Model independent searches for New Physics using Machine Learning at the ATLAS experiment

Fabricio Jimenez

UNIVERSITÉ CLERMONT AUVERGNE

ÉCOLE DOCTORALE DES SCIENCES FONDAMENTALES

THÈSE

présentée pour obtenir le grade de
DOCTEUR D'UNIVERSITÉ
Spécialité: **PARTICULES, INTERACTIONS, UNIVERS**

par

Fabricio JIMÉNEZ

Model independent searches for New Physics using Machine Learning at the ATLAS experiment

---

Thèse soutenue publiquement le 16 septembre 2019, devant la commission d'examen:

	Rapporteur
M. Tommaso DORIGO	*Président du jury, rapporteur*
M. David ROUSSEAU	*Examinatrice*
Mme. Giovanna MENARDI	*Examinatrice*
Mme. Anne YAO LAFOURCADE	*Examinateur*
M. Yann COADOU	*Directeur de thèse*
M. Julien DONINI	

UNIVERSITÉ CLERMONT AUVERGNE

Abstract

École Doctorale de Sciences Fondamentales
Laboratoire de Physique de Clermont

Doctor of Philosophy

Model independent searches for New Physics using Machine Learning at the ATLAS experiment

by JIMÉNEZ Fabricio

We address the problem of model-independent searches for New Physics (NP), at the Large Hadron Collider (LHC) using the ATLAS detector. Particular attention is paid to the development and testing of novel Machine Learning techniques for that purpose. The present work presents three main results. Firstly, we put in place a system for automatic generic signature monitoring within TADA, a software tool from ATLAS. We explored over 30 signatures in the data taking period of 2017 and no particular discrepancy was observed with respect to the Standard Model processes simulations. Secondly, we propose a collective anomaly detection method for model-independent searches for NP at the LHC. We propose the parametric approach that uses a semi-supervised learning algorithm. This approach uses penalized likelihood and is able to simultaneously perform appropriate variable selection and detect possible collective anomalous behavior in data with respect to a given background sample. Thirdly, we present preliminary studies on modelling background and detecting generic signals in invariant mass spectra using Gaussian processes (GPs) with no mean prior information. Two methods were tested in two datasets: a two-step procedure in a dataset taken from Standard Model simulations used for ATLAS General Search, in the channel containing two jets in the final state, and a three-step procedure from a simulated dataset for signal (Z') and background (Standard Model) in the search for resonances in the $t\bar{t}$ invariant mass spectrum case. Our study is a first step towards a method that takes advantage of GPs as a modelling tool that can be applied to several signatures in a more model independent setup.

Acknowledgements

During the last three years, I have been fortunate to interact with many people and institutions, several of which played a critical role in the research presented in this document. I am greatly indebted to all of them.

I would like to start by thanking my laboratory (LPC) and especially the director, Dr. Dominque Pallin, for having welcomed and hosted me as a doctoral student. Also, I am grateful to all former and current members of the ATLAS group that supported me and provided fertile ground for regular discussions since the start of my work here. Prof. Julien Donini, my supervisor, played a crucial role in most aspects of this work; his patient advice and insights have been the key to achieving the conclusion of this thesis, for which I am grateful.

My doctoral program was funded by the AMVA4NewPhysics Innovative Training Network, an EU Marie Sklodowska Curie Action. This gave me the somehow rare freedom to collaborate and interact with colleagues around the globe, as well as the opportunity to take part in enriching training periods and events. I would like to thank the people that supervised me during my training periods: Professors Giovanna Menardi and Bruno Scarpa from the Statistics Department at the University of Padova, Prof. Daniel Whiteson from the University of California at Irvine, and Dr. Ilya Narsky from The Mathworks, Inc. I am grateful for all the discussions and experiences with fellow doctoral students from the Network, in particular to Dr. Grzegorz Kotkowski.

During my qualification task and beyond, Dr. Markus Elsing and Dr. Simone Amoroso from CERN provided very valuable guidance in a wide variety of topics related to the ATLAS collaboration.

Last, but certainly not least, I would like to thank my family for their constant unconditional support, and Gabriela for her love and company during the last years.

This Report is part of a project that has received funding from the **European Union's Horizon 2020 research and innovation programme under grant agreement Nº675440**

Contents

List of Figures

List of Tables

To my family.

Chapter 1

Introduction

The Standard Model (SM) of particle physics is the theory that describes with great success three of the four fundamental interactions present in nature. Since its inception, decades ago, this theory has undergone many experimental tests its predictions have been confirmed to great accuracy. However, it is believed that the SM is not a complete theory but rather a low-energy limit of a more general effective field theory, as there are some both theoretical and experimental aspects that are not satisfactory. This has inspired theorists to hypothesize scenarios in which new symmetries or interactions are introduced, and incentivized experimentalists to probe the SM as far as possible. New Physics (NP) is the generic term used to refer to phenomena beyond the SM description.

Experiments such as CERN's Large Hadron Collider (LHC) as well as several other preceding collider experiments have had the search for NP as one of their main goals. The collaborations such as ATLAS and CMS design and analyze the data from detectors with the aim of finding evidence for NP. Most efforts in searching for NP concentrate in probing a specific parameter space associated to a hypothesis where a given NP theory should show some measurable effects if it is true. Such searches are deemed model dependent, while searches that reduce as much as possible the assumptions on the nature of NP are called model independent. Model dependent searches for NP have so far lead to null results at the LHC, which has set limits in the parameter spaces explored for the corresponding models, and sparked interest in model independent searches and those that automatically explore many generic signatures.

The optimal analysis of the data is a crucial necessity in NP searches. The development and performance of Machine Learning (ML) algorithms has seen a revolution in the last decade, that has translated into an improvement in many High Energy Physics tasks, leading to an increased NP discovery potential.

This thesis focuses on model-independent NP searches, with an emphasis in the study and development of ML methods for that purpose. Among the main challenges addressed are the detection of NP signals that are faint or located in heavily populated background regions; also, reducing spurious detection in the absence of signal is crucial, as well as having a method that handles complex datasets. The methods proposed and studied here provide proofs of concepts for alternatives to existing methods, where our results already show promising paths for further development.

This document is organized as follows. Chapters 2 and 3 are devoted to introduce, respectively, the theoretical context (the SM and beyond) and experimental facilities (the LHC and the ATLAS detector). Then chapter 4 presents TADA, a fast monitoring tool from ATLAS, which is used in this work to automatically monitor many (generic) signatures. A review of the ML methods currently used in High Energy Physics follows in chapter 5. Chapter 6 presents a method that uses Gaussian

Mixture Models to perform anomaly detection in a semi-supervised setup, via a penalized likelihood. A dataset was simulated with SM processes (background) and a benchmark NP scenario (signal) to provide a proof of concept for the method in High Energy Physics. Then, in chapter 7 we present studies of Gaussian Process methods to model background and signal invariant mass spectra from simulated datasets used for NP searches in ATLAS, in two cases: the dijet signature from the General Search and a dataset used for resonant searches decaying into top quark pairs. Finally, conclusions and an outlook are given in 8.

Chapter 2

The Standard Model and Beyond

2.1 Introduction

The Standard Model (SM) of Particle Physics [1–4] is a highly successful theory that describes elementary particles and their interactions at the most fundamental level. Three of the four fundamental forces of nature, namely electromagnetism, the weak, and the strong nuclear forces are described in the SM. Besides gravitation, the interactions comprised in the SM underlie all physical phenomena in nature. This decades-old theory has endured many experimental tests and it is considered today as one of the most successful scientific frameworks.

Particles in the SM can be classified in two kinds, namely *fermions* and *bosons*. This distinction is made on the basis of the quantum-mechanical spin: fermions are particles with half integer spin and obey the Fermi-Dirac statistics, whereas bosons have integer spin and follow the Bose-Einstein statistics. Some properties of the particles contained in the SM are presented in the diagram in figure 2.1[1].

Fermions in the SM come in three *families* or *generations*, these correspond to the three columns of fermions in figure 2.1. Fermion families each contain two quarks, and two leptons; for example, the first generation consists of the *up* and *down* quarks (respectively u and d), the electron (e) and the electron neutrino (ν_e). Atomic nuclei are aggregates of neutrons and protons (which, in turn, are bound states of u and d quarks), and are surrounded by electrons to form atoms; then fundamental particles from the first family are the constituents of all ordinary matter. Besides protons and neutrons, quarks from any family (except the top) form bound states that are collectively called *hadrons*.

The second generation is composed of the *charm* (c) and *strange* (s) quarks, the muon (μ) and the muon neutrino (ν_μ); the third generation consists of the *top* (t) and *bottom* (b) quarks, and the tau (τ) and tau neutrino (ν_τ). Fermions from the second and third generations are unstable and heavier than those of the first family, and they have a short lifetime before decaying into ordinary matter. Due to the similarity in quantum numbers among the families, u, c, and t quarks (same electric charge) are named up-type quarks and analogously, d, s, and b, down-type quarks. The electron, muon and tau, all with electrical charge equal to -1, are known as charged leptons, as opposed to the zero-charged neutrinos.

Gauge bosons mediate interactions among fermions and in some cases among themselves, as it will become clear later in this chapter. Respectively, the strong, electromagnetic, and weak forces are mediated by gluons, photons, and the weak bosons $W^\pm$ and Z; all of which have spin 1 and are known as vector bosons. In contrast, the scalar (spin-0) Higgs boson is responsible for the mass acquisition of other particles in the SM.

[1] Here and in the rest of this thesis we use the convention $\hbar$(reduced Planck constant)$=$ c(speed of light)=1, unless we explicitly state otherwise.

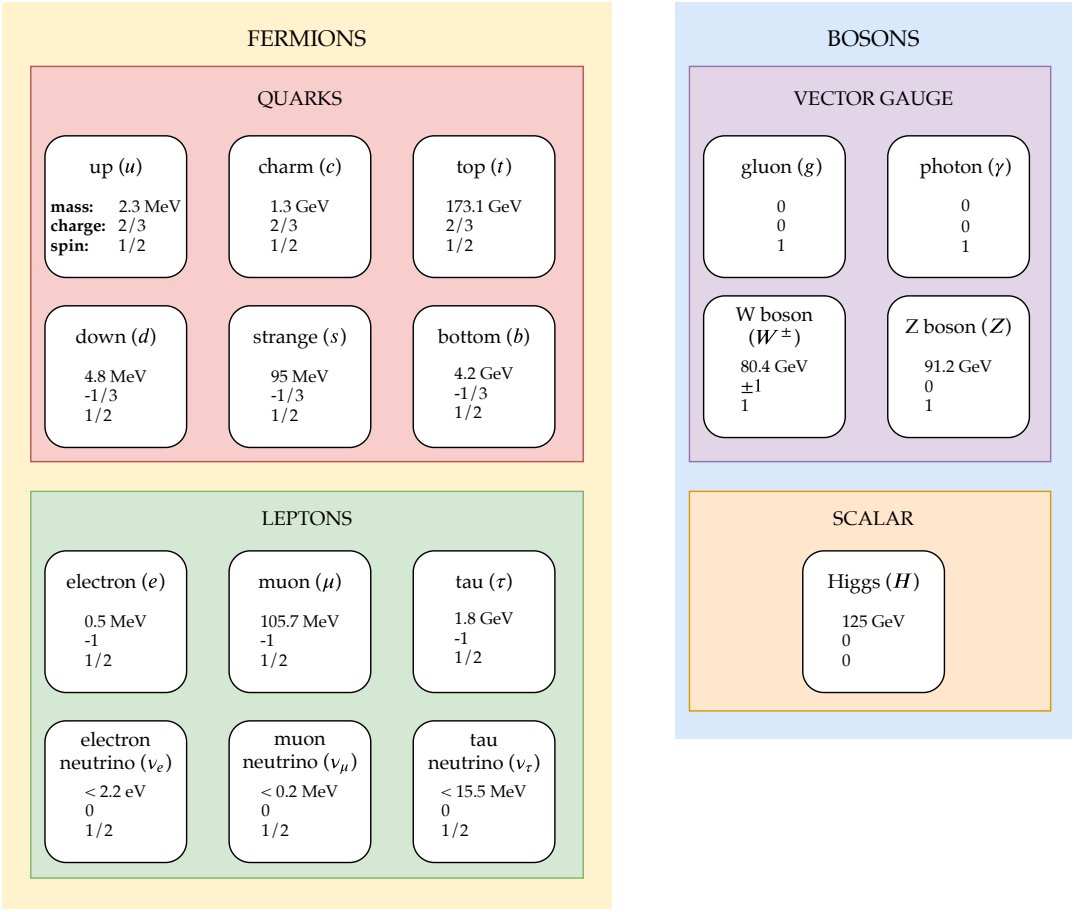


FIGURE 2.1: Particles of the Standard Model and a few of their properties. Electrical charges are in units of the electron charge magnitude. Values taken from ref. [5].

All particles we have mentioned have another particle associated, with the same mass but opposite physical charges; those are known as antiparticles.

2.2 Theoretical basis

The SM is a theory that is based on local gauge symmetries. The symmetry group of the SM is $\mathcal{G} = SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$; the strong interaction is governed by the $SU(3)_C$ symmetry group and the electromagnetic and weak (together, *electroweak*) interactions by $SU(2)_L \otimes U(1)_Y$. The subscripts C, L, Y, denoting *color charge*, *left-handedness*, and *hypercharge*, are associated with the structure of the groups that will be explained in subsequent sections.

Interactions in the SM are governed by the transformations of the fields under those symmetry groups. Table 2.1 contains a summary of the fermions transformations under the SM gauge groups. Left-handed fermions (subscript L in the table) transform as doublets (denoted 2) under $SU(2)_L$ whereas the right-handed (R) ones transform as singlets (1). There are no right-handed neutrinos in the SM. All particles in the SM that do not interact via the strong force are color singlets (1) under $SU(3)_C$. Quarks are color triplets (3) in under $SU(3)_C$ and gluons transform as color octets. In the table, $\alpha = 1, 2, 3$ is a color index for quarks, and $i = 1, 2, 3$ is used for

identifying families. The hypercharge Y is associated with the transformation under the $U(1)_Y$ symmetry group.

Notation	Family			Group transformation		
	1st	2nd	3rd	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$
Q^α_{iL}	$\begin{pmatrix} u^\alpha_L \\ d^\alpha_L \end{pmatrix}$	$\begin{pmatrix} c^\alpha_L \\ s^\alpha_L \end{pmatrix}$	$\begin{pmatrix} t^\alpha_L \\ b^\alpha_L \end{pmatrix}$	3	2	1/3
u^α_{iR}	u^α_R	c^α_R	t^α_R	3	1	4/3
d^α_{iR}	d^α_R	s^α_R	b^α_R	3	1	$-2/3$
ψ_{iL}	$\begin{pmatrix} \nu_{eL} \\ \bar{e}_L \end{pmatrix}$	$\begin{pmatrix} \nu_{\mu L} \\ \bar{\mu}_L \end{pmatrix}$	$\begin{pmatrix} \nu_{\tau L} \\ \bar{\tau}_L \end{pmatrix}$	1	2	-1
$\bar{e}_{iR}$	$\bar{e}_R$	$\bar{\mu}_R$	$\bar{\tau}_R$	1	1	-2

TABLE 2.1: Transformation of SM fermions under gauge groups.

The SM also includes a scalar field, known as the Higgs field, that transforms as a doublet under $SU(2)_L$. This field is responsible for breaking the electroweak $SU(2)_L \otimes U(1)_Y$ symmetry, allowing for a mass acquisition mechanism of all fermions except the neutrinos, and the weak bosons Z, $W^\pm$.

2.2.1 Electromagnetism

The quantum field theory that describes the electromagnetic interactions in the SM is known as Quantum Electrodynamics (QED) [6, 7]. The Lagrangian (density) of a free fermion with mass m can be written in the form

$$\mathcal{L}_{\text{free}} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi\,, \tag{2.1}$$

where $\psi = \psi(x)$ represents the fermionic field for space-time four-coordinates x, and γ^μ are the Dirac gamma matrices[2].

Electromagnetism is governed by a $U(1)_Q$ symmetry, where transformations are local phases $\theta = \theta(x)$:

$$\psi \to \exp(i\alpha\theta)\psi\,, \tag{2.2}$$

where α is known as the coupling strength of the interaction. In order to preserve the invariance of the Lagrangian, one turns to defining the so-called covariant derivative

$$D_\mu = \partial_\mu - ieA_\mu\,, \tag{2.3}$$

where $-e$ is the electron charge, and A_μ is the associated gauge field that transforms as

$$A_\mu \to A_\mu + \frac{1}{e}\partial_\mu\theta\,, \tag{2.4}$$

[2]Defined by the anticommutation relationship $\{\gamma^\mu, \gamma^\nu\} = 2\eta^{\mu\nu}$, with the Minkowski metric $\eta^{\mu\nu} = \text{diag}(+,-,-,-)$.

that is identified with the electromagnetic four-potential. The antisymmetric electromagnetic field strength tensor is:

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \,. \tag{2.5}$$

The propagation of such field is taken into account by adding an invariant term proportional to:

$$F_{\mu\nu}F^{\mu\nu} \,. \tag{2.6}$$

The total QED Lagrangian then reads:

$$\mathcal{L}_{\text{QED}} = \bar{\psi}(i\gamma^\mu D_\mu - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \tag{2.7}$$

$$= \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi - e\bar{\psi}\gamma^\mu A_\mu \psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \,. \tag{2.8}$$

Notice that the first term is the free Lagrangian (eq. (2.1)), while the second term is the interaction of the fermion with the gauge field A_μ, whose associated field is the photon; the last term dictates the kinematics of the gauge field propagation. The strength of the electromagnetic interaction, in the second term, is related to the coupling by $\alpha = \frac{e^2}{4\pi}$ and it is one of the fundamental parameters of QED. This parameter is not a constant and its value depends on the energy scale in which the studied interactions are taking place: at low energies (e.g. those of atomic reactions) its value is approximately $1/137$, but at energies of the order of the weak bosons masses (~ 100 GeV) it is rather around $1/127$. The variation of the value of this parameter (as well as other observables in quantum field theories, in general) is affected by the high-order terms in a perturbative expansion, that need to be taken into account when changing the energy scale.

2.2.2 Electro-weak interactions

Weak interactions are responsible for a number of observed phenomena in nuclear processes, notably the beta decay. As we have seen in table 2.1, *left-handed* fermions are represented in doublets, while *right-handed* ones appear as singlets. Weak interactions distinguish between left- and right-handed fermions, that in the theory are described by the chiral operators:

$$\psi_L = P_L\psi = \frac{1}{2}(1-\gamma^5)\psi \,, \tag{2.9}$$

$$\psi_R = P_R\psi = \frac{1}{2}(1+\gamma^5)\psi \,, \tag{2.10}$$

where the product of gamma matrices $\gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3$, and the $P_{R,L}$ are projection operators, i.e. satisfy $P^2 = P$. It has been observed experimentally that parity is maximally violated[8][3].

In the SM, the electromagnetic and weak interactions are described together in a $SU(2)_L \otimes U(1)$ gauge theory [1, 2, 9]. $SU(2)_L$ preserves the invariance under local phase transformations $\boldsymbol{\theta} = \boldsymbol{\theta}(x)$ of the kind

$$\psi \to \exp(ig\boldsymbol{\theta}\cdot\boldsymbol{T}_L)\psi \tag{2.11}$$

[3]There exist three discrete symmetries in the SM, charge conjugation ($\mathcal{C}$), parity ($\mathcal{P}$) or mirroring, and time reversal ($\mathcal{T}$). Quantum Field Theories require that altogether $\mathcal{CPT}$ is conserved.

where g is the $SU(2)_L$ coupling constant, and $\boldsymbol{T}_L$ are the three generators of the $SU(2)_L$ gauge group, i.e. $T_{kL} = \sigma_k/2$ with Pauli spin matrices being σ_k[4].

In the $SU(2)_L \otimes U(1)_L$ symmetry group, four associated gauge bosons appear: three from $SU(2)_L$ and one for $U(1)_Y$, namely W^a (a =1,2,3), with electric charges[5] $+1$, -1, and 0, and B, with charge 0. The Lagrangian for electro-weak interactions then contains all possible terms for the fermions plus the kinematics of the gauge fields:

$$\begin{aligned}\mathcal{L}_{\rm EW} = {} & \bar{\psi}_{iL}(i\gamma^\mu D^L_\mu)\psi_{iL} + \bar{Q}_{iL}(i\gamma^\mu D^L_\mu)Q_{iL} && \text{(left-handed fermions)} \\ & + \sum_{f=u,\,d,\,e} \bar{f}_{iR}(i\gamma^\mu D^R_\mu) f_{iR} && \text{(right-handed fermions)} \\ & + \frac{1}{4}W^a_{\mu\nu}W^{\mu\nu\,a} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} && \text{(gauge field propagation).}\end{aligned} \tag{2.12}$$

In this expression, analogously to the pure electromagnetic case, we have introduced covariant derivatives

$$D^L_\mu = \partial_\mu + i\frac{g}{2}T_{kL}W^k_\mu + i\frac{g'}{2}YB_\mu\,, \tag{2.13}$$

$$D^R_\mu = \partial_\mu + i\frac{g'}{2}YB_\mu\,, \tag{2.14}$$

where Y is the generator of $U(1)_Y$ (the hypercharge), and g' the coupling constant. Field strength tensors are defined in terms of the gauge fields:

$$B_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu \tag{2.15}$$

$$W^a_{\mu\nu} = \partial_\mu W^a_\nu - \partial_\nu W^a_\mu - g'\epsilon_{abc}W^b_\mu W^c_\nu\,, \tag{2.16}$$

where ϵ_{abc} is the antisymmetric symbol.

After the spontaneous breaking of the electroweak symmetry, that we will discuss below, one obtains a relationship between the weak fields postulated above (W^a, B), and the physical ones associated with the SM ($W^\pm$, Z, A):

$$W^\pm = \frac{1}{\sqrt{2}}(W^1 \mp iW^2)\,, \tag{2.17}$$

$$\begin{pmatrix} A \\ Z \end{pmatrix} = \begin{pmatrix} \cos\theta_W & \sin\theta_W \\ -\sin\theta_W & \cos\theta_W \end{pmatrix}\begin{pmatrix} B \\ W^3 \end{pmatrix}\,, \tag{2.18}$$

where the weak mixing angle θ_W is defined as

$$\tan\theta_W = \frac{g'}{g}\,. \tag{2.19}$$

2.2.3 The Higgs mechanism

Bosons postulated to mediate weak interactions, as they appear above, do not include a mass term in the Lagrangian. As we experimentally know that those bosons ($W^\pm$, Z) are indeed massive, a mechanism is needed to include such mass terms. This is done via the so-called *Brout-Englert-Higgs (BEH) mechanism* or simply the

[4] They comply with the commutation relation $[T_{iL}, T_{jL}] = i\epsilon_{ijk}T_{kL}$.

[5] In units of the electron charge magnitude.

Higgs mechanism, that describes the spontaneous breaking of the electroweak symmetry.

This symmetry is broken following the pattern

$$SU(2)_L \otimes U(1)_Y \rightarrow U(1)_Q \,, \tag{2.20}$$

from which it is possible to find a relationship between the electromagnetic charge (Q) and the weak hypercharge (Y):

$$Q = T_3 + \frac{1}{2}Y \,, \tag{2.21}$$

with T_3 the eigenvalue of the diagonal generator of $SU(2)_L$.

The terms introduced in the total Lagrangian to break the symmetry describe an $SU(2)$ doublet of complex scalar fields,

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} = \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix} \,, \tag{2.22}$$

in the following way:

$$\mathcal{L}_{\text{Higgs}} = (D_\mu \phi)^\dagger (D^\mu \phi) - V(\phi) \,. \tag{2.23}$$

The covariant derivative of this equation corresponds to 2.13, and $V(\phi)$ corresponds to the Higgs potential, given by:

$$V(\phi) = \mu^2 \phi^\dagger \phi + \lambda (\phi^\dagger \phi)^2 \,, \tag{2.24}$$

which is invariant under the local $SU(2)_L$ transformation. The parameter λ is required to be positive, so that the potential is bounded from below; for $\mu^2 > 0$ the minimum of the potential is located at $\phi = 0$ but, since there is no reason for this requirement, it is possible to have a non-zero vacuum when $\mu^2 < 0$. Without loss of generality, one can choose among all possible minima

$$\phi_{\text{min}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix} \,, \tag{2.25}$$

where the vacuum expectation value

$$v = \sqrt{\frac{-\mu^2}{\lambda}} \tag{2.26}$$

is a constant real value. This leaves three degrees of freedom which are "gauged away." By choosing a direction for the minimum, with only one of the doublet components getting a vacuum expectation value, the $SU(2)$ symmetry is said to be broken.

The spectrum for this potential can be obtained by making a perturbative expansion around the minimum, i.e.

$$\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + h \end{pmatrix} \,, \tag{2.27}$$

with $h = h(x)$. If we substitute this value in the first term of the Lagrangian in eq. (2.23) we get a term for the gauge fields

$$\frac{1}{8}v^2g'^2((W_\mu^1)^2 + (W_\mu^2)^2) + \frac{1}{8}(gB_\mu - g'W_\mu^3)^2 . \tag{2.28}$$

This expression corresponds to mass terms for the $W^\pm$ and Z bosons, where

$$m_W = \frac{1}{2}vg' , \qquad \text{and} \qquad m_Z = \frac{1}{2}v\sqrt{g^2 + g'^2} , \tag{2.29}$$

leaving the photon massless $m_A = 0$. The Higgs boson mass, is given by $m_h = \sqrt{-2\mu^2}$. Finally, a relationship between massive weak bosons can be obtained

$$m_Z = \frac{m_W}{\cos\theta_W} . \tag{2.30}$$

It is possible to write terms that involve interactions between the Higgs doublet and the SM fermions which are invariant under $SU(2)$. These are known as *Yukawa* interactions[6]:

$$\mathcal{L}_{\text{Yukawa}} = \sum_{i,j=1,2,3} y_{ij}^d \bar{Q}_{iL}\phi d_{jR} + y_{ij}^u \bar{Q}_{iL}\tilde{\phi} u_{jR} - y_{ij}^e \bar{\psi}_{iL}\phi e_{jR} + \text{h.c.} , \tag{2.31}$$

where we introduced

$$\tilde{\phi} = -i\sigma_2\phi^* = \begin{pmatrix} -\phi^{0*} \\ \phi^- \end{pmatrix} , \tag{2.32}$$

and the Yukawa matrix couplings y for each of the quarks and leptons. Once the Higgs field acquires the vacuum expectation value from eq. (2.31), we obtain the mass terms for quarks and charged leptons:

$$\mathcal{L}_{\text{Yukawa}} = \sum_{f=u,d,e} m_{ij}^f \bar{f}_{iL} f_{jR} + \text{h.c.} , \tag{2.33}$$

masses are related to the Yukawa couplings by

$$m_{ij}^f = \frac{v y_{ij}^f}{\sqrt{2}} ,$$

for $f = u, d, e$. In a basis where the Yukawa matrices are diagonal (known as the mass basis), m_{ii}^f correspond to the physical mass of the fermion f.

The fact that the weak interaction (flavor) basis and the mass basis are not the same leads to the so-called quark mixing. (Under the assumption of massless neutrinos, leptons do not mix as it is always possible to perform a unitary transformation where the interaction between leptons and $W^\pm$ remains unaltered.) This means that mass eigenstates can be written as a mixture of flavor eigenstates via a unitary matrix known as the Cabibbo-Kobayashi-Maskawa (CKM) matrix [10, 11], whose

[6] Here we assume neutrinos as massless but, in reality, they have a small but non-vanishing mass. Neutrino masses and their implications are a major research interest in particle physics, and is beyond the scope of this Chapter.

measured entries are [5]:

$$V_{\text{CKM}} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} = \begin{pmatrix} 0.97446 \pm 0.00010 & 0.22452 \pm 0.00044 & 0.00365 \pm 0.00012 \\ 0.22438 \pm 0.00044 & 0.97359^{+0.00010}_{-0.00011} & 0.04214 \pm 0.00076 \\ 0.00896^{+0.00024}_{-0.00023} & 0.04133 \pm 0.00074 & 0.999105 \pm 0.000032 \end{pmatrix} . \tag{2.34}$$

2.2.4 Strong interactions

Quantum Chromodynamics (QCD) is the theory of the strong force, that describes interactions among quarks and gluons. These particles carry what is known as color charge, and all observed hadrons (quark bound states) in nature are color singlets (or *colorless*).

QCD is a gauge theory described by the $SU(3)_C$ group of local symmetries, where quarks are represented as color triplets and transform ($\theta = \theta(x)$) via

$$\psi \to \exp\left(\frac{i}{2}\lambda^\alpha \theta^\alpha\right)\psi\,, \tag{2.35}$$

where λ^α are the eight generators of the $SU(3)_C$ group in the fundamental representation, known as the Gell-Mann matrices, and $\alpha = 1, \ldots, 8$ the color index. These generators satisfy $[\lambda^\alpha, \lambda^\beta] = if^{\alpha\beta\gamma}\lambda^\gamma$, with $f^{\alpha\beta\gamma}$ the structure constants of the group and $\alpha, \beta, \gamma = 1, \ldots, 8$. The field strength associated to this symmetry is denoted $G^\alpha_{\mu\nu}$, which represents the gluon strength tensor. We can then write, analogously to other symmetries in the SM:

$$\mathcal{L}_{\text{QCD}} = \sum_{\text{quark flavors}} i\bar{q}\gamma^\mu D_\mu q - \frac{1}{4}G^\alpha_{\mu\nu}G^{\alpha\mu\nu}\,, \tag{2.36}$$

with the covariant derivative

$$D_\mu = \partial_\mu + ig_s\lambda^\alpha G^\alpha_\mu\,, \tag{2.37}$$

where g_s is the coupling of the strong force, and the relation between the field strength and the gluon field G^α_μ, a color octet, is

$$G^\alpha_{\mu\nu} = \partial_\mu G^\alpha_\nu - \partial_\nu G^\alpha_\mu + g_s f^{\alpha\beta\gamma} G^\beta_\mu G^\gamma_\nu\,. \tag{2.38}$$

Putting all these elements together in the lagrangian in eq. (2.35) leads to interaction terms between quarks and gluons, and gluon self-interactions from the last term in eq. (2.38).

There are two peculiar features of strong interactions, known as confinement and asymptotic freedom. No free quarks have even been observed, as they are color triplets; instead, quarks (except the top) are confined into hadrons that combine color charge such that the final bound state is colorless, the two most common examples of bound states being mesons (two quarks) and baryons (three quarks). At distances above 10^{-15}m, the strong force by gluon exchange is believed to be constant, and thus the energy stored, e.g. between two quarks grows linearly with the distance among them. Once there is enough energy, a quark-antiquark pair is created from the gluon field, leading to new colorless bound states and screening the original interaction.

The magnitude of the strong coupling $\alpha_s = g_s^2/(4\pi)$ depends on the typical energy (or momentum transfer) scale of the interaction Q:

$$\alpha_s = \frac{12\pi}{(33 - 2n_f)\ln(Q^2/\Lambda^2)}, \tag{2.39}$$

where n_f is the number of quark flavors and Λ is the QCD energy scale (~ 0.3 GeV), where the coupling as a function of Q^2 diverges and perturbative calculations are not possible. As the energy scale grows, the strong coupling tends to zero, a limit in which quarks appear free and they don't form bound states; this phenomenon is the asymptotic freedom.

2.3 Limitations of the Standard Model and potential extensions

The SM is a highly successful theory that has endured decades of experimental confirmation. Its predictions included that of the existence of a third lepton family, and the Higgs boson, that were eventually observed [12–15]. Furthermore, there are many measurable quantities that have been experimentally tested to different orders of accuracy, e.g. the electron anomalous magnetic dipole moment [16] or in the electroweak sector as reviewed in Chapter 10 of [5], thus setting constraints in the existence of physics beyond the SM. However, there is a number of limitations to the SM that have lead to believe that it is not a complete theory of fundamental particles and interactions.

2.3.1 Limitation of the Standard Model

Below, we list several of the most important of the SM limitations:

- **Matter-antimatter asymmetry** The Universe is vastly dominated by the presence of matter with respect to antimatter, but the mechanisms present in the SM to create a matter-antimatter imbalance cannot explain the great extent of this asymmetry. It has been suggested that such mechanism, referred to as *baryogenesis*, are related to CP violation and could lead to the current asymmetry from an initial state that contained matter and antimatter in equal parts. A review of the observational evidence and theoretical frameworks can be consulted in [17].

- **Lack of dark matter candidate** From astronomical observations, such as that of the rotation of galaxies, lensing effects, study of the Cosmic Microwave Background, among other, we have known for decades that most ($\sim$85%) of the matter in the Universe interacts only gravitationally. This kind of matter, known as Dark Matter, has not been directly observed and there are no candidates in the SM that satisfy the observations. Theoretical frameworks and experimental statuses are given in [18, 19]; an accessible recent review of this problem is addressed in [20].

- **Neutrino masses** Neutrinos in the SM are massless, but atmospheric, solar, and reactor neutrino experiments have determined that neutrino masses are non-zero, although orders of magnitude smaller than those of charged leptons [21]. There are several theoretical complications for including neutrino mass

terms in the SM, and mechanisms beyond the SM have been proposed for such purpose [22].

- **Gravity not included** This force is, by many orders of magnitude, weaker than the other three fundamental interactions that are described in the SM. The existence of a massless spin-2 gauge boson, the graviton, has been hypothesized but its detection is far from current experimental capabilities. Many efforts in the theoretical community have been devoted to reconcile the current theory of gravity, i.e. General Relativity (a classical theory of the dynamics of curved space-time), and the (quantum-mechanical) Standard Model.

- **Hierarchy and fine tuning** The value of the Higgs boson mass (m_H) gets quantum radiative corrections from the virtual effects of every particle that couples to the Higgs boson. At one loop, taking a generic fermion-Higgs coupling $-\lambda_f \phi \bar{f} f$, this fact can be expressed as

$$\Delta m_H^2 = -\frac{|\lambda_f|}{8\pi^2}\Lambda^2 \,, \tag{2.40}$$

  where Λ is the (ultraviolet) momentum cutoff imposed to regulate the integral coming from the loop. This is in itself not a problem of the Standard Model. However, we know that at the Planck scale ($\Lambda_P \sim 10^{19}$ GeV), some new framework is required to describe the quantum gravitational effects that become important; if the SM is able to describe Nature up to high energy scales (still below Λ_P), the corrections in eq. (2.40) become large and it is considered "difficult" to keep the Higgs mass at the electroweak scale of ~ 100 GeV. This aspect of the SM has been referred to as the hierarchy problem and, if the SM is true at scales much higher then the electroweak scale, it would require that the radiative corrections cancel out with great precision to keep the Higgs boson mass value; this is known as fine-tuning.

These limitations have inspired a plethora of extensions to the SM, that aim to solve the issues presented in the list above. Those extensions provide a framework in which the SM is a low-energy effective field theory.

2.3.2 An extension of the Standard Model: RPV-MSSM

Many extensions of the SM have been proposed to address one or several of its limitations. Extensions include, for example, the introduction of new symmetries: that between fermions and bosons known as Supersymmetry [23], or between the left and right electroweak sector [24–26]; the extension of the Higgs sector with an extra $SU(2)$ doublet [27]; the postulation of extra spatial dimensions (reviewed in ch. 117 of [5]); among many other. In this section we discuss a few aspects of R-Parity Violating Minimal Supersymmetric Standard Model (RPV-MSSM); this will be of interest in the studies presented in Chapter 6.

Supersymmetry

Supersymmetry (often referred to as SUSY) is a hypothesized space-time symmetry that relates bosons to fermions. SUSY has become popular because it provides several explanations to the some limitations of the SM, and phenomenology that could be potentially verified at the TeV scale[7]. Notably, under some scenarios, the

[7] Although no evidence for SUSY has been found at LHC searches to date.

Spin 1	Spin 1/2	Spin 0
gluons g photon γ	gluinos $\tilde{g}$ photino $\tilde{\gamma}$	
$W^{\pm}$ Z	winos $\widetilde{W}^{\pm}_{1,2}$ zinos $\widetilde{Z}_{1,2}$ higgsino $\tilde{h}^0$	$H^{\pm}$ H $h,\ A$ } Higgs bosons
	leptons l quarks q	sleptons $\tilde{l}$ squarks $\tilde{q}$

TABLE 2.2: Minimal particle content of the Supersymmetric Standard Model. Supersymmetric partners are denoted with a tilde. Table taken from [28].

lightest supersymmetric particle can be a Dark Matter candidate; furthermore, SUSY resolves several aspects of the hierarchy and fine tuning problems.

The introduction of supersymmetry implies that for each of the particles in the Standard Model there is a partner (known as superpartner) with a spin differing by 1/2. If SUSY is an actual symmetry of nature, we know that it needs to be spontaneously broken at some high-energy scale, as no superpartners have been observed at the mass of its corresponding SM particle. The contributions of the supersymmetric partners to eq. (2.40) balance out at orders presumably not much higher that a few TeVs to keep the Higgs boson mass stable without fine tuning.

The minimal supersymmetric Standard Model (MSSM) is the supersymmetric extension to the SM that introduces the fewest new interactions and states. The particle content of the MSSM is summarized in table 2.2. Fermions superpartners are denoted by prepending an "s" to the SM fermion name and bosons superpartners by appending "ino"; e.g. the partner of the top quark is a boson called *stop*.

In the MSSM, left handed fermions are chiral super multiplets denoted[8] L, Q, e^c, u^c. Right handed fields e_R, u_R and d_R are also represented in terms of left-handed super multiplets via charge conjugation (e^c, u^c and d^c). Gauge bosons are promoted to gauge vector superfields with their fermionic counterpart named gauginos. The Higgs sector usually is extended to two $SU(2)$ doublets:

$$\mathrm{H_u} = \begin{pmatrix} \mathrm{H_u^0} \\ \mathrm{H_u^-} \end{pmatrix} \quad \mathrm{H_d} = \begin{pmatrix} \mathrm{H_d^+} \\ \mathrm{H_d^0} \end{pmatrix} . \tag{2.41}$$

The MSSM superpotential is then written:

$$W_{MSSM} = \mu H_u H_d + y^e_{ij} H_d L_i e^c_j + y^d_{ij} H_d Q_i d^c_j - y^u_{ij} H_u Q_i u^c_j \,, \tag{2.42}$$

which contains a mass term for the Higgs doublets (first term) and Yukawa terms. This prescription leads to five states: two charged Higgs bosons ($H^{\pm}$), a pseudoscalar A and two neutral Higgs scalars H and h, the latter of which is identified with the Higgs boson. The superpartners of neutral, non-colored SM gauge bosons (W^3, B,

[8] We skip indices for notation clarity.

before SSB) plus the neutral Higgs states (H_d^0, H_u^0) mix to form four neutral states called neutralinos and denoted $\tilde{\chi}_1^0, \tilde{\chi}_2^0, \tilde{\chi}_3^0, \tilde{\chi}_4^0$. In a similar way, the superpartners of charged gauge bosons and the charged Higgs states mix to form the charginos $\tilde{\chi}_1^\pm, \tilde{\chi}_2^\pm$.

R Parity Violation

An extra discrete Z_2 symmetry, called R-parity is introduced in the MSSM:

$$R = (-1)^{3B+L+2s}, \tag{2.43}$$

where B and L are the baryon and lepton numbers and s is the spin. Imposing R-parity has desirable consequences in the MSSM, such as automatically forbidding proton decay and conserving baryon and lepton numbers in all renormalizable couplings.

More generally, the superpotential of the MSSM with R-parity violating terms can be written as

$$W_{RPV} = \mu_i H_u L_i + \frac{1}{2}\lambda_{ijk} L_i L_j e_k^c + \lambda'_{ijk} L_i Q_j d_k^c + \frac{1}{2}\lambda''_{ijk} u_i^c d_j^c d_k^c, \tag{2.44}$$

where $\lambda_{ijk} = -\lambda_{jik}$ and $\lambda''_{ijk} = -\lambda''_{ikj}$. The couplings λ couplings are tightly constrained, as no flavor or baryon number violation processes or proton decays have been observed. We will see in chapter 6 an example of a search where the simulated benchmark signal was simulated from an R-parity violating process.

Chapter 3

The Large Hadron Collider and the ATLAS detector

3.1 Introduction

Major discoveries in Particle Physics have been achieved in the last decades by studying high energy collisions. In pursuing the exploration of the so-called energy frontier, physicists have built increasingly powerful devices to produce and study particle collisions at the highest energy possible. The energy available in such collisions allows for the creation of heavy particles, whose presence can be subsequently inferred with data recorded by detectors located at collision points. The discovery of the last two fundamental particles in the Standard Model (SM), namely the top quark and the Higgs boson, happened respectively at Fermilab's Tevatron (a proton-antiproton collider) [12, 13] and at CERN's Large Hadron Collider (LHC, a proton-proton collider[1]) [14, 15].

The LHC is the most powerful particle accelerator and collider ever built and it aims to probe the predictions of the SM and shed light on the existence of New Physics (NP). The main experiments of the LHC physics program are four: ATLAS, CMS, LHCb and ALICE. ATLAS and CMS are known as general purpose detectors, since they cover nearly the full solid angle and have been designed to study a wide range of phenomena from the collisions. LHCb is a forward detector specialized in the studying CP violation with hadrons containing b quarks, among other topics. The purpose of A Large Ion Collider Experiment (ALICE) is to study heavy-ion collisions, that produce quarks and gluons at extreme energy densities, in a phase of matter known as quark-gluon plasma that is thought to have existed at the very early universe.

Below we provide a brief introduction to and description of the experimental setup and facilities of interest in this thesis, namely proton-proton collisions at the LHC recorded by ATLAS.

3.2 The Large Hadron Collider

The LHC is a circular collider located across the French-Swiss border. Its main ring has a circumference of about 27 km in a tunnel located approximately 100 m underground. Superconducting magnets are responsible for accelerating charged particles (protons or heavy ions) in opposite directions of the ring. The accelerated particle beams are then focused to intersect each other at different points, where the detectors are placed. The LHC is designed to produce collisions up to center of momentum

[1] Collisions of other nature, e.g. of heavy ions, are also studied at the LHC, which is a subject out of the scope of this thesis.

energy of $\sqrt{s} = 14$ TeV and deliver an instantaneous luminosity higher than 10^{34} $cm^{-2}s^{-1}$.

3.2.1 Proton acceleration

The LHC is served by a chain of smaller accelerators that inject (bunches of[2]) particles in the main ring. The proton acceleration procedure happens as follows, and can be traced with the grey arrows in figure 3.1. Protons are obtained by ionizing hydrogen with an electric field; then, they are injected and accelerated at the LINAC 2, a linear accelerator, to an energy of 50 MeV. A sequence of three further steps of injection and acceleration happen at increasing energies: firstly, reaching 1.4 GeV at the Proton Synchrotron Booster (PSB), then 25 GeV at the Proton Synchrotron (PS), and finally 450 GeV at the Super Proton Synchrotron (SPS).

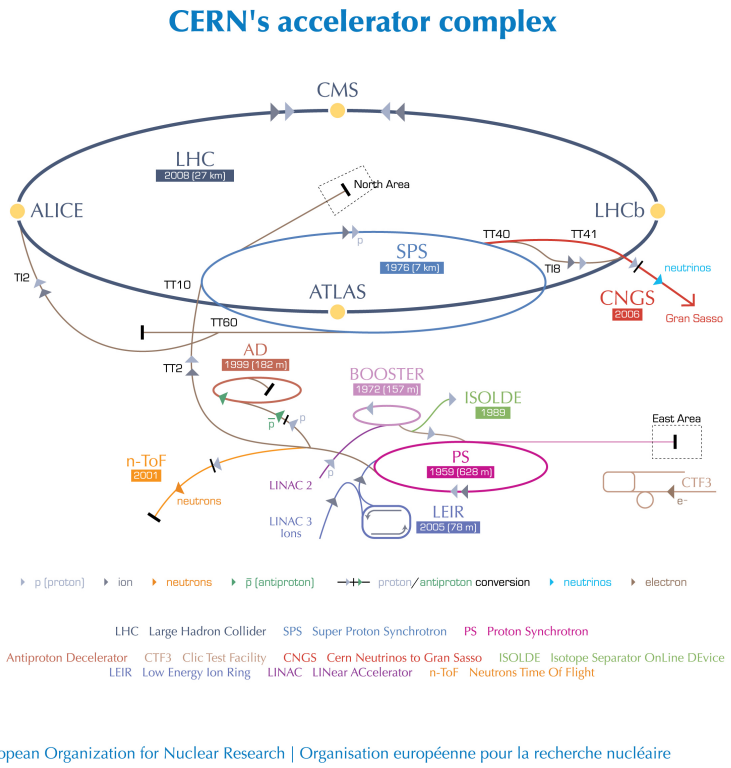


FIGURE 3.1: Accelerator complex at CERN [29]. Experiments are shown with their respective year of start of operations and circumference length.

The final acceleration of proton bunches happens at the main LHC ring, to reach a maximum of 6.5 TeV of energy per beam, that circulate in vacuum tubes. Electric fields are used to accelerate proton beams and magnetic fields for bending them, i.e. to keep them in orbit, and focusing for collisions. Sixteen radiofrequency cavities operating in a superconducting state are used to create the electric field for acceleration, reaching a maximum tension of 2 megavolts. Over 1200 dipole magnets that generate an 8.3 tesla magnetic field are used to bend the beams, and quadrupole magnets are used to keep the beam squeezed (i.e. avoid spread in the plane perpendicular to the beam direction). The magnet system also operates in a superconducting state, which is achieved by a cryogenic system with liquid helium cooled down to 1.9 K.

[2] A proton bunch contains approximately 10^{11} protons.

Finally, a system of quadrupoles are used to focus the beams to intersect; this reduces their spread from 0.2 millimeters down to 16 micrometers across. Figure 3.2 presents a transversal view of the LHC dipole.

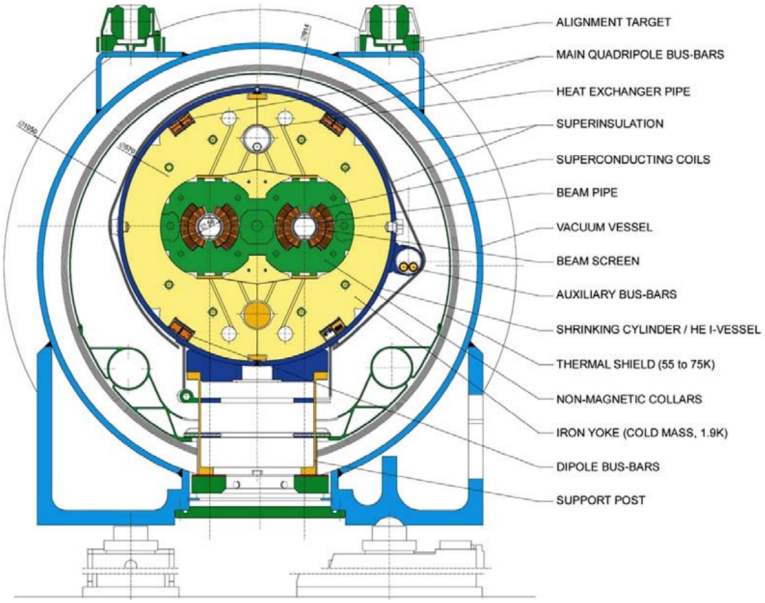


FIGURE 3.2: Schematic transversal view of the LHC dipole, taken from [30].

3.2.2 Proton collisions at the LHC

One defining property of a collider is the instantaneous luminosity ($\mathcal{L}$), which is a quantity specifying the potential number of collisions per second. The number of events for a given process X from a proton-proton (pp) collision is proportional to the process cross section ($\sigma_{pp\to X}$):

$$\frac{dN_{pp\to X}}{dt} = \mathcal{L}\,\sigma_{pp\to X}\,. \tag{3.1}$$

Given that most of the physics of interest at the LHC correspond to rare processes, it is crucial to design colliders with the maximum luminosity possible.

The luminosity can be expressed as:

$$\mathcal{L} = \frac{N_b^2 n_b f_{\text{rev}} \gamma}{4\pi\epsilon_n \beta^*} F\,, \tag{3.2}$$

where N_b is the number of protons per bunch and n_b the number of bunches per beam, f_{rev} the revolution frequency, γ the Lorentz factor, ϵ_n the normalized transverse emittance of the beam and β^* is a function that quantifies the oscillations of protons at the collision point. Finally, the geometric factor F accounts for the crossing angle at the intersection point θ_c:

$$F = \left(1 + \left(\frac{\theta_c \sigma_z}{2\sigma_*}\right)^2\right)^{1/2}\,, \tag{3.3}$$

where σ_z and σ_* are respectively the RMS of the longitudinal and transversal beam spreads.

