DIPLOMARBEIT

DeepLepton: Muon identification in 13 TeV pp collisions at the CMS experiment using deep learning techniques

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Physik

eingereicht von

Georg Mörtl, BSc
Matr.Nr.: 01326471

Ausgeführt am Atominstitut
der Fakultät für Physik der Technischen Universität Wien
in Zusammenarbeit mit dem Institut für Hochenergiephysik (HEPHY)
der Österreichischen Akademie der Wissenschaften

Betreuung
Betreuer: Univ.Prof. Dr. **Jochen Schieck**
Mitwirkung: Dr. **Robert Schöfbeck**

Wien, 03.04.2019

(Unterschrift Verfasser) (Unterschrift Betreuer)

Kurzfassung

Myonen und Elektronen welche in Teilchenkollisionen im Zentrum des Compact Muon Solenoid (CMS) Experiments am Large Hadron Collider (LHC) entstehen, werden aufgrund ihrer Signale im Detektor rekonstruiert. Die Bestimmung des Ursprungs eines Leptons, gemäß seiner Herkunft im Streuprozess, ist wesentlich für viele Analysen. Dazu erfolgt eine Trennung in prompte, nicht-prompte und misrekonstruierte Leptonen, was auf eine Klassifikationsaufgabe mit mehreren Klassen führt.

In dieser Arbeit wird ein neuartiger Ansatz für die Identifizierung von Myonen in den Daten des CMS Detektors aus dem Jahr 2016 präsentiert, wo die Proton-Proton Kollisionen bei einer Schwerpunktsenergie von $\sqrt{s} = 13$ TeV durchgeführt wurden. Die Kombination von relevanten Observablen aus der Rekonstruktion der Leptonen mit der weiteren Teilchenaktivität in der Umgebung des Leptons ist dabei die zentrale Idee. Das soll zu einer höheren Effizienz bei der Identifizierung führen, da die umgebenden Teilchen Aufschluss über den Produktionsmechanismus des Leptons geben. Dieser Ansatz erfordert es, die Teilchen in der Umgebung eines Leptons der Reihe nach zu verarbeiten.

Die Anwendung der Technik des überwachten Lernens und der Einsatz eines tiefen neuronalen Netzwerks, welches aus konvolutionalen Schichten, Long Short Term Memory (LSTM) Einheiten und vollständig verbundenen neuronalen Netzen besteht, wird den Anforderungen eines Klassifikationsproblems mit mehreren Klassen gerecht, und ermöglicht eine Verarbeitung von sequentiellen Daten.

Im Vergleich mit anderen Ansätzen bewirkt die Einbindung der Umgebung der Leptonen auf einer niedrigeren Ebene der Rekonstruktion einen signifikanten Leistungszuwachs hinsichtlich der Effizienz bei der Identifizierung von Myonen. In simulierten Daten beläuft sich die Steigerung der Signaleffizienz auf bis zu 5% für gegebene Hintergrundeffizienz. Vergleiche von simulierten und realen Daten unterstreichen die Robustheit dieses neuartigen Identifikationskriteriums. Der neue *DeepLepton* Algorithmus bildet damit die Grundlage für Datenanalysen mit starkem Hintergrund von nicht-prompten und misrekonstruierten Leptonen.

Abstract

Muons and electrons emerging in particle collisions in the center of the CMS experiment at the LHC, are reconstructed on the basis of the measured detector hits. The identification of the origin of a lepton, according to its source from the hard scatter, is a central necessity in many analyses. To this end, a separation of prompt, non-prompt and fake leptons is made, which leads to a multiclass classification task.

In this work, a novel approach for muon identification is presented for the 2016 detector data, taken by the CMS experiment, where proton-proton collisions were performed at a center-of-mass energy of $\sqrt{s} = 13$ TeV. The central idea is to combine characteristic lepton reconstruction observables with the particle activity in the vicinity of the lepton. This is supposed to lead to a higher identification efficiency, since the surrounding particles give indication on the production mechanism of the lepton. The approach requires to process all the information from the vicinity of a lepton, particle by particle.

The application of the supervised learning technique and the usage of a deep neural network, composed of convolutional layers, Long Short Term Memory (LSTM) units and fully connected (FC) neural networks, meets the requirements of the multiclass classification problem, and enables the processing of the sequential data.

In terms of muons, the inclusion of the low-level information of the lepton vicinity effects a significant performance gain in the identification efficiency when compared to previous approaches. In simulated data the improvement in signal efficiency amounts up to 5% for a given background efficiency. Comparisons of simulated and real data show the validity of this novel muon identification criterion. It is the founding stone of *DeepLepton*, a lepton identification algorithm, to be applied in analyses where backgrounds from non-prompt and fake leptons are a substantial problem.

Contents

Chapter 1

Introduction

1.1 Problem statement and motivation

In various fields of research, scientific measurements do not yield plain and simple results. Although accurate theory predictions are typically available in the field of particle physics, there are often noise or background contributions, which interfere with the signal.

The quality and validity of experimental analyses at the LHC [1] hinges on a proper event selection. An analysis, that targets a certain process, will only be convincing, if the selected events are dominated by the considered process. Especially when processes with high and/or complicated backgrounds are addressed, high requirements are put on the efficiency of the event selection. The efficiency of the event selection, in turn, depends on the accuracy of the reconstruction and the identification quality of the particles. At the CMS experiment [2], electrons, muons, charged and neutral hadrons, as well as photons are reconstructed on the basis of the measured detector hits. Various identification methods are utilized to subsequently classify the reconstructed particles.

In terms of lepton identification, a classification of electrons and muons into three categories, according to their origin from the hard scatter within an event, is made. In most analysis use cases, prompt leptons are of interest. These leptons arise from the decay of heavy particles, which are produced in the proton-proton collision, e.g. the W, Z, or Higgs bosons. Non-prompt leptons, emerging in the hadronization of high energetic jets, and fake leptons, which are particles that are misreconstructed as leptons, are usually considered as background. Various approaches have been developed in order to achieve a proper lepton identification, with varying degrees of success. These range from simple cut-based algorithms up to advanced machine learning techniques.

Nevertheless, there is still demand for further improvements, since the requirements concerning lepton identification increase for analyses of rare signal processes or events with complex backgrounds. Thus, besides the general ambition of improving existing identification techniques, the motivation of this master's thesis is driven by specific analysis use cases with difficult backgrounds, like in e.g. searches for processes beyond the standard model of particle physics (SM) [3–5].

1.2 Aim of the work

The aim of this master thesis is to provide a novel moun identification criterion (ID) for the 2016 detector data taken by the CMS experiment at the LHC, and outperform existing approaches. It should be the founding stone of a lepton identification tool, to be applied in analyses, where backgrounds from non-prompt and fake leptons are a substantial problem. In particular, the event selection efficiency of the $t\bar{t}$ dilepton final state is intended to be enhanced for the same level of fake and non-prompt backgrounds.

1.3 Methodological approach

The central idea is to combine characteristic lepton reconstruction observables with the particle activity in the vicinity of the lepton. This is supposed to lead to a higher identification efficiency, since the surrounding particles give indication on the production mechanism of the lepton. This approach requires to process all the information from the vicinity of a lepton, particle by particle. A tool that has already solved this technical issue, is the so called *DeepJet* framework [6]. It is designed for jet identification and is based on a deep neural network, utilizing advanced machine learning techniques. The computation of the muon ID, addressed in this master thesis, is highly inspired by algorithms provided by the *DeepJet* framework and therefore, it is referred to as *DeepLepton* ID in the following.

Since lepton identification is a typical classification task, the supervised learning technique is considered in this case. Simulated data, produced by MC generators, is applied for the training of the *DeepLepton* classifier, as this data provides the information of the origin of the lepton, which is essential in terms of supervised learning. The performance, of the *DeepLepton* classifier in the simulated data, is evaluated by means of Receiver Operating Characteristic (ROC) curves. Two existing machine learning based lepton IDs are used as a reference. Comparisons of simulated and real data show the validity of the novel muon ID.

1.4 Structure of the work

The master thesis consists of two main parts, a theoretical and an experimental part.

The theoretical part starts with an introduction into the physics at the LHC (Ch. 2) and the related CMS experiment (Ch. 3). Then, the processes contributing to the dilepton final state and relevant state-of-the-art methods in lepton identification at the CMS experiment are illustrated (Ch. 4). Finally, basic concepts in deep learning and its applications in particle physics are introduced (Ch. 5).

The experimental part, treats the practical implementation of the work, including the data preprocessing, the training and the evaluation procedures of the *DeepLepton* classifier (Ch. 6). The results of the application of the *DeepLepton* framework are presented in Ch. 7, followed by a more detailed discussion and an outlook in prospective follow-up analyses in Ch. 8. A conclusion is provided in Ch. 9.

PART I: Theory

Chapter 2

The Large Hadron Collider (LHC)

2.1 Introduction and main goals of the LHC

The Large Hadron Collider (LHC) is the largest and most powerful particle accelerator so far. It accelerates particles (protons and lead ions) up to collisions at a center-of-mass energy of 13 TeV. The LHC explores the nature at the small distances of elementary particles and forces, and enhances the knowledge about the fundamental constituents of matter. In Run 1 (8 TeV proton-proton collisions), the discovery of the SM Higgs boson in 2012 [7] was a landmark in the history of particle physics. It completed the particle zoo of the SM. The Run 2, which was finished in 2018, enabled the rediscovery of the Higgs boson and a more detailed study of other SM parameters, because a higher center-of-mass energy of $\sqrt{s} = 13$ TeV could be reached. This higher energy also increased the discovery potential. However, so far only tighter constraints for physical models beyond the SM were obtained [8].

The SM is a complete and self-consistent theory. However, it does not answer all fundamental physics questions, based on the observations. The main questions an searches, which are addressed at the LHC, are outlined in the following:

- Supersymmetry (SUSY): SUSY [9] is a popular SM extension, since it solves several theoretical problems. As an example, it can implement the grand unification of the three fundamental forces, and therefore is a grand unified theory (GUT) [10]. Furthermore, it can provide dark matter particle candidates. There are significant theoretical and experimental efforts in the field of supersymmetric models and SUSY analyses make up a major sector within the experiments at the LHC. However, so far no experimental evidence of SUSY has been found [8].

- Dark matter and dark energy: The known matter represents just around 5% of the total matter of the whole universe [11]. The remaining 95% are not interacting with ordinary matter and are referred to as dark matter and dark energy. In ongoing searches at the LHC, scientists try to find particles, like e.g. weakly interacting and massive particles (WIMPs) [12], or other phenomena that could explain dark matter [8, 13].

- The origin of mass: With the discovery of the Higgs boson at the LHC, the Brout-Englert-Higgs mechanism [14] was confirmed. This mechanism explains, the interaction of SM particles with the Higgs field, a doublet of complex scalar fields. The particles receive their masses, by means of their interaction with the Higgs particle. Particles, interacting more strongly with the Higgs field, are heavier, than weakly interacting particles. Measurements in the Higgs sector are now focusing on the properties of the Higgs boson, as well as examining its decay modes. Especially non-SM decays, or even searches for additional neutral and charged Higgs bosons, are of interest in terms of physics beyond the SM [8].

Several so called beyond standard model (BSM) searches in the Higgs sector are illustrated in Fig. 2.1.

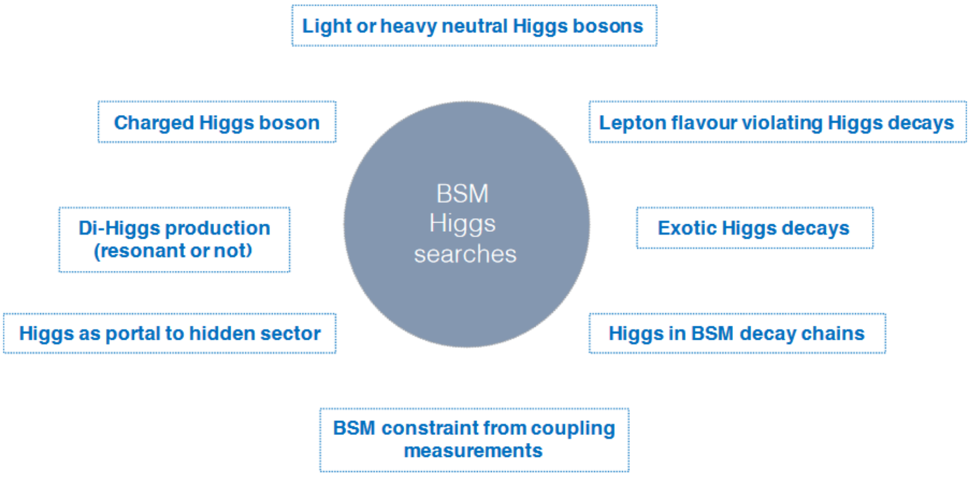

Light or heavy neutral Higgs bosons
Charged Higgs boson
Lepton flavour violating Higgs decays
BSM Higgs searches
Di-Higgs production (resonant or not)
Exotic Higgs decays
Higgs as portal to hidden sector
Higgs in BSM decay chains
BSM constraint from coupling measurements

Fig. 2.1: Searches for BSM Higgs bosons [8].

- The matter-antimatter asymmetry: According to the cosmological standard model, the distribution of matter and antimatter was uniform in the early universe following the Big Bang [15]. However nowadays, matter particles such as protons, neutrons and electrons, are observed. The baryon asymmetry is expected to be almost 100%. Charge and parity (CP) violating processes could explain that observation and are subject to tests at the LHC [16]. It is worth noting, that the ratio of protons and antiprotons produced in heavy ion collisions at the LHC is close to 1 [17, 18].
- The quark-gluon plasma: A state, that appears at very high temperatures and densities, conditions which may have prevailed e.g. at the beginning of the universe, is the quark-gluon plasma [19]. It is considered as a new state of nuclear matter, where quarks and gluons are not confined in composite states of hadrons, like in e.g. protons, neutrons, kaons, or pions. A quark-gluon plasma is produced in collisions of heavy ions. These measurements help to better understand the early universe and collective phenomena in quantum chromodynamics (QCD).

2.2 Accelerator complex at CERN

The LHC is the most recent accelerator at the CERN complex (see Fig. 2.2) [20]. The acceleration process starts with a gas cylinder filled with compressed hydrogen. Hydrogen atoms are fed into the front chamber of a linear accelerator, the LINAC 2, where the electrons are stripped off [21, 22]. The remaining ionized hydrogen atoms, i.e. protons, are accelerated by an electric field in drift tubes, due to the positive charge $q = +e$. The bunch of protons leaves the LINAC 2 at a velocity of $\beta = \frac{v}{c} \approx 0.3$, relative to the speed of light c. The kinetic proton energy is $K_{\text{LINAC2}} =$ 50 MeV.

The second stage is the Proton Synchrotron Booster [20, 22]. The booster is circular with a circumference of 157 m, as linear acceleration is now impractical. The protons are split into four beams in order to increase the beam intensity. The protons are accelerated in separate synchrotron rings up to a kinetic energy $K_{\text{booster}} = 1.4$ GeV and a velocity of around 91.6%

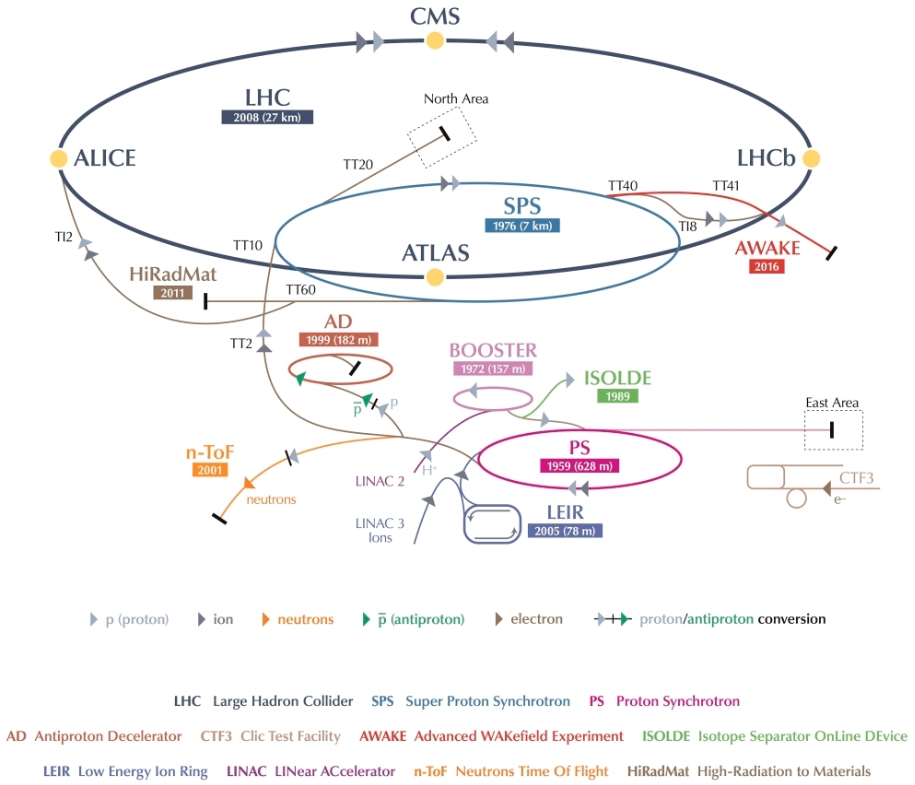


Fig. 2.2: The accelerator complex at the LHC [20].

of the speed of light. This is achieved by radio frequency (RF) cavity resonators. A certain resonance frequency, is amplified inside the RF cavities and forms a strong electric field. Since the electric field is oscillating, the transition of the proton bunches has to be synchronized, in order to achieve an accelerating effect. In addition to the alternating electric field, a homogeneous magnetic field $\vec{B}$ is forcing and holding the protons on their circular path through the booster, due to the Lorentz force. It is generated by powerful dipole magnets, orientated at right angles to the beam direction. Both the homogeneous and the alternating field have to be enhanced per circulation, synchronously according to the proton momentum $\vec{p}$. The following relation has to be fulfilled for the homogeneous magnetic field, where r refers to the radius of the synchrotron.

$$|\vec{p}| = q \cdot r \cdot |\vec{B}| \tag{2.1}$$

Thus, the maximum output energy of the protons is limited by the radius and the maximum magnetic field of the booster [23]. After re-merging the four proton packs, the beam is sent to the Proton Synchrotron (PS) [22], which has a circumference of 628 m. At this stage the protons reach an energy of $K_{\text{PS}} = 25$ GeV ($\beta > 0.999$), due the (oscillating) electric field in the RF cavity resonators. In terms of the beam guidance, again conventional (room-temperature) dipole magnets are applied. Furthermore, quadrupole magnets are utilized to focus the proton beam. The fourth stage in the accelerator process is the Super Proton Synchrotron (SPS) [24]. It has a circumference of approximately 7 km and further increases the energy of the protons up to $K_{SPS} = 450$ GeV, before the beam is finally sent to the Large Hadron Collider (LHC) with a velocity of 99,9998% of the speed of light [20].

2.3 LHC facts and figures

With a circumference of nearly 27 km, the LHC is the largest particle accelerator built so far [20]. It consists of two separate pipes, where the proton bunches travel in opposite directions. An ultra-high vacuum is maintained inside the tubes. The protons are bent on their way through the separate collider pipes due to a strong magnetic field $\vec{B}$, generated by 1232 main dipole magnets. In order to achieve $|\vec{B}| = 8.3$ Tesla, superconducting magnets are needed. Below a certain temperature, these magnets enter the superconducting state, which is characterized by a vanishing electrical resistance. Thus, the dipole magnets at the LHC are operated at a temperature of 1.9 K (-271.3 °C), unlike the conventional dipole magnets, operating at room-temperature. The focusing of the proton bunches is accomplished by 858 quadrupole magnets.

At the LHC, the protons are accelerated for around 20 minutes until the final energy of $K_{\mathrm{LHC}} = 6.5$ TeV is reached [20]. One beam is accelerated clockwise and the other one anti-clockwise, each by eight RF cavities. The operating frequency of the cavities is at 400 MHz, which generate an accelerating field of 5 MV/m. The beams are collided at four main experimental sites, where the experiments are hosted: CMS [25], ATLAS [26], ALICE [27] and LHCb [28]. In terms of a collision experiment, the maximum center-of-mass energy $\sqrt{s}$, that can be achieved, is proportional to the energy of the colliding particles [29].

$$\sqrt{s_{collision}} = \sqrt{\left(\begin{pmatrix} E_{beam} \\ p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} E_{beam} \\ -p_x \\ -p_y \\ -p_z \end{pmatrix}\right)^2} = 2 \cdot E_{beam} \tag{2.2}$$

$$\tag{2.3}$$

For the proton-proton collisions at the LHC this results in a center-of-mass energy of $\sqrt{s} = 13$ TeV. It is worth to note that the full center-of-mass energy of 13 TeV is not coming into effect in a fundamental particle reaction. According to the parton-model [30], not the protons, but its constituents, the quarks and gluons, are reacting. Hence, the actual center-of-mass energy of the parton collisions $\sqrt{\hat{s}}$ is dependent on the fractions of the protons momenta x_1 and x_2, that the respective partons are carrying [8].

$$\sqrt{\hat{s}_{collision}} = \sqrt{x_1 \cdot x_2 \cdot s_{collision}} \tag{2.4}$$

However, x_1 and x_2, and therefore the initial longitudinal momenta, are unknown at the parton-parton collisions, while the initial transverse momenta cancel each other [29]. Hence, the sum of the transverse momenta p_{T} of the final state particles is expected to be zero too.

Besides the energy, the rate of collisions dN/dt is another important parameter of a particle collider [31]. Physical processes have production rates $dN_{process}/dt$ proportional to their cross section $\sigma_{process}$. The cross section has the dimension of an area and is usually given in barn ($1\,\mathrm{b} = 10^{-28}\,\mathrm{m}^2$). The cross sections of various SM processes, as a function of the center-of-mass energy of colliding proton-proton beams, are shown in Fig. 2.3.

The parameter that combines the production rate of a certain process with its cross section, is the luminosity $\mathcal{L}$ [31].

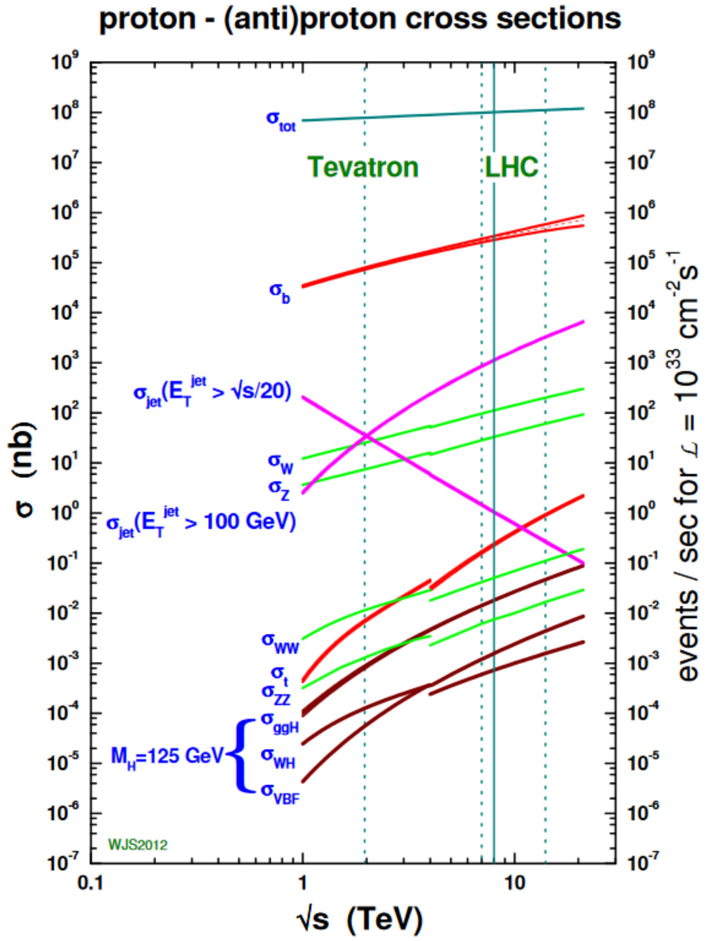


Fig. 2.3: Cross sections of SM processes dependent on the center-of-mass energy of proton-(anti)proton collisions [32].

$$\frac{dN_{process}}{dt} = \mathcal{L} \cdot \sigma_{process} \tag{2.5}$$

At the LHC, a beam contains $n_b = 2808$ bunches, with around $N_p \approx 1.15 \cdot 10^{11}$ protons per bunch. The distance in time between two bunches is 25 ns, which yields a bunch crossing rate of 40 MHz. A theoretical maximum number of bunches of 3564 can not be achieved, due to the need for an abort gap for the beam dump. The distance in time corresponds to a path difference of around 7.5 m and a revolution frequency of $f_{rev} = 11245\,\mathrm{Hz}$ [20]. This yields a luminosity $\mathcal{L}_{\mathrm{LHC}}$ at the order of $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, which is calculated by

$$\mathcal{L}_{\mathrm{LHC}} = \frac{f_{rev} n_b N_p^2}{4\pi\sigma_x\sigma_y} \cdot F(\theta_c, \sigma_x, \sigma_z) \approx 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1} = 10\,\mathrm{nb}^{-1}\,\mathrm{s}^{-1}. \tag{2.6}$$

In this case σ_i describes the beam width along the respective axis $i = x, y, z$ and the correction factor F takes into account, that the luminosity is decreased, dependent on a beam crossing angle θ_c [8]. The highest instantaneous luminosity was $2.06 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ by now, which could be reached in 2017 [20]. With a total cross section in the proton-proton collisions of about $\sigma_{pp\to x} \approx 10^8\,\mathrm{nb}$, an average collision rate dN_{total}/dt of around 1 billion events per second follows.

$$\frac{dN_{total}}{dt} = \mathcal{L}_{LHC} \cdot \sigma_{pp\to x} \approx 10\,\mathrm{nb}^{-1}\,\mathrm{s}^{-1} \cdot 10^8\,\mathrm{nb} = 10^9\,\mathrm{s}^{-1} \tag{2.7}$$

The average number of interactions per bunch crossing $\langle\mu\rangle$ is determined, by means of the beam revolution frequency f_{rev} and number of proton bunches per beam n_b [8].

$$\langle\mu\rangle = \frac{dN_{total}}{dt} \cdot \frac{1}{f_{\mathrm{rev}} \cdot n_b} = \frac{\mathcal{L}_{LHC} \cdot \sigma_{pp\to x}}{f_{\mathrm{rev}} \cdot n_b} \tag{2.8}$$

In 2018, on average $\langle\mu\rangle = 39$ interactions per bunch crossing were observed in the proton-proton collisions at the CMS experiment [33].

The reason for the usage of protons within the acceleration procedure is, that they fulfill two main criteria, which possible particle candidates should feature. Firstly they are charged, which enables the acceleration by an electric field, and secondly they are stable. Furthermore, a higher mass is an advantage against electrons, in order to realize high-energy particle collisions. In addition to protons, lead ions can be accelerated at the LHC too [8]. Lead-lead collisions reach a center-of-mass energy up to $\sqrt{s} = 5$ TeV. Proton collisions are usually measured at the ATLAS, CMS and LHCb experiments, whereas ALICE is specialized for the heavy ion collisions.

The ”A Toroidal LHC Apparatus” (ATLAS) experiment is constructed for a wide range of analyses in particle physics, from a precise characterization of known particles like the Higgs boson, to searches for new physics beyond the SM. The core of the detector is built of eight superconducting magnets, which form a huge toroidal magnet system around the beam pipes. It has a diameter of 26 m and is the largest collider detector in terms of volume so far. The physical measurements performed at the Compact Muon Solenoid (CMS) experiment are similar to the ones at ATLAS. However, the detector and the magnet system are designed a bit different. An introduction into the CMS experiment and its detector system is given in Ch. 3. As already mentioned, ”A Large Ion Collider Experiment” (ALICE) is specialized in lead ion collisions. Here, the quark-gluon plasma is studied, where the quarks and gluons are not bound inside hadrons any more. At the ”Large Hadron Collider - beauty” (LHCb) experiment the imbalance between matter and antimatter is examined. Interactions of hadrons containing a bottom quark, like B-mesons, show a slight asymmetry referring to this issue.

Besides this four main measurement sites at the beam crossing points another three, but smaller experiments exist at the LHC. They are named as ”Large Hadron Collider forward” (LHCf) [34], ”TOTal Elastic and diffractive cross section Measurement” (TOTEM) [35] and ”Monopole and Exotics Detector at the LHC” (MoEDAL) [36].

Chapter 3

The Compact Muon Solenoid (CMS) experiment at the LHC

3.1 Introduction and main goals of the CMS experiment

The Compact Muon Solenoid (CMS) [2, 37, 38] is one of the four main experiments at the LHC. The CMS detector is a general-purpose apparatus, constructed to measure phenomena in proton-proton collisions provided by the LHC. The heart of the detector is its high magnetic field, generated by a powerful superconducting solenoid. A high magnetic field is required for the precise measurement of the momenta of charged particles, e.g. muons.

The high instantaneous luminosity up to $\mathcal{L} \approx 2 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in 2017, lead to on average 37 interactions per bunch crossing (every 25 ns) in 2017. Each interaction can produce thousands of particles in the detector. This poses challenges in terms of triggering and recording events. In 2017, a total integrated luminosity

$$L = \int \mathcal{L} \cdot dt \tag{3.1}$$

of $L_{2017}^{\mathrm{CMS}} = 44.98\,\mathrm{fb}^{-1}$ was recorded out of a delivered luminosity of $L_{2017}^{\mathrm{LHC}} = 49.79\,\mathrm{fb}^{-1}$.

The coordinate system used in the CMS experiment has its origin at the nominal collision point in the center of the detector, where the z axis points along the beam path (Fig. 3.1). The y axis is pointing in the vertical direction and the x axis runs alongside an imaginary line between the collision point at the CMS experiment and the center of the LHC. The azimuth angle ϕ is defined by the angular deviation from the x axis in the x-y plane. Thus, the polar angle θ indicates the angular deviation from the z axis. Instead of θ, the pseudorapidity

$$\eta \equiv -\ln(\tan(\theta/2)) \tag{3.2}$$

is used. Its key advantage is the Lorentz invariance of $\Delta\eta$ along the z axis, if the masses of the particles are negligible. Hence, the angular separation $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ of two tracks in the detector is Lorentz invariant too, as $\Delta\phi$ is the azimuthal angle in the x-y plane.

The CMS detector is designed to study various physical phenomena in proton-proton and lead-lead ion collisions. In the following, a selection of current goals and experimental analyses at the CMS experiment, is given [33]:

- Higgs boson physics: As already mentioned, the discovery of the Higgs boson [7], jointly announced by the CMS and ATLAS experiments in 2012, was a landmark in particle

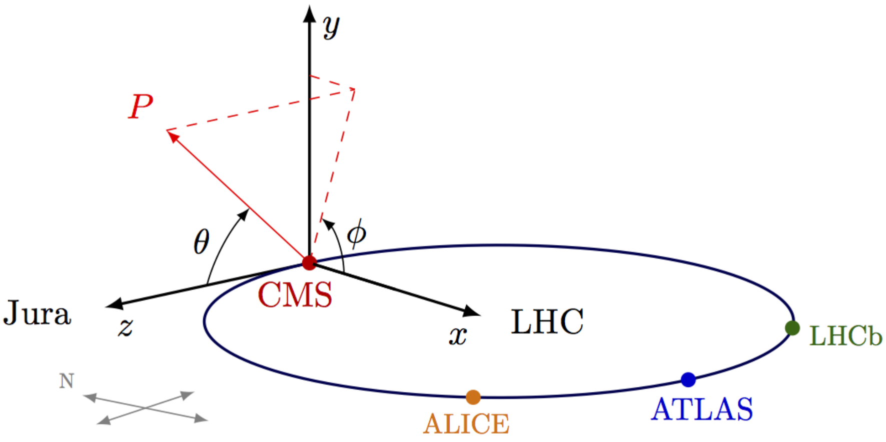


Fig. 3.1: The CMS coordinate system [39].

physics. Ongoing measurements are now focusing the properties of the Higgs boson and its couplings to the other particles. E.g. the observation of the $t\bar{t}$+Higgs process [40] enables a direct measurement of the Yukawa coupling of the Higgs boson to the top quark [41]. An inconsistency with the SM would indicate BSM physics. However, the measured cross section of this process agrees with the prediction of the SM. Another search for the Higgs boson pair production [42] is performed at the CMS experiment, in order to examine the self-coupling of the Higgs boson. In fact, the predicted SM cross section of diHiggs production is too low to be detected, but various BSM models predict higher cross sections of this process [42]. So far, no evidence for such events has been found, and constraints on the upper limit of the cross section could be determined.

- Probing the SM: At the CMS experiment, several analyses are testing the consistency of the data with the SM. This includes precision measurements of the cross sections of SM processes, like e.g. the WZ diboson production in leptonic channel with three final state leptons [43], or the W+c production in the W → $\mu\nu$ channel [44], and single (anti)-top events [45]. So far, all results are compatible with the predictions of the SM.
- BSM physics: Several fields of physics beyond the SM are in focus at the CMS experiment: like Supersymmetry, other exotic BSM models, as well as dark matter, and emerging jets. In terms of dark matter, especially final states with a large missing transverse energy are of interest, e.g. the mono-photon final state [46] and the mono-Z final state [47]. Furthermore, emerging jets [48] could indicate dark matter particles, predicted by models containing dark baryons [49, 50]. No significant evidence for new physics has been found yet.

3.2 Detector system

The CMS detector is located around 100 m underground with a total weight of over 12,500 tons. It has a diameter of 14.6 m and a length of 21.6 m [37]. The detector is built of several sub-detector layers around a powerful superconducting solenoid. It is separated into a barrel and an endcap region, referring to the lateral and top surface of the cylindrical detector geometry. An illustration of a slice in the barrel region of the detector is given in Fig. 3.2.

In the following the main parts of the CMS detector are outlined. Further details can be found in [2, 37, 38].

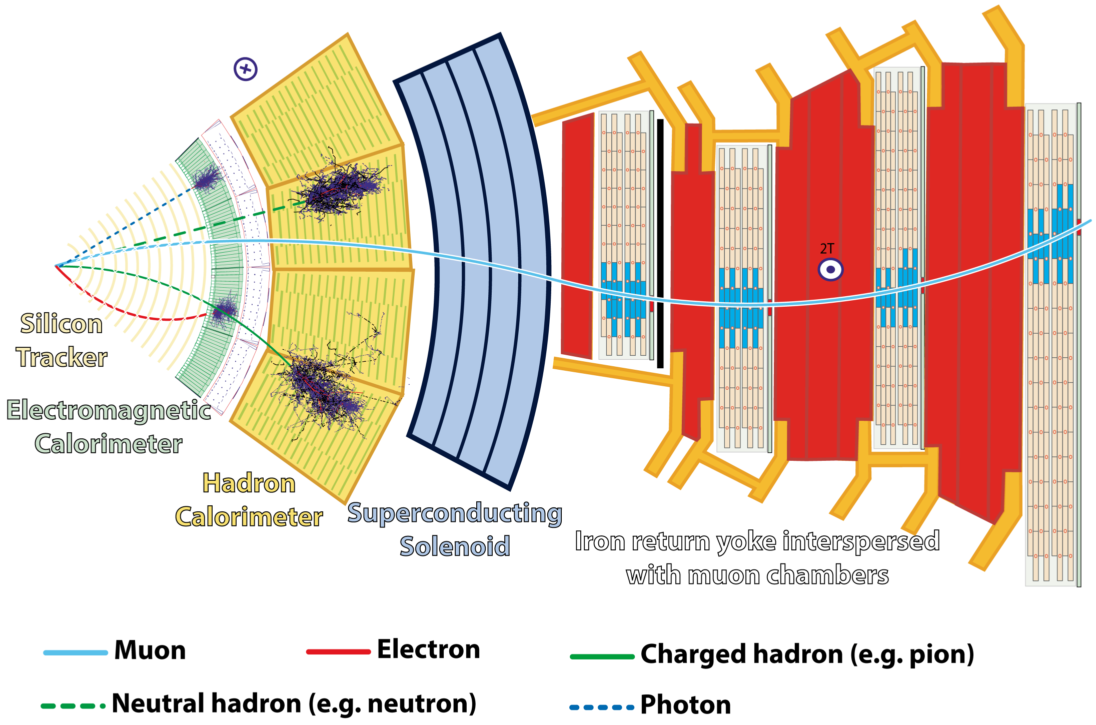

Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with muon chambers
2T
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

Fig. 3.2: A schematic slice of the CMS detector [51]. Charged particles, like electrons, muons and charged hadrons cause hits in the inner tracker. Muons leave an additional track signature in the muon chambers. In contrast, neutral hadrons and photons, do not have a track. Electrons and photons are showering in the ECAL, while neutral and charged hadrons can be detected in the HCAL.

3.2.1 Superconducting solenoid

The superconducting solenoid is designed to provide a magnetic field B_{coil} of 4 T. This is achieved by a winding in four layers, with a number of turns $N = 2168$ over a length of $L = 12.5$ m and an inner diameter of 6.3 m. The cable is made of a niobium-titanium alloy, within an aluminum matrix. The so-called Rutherford type cable [52] carries a current of around $I = 20$ kA and in total an energy of 2.6 GJ is stored at full current. The magnet is cooled to 4.45 K by a He refrigeration plant, to ensure its superconducting properties. The magnetic field inside the cylindrical coil is given by

$$B_{coil} = \mu_0 \cdot \mu_r \cdot \frac{N \cdot I}{L} \approx 4\,\mathrm{T}, \tag{3.3}$$

where μ_0 refers to the magnetic field constant

$$\mu_0 = 1,257 \cdot 10^{-6} \frac{\mathrm{V} \cdot \mathrm{s}}{\mathrm{A} \cdot \mathrm{m}} \tag{3.4}$$

and μ_r is relative permeability, which is equal to 1 in the vacuum.

The purpose of the solenoid is to bend the path of charged particles in the detector. Due to the applied magnetic field, the resulting Lorentz force

$$\vec{F}_L = q\vec{v} \times \vec{B} \tag{3.5}$$

affects the trajectory of the charged particles. Oppositely charged particles are bent in opposite directions and particles at high momentum are bent less than the ones with a low-momentum. This enables a determination of the charge q of the particles, as well as a precise measurement of their momenta ($\vec{p} = m\vec{v}$), based on the reconstructed tracks.

3.2.2 Inner tracking system

The function of the inner tracker is to precisely determine the trajectories of the charged particles and to identify the primary vertex (PV) of the collision and possible secondary vertices (SVs), due to in-flight decays of short lived particles, like e.g. B mesons. It is the innermost layer of the detector, built around the nominal collision point with a length of 5.8 m and a radius of $r_{\text{tracker}} = 1.25$ m.

This subdetector comprises silicon pixel and strip detectors. It is divided in three segments, for the barrel and the endcap regions. In the barrel, three layers of silicon pixel detectors, with a size of 100×150 µm^2 are located near the PV at the radii of r_{tracker} =4.4 cm, 7.3 cm and 10.2 cm. Next, silicon microstrip detectors with the dimensions of 10 cm × 80 µm are placed in the middle region (22 cm $< r_{\text{tracker}} <$ 55 cm). Finally, silicon microstrip detectors with a size of 25 cm × 180 µm form the outer region of the tracker system up to a radius of $r_{\text{tracker}} = 1.1$ m. In the forward region, there are two layers of pixel detectors, followed by 3 disks of silicon microstrips, in the transitional region between the barrel and the endcap, and lastly, nine layers of microstrip detectors in the endcap, on both sides.

Charged particles electromagnetically interact with the silicon on their path through the inner tracker. The small electric signal produced by the interaction, is amplified and constitutes a so-called hit. Then, the particle tracks of electrons, high-energy muons and charged hadrons are reconstructed from the measured hits. Furthermore the momenta of these particles can be determined from their bent tracks and the strength of the applied magnetic field (Sec. 3.2.1. The expected resolution of the impact parameters (Sec. 4.2) and the transverse momentum in the inner tracker is shown in Fig. 3.3.

3.2.3 Electromagnetic calorimeter

The electromagnetic calorimeter (ECAL), which is surrounding the inner tracker, is the second layer of the CMS detector. It is designed to measure the energy of electrons and photons. The ECAL is built of lead tungstate crystals (PbWO$_4$), characterized by a radiation length of $X_0 =$ 0.89 cm and a Molière radius of $r_m = 2.2$ cm. The radiation length is the distance, where high-energetic particles carry a fraction of $\frac{1}{e}$ of their initial energy, because of electromagnetic interaction with the detector material. The Molière radius defines the radius of a cylinder, that contains 90% of the energy of an electromagnetic shower, caused by electrons or photons. Both parameters are relative small, compared to other materials used in calorimeters. This yields a fine granularity. Furthermore, the lead tungstate crystals are radiation resistant and allow a fast emission of the scintillation photons, which is needed for the 40 MHz bunch crossing rate, provided by the LHC.

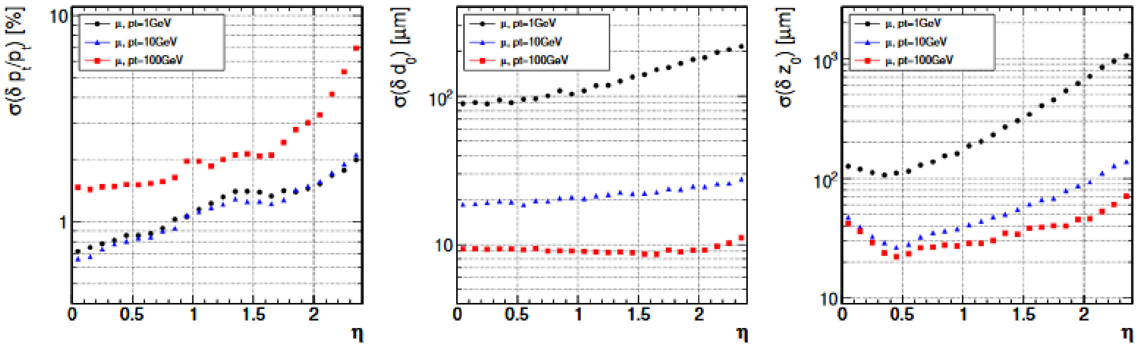


Fig. 3.3: Resolution of the transverse momentum **(left)**, the transverse impact parameter **(middle)** and the longitudinal impact parameter **(right)** of single muons with transverse momenta of 1, 10 and 100 GeV [2].

The ECAL barrel, with an inner radius of 1.29 m and an outer radius 1.77 m, covers a region, where $|\eta|$ is smaller than 1.479. It consists of 61,200 crystals with a size of the front face of around 22×22 mm^2 and a length of 23 cm (25.8 X_0). The scintillation light is amplified and detected by silicon avalanche photodiodes. In each of the ECAL endcaps 7324 crystals are located at a distance of 314 cm from the nominal collision point. The pseudorapidity range of $1.479 < |\eta| < 3.0$ is covered by structures in units of 5×5 crystals, with a front face size per crystal of 28.6×28.6 mm^2 and a length of 22 cm ($24.7X_0$). Vacuum phototriodes are utilized for the detection of the scintillating signal in the endcap tungstate crystals.

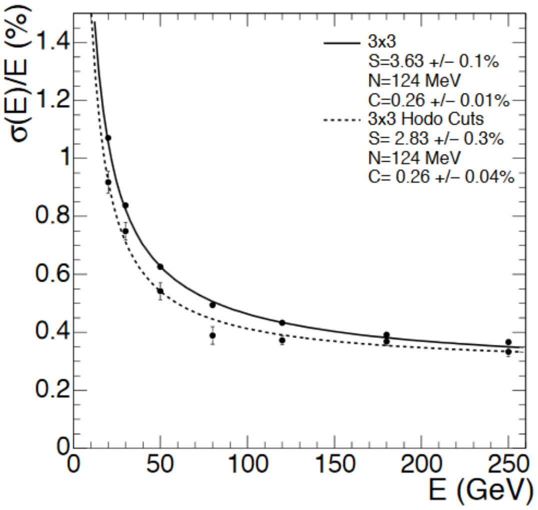


Fig. 3.4: Energy resolution of the CMS ECAL. $\frac{\sigma E}{E}$ is given as a function of the energy of an electron test beam. The measured energy in the ECAL is obtained from a 3×3 crystal module, with an electron hit in the central crystal [37].

The energy resolution of the ECAL in a test beam is shown in Fig. 3.4 and is calculated by

$$\left(\frac{\sigma E}{E}\right)^2 = \left(\frac{S}{\sqrt{E}}\right)^2 + \left(\frac{N}{E}\right)^2 + C^2, \tag{3.6}$$

where S refers to the stochastic term (e.g. fluctuations in the lateral shower containment), N is the noise (e.g. pileup or electronics noise) and C is a constant term (e.g. calibration errors).

The energy resolution achieved for photons in the ECAL barrel and endcap region is given by

$$\frac{\sigma E}{E} = \frac{2.7\%}{\sqrt{E}} \pm 0.5\%, \text{ for the barrel, and} \tag{3.7}$$

$$\frac{\sigma E}{E} = \frac{5.7\%}{\sqrt{E}} \pm 0.5\%, \text{ for the endcaps.} \tag{3.8}$$

3.2.4 Hadron calorimeter

The third subdetector inside CMS is the hadron calorimeter (HCAL). It is a sampling calorimeter, which contains alternating layers of absorber plates, made of brass, and plastic scintillators. Since brass is non-magnetic and provides a short interaction length for hadronic particles, it is qualified to be used as an absorber material within the CMS detector. The HCAL is designed to measure the energy and the timing of the neutral and charged hadrons, emerging in an event, as effectively as possible. This is essential to provide a good measure for the missing transverse energy $\not{E}_{\mathrm{T}}$ within an event, which indirectly measures the momentum of non-interacting particles, like neutrinos.

The HCAL barrel covers the region $|\eta| < 1.4$ and is located between the ECAL and the superconducting solenoid. In order to the achieve a maximum thickness of the absorber material in the barrel, very thin scintillator plates (3.7 mm) are used. An additional scintillator plate with a thickness of 9 mm is installed between the ECAL and the first absorber layer. The scintillators are connected to wavelength shifting fibres. These fibres collect the scintillation light and carry it to the read out system, which is based on multi-channel hybrid photodiodes. Furthermore, another calorimeter part is physically located outside the superconducting solenoid, at the barrel of the muon detector. This outer calorimeter ($|\eta|$ <1.26) makes use of scintillator plates with a thickness of 10 mm and acts as a “tail catcher” for hadrons, that are showering late and therefore pass the inner calorimeter. It helps to improve the energy and $\not{E}_{\mathrm{T}}$ resolution of the HCAL. Both calorimeters together yield a total thickness of the absorber material, that corresponds to more than 10 interaction lengths of hadronic particles.

The pseudorapidity range of $1.3 < |\eta| < 3.0$ is covered by the two HCAL endcaps. Again brass plates, with the thickness of around 10 interaction lengths and thin plastic scintillators (9 mm), which are attached to wave shifting fibres, are applied. An additional calorimeter is placed in the forward region with a high pseudorapidity ($3.0 < |\eta| < 5.0$). The particle fluxes hitting this forward calorimeter are considerable higher, than for the other HCAL parts. Thus, a high radiation resistance is required for the calorimeter material. In this case, steel is selected for the absorber plates and quartz fibres are used as scintillators. The Cherenkov light, which is emitted by the quartz fibres, is then carried to photomultipliers, to amplify the signal.

The designed energy resolution of the HCAL [53] is given in Eq. (3.9).

$$\left(\frac{\sigma E}{E}\right)^2 = \left(\frac{100\%}{\sqrt{E}}\right)^2 + 4.5\% \tag{3.9}$$

3.2.5 Muon system

The fourth and outermost subdetector is the muon system. Muon detection is a central task at the CMS experiment, as its middle name already indicates. The reason for this is, that multi-muon final states may hide a large discovery potential for both predicted SM processes and new BSM physics. The main purposes of the system are the identification of muons, the measurement of their momenta, as well as the triggering of events. Muons loose less energy in the tracker material, because of radiation, than electrons. Thus, muons can be detected with a better energy resolution. The muon system covers an area of about 25,000 m^2 with its gaseous particle detectors. The structure of the muon system and the locations of the different detectors are shown in Fig. 3.5.

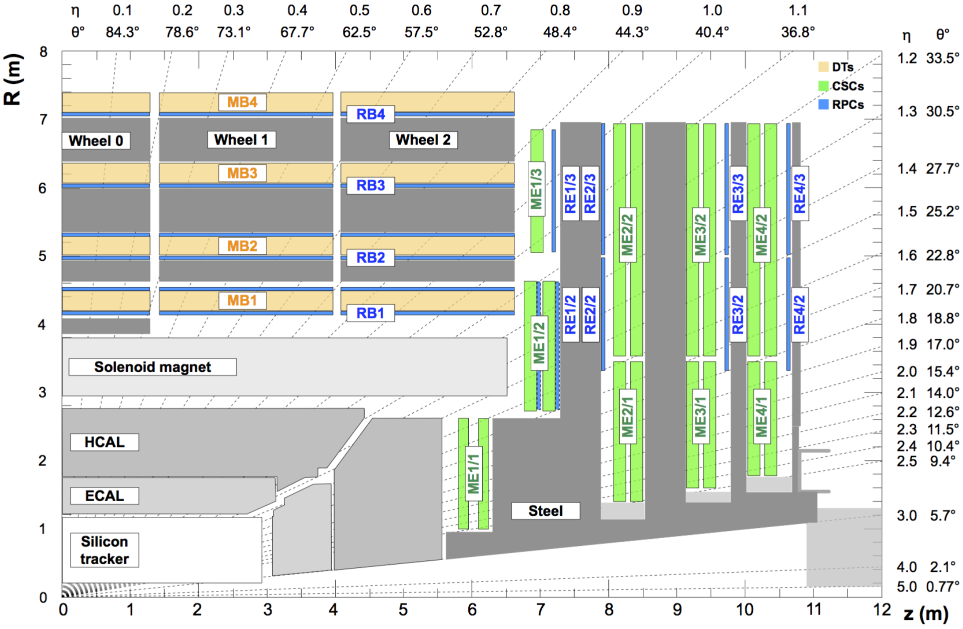


Fig. 3.5: The locations of the different gaseous particle detectors in muon system at the CMS experiment in the r-z plane [54].

Drift tube chambers (DTs) are utilized in the barrel region ($|\eta| < 1.2$), where the muon and background rates are low and the residual magnetic field outside the coil is low too. These chambers are separated into four segments. The first three stations contain DTs to measure the z position, as well as the muon coordinates in the r-ϕ plane. The last muon segment does only provide the chambers, that measure the coordinates in the r-ϕ plane. The DTs are arranged in a way, that optimizes the angular resolution of the muon system

Cathode strip chambers (CSCs) are applied in each of the two endcaps ($0.9 < |\eta| < 2.4$). In this region the magnetic field is non-uniform and large. Furthermore the muon and background rates are higher, than in the barrel. The CSCs in the endcaps are orientated perpendicular to the beam direction, in order to measure the coordinates in η and the r-ϕ plane. As in the barrel, four segments of chambers are installed in each endcap.

An third type of gaseous particle detectors is additionally applied in both the barrel and the endcaps. So-called resistive plate chambers (RPCs) are characterized by a fast response and a good time resolution. Therefore they are utilized to determine the actual bunch crossing and are deployed as an independent and fast trigger. The RPCs cover a region with $|\eta| < 1.6$.

3.3 Data acquisition and event triggering

The LHC provides a bunch crossing rate of 40 MHz (25 ns) for the proton beams. On average, 20-40 collisions per bunch crossing occur. Assuming that per bunch crossing approximately 1-2 MByte of data is generated, storing each event would demand tremendous requirements on the data transfer and the storage space. Therefore, triggers are applied to reduce the amount of the recorded data.

At the CMS experiment, two trigger stages are utilized to preselect events from the data. First, the level-1 (L1) trigger ensures a reduction of the data rate of a factor of about 1,000. It is a custom designed hardware trigger which uses only a fraction of the total event information. The L1 trigger is based on programmable electronics, processing data from the calorimeters and the muon system. It has a maximum output rate of 100 kHz. Events that pass the L1 trigger are then facing the high-level trigger (HLT). In a second step, this trigger further attenuates the data rate by another factor of approximately 1,000, such that the final output rate is in the range of a few 100 Hz. In contrast to the L1 trigger, the HLT is based on a reconstruction software, that utilizes the full event data. It is physically located at a computer farm, where the HLT algorithms select the remaining events. Both, the trigger and the data acquisition system (DAQ), are permanently refined and improved [55]. A schematic illustration of the L1 trigger and the DAQ is shown in Fig. 3.6.

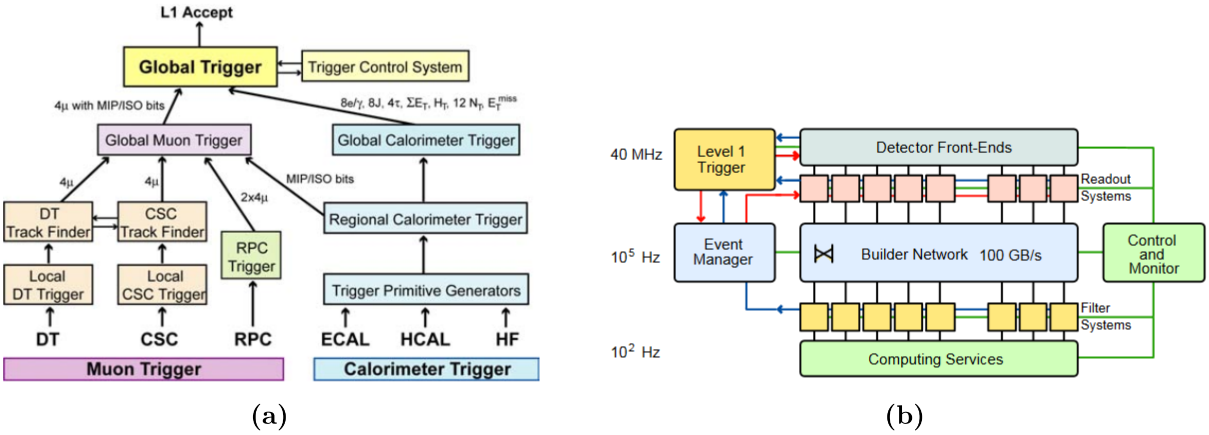


Fig. 3.6: Schematic figure of the L1 trigger **(a)** and the data acquisition system **(b)** at the CMS experiment [2]. The L1 trigger is using data from the muon chambers and the calorimeters. The DAQ provides a final output rate of around 100 Hz.

3.4 Particle reconstruction and particle flow algorithm

The purpose of the CMS particle flow (PF) algorithm is to reconstruct particles of the following types: electrons, muons, neutral hadrons, charged hadrons, and photons. The PF algorithm combines the calorimetry, tracking and muon information, in order to identify particles based on compatible detector signals according tho their type. It is the basis for further reconstruction and identification algorithms, like the reconstruction of jets, the identification of tau and bottom quark jets, or the determination of the missing transverse energy $\not{E}_{\mathrm{T}}$ of the event.

In the following the reconstruction algorithms of the various particle types are outlined according to [56–58]. Especially, the electron and muon reconstruction is treated in more detail. It is worth to note that lepton reconstruction and identification are two different tasks, with different purposes. The latter is discussed in Sec. 4.3.

3.4.1 Electrons

The electron reconstruction is performed by combining the information of the hits in the inner tracker and the energy deposits in ECAL. So, there are two main tasks, in order to achieve a proper reconstruction of electrons. Firstly, the mapping of the energy deposits into so-called ECAL clusters and superclusters, to determine the initial energy of an electron. Secondly, the track reconstruction by means of various tracking algorithms and compatible ECAL clusters.

In terms of energy reconstruction of the electrons, the radiated bremsstrahlung along the electron trajectory has to be taken into account. The photons are emitted in different directions in ϕ, due to the azimuthal bending in the magnetic field, while their distribution in η is minimal. This causes a distinct electron shower shape in the ECAL, characterized by a tail of bremsstrahlung photons. Single photons and non-radiating electrons, on the contrary, have a negligible distribution of energy deposits in the ϕ direction. Furthermore, usually about 97% of their energy is placed within a 5 × 5 array of crystals in the ECAL.

In order to reconstruct the initial electron energy, seed crystals are identified. This is achieved by requiring a measured transverse energy E_T in the respective ECAL crystal, that exceeds a minimum energy E_T^{min}. The seed crystals correspond to the centers of potential electron showers in the ECAL. Then, two different techniques are applied in the barrel and endcap regions.

