Imperial College London

École Polytechnique Fédérale de Lausanne

Department of Physics

Identification of Hadronic Tau Lepton Decays with Domain Adaptation using Adversarial Machine Learning at CMS

Author:
Lucas C. Russell

Supervisor:
Dr. K. Androsov

CERN-THESIS-2022-126
28/06/2022

Submitted in partial fulfilment of the requirements for the degree of Master in Science of Imperial College London.

July 4, 2022

Abstract

This thesis reports improved machine learning-based techniques to discriminate genuine decays of tau leptons into hadrons and a neutrino against all main backgrounds at the CMS experiment. The deep convolutional neural network, DeepTau, used for tau identification by physics analyses of the 2016-2018 data-taking period at CMS, shows a sizeably different performance on collider data versus Monte Carlo simulations. This effect is particularly prominent for regions of parameter space that have high genuine tau purity. The effects of this mismodelling on discrimination against quark and gluon jets are reduced by introducing domain adaptation into the training workflow. This approach was validated by comparing the performance of the resulting network on proton-proton collision data and simulated events. The use of these adversarial machine learning techniques reduced the discrepancies from 13.3% to 0.80% in the region where the purity of hadronic taus is expected to be above 96%, while having no significant impact on tau identification performance.

Preface

This work was undertaken at the EPFL Laboratory for High Energy Physics led by Prof. Lesya Shchutska (LPHE-LS) in Lausanne, Switzerland, during my exchange year from Imperial College London. I worked within the CMS Tau Physics Object Group (POG) as a member of the CMS collaboration under the supervision of Dr. Konstantin Androsov, within the tau algorithms subgroup.

I developed the domain adaptation techniques that have been integrated into the DeepTau version 2.5 training workflow. This includes the event selection for the adversarial dataset, necessary modifications to the network architecture and backpropagation techniques, adversarial model tuning, training and evaluation.

Training of the default model that was used as a starting point for the adversarial method, along with the working point calculations for the final model, were performed by other members of the subgroup.

I presented the final DeepTau v2p5 model described in this thesis during CMS week on the 28th of June 2022, at CERN in Meyrin, Switzerland, where it was approved by the Tau POG. The model has been integrated into central CMS software, and scale factor measurements will be determined by the tau calibration-quality-modelling subgroup towards the end of 2022.

Contents

Chapter 1

Introduction

The tau lepton (τ) is the heaviest Standard Model (SM) lepton, with a mass of 1.78 GeV. This elementary particle plays a crucial role in many physics analyses at the Compact Muon Solenoid (CMS) [1], from measurements of the properties of the Higgs boson to beyond standard model (BSM) searches. The τ has a short lifetime and is typically observed through its visible decay products. The majority of tau leptons decay into a tau neutrino and hadrons, these are known as "hadronic taus" and denoted τ_h. Signatures of jets originating from quarks or gluons, which are copiously produced in proton-proton collisions, as well as electrons and muons, can be incorrectly reconstructed as genuine hadronic tau decays.

At CMS, hadronic tau candidates are first reconstructed using the hadron-plus-strips (HPS) algorithm [2–4]. The candidates are then identified as genuine τ_h decays using a deep convolutional neural network (CNN) called DeepTau. This network simultaneously discriminates candidates against all main backgrounds: electrons, muons and quark or gluon jets. Tau identification relies on a combination of high level input variables, along with information from reconstructed particles in the vicinity of the candidate. DeepTau version 2.1 (v2p1) [5] was used for all recent Run 2 analyses involving taus at CMS. This work is based on DeepTau version 2.5 (v2p5), which is currently being prepared for future analyses including data from the early Run 3 period that started in 2022.

DeepTau is trained on Monte Carlo (MC) simulated events, but these simulations do not provide a perfect description of the proton-proton (pp) collision data. This mismodelling in MC samples leads to discrepancies in the DeepTau v2p1 identification performance. Previously, this was solved by performing a set of dedicated calibration measurements, where simulations were fitted to the data to extract corrections in the form of scale factors.

The goal of this work is to reduce the differences in tau identification performance for discrimination against jets at a training level. A reduction in the effects of mismodelling could allow the DeepTau discriminator score against jets to be used as an input for analyses, and bring the scale factors closer to unity, with smaller uncertainties.

The method chosen is known as domain adaptation [6], and is implemented using adversarial machine learning techniques. Similar methods to those introduced in this work have been used at CMS previously [7]. It involves the introduction of a domain classification subnetwork, to compete with the tau type classification. The additional subnetwork discriminates between source and target domains: MC events and pp collision data respectively. This method promotes the use of features that are useful for τ_h identification but that are less sensitive to a change in domain. This approach could be more effective than dropping mismodelled variables completely, as they may still have some discriminating power. Furthermore, mismodelling could manifest as a multidimensional effect which is not visible on individual variable distributions.

This thesis is structured as follows. After a brief introduction to the Standard Model of particle physics and tau leptons, a summary of the CMS experiment and τ_{h} reconstruction is given. This is followed by an introduction to key machine learning concepts, and the algorithms used. A description of the DeepTau algorithm along with the identification performance discrepancies is then given. Next, the adversarial training methods implemented in this work are introduced, before an overview of the optimal configuration and resulting performance. Finally, conclusions and future developments are discussed.

Chapter 2

Tau Lepton in the Standard Model

2.1 Elementary Particles and Interactions

The Standard Model (SM) provides a description of elementary particles and their interactions, accounting for the three fundamental forces that dominate at the particle physics scale: the strong, weak, and electromagnetic interactions.

Elementary particles are classified into two groups: fermions ("matter" particles), and bosons (force carriers).

The elementary SM fermions, which all have spin 1/2, and their electric charge and mass[1] are given in Table 2.1. These all have quantum number conjugate antiparticles.

Lepton	Electric Charge	Mass [MeV]
e^-	-1	0.511
ν_e	0	$<1.1\times10^{-6}$
μ^-	-1	105.7
ν_μ	0	<0.19
τ^-	-1	1776.9
ν_τ	0	<18.2

(a) Leptons

Quark	Electric Charge	Mass [MeV]
u	+2/3	2.2
d	-1/3	4.7
c	+2/3	1.3×10^3
s	-1/3	93
t	+2/3	172.8×10^3
b	-1/3	4.2×10^3

(b) Quarks

Table 2.1: Physical properties of elementary fermions [8]. All values in natural units.

There are six quark flavours: up (u), down (d), charm (c), strange (s), top (t) and bottom (b), which can combine in pairs or triplets to form hadrons.

There are three charged leptons (ℓ): electron (e^-), muon (μ^-) and tau (τ^-), and three very weakly interacting neutral leptons: electron neutrino (ν_e), muon neutrino (ν_μ) and tau neutrino (ν_τ). There is a lepton number associated with each flavour of lepton (electron, muon, tau) which is conserved in all SM interactions.

The physical properties of bosons are listed in Table 2.2, those with spin 1 are gauge bosons, which are quantisations of the fundamental force fields. Gluons (g) are associated with the strong force, photons (γ) with the electromagnetic force, Z and $W^\pm$ with the weak force. The Higgs (H) is a scalar boson (with spin 0), and is a quantisation of the Higgs field. The elementary particles acquire their masses due to the interaction with the Higgs field via the Brout–Englert–Higgs mechanism [9, 10].

[1] Natural units (c=$\hbar$=1) are used throughout this work.

Boson	Electric Charge	Mass [MeV]	Spin
g	0	0	1
γ	0	0	1
Z	0	91.2×10^3	1
$W^\pm$	±1	80.4×10^3	1
H	0	125.1×10^3	0

Table 2.2: Physical properties of bosons. [8]

2.2 Tau Lepton

The tau lepton (τ) belongs to the third generation of the SM leptons. Measurements of the physics processes with taus in the final state provide stringent tests of the Standard Model. Any deviations from the SM predictions would be evidence of BSM physics.

The τ was discovered at the Stanford Linear Accelerator Center (SLAC) in 1975 through observations of $\tau^-\tau^+$ pair production in e^+e^- collisions, and subsequent decay to neutrinos, an electron and a muon [11]. The first precision measurements of the τ mass and spin were then established at SLAC and at the German Electron Synchrotron (DESY) Hamburg in 1978 [12, 13]. The lifetime of the τ was first measured from the flight length of a tau decaying into three hadrons and a neutrino at SLAC in 1982 [14].

Tau leptons play a crucial role in many physics analyses of proton-proton (pp) collisions at the LHC. These include SM precision measurements, measurements of the Higgs boson coupling to taus, its charge-parity properties and Higgs self-coupling [15–18]. Some BSM scenarios predict enhanced coupling to the third generation. Taus are instrumental in searches for additional charged and neutral scalar bosons, heavy neutral leptons and super-supersymmetric particles [19–23].

The tau is the most short lived lepton, with a mean life time of $(290.3 \pm 0.5)\times10^{-15}$s [8], it typically only travels a few millimetres before decaying. It is also the only lepton massive enough to decay into hadrons and a neutrino, with a mass of 1776.86 ± 0.12 MeV [8].

The possible tau decays and their branching fractions ($\mathcal{B}$) are shown in Table 2.3. The neutrinos produced are typically undetected due to their weak interactions with matter. Leptonic decays account for 35.2% of cases, however, in the majority (64.8%) of cases, the tau decays into hadrons and a neutrino. These are referred to as hadronic taus (τ_h). The most frequent hadronic decays produce one or three charged hadrons, typically pions ($\pi^\pm$), which can be accompanied by one or more neutral hadrons, usually neutral pions (π^0).

Decay mode	Resonance	$\mathcal{B}$ (%)	
Leptonic decays		35.2	
$\tau^- \to e^- \overline{\nu}_e \nu_\tau$			17.8
$\tau^- \to \mu^- \overline{\nu}_\mu \nu_\tau$			17.4
Hadronic decays		64.8	
$\tau^- \to \mathrm{h}^- \nu_\tau$			11.5
$\tau^- \to \mathrm{h}^- \pi^0 \nu_\tau$	$\rho(770)$		25.9
$\tau^- \to \mathrm{h}^- \pi^0 \pi^0 \nu_\tau$	$\mathrm{a}_1(1260)$		9.5
$\tau^- \to \mathrm{h}^- \mathrm{h}^+ \mathrm{h}^- \nu_\tau$	$\mathrm{a}_1(1260)$		9.8
$\tau^- \to \mathrm{h}^- \mathrm{h}^+ \mathrm{h}^- \pi^0 \nu_\tau$	$\rho'(1600)$		4.8
Other			3.3

Table 2.3: Tau lepton decays and their branching fractions ($\mathcal{B}$) [8]. $\mathrm{h}^\pm$ denotes a charged hadron. The decays for τ^+ are charge conjugates to those shown here, and their branching fraction values are identical. The most common intermediary resonance is shown, although multiple are possible.

Chapter 3

The CMS Experiment at the LHC

3.1 The LHC

The Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN) is a 26.7 km long two-ring-superconducting-hadron accelerator, situated under the Franco-Swiss border near Geneva [24]. It was designed to accelerate and collide protons (and lead nuclei) at centre of mass energies of up to $\sqrt{s}$ =14 TeV (5.5 TeV for lead nuclei).

A series of accelerators increase the energy of protons extracted from a hydrogen source to 450 GeV at injection into the LHC [25, 26]. In the LHC, two counter-rotating beams with 2808 bunches of 1.2×10^{11} protons are further accelerated using radiofrequency cavities, and directed around the rings using superconducting magnets. The beams can then be collided at different interaction points (IP), as shown on Figure 3.1.

There are four major experiments on the LHC which study these high energy collisions: A Toroidal LHC Apparatus (ATLAS), A Large Ion Collider Experiment (ALICE), LHC-beauty (LHCb) and the Compact Muon Solenoid (CMS).

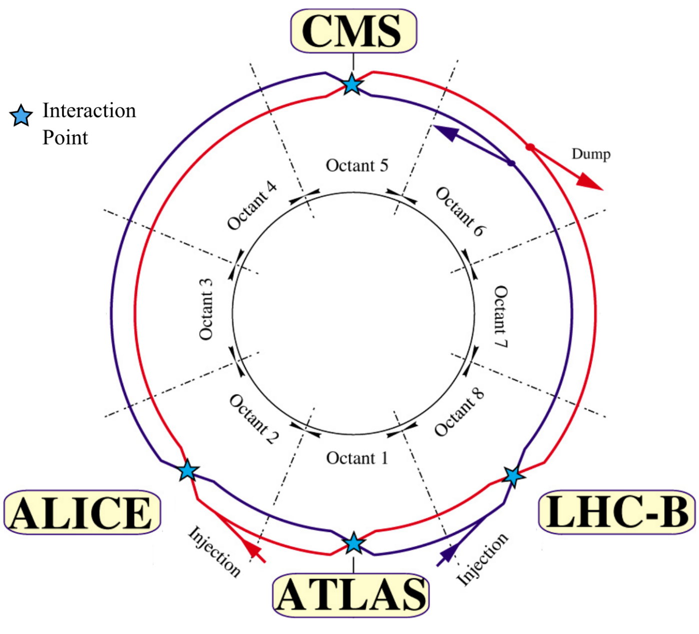


Figure 3.1: Diagram of the LHC. Beams are injected into each ring and accelerated, before colliding at an IP. Adapted from [24].

In particle physics, the probability that a certain event will take place is related to its cross section (σ_{event}), expressed in barns [b] (1b $= 10^{-24}$ cm^2). The event rate for a specific physics process is:

$$\dot{N}_{event} = L \cdot \sigma_{process} \tag{3.1}$$

where L is the instantaneous luminosity [cm^{2}s^{-1}], a characteristic of the beam at the IP. A common related unit is the integrated luminosity (L_{int}), from which the number of events that occurred in a given time period can be determined:

$$N_{event} = \sigma_{process} \int L_{int}(t)dt \tag{3.2}$$

where L_{int} is typically expressed in inverse femto-barns [fb^{-1}]. The integrated luminosity delivered by the LHC to CMS during the Run 2 data-taking period (2015-2018) was 163.56 fb^{-1} [27–29]. This value is expected to increase to a total of around 460 fb^{-1} by the end of Run 3 (2025).

Multiple pp collisions often occur simultaneously per bunch crossing, this effect is known as pileup, and makes it more difficult to study events of interest. The distributions of the mean number of (inelastic) collisions per bunch crossing in data collected between 2016 and 2018 by CMS are shown on Figure 3.2, the mean over the period was 29. The cross section of pp collisions (σ_{in}^{pp}) at the corresponding energy ($\sqrt{s} = 13$ TeV) was 69.2 mb [30].

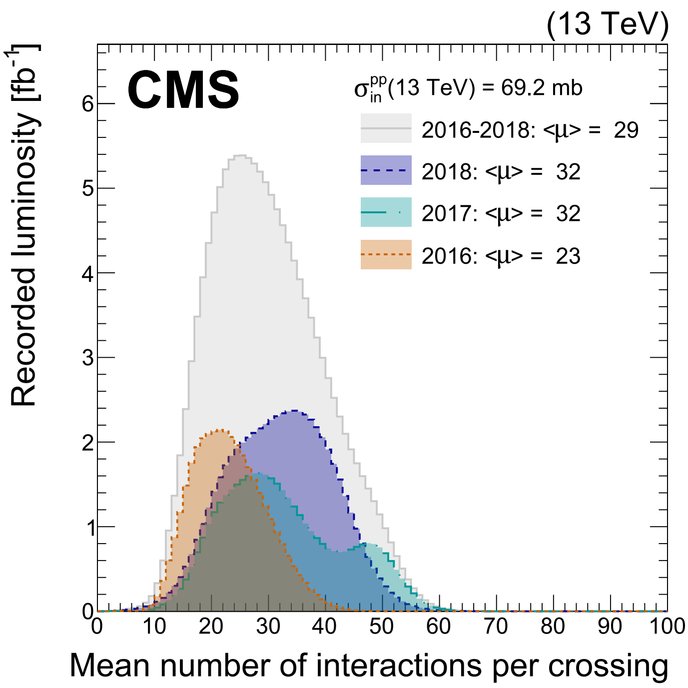


Figure 3.2: Mean number of collisions per bunch crossing for data collected by CMS between 2016 and 2018 [30].

3.2 The CMS Experiment

3.2.1 The CMS Detector

The Compact Muon Solenoid (CMS) experiment, located 100 m underground, is designed to study proton-proton (and lead-lead) collisions at centre of mass energies of up to 14 TeV (5.5 TeV for lead). The current detector supports luminosities of up to $2\times10^{34}\text{cm}^{-2}\text{s}^{-1}$ for protons. The overall layout of the CMS detector is shown in Figure 3.3, it has a cylindrical shape, with a length of 21.6 m, a diameter of 14.6 m and a weight of 12 500 t.

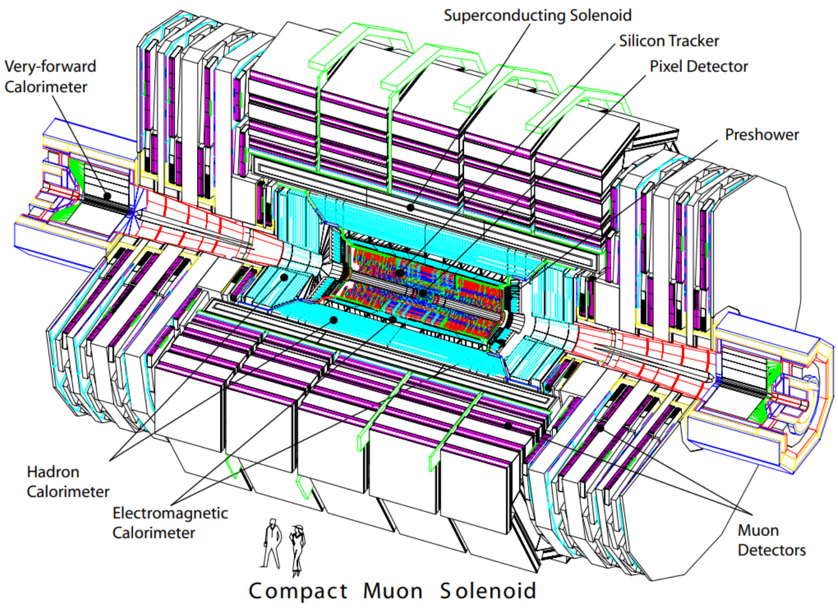


Figure 3.3: The CMS detector [1].

The coordinate system used by CMS is depicted on Figure 3.4. The x axis points radially inwards, the y axis points upwards, and the z axis along the beamline in the anticlockwise direction. The azimuthal angle ϕ is defined from the x axis in the x-y plane. The angle of a particle relative to the beamline is described by its pseudorapidity (η), which is defined as:

$$\eta = -\ln(\tan(\theta/2)) \tag{3.3}$$

where θ is the polar angle defined from the z axis.

The central feature of the detector is a 3.8 T superconducting solenoid magnet, enclosing multiple subdetectors. It is used to bend the trajectories of charged particles in order to measure their momenta and charge.

The inner tracker reconstructs the paths of charged particles (referred to as tracks), with pseudorapidity coverage $|\eta| < 2.5$. The innermost part is a silicon pixel detector, consisting of four layers in the barrel and three disk layers in the endcap regions, with a total of 124 million pixels [32]. This provides high precision three dimensional particle tracking and vertex location near to the IP, which is crucial for identifying decay signatures of particles with a distinguishable secondary decay vertex such as τ leptons and b quark hadronisation products (b hadrons). A lower granularity silicon micro-strip tracker

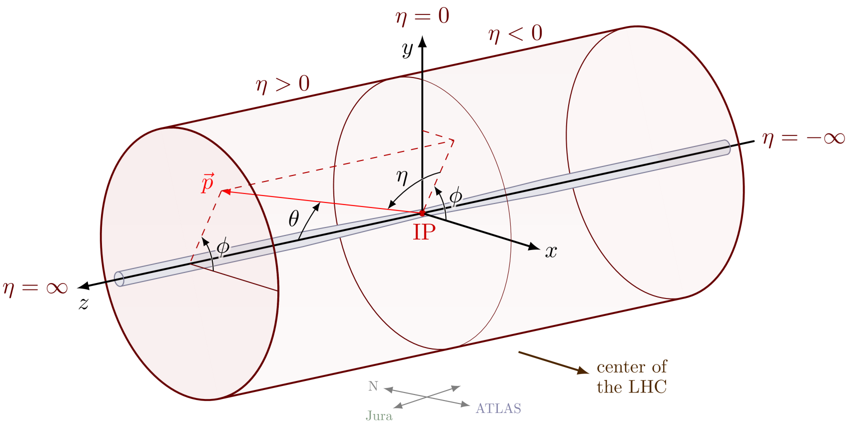


Figure 3.4: The coordinate system used at CMS [31].

extends the coverage of the inner tracker to a total diameter of 2.5 m, and consists of 10 (12) layers of micro-strip detectors in the barrel (endcap) regions.

Two calorimeters, each with a barrel and two endcap sections cover $|\eta| < 3.0$. These determine energy by measuring the cascade of secondary particles (shower) produced when the original particle enters the material.

The lead tungstate crystal electromagnetic calorimeter (ECAL) measures the energy of electrons and photons [33]. The scintillation light is absorbed by silicon avalanche photodiodes in the barrel and vacuum phototriodes in the endcaps. The ECAL absorbes 98% of the energy of electrons and photons with energies of up to 1 TeV. A lead and silicon preshower system, with finer granularity, is installed in front of the ECAL in the endcap regions for π^0 rejection against highly energetic photons.

The brass and scintillator hadronic calorimeter (HCAL) measures the energy of charged and neutral hadrons [34]. After being converted by wavelength-shifting fibres, scintillation light is detected by photodiodes.

Additionally, forward calorimetry extends the coverage provided by the inner calorimeters to $|\eta| < 5.0$.

The solenoid is surrounded by a steel flux-return yoke with four planes of gas-ionization chambers embedded within it for muon detection with $|\eta| < 2.4$. The detection planes consist of drift tubes, cathode strip chambers, and resistive-plate chambers depending on the η region.

A more detailed description of the CMS detector can be found in [1].

The LHC bunch crossing rate is approximately 40 MHz, it is not possible to store the large number of resulting events. A two-tiered trigger system is therefore implemented to select potentially interesting events for physics analyses. The first level (L1) trigger reduces the event rate to around 100 kHz using custom hardware to process signals from the calorimeters and muon detectors [35]. A second high-level trigger (HLT) then runs event reconstruction software optimised for fast processing, reducing the rate to around 1 kHZ before storage [36].

3.2.2 Event Reconstruction

Local event reconstruction within each subdetector provides information about measured particles. Examples of locally reconstructed properties include tracks and vertices in the inner tracker, characteristics of energy clusters detected in the calorimeters, and tracks/hits in the muon chambers. Individual stable particles in an event are reconstructed and identified using the particle-flow (PF) algorithm [37], by combining these properties to obtain global event characterisation. A simplified diagram illustrating the reconstruction of various particles is shown on Figure 3.5.

Typical signatures of different particle types are described below:

Photons are not detected in the tracker as they are not charged, but are absorbed in the ECAL, and their energy is determined from the resulting electromagnetic shower.

Electrons are reconstructed from their tracks and their energy deposits in the ECAL. Their momentum is determined from the momentum measurement in the tracker combined with the electron energy measurement in the ECAL and the energy sum of all photons emitted by the electron in the tracker (due to the bremsstrahlung effect).

Muons are reconstructed from tracks in the inner (silicon) tracker, and the outer muon system. The charge and momentum of the muon are obtained from the curvature of the track.

Charged hadron energy is determined from the momentum measured in the tracker and energy deposits in the ECAL and HCAL. Neutral hadron energy is determined only from the ECAL and HCAL measurements. Neutral and charged hadrons can initiate hadronic showers in the ECAL, which are then fully absorbed by the HCAL.

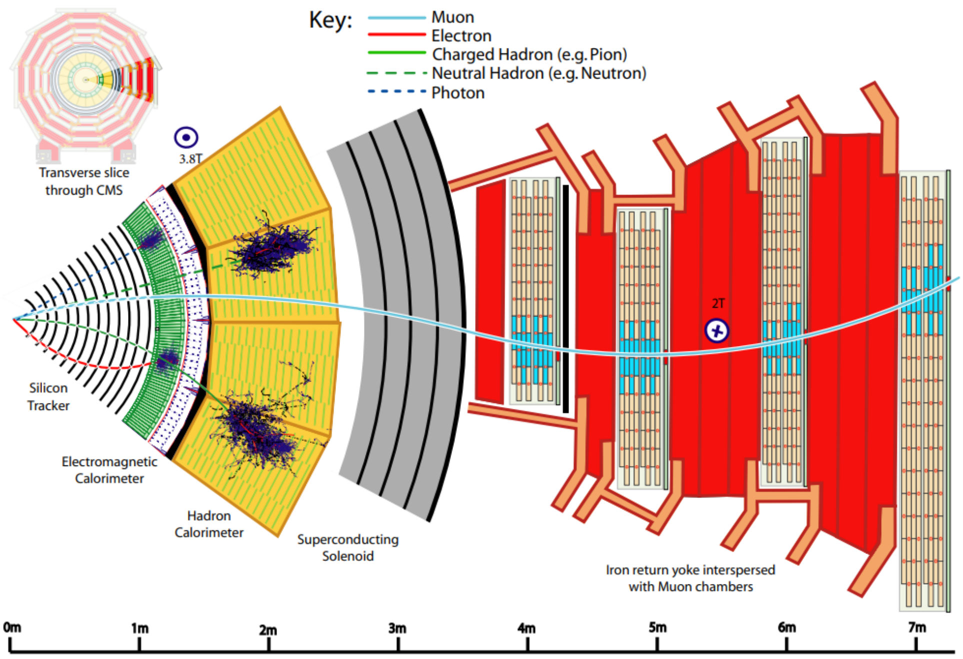


Figure 3.5: Simplified sketch of particle interactions in a transverse slice of the CMS detector. [37]

For further event characterisation, jets are clustered from PF candidates using the anti-k_T algorithm with a distance parameter of 0.4 [38, 39]. The momentum of the jet is the vectorial sum of all particle momenta within it.