We present in figure 3.3 plots corresponding to instantaneous luminosity (left) and integrated luminosity over time (right). On the left plot, we can see that during the data taking period of 2017, the instantaneous luminosity exceeded the design value of 10^{34} cm^{-2} s^{-1}. During the so-called Run 2 LHC data-taking period in 2015-2018, the integrated luminosity recorded by ATLAS has reached 147 fb^{-1}, as it is visible in the plot on the right.

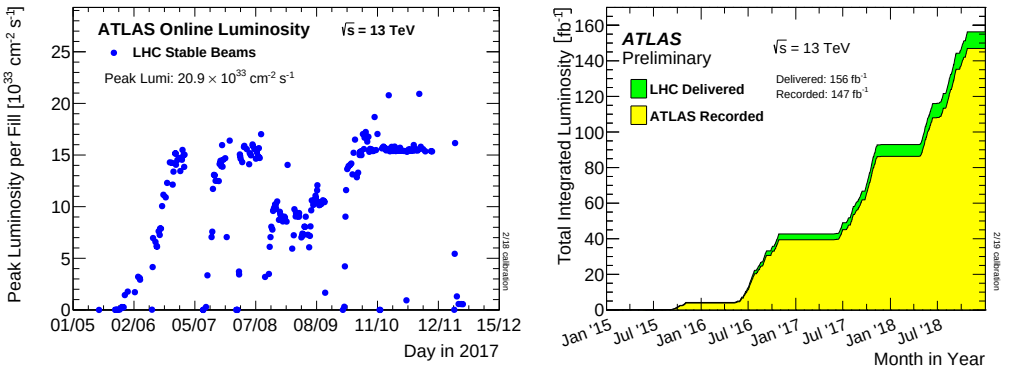


FIGURE 3.3: Left: ATLAS peak luminosity per fill in 2017. Right: LHC delivered (green) and ATLAS recorded (yellow) luminosity between 2015 and 2018. Taken from [31].

The LHC collides bunches of protons every 25 nanoseconds. At each bunch crossing, on the order of 20 interactions from the colliding protons take place on average. One of the experimental challenges of the LHC detectors is to distinguish the outcome of different interactions that "pile-up" in the detectors. There can also be pile-up effects from different bunch crossings due to short time separation among bunches and slow response from the detector.

3.3 A Toroidal LHC ApparatuS (ATLAS)

ATLAS is a general purpose detector at the LHC that has been designed primarily to make measurements of the SM and searching for NP with proton-proton collisions. It is the largest of all LHC detectors, with a cylindrical shape of 44 m long and 25 m in diameter, and weighs 7000 tons.

ATLAS comprises several specialized subsystems for different purposes. In order to identify and measure the properties of the particles created in the collisions, several concentric subdetectors are disposed in layers around the collision point. The subdetectors can be separated in four parts: the inner detector that is responsible for measuring the trajectory of charged particles, the calorimeters, responsible for measuring the energy and direction of particles, and finally a detector that measures properties of the muons that penetrated the previous layers, called the muon spectrometer. A magnet system is responsible for bending the trajectories of charged particles, which is used to measure their momenta. Figure 3.4 presents a scheme of the ATLAS detector, its dimensions and subcomponents. Along with the subdetectors and magnets, ATLAS also has a triggering system that is responsible for selecting only events that contain potentially interesting physics, and a computer system for developing software for the storage, processing and analysis of the data

recorded. We present in the following subsections a brief review of several aspects of the detector.

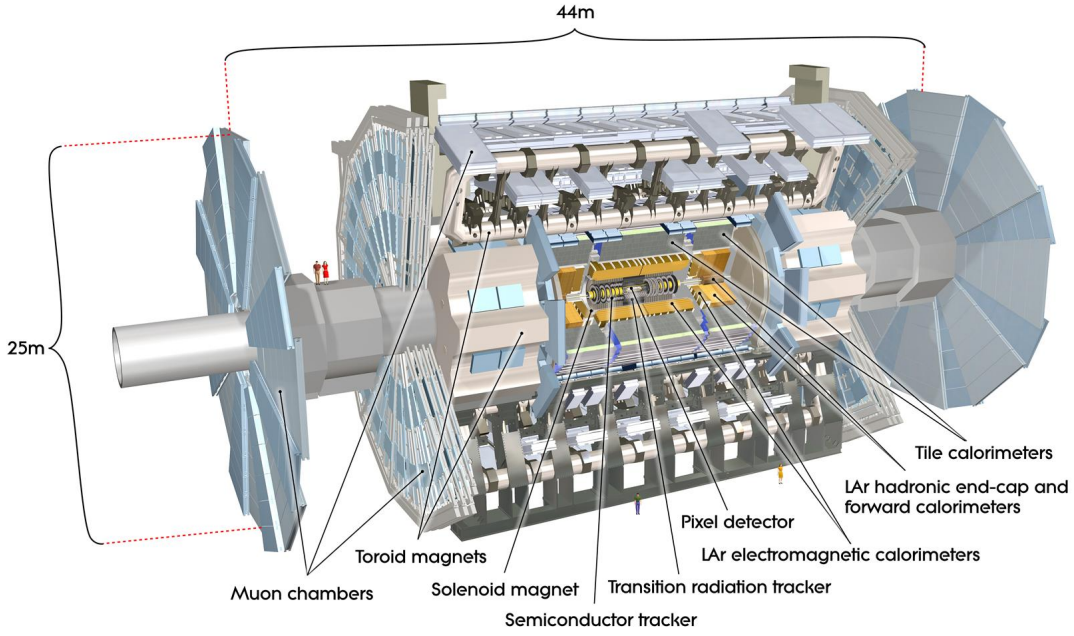


FIGURE 3.4: Schematic view of the ATLAS detector. Taken from [32].

3.3.1 Geometric conventions

A convenient set of conventional spatial coordinates are used throughout the collaboration. The xyz-coordinates are defined as follows: the z-axis is defined to be parallel to the (counter-clockwise) beam direction, the x-axis points towards the center of the LHC ring, from the collision point, and the y-axis points upwards towards the Earth surface. The xy-plane, perpendicular to the beam direction, is referred to as the transverse plane, and the projection of the momentum of a particle in that plane is called the transverse momentum, denoted p_T.

Angular coordinates are also defined: θ is the polar angle, measured from the z-axis, and ϕ is the azimuthal angle, measured from the x-axis. Usually the polar angle is translated to pseudorapidities:

$$\eta = -\ln \tan \frac{\theta}{2}, \tag{3.4}$$

because η differences are Lorentz-invariant under a boost along the z-axis[3]. The distance in the η-ϕ plane is defined as

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}. \tag{3.5}$$

[3] In the ultrarelativistic limit.

3.3.2 Magnet system

ATLAS has two main superconducting magnet systems [33] for bending charged particle trajectories: the central solenoid and the toroid system. The central solenoid is responsible for providing the magnetic field for the inner detector, and is composed of four magnets aligned with the with the beam axis, reaching to a 2 Tesla axial magnetic field. The toroid system is composed of three parts, for providing a magnetic field used for muon spectrometry: the barrel toroid with eight coils for the central region (2.5 Tesla) and the two end-cap toroids with eight coils each for the more forward region (0.35 Tesla). Figure 3.5 shows a model of the magnet systems.

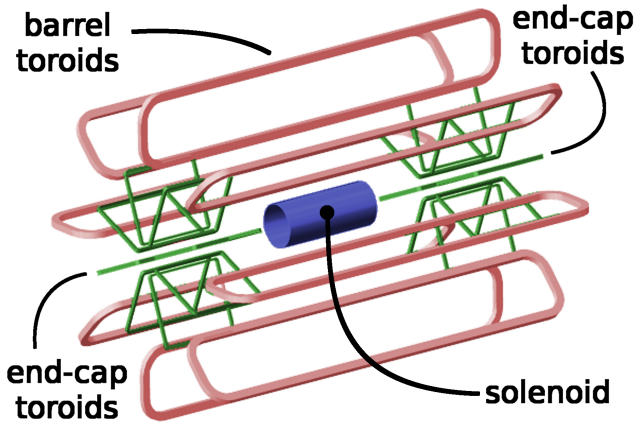


FIGURE 3.5: ATLAS magnet systems. Taken from [34].

3.3.3 Inner detector

The inner detector is the closest to the interaction point, and is responsible for measuring the direction, momentum and charge of charged particles. It achieves a transverse momentum resolution of $\sigma_T/p_T = 0.05\% p_T \oplus 1\%$. This detector plays an important role in the reconstruction of primary and secondary vertices.

The inner detector has three main components, i.e. the Pixel Detector, the Semiconductor Tracker (SCT) and the Transition Radiation Tracker (TRT). It is immersed in the magnetic field produced by the central solenoid. Figure 3.6 presents schemes of the inner detector and a view of a transversal slice.

Pixel Detector

The Pixel Detector is the closest component to the interaction point. Its main purpose is the identification of secondary vertices that are then used to identify particle jets coming from a b quark (known as b-jets). The Pixel Detector has a total of 80 million pixels (channels) made of polarized PN junctions, where the passage of a charged particle leads to the creation of electron-hole pairs that, in the presence of an electric field, can be transformed into a signal sent to the readout. The nominal pixel size is $50 \times 400\ \mu m^2$. This detector covers the full ϕ range and values of pseudorapidity such that $|\eta| < 2.5$.

Besides the three layers shown in figure (3.6) (right), an insertable B-layer [36] was placed closer to the beam pipe (occupying the 31-40 mm R region). This allows for an improvement in the reconstruction of primary and secondary vertices. The

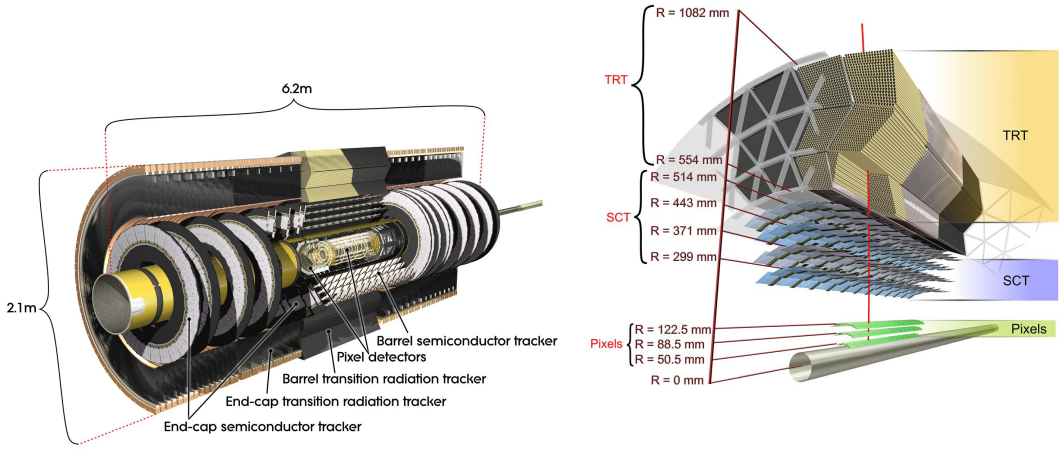


FIGURE 3.6: Left: Schematic view of the inner detector dimensions and subcomponents. Right: Section of a transversal slice of the inner detector subcomponents; the R value indicates distance to the proton beam. Taken from [35].

Pixel Detector has a resolution of 40 μm in the direction parallel to the beam axis (known as longitudinal direction) and 8 μm in the R-ϕ plane [37].

Semiconductor Tracker

The Semiconductor Tracker (SCT) uses silicon microstrip sensors with a technology similar to that of the Pixel Detector. The sensors are distributed over four coaxial cylindrical layers and 18 (plane) disks, perpendicular to the beam axis, at the endcap; it has 6 million channels. The SCT covers the full ϕ region and pseudorapidities $|\eta| < 2.5$. The resolution of the SCT is 17 μm per layer in the R-ϕ plane and 580 μm in the z-axis.

Transition Radiation Tracker

The Transition Radiation Tracker (TRT) is the most external part of the inner detector. Its fundamental element is a straw tube with a diameter of 4 mm, in the center of which a 0.03 mm-diameter gold-plated Tungsten wire is placed; there is an electric field between the wire and the surface of the tube. The tube is filled with a mixture of gases (argon, carbon dioxide and oxygen[4]), that is ionized with the passage of charged particles. The electrons resulting from the ionization are driven to the wire to create an electric signal. The tracks reconstructed by the TRT provide good discrimination between electrons and charged $\pi^{\pm}$ mesons, via the transition radiation produced in materials of different dielectric constant that are interleaved with the straws.

[4]This mixture used the more expensive Xenon instead of Argon in the past.

3.3.4 Calorimeters

The calorimeters are designed to measure the energy that a particle loses as it interacts with the detector material. The aim is to make particles lose as much energy as possible, often stopping them completely (total absorption). Electromagnetic and hadronic calorimeters in ATLAS are sampling calorimeters, i.e. they have parts made of an active material where the energy is deposited from a sequence of interactions that create a shower, and other parts from a different absorber material for measuring the deposited energy in the active material. Figure 3.7 presents a schematic view of the ATLAS calorimeters.

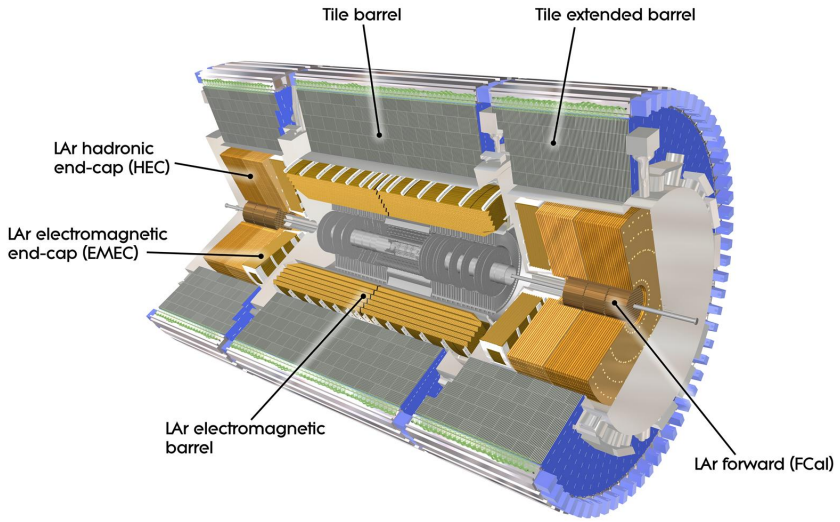


FIGURE 3.7: ATLAS calorimeter system. Taken from [38].

Liquid Argon Calorimeter

ATLAS' electromagnetic calorimeter (ECAL) uses liquid Argon (LAr) as the active medium, and lead plates as absorbers. The main goal of the ECAL is to measure the energy of electrons and photons, and the electromagnetic component of hadronic jets. The ionization induced in the LAr by the passage of charged particles is then collected by readout electrodes that are kept at high voltage (2 kV) with respect to the absorbers.

Full ϕ coverage is ensured by the accordion-like geometry of the layers. Details of an ECAL module are presented in figure 3.8. The electromagnetic barrel (EMB) calorimeter covers the region $|\eta| < 1.475$ (except a small region near $\eta = 0$) and consists of two half-barrels. Each of the two electromagnetic endcap (EMEC) calorimeters consist of two wheels, the inner wheel and the outer wheel, covering respectively ranges $1.375 < |\eta| < 2.5$ and $2.5 < |\eta| < 3.2$. A cryostat system is responsible for maintaining the LAr at a temperature of about 90 K.

The modules of the ECAL are segmented in three main layers with different cell granularities in the η-ϕ plane (figure 3.8, left). Additionally, a preshower (PS) layer covers a region of $|\eta| < 1.8$. A small fraction of the energy is deposited in the first layer and the PS layer, while most of the energy is deposited in the second layer. The third layer is responsible for measuring the last part of the shower of the most

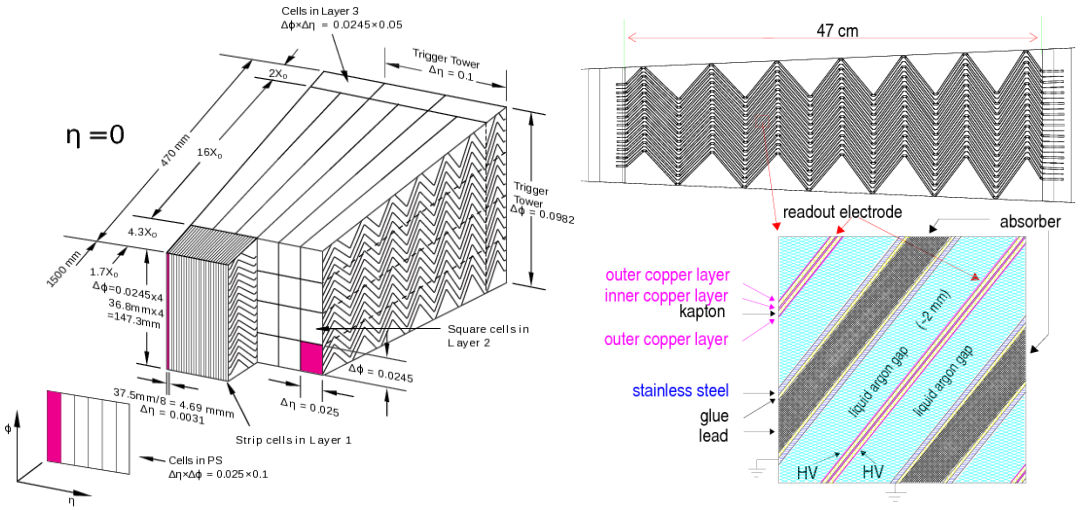


FIGURE 3.8: Left: Sketch of an electromagnetic barrel module in the most central region, showing the respective layers and dimensions [39]. Right: detail of the interleaving of the elements in the ECAL [40].

energetic particles. The energy resolution of the ECAL is $\frac{\sigma_E}{E} = \frac{10\%}{\sqrt{E/GeV}} \oplus 0.7\%$ for the passing electrons and photons.

Tile Calorimeter

The Tile Calorimeter (also TileCal) uses scintillating plastic tiles as the active medium and steel as absorber. It consists of a central barrel covering $|\eta| < 1.0$ and two extended barrels in $0.8 < |\eta| < 1.7$. Each of the 64 barrel modules has a wedge shape covering approximately a range of $\Delta\phi = 0.1$. The segmentation in η for each module varies across the three layers present: $\Delta\eta = 0.1$ for the first two and $\Delta\eta = 0.2$ for the third layer. The TileCal achieves an energy resolution for hadronic jets of $\frac{\sigma_E}{E} = \frac{50\%}{\sqrt{E/GeV}} \oplus 3\%$ in the central barrel and $\frac{\sigma_E}{E} = \frac{100\%}{\sqrt{E/GeV}} \oplus 10\%$ in the extended barrel. Figure 3.9 presents a scheme of a single TileCal module wedge.

The Hadronic Endcap is composed of two independent wheels per endcap, behind those of the EMEC. It uses a copper plates and LAr technology, covering a region of $1.5 < |\eta| < 3.2$.

Forward Calorimeter

This calorimeter covers the most forward region of $3.1 < |\eta| < 4.9$ where the most intense flux of particles is present. It uses also a LAr technology and consists of three parts: one closer to the interaction point that uses copper as absorber and is in charge of the measurement of electromagnetic interactions, and other two parts that use Tungsten as absorber and measure hadronic interactions. The Forward Calorimeter uses concentric rods inside tubes that are parallel to the beam axis.

3.3.5 Muon Spectrometer

The Muon Spectrometer (MS) is the largest and outermost of the ATLAS subdetectors. It is a system designed for detecting the charged particles that escape the calorimeters, mostly muons, and measuring their trajectory, momentum and charge.

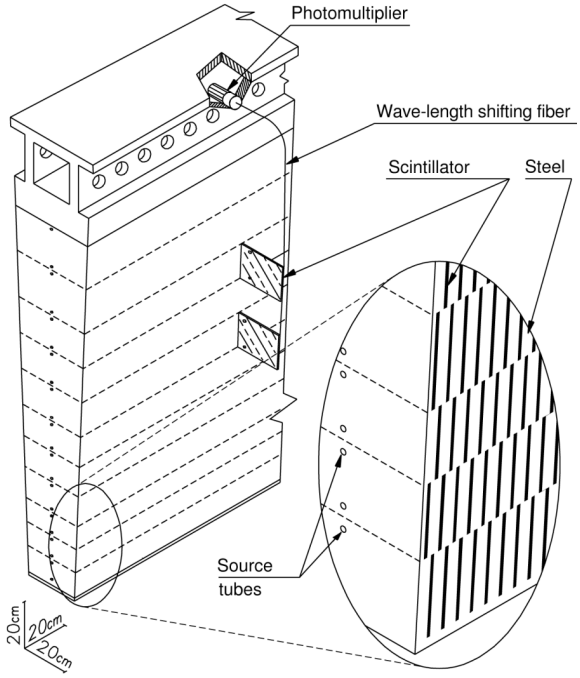


FIGURE 3.9: A sketch of a TileCal wedge. Taken from [41].

The MS covers a region of $|\eta| < 2.7$ and is able to measure particle momenta in the range 3 GeV - 1 TeV, with a p_T resolution of 10% for 1 TeV muons. The toroidal system provides the magnetic field used to bend the charged particles in a plane parallel to the beam axis, that are then measured by the MS.

There are mainly four elements in the MS, that are displayed in figure 3.10:

- The Monitored Drift Tubes (MDTs) are aluminum cylinders with a 30 mm diameter (and length of 0.85-6.5 m) with a tungsten wire in their center, filled with a mixture of gaseous argon and CO_2 (93% and 7% respectively). The ionization of this gas by the muons is collected by the tungsten wire, there is a total of over 354000 tubes. The main purpose of the MDTs is to precisely measure the position of the bent particles.

- The Cathode Strip Chambers (CSCs) have the same purpose as the MDTs but with more precision, and operate in the inner part of the endcap ($2.0 < |\eta| < 2.7$) with 70000 channels. CSCs are multi-wire proportional chambers filled with a gas mixture of 80% Ar and 20% CO_2. Cathode strips are perpendicular to the (anode) wires of the chambers, allowing for the measurement of the charge distribution.

- Resistive Plate Chambers (RPCs) consist of a parallel pair of resistive plates with a 2 mm gap, that are kept with an electric field of about 5 kV/mm. The gap is filled with a gas mixture of 94.7% $C_2H_2F_4$, 5% Iso-C_4H_10 and 0.3% SF_6. RPCs provide triggering information with over 380000 channels; they and are disposed in three cylindrical layers covering the central region, concentric with the beam axis.

- Thin Gap Chambers (TGCs) are used for triggering in the endcap region ($1.05 < |\eta| < 2.4$). TGCs are multi-wire proportional chambers of high granularity that

operate in narrow time windows <25 ns (proton bunch time separation), as it is the case of RPCs. A gas mixture of 55% CO_2 and 45% n-C_5H_{12} is used to fill the TGCs.

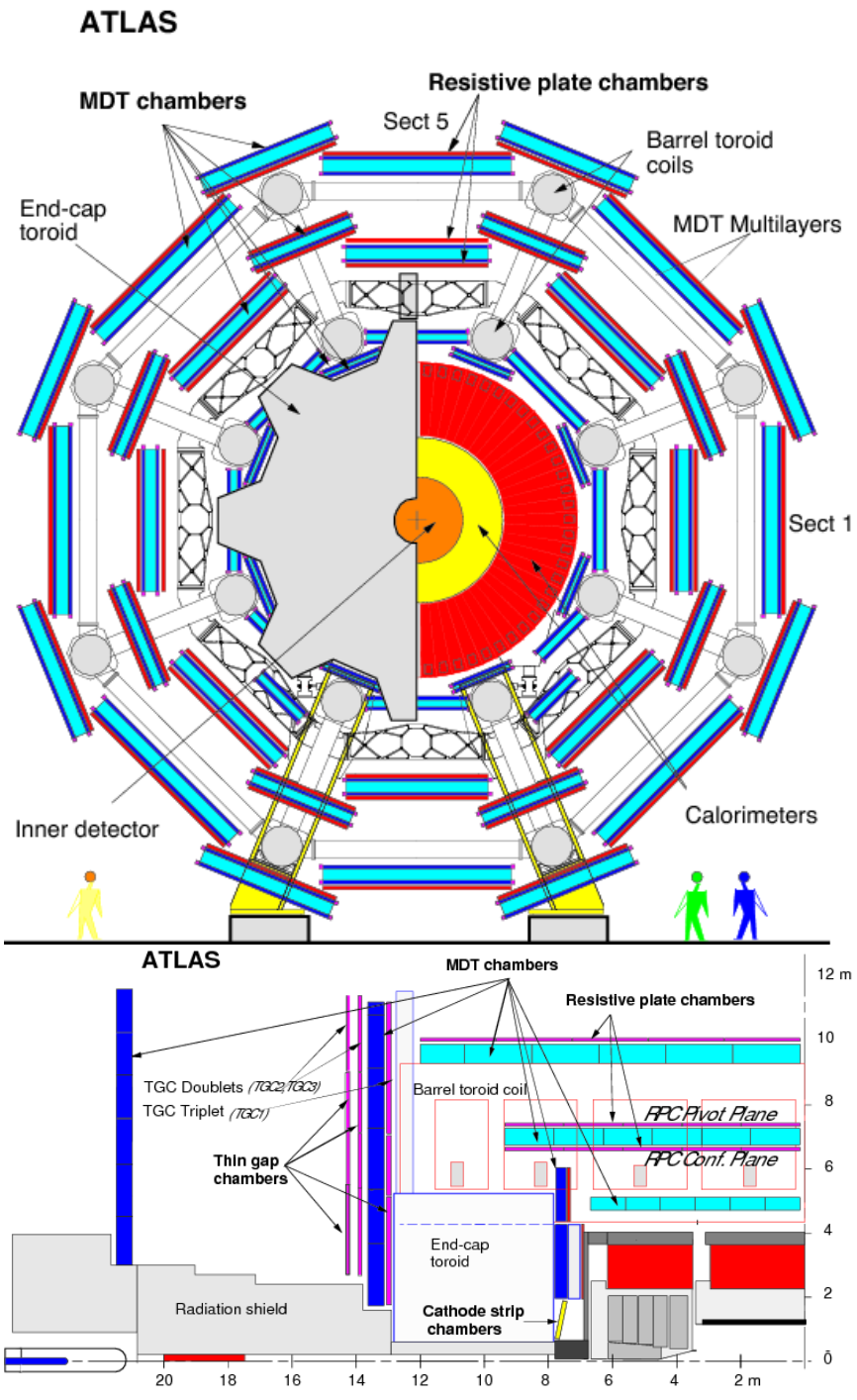


FIGURE 3.10: Schematic view of the muon spectrometer in the x-y (top) and z-y (bottom) projections. Taken from [42].

3.3.6 Trigger System

This system is in charge of selecting proton (bunch) crossings containing "interesting collisions" to be recorded, and discarding the rest[5]. Since the LHC collisions happen at a rate of 40 MHz, the trigger system needs to be able to make the decision in a suitable time window. The ATLAS trigger system has two major parts: the Level-1 trigger, that operates at a hardware level and reduces the rate of events to the order of 100 kHz, and a software-based High Level Trigger (HLT) that reduces the accepted events further down to about 1 kHz.

The hardware-based Level 1 trigger is implemented in dedicated electronics (FPGAs and ASICs) that allow for fast decision making. This trigger uses information from the MS and the calorimeters, and also computes the missing transverse momentum E_T^{miss} and the event-wise sum of jet energy, to produce data to be processed by the Central Trigger Processor (CTP). The Level 1 Calorimeter trigger uses a sliding window algorithm to search for energy maxima in trigger towers (see figure 3.8, left), and compares the value found with a predefined threshold. The Level 1 Muon trigger uses hits in the RPCs (central region) and TGCs (endcaps) to identify muons for different p_T thresholds. Both the calorimeter and muon Level 1 triggers define Regions Of Interest (ROI) in solid angle, where the particles have been identified. The CTP takes the trigger decision based on a combination of predefined criteria (ROIs, predefined thresholds, multiplicity of identified objects, etc.) that are stored in the hardware.

The High-Level Trigger (HLT) uses as inputs the events that passed the Level 1 trigger. However, the HLT is able to process events in parallel and asynchronously, unlike the sequential Level 1 trigger, with algorithms that run in a computer farm. The HLT uses the whole granularity of the detector and the ROIs to reconstruct the different physics objects. The selections imposed in the HLT are designed to minimize the differences with the offline selections imposed in physics analyses.

HLT selections are stored in so-called trigger chains that contain labels that identify the requirements imposed, following the structure:

```
[LEVEL]_[N TYPE][THRESHOLD]_[QUALITY]_[ISOLATION]
```

where `LEVEL` identifies the Level 1 trigger or HLT, `N TYPE` is the multiplicity and type(s) of object(s) used, `THRESHOLD` the p_T threshold used, and `QUALITY` and `ISOLATION` define quality requirements and isolation working points on the objects, respectively.

Finally, the events that pass the HLT selection are stored at the CERN computing center (Tier 0) for reconstruction. The data is also distributed across the Worldwide CERN Computing Grid [43] for further processing and analysis.

3.4 Conclusions

We have reviewed the key aspects of the LHC and the ATLAS detector. With that experimental equipment, the current knowledge of the energy frontier is being explored by performing SM measurements and searching for New Physics. The subcomponents of the ATLAS detectors have been designed for such purpose, allowing for efficient reconstruction and precise measurements of the particles produced in the collisions, covering almost the full solid angle. A major upgrade in ATLAS is planned for the new luminosity conditions at the LHC, the High-Luminosity LHC,

[5] Most events at the LHC are low-pT inelastic collisions. Also, handling (processing and storing) all events would turn to be very expensive and impractical.

to start in 2025. Other experiments proposed and ongoing offer promising research paths, such as precision measurements ar Higgs boson factories (e.g. the International Linear Collider) or increasing the energy by an order of magnitude (Future Circular Collider).

Chapter 4

Monitoring Generic Signatures in ATLAS

In this Chapter, we describe the use of the TAg DAta (TADA) monitoring system [44] for automatically exploring generic signatures in ATLAS. TAG data refers to a condensed file format that contains information from the (simulated or recorded) collision events. The exploration of generic signatures with TADA is inspired from the General Searches for New Physics (e.g. that in ref. [45]), aiming to extend the already broad set of signatures monitored by the system.

4.1 TADA: A fast monitoring system for ATLAS

There are two main purposes for TADA: to serve as an early warning system for New Physics searches, and to provide a tool for physics validation and performance of the data and MC. TADA uses data runs that were processed and updated twice a day during data-taking periods. Both the running of the monitoring system and the writing of the data in TAG format happen in the so-called Tier-0 of the ATLAS computing infrastructure [46] whose purpose, among others, is to promptly process the raw data that is output from the detector's data acquisition system. The final output of the TADA system consists in a webpage with hundreds of channels with thousands of histograms filled, that is available to the members of the ATLAS collaboration in `https://atlas.web.cern.ch/Atlas/fastphys/tagmon/`.

The TADA monitoring software is written in C++ and Python. This solution allows the efficient handling of the more computationally-intensive tasks such as the processing of the TAG files (with C++) as well as a clean interface for other functionalities like the use of metadata, job management, and the generation of the system webpage (with Python). Python dictionaries are used for storing the definitions of channels, i.e. trigger and object requirements, histograms to be booked and filled, etc.

There is a broad number of physics channels that are monitored by TADA. Those use a selection of events and requirements that are inspired in real dedicated analyses that perform e.g. studies searching for New Physics. A set of histograms are filled and other kind of plots and tables are produced to validate the recorded data and check their compatibility with MC simulations. The main categories for the signal regions are *Standard Model Top*, *Higgs*, *Exotics*, and *Susy*. Being a fast monitoring tool, TADA cannot lead to making sensible claims on the existence of New Physics, but the appearance of a feature in data in one of such channels would trigger dedicated inspection. A number of performance and validation control plots are also presented in the webpage.

Despite its great performance as a monitoring tool, there are a few limitations to the TADA monitoring system. Arguably the most relevant is that systematic errors are not included. Also, there are no background (SM) samples estimated from data-driven techniques available; instead, k-factors are applied in some cases (e.g. in multijet MC samples).

4.1.1 TAG file writing and analysis model

The TAG file format is used as an input for TADA after a sequence of processing in the Tier-0. Such format is derived from the Analysis Object Data (AOD) files, which contain a summary of the reconstructed objects of the events. AOD files are themselves derived from raw data formats that come from the detector after the filter of the High Level Trigger (for data) or its simulated output (for MC).

The TAG format stores a reduced representation of the data. It contains the following objects per event[1]: 6 electrons, 6 muons, 4 taus, 4 photons, and 10 jets; as well as missing transverse energy, trigger counters and global event information. For each of the objects attributes corresponding to kinematic quantities (p_T, η, ϕ), and particle identification, isolation and quality information are stored. Moreover, the objects and the event-wide quantities that are written in the TAG file are required to satisfy a set of criteria, motivated by the detector specifics, data quality to be used and physics of interest[2]. The selection of events and objects is further constrained during the TADA processing, that we describe below.

The analysis code from TADA operates in three steps taking as an input the TAG files. On the first step, objects are defined and selected using the criteria where acceptance regions for the transverse momentum and the pseudorapidity of each object are defined as well as other quality and identification requirements. Table 4.1 presents a summary of those criteria along with some coming from the TAG writing. Electrons are required to pass the loose likelihood identification (ElectronIDLikelihoodLoose) [47]. A combined reconstruction (isCombined) is required for the muons, i.e. the tracks reconstructed in the Muon Spectrometer are matched with those in the Inner Detector [48]; muons are also required to pass loose identification requirements (LooseID), as defined in the same reference. Further, a veto for cosmic muons is applied and nearby second muons $dR < 0.01$ is applied. For photons, a loose identification (PhotonIDLoose) defined in ref. [49] is required. In the case of hadron jets, the identification is performed with the anti-k_t algorithm [50] with $R = 0.4$, and the b-jets are identified using the `mv2c10` b-tagger at 70% efficiency as described in [51]. Tau jets are required to pass a medium selection working point (JetBDTSigMedium) from a BDT output from ref. [52].

Also on the first step, a sequence of overlap removal rules are applied on the objects for an angular separation dR, a procedure that is summarized in Table 4.2[3].

The second step uses the definitions of the signal regions for the monitored channels; there, all histograms are booked and filled. Finally, on the third step, postprocessing tasks are performed: plots and tables are produced and the webpage is built with all the results.

[1] Ordered by leading p_T, without charge distinction.

[2] The event and object criteria required, calibration and overlap removal tools used in the TAG writing can be found in the following link: `https://twiki.cern.ch/twiki/bin/view/AtlasProtected/TagForEventSelection210` (restricted access to ATLAS collaboration members).

[3] This table is taken from the internal note available in `https://cds.cern.ch/record/2226510/files/ATL-PHYS-INT-2016-023.pdf`.

Object	p_T	$\|\eta\|$	Other requirements
e	> 10 GeV	$\in (0, 1.37) \cup (1.52, 2.47)$	ElectronIDLikelihoodLoose
μ	> 10 GeV	< 2.7	isCombined LooseID Cosmics veto Reject second muons with $dR < 0.01$
γ	> 20 GeV	$\in (0, 1.37) \cup (1.52, 2.37)$	PhotonIDLoose
(b-)jet	> 40 GeV	< 2.8	AntiKt4TopoJets LooseBadTool (`mv2c10` b-jet tagger)
τ	> 20 GeV	$\in (0, 1.37) \cup (1.52, 2.5)$	JetBDTSigMedium

TABLE 4.1: Object selection requirements in TADA. An explanation for the keywords in the requirements is provided in the text.

Rank	Overlap removal	separation
1	remove jets overlapping with electrons	$dR < 0.2$
2	remove taus overlapping with muons	$dR < 0.2$
3	remove jets overlapping with taus	$dR < 0.4$
4	remove electrons overlapping with jets	$dR < 0.4$
5	remove muons overlapping with jets	$dR < 0.4$
6	remove photons overlapping with electrons	$dR < 0.2$
7	remove jets overlapping with photons	$dR < 0.4$

TABLE 4.2: Overlap removal rules applied in TADA.

4.2 Generic Signatures in TADA

Monitoring only the channels that have a corresponding signature-specific analysis has a clear disadvantage: it leaves a vast amount of data unexplored. This is one of the main motivations for General Searches, where data are compared to simulations in a broad set of channels using an automatic procedure. We follow the same approach in TADA by implementing a system that is able to combine different objects at different multiplicities, thereby establishing a mechanism to bridge the gap between the explored channels and the available data. The webpage containing this generic search can be found in `cern.ch/Atlas/fastphys/tagmon_fabricio/tagmon/`.

Generic searches were present in TADA during a part of LHC Run-1, but they were not maintained because of lack of personpower. The effort of updating generic searches required a major overhaul of the obsolete data-processing code which was adapted to automatically produce both data files and associated information which need to be transparent to the TADA processing and postprocessing and web page systems[4].

4.2.1 Automated Generation of signatures

The generic search system allows the end-user to input a set of constraints and physics objects to be combined automatically. There is a library of functions that

[4]The branch of the software containing the generic channels can be accessed by the collaboration members in `/afs/cern.ch/atlas/project/fastphys/tagmon/dev_fabricio/tagmon`.

```
1     'Selection': '''
2         ( TrigEmuOneMu || TrigEmuOneEl) &&
3         ( IsGoodJetMET ) &&
4         ( (NLooseElectron + NLooseMuon) == {nlep} ) &&
5         ( LooseElectronPt1 > 26000 || LooseMuonPt1 > 26000 ) &&
6         ( NJet == {njet} ) &&
7         ( HT >= {ht} )
8     ''',
9     'SelectionFunc': [require_good_lepton, require_good_jet],
10    'Variables': {
11        'nlep' : [1,2],
12        'njet' : [2,3,4,5],
13        'ht'   : [1000*_GeV, 2000*_GeV],
14        }
```

LISTING 1: Three Python dictionary keys used for generic selections.

allows to take that input and generate several data objects (e.g. selection strings, histogram definitions) that are understandable to TADA; the input is entered in the form of a python dictionary

An example is shown in Listing 1. Here the idea is to have events that pass trigger and quality requirements, and then combine different possible values for the number of jets, leptons and minimum sum of the transverse momentum of all objects in the event (H_T). Lines 1-8 contain the `Selection` key, that is a string with all the criteria required for the events: the first two lines (2 and 3) are the trigger and quality requirements, and line 5 imposes that the leading lepton has a transverse momentum greater than 26 GeV; lines 4, 6, and 7 contain the values that are combined (number of leptons, number of jets and minimum H_T values), indicated by curly braces. The key `SelectionFunc` in line 9 defines an array of functions where, respectively for all leptons and all jets, the functions impose a minimum transverse momentum and quality requirements. The minimum jet p_T values imposed by the function are 420 GeV for the leading jet and 40 GeV for sub-leading ones, whereas the requirement for the leptons' p_T is 10 GeV. The choice of those specific p_T minimum values comes from trigger thresholds, that aim to minimize turn-on effects. The generic variables (`Variables`) defined in lines 11-13 correspond to the possible values of those inside braces in the selection string, i.e. one or two leptons, two to five jets and a minimum H_T of 1 or 2 TeV. This finally leads to 16 different selections.

In the definition of the dictionary for generic selections there is also a key to define the histograms to be filled. For performance purposes we have only used four variables: the invariant mass of all objects selected (M_{inv}), H_T, the missing transverse energy (E_T^{miss}), and the effective mass ($M_{\text{eff}} = H_T + E_T^{\text{miss}}$). There are functions to also automatize the creation of those.

4.2.2 Generic channels monitored

Once the system for creating generic selections was put in place, we defined several channels for monitoring. The channels are organized in groups called *Multijets*, *Multiobjects*, *Several Photons*, and *Leptons plus Jets*; which will be presented below. The system indeed allows for a broader exploration than that shown here, and is left for future work.

The generic channels were put in place during the data taking of 2017, as part of LHC run II, of proton-proton collisions at a center-of-mass energy of $\sqrt{s} = 13$ TeV. The LHC runs recorded by ATLAS that were used by TADA, and are presented in the following, amount to a luminosity of 43.8 fb^{-1} and were processed using release 21 of the ATLAS Athena software. Data from 2015 and 2016 are also available and the merged data for the 2015-2017 period adds up to 68.6 fb^{-1}; this merged dataset uses the 2017 trigger setup.

The simulation campaign from which the Standard Model Monte Carlo samples were derived goes by the name of MC16a. The default set of MC samples defined in TADA were included in the generic signatures below (unless explicitly stated otherwise) and those are:

- $t\bar{t}$+single-t. Samples for top quark pairs ($t\bar{t}$) were produced using Powheg [53, 54] (limiting the `hdamp` parameter to 1.5 times the top mass), Pythia 8 [55, 56] (using the A14 tune [57] and `nnpdf23` [58] at leading-order (LO)) and EvtGen [59]. Single top-quark (single-t) samples were generated using Powheg [53, 54], Pythia 6 [55] (with the Perugia 2012 tune [60]) and EvtGen [59].

- Dibosons. These samples, corresponding to processes generating WW, ZW or ZZ, were generated using Sherpa 2.2.1 [61] using `nnpdf30` at next-to-next-to-leading-order (NNLO).

- W/Z+jets. The processes corresponding to final states with a weak boson plus jets, were also generated with Sherpa 2.2.1 [61] using `nnpdf30` at NNLO for leptonic decays and Sherpa 2.1.1 and the `CT10` pdf [62].

- Multijets. These were dijet samples generated with Pythia 8 [55, 56] using the A14 tune [57] and `nnpdf23` [58] LO and EvtGen [59]. Multijet samples are the combination of samples generated at different ranges of the leading jet p_T value (known as *slices*).

The total MC is normalized to the luminosity of the data collected. In the case of multijet samples, which is a difficult process to model, no data-driven technique is used for estimation, but correction factors are applied to those distributions. The value of such factors was taken from the derivation done in ref. [63]. There, the generated events are reweighted according to a ratio of the cross section of the QCD dijet process calculated at Next-to-Leading-Order (NLO) from the NLOJet++ package [64–66] and that of the same process at LO from the matrix element plus showering from Pythia 8 [55, 56].

Multijets

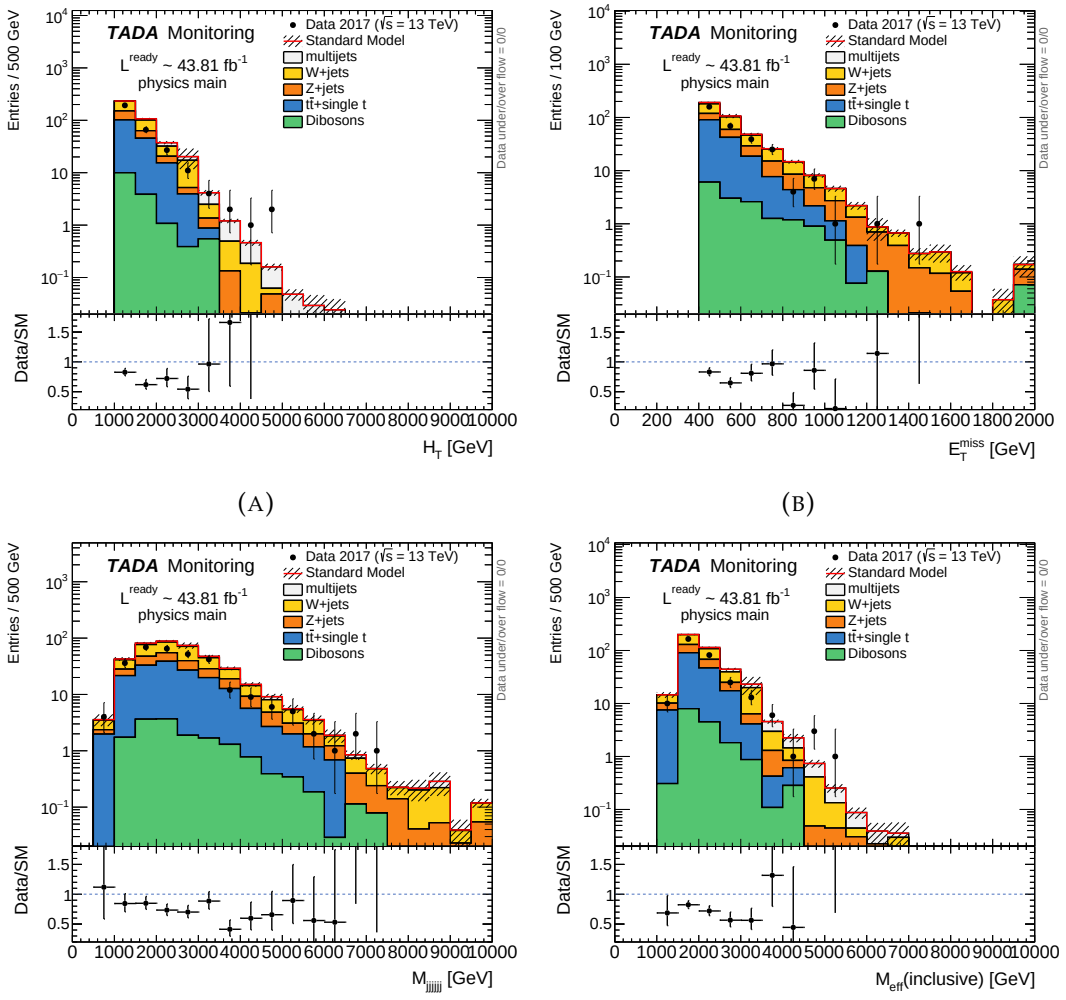
The signatures explored are those with many jets in the final state, with high H_T. Other dedicated set of channels in TADA explores signatures of jets at lower multiplicities, under the Exotics group of signatures. The one-jet trigger `HLT_j420` is required (events with one jet with $p_T > 420$ GeV and a mass of at least 35 GeV) as well as jet/E_T^{miss} quality requirements from the detector, on the events. Here we combine different values of two variables:

- The number of jets, from 6 to 10.

- The minimum value of H_T, 1 and 2 TeV.

There are two further criteria for jets in the events: firstly, the events are required to be separated in the angular plane with $R = 0.4$ or larger (having in mind that jet cones with that same R value are used in the jet clustering algorithm); secondly, the leading jet is required to have $p_T > 420$ GeV and all the sub-leading ones $p_T > 40$ GeV.

Figure 4.1 presents one example of the multijet channels. In the histograms, we observe an approximate agreement between data and MC: a part of the discrepancy arguably has its origin in the multijet MC samples, that are known to be difficult to model, as we have pointed out before. Besides that, the other important contributions in this signature are those from W/Z+jets. During the data-taking periods, no peculiar feature or anomaly appeared on the data.



(A) (B)

(C) (D)

FIGURE 4.1: Example multijet selection: $H_T > 1$ TeV, number of jets equal to six. Plots correspond to each variable monitored: H_T (4.1a), missing transverse energy (4.1b), invariant mass (4.1c), and effective mass (4.1d). Bottom panels present a ratio between the data and Standard Model Monte Carlo.