5 × 1 arrays of crystals (in the $\eta \times \phi$ plane) are assigned to a seed crystal, within a $\Delta\phi^{range}$ in the azimuthal direction of the barrel region. The arrays are added as long as the sum of the energy deposits within the array surpasses a certain threshold $E_{\mathrm{T,\ array}}^{min}$. An additional threshold value $E_{\mathrm{T,\ seed\text{-}array}}^{min}$ is used for the seed array, that contains the seed crystal. Clusters of crystal arrays, that satisfy the requirements, are referred to as ECAL superclusters.

In the endcap regions, the missing crystal arrangement in the η-ϕ plane, necessitate a different approach. In this case, clusters of 5 × 5 crystals around the seed crystals are considered for the energy deposits. These clusters, which may partly overlap, are formed into superclusters, if their entire energy $E_{\mathrm{T,\ cluster}}$ within a particular range in $\Delta\eta^{ange}$ and $\Delta\phi^{range}$, is greater than a minimum value $E_{\mathrm{T,\ cluster}}^{min}$.

The different threshold values, utilized in the ECAL energy reconstruction, are given in Tab. 3.1. The position of the ECAL supercluster is determined as the energy-weighted mean, of the single cluster positions, by taking into account the shower depth.

Tab. 3.1: Minimum energy thresholds applied in the ECAL barrel and endcaps, in terms of energy reconstruction [58].

parameter	barrel threshold	endcap threshold
$E_{\mathrm{T,\ seed\text{-}array}}^{min}$	1 GeV	0.18 GeV
$E_{\mathrm{T,\ array}}^{min}$	0.1 GeV	
$E_{\mathrm{T,\ seed\text{-}array}}^{min}$	0.35 GeV	
$E_{\mathrm{T,\ cluster}}^{min}$		1 GeV
$\Delta\eta^{range}$		0.07
$\Delta\phi^{range}$	17 crystals ≈ 0.3 rad	0.3 rad

In the magnetic field of the CMS detector, electron (or positron) tracks differ from other charged particle tracks by their bremsstrahlung energy loss. Hence, the trajectory of an electron is likely to be more bent at the end of its path trough the inner tracker, than in the beginning. This enables a differentiation from other charged particles, like charged hadrons. In turn, a Kalman-filter based tracking algorithm is less efficient for electrons. In particular, tracker hits can get lost, when the curvature of the electron track is changing. For this reason, an additional approach using a Gaussian sum filter (GSF) [59] is applied in the reconstruction of electron tracks. The GSF takes into account abrupt and significant energy losses alongside the trajectory and is therefore more adapted to reconstruct electron tracks.

The track reconstruction starts with the so-called seeding. This refers to techniques, that try to find the first two or three hits in the inner tracker, which mark the beginning of an electron track. Two different approaches are deployed. First, the ECAL-based seeding proceeds from all ECAL clusters with a transverse energy E_T larger than 4 GeV. The cluster position and its energy deposits are utilized to deduce the location of the hits, produced in the innermost tracker layer, under the electron hypothesis. Electrons can loose a significant fraction of their energy via the emission of bremsstrahlung photons, depending on the structure of the tracker material along the trajectory. Thus, the efficiency of the ECAL-based seeding is related to quality of collecting the radiated bremstrahlung in association with the respective electron. This approach works for well isolated electrons, with an unbiased energy in the ECAL cluster. Lager inefficiencies arise, if energy deposits from other particles are overlapping the electron signature, like e.g. in jets. In this case, multiple hits in the innermost tracker layer can be compatible with the biased ECAL cluster position and energy. In order to avoid high misreconstruction rates, the ECAL-based seeding is restricted to electrons fulfilling a tight isolation criterion. Furthermore, a selection criterion according to the ratio of the energy deposits in the HCAL H and the ECAL supercluster $E_{\text{supercluster}}$, is required, to reduce the hadronic background.

$$\frac{H(\Delta R < 0.15)}{E_{\text{supercluster}}} < 0.15 \tag{3.10}$$

H is the sum over all HCAL energy deposits within a cone of $\Delta R < 0.15$ in relation to the electron direction.

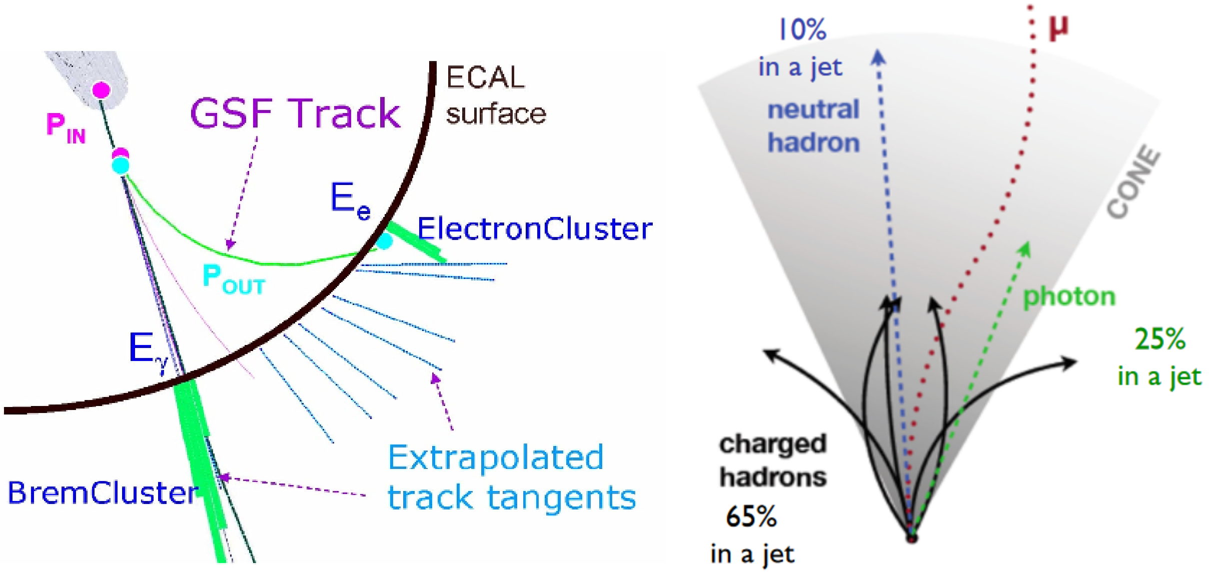
The second and complementary approach is the tracker-based seeding. Here, the potential electron tracks are matched with ECAL clusters that feature a compatible energy and position. Electrons with a transverse momentum of $p_{\text{T}} > 2$ GeV are considered. The electron tracks are obtained by an iterative tracking algorithm, which is the standard tool for tracking charged particles in the CMS detector. This algorithm yields a good performance for electrons, that do not loose a lot of their energy in the tracker, or even do not radiate at all. For such electrons, the track fit can be built with a low χ^2 and therefore is effectively assigned to a matching ECAL cluster position. Furthermore, the track p_{T} and the E_T of the calorimeter deposits is required to be in the same order of magnitude. For electrons that radiate bremsstrahlung photons, the χ^2 of the track fit is higher and tracker hits can get lost. Here, the previously mentioned GSF shows a better performance. Subsequently, a BDT is trained, using different parameters from the two track fits (the Kalman-filter and the GSF-based tracking) and the ECAL cluster, as well as the number of tracker hits. A final requirement on the output of the BDT defines the fraction of electrons, that are passing the tracker-based seeding.

In a last step, electrons form the ECAL- and tracker-based seeding collection are joined and another more advanced, but also more time consuming, GSF is applied. The electron charge is determined according to the curvature of the GSF track. The estimation of the electron momentum is performed, based on the measured data in the tracker and the ECAL.

3.4.2 Photons

At the CMS detector, photons are distinguished from electrons by energy deposits in the ECAL without a matching track in the inner tracker. Furthermore, photons have a different shower shape, because of the absence of bremsstrahlung. Their energy in the PF algorithm is calculated by forming ECAL superclusters, as described in the previous section about electron reconstruction. Separated calorimeter deposits with no compatible tracks are reconstructed as isolated photons. Converted photons are recognized by combining the measurements from the tracker and the ECAL, which enables to find possible conversion points. Bremsstrahlung photons, with their characteristic contribution to the shower shape, are assigned to the respective electron by a back-propagation of their track to the electron trajectory (Fig. 3.7a).

All other ECAL clusters, which are not compatible to any electron or positron track within the tracker acceptance ($|\eta| < 2.5$), a reconstructed as non-isolated photons, emerging e.g. in hadronic jets. In this case, neutral hadrons are a potential source of background. However, it can be observed, that neutral hadrons deposit only around 3% of the jet energy within the ECAL, while photons make up about 25% [56] (Fig. 3.7b). Above the tracker acceptance, charged and neutral hadrons can not be discriminated. Hence, photons are only reconstructed, if there are no matching energy deposits in the HCAL under the hadron hypothesis.



(a) (b)

Fig. 3.7: Photons in the PF algorithm. The landscape of ECAL energy deposits **(a)**, due to an electron radiating bremsstrahlung [60]. The electron track is reconstructed by a GSF fit, until it enters the ECAL. Energy deposits, caused by bremstrahlung photons are spread in the ϕ direction of the ECAL surface. The particle composition within jets **(b)**. Photons and charged hadrons make up around 90% of the jet energy, which can be measured with a high precision, because of a high tracking resolution (<1%) and energy resolution [60].

3.4.3 Muons

In terms of muon reconstruction, tracks from the inner tracker (tracker track) and from the muon chambers (standalone track) are treated separately, first. Tracker tracks are reconstructed by means of an iterative technique using various slightly different tracking algorithms. They make use of the measured hits in the silicon pixel and strip detectors. A tracking algorithm, based on a so-called Kalman-filter [61] is utilized to reconstruct the standalone tracks. In this case the respective information is obtained from the DTs, CSCs and RPCs alongside the muon track.

In a second step tracker tracks and standalone tracks are combined. Therefore two distinct approaches are applied. On the on hand, an inside-out algorithm extrapolates a tracker track to the muon chambers. This is done for all muons with a total momentum $p > 2.5$ GeV and a transverse momentum $p_\mathrm{T} > 0.5$ GeV. An track, that is associated to a so-called arbitrated tracker muon, is generated, if the extrapolated trajectory matches at least one of the hits in the muon chambers. On the other hand, an outside-in approach links the standalone tracks to appropriate tracker tracks by applying a Kalman-filter algorithm. If two tracks are compatible, the hits in the muon chambers are combined with hits from the inner tracker and form a global muon track.

Muon reconstruction is very effective. About 99% of the muons emerging in the detector are reconstructed as tracker muons, global muons or as both. In the last case, a double counting is prevented. Muons that reach only the innermost muon segment, are reconstructed as a tracker muon. Muons traversing the whole muon system are usually detected as global muons. Hence, the inside-out algorithm (arbitrated tracker muons) yields better efficiencies for low-p_T muons, while the outside-in approach (global muons) is more efficient for high-p_T muons. However, late showering hadrons, that pass the HCAL, may also be detected in the first muon station. These punch-through particles are considered as a serious background for the tracker muons. Global muons have a lower background rate due to punch-through hadrons. Furthermore, high-p_T global muons ($p_\mathrm{T} > 20$ GeV) profit from a higher momentum resolution, compared to the tracker-only fit, because of the additional information of the hits in the muon chambers.

Finally, the reconstructed muons are considered for further selection criteria in the PF algorithm, which forms the basis of the muon identification (Sec. 4.3.1.2). Here, the muon tracks are associated with the energy deposits in the ECAL and the HCAL. The selection criteria for PF muons are presented in the context of the POG muon ID in Sec. 4.3.1.2. Especially high-p_T charged hadrons that reach the first muon segment are removed from the reconstructed muons by this algorithm.

The momentum p of the muons is determined from the inner track, if their p_T is smaller than 200 GeV. Otherwise, the track with the smallest χ^2 of its fit is considered. Reconstructed muons passing the muon PF algorithm are excluded from the possibility to be reconstructed as any other particle type. However, if the momentum of a charged hadron is assessed to be too high, muon identification is reviewed with looser selection criteria.

3.4.4 Hadrons

Charged and neutral hadrons usually leave their energy in both calorimeters, the ECAL and the HCAL. A neutral hadron is reconstructed, if no matching track can be found for energy deposits within a HCAL cluster. In contrast, HCAL energy clusters, that do have a compatible track and momentum in the inner tracker are recognized as charged hadrons. Furthermore, residual ECAL clusters, that correspond to the charged hadron track are taken into account. Outside the tracker acceptance ($|\eta| > 2.5$), matching ECAL clusters are considered for the neutral hadron energy too, for reasons, that have been stated in the photon reconstruction. Thus, the energy assigned to a hadron in the PF algorithms depends, whether it is a charged or a neutral hadron, as well as if it occurs within the tracker acceptance or not. An overview is given in Tab. 3.2.

Tab. 3.2: Energy reconstruction of neutral and charged hadrons, by means of ECAL and HCAL energy deposits [56].

hadron type	ECAL deposits	HCAL deposits	track
neutral hadron within $\|\eta\| < 2.5$	allowed	yes	no
neutral hadron within $\|\eta\| > 2.5$	yes	yes	no
charged hadron	yes	yes	yes

The momentum of a charged hadron is based on its tracker track, but is optimized by a χ^2 fit, which additionally takes in consideration the position of the compatible clusters in the ECAL and the HCAL.

3.4.5 Jet energy corrections and missing transverse momentum

The jet energy correction (JEC) [62] is performed on jets with a $p_\mathrm{T} > 10$ GeV and a pseudorapidity of $\eta < 5.2$. It is composed of three steps. First, a pileup correction is applied, in order to exclude particles from the event, that arise from other proton-proton collisions. In-time pileup, due to additional events within the same bunch crossing, is reduced by rejecting particles, that originate from well reconstructed pileup vertices. Out-of-time pileup, from events of different bunch crossings, can be mitigated by using the time related measurement data of the calorimeters. Another technique is the so-called pileup per particle identification (PUPPI) weight [63], that indicates the probability of being from a pileup vertex, for each particle within the jet. Next, initial state radiation (ISR) and final state radiation (FSR) are corrected by means of comparisons between detector data and simulated MC data (Sec. 3.5). Finally, the η and p_T dependent detector response is determined and taken into account for the JEC.

Furthermore, energy corrections depending on the jet flavor can be applied. The response for heavy-flavor jets, due to bottom or charm quarks, is lower than for light-flavor jets. The reason for this is, that heavy-flavor quarks can decay e.g. into lighter quarks and leptons.

The accuracy of the energy reconstruction of all objects within an event, including the jet energy, is very crucial for the determination of the missing transverse energy $\not{E}_\mathrm{T}$ [56]. It indicates the occurrence of particles, that pass the detector, without interacting with the material, e.g. neutrinos. They can be detected only indirectly, via an unbalanced vectorial sum of the transverse momenta $\vec{p}_\mathrm{T,i}$ of all particles $N_\mathrm{particles}$, that are measured by

$$\vec{\not{p}}_\mathrm{T} = -\sum_{i=1}^{N_\mathrm{particles}} \vec{p}_\mathrm{T,i}. \tag{3.11}$$

If the transverse momentum $\vec{p}_\mathrm{T}^{\,\mathrm{corr.}}$ is propagated to $\not{E}_\mathrm{T}$, the expression becomes

$$\vec{\not{p}}_\mathrm{T} = -\sum_{i=1}^{N_\mathrm{particles}} \vec{p}_\mathrm{T,i} - \sum_{j=1}^{N_\mathrm{PF\ jets}} \left(\vec{p}_\mathrm{T,j}^{\,\mathrm{corr.}} - \vec{p}_\mathrm{T,j}\right). \tag{3.12}$$

3.5 Data simulation

Simulated data plays an important role in the particle physics, observed at the LHC, as well as in many other fields of experimental physics. It creates the connection between theoretical predictions, on the one hand, and the observations by the experiments, on the other hand. The simulated data at the CMS experiment is produced by so-called Monte-Carlo (MC) event generators [64]. Matrix elements are calculated for the first orders in perturbation theory. At low energy scales, QCD processes can not be treated by means of perturbation theory. The confinement of the partons in the proton beams and the re-hadronization to final-state hadrons, after a collision is not computable, but instead must be modeled empirically. This is done with parton shower modules such as Pythia [65].

The MC enables a comparison of the state of the art theory in particle physics, i.e. the SM, with the measured data in the proton-proton collisions at the LHC. Furthermore, the predictions of BSM theories can be tested for their consistency with the observed physical processes. Another benefit of the MC is, that these events contain the full generator information, including all initial, intermediate and final state particles. This is called the MC truth information. It allows for e.g. the supervised learning technique (Sec. 5.1), which is applied in this work.

Pythia [65], POWHEG [66], Herwig [67], or MadGraph [68] are MC event generators, utilized for the simulation of high-energy proton-proton collisions at the LHC. Pythia is mainly deployed to simulate QCD processes. Higgs and diboson production, as well as $\mathrm{t\bar{t}}$ signals are procuced with the additional aid of POWHEG. MadGraph is applied in terms of e.g. $\mathrm{t\bar{t}}$ + jets events, or DY processes (Sec. 4.1), as well as for the simulation of BSM models. Parton showering is generated with Pythia 8.2 or Herwig 7.

Technically the generation of MC is performed in several steps [69]. First, the hard scattering processes of an event at a high-energy scale are simulated, based on the parton distribution functions (PDFs), which describe the incoming partons within the proton beams. Then the parton showers after the hard collision are modeled, including initial and final state radiation. The next stage of the MC event generators is the hadronization process of the partons. While the first two steps can be calculated via perturbative QCD, a model has to be built for the non-perturbative hadronization processes, for some reasons explained earlier. Furthermore, secondary decays are simulated in this third step. The fourth stage considers the evolution, i.e. the interaction and hadronization, of the residual partons, that are not directly involved in the hard scattering process. Finally the decay of unstable particles in treated.

In order to take into account the detector material, the“geometry and tracking (Geant)4” toolkit [70] is utilized. Geant is a software that simulates the transition of particles through matter, by means of MC techniques. This toolkit is modeling the geometry and the material properties of the CMS detector and yields simulated hits for the artificial event. Detector hits due to pileup events are generated too. In a addition to this, the detector electronics are imitated by adding some noise, in a separate stage [71]. Lastly, the digitization results the simulated raw data, which can be processed subsequently by reconstruction algorithms (Sec. 3.4) in an equivalent manner as the real detector data.

3.6 Data storage

Different data formats [72] are used, in order to store the detector data and the MC. First, the raw data is digitized and various algorithms are applied, in order to produce the reconstructed data (Sec. 3.4), referred to as “RECO”. On the one hand, RECO includes low-level physics information, like reconstructed hits in the inner tracker and the muon stations, or measured calorimeter deposits. On the other hand, it stores high-level physics objects, like electrons, muons, photons, jets $\not{E}_T$.

Twice a year, the reconstructed data is processed into the analysis object data (AOD). The AOD is a data format, that contains all high-level physics objects and a compressed version of the lower-level RECO information, which is sufficient for all common analysis use cases at CMS. The AOD format is centrally produced and stored at multiple CMS computing sites. In the AOD, one event makes up a size of about 480 kB.

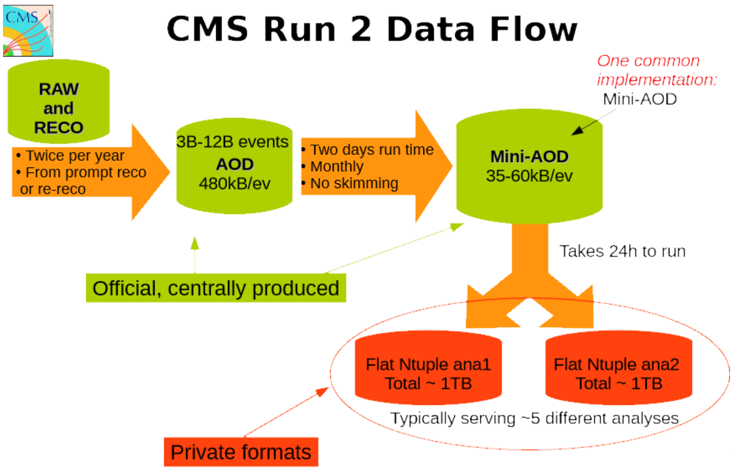


Fig. 3.8: The data flow in the CMS experiment during the LHC Run 2 [72].

Another data format, which was introduced first in CMS at the LHC Run 2, is the so-called mini-AOD. It is generated centrally from the AOD format and has about 10% of its size (40-50 kB per event). The mini-AOD reduces disk and I/O resources, which are necessary for the analysis, but also provides enough information for around 80% of the CMS analyses. This downsized format contains the following information:

- High-level physics objects: like leptons, photons, jets, and $\not{E}_T$
- The basic kinematic data of all PF candidates: reconstructed by the CMS PF algorithm (Sec. 3.4)
- The trigger information
- Simulated particles, both final state particles and generated jets, plus their reference information
- Other information, such as the primary and secondary vertices and various filter decisions

The mini-AOD is the basis for further private data formats and analyses. The schematic data flow of the CMS data in the LHC Run 2 is illustrated in Fig. 3.8. In addition to this, a novel and even more compressed data format, named nano-AOD [73], is available for typical analyses. In this case, the data size is further reduced to around 1 kb/event.

Chapter 4

Lepton identification at the CMS experiment

In this section an overview of the state-of-the-art techniques and algorithms in lepton identification at the CMS experiment is given. First, two important processes, that generate opposite charged final state lepton pairs, are described. Then, standard approaches in lepton identification, as well as more advanced approaches based on machine learning techniques are discussed.

4.1 Production of dilepton final states at the LHC

4.1.1 The Drell-Yan process

The Drell-Yan (DY) process describes a production mechanism of lepton-antilepton pairs in hadron-hadron collisions at high center-of-mass energies. It was named after Sidney Drell and Tung-Mow Yan, who first postulated this process in 1970 [74].

The initial state is a quark-antiquark pair, which annihilates into a virtual photon γ^* or a Z boson, provided that the net electric charge of the quark-antiquark pair is zero. If the net electric charge is equal to ± 1, the quark pair can annihilate into a W boson. In the first case, the Z boson can further decay into a lepton-antilepton pair with a probability of around 10% ($\mathrm{Z} \to \ell^-\ell^+$) [75]. The Feynman diagram of this process is shown in Fig. 4.1a.

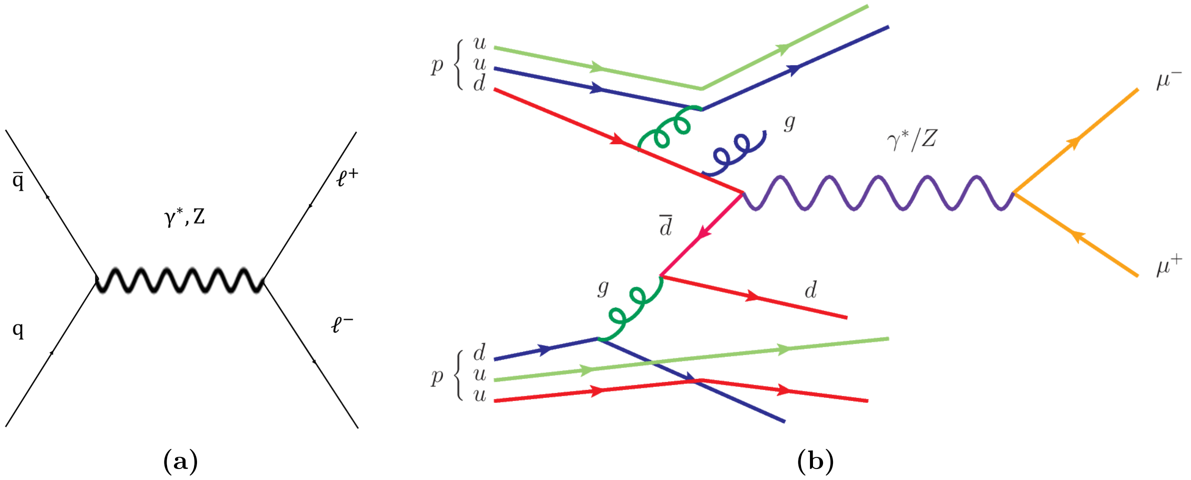


Fig. 4.1: The Drell-Yan process. The Feynman diagram **(a)** of the Drell-Yan process and the example of a DY process **(b)** in a proton-proton collision [76].

The colored initial-state partons can lead to initial-state radiation, as also shown in Fig. 4.1b. In this example of a proton-proton collision at the LHC, e.g. a down quark from one proton (duu) may annihilate with a down antiquark comming from another proton (Fig. 4.1b) [76]. Thus, the produced final state muons ($Z \rightarrow \mu^-\mu^+$) are not the only particles, which are expected in this event. The initial-state ratediaton of highly energetic partons causes the formation of extra jets in the event. Together with the leptons from the Z boson, they define the main signature of the event. Such DY processes can be measured with a high precision, because of the very clean final state.

The spectrum of the invariant mass of the lepton-pair ($m_{\ell^-\ell^+}$) is characterized by several peaks, due to mass resnonances with SM particles (Fig. 4.2). The peak at around 91.2 GeV corresponds to the resonance of the invariant mass with the mass of the Z boson (m_Z) [74]. Thus, the DY process is very important, in terms of detector calibration. It is used as a so called "standard candle" [77]. According to the invariant mass spectrum, the range of 6 GeV to around 40 GeV is referred to as low-mass and the region beyond 40 GeV is called high-mass in the following [78].

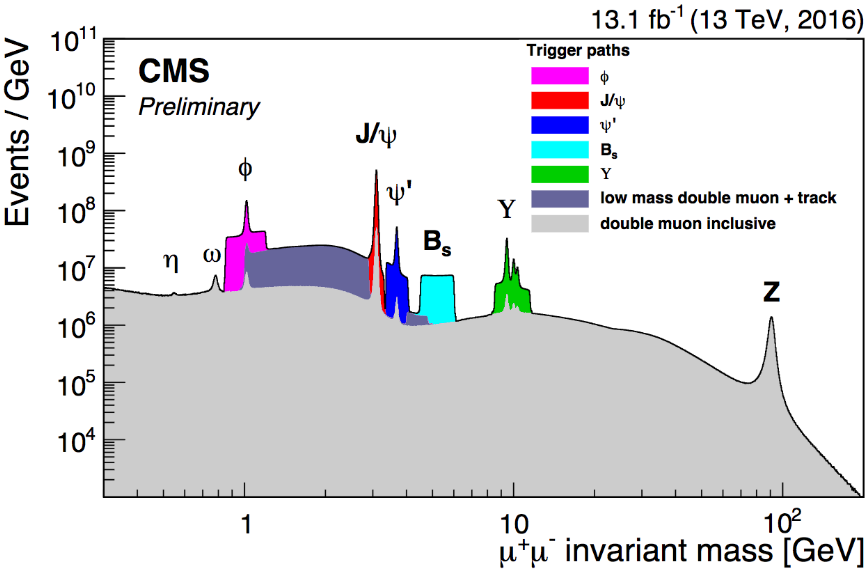
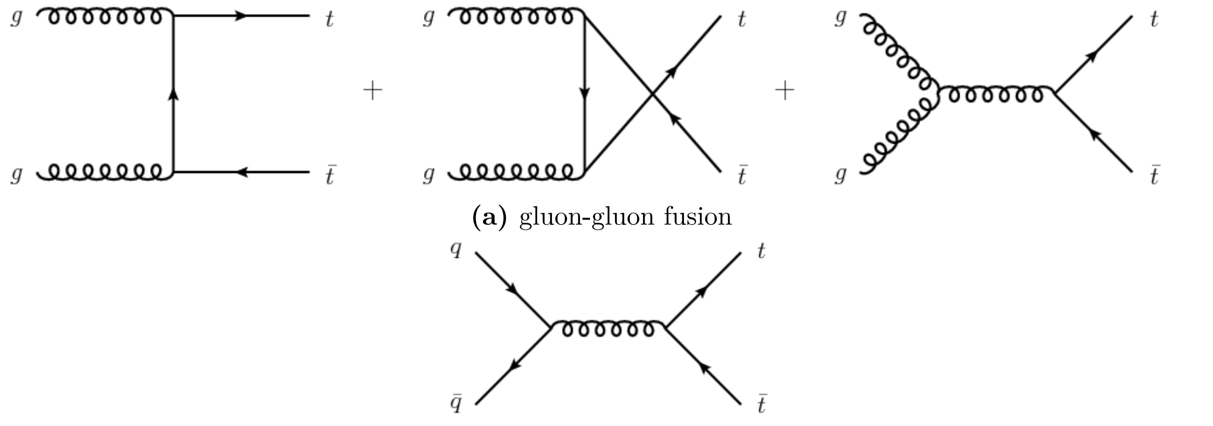


Fig. 4.2: Measured spectrum of the dimuon invariant mass with the resonance peak around the mass of the Z boson at $m_Z \cong 91.2$ GeV. The residual peaks refer to other resonances with SM particles above a continousely diminishing background of DY events [79].

Apart from the calibration, this process is also relevant for quantifying SM parameters, as well as for searches of new physics beyond the SM. Due to its high cross section and the clean final state, this process can be measured with a high efficiency [78]. Examples are the determination of the masses and production rates of the W and Z bosons, the measurement of the PDFs inside the protons and the limitation of new physics models containing new heavier particles, similar to W and Z bosons [80].

4.1.2 The $t\bar{t}$ dilepton final state

Another important process, with a final state lepton-antilepton pair, can emerge in so called $t\bar{t}$ events, where a top-antitop quark pair is produced initially in the proton-proton collision. At the LHC top pairs are normally produced through the strong interaction. Initial states can be e.g. two gluons or a quark-antiquark pair, but at the LHC the gluon-gluon fusion (Fig. 4.3a) dominates towards the quark-antiquark annihilation (Fig. 4.3b) [81].



(a) gluon-gluon fusion

(b) quark-antiquark annihilation

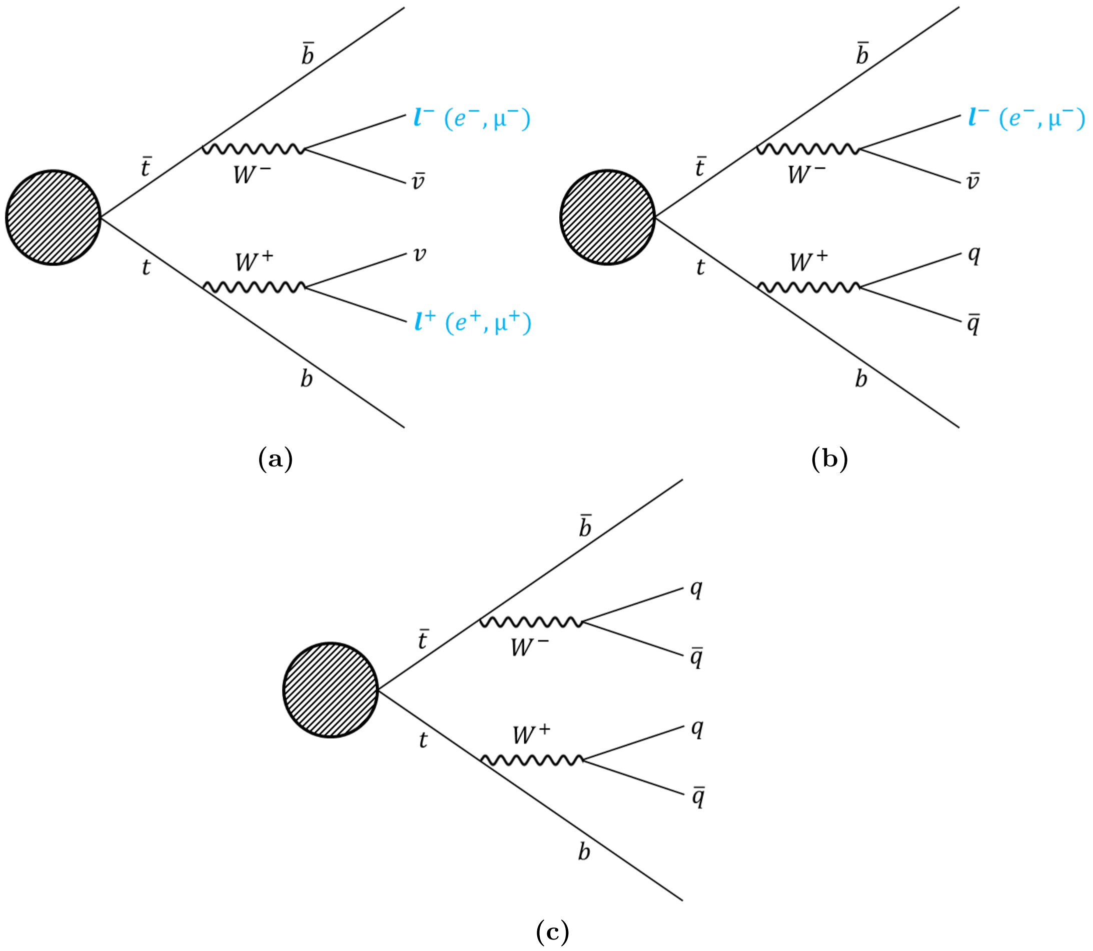
Fig. 4.3: Feynman diagrams for common top pair production at LHC [81].

The top quark is the only quark within the SM, that has a higher mass than the W boson [82]. As a consequence of this high mass, the top quark propagates just for a very small time, so that it already decays before it comes to a hadronization. Usually the top quark decays into a W boson and a bottom quark (t → Wb). Although in principle decays into strange and down quarks are possible (t → Ws or t → Wd), they are suppressed by very small values of the respective matrix elements $|V_{\mathrm{ts}}|$ and $|V_{\mathrm{td}}|$ in the unitary Cabibbo–Kobayashi–Maskawa (CKM) matrix [83, 84]. This matrix indicates the strength of the weak interaction, when mixing up- and down-type quarks via the W boson. The matrix is given by

$$\begin{aligned} |V_{CKM}| &= \begin{bmatrix} |V_{\mathrm{ud}}| & |V_{\mathrm{us}}| & |V_{\mathrm{ub}}| \\ |V_{\mathrm{cd}}| & |V_{\mathrm{cs}}| & |V_{\mathrm{cb}}| \\ |V_{\mathrm{td}}| & |V_{\mathrm{ts}}| & |V_{\mathrm{tb}}| \end{bmatrix} \\ &= \begin{bmatrix} 0.97428 \pm 0.00015 & 0.22530 \pm 0.0007 & 0.003470 \pm {}^{0.00016}_{0.00012} \\ 0.22520 \pm 0.00070 & 0.97345 \pm {}^{0.00015}_{0.00016} & 0.041000 \pm {}^{0.0011}_{0.0007} \\ 0.00862 \pm {}^{0.0026}_{0.0002} & 0.04030 \pm {}^{0.0011}_{0.0007} & 0.999152 \pm {}^{0.000030}_{0.000045} \end{bmatrix} . \end{aligned} \tag{4.1}$$

Flavor changing decays via the neutral Z boson (t → Zq) are suppressed too (by the GMI mechanism [85]). Therefore, the previously mentioned decay into a W boson and a bottom quark (t → Wb) occurs in almost 100% of the time [82]. The bottom quark hadronizes, which can be measured as a jet in the detector. The W boson subsequently decays either in a leptonic or a hadronic way. Hence, the final states are a lepton and a neutrino ($\mathrm{W}^+ \to \ell^+\nu$ or $\mathrm{W}^- \to \ell^-\bar{\nu}$), or a quark-antiquark pair ($\mathrm{W} \to \mathrm{q}\overline{\mathrm{q}}$).

Considering the $\mathrm{t}\overline{\mathrm{t}}$ events with t → Wb decays, there are three possible final states according to the combinatorics of the two different types of decay of the W boson [86]. Firstly, a final state with two leptons and two neutrinos ($\ell^+\nu$ plus $\ell^-\bar{\nu}$), which is referred to as the (di)leptonic channel (10%). Secondly, the semi-leptonic channel (44%) with a lepton plus a neutrino and a quark-antiquark pair ($\ell^+\nu$ or $\ell^-\bar{\nu}$ plus $\mathrm{q}\overline{\mathrm{q}}$), and thirdly two quark-antiquark pairs ($\mathrm{q}\overline{\mathrm{q}}$ plus $\mathrm{q}\overline{\mathrm{q}}$) in the hadronic channel (46%). All three processes are illustrated in Fig. 4.4, where the first two occurrences are also known as the $\mathrm{t}\overline{\mathrm{t}}$ dilepton (Fig. 4.4a) and single lepton (Fig. 4.4b) final states.



(a) **(b)**

(c)

Fig. 4.4: Schematic illustration of possible $\mathrm{t\bar{t}}$ events, in three different decay channels. The $\mathrm{t\bar{t}}$ dilepton final state (dileptonic channel) **(a)**, the $\mathrm{t\bar{t}}$ single lepton final state (semi-leptonic channel) **(b)**, and the $\mathrm{t\bar{t}}$ hadronic channel **(c)**.

Focusing the $\mathrm{t\bar{t}}$ dilepton final state, the signature of this kind of events is a lepton-antilepton pair, two b-jets from the hadronization of the bottom quarks and a certain amount of missing energy $\not\!\!E_{\mathrm{T}}$, due to the two neutrinos, which do not interact with the detector material [86]. This signature is often overlaid by one or more jets. They can originate e.g. from hadronizing quarks that are not directly involved in the process, or even from other events, which is known as pileup [87].

4.1.3 Dileptonic backgrounds

In practice it turns out, that often it is not trivial to unambiguously identify a process. Indeed, both the DY process (Sec. 4.1.1), and the $\mathrm{t\bar{t}}$ dilepton final state events (Sec. 4.1.2) contain a final state lepton-antilepton pair. A distinguishing characteristic is the missing energy $\not\!\!E_{\mathrm{T}}$ in $\mathrm{t\bar{t}}$ events from the two invisible neutrinos. A requirement of $\not\!\!E_{\mathrm{T}} \geq 40$ GeV is a common procedure, in order to discriminate these two processes. However, there is the probability that some DY events pass an $\not\!\!E_{\mathrm{T}}$ cut, e.g. because of mismeasurement of the jet energies [78].

As already mentioned, jets and pileup can complicate the proper reconstruction of an event. In some cases jets are falsely reconstructed as leptons [87]. Hence, $\mathrm{t\bar{t}}$ single lepton events or other

W → $\ell\nu$ + jets events, where one of the jets is misidentified as a lepton (Fig. 4.5a), are a major source of the background for both event types, DY and the $t\bar{t}$ dilepton final state [78]. Another problem is the in-flight decay of hadronized quarks into one or more leptons, e.g. a B-meson, which can derive from the hadronization of a bottom quark (Fig. 4.5b). In this case, two leptons can be detected in the semi-leptonic channel.

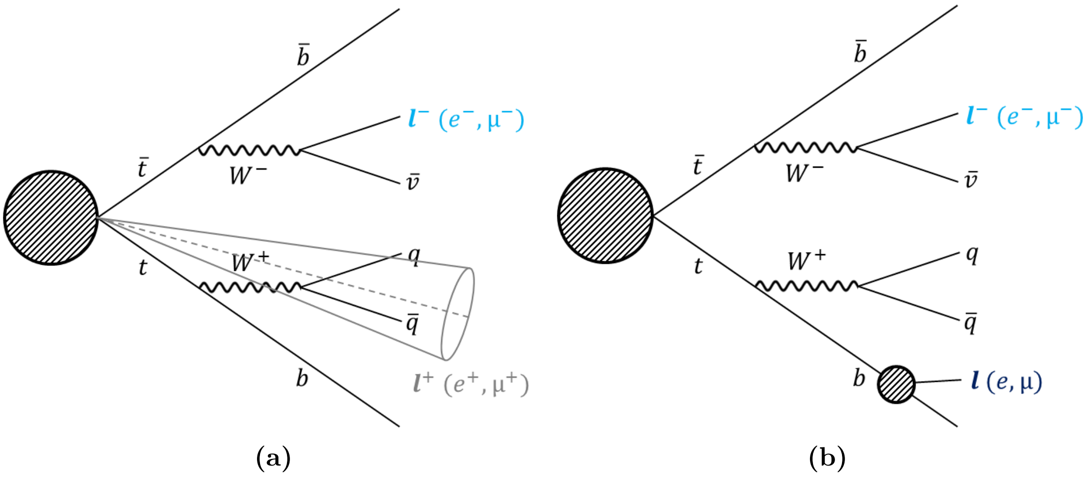


Fig. 4.5: $t\bar{t}$ events in the semi-leptonic channel with extra leptons. Additional fake lepton **(a)**, due to the misidentification of a jet. Additional non-prompt lepton **(b)**, due to the decay of a hadronized bottom quark.

Different diboson productions (WW, WZ or ZZ) with leptonic decays and the Z → $\tau^+\tau^- \to \mu^+\mu^-$ process make an additional, but smaller contribution to the background. Furthermore, QCD processes can make up a non-negligible background, especially for the low-mass region in DY [78].

On this account, it becomes clear that the determination of the origin of the final state leptons is of great importance, in order to reduce the background events within a desired signal process. The characteristics of the leptons and the different techniques which are applied to achieve an appropriate separation of the signal and background, are outlined in the following sections.

4.2 Lepton features and classes

Leptons can be divided into three different classes, regarding their source from the hard scatter. The classes are labeled as prompt, non-prompt and fake. Usually prompt leptons are considered as signal leptons, whereas non-prompt and fake leptons are forming the background. There are distinct lepton features, in order to discriminate the three lepton classes. Besides global lepton features, also flavor specific features for electrons and muons are introduced in the following.

4.2.1 Prompt leptons

Prompt leptons from the decay of vector bosons originate from the primary vertex (PV). It is the reconstructed location of the main collision, that is taking place in an event. In many analyses, prompt leptons are the objects of interest, since they can either be helpful to select certain processes or to search for BSM physics.

4.2.2 Non-prompt leptons

Non-prompt leptons are formed at a secondary vertex (SV), a second reconstructed point of interaction. A SV usually occurs later in the event, due to the decay of longer-lived particles. The main source of non-prompt leptons is the semi-leptonic decay of heavy flavor hadrons, containing bottom or charm quarks and sometimes strange quarks [88]. A typical process, is the decay of a B-meson into one or more leptons, after the hadronization of a bottom quark. But also in-flight decays of pions and kaons are a potential source [89]. Thus, non-prompt leptons usually appear in jets, sprays of particles emerging from showers and hadronizations of quarks [90]. Furthermore, photons can convert into an electron-positron pair, when interacting with matter, like e.g. the detector material in front of the ECAL [91]. Another source of non-prompt muons are cosmic muons [89].

One measure that indicates non-prompt leptons is the so called impact parameter [92]. The impact parameter describes the smallest distance between the reconstructed track of the respective particle and the PV. Non-prompt leptons are characterized by a higher impact parameter d_0, than prompt ones, since the SV is expected to be displaced from the PV for a certain distance. An example of an event with a SV, due to a b-jet is illustrated in Fig. 4.6. In case of cosmic muons, a discrimination is usallay easy, since their tracks do not match with the PV at all.

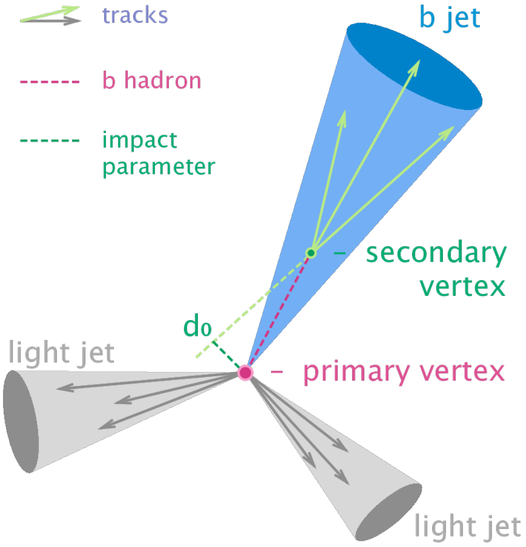
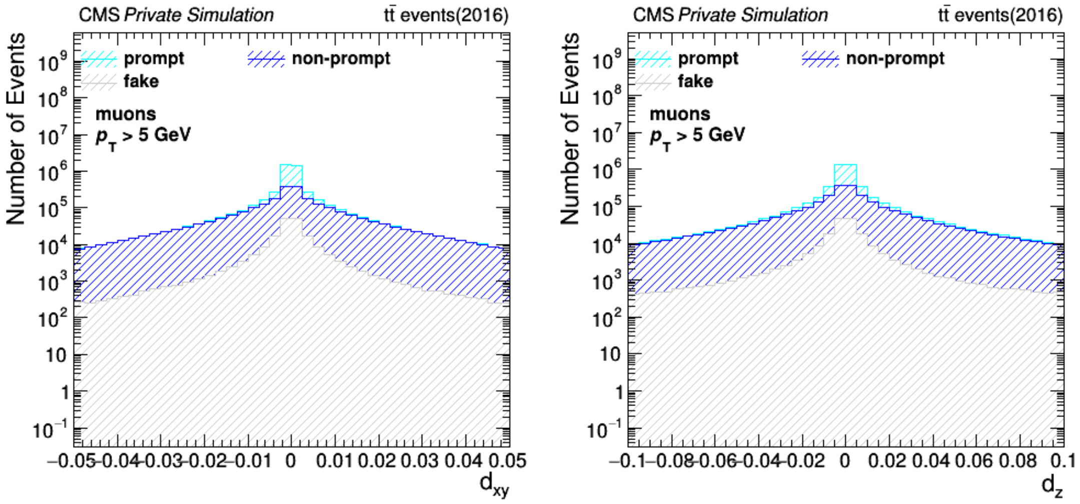


Fig. 4.6: Illustration of the impact parameter d_0 for an event with a b-jet, from the hadronization of a bottom quark, and two other light jets. The secondary vertex (SV) of the b-jet is displaced from the primary vertex (PV). The smallest distance between the reconstructed track of a particle within the b-jet and the PV is given by d_0. The image is adapted from [93].

At the LHC, the common impact parameters d_{xy} and d_z, which refer to the smallest distance in the x-y plane and along the z axis, are used as a feature in identifying non-prompt leptons. A distribution of these two parameters d_{xy} and d_z in simulated $t\bar{t}$ events is shown in Fig. 4.7.

Another distinguishing parameter is the isolation, or often also the relative isolation of a lepton. The isolation is defined as the inverse ratio of the transverse momentum of the lepton $p_{T,lep}$ and the sum over the transverse momenta of all n particles within a certain cone around the respective lepton. In case of the relative isolation



(a) d_{xy} distribution **(b)** d_z distribution

Fig. 4.7: Impact parameter distributions for prompt, non-prompt and fake muons in simulated $t\bar{t}$ events. Prompt leptons are expected to be close to the primary vertex, while non-prompts and fakes have a broader distribution.

$$I_{\text{rel.}}(\Delta R \leq 0.3) = \frac{\sum\limits_{i=1}^{n} p_{\text{T},i}}{p_{\text{T},lep}}, \tag{4.2}$$

the lepton itself is excluded from the n particles, which are added up in the sum [92].

$$\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2} = \sqrt{(\phi_{lep} - \phi_i)^2 + (\eta_{lep} - \eta_i)^2} \tag{4.3}$$

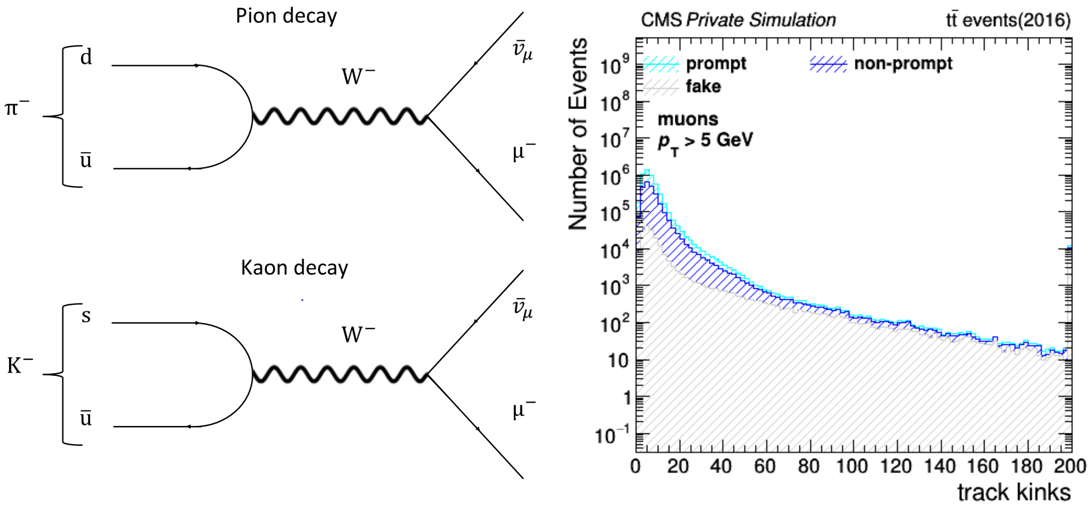
is utilized as a measure for the cone size. It indicates the angular separation between two particles in the ϕ-η plane, e.g. between the lepton and a particle i within a jet.

Prompt leptons are supposed to be more isolated than non-prompt leptons, unless a jet is overlapping with a prompt lepton by chance. The relative isolation is usually used for discriminating prompt and non-prompt leptons in the high-p_T range. The (absolute) isolation, in contrary works better for low-p_T leptons, where e.g. low-p_T bottom quarks can decay with larger opening angles [92].

Non-prompt muons emerging from in-flight decays of pions or kaons are characterized by a kink in the tracks at the point of the decay (Fig. 4.8a) [89]. The so-called track kink finder algorithm [94] addresses this issue and enables to recognize such processes. It searches for kinks along the reconstructed muon track and calculates the χ^2 between inward and outward track states for each layer of the silicon tracker (Fig. 4.8b).

4.2.3 Fake leptons

Fake leptons conceptually differentiate from prompt and non-prompt leptons, as they are no real leptons. This fake lepton class consists of objects, that are wrongly reconstructed as leptons.



(a) **(b)**

Fig. 4.8: Example Feynman diagrams of semi-leptonic meson decays **(a)**. On top a pion decay $\pi^-(\mathrm{d\bar{u}}) \rightarrow \mu^- + \bar{\nu}_\mu$ and below a kaon decay $\mathrm{K}^-(\mathrm{s\bar{u}}) \rightarrow \mu^- + \bar{\nu}_\mu$, marked by the characteristic kinks at the decay points. The distribution of the track kink finder variable **(b)** for prompt, non-prompt and fake leptons in simulated 2016 MC $\mathrm{t\bar{t}}$ samples.

Hence, there are distinct processes for fake electrons and muons leading to a false classification, which are closely related to the setup and the geometry of the CMS detector [95].

Considering electrons, the reconstruction is based on the particle tracks in the silicon tracker and the energy deposits at the electromagnetic calorimeter (ECAL). On source of fake electrons are photons, which shower at the ECAL and are linked mistakenly to a charged track at the silicon tracker. The energy deposits of a photon can be similar to the ones of an electron. Usually around 94% of the energy is located within a 3x3 array of crystals and around 97% in a 5x5 shape [96]. A parameter to distinguish electrons from photons at the ECAL is the so called shower shape. Electrons are emitting a tail of bremsstrahlung photons on their curved path through the detector, due to the applied magnetic field and the interaction with the detector material. Hence, the energy deposits of electrons are spread in ϕ. This results in a different shower shape for electrons, as no bremsstrahlung is emitted by the photons. It is worth to note, that material in front of the ECAL can cause photon conversions into non-prompt electrons. Therefore, an adequate separation of electrons and photons in the reconstruction algorithm is essential.

There are several variables describing the shower shape of a particle at the ECAL, like $R9$ or $\sigma_{i\eta i\eta}$. The observable $R9$ is calculating a simple ratio of energy deposits within crystal areas, where $E_{3\times3}$ is the energy measured in a 3×3 crystal array and $E_{\text{supercluster}}$ refers to a certain cluster of clusters of crystals [97].

$$R9 = \frac{E_{3\times3}}{E_{\text{supercluster}}}, \text{ or (sometimes) } \frac{E_{3\times3}}{E_{5\times5}} \tag{4.4}$$

However, variables like

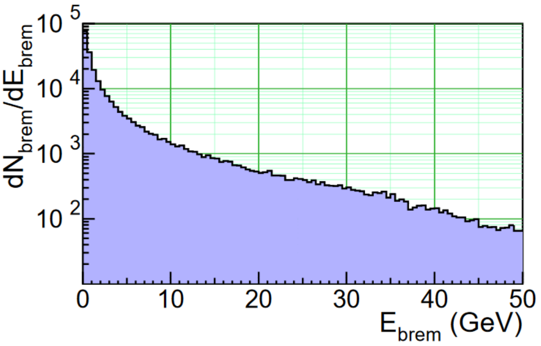


Fig. 4.9: Spectrum of the bremsstrahlung emitted by electrons (p_T =30 GeV, $|\eta|$<1.5) in the tracker material. The bremsstrahlung, which reaches the ECAL appears as a tail of the actual electron energy. The tail is spread in ϕ with constant η [96].

$$\sigma_{i\eta i\eta} = \left(\frac{\sum_i^{5\times5}(\eta_i - \bar{\eta}_{5\times5})^2 * \omega_i}{\sum_i^{5\times5}\omega_i} \right)^{\frac{1}{2}} \text{, with} \tag{4.5}$$

$$\bar{\eta}_{5\times5} = \frac{\sum_i^{5\times5}\eta_i\omega_i}{\sum_i^{5\times5}\omega_i} \qquad \text{and} \tag{4.6}$$

$$\omega_i = 4.2 + \ln\frac{E_i}{E_{5\times5}} \tag{4.7}$$

are based on the determination of the root mean square (RMS) of the shower width [98]. Instead of using the absolute η of the respective crystal, the idea is to define the

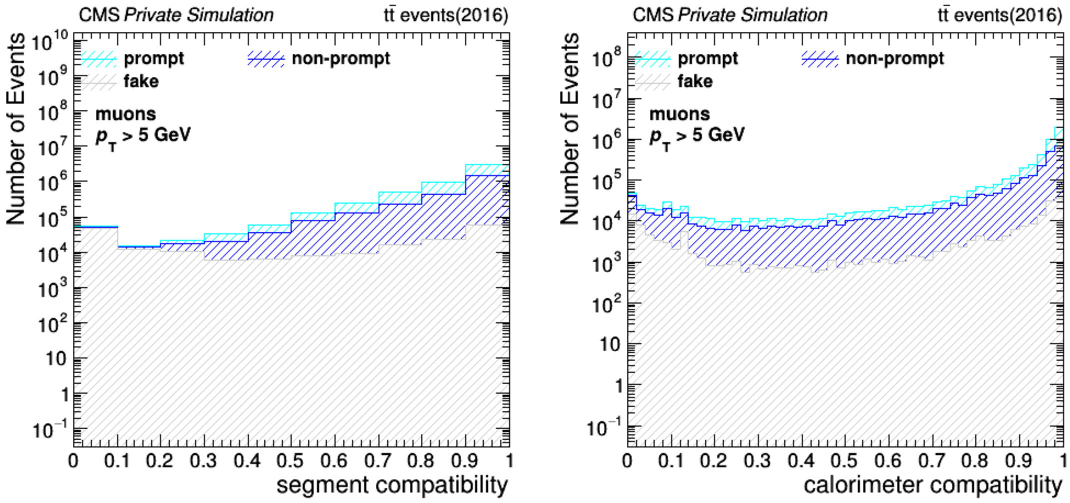
$$\eta_i = n_{\text{crystals}} \cdot 0.0175 + \eta_{\text{seed crystal}} \tag{4.8}$$

$$\tag{4.9}$$

of a certain crystal i, by simply counting the number of crystals n_{crystals} from the seed crystal $\eta_{\text{seed crystal}}$ and multiply it with the average $\Delta\eta_{\text{crystal}} = 0.0175$ of a crystal. Seed crystals are crystals containing an energy, which is higher than a particular threshold value [96].

A second source of fake electrons are hadronic jets, consisting e.g. of charged and neutral pions [95]. Jets, containing a single charged pion, can cause an appropriate track in the inner tracker. Afterwards the decay of a π_0 meson into two photons may lead to a significant deposit of the jet's energy in the ECAL. If this happens and the charged pion track is assigned to the energy deposit of the photons, a fake electron is reconstructed by the algorithm. In this case, the shower shape is again a distinguishing feature [98]. Electrons usually deposit most of their energy in one or two crystals in η, whereas jets have a broader distribution. Furthermore, muons, which are not reconstructed as such ones, can fake an electron through the combination of their track in the inner tracker and a matching deposit in the ECAL, caused by a photon. [95].

