



1

2

3

4

UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea Triennale in Fisica

5

6

7

**Electrons and Photons classification using
machine learning techniques with the
ATLAS detector at the LHC**

8

CODICE PAC: 07.05.-t

Relatore:

Prof. Leonardo Carminati

Correlatori:

Dott. Ruggero Turra
Dott. Davide Mungo

Candidato:

Edoardo Tronconi
MATRICOLA: 885655

10

ANNO ACCADEMICO 2018-2019

₁₁ Contents

₁₂	Introduction	_{vii}
₁₃	1 The Large Hadron Collider and the ATLAS experiment	1
₁₄	1.1 The LHC	1
₁₅	1.1.1 The CERN accelerator complex	1
₁₆	1.1.2 The LHC structure	1
₁₇	1.1.3 Phenomenology of Proton-Proton collisions	3
₁₈	1.1.4 Experimental Challenges	4
₁₉	Pile-up	4
₂₀	QCD Background	4
₂₁	1.2 The ATLAS experiment	5
₂₂	1.2.1 The Detector Layout	5
₂₃	The ATLAS coordinate system	6
₂₄	The Magnet System	7
₂₅	The Inner Detector	7
₂₆	The Calorimeters	8
₂₇	The Muon Spectrometer	10
₂₈	1.2.2 The Trigger System	10
₂₉	The L1 Trigger	10
₃₀	The HLT	11
₃₁	1.2.3 Search for the Standard Model Higgs Boson	11
₃₂	Production	12
₃₃	Decays and Search strategy	13
₃₄	Discovery	14
₃₅	Low Mass	15
₃₆	2 Electron and Photon Reconstruction	17
₃₇	2.1 Energy Clusters and Track reconstruction	18
₃₈	2.1.1 Energy Clusters in the EM calorimeter	18
₃₉	Sliding Window algorithm	18
₄₀	Topological clustering algorithm	19
₄₁	2.1.2 Track reconstruction and matching	19
₄₂	Track matching and conversion vertex reconstruction . .	21
₄₃	2.1.3 Supercluster formation	22
₄₄	2.2 Analysis objects building	23
₄₅	2.2.1 Identification	24
₄₆	Electrons	24
₄₇	Photons	26
₄₈	2.2.2 Isolation	26

49	3 Machine Learning and Gradient Boosted Trees	29
50	3.1 Machine learning	29
51	3.1.1 Elements of Supervised Learning	29
52	3.2 Gradient Boosted Trees	30
53	3.2.1 Decision Trees	31
54	3.2.2 Tree Ensembles and Gradient Boosting	31
55	Gradient Boosting	32
56	3.3 LightGBM	34
57	4 Photons and Electrons Classification	35
58	4.1 BDT Training	35
59	4.1.1 Single Particle sample generation and Dataset Preparation	36
60	Feature Selection	36
61	4.1.2 Hyperparameter optimization and Model Training	38
62	Model Analysis	40
63	4.1.3 BDT Prediction and Analysis	41
64	Scores distribution	41
65	Efficiency evaluation and ROC curve	41
66	4.2 Physics analysis application	42
67	4.2.1 Sample generation and Dataset Preparation	42
68	Training and Physics datasets feature comparison	44
69	4.2.2 BDT Prediction and Analysis	44
70	ROC analysis	44
71	Significance Maximisation	46
72	4.2.3 $M_{\gamma\gamma}$ distributions	46
73	4.3 Results	48
74	5 Conclusions	51
75	A Tables	53
76	A.1 Single Particle Results	53
77	A.1.1 Efficiencies	53
78	A.2 Physics Sample results	54
79	A.2.1 Efficiencies and Significance	54
80	A.2.2 Expected event distribution in $m_{\gamma\gamma}$	55
81	B Code Snippets	59
82	B.1 Model Training	59
83	B.1.1 Training Sample Preparation	59
84	B.1.2 LightGBM Dataset Preparation	60
85	B.1.3 Hyperparameter Optimization	60
86	B.1.4 BDT Training	61
87	B.1.5 Training Analysis	62
88	B.1.6 ROC Analysis	62
89	B.2 Physics Application	63
90	B.2.1 Physics Sample Preparation	63
91	B.2.2 Event Classification with the trained BDT	67
92	B.2.3 Analysis	68

⁹⁴ Introduction

⁹⁵ The Large Hadron Collider (LHC) at CERN in Geneva presently is the largest
⁹⁶ particle collider in the world. It accelerates protons up to 99.99 % the speed of light
⁹⁷ and provides proton-proton collisions 10^9 times per second, with a center of mass
⁹⁸ energy of 6.5 TeV.

⁹⁹ Along the LHC ring many experiments take advantage of the high energy col-
¹⁰⁰ lisions between protons. The ATLAS and CMS experiments are general purpose
¹⁰¹ detectors conceived to provide measurements over a wide range of physics phenom-
¹⁰² ena, both within the Standard Model (SM) and in the realm of new physics.

¹⁰³ In 2012 the measurements of both the ATLAS and the CMS experiments lead to
¹⁰⁴ the observation of a new particle compatible with the Standard Model Higgs Boson
¹⁰⁵ with mass $m_H \sim 125$ GeV. Several high precision measurements were taken in the
¹⁰⁶ following years (and are still taking place at the moment), to evaluate the particle's
¹⁰⁷ spin, parity as well as the production cross sections and branching ratios in different
¹⁰⁸ decay channels.

¹⁰⁹ The decay channel $H \rightarrow \gamma\gamma$ is especially interesting at the ATLAS experiment:
¹¹⁰ despite its small branching ratio ($\sim 10^{-3}$) it has a clear signature, a good signal to
¹¹¹ background ratio, and the signal can be clearly observed as a peak in the di-photon
¹¹² invariant mass distribution, thanks to the excellent electromagnetic calorimeter per-
¹¹³ formance.

¹¹⁴ Despite its success, the Standard Model is not a complete theory, and many SM
¹¹⁵ extensions have been developed. The possibility of a Low Mass Higgs-like boson
¹¹⁶ in the mass range (60-120 GeV) is part of some of the aforementioned extensions.
¹¹⁷ The search of a Low Mass Higgs-like particle in the $\gamma\gamma$ decay channel is especially
¹¹⁸ affected by electrons mistakenly being reconstructed as photons because of the large
¹¹⁹ number of electrons being produced around the invariant mass of 90 GeV from Z
¹²⁰ bosons decays to electrons and positrons.

¹²¹ The reconstruction of photons at ATLAS relies on the reconstruction of clusters
¹²² of energy left in the Electromagnetic Calorimeter, as well as of tracks and conversion
¹²³ vertices in the inner detector. The fact that electrons loose energy by radiating
¹²⁴ photons through brehmsstrahlung, and photons create electron-positron pairs makes
¹²⁵ the discrimination of electrons and photons a particularly challenging task. At the
¹²⁶ moment the algorithm used for this scope relies on a rigid set of selections on specific
¹²⁷ discriminating variables to distinguish between the two particles. In some cases the
¹²⁸ current algorithm doesn't take a decision: a particle in the detector is reconstructed
¹²⁹ both as a photon and an electron candidate, and they are marked as ambiguous. In
¹³⁰ these cases each analysis has to decide whether or not to include ambiguous objects.

¹³¹ In this thesis a machine learning approach is used to improve the discrimination
¹³² between electrons and photons. A Gradient Boosted Decision Tree (BDT) is used,
¹³³ in the LightGBM implementation, to effectively separate electrons from photons
¹³⁴ with higher efficiency than the current algorithms. This model assigns a score to
¹³⁵ each particle, depending on its likelihood to be an electron or a photon. By varying

¹³⁶ the threshold for which particles are classified as electrons rather than photons, it is
¹³⁷ possible to easily tune the signal efficiency and background rejection of the model.
¹³⁸ This is particularly useful because different physics analyses can opt for different
¹³⁹ thresholds, by choosing the one which maximises the significance of their signal.

¹⁴⁰
¹⁴¹ The thesis is structured in the following way: Chapter 1 describes the Large
¹⁴² Hadron Collider and the ATLAS experiment, Chapter 2 delineates the electron and
¹⁴³ photons reconstruction algorithms, in Chapter 3 a general introduction to machine
¹⁴⁴ learning algorithms and Boosted Decision Trees is given, and finally in Chapter 4
¹⁴⁵ the training of the LightGBM model and the application of the model to a potential
¹⁴⁶ search for a low mass Higgs particle is outlined.

147 CHAPTER 1

148 **The Large Hadron Collider and
149 the ATLAS experiment**

150 **1.1 The LHC**

151 The CERN Large Hadron Collider is the world's largest particle collider. It consists
152 of a 27 km ring of superconducting magnets with a number of accelerating structures
153 to boost the energy of the particles along the way.

154 Inside the accelerator, two particle beams travel at close to the speed of light in
155 opposite directions before they are made to collide. The beams are located in sepa-
156 rate beam pipes (two tubes kept at ultrahigh vacum) and they are guided around the
157 accelerator ring by the magnetic fiel generated by superconducting electromagnets.

158 The LHC complex provides pp collisions and heavy-ion collisions. In pp mode it
159 produces collisions every 25 ns, with a center-of-mass energy of 13 TeV.

160 **1.1.1 The CERN accelerator complex**

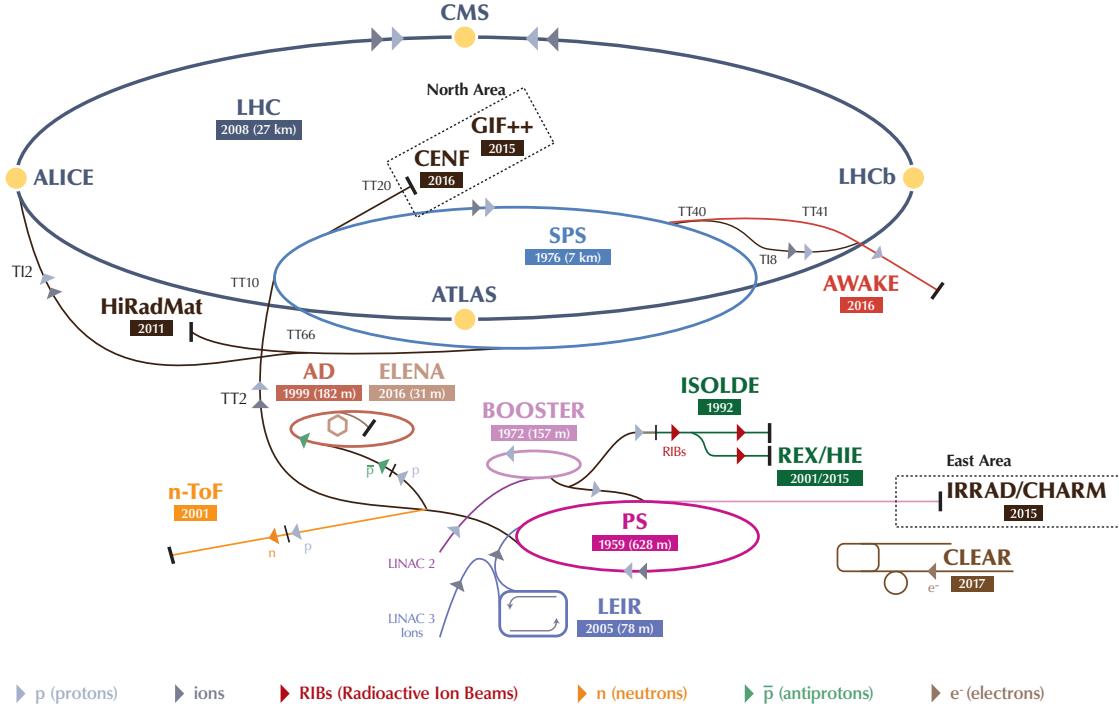
161 The accelerator complex at CERN is a succession of machines accelerating particles
162 at increasingly higher energies. Each machine injects the beam into the next one,
163 which takes over to bring the beam to an even higher energy, and so on.

164 The protons source are hydrogen atoms, whence electrons are removed by means
165 of a strong electric field. The free protons are driven into the linear accelerator
166 LINAC2 and accelerated up to an energy of 50 MeV, then they are injected into the
167 Proton Synchrotron Booster (PSB) where their energy is increased up to 1.4 GeV.
168 The beam is then fed into the Proton Synchrotron (PS) where it is accelerated to
169 25 GeV. The last step of the chain, where the protons obtain an energy of 450 GeV
170 is the Super Proton Synchrotron (SPS), after which the beams are injected into the
171 LHC where they are accelerated to 6.5 TeV. The complete CERN accelerator facility
172 is shown in Figure 1.1.

173 **1.1.2 The LHC structure**

174 The LHC [1, 2] is composed of eight arcs and eight *insertions*. The arcs contain the
175 dipole bending magnets, while each insertion consists of a long straight section plus
176 two transition regions (the *dispersion suppressors*). Within the straight sections
177 beam collisions, beam dumping and beam cleaning happen.

178 In the accelerator particles are manipulated using electromagnetic devices: su-
179 perconducting dipole magnets (Figure 1.2) keep the protons in a circular path,
180 quadrupole magnets focus the beam, and magnets of higher multipole order can be
181 used to correct small imperfections in the beam trajectory.



208 squeezed to about $20\ \mu\text{m}$ thick when they approach a collision point, to maximize
209 the chances of pp interactions.

210 The main effect of bunching is that collisions take place ad discrete time intervals:
211 interactions are 25 ns apart. The instantaneous luminosity¹ at the LHC can be
212 computed as:

$$\mathcal{L} = \frac{n_{beam} N_{bunch}^2 \nu_{rev} \gamma}{4\pi \epsilon_n \beta^*} \quad (1.1)$$

213 where n_{beam} is the number of bunches in each beam, N_{bunch} is the number of protons
214 per bunch, ν_{rev} is the revolution frequency of a bunch around the ring, γ is the
215 relativistic gamma factor, ϵ_n depends on the momentum and position distribution
216 of the protons in the ring and β^* reflects the transverse size of the particle beam
217 (which is $\sim 16\ \mu\text{m}$ at the intersection point).

218 Since bunches cross about 40 million times per second, and, on average, at each
219 crossing between two bunches, 25 interactions happen between the protons, the LHC
220 generates one billion particle collisions every second.

221 The pp collisions take place in four collision points along the LHC beam. Around
222 each one of those, huge underground caverns house the four main experiments at
223 LHC: A Large Ion Collider Experiment (ALICE) [3], A Toroidal LHC Apparatus (ATLAS) [4], the Compact Muon Solenoid (CMS) [5] and the Large Hadron
224 Collider beauty (LHCb) [6] experiment. ALICE is a detector optimized to measure
225 lead-ion collisions; it studies the properties of quark-gluon plasma. LHCb special-
226 izes in the study of the asymmetry between matter and antimatter in interactions
227 of B-particles. ATLAS and CMS on the other hand are general-purpose detectors,
228 they are designed to cover the widest possible range of physics, from precise
229 measurements of the Higgs boson to searches for new physics beyond the Standard
230 Model.

232 A few other minor experiments are installed near the main detectors, such as
233 TOTEM [7], LHCf [8] and MoEDAL [9].

234 1.1.3 Phenomenology of Proton-Proton collisions

235 The total inelastic proton-proton cross-section [2] is about 80 mb at $\sqrt{s} = 14\ \text{TeV}$.
236 The event rate² R is thus expected to be

$$R = \sigma \cdot \mathcal{L} = 80\ \text{mb} \cdot 10^{34}\ \text{cm}^{-2}\text{s}^{-1} \simeq 10^9\ \text{s}^{-1} \quad (1.2)$$

237 Two types of interactions between protons can take place at the collisions points:

- 238 • **Soft collisions:** most events are due to large-distance interactions between
239 the two incoming protons. In this case the momentum transfer of the interac-
240 tion is small, and therefore the particle scattering at large angle is suppressed.
241 The produced particles have large longitudinal momentum, but small trans-
242 verse momentum ($\langle p_T \rangle \simeq 500\ \text{MeV}$) relative to the beam line. The final states
243 arising from such interactions are called *minimum bias* events. They represent
244 the vast majority of the pp collisions but they are not interesting.

¹The instantaneous luminosity is the ratio between the number of events produced per second and their production cross section.

²The number of events produced per second by the pp interactions.

- 245 • **Hard collisions:** proton beams can be seen as beams of partons (quarks and
 246 gluons). Occasionally a head-on collision, characterized by large momentum
 247 transfer, between two partons of the incoming protons occur, producing final
 248 states at large angles with respect to the beam line (high p_T) and creating
 249 massive particles. Hard collisions, however, are rather rare compared to soft
 250 interactions. In this case the effective center-of-mass energy depends on the
 251 fraction of the protons momentum carried by the colliding partons. Producing
 252 a particle of mass 100 GeV thus requires two quarks (or gluons) which carry
 253 only 1% of the proton momentum, whereas a particle of mass 5 TeV can be
 254 produced if two partons carrying 35% of the proton momentum interact.

255 1.1.4 Experimental Challenges

256 The LHC experiments have to face two main experimental difficulties, which are
 257 caused by the machine luminosity and by the physics of pp collisions.

258 Pile-up

259 As previously mentioned, protons are grouped in bunches of $\sim 10^{11}$ particles colliding
 260 every 25 ns. Since the interaction rate is $\sim 10^9$ events/s, on average, 25 interactions
 261 occur simultaneously at each crossing. These give rise, every 25 ns, to about 1000
 262 charged particles in the detector over the pseudo-rapidity³ region $|\eta| < 2.5$.

263 Since the soft collisions are much more likely than the hard ones, most of the
 264 times the 25 interactions consist only of minimum bias events. On the other hand,
 265 when a high p_T event is produced, it is overlapped, on average, with 25 additional
 266 soft events, which are called *pile-up*.

267 In order to extract the interesting hard events from the pile-up, it is possible to
 268 exploit the fact that particles produced in minimum bias events have, as previously
 269 mentioned, a small p_T .

270 The pile-up is one of the most challenging difficulties for the LHC operation. To
 271 combat this issue, the detectors need three important features. Firstly, the LHC
 272 detectors must have a fast response time, otherwise the signal would be integrated
 273 over many bunch crossings and the pile-up effect would therefore be too large. Sec-
 274 ondly the detectors must have a fine readout granularity, to avoid that particles from
 275 the pile-up traverse the same detector elements as an interesting object. Finally the
 276 LHC detectors must be radiation resistant, because there is a high flux of particles
 277 coming from the pp collisions.

278 QCD Background

279 The rate of high- p_T events at a hadron collider is dominated by QCD jet production.
 280 Jets arise from the fragmentation of quarks and gluons in the final state, which
 281 are produced through a variety of Feynman diagrams. Jet production is a strong
 282 process and therefore has a large cross-section. Furthermore many physics channels
 283 contribute to this final state.

284 On the other hand, the most interesting processes at LHC are rare processes. For
 285 instance, the production cross section for the Higgs boson is five order of magnitude

³The pseudo-rapidity η is defined as $\eta = -\ln(\cot \frac{\theta}{2})$, where θ is the polar angle of a produced particle with respect to the beam line

286 smaller than the cross section for the production of jets with $p_T > 700$ GeV. Thus
287 there is no hope to detect a Higgs boson decaying into jets, since such final states
288 are swamped by the much larger jet rate (referred to as *QCD background*). Since
289 decays into leptons or photons have usually a smaller branching ratio than decays
290 into quarks, the price to pay to get rid of the QCD background is a smaller event
291 rate.

292 1.2 The ATLAS experiment

293 ATLAS [4, 10] is a general purpose experiment. Together with CMS it was designed
294 to exploit the full discovery potential that the LHC provides. Rather than focusing
295 on a particular physical process, ATLAS was designed to measure the broadest
296 possible range of signals.

297 At the time of the project design the main performance requirements of ATLAS
298 were summarized as follows:

- 299 • Leptons should be measured over the p_T range from a few GeV up to a few TeV.
300 This is necessary to detect the biggest variety of processes, from B-hadrons
301 decays to heavy particles decaying into leptons.
- 302 • Calorimetry should be as hermetic as possible, covering the full azimuthal
303 angle and pseudo-rapidity region up to $|\eta| \leq 5$. This is required for a reliable
304 measurement of the event transverse energy, which is in turn needed to
305 indirectly detect neutrinos. At e^+e^- colliders the total energy of the event
306 corresponds to the total center-of-mass energy, thus the presence of neutrinos
307 can be inferred from energy conservation criteria. In a pp collider, on the other
308 hand, only the energy of the protons is known, while the energy of the interacting
309 quarks, and thus of the final state, cannot be determined. However,
310 since the transverse momentum of the incident quarks is negligible, the total
311 transverse momentum of the final state has to be zero; this constraint can be
312 used to detect the presence of neutrinos.
- 313 • Excellent mass resolution ($\sim 1\%$) is needed for particles of mass up to a few
314 hundred GeV decaying into photons, electrons or muons. This is necessary,
315 for instance, to detect a Higgs boson signal in the $\gamma\gamma$ decay mode, over the $\gamma\gamma$
316 background (see Section 1.2.3).
- 317 • Triggering at the LHC must be extremely efficient, to reduce the 10^9 s $^{-1}$ event
318 rate to ~ 1000 recorded events per second, which is an affordable data storage
319 rate.

320 1.2.1 The Detector Layout

321 The ATLAS detector (shown in Figure 1.3) has a cylindrical structure, with a length
322 of 46 m, a diameter of 25 m and a weight of about 7000 t. It is the largest detector
323 ever built for a collider. The detector can be divided in three sections: an inner
324 detector, immersed in a magnetic field, which tracks charged particles and measures
325 their momenta; an electromagnetic calorimeter, which measures the energy and
326 position of electrons and photons, and contributes to their identification; a hadron

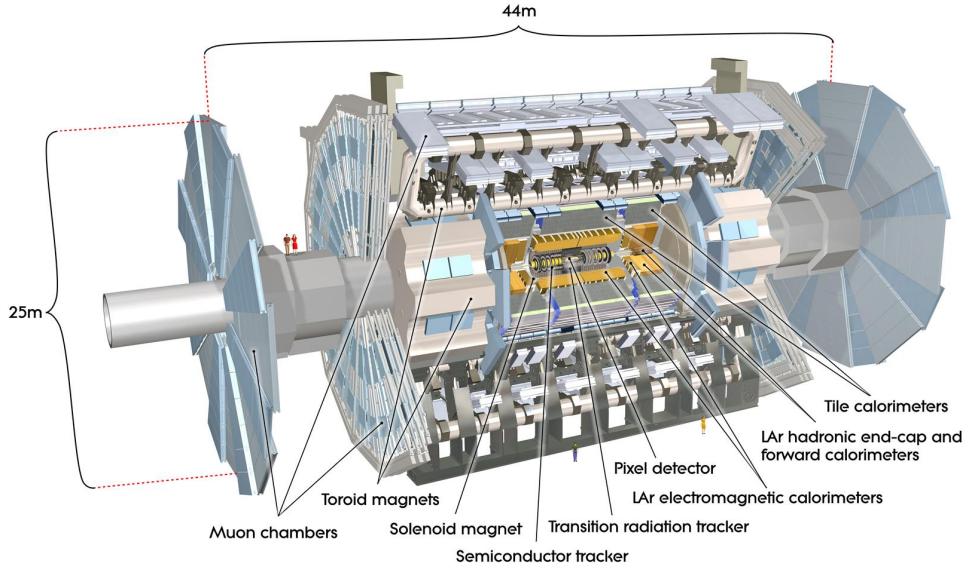


Figure 1.3: The ATLAS detector

327 calorimeter, which is used to measure the energy and position of hadrons and jets, as
 328 well as the total event missing transverse momentum (E_T^{miss}); a muon spectrometer,
 329 used to identify muons and to measure their momentum, together with the inner
 330 detector.

331 The ATLAS coordinate system

332 The position of particles inside the de-
 333 tector is described through a right-
 334 handed coordinate system (shown in fig-
 335 ure 1.4) whose origin coincides with the
 336 interaction point. The x -axis points to
 337 the center of the LHC ring, the y -axis
 338 points upwards and the z -axis points
 339 along the beam in the counterclockwise
 340 direction.

341 Polar coordinates are also used: the
 342 radial distance R represents the distance
 343 from the interaction point, ϕ describes
 344 the azimuthal angle around the z -axis,
 345 and from the polar angle θ the pseudorapidity η is defined as:

$$\eta = -\ln \left[\tan \left(\frac{\theta}{2} \right) \right] \quad (1.3)$$

346 this quantity spans the range $(-\infty, +\infty)$ rather than $(0, +\pi)$, and is in a one-to-
 347 one relation with θ . The use of η , though, is preferred over θ , because it is an
 348 invariant quantity over Lorentz boosts. Furthermore the angular distance between
 349 two particles can be defined as $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$.

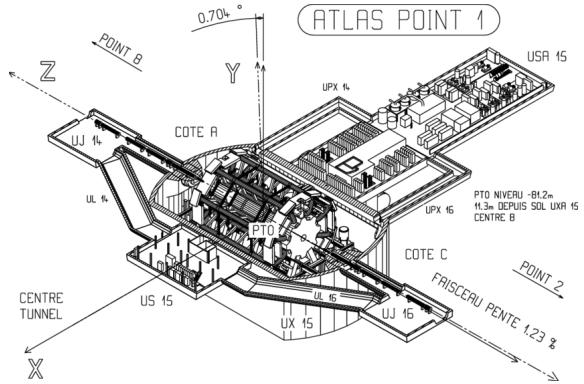


Figure 1.4: The ATLAS coordinate sys-
 tem

350 The Magnet System

351 ATLAS features a unique hybrid system of four
 352 large superconducting magnets [11]. This mag-
 353 netic system is 22 m in diameter and 26 m in
 354 length, with a stored energy of 1.6 GJ.

355 The magnet system consists of one solenoid,
 356 aligned on the beam axis and providing a 2 T
 357 axial magnetic field for the inner detector, and
 358 three toroids (one in the barrel and one in each
 359 end-cap), producing a toroidal magnetic field for
 360 the muon detector of about 1 T.

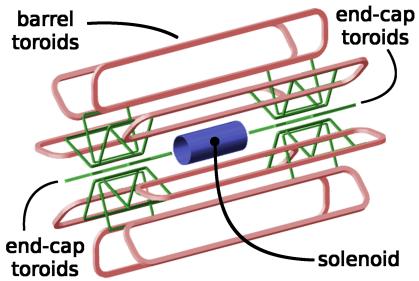


Figure 1.5: The magnet system

361 The Inner Detector

362 The ATLAS Inner Detector (ID) [12] is
 363 the innermost layer of ATLAS. Figure
 364 1.6 shows a cutout of the ID. The de-
 365 tector is designed to provide hermetic
 366 and robust pattern recognition, excel-
 367 lent momentum resolution and both pri-
 368 mary and secondary vertex measure-
 369 ments. The ID is enclosed in a cylin-
 370 drical envelope which is 6.2 m in length
 371 and 2.1 m in diameter. It spans from a
 372 radius of 31 mm to a radius of 1082 mm
 373 from the beam pipe, and it is composed
 374 of a barrel and two end-caps, to cover a pseudorapidity region of $|\eta| < 2.5$. The ID
 375 provides tracking and momentum measurements for charged particles.