Multiobjects

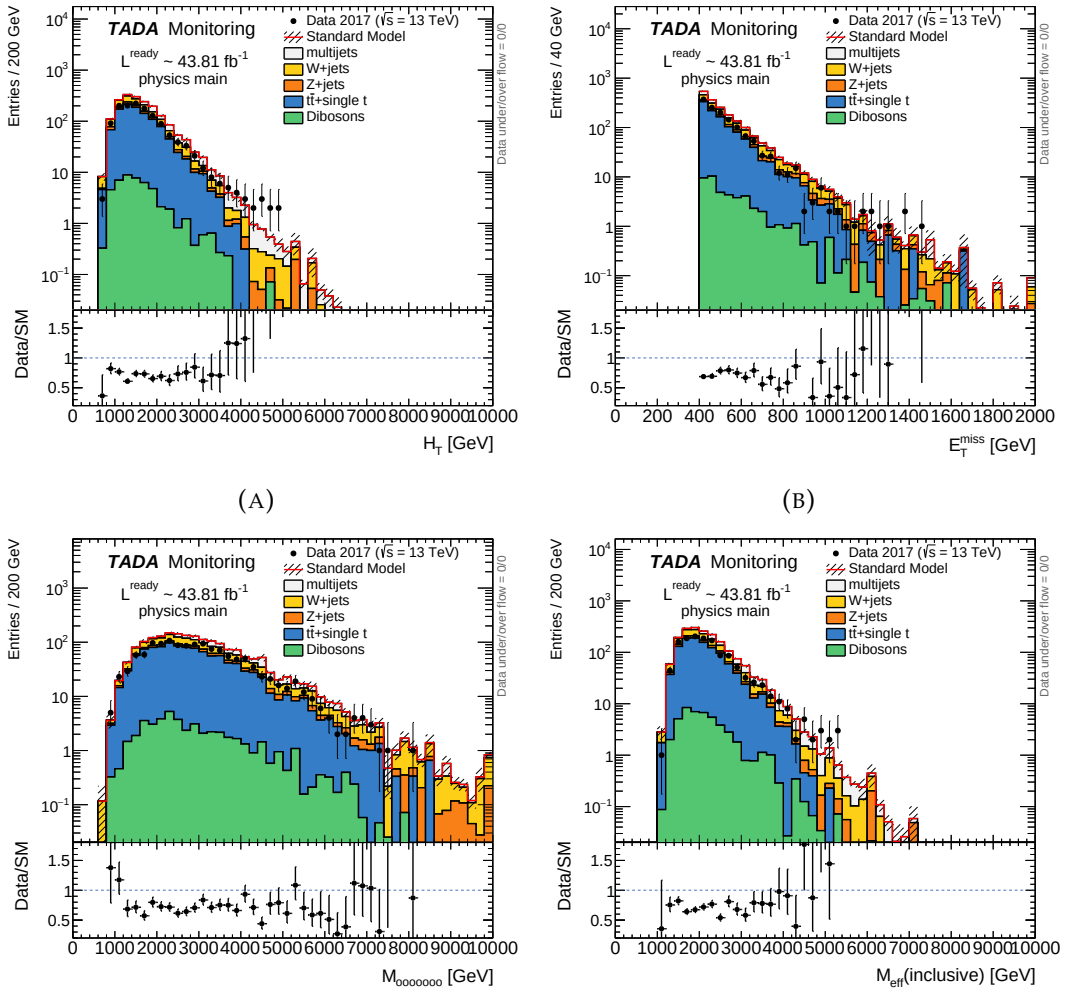
Inclusive high-multiplicity signatures are explored in this group. Similarly to the multijet case, events need to pass the one-jet trigger and jet/E_T^{miss} quality requirements. Events with $E_T^{\text{miss}} > 400$ GeV are selected, and the leading-p_T jet required to pass quality requirements and have $p_T > 420$ GeV.

The variable value entered in the generic dictionary is only one:

- The number of electrons, plus the number of muons, plus the number of jets, greater or equal than 6, 7 or 8.

This leads to three selections.

In Figure 4.2 we can see the plots for the signature of seven or more objects. Once again, we get an approximate agreement between data and Monte Carlo and an overestimation is present mostly due to the multijet contribution.



(A) (B)

(C) (D)

FIGURE 4.2: Example multiobject selection: number of objects (leptons plus jets) greater or equal to 7. Plots correspond to each variable monitored: H_T (4.2a), missing transverse energy (4.2b), invariant mass of the seven objects (4.2c), and effective mass (4.2d). Bottom panels present a ratio between the data and Standard Model Monte Carlo.

Several Photons

Channels with several photons in the final state, and with two different H_T values were monitored. The `HLT_g35_medium_g25_medium_L12EM20VH` trigger is applied on the events; it requires two reconstructed photons with $p_T > 35$ GeV and $p_T > 25$ respectively, and passing a medium identification criterion [49]. The generic values to be combined are:

- The number of photons, two or three.
- The minimum value of H_T, 250 or 500 GeV.

Furthermore, the leading photon is required to have $p_T > 35$ GeV and all the subleading selected ones $p_T > 25$.

In for this set of selections, we have included a SM diphoton sample generated using Sherpa 2.1.1 and the `CT10` pdf [62] for both the hard process and the parton shower, identified as "$\gamma\gamma$ Sherpa". The samples were generated for processes that produce two photons and 0, 1, or 2 jets, in different slices of the diphoton invariant mass, and then combined.

Figure 4.3 presents the signature in which we select two photons and impose $H_T > 250$ GeV. General good agreement is found in the SM histograms with respect to the data, except in the E_T^{miss}, where the simulations underestimate the data obtained. This deficit possibly has its origins in the fact that the pile-up spectrum obtained in data had an important disagreement with respect to simulations during the 2017 data taking. Given that pile-up affects (adds noise) to the E_T^{miss} calculation, we can expect some impact in that variable[5]

Leptons plus Jets

This corresponds to the example that we quoted at the beginning of section 4.2.1, and Listing 1. Either the single muon or the single-electron trigger, respectively labeled by the strings `HLT_mu26_ivarmedium` or `HLT_e26_lhtight_nod0_ivarloose` is applied: the single muon trigger requires events with a muon with a 26 GeV minimum p_T threshold and a medium gradient isolation requirement as defined in [48], and the single electron trigger also requires one electron with a minimum p_T of 26 GeV, a likelihood-based tight identification and a loose gradient isolation requirement, as explained in ref [67].

The variable values combined (leading to 16 selections) are:

- The number of muons plus the number of electrons equal to one or two.
- The number of jets from two to five.
- The minimum value of H_T, 1 or 2 TeV.

In Figure 4.4, we present the plots for one example: one jet, two leptons and $H_T > 1$ TeV. This and other selections inside this group are dominated by the W+jets contribution for requiring the presence of the lepton(s) on top of the jet(s). General good agreement is found between the data and Monte Carlo, and no particular data feature appeared during data-taking periods.

[5]The interested reader can check the W validation plots page, in particular `https://atlas.web.cern.ch/Atlas/fastphys/tagmon_fabricio/tagmon/shared-images/2017/val_W/val_Wenu__AvgIntPerXing_both.gif`.

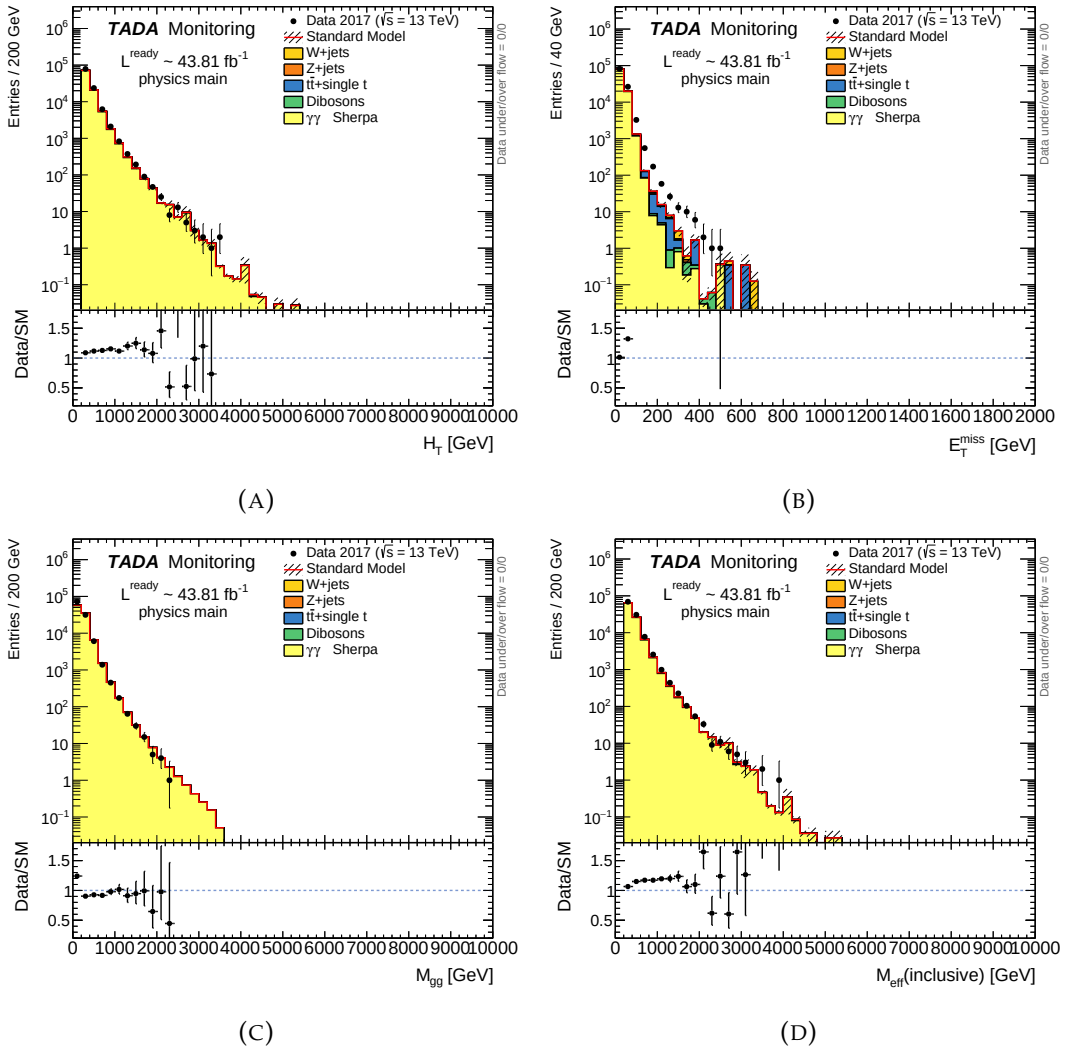
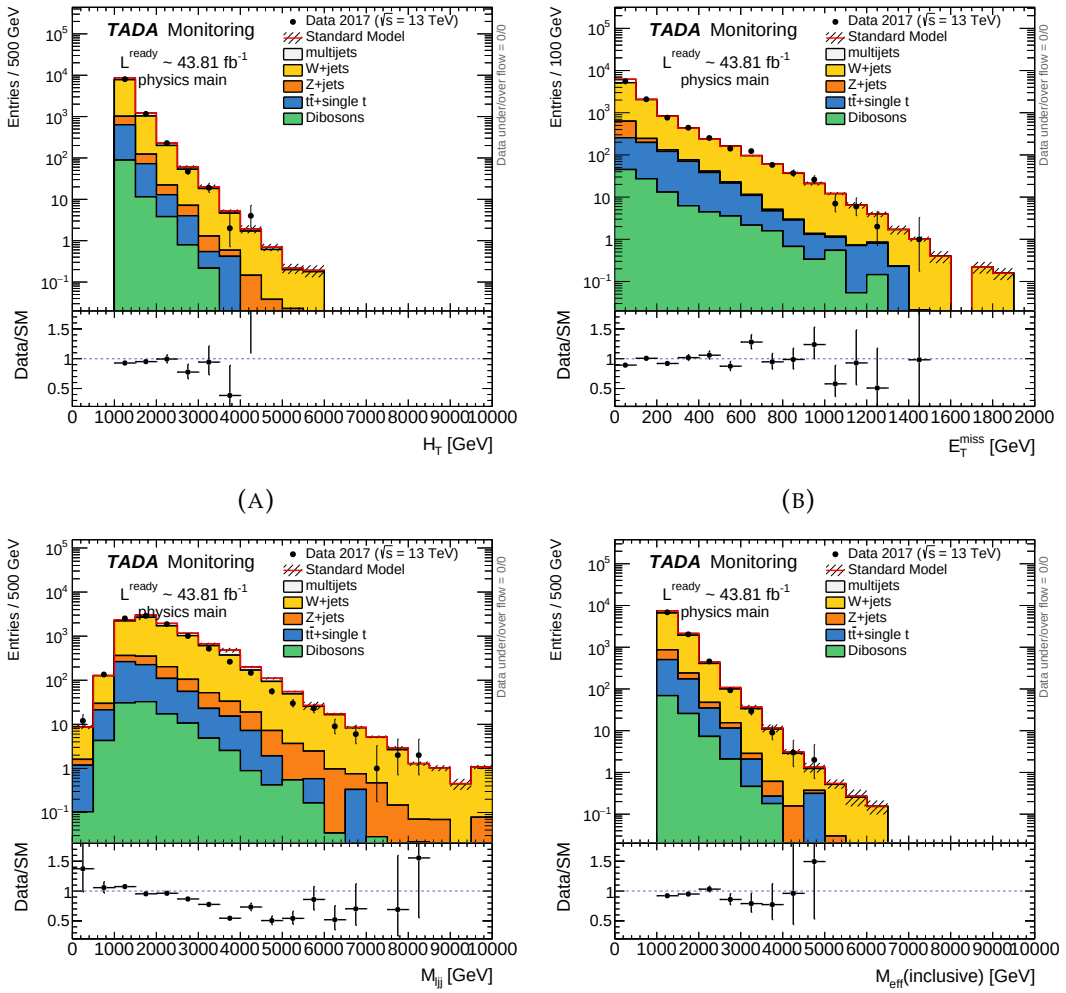


FIGURE 4.3: Example diphoton selection: $H_T > 250$ GeV. Plots correspond to each variable monitored: H_T (4.3a), missing transverse energy (4.3b), invariant mass (4.3c), and effective mass (4.3d). Bottom panels present a ratio between the data and Standard Model Monte Carlo.

4.3 Conclusion

Within the TADA monitoring system we have put in place a system that, inspired in General Searches performed by the ATLAS collaboration [45], allows for automatically combining different objects and selection criteria. Four sets of signatures were defined and monitored during data-taking period of 2017, generating over 30 different signatures that lead to over 120 histograms filled, a few of which have been shown here. Signatures monitored are summarized in table 4.3. Even if TADA cannot lead to discovery claims, during periods when the data luminosity doubles at a fast rate (e.g. 2016 and 2017). This monitoring tool becomes of great utility for the collaboration, as it can provide quick feedback to specialists with offline monitoring, processing new runs in the tier-0 twice a day. No particular anomalous behavior was observed in the data during data-taking periods. The system for creating generic selections certainly allows for a broader exploration, that is left for future work.



(A) (B)

(C) (D)

FIGURE 4.4: Example lepton plus jets selection: number of leptons equal one, number of jets equal two, $H_T > 1$ TeV. Plots correspond to each variable monitored: H_T (4.4a), missing transverse energy (4.4b), invariant mass (4.4c), and effective mass (4.4d). Bottom panels present a ratio between the data and Standard Model Monte Carlo.

Group	# Selections	Variables
Multijets	10	Number of jets = $\{6, 7, 8, 9, 10\}$ $H_T > \{1, 2\}$ TeV
Multiobjects	3	Number of objects = $\{6, 7, 8\}$
Several Photons	4	Number of photons = $\{2, 3\}$ $H_T > \{250, 500\}$ GeV
Leptons plus jets	16	Number of leptons $(e, \mu) = \{1, 2\}$ Number of jets = $\{2, 3, 4, 5\}$ $H_T > \{1, 2\}$ TeV

TABLE 4.3: Summary of the 31 generic selections in TADA. For each of the selections, four distributions were monitored.

Chapter 5

Machine Learning in High Energy Physics

5.1 Motivation

Machine Learning (ML) is a discipline at the intersection of computer science and statistics. Without being explicitly programmed to perform a specific task, ML algorithms *learn* a mathematical model, i.e. approximate a function, by using observed real or simulated data for improving their performance on that task. (Many somewhat similar definitions of ML can be found in the literature; see e.g. chapter 18 of ref. [68].) Ultimately, the algorithm should be able to make predictions on future data.

Very broadly, learning tasks can be separated into two main types: unsupervised and supervised. Supervised learning tasks consist in building models for which the desired output is known in the training dataset, whereas unsupervised tasks aim to extract information (e.g. patterns) in the training dataset where no desired outputs are provided. Some techniques lie in between the two categories above; for example, active learning [69] (access a limited amount of desired outputs for optimizing performance) or semi-supervised learning [70] (desired output available only for a fraction of the training set), the latter of which is discussed in further detail in Chapter 6.

Statistical models and tools are at the core of any experimental data analysis, and more specifically ML has found a broad set of applications in physical sciences including High Energy Physics. Applications lie in different sub-domains, such as detector design and calibration [71], simulation techniques [72], and particle identification and event discrimination [73], to name a few. Most of such applications have led to significant improvements in the discovery potential of new particles, including those proposed in beyond-the-SM scenarios, e.g. in ref. [74]. Optimal analysis is particularly needed when dealing with experiments that are expensive to build and run as they often happen in High Energy Physics.

There are a few reasons why ML techniques are well suited for particle collision data analyses. Despite the monumental success that the SM has seen as a scientific theory, it is in general difficult to build a statistical model from first principles within the SM (see Chapter 2) to compare with measured data. Further, at a quantum level interactions are probabilistic, from the collision itself to the interaction of final particles with the detectors. Our theoretical predictions are therefore based on simulations that rely on a set of approximations and effective models that are tuned with the help of already-available data, and then those simulations are tested against measurements. Finally, collisions producing interesting physics are rare and thus high-energy collider experiments are designed to produce large volumes of high-dimensional data; learning statistical models from data and dealing with high

dimensions are two of the main goals of ML techniques in general, and where they have been shown to succeed [75].

For decades, experimental particle physicists have used machine learning techniques. Artificial Neural Networks (NNs) were first introduced in particle physics back in 1987 for addressing track and cluster finding in detectors [76], and Boosted Decision Trees (BDTs) in 2004 for particle identification [77]. The use of BDTs and NNs has gone beyond their initial applications, with uses in on- and offline event selection, parameter estimation, and others. These techniques along with other used in particle physics have been traditionally referred to as Multivariate Analysis Methods, and have been reviewed e.g. in ref. [78]. Notably, Machine Learning techniques have had an impact in the early measurements of the top quark mass [79] in 1997 and the Higgs boson discovery [14, 15] in 2012.

In recent years, the advent of larger datasets, new software capabilities and theoretical developments during the last couple of decades have fueled a revolution in ML with a direct impact in science and industry. Given that research in ML advances at an increasingly-faster pace, mostly due to incentives in the industry, physicists are facing the challenge of porting all pertinent knowledge to improve their analyses, as it is stated in ref. [80].

5.2 Supervised learning

A supervised learning problem is typically put as finding a function $f : \mathcal{X} \to \mathcal{Y}$ that maps the inputs $\boldsymbol{x}$ with D features (i.e. dimensions) to a lower-dimensional space, where outputs (targets) $\boldsymbol{y}$ exist. In the training dataset, inputs $\boldsymbol{X} = (\boldsymbol{x}_1; \cdots; \boldsymbol{x}_N)$ contain N observations (e.g. collision events) and the N desired outputs $\boldsymbol{Y} = (\boldsymbol{y}_1; \cdots; \boldsymbol{y}_N)$, some quantity of interest. The case in which the desired output is not used (or not available) for finding f is referred to as unsupervised learning, discussed below in 5.3.

In supervised learning, finding the best f, i.e. training, means optimizing some *loss* function $L(f(\boldsymbol{X}), \boldsymbol{Y})$. This function measures the goodness of the prediction $f(\boldsymbol{X})$ with respect to the desired output $\boldsymbol{Y}$. Models such as NNs or BDTs are examples of functions f that are parameterized by a set of *hyperparameters* θ. Optimizing the loss function then consists in tuning hyperparameters, and several algorithms have been designed for such task and are available in the market depending on the model used, as we will see below.

The fact that ML models follow this "learning from data" approach, often with few or no assumptions on the true underlying process, can lead to some limitations. Since such models are tuned with data that can be noisy, limited and/or biased, the case is often that the resulting models are prone to overtraining; this is, simply put, training to an extent in which the model continued increasing performance on the training dataset but decreasing it on a new dataset. Overtraining normally means that our model has learnt unnecessary details from data, that leads to high variance error in unseen data. In contrast, undertrained models are overly simple and therefore prone to bias error; the goal consists in finding a good tradeoff between bias and variance [81]. Regularization techniques by means of constraining the loss function are used to avoid overtraining; a use of regularization in another context is presented in Chapter 6.

Arguably, the two most important supervised ML tasks that are performed in practice are classification and regression. On one hand, classification deals with data whose output has two or more labels, where one would aim to find boundaries in

the feature space that identify regions associated to the different labels. Regression, on the other hand, consists in approximating a function of continuous outputs. We briefly describe below a few conventional methods that appear in standard textbooks in ML such as refs. [82] or [83].

Methods that use a linear combination to map the input variables to f are known as *linear methods* and can be written:

$$f(\boldsymbol{x}, \boldsymbol{w}) \sim \boldsymbol{w} \cdot \boldsymbol{x} = w_0 + w_1 x_1 + \cdots + w_D x_D \,, \tag{5.1}$$

where $\boldsymbol{w}$ is a vector of *weights* (parameters) w_i that need to be optimally chosen, for example, via the maximization of the likelihood, among other methods. Further, this linear mapping can happen via a basis of functions $\boldsymbol{\phi}$:

$$f(\boldsymbol{x}, \boldsymbol{w}) \sim \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) = w_0 + w_1 \phi_1(\boldsymbol{x}) + \cdots + w_D \phi_D(\boldsymbol{x}) \,, \tag{5.2}$$

that allows for more flexibility than the model in eq. (5.1).

The simplest example of regression is linear regression. There, one regresses the value of the desired output y onto x, assuming a linear relationship of the kind $y \sim f(x, \boldsymbol{w}) = w_0 + w_1 x$ and finds the weights by minimizing the residual sum of squares:

$$RSS(\boldsymbol{w}) = \sum_i^N (f(x_i, \boldsymbol{w}) - y_i)^2 \,. \tag{5.3}$$

An extension of linear regression to many features (as in the case of eq. 5.1) is referred to as multiple linear regression.

In a number of cases, linear regression with the ordinary least squares method is too simple of a prescription to find the best fit. Several improvements have been devised to aim for a more robust function than least squares (i.e. less prone to overtraining), notably by adding a penalty term to eq. 5.3 as a form of regularization. The form of the loss function then becomes:

$$L(\boldsymbol{w}) = RSS(\boldsymbol{w}) + \gamma p(\boldsymbol{w}) \,, \tag{5.4}$$

where p is a penalty function on the parameters and γ a parameter that needs to be chosen to regulate the penalty. Two important examples that are now standard in the literature are the so called $L1$ and $L2$ penalties, that lead to Lasso and Ridge regression, respectively. For reference, the respective penalty terms are written:

$$p_{\mathrm{L1}} = \sum_j^D |w_j| \,, \tag{5.5}$$

$$p_{\mathrm{L2}} = \sum_j^D w_j^2 \,. \tag{5.6}$$

We will see an example of penalization in the context of anomaly detection in Chapter 6.

Linear regression can also be formulated in a Bayesian context, where one infers a (posterior) distribution of parameters. The underlying machinery is based in the use of Bayes' theorem: postulate a prior probability on the weights and use the likelihood to update our knowledge about the distribution of weights. The optimization procedure for a Bayesian linear regression is often analytically intractable and one

has to make use of methods like Markov Chain Monte Carlo to perform the optimization by sampling the posterior. An example of an exception where Bayesian linear regression is tractable is the case in which we postulate Gaussian priors over the parameters; there, maximizing the posterior is equivalent to performing least squares with an $L2$ penalty, i.e. Ridge regression.

A more general and powerful regression tool that can be devised using Bayes' theorem are Gaussian Processes (GPs) [84]. Used for non-linear regression, GPs consist in associating a N-dimensional joint Gaussian distribution to the N data points. The mean of the joint Gaussian is constrained (conditioned) by each y_i and the covariance matrix is given by the correlation between pairs of points; it is possible to infer new values of $y^* \in \mathcal{Y}$ for an arbitrary $x^* \in \mathcal{X}$ through some standard algebraic procedure called completing the squares and assuming a Gaussian likelihood. A continuous set of regressed values is achieved in the limit in which the GP has an infinite-dimensional joint Gaussian distribution. In practice, one introduces a positive-definite parametric function called *kernel* which is a measure of similarity to model the covariance between pairs of points, instead of using the covariance computed from the data. Kernel parameters are obtained via maximum likelihood estimation to perform the regression. In Chapter 7 we provide greater detail on the theoretical bases of GPs and an application in High Energy Physics.

Linear methods have been employed in classification. More formally, this consists in finding a separation of classes of events in the feature space, as it is the case of *logistic regression* and *linear discriminant analysis* (LDA). In logistic regression for binary classification, i.e. where we two possible outputs commonly labeled as $\{0, 1\}$, the linear combination is mapped to the class prediction f via the *logit* link:

$$\log\left(\frac{f(\boldsymbol{x})}{1-f(\boldsymbol{x})}\right) = \boldsymbol{w}\cdot\boldsymbol{x}\,, \tag{5.7}$$

where f can be understood as the probability of belonging to class "1". Normally one decides a threshold in f to decide whether a prediction will be identified in either class. The key idea behind LDA is to construct and maximize a discriminant quantity from the means and (co)variances of sampled events from each class, where the linear combination in eq. (5.1) defines a separation between the classes. A classic example is the Fisher discriminant,

$$F(\boldsymbol{w}) = \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^2}{\sigma_1^2 + \sigma_0^2}\,, \tag{5.8}$$

where weights can be obtained via

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\,, \tag{5.9}$$

with $\boldsymbol{\Sigma}$ the common covariance matrix of the samples. LDA is more stable than logistic regression when the classes are well separated and when the sample size is small. Detailed descriptions of those methods can be found in e.g. chapter 4 of reference [85].

Below, we describe two methods that have been frequently used in High Energy Physics for several years: Boosted Decision Trees and Artificial Neural Networks.

5.2.1 Boosted Decision Trees

Two common tasks in data analyses from detectors at the LHC are particle identification and signal versus background discrimination. An example of an analysis where Boosted Decision Trees (BDTs) have been used multiple times for such tasks can be found in ref. [73]. In that work, photons are identified by using variables of different kinds: some are related to the shape of energy deposits in the calorimeter, and other to the kinematic and angular properties of the deposits. BDTs are also used for selecting events that contain two photons with certain features, that are related to the signal of interest, and at the same time rejecting those that come from uninteresting SM physics.

An accessible example of supervised ML is a tool that uses a tree model to make decisions about the data, known as Decision Trees, and further help with tasks such as classification and regression. Decision Trees have become popular for being easily interpretable and scaling well with the dimensionality of the dataset, among other reasons. In High Energy Physics, *boosted* Decision Trees which we will discuss below, have become the de facto supervised ML tool for several tasks and have been implemented in TMVA[86].

Decision Trees contain a set of nodes where the data is split (forked) into branches, that can feed other nodes for further forking. The data enter through a node, the *root node*, and is forked into two branches according to a criterion defined by a threshold on a single variable, also known as *cut*[1]. An observation $\boldsymbol{x}_i$ belongs to one branch if it satisfies a criterion $C_j : x_{ij} > c_j$ for a variable indexed by j, otherwise the observation is forked into the other branch. The process is repeated once the data from a branch arrives to a new node. Branches that do not input data to another node are known as leaf nodes. The tree then grows until a prescribed stopping criterion is met, like a maximum number of leaves, no more purity is gained, et cetera. The idea of limiting and modifying the amount of nodes and leaves is known as *pruning*. An example model of a decision tree is presented in Figure 5.1.

Each leaf node of the Decision Tree identifies a hyperrectangle in the feature space. The goal of going through such sequence of decisions is to increase the purity in the result, as measured by some metric (e.g. the Gini index or the cross-entropy), that would ultimately amount to optimizing the loss function.

The bias-variance tradeoff is also present in DTs. As we know, the amount of nodes should optimize the performance on unseen data; on one hand, if there are few nodes, the tree can be overly simple and highly biased and, on the other hand, many nodes can lead to hyperrectangles constructed with a small number of data points, where generalization is difficult and prone to high variance error.

We will illustrate how boosting is performed with decision trees for classification. Boosting is a popular ensemble method that combines a sequence of learners, e.g. Decision Trees, to build a single, more powerful learner, that has been shown to reduce bias and variance [87]. The basic idea underlying Boosted Decision Trees (BDTs) is to iteratively grow *weak* classifiers f^{weak} (i.e. DTs that perform marginally better than random choice) to form a *strong* BDT. At a given iteration p, many DTs are grown and one f_p^{weak} is chosen, such that it minimizes the misclassification rate, from which a learner weight α_p that assigns the importance of the data points is calculated; misclassified points become more important in subsequent iterations. After

[1] Most Decision Trees used in practice are binary, i.e., contain nodes forking into two branches, but in principle there could be more branches per node; since the number of nodes grows exponentially with the amount of branches per node, however, the complexity of the tree increases much more rapidly in non-binary trees.

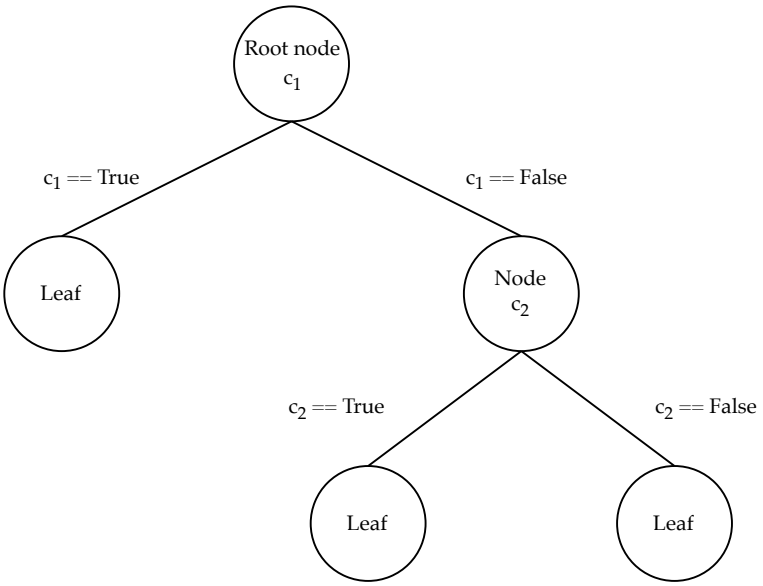


FIGURE 5.1: Binary Decision Tree of depth equal three.

P iterations we construct the BDT output

$$f(\boldsymbol{x}) = \sum_{p=1}^{P} \alpha_p f_p^{\text{weak}}(\boldsymbol{x}, \theta_p)\,, \tag{5.10}$$

where θ_p corresponds to the parameters of the p-th tree and α_p is determined by the specifics of the boosting algorithm.

Once the tree is trained, one can use an independent dataset to assess the purity of each leaf. For example, if we have a binary tree, and the output labels are denoted by $y_i = \{0, 1\}$, the result at each leaf will be a real number contained in the interval $[0, 1]$. We can then decide on an output threshold value, with which we classify the data point on either class, normally according to the specifics of our problem, e.g. preference of type of error. We will denote the inferred class $\tilde{y}_i \in \{0, 1\}$.

Among the most popular boosting algorithms used in BDTs are Adaptive Boosting (AdaBoost) [88], Gradient Boosting [89], and Extreme Gradient Boosting (XGBoost) [90]. The AdaBoost algorithm starts by uniformly initializing event weights

$$w_i^{(1)} = \boldsymbol{w}^{(1)}(x_i, y_i) = \frac{1}{N} \tag{5.11}$$

for N observations. At iteration p, the weak classifier f_p^{weak} is chosen such that it minimizes the weighted misclassification error

$$\epsilon_p = \sum_{\tilde{y}_i \neq y_i} w_i^{(p)}\,, \tag{5.12}$$

then the learner weight is calculated for that classifier, as

$$\alpha_p = \frac{1}{2} \ln\left(\frac{1 - \epsilon_p}{\epsilon_p}\right)\,, \tag{5.13}$$

and finally event weights are updated for each data point

$$w_i^{(p+1)} = \frac{w_i^{(p)} \exp\left(-\alpha_p y_i f_p^{\text{weak}}(x_i, \theta_p)\right)}{Z^{(p)}}, \tag{5.14}$$

where $Z^{(p)}$ is a normalizing constant.

Gradient Boosting is similar to AdaBoost, except that Gradient Boosting uses a differentiable residual error (loss) to perform boosting at each iteration, instead of the update rule in eq. (5.14). It is essentially applying Gradient Descent algorithm in BDTs (see section on Neural Networks below). XGBoost is a powerful type of Gradient Boosting that has become popular for its speed and performance, and has gained popularity within the High Energy Physics community after several appplications, notably at the HiggsML Challenge [91].

5.2.2 Artificial Neural Networks

Artificial Neural Networks, or simply Neural Networks (NNs), are a class of models that, loosely inspired in the human brain, process information through connected *neurons*. Neural Networks aim to approximate a complex function by composing multiple processing units (the neurons) that are simple functions.

A neuron takes a (set of) value(s) as an input and uses an *activation function* g to produce an output value; the collective effect of all neurons is f, our NN, that we will train. The parameters of a NN that need to be optimized during training are known as *weights* ($\boldsymbol{w}$) and *biases* (b), of each activation function. The amount of neurons, the connections among them and the choice of activation function(s) are collectively referred to as the *architecture* of the network. Altogether,

$$h_{\text{out}} = g(\boldsymbol{w} \cdot \boldsymbol{h}_{\text{in}} + b), \tag{5.15}$$

where the input and output values are represented, respectively, by $\boldsymbol{h}_{\text{in}}$ and h_{out}.

One of the most studied examples of NNs is the Multilayer Perceptron, which consists on a sequence of sets of neurons known as layers; the data $\boldsymbol{x}$ enter through an *input* layer, and then the information flows through the *hidden* layers, until a final *output* layer retrieves the value of $f(\boldsymbol{x})$. For a layer indexed by t, we can construct a matrix of weights $\boldsymbol{W}_t$ from all weight vectors $\boldsymbol{w}$ of each neuron, and similarly a vector $\boldsymbol{b}_t$ from the biases. If all neurons of a layer have the same activation function $\boldsymbol{g}_t$, as it often happens in practice, the output of one layer can be put as

$$\boldsymbol{h}_{t+1} = \boldsymbol{g}_t(\boldsymbol{W}_t \cdot \boldsymbol{h}_t + \boldsymbol{b}_t). \tag{5.16}$$

Note that $\boldsymbol{h}_t$, $\boldsymbol{b}_t$, and $\boldsymbol{h}_{t+1}$ are vectors with dimensions D_t, D_t, and D_{t+1} respectively, i.e. the amount of neurons on layers t and $t+1$, and the matrix of weights has dimension D_t times the number of weights in the activation function. A traditional choice of activation function is the sigmoid, defined by:

$$\sigma(z) = \frac{1}{1+\exp(-z)}, \tag{5.17}$$

which is equivalent to the logit link in eq. (5.7). A more modern choice is the Rectified Linear Unit (ReLU) activation function, defined as

$$\sigma(z) = \max(0, z), \tag{5.18}$$

that helped overcome what was known as the "vanishing gradients" problem [92]; the softmax has also found use in classification with Convolutional Neural networks, that we will discuss shortly:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{D_T} e^{z_j}} \tag{5.19}$$

where both $i, j = 0, \cdots, D_T$ with D_T the dimension of the last layer. An example of a graph that represents a Multilayer Perceptron is presented in Figure 5.2.

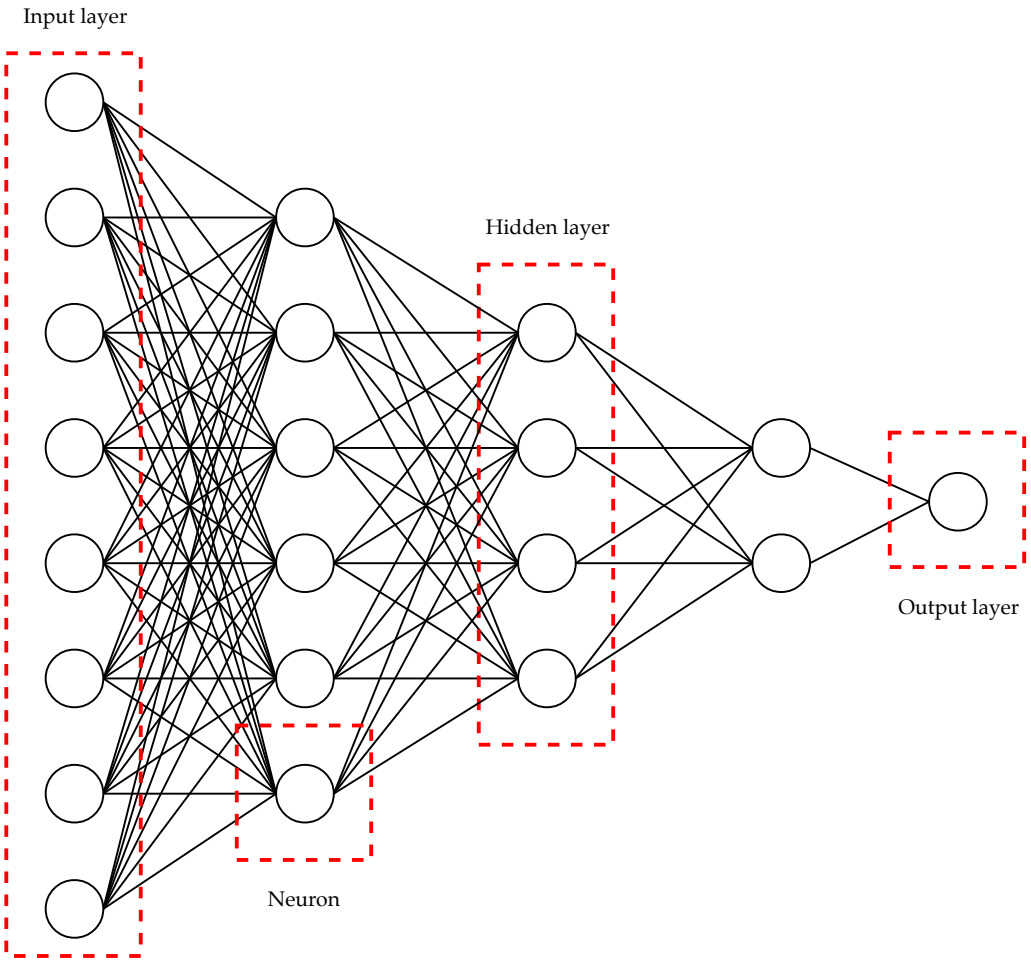


FIGURE 5.2: A feed-forward Neural Network: the fully connected Multilayer Perceptron. There are 8 neurons on the input layer, then 6-4-2 on the subsequent hidden layers, and an output layer of a single neuron.

A common algorithm for training Neural Networks is known as Gradient Descent with backpropagation. Gradient Descent consists in using a differentiable loss function where, at each iteration, a gradient step is performed in the space of parameters (weights and biases) of the NN for optimization. Since by construction NNs are composite functions, and each of the neurons uses a differentiable activation function[2], it is possible to use the chain rule to compute the gradient of the loss function.

[2]In practice, there exist some activation functions that are non-differentiable at a few points, where the value of the derivative may be prescribed. The ReLU activation is an example of this case.

The update of a gradient step in the parameter space has the form:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{X}, \boldsymbol{\theta}) \,, \tag{5.20}$$

where α is an adjustable parameter referred to as the *learning rate*, that dictates the magnitude of the gradient step. The backpropagation (short for backward propagation) of the error consists in computing the effect that each intermediate neuron, from the input to the output layer, had in the final loss.

The standard form of gradient descent in eq. (5.20) is also known as *batch* gradient descent, as the loss function is calculated on a set (all) of the training examples. A faster alternative consists in calculating the loss for one randomly chosen training example, that can lead to finding an optimal solution with less computations. This is known as Stochastic Gradient Descent. Mini-batch gradient descent uses the intermediate approach of using a few n training examples for calculating the gradient of the loss.

Multilayer Perceptrons have been used with certain success in High Energy physics for different tasks, such as particle identification (including trigger applications), parameter measurement, event selection, as it has been reviewed, e.g. in ref. [93].

Although conceptually NNs have several decades of existence, during the last decade they have regained popularity, notably in tasks such as image classification [94]. It is known that Neural Networks with as few as one hidden layer, i.e. shallow NNs, can be universal function approximators [95], but more recently it has been shown in practice that Deep NNs (with many hidden layers) are more powerful and efficient than shallow ones, as they may not require to have an intractably high number of neurons (see e.g. [96]). We will discuss further the use of Deep Learning in section 5.4 below.

5.3 Unsupervised and semi-supervised learning

A different paradigm in ML is known as *unsupervised* learning where, in contrast to supervised learning, the desired outputs are not used for training. We thus let the method use a set of input data to learn a representation that can then be useful for discovering patterns, to summarize information contained or other tasks. Reviews of unsupervised learning and methods can be found in e.g. ref. [97] and Chapter 10 of ref. [85]. The so-called *semi-supervised* paradigm takes elements of supervised and unsupervised learning, using a data set whose outputs are known partially. We devote the rest of this section to defining and briefly describing these paradigms.

5.3.1 Unsupervised learning

Unsupervised methods use input data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}, \boldsymbol{x}_i \in \mathcal{X}$ to build a representation in another space. Such representation can be useful for getting insight on the data, for example, by identifying localized concentrations (clusters) of data points, or performing dimensionality reduction; such examples are traditionally the two most studied unsupervised tasks.

Among the main motivations of using unsupervised methods is the lack of knowledge of the desired output. Further, letting a method learn in an unsupervised manner can be useful for identifying unanticipated properties of the data. For example, if we started with a high-dimensional feature space and wanted to identify clusters, it may not be evident to realize that a given point belongs to a cluster or not by visual inspection of projections of the data in, e.g. a two-dimensional space.

Clustering relies on the introduction of some metric or distance which has to be chosen based on domain knowledge of the problem. Clusters can then be built as sets of points that are close to each other according to the metric. A simple and popular clustering method is the so-called *K-means*: it consists in identifying K clusters in the data using the Euclidean distance. The method starts with randomly assigning a cluster number from 1 to K to each observation, then compute K mean vectors (known as *centroids*), one for each cluster number, and assign each observation to the cluster that has the closest centroid. The last two steps, i.e. the computation of the mean vectors and assignment, are iterated until the assignment stops changing. K-means optimizes (i.e. finds a local minimum for) the following function for all clusters $\{C_1, \cdots, C_K\}$:

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{D} (x_{ij} - x_{i'j})^2 \,, \tag{5.21}$$

where $|C_k|$ is the number of observations on cluster k.

A more general probability density model than that underlying K-means can be constructed from a mixture of Gaussian distributions. A Gaussian Mixture Model (GMM) is prescribed, for K mixture components, by:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \,, \tag{5.22}$$

where the parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, for $k = 1, \cdots, K$, are respectively positive mixing proportions, means and covariance matrices of Gaussian distributions ($\mathcal{N}$); these parameters are collectively denoted $\boldsymbol{\theta}$. (Mixing proportions satisfy $\sum_{k=1}^{K} \pi_k = 1$.) A way of estimating maximum likelihood parameters of a GMM is given by the Expectation-Maximization (EM) algorithm that we describe in some detail in Chapter 6. We recover K-means if we apply the EM algorithm to a GMM, choosing the same covariance matrix ($\sigma^2 \mathbb{1}$) and mixture proportions ($1/K$) for each mixture component, in the limit $\sigma \to 0$; then $\boldsymbol{\mu}_k$ correspond to the centroids of K-means [98].

Another well studied and widely applied clustering algorithm is DBSCAN [99]. This is a non-parametric algorithm that given two parameters, ϵ (neighborhood radius) and a minimum amount of points to define a dense region ϵ, is able to connect points in the feature space, thus defining clusters. In comparison to K-means, DBSCAN does not need an a-priori input of the number of clusters to be used, it is able to identify clusters of arbitrary shape, and it is more robust to outliers.

We can find a successful application of clustering in High Energy Physics in identifying hadron jets. Since jets are produced within a cone and leave a spread trace of different particles in the detector, we are able to use clustering to identify jets in the detection region. Popular jet clustering algorithms are the k_t algorithm [100], the Cambridge-Aachen algorithm [101] and the anti-k_t algorithm [102], the latter of which has become the default jet clustering tool at CMS and ATLAS analyses, because it has the desirable property of being safe to infrared gluon emissions. The distance in the anti-k_t algorithm is defined using the magnitude of the transverse momentum (k_t) of pairs of objects i, j:

$$d_{ij} = \min(k_{t,i}, k_{t,j}) \frac{\Delta_{ij}^2}{R^2} \,, \tag{5.23}$$

where the squared distance in the rapidity (y) and azimuthal angle (ϕ) plane is: $\Delta_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$ and R is a fixed parameter (a common value in ATLAS is 0.4).

Dimensionality reduction is a crucial task in ML for several reasons. For example, one may want to transform a dataset with many features into a lower (e.g. two) dimensional representation to be able to visualize the data, or to reduce the computational expense in some calculation such as the ones that occur in a learning algorithm. A lower-dimensional representation of the data in general implies a loss of information.

Often such reduction of the dimensionality is imperative as a pre-processing step, before inputting data into a ML algorithm. Many ML methods, $K-$means included, suffer from the so-called *curse of dimensionality*; this is, the algorithm degrades its performance or rapidly increases the computational expense when augmenting the amount of features used. The curse of dimensionality has its origin in the exponential growth of the volume in a (feature) space with the number of dimensions, see e.g. ref. [103].

A standard dimensionality reduction technique is the *Principal Component Analysis* (PCA). It consists in finding a set of uncorrelated axes (the principal components) from a linear combination of the features and ordering them by the amount of variability they carry. The aim of selecting a few of the first principal components is to retain as much information as possible in a low-dimensional space. In short, the first principal component is defined by a vector in the feature space in a direction in which the data vary the most; the second component is defined by a linear combination that is orthogonal to the first principal component and that has maximal variance; this procedure is extended to find further principal components. PCA, among others such as Independent Component Analysis or Factor Analysis, falls into the category of linear decomposition methods.

More modern techniques for dimensionality reduction employ non-linear methods or manifold learning. These approaches use e.g. non-linear combinations of features (in contrast to the linear combination used in PCA), or learn a manifold in the feature space where the data is projected. An example of these kind of methods is t-distributed stochastic neighbor embedding (t-SNE) [104].

5.3.2 Semi-supervised learning

Semi-supervised learning (SSL) is a paradigm in-between supervised and unsupervised learning [105]. The data provided for training the method is partially labeled, this is, the labeled pair of inputs and targets $\boldsymbol{X}_l, \boldsymbol{Y}_l$ and the unlabeled set $\boldsymbol{X}_u$. In this work, we adopt the conventional interpretation of assuming SSL as a form of supervised learning where additional information on the input, $\boldsymbol{X}_u$, is provided.

Semi-supervised learning, to the best of our knowledge, has rarely seen applications in experimental High Energy Physics. A method for detecting signals as collective anomalies using semi-supervised learning is presented in ref. [106]; it uses Gaussian Mixture Models in a two-step procedure. Another method that tackles the same problem (with the same dataset) by using one-class support measure machines can be found in [107]. In the use case considered in those references and the extension proposed in Chapter 6, as opposed to most SSL applications, the sample size of the labeled set $(\boldsymbol{X}_l, \boldsymbol{Y}_l)$ is not necessarily much smaller (or not smaller at all) than that of the unlabeled set $\boldsymbol{X}_u$; producing a (simulated) labeled dataset does not entail a larger computational cost than producing an unlabeled one.

In this setup, the goal is to identify, if present, a subset of observations in data that can be indicative of a New Physics signal and that deviates from the distribution of background events (i.e. the SM). More formally, the idea is to model the background distribution from a pure SM background (labeled) sample $\boldsymbol{X}_l$ in the first step (all elements in $\boldsymbol{Y}_l$ are the "background" label) running the EM algorithm on a GMM. The second step consists in identifying the signal subset of observations, that are assumed to be concentrated somewhere, among the more abundant background observations in $\boldsymbol{X}_u$; for that, the EM algorithm is run on a model where the background mixture from the first step is kept fixed (up to a global constant) and another GMM is fit to the signal distribution on top of the fixed background. We will come back to this in more detail in Chapter 6.

5.4 Deep Learning

The term *Deep Learning* refers to a set of techniques that are able to learn different levels of abstract representations of the data. Traditional applications of ML methods require *feature engineering*, i.e. the intervention of a domain expert to devise variables (features) that are sensible for the problem. Such engineered features, that are then fed to the ML algorithm, are calculated from the *raw* data (e.g. the output of a measurement instrument) and often have an interpretation in the domain. In this scenario, the human expert is creating an abstract representation of the raw data. The key idea of Deep Learning is to let the ML method learn abstract representations of the raw data on its own. A famous review of Deep Learning and its applications can be found in ref. [75].

Deep Learning saw its birth with the emergence of complex architectures in NNs, where many neuron layers are used and high levels of representation abstraction are achieved. To give a sense of such complexity, deep NNs can have hundreds of millions of parameters and a similar amount of training examples. This contrasts with the idea of *shallow* learning, e.g. a simple MLP such as the one in Figure 5.2.

The pioneering work in ref. [74] uses Deep Learning in the context of exotic particle searches at the LHC. This study showed the capability of improving the classification of signal and background events in two benchmark scenarios (new heavy Higgs bosons and SUSY particles) with the use of deep NNs and raw data, with respect to more traditional shallow techniques that use engineered high-level features.

The advent of big data, new technologies like graphical processing units (GPUs), as well as some theoretical breakthroughs, set fertile soil for Deep Learning. During the last decade, techniques that use Deep Learning have introduced advancements in various fields such as computer vision (e.g. [108]), natural language processing [109], drug discovery [110], among others, outperforming other methods and reaching super human performance in many cases.