The transverse momentum of a particle is the momentum perpendicular to the beam direction, it is defined as:

$$\vec{p_T} = (p_x, p_y) \tag{3.4}$$

where p_x and p_y are the x and y components of the momentum. The transverse energy is defined as:

$$E_T = E\,\mathrm{sin}\theta \tag{3.5}$$

where E is the energy of the particle.

The missing transverse momentum $\vec{p_T}^{miss}$ is the the total imbalance in transverse momentum, it is computed as the negative vector sum of the transverse momenta of the reconstructed jets, its magnitude is referred to as missing transverse energy [40]. The presence of undetected particles (such as neutrinos) can be inferred from this quantity.

The primary pp interaction vertex is defined as the reconstructed vertex with the largest summed physics object p_T^2, clustered from the tracks assigned to that vertex. Physics objects include the reconstructed jets and missing transverse momentum. To mitigate the effects of additional pp interactions contributing tracks and energy (pileup), charged hadrons not originating from the primary vertex are discarded and energy corrections are applied to the remaining PF jets.

Impact parameters d_z and d_{xy} are defined in the longitudinal and transverse planes respectively as the distance between a reconstructed particle track and the primary vertex.

Electrons and muons are additionally reconstructed and identified using dedicated standalone algorithms [41, 42], and identification efficiency working points are defined. Leptonic τ decays are reconstructed using these methods.

Isolation variables are defined for electrons and muons to reduce contamination at the analysis level from secondary electrons and muons produced in hadronic jet decays. These variables are based on photon and hadron PF candidates within a cone $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ of size 0.3 for electrons and 0.4 for muons. The relative isolation variables are defined as:

$$I_{rel}^{e/\mu} = \frac{\sum p_T^{\text{charged}} + \max\left[0, \sum E_T^{\text{neutral}} + \sum E_T^{\gamma} - p_T^{PU}\right]}{p_T^{e/mu}} \tag{3.6}$$

where $p_T^{e/\mu}$ is the transverse momentum of the electron or muon, $\sum p_T^{\text{charged}}$ the p_T of charged particles, and $\sum E_T^{\text{neutral}}$, $\sum E_T^{\gamma}$ to the transverse energy of neutral hadrons and photons within the cone. The estimated contribution from pileup (p_T^{PU}) is subtracted, this quantity is calculated differently for electrons and muons [18, 20].

3.2.3 Data and Simulated Events

Monte Carlo simulated event samples are used to train and validate the hadronic tau (τ_h) identification performance of the neural network as well as the τ_h reconstruction performance.

These are produced to correspond to the 2018 data-taking period, using a range of different generators. W bosons produced in association with jets (W+jets), Drell-Yan ($\mathrm{Z}/\gamma \to \ell\ell$) and diboson (WW, WZ, ZZ) event samples, are generated using MadGraph5 [43]. Events including top anti-top ($t\bar{t}$) quark pairs and decays of the 125 GeV Higgs boson (with various production mechanisms) to a pair of taus are generated using POWHEG [44]. Quantum chromodynamic (QCD) multijet production and BSM heavy gauge boson (Z') events with masses between 1-5 TeV are simulated using PYTHIA [45, 46]. These Z' events are generated to increase the number of high p_T taus available. Tau lepton decays, parton showering, hadronisation, and multiparton interactions are simulated using the PYTHIA programme with tune CP5 [47]. The generated events are then passed through a simulation of the CMS detector with Geant4 [48].

Additionally, the adversarial training, validation and evaluation techniques presented in this work use collider data collected by CMS in 2018.

3.2.4 Hadronic Tau Reconstruction

Hadronic tau candidates are reconstructed in one of their decay modes using the "hadron-plus-strips" (HPS) algorithm. This method is summarized here, a detailed description can be found in [2–4].

Seed regions are defined around the reconstructed anti-kT hadronic jets, all particles within a cone of radius $\Delta R = 0.5$ around the jet axis are considered. Neutral pions are reconstructed using dynamic strips in η-ϕ space containing the photon and electron constituents of the jet. Charged hadrons (referred to as "prongs") are selected from the PF candidates, the mass of the $\mathrm{h}^{\pm}$ candidates is assumed to be the charged pion mass. All possible τ_{h} candidates are then reconstructed from the hadrons and strips, in several decay modes (DM). The HPS algorithm targets the five most common hadronic tau decays, using seven reconstructed modes, including two which target decays where one of the charged hadrons has not been reconstructed:

- DM 0: $\mathrm{h}^{\pm}$ targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\nu_{\tau}$
- DM 1: $\mathrm{h}^{\pm}\pi^0$ targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\pi^0\nu_{\tau}$
- DM 2: $\mathrm{h}^{\pm}\pi^0\pi^0$ targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\pi^0\pi^0\nu_{\tau}$
- DM 10: $\mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}$ targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\nu_{\tau}$
- DM 11: $\mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\pi^0$ targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\pi^0\nu_{\tau}$
- DM 5: $\mathrm{h}^{\pm}\mathrm{h}^{\pm/\mp}$ (missing $\mathrm{h}^{\pm}$) targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\nu_{\tau}$
- DM 6: $\mathrm{h}^{\pm}\mathrm{h}^{\pm/\mp}\pi^0$ (missing $\mathrm{h}^{\pm}$) targeting $\tau^{\pm} \to \mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\pi^0\nu_{\tau}$

The CMS convention for numbering reconstructed tau decay modes is:

$$\mathrm{DM} = 5(n_{\mathrm{h}^{\pm}} - 1) + n_{\pi^0} \tag{3.7}$$

where $n_{\mathrm{h}^{\pm}}$ is the number of reconstructed ("signal") charged hadrons, and n_{π}^0 the number of reconstructed π^0s.

A confusion matrix giving the fraction of each generated DM reconstructed as each other DM is shown on Figure 3.6. These τ_{h} candidates were taken from a $Z \to \tau\tau$ sample, requiring $p_T^{\tau} > 20$ GeV [5]. A large fraction of τ_{h} are correctly reconstructed in their DM.

A diagram illustrating the four most commonly reconstructed decay modes (0, 1, 10, 11) is shown on Figure 3.7. DM 2, with one prong and two π^0s is often reconstructed as DM 1, so is not included. Neutral pions, which decay into two photons, are detected in the ECAL, while charged pions are typically absorbed in the HCAL. The tracks of the τ and the most common intermediary resonances are shown in the tracker.

The reconstructed τ_{h} candidates are subject to several constraints. The mass must be compatible with the relevant resonance for the targeted DM. This corresponds to the $\rho(770)$ resonance if reconstructed as $\mathrm{h}^{\pm}\pi^0$, to the $\mathrm{a}_1(1260)$ resonance if in the $\mathrm{h}^{\pm}\pi^0\pi^0$ or $\mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}$ modes, and to an expected mass spectrum if in the $\mathrm{h}^{\pm}\mathrm{h}^{\mp}\mathrm{h}^{\pm}\pi^0$ (several common resonances). The charge of the τ_{h} must be ± 1 unless reconstructed in a DM with a missing

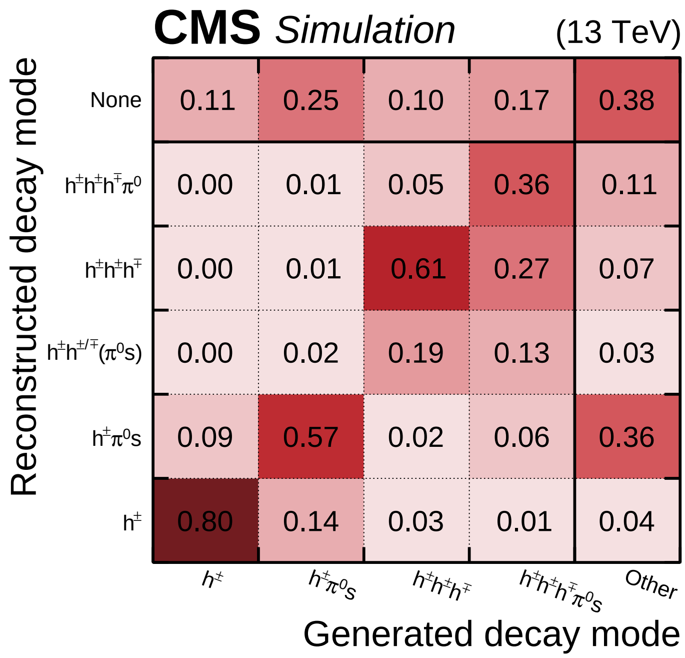


Figure 3.6: Decay mode confusion matrix [5]. For each generated DM, the fractions of τ_h reconstructed in each DM, and the fraction not reconstructed are given.

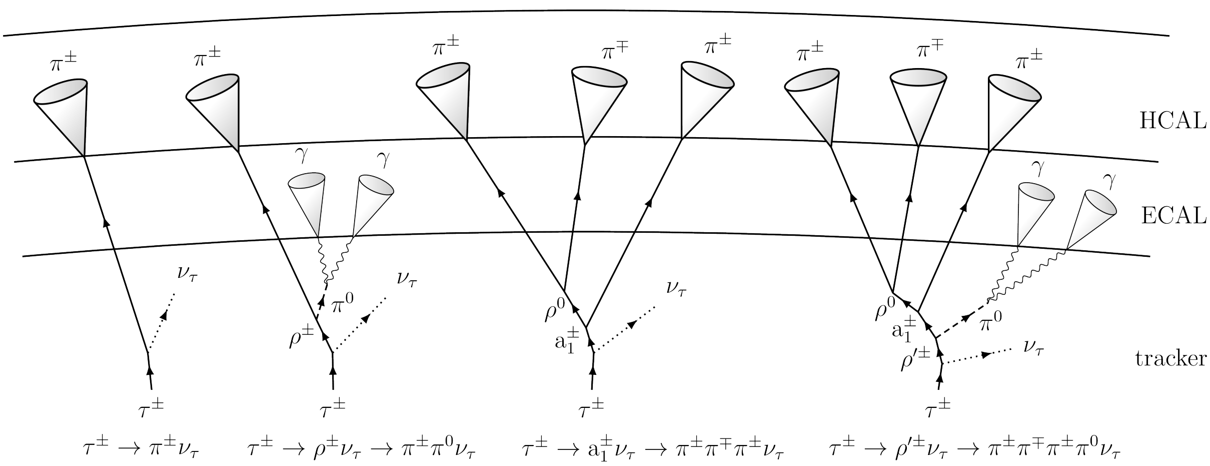


Figure 3.7: Diagram illustrating hadronic tau reconstruction at CMS [31]. $\pi^0 \to \gamma\gamma$ decays are detected in the ECAL, while $\pi^\pm$ are mostly absorbed in the HCAL, trajectories are reconstructed in the tracker. The most common intermediary resonances are shown here.

charged hadron, where the charge is set to that of the hadron with highest p_T. Finally, in this work $p_T^\tau > 20$ GeV, at the corresponding energies, τ_h decay products should be boosted and contained within a narrow cone in η-ϕ space. Therefore all reconstructed hadrons and strips need to be within a signal cone of $\Delta R = 3/p_T[\mathrm{GeV}]$, with a maximal size of $\Delta R = 0.1$, centered on the τ_h candidate axis.

Among the τ_h candidates in each seed region that pass these requirements, the one with the highest p_T is selected. The leading charged hadron is defined as the reconstructed signal charged hadron with the highest p_T.

For decays with more than one reconstructed charged hadron, a secondary vertex is fitted to the tracks associated to the signal charged hadrons to determine the point of tau decay.

Chapter 4

Machine Learning Techniques

4.1 Fundamentals

Machine learning (ML) techniques have been used in high energy physics for many years, due to their powerful ability to learn complex features. The emergence of neural networks and deep machine learning permits further extension of the use of ML to tackle very high dimensional problems. Supervised ML classification techniques are used throughout this work. In supervised deep ML, a neural network is trained to make predictions (y_{pred}) by finding correlations between input variables, called features (denoted as x) and labels (denoted as y_{true}) sampled from a dataset. Input features vary depending on the task, and can range from very low-level variables to properties of high-level reconstructed objects. Labels contain "truth" values, for example the type of generated particle. A dataset maps input features to labels. Relative importance of samples can be specified by assigning weights, this is useful for correcting imbalanced datasets. For classifiers, predictions made by the model are usually estimated probabilities that the given example corresponds to each class.

The available data is typically split into training, validation and testing sets. This is done to evaluate the ability of the model to make accurate predictions on unseen samples. A model is trained on the training set, and evaluated on the validation set. A model with good performance on the validation set is selected, and the results are confirmed on the testing dataset.

4.1.1 Neural Network Training, Validation and Evaluation

A complete pass through both training and validation datasets is referred to as an epoch.

The output returned by a model for a given sample depends on a set of trainable parameters (TPs). Inputs features are processed by layers consisting of many nodes which output values depending on their TPs (called biases and weights). The operations applied by nodes are differentiable, involving matrix multiplication and non-linear transformations. A neural network represents an ensemble of interconnected nodes. The outputs of a model, being a combination of operations applied by the various nodes are therefore differentiable with respect to the TPs of any layer.

During training, predictions are made for a batch of samples from the training dataset, this is known as the forward pass.

The loss is then calculated, which is a measure of the mismatch between the predicted labels (y_{pred}) and their true values (y_{true}). Additional performance metrics are also computed, for example accuracy, which is the weighted ratio of correct predictions over total predictions.

The model TPs are initially random, those that result in minimal loss (and therefore the most accurate predictions) are determined using gradients. These gradients are the derivatives of the loss function with respect to the TPs for each node, and are calculated using a process known as backpropagation. This involves computation of gradients starting with the final layer, and moving "backwards" until the input layers are reached. This is computationally efficient as components of the gradients from a layer are reused in the computation of those for the previous layer.

An optimisation algorithm is then used to adjust the value of each TP to minimise loss, this generally means adjusting in the direction the relevant (negative) gradient, referred to as a gradient descent "step". The optimisation task can be complicated by the presence of local minima. To overcome this, and reach optimal values more quickly, gradient descent with momentum (change in gradient between steps) is used. Along with the values of the gradients, the magnitude of the step is determined by hyperparameters, the most influential of these is the learning rate. A large learning rate leads to faster convergence, but can also result in sub-optimal and unstable performance, while a low learning rate requires more time to converge and can be more vulnerable to local minima.

During validation the TPs are not modified. Predictions, loss and metrics are computed for batches from the validation dataset. This is done to evaluate the performance of the model and to make sure that it has not been "overtrained", where the network learns specific characteristics of the testing dataset, and therefore has poor performance on unseen data. The values of the loss and metrics should be comparable between training and validation. Typically, the model with the best validation performance is selected for further testing.

To confirm the final performance of the model, predictions are computed for samples from a sufficiently large testing dataset.

Receiver operating characteristic (ROC) curves can be used to evaluate performance at different thresholds. These are typically used for binary classification, where there are only two classes, or for multi-class classification with one signal and several background classes.

ROC curves plot the false positive rate (FPR) against the true positive rate (TPR) as shown on Figure 4.1. The TPR, also known as efficiency, is the probability that the model correctly predicts the signal class. The FPR, or misidentification rate, is the probability that the model incorrectly predicts the signal class. Performance at a certain efficiency point is improved when the FPR for a given TPR is reduced.

Multi-class classifiers can also be evaluated using confusion matrices, such as the one shown in the previous section Figure 3.6. These show the probabilities for each input class y_{true} to be predicted as belonging to every other class.

In the case of hadronic tau identification, the network used (DeepTau), is a multi-class classifier, the signal class is defined as true hadronic taus, and the other classes are backgrounds. Different ROC curves are computed to evaluate performance, showing the misidentification rate for each background class against the true tau identification efficiency.

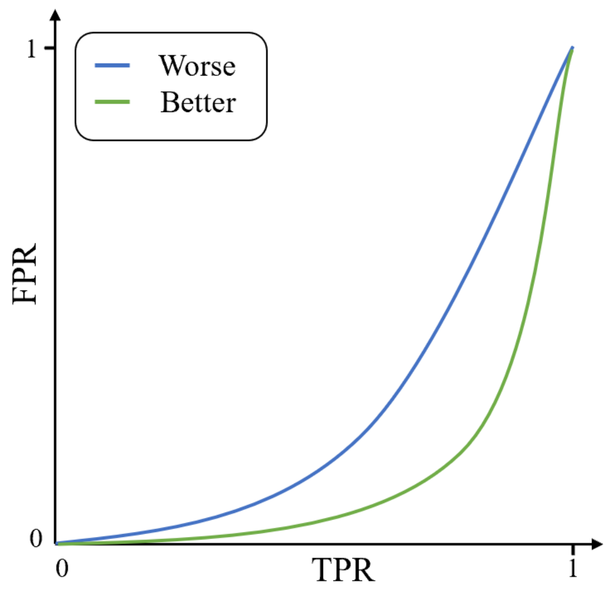


Figure 4.1: ROC curve example. Two curves with different performances are shown.

4.2 Deep Neural Networks

Deep convolutional neural networks (CNN) are a particularly effective way to tackle classification problems with grid-like inputs. These models employ dense (fully-connected) layers, along with convolutional layers.

Dense layers have connections between all nodes, the value outputted by such a layer is:

$$\sigma(\mathbf{x} \cdot \underline{\underline{W}} + \mathbf{b}) \tag{4.1}$$

where $\underline{\underline{W}}$ is the weight matrix, $\mathbf{x}$ is the tensor inputted to the layer, σ is an activation function and $\mathbf{b}$ is a bias vector that provides an arbitrary shift, both $\underline{\underline{W}}$ and $\mathbf{b}$ are trainable [49]. Each node receives inputs from all of the nodes in the previous layer.

Convolutional layers can process grids of inputs and be used to reduce dimensionality, as it is often too computationally expensive to directly process all grid cells, which can have many inputs. These layers are often used in image recognition problems, and work by successively applying a convolution operation with a "filter", this reduces the size of the grid in each dimension when no padding (artificial extension) is applied. More information on convolutional layers can be found in [50–52]. A diagram illustrating how filters are applied in convolutional layers is shown on Figure 4.2.

Regularisation can be used to prevent overtraining, in CNNs dropout regularisation is effective, where a random fraction of TPs are not used on each forward pass. This helps ensure that the model does not develop dependence on a small set of TPs [54, 55].

Activation functions

Activation functions can be used to introduce nonlinearities to the values outputted by nodes within layers. This allows the model to learn more complex correlations between inputs than a simple weighted sum.

In this work, a parametric rectified linear united activation function (PReLu) [56] is

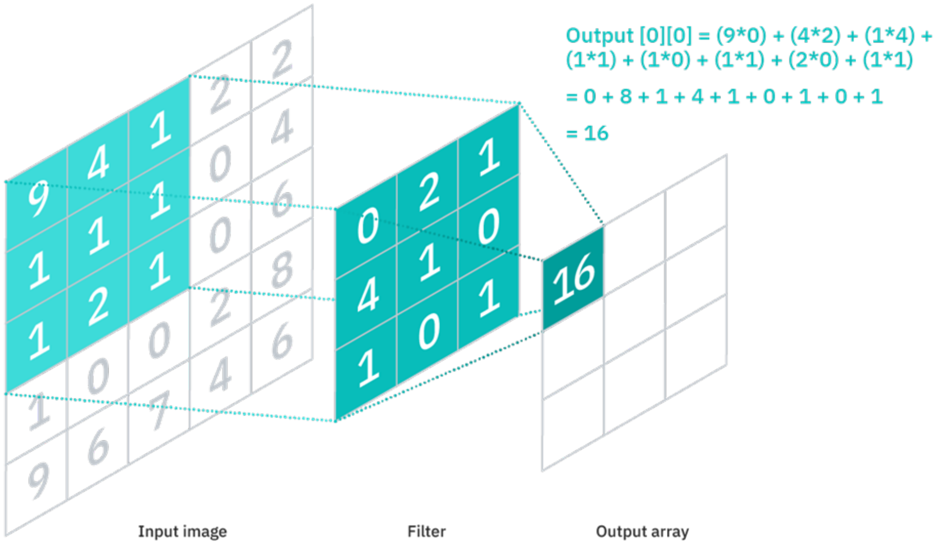


Figure 4.2: Illustration of a convolutional layer [53]. The 3×3 filter is applied successively to all areas of the grid (input image), and outputs are passed to the output array, thereby reducing the dimensionality.

used extensively, which is defined for each element x_i of the input vector $\mathbf{x}$ as:

$$\sigma_{PReLU}(x_i) = \begin{cases} x_i, & \text{if } x_i > 1 \\ \alpha_i x_i, & \text{if } x_i < 0 \end{cases} \tag{4.2}$$

where α_i is an element of $\boldsymbol{\alpha}$, a vector of TPs.

Additionally, activation functions can be used to transform an output. For example, the sigmoid function converts the weighted sum of inputs to a value between 0 and 1:

$$\sigma_{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{4.3}$$

Another example is the softmax function, useful for multi-class classification, which returns the estimated probabilities y_α that an example belongs to each class α, ensuring that the sum of all of the probabilities equals unity. The probability of each class y_α is:

$$y_\alpha = \sigma_{softmax}(x)_\alpha = \frac{e^{x_\alpha}}{\sum_j e^{x_j}} \tag{4.4}$$

where $\sum_j$ is a sum over all of the outputs for normalisation of the probabilities.

Loss functions and minimisation

Various loss functions are used in this work.

Binary cross entropy loss is used for the binary classifier that discriminates between domains (data/MC), where the label for any sample is zero or one. It is defined as:

$$H_{bin}(y^{true}, y^{pred}) = -y^{true}\log(y^{pred}) - (1-y^{true})\log(1-y^{pred}) \tag{4.5}$$

The loss function used for tau type classification includes categorical cross entropy and focal loss components. Categorical cross entropy is defined as [57]:

$$H(\mathbf{y}^{true}, \mathbf{y}^{pred}) = -\sum_\alpha \omega_\alpha y_\alpha^{true}\log(y_\alpha^{pred}) \tag{4.6}$$

where α includes all of the classes, and ω_α is used for sample normalisation.

To give more importance to areas of parameter space where it is difficult to assign a class, focal loss functions are used, defined as [58]:

$$F(\mathbf{y}^{true}, \mathbf{y}^{pred}; \gamma) = -\mathbf{y}^{true}(1 - \mathbf{y}^{pred})^{\gamma}\log(\mathbf{y}^{pred}) \tag{4.7}$$

In this work, TPs are initialised using *He* uniform variance scaling [56]. Minimisation of loss is done by adjusting the TPs with optimiser algorithms. These take gradients of the loss with respect to the TPs, obtained by backpropagation, as inputs. Two such algorithms are used, the adaptive momentum estimation (*Adam*) [59], and its closely related Nesterov-accelerated variant (*NAdam*) [60].

4.3 Domain Adaptation and Adversarial Machine Learning

In particle physics, neural networks are often trained on simulated event samples, as the generator level truth is available, and defining regions in collider data with sufficiently high purity is difficult. However, these simulations may not provide a perfect description of the data. Mismodelled variables can lead to performance discrepancies when the network trained on simulations is applied to data. Domain adaptation is a technique where a model is trained with the additional goal of performance invariance between a source domain (e.g. MC samples) and a target domain (e.g. pp collision data). This method encourages the use of features that are useful for the classification task but that are invariant to a domain change.

Domain adaptation by backpropagation, is a technique where a domain classifier with loss L_{adv} is introduced in parallel to a label classifier with loss L_{class}, after a set of feature extracting common layers. This has been used previously at CMS for displaced jet identification [7]. The method relies on sign reversal of the L_{adv} gradients during backpropagation. This leads to minimisation with reduced sensitivity to mismodelled variables, as the optimiser effectively minimises $L_{class} - \lambda L_{adv}$, where λ is a hyperparameter that determines the magnitude of the domain loss. A mathematical description of this gradient reversal technique can be found in [6].

The approach taken in this work is similar, with adversarial machine learning techniques used to implement domain adaptation, with minimal modifications as described in chapter 6. Two subnetworks are trained with competing goals, one for tau type classification and the other for domain prediction (data/MC).

Chapter 5

Hadronic Tau Identification at CMS

5.1 DeepTau: CNN-based Tau Identification Algorithm

Hadronic tau candidates that have been reconstructed by the HPS algorithm are identified using DeepTau, a deep convolutional neural network (CNN). DeepTau simultaneously discriminates hadronic tau candidates against quark and gluon jets, muons and electrons. It uses a combination of high level input variables and information from particles in the vicinity of the candidate. The network outputs an estimation of the probabilities that a candidate is an electron, muon, hadronic tau, or quark/gluon jet.

DeepTau version 2.1 (v2p1) [5] was used for all recent Run 2 analyses involving taus at CMS. This work is based on the next iteration, version 2.5 (v2p5), which is currently being prepared for early Run 3 analyses and Run 2 and early Run 3 combinations.

