.wikipedia.org/wiki/Poisson_distribution`.

32. J. S. Conway. Incorporating nuisance parameters in likelihoods for multisource spectra. In *PHYSTAT 2011 Workshop on Statistical Issues Related to Discovery Claims in Search Experiments and Unfolding, CERN, Geneva*, January 2011.

33. A. L. Read. Linear interpolation of histograms. *Nucl. Instrum. Meth*, 425, April 1999.

34. A. L. Read. Presentation of search results: The CL_s technique. *Journal of Physics G*, 28, September 2002.

35. A. L. Read. Modified frequentist analysis of search results (The CL_s method). In *Workshop on confidence limits, CERN, Geneva, Switzerland, 17-18 Jan 2000*, August 2000.

36. Thomas Junk. Confidence level computation for combining searches with small statistics. *Nuclear Instruments and Methods A*, 434, September 1999.

37. Eilam Gross Glen Cowan, Kyle Cranmer and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *European Physics Journal C*, 71, February 2011.