Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

A Model-Agnostic Search for Beyond the Standard Model Physics in the Dijet Topology using Machine Learning

Dissertation
zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Physik
der Universität Hamburg

vorgelegt von
Tobias Quadfasel

Hamburg
2024

Gutachter/innen der Dissertation:	Prof. Dr. Gregor Kasieczka Prof. Dr. Peter Schleper
Zusammensetzung der Prüfungskommission:	Dr. Sarah Heim Prof. Dr. Gregor Kasieczka Prof. Dr. Konstantinos Nikolopoulos Prof. Dr. Arwen Pearson Prof. Dr. Géraldine Servant
Vorsitzende/r der Prüfungskommission:	Prof. Dr. Arwen Pearson
Datum der Disputation:	24.05.2024
Vorsitzender des Fach-Promotionsausschusses PHYSIK:	Prof. Dr. Markus Drescher
Leiter des Fachbereichs PHYSIK:	Prof. Dr. Wolfgang J. Parak
Dekan der Fakultät MIN:	Prof. Dr.-Ing. Norbert Ritter

Eidesstattliche Versicherung / Declaration on oath

Hiermit versichere ich an Eides statt, die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt zu haben.

Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Hamburg, den 27.05.2024

Unterschrift der Doktorandin/des Doktoranden

Zusammenfassung

Trotz signifikanter Anstrengungen der Kollaborationen am LHC und darüber hinaus nach Physik jenseits des Standardmodells zu suchen, konnten bisher keine Beweise für deren Existenz gefunden werden. Modellunabhängige Suchen ergänzen aktuelle Anstrengungen, da sie den Nachweis potenzieller neuer Physik-Anomalien erlauben, ohne auf ein spezifisches Signalmodell abzuzielen.

Diese Arbeit diskutiert die Entwicklung und Anwendung neuartiger, datengetriebener Methoden für modellunabhängige Erkennung von Anomalien. Insbesondere wird eine neue Methode – genannt CATHODE – auf Basis von neuronaler Dichteabschätzung und schwacher Klassifikation entwickelt, welche die bisher beste Leistung auf einem häufig genutzten Benchmark-Datensatz erreicht. Die erste Anwendung von CATHODE sowie anderen auf maschinellem Lernen basierenden Methoden auf Proton-Proton Kollisionsdaten vom CMS experiment, welche von 2016 bis 2018 bei einer Schwerpunktsenergie von $\sqrt{s} = 13\,\text{TeV}$ gemessen wurden, wird diskutiert. Insbesondere wird nach hardonischen Resonanzen im Endzustand mit zwei Jets gesucht. Eine generische Suche für neue Physik im invarianten Massenspektrum des Zwei-Jet-Systems zeigte keinen signifikanten Überschuss an Ereignissen. In Bezug auf den Grenzwert des Wirkungsquerschnitts erreicht CATHODE die sensitivsten Ergebnisse für mehrere der gesteten Massehypothesen eines $X \to YY' \to 4q$ Signalmodells, wobei die optimale Verbesserung gegenüber einer inklusiven Suche den Faktor 1.9 betrug.

Mehrere Verbesserungen des ursprünglichen CATHODE-Algorithmus werden vorgeschlagen. Latent CATHODE ermöglicht die Extraktion des Signals auch im Falle von starken Korrelationen zwischen den Eingabemerkmalen und der Masse, indem die schwache Klassifikation im latenten Raum durchgeführt wird. Es wird gezeigt, dass die Nutzung von Gradient-Boosting-Klassifikatoren anstelle von auf Deep Learning basierenden Methoden in der schwachen Klassifikation deutlich robuster gegenüber nicht informativen Merkmalen ist, was modellunabhängige Suchen ohne deren vorherige Auswahl ermöglicht.

Abstract

Despite significant efforts to search for new beyond the Standard Model phenomena at the LHC physics program and beyond, no evidence has been found so far. Model-agnostic searches complement current search efforts, allowing for the detection of potential new physics anomalies without targeting a particular signal model.

This thesis discusses the development and application of novel data-driven methods for model-agnostic anomaly detection. In particular, a new method based on neural density estimation and weak classification – named CATHODE – is developed, achieving state-of-the-art performance on a commonly used benchmark data set. The first application of CATHODE and other Machine Learning-based methods on proton-proton collision data taken at the CMS experiment from 2016 to 2018 at a centre-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$ is discussed. Specifically, hadronic resonances in the two-jet final state are targeted. A generic search for new physics in the invariant mass spectrum of the two jets revealed no significant excess. In terms of cross-section limits, CATHODE achieves the most sensitive results for several of the tested mass hypotheses of an $X \to YY' \to 4q$ signal model, with an optimal improvement over an inclusive search of factor 1.9.

Several improvements to the original CATHODE algorithm are proposed. Latent CATHODE allows for signal extraction in the case of strong correlations between input features and mass by conducting the weak classification in latent space. Using Gradient Boosting classifiers instead of Deep Learning-based methods in weak classification is shown to be considerably more robust against uninformative features, allowing for model-agnostic searches without prior feature selection.

Acknowledgements

At first, I would like to thank Prof. Dr. Gregor Kasieczka and Prof. Dr. Peter Schleper for the excellent supervision. I would like to thank Prof. Dr. Kasieczka specifically for giving me the chance to work in his group on such an interesting project and providing many opportunities to present my research at both national and international scientific conferences.

My sincere gratitude also goes to Dr. Sarah Heim, Prof. Dr. Arwen Pearson and Prof. Dr. Géraldine Servant for being part of my examination committee.

Furthermore, I would like to thank Dr. Louis Moureaux for his extensive support during the final weeks for commenting and proofreading this work. In addition, I would like to thank again Dr. Louis Moureaux and my colleague Manuel Sommerhalder for the many years of successful collaboration.

In a similar vein, special thanks to the entire working group of Prof. Dr. Kasieczka who provided a great working environment even during the troublesome times of a global pandemic.

I would also like to thank the Institute for Experimental Physics at the University of Hamburg, which has shaped me both as a researcher and as a person in the seven years I had the honour to be a student there. Starting from detector-related studies in the group of Prof. Dr. Garutti as a bachelor student and continuing with the application of Machine Learning-based techniques in analysis for both my masters and doctoral studies, I was given the incredible opportunity to experience a large variety of the many facets of experimental Particle Physics in the CMS collaboration.

Finally, I would like to thank my friends and family for their incredible support and patience throughout the entire time of my doctorate.

Contents

Chapter 1

Introduction

The Standard Model (SM) of Particle Physics is the most successful model describing all the fundamental particles known to date as well as their interactions through the electromagnetic, weak and strong nuclear forces. Furthermore, it describes how fermions acquire mass through the Higgs mechanism and the experimental discovery of the Higgs boson by the CMS and ATLAS collaborations in 2012 [1, 2] was a major breakthrough in high energy physics research. Since then, the SM has been verified time and again with unprecedented precision.

Despite its success the SM is not complete, since it does not describe gravity, another fundamental interaction of matter. Additionally, a variety of physics phenomena exist that cannot be explained by the SM in its current form. Among other things, these include the baryon asymmetry, the existence of Dark Matter and Dark Energy as well as neutrino masses. Due to these shortcomings, significant effort by the high energy physics community was put into extending the SM or devise entirely new theories that describe said phenomena and observe the novel particles emerging from them. To this date, however, no evidence for physics beyond the standard model (BSM) has been found.

In principle, there are three possible reasons for this result: First, it could be that there simply is no new physics at the scale which is achieved by current collider technology. Second, the rarity of the new physics processes does not allow for a discovery given the currently available amount of collected data. Third, the BSM physics model that is realized in nature has not yet been found and/or searched for in the correct region of phase space. The first two reasons can only be mitigated through novel technologies or patience for more data being collected. The third point, however, can be solved through different means. One method is to probe the existence of signal models which emerge from contemporary BSM theories and which were not previously searched for in yet unexplored regions of phase space. This is the de facto default search method employed at most experiments at the Large Hadron Collider (LHC) and beyond. These "model-specific" searches should certainly be continued given their past success in discovering all the known fundamental particles of the SM.

However, given the long period since the discovery of the Higgs boson without any evidence for new physics despite significant research efforts, new methods should also be considered. The key problem with traditional search methods is two-fold. First, they require the realization of the BSM theory underlying the targeted signal model in nature and it is unclear whether this is the case for any of the contemporary BSM theories. Second, the regions of yet unexplored phase space at the LHC are vast and the number of possible signal model hypotheses is large. Therefore, conducting a dedicated search for all of these models in the entirety of the phase space is not feasible due to limited

computational and human resources.

This work discusses the application of model-agnostic methods, which are able to search for anomalous signals in a large region of phase space without targeting specific signal models. While in this approach the sensitivity to a particular signal model is significantly reduced, it enables the search for multiple signal model hypotheses and a model-independent scan for anomalous events in a single analysis. Therefore, it constitutes a trade-off that complements current search efforts effectively.

Model-independent searches have been conducted in collider experiments for a long time already. However, many of them are based on Monte Carlo (MC) simulations of the known SM background model and look for deviations between the expected background and the data in bins of different final states [3–8]. Naturally, this results in these methods to be entirely reliant on the accuracy of the simulations. Therefore, it is difficult to assess whether a significant excess seen by such a method originates from an actual new physics signal or a mismodeling of the SM background. The emergence of novel Machine Learning (ML) algorithms, in particular in the realm of probabilistic generative models, enables an entirely new approach to model-agnostic searches that removes the need of MC simulations. These models can learn the SM background directly from data and use this information to scan for anomalies that could hint at the existence of new physics.

While entirely model-independent methods that can be applied in any scenario and topology would be desired, no such methods exist yet. Therefore, minor assumptions regarding the kind of anomaly to target have to be made. This work in particular focuses on the search for hadronic resonances with two large-radius jets in the final state that contain anomalous jet substructure. While this certainly adds some model-dependence, there are many theories predicting particles that would result in the emergence of hadronic resonances [9]. Therefore, this analysis is still model-agnostic within a large family of theory models not yet covered by a dedicated search and the model-dependence introduced is deemed acceptable.

This work discusses the development and application of state-of-the-art, data-driven ML algorithms in model-agnostic searches for new physics in the two-jet final state. In particular, their foundations as well as their validation and estimated sensitivity are considered based on synthetic data sets, followed by their application in a first-of-its-kind analysis on proton-proton collision data taken at the CMS experiment during Run 2 (years 2016 to 2018) at a centre-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$.

Chapter 2

The Standard Model of Particle Physics

The Standard Model (SM) of particle physics is among the most successful physics models to date, describing all fundamental particles known to date as well as their interactions with high precision. While a full description of the SM and all its aspects is beyond the scope of this work, a general outline is provided in this chapter, in particular regarding the Lagrangian formalism. This outline is based mainly on the content and discussions in Refs. [10–12].

The SM incorporates three of the four fundamental forces, namely the strong interaction, the weak interaction and the electromagnetic interaction. Particles represent the fundamental constituents of matter and are distinguished based on several physics properties. Bosons are particles with integer spin and the gauge bosons in particular mediate the interactions between particles, being carriers of the forces. The Higgs boson is the only spin 0 particle and through its interaction, the particles acquire their mass. Particles with non-integer spin are fermions, which are divided into leptons that interact only through the electromagnetic and weak interactions and quarks, that additionally interact through the strong force due to their colour charge. An overview of the known fundamental particles and their key physics properties is shown in Table 2.1.

In this work, a notation based on four-vectors in Minkowski space (space-time coordinates) is used for the mathematical formulation of the SM:

$$x^{\mu} = (t, x, y, z), \tag{2.1}$$

where t denotes the time and (x, y, z) are the coordinates in three dimensional space. Note that the time component is actually ct such that all elements in x^{μ} corresponds to a length unit, but natural units are assumed, where $c = 1$. The corresponding covariant vector is given by:

$$x_{\mu} = \begin{pmatrix} t \\ -x \\ -y \\ -z \end{pmatrix}. \tag{2.2}$$

The translation from a contravariant to a covariant four-vector (and vice-versa) can then be performed

Fermions	**Quarks**						
	Name	up	down	charm	strange	top	bottom
	Mass [MeV]	2.2	4.7	$1.27 \cdot 10^3$	93	$173 \cdot 10^3$	$4.18 \cdot 10^3$
	Electric charge [e]	$+\frac{2}{3}$	$-\frac{1}{3}$	$+\frac{2}{3}$	$-\frac{1}{3}$	$+\frac{2}{3}$	$-\frac{1}{3}$
	Spin	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	Weak isospin	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$
	Coloured	yes	yes	yes	yes	yes	yes
	Leptons						
	Name	electron	ν_e	muon	ν_μ	tau	ν_τ
	Mass [MeV]	0.511	$< 10^{-6}$	105.7	$< 10^{-6}$	$1.777 \cdot 10^3$	$< 10^{-6}$
	Electric charge [e]	-1	0	-1	0	-1	0
	Spin	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	Weak isospin	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$
	Coloured	no	no	no	no	no	no

Bosons	**Gauge Bosons**					**Higgs**
	Name	photon	W	Z	gluon	H_0
	Mass [GeV]	0	80.4	91.2	0	125.25
	Electric charge [e]	0	± 1	0	0	0
	Spin	1	1	1	1	0
	Interaction	electric charge	weak isospin	weak isospin	colour charge	mass

Table 2.1: Overview of the currently known elementary particles of the Standard Model and their physical properties [13].

as:

$$x_\mu = g_{\mu\nu} x^\nu \tag{2.3}$$
$$x^\mu = g^{\mu\nu} x_\nu, \tag{2.4}$$

with

$$g_{\mu\nu} = g^{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{2.5}$$

The four-momentum vectors are given in this notation by:

$$p^\mu = (E, p_x, p_y, p_z) \tag{2.6}$$

and the four-vector derivatives by:

$$\partial_\mu = \frac{\partial}{\partial x^\mu} = \begin{pmatrix} \frac{\partial}{\partial t} \\ \vec{\nabla} \end{pmatrix} = \begin{pmatrix} \partial_t \\ \partial_x \\ \partial_y \\ \partial_z \end{pmatrix}. \tag{2.7}$$

The overarching theoretical framework for the SM is provided by quantum field theory, using a Lagrangian approach to describe the kinematics and dynamics. Within this framework, each particle is described as a field in space-time. The Standard Model is built by initially proposing a set of symmetries for the system and subsequently constructing the most general renormalizable Lagrangian based on its particle (field) content while at the same time adhering to these symmetries. The SM Lagrangian consists of several parts describing the different force interactions between particles. The different parts are discussed in the following sections.

2.1 Quantum Electrodynamics

Quantum electrodynamics (QED) is the theory of electromagnetic interactions between elementary particles that carry electric charge. The theoretical motivation behind QED is the generalization of a global to a local gauge symmetry. Local gauge invariance requires the existence of vector fields, namely the gauge bosons, as well as interactions between these fields and the charged particles. In particular, the local gauge invariance for a $U(1)$ phase transformation is discussed. $U(1)$ is the unitary group of degree 1, i.e. the group of 1×1 matrices that are unitary ($U^\dagger U = 1$). For a spin-1/2 particle, the Lagrange density for the Dirac equation is given by:

$$\mathscr{L} = \bar{\psi} i \gamma^\mu \partial_\mu \psi - m \bar{\psi} \psi, \tag{2.8}$$

where the wave function ψ is a four-spinor and the γ^μ are the γ-matrices:

$$\gamma^0 = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix},\ \gamma^i = \begin{pmatrix} 0 & \sigma_i \\ -\sigma_i & 0 \end{pmatrix},\ \gamma^5 = \begin{pmatrix} -I & 0 \\ 0 & I \end{pmatrix}. \tag{2.9}$$

Here, I are the 2×2 identity matrices and the σ_i are the Pauli matrices.

Laws of physics should generally not be associated with global transformations, as these transformations act simultaneously everywhere, thereby contradicting the fundamental principle of causality. Instead, a local $U(1)$ gauge symmetry is postulated, such that α can be a continuous function of space-time. The local gauge transformation under which the Dirac spinors ψ and $\bar{\psi}$ should be invariant can be written as:

$$\begin{aligned} \psi(x) \rightarrow \psi'(x) &= e^{iq\alpha(x)}\psi(x) \\ \bar{\psi}(x) \rightarrow \bar{\psi}'(x) &= e^{iq\alpha(x)}\bar{\psi}(x). \end{aligned} \tag{2.10}$$

In these transformations, space-time coordinates were used ($x = (t, x, y, z)$), which will be the case for the remainder of this chapter. q is a real-valued constant and α a real phase. For gauge invariance, it has to be shown that:

$$\mathscr{L}' = \bar{\psi}' i\gamma^\mu \partial_\mu \psi' - m\bar{\psi}'\psi' = \mathscr{L}. \tag{2.11}$$

Applying the transformation (Equation 2.10) yields:

$$\begin{aligned} \mathscr{L}' &= \bar{\psi}' i\gamma^\mu \partial_\mu \psi' - m\bar{\psi}'\psi' \\ &= e^{-iq\alpha(x)}\bar{\psi} i\gamma^\mu e^{iq\alpha(x)}(\partial_\mu \psi + iq\psi\partial_\mu \alpha(x)) - m\bar{\psi}\psi \\ &= \bar{\psi} i\gamma^\mu \partial_\mu \psi - m\bar{\psi}\psi - q\bar{\psi}\gamma^\mu \psi \partial_\mu \alpha(x) \\ &= \mathscr{L} - q\bar{\psi}\gamma^\mu \psi \partial_\mu \alpha(x). \end{aligned} \tag{2.12}$$

Due to the non-invariance of the $\partial_\mu \psi$ derivative term, the Lagrange density is not invariant under a local $U(1)$ symmetry. However, the gauge invariance can be recovered by replacing the derivative with a covariant derivative:

$$D_\mu \equiv \partial_\mu + iqA_\mu(x), \tag{2.13}$$

with a new vector field $A_\mu(x)$, which transforms as:

$$A_\mu \rightarrow A'_\mu(x) = A_\mu(x) - \partial_\mu \alpha(x). \tag{2.14}$$

Using this covariant derivative, one can show that the first term of Equation 2.8, $\bar{\psi} i\gamma^\mu D_\mu \psi$, is now gauge invariant:

$$\begin{aligned} D'_\mu \psi' &= (\partial_\mu + iqA'_\mu)\psi' \\ &= \partial_\mu \left(e^{iq\alpha}\psi\right) + iq(A_\mu - \partial_\mu \alpha)e^{iq\alpha}\psi \\ &= e^{iq\alpha}(\partial_\mu \psi + iq\psi\partial_\mu \alpha + iqA_\mu \psi - iq\psi\partial_\mu \alpha) \\ &= e^{iq\alpha} D_\mu \psi \end{aligned} \tag{2.15}$$

and therefore:

$$\bar{\psi}' i\gamma^\mu D'_\mu \psi' = e^{-iq\alpha}\bar{\psi} i\gamma^\mu e^{iq\alpha} D_\mu \psi = \bar{\psi} i\gamma^\mu D_\mu \psi. \tag{2.16}$$

Thus, using the covariant derivative, $\mathscr{L}' = \mathscr{L}$. Writing out the Lagrangian yields:

$$\begin{aligned}\mathscr{L} &= \bar{\psi} i\gamma^\mu D_\mu \psi - m\bar{\psi}\psi \\ &= \bar{\psi} i\gamma^\mu \partial_\mu \psi - m\bar{\psi}\psi - q\bar{\psi}\gamma^\mu \psi A_\mu.\end{aligned} \tag{2.17}$$

Therefore, the requirement of a local $U(1)$ gauge invariance postulates a new vector field A_μ as well as its interaction with the fermions, which is described by the last term in Equation 2.17, whereas the first term is the kinetic term and the second one is the mass term. The vector field can be identified as the photon, the ψ spinors are the fermions and the constant q describes the coupling strength, given by the elementary charge ($q = -e$), which is related to the fine structure constant:

$$\alpha_{\text{em}} = \frac{e^2}{4\pi} \approx \frac{1}{137}. \tag{2.18}$$

Since the new vector field $A_\mu(x)$ constitutes a new degree of freedom it could have a mass and kinetic term. However, the mass term would break the gauge symmetry for non-zero mass:

$$m_A^2 A'^\mu A'_\mu = m_A^2 \left(A^\mu A_\mu - 2A^\mu \partial_\mu \alpha + \frac{1}{q^2} \partial^\mu \alpha \partial_\mu \alpha \right) \tag{2.19}$$

and therefore, $m_A = 0$. Also the kinetic term for A_μ, containing derivatives $\partial^\mu A^\nu$ would not be gauge invariant, which is why the field strength tensor is used:

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu. \tag{2.20}$$

One can show that this is gauge invariant:

$$\begin{aligned}F'_\mu\nu &= \partial_\mu A'_\nu - \partial_\nu A'_\nu \\ &= \partial_\mu (A_\nu - \partial_\nu \alpha) - \partial_\nu (A_\mu - \partial_\mu \alpha) \\ &= F_\mu\nu\end{aligned} \tag{2.21}$$

The gauge invariant Lagrange density for the kinetic term of the vector field is then given by:

$$\mathscr{L}_A = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} \tag{2.22}$$

In total, the QED Lagrangian is:

$$\mathscr{L}_{\text{QED}} = \bar{\psi} i\gamma^\mu \partial_\mu \psi - m\bar{\psi}\psi - q\bar{\psi}\gamma^\mu \psi A_\mu - \frac{1}{4} F^{\mu\nu} F_{\mu\nu}. \tag{2.23}$$

2.2 Quantum Chromodynamics

Quantum Chromodynamics (QCD) describes the interactions between particles that carry colour charge, which are quarks and gluons, the gauge bosons of QCD. Similar to QED, where the pho-

ton emerged from postulating a local $U(1)$ gauge invariance, quantum chromodynamics (QCD) can be derived assuming an invariance under $SU(3)$ transformations. $SU(3)$ describes the group of 3×3 unitary matrices that have a determinant of 1. Consider the Lagrange density of three spinors with identical mass m, one of each colour (r =red, g =green, b =blue):

$$\mathscr{L} = \bar{\psi}_r(i\gamma^\mu \partial_\mu - m)\psi_r + \bar{\psi}_g(i\gamma^\mu \partial_\mu - m)\psi_g + \bar{\psi}_b(i\gamma^\mu \partial_\mu - m)\psi_b. \tag{2.24}$$

This can be written in a simpler form by introducing colour triplets as a generalization of the Dirac field:

$$\psi = \begin{pmatrix} \psi_r \\ \psi_b \\ \psi_g \end{pmatrix}, \ \psi_r = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \psi_b = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \psi_g = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{2.25}$$

Including the summation over the colour indices implicitly, the Lagrangian is then given by:

$$\mathscr{L} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi. \tag{2.26}$$

Similar to QED, a local gauge invariance is postulated, this time to $SU(3)$ transformations. Such transformations can be written using the generators of $SU(3)$ as:

$$\psi \to \psi' = e^{ig_s\alpha^a(x)T^a}\psi, \tag{2.27}$$

where the T_a are the 8 linear independent hermitian 3×3 matrices in colour space, the Gell-Mann matrices, $\alpha^a(x)$ are real functions of space-time and g_s is a constant. Similar to the derivation for QED, the Lagrangian is not invariant under $SU(3)$ transformations due to the derivative term:

$$\partial_\mu \alpha^a(x)\psi \neq \alpha^a(x)\partial_\mu \psi. \tag{2.28}$$

Therefore, again the derivative ∂_μ is replaced by the covariant derivative:

$$\partial_\mu \to D_\mu \equiv \partial_\mu + ig_s T^a G^a_\mu. \tag{2.29}$$

The new vector fields G^a_μ transform as:

$$G^a_\mu \to G'^a_\mu - \partial_\mu \alpha^a(x) - g_s f^{abc}\alpha^b(x)G^c_\mu. \tag{2.30}$$

This is very similar to the transformation of the photon field (compare to Equation 2.14), except for the final term. This is necessary to include due to the non-commutative generators T_a. The f^{abc} are the $SU(3)$ structure constants. Using the covariant derivative yields the gauge invariant Lagrange density:

$$\begin{aligned} \mathscr{L} &= \bar{\psi}(i\gamma^\mu D_\mu - m)\psi \\ &= \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi - g_s\bar{\psi}(\gamma^\mu T^a G^a_\mu)\psi. \end{aligned} \tag{2.31}$$

Similar to QED where the photon field emerged as a new vector field, for QCD eight new vector fields (G^a_μ) are introduced, which describe the interaction of coloured particles and correspond to the

gluons. One can observe from the last term in Equation 2.31 that the generators are defining which gluon can interact with a given quark colour state. Similar to the photon field, the gluons require a kinetic term, which, similarly to Equation 2.22, is given by:

$$\mathscr{L}_G = -\frac{1}{4} G^a_{\mu\nu} G^{a\mu\nu}, \tag{2.32}$$

where

$$G^a_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu - g_s f^{abc} G^b_\mu G^c_\nu. \tag{2.33}$$

It can be seen that one tensor exists for each gluon and the summation in the kinetic term runs over all tensors. Just as in QED, $\mathscr{L}_G$ contains quadratic derivatives that describe the kinetic energy of the field. However, it also contains higher order terms that are proportional to $g_s(G^a_\mu)^3$ and $g_s^2(G^a_\mu)^4$, which account for the gluon self-interaction. This is a key difference between QED and QCD: since photons do not carry electromagnetic charge, self-interaction is not possible. Gluons on the other hand also carry colour charge, which enables self-interaction. The mathematical reason for this phenomenon lies in the non-commutative tensors ($f^{abc} \neq 0$) and is responsible for the increase of couplings at large distances, also known as the confinement of the quarks and gluons. This confinement is the reason why they cannot exist as free particles and only are found in bound states. Finally, the QCD Lagrangian can be written as:

$$\mathscr{L}_{\text{QCD}} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi - g_s \bar{\psi}(\gamma^\mu T^a G^a_\mu)\psi - \frac{1}{4} G^a_{\mu\nu} G^{a\mu\nu}, \tag{2.34}$$

where the first term describes the kinematics and masses of the quarks, the second term the quark-gluon coupling and the third term the kinematics and self-interaction between gluons.

2.3 The Weak Interaction and Electroweak Unification

In the mathematical description of the weak interaction, the fermions are arranged based on their chirality. The chirality determines whether a particle is considered "left-handed" or "right-handed". For massless particles, it is equal to the helicity, i.e. it is right-handed when spin points towards the direction of motion and left-handed when spin and motion point in opposite directions. For particles with mass the mathematical construct is more complex and depends on whether they transform in a left- or right-handed representation of the Poincaré group, which is the group of isometries in Minkowski space. The charge of the weak interaction is the weak isospin and both the quarks and leptons each are organized in three left-handed isospin doublets:

$$\begin{pmatrix} \nu_e \\ e \end{pmatrix}_L, \begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}_L, \begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}_L \tag{2.35}$$

$$\begin{pmatrix} u \\ d' \end{pmatrix}_L, \begin{pmatrix} c \\ s' \end{pmatrix}_L, \begin{pmatrix} t \\ b' \end{pmatrix}_L, \tag{2.36}$$

where d', s' and b' are mass eigenstates that relate to the physical quark states by the Cabibbo-Kobayashi-Maskawa (CKM) matrix:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{2.37}$$

The transition between the isospin states is possible though W boson exchange. For example, a quark in the u state is transformed in the d' state by the emission of a W^+ boson, which corresponds to a rotation in $SU(2)$ space, the group of 2×2 unitary matrices with determinant 1. While the gauge transformation in $SU(3)$ that was discussed for QCD is described as a rotation in colour space, the gauge transformation for the weak interaction is a rotation in the isospin space for left-handed particles, denoted as $SU(2)_L$. The right-handed fermion states ψ_R are arranged in singlets for which no isospin transitions are possible. Similar as for QCD and QED, the local gauge invariance in isospin space has to be shown with respect to the following transformation:

$$\psi_L \rightarrow \psi'_L = e^{ig\alpha^a(x)T^a}\psi_L, \tag{2.38}$$

where g is a coupling constant, $\alpha^a(x)$ are functions of space-time x and the T^a are the generators of the $SU(2)_L$ group, which can be defined using the Pauli matrices:

$$T^a = \frac{\sigma^a}{2}. \tag{2.39}$$

Just as in the QCD and QED cases, the derivatives acting on the function $\alpha_a(x)$ are not gauge invariant:

$$\partial_\mu e^{ig\alpha^a(x)T^a}\psi \neq e^{ig\alpha^a(x)T^a}\partial_\mu\psi \tag{2.40}$$

and therefore the derivative is again replaced by the covariant derivative:

$$D_\mu = \partial_\mu + igT^aW^a_\mu. \tag{2.41}$$

Therefore, three new vector fields W^a_μ are introduced and the Lagrange density becomes:

$$\begin{aligned} \mathscr{L} &= \bar{\psi}(i\gamma^\mu D_\mu)\psi \\ &= \bar{\psi}(i\gamma^\mu\partial_\mu)\psi - g\bar{\psi}(\gamma^\mu T^a W^a_\mu)\psi, \end{aligned} \tag{2.42}$$

which is invariant under gauge transformations if the new vector fields transform as:

$$W^a_\mu \rightarrow W'^a_\mu = W^a_\mu - \partial_\mu\alpha^a(x) - gf^{abc}\alpha^b(x)W^c_\mu. \tag{2.43}$$

Here, f^{abc} are again the structure constants of the group, which for $SU(2)$ are given by the totally anti-symmetric Levi-Civita tensor:

$$f^{abc} = \epsilon^{abc}. \tag{2.44}$$

Finally, the kinetic term for the new vector field is added in the same way it was done for the photon and gluon fields:

$$\mathscr{L}_{W,\text{kin.}} = -\frac{1}{4} W^{a\mu\nu} W^a_{\mu\nu}, \tag{2.45}$$

where the $W^a_{\mu\nu}$ are the field strength tensors:

$$W^a_{\mu\nu} = \partial_\mu W^a_\nu - \partial_\nu W^a_\mu - g f^{abc} W^b_\mu W^c_\nu. \tag{2.46}$$

The kinetic term $\mathscr{L}_{W,\text{kin.}}$ therefore contains not only the kinetic energy of the gauge bosons (quadratic terms) but, analogous to the gluons, terms proportional to $g(W^a_\mu)^3$ and $g(W^a_\mu)^4$, which represent the self-coupling of the W boson at three and four-boson vertices. The full weak interaction Lagrangian is then given by:

$$\mathscr{L}_{\text{weak}} = \bar{\psi}(i\gamma^\mu \partial_\mu)\psi - g\bar{\psi}(\gamma^\mu T^a W^a_\mu)\psi - \frac{1}{4} W^{a\mu\nu} W^a_{\mu\nu}, \tag{2.47}$$

with the three terms describing the kinematics of the fermions, the coupling of the isospin doublet to the W and the kinematics of the W boson and its self-interaction, respectively. While this theory correctly predicts the transitions of particles within the isospin doublets, there are still two problems that have to be adressed. First, it is known that the W boson is not massless. However, adding a mass term to the Lagrangian would break the local gauge invariance. Second, there are only two W bosons and a Z boson with different couplings, while the postulation of local gauge invariance with respect to $SU(2)$ results in three W bosons that all share the same coupling g.

The latter problem can be solved by electroweak unification. In a first step, a new local gauge symmetry with respect to $U(1)$ transformations is postulated. The interactions originating from this symmetry are due to a new kind of charge, which is the weak hypercharge that is defined as:

$$Y = 2(Q - T_3), \tag{2.48}$$

where Q is the electric charge and T_3 is the third component of the weak isospin. The derivation of the gauge-invariant Lagrangian is now analogous to QED, with only minor differences. The covariant derivative is defined as:

$$D_\mu = \partial_\mu + ig' \frac{Y}{2} B_\mu(x), \tag{2.49}$$

where g' is a coupling constant and B_μ is a new vector field emerging from the symmetry. The full Lagrange density for the interactions due to the weak hypercharge, including the kinetic term, is given by:

$$\begin{aligned} \mathscr{L}_Y &= \bar{\psi}(i\gamma^\mu D_\mu)\psi - \frac{1}{4} B^{\mu\nu} B_{\mu\nu} \\ &= \bar{\psi}(i\gamma^\mu \partial_\mu)\psi - g' \frac{Y}{2} \bar{\psi}(\gamma^\mu B_\mu)\psi - \frac{1}{4} B^{\mu\nu} B_{\mu\nu}, \end{aligned} \tag{2.50}$$

with the field strength tensors:

$$B^{\mu\nu} = \partial^\mu B^\nu - \partial^\nu B^\mu. \tag{2.51}$$

Since the right-handed singlets do not carry weak isospin, Y is different for left- and right-handed

particles, such that the Lagrangian can be split up as:

$$\mathscr{L}_Y = \bar{\psi}(i\gamma^\mu D_\mu)\psi - \left(g'\frac{Y_R}{2}\bar{\psi}_R(\gamma^\mu B_\mu)\psi_R + g'\frac{Y_L}{2}\bar{\psi}_L(\gamma^\mu B_\mu)\psi_L\right) - \frac{1}{4}B^{\mu\nu}B_{\mu\nu}, \tag{2.52}$$

which shows that the interactions of left- and right-handed components are different under $U(1)_Y$.

The Lagrange density for the electroweak unification is then the sum of the discussed $SU(2)_L$ and $U(1)_Y$ interactions:

$$\mathscr{L}_{\mathrm{EW}} = -\frac{1}{4}W^{a\mu\nu}W^a_{\mu\nu} - \frac{1}{4}B^{\mu\nu}B_{\mu\nu} + \sum \bar{\psi}_L i\gamma^\mu D_\mu \psi_L + \sum \bar{\psi}_R i\gamma^\mu D_\mu \psi_R. \tag{2.53}$$

The sums in the last two terms run over all left-handed doublet states and right-handed singlet states, respectively. The covariant derivative is given by:

$$D_\mu = \partial_\mu + igT^a W^a_\mu + ig'\frac{Y}{2}B_\mu. \tag{2.54}$$

Next, a new basis for the gauge boson fields is introduced:

$$W^\pm = \frac{1}{\sqrt{2}}(W^1 \mp iW^2) \tag{2.55}$$

$$T^\pm = \frac{1}{\sqrt{2}}(T^1 \pm iT^2). \tag{2.56}$$

After some calculation, one can show that D_μ can be written in this new basis as:

$$\begin{aligned} D_\mu &= \partial_\mu + ig(T^+W^+ + T^-W^-)_\mu + \left(igT^3W^3 + ig'\frac{Y}{2}B\right)_\mu \\ &= \partial_\mu + D^W_\mu + D^{\gamma Z}_\mu. \end{aligned} \tag{2.57}$$

In this case, the components with off-diagonal elements D^W_μ, which are the flavour-changing currents caused by the W bosons, were separated from the flavour-conserving components $D^{\gamma Z}_\mu$ with only diagonal elements. The new definition of the covariant derivative results in three distinct terms in the Lagrange density $\mathscr{L} = \bar{\psi}i\gamma^\mu D_\mu\psi$. The first term, $\bar{\psi}i\gamma^\mu\partial_\mu\psi$ describes the kinetic energy of the fermions. The second term, $\bar{\psi}i\gamma^\mu D^W_\mu\psi$ is the interaction term of the $W^\pm$ with the fermions and finally, the term $\bar{\psi}i\gamma^\mu D^{\gamma Z}_\mu\psi$ describes the interaction of the photon and Z boson with the fermions.

To obtain the photon and Z boson fields, the last term is again considered:

$$\mathscr{L}_{\gamma,Z} = \sum \bar{\psi}i\gamma^\mu i\left(gT^3W^3_\mu + g'\frac{Y}{2}B_\mu\right)\psi, \tag{2.58}$$

where the sum runs over both left- and right-handed spinors. Similar as before, a new basis is chosen such that W^3_μ and B are expressed in terms of two other bosons:

$$\begin{pmatrix} W^3_\mu \\ B_\mu \end{pmatrix} = \begin{pmatrix} \cos(\theta_W) & \sin(\theta_W) \\ -\sin(\theta_W) & \cos(\theta_W) \end{pmatrix}\begin{pmatrix} Z_\mu \\ A_\mu \end{pmatrix}, \tag{2.59}$$

which describes a rotation by the Weinberg angle θ_W. Inserting the new definition of W^3_μ and B_μ into

the Lagrangian yields:

$$\begin{aligned}\mathscr{L}_{\gamma,Z} = \sum i\bar{\psi} i\gamma^{\mu} \left(g \sin(\theta_W) T^3 + g' \cos(\theta_W) \frac{Y}{2} \right) A_{\mu} \psi \\ + i\bar{\psi} i\gamma^{\mu} \left(g \cos(\theta_W) T^3 - g' \sin(\theta_W) \frac{Y}{2} \right) Z_{\mu} \psi .\end{aligned} \tag{2.60}$$

The first term describes the photon-fermion coupling and the second term the Z boson-fermion coupling. When requiring that the A_{μ} is the photon field discussed for QED, then the electrical charge Q of a particle in units of the elementary charge e is given by:

$$eQ = g \sin(\theta_W) T^3 + g' \cos(\theta_W) \frac{Y}{2} = eT^3 + e\frac{Y}{2}, \tag{2.61}$$

which yields:

$$Q = T^3 + \frac{Y}{2} \text{ and } e = g \sin(\theta_W) = g' \cos(\theta_W) \tag{2.62}$$

From Equation 2.60, the Z boson coupling is given by:

$$g_Z = g \cos(\theta_W) T^3 - g' \sin(\theta_W) \frac{Y}{2} \tag{2.63}$$

which can be written using Equation 2.62 as:

$$\mathscr{L} = \sum i\bar{\psi} i\gamma^{\mu} (eQA_{\mu} + g_Z Z_{\mu}) \psi . \tag{2.64}$$

While the electroweak unification explains the $W^{\pm}$ bosons, the Z boson as well as their different couplings, it is still unable to solve the problem that the gauge invariance is violated when mass terms are added to the Lagrangian. This can be achieved using the Brout-Englert-Higgs (BEH) mechanism, which is described in the following section.

2.4 The Higgs Mechanism

The Higgs mechanism, or more correctly, the Brout-Englert-Higgs (BEH) mechanism [14–18] allows the fermions and gauge bosons to acquire mass. The key problem is that the Lagrange density of fermions contains mass terms $\mathscr{L}_m = -m\bar{\psi}\psi$ that are not invariant under gauge transformations. Considering the example of the mass term in the Lagrange density of electrons and electron-neutrinos one obtains:

$$\mathscr{L}_m = -m_{\nu} \bar{\nu} \nu - m_e \bar{e} e = -m_{\nu} \bar{\nu}_R \nu_L - m_{\nu} \bar{\nu}_L \nu_R - m_e \bar{e}_R e_L - m_e \bar{e}_L e_R, \tag{2.65}$$

where e and ν are written as superpositions of left- and right-handed components ($e = e_L + e_R$) and $\bar{e}_R e_R = 0$, $\bar{e}_L e_L = 0$ was used. When writing $\mathscr{L}_m$ in matrix notation and applying an $SU(2)_L$ gauge transformation, this becomes:

$$\mathscr{L}'_m = -(\bar{\nu}_R, \bar{e}_R) \begin{pmatrix} m_{\nu} & 0 \\ 0 & m_e \end{pmatrix} U \begin{pmatrix} \nu_L \\ e_L \end{pmatrix} - (\bar{\nu}_L, \bar{e}_L) U^{\dagger} \begin{pmatrix} m_{\nu} & 0 \\ 0 & m_e \end{pmatrix} \begin{pmatrix} \nu_R \\ e_R \end{pmatrix}. \tag{2.66}$$

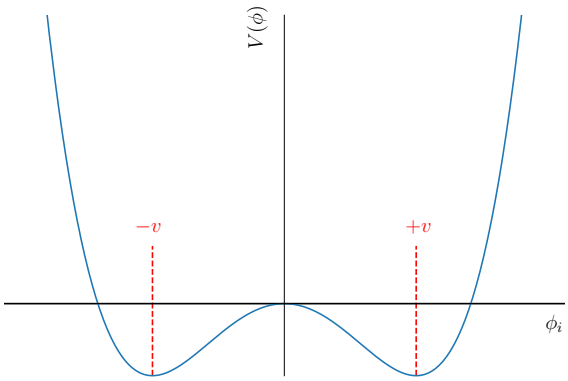


Figure 2.1: The Higgs potential $V(\phi)$ as a function of ϕ

This term is only invariant if the left and right-handed components transformed in the same way and if the electron and neutrino masses were identical, both of which is not found experimentally.

This problem is solved using the BEH mechanism. At first, a new complex scalar field ϕ is postulated, which is a $SU(2)_L$ doublet with hypercharge $Y = 1$ and spin 0:

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} = \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix}, \tag{2.67}$$

where the ϕ^0 component is chosen to not carry electromagnetic charge such that the vacuum state is neutral. Since the postulated field has spin zero, it follows the Klein-Gordon-Equation, which can be written in the Lagrangian formalism as:

$$\mathcal{L}_\phi = (D^\mu \phi)^\dagger (D_\mu \phi) - V(\phi) = |D_\mu \phi|^2 - V(\phi), \tag{2.68}$$

where the covariant derivative of $SU(2)_L \otimes U(1)_Y$ from Equation 2.57 is used. $V(\phi)$ describes a potential (also referred to as the Higgs potential) that is postulated and is designed in a way that it is invariant under gauge transformations, i.e. rotations of ϕ^+ and ϕ^0 and changes of the complex phase:

$$V(\phi) = \mu^2 |\phi|^2 + \lambda |\phi|^4 \tag{2.69}$$

where μ^2 and λ are new natural constants that emerge from the theory. If $\mu^2 > 0$, the ground state $|\phi| = 0$. However, if $\mu^2 > 0$, a spontaneously broken symmetry is obtained, since the resulting shape of V has no minimum at $|\phi| = 0$. Instead, the minimum is located at $|\phi| = v$, where v is the vacuum expectation value:

$$v = \sqrt{\frac{-\mu^2}{\lambda}}. \tag{2.70}$$

The Higgs potential has a shape similar to the characteristic form of a Mexican Sombrero hat, which is why it is also referred to as “Mexican Hat Potential”. An illustration of the potential is shown in Figure 2.1. In the four dimensions of the Higgs field, v is not a single minimum, but a continuum of minima. Therefore, a particular choice of the ground state can be made. Often a state is chosen where

only one real component is kept and the other components are zero, e.g.:

$$\phi_{\text{vac}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix}, \tag{2.71}$$

where ϕ_3 is the only non-zero component. The physical Higgs particle is then interpreted as the excited states of the Higgs field, which are obtained by developing the field around its minimum:

$$\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + H(x) \end{pmatrix}. \tag{2.72}$$

Inserting this into Equation 2.69 yields:

$$\begin{aligned} V(\phi) &= \frac{1}{2}\mu^2(v+H)^2 + \frac{1}{4}\lambda(v+H)^4 \\ &= (\mu^2 + \lambda v^2)vH + \frac{1}{2}(\mu^2 + 3\lambda v^2)H^2 + \lambda v H^3 + \frac{1}{4}\lambda H^4 \end{aligned} \tag{2.73}$$

which simplifies when inserting $v = \sqrt{-\mu^2/\lambda}$ to:

$$V(\phi) = -\mu^2 H^2 + \lambda v H^3 + \frac{1}{4}\lambda H^4. \tag{2.74}$$

The first term can be interpreted as the mass of the Higgs boson and the second and third terms describe the Higgs self-interaction, which means that three- and four-Higgs vertices should exist.

Finally, the interactions of the gauge bosons and the fermions with the Higgs have to be considered. For the former, the terms in the Lagrange density where both the Higgs and the gauge bosons occur. In particular, the following Lagrangian is considered:

$$\mathscr{L}_\phi = (D^\mu \phi)^\dagger (D_\mu \phi) - V(\phi) = |D_\mu \phi|^2 - V(\phi) = \left| \left(\partial_\mu + igT^a W_\mu^a + ig' \frac{Y}{2} B_\mu \right) \phi \right|^2 - V(\phi) \tag{2.75}$$

which after some calculations can be shown to be written as:

$$\mathscr{L}_\phi = \frac{1}{2}(\partial^\mu H)(\partial_\mu H) + \frac{1}{4}g^2(v+H)^2 W^{+\mu} W^-_\mu + \frac{1}{2}(g^2 + g'^2)(v+H)^2 Z^\mu Z_\mu - V(\phi). \tag{2.76}$$

The different terms in the Lagrangian can be interpreted as follows:

$$\frac{1}{2}(\partial^\mu H)(\partial_\mu H) \tag{2.77}$$

describes the kinetic energy of a scalar field H. Its excitation is the Higgs particle.

$$\frac{1}{4}g^2 v^2 W^{+\mu} W^-_\mu + \frac{1}{2}(g^2 + g'^2)v^2 Z^\mu Z_\mu \tag{2.78}$$

shows that the W and Z boson masses are determined by the weak coupling constants and the Higgs potential:

$$m_W = \frac{1}{2}gv, \; m_Z = \frac{1}{2}v\sqrt{g^2 + g'^2}. \tag{2.79}$$

The interactions between the Higgs and the W and Z bosons are described by the term:

$$\frac{1}{4}g^2 W^{+\mu} W^-_\mu H + \frac{1}{2}(g^2 + g'^2) Z^\mu Z_\mu H \tag{2.80}$$

It can also be seen that the coupling is proportional to the W and Z masses derived above. The four-vertices between the W and Z bosons are described by the term:

$$\frac{1}{4}g^2 W^{+\mu} W^-_\mu HH + \frac{1}{2}(g^2 + g'^2) Z^\mu Z_\mu HH \tag{2.81}$$

and finally, $V(\phi)$ describes the mass of the Higgs as well as its self-coupling, as previously discussed.

For the interactions with the fermions, consider the example of electron and electron-neutrino from Equation 2.65 and Equation 2.66, with the notation:

$$\psi_L = \begin{pmatrix} \nu_{e,L} \\ e_L \end{pmatrix}. \tag{2.82}$$

The terms containing ϕ, ψ_L and e in the Lagrange density can be written as:

$$-g_e(\bar{\psi})_L \phi e_R + \bar{e}_R \phi^\dagger \psi_L), \tag{2.83}$$

where g_e describes the coupling strength between the Higgs and fermion fields. Inserting the Higgs field from Equation 2.72, the Lagrange density becomes:

$$\begin{aligned} \mathscr{L} &= -g_e \frac{1}{\sqrt{2}}(\bar{e}_L(v + H)e_R + \bar{e}_R(v + H)e_L) \\ &= -g_e \frac{v}{\sqrt{2}}(\bar{e}_L e_R + \bar{e}_R e_L) - g_e \frac{1}{\sqrt{2}} H(\bar{e}_L e_R + \bar{e}_R e_L), \end{aligned} \tag{2.84}$$

which can be written as:

$$\mathscr{L} = -m_e \bar{e}e - \frac{m_e}{v} He\bar{e}, \tag{2.85}$$

which includes the mass term of the form $m\bar{\psi}\psi$ and the mass term of the electron can be interpreted as:

$$m_e = g_e \frac{v}{\sqrt{2}}. \tag{2.86}$$

This derivation shows that the fermions acquire their mass due to the interaction with the Higgs field and the coupling is stronger the higher the mass. Opposite to the W and Z mass terms which are computed from the coupling constants g and g', the fermion masses depend on a free parameter (in the above case, g_e), that is not predicted by the theory.

In summary, the Standard Model predicts all known fundamental particles as well as their interactions. The interactions through colour charge are modelled by QCD, whereas the weak and electromagnetic forces are described by electroweak unification of QED and the weak sectors. The bosons and fermions acquire their mass by interacting with the Higgs field.

While the development and experimental verification of the SM constitutes a significant milestone in our understanding of the known universe, there are still phenomena not described by it. These are based on physics beyond the Standard Model (BSM), which is discussed in the following chapter.

Chapter 3

Physics Beyond the Standard Model

Even though the Standard Model is the best model to date describing the known fundamental particles and interactions in our universe, it is not complete. Several theoretical and experimental aspects exist that suggest the existence of physics beyond the Standard Model (BSM). However, despite numerous efforts, no such phenomena have been observed so far. In the following, hints for BSM physics as well as the foundations of the theoretical models aiming to describe them are discussed.

3.1 Theoretical Aspects

One key problem in the theoretical description of the SM is referred to as the hierarchy problem. Very different energy scales exist in high energy physics, such as the QCD scale $\Lambda_{\mathrm{QCD}} \approx 250\,\mathrm{MeV}$, the electroweak scale $v \approx 246\,\mathrm{GeV}$, the scale at which the electromagnetic, weak and strong forces unify $\Lambda_{\mathrm{GUT}} \approx 10^{16}\,\mathrm{GeV}$ and the Planck scale $M_P \approx 10^{19}\,\mathrm{GeV}$, which occurs at energies corresponding in magnitude to the universal constants used in Planck units i.e. $E_P = \sqrt{\hbar c^5/G} = 1.9561 \cdot 10^9\,\mathrm{J} \approx 10^{19}\,\mathrm{GeV}$ and where the fundamental laws of physics break down. The large differences between the scales are also referred to as "hierarchies". It is expected that, due to these differences, large radiative corrections are needed to extrapolate between them. These quantum corrections are added to the true "bare" parameter contained in the Lagrangian to obtain the experimentally measured value. A particular problem occurs in the SM for the Higgs sector, since the Higgs mass corrections have a quadratic dependence on the scale. Assuming that the SM is valid up to the Planck scale, this leads to very large correction terms. Therefore, in order to keep the Higgs mass around the electroweak scale, the bare parameter must be fine-tuned to almost completely cancel the corrections. This level of fine-tuning, however, is considered to be not very natural or elegant.

Another problem related to the theory itself is its large number of parameters, which is 19. The parameters can be measured in experiments, but their origin is unknown. It is also often argued that a theory with such a large number of parameters is not a very elegant description of nature.

The SM also does not explain the problem referred to as "strong CP problem". The problem is rooted in the fact that the weak interaction violates CP-symmetry. A central feature of the weak interaction is that quarks are mixed states which makes flavour changes, cross-generational changes between quarks, possible (see Equation 2.37). This causes an interesting phenomenon referred to as "CP violation", which was first observed in the kaon system. C and P are operators that act on a quantum mechanical state. While C changes the charge of a particle, i.e. transforms it into its

antiparticle, the parity operator P inverts its spatial coordinates, turning it into its "mirror" particle. Originally, it was believed that physics processes are invariant under the combined application of both the C and P operator. Experimentally, however, it was observed that the weak interaction violates this symmetry in neutral kaon decays [19–21]. The mathematical formulation of the SM describes this violation for the weak interaction, but also allows it in the strong interaction. Experimentally, no CP-violating processes of the strong interaction have been observed so far. Therefore, this apparent CP symmetry in QCD processes is another unexplained phenomenon in the SM.

3.2 Beyond the Standard Model Phenomena

A key problem of the SM is that it does not explain gravitation. Gravitation is described by the theory of general relativity, which cannot be explained in terms of quantum field theory because it breaks down before reaching the Planck scale $M_P = 10^{19}$ GeV [22,23]. Postulating another symmetry in the SM that results in the emergence of a graviton, a boson describing the interaction between particles due to gravity, does not yield the results that are experimentally observed.

The SM also does not predict neutrino masses. Neutrinos do not acquire mass because the terms in Equation 2.84 contain right-handed neutrino (and left-handed antineutrino) singlets, which carry neither weak isospin nor electromagnetic charge. Therefore, they cannot interact with other SM particles and it is unknown whether they exist. However, the discovery of neutrino oscillations [24, 25] can only be explained by massive neutrinos, which are not contained in the SM.

Another phenomenon not currently explained is the asymmetry of matter and antimatter found in the universe. This is also referred to as baryon asymmetry. The SM does not provide an explanation for this phenomenon and assumes that the universe is neutrally charged with all charges being conserved. Since in theory the Big Bang should have produced matter and antimatter in the same quantities, there must be physics processes that act differently on matter compared to antimatter. However, no such processes have been observed experimentally and no consistent theory exists describing this phenomenon. While CP violation of weak processes does partly explain baryon asymmetry, the amount is by far not sufficient to describe it entirely.

The fundamental particles that the SM describes are foundational building blocks of ordinary matter. However, only about 5 % of all the matter in the universe is of this type, whereas Dark Matter (DM) accounts for 26 % and Dark Energy (DE) for 69 %, respectively. The existence of DM is motivated by experimental evidence for gravitational effects that require more mass than ordinary matter. One example of such an effect is based on the rotation curves of galaxies, which describe the rotational velocity of a star as a function of the distance from the centre of the galaxy. In a first order approximation, the expectation from Newtonian mechanics can be obtained by setting the centripetal and gravitational forces equal:

$$mr\omega^2 = G\frac{mM}{r^2} \tag{3.1}$$

$$\omega^2 = \frac{GM}{r^3}, \tag{3.2}$$

where m would be the mass of the star, M the mass at the centre of the galaxy, r the radial distance of the star, ω the angular velocity and G is the gravitational constant. Since $\omega^2 \propto 1/r^3$, the angular

velocity should decrease for increasing distance from the centre. However, the observed rotation curves are largely flat in ω for increasing r contrary to the expectation. This phenomenon can only be accounted for by an additional halo surrounding the galaxy, consisting of matter that is not visible. Therefore, this kind of matter referred to as "Dark Matter". Furthermore, also experimental evidence from galaxy clusters, gravitational lensing and other phenomena exist that require more mass than what the visible mass is accounting for [26]. In the SM, no DM particles are described, which is why it constitutes another BSM phenomenon.

Similar to DM, dark energy is a phenomenon not explained by the SM. Experimental evidence for DE comes from the redshift observations of Type Ia supernovae, which provide a measure of how fast the respective star is moving further away from the solar system. Using this method, experiments have shown that the expansion of the universe is not constant, but accelerating [27, 28]. The fact that the vacuum energy of the SM is significantly smaller than the energy needed to drive this accelerated expansion therefore constitutes another example of a BSM phenomenon.

3.3 Extensions of the Standard Model

Due to the many apparent BSM phenomena, a variety of theories have been developed to explain them and also provide a more elegant theoretical description of nature. Some of the major BSM theories are discussed in the following.

3.3.1 Grand Unified Theories

Grand Unified Theories (GUTs) suggest that the three forces described in the SM merge into a single force at high energies, which is also referred to as the GUT scale and lies at energies of $\Lambda_{\text{GUT}} = 10^{16}\,\text{GeV}$. The main theoretical motivation behind this assumption is the connection between the electric charge of leptons and quarks. Since all known particles are electromagnetically charged by multiples of one third of the elementary charge ("charge quantization"), a common representation should exist. Thus, also interactions due to the weak and strong forces could be embedded in a single symmetry group that contains the SM. For example, the simplest symmetry group that the SM could be embedded in is $SU(5)$, the group of 5×5 unitary matrices with a determinant of one. A new theory based on a single group would yield a much more elegant description of nature compared to the highly fine-tuned SM and could also solve the baryon asymmetry. Also, the couplings of the SM indeed are very close at the GUT scale, suggesting that such a unification could exist.

Since the new particles emerging in GUT models are produced at energies around Λ_{GUT}, which is far out of reach of the current collider technology, they cannot be discovered directly. However, indirect measurement methods are possible. Other than the doublets in the electroweak theory, quarks and leptons are combined in the *same* multiplets in the GUTs. The transition between the states is induced by new heavy gauge bosons X. In these processes the lepton and baryon numbers are not conserved, which is a desired property in order to describe the matter-antimatter-asymmetry observed in the universe. A consequence of these transitions is the decay of the proton that should be observable experimentally, but so far, no evidence for such a decay has been found.

3.3.2 Supersymmetry

Another theory that belongs to one of the most studied at collider experiments is Supersymmetry (SUSY). It results in an elegant description of the fundamental interactions and also predicts several of the BSM phenomena.

A key motivation of this theory is that the SM does not include a description of gravity. It can be shown that a potential graviton, the hypothetical gauge boson of the gravitational force, must have a spin of 2, since otherwise a renormalizable quantum field theory of gravity cannot be achieved. Therefore, the graviton makes it difficult to unify all forces, since the it belongs to a different representation of the Poincaré algebra. SUSY solves this by introducing a new supersymmetric algebra that describes a symmetry between fermions and bosons. It introduces generators Q that change the spin of a particle and therefore changes fermions into bosons and vice-versa. Therefore, each fermion in SUSY has an associated boson and each boson has an associated fermion, which are called the "superpartners". Since the superpartners should be equal in all quantum numbers except for the spin, their couplings and interactions should be identical to the ones in the SM. Since no such superpartners have been observed to date, this new symmetry has to be broken.

Another advantage of SUSY models is that the large radiative corrections that constitute the hierarchy problem would be significantly reduced, because corrections cancel between the partners and superpartners. Additionally, SUSY models alter the running gauge couplings of the electromagnetic, weak and strong forces such that they match almost exactly at the GUT scale, allowing for great unification [29].

3.3.3 Extra-Dimensional Theories

A key question in the hierarchy problem is why the forces described in the SM are much stronger than the gravitational force. One set of theories solves this problem by assuming that additional dimensions to the four space-time dimensions exist [30]. While the known gauge bosons and fermions exist only in the four dimensions, gravity propagates both in space-time and the other dimensions. The gravitational force is therefore spread out across multiple dimensions and thus acts much more weakly in space-time. Therefore, the Planck scale is only large in 4 dimensions, whereas the actual fundamental scale can be significantly smaller, which effectively solves the hierarchy problem. Searches in the invariant dijet mass spectrum are a key strategy for the discovery of a massive graviton interacting through quark-quark scattering [31].

3.4 Theoretical Motivation for Two-Body Resonance Searches

Resonance searches are a fundamental strategy for finding new particles at collider experiments and have been applied successfully at numerous occasions, such as the discovery of the Higgs boson [1,2], the direct observation of the Z boson [32,33], the discovery of the Υ [34], the J/ψ [35,36] and many more. These successes alone constitute a strong motivation for searches of two-body resonances in the vast phase space of the LHC, independent of a target theory.

In addition, many contemporary theory models also predict a variety of two-body resonances. Composite theories assume that the current fundamental particles are consisting of additional subcomponents, called preons [37]. One type of evidence to support these theories would be the discovery

of excited fermion states, which could de-excite back into the SM ground state by emitting other particles, such as an additional vector boson (e.g. $Q* \rightarrow qW'$) [38]. There exist also composite Higgs models that target the fine-tuning or naturalness problem discussed previously [39]. Some of these models contain new bosons that can decay into vector-like quarks, which can result in fully hadronic resonances (e.g. $W' \rightarrow tB' \rightarrow tbZ \rightarrow$hadronic) [40]. Extra-dimensional theories also predict (hadronic) two-body decays. For example, Kaluza-Klein excitations of gauge bosons can relax back into the SM state, emitting new particles such as a radion (an excitation of the gravitational field that emerges from the theory and constitutes a scalar in four-dimensional space-time), which then again can decay into SM particles (e.g. $W_{kk} \rightarrow RW \rightarrow WWW$) [41,42]. Warped extra-dimensional theories also predict new heavy Higgs-like scalars that can decay into pairs of two lighter Higgs-like scalars (H) [43].

In summary, there exist several BSM models that predict new particles which produce (hadronic) resonances. Thus, apart from the previous experimental successes of resonance searches, there is also significant motivation on theory level to conduct searches in this topology. An overview of different resonant topologies and searches targeting them is provided in refs. [9] and [44].

Chapter 4

The CMS Experiment at the LHC

4.1 The Large Hadron Collider (LHC)

The Large Hadron Collider (LHC) is, at the time of writing, the largest and most powerful particle accelerator and collider of the world, with a design centre-of-mass energy for proton-proton collisions of $\sqrt{s} = 14\,\text{TeV}$ [45, 46]. It consists of two rings with counter-rotating beams of either protons or lead ions that are intersecting at four interaction points, where different experiments have been constructed to both probe SM processes and search for evidence for new physics phenomena. The rings of the LHC are about 27 km in diameter and they contain the vacuum tubes for the beam lines, the 1232 superconducting dipole magnets for keeping the beam in circular motion as well as about 4800 orbit and multipole correction magnets. The superconducting dipoles are cooled down to a temperature below 2 K using superfluid helium and operate at magnetic fields above 8 T. To reach the high centre-of-mass energies the LHC was designed for, several pre-acceleration steps are necessary. First, protons are supplied from the linear accelerator Linac2 at an energy of 50 MeV, which are then further accelerated by the Proton Synchroton Booster (PSB), the Proton Synchrotron (PS) and finally the Super Proton Synchrotron (SPS) to per-beam energies of 1.4 GeV, 25 GeV and 450 GeV, respectively. Accelerated particles are travelling through the beam pipes in "bunches", containing several protons or lead ions. Each proton beam consists of up to 2808 such bunches, with a nominal bunch spacing of 25 ns, leading to a peak collision rate of 40 MHz. However, several bigger gaps are introduced between the bunches for example to allow the kicker magnets that dump the beam enough time to get their magnetic field up. Therefore, the crossing rate is reduced to:

$$r_{\text{cross.}} = n_b \cdot f_{\text{rev}} = 2808 \cdot 11\,245\,\text{s}^{-1} = 31.6\,\text{MHz}, \tag{4.1}$$

where n_b is the number of bunches per beam and f_{rev} is the revolution frequency.

The actual number of collision events occurring in the LHC additionally depends on several beam parameters and is given by:

$$N_{\text{event}} = L_{\text{int.}} \cdot \sigma_{\text{event}}, \tag{4.2}$$

where $L_{\text{int.}}$ is the integrated luminosity and σ_{event} is the cross section of the respective physics process. The integrated luminosity is defined as the integral of the instantaneous luminosity over time:

$$L_{\text{int.}} = \int L_{\text{instant.}} dt. \tag{4.3}$$

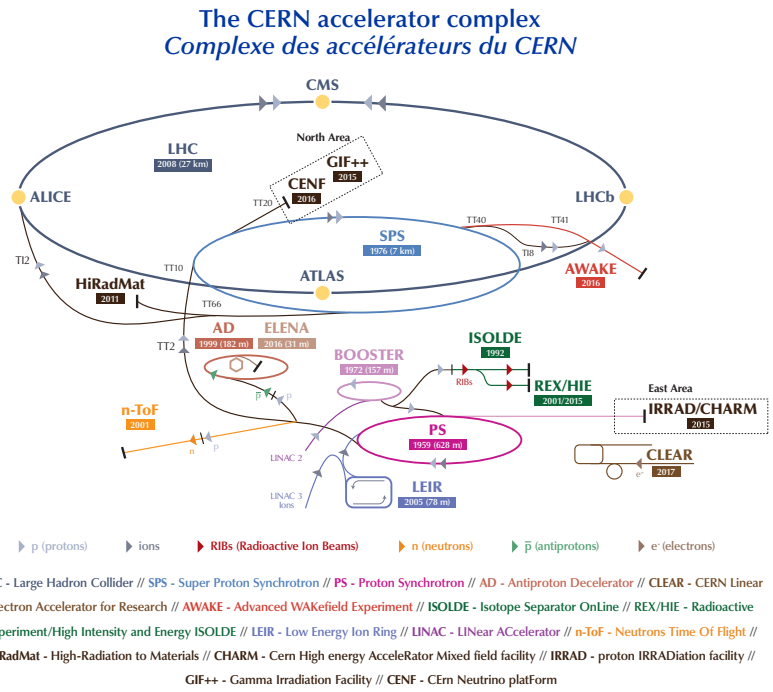


Figure 4.1: The CERN accelerator complex, including the locations of the four major experiments described in the text [48].

The instantaneous luminosity $L_{\text{instant.}}$ is a key machine parameter that quantifies the number of collision events per time and is given by:

$$L_{\text{instant.}} = \frac{N_b^2 n_b f_{\text{rev}} \gamma_r}{4\pi \epsilon_n \beta^*} F, \tag{4.4}$$

where N_b is the number of particles in a bunch, n_b and f_{rev} are defined as in Equation 4.1, γ_r is the relativistic gamma factor, ϵ_n the normalized transverse beam emittance, β^* describes the beta function at the collision point and F is a geometric luminosity reduction factor due to the crossing angle at the interaction point. The peak luminosity for proton-proton-collisions at the LHC is $L_{\text{instant.}} = 2.1 \cdot 10^{34}\,\text{cm}^{-2}\,\text{s}^{-1}$, which are used by the two high-luminosity experiments ATLAS and CMS. The other two experiments use lower luminosities: LHCb is designed for an average running luminosity of $L_{\text{instant.}} = 2 \cdot 10^{32}\,\text{cm}^{-2}\,\text{s}^{-1}$ while for the ALICE experiment it is about $L_{\text{instant.}} = 5 \cdot 10^{26}\,\text{cm}^{-2}\,\text{s}^{-1}$ for Pb-Pb collisions [47].

Each of the four major experiments at the LHC have a different design, focusing on various physics aspects to look for phenomena beyond the Standard Model. The LHCb experiment [49] focuses on flavour physics, specifically targeting decays of b quarks. Since b and $\bar{b}$ hadrons are produced at small angles with the beam pipe at high energies, the detector design is based on a single-arm spectrometer with a high forward angular coverage [50, 51]. ALICE (A Large Ion Collider Experiment) [52, 53] is the only detector specialized in heavy-ion collisions and focuses on studying QCD phenomena, probing the strong interaction in the Standard Model. Its purpose is to study the physics of the strong interactions, including highly exotic states of matter such as the quark-gluon plasma that occurs at high energy densities and temperatures in collisions of heavy nuclei. It has a cylindrical design, containing a general-purpose detector in the "barrel region" for particles having a large angle with

the beam pipe as well as "muon arm" in the forward region covering smaller angles, in particular to measure heavy-quark resonances [54].

The CMS (Compact Muon Solenoid) [55, 56] and ATLAS (A Toroidal LHC Apparatus) [57–59] experiments are general-purpose detectors and focus on the analysis of proton-proton collisions. The key objective of these experiments is to obtain high precision measurements of the SM and to search for BSM phenomena. One distinctive detector component in the ATLAS detector is its air-core toroid system for muon spectrometry, which is about 20 m in diameter. The inner part of ATLAS consists of a tracking system using both silicon and gas-based detectors. For precise identification and measurement of electrons and photons, a highly granular electromagnetic calorimeter based on liquid argon technology is used, whereas the hadronic calorimeter uses steel as absorbing material and plastic scintillators for sampling. To create curvature in the trajectories of charged particles for measuring the momentum, solenoidal magnetic fields are used for the inner part and toroidal fields for the outer part of the detector, which are generated by two dedicated magnet systems. While the CMS and ATLAS detectors study similar phenomena and share key physics objectives, they differ significantly in the employed detector technologies and design. Therefore, they complement each other to be sensitive to a larger variety of different phenomena and to confirm and validate each other's findings. Since this work focuses on the analysis of data from the CMS experiment, its detector components are discussed in more detail in the following sections.

An overview of the LHC complex, including the four discussed experiments and the different pre-acceleration stages can be seen in Figure 4.1.

4.2 CMS Detector Components

The CMS detector [55,56] is a general-purpose detector, designed to probe several aspects of both SM and BSM physics. A key achievement of the CMS collaboration was the discovery of the Higgs boson in 2012, together with the ATLAS collaboration [1, 2, 60], as well as the subsequent measurement of several of its properties. Since in this work, CMS data of the "Run 2" era (2016 to 2018) will be used, the detector parameters described in this chapter are related to the state after the Phase-1 upgrade, which took place in the extended year-end technical stop of the LHC in 2016/2017. The length of the detector is 22 m, it has a diameter of 15 m and weighs 14 000 t. It is designed to identify electrons, muons, photons as well as both charged and neural hadrons. One of the main features of CMS is its eponymous superconducting solenoid, with a 6 m internal diameter and a length of 12.5 m, providing a magnetic field of 3.8 T. The silicon-based tracking detectors as well as the electromagnetic and hadronic calorimeters (ECAL and HCAL) are located within this magnetic field. The outermost detector component are the muon chambers, which is embedded a steel return yoke situated outside of the solenoid magnet.

The typical convention regarding the detector coordinates, which is also followed in this work, is that the longitudinal direction along the beam line is defined as the z axis, whereas the xy-plane is perpendicular to the beam. The y axis is pointing upward and the x axis pointing inward towards the centre of the LHC. The azimuthal angle with respect to the x-axis in the xy-plane is defined as ϕ and the polar angle, measured from the z-axis, is defined as θ. Instead of θ, the pseudorapidity η is often preferred in hadron collider experiments, since differences in η are invariant under Lorentz

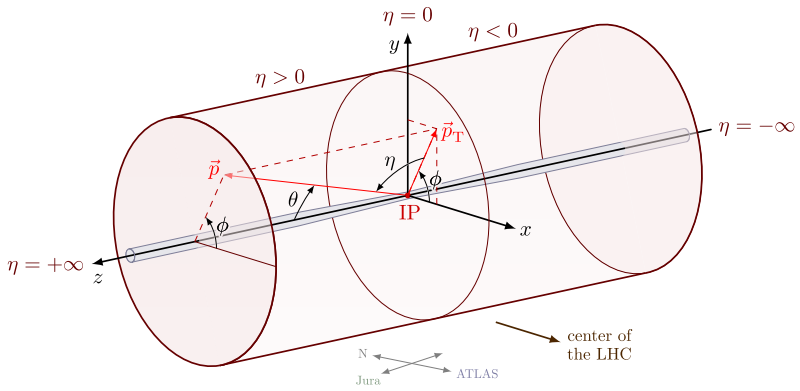


Figure 4.2: Coordinate system of the CMS experiment, shown for a particle with momentum vector $\vec{p}$ [61].

transformations along the longitudinal axis. This is a beneficial feature of the pseudorapidity, since the partons that interact in a collision carry different momentum fractions of their respective hadrons and therefore, the rest frames of the parton-parton systems will have different boosts along z. The pseudorapidity is related to the polar angle by:

$$\eta = -\ln(\tan(\theta/2)). \tag{4.5}$$

The transverse momentum and energy are denoted by p_T and E_T. An imbalance of energy in the transverse plane, which can for example be caused by neutrinos in the final state of a decay, is defined as E_T^{miss}. It is also referred to as missing transverse energy. A schematic of the described coordinate system at the example of a particle with momentum vector $\vec{p}$ can be seen in Figure 4.2. In order to cover a large pseudorapidity range, each detector layer separated in different sub-detectors, covering different ranges in η. The barrel region covers (depending on the subsystem) a range of about $|\eta| < 1.4$, whereas the endcaps cover values of $1.3 < |\eta| < 3.0$. Finally, the forward region covers the $3.0 < |\eta| < 5.2$ range.

The individual components of the CMS detector are described in more detail in the following sections. A schematic showing a detector "slice" can be seen in Figure 4.3.

4.2.1 Tracking System

The CMS tracking system measures the trajectories of charged particles that were generated in the parton interaction [63]. The solenoid magnetic field inside the detector causes these trajectories to be curved due to the Lorentz force, which makes it possible to identify the charge of a particle and measure its momentum. The tracker is silicon-based, using the fundamental principle of the p-n diode operated in reverse mode for the detection of particle hits. When a charged particle is crossing the silicon, it ionizes the material, creating electron-hole pairs that travel to the respective poles of the diode. Thus, a current can be measured that indicates a particle has hit the detector.

The innermost layer of the tracking system is the pixel tracker. It is located close to the detector centre and provides points in three-dimensional space, which enable tracking and reconstruction of vertices at high precision. It is composed of four layers in the barrel region and three disks of silicon sensors that are positioned on either side of the interaction point. In total, it has 124 million readout

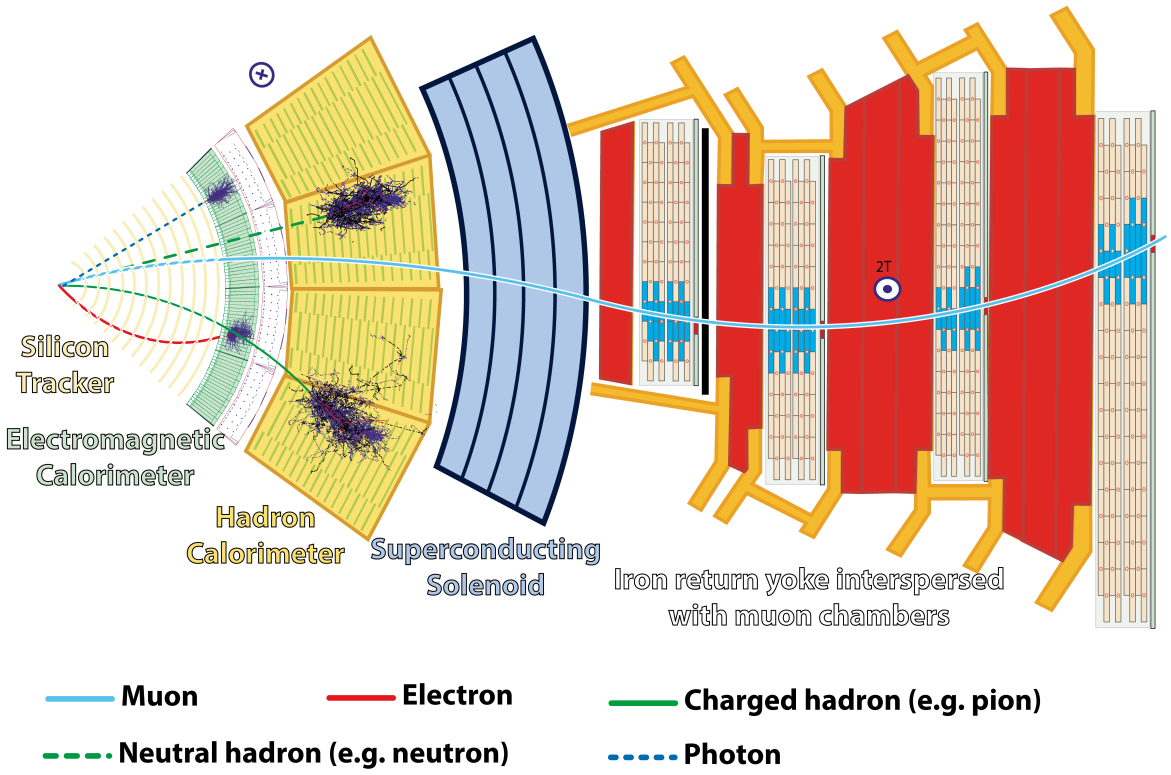


Figure 4.3: A slice of the CMS detector, showing its different components as well as the signatures of each detectable particle [62].

channels with a readout rate at 400 MB/s. The pixel tracker is cooled using a two-phase CO_2 cooling system. The barrel layers are located at radii of 29, 68, 109 and 160 mm and three disks at each end are located at distances of 291, 396 and 516 mm from the detector centre. The foundational building block is a silicon sensor module containing 160 × 416 pixels, each of which is 100 × 150 µm^2 in size. The pitch (distance between the centres of two pixels) is 100 µm [63]. 1856 such modules are contained in the pixel tracker. The spatial resolution – a key performance metric for a tracking detector – depends on various factors, such as the Lorentz angle and whether the particle hits multiple sensors in a layer, such that the measured charge is distributed across neighbouring cells. In the ideal case where only one sensor in a layer is hit, the probability for a particle hitting is $p(x) = 1/d$, a uniform distribution over a sensor with pitch d. The expected value is then given by:

$$\langle x \rangle = \int_{-d/2}^{d/2} x \cdot p(x)\, dx = \frac{1}{d} \int_{-d/2}^{d/2} x\, dx = 0 \tag{4.6}$$

and therefore, the resolution can be computed as:

$$\sigma^2 = \langle x - \langle x \rangle \rangle^2 = \langle x \rangle^2 = \int_{-d/2}^{d/2} x^2 \cdot p(x)\, dx = \frac{d^2}{12} \tag{4.7}$$

$$\sigma = \frac{d}{\sqrt{12}}. \tag{4.8}$$

Therefore, in this case the resolution of the pixel tracker is $\sigma \approx 28.9$ µm.

The outer part of the tracking system is the silicon strip tracker (SST). It consists of 9.3 million silicon strips distributed over 15 148 modules. It is 5 m long and has a diameter of 2.5 m. In the

barrel region, there are ten layers of strip modules, whereas the endcaps contain up to seven rings. The thickness of the silicon sensors varies, being 320 µm and 500 µm for the inner and outer layers, respectively. The distance between the strips (pitch) varies from 80 µm to 205 µm, corresponding to spatial resolutions between 23.1 µm and 59.2 µm. The ratio of the pitch with the strip width is always kept constant at 0.25 [64]. The barrel layer modules measure the coordinates r and ϕ, while the layers in the endcaps measure ϕ and the longitudinal coordinate z. The signals from the strips are processed by an analogue chip that has 128 readout channels for 128 strips.

The transverse momentum resolution of the tracking system for high momentum tracks at around 100 GeV is 1-2 % up to a pseudorapidity of $|\eta| \approx 1.6$. The transverse impact parameter resolution is 10 µm for high p_T tracks.

