



UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

CORSO DI LAUREA IN FISICA

TESI DI LAUREA TRIENNALE

A machine learning approach to the electrons and photons classification with the ATLAS detector at the LHC

Autore:

Pietro Daniele

Matricola:

906962

Codice P.A.C.S.:

07.05.-t

Relatore:

**Prof. Leonardo Carlo
Carminati**

Corelatori:

**Dott. Ruggero Turra
Dott. Davide Mungo**

Anno accademico 2019-2020

Contents

1	LHC and the ATLAS detector	3
1.1	The LHC	3
1.1.1	LHC Layout	3
1.1.2	The CERN accelerator complex	5
1.1.3	Proton-proton collisions	5
1.2	ATLAS	8
1.2.1	Coordinate system	8
1.2.2	Magnet system	9
1.2.3	Inner detector	10
1.2.4	Calorimetry system	11
1.2.5	Muon spectrometer	13
1.2.6	Trigger system	14
1.3	ATLAS Physics Programme	15
2	Electron and photon reconstruction	18
2.1	The topo-cluster reconstruction	20
2.2	Track reconstruction	20
2.3	Track-cluster matching and photon conversion reconstruction	21
2.4	Supercluster reconstruction	22
2.5	Creation of electrons and photons for analysis	23
2.6	Calibration	24
2.7	Identification	25
2.7.1	Electron identification	25
2.7.2	Photon identification	26
2.8	Electron and photon isolation	29
2.8.1	Electron isolation criteria and efficiency measurements	30
2.8.2	Photon isolation criteria and efficiency measurements	30
3	Machine Learning and Gradient Boosted Trees	31
3.1	Supervised Learning	32
3.1.1	Model and Parameters	32
3.1.2	Objective Function	32
3.2	Gradient boosted trees	33

3.2.1	Decision Tree	33
3.2.2	Decision Tree Ensembles	33
3.2.3	Gradient Bosting Decision Trees	34
3.2.4	Learn the tree structure	36
3.3	LightGBM	36
4	BDT: Electron and photon classification	38
4.1	BDT training	39
4.1.1	Discriminating features	39
4.1.2	Training and Model	42
4.1.3	Parameters and Hyperparameter optimization	44
4.2	BDT applied on test set	45
4.3	BDT results	46
5	Conclusion	49
A	Conversion symmetry	51
B	Parameters and Hyperparameters description	53
C	Ambiguity Tools	54

Chapter 1

LHC and the ATLAS detector

1.1 The LHC

The CERN Large Hadron Collider (LHC) [1] is a two-ring, superconducting protons and heavy ions accelerator and collider installed in the 27 km-long LEP tunnel. It provides pp collisions at an unprecedented center of mass energy $\sqrt{s} = 13$ TeV.

Inside the accelerator, two high-energy particle beams travel close to the speed of light before they are made to collide. The beams travel in opposite directions in separate beam pipes (two tubes kept at ultrahigh vacuum). They are guided around the accelerator ring by strong magnetic fields maintained by superconducting electromagnets[2],[3].

1.1.1 LHC Layout

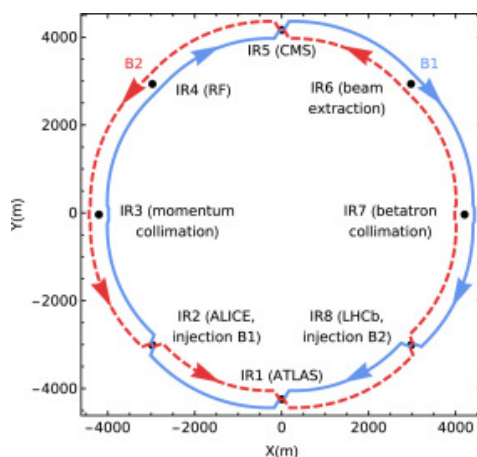


Figure 1.1: LHC structure

The basic layout of the LHC follows the LEP tunnel geometry, as shown in Figure 1.1. The LHC has eight arcs and eight straight sections. Each straight section is approximately 528 m long and can serve as an experimental or utility insertion. The two high

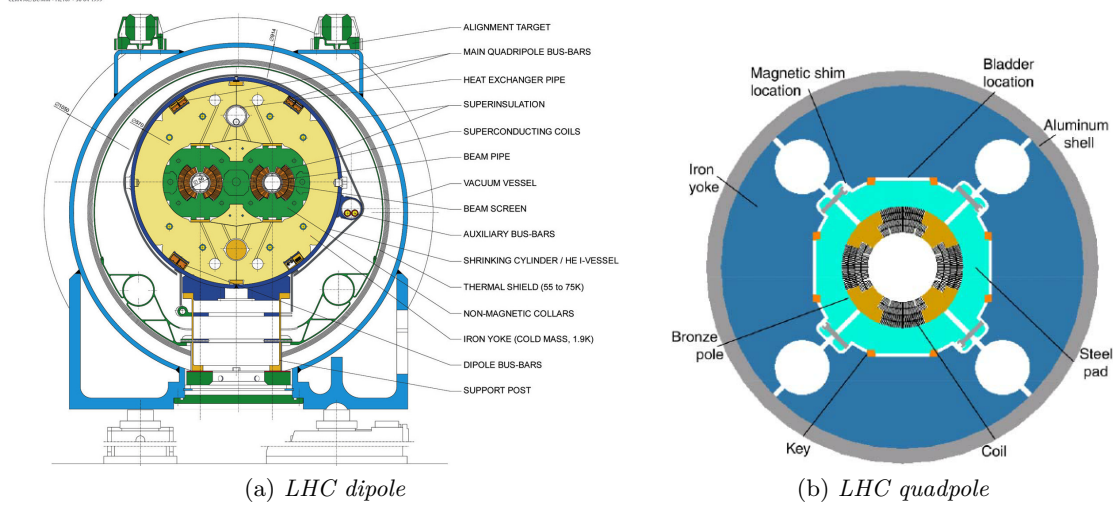
LHC DIPOLE : STANDARD CROSS-SECTION

Figure 1.2: LHC's superconducting magnets

luminosity experimental insertions are located at diametrically opposite straight sections: the ATLAS experiment is located at point 1 (IR1) and the CMS experiment at point 5 (IR5).

Two more experimental insertions are located at point 2 and point 8 which also contain the injection systems for Beam 1 and Beam 2, respectively. The injection kick occurs in the vertical plane with the two beams arriving at the LHC from below the LHC reference plane. The beams only cross from one magnet bore to the other at these four locations.

The remaining four straight sections do not have beam crossings. Insertion 3 and 7 each contain two collimation systems. Insertion 4 contains two RF systems, one independent system for each LHC beam. The straight section at point 6 contains the beam dump insertion where the two beams are vertically extracted from the machine using a combination of horizontally deflecting fast-pulsed ('kicker') magnets and vertically-deflecting double steel septum magnets. Each beam features an independent abort system. [1] The protons travel inside along the LHC ring in opposite direction. The LHC beams are controlled by superconducting magnets, which have a working temperature of 1.9 K. There are two kinds of superconducting magnets (Figure 1.2):

- the superconducting dipole magnets, which thanks to a 8.33 T magnetic field drive protons along the ring (circular orbit);
- superconducting quadrupole magnets, which keep the beams focused.

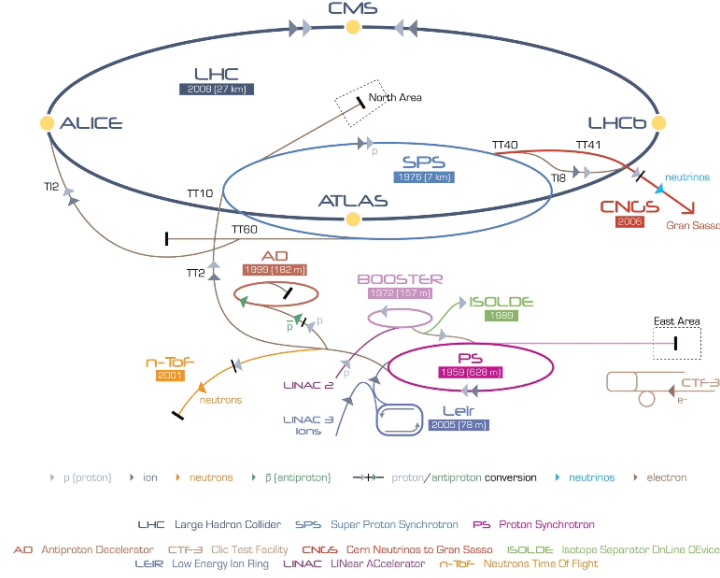


Figure 1.3: LHC and CERN's complex accelerator

1.1.2 The CERN accelerator complex

Before being insert into LHC ring, particle beam is accelerated by the CERN accelerator complex [4]. It is a succession of machines which accelerate particles at increasingly higher energies. Each machine injects the beam into the next one, which takes over to bring the beam to an even higher energy, and so on. In the LHC, the last element of this chain, each particle beam is accelerated up to the record energy of 6.5 TeV. At the CERN complex, protons are obtained by stripping electrons from hydrogen atoms, which are taken from a bottle containing hydrogen. Then protons are injected into the PS Booster (PSB) at an energy of 50 MeV from Linac2. The booster accelerates them to 1.4 GeV. The beam is then fed to the Proton Synchrotron (PS) where it is accelerated to 25 GeV. Protons are then sent to the Super Proton Synchrotron (SPS) where they are accelerated to 450 GeV. They are finally transferred to the LHC (both in a clockwise and an anticlockwise direction) where they are accelerated for 20 minutes to 6.5 TeV. Beams circulate for many hours inside the LHC beam pipes under normal operating conditions. In addition to accelerating protons, the accelerator complex can also accelerate lead ions.

1.1.3 Proton-proton collisions

The number of events per second generated in the LHC collisions is given by:

$$N_i = L\sigma_i$$

where σ_i is the cross section for the process under study and L the instantaneous machine luminosity. L depends only on the beam parameters and can be written for a Gaussian

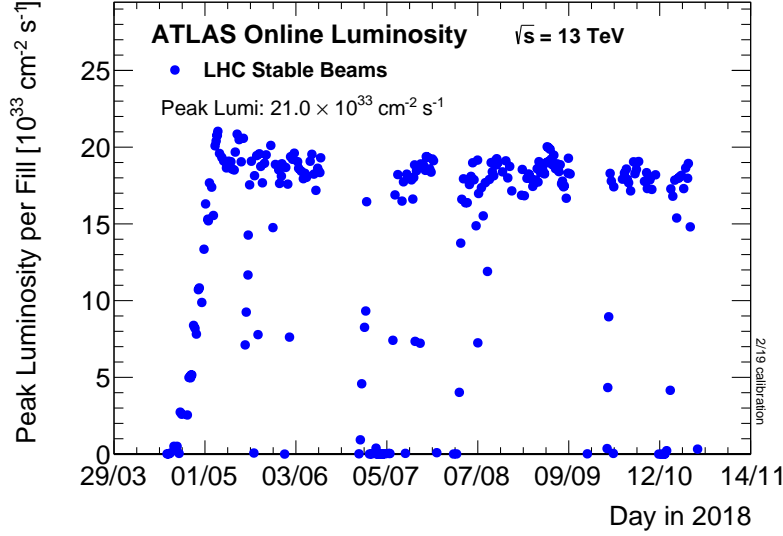


Figure 1.4: The peak instantaneous luminosity delivered to ATLAS during stable beams for pp collisions at 13 TeV centre-of-mass energy per fill 2018

beam distribution as:

$$L = \frac{N_b^2 n_b f_{rev} \gamma_r}{4\pi \epsilon_n \beta^*} F$$

where N_b is the number of particles per bunch, n_b the number of bunches per beam, f_{rev} the revolution frequency, γ_r the relativistic gamma factor, ϵ_n the normalized transverse beam emittance, β^* the beta function at the collision point and F the geometric luminosity reduction factor due to the crossing angle at the IP. The luminosity had various values during ATLAS activity as shown in the Figure 1.4.

For the nominal value of the luminosity $10^{34} \text{cm}^{-2} \text{s}^{-1}$ the total inelastic proton-proton cross-section is about 80 mb [2] at $\sqrt{s} = 14$ TeV. Therefore, the event rate R , defined as the number of events produced per second by the pp interactions, is expected to be:

$$R = \sigma L = 80 \text{mb} \times 10^{34} \text{cm}^{-2} \text{s}^{-1} \simeq 10^9 \text{s}^{-1}$$

e The number of events for each process is related to luminosity, it is proportional to the integrated luminosity shown in Figure 1.5.

There are two types of pp collisions:

- **Soft collisions:** they are the most of the collisions and they are large-distance collisions between the two incoming protons. They are called "soft" because the momentum transfer of the interaction is small. Due to this feature, particle scattering at large angle is suppressed and so, after collisions, particles have a large longitudinal momentum, but small transverse momentum ($< p_T > \simeq 500$ MeV) relative to the beam line. The final states arising from such interactions are called minimum bias events.

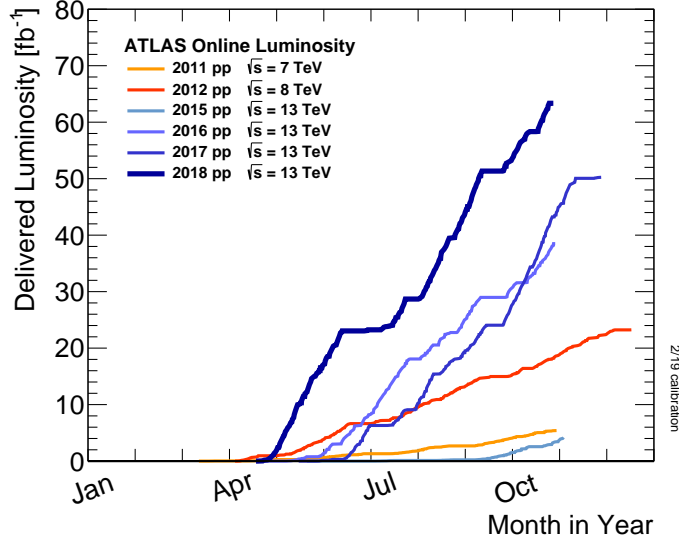


Figure 1.5: Luminosity measured in inverse femtobarn versus time for 2011-2018 (p-p data only)

- **Hard collisions:** monochromatic proton beams can be seen as beams of partons (quarks and gluons) with a wide band of energy. Occasionally, head-on collisions occur between two partons of the incoming protons. These are interactions at small distances, and therefore are characterised by large momentum transfers ("hard scattering"). In this case, particles in the final state can be produced at large angles with respect to the beam line (high p_T) and massive particles can be created. These are the interesting physics events at a collider but they are, however, rare compared to the soft interactions.

