UNIVERSITY OF SOUTHAMPTON

Faculty of Science
School of Physics and Astronomy

Extracting Jet Signals of New Higgs Physics: from Traditional Analysis to Machine Learning

by

Henry Ann Day-Hall
MPhys
ORCiD: 0000-0002-9710-2980

A thesis for the degree of
Doctor of Philosophy

December 2021

University of Southampton

Abstract

Faculty of Science
School of Physics and Astronomy

Doctor of Philosophy

Extracting Jet Signals of New Higgs Physics: from Traditional Analysis to Machine Learning

by Henry Ann Day-Hall

This thesis investigates possible parameter values of, and optimum jet reconstruction for the signals from, the two Higgs doublet model (2HDM). Possible parameter values are investigated by way of recasting a parameter scan published by ATLAS. The original analysis, performed with 36.1 fb^{-1} of run 2 data, investigated the possibility of observing cascade decays from the 2HDM. The study considered the process $A \to ZH \to l^+l^-b\bar{b}$ (where $l = e, \mu$) in the context of the standard four Yukawa types. A parameter space in the physical basis of the 2HDM is explored, seeking parameter combinations that are not forbidden by theoretical constraints or existing observations, and to which a detector would be sensitive. The existing study is recast in two directions, firstly the possibility of exchanging A and H is investigated. Secondly, the extrapolation to run 3 is calculated. Under exchange of H and A all detectable parameter combinations are forbidden. More promisingly, however, it is seen that run 3 will offer sensitivity to considerable areas of permissible parameter space. It is clear that these decay channels already offer potential for finding the 2HDM at the LHC. Another line of investigation that might compliment this, is the potential to improve sensitivity by better signal reconstruction techniques. In particular, jet reconstruction techniques that might expand sensitivity to cascade decays from the 2HDM ending in a four b-quark final state are sort. Firstly, the challenges of reconstructing these states with existing algorithms is evaluated, and the limitations posed by cuts in the trigger illustrated. A comparison is made between the prevalent anti-k_T algorithm and a somewhat unusual algorithm termed variable-R. This finds that variable-R performs this task best, both in terms of mass peak reconstruction, and jet multiplicity. The second investigation into optimum jet construction aims to apply a novel method, spectral clustering, to the jet formation problem. Again, it is driven by an interest in reconstructing cascade decays from the 2HDM. This method proves to be insensitive to infra-red singularities in a practical sense. It is also shown to be very flexible, capable of clustering a range of signal types, without requiring alterations to its parameter settings.

Contents

List of Figures

List of Tables

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as: [1], [2] and [3]

Signed:..... Date:.....

Acknowledgements

Firstly, I owe a great deal to my supervisors; Stefano Moretti, Srinandan Dasmahapatra and Claire Shepherd-Themistocleous. They have offered me all the support and opportunities that a student could hope for, and I am hugely grateful for this.

Thanks also, to many fellow students who provided helpful conversation, collaboration and advice; Billy Ford, Souad Semlali, Giorgio Cerro, Shubhani Jain and Jacan Chaplais.

Thanks to the senior academics who offered wisdom, training and critique. There are more people that I could mention here, but thanks in particular to; Emmanuel Olaiya, Rachid Benbrik, Amit Chakraborty, Ian Hawk, Harri Waltari and Jim Pivarski.

The University of Southampton is blessed with generous technical and administrative teams. To Jacqui Bonnin, Elena Vataga and Alister Boags, many thanks for all your support.

For computational resources, I am in debt to the IRIDIS High Performance Computing Facility, and associated support services, at the University of Southampton. For funding, I am suported by the NExT institute, and the NGCM centre for doctoral training.

Thank you to my parents, Donna Day Lafferty and Ben Hall, and my partner, Ūla Mazūraitė for invaluable personal support.

Finally, I must thank David Fisher, for inspiring me a long time ago. To paraphrase his wisdom; *"On the far side of the universe there may be aliens we can never meet. They won't have Mozart, or Shakespeare, perhaps not even paintings or flowers. But however different we are, we are made by the same physics. Finding beauty in physics unifies us."*

Definitions and Abbreviations

θ	Polar angle
ϕ	Azimuthal angle
σ	Standard deviation
m	Mass
p	Momentum
p_x	Momentum in the x direction
p_y	Momentum in the y direction
p_z	Momentum in the z direction
p_T	Transverse momentum
E	Energy
$\sqrt{s}$	Centre of mass energy
η	Pseudorapidity
y	Rapidity (or the y coordinate, context dependant)
$\delta_{i,j}$	The Kronecker delta
$\delta x_{i,j}$	The displacement between x_i and x_j, in the relevant coordinate system
eV	Electron Volts
MeV	Mega electron Volts
GeV	Giga electron Volts
TeV	Tera electron Volts
fb	Femto barns
SM	Standard Model
MSSM	Minimal Supersymmetric Standard Model
BSM	Beyond Standard Model
2HDM	2 Higgs Doublet Model
CP	Charge Parity
FCNC	Flavour Changing Neutral Current
EWPO	Electro Weak Precision Observable
IR	Infra Red
LO	Leading Order
NLO	Next to Leading Order
QCD	Quantum Chromo Dynamics
SV	Secondary Vertex

LHC	Large Hadron Collider
CMS	Compact Muon Solenoid
MC	Monte Carlo
ML	Machine Learning
BDT	Boosted Decision Tree
NN	Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit

Chapter 1

Introduction

A quiet excitement can be felt in the hunt for an extended Higgs sector. Such a discovery would be very welcome as a great number of questions remain unanswered by the SM, and do not appear to be answerable without additional particles. Experimental questions include the muon's anomalous magnetic moment and the elusive interactions of dark matter. On the theoretical side, there are multiple hierarchy problems, the fermion mass spectrum and that of the Higgs field. The fermion mass spectrum ranges over 5 orders of magnitude, an explanation for these varying scales would be desirable. The Hierarchy problem for the Higgs is of a larger order again, there is no symmetry that protects the Higgs from receiving corrections to its mass from energy scales all the way up to the Plank mass. So for the Higgs to obtain a mass of 125 GeV there must be cancellations in the radiative corrections across 17 orders of magnitude. Gravity also has no place in the SM, and needs to be written into the interactions of massive particles. Furthermore, the universe is predominantly matter, with a surprising absence of antimatter, and a new model is wanted that could generate this asymmetry. Additional Higgs particles are not likely to answer all of these directly, but they would give good indications of where next to look. There are no guarantees that there are additional Higgs particles to be found, and it is the uncertainly of the outcome that makes the challenge so gripping.

This thesis concerns two aspects of the search for additional Higgs particles. Firstly, identifying parameter space that is not ruled out by existing observation, yet for which a signal might soon be detected, and secondly identifying jet formation techniques that would be most sensitive to signals produced by that model. It is focused on the 2HDM, although some methods are more generally applicable.

The first aim, to find parameters of interest, looks at the influence of parameter choices on the expected signal. To date, no such signals have been observed, so for any given signal, either the observations match the SM predictions, or they have yet to be measured. Of those that have yet to be measured, only some are possible to measure given

the sensitivity of the detectors we have now, or expect to create in the near future. This is the subset of parameter choices offering the potential for a discovery. Identifying them will guide searches, and help tailor the tools designed for this search.

Once the signals that the 2HDM might produce are described, the tools used to select signals in data may be optimised to locate them. One step in the analysis process is the formation of jets. A jet is group of particles, defined by some clustering mechanism, which is thought to originate from the decay of a single object in the hard event. By correctly gathering such groups, properties of the decayed object can be reconstructed. The most common decay channel for a Higgs is a pair of b-quarks, which decay quickly into a great many particles. As such, jet formation, in particular b-jet formation, is important for finding signals in Higgs physics.

The 2HDM would produce events with different topologies and kinematics than those of the SM Higgs. Cascade decays may produce four-jet events, and depending on the mass difference between the Higgs particles of the model, these events may be significantly boosted. The mass of the decaying Higgs will also influence the kinematics of any b-quarks produced, which in turn modifies the distributions for the decay products. So there may well be room for improving on the strategies used to define jets in these events.