4.2.2 Electromagnetic and Hadronic Calorimeters

The CMS detector has two calorimeters, an electromagnetic calorimeter (ECAL) for measuring the energy of photons and electrons and a hadronic calorimeter (HCAL) for the energy measurement of both charged and neutral hadrons. The ECAL is a homogeneous calorimeter, which means that the deposition of energy and the sampling occur in the same active medium [65]. This is different from a sampling calorimeter, which is made of alternating layers of high Z material, where the electromagnetic showers are induced, and active media such as scintillators where the energy of the showered particles is then measured. Because some energy dissipates through ionization in the absorbing material, sampling calorimeters typically have a worse energy resolution than homogeneous calorimeters. Since the Higgs decay into two photons was a promising candidate for its discovery, a good energy resolution in the ECAL was of primary importance, which lead to the choice of a homogeneous calorimeter over a sampling calorimeter.

The primary way of measuring photon and electron energies in electromagnetic calorimeters is through electromagnetic showering and subsequent ionization of a scintillating material. An incident high energy photon creates an electron-positron-pair in the field of the high Z nuclei in the material. These again get deflected by the electric fields of the nuclei, emitting high energy photons through Bremsstrahlung. Subsequently, electron-positron pairs are again produced. This showering continues until the energy of the electrons and positrons is too low to sustain the continued pair production process. In this case, they deposit their energy in the form of ionization. In a scintillator, this ionization causes electrons in the material to be excited to a higher energy state and the de-excitation back to the ground state results in the emission of a photon, typically in the visible spectrum. This light can then be measured by photodetectors, which results in an implicit measurement of the energy of the incident particle.

In the case of the CMS ECAL, the homogeneous medium is made of 61 200 lead tungstate ($PbWO_4$) crystals in the barrel region and 7324 crystals in each of the endcaps. The photodetectors used to detect the scintillation light are avalanche photodiodes (APDs) in the barrel and vacuum phototriodes (VPTs) in the endcaps. The main advantages of the lead tungstate material is its high density of 8.28 g/cm^3 and its short radiation length (i.e. the mean distance in the material at which the energy of an electron is reduced by a factor $1/e \approx 36.8\,\%$) $X_0 = 0.89$ cm, which allows for the design of a compact and highly granular calorimeter. The crystals are 230 mm in length, which corresponds to 25.8 X_0. At the front face, they have a cross-section area of 22×22 mm^2 and an area of

$26 \times 26\,\mathrm{mm}^2$ at the rear face. The front faces of the crystals are located at a radius of 1.29 m from the detector centre. In total, the crystal volume in the barrel is $8.14\,\mathrm{m}^3$ with a weight of 67.4 t.

A key performance metric of calorimeters is their energy resolution, which can be parameterized by:

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S}{\sqrt{E}}\right)^2 + \left(\frac{N}{E}\right)^2 + C^2, \tag{4.9}$$

where S is a stochastic term, N a noise term and C a constant term. The main contributors to the S term are the event-to-event fluctuations in lateral shower containment as well as the photostatistics. The noise term quantifies the impact of electronics, digitization and pileup noise. The constant term describes the contribution of intercalibration errors, non-uniformity of the light collection and energy leakage from the back of the crystals. Using measurements in a dedicated test beam, the energy resolution for a barrel module has been found to be:

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{2.8\,\%}{\sqrt{E}}\right)^2 + \left(\frac{0.12}{E}\right)^2 + (0.3\,\%)^2 \tag{4.10}$$

The energy resolution in Run 2 has been found to be comparable to Run 1 [66]. Concerning different physics objects, the resolution for electrons with transverse energy $E_T \approx 45\,\mathrm{GeV}$ from Z boson decays was found to be better than 2 % in the central part of the barrel region ($|\eta| < 0.8$) and between 2 % and 5 % elsewhere. The energy resolution for photons from Higgs boson decays with $E_T \approx 60\,\mathrm{GeV}$ varies in the barrel region between 1.1 % to 2.6 % and from 2.2 % to 5 % in the endcaps.

The CMS hadron calorimeter (HCAL) [67] is used to measure the energies of charged and neutral hadrons, as well as neutrinos and other exotic particles resulting in missing transverse energy E_T^{miss}. Therefore, the HCAL is also playing a crucial role in jet identification and measurement. In general, the showering processes in hadron showers are significantly more complex compared to electromagnetic showers due to the additional nuclear interactions that take place. The main interaction processes are the creation of new hadrons through hadronization, nuclear spallation and nuclear de-excitation causing the evaporation of soft nucleons. In hadronic showers, there is a significant "invisible fraction" of energy from interactions where the energy gets absorbed as nuclear binding energy or recoil in the material. Additionally, neutral pions (π^0) can be created, which decay exclusively to photons that then create an electromagnetic shower. Therefore, these pions deposit all their energy via electromagnetic processes. Assessing the fraction of electromagnetically deposited energy f_{EM} as well as the components of the different nuclear interactions is challenging and makes complex Monte Carlo simulations necessary for proper calibration.

The innermost part of the HCAL barrel is located at a distance of 1.77 m and reaches up to 2.95 m from the centre of the detector. In order to entirely absorb incident hadrons with a high probability, an additional hadron calorimeter is placed outside of the solenoid to measure hadrons which passed through the inner HCAL. Contrary to the CMS ECAL, the HCAL is a sampling calorimeter, using alternating layers of brass absorber plates and plastic scintillators. The brass absorbers are composed of 70 % Cu and 30 % Zn and have a density of $8.53\,\mathrm{g/cm}^3$. This results in a radiation length of 1.49 cm and a nuclear interaction length of 16.42 cm. The latter describes the mean distance that a hadronic particle can travel until an inelastic nuclear interaction takes place.

The HCAL energy resolution obtained in dedicated test beams for single pions is [68]:

$$\frac{\sigma}{E} = \frac{52.9\,\%}{\sqrt{E}} + 5.7\,\%. \tag{4.11}$$

Considering measurements of particle jets, the jet energy resolution that is typically achieved is 15 % at 10 GeV, 8 % at 100 GeV and 1 % at 1 TeV when both tracking and calorimeter information is used [69].

4.2.3 Muon System

The CMS muon system is designed to identify and trigger on muons produced in the proton collisions as well as measure their momentum. A good momentum resolution is achieved by the strong magnetic field created by the solenoid magnet. In total, three different types of gaseous detectors are used to identify muons [70]. The muon system consists of a barrel section and two endcaps. In the barrel region, drift tube (DT) chambers are used as detectors, organized in 4 stations that form concentric cylinders. The 3 innermost cylinders contain 60 chambers each, while the outer cylinder contains 70. In total, they contain about 172 000 wires, each with a length of around 2.4 m. The gas mixture used is about 85 % Ar and 15 % CO_2. In order to increase detection efficiency, the drift cells in each chamber are arranged in an overlapping way, where each cell is shifted by half a cell width with respect to its neighbour. Each chamber is made of 2 or 3 "superlayers" (SLs), consisting of 4 layers of rectangular drift cells. In the outer 2 SLs, the wires are arranged parallel to the z direction to provide a measurement in the $r - \phi$ plane, while in the inner SLs the wires are orthogonal to the beam line and measure the z coordinate.

For the endcaps, which detect muons with a pseudorapidity of $0.9 < |\eta| < 2.4$, cathode strip chambers (CSC) are used. 4 stations are contained in each endcap, containing 468 CSCs in total. The chambers are arranged perpendicular to the beam line, measuring the $r - \phi$ plane and are based on multiwire proportional chambers, containing 6 anode wire planes and 7 cathode panels. The CSCs provide both muon measurement and triggering capabilities.

Finally, resistive plate chambers (RPCs) provide an additional, independent trigger system parallel to the DTs and CSCs for improved background rejection and beam crossing time measurements. The RPCs are gaseous parallel-plate detectors operated in avalanche mode and have a good time resolution, while the position resolution is more coarse compared to the DTs and CSCs. In particular an RPC can tag the time of an ionising event in a significantly shorter time than the 25 ns between two bunch crossings. In the barrel region, 6 layers of RPCs are included, 2 in each of the first 2 stations and 1 in each of the last 2 stations. In the endcap region, 1 layer of RPCs is contained in each of the first 3 stations. To further improve the muon momentum resolution, a dedicated alignment system measures the positions of the muon detectors with each other and with respect to the inner tracker. Using this design, a very high reconstruction efficiency of 95-99 % is reached throughout a large angular region of $10° < \theta < 170°$. The momentum resolution of highly energetic muons is between 1 % and 8.25 % in the barrel region and between 2 % and 13.1 % in the endcaps for p_T up to 2000 GeV using combined tracker and muon system information. This resolution also depends on the alignment of tracker and muon system [71]. Due to its high efficiency and performance, the muon system is an integral part of the CMS detector and allows for a good identification of processes that involve decays into muons,

one of which is the $H \to ZZ^* \to 4\mu$ decay channel of the Higgs boson, which was one of the major channels considered in its discovery.

4.3 Trigger and Event Reconstruction

4.3.1 The CMS Trigger System

At the LHC, bunch crossings occur about every 25 ns, leading to a collision rate of approximately 40 MHz. The reconstruction and subsequent storage of each of these collision events, however, is not feasible due to computational and hardware storage limitations. Therefore, the rate of events to analyze further has to be reduced significantly, which is achieved in CMS using a two-tier trigger system [72, 73]. First, the hardware-based Level-1 Trigger (L1T) [74] reduces the event rate down to 100 kHz. Second, the software-based High-Level Trigger (HLT) further reduces this rate, which was around 1 kHz during the Run 2 data taking period.

The L1T decision is based on a coarse subset of the read out data. Every event is analyzed by the muon and calorimeter triggers. The energy deposits of all three muon system components, namely CSCs, DTs and RPCs are processed by a pattern comparator or a system of segment- and track-finders and the subsequent information is combined in the global muon trigger (GMT). The calorimeter information of the HCAL and ECAL are first regionally processed by the Regional Calorimeter Trigger (RCT) and then by the Global Calorimeter Trigger (GCT). The global trigger (GT) then combines the information of the various objects provided by the RCT and GCT. Around 400 rules are evaluated for the trigger objects, applying a logical "OR" for all of them, which constitutes the L1 trigger menu.

The HLT is a server farm that contained around 26 000 CPU cores during Run 2, running the CMS analysis software which decides based on the entire detector information whether an event should be stored for offline analysis [75]. The software is applying optimized reconstruction algorithms, which run around 100 times faster compared to their offline counterparts. It consists of hundreds of HLT paths that target a variety of event topologies.

4.3.2 Particle Flow

The fundamental unit in CMS data analysis is a collision *event*, containing the measured detector information of all the particles that have been created in the corresponding bunch crossing. The detector measures these indirectly, using tracks for momentum and trajectory measurements and calorimeter clusters for measurements of their energy. To reconstruct the particles in an event, complex reconstruction algorithms are necessary. In CMS in particular, this is done with the Particle Flow (PF) algorithm [76, 77], which uses global information from all the subdetectors to provide a comprehensive list of all final-state particles in an event, also referred to as "PF candidates".

The approach of the PF algorithm is based on a global event description, correlating the fundamental detector signals (tracks and clusters) from the respective subdetector components. The algorithm starts with the tracks and vertices from charged particles. A key challenge in track reconstruction is the high combinatorics involved in assigning the tracker hits to a track. Therefore, an approach based on Kalman Filtering (KF) [78] is used for a three-stage track reconstruction: initially, seeds are generated with a few hits that are compatible with the trajectory of a charged particle. Second,

the trajectory is built to gather hits from all tracker layers along its path. Finally, a fit is done to determine the key properties of the charged particle, namely its point of origin, transverse momentum and direction. To further increase the tracking efficiency while at the same time achieving a similar misidentification rate, an iterative tracking approach was chosen, where the described track finding algorithm was applied in several iterations, progressively adding to the total efficiency. Overall, 10 iterations are used by the algorithm.

After the track reconstruction is done, the calorimeter clusters are built. The clustering algorithm obtains the energy and direction of stable neutral particles, such as photons and neutral hadrons and separates them from the energy deposits of charged hadrons. It also identifies and reconstructs electrons and the corresponding photons caused by bremsstrahlung and improves the energy measurements of charged hadrons with low-quality but high-p_T tracks, for which the track parameters could not be accurately determined [77]. The clustering algorithm is applied to each subdetector separately. As a first step, cluster seeds are assigned to cells with an energy larger than a pre-defined threshold and with higher energy than their neighbouring cells. The second step is the growth of topological clusters from the seeds by aggregation of cells with at least one corner shared with a cell already in the cluster and with an energy of at least twice the noise level. Finally, to reconstruct the clusters within a topological cluster, an algorithm based on a Gaussian mixture model is used. It is based on the assumption that the M individual cells in the topological cluster originate from N Gaussian energy deposits, where N is the number of seeds. The algorithm is an iterative maximum likelihood fit, which is repeated until convergence, after which the positions and energies of the obtained Gaussians are assigned as the cluster parameters.

Since a particle is expected to create several PF elements in the different subdetectors of CMS, the reconstruction continues with a link algorithm, connecting this subdetector information. The link algorithm compares the pairs of elements that are closest in the (η, ϕ) plane, obtained with a k-dimensional tree. If two elements are linked by the algorithm, it defines a distance between them which quantifies the quality of the link. The algorithm then builds PF blocks of elements that are either associated by a direct link or by an indirect link through common elements. The specific conditions on the basis of which a link is created depend on the nature of the linked elements. A link between a track in the tracking system and a calorimeter cluster is created by first extrapolating the track from the last measured tracker hit to the point in the ECAL corresponding to the expected maximum of a longitudinal electron shower profile and to the HCAL corresponding to one interaction length. The track is linked to a cluster when its extrapolated position is within the cluster area, which is defined as the areas of all its cells in the (η, ϕ) plane for the barrel region and the (x, y) plane for the endcaps. To mitigate gaps in the cluster, the cluster area is enlarged by up to one cell in each direction. The link distance that is assigned by the algorithm is defined between the extrapolated position and the cluster position in (η, ϕ).

To compute the energy of bremsstrahlung photons emitted by electrons, tangents to the tracks are extrapolated from their intersection with each of the tracker layers to the ECAL. Clusters are linked to the track if the extrapolated position is within the cluster, as defined above. Furthermore, a dedicated conversion finder was developed that links tracks from additional electron-positron pairs that might have been created by pair production of the bremsstrahlung photon to the original track [79]. Links between ECAL and HCAL are created when the cluster position in the ECAL is within the cluster

envelope of the HCAL. The link distance is here defined as the cluster positions in the (η, ϕ) plane. Finally, a link between tracks in the central tracker and hits in the muon system is established. Muon tracking is based on three different types, depending on various criteria: For standalone muons, DT and CSC hits are clustered forming track segments, which are used as seeds for subsequent pattern recognition in the muon spectrometer to gather all DT, CSC and RPC hits along the trajectory. A final fit yields a standalone-muon track. Each standalone-muon track is matched to a track in the inner tracker and the two are combined in a fit to form a global muon track. Additionally, inner tracks with $p_T > 0.5\,\text{GeV}$ and $p > 2.5\,\text{GeV}$ are extrapolated to the muon system. If at least one segment in the muon system matches the extrapolated track within a certain distance in (x, y), they are assigned as tracker muon.

After the link algorithm created the PF blocks, the particle candidates are identified in each block. At first, the muon cnadidates are identified and reconstructed and all corresponding tracks and clusters are removed from the block. Then, electrons are reconstructed, including the collection of bremsstrahlung photons. Again, the corresponding tracks and ECAL clusters are removed from the PF block. The remaining elements are reconstructed using a cross-identification approach of charged hadrons, neutral hadrons and photons that originate from parton fragmentation, hadronization and decays within jets. When all blocks have been processed and all particles identified, the algorithm stops after a final processing step to further reduce misidentification and misreconstruction of particles. The output of the algorithm is a list of reconstructed particles, which are the previously mentioned PF candidates.

4.3.3 Jet Clustering

A key object in the analysis of particle physics collision events is the jet. Jets are collimated sprays of particles that originate from hadronization processes of quarks or gluons. At the CMS experiment, jets are clustered from the PF candidates using the anti-k_T algorithm [80]. A desirable feature of any jet clustering algorithm is infrared and collinear (IRC) safety. A clustering algorithm is infrared safe when the resulting jet is not changed by the radiation of soft particles from its constituents. Collinear safety relates to the robustness of the clustered jets with respect to collinear splittings of the input particles. IRC safety is an important feature, since a jet should convey information of the initiating particle, which decays in a hadronic showering process that can contain several collinear splittings before the start of the hadronization. If such a splitting would cause the algorithm to produce a second jet, the information of the initial particle would be distorted. Additionally, the clustering should not be affected by small perturbations, for example caused by detector effects.

The anti-k_T algorithm is an IRC safe, soft-resilient clustering algorithm, which means that the resulting jet shapes are not influenced by soft radiation. The algorithm is based on the distances d_{ij} between entities i and j, where an entity is either a particle or an intermediate clustering result called pseudo-jet. This distance is defined as:

$$d_{ij} = \min(k_{ti}^{2p}, k_{ti}^{2p}) \frac{\Delta_{ij}^2}{R^2}, \tag{4.12}$$

$$d_{iB} = k_{ti}^{2p}, \tag{4.13}$$

where $\Delta_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$, d_{iB} describes the distance between entity i and the beam B

and k_{ti}, y_i and ϕ_i correspond to the transverse momentum, rapidity and azimuthal angle of entity i, respectively. The radius parameter is defined as R and the parameter p describes the relative power between the energy and geometrical (Δ_{ij}) scales. The algorithm starts by computing d_{ij} between all pairs of entities and combines those entities i and j where d_{ij} is minimal into a pseudo jet. If the minimal distance is d_{iB}, however, it defines i as a jet and removes it from the list of entities. Then, the distances are recomputed and the procedure is repeated until no entities are left. The parameter p is chosen prior to the clustering and for the anti-k_t algorithm it is set to $p = -1$, hence its name. The R parameter defines the radius of the resulting jet cone and can also be chosen prior to the clustering. In CMS data analysis, typically two values are used. Regular jets use a value of $R = 0.4$, also referred to as "AK4 jets", whereas large-radius jets are also used with a value of $R = 0.8$, also referred to as "AK8 jets". The clustered jets and their corresponding kinematic information can then be used for further analysis.

Chapter 5

Foundations of Machine Learning

5.1 Classical Algorithms

Machine learning is the field of extracting intricate patterns from data using a variety of algorithms. Given a set of previously obtained data X, the main objective of most of these methods is to learn a function $f(X)$ that maps the input data to a target variable of interest, Y. Once the function is learned, it can be applied to new, previously unseen samples of data originating from the same distribution $\tilde{X} \sim X$ to predict the corresponding targets $\tilde{Y}$. Typically, the function f also depends on a set of parameters $\boldsymbol{\theta}$, such that the desired mapping becomes:

$$f_{\boldsymbol{\theta}}(X) \to Y \tag{5.1}$$

Three major learning paradigms exist, depending on the task to optimize and the prior knowledge of the target variable: In supervised learning, for each data point $x_i \in X$, the exact value of the corresponding target $y_i \in Y$ is known. In unsupervised learning, the target values y_i are entirely unknown and truth-level information is inaccessible throughout the entire learning process. Finally, in reinforcement learning, the task is to optimize the actions of an agent in a dynamic environment by maximizing the cumulative reward. Furthermore, several other paradigms exist that combine characteristics of both supervised and unsupervised learning. In semi-supervised learning, there exists truth-level information for a typically small set of data points, while for most other samples in the data set they are unknown. Semi-supervised learning then combines the information of both kinds of datapoints to gain performance with respect to a classifier that was trained either on the truth-valued samples or on the samples with unknown truth information only. In self-supervised learning, the truth value information is not available, but can be obtained from the data directly. This can be done for example by using domain knowledge, or by exploiting known correlations between input variables and the target variable.

Machine learning algorithms are suited to solve a plethora of complex tasks in a variety of fields, such as computer vision or natural language processing. Two major categories of tasks exist that machine learning is frequently used for: classification and regression. In regression, the target variable of interest, Y, is a continuous variable and the task is to predict the values of each datapoint y_i for each corresponding input variable x_i. For example, one could try to predict the amount of energy deposited in a detector cell from input variables such as particle type, particle energy, angle of inci-

dence, depth of material etc. In classification, each datapoint belongs to a certain class of objects, which are represented numerically by integer numbers, typically starting from 0. I.e. if there are N classes, then the target variable of the individual datapoints are represented as $y_i \in 0, ..., N-1$. Thus, a classification algorithm learns to predict the class of each datapoint y_i from input variables x_i. One example in a particle physics scenario is quark-gluon-tagging, where the task is to distinguish particle jets that originate from gluons and jets which were initialized by a quark [81]. Here, the algorithm would learn the assignment of correct class label y_i (0 or 1) based on a set of input variables x_i, which could for example be the transverse momentum p_T , the pseudorapidity η and the mass of a jet etc.

Various algorithms exist to solve regression and classification tasks and their choice depends on several factors, in particular the structure of the input data. For example, there are machine learning algorithms that are specifically designed to solve problems in computer vision, such as convolutional neural networks [82], while others are designed for natural language processing tasks, such as transformers [83]. In many cases, however, the objective is to run a classification or regression on a set of entities, e.g. particle jets or entire particle collision events, for which different variables or "features" have been recorded. In this case, the input data can be thought of as a table, where each entity represents a row and each feature a column. Data in this format is also referred to as "tabular data". The key difference between tabular data and data represented in other ways, such as images, point clouds or sentences, is that there is typically no inherent structure in the data that could be exploited by a specific algorithm. In images for example, neighbouring pixels are highly correlated while pixels far away are usually uncorrelated. In a data table on the other hand, no such correlation between table cells can be assumed a priori, and operations such as switching feature columns should not affect the classification result, which is not the case for pixel columns in an image. Surprisingly, tabular data is often referred to as "structured data" in machine learning literature, while other data representations are referred to as "unstructured data".

5.1.1 Linear Models

For solving machine learning tasks based on tabular data, several algorithms could be used, from simple models to complex deep learning architectures. Algorithms that precede the era of deep learning are often referred to as "classical machine learning" and they still enjoy a high popularity, in particular for tabular data problems [84]. The least complex class of models are linear models [85]. For regression, these models simply fit a linear equation to the data, which can then be used to obtain the corresponding value for the variable of interest. In particular, for a data point $\mathbf{x}$ containing K input features, the equation becomes:

$$f(\mathbf{x}) = \theta_0 + \sum_{k=1}^{K} \theta_k x_k \tag{5.2}$$

where the θ_k are the coefficients corresponding to the k-th input feature and θ_0 describes the intercept of the linear equation. The function is learned by optimizing the coefficients such that the residuals between the fitted curve and the variable of interest of the datapoints y_i are minimized:

$$\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} \|X\boldsymbol{\theta} - Y\|_2^2 \tag{5.3}$$

In addition to linear regression, linear models can also be used for classification. This is called logistic regression and essentially works the same as the linear regression model in logit space [86].

For simplicity, the focus of this section will be on the binary classification case, but the problem naturally extends to a multi-class classification problem. Classification models typically do not output the predicted class assignment of a datapoint directly, but a probability referring to the datapoint belonging to a particular class. In logistic regression, the probability can be obtained from fitting a line in logit space and then using the logistic function to map back into probability space. The probability of a data point $\mathbf{x}$ given the weights $\boldsymbol{\theta}$ can be computed as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1 + e^{-(\theta_0 + \sum_{k=1}^{K} \theta_k x_k)}} \tag{5.4}$$

To find the optimal model for a given data set, the coefficients are optimized such that the negative logarithmic likelihood over the entire data set is minimal:

$$\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} - \sum_{i=1}^{N} (1 - y_i) \ln(1 - p_{\boldsymbol{\theta}}(\mathbf{x}_i)) + y_i \ln(p_{\boldsymbol{\theta}}(\mathbf{x}_i)) \tag{5.5}$$

Equation 5.5 is also referred to as binary cross entropy and is frequently used in binary classification problems.

5.1.2 Decision Trees

While linear models are suited to solve both classification and regression tasks in a simple way, their main shortcoming is the inability to model non-linear functions. An alternative algorithm in classical machine learning that can also learn non-linear functions is a decision tree and similar to linear models, they can be used to solve both classification and regression problems [87,88]. Decision trees consist of two kinds of nodes: First, there are internal nodes, which are nodes that have connections – so-called branches – to nodes further down the tree. In decision tree learning, binary trees are typically used, which means that each internal node is connected to exactly two nodes in the next level of the tree. Second, there are leaf nodes, which are ”end” nodes that do not connect to any other nodes. When training a decision tree, the input samples are split on learned decision boundaries at each internal node and eventually end up at a leave, which determines the output of the model. The way the decision boundaries are chosen depends on the task that should be solved. In a regression task, the following algorithm is executed at each internal node:

1. Take the p-th input feature of the data set and sort all values such that $x_{i,p} \leq x_{i+1,p}$.

2. For each pair of consecutive unique values, compute the mean value. These means (t_j) constitute the possible thresholds to split the data.

3. For $j = 0$ compute the average values for datapoints below and above the threshold:

$$\bar{x}_{\text{low},j=0,p} = \frac{1}{N_{\text{low}}} \sum_{l} x_{l,p} \forall x_{i,p} < t_0$$
$$\bar{x}_{\text{high},j=0,p} = \frac{1}{N_{\text{high}}} \sum_{m} x_{m,p} \forall x_{i,p} > t_0$$

4. Compute the sum of squared residuals between the values and the corresponding mean:

$$\mathrm{SSR}_{j=0,p} = \sum_{l=1}^{N_{\mathrm{low}}} (x_{l,p} - \bar{x}_{\mathrm{low},p})^2 + \sum_{m=1}^{N_{\mathrm{high}}} (x_{m,p} - \bar{x}_{\mathrm{high},p})^2$$

5. Repeat steps 3 and 4 for all other $j > 0$

6. The threshold value that yields the lowest SSR is defined as the optimal threshold for the p-th feature:

$$\mathrm{SSR}_p = \min_j \mathrm{SSR}_{j,p}$$

7. When multiple features are present, all steps above have to be repeated for all available features $k \neq p$ and the feature yielding the lowest SSR overall is chosen for the split:

$$p_{\mathrm{opt}} = \arg\min_p SSR_p$$

When the optimal feature and threshold has been chosen, the data set is split up: all datapoints that have lower values in said feature go in one branch (typically left) to the next tree level, while datapoints with higher values go in the other (typically right) branch. Each branch leads to another node, which can be an internal node where the procedure outlined above will be repeated for the remaining datapoints, or a leaf node where the average of the datapoints is defined as the final output. The point at which the further splitting of samples is stopped and a node is declared a leaf depends on several factors and is often considered a hyperparameter that has to be optimized. The simplest stopping criterion is the situation when a tree is pure, i.e. when all leaves yield residuals of zero. However, this scenario is often not desirable since such trees tend to overfit quickly on the data they were trained on. Typically either a maximum depth, a maximum number of leaf nodes or a minimum number of samples per leaf is defined as a stopping criterion.

For classification, the tree is built in a similar fashion as in the regression task: Samples are split at each internal node and for each feature, all possible thresholds are used to determine the optimal feature and threshold values. However, instead of minimizing the SSR, a different value is typically used in classification, namely the Gini impurity. Given a threshold t, the number of samples below the threshold that have class $y_l = 0$ are denoted as $N_{\mathrm{low},0}$ and those with class $y_l = 1$ as $N_{\mathrm{low},1}$. The Gini impurity for samples below the threshold is then given by:

$$I_{G,\mathrm{low}} = 1 - \left(\frac{N_{\mathrm{low},0}}{N_{\mathrm{low},0} + N_{\mathrm{low},1}}\right)^2 - \left(\frac{N_{\mathrm{low},1}}{N_{\mathrm{low},0} + N_{\mathrm{low},1}}\right)^2 = 1 - q_{\mathrm{low},0}^2 - q_{\mathrm{low},1}^2 \tag{5.6}$$

where $q_{\mathrm{low},c}$ describes the relative frequency of class c amongst samples below the threshold. The same calculation is repeated for samples above the threshold:

$$I_{G,\mathrm{high}} = 1 - q_{\mathrm{high},0}^2 - q_{\mathrm{high},1}^2 \tag{5.7}$$

and finally, the Gini impurity of the split is calculated as the weighted sum of the impurities for samples above and below the threshold:

$$I_G = \frac{N_{\text{low}}}{N} I_{G,\text{low}} + \frac{N_{\text{high}}}{N} I_{G,\text{high}} \tag{5.8}$$

As for the regression, all possible thresholds for all available features p are scanned, and the split with the lowest Gini impurity is finally chosen for the current split. The splitting at internal nodes continues until either all leaf nodes are pure, meaning they only contain samples from a single class, or until one of the previously discussed stopping criteria is met. Finally, the output of a leaf is determined by the majority class of the samples that ended up in it.

While decision trees are suited to solve nonlinear problems, they also have weaknesses that need to be taken into account: Decision trees often suffer from poor generalization performance. This means that, while a good performance is achieved on the samples that the tree was trained on, the performance on an independent sample from the same source is significantly lower. This phenomenon is also referred to as overfitting. Also, decision trees are often sensitive to small changes in the training data, since these may cause different data splittings to occur, which in turn could result in a different tree. Also, even though a decision tree can deal with nonlinear problems, it is still piece-wise constant and therefore is also limited when capturing highly complex non-linear relationships. To overcome these problems, an entire class of models exist that combine multiple individual trees into a single, large model. These models are referred to as ensemble methods and will be discussed in the following chapter.

5.2 Ensemble Methods

5.2.1 Introduction

In the previous chapter, decision trees and their advantages and disadvantages for solving complex problems were outlined. Another class of algorithms exists that improves on the difficulties of individual decision trees by combining multiple trees into a single, large ensemble model. Methods based on ensembles of trees are also referred to as ensemble models and two major classes of ensemble models will be discussed in this chapter: bagging models and (gradient) boosting models.

5.2.2 Bagging

As previously discussed, a key problem with individual decision tree models is their high variance, which often leads to significant overfitting. The general idea of ensemble models is to create a large variety of different decision tree models, which individually might tend to overfit, but then take the average of these predictions such that individual errors cancel out. Since decision trees as discussed in the previous chapter are built in a deterministic way, the randomness needs to be artificially introduced by the algorithm. This is often done using a method called ”bagging” [89], which is a combination of ”bootstrapping” and ”aggregating”. The bootstrapping is done by drawing a sample of datapoints with replacement from the training set, and training an individual tree from the bootstrapped sample. This procedure is repeated for each tree in the ensemble, which leads to a variety of different trees in the final model. Additionally, another source of randomness is often introduced: when constructing the tree, only a random subset of all available features is considered to determine the splittings at internal nodes. This method is often referred to as the random subspace method [90]. The aggregating is

done when all individual trees in the ensemble have been trained: the output of the final model is the average of all the individual tree outputs. For classification tasks, the output class is typically chosen by majority vote of the ensemble. For regression tasks, the output values of each of the trees are averaged.

Bagging can be used with a variety of classification or regression algorithms and is a general technique to reduce variance in the results of individual models. A commonly used method based on ensembles of trees that uses bagging is the random forest [91]. Using random forests instead of individual trees not only improves the generalization performance of the model, but also often turns out to improve the performance overall. One reason for this is that ensembles are more complex models that as a whole can capture non-linear effects in the data better than an individual tree. Additionally, since each tree is built separately, the training of the ensemble can be parallelized for efficient computation. While random forests are superior to individual trees in many regards, still challenges do exist that leave room for further improvements: Random forests are typically made of deep trees that would overfit individually. Since building deep trees is computationally expensive, random forests can take a long time to fit, even when using parallel computing. Another problem is that trees are fit individually, without the possibly useful information that other trees in the ensemble have learned. This means that often a large amount of trees is needed to achieve a satisfactory performance. To further improve ensemble methods with respect to said problems, another family of algorithms, called (gradient) boosting, was developed. These models are amongst the most popular in modern machine learning applications [84] and thus will be discussed in detail in the following chapter.

5.2.3 Boosting

Similar to the previously discussed ensemble models, boosting methods are based on an ensemble of different individual models. However, the fundamental idea behind boosting is to use an ensemble of many ”weak” models and training them in an iterative fashion, such that errors made by a previous model can be corrected by the following models. This stands in contrast to bagging models such as random forests, since these models are trained individually, without knowledge of the errors other models make. Additionally, bagging models train an ensemble of ”strong” models (e.g. deep trees) that individually overfit the training data, while boosting trains a combined ensemble of ”weak” models (e.g. shallow trees or ”stumps”) that individually underfit. Several variants of boosting algorithms exist and in this chapter, three of the mainly used variants are discussed: AdaBoost [92], XGBoost [93] and LightGBM [94]. Since the studies following this chapter mainly relate to classification problems, only the case of classification will be discussed in detail. However, all of the mentioned boosting methods are also suitable to solve regression problems in a similar fashion.

AdaBoost

The AdaBoost algorithm [92], which is shorthand for ”adaptive boosting”, combines weak learners in a way such that they can adapt to samples that are difficult to classify in the training process and thus improve in predictive performance. In AdaBoost, the individual learners that are used are typically very weak, often achieving only slightly better than random performance. The learners are based on so-called decision stumps, which are trees with a single internal node split and two leafs. During training, the individual samples are assigned weights that signify which of them were difficult to

classify by the previous model. At the beginning of the training (i.e. at the iteration indexed by $j = 0$), all samples x_i are assigned equal weights, $w_{j=0,i}$:

$$w_{j=0,i} = \frac{1}{N} \; \forall \; i \tag{5.9}$$

where N denotes the total number of samples in the training set. Just as described in subsection 5.1.2, the stump is now built using a criterion such as the Gini impurity to split the samples into two leaf nodes. Now, the total error of the stump is computed as the sum of weights of all misclassified samples. Let y_i be the class label of the datapoint x_i and $\hat{y}_i$ the class label that is predicted by the stump. The total error for the current stump with index $j = 0$ is then given by:

$$\varepsilon_{\text{tot.},j=0} = \sum_i w_{0,i} \; \forall \; i \text{ , where } y_i \neq \hat{y}_i \tag{5.10}$$

Then, a weight $\gamma_{j=0}$ is computed for the current stump, which will be used to calculate the entire model output later:

$$\gamma_{j=0} = \frac{1}{2} \log \left(\frac{1 - \varepsilon_{\text{tot.},j=0}}{\varepsilon_{\text{tot.},j=0}} \right) \tag{5.11}$$

the γ_j weight is designed such that stumps with a small error get a higher contribution to the final output compared to stumps with a larger error.

Next, the sample weights will be adjusted such that they are decreased for correctly classified samples and increased for incorrectly classified samples:

$$\widetilde{w}_{j+1,i} = w_{j,i} \, e^{\gamma_j} \; \forall \; i \text{ , where } y_i \neq \hat{y}_i \tag{5.12}$$

$$\widetilde{w}_{j+1,i} = w_{j,i} \, e^{-\gamma_j} \; \forall \; i \text{ , where } y_i = \hat{y}_i \tag{5.13}$$

Finally, the computed weights are normalized:

$$w_{j+1,i} = \frac{1}{\sum_{i=1}^{N} \widetilde{w}_{j+1,i}} \, \widetilde{w}_{j+1,i} \tag{5.14}$$

Now, there are two possibilities to use the updated weights in the next training iteration: One option is to resample the set of training datapoints X according to the computed weights. This will create a data set where duplicates of previously misclassified samples will occur more frequently relative to those that were correctly classified. Therefore, the misclassified samples have a higher impact on the classifier training and will force the next decision stump to focus on classifying them correctly. A second option is to use a weighted Gini impurity by taking into account the sample weights in the computation of the relative frequencies. Let c be the class to consider and N_c the number of samples with class c in the current node and N_{node} the number of total samples in the current node, then the weighted relative frequency $q_{w,c}$ of class c becomes:

$$q_{w,c} = \left(\frac{\sum_{i=1}^{N_c} w_{j+1,i}}{\sum_{i=1}^{N_{\text{node}}} w_{j+1,i}} \right) \tag{5.15}$$

By plugging the weighted relative frequency into Equation 5.6 and Equation 5.7 for the child

nodes below and above the current threshold, respectively, the weighted Gini impurity for each node can be computed as

$$I_{G,w,\text{high}} = 1 - q^2_{\text{high},w,0} - q^2_{\text{high},w,1} \tag{5.16}$$

$$I_{G,w,\text{low}} = 1 - q^2_{\text{low},w,0} - q^2_{\text{low},w,1} \tag{5.17}$$

and then the combined weighted Gini impurity for the entire split becomes

$$I_{G,w} = \sum_{i=1}^{N_{\text{low}}} w_{j+1,i}\, I_{G,w,\text{low}} + \sum_{k=1}^{N_{\text{high}}} w_{j+1,k}\, I_{G,w,\text{high}} \tag{5.18}$$

This criterion is then used to build a new stump and the procedure outlined previously is repeated. The training can again end based on different conditions, such as reaching a predefined maximum number of stumps. Finally, the model outputs need to be computed. This is done using weighted majority vote over all stumps based on the error weight that was assigned during the training in Equation 5.11. Let γ_s be the weight of stumps s that predict the class label $\hat{y} = 0$ for a new datapoint x and γ_t the corresponding weight of stumps t that predicts label $\hat{y} = 1$. The decision of the entire AdaBoost model is then given by:

$$\hat{y} = \Theta\left(\sum_t \gamma_t - \sum_s \gamma_s\right) \tag{5.19}$$

where Θ is the Heaviside function.

Gradient Boosting

AdaBoost constitutes a powerful model for classification and regression tasks that iteratively improves on a given problem by adjusting weights of samples that are difficult to classify. However, one feature that is often desired in practice is to be able to minimize arbitrary loss functions when training a model. This is a capability that AdaBoost lacks and a new family of models was developed to include this feature in boosted ensemble learning: gradient boosting algorithms. Before outlining the specific implementations of modern gradient boosting methods, the foundational algorithm [95] will be discussed. The algorithm is described as follows:

The default implementation of the gradient boosting algorithm can be used with any loss function and can be applied to any task. For binary classification problems, which are the main focus of this work, the binary cross-entropy described in Equation 5.5 is typically used. The algorithm starts out with defining a constant value that minimizes the loss function for the initial model of the ensemble, $f_0(x)$. For the BCE loss, the optimal constant value minimizing it is the logit of the observed class probabilities in our data:

$$f_0(x) = \text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \tag{5.20}$$

where p is the class probability of class $y = 1$. In general, gradient boosted trees for classification operate in logit rather than probability space. Next, the individual tree learners are constructed. This is done by computing the residuals as the negative gradient of the loss with respect to the output of

Algorithm 1: Gradient Boosting

Input: Data $(x_i, y_i)_{i=1}^n$ and a differentiable loss function $\mathscr{L}(y_i, f(x))$
Output: The fitted gradient boosting model.

`/* Initialize model with constant value a */`

$f_0(x) = \arg\min_a \sum_{i=1}^n \mathscr{L}(y_i, a)$

`/* Iterate over all` M `learners in the model */`

for $m \leftarrow 1$ **to** M **do**

  Calculate residuals $r_{im} = -\left[\frac{\partial \mathscr{L}(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)}\right]$ for $i = 1, \dots, n$

  Fit regression tree to the residuals r_{im} using the typical procedure for regression trees. Refer to the leafs of the tree as R_{jm} for $j = 1, \dots, J_m$

  `/* Iterate over all leafs and compute a scaling factor` γ_{jm} `that minimizes the loss for each of them. */`

  for $j \leftarrow 1$ **to** J_m **do**

    Calculate $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} \mathscr{L}(y_i, f_{m-1}(x_i) + \gamma)$

  Update the model: $f_m(x) = f_{m-1}(x) + \eta\, \gamma_{j(x,m)\,m}$,where $j(x, m)$ describes the index of the leaf node that sample x ends up in.

$$j(x, m) = \arg\max_j I(x \in R_{jm})$$

return $f_M(x)$;

the previous model. Similar to logistic regression, gradient boosting for binary classification works by doing regression in the logit space. The output of the model is then not the predicted probability but the logit of it:

$$f(x_i) = \text{logit}(\hat{y}_i) \tag{5.21}$$

and to get $\hat{y}$, the output has to be mapped back into a probability by using the logistic function:

$$\hat{y} = \frac{1}{(1 + e^{-\text{logit}(\hat{y})})} = \frac{1}{(1 + e^{-f(x)})} \tag{5.22}$$

Since the individual trees operate in logit space, it is helpful to denote the binary cross entropy loss defined in Equation 5.5 as a function of $\text{logit}(\hat{y})$ instead of $\hat{y}$:

$$\mathscr{L}_{\text{BCE}}(y_i, f(x_i)) = -y\,\text{logit}(\hat{y}_i) + \ln(1 + e^{\text{logit}(\hat{y}_i)}) = -y f(x_i) + \ln(1 + e^{f(x_i)}) \tag{5.23}$$

The residuals based on the previous model with index $m - 1$ in the ensemble can now be easily computed using the derivative of Equation 5.23 with respect to $f(x_i)$ and:

$$r_{im} = y_i - \frac{e^{f_{m-1}(x_i)}}{1 + e^{f_{m-1}(x_i)}} \tag{5.24}$$

Once the residuals are obtained, the next tree in the ensemble is obtained by fitting it to the residuals r_{im} as discussed in subsection 5.1.2. Then, for all leafs in the new tree, the optimal value for the scaling value γ_{jm} is computed. For the loss function Equation 5.23, the optimal value for γ can be computed with a Taylor approximation:

$$\mathscr{L}(y_i, f_{m-1}(x_i) + \gamma) \approx \mathscr{L}(y_i, f_{m-1}(x_i)) + \frac{d}{df}(y_i, f_{m-1}(x_i))\gamma + \frac{1}{2}\frac{d^2}{df^2}(y_i, f_{m-1}(x_i))\gamma^2 \tag{5.25}$$

Using the derivative with respect to γ, one obtains:

$$\frac{d}{d\gamma}\mathscr{L}(y_i, f_{m-1}(x_i) + \gamma) \approx \frac{d}{df}(y_i, f_{m-1}(x_i)) + \frac{d^2}{df^2}(y_i, f_{m-1}(x_i))\gamma^2 = 0 \tag{5.26}$$

After some calculation, the solution for γ_{jm} is given by:

$$\gamma_{jm} = \frac{\sum_{x_i \in R_{jm}} r_{im}}{\sum_{x_i \in R_{jm}} e^{-f_{m-1}(x_i)} (1 + e^{-f_{m-1}(x_i)})^{-2}} \tag{5.27}$$

Finally, model at the current iteration, m, is updated:

$$f_m(x) = f_{m-1}(x) + \eta\, \gamma_{lm} \tag{5.28}$$

where l is the index of the leaf that x ends up in and η is the learning rate, which is set before the training. In summary, the fundamental gradient boosting algorithm works by iteratively training decision tree models that do a regression on the residuals of the previous models. By doing this, the loss function is continuously improved at each iteration.

XGBoost

XGBoost [93] is a popular gradient boosting algorithm, in particular for tabular data sets. It is based on the gradient boosting algorithm discussed in the previous chapter, but employs important improvements for better performance and faster training.

One difference of XGBoost compared to the default gradient boosting algorithm is that it employs specific regularization techniques. In particular, the loss function for XGBoost is given by:

$$\mathscr{L}_{\text{XGBoost}}(y_i, f_m(x_i)) = \mathscr{L}(y_i, f_m(x_i)) + \omega\, T + \frac{1}{2}\, \lambda\, \gamma^2_{j(x_i,m)m} \tag{5.29}$$

where again $j(x, m)$ describes the index of the leaf node that sample x ends up in and T is the total number of leafs in the current tree. In Equation 5.29, two regularization terms are added. ω is a term that controls how trees are "pruned", which describes a regularization technique where, after a tree is fitted, leaf nodes are deleted from the tree to reduce overfitting. Finally, the term λ penalizes the values of the individual leaf weights, γ_{jm}.

Another difference is that XGBoost works with any differentiable function for the loss. This is because instead of computing gradient descent in function space, XGBoost uses the Newton-Raphson method, which allows to compute a generic loss function based on a second order Taylor approximation. First, the gradients and hessians are calculated at each iteration m:

$$g_i = \frac{\partial \mathscr{L}(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \tag{5.30}$$

$$h_i = \frac{\partial^2 \mathscr{L}(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)^2} \tag{5.31}$$

The Taylor approximation can then be written as:

$$\mathscr{L} \approx \sum_{i=1}^{N} (g_i \cdot f_m(x_i) + \frac{1}{2} \cdot h_i f_m^2(x_i)) + \omega\, T + \frac{1}{2} \lambda \gamma^2_{j(x_i,m),m} \tag{5.32}$$

Since the individual functions $f_m(x)$ are tree-based learners with leaf weights $\gamma_{j(x,m),m}$, Equation 5.32 can be simplified to:

$$\mathscr{L} \approx \sum_{i=1}^{N} (g_i \cdot \gamma_{j(x_i,m),m} + \frac{1}{2} \cdot (h_i + \lambda) \gamma^2_{j(x_i,m)m}) + \omega\, T \tag{5.33}$$

Setting Equation 5.33 to zero and solving for the value of γ_{jm} yields:

$$\gamma_{jm} = -\frac{\sum_{x_i \in R_{jm}} g_i}{\sum_{x_i \in R_{jm}} h_i + \lambda} \tag{5.34}$$

Finally, the value of γ_{jm} can be substituted in the loss function:

$$\mathscr{L} \approx -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{x_i \in R_{jm}} g_i)^2}{\sum_{x_i \in R_{jm}} h_i + \lambda} + \omega T \tag{5.35}$$

Using the Newton-Raphson method outlined above allows to disconnect the XGBoost implementation from the desired loss function. As long as the function is differentiable, any function can be set as the cost function by the user.

Another important difference between XGBoost and classic gradient boosting is the way the trees are built. Instead of using the Gini impurity to choose the optimal split value, a score is computed that is based on the loss outlined in Equation 5.35:

$$I_{\text{XGBoost},j} = \frac{1}{2} \frac{(\sum_{x_i \in R_{jm}} g_i)^2}{\sum_{x_i \in R_{jm}} h_i + \lambda} \tag{5.36}$$

When splitting samples, $I_{\text{XGBoost},j}$ is computed for the root node, as well as the leafs for samples below and above the current candidate threshold. Then, the gain G is computed to decide which feature and split value is optimal:

$$G = I_{\text{XGBoost,low}} + I_{\text{XGBoost,high}} - I_{\text{XGBoost,root}} \tag{5.37}$$

Using a loss-based score for splitting the samples has the major advantage that the optimization of the gradient is done at each split and not only at the end of the tree-building process as in classical gradient boosting.

Additionally, several performance optimizations are employed in XGBoost that significantly de-

crease training time, in particular for larger data sets. Classic gradient boosting builds trees as discussed previously in subsection 5.1.2: When splitting the sample, each possible threshold value for each feature in the data set is tested as a split candidate. This "greedy" algorithm is computationally expensive, in particular when the training set consists of many samples. Therefore, XGBoost implements an approximate greedy algorithm that is based on weighted quantiles instead of unique values for threshold candidates. First, using sketching algorithms and parallel computing, the entire data set is split up across processors, histogrammed and approximate quantiles are computed. The used weights are based on the hessians, h_i that were discussed previously. The quantiles are computed such that their sum of weights is equal and the quantile values are then defined as possible threshold candidates. In total, approximately 33 of them are typically used in XGBoost per feature, leading to a significant speed-up compared to the classic greedy algorithm.

Furthermore, XGBoost implements improved handling of missing or frequent zero values. It defines a default branch where all missing values go by using the optimal gain for both branch options at each split. This groups samples with missing values together and learns to deal with them in an optimal way. Finally, XGBoost uses several hardware optimizations to be more computationally efficient compared to classical algorithms. One feature is cache-aware access, where gradients and hessians are stored in the CPU cache for fast computation of splitting scores and output values. Another important feature is that XGBoost uses blocks for out-of-core computation: If the data set is too large to fit the cache or RAM of the machine, data has to be read from or written to the hard drive memory, which is slow. XGBoost uses block compression to save disk space and time writing the data to disk. When the data is accessed, it is uncrompressed on the fly by the CPU. When multiple disks are available, XGBoost uses sharding to distribute data across multiple disks, such that they can be read in parallel and the throughput is increased.

LightGBM

LightGBM [94] is an improved gradient boosting algorithm, which was implemented after the emergence of XGBoost. Therefore, it contains most of the improvements mentioned in the previous sections and further optimizes gradient boosting in terms of performance and computational costs. One key difference between XGBoost and LightGBM is the way that trees are constructed: XGBoost and the classical gradient boosting algorithms use *level-wise* tree growth, whereas LightGBM uses *leaf-wise* tree growth. In level-wise tree growth, each leaf node at the current level of the tree is split, leading to a balanced tree with a similar amount of branches left and right of the root node. In leaf-wise tree growth, however, the leaf that reduces the loss most is identified and only this leaf is split. This means that it is possible that only leafs in a certain branch can be split further, while other leafs on higher levels in the trees are ignored. This leads to an asymmetrical tree, where one side of the tree contains more branches then the other. Leaf-wise tree growth leads to lower achieved loss values but also to more complex trees, which increases overfitting. Therefore, leaf-wise tree growth is especially suited when dealing with large data sets, whereas smaller data sets are quickly overfitted.

Similar to XGBoost, LightGBM reduces the amount of candidate thresholds to consider when splitting. However, it does so using histogrammed gradient boosting techniques. This means that the features are histogrammed in a predefined number of bins and each bin edge is used as a candidate threshold. This significantly reduces the amount of thresholds to be scanned and is also a more simple

and resource-efficient implementation compared to the weighted quantile sketch used in XGBoost. LightGBM also speeds up the training by using a technique called gradient-based one-side sampling (GOSS). Using only a sub-sample of the entire training data to fit a tree is a common technique to introduce variance in an ensemble. At the same time, this can be thought of as a form of regularization, since the model cannot overfit on the entire training data any more. LightGBM uses an efficient downsampling technique, which is based on the fact that the contribution of samples depends on their gradients. Therefore, samples that contribute most due to having a high gradient should be kept such that the tree building process is a s efficient as possible. However, simply discarding all samples with low gradient values would lead to a distortion of the actual feature distributions, which will reduce the model performance. Therefore, LightGBM implements the above mentioned GOSS algorithm, which allows subsampling the samples with the highest gradients, while at the same time making sure the original distributions are not being distorted. It does so by first selecting the samples with large gradients and then performing random sampling on those with small gradients. Finally, when computing the gain to split the samples at an internal node, a weight w is applied to the low-gradient samples:

$$w = \frac{1 - \epsilon_{\text{high}}}{\epsilon_{\text{low}}} \tag{5.38}$$

where ϵ_{high} and ϵ_{low} are the fractions of selected high gradient and subsampled low gradient data points with respect to the full data, respectively. This factor ensures that the original distribution is retained while at the same time being able to focus more on samples with high gradient values. Lastly, LightGBM uses a feature called Exclusive Feature Bundling (EFB) to deal with sparsity in the data. This relates to the fact that in many real-world applications features are sparse, such as one-hot encoded categorical features. One characteristic of such features is that they are often exclusive in a sense that they rarely take on non-zero values simultaneously. Therefore, these features can be bundled effectively, which is done using the complex graph-based EFB algorithm in LightGBM.

5.3 Deep Learning

While the classical machine learning algorithms described in section 5.2 have been shown to achieve state-of-the-art performance on tabular data tasks, they lack the ability to model complex, high-dimensional nonlinear problems. In particular, machine learning tasks like Natural Language Processing (NLP) or computer vision are difficult to solve for these models. One class of machine learning algorithms that is specifically suited for such tasks is deep learning. These algorithms use highly complex nonlinear functions with a large number – often millions – of parameters to model the problem at hand. According to the universal approximation theorem [96], this allows for an approximation of any arbitrarily complex function, given an appropriate neural network architecture. Due to the advent of novel algorithms as well as dedicated, highly parallel computing infrastructure such as Graphics Processing Units (GPUs), deep learning has received significant interest in research and application in the last decade. In the subsequent sections, the fundamental principles of this field will be discussed.

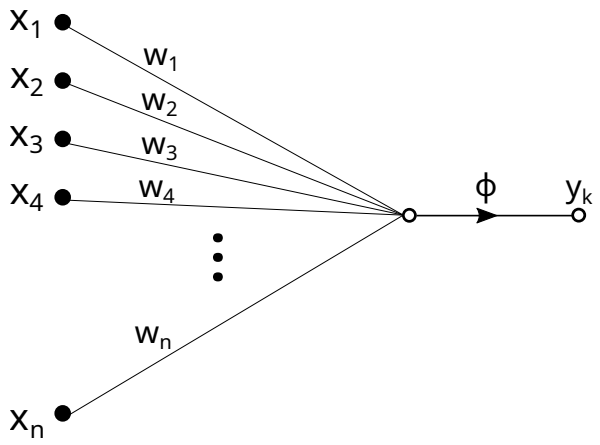


Figure 5.1: Diagram of an artificial neuron.

5.3.1 Artificial Neurons

Deep learning algorithms are based on layers of units that are called artificial neurons [97] or nodes. An algorithm that uses multiple of such layers that are interconnected is also referred to as Deep Neural Network (DNN) [98]. The artificial neurons are the foundational building blocks for any kind of DNN and they define a function $f\colon \mathbb{R}^n \to \mathbb{R}$, which is given by:

$$f(\mathbf{x}) = \phi\left(\sum_{i=1}^{N} w_i x_i + b\right) \tag{5.39}$$

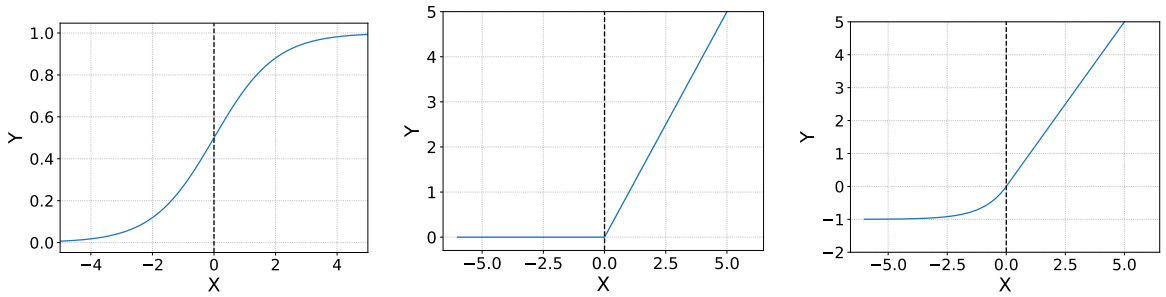
From Equation 5.39 it can be seen that the structure of an artificial neuron is rather simple: There are N inputs, denoted as $(x_1, \dots, x_n)$ and each input x_i is assigned a weight w_i that it gets multiplied with. A bias term, b, is also included, which acts as an additional parameter. A function ϕ, also referred to as the "activation function", is then applied to the sum of weighted inputs plus the bias term, which yields the output of the neuron, $f(\mathbf{x})$. A diagram of an artificial neuron is shown in Figure 5.1.

If ϕ were a linear function, artificial neurons would only be able to model linear problems. Therefore, nonlinear functions are typically used as activation functions. Several activation functions exist and the particular choice depends on a variety of different factors, since each of them comes with advantages and disadvantages. Neural networks – similar to gradient boosting – are trained using the gradients of a loss function with respect to the weights of their nodes. For simplicity, consider the optimization of a weight in a neural network consisting only of the neuron described in Equation 5.39. Since weights are typically optimized in an iterative manner, the formula to update a weight at time step $t-1$ to the value at the current step t is given by:

$$w(t) = w(t-1) - \eta \frac{\partial \mathscr{L}(f(\mathbf{x}), y)}{\partial w(t-1)}, \tag{5.40}$$

where η is a pre-defined learning rate, $\mathscr{L}$ is the loss function to be optimized and y is the truth label that the neuron should predict. As the optimization depends on the gradient of $\mathscr{L}$, which itself depends on the node output $f(\mathbf{x})$, the activation function ϕ benefits from being continuously differentiable. However, activations exist that do not exhibit this property for some points or ranges within their domain. A selection of frequently used activation functions is shown in Figure 5.2.

The sigmoid function is defined between zero and one and therefore keeps the values within a



(a) Sigmoid (b) ReLU (c) ELU

Figure 5.2: Different commonly used activation functions.

well-defined range. It allows to turn the output of a model into predictions of class probabilities, which is used in classification tasks. It is also a continuously differentiable function, which allows to compute the gradient at any point. One problem with the sigmoid function is that the gradient gets small for large positive and negative values. Once this happens, the values of the gradients in Equation 5.40 are too small to cause an actual update of the weight and the training stalls out. This phenomenon is also referred to as the vanishing gradient problem. One way to mitigate this problem is to use an activation that has a constant gradient value throughout most of its domain.

Such a function is the rectified linear unit (ReLU), which is given by:

$$f(x) = \max(0, x)\,. \tag{5.41}$$

This function has a non-vanishing gradient for large values, while it is zero for negative values. This helps the network to learn for a large range of node input values, while also being able to shut down a node, allowing the network to learn sparse representations and making the training more efficient. The ReLU activation is not continuously differentiable, since the derivative cannot be computed for $x = 0$. However, this is not relevant in practice, since the gradient can simply be set to zero at this point. One problem that can occur is a phenomenon called "dying ReLU": When a node with a ReLU activation function gets in a state where it is never getting activated since its output value (and therefore its gradient) is always zero. If many nodes in a network get pushed into such a dead state, the learning capacity is reduced significantly, since their gradient-based information is no longer available. To remedy the occurrence of dead nodes, variations of ReLU exist that also have non-zero gradients for negative values. One such variation is the Exponential Linear Unit (ELU) function, which is given by:

$$f(x) = \begin{cases} x & \text{if } x \geq 0\,, \\ \alpha \cdot (e^x - 1) & \text{if } x < 0\,. \end{cases} \tag{5.42}$$

5.3.2 Deep Neural Networks

Artificial neurons are the foundational building blocks of deep learning algorithms. In DNNs, many interconnected layers of neurons are typically used to create a model that is able to extract intricate nonlinear patterns from the input data [98]. Due to this ability, DNNs have shown to achieve state-of-the-art results in complex tasks. In NLP, a DNN architecture named transformer [99] has been shown to yield the best results in translation [84, 100] and also language generation tasks [101], which

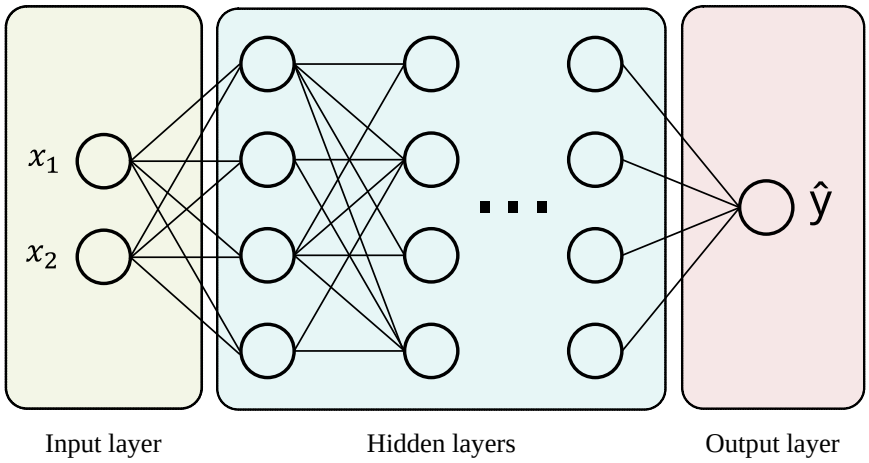


Figure 5.3: A diagram of a generic deep neural network (DNN).

culminated in the creation of the generative pre-trained transformer (GPT) models and their use in popular applications such as ChatGPT [102]. But also for other tasks such as computer vision [103] or speech recognition [104], dedicated DNN architectures exist that show state-of-the-art performance. In particle physics, deep learning – based approaches have successfully been used to improve on various tasks, from jet tagging to tracking to fast event simulation and beyond [105].

The most fundamental architecture of a DNN is referred to as a fully connected network and it consists of three kinds of layers: An input layer, one or more intermediate hidden layers and finally an output layer. A diagram of such a network can be seen in Figure 5.3. In this example, two input features x_1 and x_2 are used and the "fully connected" approach means that each node in a layer is connected to each node in the next layer. Also, a single output $\hat{y}$ is produced by the depicted DNN, which can be the case in a regression task, where a continuous variable is predicted or for a binary classification task, where the output is the predicted probability that the current sample belongs to the "positive" class with index 1. However, the output layer might have any number of nodes, depending on the task at hand. Mathematically, one layer in a DNN can be expressed as a collection of N nodes. Since it allows for a more compact description, the vector/matrix notation will be used in the following, where boldfaced lowercase letters correspond to vectors and boldfaced uppercase letters to matrices. In this notation, a signle DNN layer can be denoted as:

$$\mathbf{o} = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{5.43}$$

where $\mathbf{o}$ is the N-dimensional output vector of the layer containing N neurons, $\mathbf{x}$ is the input vector to the layer, $\mathbf{W}$ is a $N \times D$ matrix where D is the dimension of the input vector and $\mathbf{b}$ is the bias vector.

In a fully connected DNN, the inputs of the hidden layers and the output layer are the outputs of the previous layers. Therefore, the output of an entire network, which is the prediction $\hat{\mathbf{y}}$ for the input $\mathbf{x}$, can be written as a combination of the output functions of L layers:

$$\begin{aligned}
\hat{\mathbf{y}} &= \phi^L(\mathbf{W}^L\mathbf{o}^{L-1} + \mathbf{b}^L) \\
&= \phi^L(\mathbf{W}^L\phi^{L-1}(\mathbf{W}^{L-1}\mathbf{o}^{L-2} + \mathbf{b}^{L-1}) + \mathbf{b}^L) \\
&= \phi^L(\mathbf{W}^L\phi^{L-1}(\mathbf{W}^{L-1}\phi^{L-2}(\mathbf{W}^{L-2} \ldots \phi^1(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1) \ldots + \mathbf{b}^{L-2}) + \mathbf{b}^{L-1}) + \mathbf{b}^L).
\end{aligned} \tag{5.44}$$

This equation intuitively shows a key characteristic of neural networks: While the fundamental building blocks, the artificial neurons, are relatively simple entities, the combination of neurons into layers and multiple layers into large networks give rise to a highly complex construct that is capable of extracting intricate patterns from data. Due to their complex structure, neural networks also often contain a large number of parameters that need to be optimized for a given problem, which is a non-trivial task that requires dedicated algorithms for accurate and efficient computation. Thus, the foundations of selected state-of-the-art methods for training and optimizing deep learning algorithms are discussed in the following chapter.

5.3.3 Training of Deep Learning Algorithms

Gradient Descent and Optimization

Similar to classical machine learning methods, DNNs learn complex functions by minimization of a loss term. During training, the weights and biases of all layers are changed to minimize the loss using a gradient descent-based approach. Each weight w in the network is updated as:

$$w(t) = w(t-1) - \eta \frac{\partial \mathscr{L}(\hat{\mathbf{y}}, \mathbf{y})}{\partial w(t-1)}, \tag{5.45}$$

where η is again a pre-defined learning rate, $\mathscr{L}$ is the loss function, $\hat{\mathbf{y}}$ is the model output and $\mathbf{y}$ is the truth value to be predicted. t defines the index of the optimization step.

Gradient descent is based on the fact that, given the loss function and the currently assumed values of the weights and biases, the fastest decrease of the loss is in the direction of the negative gradient with respect to these weights. Therefore, using Equation 5.45 iteratively means to take steps towards lower loss values until a minimum is finally reached. In order to make sure that this can be achieved, the learning rate has to be carefully chosen: If the value is set too low, the training will stagnate and the minimum will not be reached. If it is chosen too high, the weight updates will be too large and the algorithm will step over or out of the valid minimum.

Another challenge in classical gradient descent is that for the gradient computation, the entirety of input-output-pairs present in the data set is considered. Therefore Equation 5.45 could also be written explicitly as:

$$w(t) = w(t-1) - \frac{\eta}{n} \sum_{i=1}^{n} \frac{\partial \mathscr{L}(\hat{\mathbf{y}}(\mathbf{x}_i), \mathbf{y}_i)}{\partial w(t-1)}. \tag{5.46}$$

Evaluating the gradient term for all datapoints can be computationally expensive. Additionally, in particular for large data sets, fitting the entire data at once is unfeasible given the memory limitations of the hardware. Therefore, stochastic optimisation methods were developed, which compute the weight updates based on a sub-sample of the data. In Stochastic Gradient Descent (SGD) [106], the data set is first randomly shuffled and then a weight update is done for each individual datapoint. Naturally, using a single point results in a poor estimate of the gradient, which introduces a lot of noise to the optimization process. Thus, SGD converges only slowly, since the variance of the achieved loss value for each update is large. In order to achieve a better trade-off between computational complexity and gradient estimation, mini-batch gradient descent was introduced, which uses small sub-samples, so-called "batches" of data instead of individual data points for each weight update. Not only does this enhance the accuracy of the gradient estimate, but it also enables the application of vectorization

methods for efficient calculations on highly parallel processing devices like GPUs.

Another caveat in classical gradient descent is that the individual optimization steps only take into account the gradient at the current position in loss space. However, this space often contains shallow local minima, saddle points or local curvatures that lead to suboptimal steps during minimization. This behaviour can result in slow convergence or in the algorithm ending up in an incorrect local minimum. Therefore, an approach using a term called "momentum" was developed to mitigate the mentioned effects. The main idea behind momentum is to not only take into account the local gradient, but also the general trend of the loss space based on the gradients from previous updates. For a weight update using momentum, a new change term $\beta(t)$, where t is the index of the current optimization step, is introduced:

$$\beta(t) = \frac{\partial \mathscr{L}}{\partial w(t)} + \mu\, \beta(t-1)\,, \tag{5.47}$$

where μ is the momentum parameter and w is the weight to be updated. It can be seen that the first part of the sum is simply the gradient of the loss with respect to the weight. The second term of the sum introduces the momentum μ and defines a recursion depending on the $\beta(t-1)$ value from the previous iteration. This adds "short term memory" based on the optimization history to the equation and μ, which is a real-valued parameter that assumes values between zero and one, determines the weight of this history in the update. The update of the weight $w(t)$ using momentum is then given by:

$$w(t) = w(t-1) - \eta\, \beta(t)\,, \tag{5.48}$$

where η is again the learning rate. Plugging in the definition of $\beta(t)$, one can see that for a momentum value $\mu = 0$, the original gradient descent formula (Equation 5.45) is recovered. For higher values of μ, however, the historic optimization term is increased, leading to fewer oscillations caused by local curvatures and additional inertia to overcome shallow minima.

Another extension of gradient descent-based optimization is Root Mean Squared Propagation (RMSProp). The fundamental idea is, instead of keeping the learning rate η fixed, to compute an adaptive learning rate for each parameter. This is achieved by normalizing the intial learning rate by the square root of the sum of squared gradients:

$$\tilde{\eta}(t) = \frac{\eta}{\sqrt{V(t-1)}}, \text{ where } V(t) = \sum_{\tau=1}^{t} g(\tau)^2 = \sum_{\tau=1}^{t} \left(\frac{\partial \mathscr{L}}{\partial w(\tau)} \right)^2 . \tag{5.49}$$

This equation describes the AdaGrad approach [107]. While this adaptation of the learning rate can improve the convergence, it also often yields too small learning rates, leading to a stagnation of the optimization procedure. RMSProp mitigates this effect by using a decaying moving average of the past and current gradients instead of the squared sum:

$$G(t) = \rho\, G(t-1) + (1-\rho) \left(\frac{\partial \mathscr{L}}{\partial w(t)} \right)^2 , \tag{5.50}$$

where ρ is another hyperparameter. ρ can be understood as a momentum parameter in this equation, assigning a higher weight to past gradient values at the expense of a lower weight for current values.

The adapted learning rate is then given by:

$$\tilde{\eta}(t) = \frac{\eta}{\sqrt{G(t-1)}} \tag{5.51}$$

and the weight update is computed as usual by simply replacing the fixed learning rate with the adapted one.

$$w(t) = w(t-1) - \tilde{\eta}(t)\,\frac{\partial \mathscr{L}}{\partial w(t-1)}\,. \tag{5.52}$$

Both AdaGrad and RMSProp lead to significant improvements for many optimization tasks. Since AdaGrad decreases the learning rate over time based on the sum of squares of past gradients, parameters that receive infrequent updates are assigned a higher learning rate compared to those with more frequent updates. This makes AdaGrad perform well on problems with sparse data, such as natural language processing or computer vision. RMSProp on the other hand adapts the learning rate based on the average of *recent* gradient values, which leads to an efficient scaling for quickly changing gradients. Therefore, RMSProp is recommended for problems which are noisy or non-stationary (i.e. when optimal parameters change over time).

To combine the strength of both approaches, another optimization algorithm, named Adam, was developed [108]. It uses both the average first moment (the mean) as in RMSProp and the non-centered second moment of the gradients as in AdaGrad to adjust the learning rate. In particular, Adam computes an exponential moving average for both the gradients and squared gradients and the respective parameters β_1 and β_2 control their decay rate. At first, two new variables, $m(t)$ and $v(t)$ are introduced for the moving averages:

$$m(t) = \beta_1\, m(t-1) + (1-\beta_1)\,\frac{\partial \mathscr{L}}{\partial w(t-1)}\,, \tag{5.53}$$

$$v(t) = \beta_2\, v(t-1) + (1-\beta_2)\left(\frac{\partial \mathscr{L}}{\partial w(t-1)}\right)^2\,. \tag{5.54}$$

Because the initial values, $m(0)$ and $v(0)$ are set to zero, there is an initial bias in the update of the weights that causes the large update steps in the first iterations. To mitigate this a bias correction is applied:

$$\hat{m}(t) = \frac{m(t)}{1-\beta_1^t}\,, \tag{5.55}$$

$$\hat{v}(t) = \frac{v(t)}{1-\beta_2^t}\,. \tag{5.56}$$

Finally, the weight update at iteration t is computed as:

$$w(t) = w(t-1) - \eta\left(\frac{\hat{m}(t)}{\sqrt{\hat{v}(t)}+\epsilon}\right), \tag{5.57}$$

where η is the learning rate and ϵ is a small real number that gets added to the denominator such that the value at the initial step is not infinite.

Adam is implemented in many modern machine learning frameworks and the default values for the respective hyperparameters are very similar. Table 5.1 shows the respective default values in the `PyTorch` [109] framework. Since Adam has been shown to achieve accurate results fast for many

Parameter	Default value
η	10^{-3}
β_1	0.9
β_2	0.999
ϵ	10^{-8}

Table 5.1: Default parameter values of the Adam optimizer, as implemented in the `PyTorch` framework [109].

problems, in particular in deep learning applications, it is a widely used optimizer. Also in this work, it will be used throughout many of the studies that are discussed.

Layer-Wise Backpropagation

As discussed previously, deep neural networks contain a vast amount – often millions – of trainable parameters. Updating each weight as in Equation 5.45 efficiently for a highly complex neural network, as defined in Equation 5.44 is computationally challenging. Therefore, modern machine learning frameworks use a method called layer-wise backpropagation [110, 111] to compute the gradients. In the following description of this algorithm, some approaches are employed to simplify the notation: As in the beginning of this section, the vector/matrix notation is used. Additionally, the bias parameters of the network are not considered, as they simply constitute another set of parameters that are updated in a very similar way as the weight parameters.

The general idea behind backpropagation is that computing the gradients in a neural network can be simplified by acknowledging that each weight in the weight matrix $\mathbf{W}^l$ of layer l can affect the loss only in the following layers and only in a linear way. Consider the gradient of the loss function with respect to the weights $\mathbf{W}^l$:

$$\frac{\partial \mathscr{L}}{\partial \mathbf{W}^l} = \frac{\partial \mathscr{L}}{\partial \mathbf{z}^l} \frac{\partial \mathbf{z}^l}{\partial \mathbf{W}^l} , \tag{5.58}$$

where $\mathbf{z}$ describes the weighted inputs to layer l:

$$\mathbf{z}^l = \mathbf{W}^l \mathbf{o}^{l-1} . \tag{5.59}$$

To further simplify the algebra, the error term $\boldsymbol{\delta}^l$ is introduced:

$$\boldsymbol{\delta}^l = \frac{\partial \mathscr{L}}{\partial \mathbf{z}^l} . \tag{5.60}$$

Finally, plugging in Equation 5.59 and Equation 5.60 into Equation 5.58 yields:

$$\frac{\partial \mathscr{L}}{\partial \mathbf{W}^l} = \delta^l \mathbf{o}^{l-1} . \tag{5.61}$$

The $\mathbf{o}^{l-1}$ is the output of the previous layer, which can simply be cached when the data is passed through each layer in the network, which is also referred to as “forward pass”. Next, an expression

for δ^l needs to be derived which is only depending on the subsequent layers.

$$\delta^l = \frac{\partial \mathscr{L}}{\partial \mathbf{z}^l} = \frac{\partial \mathscr{L}}{\partial \mathbf{z}^{l+1}} \frac{\partial \mathbf{z}^{l+1}}{\partial \mathbf{z}^l} = \delta^{l+1} \mathbf{W}^{l+1} (\boldsymbol{\phi}^l)'(\mathbf{z}^l) \tag{5.62}$$

Now, once the error for the final layer, L, is obtained, the errors of all previous layers can be computed using Equation 5.62. This is also the reason for the name of the backpropagation method, since it is possible to efficiently compute the gradient with respect to the weights by starting with the error term on the final layer and then iterate back over all layers in the network up to the input layer. The needed term for the error in the final layer can be computed from the respective loss:

$$\mathscr{L}(\mathbf{y}, \hat{\mathbf{y}}) = \mathscr{L}(\mathbf{y}, \mathbf{o}^L) \,, \tag{5.63}$$

and the error is then given by:

$$\delta^L = \frac{\partial \mathscr{L}}{\partial \mathbf{z}^L} = \frac{\partial \mathscr{L}}{\partial \mathbf{o}^L} \frac{\partial \mathbf{o}^L}{\partial \mathbf{z}^L} = \frac{\partial \mathscr{L}}{\partial \mathbf{o}^L} (\boldsymbol{\phi}^L)'(\mathbf{z}^L) \,. \tag{5.64}$$

These are all the ingredients needed to efficiently compute the gradient of the loss with respect to the weights for any layer l. As previously discussed, vector and matrix notations have been used in the description of the layer-wise backpropagation method for simplicity. However, it naturally extends to individual indices for each weight. For example, the weight in layer l of the neural network that multiplies the output node k of the previous layer and connects to the node j of the current layer can be written analogously to Equation 5.61:

$$\frac{\partial \mathscr{L}}{\partial w^l_{jk}} = \delta^l_j o^{l-1}_k \,. \tag{5.65}$$

In the above derivation of the backpropagation algorithm, a single input-output-pair $(\mathbf{x}, \mathbf{y})$ was assumed. However, mini-batch gradient descent is often used in practice, as discussed in the previous subsection. Therefore, the weight update is done using the average gradient of all samples within the mini-batch:

$$w(t) = w(t-1) - \frac{\eta}{n} \sum_{i=1}^{n} \frac{\partial \mathscr{L}(\hat{y}(x_i), y_i)}{\partial w(t-1)} \,. \tag{5.66}$$

There are two main reasons why using batched gradient descent is favoured over computing the weight updates for the entire data set: Firstly, there is evidence that using very large batch sizes, or even the entire data set, at once might lead to convergence towards sharp minima in the loss space that show worse generalization to unseen data [112, 113]. Secondly, using batches has several advantages in terms of computational cost. Large data sets often do not fit into the memory of a GPU entirely. Using smaller batches of samples reduces the required memory significantly and allows for training on data sets of almost any size. Additionally, many modern machine learning frameworks make use of vectorization such that the gradients for the samples in a batch can be computed in parallel, which considerably speeds up the training. In practice, the number of samples per batch – the batch size – is chosen before the training depending on various parameters, such as the available hardware, memory and data set size. The data set is then split into batches of the chosen size and for each batch, the weights are updated. Once all batches have been used, which is referred to as epoch, the batches are typically reshuffled and further epochs are trained until either a pre-defined limit is reached or the loss

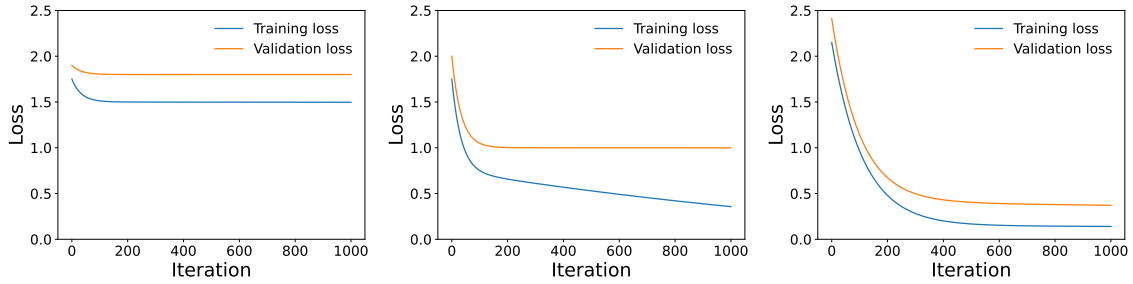


Figure 5.4: Loss curves for the cases of underfitting (left panel), overfitting (center panel) and a good fit (right panel).

does not improve within a certain number of epochs.