376
 377 The main components of the Inner Detector are three sub-detectors:

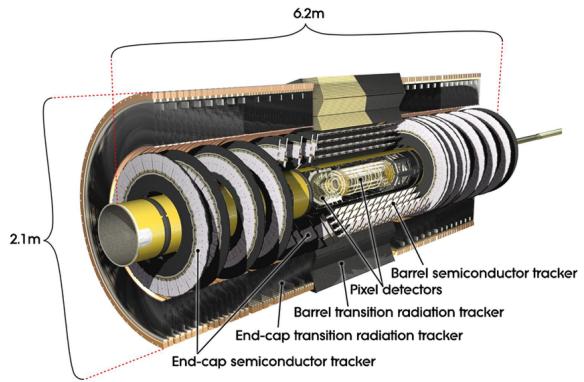


Figure 1.6: The Inner Detector

- 378 • **The Pixel Detector:** it is the first detector crossed by the particles produced
 379 in the pp collisions. The system consists of three barrels (located respectively
 380 at 50.5 mm, 88.5 mm and 122.5 mm from the beam line) and three disks
 381 on each side. A fourth layer (called Insertable B-layer) [13] was placed at
 382 a distance of 31 mm from the beam line in the barrel during the shutdown
 383 between Run1 and Run2. The Pixel Detector contains 1744 modules, each of
 384 which is a 62.4 mm \times 21 mm silicon wafer containing 46080 pixel elements.
 385 This is to provide high granularity, high precision sets of measurements as
 386 close to the interaction point as possible.

387 The intrinsic hit position resolution in the Pixel detector is $\sim 12 \mu\text{m}$ in the
 388 $R \cdot \phi$ direction and varies from 72 to 115 μm in the z direction, depending on
 389 the layer.

- 390 • **The Semi-Conductor Tracker (SCT):** it is a silicon microstrip tracker,
 391 made up of 4088 modules and over 6 million implanted readout silicon strips.
 392 In the barrel the sensors are organized in four concentric layers, placed at a
 393 distance from the beam line between 299 mm and 514 mm.

394 The intrinsic hit position resolution is $\sim 16 \mu\text{m}$ in the $R \cdot \phi$ direction and
 395 $580 \mu\text{m}$ in the z direction

- 396 • **The Transition Radiation Tracker (TRT):** it is the outer layer of the
 397 Inner Detector. The detecting elements are drift tubes (straws), filled with gas
 398 which becomes ionized when a charged particle hits the straw. The oppositely
 399 charged ions are driven by a fixed voltage to the opposite ends of the straws,
 400 where they generate a hit signal; this allows the reconstruction of the particle
 401 track. The TRT has a poorer spatial resolution than the Pixel Detector and
 402 the SCT, and can only provide measurements in the $R \cdot \phi$ direction, with a
 403 resolution of $\sim 130 \mu\text{m}$, but it can provide a large number of hits per track.
 404 Furthermore, the straws are surrounded by materials with a different refraction
 405 index, which causes ultra-relativistic particles ($\gamma \gg 1$) to produce transition
 406 radiation. This effect can be employed to discriminate electrons from pions.

The overall momentum resolution achieved in Run1 with the ID was:

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

407 The Calorimeters

408 The ATLAS calorimeters [14, 15], shown
 409 in Figure 1.7, have the task of measur-
 410 ing the energy of all charged and neu-
 411 tral particles (except for muons, which
 412 are highly penetrating, and neutrinos).
 413 The calorimeters cover the range $|\eta| <$
 414 4.9, using different techniques, suited
 415 to the widely varying requirements of
 416 the physics processes of interest over
 417 this large η range. Over the η region
 418 matched to the inner detector, the fine
 419 granularity of the EM calorimeter is
 420 ideal for precision measurements of elec-
 421 trons and photons. The coarser granularity of the rest of the calorimeter is sufficient
 422 to satisfy the physics requirements for jet reconstruction and E_T^{miss} measurements.

423 Calorimeters must provide good containment for electromagnetic and hadronic
 424 showers, and must also limit punch-through into the muon system. Calorimeter
 425 depth is thus an important design consideration. The total thickness of the elec-
 426 tromagnetic calorimeter is > 22 radiation lengths in the barrel and > 24 radiaton
 427 lengths in the end-caps. The thickness of the hadronic calorimeters is ~ 8 interac-
 428 tion lengths, which provides, including the material in front of it, a total of ~ 10
 429 interaction lengths.

430 There are three calorimeters in the ATLAS detector:

- 432 • **The Electromagnetic Calorimeter (EM)** is a lead-LAr detector, with
 433 accordion-shaped (see Figure 1.8) copper electrodes and lead absorber plates
 434 over its full coverage. It is 6.65 m long and has an outer radius of 2.25 m. The

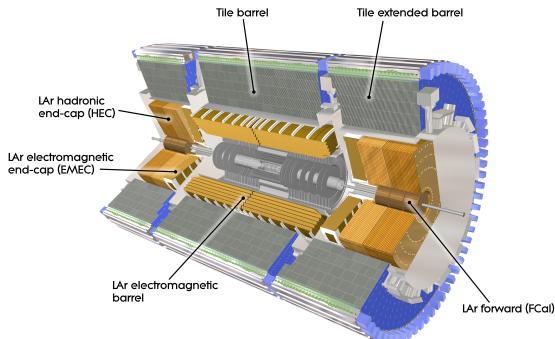


Figure 1.7: The Calorimeters System

435 accordion geometry provides complete ϕ coverage without azimuthal cracks,
 436 while also allowing for fast extraction of the signal at the rear or at the front
 437 of the electrodes. The calorimeter is divided into a barrel part ($|\eta| < 1.475$)
 438 and two end-cap components ($1.375 < |\eta| < 3.2$), each housed in their own
 439 cryostat. In the region of $|\eta| < 1.8$, a presampler detector is used to correct
 440 for the energy lost by electrons and photons upstream of the calorimeter. The
 441 presampler consists of an active LAr layer of thickness 1.1 cm in the barrel and
 442 0.5 cm in the end-caps. The EM calorimeter is segmented into 190000 cells
 443 along η and ϕ , and it is composed of four layers (including the presampler)
 444 to record the longitudinal development of the electromagnetic shower. The
 445 first layer after the presampler has the finest granularity (see Table 1.1), with
 446 $\Delta\eta$ up to 0.003; this fine segmentation allows to discriminate between prompt
 447 photons⁴ and two almost collinear photons which are the decay products of π^0
 pions.

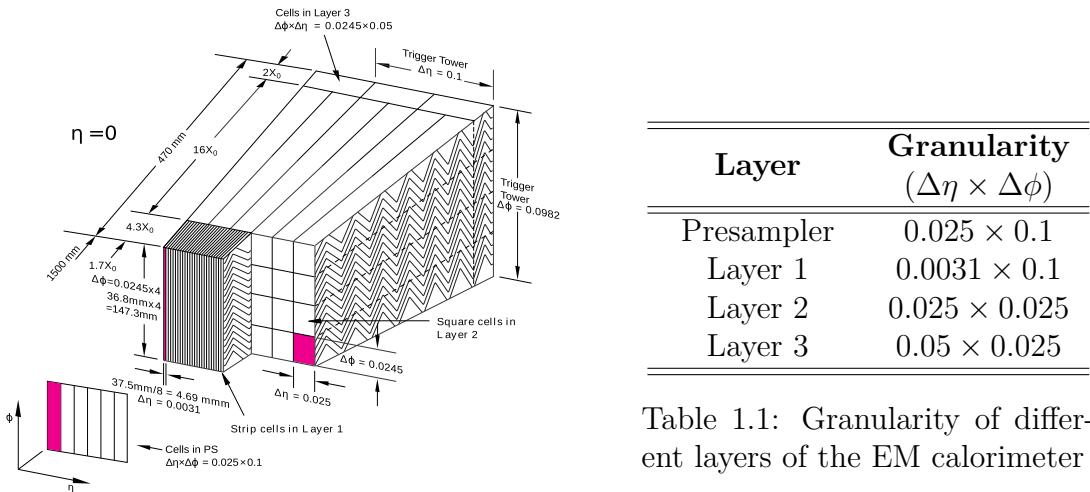


Figure 1.8: The EM calorimeter accordion geometry

- 448
- **The Hadronic Calorimeter (Had)** is 6.1 m long, has an outer radius of 4.25 m and covers the pseudorapidity region $|\eta| < 3.2$. It is composed of two different parts: the Tile calorimeter, and the LAr hadronic end-cap calorimeter. The first one covers the region up to $|\eta| < 1.7$, and is a sampling calorimeter using steel as the absorber and scintillating tiles as the active material. The second one consists of two independent wheels per end-cap, located directly behind the EM calorimeter, covering the region up to $|\eta| < 3.2$. Copper plates act as the absorber, and the active medium is liquid Argon.

- 450
- **The LAr Forward Calorimeters (FCal)** are integrated into the end-cap 451 criostats. They start at $|\eta| = 3.1$ and bring the pseudorapidity coverage of 452 the ATLAS calorimetric system to $|\eta| < 4.9$. The close proximity between the 453 calorimeters minimises energy losses and limits the background which reaches 454 the muon system. Each FCal is split into three modules in the longitudinal 455 direction: an electromagnetic module (FCal1) and two hadronic modules (FCal2)

⁴A prompt photon is produced directly at the pp collision point, rather than by the decay of other particles in the detector

463 and FCal3). The electromagnetic module consists of a copper matrix in which
464 copper rods are inserted. A gap filled with liquid Argon between the rods and
465 the matrix is used as active material. In the hadronic sections the copper is
466 replaced by tungsten.

467 The Muon Spectrometer

468 The Muon Spectrometer [16], shown in
469 Figure 1.9, forms the outer part of the
470 ATLAS detector and is designed to de-
471 tect charged particles exiting the barrel
472 and end-cap calorimeters and to mea-
473 sure their momentum in the pseudora-
474 pidity range $|\eta| < 2.7$.

475 The magnetic field provided by the
476 toroid magnets bends the muon trajec-
477 tories, allowing precision measurements
478 of the muons momentum. The mo-
479 mentum measurement is carried out by
480 muon chambers of two types: Monitored
481 Drift Tubes (MDTs) in the barrel and Cathode Strip Chambers (CSCs) in the end-
482 cap. The MDTs have a momentum resolu-
483 tion between 2-3% and $\sim 10\%$ in a p_T
range between 10 GeV and 1 TeV.

484 Two additional types of muon chambers are present in the spectrometer: the
485 Resistive Plate Chambers (RPCs) and the Thin Gap Chambers (TGCs). These
486 provide fast triggering on muon tracks respectively in the barrel region and in the
487 end-caps.

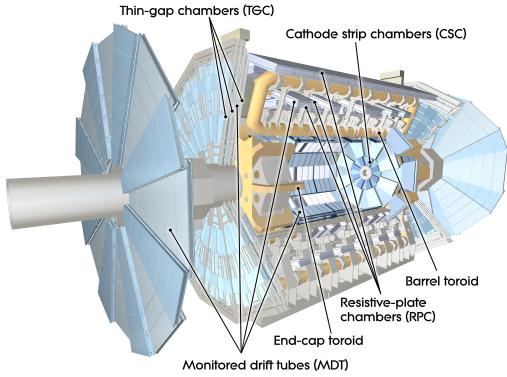


Figure 1.9: The muon spectrometer

488 1.2.2 The Trigger System

489 The LHC provides to the ATLAS detector an unmanageable amount of data. An
490 efficient and extremely selective trigger system is thus fundamental for the detec-
491 tor operation. The ATLAS trigger consists of two levels of event selection: the
492 Level-1 (L1) and the High-Level Trigger (HLT).

493 The L1 trigger is implemented using custom-made electronics, while the HLT
494 is almost entirely based on commercially available computers and networking hard-
495 ware.

496 The L1 Trigger

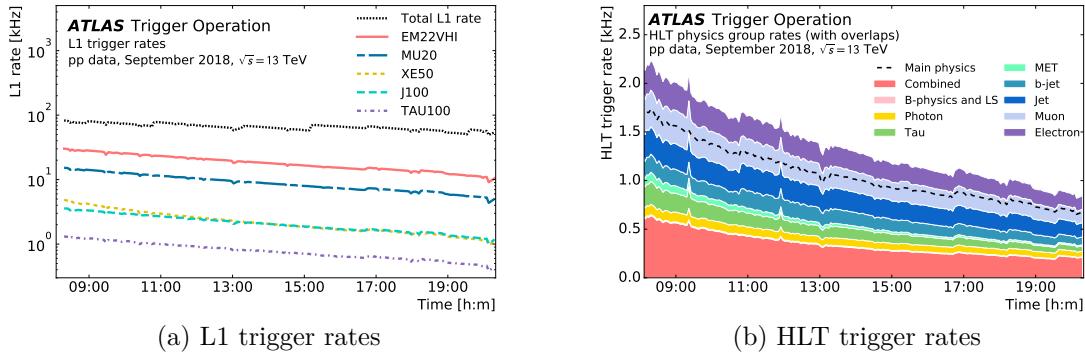
497 The L1 trigger searches for signatures from high- p_T muons, electrons, photons, jets
498 and τ -leptons decaying into hadrons. It also selects events with large missing trans-
499 verse energy (E_T^{miss}), and large total transverse energy. The trigger uses reduced-
500 granularity information from a subset of detectors: the RPCs and TGCs from the
501 muon detectors, and all the calorimeters sub-system.

502 The L1 trigger also defines *Regions of Interest* (RoI's). These are the regions
503 where the L1 trigger has identified possible trigger objects within the event; infor-
504 mation on coordinates, energy and type of signature are also included in the RoI.

505 **The HLT**

506 The HLT trigger is seeded by RoI's: L2 selections use, at full granularity and precision,
 507 all the available detector data within the RoI's. Feature Extraction algorithms
 508 are used to identify objects and determine their properties or to determine the global
 509 characteristics of the event.

510 Offline analysis procedures on fully-built events are used to further select events
 down to a rate of approximately 1000 Hz.



511 Figure 1.10: Trigger rates at the ATLAS detector

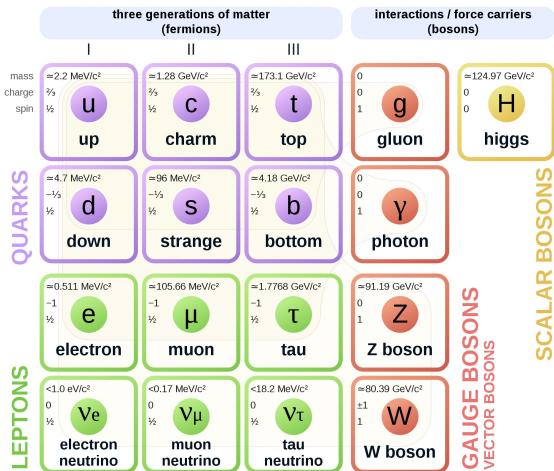
512 **1.2.3 Search for the Standard Model Higgs Boson**

513 The Standard Model (SM) is the theoretical framework upon which the work
 514 in this thesis is based. It describes
 515 the fundamental particles which com-
 516 pose matter and their interactions.

517 The SM describes matter by means
 518 of 12 elementary particles (Figure 1.11),
 519 each of which has its anti-particle coun-
 520 terpart, mathematically described by
 521 fermionic fields.

522 The 12 particles are grouped into
 523 two categories, depending on the types
 524 of interactions they are subject to:
Quarks, which interact through Strong
 525 and Electroweak forces, and *Leptons*,
 526 which only interact through the Elec-
 527 troweak force. Each category is further
 528 divided into three *Generations*.

529 Quarks have the peculiarity of always being bound together to form composite
 530 particles (called *Hadrons*). Hadrons can be composed of either three (*Baryons*) or
 531 two (*Mesons*) quarks. If an attempt is made to separate quarks, the great increase
 532 of strong force potential energy leads to the formation of new quarks, in a process
 533 called *Hadronization*. The newly formed quarks combine with the separated quarks,
 534 and generate streams of Hadrons, called *Jets*.



535 Figure 1.11: Particles in the Standard
 Model

537 The mathematical description of particles in the SM implies that gauge and mat-
 538 ter fields should be massless. The mass term for gauge bosons and fermions would
 539 break the local symmetries upon which the SM is based. Experimental evidence,
 540 though, shows that this is not the case: both Fermions and the W/Z bosons do have
 541 a nonzero mass. The introduction of a spontaneous symmetry breaking mechanism
 542 [17, 18], known as the *Higgs Mechanism*, solves this inconsistency. Particles gain
 543 mass by interacting with the Higgs field, to which a new scalar boson is associated:
 544 the Higgs boson.

545 Production

546 The production of the Standard Model Higgs boson at the LHC is expected to
 547 proceed mainly through four processes [2] (see Figure 1.12):

- 548 • **Gluon-gluon fusion** through a top-quark loop (Figure 1.12a) is the dominant
 549 production channel for all masses.
- 550 • **Vector boson (WW, ZZ) fusion** (Figure 1.12b) becomes increasingly im-
 551 portant with increasing Higgs boson masses, attaining a cross-section similar
 552 to that of gluon-gluon fusion for $m_H \sim 1$ TeV. This process leads to the very
 553 distinctive signature of two jets emitted at small angle with the beam axis
 554 (called *forward jets*).
- 555 • **Associated Higgs production with a $t\bar{t}$ pair** (Figure 1.12d) or a **W/Z
 556 boson** (Figure 1.12c) has a significantly smaller cross-section, but gives rise to
 557 final states which are relatively easy to extract from the background, thanks
 558 to the additional signature (for instance leptons) produced in the decay of the
 559 accompanying particles.

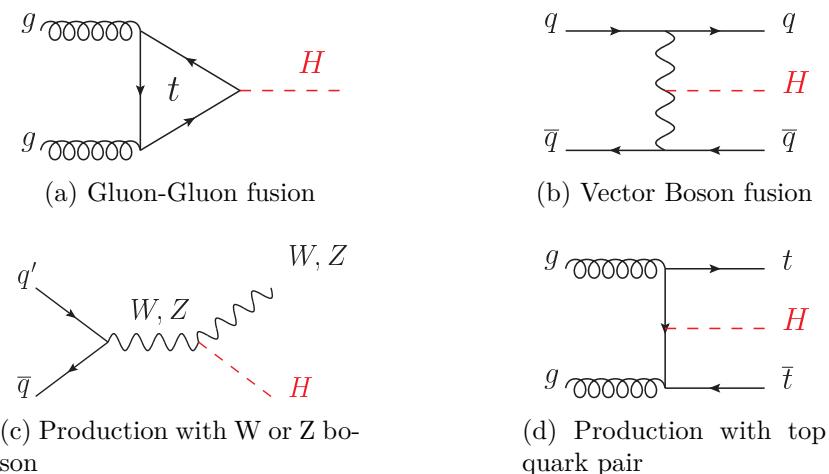


Figure 1.12: Feynman diagrams for the main production processes of the Higgs boson at LHC

560 The mass of the Higgs boson is not determined by the theory, and its production
 561 cross section depends on M_H , as shown in Figure 1.13.

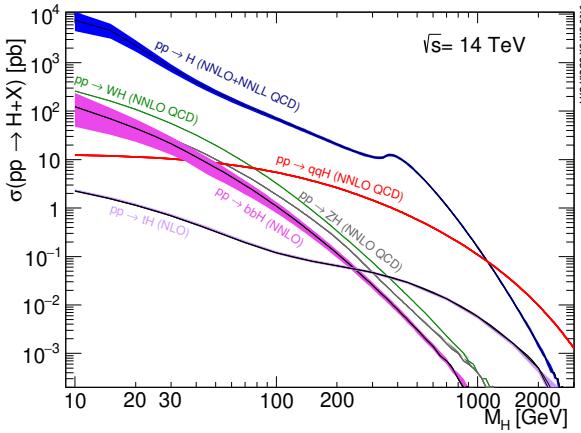


Figure 1.13: Higgs boson production cross section for different M_H

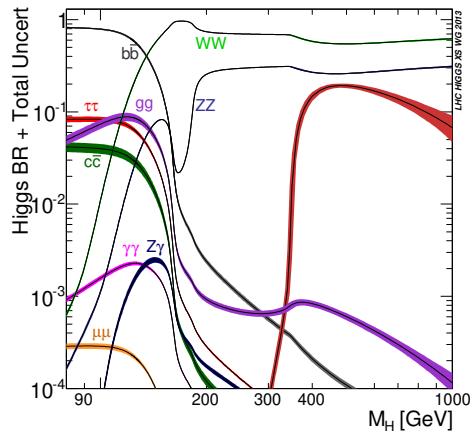


Figure 1.14: Branching ratios for different decay processes of the Higgs boson

562 Decays and Search strategy

563 Hadronic events are the most copious final states expected from a Standard Model
 564 Higgs boson production and decay (see Figure 1.14). However, as mentioned in
 565 Section 1.1.4, these events cannot be extracted from the QCD background, and
 566 topologies with leptons and photons have to be exploited instead, despite the smaller
 567 branching ratio.

568 The main channels that are used at the LHC to look for the Higgs boson are the
 569 following:

- 570 • Low mass region ($m_H < 130$ GeV). Two decays mode are experimentally
 571 important in this region: $H \rightarrow b\bar{b}$ and $H \rightarrow \gamma\gamma$. The first one has a branching
 572 ratio close to 100%, but cannot be observed due to the QCD background. If
 573 the Higgs boson production is associated with $t\bar{t}$, Z or W , the additional lepton
 574 coming from the decay of the accompanying particles can provide a signature
 575 useful to extract the events from the background, despite the significantly
 576 smaller cross-section of the associated production.
- 577 • Intermediate mass region ($130 \text{ GeV} \leq m_H \leq 2 m_Z$). The most promising chan-
 578 nels for the experimental search are $H \rightarrow WW^* \rightarrow \ell\nu\ell\nu$ and $H \rightarrow ZZ^* \rightarrow 4\ell$.⁵
- 579 • High mass region ($m_H > 2 m_Z$). This is the best region to discover a Higgs
 580 signal at the LHC, since the $H \rightarrow ZZ \rightarrow 4\ell$ gives rise to a almost background
 581 free signature.

582 **The $H \rightarrow \gamma\gamma$ channel:** This channel allows the
 583 observation of a Higgs boson over the mass range
 584 $80 \leq m_H \leq 150$ GeV. The small branching ratio is
 585 due to the fact that the $H_{\gamma\gamma}$ coupling is forbidden at
 586 the tree level and therefore the $H \rightarrow \gamma\gamma$ decay occurs
 587 only at higher order through a W loop (see Figure
 588 1.15).

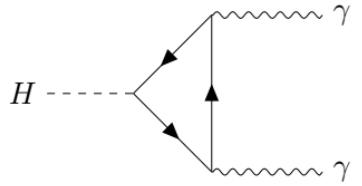


Figure 1.15: $H \rightarrow \gamma\gamma$ decay

⁵The ℓ symbol denotes electrons and muons

589 The final state consists of two high- p_T photons ($p_T \sim 60$ GeV) with invariant
 590 mass compatible with the Higgs boson mass. Despite the simple signature, this is
 591 the most challenging channel for the LHC electromagnetic calorimeters. The reason
 592 is that there are two large backgrounds to fight:

- 593 • $\gamma\gamma$ production, which is an *irreducible background* (it gives rise to exactly the
 594 same final state as the signal). It has no resonant structure and decreases
 595 smoothly with the invariant mass of the two photons. The $\gamma\gamma$ cross-section is
 596 60 times larger than the $H \rightarrow \gamma\gamma$ cross-section in the region $m_{\gamma\gamma} \sim 100$ GeV.
- 597 • γj and jj production, where one or both jets fake a photon. In general a jet
 598 consists of many particles, and therefore it can be easily recognised from a
 599 single photon for instance from the broader shower size in the calorimeters, or
 600 from the presence of several tracks in the inner detector associated with the
 601 calorimeter shower. There are however rare cases in which a jet of particles
 602 can fake a single photon, and since the cross section for jet production is
 603 $\sim 10^6$ times larger than the one for the $\gamma\gamma$ continuum a large jet rejection is
 604 necessary.

605 The $\gamma\gamma$ background is by definition irreducible, but a resonant signal can be
 606 observed over the $\gamma\gamma$ continuum if the peak is narrow enough. The signal significance
 607 is approximated by:

$$S = \frac{N_s}{\sqrt{N_b}} \quad (1.4)$$

608 where N_s and N_b are the number of detected signal and background events. Since
 609 the Higgs boson resonance is very narrow (~ 4 MeV), the width of the reconstructed
 610 peak is solely dependent on the energy resolution of the EM calorimeter and on the
 611 ability to precisely reconstruct the photon's trajectory angle.

612 Discovery

A particle compatible with the Higgs boson was observed in 2012 by the ATLAS [19] and CMS [20] collaborations. In the following years, further analyses [21] lead to the characterization of the Higgs boson. The final result for the measurement of the Higgs boson mass is:

$$m_H = 125.05 \pm 0.24 \text{ GeV}$$

613 Other properties have been identified,
 614 such as the Higgs boson spin ($J = 0$)
 615 and parity ($P = 0$). Furthermore the
 616 observation of the $H \rightarrow \gamma\gamma$ decay chan-
 617 nel proved that the Higgs boson must be
 618 a even eigenstate of charge conjugation
 619 ($C = 1$).

620 The compatibility with the theoreti-
 621 cal SM predictions is measured by the

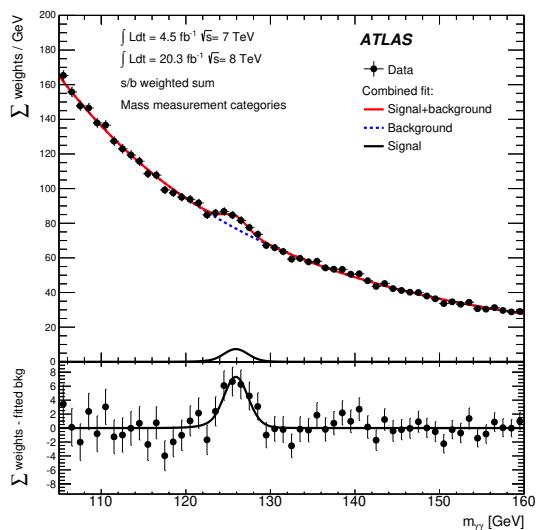


Figure 1.16: Invariant mass distribution for the $H \rightarrow \gamma\gamma$ analysis, showing a resonant peak at 125 GeV

proportion between the theoretical production cross section and branching ratios and the relative measured quantities. To be compatible with the SM these quantities should be equal to 1. The only value which can be obtained from data is the number of selected event, which depends on the product of cross section and branching ratio: $N_{H \rightarrow \gamma\gamma} = \mathcal{L} \cdot \sigma_H \cdot \text{BR}_{\gamma\gamma}$. The ratio between the theoretical and measured number of events was measured as $\mu = 1.09^{+0.11}_{-0.10}$, thus confirming the compatibility.

628 Low Mass

Many SM extensions [22] predict resonances decaying into two photons. The search for narrow resonances X of mass m_X in the low-mass range of 65 to 119 GeV exhibits two peculiarities compared to searches at higher invariant masses.

A first effect to consider is that selecting photon candidates with transverse energies close to the trigger threshold sculpts the invariant mass spectrum at its lower edge, requiring careful fitting of the background model.

Secondly, in addition to the continuous falling background composed of $\gamma\gamma$ as well as γj and jj with misidentified jets, a Drell-Yan (DY) [23] component is present which originates from the $Z \rightarrow e^+e^-$ decay, where the electrons are both reconstructed as photons. Due to the large Bremsstrahlung that these electrons typically undergo, the reconstructed invariant mass of this background exhibits a peak just left of the Z boson mass.

Rejection of electrons faking photons is thus especially important in this search range, but methods that do not compromise the photon efficiency have to be employed to detect the small signals over the irreducible $\gamma\gamma$ background.