The outline of this thesis is as follows; in chapter 2, a review of the 2HDM is given as general background. A study of the implications of the decays $A \to ZH$ and $H \to ZA$ for the models parameter space is presented in chapter 3. Following this, chapter 4 provides an overview of jet physics, starting from the collision point, following the journey of the particles through the collider, and then ending with the business of jet formation itself. This gives the background for chapter 5, which presents a study exploring the application of existing methods of jet formation to this signal. In chapter 6 and chapter 7 there are reviews of machine learning in jet physics, and clustering techniques in machine learning, respectively. These are presented as two complimentary views on the possibilities for new, machine learning driven, jet formation techniques. In the penultimate chapter, chapter 8, a machine learning technique, known as spectral clustering, is presented as a means for jet formation. Finally, there is a conclusion.

Chapter 2

Parameters of the Two Higgs Doublet Model

Much of this thesis concerns the two 2HDM. In the next chapter, a scan of the parameter space of the 2HDM is presented, and in the subsequent chapters, signals of the 2HDM are studied. So in preparation for this, this chapter presents a review of the theory for the 2HDM and some justification for taking an interest in it.

2.1 Review of 2HDM

Many models for additional Higgs particles exist, the 2HDM enjoys the status of being the simplest of all of these. For that reason alone one might take a particular interest in the 2HDM. Although this thesis focuses on phenomenology, some review of the theory, is an important starting point. Particular attention will be paid to the motivations for the model and production and decay modes for the particles it introduces.

A cursory description of the SM and the SM Higgs is good preparation for a summary of the 2HDM. The SM Higgs must certainly appear in the 2HDM, and so its interactions should be kept in mind. Furthermore, a review of the SM is a good way to highlight some of the questions that motivate the development of new models, such as the 2HDM. Once this is complete, the 2HDM will be broached.

2.1.1 Standard Model

Development towards the SM could be said to have started in 1897, with the discovery of the electron [4]. The electron is a fundamental particle, that is to say, to the best of our knowledge it is not composed of any smaller parts. It was the first of 18 particles

	Generation I	II	III	I_3	Y	Q
Leptons	$\begin{pmatrix}\nu_e\\ e\end{pmatrix}_L$	$\begin{pmatrix}\nu_\mu\\ \mu\end{pmatrix}_L$	$\begin{pmatrix}\nu_\tau\\ \tau\end{pmatrix}_L$	$+1/2$	$-1/2$	0
	e_R	μ_R	τ_R	0	-1	-1
Quarks	$\begin{pmatrix}u\\ d\end{pmatrix}_L$	$\begin{pmatrix}t\\ b\end{pmatrix}_L$	$\begin{pmatrix}c\\ s\end{pmatrix}_L$	$+1/2$	$1/6$	$2/3$
	u_R	t_R	c_R	0	$2/3$	$2/3$
	d_R	b_R	s_R	0	$-1/3$	$-1/3$

TABLE 2.1: Table of electroweak quantum numbers for the fermions. I_3 is the weak isospin, Y is the weak hypercharge and Q is the electric charge. This table is inspired by the presentation in [6], but maintains the convention $Y = Q - I_3$, as in [5].

that are today considered to be fundamental building blocks. The interactions of these particles are summarised in the SM Lagrangian. Terms in the Lagrangian are built up from fields, the excitations of which are the particles, and the derivatives of those fields. Each term of Lagrangian will be described in the following two sections, with notation matching that of [5].

Of the 18 particles in the SM, there are the 12 fermions, often referred to as "matter particles". They are named for their half integer spin, which means they obey Fermi-Dirac statistics. Particles with Fermi-Dirac statistics cannot occupy the same energy state, and so behave in a solid, matter-like, way. The 12 fermions can be further divided into 6 quarks and 6 leptons, each of which contains 3 generations. In the quarks these generations are up and down, top and bottom, charm and strange. In the leptons each generation is composed of an electron, muon or tau and its associated neutrino. Kinetic and interactions terms of the fermion's Lagrangian can be written as

$$\mathcal{L}_{\text{fermion}} = \sum_{\text{quarks}} i\bar{q}\gamma^\mu D_\mu q + \sum_{\psi_L} \bar{\psi}_L \gamma^\mu D_\mu \psi_L + \sum_{\psi_R} \bar{\psi}_R \gamma^\mu D_\mu \psi_R \tag{2.1}$$

where q represents the quark fields, ψ_L represents the left handed lepton fields, ψ_R represents the right handed lepton fields and D represents the appropriate covariant derivative. There are no right handed neutrinos in the SM, and so the final sum over ψ_R only includes electrons, muons and taus. This will be important to remember when the mass terms are added later, as they require mixing the left and right handed components, thus, neutrinos in the SM are massless. Electroweak quantum numbers for the fermions are given in Table 2.1.

After the 12 fermions, there are 6 are bosons, which act as force carriers. They have integer spin, and therefore obey Bose-Einstein statistics. Of the fundamental bosons, all but the Higgs have spin 1. The SM Higgs has spin 0, and will be discussed in more depth in the next section. To start off, the photon (denoted γ) is the mediator of the electromagnetic force, and couples to all charged particles. The $W^\pm$ and Z bosons

		I_3	Y	Q
Gauge Bosons	γ	0	0	0
	Z	0	0	0
	$W^{\pm}$	∓ 1	0	± 1

TABLE 2.2: Table of electroweak quantum numbers for the bosons, as in Table 2.1.

mediate the weak force, and couple to all left handed fermions, photons and each other. To complete the set of gauge bosons, there are the gluons, which mediate the strong force. Gluons couple to themselves, and to the quarks, due to their colour charge. The self interaction and kinetic terms for these gauge bosons are

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}W^i_{\mu\nu}W^{\mu\nu i} - \frac{1}{4}G^a_{\mu\nu}G^{\mu\nu a}. \tag{2.2}$$

The first two terms correspond to the electroweak sector, containing the photon, $W^{\pm}$ and Z bosons. Gauge bosons have somewhat simpler quantum numbers, these are given in Table 2.2.

At this point, there are still two significant section of the SM Lagrangian missing from this account; the Higgs sector, $\mathcal{L}_{\text{Higgs}}$, and the Yukawa sector, $\mathcal{L}_{\text{Yukawa}}$. These parts of the SM are of most interest to this thesis. In the next section they are explored.

2.1.2 Standard Model Higgs

The Higgs field is a weak isospin doublet, that is to say, it is comprised of two fields, which have an $SU(2)$ symmetry giving them the same interactions with the weak force. This can be written with four components;

$$\phi = \frac{1}{\sqrt{2}}\begin{pmatrix}\phi_1 + i\phi_2 \\ \phi_3 + i\phi_4\end{pmatrix} \tag{2.3}$$

It has weak isospin $I_3 = -1/2$ and weak hypercharge $Y = 1/2$. The potential energy of the Higgs field is

$$V(\phi) = \mu^2\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2. \tag{2.4}$$

Where $\mu^2 < 0$ and $\lambda > 0$. This gives the classic wine-bottle potential. The Higgs sector of the SM Lagrangian can now be written as $\mathcal{L}_{\text{Higgs}} = (D_\mu\phi)^\dagger D_\mu\phi - V(\phi)$, with D_μ being the covariant derivative.

Exploiting the freedom to choose the axes, the symmetry can be broken to

$$\phi_0 = \frac{1}{\sqrt{2}}\begin{pmatrix}0 \\ v\end{pmatrix}, \tag{2.5}$$

where v is the vacuum expectation value (vev); $v = \frac{|\mu|}{\sqrt{\lambda}}$. This choice corresponds to the unitarity gauge. With this choice fluctuations about the minimum are written as $\phi(x) = \phi_0 + h(x)$, and $h(x)$ is the scalar Higgs field. While there are 4 parameters available in Equation 2.3, three of these give mass to $W^{\pm}$ and Z bosons, and only one remains to form the Higgs field.