5.3.4 Model Selection and Regularization

As discussed previously, the optimization of a machine learning model is typically done in an iterative way. At each step, the parameters of the model are adjusted based on the gradient to further minimize the loss. A key question in machine learning tasks is at which iteration to stop the training and select the current model state for further analysis. This problem is often dealt with by investigating the learning curves, also referred to as loss curves, which show the value of the loss as a function of the iterations. One factor to consider in the training and evaluation of machine learning models is that they are trained on a specific set of data points, also referred to as the training set. Naturally, evaluating and selecting the model solely based on the performance on this set leads to a significant bias, which is mitigated by using a statistically independent sample of the data for validation. Thus, this sample is referred to as the validation set and it allows to assess how well the performance achieved on the training set generalizes to data that is yet unknown to the model. The loss curves of both the training and the validation set can then be studied to optimize model selection to obtain the model with the best generalization performance. Typically, variations of three main regimes can be distinguished from these curves: the overfitting and underfitting regimes as well as the regime of a good fit. Conceptual versions of these curves can be seen in Figure 5.4.

In the underfitting regime, both the validation and training loss only reduce marginally within the first few iterations and then stagnate. Additionally, the loss values are high in comparison with the other curves in the figure. This shows that the model is unable to learn the task, which is often the case when the complexity of the model is not sufficient to capture the function underlying the problem. In the overfitting regime on the other hand, the training loss decreases continuously, whereas the validation loss only decreases in the first few iterations and then stagnates, leading to an increasing performance gap between the two data sets. In this case, the model does not learn information meaningful to the task. Instead, it memorizes the noise in the training data set, which leads to the poor generalization performance as shown in the stagnating validation loss. This regime is often observed for models that have a high complexity, such as DNNs. Finally, a good fit is achieved when both the training and the validation losses decrease to a valid minimum value and the difference in performance between the two data sets is small.

There are different ways to mitigate underfitting and overfitting when training machine learning models. Underfitting can often be remedied by increasing the complexity of the model or switching to

a more complex algorithm. For overfitting on the other hand, the complexity of the algorithm should be reduced. This can be achieved by reducing the amount of parameters of the model. For a DNN, the number of nodes per layer or the overall number of layers in the network can be decreased. To reduce overfitting in tree-based methods, leaves can be pruned or the maximum tree depth can be reduced. Another method to mitigate overfitting is to penalize large parameter values in the model using regularization. Consider the example of a layer in a DNN, where several of the trained weights have large values. Small changes to the inputs will have a large impact on the resulting output and thus the network will likely show poor performance when used on new datapoints. Therefore, regularization methods exist, which are designed to keep the weight values small and thus reduce overfitting.

One of these methods is to regularize the training based on the magnitude of the weight values, which can be done by taking into account the absolute sum of the weights (L1 regularization) [114] or the sum of squared values of the weights (L2 regularization) [114–117]. In practice, an additional penalty term is added to the original loss function. For a L1 penalty, the regularized version of the loss is given by:

$$\mathscr{L}_{\mathrm{L1}} = \mathscr{L} + \lambda_1 \sum_{i=1}^{W} |w_i| \tag{5.67}$$

and accordingly for a L2 penalty:

$$\mathscr{L}_{\mathrm{L2}} = \mathscr{L} + \frac{\lambda_2}{2} \sum_{i=1}^{W} w_i^2 \,, \tag{5.68}$$

where W is the total number of weights of the algorithm and the λ_i are penalty terms that are defined before the training. The penalizing nature of the regularization can be seen when considering the weight updates:

$$w_i(t) = w_i(t-1) - \eta \frac{\partial \mathscr{L}_{\mathrm{L1}}}{\partial w_i(t-1)} = w_i(t-1) - \eta \frac{\partial \mathscr{L}}{\partial w_i(t-1)} - \eta \lambda_1 \operatorname{sign}(w_i(t-1)) \tag{5.69}$$

and respectively for the L2 regularization:

$$w_i(t) = w_i(t-1) - \eta \frac{\partial \mathscr{L}_{\mathrm{L1}}}{\partial w_i(t-1)} = w_i(t-1) - \eta \frac{\partial \mathscr{L}}{\partial w_i(t-1)} - \eta \lambda_2 w_i(t-1) \,. \tag{5.70}$$

Both equations show that the penalizing term works in the opposite direction of the weights magnitude: If the previous weight had a large positive value, the penalty will decrease the weight. If the previous weight was negative, the penalty will increase it. L1 and L2 regularization can be used separately or in combination. L1 regularization encourages sparsity in the model, pushing weights towards a value of zero, while L2 regularization assigns a higher penalty to large values due to the quadratic term. L2 regularization also reduces large weights, but does not usually push them to zero.

As with L1 regularization, introducing sparsity to an algorithm reduces overfitting. This is because a sparse model has fewer parameters and therefore corresponds to a less complex model, which makes it more difficult to memorize the noise in the training data set and improves generalization. For DNNs, another regularization technique exists that introduces sparsity to reduce overfitting which is called dropout [118, 119]. Dropout randomly sets outputs of a given neural network layer to zero during each training update. A hyperparameter called the dropout rate is introduced that defines the probability that a node will be dropped out. On the one hand, this reduces the model complexity,

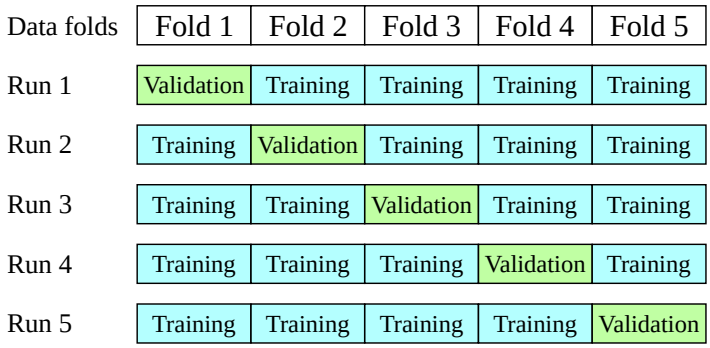


Figure 5.5: Schematic of k-fold cross-validation, using a value of $k = 5$.

since the effective number of weights per training iteration is reduced. On the other hand, it adds stochasticity to the training. Since the nodes to be dropped out are chosen randomly, the actual model architecture changes at each iteration. This has the effect that a model trained using dropout can be thought of as a single-model-representation of an ensemble of various neural networks. As discussed in section 5.2, training an ensemble of individually overfitting classifier models – such as a random forest – can significantly reduce the overfitting that is present in individual learners. For large models such as DNNs however, this is computationally expensive. Therefore, dropout offers a computationally effective method to train an approximation of a DNN ensemble using a single model.

Regularization and model selection can be viewed as two sides of the same coin, since the learning curves typically switch regimes during training. In the first iterations, both validation and training loss decrease, showing the desired behaviour of the "good fit" regime. At some point, the model stops to learn information regarding the task at hand and starts to focus on optimizing the individual data points of the training set. At this point, the validation loss stalls out or increases, which describes the transition to an overfitting regime. An effective method to reduce computation and make sure the selected model state is still in the good fit regime is to detect this transition point. This is done using a technique called early stopping. In early stopping, the validation loss is tracked at each iteration during training and if it does not decrease by a given amount within a pre-defined number of iterations, the training is stopped. Finally, the model state at the iteration where the minimum validation loss was achieved is selected for further analysis.

Another aspect of model selection is how to split the data set into training and validation sets to get a realistic estimate of the generalization performance of the model. Two main options exist for this cross-validation scheme: using a fixed training/validation split or using k-folding. For the former, fixed percentages are chosen and the training and validation data sets are randomly split according to these values. Commonly used values are approximately 80 % for the training and 20 % for the validation data set. k-folding on the other hand randomly splits the data into a number of k different sub-sets and then reruns the entire training k times, where each fold is assigned the validation set exactly once while the remaining $k - 1$ folds are used for training. An illustration of the k-folding is shown in Figure 5.5. Using k-folding is computationally more expensive, since k trainings are needed for the cross-validation scheme. However, it is possible to compute the generalization performance and other parameters of interest based on the average of the validation folds of each run, which yields a better estimate of the values and their variance.

So far, model selection was only discussed regarding the training set and the validation set. By

choosing the model with the lowest validation loss iteration, a bias is introduced, since this choice is based on the optimal generalization performance on the particular validation data set. Therefore, the subsequent performance evaluation has to be done on yet another statistically independent data set. This data set is referred to as "test data set" and it can also simply be split off the full data. The final metrics to compare the performance of different methods should typically be computed based on the test set as it represents a data set that is independent of both the training and model selection procedures. Similar to the training and validation sets, the test set can either be split off based on a percentage of a fixed training/validation/test-split or embedded in a $k - l$-folding scheme, for example by first k-folding for the test set and then introducing an additional training/validation l folding level for the remaining datapoints.

5.4 Normalizing Flows

In the previous section, deep neural networks (DNNs) were introduced using the example of a fully connected neural network, where the nodes of each layer are connected to all nodes of the subsequent layer. As mentioned before, many other neural network architectures exist that are tailored to solve a specific problem, such as convolutional neural networks for computer vision problems. Another task where deep learning methods excel at is generative modeling [120, 121]. These models have received increased research interest recently, as they have been shown to be able to generate high-quality artificial images [122], video [123, 124] or text [125] but also complex particle physics structures such as jet images [126] or point clouds of calorimeter hits [127].

From a machine learning perspective, two main approaches exist: the discrimiative approach and the generative approach. The difference between the two families of methods is what probability they try to approximate. Discriminative models try to learn the probability of a target variable Y, given a datapoint $x \in X$, which is the conditional probability $P(Y|X)$. Generative models on the other hand aim to approximate the full joint distribution $P(X, Y)$ and are often used to then generate new samples from the learned distribution that are similar to the input data.

Several generative methods exist to achieve this: Generative Adversarial Networks (GANs) consist of two neural networks, a generator and a discriminator that are trained at the same time. The objective of the generator is to generate datapoints that are similar to the input data samples, while the discriminator is trained to distinguish actual samples from generated ones. Using this adversarial training scheme, GANs can effectively generate highly accurate artificial data samples. The second class of generative models is autoencoder-based methods. Autoencoders encode the input data using an "information bottleneck" to map the input to a lower-dimensional space, which is also referred to as "latent space". A decoder then maps back from the latent space to the output space, which is trained to resemble the original input as closely as possible, typically by minimizing the difference between input and output. Another family of methods are probabilistic models. These algorithms try to model the probability distribution directly, which is often done by mapping the (typically complex) input distribution to a more simple distribution, such as a standard normal distribution. Examples of probabilistic models are variational autoencoders (VAEs), which – similar to standard autoencoders – encode and then decode the input using a lower dimensional latent space. However, different from autoencoders, VAEs model the latent space with a probability distribution, which allows an assess-

ment of the likelihood of input datapoints up to a lower bound. Another example for a probabilistic model is a normalizing flow, which learns to map the complex input distribution to a simple distribution using a chain of invertible transformations. Different from VAEs, they provide access to the full likelihood of a sample and are therefore able to estimate the probability densities of datapoints directly. Flow-based models will be used extensively in this work and therefore are discussed in detail in the following.

5.4.1 General Principle

Normalizing flows [128] are probabilistic generative models that learn a function mapping an input sample $x_i \in X$ to its respective probability density $p_Z(x_i)$ under the latent space distribution. It does so using a bijective function from the typically complex input distribution to a simpler distribution where the probability can be easily computed.

At the heart of normalizing flows lies the general change of random variables formula. Consider a function f that maps a random variable X to the transformed random variable Z. The probability density of a transformed data point is then given by:

$$p_Z(z) = p_X(x) \left| \frac{df(x)}{dx} \right|^{-1} . \tag{5.71}$$

Essentially, the probability of the sample under the new distribution is given by the probability under the original distribution, normalized by the absolute value of its derivative. The normalization is necessary such that the integral over the new probability density yields a value of one. Equation 5.71 describes the transformation in a univariate case. In a multivariate case, the equation becomes:

$$p_Z(\mathbf{z}) = p_X(\mathbf{x}) \left| \det \left(\frac{\partial f}{\partial \mathbf{x}} \right) \right|^{-1} = p_X(\mathbf{x}) \left| \det \mathbf{J}_f \right|^{-1} . \tag{5.72}$$

In this case, the Jacobian determinant ($\det \mathbf{J}_f$) needs to be computed instead of the single derivative. Equation 5.72 describes a map between two probability distributions. Using a function f to map from the input distribution to a simple distribution, such as a standard Gaussian, the likelihood of a sample can be easily computed.

However, using only this map does not allow for the generation of new samples. To enable normalizing flows for generation, the equality between the inverse of the Jacobian determinant and the Jacobian determinant of the inverse function is used:

$$\left| \det \mathbf{J}_f \right|^{-1} = \left| \det \left(\frac{\partial f}{\partial \mathbf{x}} \right) \right|^{-1} = \left| \det \left(\frac{\partial f^{-1}}{\partial \mathbf{z}} \right) \right| = \left| \det \mathbf{J}_{f^{-1}} \right| \tag{5.73}$$

Therefore, Equation 5.73 can also be written as:

$$p_X(f^{-1}(\mathbf{z})) = p_X(\mathbf{x}) = p_Z(\mathbf{z}) \left| \det \mathbf{J}_{f^{-1}} \right|^{-1} = p_Z(\mathbf{z}) \left| \det \mathbf{J}_f \right| \tag{5.74}$$

Using this formula, the normalizing flow can generate a sample from the simple distribution $p_Z(\mathbf{z})$ and use the inverse map f^{-1} to produce samples that resemble the original input distribution $p_X(\mathbf{x})$.

In practice, f is implemented as a neural network with parameters θ. Since it is difficult to trans-

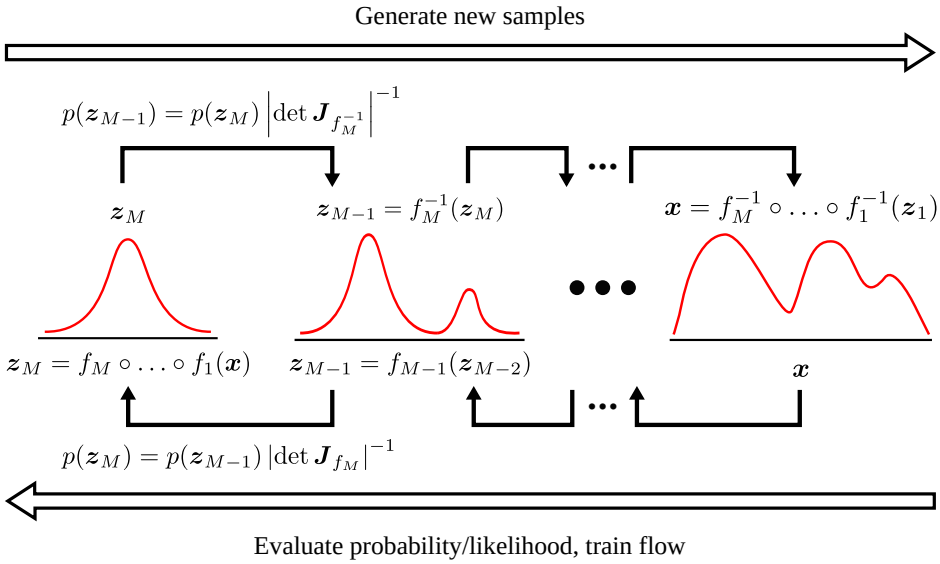


Figure 5.6: A diagram showing the working principle of normalizing flows.

form a complex distribution to a simple one using only a single function, normalizing flows use multiple functions f_i, implemented as subsequent layers in the neural network. Using a chain of M transformations instead of a single one, the Jacobian determinants of each function can simply be multiplied to get to the latent space:

$$p_{Z_M}(\mathbf{z}_M) = p_X(\mathbf{x}) \prod_{i=1}^{M} \left|\det \mathbf{J}_{f_i}\right|^{-1}, \text{ where } \mathbf{J}_{f_k} = \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \text{ and } \mathbf{z}_0 = \mathbf{x}\,. \tag{5.75}$$

Here, $\mathbf{z}_M$ describes the variable of the final latent space. Similarly, new samples can be generated using the inverse functions f_i^{-1}:

$$p_X(\mathbf{x}) = p_{Z_M}(\mathbf{z}_M) \prod_{i=1}^{M} \left|\det \mathbf{J}_{f_i^{-1}}\right|^{-1} \tag{5.76}$$

The general working principle of a normalizing flow is shown as a simplified diagram in Figure 5.6. Using a series of bijective functions f_i, the data can be mapped from the input distribution to the latent space distribution and back. For each subsequent transformation, the input distributions get more similar to the Gaussian latent space. Thus, the data set can "flow" through the transformations, while staying normalized by division of the Jacobian determinant for each of the transformations, hence the name.

A normalizing flow can simply be trained using a negative logarithmic likelihood minimization approach. In particular, the weights of the network are updated to minimize the negative log-likelihood of the transformed datapoints under the latent space distribution, which is equivalent to learning a function that maximizes the likelihood of our data:

$$\mathscr{L}(\mathbf{x}) = -\log(p_X(\mathbf{x})) = -\log(p_{Z_M}(\mathbf{z}_M)) - \sum_{i=1}^{M} \log \left|\det \mathbf{J}_{f_i}\right| \tag{5.77}$$

5.4.2 Masked Autoregressive Flow (MAF)

In order to compute the loss efficiently, the functions f_i, which correspond to the different layers in a normalizing flow network, have to be specifically designed. Naturally, the f_i have to be invertible and have a tractable Jacobian that can be computed easily. In this work, masked autoregressive flows [129] will be used and are discussed in detail in this section. The fundamental idea behind autoregressive flows is to make use of the chain rule of probability that allows to write the density $p_X(\mathbf{x})$ as a product of conditional densities based on the previous components of a variable x_i:

$$p_X(\mathbf{x}) = \prod_{i=1}^{n} p_X(x_i|\mathbf{x}_{1:i-1}) \tag{5.78}$$

Therefore, in a MAF, the f_i are designed in a way such that in their output, one component of the input x_i only depends on the previous components $\mathbf{x}_{1:i-1}$. The output of each layer are the parameters μ_i and α_i of a normal distribution, and the i-th component of $\mathbf{x}$ is modeled as:

$$p_X(x_i|\mathbf{x}_{1:i-1}) \sim \mathcal{N}(\mu_i, (\exp \alpha_i)^2) \text{ , where } \mu_i = f_{\mu_i}(\mathbf{x}_{1:i-1}) \text{ and } \alpha_i = f_{\alpha_i}(\mathbf{x}_{1:i-1}) \tag{5.79}$$

Therefore, the transformation of the i-th component of $\mathbf{x}$ to the latent space can be written as:

$$z_i = f(x_i) = \frac{x_i - \mu_i(x_1, \dots, x_{i-1})}{\exp(\alpha_i(x_1, \dots, x_{i-1}))} \tag{5.80}$$

This transformation $f(\mathbf{x})$ can be easily inverted:

$$x_i = f^{-1}(z_i) = z_i \exp(\alpha_i(x_1, \dots, x_{i-1})) + \mu_i(x_1, \dots, x_{i-1}) \tag{5.81}$$

Since f was defined in an autoregressive way, the Jacobian of Equation 5.80 is a lower triangular matrix such that the determinant can simply be computed by multiplying the diagonal elements, which yields:

$$\left|\det \mathbf{J}_f\right| = \left|\det\left(\frac{\partial f}{\partial \mathbf{x}}\right)\right| = \exp\left(-\sum_i \alpha_i\right) \tag{5.82}$$

Since f was constructed to be both invertible and having a tractable Jacobian, it is an ideal candidate for the implementation of a flow. The remaining challenge is to implement a neural network that ensures the autoregressive property of the μ_i and α_i defined in Equation 5.79. In a MAF, this is achieved using an architecture referred to as Masked Autoencoder for Distribution Estimation (MADE) [130].

A diagram showing the conceptual design of the MADE architecture can be seen in Figure 5.7. In principle, MADE is a fully connected network where connections are dropped in such a way that the outputs only depend on the inputs of the previous components of $\mathbf{x}$. First, each component is assigned an index l_i between 1 and D, where D is the dimension of $\mathbf{x}$. Each node in the input layer is then assigned the corresponding index. For the hidden layers, numbers m_j that range between 1 and $D - 1$ are randomly assigned to each of the nodes, with each number occurring at least once. Then, connections are dropped for all nodes where $l_i > m_j$. This procedure is repeated for all hidden layers in the network. Finally, the output layer nodes are assigned numbers n_k from 0 to $D - 1$. This is done twice to get values for both the α_i and μ_i. Again connections are dropped between nodes where

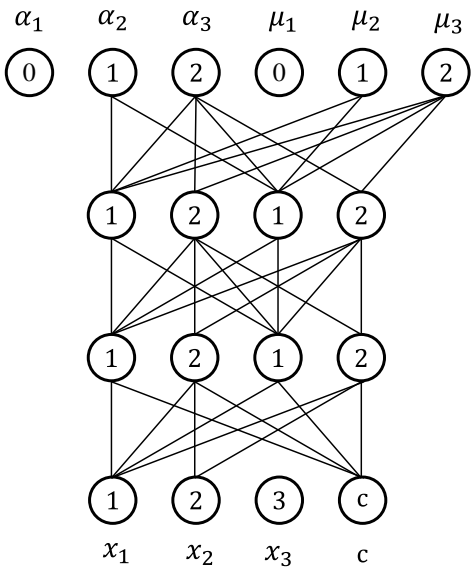


Figure 5.7: Diagram showing the conceptual design of a MADE neural network layer, as it is used in the MAF architecture. A detailed description of the shown components as well as the relation of the node values and the connectivity is provided in the text.

$m_j > n_k$, where m_j denotes the index of the j-th node in the last hidden layer. This procedure yields a network architecture where the i-th output only depends on the input components $1{:}\, i-1$. It can also be seen from Figure 5.7 that the input component with the highest index, here x_3, is not connected to any nodes, since it is the final component and should not occur as a conditional in the autoregressive probabilities. Additionally, the output components with the lowest index – α_1 and μ_1 – also have no connections with any other node, since no lower-indexed components exist that they should depend on. In a MAF, multiple such MADE layers can be stacked to implement the f_i of a normalizing flow as described in Equation 5.75 and Equation 5.76.

In summary, normalizing flows are neural-network-based generative algorithms that allow for a direct estimation of the density $p(\mathbf{x})$. Therefore, they provide direct access to the likelihood of an input data point $\mathbf{x}$ and also allow for the generation of new data points under the approximated input distribution. MAFs in particular are flow models that use stacked autoregressive models implemented as MADE layers to efficiently compute the map between the input and latent space distributions. In this work, MAFs are used as the central components of density estimators that yield state-of-the-art performance in anomaly detection tasks, as will be further discussed in the following chapters.

Chapter 6

Resonant Anomaly Detection

Since the discovery of the Higgs boson by the CMS and ATLAS collaborations [1,2], significant effort was put into searches for new phenomena beyond the Standard Model (BSM) by the LHC physics program. Despite said efforts, no evidence for BSM physics has been found so far. A key problem with most searches for new physics is that they rely on specific signal and background models.

When designing a new search, analysts focus on two major aspects: First, the sensitivity to the signal of interest should be maximized and second, an accurate estimation of the Standard Model background in the region of phase space that is analyzed should be obtained. To achieve the first objective, analysts look for a region in phase space where the ratio of the expected number of signal events and the expected number of background events is high. This region is referred to as "analysis region" and its definition requires significant knowledge about the physics behaviour of the signal, which is why most searches have a high signal model dependence. In practice, this behaviour is modelled using complex Monte Carlo simulations and the simulated events are then used to optimize the definition of the analysis region for maximum signal yield.

For the background estimation, different methods are employed that differ in magnitude of the background model dependence. Similar to the signal case, Monte Carlo simulations are often used to model the background processes corresponding to the investigated signal. For searches that have well-understood background processes, such as electroweak phenomena, the background estimation can be based on the simulation directly, which is the method with the highest dependence on the background model. This method is also often used in the case of rare processes where only few background events are present in the experimental data. In most searches, however, the background estimation is done based on a control region, which is a region in phase space that is orthogonal to the signal region and therefore expected to mainly consist of background events, with the signal contamination being as low as possible. In order to define such a signal-free region, knowledge of the model to search for is required. Inside the control region, data and simulation can be compared and it can be assessed whether the background simulation is modelling the data properly. If this is not the case, calibration factors can be derived to improve its accuracy. The control region method still has a high degree of background model dependence. However, since the background in the analysis region is constrained by the control region, it is more independent compared to the direct estimation approach.

While more data-driven background estimation methods exist, most searches rely on simulation-based approaches. Therefore, Monte Carlo simulation of the physics of both signal and background processes is a key element in these model-dependent analyses. In the past, the model-dependent

approach was highly successful in finding particles and physics processes such as the Higgs boson, which lead to the experimental verification and completion of the Standard Model. For describing new physics beyond the SM, a plethora of theoretical approaches exist, from minimal extensions of the SM to entirely new models. Additionally, many regions in the vast phase space of the LHC are yet unexplored by any search. However, conducting a dedicated search for each theory model in each possible analysis region is not feasible given limited computational and human resources. Therefore, new model-independent methods need to be developed to complement existing search efforts. While model-agnostic searches will be less sensitive to a particular signal model compared to a model-specific search, they enable analysts to search for any kind of anomaly throughout a large region of phase space in a data-driven way.

Several methods have been proposed, some of which were already applied successfully to searches on particle physics data. This work in particular focuses on methods for detecting resonant signals using weakly supervised classification techniques, which will be discussed in the following sections.

6.1 Hadronic Resonances

Many theoretical models that describe BSM phenomena predict the emergence of new particles with non-zero mass that show up as resonances in the respective mass spectrum. A particle with decay rate Γ, resonance mass m_0 and invariant mass m can be modelled as a relativistic Breit-Wigner distribution in the centre-of-mass frame:

$$f(m; m_0, \Gamma) = \frac{2}{\pi} \frac{m_0 \Gamma}{(m^2 - m_0^2)^2 + m_0^2 \Gamma^2}. \tag{6.1}$$

This shape is frequently used in Particle Physics to describe decays of unstable particles. The Heisenberg uncertainty principle ($\Delta E \cdot \Delta t \geq \frac{\hbar}{2}$) relates uncertainty in energy to uncertainty in time. In the case of an unstable particle, the uncertainty in time, Δt, corresponds to the lifetime of the particle, which is inversely proportional to the decay rate:

$$\Gamma = \frac{\hbar}{\tau}. \tag{6.2}$$

Combining this with the uncertainty principle yields:

$$\Delta E \geq \frac{\Gamma}{2}. \tag{6.3}$$

Since the energy is equal to the invariant mass m in the centre-of-mass frame, unstable particles can be produced at a mass that differs from its pole mass m_0 by an amount of the order of its decay width. Thus, instead of a delta function at m_0, a resonance is produced with a width characterized by Γ. It is the objective of resonant anomaly detection methods to detect these localized resonances within distributions of their respective background.

A key point to note when discussing resonant anomaly detection within the scope of model-agnostic searches is that requiring a localized resonance is already increasing the model-dependence significantly. For example, new physics phenomena could result in broad resonances, extending throughout almost the entirety of the mass spectrum or they could result in low-mass/massless parti-

cles, in which case no resonance would appear. Additionally, the mass spectrum that is used depends on the final state of the decay of the hypothesized particle: For particles with entirely hadronic decays, the resonance will appear in the invariant mass of n-jets system in the final state, while for leptonic decays the invariant mass of n-leptons has to be considered. Additionally, one could also consider highly exotic decays with a variety of different jets, leptons and missing transverse energy (MET) in the final state. Therefore, a selection of how the new particle is decaying has to be made for an analysis based on resonant anomaly detection. However, choosing decays into final states that are predicted by a substantial number of theory models allows their analysis throughout large regions of phase space in a single analysis and therefore, the advantage over model-specific searches is retained.

This work focuses specifically on signal models with fully hadronic decays, where final states contain a multiplicity of jets. The major background process to consider for hadronic decays consists of quantum chromodynamics (QCD) events with multiple jets in the final state. The multi-jet QCD background $pp \rightarrow$ two jets is difficult to model due to the many contributing parton-level processes as well as the higher order corrections that would need to be taken into account for an accurate description. However, a simplified assertion can be made, considering the proportionality of the differential cross section:

$$\frac{d\sigma}{dQ^2} \propto \frac{f_1(x_1)f_2(x_2)}{Q^4} = \frac{f_1(x_1)f_2(x_2)}{(sx_1x_2)^2}, \tag{6.4}$$

where the x_i are the momentum fractions of the interacting partons, the f_i are their respective parton distribution functions (PDFs), Q is the four momentum transfer and s is the square of the centre-of-mass energy of the colliding protons. Since the invariant mass of the two-jet system must be equal to the parton-level centre-of-mass energy ($\sqrt{x_1x_2s}$), this approximation can be written in terms of the invariant mass as:

$$\frac{d\sigma}{dm_{jj}} \propto \frac{f_1(x_1)f_2(x_2)}{m_{jj}^4}, \tag{6.5}$$

which shows that the background distribution for this process approximately follows a power law behaviour. Since QCD processes have a high cross section whereas the expected cross section for a potential new physics process is low, the overall distribution in case of the existence of signal is a resonance that is "buried" under a vast amount of background. This behaviour can be seen conceptually in Figure 6.1. Naturally, a key objective in searches for new physics is to identify this resonance in data. One method to achieve this is the so-called "Bump Hunt" which will be discussed in the following subsection.

6.2 The "Bump Hunt"

Before discussing the bump hunt as a method to assess the presence of a new physics resonance from the data, another method, the counting experiment, is considered. In such an experiment, the first step is to compute the expected number of background events, $N_{\text{bg.,exp.}}$, which is then subtracted from the number of events actually observed from the data, $N_{\text{data,obs.}}$, to obtain the excess number of events that are defined as the number of signal events:

$$N_{\text{sig.}} = N_{\text{data,obs.}} - N_{\text{bg.,exp.}}. \tag{6.6}$$

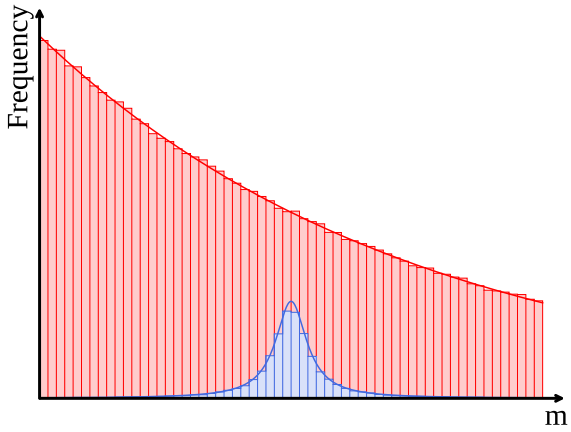


Figure 6.1: Conceptual depiction of a resonant anomaly detection problem. A resonance corresponding to a new particle (blue) is located within a large bulk of (QCD) background events (red).

Ignoring systematic effects for now, the statistical significance of the signal can then simply be obtained by:

$$Z = \frac{N_{\text{sig.}}}{\sqrt{N_{\text{bg.,exp.}}}} \tag{6.7}$$

This definition of the significance uses the Poisson error and is a based on a counting experiment in the limit of infinite events ($N \to \infty$), which is often a valid approximation in the case of high energy physics experiments. Throughout this work, this definition of the significance will be referred to as "statistical significance". The number $N_{\text{bg.,exp.}}$ has to be estimated from data. This is often done by fitting a suitable power law function to the invariant mass spectrum, which is motivated by the physics considerations that led to Equation 6.5. After the fit, the function is then integrated to obtain $N_{\text{bg.,exp.}}$. Conducting the counting experiment using this integral over the entire range of the invariant mass will rarely result in a high significance, since the amount of QCD background is so large that the signal contribution is likely considerably smaller than the statistical error on $N_{\text{bg., exp.}}$.

Therefore, the phase space is typically divided in two regions, namely a signal region (SR), which is a window that covers a large part of the centre of the resonance and a sideband (SB) region, which is defined either as the entire phase space outside of the SR or also as windows spanning the regions adjacent to it. The background function is then fitted either to the SB region and interpolated into the SR or it is fitted on the entirety of the invariant mass spectrum. To obtain the significance, the resulting function is now only integrated within the SR and the statistical significance can be again obtained using Equation 6.7. In a data-driven search, it is unknown where a new physics resonance is located in phase space. Therefore, to conduct a search on the entire mass range, the SR window has to be *scanned* through it, repeating the counting experiment at different points in invariant mass.

One advantage of the counting experiment is that it is simple and entirely agnostic to the actual shape of the signal. As long as the SR window is covering the largest part of the signal contribution, it is irrelevant whether it actually follows a Breit-Wigner distribution or a different, for example more heavy-tailed distribution. On the other hand, using the shape information of the resonance will likely yield a more powerful statistical test. This is what is done in a "Bump Hunt", where a signal-plus-background-fit is employed, that fits the background function with a signal resonance component added. The combined fit can then be compared to the background-only fit and the respective p-value

and significance can be extracted based on their likelihood ratio [131]. This approach allows for the usage of the shape information to achieve a better description of the signal and also the incorporation of systematic uncertainties.

For hadronic resonances, the sensitivity of the bump hunt is limited by the overwhelmingly large QCD background. Anomaly detection methods overcome this limitation by learning to tag anomalous events, increasing the signal-to-noise ratio.

6.3 Weak Supervision

In order to find new physics, it has to be decided for each event whether it originates from known SM phenomena or from a yet unknown process that is not described by the SM. This can be formulated as a hypothesis test, where the null hypothesis, H_0 is defined as the event being a SM event, whereas the alternative hypothesis H_1 states that the event is caused by BSM phenomena, such as a new hadronic resonance. A key metric of a test that tries to distinguish between the two hypotheses is the statistical power, which is defined as $1 - \beta$. The power of a test is the probability that the H_0 is correctly rejected when H_1 is true. This is also referred to as the chance to detect a "true positive". β then describes the chance of failing to reject H_0, even though it is false, which is also referred to as type II error. From a particle physics perspective, the power of a hypothesis test is crucial to detect new phenomena and reject the SM null hypothesis when a different model is actually true. Therefore, obtaining a test with optimal power is desired.

According to the Neyman-Pearson Lemma, one possibility to construct a test with optimal power to distinguish the signal resonance from the background is based on the likelihood ratio:

$$R_{\text{optimal}}(\mathbf{x}, m) = \frac{p_{\text{sig.}}(\mathbf{x}, m)}{p_{\text{bg.}}(\mathbf{x}, m)}, \tag{6.8}$$

where $\mathbf{x}$ is a vector of physics variables. In the case of resonant anomaly detection, it is usually not possible to estimate $p_{\text{sig.}}$ directly from the data, since the signal only contributes marginally to the background-dominated distribution. Therefore, the most viable alternative approach is to estimate the combined probability densities of signal and background (i.e. the density of the entire data) and use the ratio with the background-only density instead:

$$R(\mathbf{x}, m) = \frac{p_{\text{sig.+bg.}}(\mathbf{x}, m)}{p_{bg}(\mathbf{x}, m)} = \frac{p_{\text{data}}(\mathbf{x}, m)}{p_{bg}(\mathbf{x}, m)} \tag{6.9}$$

The density p_{data} can be obtained directly from the data, while the background density is typically estimated from a signal-free region of phase space. Once the likelihoods have been obtained, R can be used as a measure to identify signal events. If a datapoint is signal-like, i.e. its background probability is low, then $R \gg 1$. Otherwise, if the signal probability is low, then $p_{\text{data}} \approx p_{\text{bg.}}$ and therefore $R \approx 1$.

Most methods for resonant anomaly detection approximate the likelihood ratio R as defined in Equation 6.9. Since the development of new methods is an active area of particle physics research, a plethora of different approaches exist and an extensive discussion on each of them is out of the scope of this work (see [132] for an overview). Instead, one family of methods, namely anomaly detection based on weakly supervised classification, is discussed in depth.

Weak supervision is a classification scheme where, instead of distinguishing signal and back-

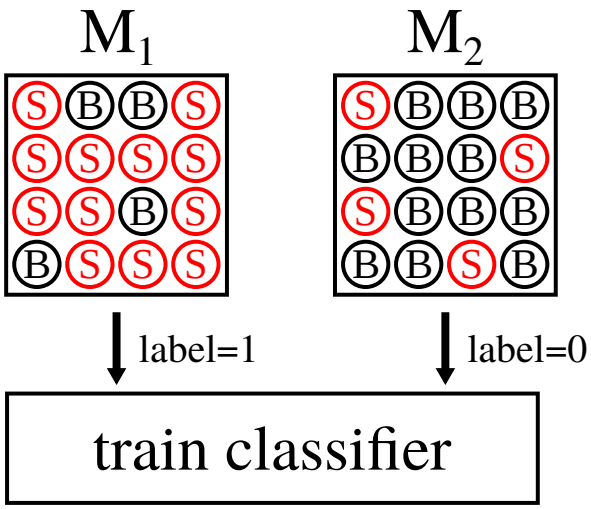


Figure 6.2: Illustration of a weakly supervised classifier training. The classifier distinguishes a signal-enriched from a signal-depleted sample. Signal events are depicted in red while background events are shown in black.

ground directly, the model is trained to discern the origin of a data point from one of two distinct *mixtures* of signal and background samples. This situation is illustrated in Figure 6.2. Originally, this method was proposed in Ref. [133] where it was named "Classification Without Labels" (CWoLa). There, it was shown that the optimal classifier for this weakly supervised task is also optimal for distinguishing signal and background, as long as the signal fraction in one of the mixed samples is higher than in the other. In particular, given two mixed samples M_1 and M_2 in terms of pure signal and background samples S and B and signal fractions $f_1 > f_2$, the optimal classifier to distinguish the two mixtures is $R_{M_1/M_2}(\mathbf{x}) = p_{M_1}(\mathbf{x})/p_{M_2}(\mathbf{x})$. The optimal classifier to distinguish S and B is $R_{S/B}(\mathbf{x}) = p_S(\mathbf{x})/p_B(\mathbf{x})$. The two optimal likelihood rations can be related as:

$$R_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1 p_S + (1-f_1)p_B}{f_2 p_S + (1-f_2)p_B} = \frac{f_1 R_{S/B} + (1-f_1)}{f_2 R_{S/B} + (1-f_2)}. \tag{6.10}$$

Considering the derivative of R_{M_1/M_2} with respect to $R_{S/B}$ yields:

$$\frac{\partial R_{M_1/M_2}}{\partial R_{S/B}} = \frac{f_1 - f_2}{(f_2 R_{S/B} - f_2 + 1)^2}. \tag{6.11}$$

Since $\partial R_{M_1/M_2}/\partial R_{S/B} > 0$ if $f_1 > f_2$, the classifier R_{M_1/M_2} is a monotonically increasing rescaling of $R_{S/B}$. On the other hand, if $f_2 > f_1$, the reverse classifier is obtained and therefore $R_{S/B}$ and R_{M_1/M_2} describe the same classifier.

This feature of weak classification is also used in a variety of methods for resonant anomaly detection. In practice, the mixtures with different signal fractions are obtained by separating the mass spectrum in different regions: A signal region (SR) is defined around the signal mass hypothesis. For mass values below and above the signal region, sidebands (SB) are defined. Depending on the particular method, these can extend through the entirety of the remaining phase space or be defined as smaller regions. Since the signal region events have a higher signal fraction when new physics is present, they can be used as a signal-enriched sample to be classified against a signal-depleted sample from the sidebands to get an approximation of $R(\mathbf{x}|m)$.

This approach, also referred to as CWoLa hunting [134] uses the data in the signal region and sidebands as a direct estimate of p_{data} and $p_{\text{bg.}}$, respectively. The estimation of the optimal unsupervised likelihood ratio R from the data is a key feature of weakly supervised anomaly detection and marks a clear distinction from purely outlier-based anomaly detection methods such as autoencoders. Weak supervision allows the identification of *groups* of events that share specific patterns in $\mathbf{x}$, while outlier-based methods only estimate $p_{\text{bg.}}$ and tag any data points that significantly deviate from this estimate as anomalous, ignoring commonalities between them. While utilizing patterns in $\mathbf{x}$ for detecting groups of signal events is a clear advantage of weakly supervised methods, it comes with the disadvantage of being more model-specific, since it is required that the signal must be localized.

While CWoLa hunting represents the foundation of weak supervision, it assumes that the distributions of features $\mathbf{x}$ are similar in the SR and SB regions. If this is not the case, the classifier will not learn an approximation of the likelihood ratio R, but instead detect the thresholds defining these different regions based on m, which significantly decreases performance.

To mitigate this problem, methods based on conditional density estimation were developed using the probabilistic generative models described in section 5.4. In this approach, the conditional likelihood $p_{\text{bg.}}(\mathbf{x}|m \in \text{SB})$ and for some methods also $p_{\text{data}}(\mathbf{x}|m \in \text{SR})$ are learned from the signal regions and sidebands, respectively. Then the background density is interpolated into the signal region by providing the respective network with values of $m \in SR$ for the conditional. Finally the likelihood ratio $R(\mathbf{x}|m \in \text{SR})$ can be estimated by either training a classifier or taking the ratio of estimated likelihoods directly. The interpolation ensures that the model does not learn to identify the thresholds in m, even for large mass correlations between signal regions and sidebands. The foundational method for this density estimation-based approach was developed in Ref. [135] and named "Anomaly Detection with Density Estimation" (Anode). Additionally, several other such approaches have been proposed, some of which achieve state-of-the-art performance on major benchmarks in the field of anomaly detection. Anode as well as one other approach called Cathode are used in this work and will be discussed in detail later.

Once an anomaly score is obtained from a weakly supervised method, it can be used to identify signal-like events in the data. If a considerable amount of signal events is present, selecting data points with a high anomaly score will result in the signal resonance peak to emerge from the smoothly falling background distribution. This situation is depicted conceptually in figure Figure 6.3. The higher the anomaly score value for selecting events becomes, the more background-like events are rejected, whereas a high number signal-like events is retained. Thus, the peak becomes more prominent for tighter cuts on the score. The significance of the emerging signal peak can then be quantified by conducting the "bump hunt" discussed in section 6.2, using a combined fit of a signal and a background shape.

For the functional form of the background, heuristic functions that have been shown to describe the respective background well are often used in particle physics analyses [136–138]. These originate from the theoretical considerations of the $pp \rightarrow$ jets cross section, as described in the discussion of Equation 6.5. While this approach typically results in a decent description of the background, it has two major caveats. First, the background mass spectrum must be smooth for the fit to succeed. That is, if there are strong correlations between the mass and the anomaly score, additional features will be sculpted into the background, which might cause the emergence of additional "fake bumps" that

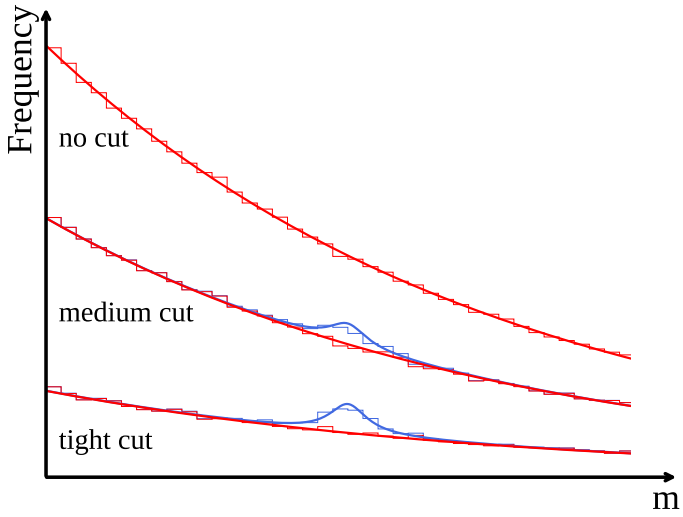


Figure 6.3: Conceptual depiction of the distributions after different cuts on the R anomaly score in the presence of signal.

can be falsely identified as signal or yield an unreliable background estimate. Second, this method adds significant background model dependence to the analysis, since a specific parametric model is considered for the background at hand. Other methods to model the background that do not rely on pre-defined functions exist, such as Gaussian processes [139, 140], but are not yet widely adopted in particle physics searches.

For the signal shape, different resonance parameterizations can be used. A commonly used functional form is a double Crystal Ball function, described by a Gaussian core and power-law tails on both sides. This shape is very flexible as it can also describe resonances that are more heavy-tailed to either side of the distribution. As mentioned in section 6.2, the bump hunting procedure is often repeated and several mass hypotheses are scanned for excesses. Due to the large phase space the number of tested invariant mass points is also large, which can lead to the occurrence of significant results simply by chance. This is also known as the "look-elsewhere-effect", which should be taken into account in analysis.

In summary, the bump hunt is a commonly employed method when looking for new physics resonances in a data-driven way. However, some model-dependence still exists due to the requirement of a localized resonance and the assumption of a particular background parameterization. The bump hunt can be significantly enhanced by employing anomaly detection methods that increase the signal-to-background ratio and ideally lead to the emergence of a resonance bump on top of a smoothly falling background distribution. One family of such methods are weakly supervised methods that distinguish a signal-enriched from an almost pure background sample to tag anomalous events. Using the bump hunting procedure, the significance of the emerged excess can then be computed from a combined fit of a suitable signal resonance shape and the background function.

6.4 Benchmarks and Metrics

6.4.1 The LHC Olympics 2020 Challenge R&D data set

A commonly used data set in high energy physics that is used to benchmark different methods for resonant anomaly detection is the R&D data set of the 2020 LHC Olympics community challenge [132]. The original challenge consisted of three "black box" data sets with unknown signal content that had to be analyzed, as well as the mentioned R&D data set for validation and benchmarking. The latter is a data set for dijet anomaly detection based on simulated proton-proton collisions at a centre-of-mass energy at $\sqrt{s} = 13$ TeV. It consists of one million QCD dijet events and 100 000 signal events. The signal model is the fully hadronic decay of a heavy Z boson, denoted as Z' [9], with $Z' \to XY$ and $X \to q\bar{q}$, $Y \to q\bar{q}$. The masses of the new particles are set to $m_{Z'} = 3.5$ TeV, $m_X = 500$ GeV and $m_Y = 100$ GeV. The events were produced with Pythia 8.219 [141, 142] and Delphes 3.4.1 [143–145], using the default settings without pileup mitigation or multiparton interactions. Event selection was based on a trigger requirement of a single large-radius jet ($R = 1$), clustered with the anti-k_T algorithm and a respective p_T threshold of 1.2 TeV.

The data is provided in form of the detector coordinates (p_T, η, ϕ) for the particles reconstructed by Delphes, which are assumed to be massless [146]. For each event, up to 700 particles are considered. If fewer particles are reconstructed, remaining values are zero-padded to ensure a consistent array size. Finally, the truth bit which is zero for background and one for signal events is added as a feature, leading to a total of 2101 features per event. In addition to the previously described Z' signal, a second signal with a three-prong substructure was also added for benchmarking, using 100 000 events of a heavy W' boson decaying again hadronically as $W' \to XY$, with $X \to qqq$ and $Y \to qqq$. Particle masses, trigger requirements, Pythia and Delphes configurations were kept the same as previously mentioned. Furthermore, a data set with high-level features of the anti-k_T-clustered jets was also provided, which contains the momentum coordinates p_x, p_y, p_z, the mass m_{jet} as well as the n-subjettinesses τ_1, τ_2 and τ_3 for each of the two jets.

While the jet momenta are commonly used features in many analyses, the meaning and importance of the n-subjettiness merits further discussion. The sub-jet information of the jets is often a key feature for distinguishing new physics signals from the large QCD background in fully hadronic final states. Therefore, it is often used in model-agnostic methods that are applied in such a scenario. In many BSM theories, new physics particles are predicted to have a high mass, such as the X and Y particles in the LHCO R&D data set. Since these particles are decaying into significantly lighter SM quarks, they will produce highly boosted jets, that are very close in η and ϕ coordinates. The resulting quark jets can therefore be clustered inside a single, large radius jet, containing two "sub-jets", which is shown conceptually in Figure 6.4. A commonly used variable for describing the subjet structure of a given jet is the n-subjettiness [147], which is defined as:

$$\tau_n = \frac{1}{d_0} \sum_k p_{T,k} \min\{\Delta R_{1,k}, \Delta R_{2,k}, \dots, \Delta R_{n,k}\}, \tag{6.12}$$

where k is the index of the constituent particles in a given jet and $\Delta R_{J,k}$ is the distance in (η, ϕ)-coordinates between a candidate subjet J and the constituent with index k: $\Delta R_{J,k} = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$.

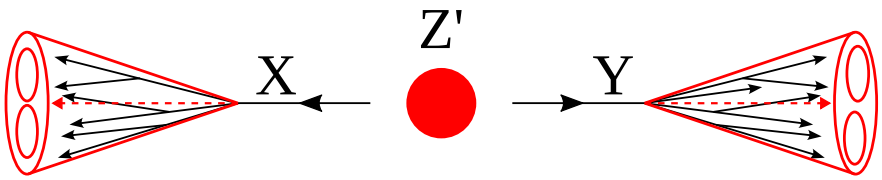


Figure 6.4: Conceptual depiction of the hypothetical Z' particle decaying into particles X and Y, which then decay hadronically. Due to their large masses, the SM quark jets are clustered as sub-jets within a single, large-radius jet.

d_0 is a normalization factor, given by:

$$d_0 = \sum_k p_{T,k} R_0, \tag{6.13}$$

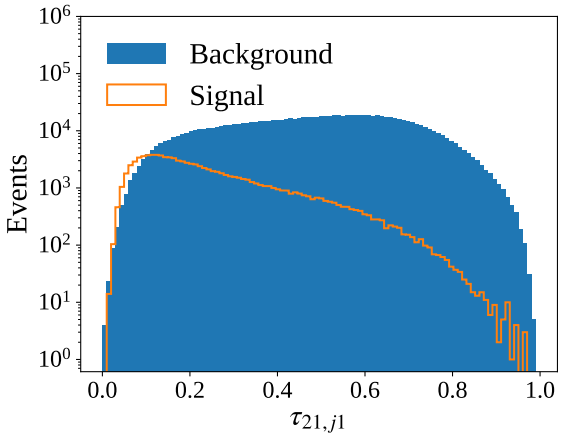
with R_0 being the jet radius used in the original jet clustering algorithm. The candidate subjets J are determined by using a exclusive k_T algorithm that is forced to yield exactly N jets [148, 149].

It can be seen that Equation 6.12 constitutes a variable that quantifies to what extent a jet can be regarded to be composed of n subjets. In high energy physics jargon, this can be referred to as the degree of how "n-subjetty" or "n-pronged" a jet is. The main part of Equation 6.12 is the minimum distance in R between each candidate subjet and the respective constituent k. If a jet contains more than n subjets, the actual clusters of constituents will be distributed further away from the candidate subjets and τ_n is large. If the jet contains exactly n or fewer subjets, the minimum $\Delta R_{j,k}$ will be small and so will τ_n. Therefore, τ_n can be interpreted as the degree that a jet contains $n+1$ or more subjets. However, it is often desired to obtain a variable that directly quantifies how n-pronged a jet is, which is not possible with the τ_n alone. Instead, the subjettiness ratio is often preferred:

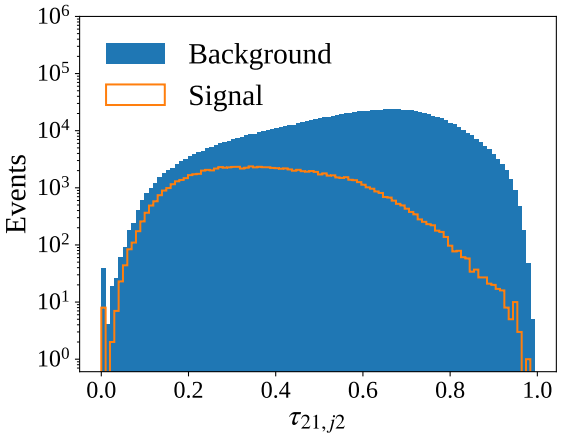
$$\tau_{n,n-1} = \frac{\tau_n}{\tau_{n-1}}. \tag{6.14}$$

If the jet has fewer than both n and $n-1$ subjets, both the numerator and denominator in this equation will be small and $\tau_{n,n-1}$ should be a number $\gg 0$. If the jet has more than n subjets, both τ_n and τ_{n-1} will be large and the subjettiness ratio should again be a number $\gg 0$. In the case where the jet has exactly n subjets, however, τ_n is small while τ_{n-1} is a large value. Therefore, a small $\tau_{n,n-1}$ indicates an n-pronged jet. Furthermore, jets that do not contain subjets and are initiated by a single quark or gluon have a smooth radiation pattern and therefore assume small τ_n for almost any value of n. This leads to subjettiness ratios $\tau_{n,n-1} \gg 0$ for these jets. Thus $\tau_{n,n-1}$ is a good candidate to distinguish between n-pronged jets from boosted particle decays and QCD jets.

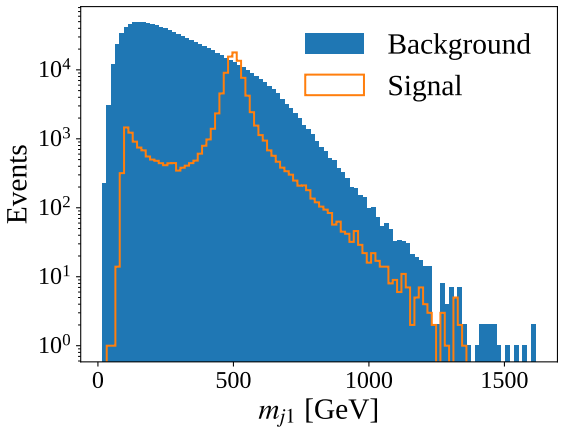
Since in this work, the focus is mainly on methods using high-level observables, some of their distributions are shown in figure Figure 6.5 for the LHCO R&D data set. Distributions are shown for the background events and the Z' signal model separately. The masses of the individual particles can clearly be seen from the plots: Considering the invariant dijet mass distribution, the Z' peak at 3.5 TeV is observed. Similarly, the particles X and Y produce sharp peaks in the invariant jet masses of jet 1 and jet 2 at 500 GeV and 100 GeV, respectively. Both in case of the dijet and individual jet masses, the typical QCD pattern of an steeply falling mass distributions can be observed. However, this is modulated by a turn-on, caused by the trigger requirement in low-mass regions. In other words, the trigger requirement of at least one large-radius jet with a p_T of 1.2 TeV or higher is not achieved



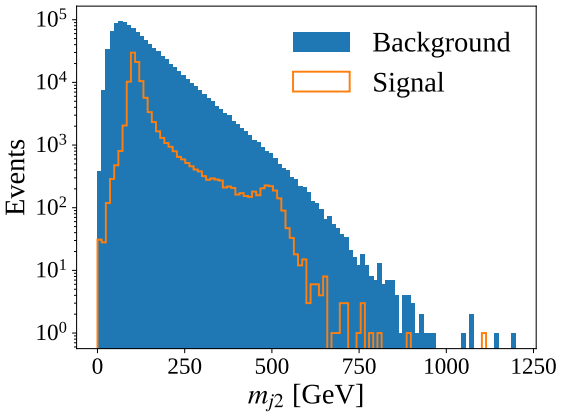
(a) Subjettiness ratio τ_{21} of jet 1.



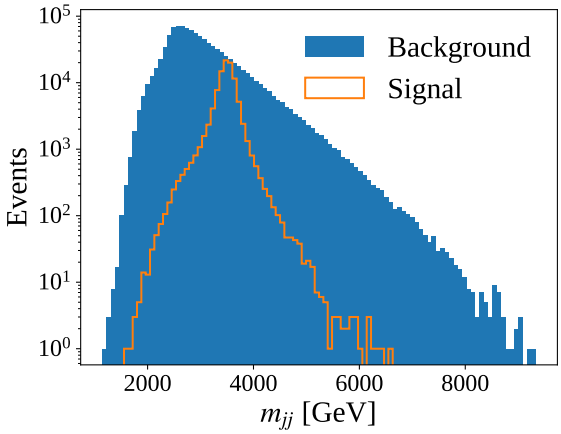
(b) Subjettiness ratio τ_{21} of jet 2.



(c) Invariant mass distribution of jet 1.



(d) Invariant mass distribution of jet 2.



(e) Invariant mass distribution the two-jet system.

Figure 6.5: Several distributions of high-level kinematic variables of the two most massive jets of each event in the LHCO R&D data set. Distributions are shown for the QCD background events (blue) and for the Z' signal events described in the text (orange).

for many of the low-mass events and therefore the efficiency of the trigger is lower than 100 % in that region, as can be seen from the background mass distributions. Finally, the subjettiness ratio τ_{21} of the jets shows the shape that would be expected from the previous discussion of the n-subjettiness. Since the QCD jets have a smooth radiation pattern, both τ_2 and τ_1 are two similarly small numbers, leading to a smooth distribution that leans towards higher values of τ_{21}. For the Z' signal, which is 2-pronged, τ_2 is small while τ_1 is still rather large, leading the distributions to be shifted significantly towards lower values.

In summary, the LHCO R&D data set is a commonly used benchmark for model-agnostic searches looking for anomalous resonances. The fully hadronic final state represents a common scenario, where the objective is to tag a resonance "bump" within an exponentially falling (QCD) background distribution. Therefore, the data set can be used as is often done in model-agnostic searches, where – after a cut on an anomaly score – bump hunting procedures are employed to extract the significance of the signal. Additionally, since truth label information is available, the performance can be easily compared between different algorithms, which makes this data set a cornerstone for benchmarking model-agnostic methods.

6.4.2 The Significance Improvement Characteristic (SIC) Metric

A common measure of interest when searching for new physics is the sensitivity of the employed method. The sensitivity describes the degree to which a method is able to tag signal. Several methods exist to quantify the sensitivity, one of which is the achieved significance. When only taking into account statistical errors (i.e. considering a counting experiment), the significance is simply given by:

$$Z = \frac{N_{\text{sig.}}}{\sqrt{N_{\text{bg.}}}}, \tag{6.15}$$

where $N_{\text{sig.}}$ describes the number of signal events and $N_{\text{bg.}}$ the number of background events, respectively. A difficulty in a model-agnostic search making use of unsupervised machine learning methods is that the sensitivity depends on the amount of signal events present in the analyzed data. One measure of sensitivity is the significance *improvement*, which relates the achieved significance Z at a specific cut on the classifier score with the value of Z that would be obtained without any cut:

$$\text{SIC} = \frac{Z|_{\text{cut}}}{Z|_{\text{no cut}}} = \frac{\text{sig.}}{\sqrt{\epsilon_{\text{bg.}}}}, \tag{6.16}$$

where $\epsilon_{\text{sig.}}$ denotes the signal efficiency (also referred to as true positive rate) and $\epsilon_{\text{bg.}}$ the background efficiency (also referred to as the false positive rate). Thus, a model with a higher SIC is able to tag more signal events given a specified amount of signal injected in a data set.

Similar to the Receiver Operating Characteristic (ROC), the SIC can be computed for different thresholds on the anomaly score to achieve a curve that can be used to compare different models. In contrast to the ROC curve, the SIC does not only contain information about how much signal was selected at a given threshold, but also factors in the background rejection. Since for most anomaly detection scenarios in particle physics the primary objective is to reduce the overwhelmingly large background of SM processes, the SIC curve is the preferred measure for the sensitivity.

It should be noted that the computation of the SIC requires truth-level information to be present,

which is not the case in a model-agnostic, data-driven search. However, for the validation of new methods, data sets based on MC simulations are frequently used such that the expected performance on a certain family of signal models can be assessed. In the case of resonant anomaly detection, a data set such as the previously described LHCO R&D data set can be used for this purpose.

6.4.3 The Idealized Anomaly Detector (IAD)

When comparing different methods for model-agnostic anomaly detection using a benchmark, the construction of an absolute "optimal" or ideal method that these methods can be compared to is often desired. Naturally, one such upper limit for unsupervised methods is to use a supervised classifier with access to the full truth-level information. Using this model as a benchmark, however, leads to an unfair comparison, since no model-agnostic method can be expected to yield similar performance compared to a model with perfect information. In order to find a more suitable upper limit in an unsupervised scenario, it is important to consider the foundational principles of resonant anomaly detection.

As discussed in section 6.1, methods for resonant anomaly detection try to approximate the likelihood ratio R defined in Equation 6.9. Most methods do so using a signal region and sidebands approach: While the signal region should contain the majority of events in the new physics resonance, the sidebands contain almost exclusively background events. The background density $p_{\text{bg.}}$ is then estimated from the sidebands and interpolated, such that R can be computed within the signal region of interest. Several methods have been proposed for the estimation and interpolation of the background density [135, 150–154], some of which will also be discussed in detail in this work. However, an optimal anomaly detection method should exhibit both perfect density estimation of $p_{\text{bg.}}(\mathbf{x}|m \in \text{SB})$ in the sidebands, as well as perfect interpolation into the signal region to achieve the optimal estimate of $p_{\text{bg.}}(\mathbf{x}|m \in \text{SR})$. A simple way to define such an "Idealized Anomaly Detector" (IAD) in a data set that is based on MC simulations is to simply generate more background events inside the signal region using the same simulation parameters. This will create another background-only data set that originates from the exact same distribution as the background distribution of the initial data set and therefore constitutes a perfect estimate of $p_{\text{bg.}}(\mathbf{x}|m \in \text{SR})$. A classifier trained to estimate R on a data set where the background events in the data sample and in the background sample are realizations of the same distribution can therefore be defined as an IAD. It constitutes an idealized benchmark in the limit of both perfect density estimation and interpolation, which is therefore an upper limit for any unsupervised anomaly detection method. Thus, in the studies discussed in this work, the idealized anomaly detector (often abbreviated as IAD), is used for this purpose.

6.5 Anomaly Detection With Density Estimation (ANODE)

ANODE is an algorithm for model-agnostic anomaly detection using direct density estimation. It is one of the first density-estimation-based algorithms in the field and was introduced in [135]. In the original work, it was also applied to the LHCO R&D data set discussed in subsection 6.4.1 and achieved state-of-the-art performance.

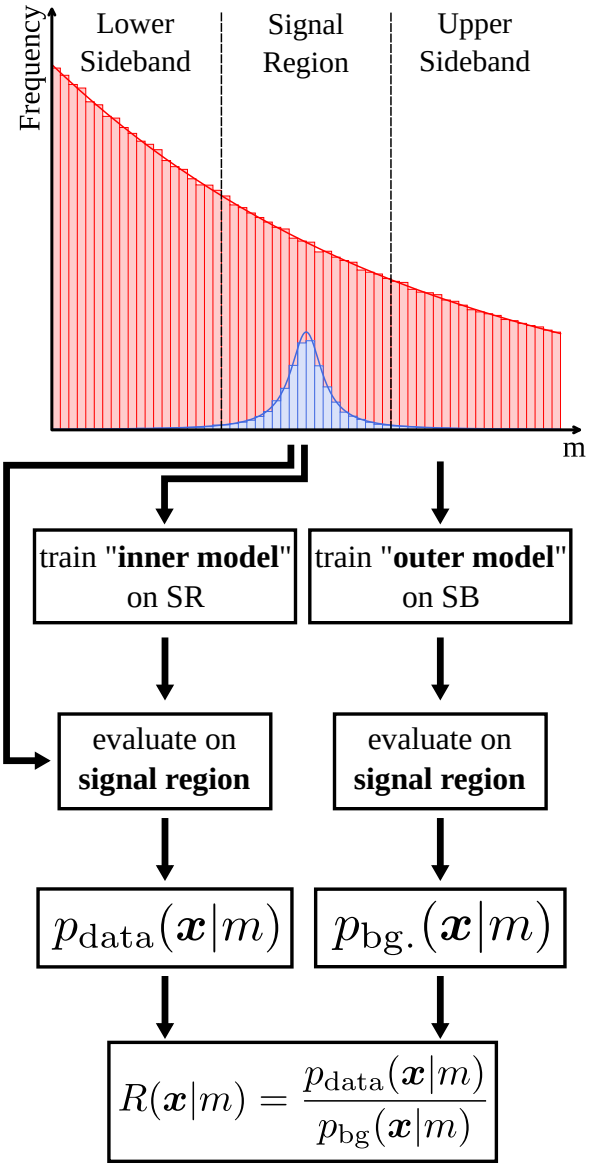


Figure 6.6: A flowchart describing the ANODE algorithm.

6.5.1 The Algorithm

The working principle of ANODE can be seen in Figure 6.6. Similar to other methods for resonant anomaly detection, the mass spectrum of the hypothesized new particle is split up into signal region (SR) and sidebands (SB). The sidebands are defined as the entirety of the mass space that is outside the signal region. ANODE then trains two density estimators: one on the SR, which is also referred to as the "inner model" and one on the SB, being referred to as the "outer model". In the original ANODE paper, a MAF (see subsection 5.4.2) was used for the density estimation, but other flow models or density estimation techniques could also be considered. The inner model learns the density of the data in the SR, $p_{\text{data}}(\mathbf{x}|m \in \text{SR})$, while the outer model learns the background density in the SB $p_{\text{bg.}}(\mathbf{x}|m \in \text{SB})$. As previously discussed, one advantage of the MAF is that it can model conditional distributions, such as the probability densities conditioned on the mass variable. Additionally, they can be easily interpolated into other regions of the mass space by simply providing the respective values as a conditional. Therefore, by using the $\mathbf{x}$ and m values of events in the SR and simply

evaluating the outer model on them, the background density is effectively interpolated from the SB into the SR. The events in the SR are then evaluated on the inner model as well, yielding the data likelihood in the SR. Finally, the ratio of the two values is taken as an estimate of the anomaly score ($R = p_{\text{data}}/p_{\text{bg.}}$) to find signal events. While in the discussion of the likelihood ratio (in particular Equation 6.10 and Equation 6.11) the argument regarding the Neyman-Pearson optimality was made using the joint distributions $p_{\text{data}}(\mathbf{x}, m)/p_{\text{bg.}}(\mathbf{x}, m)$, it should be noted that this argument naturally extends to the conditional distributions:

$$R(\mathbf{x}, m) = \frac{p_{\text{data}}(\mathbf{x}, m)}{p_{\text{bg.}}(\mathbf{x}, m)} = \frac{p_{\text{data}}(\mathbf{x}|m)}{p_{\text{bg.}}(\mathbf{x}|m)} \cdot \frac{p_{\text{data}}(m)}{p_{\text{bg.}}(m)} = \frac{p_{\text{data}}(\mathbf{x}|m)}{p_{\text{bg.}}(\mathbf{x}|m)} = R(\mathbf{x}|m). \tag{6.17}$$

In the penultimate step, it was used that $p_{\text{data}}(m)/p_{\text{bg.}}(m) = 1$ since both the inner and outer density estimators are evaluated on the same SR events using their invariant mass values for the conditional and therefore $p_{\text{data}}(m) = p_{\text{bg.}}(m)$.

6.5.2 Benchmark Performance

In the original ANODE paper [135], the LHCO R&D data set was used for benchmarking performance. The SR was defined around the resonance mass of the Z' with the invariant dijet mass $m_{jj} \in [3.3, 3.7]\,\text{TeV}$. For ANODE, the SB are defined as the entire space in m_{jj} above and below the SR. Features were chosen that capture the substructure information of the two jets. In particular $\mathbf{x}$ contained the mass of the second most massive jet in the event, m_{j1}, the mass difference of the two most massive jets in the event, $\Delta m_j = m_{j2} - m_{j1}$ and the subjettiness ratios $\tau_{21,j1}$, $\tau_{21,j2}$ of the two jets. For the training, 1000 events of the Z' signal model were injected into the full background and the entire sample was split into two equal sets for training and testing. In the SR, this corresponds to about 60 000 background and about 400 signal events in the training and test sets and a signal-to-background ratio of 0.6 %. Additionally, in this paper the performance of ANODE was compared against one of the state-of-the-art algorithms at the time of publication, CWOLA. For CWOLA, the SB was defined using bins of 200 GeV and the events were weighted such that the contribution of events in the upper and lower sidebands are the same and their combined weight is the same as the SR contribution. Other than that, the same number of injected signal events and the same input features were chosen. The classifier used for the CWOLA algorithm was implemented as a fully connected neural network with four hidden layers of 64 nodes each, with a dropout of 10 % being used for each intermediate layer.

The comparison of ANODE and CWOLA alongside a fully supervised classifier and the random performance is shown in Figure 6.7. The figure shows the SIC curves as a function of the signal efficiency. From the figure, it can clearly be seen that the supervised classifier outperforms the more model-agnostic methods by far, which is expected. ANODE and CWOLA perform similarly throughout a wide range of signal efficiencies. For very low signal efficiencies (i.e. very tight cuts on the anomaly score), the SIC values get more affected by the limited statistics of the selected events and therefore have a high variance, at least for the supervised and CWOLA curves. Nevertheless, while the performance of ANODE and CWOLA are similar, still CWOLA at least slightly outperforms ANODE for most signal efficiencies. One reason mentioned in [135] is that CWOLA is training to learn the likelihood ratio R directly, while ANODE only learns the numerator and denominator of R separately.

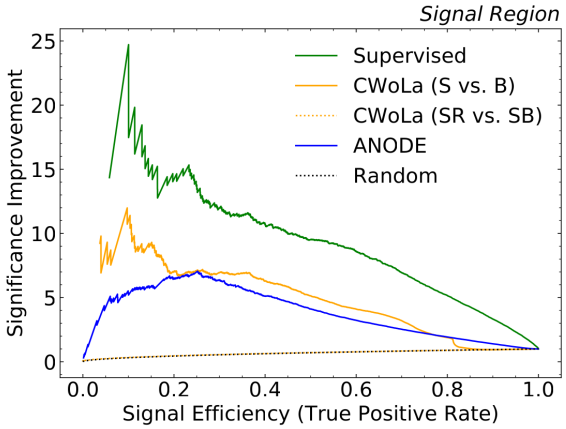


Figure 6.7: SIC curve comparison between ANODE and CWOLA. The supervised performance and the random performance is indicated on the plot for reference. Taken from [135].

Another disadvantage of the ANODE method is that existing generative models often struggle to learn patterns of minority classes inside their training data. Since there are typically only few signal events inside the SR in a particle physics scenario, it is difficult for the MAF to learn the signal contribution within the overwhelmingly large amount of background. This leads to a poor estimate of p_{data} in the SR and therefore to a worse approximation of R, compared to taking data from the SR directly as CWOLA does.

CWOLA has, however, a significant disadvantage compared to ANODE, which is that its performance is significantly impacted by correlations between input variables $\mathbf{x}$ and m. If such correlations exist, a classifier will not learn to distinguish the data from the background likelihoods, but instead simply learn the previously defined thresholds of the SR in m. The key benefit of ANODE is the conditional density estimation and interpolation, which makes it possible to directly learn an estimate of the background density in the SR itself. Since both the inner and outer model are evaluated in the SR exclusively, the thresholds in m cannot be reflected in the learned R score. To further investigate this behaviour, another study has been conducted, where a strong correlation between some the input features $\mathbf{x}$ and m_{jj} was artificially introduced. In particular, the following transformation was applied to the jet masses:

$$m_{j_1,j_2} \rightarrow m_{j_1,j_2} + 0.1 \cdot m_{jj} \tag{6.18}$$

The data set with the shifted features was then used to re-train all of the algorithms in the previous study and again the performance was compared using the respective SIC curves. The result of this study is shown in Figure 6.8. From the figure, the major advantage of ANODE in the presence of correlations can be seen: While all algorithms except for the supervised classifier show a drop in performance, the SIC curve of CWOLA breaks down entirely towards the random line. ANODE on the other hand still achieves SIC values around 4, which is a significant fraction of the performance without correlated features.

ANODE constituted a significant milestone for resonant anomaly detection, being the first application of density estimation techniques for this task and also showing robustness with respect to the presence of correlated features.

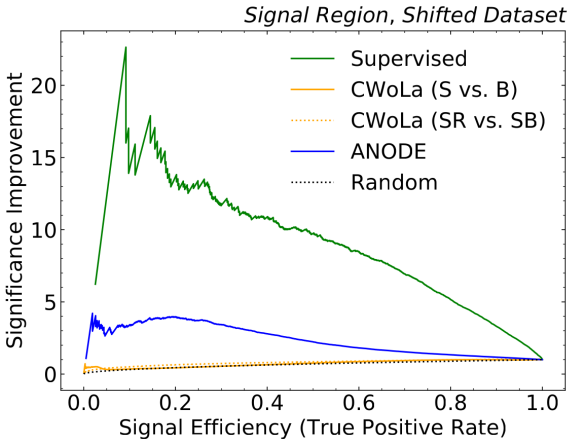


Figure 6.8: SIC curve comparison between ANODE, CWOLA and a fully supervised algorithm in the case of the "shifted" data set, where a correlation between the jet masses and the dijet invariant mass was introduced on purpose. Taken from [135].

6.6 Recent Efforts in Experimental High Energy Physics

Model-independent searches have a long history in experimental particle physics. However, most of them are based on the principle of comparing SM simulations with data to look for differences between them. Examples of these simulation-based searches are found throughout various particle physics experiments, such as DØ [155–157], H1 [158, 159], CDF [160, 161], CMS [3–5] and ATLAS [6–8]. This kind of searches has to be clearly distinguished from the approach that is applied in this work. In principle, these searches are employing counting experiments in various bins, corresponding to different final states. Both the data and the corresponding Monte Carlo (MC) simulated events are counted and compared in each of these bins. The MC events serve as the SM background estimate and it is tested whether a significant excess in the number of data events can be seen. Therefore, this method is entirely dependent on the accuracy of the SM background model that is implemented in the MC simulations.

The enhanced bump hunt on the other hand is entirely data driven and does not rely on MC simulations at all. The weakly supervised methods learn the anomaly score directly from data and the shape of the background estimate – even though requiring a parametric model – is also acquired from a fit to data. Therefore, the classic model-agnostic searches by high energy physics experiments are in contrast with the data-driven approach applied in this work. Other data-driven searches that do not rely on simulations are still rare. However, two analyses conducted by the ATLAS collaboration are the first to use these techniques on measured physics data.

In [162], a resonance search in the invariant mass spectrum of a two-jet system (also referred to as "dijet" system) was done. This search used weak supervision, in particular the CWOLA hunting method described in [133], to search for dijet resonances in the form of a generic, fully hadronic decay of particle A into particles B and C (denoted as $A \to BC$). For these particles, different mass hypotheses for m_A, m_B and m_C were scanned and for each combination, exclusion limits were computed. Mass variations used were $m_A \in \{3\,\text{TeV}, 5\,\text{TeV}\}$ and $m_B, m_C \in \{80\,\text{GeV}, 200\,\text{GeV}, 400\,\text{GeV}\}$. Depending on the mass combinations, several achieved exclusion limits have improved significantly compared to an inclusive dijet search, in some cases up to a factor 10. However, no evidence of new

physics was observed.

In [163], another search for new resonances was performed by the ATLAS collaboration. In this analysis, however, a particle decaying into a jet + Y system was investigated, where the particle Y could be an electron, muon, photon, b-jet or a light jet. Instead of a weakly supervised approach, an autoencoder-based model was used to tag anomalous events. The autoencoder was trained on a sub-sample of 1 % of pre-selected collision events and then evaluated on the remaining 99 % to find anomalies in data. The input features were based on the rapidity-mass matrix (RMM) proposed in [164], which contains a set of variables that is often used for BSM searches, such as missing transverse energy, transverse energies and masses, Lorentz factors, two-particle invariant masses and two-particle rapidity differences. For the anomaly score, the logarithm of the reconstruction loss was used, which is defined as the mean squared error between the input and output of the autoencoder. Using this strategy, the sensitivity is improved significantly, which was illustrated using several benchmark BSM models. However, also in this search no evidence for new physics was found.