In the hard-scattering interactions of quarks and gluons at a hadron collider, the effective centre-of-mass energy of the interaction ($\sqrt{\hat{s}}$) is smaller than the centre-of-mass energy of the machine (\sqrt{s}) and is given by:

$$\sqrt{\hat{s}} = \sqrt{x_a x_b s}$$

where x_a and x_b are the fractions of the proton momentum carried by the two colliding partons. If $x_a \simeq x_b$, then the above relation becomes

$$\sqrt{\hat{s}} \simeq x\sqrt{s}$$

Therefore, in order to produce a particle of mass 100 GeV, two quarks (or gluons) which carry only 1% of the proton momentum are needed ($x \sim 0.01$), whereas a particle of mass 5 TeV can only be produced if two partons with $x \sim 0.35$ interact. The momentum distributions of quarks and gluons inside the proton are called *parton distribution functions*.

1.2 ATLAS

ATLAS [6] is one of two general-purpose detectors at the Large Hadron Collider. It investigates a wide range of physics, from the search for the Higgs boson to extra dimensions and particles that could make up dark matter. Although it has the same scientific goals as the CMS experiment, it uses different technical solutions and a different magnet-system design.

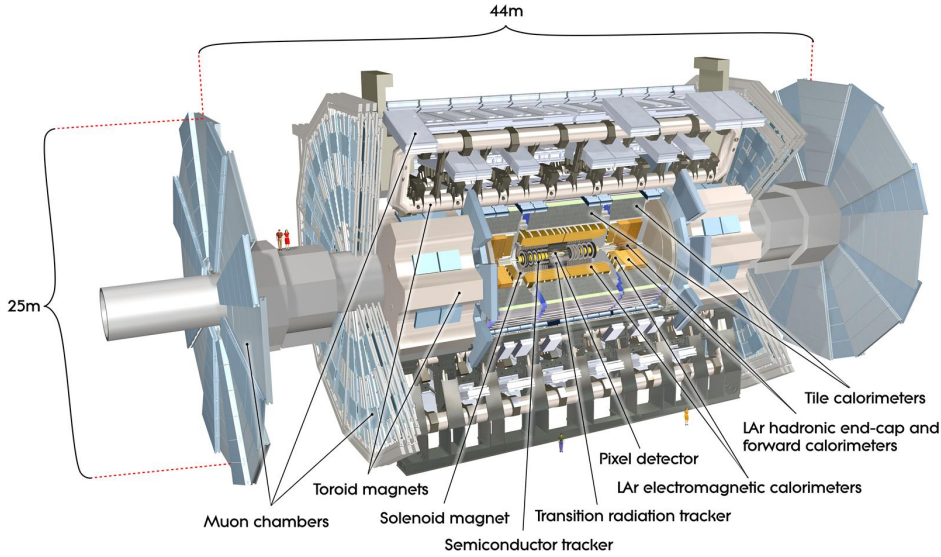


Figure 1.6: ATLAS structure

Beams of particles from the LHC collide at the centre of the ATLAS detector making collision debris in the form of new particles, which fly out from the collision point in all directions. Six different detecting subsystems arranged in layers around the collision point record the paths, momentum, and energy of the particles, allowing them to be individually identified. A complex magnet system bends the paths of charged particles so that their momenta can be measured.

The interactions in the ATLAS detectors create an enormous flow of data. To digest the data, ATLAS uses an advanced “trigger” system to tell the detector which events to record and which to ignore. Complex data-acquisition and computing systems are then used to analyse the recorded collision events. At 46 m long, 25 m high and 25 m wide, the 7000-tonne ATLAS detector is the largest volume particle detector ever constructed.

1.2.1 Coordinate system

The origin of the coordinate system is set in the nominal point of interaction. The beam direction defines the z-axis and the x-y plane is transverse to the beam direction. X-axis points from the interaction point to the centre of the LHC ring and Y-axis points

upwards. The side-A of the detector is defined as that with positive z and side-C is that with negative z .

Polar coordinate are also used: azimuthal angle ϕ is measured as usual around the beam axis, and the polar angle θ is the angle from the beam axis. Using θ , pseudo rapidity is defined as $\eta = -\ln \tan \frac{\theta}{2}$.

The transverse momentum p_T , the transverse momentum E_T , and the missing transverse energy E_T^{miss} are defined in the x-y plane unless stated otherwise. The distance ΔR in the pseudo rapidity-azimuthal angle space is defined as $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$.

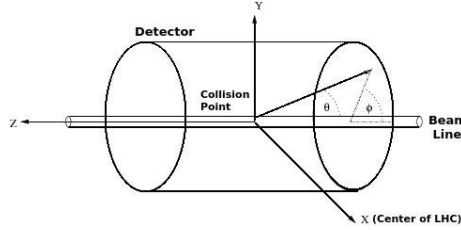


Figure 1.7: ATLAS coordinate system

1.2.2 Magnet system

ATLAS features a unique hybrid system of four large superconducting magnets. This magnetic system is 22 m in diameter and 26 m in length, with a stored energy of 1.6 GJ.

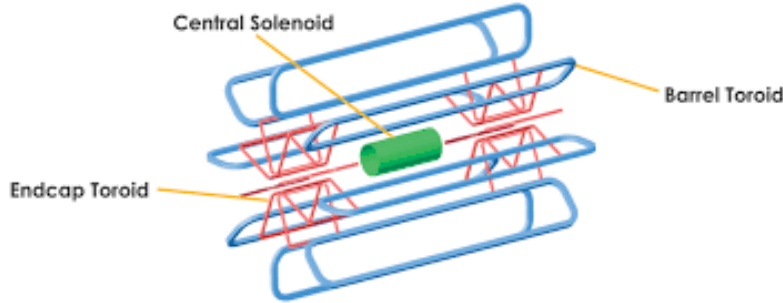


Figure 1.8: ATLAS magnet system

The ATLAS magnet system consists of:

- a solenoid ("central solenoid"), which is aligned on the beam axis and provides a 2 T axial magnetic field for the inner detector, while minimising the radiative thickness in front of the barrel electromagnetic calorimeter;
- a barrel toroid and two end-cap toroids, which produce a toroidal magnetic field of approximately 0.5 T and 1 T for the muon detectors in the central and end-cap regions, respectively.

1.2.3 Inner detector

The ATLAS Inner Detector (ID) is the inner-most ATLAS layer and it is immersed in a 2 T solenoidal field. It is designed to provide hermetic and robust pattern recognition, excellent momentum resolution and both primary and secondary vertex measurements for charged tracks within the pseudorapidity range $|\eta| < 2.5$. It also provides electron identification over $|\eta| < 2.0$ up to energy of about 150 GeV.

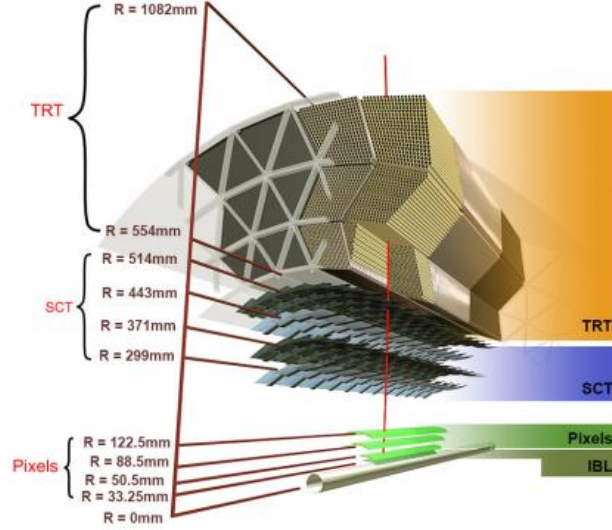


Figure 1.9: ATLAS Inner Detector structure

The ID is composed by three independent but complementary sub-detectors:

- **Pixel Detector:** [7] [8] it is the inner-most part of the ATLAS tracking system. It consists of 4 layers of barrel pixel detector and two end caps of three pixel disks each. The innermost pixel layer is a high-resolution pixel detector, called Insertable B-Layer (IBL). The Pixel Detector sits inside the 2T solenoidal magnetic field and contributes to the charged particle tracking of the ATLAS Inner Detector in the pseudo rapidity range of $|\eta| < 2.5$. Due to its high spatial resolution and 3-dimensional space-point measurement the Pixel Detector has a key-role in reconstruction of charged particle tracks. The 4-Layer Pixel Detector is crucial in the reconstruction of primary and secondary vertices which is essential for example in the photon conversions reconstruction.

The single point resolution is $12\ \mu\text{m}$ in the $R \cdot \phi$ and $75\text{-}112\ \mu\text{m}$ in z direction.

- **Semiconductor Tracker (SCT):** [7] [9] it consists of $61\ \text{m}^2$ of active silicon-strip detector modules and it is immersed in a 2 T solenoidal magnetic field. The SCT covers the radial region from 30 to 52 cm, with hermetic azimuthal coverage out to $|\eta| = 2.5$. Four cylindrical layers in the central region form the “barrel” detector,

and nine annular disks on each end of the barrel form the “endcaps”. The barrel layers are organized in 4 cylinders made of two layers of sensors glued back-to-back with a stereo angle of 40 mrad . The layout has been designed so that 4 space points can be provided for energetic charged particles will pass through at least four layers everywhere in the acceptance region.

The single point resolution is $16\text{ }\mu\text{m}$ in the $R \cdot \phi$ and $580\text{ }\mu\text{m}$ in z direction.

- **Transition Radiation Tracker (TRT):** [7] [10] it is the outmost of the three tracking subsystems of the ATLAS Inner Detector. The TRT is a straw-tube tracker. When a charged particle traverses the TRT, it ionises the gas inside the straws. The resulting free electrons drift towards the wire, where they are amplified and read out.

The spaces between the straws are filled with polymer fibres (barrel) and foils (endcaps) to create transition radiation, which may be emitted by highly relativistic charged particles as they traverse a material boundary. This effect depends on the relativistic factor $\gamma = E/m$ and is strongest for electrons, so it could be a discriminating factor. This design makes the TRT complementary to the silicon-based tracking devices: the intrinsic single-point resolution of $130\text{ }\mu\text{m}$ is larger than that of the silicon trackers, but this is compensated by the large number of hits per track (typically more than 30).

The overall momentum resolution is:

$$\frac{\sigma_{p_T}}{p_T} = 0.05\%p_T \oplus 1\%$$

1.2.4 Calorimetry system

[12]Calorimeters measure the energy a particle loses as it passes through the detector, so the energy of all charged and neutral particles. It is usually designed to stop or “absorb” the particles coming from a collision, forcing them to deposit all of their energy within the detector. Calorimeters typically consist of layers of “passive” or “absorbing” high-density material interleaved with layers of an “active” medium such as solid lead-glass or liquid argon. Calorimeters are designed to stop most known particles except muons and neutrinos.

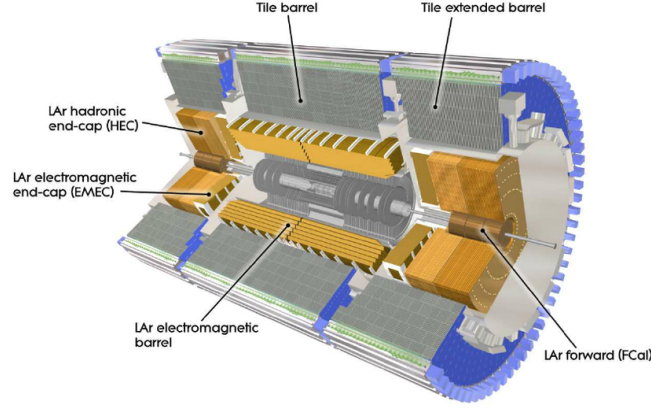


Figure 1.10: ATLAS calorimetry system

The components of the ATLAS calorimetry system are: [7] [11]

- **electromagnetic calorimetry:** the ATLAS EM calorimeter is a lead–liquid argon (LAr) sampling detector with accordion-shaped electrodes and lead absorber plates over its full coverage. The calorimeter is divided into a Barrel part and two End-Caps. Each End-Cap is divided into two coaxial wheels: an outer wheel and an inner wheel covering, respectively, $1.375 < |\eta| < 2.5$ and $2.5 < |\eta| < 3.2$. The absorber lead thickness is constant over large areas. The argon gap thickness is constant in the Barrel but changing with the radius in the End-Cap. In the range $|\eta| < 1.8$, the calorimeter is preceded by a presampler to recover the energy lost in the upstream material (cryostat, super-conducting coil, inner detector, etc.). The typical energy resolution is:

$$\frac{\sigma(E)}{E} = \frac{10\%}{\sqrt{E}} \oplus 0.7\%$$

Layer	Granularity ($\Delta\eta \times \Delta\phi$)
Presampler	0.025×0.1
Layer 1	0.0031×0.1
Layer 2	0.025×0.0245
Layer 3	0.05×0.245

Table 1.1: Granularity of different layers of the EM calorimeter

- **hadronic calorimeters:** in the range $|\eta| < 1.6$, the ATLAS hadronic calorimeter is an iron-scintillating tiles calorimeter. The Hadronic Tile calorimeter is located behind the solenoid coil and the EM calorimeter. It is a sampling calorimeter using iron as absorber material and scintillating tiles as active material. The Hadronic

Accurate knowledge of the field is required in order not to degrade the momentum resolution. During ATLAS running a large number of magnetic field sensors will measure the local field ensuring a knowledge of the bending power with a precision better than 0.3%.

Figure 1.12a shows the transverse view of the barrel part of the muon spectrometer. In the barrel a particle typically traverses three measurement planes. The inner most stations are situated directly after the hadronic calorimeter, just before the toroidal magnetic region. They are equipped with MDT chambers which allow a high precision measurement of the muon trajectory. The second layer of stations is situated in the magnet. The stations in this layer consist of a combination of one MDT chamber and two Resistive Plate Chambers (RPC). A third layer of stations is located just outside the magnetic field. The outer stations are formed by a combination of a MDT chamber and a RPC.

The design of the muon system in the end-caps is different as it is not possible to install stations inside the end-cap magnets and the background rate is much higher. In the end-cap, the first layer of stations sits in front of the magnet. The region closest to the beam pipe, where the background counting rates are highest, is equipped with Cathode Strip Chambers (CSC) instead of MDT chambers because of their higher rate capability. MDT chambers provide the remaining coverage. Thin Gap Chambers (TGC) are installed providing the trigger signal. The second station layer is installed behind the end-cap magnet and is equipped with one layer of MDT chambers and two layers of TGCs. The outer station layer is equipped with MDT chambers.

The typical energy resolution is resolution is 2-3% to 10 GeV and goes up to 10% to 1 TeV.