Muon reconstruction, on the contrary, is based on the hits in the outer muon chambers (standalone muons). In combination with a matching inner track a so called global muon is reconstructed (see Sec. 3.4). A particle which fakes a muon, therefore must have an adequate lifetime to reach the muon detector system [95]. Potential candidates are charged hadrons and punch-through particles arising from high-energy hadronic showers [89]. If, in addition to this, a random track in the silicon tracker can be assigned to these particles by chance, it is possible that a fake muon is reconstructed out of this bunch of unrelated hits [95]. Parameters which are useful to separate fake muons are e.g. the segment compatibility and the calorimeter compatibility. The muon segment compatibility is a measure for matching hits in the single muon chambers (segments). Fake muons tend to have a worse segment compatibility than prompt and non-prompt muons (Fig. 4.10a). The calorimeter compatibility indicates, if the calorimeter energy deposits of a respective object is consistent with a potential muon. High energy bunch-through particles can be recognized by large energy deposits and thus a low calorimeter compatibility (Fig. 4.10b).



(a)

(b)

Fig. 4.10: Discriminating features in fake muon identification. The segment compatibility **(a)** and the calorimeter compatibility **(b)**.

4.3 Approaches in lepton identification

A reliable identification is indispensable for any physics analysis using leptons. Especially for multi-lepton final states, high efficiencies in terms of identifying signal (prompt) and background (non-prompt or fake) leptons are required. Furthermore, the fraction of background leptons increases in the low-p_T region. Different approaches have been developed over time, from simple cut based IDs to advanced machine learning algorithms. An overview of the common techniques is given in the following sections.

4.3.1 Cut based lepton IDs

Starting with cut based electron and muon IDs, simple cut values for common lepton parameters are recommended by the CMS physics object group (POG). The lepton isolation and impact parameters are treated separately, as they are related closely to the respective analysis use case.

Lepton IDs for different standard working points are suggested by the electron and muon POG. A loose working point is usually chosen, if the backgrounds are fairly low. For analysis with events containing W and Z bosons, a medium working point is advised. In case of high and difficult backgrounds, a tight selection should be preferred. In addition to this also a very loose working point, referred to as veto, is offered. The fraction of background leptons is decreasing from a loose to a tight selection, but also the amount of signal leptons passing the applied cuts is shrinking towards a tighter selection. Therefore the optimal working point is very sensitive to the particular physics of the analysis.

4.3.1.1 Electron POG ID

Regarding particles which were reconstructed as electrons, distinct cuts are tuned for the ECAL barrel ($|\eta_{\text{supercluster}}| \leq 1.479$) and end-cap ($|\eta_{\text{supercluster}}| \geq 1.479$) regions. The current cut based electron IDs for 2016 detector and MC data are given in Tab. 4.1 [99].

Tab. 4.1: Cut based electron POG ID for 2016 detector and MC data samples [99].

	barrel				end-cap			
electron feature	veto	loose	medium	tight	veto	loose	medium	tight
full 5x5 $\sigma_{i\eta i\eta}$ <	0.0115	0.011	0.00998	0.00998	0.037	0.0314	0.0298	0.0292
$\lvert\Delta\eta_{seed}\rvert$ <	0.00749	0.00477	0.00311	0.00308	0.00895	0.00868	0.00609	0.00605
$\lvert\Delta\phi_{in}\rvert$ <	0.228	0.222	0.103	0.0816	0.213	0.213	0.045	0.0394
H/E <	0.356	0.298	0.253	0.0414	0.211	0.101	0.0878	0.0641
$I_{\text{rel.}}$ with effective area <	0.175	0.0994	0.0695	0.0588	0.159	0.107	0.0821	0.0571
$\lvert 1/E - 1/p\rvert$ <	0.299	0.241	0.134	0.0129	0.15	0.14	0.13	0.0129
expected missing inner hits ≤	2	1	1	1	3	1	1	1
pass conversion veto	yes	yes	yes	yes	yes	yes	yes	yes

4.3.1.2 Muon POG ID

The basic module of the POG muon IDs is the PF muon identification algorithm [56]. Here, isolated muons are obtained by requiring an selection criterion, based on the particle tracks and calorimeter deposits in the vicinity of the muons. This selection contains all reconstructed muons, where the sum of the transverse momenta p_{T} of all tracks and the transverse energy E_T of the calorimeter deposits, within a cone of ΔR <0.3 around the muon direction in the η-ϕ plane, is smaller than 10% with respect to the muon p_{T}. As a result, hadrons with a misidentification potential as muons are removed. Non-isolated muons, i.e. muons within jets, are identified by a tighter selection [100], which is similar to the tight muon POG ID in Tab. 4.2b. Further on, either track-matching hits in at least three muon stations or compatible energy deposits are required. This selection sorts out high-p_{T} (punch-through) hadrons. Muons, that do not pass the tight selection are recovered, if they have either a high quality standalone track (at least 23 out of 32 DT hits, or 15 out of 24 CSC hits), or a high quality inner tracker track (at least 13 hits) plus compatible calorimeter deposits.

The loose muon ID is based on particles, that withstand the previously mentioned PF muon identification algorithm and are pooled in the PF muon ID (`isPFMuon`) [100]. In addition to this, the particles must have been reconstructed either as global-muons (`isGlobalMuon`) or as arbitrated tracker muons (`isTrackerMuon`), in order to reject standalone and non-arbitrated tracker muons (see Sec. 3.4). A medium working point is defined with muons passing the loose ID plus extra requirements on the quality of the muon track and the particle itself (Tab. 4.2a) [101]. The χ^2 of the local position is a measure for the quality of the tracker-standalone position match of the reconstructed muon tracks, whereas the global-track χ^2 indicates, whether the reconstructed global track is consistent with the measured hits or not. It is normalized by a division by the number of degrees of freedom N_{dof}, which is defined as the difference between the number of data points (hits) and the number of fitted parameters in the reconstructed global-track. Therefore a normalized global-track $\frac{\chi^2}{N_{\mathrm{dof}}}$ roughly equal to 1 is preferable.

Tab. 4.2: Cut based muon POG IDs for run-II detector and MC data samples [101].

(a) medium muon POG ID

muon feature	requirement		
`isPFMuon`	yes		
`isGlobalMuon` or `isTrackerMuon`	yes		
fraction of valid tracker hits >	0.8		
AND			
segment compatibility >	0.451	OR	0.303
track kick finder <	-		20
χ^2 local position <	-		12
normalized global-track χ^2 <	-		3
`isGlobalMuon`	-		yes

(b) tight muon POG ID

muon feature	requirement
`isPFMuon`	yes
`isGlobalMuon`	yes
normalized global-track χ^2 <	10
# of valid muon chamber hits for global track fit >	0
# of matchted muon stations >	1
tracker-track $\lvert d_{xy}\rvert$ w.r.t PV <	0.2 cm
tracker-track $\lvert d_z\rvert$ w.r.t PV <	0.5 cm
# of valid pixel hits >	0
# of valid tracker layers with hits >	5

The loose and medium muon IDs are tuned for the identification of prompt muons, as well as for non-prompt muons emerging from heavy flavor decays. A third working point, the tight muon ID is designed to reduce the fraction of non-prompt muons from in-flight decays and to suppress punch-through particles and cosmic muons [101]. The requirements of the tight muon ID are listed in Tab. 4.2b. The cuts on the impact parameters $|d_{xy}|$ and $|d_z|$ of the tracker-track, w.r.t the primary vertex, are loose. They are selected to further decrease non-prompt muons from heavy-flavor decays, like from bottom and charm quarks. The cuts on the impact parameters can be tightened even more, if critical background is an issue due to the physics of the analysis use case (see Sec. 4.3.1.3).

4.3.1.3 Isolation and impact parameter

As already mentioned, the relative lepton isolation ($I_{\mathrm{rel.}}$) and the impact parameters (d_{xy}, d_z) are important features, in order to discriminate prompt and non-prompt leptons. These features are not included in the POG lepton IDs, but are treated separately. The reason for this is, that the efficiency of such isolation and impact parameter cuts is closely related to the physics of the considered event [99]. Furthermore, for some analyses non-prompt leptons might be of interest, as they are real leptons too. However, analyses addressing prompt final state leptons are more common.

Nevertheless, some standard cut values, which turned out to be good starting working points, are advised. Regarding impact parameters, common cut values to identify prompt electrons in the barrel are $d_{xy} < 0.05$ cm and $d_z < 0.1$ cm, while for end-cap electrons slight looser cuts, $d_{xy} < 0.1$ cm and $d_z < 0.2$ cm, are recommended. In terms of prompt muon identification a loose selection is applied by using the impact parameter cuts in Tab. 4.2b. Considering relative isolation (Eq. (4.2)), typical working points are listed in Tab. 4.3 [101].

4.3.2 Machine learning in lepton identification

Simple cut-based lepton IDs seem to be sufficient for events with high production rates and a clean final state, where backgrounds are not an issue. For special analysis use cases, where production rates are rather low, or when searching for new physics, higher requirements are

Tab. 4.3: standard working points in relative lepton isolation [101]

working point	$I_{\text{rel.}} <$
Very Loose	0.40 cm
Loose	0.25 cm
Medium	0.20 cm
Tight	0.15 cm
Very Tight	0.10 cm
Very very Tight	0.05 cm

imposed on the efficiencies in lepton identification. Recently, machine learning has boomed and its techniques are utilized in various scientific fields, likewise in particle physics. In the following sections a selection of approaches to improve lepton identification, by different machine learning techniques, is presented.

4.3.2.1 POG electron MVA ID

One of the first machine learning developments in lepton identification is a multivariate analysis (MVA) for electrons [102], which is based on a boosted decision tree (BDT) [103]. The basic concept of a BDT is to combine a series of rather weak single decision trees into a strong discriminator. MVAs, in general, learn to combine the information of multiple input parameters and output a single value for the respective object, e.g. an electron. This discriminator value can be applied for a signal and background classification, while in terms of the cut-based IDs, several cut values have to be tuned.

For the BDT training, DY + jets MC samples are utilized. Prompt electrons are considered as signal and non-prompt, as well as unmatched electrons form the background sample. The training is performed separately for three different η regions, the inner barrel, the outer barrel and the end-cap, as well as for two different p_{T} ranges, low-p_{T} (5 Gev $< p_{\mathrm{T}} <$10 GeV) and high-p_{T} ($p_{\mathrm{T}} >$ 10 GeV). In the MVA, provided by the POG, around 20 different electron variables are considered, such as p_{T}, $\eta_{\text{supercluster}}$, $\sigma_{i\eta i\eta}$, $R9$, kink finder hits, kink finder χ^2, or the normalized global-track χ^2, which are already introduced in the previous sections. Furthermore, PF isolation features are among the input variables. Two different versions of the electron MVA ID are offered, one with the isolation features and one without [102].

The full list of electron variables used e.g. for the 2017 version of the BDT training can be found in [104].

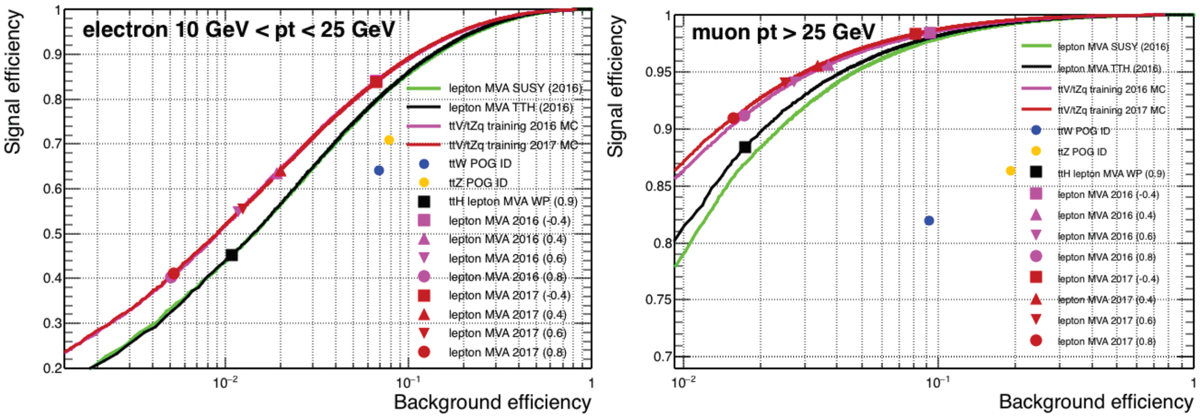
4.3.2.2 Lepton MVA IDs - *TTH* and *TTV* BDTs

A set of advanced lepton MVA IDs [105] have been developed for the purposes of $\mathrm{t\bar{t}V}$, $\mathrm{t\bar{t}H}$ and SUSY analyses. The IDs are based on BDTs using a forest of 1000 trees and a maximum depth of four nodes. They are trained and evaluated with the so called Toolkit for Multivariate Data Analysis (TMVA) [106], likewise the previous mentioned POG electron MVA. As input variables an appropriate selection of high discriminating features are considered (Tab. 4.4), which mainly cover the isolation and impact parameter part in order to separate prompt and non-prompt leptons. Regarding electrons, the POG electron MVA ID is included additionally. For muons the segment compatibility is an extra training parameter. Besides the common features, also an auxiliary b-tagger is implemented in the MVAs, specialized to identify jets arising from the hadronization of a bottom quark (see Sec. 4.3.2.3).

Tab. 4.4: Lepton MVA input training features [105].

feature type	variable
global features	lepton p_T, lepton η
isolation and closet jet	track multiplicity of closest jet, mini $I_{\text{rel.}}$ for charged hadrons, mini $I_{\text{rel.}}$ for neutral hadrons, $I_{\text{rel.}}(\Delta R \leq 0.3)$, jet $p_T^{\text{rel.}}$ v2, jet p_T ratio v2, `DeepCSV` b-tagger for closest jet
impact parameter	$\lvert d_{xy}\rvert$, $\lvert d_z\rvert$, significance d_{3D}
electron specific	electron MVA ID (no isolation)
muon specific	segment compatibility

Different lepton MVA IDs are available for 2016 and 2017 data samples, separated into electrons and muons. They have further improved the efficiencies in lepton identification. Especially if non-prompt backgrounds from leptonic decays of bottom quarks are an issue, like in several $t\bar{t}$ analyses, the lepton MVAs outperform the simple cut-based POG IDs. An example analysis use case is the “search for tZq in trilepton events” [105], where the BDT based lepton IDs are applied successfully.



(a) low-p_T electrons **(b)** high-p_T muons

Fig. 4.11: Efficiencies of various lepton MVA IDs utilized in the search for tZq in trilepton events. The BDT based lepton IDs clearly surpass the cut-based $t\bar{t}V$ and $t\bar{t}H$ POG working points [105].

4.3.2.3 b-tagging - The deepCSV algorithm

The identification of jets emerging from heavy flavor quarks, meaning bottom and charm quarks, is an important task for several analyses at the LHC, since these jets are a major source of non-prompt leptons. Hence, an efficient b-tagging (or c-tagging) helps to reduce such backgrounds [107]. Various algorithms have been developed to address this issue, mainly based on machine learning methods, like BDTs and neural networks. They are combining the information of the impact parameters, secondary vertices, possible nearby leptons and other jet features.

Therefore, these b-tagging techniques go under the name of Combined Secondary Vertex (CSV) algorithms [108]. They provide a single discriminator value for each analyzed jet. This value indicates the probability (calculated by the CSV algorithm) of being e.g. a b-jet, in case of a b-tagger.

One of the latest version is the DeepCSV b-tagger [108], which has been trained on a deep neural network (DNN), optimized for heavy flavor tagging. It is part of the *DeepJet* framework [6, 109], that is designed for jet identification and classification. It is introduced in Sec. 5.3.1. The DeepCSV tagger uses typical b-tagging features, containing the information of displaced tracks and vertices, as well as global features, like the jet transverse momentum $p_{\mathrm{T,\,jet}}$, or the number of charged particles in the jet. In total around 60 input features are trained in a DNN consisting of six layers, with 100 nodes per layer. The performance of the DeepCSV b-tagger compared to other CSV algorithms is illustrated in Fig. 4.12. It shows, that the DNN clearly has the potential to outperform those other b-taggers.

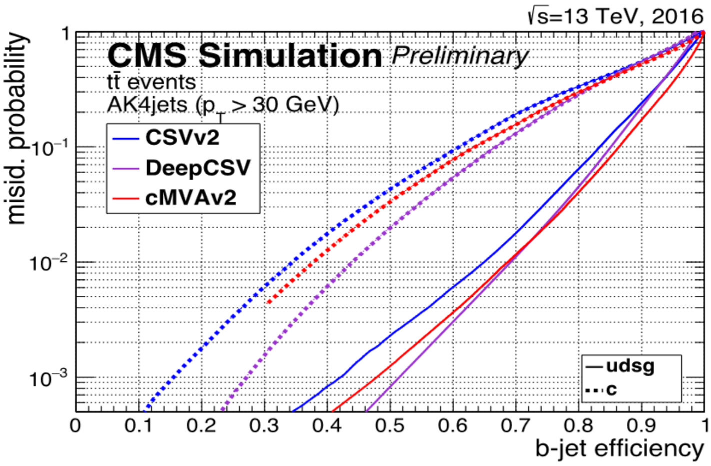


Fig. 4.12: ROC curves showing the performance of the DeepCSV b-tagger compared to other CSV algorithms [109].

4.3.2.4 Lepton isolation - *DeepIsolation*

The lepton isolation is another important feature, in discriminating prompt and non-prompt or fake leptons, since prompt leptons are expected to be more isolated. The standard approach, which has nothing to do with machine learning so far, is to calculate the relative isolation. It is the simple ratio of the sum of the transverse momenta of PF candidates within a fixed-size ΔR cone around the lepton and the transverse momentum of the lepton itself (see Eq. (4.2)). One machine learning technique is to train a BDT with the various isolation variables in order to provide an improved relative isolation. In general, isolation variables are high-level features, composed of lower-level features, which contain the information of the PF candidates around the respective lepton. By implication, this means that some information may get lost in those high-level features [110].

The *DeepIsolation* study [110] is a current approach in lepton isolation, using advanced machine learning tools. It addresses the previously mentioned issue of information loss, due to high-level features, by training a DNN with lower-level input features of the PF candidates within a $\Delta R \leq 0.5$ cone, in addition to global lepton features. Features which are used to represent the

PF candidates in the vicinity of a lepton are e.g. the PF candidate momentum divided by lepton momentum, the component of PF candidate momentum parallel to the lepton four-momentum, ΔR and the polar angle ($\alpha = tan^{-1}(\eta, \phi)$) w.r.t the lepton, or the so called PUPPI weight [63], which indicates the probability to be from pileup. For the charged PF candidates, features, that are representing the track quality and the compatibility with the primary vertex (PV), are included. The global features consist of typical kinematic and isolation features, as well as of the number of PF candidates per particle type and the number of vertices.

The *DeepIsolation* study showed, that a DNN trained with these input features, does outperform not only the $I_{\text{rel.}}$ variable, but also a BDT based isolation discriminator (see Fig. 4.13a) [110]. The architecture of the *DeepIsolation* neural network is heavily inspired by the *DeepJet* framework and is illustrated in Fig. 4.13b.

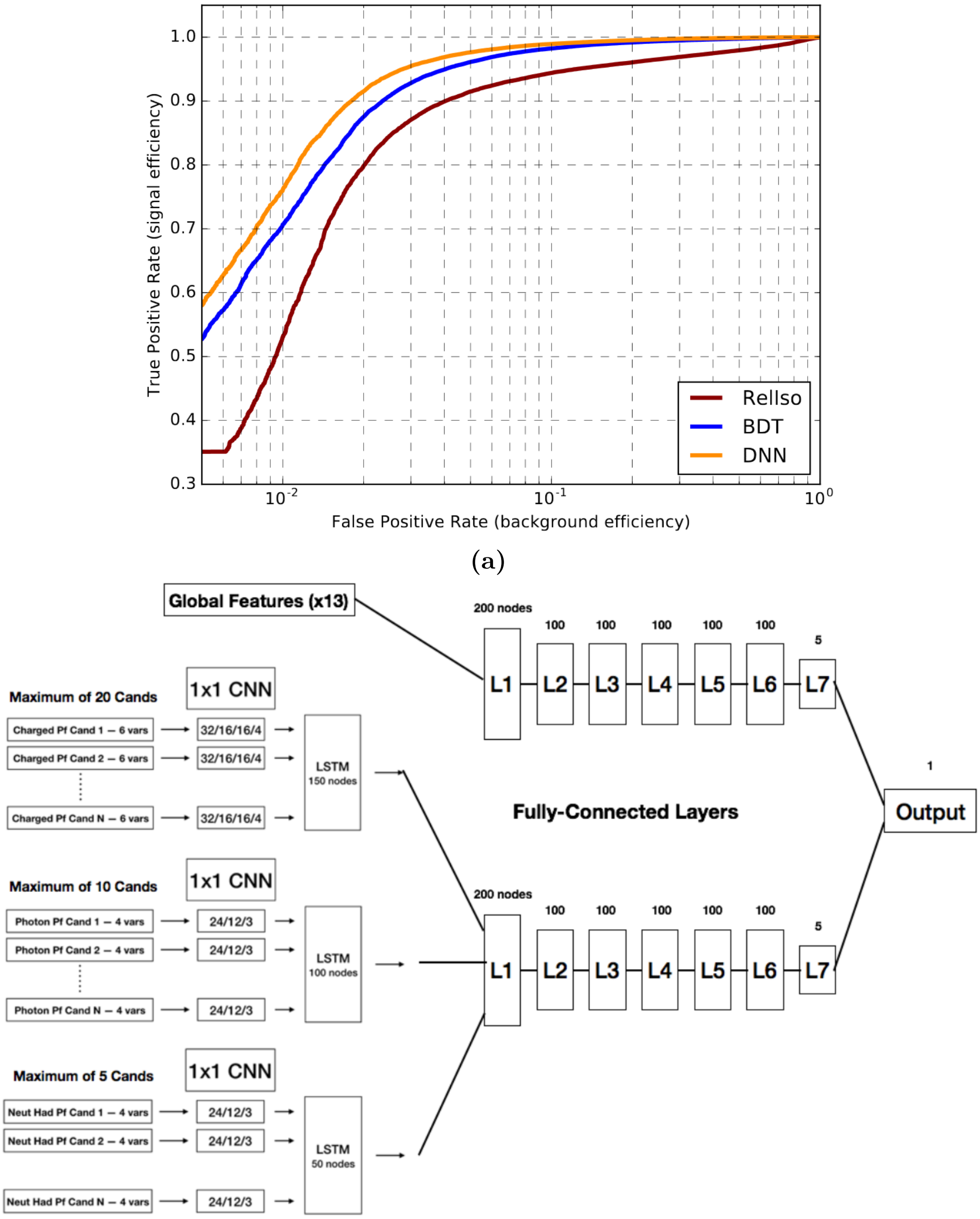
4.3.2.5 *DeepLepton* ID - the missing piece

So far, many efforts have been made in improving the discriminating power of single features or feature categories by means of machine learning methods, with success. However, there have been no recent studies in providing an overall lepton ID, that utilizes all these advanced algorithms, like the DeepCSV b-tagger or the lower-level PF candidate information in combination with a DNN. A *DeepLepton* ID is not only a logic consequence of the previous developments, but also a missing piece in the landscape of lepton IDs.

The *DeepLepton* neural network, which is the subject of this master thesis, is designed to compute exactly such a missing ID, which is long overdue. Its architecture is heavily inspired by the *DeepJet* framework and previous mentioned approaches in terms of b-tagging and isolation are combined in the *DeepLepton* input features. Its general purpose is to further improve efficiencies in lepton identification and thus, in event selection. In particular, it is designed especially for analyses, where difficult backgrounds are an issue, like the following use cases:

- "The search for stop quarks in opposite-charged dilepton final states" [3, 4], where a difficult background appears, with a lost and a non-prompt or fake lepton. By now, tight cuts on the conventional relative isolation are applied, to reduce the non-prompt backgrounds. However, this approach decreases the signal efficiency too.
- "The search for supersymmetry with extremely compressed spectra (Compressed SUSY)" [5], where both isolation and relative isolation features do not have a sufficient discriminating power in terms of background leptons.
- "The search for $\mathrm{t\bar{t}}$+Z in the three and four lepton channel", where backgrounds add up, because of enumerative combinatorics with non-prompt and fake leptons. In connection with this analysis, the previously mentioned *TTV* lepton MVA was developed (Sec. 4.3.2.2).

The practical computation of the *DeepLepton* ID is treated in Ch. 6. The results of the *DeepLepton* classifier are presented in Ch. 7 and discussed in more detail in Ch. 8.



(a)

(b)

Fig. 4.13: Performance and architecture of the *DeepIsolation* neural network from [110]. The ROC curve **(a)** of the newly trained *DeepIsolation* feature compared to the standard $I_{\text{rel.}}$ (`RelIso`) and a BDT. The network architecture **(b)** utilized in the *DeepIsolation* study.

Chapter 5

Deep learning in particle physics

This chapter gives an introduction into deep learning and its applications in particle physics. First, the basic terminology is discussed. Then, the common components of a deep neural network are illustrated. Finally, a selection of different example use cases for deep learning in particle physics is provided.

5.1 Deep learning - Overview

Nowadays the terms artificial intelligence (AI), machine learning and deep learning are often mixed up. While AI describes a general concept of a non-human intelligence of machines, or computers, machine learning refers to a practical approach to acquire narrow AI, by analyzing data with various techniques. A distinct subset of advanced algorithms inside the machine learning tools is then called deep learning [111].

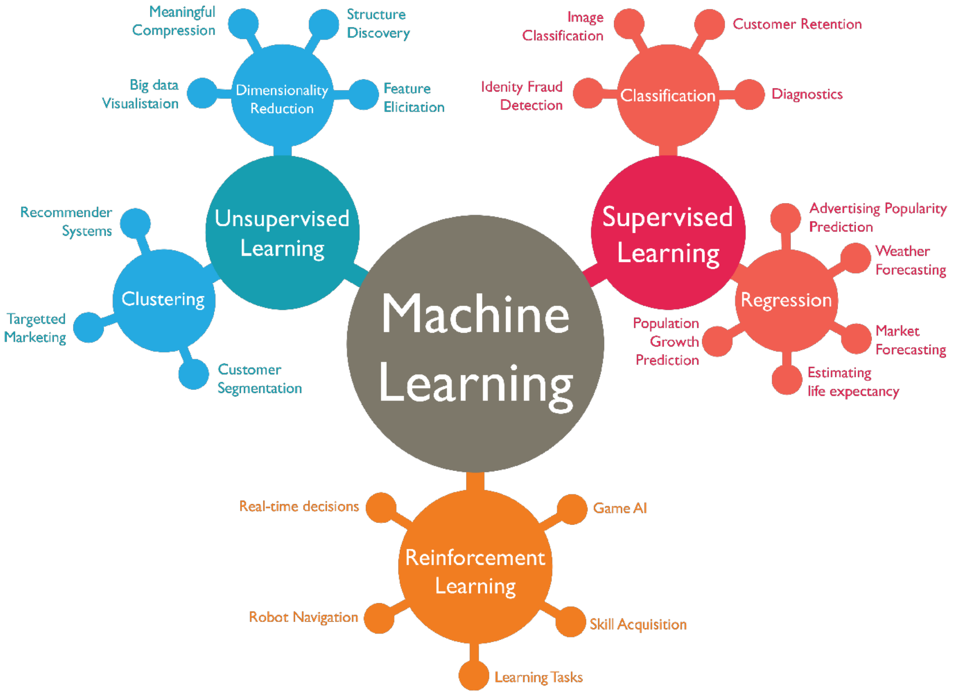


Fig. 5.1: Different domains in machine learning [112].

Machine learning is divided in three main areas [113], which are shown in Fig. 5.1. Firstly, supervised learning can be applied for classification tasks, or forecasts. Therefore the data has to consist of input-output pairs $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is a set of input parameters and $\boldsymbol{y}$ is the desired and known output, that should be learned by the respective machine learning algorithm. Examples are the classification and discrimination of images of birds and planes, or the weather forecast, based on prior weather data. Secondly, unsupervised learning is deployed in order to discover new structure and reveal information from unlabeled data. In this case, unlabeled means that the respective output of a distinct input is unknown for the data, so that no input-output pairs can be provided. An application of unsupervised learning is the finding of clusters within some unknown data, which may hint at a class structure of the studied data. Thirdly, the reinforcement learning is used to develop systems, that can acquire and improve skills, by interacting with their environment. This approach is related to the supervised learning, but is based on a reward system. Instead of making a correct classification, a reward function indicates the goodness of a certain action of the system. Hence, the system learns a proper behavior via trial and error. A popular example of reinforcement learning is its usage in the development of self-driving cars [114].

Typical machine learning techniques are random forests [115] and boosted decision trees (BDTs) [116], artificial (deep) neural networks [117], kernel density estimation [118], support vector machines [119], and genetic algorithms [120]. Within high energy physics (HEP) these algorithms are also called multivariate analysiss (MVAs), as several variables are considered simultaneously for the learning process [121]. Deep learning [111] represents a subdomain of the algorithms utilized in machine learning. It is commonly based on so-called deep neural networks, composed of a sequence of different types of neural networks, like convolutional neural networks (CNNs), recurrent neural networks (RNNs) and ordinary FC, or dense, neural networks. An introduction into deep neural networks (DNNs) is given in the following section.

5.2 Deep neural networks

Modern DNNs consist of different modules with distinct functions. The basic components, which are important for the understanding of the DNN as applied in this master thesis, are outlined in the following subsections, mainly inspired by Ref. [111, 121].

5.2.1 Fully connected neural networks

A neural network is commonly used in the supervised learning approach, in order to find a function $f(\boldsymbol{x})$, that maps best a phase space of a set of input parameters $\boldsymbol{x}$ into a lower-dimensional space of labeled output categories $\boldsymbol{y}$. In other words, a neural network is a function approximator. The approximated function $f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{y}_{\text{pred.}}$ is therefore related to some model parameters $\boldsymbol{\theta}$. The optimal configuration of $\boldsymbol{\theta}$ is tried to achieve via an iterative learning process, called training. To this end, a loss function $L(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{\theta}))$ is defined, which expresses a measure for the difference between the predicted output of the neural network $\boldsymbol{y}_{\text{pred.}}$ and the actual true output $\boldsymbol{y}$. As a consequence, the best possible parameter setting of the respective model is reached at the minimum of the loss function.

A neural network is composed of so-called layers with a certain amount of nodes, or units. Each node is connected to all nodes of the previous layer via weights, which is referred to as “fully connected”. The first and the last layer, is called input and output layer, respectively. The intermediate layers are referred to as hidden layers. The values of the nodes of the i-th of N hidden layers in the neural network are given by a vector $\boldsymbol{h}_i$. Additionally, the conditions $\boldsymbol{h}_0 \equiv \boldsymbol{x}$ and $\boldsymbol{h}_{N+1} \equiv \boldsymbol{y}_{\text{pred.}}$ are fulfilled for the input and the output layer. The weights between two layers $\boldsymbol{h}_i$ and $\boldsymbol{h}_{i+1}$ are represented by the matrix W_i. Thus, the predicted outputs of a neural network $\boldsymbol{y}_{\text{pred.}}$ are a result of a series of transformations

$$\boldsymbol{h}_{i+1} = g_i(W_i \boldsymbol{h}_i + \boldsymbol{b}_i), \tag{5.1}$$

which map the states of the nodes in layer $\boldsymbol{h}_i$ into the states in $\boldsymbol{h}_{i+1}$, where g_i is some function and $\boldsymbol{b}_i$ are the biases.

Shallow neural networks have just one hidden layer, while such networks with many hidden layers are called deep. Although it has been shown that a shallow neural network can approximate any function [122], they have practical disadvantages against DNN. On the one hand they need a large number of nodes in the single hidden layer and on the other hand they can fail in finding an appropriate function, especially when high-dimensional data sets are analyzed. Furthermore, DNNs have the potential to clearly outperform neural networks with just one hidden layer [123], and are therefore preferred for many state of the art applications.

An issue of DNNs that increases with the number of hidden layers is the so-called vanishing gradient problem [124]. As already mentioned, a neural network is trained by optimizing the loss function. This task requires the calculation of the gradient of the of loss function $\nabla_\theta L(\boldsymbol{y}, f(\boldsymbol{x};\boldsymbol{\theta}))$ w.r.t the model parameters $\boldsymbol{\theta}$, meaning the weights W_i and the biases b_i. In practice, the determination of the gradient is achieved by a backpropagation of the difference between the predicted $f(\boldsymbol{x};\boldsymbol{\theta}) = y_{\text{pred.}}$ and true outputs $\boldsymbol{y}$ through the layers of the neural network. This can be a difficult task if the gradient vanishes at some stage. Dropout [125], a regularization technique, as well as the usage of larger training samples, which has been enabled by the development of graphics processing units (GPUs), helped to deal with the vanishing gradient problem. Autoencoders [126], that generate abstract features from high-dimensional data sets, provide another solution for this problem. A standard technique in order to find the global minimum of the loss function in a neural network, is the gradient descent method [127].

In Eq. (5.1), g_i refers to a distinct function, which is also known as activation function. There are various activation functions available for the usage within a neural network [128]. Some examples are given in Fig. 5.2. A general requirement on these functions is their nonlinearity. This enables the solution of nonlinear problems through neural networks. Furthermore activation functions should be monotonic and continuously differentiable, in order to achieve a proper convergence into a global minimum of the loss function. Although the sigmoid and the tanh activation functions feature all three properties, they struggle with the vanishing gradient problem. This is due to their nearly horizontal curve shape at higher values of the argument $|x|$, illustrated in the related graphs in Fig. 5.2. The rectified linear unit (ReLU) partially overcomes this problem, but there are still difficulties for argument values below zero, since in this region the gradient of this activation function is 0. Hence, nodes in the neural network become inoperative, once their value is zero. This phenomenon is called the dying ReLU problem [129]. Modifications of the ReLU activation function, like e.g. the leaky ReLU or the exponential linear unit (ELU), provide a solution for this issue.

Another type of activation function, commonly applied in the output layer of neural network, is the so-called softmax function. It is utilized especially in multi-classification tasks, in order to indicate the probability p_i for each of the N output classes.

$$p_i = \frac{\exp(y_{\text{pred.},i})}{\sum_j^N \exp(y_{\text{pred.}j})} \tag{5.2}$$

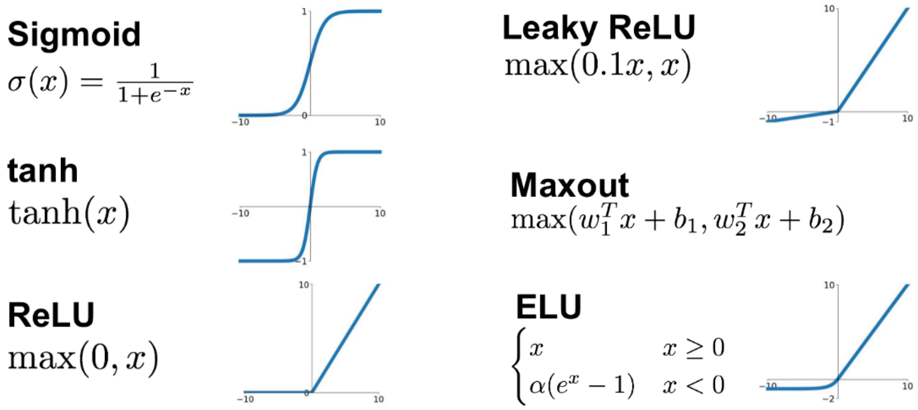


Fig. 5.2: Example of activation functions and their graphs [130].

The softmax function returns output values p_i between 0 and 1, whereby the probabilities add up to 1 ($\sum_i^N p_i = 1$).

The performance of a neural network and its training time are highly dependent on the settings of the hyper-parameters [131] of the training model, like batch size, learning rate, or momentum. A neural network usually is not able to process the whole training sample at once, because of a large bulk of data. Therefore the training data has to be split up into smaller subsets, called batches, with a certain batch size. First, the weights W_i between the nodes and the biases $\boldsymbol{b}_i$ are initialized with random values. Then, they are updated in an iterative manner after each batch and the loss function is calculated. The gradient of the loss function is utilized to set the direction of the parameter update for the next iteration. If all batches have passed the network, one epoch has finished. The step size, which is taken for an update of the weights and biases is given by the learning rate. If a neural network gets stuck in a local minimum of the loss function, an alternative step size is chosen to escape the local minimum. This step size is referred to as momentum. The learning rate and the momentum are closely related. It is a common practice to use a rather small learning rate and a higher momentum, which can be adjusted cyclically [131]. The goal of the neural network is to optimize the loss function until it converges, ideally into the global minimum.

One crucial requirement on a neural network is generalization. This means, that the trained model performs similar on both, the training data, and some test data set, which was not used during the training procedure. Overtraining, or overfitting describes a problem, where the model yields much more better results, when it is applied on the training data, than on the test data. In this case the neural network is too powerful, so that is has learned some detailed information about the training data, which has no general discriminating power, in terms of the desired output classes. Thus, the performance on the test data set is significantly worse. Besides the reduction of the network size (number of layers and nodes) and the increment of the training data, so-called regularization techniques can help to prevent a neural network from overfitting. One method to address this problem, is the deployment of dropout [125]. Here, the idea is to blind out randomly a certain percentage of nodes. This effects an equal optimization of the weights of all nodes in the neural network, instead of focusing just on a few major nodes.

Another problem emerging in neural networks is, that the distribution of the input data changes from layer to layer inside the network. This issue is also referred to as internal covariate shift and can be solved by a technique called batch normalization [132]. Higher learning rates can be achieved, by the normalization of the batch data after each layer, and usage of dropout may become dispensable. Therefore batch normalization is applied as another regularization tool.

5.2.2 Recurrent neural networks

In ordinary dense neural networks, as described in the the previous Sec. 5.2.1, the information flow through the network is straightforward: form the input $\boldsymbol{x}$, via the nodes of the hidden layers $\boldsymbol{h}_i$, to the predicted output $\boldsymbol{y}_{\text{pred.}}$. Therefore these network models are also called feedforward [133]. However, there exist also models, which are based on the idea to feed back the model output as an input again, in an iterative manner. These, so-called recurrent neural networks (RNNs) [134] are very useful when processing sequential data with a variable length, like e.g. words in a sentence, chronological weather data in a timeline, or a list of particles emerging in the vicinity of another particle of interest, as in this master thesis.

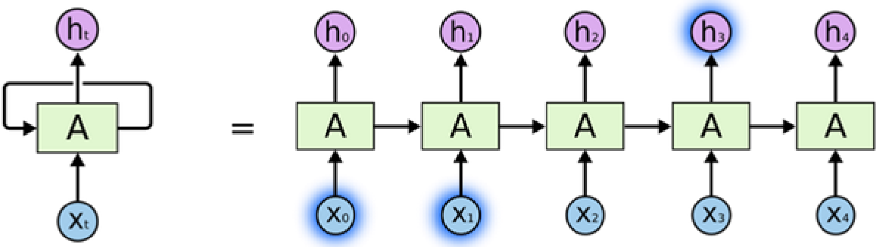


Fig. 5.3: The loop structure of RNNs allows to keep information inside the neural network, when processing sequential data. The image is adapted from [135].

The goal of RNNs is to keep the information of an element, that once has occurred in the sequence, inside the neural network. While the residual sequence is processed, the acquired information can serve as an auxiliary feature for the elements, occurring later in the sequence. The loop structure of RNNs enables such neural networks to develop a kind of memory (Fig. 5.3). The problem of simple RNNs is, that they perform good for short-term dependencies in sequential data, but fail to discover long-term dependencies. In other words, the acquired information gets lost after a certain number of loops (Fig. 5.4). Furthermore, simple RNNs can suffer from vanishing or soaring gradients

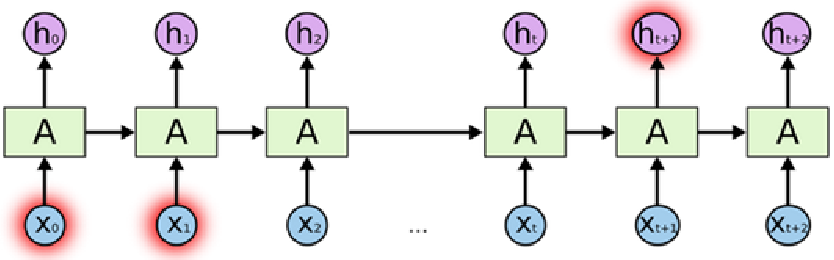


Fig. 5.4: Simple RNNs fail to connect the information in long-term dependencies [135].

Special units inside advanced variants of RNNs are using a kind of gating technique, in order to control the information flow through the network and thus, overcome the problems of simple RNNs. Examples for units inside so-called gated RNN [136] are Long Short Term Memory (LSTM) units [137] and gated recurrent units (GRUs) [138]. LSTM RNNs solve the problems with vanishing or soaring gradients, and also address the issue of learning long-term dependencies in sequential data. An illustration of a LSTM unit U_t^l in the l-th layer and the t-th sequential (time) step within a RNN is shown in Fig. 5.5. The specific about LSTM units is, that they have two output pipes. On the one hand, the activation of the hidden unit h_t^l is calculated. This ordinary output h_t^l subsequently serves as the input for both, the unit of the next layer (and the same sequential step) U_t^{l+1}, and the unit of the next sequential step within the same layer U_{t+1}^l. One the other hand, an additional feature, the so-called cell state C_t^l is determined, which represents a kind of memory line, that horizontally flows through the sequential steps of the network.

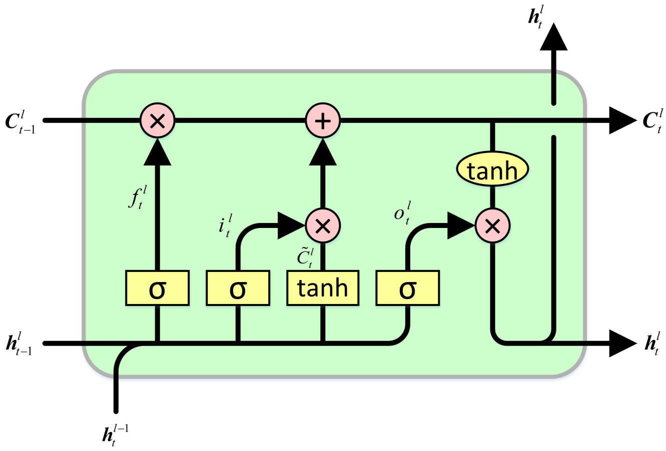


Fig. 5.5: The information flow inside a LSTM unit is controlled by several gates. The image is adapted from [139].

Hence, a LSTM unit has three input pipes: the cell state C_{t-1}^l and the output h_{t-1}^l of unit of the previous time step, as well as the output h_t^{l-1} of the unit in the preceding layer, where h_t^0 is given by the t-th element x_t of the input sequence. Inside a LSTM unit, the information flow is regulated by four gates in total [113]. First, a sigmoid function in the forget gate f_t^l decides, to which extend the information of the previous cell state C_{t-1}^l is relevant for the actual cell. The forget gate is also able to reset the cell state, in order to prevent an unlimited increase of the memory. Next, two input gates i_t^l and $\tilde{C}_t^l$ are responsible for the update of the current cell state C_t^l, by adding new memory to some extend. Finally, the output gate determines the output of the hidden unit h_t^l, taking account of the momentary cell state C_t^l.

5.2.3 Convolutional neural networks

Another type of neural network, originally developed for the recognition of handwritten digits, is the so-called convolutional neural network (CNN) [140]. CNNs are designed for the processing of such data, that features a grid-like structure [113]. Popular applications are image data, where the image pixels represent a 2D grid, but also sequential data, which can be seen as a 1D grid structure. Therefore, CNNs have become a very useful tool in the field of supervised learning, especially for image recognition and classification. In the following the basic concept of CNNs is outlined according to [141].

The architecture of a CNN is composed of several layers. Usually a structure of alternating convolutional and pooling layers is chosen. Therefore, the basic algorithm of CNNs, explained with the example of image processing, can be separated into two main steps, according to the function of the two different layers. Firstly, the image is split into smaller, overlapping, sub-images and the same so-called filter is applied on each sub-image. In other words, the function of a convolutional layer is to screen the image with a kind of filter, that has a certain kernel size. Thus, the output of a convolutional layer $\boldsymbol{h}_i$ is an array again, which is calculated by means of its input array $\boldsymbol{h}_{i-1}$, a non-linear activation function g_i and an linear operator W_i, that represents the filter.

$$\boldsymbol{h}_i = g_i W_i \boldsymbol{h}_{i-1} \tag{5.3}$$

Rectifiers, like e.g. the ReLU, or a sigmoid function are commonly used as an activation function for such convolutional layers. The linear operator W_i can be understood as a stack of (convolutional) filters. Hence, the output of a convolutional layer can be considered as a sum of convolutions of its input and the filters. The same linear operator, or filter stack, is applied to each sub-image, within a convolutional layer. This causes a translation invariance of CNNs.

The second step inside a CNN is called sub-sampling or pooling. Its purpose is to downsize the output array produced by a convolutional layer. To this end, usually the so-called max-pooling algorithm [142] is applied. This technique divides the output array into e.g. 2 × 2 squares and keeps only the most significant value, instead of calculating the mean of the four values. This helps to reduce the data size, but retains the important information. Another advantage of the pooling layers is, that a subsequent convolutional layer can learn to extract higher-level features from the downsized array. For example, a first convolutional layer may identify edges and curves in the image data, while a second convolutional layer, after a pooling step, may possibly discover circles or squares, and so on. The procedure can be repeated, as often as desired.

In terms of supervised learning, in general a conventional FC (or dense) neural network is deployed, for a further processing of the information, provided by the CNN. An example architecture of a DNN containing a CNN is illustrated in Fig. 5.6.

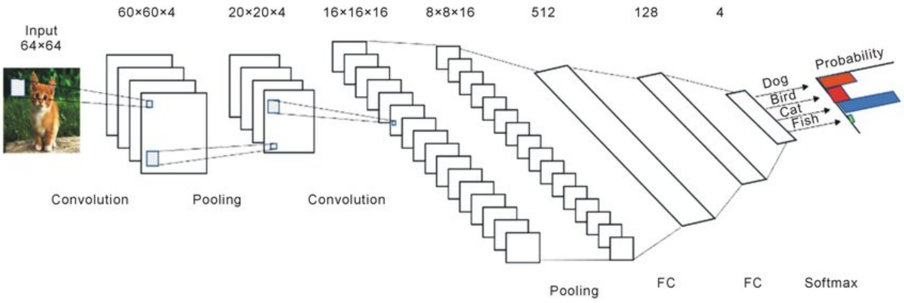


Fig. 5.6: Example architecture of a DNN consisting of a CNN and a subsequent FC neural network. First, an input image is processed by alternating convolutional and pooling layers. Then, the output of the CNN is fed into a dense neural network with two fully connected layers. Finally, the output layer indicates the probabilities for the defined output classes [143].

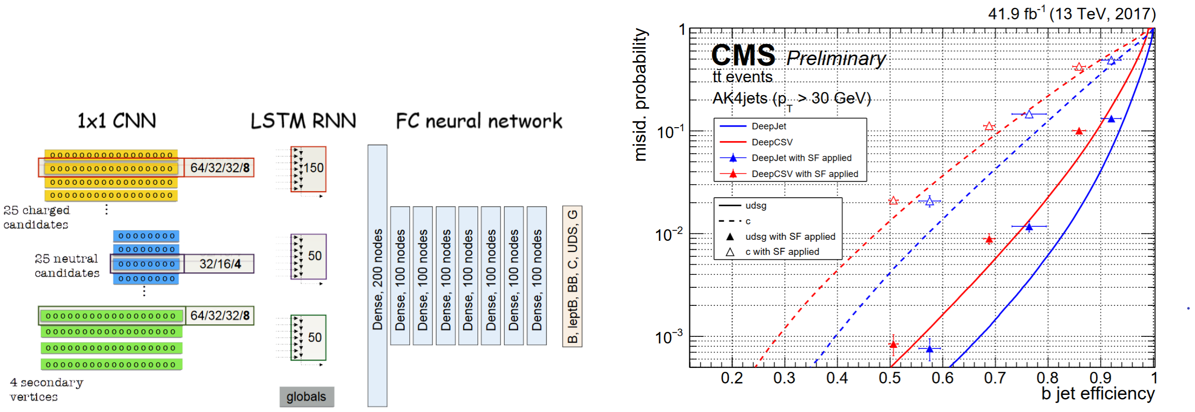
5.3 Applications in particle physics

Machine learning has a wide range of applications in particle physics. At the LHC, various classical machine learning algorithms are deployed at different stages [144]: starting with the detector simulation in the generation of MC data and the trigger decisions in real detector data, via the object reconstruction and identification, right through to physics analysis. The more advanced deep learning techniques boomed in 2012 [121], when several DNNs yielded astonishing results in image classification and recognition [145–147]. Moreover, the technological progress of e.g. GPUs, enabled to train large neural networks with an increased amount of data.

This boom established the development and improvement of applications in HEP, particularly in the field of object reconstruction and identification, and for physics analysis. Deep learning has proofed its strengths e.g. in classification tasks with difficult signal to background ratios, where the performances shallow neural networks and other machine learning techniques are limited. Related to the physics at the LHC, a study [148] proofed, that DNNs using lower-level data clearly outperform shallow neural networks, that make use of higher-level features, like e.g. the reconstructed invariant mass. In the following, some examples for the application of deep learning techniques, at different stages in HEP, are introduced.

5.3.1 Jet identification - The *DeepJet* framework

A parade example for a DNN is the architecture, which was utilized in the so-called *DeepJet* framework [6, 149]. It is designed for jet-flavor tagging and multiclassification, via a supervised learning approach. In this case, all three neural network components, which were previously introduced, are applied. A schematic illustration of the network architecture is shown in Fig. 5.7a.



(a) (b)

Fig. 5.7: The *DeepJet* neural network. The Network architecture **(a)** used in the *DeepJet* framework for jet classification [150] and the performance of the *DeepJet* neural network **(b)** against the standard DeepCSV classifier, visualized by a ROC curve of the b-tagging efficiency [151].

First, 1D CNNs are transforming and compressing lists of particles and SVs, that are assumed to be constituents of a jet. Then, LSTM RNNs process the variable length particle sequences, in order to discover jet discriminating features from the input data. Finally, these intermediate features, together with some global jet variables, are fed into a conventional FC neural network, which outputs the predicted probabilities for the following jet classes: b-jet, double b-jet, leptonic b-jet, c-jet, uds-jet and gluon-jet. In Fig. 5.7b the b-tagging performance of the *DeepJet* neural network is compared with the DeepCSV algorithm [108], the standard tool for jet classification in the CMS experiment.

5.3.2 Track reconstruction - The HEP.TrkX project

The HEP.TrkX project [152] is an ongoing collaboration about the reconstruction of particle tracks in HEP. The aim is to develop machine learning based tracking algorithms for experiments, where densely packed and overlapping particle tracks, as well as noise are a serious problem. Hence, at collider experiments, such as the LHC, a proper track reconstruction can be a complex task, due to a high collision energy and instantaneous luminosity.

In general track, reconstruction is very computing intensive, when applying traditional reconstruction algorithms based on Kalman or Gaussian filters. However, the connection of single pixel hits into a reconstructed track, can be understood as a classification task too. In this case the training model has to discriminate matching from non-matching hits. First approaches, that address this issue, make use of 2D or 3D CNNs. To this end, the input tracking data, has to be provided in an image-like format of the pixel hits in the different layers of a detector. However, these models show weaknesses at the level of realistic data with high track multiplicity, because of the high complexity and dimensionality of the data. By contrast, a 1D sequence of the pixel hits in the detector, is another way to look at the tracking data. In this case LSTM RNNs are the optimal choice for the processing of this sequential data, with a variable length. Special next-layer LSTM [152] models are applied, in order to make forward predictions for the positions of possible hits in the next layer, without processing the data of the respective layer apriori (Fig. 5.8). These RNN-based algorithms are competitive with state of the art reconstruction techniques, which are using Kalman filters (Sec. 3.4).

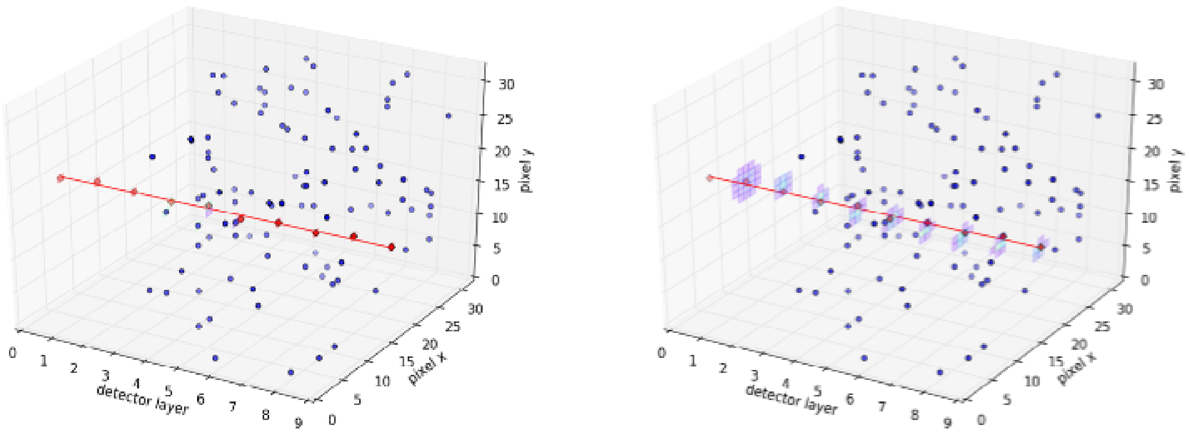


Fig. 5.8: Toy Example of 3D track reconstruction using LSTM units. The pixel hits in the different detector layers are linked by a simple LSTM RNN **(left)** and a predictive next-layer LSTM model **(right)**, that makes a forecast for an area containing potential pixel hits in the next detector layer [152].

A third way to represent the tracking data, is via a graph of points, that are connected in a plausible way [153], as illustrated inFig. 5.9a. The pre-linking algorithm can be deployed by considering some criteria for the pixel hits, like e.g. geometric constraints. After this preprocessing step, so-called graph neural network (GNN) from geometric deep learning [154] are utilized, to extract information from the nodes and edges in the graph, as well as from the global state of the graph. More weight is given to connections between matching pixel hits, than to non-matching hits, in iterative learning steps (Fig. 5.9b). First results of this study are promising, but there are still some challenges to proof this algorithm for realistic tracking data.

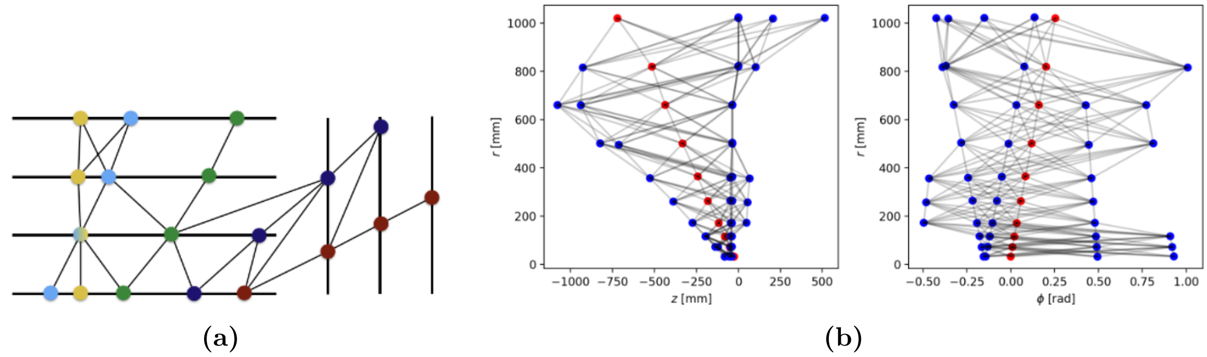


Fig. 5.9: Track reconstruction with geometric deep learning. Simple example of reasonably connected pixel hits **(a)** in the different detector layers, according to certain criteria, like geometry constraints [153]. Toy example for the classification of pixel hits **(b)** in a graph of connected points in the r-z and r-ϕ plane. Red nodes indicate matching pixel hits for a distinct track, while blue nodes a excluded by the model [153].

5.3.3 Fast data simulation - Generative adversarial networks

Deep learning can be used not only to analyze data, but also to simulate data. MC event generators (Sec. 3.5), which are traditionally applied to model the particle interactions, consume a lot of computing resources and are very time intensive. An approach, that could speed up these computations, is the so-called generative adversarial network (GAN) [155].

In a GAN, two competing networks are trained simultaneously (Fig. 5.10). One network outputs fake data via an generative model G, that tries to imitate the distribution of some original data. An adversarial second network creates a discriminative model D, which serves to classify the two categories: the original data and the data, produced by the model G. The goal is to continue the training process, until model D is not able to differentiate the generated data from the original data any more.