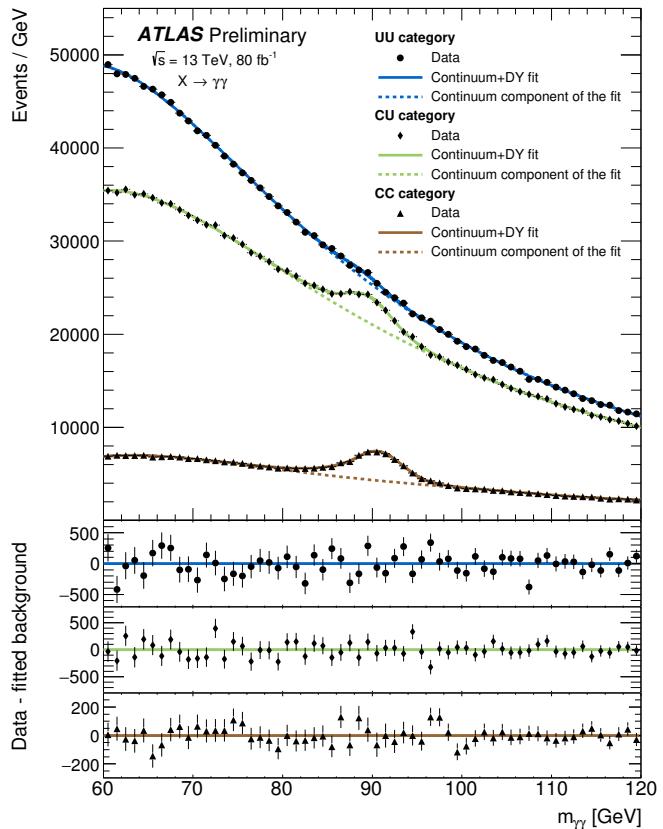


Figure 1.17: Low Mass $m_{\gamma\gamma}$ distribution. The DY background is clearly visible

659 CHAPTER 2

660 **Electron and Photon**
661 **Reconstruction**

662 The reconstruction of electrons and photons in the ATLAS detector [24] is based on
663 the reconstruction of clusters of energy in the EM calorimeter and of tracks in the
664 inner detector. In general an electron is defined as an object consisting of a cluster
665 built in the calorimeter with a track pointing to it (*matched*), a converted photon¹
666 is a calorimeter cluster matched to a conversion vertex, and an unconverted photon
667 is a cluster matched to neither an electron track nor a conversion vertex.

668 Since electrons and photons interact with the ATLAS detector in a similar way
669 (see Figure 2.1), the respective reconstruction algorithms run in parallel. Objects
670 may consequently be reconstructed as both an electron and a photon; an ambiguity
671 resolution is performed to remove part of the overlap. However, some overlap is
672 allowed to remain, in order to maintain a high reconstruction efficiency for both
673 electrons and photons, to which physics analysis may apply their own discrimination
674 criteria.

675 The final electrons and photons are then built and calibrated with dedicated
676 calibration coefficients.

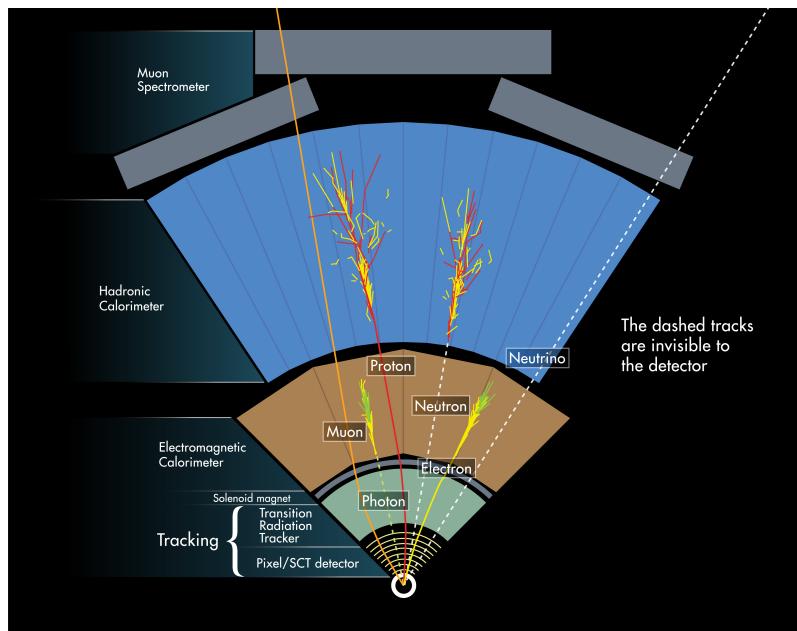


Figure 2.1: The different ways particles interact with the ATLAS detectors

¹The difference between a converted photon and an unconverted photon is that the former materializes into an electron-positron pair interacting with the inner detector material

677 2.1 Energy Clusters and Track reconstruction

678 The reconstruction of photons and electrons begins in the same way for the two
679 particles: energy clusters are constructed in the EM calorimeter and hits in the
680 inner detector are fitted to reconstruct tracks.

681 2.1.1 Energy Clusters in the EM calorimeter

682 Over time two different algorithms have been used to reconstruct the particles energy
683 deposition in the calorimeter cells. Up to 2017 a *sliding window* algorithm [25] was in
684 use, while a *topological clustering* algorithm [24], which allows for large improvements
685 in energy resolution, has been adopted since then.

686 For the analysis discussed in this thesis the full Run2 dataset (2015 - 2018) was
687 reconstructed using the topological clustering algorithm.

688 Sliding Window algorithm

689 The algorithm in use up to 2017 is based on a rectangular window of fixed size,
690 which is shifted around the EM calorimeter to identify energy deposit of physical
691 interest.

692 The algorithm operation can be summarized in the following three steps:

693 • **Tower building.** The EM calorimeter is segmented into a grid (in the $\eta -$
694 ϕ directions) of $N_\eta \times N_\phi$ towers of size $\Delta\eta^{tower} \times \Delta\phi^{tower} = 0.025 \times 0.025$
695 (which is the same granularity of the cells in the second layer of the EM
696 calorimeter). Each tower is made up of several cells in the three layers of the
697 calorimeter. The sum of the energies of the cells in each tower (the *tower*
698 *energy*) is computed, making sure to divide the energy of cells belonging to
699 multiple towers proportionally to the fractional area of the cell inside each
700 tower.

701 • **Seed cluster finding.** A window composed of a fixed number of towers
702 ($N_\eta^{window} \times N_\phi^{window} = 3 \times 5$) is moved along the tower grid. The energy of the
703 window is calculated as the sum of the transverse energies of the 15 towers
704 contained in the window. The 3×5 area where the energy reaches a local
705 maximum are identified as the seed clusters (if the total energy is above a
706 threshold of 2.5 GeV). The (η, ϕ) coordinates of the seed cluster are defined
707 as the energy-weighted barycenter of the cells within a 3×3 window around
708 the tower at the center of the sliding window.

709 • **Cluster filling.** A cluster is built around the seed position in each layer,
710 including cells within a rectangle of size $N_\eta^{cluster} \times N_\phi^{cluster} = 3 \times 7$ in the barrel
711 and 5×5 in the end-caps. The layers are processed in sequence: firstly the
712 L2 layer cluster is built around the seed cluster position. A new weighted
713 barycenter, computed using only the energy deposit in L2 is used as seed
714 position to define clusters in L1 and L3. The PreSampler cluster is then
715 created around the barycenter of energy deposits in L1.

716 **Topological clustering algorithm**

717 Using dynamic, variable sizes clusters offers numerous advantages over a method
718 using fixed-size cluster. A primary motivation is the ability to recover low energy
719 photons radiated by brehmsstrahlung in the Inner Detector, and to connect them
720 to their associated electron or converted photon, forming what might be called
721 *superclusters*. Another important motivation for using topo-clusters is that they
722 grow in a similar fashion to the natural development of electromagnetic showers in
723 the calorimeter.

724 The topo-cluster algorithm follows these steps:

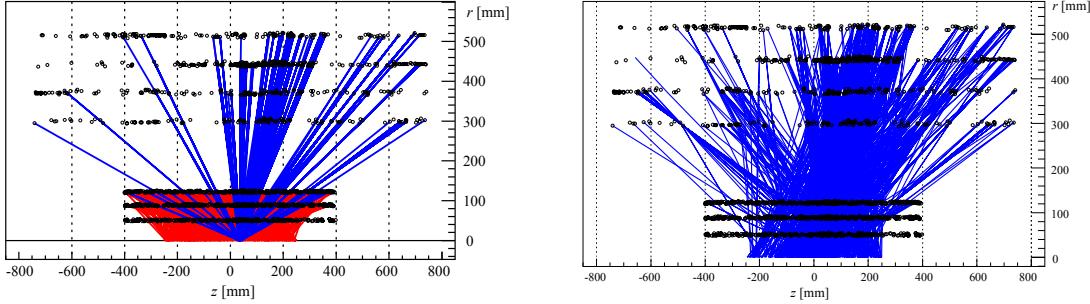
- 725 • Proto-clusters are created in a three step process known as “4-2-0 topo-cluster
726 reconstruction”:

- 727 – A significance value defined as $\zeta_{cell}^{EM} = \left| \frac{E_{cell}^{EM}}{\sigma_{noise, cell}^{EM}} \right|$ (where E_{cell}^{EM} is the total
728 cell energy and $\sigma_{noise, cell}^{EM}$ is the expected cell noise) is computed for each
729 cell in the calorimeter.
- 730 – *Proto-clusters* are formed using cells with $\zeta_{cell}^{EM} \geq 4$ as seeds. In this initial
731 stage, cells from the presampler and the L1 are excluded from initiating
732 proto-clusters, as this suppresses the formation of noise clusters. The
733 cells which pass the selection are included in a *seed list*.
- 734 – The algorithm then collects all cells neighbouring those in the seed list,
735 with significance $\zeta_{cell}^{EM} \geq 2$ and adds them to the corrisponding proto-
736 cluster. If a cell which satisfies the significance requirement is adjacent
737 to more than one proto-cluster, the two clusters are merged. The selected
738 cells are included in a *neighbour cells list*, which takes over as the seed
739 list. This process is iterated until the seed list is empty, i.e. until there
740 are no cells near a proto-cluster with significance ≥ 2 .
- 741 – Lastly a final set of neighbouring cells with $\zeta_{cell}^{EM} \geq 0$ is added to the
742 clusters.

- 743 • Proto-clusters containing more than one local maximum can be split. A local
744 maximum is defined as a cell with $E_{cell}^{EM} > 500$ MeV, with at least four neigh-
745 bors, none of which has a larger E_{cell}^{EM} value. Cells can thus be shared by two
746 proto-clusters, in which case the energy of the cell is divided among the two
747 depending on the energies of the clusters.
- 748 • The proto-clusters are not limited to the cells of the EM calorimeter, but could
749 include cells from the Hadronic Calorimeter as well. A selection is made on
750 the fraction of the cluster energy recorded in the EM calorimeter, to ensure
751 that the energy deposit in the calorimeters is mainly due to electromagnetic
752 showers.

753 **2.1.2 Track reconstruction and matching**

754 After the energy cluster building the electron and photon reconstruction algorithm
755 proceeds to the second step: searching for a track in the Inner Detector which can
756 be matched to the energy clusters.



(a) Constrained seed search: red two-SpacePoints tracks are used to determine vertex position and constrain blue three-SpacePoints seed tracks

(b) Unconstrained seed search: a notable increase of the number of track candidates is seen

Figure 2.2: Track seeds for the same event with and without z-vertex-constraint

757 Track reconstruction [26, 27] begins with fitting the hits left by charged particles
 758 in the ID. This process is done in an inside out manner:

- 759 • **SpacePoint creation.** The first step in the track reconstruction is the cre-
 760 ation of three-dimentional representation of the silicon detector measurements.
 761 Pixel clusters provide local two-dimensional positions on a fixed module surface
 762 and therefore map directly into space points. On the contrary, SCT clusters
 763 do not transform directly into space points because a single strip provides
 764 a precise measurement in only one direction (orthogonal to the strip itself).
 765 Instead, space points are formed by combining the information from pairs of
 766 clusters from a SCT module.
- 767 • **Track seeds creation.** Sets of three Space Points created in the previous
 768 step are used as track seeds. The track seeds can be searched with a z *vertex*
 769 *constraint* (Figure 2.2a): pairs of SpacePoints from the pixel detector are
 770 found in a first step and z interaction vertices are built from these pairs; a
 771 fast primary vertex search is performed and the best fitting trimary vertex is
 772 used to constrain the seeds with three space points. As of now, though, the
 773 default is an unconstrained seed search (Figure 2.2b), which clearly leads to a
 774 significantly higher number of initial track seeds; the unconstrained search is
 775 evidently time consuming, but more efficient in finding tracks in events with
 776 loosely constraint primary vertices.
- 777 • **Track candidates finding.** After the seeds search has terminated, the *road*
 778 *building* process is started: seeds provide enough directional information to
 779 build roads of detector elements for the further search of hits associated to one
 780 track candidate. A Kalman [28] fitter-smoother algorithm is used to include
 781 successive hits in the track candidate fit. Only about 10% of track seeds are
 782 successfully extended into a track candidate.
- 783 • **Ambiguity resolution.** After the last step a large number of track candidates
 784 are found, which have to be resolved before extending the tracks to the TRT
 785 detector. Many of the track candidates share hits, are incomplete or describe

fake tracks². A *track scoring* strategy has been developed [29], that associates to morphologic parameters of the track a beneficial or penalty score, which in turn form an overall track score (Table 2.1 gives an overview of the different benefits and penalties of tracks). After the scoring has happened, hits that are shared between tracks are assigned to the track with higher score, while the remaining track is refitted without the formerly shared hit, and then scored again.

In an iterative procedure, the tracks with the highest score are bundled, and tracks that fall beyond a certain quality cut are neglected.

Track Characteristics	Detector	Effect on the track score
B layer hole	pixel	strong penalty
Layer hole	pixel	penalty
Overlap hit	pixel, SCT	strong benefit
Sensor hole	SCT	weak penalty
Layer hole (module)	SCT	strong penalty

Table 2.1: Track characteristics effect on track score

- **TRT track extension.** After the ambiguity resolution, tracks are extended with the hits in the TRT detector, improving momentum resolution. If the score of the original track is higher than the one after refitting with the TRT hits, the silicon track is kept and the hits in the TRT detector are rejected.

The inside-out sequence relies on a track seed found in the silicon detector. In the track reconstruction process, some of these initial track seeds may not be found or might even not exist. In particular tracks coming from secondary decay vertices, such as photon conversions, may not have sufficient silicon hits to comply with the inside-out sequence ambiguity processor. A second sequence is thus established in track reconstruction, following an outside-in approach.

Figure 2.3 depicts such approach: red hits show the extensions into the TRT that originate from standard inside-out fitting, while black circles show hits that have been associated to TRT segments, which are the starting point for the outside-in sequence.

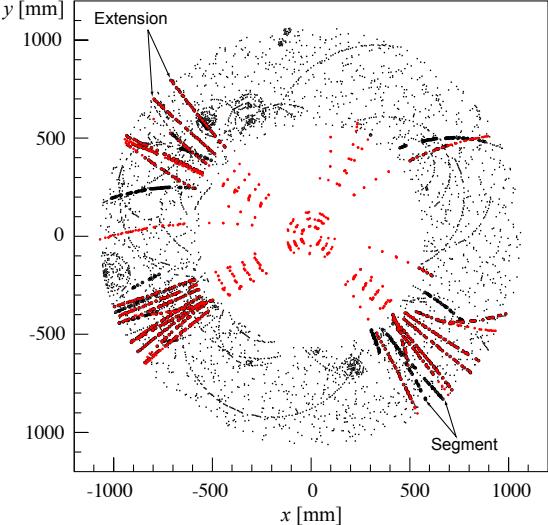


Figure 2.3: Outside-in track fitting

817 Track matching and conversion vertex reconstruction

818 After the tracks have been reconstructed, they are matched to Energy clusters in the
819 EM calorimeter. Clusters are used to create Regions of Interest (ROIs), consisting

²Tracks where the majority of associated measurements do not correspond to a single particle

820 of an area of $\Delta R = 0.3$ around the cluster barycenter, and tracks intersecting
821 the ROIs are considered loosely matched to the cluster. Tracks are then refitted
822 with the global χ^2 fitter [30], and then matched to the EM clusters with stricter
823 requirements. If multiple tracks are matched to a cluster, they are sorted depending
824 on track characteristics (such as number of hits in the Pixel and SCT detectors and
825 the ΔR^3 between the track and cluster coordinates), and the best track is used to
826 determine the particle properties.

827 The reconstructed and loosely matched tracks also serve as input to the con-
828 version vertex reconstruction. Two-track conversion vertices are reconstructed from
829 two tracks forming a vertex consistent with that of a massless particle, while single-
830 track vertices are essentially tracks without hits in the innermost sensitive layers.
831 To increase the converted photon purity, the tracks used to build conversion ver-
832 tices must generally have a high probability to be electron tracks as determined by
833 the TRT [31], especially for single-track vertices and conversion vertices constructed
834 from standalone-TRT tracks.

835 Conversion vertices are matched to a seed cluster by extrapolating the track(s) to
836 the EM calorimeter. In the case of silicon tracks a conversion vertex is successfully
837 matched to a cluster if $\Delta\phi$ and $\Delta\eta$ between the extrapolated track and the cluster
838 barycenter is less than 0.05. Looser requirements are applied to TRT only tracks [32].
839 If there are multiple conversion vertices matched to a cluster, double-track conversion
840 vertices with silicon hits are preferred over other double-track conversions, followed
841 by single-track conversions. Within each category, the vertex with the smallest
842 conversion radius is preferred.

843 The properties of tracks and conversion vertices matched to an EM cluster take
844 part in determining whether a reconstructed particle is an electron, a converted pho-
845 ton or a photon, as described in the following sections.

846 2.1.3 Supercluster formation

847 The reconstruction of electron and photon superclusters proceeds in two stages [33]:
848 firstly topo-clusters (section 2.1.1) are evaluated for use as seed cluster candidates,
849 then clusters near the seed candidates are identified as satellite cluster candidates,
850 which may emerge from brehmsstrahlung or photon conversion. Satellite clusters
851 are added to the seed candidates to form the final superclusters, provided that they
852 fulfill the necessary criteria.

853 The initial list of potential seed topo-clusters is sorted according to descending
854 p_T ; this is done to ensure that higher p_T clusters collect only lower p_T satellites. For
855 a cluster to become an electron supercluster seed, it is required to have a minimum
856 transverse energy of 1 GeV and must be matched to a track with ≥ 4 hits in the
857 silicon tracking detector. For photon reconstruction, a cluster must have a transverse
858 energy greater than 1.5 GeV to qualify as a supercluster seed.

859 Once a cluster is denoted a supercluster seed, the satellite finding stage begins,
860 in which all the unused clusters are examined for association to the seed cluster.
861 This is when the electron and photon reconstruction algorithms diverge. The pro-
862 cess is schematized in Figure 2.4. For both electrons and photons, a window of
863 $\Delta\eta \times \Delta\phi = 0.075 \times 0.125$ (3 \times 5 cells in the L2 of the calorimeter) around the seed
864 cluster barycenter is drawn, and if a cluster falls within this window it is considered

³The angular distance between two points is defined as $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$

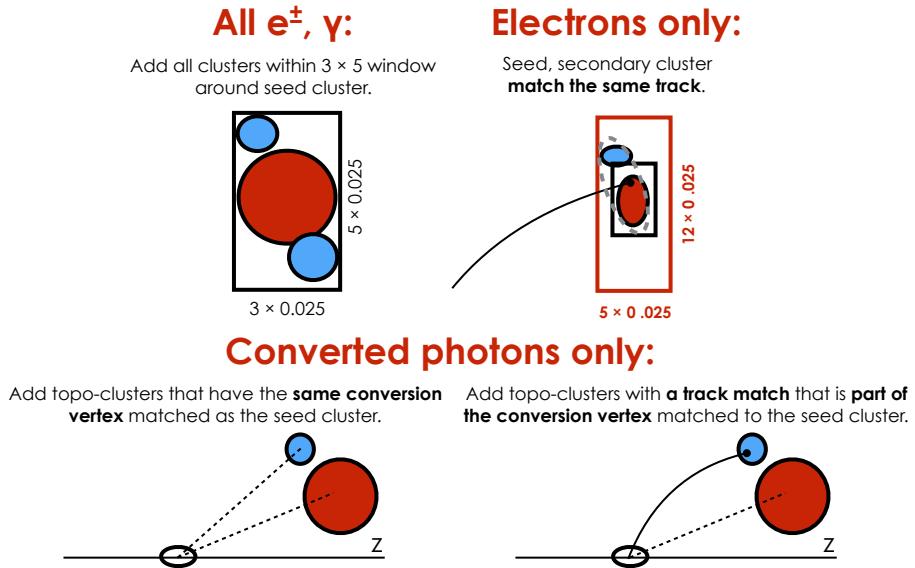


Figure 2.4: Diagram of the superclustering algorithm for electrons and photons

a satellite cluster. For electrons a cluster is also considered a satellite if it sits within a window around the seed cluster barycenter of $\Delta\eta \times \Delta\phi = 0.125 \times 0.300$ (5×12 cells), 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 between distant radiative photons and conversion electrons from pile-up noise or other unrelated clusters.

The seed clusters with their associated satellite clusters are called *superclusters*. 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 (0.125) in the η direction in the barrel (end-cap) region. There is no limit in the ϕ direction, since tracks are naturally bent in this direction by the magnetic field in the ID provided by the solenoid magnet.

2.2 Analysis objects building

As previously mentioned, the photon and electron reconstruction algorithms run in parallel, thus a given seed cluster can produce both an electron and a photon. In such cases an ambiguity resolution procedure is applied (see Figure 2.5): the purpose is that if a particular object can be easily identified only as a photon (an energy 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

892 photon object are created, and marked specifically as ambiguous, allowing the final
 893 classification of these objects to be determined based upon the specific requirements
 894 of each analysis.

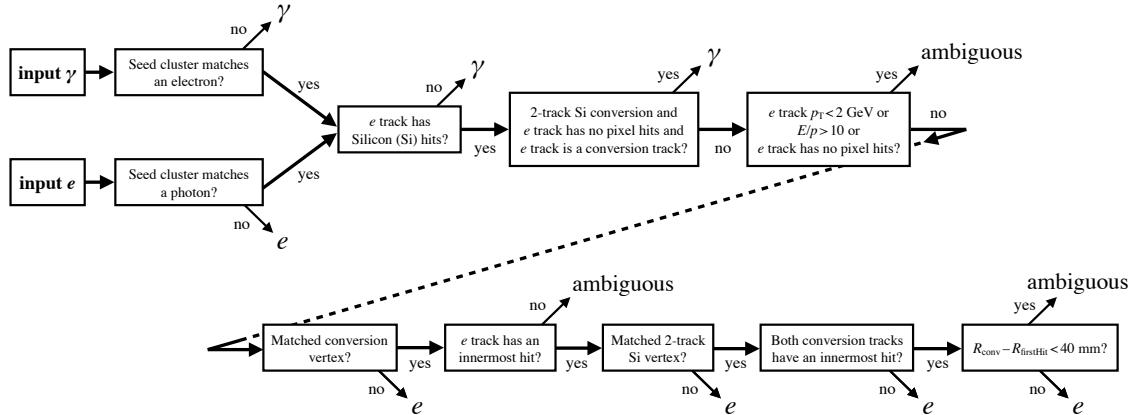


Figure 2.5: Flowchart showing the ambiguity resolution for particles initially described both as an electron and a photon

895 2.2.1 Identification

896 Further quality criteria, called *identification selections*, are used to improve the
 897 purity of the reconstructed electron and photon objects [34, 35]. These are based
 898 upon discriminating variables which are calculated from the reconstructed photons
 899 and electrons. Table 2.2 shows a list of discriminating variables, along with an
 900 indication if they are used for electron or photon identification.

901 Electrons

902 The identification of prompt electrons relies on a *likelihood discriminant*, constructed
 903 from quantities calculated in the ID, the EM calorimeter and the two combined.

904 The variables used can be grouped into properties of the electron track, the lateral
 905 and longitudinal development of the electromagnetic shower in the EM calorimeter,
 906 and the spatial compatibility of the primary electron track with the reconstructed
 907 cluster:

- 908 • The primary electron track is required to fulfil a set of quality requirements:
 909 hits in the two inner tracking layers closest to the beam line, hits in the SCT.
 910 The transverse impact parameter⁴ of the track is also used to construct the
 911 likelihood discriminant.
- 912 • The lateral development of the electromagnetic shower is characterized with
 913 variables calculated separately in the first and second layer of the EM calorime-
 914 ter. To reject clusters from multiple incident particles, $w_{s\,tot}$ is used (see Table
 915 2.2). The lateral shower development is measured with R_ϕ and R_η .

⁴The transverse impact parameter d_0 is the distance of closest approach of the track to the beam line. Its significance $\frac{d_0}{\sigma_{d_0}}$ is also used in the identification process

Category	Description	Name	Usage
Hadronic leakage	Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM cluster (used over the ranges $ \eta < 0.8$ and $ \eta > 1.37$)	R_{had_1}	e/γ
	Ratio of E_T in the hadronic calorimeter to E_T of the EM cluster (used over the range $0.8 < \eta < 1.37$)	R_{had}	e/γ
EM third layer	Ratio of the energy in the third layer to the total energy in the EM calorimeter	f_3	e
EM second layer	Ratio of the sum of the energies of the cells contained in a $3 \times 7 \eta \times \phi$ rectangle (measured in cell units) to the sum of the cell energies in a 7×7 rectangle, both centred around the most energetic cell	R_η	e/γ
	Lateral shower width, $\sqrt{(\sum E_i \eta_i^2)/(\sum E_i) - ((\sum E_i \eta_i)/(\sum E_i))^2}$, where E_i is the energy and η_i is the pseudorapidity of cell i and the sum is calculated within a window of 3×5 cells	w_{η_2}	e/γ
	Ratio of the sum of the energies of the cells contained in a $3 \times 3 \eta \times \phi$ rectangle (measured in cell units) to the sum of the cell energies in a 3×7 rectangle, both centred around the most energetic cell	R_ϕ	e/γ
EM first layer	Total lateral shower width, $\sqrt{(\sum E_i (i - i_{\max})^2)/(\sum E_i)}$, where i runs over all cells in a window of $\Delta\eta \approx 0.0625$ and i_{\max} is the index of the highest-energy cell	$w_s \text{ tot}$	e/γ
	Lateral shower width, $\sqrt{(\sum E_i (i - i_{\max})^2)/(\sum E_i)}$, where i runs over all cells in a window of 3 cells around the highest-energy cell	$w_s 3$	γ
	Energy fraction outside core of three central cells, within seven cells	f_{side}	γ
	Difference between the energy of the cell associated with the second maximum, and the energy reconstructed in the cell with the smallest value found between the first and second maxima	ΔE_s	γ
	Ratio of the energy difference between the maximum energy deposit and the energy deposit in a secondary maximum in the cluster to the sum of these energies	E_{ratio}	e/γ
	Ratio of the energy measured in the first layer of the electromagnetic calorimeter to the total energy of the EM cluster	f_1	e/γ
Track conditions	Number of hits in the innermost pixel layer	$n_{\text{innermost}}$	e
	Number of hits in the pixel detector	n_{Pixel}	e
	Total number of hits in the pixel and SCT detectors	n_{Si}	e
	Transverse impact parameter relative to the beam-line	d_0	e
	Significance of transverse impact parameter defined as the ratio of d_0 to its uncertainty	$ d_0/\sigma(d_0) $	e
	Momentum lost by the track between the perigee and the last measurement point divided by the momentum at perigee	$\Delta p/p$	e
Track–cluster matching	Likelihood probability based on transition radiation in the TRT	$e_{\text{ProbabilityHT}}$	e
	$\Delta\eta$ between the cluster position in the first layer of the EM calorimeter and the extrapolated track	$\Delta\eta_1$	e
	$\Delta\phi$ between the cluster position in the second layer of the EM calorimeter and the momentum-rescaled track, extrapolated from the perigee, times the charge q	$\Delta\phi_{\text{res}}$	e
	Ratio of the cluster energy to the measured track momentum	E/p	e

Table 2.2: Discriminating variables for electron and photon identification (from [33])

- 916 • The longitudinal shower shape variables as well as hadronic leakage are taken
 917 into consideration to discriminate against hadronic showers.
- 918 • The reconstructed track and the EM cluster are matched using $\Delta\eta_1$ and $\Delta\phi_{\text{res}}$.
- 919 A discriminant is formed from the likelihoods for a reconstructed electron to
 920 originate from a signal (L_S) or background (L_B).
 921 The likelihoods are constructed from probability density functions, which are

922 created by smoothing histograms of the discriminating variables.

$$L_{S/B}(\mathbf{x}) = \prod_{i=1}^n P_{S/B}^i(x_i) \quad (2.1)$$

923 where the signal and background probability density functions take the values $P_S^i(x_i)$
924 and $P_B^i(x_i)$ respectively, for the discriminating variable i .