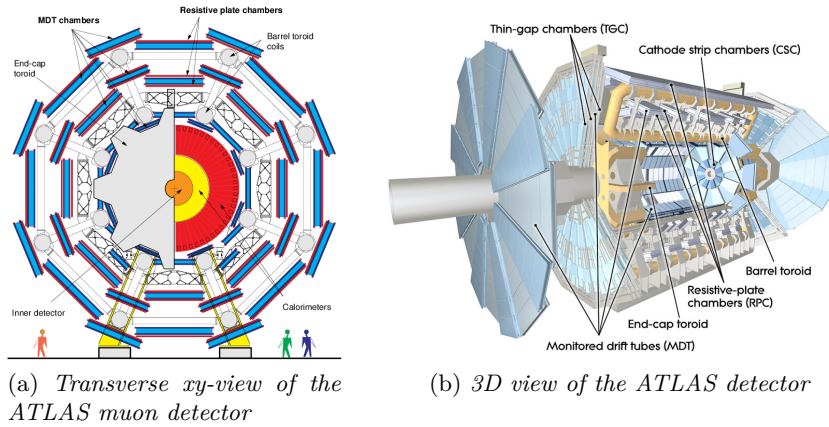


Figure 1.12: ATLAS muon detector

1.2.6 Trigger system

[14]ATLAS is designed to observe up to 1.7 billion proton-proton collisions per second, with a combined data volume of more than 60 million megabytes per second. However,

only some of these events will contain interesting characteristics that might lead to new discoveries. To reduce the flow of data to manageable levels, ATLAS uses a specialised two-level online event selection system - the Trigger System - which selects events with distinguishing characteristics that make them interesting for physics analyses. The Trigger System selects approximately 1000 of the 1.7 billion collisions that occur each second in the centre of the ATLAS detector.

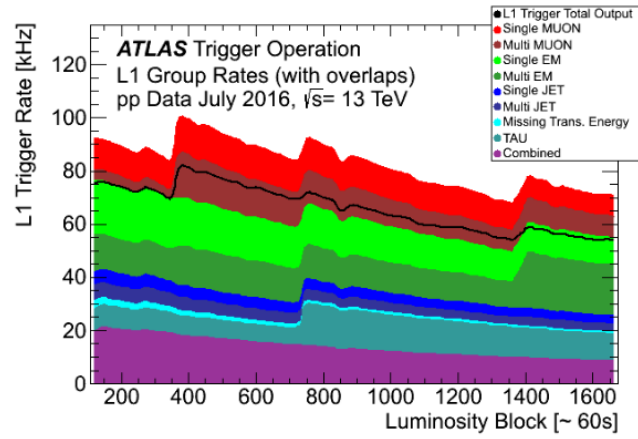
The two independent levels (Figure 1.13) are, a hardware-based first level (L1) and a software-based high level trigger (HLT) [15].

- The L1 trigger is implemented in fast custom-made electronics and runs with a fixed latency of $2.5\mu\text{s}$. L1 reduces the event rate from the LHC interaction rate of 40 MHz to ≈ 100 kHz. Up to 512 decision items are built, based on Regions of Interest (RoI) in η/φ retrieved from the muon (L1Muon) and calorimeter (L1Calo) systems. The L1 trigger decision is formed by the Central Trigger Processor (CTP). The L1Topo trigger, which is a system introduced for Run-2, performs selections based on geometric or kinematic association between trigger objects received from L1Calo or L1Muon.
- In the HLT, offline-like reconstruction algorithms run in a large farm of $\approx 40,000$ processor cores and a decision is formed typically within 300 ms. The HLT is a software trigger providing typically 2500 independent trigger chains. These are sequences of offline-like algorithms executed within the L1 RoIs. Furthermore, full-event reconstruction is possible at the HLT. Events accepted by the HLT are written into different data streams to be used for physics analysis, trigger level analysis, monitoring or detector calibration. Depending on the datastream, the full event or only partial event information is written out, allowing for higher rates without consuming a significant amount of the available bandwidth.

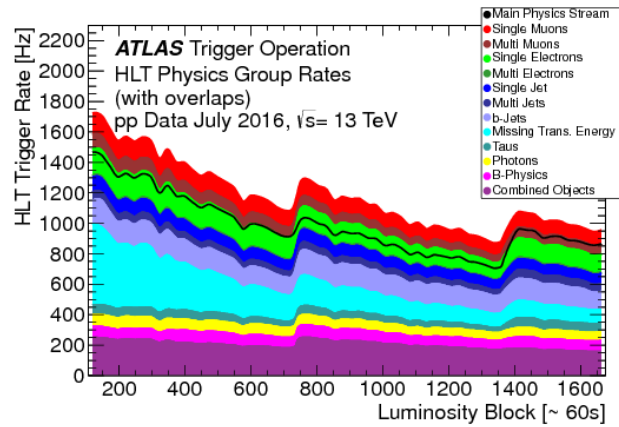
1.3 ATLAS Physics Programme

[16]ATLAS explores a range of physics topics, with the primary focus of improving our understanding of the fundamental constituents of matter and their interactions. Currently this is well described by the so called Standard Model (SM), a quantum field theory based on $SU(2) \otimes U(1) \otimes SU(3)$. ATLAS is studying the processes predicted by the SM such as w, z and top production and compares the measured cross section with the model predictions. ATLAS is also useful in the study of forces that govern their interactions. Indeed, the Standard Model also describes the fundamental forces of Nature and how they act between fundamental particles. In addition ATLAS is looking for new physics signatures like the ones proposed in Supersymmetry models, where forces combine with very high energies.

Due to the Proton-proton and heavy-ion collisions, LHC recreate the conditions immediately following the Big Bang when the Universe was governed by high-energy particle physics and later by a primordial soup of quarks and gluons, and allow ATLAS to study fundamental questions such as the Brout-Englert-Higgs field or Dark Matter.



(a)



(b)

Figure 1.13: L1 (a) and HLT (b) physics trigger rates grouped by trigger signatures as a function of the luminosity block number, in a fill taken in July 2016 with a peak instantaneous luminosity of $1.2 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$

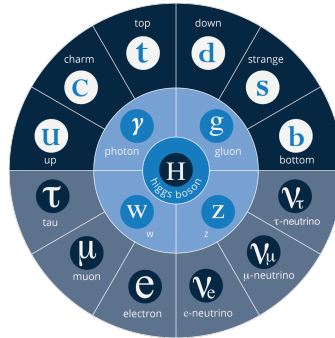


Figure 1.14: Particles in the Standard Model

One of the most important ATLAS discoveries is the proof of the existence of the Higgs boson. On 4 July 2012, the ATLAS and CMS [17] experiments at CERN announced that they had independently observed a new particle in the mass region of around 126 GeV: a boson consistent with the Higgs boson [18].

Chapter 2

Electron and photon reconstruction

Events with electrons and photons in the final state are important signatures for many physics analyses envisaged at the LHC. Their reconstruction mainly exploits data coming from the electromagnetic calorimeter (clusters) and the Inner Detector (ID) systems (tracks) [20]:

- an electron is defined as an object consisting of a cluster built from energy deposits in the calorimeter (supercluster) and a matched track (or tracks);
- a converted photon is a cluster matched to a conversion vertex (or vertices), and an unconverted photon is a cluster matched to neither an electron track nor a conversion vertex. About 20% of photons at low $|\eta|$ convert in the ID, and up to about 65% convert at $|\eta| \simeq 2.3$.

The reconstruction of electrons and photons is based on the following algorithm (Figure 2.1). It first prepares the tracks and clusters it will use. It selects clusters of energy deposits measured in topologically connected EM and hadronic calorimeter cells, denoted as topo-clusters (Section 2.1). These clusters are matched to ID tracks, which are re-fitted accounting for bremsstrahlung (Section 2.2). The algorithm also builds conversion vertices and matches them to the selected topo-clusters (Section 2.3). The electron and photon supercluster-building steps then run separately using the matched clusters as input (Section 2.4). After applying initial position corrections and energy calibrations to the resulting superclusters, the supercluster-building algorithm matches tracks to the electron superclusters and conversion vertices to the photon superclusters. The electron and photon objects to be used for analyses are then built (Section 2.5), their energies are calibrated (Section 2.6), and discriminating variables used to separate electrons or photons from background are added (Sections 2.7 and 2.8).

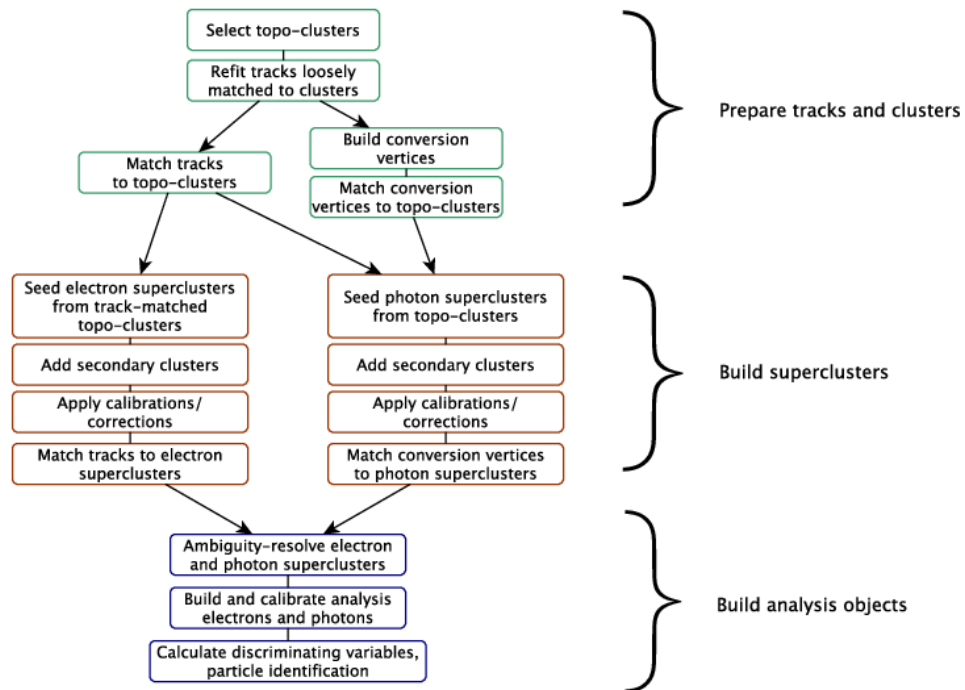


Figure 2.1: Algorithm flow diagram for the electron and photon reconstruction.

2.1 The topo-cluster reconstruction

The topo-cluster reconstruction algorithm begins by forming proto-clusters in the EM and hadronic calorimeters using a set of noise thresholds in which the cell initiating the cluster is required to have significance $|\zeta_{cell}^{EM}| \geq 4$ where

$$\zeta_{cell}^{EM} = \frac{E_{cell}^{EM}}{\sigma_{cell}^{EM}}$$

E_{cell}^{EM} is the energy cell at the EM scale and σ_{cell}^{EM} is the expected cell noise, which includes the known electronic noise and an estimate of the pile-up noise corresponding to the average instantaneous luminosity expected for Run 2. In order to suppress the formation of noise cluster, in this initial stage, cells from the presampler and the first LAr EM calorimeter layer are excluded from initiating proto-clusters. An important role is played by the neighbouring cells. If they have a significance $|\zeta_{cell}^{EM}| \geq 2$, these cells are collected by the proto-cluster. Each neighbour cell passing the threshold of $|\zeta_{cell}^{EM}| \geq 2$ becomes a seed cell in the next iteration, collecting each of its neighbours in the proto-cluster. If two proto-clusters contain the same cell with $|\zeta_{cell}^{EM}| \geq 2$ above the noise threshold, these proto-clusters are merged. A crown of nearest-neighbour cells is added to the cluster independently on their energy ($|\zeta_{cell}^{EM}| \geq 0$). This set of thresholds is commonly known as ‘4-2-0’ topo-cluster reconstruction.

In order to avoid a topo-cluster to collect energies from different physical objects a dedicated splitting procedure is defined. A cell with $|E_{cell}^{EM}| > 500$ MeV, at least four neighbours, and when none of the neighbours has a larger signal, is defined as a local maximum. Proto-clusters with two or more local maxima are split into separate clusters.

2.2 Track reconstruction

Track finding is one of the most challenging tasks in reconstructing events from proton-proton collisions recorded by the ATLAS detector. The process consists in finding a track from hits recorded in the ID. The ID track reconstruction consists of several sequences with different strategies and the main sequence is referred to as inside-out track finding: [23]

- **Space point formation:** the initial step of the ID reconstruction consists of the cluster and drift circle creation and the transformation of clusters in the silicon detectors into 3D space points. Clusters are formed by finding connected cells in the pixel and strip detectors. From these clusters, three-dimensional measurements referred to as space-points are created [22]. In the pixel detector, each cluster corresponds to one space-point, while in the SCT, clusters from both stereo views of a strip layer must be combined to obtain a three-dimensional measurement.
- **Space point seeded track finding:** track seeds are formed from sets of three space-points in the silicon-detector layers. Seeds can be built from space points in

the pixel detector only (referred to as PPP seeds), the strip detector only (SSS) or any mixed setup (PSS,PPS) [23]. To reduce the number of potential seeds, initial selections are applied and dedicated care is taken not to extensively use space points in multiple seeds. Seeds that pass the initial requirements are then input to a track finding algorithm that uses a combinatorial Kalman filter technique [28] and aims to complete the track candidates within the silicon detector.

- **Ambiguity solving:** track candidates are then further processed in an ambiguity solving module that aims to eliminate track candidates from random hit combinations (often referred to as "fakes") or track duplicates, which can be identified by measurements that are shared with other track candidates. The ambiguity solving relies on a scoring function applying positive scores for unique measurements and good fit quality, while penalising missing measurements where they would be expected (also called holes) or shared measurements with other track candidates.
- **TRT extension:** tracks that successfully pass the ambiguity solving stage and are within the coverage of the TRT detector are then extended into the TRT and completed for measurements in the outermost tracking detector. A successful TRT extension increases the momentum resolution significantly by exploiting the longer lever arm for field integration.

2.3 Track-cluster matching and photon conversion reconstruction

After cluster and track reconstruction, the next step is the track-cluster matching [20]. Fixed-size clusters in the calorimeter are used to create regions-of-interest (ROIs) and tracks intersecting these regions are considered loosely matched to the cluster. Track candidates are then fitted with the global χ^2 fitter [29]. The loosely matched, re-fitted tracks are then matched to the EM topo-clusters (with requirements, see [20]). If multiple tracks are matched to a cluster, they are ranked in order to choose the best one. To assign the rank, tracks are compared on the number of hits in the Pixel and SCT detectors. Another rank characteristic is ΔR : tracks with a better ΔR match between the track and cluster barycenter in the second layer coordinates are preferred. So the highest-ranked track is used to define the reconstructed electron properties.