Below we briefly describe three of the most popular Deep Learning methods used in the market.

Convolutional Neural Networks Known also as ConvNets or CNNs, these are deep multilayer perceptrons with a distinctive architecture that is convenient for the processing of data in the form of multiple arrays. *Convolutional* layers perform filtering, or discrete convolution, on the input in order to create a feature map of relevant characteristics for the task (for example, by applying a filter for edge detection on an image). *Pooling* layers combine or summarize the sets of features into single neurons

in the next layer. Pooling and convolutional layers are often used at the first stages of the network (just after the input layer), along with a non-linear activation such as ReLU to reduce the dimensionality of the data, thereby creating a high-level representation of the input. *Dense* layers, follow the same structure of a fully-connected MLP, which gives the final output, usually having a softmax activation in the last layer. Since their conception and initial developments [111, 112], ConvNets have performed well in different computer vision tasks, but notably in the last decade they have received increased attention for their performance on image classification [113]. In Figure 5.3 a representation of a CNN for image classification is displayed with the respective types of layers and activations.

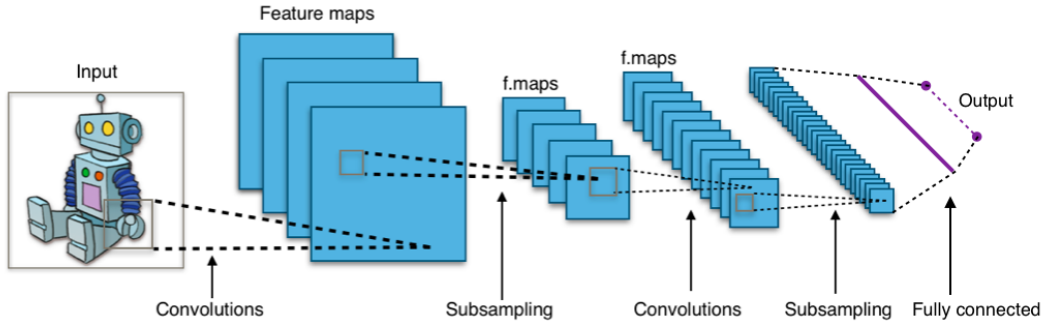


FIGURE 5.3: A Convolutional Neural Network for image classification. Feature learning is performed by the convolutional (subsampling) and pooling layers, while the classification is finally left to a fully connected MLP. Taken from [114].

Recurrent Neural Networks (RNNs) This architecture [115] is conceived to deal with sequential input, by storing on the hidden states information processed from the previous parts of the input. An example of sequential input is language (text or audio), where RNNs can be used, for example, to predict the next word in a sentence or to represent meaning. At a given step t in a sequence, a *recurrent unit* takes as an input the t-th element of the sequence and the previous state of the network, i.e. the output of the $t-1$ unit. The operations performed on the input state and data (i.e. the hyperparameters) are shared across all steps in the RNN. Popular examples of recurrent units are the so-called long-short-term-memory [116] and gated recurrent units [117]. A diagram for this architecture is presented in figure 5.4.

Generative Adversarial Networks (GANs) The core idea of GANs is to have a supervised setup with two deep NNs, one that is a generative model (the simulator) and a discriminative model, that tries to detect whether the other network is generating examples that are similar to those of the training set. For an application to LHC physics GANs were trained with events from the output of a detector (e.g. a calorimeter), the generator can be used to sample events, and were presented as an alternative to other traditional simulators [71]. A recent work [72] demonstrates that it is possible to recover many physical observable distributions from dijet events by training a GAN as an alternative to the more traditional chain of simulators MadGraph[119], Pythia[56] and Delphes[120].

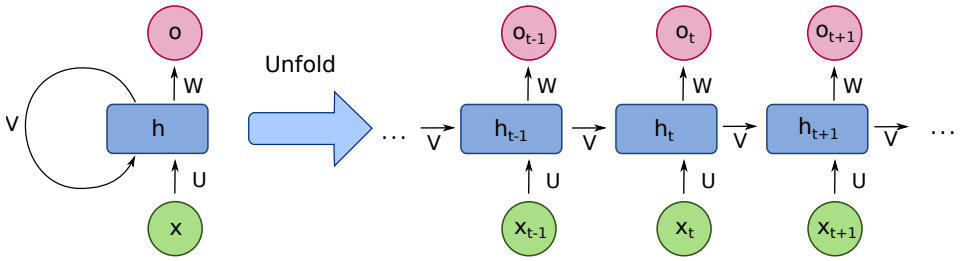


FIGURE 5.4: A folded representation of a Recurrent Neural Network (left) and unfolded equivalent (right). Time steps are denoted by t, (hidden) units by h and outputs by o. Network parameters are represented by U, V, W. Taken from [118].

Deep Learning has proven to be useful in our domain. Tasks such as simulation of events and particle identification have taken profit from different novel Deep Learning methods, that we will discuss shortly. A recent review of this topic with a survey of applications can be found in [121]. It is remarkable that many of the methods used were conceived outside the particle physics community to tackle problems in various, a priori dissimilar domains, as it is the case of ConvNets and RNNs.

Posing the classification of measured objects as an image detection problem has allowed to port ConvNets and their advances in image processing to our field. The parallel is made by creating a "picture" of the object from the different layers of the detector. The identification of jet images calorimeter cells use this idea, and further, some jet classification methods also use jet substructure information to perform the task. Neutrino experiments such as NOvA take advantage of the ConvNet architecture to categorize the interaction of a neutrino with scintillating material [122]. A sequence of track parameters have also been used for training RNNs for the classification of jets [123].

5.5 Methods for General Searches for New Physics in particle colliders

So far, there is no result from LHC that points to the existence of New Physics. Since most of the effort underlying these searches consists in testing a set of hypotheses specified by the New Physics model studied, there is a renewed interest in developing approaches that reduce the number of assumptions on New Physics. Such approaches are called model-independent and rely more heavily on the experimental data available and Monte Carlo simulations of the SM. Further, under the assumption that there is no preferential experimental signature for New Physics to appear, methods have been designed for exploring signatures generically. Some effort in examining a large amount of signatures has been devoted in collider experiments [45, 124–136] and are commonly referred to as General Searches (GS).

Historically, the first attempt to make a GS used data collected by the L3 collaboration at the LEP collider about 20 years ago [137]. This first study used a global comparison between data and Monte Carlo simulations and was performed on 280

exclusive signatures[3]; the number of events and a few kinematic distributions were visually inspected for the most discrepant signatures. More systematic efforts were put in place afterward at the CDF and D0 experiments at the Tevatron and at the H1 experiment at HERA, which were then continued by the next generation of collider experiments at the LHC.

There have been two seminal works that have introduced methods to be used in General Searches; respectively, they are the SLEUTH algorithm [124], first used by the D0 collaboration, and the one used by the H1 collaboration [130].

5.5.1 The Sleuth Algorithm

This algorithm, initially known as Sherlock, was first defined and applied in the study in ref. [124], and used in multiple signatures containing a muon and an electron in the final state. Some modification or extension of it was then employed in future GS with data collected by the same experiment [125–127], and by its competing collaboration at the Tevatron, the CDF experiment [128, 129].

The Sleuth algorithm uses a two-to-four dimensional feature space according to the particle content in the final state of the signature. The motivation for the variables chosen, the selection criteria on the objects and further experimental details can be found in the references above. We proceed to describe the more statistical aspects of the method.

The algorithm starts by defining a set of regions R in the feature space, for any chosen subset of data points $N = 1, ..., N_{\text{data}}$. A transformation is applied to the data into a unit hypercube, in a way in which the expected sample of b background events (provided) is equivalent to the volume of the region. A partition of the feature space from the transformed data points is created via a Voronoi tessellation; this is, for each of the data points a cell is defined as all the points in the feature space that are closer to that data point than to other data points [138]. (This is analogous to the labeling of points after running K-means, using the data points as centroids.) As an illustration, figure 5.5 presents a Voronoi tessellation in the plane.

Regions R are thus defined by a set of contiguous Voronoi cells, that identifies $2^{N_{\text{data}}}$ different regions. This number of regions is reduced by the application of several plausibility criteria.

The number of expected background events $\hat{b}_R$ is then estimated from MC simulations for each region R (i.e. the volume of the region). Further, the weighted probabilities p_N^R that the background estimate for R can fluctuate to at least N is computed; the R that minimizes p_N^R is calculated for each N, and these minimum probabilities are deemed p_N.

The comparison among regions to find the most interesting one is carried out using an ensemble of pseudo experiments, also known as *toy* experiments. The fraction P_N of toys in which the minimum p_N(toys) is smaller than p_N(data) is calculated, and from all P_N define P as the one that is minimal with respect to N. Finally, the fraction $\mathcal{P}$ is defined as the fraction of toys in which P(toys) is smaller than P(data).

Here $\mathcal{P}$ represents the quantity that will tell whether a region is interesting and to which extent. The criterion to signal a signature as interesting in the work in [124] is $\mathcal{P} \lesssim 0.01$ The fact that Sleuth uses a d-dimensional space (where $d = 2, 3$ or 4) leads to high trial factors, i.e an increase in the look-elsewhere effect[4], due to the number of potential regions to be explored, even after applying the plausibility criteria. This reduces the discovery potential of the algorithm with respect to other methods that

[3] As a general rule, only exclusive signatures are used in GS analyses.

[4] Discussed e.g. in the context of HEP in [140].

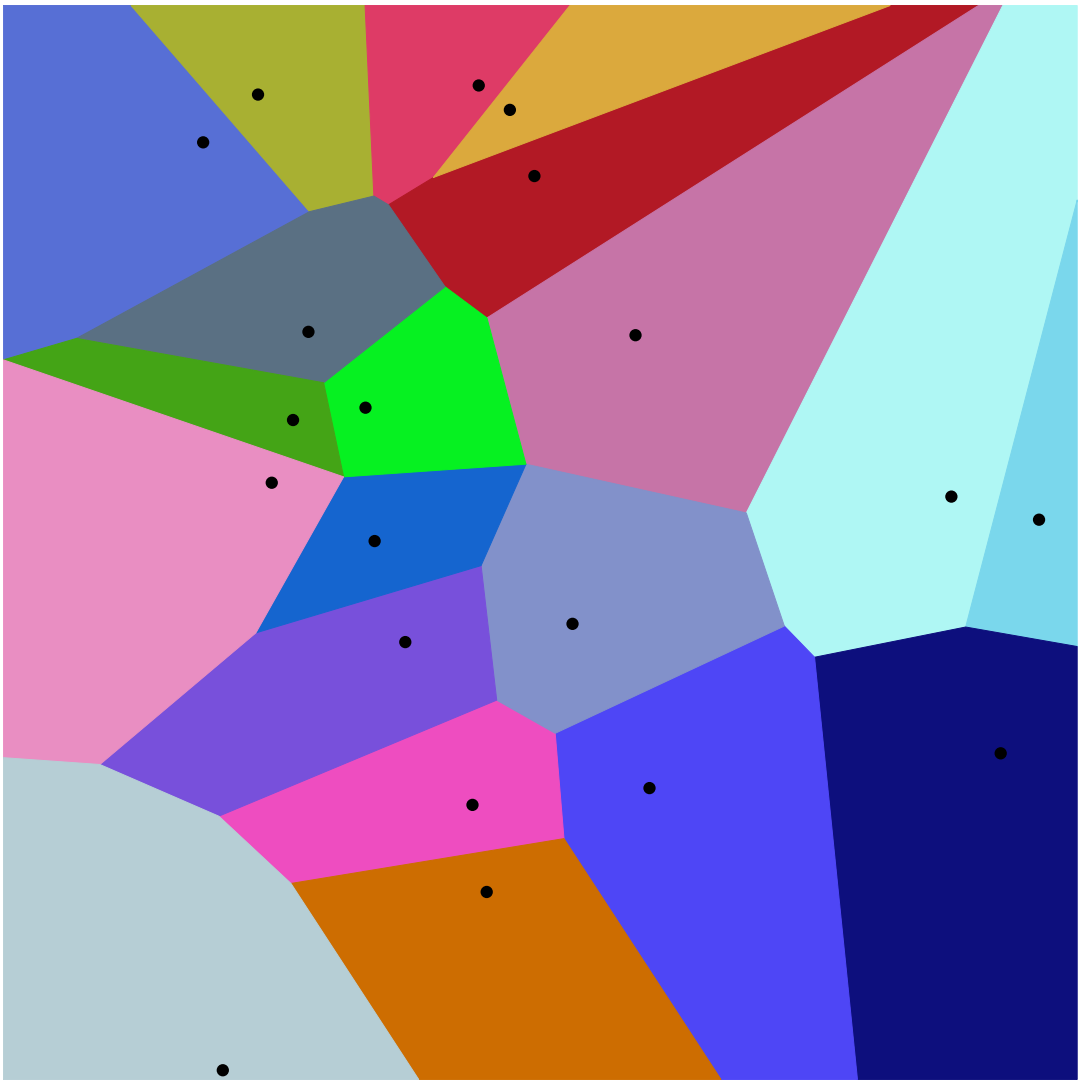


FIGURE 5.5: Voronoi tessellation in 2D using Euclidean distance. Colored regions encompass voronoi cells around black points. Taken from [139].

use one-dimensional approaches on each variable, as we will se below, at the cost of potentially losing information in such reduced dimensionality.

5.5.2 The H1 Algorithm

The algorithm introduced by the H1 collaboration [130] at the HERA collider uses an alternative approach for the same task of analyzing multiple signatures in General Searches. This method (with a few improvements) has been further used by a subsequent analysis at the same experiment [131], and by the CMS [135, 136] and ATLAS [45, 132–134] collaborations. We present the version of the method from ref. [45].

The algorithm takes as an input two binned distributions for one variable: one for observed events and another one for the background events from simulations. A subset of the most discrepant contiguous bins (known as windows) for the two

distributions is chosen, where the discrepancy is measured with the following estimator:

$$p_0 = 2 \cdot \min\left[P(n \leq N), P(n \geq N)\right] , \tag{5.24}$$

$$P(n \leq N) = \int_0^\infty \mathrm{d}x\, \mathcal{N}(x; b, \delta b) \cdot \sum_{n=0}^{N} \frac{e^{-x} x^n}{n!} + \int_{-\infty}^0 \mathrm{d}x\, \mathcal{N}(x; b, \delta b) , \tag{5.25}$$

$$P(n \geq N) = \int_0^\infty \mathrm{d}x\, \mathcal{N}(x; b, \delta b) \cdot \sum_{n=N}^{\infty} \frac{e^{-x} x^n}{n!} , \tag{5.26}$$

where n is the independent variable of the Poisson distribution, and N, b and δb, respectively, the number of observed events, the number of expected background events and their error, within the window. The first integral in eq. (5.25) is a convolution of a Gaussian distribution centered at the number of background events with a width equal to the background error, with a Poisson distribution; they take into account the statistical and systematic components of the uncertainties. (The second integral of the same equation accounts for the observation of no events given a negative number of expected background events from variations of the systematic uncertainties.) Respectively, eqs. (5.25) and (5.26), are the probability of observing no more than N (events in data), and the probability of observing at least N. The algorithm scans all possible windows in the spectrum and chooses the smallest p_0 found.

This procedure is repeated many times with pseudo experiments (toys), in order to get a distribution of smallest p_0-values that is then compared to the value obtained with data. The comparison is done by studying the fraction of toys that have their smallest p_0-values below a certain threshold, which gives the probability of having observed that deviation by chance. This procedure is then run across all signatures (686) and the cumulative distributions of fractions of pseudo experiments with at least m signatures with smallest p_0-values below the threshold gives the final assessment of a sign of new physics; m is taken to be 1, 2 or 3. A figure representing this procedure, with distributions of fractions of pseudo-experiments from the General Search analysis in [45] is presented in figure 5.6.

5.6 Machine Learning Methods for Model-Independent New Physics Searches

During the last couple of years, several ML methods for performing model-independent searches for New Physics have been proposed. The methods, being model-independent, share the goal of performing a search that is as agnostic as possible to the underlying physical process that may be responsible for the New Physics signal. We provide references and a short description of those methods below.

In the method in [141], Variational Autoencoders (VAEs) are used to detect outlier events that may correspond to New Physics. Autoeconders are a NN architecture that can be understood as two operators: the encoder, that compresses the input data into a lower-dimensional space via a feed-forward NN with a decreasing number of neurons on each layer, and the decoder, that is another NN with has the same amount of layers and an increasing amount of neurons, which are the same numbers as in the encoder but in reverse order. Thus, autoencoders have the same number of input and output neurons, and the network is trained such that the reconstruction error (between input and output samples) is minimized. This architecture has been

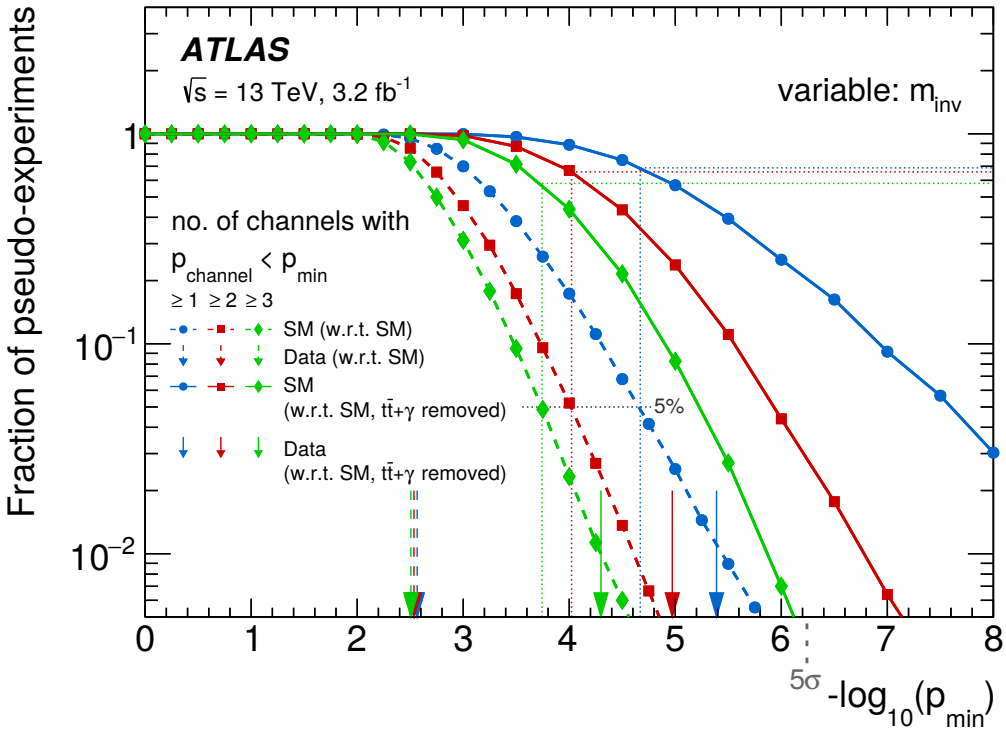


FIGURE 5.6: Distribution of pseudo-experiment fraction that have at least $m = 1$(blue), 2(red), or 3(green) channels below a certain p-value threshold (horizontal axis) for discrepancies found in the invariant mass spectra. Results are given for both the toys for SM expectation and tested against the nominal expectation (dashed) and for those tested against the modified hypothesis ('SM, $t\bar{t}\gamma$ removed') expectation in which that SM process is removed (solid). Dashed arrows are the results for the SM hypothesis and solid arrows the results for the modified hypothesis. Taken from the General Search performed in [45].

used for non-linear dimensionality reduction (the output of the trained encoder) and for anomaly detection, by using the reconstruction error of an autoencoder trained only with non-anomalous data. Although similar in architecture, VAEs are conceptually richer than plain autoencoders, as they allow to learn a representation of the data in a latent space, after the data is encoded; that representation is described by a set of parameters of a (usually Gaussian) distribution and the loss function is extended to have a Kullback-Leibler (KL) divergence between the latent and another distribution. The KL divergence is a measure of dissimilarity between two distributions p and q, defined as:

$$D_{\mathrm{KL}}(p\|q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx\,, \tag{5.27}$$

In Figure 5.7 we present a diagram of a VAE. In ref. [141], VAEs are trained on pure SM background samples using 21 features and tested on a number of benchmark beyond-the-SM simulated scenarios.

In the method proposed in ref. [142], a test statistic is constructed to compare two samples, the first of them corresponding to pure Standard Model background

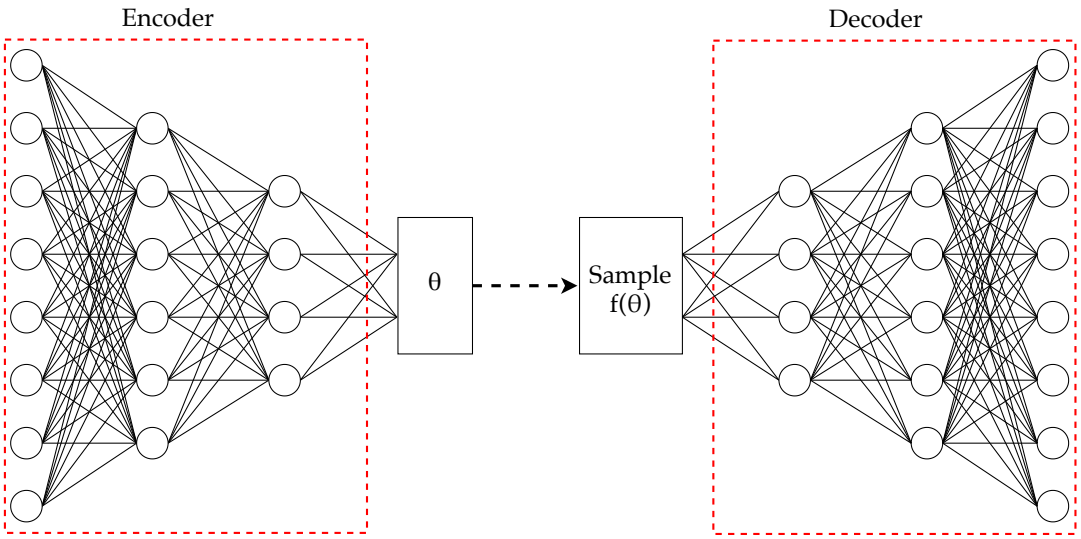


FIGURE 5.7: A diagram of a Variational Autoencoder. The leftmost and rightmost layers are respectively the input and output. The first three layers correspond to the encoder and the last three to the decoder. The set of parameters in the latent space learnt by the encoder are denoted θ from which the sampling is performed. The goal is that the output is able to reconstruct the input.

processes and a second one that can contain signs of New Physics, but is still mostly comprised of background events. A Nearest Neighbors [83] method is applied on events to construct a probability density ratio from two samples. There, the KL divergence is used as a measure of discrepancy among samples; the test proposed is thus unbinned and non-parametric. The work presents results for two examples: a synthetic dataset, comparing samples generated from similar multidimensional Gaussian distributions, and for a simulated LHC collisions taking a Dark Matter model for simulating the signal.

A method that shares some of the ideas we just discussed, is presented in [143]. They exploit the advantages of NNs as universal approximators, and train them using a loss function that is proportional to the Neyman-Pearson test statistic, from which the background-only or non-background-only hypotheses can be tested.

In ref. [144], ML is used to improve a traditional method know as "bump hunting" [145]. The method goes beyond the one-dimensional histogram (used in bump hunting) to consider auxiliary information and finds an optimal (generic) classifier trained on a small sample, where the signal over background proportion ratio is known.

5.7 Conclusions

Machine Learning methods that are relevant to High Energy Physics were presented. An introduction to basic concepts in ML were introduced to motivate the methods and substantiate their relevance. Due to their crucial importance in our field, methods like NNs and BDTs were described to some extent; some recent applications of Deep Learning were surveyed, as well as applications of ML techniques in the context of Model-Independent searches for New Physics. Methods that have been

historically used for model-independent and multi-signature searches were also described. We expect in the near future many more applications at the intersection of ML and HEP, where both communities have profited with the advancement and developments of new techniques, as well as with the appearance of more challenging problems.

In Chapters 6 and 7 we make use of ML methods in HEP. In Chapter 6 we study a method that uses GMMs for modelling background distributions and detecting signals (anomalies) on top of the background, while performing variable selection via a penalized likelihood, in a semi-supervised setup. This method is tested in the context of a model-independent search for New Physics. In Chapter 7 we make use of Gaussian Processes, which are a powerful and flexible tool for making regression; we apply this method in the search for resonances in invariant mass spectra.

Chapter 6

Searching for New Physics Using A Penalized Anomaly Detection Method

In this chapter, we present a novel method for performing the detection of an anomalous set of observations via a semi-supervised setup, with an application in High Energy Physics. The method extends that proposed in ref. [106] for approaching the same problem by, in a single procedure, performing anomaly detection and feature selection, which is achieved by imposing penalty terms in the likelihood.

Note that anomalies are defined here in a *collective* sense. This is, we do not deem individual data points as anomalous, but rather a set of points whose behavior as a whole (i.e. their distribution) deviates from that of the set of non-anomalous observations. The majority of points in the dataset are assumed to be generated by *background* (non-anomalous) processes whereas a minority of the observations, if present, would correspond to a *signal* (anomalous) process. The method presented here aims to tackle the problem of detecting signals that are either faint or located in regions of the feature space that are heavily populated by background observations.

Searches for New Physics can be posed as a collective anomaly detection problem. The background observations correspond to SM processes and the appearance of an anomalous collection of observations can be indicative of a New Physics signal. More specifically, we focus on model-independent searches that impose the fewest amount of physical constraints on the New Physics signal. Thus, the datasets available are that of the pure background sample coming from simulations of the SM, and the one measured by the experiment that can contain some unknown signal. We will see later in this chapter an application of our method in one of such searches.

The semi-supervised approaches that we discuss in this chapter exploit the information on the datasets via a two-step procedure. Firstly, the pure-background dataset is used to fit, i.e. learn, a background model, and secondly, a signal model is added to the background one and a fit is performed by keeping the latter fixed. Finite mixtures of Gaussians are used in both steps for modelling and their parameters are optimized by using the Expectation-Maximization (EM) algorithm [146]; however, this approach suffers from the curse of dimensionality, as it is noted in [106], where it is proposed to perform Principal Component Analysis (PCA)[1] on the input data before applying the method in a lower dimensional representation of the data.

Our method introduces a penalized likelihood to perform both feature selection and collective anomaly detection. In order to achieve the desired task, the penalty term constrains the parameters of the Gaussian mixtures, namely their means and

[1] See Chapter 5, Section 5.3.1 for a definition.

covariance matrices, and a variation of the EM algorithm is crafted for this purpose, as we will describe below.

6.1 Fixed-background model

We proceed to describe the fixed-background model that is proposed in [106]. Two sets of data are used in this setup: the labeled pure-background dataset $\boldsymbol{X}^{\mathrm{l}} = (\boldsymbol{x}_1^{\mathrm{l}}; \cdots; \boldsymbol{x}_N^{\mathrm{l}})$, and the unlabeled dataset $\boldsymbol{X}^{\mathrm{u}} = (\boldsymbol{x}_1^{\mathrm{u}}; \cdots; \boldsymbol{x}_M^{\mathrm{u}})$; if present, the anomaly will appear as a small subset of observations in the unlabeled set.

The realizations $\boldsymbol{x}$ of either the labeled or the unlabeled set live in a D-dimensional feature space, and are assumed to be generated from probability density functions (pdf), respectively, the background pdf $f_B(\cdot)$, and the signal-plus-background pdf $f_{SB}(\cdot)$. If a signal process appears in the unlabeled dataset with pdf $f_S(\cdot)$, then we can write the signal-plus-background model as a mixture:

$$f_{SB}(\boldsymbol{x}) = (1-\lambda) f_B(\boldsymbol{x}) + \lambda f_S(\boldsymbol{x})\,, \tag{6.1}$$

where λ is a mixture coefficient between 0 and 1.

The density functions of the background and signal processes are assumed to be finite mixtures of Gaussian distributions, that are able to accommodate complex shapes [147]:

$$f_B(\boldsymbol{x}|\boldsymbol{\theta}_B) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k)\,, \tag{6.2}$$

$$f_S(\boldsymbol{x}|\boldsymbol{\theta}_S) = \sum_{q=K+1}^{K+Q} \pi_q \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_q, \Sigma_q)\,, \tag{6.3}$$

where K and Q are the number of Gaussian components in the background and signal model, respectively, and the summands are Gaussian distributions (with given means $\boldsymbol{\mu}$'s and covariance matrices Σ's) weighted by non-negative mixing proportions (π's), which are constrained to:

$$\sum_{k=1}^{K} \pi_k = \sum_{q=K+1}^{K+Q} \pi_q = 1\,.$$

The sets of parameters of the background and the signal are then

$$\boldsymbol{\theta}_B = \{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}\,, \qquad \text{and} \qquad \boldsymbol{\theta}_S = \{\pi_q, \boldsymbol{\mu}_q, \Sigma_q\}_{q=K+1}^{K+Q}\,.$$

The log-likelihood of the background parameters in the model in eq. (6.2), for the labeled dataset ($\boldsymbol{x}_i \in \boldsymbol{X}^{\mathrm{l}}$), can be written as

$$\log \mathcal{L}(\boldsymbol{\theta}_B) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \Sigma_k) \right)\,. \tag{6.4}$$

An optimal solution for fitting the model to the data is found by maximizing the likelihood. Numerical methods such as the Expectation-Maximization (EM) algorithm are used in practice, as there is no analytic solution available. This algorithm is guaranteed to increase the (log) likelihood until a local maximum in the space of

all possible $\boldsymbol{\theta}_B$ parameters is reached. In addition, the EM algorithm may be initialized at different points in the parameter space to improve the chance of reaching a global optimum, which is not guaranteed by the algorithm.

At a given iteration, with current estimates of the parameters $\hat{\boldsymbol{\theta}}_B$, the EM algorithm first proceeds to calculate the posterior probability of each observation, indexed by i, to have been generated by each of the components of the mixture, which is known as the expectation (E) step:

$$\tau_{ik} = \frac{\hat{\pi}_k \mathcal{N}(\boldsymbol{x}_i | \hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k)}{f_B(\boldsymbol{x}_i | \hat{\boldsymbol{\theta}}_B)}, \tag{6.5}$$

then the maximization (M) step uses these posteriors to update the values of $\hat{\boldsymbol{\theta}}_B$:

$$\pi_k \leftarrow \frac{1}{N} \sum_{i=1}^{N} \tau_{ik}, \tag{6.6}$$

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{i=1}^{N} \tau_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{N} \tau_{ik}}, \tag{6.7}$$

$$\Sigma_k \leftarrow \frac{\sum_{i=1}^{N} \tau_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} \tau_{ik}}, \tag{6.8}$$

that increase the likelihood. The E and M steps are performed until a local minimum is found. A background model, specified by $\hat{\boldsymbol{\theta}}_B$, is thus obtained by running the EM on the labeled sample.

The key idea of a fixed-background model is to perform a second optimization of the signal-plus-background model in eq. (6.1) on the unlabeled data. The method is said to be semi-supervised because it operates in two steps. In the first step, the parameters $\hat{\boldsymbol{\theta}}_B$ are obtained from the labeled data as we have described above. In the second step, the background parameters are kept fixed, up to a mixing factor $(1 - \hat{\lambda})$ in all background components, thus accommodating a mixture model for the signal process on the unlabeled data. For this purpose, the EM algorithm can be extended to the signal-plus-background model using the unlabeled dataset, and one can rewrite the likelihood accordingly.

The posterior probability of an observation to be generated by the background distribution is

$$\tau_{iB} = \frac{(1 - \hat{\lambda}) f_B(\boldsymbol{x}_i | \hat{\boldsymbol{\theta}}_B)}{f_{SB}(\boldsymbol{x}_i | \hat{\boldsymbol{\theta}}_{SB})}, \tag{6.9}$$

and that for the component q of the signal mixture

$$\tau_{iqS} = \frac{\hat{\lambda} \hat{\pi}_q \mathcal{N}(\boldsymbol{x}_i | \hat{\boldsymbol{\mu}}_q, \hat{\Sigma}_q)}{f_{SB}(\boldsymbol{x}_i | \hat{\boldsymbol{\theta}}_{SB})}, \tag{6.10}$$

where $\hat{\boldsymbol{\theta}}_{SB}$ corresponds to all the estimated parameters for the signal-plus-background model, namely those in $\hat{\boldsymbol{\theta}}_B$, $\hat{\boldsymbol{\theta}}_S$, and the mixture proportion $\hat{\lambda}$.

6.2 Penalized anomaly detection

The fixed-background model is applied in ref. [106] in a space of $D = 2$, that comes from performing PCA on the labeled background sample. This preprocessing is

needed to mitigate on one hand the curse of dimensionality[2], as a mixture of K Gaussians requires the estimation of $K(D+1)(D+2)/2-1$ parameters and, on the other hand, convergence problems of the method in higher-dimensional spaces. Moreover, the use of PCA on the background data reduces the power of the method not only because the projected data carries less information, but also because there is no guarantee that the projected space (computed with background data) will be sensitive to the appearance of the signal. The penalized anomaly detection method we introduce is a variant of the fixed-background model where both the parameter estimation and dimensionality reduction are performed simultaneously.

6.2.1 Penalization of the background

For the moment, we will focus on the case of the background distribution modelling and in subsequent sections extend the idea to the full signal-plus background model.

For a mixture of Gaussians, following the work in ref. [148], the task can achieved by adding a penalty (or regularization) term to the log-likelihood, in the same fashion of Ridge regression or the Lasso[3]. The penalized parameters here are those in the mean of the Gaussian components, and assuming the (D-dimensional) identity covariance matrix for all components. This method is applied to standardized data, i.e. the observations are transformed such that their distribution on each variable have mean equal zero and standard deviation equal to unity. The log-likelihood then takes the form:

$$\log \mathcal{L}_p(\boldsymbol{\theta}_B) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \mathbb{1}) \right) - \gamma p(\boldsymbol{\theta}_B) \,, \tag{6.11}$$

where the penalty term

$$p(\boldsymbol{\theta}_B) = \sum_{k=1}^{K} \sum_{j=1}^{D} |\mu_{kj}| \tag{6.12}$$

and γ is the strength of the penalty, also known as *shrinkage* parameter, as it drives the estimates of the means of the Gaussian components to the zero vector. (Here, μ_{kj} is the value of the k-th Gaussian component mean in the j-th dimension.) In order to perform dimensionality reduction, the EM algorithm is adjusted to the penalized maximum likelihood which allows for the identification of features that are informative, i.e. the ones in which the means of the Gaussian components are near zero are deemed uninformative and discarded, whereas those where the component means are far from zero are retained.

The approach in [149] instead penalizes values on the component means by using the likelihood in eq. (6.11) with

$$p(\boldsymbol{\theta}_B) = \sum_{j=1}^{D} \sqrt{\sum_{k=1}^{K} \mu_{kj}^2} \,, \tag{6.13}$$

where the parameters inside the square root are simultaneously penalized or shrunk across all K components, leading to a performance improvement in identifying uninformative variables with respect to using the penalty in eq. (6.12). This conclusion

[2] Ibid.

[3] See Section 5.2.

was reached in [149] after testing the method in synthetic data in different set-ups (including cases in which noise variables were added), and in gene expression and molecular subtype discovery real datasets. Further work in ref. [150] considered a second penalty term in the covariance matrices, that are there required to be diagonal (and not the identity), where the penalty drives their variances to unity.

The approach we present aims to be a more general way to exploit the information in the model parameters for both modelling the distributions and variable selection. We relax of the requirement in the covariance matrix to be merely positive-definite, and include two penalty terms to be added to the likelihood. One term penalizes the means and mixing proportions, and one the eigenvalues of the covariance matrices of the components, respectively

$$p_1(\boldsymbol{\theta}_B) = \sum_{j=1}^{D} \sqrt{\sum_{k=1}^{K} \pi_k \mu_{kj}^2}\,, \tag{6.14}$$

and

$$p_2(\boldsymbol{\theta}_B) = \sum_{k=1}^{K} \sum_{j=1}^{D} \max(\delta_{kj}, \epsilon_k)\,. \tag{6.15}$$

where δ_{kj} is the j-th largest eigenvalue of Σ_k, and ϵ_k is a small positive value for the k-th component. Thus, the shrinkage of the covariance parameter happens via the second penalty by driving each of the eigenvalues of the covariance matrices towards their respective small value ϵ_k. The average of the subset of the L_k smallest eigenvalues is used to estimate each ϵ_k, where L_k is chosen using sequential tests. The details of the procedure to choose the eigenvalues can be found in [151].

The full penalized likelihood then reads

$$\log \mathcal{L}_p(\boldsymbol{\theta}_B) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \Sigma_k) \right) + \gamma_1 \sum_{j=1}^{D} \sqrt{\sum_{k=1}^{K} \pi_k \mu_{kj}^2} + \gamma_2 \sum_{k=1}^{K} \sum_{j=1}^{D} \max(\delta_{kj}, \epsilon_k)\,, \tag{6.16}$$

where γ_1 and γ_2 are the shrinkage parameters of the penalties. The posteriors and update rules on the respective steps of the EM algorithm then need to be accordingly modified for this penalized likelihood. Details on how variable selection is performed are given in Appendix A.

6.2.2 Penalization of the signal-plus-background model

We now proceed to include the framework we just described in 6.2.1 within the fixed-background model described in Section 6.1. The automatic variable selection is performed taking into the account both the labeled and unlabeled datasets; thereby avoiding the risk of discarding variables that may turn to be relevant for the signal. This implies that both the penalized likelihood for the full signal-plus-background model depends on the estimation of background parameters, and the likelihood of

the background estimation depends on the estimated signal parameters. Both likelihoods then need to be optimized simultaneously. They are explicitly for the background model:

$$\begin{aligned}\log \mathcal{L}_p(\boldsymbol{\theta}_B|\hat{\boldsymbol{\theta}}_S) = &\sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i^{\mathrm{l}}|\boldsymbol{\mu}_k, \Sigma_k)\right) \\ &+ \gamma_1 \sum_{j=1}^{D} \sqrt{\sum_{k=1}^{K} \pi_k \mu_{kj}^2 + \sum_{q=K+1}^{K+Q} \hat{\pi}_q \hat{\mu}_{qj}^2} \\ &+ \gamma_2 \sum_{j=1}^{D} \sum_{k=1}^{K} \max(\delta_{kj}, \epsilon_k)\,,\end{aligned} \tag{6.17}$$

where the $\hat{\boldsymbol{\theta}}_S = \{\hat{\pi}_q, \hat{\boldsymbol{\mu}}_q, \hat{\Sigma}_q\}_{q=K+1}^{K+Q}$ have been estimated using the unlabeled dataset $\boldsymbol{X}^{\mathrm{u}}$; and analogously for the signal(-plus-background) model:

$$\begin{aligned}\log \mathcal{L}_p(\boldsymbol{\theta}_S|\hat{\boldsymbol{\theta}}_B) = &\sum_{i=1}^{M} \log\left((1-\lambda)\sum_{k=1}^{K} \hat{\pi}_k \mathcal{N}(\boldsymbol{x}_i^{\mathrm{u}}|\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k) + \lambda \sum_{q=K+1}^{K+Q} \pi_q \mathcal{N}(\boldsymbol{x}_i^{\mathrm{u}}|\boldsymbol{\mu}_q, \Sigma_q)\right) \\ &+ \gamma_1' \sum_{j=1}^{D} \sqrt{\sum_{k=1}^{K} \hat{\pi}_k \hat{\mu}_{kj}^2 + \sum_{q=K+1}^{K+Q} \pi_q \mu_{qj}^2} \\ &+ \gamma_2' \sum_{j=1}^{D} \sum_{q=K+1}^{K+Q} \max(\delta_{qj}, \epsilon_q)\,,\end{aligned} \tag{6.18}$$

where the $\hat{\boldsymbol{\theta}}_B = \{\hat{\pi}_k, \hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k\}_{k=1}^{K}$ have been estimated using the labeled dataset $\boldsymbol{X}^{\mathrm{l}}$. The optimization is then performed alternating equations (6.18) and (6.17) with the respective modified EM algorithm that maximizes the penalized likelihood following the procedure in appendix A, as the signal and background parameters need to be estimated for updating one another.

The shrinkage parameters (γ) are chosen in the program using the Bayesian Information Criterion (BIC) [152], defined as follows:

$$\mathrm{BIC} = \ln(N)P - 2\ln(\widehat{\mathcal{L}_p}) \tag{6.19}$$

where $\widehat{\mathcal{L}_p}$ is the maximized penalized likelihood, N the number of data points observed[4], and P the number of parameters in the model defined by $\{\boldsymbol{\theta}_B, \boldsymbol{\theta}_S, \gamma_1', \gamma_2', \gamma_1, \gamma_2, \lambda\}$. The BIC provides a tradeoff between the complexity of a model and the likelihood achieved, which is used to choose among models.

6.3 Application to High Energy Physics

For a proof of concept of the method in the context of High Energy Physics, we put in place a simple physics analysis using simulated data. We provide some details on the simulation setup and on the data extracted, to be used in the method. Finally, we present results for the method and a comparison with the fixed background model.

[4] In the case of the labeled dataset this is N. For the unlabeled dataset it should be M.

6.3.1 Data description

In this application, we are interested in the signature containing two jets in the final state (known also as "dijet" final states), denoted as "jj" coming from proton-proton collisions at the LHC; for such signatures, background and hypothetical signal collision events have been generated as described below. A simple analysis selection is then performed on the samples and finally the anomaly detection method is tested.

We generated a set of MC samples using standard simulation software for high-energy collisions. The simulated phenomena correspond to proton-proton collisions at a center-of-mass energy $\sqrt{s}$ = 13 TeV and measured by a simplified ATLAS detector simulation implemented in the DELPHES [120] package. The main source of Standard Model (SM) background in our signature is the production of two jets via QCD processes, namely from the production of a pair of gluons, a pair of quarks, or a gluon and a quark. The simulated signal corresponds to the resonant production of a stop quark decaying into two jets, in the R-parity violating Minimal Supersymmetric Model (RPV-MSSM) [28, 153], using the package in [154]; a motivation for the physics of this R-parity-violating process leading to this and other multi-jet final states at the LHC is given e.g. in [155]. A more in-depth description of the simulations performed for signal and background follows.

Signal

As a benchmark for testing the anomaly detection algorithm, we produced a sample of 5 * 10^5 stop quark signal events. The hard process was simulated using MADGRAPH 5.2.6.5 [119] at Leading Order, where we used the four-flavor scheme and `nn23lo1` [156] to model the proton parton distribution functions. The mass of the resonant stop was fixed to a value of 1000 GeV. All default parameters from MADGRAPH 5 were used except the value of the pseudorapidity, which was restricted to $|\eta| < 2.5$. The resulting events were then ported to PYTHIA 8.240 [56] for showering, decay and hadronization.

Within the RPV-MSSM model used, the production and decay (to two jets) of a resonant stop happens via a single Feynman diagram in a four-flavor scheme, as depicted in figure 6.1.

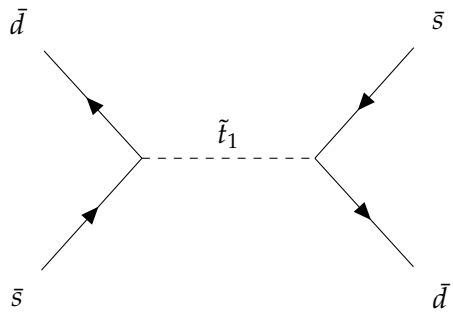


FIGURE 6.1: Feynman diagram for the production of a stop quark decaying into two light quarks, in the RPV-MSSM [28, 153].

The production cross section reported by MADGRAPH 5 is 18.011 $\pm$ 0.003 pb.

Background

The production of 5 * 10^5 dijet events from QCD processes was performed using the same MADGRAPH 5 and PYTHIA 8 versions as the ones for the signal. There

are many ways in which QCD interactions can lead to a pair of jets in the hard process, namely all possible diagrams that contain two gluons, a quark and a gluon or two quarks in the final state. In a four-flavor scheme at tree level, there are 65 such processes. The production cross section corresponding to the QCD background reported by MADGRAPH 5 is: 3.382 ± 0.001 mb.

Detector simulation

Both signal and background event samples have been passed through a fast detector simulation using the DELPHES 3.4.1 software. We have kept all default parameters except the jet cone parameter $\Delta R = 0.4$, and used the ATLAS detector card that comes with the software distribution.

6.3.2 Event selection and variables used

The signal and background simulated samples are then analyzed and an event selection is applied. A set of requirements are imposed on the object properties and event variables, inspired by realistic experimental analyses, in order to e.g. mitigate detector effects, simulate trigger selection. Furthermore, several features (variables) are extracted and calculated. The features of the events that pass the selection comprise the input of the anomaly detection algorithm. Typical values of object selection are already included in the DELPHES ATLAS detector card; at this level, the only additional requirement we impose is that the event contains only two jets and each of them has a transverse momentum of 20 GeV or more.

Given that we are performing a model-independent search, the selection requirements are not optimized for any particular signal, even if we could, in principle, devise such a procedure for the stop production described above.

The variables extracted and calculated for each event are the following:

- The energy of each jet E_1, E_2.
- The three momenta of each jet ($i = 1, 2$): $(p_{T,i}, \eta_i, \phi_i)$.
- Reconstructed invariant mass of the dijet system: $M_{\text{inv}}(j_1, j_2)$.
- Missing transverse momentum: E_T^{miss}.
- Angular distance of the two jets in the $\eta - \phi$ plane: $\Delta R(j_1, j_2)$.
- Sphericity as defined in [157].
- Centrality defined as $C = \frac{E_{T,i}}{\sum_i E_i}$ where $i = 1, 2$ is the jet index.

These add to 13 variables for each event. An example of a normalized distribution, the invariant mass, for the background and signal described above can be found in figure 6.2 (left).

6.3.3 Method performance

Preprocessing and application of the method

As it is manifest in figure 6.2 (left), the distribution of events in several of the variables can become heavily skewed. Even if, in principle, finite mixtures of Gaussians lead to models with great flexibility, skewed data may require a high number of

components to be accommodated. Therefore a preliminary processing of the data is performed with two transformations: a Tukey ladder of powers transformation [158] is used, and then all the data is standardized with respect to the background dataset. The Tukey ladder of powers consists in transforming the data per variable x, by applying a transformation of the type

$$f(x) = \begin{cases} x^\alpha\,, & \text{for } \alpha > 0 \\ -x^\alpha\,, & \text{for } \alpha < 0 \\ \ln(x)\,, & \text{for } \alpha = 0 \end{cases} \tag{6.20}$$

that effectively reduces skewness and makes the distribution of values more Gaussian-like, as it is shown in 6.2 (right).

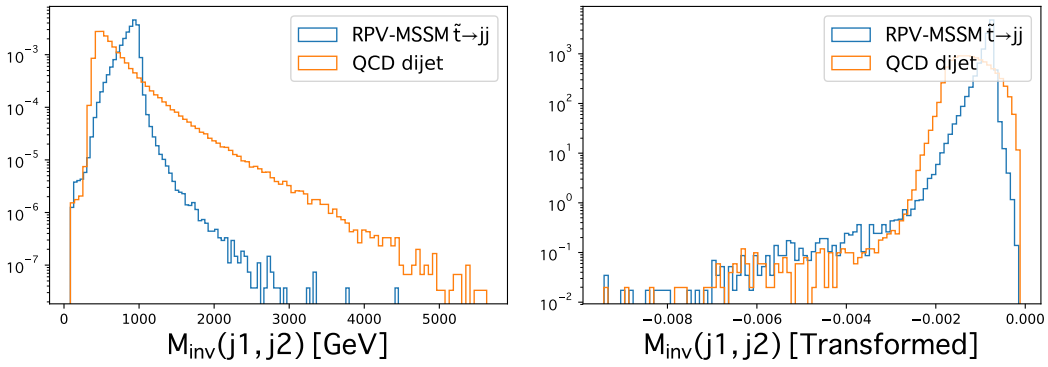


FIGURE 6.2: Left: Normalized distributions of signal and dijet background for the invariant mass. Right: power transformation of the invariant mass distributions with a coefficient $\alpha = -1.05$.

For obtaining the Tukey transformation, for each variable, a range of possible α values is given [5] and the transformation chosen is such that it maximizes the Shapiro-Wilk test statistic [160]. For the variables in this dataset, the α values are presented in table 6.1.

Variable	E_1	η_1	ϕ_1	p_{T1}
α	-0.65	0.6	0.775	-2.1

Variable	E_2	η_2	ϕ_2	p_{T2}
α	-0.6	0.55	0.825	-0.475

Variable	$\Delta R(j_1, j_2)$	$M_{\text{inv}}(j_1, j_2)$	E_T^{miss}	Sphericity	Centrality
α	-0.05	-1.05	0.125	0.25	0.5

TABLE 6.1: Powers obtained per variable from a Tukey ladder of powers transformation.