5.1.1 Inputs

Low-level inputs are arranged in inner and outer grids, centred on the candidate axis, defined in η-ϕ space, with 11×11 and 21×21 cells respectively. These overlapping grids are shown on Figure 5.1, the inner cells have a size of 0.02×0.02, and the outer cells 0.05×0.05. Properties of contained reconstructed particles are stored in each cell, which are divided into three blocks: electron/photon (e/γ), muon (μ) and hadron ($\mathrm{h}^{\pm}/\mathrm{h}^{0}$). Seven different types of particles can be used as inputs, the five kinds that can be reconstructed by the PF algorithm (e, γ, μ, $\mathrm{h}^{\pm}$, h^{0}), as well as PAT (physics analysis toolkit) muons and electrons that have been reconstructed from dedicated standalone algorithms [41, 42].

The low-level inputs include basic kinematic properties of each object: transverse momentum (p_T), and distance from the τ_h axis in η-ϕ space ($\Delta\eta$ and $\Delta\phi$).

PAT electron information includes the electron track quality and compatibility with the measured ECAL cluster, as well as cluster shape. PF electron and photon inputs include information about the track, distance from the primary and secondary vertices, and an estimated probability that the particle comes from another pp interaction (pileup), computed using the pileup-per-particle identification (PUPPI) algorithm [61].

PAT muon information includes track quality, the hits in each muon station (layers of the muon system), and ECAL deposits. PF muon inputs include the track quality, primary and secondary vertex association, and the PUPPI.

PF charged and neutral hadron information includes PUPPI and the fraction of energy deposited in the HCAL. Additionally, PF charged hadrons have track quality, as well as primary and secondary vertex association.

The inner grid encapsulates the signal cone, within which the τ_h decay products are found, hence the finer binning. The outer grid contains the isolation cone of radius ΔR=0.5, the cells in this grid are generally less occupied.

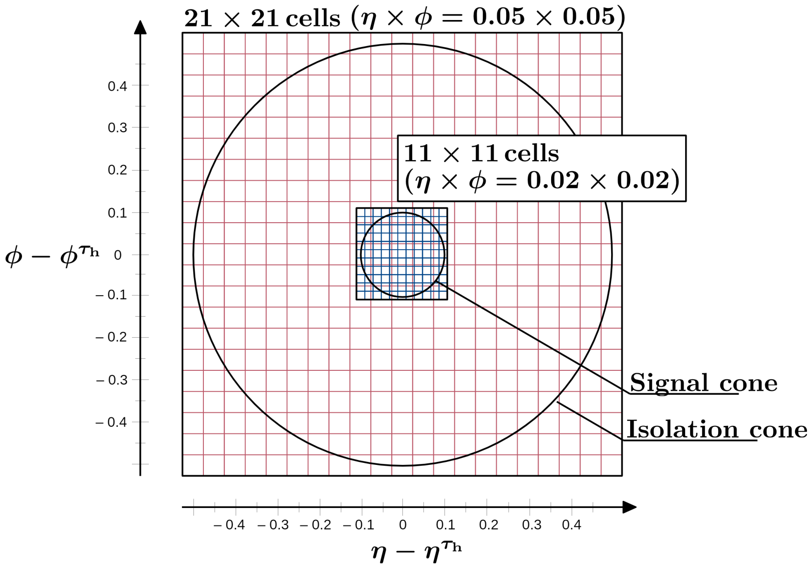


Figure 5.1: Inner and outer grid layout in η-ϕ space [5]. The inner grid encapsulates the signal cone of maximal radius 0.1 (which contains the $\pi^{\pm}$ and π^0 candidates) and consists of 11×11 cells with a size of 0.02 × 0.02 each. The outer grid contains the isolation cone of radius 0.5, and consists of 21 × 21 cells with a size of 0.05 × 0.05 each.

There are 43 high-level input variables, these include τ_h kinematic quantities such as η, ϕ, p_T, the τ_h energy E, the median energy density in the event (ρ), defined as the median value of the scalar p_T sums of all PF candidates across various detector regions (this variable is strongly correlated to the number of pp interactions in the given event). Other properties include reconstructed tau charge, number of charged and neutral hadrons associated to the τ_h by HPS decay mode reconstruction and characteristics of energy deposits from various particle types in the isolation cone. Information about the tracks is also used: impact parameters, secondary vertex information if reconstructed in a multi-prong DM and flight length. Observables related to the η and ϕ distributions of the reconstructed energy in the π^0 strips, estimated pileup density, and calorimeter variables that provide good discrimination against electrons are also included. The high-level features mostly correspond to variables that were previously found useful in boosted decision tree classifiers [2].

Integer inputs and variables with finite domain are transformed linearly to limit their values to the $[-1; 1]$ range. Other inputs are standardised using their mean values (μ) and standard deviation (σ):

$$x_{std} = \frac{x_{orig} - \mu}{\sigma} \tag{5.1}$$

where x_{std} is the standardised input, which is then clipped to [-5,5] to remove outliers, and x_{orig} is the original.

MC candidates are assigned a tau type: electron fake τ_e, muon fake τ_μ, quark/gluon jet fake τ_j and genuine hadronic tau τ_h. τ_e, τ_μ and τ_h are assigned by matching reconstructed candidates to generated leptons from the primary pp interaction within a cone $\Delta R < 0.2$

of them. Candidates matched to generated electrons and muons with $p_T > 8$ GeV, including those originating from leptonic tau decays, are assigned the τ_e or τ_μ types respectively. Those matched to generated hadronic taus with visible $p_T > 15$ GeV are assigned τ_h. If a candidate is not matched to a generated lepton, it is matched to a jet, and assigned τ_j. Only jets originating from the primary pp interaction are considered.

A balanced mix of around 10^8 candidates with different tau types from various MC events is used for training DeepTau in the default configuration. τ_e, τ_μ, τ_h and τ_j are sourced from Drell-Yan to ($Z/\gamma \rightarrow \ell\ell$), $t\bar{t}$ (semi-leptonic and hadronic decays) and W+jets events. Additional τ_j are obtained from QCD multijet samples, and additional τ_h from $\mathrm{H} \rightarrow \tau\tau$ events (with various production mechanisms) and τ gun, where only the tau decay was simulated (without the pp interaction). Additional τ_e were obtained from Z'$\rightarrow ee$ decays with $m_{Z'}$ between 1 and 5 TeV.

A loose selection is applied, with the candidate required to be correctly reconstructed by the HPS algorithm, and have a transverse momentum between $20 < p_T^\tau < 1000$ GeV. Additionally, limits are imposed on the pseudorapidity $|\eta^\tau| < 2.5$ and longitudinal impact parameter $|d_z| < 0.2$ cm (distance between the leading charged track and primary vertex).

5.1.2 Architecture and Loss Function

The DeepTau v2.5 architecture is shown on Figure 5.2, it is similar to the v2.1 architecture. There are three subnetworks, which process the high level features, inner and outer grids separately, before they are combined for the final classification.

The high level features are processed through fully connected layers, yielding 52 outputs. The inner and outer grids are processed in similar ways. The three individual blocks of a grid are first processed separately, before being concatenated and passed through more dense layers, resulting in 64 outputs per grid cell. Convolutional layers are then used to reduce the dimensions of the outputs to a flat representation (1×1 grid), leaving 64 outputs from each grid. The kernel/filter size used is 3×3, without applying padding, this means that the size of the grid is reduced by two in each dimension for every layer. The number of layers necessary to process the inner grid is therefore 5, and the outer grid 10.

These outputs are then concatenated and processed through final dense layers, before a final layer with four nodes which uses a softmax activation function to yield class probability estimates. The predicted output takes the form: $\mathbf{y}^{pred} = (y_e, y_\mu, y_\tau, y_\mathrm{jet})$.

Batch normalisation [62] and regularisation (with dropout rate of 0.2) are applied after each fully connected and convolutional layer. Nonlinearities are introduced using the PReLu activation function.

The loss function used during NN training is a sum of a cross entropy term [57] for the hadronic tau target class, a focal loss component [58] for classification against all backgrounds combined, and three individual components for classification as muons, electrons or jets when $y_\tau > 0.1$. Focal loss components are added as binary cross entropy assigns more importance to high efficiency regimes, which tend to have higher misidentification rates and are therefore not usually the regions used for physics analyses. Furthermore, in regions where τ_h ID efficiency is low, binary signal against background classification is more important than distinguishing between background types.

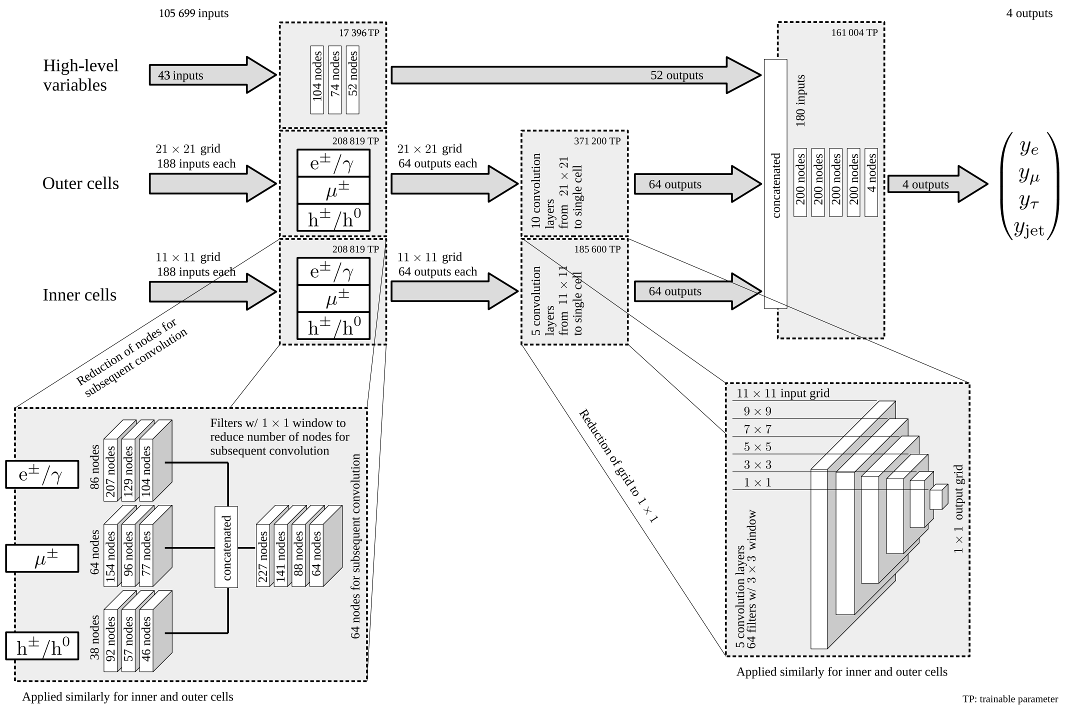


Figure 5.2: Diagram of the DeepTau v2.5 architecture - Default Configuration. Adapted from [5]. Each group of input variables (high-level features, inner cells and outer cells) are initially processed separately. The high level features and grids are first processed using dense layers. The dimensionality of each grid is then reduced to 1×1 using convolutional layers. All processed features are then concatenated and passed through dense layers, before a final softmax layer which yields probability estimates that the candidate is an electron, muon, hadronic tau or jet.

The loss is defined as:

$$\begin{aligned} L(\mathbf{y}^{true}, \mathbf{y}^{pred}) = & \underbrace{\kappa_\tau H_\tau(\mathbf{y}^{true}, \mathbf{y}^{pred}; \boldsymbol{\omega})}_{\text{Separation for all } \alpha} \\ & + \underbrace{(\kappa_e + \kappa_\mu + \kappa_j)\overline{F}_{cmb}(1 - y_\tau^{true}, 1 - y_\tau^{pred}; \gamma_{cmb})}_{\text{Focused separation of } e,\mu,jet \text{ from } \tau_\text{h}} \\ & + \underbrace{\kappa_F \sum_{i \in \{e,\mu,j\}} \kappa_i \hat{\theta}(y_\tau - 0.1)\overline{F}_i(y_i^{true}, y_i^{pred}; \gamma_i)}_{\text{Focused separation of } \tau_\text{h} \text{ from } e,\mu,jet \text{ for } y_\tau > 0.1} \end{aligned} \tag{5.2}$$

where $\alpha \in \{e, \mu, \tau, \text{jet}\}$, $\mathbf{y}^{pred}$ and $\mathbf{y}^{true}$ are the predictions and generator level truth respectively. H_τ is categorical cross entropy loss with $\boldsymbol{\omega}$ a varying parameter for sample normalisation. $\overline{F}_x$ is normalised focal loss, $\hat{\theta}$ is a smooth step function that approaches 1 for $y_\tau > 0.1$ and 0 for $y_\tau < 0.1$, this disregards discrimination between e, μ and jets when the probability of a genuine τ_h is low. The default values of the κ and γ terms are given in Table 5.1.

The setup uses the TensorFlow [63] Python library with Keras [49] as an interface.

The discriminators against jets, muons and electrons are defined as:

$$D_\alpha(\mathbf{y}) = \frac{y_\tau}{y_\tau + y_\alpha} \tag{5.3}$$

Parameter	Value	Emphasis on
κ_e	0.4	τ_e separation
κ_μ	1.0	τ_μ separation
κ_τ	2.0	τ_h separation
κ_j	0.6	τ_j separation
κ_F	5.0	High τ_h ID efficiency
γ_e	2.0	τ_e separation
γ_μ	2.0	τ_e separation
γ_j	2.0	τ_e separation
γ_{cmb}	2.0	τ_e, τ_μ,τ_j separation combined

Table 5.1: Default values of the parameters used in the classification loss function for DeepTau v2p1 and v2p5 training.

where $\alpha \in \{e, \mu, \text{jet}\}$ and y_τ is the predicted probability of a hadronic tau. D_e, D_μ and D_{jet} are referred to as DeepTau scores VSe, VSmu and VSjet respectively. Each discriminator has a defined set of working points (WP) which are used to guide physics analyses, each with target tau ID efficiencies (probability for a genuine τ_h to pass the discriminator) reported in Table 5.2.

	VVTight	VTight	Tight	Medium	Loose	VLoose	VVLoose	VVVLoose
D_e	60%	70%	80%	90%	95%	98%	99%	99.5%
D_μ			99.5%	99.8%	99.9%	99.95%		
D_{jet}	40%	50%	60%	70%	80%	90%	95%	98%

Table 5.2: Target τ_h identification efficiencies for each working point [5].

5.2 Interdomain Discrepancies in Identification Performance

DeepTau v2p1 was trained exclusively on MC simulated events, and while these generally provide a good representation of the pp collisions, some of the features used as inputs are not perfectly modelled. As a consequence, the previous setup shows increasing discrepancies between observed data and MC predictions for high D_jet scores, as can be seen on Figure 5.3. This is particularly problematic since the affected region is the most important for analyses, as it has the highest genuine τ_h purity.

Previously, this mismodelling was corrected using a set of dedicated calibration measurements, known as scale factors, for each DeepTau working point, where simulations were fitted to the data to extract corrections. However, such corrections render the use of raw DeepTau score as an input for analysis unreliable, and lead to larger uncertainties. Furthermore, the presence of these discrepancies during the training step, which is performed entirely on MC events, could be leading to sub-optimal performance.

Simply removing mismodelled variables is not possible, as preliminary studies do not point to which subset of variables is affected. Also, some of these variables could still be important for classification even if they are not perfectly modelled.

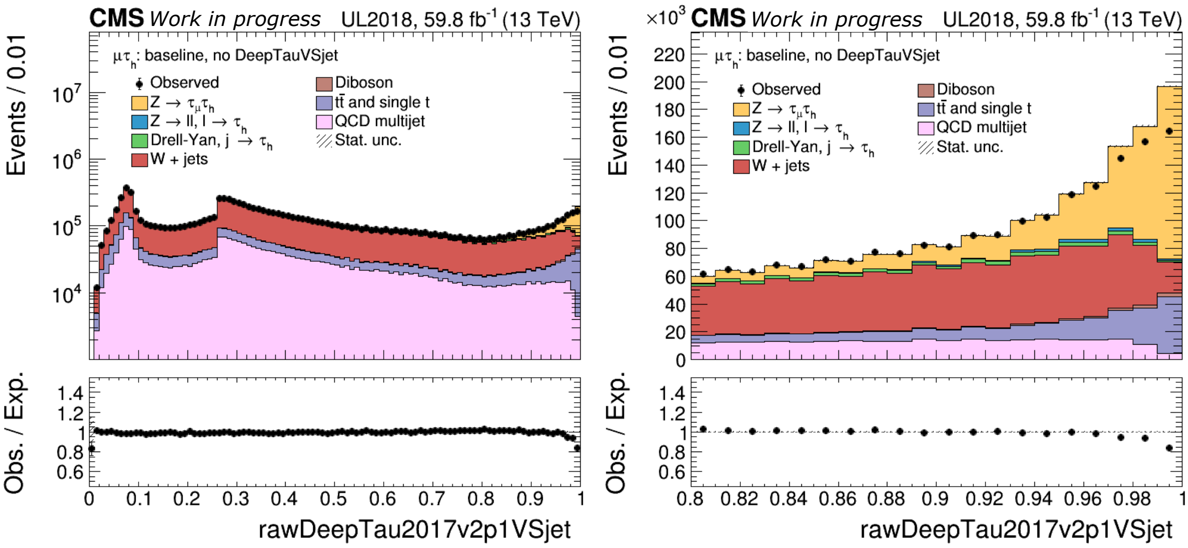


Figure 5.3: DeepTau v2.1 discriminator score distribution against jets for observed data and MC simulations. Increasing discrepancies can be observed for high D_{jet} scores as a result of mismodelling in the MC samples used for training. Courtesy of Izaak Neutelings.

Chapter 6

Adversarial Domain Adaptation for Tau Identification

The performance discrepancies between data and MC samples can be reduced at a training level by implementing domain adaptation using adversarial machine learning techniques. This involves simultaneously training two subnetworks with competing goals. In this case, one is used for τ_h classification, and the other for domain discrimination between data and MC events. The goal is to maximise the classification performance while minimising data/MC discrimination.

The advantages of this method are that the scale factors to correct residual differences could be brought closer to unity with smaller uncertainties. The use of the raw DeepTau VSjet score as an input for analyses could sizeably improve sensitivity, but is not currently recommended because of the large effects of mismodelling, if these effects are reduced it would become a viable option. Additionally, the network optimisation algorithm would have the opportunity to find trainable parameters (TP) values that provide good τ_h discrimination, but that are less sensitive to mismodelling.

The challenge is that collider data is required to train the data/MC discriminator, and a region with sufficient tau purity and control over background for this adversarial dataset therefore has to be defined. Also, this approach is not a fix of the underlying problem, which is MC mismodelling, a problem that is much harder to resolve.

6.1 Event Selection and Mixing

Training the adversarial subnetwork requires an adversarial control region dataset, with a mix of data and MC events to discriminate. Due to memory constraints and a limited number of available events, batches of 100 taus were chosen, with 50 data, and 50 MC events (from various event types).

6.1.1 Baseline Selection

The adversarial control region baseline selection is similar to that used by $\mathrm{H} \to \tau\tau$ analyses [18, 20] in the $\mu\tau_\mathrm{h}$ channel for 2018. This decay channel is chosen as there is good control over τ_h purity and background.

Selected events are required to contain a muon which satisfies several criteria. It must have a transverse momentum $p_T^\mu > 25$ GeV, pseudorapidity $|\eta^\mu| < 2.1$, relative isolation $I_{rel}^\mu < 0.15$ and impact parameters (distance between the muon track and primary vertex)

$|d_z| < 0.2$ cm, $|d_{xy}| < 0.045$ cm. It must also pass the medium muon ID as defined in [42].

Additionally, events must contain a τ_h candidate correctly reconstructed by HPS in a decay mode, with $p_T^\tau > 20$ GeV, $|\eta^\tau| < 2.3$, and impact parameter (distance between leading charged track and primary vertex) $|d_z| < 0.2$ cm. The candidate must also pass the medium DeepTau v2p1 jet discriminator ($D_{jet}^{\mathrm{v2p1}} > 0.883$).

If there is more than one muon/tau candidate fulfilling the criteria above, the most isolated candidate is selected, unless they are equal in which case the highest p_T candidate is chosen. The candidates in the selected pair are required to be separated by $\Delta R > 0.5$, and to have opposite electric charges. The transverse mass m_T of the muon plus the missing transverse energy (MET) is required to be less than 30 GeV, where $m_T = \sqrt{2p_T^\mu p_T^{miss}(1-\cos(\Delta\phi))}$ with $\Delta\phi$ the azimuthal separation between the muon and MET. This cut suppresses the W+jets background.

Finally, extra lepton vetoes are applied:

- The event must contain no additional muon with $p_T^\mu > 10$ GeV, $|\eta^\mu| < 2.4$, $I_{rel}^\mu < 0.30$, passing the medium muon ID.
- The event must contain no additional electron with $p_T^e > 10$ GeV, $|\eta^e| < 2.5$, $I_{rel}^e < 0.30$, passing the 90% electron ID working point as defined in [41].
- The event must contain no pair of muons with opposite charge separated by $\Delta R > 0.15$, both with $p_T^\mu > 15$ GeV, $|\eta^\mu| < 2.4$, $I_{rel}^\mu < 0.30$, $|d_z| < 0.2$ cm, $|d_{xy}| < 0.045$ cm, and passing the loose muon ID [42].

Events were selected using the SingleMuon trigger, and requiring the reconstructed muon to match the online muon reconstructed at HLT.

This baseline selection was applied to 2018 collider data collected by CMS, as well as Drell-Yan, $t\bar{t}$ and W+jets MC events.

6.1.2 Final Selection and Sample Weighting

A final selection was then applied to the pre-selected samples, requiring the transverse momentum of the τ_h candidate $p_T^\tau > 30$ GeV, a DeepTau v2p1 VS jet score $D_{jet}^{\mathrm{v2p1}} > 0.90$ to focus on region with highest discrepancies, that the candidate passes DeepTau v2p1 very very loose WP VS electrons ($D_e^{\mathrm{v2p1}} > 0.168$), and tight WP VS muons ($D_\mu^{\mathrm{v2p1}} > 0.875$). Finally, the reconstructed tau decay mode was required to be 0, 1, 10 or 11 (1 or 3 prong with or without a π^0).

To recreate the mix of contributions from different event types that are present in data, MC τ_h were taken from Drell-Yan and $t\bar{t}$, τ_j from W+jets and $t\bar{t}$ semi-leptonic events. Additionally, τ_j were taken from QCD multijet samples, and τ_μ from different Drell-Yan files, with a looser selection necessary due to a lack of events after the baseline selection. This looser selection only required the candidate to be correctly reconstructed by HPS, pass the final selection above, $|d_z| < 0.2$ cm and $|\eta^\tau| < 2.3$ cuts.

The proportions of different MC contributions present in the final mix were determined from those in the DeepTau v2p1 discriminator distribution shown on Figure 6.1, kindly provided by the Imperial College London CMS $\mathrm{H} \to \tau\tau$ group.

The 50 MC taus per batch were split into: 36 τ_h from Drell-Yan (deemed equivalent to $\mu \to \tau$ embedding) , 2 τ_h from $t\bar{t}$, 5 τ_j from W+jets, 5 τ_j from QCD multijet, 1 τ_j from $t\bar{t}$ and 1 τ_μ from Drell-Yan (equivalent to $Z \to \mu\mu$).

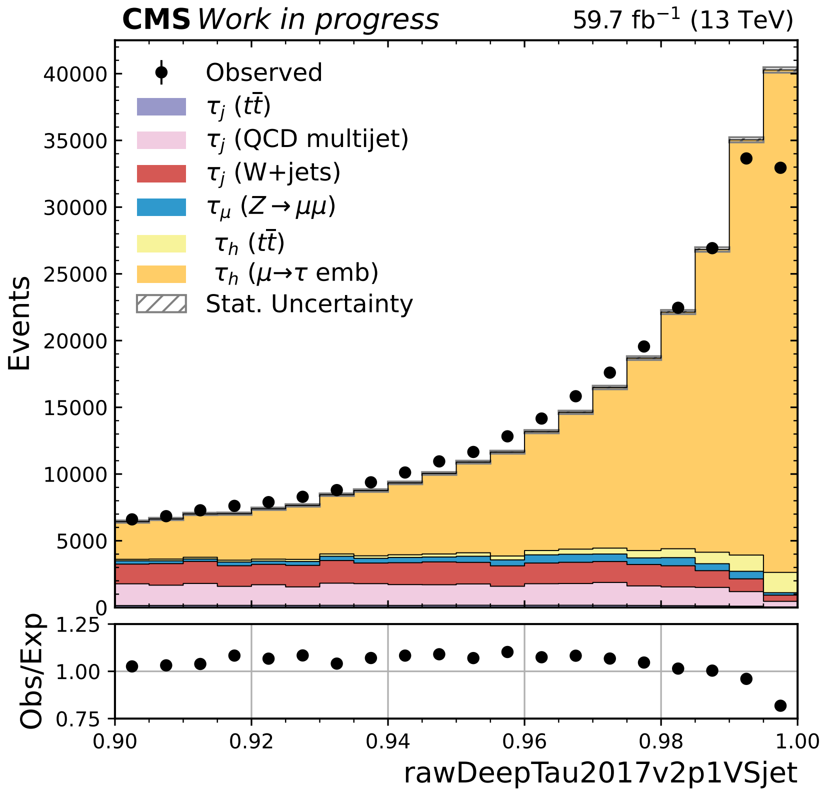


Figure 6.1: DeepTau v2p1 VSjet score distribution courtesy of the Imperial College London CMS H $\to \tau\tau$ group. The increasing data/MC discrepancies seen previously for high D_{jet} scores are present.