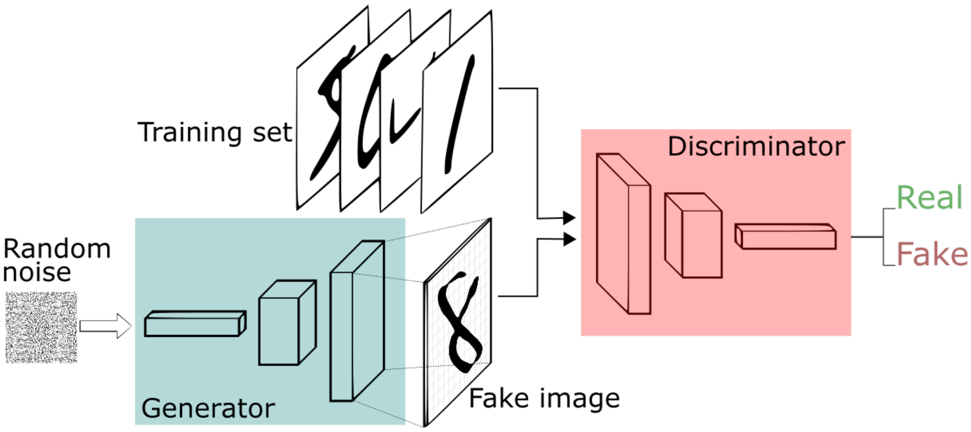


Fig. 5.10: Principle of a GAN. The generator network creates fake data from a random input and tries to fool the discriminator network, which is classifying the real and fake data [156].

An example, for the usage of GANs at the LHC, is the simulation of electromagnetic showers within a calorimeter, consisting of several layers [157]. The framework yields high-fidelity ECAL energy deposits and could tremendously speed up the computations, compared to traditional models. However, an acceptable accuracy could not be provided for the whole phase space, so far. Another similar application of GANs is the generative modeling of jet images. Jet images are calorimeter energy deposits of jet particles, which are visualized as a 2D picture in the η-ϕ plane. These generated images are comparable over a large scale to the ones, that are traditionally created by MC event generators, like Pythia (Sec. 3.5). The central processing unit (CPU) computing time of GAN produced images could be reduced by a factor of about 14× against the images, which are produced by Pythia. An illustration of average jet images of both approaches is shown in Fig. 5.11.

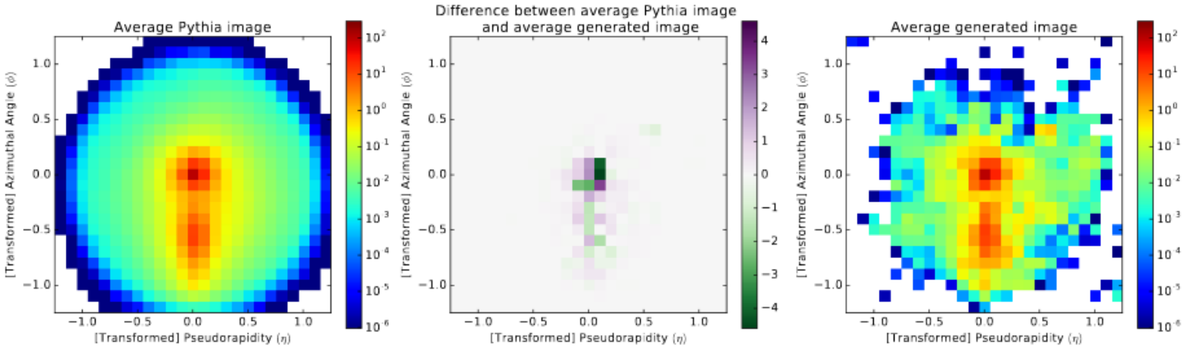


Fig. 5.11: Example jet images, which are produced by Pythia **(left)** and a GAN **(right)**, plus the difference of the two representations **(middle)** [156].

PART II: Experimental

Chapter 6

Materials and methods

This chapter covers the practical implementation of the master thesis, which is about the computation of a new kind of lepton IDs, based on advanced machine learning techniques. It is referred to as *DeepLepton* ID in the following, since the conception is heavily inspired by the network architecture, which is applied in the *DeepJet* framework (Sec. 5.3.1) and the *DeepIsolation* study (Sec. 4.3.2.4).

First, the generation of the training data set is described. Next, the central part of the work, the network architecture and input features, as well as the actual training procedure is outlined. The third and last section treats the evaluation of the training model on MC data and comparisons of the MC and detector data.

6.1 Processing of training data

This section outlines the steps, which are necessary, in order to generate the appropriate training data set for the later training within the *DeepLepton* framework.

As input data, MC samples from the 13 TeV *RunIISummer16MiniAODv2* campaign are used. Unlike the real detector data, these data samples contain information about the generated particles. By means of this information, leptons are identified as prompt, non-prompt or fake, according to Sec. 6.2.2.

6.1.1 Sample selection

In a first attempt, a combination of generated DY and QCD samples is used as a training data set. The prompt leptons are taken from DY processes and the non-prompt and fake leptons are obtained from simulated QCD multijet processes. The Z bosons from the DY process can be measured with a high precision, at the LHC. Furthermore, it has a very clean final state and a particular signature, due to a resonance, of the invariant mass of the two final state leptons $m_{\ell^+\ell^-}$ with the mass of the Z boson m_Z [74, 77]. Therefore, it is considered as an appropriate choice for the initial training sample. A full list of the 2016 MC samples, which are utilized within this training sample, is given in Tab. 6.1. It is referred to as the DY+QCD training data set.

A further training focuses on the $\mathrm{t\bar{t}}$ process, for e.g. $\mathrm{t\bar{t}}$+Z in the three and four lepton channel, or the stops-dilepton search [3, 4] and other SUSY scenarios (Sec. 4.3.2.5). For this training all leptons are taken from simulated $\mathrm{t\bar{t}}$ processes from samples in Tab. 6.2. In this case, no separate samples for the three lepton classes are defined.

Tab. 6.1: 2016 DY+QCD training data set

Short Name	Sample Name	σ_{eff}	Lepton Class
DY1JetsToLL_M50_LO	DY1JetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	1012.5	prompt
DY2JetsToLL_M50_LO	DY2JetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	332.8	prompt
DY3JetsToLL_M50_LO	DY3JetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	101.8	prompt
DY4JetsToLL_M50_LO	DY4JetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	54.8	prompt
QCD_Pt15to20_Mu5	QCD_Pt-15to20_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	3819570.0	non-prompt, fake
QCD_Pt20to30_Mu5	QCD_Pt-20to30_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	2960198.4	non-prompt, fake
QCD_Pt30to50_Mu5	QCD_Pt-30to50_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	1652471.5	non-prompt, fake
QCD_Pt50to80_Mu5	QCD_Pt-50to80_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	437504.1	non-prompt, fake
QCD_Pt80to120_Mu5	QCD_Pt-80to120_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	106033.7	non-prompt, fake
QCD_Pt80to120_Mu5_ext	QCD_Pt-80to120_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	106033.7	non-prompt, fake
QCD_Pt120to170_Mu5	QCD_Pt-120to170_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	25190.5	non-prompt, fake
QCD_Pt170to300_Mu5	QCD_Pt-170to300_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	8654.5	non-prompt, fake
QCD_Pt170to300_Mu5_ext	QCD_Pt-170to300_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	8654.5	non-prompt, fake
QCD_Pt300to470_Mu5	QCD_Pt-300to470_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	797.4	non-prompt, fake
QCD_Pt300to470_Mu5_ext	QCD_Pt-300to470_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	797.4	non-prompt, fake
QCD_Pt300to470_Mu5_ext2	QCD_Pt-300to470_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	797.4	non-prompt, fake
QCD_Pt470to600_Mu5	QCD_Pt-470to600_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	79.0	non-prompt, fake
QCD_Pt470to600_Mu5_ext	QCD_Pt-470to600_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	79.0	non-prompt, fake
QCD_Pt470to600_Mu5_ext2	QCD_Pt-470to600_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	79.0	non-prompt, fake
QCD_Pt600to800_Mu5	QCD_Pt-600to800_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	25.1	non-prompt, fake
QCD_Pt600to800_Mu5_ext	QCD_Pt-600to800_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	25.1	non-prompt, fake
QCD_Pt800to1000_Mu5	QCD_Pt-800to1000_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	4.7	non-prompt, fake
QCD_Pt800to1000_Mu5_ext	QCD_Pt-800to1000_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	4.7	non-prompt, fake
QCD_Pt800to1000_Mu5_ext2	QCD_Pt-800to1000_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	4.7	non-prompt, fake
QCD_Pt1000toInf_Mu5	QCD_Pt-1000toInf_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	1.6	non-prompt, fake
QCD_Pt1000toInf_Mu5_ext	QCD_Pt-1000toInf_MuEnrichedPt5_TuneCUETP8M1_13TeV_pythia8	1.6	non-prompt, fake

Tab. 6.2: 2016 $t\bar{t}$+Jets and $t\bar{t}$ training data set

Short Name	Sample Name	σ_{eff}	$t\bar{t}$+Jets	$t\bar{t}$
TTJets_DiLepton	TTJets_DiLept_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	87.3	yes	yes
TTJets_DiLepton_ext	TTJets_DiLept_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	87.3	yes	yes
TTJets_SingleLeptonFromTbar	TTJets_SingleLeptFromTbar_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	182.2	yes	yes
TTJets_SingleLeptonFromTbar_ext	TTJets_SingleLeptFromTbar_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	182.2	yes	yes
TTJets_SingleLeptonFromT	TTJets_SingleLeptFromT_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	182.2	yes	yes
TTJets_SingleLeptonFromT_ext	TTJets_SingleLeptFromT_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	182.2	yes	yes
TTJets	TTJets_TuneCUETP8M2T4_13TeV-amcatnloFXFX-pythia8	831.8	no	yes
TTJets_LO	TTJets_TuneCUETP8M1_13TeV-madgraphMLM-pythia8	831.8	no	yes
TTLep_pow	TTTo2L2Nu_TuneCUETP8M2_ttHtranche3_13TeV-powheg-pythia8	87.3	no	yes
TTSemiLep_pow	TTToSemilepton_TuneCUETP8M2_ttHtranche3_13TeV-powheg-pythia8	364.4	no	yes
TT_pow	TT_TuneCUETP8M2T4_13TeV-powheg-pythia8	831.8	no	yes
TT_pow_ext3	TT_TuneCUETP8M2T4_13TeV-powheg-pythia8	831.8	no	yes

Concerning supervised learning, usually a larger training data sample improves the performance of the training model, in terms of generalization. In order to further increase the amount of the training data, a combined sample, which consists of all the previous mentioned 2016 MC samples, is produced too. Apriori it is not clear, whether merging the DY+QCD and $t\bar{t}$ samples into one big training sample, has an impact on the performance of the training model. The dependence on the different training samples is discussed in Sec. 8.1.7.

6.1.2 Input data processing with CRAB

The input data samples are processed via the so called CMS Remote Analysis Builder (CRAB) utility [158]. CRAB does not only provide access to the CMS detector data and MC data, but also enables to utilize CPUs and storage resources of computing sites linked to the CMS experiment.

Lepton identification represents a typical classification task in supervised learning. In order to train the *DeepLepton* classifier, single leptons instead of full events are required, regarding the training data. Hence, unlike the common terminology, a single lepton is considered as an event, when producing the input data samples with CRAB. The samples are processed as files of flat n-tuples of leptons, containing the lepton features. The files are stored in the ROOT [159] file format, which is a machine-independent compressed binary format. It is designed to store and handle a huge amount of data, without the dependence on evolving software infrastructure.

The central approach of the *DeepLepton* classifier, is to consider low-level information of the vicinity of the leptons, for the classification task. Thus, relevant features of the PF candidates and potential SVs, close to the respective leptons, are included in the sample files. For this purpose, ΔR is chosen as a measure for the vicinity. PF candidates, within a cone of $\Delta R < 0.5$ around a respective lepton, and SVs, within a cone of $\Delta R < 0.6$, are selected. The different PF candidate species (neutral and charged hadrons, photons, electrons and muons) are listed separately and the respective lepton itself is excluded. The low-level PF candidate and SV variables, which are utilized as input features, in addition to the lepton features, are given in Tab. 6.6 and are discussed later on in Sec. 6.2.1.

Besides the training input variables, the true output class, which needs to be provided in terms of supervised learning, is stored in the sample files too. The true output, meaning the actual lepton class, is represented by the information of the origin of the final state leptons, which is available in the MC data. This information is also referred to as MC truth. Based on the MC truth, each lepton is assigned to one of the lepton classes: prompt, non-prompt and fake (See 6.2.2).

When processing the input data samples, a preselection is made on the leptons. First, a cut on the relative isolation of the leptons $I_{\mathrm{rel.}}(\Delta R \leq 0.3) < 0.5$ is applied. Second, only leptons which are fulfilling $p_{\mathrm{T}} > 5$ GeV are taken into account.

6.1.3 From input data to training data

After the input data samples are processed, another few working steps are necessary, to provide the training samples in an appropriate data structure, for the training of the *DeepLepton* neural network. The whole training data preprocessing procedure is discussed in the following sections. An overview of the various working steps is illustrated in Fig. 6.1.

6.1.3.1 Data shuffling and balanced classes

During the training, usually a neural network is not able to process the whole training data set at once, due to a large bulk of data. Hence, the training data is split into smaller sub samples, so called batches, which are fed into the neural network one after another. After each batch, the weights within the neural networks are updated (5.2). In terms of generalization, the data of a single batch should ideally represent the distribution of the entire training data. Therefore, a well shuffled training sample is essential to avoid biased batches, containing only a certain phase space of the training data.

Another important issue, is a balanced ratio of the true output classes in the input training data. Imbalanced data sets are considered as problematic in the machine learning community and should be avoided [160].

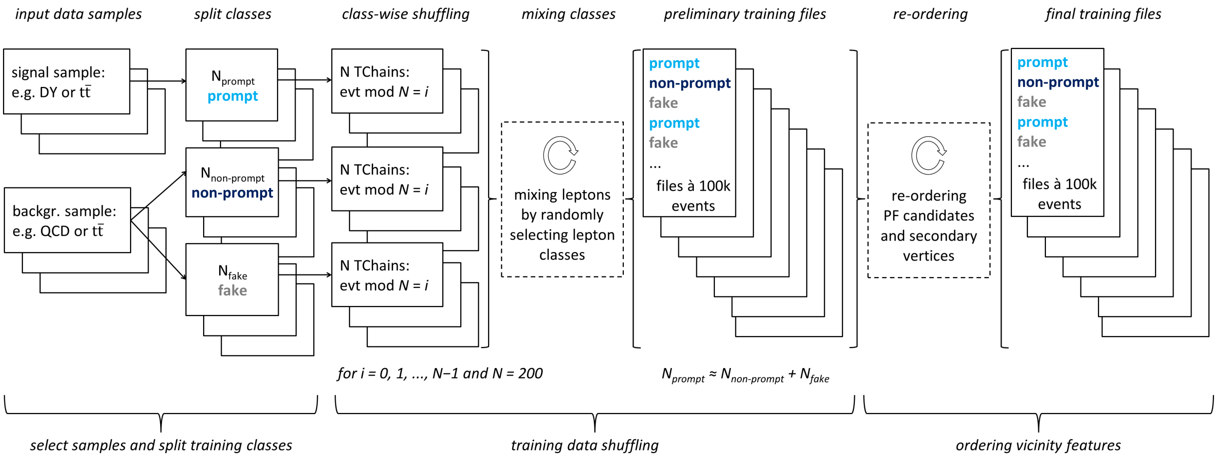


Fig. 6.1: The single steps in the training data preprocessing.

In terms of the preprocessing of the training data, which is utilized for the *DeepLepton* classifier, the first step is to split up the leptons into the three classes, according to the MC truth information: prompt, non-prompt and fake. Then all leptons, that fulfill the criteria

$$\texttt{evt} \bmod N = i \qquad \text{for } i = 0, 1, ..., N-1, \tag{6.1}$$

regarding their event number (evt), are selected. They are buffered in N separate ROOT TChains, a ROOT class, that enables to chain several ROOT files.

This effected a kind of shuffling within each of the three categories, as the event number is uncorrelated to the lepton itself. Finally, the different classes are mixed together again, by selecting leptons successively from one of the categories. In this process the lepton classes are chosen randomly, but to ensure a balanced data set, the number of signal (prompt) and background (non-prompt, fake) leptons is considered to be approximately equal.

$$N^{lep}_{prompt} \cong N^{lep}_{prompt} + N^{lep}_{fake} \tag{6.2}$$

The reason for this rather unconventional style of data shuffling is that, ROOT files are optimized to be processed entry by entry, with an increasing entry number. Thus, a shuffling, by simply selecting random entries from the files, would be inefficient.

Electrons and muons are treated separately and in this work only muons are addressed. The number of muons, which are available for the training data set, after the above mentioned pre-processing steps, is shown in Tab. 6.3.

During the preprocessing of the training data set, a p_T selection is applied, too. Leptons with a transverse momentum $p_\mathrm{T} \geq 15$ GeV are considered for the initial DY+QCD sample and a p_T ≥ 5 GeV cut is chosen for the later $\mathrm{t\bar{t}}$ samples ($\mathrm{t\bar{t}}$+Jets and $\mathrm{t\bar{t}}$).

6.1.3.2 Particle flow candidates and secondary vertices ordering

The inclusion of the vicinity of the leptons in the training data, raises the question, if and how to order the PF candidates and the SVs per lepton. To process the information of the PF

Tab. 6.3: The size of the muon training data set, after the preprocessing steps. A balanced signal to background ratio is required for the samples.

Training Sample Name	N^{μ}_{total}	N^{μ}_{prompt}	N^{μ}_{prompt}	N^{μ}_{fake}
DY+QCD	6 718 784	3 359 927	3 043 701	315 156
training	5 375 989	2 688 424	2 435 548	252 017
evaluation	1 342 795	671 503	608 153	63 139
$\mathrm{t\bar{t}}$+Jets	6 213 194	3 108 750	2 845 990	258 454
training	4 972 818	2 488 108	2 277 700	207 010
evaluation	1 240 376	620 642	568 290	51 444
$\mathrm{t\bar{t}}$	26 620 073	13 303 086	12 255 741	1 061 246
training	21 324 472	10 658 812	9 815 211	850 449
evaluation	5 295 601	2 644 274	2 440 530	210 797

candidates and SVs within the *DeepLepton* neural network so called LSTM RNNs [137] are used. The performance of these special type of neural networks is closely related to the order of the sequential input (Sec. 5.2.2). Therefore, a reordering of the PF candidates and SVs per lepton, by a decreasing importance, is supposed to yield the best results.

Inspired by [110], the relative transverse momentum $p_{\mathrm{T}}^{\mathrm{rel.}}$ in relation to the associated lepton, is considered as a descending key for the PF candidates. This implies, that PF candidates, which are closer to the lepton, as well as higher in p_{T}, are more relevant, than the others. Regarding the SVs, the total p_{T} of all particles originating from the associated vertex, is selected as a discriminating feature for the importance of a SV.

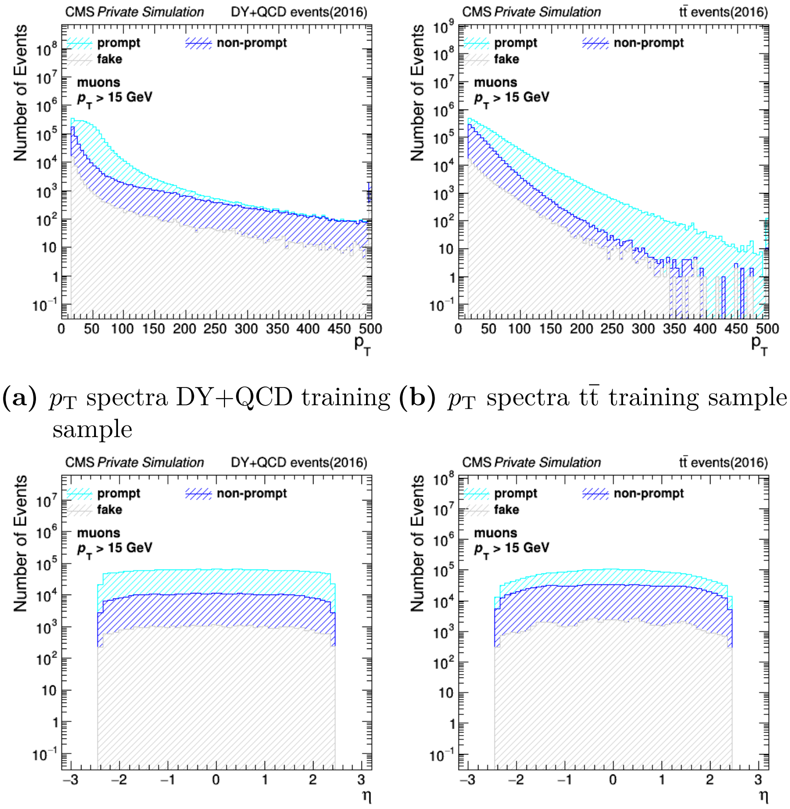
6.1.3.3 Reweighting in p_{T} and η

Concerning the training data set, prompt, non-prompt and fake leptons may not have the same p_{T} spectra (Fig. 6.2 a and b), due to different physical spectra, or the p_{T} binning of the QCD samples, which are utilized in the initial training sample (Tab. 6.1). The η spectrum of the three classes (Fig. 6.2 c and d) may vary too. A reweighting for both variables in the training data is applied, in order to prevent the DNN from learning any discriminating feature from the p_{T} and η distribution of the leptons. The bins, which are selected for the reweighting, are listed in Tab. 6.4.

Tab. 6.4: The bins for the p_{T} and η reweighting to a certain reference class in the training sample.

lepton variable	bins
p_{T}	5, 7.5, 10, 12.5, 15, 17.5, 20, 25, 30, 35, 40, 45, 50, 60, 75, 100, 125, 150, 175, 200, 250, 300, 400, 500, 600, 2000
η	-2.5, -2., -1.5, -1. ,-0.5 ,0.5, 1, 1.5, 2.,2.5

For each η and p_{T} bin and each class in the input training data, all leptons are reweighted to a certain reference class. Different configurations are tested. Initially, the prompt leptons are



(a) p_T spectra DY+QCD training sample **(b)** p_T spectra $\mathrm{t}\bar{\mathrm{t}}$ training sample

(c) η spectra DY+QCD training sample **(d)** η spectra $\mathrm{t}\bar{\mathrm{t}}$ training sample

Fig. 6.2: p_T **(a)-(b)** and η **(c)-(d)** spectra of prompt, non-prompt and fake leptons in various training data sets.

selected as a reference class by default. However, the non-prompt leptons are utilized in the advanced training models. The results, of these different settings, are discussed afterwards in Sec. 8.1.6. Practically, the p_T and η reweighting is applied within the data conversion step of the *DeepJet* framework, since a reweighting tool has been already implemented there.

6.1.3.4 Feature scaling and normalization

Another important preprocessing step, which is necessary to achieve a proper training data structure, is the normalization of the input features. To this end, the mean μ and the standard deviation σ are calculated per input feature. After that, the feature values are scaled by

$$x_{new} = \frac{x_{old} - \mu}{\sigma}, \tag{6.3}$$

where x_{old} and x_{new} refer to the old and the new feature value, of the respective lepton.

This procedure ensures, that the new rescaled features x_{new} follow a standard normal distribution, with $\mu = 0$ and $\sigma = 1$. The purpose of the feature normalization is, that it can speed up the training and prevent from getting stuck in a local minimum of the loss function (Sec. 5.2). Again, a tool for this task is provided, by the *DeepJet* framework in the data conversion step.

6.1.3.5 Splitting training and test data sample

Whenever a neural network is trained, there is the possibility of overfitting. In this case, the training model performs well on the training data, but not on some other related data, that has not been used during the training procedure. To address this issue, it is necessary to split the available data into a training sample and a test sample. The test sample is needed for the evaluation of the training model afterwards. To this end, 80 percent of the full data sample are considered for the training data set, while the remaining 20 percent are reserved for the evaluation (Tab. 6.3). In practice, the number of leptons may not fulfill precisely the previously mentioned ratio, since for reasons of simplicity the number of sample files ($\grave{a} \sim 10^5$ leptons) is divided into the two sub-samples, instead of the number of leptons.

6.2 Deep neural network training

This section addresses the training procedure of the *DeepLepton* neural network. The input features and output classes are outlined, as well as the architecture of the DNN and the settings of the training model. The further development and optimization of the network architecture has been a constantly ongoing process during this master thesis. For that reason, the *DeepLepton* neural network is introduced according to a reference network architecture, in the following sections. Slightly different variants, e.g. different activation functions, are tested. The results of these modifications are discussed afterwards in Ch. 7 and 8.

6.2.1 Input features

The central idea of the *DeepLepton* classifier, is to combine the information of the lepton and the particles within its vicinity, in order to provide a novel lepton ID, which is based on a deep neural network. Regarding the lepton itself, both common features, which have been used already in previous lepton identification approaches, like the *TTH* and *TTV* BDTs, and also other additional lepton features are considered. In this work only muons are addressed. A full list of the muon features, which are used for the *DeepLepton* training model, is shown in Tab. 6.5. Related histograms of the input training features can be found in the appendix.

Tab. 6.5: The lepton features utilized in the *DeepLepton* neural network training. The variables in parentheses are used only in initial training test, but are excluded from the input features in the final version of the *DeepLepton* neural network.

Type	Features
global features	p_T, η, d_{xy}, d_z, (uncertainty d_{xy}), (uncertainty d_z), d_{3D}, significance d_{3D}, inner track χ^2, inner track valid hit fraction, p_T error track, ρ, jet ΔR, tracker layers, pixel layers, tracker hits, lost inner hits, lost outer hits
isolation features	$I_{\text{rel.}}$, mini $I_{\text{rel.}}$ for charged hadrons, mini $I_{\text{rel.}}$ for neutral hadrons, (jet p_T ratio v1), (jet $p_T^{\text{rel.}}$ v1), jet p_T ratio v2, jet $p_T^{\text{rel.}}$ v2
high-level features	`jetBTagDeepCSV`
muon specific features	segment compatibility, muon inner track relative error, `isGlobalMuon`, χ^2 local position, χ^2 local momentum, global track χ^2, global track probability, track kinks, calorimeter compatibility, N_{stations}

In terms of the vicinity of the leptons, all PF candidates, within a cone of $\Delta R < 0.5$, and all SVs, within a cone of $\Delta R < 0.6$ around the associated lepton, are taken into account (Sec. 6.1.2). The different PF candidate species, i.e. neutral and charged hadrons, as well as photons, electrons and muons, are treated separately. The utilized input training features of the PF candidates and SVs are listed in Tab. 6.6.

Tab. 6.6: The PF candidates and SVs input training features, representing the vicinity of the associated lepton. Again, the variables in parentheses are excluded in the final version.

Type	N_{max}	Features
neutral hadrons	5	$p_{\mathrm{T}}^{\mathrm{rel.}}$, ΔR, p_{T}, puppi weight, `fromPV`
charged hadrons	25	$p_{\mathrm{T}}^{\mathrm{rel.}}$, ΔR, p_{T}, puppi weight, `fromPV`, d_{xy}, d_z, d_z associated PV
photons	10	$p_{\mathrm{T}}^{\mathrm{rel.}}$, ΔR, p_{T}, puppi weight, `fromPV`
electrons	3	$p_{\mathrm{T}}^{\mathrm{rel.}}$, ΔR, p_{T}, d_{xy}, d_z
muons	3	$p_{\mathrm{T}}^{\mathrm{rel.}}$, ΔR, p_{T}, d_{xy}, d_z
secondary vertices	4	p_{T}, ΔR w.r.t. the lepton, χ^2, N_{dof}, d_{xy}, (uncertainty d_{xy}), d_{3D}, uncertainty d_{3D}, significance d_{3D}, $cos(\Theta)$, highest d_{xy}, 2^{nd} highest d_{xy}, highest d_{3D}, and 2^{nd} highest d_{3D} of tracks

A special type of neural networks, so called LSTM recurrent neural networks or LSTM units, are applied in the *DeepLepton* architecture (Sec. 6.2.3). The usage of these LSTM units is very computing intensive. Thus, only a certain number N_{max} of each PF candidate species and SVs is considered for the input of the DNN, instead of the full list of particles. This measure effects, that less important particles are neglected, due to the sorting by a decreasing relevance.

A short explanation of the input variables of the *DeepLepton* classifier that are not discussed in Ch. 4, is given in Tab. 6.7.

6.2.2 Output classes

The *DeepLepton* neural network is based on the supervised learning task (Sec. 5.1). Therefore, it is essential to provide a training data set, that consists of input-output pairs, where the input is a vector of the training features (Sec. 6.2.1) and the output refers to the actual lepton class. The output classes are defined, according to the MC truth, which has been stored in the training files. They also specify the structure, i.e. the number of nodes, of the last layer in the DNN, the output layer. The leptons are divided into the three categories, according to their source from the hard scatter: prompt, non-prompt and fake.

Besides this multiclassification task, a simplified training model is performed with just two output classes, prompt and not-prompt. Here, the non-prompt and fake leptons are merged into a single background class. The impact, of this simplification, is discussed later in Sec. 8.1.6.

Practically, the true output class, of a final state lepton, is obtained from the MC truth information in the training samples. To this end, the variables `mcMatchId` and `mcMatchAny` are used to define the sets M and N:

$$M = \{lepton : |\texttt{mcMatchId}(lepton)| \in \{6(\mathrm{t}), 23(\mathrm{W}^+), 24(\mathrm{Z}), 25(\mathrm{H}), 37(\mathrm{H}^+)\}\} \tag{6.4}$$

$$N = \{lepton : |\texttt{mcMatchAny}(lepton)| \in \{4(\mathrm{c}), 5(\mathrm{b})\}\}. \tag{6.5}$$

Tab. 6.7: Additional descriptions of the input features of the *DeepLepton*, which are not discussed in Ch. 4.

feature	description
lepton features:	
d_{3D}	3D distance w.r.t. the primary vertex (cm)
inner track χ^2	normalized χ^2 of the inner track of a lepton
inner track valid hit fraction	fraction of valid hits on the inner track of a lepton
p_{T} error track	p_{T} error for the GSF track or the best muon track
jet ΔR	ΔR between the lepton and the nearest jet
tracker layers	# of tracker layers with hits
pixel layers	# of layers in the pixel detector with hits
tracker hits	# of total hits in the tracker
lost inner hits	# of lost hits on the inner track
lost outer hits	# of lost hits on the outer track
jet p_{T} ratio	p_{T} (lepton)/p_{T} (nearest jet)
jet $p_{\mathrm{T}}^{\mathrm{rel.}}$	p_{T} of the lepton transverse to the jet axis (subtracting the lepton)
`jetBTagDeepCSV`	discriminator of the DeepCSV b tagging algorithm
muon specific features:	
muon inner track relative error	relative p_{T} error on the inner track of a muon
χ^2 local position	matching in position of a tracker muon
χ^2 local momentum	matching in momentum of a tracker muon
global track χ^2	normalized χ^2 of the global muon track
global track probability	pseudo-probability of the global muon track
N_{stations}	# of matched muon stations
PF candidate features:	
puppi weight	probability that the respective particle is from the PV [63]
`fromPV`	number between 0 and 3, which defines how tight the association of the respective candidate with the PV is (3: tightest association - the track is used in the PV fit, 0: the track is used in the fit of another PV)
d_z associated PV	longitudinal distance w.r.t. the PV associated to the respective candidate (cm)
SV features:	
χ^2	χ^2 of the vertex fit
N_{dof}	# of degrees of freedom of the fit $(2 \cdot N_{tracks} - 3)$
d_{xy}	transverse distance from the PV (cm)
d_{3D}	3D distance from the PV (cm)
$cos(\Theta)$	cosine of the angle between the 3D displacement and the momentum
(2^{nd}) highest d_{xy} or d_{3D}	(second) highest $\|d_{xy}\|$ or $\|d_{3D}\|$ of vertex tracks

With these sets and their complementary sets M^{C} and N^{C}, the following conditions are formed, in order to separate the three lepton classes from each other:

$$
\begin{aligned}
prompt &= M && (6.6)\\
not-prompt &= M^{\mathrm{C}} && (6.7)\\
non-prompt &= M^{\mathrm{C}} \cap N && (6.8)\\
fake &= M^{\mathrm{C}} \cap N^{\mathrm{C}}. && (6.9)
\end{aligned}
$$

Prompt leptons arise out of the decay of a W, Z or Higgs boson, as well as from a top quark, which further decays e.g. into a W boson and a bottom quark. Non-prompt leptons originate from decays of longer-lived particles, like heavy-flavour hadrons containing b or c quarks. There is also the possibility, that particles are misidentified as leptons, for example hadrons within a jet. In this case the particle is referred to as a fake lepton. More theoretical background about the lepton classes can be found in Sec. 4.2.

6.2.3 Deep neural network architecture

The architecture of the *DeepLepton* neural network is heavily inspired by the *DeepJet* framework. It consists of two main parts: on the one hand, a set of parallel arranged LSTM RNNs [137], in combination with preceding one-dimensional CNNs and on the other hand, a well-established FC neural network. The first part, with the convolutional and LSTM components, has the function to process the list of PF candidates and SVs. The output of the LSTM RNNs together with the lepton features represent the input for the second part of the neural network, the dense layers. A schematic layout of the reference architecture is shown in Fig. 6.3. In the following the sequence and function, of the single layers in the *DeepLepton* neural network, is treated. More detailed information about DNNs and their components can be obtained from Ch. 5.

The vicinity of the lepton is included in the *DeepLepton* network training, in form of low-level features of the PF candidates and SVs. In this case, vectors with a variable length have to be handled by the DNN, since the leptons have varying numbers of PF candidates and SVs. Hence, the vicinity features can not be fed directly into simple dense layers. LSTM RNNs are applied, in order to deal with this vectorial training data structure. RNNs are able to process variable-length inputs into a fixed-length output (Sec. 5.2.2). A separate LSTM RNN is applied for each of the PF candidate species, as well as for the SVs. For the SVs and the charged hadrons LSTM units with 150 nodes are utilized, whereas 50 nodes are selected elsewise. It is assumed, that the SVs and the charged hadrons contain the most information within the vicinity features, compared to the other PF candidate species.

During the experimental part of this work, bidirectional LSTM RNNs are tested, too. Bidirectional RNNs are a modification of traditional RNNs, where the input array is processed twice: firstly, in the given order and secondly, in the reverse order. The presumption is, that the performance of the training model would benefit from a second iteration, if there are more than one relevant features for the ordering of the input data. The results of these tests are discussed in Sec. 8.1.1.

One disadvantage of LSTM RNNs is that they are computing intensive. In order to reduce the training time, two measures are taken. For one thing, the maximum number N_{max} of PF

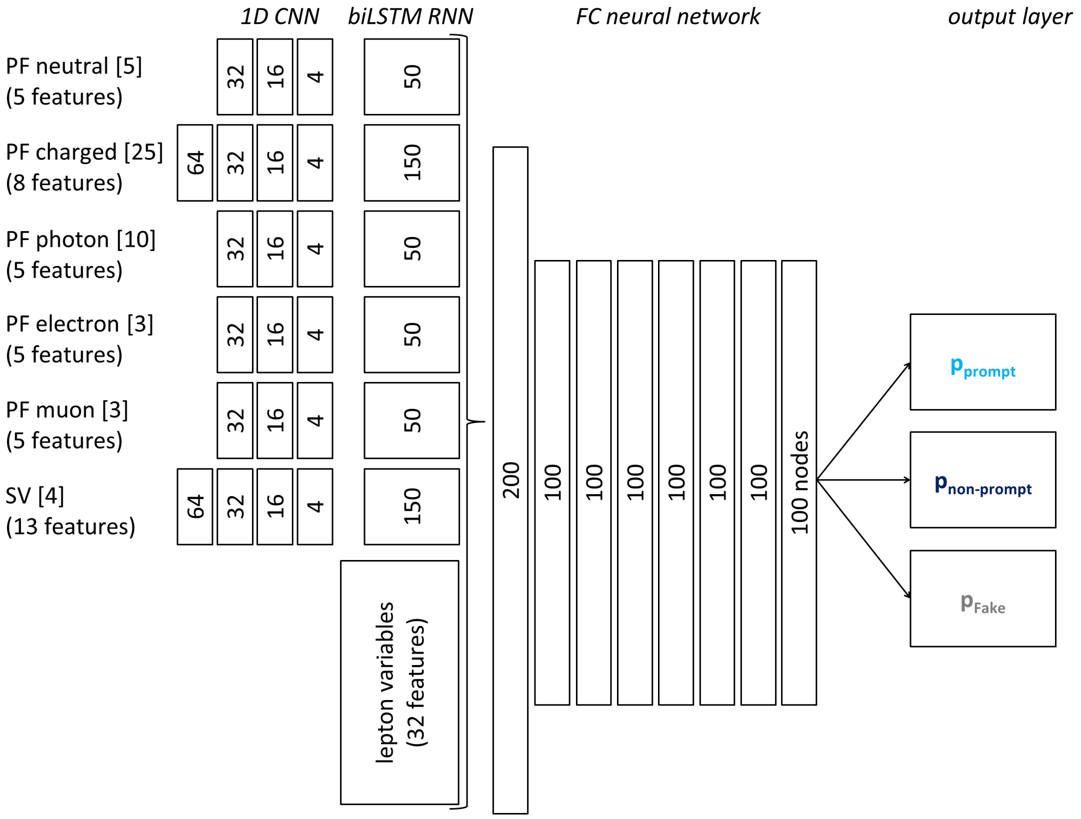
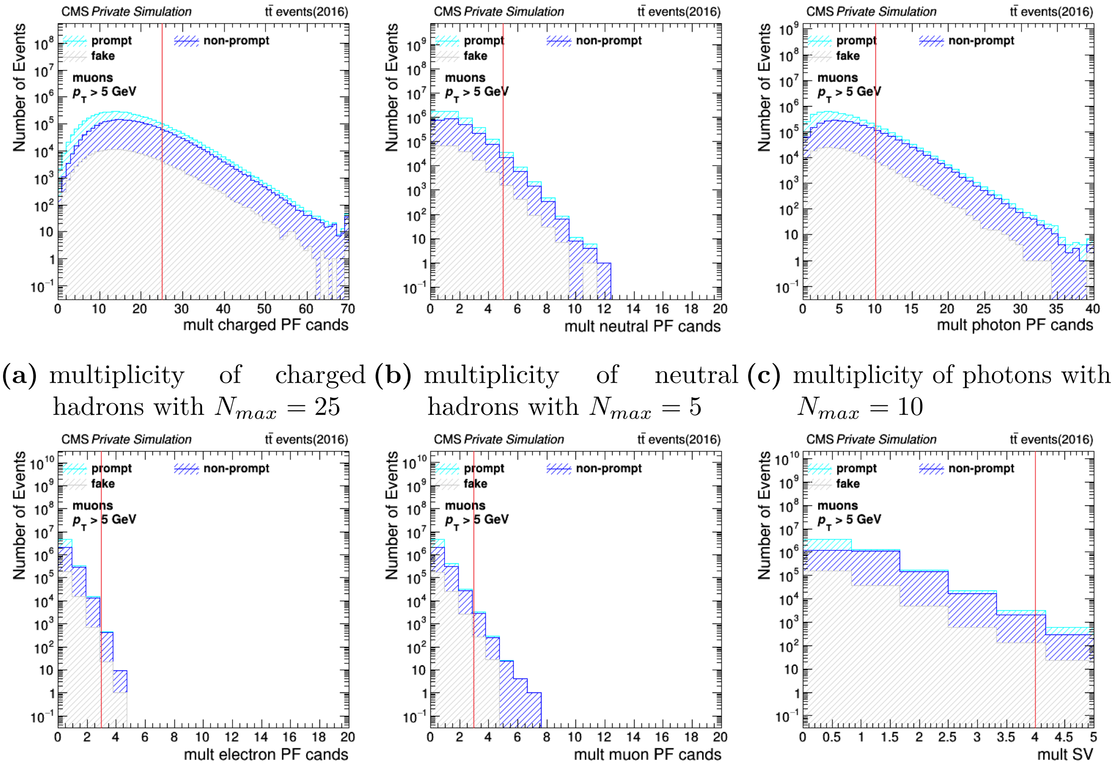


Fig. 6.3: The reference network architecture of the *DeepLepton* classifier, is closely related to the *DeepJet* [6] framework.

candidates and SVs per lepton is limited. The selected values for N_{max} are illustrated in the respective histograms of the $t\bar{t}$ training data sample in Fig. 6.4.

As a second measure, 1 × 1 convolutional layers are applied in advance, in order to generate a lower-dimensional representation of the training data, which is then processed by LSTM units. Four layers with 64, 32, 16 and four nodes are implemented in terms of the SVs and the charged hadrons, whereas three layers with 32, 16 and four nodes are chosen for the neutral hadrons, photons, electrons and muons. The ReLU is selected as an activation function. The auxiliary convolutional layers ensure, that the same transformation is applied to each PF candidate and SV [110]. This is achieved by setting the kernel size to 1. The kernel size is related to the size of the filter mask of the CNN and a kernel size equal to 1 means, that the PF candidates and SVs are handled one after another. Thus, the 1 × 1 convolutional layers do not act as common CNNs, but like dense layers, which are able to process sequential input training data [110]. Alternative training models including pooling layers, which are well-established in CNNs, are considered too. Furthermore, network architectures without the CNNs are tested. The results are summarized in Sec. 8.1.1.

The output of the LSTM units, i.e. the processed vicinity features of the PF candidate species and the SVs, together with the selected lepton variables, formed the input for the last stage in the *DeepLepton* network architecture, which is an ordinary FC neural network. This part of the network is initially implemented with eight fully connected dense layers, one layer with 200 nodes and seven more, with 100 nodes per layer. Modifications, of the layers in the FC neural network, are discussed in Sec. 8.1.1. As for the convolutional layers, the ReLU activation function



(a) multiplicity of charged hadrons with $N_{max} = 25$

(b) multiplicity of neutral hadrons with $N_{max} = 5$

(c) multiplicity of photons with $N_{max} = 10$

(d) multiplicity of electrons with $N_{max} = 3$

(e) multiplicity of muons with $N_{max} = 3$

(f) multiplicity of secondary vertices with $N_{max} = 4$

Fig. 6.4: The limitation of the number of PF candidates and SVs per lepton (red line). The multiplicity distributions are plotted for muons of the $t\bar{t}$ training data sample with a $p_T \geq 5$ GeV.

is chosen. The final layer of the whole DNN, the output layer, is formed by another dense layer with three nodes, referring to the three output classes. The so called softmax activation function is applied in this layer. It is defined in Eq. (5.2). The softmax function effects, that the entries in the nodes of the output layer add up to one.

Besides the usage of the ReLU activation function, the impact of the ELU and the scaled exponential linear unit (SELU) [161], on the performance of the *DeepLepton* classifier, is tested. The results are discussed in Sec. 8.1.3.

6.2.4 Training model hyper-parameters

In this subsection the settings of the hyper-parameters of the *DeepLepton* training model, like batch size, learning rate, momentum, number of epochs, as well as batch normalization and dropout (Ch. 5), are outlined.

In general, batch normalization [132], followed by dropout [125], is applied after each layer in the *DeepLepton* neural network, including an initial batch normalization before the first convolutional layer. The normalization of the input features of each layer, prevents the distribution of the input features from changing from layer to layer. This helps to increase the stability of the neural network. Dropout, as a regularization method, decreases the probability of overfitting, by randomly ignoring a certain percentage of nodes, for each iteration step of the training procedure. In the *DeepLepton* neural network a standard dropout rate of 0.5 (50%) is selected.

There are various approaches for the optimal choice of the batch size in the machine learning community. A rather big batch size of 10,000 is considered for the *DeepLepton* neural network, because of a large bulk of training data (Tab. 6.3). After a number of trials, the initial learning rate was set to 0.001. This learning rate is halved each time, when five sequential epochs of no progress appear. A graph, of the evolution of the learning rate and the validation and training losses in the *DeepLepton* network training, is shown in Fig. 8.6. The momentum, i.e. the step size taken after an epoch where the validation loss did not improve, is set to 0.2. The training is performed for 100 epochs. The time consumption is about five to ten hours, depending on the effective training data set and the network architecture. In terms of optimization, the so called Adam optimizer [162] is applied.

6.2.5 Hardware

A NVIDIA GTX 1080TI GPU is utilized for the practical execution of the *DeepLepton* neural network training. It is part of an in-house server consisting of four GPUs in total (Fig. 6.5). The specifications of the GPU-server are listed below:

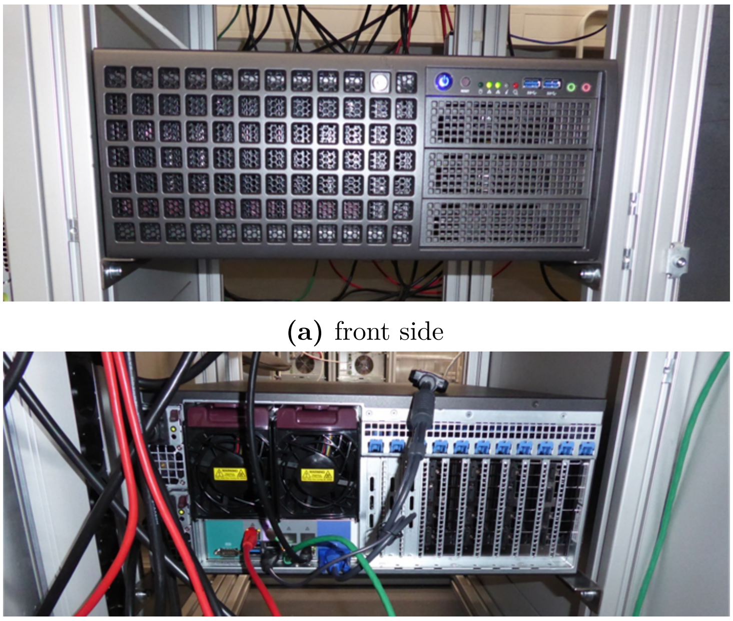
- 1x System Building Block 7049-TRT
- 2x Intel Processor SKL-SP SILVER 4114
- 10C 2.2GHz 13.75M
- 8x 16GB Central Memory DDR4-2666MHz ECC RDIMM
- 1x 960GB 2.5” SSD SATA 6Gbps 3D TLC with 3DWPD
- 2x 12TB 3.5” HDD SATA 6Gbps 7200rpm 24x7 operation
- 4x NVIDIA GTX 1080TI GPU with radial cooling
- 1x 19” Rackmount Converstion Kit

Performing the neural network training on a GPU, drastically reduced the training time, compared to initial attempts on CPUs. This enabled both, the usage of larger training sets, and also the testing of more computing intensive network components, like the bidirectional LSTM units.

6.2.6 Software

The implementation of the *DeepLepton* neural network is heavily inspired by the “DL4Jets” framework [163], also known as *DeepJet*. It is designed for jet identification and classification. The lepton classification task, which is the subject of this master thesis, represents a similar exercise, in terms of machine learning. It may not be as complex as discriminating jets, though.

The *DeepJet* classifier, and thus also *DeepLepton*, is based on Keras [164]. It is an application programming interface (API) for neural networks, which is using the Python programming language. TensorFlow [165] is applied, as a backend. The function of TensorFlow is to handle the required tensor manipulation. The Compute Unified Device Architecture (CUDA) [166], developed by NVIDIA (Santa Clara, US), is deployed, in order to enable parallelized operations on the GPU.



(a) front side

(b) back side

Fig. 6.5: The front and back side of the in-house GPU-server at the HEPHY.

6.3 Training model evaluation

This section handles the evaluation of the training model, of the *DeepLepton* classifier, on a (MC) test data sample. Furthermore, it addresses the validity of the training model in real detector data, via so called data-MC comparisons. The whole workflow of the evaluation is illustrated in Fig. 6.6 and the single steps are outlined in the following sections.

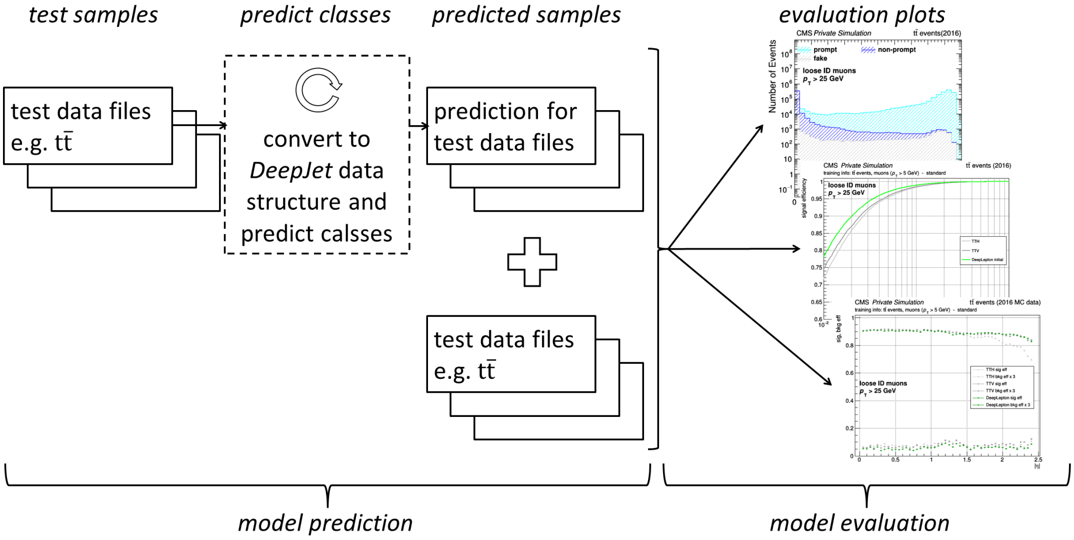


Fig. 6.6: *DeepLepton* evaluation. First, the application of the *DeepLepton* training model on a test data set results in the predicted output classes. Then, discriminator shapes and various ROC plots can be calculated.

6.3.1 Performance in MC samples

6.3.1.1 Prediction of output classes

The first step in the evaluation procedure is the deployment of the *DeepLepton* training model on a test data set, in order to receive the predicted probabilities for the output classes, per lepton in the test sample. Attention is paid on an independent data set, which has not been used during the training of the classifier.

In practice, this step is executed by a tool, that is provided by the *DeepJet* framework. First, the test data ROOT files are converted into the needed data structure. Then the prediction of the output classes is performed. As a result, separate ROOT files are produced, which contain the predicted outputs, according to the respective training model. A file entry refers to the probability p_i, that a certain lepton is associated with an output class i (prompt, non-prompt and fake). Due to the softmax activation function (Eq. (5.2)), which is utilized in the output layer, the probabilities per lepton add up to one.

$$\sum_{i=1}^{N_{classes}} p_i = 1 \tag{6.10}$$

6.3.1.2 Discriminator shapes

In a next step, the discriminator shapes are calculated. A discriminator shape represents the distribution of the predicted probabilities (discriminator values) for a certain output class. As already mentioned, the MC samples contain the information of the actual lepton class (Sec. 6.2.2). Hence, it is possible, to display the distribution of the discriminator values of each lepton class, within the whole shape, that is associated with a single output class, too. To this end, histograms are filled with the discriminator values, that have been predicted for the leptons of the test sample.

In principle, discriminator shapes can be computed for each output class. An example plot is illustrated in Fig. 6.7.

The goal of the classifier is to provide a good separation between the signal and the background leptons, which can be visualized by the discriminator shape. Thus, the discriminator shape gives a first impression of the quality of the training model. Moreover, smooth shapes are desirable, while leaps and bumps indicate weaknesses of the model.

6.3.1.3 ROC-curves

In order to get a clearer picture of the quality of the training model, the so called ROC curve is determined. ROC curves are a very useful tool, to check the performance of a training model. Furthermore, different models can be compared with each other. The ROC curve visualizes the true positve rate (TPR) dependent on the false positve rate (FPR).

$$TPR = \frac{\sum TP}{\sum \text{ positive actual condition}} = \frac{\sum TP}{\sum TP + \sum FN} \tag{6.11}$$

$$FPR = \frac{\sum FP}{\sum \text{ negative actual condition}} = \frac{\sum FP}{\sum FP + \sum TN} \tag{6.12}$$

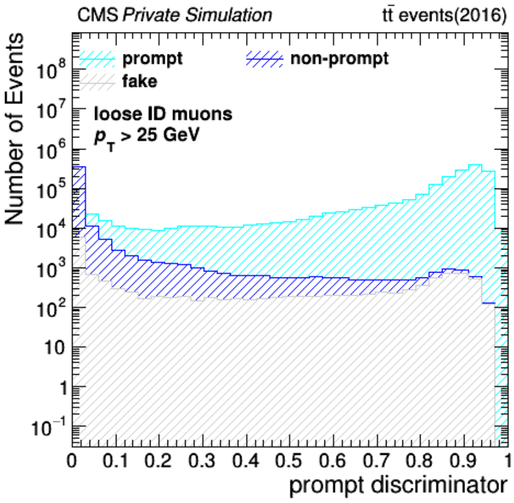


Fig. 6.7: The prompt discriminator shape, computed for muons ($p_{\mathrm{T}} > 25$ GeV) in the $t\bar{t}$ test data sample. Besides the global shape, the distribution of the discriminator values within each of three actual lepton classes is displayed. A high discriminator value indicates a high predicted probability of being a prompt lepton.

The TPR, also referred to as sensitivity, and the FPR, or 1 - specificity, are defined with respect to the terminology of the confusion matrix in Tab. 6.8. The TPR and FPR are also labeled as signal and background efficiency.

Tab. 6.8: The confusion matrix illustrates the relation between actual and predicted conditions, in terms of classification.

		predicted condition	
		positive	**negative**
actual condition	**positive**	True Positive (TP)	False Negative (FN)
	negative	False Positive (FP)	True Negative (TN)

With regard to the lepton classification task, prompt leptons are considered as signal events and non-prompt, as well as fake leptons, are representing the background events. In general, the signal and background efficiencies are not constant, but they depend on the respective working point. This means, that different discriminator cut values p_{cut}, for the predicted probabilities p_i on the x-axis of the discriminator shape plot (Fig. 6.7), yield different efficiencies. Hence, each discriminator cut value p_{cut} results in a certain point of the ROC curve.

In principle, the signal and background efficiencies are calculated according to

$$\epsilon_{\text{sig}}(p_{cut}) = \frac{N_{sig}^{p_{sig}>p_{cut}}}{N_{sig}} = \frac{N_{prompt}^{p_{prompt}>p_{cut}}}{N_{prompt}} \text{ and} \tag{6.13}$$

$$\epsilon_{\text{bkg}}(p_{cut}) = \frac{N_{bkg}^{p_{bkg}>p_{cut}}}{N_{bkg}} = \frac{N_{not-prompt}^{p_{prompt}>p_{cut}}}{N_{not-prompt}}. \tag{6.14}$$

For this purpose, the actual non-prompt and fake leptons are merged into a single background class, labeled as not-prompt in the following. N_{prompt} and $N_{not-prompt}$ are related to the total number of leptons in the particular class. $N_{prompt}^{p_{prompt}>p_{cut}}$ and $N_{not-prompt}^{p_{prompt}>p_{cut}}$ refer to those leptons of the respective class, which fulfill the condition, that the predicted probability of being a prompt lepton, p_{prompt} is greater than the selected cut value p_{cut}.

To take into account the fact, that the leptons in the training and test data set are mixed together from samples, with different cross sections σ^{sample} (Tab. 6.1, 6.2), the equations are modified appropriately. A relative scale factor for the luminosity

$$w_L = \frac{\sigma^{sample}}{N_{sim}^{sample}} \tag{6.15}$$

is determined per lepton, where N_{sim}^{sample} is the number of simulated leptons in the associated sample. After this modification, the effective formulas for the calculation of the signal and background efficiencies are given by

$$\epsilon_{\text{sig}}(p_{cut}) = \frac{\sum_{i=1}^{N_{prompt}^{p_{prompt}>p_{cut}}} w_{L,i}}{\sum_{i=1}^{N_{prompt}} w_{L,i}} \text{ and} \tag{6.16}$$

$$\epsilon_{\text{bkg}}(p_{cut}) = \frac{\sum_{i=1}^{N_{not-prompt}^{p_{prompt}>p_{cut}}} w_{L,i}}{\sum_{i=1}^{N_{not-prompt}} w_{L,i}}. \tag{6.17}$$

Here, the relative luminosity weight $w_{L,i}$, per lepton i, is accumulated.

For the comparison of the results, of the *DeepLepton* training model with the existing *TTV* and *TTH* lepton IDs, a preselection is applied to the leptons in the test sample. The goal is to sort out easy identifiable background leptons beforehand. To this end, a loose selection is chosen. The requirements, to pass this loose selection (`loose ID`), are listed in Tab. 6.9.