Even after this preprocessing, the variables $\eta_{1,2}$ were removed, because they still show complex non-elliptical patterns. This leaves a total of 11 input variables to the penalized anomaly detection method. Plots for the standardized distributions and the corresponding Tukey-transformed versions can be found in appendix B.

[5] In the implementation that we use (transformTukey) [159], the default range is from -10 to 10 in steps of 0.025, that we keep for our problem.

Results

We applied the method in 50 data subsets taken from the background and signal simulations, using different proportions of injected signal events λ. The number of events was chosen to be $M = N = 4000$ with proportion values 5, 10, 15, and 20 percent[6]. The method was constrained to use one Gaussian component for the signal, which can already extract signal information from the data if it is present.

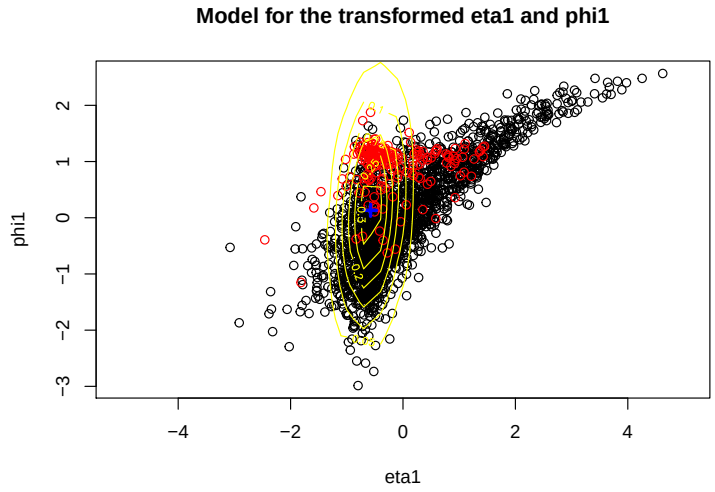


FIGURE 6.3: Signal (red circles) and background (black circles) events in two variables η_1 and ϕ_1. The signal component is presented with mean at the blue cross and yellow contour curves.

A representation of the method for two variables can be observed in figure 6.3. Given that signal events may lie in heavily-populated background regions, as it is the case in our dataset, the signal posterior probabilities of the modified EM algorithm[7] will yield low values; this can be addressed by using the area under the Receiver Operating Characteristic curve (AUC) [161]. The points on this curve are calculated for different threshold values of the EM posterior (from 0 to 1), in eq. (6.10), by using the true positive rate (Sensitivity) and false positive rate (1− Specificity) when classifying the observations as signal or background.

An example of Receiver Operating Characteristic curve is presented in figure 6.4, where the injected signal proportion as 10% and the AUC is 0.768. In that case two of the variables were identified as uninformative, p_{T2} and E_T^{miss}.

Results are shown in table 6.2 for both methods, the penalized anomaly detection (PAD), defined in Section 6.2, and the fixed background model (FBM), defined in Section 6.1. We present the estimated mixture proportion, the ARI and the AUC with respective errors, taken from repeating the model fit with 50 signal samples. The λ values are underestimated in general for both methods, but the penalized anomaly detection performs better in terms of AUC, improving as λ grows. The spurious detection test (no signal present) lead to an estimated mixing parameter $\hat{\lambda}$ of 0.103 (0.027); a value of λ =1.3 (mean value plus error) can be taken as an indication of the capability of the method to detect faint signals.

[6]The sample sizes of the labeled and unlabeled datasets are respectively M and N, as it was stated at the beginning of this chapter.

[7]As defined in appendix A, or in the case of the Fixed Background Model, eq. (6.10).

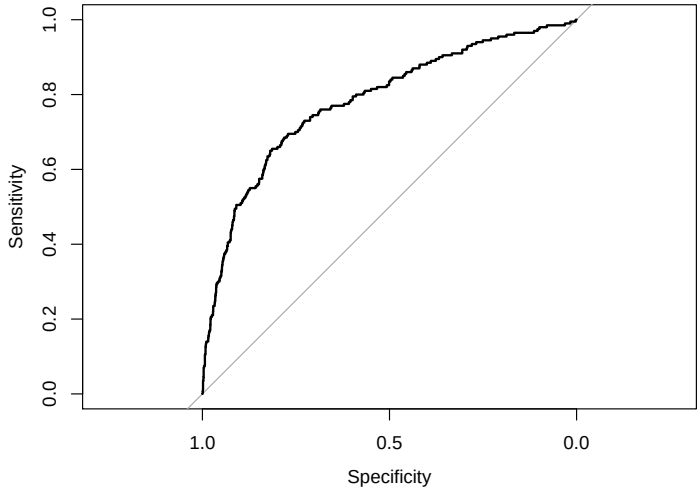
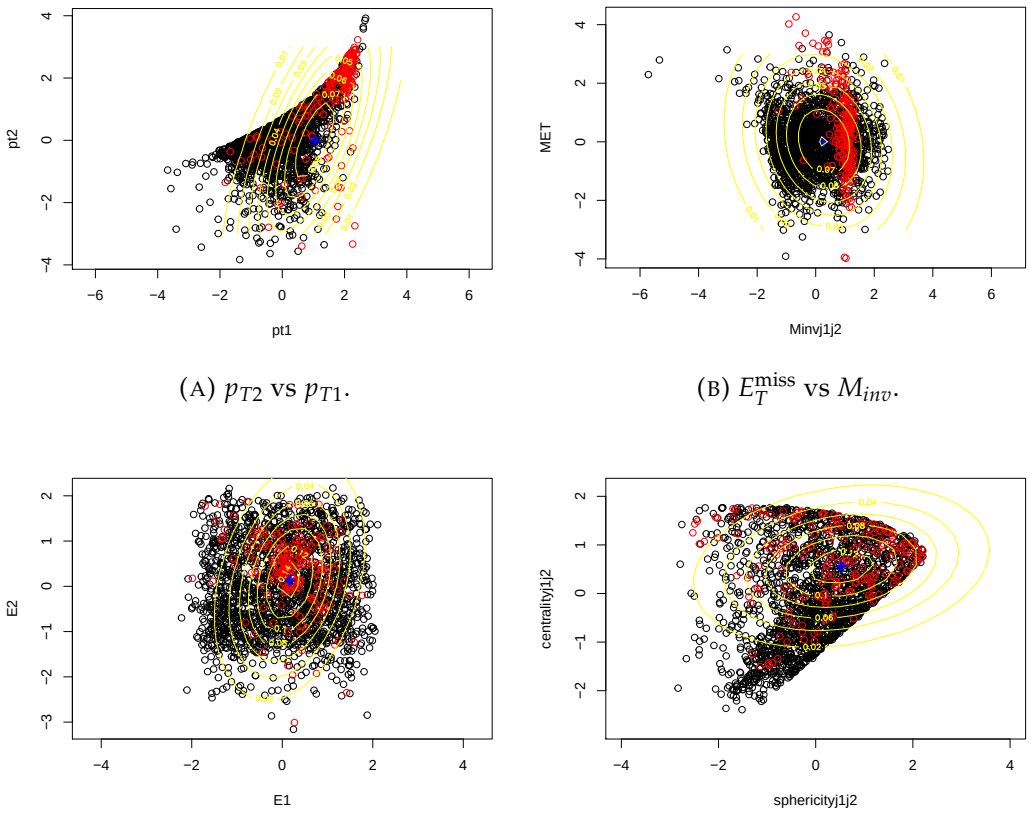


FIGURE 6.4: Receiver Operating Characteristic curve for a signal with mixture parameter $\lambda = 0.1$. Sensitivity (True Positive Rate) versus Specificity (1 − False positive Rate) values are presented in the solid black line. A dotted diagonal (random choice) is presented for reference.

6.4 Conclusion and outlook

In this chapter we presented a method that uses a penalized likelihood to perform both variable selection and anomaly detection, in a semi-supervised setup. We applied that method in the case of dijet events, where both background and signal samples were produced using a chain of software packages for MC simulations that recreate the processes taking place in collision events and their detection. The method is able to achieve its task, performing in general better than an alternative (the fixed-background model) as the AUC shows. However, the penalized anomaly detection method underestimates the signal proportion, in particular for the higher values of λ (0.15, 0.20). Also, spurious anomalies were detected when testing the method in the absence of signal; the extracted mixture value in that case was $\lambda = 0.103 \pm 0.027$.

The studies we presented are a proof of concept for the penalized anomaly detection method, but there are several promising directions for improvement. Future studies on this method may include the exploration of other penalties that may be more effective in performing shrinkage, beyond what we have reviewed in this chapter, and the use of other kinds of finite mixture models [147] that could alleviate the problem of pre-selecting variables for the method. The method could also see some improvements in the automatization of the running of the software and handling of the samples, that we leave for future work.



(A) p_{T2} vs p_{T1}.

(B) E_T^{miss} vs M_{inv}.

(C) E_2 vs E_1.

(D) Centrality vs Sphericity.

FIGURE 6.5: Scatter plots for the background (black circles) and signal (red circles) from the unlabeled dataset in pairs of transformed variables. The signal fit is overlayed with mean (blue cross) and curve levels (solid yellow). Variables p_{T2} and E_T^{miss} (vertical axes in the figures at the top) are uninformative and have mean equal zero.

TABLE 6.2: Summary of the anomaly detection results performed by the penalized anomaly detection (PAD), in Section 6.2, and the fixed background model (FBM), in Section 6.1, for datasets with different signal proportions λ. For each scenario, 50 datasets are generated to obtain a mean result with the respective standard deviations presented in brackets.

Method	λ	Average estimate $\hat{\lambda}$	Average AUC
PAD	0 (spurious)	0.103(0.027)	-
PAD	0.05	0.040(0.012)	0.725(0.109)
PAD	0.10	0.057(0.013)	0.818(0.078)
PAD	0.15	0.086(0.006)	0.876(0.017)
PAD	0.20	0.112(0.006)	0.882(0.012)
FBM	0 (spurious)	0.123(0.031)	-
FBM	0.05	0.025(0.009)	0.708(0.118)
FBM	0.10	0.046(0.008)	0.764(0.078)
FBM	0.15	0.070(0.006)	0.771(0.073)
FBM	0.20	0.096(0.012)	0.780(0.054)

Chapter 7

Model Independent Search For Generic Resonant Signals Using Gaussian Processes

In the present chapter we make use of Gaussian processes (GPs) for regression. This is a flexible method that allows for a smooth modeling of some physical spectra (e.g. invariant mass) as measured by the ATLAS experiment at the LHC. We begin by providing a definition and a few relevant results in the theoretical formulation of GPs. Then, we describe a previous analysis where GPs were used in the dijet mass spectrum [162] for modelling signal and background distributions in a two-step procedure for the dijet spectrum, and provide details on our GP method, which is a modification of that procedure. We compare the performance of the GP method with a simpler approach based on parametric fits in the dijet spectrum. Further, we propose a three-step procedure for modelling the $t\bar{t}$ invariant mass spectrum and extracting potential resonances.

7.1 Bayesian Learning and Gaussian Processes

Gaussian processes (GPs) are a Bayesian inference method in a function space, defined as "a collection of random variables, a finite collection of which have a joint Gaussian distribution" [84]. In the GP setup we have knowledge of the desired output (responses), and thus GPs are a supervised method. Given a prior distribution over the parameters of a set of functions in the function space and using the likelihood of the observations, we can obtain a posterior distribution over such parameters through Bayes' rule.

Let f be the function that is regressed and x and x' arbitrary points in the input space $\mathcal{X}$, the prior on the regression can be noted as

$$f(x) \sim \mathcal{GP}(\mu(x), \Sigma(x, x'))\,, \tag{7.1}$$

where $\mathcal{GP}$ is the infinite-dimensional function space associated with the joint Gaussian distribution, from which f is sampled. Thus, there are two functions that are defined to specify a GP: the *mean* and the *covariance* or *kernel*, respectively,

$$\mu(x) = \mathbb{E}[f(x)]\,, \tag{7.2}$$
$$\Sigma(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]\,. \tag{7.3}$$

The kernel and mean functions above, as their names suggest, dictate the mean value and the covariance of the GP distribution for points in the input space. In practice,

the kernel is often specified via a function with a set of hyperparameters, instead of obtaining the covariance directly from data samples.

For a finite set of points, the prior and posterior are joint Gaussian distributions with as many dimensions as observations there are, and there is a formalism that allows to predict new output values for arbitrary inputs. In ref. [162], GPs are used to model the Poisson process corresponding to a binned distribution of events, i.e. the spectrum. Let the bin centers be denoted by the vector $\boldsymbol{x} = (x_1, \ldots, x_N)$, the corresponding observed responses $\boldsymbol{y} = (y_1, \ldots, y_N)$, the function f is then averaged within the corresponding bin to produce a set of expected counts $\bar{\boldsymbol{f}}(\boldsymbol{x}) = (\bar{f}(x_1), \ldots, \bar{f}(x_N))$. The spectrum is approximated via the product of two multidimensional Gaussians by the probability model:

$$p(\boldsymbol{y}(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{y}|\bar{\boldsymbol{f}}(\boldsymbol{x}), \boldsymbol{\sigma^2}(\boldsymbol{x}))\mathcal{N}(\bar{\boldsymbol{f}}(\boldsymbol{x})|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{7.4}$$

where σ are the uncertainties in the values of $\bar{\boldsymbol{f}}$, that is an approximation of the Poisson noise via a Gaussian. The second factor is an N multivariate Gaussian distribution. Note that we use the vector $\boldsymbol{\mu} = (\mu(x_1), \ldots, \mu(x_N))$ and the matrix $\boldsymbol{\Sigma}$ with $\Sigma_{ij} = \Sigma(x_i, x_j)$, constructed from the mean and covariance functions respectively.

After some standard algebraic manipulation, known as completing the square, it is possible to obtain explicit expressions to infer the response of the function f at a new arbitrary input value. The posteriors of the mean and covariance of the GP corresponding to a new set of data points $\boldsymbol{x_\star}$, given $\boldsymbol{x}$ and $\boldsymbol{y}$, can then be calculated by:

$$\mathbf{mean}(\boldsymbol{f_\star}) = \boldsymbol{\mu_\star} + \boldsymbol{\Sigma_\star}[\boldsymbol{\Sigma} + \boldsymbol{\sigma^2}(\boldsymbol{x})\mathbb{1}]^{-1}(\boldsymbol{y} - \boldsymbol{\mu}), \tag{7.5}$$

$$\mathbf{cov}(\boldsymbol{f_\star}) = \boldsymbol{\Sigma_{\star\star}} - \boldsymbol{\Sigma_\star}[\boldsymbol{\Sigma} + \boldsymbol{\sigma^2}(\boldsymbol{x})\mathbb{1}]^{-1}\boldsymbol{\Sigma_\star}, \tag{7.6}$$

where $\boldsymbol{f_\star} = \boldsymbol{f}(\boldsymbol{x_\star})$, $\boldsymbol{\Sigma_\star} = \boldsymbol{\Sigma}(\boldsymbol{x}, \boldsymbol{x_\star})$, and $\boldsymbol{\Sigma_{\star\star}} = \boldsymbol{\Sigma}(\boldsymbol{x_\star}, \boldsymbol{x_\star})$; $\boldsymbol{\mu_\star}$ is the mean prior corresponding to $\boldsymbol{x_\star}$. Note that the dimension of the GP multivariate gaussian is extended by the dimension of $\boldsymbol{x_\star}$.

It is possible to initialize the prior mean $\boldsymbol{\mu}$ with a function from our domain knowledge but setting it equal to zero is common practice and does not necessarily pose a limitation when using GPs. The most important ingredient to be specified is then the kernel, where there exist several common choices in the literature (e.g. the exponential squared or other radial kernels) [84, 163], or a new kernel could be crafted for a particular application[1]. The hyperparameters in the kernel need to be adjusted as well; this is usually done finding the maximum log marginal likelihood of the GP:

$$\log \mathcal{L} = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - (\mathbf{y} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) - \frac{N}{2}\log 2\pi\,. \tag{7.7}$$

The standard algorithm to obtain the value $\boldsymbol{f_\star}$, its variance and the likelihood involves solving triangular systems and matrix inversion, as it is implied in the expression for the kernel, eq. (7.6). The details of the algorithm and some optimizations can be consulted in reference [84].

Kernels that depend only on the (Euclidean) distance between the inputs, $|\boldsymbol{x} - \boldsymbol{x}'|$, are referred to as stationary kernels. A commonly-used stationary kernel is the radial

[1] A set of example kernels, how to compose them, and an explanation on how they can express the structure can be found in Chapter 2 of Duvenaud's thesis in ref. [163].

basis function

$$\Sigma(x, x') = A \exp\left(-\frac{|x - x'|^2}{2l^2}\right), \tag{7.8}$$

that will become relevant in subsequent discussions; A represents the strength of the correlation and l a typical correlation length between points.

7.2 Modeling backgrounds and signals with Gaussian Processes

We proceed to present here some aspects of the work developed in [162], where a method to model backgrounds and generic signals with GPs is proposed for the search for new physics in the dijet invariant mass spectrum. There, the GP approach is presented in contrast of a more traditional framework, referred to as the parametric method. For a few decades, the parametric method has been used to model the background in the spectrum, which is heavily dominated by processes from strong interactions. For the studies presented in this Chapter, we will use simulated ATLAS dijet events.

The parametric approach is given by a formula of the form

$$f(x|\theta) = \theta_0(1 - x)^{\theta_1} x^{\theta_2} x^{\theta_3 \log x}, \tag{7.9}$$

where x is the invariant mass divided by the center of mass energy of the collisions, $x = \frac{m_{jj}}{\sqrt{s}}$. This kind of ad-hoc functions have been used and evolved empirically to provide a good fit for the observed spectrum, e.g. as it is shown in a recent search for new phenomena in that spectrum [164] using the ATLAS detector, and does not make use of information from the underlying physical process. We provide further details on the parametric approach below in section 7.5.2.

The prescription of the GPs in [162] is given by using a mean and kernel functions:

$$\mu(x) = f(x|\theta) \tag{7.10}$$

$$\Sigma_B(x, x') = A \exp\left(\frac{d - (x + x')}{2a}\right) \sqrt{\frac{2l(x)l(x')}{l(x)^2 + l(x')^2}} \exp\left(\frac{-(x - x')^2}{l(x)^2 + l(x')^2}\right). \tag{7.11}$$

The mean function μ here is given by the fitted parametric background model, with its parameters fixed. The proposed kernel Σ_B is non-stationary and constructed following several physical considerations. We can observe that the last factor is a modified squared exponential, where $l(x) = bx + c$ amounts for a typical length related to the dijet mass resolution that has a linear relationship with mass; this leads to a total of five hyperparameters A, a, b, c, and d, that will collectively be referred to as θ_B. The product of the last two factors in Σ_B is referred to as Gibbs kernel [84]; whereas the first factor assumes that fluctuations in the correlation follow an exponentially decaying regime for high masses, with a typical length of a and shifting term d. The first factor A is analogous to the ones appearing in kernels presented previously in eq. (7.8). Figure 7.1 presents a correlation plot from the background kernel, where it is possible to observe how the band around the diagonal increases with mass values, as dictated by the Gibbs kernel.

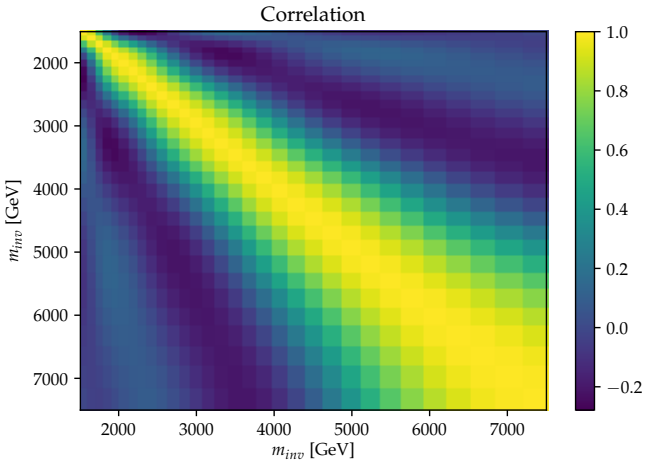


FIGURE 7.1: Correlation from the background kernel, Σ_B in eq. (7.10), after a fit is performed.

The work in [162] presents modeling studies for backgrounds and generic signals. Two tests are performed for background modeling, namely background-only tests to assess the power of the method to model that distribution without assuming any signal, and signal-plus-background tests, where a GP model of the background is fitted on data where there is some signal injected For modeling signals, a two-step approach is presented, where the background GP is fitted on background only simulations, and the data with signal injected is modeled via an addition of kernels:

$$\Sigma_{SB} = \Sigma_B + \Sigma_S \,, \tag{7.12}$$

with

$$\Sigma_S(x, x') = A_S \ \exp\left(-\frac{1}{2}\left(x - x'\right)^2 / l^2\right) \exp\left(-\frac{1}{2}\left((x-m)^2 + \left(x'-m\right)^2\right) / t^2\right) . \tag{7.13}$$

The first exponential of this kernel with the prefactor A_S (signal correlation strength) is equivalent to eq. (7.8) and the other exponential localizes the signal in a mass window of width t and mean m. The parameters of Σ_S, namely A_S, l, t, and m are collectively referred to as θ_S. For illustration, figure in 7.2 presents a plot of the signal kernel, with hyperparameters chosen after fitting an injected Gaussian signal at 3 TeV, that we will explain below.

7.3 Methods for searching generic resonances

We take as a starting point what has been described in the previous section and proceed to describe our extensions, as well as provide further detail for the parametric fit, that is used as a benchmark.

In general, we are interested in probing how sensitive is a method in detecting the presence of new physics in a spectrum. Typically, such presence becomes manifest in the invariant mass spectrum as an excess of events within a (narrow) mass interval, that can be generically accounted for as a signal sampled from a Gaussian distribution (or "bump") on top of the background distribution coming from SM

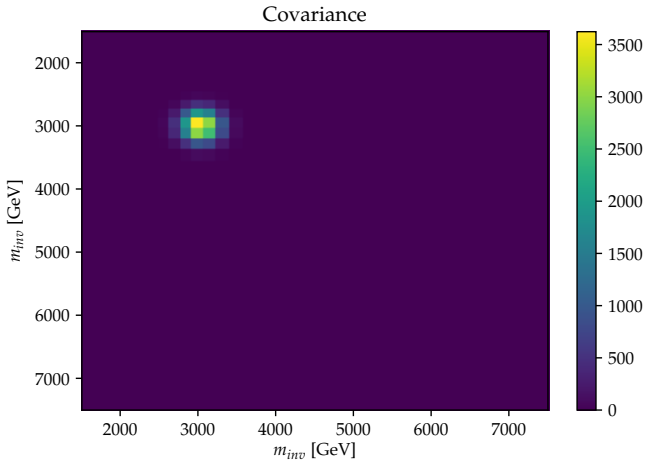


FIGURE 7.2: Covariance from the signal kernel, in eq. (7.13), after a fit is performed.

physics processes. The idea is then to have signal hypotheses for different parameters of the Gaussian distribution, and evaluate the extent to which the method is able to identify the signal. As it is in general easier to identify narrow and intense signals lying in the less-populated regions of the background, our interest is focused on detecting signals that are as broad and faint as possible on top of more populated background regions.

7.3.1 GP methods

Here we present an extension of the GP framework that is able to operate in a number of signatures given histograms corresponding to measured data and a Monte Carlo (MC) simulation of the background. The fundamental difference is that we will not take as an input the mean function μ of the GP from the parametric approach as in general such function is not available, but set $\mu(x) = 0$. As it is discussed in [84], and mentioned earlier in this Chapter, Gaussian Processes are in general flexible enough to regress a function even if the mean is set to zero.

We propose two procedures: one that involves two steps and another one with three steps. The two-step procedure stems from the work in [162], whereas the three step procedure fits the background shape in two steps, allowing for modelling of turn-on regions, and a final step accommodates the signal. We describe in further detail those two proposals below.

Two-step procedure

The distribution histogram from the background simulation is used for our tests in a two-step procedure, that are performed as follows:

First step A GP fit is performed on a pure background distribution (e.g. from simulation) using the background kernel Σ_B, with $\boldsymbol{\mu} = \mathbf{0}$. We obtain the posterior mean and covariance (that comes from the kernel with optimized θ_B).

Second step The optimized hyperparameters of Σ_B of that background fit are fixed in a second fit using the $\Sigma_{SB} = \Sigma_B + \Sigma_S$, to obtain a GP for the full model, including

the parameters of the signal θ_S, that leads to identify a concentrated excess or deficit centered at m with width t (recall eq. (7.13)).

The tests are performed as follows. The first step is applied directly on the simulated background, from which the parameters θ_B are extracted. In order to proceed with the second step, we generate many datasets from the simulated background sample, known as pseudodata or *toy* experiments, following a Poisson distribution per bin. A simulated or artificial signal is then injected in a toy, where the fit of the second step is performed. The process of injecting a signal into a background toy is repeated several times (each time in a different toy) for each signal hypothesis to obtain a distribution of values for the extracted parameters. We also test the method by fitting the GP in the second step in the absence of any signal (i.e. in a background toy histogram) to inspect the extent to which the method performs a spurious detection.

Three-step procedure

The three-step procedure operates in the following manner: the first two steps model the background and the third step the signal plus background distribution. This procedure has its origin in the difficulties found in using a single background step, as described in the two-step procedure, in distributions where there is a turn-on regions, as it is the case for the invariant mass top-quark pair spectrum, that we will study below. The three steps are the following:

First step This is the same as the first step in the two-step procedure. We obtain θ_B, which is a base for the background fit.

Second step The optimized hyperparameters of Σ_B of the first step are fixed for a second GP background fit using the $\Sigma_{BT} = \Sigma_B + \Sigma_S$, on the same background distribution. This "signal" component of the background accommodates the turn-on.

Third step The optimized hyperparameters of Σ_{BT} from the previous two steps are fixed for a third fit using the $\Sigma_{BTS} = \Sigma_{BT} + \Sigma_S$, to obtain a GP for the background (plus turn-on) plus signal, where it is possible to extract signal parameters.

The tests are performed in a way similar to that of the two-step procedure: once the background parameters are obtained with the first and second step (using the background sample), the artificial or simulated signal is injected in different background toys, for the signal parameter extraction.

7.4 Datasets and signal injection

7.4.1 Dijet dataset

This dataset is taken from the ATLAS General Search [45] simulated Standard Model background. It is a simulation of the QCD processes from proton-proton collisions with a center-of-mass energy of $\sqrt{s} = 13$ TeV. This simulated dataset was produced

for studying the data recorded by the ATLAS detector in 2015 (3.2 fb^{-1}). The distribution invariant mass values of simulated ATLAS dijet events that we use is displayed in figure 7.3, which comes from the dataset used for the General Search in [45].

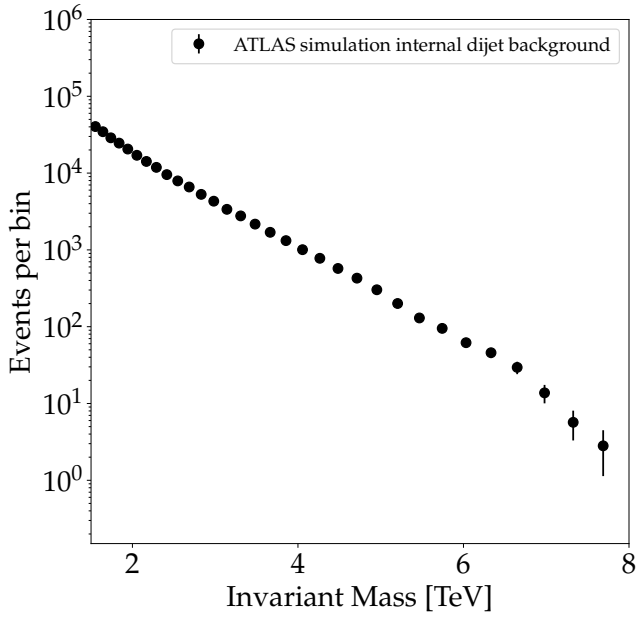


FIGURE 7.3: Distribution of simulated dijet events invariant mass used for the ATLAS General Search in [45].

The events are produced with the Pythia 8 MC generator [56] for the hard process in the collision and hadronization, and the GEANT 4 package [165] for the detector response, as it is described in reference [45]. Details on the trigger and offline selections on the events are also described in the given reference. In the specific case of events containing two jets the selection corresponds to passing a requirement of missing transverse energy (E_T^{miss}) higher than 200 GeV in the event as well as the E_T^{miss} trigger, or an event containing a jet of transverse momentum (p_T) greater than 500 GeV, passing the single jet trigger. Jets are reconstructed using the anti-k_t algorithm [50] with a parameter $R = 0.4$, and are required to pass a minimum p_T of 60 GeV and an absolute pseudorapidity ($|\eta|$) lower than 2.8. The bin width of the histogram is constructed using the formula:

$$h(x) = \sqrt{\sum_{i=1}^{N_{\text{objects}}} k^2 \sigma_i^2(x/2)}\,, \tag{7.14}$$

where k is the width of the bin in standard deviations and $\sigma_i^2(x/2)$ is the expected detector resolution for the p_T of object i (in our case, each of the two jets) evaluated at $p_T = x/2$ and $\eta = 0$. In the formula, x represents the variable used that is the invariant mass in our case. The bin widths vary in an approximate range of (90, 360) GeV, increasing as the mass values are higher.

For the dijet mass spectrum, the parameters are $N_{\text{objects}} = 2$, and $k = 2$ that corresponds to ± 1 standard deviations. Also, the resolutions are respectively 2.4% and 2.0% for dijet masses of 2 TeV and 5 TeV for the ATLAS detector, at a center of mass energy $\sqrt{s} = 13$ TeV [166].

7.4.2 Top-quark pair data

We test the method in a second simulated dataset, that is used in searches for heavy particles decaying into top quark pairs with the ATLAS detector, presented in ref. [167]. There is a number theories that postulate the production of heavy resonances decaying in that final state, e.g. topcolor-assisted-technicolor Z' production [168], or a Kaluza-Klein excitation of the Graviton [169–171]; here we use simulated samples for the first case (Z').

The studies performed here use simulated samples for both SM processes and for the hypothetical Z' signal, in the invariant mass spectrum. For the SM background, the contributions of processes in decreasing order of importance are: the production of a top-quark pair ($t\bar{t}$), a weak boson, W or Z, in association with jets (V+jets), a single top quark, multiple jets and dibosons. All background process are generated using MC simulations except the multiple jet contribution, that is estimated from data. The background spectrum is displayed in figure 7.4.

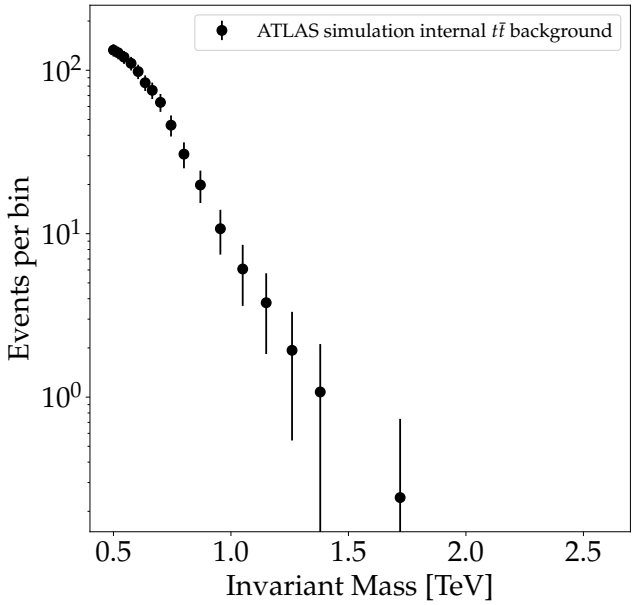


FIGURE 7.4: Distribution of simulated $t\bar{t}$ events invariant mass used for the analysis in [167].

For the signal, our studies use simulations where the mass was fixed to two values: 750 or 1250 GeV, with an amplitude of 1.85 pb and 1 pb respectively. These amplitude values come from the 95% confidence level limit on the Z' cross-section derived in that study [167]. We also applied different amplification factors to those signals, in order to perform the tests.

All experimental details (MC simulations used, simulated triggers, object and event selection, spectrum binning choice, etc.) are provided in the given reference ([167]). In the studies presented here, we work only with the selection (identified as "category 3, resolved selection" in [167]), where one top quark candidate decays hadronically, and another leptonically (muon-neutrino), and both have associated b-tagged jets.

7.4.3 Identifying signals in invariant mass spectra

We inject signals of specified standard deviation and mass (mean) of the signal Gaussian distribution. Some care needs to be taken regarding amplitude values, as often invariant mass spectra span values of several different orders of magnitude in the event counts: signals with the same amplitude can be much smaller than statistical noise if lying on a region heavily populated by the background, or a very obvious discrepancy if located at the very tail of the spectrum. Furthermore, the resolution of the detector also varies through different values of the invariant mass as we have seen in the previous subsection.

Instead of directly using the amplitude, we prescribe values for a quantity defined from a signal-over-background ratio (R). We calculate the amplitude of the Gaussian signal from the mean and standard deviation: R is the ratio of signal to background events in a window constructed from the interval given by the signal mean and the standard deviation. The number of events taken into account in a given window is given by the bin counts of the distribution contained in the window; bins whose center are outside the range (mean $\pm\sigma$) are not counted.

$$R = \frac{\text{Injected signal events in the window}}{\text{Background events in the window}} . \tag{7.15}$$

Analogously, the extraction of the R (and hence the amplitude) of the signal comes from the same ratio within a window defined by the mean and width of the extracted signal from the parameters of the signal model. We present in figure 7.5 an illustration of a Gaussian signal injection in the dijet spectrum. (The window marks the edges of the bins taken into account for R and are therefore not necessarily symmetric with respect to the signal mean.)

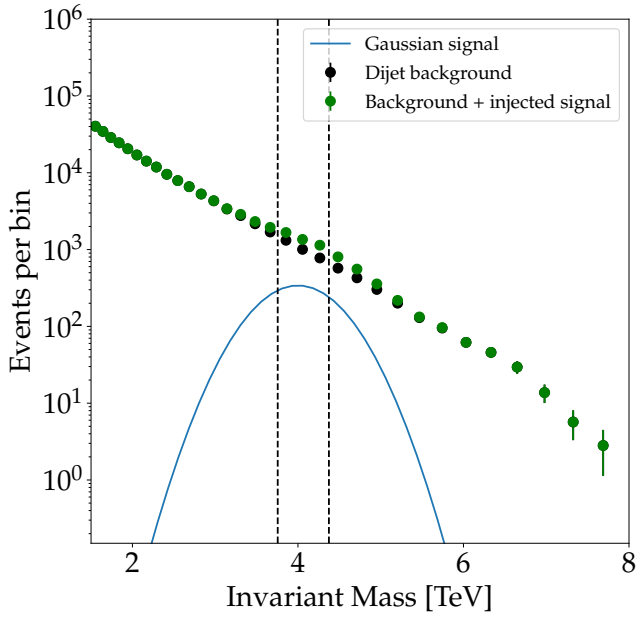


FIGURE 7.5: Illustrative plot for the definition of R. The background and background-plus-signal histograms, and the original Gaussian function are plotted. Vertical dashed lines indicate the identified window where signal and background events are counted.

In order to have a sense of how to convert from R to the number of signal events within a window in the dijet spectrum, we present table 7.1 with corresponding

R	0.1	0.2	0.3	0.4
# signal events	720 ± 20	1450 ± 50	2160 ± 50	2890 ± 60

TABLE 7.1: Number of injected signal events within a window for values of R, for a signal of 3 TeV mass and 150 GeV width. The values and errors obtained are the mean and standard deviation of the distribution of values obtained after repeating the injection in 100 background toys.

values for the case of a signal at 3 TeV and a width of 150 GeV. Since a signal of a specified R will get a (different) amplitude value for each background toy sampled where it is injected, we get a distribution of values for the number of events injected, from which we calculate the mean and standard deviation, presented in the table.

We choose a total of 60 signal hypotheses to be injected in the dijet spectrum. This quantity comes from the different possible combinations of values for the Gaussian distribution used for the signal, namely the mass (3, 3.5, 4, 4.5, and 5 TeV), the width (150, 300, and 450 GeV), and the R ratio (0.1, 0.2, 0.3, and 0.4). The values were chosen to cover a wide range of the spectrum and different intensities of the signal hypotheses. For every signal hypothesis, the sampling of the signal Gaussian is performed and injected 100 times, one per background toy, and thus a distribution of 100 values for each signal parameter is obtained for each hypothesis.

The spurious signal detection test is performed as follows. We perform the background GP fit in the background distribution and the signal-plus-background GP fit in 100 different toy backgrounds. The extracted (spurious) signal parameters are then used as a benchmark for comparison with the distribution of the extracted parameters from genuine detections.

7.5 Results

In this section we present the results of applying the GP methods described above for identifying an excess of events for an invariant mass spectrum that can be indicative of new physics.

7.5.1 Two-step procedure GP fit on the dijet spectrum

In this subsection we present results for the GP method in the dijet invariant mass spectrum. We also present results for the benchmark parametric function fit.

We start by discussing the plot on figure 7.6 that displays two GP fits and the background toy with an injected signal. The excess coming from the signal is visible in the neighborhood of the 3.5 TeV in the significance of the dataset with respect to the background-only fit (middle panel). The plotted significances correspond to "signed z-values only if p-value < 0.5", as defined in [172], where the p-value is calculated assuming that each bin count follows a Poisson distribution. In appendix C we give details on parameters chosen to initialize and run the GP method.

The residual plot shown in figure 7.7 shows a clearer representation of the signal injected and the identification by the GP fit method.

In the example displayed in the last figures, we can qualitatively say that the identification is performed successfully, as the extracted parameters are close to the signal parameters. However, one example of misidentification due to the width of the injected signal with respect to the binning is presented in figures 7.8 and 7.9. Even if the same procedure was applied, except that the signal now has a mass of 4

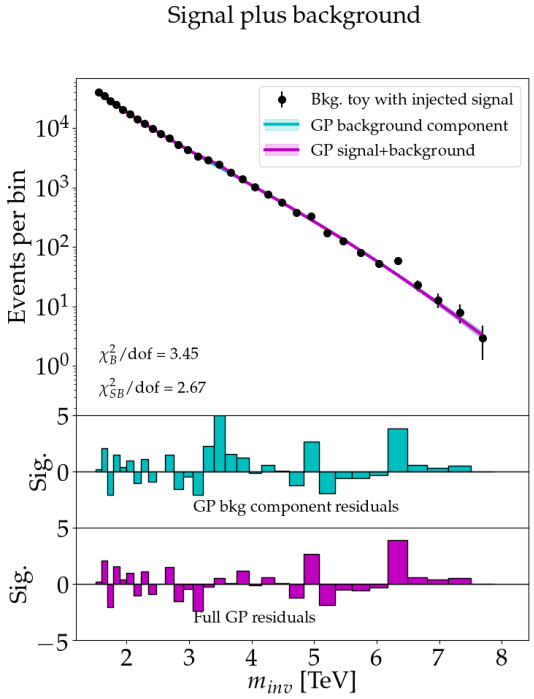


FIGURE 7.6: Top panel: invariant mass spectrum displaying a GP background fit, event counts for a background toy with signal injected centered at 3.5 TeV with a width of 150 GeV and R of 0.1, and a signal plus background fit. The magenta line is the posterior mean of the GP fit using the Σ_{SB} kernel; the blue line represents the background-only component of the GP fit. Middle and bottom panels: per-bin significance of the discrepancy between the event counts and respective fits.

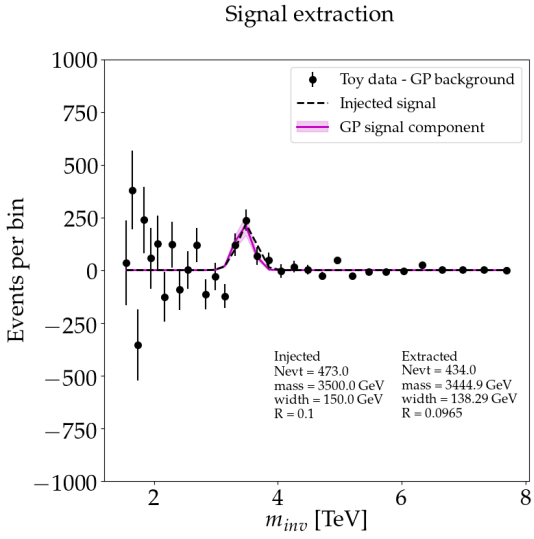


FIGURE 7.7: Residual plot corresponding to figure 7.6. The GP signal component (solid magenta) and the signal injected (dashed black line) are displayed as well as a subtraction of the toy data set with a signal injected minus the background GP fit (black dots with error bars). Injected and extracted signal values are shown.

TeV, the signal GP fits a narrow excess (a statistical fluctuation) at low values of the spectrum, near 1.7 TeV. The performance of the method is sensitive to the variable bin widths with respect to the injected signal width as we will discuss below. Ultimately, a sensible statement on the performance of the method can only be made on the basis of repeating the signal injection in many background toys and performing a fit for each of them; this procedure needs to be followed for each of the signal hypotheses studied.

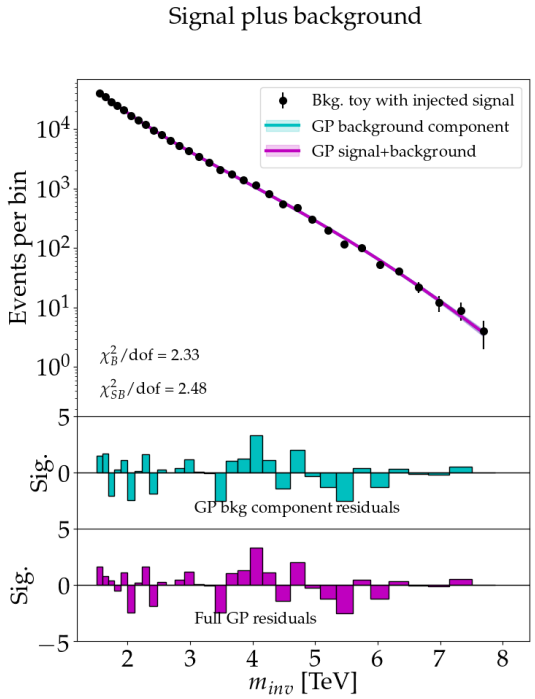


FIGURE 7.8: Top panel: invariant mass spectrum displaying a GP background fit, event counts for a background toy with signal injected centered at 4 TeV with a width of 150 GeV and R of 0.1, and a signal-plus background fit. Middle and bottom panels: per-bin significance of the discrepancy between the event counts and respective fits.

Extracted parameters

The plots shown in figure 7.10 display the relationship between injected and extracted values for the width parameter. From the plot we notice that the higher the values of R, the closer the extracted distribution of width values approaches the true injected signal.

We have mentioned in section 7.4.3 that the identification is more difficult as the signal becomes wider. This statement does not hold in all the width plots, specifically in the 150 GeV case; an explanation follows. We need to take into account that the tests are performed on a dataset that comes in the form of histograms (i.e. we do not have access to per-event information) that have a prescribed binning according to the criteria that were used in the General Search analysis in [45]. Then, injected signals occupy less bins as the signal hypothesis is higher, and in particular injected signals with a width of 150 GeV may be poorly identified, because virtually all the injected signal events are concentrated in a single bin for the higher mass points.

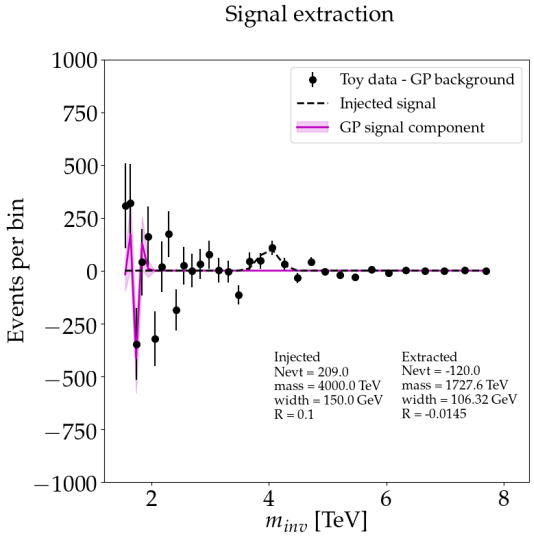


FIGURE 7.9: Residual plot corresponding to figure 7.8. The GP signal component and the signal injected are displayed as well as a subtraction of the toy data set with a signal injected minus the background GP fit (black dots with error bars). The GP fit signal component (solid magenta) incorrectly identified the injected signal (dashed black line). Injected and extracted signal parameter values are shown.

Similar linearity plots for the mass parameter can be found in figure 7.11. In general, good agreement between the injected and extracted parameter distribution is found. The distributions for higher mass values, and in particular those of the 5 TeV signal hypotheses with lower R, lead to incorrect identification. An explanation is that a fixed R can lead to different levels of identification difficulty across the spectrum, despite being a more uniform quantity for injecting signals than the number of events.

A set of plots corresponding to the last parameter, the R, is presented in figure 7.12. We can see good agreement for most hypotheses, but for higher values of mass (lower rows), the distribution of values becomes more inaccurate. This is showing once again that having the same R in the lower populated background regions, i.e. higher mass in this case, as elsewhere may lead to undesirably faint signals that are far beyond detection in some regions.

A plot summarizing all extracted R values plus the extracted spurious signals is presented in figure 7.13. We can notice here and also in figure 7.12 that in some cases the extracted value of the R is negative. Hence, the GP signal kernel has identified a deficit as a signal (or, technically, the integral of the signal component of the GP mean prediction is a negative number). The cases in which such negative values appear are indicative of an incorrect identification.

The spurious measurements reported are $R = 0.05 \pm 0.2$ which is compatible with no signal and leads to an upper error of 0.25.

Comparison with other two-step procedure options

We explored also two variants of the GP two-step procedure we proposed in sec. 7.3, as follows:

- **Option A: Freeze background mean and hyperparameters**: Similar to the default option, but prescribing the posterior mean of the first step as an input

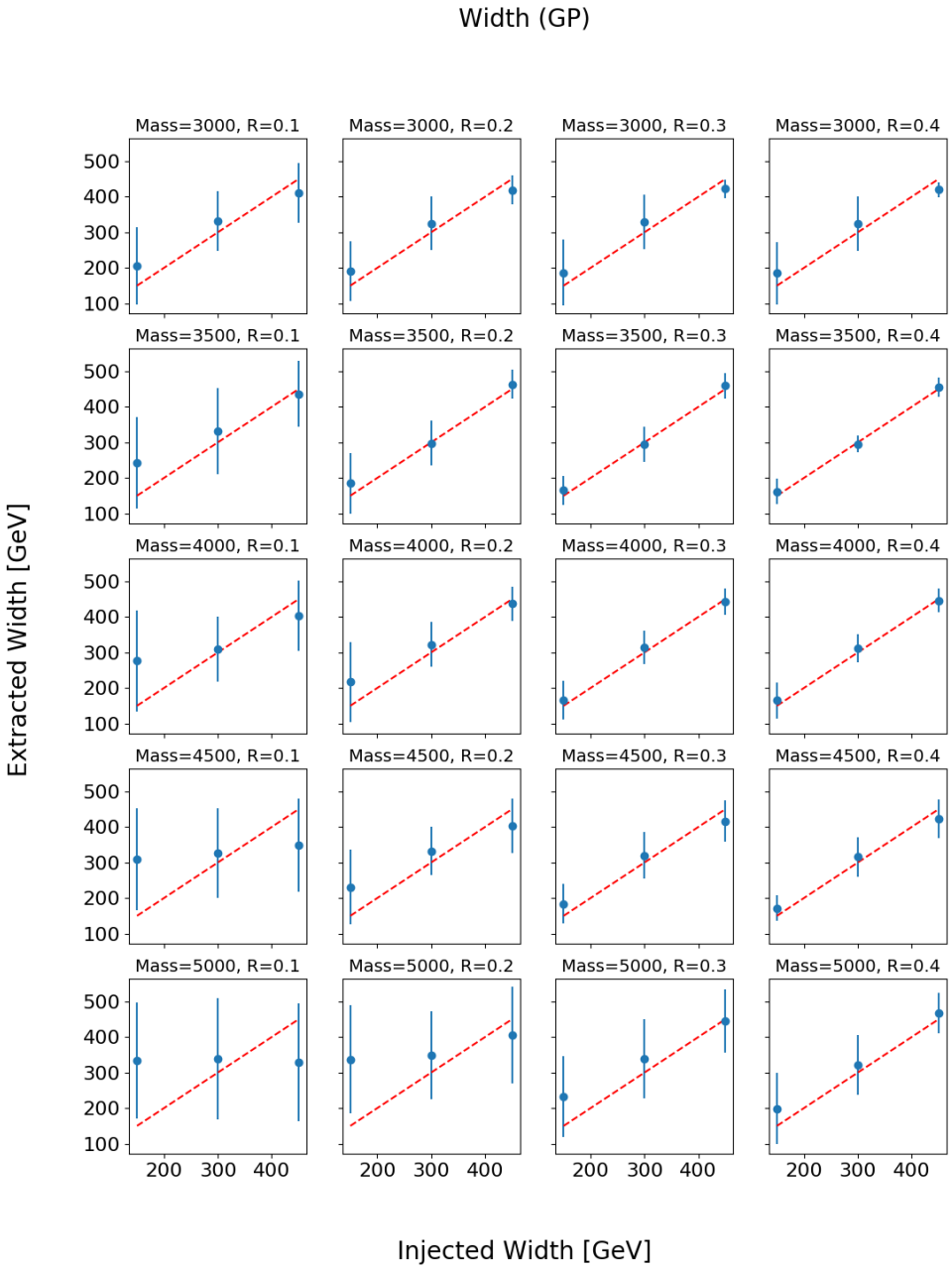


FIGURE 7.10: Linearity plots for the width of the signal injected in the dijet spectrum; each plot corresponds to indicated mass M (in GeV) and R pair of values. The values of R are the same for each plot column, and those of the width are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted mass values. A dashed red $x = y$ line is plotted for reference.

mean for the second fit. Such prescription aims to constrain the second step fit from the (background-) data-driven first step.