Previous to the dijet resonance search [165] that will be discussed in detail later, no model-agnostic search applying unsupervised machine learning methods to experimental data had been conducted by the CMS collaboration. However, a rather similar approach was used in [166], where a three-dimensional maximum likelihood fit in the space of the invariant dijet mass and the individual jet masses ($m_{jj}, m_{\text{jet1}}, m_{\text{jet2}}$) was used to search for (hadronically decaying) di-boson resonances. Compared to previous analyses in this space, this allowed the incorporation of correlations between the individual jet masses and the invariant dijet mass of the SM background processes. For the signal, a double crystal ball shape was fitted to a set of simulated events at different mass hypotheses and then interpolated to intermediate mass values to allow for a more granular scan. With this method, the sensitivity was improved up to 30 % compared to previous searches and the exclusion limits set were the best to date in the dijet final state.

Apart from actual analyses on experimental data, several new methods have been proposed that can be used to search for new physics in a model-agnostic way. While an in-depth discussion on each of them is beyond the scope of this thesis, they will be shortly summarized in the following. Three main categories of methods can be distinguished in principle: Out-of-distribution (OoD) detection methods learn an estimate of the background and tag any event that significantly deviates from it as anomalous. In particular, these are often implemented using autoencoder-based methods, which have also been proposed for model-agnostic searches in high energy physics [167–172]. As discussed previously, weakly supervised methods are another class of model-agnostic search methods, that learn the likelihood ratio R between the data and the background-only hypothesis [133, 135, 150–152, 173–175]. In this field, approaches with and without using probabilistic modeling to estimate R exist. Finally, the third category is based on semi-supervised methods that are more model-specific but make use of signal "priors" to encode the latent space of a probabilistic model [176]. While this approach needs MC simulations for at least some signal models, it has the advantage of being sensitive to new signals that share different anomalous features from the ones that the algorithm was trained on.

While the first analyses using model-agnostic anomaly detection to experimental data achieved promising results, there is still significant potential for new searches. The discussed analyses made use only of a single method. In a model-agnostic scenario, however, the method to yield the optimal sensitivity cannot be defined, since the signal model is entirely unknown. Therefore, it is expected

that different algorithmic concepts are complementary to each other with respect to different kinds of anomalies and should therefore be analyzed. Furthermore, the input features used for training the unsupervised models have been rather limited so far, typically resorting to the dijet and individual jet invariant masses. However, as many features as possible should be used in the analysis of a model-agnostic setting, since an anomalous pattern could emerge from any of them, or from complex non-linear correlations between them. Therefore, the main focus of this work is a new model-agnostic search performed by the CMS collaboration, considering a large variety of methods – one from each of the discussed categories – as well as a plethora of different signal models and mass hypotheses. This is done using different input features for each method. Some methods are entirely based on low-level particle information, others depend on high-level features computed for each event or jet, while others combine the two kinds of features. This broad search scheme is unique in the field of high energy physics so far, and enables significant potential for new discoveries directly from data. The respective analysis is discussed in the following chapter.

Chapter 7

Applying Anomaly Detection to CMS Experimental Data

In this chapter, the development and application of CATHODE, a novel method combining density estimation and weak supervision for data-driven anomaly detection is discussed. First, the method is introduced based on simulated data from the LHCO R&D data set. Then, CATHODE is applied to a resonance search in the dijet final state with anomalous jet substructure using proton-proton collision data measured by the CMS experiment during run 2 in the years 2016 to 2018 at a centre-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$. This constitutes a first-of-its-kind search, comparing not only CATHODE, but several state-of-the-art methods regarding expected sensitivity and achieved exclusion limits on the production cross-section of different signal models. Additionally, results for a generic, largely model-agnostic search through the invariant dijet mass spectrum are discussed.

7.1 Classification Through Outer Density Estimation (CATHODE)

The work presented in this section has been previously published in [150], in collaboration with Anna Hallin, Joshua Isaacson, Gregor Kasieczka, Claudius Krause, Benjamin Nachman, Matthias Schlaffer, David Shih and Manuel Sommerhalder. The figures and the written content closely resemble or match the information found in this article. My contribution to the publication consists of contributing to the analysis code of the method, conducting and documenting the background sculpting and bias studies, general paper writing and proof-reading and the assessment and discussion of key research questions and hypotheses during the development of the method with the mentioned collaborators.

In the following, the method "Classification THrough Outer Density Estimation" (CATHODE) is discussed. Similar to ANODE, CATHODE is a method for model-agnostic anomaly detection using neural density estimation. Additionally, CATHODE combines this approach with the weakly supervised approach employed in the CWOLA algorithm. As discussed in section 6.5, CWOLA shows poor performance in the case where correlations between input features $\mathbf{x}$ and the resonant variable m exist. While ANODE overcomes this problem through the use of conditional density estimation and interpolation, it shows a slight decrease in performance compared to CWOLA, since it does not learn the Neyman-Pearson optimal likelihood ratio, R, directly. Additionally, it is difficult for the inner

density estimator in Anode to learn the signal component within the large number of background events. Cathode aims to combine weak supervision and density estimation to achieve the "best of both worlds": Being able to learn R directly while still being robust against correlations between $\mathbf{x}$ and m.

7.1.1 The Algorithm

The working principle of Cathode is shown as a flow chart in Figure 7.1. Similar to Anode, a signal region (SR) window is defined in the mass spectrum, where the hypothesized resonance is located. The remainder of the mass phase space is defined as sidebands (SB). The algorithm then proceeds to train a conditional density estimator on the events in the SB. In the original Cathode paper, a MAF is again used for this purpose [150]. Since it is assumed that the SB contain almost exclusively background events, this "outer" density estimator learns the density $p_{\text{bg.}}(\mathbf{x}|m \in \text{SB})$. Again similar to the Anode algorithm, this density is interpolated into the signal region. However, instead of simply evaluating the data events in the SR using the MAF model, its generative capabilities are used. A sample of the learned density is created, using m values from within the signal region as a conditional in the process. This sample is also referred to as "background template". To get an approximation of the mass distribution in the SR, a kernel density estimator (KDE) is trained on the mass values of the SR events and from this KDE, the mass values for conditioning the outer MAF are sampled. Sampling from the MAF in this way results in a data set that should approximate the background distribution in the SR, $p_{\text{bg.}}(\mathbf{x}|m \in \text{SR})$. Then, similar to the CWoLa algorithm, an actual data sample is drawn from the SR and then a weakly supervised classifier is trained to distinguish this data sample from the background template. This classifier learns an approximation of the likelihood ratio R, based solely on the approximation of densities in the SR.

7.1.2 Benchmark Performance

The Cathode algorithm was benchmarked using the LHCO R&D data set. In a dedicated study, the performance of Cathode was compared to other state-of-the-art algorithms as well as fully supervised classification and the idealized anomaly detection scheme described in subsection 6.4.3. The used input features were again the invariant mass of the lighter of the two jets, m_{j_1}, the difference in invariant mass between the jets, $\Delta m_j = m_{j_2} - m_{j_1}$ as well as the subjettiness ratios τ_{21,j_1} and τ_{21,j_2} for each of the jets. As the conditional feature of the MAF, the invariant dijet mass, m_{jj} was used. Before training the density estimator, the input features were preprocessed by first scaling them to the range $(0, 1)$, then logit transforming them and finally standardizing them by subtracting the mean and dividing by the standard deviation. For the signal model, the hadronically decaying Z' was investigated and the signal region window was defined as before with $m_{jj} \in [3.3, 3.7]\,\text{TeV}$, while the remainder of m_{jj} was defined as the sidebands. The entirety of the 1 000 000 QCD background samples is used in this study, with 1000 signal events being injected. The resulting data set contains approximately 120 000 background and 772 signal events in the SR and 880 000 background and 228 signal events in the SB. For the SR, this again corresponds to a signal-to-background ratio of about 0.6 %. For the training of the outer density estimator, the data in the SB region is divided into a training set that consists of 500 000 events and a validation set that contains the remainder of 378 876 events. The MAF density

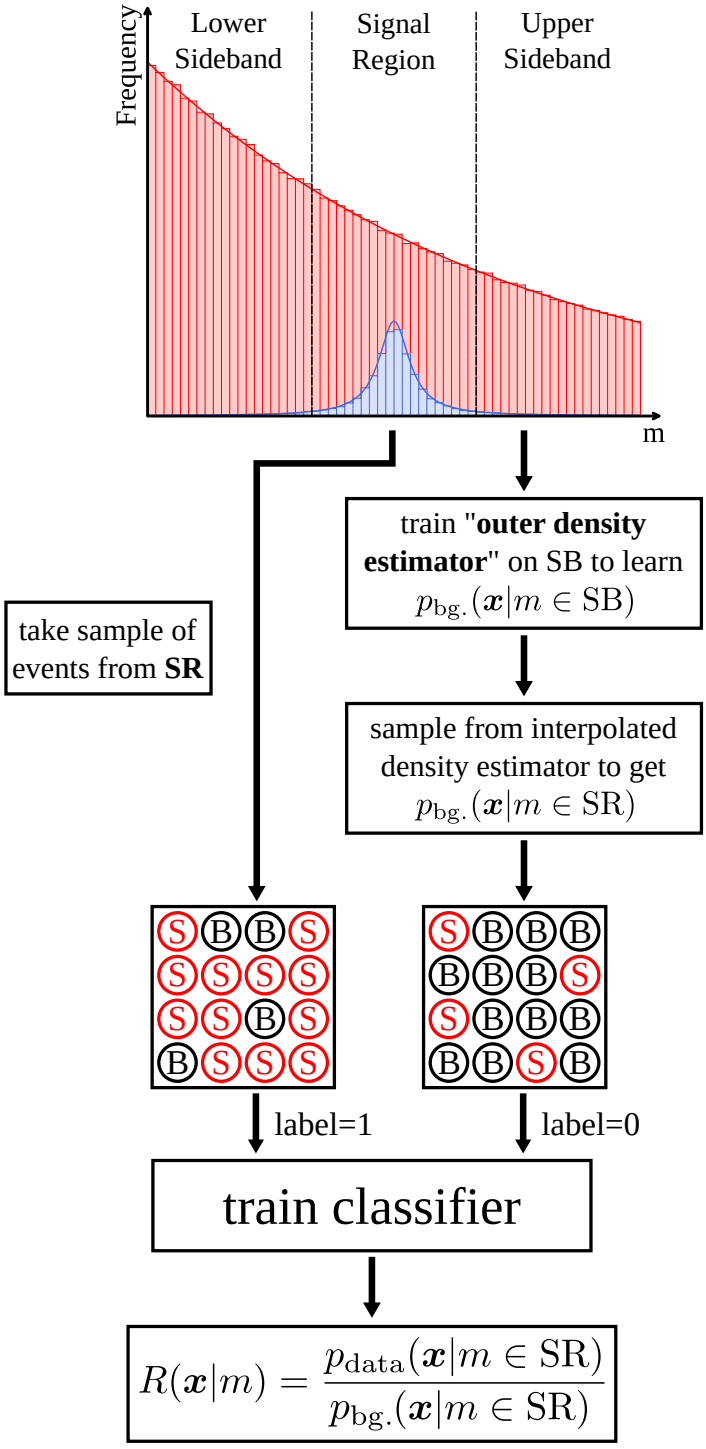


Figure 7.1: A flowchart describing the CATHODE algorithm.

Hyperparameter name	Hyperparameter value
No. of epochs	100
Optimizer	Adam [108]
Batch size	256
Learning rate	10^{-4}
Momentum for batch normalization	1.0
Number of MADE blocks	15
Hidden layers per MADE block	1
Nodes per hidden layer	128

Table 7.1: Hyperparameters of the MAF density estimator model used in CATHODE.

estimator is implemented using the PyTorch framework [109] and the hyperparameters used in the training are shown in Table 7.1.

For model selection, not only the single best validation loss epoch was used, but instead, the model states at the ten lowest validation loss epochs were selected for further analysis. This allowed for ensembling different model states and increased the robustness of the overall result. Prior to sampling, the KDE was trained to fit the m_{jj} values of the training set of the SR data. It was implemented using the `scikit-learn` library [177], with a Gaussian kernel and a bandwidth of 0.01. For each of the ten selected MAF model states, new events were now sampled, using m_{jj} values produced by the KDE as a conditional. However, instead of sampling the same number of events in the SR, it has been shown that oversampling background template events had a positive impact on the weak classification result. Therefore, from each of the ten selected model states, 40 000 events were sampled, leading to 400 000 background template events in total. For the weakly supervised training, these were split equally into 200 000 events each for training and validation sets.

A comparison of the data distribution in the SR and the background template distribution is shown in Figure 7.2. For these plots, the preprocessing of the features was reversed such that their distribution is assessed in the physical input space. As can be seen, the artificially generated samples of the background template describe the input distributions of the background in the actual data well and there are only marginal mismodellings. Considering the m_{jj} distribution, it can also be seen that the KDE samples follow a similar distribution as the background in the SR. At the same time, the KDE does not overfit on the exact background distribution and still retains some variance.

Since the density estimation and subsequent interpolation yield a proper description of the background density in the SR, the weakly supervised classifier can be trained to estimate the likelihood ratio R. The classifier model is also implemented as a PyTorch model, with the hyperparameters chosen as in Table 7.2. For the training, the approximately 120 000 data events in the SR are split into 60 000 events each for the training and validation sets. Therefore, the objective in the weak classification task is to distinguish the 60 000 training set events from the signal region, from the

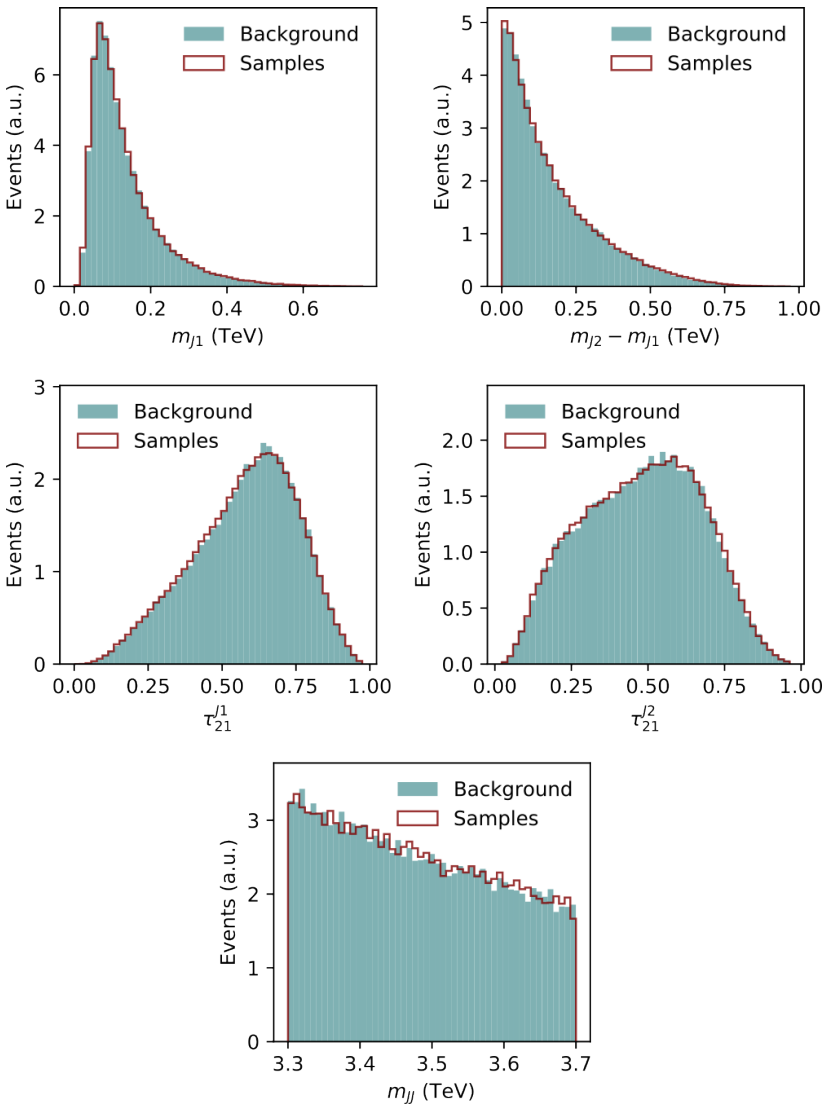


Figure 7.2: A comparison of normalized feature distributions of the background events from the data and events from the background template in the signal region.

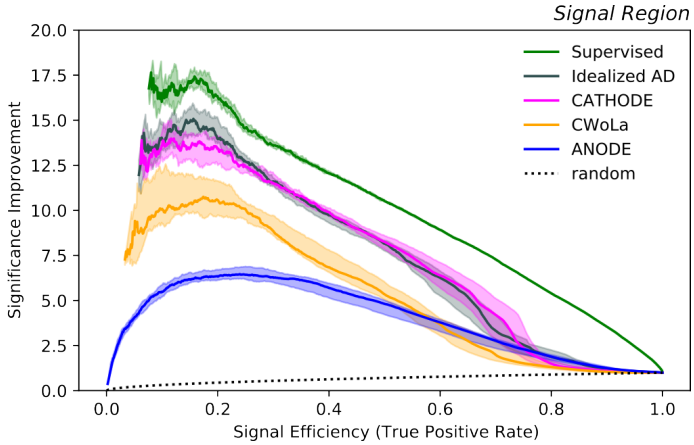


Figure 7.3: SIC curve comparison of CATHODE with various other algorithms. Solid lines indicate to the median performance of 10 independent re-trainings of the respective algorithm, while shaded areas show to the inner 68 % of this distribution over SIC curves.

200 000 training set events of the background template. Since the oversampling of density estimators causes the two classes to be imbalanced, a class weight is computed to re-balance events both for the training and validation sets such that their contribution is equal. Additionally, the preprocessing used for classification is different from the one used in density estimation: The datapoints are simply re-standardized to have a mean of zero and a standard deviation of one. Applying a scaling followed by a logit transformation consistently caused sub-optimal performance in classification, which is why these steps were omitted. Similar as before, the training and validation losses of the weak classifier are recorded. For the model selection, again the 10 lowest validation loss epochs are chosen and an ensemble prediction is constructed in the final evaluation by averaging their individual predictions. This ensemble prediction again has proven to yield a more robust result, significantly reducing the variance of this noisy, weakly supervised classification task.

For the final performance comparison, the fully trained CATHODE model is evaluated on the test set. For the training and validation sets, the labels only contained the information whether an event was from the SR data or from the background template. The final evaluation, however, is done on the full truth-level signal vs. background information. While this information would not be readily available in an actual, data-driven search for new physics, it is useful in the synthetic scenario of the LHCO R&D data set to benchmark the performance of CATHODE and compare it against other methods. The result of this comparison can be seen in Figure 7.3. In this figure, CATHODE is compared to four other methods.

First, a fully supervised classifier, having access to the full truth-level information about an event being signal or background, was also trained and evaluated for comparison. This classifier constitutes an overall upper bound on the performance. The classifier model was implemented as a fully connected neural network, with the exact same hyperparameter settings as for the weak classifier of the CATHODE model, summarized in Table 7.2. The concept of the idealized anomaly detector (IAD) was already introduced in subsection 6.4.3: it is a weakly supervised classifier, trained in the limit of perfect density estimation. Therefore, it constitutes an upper performance limit for any weakly supervised method. The perfect density estimation is implemented using additional events from the true $p_{\text{bg.}}$ by simply generating more MC events with the exact same settings as in the original LHCO

Hyperparameter name	Hyperparameter value
No. of hidden layers	3
No. of nodes per hidden layer	64
Batch size	128
No. of epochs	100
Optimizer	Adam [108]
Learning rate	10^{-3}
Momentum	0.9

Table 7.2: Hyperparameters of the DNN classifier model and its training.

R&D data set. In total, a sample of additional 612 000 QCD events in the SR were generated this way. From those, 272 000 events are taken as the idealized version of the background template and split into half for the training and validation set, respectively. The task of the IAD is then to distinguish these events from the data of the original LHCO R&D data set in the SR.

Furthermore, CATHODE is compared with two other state-of-the-art anomaly detection algorithms, ANODE and CWOLA. For ANODE, the same MAF architecture and training settings were used as for CATHODE. An outer model was trained on the SB region and another, inner, model on the SR. Then, the R score was built – as described in the previous section – by evaluating both the inner and outer models on the SR and computing the respective likelihood ratio. For the CWOLA model, a classifier was trained using the exact same hyperparameter settings as the CATHODE classifier. For the SR, the same m_{jj} window was used as before. The SB region, however, was reduced to 200 GeV wide strips adjacent to the SR. The CWOLA SB region contained a total number of 130 232 background-like events that are split equally into training and validation sets. Events are also reweighted for the training, such that the upper and lower SB events contribute equally and in total have the same weight as the SR events. For all of the described methods, 10 independent re-trainings were conducted in order to assess the variance of the results. In the figure, the solid lines correspond to the median of the 10 SIC curves, while shaded areas correspond to the inner 68 % of this distribution.

From the plot it can again be seen that a fully supervised classifier outperforms all other methods, including the idealized anomaly detector, throughout the entire range of signal efficiencies. It also has by far the lowest variance of results. The IAD shows the next best performance, being closely followed by CATHODE. Only marginal differences can be seen between the two methods, the largest being in the lower signal efficiency region. The variance of results is also similar comparing IAD and CATHODE models. This is a significant result and shows that CATHODE almost saturates the optimal performance. It also underlines the findings of Figure 7.2, that the density estimation and interpolation is close to the actual background distribution in the SR, since even perfect density estimation does not improve the final performance significantly. CATHODE also considerably outperforms other state-of-the-art models, with the performance of CWOLA and ANODE being worse for all signal efficiencies. Similar

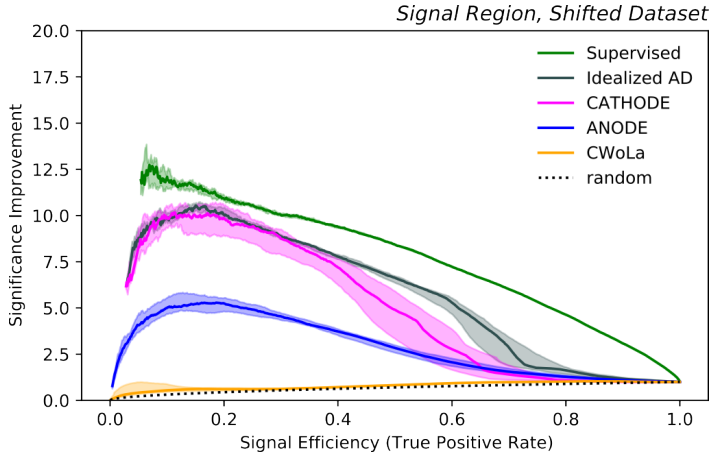


Figure 7.4: SIC curve comparison of CATHODE with various other algorithms on a shifted data set, where correlations between input features $\mathbf{x}$ and m_{jj} were artificially introduced. Solid lines correspond to the median performance of 10 independent re-trainings of the respective algorithm, while shaded areas correspond to the inner 68 % of this distribution over SIC curves.

to the previously discussed studies from the ANODE paper, CWOLA outperforms ANODE significantly. This is probably due to the chosen input features not being highly correlated, such that CWOLA can learn the likelihood ratio well. ANODE on the other hand suffers again from the caveat that it does not directly optimize R, but instead learns only its individual components, which – similar to the results in the original paper – seems to result in a worse estimate of R compared to the weakly supervised methods.

Another advantage of CATHODE is its ability to interpolate the learned background density from the SB region into the SR, using conditional density estimation. Similar to ANODE, this should lead to increased robustness with respect to correlations between input features $\mathbf{x}$ and the invariant dijet mass, m_{jj}. This was tested by artificially introducing such correlations using the following transformations:

$$\begin{aligned} m_{j_1} &\to m_{j_1} + 0.1 m_{jj} \\ \Delta m_j &\to \Delta m_j + 0.1 m_{jj} \end{aligned} \tag{7.1}$$

Then, the same performance comparison study as in Figure 7.3 was repeated using the shifted features. The results can be seen in Figure 7.4. All algorithms show reduced performance compared to the non-transformed features. A probable reason for this is that the original input features, m_{j_1} and Δm_j, get distorted by the transformation and the original signal patterns are more difficult to extract, even when full truth-level information is available. In other words, the contribution of m_{jj} adds significant noise to the original features, leading to a performance decrease. Considering the relative performance between algorithms, the overall situation is very similar to the original study. The supervised classifier still outperforms all other algorithms and the IAD is the second best performing algorithm, while showing only marginally better performance than CATHODE. CATHODE again saturating the idealized limit, even in the presence of strongly correlated features, is a remarkable finding. It shows again that density estimation and interpolation indeed leads to increased robustness against correlations, retaining much of the original significance improvement. This is also shown considering the performance of ANODE, which also only saw a marginal performance decrease compared to the non-correlated scenario. Another notable difference can be seen considering the SIC curve of the

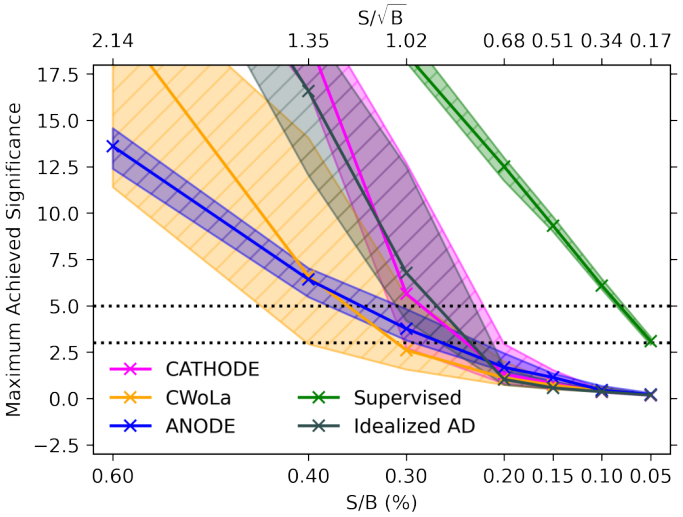


Figure 7.5: It shows the maximum achieved significance, which is the product of the SIC achieved at the anomaly score cut and the uncut significance, as a function of the signal-to-background-ratio (S/B) and the statistical significance $S/\sqrt{B}$. Again the solid lines correspond to the median and shaded areas to the inner 68 % of the SIC distribution based on 10 independent retrainings. Different from previous studies, not only the classifiers were re-trained, but also different random realizations of the injected 1000 signal events were used. Dotted lines indicate significance value corresponding to 3 σ and 5 σ.

CWoLa algorithm. In the presence of highl correlations between input and conditional features, it breaks down entirely to random performance, which again shows this key weakness of this algorithm. In an actual, model-agnostic search, as many features as possible should be used, since it cannot be known which of them contain patterns that a new physics signal could emerge from. Therefore, having to select non-correlated features by hand prior to analysis is a major caveat for any model-agnostic method. This shows the significance of the ANODE and CATHODE methods in the field, being able to also find anomalies in highly correlated scenarios. CATHODE in particular also shows exceptional performance, saturating the theoretical upper limit of any weakly supervised method.

As discussed previously, the performance of weakly supervised methods depends significantly on the amount of signal present within the SR. Therefore, another important matter to study is how the performance of the investigated algorithms behaves for different numbers of signal events injected into the QCD background. Thus, another study was done investigating the sensitivity of the methods with respect to different injections. The result is shown in Figure 7.5. The same settings as in the previous study were used for the training of the different algorithms. However, one key difference of this study is that not only the classifiers were re-trained for assessing the variance of the results, but also the 1000 injected signal events were chosen randomly each time, as were the training and validation sets. Starting from the original injection of $S/B = 0.6\,\%$ in the SR, corresponding to a significance of $S/\sqrt{B} = 2.14$, the signal injections are subsequently reduced down to $S/B = 0.05\,\%$. In total, six different injections are investigated. From the plot, it can be seen that up to injections of $S/B = 0.2\,\%$ none of the weakly supervised methods shows a considerable significance improvement. For higher injections, however, the different methods start to diverge in performance. At $S/B = 0.3\,\%$, the IAD and CATHODE have already achieved a significance larger than 5 σ, which in high energy physics is

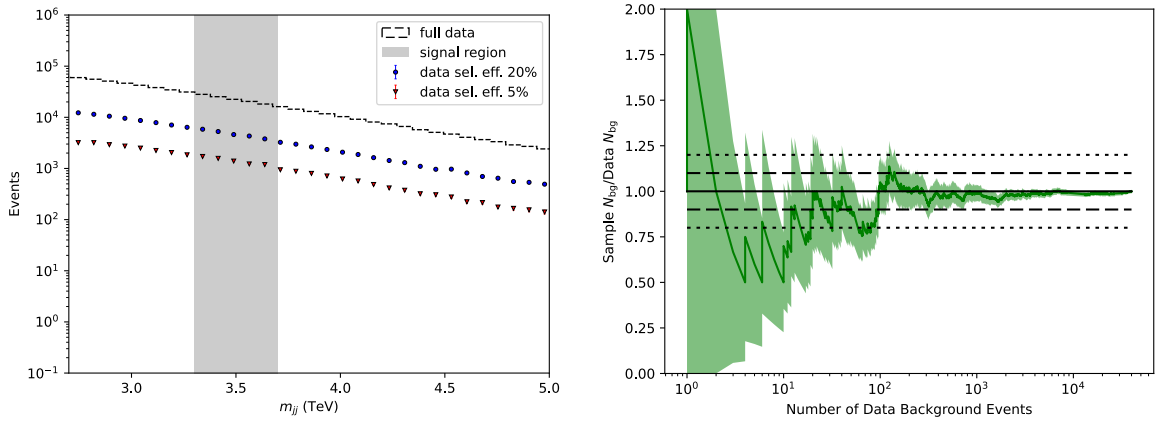


Figure 7.6: Results shown are based on trainings on background events only. Left panel: The plot shows the m_{jj} distribution without any selection applied (dashed histogram) as well as after selecting the 20 % and 5 % most anomalous events according to the learned anomaly score. The SR is indicated in light grey. Right panel: Ratio of selected template and selected data background events in the SR after different cuts on the anomaly score, as a function of the number of actual background events after the respective cut. Shaded areas correspond to the statistical uncertainty on the number of template and actual background events, propagated to the ratio. Dashed lines describe the values for a 10 %, dotted lines for a 20 % deviation from the optimal value of 1.

considered the threshold for discovery. At the same point, ANODE achieves a significance larger that 3 σ, which is corresponds to evidence of a new particle. CWOLA, however, achieves a median performance just below that value. In general, the IAD and CATHODE are the methods where the median performance crosses the 3 σ and 5 σ lines first, except for the fully supervised line. Two caveats of weakly supervised methods in general can also be seen in this plot. First, due to the noisy nature of the weak classification task, the variances of the results are large and overlap throughout almost the entire plotting range. This makes it difficult to assess whether one method is significantly better than another and makes ensembling or several re-trainings necessary, which increases the computational costs of studying these methods. Second, weakly supervised methods in general are a trade-off to increase model agnosticity at the price of lower sensitivity compared to dedicated searches. This can clearly be seen from this study, where a supervised classifier already reaches the threshold of evidence at the lowest injection, whereas even the idealized weak classifier is reaching this value at an injection that is approximately five times as high. While this trade-off is known for model-agnostic methods and has been previously discussed, the study shows that improving the sensitivity for weakly supervised approaches should be a key objective of future research efforts.

When searching for new physics resonances in a model-independent way, a key factor is the extraction of the signal. In many cases, the approach of "bump-hunting" which was discussed in section 6.2 is used for this purpose. Therefore, one important feature of any model-agnostic approach is that it must not sculpt artificial bumps or features into the invariant mass spectrum. In other words, after selecting the most anomalous events based on the respective anomaly score, it must be possible to apply a smooth fit to m_{jj}. To investigate the sculpting of CATHODE, the algorithm was trained only on background events and the invariant dijet mass distributions at different selection efficiencies were then compared to the uncut distribution. The result of this study can be seen in the left panel of Figure 7.6. The plot shows that CATHODE does not sculpt artificial features into the invariant dijet

mass spectrum, neither within nor outside the SR window, even for percent-level cuts that are usually applied for analyses in high energy physics.

Another approach to study possible sculpting issues with the CATHODE approach is to investigate the bias of the learned estimate of $p_{\text{bg.}}$. The main idea behind this investigation is that if this estimate is unbiased, the same number of events should be selected from the data and the template events in the SR. Therefore, as many template events were generated within the SR as background test set events exist. Then, both template and actual background events were evaluated on the CATHODE model and the ratio of the number of events being selected after a cut on the anomaly score was computed. The results of this study are shown in the right panel of Figure 7.6. From the figure, it becomes clear that there is no significant bias present in the SR background estimate, since the ratio is close to 1 and stays within a fluctuation of 10 % throughout a large part of the domain. Deviations increase only in the region of low statistics, which is expected. Therefore, based on these studies it can be assessed that CATHODE does not sculpt the background and produces a largely unbiased estimate of the background density, $p_{\text{bg.}}$ in the SR. Signal extraction methods typically applied to model-agnostic approaches such as bump hunting can thus be used for signal extraction.

In summary, CATHODE marks a significant contribution in the development of methods for resonant anomaly detection. It constitutes a method that is not only performing similar to the theoretical upper limit of any weakly supervised approach, but also is highly robust against correlations between input features $\mathbf{x}$ and the conditional m. This enables the use of CATHODE in model-agnostic searches without the need of manual selection of uncorrelated features. Additionally, it does not introduce significant sculpting, which allows for the use of standard signal extraction techniques in an actual search. Due to these advantages, CATHODE was applied to a first-of-its-kind search for new physics resonances by the CMS collaboration [165], which is discussed in the following sections.

7.2 Analysis Strategy

The work presented in this and the following sections of this chapter are mainly based on the analysis EXO-22-026 [165], a recent effort of the CMS collaboration. The main collaborators of this analysis are Thea Aarrestad, Oz Amram, Aritra Bal, Samuel Bright-Thonney, Nadya Chernyavskaya, Phil Harris, Gregor Kasieczka, Benedikt Maier, Petar Maksimovic, Patrick McCormack, Louis Moureaux, Jennifer Ngadiuba, Sang Eon Park, Maurizio Pierini, David Shih, Manuel Sommerhalder, Kinga Wozniak and Irene Zoi. The application of CATHODE specifically was done by collaborators from the University of Hamburg, namely Gregor Kasieczka, Louis Moureaux, Manuel Sommerhalder and myself. Several figures and written content closely resemble or match the information contained in the collaboratively developed documentation of this analysis. My contributions to this analysis are:

- Initial studies for the application of ANODE algorithm to analysis, including studies of density estimator performance (background estimation and interpolation), hyperparameter optimization, testing on preliminary Monte Carlo data set (background-only and for various signals) and studies of background sculpting.
- Significant contributions to the implementation, testing and continuous development of the "bump hunting" framework, in particular testing various parametric fit functions and assessing the quality of the background estimation as well as signal extraction. Implementation of the

template fit using a double crystal ball function for signal templates as well as the interpolation of parameters.

- Studies regarding possible scultping of the signal shape for the CATHODE method.
- Extensive studies regarding significance/p-value scans across mass points, both in the background-only case and the case including signal. Investigation and optimization of results for Monte Carlo samples in analysis region and data in the validation region. Detailed studies about possible fit bias considering the distribution of p-values for repeated experiments and for different selection efficiencies.
- Preparation and provisioning of all used background Monte Carlo and data samples using CERN infrastructure, as well as production of the respective data/MC comparison plots.
- Contribution to the analysis framework used for the application of CATHODE. In particular, implementing the normalizing flow for CATHODE based on the `pyro` [178] software package, as well as several studies regarding its hyperparameters, training stability and performance. Implementation of the sample preprocessing for training, including k-fold cross-validation.
- Several studies regarding feature selection for broad sensitivity across all investigated signal models for CATHODE.
- Implementation and generation of analysis plots for assessment of background fit on selected events in each SR as well as ΔNLL plots.

As discussed in chapter 6, model-agnostic searches are a promising way forward to search for BSM physics phenomena at the Large Hadron Collider and beyond. In an entirely data-driven way, large regions of phase space can be scanned for any anomaly in a single analysis. While model-agnostic search efforts have been conducted by many high energy physics experiments, most of them were based on identifying deviations between measured data and Monte Carlo simulation, introducing dependence on the background model. However, at the time of writing, two analyses by the ATLAS collaboration exist, using state-of-the-art unsupervised machine learning techniques for model-agnostic anomaly detection [162, 163].

The analysis described in this chapter is the first such effort in the CMS collaboration and uses a unique approach, testing a large variety of different anomaly detection methods for many signal models. Using not only a single, but several models significantly enhances any model-agnostic search, since each method has different strengths and weaknesses and it cannot be known which method is optimal to find an entirely unknown anomaly a priori. Additionally, testing a variety of signal models – many of which have not been targeted with a dedicated search – allows to acquire exclusion limits for all of them in one analysis. It also yields important insights into which models perform better in what kind of new physics scenarios.

Being a first-of-its-kind analysis, it does therefore not only mark a significant milestone for the CMS collaboration, but also for high energy physics in general.

When searching for physics phenomena beyond the Standard Model, the majority of current search efforts use specific signal models that target one or more aspects of the experimental or theoretical motivations for BSM physics. In these analyses, the search strategy is developed using a combination

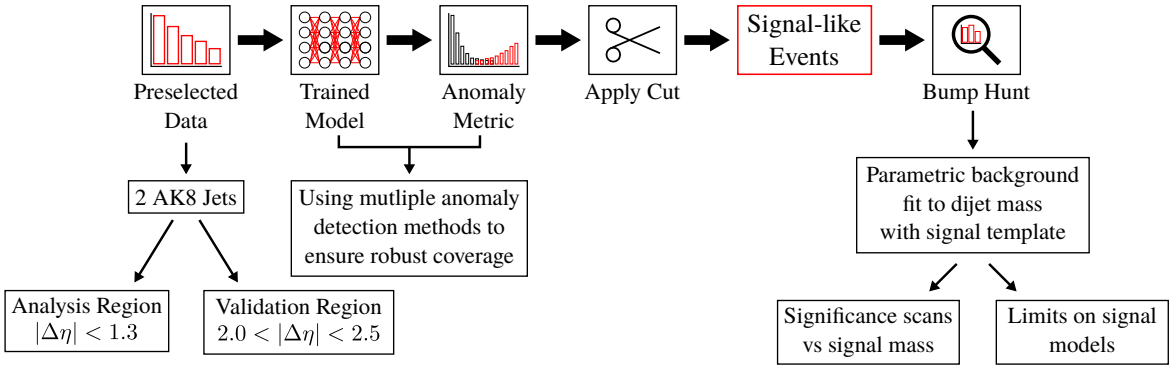


Figure 7.7: Overview of analysis strategy depicted as a flow chart of individual steps.

of signal events generated with Monte Carlo (MC) simulations and SM background MC events. Once the strategy is successfully applied and validated on MC, it is ultimately applied to data. Due to their reliance on MC simulations, these searches are strongly dependent on both the signal and the SM background models. Since the discovery of the Higgs boson by the CMS and ATLAS collaborations in 2012, many of BSM searches have been conducted at the LHC physics programme and so far, no evidence for new physics was found. Therefore, it is likely that there exists a BSM scenario in nature which is not amongst the typically tested models. Additionally, there are vast regions of phase space at the LHC as well as various signal models that are yet unexplored [44,179]. As discussed in section 6.5, efforts to employ model-agnostic searches were done by many High Energy Physics collaborations, such as D0 [155–157], H1 [158, 159], ALEPH [180], CDF [160, 161], CMS [4, 5] and ATLAS [6–8]. These searches look for deviations between data and SM simulations in many "bins" of final states. This approach is signal-model independent, but is entirely dependent on the correctness of the simulated SM background model. Additionally, looking for deviations in such a large number of bins introduces a significant look-elsewhere-effect, which considerably reduces the sensitivity and devising systematic uncertainties for several hundred final states and multiple kinematic distributions [3] often constitutes a challenge.

In the analysis described in this work, state-of-the-art machine learning methods are used for conducting a purely data-driven search that is agnostic to specific signal models. The target is to detect new resonances in the dijet final states using several such methods, which have been previously applied to DELPHES simulated data by the same collaborators in [150, 172, 175, 176]. The search methods employed are based on either of three paradigms: weak supervision, out-of-distribution detection or semi-supervised learning, and therefore represent different conceptual approaches to anomaly detection. As mentioned previously, using different methods instead of a single one ensures sensitivity for a broader range of possible signal anomalies.

Similar to the LHCO R&D data set described in subsection 6.4.1, the target of this analysis is the dijet final state of a narrow resonance A with mass $O(\text{TeV})$ decaying to particles B and C that can be SM or BSM particles that decay hadronically. The mass of B and C are chosen to be much smaller than m_A, such that their decay products can be merged in a single, large-radius jet. In this scenario, jet substructure variables can be used to look for anomalous events and reduce the background significantly.

The individual steps of this strategy are shown in the diagram in Figure 7.7. First, a common preselection is applied to the data and the two most massive large-radius jets are considered for fur-

ther analysis. Addtionally, the data is split into an analysis region and a validation region based on kinematic criteria. Data in the validation region is used to assess whether the employed machine learning methods are well calibrated and do not learn spurious information from data. It should also be noted that, even though the search itself is data-driven, Monte Carlo simulations are still used for initial validation studies as well as setting limits for specific signal models. After the preselection is applied, the respective machine learning methods are trained on data and a cut on the resulting anomaly metric is applied based on a pre-defined working point. This results in the most signal-like events being selected and considered for the final step, which is the bump hunt that was described in section 6.2. In this analysis, a parametric background function is chosen for estimating the background, while the signal is modelled using a generic double crystal ball template. By conducting a combined background and template fit, a p-value is extracted at several mass points in the invariant dijet mass and limits are set on the benchmark signal models. The p-values for each of the methods as well as the achieved limits for all the signal models are therefore the main outputs of this analysis.

7.3 Machine Learning Methods

A central part of the strategy of this analysis is that several machine learning-based methods are employed in order to have a high sensitivity for a large family of different potential anomalies. While a detailed description of all these methods and their results is beyond the scope of this work, a general summary is provided in this section. Furthermore, this work specifically focuses on the results of applying the CATHODE method to CMS data, which will be discussed thoroughly in the following sections, while comparisons with other analysis methods are also provided for some key studies.

All employed methods belong to one of three overarching anomaly detection principles:

- **Weak supervision:** In weak supervision, instead of using the actual signal/background information, noisy labels are used that are derived from the data directly. Therefore, a weak classifier learns to distinguish two *mixtures* of samples with different signal fractions. The topic of weak supervision for model-agnostic anomaly detection is also discussed extensively in chapter 6.

- **Out-of-distribution detection:** These models learn to encode the background distribution and use the likelihood of an event under this learned distribution as a tagger. Events that have a low background likelihood are therefore tagged as anomalous. Probabilistic generative models such as variational autoencoders [181] or normalizing flows (see section 5.4) are specific methods that are frequently used for this task.

- **Semi-supervised learning:** These methods are in between fully supervised and unsupervised learning. They learn to encode a complex space containing both the background distribution as well as one or more signal distributions. To do this, MC simulations of both signal and background models are typically used. The signal models are chosen to cover a broad range of different BSM characteristics that are combined into a single probability density. This results in a model which generalizes the main features of the BSM theories and thus is able to find signal models it was not directly trained on. In practice, probabilistic generative models are again used to encode background and individual signal model likelihoods.

A central point in model-agnostic searches is that an optimal algorithm cannot be defined, since the characteristics of the potential signal and the sensitivity of any method to detect it are not known. The use of anomaly detection paradigms that are conceptually different ensures that a set of complementary methods is used that yields a broader coverage of possible signal models compared to using only a single method.

In total, six methods are employed in this analysis, four of which are weakly supervised and one for each of out-of-distributions detection and semi-supervised learning. Each analysis method used is discussed in the following sections.

7.3.1 Weakly Supervised Methods

All of the weakly supervised methods employed in this search assume a narrow, localized resonance and make use of a signal region (SR) and sidebands (SB) in the dijet invariant mass, m_{jj}, to obtain signal-enriched and signal-depleted samples for classification (see chapter 6). In particular, if a signal resonance is present in the SR, the weak classifier will distinguish between an SR sample with a high signal fraction and the QCD background sample from the sideband. Conversely, if no signal is present, the classifier tries to distinguish QCD events from other QCD events and the tagging performance is close to random. To define these regions, a specific mass hypothesis must be chosen. In order to not restrict the analysis to a single SR in the large invariant mass phase space, 8 different m_{jj} windows are chosen and the analysis procedure is repeated for each of them. The width of the windows is chosen to be larger than the mass resolution in order to make sure that the majority of the resonance is contained within them. A second set of bins is defined that is shifted by approximately half the bin size with respect to the original set in order to mitigate cases where the resonance is close to the boundaries of the SR window. The two sets of windows are being referred to as mass windows A and B respectively and are summarized in Table 7.3. The table shows the mass ranges of each SR bin as well as the number of data events contained within them. The scanned mass points refer to the values where a combined fit of a signal resonance shape and a parametric background function is used to extract the corresponding p-value, which is further discussed in a later section. Bins A0, A7 as well as B0 and B7 are not used as signal regions for weakly supervised analysis, since the respective methods require both an upper and lower sideband. Therefore, 12 signal regions are studied in total. While the settings regarding the SR windows is kept the same for all anomaly detection methods based on weak supervision, several aspects vary. Therefore, each of the methods is shortly described in the following.

CWoLa Hunting

CWoLa is a weakly supervised method that was already described extensively in section 6.3. It can be seen as the most foundational weak supervision method, since events from the SR are directly trained against events from the SB region to learn the likelihood ratio $R = p_{\text{data}}(\mathbf{x})/p_{\text{bg.}}(\mathbf{x})$. For CWoLa, the SB region is defined by the bins adjacent to the considered SR. Due to the exponentially falling shape of the invariant dijet mass distribution, there are significantly fewer events in the upper SB compared to the lower SB, hence upper SB events are reweighted in the training to match the total weight of the lower SB events. In this search, CWoLa is applied using two per-jet classifiers, one for first jet and one for the second jet in terms of jet mass. This allows an additional reweighting of

Bin name	**Range** [GeV]	**Eff. cut**	**Scanned masses** [GeV]	**Num. data events**
A0	1350 – 1650	-	-	5.6M
A1	1650 – 2017	1 %	1700, 1800, 1900	3.8M
A2	2017 – 2465	1 %	2200, 2300	1.2M
A3	2465 – 3013	1 %	2600, 2700, 2800	340k
A4	3013 – 3682	1 %	3200, 3300, 3400, 3500	90k
A5	3682 – 4500	1 %	3900, 4000, 4100, 4200	20k
A6	4500 – 5500	1 %	4800, 4900, 5000, 5100, 5200	3.5k
A7	5500 – 8000	-	-	479
B0	1492 – 1824	-	-	6.6M
B1	1824 – 2230	1 %	2000, 2100	2.1M
B2	2230 – 2725	1 %	2400, 2500	630k
B3	2725 – 3331	1 %	2900, 3000, 3100	170k
B4	3331 – 4071	1 %	3600, 3700, 3800	42k
B5	4071 – 4975	1 %	4300, 4400, 4500, 4600, 4700	8.7k
B6	4975 – 6081	1 %	5300, 5400, 5500, 5600, 5700, 5800	1.3k
B7	6081 – 8000	-	-	144

Table 7.3: Signal region windows used by the weakly supervised methods for analysis, including ranges, selection efficiencies (used by CATHODE-based methods), mass points scanned and number of events.

jets in the SB region to match the p_T distribution of the jets in the SR, which reduces the correlation between the resulting anomaly score and m_{jj}. The input features used are the soft drop mass of the jet m_{SD} [182], the n-subjettiness ratios τ_{21}, τ_{32} and τ_{43}, the number of PF candidates in the jet, n_{PF}, the lepton subjet fraction LSF_3 (see subsection 7.6.4) and the maximum b-tagging score of the two leading subjets of the large-radius jet, based on the DeepCSV algorithm. The final anomaly score of the two classifiers is obtained by first computing the percentiles of the anomaly score distributions for each classifier separately. Then, the maximum of these anomaly-percentile scores are taken as the final event-level score.

Tag'N'Train (TNT)

Tag'N'Train (TNT) is a method based on CWOLA hunting [174], that additionally uses low-level information to enhance anomalous events in the SR sample. In particular, an autoencoder based on an image representation of the jet is trained [167, 168], representing the energy density in the (η, ϕ) plane. The use of jet images can enhance the anomaly detection scheme, since the images use the low-level PF candidate information which complements the high-level information used in the weak classification.

The image-based autoencoder (AE) takes the images and encodes them using a six-dimensional latent space. The difference between the decoded images and the input images (also referred to as reconstruction loss) is taken as an anomaly score. This score is computed for each of the two jets in the event separately.

The signal-enriched samples are now obtained not only using the SR thresholds in m_{jj}, but also using the AE score. First, jets in each event are randomly assigned into groups J1 and J2, such that the "first" and "second" jet assignment is no longer mass-ordered, but randomly ordered. Due to this

randomization, all jets have the same underlying distribution and a single AE is trained on the entirety of the jets for events from the SB region. Then, the jets in all events both in the SR and SB regions are evaluated and the reconstruction loss is computed for each jet in the event individually. The signal-enriched sample for the J2 group is then created by taking the J2 jet of SR events where the J1 jet is in the top 20% in terms of AE score. The background-like sample on the other hand is created for the J2 group by taking all J2 jets of events in the SB region as well as all J2 jets of events in the SR where the J1 jet is in the lower 40% in terms of AE score. That is, the SR and SB region definitions in m_{jj} as well as the J1 reconstruction loss of the AE define the signal/background-enriched samples for the jets in group J2. The same procedure is applied vice-versa, with jets in group J2 defining signal and background-enriched samples of jets in group J1. Thus, the selection of signal-enriched samples is based on the assumption that both jets are anomalous.

Finally, the two background-enriched samples as well as the two singal-enriched samples are merged and a weakly supervised classifier is trained to distinguish the two. The event-level anomaly scores are then obtained by multiplying the final classifier scores of the two jets in each event. A key benefit of TNT is that the AE-based tagging can significantly increase the signal fraction in the signal-enriched sample for the weak classification task. The major caveat of this method is that since the anomalous behaviour of one jet in the event is used to define the other jet as also being anomalous, it is only suited for cases where both jets in the event are anomalous. For signal models where only one of the jets contains anomalous substructure while the other is a quark/gluon jet, the AE assignment will lead to non-anomalous jets being contained in the signal-enriched sample, which will reduce the sensitivity significantly.

The input features for the final weakly supervised classifier are the same as in the previously discussed CWoLa approach. The images for the AE input are based on the p_T, η and ϕ values of the PF candidates and the same processing steps as in [183] are applied.

CATHODE and CATHODE-b

Another weakly supervised method employed in this analysis is CATHODE and its application to experimental data will be discussed in detail. CATHODE uses density estimation techniques based on normalizing flows to first learn the background density $p_{\text{bg.}}(\mathbf{x})$ from the SB region. Then it interpolates this density into the SR and trains a weak classifier between samples from the interpolated background density and data samples in the SR. Therefore, the weak classifier approximates the most powerful classifier possible for any weakly supervised search, given by $R = p_{\text{data}}(\mathbf{x})/p_{\text{bg.}}(\mathbf{x})$.

In this analysis, the input features used for CATHODE are the most massive jet in the event, m_{j_1}, the mass difference between the most and second most massive jets $\Delta m_j = m_{j_1} - m_{j_2}$, as well as the subjettiness ratios τ_{41,j_1} and τ_{41,j_2} of the two jets. During experimentation it was also observed that adding the b-tagging score significantly increased the performance on signal models containing b jets, while it acted as an additional noise feature for signal models without one, causing a decrease in performance. Therefore, two versions of CATHODE are used in this analysis, one with the mentioned input features referred to as CATHODE and one with the DeepCSV b-tagging scores DeepB_{j_1} and DeepB_{j_2} of the two jets added, referred to as CATHODE-b.

The CATHODE method has been described in section 7.1. The training parameters and study results for CATHODE and CATHODE-b on experimental data will be discussed in the following sections.

7.3.2 Out-of-Distribution Detection

Variational Autoencoder with quantile regression (VAE-QR)

For the out-of-distribution detection method employed in this analysis, a variational autoencoder (VAE) is used. Similar to an autoencoder, a VAE learns to encode an higher-dimensional input space in a latent space with significantly fewer dimensions. A decoder network then learns to reconstruct the original input using the information encoded in the latent space. In order train a neural network to do this, the loss is often defined as the mean squared difference between the reconstructed output and the original input. So far, the procedure corresponds to that of a standard autoencoder network. However, the *variational* autoencoder is different in terms of the latent space. In particular, the encoder learns to compress the input space into a *distribution* over the latent space, typically modelled by a standard normal distribution. Therefore, the loss of a VAE has two components: the reconstruction loss and a loss term that maximizes the likelihood of the encoded distribution under a standard Gaussian. For the latter, the Kullback-Leibler divergence (KL-divergence) between the encoded distribution and the standard normal distribution is typically used.

As inputs, the VAE uses the x, y and z components of the momentum p of each of the PF candidates of a jet. Specifically, the 100 highest-p_{T} candidates are used for the training. The VAE is trained in a $\Delta\eta$ control region where no significant signal contribution is expected. After the training, it is evaluated on events in the analysis region and the minimum reconstruction loss of the two jets in each event is used as an anomaly score, meaning that the scenario of both jets being anomalous is targeted. Since it has been observed that there are significant correlations between m_{jj} and the resulting VAE score, a quantile regression (QR) is performed. The QR model learns a cut as a function of the invariant mass corresponding to a fixed selection efficiency in the signal region. For the model-independent search, a cut corresponding to the 10 % most anomalous data events is used.

7.3.3 Semi-Supervised Learning

QUAK

The semi-supervised method used in this analysis is referred to as *QUAK* [176]. In this method, density estimators based on normalizing flows are used to encode a signal prior that is based on Monte Carlo samples of several considered signal models. Additionally, a similar prior is learned for the background, again using a density estimator trained on simulated QCD samples. The input features used are $\rho_{j_1} = m_{j_1}/p_{\mathrm{T},j_1}$ and ρ_{j_2}, as well as the subjettiness ratios τ_{21}, τ_{32}, τ_{43} and a modified n-subjettiness metric $\tau_s = \sqrt{\tau_{21}}/\tau_1$. Furthermore, the DeepB score and the number of PF constituents are used.

To be sensitive to many hypothetical signal models, six signal flows are trained on combinations of different signal masses that are grouped with respect to the B and C particle masses. A signed L5 norm is then used as an anomaly score which is computed from the six flow outputs. Additionally, a single flow is trained for the background events. Events that are likely anomalous are expected to be located in the region of a high signal score and a low background score. To select events, a template fit is conducted, which is implemented as a binned 2D histogram of the QUAK space, using pre-defined signal regions and sidebands. The events located within the bins with an excess of events in the signal region are then selected and a bump hunt in m_{jj} is performed on them.

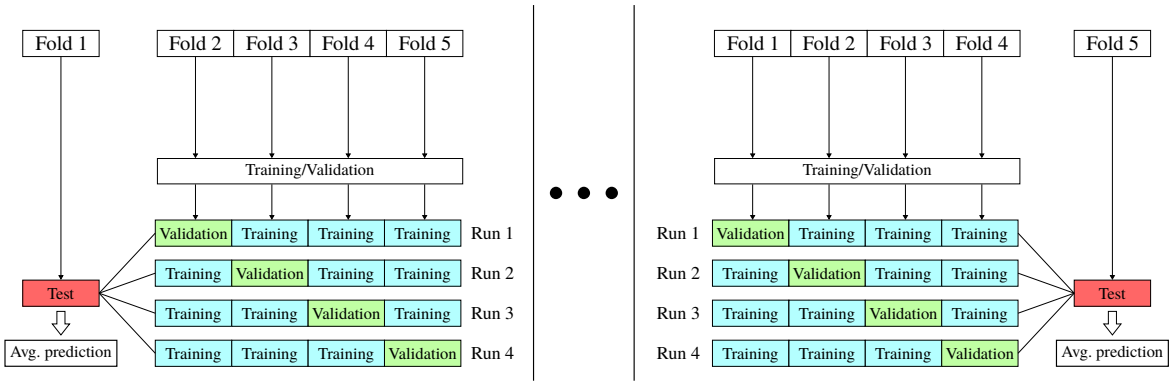


Figure 7.8: Diagram showing the *k*-*l*-folding procedure used for CATHODE(-b) and other weakly supervised methods

QUAK is used in two modes in this analysis: A "model-agnostic" mode, where all the benchmark signal samples are used for the prior and a "model-specific" mode, where the prior is purely based on the targeted signal. The former is used to generally look for anomalies in data, whereas the latter is used for the model-specific limit setting.

7.4 Cross-Validation & Training Procedure

7.4.1 k-l-folding

For the weakly supervised methods, a two-stage *k*-folding strategy is employed. The first stage involves a standard *k*-folding technique that allows for separate, independent test folds in each retraining. By combining the results of these folds, it is possible to obtain anomaly scores for the entire data. Due to the noisy nature of weak supervision and the rare occurrence of anomalies, small changes in the training can have a large impact in the final results. In particular, the outcome of weakly supervised methods can be impacted considerably by the stochasticity of the model initialization and training processes. In order to control this, *l*-folds are defined in the second stage of the folding strategy for the CWOLA Hunting, TNT and CATHODE(-b) methods. The cross-validation is almost the same between those methods and differs only in nuances. In this section, the procedure used for CATHODE(-b) will be described.

To better illustrate the procedure, it is depicted as a diagram in Figure 7.8. The data samples are randomly split into 5 *k*-folds. Then, one of them is assigned as test fold and the remaining 4 are used for training and validation. In particular, one of the 4 remaining folds (which will be referred to as *l*-folds) is used for validation, while the other 3 *l*-folds are used for training. The *l*-fold training is repeated four times, such that all possible *l*-fold assignments are realized once. This results in four models, each of which is evaluated on the test *k*-fold that was assigned in the beginning. The entire procedure is then repeated for each possible value of *k*, which leads to a total of $5 \times 4 = 20$ models.

This *k*-*l*-folding procedure is applied both to the density estimator and weak classification parts of the CATHODE(-b) algorithm. The density estimator models of each *k*-*l*-fold setting produce samples for the respective classifier models to train on. The full evaluation of the weak classifiers is then based on an ensemble of the four models trained for each *k*-fold, using the average prediction on each test set event as the final score. Since each fold is assigned as test fold exactly once, the average predictions

Hyperparameter	Value
Number of MADE blocks	15
Number of hidden layers	1
Number of nodes per hidden layer	128
Activation function	ReLU
Batch normalization	After each MADE block
Batch normalization momentum	0.1
Log(loss) clipping	Between -3 and 3
Optimizer	Adam [108]
Learning rate	10^{-4}
Weight decay	10^{-6}
Number of epochs	100
Batch size	256

Table 7.4: Hyperparameter settings used for the CATHODE(-b) density estimator models.

of all test folds can be combined to obtain anomaly scores for the entirety of the data set. These scores are then used to tag anomalies based on a previously defined selection efficiency.

7.4.2 Training Procedure for CATHODE(-b)

Independent of what data set CATHODE(-b) is trained on, the same training procedure is used throughout this analysis. Starting from the data after the common preselections have been applied, it gets split into SR and SB region samples. Each of these is then again split into five distinct k-folds. In the next step, the density estimator models are trained using the full k-l-folding procedure described in the previous section. For each of these trainings, the data is preprocessed based on the values of the samples in the training set: First, the input features are transformed to be forced in a range between 0 and 1 by subtracting the minimum and dividing by the difference between maximum and minimum values. Second, a logit transformation is applied to the features. This is done in order to move the data away from the edges of the phase space and mitigate poor density estimation of distributions that are peaked at either boundary. Finally, the features are standardized to have a mean of zero and a standard deviation of one by subtracting the mean and dividing by the standard deviation. The same transformations are then also applied to the validation set and the density estimators are trained on the samples from the SB region, using the invariant dijet mass, m_{jj}, as a conditional. The models that are used for density estimation are based on normalizing flows, implemented using the `pyro` [178] software package. In particular, MAFs are used (see subsection 5.4.2 for details) and the specific hyperparameters are described in Table 7.4.

Once the density estimators have been trained, they are used to generate the background template samples for the weak classification task. In order to increase the stability of the classifier trainings, it has been shown to be beneficial to produce a template sample based on an ensemble of multiple density estimator models. Therefore, the model states at the five lowest validation loss epochs are used for each model. The number of template events to be sampled is set to be four times the number

Hyperparameter	Value
Number of hidden layers	3
Number of nodes per hidden layer	64
Optimizer	Adam
Learning rate	10^{-3}
Number of epochs	100
Batch size	128

Table 7.5: Hyperparameters used for the weak classifier models in Cathode(-b)

of events in the SR training samples. This "oversampling" was also used in the original Cathode work [150] and has been shown to improve performance compared to a template sample of the same size. The ensembling is realized by simply generating one fifth of the needed template samples with each model.

In order to interpolate the learned density from the SB region into the SR, the flow models need to be conditioned on the SR m_{jj} values. To make sure that no exact values are re-used during oversampling, a kernel density estimator (KDE) is trained on the m_{jj} of SR events in the training set. Then, a sample of the needed size is drawn from the KDE to use as an input for the conditional feature of the flows. Since the density estimators are trained on the preprocessed features, the template samples drawn from them will also be in this space. Once the interpolated template samples have been obtained, the preprocessing is reverted and the template samples are merged with the training set samples from the SR. As a preprocessing for the classifier training, the features are again normalized by subtracting the mean of the training set events and dividing by the standard deviation. The same preprocessing is also applied to the validation set.

Finally, the weak classifiers are trained, distinguishing actual SR events (which are given the label 1) and the background template events (which are given the label 0). The classifiers used are implemented as fully connected neural networks using the `PyTorch` [109] software package. The corresponding hyperparameters are described in Table 7.5. Once the weak classifiers have been trained, they can be evaluated on their respective test k-fold. Again to reduce the impact of the stochasticity on the final result, each classifier model is defined as an ensemble based on the 10 lowest validation loss epochs. After applying the same normalization preprocessing step, the test set both in the SR and the SB region is evaluated using each of the 10 model states and the predictions are averaged to obtain an ensemble prediction. This is done for each l-fold, such that four such ensemble predictions exist for each test k-fold. For the final prediction, the four l-fold model predictions are again averaged. Finally, the predictions of all k-folds are merged, yielding anomaly scores for all events in the data set.

7.5 Data and Simulated Samples

7.5.1 Data Samples

The data samples used in this analysis are based on proton-proton interactions collected with the CMS detector at a center-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$ during the data-taking years 2016, 2017

Year	Golden JSON file name	Integrated luminosity [fb^{-1}]
2016	Cert_271036-284044_13TeV_Legacy2016 Collisions16_JSON.txt	36.4
2017	Cert_294927-306462_13TeV_UL2017_Collisions17 GoldenJSON.txt	41.5
2018	Cert_314472-325175_13TeV_Legacy2018 Collisions18_JSON.txt	59.8
Total		137.6

Table 7.6: Golden JSON files used in analysis for each year as well as the respective integrated luminosity.

and 2018 (full Run 2). These samples correspond to an integrated luminosity of 137.6 fb^{-1}, which is measured based on the "golden JSON" files listed in Table 7.6 for the mentioned years. The JSON files list the sections in luminosity that passed the CMS "golden" certification, meaning that all of the sub-detectors were flagged to operate within normal parameters and thus the collected data can be used in any kind of physics analysis [184, 185]. The data set used is based on the JetHT physics data set, where data was collected with a combination of `PFHT` and `PFJet` hadronic triggers applied. These trigger on the sum of the transverse momenta of the PF candidates (see subsection 4.3.2) in the event, denoted as H_T and on the tranverse momenta, p_T, of the jets clustered from these candidates, respectively. The data was preprocessed starting from samples of the ultra legacy (UL) campaign in MiniAOD format, from which NanoAOD files were produced using the JetMET `PFNano` producer. The full list of used samples can be seen in Appendix B. These files contain the same information as in the official CMS campaign, but additionally include information of the individual jet constituents that are used as input to some of the above-mentioned machine learning methods. The production of samples using `PFNano` started from MiniAODv1 for the 2016 data set, whereas for the 2017 and 2018 data sets MiniAODv2 was used. For all eras, the processing is based on the CMS software version `CMSSW_10_6_20` using the `106X_dataRun2_v32` global tag.

7.5.2 Simulated Samples

Background

For the generic hadronic dijet resonance A→BC, the main background process consists of QCD multijet events. Minor backgrounds include W and Z bosons produced in association with additional jets as well as top-antitop quark pairs and single top quarks.

The QCD background events are simulated and showered using PYTHIA8 [142, 186]. Top-antitop and single top quark backgrounds are modelled with POWHEG [187, 188] at next-to-leading-order (NLO) and also showered using PYTHIA. The vector boson production in association with jets is generated at leading-order (LO) using MADGRAPH5_aMC@-NLO [189, 190] matched with PYTHIA. The generated samples are then processed by a simulation of the CMS detector based on GEANT4 [191]. Additional proton-proton collisions in the same or adjacent bunch crossings (pileup events) are simulated by adding more inelastic events that are generated using PYTHIA and superimposed on hard scattering events. The entirety of the background samples from the UL campaign used in this analysis is summarized in Appendix B for the three data-taking years, including the respective cross sections, which are based on [192] for the p_T-binned QCD and V+jets samples. For single top quarks produced in the tW-channel, cross-sections are taken from [193] and for the t-channel from the cross-

Process	$\mathbf{m}_\mathrm{A}$ [TeV]	$\mathbf{m}_\mathrm{B}$ [GeV]	**n subjets B**	$\mathbf{m}_\mathrm{C}$ [GeV]	**n subjets C**
$Q^* \to qW'$	[2, 3, 5]	≈ 0	1	[25, 80, 170, 400]	2
$X \to YY'$	[2, 3, 5]	[25, 80, 170, 400]	2	[25, 80, 170, 400]	2
$W_{kk} \to RW \to WWW$	[2, 3, 5]	[170, 400]	4	m_W	2
$W' \to tB' \to tbZ$	[2, 3, 5]	m_t	3	[25, 80, 170, 400]	3
$Z' \to T'T' \to tZtZ$	[2, 3, 5]	[400]	5	[400]	5
$Y \to HH \to tttt$	[2, 3, 5]	[400]	6	[400]	6

Table 7.7: Specific signal models of generic decay A→BC used in this analysis, including particle masses and subjets expected within single large-radius jets.

section database XSDB. Similarly to the data file processing, MiniAOD files are initially used for the simulated samples from which NanoAOD samples are produced using `PFNano`, including the information of individual PF candidates. This processing is done using the CMS software release `CMSSW_10_6_20` and the global tags `106X_mcRun2_asymptotic_preVFP_v11` (2016 preVFP era), `106X_mcRun2_asympto-tic_v17` (2016 postVFP era), `106X_mc2017_realistic_v9` (2017 era) and `106X_upgrade2018_realistic_L1v1` (2018 era). For all of these eras, the initial data format was MiniAODv2.

In most analyses, Monte Carlo samples are weighted to match the respective cross section of the process. In this search, however, using MC weights in the trainings of complex ML methods has proven impractical. Therefore, an unweighted sample is constructed from event samples of each background process. The number of events sampled corresponds to the expected number corresponding to the cross section. This data set is constructed to simulate the Poisson statistics of a real data set and is therefore referred to as "mockup MC data set". It is based on the lowest effective luminosity of any process in the MC UL data set for each era, $\mathscr{L}_{\mathrm{min.,era}}$, which is given by the lowest p_T bin of the QCD background process. The corresponding luminosities are $6.8\,\mathrm{fb}^{-1}$ for the 2016 pre-VFP era, $5.6\,\mathrm{fb}^{-1}$ for the 2016 post-VFP era, $7.1\,\mathrm{fb}^{-1}$ for the 2017 and $7.4\,\mathrm{fb}^{-1}$ for the 2018 era. The number of events sampled for each process is then given by $N_\mathrm{sampled} = \mathscr{L}_{\mathrm{min.,era}} \cdot \epsilon_\mathrm{pre.} \cdot \sigma$, where $\epsilon_\mathrm{pre.}$ is the efficiency of applied preselections and σ is the cross section of the physics process. Such a mockup data set is then constructed for each of the four data taking periods, resulting in a data set with a total effective luminosity of $26.9\,\mathrm{fb}^{-1}$ for all eras combined.

Signal

For the signal samples of the new physics particles A decaying to particles B and C, a variety of different mass combinations for the three particles and number of hadrons that particles B and C are decaying to is used. This is done to cover a large kinematic and jet substructure space in the search. In terms of jet substructure, decays from 2 to 6 final-state quarks are considered for B and C, as well as single quark or gluon decays. The specific models that are considered, including the mass values scanned for each particle, are shown in Table 7.7. For particle A, mass values between 2 and 5 TeV are scanned, while for the daughter paricles B and C, values between 25 and 400 GeV are used. When one of the final state particle is a SM particle, only the mass of the new physics particle is scanned. Similar to the background samples, all signal samples are based on the ultra legacy (UL) campaign and are

produced for the same respective eras. The resonance is generated with a narrow width in all cases. The invariant dijet mass spectra of all investigated signals can be seen in Appendix C. Any resonant signal model and subjet scenario can be investigated using all methods employed in this analysis since they are generic. However, the focus of this analysis are the shown models, which represent a wide range of possible mass and subjet realisations.

The simulated signal events are generated at LO in QCD with MadGraph5_aMC@NLO [189], using the NNPDF 3.1 parton distribution functions [194–196] and being interfaced to PYTHIA8 [142] with the CP5 tune [197] to simulate parton showers and hadronization. Similar as for the background samples, the starting point of the processing was the MiniAOD data format of the UL samples, which were enhanced with particle candidate information for the production of the extended NanoAOD samples. In particular, the initial data format was MiniAODv2 for all data-taking years. The production was performed with the CMS software release `CMSSW_10_6_20` and the following global tags:

- `106X_mcRun2_asymptotic_preVFP_v9` (2016 preVFP)
- `106X_mcRun2_asymptotic_v15` (2016 postVFP)
- `106X_mc2017_realistic_v8` (2017)
- `106X_upgrade2018_realistic_v15_L1v1` (2018).

7.6 Physics Objects and Event Selection

7.6.1 Noise Filters and Primary Vertex Selection

A key aspect for a thorough analysis of experimental data is to remove events that are misreconstructed or affected by instrumental defects. In particular for the data used in this analysis, the missing transverse energy, MET, can be mis-reconstructed due to anomalous signals in the hadronic calorimeter (HCAL) and anomalous energy deposits in the electromagnetic calorimeter (ECAL). Therefore, the MET physics object group (POG) recommends the use of several noise filters to be applied prior to analysis, which is also done in this work. Specifically, the following filters were applied for all eras, if not otherwise specified:

- goodVertices
- globalSuperTightHalo2016Filter
- HBHENoiseFilter
- HBHENoiseIsoFilter
- EcalDeadCellTriggerPrimitiveFilter
- BadPFMuonFilter
- BadPFMuonDzFilter
- eeBadScFilter
- CSCTightHaloFilter (only applied for 2016 era)
- ecalBadCalibFilter (not applied for 2016 era)

Criterion	Threshold
Neutral Hadron Fraction	< 0.90
Neutral EM Fraction	< 0.90
Number of Constituents	> 1
Charged Hadron Fraction	> 0
Charged Multiplicity	> 0

Table 7.8: Jet identification criteria applied to AK8 PUPPI jets with $|\eta| < 2.5$

Furthermore, another selection criterion is that in each event, at least one primary vertex must be reconstructed within a 24 cm window along the beam axis, having a transverse distance from the nominal pp interaction region of less that 2 cm [198]. If there are multiple vertices passing these requirements, the vertex with the highest total p_{T}^2 summed over all associated tracks is designated as the primary-event vertex.

One particular aspect is taken into account for the 2018 data taking era, where several HCAL module failures were negatively impacting the measurements of the jet energy in the affected region. Since this search is a generic anomaly search, it is important to remove anomalies that arise from such detector malfunctions. Therefore, a veto is applied for all events from this era for which one of the two highest p_{T} jets lies in the affected region ($-1.57 < \phi < -0.87$ and $-2.5 < \eta < -1.3$), which results in a marginal $\approx 1\,\%$ signal efficiency loss for the 2018 era.

7.6.2 Preselection

Events are reconstructed using the particle flow (PF) algorithm [199], combining information from various CMS detector parts to identify individual particle candidates. From these candidates, jets are clustered using the anti-k_{T} [200] jet clustering algorithm with a distance parameter of $R = 0.8$, also referred to as AK8 jets, as implemented in the FastJet software package [201]. To mitigate the impact of pileup on the anomaly search, especially to the jet substructure, the pileup per particle identification (PUPPI) is employed [202]. PUPPI combines information of the local shape of charged pileup, event pileup properties and tracking information to compute a weight describing the likelihood of a particle originating from a pileup interaction. Then, the four-momentum for each neutral and charged PF candidate is scaled with this weight. Specifically, charged candidates not originating in the primary vertex are assigned a weight of 0. Furthermore, several quality criteria for jet identification are defined by the JetMET POG for the UL samples used in this analysis. In particular, the tight jet identification requirements [203] are applied, which are summarized in Table 7.8. Jet corrections are applied for nonlinearities in p_{T} and rapidity, using the standard jet energy corrections at CMS, summarized in [204]. Since the latest recommended corrections are already implemented in the UL samples used in this work, no further corrections had to be applied in the analysis workflow.

After the jet reconstruction and application of the discussed corrections, events are selected based on kinematic criteria. In particular, at least two jets with a transverse momentum $p_{\mathrm{T}} > 300\,\mathrm{GeV}$ passing the tight jet ID criteria and $|\eta| < 2.5$ are required. The two jets with the highest p_{T} passing these criteria are selected as the candidates of the dijet system. The analysis region is defined by requiring a separation of $|\Delta\eta| < 1.3$ in order to further reduce the QCD multijet background. Additionally, an

invariant dijet mass of $m_{jj} > 1455\,\text{GeV}$ is required in order to be above the trigger efficiency plateau, where the applied triggers are $> 99\,\%$ efficient for all eras. The region of higher separation between the jets, $|\Delta\eta|$, is used to validate the analysis methods and will be described in more detail later. If the two jets to not pass the $|\Delta\eta|$ and m_{jj} criteria of the analysis or validation region, the event is discarded. In summary, the preselection criteria for the analysis region are as follows:

- Two jets:
  - Passing PF jet tight ID criteria
  - Jet $\eta < 2.5$
  - Jet $p_\text{T} > 300\,\text{GeV}$
- $|\Delta\eta| < 1.3$
- $m_{jj} > 1455\,\text{GeV}$

Data-taking era	H_T trigger	Jet p_T trigger
2016 runs B-G	HLT_PFHT800	HLT_PFJet450
2016 run H	HLT_PFHT900	HLT_PFJet450 HLT_AK8PFJet450
2017 runs B-F	HLT_PFHT1050	HLT_AK8PFJet500
2018 runs A-D	HLT_PFHT1050	HLT_AK8PFJet500

Table 7.9: Jet-based triggers used for online data collection during the data-taking periods considered in this analysis.

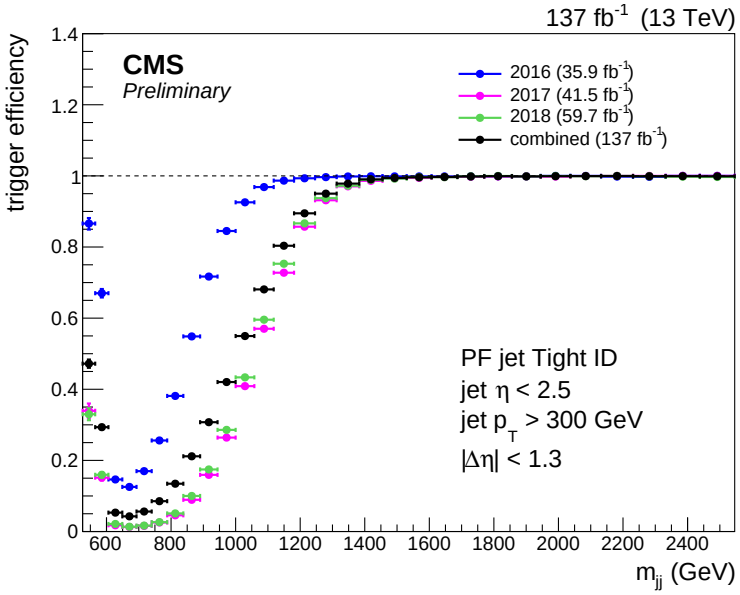


Figure 7.9: Trigger efficiency as a function of the dijet invariant mass for each year of data taking, as well as for all years combined.