Tracks loosely matched to fixed-size clusters serve as input to the reconstruction of the conversion vertex. Both tracks with silicon hits and tracks reconstructed only in the TRT are used for the conversion reconstruction. Two-track conversion vertices are reconstructed from two opposite-charge tracks forming a vertex consistent with that of a massless particle, while single-track vertices are essentially tracks without hits in the innermost sensitive layers. To increase the converted-photon purity, the tracks used to build conversion vertices must have a high probability to be electron tracks as determined by the TRT.

The conversion vertices are then matched to the EM topo-clusters. Using tracks with silicon hits, a conversion vertex is considered matched if extrapolated tracks match the

cluster to within $|\Delta\eta|$ and $|\Delta\phi| < 0.05$. With TRT only tracks these requirements are looser [24]. If there are multiple conversion vertices matched to a cluster, double-track conversions with two silicon tracks are preferred over other double-track conversions, followed by single-track conversions. Within each category, the vertex with the smallest conversion radius is preferred.

2.4 Supercluster reconstruction

The reconstruction of electron and photon superclusters proceeds independently, each in two stages [20]:

1. in the first stage, EM topo-clusters are tested for use as seed cluster candidates, which form the basis of superclusters;
2. in the second stage, EM topo-clusters near the seed candidates are identified as satellite cluster candidates, which may emerge from bremsstrahlung radiation or topo-cluster splitting.

Superclusters are built through the following steps:

- the initial list of EM topo-clusters is sorted according to descending E_T , calculated using the EM energy.
- the clusters are tested one by one in the sort order for use as seed clusters. There are two seed's kind:
 - i. electron supercluster seed: a cluster with a minimum E_T of 1 GeV and matched to a track with at least four hits in the silicon tracking detectors.
 - ii. photon supercluster seed: a cluster with E_T greater than 1.5 GeV with no requirement made on any track or conversion vertex matching.

A cluster cannot be used as a seed cluster if it has already been added as a satellite cluster to another seed cluster.

- if a cluster meets the characteristics of the previous point, the algorithm attempts to find satellite clusters, through the process summarized in figure 2.2. For both electrons and photons, a cluster is considered a satellite if it falls within a window of $\Delta\eta \times \Delta\phi = 0.075 \times 0.125$ around the seed cluster barycentre, as these cases tend to represent secondary EM showers originating from the same initial electron or photon. For electrons, this window could be larger, $\Delta\eta \times \Delta\phi = 0.125 \times 0.300$, and its 'best-matched' track is also the best-matched track for the seed cluster. For photons with conversion vertices made up only of tracks containing silicon hits, a cluster is added as a satellite if its best-matched (electron) track belongs to the conversion vertex matched to the seed cluster. These steps rely on tracking information to discriminate distant radiative photons or conversion electrons from pile-up noise or other unrelated clusters.

The seed clusters with their associated satellite clusters are called superclusters.

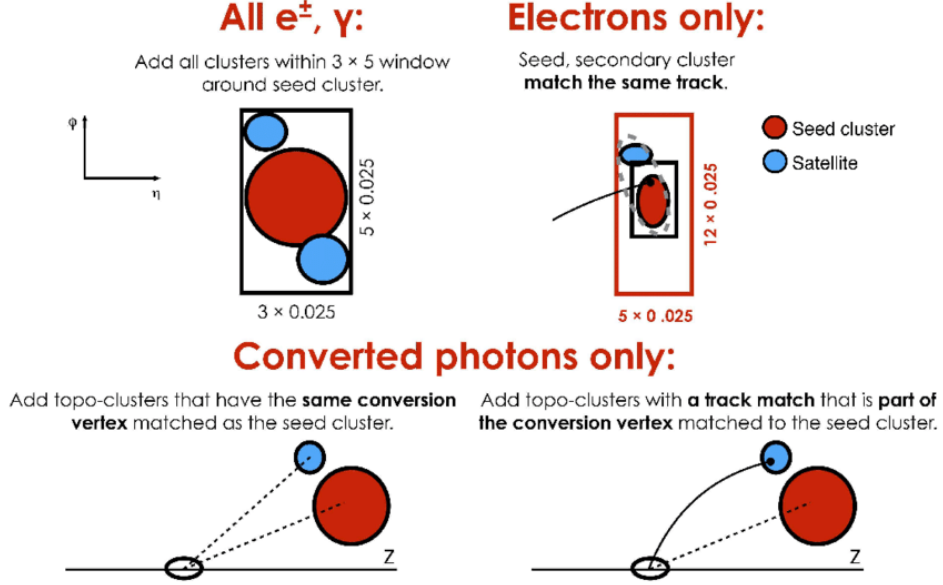


Figure 2.2: Diagram of the superclustering algorithm for electrons and photons. Seed clusters are shown in red, satellite clusters in blue.

- The final step in the supercluster-building algorithm is to assign calorimeter cells to a given supercluster. Only cells from the presampler and the first three LAr calorimeter layers are considered, except in the transition region of $1.4 < |\eta| < 1.6$, where the energy measured in the scintillator between the calorimeter cryostats is also added. To limit the superclusters' sensitivity to pile-up noise, the size of each constituent topo-cluster is restricted to a maximal width of 0.075 or 0.125 in the η direction in the barrel or endcap region, respectively. Because the magnetic field in the ID is parallel to the beam-line, interactions between the electron or photon and detector material generally cause the EM shower to spread in the φ direction, so the restriction in η still generally allows the electron or photon energy to be captured. No restriction is applied in the φ -direction.

2.5 Creation of electrons and photons for analysis

After the electron and photon superclusters are built, tracks are matched to electron superclusters and conversion vertices to photon superclusters [20]. Then the analysis-level electrons and photons are created. Because electron and photon superclusters are built independently, a given seed cluster can produce both an electron and a photon. In such cases, the procedure presented in figure 2.3 is applied. The purpose is that if a particular object can be easily identified only as a photon (a cluster with no good track attached) or only as an electron (a cluster with a good track attached and no good photon conversion vertex), then only a photon or an electron object is created for

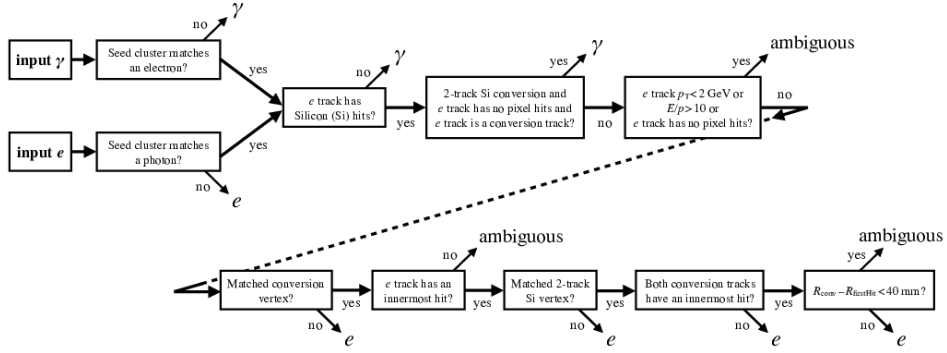


Figure 2.3: Flowchart showing the logic of the ambiguity resolution for particles initially reconstructed both as electrons and photons.

analysis; otherwise, both an electron and a photon object are created. Furthermore, these cases are marked explicitly as ambiguous, allowing the final classification of these objects to be determined based upon the specific requirements of each analysis.

2.6 Calibration

The energy calibration covers the region $|\eta| < 2.47$, which corresponds to the acceptance of the ID and the highly segmented EM calorimeter. The procedure to calibrate the energy response of electrons and photons from the energy of a cluster of cells in the EM calorimeter is composed of various steps:

- a multivariate regression algorithm, trained on samples of simulated events, is used for the estimation of the energy of the electron or photon from the energy deposits in the calorimeter: the properties of the shower development are used to optimize the energy resolution and to minimize the impact of material in front of the calorimeter;
- in order to have the correct extrapolation of the energy calibration to the full energy range of electrons and photons, the adjustment of the relative energy scales of the different layers of the EM calorimeter is necessary. It is based on studies of

muon energy deposits and electron showers and it is applied as a correction to the data before the estimation of the energy of the electron or photon;

- the next step is the correction for residual local non-uniformities in the calorimeter response affecting the data;
- the adjustment of the overall energy scale in the data: this is done using a large sample of Z boson decays to electron-positron pairs. At the same time, a correction to account for the difference in energy resolution between data and simulation is derived, and applied to the simulation. These correction factors are assumed to be universal for electrons and photons;
- the last step is to check the results comparing data and simulation with independent samples: $J/\psi \rightarrow ee$ decays probe the energy response for low-energy electrons. Radiative Z boson decays are used to check the energy response for photons.

2.7 Identification

Excellent electron and photon identification capabilities [25][26] are crucial for many aspects of the ATLAS physics program, from Standard Model measurements (including Higgs boson) to new physics searches. The identification of prompt photons and the rejection of backgrounds, mostly coming from photons from hadron decays, relies on the high granularity of the ATLAS calorimeter. Electron identification is based on a likelihood (LH) discrimination to separate isolated electron candidates from candidates originating from photon conversions, hadron misidentification and heavy flavor decays.

2.7.1 Electron identification

The quantities used in the electron identification [20] are chosen according to their ability to discriminate prompt isolated electrons from energy deposits of hadronic jets, from converted photons and from genuine electrons produced in the decays of heavy-flavour hadrons. The variables can be grouped into properties, reported in Table 2.1:

- the primary electron track, which is required to fulfil a set of quality requirements, namely hits in the two inner tracking layers closest to the beam line, as well as a number of hits in the silicon-strip detectors. The transverse impact parameter of the track and its significance are used to construct the likelihood discriminant;
- the lateral development of the electromagnetic shower, which is characterized by variables calculated separately in the first and second layer of the electromagnetic calorimeter. To reject clusters from multiple incident particles, w_{tot} is used. The lateral shower development is measured with R_φ and R_η ;
- the longitudinal development of the electromagnetic shower: the electron identification uses f_1 and f_3 , which represent the ratio of the energy in the strip or back layer to the total energy in the EM calorimeter, respectively;

- the spatial compatibility of the primary electron track with the reconstructed cluster, which are matched using $\Delta\eta_1$ and $\Delta\phi_{res}$.

Then, the combination of information from the tracker and the matching, information from the electromagnetic calorimeter and hadronic leakage are put together in likelihoods for a reconstructed electron to originate from signal, L_S , or background, L_B . They are calculated from probability density functions (pdfs), P :

$$L_{S/B}(\mathbf{x}) = \prod_{i=1}^n P_{S/B,i}(x_i).$$

For signal and background the pdfs take the values $P_{S,i}(x_i)$ and $P_{B,i}(x_i)$, respectively, for the quantity i at value x_i . The likelihood discriminant d_L is defined as the natural logarithm of the ratio of L_S and L_B :

$$d_L = \ln \frac{L_S}{L_B}$$

and each electron is assigned a specific value.

Using the same variables but different selections on the d_L value, three different operating points are defined: Loose, Medium and Tight. They are sorted with increasing threshold values, and are chosen in order to have efficiency for electrons with $E_T > 40$ GeV of 93%, 88%, and 80% respectively.

2.7.2 Photon identification

The photon identification criteria [20] are designed to efficiently select prompt, isolated photons and reject backgrounds from hadronic jets. The photon identification is constructed from one-dimensional selection criteria, or a cut-based selection, using the shower shape variables, listed in Table 2.2.

The variables using the EM first layer play a particularly important role in rejecting π^0 decays into two highly collimated photons. The primary identification selection is labelled as Tight, with less restrictive selections called Medium and Loose, which use the R_{had} , R_{had1} , R_η , and w_{η_2} shower shape variables. The Medium selection adds a loose cut on E_{ratio} . Because the reconstruction of photons in the ATLAS trigger system does not differentiate between converted and unconverted photons, the Loose and Medium identification criteria, which are common between trigger and offline, are the same for converted and unconverted photons. The Tight identification criteria are designed to select a subset of the photon candidates passing the Medium criteria.

Because the shower shapes vary due to the geometry of the calorimeter, the cut-based selection of Loose, Medium and Tight are optimized separately in bins of $|\eta|$. The Tight identification is also optimized in separate bins of E_T . The Tight identification is performed separately for converted and unconverted photons. In fact, the shower shapes

Type	Description	Simbol
Hadronic leakage	- Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM calorimeter (in the pseudorapidity range $ \eta < 0.8$ or $ \eta > 1.37$)	R_{had1}
	- Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM calorimeter (in the pseudorapidity range $0.8 < \eta < 1.37$)	R_{had}
Layer 3 of the EM calorimeter	Ratio of the energy in the back layer to the total energy in the EM calorimeter. This variable is only used below 100 GeV because it is known to be inefficient at high energies.	f_3
Layer 2 of the EM calorimeter	- $\frac{\text{Lateral shower width}}{\sqrt{(\sum_i E_i \eta_i^2)/(\sum_i E_i) - ((\sum_i E_i \eta_i)/(\sum_i E_i))^2}}$, where E_i is the EM calorimeter energy and η_i is the pseudorapidity of cell i and the sum is calculated within a window of 3×5 cells	w_n^2
	- Ratio of the energy in 3×3 cells over the energy in 3×7 cells centered at the electron cluster position	R_ϕ
	- Ratio of the energy in 3×7 cells over the energy in 7×7 cells centered at the electron cluster position	R_η
Layer 1 of the EM calorimeter	- Shower width, $\sqrt{(\sum_i E_i (i - i_{max})^2)/(\sum_i E_i)}$ where i runs over all strips in a window of $\Delta\eta \times \Delta\varphi \sim 0.0625 \times 0.2$, corresponding typically to 20 strips in η , and i_{max} is the index of the highest-energy strip	w_{stot}
	- Ratio of the energy difference between the largest and second largest energy E ratio deposits in the cluster over the sum of these energies	E_{ratio}
	- Ratio of the energy in the strip layer to the total energy in the EM accordion calorimeter	f_1
Track conditions	- Number of hits in the innermost pixel layer; discriminates against photon conversions	n_{Blayer}
	- Number of hits in the Pixel Detector	n_{Pixel}
	- Number of total hits in the pixel and SCT detectors	n_{Si}
	- Transverse impact parameter with respect to the beam-line	d_0
	- Significance of transverse impact parameter defined as the ratio of d_0 and its uncertainty	d_0/σ_{d_0}
	- Momentum lost by the track between the perigee and the last measurement point divided by the original momentum	$\Delta p/p$
TRT	Likelihood probability based on transition radiation in the TRT	eProbabilityHT
Track-cluster matching	- $\Delta\eta$ between the cluster position in Layer 1 and the extrapolated track	$\Delta\eta_1$
	- $\Delta\Phi$ between the cluster position in the Layer 2 and the track extrapolated from the perigee	$\Delta\Phi_2$
	- Defined as $\Delta\Phi_2$, but the track momentum is rescaled to the cluster energy before extrapolating the track from the perigee to the Layer 2 of the calorimeter.	$\Delta\phi_{res}$
	- Ratio of the cluster energy to the track momentum	E/p

Table 2.1: Discriminating variables used for electron identification [25]

of converted photons differ from unconverted photons due to the opening angle of the e^+e^- conversion pair, which is amplified by the magnetic field, and from the additional interaction of the conversion pair with the material upstream of the calorimeters.