To avoid large values for the final p_T weights, a fraction of events in the high p_T bins from the QCD multijet, and τ_h Drell-Yan samples were dropped using a probabilistic approach. The fractions of events per bin that were expected from these contributions was determined from p_T spectra provided by the H $\to \tau\tau$ group.

Each contribution was then reweighed to the corresponding expected p_T spectrum from H $\to \tau\tau$, no η reweighting was necessary. The resulting p_T and η spectra of the adversarial control region dataset are shown on Figure 6.2.

The DeepTau v2p1 VS jet scores of this dataset are shown on Figure 6.3. The distribution is similar to the analysis configuration in Figure 6.1, with the familiar trend of increasing discrepancies in the high τ_h purity bins (signal region) present. The final adversarial dataset contained 1875 batches of 100 taus. 1125 batches were used for training and validation, and 750 for testing. Data event were labelled unity, and MC events were labelled zero.

6.2 Adversarial Network Architecture and Backpropagation

The second step in implementing adversarial training was modifying the network architecture to introduce the adversarial subnetwork. The DeepTau v2p5 architecture in adversarial configuration is shown on Figure 6.4.

A new adversarial block was introduced, which in the same way as the final classifi-

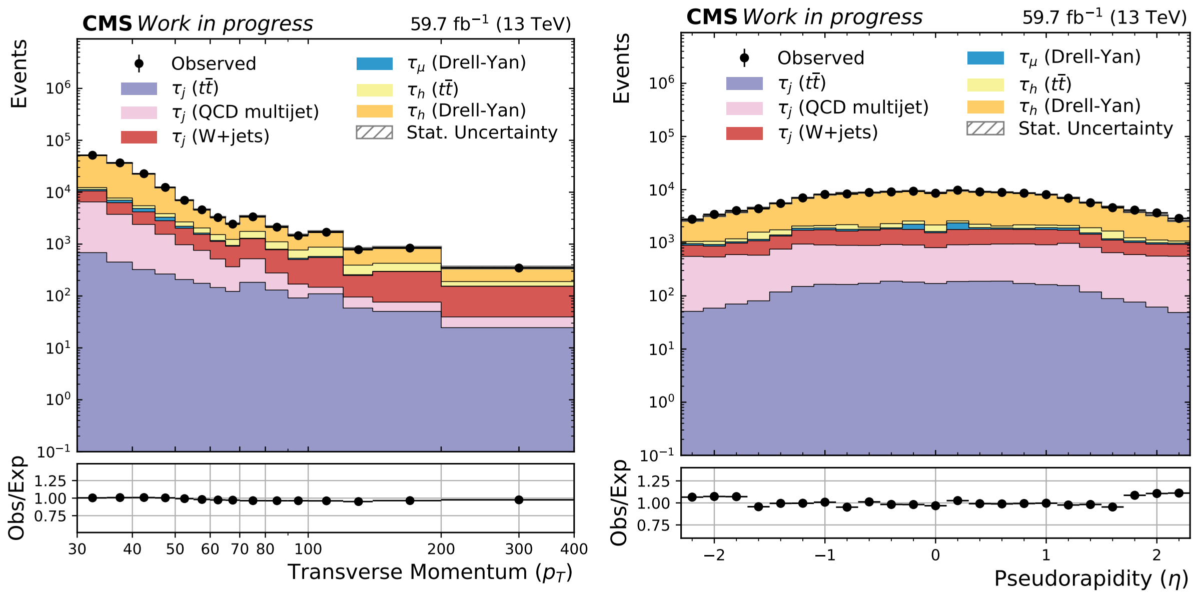


Figure 6.2: p_T and η spectra of the resulting adversarial control dataset. After p_T weighting only, the data/MC agreement for both distributions is good and no η weighting is necessary.

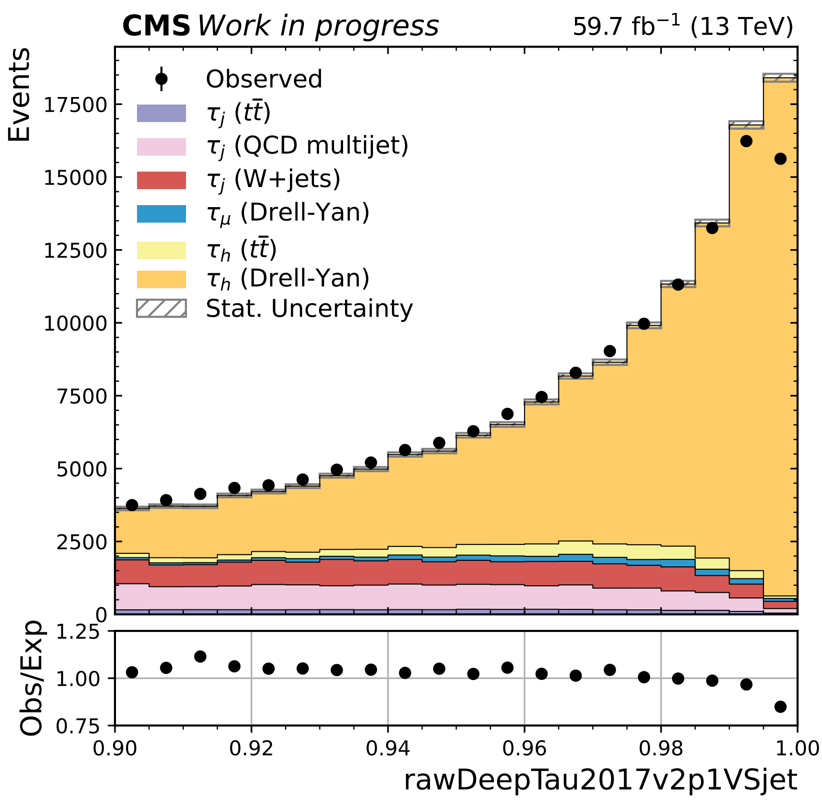


Figure 6.3: DeepTau v2p1 VSjet score distribution for the adversarial control region dataset. This is not a perfect replica of the H $\rightarrow \tau\tau$ analysis distribution but displays the familiar data/MC discrepancies in the high purity bins.

cation layers, takes the processed high and low level variables as inputs, before passing them through four dense layers. A sigmoid activation layer is then used to cast the output

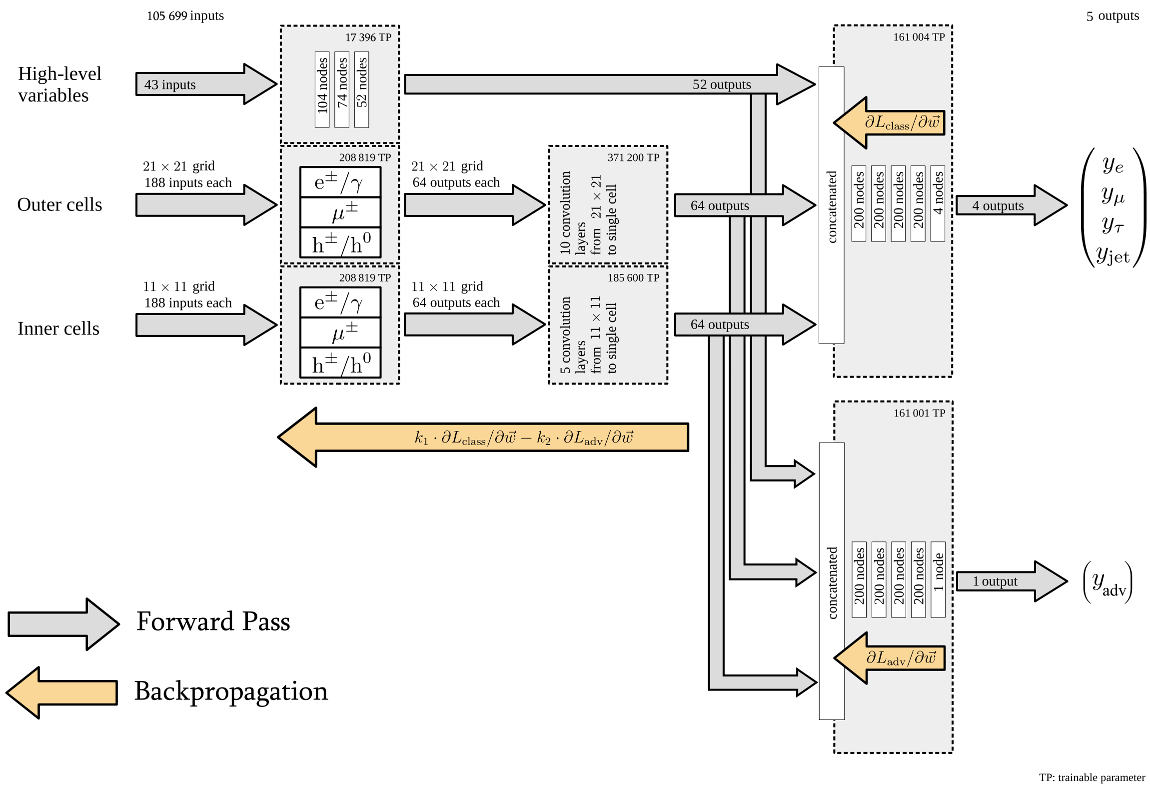


Figure 6.4: Diagram of the DeepTau v2p5 architecture - Adversarial Configuration. Adapted from [5]. A set of final adversarial layers was introduced for domain (data/MC) discrimination, this consists of several dense layers before a softmax layer which yields an output y_{adv} between zero and unity. The modified backpropagation is shown, gradients passed to the optimisation algorithm in the common layers are a linear combination of classification loss gradients and sign reversed adversarial loss gradients.

to a value between zero and unity, denoted as y_{adv}. With the chosen labelling convention $y_{adv} = 1$ indicates a candidate is from data, and $y_{adv} = 0$ indicates an event is from MC.

The adversarial loss function L_{adv}, used to compare this prediction to the labels, is defined as:

$$L_{adv} = H_{bin}(y_{adv}^{true}, y_{adv}^{pred}) \tag{6.1}$$

where H_{bin} is defined in Equation 4.5. Additionally, an adversarial binary accuracy metric was introduced, which evaluates the fraction of candidates for which the network can successfully predict the domain (data or MC). The high level, inner and outer grid processing layers (including the convolutional layers) are referred to as common layers, as inputs for both the final adversarial and classification layers pass through these.

The training workflow for DeepTau v2p5 has two steps. First, the model is trained with the default architecture shown in Figure 5.2 for 5 epochs. Each epoch corresponds to approximately 308 000 batches of 250 taus from the default training dataset, and takes between 1.5-2 days to complete. The purpose of this default training is to obtain a good tau type classification performance baseline, before applying adversarial domain adaptation methods. The second step is to continue training the model on a mixed dataset for 1 epoch, introducing the adversarial architecture. For this step, 1 epoch corresponds to 100 000 batches, where each batch contains 100 taus from the default dataset, and 100 taus from the adversarial control region dataset. The number of adversarial taus used for training is 75 000, the network cycles through these in a loop. 32 500 adversarial taus were used for validation. One epoch of adversarial training takes between 12 and 15 hours. Training was performed using NVIDIA Tesla V100, NVIDIA GeForce GTX 1080 Ti or NVIDIA Tesla T4 GPUs.

Alongside the architecture change, the network backpropagation procedure was modified to change the gradients passed to the optimisation algorithm for the common layers.

The classification loss (default), given in Equation 5.2, is denoted as L_{class} and is computed on the tau type classification output, with only the taus from the default dataset. The adversarial loss, L_{adv} is computed on the adversarial output with the taus from the adversarial dataset.

The gradients for the final classification and adversarial blocks are first calculated normally, as the gradients of L_{class} and L_{adv} respectively. These are referred to as classification and adversarial gradients. To prevent data/MC discrimination in the common layers, the signs of the adversarial gradients are then reversed and combined linearly with the classification gradients. This is expressed in the form:

$$k_1 \cdot \frac{\partial L_{class}}{\partial \vec{w}} - k_2 \cdot \frac{\partial L_{adv}}{\partial \vec{w}} \tag{6.2}$$

where k_1 and k_2 are the adversarial hyperparameters that determine the relative importance of tau type classification and avoiding data/MC discrimination. The sign reversal of the adversarial component of the gradients in the common layers means that the optimiser partly attempts to adjust the layer weights in the opposite direction to one which leads to good data/MC discrimination. There is no sign reversal in the final adversarial layers, as this provides a good measure of how much the network can actively discriminate data and MC with the information available at the output level of the common layers.

The gradients that are passed to the optimiser therefore encourage minimisation with reduced sensitivity to mismodelling. Two optimisers are used, each with a different learning rate. The principal optimiser targets the common layers and final classification layers, the adversarial optimiser the final adversarial layers.

The DeepTau algorithm is principally written in Python with the data loading done using C++ for efficiency. The current version, including all of the changes necessary for the adversarial domain adaptation implemented in this work is publicly available [64].

Chapter 7

Model Optimisation, Results and Discussion

7.1 Model Tuning and Cross Checks

Various hyperparameters and properties of the NN model can be adjusted to optimise identification performance while minimising data/MC discrepancies. These include the adversarial hyperparameters k_1 and k_2, the learning rates of the principal and adversarial optimisers, as well as the optimisation algorithms used (*NAdam*, *Adam*). For all testing described here, a binary cross entropy loss function was used for L_{adv}.

7.1.1 Evaluating Data/MC Discrimination Performance

Along with the adversarial accuracy metric, the adversarial output y_{adv} provides a method to evaluate data/MC discrimination. By plotting the predictions for data and MC events, the domain predicting power of the model can be determined. For a model which accurately discriminates data and MC, each distribution should lean towards a different value, 1 or 0 for data and MC respectively. If the network cannot discriminate between the two domains, the distributions of y_{adv} should be comparable.

To test whether the DeepTau v2p5 NN architecture is potentially capable of learning differences between data and MC domains, a pilot training with the adversarial hyperparameters set to $k_1 = 0$ and $k_2 = -1$, and started without a pre-trained network, was performed. This meant that the optimiser for the common layers did not try to achieve good classification, and that the effects of the adversarial gradient reversal in Equation 6.2 were removed. The principal and adversarial network optimisers therefore both attempted to achieve good data/MC discrimination. Training was performed for 10 000 batches on the mixed dataset (with 1000 batches validation). A plot of the y_{adv} prediction for data and MC contributions is shown on Figure 7.1. Along with the adversarial accuracy value of 69% during both training and validation, the skewed distributions on the plot confirm the ability of the network to pick up differences between data and MC based on the provided input features defined in subsection 5.1.1.

Trials using regularisation to monitor the input weights of each layer, as well as disabling certain subsets of variables yielded no conclusive results. The mismodelling is therefore most likely a multidimensional effect.

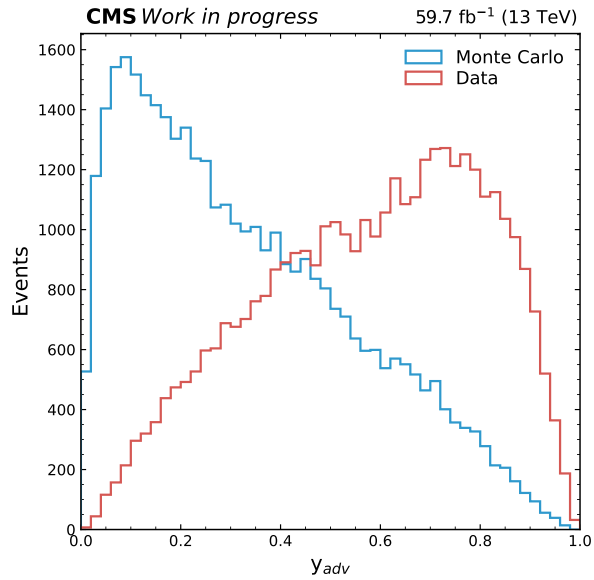


Figure 7.1: Testing ability of network to discriminate data and MC. The two distributions are skewed towards their respective domain labels (MC : 0, data : 1)

7.1.2 Choice of Starting Model

Hyperparameter optimisation for the default DeepTau v2.5 training to be used as a baseline for the adversarial training step was performed within the CMS Tau Physics Object Group (POG). First, a model was trained for 3 full epochs with the *NAdam* optimiser and a learning rate of 10^{-3} as a baseline. Various optimiser algorithms and learning rates were then tested, continuing training from the baseline for 2 epochs. The best performing models used the *NAdam* and *Adam* optimisers and learning rates of 10^{-4}. Particularly, these showed slight increases in performance against electrons compared to the 3 epoch default. The performance of the *NAdam* and *Adam* models were very similar.

The model selected as a baseline for the adversarial training step was the training continued with *NAdam* and a learning rate of 10^{-4}. This was done as slight degradation in identification performance against electrons was noted in preliminary testing of adversarial training, so a starting model with the best possible performance against this type was preferable.

The DeepTau v2p5 VS jet score distribution for this default trained model was evaluated by computing classification predictions using the adversarial testing dataset. The distribution is shown on Figure 7.2. The familiar data/MC discrepancies are present in the signal region (final bins), with the differences as high as 13.3% in the last bin.

7.1.3 Adversarial Hyperparameters k_1, k_2

The most important hyperparameters are k_1 and k_2, as their relative values dictate how much reducing the data/MC discrepancies is prioritised. Setting k_1 to zero as done in the discrimination test above is not an option as good tau type identification performance must be maintained. $k_1 = 1$ was chosen for all of the following tests, so that only k_2 must be varied, this is valid as the relative values of the classification and adversarial components are more important than the absolute values of the combined gradients during optimisation.

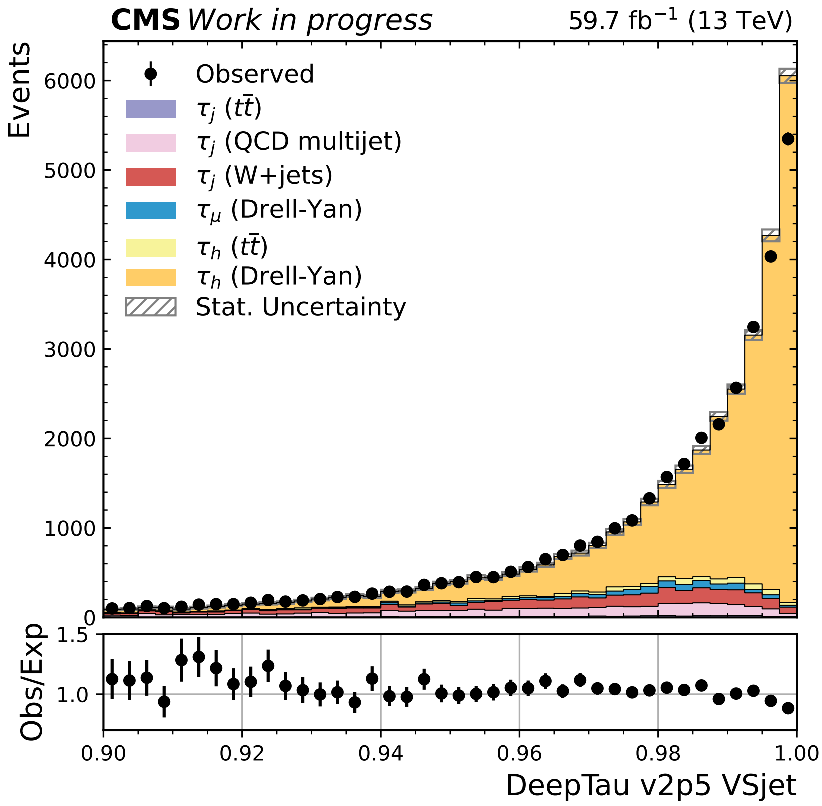


Figure 7.2: DeepTau VSjet distribution for the default trained model. The familiar data/MC discrepancies in the signal region are present, reaching as high as 13.3% in the final bin.

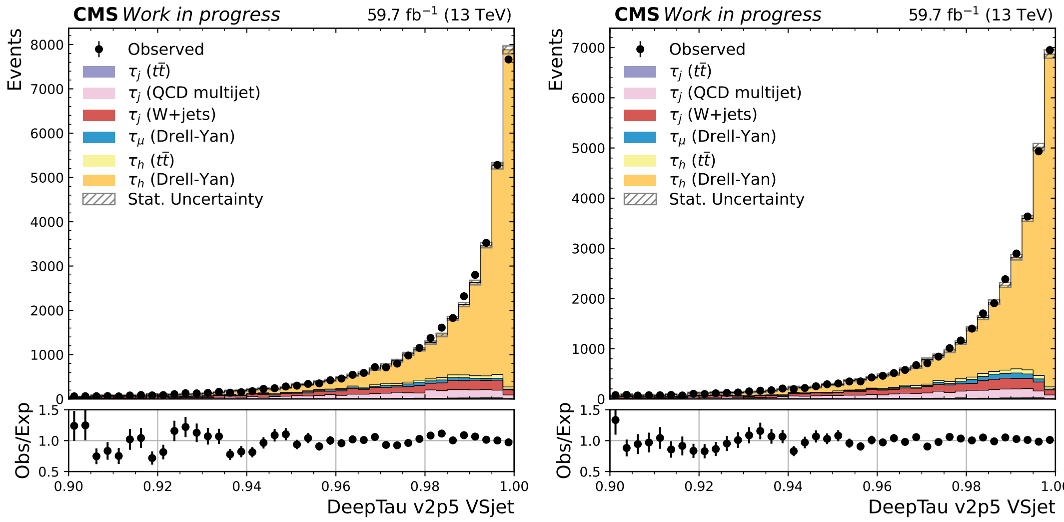
The models were trained for 50 000 batches (validation 5 000) with the chosen 5 epoch default model as a starting point. The *NAdam* optimiser was used, with a learning rate of 10^{-3} for the common and classification layers, the value typically used for DeepTau classification. A higher learning rate of 10^{-2} was selected for the adversarial layers, to ensure that the corresponding final layer weights could rapidly learn to identify data/MC discrepancies. The first adversarial hyperparameter was set to $k_1 = 1$, and various k_2 values between 1 and 10 were tested. The DeepTau v2p5 VS jet score distributions for $k_2 = 1$ and $k_2 = 10$ are shown on Figure 7.3.

The model with best data/MC agreement in the final bins was with $k_2 = 10$, shown on Figure 7.3b. Full training for values higher than this was not attempted, in order to avoid degrading tau type classification, which had been observed in preliminary trials.

7.1.4 Relative Class Importance in L_{class}

The relative importance of each class in the classification loss is determined by constants κ_e, κ_μ, κ_τ and κ_j as shown previously in Equation 5.2. These values are normalised to a sum of four. The default unnormalised values of these are $[\kappa_e = 1, \kappa_\mu = 2.5, \kappa_\tau = 5, \kappa_j = 1.5]$.

Degradation of tau identification performance against electron and muon fakes after the adversarial training step was noted during testing. To reduce these effects, particularly for discrimination against electrons, training was performed with modified loss constants, assigning more importance to electron and muon separation. Along with the



(a) $k_1 = 1,\ k_2 = 1$ (b) $k_1 = 1,\ k_2 = 10$

Figure 7.3: Comparing DeepTau v2p5 distribution for different adversarial hyperparameters. The best agreement in the signal region was for $k_1 = 1$ and $k_2 = 10$.

default values, three sets of loss constants were tested, the raw values and normalisation factors are shown in Table 7.1. Each value is multiplied by the normalisation factor for the classification loss calculation.

These models were trained with the same configuration as the $k_1 = 1$, $k_2 = 10$ model described above.

Note	κ_e	κ_μ	κ_τ	κ_j	Normalisation factor
Default	1	2.5	5	1.5	4/10
	1.5	5	5	1.5	4/13
Increased τ_e and τ_μ separation	2	5	3	1	4/11
	2	5	6	1	4/14

Table 7.1: Loss constant values tested to improve ID performance against electrons and muons.

Minimal identification performance degradation against electrons was for raw values $[\kappa_e = 2, \kappa_\mu = 5, \kappa_\tau = 6, \kappa_j = 1]$.

7.1.5 Optimiser Configuration and Minimisation Stability

The optimiser algorithms tested were *Adam* and variant *NAdam*, which DeepTau v2p5 default training hyperparameter tuning showed have similar performances. It is important to ensure stability of the adversarial training, with minimal fluctuations between runs for repeatability.

The starting learning rates were chosen as 10^{-3} for the principal optimiser (common layers + final classification), and 10^{-2} for the adversarial optimiser (final adversarial layers/domain prediction). These values were chosen to match the order of magnitude difference between the domain discrimination and tau type classification importance in the common layers done by setting $k_1 = 1$ and $k_2 = 10$.

The DeepTau discriminator distributions against jets for models trained for half an epoch (50 000 batches) were compared for the *NAdam* optimiser with no decay, and the *Adam* optimiser with a learning rate that decays by a factor 10^3 during training (*NAdam* did not support decaying learning rates). The DeepTau score distributions shown were evaluated on the adversarial testing dataset of 75 000 taus.

The results of three runs for *NAdam* with constant learning rate 10^{-3} are shown on Figure 7.4. Fluctuations between in the ratios for the final five bins range between 3 and 8 %.