- **Option B: Single step background plus signal**: Perform a single GP fit using Σ_{SB} on data that may contain signal. Thus, this procedure is completely data driven, as it does not rely on the pure background sample.

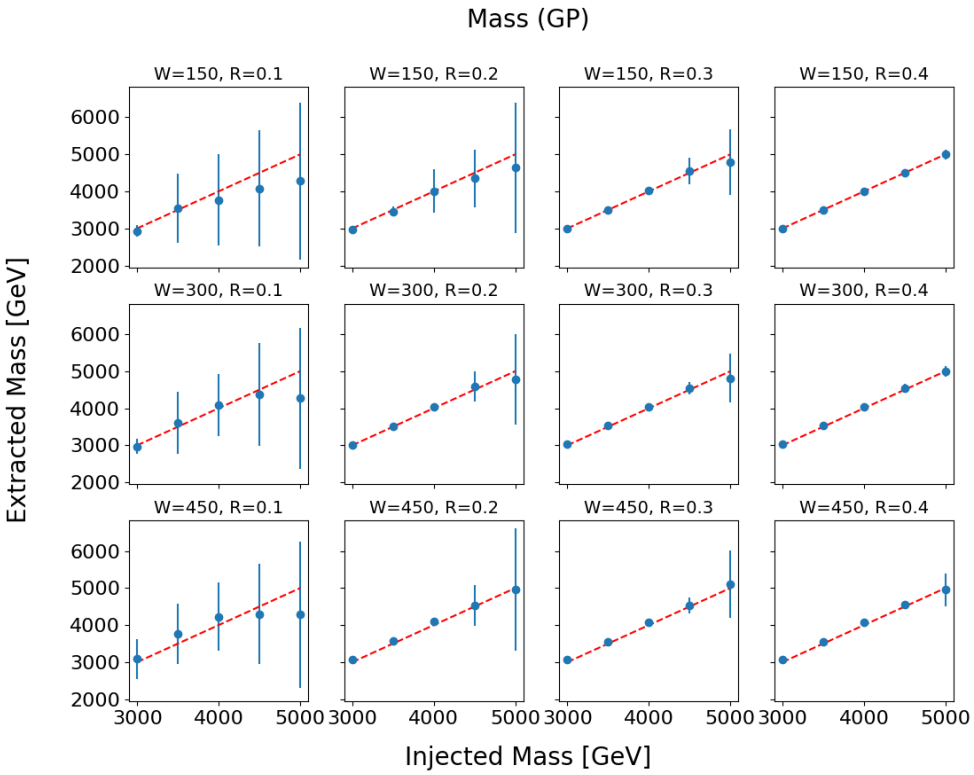


FIGURE 7.11: Linearity plots for the mass of the signal injected in the dijet spectrum; each plot corresponds to indicated width W (in GeV) and R pair of values. The values of R are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red $x = y$ line is plotted for reference.

Corresponding plots for the distribution of R values for options A and B can be found respectively in figures 7.14 and 7.15.

General good agreement is found in the linearity plots for all options. However, we can see that the higher mass hypotheses tend to be identified poorly; this is due to the specific binning of this spectrum, where R can lead to undesirably faint signals at high mass values (because of small event counts in such bins).

The values presented as the upper error of the spurious signal detections serve as an indication of the ability of the method to detect genuine signals. The method achieves upper errors of R =0.25 for the default and option A, which indicates that there is not a particular gain in detecting spurious signals when the mean constraint from Option A is imposed. As we stated before in subsection 7.3.1, GPs are in general tools flexible enough to model the distribution without prescribing a mean function, and in terms of signal extraction it did not represent a significant gain in the extraction power.

Finally, in Figure 7.16 we present normalized plots for the χ^2/ndof distribution obtained for the fits from the three options. The values are extracted for all possible hypotheses using the 100 toys for each hypothesis; for each option, this means 6000 values. Here it is possible to notice once again that the difference between Options 1 and 2 is not significant in terms of reproducing the spectrum (with the signal injected). Option B performs slightly better in terms of reproducing the spectrum with

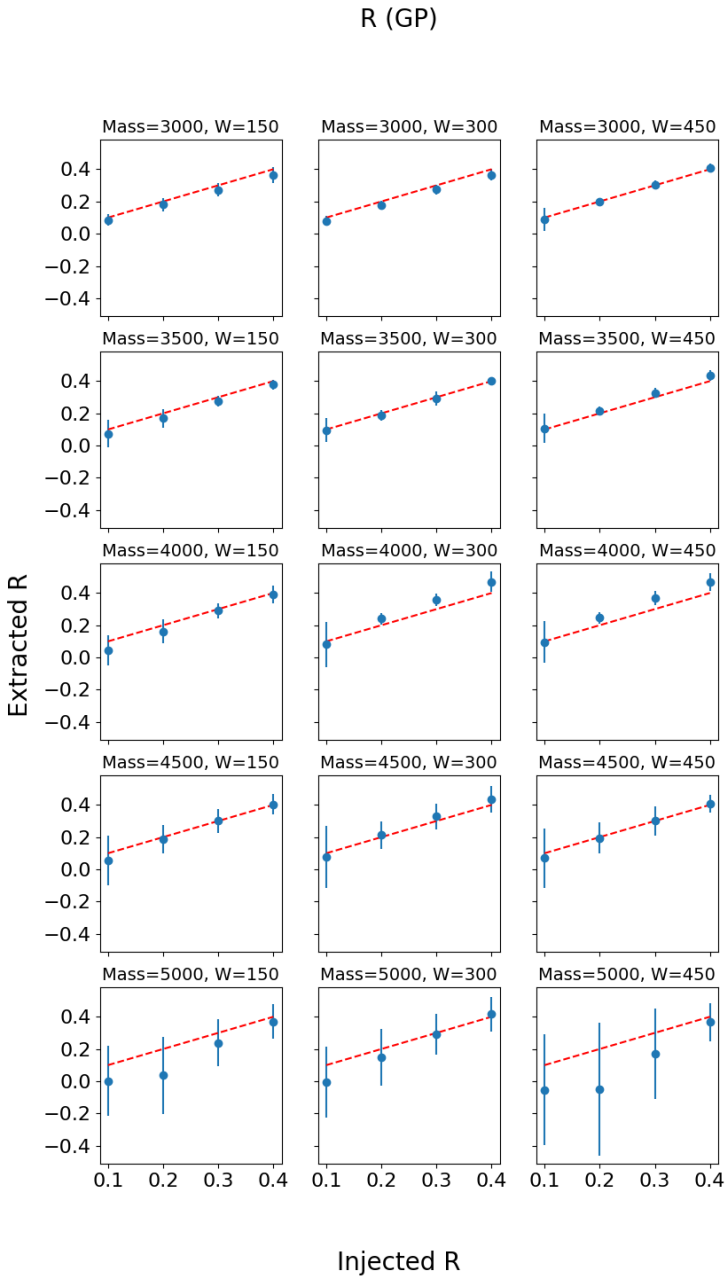


FIGURE 7.12: Linearity plots for the R of the signal injected in the dijet spectrum; each plot corresponds to indicated mass M and width W pair of values (both in GeV). The values of the width are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red $x = y$ line is plotted for reference.

signal, which may come from the fact that a 9-parameter space is being explored during optimization (more flexible than the 4-parameter one in the second step of the default and option A, for the signal kernel). However, the flexibility in Option B comes with the price of being more prone to spurious detection, reaching to R values of 0.37.

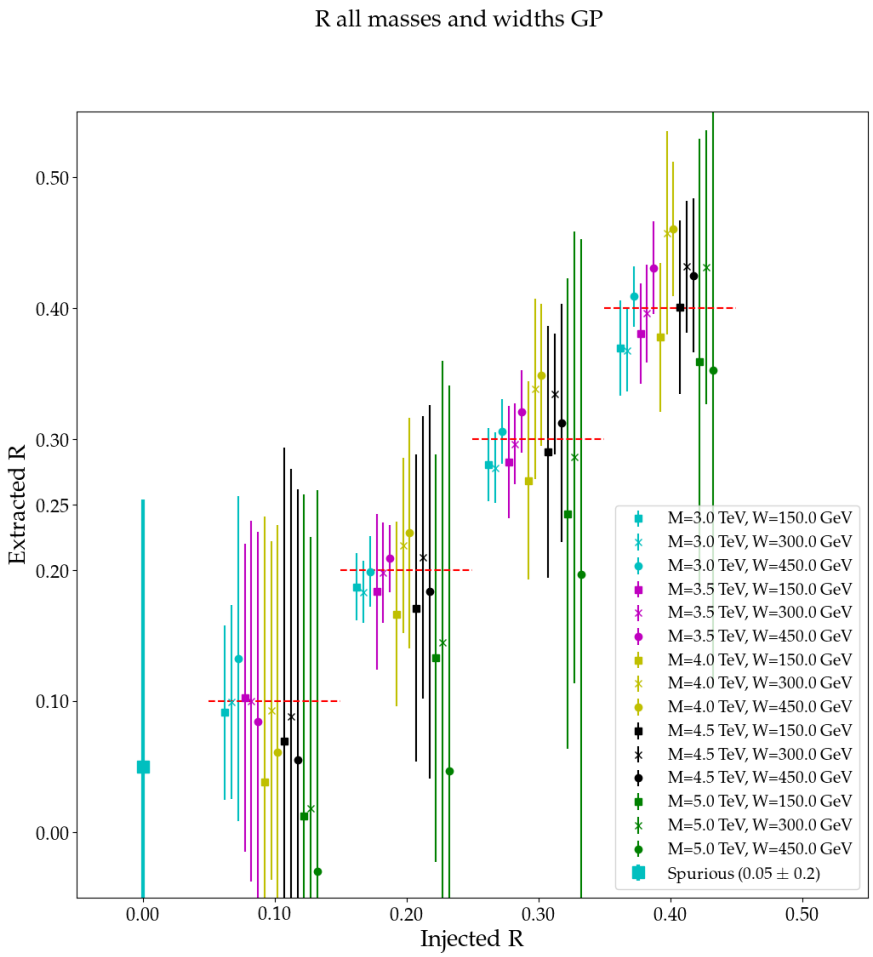


FIGURE 7.13: Linearity plots for the R parameter, for all masses and widths of the signal injected in the dijet spectrum, and spurious detection (injected R = 0). The error bar comes from the RMS of the distribution of extracted values. Values of R increase by 0.1 from 0 to 0.4, included; points appear slightly shifted in the horizontal axis for better visibility. The upper value of the error bar in the spurious detection (0.25) is presented. Dashed horizontal red lines are plotted for reference.

7.5.2 Benchmark: Parametric fit of the dijet mass spectrum

As a benchmark scenario for the dijet invariant mass spectrum, we use two parametric models to fit the background given by the following functional forms:

$$\text{Three-parameter background fit function:} \quad f_3(x|\theta) = \theta_0(1-x)^{\theta_1}x^{\theta_2}\,, \tag{7.16a}$$

$$\text{Five-parameter background fit function:} \quad f_5(x|\theta) = \theta_0(1-x)^{\theta_1}x^{\theta_2}x^{\theta_3\log(x)}x^{\theta_4(\log(x))^2}\,. \tag{7.16b}$$

To give a sense of how these functions behave in the context of the dijet spectrum, we provide a depiction of both background fits in figure 7.17. The two resulting functions are close to each other in a way that the difference is not visible, but a value of the chi-squared is provided for each plot. We can perform an F-test [173] to assess the difference between the two models; that essentially evaluates how much

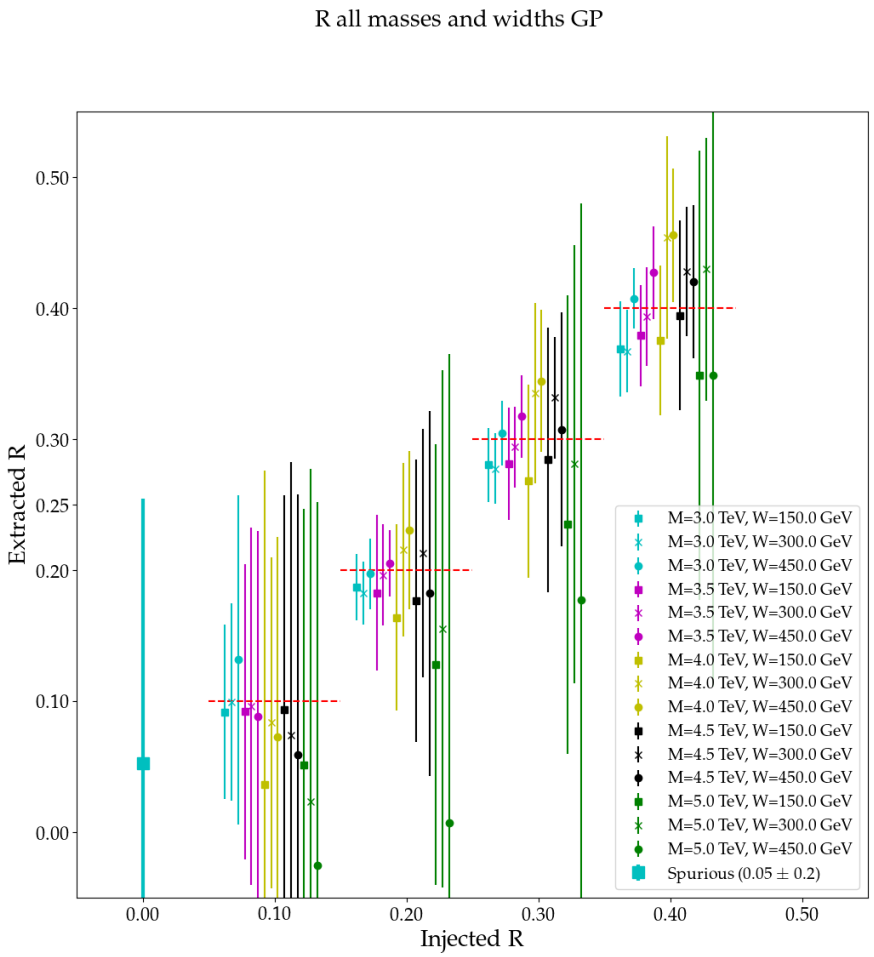


FIGURE 7.14: **Option A** Linearity plots for the R parameter, for all masses and widths of the signal injected in the dijet spectrum, and spurious detection (injected $R = 0$). The error bar comes from the RMS of the distribution of extracted values. Values of R increase by 0.1 from 0 to 0.4, included; points appear slightly shifted in the horizontal axis for better visibility. The upper value of the error bar in the spurious detection (0.25) is presented. Dashed horizontal red lines are plotted for reference.

it is gained from adding the two extra factors (the ones containing θ_3 and θ_4) in the five-parameter model. The criterion we use is that less than 0.9 will lead to prefer the new model (with more parameters); the closer this number is to one, the less one has gained from adding the extra terms with new parameters. The value obtained for an F-test is 0.9404, so we keep the three parameters for the tests presented here. We leave the results corresponding to the five-parameter model in appendix D and carry on with the three-parameter model, as both lead to a similar background fit.

Two-step procedures similar to that presented in section 7.3 are used for each of the background fit functions. A maximum likelihood fit is performed on the background distribution and then signals are injected in toy distributions generated from the background, where we perform a signal-plus-background fit, using a sum of the

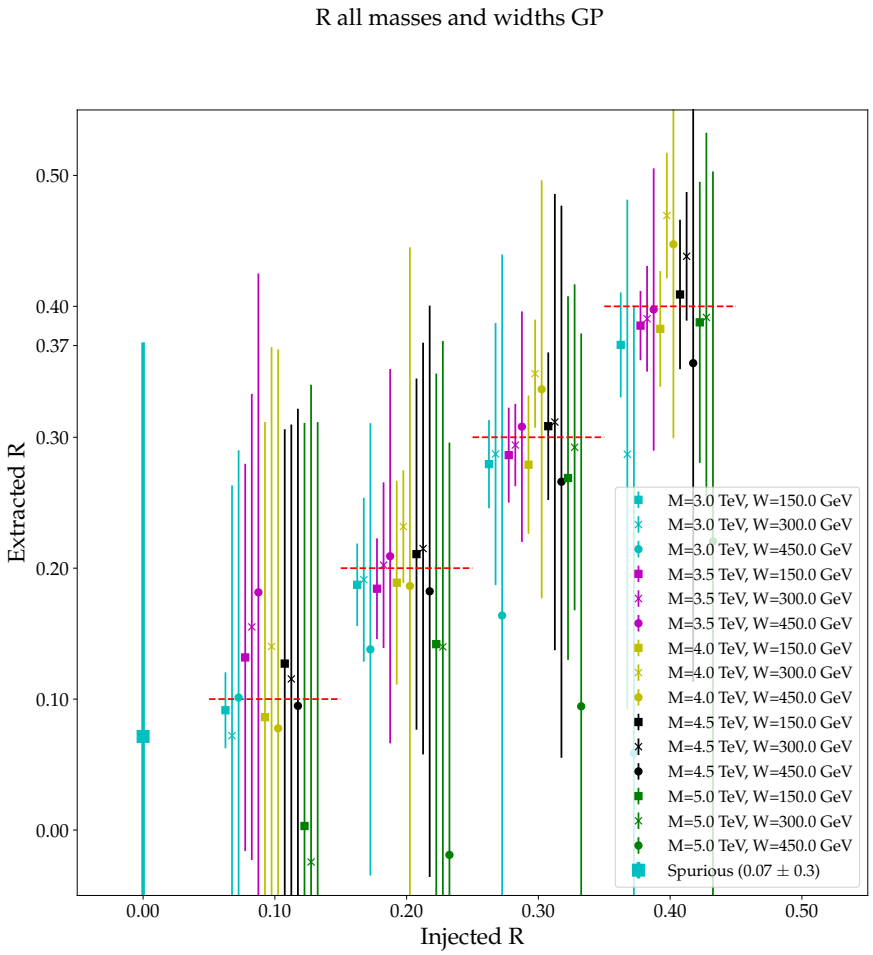


FIGURE 7.15: **Option B** Linearity plots for the R parameter, for all masses and widths of the signal injected in the dijet spectrum, and spurious detection (injected R = 0). The error bar comes from the RMS of the distribution of extracted values. Values of R increase by 0.1 from 0 to 0.4, included; points appear slightly shifted in the horizontal axis for better visibility. The upper value of the error bar in the spurious detection (0.37) is presented. Dashed horizontal red lines are plotted for reference.

background fit function used plus a Gaussian function:

$$f(x|\theta_{SB}) = f_{\text{3-or-5}}(x|\theta_B) + \text{Gaussian}(x|\theta_S) \tag{7.17}$$

where θ_B are the set of (three or five) parameters of the background model and θ_S corresponds to the amplitude, mean and width of the Gaussian distribution. As the parametric fit is in general more rigid than the GP approach, the parameters obtained in the background fit are used to initialize the signal-plus-background fit, without keeping them fixed.

As a reference comparison scenario for the specific case of two jets, we present results for the parametric approach, where we study the same cases as before injecting different signal hypotheses and testing both genuine and spurious detections. The setup for constructing the datasets is identical to the one used in the GP approach in 7.4.3.

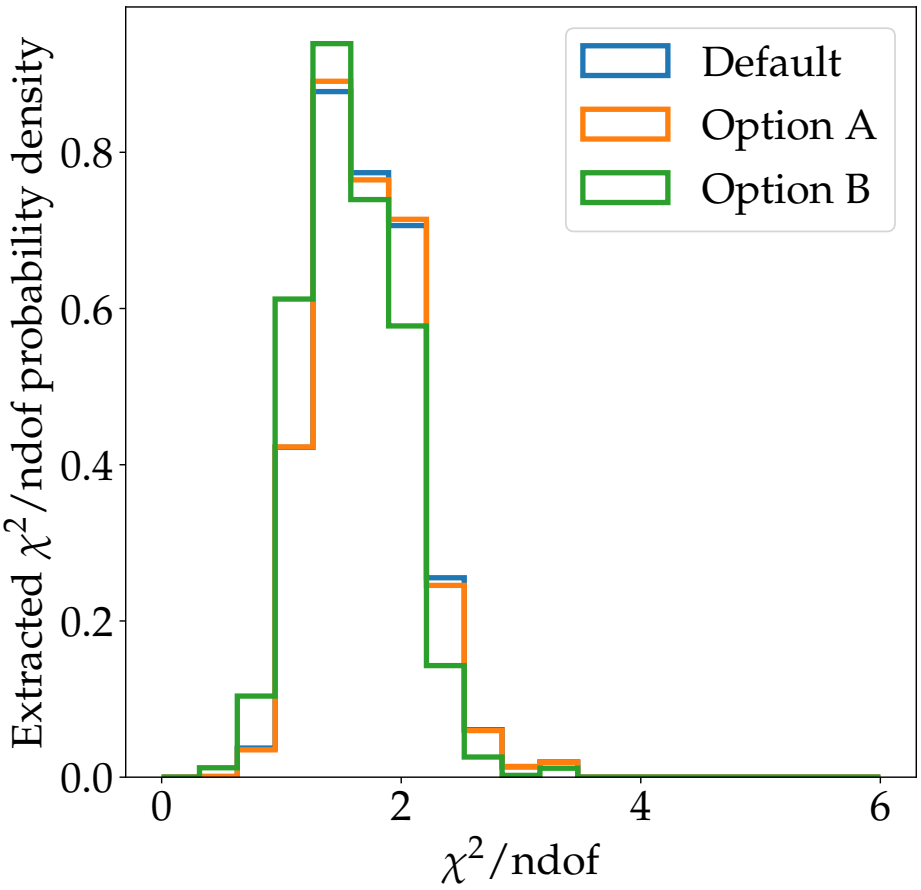


FIGURE 7.16: χ^2/ndof normalized distributions for the three options.

In figures 7.18, 7.19 and 7.20 we present linearity plots for extracted signal values using the three-parameter function to fit the background plus a Gaussian function for the signal. The figures correspond respectively to the extracted width, mass, and R. Despite finding a general good agreement between the injected and extracted values and obtaining smaller error bars from the distributions than those found in the GP approach, there are two important observations from the plots in figure 7.18. First, the fact that the extraction becomes more difficult for higher masses and low R (plots closer to the bottom-left corner) as it appeared in the GP approach. Second, the extracted values appear to be underestimated for the highest injected value (450 GeV) with R 0.3 and 0.4.

The corresponding mass plots in figure 7.19 reflect the features we have discussed for the width regarding higher mass and low R. Besides that, the distributions of mass values show good agreement for all cases, including the 3 TeV mass signals with higher widths, unlike the values shown in the previous case in figure 7.18.

Figure 7.20 shows plots for the relation between injected and extracted R values in the case of the three-parameter function used to fit the background. Once again, here it is evident that higher masses become more challenging for the range of Rs we have injected in all widths.

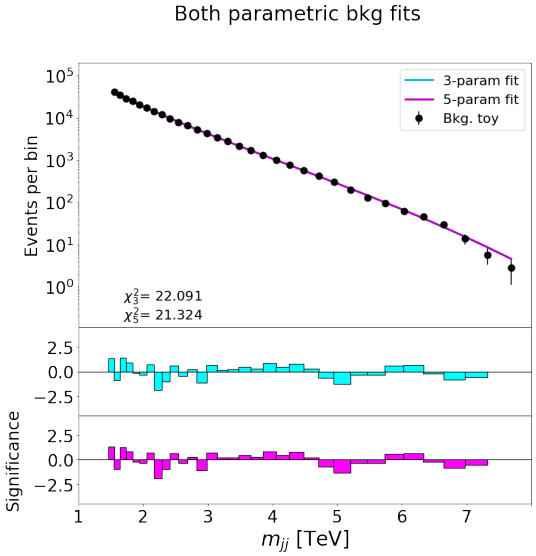


FIGURE 7.17: Parametric three- and five-parameter background fits in the General Search dijet background mass spectrum. The p-values corresponding to the χ^2 score given the degrees of freedom are respectively $p(\chi^2_3, 29) = 0.816$ and $p(\chi^2_5, 27) = 0.771$. This spectrum is simulated using the Pythia event generator [56], details on the text.

A summary plot for all R measurements and the spurious signal detection reference appears in figure 7.21. For the distribution of spurious detection values, see that the upper bound of the error bar reaches 0.14 in R, which is lower than what we have found for the GP approach.

Even if the parametric approach, as we have seen, is more accurate in extracting signal parameters, the GP method remains an option with several advantages. We should note that the GP method provides a tool that is more flexible to shapes, and includes some physical information encoded in the kernel, which is preferable than the ad-hoc nature of the parametric form. Also, the parametric background model explicitly contains a distribution designed for the spectrum, and the signal shape used on top is exactly the one used for the shape generation sample (Gaussian), that leads to such better performance.

7.5.3 Three-step procedure in $t\bar{t}$ invariant mass spectrum

We present results of applying the three-step GP procedure in the invariant mass spectrum of top quark pairs, described in sec. 7.3.1. In an effort to use the two-step procedure in this spectrum, we observed that the signal kernel was not able to identify injected signals, but always accommodated the turn-on part of the background. Thus, we used the $\Sigma_{TB} = \Sigma_B + \Sigma_S$ to model the background in two steps; an illustration of this is presented in figure 7.22. The contribution of the turn-on component is evident at low masses, where the optimization leads to identifying a concentration of events centered at the beginning of the spectrum, and with a width of 144 GeV (that was left to fluctuate between 100 and 500 GeV in the optimization) covering the turn-on region.

The third step is then used to identify a signal. The two signal Z' hypotheses available are used for injection, at 750 GeV (1.85 pb) and 1250 GeV (1 pb), which is repeated in 100 background toys. The three step procedure is able to identify the signals only if they are amplified by a factor with respect to their original amplitude

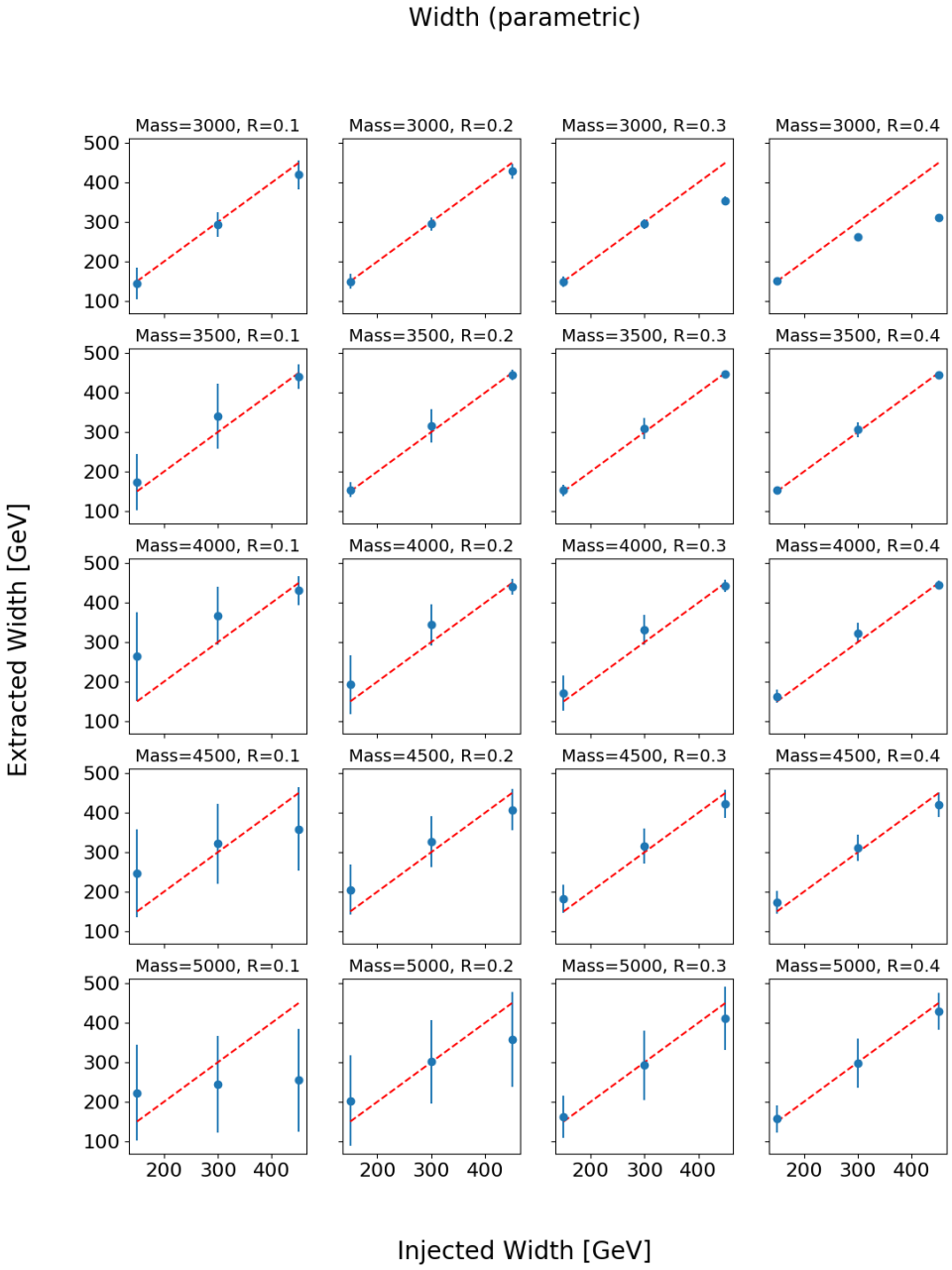


FIGURE 7.18: Parametric approach (three-parameter background fit): Linearity plots for the width of the signal injected in the dijet spectrum; each plot corresponds to indicated mass M (in GeV) and R pair of values. The values of R are the same for each plot column, and those of the width are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted mass values. A dashed red line $x = y$ is plotted for reference.

value, depending on the hypothesis. Also, we imposed different lower thresholds in the minimum value of the allowed signal mass mean range to 550 and 600 GeV. In figure 7.23 we present an example of a 750 GeV Z' signal injected and amplified by a factor 15; the corresponding signal extraction plot appears in 7.24.

The injected signal values are presented in table 7.2. To give a sense on where the signal is located, we report the center of the bin that contained the highest amount

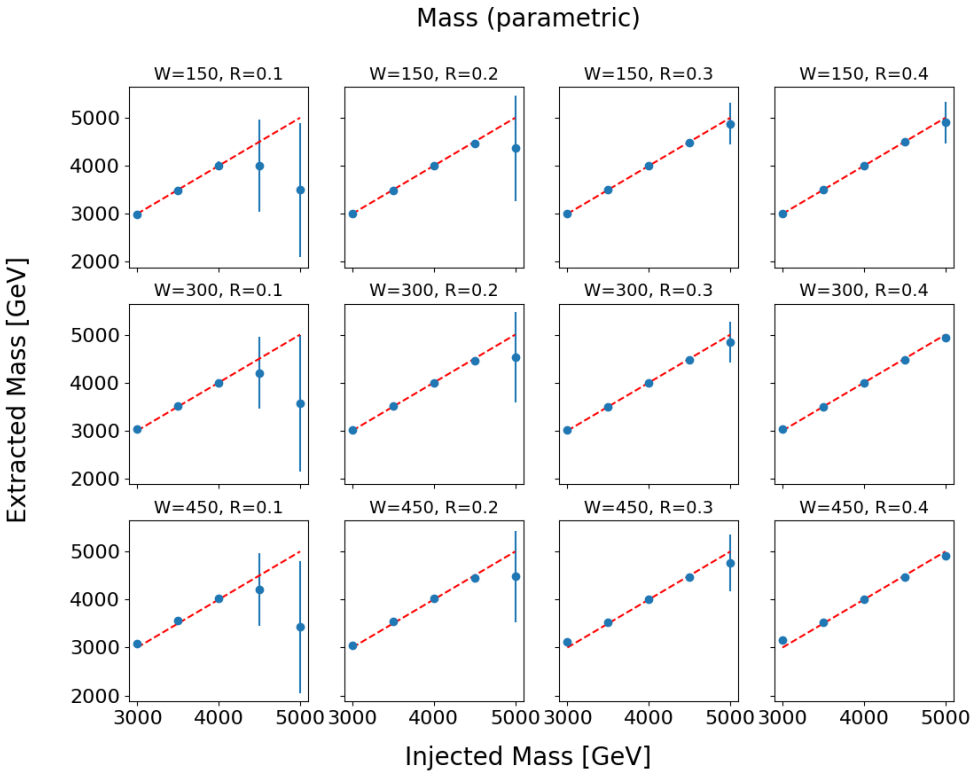


FIGURE 7.19: Parametric approach (three-parameter background fit): Linearity plots for the mass of the signal injected in the dijet spectrum; each plot corresponds to indicated width W (in GeV) and R pair of values. The values of R are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red $x = y$ line is plotted for reference.

of events (bin(max)), and a window defined around that bin by including on each side as many as necessary to reach 34% of the events; that way the window contains a value roughly above 68% of the signal distribution events, which corresponds to ± 1 standard deviations. The table reports the bin with the highest amount of events and half of the window size. Further, an extracted R value using the window and its standard error from repeating the injection in 100 background toys are presented in the table. In contrast to the dijet spectrum, that spans from approximately 1 to 8 TeV, the $t\bar{t}$ spectrum appears in the range of 0.5 to 2.5 TeV. Therefore, the signals here are relatively wider than in the dijet case.

Results for extracted values of mass and width of the signal appear in tables 7.3 and 7.4 for minimum mass thresholds of 550 and 600 GeV respectively. The mass and width parameters are taken from the optimized hyperparameters of the signal kernel obtained in the third step; the tables present the mean and average value for fits from injecting each of the hypotheses in 100 background toy experiments.

For a minimum mass threshold 550 GeV, in table 7.3, we observe different performance for the two injected mass hypotheses, as compared with the values reported in table 7.2. For 750 GeV, with the smaller amplification factor of 5, the signal mean value is lower than the center of (bin(max)) and the width value was overestimates the half window, but for the higher amplifications both the mean and width were compatible within error. The values of R are all above that of the reported injected

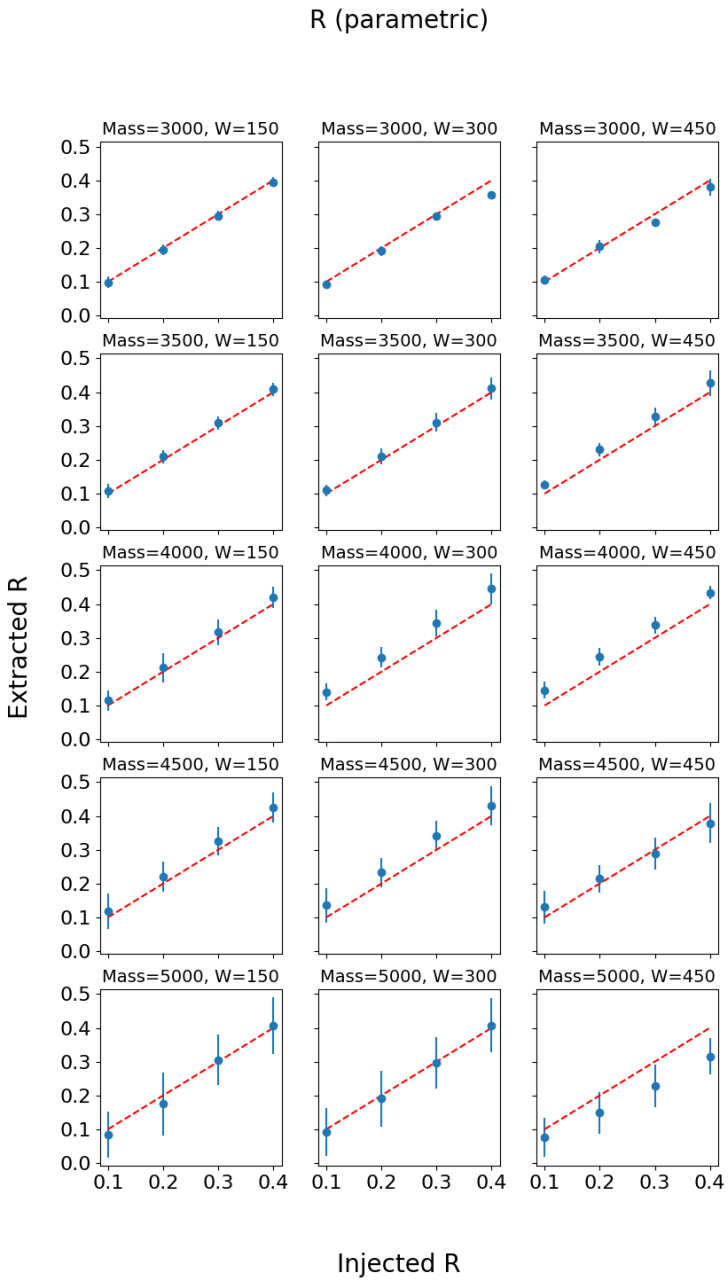


FIGURE 7.20: Parametric approach (three-parameter background fit): Linearity plots for the R of the signal injected in the dijet spectrum; each plot corresponds to indicated mass M and width W pair of values (both in GeV). The values of the width are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red line $x = y$ is plotted for reference.

R. In the case of 1250 GeV hypothesis, the method has difficulties extracting both the mass and the R value for all amplification factors (even if the mass appears to approach the correct value as the amplification increases). Also, width values are within error properly estimating the size of the window.

We also report results when imposing a minimum mass of 600 GeV, in table 7.4.

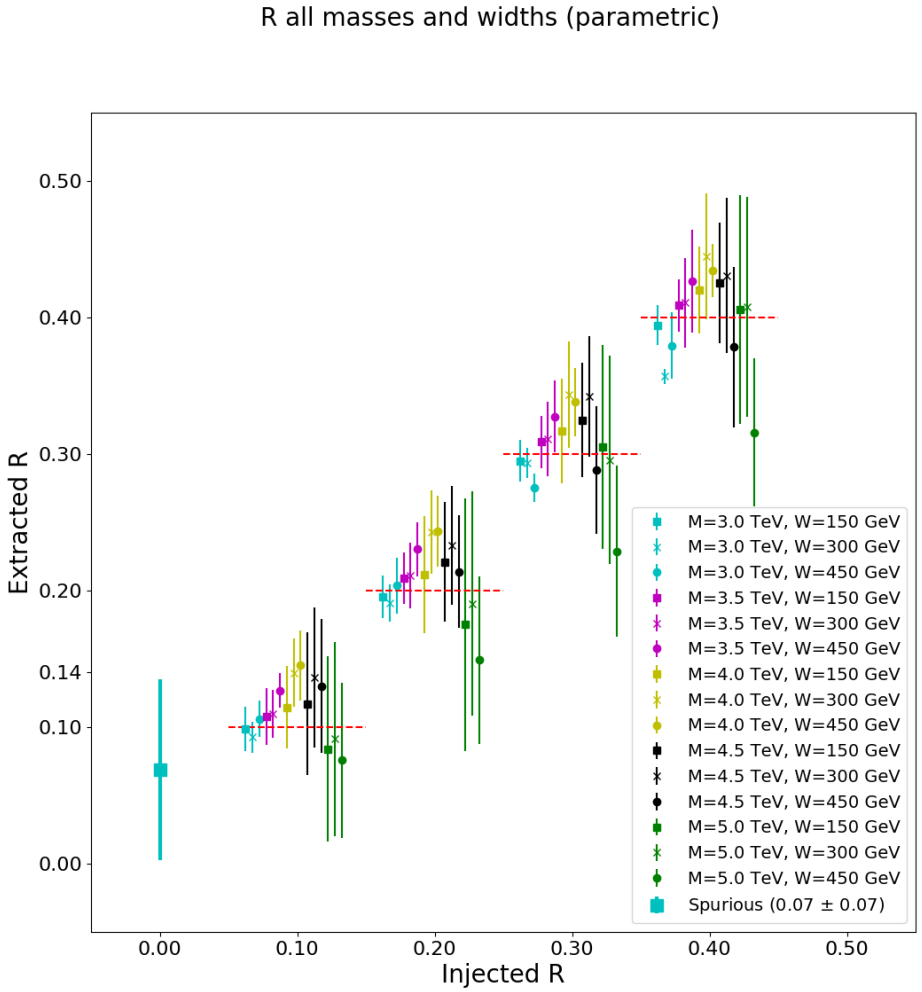


FIGURE 7.21: Parametric approach (three-parameter background fit): Linearity plots for the R for all masses and widths of the signal injected in the dijet spectrum and spurious detection. The upper value of the error bar in the spurious detection (0.14) is presented. Values of R are from 0.1 to 0.4 in intervals of 0.1; points appear shifted in the horizontal axis for better visibility. A dashed red line is plotted for reference.

With respect to the case of minimum mas 550 GeV, the behavior is somehow similar. For the 750 GeV hypothesis, the amplification by a factor 5 lead to underestimation of the mass and overestimation of the width, but the R value is compatible with the injected one. For higher amplifications in the same hypothesis, the mean and width are consistent with injected values (except the width at a factor 15 that is slightly underestimated); however, the R values were underestimated. All values are underestimated in the 1250 hypothesis, except that of the signal width when amplifying the signal by 5, and in that scenario it is unlikely that we are in the presence of a genuine detection, as the R value is marginally above zero.

A spurious detection test was also performed for the two cases of minimum mass threshold, denoted in the table as "No signal" in the first row of tables 7.3 and 7.4. We can notice that the extracted mass values on each case are driven towards the minimum allowed value for the mass, in which case both the turn-on component of

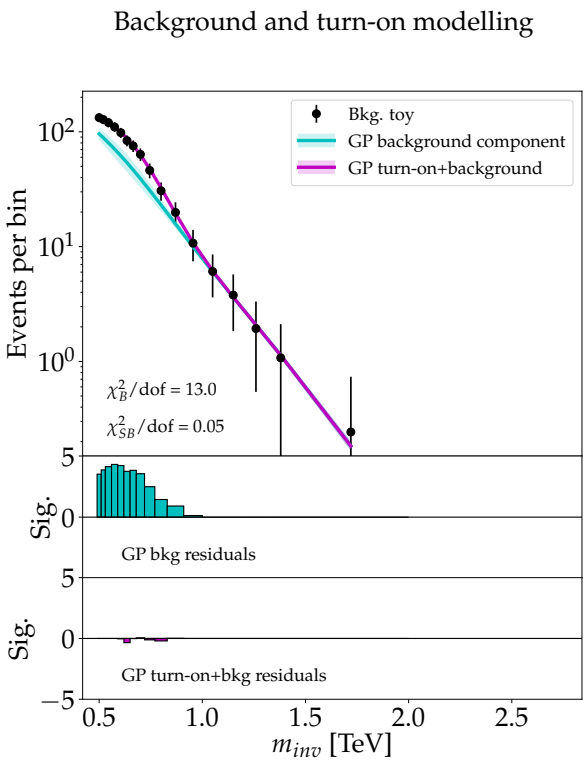


FIGURE 7.22: Background of the $t\bar{t}$ invariant mass spectrum (black dots), modeled with the first two steps of the three-step procedure (solid magenta). The pure Σ_B GP component is also presented (solid blue).

Hypothesis (factor)	bin(max) [GeV]	half window [GeV]	R
750 GeV (5)	700	82.5	0.100 ± 0.001
750 GeV (10)			0.200 ± 0.002
750 GeV (15)			0.300 ± 0.003
1250 GeV (5)	1260	335.0	0.18 ± 0.01
1250 GeV (10)			0.35 ± 0.01
1250 GeV (15)			0.53 ± 0.01

TABLE 7.2: Injected values in the $m_{t\bar{t}}$ spectrum. The center of the bin containing the maximum number of signal events (bin(max)) is reported, as well as half of the length of the window used. The reported R value is an average calculated by repeating the injection in 100 background toys; the error on that value is taken as the standard deviation of all obtained R values.

Σ_{TB} and the signal kernel contribute to the background turn-on.

7.6 Conclusions and outlook

In this chapter we explored different aspects of GP approached that are able to perform background modelling and signal detection, prescribing $\mu(x) = 0$ (no mean information). The two-step procedure was able to detect signals with different widths, intensities, and locations in the dijet invariant mass spectrum. We use R, a ratio

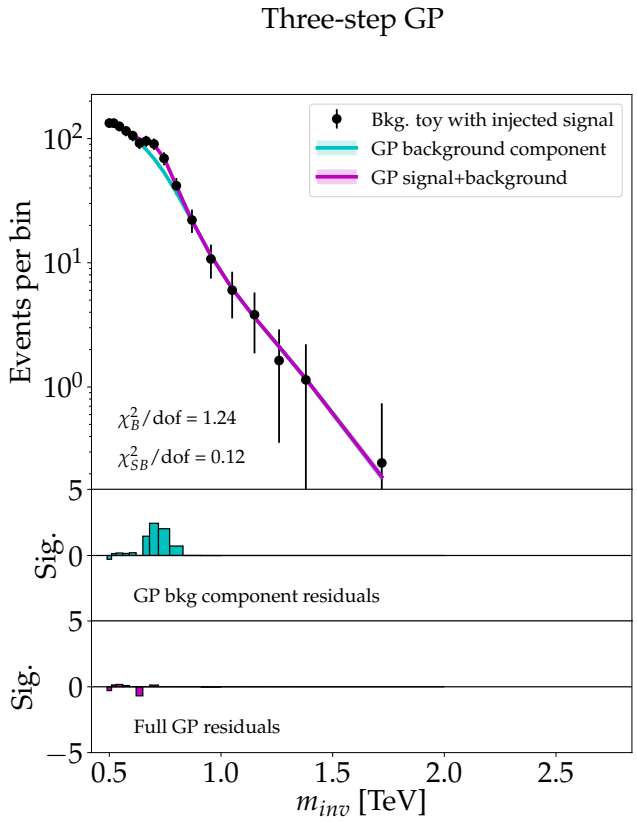


FIGURE 7.23: Top panel: $t\bar{t}$ invariant mass spectrum displaying a GP background fit (with turn-on), event counts for a background toy with Z' signal injected centered at 750 GeV amplified by a factor 15, and a signal plus background fit. The magenta line is the posterior mean of the GP fit using the Σ_{BTS} kernel; the blue line represents the background-plus-turn-on component of the GP fit. Middle and bottom panels: per-bin significance of the discrepancy between the event counts and respective fits.

Hypothesis (factor)	m [GeV]	w [GeV]	R
No signal	557.4 ± 14.8	137.3 ± 8.8	0.02 ± 0.03
750 GeV (5)	625.1 ± 24.0	120.4 ± 14.5	0.23 ± 0.07
750 GeV (10)	682.2 ± 17.1	87.1 ± 17.4	0.42 ± 0.07
750 GeV (15)	703.6 ± 7.9	71.4 ± 9.82	0.48 ± 0.04
1250 GeV (5)	809.0 ± 88.9	341.0 ± 44.1	0.01 ± 0.01
1250 GeV (10)	845.7 ± 81.3	339.0 ± 47.9	0.09 ± 0.05
1250 GeV (15)	947.4 ± 155.4	278.5 ± 98.5	0.21 ± 0.13

TABLE 7.3: Three step procedure extracted values for signals in the $m_{t\bar{t}}$ spectrum. Minimum mass threshold 550 GeV.

of events defined within a window to measure the intensity of the signal; since the spurious detection lead to identifying signals with strengths $R = 0.05 \pm 0.2$ we use that value as an indication of the typical faint signal that the method is capable to identify. Two modifications of the two-step procedure were also tested but did not lead to significantly better results when compared to the default.