7.6.3 Triggers

The event selection is based on single-jet-p_T- and H_T-based triggers, which are summarized in Table 7.9. Similar to other studies focusing on hadronic dijet resonances, the background estimation is done using a smooth parametric function fitted to the dijet invariant mass spectrum, as will be discussed in detail in the next section. In order to conduct such a smooth fit, significant trigger turn-ons, i.e. inefficiencies in the low m_{jj} region that significantly change the shape of the spectrum, should be mitigated. Therefore, the invariant mass threshold that is applied is chosen such that the logical OR of all the triggers is > 99 % efficient for each of the data-taking years. The specific threshold used is a result of a dedicated study, performed for each year separately. This was done using a data set disjoint from the one used in the analysis, namely the single muon data set. The reference trigger paths to be passed in this case are `HLT_IsoMu27` or `HLT_Mu50`. The trigger efficiency as a function of the dijet invariant mass can be seen in Figure 7.9. From this plot, it can be seen that for the 2016 eras, the trigger plateau is reached for invariant dijet mass values $m_{jj} > 1181$ GeV, while for 2017 and 2018 it is reached for $m_{jj} > 1455$ GeV. Since the latter value is the highest point in invariant mass at which the desired efficiency is achieved for any of the eras, it is chosen as a threshold for the analysis.

7.6.4 Jet Features

The various anomaly detection methods used in the analysis use different jet features as input. In particular, these features are:

- The softdrop mass of the jet.
- The 100 highest p_T 4-vectors of the particle flow constituents of the jet, as well as the number of jet constituents (which might be larger than 100, in which case the remaining constituents are dropped).
- The n-subjettiness features [147] τ_1 to τ_4.
- A b-tagging score, referred to as DeepB. This is defined as the maximum DeepCSV [205] b-tagging score of the two subjets of the large-radius AK8 jet. If only a single subjet exists within an AK8 jet, its b-tagging score is used. If no subjet exists, it is set to 0.
- LSF_3, the lepton subjet fraction of the jet, which describes the fraction of the total jet momentum carried by its highest-p_T lepton.

7.6.5 Data and MC Comparison

Since in this analysis MC simulations are purely used for validation purposes and the actual methods are directly applied to data, a dedicated MC-based analysis including precise data-to-MC comparisons and studies of systematic effects is not necessary. However, for the validation studies to produce valid results, a good agreement between the mockup MC data set and the measured data is still of high importance. Therefore, dedicated data-MC comparison studies were performed, the results of which can be seen in Figs. 7.10 to 7.14 below. The plots show several kinematics and jet substructure distributions after the application of the previously described preselections. The collected data shown is the sum of all data-taking years. Since the luminosity of the mockup MC data set is not the same as for the data, events are reweighted to match the expected luminosity of the corresponding data-taking period. Additionally, the distributions of the events from the simulated Q^* signal are also shown.

In general, a good agreement between data and the mockup MC data set can be observed. Most larger deviations are confined to the low-statistics regions in the tails of the distributions. For many variables, there is a constant $\approx 20\,\%$ difference that can be seen from the ratios shown in the lower panel of each plot. While the reason for this deviation is unknown, it is also not particularly concerning for the analysis, as it is simply a change of a constant factor and the overall shapes of the distributions show a good agreement.

7.7 Statistical Methods

7.7.1 Background Estimation and Signal Extraction

The central statistical method applied in this analysis is the "bump hunting" procedure, which was already outlined in section 6.2. The bump hunt consists of a combined fit of a parametric function that describes the exponentially falling background of the QCD multijet events and a narrow signal resonance template. For the weakly supervised methods in particular, the bump hunt is applied to the

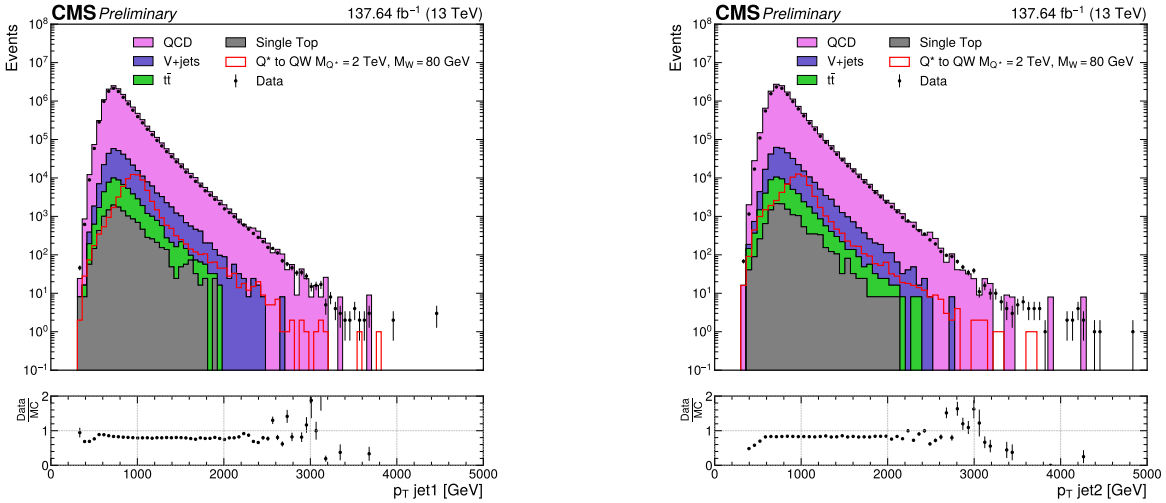


Figure 7.10: Transverse momentum p_T of the jets with the highest (left) and the second highest soft-drop mass (right).

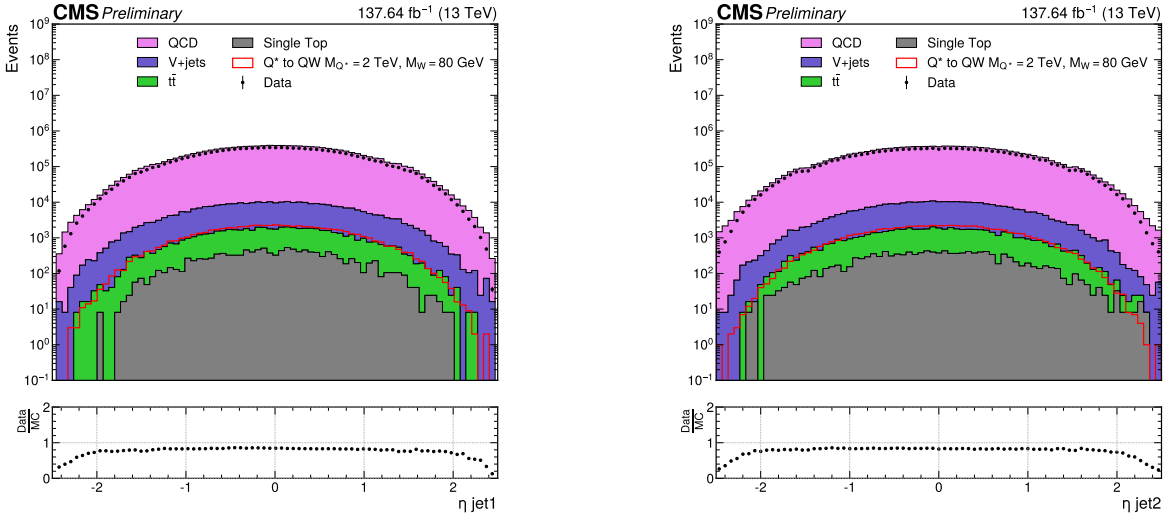


Figure 7.11: Pseudorapidity η of the jets with the highest (left) and the second highest softdrop mass (right).

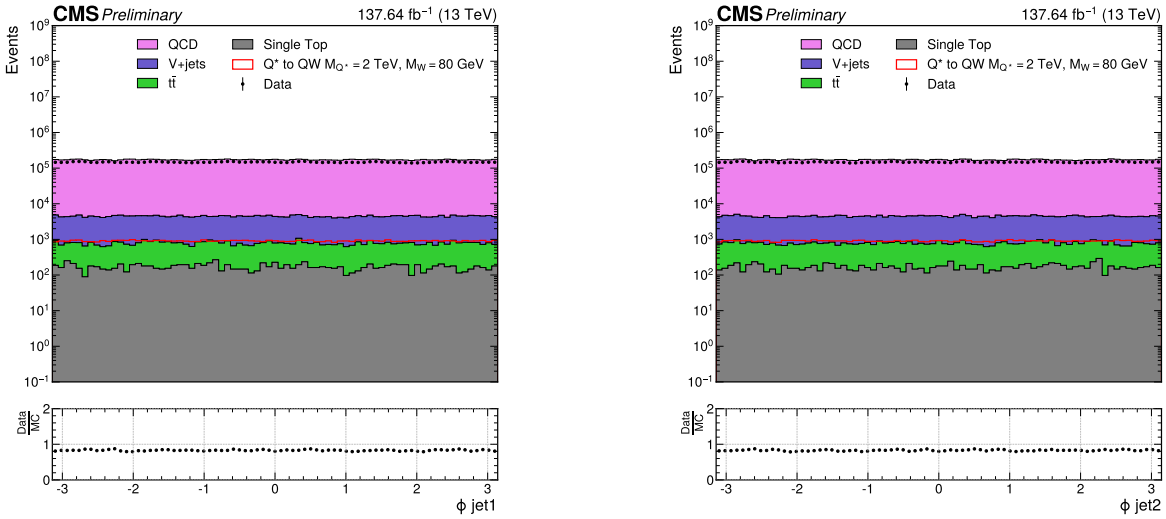


Figure 7.12: ϕ of the jets with the highest (left) and the second highest softdrop mass (right).

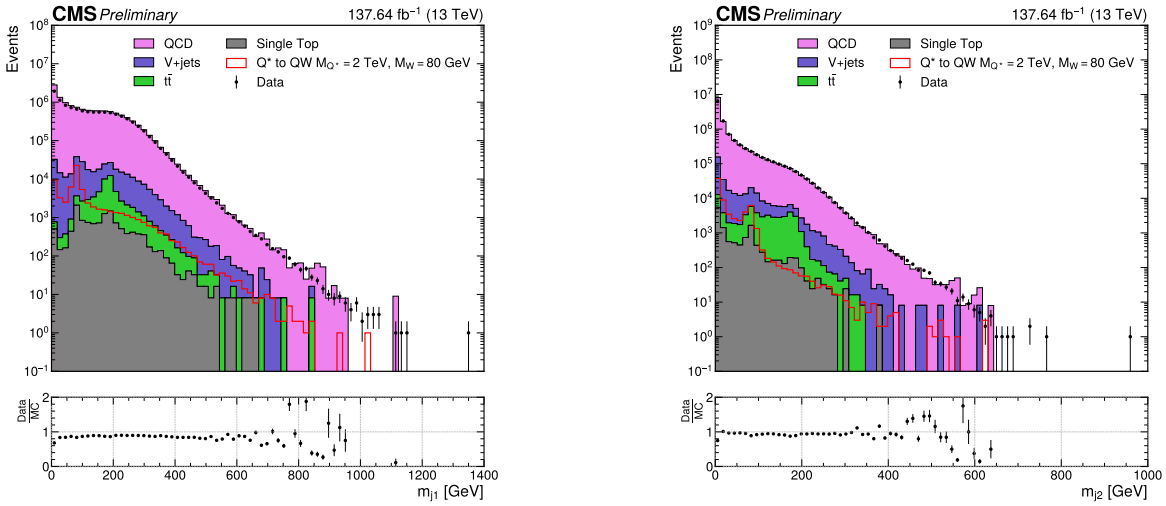


Figure 7.13: Softdrop mass distribution of the jets with the highest (left) and the second highest softdrop mass (right).

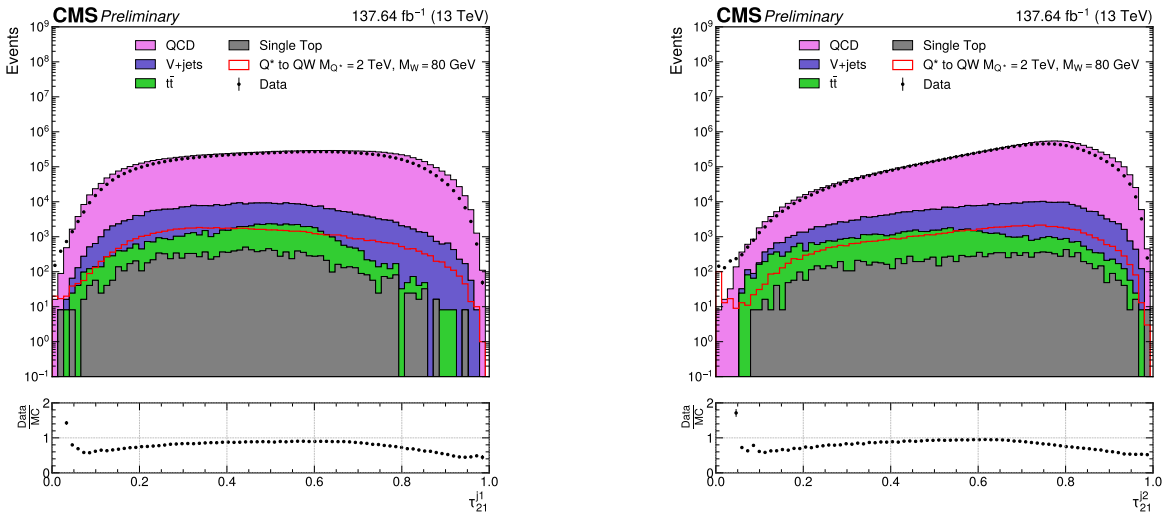


Figure 7.14: Subjettiness ratio τ_{21} of the jets with the highest (left) and the second highest softdrop mass (right).

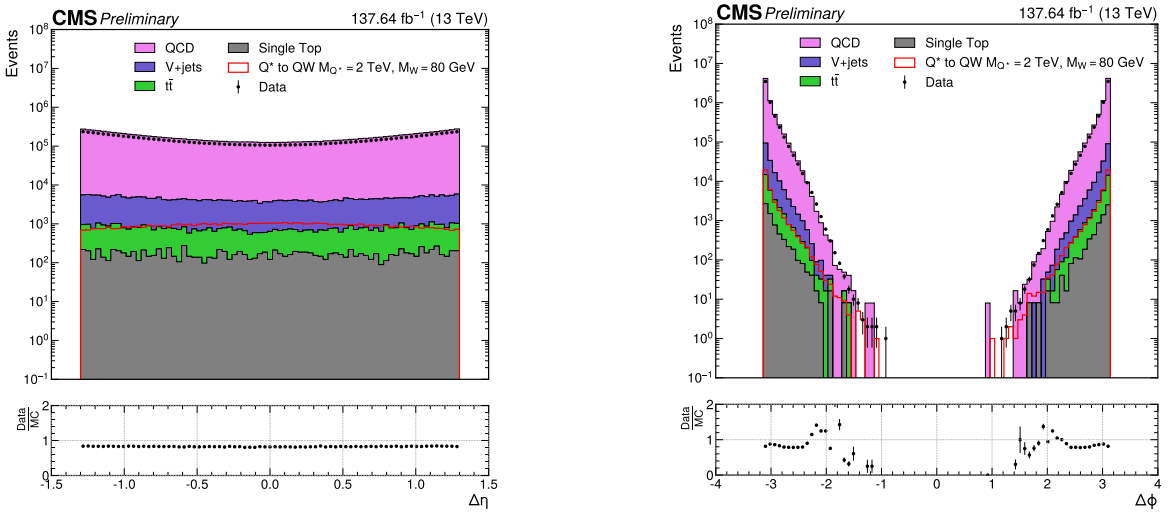


Figure 7.15: Distributions of $\Delta\eta$ (left) and $\Delta\phi$ (right) between the jet with the highest (jet 1) and the second highest softdrop mass (jet 2).

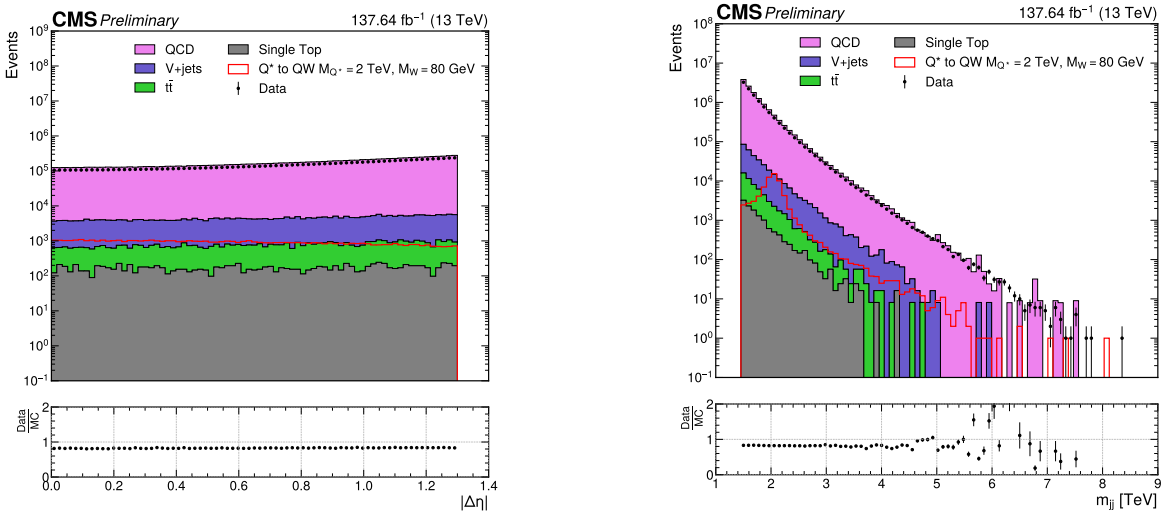


Figure 7.16: Distributions of $|\Delta\eta|$ (left) and invariant dijet mass m_{jj} (right) of the jet with the highest (jet 1) and the second highest softdrop mass (jet 2).

most anomalous events according to a pre-defined selection efficiency, which is based on the statistics of the considered SR bin (see Table 7.3). The output of the fit is a p-value at the investigated mass point. The procedure is scanned through all the SR bins and several mass points per bin such that many signal mass hypotheses are probed for an excess.

The same background function as in previous CMS analyses that look for excesses in the dijet mass spectrum is used [136, 137], which is defined as:

$$\frac{dN}{dm_{jj}} = \frac{\Omega_0(1 - m_{jj}/\sqrt{s})^{P_1}}{(m_{jj}/\sqrt{s})^{P_2+P_3\times\log(m_{jj}/\sqrt{s})+P_4\times\log(m_{jj}/\sqrt{s})^2}}, \tag{7.2}$$

where $\sqrt{s}$ is the centre-of-mass energy, Ω_0 is a normalization parameter and $P_1, \ldots, P_4$ are the parameters describing the shape of the function. However, not all shape parameters are always considered in a fit. Instead, the optimal number of parameters is assessed using Fisher's F-test. Starting with parameters P_1 and P_2, the fit is performed and the goodness of fit is computed based on the χ^2/n_{DOF} value, where n_{DOF} is the corresponding number of degrees of freedom. Then, P_3 is added and the fit as well as the goodness of fit is computed again. This procedure is repeated for P_4 so that all possible parameterizations are fitted once. The optimal number of parameters is then computed using Fishers test statistic, which is comparing two models, indexed by 1 and 2 and is given by:

$$F_{12} = \frac{\chi_1^2 - \chi_2^2}{\chi_2^2} \times \frac{n_{\text{data}} - n_{p_2}}{n_{p_2} - n_{p_1}}, \tag{7.3}$$

where χ_i^2 is the χ^2 value of model i, n_{p_i} is the number of parameters of model i and n_{data} corresponds to the number of datapoints. Due to the iterative procedure of the fits, $n_{p_2} > n_{p_1}$ for all comparisons. While a more complex model will at least fit the data as well as the model with one less parameter, the F-test asserts whether a higher number of parameters results in a *significantly* better fit. This is quantified by a confidence level that the simpler model already represents the correct parameterization. In this analysis, if a confidence level lower than 10 % is obtained, one parameter is added to the function and the F-statistic comparison is repeated, now comparing the models with the next higher number of parameters. This is done until the maximum number of four parameters (P_1 to P_4) is reached.

For the weakly supervised methods, the background fitting and parameter selection procedure is repeated for each of the SR bins in Table 7.3. This is caused by the fact that the models trained in the different regions will likely select different events and their distributions – even though it is required to be smooth in all cases – will have different shapes.

The fit range starts at the point where the trigger efficiency is above the plateau, wich is at 1455 GeV and ends at the maximum of either 5 GeV above the highest m_{jj} value or 20 % above the resonance mass hypothesis. The fit is performed on a fine binning, using a bin width of 4 GeV in m_{jj}. This is done to obtain a good approximation of an unbinned likelihood fit and reduce the computational complexity that fitting the large number of events individually would introduce. For visualizations of the fit results as well as for the χ^2 computation, however, the more coarse "dijet binning" is used that was also employed in [136, 137] and approximately follows the detector resolution. When the fit range is smaller than the full range of dijet bins due to the above mentioned criteria, the dijet bins outside of the fit range are removed. Additionally, in order to reduce the impact of low statistics bins on the χ^2 value, these bins are merged such that all bins contain at least 5 events. Since these criteria

are re-defined for each SR bin for the weakly supervised methods, the binning is slightly different for each of them.

In addition to the background parameterization, a signal resonance shape is needed to conduct the bump hunt. In this analysis, the signal shape is chosen to be a double Crystal Ball function, which describes a distribution with a Gaussian core and power law tails. This function is given as:

$$f(x) = N \begin{cases} A_1 \times \left(B_1 - \frac{x-\mu}{\sigma}\right)^{n_1} & \text{if } \frac{x-\mu}{\sigma} \leq \alpha_1 \\ \exp\left(\frac{(x-\mu)^2}{\sigma^2}\right) & \text{if } \alpha_1 < \frac{x-\mu}{\sigma} < \alpha_2 \\ A_2 \times \left(B_2 - \frac{x-\mu}{\sigma}\right)^{n_2} & \text{if } \frac{x-\mu}{\sigma} \geq \alpha_2 \end{cases}, \tag{7.4}$$

where

$$A_i = \left(\frac{n_i}{|\alpha_i|}\right)^{n_i} \exp\left(-\frac{\alpha_i^2}{2}\right), \; B_i = \frac{n_i}{|\alpha_i|} - |\alpha_i|,$$

N is a normalization constant, and μ, σ and α_i are parameters describing the shape of the resonance.

The procedure regarding the signal shape is different when looking for anomalies in data in a model-agnostic way compared to when a limit for a specific signal model should be computed. For the limit setting, the shape defined in Equation 7.4 is fitted to the entirety of the MC samples available for the signal model in question. For the model-agnostic search, a generic signal shape is used. In particular, the MC samples of the signal $X \to YY'$ with $M_Y = 80\,\text{GeV}$ and $M_{Y'} = 170\,\text{GeV}$ is used, as it produces jets at a reasonable mass that are still light enough to ensure they are fully merged within the large-radius AK8 jets. Therefore, the fit results in a reasonable resonance shape without unwanted features such as large tails. The fit is done on the MC samples of all available mass points for the decaying particle X ($M_X = 2\,\text{TeV}$, $M_X = 3\,\text{TeV}$ and $M_X = 5\,\text{TeV}$). In order to be able to scan a large range of different mass hypotheses also in-between those values where MC samples are available, an interpolation-based approach is chosen. A similar procedure was also used in the CMS diboson resonance search [166]. In particular, the double crystal ball parameters are interpolated between the three available mass point fit values. This is done using a cubic spline for the parameters μ and σ and a linear interpolation for all other parameters. With these interpolations, signal shapes for mass points from 1800 GeV to 5800 GeV are generated in steps of 100 GeV. The parametric shape uncertainties caused by jet energy scale and resolution systematics are accounted for by varying the Gaussian peak position (1.2 %) and width (8 %). The resulting signal shapes are shown in Figure 7.17. It can be seen that the double crystal ball fits and the interpolations result in well-defined resonance shapes. In order to validate the interpolation, a test study was done where only the MC-simulated events of the 2 TeV and 5 TeV mass points were fitted and the interpolation was done based only on the parameters of the two fits. Then, the double crystal ball was interpolated to 3 TeV and compared to the signal shape that would have been obtained by actually fitting the signal MC at that point. The result of this study can be seen in Figure 7.18. The plot shows a good agreement between the interpolated signal shape and the one directly fitted on MC and it can therefore be assessed that the interpolation method produces valid resonance shapes.

As discussed previously, the entirety of the simulated signal samples of the particular signal model and mass point is used to fit the resonance shape in the limit setting. For the weakly supervised methods, special caution must be taken that the weak classification model does not distort the signal distribution. Otherwise, the fitted resonance shape does not correctly describe the signal

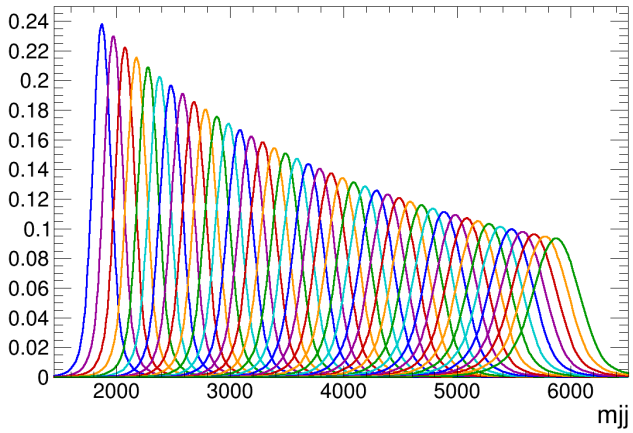


Figure 7.17: All signal shapes resulting from the fits of the $X \rightarrow YY'$ model at 2, 3 and 5 TeV as well and the interpolation procedure described in the text.

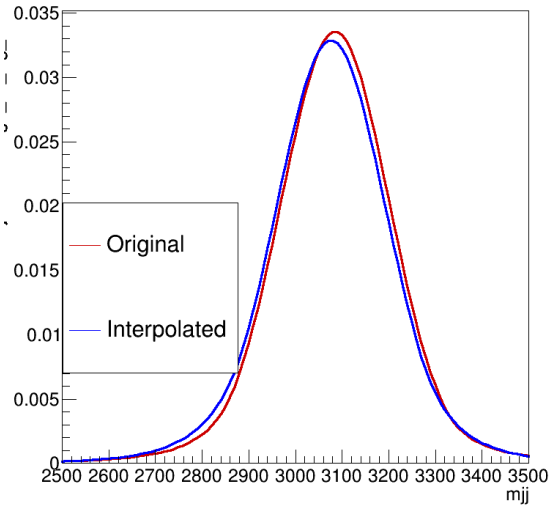


Figure 7.18: Signal shape comparison of the double Crystal Ball function at 3 TeV. The interpolated shape is obtained using fitted parameters from the 2 TeV and 5 TeV mass points. The original shape is based on a fit of the actual signal MC at the 3 TeV mass point.

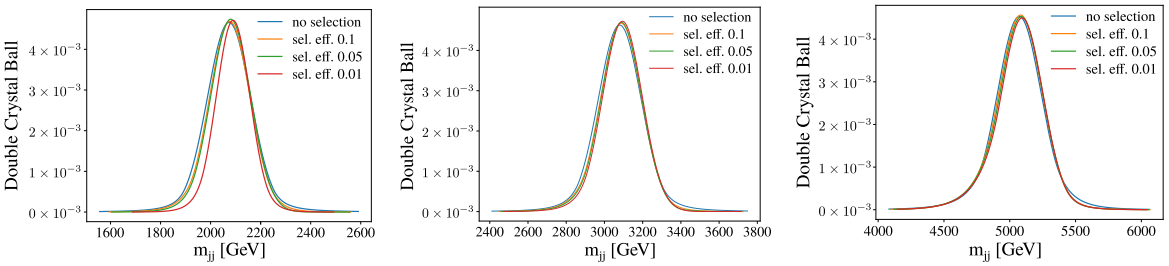


Figure 7.19: Comparison of signal shapes fitted on the entire $X \to YY', m_Y = 80\,\text{GeV}, m_{Y'} = 170\,\text{GeV}$ signal MC (no selection) as well as on the signal MC events after evaluation and selection based on the anomaly score from a CATHODE model. For training this model, an amount of signal events was injected into the MC background that corresponds to the cross section limit obtained from the inclusive fit at each mass point. Three different selection efficiency cuts are investigated: 10%, 5% and 1%. Results are shown for the 2 TeV (left panel), 3 TeV (center panel) and 5 TeV mass points.

distribution after selection, which will result in inaccurate limits. To assess whether the CATHODE-based methods shape the signal distribution significantly, another comparison was made between the shape obtained from a fit of the full signal MC events (using the same signal model as before, i.e. $X \to YY', m_Y = 80\,\text{GeV}, m_{Y'} = 170\,\text{GeV}$) and from CATHODE-selected events. Here, the CATHODE models were trained on background MC samples where signal was injected corresponding to the cross section limit obtained from an inclusive fit at the respective mass point. For the resonance at 2 TeV, CATHODE was trained on the SR bin [1.824 TeV, 2.23 TeV], for the resonance at 3 TeV on the SR bin [2.725 TeV, 3.331 TeV] and for the resonance at 5 TeV on the SR bin [4.5 TeV, 5.5 TeV]. Then events are selected at efficiency cuts of 10 %, 5 % and 1 % and the double Crystal Ball shape is fitted. The resulting resonance shapes are shown in Figure 7.19. From these figures, it can be seen that the CATHODE selection only introduces minor changes in the resonance shape for the 2 TeV and almost no change for the higher mass points. Therefore, using the entirety of the signal MC yields a resonance shape that also describes the selected signal events well. Therefore using the full signal MC for fitting the shape in the limit setting is legitimate in the chosen range of selection efficiencies.

Once a signal shape is obtained, the combined signal and background fit is conducted, using the events selected by the respective anomaly detection method. It should be noted that this is a *template* fit, where the (interpolated) signal shape parameters are fixed and only the parameters of the dijet background function are floating. The fitting procedure is based on a binned likelihood fit, following the considerations in [131], which is shortly outlined in the following. The binned likelihood is defined as:

$$\mathscr{L}(\mu, \boldsymbol{\theta}) = \prod_{j=1}^{N} \frac{(\mu s_j + b_j)^{n_j}}{n_j!} e^{-(\mu s_j + b_j)} \prod_{k=1}^{M} p_k(u_k) \tag{7.5}$$

Here, μ is the signal strength parameter, such that $\mu = 0$ corresponds to the background-only hypothesis whereas $\mu = 1$ corresponds to the signal hypothesis. The indices j correspond to the bin indices of the invariant dijet mass histogram, for which the search for the signal is performed. The s_i and b_i correspond to the signal and background events in bin i and are given by:

$$s_i = s_{\text{tot}} \int_{\text{bin } i} f_s(x; \boldsymbol{\theta}_s) dx \tag{7.6}$$

$$b_i = b_{\text{tot}} \int_{\text{bin } i} f_b(x; \boldsymbol{\theta}_b) dx, \tag{7.7}$$

where f_s and f_b are the probability density functions (PDFs) for signal and background (i.e. the double crystal ball shape and the parametric background function), respectively and s_{tot}, b_{tot} are the total number of signal and background events. The integral runs over x, which in the case of this analysis is the invariant dijet mass, m_{jj}. The parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_b$ are shape parameters of the respective PDFs. The u_i in Equation 7.5 are functions of the nuisance parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_s, \boldsymbol{\theta}_b, b_{tot})$ and run over M bins of a histogram created from subsidiary measurements that help constrain them. The density p_k is pre-determined, typically using a log-normal distribution around the nominal parameter value. In case of this analysis, three of the used nuisances are constrained this way, namely the jet energy scale and resolution (JES and JER) as well as the uncertainty in the selection efficiency ϵ needed for the limit setting (see discussion in section 7.9 for details). The remaining nuisances are left freely floating in the fit. For a hypothesized value of μ, the profile likelihood ratio is considered:

$$\lambda(\mu) = \frac{\mathscr{L}(\mu, \hat{\hat{\boldsymbol{\theta}}})}{\mathscr{L}(\hat{\mu}, \hat{\boldsymbol{\theta}})}. \tag{7.8}$$

The $\hat{\hat{\boldsymbol{\theta}}}$ describes the value of $\boldsymbol{\theta}$ that maximizes $\mathscr{L}$ for the hypothesis μ, whereas $\hat{\mu}$ and $\hat{\theta}$ are the maximum likelihood estimators of $\mathscr{L}$. For the extraction of the p-value, the following test statistic is used:

$$t_\mu = -2\ln(\lambda(\mu)). \tag{7.9}$$

In the large sample limit, an asymptotic approximation can be made based on Wilk's theorem, which states that t_μ approaches a chi-square distribution with a number of degrees of freedom corresponding to the difference in parameters between the signal-plus-background and background-only hypotheses. Based on this distribution, the p-value and significance can be extracted from the t_μ obtained from the fit. A plot of the profile likelihood, i.e. t_μ as a function of the signal strength can be seen for CATHODE using both only background events and an injection of the $X \to YY'$ signal model in Figure 7.20. Smooth likelihood contours are observed, following parabolic shapes. When injecting signal, the contour is asymmetric with larger errors on the signal strength in positive direction compared to the negative direction. This effect is caused by the normalization uncertainty, which describes a fractional uncertainty on the signal yield. Therefore, a larger contribution to the uncertainty at high signal strengths is observed.

In order to test many possible signal mass hypotheses, the weakly supervised methods need to be retrained on various SR bins, which have been summarized in Table 7.3. In particular, 12 overlapping signal region windows are considered, in order to cover the vast majority of the phase space. As mentioned previously, the signal mass hypotheses for the model-agnostic part of the search are starting from 1800 GeV to 5800 GeV in steps of 100 GeV. For each signal mass point, the corresponding SR window is chosen where the distance from this point to the SR center is lowest. For CATHODE(-b), the SB regions are defined as the entirety of the invariant dijet mass outside of the SR window. The lowest and highest bins of each of the two sets of bins, A and B are not used as signal regions, since sidebands must always be present on both sides of the SR window for the weakly supervised methods to work. Since each SR training will yield different sets of selected events, the described fitting procedure is performed separately at each of the SR bins.

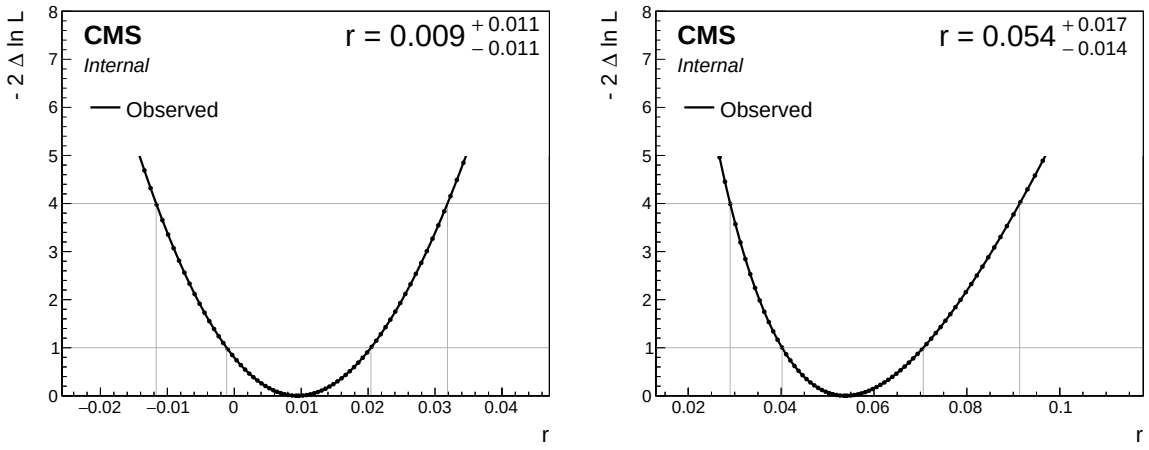


Figure 7.20: $-2\ln(\lambda)$ (denoted as -2Δ ln L) as a function of the signal strength parameter (denoted as r) for CATHODE trained on the background-only mock MC sample (left) and the same sample with an amount of $X \to YY'$ signal events injected that leads to an excess of about 5σ significance (right).

7.7.2 Limit Setting Procedure

One challenge for setting limits in this search is that the performance – and therefore the tagging efficiency – of the weakly supervised methods depends on the amount of signal present in the SR window. If there is a high number of signal events in the data, the resulting weak classifiers will yield a high signal efficiency, whereas when no signal is present, only random noise is learned and the classifiers will not be sensitive to any signal. In turn, the limit setting cannot be performed using the classifier models trained directly on data. Instead, a different method is proposed for limit setting in this scenario: Signal events corresponding to varying cross sections are injected into the data, the weakly supervised training is re-run and finally the evaluation of the training results in a signal efficiency for each of those injections.

The limit on the number of excess events, $N_{\text{exc.}}$ at a given dijet invariant mass is fixed based on the actual observed value on data. To be able to translate this quantity into a final cross-section limit of a specific signal model, $\sigma_{\text{exc.}}$, the selection efficiency of the weak classifier for this signal, ϵ_{NN}, must be known. It is this quantity that is obtained by the re-trainings of varying injections outlined above. The excluded cross section which, in such a scenario, depends on the injected cross section can then be computed as:

$$\sigma_{\text{exc.}}(\sigma_{\text{inj.}}) = \frac{N_{\text{exc.}}}{\mathscr{L} \times \epsilon_{\text{pre.}} \times \epsilon_{\text{NN}}(\sigma_{\text{inj.}})}, \tag{7.10}$$

where $\mathscr{L}$ is the integrated luminosity and $\epsilon_{\text{pre.}}$ denotes the preselection efficiency. For any signal injection, the valid limit is given by the maximum of $\sigma_{\text{exc.}}$ and $\sigma_{\text{inj.}}$.

To further illustrate this procedure, Figure 7.21 is considered. In this illustration, several scenarios exist. First, consider a very high injected cross section. A weakly supervised tagger trained on a high number of signal events will typically yield a good performance and therefore, the limit will be low. This situation corresponds to the point in the lower right of the plot. However, this obtained limit would be invalid, since if there had been only so few events, the classifier performance would have been significantly worse and the signal efficiency would therefore be lower. In such a scenario, the only statement that can be made is that the obtained limit is too optimistic and a conservative estimate

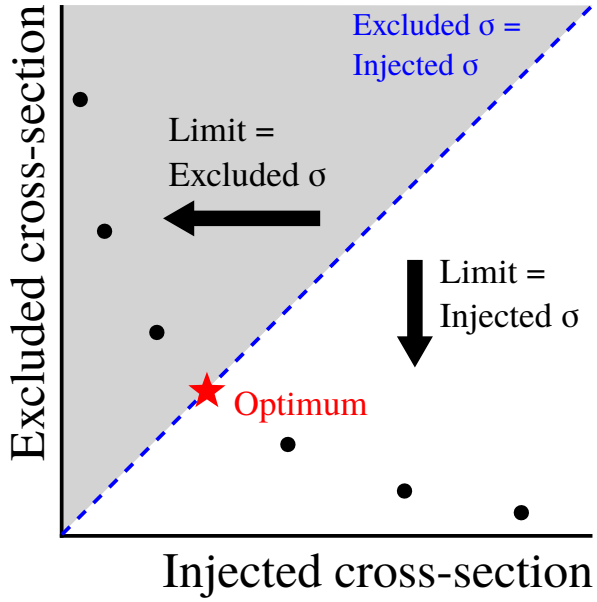


Figure 7.21: Simplified depiction of the limit setting procedure employed for the weakly supervised methods. A detailed discussion is provided in the text.

of the actual limit is acquired by setting it to the injected value.

On the other hand, when the injected cross section is low, the classifier resulting from the respective training will likely have a rather poor performance, since only few signal events were present. In this case, the signal efficiency is low, resulting in a large exclusion limit on the cross section. In particular, the excluded cross section is significantly larger than the injected one. This situation occurs in the upper left of the plot. This time the limit is valid, since a classifier trained on an injected signal amount corresponding to the excluded cross section will likely not be worse than the one trained on the low injection. However, since the injected and excluded cross sections differ by such a large margin, it might be possible to obtain a better limit by using a higher injection, which would improve the classifier. Thus, it is possible to conduct a scan of different injections and repeatedly evaluate the limit until a point is reached where $\sigma_{\text{inj.}} = \sigma_{\text{exc.}}$, which is considered the optimal case.

To set limits for specific signal models, the procedure outlined above is used, conducting a binary scan to find the optimal value where injected and excluded cross sections are similar. For each injection point, 5 retrainings are done with different realizations of the injected signal events. While this procedure increases the computational cost of the methods due to frequent retrainings, it allows for the limit setting for weakly supervised methods.

7.8 Method Validation

In order to ensure that the employed anomaly detection methods yield valid results, it has to be studied whether they produce smooth invariant mass spectra after selection. On the other hand, if the methods significantly sculpted the dijet mass distribution and produced spurious features such as artificial bumps, these would be falsely identified as anomalies by the fit. Additionally, a good fit quality with the used parametric background model has to be ensured for each signal region. To investigate this, all methods were tested on dedicated validation data sets and the results of these studies, in particular for

CATHODE(-b), are discussed in this section. Two data sets are used for the validation procedure: The "mock MC data set" described in subsection 7.5.2 and a data set based on experimentally measured data from a validation region orthogonal to the analysis region.

7.8.1 Validation on simulated samples

The MC data set contains the major and minor backgrounds of the investigated topology and has shown a good agreement with the data sample in the analysis region. The only difference between this data set and the measured data is that it contains fewer events due to the limited availability of QCD MC samples. In particular, it corresponds to $26.8\,\text{fb}^{-1}$ and is therefore about a factor 5 smaller than the full data set. After applying the preselections for the analysis region, the MC data set contains approximately 2.6 million events. CATHODE(-b) is then trained on this data set, using the inputs described in subsection 7.3.1 and the full cross-validation and training procedure described in section 7.4.

The entire procedure is repeated for each SR bin considered in the final analysis. For each of them, events are selected based on a 1 % selection efficiency cut computed from the events in the considered SR and then applied to the SR and SB alike. The m_{jj} spectra of the selected events are then fitted using the combined fit of the generic signal template and the parametric background function, as discussed in section 7.7. The fits are repeated using the interpolated signal shapes every 100 GeV. For each of the scanned points, the fits are based on the trainings of the SR bin with the shortest distance between its centre and the current mass point. The assignments of mass points to the respective SR windows can also be seen in Table 7.3.

The quality of the fits is assessed for each of the SR bins, which is shown for the CATHODE method in Figure 7.22, as well as for the CATHODE-b method in Appendix D. The plots show a good agreement between the parametric fit functions and the selected events, with the vast majority having a χ^2/DoF close to one. This is also reflected in the fit probabilities being around 30 % or higher for most of the plots. In cases where the fit probability is small, this is often due to discrepancies in the low-statistics tails of the distribution.

In one case (bottom centre plot) there is also some sculpting seen in the low mass regions. This behaviour is occasionally observed and can be attributed to the fact that this training was done on a high mass SR window. Since the weak classifier has seen only the features of high mass events, it can tag low-mass events as less anomalous, resulting in a lower selection efficiency in this region. However, the 3-parameter background function successfully accommodates for this shaping and in particular, the distribution in the SR window of interest (in this case between 3331 GeV – 4071 GeV) is smoothly falling and therefore allows for proper signal extraction.

The obtained p-values and local significances of the combined fits are then recorded and plotted against the respective mass values. For CATHODE, this can be seen in Figure 7.23. The same plot for CATHODE-b is shown in the appendix in Figure D.1. From both plots it can be seen that no significant excess is observed when training the CATHODE-based methods on the background-only mock MC data set. For CATHODE, all except one mass point are below a local significance of 2σ and the single high point is still below 3σ. A very similar situation is also seen for CATHODE-b.

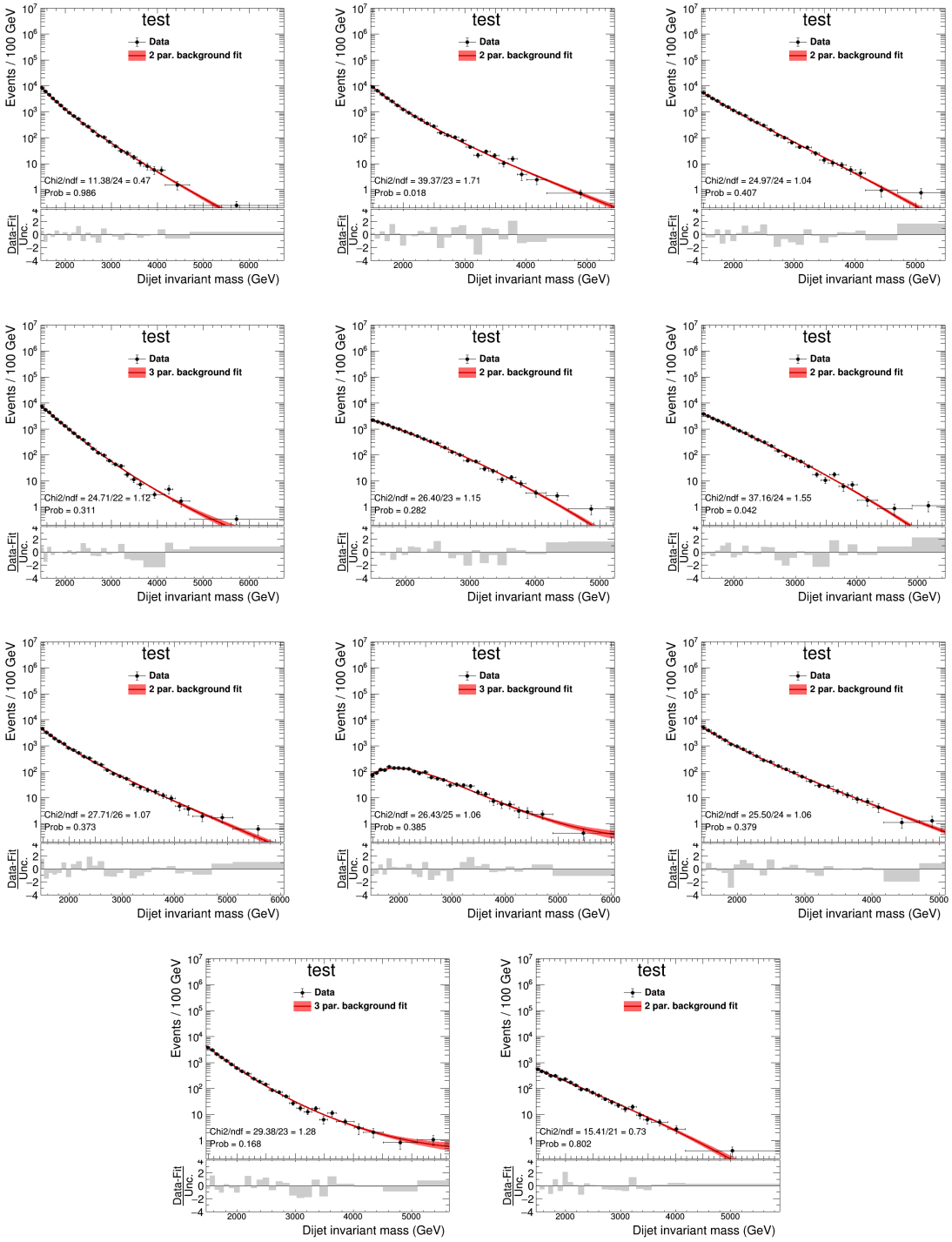


Figure 7.22: Background-only fits of the CATHODE-selected events from the mock MC data set for each considered SR window. Events were selected based on a selection efficiency of 1 %. The plots are sorted by the center SR window mass from lowest to highest in a top left to bottom right order. The lower panel of each plot shows the pull distributions of the fit.

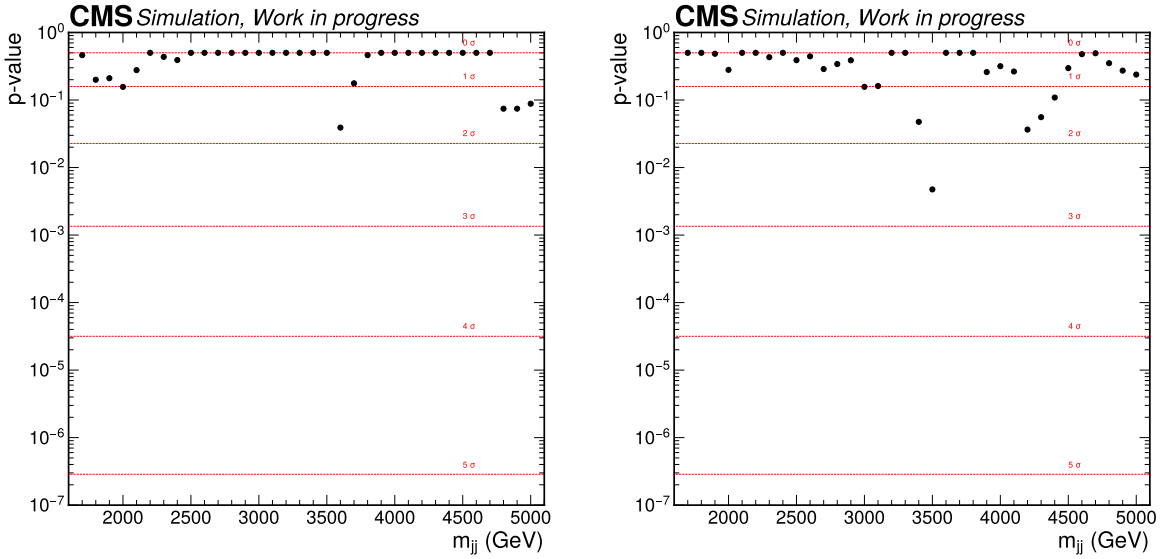


Figure 7.23: CATHODE p-value scan for the mock MC data set. Shown is the p-value obtained from the combined signal template and background fit of the CATHODE-selected events against the dijet invariant mass, m_{jj}. The results are shown for the two selection efficiencies of 10 % (left panel) and 1 % (right panel).

7.8.2 Validation on Data Control Region

The validation using MC samples in the analysis region is an important cross-check to ensure that signal extraction can be done using experimentally measured data. However, purely relying on simulations for the validation procedure is not optimal, since additional effects might be present in data that are not or poorly modelled in MC. Therefore, an additional study was performed, testing the sculpting of CATHODE-based and other weakly supervised methods on measured data. This is done in a specifically designed region that is orthogonal to the final analysis region. The results of this study are discussed in this section.

In order to define a proper control region (CR), two main properties have to be fulfilled: First, the CR must be orthogonal to the analysis region and second, no significant signal contamination must be present in the CR. If these criteria are met, the CR can be used to assess the validity of a method in a background-only case that is statistically independent from the final search. One particular challenge in model-agnostic searches is to define a CR without significant signal presence, since one could probably construct highly exotic signal scenarios that peak in any region of the phase space. Therefore, the CR was designed to suppress a family of signal models that represent the large variety of potential anomalies targeted in the analysis. These signal models are summarized in the appendix in Table D.1 and the corresponding samples in Table D.2. The CR definition is then based on the ratio between the statistical significance $\sigma_{\text{stat.}} = S/\sqrt{B}$ of a signal model in the CR and in the analysis region. In particular, the requirement for a suitable CR is defined as suppressing all of the signal models (for all possible mass combinations) such that a ratio of 10 % or lower is achieved.

To define an orthogonal CR, a $\Delta\eta$ region different from the analysis region could simply be chosen. However, several problems exist by only requiring $|\Delta\eta| > 1.3$. First, the observed suppression of signal is not strong enough for many of the considered models. Second, it results in an m_{jj} distribution that has a significantly different shape than the one in the analysis region. To ensure both

signal suppression and a comparable invariant dijet mass distribution, a different region in $\Delta\eta$ was found. In particular, the CR is defined between $2.0 < |\Delta\eta| < 2.5$. This selection alone reduces many of the signal models to significance values below the desired threshold. However, for the b^* and W_{kk} models, the achieved suppression is still not sufficient. One feature these signals have in common is that their decay involves the production of particles that are heavier than a W or Z boson, i.e. a top quark for the b^* and the radion for the W_{kk} for large values of R_0. In many cases, these are not sufficiently boosted and are therefore resolved into separate jets such that $\Delta\eta$ is not as discriminant as for the signal models resulting in fully merged jets. In order to control this effect, events with a third resolved jet with $p_\mathrm{T} > 300\,\mathrm{GeV}$ are rejected. This cut significantly reduces all but the lightest b^* models and the W_{kk} models with $R_0 > 0.1$ while still being approximately 90 % efficient in Run II data.

When $R_0 \leq 0.1$, the radion of the W_{kk} is sufficiently boosted such that it can be merged in a single, large-radius jets. In these decays, the resulting jets are better balanced in the transverse plane compared to the data, which is why an additional p_T asymmetry cut is employed:

$$\left|\frac{p_{\mathrm{T},j_1} - p_{\mathrm{T},j_2}}{p_{\mathrm{T},j_1} + p_{\mathrm{T},j_2}}\right| > 0.1. \tag{7.11}$$

This cut results in the desired reduction below the significance fraction of 0.1 for all W_{kk} samples with $R_0 \leq 0.1$. Since the additional cut results in a data efficiency of only 37 %, an additional kinematics-based cut is employed and events passing either of them are selected. The additional cut is given by:

$$2\frac{p_{\mathrm{T},j_1} p_{\mathrm{T},j_2}}{m_{jj}^2}(\cosh(\Delta\eta) - \cos(\Delta\phi)) \notin [0.95, 1] \tag{7.12}$$

For highly relativistic particles (i.e. if $E \gg m$), the above expression evaluates to 1. When slightly heavier particles are produced, however, this value gets shifted to values just below 1. Cutting away the region between 0.95 and 1 therefore allows to suppress W_{kk} signals with small R_0 values, while models with larger R_0 stay almost unaffected. Adding this cut and accepting events to pass it or the p_T asymmetry cut increases the efficiency of all cuts to 57 %. The entirety of the cuts can be summarized as:

$$\text{AND}\begin{cases} 2.0 < \Delta\eta < 2.5 \\ \text{No third jet with } p_\mathrm{T} > 300\,\mathrm{GeV} \\ \text{OR}\begin{cases} \left|\frac{p_{\mathrm{T},j_1} - p_{\mathrm{T},j_2}}{p_{\mathrm{T},j_1} + p_{\mathrm{T},j_2}}\right| > 0.1 \\ 2\frac{p_{\mathrm{T},j_1} p_{\mathrm{T},j_2}}{m_{jj}^2}(\cosh(\Delta\eta) - \cos(\Delta\phi)) \notin [0.95, 1] \end{cases} \end{cases} \tag{7.13}$$

After the definition of the CR, the exact same study as for the mock MC data set was done. Both CATHODE and CATHODE-b were trained on the data from the CR using the same settings for the training as before. To assess the goodness of fit, the individual background-only fits and χ^2 values of each SR window are plotted. For the CR data set, the fit range is reduced and starts at 2 TeV in order to mitigate the later m_{jj} turn-on compared to the analysis region. Due to this reason, the two lowest SR windows used in the analysis region are also skipped and the mass scan starts at 2.3 TeV. The fit results can be seen in Figure 7.24 for CATHODE and in the appendix in Figure D.3 for CATHODE-b. The plots again show a good overall agreement between the selected event distributions and the background-only fits,

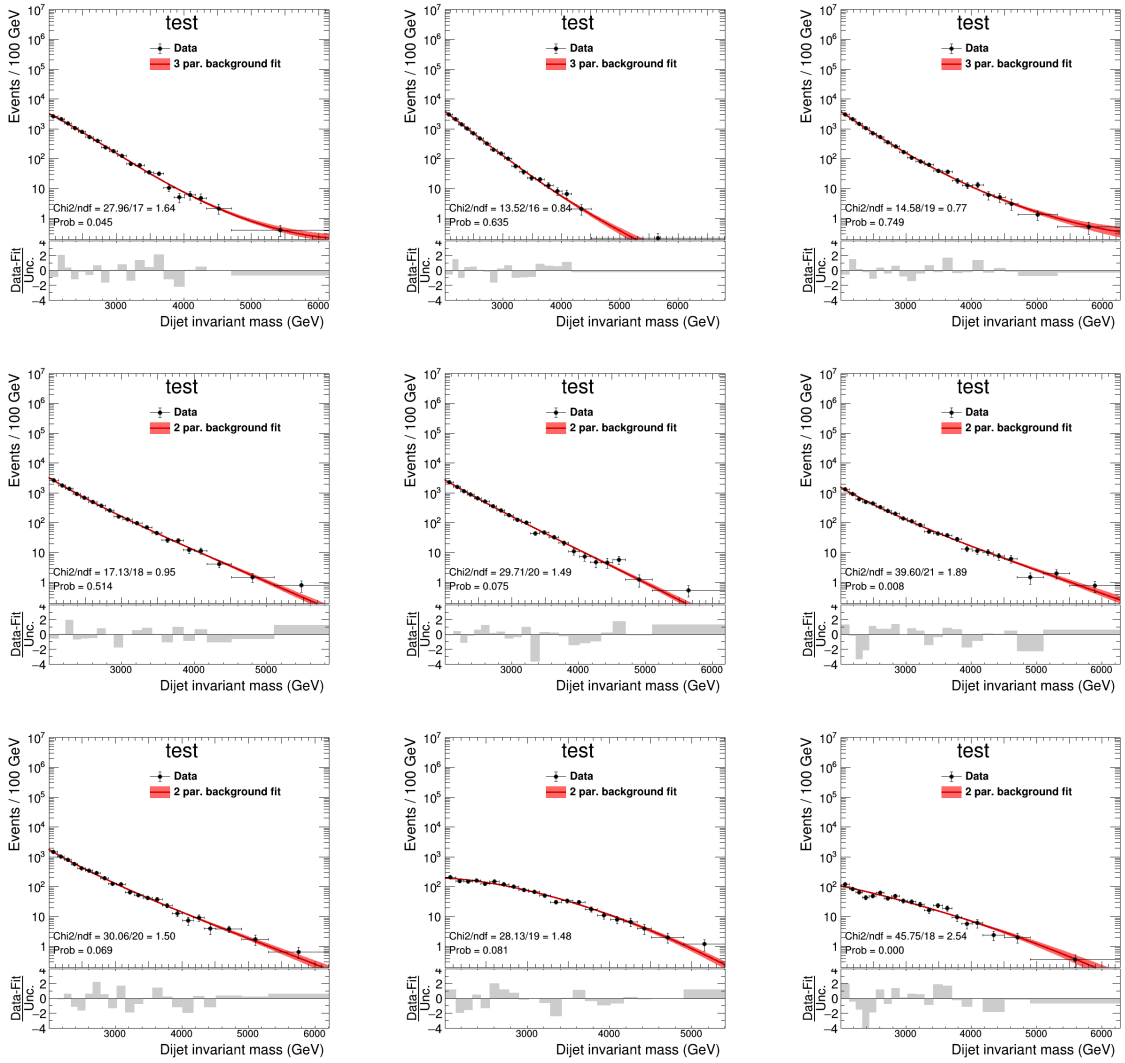


Figure 7.24: Background-only fits of the CATHODE-selected events from the control region data set in each considered SR window. Events were selected based on a selection efficiency of 1 %. The plots are sorted by the centre SR window mass from lowest to highest in a top left to bottom right order. The lower panel of each plot shows the pull distributions of the fit.

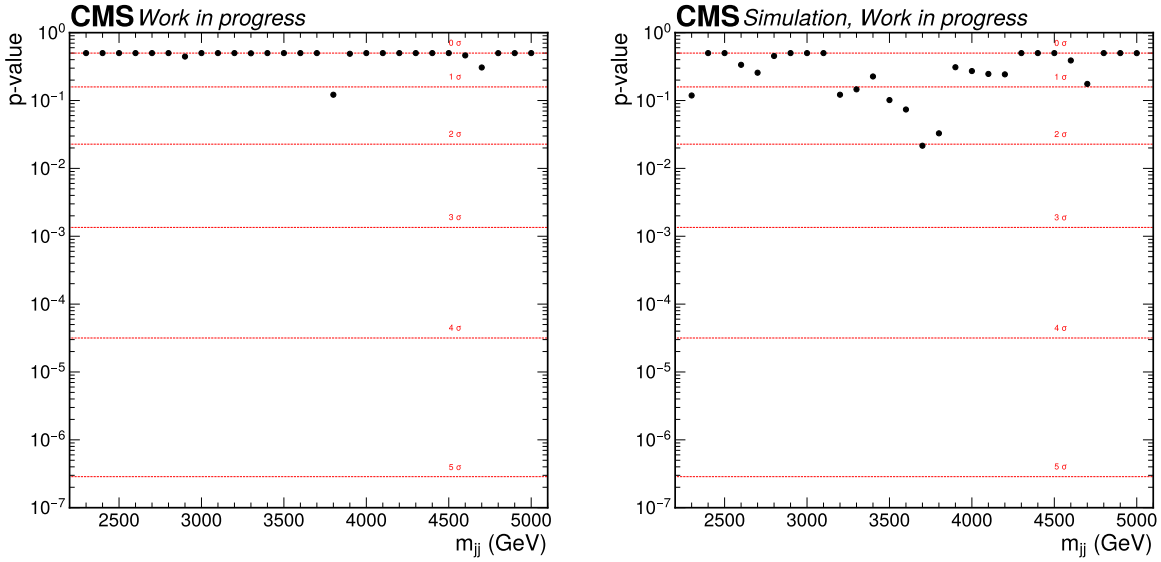


Figure 7.25: p-value scan for the data set from the CR, using a selection efficiency of 1 %. The plots show the p-value obtained from the combined signal template and background fit of the selected events against the dijet invariant mass, m_{jj}. The results are shown for the CATHODE (left panel) and CATHODE-b methods (right panel), respectively.

including satisfactory χ^2 values for most of them. In particular, no significant sculpting of spurious features is observed within the considered SR windows.

The resulting p-value scans can be seen in Figure 7.25 for both CATHODE and CATHODE-b at a 1 % selection efficiency. Similar to the mock MC data set, no significant excess is observed for CATHODE or CATHODE-b using the data in the CR. In particular, all points show a significance lower than 2 σ. To summarize, it can be concluded that the CATHODE-based methods do not learn spurious information from background-like samples and produce smooth distributions after selection that allow for proper signal extraction in both simulation and data.

7.8.3 Signal Injection Tests

While the previous section focused on the validation of CATHODE-based methods in background-only scenarios, the performance in the presence of signal remains to be studied. Therefore, signal injection tests were performed on the simulated samples of the mock MC data set to assess the expected significance extracted from the fitting procedure. In particular, the signal model $X \to YY'$ with $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$ and $m_{Y'} = 80\,\text{GeV}$ is used as a benchmark. The injection tests are done by scanning over injected cross sections, using steps of 5 fb. As in the final analysis on data, a selection efficiency of 1 % is used. To reduce the computational cost of this study, not all SR windows are being scanned for each cross section, but only the three SR windows around the mass value of the benchmark signal. This corresponds to scanned mass hypotheses between 2.6 TeV and 3.5 TeV. The significance plots for this study are shown in Figure 7.26 for CATHODE and in the appendix in Figure D.4 for CATHODE-b. In these plots, the values at the scanned mass points between 2.6 TeV and 3.5 TeV are based on the signal-injected trainings of the considered SR regions, while points outside that region are showing the background-only results and thus do not change. The CATHODE plots show that the significance around the mass value of 3 TeV starts to increase at an injection of 15 fb, already achieving the thresh-

old of 3 σ. For even higher injections, the significance increases significantly, achieving almost 20 σ for the highest injection. In the latter case, there is so much signal that for some SR window fits, the background only fit does not converge, which is shown by the red stars in the plot. This, however is not problematic, since such a large signal is not expected in data and if it was present, the large resonance shape can clearly be seen by eye. For CATHODE-b the situation is very similar. Starting from the 15 fb injection, the resonance becomes clearly visible and steadily increases up to a value of 14 σ for the highest injection. In addition to the significance plots, the individual signal plus background fits at the injected mass of 3 TeV are shown in the appendix in Figure D.5 and Figure D.6 for CATHODE and CATHODE-b, respectively, again showing a good agreement for the combination of the background and the injected signal events.

7.9 Systematic Uncertainties

Since this analysis is based on entirely data-driven approaches, no systematic uncertainties regarding the background have to be taken into account. The model-agnostic significance scan using the generic signal shape, however, is affected by systematic uncertainties regarding the simulated signal events. These uncertainties have an impact on the resulting resonance shape used in the combined fit and therefore have to be considered. For the limit setting procedure, the weakly supervised methods in particular are affected, which is connected to the specific limit setting approach that has been discussed.

7.9.1 Shape Uncertainties

The uncertainties regarding the signal shape are based on the jet energy scale (JES) and the jet energy resolution (JER). The uncertainties were estimated by varying the four-vectors of the two AK8 jets according to the respective corrections and then re-fitting the double Crystal Ball shapes. This causes the μ and σ parameters to shift up and down and these variations are taken as their uncertainties. In the combined signal-plus-background shape fit, the parameters are allowed to float but have a Gaussian constraint based on the obtained uncertainty from the up-/down-variations. For the generic template which is based on the $X \to YY'$ model this procedure was applied to the 2 TeV, 3 TeV and 5 TeV mass points and for each of them, similar fractional uncertainties were obtained. In particular the resulting uncertainty was 1 % for μ and 3.5 % on σ. Since the uncertainties showed a good agreement between shapes of different mass points, the obtained values are used for all interpolated templates in the significance scan. For the signal-specific limit setting, the described procedure is repeated for the derivation of uncertainties for the particular model in question. However, values similar to the ones obtained from the generic model are seen also in general.

7.9.2 Normalization Uncertainties

The signal normalization uncertainties have to be considered in the limit setting, since they affect the tagging efficiency of the methods. Since this work focuses on the weakly supervised methods in particular, the application of systematic uncertainties in the limit setting procedure is discussed in more detail in the following section.

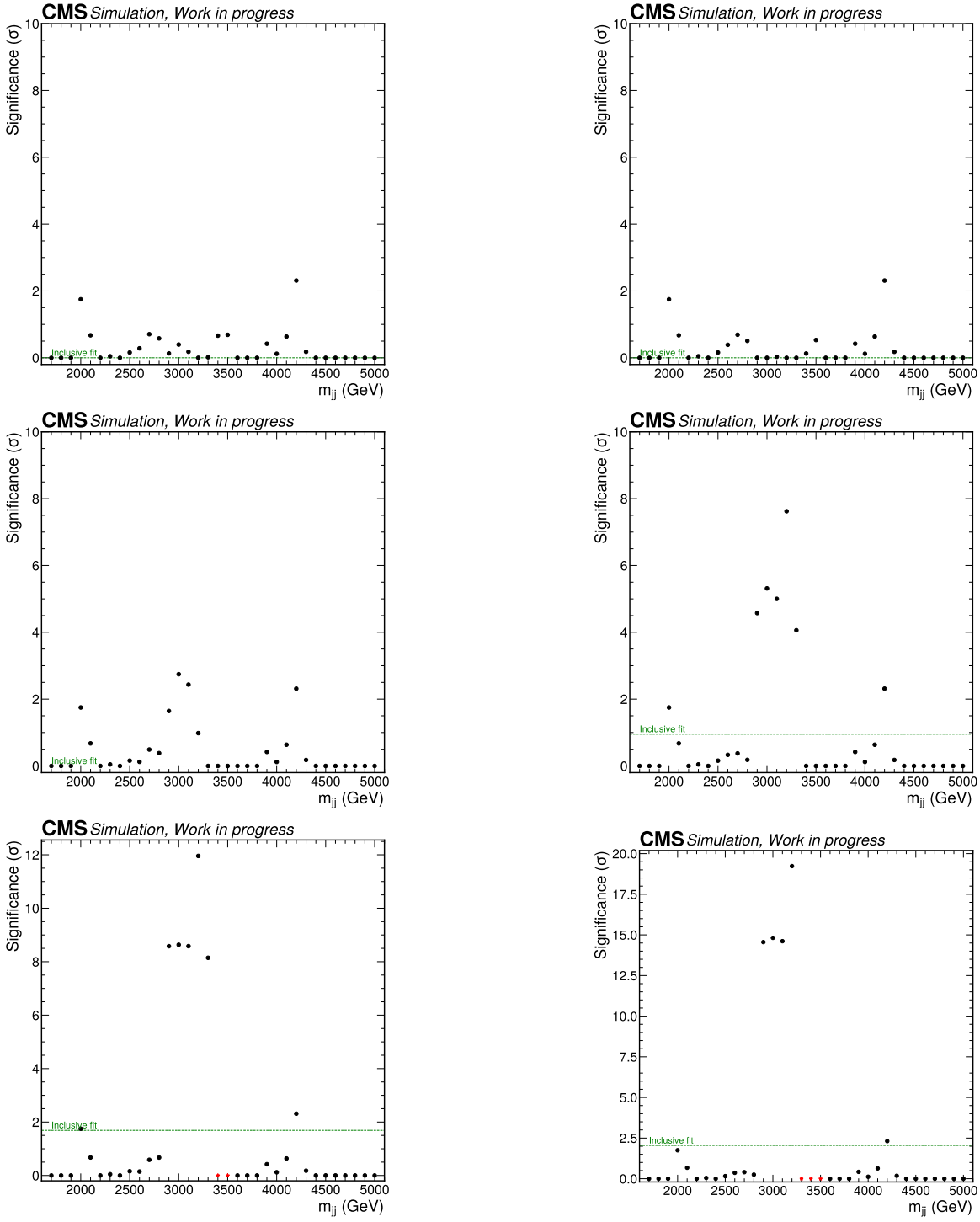


Figure 7.26: Significances obtained from CATHODE when injecting an increasing amount of $X \to YY'$ signal in the MC mock data set, with $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$, and $m_{Y'} = 80\,\text{GeV}$. Note the dynamic y range. The injected cross sections are starting at 5 fb in steps of 5 fb and the plots are ordered by increasing injections from top left to bottom right. Black dots represent the significance obtained from the combined signal shape and parametric background fits on CATHODE-selected events, using a selection efficiency in the signal region of 1 %. The red stars for the injection of 30 fb correspond to failed fits. The dashed green lines represent the significance achieved using an "inclusive" fit, where the combined fit is applied to the resonance mass of 3 TeV without any selection.

The considered uncertainties are listed below and all of them are eventually added in quadrature, resulting in a single nuisance on the signal normalization used in the fit.

1. **Sub-structure modelling**

   A key feature of this analysis is that highly exotic signal models with a large number of sub-jets are considered. Usually, when estimating uncertainties on the tagging efficiency for a specific model, control regions in data and MC are used to derive a corresponding scale factor. However, this approach is not possible in this work, since no Standard Model proxy exists that produces more than three subjets in its decay. Instead, an entirely new procedure is employed using per-prong substructure corrections, which can in principle be applied to any n-pronged signal model [206]. A detailled discussion of this new procedure is beyond the scope of this work. In short, a data/MC ratio of the Lund Plane of the subjet is derived, using a control region of boosted W's. This results in a central value and an uncertainty on a tagging scale factor for any substructure-based selection. This uncertainty has been found to be dominant for CATHODE-based methods.

2. **Variation in NN performance**

   The training, in particular of the weakly supervised methods, is subject to the stochasticity of the model initialization and minimization procedures, which impacts the final tagging performance. Therefore, multiple re-trainings are done based on the injection of different, random samples of signal events. An uncertainty on the tagging efficiency is assigned reflecting this performance variation. This is has been found to be the second largest uncertainty for CATHODE-based methods, with the injection uncertainty having a larger impact than the retraining. Except for this uncertainty and the uncertainty due to sub-structure modelling, the remaining uncertainties listed below have been found to only have a minor impact on the final result.

3. **Jet energy/mass scale and resolution**

   The effect of the jet energy scale (JES), jet energy resolution (JER), jet mass scale (JMS) and jet mass resolution (JMR) correction uncertainties on the signal normalization are assessed. This done by producing samples where the four-vectors of both AK8 jets are shifted by the provided uncertainty. For the JMS, a 5 % uncertainty is applied to both AK8 jets. For the JMR, the default central values and corresponding uncertainties are used, which are derived from W-tagging. These uncertainties are at 8 % for the considered years [207–209].

4. **Pileup reweighting**

   This uncertainty is evaluated by shifting the minimum bias cross section up and down by a value of 4.6 % [210]. A new weight is computed and applied for each shift in order to acquire the up and down variation of the signal shape and yields.

5. **Parton Distribution Function (PDF) uncertainties**

   The standard deviation of the 100 PDF variation weights is computed for each event. The up and down variations are acquired as the nominal weight plus/minus the standard deviation.

6. **L1 prefiring**

   This uncertainty is related to a detector effect that impacted the CMS level 1 (L1) trigger during the 2016 and 2017 data taking eras [211]. It caused the loss of a fraction of events, which has to be compensated for in MC. Therefore, simulated events are reweighted based on the recommendations of the CMS L1 detector performance group. From the probability of each AK4 jet or photon to cause this effect, an event weight is calculated and the corresponding uncertainty is obtained by up/down variation of said weight.

7. **B-tagging**

   This uncertainty is relevant for methods using b-tagging scores as inputs, which is the case for CATHODE-b. Therefore, the effect of the corresponding uncertainties on the shape of the score distribution has to be taken into account. As for the other uncertainties, the evaluation is based on the central CMS recommendations [212]. In total, eleven sources of uncertainty are considered, depending on whether a jet is a b-jet, a c-jet or a light jet. To reduce the computational cost of these uncertainties for the limit setting procedure, the variations of all the sources are added up in quadrature to produce a single up and down weight per event. The resulting weights are normalized such that they only affect the shape of the DeepCSV score distribution and the normalization stays unchanged.

8. **Luminosity**

   An uncertainty of 1.6 % is applied for the total Run-II luminosity [213].

9. **Initial State Radiation (ISR)**

   The parton shower uncertainties for the initial state radiation are used based on PYTHIA8 [214]

10. **Renormalization & factorization scales** The scales are shifted up and down by a factor of two. Three sets of scale variations are considered, one where each of the renomalization and factorization scales are varied individually and one where both are varied in the same direction simultaneously.

11. **Top p_T reweighting**

    Compared to MC simulations, it was found that the p_T spectra of top quarks are significantly softer in data. Therefore, a systematic uncertainty for this mismodeling is considered according to CMS recommendations [215].

7.9.3 Systematic Uncertainties for Weakly Supervised Limit Setting

As discussed in subsection 7.7.2, a specific procedure must be followed to obtain valid limits for weakly supervised methods. Another dimension of this problem that warrants further discussion is how to incorporate systematic uncertainties in this scenario. Changes in the input feature distributions of signal events due to systematics have an impact on the signal efficiency resulting from the training. Therefore, this impact should be evaluated by retraining the classifiers using the shifted distributions for each source of uncertainty.

First, the optimal injected cross section according to the procedure outlined in subsection 7.7.2, $\sigma_{\text{inj.}}^{\text{opt.}}$ is computed. Then, signal injections at this cross section are sampled using the shifted distributions and the classifiers are retrained. This procedure, however, is computationally expensive due to the vast amounts of classifiers that are already trained for the *k*-*l*-folding and for the quantification of the NN performance variation. Adding another retraining for each considered systematic uncertainty would make conducting this analysis unfeasible. Therefore, the number of systematics for which a retraining is performed is reduced based on a specific criterion. This is done by considering the magnitude of the distribution change due to the up and down variations and relating it to the statistical uncertainty of the sample. Since only a limited number of signal events is injected, any deviation that is considerably smaller than the statistical error is likely having only a marginal impact on the training result. Using a histogram with a bin size of 10 for each input feature, a retraining is performed only if the deviation due to systematics in any of the bins is larger than 20 % of its respective statistical uncertainty.

To evaluate the uncertainty of the statistical fluctuations of the injected signal events, four additional sets of events are randomly drawn from the full available MC sample corresponding to a cross section of $\sigma_{\text{inj.}}^{\text{opt.}}$. For each of these injected sets of events, a full retraining is performed, such that in total a number of 5 signal efficiency estimates are obtained based on the 5 different realizations of the signal sample. The average of these efficiencies, ϵ'_{NN} is defined as the nominal value and the minimum/maximum as the up/down variation for the uncertainty regarding the choice of signal events to inject.

For any systematic uncertainty that does not require retraining, the five models are evaluated with the distributions shifted up and down. The mean variation in each direction is than taken as the estimated value for the respective uncertainty on the signal efficiency. For systematics where a retraining is necessary, the uncertainty on the signal efficiency is defined as the change in efficiency with respect to the model trained on the shifted sample. Finally, all sources of uncertainty on the tagging efficiency are added in quadrature to obtain a single value as a systematic. This is then added as a log-normal uncertainty on the signal yield and the limit setting procedure is repeated. The number of excluded events including the systematics, $N'_{\text{exc.}}$ is always strictly larger than the original value obtained without incorporating these uncertainties, $N_{\text{exc.}}$. Therefore the new exclusion limit, $\sigma'_{\text{exc.}}$ is also generally larger by construction.