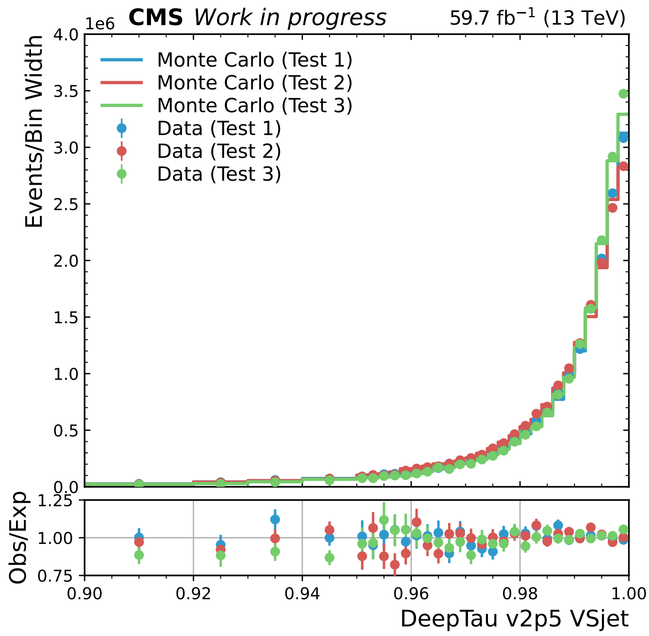


Figure 7.4: DeepTau score distributions against jets for 3 runs. *NAdam* optimisers were used with no decay in learning rate, there are noticeable fluctuations in the data/MC ratios in the signal region.

The results for *Adam* with the exponentially decaying learning rate, starting at 10^{-3} and reducing to 10^{-6} after 50 000 batches are shown on Figure 7.5. Fluctuations in the final bins were significantly reduced, with variations in the final ratios ranging between 0.7% and 3%. Similar results were found using an exponentially decaying rate with a factor 10^2 decay after half an epoch.

The *Adam* optimiser was therefore chosen with a decay rate of 10^3 after a full epoch (100 000 batches). Other algorithms could be tested in the future, but *Adam* is similar to *NAdam*, which was extensively tested during DeepTau development and is known to provide good performance.

7.2 Optimal Configuration

The final model was trained for a full epoch of 100 000 batches with the adversarial subnetwork introduced. The starting model chosen had been trained with the default DeepTau v2p5 configuration with the *NAdam* optimiser for three epochs with a learning rate of 10^{-3} followed by 2 epochs with 10^{-4}, resulting in a total of 5 epochs.

The adversarial hyperparameters were set to $k_1 = 1$ and $k_2 = 10$. The *Adam* optimisation algorithm was used, with initial adversarial and classification learning rates of 10^{-2}

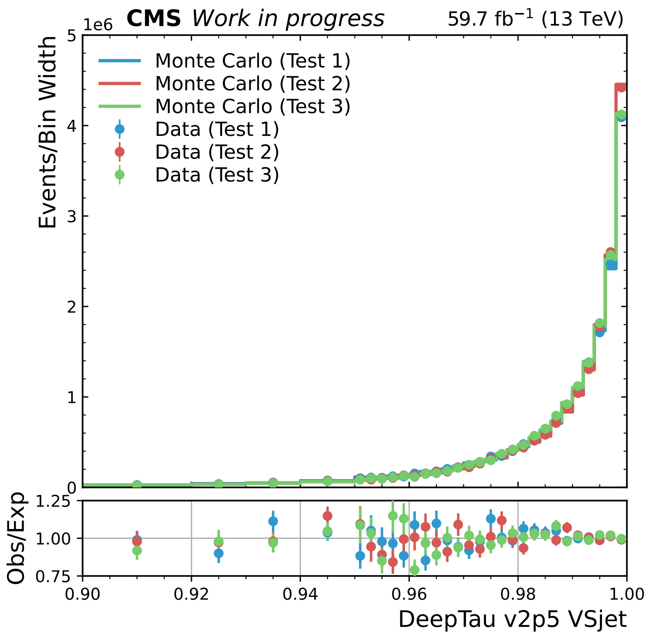


Figure 7.5: DeepTau score distributions against jets for 3 runs. *Adam* optimisers were used with a factor 10^3 decay in learning rate by the end of training. There is a reduction in fluctuations between runs in the signal region.

and 10^{-3} respectively, decaying by a factor 10^3 after a full epoch. Modified classification loss constants were used: $[\kappa_e = 2, \kappa_\mu = 5, \kappa_\tau = 6, \kappa_j = 1]$.

7.2.1 Data/MC Discrimination Performance

The data/MC discrimination performance was evaluated on the adversarial control region testing dataset of 75 000 data and MC taus. The adversarial accuracy during training and validation was 51%. The adversarial output y_{adv} is plotted for data and MC events on Figure 7.6. The two distributions are similar, with mean values (μ) of 0.4954 and 0.4943 for data and Monte Carlo respectively, and similar standard deviations (σ) of 0.013 and 0.014. The model therefore successfully avoided discriminating between data and MC events.

The impact of this on the DeepTau v2p5 discriminator distribution against quark/gluon jets for the final model (after adversarial training) is shown on Figure 7.7. The model was evaluated on the 75 000 taus from the adversarial testing dataset, and the contributions were separated into different MC event types and data. There is a significant improvement in data/MC discrepancies in the final bins after adversarial training. The differences in the final bin are reduced from 13.3% to 0.8%, the τ_h purity in this region is estimated from the proportions of different MC samples in the final model distribution to be above 96%. Agreement in the signal region overall is very good. There are very few events with scores below 0.9, which is due to the cut on DeepTau v2p1 VSjet for the adversarial dataset. The number of events for both data and MC events in the final bins is larger than for the starting model.

The adversarial training therefore successfully reduced the effects of MC mismodelling on the DeepTau VS jet score distribution.

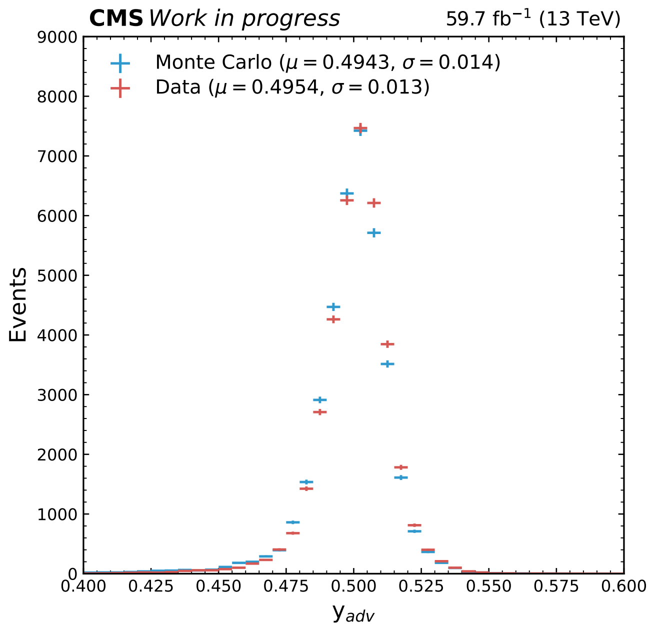


Figure 7.6: Adversarial output. The data and MC distributions are similar, the network successfully avoided discriminating between domains.

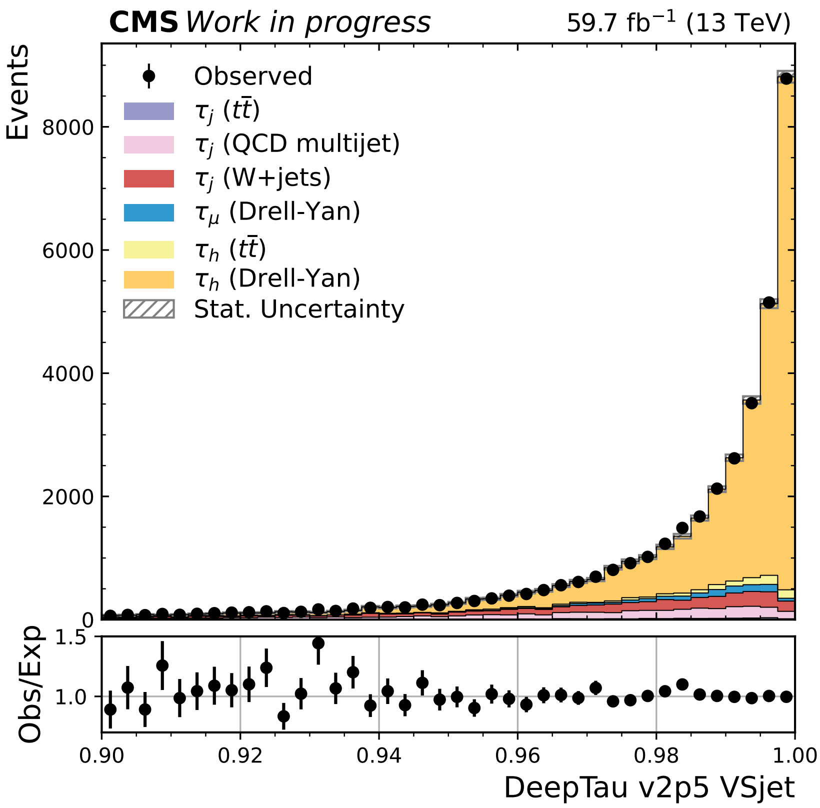


Figure 7.7: DeepTau v2p5 discriminator distribution against jets after adversarial training. There is significant improvement in data/MC agreement in the signal region, with the discrepancies in the final bin reduced from 13.3% to 0.8%, where the τ_{h} purity is estimated to be above 96%.

7.2.2 Tau Type Classification Performance

While data/MC disagreements in the DeepTau VS jet distribution have been significantly reduced in this final model, good tau type identification (ID) performance must be maintained. To evaluate identification performance, ROC curves were plotted which compare DeepTau v2p1 with DeepTau v2p5 both before and after the final adversarial training step. These show the misidentification rate (probability for an electron, muon or quark/gluon jet to be incorrectly identified as a τ_h) against the τ_h ID efficiency (probability of a true τ_h to be successfully identified).

Inclusive plots, showing all of the commonly reconstructed decay modes combined, can be found in this section. These plots are shown for the central pseudorapidity region ($0 < |\eta| < 2.3$), and are separated into low (20-100 GeV) and high (100-1000 GeV) p_T. Identification performance for individual decay modes and for the forward pseudorapidity region ($2.3 < |\eta| < 2.5$) can be found in Appendix A.

The model performances were first evaluated for MC events corresponding to the 2018 data-taking period, these are referred to as Run 2 events. The models were then evaluated on preliminary MC events corresponding to the upcoming Run 3 period (2022-2025), and the performance was compared with equivalent Run 2 events.

Working points

Working points (WP) were determined for the final DeepTau v2p5 model, with the same target efficiencies as shown in Table 5.2. The corresponding thresholds for each discriminator were calculated and are shown on Table 7.2. These are the cuts on discriminator scores that analyses should apply for each WP.

	VVVLoose	VVLoose	VLoose	Loose	Medium	Tight	VTight	VVTight
D_e	0.0990	0.2384	0.4136	0.6880	0.8704	0.9669	0.9882	0.9951
D_μ			0.2949	0.4746	0.7454	0.9172		
D_{jet}	0.4083	0.6285	0.8052	0.9233	0.9632	0.9799	0.9884	0.9931

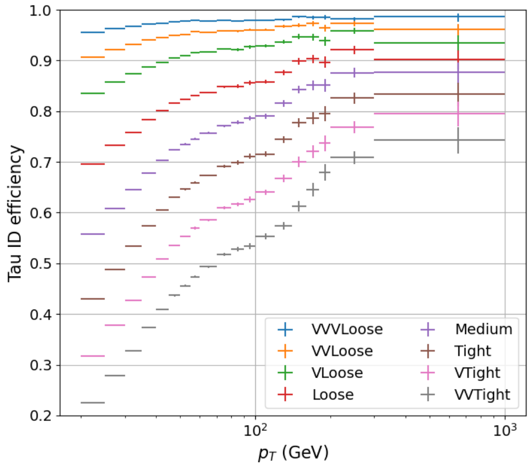
Table 7.2: Working point discriminator thresholds for the final DeepTau v2p5 model.

The τ_h identification (ID) efficiencies for different WPs as a function of p_T are shown on Figure 7.8 calculated for MC events corresponding to the 2018 data-taking period. The efficiencies generally increase as a function of p_T except for discrimination against electrons, where for the Medium to VVTight WPs there is a decrease for $p_T > 100$ GeV. This is a problem that should be addressed in future versions of DeepTau, by increasing the number of high p_T taus and their relative importance during training.

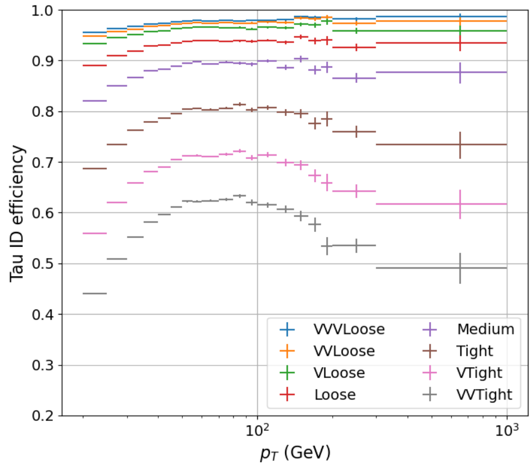
Run 2 Performance

The final DeepTau v2.5 identification performance was first evaluated on events corresponding to the Run 2 data-taking period. τ_h were selected from $\mathrm{H} \to \tau\tau$ events, τ_j from $t\bar{t}$ and τ_e and τ_μ from Drell-Yan events. The WPs for DeepTau v2p1 and the final DeepTau v2p5 model have been added to these curves and are displayed as points.

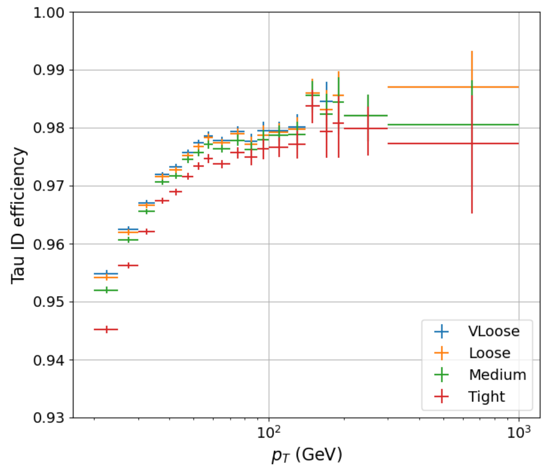
The performance against quark/gluon jets is shown on Figure 7.9. The DeepTau v2p5 performance before and after the adversarial training step was applied are almost identical, and show significantly improved performance compared to DeepTau v2p1. Good tau identification performance against jets was therefore maintained.



(a) Against jets



(b) Against electrons



(c) Against muons

Figure 7.8: τ_h ID efficiencies as a function of p_T for the final DeepTau v2p5 working points (WP). There is an increase in efficiency with p_T for against all types, except electrons where for the tighter WPs there is a decrease for $p_T > 100$ GeV.

Identification performance against electrons is shown on Figure 7.10. There is slight degradation of the performance compared to the starting v2.5 model, particularly in the low p_T region. The final model ID performance is still significantly better than v2.1, so these small losses are acceptable. Furthermore, the ID efficiency and misidentification rates will be recalculated after scale factors (SF) are calculated, this may reduce the performance degradation observed.

The identification performance against muons is shown on Figure 7.11. There is a more notable performance degradation than against other backgrounds, the final model performance is comparable to DeepTau v2.1. However, discrimination against muons is already good compared to other tau types, and is not a priority. Furthermore, there are known discrepancies between data and MC performance for muons, and adversarial training techniques to reduce these will be implemented in the coming year, which may improve performance. As for electrons, the performance will be reevaluated once the final

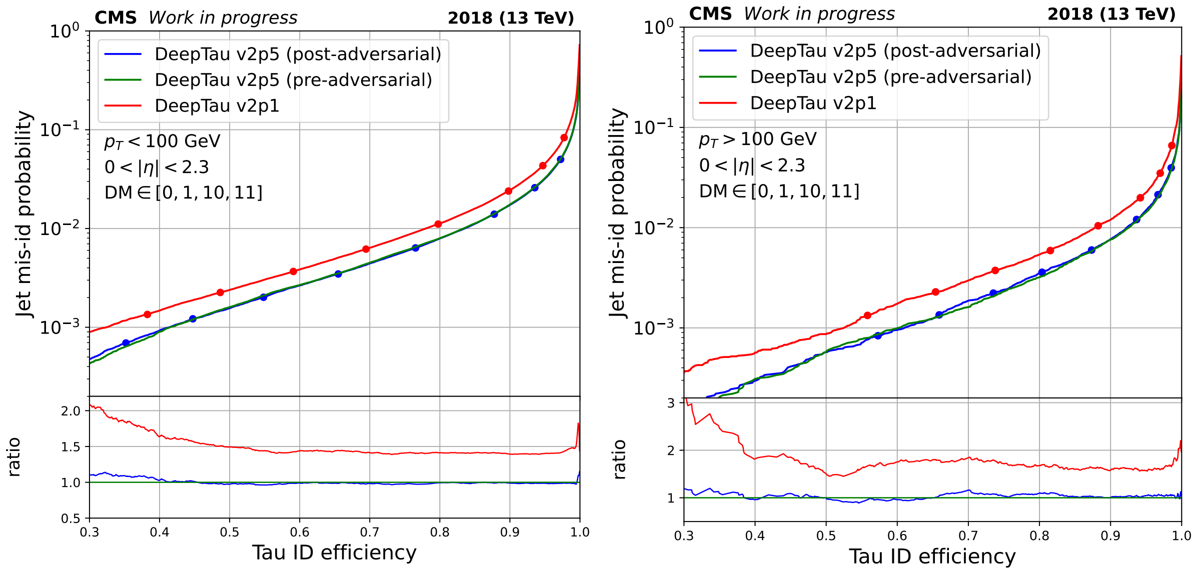


Figure 7.9: Inclusive DeepTau ID performance against jets. Good performance is maintained after adversarial training.

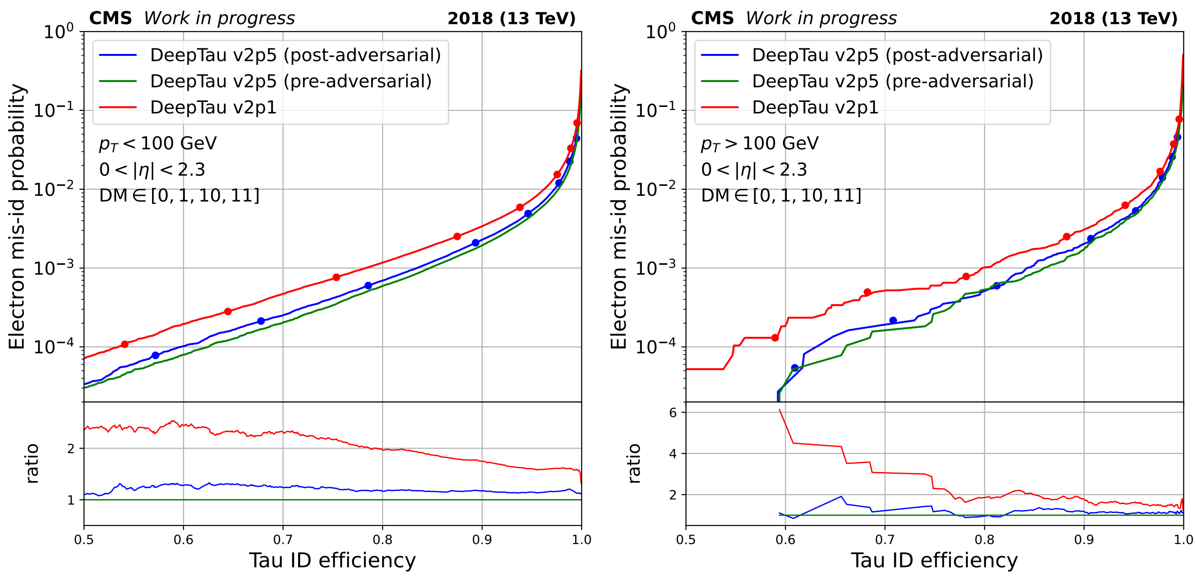


Figure 7.10: Inclusive DeepTau ID performance against electrons. Slight performance degradation after adversarial training.

SFs are computed.

Overall, good ID performance on Run 2 events was maintained after adversarial training, corresponding to a significant improvement compared to DeepTau v2p1 against most types. The observed performance losses against electrons and muons are acceptable given what is gained in DeepTau VSjet data/MC agreement. These losses may be less significant once the model with the final SFs is evaluated.

Preliminary Run 3 Performance

The identification performance of the models was then evaluated on preliminary MC events corresponding to the upcoming Run 3 data-taking period. τ_h were selected from

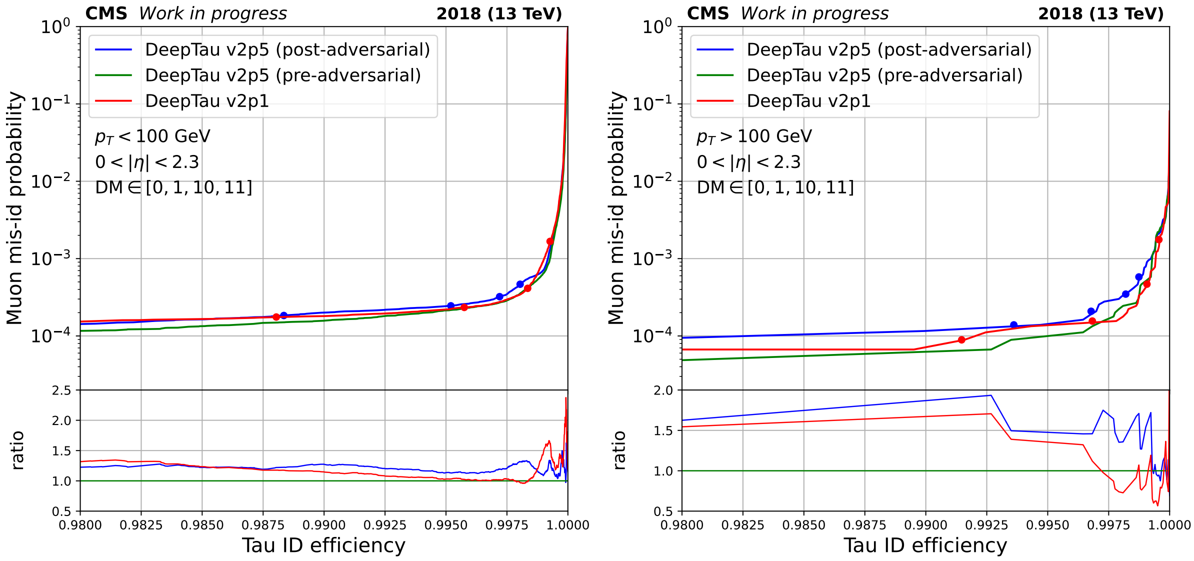


Figure 7.11: Inclusive DeepTau ID performance against muons. Acceptable performance losses after adversarial training.

Run 3 Drell-Yan events (due to lack of available $\mathrm{H} \to \tau\tau$), τ_j were selected from $t\bar{t}$ and τ_e/τ_μ from Drell-Yan. For comparison, performance was evaluated on equivalent Run 2 events: τ_e, τ_μ, τ_h from Run 2 Drell-Yan, and τ_j from Run 2 $t\bar{t}$.

The Run 3 performances against jets are compared with Run 2 on Figure 7.12. DeepTau v2p5 both before and after the adversarial training step was applied shows relatively stable performance when evaluated on Run 3 instead of Run 2, with very little degradation in both regions. DeepTau v2p1 is more sensitive to this change, with larger losses between the two periods. The Run 3 performances of DeepTau v2p5 before and after the adversarial training was applied are compatible, with no significant performance losses.

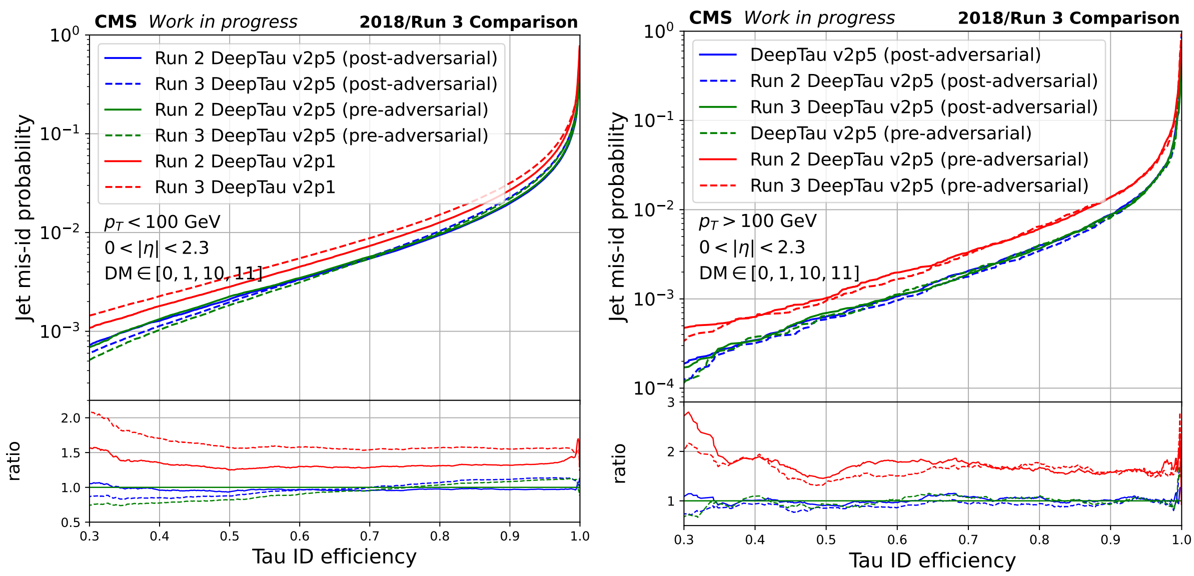


Figure 7.12: Comparison of inclusive DeepTau ID Run 2/3 performances against jets. Some degradation for Run 3 v2p1 compared to Run 2. No notable performance degradation after adversarial training.