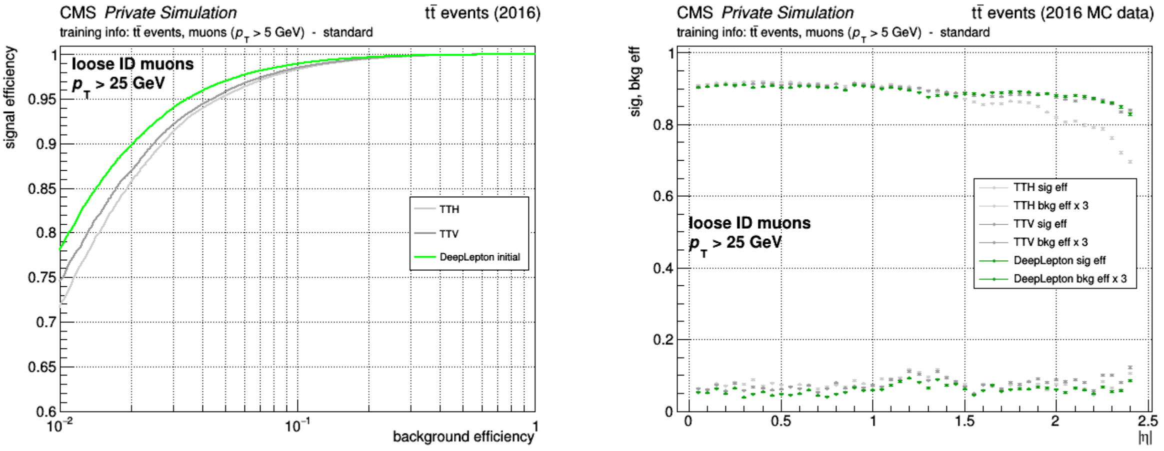
An example ROC curve, which compares the *DeepLepton* classifier with lepton IDs of the *TTV* and *TTH* BDTs, is illustrated in Fig. 6.8a. In some cases, the area under the curve (AUC) is taken as a measure for the quality of a classifier, instead of the curve itself. However, the AUC value is not considered for the assessment of the performance of the *DeepLepton* classifier, since ROC curves with the same AUC value may have different shapes and thus may intersect.

6.3.1.4 Efficiencies binned in p_T, η and number of vertices

ROC plots are useful to evaluate the average performance of a training model on the whole test sample. However, it might be of interest, if the performance varies for certain phase spaces of the

Tab. 6.9: The `loose ID` selection for muons in the evaluation of the *DeepLepton* training model. The `pdgId` refers to a numeric particle ID, which is assigned by the POG [167]. A `pdgId` $= \pm 13$ is associated with muons and antimuons ($\mu^{\pm}$).

muon feature	feature requirements
$\lvert$`pdgId`$\rvert$	13
$p_T >$	10
$\lvert\eta\rvert <$	2.4
mini $I_{\text{rel.}} <$	0.4
significance $d_{3D} <$	8
$\lvert d_{xy}\rvert <$	0.05
$\lvert d_z\rvert <$	0.1
`isPFMuon`	yes
is POG medium Muon ID	yes



(a) **(b)**

Fig. 6.8: Example plots for the evaluation of a *DeepLepton* training model on MC test data. ROC curves **(a)** and signal and background efficiencies binned in η **(b)**.

test data set. A correlation between the efficiencies and event properties, such as the amount of pileup, is an undesired behavior. Stability versus η, in turn, is a desirable property. To address this issue, the signal and background efficiencies are plotted in bins of η, for a fixed working point. Furthermore, ROC curves as a function of the transverse momentum p_T and the number of vertices N_{vertex} are produced. A standard working point of an overall signal efficiency (=TPR) of 90% is defined to this end.

An example plot is shown in Fig. 6.8b. The statistical error of the signal and background efficiencies ϵ_i, with $i =$ sig or bkg (Eq. (6.13) and 6.14), is given by the standard error of the mean (SEM)

$$SEM(\epsilon_i) = \frac{\sigma(\epsilon_i)}{\sqrt{N_i}}, \tag{6.18}$$

where σ refers to the standard deviation

$$\sigma(\epsilon_i) = \sqrt{\frac{1}{N_i} \cdot \sum_{\ell=1}^{N_i} (x_\ell - \bar{x})^2}, \text{ with} \tag{6.19}$$

$$\bar{x} = \frac{1}{N_i} \cdot \sum_{\ell=1}^{N_i} x_\ell \tag{6.20}$$

and N_i is the total number of actual signal or background leptons in the respective bin selection. The signal or background efficiency regarding a single lepton x_ℓ can either be 1 or 0. Hence, the mean $\bar{x}$ and the standard deviation $\sigma(\epsilon_i)$ simplify in this case of a binomial distribution of the single efficiency values:

$$\bar{x} = \frac{N_i^{p_{sig} > p_{cut}}}{N_i} = \epsilon_i \tag{6.21}$$

and

$$\begin{aligned} \sigma(\epsilon_i) &= \sqrt{\frac{1}{N_i} \cdot \sum_{\ell=1}^{N_i} (x_\ell - \epsilon_i)^2} \\ &= \sqrt{\frac{1}{N_i} \cdot [N_{sig}^{p_{sig} > p_{cut}} \cdot (1 - \epsilon_i)^2 + (N_i - N_i^{p_{sig} > p_{cut}}) \cdot \epsilon_i^2)]} \\ &= \sqrt{\epsilon_i \cdot (1 - \epsilon_i)} \end{aligned} \tag{6.22}$$

For the signal and background efficiencies ϵ_i, this finally results in the SEM of a Bernoulli distribution

$$SEM(\epsilon_i) = \sqrt{\frac{\epsilon_i * (1 - \epsilon_i)}{N_i}}, \tag{6.23}$$

which is a special case of the binomial distribution, due to the Boolean valued outcomes of the single lepton efficiencies.

6.3.2 Data-MC comparisons

So far, the evaluation of the *DeepLepton* classifier is treated solely on artificial MC data. Apriori, it is not clear, if the selected MC samples are representing the detector data in an appropriate way, and how the *DeepLepton* training model performs in real detector data. Therefore, the simulation is compared to data and it is checked, whether the efficiencies, that can be achieved in MC data, are still valid for the real data or not.

Comparisons of the DY and the $t\bar{t}$ MC samples, with the related detector data, are considered, since these samples are also used in the training procedure. The selection of particles of a certain type, that arise from a certain process, can be made very easily in the simulated MC samples, by means of the MC truth information. However, this information is not available in the real detector data. Thus, a method has to be applied, that enables to select leptons, emerging from the desired DY and $t\bar{t}$ events in the detector data. Both processes are characterized by two opposite charged final state leptons. An approach, that addresses this issue, is the "Tag & Probe" method [168]. It is discussed in the following, where the 2016 *Double Muon* data samples (Tab. 6.11) with a total integrated luminosity of $L = 36\,\mathrm{fb}^{-1}$ are examined. These samples contain real events with at least two muons.

6.3.2.1 Tag & Probe method

The "Tag & Probe" method is a technique for the real detector data, that allows to select particles of a desired type with a small fake rate, by making use of a known mass resonance. The mass resonance of the Z boson is utilized, in order to identify muons, that derive from the DY process (Sec. 4.1.1). First, events with tag muons are obtained from the data, by applying very tight selection criteria. The selection criteria (Tab. 6.10) are designed in a way, so that prompt muons are identified with a reasonable small background rate.

In a second step, a set of reconstructed muons, passing a loose selection (Tab. 6.10), are obtained from the detector data, by pairing them with the tags on condition, that the invariant mass of the muon-antimuon pair $m_{\mu^+\mu^-}$ corresponds to the mass of the Z boson m_Z. $m_{\mu^+\mu^-}$ is calculated, according to the definition for a collider experiment:

$$m_{\mu^+\mu^-} = \sqrt{2 \cdot p_T(\mu^+) \cdot p_T(\mu^-) \cdot (\cosh(\eta(\mu^+) - \eta(\mu^-)) - \cos(\phi(\mu^+) - \phi(\mu^-)))}. \tag{6.24}$$

These muons are referred to as probes. Despite the loose probe selection criteria, muons are identified with a high efficiency, due to the Z mass condition. A probe muon may also pass the tight selection criteria and therefore get assigned to the set of tags. Double counting is accounted for in the algorithm.

Tab. 6.10: The tight and loose muon selection criteria for tags and probes.

feature requirement	tag muon	probe muon
$p_T > 5$	yes	yes
$\vert$`pdgId`$\vert = 13$	yes	yes
$\vert\eta\vert < 2.4$	yes	yes
$\vert\text{mini } I_{\text{rel.}}\vert < 0.4$	yes	yes
significance $d_{3D} < 8$	yes	yes
$\vert d_{xy}\vert < 0.05$	yes	yes
$\vert d_z\vert < 0.1$	yes	yes
`isPFMuon`	yes	yes
$p_T > 10$	yes	no
is POG medium Muon ID	yes	no
`jetBTagDeepCSV` < 0.8958	yes	no
`mvaTTV` > 0.4	yes	no

For data-MC comparisons of muons from the DY process, a corridor of ± 15 GeV around the mass of the Z boson $m_Z \cong 91.2$ is chosen.

$$|m_{\mu^+\mu^-} - 91.2| < 15 \tag{6.25}$$

At the LHC, the DY process most likely co-occurs with one or more jets, deriving from the residual gluons and quarks within the protons, that are involved in the collision (Fig. 4.1b). For that reason, at least one neighboring jet is required in the event sample.

The main background, for this event selection, are high p_T single and dilepton $\mathrm{t\bar{t}}$ events (Fig. 6.9a). However, the $\mathrm{t\bar{t}}$ dilepton events are characterized by a missing transverse energy $\not{E}_\mathrm{T}$ of ≥ 40 GeV, due to the two neutrinos, which do not interact with the detector material (Fig. 6.9c). In contrast, the DY events are ranging more likely in a region with a $\not{E_T}$ below 40 GeV.

Considering the $\mathrm{t\bar{t}}$ data-MC comparisons, the missing mass resonance complicates an appropriate muon selection. Nevertheless, the "Tag & Probe" procedure is applied as described before. However, the condition, related to the invariant mass of the muon-antimuon pair, is modified to

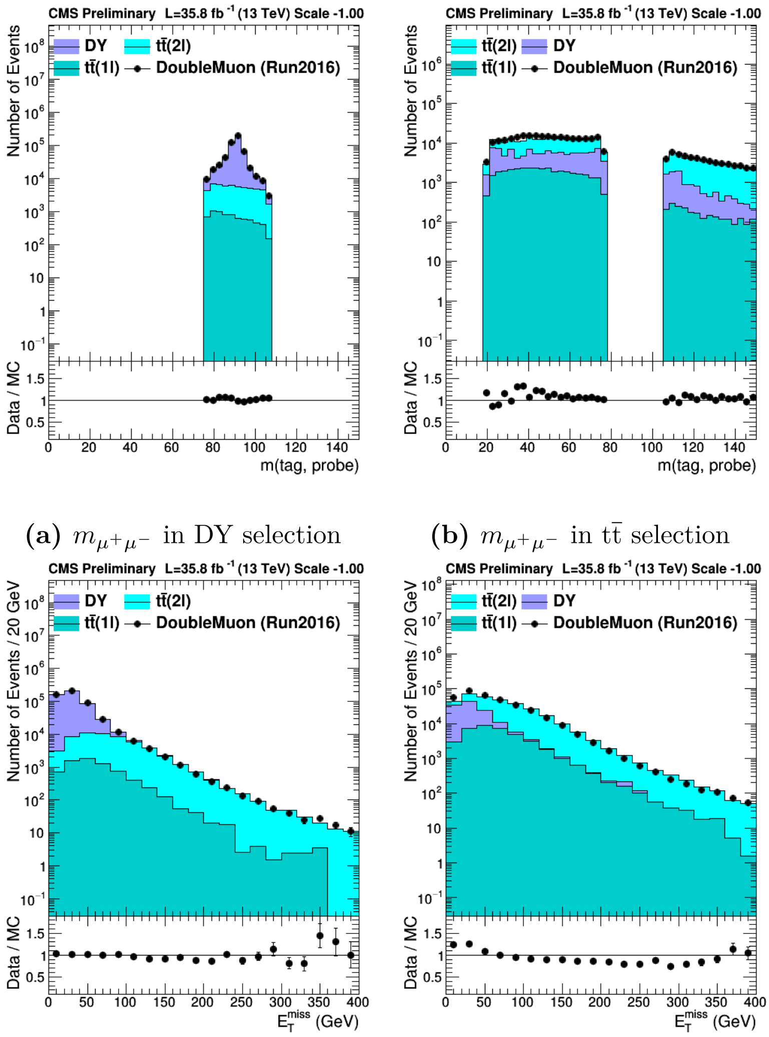
$$m_{\mu^+\mu^-} > 20 \wedge |m_{\mu^+\mu^-} - 91.2| > 15. \tag{6.26}$$

In addition to the mass resonance corridor of the Z boson, defined previously (Eq. (6.25)), the region below 20 GeV is excluded for the $\mathrm{t\bar{t}}$ event selection, in order to prevent from biasing low-mass DY events (Fig. 6.9b).

Again, at least one jet is required. Furthermore, one or more leading b jets are postulated for the $\mathrm{t\bar{t}}$ events in the detector data, due to the fact, that a top quark most likely decays into a W boson and a bottom quark. In order to identify jets, deriving from the hadronization of bottom quarks, the so called DeepCSV b tagger [108] is utilized. The main background for the applied $\mathrm{t\bar{t}}$ event selection is DY. The associated $\not{E}_\mathrm{T}$ distribution, shown in Fig. 6.9d, is again a discriminating feature between the two processes.

The distributions of the various input training features of the *DeepLepton* classifier show, whether the MC represents the data. Separate plots are produced for the probes and the tags in the DY and $\mathrm{t\bar{t}}$ selections. Furthermore, comparisons of the discriminator shapes in real and simulated data are made. These plots give some indication of the performance of the *DeepLepton* classifier in real data.

The 2016 detector data and MC data samples, which are considered for the data-MC comparisons, are listed in Tab. 6.11. A pileup reweighting, as well as a scale factor for the DeepCSV b tagger is applied in the MC data. As the application of the *DeepLepton* discriminator is computing intensive, due to large bulk of data, only 10% of the MC data is taken into account. The results of the data-MC comparisons are presented in Sec. 7.2.



(a) $m_{\mu^+\mu^-}$ in DY selection

(b) $m_{\mu^+\mu^-}$ in $\mathrm{t\bar{t}}$ selection

(c) $\not\!\!E_{\mathrm{T}}$ in DY selection

(d) $\not\!\!E_{\mathrm{T}}$ in $\mathrm{t\bar{t}}$ selection

Fig. 6.9: The data-MC plots, showing the distribution of the invariant mass of the muon-antimuon pair $m_{\mu^+\mu^-}$ and the missing transverse energy $\not\!\!E_{\mathrm{T}}$ in real and simulated data, for the DY (Eq. (6.25)) and $\mathrm{t\bar{t}}$ (Eq. (6.26)) selections.

Tab. 6.11: The 2016 detector data and MC data samples, which are used for the data-MC comparisons. The integrated luminosity amounts to $L = 36\,\mathrm{fb}^{-1}$.

Short Name	Sample Name	dataMC
`DoubleMuon_Run2016B_v2`	`/DoubleMuon/Run2016B-07Aug17_ver2-v1/MINIAOD`	data
`DoubleMuon_Run2016C`	`/DoubleMuon/Run2016C-07Aug17-v1/MINIAOD`	data
`DoubleMuon_Run2016D`	`/DoubleMuon/Run2016D-07Aug17-v1/MINIAOD`	data
`DoubleMuon_Run2016E`	`/DoubleMuon/Run2016E-07Aug17-v1/MINIAOD`	data
`DoubleMuon_Run2016F`	`/DoubleMuon/Run2016F-07Aug17-v1/MINIAOD`	data
`DoubleMuon_Run2016G`	`/DoubleMuon/Run2016G-07Aug17-v1/MINIAOD`	data
`DoubleMuon_Run2016H`	`/DoubleMuon/Run2016H-07Aug17-v1/MINIAOD`	data
`TTJets_DiLepton`	`TTJets_DiLept_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC
`TTJets_DiLepton_ext`	`TTJets_DiLept_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC
`TTJets_SingleLeptonFromTbar`	`TTJets_SingleLeptFromTbar_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC
`TTJets_SingleLeptonFromTbar_ext`	`TTJets_SingleLeptFromTbar_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC
`TTJets_SingleLeptonFromT`	`TTJets_SingleLeptFromT_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC
`TTJets_SingleLeptonFromT_ext`	`TTJets_SingleLeptFromT_TuneCUETP8M1_13TeV-madgraphMLM-pythia8`	MC

Chapter 7

Results

This chapter summarizes the results of the *DeepLepton* training model, that yields the best performance, in terms of muon identification in the test data set. First, the model efficiencies in MC data are presented. This includes discriminator shapes, ROC curves, as well as the signal and background efficiencies dependent on selected event features (p_{T}, η, and the number of vertices). The second part of this chapter treats comparisons of 2016 simulated MC data and CMS detector data, in order to demonstrate the validity of the *DeepLepton* classifier for real data. Alternative training models are discussed in Ch. 8.

7.1 Performance in MC samples

In general, two separate p_{T} regions are considered for the evaluation of the *DeepLepton* training model in MC data. Muons in the range of 10 GeV $< p_{\mathrm{T}} <$ 25 GeV, are associated with a low-p_{T} region and muons with a $p_{\mathrm{T}} >$ 25 GeV, are referred to as high-p_{T}. Its worth to note, that the training is performed, by using muons with a p_{T} down to 5 GeV. However, the lower-limit of 10 GeV for the evaluation is chosen, for reasons of comparability with the *TTV* and *TTH* classifiers, since these MVAs are trained on leptons with a $p_{\mathrm{T}} >$ 10 GeV.

Different configurations are tested during development of the *DeepLepton* framework. The final network architecture, with the best performance output among these tests, is shown in Fig. 7.1. Modifications towards the initial reference architecture (Fig. 6.3), as described in Sec. 6.2.3, are the deployment of biLSTM units, instead of ordinary LSTM units, and the splitting of the FC neural network. The intermediate vicinity features, which are represented by the outputs of the biLSTM RNNs, and the global lepton features are initially processed in separate dense layers. Afterwards, they are merged together and are fed into another final FC neural network. Instead of the ReLU, the ELU is utilized as an activation function, for each layer within the neural network, except from the biLSTM units, which use a special gating structure of sigmoid and tanh functions (Fig. 5.5).

Muons, from the $\mathrm{t\bar{t}}$ sample in Tab. 6.2, are used as a training data set. The lists of PF candidates are sorted by a descending $p_{\mathrm{T\ rel}}$, while the order of the SVs is given by their total p_{T} (Sec. 6.1.3.2). Non-prompt leptons are chosen as a reference class for the reweighting of the training data, in p_{T} and η, as explained in Sec. 6.1.3.3. The training procedure is performed according to the settings in Sec. 6.2.

Fig. 7.2 shows the distribution of the predicted prompt discriminator values for the different classifiers in the $\mathrm{t\bar{t}}$ test data set. The non-prompt and fake discriminator shapes of the *DeepLepton* classifier are plotted in Fig. 7.3.

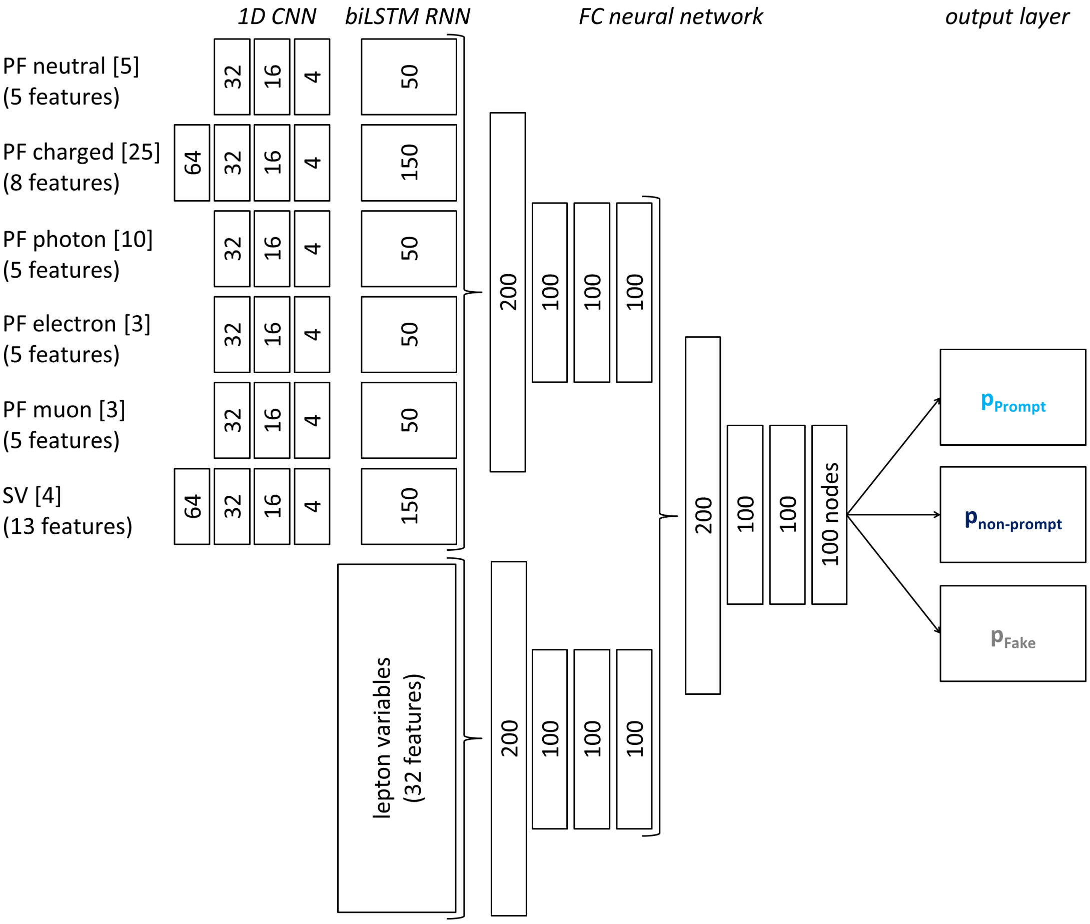
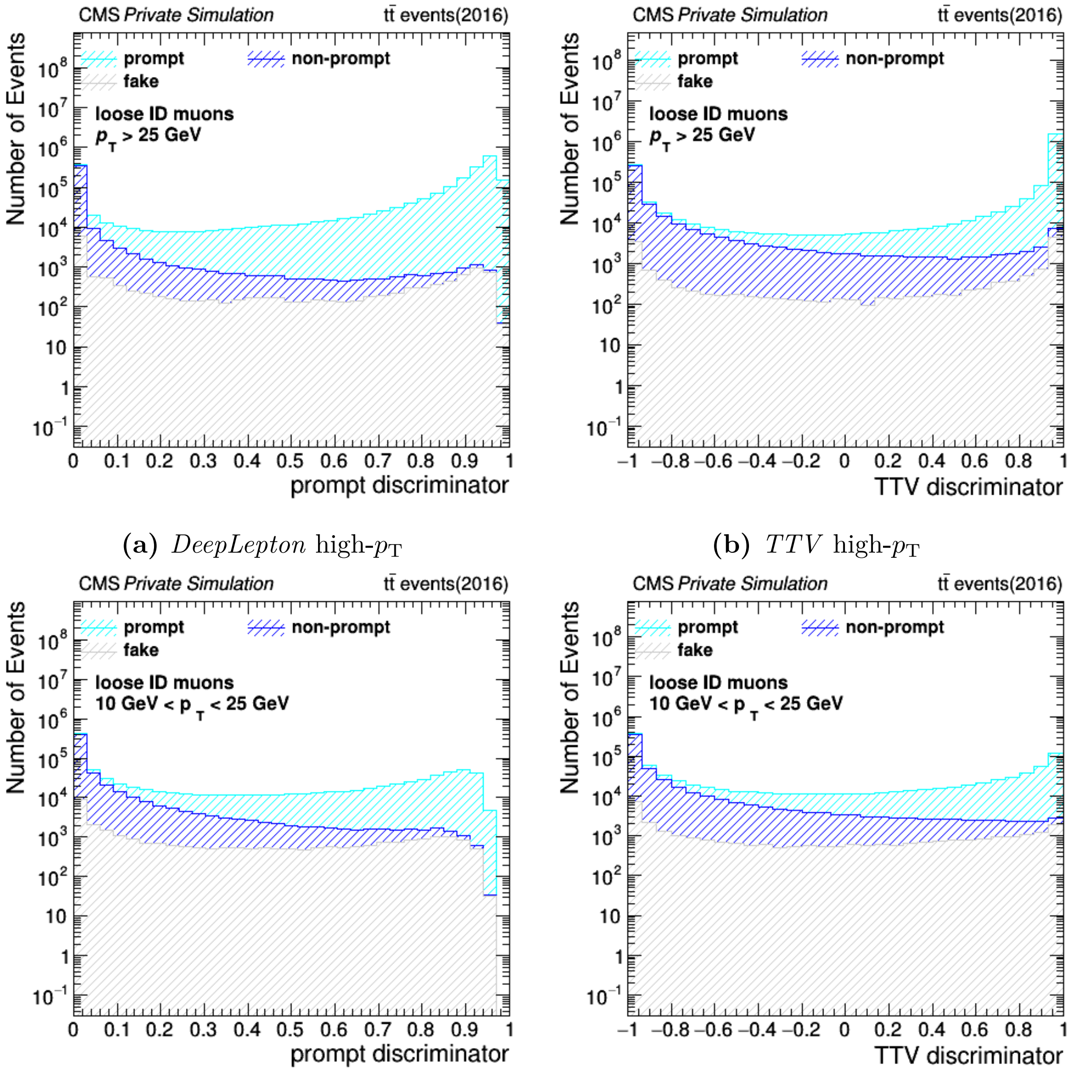


Fig. 7.1: The final network architecture of the *DeepLepton* neural network, which yields the best performance in MC data.

The ROC curves are calculated, by means of the predicted discriminator values of the final *DeepLepton* training model. In addition to this, the relating ROC curves, of the competitive *TTV* and *TTH* MVAs, are computed. These curves are plotted as a reference, in order to assess the performance of the *DeepLepton* classifier. The determined ROC plots in Fig. 7.4 show, that the *DeepLepton* ID clearly has the potential to outperform the two other classifiers, which are based on BDTs.

For a medium working point, with a signal efficiency of around 80%, *DeepLepton* results in an improvement of the background efficiency, against the *TTV* and *TTH* approaches, of about 1-2% for low-p_{T} muons and 0.1-0.2% for the high-p_{T} range. Whereas in case of a given background efficiency, the increase in the signal efficiency amounts up to 4-5% for both p_{T} regions. In general, muon identification is much more difficult in the low-p_{T} range, than for high-p_{T} muons.

Signal and background efficiencies are plotted in bins of p_{T}, η and the number of vertices N_{vertex}. To this end, a fixed working point with an overall signal efficiency of 90 % is selected. The resulting plots are used as a kind of quality check for the *DeepLepton* classifier. By means of the distributions in the plots, it is proofed, if the performance of the training model is related to different phase spaces of these three features. Since the efficiencies, achieved by the *DeepLepton* classifier, are desired to be independent from p_{T}, η and N_{vertex}, ideal efficiencies are supposed to be represented by flat horizontal curves.



(a) *DeepLepton* high-p_{T}

(b) *TTV* high-p_{T}

(c) *DeepLepton* low-p_{T}

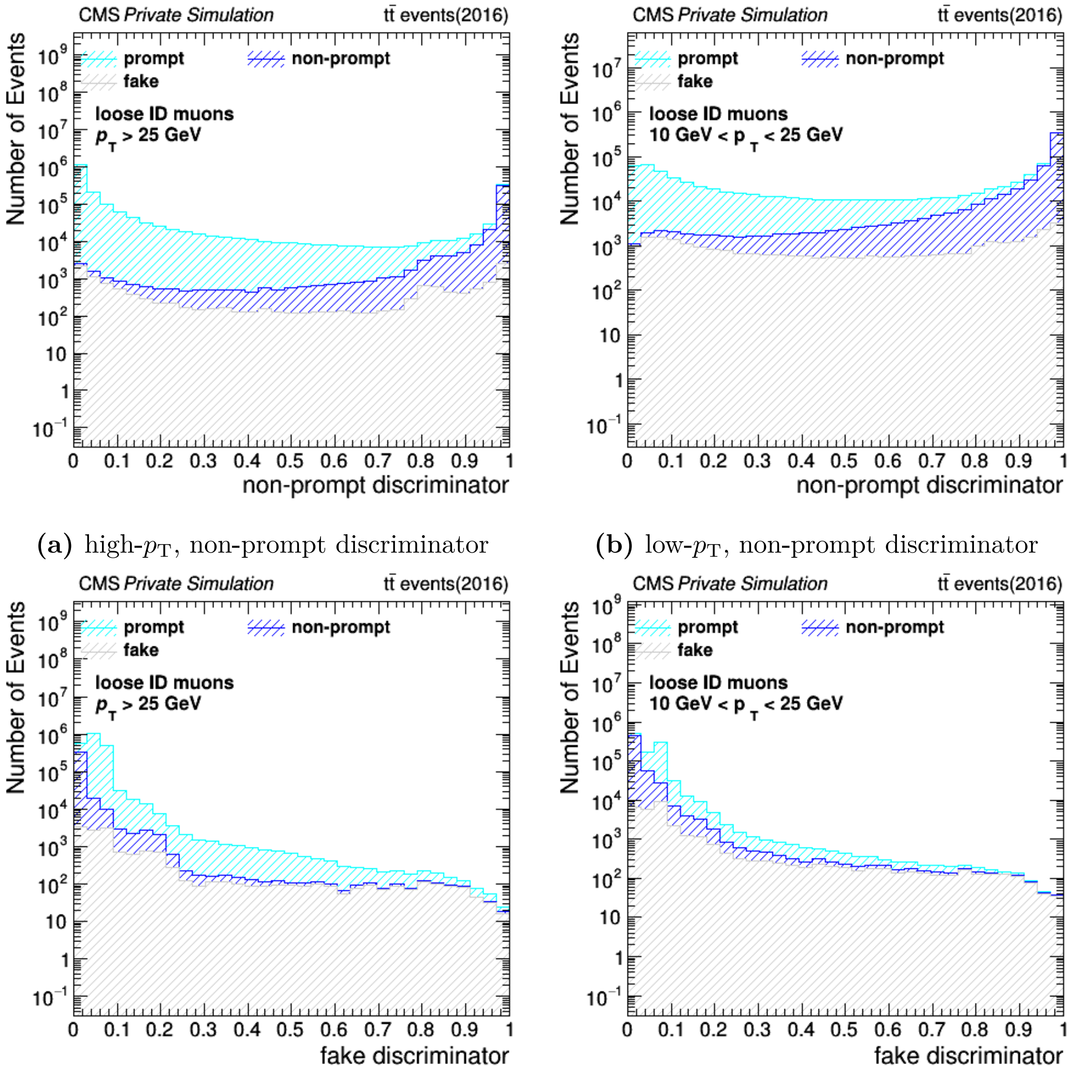
(d) *TTV* low-p_{T}

Fig. 7.2: The prompt discriminator shapes of the *DeepLepton* **(a)+(c)** and *TTV* **(b)+(d)** classifiers in separate p_{T} regions. The plots show smooth shapes for both, *DeepLepton* and *TTV*, but the *DeepLepton* shape has an abrupt decrease for high discriminator values.

The efficiencies binned in η, which are shown in Fig. 7.5, are reasonably flat. Noticeable dependencies just occur for high-p_{T} muons in the endcap regions of the detector, with $|\eta| > 2.0$. They are more significant for the *TTH* classifier.

Slight dependencies are displayed concerning the efficiencies, binned in N_{vertex}, towards higher multiplicities in Fig. 7.6. However, no considerable differences in the efficiency shapes of the three competitors are observable.

An inconsistent picture becomes apparent for the signal and background efficiencies, dependent on p_{T}. Stronger dependencies characterize the the low-p_{T} region, until a rather flat plateau is reached, above a muon $p_{\mathrm{T}} > 50$ GeV (Fig. 7.7a). The decreased signal efficiencies in the low-p_{T}



(a) high-p_T, non-prompt discriminator

(b) low-p_T, non-prompt discriminator

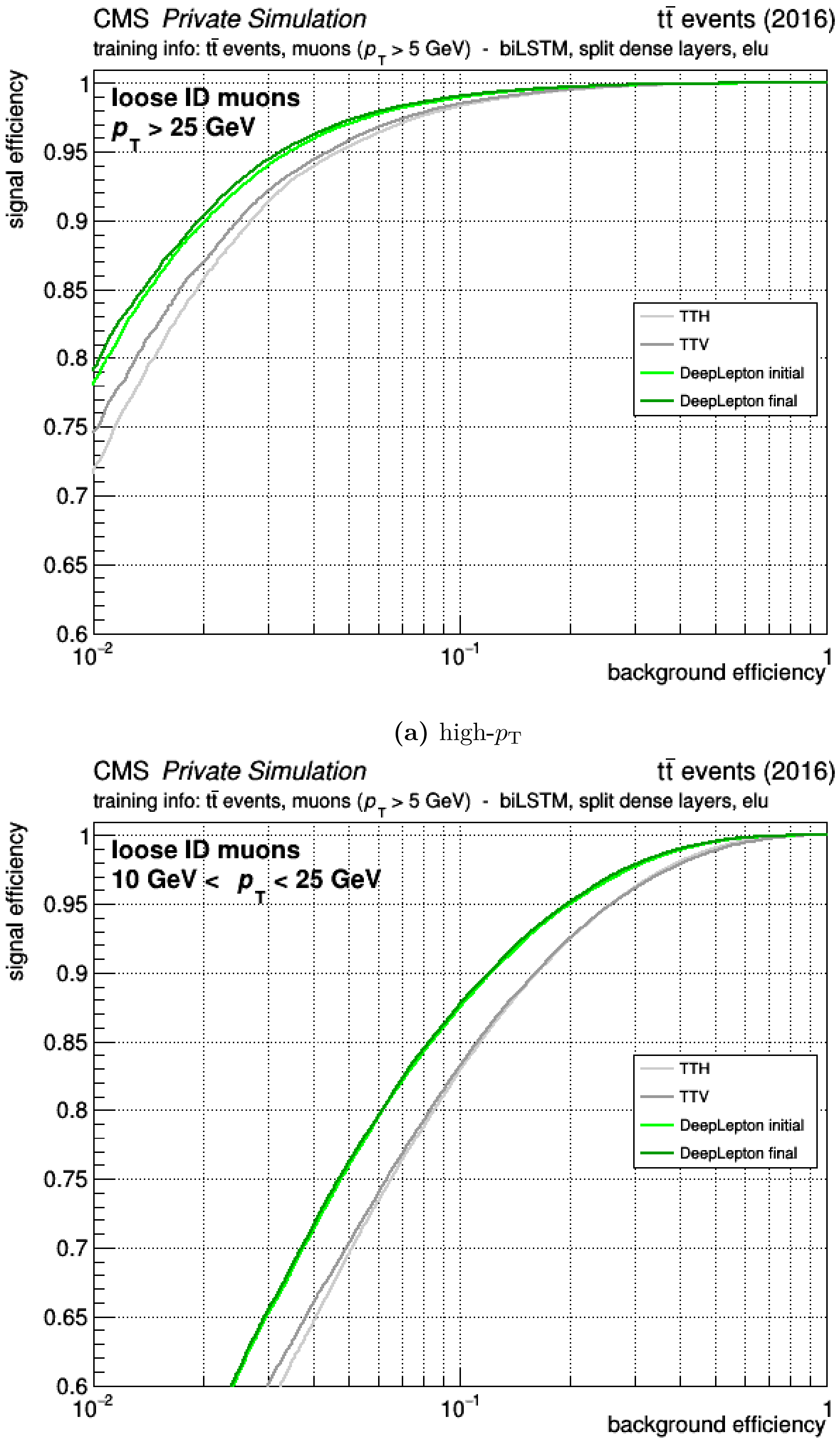
(c) high-p_T, fake discriminator

(d) low-p_T, fake discriminator

Fig. 7.3: The non-prompt **(a)-(b)** and fake **(c)-(d)** discriminator shapes, resulting from the final *DeepLepton* training model.

region can be explained by a difficult signal to background ratio and higher uncertainties in the test data.

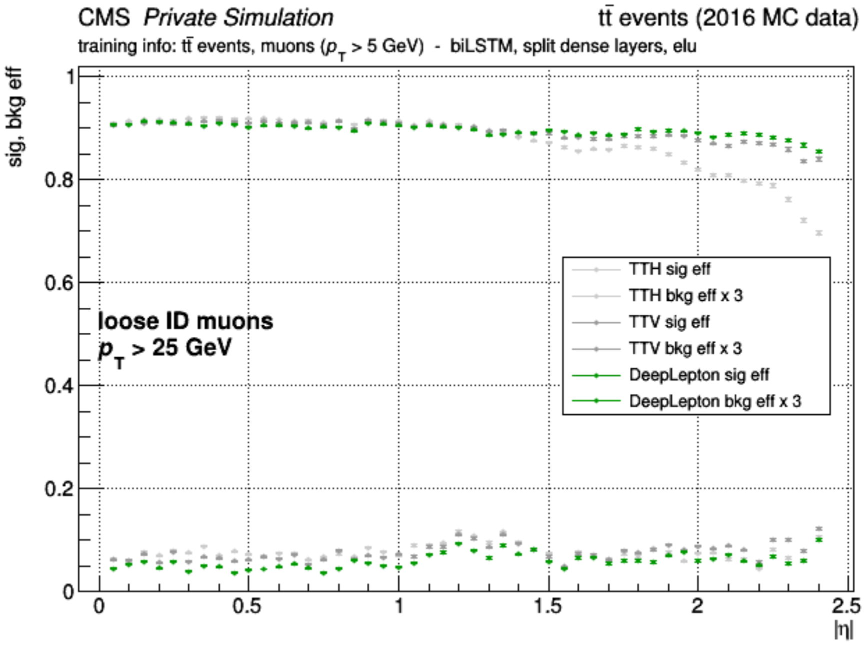
The trend versus p_T of the *DeepLepton* signal efficiency in Fig. 7.7b is a flatter than the corresponding *TTV* and *TTH* graphs. Regarding the background efficiency, the conditions are exactly vice versa. This behavior indicates, that for a certain working point with a given overall signal efficiency, the *DeepLepton* classifier yields more equally distributed signal efficiencies in the single p_T bins, than the competitive BDTs, while it is the other way round, in terms of the background efficiency. In order to understand this phenomenon in more detail, the evolution of the discriminator cut value, dependent on the respective working point and the observed p_T range is investigated in Sec. 8.1.5.



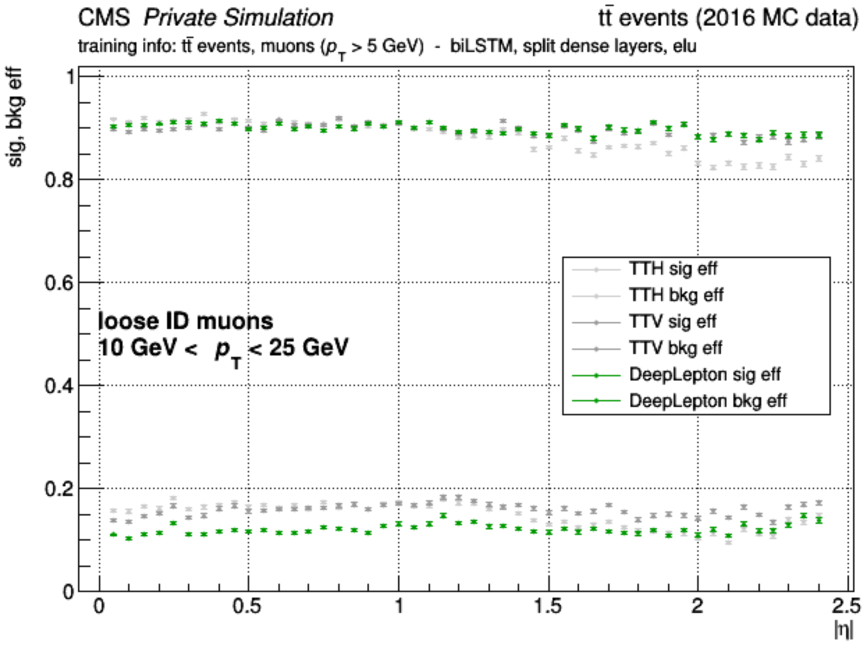
(a) high-p_{T}

(b) low-p_{T}

Fig. 7.4: The *DeepLepton* ROC curves, relating to the reference (labeled as “initial” in the plots) and the final network architecture, in comparison with the *TTV* and *TTH* BDTs. The *DeepLepton* classifier significantly improves the signal (prompt) muon identification in the test data sample, against the two other competitive IDs.

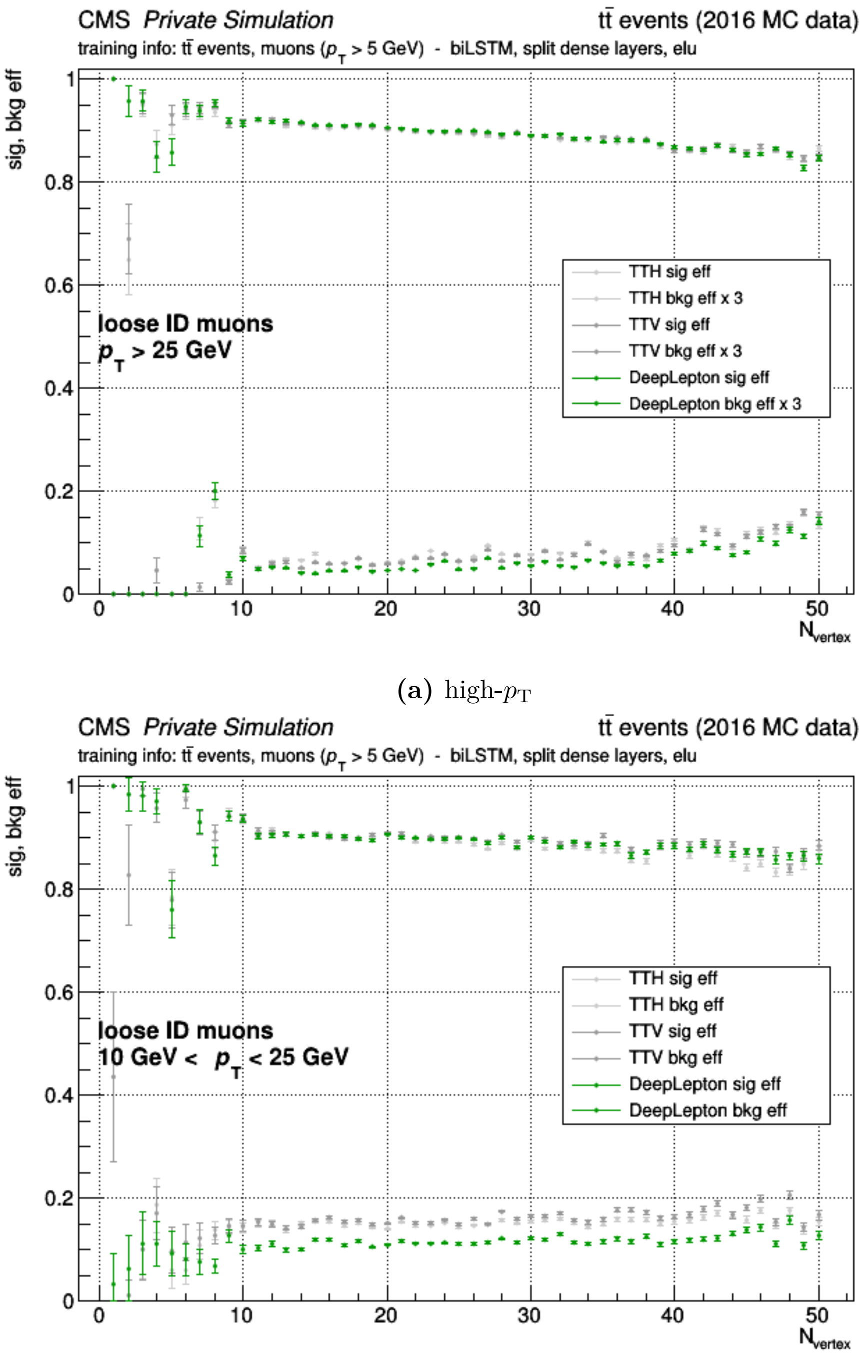


(a) high-p_T



(b) low-p_T

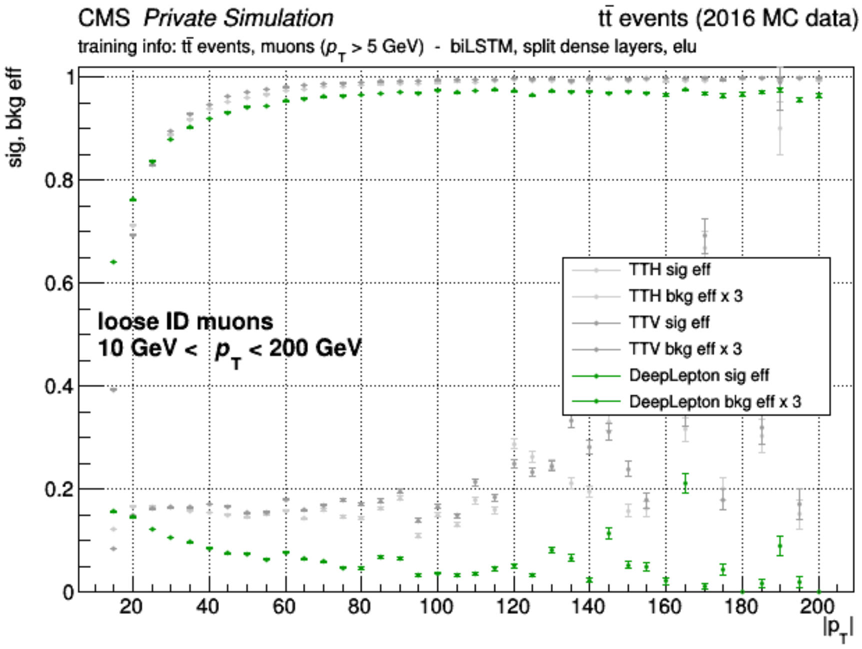
Fig. 7.5: The signal and background efficiencies, binned in η. The *DeepLepton* classifier, as well as the *TTV* and *TTH* MVAs, do not show significant dependencies, except for high-p_T muons in the far endcaps.



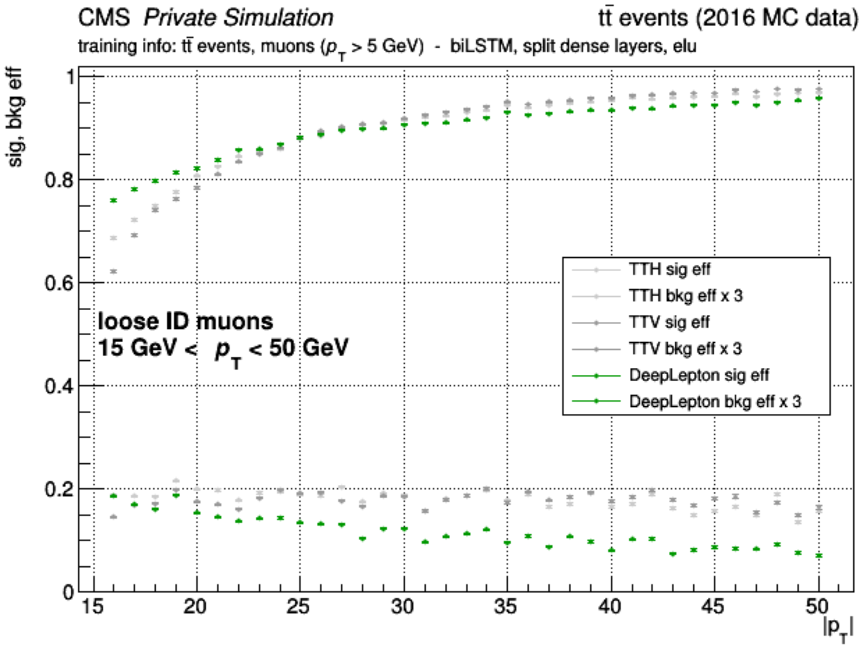
(a) high-p_{T}

(b) low-p_{T}

Fig. 7.6: The signal and background efficiencies, binned in N_{vertex}. Little dependencies are revealed for *DeepLepton*, as well as for TTV and TTH.



(a) 10 GeV $< p_{\mathrm{T}} <$ 200 GeV

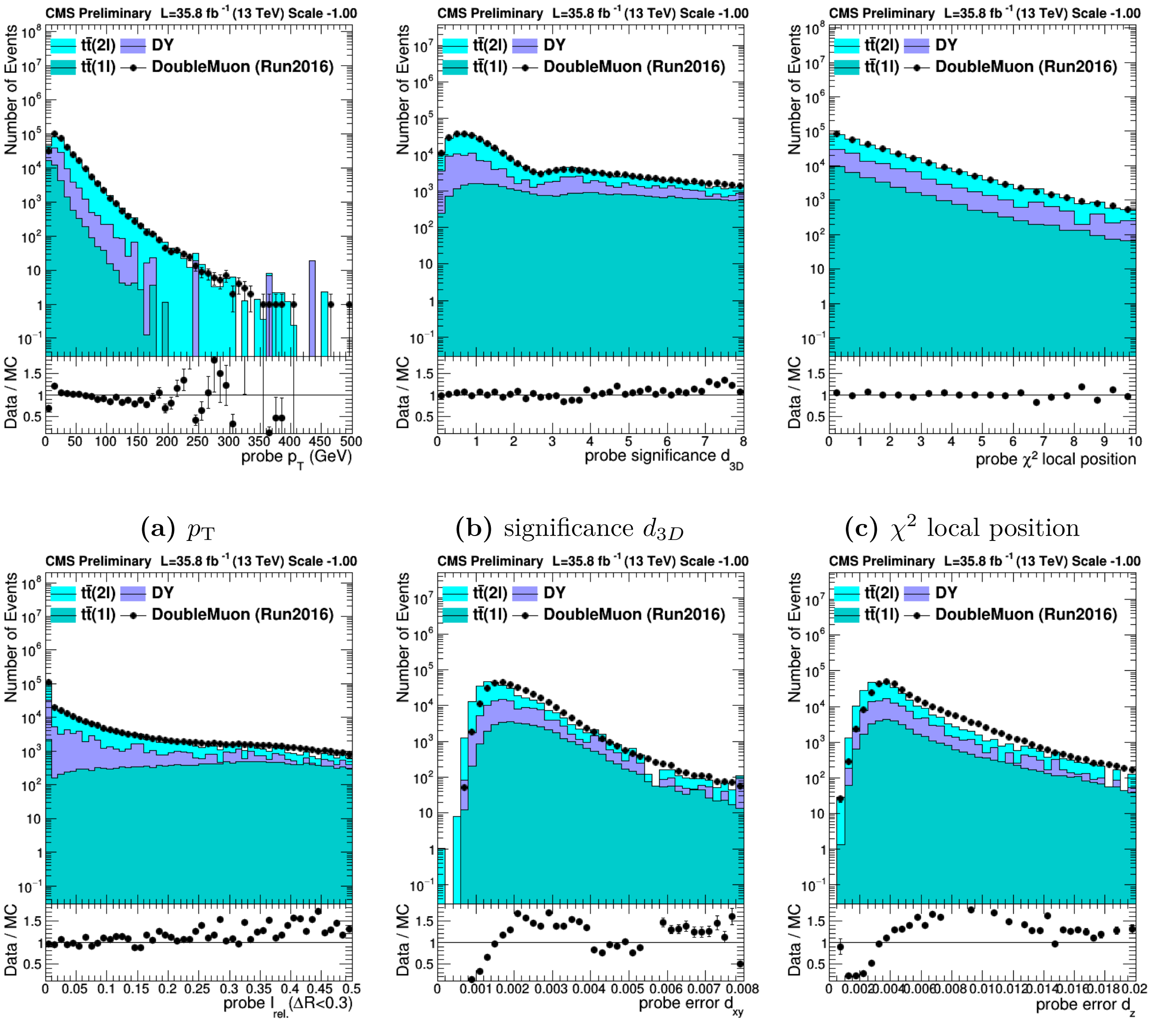


(b) 15 GeV $< p_{\mathrm{T}} <$ 50 GeV

Fig. 7.7: The signal and background efficiencies, binned in p_{T}. The left plot shows lower signal efficiencies in the low-p_{T} range and reasonable flat dependencies for a muon $p_{\mathrm{T}} > 50$ GeV. Unlike trends of the *DeepLepton* efficiencies, compared to *TTV* and *TTH*, are observable in the right plot.

7.2 Data-MC comparisons

The validity of the *DeepLepton* training model in real detector data, is shown by data-MC comparisons. The utilized samples, from the 2016 detector and MC data, are listed in Tab. 6.11. Comparisons are made for opposite charged dimuon final state events, which are obtained via the “Tag & Probe” method, according to Sec. 6.3.2. In this case, events, based on DY and $t\bar{t}$ processes, are treated separately, by applying appropriate selection criteria on the invariant mass of the muon-antimuon pair $m_{\mu^+\mu^-}$ (Eq. (6.25) and 6.26). The corresponding data-MC plots, with the distribution $m_{\mu^+\mu^-}$ in the DY and $t\bar{t}$ selection, are shown in Fig. 6.9a and 6.9b.



(a) p_{T}

(b) significance d_{3D}

(c) χ^2 local position

(d) $I_{\mathrm{rel.}}$

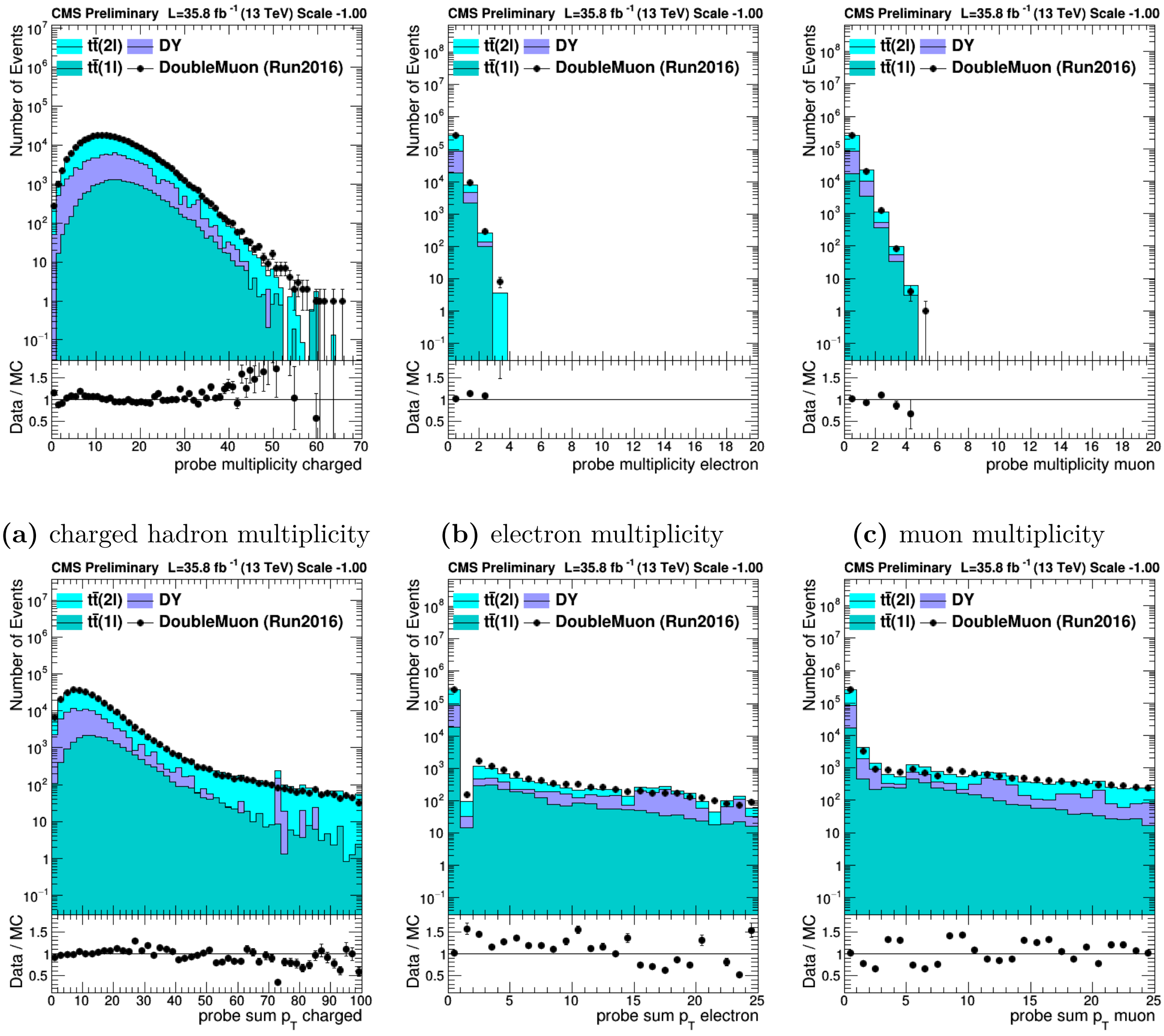
(e) uncertainty d_{xy}

(f) uncertainty d_z

Fig. 7.8: The data-MC plots of selected muon input training features. Most of the training variables are represented appropriately by the MC data samples **(a)-(d)**. However, **(e)** and **(f)** show disagreement and are therefore excluded from the final training of the *DeepLepton* classifier.