Category	Description	Name	<i>loose</i>	<i>tight</i>
Acceptance	$\eta < 2.37$, with $1.37 < \eta < 1.52$ excluded	-	*	*
Hadronic leakage	- Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM calorimeter (used over range $ \eta < 0.8$ or $ \eta > 1.37$)	R_{had1}	*	*
	- Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM calorimeter (in the pseudo-rapidity range $0.8 < \eta < 1.37$)	R_{had}	*	*
EM Middle layer	- Ratio of $3 \times 7 \eta \times \phi$ to 7×7 cell energies	R_η	*	*
	- Lateral width of the shower	$w_{\eta 2}$	*	*
	- Ratio of $3 \times 3 \eta \times \phi$ to 7×7 cell energies	R_ϕ		*
EM Strip layer	- Shower width calculated from three strips around the strip with maximum energy deposit	w_{s3}		*
	- Total lateral shower width	w_{stot}		*
	- Energy outside the core of the three central strips but within seven strips divided by energy within the three central strips	F_{side}		*
	- Difference between the energy associated with the second maximum in strip layer(L1) and the energy reconstructed in the strip with the minimum value found between the first and second maxima	ΔE		*
	- Ratio of the energy difference associated with the largest and second largest energy deposits to sum of these energies	E_{ratio}		*

Table 2.2: Discriminating variables used for *loose* and *tight* photon identification [25]

2.8 Electron and photon isolation

The isolation selection, or isolation cut, is applied in order to reduce the backgrounds. Different criteria are defined, with different levels of efficiency in rejecting the background. The isolation criterium uses variables from the tracking and from the calorimeter. [27]

The activity near leptons and photons can be quantified from the tracks of nearby charged particles, or from energy deposits in the calorimeters, leading to two classes of isolation variables [20]:

- The raw calorimeter isolation ($E_{T,raw}^{isol}$) is built by summing the transverse energy of positive-energy topological clusters (at EM scale) whose barycentre falls within a cone centred around the electron or photon cluster barycentre. The raw calorimeter isolation includes the EM particle energy ($E_{T,core}$), which is subtracted by removing the energy of the EM calorimeter cells contained in a $\Delta\eta \times \Delta\varphi = 5 \times 7$ (in EM-middle-layer units) rectangular cluster around the barycentre of the EM particle cluster. Other corrections are done to account for the pile-up contribution.

The fully corrected calorimeter isolation variable is finally computed as:

$$E_T^{coneXX} = E_{T,raw}^{isolXX} - E_T^{core} - E_{T,leakage}(E_T, \eta, \Delta R) - E_{T,pile-up}(\eta, \Delta R)$$

where XX refers to the size of the employed cone (a cone size of $\Delta R = 0.2$ is used for the electron or photon working point, whereas cone size of $\Delta R = 0.4$ is only used for photon working points) and the $E_{T,leakage}$ term refers to a correction for the e/γ candidate energy leaking in the isolation cone.

- The track isolation variable (p_T^{coneXX}) is computed by summing the transverse momentum of selected tracks within a cone centred around the electron track or the photon cluster direction (excluding tracks matched to the electron or converted photon)[27]. Since for electrons produced in the decay of high-momentum heavy particles can be very close to the electron direction, the track isolation for electrons is defined with a variable cone size:

$$\Delta R = \min\left(\frac{10}{p_T[GeV]}, \Delta R_{max}\right)$$

where ΔR_{max} is the maximum cone size (typically 0.2).

The tracks considered are required to have $p_T > 1$ GeV and $|\eta| < 2.5$, at least seven silicon (Pixel + SCT) hits, at most one shared hit, at most two silicon holes and at most one pixel hole. In addition, for electron isolation, the tracks are required to have a loose vertex association, i.e. the track was used in the primary vertex fit, or it was not used in any vertex fit but satisfies $|\Delta z_0| \sin\theta < 3$ mm, where $|\Delta z_0|$ is the longitudinal impact parameter relative to the chosen primary vertex; for photon isolation, all selected tracks satisfying $|\Delta z_0| \sin\theta < 3$ mm are used.

2.8.1 Electron isolation criteria and efficiency measurements

The implementation of isolation criteria is specific to the physics analysis needs, as it results from a compromise between a highly-efficient identification of prompt electrons, isolated or produced in a busy environment, and a good rejection of electrons from heavy-flavour decays or light hadrons misidentified as electrons. The different electron-isolation working points used in ATLAS are presented in Table 2.3. The working points can be defined in two different ways, targeting a fixed value of efficiency or with fixed cuts on the isolation variables. The Gradient working point is designed to give an efficiency of 90% at $p_T = 25$ GeV and 99% at $p_T = 60$ GeV, uniform in η .

Working point	Calorimeter isolation	Track isolation
Gradient	$\epsilon = 0.1143 \times p_T + 92.14\%(with E_T^{cone20})$	$\epsilon = 0.1143 \times p_T + 92.14\%(with p_T^{varcone20})$
HighPtCaloOnly	$E_T^{cone20} < \max(0.015 \times p_T, 3.5 \text{ GeV})$	
Loose	$E_T^{cone20}/p_t < 0.20$	$p_T^{varcone20}/p_t < 0.15$
Tight	$E_T^{cone20}/p_t < 0.06$	$p_T^{varcone20}/p_t < 0.06$

Table 2.3: Definition of the electron isolation working points and isolation efficiency ϵ .

2.8.2 Photon isolation criteria and efficiency measurements

Three photon isolation operating points are defined using requirements on the calorimeter and track isolation variables, as summarized in Table 2.4. The photon isolation efficiency is studied in two main signatures: radiative Z decays (valid for $10 < E_T < 100$ GeV) and inclusive photons (used in the $25 \text{ GeV} < E_T < \sim 1.5 \text{ TeV}$ range).

Working point	Calorimeter isolation	Track isolation
Loose	$E_T^{cone20} < 0.065 \times E_T$	$p_T^{cone20}/E_t < 0.05$
Tight	$E_T^{cone40} < 0.022 \times E_T + 2.45 \text{ GeV}$	$p_T^{cone20}/p_t < 0.06$
TightCaloOnly	$E_T^{cone40} < 0.022 \times E_T + 2.45 \text{ GeV}$	

Table 2.4: Definition of the photon isolation working points.

Chapter 3

Machine Learning and Gradient Boosted Trees

Machine Learning (ML) is an application of artificial intelligence (AI) [30] with the aim of generating systems with the ability to automatically learn and improve from experience without being explicitly programmed. The ML algorithm is exposed to a training dataset in order to generate a mathematical model, which is then applied in real-life performance tasks.

ML can be divided into three main categories [31]:

1. **Supervised Learning:** the goal is to learn a mapping from inputs \mathbf{x} to outputs y , given a labeled set of input-output pairs $D = \{(x_i, y_i)\}_{i=1}^N$. Here D is called the training set, and N is the number of training examples. In the simplest setting, each training input \mathbf{x}_i is a K -dimensional vector of numbers, which are called features, attributes or covariates. In general, however, \mathbf{x}_i could be a complex structured object.

The form of the output, or response variable, can in principle be anything, but most methods assume that y_i is a categorical or nominal variable from some finite set, $y_i \in \{1, \dots, C\}$ or that y_i is a real-valued scalar. When y_i is categorical, the problem is known as *classification* or pattern recognition, and when y_i is real-valued, the problem is known as *regression*.

2. **Unsupervised learning:** only an input dataset $D = \{(x_i)\}_{i=1}^N$ is given to the algorithm and the goal is to find “interesting patterns” in the data. This is sometimes called knowledge discovery.
3. **Reinforcement learning:** this is useful for learning how to act or behave by given occasional reward or punishment signals. A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty.

3.1 Supervised Learning

Since in this thesis a supervised learning algorithm [32] is used, the main features of these class of models will be briefly analysed in this section.

3.1.1 Model and Parameters

As written above, the model in supervised learning usually refers to the mathematical structure by which the prediction y_i is made from the input \mathbf{x}_i .

The parameters of the model are the undetermined part that the algorithm has to learn from training dataset. Usually $\boldsymbol{\theta}$ is used to denote the set of parameters of a model.

3.1.2 Objective Function

In order to train the model, we need to define a function of the parameters to measure how well the model describe the training data. This function is called *Objective function*.

A characteristic of objective functions is that they are composed of two parts (also dependent on parameters), training loss L and regularization term Ω :

$$obj(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \Omega(\boldsymbol{\theta}).$$

The training loss measures how predictive our model is with respect to the training data. Common choices of L are the mean squared error, which is given by:

$$L(\boldsymbol{\theta}) = \sum_i (y_i - y_{i_{pred}})^2$$

or logistic loss, to be used for logistic regression:

$$L(\boldsymbol{\theta}) = \sum_i [y_i \ln(1 + e^{-y_{i_{pred}}}) + (1 - y_i) \ln(1 + e^{-y_{i_{pred}}})].$$

The regularization term controls the complexity of the model. A commonly used regularization term is:

$$\Omega(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_1$$

where λ is a hyperparameter, which will be analysed in the Section 4.1.3 and $\|\cdot\|_1$ is the \mathcal{L}_1 norm. The regularization term helps to avoid overfitting, which happens when a model learns the detail and noise in the training dataset to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize. Figure 3.1 shows a visual example of overfitting. The general goal is to create a simple and predictive model. The tradeoff between the two is also referred as bias-variance tradeoff in machine learning.

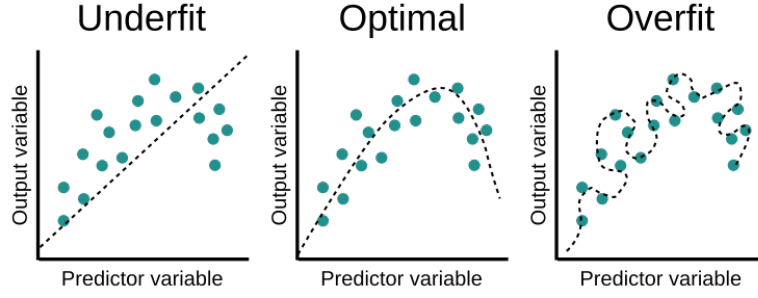


Figure 3.1: Underfit, optimal and overfit regression models

3.2 Gradient boosted trees

There are many model, which ML algorithms can use, and one of them is a Decision Tree ensemble. If it is trained, it becomes a Gradient Boosted Tree. The basic unit is decision tree.

3.2.1 Decision Tree

Decision Trees (DTs) [33] are supervised learning methods used for classification and regression. A decision tree is a hierarchical structure consisting of nodes and directed edges. The tree has three types of nodes:

- a root node that has no incoming edges and zero or more outgoing edges.
- internal nodes, which have exactly one incoming edge and two or more outgoing edges.
- leaf or terminal nodes, which have exactly one incoming edge and no outgoing edges.

Each internal node represents a test on a feature of the input variable. In this way the tree progressively classify the input sample into different categories. The terminal node contains a decision value.

3.2.2 Decision Tree Ensembles

The tree ensemble model consists of a set of classification and regression trees (CART) [32]. A CART is a bit different from decision trees, in which the leaf only contains decision values. In CART, a real score is associated with each of the leaves. This also allows for a principled, unified approach to optimization.

Usually, a single tree is not strong enough to precisely represent a complex model, so the ensemble model is used. In this model the prediction scores of each individual tree

are summed up to get the final score:

$$y_{i_{pred}} = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (3.1)$$

where K is the number of trees, f is a function in the functional space \mathcal{F} of all possible CARTs (considering a tree f as a function which maps the input features into a score). In this way the final decision is taken using the output of different decision trees to improve the prediction capabilities of the algorithm. The objective function to be optimized is given by:

$$obj(\theta) = \sum_i^n l(y_i, y_{i_{pred}}) + \sum_{k=1}^K \Omega(f_k) \quad (3.2)$$

so the functions f_k are the parameters of the optimisation.

3.2.3 Gradient Bosting Decision Trees

After introducing the model, we focus on trees training. In this case, as mentioned above, the parameters of optimisation are the function f_i , each containing the structure of the tree and the leaf scores. The training proceeds by optimising the objective function (eq. (3.2)), which will be optimize:

There are many training methods and one of them is *Additive Training*. Due to the difficulty of training many trees at the same time, an additive approach is used:

$$\begin{aligned} y_{i_{pred}}^{(0)} &= 0 \\ y_{i_{pred}}^{(1)} &= f_1(\mathbf{x}_i) = y_{i_{pred}}^{(0)} + f_1(\mathbf{x}_i) \\ y_{i_{pred}}^{(2)} &= f_1(\mathbf{x}_i) + f_2(\mathbf{x}_i) = y_{i_{pred}}^{(1)} + f_2(\mathbf{x}_i) \\ &\dots \\ y_{i_{pred}}^{(t)} &= \sum_{k=1}^t f_k(\mathbf{x}_i) = y_{i_{pred}}^{(t-1)} + f_t(\mathbf{x}_i) \end{aligned} \quad (3.3)$$

In each step, the tree used has to be the one that optimizes objective function. In the additive approach eq. (3.2) becomes:

$$obj^{(t)} = \sum_i^n l(y_i, y_{i_{pred}}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) + costant \quad (3.4)$$

Then using a Taylor expansion of the loss function up to the second order in (3.4):

$$obj^{(t)} = \sum_i^n l(y_i, y_{i_{pred}}^{(t-1)} + g_i f_t(\mathbf{x}_i)) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) + \Omega(f_t) + costant \quad (3.5)$$

where g_i and h_i are:

$$\begin{aligned} g_i &= \partial_{y_{i_{pred}}^{(t-1)}} l(y_i, y_{i_{pred}}^{(t-1)}) \\ h_i &= \partial_{y_{i_{pred}}^{(t-1)}}^2 l(y_i, y_{i_{pred}}^{(t-1)}) \end{aligned} \quad (3.6)$$

So the specific objective at step t , without all the constants, becomes:

$$\sum_i^n [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t) \quad (3.7)$$

The other part of objective is the regulation term, which represents the complexity of the tree f_t . In a more general formalism f_t can be defined as:

$$f_t(\mathbf{x}) = w_{q(\mathbf{x})}, \quad w \in R^T, \quad q : R^d \rightarrow \{1, 2, \dots, T\} \quad (3.8)$$

where w is the vector of scores in leaves, q is a function assigning each data point to the corresponding leaf (so representing the tree structure), and T is the number of leaves. So the complexity Ω can be written as:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (3.9)$$

So, using eq. (3.5), (3.8), (3.9), the objective value with the t -th tree becomes:

$$\begin{aligned} obj^{(t)} &\simeq \sum_{i=1}^n [g_i w_{q(\mathbf{x}_i)} + \frac{1}{2} h_i w_{q(\mathbf{x}_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \quad (3.10)$$

where $I_j = \{i | q(\mathbf{x}_i) = j\}$ is the set of indices of data points assigned to the j -th leaf. The equation (3.10) can be simplified by defining $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$:

$$obj^{(t)} = \sum_{j=1}^T [G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2] + \gamma T \quad (3.11)$$

Since w_j are independent with respect to each other and the form $G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2$ is quadratic, the best w_j for a given structure $q(\mathbf{x})$ is:

$$w_j^{best} = -\frac{G_j}{H_j + \lambda} \quad (3.12)$$

Then using the best w_j , the objective reduction becomes the best objective reduction, which measures how good a tree structure $q(x)$ is:

$$obj^{(t)_{best}} = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (3.13)$$

3.2.4 Learn the tree structure

Now thanks to the equation (3.13) it is possible to measure how good a tree is, ideally it is possible to enumerate all possible trees and pick the best one. In practice this is intractable (there are infinite possible trees), so the tree to add will be optimized one level at a time: a split is added to the tree if it brings a gain to its object function.