The Run 3 performance against electrons is compared to Run 2 on Figure 7.13. There are performance losses for all models evaluated on Run 3 events, these are particularly significant in the high p_T region for low τ_h efficiencies. There is slight performance degradation after adversarial training in the low p_T region.

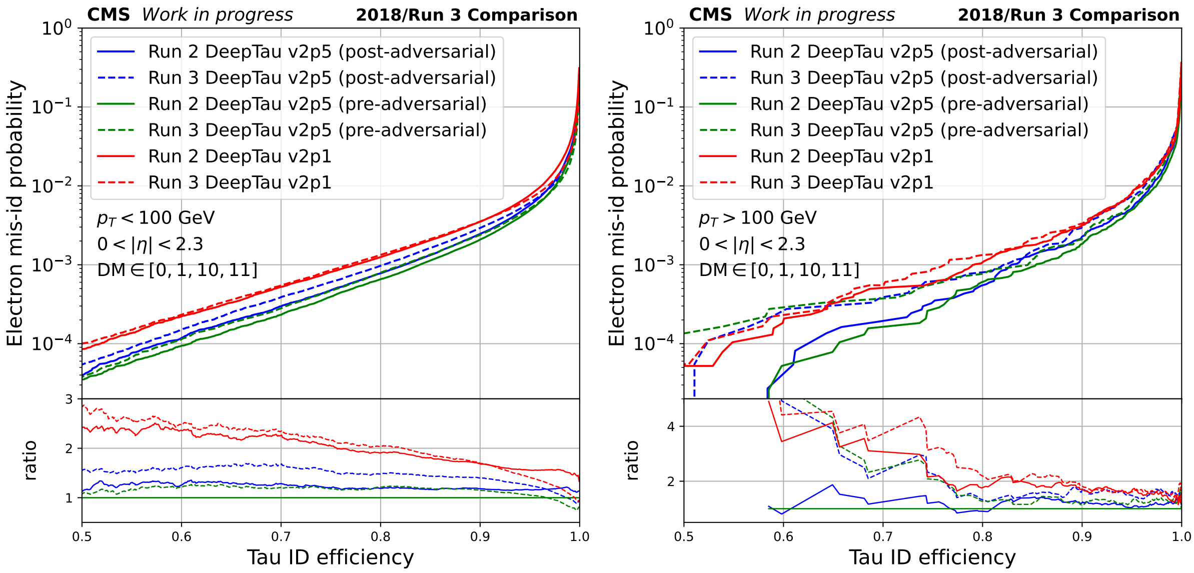


Figure 7.13: Comparison of inclusive DeepTau ID Run 2/3 performances against electrons. Performance degradation for all Run 3 models compared to Run 2, particularly for high p_T. Slight performance degradation after adversarial training.

Finally, the Run 3 performance against muons is compared to Run 2 on Figure 7.14. There is relatively stable performance between the models evaluated on Run 2 and Run 3 events in the low p_T region, but significant degradation for high p_T. There are slight performance losses for low p_T after adversarial training, with the most significant losses seen in the high p_T region, where DeepTau v2p1 outperforms post-adversarial v2p5 by a notable margin. This degradation is likely related to a strong dependence on the p_T and η spectra of the events used for evaluation. For this plot, τ_h had to be taken from Drell-Yan for a fair Run 2/3 comparison, which have different spectra than $H \to \tau\tau$ events. The effect can be seen by referring back to Figure 7.11, where the Run 2 high p_T performances between v2p1 and v2p5 post-adversarial are more compatible than the Run 2 performances shown here.

The overall Run 3 performances for all models are degraded compared to Run 2, particularly against electrons and muons. These differences can likely be reduced in the future, as DeepTau v2p5 was trained exclusively on events corresponding to the end of Run 2 (2018). There are plans to include samples from Run 3 MC events in the training dataset for the next DeepTau versions.

Similarly to Run 2, performance for the DeepTau v2p5 model before and after training was compatible for discrimination against jets. There was some slight performance degradation against electrons, and quite significant degradation against muons, particularly for high p_T. This is acceptable given the gains in data/MC agreement.

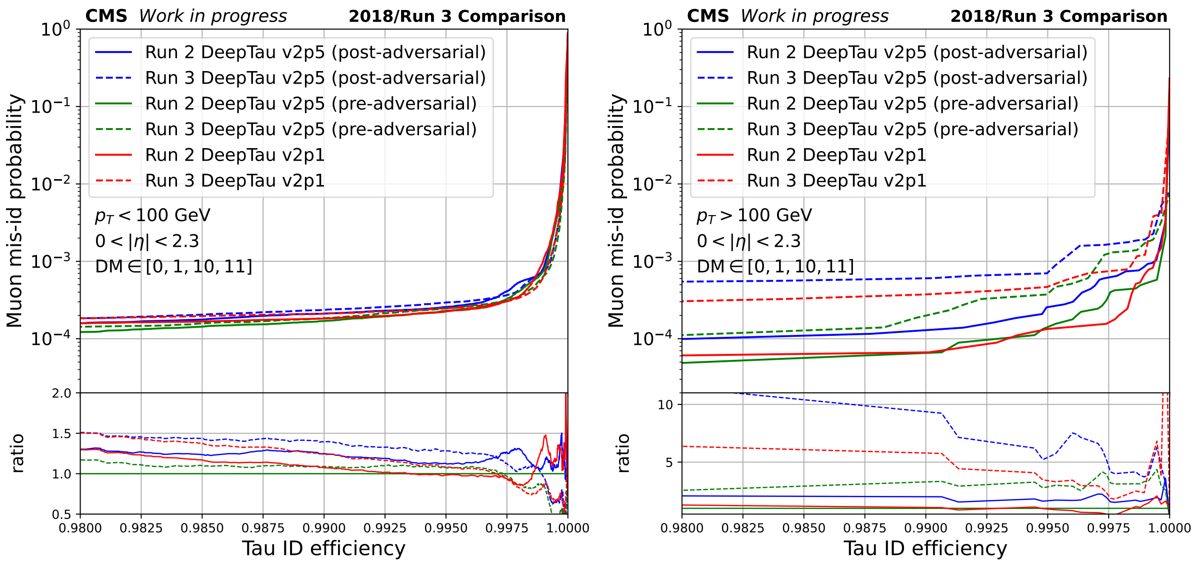


Figure 7.14: Comparison of inclusive DeepTau ID Run 2/3 performances against muons. Significant performance degradation for all Run 3 models compared to Run 2 for high p_T. Significant performance degradation after adversarial training for high p_T.

Chapter 8

Conclusion

The training of the new version of the deep convolutional neural network, DeepTau v2.5 (v2p5) which simultaneously discriminates reconstructed hadronic tau (τ_h) candidates against electrons, muons and quark/gluon jets at CMS was presented. The final model includes domain adaptation techniques to reduce the effects of mismodelling in the Monte Carlo (MC) simulated events used for training on the discrimination performance against jets.

After an introduction to tau leptons in the Standard Model, the CMS experiment and τ_h reconstruction were presented. This was followed by descriptions of machine learning concepts and the DeepTau algorithm along with the previous identification performance discrepancies. The adversarial training methods implemented in this work, the hyperparameter tuning that led to the final model, and the resulting performance were then discussed.

Domain adaptation using adversarial machine learning techniques was introduced into the DeepTau v2p5 workflow. This involved adding an adversarial subnetwork, that attempted to discriminate between collider data and Monte Carlo (MC) events, in parallel to, and using the same processed features, as the tau type classifier. The goal was to maximise classification performance with poor data/MC discrimination. Modifications were made to the network architecture and gradient calculation used for trainable parameter optimisation.

The optimal configuration was determined after testing various hyperparameter combinations, and the resulting model was successfully prevented from discriminating between data and MC. This led to a significant improvement in agreement in the signal region of the DeepTau discriminator distribution against jets, with discrepancies being reduced from 13.3% to 0.80% in the region where genuine τ_h purity is above 96%.

Good overall tau type identification (ID) performance on 2018 MC events was maintained, with slight degradation against electrons, and acceptable performance losses against muons. Preliminary ID performance on MC events corresponding to the Run 3 (2022-2025) data taking period was slightly worsened, but improved with respect to DeepTau v2p1 in most regions. This degradation in performance was expected, because the NN was not trained on Run 3 datasets due to their limited availability at the time when the training was performed.

The final adversarial trained model has been reviewed and approved by the CMS Tau Physics Object Group (Tau POG). The corresponding code has been integrated into central CMS software. DeepTau v2p5 will be used for early Run 3 physics analyses, as well as Run 2 and early Run 3 combinations.

The CMS Tau POG plans to implement similar adversarial domain adaptation methods to reduce the effects of mismodelling in regions dominated by the other tau types (τ_e, τ_μ and τ_j). The first of these will be for the discriminator against muons, for which there are large discrepancies between the data and simulations in the endcap regions, reaching as high as factor 5.

Acknowledgements

First and foremost, I would like to thank my supervisor Konstantin Androsov, for providing excellent guidance throughout my project, for giving me the opportunity to work on the development of DeepTau and to present version 2.5 at CMS week.

I would like to thank Imperial and EPFL for giving me the opportunity to study abroad, particularly Gavin Davies for his support as academic visitor, and the Imperial HEP group for registering me at CMS. I am very grateful to Lesya Shchutska for hosting me in her group, and to Olivier Schneider for helping to find the project, and teaching me my first particle physics course. I would also like to thank my personal tutor Michael McCann, for his guidance during the last 3 years and for fuelling my interest in HEP.

I would like to thank the post-doctoral researchers and PhD students, past and present, of LPHE for providing such a friendly and welcoming atmosphere, and for helping to make my time in Lausanne so enjoyable.

I would also like to thank everyone in the tau algorithms subgroup, particularly others involved in the development of DeepTau version 2.5: Oleg Filatov, Luca Guzzi and Mykyta Shchedrolosiev. I am also grateful to George Uttley for providing $H \to \tau\tau$ analysis distributions, and to Izaak Neutelings for the DeepTau v2.1 score distribution showing the original discrepancies.

Finally, I would like to thank my friends and family, in Switzerland and abroad, for their unconditional support throughout this year.

Bibliography

[1] CMS Collaboration, "The CMS experiment at the CERN LHC," *JINST*, vol. 3, 2008. DOI: 10.1088/1748-0221/3/08/S08004.

[2] CMS Collaboration, "Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV," *JINST*, vol. 13, 2018. DOI: 10.1088/1748-0221/13/10/P10005.

[3] CMS Collaboration, "Reconstruction and identification of τ lepton decays to hadrons and ν_τ at CMS," *JINST*, vol. 11, 2016. DOI: 10.1088/1748-0221/11/01/P01019.

[4] CMS Collaboration, "Performance of tau-lepton reconstruction and identification in CMS," *JINST*, vol. 7, 2012. DOI: 10.1088/1748-0221/7/01/P01001.

[5] CMS Collaboration, "Identification of hadronic tau lepton decays using a deep neural network," Submitted to *JINST*, 2022. arXiv: 2201.08458 [hep-ex].

[6] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 2014. DOI: 10.48550/ARXIV.1409.7495.

[7] CMS Collaboration, "A deep neural network to search for new long-lived particles decaying to jets," *Machine Learning: Science and Technology*, vol. 1, 2020. DOI: 10.1088/2632-2153/ab9023.

[8] Particle Data Group, P. A. Zyla, *et al.*, "Review of particle physics," *Prog. Theor. Exp. Phys.*, vol. 2020, 2020. DOI: 10.1093/ptep/ptaa104.

[9] P. W. Higgs, "Broken symmetries and the masses of gauge bosons," *Phys. Rev. Lett.*, vol. 13, 1964. DOI: 10.1103/PhysRevLett.13.508.

[10] F. Englert and R. Brout, "Broken symmetry and the mass of gauge vector mesons," *Phys. Rev. Lett.*, vol. 13, 1964. DOI: 10.1103/PhysRevLett.13.321.

[11] M. L. Perl *et al.*, "Evidence for anomalous lepton production in $e^+ - e^-$ annihilation," *Phys. Rev. Lett.*, vol. 35, 1975. DOI: 10.1103/PhysRevLett.35.1489.

[12] W. Bacino *et al.*, "Measurement of the threshold behavior of $\tau^+\tau^-$ production in e^+e^- annihilation," *Phys. Rev. Lett.*, vol. 41, 1978. DOI: 10.1103/PhysRevLett.41.13.

[13] DASP Collaboration, "Measurements of tau Decay Modes and a Precise Determination of the Mass," *Phys. Lett. B*, vol. 73, 1978. DOI: 10.1016/0370-2693(78)90183-1.

[14] G. J. Feldman *et al.*, "Measurement of the τ lifetime," *Phys. Rev. Lett.*, vol. 48, 1982. DOI: 10.1103/PhysRevLett.48.66.

[15] CMS Collaboration, "Observation of the Higgs boson decay to a pair of τ leptons with the CMS detector," *Phys. Lett. B*, vol. 779, 2018. DOI: 10.1016/j.physletb.2018.02.004.

[16] CMS Collaboration, "Search for Higgs boson pair production in events with two bottom quarks and two tau leptons in proton–proton collisions at $\sqrt{s} = 13$ TeV," *Phys. Lett. B*, vol. 778, 2018. DOI: 10.1016/j.physletb.2018.01.001.

[17] CMS Collaboration, "Analysis of the CP structure of the Yukawa coupling between the Higgs boson and τ leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 06, 2022. DOI: 10.1007/JHEP06(2022)012.

[18] CMS Collaboration, "Measurements of Higgs boson production in the decay channel with a pair of τ leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV," 2022. DOI: 10.48550/ARXIV.2204.12957.

[19] CMS Collaboration, "Search for additional neutral MSSM Higgs bosons in the $\tau\tau$ final state in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 09, 2018. DOI: 10.1007/JHEP09(2018)007.

[20] "Searches for additional Higgs bosons and vector leptoquarks in $\tau\tau$ final states in proton-proton collisions at $\sqrt{s} = 13$ TeV," 2022. [Online]. Available: https://cds.cern.ch/record/2803739.

[21] CMS Collaboration, "Search for lepton flavour violating decays of a neutral heavy Higgs boson to $\mu\tau$ and $e\tau$ in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 03, 2020. DOI: 10.1007/JHEP03(2020)103.

[22] CMS Collaboration, "Search for heavy neutrinos and third-generation leptoquarks in hadronic states of two τ leptons and two jets in proton-proton collisions at $\sqrt{s} =$ 13 TeV," *JHEP*, vol. 03, 2019. DOI: 10.1007/JHEP03(2019)170.

[23] CMS Collaboration, "Search for direct pair production of supersymmetric partners to the τ lepton in proton-proton collisions at $\sqrt{s} = 13$ TeV," *Eur. Phys. J. C*, vol. 80, 2020. DOI: 10.1140/epjc/s10052-020-7739-7.

[24] L. Evans and P. Bryant, "LHC machine," *JINST*, vol. 3, 2008. DOI: 10.1088/1748-0221/3/08/s08001.

[25] L. Arnaudon *et al.*, "Linac4 Technical Design Report," CERN, Geneva, Tech. Rep., 2006. [Online]. Available: https://cds.cern.ch/record/1004186.

[26] H. Bartosik and G. Rumolo, "Performance of the lhc injector chain after the upgrade and potential development," 2022. DOI: 10.48550/ARXIV.2203.09202.

[27] CMS Collaboration, "Precision luminosity measurement in proton-proton collisions at $\sqrt{s} = 13$ TeV in 2015 and 2016 at CMS," *Eur. Phys. J. C*, vol. 81, 2021. DOI: 10.1140/epjc/s10052-021-09538-2.

[28] CMS Collaboration, "CMS luminosity measurement for the 2017 data-taking period at $\sqrt{s} = 13$ TeV," CMS Physics Analysis Summary CMS-PAS-LUM-17-004, 2018. [Online]. Available: https://cds.cern.ch/record/2621960.

[29] CMS Collaboration, "CMS luminosity measurement for the 2018 data-taking period at $\sqrt{s} = 13$ TeV," CMS Physics Analysis Summary CMS-PAS-LUM-18-002, 2019. [Online]. Available: https://cds.cern.ch/record/2676164.

[30] A. Sirunyan, "Pileup mitigation at CMS in 13 TeV data," *Journal of Instrumentation*, vol. 15, no. 09, 2020. DOI: 10.1088/1748-0221/15/09/p09018.

[31] I. Neutelings. "Figures related to the CMS detector and analysis of proton-proton collision data." (2017), [Online]. Available: https://tikz.net/tag/cms/ (visited on 05/19/2022).

[32] CMS Tracker Group of the CMS Collaboration Collaboration, "The CMS Phase-1 Pixel Detector Upgrade," *JINST*, vol. 16, 2020. DOI: 10.1088/1748-0221/16/02/P02027.

[33] CMS Collaboration, "The CMS electromagnetic calorimeter project: Technical Design Report," 1997. [Online]. Available: https://cds.cern.ch/record/349375.

[34] CMS Collaboration, "The CMS hadron calorimeter project: Technical Design Report," 1997. [Online]. Available: https://cds.cern.ch/record/357153.

[35] CMS Collaboration, "Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13\,\text{TeV}$," *JINST*, vol. 15, 2020. DOI: 10.1088/1748-0221/15/10/P10017.

[36] CMS Collaboration, "The CMS trigger system," *JINST*, vol. 12, 2017. DOI: 10.1088/1748-0221/12/01/P01020.

[37] CMS Collaboration, "Particle-flow reconstruction and global event description with the CMS detector," *JINST*, vol. 12, 2017. DOI: 10.1088/1748-0221/12/10/P10003.

[38] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *JHEP*, vol. 04, 2008. DOI: 10.1088/1126-6708/2008/04/063.

[39] M. Cacciari, G. P. Salam, and G. Soyez, "FASTJET user manual," *Eur. Phys. J. C*, vol. 72, 2012. DOI: 10.1140/epjc/s10052-012-1896-2.

[40] CMS Collaboration, "Performance of missing transverse momentum reconstruction in proton-proton collisions at $\sqrt{s} = 13$ TeV using the CMS detector," *JINST*, vol. 14, 2019. DOI: 10.1088/1748-0221/14/07/P07004.

[41] CMS Collaboration, "Performance of electron reconstruction and selection with the CMS detector in proton-proton collisions at $\sqrt{s} = 8$ TeV," *JINST*, vol. 10, 2015. DOI: 10.1088/1748-0221/10/06/P06005.

[42] CMS Collaboration, "Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV," *JINST*, vol. 13, 2018. DOI: 10.1088/1748-0221/13/06/P06015.

[43] J. Alwall *et al.*, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations," *JHEP*, vol. 07, 2014. DOI: 10.1007/JHEP07(2014)079.

[44] S. Alioli, P. Nason, C. Oleari, and E. Re, "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX," *JHEP*, vol. 06, 2010. DOI: 10.1007/JHEP06(2010)043.

[45] J. M. Campbell, R. K. Ellis, P. Nason, and E. Re, "Top-pair production and decay at NLO matched with parton showers," *JHEP*, vol. 04, 2015. DOI: 10.1007/JHEP04(2015)114.

[46] T. Sjöstrand *et al.*, "An introduction to PYTHIA 8.2," *Comput. Phys. Commun.*, vol. 191, 2015. DOI: 10.1016/j.cpc.2015.01.024.

[47] CMS Collaboration, "Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements," *Eur. Phys. J. C*, vol. 80, 2020. DOI: 10.1140/epjc/s10052-019-7499-4.

[48] GEANT4 Collaboration, "GEANT4—a simulation toolkit," *Nucl. Instrum. Meth. A*, vol. 506, 2003. DOI: 10.1016/S0168-9002(03)01368-8.

[49] F. Chollet *et al.* "Keras." (2015), [Online]. Available: https://keras.io.

[50] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, 1989. DOI: 10.1162/neco.1989.1.4.541.

[51] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," 1990. [Online]. Available: http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, 1998. DOI: 10.1109/5.726791.

[53] IBM Cloud Education. "Convolutional neural networks." (2020), [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks (visited on 06/30/2022).

[54] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012. arXiv: 1207.0580.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," 2015. DOI: 10.48550/ARXIV.1502.01852.

[57] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.

[58] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *TPAMI*, vol. 42, 2018. DOI: 10.1109/TPAMI.2018.2858826.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Conference Track Proceedings, 3rd International Conference on Learning Representations (ICLR 2015)*, 2015. arXiv: 1412.6980.

[60] T. Dozat, "Incorporating Nesterov momentum into ADAM," in *Conference Track Proceedings, 4th International Conference on Learning Representations (ICLR 2016)*, 2016.

[61] D. Bertolini, P. Harris, M. Low, and N. Tran, "Pileup per particle identification," *JHEP*, vol. 10, 2014. DOI: 10.1007/JHEP10(2014)059.

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. arXiv: 1502.03167.

[63] M. Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[64] CMS Tau POG. "Tau machine learning tools." (2022), [Online]. Available: https://github.com/cms-tau-pog/TauMLTools (visited on 07/01/2022).

Appendix A

Identification Performance Plots

A.1 Run 2 Samples

A.1.1 Forward Region all DMs

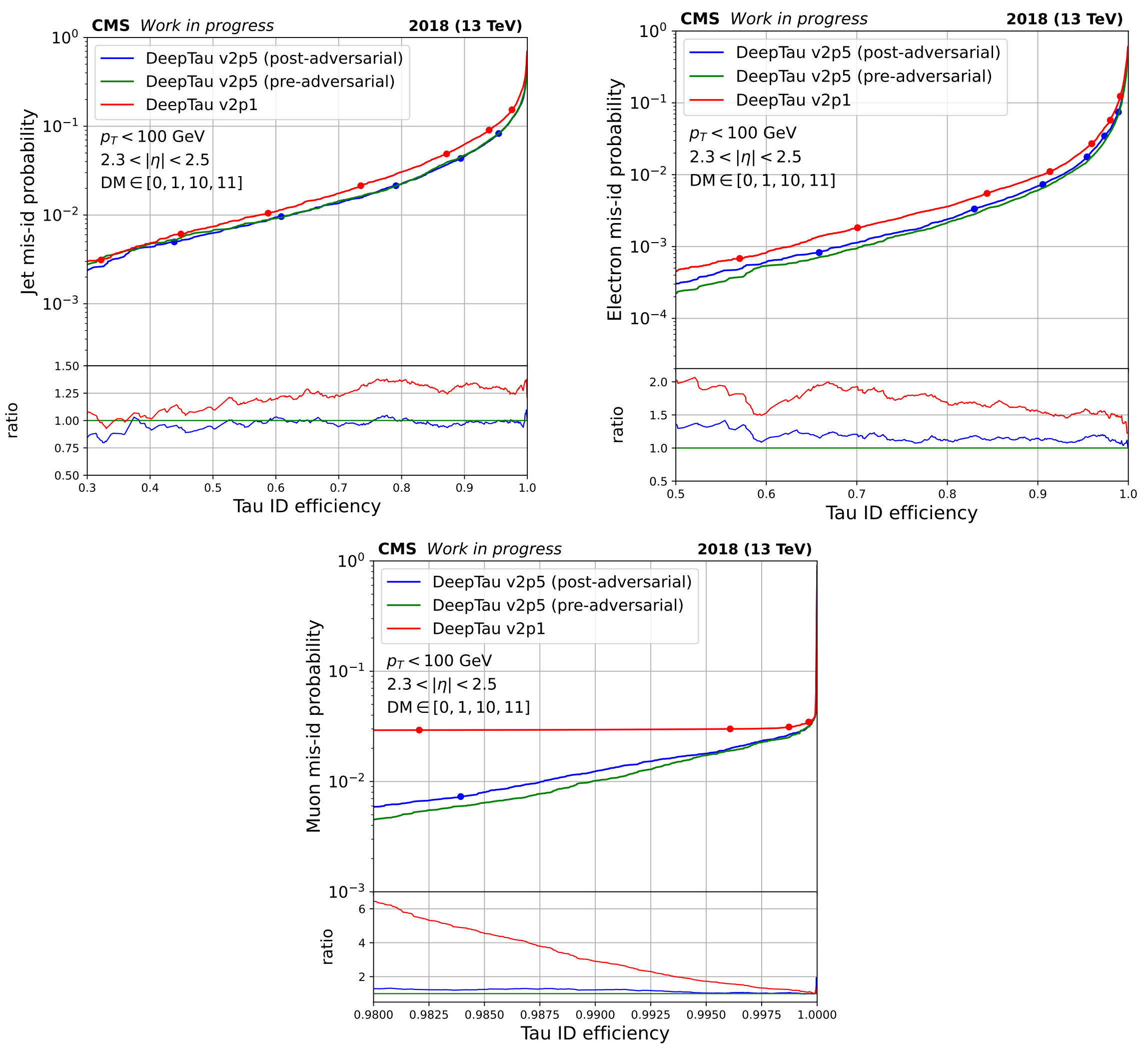


Figure A.1: Run 2 inclusive identification performance in the forward pseudorapidity region.