Firstly, data-MC plots are produced for the input training features of the *DeepLepton* classifier (Tab. 6.5 and Tab. 6.6). By means of these plots, it is checked, if the MC data adequately describes the distribution of the respective variables in real detector data. It turned out, that most of the lepton features are modeled very well, in particular: the lepton p_T, the significance of d_{3D}, or the local position χ^2, as well as the isolation variables (Fig. 7.8 (a)-(d)). However, data-MC comparisons, of the error of the impact parameters d_{xy} and d_z, which are plotted in Fig. 7.8 (e) and (f), are revealing some shifted distributions. Thus, the uncertainty of d_{xy} and d_z are excluded from the list of the training input features. A full set of the data-MC plots, related to the training features, is in the Appendix.

Concerning the vicinity features, sufficient congruities between the 2016 detector and MC data can be observed for the charged PF candidates and the SVs. Some shift occurs for the multiplicity of the photons. The respective plots are shown in (Fig. 7.9 and 7.10). Altogether, the distribution of the training features in real data, is represented sufficiently well.

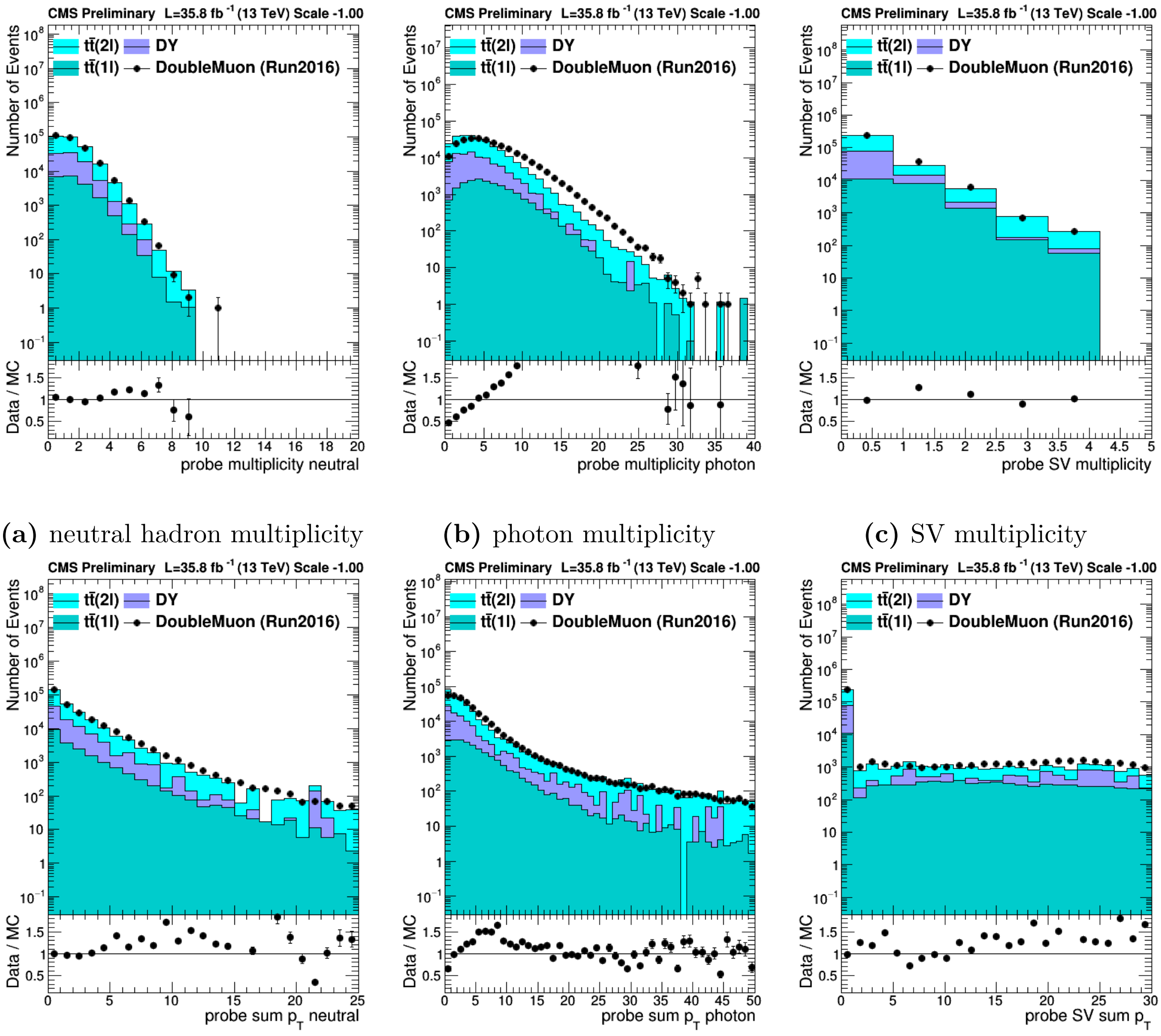


(a) charged hadron multiplicity **(b)** electron multiplicity **(c)** muon multiplicity

(d) charged hadron $\sum p_\mathrm{T}$ **(e)** electron $\sum p_\mathrm{T}$ **(f)** muon $\sum p_\mathrm{T}$

Fig. 7.9: The data-MC plots of charged PF candidate species.

Secondly, global event features are examined. The p_T of the leading and the subleading jet, the $\not{E}_\mathrm{T}$ and its distribution in ϕ, and the jet and vertex multiplicity, N_jets and N_vertex, are shown in Fig. 7.11. Again, no significant discrepancies between the real and simulated data samples are observed.

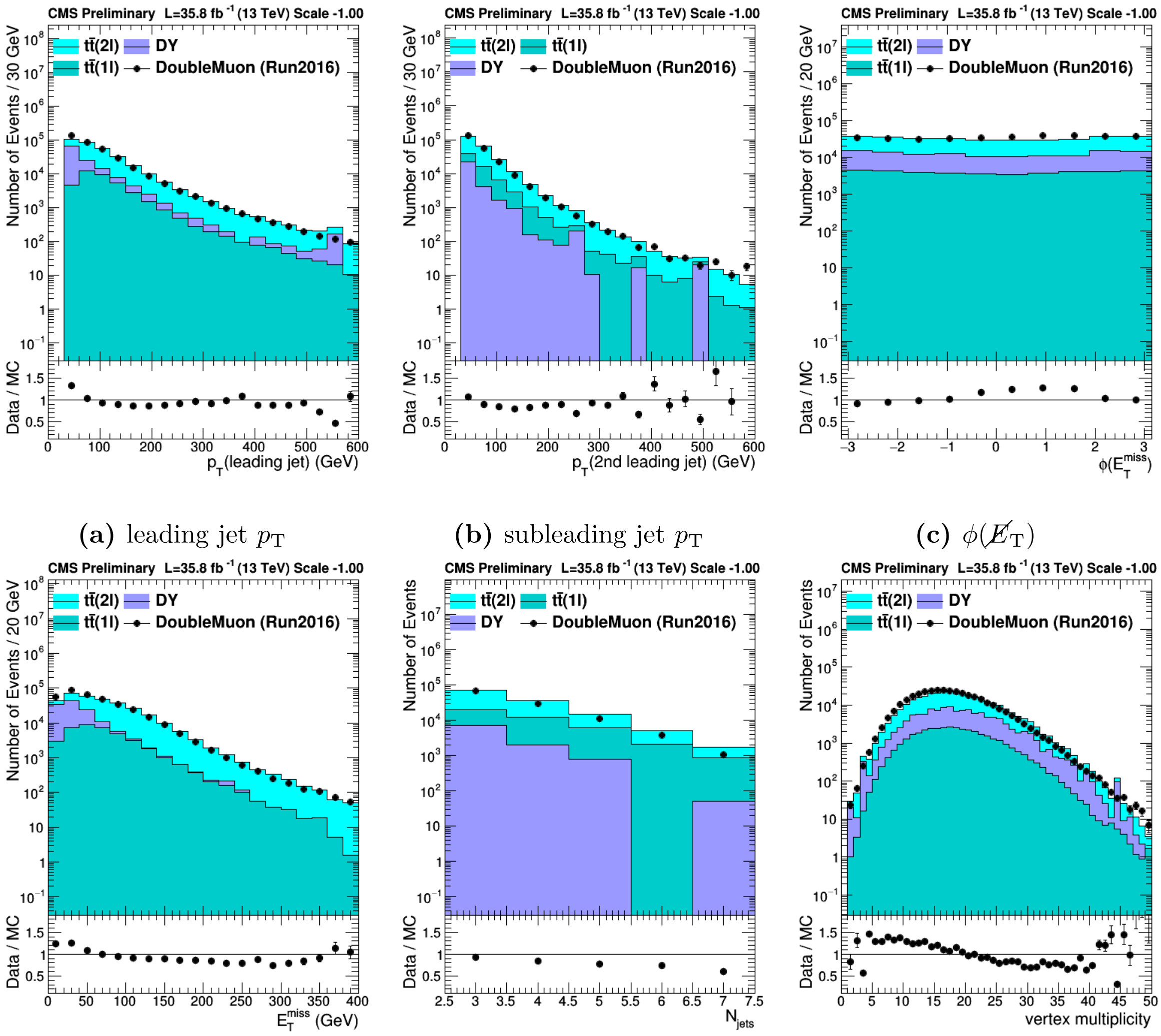


(a) neutral hadron multiplicity **(b)** photon multiplicity **(c)** SV multiplicity

(d) neutral hadron $\sum p_{\mathrm{T}}$ **(e)** photon $\sum p_{\mathrm{T}}$ **(f)** SV $\sum p_{\mathrm{T}}$

Fig. 7.10: The data-MC plots of neutral PF candidate species and the SVs.

Finally, the *DeepLepton* training model is deployed on the full events of the detector data. The resulting discriminator shapes are compared with the ones, that could be achieved in MC data (Fig. 7.12). Due to the adequate agreement between detector and simulated data, a similar performance of the *DeepLepton* classifier is expected for the real data. Future scale factor measurements will quantify the residual discrepancy.



(a) leading jet p_{T}

(b) subleading jet p_{T}

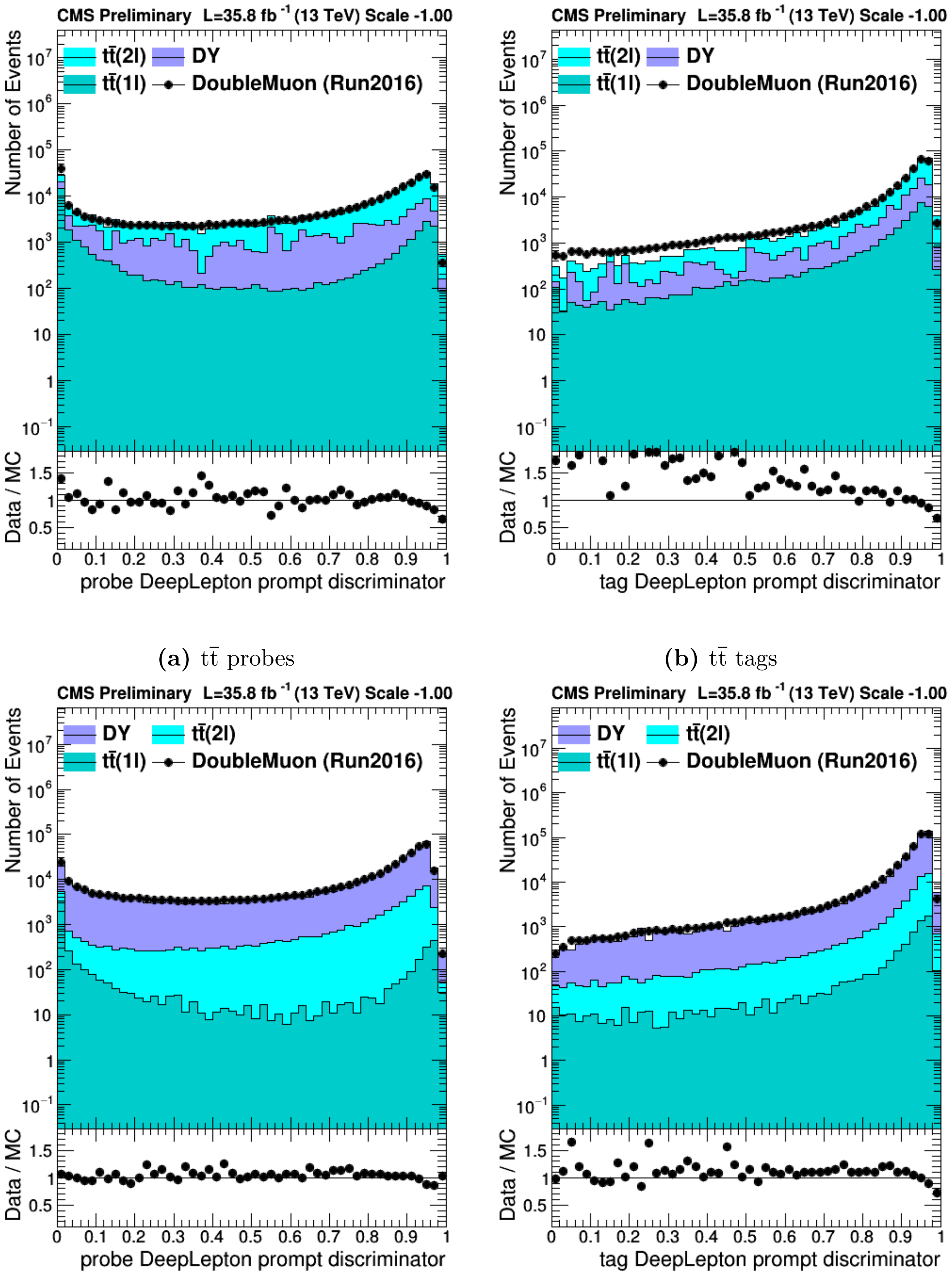
(c) $\phi(\not{E}_{\mathrm{T}})$

(d) $\not{E}_{\mathrm{T}}$

(e) N_{jets}

(f) N_{vertex}

Fig. 7.11: The data-MC plots of the global event features.



(a) $t\bar{t}$ probes

(b) $t\bar{t}$ tags

(c) DY probes

(d) DY probes

Fig. 7.12: The prompt discriminator shapes for “Tag & Probe” muons in data-MC comparisons.

Chapter 8

Discussion

8.1 Results for different network configurations

In the following the milestones in the development of the *DeepLepton* training model, as well as tests of alternative training models are discussed. This includes tests of various network architectures, the estimation of the contribution of the low-level vicinity features to the performance of the *DeepLepton* classifier, the evolution of the discriminator cut value, a check for overfitting, as well as the performance of the training model in different test data samples.

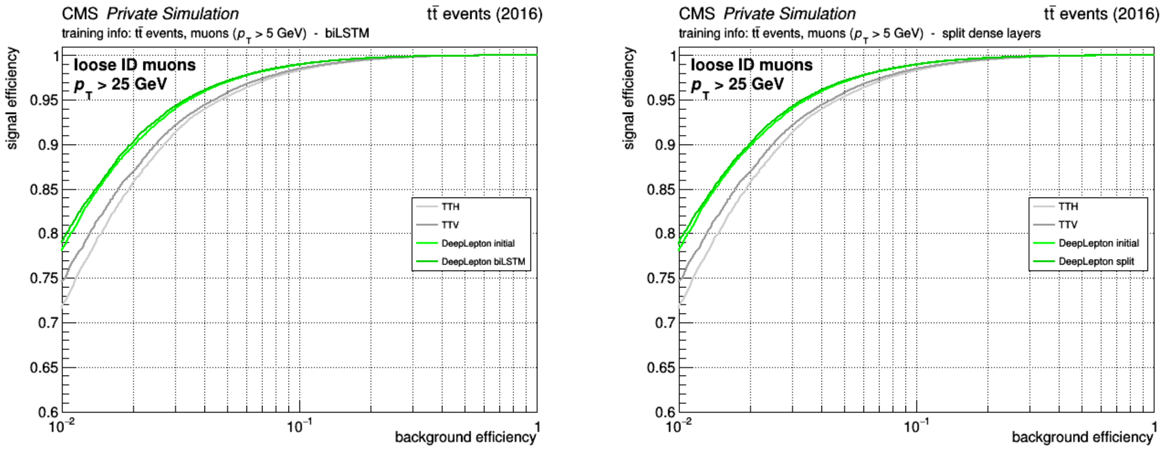
8.1.1 The *DeepLepton* architectures

The development of the network architecture of the *DeepLepton* classifier has been an ongoing process, during the experimental part of this master thesis. Earlier approaches are closely related to the architecture, which is used in the *DeepJet* framework [6] and refer to the reference architecture (Fig. 6.3), as described in Sec. 6.2. Modifications of this reference model are tested as well. Improvements of the performance are achieved by the biLSTM units, as well as by an alteration of the FC neural network. As already mentioned in Sec. 7.1, separate dense layers are applied for the lepton features and the vicinity features, before both fractions are processed together within a final FC neural network (Fig. 7.1).

On the one hand, the network model seems to benefit from the bidirectional processing of the inputs, which are fed into the biLSTM units (Fig. 8.1a). On the other hand, the splitting of FC neural network enables to slightly improve the efficiencies in terms of high-p_T muons (Fig. 8.1b). No significant improvements can be observed in the low-p_T region, in these modifications.

Attempts to modify the CNN part of the architecture do not yield any performance improvements. The application of pooling layers, which are a very common tool inside CNNs, results in a performance reduction (Fig. 8.2a). The task of the pooling layers in a CNN is to downsize the data, so as to enable the learning of high-level features. However, in case of the *DeepLepton* classifier, the CNN is used to processes sequential PF candidates and SVs via 1 × 1 convolutional layers. This means, that only one PF candidate is considered at a time [110]. Hence, useful information gets lost by applying pooling layers, and the DNN performs worse than without the pooling layers.

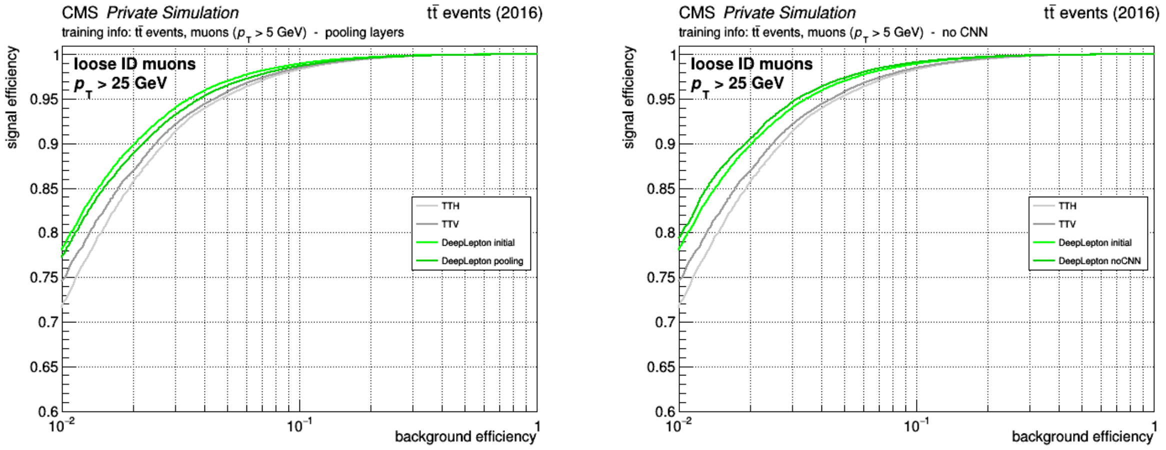
In principle, the LSTM units can handle the input data without applying the convolutional layers at all. But the 1D CNN ensures that the same transformation is applied to each PF candidate [110]. Tests without the CNN surprisingly yield a performance boost (Fig. 8.2b), but show bumpy discriminator shapes, which are not desirable (Fig. 8.3). Nevertheless, there seems to be still potential for an improvement of the efficiencies.



(a) biLSTM units

(b) split dense layers

Fig. 8.1: Dense layer and RNN architecture tests.



(a) additional pooling layers in the CNN

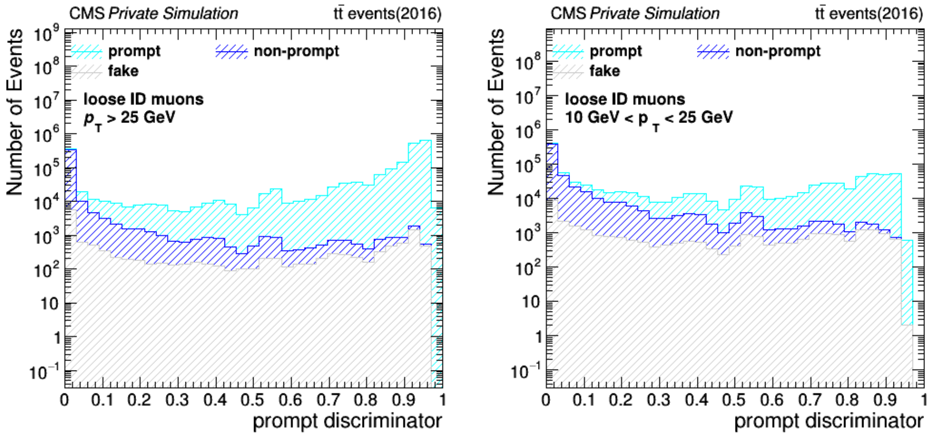
(b) no CNN at all

Fig. 8.2: The training model tests, concerning an alternative and an omitted CNN architecture. Additional pooling layers do not yield a performance gain **(a)**, while a potential for further improvements is revealed by the omission of the convolutional layers **(b)**.

8.1.2 Impact of the vicinity features

A training model, using solely the global lepton features, is considered, in order to estimate the contribution of the vicinity features to the performance of the *DeepLepton* classifier. Two separate models are tested. One model makes use of all global lepton features, which are applied in *DeepLepton*. Another model uses those features that are also deployed for the training of the *TTV* and *TTH* BDTs. The former model performs similar to the *TTV* MVA (Fig. 8.4a), whereas the latter model covers the performance of the *TTH* MVA (Fig. 8.4b). Moreover, there are only slight differences noticeable between the two models for high-p_T muons.

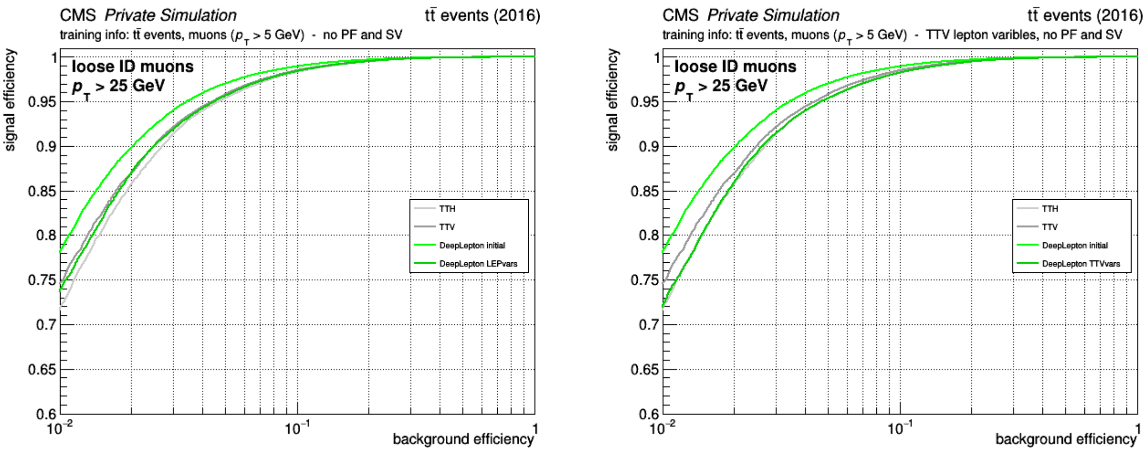
The majority of the lepton information is apparently already covered by features utilized in the *TTV* training. Reproducing the competing MVAs in this way ensures that the training of *DeepLepton* is close to optimal. Most of the performance gain of the *DeepLepton* classifier derives from the low-level vicinity features that are included additionally.



(a) high-p_T

(b) low-p_T

Fig. 8.3: The bumpy discriminator shapes become apparent, when the training in performed without CNNs.



(a) *DeepLepton* muon features

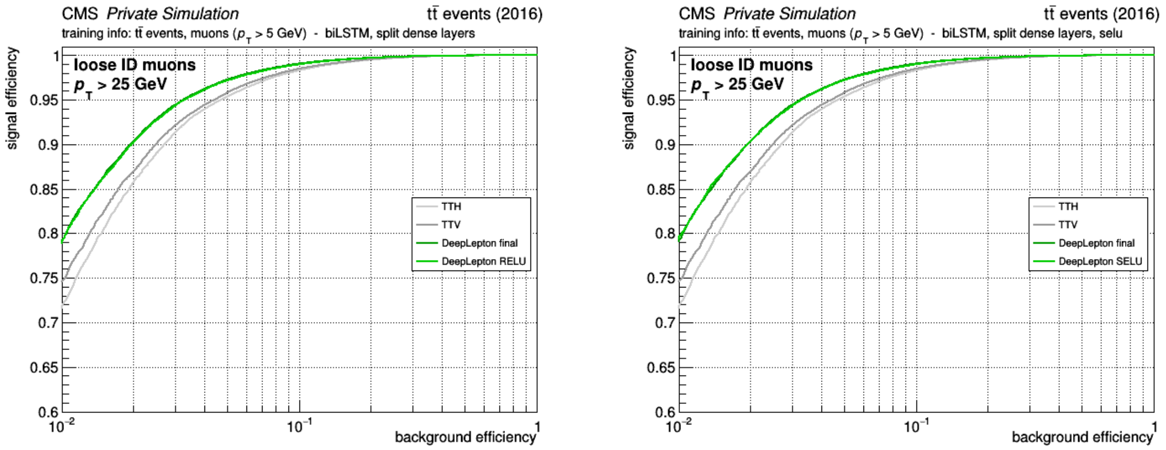
(b) *TTV* muon features

Fig. 8.4: Training tests using only the global lepton features.

8.1.3 Dependence on the activation function

A further small gain in performance results from the usage of the ELU and SELU activation functions, in comparison to the ReLU (Fig. 8.5a). In contrast to the ReLU, the ELU is able to output negative values. Therefore, it do not suffer immediately from the dying ReLU problem [129], where nodes stop responding to the neural network, because of a vanishing gradient (Sec. 5.2.1). The SELU activation function is a slight modification of the ELU [161].

Both, ELU and SELU yield nearly identically results (Fig. 8.5a). The SELU has slight advantages for tighter working points, while the ELU is marginally more qualified for loose working points. Because of technical issues in the deployment of the SELU, the ELU is selected for the final configuration of the training model.



(a) ELU vs ReLU

(b) ELU vs SELU

Fig. 8.5: The comparison of different activation functions. Marginal improvements can be achieved by the usage of the ELU and SELU, against the ReLU.

8.1.4 Training loss and overfitting

In general, dropout (0.5) [125] is applied as a generalization tool, in order to prevent the *DeepLepton* neural network from overfitting.

Furthermore, the evolution of the training and validation loss over the epochs, is monitored during the training procedure. Overfitting can be recognized at an early stage, if the validation loss starts to increase, whereas the training loss continues to decrease. Regarding the *DeepLepton* network training, a converging training and validation loss can be observed. A graph of the two losses, as well as of the tuning of the learning rate, during the *DeepLepton* network training, is shown in Fig. 8.6.

In contrast to the validation set, the test data set is applied to evaluate the training model afterwards. At this stage, overfitting can be identified, by a training model, that performs worse on the test sample, than on the training sample. The ROC curves of the *DeepLepton* classifier, evaluated on the training data set, are plotted in Fig. 8.7. A comparison with the results of the evaluation on the test data set in Fig. 7.4, reveals a very similar performance. Thus, no indication of overfitting can be found in the *DeepLepton* classifier training.

8.1.5 Evolution of the discriminator cut value

The p_T dependence of the optimal *DeepLepton* discriminator cut value is studied in more detail in order to understand the dependencies in p_T which are presented in Sec. 7.1. ROC curves for different p_T ranges are calculated with a color-encoded representation of the discriminator cut values corresponding to the respective working points in the ROC curve. This complements Fig. 7.7b.

For the *TTV* classifier a discriminator cut that depends on p_T flattens the background efficiencies (Fig. 8.8a). The optimal discriminator cut value to achieve a given background efficiency is found to be rather stable in p_T, whereas it is p_T dependent in terms of a given signal efficiency.

The working point dependent evolution of the discriminator cut values of the *DeepLepton* classifier in the various p_T ranges is shown in (Fig. 8.8b). This classifier provides a more hybrid behavior.

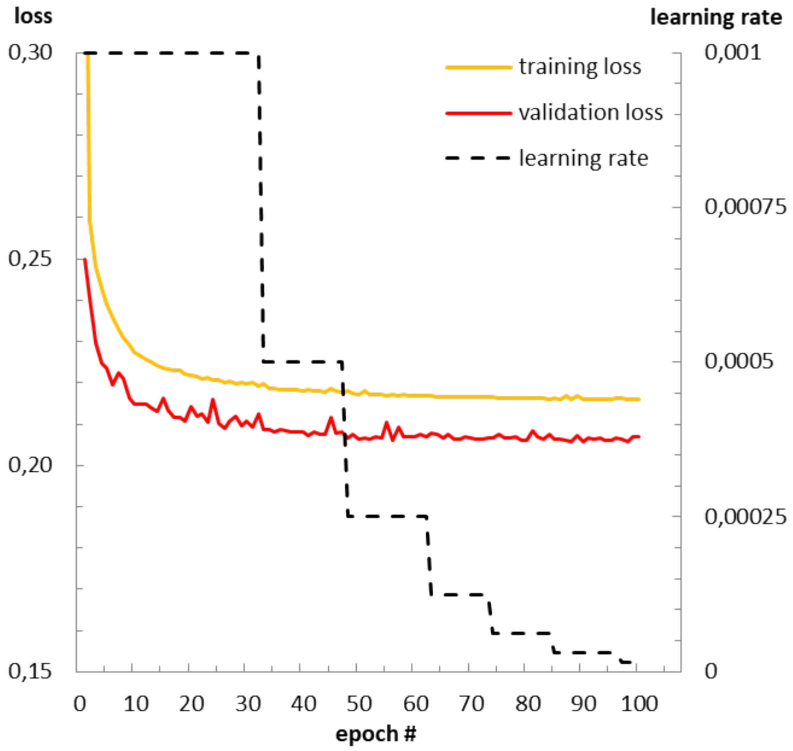


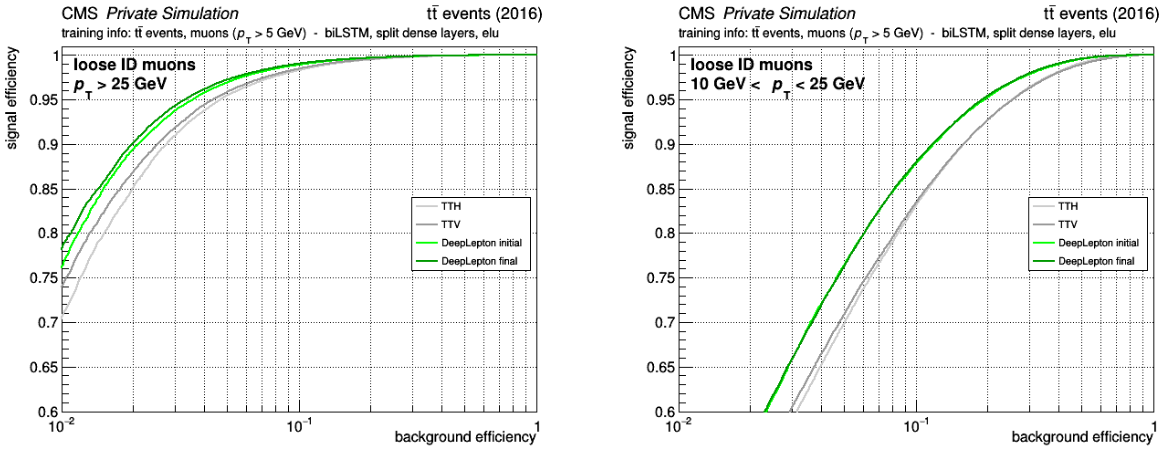
Fig. 8.6: The evolution of the training and validation loss in the *DeepLepton* neural network. The utilized neural network API Keras [164] splits the data set which is intended for the network training into two subsets: one real training data set (85%) and one validation data set (15%). The training data set serves as the input data for the neural network, while the validation data set is utilized solely for the validation of the network model, during the training procedure. The learning rate is adjusted after five epochs of no progress which accounts for the spikes in the validation loss.

The discriminator cut value slightly varies with p_T, either for a given background efficiency, or for a given signal efficiency. Hence, it is less stable when compared to the *TTV* BDT. However, the *DeepLepton* classifier gives a higher performance in muon classification. Applying two cut values for the low-p_T and the high-p_T region can easily overcome the residual dependence.

8.1.6 Tests on reference and output classes

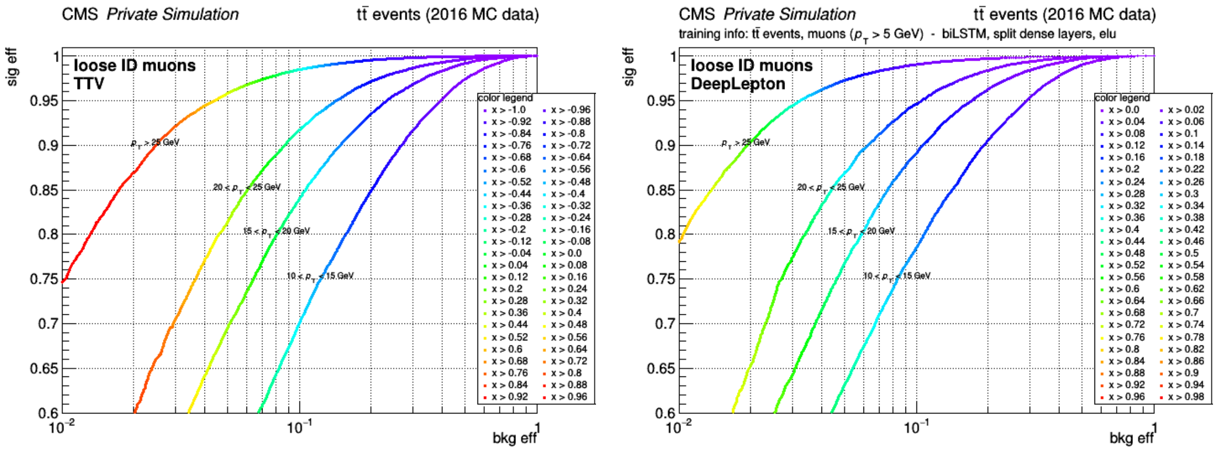
Usually the non-prompt muons are used as a reference class for the reweighting in p_T and η of the training input data (Sec. 6.1.3.3). An alternative test training model, where the reference class is set to the prompt muons, gives a worse performance (Fig. 8.9a). Furthermore, it becomes apparent, that extremely high discriminator cut values have to be chosen, in order reach sensible working points (Fig. 8.9b). Such a behavior is undesirable. Thus, the selection, of non-prompt muons as a reference class, is considered to be more beneficial regarding both, the performance of the training model and the evolution of the discriminator cut value.

A balanced signal (prompt leptons) to background (non-prompt or fake leptons) ratio is considered for the *DeepLepton* training data (Eq. (6.2)). However, strictly speaking, a balanced ratio of all trained output classes, would be ideal, in terms of a multiclassification task. Adjusting the lepton multiplicity, of each class, to the level of the limiting class of fake muons, would involve a



(a) high-p_T **(b)** low-p_T

Fig. 8.7: The performance of the *DeepLepton* classifier in the utilized training data set.



(a) *TTV* **(b)** *DeepLepton*

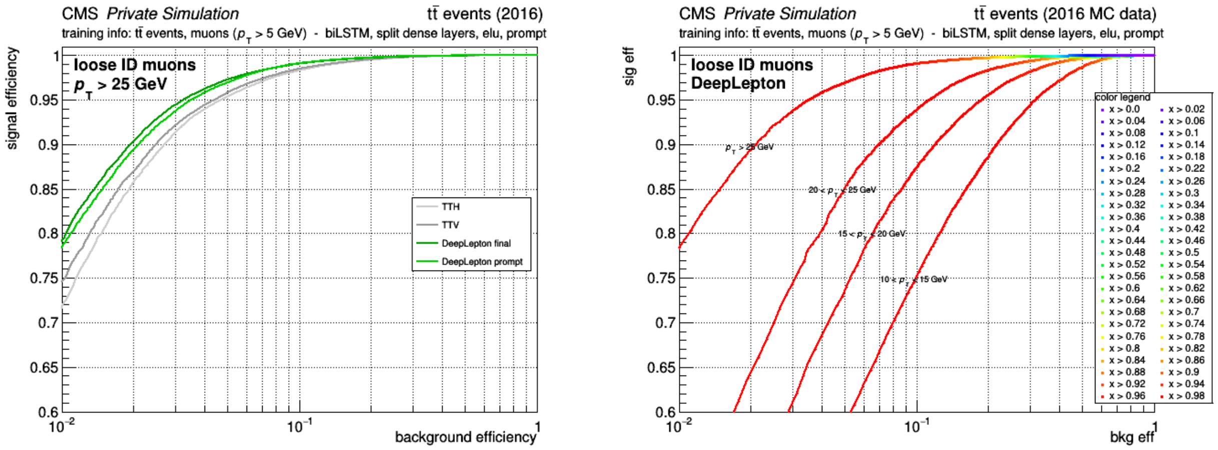
Fig. 8.8: The evolution of the discriminator cut value in the *DeepLepton* and *TTV* classifiers, dependent on different working points and p_T selections.

tremendous reduction of the amount of training data. Hence, a simplified training model with only two output classes is tested, where the non-prompt and fake muons are merged into a single background class, referred to as not-prompt. This ensures a balanced ratio of the output classes, without loosing any training data.

This simplification yields a slightly worse performance of the training model for loose working points in both, the high-p_T (Fig. 8.10a) and the low-p_T region (Fig. 8.10b). Thus, merging non-prompt, but real muons, together with fake muons is not completely satisfying.

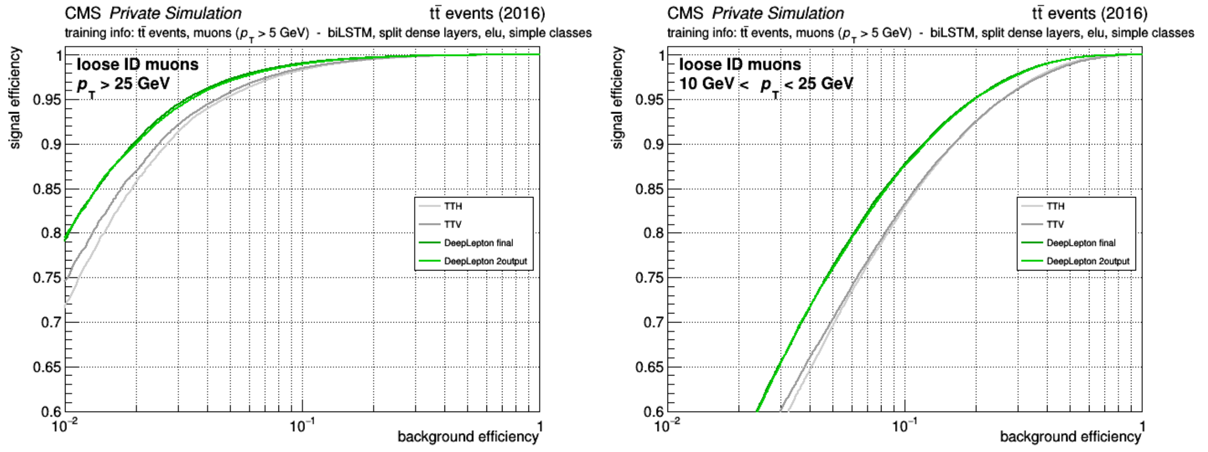
8.1.7 Dependence on training and test samples

Regarding the evaluation of the *DeepLepton* training model on a test data set, the main question is sample dependence. It is furthermore interesting, how a classifier that is trained on certain MC samples performs on other MC samples.



(a) high-p_T ROC **(b)** color-encoded ROC curves

Fig. 8.9: The selection, of prompt muons as a reference class for the p_T and η reweighting during the training, downgrades the performance of the *DeepLepton* classifier **(a)**. The color-encoded evolution of the discriminator cut value, in different p_T selections, is shifted towards very high values **(b)**.



(a) high-p_T **(b)** low-p_T

Fig. 8.10: A training with just two output classes (prompt and not-prompt) yields worse performance in the ROC curves.

To this end a "matrix comparison" is made. Three different samples are considered: the DY+QCD sample (according to Tab. 6.1), the tt̄ sample (as given in Tab. 6.2), and a combination of these two samples. Each sample is split into a training (80%) and an independent test data set (20%). Then, three separate classifiers (one per training sample) are trained, with the final *DeepLepton* network architecture, according to Ch. 7. However, muons with a p_T higher than 15 GeV, instead of 5 GeV, are considered for these training procedures. The matrix comparison, for muons in the low-p_T range, is shown in Fig. 8.11.

Firstly, the training on the DY+QCD sample yields reasonable results for the DY+QCD and the combined test sample when compared to TTV and TTH. In contrast, its performance in the

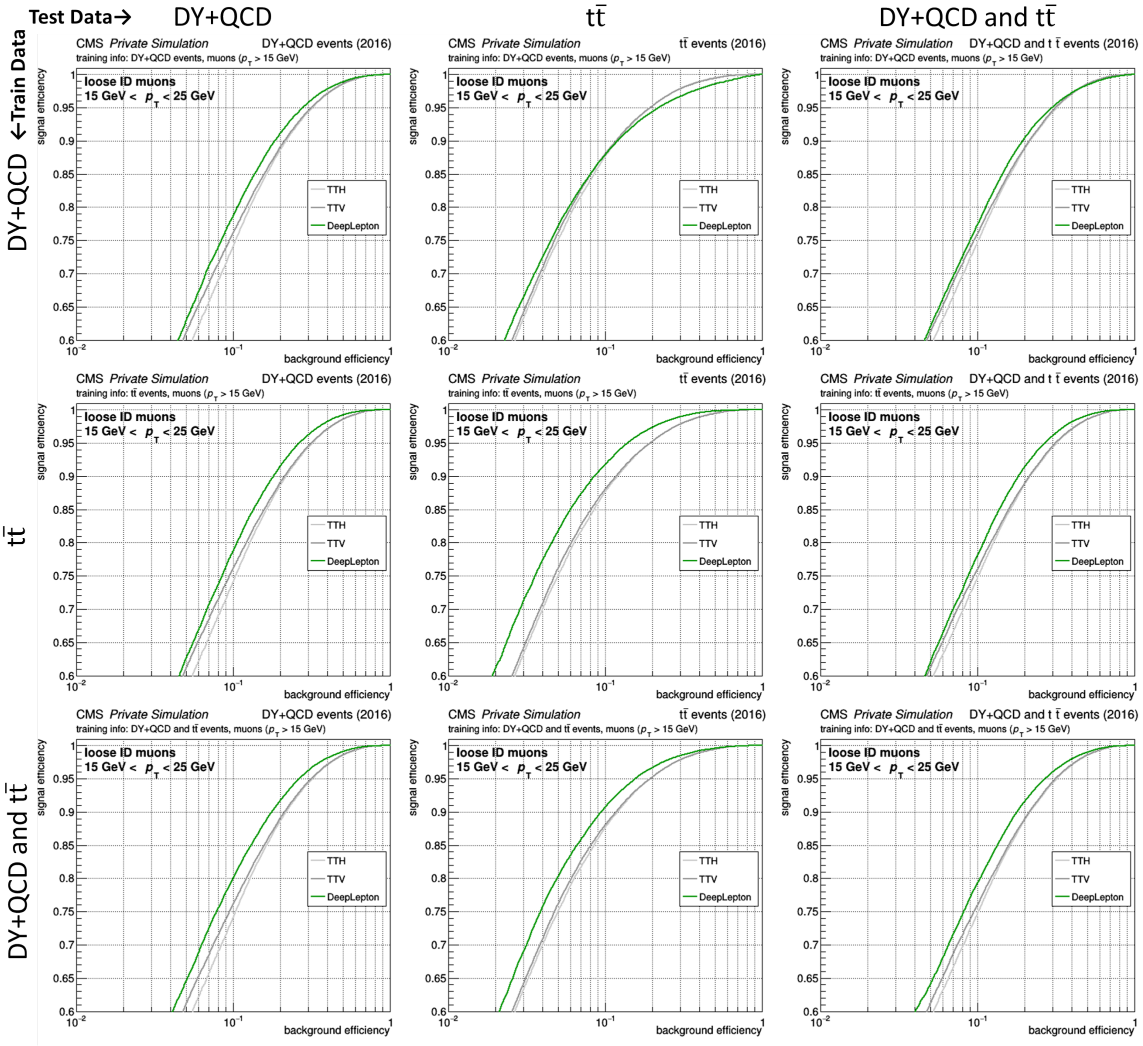


Fig. 8.11: Matrix comparison for the low-p_{T} range. Training models, based on different training samples, are evaluated on different test data samples.

$t\bar{t}$ test sample is moderate. Good performance is obtained for tight working points, but weak performance is observed for loose working points.

Secondly, the *DeepLepton* classifier, that is based on the $t\bar{t}$ training sample, provides good results for all three test samples when compared to TTV and TTH. It even outperforms the *DeepLepton* classifier that is trained on the DY+QCD training sample.

Thirdly, the combination of the DY+QCD and the $t\bar{t}$ training samples can further improve the performances in the test samples, except for the $t\bar{t}$ test sample. In the latter case, the classifier still outperforms the TTV and TTH approaches, but yields worse results, compared to the *DeepLepton* classifier, that is trained solely on the $t\bar{t}$ sample. Increasing the training data set by adding muons from the DY and QCD processes seems to have counterproductive effects, in terms of muon identification in the $t\bar{t}$ test sample.

The incompatibility between the DY+QCD and the $t\bar{t}$ samples becomes even more distinct in the matrix comparison for the high-p_{T} muons, which can be found in the appendix. One reason might be, that the QCD samples with cross sections, that are varying over a large order of magnitudes, from sample to sample, are not optimal for the learning process in the neural network training, since they are binned in p_T (Tab. 6.1).

8.2 Challenges of the training and outlook

One problem, that appeared in initial training attempts was an oscillating validation loss and validation accuracy, over the epochs in the training procedure. This issue could be overcome, by adjusting the model hyper-parameters, like the learning rate (0.001), momentum (0.2) and dropout (0.5), as well as by limiting the number of PF candidates per species fore each lepton (Fig. 6.4). The latter measure is inspired by [110].

Another point is the necessity of cutting at extremely high discriminator values, in order to achieve sensible working points. This problem occurs, when prompt muons are selected as a reference class for the p_T and η reweighting of the training data. It does not occur when the non-prompt muons are the reference class (Sec. 8.1.6).

Ongoing studies at the HEPHY in the work group of Robert Schöfbeck et al., are currently addressing the training and evaluation of the *DeepLepton* framework on electrons, as well as for 2017 and 2018 data, to complete the picture of this novel lepton ID. Furthermore, scale factor measurements have to be done, as a future task, to asses the performance of the *DeepLepton* classifier in real data.

Whether the potential of the *DeepLepton* classifier is already fully exploited or not, is one of the open questions, that arise for this master thesis. As discussed in Sec. 8.1.1, there might be still potential for a small performance gain by further optimizing the network architecture. The implementation of conceptually new network types is a topic for future work. An example could be the usage of GNNs from geometric deep learning [154] in order to learn the geometric structure and substructure of particles in the vicinity of the lepton.

Chapter 9

Conclusion

In this work, I developed, described and evaluated the novel classification algorithm *DeepLepton*. Earlier lepton ID criteria only take into account the high-level features of the lepton. In contrast, I also include the particles in the vicinity of leptons, inspired by the *DeepJet* framework [6]. Various features of the particle candidates and the secondary vertices in the vicinity of the lepton are considered as additional input for the network. It is found that the inclusion of the particle activity provides a significant performance boost when compared to existing lepton identification algorithms.

DeepLepton is the first approach for lepton identification that utilizes advanced deep learning techniques. The architecture of the *DeepLepton* neural network consists of CNNs, biLSTM units, as well as FC neural networks. Different architecture configurations and model settings are tested in order to adapt and refine the training model.

First results of the *DeepLepton* classifier in simulated data yield a promising performance for muon identification. Considering working points with a signal efficiency of around 80%, *DeepLepton* achieves an improvement of the background efficiency of about 1-2% for low-p_T muons (10 GeV $< p_\mathrm{T} <$ 25 GeV) and 0.1-0.2% for high-p_T mouns ($p_\mathrm{T} >$ 25 Gev) when compared to existing multivariate identification criteria. For a background efficiency in the range of 1-3%, values frequently used in data analysis, the increase in the signal efficiency amounts up to 4-5% for both p_T regions and the performance gain even multiplies when final states with higher muon multiplicity are considered. *DeepLepton* thus provides a corner stone for the quality of many future data analysis with the CMS experiment.

Chapter 10

Acknowledgements

Within almost one year of intensive work on this master's thesis, I gained a lot of experience in various fields, from the particle physics at the LHC, via the big data analysis by means of deep learning techniques, through to writing a scientific documentation. However, this would not have been possible without the help and the encouragement of quite a few people.

First, I cordially thank Jochen Schieck for supervising this master's thesis. Then, I would particularly like to thank Robert Schöfbeck for going to great effort to support my work and encouraging me to make the best of this project. Furthermore, I give thanks to the whole CMS analysis group at the HEPHY, under the direction of Wolfgang Adam, which members always had time for suggestions concerning problems encountered.

Moreover, it has to be thanked Dietrich Liko, for his effort regarding the solution of technical issues, that occurred during this work. Many thanks also go to Jan Kieseler, who introduced me to the *DeepJet* framework and gave advice in order to set the framework going.

Not forgetting, that I owe a great time in Vienna to my fellow students Michi, Tobi and Bernhard, as well as to the "STUWO-gang" from the dormitory.

Finally, I would like to say a big thank you to my parents, my brother and my lovely Lisa, for giving me strength an keeping me grounded, by their love and support.