925 The likelihood discriminant d_L is then defined as:

$$d_L = \ln \left(\frac{L_S}{L_B} \right) \quad (2.2)$$

926 Three working points (*Loose*, *Medium* and *Tight*) have been setup, based on the
927 value of d_L : they lead to identification efficiencies of 93%, 88% and 80% respec-
928 tively (averaged over p_T , generally higher p_T electrons have higher identification
929 efficiencies), with increasing background rejection.

930 **Photons**

931 The photon identification criteria are optimized to efficiently select prompt, isolated
932 photons and reject background from hadronic jets. The photon identification is
933 constructed from one-dimentional selection criteria (*cut-based selection*), using the
934 shower shape variables in Table 2.2. The variables using the EM first layer play an
935 especially important role in rejecting π^0 decays into two collimated photons.

936 Similarly to the electron case, three identification selections are available: *Loose*,
937 *Medium* and *Tight*. The Tight selection is the primary one, with the Medium and
938 Loose used for trigger algorithms.

939 The Loose identification criteria uses R_{had} , R_{had_1} , R_η and w_{η_2} shower shape
940 variables. The Medium selection adds a loose cut on E_{ratio} . The Tight identification
941 is performed separately for converted and unconverted photons. The shower shape
942 of converted photons, in fact, differs from the one from unconverted photons due to
943 the opening angle of the e^+e^- conversion pair, amplified by the magnetic field. The
944 Tight selection applies tighter thresholds on the variables used on the Loose menu,
945 and adds informations from the finely segmented strip layer of the calorimeter. It
946 is optimized for seven different intervals in η and E_T , to account for the calorimeter
947 geometry and for different effects on the shower shapes from the material upstream
948 of the calorimeter.

949 **2.2.2 Isolation**

950 Requiring isolation criteria for electrons and photons, allows for a further rejection of
951 background particles. The activity near leptons and photons can be quantified from
952 the tracks of nearby charged particles or from energy deposits in the calorimeters,
953 leading to two classes of isolation variables.

954 The raw calorimeter isolation ($E_{T,raw}^{isol}$) is built by summing the transverse ener-
955 gies of topo-clusters whose barycenter falls within a cone centered around the elec-
956 tron or photon cluster barycenter. The raw calorimeter isolation includes the EM
957 energy of the particle itself ($E_{T,core}$), which is subtracted by removing the energy of
958 the EM calorimeter cells contained in a $\Delta\eta \times \Delta\phi = 5 \times 7$ (cells) rectangular window
959 around the barycenter. Additional leakage and pile-up corrections are parametrized

960 using MC simulation. The fully corrected calorimeter isolation variable is thus com-
 961 puted as [36]:

$$E_T^{coneXX} = E_{T,raw}^{isolXX} - E_{T,core} - E_{T,leakage}(E_T, \eta, \Delta R) - E_{T,pile-up}(\eta, \Delta R) \quad (2.3)$$

962 where XX refers to the size of the opening angle of the employed cone, with $\Delta R =$
 963 $XX/100$. A cone size of $\Delta R = 0.2$ is used for the electron working point, whereas
 964 cone size of $\Delta R = 0.2$ and 0.4 are used for photon working points.

965 The track isolation variable (p_T^{coneXX}) is computed by summing the transverse
 966 momentum of selected tracks within a cone centered around the electron track or
 967 the photon cluster direction. Tracks matched to the electron (or converted photon)
 968 are excluded. Since for electrons produced in the decay of high-momentum heavy
 969 particles, other decay products can be very close to the electron direction, the
 970 track isolation for electrons is defined with a variable cone size ($p_T^{varconeXX}$)⁵ (the
 971 cone shrinks for larger transverse momentum). The considered tracks are required
 972 to have $p_T > 1$ GeV and $|\eta| < 2.5$, at least seven silicon (pixel + SCT) hits,
 973 at most one shared hit⁶, at most two silicon holes and at most one Pixel hole.
 974 In addition, for electron isolation, the tracks are required to have a loose vertex
 975 association, meaning that the track was either used in the primary vertex fit, or
 976 satisfies $|\Delta z_0| \sin \theta < 3$ mm, where $|\Delta z|$ is the longitudinal impact parameter⁷.

977 Electron and photon isolation requirements are based on cuts on these variables
 978 (commonly depending on the energy of the particle, i.e. the cuts are applied on
 979 E_T^{coneXX}/E_T and p_T^{coneXX}/E_T).

⁵Where the cone size is determined by: $\Delta R = \min\left(\frac{10}{p_T[\text{GeV}]}, \Delta R_{max}\right)$, with ΔR_{max} typically equal to 0.2

⁶Shared hits are defined as $n_{Pixel}^{sh} + n_{SCT}^{sh}/2$, where $n_{Pixel/SCT}^{sh}$ are the number of hits assigned to several tracks in the Pixel/SCT detector

⁷The longitudinal impact parameter is the distance along the z axis between the point of closest approach of the track to the beam axis and the reconstructed primary vertex z coordinate

980 CHAPTER 3

981 **Machine Learning and Gradient
982 Boosted Trees**

983 The following chapter presents the general features of the Machine Learing approach
984 to data modelling, as well as describing the concept and charachteristics of a Gra-
985 dient Bosted Tree, and its LightGBM implementation.

986 **3.1 Machine learning**

987 Machine Learning is a subfield of artificial intelligence. A formal definition of algo-
988 rithms studied in the machine learning field was provided by Tom M. Mitchell [37]:
989 “A computer program is said to learn from experience E with respect to some class
990 of tasks T, and performance measure P, if its performance at tasks in T, as measured
991 by P, improves with experience E”.

992 The goal of machine learning algorithms is generally to build a mathematical
993 model of sample data (called *training dataset*), in order to make predictions or
994 decisions without being explicitly programmed to perform the task. Each element
995 of the training set (x_i) is represented by a set of *features*, that the algorithm employs
996 in the learning process.

997 Machine Learning tasks can be separated into two main macro-categories: *su-*
998 *pervised learning* and *unsupervised learning*.

- 999
 - 1000 • In **supervised learning** algorithms, the aim is to create a model that maps
1001 an input variable x (where x usually contains multiple features) to an output
1002 variable y : the algorithm has to find the mapping function $y = f(x)$, with a
1003 fixed *model*¹, for which the machine learning algorithm has to identify optimal
1004 parameters. In such cases the training dataset contains both the input (x_i)
1005 and the correct output (y_i^{true}) variables.
 - 1006 • In **unsupervised learning** the training set has no output variables y : the
1007 algorithm is used to find pattern in data and improve the knowledge of the
1008 dataset.

1008 **3.1.1 Elements of Supervised Learning**

1009 As previously said in supervised learning a fixed model is optimized to fit a set of
1010 training data, by tuning its parameters: the algorithm goal is to find the best pa-

¹The model is the mathematical structure by which the prediction y_i^{pred} is made from the input x_i . A well known example is the linear model: $y_i = \sum_{j \in \mathcal{F}} w_j(x_i)_j$, where \mathcal{F} is the set of features of the training set variables, and w_j are the model parameters

1011 parameters, i.e. the ones which make the model as predictive and stable as possible.
 1012 Another set of variables, called *hyperparameters*, are chosen by the user, and deter-
 1013 mine either fixed characteristics of the model or the behaviour of the optimization
 1014 algorithm.

1015 The predictive performance of the model is evaluated by means of an *Objective*
 1016 *Function*, which has to be minimized [38, 39, 40]. The standard form of an objective
 1017 function is:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \quad (3.1)$$

1018 where Θ represents the set of parameters to be optimized, $L(\Theta)$ is called *Training*
 1019 *Loss* and $\Omega(\Theta)$ is the *Regularization* term.

1020 The Training Loss measures how well the model fits the training data. It can
 1021 take many forms, such as the mean square error $L(\Theta) = \sum_i (y_i^{true} - y_i^{pred})^2$ or the
 1022 cross entropy, which is often used in classification problems:

$$L(\Theta) = \sum_i \left[y_i^{true} \ln \left(1 + e^{-y_i^{pred}} \right) + (1 - y_i^{true}) \ln \left(1 + e^{y_i^{pred}} \right) \right] \quad (3.2)$$

1023 The Regularization term, on the other hand, measures how complex the model is.
 1024 It can take many forms such as $\Omega(\Theta) = \lambda \cdot \|\Theta\|_1$ (the L1 norm of Θ) or $\Omega(\Theta) = \lambda \cdot \|\Theta\|$
 1025 (the regular \mathbb{R}^n L2 norm). The λ hyperparameter, chosen by the user, allows to
 1026 control how relevant the regularization term is.

1027 Containing two components in the objective allows to obtain both a simple and
 1028 predictive model (a concept known as *bias-variance tradeoff*), preventing both un-
 1029 derfitting and overfitting (see Figure 3.1):

- 1030 • Optimizing the training loss encourages predictive models, which fit the train-
 1031 ing data well
- 1032 • Optimizing the regularization encourages simple, stable models, which have
 1033 smaller variance in predictions different than the training set

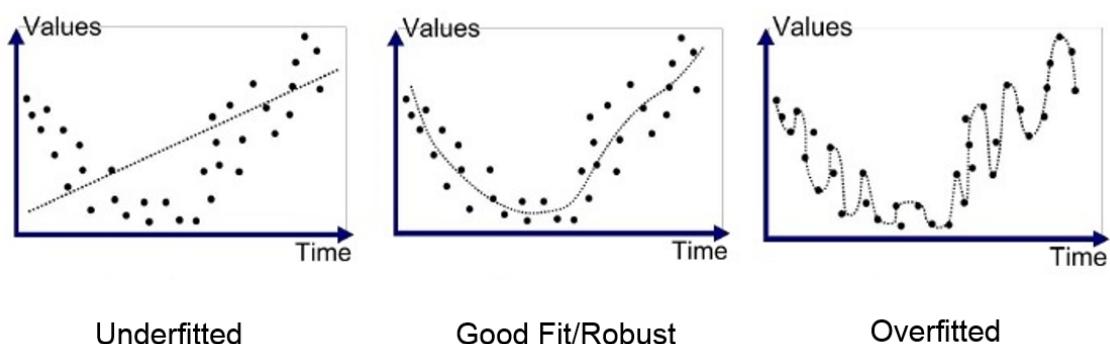


Figure 3.1: Examples of under/over fitting

1034 3.2 Gradient Boosted Trees

1035 One of many types of model which can be used by a machine learning algorithm is a
 1036 Decision Tree ensemble; a trained Decision Tree ensemble takes the name of *Gradient*

1037 *Boosted Tree.* Its basic building block is a Decision Tree, which is described in the
1038 next section.

1039 3.2.1 Decision Trees

1040 A Decision Tree is a flowchart-like structure in which each internal node represents a
1041 “test” on a feature of the input variable x_i (a logical statement, such as a threshold,
1042 whith a binary answer). Each node thus splits the input variable space, gradually
1043 classifying the input sample into different categories. The final nodes are called
1044 *leaves* and each leaf contains a decision value. Decision trees have the advantage of
1045 being easy to understand and interpret, but being too simple of a model, they are
1046 unable to deal with complex datasets.

1047 3.2.2 Tree Ensembles and Gradient Boosting

1048 A single decision tree is a weak learner: it is unable to correctly represent a complex
1049 dataset. A solution is to use Tree Ensembles: the ensemble consists of a set of Clas-
1050 sification and Regression trees (CART). CARTs differ from decision trees because
1051 each leaf has a score (a real number), rather than a decision value.

1052 The ensemble model sums the outputs of multiple, different decision trees. Trees
1053 can thus complement each other: the strenghts of one tree balance the weaknesses
1054 of others.

1055 The Regression Tree Ensemble model is thus defined as:

$$y_i^{score} = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} \quad (3.3)$$

1056 where \mathcal{F} is the space containing all possible regression trees (seen as functions that
1057 map the variable’s features to the predicted score) and K is the number of trees in
1058 the model.

1059 The parameters that the machine learning algorithm has to learn are the func-
1060 tions f_k , each of which includes the structure of the tree and the scores in the leaves.
1061 A Tree Ensemble model can be trained in two different ways:

- 1062 • **Random Forest.** In a Random Forest algorithm each tree is built using a
1063 random subset of the features (a process called *feature bagging*). This is done
1064 to prevent that a single feature which is a strong predictor leads to correlation
1065 in the trees, and thus an increase in specificity of the model (overfitting).
- 1066 • **Gradient Boosting.** In a Gradient Boosted Decision Tree (GBDT) trees are
1067 added iteratively, chosing at each step the tree which optimizes the Objective
1068 function.

1069 The general machine learning approach is valid for any model, including Tree
1070 Ensembles: an objective function has to be defined and optimized:

$$Obj = \sum_{i=1}^N l(y_i^{true}, y_i^{score}) + \sum_{k=1}^K \Omega(f_k) \quad (3.4)$$

1071 The training loss, as usual, determines the predictive power of the model, and can
1072 be defined as the mean square error, the cross entropy or any other appropriate
1073 function.

1074 The regularization term, describes the complexity of each tree in the model; the
 1075 function Ω can be defined in many ways, such as the number of leaves in the tree or
 1076 the tree depth.²

1077 Gradient Boosting

1078 Since it is intractable to learn all trees at once, an additive strategy [41] is used:
 1079 starting from a constant value a tree is added at each step³:

$$\begin{aligned} y_i^{pred(0)} &= 0 \\ y_i^{pred(1)} &= f_1(x_i) = y_i^{pred(0)} + f_1(x_i) \\ y_i^{pred(2)} &= f_1(x_i) + f_2(x_i) = y_i^{pred(1)} + f_2(x_i) \\ &\vdots \\ y_i^{pred(t)} &= f_1(x_i) + f_2(x_i) + \dots + f_t(x_i) = y_i^{pred(t-1)} + f_t(x_i) \end{aligned} \quad (3.5)$$

1080

1081 The Objective function at round (t) is thus defined as:

$$\begin{aligned} Obj^{(t)} &= L^{(t)} + \Omega^{(t)} = \sum_{i=1}^n l(y_i^{true}, y_i^{pred(t)}) + \sum_{j=1}^t \Omega(f_j) = \\ &= \sum_{i=1}^n l(y_i^{true}, y_i^{pred(t-1)} + f_t(x_i)) + \Omega(f_t) + const \end{aligned} \quad (3.6)$$

1082

1083 The function f_t added at the (t) round, is the one that minimizes the Objective
 1084 function. Taylor expansion of the Objective function defined in Equation 3.6 yields:

$$Obj^{(t)} \simeq \sum_{i=1}^n \left[l(y_i^{true}, y_i^{pred(t-1)}) + g_i \cdot f_t(x_i) + \frac{1}{2} h_i \cdot f_t^2(x_i) \right] + \Omega(f_t) + const \quad (3.7)$$

1085 where:

$$g_i = \partial_{y_i^{pred(t-1)}} \left[l(y_i^{true}, y_i^{pred(t-1)}) \right]$$

$$h_i = \partial_{y_i^{pred(t-1)}}^2 \left[l(y_i^{true}, y_i^{pred(t-1)}) \right]$$

1088 The regularization term $\Omega(f_t)$ can be defined in many ways, and represents the
 1089 complexity of the tree f_t . A tree can be defined in a more mathematical way as:

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

1090 where T is the number of leaves, d is the number of features, w is a vector containing
 1091 the scores on the leaves and q is a function which assigns each data point to the
 1092 corrisponding leaf, thus representing the tree structure.

²The maximum number of splits before reaching a leaf

³This is how the Objective function is computed, but each tree is multiplied by a constant ε
 (called *shrinkage rate* or *learning rate*) just before being added to the model. This means that each
 round is only partially optimized: the reason is that proceeding in small steps prevents overfitting

1093 The model complexity could then be defined as:

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

1094 accounting for the number of leaves as well as the norm of the scores, with the
1095 hyperparameter λ controlling the relevance of the regularization term.

1096 After reformulating the tree model, it is possible to write the Objective function
1097 in Eq. (3.7) for the t^{th} tree (ignoring the constant parts, which is irrelevant for the
1098 optimization), by summing over the leaves rather than the elements in the training
1099 dataset:

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

1100 where $I_j = \{ i \mid q(x_i) = j \}$ is the set of indices of data points which are assigned to
1101 the j^{th} leaf.

1102 By defining $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$ the equation can be further com-
1103 pressed to:

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

1104 Since the Objective form in Eq. (3.11) is quadratic in w_j , the best weight vector for
1105 a given tree structure $q(x)$ has to satisfy: $w_j^{best} = -\frac{G_j}{H_j + \lambda}$.

1106 The Objective function for the t^{th} tree can then be written for the best possible
1107 choice of weights, thus providing a measure of how good a tree structure $q(x)$ is:

$$Obj^{(t)} = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (3.12)$$

1108 **Growing Trees:** Since we now have a way to determine how good a tree structure
1109 is, the ideal procedure would be enumerating every possible tree and picking the best
1110 one. Unfortunately there are infinite possible trees, so a technique called *additive*
1111 *tree learning* is employed: the optimization proceeds one level at a time, and the
1112 tree is grown until further splitting of the branches would increase its Objective
1113 function.

1114 When a leaf is split into two leaves the variation in the Objective function is:

$$Gain = \frac{1}{2} \left[\frac{G_l^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \lambda \quad (3.13)$$

1115 where the first term represents the score on the new left leaf, the second the score
1116 on the new right leaf, the third the score on the original leaf and the last the
1117 increase in the regularization term caused by the additional leaf. The Gain from
1118 the split depends on the chosen feature as well as the splitting threshold. To find
1119 the best split each feature is considered at once: the values are sorted and the split
1120 which maximises the Gain along that feature is found. The best splits over different
1121 features are then compared and the best one is chosen.

1122 The additive method is generally effective and a necessary step to grow trees,
1123 but can lead to a degenerate model because it doesn't consider correlation between
1124 features.

1125 The Gain of a split can be negative: this happens if the increase in the complexity
1126 score is bigger than the reduction in training loss. In such a case two different
1127 strategies can be employed: the split can either be immediately stopped if its gain
1128 is negative, or the tree can be grown to maximum depth and then the splits with
1129 negative score can be recursively removed (which is helpful in cases where a split
1130 with poor Gain is necessary for a future high-gain split). This is known as *pruning*.

1131 3.3 LightGBM

1132 The Machine Learning Algorithm used
1133 in this thesis is Light Gradient Boosting
1134 Machine (LightGBM), a gradient boost-
1135 ing framework developed by Microsoft
1136 Research and Peking University [42].

1137 LightGBM is based upon a GBDT,
1138 but implements features which im-
1139 prove training speed and accuracy:
1140 histogram-based algorithms are used to
1141 reduce the time needed to calculate
1142 which is the optimal split when grow-
1143 ing a tree, improving speed and trees
1144 are grown leaf-wise rather than level-
1145 wise [43] (see Figure 3.2).

1146 The LightGBM package offers multiple hyperparameters [44] to determine the
1147 model's features (such as the maximum depth or the maximum number of leaves for
1148 each tree in the model), and to control the training (such as the learning rate, the
1149 number of boosting rounds and regularization).

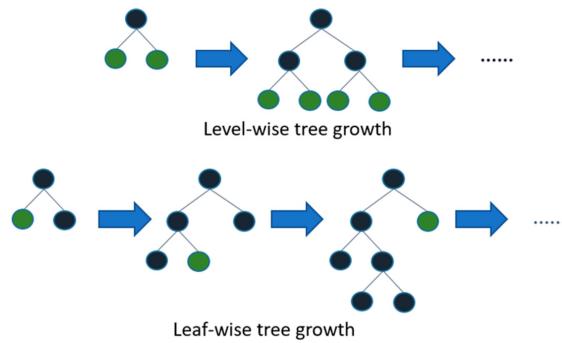


Figure 3.2: Depth-wise and leaf-wise tree growing

1150 CHAPTER 4

1151 **Photons and Electrons**
1152 Classification

1153 In the ATLAS detector, electrons and photons which cannot be easily classified are
1154 reconstructed in both ways, and each physics analysis can resolve the ambiguity in
1155 its preferred way.

1156 A Gradient Boosted Decision Tree is a very useful tool in such a situation: using
1157 Monte Carlo (MC) simulations of single particles in the detector, with $y_{photon}^{true} = 1$
1158 and $y_{electron}^{true} = 0$, a model can be trained to classify ambiguous electron and photons
1159 and improve the reconstruction efficiency, without compromising the background
1160 rejection.

1161 Another advantage of Classification Trees is that the output is a score, rather
1162 than a discrete value. This allows to easily determine a working point, as shown
1163 in Figure 4.1, by varying the threshold on the output score needed to classify the
1164 particle as a photon rather than an electron. The threshold can be optimized for
1165 each specific physics analysis, by requiring that the expected significance on the
1166 signal under test is maximised.

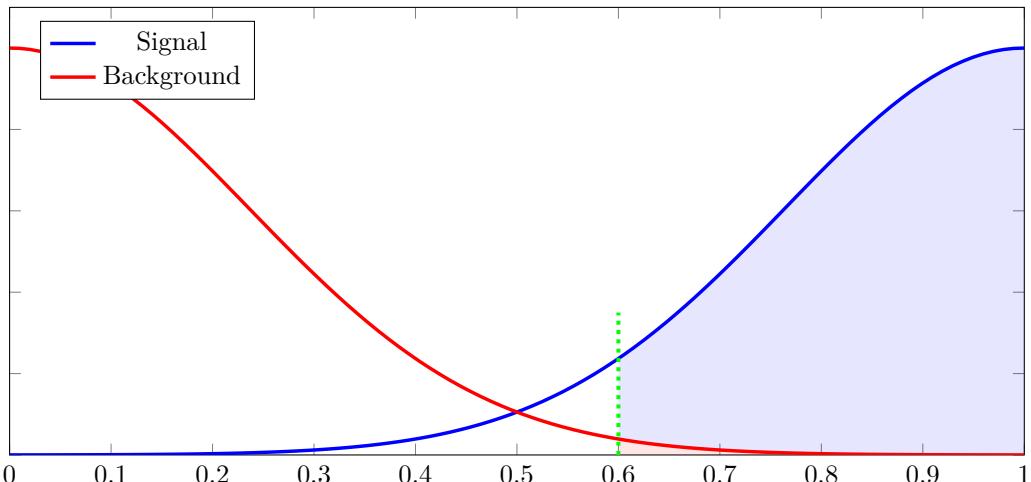


Figure 4.1: Ideal Classification scores distribution (with threshold set at 0.6)

1167 **4.1 BDT Training**

1168 The following sections focus on the model training, which is carried out using a MC
1169 sample of single particles.

1170 **4.1.1 Single Particle sample generation and Dataset Prepa-**
1171 **ration**

1172 The training of the Gradient Boosted Decision Tree is performed on single-particle
1173 samples of electrons and photons, with pile-up. The single particle samples are
1174 simulated using a GEANT4-rated [45] model of the ATLAS detector. For each one
1175 of the two samples the particles reconstructed as photons and electrons are saved
1176 separately in Pandas [46] dataframes. Four dataframes are thus created in total:
1177 one for each possible combination of *True* and *Reco* particle.

1178 The data is then prepared to be used for the BDT training (see appendix B.1.1)
1179 with the following procedures:

- 1180 • Particles generated from pile-up interactions are removed by means of a cut
1181 on the particles p_T , set at 20 GeV.
- 1182 • Particles are deleted from the sample if their reconstructed and true p_T differ
1183 by more than 10%. This is done to ensure that the reconstructed particle is
1184 matched to the correct true particle.
- 1185 • The two dataframes corresponding to each *True* particle are then merged,
1186 aligning each particle marked as ambiguous with its counterpart: for each
1187 event that led to the reconstruction of both an ambiguous photon and an
1188 ambiguous electron, the features of both particles are next to each other in a
1189 single row of the final dataframe.

1190 Two dataframes are thus now present: *TruePhoton* and *TrueElectron*.

- 1191 • A column indicating the true particle (with the convention $y_{ph} = 1$ and $y_{el} = 0$)
1192 is added to the dataframes, before concatenating them in a unique dataset.
- 1193 • Duplicate features (such as E , η or ϕ , which are computed for both electron
1194 and photons) are removed, and only the ones necessary for the training are
1195 left in the final dataframe.
- 1196 • A Scikit Learn [47] function is used (see appendix B.1.2) to separate the
1197 dataframes into training and testing sets (80% and 20% respectively). This is
1198 important because the LightGBM algorithm evaluates the loss function after
1199 each boosting round on a dataset different than the one used for training, and
1200 stops the iteration if the score doesn't improve for a fixed number of consec-
1201 utive rounds, preventing the model from being overfitted on the training set.
1202 After preparation the full dataset consists of $\sim 1.2 \cdot 10^6$ Photons and $\sim 6.6 \cdot 10^5$
1203 Electrons.
- 1204 • The dataframes are finally converted into LightGBM *Dataset* objects.

1205 **Feature Selection**

1206 The features to be used for the training are chosen by comparing the features dis-
1207 tributions for true photons and true electrons, and choosing the ones that are the
1208 most discriminating (see Figure 4.2).

1209 Some of the general discriminating characteristics are the following:

- 1210 • The distributions of features describing hits in the Inner Detector peak at a
 1211 smaller number of hits for true photons than for true electrons.
- 1212 • The features relative to photons conversions, when plotted for true electrons,
 1213 show “mistakes” in the reconstruction process. For instance the true electron
 1214 distribution of the photon conversion radius strongly peaks at ~ 31 mm, which
 1215 is the distance of the insertable B-layer from the beam line (see section 1.2.1),
 1216 indicating that an electron track, originating at the first Pixel layer, was mis-
 1217 takenly reconstructed as a photon single track conversion. True electrons are
 1218 also mainly classified as converted photons, while the true photon distribution
 1219 is shifted towards the unconverted side.