A comparison of the impact of systematic uncertainties on the signal efficiency is shown in Figure 7.27. The plots show the fractional uncertainty on the signal efficiency for reevaluation and retraining of the weak classifiers at the example of the $X \to YY'$ signal model at $m_X = 3\,\text{TeV}$, $m_Y = 80\,\text{GeV}$ and $m'_Y = 170\,\text{GeV}$. The uncertainties producing the largest deviations from the nominal distributions and therefore required retraining are the jet energy corrections (JES, JER, JMS, JMR) as well as the Lund plane-based systematic for substructure tagging and the injection systematic. The other plot shows all the considered systematics for the reevaluated classifiers. For systematic effects shown in both plots, the maximum deviation is taken as the final value. The reevaluated results again show a large impact by the Lund Plane-based effects as well as the injection uncertainty, while other systematics are relatively small, with many deviations below percent level. While these results represent only the case for the mentioned $X \to YY'$ model as an example, the results for most other models are similar in terms of which uncertainties had the largest impact overall. This illustrates both the

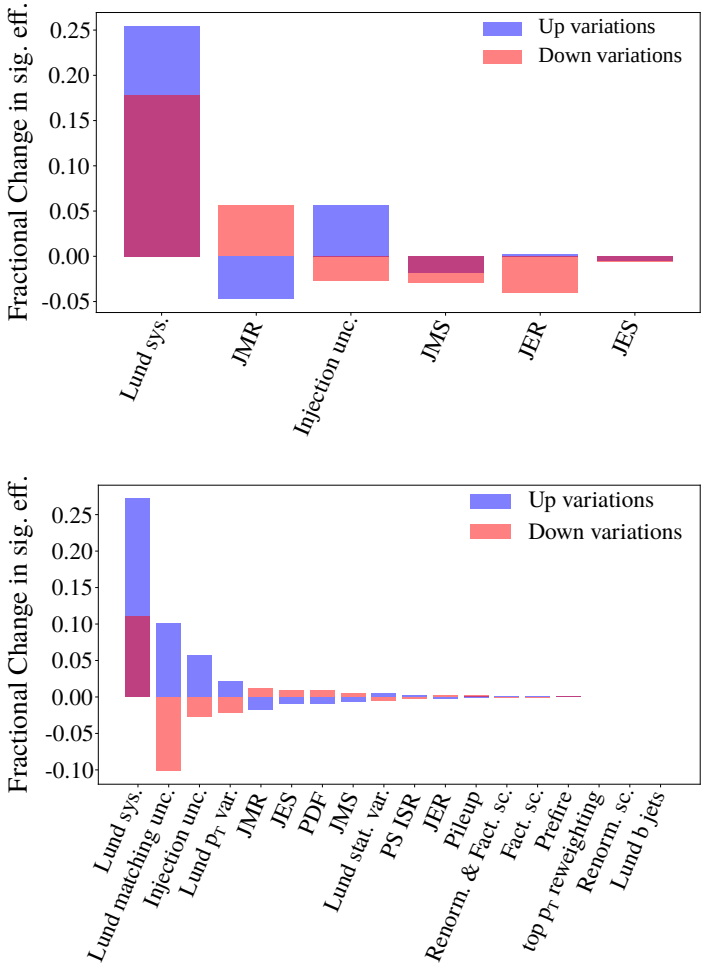


Figure 7.27: Fractional change in signal efficiency for each systematic in the limit setting of the $X \to YY'$ signal model at $m_X = 3\,\text{TeV}$, $m_Y = 80\,\text{GeV}$ and $m'_Y = 170\,\text{GeV}$ in data using CATHODE. Results are shown for up and down variations in blue and red, respectively. The upper plot shows the systematics for which the weakly supervised classifiers had to be retrained whereas the lower plot shows the variations for the reevaluated classifiers that were trained on the nominal distributions.

sensitive dependence of weakly supervised methods on the specific signal events injected as well as the challenge in tagging events with highly anomalous substructure, which cannot be constrained by existing SM processes.

7.10 Results

Having established and validated the machine learning and statistical methods employed in this search, this section mainly discusses the results on data measured by the CMS experiment in the analysis region. In particular, the results of CATHODE and CATHODE-b as well as several comparisons of all methods are being considered.

7.10.1 Performance Comparison

Before considering the results obtained on measured data, the expected sensitivity of the employed methods is discussed in more detail. To test this, two signal models were chosen as a benchmark and signal events corresponding to increasing cross sections were injected into the mock MC data set. In particular, the 2/2-pronged signal model $X \to YY'$ with $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$ and $m_{Y'} = 170\,\text{GeV}$ as well as the 3/3-pronged signal model $W' \to B't \to bZt$ with $m_{W'} = 3\,\text{TeV}$ and $m_{B'} = 400\,\text{GeV}$ are used. For each of the injections, all the methods are trained and the corresponding p-value is computed for each of them, using the previously discussed statistical methods. This study therefore does not only yield a comparison of the method performance for the two benchmark signals, but also assesses how the sensitivity behaves as a function of the injected amount of signal.

As an additional reference, several standard analysis methods are compared with the machine learning-based ones. Specifically, the results of an inclusive search without applying any selection is shown, as well as a model-specific selection for two-prong ($\tau_{21} < 0.4$) and three-prong ($\tau_{32} < 0.65$) jets, with an additional requirement of $m_{\text{SD}} > 50\,\text{GeV}$. Finally, a result is also added for a fully model-specific search, which is implemented using the *QUAK* method in a setting where it is trained on the exact same signal model that is considered in the respective plot.

The results of this study are shown in Figure 7.28. From the plots it can be seen that CATHODE-based methods are promising candidates for an application in an actual search on experimental data, as their performance is amongst the best of the unsupervised ones. Apart from *QUAK*, which is a semi-supervised method, CATHODE outperforms all other unsupervised methods on the $X \to YY'$ signal model. In particular, among these methods CATHODE is the one that achieves significances of $3\,\sigma$ and $5\,\sigma$, the thresholds for scientific evidence and discovery, at the lowest injected cross section. CATHODE-b is also performing well on this signal model, with slightly lower performance than CATHODE. The performance decrease is expected, since no b-tagged jets are present in the decay. Therefore, the b-tagging feature is expected to be largely uninformative and acts as an additional source of noise in the weak classification task. In cases where b-tagging information is present, however, a significant performance increase of CATHODE-b vs. CATHODE can be observed.

Considering the results of the W' signal model in the right panel of the plot, one can see that CATHODE is amongst the worst of all methods, while CATHODE-b is amongst the best, only surpassed in sensitivity by the TNT method. This underlines the value of incorporating different methods using different features in a search, broadening the sensitivity to a larger range of different signal mod-

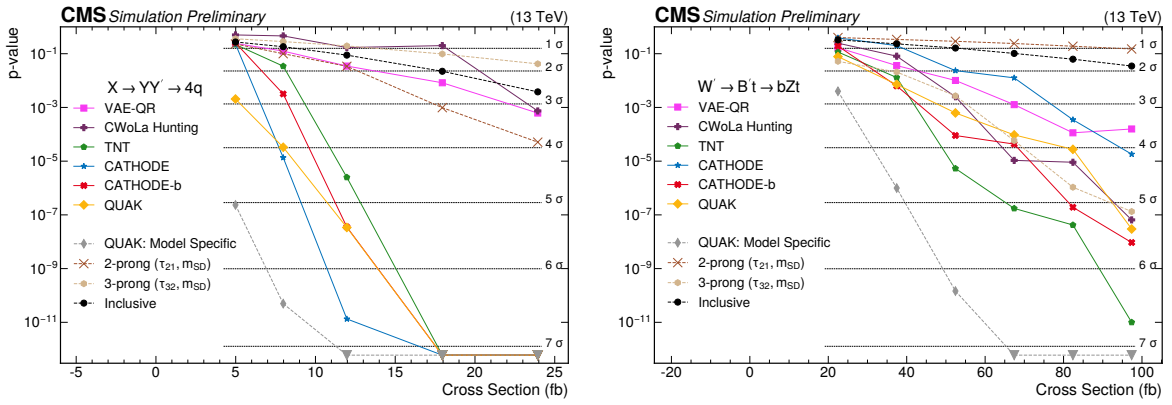


Figure 7.28: Comparison of the p-value achieved as a function of the injected signal cross-section for each method employed in this analysis. The plot shows the results for the $X \to YY'$ (left panel) and the $W' \to B't \to bZt$ (right panel) signal models. For reference, the results corresponding to an inclusive search (no selection), two model-specific selections and a fully supervised search based on the *QUAK* method are also shown.

els. This is also reflected in the fact that no single method is optimal in both cases. In general, it can be seen that for both signal models, many of the employed methods significantly outperform an inclusive search, as well as the model-specific cut-based approaches, often increasing the sensitivity beyond discovery level. This emphasizes that unsupervised and semi-supervised methods are promising candidates to conduct model-agnostic searches for anomalies in data.

7.10.2 Significance Scan

For the significance scan using a generic signal shape, the CATHODE-based models were trained using the training procedure outlined in section 7.4 and the statistical analysis was based on the combined signal-plus-background fit described in subsection 7.7.1. Before the p-values/significances obtained from the data are discussed, the background estimation in the analysis region is considered. Since the bump hunting procedure is repeated for each SR window training, the m_{jj} spectra of events selected by CATHODE(-b) are plotted for each SR bin, together with the background-only fit result. This can be seen in Figure 7.29, where the upper and lower rows contain the results for CATHODE and CATHODE-b, respectively. The columns relate to the set of SR bins used, namely set "A" (left) and set "B" (right). Since each SR window is based on a different training, the selected spectra differ in shape, leading to the "stitched" discontinuous spectra for each bin. It can be observed that smooth spectra are obtained for both methods and the background-only fits describe the overall shape of the invariant mass distributions well. For reference, the full fits including the SB regions and goodness of fit values are added to the appendix and can be seen in figure Figure E.1 for CATHODE and Figure E.2 for CATHODE-b. Also in these plots, the smooth spectra can be seen, both in the SR and SB regions. The fit generally describes the data well, with satisfactory goodness-of-fit values and fit probabilities being achieved.

The p-values and corresponding significances were again obtained using the combined signal plus background fit using the generic signal shape template, interpolated to the mass point to be scanned. The scan starts at $m_{jj} = 1700\,\text{GeV}$ and covers mass points in steps of 100 GeV up to a value of $m_{jj} = 5000\,\text{GeV}$. The resulting p-values of this scan on events selected by CATHODE(-b) can

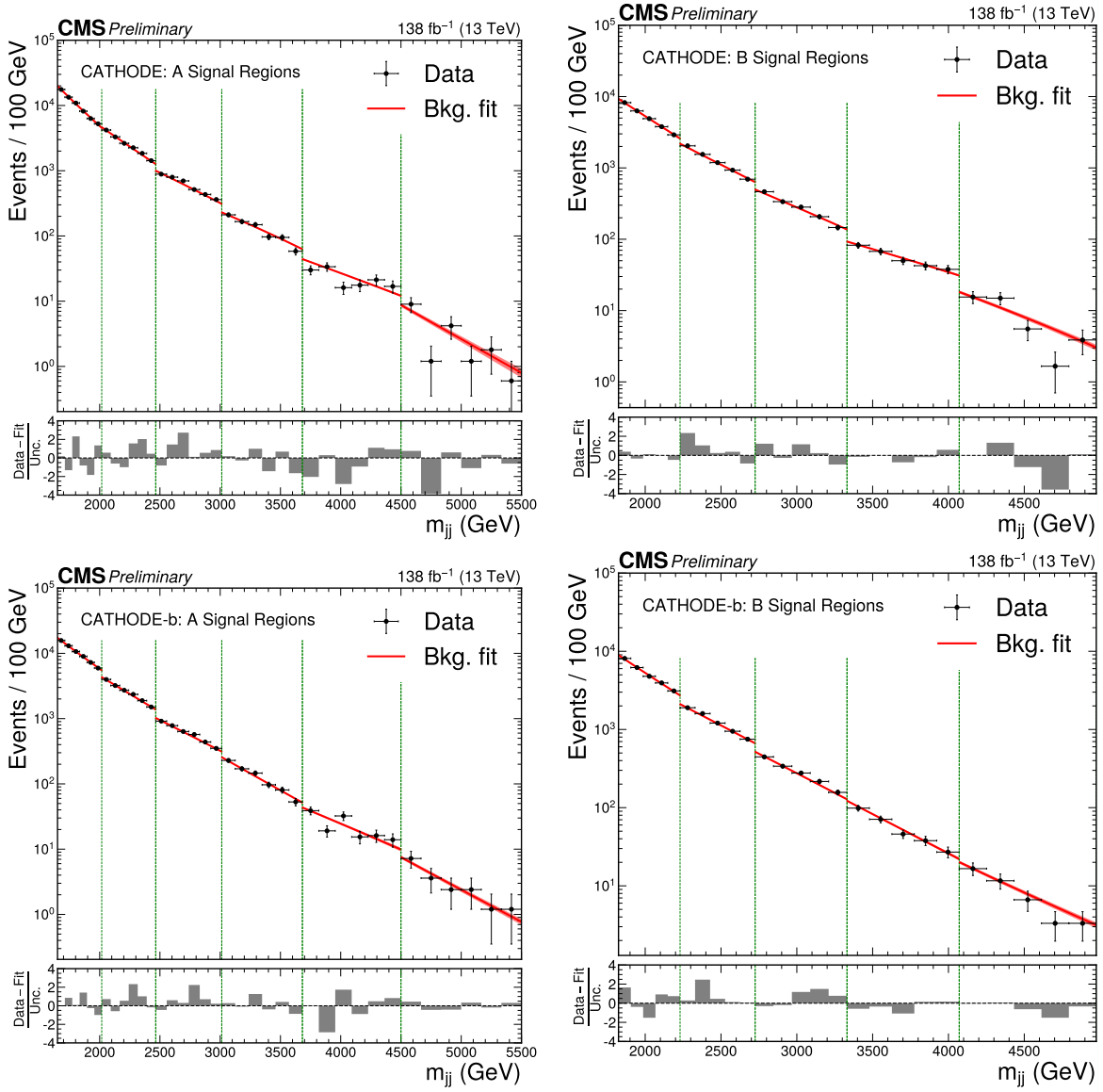


Figure 7.29: Invariant mass distributions and parametric background fits of CATHODE(-b) selected events in the data analysis region for all considered SR bins. The plots are based on a 1 % selection efficiency in the SR. The upper and lower rows show the results for CATHODE and CATHODE-b, respectively. The left column corresponds to results for the set of SR bins labelled "A" while the right column corresponds to the set of SR bins labelled "B" (see Table 7.3 for reference). The lower panel of each plot shows the pull distribution of the fit in each invariant mass bin.

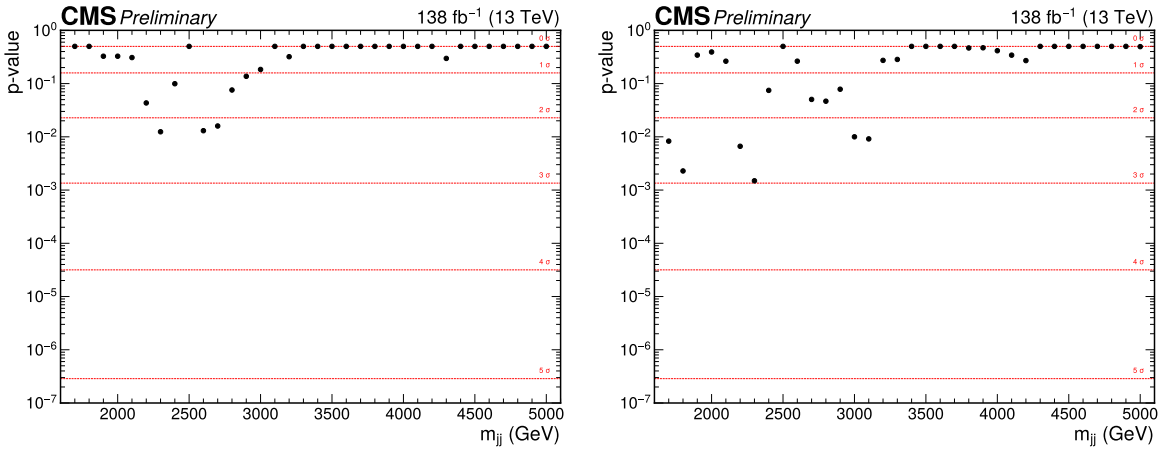


Figure 7.30: P-values for different mass points obtained from the combined fit using a generic signal template and the parametric background function. Red dashed lines are plotted for reference, showing the corresponding values for different significance levels. Results are shown for CATHODE (left panel) and CATHODE-b (right panel).

be seen in Figure 7.30. The plots show that no significant excess can be observed, with none of the points reaching beyond evidence-level significances. The highest observed local significances were 2.2 σ for CATHODE and 3.0 σ for CATHODE-b, both for a signal with a mass of 2300 GeV. The other employed methods did not find significance values above the evidence level as well, with the highest local significances obtained being 2.6 σ and 2.3 σ for the *QUAK* and *VAE-QR* methods, respectively. The *CWoLa Hunting* and *TNT* methods did not find any values higher than 1.5 σ. Therefore, it has to be concluded that no evidence of new physics anomalies is observed in the data.

7.10.3 Limits

Since no evidence for new physics was found in the generic search, the expected sensitivities with respect to specific signal models were compared for the employed anomaly detection methods. For this study, a subset of considered signals was chosen as a common benchmark. In particular, the $X \to YY'$, $W' \to B't$, $W_{kk} \to WR$ and $Y \to HH$ models were selected as a representative set of the general $A \to BC$ search, covering a broad range of expected sub-jets in the two jets of the final state (2/2, 3/3, 2/4 and 6/6, respectively). The masses were chosen to be $m_A = 3$ TeV for all benchmark models and $m_Y = m_{Y'} = 170$ GeV as well as $m_{B'} = m_R = m_H = 400$ GeV. Except for W_{kk}, no dedicated search was previously done to constrain these models.

For each considered benchmark model, all the anomaly detection methods were trained and the respective selection efficiencies were evaluated. For the weakly supervised methods, this was done using the efficiency scanning procedure discussed in subsection 7.7.2, including all systematics outlined in section 7.9. The sensitivity of each method is quantified by the cross section that would be needed to achieve an excess of 3 σ or 5 σ expected significance. The sensitivity was also compared to an inclusive search as well as a traditional cut-based approach that targets jets in the final state with two or three subjets. The result of this study can be seen in Figure 7.31. Considering this plot it can be observed that for all benchmark models, the anomaly detection methods show a notable improvement in sensitivity compared to both the inclusive search and the cut-based approach. The largest improvement can be seen for the $Y \to HH$ model, which can be attributed to the targeted substructure

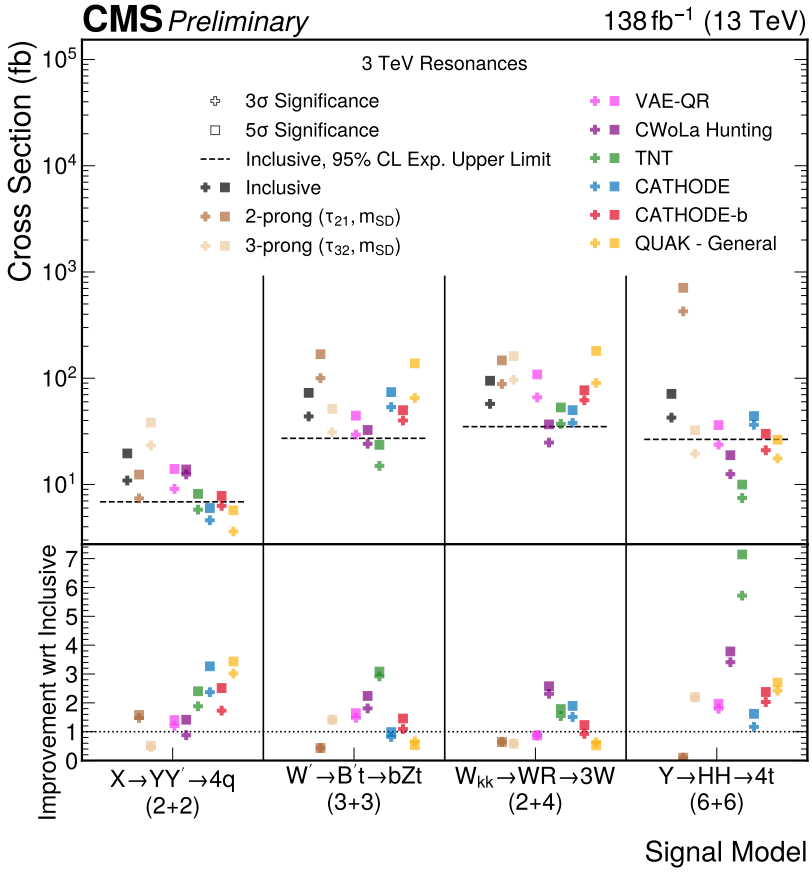


Figure 7.31: Sensitivity comparison of the employed anomaly detection methods for a representative set of considered signal models at $m_A = 3\,\text{TeV}$. For each method and signal model, the cross section needed to achieve a significance of $3\,\sigma$ (cross markers) and $5\,\sigma$ (square markers) are shown. For reference, the corresponding values of an inclusive search (black) as well as standard cut-based approaches (brown and light brown) are shown. The lower panel of the plot shows the improvement with respect to the inclusive value.

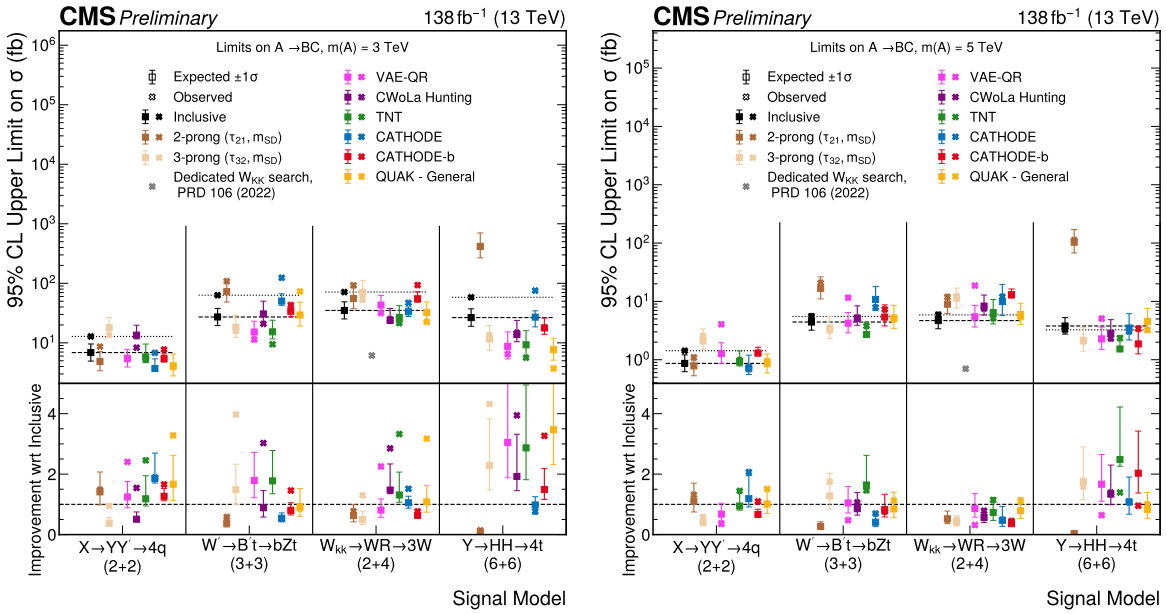


Figure 7.32: Upper limit at 95 % confidence level for the chosen benchmark $A \to BC$ models. The limits are shown for masses $m_A = 3\,\text{TeV}$ (left panel) and $m_A = 5\,\text{TeV}$ (right panel). For reference, the limits obtained for an inclusive search (black) as well as classic cut-based approach based on substructure variables (brown and light brown) are shown. The observed limit from a previously done model-specific analysis in CMS for the W_{kk} signal model is shown in grey.

anomalies being particularly present in this case.

Additionally, it can be noted that for each signal model there exists at least one method that achieves an expected significance at the $5\,\sigma$ level for a cross section equal or lower than the upper limit of the inclusive search. A comparison of this study with Figure 7.28 reveals different results regarding the performance of CATHODE-based methods. For the $X \to YY'$ case, CATHODE still shows the optimal improvement of all the unsupervised methods and also CATHODE-b does improve on the inclusive search value. However, the good relative performance of CATHODE-b on the W_{kk} model could not be retained. This effect can be attributed to the inclusion of systematic uncertainties, which significantly affect the resulting signal efficiencies. CATHODE-b in particular has to additionally account for the b-tagging systematic. However, the CATHODE-based methods still show a significant improvement over the inclusive search for most of the signal models and for $X \to YY'$ in particular.

Exclusion limits were computed for the benchmark signal models, again using the previously outlined limit setting procedure, including systematic uncertainties. The obtained limits are compared amongst methods as well as to those of an inclusive search and a traditional cut-based approach. Additionally, the exclusion limit of the model-specific W_{kk} search is compared for reference. In Figure 7.32, these limits are shown for resonance masses $m_A = 3\,\text{TeV}$ and $m_A = 5\,\text{TeV}$. Similar to the previous study it can be seen that the employed anomaly detection methods improve on the limits obtained by an inclusive search as well as a traditional cut-based approach for all the signal models. Generally, the improvement is higher for the 3 TeV mass point than for the 5 TeV mass point, as most methods use tighter selections at lower masses that causes the QCD background to be suppressed more effectively. It can also be seen that the dedicated search results in a significantly more stringent limit than the more model-agnostic methods, which is expected. CATHODE in particular shows signif-

Signal Model (m_A = 3 TeV)	**m_B/m_C (GeV)**	**Method**	**Exp. (Obs.) Limit (fb)**	**Improvement w.r.t Inclusive**
$Q^* \to qW'$	m_q/25	CWoLa	61.1 (30.1)	0.3
$Q^* \to qW'$	m_q/80	Cathode	50.0 (95.2)	0.4
$Q^* \to qW'$	m_q/170	VAE-QR	52.5 (37.5)	0.4
$Q^* \to qW'$	m_q/400	CWoLa	45.8 (24.3)	0.5
$X \to YY'$	25/25	Cathode	8.0 (9.9)	0.9
$X \to YY'$	25/80	Cathode	7.6 (13.2)	0.9
$X \to YY'$	25/170	Cathode	10.3 (18.4)	0.7
$X \to YY'$	25/400	VAE-QR	13.6 (12.5)	0.6
$X \to YY'$	80/80	Cathode	4.2 (8.0)	1.6
$X \to YY'$	80/170	Cathode	5.7 (11.4)	1.2
$X \to YY'$	80/400	Cathode	6.0 (7.3)	1.2
$X \to YY'$	170/170	Cathode	3.7 (6.8)	1.9
$X \to YY'$	170/400	VAE-QR	4.4 (4.0)	1.7
$X \to YY'$	400/400	VAE-QR	2.1 (1.9)	4.2
$W' \to B't \to bZt$	25/m_t	TNT	25.2 (17.4)	1.5
$W' \to B't \to bZt$	80/m_t	TNT	22.3 (14.6)	1.5
$W' \to B't \to bZt$	170/m_t	TNT	12.2 (7.3)	2.1
$W' \to B't \to bZt$	400/m_t	VAE-QR	15.2 (11.4)	1.8
$W_{kk} \to RW \to 3W$	170/m_W	TNT	25.1 (20.1)	1.4
$W_{kk} \to RW \to 3W$	400/m_W	CWoLa	23.8 (25.0)	1.5
$Z' \to T'T' \to tZtZ$	400/400	QUAK	28.3 (13.9)	2.7
$Y \to HH \to 4t$	400/400	QUAK	7.7 (3.7)	3.5

Table 7.10: Expected and observed 95 % CL upper limits for different signal models at a resonance mass of 3 TeV on the respective cross section. Presented are only the results from the anomaly detection method yielding the best performance. For reference, the improvement related to the expected limit of an inclusive search is also shown.

icant improvements with respect to the inclusive limit for the $X \to YY'$ model, obtaining the overall best improvement on the observed limit at the 5 TeV mass point. For other benchmark signal models, however, other unsupervised methods generally yield better results.

In addition to the signal models chosen as benchmarks, tables containing the limits on the other signal models and mass points can be found in Table 7.10 and Table 7.11 for resonance masses of 3 TeV and 5 TeV, respectively. Shown are only the results of the model that yields the best improvement compared to an inclusive search. The entire list of results only for Cathode and Cathode-b is shown in Appendix E. From the tables it can be seen that for most of the signal models, the limits obtained from the anomaly detection methods yield an improvement over the inclusive result. However, there is still a significant number of signal models where no such improvement is achieved even by the best-performing method. In many cases, this occurs when one of the two jets in the final state does not contain substructure that an anomaly detection method can exploit. One example is the $Q^* \to qW'$ model, where one of the jets is a quark jet. Another case resulting in jets without substructure is when one of the daughter particle masses is very low, leading to highly boosted jets where the substructure

Signal Model (m_A = 5 TeV)	**m_B/m_C (GeV)**	**Method**	**Exp. (Obs.) Limit (fb)**	**Improvement w.r.t Inclusive**
$Q^* \to qW'$	m_q/25	QUAK	3.5 (3.1)	0.7
$Q^* \to qW'$	m_q/80	QUAK	3.2 (2.8)	0.8
$Q^* \to qW'$	m_q/170	QUAK	3.3 (3.6)	0.8
$Q^* \to qW'$	m_q/400	QUAK	3.9 (9.9)	0.7
$X \to YY'$	25/25	QUAK	1.7 (1.6)	0.5
$X \to YY'$	25/80	QUAK	1.3 (1.3)	0.7
$X \to YY'$	25/170	QUAK	1.1 (1.1)	0.8
$X \to YY'$	25/400	VAE-QR	1.0 (3.4)	0.9
$X \to YY'$	80/80	TNT	1.1 (1.2)	0.8
$X \to YY'$	80/170	QUAK	0.9 (1.0)	0.9
$X \to YY'$	80/400	VAE-QR	0.9 (3.0)	0.9
$X \to YY'$	170/170	CATHODE	0.7 (0.7)	1.2
$X \to YY'$	170/400	VAE-QR	0.7 (2.3)	1.2
$X \to YY'$	400/400	VAE-QR	0.4 (1.1)	2.3
$W' \to B't \to bZt$	25/m_t	TNT	4.4 (6.2)	1.3
$W' \to B't \to bZt$	80/m_t	TNT	3.9 (5.7)	1.4
$W' \to B't \to bZt$	170/m_t	TNT	2.8 (3.5)	1.6
$W' \to B't \to bZt$	400/m_t	TNT	2.7 (3.8)	1.6
$W_{kk} \to RW \to 3W$	170/m_W	TNT	6.1 (7.2)	0.8
$W_{kk} \to RW \to 3W$	400/m_W	VAE-QR	5.4 (18.6)	0.9
$Y \to HH \to 4t$	400/400	TNT	1.5 (2.3)	2.5

Table 7.11: Expected and observed 95 % CL upper limits for different signal models at a resonance mass of 5 TeV on the respective cross section. Presented are only the results from the anomaly detection method yielding the best performance. For reference, the improvement related to the expected limit of an inclusive search is also shown.

cannot be resolved. This can be seen from the tables in the $X \to YY'$ signal model where m_Y masses are small.

CATHODE shows a high sensitivity for the $X \to YY'$ signal model, yielding the best limits for most of the mass combinations in the 3 TeV case. For the 5 TeV mass point, however, it results in the best limit only once, namely in the case $m_Y = m_{Y'} = 170\,\text{GeV}$. In general, the 5 TeV is challenging for all methods, which is shown by only few of the limits being improved and if so, the improvement is often marginal. As discussed previously, the reason for this phenomenon is probably again the less tight selection efficiency used by many methods for this mass point. CATHODE-b is not resulting in the best limit for any of the considered models and mass points, apparently being outperformed by other methods for signal models containing b jets. In general, being able to set limits on this large number of different signal models in one analysis, covering decays resulting in jets with various substructure patterns and mass combinations, is a powerful display of the capabilities of unsupervised methods for anomaly detection. This first-of-its-kind analysis constitutes a significant milestone in the application and development of data-driven searches for new physics. Leveraging the potential of novel machine learning-based approaches, it was possible to scan vast regions of phase space for anomalies in data in a largely model-agnostic way. Additionally, limits were set for several, previously unconstrained signal models and different mass hypotheses at once, improving the inclusive limit significantly for some of them. Anomaly detection methods based on density estimation such as CATHODE played an important role in this analysis, showing the highest sensitivity and yielding the best limits at most mass hypotheses for the $X \to YY'$ model at 3 TeV. However, several limitations still remain for weakly supervised methods. Two key problems, namely the impact of uninformative features on the performance as well as the sculpting in the presence of highly correlated features, have been considered in more detail. This lead to significant enhancements and while the respective studies were not finished in time to be applied to CMS experimental data, they have been studied using the LHCO R&D data set. The respective results are discussed in the following chapter.

Chapter 8

Improvements to the CATHODE Algorithm

8.1 Using Tree-Based Algorithms for Weak Classification

The work presented in this chapter has been previously published in [216], in collaboration with Thorben Finke, Marie Hein, Gregor Kasieczka, Michael Krämer, Alexander Mück, Parada Prangchaikul, David Shih and Manuel Sommerhalder. The figures and the written content closely resemble or match the information found in this article. My contribution to the publication consists of producing the analysis code for model comparison and rotational invariance studies, as well as using this code to assess correct reproducibility of performance studies. Furthermore, paper writing and proof-reading and the assessment and discussion of key research questions and hypotheses during the development of the method with the mentioned collaborators. I also did major writing for a derivative submission, which was accepted to the Machine Learning and the Physical Sciences workshop at the Thirty-seventh Annual Conference on Neural Information Processing Systems (NeurIPS) [217]. The model selection studies are not part of the original publication and are my own work entirely.

It has been shown that CATHODE does not only achieve state-of-the-art performance in resonant anomaly detection, but also almost saturates the optimal limit of an unsupervised search defined by the IAD. While the development of CATHODE constitutes a significant advancement of data-driven methods for anomaly detection, several challenges remain.

In section 7.1, the performance of CATHODE was demonstrated on the LHCO R&D data set using a single signal model with two two-pronged jets in the final state for the benchmark. However, in a realistic search for new physics, a plethora of signal models with profoundly different signatures could exist. Additionally, not only was the analysis limited to this specific model, but the CATHODE input features were chosen such that they contain significant information for the signal model in question. Both the subjettiness-based as well as the mass-based features contain information about the jet substructure and are therefore well suited for detecting the signal. In particular, the optimal subjettiness ratio for distinguishing the two-pronged jets in the final state of the signal model from the one-pronged QCD background jets, τ_{21}, was used.

Since the signal model is unknown in an entirely model-agnostic search, it is also unknown what features would be informative to detect it. Therefore, it is crucial to use as many features as possible to ensure sensitivity to any kind of anomaly. Furthermore, it can be expected in such a scenario that only

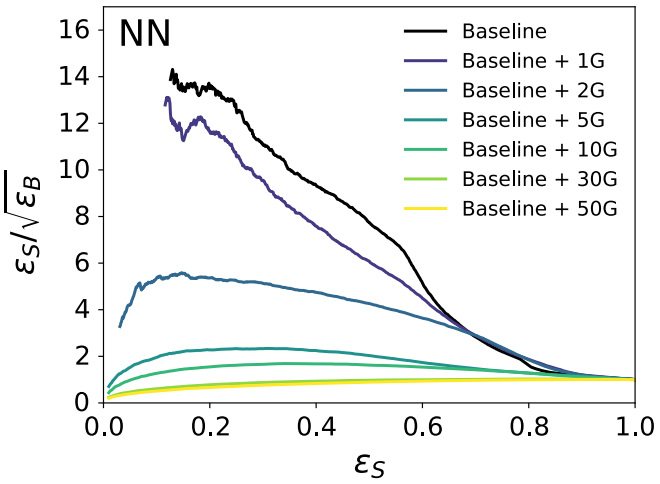


Figure 8.1: The effect of uninformative features on the performance of a neural network (NN) – based idealized anomaly detector. SIC curves are shown for the baseline feature set and for the feature set with noise added, where nG refers to the addition of n independently sampled Gaussian features.

a small fraction of these features will contain information to distinguish anomalous events from the background. At the same time, the inclusion of a large number of uninformative features significantly increases the complexity of the anomaly detection, since it introduces additional noise to the already noisy weak classification task.

8.1.1 Impact of Uninformative Features

The impact of uninformative features on weak classification was studied by using the features in the original LHCO R&D benchmark for CATHODE and then adding features that were generated by sampling randomly from a standard normal distribution. These Gaussian features constitute a proxy for the inclusion of features without any information regarding the discrimination of signal and background events.

In order to eliminate the impact of imperfections in the density estimation, an idealized anomaly detector (IAD), as described in subsection 6.4.3, was used for this study. In the IAD, the background template is not estimated by a generative model, but instead events from the true background distribution inside the signal region are used in the classifier training. Thus, the IAD yields the performance in the limit of perfect density estimation. Since the density estimation and interpolation of the background template into the signal region do not need to be considered in the idealized scenario, the findings of this study extend to any weakly supervised anomaly detection method.

For the model, a fully connected neural network model was trained, using the exact same hyperparameters as in the original CATHODE study [150] (see Table 7.2 for reference). Jets in the event are sorted by mass, such that the heaviest jet is referred to as the first jet, the second heaviest jet as the second jet and so on. Also, the "baseline" features used in this study are chosen the same as in the original CATHODE work, namely the mass value of the second jet, m_{J_1}, the mass difference between the first and second jet, Δm_J, as well as the subjettiness ratios τ_{21,J_1} and τ_{21,J_2} of the first two jets. For the training and validation sets, a 50%/50% split of approximately 272 000 background events and 120 000 data events in the signal region were used. The final performance is estimated on a separate

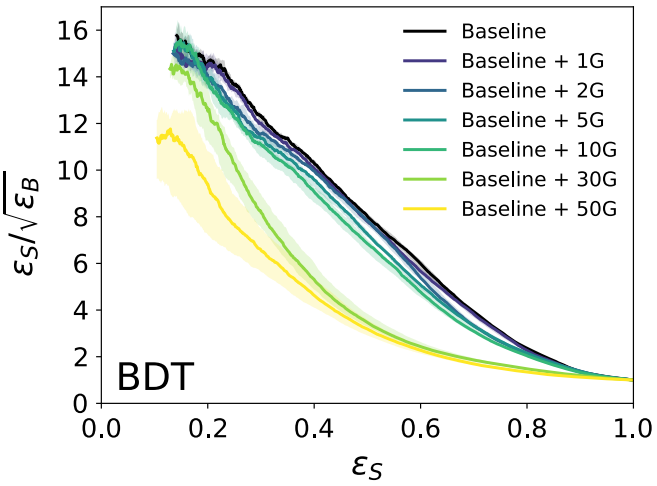


Figure 8.2: The results of the same study as shown in Figure 8.1, but using a histogrammed gradient boosted tree (BDT) – based classifier instead of a neural network. The solid lines of the SIC curves correspond to the median performance of 50 independent classifier runs (with the exception of 30 and 50 Gaussian features, where 100 runs were used instead), while the shaded areas describe the inner 68% band of the SIC curve distributions based on these runs.

test set based on the truth-level signal and background labels, which contains 340 000 background and 20 000 signal events from the signal region. Figure 8.1 shows the performance in terms of SIC curves for this study. The "Baseline" refers to the previously mentioned features, while "nG" refers to n Gaussian noise features added. It can clearly be seen that the addition of uninformative features significantly impacts the performance of the DNN-based IAD: Adding only two noise features leads to a decrease of more than a factor two in maximum SIC and adding even more noise makes the SIC curve drop to the performance of a classifier choosing events randomly.

This behaviour constitutes a considerable problem for the use of weak supervision in model-agnostic searches: In essence, it means that weakly supervised methods require specifically selected features that are guaranteed to be informative for a particular signal model or at least for a family of similar signal models. This, of course, strongly conflicts with the objective of being model-agnostic, since the targeted signal model needs to be known for feature selection. One possible way of mitigating this issue would be to test a random subset of all available features and redo the entire analysis for each trial. This approach, however, requires significant computational resources and it is questionable – given the vast amount of both high and low-level features available in a particle physics scenario – that it would be feasible.

To overcome this shortcoming of weakly supervised methods, a classifier is needed that is both performant and highly robust against uninformative features. Most of these methods use deep neural networks (DNNs) for the weak classification task. While deep learning methods achieve state-of-the-art performance in many machine-learning tasks such as computer vision, generative modelling or natural language processing, their relatively poor performance on tabular data is an active area of computer science research [114, 218, 219]. As described in section 5.1, classical machine learning methods based on gradient boosted decision trees (BDTs) often outperform even complex deep learning algorithms such as transformers [99] significantly on tabular data [218]. Additionally, BDT-based

methods were found to be robust against the inclusion of uninformative features in supervised classification. Since tabular data – in form of various high-level features per event – is also used in most weakly supervised methods for anomaly detection, the application of BDTs instead of DNNs to the weak classification task was studied.

To test BDT-based classifiers for weak supervision and investigate the impact of including uninformative features on their performance, the study described above was repeated, using the baseline features and additional Gaussian noise features in an idealized setting was repeated. In particular, the histogrammed gradient boosting (HGB) algorithm based on LightGBM [94] implemented as `HistGradientBoostingClassifier` in the `scikit-learn` package [177] was used. The reason to employ this particular algorithm is two-fold: First, LightGBM typically ranks amongst the best performing algorithms in many tabular data competitions [84]. Second, histogramming the features allows for considerably faster training. As described in chapter 5, decision trees decide at which threshold to split the data into subsequent nodes or leaves by scanning all possible cuts between the unique values for each feature. Since the considered features are based on floating point numbers, the number of unique values in a feature is similar to the number of data points. Therefore, computing the optimal cut based on all data points in each of the features is computationally intensive, especially when the data sets are large, as is frequently the case in a particle physics scenario. Histogramming the features reduces the possible cut values to a significantly smaller set of bin values, which substantially improves the time to train the BDT.

The fast training of BDT-based classifiers also allows for the inclusion of ensembling: Instead of a single training run, 10 classifiers are trained independently, and the mean prediction of all the classifier models is used for the performance evaluation. To further improve training time, early stopping is employed: The binary cross-entropy loss is evaluated for a separate validation data set at each iteration and the training is stopped if validation loss does not decrease within ten subsequent boosting iterations. The final evaluation of the performance is done on another statistically independent test data set, using the model state at the minimum validation loss iteration. The parameters used in the training of the gradient boosted classifier are based on the `scikit-learn` default settings. Dedicated hyperparameter optimization studies were conducted using the `optuna` [220] software package, but revealed no significant performance increase with respect to the defaults. Since the weak classification task is noisy, its outcome is expected to have a high variance, in particular when many Gaussian noise features are included in the training. To study this variance, instead of training a single BDT ensemble, 50 independent ensemble trainings were conducted for the case of up to 10 additional Gaussian features, while for 30 and 50 Gaussian features 100 trainings were run.

The results of the study are summarized in Figure 8.2. Comparing the baseline SIC curves of the BDT and the DNN in the corresponding plot in the left panel, it can be seen that the BDT slightly outperforms the DNN when no entirely uninformative features are present. The maximum SIC that is achieved in this case is around 16 for the BDT, while being 14 in the DNN case. This result confirms the findings in literature regarding the superior performance of gradient boosted classifiers over deep learning methods on tabular data and that they also apply to the weakly supervised regime. The most significant difference between the DNN and BDT results can be observed when comparing the curves when Gaussian noise features are included in the training: The performance of the BDT-based classifiers is barely affected for up to ten noise features, regarding both the median SIC values and their

Name	# features	Features
Baseline	4	$\{m_{J_1},\ \Delta m_J,\ \tau_{21}^{\beta=1,J_1},\ \tau_{21}^{\beta=1,J_2}\}$
Extended 1	10	$\{m_{J_1},\ \Delta m_J,\ \tau_{N,N-1}^{\beta=1,J_1},\ \tau_{N,N-1}^{\beta=1,J_2}\}$ for $2 \leq N \leq 5$
Extended 2	12	$\{m_{J_1},\ \Delta m_J,\ \tau_{N}^{\beta=1,J_1},\ \tau_{N}^{\beta=1,J_2}\}$ for $N \leq 5$
Extended 3	56	$\{m_{J_1},\ \Delta m_J,\ \tau_{N}^{\beta,J_1},\ \tau_{N}^{\beta,J_2}\}$ for $N \leq 9$ and $\beta \in \{0.5, 1, 2\}$

Table 8.1: Content of the different feature sets considered for the respective comparison study.

variance, which is shown in the inner 68%-band as area of reduced opacity in the plot. Starting from 30 Gaussian features, the performance starts to decrease also for the BDT, in particular at higher signal efficiencies, while the variance increases noticeably. However, considering the median performance, much of the original significance improvement can still be retained even in the challenging 50G case, where informative features represent only a small fraction of all features. This is vastly different from the DNN classifier, whose performance reduces to almost no significance improvement with as few as ten Gaussian features.

These results clearly show that using algorithms based on (histogrammed) gradient boosting allows for model-agnostic searches without the need of feature selection, being performant even in challenging scenarios where the majority of features does not contain information to discriminate between signal and background. Additionally, they are significantly faster to train then DNNs, reducing the computational requirements of weakly supervised anomaly detection tasks considerably.

8.1.2 Using Different Physics Features

The study considering the addition of noise features employed a worst-case scenario, since the Gaussian features constitute pure noise and do not contain any information. In a different study, a more realistic approach was investigated: This time, different sets of physics features were tested and the same comparison between DNN and BDT-based classifiers was done. The additional features are divided in three different sets, labeled "Extended Set 1" to "Extended Set 3" and their content is described in Table 8.1. The extended sets contain different features regarding the subjettiness of the jets: For extended set one, additional subjettiness ratios ($\tau_{N,N-1}$) are included alongside the baseline features. Extended sets two and three contain the plain subjettiness values (τ_N) instead of their ratios. In extended set three, subjettinesses for different values of β are also added, which is an angular weight used in the computation of the subjettiness. It should be noted that each extended set contains more information than the previous feature set. Therefore, even if the additional features were not informative with respect to our signal model, the performance of a robust classifier trained on a larger feature set should be at least as good as on the previous feature set.

The performance comparison of the BDT and DNN-based classifiers for the different feature

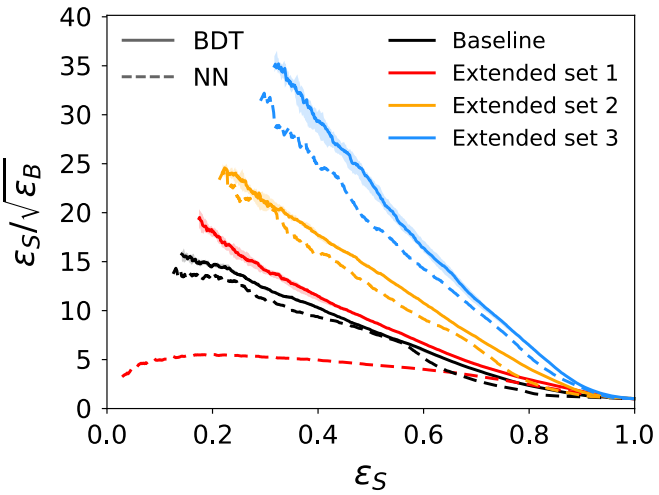


Figure 8.3: The effect of different feature sets on the performance of weakly supervised classifiers. The comparison is made for DNN-based classifiers (dashed lines) and BDT-based classifiers (solid lines). Shown are SIC curves for different sets of input features as described in Table 8.1. For the BDT-based classifiers, 50 independent classifier runs are used and the solid lines correspond to the median performance while the areas of reduced opacity describe the inner 68% band of the SIC curve distributions. For the DNN-based classifiers, only a single training run was used due to limited computational resources.

sets can be seen in Figure 8.3. Except from changing the features used in the training, all training parameters were kept the same as in the previous study. From this plot, it can clearly be observed that DNNs are suboptimal classifiers for weakly supervised anomaly detection: Not only is their overall performance slightly worse compared to the BDT classifiers, but it also depends on the specific feature set used. This can be seen specifically by comparing the Baseline SIC curve of the DNN with the corrseponding curve of extended feature set one: Even though the extended set contains more information than the baseline set, its performance is significantly worse. The reason for this worsening is that τ_{21} already constitutes the optimal feature choice for our signal model, while higher-pronged subjettiness ratios are expected to contain only marginal information for it and act mainly as noise. Therefore, the finding of the previous study extends from pure noise features to actual physics features and underlines that using DNNs in weakly supervised anomaly detection requires model-specific selection of informative features to ensure sensitivity.

For BDT-based classifiers, the case is different: Increasing the information content of the features also increases the model performance. The performance on extended set one also increases slightly with respect to the baseline performance, showing that BDTs are also able to improve when only a modest amount of information is added. This behavior shows that gradient boosted trees are an optimal choice for weakly supervised anomaly detection, both when many informative features are included and in the case where most features act as noise.

8.1.3 Causes of Increased Robustness against Noise

Given the markedly different results when comparing DNNs and BDTs, further studies assessing the cause of this difference were conducted. Three main factors were identified that lead to a performance

improvement of histogrammed gradient boosting algorithms compared to DNNs:

1. Due to the significantly faster training time of (histogrammed) BDTs, it is computationally feasible to train an ensemble of many independent classifier models, which greatly improves the performance in the presence of noise compared to individual models or ensembles based on the model state at different epochs of a single model instance.

2. The use of histogramming reduces the noise of the "background-vs-background" part of the weak classification task, since small event-by-event differences are averaged out over the more coarse, binned structure of the feature histograms. This effect has not been studied in this work, but it represents a viable explanation for the results attained when comparing the performance of different tree-based algorithms.

3. DNNs are invariant under the rotation of features (e.g. by applying a random $SO(N)$ matrix to the feature vectors of each sample). Therefore, a DNN has to learn the optimal orientation of the data in addition to the patterns separating signal and background events. This is a significantly more difficult task than the classification alone and an addition of noisy features further complicates this task, as it increases the dimensionality of the problem. This increased complexity leads to a reduced performance of DNNs on tabular data, in particular in a noisy, weakly supervised regime.

Each of these three factors and the corresponding studies will be discussed in further detail in this section.

Ensembling

One key difference between the methodology used in the initial work describing CATHODE [150] and the BDT-based studies is the way ensembling was performed. Originally, ensembling for CATHODE was done by taking the average predictions of the models corresponding to the ten epochs that achieved the lowest values of the validation loss. However, this ensemble is still based on the same model initialization and thus carries the same inductive bias. While it has shown to reduce the variance of the final result, the overall performance did not change significantly based on whether or not ensembling was used. Given the short training time of BDTs, it is computationally feasible to create an ensemble of entirely re-trained classifier models and average over their respective predictions. This kind of ensembling was used in all of the previously described studies for BDT-based classifiers. The effect of using the classifier ensembling can be seen in Figure 8.4.

From this figure, the advantages of an ensemble of individual classifier models can clearly be seen: The ensemble outperforms all classifier models for the BDT and is almost as high as the best model for the DNN. Additionally, the median performance shows a considerably smaller significance improvement compared to the ensemble, which leads to the conclusion that ensembling the model predictions improves the performance and not only the variance of the final result as is the case for epoch ensembling. It is also again the case that the NN models perform significantly worse than the BDT models, with only few models showing a significance improvement larger than one. Interestingly, several BDT models with rather low significance improvement values can also be seen in the

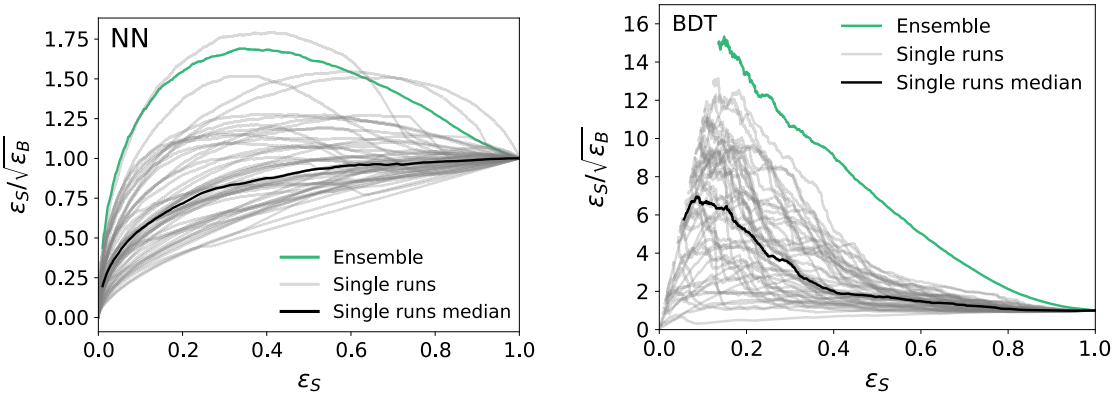


Figure 8.4: The effect of ensembling on the performance of weakly supervised classifiers, using the baseline features and ten additional Gaussian noise features (note the different y-axis scales). The comparison is done between DNN-based classifiers (left panel) and BDT-based classifiers (right panel). Shown are SIC curves of individual classifier models in light grey, as well as the median of these SIC curves in black. The green curves show the performance of the entire ensemble, i.e. the average prediction over all models constituting the ensemble.

right panel plot, which shows that also BDT-based classifiers vary significantly in performance under noisy conditions.

Given these results, the reason how the ensembling improves the individual model results remains unanswered. To better understand this phenomenon, the distributions of the individual and ensemble model predictions were compared, which can be seen in Figure 8.5. In this figure, an ensemble of

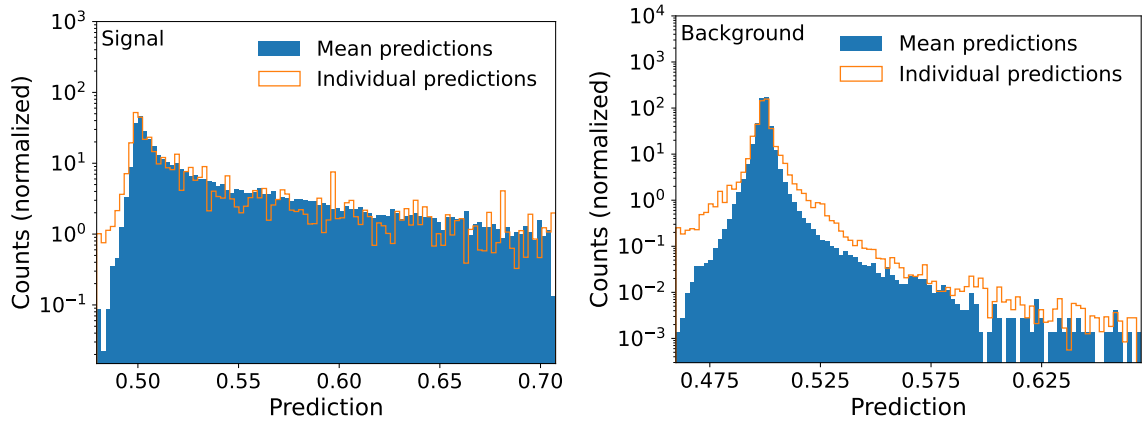


Figure 8.5: Normalized prediction histograms for signal (left) and background events (right) for ten independent BDT classifier runs. The comparison is done between the mean ensemble predictions (blue) and the predictions for each individual model inside the ensemble (orange).

ten independent BDT models was trained and then two evaluations based on the test set events were done: At first, each individual model was evaluated separately and the predicted values for all models were histogrammed, which is shown in orange. Then, the ensemble mean was computed over the predictions of the ten models and the mean predictions were also histogrammed, as shown in blue. This was done separately for signal events (left panel) and background events (right panel). The figure clearly shows the reason for the overall performance increase: For signal events, some models predict anomaly scores lower than the random threshold of 0.5, but when considering the ensemble average,

these events get a significantly higher score, leading to fewer events with random or lower than random predictions and a more pronounced tail towards higher (i.e. more anomalous) values. Similarly, for background events, individual models predict high scores for some events but the ensemble mean reduces the high (and low) tails of predicted values considerably, leading to a distribution which is more contained around the random prediction value. In summary, ensembling leads to both fewer false negatives and fewer false positives, increasing the overall performance compared to individual models.

Histogramming

A second factor explaining the improved robustness with respect to noise is that the HGB classifier uses histograms of the features to define the split at each node. This is done to decrease the number of possible split values and significantly speed up the training. Additionally, this key difference compared to the other investigated algorithms could also be a major cause for the improvement of the robustness against adding noise. Since the signal fraction in the data is small, the largest part of the weakly supervised classification lies in distinguishing the background template samples from the background samples in the data. In an ideal case, their distributions align perfectly and the prediction distribution is a sharp peak around the random prediction value of 0.5, with a small width due to the statistical variance of the different samples.

When several Gaussian noise features are added randomly to these events, it is unlikely that similar events from data and from the background template receive also similar noise values in all of the additional dimensions. Therefore, adding noise causes the emergence of patterns that reflect differences between background events from the two samples, which leads to more frequent misclassifications and thus an increase in the width of the prediction distribution. In turn, the background rejection is decreased which causes a decrease in performance. Naturally, this effect is largest if the noise of each individual sample is considered in each feature for each node split, as it is typically done in tree-based algorithms. However, when histogramming the features – and in particular the noise features – individual noise patterns get averaged over the bins. This can significantly reduce misclassifications due to small noise effects while still being able to capture the larger-scale patterns that govern the differences in signal and background. While these effects have not been studied in detail in the foundational work of this chapter [216], their conceptual discussion represents a possible explanation of the significant differences in noise robustness between tree-based algorithms.

Rotational Invariance

As previously discussed, the better performance of tree-based models compared to deep learning algorithms on tabular data is an active area of computer science research [114, 218, 219]. One reason that is frequently mentioned in literature is the robustness of tree-based algorithms against features that are uninformative regarding the classification task. One hypothesis for this behaviour that has been discussed in literature is that DNNs are invariant under rotation of the features [114] and therefore have to learn the optimal orientation of the features in addition to extracting patterns that distinguish the different classes. This of course is a significantly more difficult problem compared to the classification alone, in particular when many additional dimensions that carry no information are added. Tabular data, however, is not rotationally invariant and the original orientation of the data is therefore

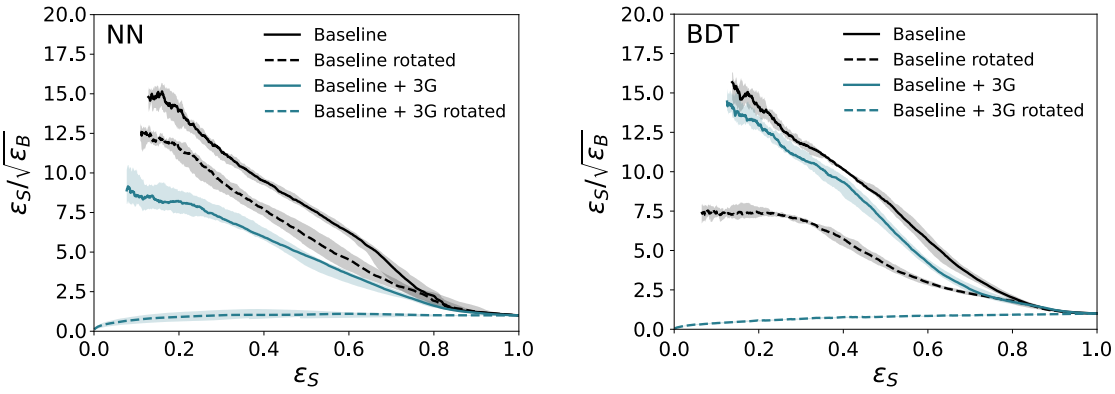


Figure 8.6: Performance comparison of rotated and non-rotated data sets. SIC curves are shown for an NN-based classifier (left panel) as well as for the histogrammed gradient boosting classifier (right panel). The comparison is done for the baseline feature set (black) as well as the baseline feature set with three Gaussian noise features added (turquoise). For both data sets a random rotation was applied and the classifiers were retrained using the rotated features. The results for the rotated features are shown by the dashed lines. In all plots, solid colours represent the median performance of ten independent classifier retrainings, while shaded areas describe the inner 68% of the respective SIC curve distributions.

already the optimal one. Thus, applying a deep learning classifier to a tabular data problem means applying a rotationally invariant learner to a non rotationally invariant data set. Tree-based algorithms on the other hand are not rotationally invariant and therefore only operate on the optimal, original orientation of the input data and focus only on extracting the relevant patterns to distinguish signal from background.

The features from the LHC Olympics data set that were used in the previous studies are also in a tabular data format: Each event that is classified into data and background can be considered a row while the per-event features constitute the columns. Therefore, the aforementioned findings in computer science literature should also extend to this data set. However, computer science research regarding the robustness against uninformative features and the rotational invariance of classifiers mainly focuses on the fully supervised scenario while the behaviour in a weakly supervised setting has yet to be studied. Therefore, another study was performed using the baseline feature set as well as the baseline feature set with three Gaussian noise features added. First, for each data set a DNN classifier as well as an HGB classifier was trained. Then, the features of both data sets were rotated by applying a random rotation matrix drawn from the Haar distribution (a uniform distribution on $SO(N)$) using the `scipy` software package [221]. New DNN and BDT-based classifiers were then trained and evaluated on the rotated data sets. The results of this study can be seen in Figure 8.6.

The figure shows that the findings regarding the rotational invariance of classifiers in fully supervised tasks largely extent to the weakly supervised setting: The performance of the non-rotated baseline feature set is similar to the rotated feature set for the DNN-based classifier, while it is significantly worse for the tree-based algorithm, with about only half of the original performance retained. Notably, the HGB classifier shows worse performance than the DNN in this case. When Gaussian noise features are added, the rotation further distorts the information contained in the original features and both the DNN and the HGB classifier break down entirely to random performance.

To summarize, one reason for the low robustness of DNN-based classifiers to noise in tabular data tasks is their rotational invariance, which was shown to extend to weakly supervised scenarios. Tree-based algorithms, however, only work on the original, physically meaningful orientation of the data and therefore retain more of the original performance when noise features are added.

8.1.4 Performance Comparison of Different Tree-Based Algorithms

In the studies discussed so far, performance comparisons were exclusively done using DNNs and the HGB algorithm. To assess whether the improved robustness against the presence of noise is a general feature of tree-based classifiers or whether it is specific to the HGB classifier in particular, another comparison study has been conducted: Three different tree-based algorithms were trained, using the baseline feature set and the same feature set with ten Gaussian noise features added. The algorithms were chosen to be representative of a broad range of tree-based classifier methods: A random forest classifier, which is the simplest of the tree-based ensemble classifiers, not making use of advanced gradient boosting techniques. An AdaBoost classifier [92], which employs a slightly different boosting approach compared to modern gradient boosting methods (assigning higher weights to misclassified events rather than minimizing a loss function by predicting residuals of the previous iteration) and finally a ROOT TMVA BDT [222], which is frequently used in Particle Physics use-cases and is also based on AdaBoost. For the ROOT BDT, the default hyperparameter settings of the software version 6.28.4 were used and the HGB classifier parameters were kept the same as in the previous studies. For the AdaBoost and random forest (RF) classifiers, an optimization study was done using the `optuna` [220] software package and the baseline feature set. The achieved optimal hyperparameters can be seen in Table A.2 for AdaBoost and Table A.1 for the RF. The variance of the classifier results was assessed using $N = 10$ independent retrainings as opposed to $N = 50$ which was used before. Some of models took significantly longer to train and therefore retrainings had to be reduced to meet the computational resource constraints.

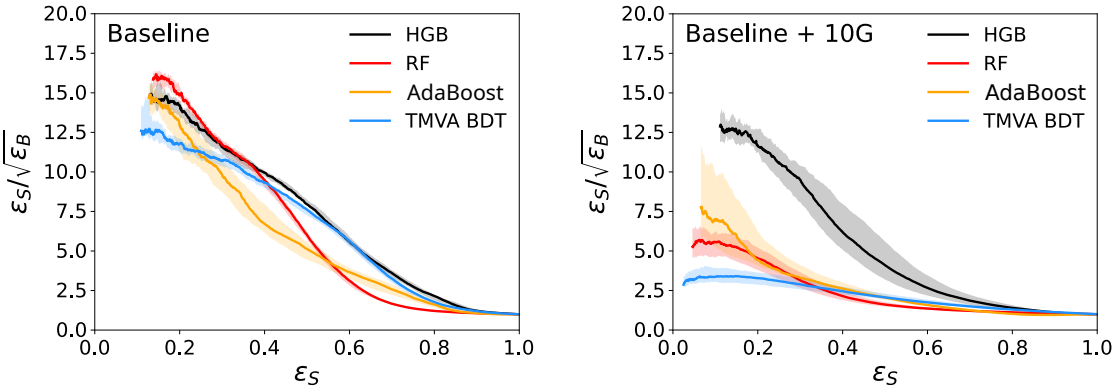


Figure 8.7: SIC curve comparison for different tree-based classification algorithms. Performance is compared for histogrammed gradient boosting (HGB), random forest (RF), AdaBoost and the ROOT TMVA BDT algorithms. Results are shown for the baseline feature set (left panel) as well as the baseline feature set with ten Gaussian noise features added (right panel). Solid lines correspond to the median performance of ten independent classifier ensemble trainings while shaded areas correspond to the inner 68% of the SIC curve distribution of said ensembles.

The comparison of the SIC curves of the discussed algorithms together with the SIC curve of

the histogrammed gradient boosting classifier can be seen in Figure 8.7. Considering the baseline feature set, it can be seen that the performance of the different tree-based classifiers is similar, in particular for low signal efficiencies where the error bands of most algorithms overlap. The maximum achieved significance improvement varies between 12.5 and 16. Interestingly, the simple random forest classifier shows the best performance amongst tree-based models. For higher signal efficiencies, however, the ROOT TMVA BDT as well as the HGB classifier outperform the other models.

When adding Gaussian noise features, the performance of the different algorithms changes significantly: While the HGB classifier retains much of its original singificance improvement with a median maximum SIC of about 13, the other algorithms see a considerable performance drop with only half of the original performance at best. From the algorithms other than the HGB classifier, the AdaBoost classifier shows the best median performance, but also the variance of the results increases significantly compared to the baseline feature set. One possible hypothesis for this increase is based on the hyperparameter choice: As mentioned in subsection 5.2.3, AdaBoost models consists of an ensemble of many weak learners, which are often chosen to be single-node decision trees, so called "decision stumps". However, using the `optuna` package for hyperparameter optimization, a more complex architecture was preferred for the individual decision trees. While this can lead to a better performance, it also results in overfitting, as described by the bias-variance-tradeoff: Increasing the complexity of a model generally reduces the bias, since more complex functions can be learned. On the other hand, the increased complexity makes it easier for the noise of the training data to be memorized by the algorithm, leading to poor and varying performance on an independent test set, which is further amplified by the presence of Gaussian noise features. In this study, each retraining is based on a different selection of training and validation set events, while the test set is kept constant. Due to the high complexity of the AdaBoost model, the performance varies considerably for different training/validation set assignments, as is expected.

Interestingly, the AdaBoost-based ROOT BDT does not show a similar increase in variance, possibly due to the lower complexity of the model, which is limited to a maximum tree depth of three, while the AdaBoost model is unbound in depth and is only limited by the maximum number of leafs, which is set to 31. However, the ROOT BDT also has a significantly lower performance compared to the AdaBoost model, with a maximum median significance improvement of a factor of three. This comparison between the two AdaBoost-based models shows that AdaBoost can still retain some performance in the presence of noise when the model is complex enough, but at the cost of considerably increased variance.

The random forest classifier performance lies between the AdaBoost-based models in the presence of noise. The error bands also increase in size, but are still not as large as for the complex AdaBoost model. Except for the lowest signal efficiencies and considering the error bands, the performance of the AdaBoost and random forest models can be considered similar. Taking into account how the algorithms work – as described in section 5.2 – this behavior is expected. In a random forest, the classification decision is achieved by the majority vote of the individual trees in the forest. Due to the weakly supervised nature of the task and the significant number of noise features added, most events will be classifed as signal or background randomly. Some of the most anomalous signal events as well as the least anomalous background events, however, might consistently receive a slight majority vote to actually be signal and background respectively, which causes at least a critical amount of

the events to be correctly classified and some of the original performance to be retained.

For AdaBoost-based classifiers, the situation is expected to be similar to the random forest in a weakly supervised setting: AdaBoost typically assigns two sets of weights, one to increase the importance of previously incorrectly classified samples (Equation 5.14) and one to increase the weight of an individual learner if its classification error was low (Equation 5.11). In a noisy classification task, most samples will randomly be classified as being either signal or background by the subsequent individual learners, which means that their weights w_i will also be similar. Therefore, the sample weights will have only a marginal impact on the final classification result.

The same is true for the weight of the individual learners γ_j: If one tree in the ensemble shows a good performance, the weight assigned to it in the final weighted sum over the ensemble is high and when it shows a poor performance, the respective weight will be low. Again, due to the noisy nature of the classification tasks, most individual trees will select random regions of phase space, which leads to similarly low weights for all the trees in the ensemble. Since both sample and individual tree weights are therefore expected to have only a marginal impact on the final classification result. Thus, AdaBoost-based classifiers are expected to behave like random forests in a weakly supervised scenario and differences in performance can mainly be attributed to the hyperparameters of the individual weak learners as well as implementational details such as the use of bootstrapping for samples and/or features.

Given the results from the previous discussion, it seems that robustness with respect to noise in weakly supervised classification tasks is not a feature of tree-based algorithms in general. However, as shown before, the HGB algorithm consistently outperforms other algorithms even if the vast majority of features is pure Gaussian noise. Compared to the other investigated tree-based models, two key algorithmic differences exist with respect to the HGB classifier.

First, the HGB classifier uses gradient boosting instead of the adaptive boosting method that is the namesake of the AdaBoost algorithm. Therefore, it directly optimizes the loss function, which is the binary cross-entropy of the data versus background classification task. Since the loss is directly related to the Neyman-Pearson optimal likelihood ratio, its iterative minimization should continuously improve the performance of the model in the final signal vs background evaluation as long as the model is not yet in the overfitting regime. Second, the HGB classifier uses a histogrammed version of the features to determine splits when building the tree. As discussed in subsection 8.1.3, this leads to a reduction of the individual event-by-event differences due to the noisy features and therefore improves the overall classification result.

To summarize, in a weakly supervised scenario, the benefit of the adaptive boosting paradigm is neutralized due to the significant noise in the classification task and the model is largely reduced to a random forest–like model, while gradient boosting still iteratively improves on a loss that constitutes a valid proxy for actually discriminating signal from background.

8.1.5 Sensitivity

In the previous studies, the fraction of signal events within the signal region window was fixed to about 0.6 %, which corresponds to a statistical significance of $2.2\,\sigma$. While the behaviour under the addition of noise strongly favours the use of tree-based classifiers compared to DNN-based classifiers, it is also important to consider the difference in sensitivity, i.e. how the performance behaves with

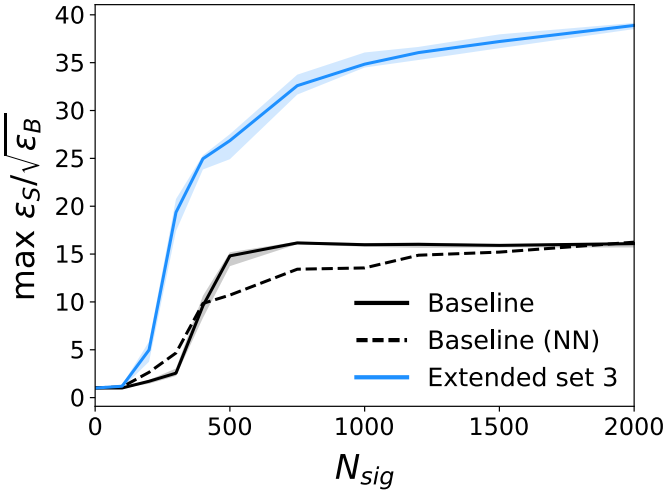


Figure 8.8: Maximum SIC values for different amounts of injected signal. The comparison is done for both the baseline feature set (black) and the extended feature set 3 (blue). The dashed line shows the performance of a DNN classifier on the baseline feature set for comparison.

varying amounts of signal. To study this, both DNN and BDT-based classifiers were trained using different amounts of signal events, while the number of background events in the signal region was kept constant at the previously used value of approximately 120 000. For the comparison between DNN and BDT-based classifiers, the baseline feature set was used in order to focus on sensitivity in an idealized scenario without additional noise features. To also test the impact of increased information content on sensitivity, the HGB classifier was trained on the extended feature set three, again for varying amounts of signal. The number of injected signal events has been varied using the following values: The result can be seen in the left panel of Figure 8.8.

For both the HGB and the DNN classifiers, a significant increase in sensitivity can be observed between 400 and 500 events, which corresponds to a significance in the signal region of 1.15 σ and 1.44 σ, respectively. However, the tree-based classifier outperforms the DNN-based classifier for a wide range of signal injections, except for the very lowest ones, where performance is similar. It can also be observed that the sensitivity also depends on the information content present in the features: For the extended feature set three, a sharp performance increase already takes place much earlier compared to the baseline feature set, namely between 200 and 300 injected signal events (corresponding to 0.58 σ and 0.87 σ, respectively). In general, this feature set also achieves significantly higher SIC values over the entire range of injections compared to the baseline.

Another key question regarding the sensitivity of the different classifier models is whether their performance is mainly a function of the number of signal events present in the data, or whether it is also depending on the overall amount of data available. To study this further, the HGB classifier was trained not only for varying amounts of signal events, but also for different numbers of background events in the signal region. The results of these study are summarized in the right panel plot of figure Figure 8.9. The individual cells in this figure show the (colour-coded) maximum achieved SIC value for an HGB classifier trained on the corresponding amount of signal and background events. Since both numbers are varied, their change is reflected in the statistical significance (shown on the y-axis) and the number of background events (shown on the x-axis). From this figure, it can be seen that

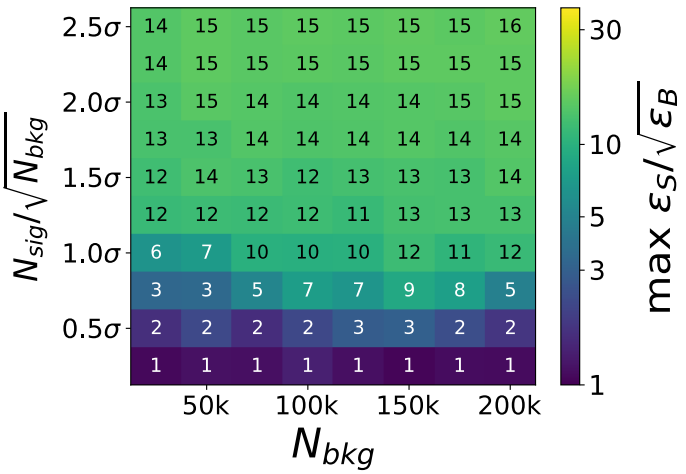


Figure 8.9: Maximum SIC values achieved when varying both the number of injected signal events as well as the number of background events. Results are compared for different values of the statistical significance (y-axis) and the number of background events in the signal region (x-axis).

the maximum achieved significance improvement is mainly a function of the significance of the input data set. This is an important finding also for model-agnostic searches in future Particle Physics experiments, since collecting more data will increase their sensitivity.

8.1.6 Model Selection

When training a weakly supervised classifier, one tries to optimize the ratio of the data and background densities $p_{\text{data}}(x)/p_{\text{background}}(x)$. These two likelihoods differ only with respect to the small signal contribution and the approximation of their ratio by the classifier is imperfect due to the limited amounts of data and the likely presence of suboptimal features. Therefore, it is not a priori clear whether selecting the model at the minimal loss function value of this data-vs-background task will also yield the optimal classifier model in the signal-vs-background task that is ultimately of interest. An accurate and robust model selection procedure, however, is important not only to select the best performing model for a given architecture, but also for correctly optimizing the hyperparameters in a weakly supervised setting.

To assess the correlation between the weakly supervised loss function with the final signal-to-background performance, the maximum SIC value at each iteration was computed and compared to the loss curves of an HGB classifier model. The loss curve comparison was done for the entire data set, as well as for the signal and background components, in order to assess how they behave in a weakly supervised setting. By comparing the maximum SIC and loss curves, it can be studied whether the current strategy of picking models based on validation loss is a viable strategy to ensure a high performance also on the actual task of interest, namely distinguishing an anomalous signal from background events. For this study, the model was trained using the same parameters as in the first study in subsection 8.1.1 and the baseline feature set. The number of signal events for the training was also set to the same value as in the first study, corresponding to approximately 0.6% signal to background ratio in the signal region. The results can be seen in Figure 8.10.