A.1.2 Central Region by DM

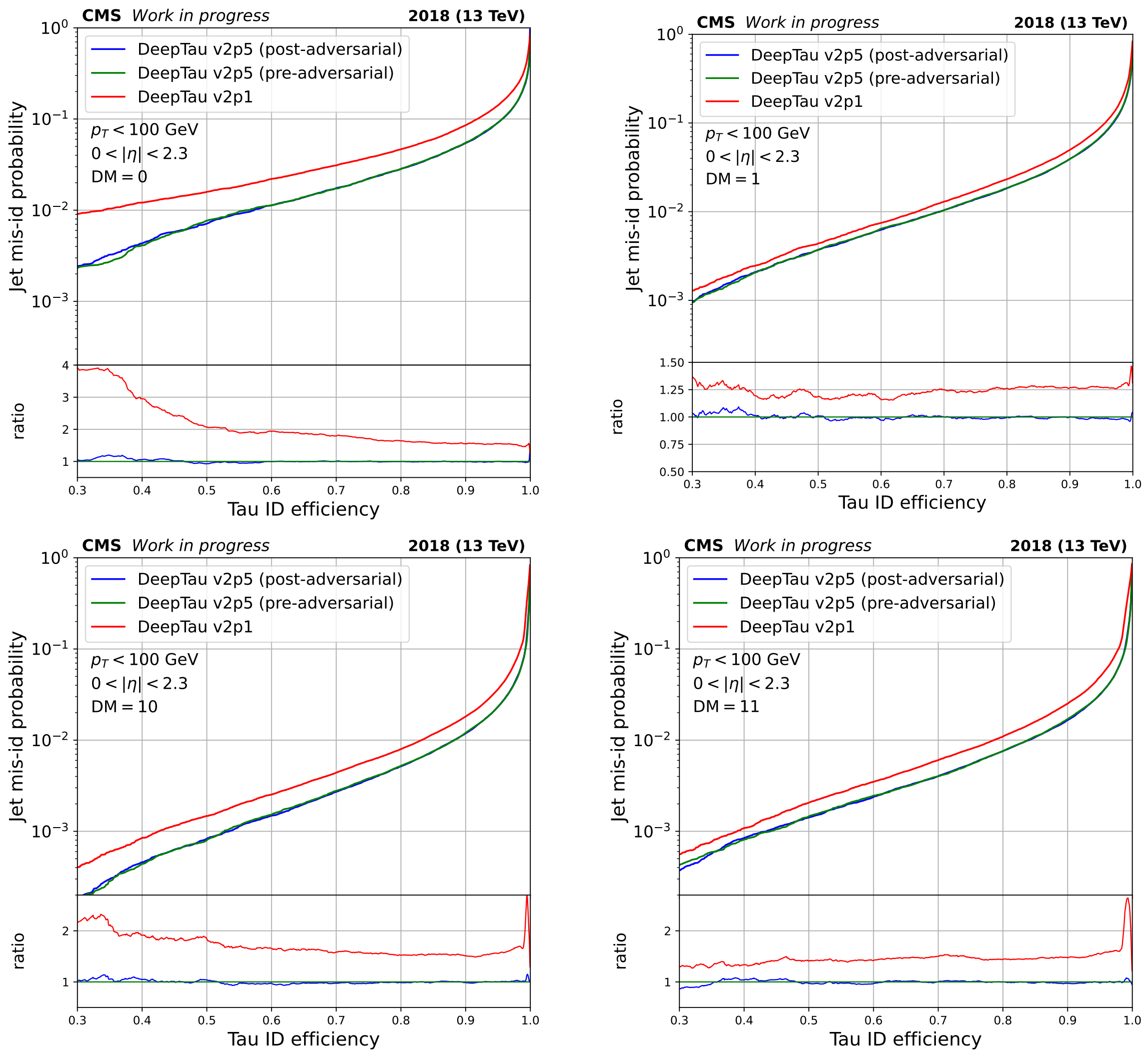


Figure A.2: Run 2 individual DM identification performance against jets in the low p_T central region.

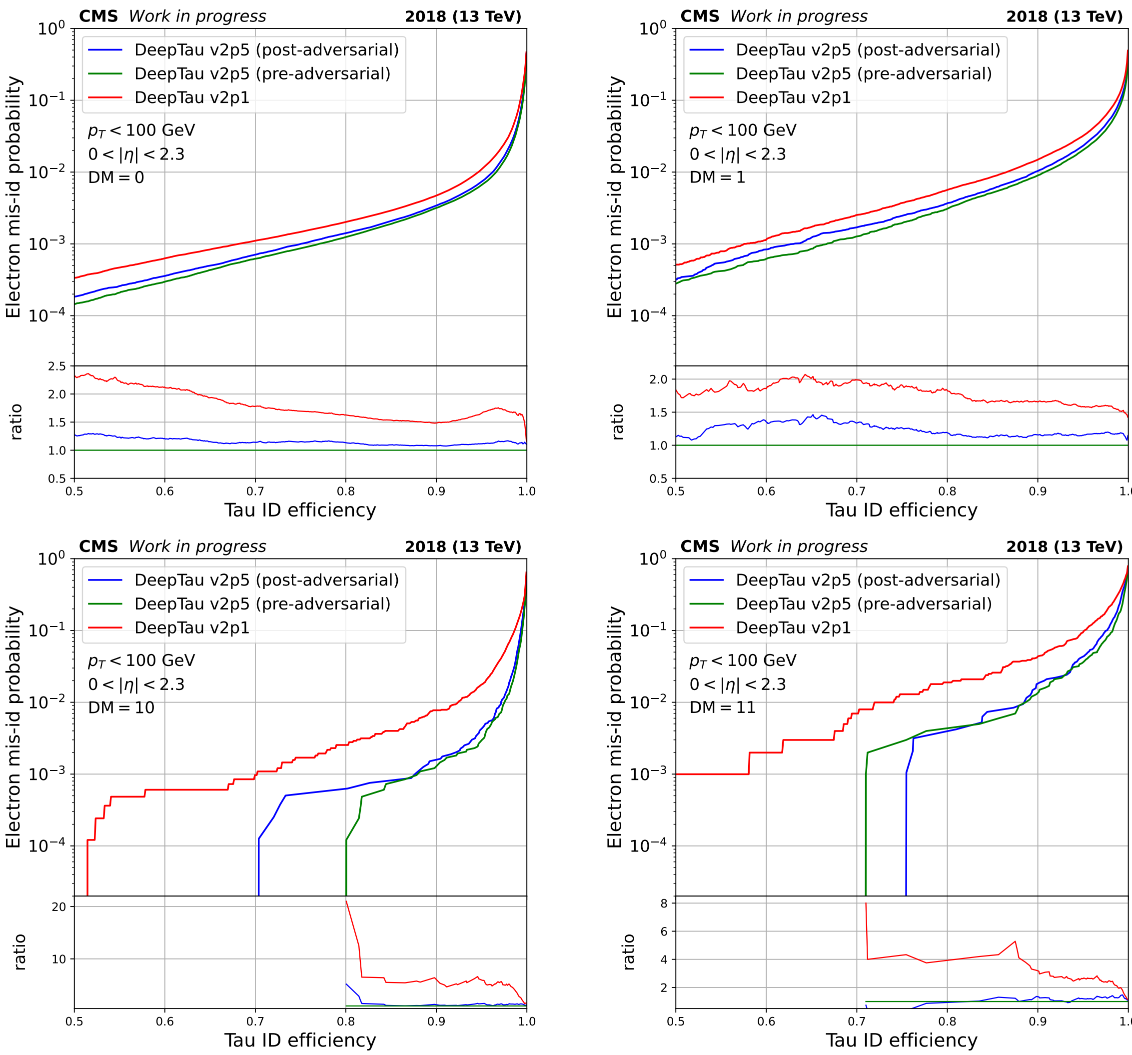


Figure A.3: Run 2 individual DM identification performance against electrons in the low p_T central region.

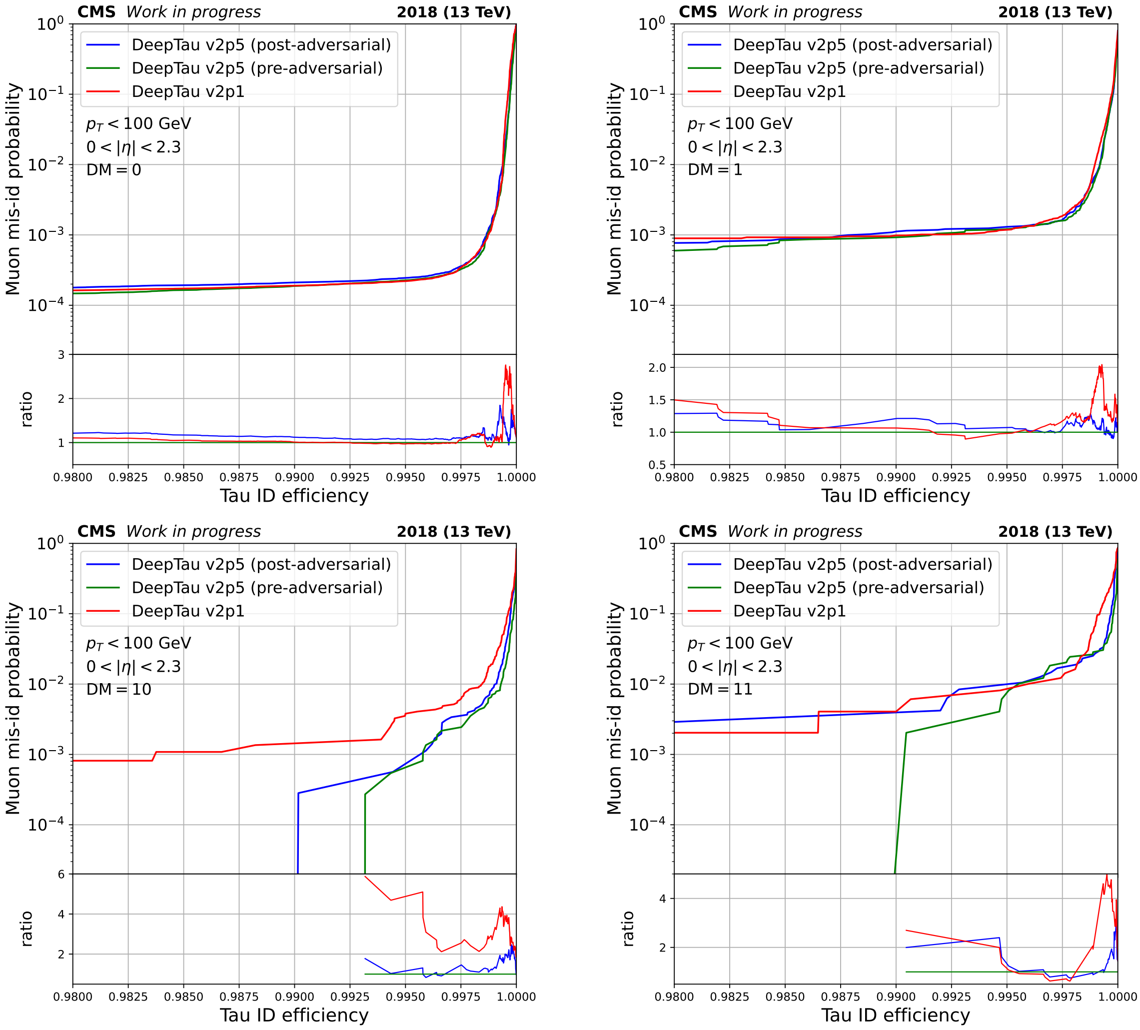


Figure A.4: Run 2 individual DM identification performance against muons for individual decay modes in the low p_T central region.

A.1.3 Forward Region by DM

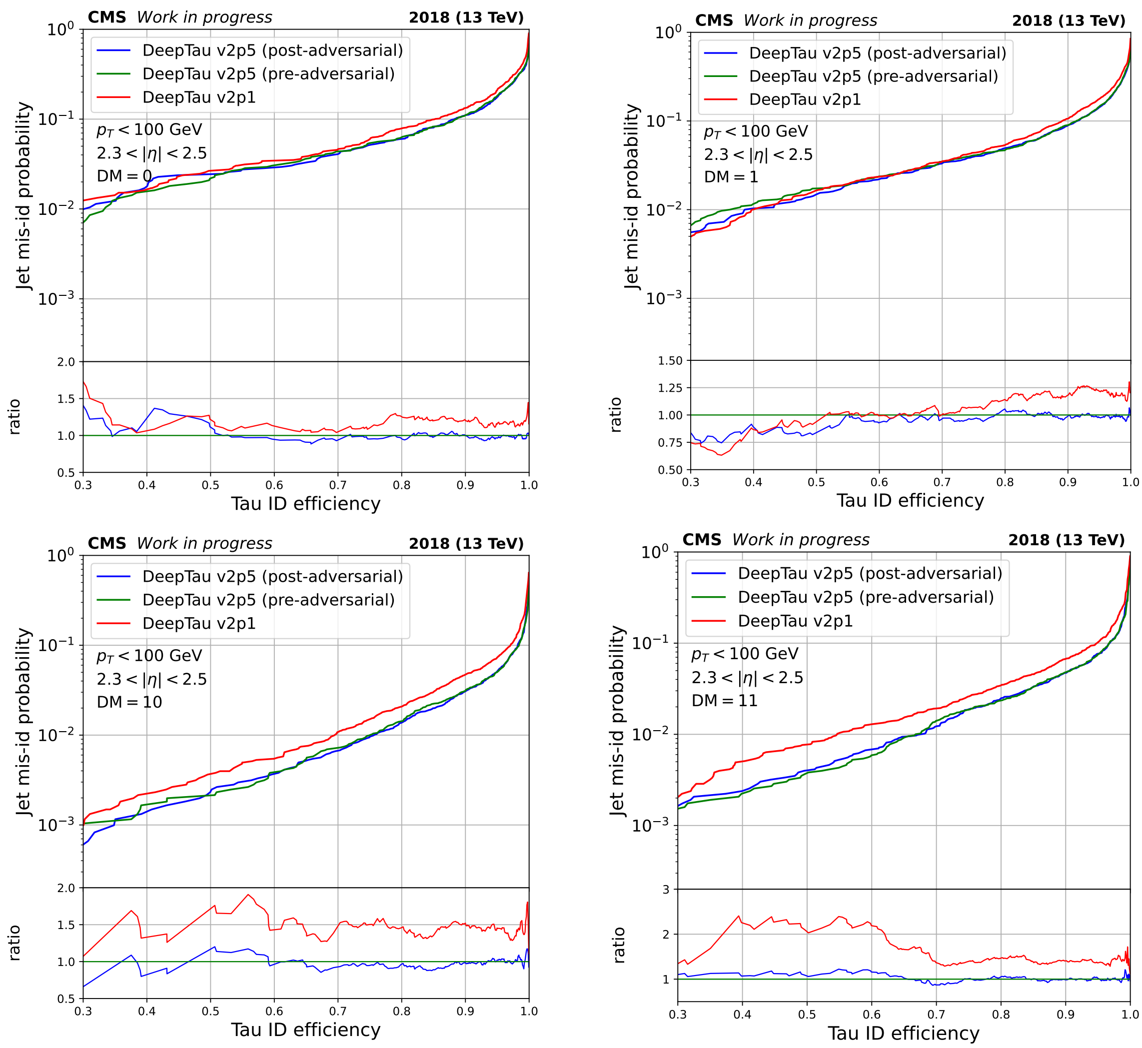


Figure A.5: Run 2 individual DM identification performance against jets in the low p_T forward region.

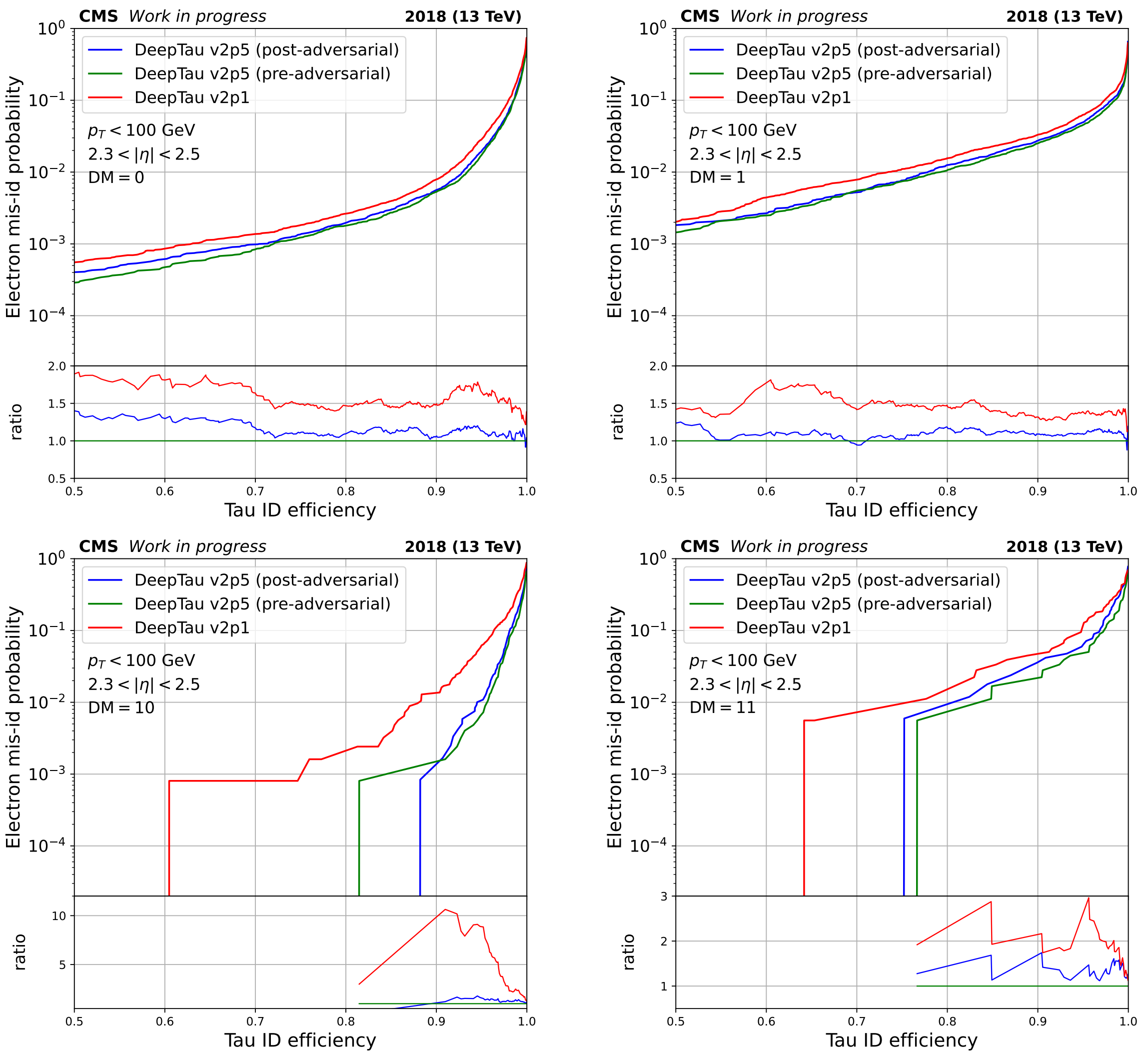


Figure A.6: Run 2 individual DM identification performance against electrons in the low p_T forward region.

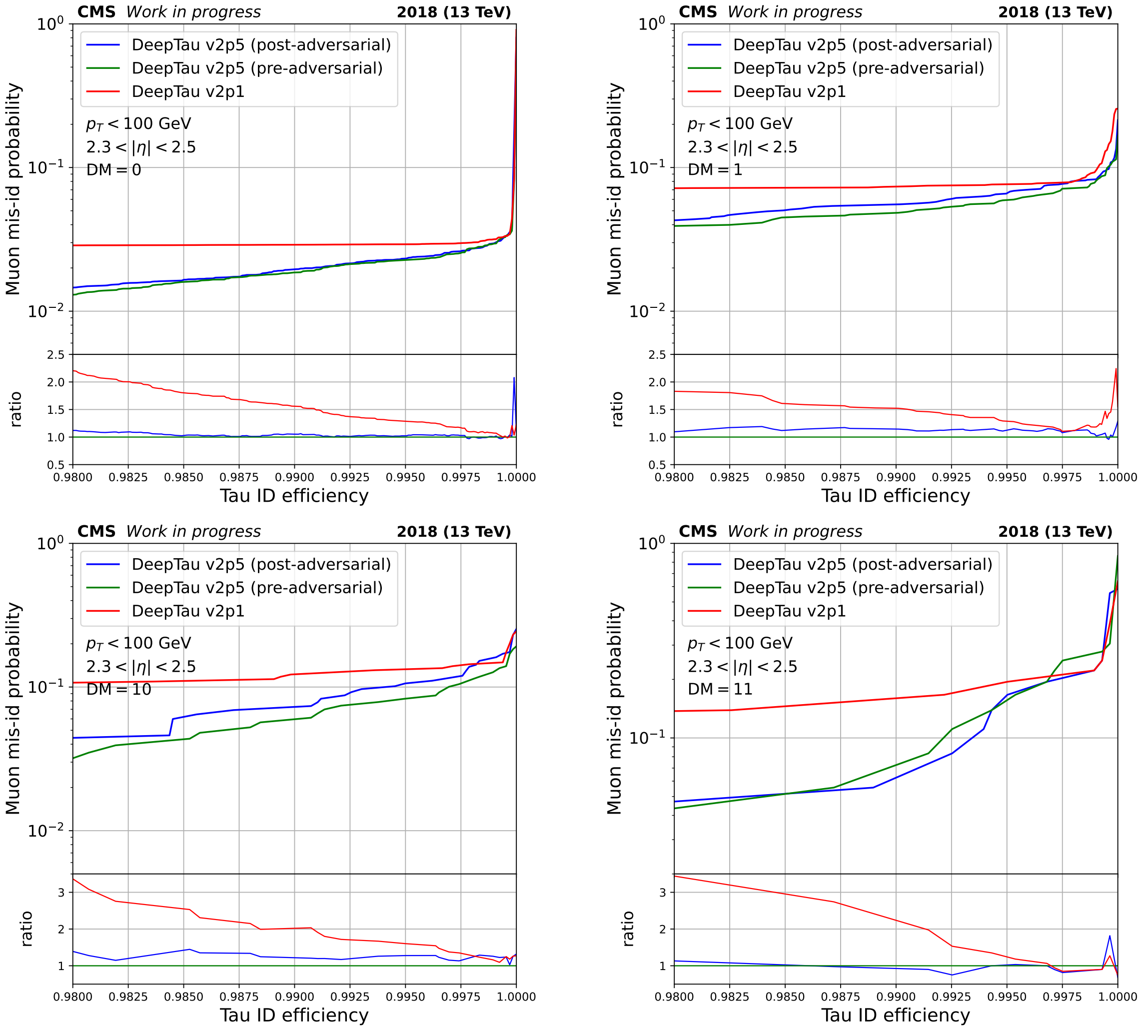


Figure A.7: Run 2 individual DM identification performance against muons in the low p_T forward region.