When trying to split a leaf into two leaves the score it gains is:

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (3.14)$$

This formula can be decomposed as:

- the score on the new left leaf
- the score on the new right leaf
- the score on the original leaf
- regularization on the additional leaf

The best split for each node is obtained by scanning separately each feature (sorted by feature value) and taking the best split option along all the features. If the Gain is negative this means that the split has increased the complexity score more than the reduction in training loss. In this case it would be better not to add that branch. This is exactly the pruning techniques, which reduces the complexity by removing sections of the tree.

3.3 LightGBM

LightGBM [34] is a gradient boosting framework, released by Microsoft, that uses tree based learning algorithms. It is based on GDBT (Section 3.2.3). LightGBM grows trees leaf-wise (best-first), as shown in Fig.3.2. It will choose the leaf with max delta loss to grow. Holding "leaf" fixed, leaf-wise algorithms tend to achieve better performance (lower loss) than level-wise algorithms.

Leaf-wise may cause over-fitting when training dataset is small, so LightGBM includes the max_{depth} parameter to limit tree depth. However, trees still grow leaf-wise even when max_{depth} is specified. LGBM has the possibility to control the model and the training through specific parameters such as learning rate, feature fraction, number of leaves.

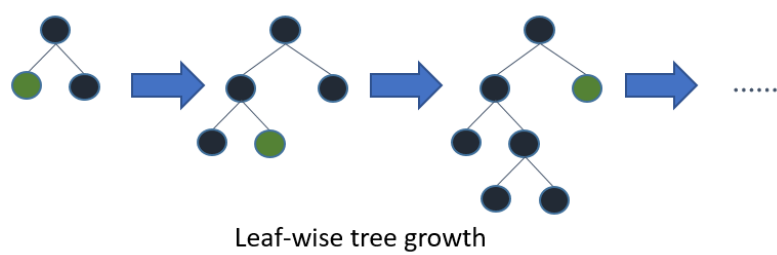


Figure 3.2: The LGBM leaf-wise tree growth

Chapter 4

BDT: Electron and photon classification

As said before, during the reconstruction not all the particles are easily reconstructed and therefore they are classified both as electrons and photons. These particles are the 10% of true electron and 30% of true photon.

The analysis of these "ambiguous" particles is the last step in the discrimination process between electrons and photons. This analysis can be done in several ways. One of them is a classification algorithm with a gradient boosted decision tree (*classification Supervised Learning*). In order to create a predictive model the GBDT is trained on single-particle samples of electrons and photons, with simulated pile-up. The general set is composed of single MC particle events, which have been reconstructed both as electron and photon (double reconstructions). each reconstructed event is assigned *truth* = 0 (or 1) in case the initial event was an electron (or a photon). Therefore the input particles are represented by y_i , which is 0 for true electron and 1 for true photon, and the score (model output) is $y_i^{pred} \in [0, 1]$.

After removing the pile-up using the Pandas [37] function *query* with the feature `ph(or el)_is_pileup==False`, the dataframe is splitted in the train (80%) and the test (20%) sets, using a Scikit Learn [35] function. The train dataset is used to train the algorithm, whereas the test dataset is used as model input. If we did not do this split, we would introduce a bias.

In order to explore as many cases as possible, three models have been created: all use the same discriminating variables, but they are trained on different portions of the training dataset.

1. a model generated by a GBDT trained on double reconstruction, which are the input objects of ambiguity tools;
2. a model generated by a GBDT trained on reconstruction flagged as "ambiguous" by the old ambiguity tool (Fig 2.3);
3. a model generated by a GBDT trained on reconstruction flagged as "ambiguous" by the new ambiguity tool.

Since at reconstruction level it is not possible to use a BDT and it is not possible to save the double reconstructions for each particle, due to the enormous memory cost it would involve, pre-classifiers (ambiguity tools Appendix C) are used (Fig 2.3). To see the performance of these pre-classifiers, the models above are compared. The first is the ideal model, which has been trained as if there were infinite memory, and therefore is the theoretical limit. Therefore, the more the models based on ambiguity tools approach the performance of the double reconstructions model, the better the applied preclassification is.

4.1 BDT training

4.1.1 Discriminating features

In order to create a good model, the choice of the features it will have to work on is crucial. The most useful characteristics are those that present the greatest differences between photons and electrons, which are therefore more discriminating. It is also important to use features that guarantee the generality of the model. The features can be divided into two categories: features that come from the particle being reconstructed as an electron or as a photon. Some features, as they are not specific to either of the two reconstructions, cannot be included in these two categories and they are:

- `el_cl_pt` is the cluster momentum p_T of the reconstructed electron (Figure 4.1).
- `el_cl_eta` is the cluster η of the reconstructed electron. Since photons convert more into the material we can see an increase of electrons reconstructed from real photons for $|\eta| > 1.52$ (Figure 4.2).

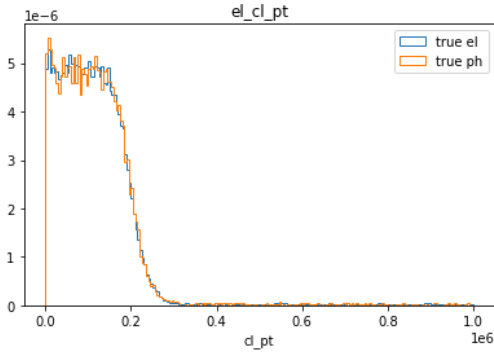


Figure 4.1: The `el_cl_pt` distribution

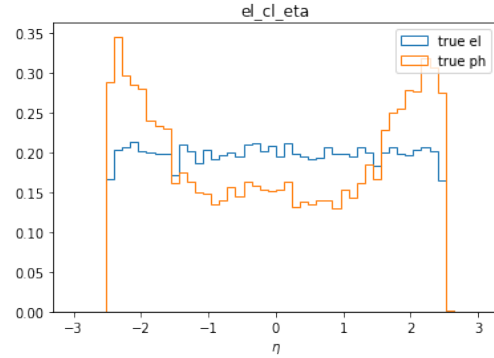


Figure 4.2: The `el_cl_eta` distribution

Electron features

Electron features are those that characterize the particles reconstructed as electrons. Electrons have many hits in the first layers of detectors, as opposed to photons, because the most part of photons is still unconverted in the ID. So electron reconstructed as electrons (`true_el/reco_el`) should have more hits than photons reconstructed as electrons (`true_ph/reco_el`). This can be seen in the characteristics `el_track_hasInnermostHits` (a boolean, which says if the particle has hit the first expected layer or not), `el_trkPixelHits` (a pixel hit counter), `el_trkSiHits` (a silicon hit counter) shown in the Figure 4.3.

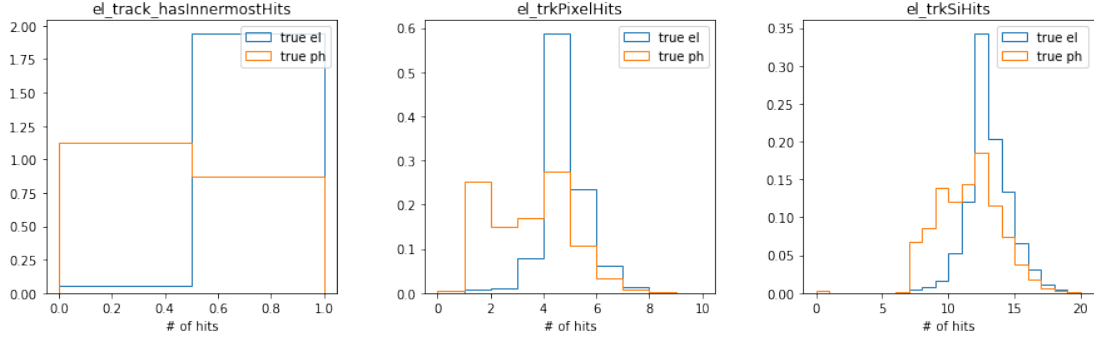


Figure 4.3: Three of the most discriminating features from electron reconstruction

The track quality, as it is determined from the pixel and SCT hits, is different in the two cases as the physics quantities reconstructed from the track. These quantities are:

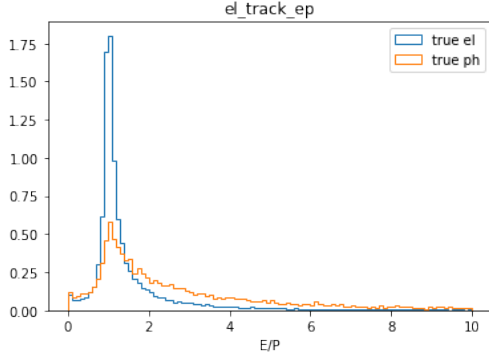
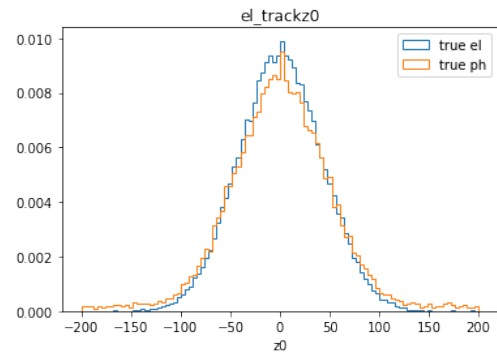
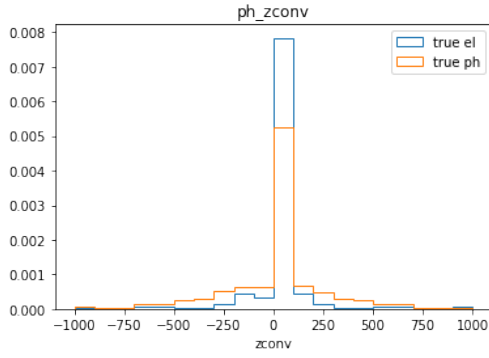
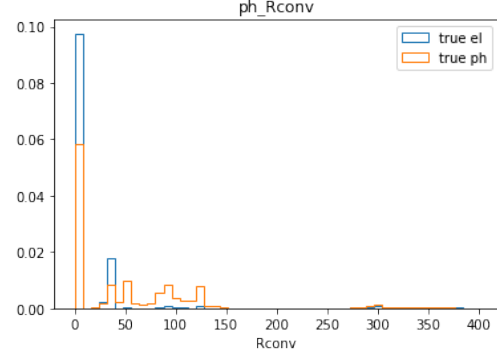
- `el_track_ep` is the ratio between the energy measured in the calorimeter and the moment p_T measured in the ID. Photons tracks are worse than those of electrons and therefore they are not always matched with the right energy. So the E/p_T ratio could be far from unity (Figure 4.4).
- `el_trackz0` is the z coordinate, which runs along the beam axis, where the measurement of the reconstructed electron takes place (Figure 4.5).

Photon features

As for electron, the converted photon quality track is discriminating and therefore also the Pixel/SCT hits and the p_t of the associated tracks.

Other useful photon features for classification are those related to conversion:

- `ph_zconv` (Fig. 4.6) and `ph_Rconv` (Fig. 4.7): these characteristics contain information of the conversion point in the R and z coordinates. `True_el/reco_ph` have a conversion point close to the point of interaction as their track starts from the beginning.

Figure 4.4: The `el_track_ep` distributionFigure 4.5: The `el_trackz0` distributionFigure 4.6: The `el_track_ep` distributionFigure 4.7: The `el_trackz0` distribution

- `ph_pt1conv`, `ph_pt2conv` and `ph_ptconv` (Fig. 4.8). All these three features have been scaled with `ph_c1_pt` to eliminate all kinematics information.

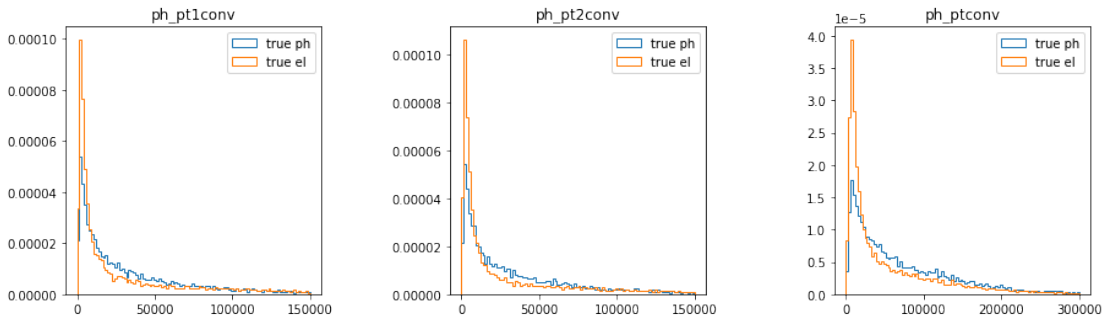


Figure 4.8: The reconstructed photons momentum distributions

- `pt1conv/ptconv` contains information about conversion symmetry (Fig. 4.9). This characteristic for `true_el/reco_ph` tends to be close to 1, since most of them have a single track. More information is presented in the Appendix A

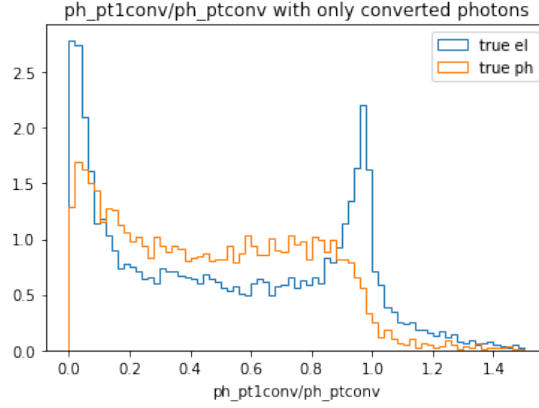


Figure 4.9: The reconstructed photons `pt1_conv/pt_conv` distributions

4.1.2 Training and Model

The algorithm training is based on a LGBM function `lgb.train`, which has as inputs: parameter set, train dataset, the maximum number of training steps, validation sets, early stopping rounds. The validation sets are those on which the loss function is calculated during training. For training to create the model they must be `lgb_train` and `lgb_test`, whereas for the hyperparameters optimization training they are `lgb_train` and `lgb_valid`, as said before.