Substituting this vev back into Equation 2.4 it can be seen that the mass of the Higgs field (coefficient of the h^2 term) is $M_H^2 = 2\lambda v^2$, $M_H = \sqrt{2}|\mu|$.

Finally, the Higgs field also features in the Yukawa sector of the SM Lagrangian. With a Higgs field it is possible to formulate mass terms for the massive gauge bosons and fermions without violating unitarity [7]. This section provides masses for the massive particles and also couples the Higgs to the massive particles. Being a spin 0 particle, it must couple between fermions of opposite charity;

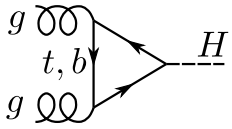
$$\mathcal{L}_{\text{Yukawa}} = -Y_l \bar{L}_L \phi l_R - Y_d \bar{Q}_L \phi d_R - Y_u \bar{Q}_L \tilde{\phi} u_R + \text{h.c.} \tag{2.6}$$

where L_L and Q_L represent the left handed lepton and quark doublets, l_R represents the right handed lepton singlets, and d_R and u_R represent the right handed down, bottom, strange and up top charm singlets respectively.

Couplings to the massive particles enable various production and decay modes for the Higgs. As each production mode contributes to the cross section of the Higgs, they collectively influence our ability to accurately predict the cross section of the Higgs. The production modes of the SM Higgs are closely linked to the production modes of the Higgs in the 2HDM, as will be seen later, in section 2.1.4.3.

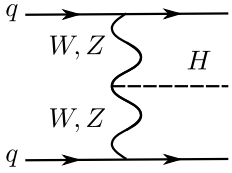
The production modes are [8];

- Gluon fusion; this is the dominant process for Higgs production at the LHC. Either a top to bottom quark loop mediates two gluons to a Higgs.



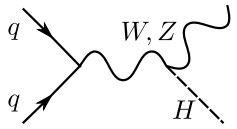
  Top quark loops are the largest contribution, as the coupling of the fermion to the Higgs scales with mass. At $\sqrt{s} = 13$ TeV the cross section comes to 42.9 pb [9].

- Vector-boson fusion; two W or Z bosons are emitted by quarks, they fuse to generate a Higgs.



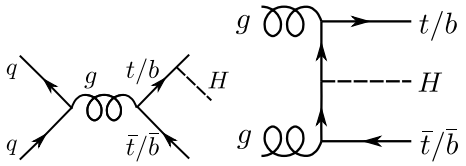
This is the next largest contribution after gluon fusion, and it grows with increasing energy. While it has no loops, the coupling of the Higgs to the Z or $W^{\pm}$ is not as strong as to the top quark. At $\sqrt{s} = 13$ TeV the cross section comes to 3.748 pb [9].

- Higgs-strahlung; the Higgs is emitted from a W or Z boson as a Bremsstrahlung like emission.



Once the energy threshold of the Higgs and the vector boson have been reached, increasing the energy will does not increase the cross section. At $\sqrt{s} = 13$ TeV the cross sections are $\sigma_{W^{\pm}H} = 1.380$ pb and $\sigma_{ZH} = 0.8696$ pb [9].

- Associated production [10], also known as Charged Higgs production; The Higgs boson is produced in association with top or bottom quarks, for charged Higgs' beyond the Standard model this could produce a charged Higgs.



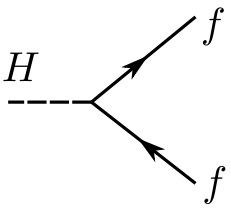
By this point the cross sections are around 2 orders of magnitude smaller than the leading contribution. At $\sqrt{s} = 13$ TeV the cross sections are $\sigma_{bbH} = 0.5116$ pb and $\sigma_{ttH} = 0.5085$ pb [9].

- Plus further smaller contributions.

Also relevant to this thesis are the decay modes. The Higgs will always decay long before it could reach a detector element, so the decay modes characterise the signals with which its existence can be inferred. Even the particles to which it decays, in the final state of these decay modes, will decay before they reach detector elements. The reconstruction process works in stages; first common endpoints, such as b-quarks are reconstructed, then from this intermediate stage, heavier particles, such as the Higgs are reconstructed. Hence, the second step of the reconstruction process requires knowledge of the decay modes of the particle of interest.

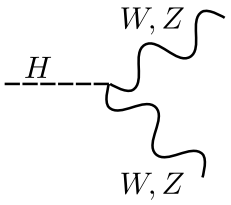
The decay modes are;

- Fermionic Higgs decay; the Higgs decay into a pair of fermions.



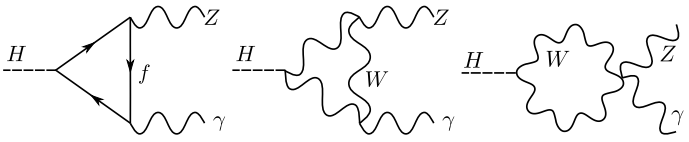
The most common of these is Higgs to a pair of b-quarks.

- Vector-boson Higgs decay; the Higgs decay into a pair of W or Z bosons.



This is less common than fermionic decay.

- Higgs decay to gluons; mediated by a top or bottom loop the Higgs decays to gluons. This is the reverse of gluon fusion.
- Higgs decay to photons; mediated by a quark loop, the Higgs decays to a pair of photons. This is like the decay to gluons, with the photons taking the place of the gluons and the loop being any quark rather than restricted to top or bottom. This decay is not common, but it has low background due to how well photons are reconstructed, so it is important for establishing the Higgs mass.
- Higgs decay to photon and Z boson, and Dalitz decays; these are yet further loop mediated decays.



In 2013 observation of the 125 GeV Higgs boson was announced[1] [11]. Francois Englert and Peter Higgs were awarded the 2013 Nobel Prize in physics for their predictions, sadly, Robert Brout died in 2011, 2 years too soon. Since then new data has only strengthened this discovery, and further refined the mass measurement [12].

2.1.3 Open Questions

Choosing the next step can be guided by the problems that might still be solved. There isn't a shortage of problems available. Working from the list given in [13];

[1] Delightfully, and for reasons best known to the presentor, the presentation accompanying this announcement was done in comic sans.

1. The Higgs mechanism is added to the SM ad hoc, that is to say, one might accuse the addition of being an attempt to "patch-up" a gap in theory [14]. It is the only fundamental scalar field in the SM, all other scalars being dynamical combinations of other fields. Having actually observed a Higgs somewhat ameliorates this worry, but none the less, it has been a continued point of discussion [15, 16].

2. The so called hierarchy problem for the Higgs mass; the radiative corrections to the mass of most particles are limited by symmetries, but those of the Higgs are not because its mass term is invariant. Quadratically divergent contributions come from top quark loops, self interactions and loops with gauge bosons. Cancellations that must occur between the tree level mass and the large loop corrections eclipse the scale of the observable Higgs mass [17]. Without these quantum corrections, the Higgs could have a mass that was larger by about 17 orders of magnitude. Fermion masses do not suffer from this, as contributions to the mass that do not contain the mass term itself are forbidden by chiral symmetry. Similarly, gauge invariance protects the vector bosons' mass. The Higgs benefits from neither of these protections. Strictly speaking, there is no inconstancy between theory and observation here, there could just be some unnervingly fine tuned numbers[2].

3. Fermion masses and mixing angles are still arbitrary, and present a similar hierarchy problem, all be it 'only' over 5 orders of magnitude [18].

4. The gauge coupling in the SM, electromagnetic, weak and strong forces, do not unify. Theories that unify these forces into a single Lie group are known as Grand Unified Theories[3].

5. Observations indicate that dark matter exists [19]. That is quite exciting, because there is no particle in the SM that matches the properties needed for dark matter, so there really ought to be something more to find. While neutrinos don't interact, and may in fact have mass, they don't have the properties required to form large scale structures observed in dark matter.