We applied a GP method also in the more challenging case of the search for a

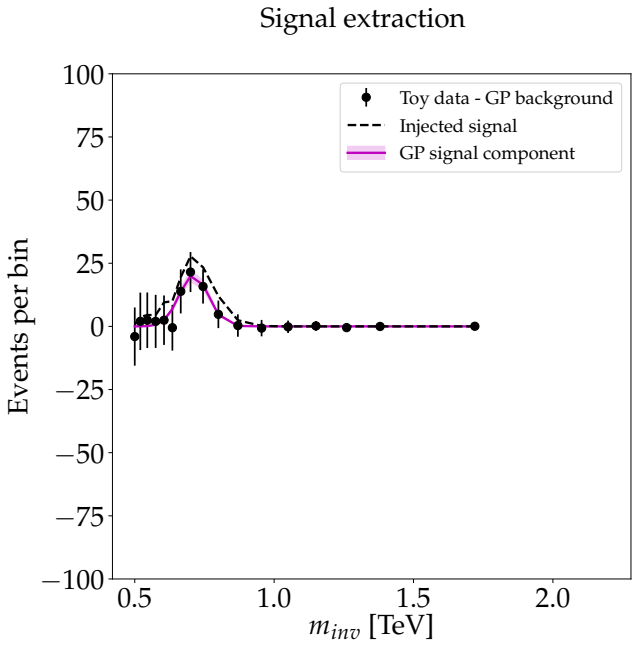


FIGURE 7.24: Signal extraction plot corresponding to fig. 7.23. The GP signal in the third step (solid magenta) and the signal injected (dashed black line) as well as a subtraction of the background toy with a signal injected minus the background GP fit (black dots with error bars).

Hypothesis (factor)	m [GeV]	w [GeV]	R
No signal	610.8 ± 53.0	125.5 ± 46.7	0.04 ± 0.05
750 GeV (5)	626.6 ± 20.9	119.3 ± 14.4	0.16 ± 0.06
750 GeV (10)	680.9 ± 19.5	84.7 ± 22.0	0.36 ± 0.08
750 GeV (15)	705.7 ± 8.7	63.1 ± 12.7	0.39 ± 0.07
1250 GeV (5)	929.4 ± 106.5	307.0 ± 74.9	0.01 ± 0.01
1250 GeV (10)	1047.6 ± 112.0	224.5 ± 77.6	0.19 ± 0.07
1250 GeV (15)	1115.6 ± 32.7	177.7 ± 32.3	0.35 ± 0.07

TABLE 7.4: Three step procedure extracted values for signals in the $m_{t\bar{t}}$ spectrum. Minimum mass threshold 600 GeV.

resonant signal in the $t\bar{t}$ invariant mass spectrum. There, a three-step method was presented and used: the first two steps model the background, including the turn-on region, and the third step is used for extracting the signal. We used two simulated Z' signal hypothesis samples, one with a mass of 750 GeV and the other at 1250 GeV. The original signal amplitudes were faint, and the method was not able to detect them; thus, the signals were amplified with different factors up to a level in which detection was possible.

The GP methods that we used provide the first steps towards alternative background modelling and signal identification techniques. However, the procedures presented here can see some improvement. In particular, in the definition of the intensity of the signal injected, where our definition of R is better than prescribing an single amplitude value across the spectrum, but fails at high bin widths, as we could see in the dijet spectrum tests. Also, the interest of model-independent searches can be better suited by having more generic methods likely by exploring other kernels,

and not relying e.g. in the construction of further fit steps, as it happens in the three-step procedure.

Further studies of this method may aim towards more data-driven approaches, by applying a GP method that is flexible enough to model a background-only distribution in the presence of signal. That direction has the advantage of not needing a MC simulation of the background, which in some cases provide a poor background estimation.

Chapter 8

Conclusions and outlook

This work revolves around the issue of model-independent searches New Physics in ATLAS with a special focus on Machine Learning techniques. In order to substantiate that problem, we presented in chapters 2 and 3 theoretical and experimental details respectively; further, we reviewed in chapter 5 a number of Machine Learning methods applied in the context of High Energy Physics, in the search for New Physics.

Inspired in General Searches performed by the ATLAS collaboration [45], we presented the result of monitoring many generic signatures using the TADA system. TADA is designed to provide quick feedback to specialists in the case a discrepancy appears in a particular signature, and also serves as a tool for validation and performance studies of the data taken and simulations. We put in place a system that automatically fills histograms in generic signatures, and that was used during the 2017 data taking period of ATLAS; no particular feature in the generic signatures triggered an alert to other groups during that period. The generic signature monitoring system can be easily expanded, as the software infrastructure is already built, and could serve for future data to be taken by the collaboration.

We also presented a method for collective anomaly detection with an application in HEP. It is based on a previous work that uses Gaussian Mixture Models in a semi-supervised two-step procedure. Our method uses a penalized likelihood for automatically performing variable selection. We used a set of simulation software for generating LHC-like proton-proton collision events and a fast simulation for emulating the detector response of ATLAS. The processes simulated correspond to Standard Model QCD processes as a background and a heavy (stop) resonance as a signal that leads to a final state of two jets. Our method performs slightly better than presented in the previous work, but with the advantage of having a built-in system for variable selection. We reported that the method underestimates signal proportions, particularly in the presence of the stronger signals injected ($> 15\%$). Further improvements of the presented method include the exploration of other penalties as well as the automation of several procedures within the algorithm for alleviating preprocessing.

Finally, we tested and modified a method that uses Gaussian Processes for modelling background and signal distributions in invariant mass spectra. We present two procedures NP searches in ATLAS in two cases: the dijet signature from the General Search and a dataset used for resonant searches decaying into top quark pairs. For the two-step procedure in the dijet dataset, we were able to compare several proposed options within the Gaussian Process method as well as with a traditional parametric fit method. The method was able to properly detect injected artificial Gaussian signals down to an R value[1] of 0.05 ± 0.2. The necessity of a

[1] R is a ratio of number of signal over background events defined within a window given by the (injected or extracted) signal distribution.

three-step procedure appeared when modelling the turn-on on the $t\bar{t}$ invariant mass spectrum. For this spectrum, we made use of the more challenging Z' signal hypothesis samples, where only in the case of amplifying a signal by a factor of at least 10 a proper detection was able to be achieved. These Gaussian process methods can see improvements e.g. in the definition of the R quantity, and automatization of different parts of the method. Improvements can also come from the exploration of new kernels, will help improve the methods presented here.

More generally, we foresee that Machine Learning will continue to help improving the techniques used for New Physics searches at many levels, as we have discussed. The challenges posed by the LHC datasets, and the absence of signs of New Physics so far, pose a unique scenario for pushing active research areas like deep learning or unsupervised methods to mine the available data.

Appendix A

Variable selection for the Penalized Anomaly Detection method

As we have seen in Chapter 6, variable selection is achieved by having (a set of) shrunk parameters that indicate which variable is to be removed. In the case of the penalty in eq. (6.12), from ref. [148], covariances are constrained to be the identity, and shrinkage drives means to zero for an uninformative variable indexed by p where the factorization of that variable can be performed for the Gaussian component k. For a given observation:

$$\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \mathbb{1}) = \mathcal{N}(x_{ip}|0,1)\mathcal{N}(\boldsymbol{x}_i^{D-1}|\boldsymbol{\mu}_k^{D-1}, \mathbb{1}^{D-1}) \,, \tag{A.1}$$

where the right hand side contains a product of a unidimensional Gaussian density for the p-th variable, and a $(D-1)$-dimensional Gaussian density where the arguments have the p-th variable removed. This is used to simplify the posterior for the adjusted EM algorithm that is[1]

$$\tau_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \mathbb{1})}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{k'}, \mathbb{1})} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i^{D-1}|\boldsymbol{\mu}_k^{D-1}, \mathbb{1}^{D-1})}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{x}_i^{D-1}|\boldsymbol{\mu}_{k'}^{D-1}, \mathbb{1}^{D-1})} \,. \tag{A.2}$$

A similar procedure is derived for the mean and eigenvalue penalization we described in the Chapter. We start by partitioning the features into two sets labeled by a and b and thus $\boldsymbol{X} = (\boldsymbol{X}^a, \boldsymbol{X}^b)$ such that the dimension of the realizations in $\boldsymbol{X}^a$ and that of those in $\boldsymbol{X}^b$ add to D; similarly the means are partitioned $\boldsymbol{\mu}_k = (\boldsymbol{\mu}_k^a, \boldsymbol{\mu}_k^b)$, and the covariances written in blocks of appropriate dimensions

$$\Sigma_k = \begin{pmatrix} \Sigma_k^{aa} & \Sigma_k^{ab} \\ \Sigma_k^{ba} & \Sigma_k^{bb} \end{pmatrix} \,. \tag{A.3}$$

Then, the factorization of variables in b (that are assumed to be the uninformative) for component k and a given observation $\boldsymbol{x}_i$:

$$\begin{aligned} \mathcal{N}\left(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \Sigma_k\right) = \;& \mathcal{N}\left(\boldsymbol{x}_i^b \,\middle|\, \boldsymbol{\mu}_k^b, \Sigma_k^{bb}\right) \\ & \times \mathcal{N}\left(\boldsymbol{x}_i^a \,\middle|\, \Sigma_k^{ab}\left(\Sigma_k^{bb}\right)^{-1}\left(\boldsymbol{x}_i^b - \boldsymbol{\mu}_k^b\right), \Sigma_k^{aa} - \Sigma_k^{ab}\left(\Sigma_k^{bb}\right)^{-1}\Sigma_k^{ba}\right) \,. \end{aligned} \tag{A.4}$$

The conditions that are required to be satisfied for this approach are that means in b are zero, the covariances in b across the Gaussian components are all equal to Σ^{bb}, and off-diagonal covariance blocks Σ_k^{ab}, and Σ_k^{ba} are blocks of zeros with appropriate

[1] In this subsection we skip the caret over the current parameter estimates to lighten up notation.

dimensions, for all k values; where

$$\Sigma^{bb} = \sum_{k=1}^{K} \pi_k \Sigma_k^{bb} \,. \tag{A.5}$$

Putting all this information together for modifying the EM algorithm leads to a posterior:

$$\begin{aligned} \tau_{ik} &= \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_i^b \,|\, \mathbf{0}^b, \Sigma^{bb}\right) \mathcal{N}\left(\boldsymbol{x}_i^a \,\middle|\, \mu_k^a + 0^{ab}\boldsymbol{x}_i^b,\, \Sigma_k^{aa} - 0^{ab}\Sigma^{ba}\right)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}\left(\boldsymbol{x}_i^b \,|\, \mathbf{0}^b, \Sigma^{bb}\right) \mathcal{N}\left(\boldsymbol{x}_i^a \,\middle|\, \mu_{k'}^a + 0^{ab}\boldsymbol{x}_i^b,\, \Sigma_{k'}^{aa} - 0^{ab}\Sigma^{ba}\right)} \\ &= \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_i^a \,|\, \boldsymbol{\mu}_k^a \Sigma_k^{aa}\right)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}\left(\boldsymbol{x}_i^a \,|\, \boldsymbol{\mu}_{k'}^a, \Sigma_{k'}^{aa}\right)} \end{aligned} \tag{A.6}$$

where the effect of the uninformative variables labeled b cancels out. The condition on the means is ensured by the penalty term p_1 but for the covariance condition model selection needs to be performed among all potential sets of uninformative variables, that satisfy the condition on the means, by using the Bayesian Information Criterion. More details can be found in ref. [151].

Appendix B

Tukey-transformed distributions for the PAD method

A Tukey ladder of powers [158] was used as a preprocessing step, as described in chapter 6. Here we present the plots corresponding to the coefficients presented in Table 6.1, in figures B.1 and B.2.

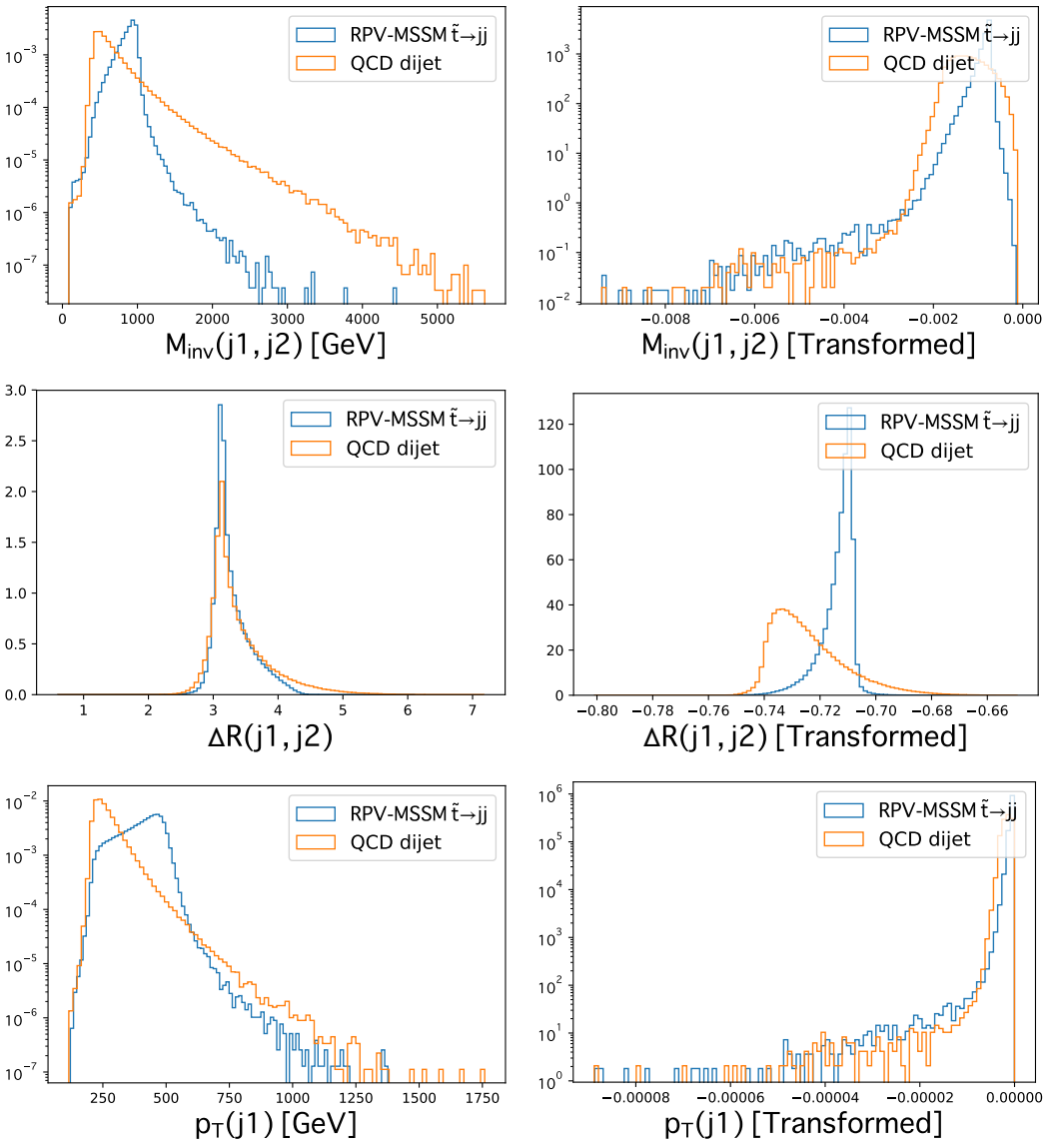


FIGURE B.1: Normalized distributions of signal and background dijet for kinematic and angular variables (left) and after the Tukey transformation (right).

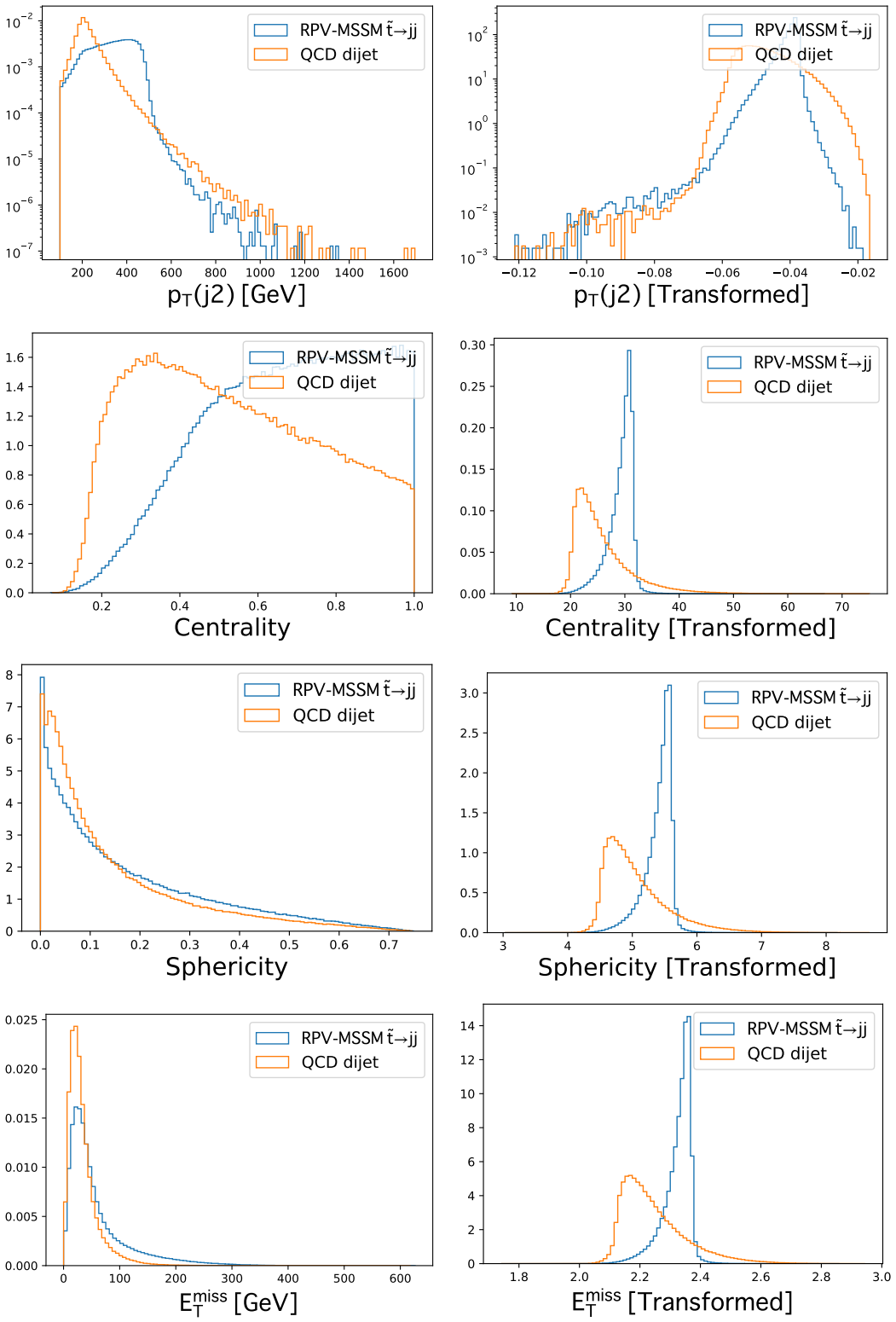


FIGURE B.2: Normalized distributions of signal and background dijet for kinematic and angular variables (left) and after the Tukey transformation (right).

Appendix C

Parameter initialization

We specify here values provided to perform the maximization of the likelihood with the minuit algorithm [174] via its Python interface [175].

C.0.1 Gaussian process method

Background

The initialization parameters for the background are:

- A = random(0, 10^6)
- a = random(0, 400)
- b = random(0, 10)
- c = random(0, 100)
- d = random(0, 650)

Limits on the values of those parameters:

- A = (100, 10^{15})
- a = (1, 3000)
- b = (0.1, 1000)
- c = (10, 3000)
- d = (200, 2000)

Obtained values for these parameters: $A = 2.47448640 * 10^{10}$, $a = 3.09264468 * 10^2$, b = 22.7827200, c = 2999.99668, d = 1999.85146.

Signal plus Background

Background kernel hyperparameters are kept frozen. The initialization for signal kernel parameters is:

- A = random(0, 3000)
- m = random(0, 3000)
- t = random(0, 200)
- l = random(0, 50)

Limits on the values of those parameters:

- A = (0, 100000)
- m = (1000, 7000)
- t = (50, 500)
- l = (10, 10000)

C.0.2 Parametric fit

Background

The initialization is the same for both the three- and five-parameter background fit. In the three-parameter fit, values for θ_3 and θ_4 are not used.

- θ_0 = random(0, 1)
- θ_1 = random(0, 8)
- θ_2 = random(0, 6)
- θ_3 = random(0, 1)
- θ_4 = random(0, 1)

Limits on the values of those parameters:

- θ_0 = (0, 10)
- θ_1 = (-20, 20)
- θ_2 = (-20, 20)
- θ_3 = (-20, 20)
- θ_4 = (0, 50)

Obtained for the three-parameter background model: θ_0 = 5.131641680942489, θ_1 = 8.252253105791226, θ_2 = -2.604810239837292.

Obtained for the five-parameter background model: θ_0 = 9.993937994190874, θ_1 = 8.655403124376392, θ_2 -1.8369620252642136, θ_3 = 0.3456933396129145, θ_4 = 0.056820004297544746.

Signal plus Background

Initial values for the background are the ones obtained in the background fit. Initial values on the Gaussian signal are:

- Amp = random(0, 1000)
- Mean = random(1000, 6000) (GeV)
- Width = random(100, 450) (GeV)

Limits on the values of those parameters:

- Amp = (0, 6000)
- Mean = (1000, 8000) (GeV)
- Width = (100, 450) (GeV)

Appendix D

Five-parameter background fit

Using the same parametric procedure described in our work, but using the five-parameter formula in eq. (7.16b) instead. We show analogous sets of plots for the extraction of the signal width, mass, and R parameters in figures D.1, D.2 and D.3 respectively. The extracted width values shown in figure D.1 are similar to those presented for the three-parameter case (cf. figure 7.18). Both features, the one regarding the poorer performance at high masses and the bias at 3 TeV for higher Rs, are also present in this plot.

Analogous comments to those made in the three-parameter case apply to the mass plots in D.2 (cf. figure 7.19).

The set of plots for the R parameter appearing in figure D.3 show a similar behavior (overall) to that found in the other parametric case (cf. figure 7.20). There is a difference in the plot corresponding to the signal at 3 TeV with 450 GeV width, where the 0.4 R is underestimated.

As a final plot in figure D.4 we present the summary of the extracted R values and the one coming from spurious detections. The distribution of spurious signal R values leads to a higher error upper bound (0.22) in spurious detections than that obtained for the three-parameter case (see figure 7.21).

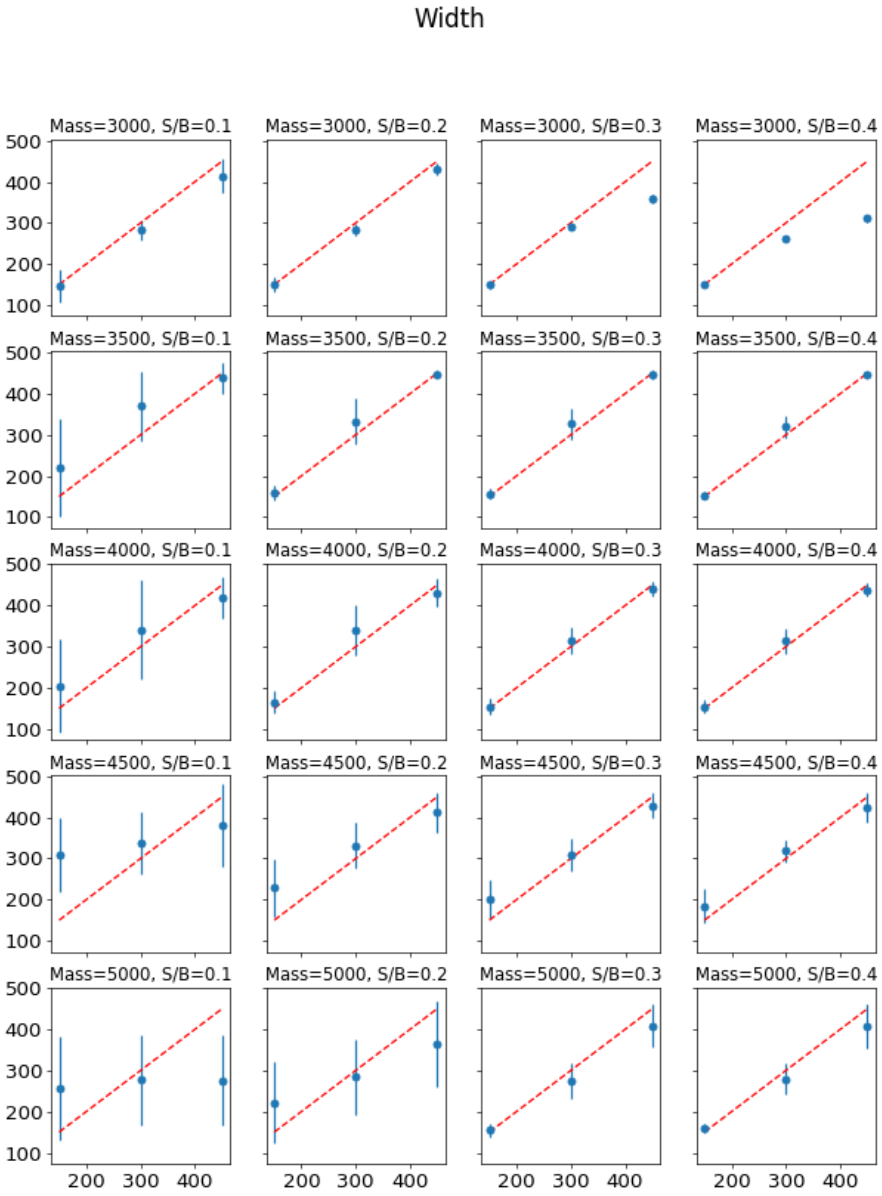


FIGURE D.1: Parametric approach (five-parameter background fit): Linearity plots for the width of the signal; each plot corresponds to indicated mass and R pair of values. The values of R are the same for each plot column, and those of the width are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted mass values. A dashed red line is plotted for reference.

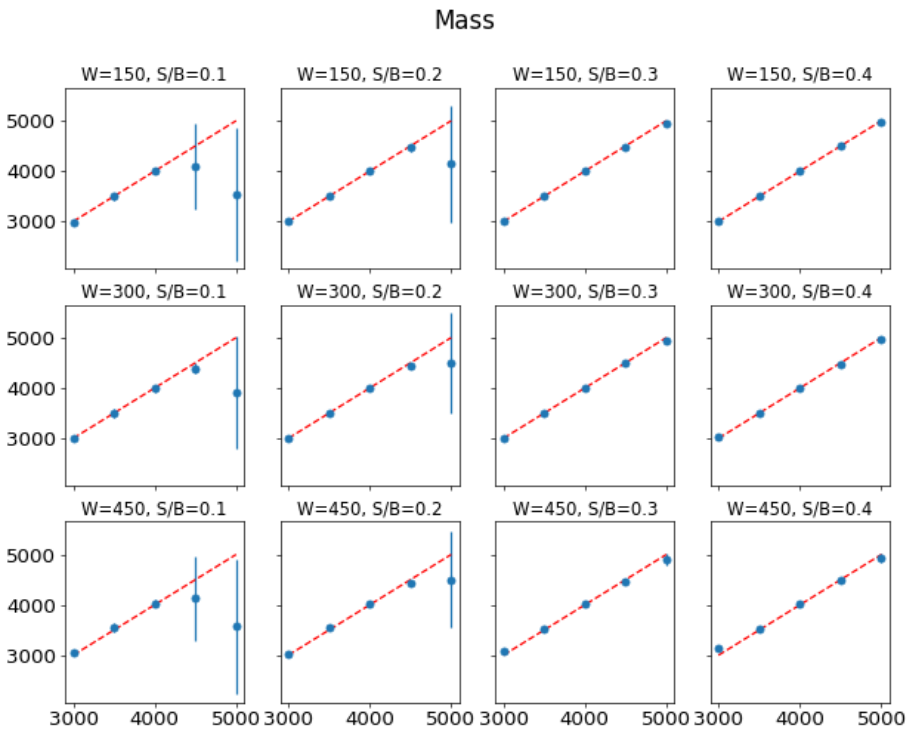


FIGURE D.2: Parametric approach (five-parameter background fit): Linearity plots for the mass of the signal; each plot corresponds to indicated width and R pair of values. The values of R are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red line is plotted for reference.

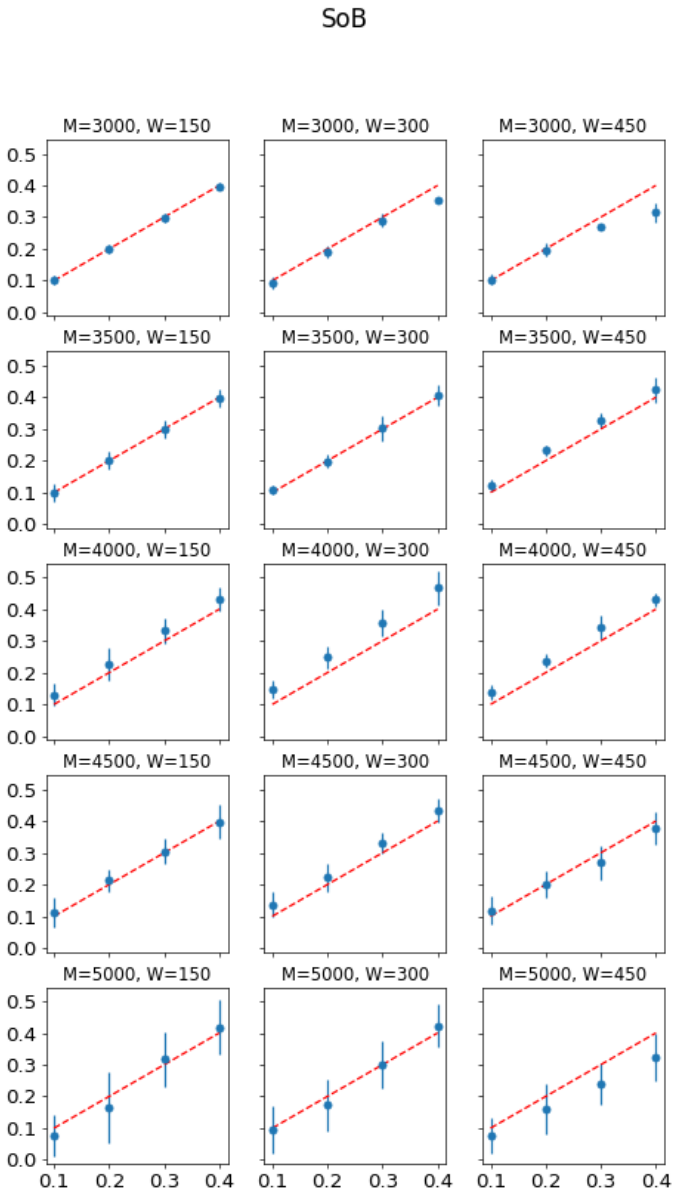


FIGURE D.3: Parametric approach (five-parameter background): Linearity plots for the R of the signal; each plot corresponds to indicated mass and width pair of values. The values of the width are the same for each plot column, and those of the mass are fixed through each plot row. Points and error bars (means and rms) are calculated from the distribution of extracted width values. A dashed red $x = y$ line is plotted for reference.

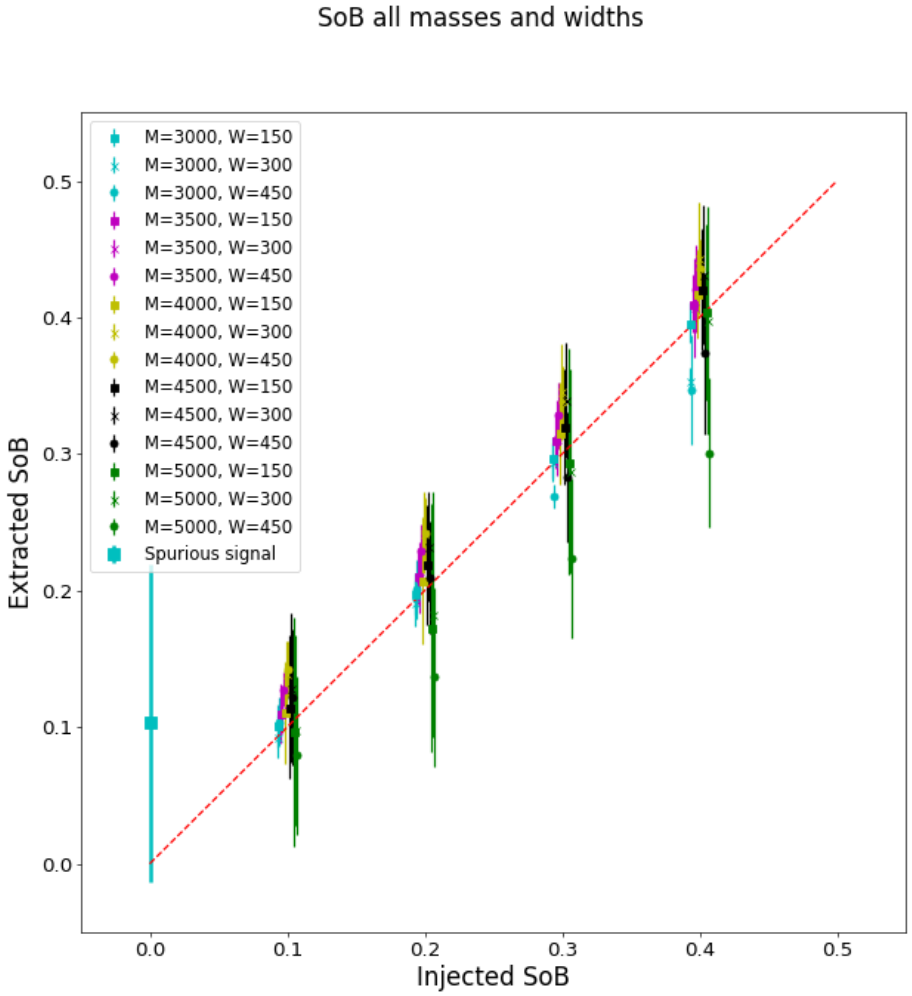


FIGURE D.4: Parametric approach (five-parameter background): Linearity plots for the R for all masses and widths of the signal and spurious detection. Values of R are from 0.1 to 0.4 in intervals of 0.1; points appear slightly shifted in the horizontal axis for better visibility. A dashed red line is plotted for reference.

Appendix E

A First Implementation of Generalized Additive Models in MATLAB

This appendix documents the result of an industrial secondment performed during twelve weeks at The Mathworks, Inc., within the training program of the AMVA4New-Physics Innovative Training Network, a Marie-Sklodowska Curie Action of the European Union. The company develops scientific software and it is known for their flagship packages MATLAB and SIMULINK. In the following we present several aspects of an implementation of Generalized Additive Models for the Statistics and Machine Learning Toolbox of MATLAB.

E.1 Introduction

Generalized Additive Models (GAMs) are a class of models that combine features from Generalized Linear Models (GLMs) and Additive Models. This class of models provides a more flexible tool than GLMs, giving a smooth multidimensional model while retaining interpretability on the predictors.

Posed about three decades ago by Hastie and Tibshirani [176, 177], GAMs have been long studied, and several implementations are available. An implementation by Hastie and Tibshirani [178] and another one by Simon Wood [179, 180] are available in R; more recently, an implementation in Python appeared [181]. The present work is the first step towards making GAMs part of the MATLAB Statistics and Machine Learning Toolbox.

In this document we describe some of the key concepts and algorithms relevant to GAMs as well as some details in our implementation. We then show some tests in a few scenarios.

E.2 GAMs and the Backfitting algorithm

An easy way to illustrate GAMs is starting from GLMs. Given a training data set with P predictor vectors $\{X_1, ..., X_P\} = X$, and response vector Y, the task of fitting a GLM consists in finding values for a coefficient matrix β of size $N \times P$. (Assuming the length of each predictor and of the response, or number of observations to be N.) A *link function* $g(\cdot)$ relates the linear model on the covariates, and the response:

$$g(Y) = s_0 + \sum_j \beta_j X_j, \tag{E.1}$$

where s_0 is an offset term. There exists a set of commonly-used link functions (e.g. logit, identity, log) from which one has to decide taking into account the type of problem to be solved.

GAMs incorporate the additive feature by using one smoothing function s per covariate:

$$g(Y) = s_0 + \sum_j s_j(X_j)\,. \tag{E.2}$$

Fitting a GAM consists in finding suitable smooth functions s, which will be obtained using a non-parametric procedure.

Given an Additive Model like the one on the RHS of equation (E.2), it is possible to estimate each of the smoothers in an iterative procedure, by defining partial residuals:

$$R_k = Z - s_0 - \sum_{j\neq k} s_j(X_j)\,, \tag{E.3}$$

for a response variable Z. That way, smooth functions are obtained for all residuals, and the procedure is iterated until some convergence criterion is satisfied. This algorithm is known as the *backfitting* algorithm.

A modification of backfitting, known as the *local scoring* algorithm is used to fit GAMs. We initialize $s_0 = g(E(Y))$, and $s_k^{(0)} = 0$ for all k. Then the procedure iterates in m, defining the quantities[1]:

$$\eta^{(m-1)} = s_0 + \sum_j s_j^{(m-1)}(X_j) \tag{E.4a}$$

$$\mu^{(m-1)} = g^{-1}\left(\eta^{(m-1)}\right) \tag{E.4b}$$

$$W = \left(V^{(m-1)}\right)^{-1}\left(\frac{\partial\mu}{\partial\eta}\right)^2_{(m-1)} \tag{E.4c}$$

$$Z = \eta^{(m-1)} + \left(Y - \mu^{(m-1)}\right)\left(\frac{\partial\mu}{\partial\eta}\right)_{(m-1)} \tag{E.4d}$$

where the entries of $V^{(m-1)}$ are the variances of Y at each entry of $\mu^{(m-1)}$. We use the backfitting algorithm to fit an Additive Model to Z with weights given by W, to estimate the smooth functions $s_k^{(m)}$ from the partial residuals as they appear in eq. (E.3).

There are several choices of smoothers one can use. In this work, the *running lines* and *loess* smoothers are used, but most implementations also include estimations via *splines*, which is left for future work. Both running lines and loess are based on performing multiple local (respectively linear and polynomial) least-squares regressions in sliding windows to provide a smooth one-dimensional model, or an estimate of a residual R_k in our use case.

[1] Operations in the following equations (addition, exponentiation, multiplication, differentiation) on column vectors are element-wise.

E.3 Implementation

Two utilities are available for fitting Generalized Linear Models in the Statistics and Machine Learning Toolbox, namely `glmfit` and the more recent `fitglm`. Since GAMs are an extension of GLMs, we borrow several functionalities from both GLM packages in our code.

E.3.1 The GeneralizedAdditiveModel class

An object of the `GeneralizedAdditiveModel` class contains methods (`fitgam`, `predict`) and properties (`grid` and smoothers `S`), that are relevant to fit GAMs given a training data set, and to make predictions for test data sets.

In our implementation, the smoothing functions are stored using cell arrays that contain function handles. Since the local least-squares regressions for a particular residual result in a piece-wise function defined at the neighborhood of the input points, we store a function handle containing two coefficients per regression.

The number of regressions performed (or handles stored) M_j is at most the number of points in the covariate (N), where the number decreases in the presence of repeated points. By creating a grid in the training step, before starting the backfitting iterations, we run once a sorting algorithm in each covariate that is useful for two purposes: identifying the neighboring and repeated points for each local regression in the smoother, and for matching points in a testing data set with their corresponding handle (using the sorting indices) as we discuss below.

A few other options regarding the storage of smoothers and matching with testing points were considered but discarded in favor of our proposal:

- Create function handles that return a zero value everywhere except in the interval surrounding the regression point. This would have the apparent advantage of avoiding the matching in the prediction step described in detail below, as the functions will return non-zero values for an input point only in the appropriate interval. However, this is inconvenient for at least two reasons: firstly, each function handle would have to store the two extrema to identify the non-trivial interval, and since for every two consecutive intervals the first interval's upper bound is the lower bound of the second, we would be storing repeated values. Secondly, using such functions implies running each of the M_j handles in each of the points for a covariate in the test set that will, in general, return zero in most cases. By having a grid constructed as we propose, both the number of values for the intervals is reduced and the smoothing functions are run at points within the relevant interval.

- Not storing a grid but matching the smoother handles in the prediction step by using the training covariates. This implies storing as many smoothers as input values in the covariate matrix and in the same order. The `predict` method would take the input covariates and the test covariates (X, X^{test}), and an array of smoother handles. For each point in every test covariate, the the method would have to find the index of the nearest neighbor in the corresponding training covariate to retrieve the relevant smoother. A typical use of a model like GAMs involves running the training method fewer times than the predicting one. Therefore, we choose to reduce the amount of operations in the `predict` method as much as possible, moving them to the training step.

Below we describe the properties and methods in the class.

The grid property

Given P covariates $\{X_1, ..., X_P\}$ each of length N, the `grid` property is a cell array that stores P one-dimensional grids, i.e. one per covariate. Each 1D grid is an ordered array constructed from a specific X_j that contains the extrema of the covariate and at most $N-1$ values half-way between each ordered pair of unique points from X_j. If we take M_j to be the length of the 1D grid for covariate X_j, then $M_j \leq N+1$, where the inequality holds in the case of repeated values in X_j.

The S property

Smooth functions of the Additive Model are stored in a cell array `S` that contains P cell arrays, one per covariate. The j-th cell array contains $M_j - 1$ function handles, corresponding to intervals from the j-th 1D grid; the intervals are bounded by each pair of consecutive values in that 1D grid.

The cell arrays contained in `S` are ordered to match values on the grid, i.e. given a test set of covariates X^{test}, for each k-th observation on the j-th covariate one should find the index `i` in the 1D grid that satisfies

$$\texttt{grid\{j\}(i)} < X_{kj}^{\text{test}} \leq \texttt{grid\{j\}(i+1)}. \tag{E.5}$$

On average checking the condition above requires $M_j/2$ operations as the array stored in `grid{j}(i)` is sorted. Then, we would apply the smoothing function stored in `S{j}{i}` to X_{kj}^{test}.

The fitgam method

This method takes as input a training data set: X and the responses Y; and a set of arguments detailed in the code. The method returns an instance of the `GeneralizedAdditiveModel` class where the `S` and `grid` properties have been set. A property containing the inverse of the link function `ilinkfun` is also stored, which is used in the `predict` method (see below).

Below an outline of the tasks that this method performs:

Copied from or inspired by glmfit

- Parse the arguments via `parseArgs`.
- Set handles for the link function, its derivative, and its inverse via `stattestlink`.
- Initialize the μ and η vectors via `startingVals`.
- Variance estimation via `getGLMVariance`.

The gridder function This function constructs and returns a cell array of 1D grids from a given $N \times P$ matrix of covariates, to set the `grid` property. It follows the idea described in E.3.1 and uses the `unique` command to sort the values and create the 1D grids. Arrays of indices to map the covariates to the 1D grids and vice versa, `igrid` and `idata` respectively, are also returned.

The local scoring algorithm Before the local scoring iterations, we initialize a matrix for the residuals, `Rold` and a cell array for the handles in `S`. At each iteration, we set the values of the weights W, update the value of the transformed response Z and call the `backfit` function (described below), to update η, the residual matrix and the smoothers in `S`, as described in section E.2.

The predict method

This method takes as an input the trained model (an instance of the GeneralizedAdditiveModel class) and a matrix of P test covariate predictors $X^{\text{test}} = \{X_1^{\text{test}}, ..., X_P^{\text{test}}\}$, and returns the model evaluated on each of the input points, i.e. a predicted response of the same length as each of the covariates.

The method performs two tasks, matching the values of X^{test} in the grid and applying the smoothers. For the first task the method loops on each test predictor, and for the every entry therein finds the index that satisfies eq. (E.5), with the help of the `find` command. With the relevant indices, it is possible to construct the additive model finding the smoothers in `S`.

E.3.2 Algorithms: Backfitting and Boost

Backfitting

The `backfit` function implements the backfitting algorithm that was previously described in section E.2. It takes as an input the input covariates X and a response Y (that is Z in the case of local scoring), the type of smooth function to be used, (running line or loess), the weights, the fitting algorithm, the width of the smoother window, a successive over-relaxation parameter[2] ω, the indices `igrid` and `idata` (see the `gridder` function), and the residual matrix and `S` from previous iterations.

This function iterates P times (runs the smoother once per covariate) and returns updated values of the function handles (`S`), the Y (or Z) estimates at the training set points and the updated residual matrix.

Boost

We have written an implementation of this algorithm from [183] which shares several similarities with backfitting. Boost is intended to be used with regression trees, which is a part of the code that is still under testing, see E.4.3.

E.3.3 Smoothers: running lines, loess

Running Lines

The `runline` function takes one covariate, a response, weights, the width of the regression window and the `igrid` and `idata` indices (see the `gridder` function). The regression is performed using the backslash operator on the set of points that are near (within the provided window) each unique point in the covariate, labeled by the `igrid` indices. This way, the number of regressions performed is M_j. The function returns a cell array of function handles with stored coefficients, and estimates of the smooth functions mapped back to each point of the input covariate with the `idata` indices. Such mapping has to be performed to return a residual vector of the appropriate length for the next backfitting iteration.

[2] This parameter wasn't discussed in section E.2 but an explanation is available in [182].

Loess

The `loess` function takes the same input parameters as `runline` and also returns a cell array of handles with corresponding coefficients and the estimates at the covariate points. The local regression is also performed a set of points close to each unique point in the covariate, but the regression weights are multiplied by a tri-cube function.

Trees

Under test. See E.4.3.

E.4 Tests

We have tested the functionality of our code with two link functions for classification and regression: logit, and identity respectively. Other links (e.g. log, probit, loglog) are left for future work.

E.4.1 Classification using a logistic model

Our first test example consists in performing binary classification in a data set. To realize this model, we use the *logit* link function defined as

$$g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right) . \tag{E.6}$$

From that prescription one can derive the explicit expressions for equations (E.4) and apply the local scoring algorithm.

Two-dimensional Gaussian distributions

The training data set is constructed sampling 2000 points from two Gaussian distributions in two dimensions (1000 points each) for the covariates, labelling the response with zero and one the points coming from either Gaussian distribution. An illustration of the resulting sampled covariates and labels appears in fig. E.1.

Two tests have been performed, changing only the smoother from running lines to loess in the training. We indicate the rest of the arguments in the training method: use a binomial distribution, a logit link, a fixed amount of (20) iterations, and the backfit algorithm.

As a testing data set we sample another set from two Gaussian distributions with same parameters. The result of the predict method can be visualized and assessed using the true labels.

In figures E.2 and E.3 we see plots for running lines and loess respectively. The AUC figure of merit is the area under the ROC curve (True Positive Rate vs. False Positive Rate), constructed by using different working points as a criterion to separate one class from the other. The greater the AUC, the better discriminant power the classifier has. In the tests the two smoothers lead to a similar AUC.

The script that makes the tests above is `tgam_logit`.