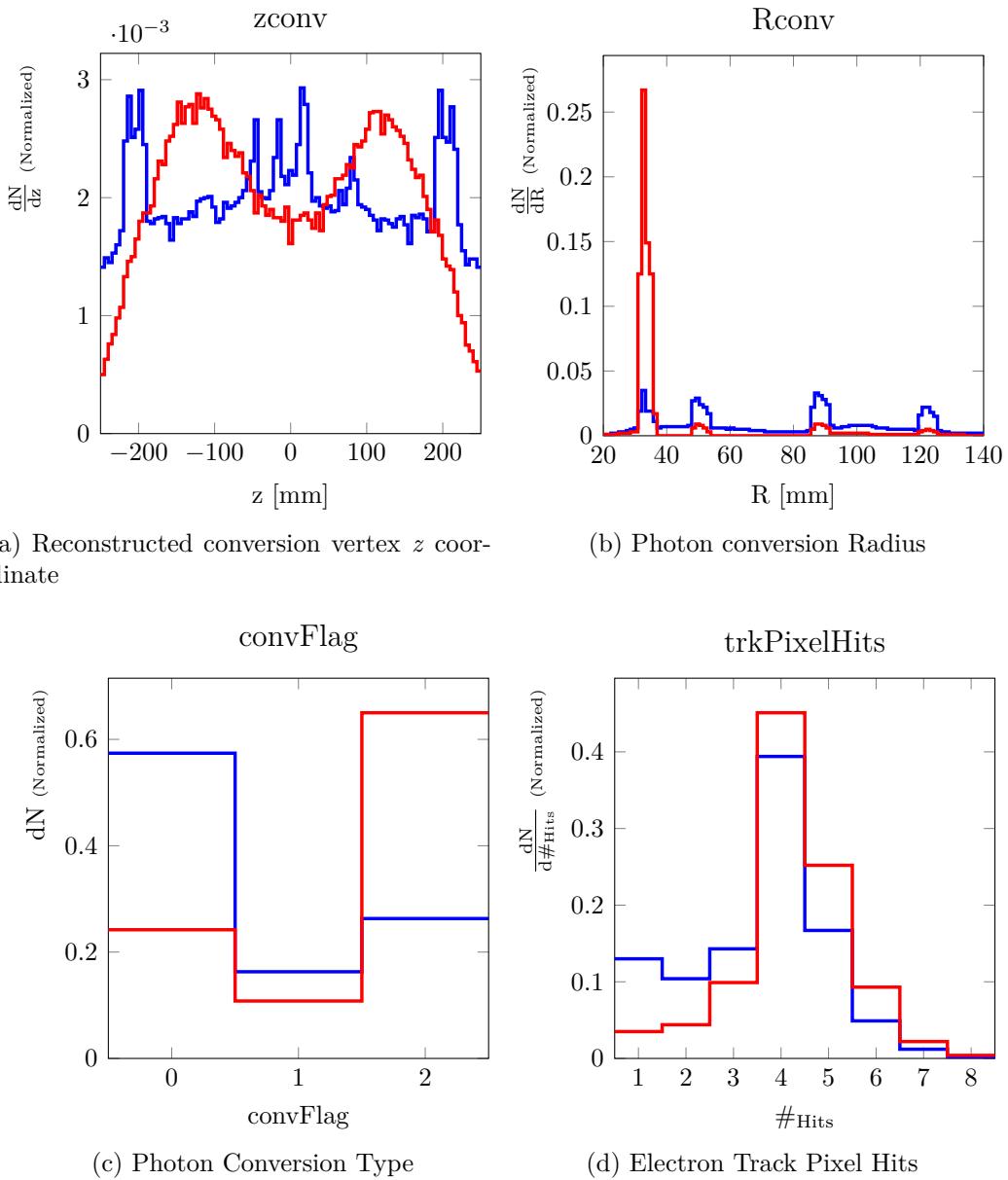


Figure 4.2: Normalized distributions of features showing high discriminating power between true photons (blue) and electrons (red)

1220 The final features used in the BDT training are:

- 1221 • **General Features** of the particle: the reconstructed η , ϕ as well as the
1222 particle energy¹ and p_T .
- 1223 • **Electron Features** representing the track quality (number of hits in the Pixel
1224 Detector and in the Semi Conductor Tracker) as well as physics quantities
1225 reconstructed from the track (the track transverse momentum and pseudo-
1226 rapidity, the longitudinal impact parameter z_0 , the ratio between the energy
1227 deposited in the calorimeter cluster and the track momentum).
- 1228 • **Photon Features** representing characteristics of the conversion (a *flag* identi-
1229 fying whether the photon has undergone a conversion, together with the radius
1230 and z coordinate of the reconstructed conversion vertex) as well as attributes
1231 of the tracks left in the ID by the e^-e^+ pair (Pixel/SCT Hits and p_T)

1232 4.1.2 Hyperparameter optimization and Model Training

1233 As mentioned in section 3.3, the LightGBM package offers multiple Hyperparam-
1234 eters which control the algorithm behaviour. Although there is some logic behind
1235 possible choices (e.g. reducing the number of leaves in a tree decreases the chances
1236 of overfitting, whilst a large learning rate increases them), which parameters are
1237 best strongly depends on the tackled problem, and finding them is an optimization
1238 task in and of itself.

1239 The optimization process requires training the BDT with different hyperparam-
1240 eters, and choosing the ones which perform best.²

1241 The most simple and most common strategy is a *grid search*: the best hyper-
1242 parameters are searched by sampling points evenly distributed in a prefixed search
1243 range. This method can theoretically find the best parameters with arbitrary pre-
1244 cision, by sampling adequately many points from the range. In practice though, a
1245 large amount of sampling points is needed to achieve a sufficient precision, and the
1246 number of iterations grows exponentially with the number of hyperparameters to
1247 be optimized. Since each BDT training takes a few minutes, the optimization time
1248 would be unreasonably high and other techniques are often preferable.

1249 Another method is the *random search*, in which points are selected at random
1250 within the search space. It can be more efficient than the grid search, especially
1251 when many hyperparameters are being optimized at once, because it can effectively
1252 scan the whole parameter space using fewer iterations than a grid search.

1253 The Python package Hyperopt [48] provides useful functions to search for optimal
1254 parameters in an even more efficient way. As the hyperparameter space is searched,
1255 the algorithm uses previous search results to identify the most promising area of
1256 the parameters space and samples points more densely near the maximum (see
1257 Figure 4.3).

¹The particle energy is slightly different when computed for an electron object or a photon object due to different calibrations. The difference is negligible, though, and the photon variable is used.

²The quantity determining which parameters are the best can be any indicator of the predictive power of the trained model, such as the test dataset loss, or the area under the ROC curve. In this work the test dataset cross entropy is used.

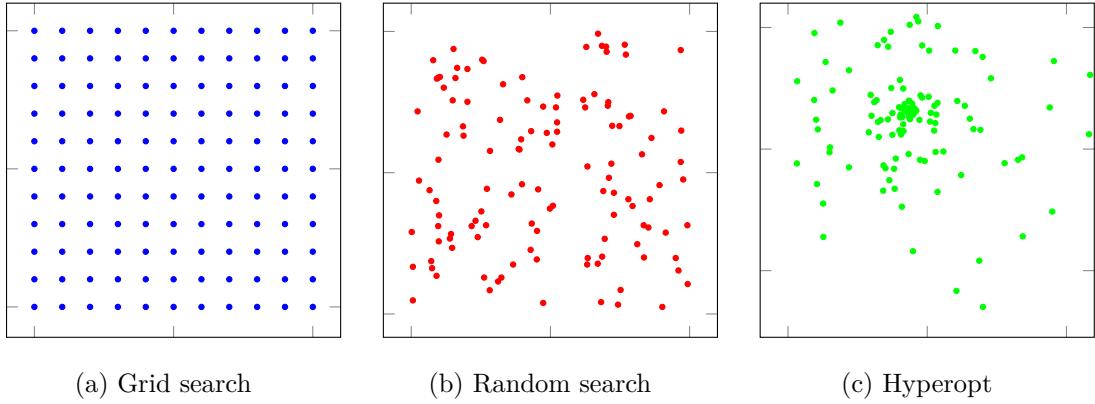


Figure 4.3: Different hyperparameter searching algorithms

1258 LightGBM offers multiple parameters, of which the following are optimised (see
 1259 appendix B.1.3):

- 1260 • **Number of leaves.** The maximum number of leaves in a tree: having too
 1261 many can lead to an overfitted model, whilst having not enough causes under-
 1262 fitting.
- 1263 • **Feature Fraction.** To prevent overfitting a subset of the available features is
 1264 randomly selected before training each tree in the model. This hyperparameter
 1265 controls which percentage of the features is selected with every iteration.
- 1266 • **Bagging fraction and bagging frequency.** Only a fraction of the available
 1267 data is selected to train the trees. The resampling happens at a frequency
 1268 determined by a hyperparameter.
- 1269 • **Learning rate.** The shrinkage rate, cited in section 3.2.2. It can be used to
 1270 prevent overfitting.

1271 Some hyperparameters were also fixed without optimization, most importantly
 1272 the Objective function used by the algorithm (loss and regularization) and the *early*
 1273 *stopping*.³

The main optimized hyperparameters are summarized in the following table:

Hyperparameter	Value
Loss function	Cross Entropy
Regularization	L2, $\lambda = 0.1$
Number of leaves	30
Feature fraction	0.66
Bagging	0.8
Bagging frequency	14
Learning rate	0.07
Early stopping	5

1274

³The Early Stopping parameter automatically stops the algorithm if the loss on the test set hasn't improved for a fixed number of boosting rounds, thus preventing overfitting.

1275 The Hyperopt package is provided with a *Trials* object, which records the loss
 1276 obtained with every tested parameters choice. When variation of the hyperparameters
 1277 shows no notable effect on the loss, the simpler model is preferred (e.g. the
 1278 optimal number of leaves is searched in a range from 0 to 100; although the mini-
 1279 mum loss is recorded at $N_{leaves} = 86$, adding more than 30 leaves doesn't improve
 1280 the cross entropy significantly, and the simpler model is likely to be more consistent
 1281 when applied to other datasets).

1282 Model Analysis

1283 Parameters of the trained model are plotted (see appendix B.1.5) and analized to
 1284 understand the results of the training:

- 1285 • The feature importance plots show which features are the most discriminating.
 1286 For each feature both the number of times it was used in the splitting of a
 1287 branch, and the overall Gain provided to the model are plotted (see Figure
 1288 4.4).

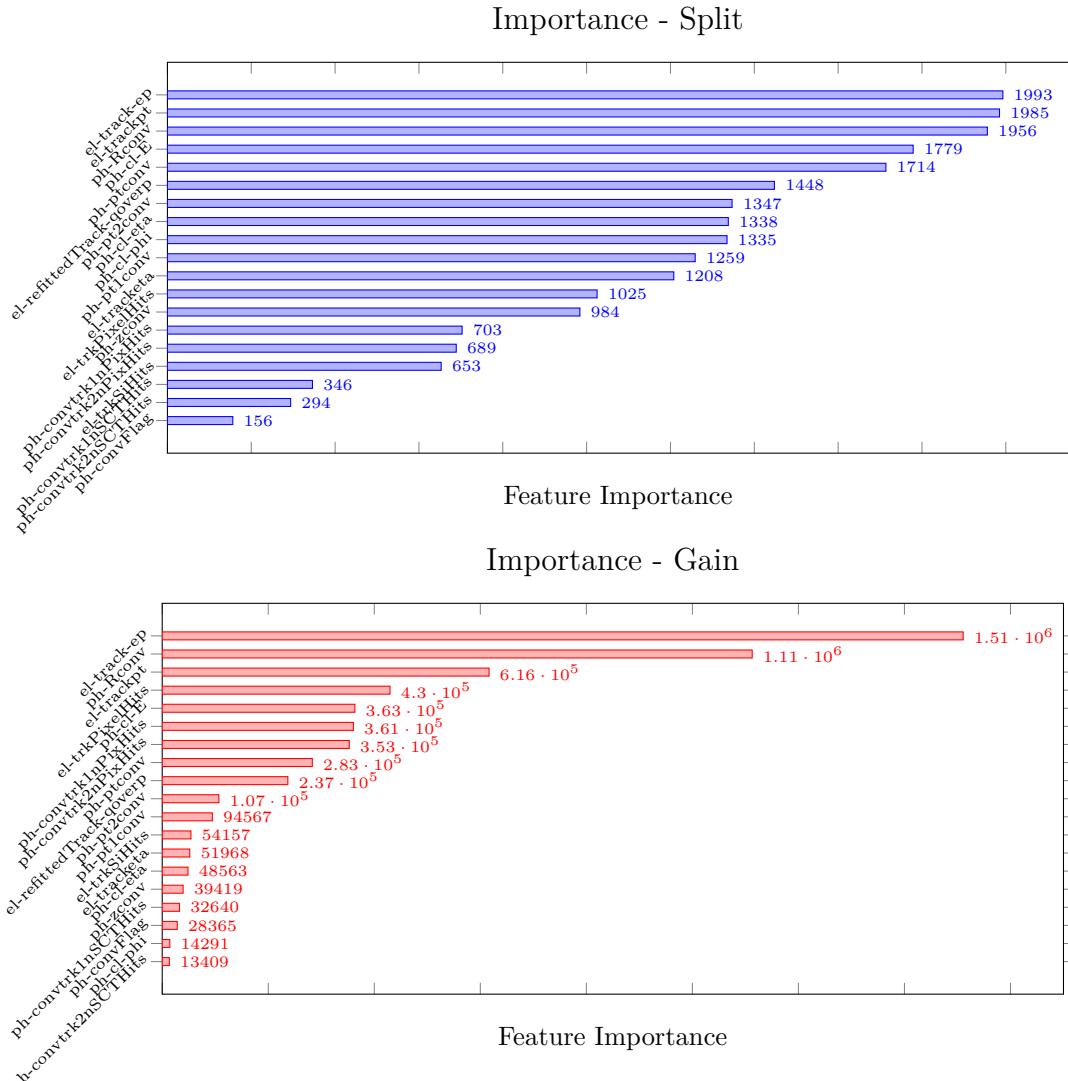


Figure 4.4: The importance plots of the training features

- 1289 • The value of the Objective function after each iteration is plotted (see Figure
 1290 B.2 in section B.1.5) for both the training and test set. The analysis of this
 1291 plot can help identify overfitting, by showing a large difference between the
 1292 train and test datasets.

1293 **4.1.3 BDT Prediction and Analysis**

1294 The trained BDT is used to predict the scores of the particles in the test dataset,
 1295 and an analysis of the classification capabilities of the model is performed. In par-
 1296 ticular the scores distribution is plotted and analysed, the efficiency on signal and
 1297 background is computed and plotted as a function of the cut on the score, and the
 1298 ROC curve and AUC are evaluated.

1299 **Scores distribution**

1300 A histogram (Figure 4.5) is drawn of the scores distribution (see appendix B.1.6):
 1301 it shows great separation between photons and electrons.

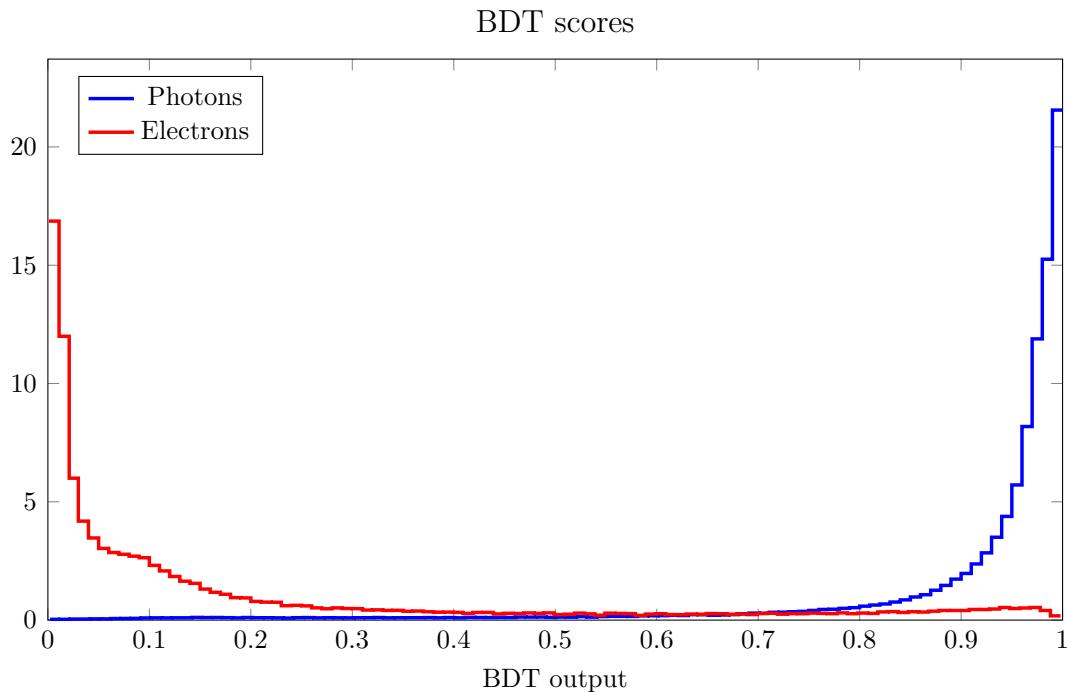


Figure 4.5: Scores distribution for True electrons and True photons reconstructed as ambiguous

1302 **Efficiency evaluation and ROC curve**

1303 The photon and electron efficiencies are defined, respectively, as the True Positive
 1304 Rate (TPR)⁴ and as the False Positive Rate (FPR).⁵ Both efficiencies clearly depend
 1305 on the chosen threshold, since they represent the integral from the threshold to 1

⁴ $\varepsilon_{\text{ph}} = \text{TPR} = \frac{\text{Photons reconstructed as Photons}}{\text{All Photons}}$

⁵ $\varepsilon_{\text{el}} = \text{FPR} = \frac{\text{Electrons reconstructed as Photons}}{\text{All Electrons}}$

1306 of the distribution shown in Figure 4.5. From the efficiency plot (see Figure 4.6⁶)
1307 it is possible to see that, by selecting an appropriate threshold, a high background
1308 rejection⁷ can be achieved without compromising the signal efficiency.

1309 An effective parameter used to evaluate the predictive capabilities of a trained
1310 model is the ROC curve: for each possible threshold a point is added to a graph with
1311 coordinates $(x, y) = (\text{FPR}, \text{TPR})$. The resulting curve (Figure 4.7) represents the
1312 predictive capabilities of the model: the line connecting $(0,0)$ to $(1,1)$ represents the
1313 predictive power of a model which assigns data to a random category (no discrimi-
1314 nation capacity), if the curve is above this line the discrimination capacity increases
1315 up to the point when the model is able to perfectly distinguish the two classes.

1316 To encapsulate in a single value the information contained in the ROC curve,
1317 the area under the curve (AUC) is often used; the closer the AUC is to 1, the more
1318 predictive the model is. The LightGBM model trained on the single particle dataset
1319 achieved a ROC AUC of 0.964 on the test set.

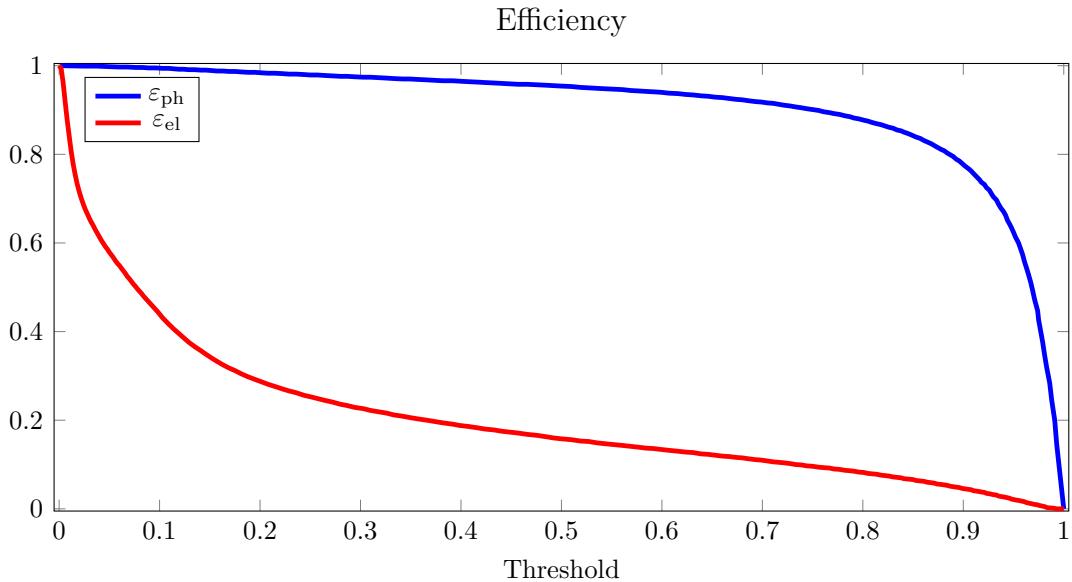


Figure 4.6: Efficiency for ambiguous True electrons and True photons

1320 4.2 Physics analysis application

1321 The trained model is then tested on other MC datasets. A low-mass $H \rightarrow \gamma\gamma$ signal
1322 is used together with background from the $Z \rightarrow ee$ decay.

1323 4.2.1 Sample generation and Dataset Preparation

1324 The first sample contains photons objects reconstructed from true diphoton decays
1325 of a hypothetical low mass Higgs-like particle. The invariant mass of the signal in
1326 the generated sample is 80 GeV, but it is changed to 90 GeV to see the effect of

⁶A detailed description of the data used in the efficiency plot and the ROC curve can be found in appendix A.1.1

⁷The background rejection is defined as $(1 - \varepsilon_{\text{el}})$, and thus represented by the True Negative Rate

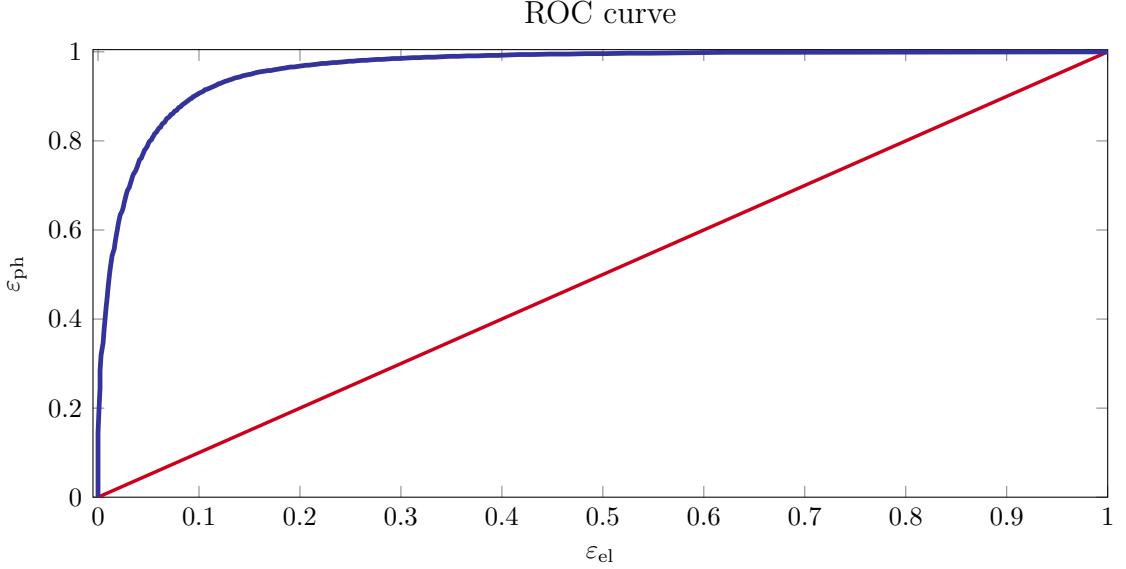


Figure 4.7: ROC curve of the trained model, applied to the ambiguous particles in the test set

1327 the BDT in the worst case scenario: directly above the $Z \rightarrow ee$ resonant peak. This
 1328 is done by shifting the mass distribution by 10 GeV to the right, and correcting
 1329 the number of expected events to account for the changes in the production cross
 1330 section and in the $\gamma\gamma$ branching ratio.

1331 The background sample on the other hand contains reconstructed photon ob-
 1332 jects, which come from true electrons from a Z decay, faking photons (representing
 1333 the Drell-Yan background). In both samples reconstructed photons classified as
 1334 ambiguous have been *decorated* with the variables of the corresponding ambiguous
 1335 electron. Since the BDT needs the data from both electron and photon reconstruc-
 1336 tion algorithms, only ambiguous objects, for which both options are saved, can be
 1337 classified. Unambiguous objects classified uncorrectly can't be fixed, but the per-
 1338 centage of events in the $Z \rightarrow ee$ dataset in which both electrons have been mistakenly
 1339 classified as unambiguous photons is small (< 5 % of all $Z \rightarrow ee$ events reconstructed
 1340 as $\gamma\gamma$).

1341 The datasets are prepared (see appendix B.2.1) by applying a quality cut [49],
 1342 which consists of the following requirements:

- 1343 • A cut on the p_T of the reconstructed particle is set at 22 GeV.
- 1344 • The photons are required to satisfy the *Tight* Identification criteria (see section
 1345 2.2.1).
- 1346 • The particles are required to fulfill the *FixedCutLoose*⁸ isolation requirement
 1347 (see section 2.2.2).

1348 It is important to notice that, unlike the single particle sample, the $H \rightarrow \gamma\gamma$ and
 1349 $Z \rightarrow ee$ samples contain *events*, each of which is made up of two photon candidates.
 1350 It is necessary to keep this information while applying the GBDT, since even if only

⁸The requirements of the FixedCutLoose isolation are $E_T^{cone20} < 0.065 \cdot p_T$ and $p_T^{cone20} < 0.05 \cdot p_T$

1351 one of the two particles in an event is classified as an electron rather than a photon,
1352 the whole event can be removed from the analysis.

1353 Events in the physics dataset also have a *weight*, which depends by the MC
1354 simulation. The weights account for the selection efficiency as well as the production
1355 cross section and decay branching ratio.

1356 **Training and Physics datasets feature comparison**

1357 The distribution of the features in the single-particle datasets used in training are
1358 compared to the ones of the physics datasets, to ensure that it is reasonable to
1359 expect that the single-particle trained model can be predictive even with photons
1360 and electrons originating from real physics processes. An example of the process for
1361 two features is shown in Figure 4.8.

1362 **4.2.2 BDT Prediction and Analysis**

1363 The model previously trained on single particles is used (see appendix B.2.2) to get
1364 a score for each event. The event score is calculated by firstly assigning a score of 1
1365 to all unambiguous particles, and then selecting, for each event, the smallest score
1366 among the two particles in it. This reduces the number of calculations needed for
1367 the ROC analysis, and, since an event is considered a $H \rightarrow \gamma\gamma$ event if both particles
1368 in the event have a score bigger than the fixed threshold, does not result in loss of
1369 information.

1370 The scores distribution is drawn in Figure 4.9, and shows that the algorithm is
1371 able to effectively classify photons and electrons even in the physics sample. Unam-
1372 biguous particles (from both the $H \rightarrow \gamma\gamma$ and $Z \rightarrow ee$ datasets) are not included in
1373 the plot.

1374 **ROC analysis**

1375 The efficiency curves are shown in Figure 4.10⁹: although less effectively than in
1376 the single particle dataset (partly because of the unambiguous events, and partly
1377 because of a poorer performance of the BDT model on the physics sample), the
1378 BDT is able to increase the background rejection while mantaining a high signal
1379 efficiency.