6. Observations also indicate that gravity exists. A model that could include this force would also be exciting.

7. The universe appears to be mostly matter, we have not observed much antimatter [20]. This implies that early on in the universe some antisymmetric process took place, which produced more baryons than antibaryons, known as baryogenesis. The conditions required to produce this baryogenesis are known as the

[2] Many people feel that forbidding the universe from being fine tuned is almost as important as requiring it to be internally consistent. Pretending that the universe is obliged to make sense at all is more or less a requirement for doing physics.

[3] Commonly abbreviated as GUT. An acronym that goes a long way to explaining why other acronyms are often a little convoluted.

Sakharov conditions [21]. The parameters of the SM prevent it from satisfying these conditions, so this is also a challenge for a new model.

8. Tensions between the predictions made by the SM and the results measured in experiments are few and far between, but have been seen. An example of this is the muon's anomalous magnetic moment, which has been observed by both Fermilab and Brookhaven National Laboratory [22], to a combined significance of 4.1 sigma. Such a tension could be resolved by further testing, but if it persists, then in would represent both a potential guide to BSM physics, and a challenge for new physics to explain.

2.1.4 Addition of a Second Higgs Doublet

Following on from these open questions, in the remainder of this section a possible extension to the SM will be outlined. A basic description of the theory will be given, sufficient for the purposes of this thesis. Finally, the potential for extracting answers for the open questions from the 2HDM will be discussed.

In the SM we have one Higgs doublet, as described in section 2.1.2. One of the simplest extensions possible is to add another doublet, to create a two Higgs doublet model (2HDM). These two doublets will be referred to as Φ_1 and Φ_2. The most general gauge invarient, renomalisable, scalar pottential that can be constructed is;

$$\begin{aligned} V(\Phi_1,\Phi_2) =& m_{11}^2\Phi_1^\dagger\Phi_1 + m_{22}^2\Phi_2^\dagger\Phi_2 - m_{12}^2\left(\Phi_1^\dagger\Phi_2 + \Phi_2^\dagger\Phi_1\right) \\ &+ \frac{\lambda_1}{2}\left(\Phi_1^\dagger\Phi_1\right)^2 + \frac{\lambda_2}{2}\left(\Phi_2^\dagger\Phi_2\right)^2 + \lambda_3\Phi_1^\dagger\Phi_1\Phi_2^\dagger\Phi_2 + \lambda_4\Phi_1^\dagger\Phi_2\Phi_2^\dagger\Phi_1 \\ &+ \frac{\lambda_5}{2}\left[\left(\Phi_1^\dagger\Phi_2\right)^2 + \left(\Phi_2^\dagger\Phi_1\right)^2\right] + \\ &\left(\left[\lambda_6(\Phi_1^\dagger\Phi_1) + \lambda_7(\Phi_2^\dagger\Phi_2)\right](\Phi_1^\dagger\Phi_2) + \text{h.c.}\right) \end{aligned} \tag{2.7}$$

In this, m_{11}, m_{22}^2 and $\lambda_{1\ldots4}$ are real, whereas m_{12}^2 and $\lambda_{5,6,7}$ can in general be complex. All together there are 14 input parameters available. A restriction is added, requiring that CP is conserved in the Higgs sector. This restriction forces all the named variables to be real, bringing the number of parameters down to 10, or 11 including the SM Higgs vev $v = 2m_W/g_2$ [23]. Taking $\lambda_6 = \lambda_7 = 0$ is also a common choice, as these terms are odd in the doublets and so break the discreet symmetry $\Phi_{1,2} \to -\Phi_{1,2}$, breaking this symmetry causes flavour changing neutral currents. This brings the number of parameters to 9, including the SM Higgs vev. Alternative, soft CP breaking models are available [10], but they are not considered in this thesis.

At the minimum of this potential both doublets must take their vevs;

$$\langle \Phi_a \rangle_0 = \begin{pmatrix} 0 \\ v_a/\sqrt{2} \end{pmatrix} \tag{2.8}$$

where $a = 1, 2$ for either doublet. A parameter β is defined such that $\tan(\beta) = \frac{v_2}{v_1}$ and $v = \sqrt{v_1^2 + v_2^2}$. In order for the 2HDM to correctly reproduce the $W^\pm$ and Z masses v^2 must equal $\frac{1}{\sqrt{2}G_F}$ [7]. With these definitions;

$$\langle \Phi_1 \rangle_0 = \frac{v}{\sqrt{2}} \begin{pmatrix} 0 \\ \cos\beta \end{pmatrix} \qquad \langle \Phi_2 \rangle_0 = \frac{v}{\sqrt{2}} \begin{pmatrix} 0 \\ \sin\beta \end{pmatrix} \tag{2.9}$$

In full, the two doublets each have 4 free parameters;

$$\Phi_a = \begin{pmatrix} \phi_a^+ \\ (v_a + \rho_a + i\eta_a)/\sqrt{2} \end{pmatrix}, \tag{2.10}$$

again, $a = 1, 2$.

As in the SM Higgs, 3 of these parameters create mass terms for the $W^\pm$ and Z bosons. After this is done, we have 5 remaining parameters, which will lead to 5 particles;

1. h; a light, neutral, CP even Higgs.
2. H; a heavy, neutral, CP even Higgs.
3. A; a CP odd (pseudoscalar), neutral, Higgs.
4. $H^\pm$; a pair of charged, CP even, Higgs.

The input parameters of the model will be rephrased in a more convenient manner, to include the masses of these new particles, in what is sometimes known as the physical mass basis [24]. The physical mass basis is a more useful parametrisation for phenomenology.

Using the parameterisation given in Equation 2.10, and expanding about the minimum;

$$\left. \frac{\partial V}{\partial \Phi_a^\dagger} \right|_{\langle \Phi_a \rangle_0} = 0 \tag{2.11}$$

the mass terms can be obtained [7]. Firstly, these two conditions can be used to rewrite the parameters m_{11} and m_{22} in terms of the other parameters in the model, further reducing the models input parameters to 7 (again, including the SM Higgs vev). These

constraints have the form;

$$m_{11}^2 = m_{12}^2 \frac{v_2}{v_1} - 2\lambda_1 v_1^2 - \lambda_{345} v_2^2$$
$$m_{22}^2 = m_{12}^2 \frac{v_1}{v_2} - 2\lambda_2 v_2^2 - \lambda_{345} v_1^2 \tag{2.12}$$

Making the definition $\lambda_{345} = \lambda_3 + \lambda_4 + \lambda_5$ will also assist in writing elegant mass matrices. For the neutral Higgs;

$$-\frac{1}{2}(\rho_1, \rho_2) \begin{pmatrix} m_{12}^2 \frac{v_2}{v_1} + \lambda_1 v_1^2 & -m_{12}^2 + \lambda_{345} v_1 v_2 \\ -m_{12}^2 + \lambda_{345} v_1 v_2 & m_{12}^2 \frac{v_2}{v_1} + \lambda_2 v_2^2 \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \end{pmatrix} = -\frac{1}{2} \rho^\dagger \mathcal{M} \rho. \tag{2.13}$$

Where $\mathcal{M}$ has been introduced for later convenience. For the charged Higgs;

$$-[m_{12}^2 - (\lambda_4 - \lambda_5)\frac{v_1 v_2}{2}](\phi_1^-, \phi_2^-) \begin{pmatrix} \frac{v_2}{v_1} & -1 \\ -1 & \frac{v_1}{v_2} \end{pmatrix} \begin{pmatrix} \phi_1^+ \\ \phi_2^+ \end{pmatrix}. \tag{2.14}$$

For the CP odd Higgs;

$$-[m_{12}^2 - 2\lambda_5 \frac{v_1 v_2}{2}](\eta_1, \eta_2) \begin{pmatrix} \frac{v_2}{v_1} & -1 \\ -1 & \frac{v_1}{v_2} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}. \tag{2.15}$$