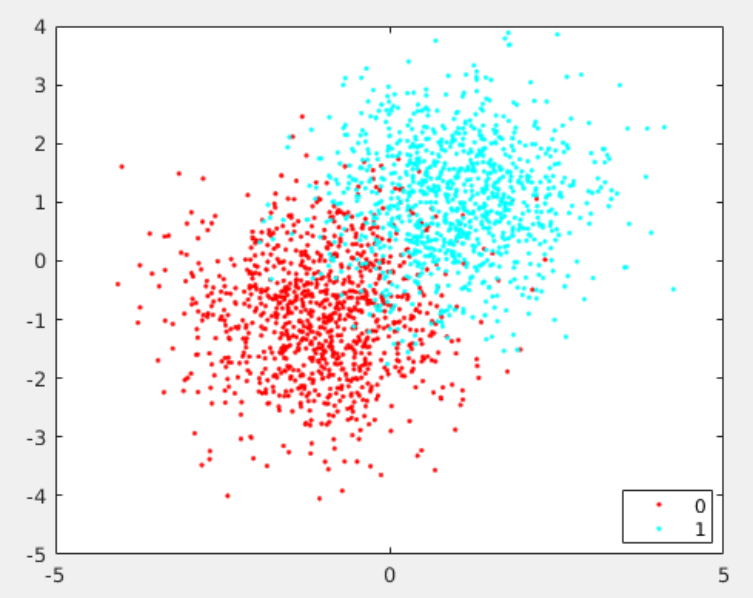


FIGURE E.1: Two 2-dimensional Gaussians

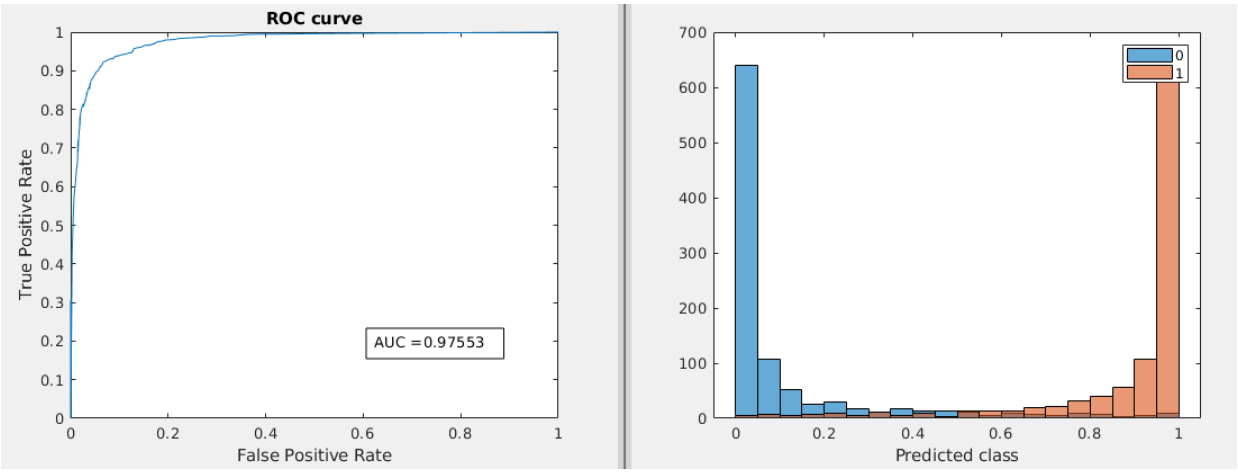


FIGURE E.2: Prediction on a test data set using running lines smoother

Repeated indices

In order to make sure that the case of repeated points was handled properly, we've written the `tgam_logit_repeated` script. There we construct a simple data set with many repeated covariate points. We then inspected manually that the repeated points were not taken into account to handle the grid, and the unique ones were used appropriately to select neighboring points for the regression in `loess` and `runline`.

Physics data set

As an example we've used a data set from a High-Energy Physics simulation. Very broadly, we would like to discriminate between signal and background classes of events (observations) that correspond to different physical processes. Our data set contains seven continuous variables that are physical observables, as measured in a collider detector, and about 10 thousand observations for each class. The most traditional approach in the domain consists in performing a univariate analysis, selecting

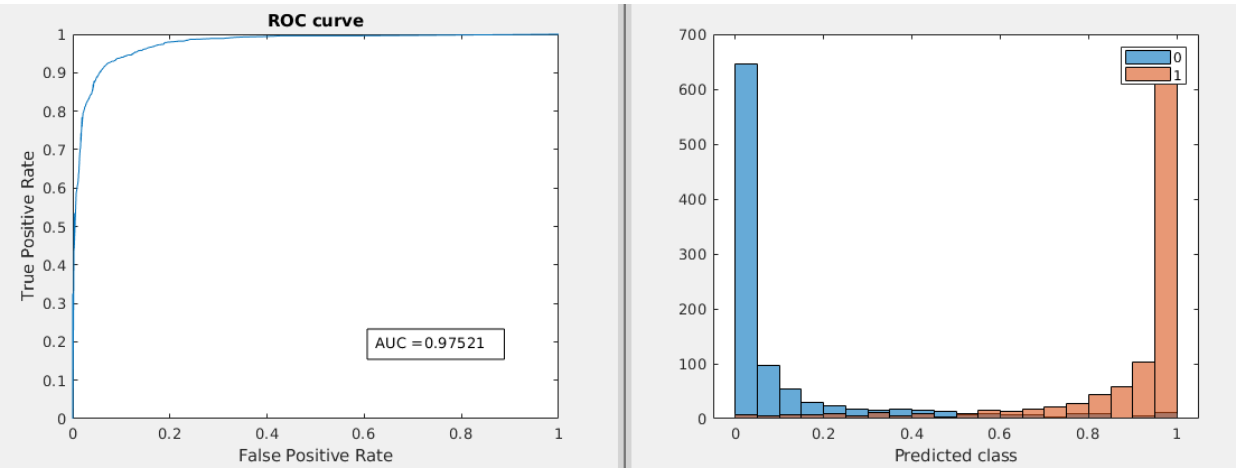


FIGURE E.3: Prediction on a test data set using loess smoother

an discriminant interval in which the signal is more prominent; we use GAMs in the case of two and five dimensions.

For constructing both training and testing data sets, we take 1000 events of each class. We pre-process the variables centering the values at the mean and dividing by the standard deviation. We will use GAMs in the two scenarios used above in E.4.1, with the same argument, and varying the smoother. Also, for each of the smoothers we train the GAM with two variables and five variables.

The resulting plots from the two variable case using the running lines and loess smoothers are respectively in figures E.4 and E.5 respectively. In the two variable case, the loess smoother leads to a slightly better classifier (as indicated by the AUC).

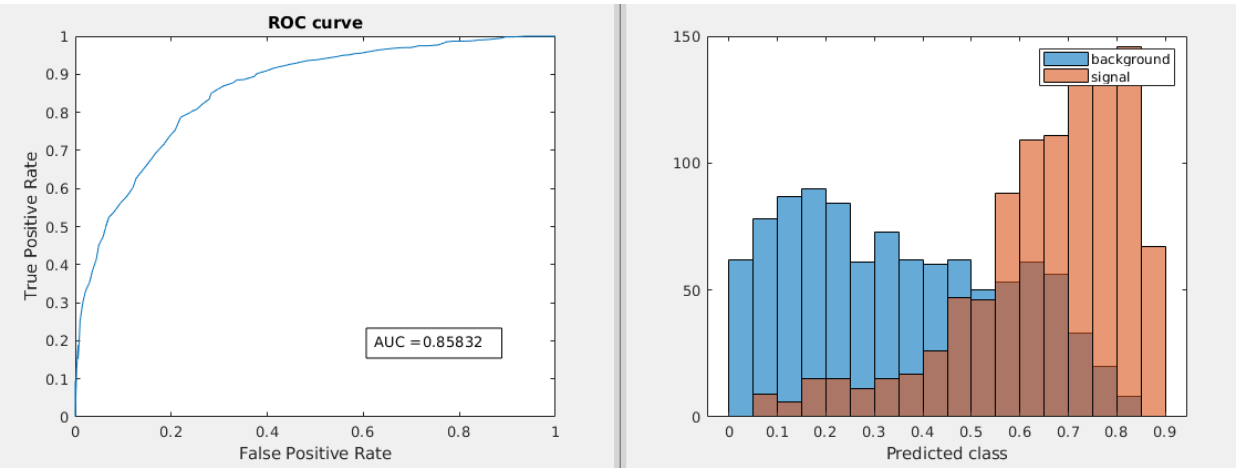


FIGURE E.4: Prediction on a physics test data set (2 variables) using running lines smoother

In the case of five variables, the corresponding plots are E.6 and E.7. Here it is also the case that loess performs better than running lines. We can also observe that using more variables makes a significant increase in the performance of the classifier in the test data set.

The script that produces the tests above is the `tgam_logit_phys`.

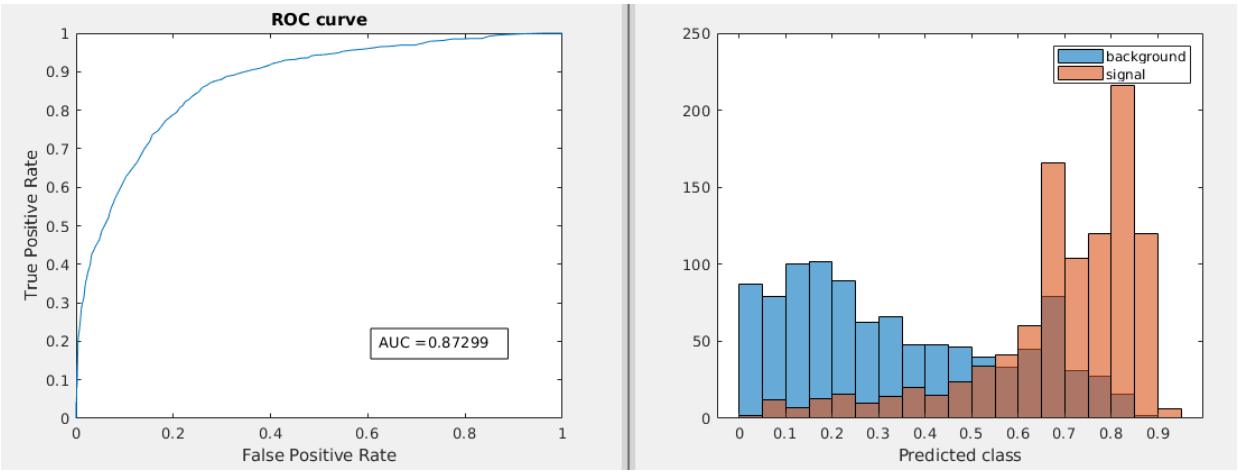


FIGURE E.5: Prediction on a physics test data set (2 variables) using loess smoother

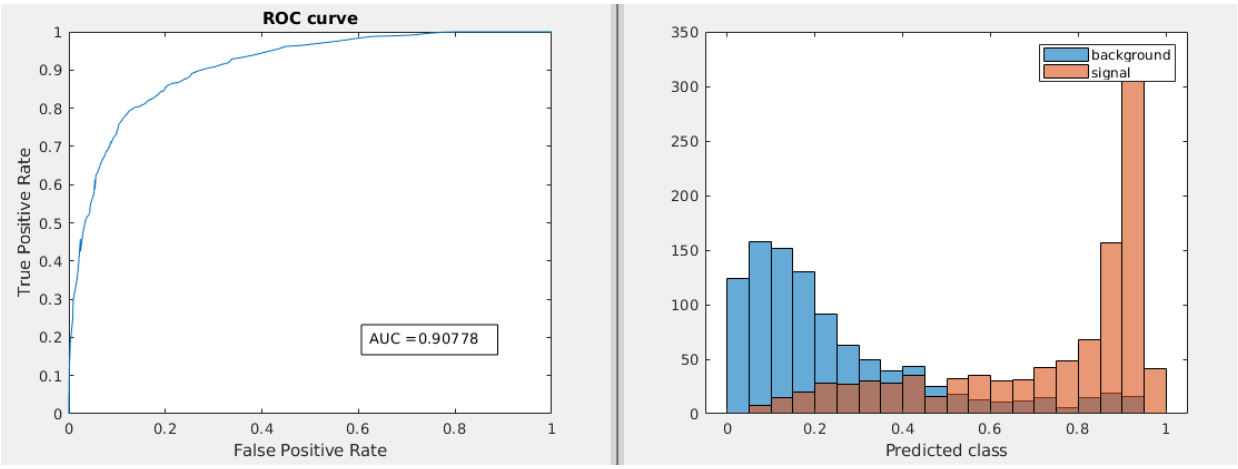


FIGURE E.6: Prediction on a physics test data set (5 variables) using running lines smoother

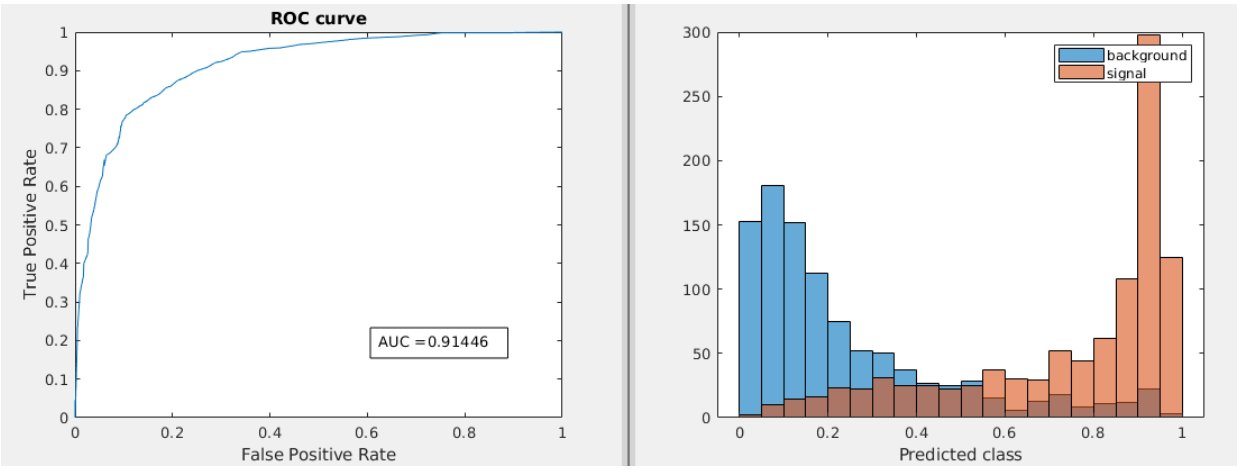


FIGURE E.7: Prediction on a physics test data set (5 variables) using loess smoother

E.4.2 Regression in a two dimensional step function (identity link)

With a more simple model prescription, we can use GAMs for regressing functions. In that case we make use of the *identity* link $g(\mu) = \mu$. For testing this scenario, we generate a four-step function in two dimensions, as depicted in figure E.8; steps have response values around 1, 2, 3, and 4; taking 400 points on each step. Each step contains a small Gaussian noise component of variance 0.2 around the corresponding response constant value. For testing we use a similar data set with 300 points on each step.

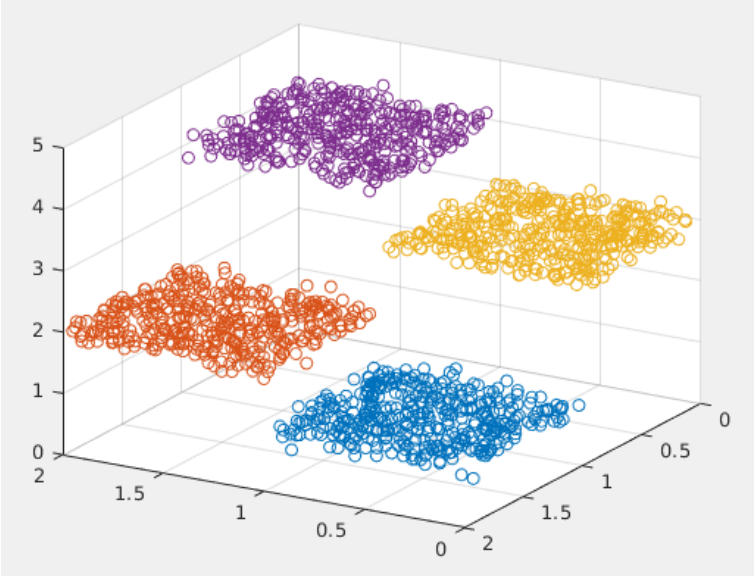


FIGURE E.8: 2-dimensional, four-step function.

As an input to the fitgam method, we specify the normal distribution, the identity link, the backfitting algorithm and the width of the smoother window (0.1). We test with both the running lines and loess smoothers as before and plot the predicted values for the testing data set. We also include box plots that illustrate the dispersion of the predicted values versus the true step value. Figures E.9 and E.10 contain plots for running lines and loess respectively.

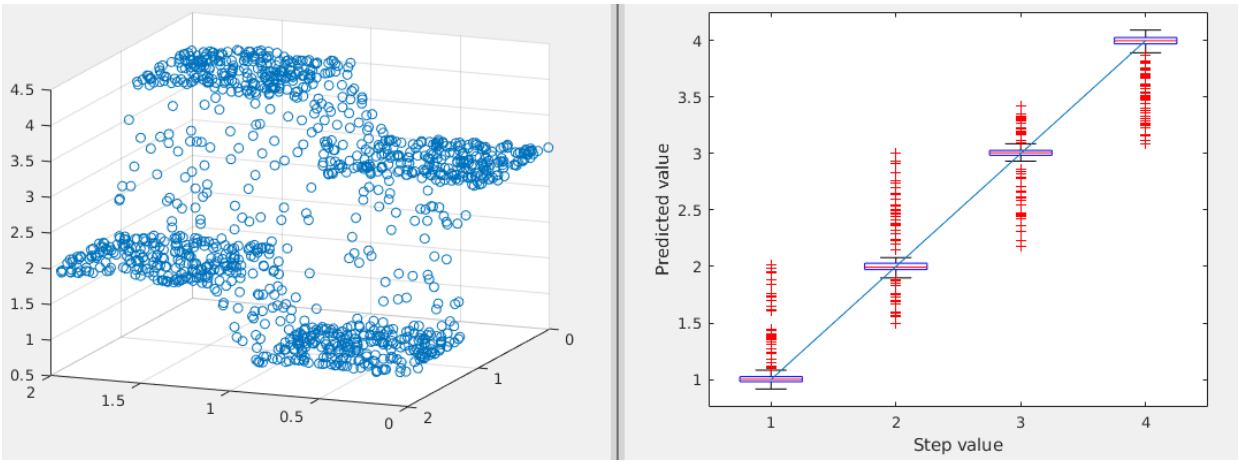


FIGURE E.9: Regression using running lines on a two-dimensional step function (left) and box plot of predicted values (right).

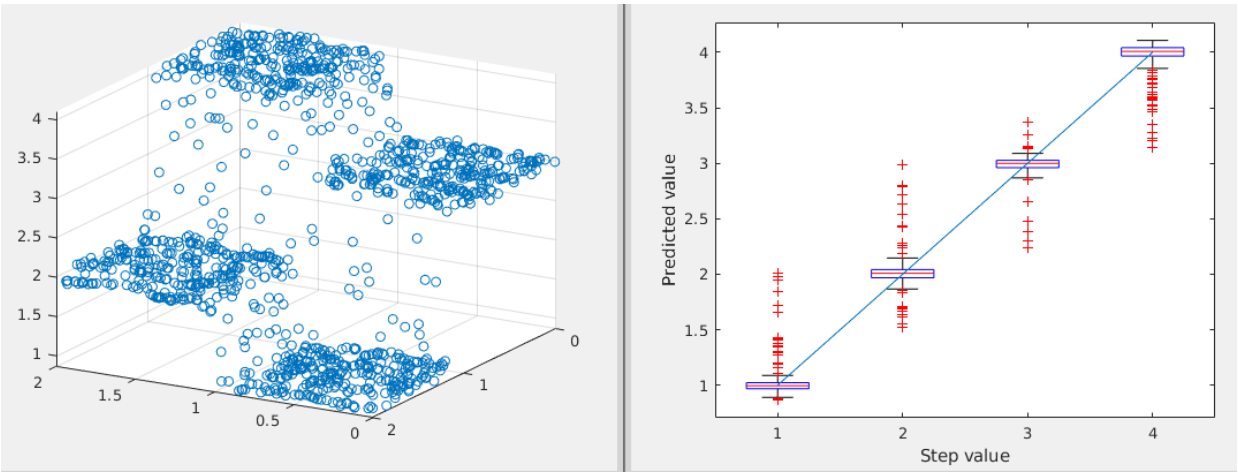


FIGURE E.10: Regression using loess on a two-dimensional step function (left) and box plot of predicted values (right).

The script that makes the tests displayed above is `tgam_id`.

E.4.3 Pieces of code under test (not fully functional)

The implementation of a boosting algorithm using regression trees is incomplete, as it can be seen in the code[3] in the local scoring iterations in the `fitgam` method, and the `boost` function. One can run some test on the training set in the code, but in order to extract the smoothers some approach different to the grid plus function handles (as used for backfitting) has to be devised.

[3]The code is proprietary by the company.

Bibliography

1. Weinberg, S. A Model of Leptons. *Phys. Rev. Lett.* **19,** 1264–1266 (21 1967).
2. Glashow, S. L. Partial Symmetries of Weak Interactions. *Nucl. Phys.* **22,** 579–588 (1961).
3. Goldstone, J., Salam, A. & Weinberg, S. Broken Symmetries. *Phys. Rev.* **127,** 965–970 (3 1962).
4. ALTARELLI, G. The Standard model of particle physics. arXiv: hep-ph/0510281 [hep-ph] (2005).
5. Tanabashi, M. *et al.* Review of Particle Physics. *Phys. Rev. D* **98,** 030001 (3 Aug. 2018).
6. Halzen, F. & Martin, A. D. *QUARKS AND LEPTONS: AN INTRODUCTORY COURSE IN MODERN PARTICLE PHYSICS* ISBN: 0471887412, 9780471887416 (1984).
7. *Introduction to High Energy Physics* 4th ed. doi:10.1017/CBO9780511809040 (Cambridge University Press, 2000).
8. Wu, C.-S., Ambler, E., Hayward, R., Hoppes, D. & Hudson, R. P. Experimental test of parity conservation in beta decay. *Physical review* **105,** 1413 (1957).
9. Salam, A. Weak and Electromagnetic Interactions. *Conf. Proc.* **C680519,** 367–377 (1968).
10. Cabibbo, N. Unitary Symmetry and Leptonic Decays. *Phys. Rev. Lett.* **10,** 531–533 (12 1963).
11. Kobayashi, M. & Maskawa, T. CP-Violation in the Renormalizable Theory of Weak Interaction. *Progress of Theoretical Physics* **49,** 652–657. ISSN: 0033-068X (Feb. 1973).
12. Abe, F. *et al.* Observation of top quark production in $\bar{p}p$ collisions. *Phys. Rev. Lett.* **74,** 2626–2631 (1995).
13. Abachi, S. *et al.* Search for high mass top quark production in $p\bar{p}$ collisions at $\sqrt{s} = 1.8$ TeV. *Phys. Rev. Lett.* **74,** 2422–2426 (1995).
14. Aad, G. *et al.* Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett.* **B716,** 1–29 (2012).
15. Chatrchyan, S. *et al.* Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys. Lett.* **B716,** 30–61 (2012).
16. Hanneke, D., Hoogerheide, S. F. & Gabrielse, G. Cavity Control of a Single-Electron Quantum Cyclotron: Measuring the Electron Magnetic Moment. *Phys. Rev.* **A83,** 052122 (2011).
17. Canetti, L., Drewes, M. & Shaposhnikov, M. Matter and Antimatter in the Universe. *New J. Phys.* **14,** 095012 (2012).

18. Bertone, G., Hooper, D. & Silk, J. Particle dark matter: Evidence, candidates and constraints. *Phys. Rept.* **405,** 279–390 (2005).
19. Feng, J. L. Dark Matter Candidates from Particle Physics and Methods of Detection. *Ann. Rev. Astron. Astrophys.* **48,** 495–545 (2010).
20. Bertone, G. & Hooper, D. History of dark matter. *Rev. Mod. Phys.* **90,** 045002 (4 2018).
21. Fukuda, Y. *et al.* Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.* **81,** 1562–1567 (1998).
22. Schechter, J. & Valle, J. W. F. Neutrino masses in SU(2) ⊗ U(1) theories. *Phys. Rev. D* **22,** 2227–2235 (9 1980).
23. Martin, S. P. A Supersymmetry primer. [Adv. Ser. Direct. High Energy Phys.18,1(1998)], 1–98 (1997).
24. Pati, J. C. & Salam, A. Lepton number as the fourth "color". *Phys. Rev. D* **10,** 275–289 (1 1974).
25. Senjanovic, G. & Mohapatra, R. N. Exact left-right symmetry and spontaneous violation of parity. *Phys. Rev. D* **12,** 1502–1505 (5 1975).
26. Mohapatra, R. N. Neutrino masses and mixings in gauge models with spontaneous parity violation. *Phys. Rev. D* **23,** 165–180 (1 1981).
27. Gunion, J. F., Dawson, S., Haber, H. E. & Kane, G. L. *The Higgs hunter's guide* In the second printing (1990) by Perseus Books in the collection Frontiers in physics, no 80, a number of errors and omissions are corrected and the references at the end of each chapter are updated. A paperback reprint of the 1990 edition has been published in 2000. https://cds.cern.ch/record/425736 (Brookhaven Nat. Lab., Upton, NY, 1989).
28. Barbier, R. *et al.* R-parity violating supersymmetry. *Physical Review* **420,** 1–202 (2005).
29. Lefèvre, C. *The CERN accelerator complex. Complexe des accélérateurs du CERN* 2008. https://cds.cern.ch/record/1260465.
30. Evans, L. & Bryant, P. LHC machine. *Journal of instrumentation* **3,** S08001 (2008).
31. Collaboration, T. A. *ATLAS Luminosity public results webpage.* 2019. https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2.
32. Pequenao, J. *Computer generated image of the whole ATLAS detector* 2008. https://cds.cern.ch/record/1095924.
33. ATLAS magnet system: Technical design report (1997).
34. Goodson, J. J. *Search for supersymmetry in states with large missing transverse momentum and three leptons including a Z-boson* PhD thesis (SUNY, Stony Brook, 2012-05-11).
35. Pequenao, J. *Computer generated image of the ATLAS inner detector* 2008. https://cds.cern.ch/record/1095926.
36. Capeans, M *et al. ATLAS Insertable B-Layer Technical Design Report* tech. rep. CERN-LHCC-2010-013. ATLAS-TDR-19 (2010). https://cds.cern.ch/record/1291633.
37. Garelli, N. Performance of the ATLAS Detector in Run-2. *EPJ Web Conf.* **164,** 01021. 10 p (2017).

38. Pequenao, J. *Computer Generated image of the ATLAS calorimeter* 2008. https://cds.cern.ch/record/1095927.
39. Aad, G. *et al.* The ATLAS Experiment at the CERN Large Hadron Collider. *JINST* **3,** S08003 (2008).
40. Aad, G. *et al.* Drift Time Measurement in the ATLAS Liquid Argon Electromagnetic Calorimeter using Cosmic Muons. *Eur. Phys. J.* **C70,** 755–785 (2010).
41. Collaboration, A. *Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter* tech. rep. CERN-LHCC-2017-019. ATLAS-TDR-028 (CERN, Geneva, 2017). https://cds.cern.ch/record/2285583.
42. Aad, G. *et al.* Commissioning of the ATLAS Muon Spectrometer with Cosmic Rays. *Eur. Phys. J.* **C70,** 875–916 (2010).
43. Bird, I *et al. Update of the Computing Models of the WLCG and the LHC Experiments* tech. rep. CERN-LHCC-2014-014. LCG-TDR-002 (2014). https://cds.cern.ch/record/1695401.
44. Sabato, G *et al.* ATLAS fast physics monitoring: TADA. *Journal of Physics: Conference Series* **898,** 092015 (Oct. 2017).
45. Aaboud, M. *et al.* A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment. *Eur. Phys. J.* **C79,** 120 (2019).
46. Elsing, M, Goossens, M, Nairz, A & Negri, G. The ATLAS Tier-0: Overview and operational experience. *J. Phys.: Conf. Ser.* **219,** 072011 (2010).
47. Aaboud, M. *et al.* Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton-proton collision data at $\sqrt{s} = 13$ TeV. *Submitted to: Eur. Phys. J.* arXiv: 1902.04655 [physics.ins-det] (2019).
48. Aad, G. *et al.* Muon reconstruction performance of the ATLAS detector in protonproton collision data at $\sqrt{s}$ =13 TeV. *Eur. Phys. J.* **C76,** 292 (2016).
49. Aaboud, M. *et al.* Measurement of the photon identification efficiencies with the ATLAS detector using LHC Run-1 data. *Eur. Phys. J.* **C76,** 666 (2016).
50. Cacciari, M., Salam, G. P. & Soyez, G. The anti-k_t jet clustering algorithm. *JHEP* **04,** 063 (2008).
51. *Optimisation of the ATLAS b-tagging performance for the 2016 LHC Run* tech. rep. ATL-PHYS-PUB-2016-012 (CERN, Geneva, 2016). http://cds.cern.ch/record/2160731.
52. *Reconstruction, Energy Calibration, and Identification of Hadronically Decaying Tau Leptons in the ATLAS Experiment for Run-2 of the LHC* tech. rep. ATL-PHYS-PUB-2015-045 (CERN, Geneva, 2015). https://cds.cern.ch/record/2064383.
53. Alioli, S., Nason, P., Oleari, C. & Re, E. A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX. *Journal of High Energy Physics* **2010,** 43. ISSN: 1029-8479 (2010).
54. Frixione, S., Nason, P. & Oleari, C. Matching NLO QCD computations with Parton Shower simulations: the POWHEG method. *JHEP* **11,** 070 (2007).
55. Sjostrand, T., Mrenna, S. & Skands, P. Z. PYTHIA 6.4 Physics and Manual. *JHEP* **05,** 026 (2006).
56. Sjostrand, T., Mrenna, S. & Skands, P. Z. A Brief Introduction to PYTHIA 8.1. *Comput. Phys. Commun.* **178,** 852–867 (2008).

57. *ATLAS Run 1 Pythia8 tunes* tech. rep. ATL-PHYS-PUB-2014-021 (CERN, Geneva, 2014). http://cds.cern.ch/record/1966419.
58. Ball, R. D. *et al.* Parton distributions for the LHC Run II. *JHEP* **04,** 040 (2015).
59. Lange, D. J. The EvtGen particle decay simulation package. *Nucl. Instrum. Meth.* **A462,** 152–155 (2001).
60. Skands, P. Z. Tuning Monte Carlo Generators: The Perugia Tunes. *Phys. Rev.* **D82,** 074018 (2010).
61. Gleisberg, T. *et al.* Event generation with SHERPA 1.1. *JHEP* **02,** 007 (2009).
62. Guzzi, M. *et al.* CT10 parton distributions and other developments in the global QCD analysis. arXiv: 1101.0561 [hep-ph] (2011).
63. Aad, G. *et al.* Search for New Phenomena in Dijet Angular Distributions in Proton-Proton Collisions at $\sqrt{s} = 8$ TeV Measured with the ATLAS Detector. *Phys. Rev. Lett.* **114,** 221802 (2015).
64. Nagy, Z. Three-Jet Cross Sections in Hadron-Hadron Collisions at Next-To-Leading Order. *Phys. Rev. Lett.* **88,** 122003 (12 2002).
65. Nagy, Z. Next-to-leading order calculation of three-jet observables in hadron-hadron collisions. *Phys. Rev. D* **68,** 094002 (9 2003).
66. Catani, S. & Seymour, M. H. A General algorithm for calculating jet cross-sections in NLO QCD. *Nucl. Phys.* **B485.** [Erratum: Nucl. Phys.B510,503(1998)], 291–419 (1997).
67. Jones, S. D. *The ATLAS Electron and Photon Trigger* tech. rep. ATL-DAQ-PROC-2017-040. 4 (CERN, Geneva, 2017). doi:10.1088/1742-6596/1085/4/042001. https://cds.cern.ch/record/2290123.
68. Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach* 3rd. ISBN: 0136042597, 9780136042594 (Prentice Hall Press, Upper Saddle River, NJ, USA, 2009).
69. Settles, B. *Active learning literature survey* tech. rep. (2010).
70. Zhu, X. *Semi-Supervised Learning Literature Survey* tech. rep. 1530 (Computer Sciences, University of Wisconsin-Madison, 2005). http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.
71. Paganini, M., de Oliveira, L. & Nachman, B. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev.* **D97,** 014021 (2018).
72. Di Sipio, R., Faucci Giannelli, M., Ketabchi Haghighat, S. & Palazzo, S. DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC. arXiv: 1903.02433 [hep-ex] (2019).
73. Collaboration, C. Search for new resonances in the diphoton final state in the mass range between 80 and 115 GeV in pp collisions at $\sqrt{s} = 8$ TeV (2015).
74. Baldi, P., Sadowski, P. & Whiteson, D. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.* **5,** 4308 (2014).
75. LeCun, Y., Bengio, Y. & Hinton, G. E. Deep learning. *Nature* **521,** 436–444 (2015).
76. Denby, B. H. Neural Networks and Cellular Automata in Experimental High-energy Physics. *Comput. Phys. Commun.* **49,** 429–448 (1988).
77. Roe, B. P. *et al.* Boosted decision trees, an alternative to artificial neural networks. *Nucl. Instrum. Meth.* **A543,** 577–584 (2005).

78. Bhat, P. C. Multivariate Analysis Methods in Particle Physics. *Ann. Rev. Nucl. Part. Sci.* **61,** 281–309 (2011).
79. Bhat, P. C., Prosper, H. B. & Snyder, S. S. Bayesian analysis of multi-source data. *Physics Letters B* **407,** 73–78 (1997).
80. Albertsson, K. *et al.* Machine Learning in High Energy Physics Community White Paper. *J. Phys. Conf. Ser.* **1085,** 022008 (2018).
81. Cranmer, K. *Practical Statistics for the LHC* in *Proceedings, 2011 European School of High-Energy Physics (ESHEP 2011): Cheile Gradistei, Romania, September 7-20, 2011* [,247(2015)] (2015), 267–308. doi:10.5170/CERN-2015-001.247,10.5170/CERN-2014-003.267. arXiv: 1503.07622 [physics.data-an].
82. Murphy, K. P. *Machine Learning: A Probabilistic Perspective* ISBN: 0262018020, 9780262018029 (The MIT Press, 2012).
83. Hastie, T., Tibshirani, R. & Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition* ISBN: 9780387848570. http://www.worldcat.org/oclc/300478243 (Springer, 2009).
84. Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* ISBN: 026218253X (The MIT Press, 2005).
85. James, G., Witten, D., Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R* ISBN: 1461471370, 9781461471370 (Springer Publishing Company, Incorporated, 2014).
86. Hoecker, A. *et al.* TMVA: Toolkit for Multivariate Data Analysis. *PoS* **ACAT,** 040 (2007).
87. Breiman, L. Bias, Variance , And Arcing Classifiers. *Technical Report 460, Statistics Department, University of California* (Nov. 2000).
88. Freund, Y. & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* **55,** 119 –139. ISSN: 0022-0000 (1997).
89. Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* **29,** 1189–1232. ISSN: 00905364 (2001).
90. Chen, T. & Guestrin, C. XGBoost: A Scalable Tree Boosting System. *arXiv e-prints,* arXiv:1603.02754 (2016).
91. Adam-Bourdarios, C. *et al.* The Higgs Machine Learning Challenge. *J. Phys. Conf. Ser.* **664,** 072015 (2015).
92. Nair, V. & Hinton, G. E. *Rectified Linear Units Improve Restricted Boltzmann Machines* in *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Omnipress, Haifa, Israel, 2010), 807–814. ISBN: 978-1-60558-907-7. http://dl.acm.org/citation.cfm?id=3104322.3104425.
93. Radovic, A. *et al.* Machine learning at the energy and intensity frontiers of particle physics. *Nature* **560,** 41–48 (2018).
94. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **60,** 84–90. ISSN: 0001-0782 (May 2017).
95. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2,** 359 –366. ISSN: 0893-6080 (1989).

96. Lin, H. W., Tegmark, M. & Rolnick, D. Why Does Deep and Cheap Learning Work So Well? *Journal of Statistical Physics* **168,** 1223–1247 (2017).
97. Ghahramani, Z. *Unsupervised learning* in *Summer School on Machine Learning* (2003), 72–112.
98. MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms* ISBN: 0521642981 (Cambridge University Press, New York, NY, USA, 2002).
99. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. *A density-based algorithm for discovering clusters in large spatial databases with noise* in (AAAI Press, 1996), 226–231.
100. Catani, S., Dokshitzer, Y. L., Seymour, M. H. & Webber, B. R. Longitudinally invariant K_t clustering algorithms for hadron hadron collisions. *Nucl. Phys.* **B406,** 187–224 (1993).
101. Dokshitzer, Y. L., Leder, G. D., Moretti, S. & Webber, B. R. Better jet clustering algorithms. *JHEP* **08,** 001 (1997).
102. Cacciari, M., Salam, G. P. & Soyez, G. The anti-ktjet clustering algorithm. *Journal of High Energy Physics* **2008,** 063–063 (2008).
103. Keogh, E. & Mueen, A. in *Encyclopedia of Machine Learning and Data Mining* (eds Sammut, C. & Webb, G. I.) 314–315 (Springer US, Boston, MA, 2017). ISBN: 978-1-4899-7687-1. doi:10.1007/978-1-4899-7687-1_192. https://doi.org/10.1007/978-1-4899-7687-1_192.
104. Van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).
105. Chapelle, O., Schlkopf, B. & Zien, A. *Semi-Supervised Learning* 1st. ISBN: 0262514125, 9780262514125 (The MIT Press, 2010).
106. Vatanen, T. *et al.* Semi-supervised detection of collective anomalies with an application in high energy particle physics. *The 2012 International Joint Conference on Neural Networks (IJCNN),* 1–8 (2012).
107. Muandet, K. & Schölkopf, B. One-class support measure machines for group anomaly detection. *arXiv preprint arXiv:1303.0309* (2013).
108. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60,** 84–90 (2017).
109. Collobert, R. *et al.* Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **12,** 2493–2537. ISSN: 1532-4435 (Nov. 2011).
110. Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E. & Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure and Activity Relationships. *Journal of Chemical Information and Modeling* **55.** PMID: 25635324, 263–274 (2015).
111. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* **36,** 193–202. ISSN: 1432-0770 (1980).
112. LeCun, Y. *et al.* in *Advances in Neural Information Processing Systems 2* (ed Touretzky, D. S.) 396–404 (Morgan-Kaufmann, 1990). http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

113. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks* in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Curran Associates Inc., Lake Tahoe, Nevada, 2012), 1097–1105. http://dl.acm.org/citation.cfm?id=2999134.2999257.

114. Commons, W. *typical CNN architecture* File: `Typical cnn.png`. https://en.wikipedia.org/wiki/File:Typical_cnn.png.

115. Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America* **79,** 2554–2558. ISSN: 00278424 (1982).

116. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9,** 1735–1780. ISSN: 0899-7667 (Nov. 1997).

117. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv e-prints,* arXiv:1412.3555 (2014).

118. Commons, W. *A diagram for a one-unit recurrent neural network (RNN). From bottom to top : input state, hidden state, output state. U, V, W are the weights of the network. Compressed diagram on the left and the unfold version of it on the right.* File: `Recurrent neural network unfold.svg`. https://commons.wikimedia.org/w/index.php?curid=60109157.

119. Alwall, J. *et al.* The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP* **07,** 079 (2014).

120. De Favereau, J. *et al.* DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP* **02,** 057 (2014).

121. Guest, D., Cranmer, K. & Whiteson, D. Deep Learning and its Application to LHC Physics. *Ann. Rev. Nucl. Part. Sci.* **68,** 161–181 (2018).

122. Aurisano, A. *et al.* A Convolutional Neural Network Neutrino Event Classifier. *JINST* **11,** P09001 (2016).

123. Guest, D. *et al.* Jet Flavor Classification in High-Energy Physics with Deep Neural Networks. *Phys. Rev.* **D94,** 112002 (2016).

124. Abbott, B. *et al.* Search for new physics in $e\mu X$ data at DØ using Sherlock: A quasi model independent search strategy for new physics. *Phys. Rev.* **D62,** 092004 (2000).

125. Abazov, V. M. *et al.* A Quasi model independent search for new physics at large transverse momentum. *Phys. Rev.* **D64,** 012004 (2001).

126. Abbott, B. *et al.* A quasi-model-independent search for new high p_T physics at DØ. *Phys. Rev. Lett.* **86,** 3712–3717 (2001).

127. Abazov, V. M. *et al.* Model independent search for new phenomena in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV. *Phys. Rev.* **D85,** 092015 (2012).

128. Aaltonen, T. *et al.* Model-Independent and Quasi-Model-Independent Search for New Physics at CDF. *Phys. Rev.* **D78,** 012002 (2008).

129. Aaltonen, T. *et al.* Global Search for New Physics with 2.0 fb^{-1} at CDF. *Phys. Rev.* **D79,** 011101 (2009).

130. Aktas, A. *et al.* A General search for new phenomena in ep scattering at HERA. *Phys. Lett.* **B602,** 14–30 (2004).

131. Aaron, F. D. *et al.* A General Search for New Phenomena at HERA. *Phys. Lett.* **B674,** 257–268 (2009).

132. *A general search for new phenomena with the ATLAS detector in pp collisions at sort(s)=7 TeV.* tech. rep. ATLAS-CONF-2012-107 (CERN, Geneva, 2012). https://cds.cern.ch/record/1472686.

133. *A general search for new phenomena with the ATLAS detector in pp collisions at* $\sqrt{s} = 8$ *TeV* tech. rep. ATLAS-CONF-2014-006 (CERN, Geneva, 2014). https://cds.cern.ch/record/1666536.

134. collaboration, T. A. A model independent general search for new phenomena with the ATLAS detector at $\sqrt{s} = 13$ TeV (2017).

135. *Model Unspecific Search for New Physics in pp Collisions at sqrt(s) = 7 TeV* tech. rep. CMS-PAS-EXO-10-021 (CERN, Geneva, 2011). http://cds.cern.ch/record/1360173.

136. Collaboration, C. MUSiC, a Model Unspecific Search for New Physics, in pp Collisions at $\sqrt{s} = 8$ TeV (2017).

137. Hebbeker, T. *A Global Comparison between L3 Data and Standard Model Monte Carlo* Note (1998), available at https://web.physik.rwth-aachen.de/~hebbeker/l3note_2305.pdf. 1998.

138. Weisstein, E. W. *Voronoi Diagram. From MathWorld—A Wolfram Web Resource* Visited on 10/7/2019. \url{http://mathworld.wolfram.com/VoronoiDiagram.html}.

139. Commons, W. *Voronoi Diagram.* File: Euclidean Voronoi diagram.svg. https://commons.wikimedia.org/w/index.php?curid=38534275.

140. Gross, E. & Vitells, O. Trial factors for the look elsewhere effect in high energy physics. *Eur. Phys. J.* **C70,** 525–530 (2010).

141. Cerri, O., Nguyen, T. Q., Pierini, M., Spiropulu, M. & Vlimant, J.-R. Variational Autoencoders for New Physics Mining at the Large Hadron Collider. arXiv: 1811.10276 [hep-ex] (2018).

142. De Simone, A. & Jacques, T. Guiding New Physics Searches with Unsupervised Learning. arXiv: 1807.06038 [hep-ph] (2018).

143. D'Agnolo, R. T. & Wulzer, A. Learning New Physics from a Machine. arXiv: 1806.02350 [hep-ph] (2018).

144. Metodiev, E. M., Nachman, B. & Thaler, J. Classification without labels: Learning from mixed samples in high energy physics. *JHEP* **10,** 174 (2017).

145. Choudalakis, G. *On hypothesis testing, trials factor, hypertests and the BumpHunter* in *Proceedings, PHYSTAT 2011 Workshop on Statistical Issues Related to Discovery Claims in Search Experiments and Unfolding, CERN,Geneva, Switzerland 17-20 January 2011* (2011). arXiv: 1101.0390 [physics.data-an].

146. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B,* 1–38 (1977).

147. McLachlan, G. J. & Peel, D. *Finite mixture models* (Wiley Series in Probability and Statistics, New York, 2000).

148. Pan, W. & Shen, X. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research* **8,** 1145–1164 (2007).

149. Xie, B., Pan, W. & Shen, X. Variable Selection in Penalized Model-Based Clustering Via Regularization on Grouped Parameters. *Biometrics* **64,** 921–930 (2008).
150. Xie, B. *Variable selection in penalized model-based clustering* (University of Minnesota Press, 2008).
151. AMVA4NewPhysics Authors. *Report on a Statistical Learning Method for Model-Independent Searches for New Physics* tech. rep. Work Package 4 - Deliverable 4.2 (2017). https://docs.infn.it/share/s/fvqlIQUATw61lSRm0-trrQ.
152. Schwarz, G. Estimating the Dimension of a Model. *Ann. Statist.* **6,** 461–464 (Mar. 1978).
153. Fuks, B. Beyond the Minimal Supersymmetric Standard Model: from theory to phenomenology. *International Journal of Modern Physics* **A27,** 1230007 (2012).
154. The Minimal Supersymmetric Standard Model with R-parity violation (2012).
155. Diglio, S., Feligioni, L. & Moultaka, G. Stashing the stops in multijet events at the LHC. *Phys. Rev.* **D96,** 055032 (2017).
156. Ball, R. D. *et al.* Parton distributions with LHC data. *Nuclear Physics* **B867,** 244–289 (2013).
157. Bjorken, J. D. & Brodsky, S. J. Statistical Model for Electron-Positron Annihilation into Hadrons. *Phys. Rev. D* **1,** 1416–1420 (5 1970).
158. Tukey, J. W. Exploratory data analysis. *Reading: Addison-Wesley* (1977).
159. Mangiafico, S. S. An R companion for the handbook of biological statistics. *Available: rcompanion. org/documents/RCompanionBioStatistics. pdf.(January 2016)* (2015).
160. SHAPIRO, S. S. & WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika* **52,** 591–611. ISSN: 0006-3444 (Dec. 1965).
161. Egan, J. P. *Signal detection theory and ROC analysis* (Academic Press, 1975).
162. Frate, M., Cranmer, K., Kalia, S., Vandenberg-Rodes, A. & Whiteson, D. Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes. arXiv: 1709.05681 [physics.data-an] (2017).
163. Duvenaud, D. *Automatic model construction with Gaussian processes* PhD thesis (University of Cambridge, 2014). https://www.repository.cam.ac.uk/handle/1810/247281.
164. Aad, G. *et al.* Search for new phenomena in dijet mass and angular distributions from pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. *Phys. Lett.* **B754,** 302–322 (2016).
165. Agostinelli, S. *et al.* GEANT4: A Simulation toolkit. *Nucl. Instrum. Meth.* **A506,** 250–303 (2003).
166. Aaboud, M. *et al.* Search for new phenomena in dijet events using 37 fb^{-1} of pp collision data collected at $\sqrt{s}$ =13 TeV with the ATLAS detector. *Phys. Rev.* **D96,** 052004 (2017).
167. Aaboud, M. *et al.* Search for heavy particles decaying into top-quark pairs using lepton-plus-jets events in protonproton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. *Eur. Phys. J.* **C78,** 565 (2018).
168. Hill, C. T. Topcolor assisted technicolor. *Phys. Lett.* **B345,** 483–489 (1995).
169. Randall, L. & Sundrum, R. A Large mass hierarchy from a small extra dimension. *Phys. Rev. Lett.* **83,** 3370–3373 (1999).

170. Agashe, K., Davoudiasl, H., Perez, G. & Soni, A. Warped Gravitons at the LHC and Beyond. *Phys. Rev.* **D76,** 036006 (2007).

171. Fitzpatrick, A. L., Kaplan, J., Randall, L. & Wang, L.-T. Searching for the Kaluza-Klein Graviton in Bulk RS Models. *JHEP* **09,** 013 (2007).

172. Choudalakis, G. & Casadei, D. Plotting the differences between data and expectation. *European Physical Journal Plus* **127,** 25 (Feb. 2012).

173. Wasserman, L. *All of Statistics: A Concise Course in Statistical Inference* ISBN: 1441923225, 9781441923226 (Springer Publishing Company, Incorporated, 2010).

174. James, F. & Roos, M. Minuit – a system for function minimization and analysis of the parameter errors and correlations. *Computer Physics Communications* **10,** 343–367 (Dec. 1975).

175. Iminuit team. *iminuit – A Python interface to Minuit* https://github.com/iminuit/iminuit. Accessed: 2018-03-05.

176. Hastie, T. & Tibshirani, R. Generalized Additive Models. *Statist. Sci.* **1,** 297–310 (Aug. 1986).

177. Hastie, T. & Tibshirani, R. *Generalized Additive Models* ISBN: 9780412343902. https://books.google.com/books?id=qa29r1Ze1coC (Taylor & Francis, 1990).

178. Hastie, T. *gam: Generalized Additive Models* R package version 1.16 (2018). https://CRAN.R-project.org/package=gam.

179. Wood, S. *Generalized Additive Models: An Introduction with R* 2nd ed. (Chapman and Hall/CRC, 2017).

180. Wood, S. N. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* **73,** 3–36 (2011).

181. Servén, D. & Brummitt, C. *pyGAM: Generalized Additive Models in Python* Mar. 2018. doi:10.5281/zenodo.1208723. https://doi.org/10.5281/zenodo.1208723.

182. Buja, A., Hastie, T. & Tibshirani, R. Linear Smoothers and Additive Models. *Ann. Statist.* **17,** 453–510 (June 1989).

183. Lou, Y., Caruana, R. & Gehrke, J. *Intelligible Models for Classification and Regression* in (ACM, 2012). ISBN: 978-1-4503-1462-6. https://www.microsoft.com/en-us/research/publication/intelligible-models-classi\%EF\%AC\%81cation-regression/.