To prevent overfitting, the model will train until the validation score stops improving. Validation score needs to improve at least every `early_stopping_rounds` round(s) to continue training [34].

Once the training is over, it is possible to analyse the model and how it uses the discriminatory variables:

- the Objective function behaviour. Its value is plotted after each iteration for both the training and test set. This plot is useful because a large difference between the train and test datasets may indicate overfitting (In Fig 4.10 one can see that no overfit is present).
- feature importance plots show the number of times that a feature was used in the splitting of a branch (Figure 4.11) and the average training loss reduction gained when using a feature for splitting (Figure 4.12).

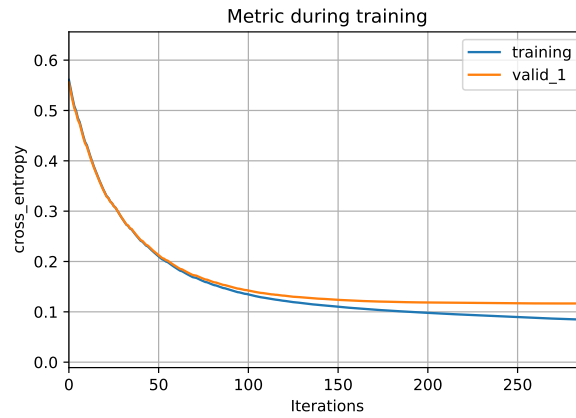


Figure 4.10: Loss value after each iteration, Model trained on double reconstruction

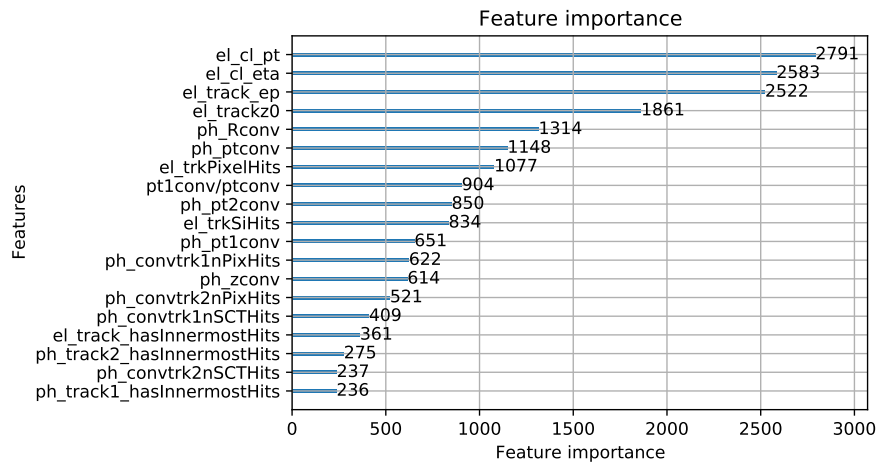


Figure 4.11: Feature importance, Model trained on double reconstruction

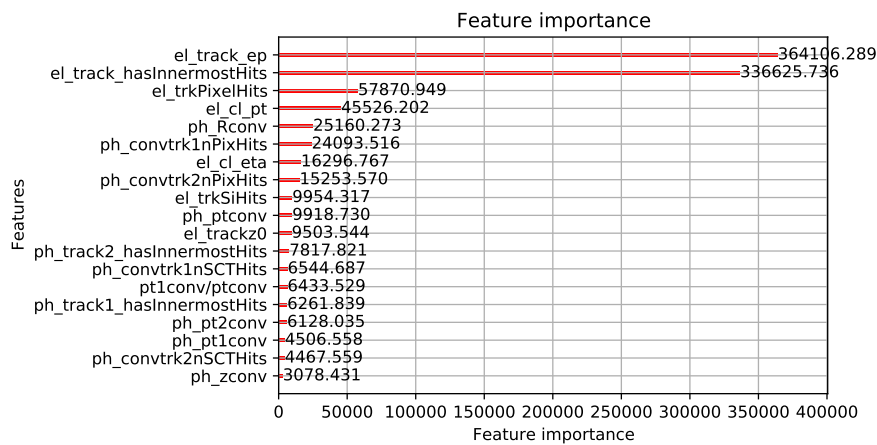


Figure 4.12: Feature importance Gain, Model trained on double reconstruction

4.1.3 Parameters and Hyperparameter optimization

Parameters and hyperparameters regulate training and determine the performance of the model created. Hyperparameters are LGMB [34] adjustable parameters chosen for the training of a model, which regulate the training process.

The objective of the hyperparameter optimization is to search the various configurations of the hyperparameters, contained in the hyperparameter space, to find a configuration with the best performance. The hyperparameter space consists of two types of hyperparameters: discrete or continuous.

So after defining the space of the hyperparameters, we move on to the search for the best configuration. This search can be of various types:

- **random search:** in random search the hyperparameter values are randomly selected from the defined search space. It allows the hyperparameter space to include both discrete and continuous hyperparameters;
- **grid search:** grid sampling performs a simple grid search on all possible values in the defined search space;
- **Bayesian search:** bayesian sampling uses the knowledge of the previous samplings when choosing hyperparameter values, effectively attempting to improve the reported primary metric.

The bayesian search is used by the Python library *Hyperopt* [36], which is the one used in this work in order to perform the optimization.

The training parameters are listed in the Table 4.1 and the ones that have been optimized are:

- **bagging:** in bagging, several models of the same type are trained on different datasets (aggregation, typical of ensemble learning), each obtained from the initial dataset by random sampling with replacement (bootstrap).
- **number of leaves:** it is the maximum number of leaves, which a tree could have. It is useful to avoid overfitting.
- **feature fraction:** 0.82 feature fraction means LightGBM will use only 82% of features randomly in each iteration for building trees.
- **learning rate:** it determines the effect of each tree on the final outcome.

Before starting with the hyperparameter optimization the train dataset has been splitted: 75% stays as the train set whereas the 25% becomes the validation set, which is used during the hyperparameter optimization instead of the test set as validation. This is done with the aim of avoiding the introduction of a bias.

After 200 searches the best hyperparameters are selected and used for the creation of the models, which have three different hyperparameters sets (shown in Table 4.1).

Parameter	Double Reco	Old ambiguous	New ambiguous
Metric	xentropy	xentropy	xentropy
Objective	xentropy	xentropy	xentropy
Bagging seed	42	42	42
Feature fraction seed	42	42	42
Is unbalance	True	True	True
Number of leaves	57	62	73
Feature fraction	0.946	0.801	0.966
Bagging	0.743	0.705	0.750
Learning rate	0.056	0.054	0.074

Table 4.1: Parameters and hyperparameters used in the models' training. The description of parameters and hyperparameters can be found in Appendix B.

4.2 BDT applied on test set

Once the models are created, these trained BDTs are applied to the test dataset. Each model gives to each particle a score between 0 and 1 according to its features. The scores are plotted as two histograms, one for *True Electrons* and the other for *True Photons*, and compared, as shown in Figure 4.13.

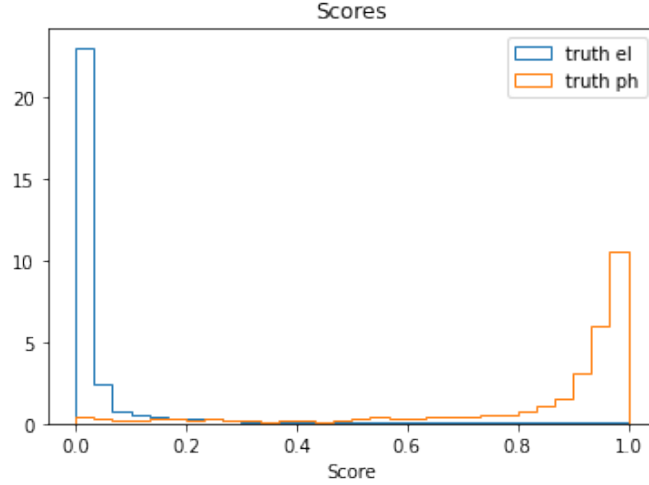


Figure 4.13: Scores histogram

From these histograms is possible to extract the *True Photons* efficiency ($P(\text{reco}_{el}|\text{true}_{el})$) and the *True Electron* efficiency ($P(\text{reco}_{ph}|\text{true}_{ph})$) by making cuts (thresholds) on the scoring range $[0,1]$.

In order to evaluate the models efficiencies the scores, ROC and AUC are plotted and calculated. The Receiver Operating Characteristic (ROC) curve is a graph showing the

performance of the classification model with all possible thresholds. This curve plots two parameters: $P_{e|e}$ versus $P_{\gamma|\gamma}$ at different classification thresholds connecting the point (1,0) to (0,1). The performance estimator is the entire two-dimensional area underneath the entire ROC curve, called AUC: the closer the AUC is to 1, the more predictive the model is. A ROC curve example is shown in Figure 4.14: it represents the ROC curve of the model trained on double reconstruction set (train=dr), which is applied on old ambiguous particles test set.

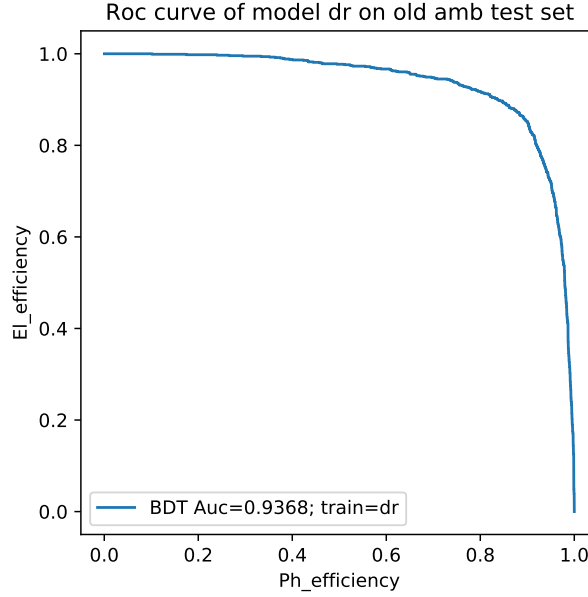


Figure 4.14: ROC curve of the model trained on double reconstructions, which is applied on the old ambiguous test set

4.3 BDT results

Finally, after the choice of the discriminating features, the optimisation of the hyperparameters and the training, the models are created and their performance can be compared on various test sets, which must be the same for the three models and consistent: for example it is not useful to apply the model trained on the old ambiguous particles on a test set composed of particles classified as ambiguous by the new ambiguity tool. The test set which provides more information is the *outer* set. It consists of single reconstructions and all double reconstructions.

Once the models are applied on the *outer* test set, their outputs must be corrected:

- the single reconstruction must be respected. If the object is reconstructed as a photon y^{pred} is fixed at 1, whereas if it is reconstructed as an electron y^{pred} is fixed

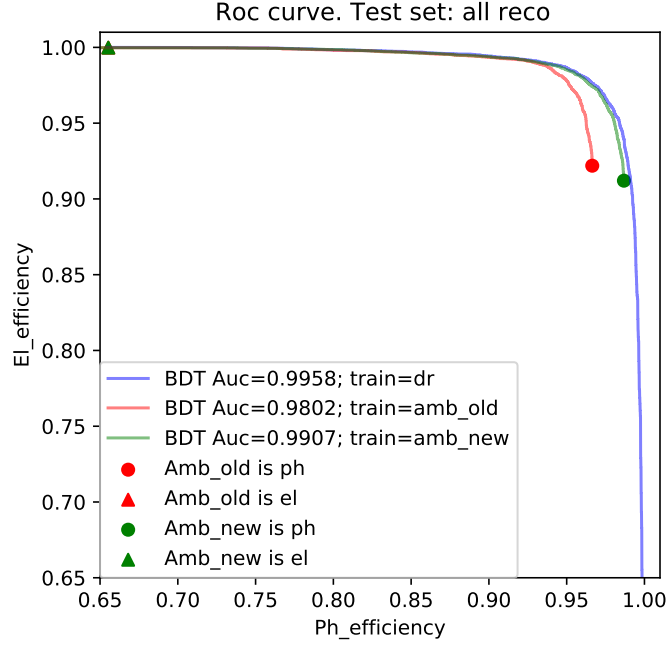


Figure 4.15: ROC curves of the three models on the *outer* set as test set

at 0.

- when `el_amb_type` and `ph_amb_type` are equal to 0 or 6, the outputs of the models trained on ambiguous particles are set with $y^{pred}=0$ or 1. This is because these models can only work on ambiguous objects and for the others they must adapt to the decisions of the ambiguity tools.

Figure 4.15 shows the ROC curves of the three models with *outer* as test set. In this figure there are three curves, one for each model, and two pairs of points. Each curve represents the performance of one of the three classification models. Instead each point $(x,y)=(\epsilon_{ph},\epsilon_{el})$ represents the photon and electron efficiencies if we consider all the ambiguous objects as photons (\bullet) or electrons (\triangle). They obviously represent the limits of the ROC curves of the two models trained on ambiguous objects.

Following the results shown in Figure 4.15, the Gradient Boosted Decision Tree algorithm is an excellent tool for resolving the ambiguity between photons and electrons. When the three models are applied on a single particles dataframe, they achieve excellent performance:

- the theoretical limit (infinite storage memory) is set by the *double reconstruction* model, which reaches an efficiency of $\sim 99.58\%$ in resolving ambiguity.
- the models based on pre-classifiers also achieve excellent performances. The *new*

amb model reaches an efficiency of $\sim 99.07\%$ in resolving ambiguity, whereas the *old amb* model reaches an efficiency of $\sim 98.02\%$. If the photon efficiency ϵ_{ph} is fixed the difference between the electron efficiencies $\epsilon_{el}^{new} - \epsilon_{el}^{old}$ is in its maximum value equal to $\sim 3\%$; on the contrary if the electron efficiency ϵ_{el} is fixed the difference between the photon efficiencies $\epsilon_{ph}^{new} - \epsilon_{ph}^{old}$ is in its maximum value equal to $\sim 7\%$.