The charged and CP odd mass matrices are both diagonalised by a rotation of β, and they get masses $m_A = [m_{12}^2/(v_1 v_2) - 2\lambda_5]v^2$ and $m_{H^\pm} = [m_{12}^2/(v_1 v_2) - (\lambda_4 + \lambda_5)]v^2$. The neutral mass matrix is the odd one out. The parameter α is defined as the rotation angle required to diagonalise the neutral mass matrix Equation 2.13. There is no neat expression for the angle α in terms of the parameters in Equation 2.7 and Equation 2.10. A messy one would be [7];

$$\tan 2\alpha = \frac{2m_{12}^2 - \lambda_{345} v^2 \sin 2\beta}{(\frac{1}{\tan\beta} - \tan\beta)m_{12}^2 - \lambda_1 v^2 \cos^2\beta + \lambda_2 v^2 \sin^2\beta} \tag{2.16}$$

Using the $\mathcal{M}$ defined in Equation 2.13 to keep the equation neat the masses of the two neutral Higgs can be written

$$m_{H,h}^2 = \frac{1}{2}\left[\mathcal{M}_{11} + \mathcal{M} \pm \sqrt{(\mathcal{M}_{11} - \mathcal{M}_{22})^2 + 4\mathcal{M}_{12}^2}\right] \tag{2.17}$$

Between all of these definitions, there is a new set of input parameters that determine parameter values in the potential in Equation 2.7;

$$\tan\beta,\ \sin(\beta - \alpha),\ m_{12}^2,\ m_h,\ m_H,\ m_A \text{ and } m_{H^\pm}. \tag{2.18}$$

There are seven parameters here, a number that has been arrived at through the requirements set out in this section. To recap, initially there were ten real parameters

of Equation 2.7, plus one more for the SM Higgs vev. Then two were removed by requiring $\lambda_6 = \lambda_7 = 0$. In Equation 2.12 two more parameters were removed using the minimisation conditions, bringing the total to the seven input parameters that remain.

2.1.4.1 Yukawa Types of 2HDM

The coupling of the doublets to fermions must be chosen carefully, or they will induce FCNCs. FCNCs do exist in the SM, but they are highly suppressed [25]. The simplest solution for the 2HDM is that fermions of the same quantum numbers couple to the same doublet, then the doublets cannot mediate any FCNCs. There are 2HDM variants that permit FCNCs and find ways to suppress them [26], but in this thesis, only the variants that lack FCNCs will feature.

Given that there are two doublets, and the quarks and leptons can be assigned individually, there are 4 possible configurations [7]. In all configurations the left handed quarks and leptons couple to the Φ_1 doublet, and the behaviour of the right handed components determines the type;

- **Type I**; All right handed quarks and leptons couple to Φ_2.
- **Type II**; Right handed quarks with $Q = 2/3$ couple to Φ_2 right handed $Q = -1/3$ quarks and leptons couple to Φ_1.
- **Lepton specific**[4]; All right handed quarks couple to Φ_2 leptons couple to Φ_1.
- **Flipped**[5]; Right handed quarks with $Q = 2/3$ and leptons couple to Φ_2 right handed $Q = -1/3$ quarks couple to Φ_1.

These coupling determine the Yukawa interactions in such a way as to avoid all FCNCs. At this point it is possible to relate back to the open questions in section 2.1.3. To begin with, solutions to the fermion mass hierarchy may be possible with simple extensions to 2HDMs [27]. Going further, finding a 2HDM would be a step in the right direction for establishing the existence of a supersymmetric model. The MSSM requires two Higgs doublets, and their Yukawa interactions must match those of the Type-II 2HDM [28]. It is not true, however that these models make all the same predictions for the Higgs sector. MSSM imposes more restrictions that the Type-II 2HDM, so while finding some Type-II 2HDM is a requirement for the MSSM, additional criteria would need to be met, even within the Higgs sector.

An MSSM would be of relevance to many of the open questions. Having the Higgs embedded in a much larger theory would certainly dispel any accusation of it being

[4] Also known as Type X or Type IV
[5] Also known as Type Y or Type III

ad hoc. An MSSM could stabilise the Higgs mass [29], offer force unification [30], and, even considering variations in only a few of its many parameters provide a candidate for dark matter [31]. It could even offer mechanisms for baryogenesis [32].

This seems almost too good to be true, and unfortunately it might be. Extensive searches for supersymmetries have yet to yield any findings [33, 34]. Variations and additions to the MSSM can be derived that explain the lack of findings, although that involves adding more parameters to an already considerable "minimal" model, and perhaps requiring those parameters take very particular values to avoid detection. To some this sounds like a bad idea; the parameter restrictions are considered another form of fine tuning, and the complex models have been accused of being "baroque" [35]. Either succeeding in, or exhausting the search for 2HDMs, would be a valuable contribution to these broader questions of theory.

2.1.4.2 Additional Constraints on 2HDM

So far in this section a number of constraints on the 2HDM have been given; CP conservation, and the prevention of FCNCs. There are a number of further considerations; unitarity, perturbativity, stability of the potential, EWPOs, flavour physics observables, existing Higgs measurements and 2HDM parameter choices eliminated by existing searches. In this next section these constraints will be described.

Unitarity is a sufficient condition to guarantee that all probabilities predicted by a theory will be less than 1, if the Hamiltonian is hermitian then the time evolution operator will be unitary. The requirements this places on the 2HDM can be expressed in terms of the λ_i defined in Equation 2.7. Defining a set of eigenvalues e_i for $i = 1, \dots 12$;

$$\begin{aligned}
&e_{1,2} = \lambda_3 + 2\lambda_4 \pm 3|\lambda_5|, \qquad e_{3,4} = \lambda_3 \pm \lambda_4, \quad e_{5,6} = \lambda_3 \pm |\lambda_5| \\
&e_{7,8} = 3(\lambda_1 + \lambda_2) \pm \sqrt{9(\lambda_1 - \lambda_2)^2 + 4(2\lambda_3 + \lambda_4|)^2}, \\
&e_{9,10} = \lambda_1 + \lambda_2 \pm \sqrt{(\lambda_1 - \lambda_2)^2 + 4|\lambda_5|^2}, \\
&e_{11,12} = \lambda_1 + \lambda_2 \pm \sqrt{(\lambda_1 - \lambda_2)^2 + 4|\lambda_5|^2}.
\end{aligned} \tag{2.19}$$

It is required that the e_i's be less than 16π for each $i = 1, \dots, 12$ [36, 37, 38].

Perturbativity[6] is needed to obtain finite answers for properties calculated for this model. The requirement is simply that $|\lambda_i| \leq 8\pi$ for all λ_i in Equation 2.7 [39, 40].

Stability of the potential simply requires that the potential does not go to $-\infty$ in any direction. This does not forbid the potential from having multiple minimum, just enforces that it is bounded from below. For the particular potential given in Equation 2.7,

[6] Often simply listed as another unitarity constraint [39]

necessary and sufficient conditions for this are [41, 10]

$$\lambda_{1,2} > 0,\ \lambda_3 > -\sqrt{\lambda_1\lambda_2},\ \lambda_3 + \lambda_4 - |\lambda_5| > -\sqrt{\lambda_1\lambda_2}. \tag{2.20}$$

EWPOs [42], such as the oblique parameters S and T [43, 44], require a level of degeneracy between the charged Higgs boson state and one of the heavier neutral Higgs bosons. Using the assumption $m_H^{\pm} = m_A$ or mH, the T parameter exactly vanishes in the alignment limit, creating compliance with EWPOs.

Flavour physics observables constrain the branching rates of a number of processes, the SM replicates these values very well already, so the new 2HDM must also. These include $\mathcal{B}(B \to X_s\gamma)$ and $\mathcal{B}(B_{d,s} \to \mu^+\mu^-)$, and many others given in [42].

Existing SM Higgs signals of course have to be honoured. The simplest way to achieve this is to chose $\sin(\beta - \alpha) = 1$, known as the alignment limit [45]. In this limit h has all the same coupling as the SM Higgs, so it naturally reproduces the SM Higgs signals.