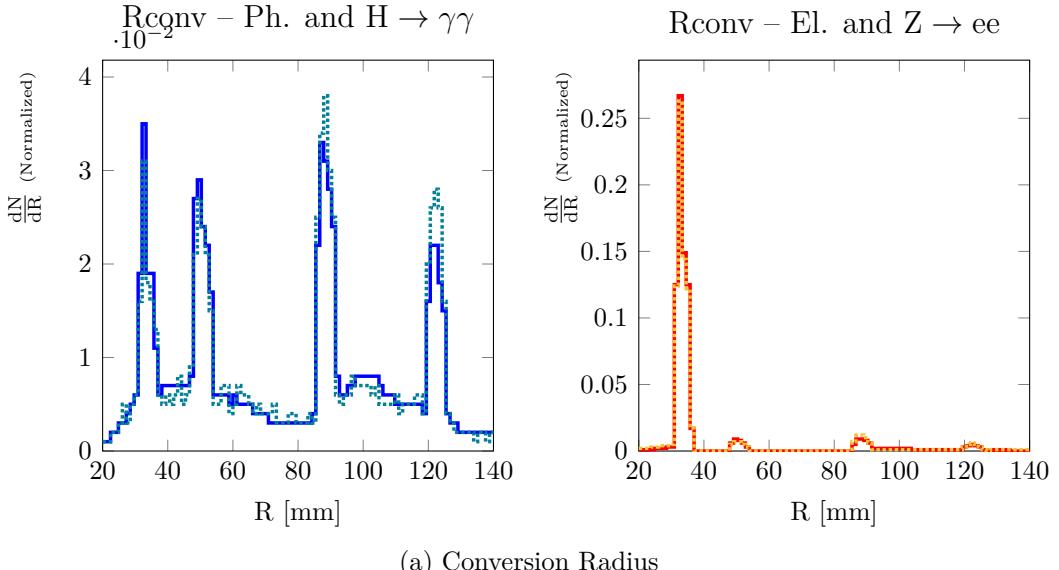
1380 The ROC curve is also plotted (see Figure 4.11). Since a number of events are
1381 not ambiguous¹⁰, the ROC curve does not start from the point (0, 0): no matter
1382 which threshold is selected, the unambiguous events will always pass the cut. The
1383 starting point of the curve, indeed, depends on the ratio of unambiguous events in
1384 the sample of particles which passed the initial quality cut:

$$\left(N_{Z \rightarrow ee}^{\text{Pass_notAmb}} / N_{Z \rightarrow ee}^{\text{Pass_tot}}, N_{H \rightarrow \gamma\gamma}^{\text{Pass_notAmb}} / N_{H \rightarrow \gamma\gamma}^{\text{Pass_tot}} \right)^{11} = (0.062, 0.602).$$

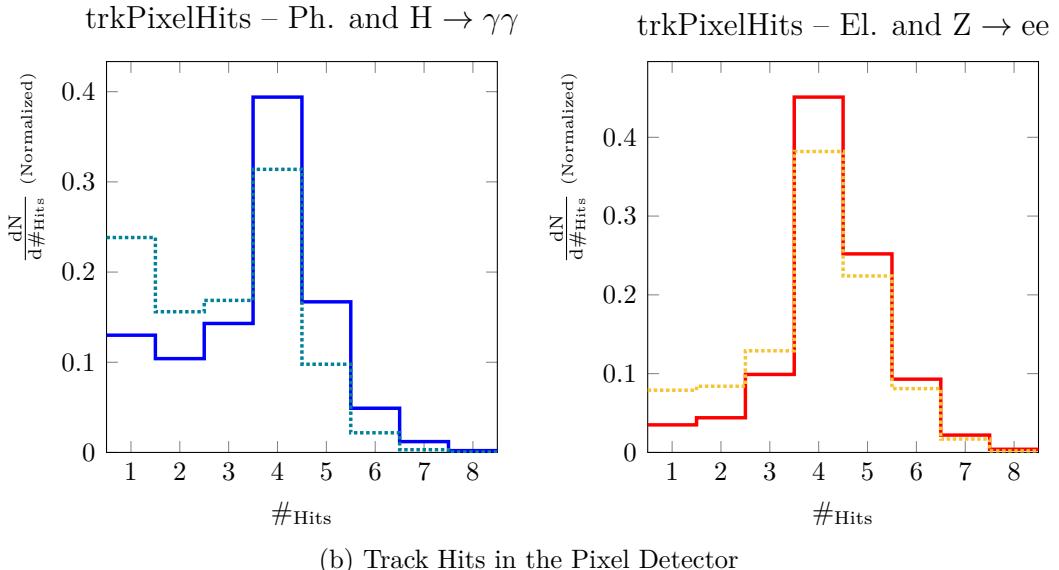
⁹A detailed description of the data used in the efficiency plot, the ROC curve and the significance plot can be found in appendix A.2.1

¹⁰If both particles in the event are not classified as ambiguous, the event is automatically considered a $H \rightarrow \gamma\gamma$ event.

¹¹Where $N_{X \rightarrow xx}^{\text{Pass_tot}}$ and $N_{X \rightarrow xx}^{\text{Pass_notAmb}}$ refer to the *weighted* number of events (i.e. the sum of the weights of the events) passing the initial quality cut and safisfying the condition



(a) Conversion Radius



(b) Track Hits in the Pixel Detector

Figure 4.8: Normalized distributions of features showing the comparison between single particle photons (—), single particle electrons (—), particles reconstructed from the H → γγ signal (···) and particles reconstructed from the Z → ee background (····)

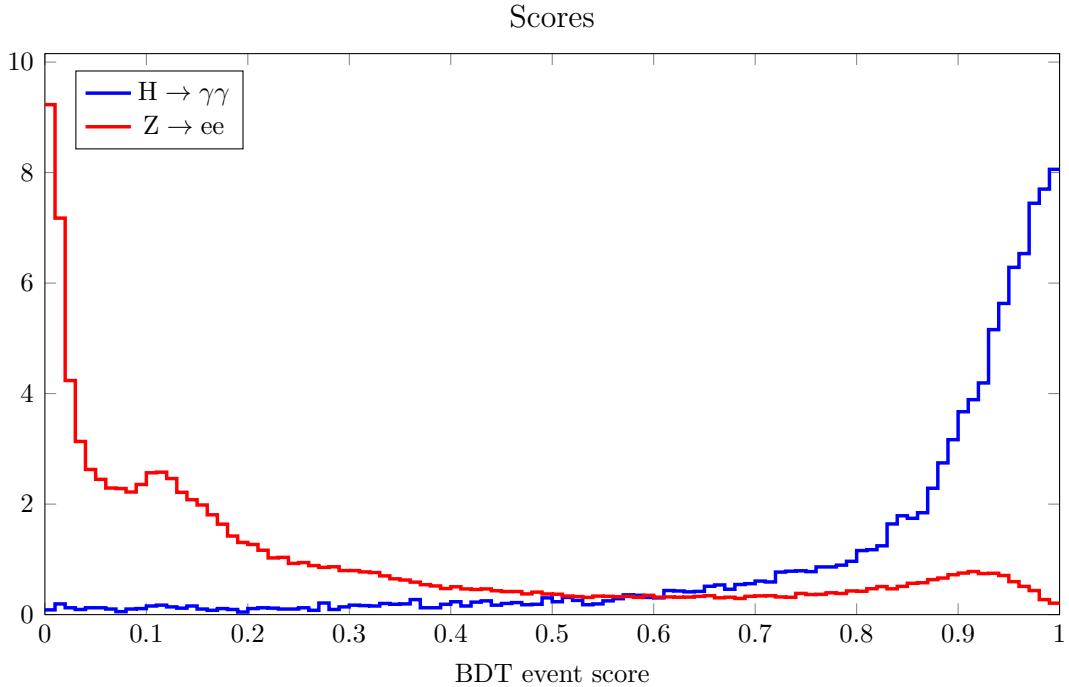


Figure 4.9: Event Scores distribution of the physics samples (ambiguous only)

1385 The red line, similarly to the single particle ROC curve plot, represents the
 1386 predictive power of a model which randomly categorizes particles.

1387 Significance Maximisation

1388 The significance of the 90 GeV $H \rightarrow \gamma\gamma$ signal is computed (using equation 1.4) by
 1389 taking into consideration the expected number of events passing the selection from
 1390 the $H \rightarrow \gamma\gamma$ signal, the $Z \rightarrow ee$ background and the continuous QCD background
 1391 (mainly composed of $\gamma\gamma$ as well as γj and jj , whith the jets faking photons).

$$S = \frac{\#_{H \rightarrow \gamma\gamma} \text{ events} \cdot \varepsilon_{H \rightarrow \gamma\gamma}}{\sqrt{(\#_{Z \rightarrow ee} \text{ events} \cdot \varepsilon_{Z \rightarrow ee}) + (\#_{\text{continuosBKG}} \text{ events} \cdot \varepsilon_{H \rightarrow \gamma\gamma})}} \quad (4.1)$$

1392 The significance is evaluated for various thresholds, for events in a mass range
 1393 between 88 GeV and 92 GeV. The threshold for which the significance is maximised
 1394 is used in the following calculations.

1395 At the moment the Low Mass $H \rightarrow \gamma\gamma$ analysis considers all photon candidates
 1396 (ambiguous or not) as photons, thus effectively using a threshold equal to 0. It is
 1397 clear, from Figure 4.12, that only a small improvement in significance, with respect
 1398 to the current situation, is achievable. This is due to the fact that the main source of
 1399 background is the continuous $\gamma\gamma$ background, which isn't reduced by the algorithm
 1400 developed in this thesis. In any case, a ~ 0.17 threshold grants a $\sim 55\%$ rejection
 1401 over the DY background, whilst mantaining the signal efficiency at around 100%.

1402 4.2.3 $M_{\gamma\gamma}$ distributions

1403 The mass distribution of the samples is drawn over the continuous $\gamma\gamma$ background,
 1404 both before and after the application of the BDT algorithm. Three categories are

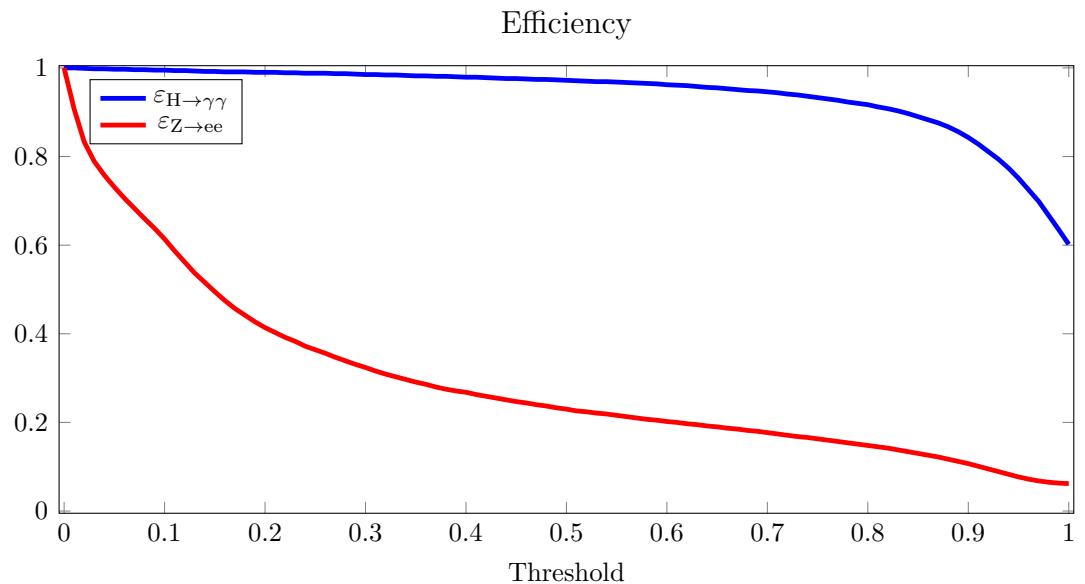


Figure 4.10: Signal and Background efficiency on the physics sample

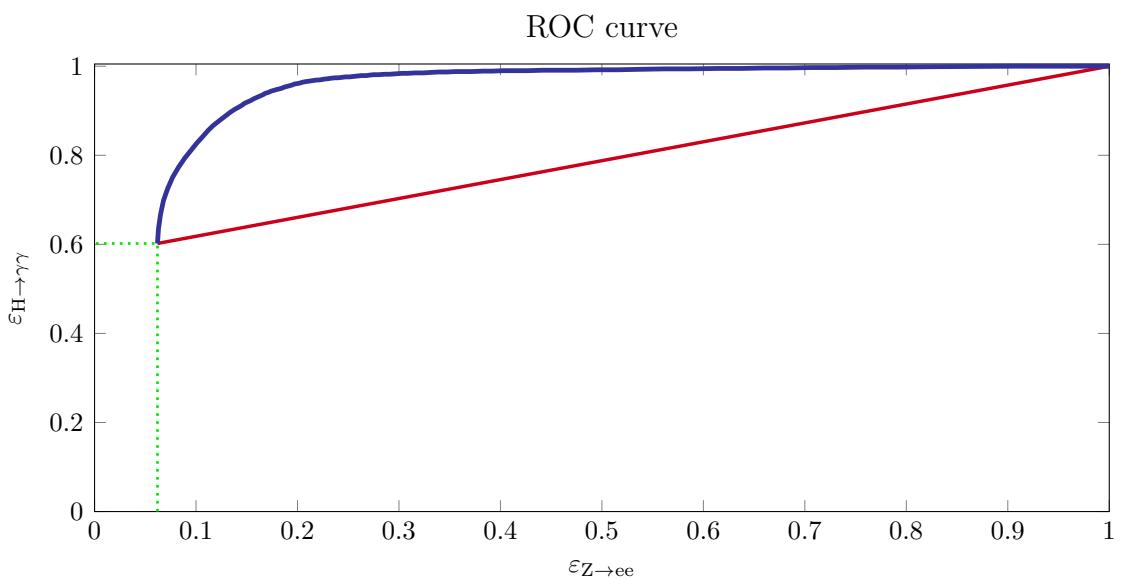


Figure 4.11: ROC curve of the trained model, applied to the physics sample

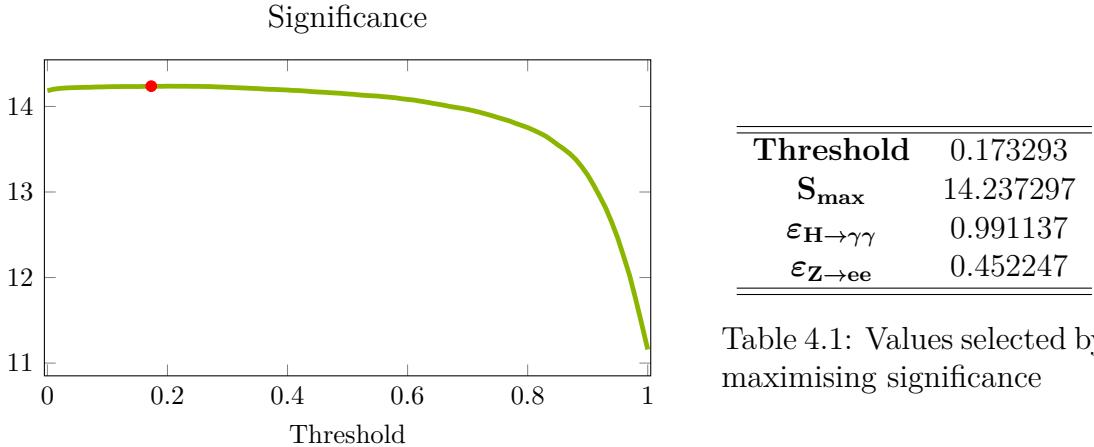


Figure 4.12: Signal significance for different thresholds

1405 used, to identify whether the two photons in the event are both unconverted (UU)
 1406 both converted (CC) or one converted and one unconverted (UC). Since most of the
 1407 fake photons from the $Z \rightarrow ee$ decay are classified as converted, the Drell-Yan peak
 1408 is mostly present in the CC category.

1409 The consistent rejection of electron faking photons achieved by the BDT shows
 1410 up in the $(\gamma\gamma_{\text{BKG}} + \text{DY}_{\text{BKG}})$ plot (Figure 4.13¹²) as a sizeable reduction of the DY
 1411 peak (see section 1.2.3). On the other hand, if a $H \rightarrow \gamma\gamma$ signal were to be present
 1412 (as in Figure 4.14), the high signal efficiency ensures that it is not lost when the
 1413 BDT is applied.

1414 The expected number of events (for $\mathcal{L} = 140 \text{ fb}^{-1}$), in a window of $90 \pm 2 \text{ GeV}$
 before and after the application of the BDT is summarized in Table 4.2.

Events	Without BDT	With BDT
Background (continuous)	346733	343660
Background (Drell-Yan)	14138	6394
Signal ($H \rightarrow \gamma\gamma$)	9766	9679

Table 4.2: Expected number of events with $m_{\gamma\gamma} \in [88 \text{ GeV}, 92 \text{ GeV}]$ before and
 after BDT cut

1415

1416 4.3 Results

1417 The Gradient Boosted Decision Tree algorithm developed in this thesis allows a
 1418 powerful and flexible resolution of the ambiguity between photons and electrons.

1419 The model performs very well in the case of single particles: by tuning the thresh-
 1420 old it is possible to achieve an electron rejection of up to 90 %, whilst mantaining
 1421 the photon efficiency over 90 %.

1422 The BDT performance is also fairly good on the physics dataset: the algorithm
 1423 is able to effectively distinguish between ambiguous electrons and photons. Requir-

¹²A detailed description of the data used in the $m_{\gamma\gamma}$ plots can be found in appendix A.2.2

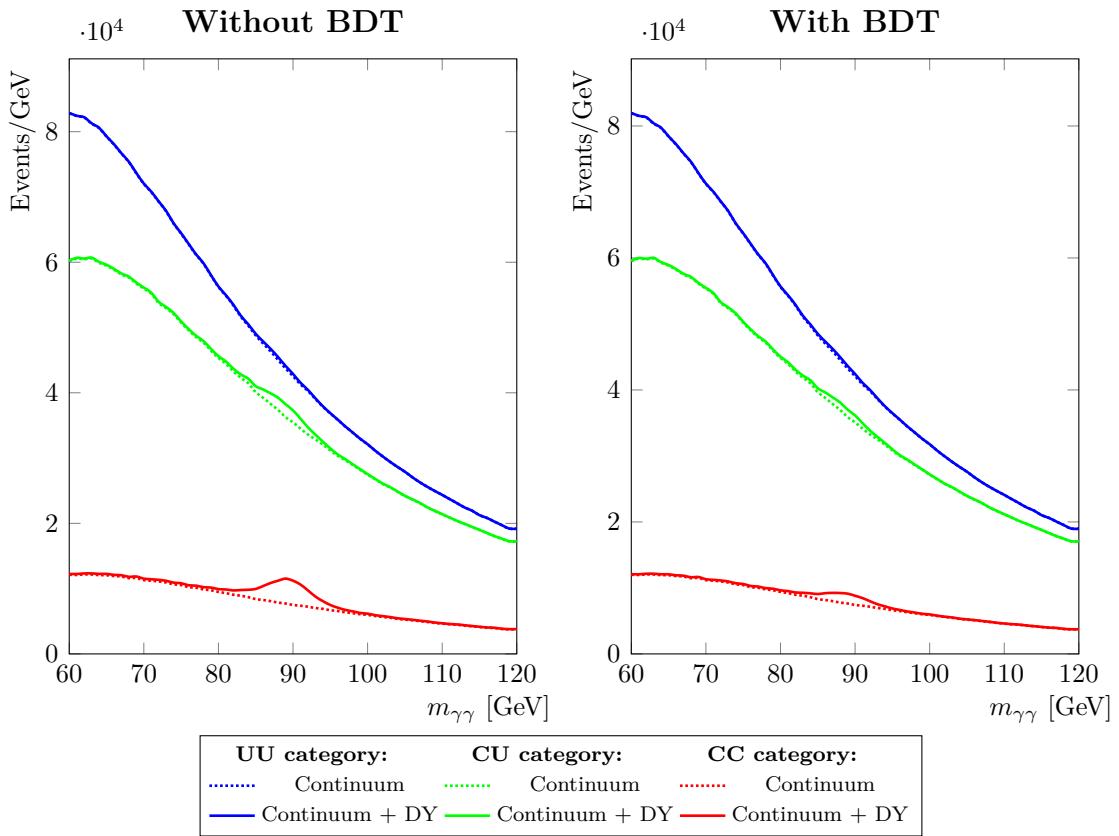


Figure 4.13: Background invariant mass distribution

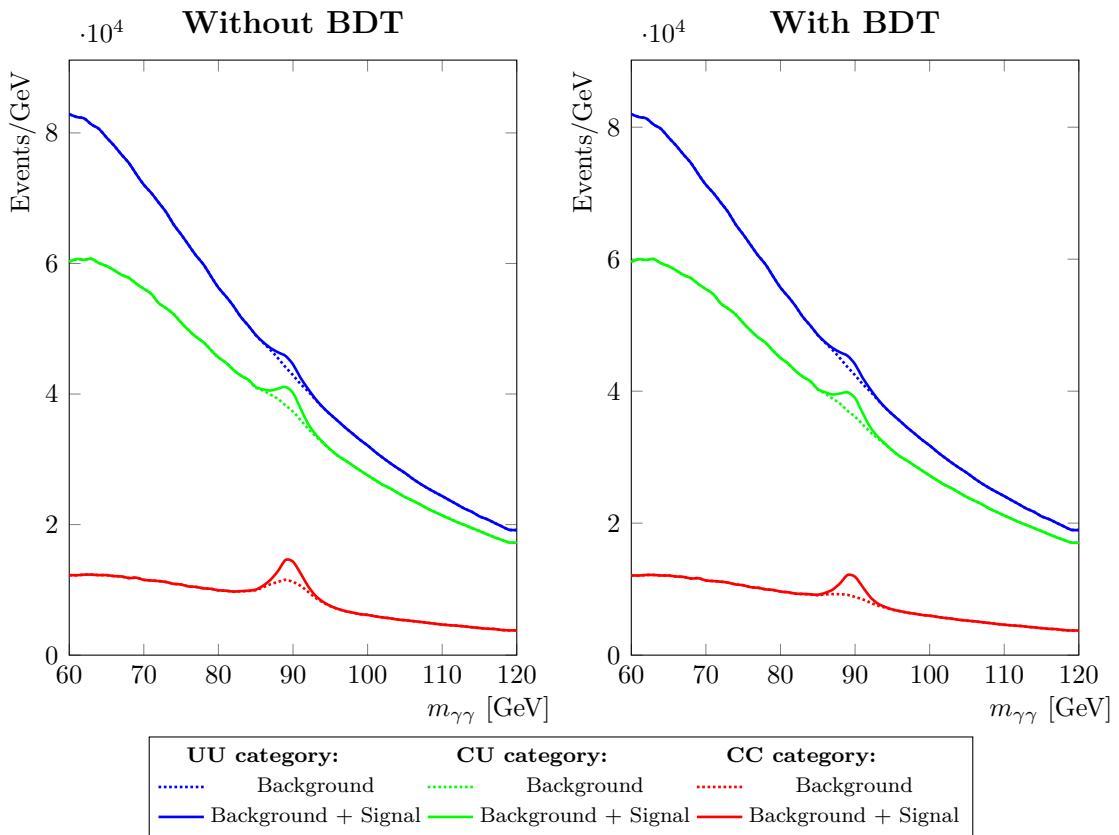


Figure 4.14: Background + Signal invariant mass distribution

1424 ing the signal efficiency to be higher than 90 %, it is possible to achieve an 80 %
1425 background rejection.

1426 The significance for a SM-like Higgs boson at $m_H = 90$ GeV after the application
1427 of the BDT is not particularly improved, mainly because of two different factors:

- 1428 • The unambiguous events cannot be modified by the BDT. The solution would
1429 be to implement looser requirements for particles to be classified as ambiguous
1430 by the ambiguity resolution algorithm (see section 2.2). This would allow a
1431 larger overlap between the reconstructed photons and electrons, thus making
1432 the application of machine learning algorithms even more effective.
- 1433 • The major source of background in the search for resonances in the low mass
1434 region is the continuous di-photon irreducible background. On the other hand
1435 there could be other physics analyses at ATLAS potentially profiting from a
1436 better electron/photon classification, such as mono-photon or di-photon SUSY
1437 analyses.

1438 Tuning the threshold to achieve the maximum significance yields a model with a
1439 signal efficiency of > 99 % and a DY background rejection of ~ 55 %. The effectiveness
1440 of the model is evident in the background – $m_{\gamma\gamma}$ plots (Figure 4.13), where the
1441 application of the BDT greatly reduces the Drell Yan peak at 90 GeV.

1442 CHAPTER 5

1443 **Conclusions**

1444 Effectively classifying photons and electrons while maintaining a good efficiency on
1445 both particles is a very important task at ATLAS. Many analyses (besides the low
1446 mass $H \rightarrow \gamma\gamma$ search) which rely on efficient and specific identification of photons
1447 and electrons in the detector, are currently taking place.

1448 The BDT developed in this work proved to be very effective, on top of the current
1449 algorithm, in discriminating between photons and electrons. On single particle
1450 simulations samples the BDT is able to correctly classify up to $\sim 90\%$ of ambiguous
1451 particles in the test sample. Requiring a photon efficiency higher than 99%, the
1452 electron rejection can be as high as 64%.

1453 The algorithm has been tested on reconstructed photons from diphoton events
1454 in the low mass region ($[60 \text{ GeV}, 120 \text{ GeV}]$). The BDT was able to perform well even
1455 on this dataset, thus proving to be independent of the training data. By properly
1456 tuning the threshold, a background rejection of more than 50% can be achieved
1457 while maintaining the signal efficiency higher than 99%.

1458 The new classification algorithm can reduce the background from 2 electrons in
1459 the di-photon analysis by approximately a factor of 2. Despite the clear improvement
1460 which can be obtained in the reduction of the Drell-Yan background, the BDT
1461 doesn't bring a sizeable improvement in the low mass $X \rightarrow \gamma\gamma$ search; in this analysis
1462 the dominant background comes from genuine $\gamma\gamma$ QCD events, which are minimally
1463 touched by the algorithm.

1464 The success in the photon and electron classification shows that machine learning
1465 techniques can be very effective in classifying particles in the ATLAS detector. In
1466 addition to the low mass diphoton analyses, many other searches and measurements
1467 could profit from an improved electron-photon classification. Further improvements
1468 can most likely be achieved by implementing other machine learning algorithms, or
1469 by applying the new selection on all candidates without any pre-classification at the
1470 reconstruction level.