Considering the upper panel of the figure, it can clearly be seen that the signal loss of both the

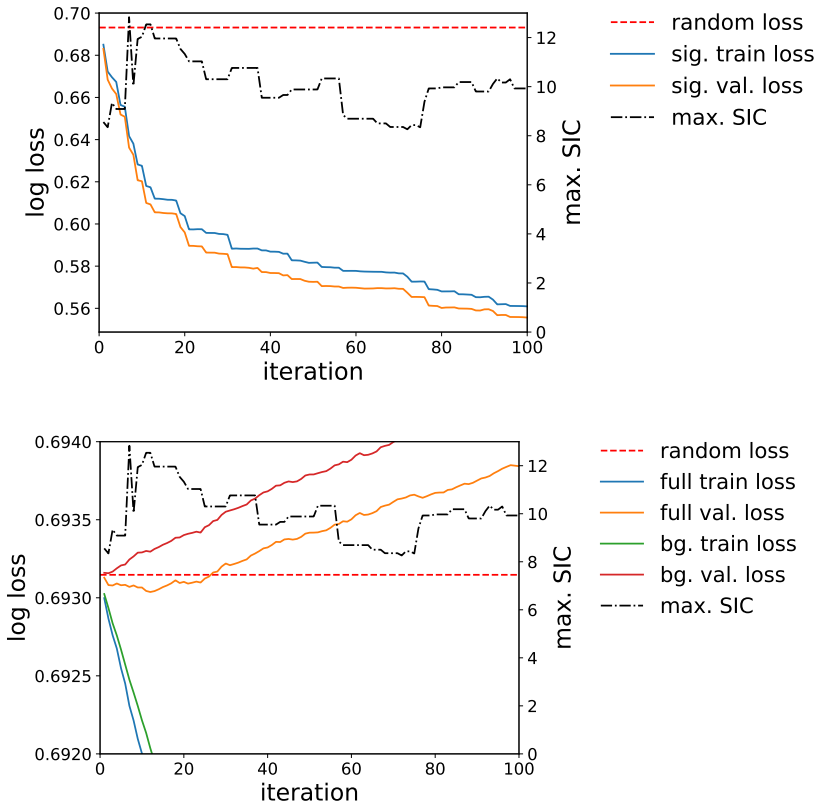


Figure 8.10: Training and Validation loss curves as well as maximum SIC values for each training iteration of an HGB classifier model. Due to the large difference of values, loss curves are compared separately for signal events (upper panel) and both background events and all events (lower panel). The lower panel is zoomed in for better visibility of the validation loss curve of the full data set. Training and validation loss curves were computed using the data and background labels, while the maximum SIC values are based on the truth level information of the independent test set. The random loss value is indicated for reference.

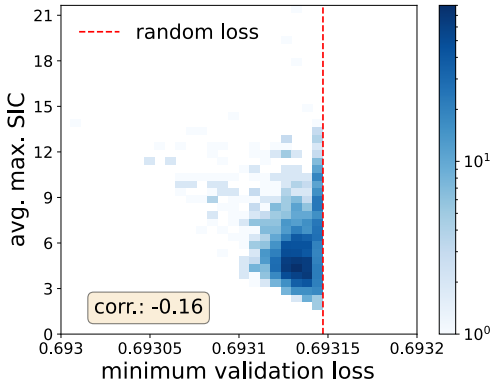
training and validation sets decreases quickly and continues to decrease even at the final iteration. In a fully supervised setting, this behaviour would lead to the consideration of training for more iterations to further improve on the signal events. Looking at the maximum SIC value, however, shows that there is no strong correlation between the losses and achieved performance. The SIC does increase within the first few iterations, but then drops to lower values and continues to stagnate throughout the training while the signal validation loss continues to improve. In an actual search, it would not be possible to select the optimal signal validation loss iteration as the truth-level information is not available. Given these results, however, it is clear that this would also be a suboptimal strategy, which underlines the relatively much higher importance of rejecting the vast amounts of background typically present in particle physics use cases.

The lower panel of Figure 8.10 shows the results for the entire data set as well as the background events only. The range of the loss axis has been reduced for the validation losses to be better visible. The training losses of both the background events and the full data set decrease sharply from the beginning and continue to do so for all iterations. The background validation loss curve shows that the classifier model is overfitting on background events from the start: The curve begins at the random loss value and increases for all subsequent iterations. This behavior is expected, as it shows

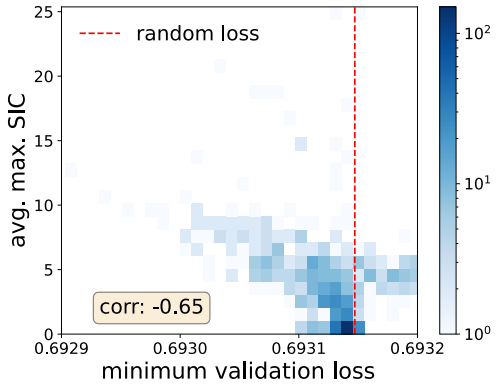
that only noise can be learned from distinguishing the background events in the data sample from the background events in the template which originate from the exact same distributions. The main loss curve of interest for model selection is the validation loss curve computed on the entire data set. This curve shows a composite effect of the signal and background validation losses: Within the first few iterations, where the signal loss decreases sharply while the background loss has yet increased only marginally, the full validation loss decreases to a minimum and then increases until the final iteration. This minimum can be used for model selection and in this training, it also corresponds to one of the highest maximum SIC values achieved. This behaviour has been shown also in several re-runs of the training, which shows that picking the minimum validation loss iteration is a viable strategy to select an iteration amongst the best performing ones. However, it can be seen that the worsening of the full validation loss does not affect the maximum SIC that is achieved, which assumes values around ten for higher iterations. This shows that there is not a clear correlation between validation loss and performance in general. To summarize, the results of this study show that, while the validation loss yields a valid minimum that typically lies in a region of sufficiently high performance, it does not significantly correlate with the maximum SIC and could potentially lead to the selection of suboptimal iterations. Therefore, further study is needed to investigate alternative metrics that can be derived from the weakly supervised labels and shows a better correlation with the signal-vs-background performance.

When selecting the optimal model iteration, the algorithm and its respective hyperparameters are already set. When conducting a model-agnostic search in an entirely new region of phase space, however, these parameters are unknown and need to be found using a dedicated optimization procedure. In an actual search, where no truth level information is available, a model-agnostic metric that has been shown to correlate with the final signal-vs-background performance must be used for this model selection. Similarly to the previous study, it is yet unknown whether the typically used metric, namely the loss on the validation set events, would exhibit such a correlation. To test this, hyperparameter optimizations were performed for three of the previously investigated classification algorithms: the HGB, DNN and AdaBoost models. Again, ensembling – this time based on five independent models – was used to reduce stochastic effects due to model initialization and training/validation sample selection. Furthermore, the baseline feature set was used with the same amount of signal events as in the previous study. The optimization was performed using the `optuna` software package, where the number of trials was set to 10 000. The hyperparameters that were tuned as well as their respective ranges for each of the models can be seen in Table A.3, Table A.4 and Table A.5. If an individual iteration or epoch took longer than ten minutes to train, the trial was pruned and the corresponding SIC value was set to zero.

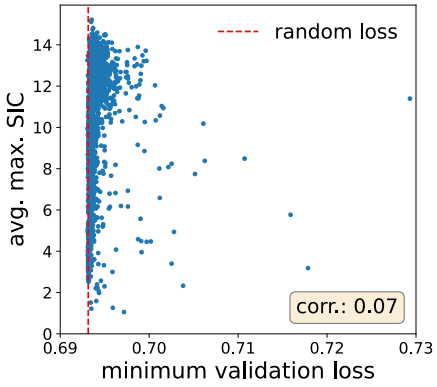
The results of this study can be seen in Figure 8.11. Each subfigure corresponds to one particular classification algorithm. Every point contributing to a bin in the 2D histograms, as well as every point in the scatter plot corresponds to one specific hyperparameter setting. The correlation between the maximum SIC value and the minimum validation loss achieved is also noted. When computing this correlation, the trials that were pruned due to time restrictions were not taken into account. From the plots it can be observed that, apart from the DNN-based classifier, none of the models show a strong correlation between validation loss and maximum SIC value. Interestingly, the HGB classifier, which has been shown to be both robust against noise and to yield good performance, only rarely reaches the



(a) HGB classifier



(b) DNN classifier



(c) AdaBoost classifier

Figure 8.11: 2D histograms and scatter plots for different hyperparameters of the investigated classifier models. Hyperparameters were set by the default optimization algorithm implemented in the `optuna` [220] package. Shown is the average maximum SIC (computed on truth-level information of the test set) as a function of the minimum validation loss. Each point in the scatter plots as well as each entry in a 2D bin corresponds to the result of an ensemble of five independently retrained classifiers using the same hyperparameter settings but different sample realizations for training and validation sets. The loss value of a perfectly random classifier as well as the correlation between the average maximum SIC value and the minimum validation loss are indicated for reference.

previously stated maximum SIC value of around 15, with most parameter settings yielding classifiers with a SIC around a value of four. This shows that thorough hyperparameter optimization is needed for these models to ensure a high performance. While the HGB classifier does show a slight negative correlation between maximum SIC and validation loss, it can be observed that the results have a high variance. The AdaBoost classifier shows an entirely different behaviour: Its maximum SIC values are uncorrelated with the validation loss and also the minimum validation loss that is achieved by these models is just slightly below random at best. In particular, models with worse-than-random validation loss still achieve state-of-the-art performance.

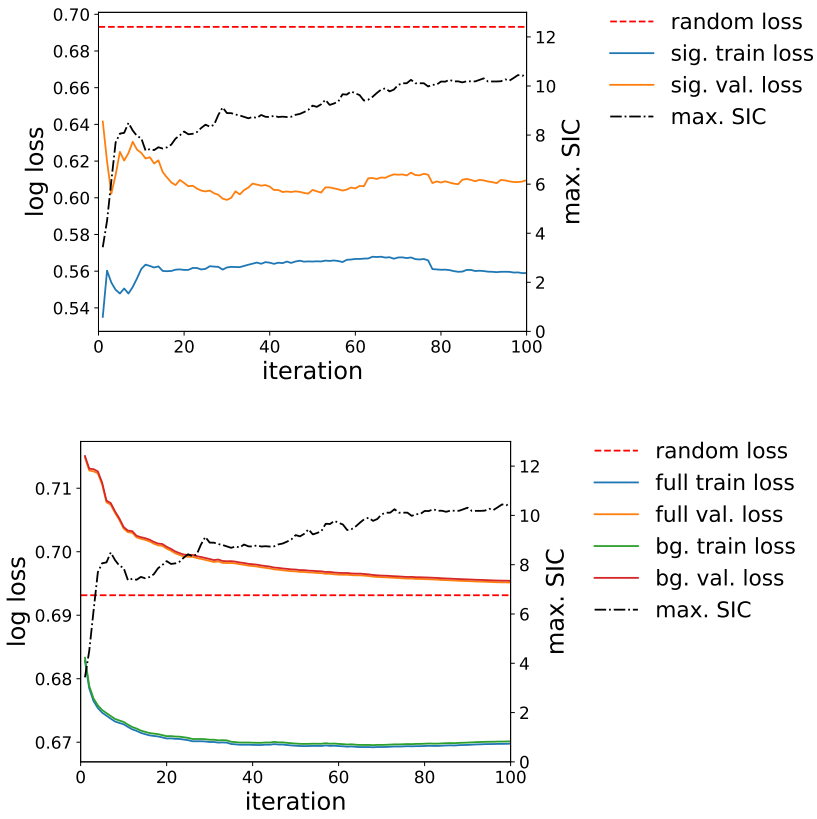


Figure 8.12: The same figure as Figure 8.10, now for the AdaBoost algorighm. Again, results are shown for signal events (upper panel) and background as well as all events (lower panel) separately.

The same plot as Figure 8.10 was repeated for the AdaBoost algorithm. It shows a comparison of the loss curves for the entire data set, the signal events and the background events of both the training and validation sets. For the full data set, the training loss starts out close to the random value and then continuously decreases, while the validation loss starts at a worse-than-random value. However, differently from the behaviour of the HGB classifier (see Figure 8.10) where the loss of the validation set is increasing from the first iteration, this kind of overfitting does not occur for the AdaBoost classifier. Instead, the validation loss continues to converge to the random loss value, which is the optimal value in a weakly supervised setting, at least for the vast majority of events which are background.

Therefore, similar to the loss curves of the HGB classifier, the loss curve of the full data set is dominated by these events. For signal events, the training loss first slightly increases and then stagnates around a similar loss value while the validation loss first decreases for about 20 iterations

and then stagnates. The behaviour of the full validation loss curve leads to the worse-than-random loss results that are shown in the scatterplot in Figure 8.11. However, it can be seen that there is a visible correlation between validation loss and the maximum SIC value achieved, which could not be seen as clearly for the HGB model. This means when using AdaBoost, optimal model states can be selected using the minimum validation loss. For different hyperparameter settings on the other hand, Figure 8.11 suggests that similar loss values can produce vastly different results, which makes the validation loss a suboptimal metric for this task.

To summarize, the correlation between the typically used, model-agnostic metric of the minimum validation loss and the maximum SIC based on truth-level information varies significantly between algorithms and tasks. For complex algorithms that tend to overfit quickly, the minimum validation loss produces correlated results for hyperparameter choices, but does not produce consistent results in optimal iteration selection. For algorithms that are more robust to overfitting, optimal iterations can be selected reliably based on minimum validation loss, but for hyperparameter optimization results vary significantly. Thus, further research is needed for finding a model-agnostic metric that produces consistent results for both iteration and algorithm selection.

8.2 Weak classification in the Latent Space: Latent Cathode

The work presented in this chapter has been previously published in [223], in collaboration with Anna Hallin, Gregor Kasieczka, David Shih and Manuel Sommerhalder. The figures and the written content closely resemble or match the information found in this article. My contribution to the publication consists of the design and description of the diagrams depicting the conceptual approach of the algorithm, general paper writing and proof-reading as well as the assessment and discussion of key research questions and hypotheses during the development of the method with the mentioned collaborators.

When conducting model-agnostic searches for new physics, signal extraction using a bump hunting strategy requires that the employed anomaly detection algorithms do not sculpt the $m_j j$ spectrum significantly. For Cathode, no sculpting is visible for different cuts on the output anomaly score, as was discussed in section 7.1. However, the studies regarding the sculpting were conducted entirely based on the LHCO R&D data set, using the features $\mathbf{x} = (m_{j_1}, \Delta m_j, \tau_{21,j_1}, \tau_{21,j_2})$, none of which is strongly correlated with the respective conditional feature, the dijet invariant mass m_{jj}. In an actual search, strongly correlated features will likely exist. This should also be considered in the training, since it is unknown whether they contain patterns of a new physics signal. Therefore, a study has been conducted that explores the sculpting of Cathode using features that are highly correlated with m_{jj}. As will be discussed below, it was shown that Cathode significantly sculpts the m_{jj} distribution, both within and outside of the SR. Therefore, an improved method was developed to mitigate this problem, which is referred to as "Latent Cathode", or LaCathode [223].

8.2.1 The Algorithm

A key feature of this new method is that it actively decorrelates the used input features from the conditional feature by moving the samples into the latent space of the density estimator and conducts

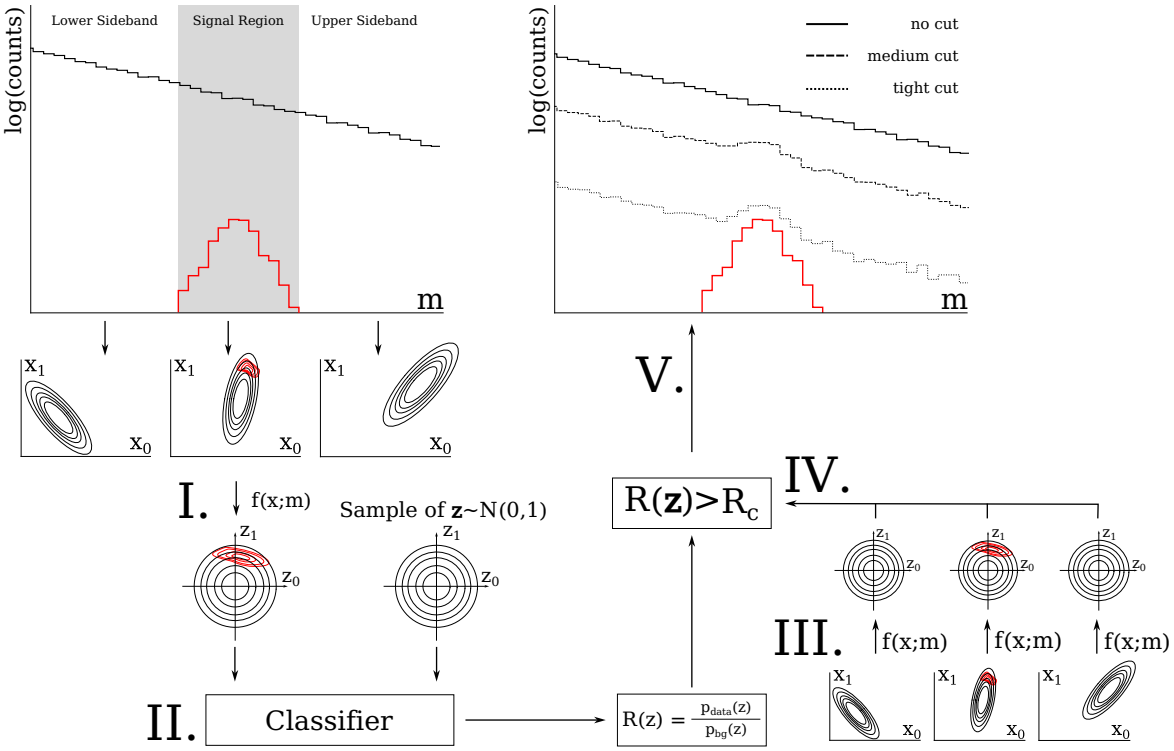


Figure 8.13: A flowchart showing the individual steps of the LaCathode methods, which are described in further detail in the text.

the weak supervision in this space. To understand this principle better, a flowchart of this new method is shown in Figure 8.13. From this diagram, the individual steps of the LaCathode method can be followed. The upper left corner of the flowchart shows the typical situation of a new physics resonance within a signal region window inside a large bulk of background events, that follow an exponential distribution. Below, two feature x_0 and x_1 are plotted against each other in the lower SB, the SR and the upper SB regions, respectively. From the position and orientation in (x_0, x_1) space, it can be seen that these features are highly correlated with m.

Similar to methods like Anode and Cathode, LaCathode uses a density estimator trained on the SB region to model the background distribution. This density estimator is implemented as a MAF, which implements a function $f(\mathbf{x}; m)$ mapping from the input space $\mathbf{x}$ to a latent space $\mathbf{z}$ using m as a conditional variable. In a MAF, each input feature is mapped to exactly one corresponding latent space feature. Therefore, the input $\mathbf{x} = (x_0, x_1)$ gets mapped to $\mathbf{z} = (z_0, z_1)$ in latent space. In step I in the figure, LaCathode first uses the MAF to map the SR data to the latent space and then generates a sample of the learned background distribution, which, in the latent space, approximately corresponds to a standard normal distribution. Next, a classifier is trained to distinguish the SR events that have been mapped to the latent space from the events drawn from a standard normal distribution.

The key feature of LaCathode is that, conducting the weakly supervised classification in latent space instead of the data space, the learned anomaly score $R(\mathbf{z})$ is entirely independent of m. This is because, while the invertible map that the MAF learns depends conditionally on m, the final latent space features are independent of it. Therefore, the learned likelihood ratio becomes:

$$R(\mathbf{z}) = \frac{p_{\text{data}}(\mathbf{z})}{p_{\text{bg.}}(\mathbf{z})} \tag{8.1}$$

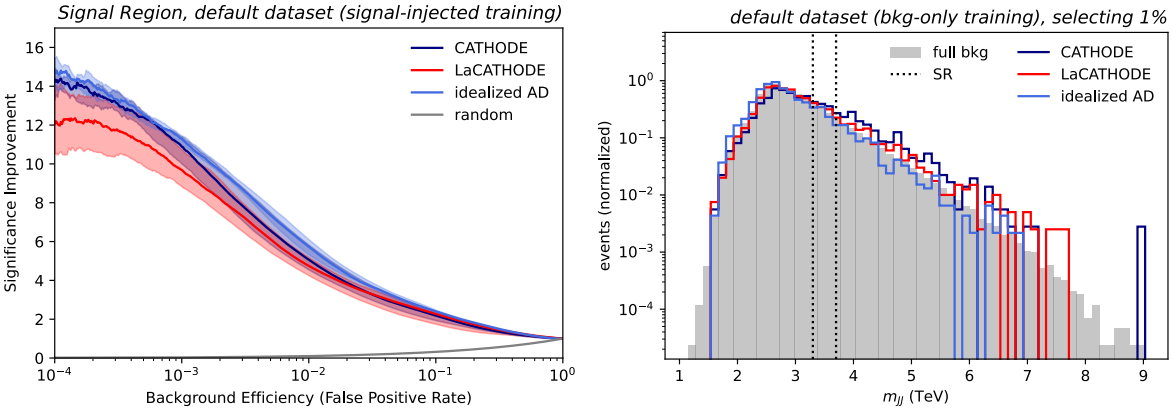


Figure 8.14: Comparison of performance and sculpting of the LaCathode algorithm with Cathode and the IAD, using the default feature set described in the text. The left panel corresponds to the top left panel of Figure 6 of the LaCathode paper [223] and shows the SIC curves of the investigated algorithms, plotted against the background efficiency. The right panel corresponds to the lower panel of Figure 6 in the original LaCathode paper and shows the normalized m_{jj} distributions of the 1 % most anomalous events for each method as well as the uncut background distribution for reference. While signal was injected for the performance comparison, the sculpting comparison was done based on background-only trainings.

After the weak classifier is trained, the entire data of the test set (both in SR and SB regions) is mapped into latent space. Then, it is evaluated on the classifier and events are selected based on a threshold cut on $R(\mathbf{z})$, which is shown in the diagram as step IV. Finally, in step V, using the m distribution of the selected events and a suitable method for signal extraction (e.g. bump hunting), the significance of a possible excess can be quantified.

8.2.2 Benchmark Performance

Several studies were performed to test the performance of this new method in different scenarios, again using the LHCO R&D data set as a benchmark. In these studies, the exact same settings as in the original Cathode paper [150], which are also described in detail in section 7.1, were used. In particular, the hyperparameter settings of the classifier and MAF models were kept the same. First, the training was done on the original feature set, $\mathbf{x} = (m_{j_1}, \Delta m_j, \tau_{21,j_1}, \tau_{21,j_2})$ and the respective comparison can be seen in in Figure 8.14. The left panel of this figure shows a SIC curve comparison between Cathode, LaCathode and an IAD. Again, 10 retrainings were conducted to show the median performance as well as the error band of the inner 68 % of the respective SIC curve distribution. As was seen previously, the Cathode performance saturates the IAD performance on the original feature set. The median performance of LaCathode, however, is below that of Cathode, especially for low background efficiencies. However, the variance of LaCathode is also larger, overlapping between methods. The reason for the performance decrease when doing the weak classification in the latent space has yet to be studied in detail. One explanation of this behaviour is that, while correlations between the input features and the invariant mass are not desired, they do contain significant information about the signal. Considering that the mass distribution of the signal is a narrow resonance shape while the background is described by an exponentially falling distribution, it becomes clear that this feature would be well suited to discriminate between the two. The decorrelation with

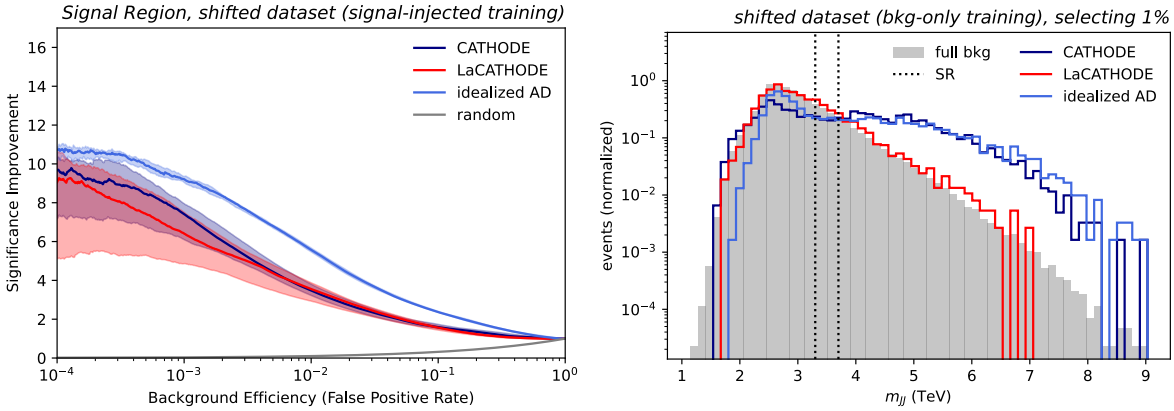


Figure 8.15: Comparison of performance and sculpting of the LaCathode algorithm with Cathode and the IAD using the shifted feature set described in the text. The left panel shows the SIC curves of the investigated algorithms, plotted against the background efficiency. The right panel shows the normalized m_{jj} distributions of the 1 % most anomalous events for each method as well as the uncut background distribution for reference. While signal was injected for the performance comparison, the sculpting comparison was done based on background-only trainings.

the invariant mass by LaCathode leads to the loss of this information, resulting in a reduction of the performance. However, this difference is not significant and only occurs at very low background efficiencies.

The right panel of Figure 8.14 shows the m_{jj} (normalized) distributions of the investigated methods after selecting the most anomalous 1 % of the test data set in comparison with the distribution of the uncut background events. In this case, none of the methods show significant sculpting. However, for Cathode some deviations in the tails can be observed. It should also be noted that an entirely non-sculpted mass spectrum is not necessary to perform signal extraction. The main requirement is that the mass spectrum is *smooth* such that the background parametric function can be fitted. The most sensitive area is the signal region, where sculpting of the background could possibly lead to a non-identification or a misidentification of a signal bump.

8.2.3 Behaviour in the Presence of Correlations

In the first study, none of the used input features is highly correlated with m_{jj}. Therefore, another study was performed using artificially correlated features. In particular, the same correlations as used in the Cathode paper were introduced, as described in Equation 7.1. After correlating the features, the previous comparison study was repeated and the respective results can be seen in Figure 8.15. As discussed previously, the artificial introduction of a correlation between $\mathbf{x}$ and m_{jj} reduces the performance of all algorithms, which is also the case in this study. While LaCathode still shows a significantly higher variance than other methods, its median performance is now similar to that of Cathode, whereas the IAD has the best performance overall. Considering the right panel of Figure 8.15, it can be seen that both the IAD and Cathode sculpt the background spectrum significantly. In fact, the sculpting is to the extent where conducting a smooth fit for signal extraction will be difficult. LaCathode, however, does not show significant sculpting, yielding a background spectrum that is similarly smooth as in the original study. This shows the major advantage of this method: By effectively decorrelating the inputs from the conditional variable, the final anomaly score is uncorrelated

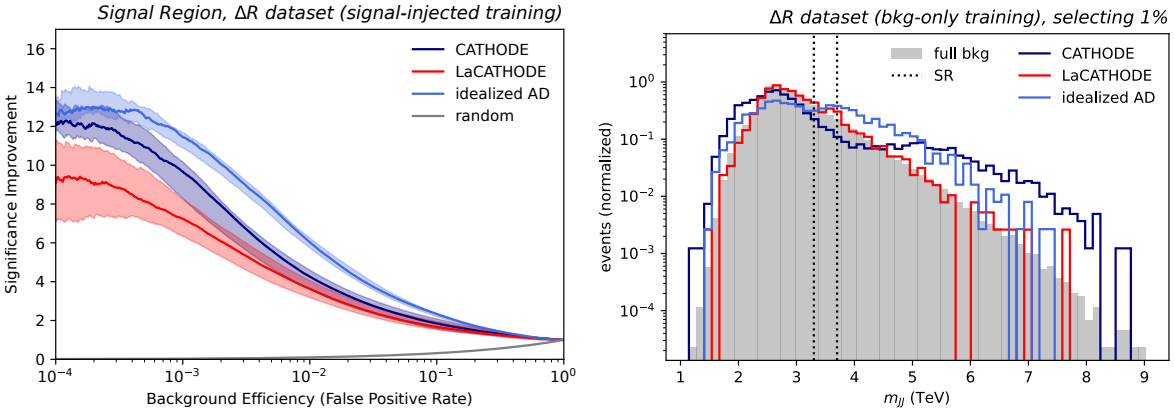


Figure 8.16: Comparison of performance and sculpting of the LaCathode algorithm with Cathode and the IAD using the original feature set with the angular distance, ΔR added. The left panel shows the SIC curves of the investigated algorithms, plotted against the background efficiency. The right panel shows the normalized m_{jj} distributions of the 1 % most anomalous events for each method as well as the uncut background distribution for reference. While signal was injected for the performance comparison, the sculpting comparison was done based on background-only trainings.

with the mass and even when applying a rather tight cut on it, a smooth spectrum is retained.

One shortcoming of the previous study is that the features were artificially forced to be correlated with the dijet invariant mass. Therefore, the sculpting behaviour of LaCathode when including physics features that are actually correlated with m_{jj} still remains to be investigated. This was done in a separate study, where the angular distance, ΔR, which is strongly correlated with the invariant mass, was added to the original feature set. Then the algorithms were re-trained on this new feature set and the previous comparison studies were repeated. The result of this study can be seen in Figure 8.16. The left panel of this figure again shows a decrease in performance when correlated features are present. However, compared to the shifted data set, the performance drop is not as significant, in particular for the IAD and Cathode algorithms. LaCathode, however, sees a similar performance drop, but the variance of results is not as high as before. The achieved significance improvement is below that of an IAD and Cathode throughout a large range of background efficiencies. However, considering the sculpting study in the right panel, it can be seen that both the IAD and Cathode sculpt the background distribution significantly, while LaCathode produces a smooth invariant mass spectrum. This shows that there is a trade-off between performance and the non-sculpting of the background. Conducting the weak classification in latent space leads to an almost entirely smooth spectrum while at the same time retaining a significant fraction of the original performance. Therefore, LaCathode constitutes a considerable improvement of weakly supervised methods for resonant anomaly detection, eliminating the need of selecting non-correlated input features in a model-agnostic search.

Chapter 9

Conclusion

In this work, the development and application of ML-based techniques for data-driven and largely model-agnostic searches have been discussed with respect to the detection of hadronic resonances. The CATHODE method was introduced, which combines density estimation to learn an estimate of the background with weak supervision in a signal region window. On the simulated LHC Olympics R&D data set, it produced state-of-the-art results and saturated the performance of an idealized anomaly detector operating at the limit of perfect density estimation and interpolation. Other than weakly supervised methods such as CWOLA, its performance does not suffer when correlations between its input features $\mathbf{x}$ and the mass variable m exist, which is achieved by interpolating the background estimate from the sidebands into the signal region using conditional density estimation.

CATHODE was also applied in a first-of-its kind analysis to CMS experimental data of the full Run 2 data-taking era (years 2016 to 2018) at a centre-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$. The target of the analysis were generic resonances of the form $A \to BC$, where B and C could be both SM or BSM particles that decay hadronically. The search was conducted using large-radius jets in the final state to be able to find anomalies using the jet substructure. Several signal models with different substructure (2 to 6 prongs) and different mass hypotheses for m_A, m_B and m_C have been tested, most of which had not yet been probed by CMS. Since it is unclear which model-agnostic algorithm is most sensitive for any anomaly, several state-of-the-art methods were employed at once and their performance and achieved limits on the production cross section were compared. The demonstration of the different methods to both set limits on a variety of signal models and scan for generic excesses in large regions of phase space within a single analysis constitutes a key advantage of data-driven approaches over model-specific ones.

In the significance scan for generic anomalies at different mass points, neither CATHODE nor any of the other employed methods found evidence for a significant excess and it must be concluded that no new physics could be found. However, significant improvements with respect to the limits of an inclusive search were observed for most of the investigated signal models. CATHODE in particular yielded the best limits for the $X \to YY' \to 4q$ model at a mass $m_X = 3\,\text{TeV}$ and several of the Y and Y' mass hypotheses, with the highest improvement over an inclusive search of factor 1.9. However, it did not yield the best results for any of the considered models at $m_A = 5\,\text{TeV}$. Nevertheless, this first-of-its-kind analysis applying different data-driven methods to search for a large variety of signal models at once constitutes a strong foundation for future applications at the CMS experiment and beyond.

While CATHODE has been shown to yield state-of-the-art results, several challenges still remain. One key problem is that even though the performance of CATHODE does not break down when correlations between input features and the mass are present, it creates spurious features in the mass spectrum, making standard signal extraction techniques such as the "bump hunt" impossible. A new method was proposed called Latent CATHODE or LaCATHODE, that conducts the weakly supervised classification task in the latent space of the density estimator instead of the data space. This effectively decorrelates the input and mass features and it was shown that LaCATHODE is able to successfully produce smooth background mass spectra even in the presence of strong correlations, with only a minor impact in performance compared to CATHODE.

Finally, a key problem for both CATHODE and other weakly supervised methods is the significant impact that features which do not carry discriminatory power between signal and background events (referred to as uninformative features) have on the classification performance. Adding only a few such features to a weakly supervised task results the performance to drop to almost random levels. Therefore, informative features would have to be selected to retain optimal performance, which is not possible in a model-agnostic scenario, where it is unknown which features are informative and which are not. It was shown that using an ensemble of (histogrammed) gradient boosting classifiers is significantly more robust to uninformative features compared to deep learning-based approaches, retaining the majority of the original performance even when a small fraction of features contains information while the remaining features consist of pure Gaussian noise.

The development of data-driven methods based on density estimation and weak supervision, such as CATHODE and its successful application on CMS experimental data constitute a significant milestone for this approach. Additionally, several remaining problems with weakly supervised have been successfully overcome using different modifications of the original algorithm. This warrants the use of an updated version of CATHODE on the next iteration of data analysis at the CMS experiment and beyond using Run 3 data, as well as the application to other resonant final states. However, several avenues for further research on weakly supervised anomaly detection also exist. Given that CATHODE-based methods saturate the limit of perfect density estimation and interpolation, the overall performance in these methods is classifier-bound. Therefore, future research should target improving the weakly supervised classification task itself, for example by developing model-agnostic methods for optimal feature selection, including physics knowledge in the classifier architecture and/or training procedure or improved techniques for signal refinement in the signal-enriched sample. Another interesting direction for future research lies in models that do not require the signal to be localized and therefore further reduce signal model dependence.

Appendix A

Hyperparameter Settings for Tree-Based Classifiers

Hyperparameter description	Hyperparameter value
Number of iterations	100
Maximum tree depth	23
Minimum number of samples per leaf	20
Maximum number of features to consider per split	`"log2"`
Minimum number of samples required to split a node	14
Maximum fraction of samples to use in bootstrapping	0.543

Table A.1: Optimal hyperparameters for the random forest algorithm, acquired using the `optuna` software package. Hyperparameters not mentioned were left at the default settings of the `RandomForestClassifier` in `scikit-learn`.

Hyperparameter description	Hyperparameter value
Number of iterations	100
Maximum tree depth	unconstrained
Minimum number of samples per leaf	20
Maximum number of leaf nodes	31
Learning rate	0.1

Table A.2: Optimal hyperparameters for the AdaBoost algorithm, acquired using the `optuna` software package. Hyperparameters not mentioned were left at the default settings of the `AdaBoostClassifier` in `scikit-learn` (using `DecisionTreeClassifier` for the base learners).

Hyperparameter description	Scanned values/range
Maximum tree depth	`None` or integer value in $(1, 20)$
Maximum number of leaves	`None` or integer value in $(2, 1000)$
Learning rate	float value in $(0.01, 0.2)$
Maximum number of bins in the histogram	integer value in $(10, 255)$
L2 regularization	float value in $(0.0, 1.0)$
Minimum number of samples that constitute a leaf	integer value in $(1, 50)$

Table A.3: Hyperparameter ranges scanned for the HGB classifiers in the model selection study.

Hyperparameter description	Scanned values/range
Batch size	$2^i, \{i \in \mathbb{N} \mid 5 \leq i \leq 12\}$
Learning rate	float value in $(10^{-5}, 10^{-2})$
Weight decay	float value in $(0.0, 10^{-4})$
Number of hidden layers	integer value in $(1, 10)$
Number of nodes per hidden layer	$2^i, \{i \in \mathbb{N} \mid 3 \leq i \leq 9\}$

Table A.4: Hyperparameter ranges scanned for the DNN classifiers in the model selection study.

Hyperparameter description	Scanned values/range
Maximum tree depth	`None` or integer value in $(1, 20)$
Minimum number of samples required for a split	integer value in $(2, 200)$
Minimum number of samples per leaf	integer value in $(1, 20)$
Maximum of features considered per split	`None` (unconstrained), $\sqrt{N_f}$ or $\log_2(N_f)$, where N_f is total number of features available
Number of trees	integer value in $(10, 1500)$
Learning rate	float value in $(10^{-4}, 10^{-1})$
Maximum number of leaves	integer value in $(2, 1000)$

Table A.5: Hyperparameter ranges scanned for the AdaBoost classifiers in the model selection study.

Appendix B

Data and Simulated Sample Lists for CMS analysis

Sample name	**Integrated luminosity** [fb^{-1}]
/JetHT/Run2016B-21Feb2020_ver1_UL2016_HIPM-v1/MINIAOD	XX
/JetHT/Run2016B-21Feb2020_ver2_UL2016_HIPM-v1/MINIAOD	5.828
/JetHT/Run2016C-21Feb2020_UL2016_HIPM-v1/MINIAOD	2.621
/JetHT/Run2016D-21Feb2020_UL2016_HIPM-v1/MINIAOD	4.286
/JetHT/Run2016E-21Feb2020_UL2016_HIPM-v1/MINIAOD	4.065
/JetHT/Run2016F-21Feb2020_UL2016-v1/MINIAOD	0.418
/JetHT/Run2016F-21Feb2020_UL2016_HIPM-v1/MINIAOD	2.717
/JetHT/Run2016G-21Feb2020_UL2016-v1/MINIAOD	7.653
/JetHT/Run2016H-21Feb2020_UL2016-v1/MINIAOD	8.740
/JetHT/Run2017B-UL2017_MiniAODv2-v1/MINIAOD	4.803
/JetHT/Run2017C-UL2017_MiniAODv2-v1/MINIAOD	9.574
/JetHT/Run2017D-UL2017_MiniAODv2-v1/MINIAOD	4.248
/JetHT/Run2017E-UL2017_MiniAODv2-v1/MINIAOD	9.314
/JetHT/Run2017F-UL2017_MiniAODv2-v1/MINIAOD	13.53
/JetHT/Run2018A-UL2018_MiniAODv2-v1/MINIAOD	14.027
/JetHT/Run2018B-UL2018_MiniAODv2-v1/MINIAOD	7.061
/JetHT/Run2018C-UL2018_MiniAODv2-v1/MINIAOD	6.895
/JetHT/Run2018D-UL2018_MiniAODv2-v1/MINIAOD	31.803
Total integrated luminosity	137.6

Table B.1: Data samples together with the corresponding integrated luminosity.

Table B.2: List of 2016 background samples used in this analysis together with the corresponding cross section.

Sample name	N_*events*	Cross section [pb]
/QCD_Pt_300to470_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	55264000	7823
/QCD_Pt_300to470_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	54096000	
/QCD_Pt_470to600_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	52408000	648.2
/QCD_Pt_470to600_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	50782000	
/QCD_Pt_600to800_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	65088000	186.9
/QCD_Pt_600to800_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	61972000	
/QCD_Pt_800to1000_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	37782000	32.293
/QCD_Pt_800to1000_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	35527000	
/QCD_Pt_1000To1400_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	19892000	9.4183
/QCD_Pt_1000to1400_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	19077000	
/QCD_Pt_1400to1800_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	10722000	0.84265
/QCD_Pt_1400to1800_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	11000000	
/QCD_Pt_1800to2400_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	5236000	0.114943
/QCD_Pt_1800to2400_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	5262000	
/QCD_Pt_2400to3200_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	2848000	0.00682981
/QCD_Pt_2400to3200_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	2999000	
/QCD_Pt_3200toInf_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	996000	0.000165445
/QCD_Pt_3200toInf_TuneCP5_13TeV_pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	1000000	
/WJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	4455853	315.6
/WJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	5144427	
/WJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	6793578	68.57
/WJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	7668058	
/WJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	6769101	34.9
/WJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	7740501	
/ZJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	6942718	145.4
/ZJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	3454056	
/ZJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	5500386	34.0
/ZJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	1623377	
/ZJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	3740901	18.67
/ZJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v2/MINIAODSIM	3726992	
/TT_Mtt-700to1000_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	33586554	76.605
/TT_Mtt-700to1000_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	18020206	
/TT_Mtt-1000toInf_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v1/MINIAODSIM	24177380	20.578
/TT_Mtt-1000toInf_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	23219884	
/ST_t-channel_top_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v3/MINIAODSIM	63073000	115.3
/ST_t-channel_top_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v3/MINIAODSIM	55961000	
/ST_t-channel_antitop_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v3/MINIAODSIM	30609000	69.09
/ST_t-channel_antitop_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v3/MINIAODSIM	31024000	
/ST_tW_top_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	2491000	35.85
/ST_tW_top_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	2300000	
/ST_tW_antitop_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODv2-106X_mcRun2_asymptotic_v17-v2/MINIAODSIM	2554000	35.85
/ST_tW_antitop_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_preVFP_v11-v1/MINIAODSIM	2300000	

Table B.3: List of 2017 background samples used in this analysis together with the corresponding cross section.

Sample name	N_*events*	Cross section [pb]
/QCD_Pt_300to470_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	55690000	7823
/QCD_Pt_470to600_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	50885000	648.2
/QCD_Pt_600to800_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	67379000	186.9
/QCD_Pt_800to1000_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	36890000	32.293
/QCD_Pt_1000to1400_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	19781000	9.4183
/QCD_Pt_1400to1800_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	10994000	0.84265
/QCD_Pt_1800to2400_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	5488000	0.114943
/QCD_Pt_2400to3200_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	2997000	0.00682981
/QCD_Pt_3200toInf_TuneCP5_13TeV_pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	1000000	0.000165445
/WJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	9927793	315.6
/WJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	14667933	68.57
/WJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	14722417	34.9
/ZJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	14884962	145.4
/ZJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	11702567	34.0
/ZJetsToQQ_HT-800toInf_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	9384525	18.67
/TT_Mtt-700to1000_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	35862238	76.605
/TT_Mtt-1000toInf_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	22724532	20.578
/ST_t-channel_top_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	127790000	115.3
/ST_t-channel_antitop_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v1/MINIAODSIM	69509000	69.09
/ST_tW_top_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	5649000	35.85
/ST_tW_antitop_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/MINIAODSIM	5674000	35.85

Table B.4: List of 2018 background samples used in this analysis together with the corresponding cross section.

Sample name	N_*events*	Cross section [pb]
/QCD_Pt_300to470_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	57910000	7823
/QCD_Pt_470to600_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	52448000	648.2
/QCD_Pt_600to800_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	67508000	186.9
/QCD_Pt_800to1000_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	37160000	32.293
/QCD_Pt_1000to1400_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	19730000	9.4183
/QCD_Pt_1400to1800_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	10982000	0.84265
/QCD_Pt_1800to2400_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	5491000	0.114943
/QCD_Pt_2400to3200_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	2997000	0.00682981
/QCD_Pt_3200toInf_TuneCP5_13TeV_pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	1000000	0.000165445
/WJetsToQQ_HT-400to600_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	9335298	315.6
/WJetsToQQ_HT-600to800_TuneCP5_13TeV-madgraphMLM-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	13633226	68.57
/WJetsToQQ_HT-800toInf/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	13581343	34.9
/ZJetsToQQ_HT-400to600/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	13930474	145.4
/ZJetsToQQ_HT-600to800/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	12029507	34.0
/ZJetsToQQ_HT-800toInf/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	9681521	18.67
/TT_Mtt-700to1000_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	30720345	76.605
/TT_Mtt-1000toInf_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	23758200	20.578
/ST_t-channel_top_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	178756000	115.3
/ST_t-channel_antitop_4f_InclusiveDecays_TuneCP5_13TeV-powheg-madspin-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v1/MINIAODSIM	95833000	69.09
/ST_tW_top_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	5649000	35.85
/ST_tW_antitop_5f_inclusiveDecays_TuneCP5_13TeV-powheg-pythia8/RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_v16_L1v1-v2/MINIAODSIM	5674000	35.85

Appendix C

Signal Shapes used in CMS analysis

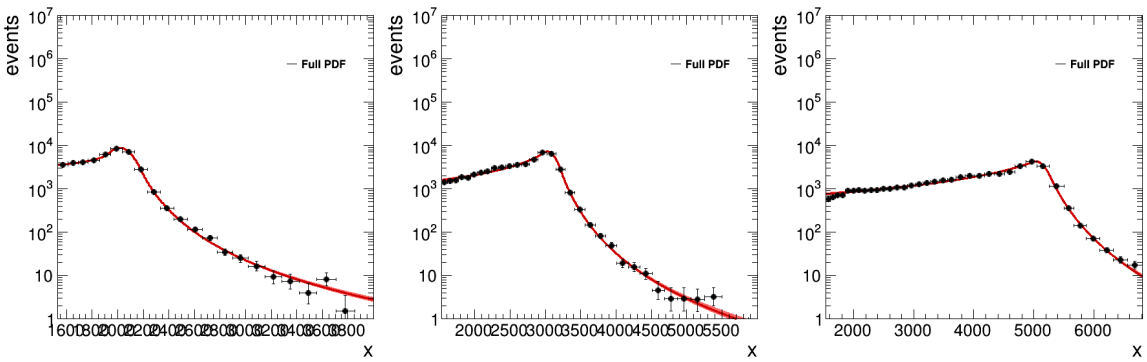


Figure C.1: Signal shapes for $W_{KK} \rightarrow WR \rightarrow WWW$, $m_R = 170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

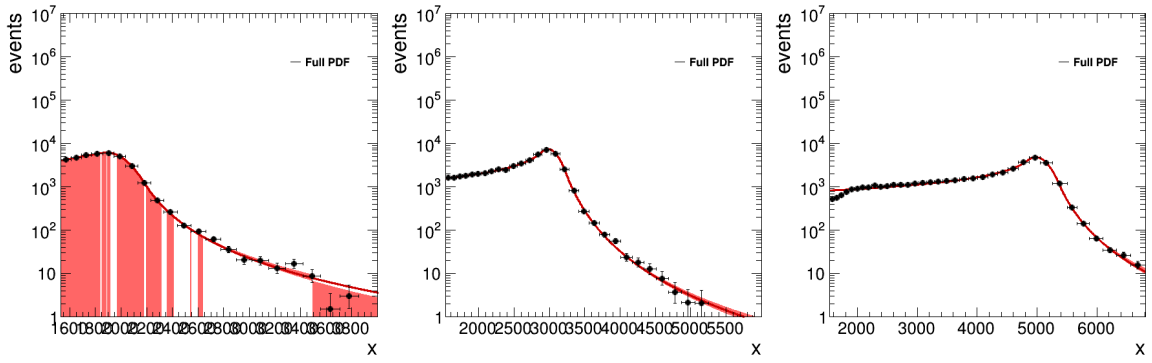


Figure C.2: Signal shapes for $W_{KK} \to WR \to WWW$, $m_R = 400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

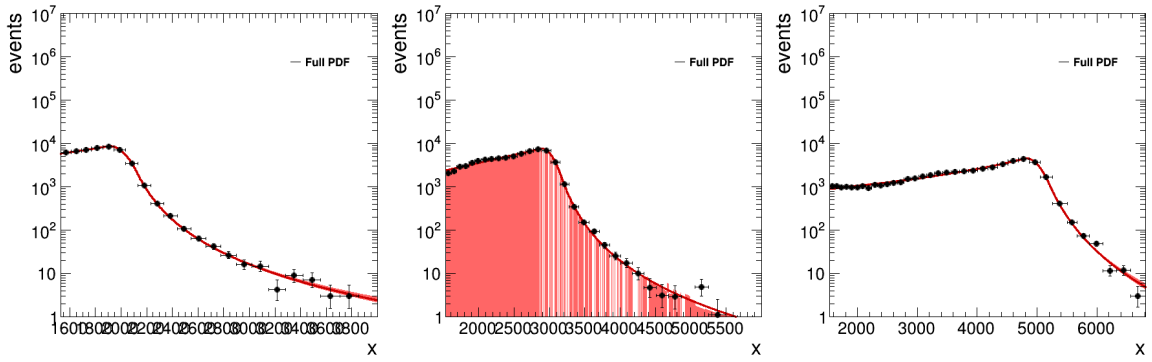


Figure C.3: Signal shapes for $W' \to B't$, $m_{B'} = 25$ GeV at 2 (left), 3 (center), and 5 TeV (right).

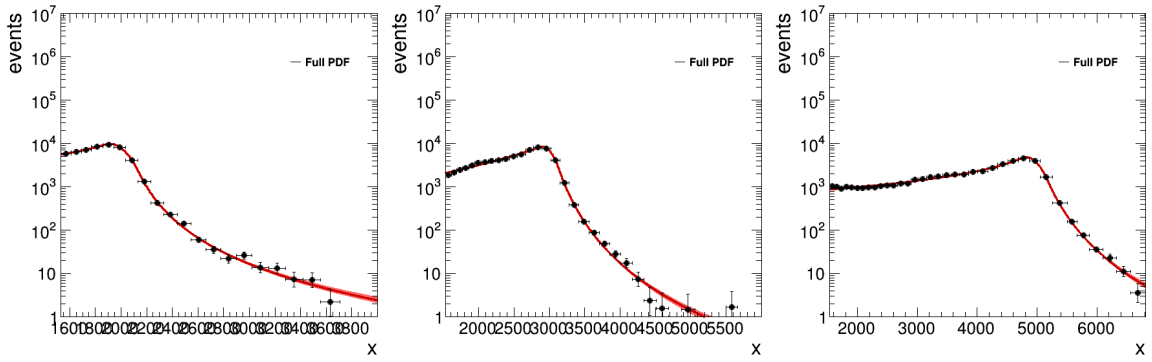


Figure C.4: Signal shapes for $W' \to B't$, $m_{B'} = 80$ GeV at 2 (left), 3 (center), and 5 TeV (right).

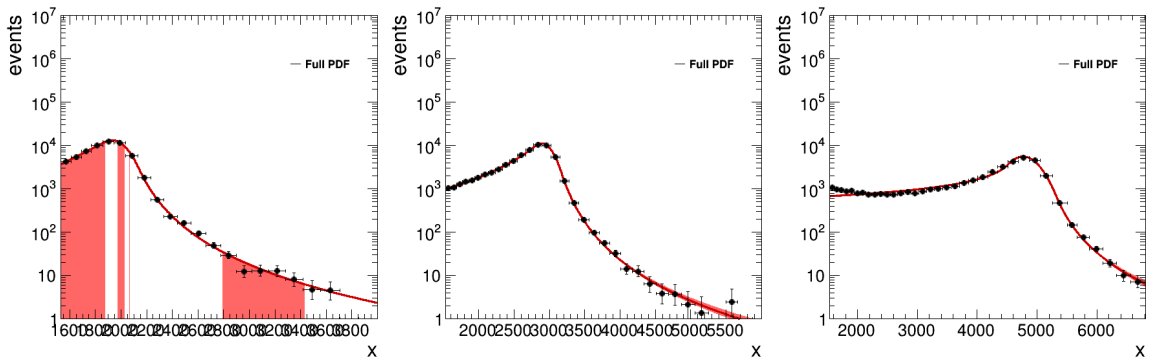


Figure C.5: Signal shapes for $W' \to B't$, $m_{B'} = 170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

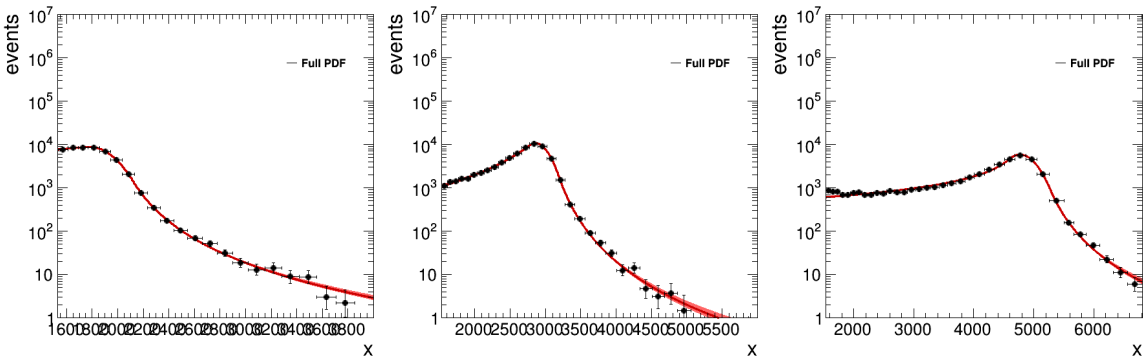


Figure C.6: Signal shapes for $W' \to B't$, $m_{B'} = 400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

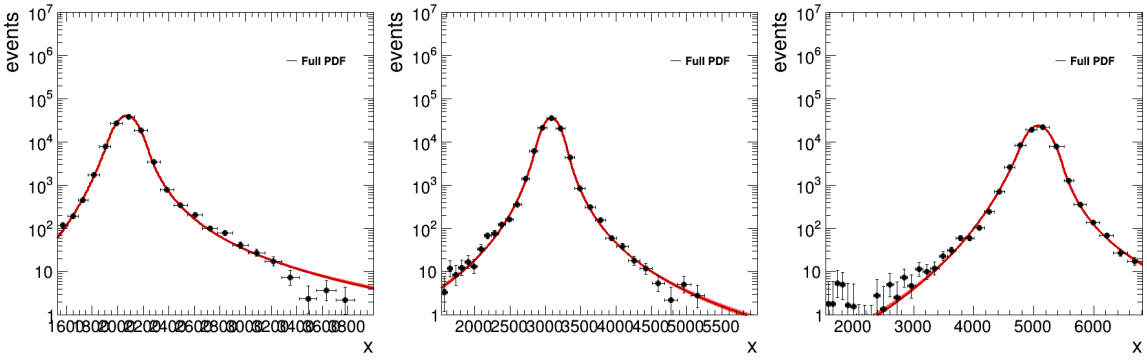


Figure C.7: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 25/25$ GeV at 2 (left), 3 (center), and 5 TeV (right).

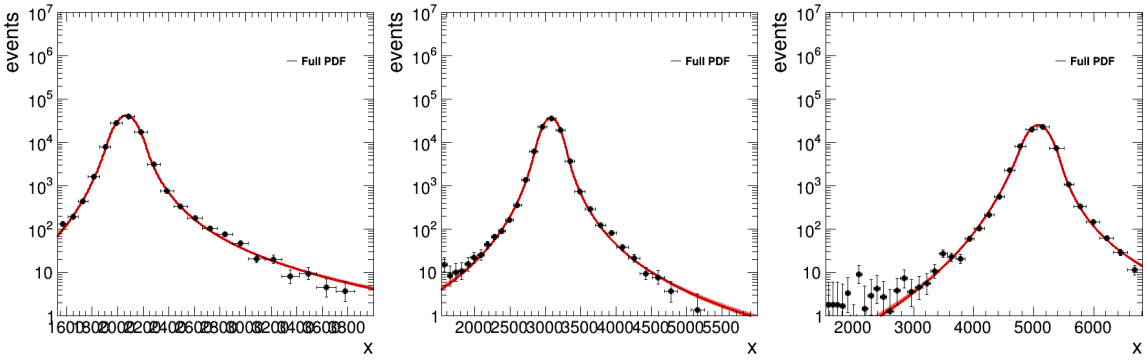


Figure C.8: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 25/80$ GeV at 2 (left), 3 (center), and 5 TeV (right).

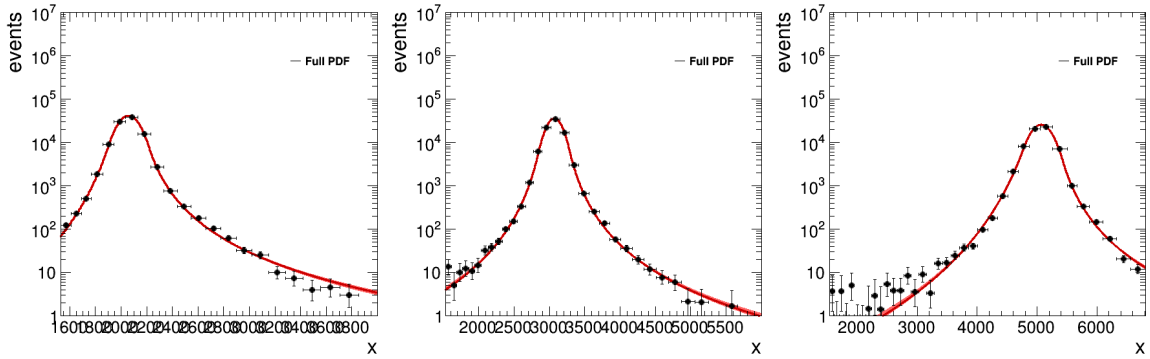


Figure C.9: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 25/170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

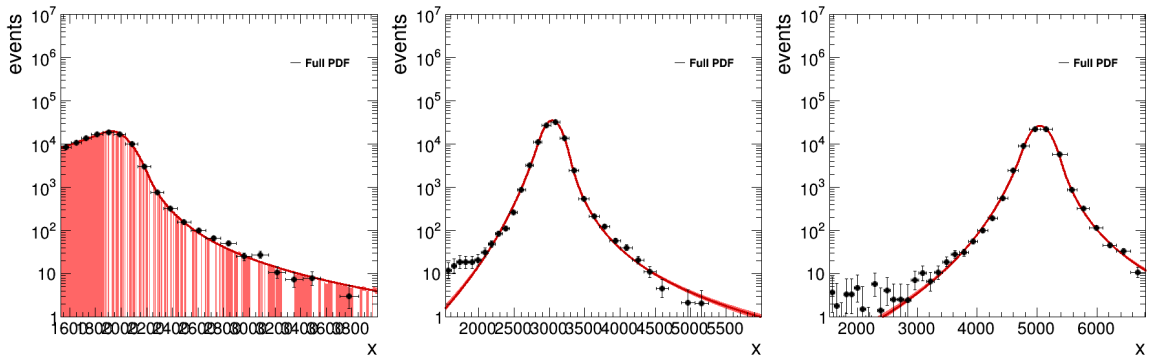


Figure C.10: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 25/400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

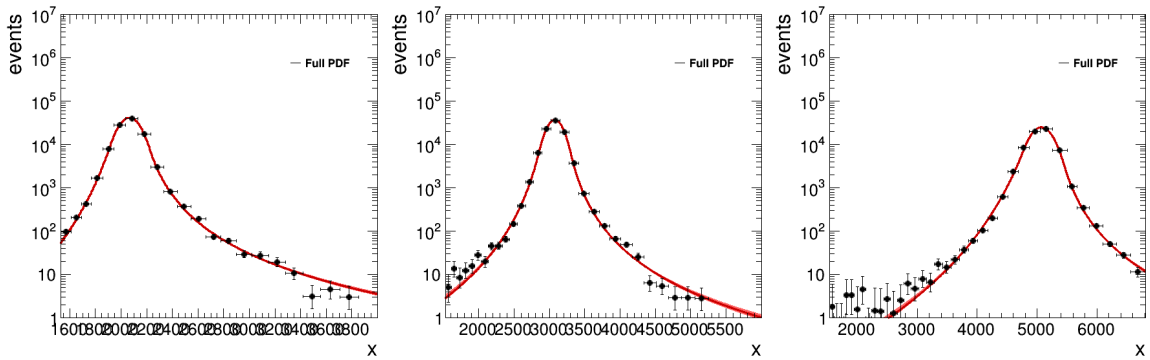


Figure C.11: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 80/25$ GeV at 2 (left), 3 (center), and 5 TeV (right).

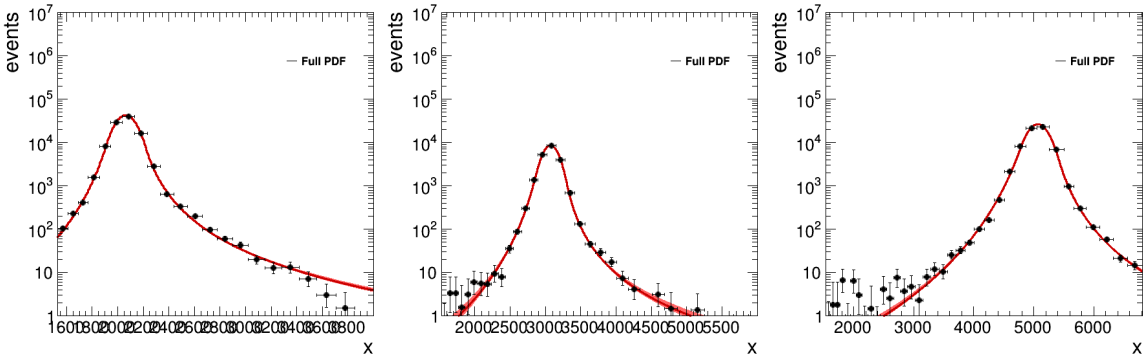


Figure C.12: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 80/80$ GeV at 2 (left), 3 (center), and 5 TeV (right).

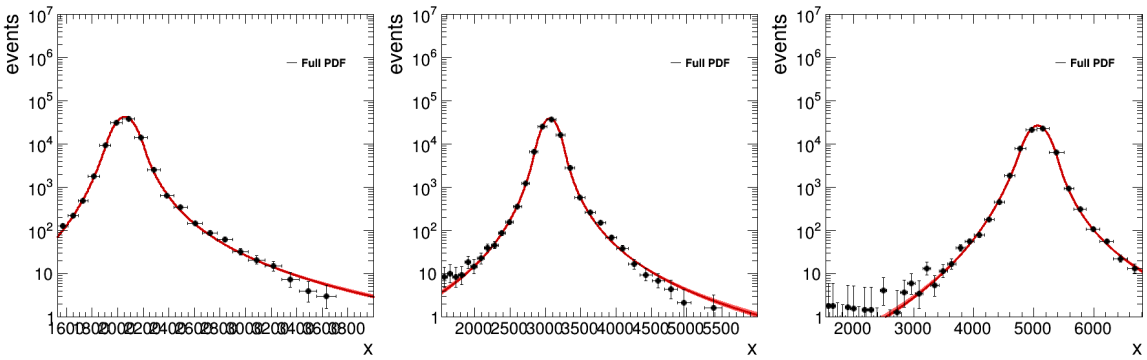


Figure C.13: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 80/170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

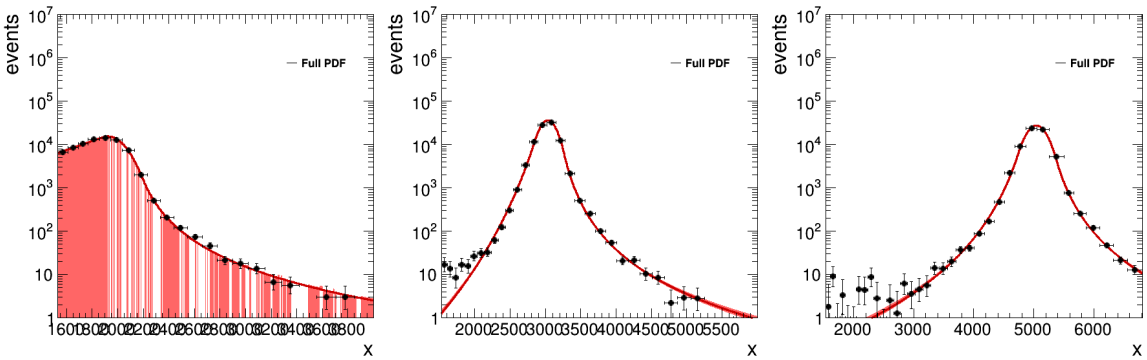


Figure C.14: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 80/400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

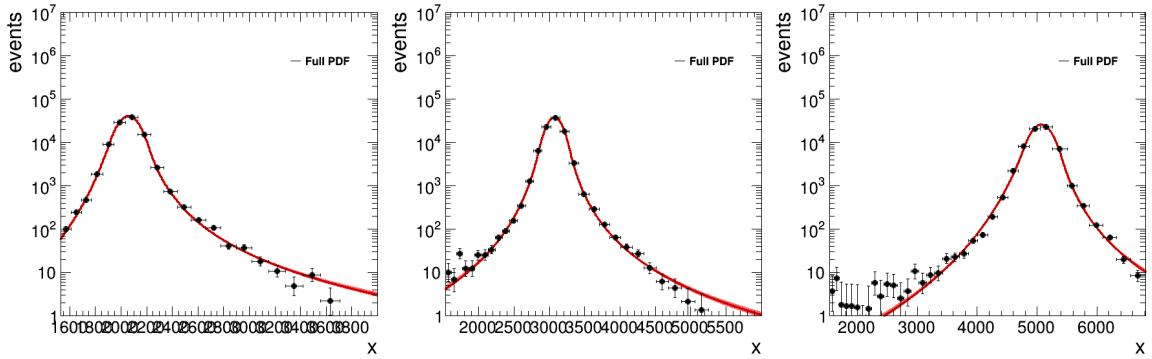


Figure C.15: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 170/25$ GeV at 2 (left), 3 (center), and 5 TeV (right).

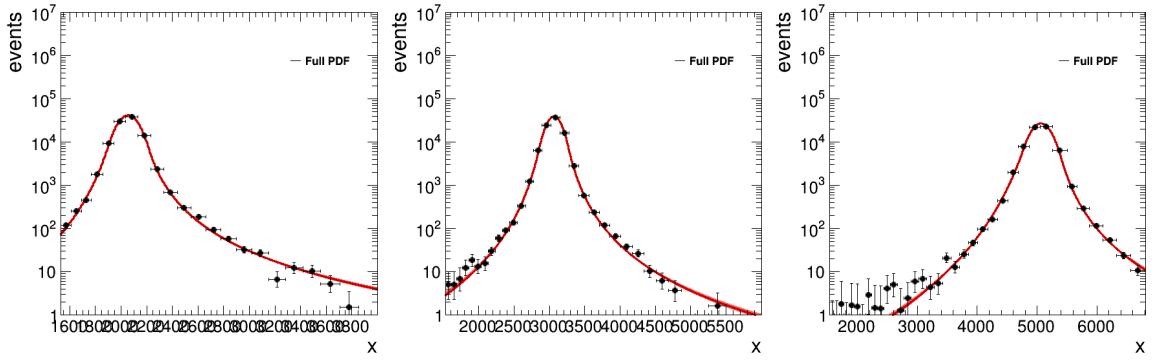


Figure C.16: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 170/80$ GeV at 2 (left), 3 (center), and 5 TeV (right).

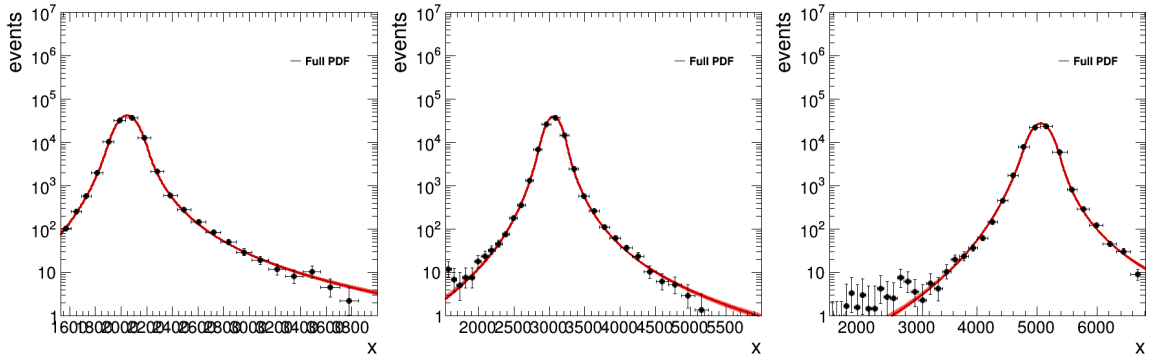


Figure C.17: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 170/170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

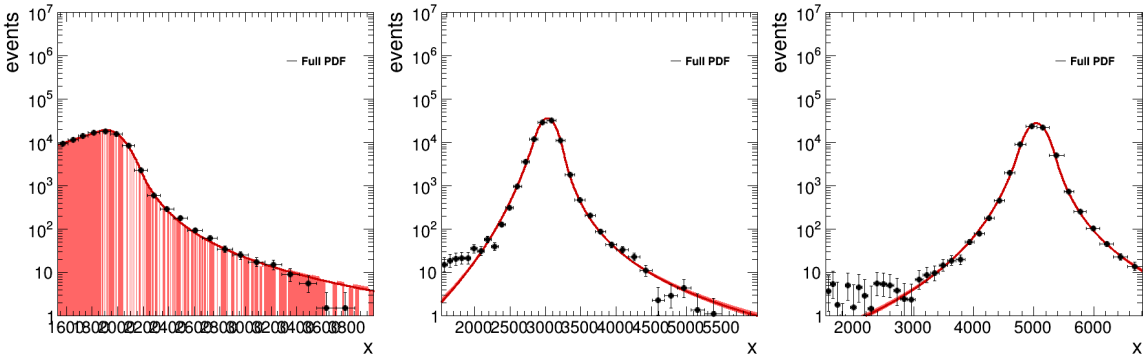


Figure C.18: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 170/400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

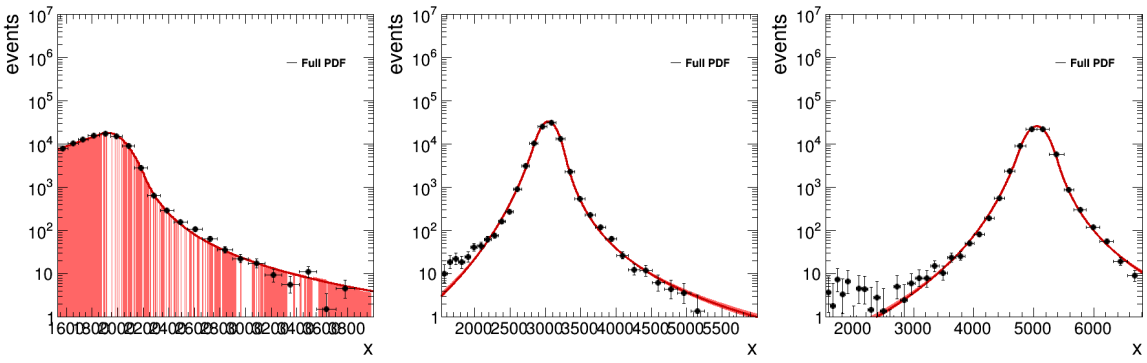


Figure C.19: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 400/25$ GeV at 2 (left), 3 (center), and 5 TeV (right).

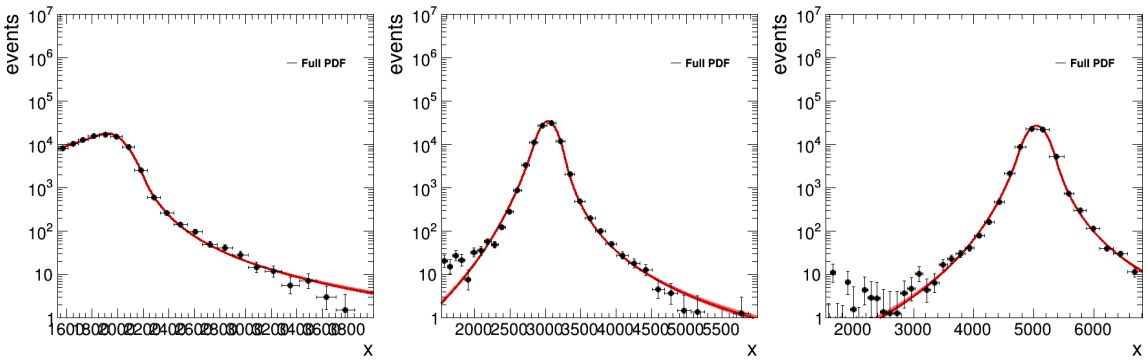


Figure C.20: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 400/80$ GeV at 2 (left), 3 (center), and 5 TeV (right).

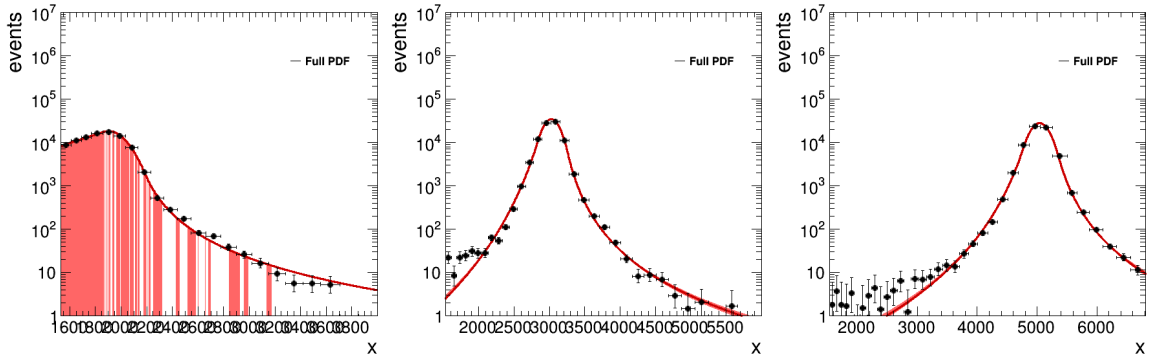


Figure C.21: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 400/170$ GeV at 2 (left), 3 (center), and 5 TeV (right).

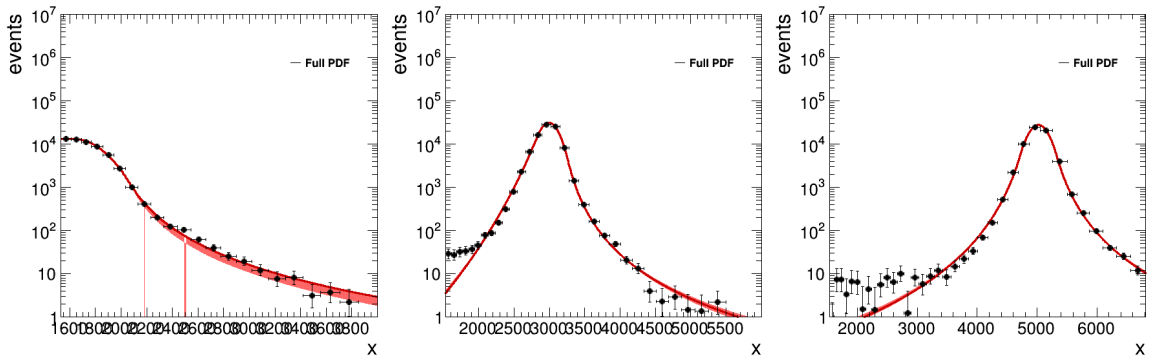


Figure C.22: Signal shapes for $X \to YY' \to 4q$, $m_X/m_{X'} = 400/400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

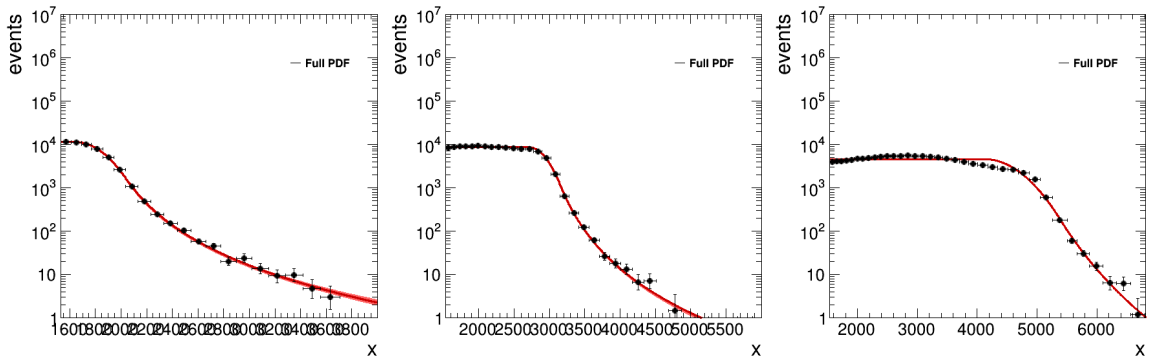


Figure C.23: Signal shapes for $Z \to T'T'$, $T' \to tZ$, $m_{T'} = 400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

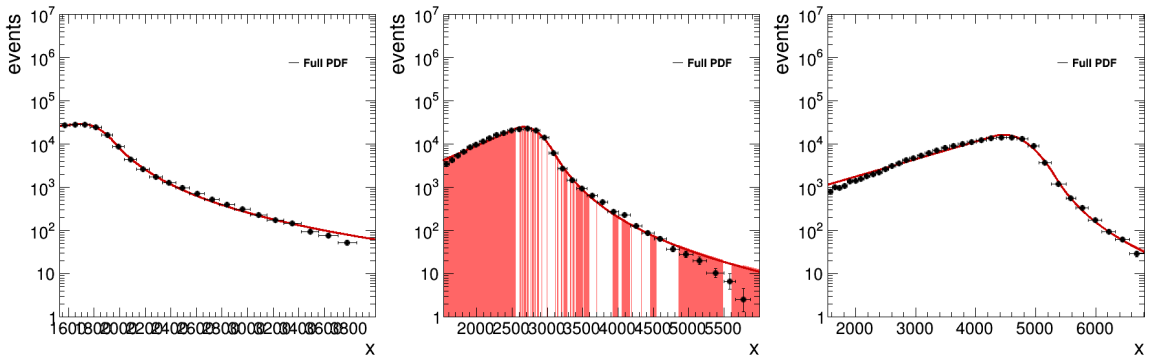


Figure C.24: Signal shapes for $Y \rightarrow HH \rightarrow 4t$, $m_H = 400$ GeV at 2 (left), 3 (center), and 5 TeV (right).