Eliminated 2HDM parameter choices from existing searches are the most challenging to account for as they are being actively developed. They must be sourced in recent publications [46].

2.1.4.3 Production and Decay of 2HDM Higgs Bosons

Production mechanisms for neutral Higgs bosons are the same as for the SM Higgs, which are given in section 2.1.2. Again, gluon fusion is the dominant process.

Charged Higgs bosons have three key production processes [47, 48];

- A Drell-Yan process might produce a pair of charged Higgs; $q^+q^- \to Z^0/\gamma \to H^+H^-$ [49]. This could also be started with an electron or muon annihilation in a linear accelerator; $l^+l^- \to Z^0/\gamma \to H^+H^-$ [50].
- Top decay can generate a charged Higgs, $t \to H^+b$. The top itself would normally be produced as part of a pair, so the full chain would become $pp \to t\bar{t} \to H^+b\bar{t}$ [48].
- b associated production can produce a single charged Higgs; $qb \to q'H^+b$. This involves a gluon exchanged between the b and the remaining quark [48].
- Direct production from quarks; $c\bar{s} \to H^+$ [51], sometimes referred to as resonant charged Higgs production [48].
- At loop level, it is possible to produce a charged Higgs in association with a $W^{\pm}$; $gg \to W^{\pm}H^{\mp}$. The loop is a quark loop [52].

It is also possible to create a charged and neutral Higgs together; $qq \to H^{\pm}S$, where S is one of the neutral Higgs, h, H or A [51]. This could be mediated by W–Z fusion.

The 2HDM also allows for Higgs to be produced by the decay of other Higgses, this is known as a Higgs cascade decay [53]. This decay mode would create a distinctive signature and makes an excellent target for investigation.

Chapter 3

Mapping Potential and Existing 2HDM Parameter Spaces

This section is drawn from the work published in [1]. This work was co-authored with Souad Semlali, Rachid Benbrik and Stefano Moretti.

Decisions made about direction and content were join decisions, with Professor Benbrik and Professor Moretti providing up to date understanding of the parameter areas of interest in the 2HDM. I constructed the data pipeline, which took data from multiple existing studies, and wrote code that called the external programs `MadGraph`, `SusHi`, `2HDMC`, `HiggsBounds` and `HiggsSignals` to perform the calculations required. Dr Semlali performed cross checks at key points in these calculations. I constructed the plots from the generated data and the group collectively analysed the findings. The text of the original publication [1] was a collective effort.

3.1 Introduction

As was described in depth in chapter 2, the 2HDM is a popular extension of the SM. While the SM Higgs fits all existing data (section 2.1.2) there are good motivations for considering extensions to the SM (section 2.1.3), and an additional Higgs doublet is one of the simplest extensions available.

As described in section 2.1.4, the Higgs particle spectrum of the 2HDM is as follows: there are two CP even (h and H, with, conventionally, $m_h < m_H$), one CP odd (A) and a pair of charged ($H^{\pm}$) Higgs bosons. Amongst the many signals that these additional Higgs states could produce, of particular relevance are those involving their cascade decays, wherein a heavier Higgs state decays to a pair of lighter ones or else into a light Higgs state and a gauge boson. This is the case as the former process gives access to the shape of the Higgs potential of the enlarged Higgs sector while the latter channel

is intimately related to the underlying gauge structure, which may well be larger than the SM one.

This chapter will focus on the second kind of processes, specifically involving only the neutral Higgs states in addition to the discovered SM-like one, which in this 2HDM is identified with the h state. In short, a study of $A \to ZH$ and $H \to ZA$ decays has been undertaken[1]. The pattern of Branching Ratios (BRs) of the two decays $A \to ZH$ and $H \to ZA$ was first discussed in [55] and [56] (albeit in a Supersymmetric version of the 2HDM) and more recently implemented in [57, 58] in the 2HDM. As for production channels, the by far most relevant one is gluon-gluon fusion, i.e., $gg \to A$ or H, with an occasional competing contribution from $b\bar{b} \to A$ or H, respectively.

LHC searches for the complete channels $gg, b\bar{b} \to A \to ZH$ and $gg, b\bar{b} \to H \to ZA$ have been carried out at both ATLAS [46] and CMS [59, 60], by exploiting leptonic decays of the gauge boson, $Z \to l^+l^-$ ($l = e, \mu$), and hadronic decays of the accompanying neutral Higgs state, in particular, H or $A \to b\bar{b}$ or $\tau^+\tau^-$. Based on this approach, current experimental data exclude heavy neutral Higgses with masses up to about 600–700 GeV, depending on the BSM Higgs spectrum and the value of $\tan(\beta)$, the ratio of the Vacuum Expectation Values (VEVs) of the aforementioned two Higgs doublets. These findings are broadly in line with previous phenomenological results obtained in [61], which had forecast the LHC scope in accessing both $A \to ZH$ and $H \to ZA$ decays in a variety of final states.

Far away from the alignment limit, $\sin(\beta - \alpha) = 1$, searches have been carried out at the LHC Run 2 looking for additional Higgs bosons decaying to $A \to hZ$ or/and $H \to hh$ leading to $l^+l^-b\bar{b}$ [62, 63] or/and $\tau^+\tau^-b\bar{b}$ [64]. While in the exact alignment limit, $A \to hZ$ and $H \to hh$ will be suppressed, $A/H \to H/AZ$ is unsuppressed if kinematically open. There are additional reasons for studying $A \to ZH$ and $H \to ZA$ decays. For a start, [65] emphasised the importance of using the $pp \to A \to Zh$ process to test the wrong-sign limit of the so-called 2HDM Type-II (see below). Furthermore, [66] highlighted the fact that this very same process echoes the dynamics of the EW Phase Transition (EWPT). It is the scope of this study to revisit these two decay chains, in particular, a synergetic approach that recasts the results of experimental searches in one mode, interpreted in terms of 2HDM constraints, into the scope of the other in the same respect will be employed. This is possible because they end in the same final state. This work considers the final state $l^+l^-b\bar{b}$ and starts from the results of [46] for the $A \to ZH$ decay in order to obtain the corresponding ones for the complementary channel $H \to ZA$, in the hope that this recasting can afford one with a much stronger sensitivity that either channel alone can offer.

[1]The case of the corresponding charged Higgs boson decays of the type $H^\pm \to W^\pm H$ and $W^\pm A$ has been recently reviewed in [54].

At the time of writing, the LHC will soon enter run 3. In run 3, the integrated luminosity will increase from 36.1 fb^{-1} to 300 fb^{-1} and the centre of mass energy will be increased from $\sqrt{s} = 13$ TeV to $\sqrt{s} = 14$ TeV. This work also predicts the sensitivity of the LHC to both channels, $A \to ZH$ and $H \to ZA$, after the upgrade.

3.2 Methodology

This section describes the procedure for a scan of the parameter space of the 2HDM, which aims to establish the sensitivity of LHC data analyses to such a BSM scenario and map the findings of one channel into the other. Following this section, the results of this scan are presented and finally, the significance of these results is explored in a concluding section.

3.2.1 Existing Data

As mentioned in the previous section, the most significant source of observed data for this study is [46], for which, the raw data is available at www.hepdata.net/record/ins1665828.

For any particular cross section times branching ratio under study, in this case $A \rightarrow ZH \rightarrow l^+l^-b\bar{b}$, two probability distributions can be constructed; an observed distribution, and an expected distribution. The values of cross section times branching ratio have dimensions of barns, normally measured in units of pb. From each distribution, a value can be identified such that the probability of the cross section times branching ratio being less than that value is 95%. This is the 95% confidence limit (CL). It has a different interpretation for the observed and expected distributions;

- **Observed** 95% **CL**; There is a 95% chance that should the true cross section times branch ratio be higher than this value it would already have been discovered in the data that has been previously gathered. This limit can rule out parameter combinations that create too large a signal, declaring them inconstant with current observations.
- **Expected** 95% **CL**; There is a 95% chance that should the true cross section times branch ratio be higher than this value it would be possible to discover with the equipment currently possessed. This is a measure of which parameter combination the equipment would be sensitive to, it filters out parameter combinations that produce too small a signal and are unlikely to be detectable.