Acronyms

AI artificial intelligence

ALICE ”A Large Ion Collider Experiment”

AOD analysis object data

API application programming interface

ATLAS ”A Toroidal LHC Apparatus”

AUC area under the curve

BDT boosted decision tree

BSM beyond standard model

CERN *”Conseil Européen pour la Recherche Nucléaire”*

CKM Cabibbo–Kobayashi–Maskawa

CMS Compact Muon Solenoid

CNN convolutional neural network

CP Charge and parity

CPU central processing unit

CRAB CMS Remote Analysis Builder

CSC Cathode strip chamber

CSV Combined Secondary Vertex

CUDA Compute Unified Device Architecture

DAQ data acquisition system

DNN deep neural network

DT Drift tube chamber

DY Drell-Yan

ELU exponential linear unit

FC fully connected

FPR false positve rate

FSR final state radiation

GAN generative adversarial network

Geant geometry and tracking

GNN graph neural network

GPU graphics processing unit

GRU gated recurrent unit

GSF Gaussian sum filter

GUT grand unified theory

HEP high energy physics

HLT high-level trigger

ID identification criterion

ISR initial state radiation

JEC jet energy correction

L1 level-1

LHC Large Hadron Collider

LHCb ”Large Hadron Collider - beauty”

LHCf ”Large Hadron Collider forward”

LSTM Long Short Term Memory

MC Monte-Carlo

MoEDAL ”Monopole and Exotics Detector at the LHC”

MVA multivariate analysis

MVAs multivariate analyses

PDF parton distribution function

PF particle flow

POG physics object group

PS Proton Synchrotron

PSB Proton Synchrotron Booster

PUPPI pileup per particle identification

PV primary vertex

QCD quantum chromodynamics

ReLU rectified linear unit

RF radio frequency

RMS root mean square

RNN recurrent neural network

ROC Receiver Operating Characteristic

RPC resistive plate chamber

SELU scaled exponential linear unit

SEM standard error of the mean

SM standard model of particle physics

SPS Super Proton Synchrotron

SUSY Supersymmetry

SV secondary vertex

SVs secondary vertices

TMVA Toolkit for Multivariate Data Analysis

TOTEM ”TOTal Elastic and diffractive cross section Measurement”

TPR true positve rate

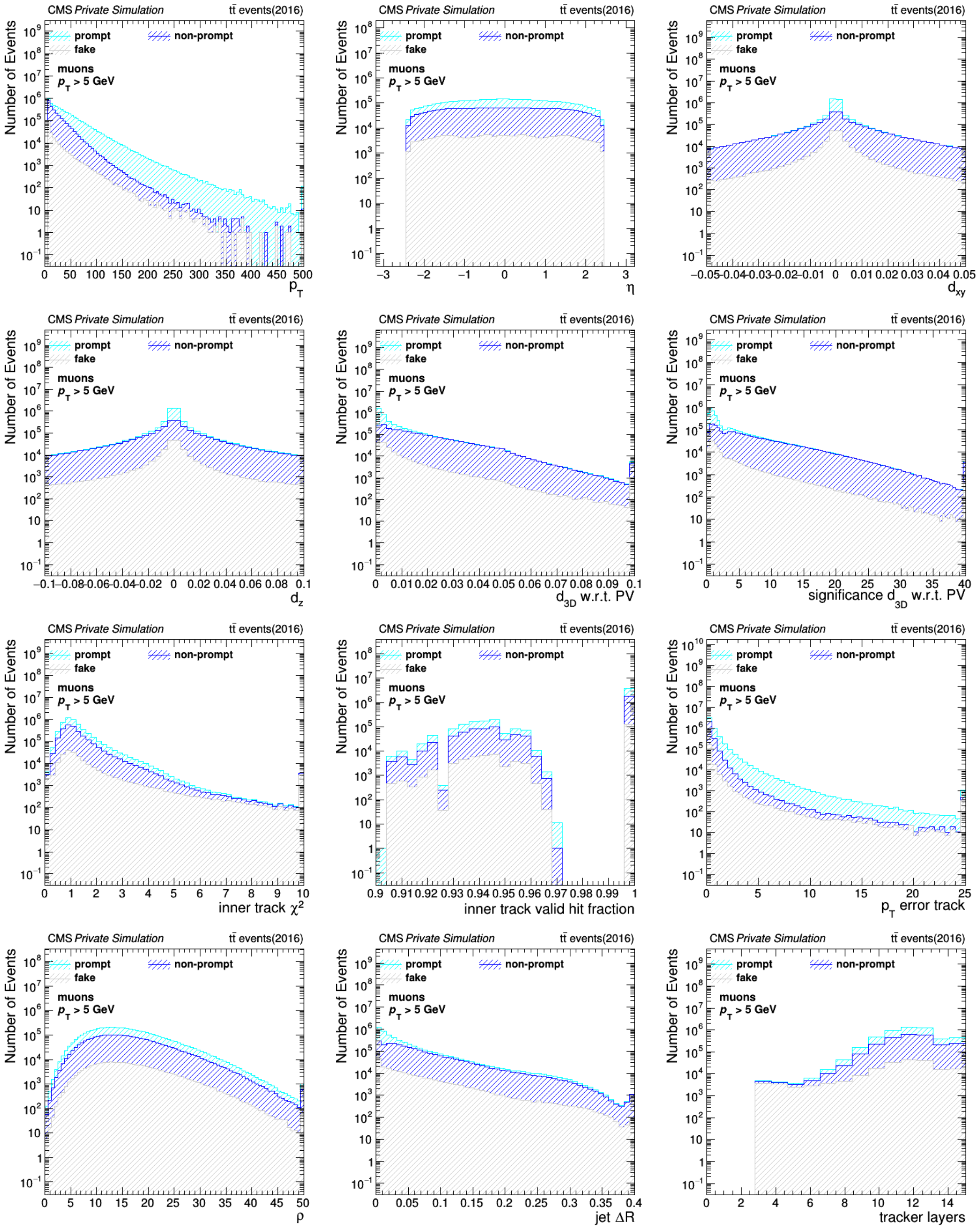
WIMP weakly interacting and massive particle

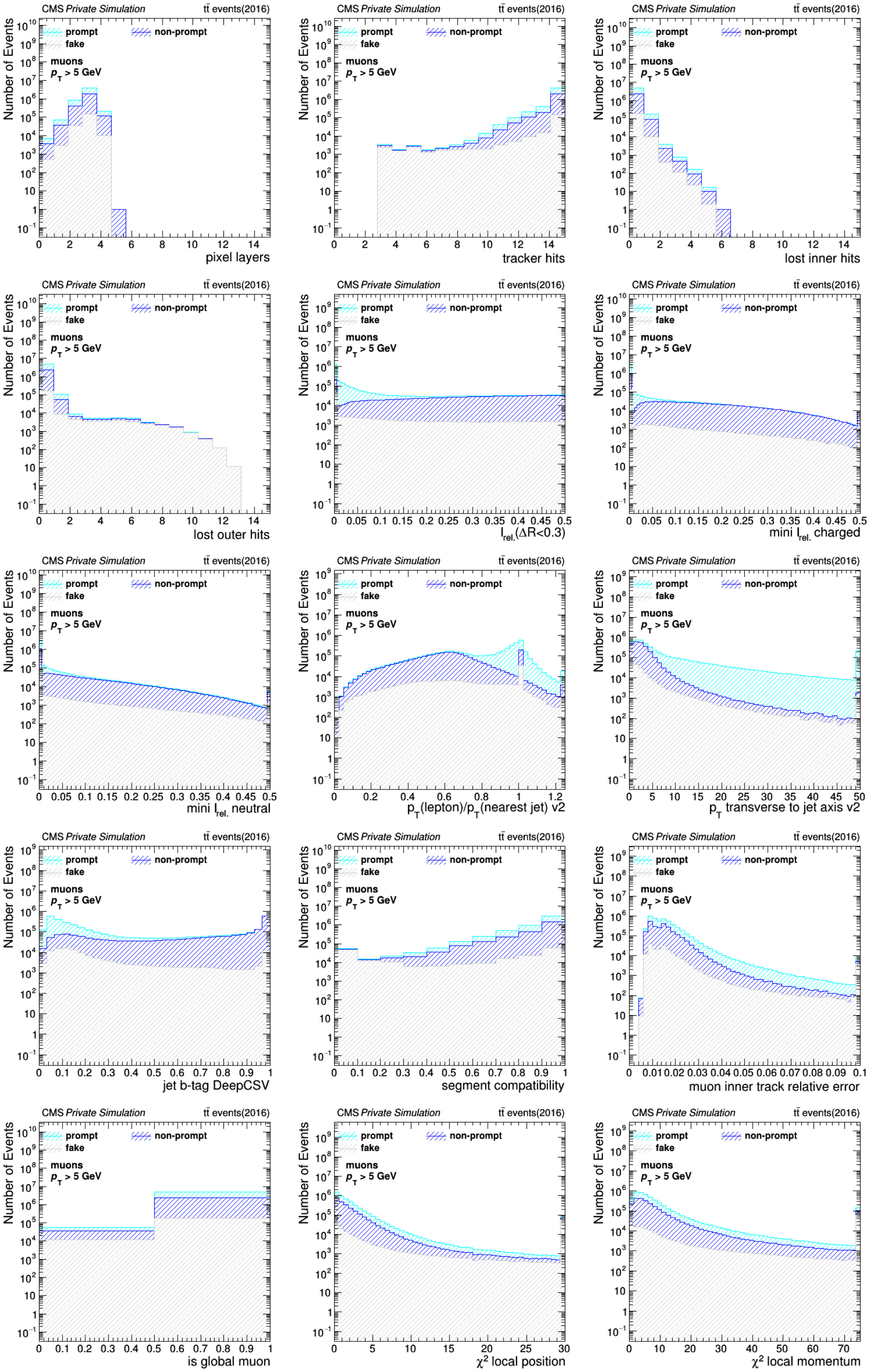
Appendices

Appendix A

Input training data

A.1 Muon features




CMS Private Simulation
tt̄ events(2016)
prompt
non-prompt
fake
muons
p_T > 5 GeV
Number of Events
pixel layers
tracker hits
lost inner hits
lost outer hits
$I_{rel.}(\Delta R<0.3)$
mini $I_{rel.}$ charged
mini $I_{rel.}$ neutral
p_T(lepton)/p_T(nearest jet) v2
p_T transverse to jet axis v2
jet b-tag DeepCSV
segment compatibility
muon inner track relative error
is global muon
χ^2 local position
χ^2 local momentum

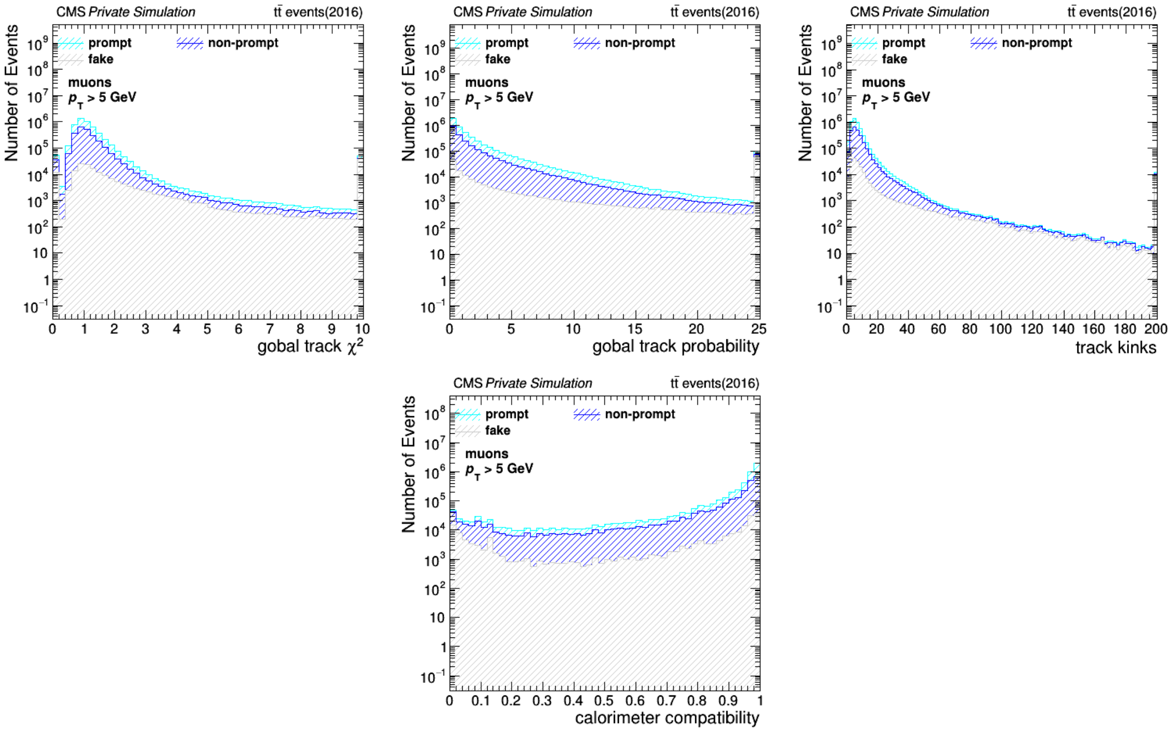


Fig. A.3: The muon features, which are selected for the training of the *DeepLepton* classifier. Muons from various $\mathrm{t\bar{t}}$ MC samples, with a $p_{\mathrm{T}} > 5\,\mathrm{GeV}$ and a $I_{\mathrm{rel.}}(\Delta R \leq 0.3) \leq 0.5$, are considered for the training data.

A.2 Vicinity features

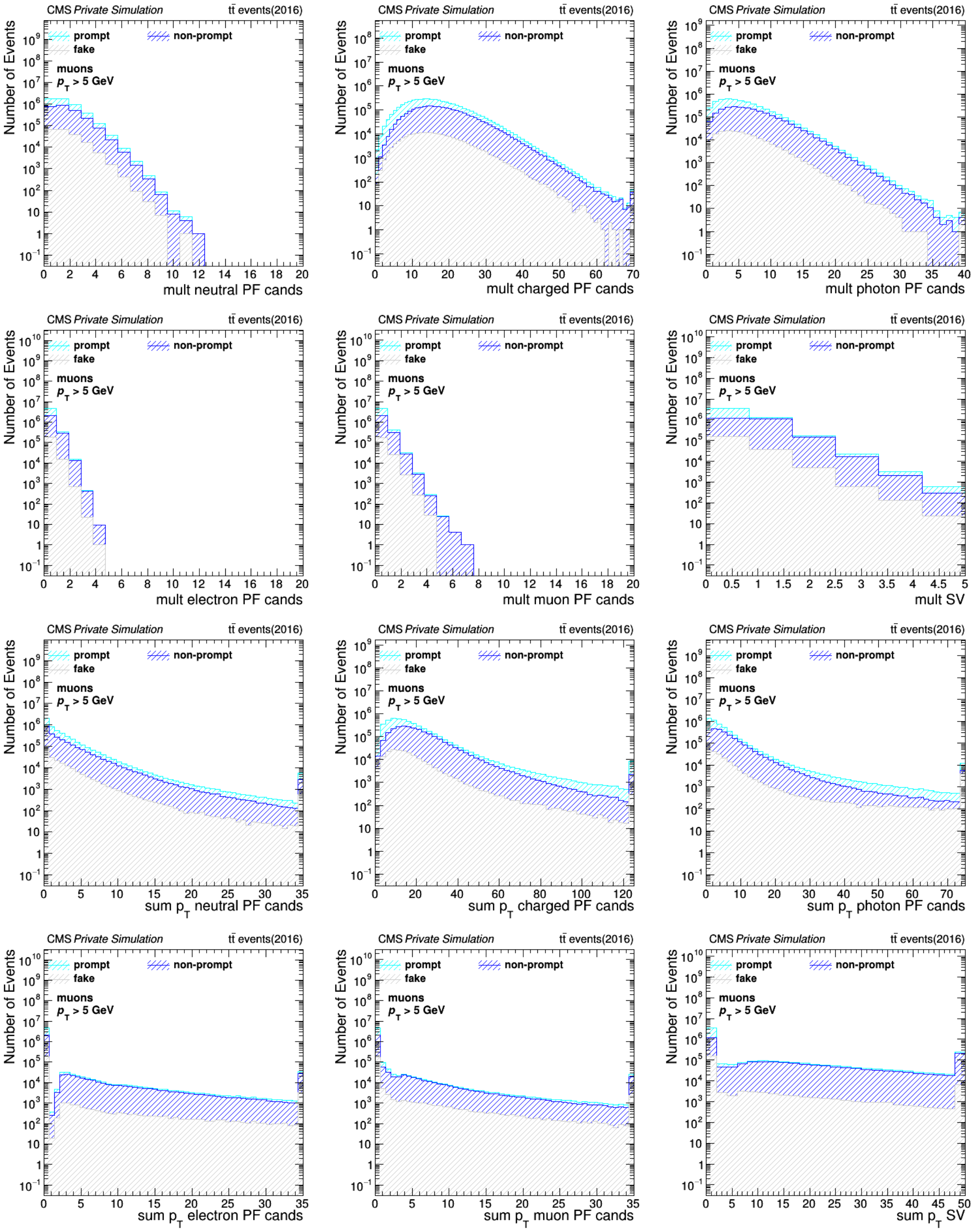
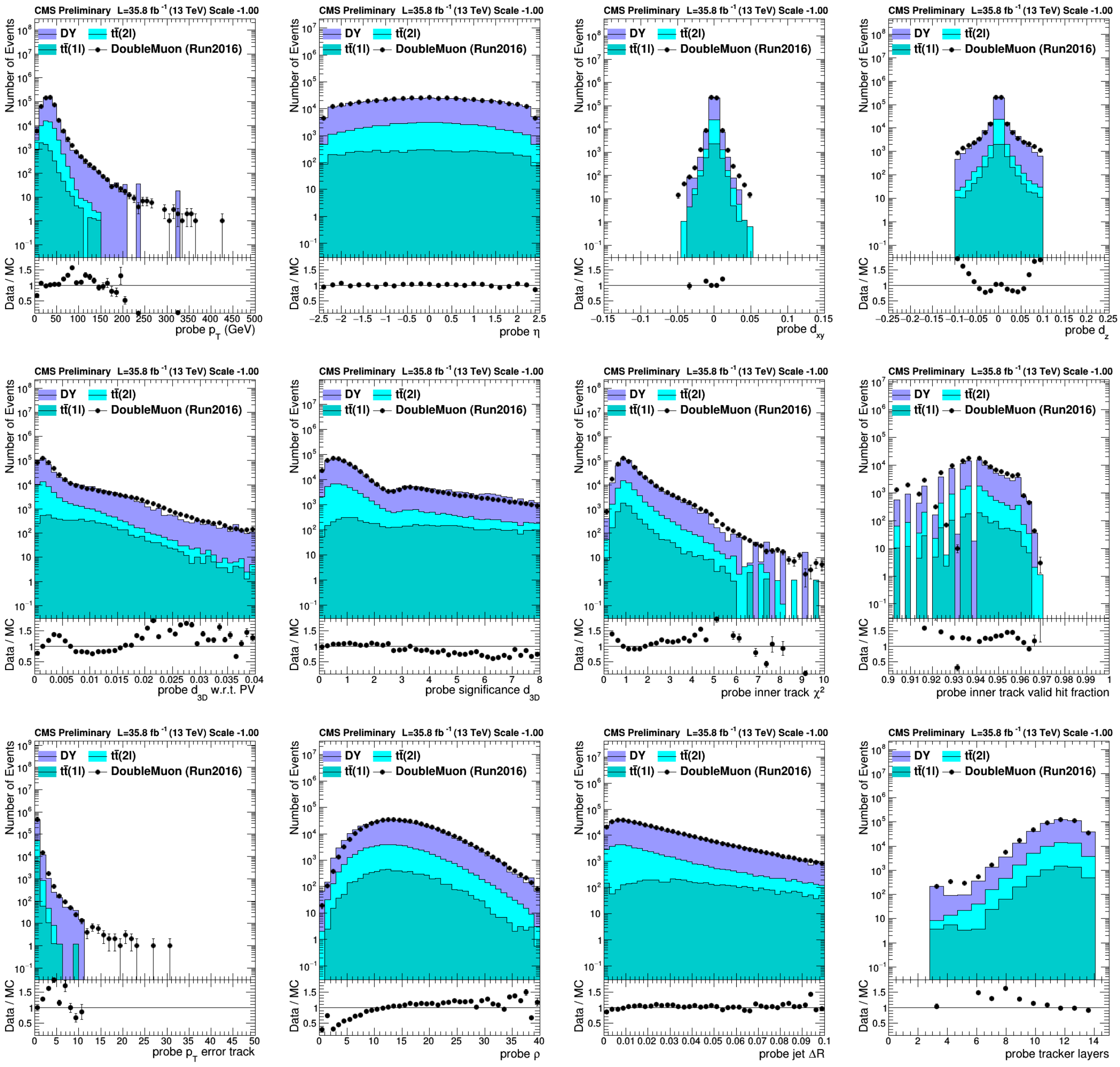


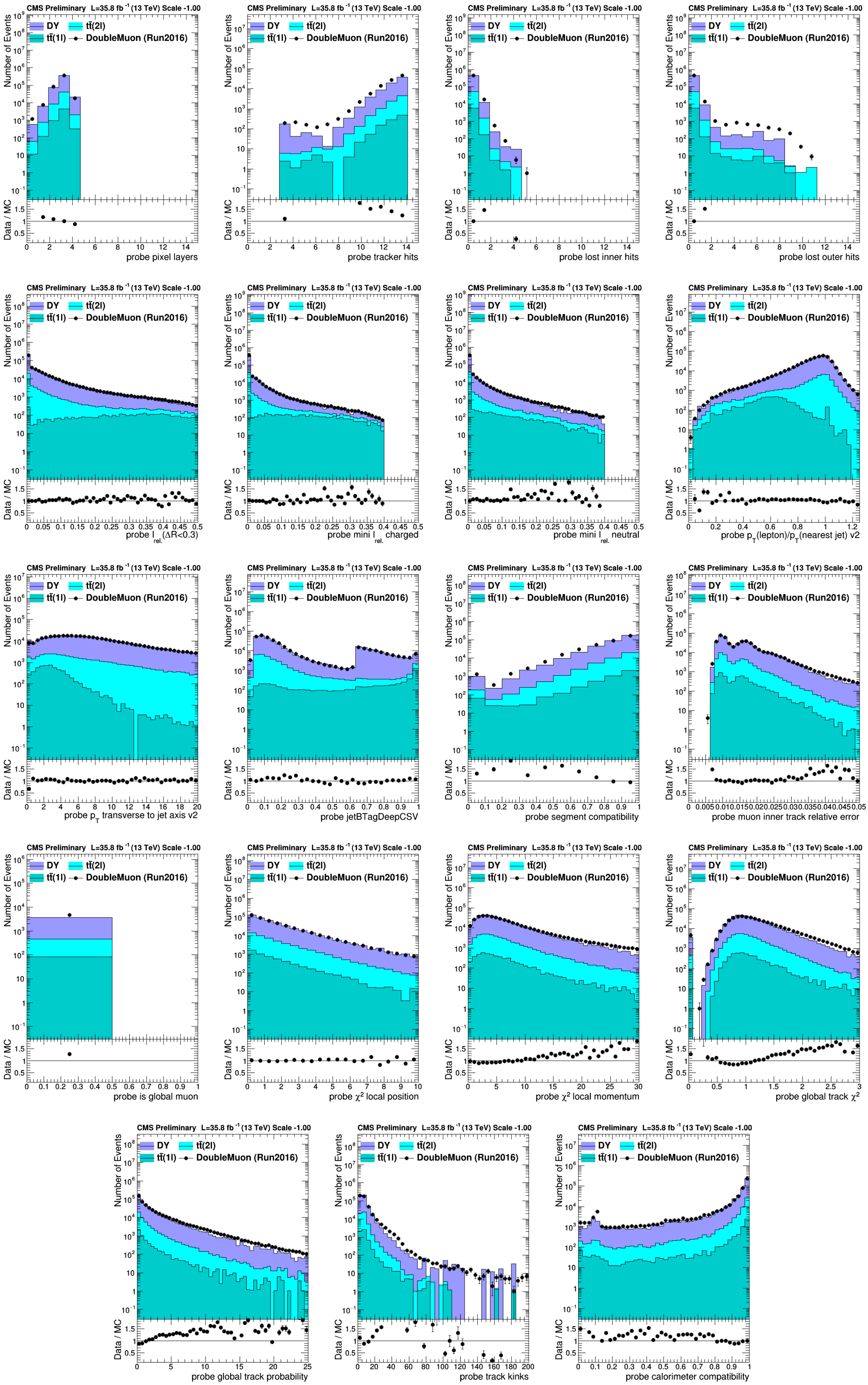
Fig. A.4: Multiplicity and $\sum p_{\mathrm{T}}$ of the vicinity feature species. The PF candidates are obtained within a cone of $\Delta R \leq 0.5$ around the associated muon. In terms of the SVs a $\Delta R \leq 0.6$ is selected.

Appendix B

Data-MC comparisons in the DY selection

B.1 Training features - probe muons




CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe pixel layers
probe tracker hits
probe lost inner hits
probe lost outer hits
probe $I_{rel.}$ (ΔR<0.3)
probe mini $I_{rel.}$ charged
probe mini $I_{rel.}$ neutral
probe p_T(lepton)/p_T(nearest jet) v2
probe p_T transverse to jet axis v2
probe jetBTagDeepCSV
probe segment compatibility
probe muon inner track relative error
probe is global muon
probe χ^2 local position
probe χ^2 local momentum
probe global track χ^2
probe global track probability
probe track kinks
probe calorimeter compatibility

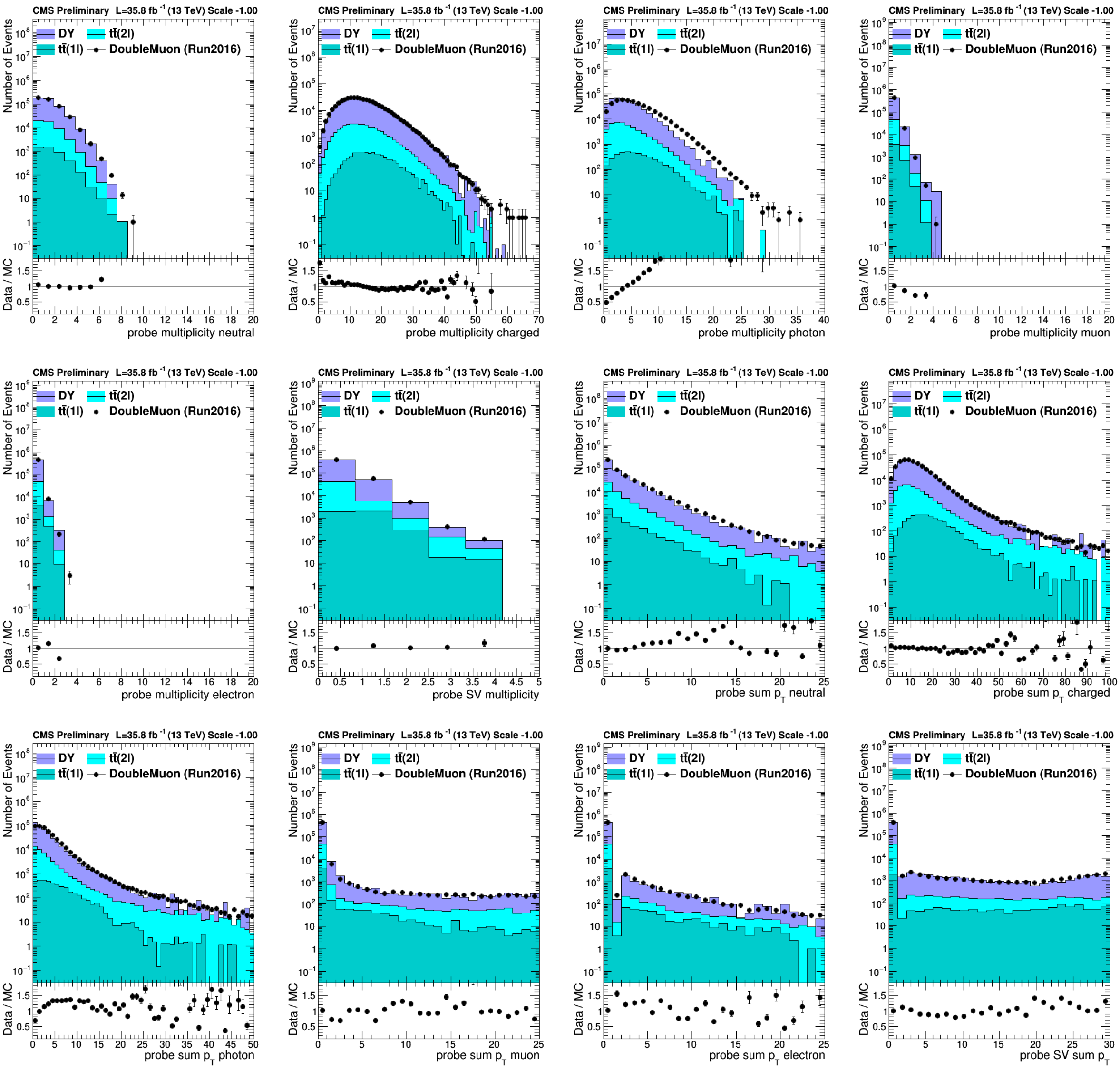
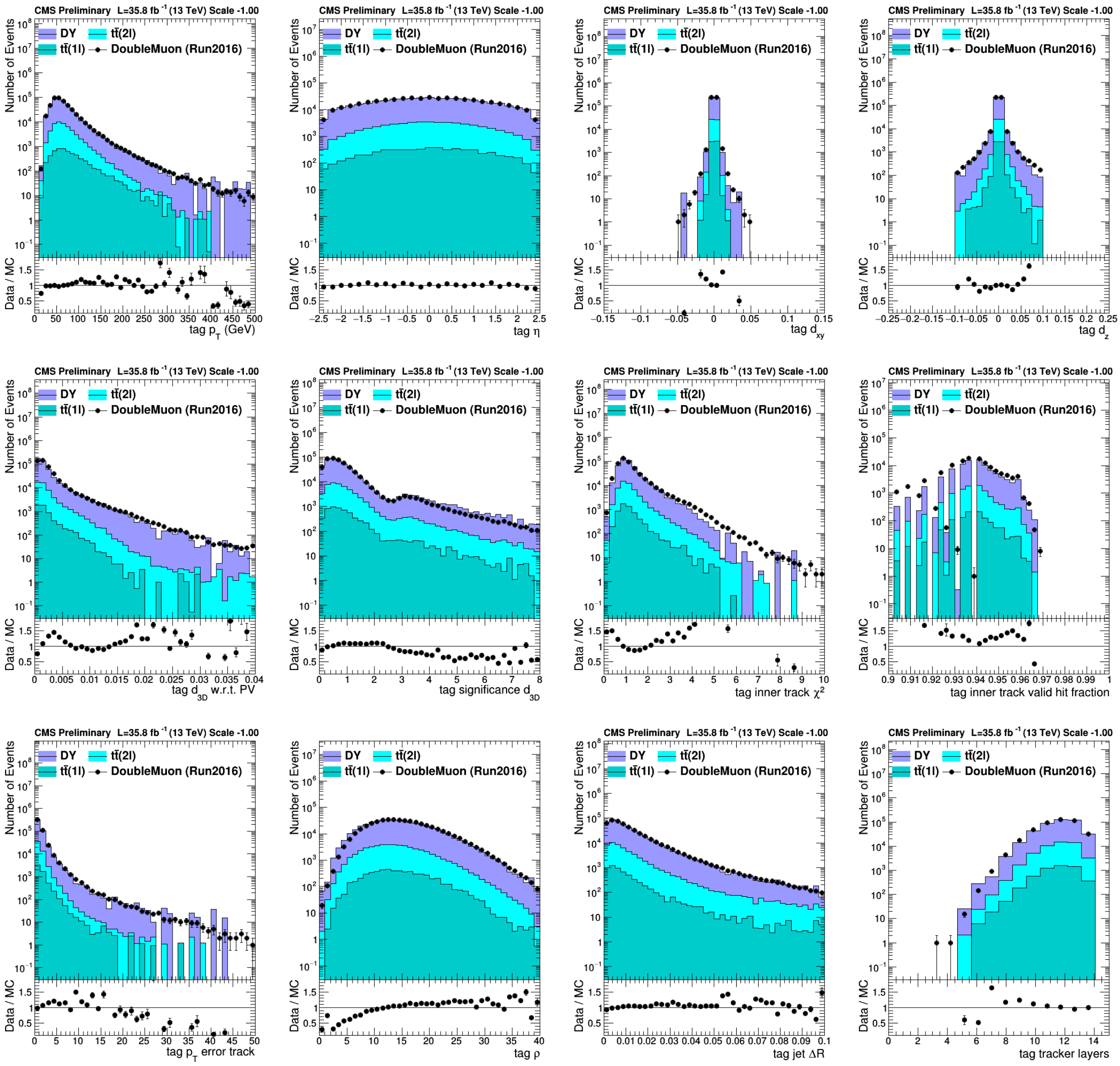
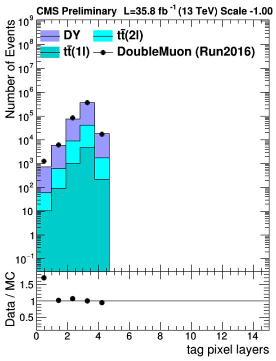
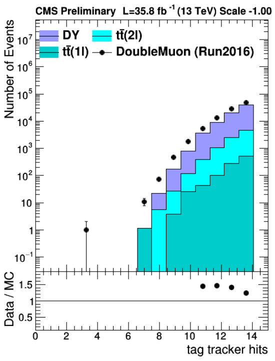


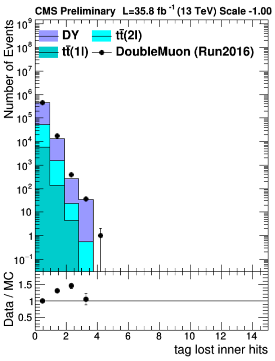
Fig. B.3: Data-MC comparisons of input training features of probe muons in the DY selection.

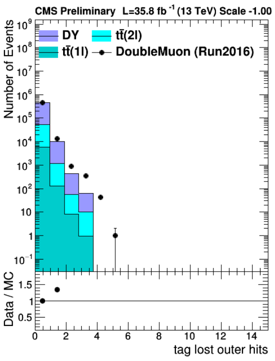
B.2 Training features - tag muons

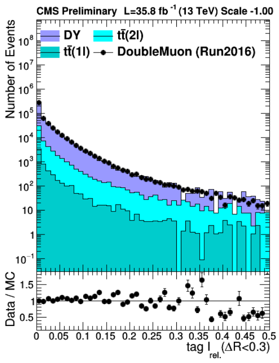


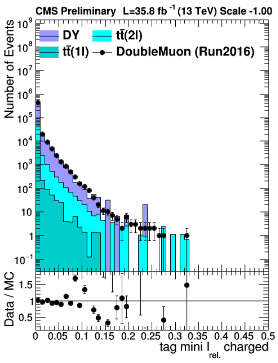

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag pixel layers

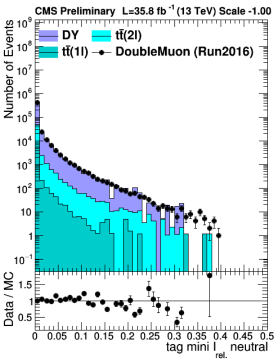

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag tracker hits

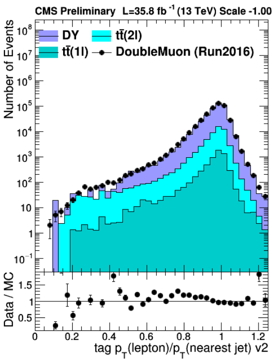

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag lost inner hits

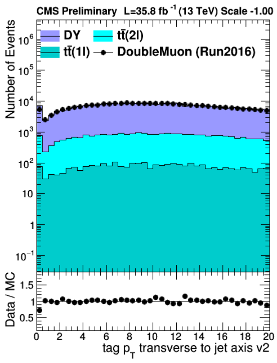

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag lost outer hits

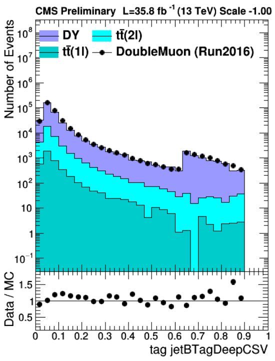

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag $I_{rel.}$ (ΔR<0.3)

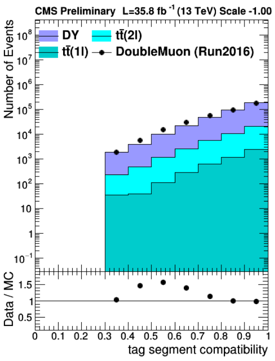

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag mini $I_{rel.}$ charged

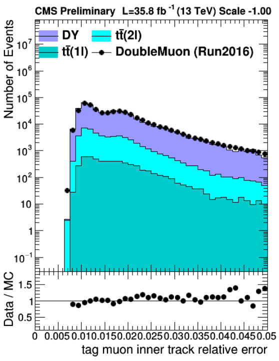

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag mini $I_{rel.}$ neutral

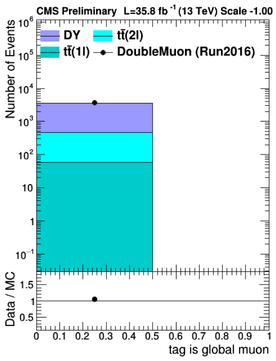

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag p_T(lepton)/p_T(nearest jet) v2

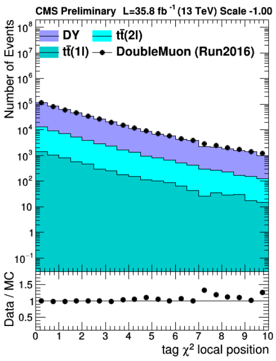

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag p_T transverse to jet axis v2

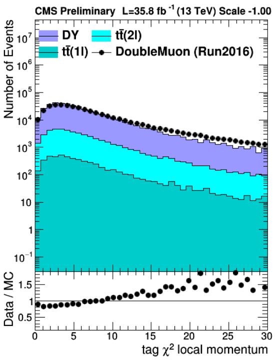

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag jetBTagDeepCSV

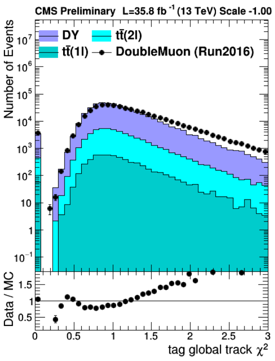

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag segment compatibility

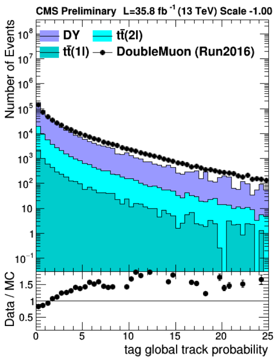

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag muon inner track relative error

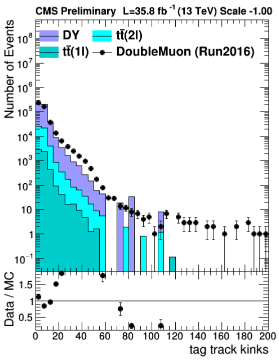

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag is global muon

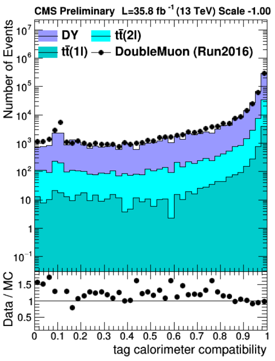

CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag χ^2 local position


CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag χ^2 local momentum


CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag global track χ^2


CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag global track probability


CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag track kinks


CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
DY
$t\bar{t}$(2l)
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag calorimeter compatibility

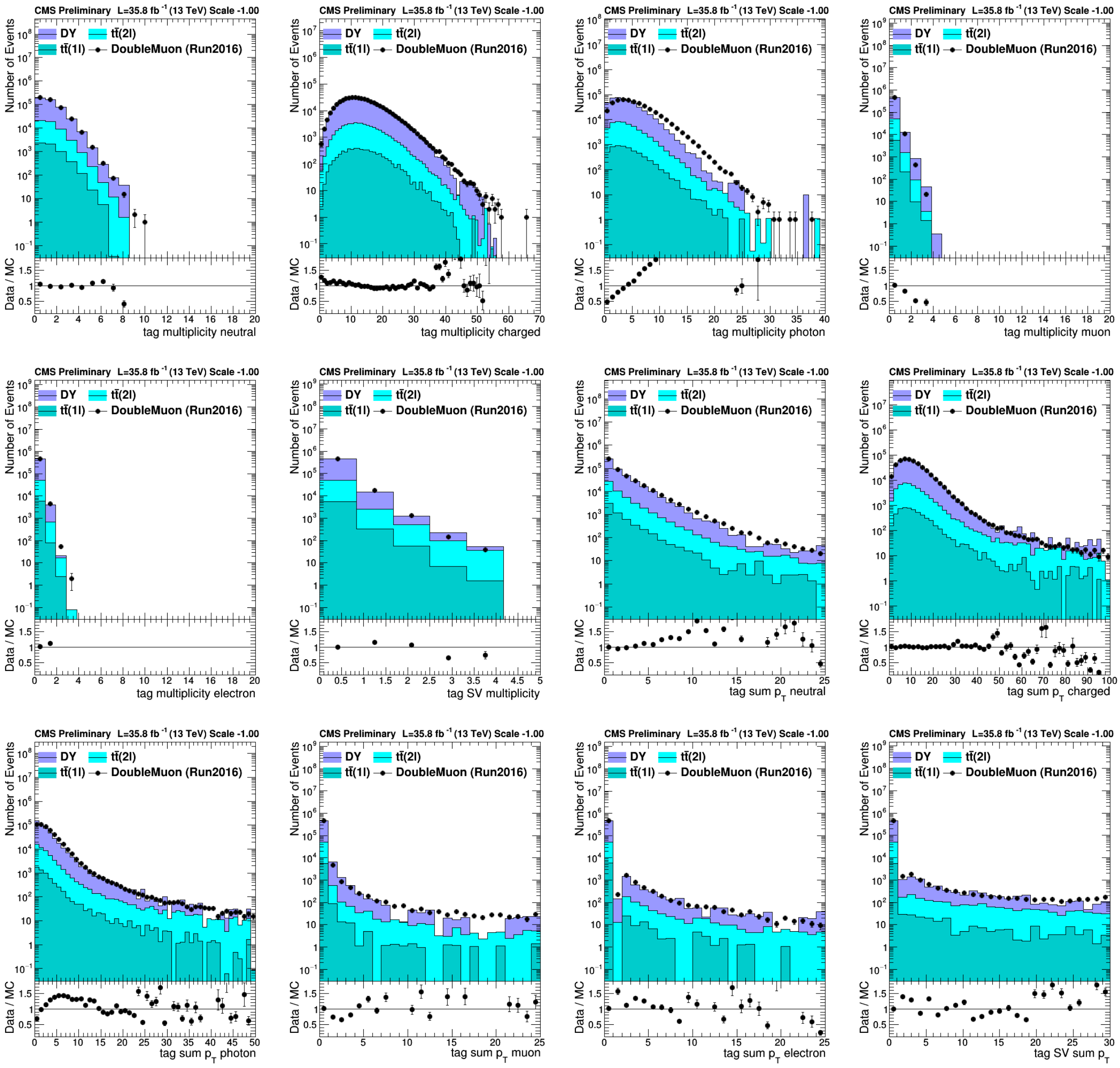


Fig. B.6: Data-MC comparisons of input training features of tag muons in the DY selection.

B.3 Global event features

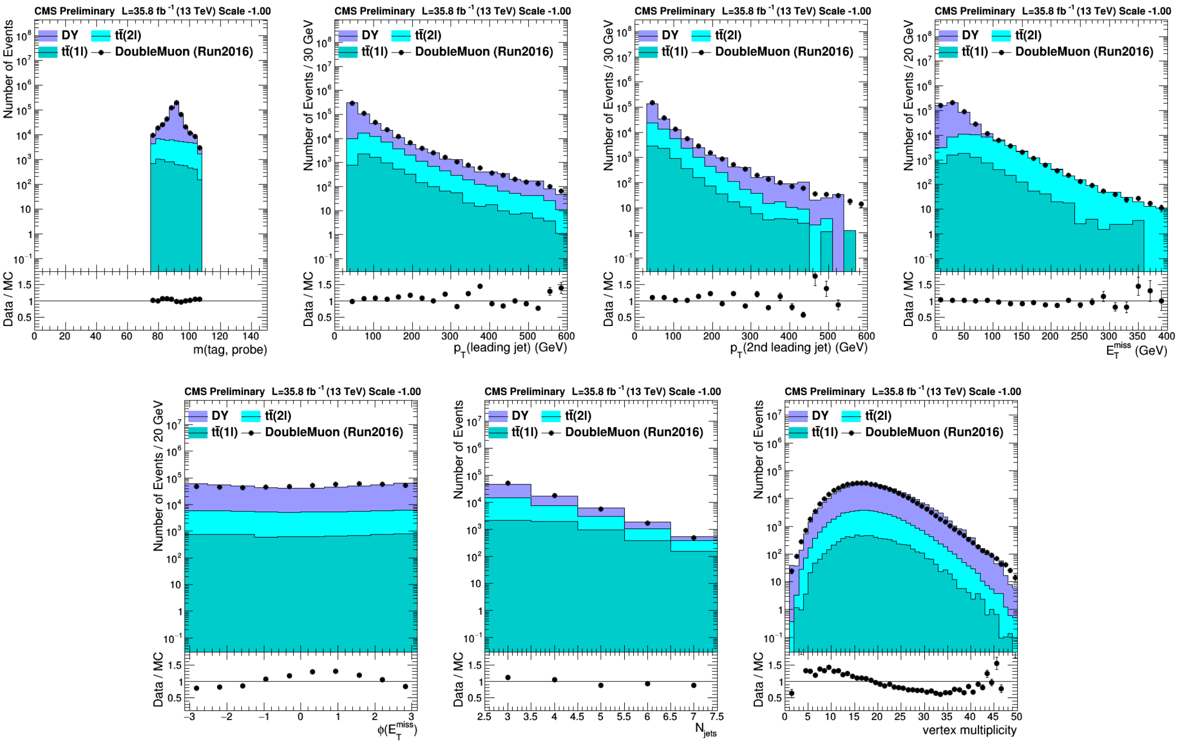


Fig. B.7: Data-MC comparisons of global event features in the DY selection.

B.4 Discriminator shapes

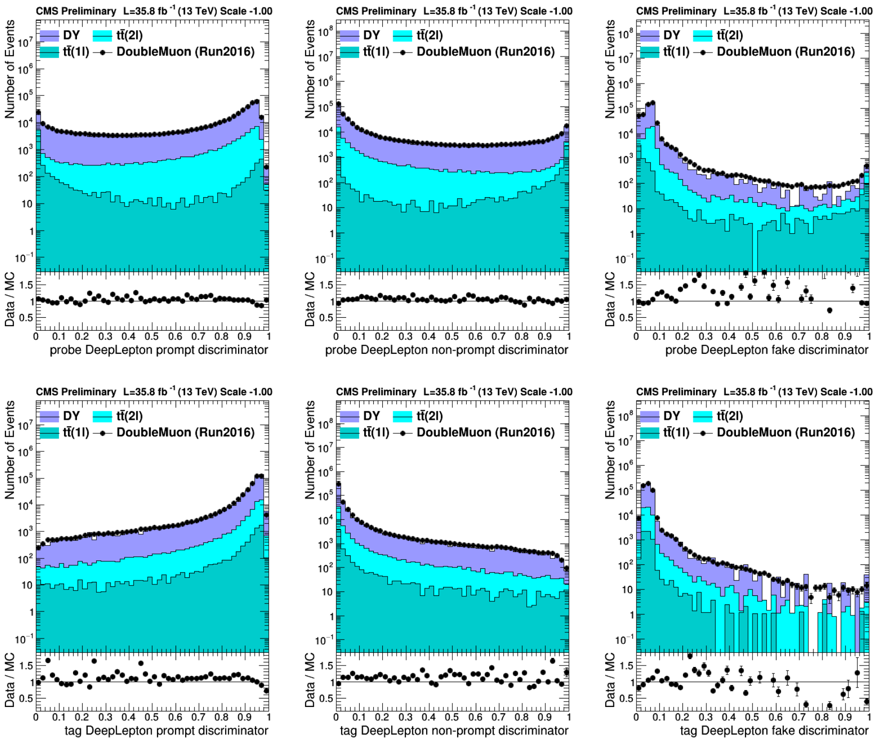
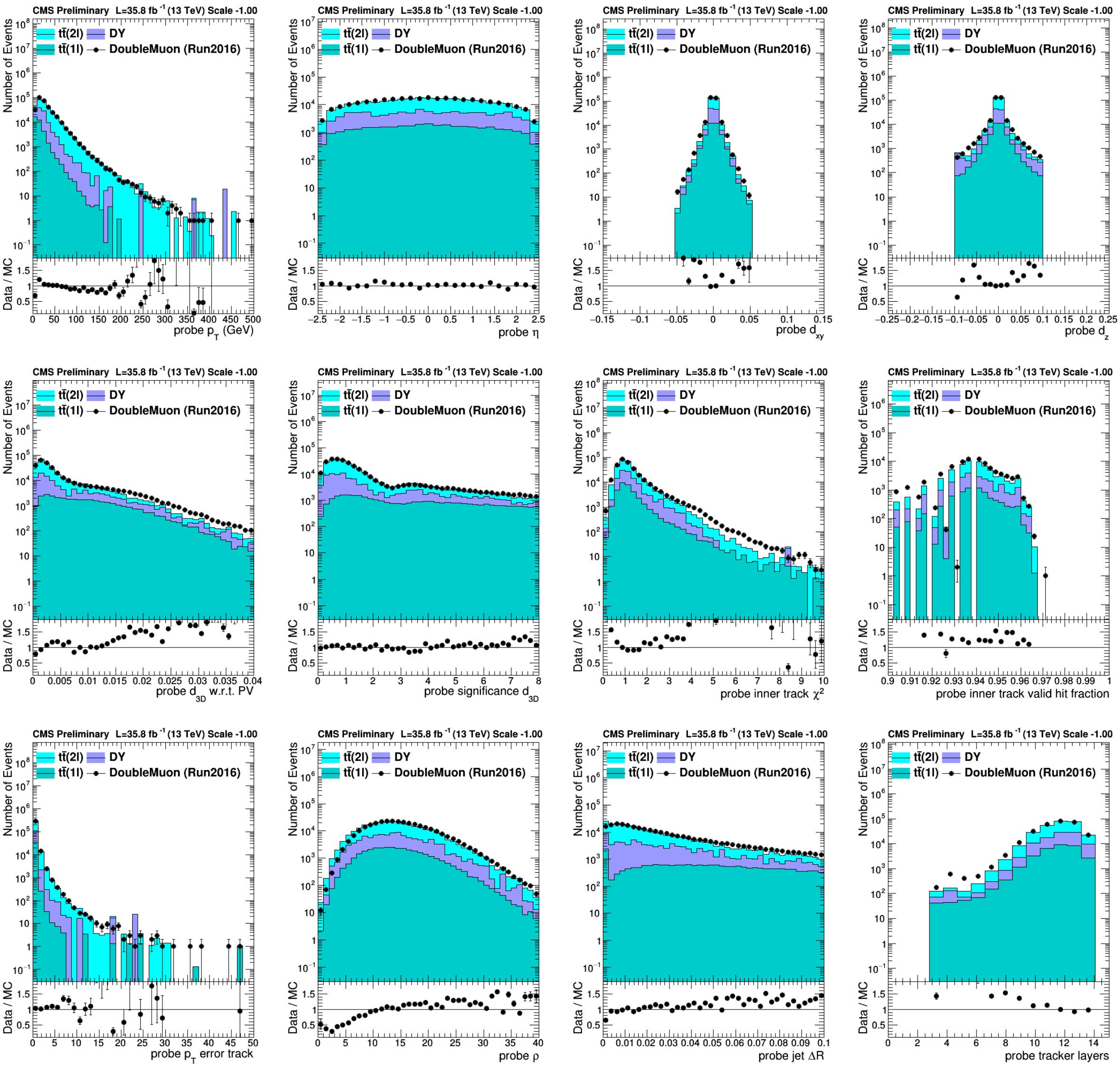


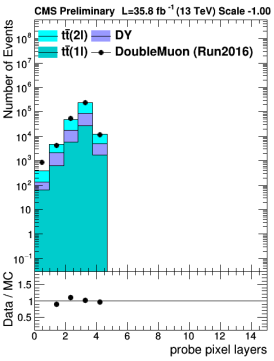
Fig. B.8: Data-MC comparisons of the discriminator shapes in the DY selection.

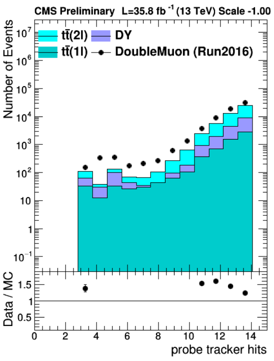
Appendix C

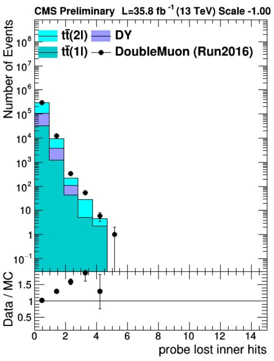
Data-MC comparisons in the $t\bar{t}$ selection

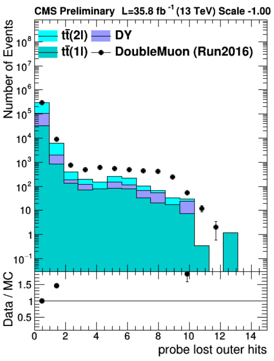
C.1 Training features - probe muons

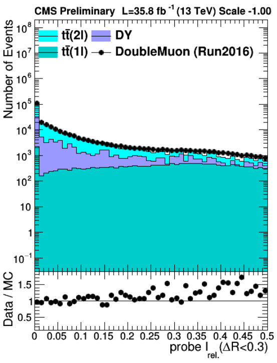


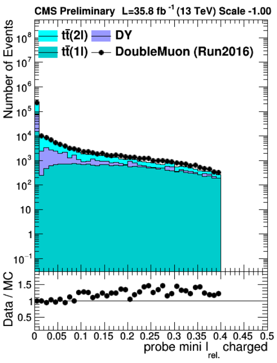

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe pixel layers

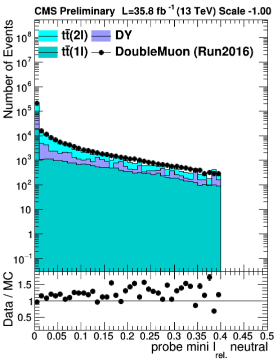

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe tracker hits

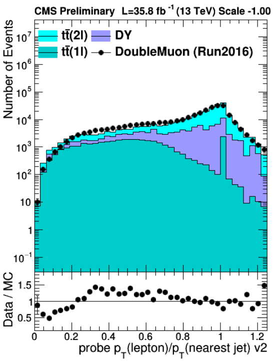

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe lost inner hits

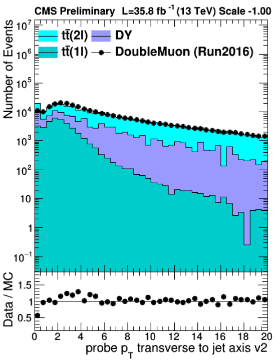

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe lost outer hits

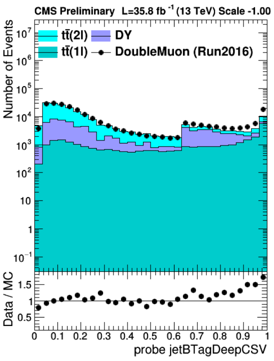

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe I_rel. (ΔR<0.3)

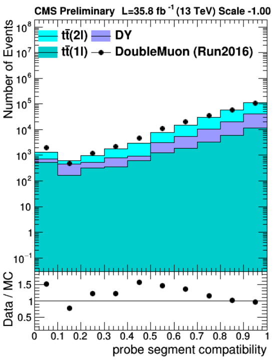

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe mini I_rel. charged

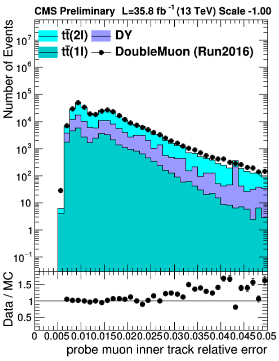

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe mini I_rel. neutral

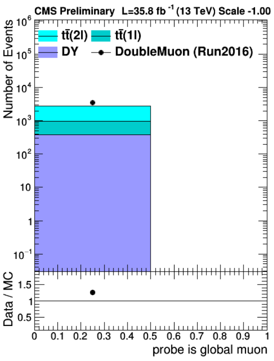

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe p_T(lepton)/p_T(nearest jet) v2

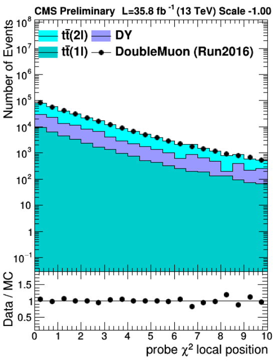

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe p_T transverse to jet axis v2

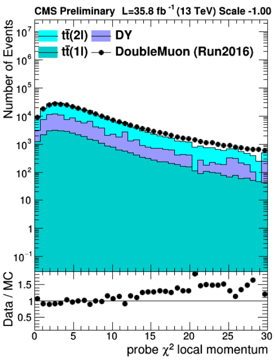

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe jetBTagDeepCSV

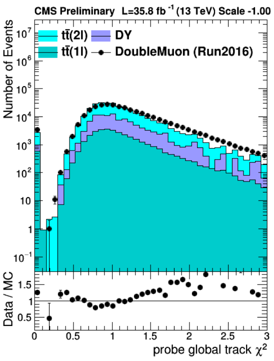

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe segment compatibility

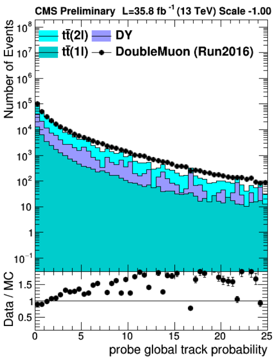

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe muon inner track relative error

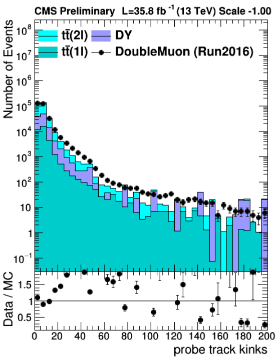

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
tt̄(1l)
DY
DoubleMuon (Run2016)
Number of Events
Data / MC
probe is global muon

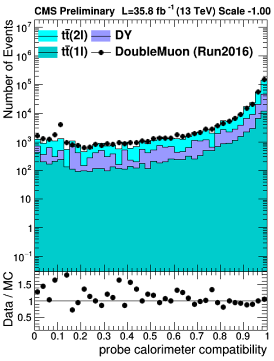

CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe χ² local position


CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe χ² local momentum


CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe global track χ²


CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe global track probability


CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe track kinks


CMS Preliminary L=35.8 fb⁻¹ (13 TeV) Scale -1.00
tt̄(2l)
DY
tt̄(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
probe calorimeter compatibility