Appendix D

Additional Figures and Tables for Validation of CATHODE(-b)

Process	$\mathbf{m}_\mathrm{A}$ [TeV]	$\mathbf{m}_\mathrm{B}$ [GeV]	n subjets B	$\mathbf{m}_\mathrm{C}$ [GeV]	n subjets C
$G \to WW$	[2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6]	m_W	2	m_W	2
$G \to ZZ$	[2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6]	m_Z	2	m_Z	2
$W' \to WZ$	[2, 2.5, 3, 3.5, 4, 4.5]	m_W	2	m_Z	2
$Z' \to WW$	[2, 2.5, 3, 3.5, 4, 4.5]	m_W	2	m_W	2
$b^* \to tW$	[2, 2.5, 3, 3.5, 4, 4.5]	m_t	3	m_W	2
$W_{kk} \to RW \to WWW$	[2, 2.5, 3, 3.5, 4, 4.5]	$R_0 = \frac{m_R}{m_{W_{kk}}} \in [0.06, 0.08, 0.1, 0.2, 0.3]$	4	m_W	2

Table D.1: Signal models considered for the control region definition, including the masses of the resonance and the daughter particles as well as the number of subjets expected in the two large-radius jets in the final state.

Sample name
/BulkGravToWW_narrow_M-2000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-2500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-3000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-3500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-4000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-4500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-5000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-5500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToWW_narrow_M-6000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-2000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-2500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-3000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-3500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-4000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-4500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-5000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-5500_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BulkGravToZZToZhadZhad_narrow_M-6000_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_new_pmx_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-2000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-2500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-3000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-3500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-4000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WprimeToWZToWhadZhad_narrow_M-4500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-2000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-2500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-3000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-3500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-4000_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/ZprimeToWW_narrow_M-4500_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BstarToTW_M-2000_LH_TuneCP5_13TeV-madgraph-pythia8/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BstarToTW_M-2600_LH_TuneCP5_13TeV-madgraph-pythia8/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BstarToTW_M-3000_LH_TuneCP5_13TeV-madgraph-pythia8/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BstarToTW_M-3600_LH_TuneCP5_13TeV-madgraph-pythia8/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/BstarToTW_M-4000_LH_TuneCP5_13TeV-madgraph-pythia8/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v1/MINIAODSIM
/WkkToWRadionToWWW_M2000-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2000-R0-2_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2000-R0-3_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2500-R0-08_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2500-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2500-R0-2_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M2500-R0-3_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3000-R0-06_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3000-R0-08_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3000-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3500-R0-06_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3500-R0-08_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3500-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M3500-R0-2_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M4000-R0-08_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M4000-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M4500-R0-06_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M4500-R0-08_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM
/WkkToWRadionToWWW_M4500-R0-1_TuneCP5_13TeV-madgraph/RunIIFall17MiniAODv2-PU2017_12Apr2018_94X_mc2017_realistic_v14-v2/MINIAODSIM

Table D.2: Signal MC used for testing different validation CR definitions, grouped into the same types of signals with varying masses.

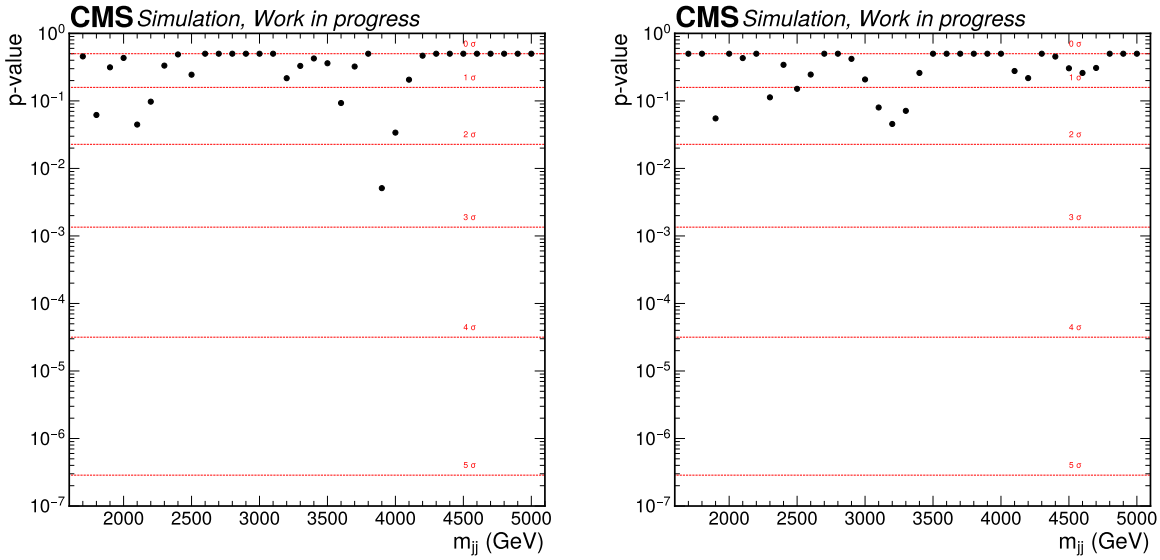


Figure D.1: CATHODE-b p-value scan for the mock MC data set. Shown is the p-value obtained from the combined signal template and background fit of the CATHODE-b-selected events against the dijet invariant mass, m_{jj}. The results are shown for the two selection efficiencies of 10 % (left panel) and 1 % (right panel).

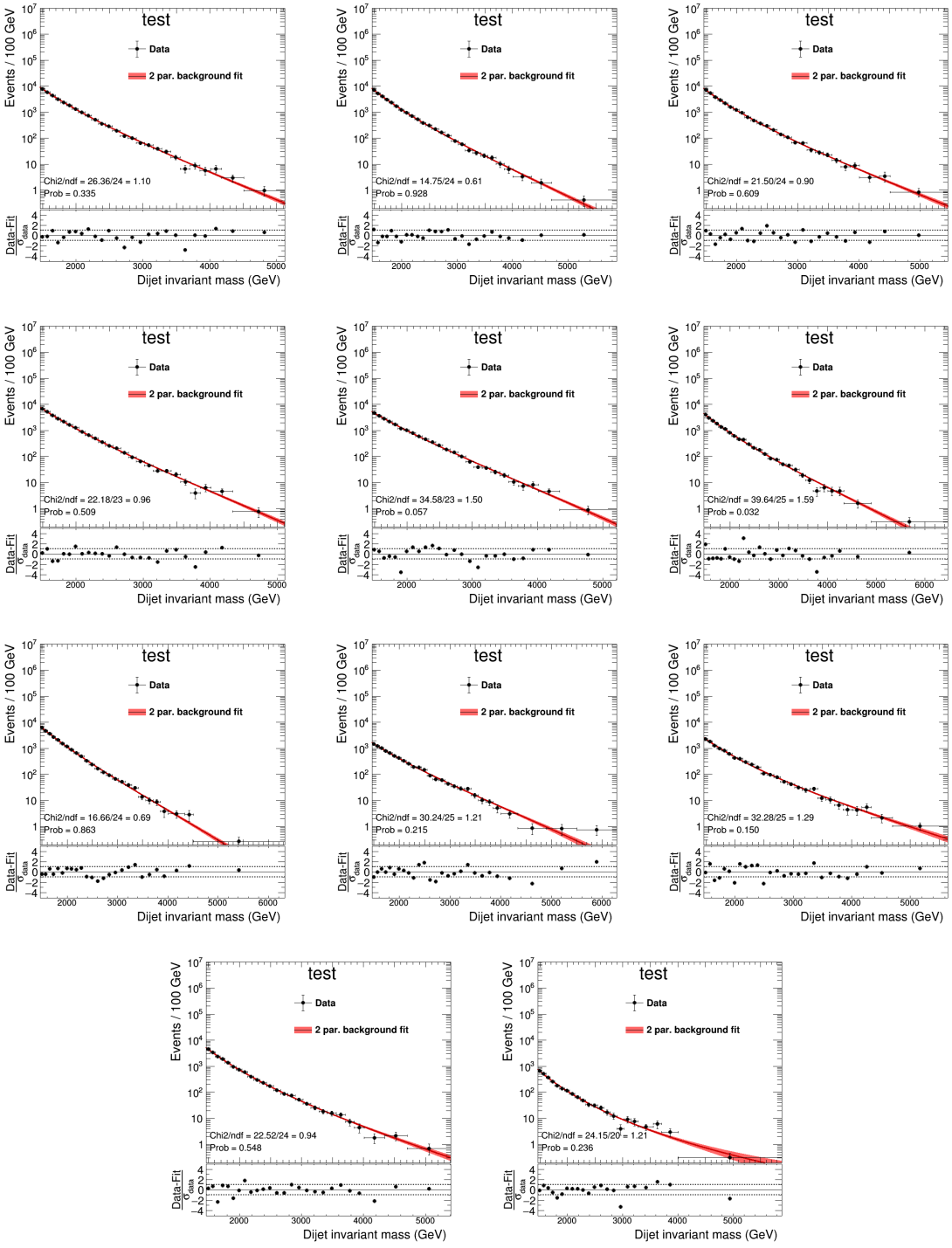


Figure D.2: Background-only fits of the CATHODE-b-selected events from the mock MC data set for each considered SR window. Events were selected based on a selection efficiency of 1 %. The plots are sorted by the center SR window mass from lowest to highest in a top left to bottom right order. The lower panel of each plot shows the pull distributions of the fit.

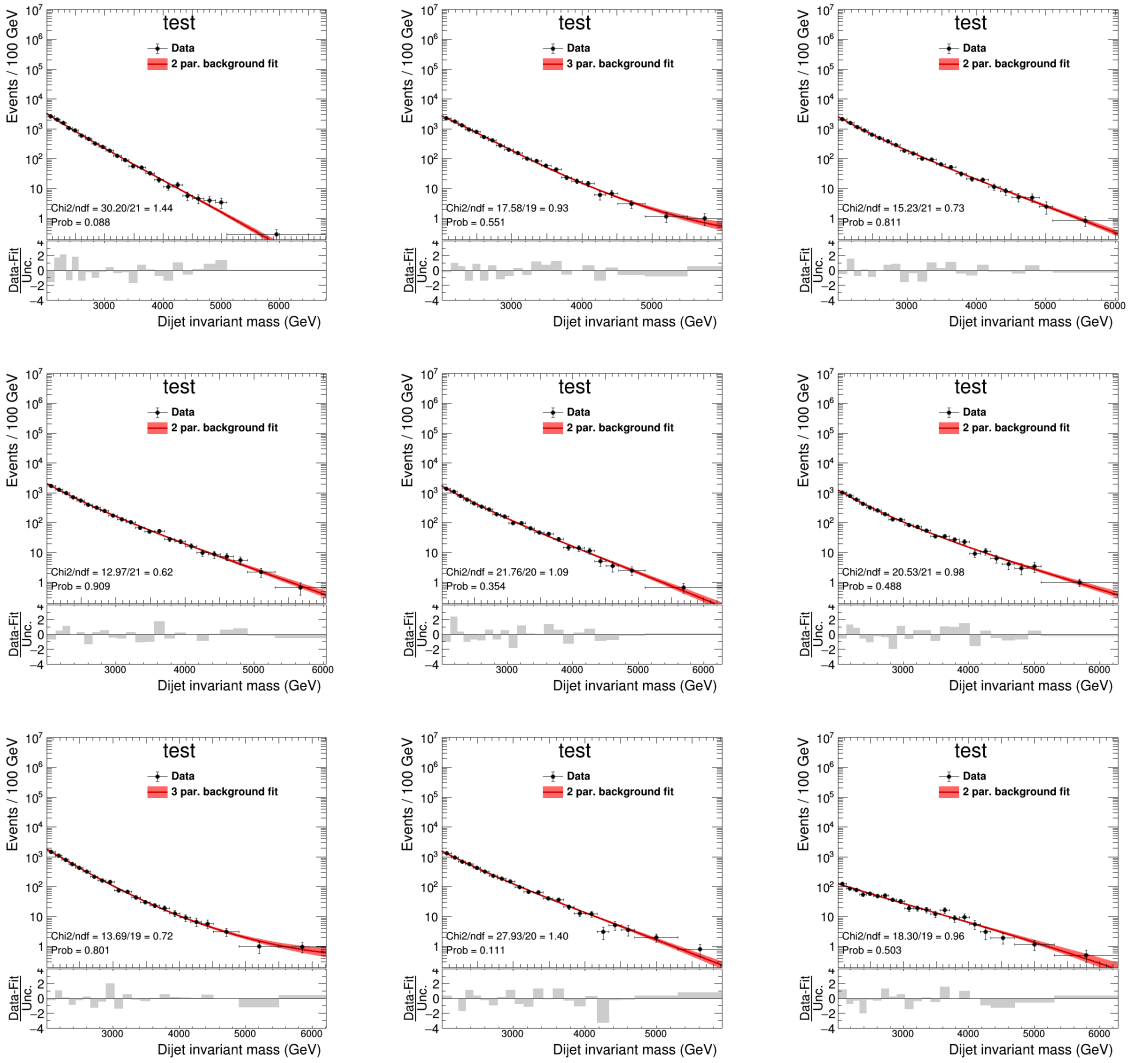


Figure D.3: Background-only fits of the CATHODE-b-selected events from the control region data set in each considered SR window. Events were selected based on a selection efficiency of 1 %. The plots are sorted by the centre SR window mass from lowest to highest in a top left to bottom right order. The lower panel of each plot shows the pull distributions of the fit.

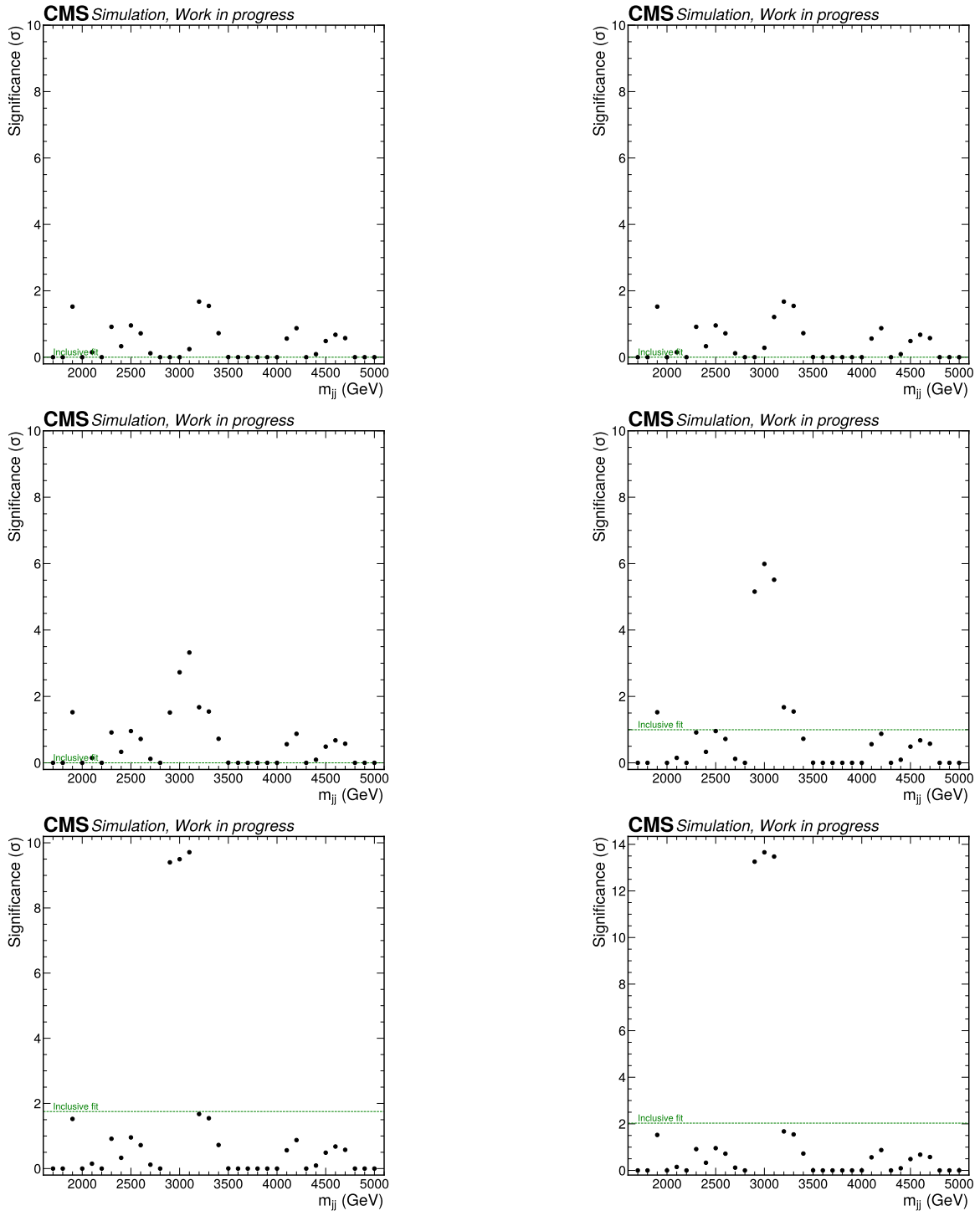


Figure D.4: Significances obtained from CATHODE-b when injecting an increasing amount of $X \to YY'$ signal in the MC mock data set, with $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$, and $m_{Y'} = 80\,\text{GeV}$. The injected cross sections are starting at 5 fb in steps of 5 fb and the plots are ordered by increasing injections from top left to bottom right. Black dots represent the significance obtained from the combined signal shape and parametric background fits on CATHODE-selected events, using a selection efficiency in the signal region of 1 %. The dashed green lines represent the significance achieved using an "inclusive" fit, where the combined fit is applied to the resonance mass of 3 TeV without any selection.

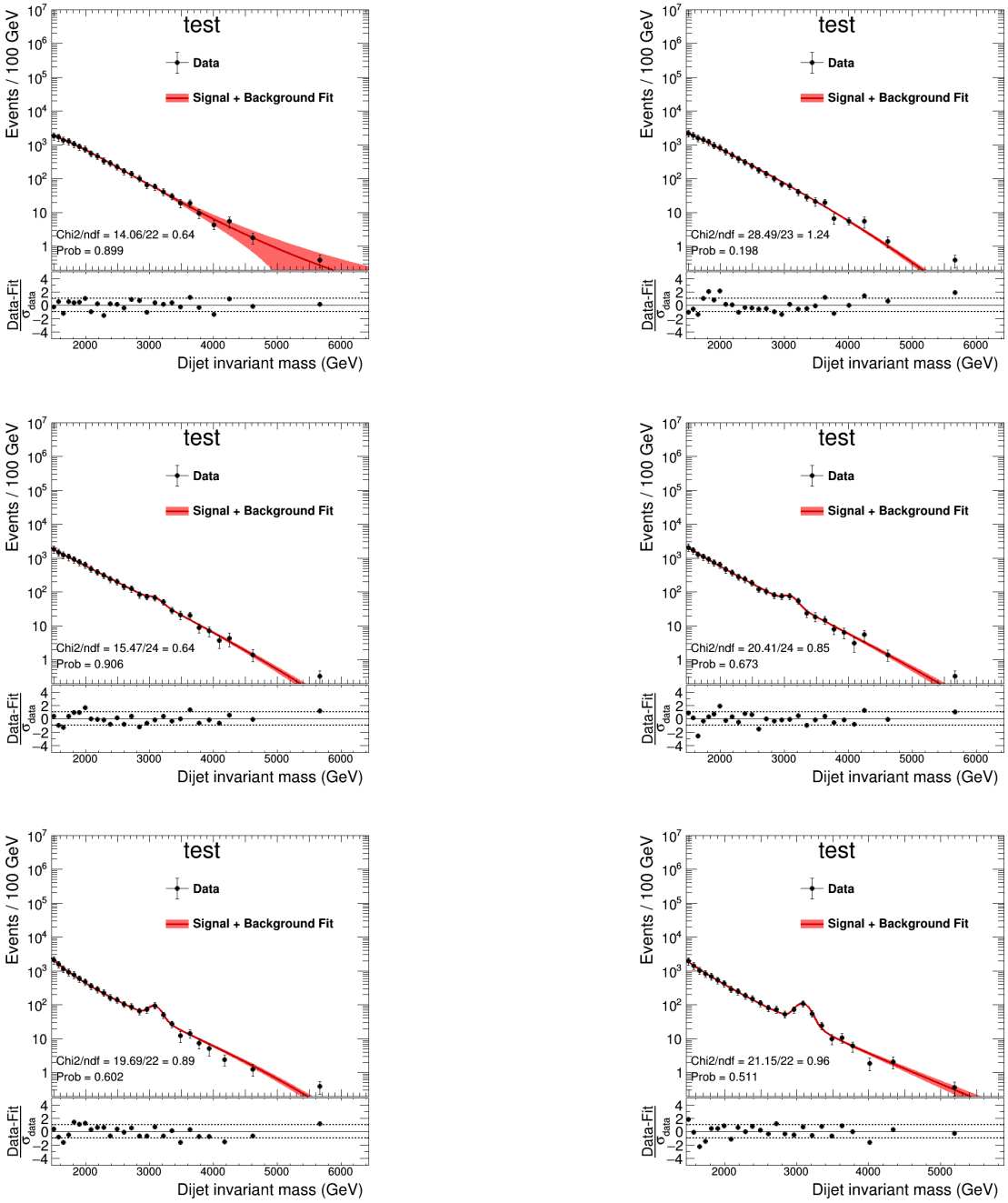


Figure D.5: Invariant dijet mass distribution of CATHODE-selected events and the combined signal plus background shape fit, using an increasing amount of injected signal events. The plots are based on the mock MC data set with events of the sigmal model $X \to YY'$, $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$, $m_{Y'} = 80\,\text{GeV}$ injected. The plots are ordered by increasing injected cross section in a top left to bottom right manner, starting from 5 fb in steps of 5 fb. The selection efficiency used was 1 %.

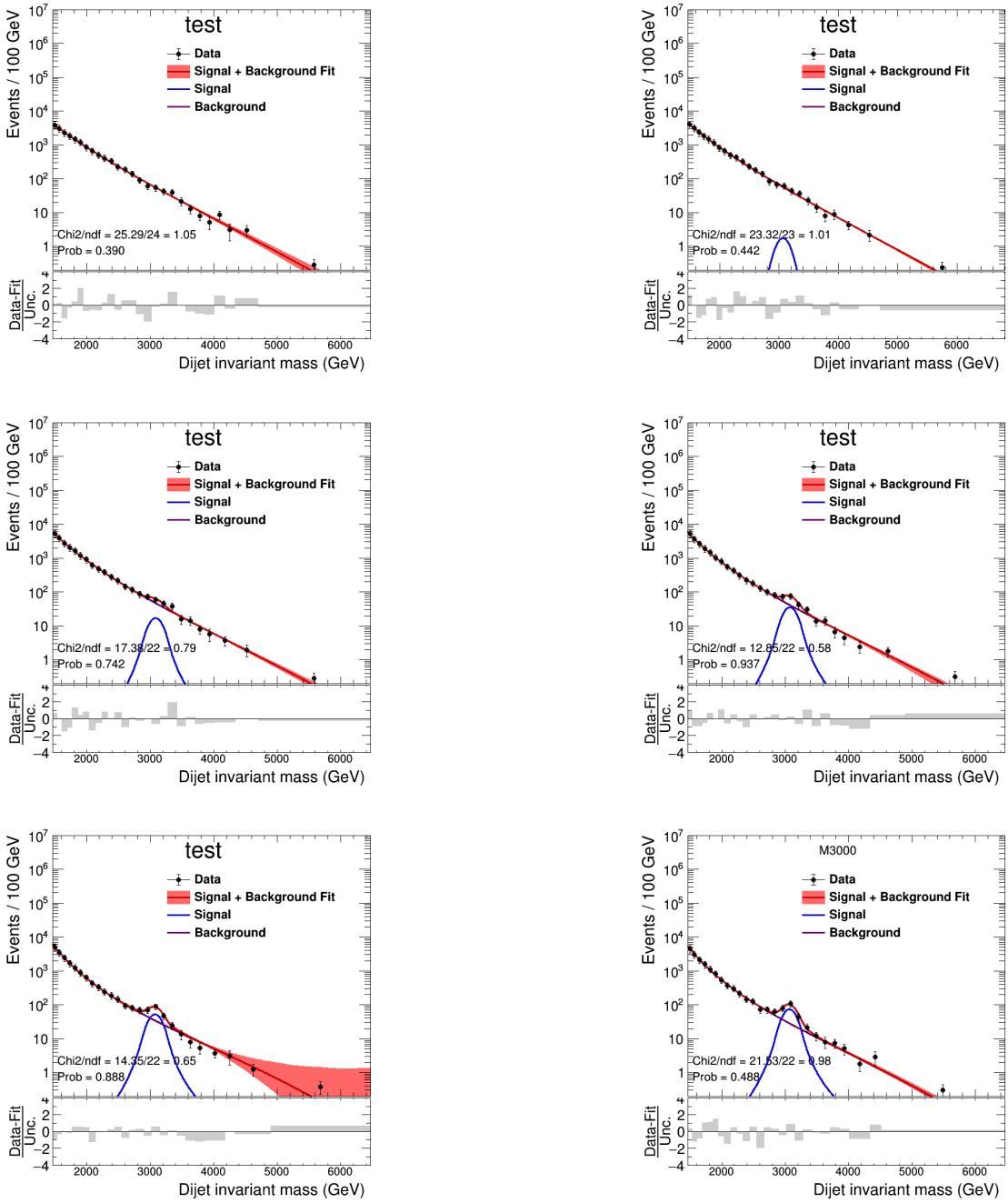


Figure D.6: Invariant dijet mass distribution of CATHODE-b-selected events and the combined signal plus background shape fit, using an increasing amount of injected signal events. The plots are based on the mock MC data set with events of the sigmal model $X \to YY'$, $m_X = 3\,\text{TeV}$, $m_Y = 170\,\text{GeV}$, $m_{Y'} = 80\,\text{GeV}$ injected. The plots are ordered by increasing injected cross section in a top left to bottom right manner, starting from 5 fb in steps of 5 fb. The selection efficiency used was 1 %.

Appendix E

Additional CATHODE(-b) results

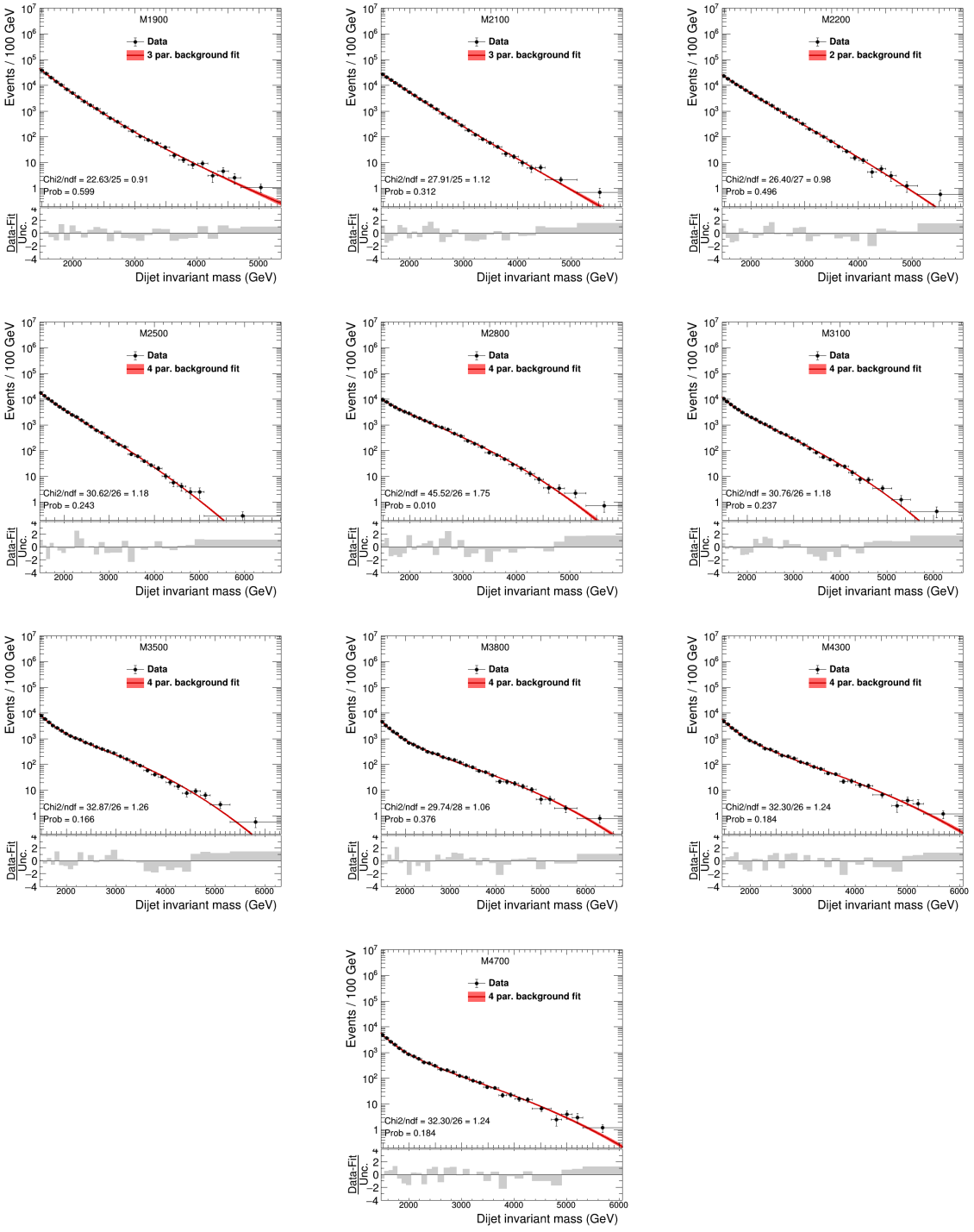


Figure E.1: The background only fit in every signal region of the CATHODE search. The plots for the signal region bins are ordered by the mass of their centre value in a top left to bottom right manner.The lower panel of each plot shows the pull distribution of the fit in each invariant mass bin.

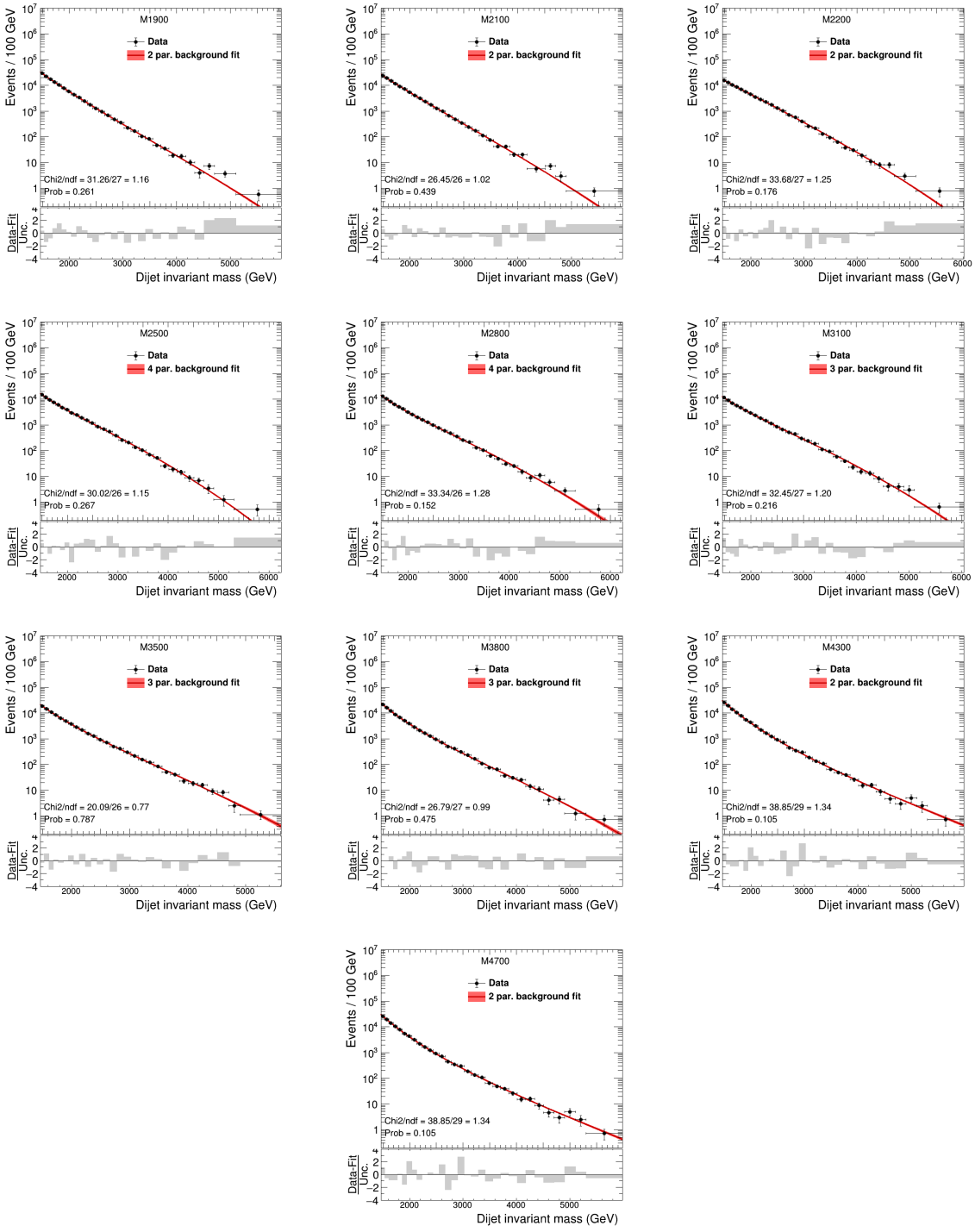


Figure E.2: The background only fit in every signal region of the CATHODE-b search. The plots for the signal region bins are ordered by the mass of their centre value in a top left to bottom right manner.The lower panel of each plot shows the pull distribution of the fit in each invariant mass bin.

Signal Model ($\mathbf{m_A}$ = **3 TeV**)	$\mathbf{m_B}/\mathbf{m_C}$ **(GeV)**	**Exp. (Obs.) Limit (fb)** CATHODE	**Improvement w.r.t Inclusive** CATHODE	**Exp. (Obs.) Limit (fb)** CATHODE-b	**Improvement w.r.t Inclusive** CATHODE-b
$Q^* \to qW'$	m_q/25	68.9 (753.5)	0.3	95.8 (167.9)	0.2
$Q^* \to qW'$	m_q/80	50.0 (95.2)	0.4	85.0 (128.3)	0.3
$Q^* \to qW'$	m_q/170	60.1 (105.5)	0.4	74.3 (113.1)	0.3
$Q^* \to qW'$	m_q/400	60.0 (92.0)	0.4	75.2 (112.6)	0.3
$X \to YY'$	25/25	8.0 (9.9)	0.9	17.0 (27.3)	0.4
$X \to YY'$	25/80	7.6 (13.2)	0.9	16.0 (18.0)	0.5
$X \to YY'$	25/170	10.3 (18.4)	0.7	16.9 (22.7)	0.4
$X \to YY'$	25/400	18.1 (22.4)	0.4	21.6 (31.1)	0.4
$X \to YY'$	80/80	4.2 (8.0)	1.6	8.5 (12.5)	0.8
$X \to YY'$	80/170	5.7 (11.4)	1.2	9.1 (13.9)	0.8
$X \to YY'$	80/400	6.0 (7.3)	1.2	9.6 (15.6)	0.8
$X \to YY'$	170/170	3.7 (6.8)	1.9	5.4 (7.7)	1.3
$X \to YY'$	170/400	5.7 (7.7)	1.4	7.1 (11.5)	1.1
$X \to YY'$	400/400	5.4 (6.2)	1.7	5.5 (7.5)	1.6
$W' \to B't \to bZt$	25	106.8 (295.8)	0.4	69.0 (81.1)	0.6
$W' \to B't \to bZt$	80	110.6 (280.2)	0.4	64.9 (79.7)	0.6
$W' \to B't \to bZt$	170	65.7 (76.6)	0.5	34.8 (45.9)	0.9
$W' \to B't \to bZt$	400	50.3 (123.8)	0.6	33.9 (43.3)	0.9
$W_{kk} \to RW \to 3W$	170	37.0 (55.0)	1.1	66.0 (119.6)	0.6
$W_{kk} \to RW \to 3W$	400	33.3 (47.0)	1.2	55.0 (93.3)	0.7
$Y \to HH \to 4t$	400	27.0 (75.5)	1.0	17.8 (17.8)	1.6

Table E.1: Expected and observed 95 % CL upper limits for different signal models at a resonance mass of 3 TeV on the respective cross section. Presented are only the results from CATHODE and CATHODE-b only. For reference, the improvement related to the expected limit of an inclusive search is also shown.

Signal Model (m_A = 5 TeV)	**m_B/m_C (GeV)**	**Exp. (Obs.) Limit (fb) CATHODE**	**Improvement w.r.t Inclusive CATHODE**	**Exp. (Obs.) Limit (fb) CATHODE-b**	**Improvement w.r.t Inclusive CATHODE-b**
$Q^* \to qW'$	m_q/25	21.5 (22.6)	0.1	24.1 (24.7)	0.1
$Q^* \to qW'$	m_q/80	16.0 (17.0)	0.2	22.0 (22.7)	0.1
$Q^* \to qW'$	m_q/170	7.8 (8.1)	0.4	14.2 (14.6)	0.2
$Q^* \to qW'$	m_q/400	5.7 (5.7)	0.5	7.1 (7.4)	0.4
$X \to YY'$	25/25	4.7 (4.5)	0.2	4.7 (4.6)	0.2
$X \to YY'$	25/80	4.8 (4.5)	0.2	5.6 (5.5)	0.2
$X \to YY'$	25/170	2.0 (1.9)	0.5	4.6 (4.5)	0.2
$X \to YY'$	25/400	1.1 (1.0)	0.8	1.7 (1.7)	0.5
$X \to YY'$	80/80	1.8 (1.7)	0.5	2.6 (2.6)	0.4
$X \to YY'$	80/170	1.5 (1.4)	0.6	2.4 (2.4)	0.4
$X \to YY'$	80/400	1.8 (1.7)	0.5	2.4 (2.3)	0.4
$X \to YY'$	170/170	0.7 (0.7)	1.2	1.3 (1.3)	0.7
$X \to YY'$	170/400	0.9 (0.8)	1.1	1.7 (1.6)	0.6
$X \to YY'$	400/400	0.5 (0.4)	2.1	0.9 (0.9)	1.1
$W' \to B't \to bZt$	25	23.6 (17.4)	0.3	9.2 (13.6)	0.7
$W' \to B't \to bZt$	80	19.5 (14.4)	0.3	8.8 (12.4)	0.7
$W' \to B't \to bZt$	170	10.6 (7.7)	0.5	6.9 (9.3)	0.8
$W' \to B't \to bZt$	400	10.8 (7.9)	0.5	5.4 (7.3)	1.0
$W_{kk} \to RW \to 3W$	170	18.1 (16.9)	0.3	23.9 (28.3)	0.3
$W_{kk} \to RW \to 3W$	400	58.5 (53.0)	0.1	13.0 (13.0)	0.4
$Y \to HH \to 4t$	400	3.5 (3.0)	1.2	1.9 (3.4)	2.2

Table E.2: Expected and observed 95 % CL upper limits for different signal models at a resonance mass of 5 TeV on the respective cross section. Presented are only the results from CATHODE and CATHODE-b only. For reference, the improvement related to the expected limit of an inclusive search is also shown.

Bibliography

[1] The CMS Collaboration, "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012.

[2] The ATLAS Collaboration, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 1–29, 2012.

[3] The CMS Collaboration, "MUSiC: a model-unspecific search for new physics in proton–proton collisions at $\sqrt{s} = 13$ TeV," *The European Physical Journal C*, vol. 81, p. 629, Jul 2021.

[4] CMS Collaboration, "MUSiC, a Model Unspecific Search for New Physics, in pp Collisions at $\sqrt{s} = 8$ TeV," tech. rep., CERN, Geneva, 2017. report no. CMS-PAS-EXO-14-016, `https://cds.cern.ch/record/2256653`.

[5] CMS Collaboration, "Model Unspecific Search for New Physics in pp Collisions at $\sqrt{s} = 7$ TeV," tech. rep., CERN, Geneva, 2011. report no. CMS-PAS-EXO-10-021, `https://cds.cern.ch/record/1360173`.

[6] ATLAS Collaboration, "A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment," *The European Physical Journal C*, vol. 79, no. 2, p. 120, 2019.

[7] ATLAS Collaboration, "A general search for new phenomena with the ATLAS detector in pp collisions at $\sqrt{s} = 8$ TeV," tech. rep., CERN, Geneva, 2014. report no. ATLAS-CONF-2014-006, `https://cds.cern.ch/record/1666536`.

[8] The ATLAS Collaboration, "A general search for new phenomena with the ATLAS detector in pp collisions at $\sqrt{s} = 7$ TeV.," tech. rep., CERN, Geneva, 2012. report no. ATLAS-CONF-2012-107, `https://cds.cern.ch/record/1472686`.

[9] J. H. Kim, K. Kong, B. Nachman, and D. Whiteson, "The motivation and status of two-body resonance decays after the LHC Run 2 and beyond," *Journal of High Energy Physics*, vol. 2020, p. 30, 2020.

[10] M. E. Peskin and D. V. Schroeder, *An introduction to quantum field theory*. Boulder, CO: Westview, 1995. Includes exercises.

[11] D. J. Griffiths, *Introduction to elementary particles; 2nd rev. version*. Physics textbook, New York, NY: Wiley, 2008.

[12] M. Thomson, *Modern particle physics*. New York: Cambridge University Press, 2013.

[13] R. L. Workman and Others, "Review of Particle Physics," *PTEP*, vol. 2022, p. 083C01, 2022.

[14] F. Englert and R. Brout, "Broken Symmetry and the Mass of Gauge Vector Mesons," *Phys. Rev. Lett.*, vol. 13, pp. 321–323, Aug 1964.

[15] P. W. Higgs, "Broken Symmetries and the Masses of Gauge Bosons," *Phys. Rev. Lett.*, vol. 13, pp. 508–509, Oct 1964.

[16] P. W. Higgs, "Broken symmetries, massless particles and gauge fields," *Phys. Lett.*, vol. 12, pp. 132–133, 1964.

[17] P. W. Higgs, "Spontaneous Symmetry Breakdown without Massless Bosons," *Phys. Rev.*, vol. 145, pp. 1156–1163, May 1966.

[18] F. Englert, R. Brout, and M. F. Thiry, "Vector mesons in presence of broken symmetry," *Nuovo Cim. A*, vol. 43, no. 2, pp. 244–257, 1966.

[19] The KTeV Collaboration, "Observation of direct CP violation in $K_{S,L} \to \pi\pi$ decays," *Phys. Rev. Lett.*, vol. 83, pp. 22–27, Jul 1999.

[20] The NA48 Collaboration, "A new measurement of direct CP violation in two pion decays of the neutral kaon," *Physics Letters B*, vol. 465, no. 1, pp. 335–348, 1999.

[21] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay, "Evidence for the 2π Decay of the K_2^0 Meson," *Phys. Rev. Lett.*, vol. 13, pp. 138–140, Jul 1964.

[22] A. O. Sushkov, W. J. Kim, D. A. R. Dalvit, and S. K. Lamoreaux, "New Experimental Limits on Non-Newtonian Forces in the Micrometer Range," *Phys. Rev. Lett.*, vol. 107, p. 171101, Oct 2011.

[23] J. F. Donoghue, "The effective field theory treatment of quantum gravity," *AIP Conference Proceedings*, vol. 1483, pp. 73–94, 10 2012.

[24] The Super-Kamiokande Collaboration, "Evidence for Oscillation of Atmospheric Neutrinos," *Phys. Rev. Lett.*, vol. 81, pp. 1562–1567, Aug 1998.

[25] T. Kajita, "Discovery of neutrino oscillations," *Reports on Progress in Physics*, vol. 69, p. 1607, may 2006.

[26] V. Trimble, "Existence and Nature of Dark Matter in the Universe," *Ann. Rev. Astron. Astrophys.*, vol. 25, pp. 425–472, 1987.

[27] G. Paál, I. Horváth, and B. Lukács, "Inflation and compactification from Galaxy redshifts?," *Astrophysics and Space Science*, vol. 191, no. 1, pp. 107–124, 1992.

[28] The Supernova Cosmology Project, "Measurements of Ω and Λ from 42 High-Redshift Supernovae," *The Astrophysical Journal*, vol. 517, p. 565, jun 1999.

[29] W. de Boer and C. Sander, "Global electroweak fits and gauge coupling unification," *Physics Letters B*, vol. 585, no. 3, pp. 276–286, 2004.

[30] N. Arkani–Hamed, S. Dimopoulos, and G. Dvali, "The hierarchy problem and new dimensions at a millimeter," *Physics Letters B*, vol. 429, no. 3, pp. 263–272, 1998.

[31] L. Fitzpatrick, J. Kaplan, L. Randall, and L.-T. Wang, "Searching for the Kaluza-Klein graviton in bulk RS models," *Journal of High Energy Physics*, vol. 2007, p. 013, sep 2007.

[32] The UA1 Collaboration, "Experimental observation of lepton pairs of invariant mass around 95 GeV/c^2 at the CERN SPS collider," *Physics Letters B*, vol. 126, no. 5, pp. 398–410, 1983.

[33] The UA2 Collaboration, "Evidence for $Z^0 \rightarrow e^+e^-$ at the CERN $\bar{p}p$ Collider," *Physics Letters B*, vol. 129, no. 1, pp. 130–140, 1983.

[34] The E288 Collaboration, "Observation of a Dimuon Resonance at 9.5 GeV in 400-GeV Proton-Nucleus Collisions," *Phys. Rev. Lett.*, vol. 39, pp. 252–255, Aug 1977.

[35] The SLAC-SP-017 Collaboration, "Discovery of a Narrow Resonance in e^+e^- Annihilation," *Phys. Rev. Lett.*, vol. 33, pp. 1406–1408, 1974.

[36] The E598 Collaboration, "Experimental Observation of a Heavy Particle *J*," *Phys. Rev. Lett.*, vol. 33, pp. 1404–1406, 1974.

[37] R. S. Chivukula and H. Georgi, "Composite Technicolor Standard Model," *Phys. Lett. B*, vol. 188, pp. 99–104, 1987.

[38] U. Baur, M. Spira, and P. M. Zerwas, "Excited-quark and -lepton production at hadron colliders," *Phys. Rev. D*, vol. 42, pp. 815–824, Aug 1990.

[39] K. Agashe, R. Contino, and A. Pomarol, "The minimal composite Higgs model," *Nuclear Physics B*, vol. 719, no. 1, pp. 165–187, 2005.

[40] D. Barducci, A. Belyaev, S. De Curtis, S. Moretti, and G. M. Pruna, "Exploring Drell-Yan signals from the 4D Composite Higgs Model at the LHC," *Journal of High Energy Physics*, vol. 2013, no. 4, p. 152, 2013.

[41] K. Agashe, P. Du, S. Hong, and R. Sundrum, "Flavor universal resonances and warped gravity," *Journal of High Energy Physics*, vol. 2017, no. 1, p. 16, 2017.

[42] K. Agashe, J. H. Collins, P. Du, S. Hong, D. Kim, *et al.*, "Dedicated strategies for triboson signals from cascade decays of vector resonances," *Phys. Rev. D*, vol. 99, p. 075016, Apr 2019.

[43] A. Carvalho, "Gravity particles from Warped Extra Dimensions, predictions for LHC," 2018. *ArXiv* preprint, `http://arxiv.org/abs/1404.0102`.

[44] N. Craig, P. Draper, K. Kong, Y. Ng, and D. Whiteson, "The unexplored landscape of two-body resonances," *Acta Phys. Polon. B*, vol. 50, p. 837, 2019.

[45] L. Evans and P. Bryant, "LHC Machine," *Journal of Instrumentation*, vol. 3, p. S08001, aug 2008.

[46] O. S. Bruning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, *et al.*, "LHC Design Report Vol.1: The LHC Main Ring," *CERN Yellow Reports: Monographs*, Jun 2004.

[47] The ALICE Collaboration, "Physics of the Muon Spectrometer of the ALICE Experiment," *Journal of Physics: Conference Series*, vol. 50, p. 361, Nov 2006.

[48] E. Mobs, "The CERN accelerator complex - August 2018. Complexe des accélérateurs du CERN - Août 2018," 2018. General Photo.

[49] The LHCb Collaboration, *LHCb : Technical Proposal*. Geneva: CERN, 1998.

[50] M. Pepe Altarelli and F. Teubert, "B Physics at LHCb," *Int. J. Mod. Phys. A*, vol. 23, pp. 5117–5136, 2008.

[51] The LHCb Collaboration, "Overview of LHCb," in *40th Rencontres de Moriond on QCD and High Energy Hadronic Interactions*, pp. 85–88, 6 2005.

[52] The ALICE Collaboration, "The ALICE experiment - A journey through QCD," tech. rep., CERN, Geneva, 2022. report no. CERN-EP-2022-227, `https://cds.cern.ch/record/2838475`.

[53] The ALICE Collaboration, *ALICE: Technical proposal for a Large Ion collider Experiment at the CERN LHC*. LHC technical proposal, Geneva: CERN, 1995.

[54] The ALICE Collaboration, "The ALICE experiment at the CERN LHC," *Journal of Instrumentation*, vol. 3, p. S08002, Aug 2008.

[55] The CMS Collaboration, *CMS, the Compact Muon Solenoid: technical proposal*. LHC technical proposal, Geneva: CERN, 1994.

[56] The CMS Collaboration, "The CMS experiment at the CERN LHC," *Journal of Instrumentation*, vol. 3, p. S08004, Aug 2008.

[57] I. Efthymiopoulos, "Overview of the ATLAS detector at LHC," Tech. Rep. 7, CERN, Geneva, 1999. report no. ATL-CONF-99-002, `https://cds.cern.ch/record/409257`.

[58] The ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *Journal of Instrumentation*, vol. 3, p. S08003, Aug 2008.

[59] The ATLAS Collaboration, *ATLAS: technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN*. LHC technical proposal, Geneva: CERN, 1994.

[60] The CMS Collaboration, "Observation of a new boson with mass near 125 GeV in pp collisions at $\sqrt{s}$ = 7 and 8 TeV," *Journal of High Energy Physics*, vol. 2013, no. 6, p. 81, 2013.

[61] I. Neutelings, "CMS coordinate system with the a cylindrical detector.," 2023. Accessed at: 05.04.2024, License: CC BY-SA 4.0 DEED, URL: `https://tikz.net/wp-content/uploads/2023/10/axis3D_CMS-004.png`.

[62] D. Barney, "CMS Detector Slice." CMS Collection, 2016.

[63] The Tracker Group of the CMS collaboration, "The CMS Phase-1 Pixel Detector Upgrade," *JINST*, vol. 16, no. 02, p. P02027, 2021.

[64] P. Azzurri, "The CMS Silicon Strip Tracker," *Journal of Physics: Conference Series*, vol. 41, p. 127, May 2006.

[65] The CMS Collaboration, *The CMS electromagnetic calorimeter project: Technical Design Report*. Technical design report. CMS, Geneva: CERN, 1997.

[66] F. Monti, "Performance of the CMS Electromagnetic Calorimeter in LHC Run2," tech. rep., CERN, Geneva, 2020. report no. CMS-CR-2020-038, `https://cds.cern.ch/record/2797453`.

[67] The CMS Collaboration, *The CMS hadron calorimeter project: Technical Design Report*. Technical design report. CMS, Geneva: CERN, 1997.

[68] F. Cavallari, "Performance of calorimeters at the LHC," *Journal of Physics: Conference Series*, vol. 293, p. 012001, Apr 2011.

[69] The CMS Collaboration, "Calibration of the CMS hadron calorimeters using proton-proton collision data at $\sqrt{s}$=13 TeV," *Journal of Instrumentation*, vol. 15, p. P05002, May 2020.

[70] The CMS Collaboration, *The CMS muon project: Technical Design Report*. Technical design report. CMS, Geneva: CERN, 1997.

[71] The CMS Collaboration, "Performance of the reconstruction and identification of high-momentum muons in proton-proton collisions at $\sqrt{s}$ = 13 TeV," *Journal of Instrumentation*, vol. 15, p. P02027, Feb 2020.

[72] Manfred Jeitler, "Upgrade of the trigger system of CMS," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 718, pp. 11–15, 2013. Proceedings of the 12th Pisa Meeting on Advanced Detectors.

[73] The CMS Collaboration, "The CMS Trigger System," *Journal of Physics: Conference Series*, vol. 2375, p. 012003, Nov 2022.

[74] The CMS Collaboration, "CMS. The TriDAS project. Technical design report, vol. 1: The trigger systems," Dec 2000.

[75] M. Tosi, "The CMS trigger in Run 2," *PoS*, vol. EPS-HEP2017, p. 523, 2017.

[76] M. Dordevic, "The CMS Particle Flow Algorithm," *EPJ Web Conf.*, vol. 191, p. 02016, 2018.

[77] The CMS Collaboration, "Particle-flow reconstruction and global event description with the CMS detector," *Journal of Instrumentation*, vol. 12, p. P10003, Oct 2017.

[78] W. Adam, B. Mangano, T. Speer, and T. Todorov, "Track Reconstruction in the CMS tracker," tech. rep., CERN, Geneva, 2006. report no. CMS-NOTE-2006-041, `https://cds.cern.ch/record/934067`.

[79] The CMS Collaboration, "Performance of photon reconstruction and identification with the CMS detector in proton-proton collisions at $\sqrt{s}$ =8 TeV," *JINST*, vol. 10, p. P08010, 2015.

[80] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-kt jet clustering algorithm," *Journal of High Energy Physics*, vol. 2008, p. 063, apr 2008.

[81] J. Gallicchio and M. D. Schwartz, "Quark and Gluon Tagging at the LHC," *Physical Review Letters*, vol. 107, Oct. 2011.

[82] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[83] J. Li and H. Sun, "An Attention Based Neural Network for Jet Tagging," 2020. *ArXiv* preprint, `http://arxiv.org/abs/2009.00170`.

[84] H. Carlens, "State of Competitive Machine Learning in 2022." `https://mlcontests.com/state-of-competitive-data-science-2022`.

[85] F. Galton, "Regression Towards Mediocrity in Hereditary Stature.," *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886.

[86] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.

[87] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[88] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning*. Springer, 2009.

[89] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, pp. 123–140, August 1996.

[90] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[91] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[92] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[93] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[94] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[95] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[96] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[97] R. A. Alzahrani and A. C. Parker, "Neuromorphic Circuits With Neural Modulation Enhancing the Information Content of Neural Signaling," in *International Conference on Neuromorphic Systems 2020*, ICONS 2020, (New York, NY, USA), Association for Computing Machinery, 2020.

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

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[100] F. Stahlberg, "Neural Machine Translation: A Review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 10 2020.

[101] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, *et al.*, "A Survey of Natural Language Generation," *ACM Comput. Surv.*, vol. 55, Dec 2022.

[102] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[103] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, *et al.*, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[104] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.

[105] HEP ML Community, "A Living Review of Machine Learning for Particle Physics."

[106] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462 – 466, 1952.

[107] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[108] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[109] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[110] H. J. KELLEY, "Gradient Theory of Optimal Flight Paths," *ARS Journal*, vol. 30, no. 10, pp. 947–954, 1960.

[111] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[112] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," in *International Conference on Learning Representations*, 2017.

[113] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, *et al.*, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour." *ArXiv* preprint, `http://arxiv.org/abs/1706.02677`.

[114] A. Y. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, (New York, NY, USA), p. 78, Association for Computing Machinery, 2004.

[115] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, (San Francisco, CA, USA), p. 950–957, Morgan Kaufmann Publishers Inc., 1991.

[116] S. Hanson and L. Pratt, "Comparing Biases for Minimal Network Construction with Back-Propagation," in *Advances in Neural Information Processing Systems* (D. Touretzky, ed.), vol. 1, Morgan-Kaufmann, 1988.

[117] A. Weigend, D. Rumelhart, and B. Huberman, "Generalization by Weight-Elimination with Application to Forecasting," in *Advances in Neural Information Processing Systems* (R. Lippmann, J. Moody, and D. Touretzky, eds.), vol. 3, Morgan-Kaufmann, 1990.

[118] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012. *ArXiv* preprint, `http://arxiv.org/abs/1207.0580`.

[119] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[120] L. Ruthotto and E. Haber, "An introduction to deep generative modeling," *GAMM-Mitteilungen*, vol. 44, no. 2, p. e202100008, 2021.

[121] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7327–7347, 2022.

[122] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[123] N. Aldausari, A. Sowmya, N. Marcus, and G. Mohammadi, "Video Generative Adversarial Networks: A Review," *ACM Comput. Surv.*, vol. 55, Jan 2022.

[124] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), p. 613–621, Curran Associates Inc., 2016.

[125] R. Goyal, P. Kumar, and V. P. Singh, "A Systematic Survey on Automated Text Generation Tools and Techniques: Application, Evaluation, and Challenges," *Multimedia Tools and Applications*, vol. 82, pp. 43089–43144, Nov 2023.

[126] S. Carrazza and F. A. Dreyer, "Lund jet images from generative and cycle-consistent adversarial networks," *The European Physical Journal C*, vol. 79, p. 979, Nov 2019.

[127] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasicezka, *et al.*, "CaloClouds: fast geometry-independent highly-granular calorimeter simulation," *Journal of Instrumentation*, vol. 18, p. P11025, Nov 2023.

[128] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res.*, vol. 22, Jan 2021.

[129] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked Autoregressive Flow for Density Estimation," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[130] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "MADE: Masked Autoencoder for Distribution Estimation," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 881–889, PMLR, 07–09 Jul 2015.

[131] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, "Asymptotic formulae for likelihood-based tests of new physics," *The European Physical Journal C*, vol. 71, no. 2, p. 1554, 2011.

[132] G. Kasieczka, B. Nachman, D. Shih, O. Amram, A. Andreassen, *et al.*, "The lhc olympics 2020 a community challenge for anomaly detection in high energy physics," *Reports on Progress in Physics*, vol. 84, p. 124201, Dec 2021.

[133] E. M. Metodiev, B. Nachman, and J. Thaler, "Classification without labels: Learning from mixed samples in high energy physics," *JHEP*, vol. 10, p. 174, 2017.

[134] J. H. Collins, K. Howe, and B. Nachman, "Extending the search for new resonances with machine learning," *Phys. Rev.*, vol. D99, no. 1, p. 014038, 2019.

[135] B. Nachman and D. Shih, "Anomaly Detection with Density Estimation," *Phys. Rev. D*, vol. 101, p. 075042, 2020.

[136] The CMS Collaboration, "Search for heavy resonances in the W/Z-tagged dijet mass spectrum in pp collisions at 7 TeV," *Phys. Lett.*, vol. B723, pp. 280–301, 2013.

[137] The CMS Collaboration, "Search for Narrow Resonances using the Dijet Mass Spectrum with 19.6fb-1 of pp Collisions at 8 TeV," Tech. Rep. CMS-PAS-EXO-12-059, CERN, Geneva, 2013. report no. CMS-PAS-EXO-12-059, `https://cds.cern.ch/record/1519066`.

[138] The CMS Collaboration, "Search for resonances in the dijet mass spectrum from 7 TeV pp collisions at CMS," *Physics Letters B*, vol. 704, no. 3, pp. 123–142, 2011.

[139] M. Frate, K. Cranmer, S. Kalia, A. Vandenberg-Rodes, and D. Whiteson, "Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes," 2017. *ArXiv* preprint, `http://arxiv.org/abs/1709.05681`.

[140] A. Gandrakota, A. Lath, A. V. Morozov, and S. Murthy, "Model selection and signal extraction using Gaussian Process regression," *Journal of High Energy Physics*, vol. 2023, no. 2, p. 230, 2023.

[141] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 physics and manual," *Journal of High Energy Physics*, vol. 2006, p. 026, May 2006.

[142] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, *et al.*, "An introduction to PYTHIA 8.2," *Computer Physics Communications*, vol. 191, pp. 159–177, 2015.

[143] J. de Favereau, C. Delaere, P. Demin, The DELPHES 3 collaboration, *et al.*, "DELPHES 3: a modular framework for fast simulation of a generic collider experiment," *Journal of High Energy Physics*, vol. 2014, p. 57, 2014.

[144] A. Mertens, "New features in Delphes 3," *Journal of Physics: Conference Series*, vol. 608, p. 012045, Apr 2015.

[145] M. Selvaggi, "DELPHES 3: A modular framework for fast-simulation of generic collider experiments," *Journal of Physics: Conference Series*, vol. 523, p. 012033, Jun 2014.

[146] G. Kasieczka, B. Nachman, and D. Shih, "R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge," Apr 2022. Zenodo data set, `https://doi.org/10.5281/zenodo.6466204`.

[147] J. Thaler and K. Van Tilburg, "Identifying boosted objects with N-subjettiness," *Journal of High Energy Physics*, vol. 2011, no. 3, p. 15, 2011.

[148] S. Catani, Y. Dokshitzer, M. Seymour, and B. Webber, "Longitudinally-invariant $k_\perp$-clustering algorithms for hadron-hadron collisions," *Nuclear Physics B*, vol. 406, no. 1, pp. 187–224, 1993.

[149] S. D. Ellis and D. E. Soper, "Successive combination jet algorithm for hadron collisions," *Phys. Rev. D*, vol. 48, pp. 3160–3166, Oct 1993.

[150] A. Hallin, J. Isaacson, G. Kasieczka, C. Krause, B. Nachman, *et al.*, "Classifying anomalies through outer density estimation," *Phys. Rev. D*, vol. 106, no. 5, p. 055006, 2022.

[151] J. A. Raine, S. Klein, D. Sengupta, and T. Golling, "CURTAINs for your sliding window: Constructing unobserved regions by transforming adjacent intervals," *Front. Big Data*, vol. 6, p. 899345, 2023.

[152] A. Andreassen, B. Nachman, and D. Shih, "Simulation Assisted Likelihood-free Anomaly Detection," *Phys. Rev. D*, vol. 101, no. 9, p. 095004, 2020.

[153] T. Golling, S. Klein, R. Mastandrea, and B. Nachman, "Flow-enhanced transportation for anomaly detection," *Phys. Rev. D*, vol. 107, p. 096025, May 2023.

[154] M. Freytsis, M. Perelstein, and Y. C. San, "Anomaly detection in the presence of irrelevant features," *Journal of High Energy Physics*, vol. 2024, no. 2, p. 220, 2024.

[155] DØ Collaboration, "Search for new physics in $e\mu X$ data at DØ using SLEUTH: A quasi-model-independent search strategy for new physics," *Phys. Rev. D*, vol. 62, p. 092004, Oct 2000.

[156] DØ Collaboration, "Quasi-model-independent search for new physics at large transverse momentum," *Phys. Rev. D*, vol. 64, p. 012004, Jun 2001.

[157] DØ Collaboration, "Quasi-Model-Independent Search for New High p_T Physics at DØ," *Phys. Rev. Lett.*, vol. 86, pp. 3712–3717, Apr 2001.

[158] H1 Collaboration, "A general search for new phenomena at HERA," *Physics Letters B*, vol. 674, no. 4, pp. 257–268, 2009.

[159] H1 Collaboration, "A general search for new phenomena in ep scattering at HERA," *Physics Letters B*, vol. 602, no. 1, pp. 14–30, 2004.

[160] CDF Collaboration, "Model-independent and quasi-model-independent search for new physics at CDF," *Phys. Rev. D*, vol. 78, p. 012002, Jul 2008.

[161] CDF Collaboration, "Global search for new physics with 2.0 fb^{-1} at CDF," *Phys. Rev. D*, vol. 79, p. 011101, Jan 2009.

[162] ATLAS Collaboration, "Dijet Resonance Search with Weak Supervision Using $\sqrt{s} = 13$ TeV *pp* Collisions in the ATLAS Detector," *Phys. Rev. Lett.*, vol. 125, p. 131801, Sep 2020.

[163] ATLAS Collaboration, "Search for New Phenomena in Two-Body Invariant Mass Distributions Using Unsupervised Machine Learning for Anomaly Detection at $\sqrt{s} = 13$ TeV with the ATLAS Detector," *Phys. Rev. Lett.*, vol. 132, p. 081801, Feb 2024.

[164] S. Chekanov, "Imaging particle collision data for event classification using machine learning," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 931, pp. 92–99, 2019.

[165] The CMS Collaboration, "Model-agnostic search for dijet resonances with anomalous jet substructure in proton-proton collisions at $\sqrt{s}$= 13 TeV," tech. rep., CERN, Geneva, 2024. report no. CMS-PAS-EXO-22-026, `https://cds.cern.ch/record/2892677`.

[166] CMS Collaboration, "A multi-dimensional search for new heavy resonances decaying to boosted WW, WZ, or ZZ boson pairs in the dijet final state at 13 TeV," *The European Physical Journal C*, vol. 80, no. 3, p. 237, 2020.

[167] M. Farina, Y. Nakai, and D. Shih, "Searching for new physics with deep autoencoders," *Phys. Rev. D*, vol. 101, p. 075021, Apr 2020.

[168] T. Heimel, G. Kasieczka, T. Plehn, and J. M. Thompson, "QCD or what?," *SciPost Phys.*, vol. 6, p. 030, 2019.

[169] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant, "Variational autoencoders for new physics mining at the Large Hadron Collider," *Journal of High Energy Physics*, vol. 2019, no. 5, p. 36, 2019.

[170] A. Blance, M. Spannowsky, and P. Waite, "Adversarially-trained autoencoders for robust unsupervised new physics searches," *Journal of High Energy Physics*, vol. 2019, no. 10, p. 47, 2019.

[171] J. Hajer, Y.-Y. Li, T. Liu, and H. Wang, "Novelty detection meets collider physics," *Phys. Rev. D*, vol. 101, p. 076015, Apr 2020.

[172] K. A. Woźniak, O. Cerri, J. M. Duarte, T. Möller, J. Ngadiuba, *et al.*, "New Physics Agnostic Selections For New Physics Searches," *EPJ Web Conf.*, vol. 245, p. 06039, 2020.

[173] J. Collins, K. Howe, and B. Nachman, "Anomaly Detection for Resonant New Physics with Machine Learning," *Phys. Rev. Lett.*, vol. 121, p. 241803, Dec 2018.

[174] J. H. Collins, K. Howe, and B. Nachman, "Extending the search for new resonances with machine learning," *Phys. Rev. D*, vol. 99, p. 014038, Jan 2019.

[175] O. Amram and C. M. Suarez, "Tag N' Train: a technique to train improved classifiers on unlabeled data," *Journal of High Energy Physics*, vol. 2021, no. 1, p. 153, 2021.

[176] S. E. Park, D. Rankin, S.-M. Udrescu, M. Yunus, and P. Harris, "Quasi Anomalous Knowledge: Searching for new physics with embedded knowledge," *JHEP*, vol. 21, p. 030, 2020.

[177] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[178] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, *et al.*, "Pyro: Deep Universal Probabilistic Programming," *Journal of Machine Learning Research*, 2018.

[179] J. H. Kim, K. Kong, B. Nachman, and D. Whiteson, "The motivation and status of two-body resonance decays after the LHC Run 2 and beyond," *JHEP*, vol. 04, p. 030, 2020.

[180] K. Cranmer, "Searching for New Physics: Contributions to LEP and the LHC," 2005. Presented on 01 Nov 2005.

[181] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[182] A. J. Larkoski, S. Marzani, G. Soyez, and J. Thaler, "Soft Drop," *JHEP*, vol. 05, p. 146, 2014.

[183] S. Macaluso and D. Shih, "Pulling Out All the Tops with Computer Vision and Deep Learning," *JHEP*, vol. 10, p. 121, 2018.

[184] The CMS collaboration, "CMS Luminosity - Public Results," 2023. Accessed at: 05.04.2024, URL: `https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults`.

[185] The CMS collaboration, "Public CMS Data Quality Information," 2024. Accessed at: 05.04.2024, URL: `https://twiki.cern.ch/twiki/bin/view/CMSPublic/DataQuality`.

[186] C. Bierlich, S. Chakraborty, N. Desai, L. Gellersen, I. Helenius, *et al.*, "A comprehensive guide to the physics and usage of PYTHIA 8.3," 2022. *ArXiv* preprint, `http://arxiv.org/abs/2203.11601`.

[187] S. Frixione, P. Nason, and C. Oleari, "Matching NLO QCD computations with Parton Shower simulations: the POWHEG method," *JHEP*, vol. 11, p. 070, 2007.

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

[189] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, *et al.*, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations," *JHEP*, vol. 07, p. 079, 2014.

[190] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer, "MadGraph 5 : Going Beyond," *JHEP*, vol. 06, p. 128, 2011.

[191] S. Agostinelli *et al.*, "GEANT4–a simulation toolkit," *Nucl. Instrum. Meth. A*, vol. 506, pp. 250–303, 2003.

[192] I. Zoi *et al.*, "A multi-dimensional search for new heavy resonances decaying to boosted WW, WZ, ZZ, WH or ZH boson pairs in the all jets final state at 13 TeV," CMS Note 2019/131, The CMS Collaboration, 2021.

[193] HiggsWG, "Single top cross sections." `https://twiki.cern.ch/twiki/bin/viewauth/CMS/SummaryTable1G25ns#Single_top`.

[194] R. D. Ball, L. Del Debbio, S. Forte, A. Guffanti, J. I. Latorre, *et al.*, "A first unbiased global NLO determination of parton distributions and their uncertainties," *Nucl. Phys. B*, vol. 838, p. 136, 2010.

[195] R. D. Ball *et al.*, "Parton distributions for the LHC Run II," *JHEP*, vol. 04, p. 040, 2015.

[196] R. D. Ball *et al.*, "Parton distributions from high-precision collider data," *Eur. Phys. J. C*, vol. 77, p. 663, 2017.

[197] The CMS Collaboration, "Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements," *Eur. Phys. J. C*, vol. 80, p. 4, 2020.

[198] The CMS Collaboration, "Tracking and Primary Vertex Results in First 7 TeV Collisions," CMS Physics Analysis Summary CMS-PAS-TRK-10-005, CERN, 2010. report no. CMS-PAS-TRK-10-005, `http://cds.cern.ch/record/1279383`.

[199] The CMS Collaboration, "Particle-flow reconstruction and global event description with the CMS detector," *JINST*, vol. 12, p. P10003, 2017.

[200] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *JHEP*, vol. 04, p. 063, 2008.

[201] M. Cacciari, G. P. Salam, and G. Soyez, "FastJet User Manual," *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[202] D. Bertolini, P. Harris, M. Low, and N. Tran, "Pileup Per Particle Identification," *JHEP*, vol. 10, p. 59, 2014.

[203] The CMS collaboration, "Jet Identification for the 13 TeV UL data," 2023. Accessed at: 06.04.2024, URL: `https://twiki.cern.ch/twiki/bin/viewauth/CMS/JetID13TeVUL`.

[204] The CMS Collaboration, "Determination of Jet Energy Calibration and Transverse Momentum Resolution in CMS," *JINST*, vol. 6, p. P11002, 2011.

[205] The CMS Collaboration, "Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV," *Journal of Instrumentation*, vol. 13, p. P05011, May 2018.

[206] The CMS Collaboration, "Lund Plane Reweighting for Jet Substructure Correction," Tech. Rep. CMS-DP-2023-046, Geneva, 2023. report no. CMS-DP-2023-046, `https://cds.cern.ch/record/2866330`.

[207] The CMS Jet Energy Resolution and Corrections (JERC) Group, "Recommended Jet Energy Corrections and Uncertainties For Data and MC." URL: `https://twiki.cern.ch/twiki/bin/view/CMS/JECDataMC`.

[208] The CMS Collaboration, "Jet energy scale and resolution measurement with Run 2 Legacy Data Collected by CMS at 13 TeV," *CMS Detector Performance Summaries*, 2021. Report number: CMS-DP-2021-033; CERN-CMS-DP-2021-033.

[209] The CMS Collaboration, "Calibration of the Jet Mass Scale using boosted W bosons and top quarks," *CMS Detector Performance Summaries*, 2023. Report number: CMS-DP-2023-044; CERN-CMS-DP-2023-044.

[210] The CMS Physics Data and Monte Carlo Validation (PdmV) Group, "Utilities for Accessing Pileup Information for Data." URL: https://twiki.cern.ch/twiki/bin/view/CMS/PileupJSONFileforData.

[211] The CMS Detector Performance Group (DPG), "Reweighting recipe to emulate Level 1 ECAL prefiring.." URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/L1ECALPrefiringWeightRecipe.

[212] The CMS b Tag & Vertexing Physics Object Group (BTV-POG), "B-tagging discriminant shape calibration using event weights with a tag-and-probe method.." URL: https://twiki.cern.ch/twiki/bin/view/CMS/BTagShapeCalibration.

[213] The CMS Luminosity Physics Object Group, "Luminosity." URL: https://twiki.cern.ch/twiki/bin/view/CMS/TopLuminositySandbox.

[214] S. Mrenna and P. Skands, "Automated parton-shower variations in pythia 8," *Phys. Rev. D*, vol. 94, p. 074005, Oct 2016.

[215] The CMS Top Physics Object Group (TOP-POG), "The modeling of the top quark p_T." URL: https://twiki.cern.ch/twiki/bin/view/CMS/TopPtReweighting.

[216] T. Finke, M. Hein, G. Kasieczka, M. Krämer, A. Mück, *et al.*, "Tree-based algorithms for weakly supervised anomaly detection," *Phys. Rev. D*, vol. 109, p. 034033, Feb 2024.

[217] T. Finke, M. Hein, G. Kasieczka, M. Krämer, A. Mück, *et al.*, "Tree-Based Algorithms for Weakly Supervised Anomaly Detection," in *37th conference on Neural Information Processing Systems (NeurIPS)*, Machine Learning and the Physical Sciences Workshop, 2023.

[218] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?," 2022. *ArXiv* preprint, http://arxiv.org/abs/2207.08815.

[219] D. C. McElfresh, S. Khandagale, J. Valverde, V. P. C, G. Ramakrishnan, M. Goldblum, and C. White, "When Do Neural Nets Outperform Boosted Trees on Tabular Data?," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[220] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[221] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, SciPy 1.0 Contributors, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[222] R. Brun and F. Rademakers, "ROOT - An Object Oriented Data Analysis Framework," *Nucl. Inst. & Meth. in Phys. Res. A*, vol. 389, pp. 81–86, Sep. 1997. Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996.

[223] A. Hallin, G. Kasieczka, T. Quadfasel, D. Shih, and M. Sommerhalder, "Resonant anomaly detection without background sculpting," *Phys. Rev. D*, vol. 107, p. 114012, Jun 2023.