All CLs discussed here are at the 95% level, this number will should be assumed from this point on.

In the aforementioned data, expected and observed CLs are provided for a scan that covers all four yukawa types, and picks values for the parameters m_H, m_A and $\tan(\beta)$

at regular intervals. These intervals are;

$$\begin{gathered} m_h = 125\ \text{GeV} \\ 130\text{GeV} < m_A < 700\ \text{GeV},\ m_A \geq m_H + 100\ \text{GeV}, \\ m_A, m_H \text{ chosen at 10 GeV intervals.} \\ \tan(\beta) \in \begin{cases} \{1,\ 2,\ 3\}, & \text{if Lepton Specific} \\ \{1,\ 5,\ 10,\ 20\}, & \text{otherwise} \end{cases} \end{gathered} \tag{3.1}$$

These choices serve to satisfy the constraints from the EWPOs [42]. The EWPOs, such as the oblique parameters S and T [43, 44], require a level of degeneracy between the charged Higgs boson state and one of the heavier neutral Higgs bosons. Using $m_{H^\pm} = m_A$ or m_H, ensures that the T parameter exactly vanishes in the alignment limit.

To actually make use of these limits, a prediction must be calculated using the theory of the 2HDM, and this number will be compared with the CL. This requires two further parameter choices;

$$\begin{gathered} \sin(\beta - \alpha) = 1 \\ m_{12}^2 = \frac{m_A^2 \tan\beta}{(1 + \tan^2\beta)} \end{gathered} \tag{3.2}$$

To calculate the production cross sections of the heavy CP-odd Higgs, A, at each point the program `SusHi` is used [67, 68, 69, 70]. This program allows the calculation to be performed at Next-to-Next-to-Leading Order (NNLO) in QCD.

`SusHi` is written in `Fortran 77`, which is capable of performing expensive calculations quickly. In comparison to all other steps in the computation, however, the cross section calculation requires significantly more time to complete. If the branching ratio calculations for a scan took hours, the cross section calculation would often require days. This is not entirely unexpected, but the scale of the difference motivated some further investigation of which aspects of the code might be optimised. Code profiling revealed that while the majority of the program's run time was spent performing calculations, significant time was spent reading and writing to disk, and a smaller fraction of time was spent printing to screen. To remedy this, some small patches were written for the program;

- To modify the program such that it could be called as a subprocess and would take input and return output via the pipe. Disk reading and writing becomes operations in the RAM.
- To modify the program to suppress excessive print statements.

With these modifications, scans in `SusHi` could be performed 30% faster than previously[2]. With this, the production cross sections for the Higgs bosons in the 2HDM can be calculated.

In order to calculate branching ratios of each Higgs state, in particular those of $A \to ZH$ and $H \to b\bar{b}$, the program `2HDMC` [24] is called. This is written and called in `C++`, some modifications to this program that enable it to compile in conjunction with interfaces to `HiggsSignals` and `HiggsBounds`, which will be needed later, were kindly shared by Mohamed Korab[3][4]. `HiggsBounds` and `HiggsSignals` are used to apply the aforementioned exclusion limits at 95% CL from Higgs searches at LEP, Tevatron and LHC, while `2HDMC` checks the theory constraints as described in section 2.1.4.2. These are; perturbativity, stability of the potential, and unitarity.

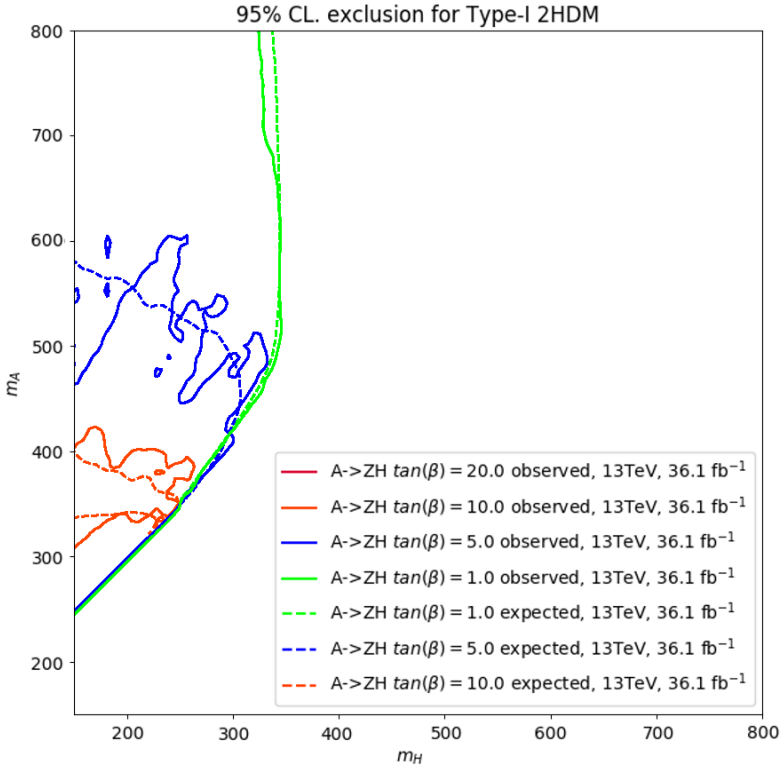


FIGURE 3.1: A reproduction of Figure 6 from [46] for type-I of the 2HDM.

By generating cross sections and branching ratios at $\sqrt{s} = 13$ TeV and luminosity $= 36$ fb^{-1} the results presented in Figure 6 of [46] are replicated. An example of this for yukawa type-I is presented in Figure 3.1.

[2]The modifications are available at hub.docker.com/repository/docker/henrydayhall/higgspheno
[3]Cadi Ayyad University
[4]These modifications are also accessible from the docker at hub.docker.com/repository/docker/henrydayhall/higgspheno

3.2.1.1 Theoretical Constraints and Existing Data

Upon checking the theoretical constraints using `2HDMC` it was discovered that the parameter choices in Equation 3.1 and Equation 3.2 did not pass the theory constrains for any interval. While the masses of the particles where fixed by the observational data available and the capacity of the detectors, the parameters in Equation 3.2 would only influence the predictions. Changes to Equation 3.2 would require changes to the predicted cross section times branching ratio, but not to the CLs given in [46].

Changing $\sin(\beta - \alpha) = 1$ would move the model away from the alignment limit, and change the predicted behaviour of the SM Higgs, as such this is undesirable. However, there is no such difficulty in changing the value of m_{12}^2.

So the easiest resolution to this issue was to scan values of m_{12}^2 in search of those that permitted the remaining parameters. The challenge here was simply the size of the parameter space available. Is not unreasonable to consider very large m_{12}^2, the range used here was; $0 < m_{12}^2 < 2 \times 10^5$ GeV. Values that satisfy the theory checks may differ for every parameter combination in the intervals described in Equation 3.1. It would be computationally wasteful to sample the whole parameter space.

Fortunately, the values of m_{12}^2 which are permissible are correlated with the values of the masses chosen by the interval. This can be seen in Figure 3.2. Once a handful of valid points have been found, a quadratic surface may be fitted to these points and further values of m_{12}^2 chosen near this surface. Using this, theoretically permitted values of m_{12}^2 could be found for the majority of the scan, as is seen the in the results in Figure 3.3 to Figure 3.6.

Where no value of m_{12}^2 could be found that satisfied all three constraints of unitarity, stability and perturbativity, the value that satisfied as many as possible was chosen.