The *new amb* model performance is very close to that of *double reconstruction* model and this means that even using pre-classifiers in the analysis phase it is possible to achieve very good results.

Chapter 5

Conclusion

Excellent performance in the electrons and photons reconstruction with the ATLAS detector at the LHC is essential to exploit the full physics potential of ATLAS, both in searches for new physics and in precision measurements. For instance, the good electron and photon reconstruction performance played a critical role in the discovery of a Higgs boson, announced by the ATLAS Collaboration in 2012 [18], and in the measurement of its properties.

At the reconstruction level only simple algorithms are used. They reconstruct only the simplest cases (single reconstructions) whereas the other objects are flagged as ambiguous. The ambiguous objects classification can be approached with machine learning techniques, which provide better results with respect to simple cut-based selections. In particular in this thesis the usage of supervised learning algorithms (GBDT) have been studied.

The GBDTs applied on single particle simulations samples can achieve excellent results. The theoretical limit is constituted by *double reconstructions* model, which has been trained on the input objects of ambiguity tools. It explores the possibility to replace the current classification algorithm with a gradient boosted decision tree for all electrons and photons candidates. This model is able to correctly classify up to $\sim 99.58\%$ of particles in the *outer* test set. The other two models can also achieve excellent particle classification capabilities: on the outer test set the *old amb* model, which has been trained on objects flagged as “ambiguous” by the old ambiguity tool, correctly classifies $\sim 98.08\%$ of particles, whereas the *new amb* model, which is based on the new ambiguity tool, $\sim 99.07\%$. The new ambiguity tool increases the performance of a GBDT based on ambiguous objects, allowing a better classification than the old one: if the photon efficiency ϵ_{ph} is fixed the difference between the electron efficiencies $\epsilon_{el}^{new} - \epsilon_{el}^{old}$ is in its maximum value equal to $\sim 3\%$; on the contrary if the electron efficiency ϵ_{el} is fixed the difference between the photon efficiencies $\epsilon_{ph}^{new} - \epsilon_{ph}^{old}$ is in its maximum value equal to $\sim 7\%$.

As shown by these results, models trained on ambiguous objects of ambiguity tools are able to achieve very good performance in classification approaching very close to the theoretical limit. In particular, with the introduction of the new ambiguity tool,

performance has improved and therefore classification efficiency has improved as well.

Appendix A

Conversion symmetry

Conversion symmetry plots of the reconstructed and converted photons.

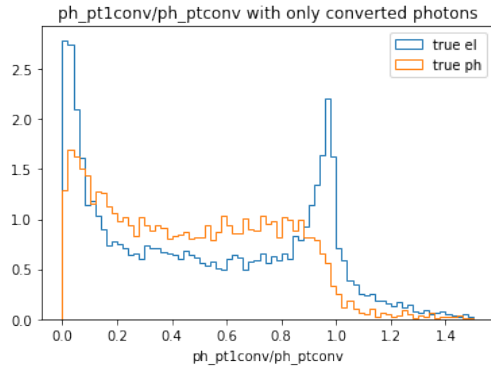


Figure A.1: The $\text{ph_pt1conv}/\text{ph_ptconv}$ distribution

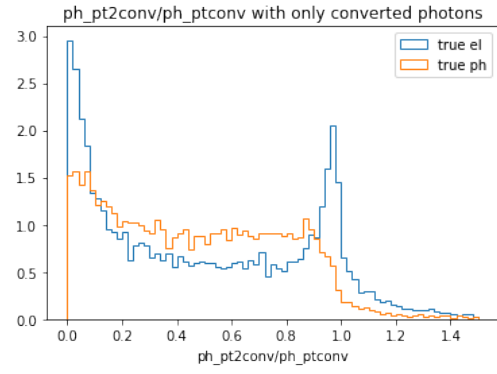


Figure A.2: The $\text{ph_pt2conv}/\text{ph_ptconv}$ distribution

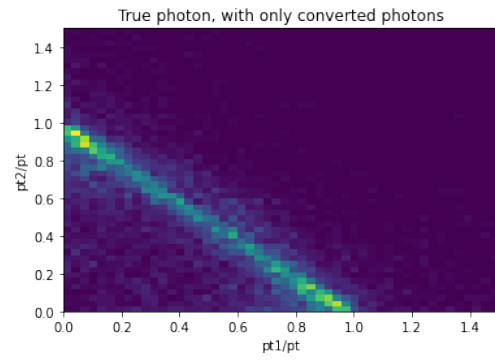
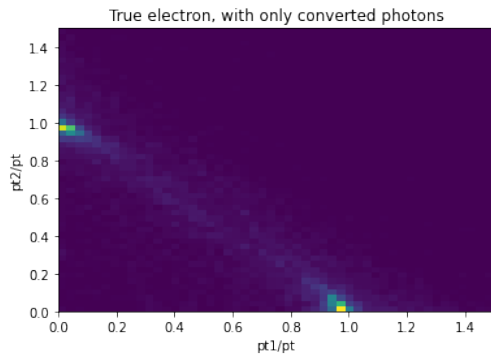


Figure A.3: 2D histogram of $pt1/pt$ and $pt2/pt$ distributions for *True Electron* Figure A.4: 2D histogram of $pt1/pt$ and $pt2/pt$ distributions for *True Photon*

Appendix B

Parameters and Hyperparameters description

The choice of which parameters and hyperparameters to use and their values is fundamental to create a model suitable for the dataframe to analyse. In this thesis the parameters and hyperparameters used by LGBM are:

- **metric:** it specifies loss for model building. Since we are dealing with a binary classification problem the *cross_entropy* metric is used;
- **objective:** it specifies the type of ML model. Since we are dealing with a binary classification problem the *cross_entropy* objective is used;
- **bagging seed:** it is the seed for *bagging*;
- **feature fraction seed:** it is the seed for *feature fraction*;
- **is unbalance:** it says whether the dataset is balanced or not. If it is not balanced (`is_unbalance==True`) the algorithm will try to automatically balance the weight of the dominated label. In this thesis this parameter is set to False (`is_unbalance==False`);
- **number of leaves:** number of leaves in full tree. It is a hyperparameter 4.1.3.
- **feature fraction:** 0.82 feature fraction means LightGBM will use only 82% of features randomly in each iteration for building trees. It is a hyperparameter.
- **bagging:** in bagging, several models of the same type are trained on different datasets (aggregation, typical of ensemble learning), each obtained from the initial dataset by random sampling with replacement (bootstrap). It is a hyperparameter.
- **learning rate:** it determines the effect of each tree on the final outcome. It is a hyperparameter.

Appendix C

Ambiguity Tools

The ambiguity tool, as already mentioned in the Section 2.5, is a tool used at the reconstruction level with the purpose that if a particular object can be easily identified only as a photon (a cluster with no good track attached) or only as an electron (a cluster with a good track attached and no good photon conversion vertex), then only a photon or an electron object is created for analysis; otherwise, both an electron and a photon object are created, ambiguous objects.

In this thesis two tools were used, the old ambiguity tool and the new one. Using the old one, which follows the flowchart shown in Figure 2.1, the ambiguous objects are tagged with `el_amb_type` and `ph_amb_type` between 1 and 5, whereas unambiguous objects with `el_amb_type` and `ph_amb_type` other than 0 (electron) or 6 (photon). The unambiguous objects are usually reconstructed in only one way, but in this thesis, in order to create the double reco model they present both reconstructions. The new ambiguity tool has an additional ambiguous tag, `el_amb_type` and `ph_amb_type` equal to 8.

Bibliography

- [1] European organization for nuclear research, *LHC design report*, CERN libraries, Geneva (2004). <http://cds.cern.ch/record/782076/files/>
- [2] F. Gianotti, *Collider physics: LHC*, EP Division, CERN, Geneva, Switzerland. <https://cds.cern.ch/record/458489/files/p219.pdf>
- [3] *The Large Hadron Collider (LHC)* <https://home.cern/science/accelerators/large-hadron-collider>
- [4] *The Large Hadron Collider (LHC): the guide* <https://cds.cern.ch/record/2255762/files/CERN-Brochure-2017-002-Eng.pdf>
- [5] *The ATLAS experiment* <https://home.cern/science/experiments/atlas>
- [6] The ATLAS Collaboration et al 2008 JINST3 S08003 <https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003/pdf>
- [7] *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade*, ATLAS Collaboration, CERN, CERN-LHCC-2020-007. ATLAS-TDR-031, Geneva, Jun 2020 <https://cds.cern.ch/record/2719855>
- [8] H. Pernegger, *The Pixel Detector of the ATLAS Experiment for LHC Run-2*, ATL-INDET-PROC-2015-001. <https://cds.cern.ch/record/1985432/files/ATL-INDET-PROC-2015-001.pdf>
- [9] J.R. Pater *The ATLAS SemiConductor Tracker operation and performance*, 2012 JINST7 C04001. <https://iopscience.iop.org/article/10.1088/1748-0221/7/04/C04001/pdf>
- [10] Adrian Vogel *ATLAS Transition Radiation Tracker (TRT): Straw Tube Gaseous Detectors at High Rates*, CERN, Geneva, ATL-INDET-PROC-2013-005, Apr 2013, ATL-INDET-PROC-2013-005. <https://cds.cern.ch/record/1537991>
- [11] *ATLAS calorimetry*, P. Puzo, Laboratoire de l'Accélérateur Linéaire, IN2P3-CNRS et Université Paris-Sud, BP 34, 91898 Orsay Cedex, France, Elsevier Science B.V., 2002, http://ific.uv.es/~cabrera/teaching/atlas_i.pdf

- [12] *The ATLAS Calorimetry* <https://atlas.cern/discover/detector/calorimeter>
- [13] *The ATLAS muon spectrometer: calibration and pattern recognition*, N. Van Eldik, CERN-THESIS-2007-045, 2007, <http://cds.cern.ch/record/1044839>
- [14] *Trigger and Data Acquisition System* <https://atlas.cern/discover/detector/trigger-daq>
- [15] *The Run-2 ATLAS Trigger System: Design, Performance and Plan*, Martin zur Nedden, CERN, ATLAS Collaboration, Geneva, Dec 2016, ATL-DAQ-PROC-2016-039, <https://cds.cern.ch/record/2238679>
- [16] *The ATLAS Physics* <https://atlas.cern/discover/physics>
- [17] *The CMS experiment at the CERN LHC*, The CMS Collaboration <https://doi.org/10.1088/2F1748-0221/2F3/2F08/2Fs08004>
- [18] *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, The ATLAS Collaboration, In: *Phys. Lett. B* 716.arXiv:1207.7214.CERN-PH-EP-2012-218 (Aug. 2012) DOI:<https://doi.org/10.1016/j.physletb.2012.08.020>, URL:<http://www.sciencedirect.com/science/article/pii/S037026931200857X>
- [19] Wu Xin, Clark Allan and Campanelli Mario *Electron and photon identification in ATLAS*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006 https://link.springer.com/chapter/10.1007/978-3-540-32841-4_21
- [20] *Electron and photon performance measurements with the ATLAS detector using the 2015-2017 LHC proton-proton collision data* <https://iopscience.iop.org/article/10.1088/1748-0221/14/12/P12006>
- [21] *A neural network clustering algorithm for the ATLAS silicon pixel detector*, The ATLAS collaboration <https://doi.org/10.1088/2F1748-0221/2F9/2F09/2Fp09009>
- [22] <https://link.springer.com/content/pdf/10.1140/epjc/s10052-019-7140-6.pdf>
- [23] *Optimisation of the ATLAS Track Reconstruction Software for Run-2*, Andreas Salzburger, CERN, Switzerland, <http://iopscience.iop.org/1742-6596/664/7/072042>
- [24] *Measurement of the photon identification efficiencies with the ATLAS detector using LHC Run-1 data*, In: *Eur. Phys. J. C*. 76.CERN-EP-2016-110, June 2016, 666. 57 p. <https://cds.cern.ch/record/2158117>
- [25] *Electron and photon identification with the ATLAS detector*, Proklova Nadezda, ATLAS Collaboration, Aug 2018, ATL-PHYS-SLIDE-2018-606, <http://cds.cern.ch/record/2634679>

- [26] *Electron and photon energy calibration with the ATLAS detector using 2015-2016 LHC proton-proton collision data*, ATLAS Collaboration, JINST 14 (2019) P03017, CERN-EP-2018-296, DOI:<https://arxiv.org/ct?url=https%3A%2F%2Fdx.doi.org%2F10.1088%2F1748-0221%2F14%2F03%2FP03017&v=7010984d>, <https://arxiv.org/abs/1812.03848>
- [27] *Search for supersymmetry with a compressed mass spectrum using the ATLAS detector*, Rossini Lorenzo, <http://hdl.handle.net/2434/683343>
- [28] *A New Approach to Linear Filtering and Prediction Problems*, R. E. Kalman, Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960)
- [29] *The global χ^2 track fitter in ATLAS*, T. G. Cornelissen et al, <https://iopscience.iop.org/article/10.1088/1742-6596/119/3/032013>
- [30] *Lifelong Machine Learning*, Zhiyuan Chen and Bing Liu, Morgan & Claypool Publishers, Nov 2016, <https://www.cs.uic.edu/~liub/lifelong-machine-learning-draft.pdf>
- [31] *Machine Learning: A Probabilistic Perspective*, Kevin P. Murphy, The MIT Press, [https://scholar.google.it/scholar?q=K.P.+Murphy,+Machine+Learning.+A+Probabilistic+Perspective+,+MITpress+\(2012\)&hl=it&as_sdt=0&as_vis=1&oi=scholar](https://scholar.google.it/scholar?q=K.P.+Murphy,+Machine+Learning.+A+Probabilistic+Perspective+,+MITpress+(2012)&hl=it&as_sdt=0&as_vis=1&oi=scholar)
- [32] <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
- [33] *4Classification:Basic Concepts,Decision Trees, andModel Evaluation*, <https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf>
- [34] *LightGBM's documentation* <https://lightgbm.readthedocs.io/en/latest/>
- [35] *Scikit-learn: Machine Learning in Python*, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- [36] *Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning*, J. Bergstra, D. Yamins, D. D. Cox, ICML, 2013
- [37] *Data structures for statistical computing in python*, McKinney, W., & others. (2010), In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56)