A.2 Run 2/Run 3 Comparison

A.2.1 Forward Region all DMs

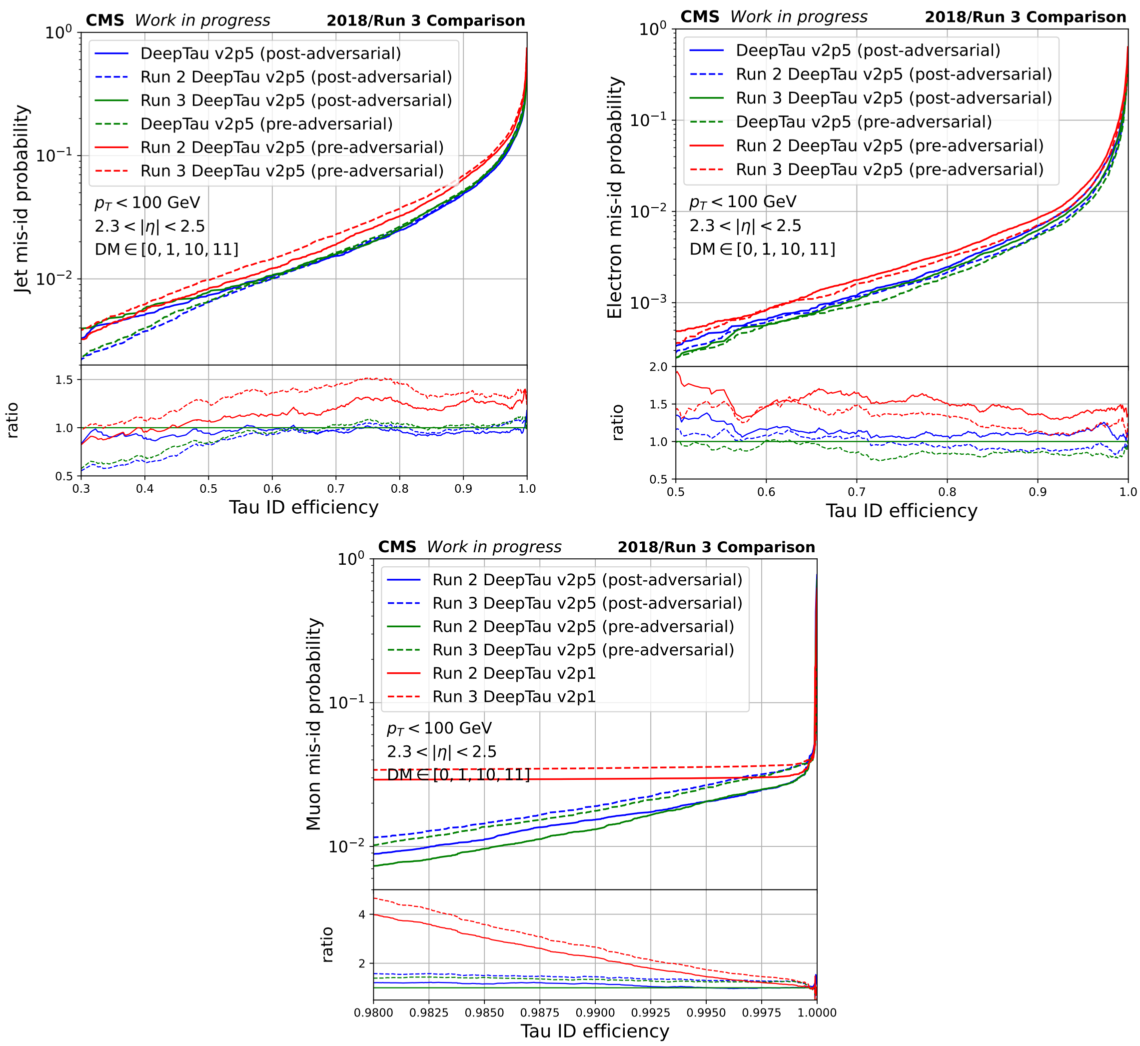


Figure A.8: Comparison of Run 2/3 inclusive identification performance in the forward pseudorapidity region.

A.2.2 Central Region by DM

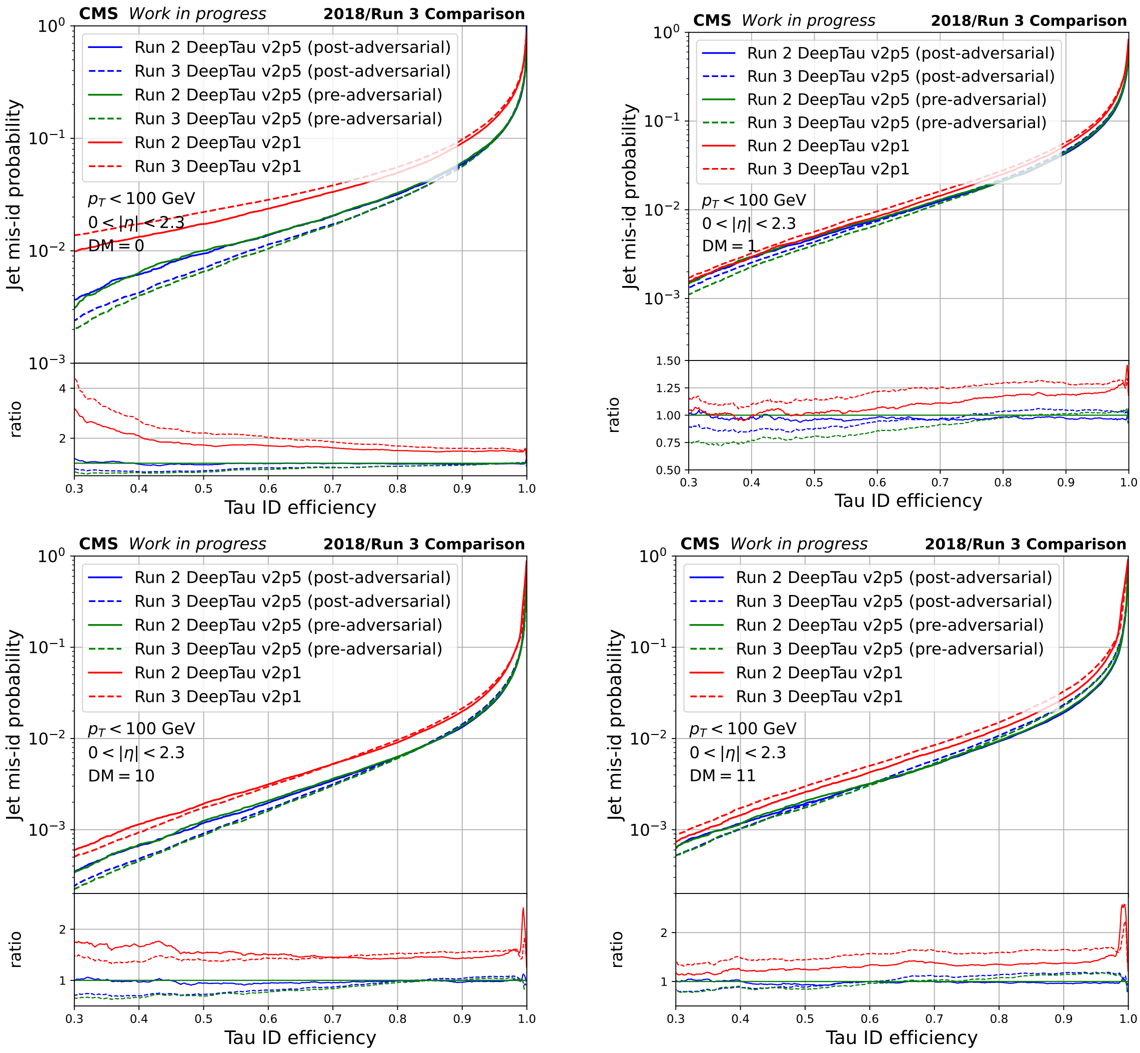


Figure A.9: Comparison of Run 2/3 individual DM identification performance against jets in the low p_T central region.

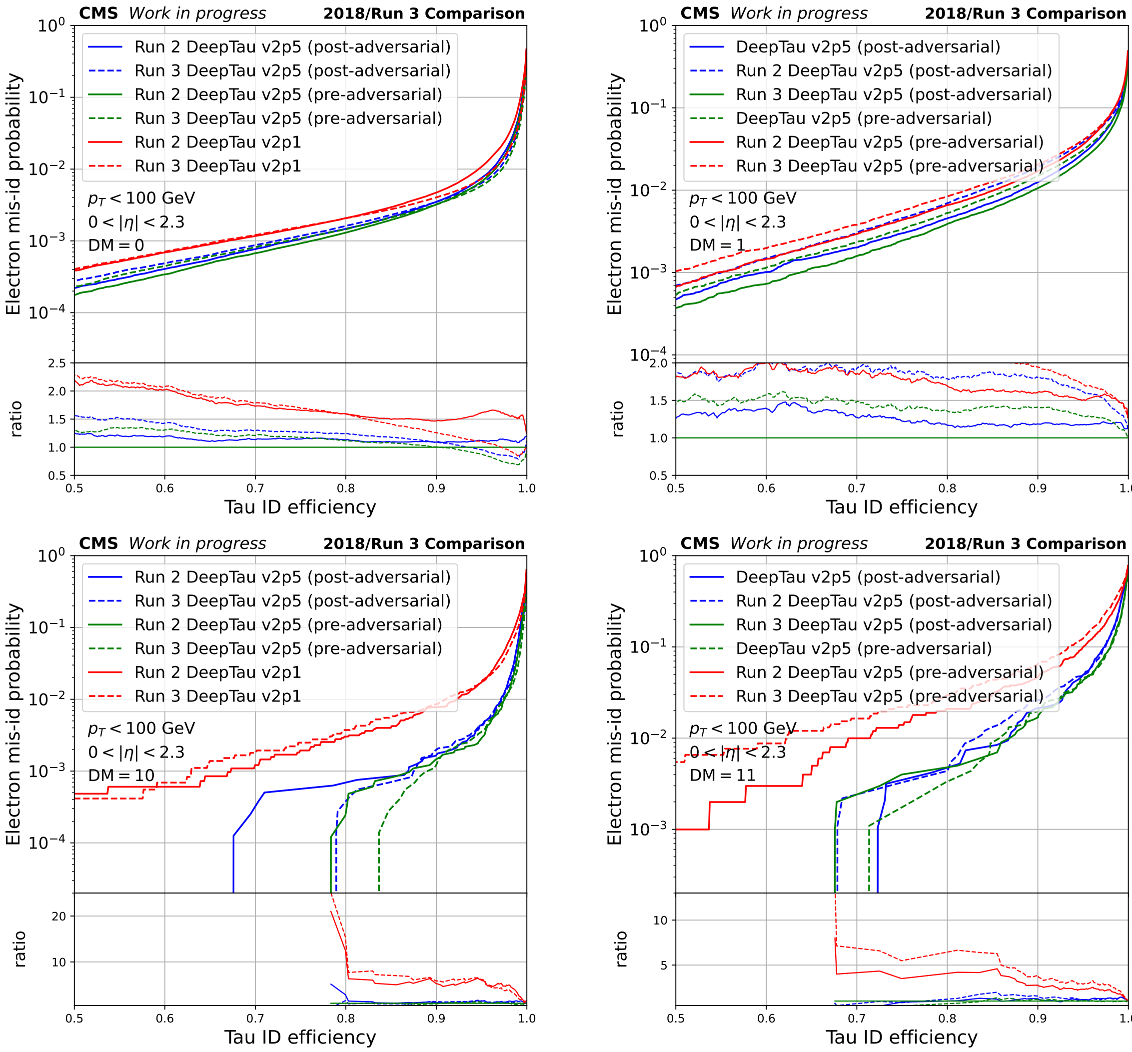


Figure A.10: Comparison of Run 2/3 individual DM identification performance against electrons in the low p_T central region.

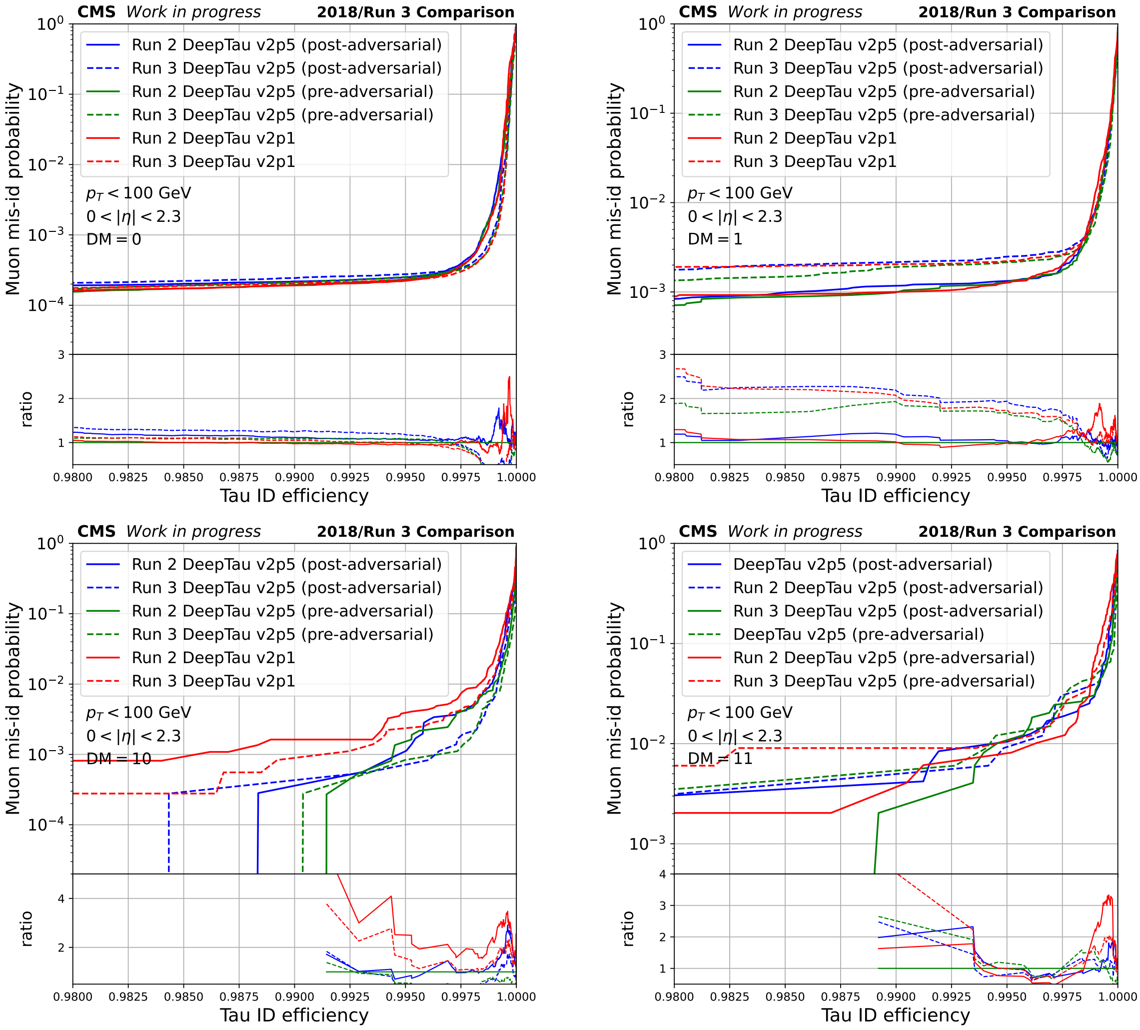


Figure A.11: Comparison of Run 2/3 individual DM identification performance against muons in the low p_T central region.

A.2.3 Forward Region by DM

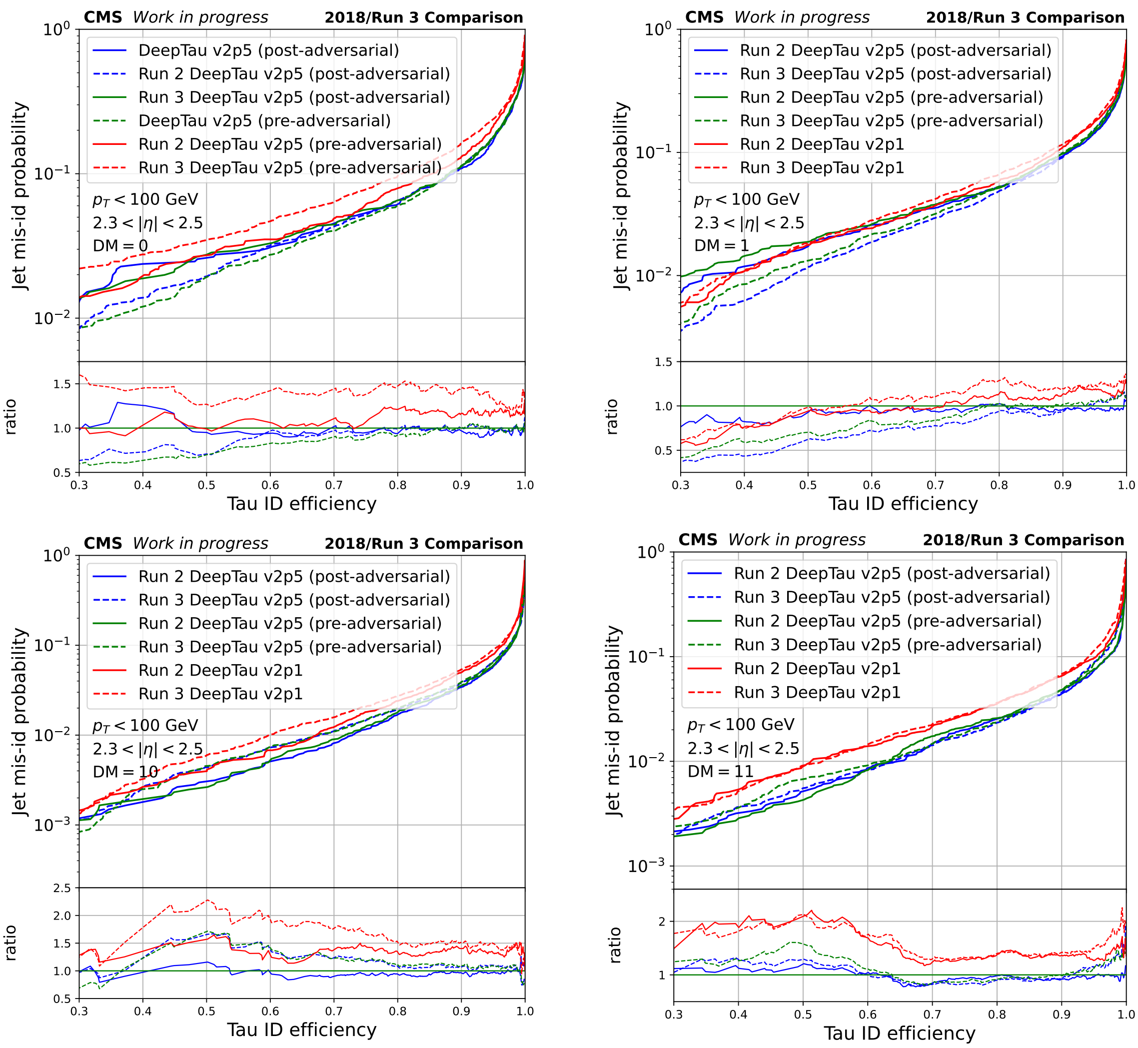


Figure A.12: Comparison of Run 2/3 individual DM identification performance against jets in the low p_T forward region.

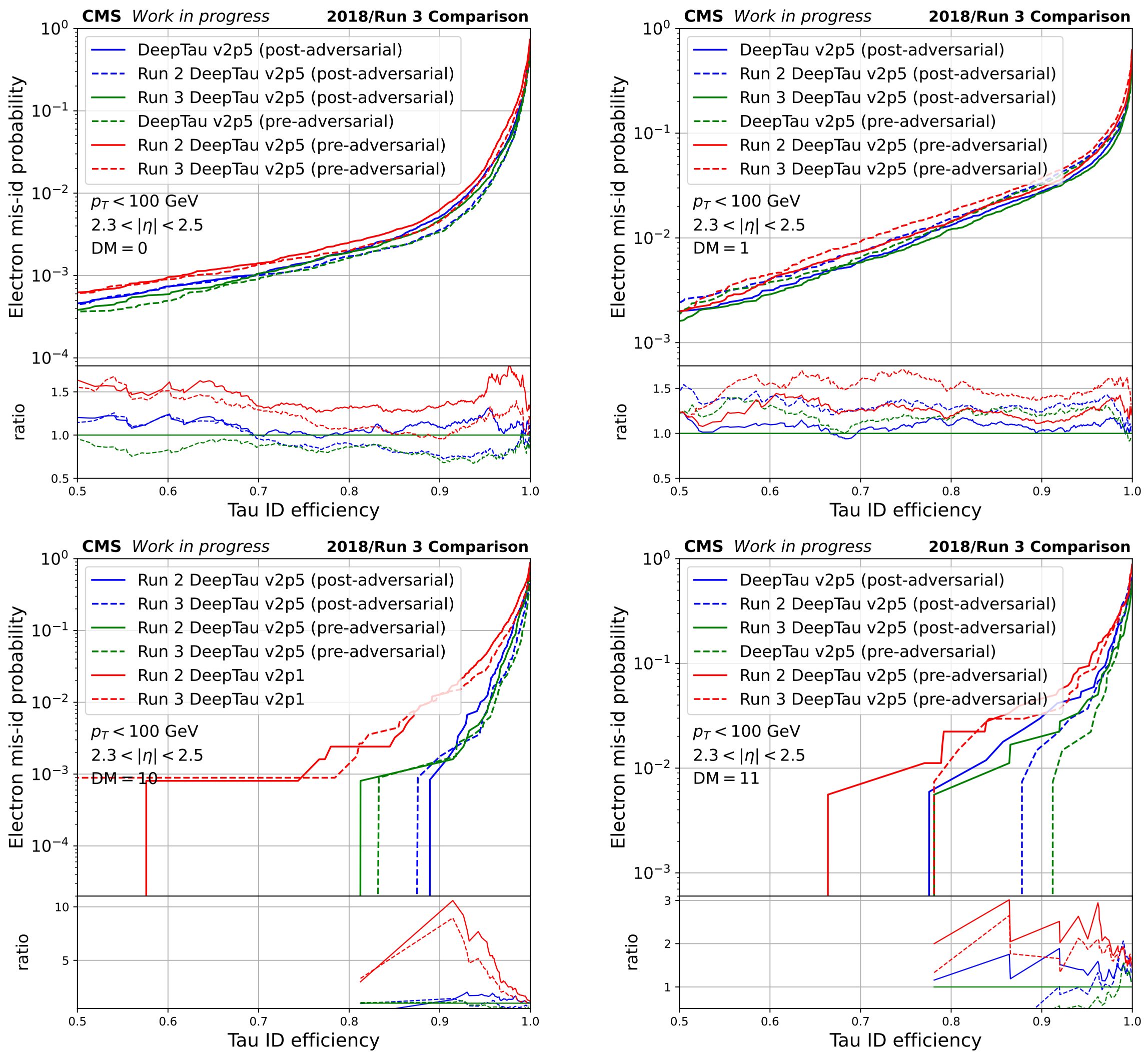


Figure A.13: Comparison of Run 2/3 individual DM identification performance against electrons in the low p_T forward region.

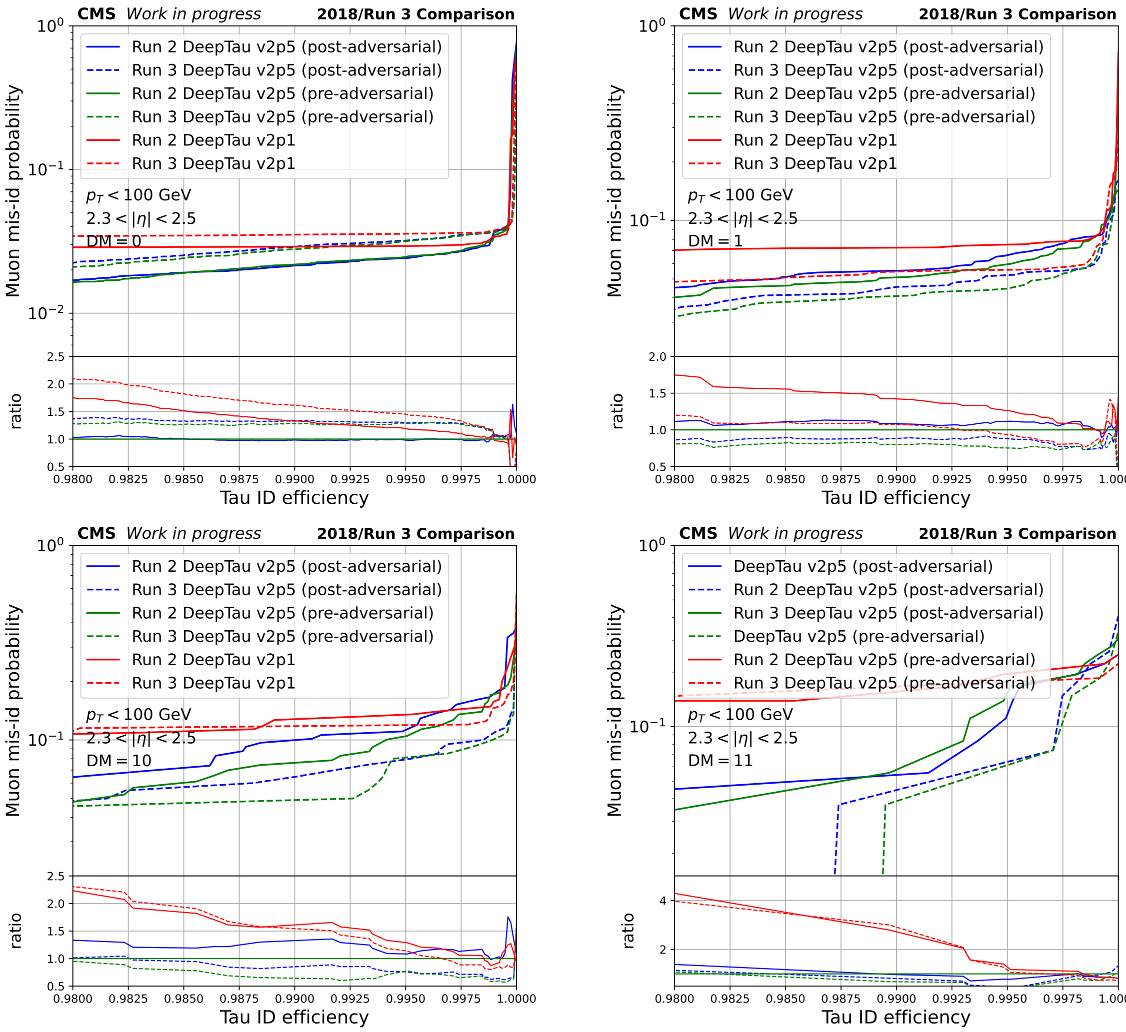


Figure A.14: Comparison of Run 2/3 individual DM identification performance against muons in the low p_T forward region.