¹⁴⁷¹ APPENDIX A

¹⁴⁷² **Tables**

¹⁴⁷³ **A.1 Single Particle Results**

¹⁴⁷⁴ **A.1.1 Efficiencies**

Threshold	ϵ_{ph}	ϵ_{el}
1.0	0.0	0.0
0.987	0.267	0.002
0.979	0.376	0.006
0.972	0.462	0.009
0.965	0.528	0.013
0.958	0.581	0.017
0.95	0.624	0.021
0.943	0.658	0.025
0.935	0.688	0.029
0.927	0.714	0.034
0.918	0.737	0.038
0.909	0.759	0.042
0.899	0.779	0.046
0.888	0.797	0.051
0.878	0.811	0.055
0.866	0.826	0.06
0.853	0.839	0.064
0.839	0.851	0.069
0.826	0.861	0.074
0.811	0.871	0.078
0.796	0.88	0.083
0.779	0.888	0.088
0.761	0.896	0.093
0.741	0.904	0.098
0.721	0.911	0.104
0.702	0.917	0.109
0.68	0.923	0.115
0.656	0.929	0.12
0.631	0.934	0.127
0.603	0.939	0.133
0.577	0.943	0.139
0.548	0.947	0.146
0.519	0.951	0.153

0.488	0.955	0.162
0.457	0.958	0.171
0.429	0.962	0.179
0.399	0.965	0.188
0.37	0.968	0.198
0.342	0.97	0.209
0.314	0.973	0.221
0.287	0.976	0.233
0.26	0.978	0.247
0.236	0.981	0.262
0.211	0.983	0.278
0.188	0.985	0.298
0.166	0.987	0.322
0.146	0.99	0.349
0.127	0.992	0.381
0.109	0.994	0.417
0.086	0.996	0.474
0.06	0.997	0.548
0.019	0.999	0.73
0	1.0	1.0

¹⁴⁷⁵ **A.2 Physics Sample results**

¹⁴⁷⁶ **A.2.1 Efficiencies and Significance**

Threshold	$\varepsilon_{H \rightarrow \gamma\gamma}$	$\varepsilon_{Z \rightarrow ee}$	Significance
0.0	1.0	1.0	14.184
0.02	0.999	0.833	14.21
0.04	0.998	0.758	14.219
0.06	0.997	0.706	14.223
0.08	0.996	0.66	14.227
0.1	0.995	0.614	14.23
0.12	0.994	0.562	14.232
0.14	0.992	0.516	14.233
0.16	0.991	0.475	14.234
0.18	0.991	0.442	14.236
0.2	0.99	0.414	14.237
0.22	0.989	0.392	14.236
0.24	0.988	0.372	14.235
0.26	0.988	0.356	14.233
0.28	0.987	0.339	14.229
0.3	0.985	0.324	14.224
0.32	0.984	0.309	14.218
0.34	0.983	0.297	14.211
0.36	0.982	0.286	14.204
0.38	0.981	0.275	14.199
0.4	0.979	0.268	14.192

0.42	0.978	0.259	14.185
0.44	0.976	0.251	14.174
0.46	0.975	0.244	14.166
0.48	0.974	0.237	14.157
0.5	0.972	0.23	14.146
0.52	0.97	0.224	14.134
0.54	0.969	0.219	14.125
0.56	0.967	0.213	14.114
0.58	0.965	0.207	14.101
0.6	0.962	0.202	14.082
0.62	0.96	0.197	14.063
0.64	0.956	0.192	14.039
0.66	0.953	0.187	14.015
0.68	0.949	0.182	13.989
0.7	0.946	0.177	13.965
0.72	0.941	0.171	13.932
0.74	0.936	0.166	13.893
0.76	0.93	0.16	13.851
0.78	0.923	0.154	13.802
0.8	0.917	0.148	13.753
0.82	0.908	0.142	13.689
0.84	0.897	0.134	13.605
0.86	0.883	0.126	13.504
0.88	0.867	0.117	13.382
0.9	0.843	0.107	13.198
0.92	0.811	0.095	12.949
0.94	0.774	0.083	12.646
0.96	0.725	0.072	12.245
0.98	0.667	0.065	11.745
1.0	0.602	0.062	11.159

¹⁴⁷⁷ **A.2.2 Expected number of events in the [60 GeV , 120 GeV]
¹⁴⁷⁸ mass range**

$m_{\gamma\gamma}$	$H \rightarrow \gamma\gamma$						$Z \rightarrow ee$					
	Without BDT			With BDT			Without BDT			With BDT		
	UU	UC	CC	UU	UC	CC	UU	UC	CC	UU	UC	CC
60	0	0	0	0	0	0	34	67	24	34	46	12
61	0	0	0	0	0	0	35	72	36	35	53	18
62	0	0	0	0	0	0	36	74	38	36	53	18
63	0	0	0	0	0	0	38	87	44	37	65	24
64	0	0	0	0	0	0	47	90	46	47	63	24
65	0	0	0	0	0	0	45	108	52	45	79	27
66	0	0	0	0	0	0	42	87	48	41	59	23
67	0	0	0	0	0	0	42	100	48	42	67	24
68	0	0	0	0	0	0	42	102	64	42	72	34
69	0	0	0	0	0	0	55	104	60	55	68	30

70	0	0	0	0	0	0	47	120	65	46	76	29
71	0	0	0	0	0	0	40	115	66	40	77	29
72	0	0	0	0	0	0	49	127	69	49	83	36
73	0	0	0	0	0	0	49	134	74	49	87	36
74	0	0	0	0	0	0	51	136	78	50	93	39
75	0	0	0	0	0	0	51	138	89	51	88	38
76	0	1	0	0	1	0	52	147	97	52	101	44
77	0	0	0	0	0	0	57	157	105	56	103	43
78	0	1	0	0	1	0	62	172	114	61	109	46
79	0	1	0	0	1	0	67	191	148	66	116	62
80	0	2	0	0	2	0	78	204	147	77	133	51
81	1	3	1	1	3	1	93	264	196	91	168	75
82	2	3	3	2	2	3	114	303	239	113	181	83
83	2	12	5	2	12	5	142	378	352	141	235	118
84	8	26	6	8	26	5	193	506	432	189	316	146
85	19	47	15	19	46	14	238	682	637	235	419	217
86	90	120	32	90	118	31	312	970	858	310	581	269
87	308	313	74	308	308	71	386	1241	1153	381	720	327
88	908	663	127	908	648	123	421	1449	1485	415	815	433
89	1714	1121	205	1714	1102	201	389	1597	1824	382	881	473
90	1654	1199	237	1654	1179	228	333	1497	1895	330	815	489
91	959	808	166	959	798	159	255	1258	1728	252	666	437
92	366	405	96	366	401	94	164	933	1404	159	495	335
93	111	161	50	111	160	49	107	617	1030	104	322	262
94	33	60	19	33	60	18	64	387	728	63	211	175
95	9	28	8	9	27	8	46	266	490	45	146	118
96	2	6	4	2	6	4	32	177	303	31	86	79
97	2	5	1	2	5	1	23	136	205	23	76	50
98	0	2	0	0	2	0	25	97	159	24	49	36
99	0	1	0	0	1	0	15	69	108	15	38	25
100	0	1	0	0	1	0	16	70	102	15	34	25
101	0	0	0	0	0	0	10	54	66	10	26	21
102	0	0	0	0	0	0	10	51	64	10	26	16
103	0	0	0	0	0	0	8	40	55	8	23	13
104	0	0	0	0	0	0	9	41	45	8	20	12
105	0	0	0	0	0	0	8	33	45	8	14	10
106	0	0	0	0	0	0	8	36	30	8	19	7
107	0	0	0	0	0	0	4	29	34	4	15	9
108	0	0	0	0	0	0	8	25	32	8	10	8
109	0	0	0	0	0	0	7	29	30	7	13	7
110	0	0	0	0	0	0	6	20	25	6	9	6
111	0	0	0	0	0	0	4	16	26	4	10	6
112	0	0	0	0	0	0	5	20	20	5	10	5
113	0	0	0	0	0	0	3	21	24	3	10	3
114	0	0	0	0	0	0	2	13	23	2	8	5
115	0	0	0	0	0	0	2	18	13	2	9	3
116	0	0	0	0	0	0	2	18	19	2	8	4
117	0	0	0	0	0	0	4	17	15	4	8	4

$$\left\| \begin{array}{ccc|ccc|ccc|ccc} 118 & 0 & 0 & 0 & 0 & 0 & 2 & 14 & 17 & 2 & 5 & 4 \\ 119 & 0 & 0 & 0 & 0 & 0 & 3 & 11 & 15 & 3 & 5 & 3 \end{array} \right\|$$

1479 APPENDIX B
1480

Code Snippets

1481 B.1 Model Training

1482 B.1.1 Training Sample Preparation

```
1483 #PARTICLES ARE LOADED FROM FILE
1484 PARTICLES = 'electron', 'photon'
1485 prefix = {'electron': 'el_', 'photon': 'ph_'}
1486 
1487 dfs = []
1488 for reco, true in tqdm(itertools.product(PARTICLES, PARTICLES), total=len(PARTICLES)
1489     ) ** 2):
1490     index = RecoTrue(reco, true)
1491     dfs[index] = load_sample(true_particle=true, reco_particle=reco)
1492 
1493 #PT CUT TO GET RID OF PILE-UP
1494 for reco, true in itertools.product(PARTICLES, PARTICLES):
1495     index = RecoTrue(reco, true)
1496     dfs[index]['%spT'%prefix[reco]] = dfs[index]['%scl_E' % prefix[reco]] / \
1497         np.cosh(dfs[index]['%scl_eta' % prefix[reco]])
1498 
1499 dfs[index] = dfs[index].query('%spT>20E3' % prefix[reco])
1500 
1501 #CUT IN DELTA(PT) TO IMPROVE SAMPLE QUALITY
1502 for reco, true in itertools.product(PARTICLES, PARTICLES):
1503     index = RecoTrue(reco, true)
1504     dfs[index] = dfs[index].query('(%spt-%struth_pt)/%struth_pt < 0.1' % \
1505         (prefix[reco], prefix[reco], prefix[reco]) )
1506 
1507 #MERGING TRUE PARTICLES OF SAME TYPE, ASSOCIATING AMBIGUOUS
1508 p = dfs[RecoTrue('photon', 'photon')].query('ph_is_ambiguous==True')
1509 e = dfs[RecoTrue('electron', 'photon')].query('el_is_ambiguous==True')
1510 df_true_photon = pd.merge(
1511     p, e, suffixes=('_ph', '_el'),
1512     left_on=['EventNumber', 'ambiguos_index'],
1513     right_on=['EventNumber', 'index'],
1514 )
1515 df_true_photon['particle']=1
1516 
1517 p = dfs[RecoTrue('photon', 'electron')].query('ph_is_ambiguous==True')
1518 e = dfs[RecoTrue('electron', 'electron')].query('el_is_ambiguous==True')
1519 df_true_electron = pd.merge(
1520     e, p, suffixes=('_ph', '_el'),
1521     right_on=['EventNumber', 'ambiguos_index'],
1522     left_on=['EventNumber', 'index'],
1523 )
1524 df_true_electron['particle']=0
1525 
1526 #CHANGE PHOTON-CONVERSION RELATED COLUMNS TO NAN IF CONVFLAG==0 (NOT CONVERTED)
1527 df_true_photon.loc[
1528     df_true_photon['ph_convFlag']==0,
1529     ['ph_Rconv', 'ph_pt1conv', 'ph_pt2conv', 'ph_ptconv',
1530     'ph_zconv', 'ph_convtrk1nPixHits', 'ph_convtrk1nSCTHits',
1531     'ph_convtrk2nPixHits', 'ph_convtrk2nSCTHits']
1532 ]=np.nan
1533 
1534 df_true_electron.loc[
1535     df_true_electron['ph_convFlag']==0,
```

```

1536:2          ['ph_Rconv', 'ph_pt1conv', 'ph_pt2conv', 'ph_ptconv',
1537:3          'ph_zconv', 'ph_convtrkinPixHits', 'ph_convtrkinSCTHits',
1538:4          'ph_convtrk2nPixHits', 'ph_convtrk2nSCTHits']
1539:5      ]=np.nan
1540:6
1541:7 #CONCATENATING INTO SINGLE DATASET
1542:8 data = pd.concat([df_true_photon,df_true_electron], sort=False)
1543:9
1544:0 #SELECTING TRAINING COLUMNS
1545:1 data = data[[
1546:2          #TRUTH
1547:3          'particle',
1548:4          #GENERAL FEATURES
1549:5          'ph_cl_eta','ph_cl_phi','ph_cl_E', 'ph_pT',
1550:6          #ELECTRON FEATURES
1551:7          'el_track_ep','el_tracketa','el_trackpt','el_trackz0',
1552:8          'el_trkPixelHits','el_trkSiHits','el_refittedTrack_qoverp',
1553:9          #PHOTON FEATURES
1554:0          'ph_convFlag','ph_convtrkinPixHits', 'ph_convtrkinSCTHits',
1555:1          'ph_convtrk2nPixHits', 'ph_convtrk2nSCTHits', 'ph_zconv',
1556:2          'ph_Rconv', 'ph_pt1conv', 'ph_pt2conv', 'ph_ptconv'
1557:3      ]]
1558:4
1559:5 data

```

	particle	ph_cl_eta	ph_cl_phi	ph_cl_E	el_track_ep	...	ph_zconv	ph_Rconv	ph_pt1conv	ph_pt2conv	ph_ptconv
0	1	0.045510	2.917993	3.659295e+04	0.990474	...	-13.872808	94.889420	3.011874e+04	4.968216e+03	3.583185e+04
1	1	-2.376774	-0.979871	8.079536e+05	0.337265	...	-300.552338	48.711056	1.889423e+04	1.044111e+06	4.636859e+05
2	1	-1.863140	-1.275137	4.081177e+05	67.726112	...	-1994.051636	638.952332	0.000000e+00	0.000000e+00	1.359827e+04
3	1	1.210259	-3.101638	1.019196e+05	25.692707	...	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
4	1	-2.459492	-2.749031	6.609997e+05	142.613861	...	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
...
661087	0	2.099148	-2.230953	2.136862e+04	5.818444	...	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
661088	0	-2.146459	-1.271842	2.394364e+04	9.243293	...	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
661089	0	-1.390564	-0.996183	3.698126e+05	11.131820	...	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00
661090	0	-2.405412	-2.145972	1.043027e+06	0.162821	...	-314.064941	35.318264	1.414834e+06	8.854499e+04	1.224421e+06
661091	0	-1.343288	-1.710631	4.555342e+05	1.211997	...	-42.389488	35.005665	2.032854e+05	3.240056e+04	2.093317e+05

1866130 rows × 21 columns

Figure B.1: Pandas Dataframe with the prepared Single Particle sample

1560 B.1.2 LightGBM Dataset Preparation

```

1561:1 #IMPORTING LIBRARIES
1562:2 import lightgbm as lgb
1563:3 from sklearn.model_selection import train_test_split as split
1564:4
1565:5 #PREPARING DATASET
1566:6 x_train, x_test, y_train, y_test = split(
1567:7                                     data.drop(columns=['particle']),
1568:8                                     data['particle'],
1569:9                                     test_size = 0.2,
1570:0                                     )
1571:1
1572:2 lgb_train=lgb.Dataset(x_train, y_train, free_raw_data=False)
1573:3 lgb_test=lgb.Dataset(x_test, y_test, reference=lgb_train, free_raw_data=False)

```

1574 B.1.3 Hyperparameter Optimization

```

1575:1 #IMPORTING HYPEROPT LIBRARIES
1576:2 from hyperopt import fmin, hp, tpe, Trials, STATUS_OK, STATUS_FAIL
1577:3 from hyperopt.pyll import scope as ho_scope
1578:4 from hyperopt.pyll.stochastic import sample as ho_sample

```

```

1579 5
1580 6 #DEFINING DICT WITH HYPERPARAMETERS TO BE OPTIMIZED
1581 7 hp_space = {
1582 8     'num_leaves' : ho_scope.int(hp.quniform(
1583 9                     'num_leaves',
1584 0                     low=5, high=100, q=1
1585 1             )
1586 2         ),
1587 3     'feature_fraction' : hp.uniform(
1588 4                     'feature_fraction',
1589 5                     low=0.65, high=1
1590 6             ),
1591 7     'bagging' : hp.uniform(
1592 8                     'bagging',
1593 9                     low=0.65, high=1
1594 0             ),
1595 1     'bagging_freq' : hp.uniform(
1596 2                     'feature_fraction',
1597 3                     low=0.65, high=1
1598 4             ),
1599 5     'learning_rate' : hp.uniform(
1600 6                     'learning:rate',
1601 7                     low=0.01, high=0.1
1602 8             ),
1603 9 }
1604 0
1605 1 #OTHER HYPERPARAMETERS
1606 2 base_params = {
1607 3     'metric' : 'xentropy',
1608 4     'objective' : 'xentropy',
1609 5     'bagging_seed' : 42 ,
1610 6     'feature_fraction_seed' : 42,
1611 7     'is_unbalance' : True,
1612 8 }
1613 9
1614 0 params = base_params.update(hp_space)
1615 1
1616 2 #FUNCTION TO BE MINIMIZED (XENTROPY OF TEST_SET)
1617 3 def best_xentropy(hyper_params):
1618 4     evals_result = {}
1619 5     gbm = lgb.train(
1620 6         hyper_params,
1621 7         lgb_train,
1622 8         num_boost_round=2000,
1623 9         early_stopping_rounds=5,
1624 0         valid_sets=[lgb_train, lgb_test],
1625 1         evals_result=evals_result,
1626 2         verbose_eval=False
1627 3     )
1628 4     results = {
1629 5         'loss' : evals_result['valid_1']['xentropy'][-1],
1630 6         'status' : STATUS_OK,
1631 7         'space' : hps
1632 8     }
1633 9
1634 0     return results
1635 1
1636 2 #TRIALS OBJECT TO KEEP TRACK OF CHANGES DURING OPTIMIZATION
1637 3 Trials = Trials()
1638 4
1639 5 #HYPERPARAMETER OPTIMIZATION
1640 6 best_params_ = fmin(
1641 7     best_xentropy,
1642 8     space=params,
1643 9     trials=Trials,
1644 0     max_evals=300,
1645 1     algo=tpe.suggest,
1646 2 )
1647 B.1.4 BDT Training
1648 1 #USING OPTIMIZED HYPERPARAMETERS
1649 2 params = base_params.update(best_params)

```

```

1650 3
1651 4 #BDT TRAINING
1652 5 evals_result = {}
1653 6 gbm = lgb.train(
1654 7         params,
1655 8         lgb_train,
1656 9         early_stopping_rounds=5,
1657 0         valid_sets=[lgb_train, lgb_test],
1658 1         evals_result=evals_result,
1659 2         verbose_eval=10,
1660 3     )
1661 4
1662 5 #SAVING TRAINED MODEL
1663 6 gbm.save_model('my_model.txt')

1664 B.1.5 Training Analysis

1665 1 #IMPORTANCE PLOTS
1666 2 lgb.plot_importance(gbm, importance_type='gain')
1667 3 plt.title('Importance - Gain')
1668 4 plt.show()
1669 5
1670 6 lgb.plot_importance(gbm, importance_type='split')
1671 7 plt.title('Importance - Split')
1672 8 plt.show()

1673 1 #XENTROPY PER ITERATION ROUND
1674 2 def render_metric(metric_name):
1675 3     ax = lgb.plot_metric(evals_result, metric=metric_name, figsize=(10,5))
1676 4     plt.title(metric_name)
1677 5     plt.xlabel('Iteration')
1678 6     plt.show()
1679 7
1680 8 render_metric('xentropy')

```

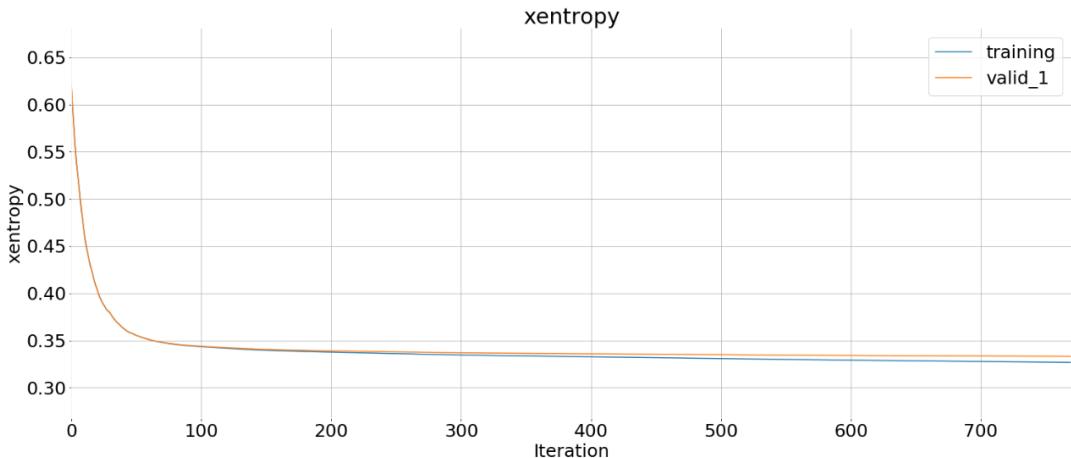


Figure B.2: Metric variation during training¹

1681 B.1.6 ROC Analysis

```

1682 1 #GETTING SCORES FOR TEST SET AND MERGING INTO A RESULTS DATAFRAME
1683 2 y_score = (gbm.predict(x_test))
1684 3
1685 4 test_results = x_test.copy(deep=True)
1686 5 test_results['particle'] = y_test['particle']
1687 6 test_results['score'] = y_score
1688 7 test_results

```

¹All plots were made using the Python plotting library Matplotlib [50]

```

1689 1 #PLOTTING SCORES DISTRIBUTION
1690 2 bins = np.linspace(0,1,101)
1691 3 plt.figure(figsize=[12,6])
1692 4 test_results.loc[test_results['particle'] == 0]['score'].hist(
1693 5                                         bins=bins,
1694 6                                         histtype='step',
1695 7                                         density=True,
1696 8                                         color='blue',
1697 9                                         label='ElectronScores',
1698 0                                         lw=1.3
1699 1                                         )
1700 2 test_results.loc[test_results['particle'] == 1]['score'].hist(
1701 3                                         bins=bins,
1702 4                                         histtype='step',
1703 5                                         density=True,
1704 6                                         color='red',
1705 7                                         label='PhotonScores',
1706 8                                         lw=1.3
1707 9                                         )
1708 0 plt.title('Scores')
1709 1 plt.legend(loc='best')

1710 1 #PLOTTING ROC CURVE AND COMPUTING EFFICIENCIES
1711 2 from sklearn.metrics import roc_curve, auc
1712 3 def plot_roc(y_test, y_score):
1713 4     fpr, tpr, thresholds = roc_curve(y_test, y_score)
1714 5     roc_auc = auc(fpr, tpr)
1715 6
1716 7     plt.figure(figsize=[15,6])
1717 8     plt.plot(fpr,tpr)
1718 9     plt.plot([0,1],[0,1],color='red',lw=1.5)
1719 0     plt.title('ROC curve BDT')
1720 1     plt.ylabel('Photon efficiency')
1721 2     plt.xlabel('Electron efficiency')
1722 3     plt.show()
1723 4
1724 5     print('ROC AUC =', roc_auc)
1725 6
1726 7 plot_roc(y_test,y_score)

1727 1 #PLOTTING EFFICIENCIES
1728 2 plt.figure(figsize=[15,6])
1729 3 plt.plot(
1730 4         df_roc['Threshold'], df_roc['FPR'],
1731 5         color='blue', lw=2,
1732 6         label='Electron Efficiency'
1733 7         )
1734 8 plt.plot(
1735 9         df_roc['Threshold'], df_roc['TPR'],
1736 0         color='red', lw=2,
1737 1         label='Photon Efficiency'
1738 2         )
1739 3 plt.xlabel('threshold')
1740 4 plt.legend()
1741 5 plt.show()

```

1742 B.2 Physics Application

1743 B.2.1 Physics Sample Preparation

```

1744 1 #FUNCTION TO LOAD RAW SAMPLE FROM FILE (OR LIST OF FILES)
1745 2 def load_sample(path, file_name, features):
1746 3
1747 4     if type(file_name)==list:
1748 5         df_list = []
1749 6         for file in file_name:
1750 7             df_list.append(load_sample(path, file, features))
1751 8         return pd.concat(df_list)
1752 9
1753 0     else:
1754 1

```

```

1755:2     filename = os.path.join(path, file_name)
1756:3
1757:4     #LOADING DATAFRAME
1758:5     df_full = uproot.open(filename)[ "CollectionTree" ].pandas.df(features)
1759:6
1760:7     #ADDING INDEX IDENTIFYING SOURCE FILE FOR FUTURE DATAFRAME MERGING
1761:8     df_full['fileName'] = file_name
1762:9     df_full.set_index('fileName', append=True, inplace=True)
1763:0     df_full = df_full.reorder_levels(['fileName', 'entry', 'subentry'])
1764:1
1765:2     print('Loaded from file:', file_name)
1766:3
1767:4     return df_full

1768:1 #FUNCTION TO CHANGE CONVERSIONFLAG VALUES FROM [0,5] TO [0,2]
1769:2 def compute_convFlag(conversionFlag):
1770:3     if conversionFlag == 3:
1771:4         return 2
1772:5     elif conversionFlag != 0:
1773:6         return 1
1774:7     else:
1775:8         return 0
1776:9
1777:0 #FUNCTION TO PREPARE SAMPLE FOR ANALYSIS
1778:1 def clean_sample(sample_raw, bdt_features):
1779:2
1780:3     #LOW MASS HYY SELECTION
1781:4     sample = sample_raw.loc[sample_raw['HGamEventInfoAuxDyn.isPassedLowMyy']==1]
1782:5
1783:6     #FIXING COLUMN NAMES
1784:7     columnsEdited = []
1785:8     for col in sample.columns:
1786:9         if col.find('HGamPhotonsAuxDyn.') != -1:
1787:0             if col.find('el_') == -1:
1788:1                 columnsEdited.append('ph_' + col.replace('HGamPhotonsAuxDyn.', ''))
1789:2             else:
1790:3                 columnsEdited.append(
1791:4                     col.replace('HGamPhotonsAuxDyn.', '') \
1792:5                         .replace('amb_', ''))
1793:6
1794:7         elif col.find('HGamEventInfoAuxDyn') != -1:
1795:8             columnsEdited.append(col.replace('HGamEventInfoAuxDyn.', ''))
1796:9         else:
1797:0             columnsEdited.append(col)
1798:1
1799:2     sample.columns = columnsEdited
1800:3
1801:4     sample = sample.rename(
1802:5         columns={
1803:6             "el_track_eta" : "el_tracketa",
1804:7             "el_track_pt": "el_trackpt",
1805:8             "ph_conversionType" : "ph_convFlag",
1806:9             "ph_phi": "ph_cl_phi"
1807:0         }
1808:1     )
1809:2
1810:3 #APPLYING CONVFLAG FUNCTION
1811:4     sample['ph_convFlag'] = sample['ph_convFlag'].map(compute_convFlag)
1812:5
1813:6 #ADDING COLUMNS TO IDENTIFY AMBIGUOUS PARTICLES
1814:7     sample['is_amb'] = sample.apply(
1815:8         lambda row: False \
1816:9             if (row['el_tracketa']==-999.) &
1817:0             (row['el_refittedTrack_qoverp']==-999.) \
1818:1             else True,
1819:2             axis=1
1820:3         )
1821:4
1822:5
1823:6 #ADDING NP.NAN TO CONVERSION-RELATED COLUMNS WHEN CONVFLAG==0 (NOT CONVERTED)
1824:7     sample.loc[
1825:8         sample['ph_convFlag']==0,
1826:9         [

```

```

18260      'ph_Rconv', 'ph_pt1conv', 'ph_pt2conv', 'ph_ptconv',
18261      'ph_zconv', 'ph_convtrkinPixHits', 'ph_convtrkinSCTHits',
18262      'ph_convtrk2nPixHits','ph_convtrk2nSCTHits'
18263      ]
18264      ] = np.nan
18265
18266 #ONLY CONSIDERING THE TWO PARTICLES PER EVENT WITH HIGHEST PT
18267 sample = sample.drop(2,level='subentry')
18268
18269 #ADDING WEIGHT COLUMN
18270
18271 weightLumi = {      #DICT WITH WEIGHTS FOR DIFFERENT MC SAMPLES
18272     'mc16a' : 36215.,
18273     'mc16d' : 44307.,
18274     'mc16e' : 58450.
18275 }
18276
18277 sample['weight'] = sample['weight'] * \
18278             sample['crossSectionBRfilterEff'] * \
18279             [
18280                 weightLumi[index[0][4:9]]\
18281                 for index in sample.index.tolist()
18282             ]
18283
18284 #ONLY KEEPING COLUMNS USED BY THE BDT
18285 sample = sample[bdt_features + ['weight','is_amb', 'm_yy']]
18286
18287 #RETURNING PREPARED SAMPLE
18288 return sample
18289
18290 #FUNCTION TO GET SUM OF WEIGHTS IN SAMPLE FOR NORMALIZATION
18291 def get_hist_cutflow_sum(path, files):
18292
18293     if type(files) == list:
18294         tot = 0
18295         for file in files:
18296             tot += get_hist_cutflow_sum(path, file)
18297
18298     return tot
18299
18300 else:
18301     filein = ROOT.TFile.Open(path+'/'+files)
18302     for k in list(filein.GetListOfKeys()):
18303         if "_noDalitz_weighted" in k.GetName():
18304             m_histo_name = k.GetName()
18305             break
18306     hist_cutflow = filein.Get(m_histo_name)
18307     norm_weight = hist_cutflow.GetBinContent(3)* \
18308                 (hist_cutflow.GetBinContent(1) / \
18309                  float(hist_cutflow.GetBinContent(2)))
18310
18311 return norm_weight

18312 #FEATURES USED BY THE BDT
18313 featuresBDT = [
18314     #GENERAL FEATURES
18315     'ph_cl_eta','ph_cl_phi','ph_cl_E',
18316     #ELECTRON FEATURES
18317     'el_track_ep','el_tracketa','el_trackpt','el_trackz0',
18318     'el_trkPixelHits','el_trkSiHits','el_refittedTrack_qoverp',
18319     #PHOTON FEATURES
18320     'ph_convFlag','ph_convtrkinPixHits','ph_convtrkinSCTHits',
18321     'ph_convtrk2nPixHits','ph_convtrk2nSCTHits','ph_zconv',
18322     'ph_Rconv', 'ph_pt1conv','ph_pt2conv','ph_ptconv'
18323 ]
18324
18325 #FEATURES TO LOAD FROM MxAOD ROOT SAMPLE
18326 features_toLoad = [
18327     #ALL PHOTON/ELECTRON FEATURES
18328     'HGamPhotonsAuxDyn*',
18329     #EVENT INFO FEATURES
18330     'HGamEventInfoAuxDyn.isPassedLowMyy',
18331     'HGamEventInfoAuxDyn.isPassed',