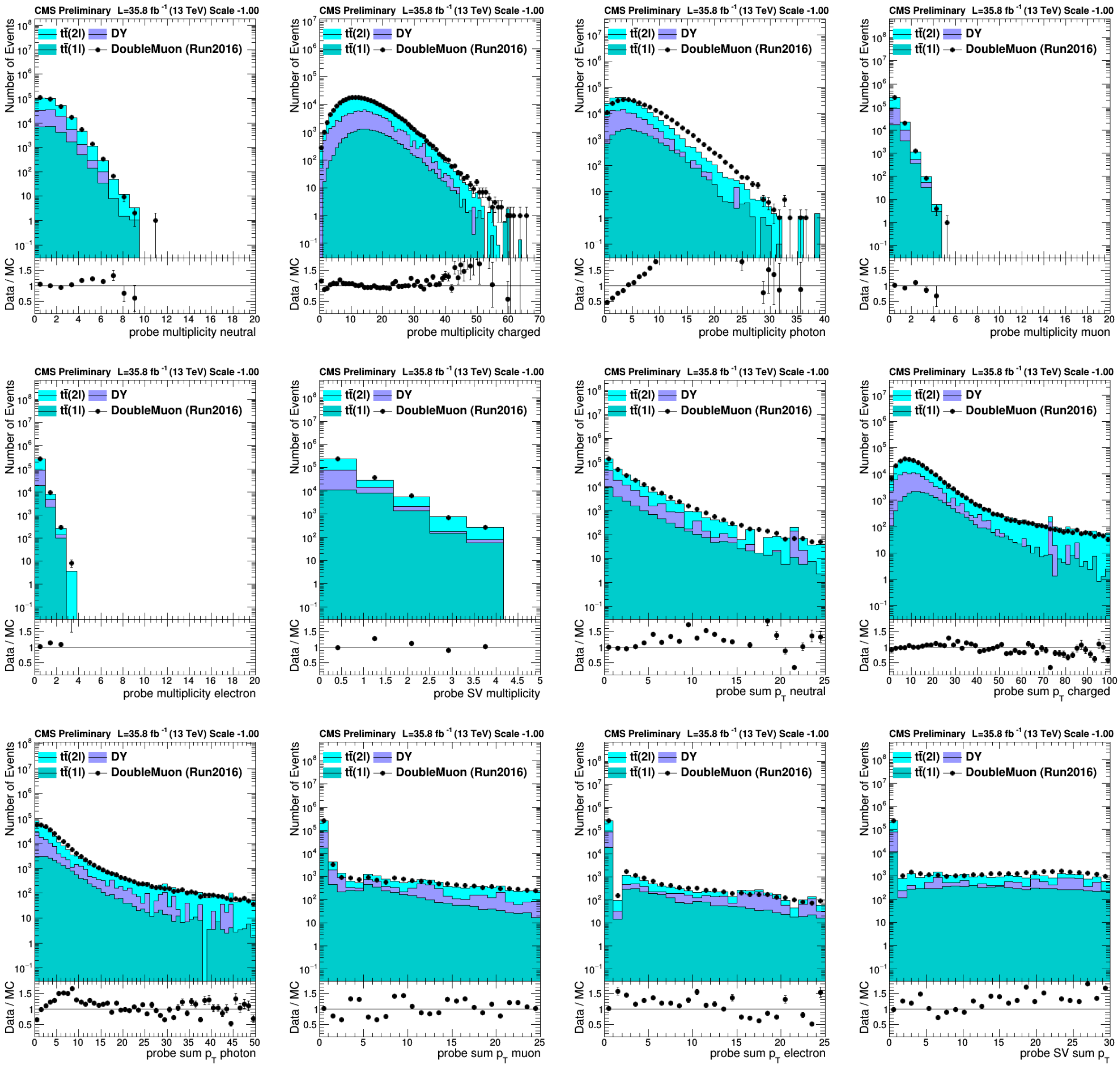
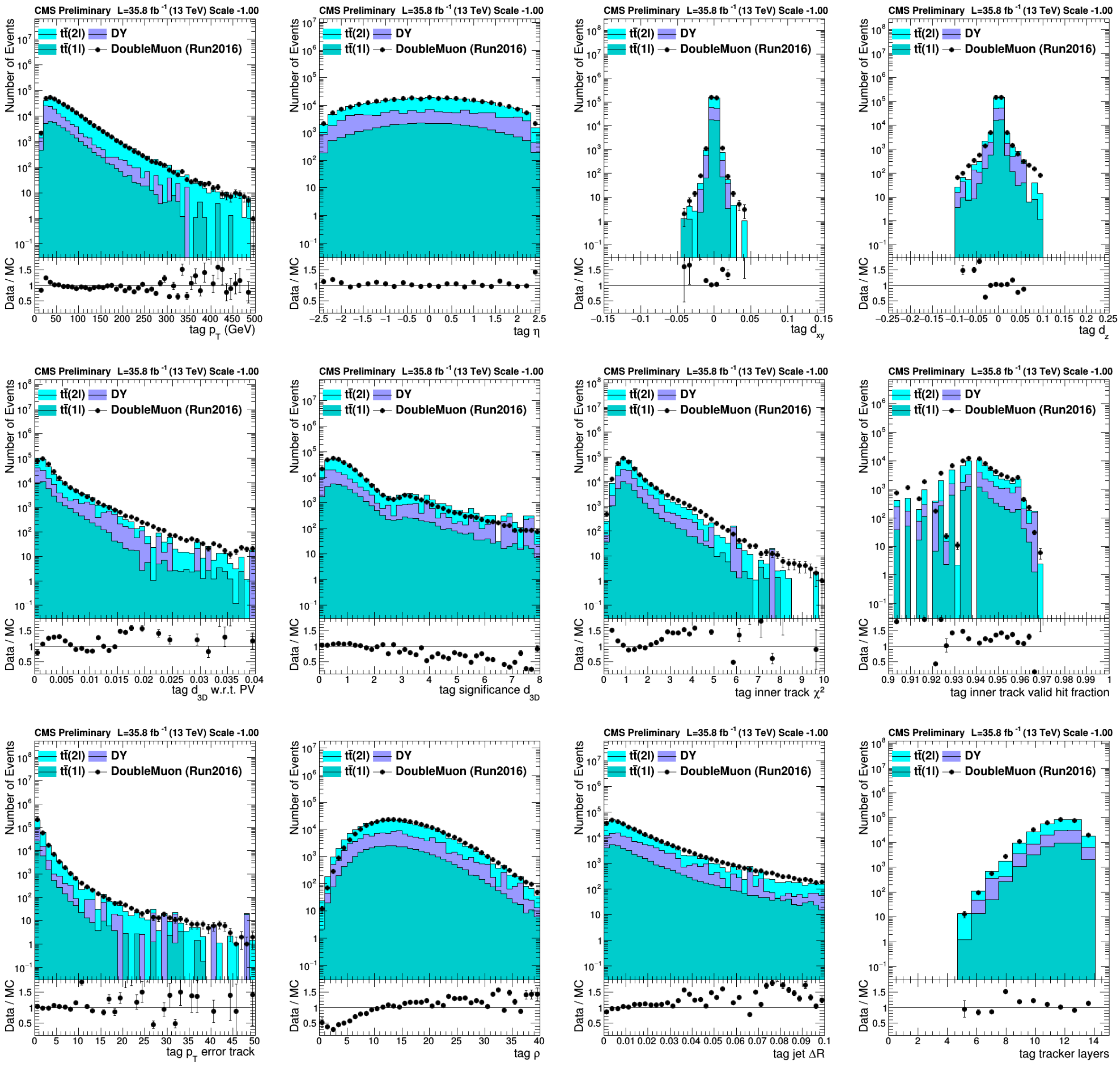
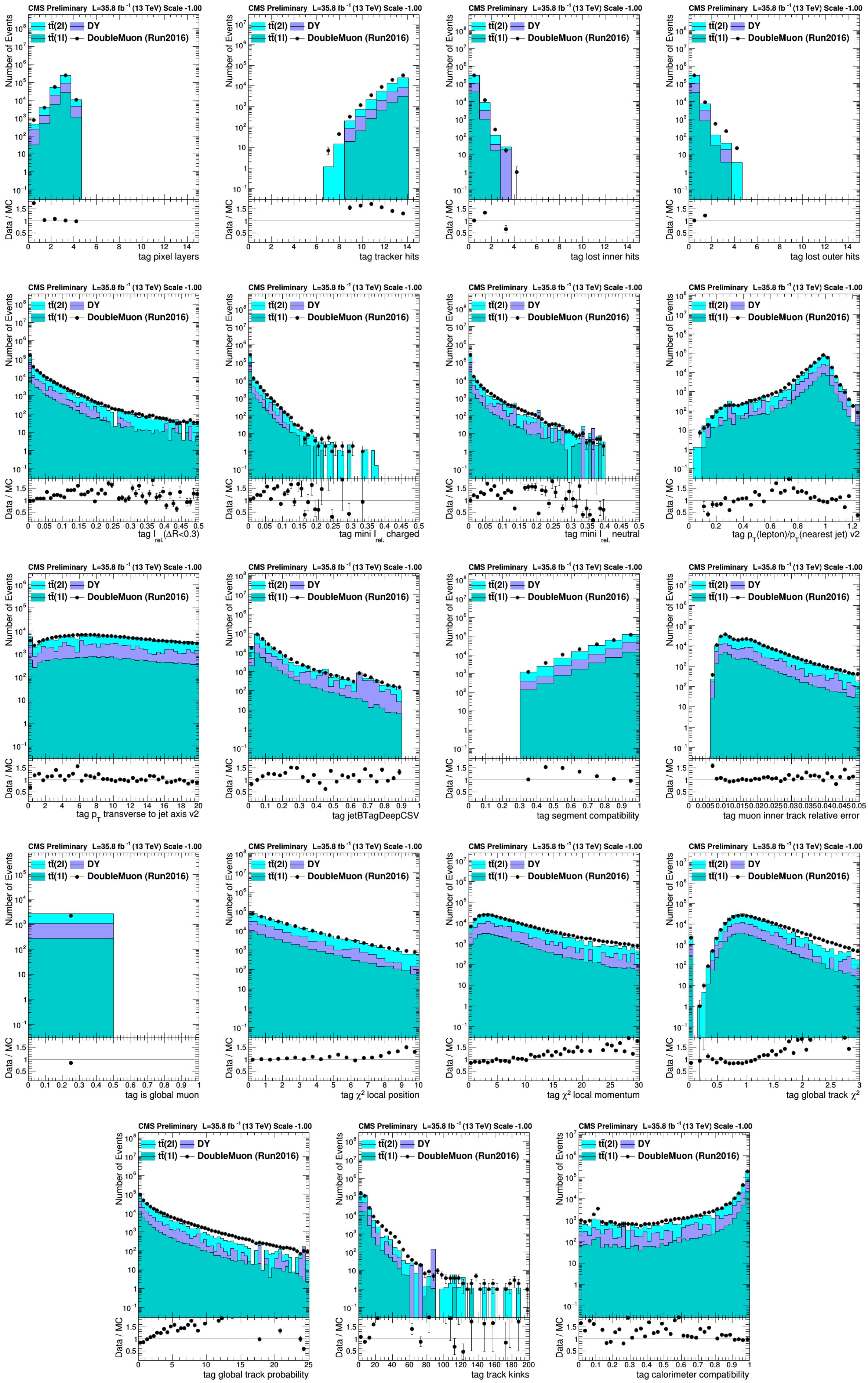


Fig. C.3: Data-MC comparisons of input training features of probe muons in the $t\bar{t}$ selection.

C.2 Training features - tag muons




CMS Preliminary L=35.8 fb^{-1} (13 TeV) Scale -1.00
$t\bar{t}$(2l)
DY
$t\bar{t}$(1l)
DoubleMuon (Run2016)
Number of Events
Data / MC
tag pixel layers
tag tracker hits
tag lost inner hits
tag lost outer hits
tag I$_{rel.}$ (ΔR<0.3)
tag mini I$_{rel.}$ charged
tag mini I$_{rel.}$ neutral
tag p$_T$(lepton)/p$_T$(nearest jet) v2
tag p$_T$ transverse to jet axis v2
tag jetBTagDeepCSV
tag segment compatibility
tag muon inner track relative error
tag is global muon
tag χ^2 local position
tag χ^2 local momentum
tag global track χ^2
tag global track probability
tag track kinks
tag calorimeter compatibility

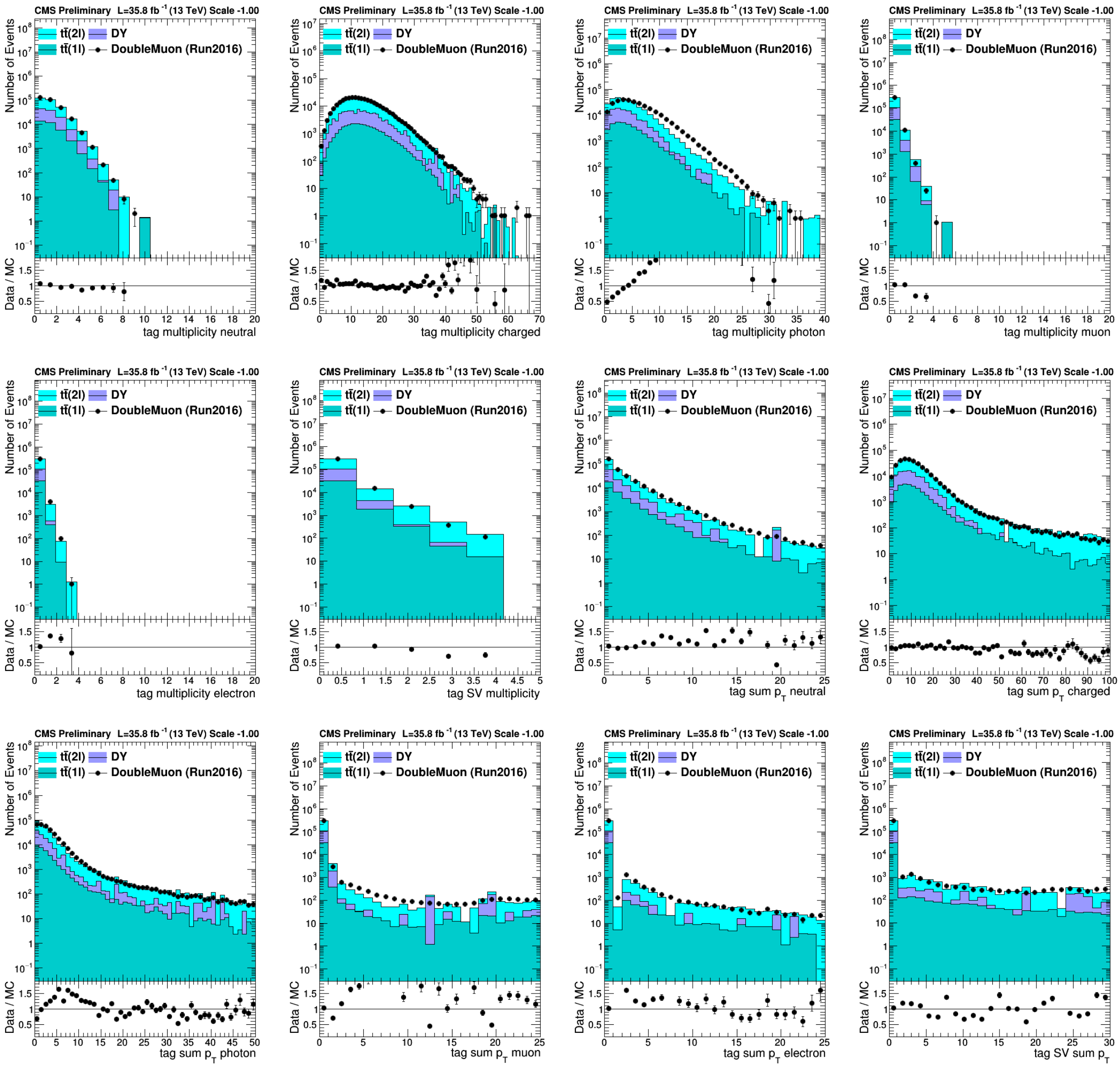


Fig. C.6: Data-MC comparisons of input training features of tag muons in the $t\bar{t}$ selection.

C.3 Global event features

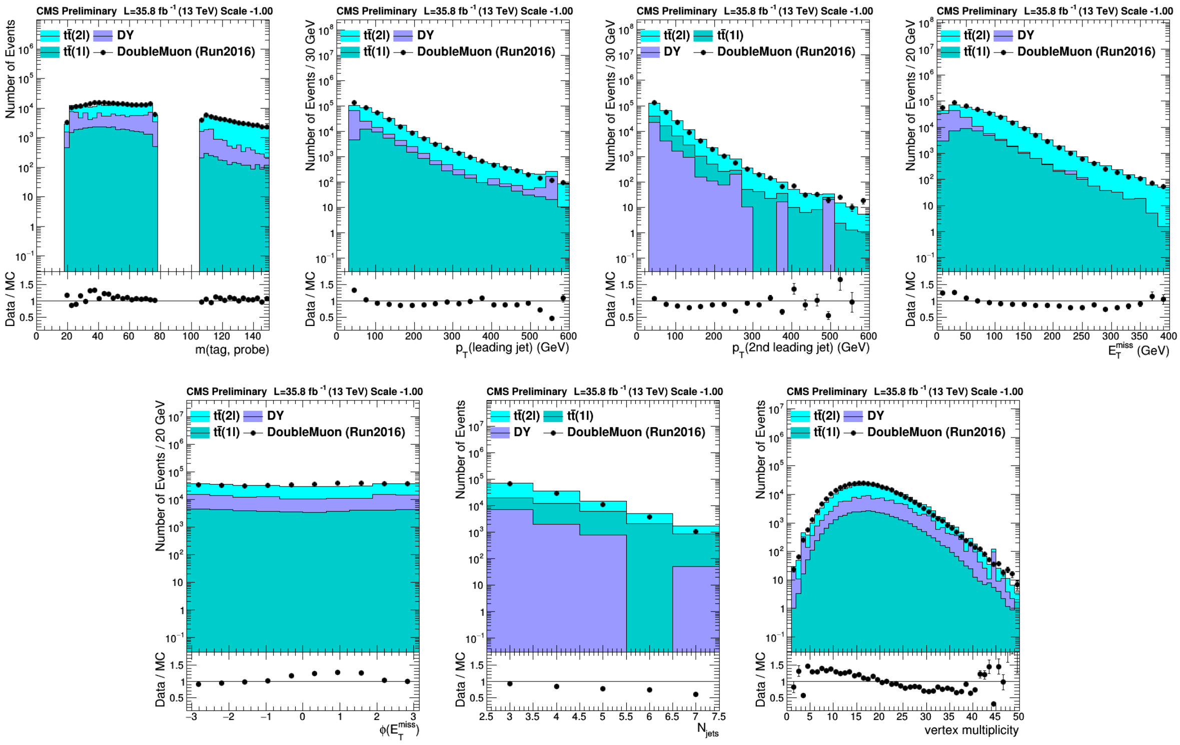


Fig. C.7: Data-MC comparisons of global event features in the $t\bar{t}$ selection.

C.4 Discriminator shapes

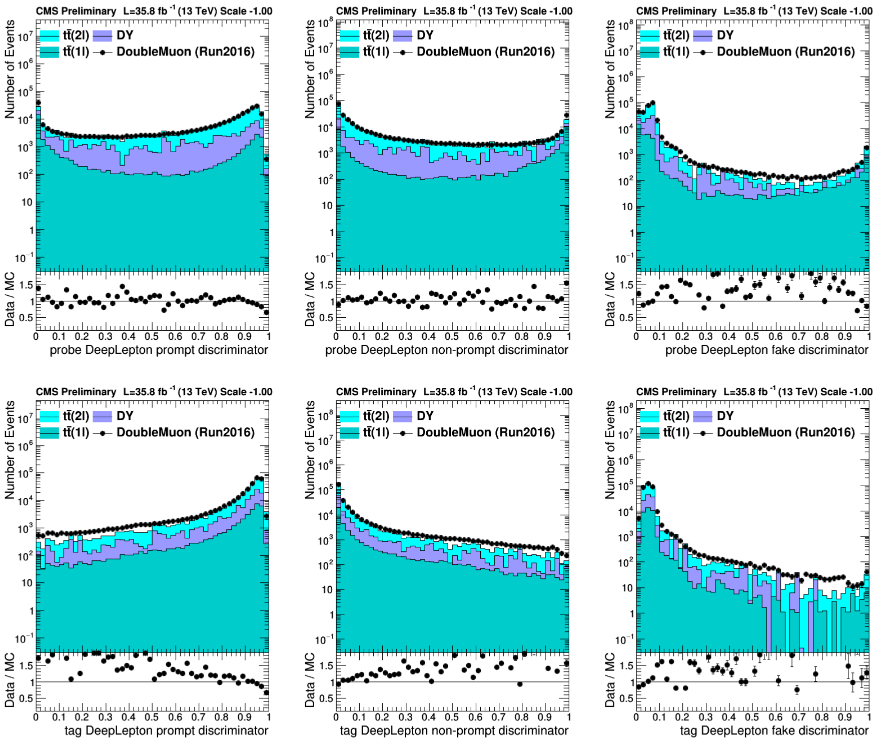


Fig. C.8: Data-MC comparisons of the discriminator shapes in the $t\bar{t}$ selection.

Appendix D

Matrix Comparison

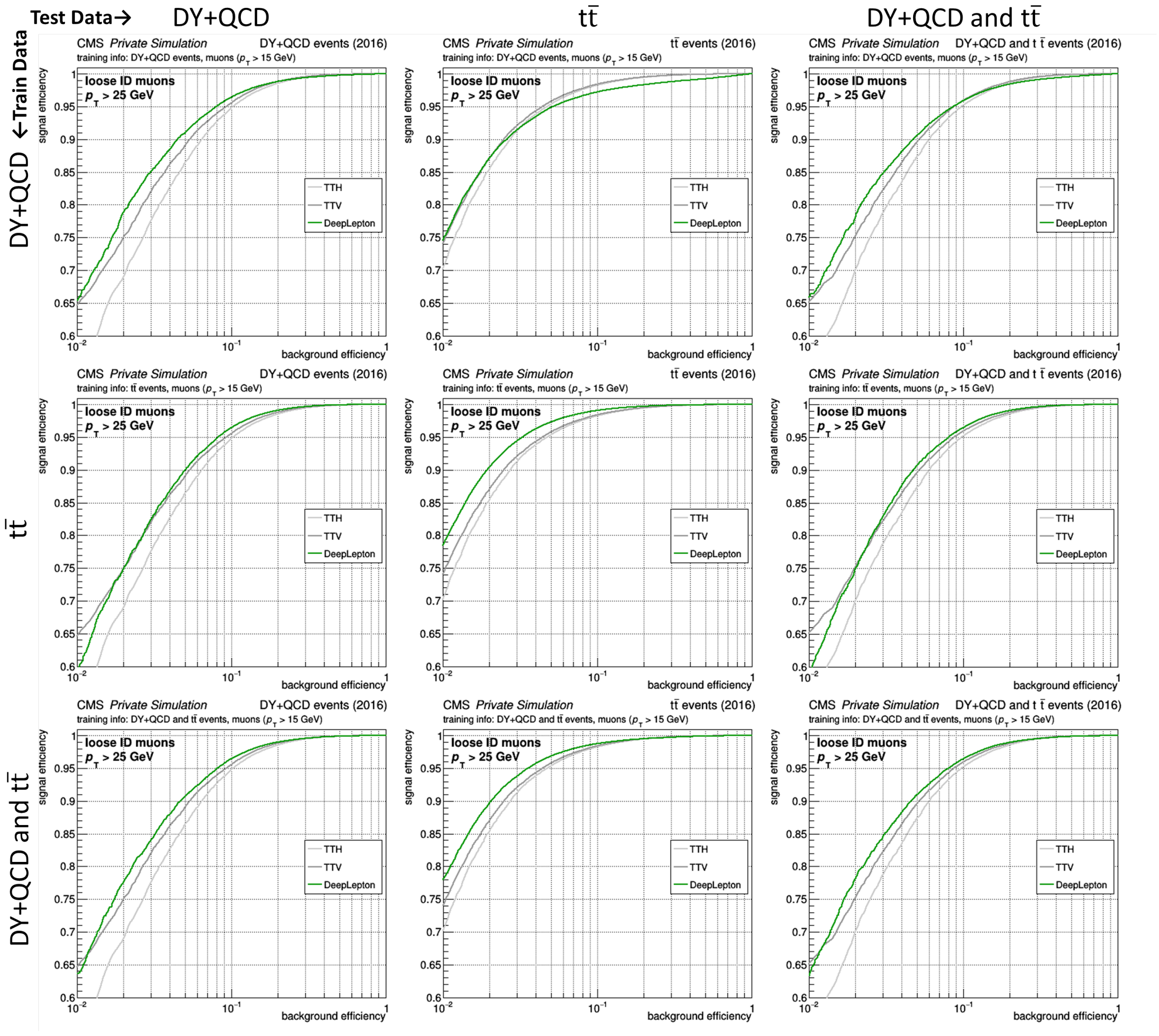


Fig. D.1: Matrix comparison for the high-p_T range. Training models, based on different training samples, are evaluated on different test data samples. The incompatibility between the DY+QCD and the $t\bar{t}$ samples becomes apparent in these comparisons.

References

[1] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. URL: https://cds.cern.ch/record/782076.

[2] The CMS Collaboration. "The CMS Experiment at the CERN LHC". In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[3] D. Spitzbart. *(YSF) Search for top squarks and dark matter particles in opposite-charge dilepton final states at CMS*. Tech. rep. CMS-CR-2018-134. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2632084.

[4] The CMS Collaboration. "Search for top squarks and dark matter particles in opposite-charge dilepton final states at $\sqrt{s} = 13$ TeV". In: *Phys. Rev. D* 97 (3 2018), p. 032009. DOI: 10.1103/PhysRevD.97.032009.

[5] R. Schoefbeck. *Search for supersymmetry with extremely compressed spectra with the ATLAS and CMS detectors*. Tech. rep. CMS-CR-2014-250. Geneva: CERN, 2014. URL: https://cds.cern.ch/record/1953439.

[6] M. Stoye, J. Kieseler, H. Qu, L. Gouskos, and M. Verzetti. *DeepJet: Generic physics object based jet multiclassclassification for LHC experiments*. 2018. URL: https://dl4physicalsciences.github.io/files/nips_dlps_2017_10.pdf (visited on 02/25/2019).

[7] The CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 30 –61. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235.

[8] A. Hoecker. *Physics at the LHC Run-2 and Beyond*. 2016. arXiv: 1611.07864.

[9] Y. Shadmi. "Introduction to Supersymmetry". In: *CERN Yellow Reports* 3.0 (2016), p. 95. DOI: 10.5170/CERN-2016-003.95. arXiv: 1708.00772.

[10] S. Raby. *Grand Unified Theories*. 2006. arXiv: hep-ph/0608183.

[11] K. Arun, S. B. Gudennavar, and C Sivaram. "Dark matter, dark energy, and alternate models: A review". In: *Advances in Space Research* 60.1 (2017), pp. 166 –186. ISSN: 0273-1177. DOI: 10.1016/j.asr.2017.03.043. arXiv: 1704.06155.

[12] L. Roszkowski, E. M. Sessolo, and S. Trojanowski. "WIMP dark matter candidates and searches - current status and future prospects". In: *Reports on Progress in Physics* 81.6 (2018), p. 066201. DOI: 10.1088/1361-6633/aab913. arXiv: 1707.06277.

[13] L. Baudis. "The Search for Dark Matter". In: *European Review* 26.1 (2018), 70–81. DOI: 10.1017/S1062798717000783. arXiv: 1801.08128.

[14] A. Maas. *Brout-Englert-Higgs physics: From foundations to phenomenology*. 2017. arXiv: 1712.04721.

[15] A. Tawfik. “Matter-Antimatter Asymmetry in the Large Hadron Collider”. In: *International Journal of Theoretical Physics* 51.5 (2012), pp. 1396–1407. ISSN: 1572-9575. DOI: 10.1007/s10773-011-1015-4. arXiv: 1011.6622.

[16] G. Cowan. *CP violation at the LHC.* Tech. rep. 2017. URL: https://cds.cern.ch/record/2277620.

[17] The ALICE Collaboration. “Midrapidity Antiproton-to-Proton Ratio in pp Collisons at $\sqrt{s} = 0.9$ and 7 TeV Measured by the ALICE Experiment”. In: *Phys. Rev. Lett.* 105 (7 2010), p. 072002. DOI: 10.1103/PhysRevLett.105.072002.

[18] A. Tawfik. “Antiproton-to-Proton Ratios for ALICE Heavy-Ion Collisions”. In: *Nuclear Physics A* 859.1 (2011), pp. 63 –72. ISSN: 0375-9474. DOI: 10.1016/j.nuclphysa.2011.04.014. arXiv: 1011.5612.

[19] R. Pasechnik and M. Šumbera. “Phenomenological Review on Quark-Gluon Plasma: Concepts vs. Observations”. In: *Universe* 3.1 (2017), p. 7. DOI: 10.3390/universe3010007. arXiv: 1611.01533.

[20] CERN. *Facts and figures about the LHC.* 2019. URL: https://home.cern/resources/faqs/facts-and-figures-about-lhc (visited on 01/23/2019).

[21] M. Vretenar. “Linear accelerators”. In: *CERN Yellow Reports* 1.0 (2013), pp. 225–249. DOI: 10.5170/CERN-2013-001.225. arXiv: 1303.6766.

[22] S. Gilardoni et al. “Fifty years of the CERN Proton Synchrotron : Volume 2”. In: CERN Yellow Reports: Monographs (2013). DOI: 10.5170/CERN-2013-005. arXiv: 1309.6923.

[23] M. Benedikt and F. Zimmermann. “Proton Colliders at the Energy Frontier”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 907 (2018), pp. 200 –208. ISSN: 0168-9002. DOI: 10.1016/j.nima.2018.03.021. arXiv: 1803.09723.

[24] N. A. Tahir, R. Schmidt, M. Brugger, R. Assmann, A. V. Shutov, I. V. Lomonosov, A. R. Piriz, D. H. H. Hoffmann, C. Deutsch, and V. E. Fortov. “The CERN Super Proton Synchrotron as a tool to study high energy density physics”. In: *New Journal of Physics* 10.7 (2008), p. 073028. DOI: 10.1088/1367-2630/10/7/073028.

[25] The CMS Collaboration. “The CMS Experiment at the CERN LHC”. In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[26] G. Aad et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.

[27] The ALICE Collaboration. “The ALICE experiment at the CERN LHC”. In: *Journal of Instrumentation* 3.08 (2008), S08002–S08002. DOI: 10.1088/1748-0221/3/08/s08002.

[28] The LHCb Collaboration. “The LHCb Detector at the LHC”. In: *JINST* 3 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[29] J. Nielsen. “Fundamentals of LHC Experiments”. In: *String Theory and Its Applications* (2011), pp. 127–152. DOI: 10.1142/9789814350525_0003. arXiv: 1106.2516.

[30] T.-M. Yan and S. D. Drell. “The Parton Model and its Applications”. In: *International Journal of Modern Physics A* 29.30 (2014), p. 1430071. DOI: 10.1142/S0217751X14300713. arXiv: 1409.0051.

[31] W. Herr and B. Muratori. “Concept of luminosity”. In: CERN-2006-002 (2006), pp. 361–378. URL: http://cds.cern.ch/record/941318.

[32] W. Stirling. *Parton luminosity and cross section plots.* 2012. URL: http://www.hep.ph.ic.ac.uk/~wstirlin/plots/plots.html (visited on 01/11/2019).

[33] S. S. Ghosh. "Highlights from the Compact Muon Solenoid (CMS) Experiment". In: *Universe* 5.1 (2019). DOI: 10.3390/universe5010028. arXiv: 1901.05340.

[34] Eugenio Berti, for the LHCf collaboration. *The LHCf experiment: present status and physics results.* 2017. arXiv: 1710.03991.

[35] G. Latino, for the TOTEM Collaboration. *The TOTEM Experiment at the LHC.* 2009. arXiv: 0905.2936.

[36] Vasiliki A. Mitsou, for the MoEDAL Collaboration. "The MoEDAL experiment at the LHC: status and results". In: *Journal of Physics: Conference Series* 873 (2017), p. 012010. DOI: 10.1088/1742-6596/873/1/012010. arXiv: 1703.07141.

[37] The CMS Collaboration. "CMS Physics: Technical Design Report Volume 1: Detector Performance and Software". In: Technical Design Report CMS (2006). There is an error on cover due to a technical problem for some items. URL: https://cds.cern.ch/record/922757.

[38] The CMS Collaboration. "CMS technical design report, volume II: Physics performance". In: *J. Phys.* G34.6 (2007), pp. 995–1579. DOI: 10.1088/0954-3899/34/6/S01.

[39] University of Zurich. *Simple example of 3D axes with spherical coordinates.* 2017. URL: https://wiki.physik.uzh.ch/cms/latex:example_spherical_coordinates (visited on 02/01/2019).

[40] The CMS Collaboration. "Observation of $t\bar{t}H$ Production". In: *Phys. Rev. Lett.* 120 (23 2018), p. 231801. DOI: 10.1103/PhysRevLett.120.231801.

[41] V. Barger, K. Hagiwara, and Y.-J. Zheng. *Probing the Higgs Yukawa coupling to the top-quark at the LHC via single top + Higgs production.* 2018. arXiv: 1807.00281.

[42] The CMS Collaboration. *Combination of searches for Higgs boson pair production in proton-proton collisions at* $\sqrt{s} = 13$ TeV. Tech. rep. CMS-PAS-HIG-17-030. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2628486.

[43] The CMS Collaboration. *Measurements of the* pp → WZ *inclusive and differential production cross section and constraints on charged anomalous triple gauge couplings at* $\sqrt{s} = 13$ TeV. Tech. rep. CMS-PAS-SMP-18-002. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2628761.

[44] The CMS Collaboration. *Measurement of associated production of W bosons with charm quarks in proton-proton collisions at* $\sqrt{s} = 13$ TeV *with the CMS experiment at the LHC.* Tech. rep. CMS-PAS-SMP-17-014. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2314570.

[45] The CMS Collaboration. *Measurement of the single top quark and antiquark production cross sections in the t channel and their ratio in pp collisions at* $\sqrt{s} = 13$ *TeV.* Tech. rep. CMS-PAS-TOP-17-011. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2628541.

[46] The CMS Collaboration. *Search for new physics in final states with a single photon and missing transverse momentum in proton-proton collisions at* $\sqrt{s} = 13$ *TeV.* Tech. rep. Geneva: CERN, 2018. arXiv: 1810.00196. URL: https://cds.cern.ch/record/2640855.

[47] The CMS Collaboration. “Search for new physics in events with a leptonically decaying Z boson and a large transverse momentum imbalance in proton-proton collisions at $\sqrt{s} = 13$ TeV”. In: *The European Physical Journal C* 78.4 (2018), p. 291. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-5740-1. arXiv: 1711.00431.

[48] The CMS Collaboration. *Search for new particles decaying to a jet and an emerging jet.* Tech. rep. CMS-PAS-EXO-18-001. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2625123.

[49] P. Schwaller, D. Stolarski, and A. Weiler. “Emerging Jets”. In: *Journal of High Energy Physics* 2015.5 (2015), p. 59. ISSN: 1029-8479. DOI: 10.1007/JHEP05(2015)059. arXiv: 1502.05409.

[50] Y. Bai and P. Schwaller. “Scale of dark QCD”. In: *Phys. Rev. D* 89 (6 2014), p. 063522. DOI: 10.1103/PhysRevD.89.063522. URL: https://link.aps.org/doi/10.1103/PhysRevD.89.063522.

[51] D. Barney. “CMS Detector Slice”. CMS Collection. 2016. URL: https://cds.cern.ch/record/2120661.

[52] N. Andreev, E. Barzi, E. Borissov, L. Elementi, V. S. Kashikhin, V. Lombardo, A. Rusy, D. Turrioni, R. Yamada, and A. V. Zlobin. “Development of Rutherford-Type Cables for High Field Accelerator Magnets at Fermilab”. In: *IEEE Transactions on Applied Superconductivity* 17.2 (2007), pp. 1027–1030. ISSN: 1051-8223. DOI: 10.1109/TASC.2007.898441.

[53] The CMS Collaboration. *The CMS hadron calorimeter project: Technical Design Report.* Technical Design Report CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/357153.

[54] The CMS Collaboration. “The performance of the CMS muon detector in proton-proton collisions at $\sqrt{s} = 7$ TeV at the LHC”. In: *JINST* 8.CMS-MUO-11-001. CERN-PH-EP-2013-072 (2013), P11002. 101 p. URL: http://cds.cern.ch/record/1558674.

[55] S. Cittolin, A. Rácz, and P. Sphicas. *CMS The TriDAS Project: Technical Design Report, Volume 2: Data Acquisition and High-Level Trigger. CMS trigger and data-acquisition project.* Technical Design Report CMS. Geneva: CERN, 2002. URL: http://cds.cern.ch/record/578006.

[56] The CMS Collaboration. “Particle-flow reconstruction and global event description with the CMS detector”. In: *Journal of Instrumentation* 12.10 (2017), P10003–P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965.

[57] The CMS Collaboration. “Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV”. In: *Journal of Instrumentation* 13.06 (2018), P06015–P06015. DOI: 10.1088/1748-0221/13/06/P06015. arXiv: 1804.04528.

[58] The CMS Collaboration. “Performance of electron reconstruction and selection with the CMS detector in proton-proton collisions at $\sqrt{s} = 8$ TeV”. In: *Journal of Instrumentation* 10.06 (2015), P06005–P06005. DOI: 10.1088/1748-0221/10/06/P06005. arXiv: 1502.02701.

[59] W. Adam, R. Frühwirth, A. Strandlie, and T. Todorov. “Reconstruction of electrons with the Gaussian-sum filter in the CMS tracker at LHC”. In: *Journal of Physics G: Nuclear and Particle Physics* 31.9 (2005), N9–N20. DOI: 10.1088/0954-3899/31/9/N01. arXiv: physics/0306087.

[60] F. Beaudette. *The CMS Particle Flow Algorithm.* 2017. URL: https://indico.in2p3.fr/event/14415/contributions/18288/attachments/15167/18631/PF-CMS.pdf (visited on 02/13/2019).

[61] R. Frühwirth. "Application of Kalman filtering to track and vertex fitting". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 262.2 (1987), pp. 444 –450. ISSN: 0168-9002. DOI: 10.1016/0168-9002(87)90887-4.

[62] J. Strologas. "Performance of Jet reconstruction in CMS at 13 TeV". In: *PoS* ICHEP2016 (2016), p. 736. DOI: 10.22323/1.282.0736.

[63] D. Bertolini, P. Harris, M. Low, and N. Tran. "Pileup Per Particle Identification". In: *Journal of High Energy Physics* 2014.10 (2014), p. 59. DOI: 10.1007/JHEP10(2014)059. arXiv: 1407.6013.

[64] A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Hoche, H. Hoeth, F. Krauss, L. Lonnblad, E. Nurse, P. Richardson, S. Schumann, M. H. Seymour, T. Sjostrand, P. Skands, and B. Webber. "General-purpose event generators for LHC physics". In: *Physics Reports* 504.5 (2011), pp. 145 –233. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2011.03.005. arXiv: 1101.2599.

[65] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands. "An Introduction to PYTHIA 8.2". In: *Computer Physics Communications* 191 (2015), pp. 159 –177. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2015.01.024. arXiv: 1410.3012.

[66] C. Oleari. "The POWHEG-BOX". In: *Nuclear Physics B - Proceedings Supplements* 205-206 (2010), pp. 36 –41. ISSN: 0920-5632. DOI: 10.1016/j.nuclphysbps.2010.08.016. arXiv: 1007.3893.

[67] J. Bellm et al. "Herwig 7.0/Herwig++ 3.0 release note". In: *Eur. Phys. J.* C76.4 (2016), p. 196. DOI: 10.1140/epjc/s10052-016-4018-8.

[68] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer. "MadGraph 5 : Going Beyond". In: *Journal of High Energy Physics* 2011.6 (2011), p. 128. DOI: 10.1007/JHEP06(2011)128. arXiv: 1106.0522.

[69] M. H. Seymour and M. Marx. *Monte Carlo Event Generators.* 2013. arXiv: 1304.6677.

[70] S. Agostinelli et al. "GEANT4: A Simulation toolkit". In: *Nucl. Instrum. Meth.* A506 (2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.

[71] V. Ivantchenko, D. Lange, and M. Hildreth. *CMS Full Simulation for Run-II.* 2015. URL: https://indico.cern.ch/event/304944/contributions/1672508/ (visited on 02/19/2019).

[72] G. Petrucciani, A. Rizzi, C. Vuosalo, and for the CMS Collaboration. "Mini-AOD: A New Analysis Data Format for CMS". In: *Journal of Physics: Conference Series* 664.7 (2015), p. 072052. DOI: 10.1088/1742-6596/664/7/072052. arXiv: 1702.04685.

[73] M. Peruzzi. *NanoAOD Documentation.* 2019. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookNanoAOD (visited on 02/18/2019).

[74] S. D. Drell and T.-M. Yan. "Massive Lepton Pair Production in Hadron-Hadron Collisions at High-Energies". In: *Phys. Rev. Lett.* 25 (1970). [Erratum: Phys. Rev. Lett.25,902(1970)], pp. 316–320. DOI: 10.1103/PhysRevLett.25.316,10.1103/PhysRevLett.25.902.2.

[75] K.A. Olive and the Particle Data Group. “Review of Particle Physics”. In: *Chinese Physics C* 38.9 (2014), p. 090001. DOI: 10.1088/1674-1137/38/9/090001.

[76] R. Ruiz. *Drell-Yan, Drell-Yan with Jets, Drell-Yan with all the Jets.* 2015. URL: https://www.quantumdiaries.org/2015/05/18/dy-resummation/ (visited on 01/16/2019).

[77] M. W. Krasny, F. Fayette, W. Placzek, and A. Siodmok. “Z-boson as the standard candle for high precision W-boson physics at LHC”. In: *The European Physical Journal C* 51.3 (2007), pp. 607–617. DOI: 10.1140/epjc/s10052-007-0321-8. arXiv: hep-ph/0702251.

[78] M. Jindal, D. Bourilkov, K. Mazumdar, and J. Singh. “Drell–Yan process at Large Hadron Collider”. In: *PRAMANA c Indian Academy of Sciences* 7615 (Mar. 2011). DOI: 10.1007/s12043-011-0014-6.

[79] R. Radogna. *Search for high-mass resonances decaying into muon pairs with the CMS experiment at LHC.* 2016. URL: https://cds.cern.ch/record/2205870/files/CERN-THESIS-2016-084.pdf (visited on 01/30/2019).

[80] Z. Rezaei and S. P. Mehdiabadi. *The LHC Drell-Yan Measurements as a Constraint for the Noncommutative Space-Time.* 2018. arXiv: 1809.04502.

[81] F. Kohn. *Measurement of the charge asymmetry in top quark pair production in pp collision data at $\sqrt{s} = 7$ TeV using the ATLAS detector.* 2012. arXiv: 1204.0952.

[82] F. Deliot, N. Hadley, S. Parke, and T. Schwarz. “Properties of the Top Quark”. In: *Annual Review of Nuclear and Particle Science* 64.1 (2014), pp. 363–381. DOI: 10.1146/annurev-nucl-102313-025655. arXiv: 1803.00656.

[83] T. Gershon. “Overview of the CKM Matrix”. In: *Pramana* 79.5 (2012), pp. 1091–1108. DOI: 10.1007/s12043-012-0418-y. arXiv: 1112.1984.

[84] T. N. Pham. *CKM Matrix Elements.* 2011. arXiv: 1110.6050.

[85] S. L. Glashow, J. Iliopoulos, and L. Maiani. “Weak Interactions with Lepton-Hadron Symmetry”. In: *Physical Review D* 2.7 (1970), pp. 1285–1292. DOI: 10.1103/physrevd.2.1285.

[86] F. Hubaut. *Top physics at LHC with ttbar events.* 2006. arXiv: hep-ph/0605029.

[87] G. Soyez. *Pileup mitigation at the LHC: a theorist’s view.* 2018. arXiv: 1801.09721.

[88] The ATLAS collaboration. *Estimation of non-prompt and fake lepton backgrounds in final states with top quarks produced in proton-proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector.* Tech. rep. 2014. URL: http://inspirehep.net/record/1319687/files/ATLAS-CONF-2014-058.pdf.

[89] J. Mellenthi. *Measurement of muon misidentification rates in $Z \to \mu\mu$ events for the ATLAS detector.* 2013. URL: https://indico.mpp.mpg.de/event/2410/session/17/contribution/29/material/slides/0.pdf (visited on 01/19/2019).

[90] The ATLAS Collaboration. “A search for prompt lepton-jets in *pp* collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector”. In: *Journal of High Energy Physics* 2016.2 (2016), p. 62. DOI: 10.1007/JHEP02(2016)062. arXiv: 1511.05542.

[91] The ATLAS Collaboration. “Search for anomalous production of prompt same-sign lepton pairs and pair-produced doubly charged Higgs bosons with $\sqrt{s} = 8$ TeV pp collisions using the ATLAS detector”. In: *Journal of High Energy Physics* 2015.3 (2015), p. 41. DOI: 10.1007/JHEP03(2015)041. arXiv: 1412.0237.

[92] The CMS Collaboration. "Search for supersymmetry with multiple charged leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV". In: *The European Physical Journal C* 77.9 (2017), p. 635. DOI: 10.1140/epjc/s10052-017-5182-1. arXiv: 1701.06940.

[93] P. Charitos. *Deep learning and the quest for new physics at the LHC.* 2017. URL: https://ep-news.web.cern.ch/content/deep-learning-and-quest-new-physics-lhc (visited on 01/18/2019).

[94] A. Calderón. *Muons in CMS.* 2012. URL: http://agenda.ciemat.es/indico/event/673/contribution/5/material/slides/0.pdf (visited on 01/19/2019).

[95] T. P. S. Gillam. "Identifying fake leptons in ATLAS while hunting SUSY in 8 TeV proton-proton collisions". en. PhD thesis. 2015. DOI: 10.17863/cam.16619.

[96] E. Meschi, T Monteiro, C. Seez, and P. Vikas. *Electron Reconstruction in the CMS Electromagnetic Calorimeter.* Tech. rep. CMS-NOTE-2001-034. Geneva: CERN, 2001. URL: http://cds.cern.ch/record/687345.

[97] C. Seez. *Shower shape variables.* 2011. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideEgammaShowerShape (visited on 01/18/2019).

[98] S. Harper and C. Shepherd-Themistocleous. *Improving Sigma Eta Eta.* 2006. URL: https://indico.cern.ch/event/27560/contributions/1618477/attachments/499353/689807/sigmaEtaEta_sharper.pdf.pdf (visited on 01/18/2019).

[99] I. Kravchenko and R. Kamalieddin. *Update on cut-based electron ID for 80X samples.* 2016. URL: https://indico.cern.ch/event/482677/contributions/2259342/attachments/1316731/1972911/talk_electron_ID_spring16_update.pdf (visited on 01/21/2019).

[100] The CMS Collaboration. "Performance of CMS muon reconstruction in pp collision events at $\sqrt{s} = 7$ TeV". In: *Journal of Instrumentation* 7.10 (2012), P10002–P10002. DOI: 10.1088/1748-0221/7/10/P10002. arXiv: 1206.4071.

[101] S. S. Cruz. *Baseline muon selections for Run-II.* 2019. URL: https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideMuonIdRun2\#Loose_Muon (visited on 01/21/2019).

[102] S. Harper. *Multivariate Electron Identification for Run2.* 2018. URL: https://twiki.cern.ch/twiki/bin/view/CMS/MultivariateElectronIdentificationRun2\#Training_Details_and_Working_Poi (visited on 01/22/2019).

[103] H. Drucker and C. Cortes. "Boosting Decision Trees." In: vol. 8. Jan. 1995, pp. 479–485. URL: https://papers.nips.cc/paper/1059-boosting-decision-trees.pdf.

[104] F. Cossutti and J. Rembser. *Electron MVA Estimator Run 2 Variables.* 2018. URL: https://github.com/cms-sw/cmssw/blob/master/RecoEgamma/ElectronIdentification/data/ElectronMVAEstimatorRun2Variables.txt (visited on 01/22/2019).

[105] T. Cornelis, D. Dobur, I. Khvastunov, and W. Verbeke. *CMS Analysis Note - Search for tZq in trilepton events.* Available on CMS information server. 2018.

[106] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, M. Backes, T. Carli, O. Cohen, A. Christov, D. Dannheim, K. Danielowski, S. Henrot-Versille, M. Jachowski, K. Kraszewski, A. K. Jr., M. Kruk, Y. Mahalalel, R. Ospanov, X. Prudent, A. Robert, D. Schouten, F. Tegenfeldt, A. Voigt, K. Voss, M. Wolter, and A. Zemla. *TMVA - Toolkit for Multivariate Data Analysis.* 2007. arXiv: physics/0703039.

[107] The CMS Collaboration. "Identification of b-quark jets with the CMS experiment". In: *Journal of Instrumentation* 8.04 (2013), P04013–P04013. DOI: 10.1088/1748-0221/8/04/P04013. arXiv: 1211.4462.

[108] The CMS Collaboration. *Heavy flavor identification at CMS with deep neural networks.* Tech. rep. 2017. URL: `https://cds.cern.ch/record/2255736`.

[109] M. Stoye. *Deep learning in jet reconstruction at CMS.* Tech. rep. CMS-CR-2017-416. Geneva: CERN, 2017. URL: `https://cds.cern.ch/record/2293134`.

[110] S. May et al. *Lepton Isolation – Machine Learning Study.* 2018. URL: `https://indico.cern.ch/event/721595/contributions/2966340/attachments/1639191/2616664/IsolationMLStudy_CmsMLForum_SamuelMay_25Apr2018.pdf` (visited on 12/19/2018).

[111] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press, 2016.

[112] D. Shewan. *10 Companies Using Machine Learning in Cool Ways.* 2017. URL: `https://medium.com/marketing-and-entrepreneurship/10-companies-using-machine-learning-in-cool-ways-887c25f913c3` (visited on 02/20/2019).

[113] S. Raschka and V. Miralili. *Machine Learning mit Python und Scikit-learn und TensorFlow - Das umfassede Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics.* 2nd. Frechen, Germany: mitp Verlags GmbH & Co. KG, 2018. ISBN: 9783958457331.

[114] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. F. R. Jesus, R. F. Berriel, T. M. Paixão, F. Mutz, T. Oliveira-Santos, and A. F. D. Souza. *Self-Driving Cars: A Survey.* 2019. arXiv: `1901.04407`.

[115] J. M. Klusowski. *Complete Analysis of a Random Forest Model.* 2018. arXiv: `1805.02587`.

[116] L.-G. Xia. *Understanding the boosted decision tree methods with the weak-learner approximation.* 2018. arXiv: `1811.04822`.

[117] J. Schmidhuber. "Deep Learning in Neural Networks: An Overview". In: *Neural Networks* 61 (2015), pp. 85 –117. ISSN: 0893-6080. DOI: `10.1016/j.neunet.2014.09.003`. arXiv: `1404.7828`.

[118] Y.-C. Chen. *A Tutorial on Kernel Density Estimation and Recent Advances.* 2017. arXiv: `1704.03924`.

[119] A. Bethani, A. J. Bevan, J. Hays, and T. J. Stevenson. "Support Vector Machines and Generalisation in HEP". In: *Journal of Physics: Conference Series* 762 (2016), p. 012052. DOI: `10.1088/1742-6596/762/1/012052`. arXiv: `1610.09932`.

[120] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune. *Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning.* 2017. arXiv: `1712.06567`.

[121] D. Guest, K. Cranmer, and D. Whiteson. "Deep Learning and its Application to LHC Physics". In: *Annual Review of Nuclear and Particle Science* 68.1 (2018), pp. 161–181. DOI: `10.1146/annurev-nucl-101917-021019`. arXiv: `1806.11484`.

[122] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359 –366. ISSN: 0893-6080. DOI: `10.1016/0893-6080(89)90020-8`.

[123] H. Mhaskar and T. Poggio. *Deep vs. shallow networks : An approximation theory perspective.* 2016. arXiv: `1608.03287`.

[124] L. Lu, Y. Su, and G. E. Karniadakis. *Collapse of Deep and Narrow Neural Nets.* 2018. arXiv: `1808.04947`.

[125] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[126] Q. Meng, D. Catchpoole, D. Skillicorn, and P. J. Kennedy. "Relational Autoencoder for Feature Extraction". In: *2017 International Joint Conference on Neural Networks (IJCNN)* (2017), pp. 364–371. DOI: 10.1109/IJCNN.2017.7965877. arXiv: 1802.03145.

[127] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai. *Gradient Descent Finds Global Minima of Deep Neural Networks.* 2018. arXiv: 1811.03804.

[128] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning.* 2018. arXiv: 1811.03378.

[129] S. C. Douglas and J. Yu. *Why ReLU Units Sometimes Die: Analysis of Single-Unit Error Backpropagation in Neural Networks.* 2018. arXiv: 1812.05981.

[130] S. Jadon. *Introduction to Different Activation Functions for Deep Learning.* 2018. URL: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092 (visited on 02/21/2019).

[131] L. N. Smith. *A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay.* 2018. arXiv: 1803.09820.

[132] S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* 2015. arXiv: 1502.03167.

[133] C. Wei, J. D. Lee, Q. Liu, and T. Ma. *On the Margin Theory of Feedforward Neural Networks.* 2018. arXiv: 1810.05369.

[134] A. Sherstinsky. *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.* 2018. arXiv: 1808.03314.

[135] C. Olah. *Understanding LSTM Networks.* 2015. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 02/21/2019).

[136] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.* 2014. arXiv: 1412.3555.

[137] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[138] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.* 2014. arXiv: 1409.1259.

[139] J. M. Zhou. *What is the architecture behind the Keras LSTM cell?* 2018. URL: https://i.stack.imgur.com/RHNrZ.jpg (visited on 02/22/2019).

[140] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems 2.* Ed. by D. S. Touretzky. Morgan-Kaufmann, 1990, pp. 396–404. URL: http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

[141] J. Koushik. *Understanding Convolutional Neural Networks.* 2016. arXiv: 1605.09081.

[142] J. Nagi, F. Ducatelle, G. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. Maria Gambardella. "Max-pooling convolutional neural networks for vision-based hand gesture recognition". In: Nov. 2011, pp. 342–347. DOI: 10.1109/ICSIPA.2011.6144164.

[143] T. Williams and R. Li. “An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification”. In: *Journal of Software Engineering and Applications* 11 (Jan. 2018), pp. 69–88. DOI: 10.4236/jsea.2018.112004.

[144] K. Albertsson et al. *Machine Learning in High Energy Physics Community White Paper.* 2018. arXiv: 1807.02876.

[145] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1.* NIPS’12. Curran Associates Inc., 2012, pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[146] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. “Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images”. In: *Advances in Neural Information Processing Systems 25* (2012), pp. 2843–2851. URL: http://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images.pdf.

[147] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. “Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013* (2013). Ed. by K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, pp. 411–418. DOI: 10.1007/978-3-642-40763-5_51.

[148] P. Baldi, P. Sadowski, and D. Whiteson. *Searching for Exotic Particles in High-Energy Physics with Deep Learning.* Tech. rep. 2014. DOI: 10.1038/ncomms5308. arXiv: 1402.4735.

[149] S. Mehta, M. Verzetti, J. Kieseler, and M. Stoye. *DeepJet: A Machine Learning Environment forHigh-energy Physics.* 2018. URL: https://openreview.net/pdf?id=B1gTR_Q0cQ (visited on 02/25/2019).

[150] J. Kieseler. *Machine Learning in CMS - More Experimental Point of View.* 2017. URL: https://indico.cern.ch/event/579660/contributions/2582146/attachments/1495720/2327116/JK_MLatCMS.pdf (visited on 02/25/2019).

[151] M. Verzetti. *Performance of the CMS.DeepJet b tagging algorithm using 41.9 /fb of data from proton-proton collisions at 13TeV with Phase 1 CMS detector.* 2018. URL: https://twiki.cern.ch/twiki/bin/genpdf/CMSPublic/BTV13TeV2017DeepJet (visited on 02/25/2019).

[152] S. Farrell, D. Anderson, P. Calafiura, G. Cerati, L. Gray, J. Kowalkowski, M. Mudigonda, Prabhat, P. Spentzouris, M. Spiropoulou, A. Tsaris, J.-R. Vlimant, and S. Zheng. “The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking”. In: *EPJ Web Conf.* 150 (2017), p. 00003. DOI: 10.1051/epjconf/201715000003.

[153] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J.-R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati, L. Gray, J. Kowalkowski, P. Spentzouris, and A. Tsaris. *Novel deep learning methods for track reconstruction.* 2018. arXiv: 1810.06111.

[154] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. “Geometric deep learning: going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42. DOI: 10.1109/MSP.2017.2693418. arXiv: 1611.08097.

[155] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Networks.* 2014. arXiv: 1406.2661.

[156] B. Blagojevic. *Generating Letters Using Generative Adversarial Networks.* 2018. URL: https://medium.com/ml-everything/generating-letters-using-generative-adversarial-networks-gans-161b0be3c229 (visited on 02/26/2019).

[157] M. Paganini, L. de Oliveira, and B. Nachman. "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeters". In: *Phys. Rev. Lett.* 120 (4 2018), p. 042003. DOI: 10.1103/PhysRevLett.120.042003. arXiv: 1705.02355.

[158] Todor, Trendafilov, and Ivanov. *Software Guide on CRAB.* 2018. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideCrab (visited on 12/04/2018).

[159] Brun, Rene and the Fons Rademakers. *ROOT Data Analysis Framework - User's Guide.* 2018. URL: https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html (visited on 12/04/2018).

[160] M. Buda, A. Maki, and M. A. Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106 (2018), pp. 249 –259. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2018.07.011. arXiv: 1710.05381.

[161] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. "Self-Normalizing Neural Networks". In: *CoRR* abs/1706.02515 (2017). arXiv: 1706.02515. URL: http://arxiv.org/abs/1706.02515.

[162] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization.* 2014. arXiv: 1412.6980.

[163] J. Kieseler et al. *DL4Jets.* 2018. URL: https://github.com/DL4Jets (visited on 01/10/2019).

[164] F. Chollet et al. *Keras.* https://keras.io. 2015.

[165] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[166] S. Cook. *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs.* 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124159334, 9780124159884. URL: http://www.hds.bme.hu/~fhegedus/C++/Shane\%20Cook\%20-\%20CUDA\%20Programming\%20-olvasOM.pdf.

[167] Particle Data Group, M. Tanabashi, et al. "Review of Particle Physics". In: *Phys. Rev. D* 98 (2018), p. 030001. DOI: 10.1103/PhysRevD.98.030001.

[168] N. Adam, J. Berryhill, V. Halyo, A. Hunt, and K. Mishra. *Generic Tag and Probe Tool for Measuring Efficiency at CMS with Early Data.* 2009. URL: http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=AN2009_111_v1.pdf (visited on 01/11/2019).