```

```

18991     'HGamEventInfoAuxDyn.weight',
19002     'HGamEventInfoAuxDyn.crossSectionBRfilterEff',
19013     'HGamEventInfoAuxDyn.m_yy'
19024   ]
19035
19046 #ROOT FILES TO LOAD
19057 Hyy_files = ['Hyy_mc16a_1.root', 'Hyy_mc16d_1.root', 'Hyy_mc16e_1.root']
19068 Zee_files = [
19079     'Zee_mc16a_1.root', 'Zee_mc16a_2.root', 'Zee_mc16a_3.root',
19080     'Zee_mc16a_4.root', 'Zee_mc16d_10.root', 'Zee_mc16d_11.root',
19091     'Zee_mc16d_12.root', 'Zee_mc16d_13.root', 'Zee_mc16d_14.root',
19102     'Zee_mc16d_15.root', 'Zee_mc16d_16.root', 'Zee_mc16d_17.root',
19113     'Zee_mc16d_18.root', 'Zee_mc16d_19.root', 'Zee_mc16d_1.root',
19124     'Zee_mc16d_20.root', 'Zee_mc16d_21.root', 'Zee_mc16d_22.root',
19135     'Zee_mc16d_23.root', 'Zee_mc16d_24.root', 'Zee_mc16d_25.root',
19146     'Zee_mc16d_2.root', 'Zee_mc16d_3.root', 'Zee_mc16d_4.root',
19157     'Zee_mc16d_5.root', 'Zee_mc16d_6.root', 'Zee_mc16d_7.root',
19168     'Zee_mc16d_8.root', 'Zee_mc16d_9.root', 'Zee_mc16e_10.root',
19179     'Zee_mc16e_11.root', 'Zee_mc16e_12.root', 'Zee_mc16e_13.root',
19180     'Zee_mc16e_14.root', 'Zee_mc16e_15.root', 'Zee_mc16e_16.root',
19191     'Zee_mc16e_17.root', 'Zee_mc16e_18.root', 'Zee_mc16e_19.root',
19202     'Zee_mc16e_1.root', 'Zee_mc16e_20.root', 'Zee_mc16e_21.root',
19213     'Zee_mc16e_2.root', 'Zee_mc16e_3.root', 'Zee_mc16e_4.root',
19224     'Zee_mc16e_5.root', 'Zee_mc16e_6.root', 'Zee_mc16e_7.root',
19235     'Zee_mc16e_8.root', 'Zee_mc16e_9.root'
19246   ]
19257
19268 #LOADING AND PREPARING HYY SAMPLE
19279 sampleHyy_raw = load_sample(
19280     '/users2/etroncon/samples/Hyy',
19291     Hyy_files,
19302     features_toLoad
19313   )
19324 norm_weight_Hyy = get_hist_cutflow_sum('/users2/etroncon/samples/Hyy', Hyy_files)
19335
19346 sampleHyy = clean_sample(sampleHyy_raw, featuresBDT)
19357 sampleHyy['weight'] = sampleHyy['weight'] / norm_weight_Hyy
19368
19379
19380 #LOADING AND PREPARING ZEE SAMPLE
19391 sampleZee_raw = load_sample(
19402     '/users2/etroncon/samples/Zee',
19413     Zee_files,
19424     features_toLoad
19435   )
19446 norm_weight_Zee = get_hist_cutflow_sum('/users2/etroncon/samples/Zee', Zee_files)
19457
19468 sampleZee = clean_sample(sampleZee_raw, featuresBDT)
19479 sampleZee['weight'] = sampleZee['weight'] / norm_weight_Zee

```

	fileName	entry	subentry	ph_cl_eta	ph_cl_phi	ph_cl_E	ph_pT	el_track_ep	...	ph_pt2conv	ph_ptconv	weight	is_amb	m_yy
		0		-0.303503	1.556904	49582.562500	47383.425781	-1.000000	...	NaN	NaN	0.370623	False	88924.859375
		1		-0.659683	-0.955010	44485.378906	36296.910156	-1.000000	...	NaN	NaN	0.370623	False	88924.859375
Hyy_mc16a_1.root	4	1	-1.092372	-2.894874	59795.582031	36056.664062	-1.000000	...	NaN	NaN	0.257440	False	93662.523438	
		5		0.082631	1.809795	55996.199219	55805.578125	-1.000000	...	NaN	NaN	0.330581	False	91554.765625
		1	1.123537	-0.261706	48773.574219	28683.248047	-1.000000	...	NaN	NaN	0.330581	False	91554.765625	

		25653	0	-0.401389	2.205446	72420.882812	66954.445312	97.098984	...	NaN	NaN	0.623626	True	89758.476562
		1	-0.380510	0.863409	66143.414062	61627.835938	-1.000000	...	NaN	NaN	0.623626	False	89758.476562	
Hyy_mc16e_1.root	25655	1	0.652494	1.593534	55416.929688	45403.808594	-1.000000	...	NaN	NaN	0.601070	False	90113.312500	
		25657	0	0.627023	2.760673	45193.308594	37563.906250	-1.000000	...	NaN	NaN	0.594928	False	88848.562500
		1	2.228465	-1.156749	116890.421875	24888.056641	22.632645	...	NaN	NaN	0.594928	True	88848.562500	

42951 rows × 23 columns

Figure B.3: Pandas Dataframe with the prepared $H \rightarrow \gamma\gamma$ sample

1948 B.2.2 Event Classification with the trained BDT

```

19491 #IMPORTING LIBRARIES AND LOADING TRAINED MODEL
19502 import lightgbm as lgb
19513 gbm = lgb.Booster(model_file='my_model.txt')
19524
19535 #GETTING SCORES FOR AMBIGUOUS PARTICLES
19546 df = sampleHyy.loc[sampleHyy["is_amb"]]
19557 df['score'] = gbm.predict(df[featuresBDT])
19568
19579 Hyy = sampleHyy.merge(
195810                 df[["score"]],
195911                 left_index=True, right_index=True, how="outer"
196012             )
196113
196214
19635 df = sampleZee.loc[sampleZee["is_amb"]]
19646 df['score'] = gbm.predict(df[featuresBDT])
19657
19668 Zee = sampleZee.merge(
19679                 df[["score"]],
196810                 left_index=True, right_index=True, how="outer"
196911             )
197012
197113 #FUNCTION TO GET A SINGLE SCORE PER EVENT
197214 def event_score(df):
197315     if df.iloc[0]['is_amb'] & df.iloc[1]['is_amb']:
197416         return df['score'].min()
197517     elif df.iloc[0]['is_amb']:
197618         return df.iloc[0]['score']
197719     elif df.iloc[1]['is_amb']:
197820         return df.iloc[1]['score']
197921     else:
198022         return 1
198123
198214 #FUNCTION TO GET THE CONVERSION CATEGORY OF THE EVENT
198315 def event_conv_type(df):
198416     if (df.iloc[0]['ph_convFlag'] == 0) & (df.iloc[1]['ph_convFlag'] == 0) :
198517         return 'UU'
198618     elif (df.iloc[0]['ph_convFlag'] != 0) & (df.iloc[1]['ph_convFlag'] != 0):
198719         return 'CC'
198820     else:
198921         return 'UC'
199022
199113 #GETTING RESULTS FOR HYY EVENTS
199214 resHyy = []
199315 for idx,df in Hyy.groupby(['fileName','entry']):
199416     resHyy.append(
199517         {
199618             'idx':idx,
199719             'score': event_score(df),
199820             'weight': df.iloc[0]['weight'],
199921             'm_yy': df.iloc[0]['m_yy'],
200022             'convType':event_conv_type(df)
200123         }
200224     )
20035 resHyy = pd.DataFrame(resHyy)
20046 resHyy.index = pd.MultiIndex.from_tuples(
20057                                         resHyy['idx'],
20068                                         names=['fileName', 'entry']
20079                                     )
20080 resHyy = resHyy.drop(columns=['idx'])
20091
20102 #GETTING RESULTS FOR ZEE EVENTS
20113 resZee = []
20124 for idx,df in Zee.groupby(['fileName','entry']):
20135     resZee.append(
20146         {
20157             'idx':idx,
20168             'score': event_score(df),
20179             'weight': df.iloc[0]['weight'],
20180             'm_yy': df.iloc[0]['m_yy'],
20191             'convType':event_conv_type(df)

```

```

2020 2
2021 3
2022 4 resZee = pd.DataFrame(resZee)
2023 5 resZee.index = pd.MultiIndex.from_tuples(
2024 6                                         resZee['idx'],
2025 7                                         names=['fileName', 'entry']
2026 8
2027 9 resZee = resZee.drop(columns=['idx'])
2028 0
2029 1 #MERGING RESULTS
2030 2 resHyy['true'] = 1
2031 3 resZee['true'] = 0
2032 4
2033 5 results = pd.DataFrame.append(resHyy, resZee)

```

		true	score	weight	m_yy	convType
fileName	entry					
	0	1	1.000000	0.370623	88924.859375	UU
	4	1	1.000000	0.257440	93662.523438	UC
Hyy_mc16a_1.root	5	1	1.000000	0.330581	91554.765625	UU
	8	1	1.000000	0.366876	89297.414062	UU
	11	1	0.838103	0.304721	89408.171875	CC
...
	408	0	0.121851	0.377709	90070.867188	CC
	415	0	0.607548	0.356035	59730.949219	UU
Zee_mc16e_9.root	417	0	0.152726	0.498538	90009.171875	CC
	418	0	0.835448	0.460804	76313.484375	UC
	419	0	1.000000	0.468607	91575.093750	UC

168393 rows × 5 columns

Figure B.4: Pandas Dataframe with the physics results

B.2.3 Analysis

```

2034 #PLOTTING SCORES DISTRIBUTION
2035 1 plt.figure(figsize=[15,10])
2036 2 bins=np.linspace(0,1,101)
2037 3
2038 4
2039 5 resZee['score'].hist(
2040 6                                     label='Zee scores', histtype='step',
2041 7                                     bins=bins, density=True,
2042 8                                     lw=2, color='blue'
2043 9
2044 0 resHyy['score'].hist(
2045 1                                     label='Hyy scores', histtype='step',
2046 2                                     bins=bins, density=True,
2047 3                                     lw=2, color='red'
2048 4
2049 5
2050 6 plt.legend()
2051 7 plt.show()

2052 1 #FUNZIONI PER CALCOLARE EFFICIENZE E SIGNIFICANZA CON EVENTI PESATI
2053 2
2054 3 def my_rates(res, threshold, weight_sum_ph, weight_sum_el):
2055 4
2056 5     TPR = res.loc[
2057 6             res['score'] >= threshold
2058 7         ]\
2059 8         .query('true==1')['weight'].sum() / weight_sum_ph
2060 9

```

```

2061.0     FPR = res.loc[
2062.1             res['score'] >= threshold
2063.2         ] \
2064.3             .query('true==0')['weight'].sum()/weight_sum_el
2065.4
2066.5     return (TPR,FPR)
2067.6
2068.7 def my_roc(res, number_of_thresholds=4000, windowHalfWidth=2000):
2069.8     res_local = res.copy(deep=True)
2070.9     res_local = res_local.loc[abs(res_local['m_yy']) - 90000] < windowHalfWidth
2071.0
2072.1     weight_sum_ph = res_local.query('true == 1')['weight'].sum()
2073.2     weight_sum_el = res_local.query('true == 0')['weight'].sum()
2074.3
2075.4     rates = []
2076.5     for threshold in np.linspace(0,1, number_of_thresholds):
2077.6         TPR, FPR = my_rates(res_local,threshold, weight_sum_ph, weight_sum_el)
2078.7         rates.append({'Threshold':threshold, 'TPR': TPR, 'FPR': FPR})
2079.8
2080.9     df_rates = pd.DataFrame(rates)
2081.0
2082.1 #SIGNIFICANZA
2083.2
2084.3     f = ROOT.TFile("bkg_analLowMass.root")
2085.4     hist_background_yy = f.Get("hyymcSR_incl_nominal_All")
2086.5     yyBackgroundEvents = hist_background_yy.\
2087.6                 Integral(
2088.7                     120-int(windowHalfWidth/250),
2089.8                     120 + int(windowHalfWidth/250)
2090.9                 ) \
2091.0                 / (weightLumi["mc16a"] + weightLumi["mc16d"]) \
2092.1                 * sum(weightLumi.values())
2093.2     f.Close()
2094.3
2095.4     df_rates['SGN'] = (df_rates['TPR'] * weight_sum_ph) / \
2096.5             np.sqrt(
2097.6                 (df_rates['FPR'] * weight_sum_el) +
2098.7                 (yyBackgroundEvents * df_rates['TPR']))
2099.8
2100.9
2101.0     return df_rates.reset_index(drop=True)
2102.1
2103.2 def my_trapz(x,y):
2104.3     if x.shape != y.shape:
2105.4         print('Dimensioni diverse!')
2106.5     return -1
2107.6
2108.7     tot = 0
2109.8     for i in range(1, len(x)):
2110.9         Y = y[i] + y[i-1]
2111.0         X = x[i] - x[i-1]
2112.1         tot += Y*X/2
2113.2     return tot
2114.3
2115.4 def plot_roc(df_rates):
2116.5
2117.6     plt.figure(figsize=[15,10])
2118.7     plt.plot(df_rates['FPR'],df_rates['TPR'], lw=2)
2119.8     plt.plot([0,1],[0,1],color='red',lw=2)
2120.9     plt.title('ROC curve BDT')
2121.0     plt.ylabel('tpr - Hyy Efficiency')
2122.1     plt.xlabel('fpr - Zee Efficiency')
2123.2     plt.show()
2124.3
2125.4     print ('ROC AUC:', -my_trapz(df_rates['FPR'],df_rates['TPR']))

```


2126 **Bibliography**

2127 **LHC and ATLAS**

- 2128 [1] Lyndon R Evans and Philip Bryant. “LHC Machine”. In: *JINST* 3 (2008).
2129 This report is an abridged version of the LHC Design Report (CERN-2004-
2130 003), S08001. 164 p. DOI: 10.1088/1748-0221/3/08/S08001. URL: <https://cds.cern.ch/record/1129806>.
- 2132 [2] F Gianotti. “Collider physics: LHC”. In: CERN-OPEN-2000-293 (). DOI: 10.
2133 5170/CERN-2000-007.219. URL: <http://cds.cern.ch/record/458489>.
- 2134 [3] “The ALICE experiment at the CERN LHC. A Large Ion Collider Experi-
2135 ment”. In: *JINST* 3 (2008). DOI: 10.1088/1748-0221/3/08/S08002. URL:
2136 <https://cds.cern.ch/record/1129812>.
- 2137 [4] The ATLAS collaboration. “The ATLAS Experiment at the CERN Large
2138 Hadron Collider”. In: *JINST* (2008). DOI: 10.1088/1748-0221/3/08/S08003.
2139 URL: <https://cds.cern.ch/record/1129811>.
- 2140 [5] “The CMS experiment at the CERN LHC. The Compact Muon Solenoid ex-
2141 periment”. In: *JINST* 3 (2008). DOI: 10.1088/1748-0221/3/08/S08004. URL:
2142 <https://cds.cern.ch/record/1129810>.
- 2143 [6] “The LHCb Detector at the LHC”. In: *JINST* 3.LHCb-DP-2008-001. CERN-
2144 LHCb-DP-2008-001 (2008). DOI: 10.1088/1748-0221/3/08/S08005. URL:
2145 <https://cds.cern.ch/record/1129809>.
- 2146 [7] “The TOTEM Experiment at the LHC”. In: arXiv:0905.2936 (2009). URL:
2147 <https://cds.cern.ch/record/1178134>.
- 2148 [8] A Tricomi et al. “The LHCf experiment at the LHC: Physics Goals and Sta-
2149 tus”. In: *Nucl. Phys. B, Proc. Suppl.* 196 (2009). DOI: 10.1016/j.nuclphysbps.
2150 2009.09.005. URL: <https://cds.cern.ch/record/1262049>.
- 2151 [9] James Pinfold. “The MoEDAL experiment at the LHC”. In: *EPJ Web Conf.*
2152 145 (2017). DOI: 10.1051/epjconf/201614512002. URL: <https://cds.cern.ch/record/2288634>.
- 2154 [10] *ATLAS website*. URL: <https://atlas.cern/discover/about>.
- 2155 [12] *ATLAS inner detector: Technical Design Report*, 1. Technical Design Report
2156 ATLAS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/331063>.
- 2157 [13] M Capeans et al. *ATLAS Insertable B-Layer Technical Design Report*. Tech.
2158 rep. CERN-LHCC-2010-013. ATLAS-TDR-19. Sept. 2010. URL: <https://cds.cern.ch/record/1291633>.
- 2160 [14] *ATLAS liquid-argon calorimeter: Technical Design Report*. Technical Design
2161 Report ATLAS. Geneva: CERN, 1996. URL: <https://cds.cern.ch/record/331061>.

- 2163 [15] *ATLAS tile calorimeter: Technical Design Report*. Technical Design Report
 2164 ATLAS. Geneva: CERN, 1996. URL: <https://cds.cern.ch/record/331062>.
- 2165 [16] *ATLAS muon spectrometer: Technical Design Report*. Technical Design Report
 2166 ATLAS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/331068>.
- 2167 [17] Peter Ware Higgs. “Broken symmetries, massless particles and gauge fields”.
 2168 In: *Phys. Lett.* 12 (1964). DOI: 10.1016/0031-9163(64)91136-9. URL: <http://cds.cern.ch/record/641590>.
- 2170 [18] F Englert and R Brout. “Broken symmetry and the masses of gauge vector
 2171 mesons”. In: *Phys. Rev. Lett.* 13 (1964). DOI: 10.1103/PhysRevLett.13.321.
 2172 URL: <http://cds.cern.ch/record/641592>.
- 2173 [19] “Observation of a new particle in the search for the Standard Model Higgs bo-
 2174 son with the ATLAS detector at the LHC”. In: *Phys. Lett. B* 716.arXiv:1207.7214.
 2175 CERN-PH-EP-2012-218 (Aug. 2012). DOI: 10.1016/j.physletb.2012.08.
 2176 020. URL: <https://cds.cern.ch/record/1471031>.
- 2177 [20] “Observation of a new boson at a mass of 125 GeV with the CMS experiment at
 2178 the LHC”. In: *Phys. Lett. B* 716.CMS-HIG-12-028. CMS-HIG-12-028. CERN-
 2179 PH-EP-2012-220 (July 2012). DOI: 10.1016/j.physletb.2012.08.021. URL:
 2180 <https://cds.cern.ch/record/1471016>.
- 2181 [21] *Measurements of the Higgs boson production and decay rates and constraints
 2182 on its couplings from a combined ATLAS and CMS analysis of the LHC pp col-
 2183 lision data at $\sqrt{s} = 7$ and 8 TeV*. Tech. rep. ATLAS-CONF-2015-044. Geneva:
 2184 CERN, Sept. 2015. URL: <https://cds.cern.ch/record/2052552>.
- 2185 [22] *Search for resonances in the 65 to 110 GeV diphoton invariant mass range
 2186 using 80 fb^{-1} of pp collisions collected at $\sqrt{s} = 13 \text{ TeV}$ with the ATLAS
 2187 detector*. Tech. rep. ATLAS-CONF-2018-025. Geneva: CERN, July 2018. URL:
 2188 <https://cds.cern.ch/record/2628760>.
- 2189 [23] S. D. Drell and Tung-Mow Yan. “Massive Lepton Pair Production in Hadron-
 2190 Hadron Collisions at High-Energies”. In: *Phys. Rev. Lett.* 25 (1970), pp. 316–
 2191 320. DOI: 10.1103/PhysRevLett.25.316, 10.1103/PhysRevLett.25.902.2.
- 2192 [45] “GEANT4: A Simulation toolkit”. In: *Nucl. Instrum. Methods Phys. Res., A*
 2193 506.CERN-IT-2002-003. SLAC-PUB-9350. 3 (July 2002), 250–303. 86 p. DOI:
 2194 10.1016/S0168-9002(03)01368-8. URL: <https://cds.cern.ch/record/602040>.
- 2196 [49] Rob Roy Mac Gregor Fletcher et al. *Search for new phenomena in low-mass
 2197 diphoton events with the ATLAS detector at $\sqrt{s} = 13 \text{ TeV}$* . Tech. rep. ATL-
 2198 COM-PHYS-2018-193. Geneva: CERN, Mar. 2018. URL: <https://cds.cern.ch/record/2306589>.

2200

Particle Reconstruction

- 2201 [24] *Electron and photon reconstruction and performance in ATLAS using a dy-
 2202 namical, topological cell clustering-based approach*. Tech. rep. ATL-PHYS-
 2203 PUB-2017-022. CERN, 2017. URL: <https://cds.cern.ch/record/2298955>.

- 2204 [25] W Lampl et al. *Calorimeter Clustering Algorithms: Description and Per-*
 2205 *formance*. Tech. rep. ATL-LARG-PUB-2008-002. ATL-COM-LARG-2008-003.
 2206 Geneva: CERN, Apr. 2008. URL: <https://cds.cern.ch/record/1099735>.
- 2207 [26] T Cornelissen et al. *Concepts, Design and Implementation of the ATLAS New*
 2208 *Tracking (NEWT)*. Tech. rep. ATL-SOFT-PUB-2007-007. ATL-COM-SOFT-
 2209 2007-002. CERN, Mar. 2007. URL: <https://cds.cern.ch/record/1020106>.
- 2210 [27] *Performance of the ATLAS Silicon Pattern Recognition Algorithm in Data*
 2211 *and Simulation at $\sqrt{s} = 7$ TeV*. Tech. rep. ATLAS-CONF-2010-072. Geneva:
 2212 CERN, July 2010. URL: <https://cds.cern.ch/record/1281363>.
- 2213 [28] R. Fruhwirth. “Application of Kalman filtering to track and vertex fitting”.
 2214 In: *Nucl. Instrum. Meth.* A262 (1987), pp. 444–450. DOI: 10.1016/0168-
 2215 9002(87)90887-4.
- 2216 [29] D Wicke. *A New Algorithm for Solving Tracking Ambiguities*. Oct. 1998. URL:
 2217 <https://cds.cern.ch/record/2625731>.
- 2218 [30] T G Cornelissen. “Track Fitting in the ATLAS Experiment”. 2006. URL: <http://cds.cern.ch/record/1005181>.
- 2220 [31] *Particle Identification Performance of the ATLAS Transition Radiation Track-*
 2221 *er*. Tech. rep. ATLAS-CONF-2011-128. CERN, 2011. URL: <https://cds.cern.ch/record/1383793>.
- 2223 [32] “Measurement of the photon identification efficiencies with the ATLAS de-
 2224 tector using LHC Run-1 data”. In: *Eur. Phys. J. C* 76.CERN-EP-2016-110.
 2225 12 (June 2016), 666. 57 p. DOI: 10.1140/epjc/s10052-016-4507-9. URL:
 2226 <https://cds.cern.ch/record/2158117>.
- 2227 [33] “Electron and photon performance measurements with the ATLAS detector
 2228 using the 2015-2017 LHC proton-proton collision data”. In: *JINST* (2019).
 2229 DOI: 10.1088/1748-0221/14/12/P12006. URL: <https://cds.cern.ch/record/2684552>.
- 2231 [34] *Electron efficiency measurements with the ATLAS detector using the 2015*
 2232 *LHC proton-proton collision data*. Tech. rep. CERN, 2016. URL: <https://cds.cern.ch/record/2157687>.
- 2234 [35] “Measurement of the photon identification efficiencies with the ATLAS de-
 2235 tector using LHC Run 2 data collected in 2015 and 2016”. In: *Eur. Phys. J. C*
 2236 (2018). DOI: 10.1140/epjc/s10052-019-6650-6. URL: <https://cds.cern.ch/record/2643391>.
- 2238 [36] “Measurement of the inclusive isolated prompt photon cross section in pp col-
 2239 lisions at $\sqrt{s} = 7$ TeV with the ATLAS detector using 4.6 fb^{-1} ”. In: *Phys. Rev. D* 89.CERN-PH-EP-2013-164. CERN-PH-EP-2013-164 (Nov. 2013), 052004.
 2240 24 p. DOI: 10.1103/PhysRevD.89.052004. URL: <https://cds.cern.ch/record/1625660>.

2243 Machine Learning

- 2244 [37] Thomas M. Mitchell. *Machine Learning*. 1st ed. USA: McGraw-Hill, Inc., 1997.
 2245 ISBN: 0070428077.

- 2246 [38] Jerome Friedman. “Greedy Function Approximation: A Gradient Boosting
2247 Machine”. In: *The Annals of Statistics* (2000). DOI: 10.1214/aos/1013203451.
- 2248 [39] Jerome Friedman. “Stochastic Gradient Boosting”. In: *Computational Statistics-
2249 & Data Analysis* 38 (2002). DOI: 10.1016/S0167-9473(01)00065-2.
- 2250 [40] *XGBoost introduction to boosted trees*. URL: <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>.
- 2251
- 2252 [41] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. “Additive Logistic
2253 Regression: A Statistical View of Boosting”. In: *The Annals of Statistics* 28
2254 (2000). DOI: 10.1214/aos/1016218223.
- 2255 [43] Haijian Shi. “Best-first Decision Tree Learning”. 2007.
- 2256 [44] *LightGBM documentation*. URL: <https://lightgbm.readthedocs.io>.

2257 Used Python Packages

- 2258 [42] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision
2259 Tree”. In: *NIPS*. 2017.
- 2260 [46] Wes McKinney et al. “Data structures for statistical computing in python”.
2261 In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX.
2262 2010, pp. 51–56.
- 2263 [47] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of
2264 Machine Learning Research* 12 (2011), pp. 2825–2830.
- 2265 [48] J. Bergstra, D. Yamins, and D. D. Cox. “Making a Science of Model Search:
2266 Hyperparameter Optimization in Hundreds of Dimensions for Vision Architec-
2267 tures”. In: *Proceedings of the 30th International Conference on International
2268 Conference on Machine Learning - Volume 28*. JMLR.org, 2013.
- 2269 [50] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in
2270 Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- 2271 [51] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.
- 2272 [52] R. Brun and F. Rademakers. “ROOT: An object oriented data analysis frame-
2273 work”. In: *Nucl. Instrum. Meth.* A389 (1997), pp. 81–86. DOI: 10.1016/S0168-
2274 9002(97)00048-X.