3.2.1.2 Flavour Physics Constraints

Flavour physics observables, namely, $B \to X_s\gamma$, $B_{s,d} \to \mu^+\mu^-$ and $\Delta m_{s,d}$, also impose constraints on this scan [42]. In order to account for these constraints, values from Figure 9 of [42] where digitised. These eliminated some areas of the scan.

3.2.2 Recasting the Scan

Two extensions to this scan are considered. Firstly, an alternative decay chain can be used; $pp \to H \to ZA \to l^+l^-b\bar{b}$. This alternative is possible because the only kinematic difference between these processes are different widths for the Higgs bosons, only minimally affecting the efficiency of an experimental selection. This alternative

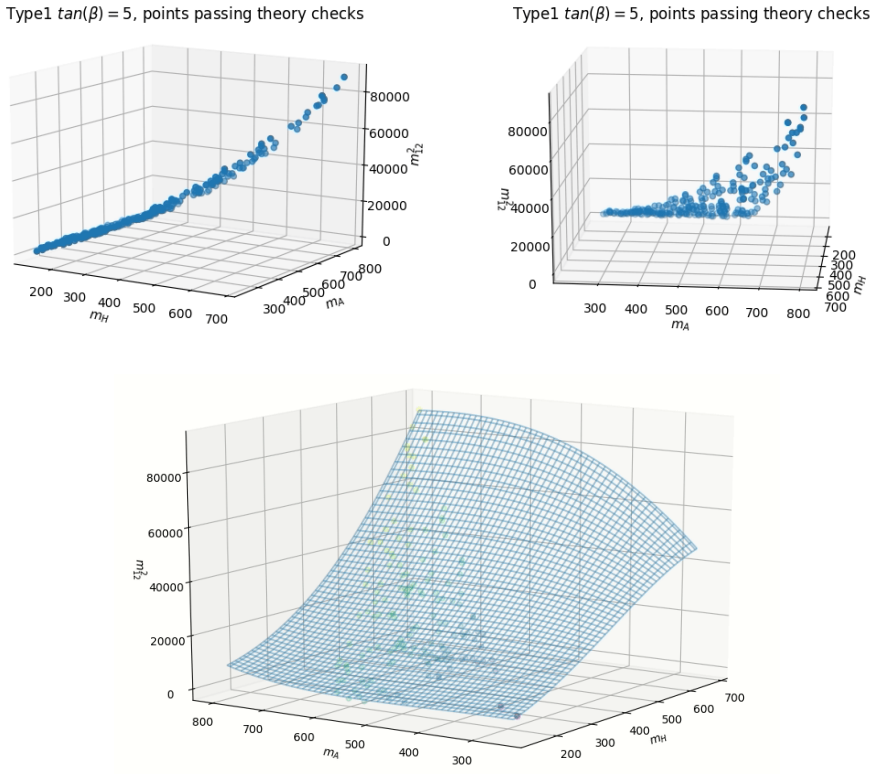


FIGURE 3.2: Randomly sampled values of m_{12}^2 that pass all theory checks for type-I, $\tan\beta = 5$, as calculated by 2HDMC are plotted against the two masses being scanned, m_A and m_H. The first row shows the same data set from two angles, the view in the top left emphasising that the values that pass the theory checks fall into a narrow but continuous band, the view in the top right showing that there are some mass combinations for which no valid m_{12}^2 can be found. In the lower plot, a quadratic surface has been fitted through the points found. Where valid points exist, they will be found on this surface.

process expands the region of parameter space that can be tested to the case when $m_H \geq m_A + m_Z$. For this extension, only additional predictions for the new production cross sections and branching ratios must be calculated. The calculations can be done with exactly the same programs as for the replication of the original scan.

Despite the symmetries that exist between these processes, neither the constraints affecting the two processes nor their sensitivity reaches should expected to be the same. On the one hand, the role played by the heavy CP-even and CP-odd Higgs states of the 2HDM in both theoretical and experimental limits is different, owing to their different quantum numbers (and hence couplings). On the other hand, their production and decay rates at the LHC are different despite leading to the same final states, including residual differences due to width effects entering their normalisation (but, as mentioned, not their kinematics), since, e.g., the A state does not decay to W^+W^- and ZZ pairs while the H state does and, conversely, the A state decays to Zh while the H state does not. However, in the alignment limit used here these decay channels are closed.

The second extension is to extrapolate these results to full Run 3 data samples[5]. To match the conditions of Run 3, the integrated luminosity factor must be increased from 36 fb^{-1} to 300 fb^{-1} and the centre of mass energy must be increased from $\sqrt{s} = 13$ TeV to $\sqrt{s} = 14$ TeV.

Increasing the luminosity can be simplistically modelled as increasing the rate at which both signal and background occur. This is accounted for by calculating the so called 'upgrade factor' for both signals and backgrounds, while retaining the acceptance and selection efficiencies of the analysis at the lower $\sqrt{s}$ value.

The change in energy will naturally affect the production cross section of signals and backgrounds differently. The signal cross sections are recalculated with `SusHi`. The increase for the background is found with `MadGraph` 5, version 2.6.4, [71]. For completeness, the background is considered to be any reducible or irreducible SM process that creates a pair of b-jets plus a pair of electrons or muons, as in [46].

[5]It is not, of course possible to predict the observed limits in Run 3. Otherwise there would be no need to actually have a Run 3 at all.

3.3 Results

In this study, lightest CP-even Higgs boson of the 2HDM is identified as the observed Higgs state at the LHC, with $m_h = 125$ GeV, and $\sin(\beta - \alpha) = 1$.

The scan includes the following parameter range:

$$\begin{gathered} m_h = 125\,\text{GeV},\ \sin(\beta-\alpha) = 1, 0 < m_{12}^2 < 2\times 10^5\,\text{GeV}, \\ 130\text{GeV} < m_X < 700\,\text{GeV},\ m_X \geq m_Y + 100\,\text{GeV}, \\ m_X, m_Y \text{ chosen at 10 GeV intervals.} \\ \tan(\beta) \in \begin{cases} \{1,\ 2,\ 3\}, & \text{if Lepton Specific} \\ \{1,\ 5,\ 10,\ 20\}, & \text{otherwise} \end{cases} \end{gathered} \tag{3.3}$$

The set of values chosen for $\tan(\beta)$, and the masses, align with the choices in [46].

- For the process mediated by $A \to ZH$, $m_X = m_A$, $m_Y = m_H$ and $m_{H^\pm} = m_A$ is chosen. (Note that this choice is consistent with that of [46].)
- For the process mediated by $H \to ZA$, $m_X = m_H$, $m_Y = m_A$ and $m_{H^\pm} = m_H$ is chosen. (Note this choice is specular to that in [46].)

After performing a scan over the parameter space delimited by Equation 3.3, the predictions of the model are compared to the observed and expected limits given in [46]. If the prediction exceeds the observed limit, then the parameter combination is excluded. When the prediction exceeds the expected limit, the signal is anticipated to be visible above background given the energies and luminosities available, hence, the experiment is sensitive to these parameters.

As described in section 3.2.1.1, the choice of $m_{12}^2 = m_A^2 \tan(\beta)/(1+\tan(\beta))^2$ reconstructs the exclusion limits at 95% CL given in [46]. However, this choice does not actually satisfy theoretical constraints anywhere in the four types of 2HDM. Therefore, this analysis required a different choice. The method described in section 3.2.1.1 was used to select values of m_{12}^2 for each interval of the scan that aimed to simultaneously satisfy as many theoretical constraints as possible.

Figure 3.3 to Figure 3.6 illustrate the outcome the scan for each Yukawa type, $\tan(\beta)$, and mass combination, (m_H, m_A). Each figure provides results for one choice of Yukawa couplings and each frame in each figure provides results at one value of $\tan(\beta)$. In the top left of each plot, where $m_A > m_H + 100$ GeV, the decay $A \to ZH$ is considered while in the bottom right of each plot, where $m_H > m_A + 100$ GeV, the decay $H \to ZA$ is considered. The corridor along the diagonal between these regions is coloured grey to indicate that neither decay is accessible.

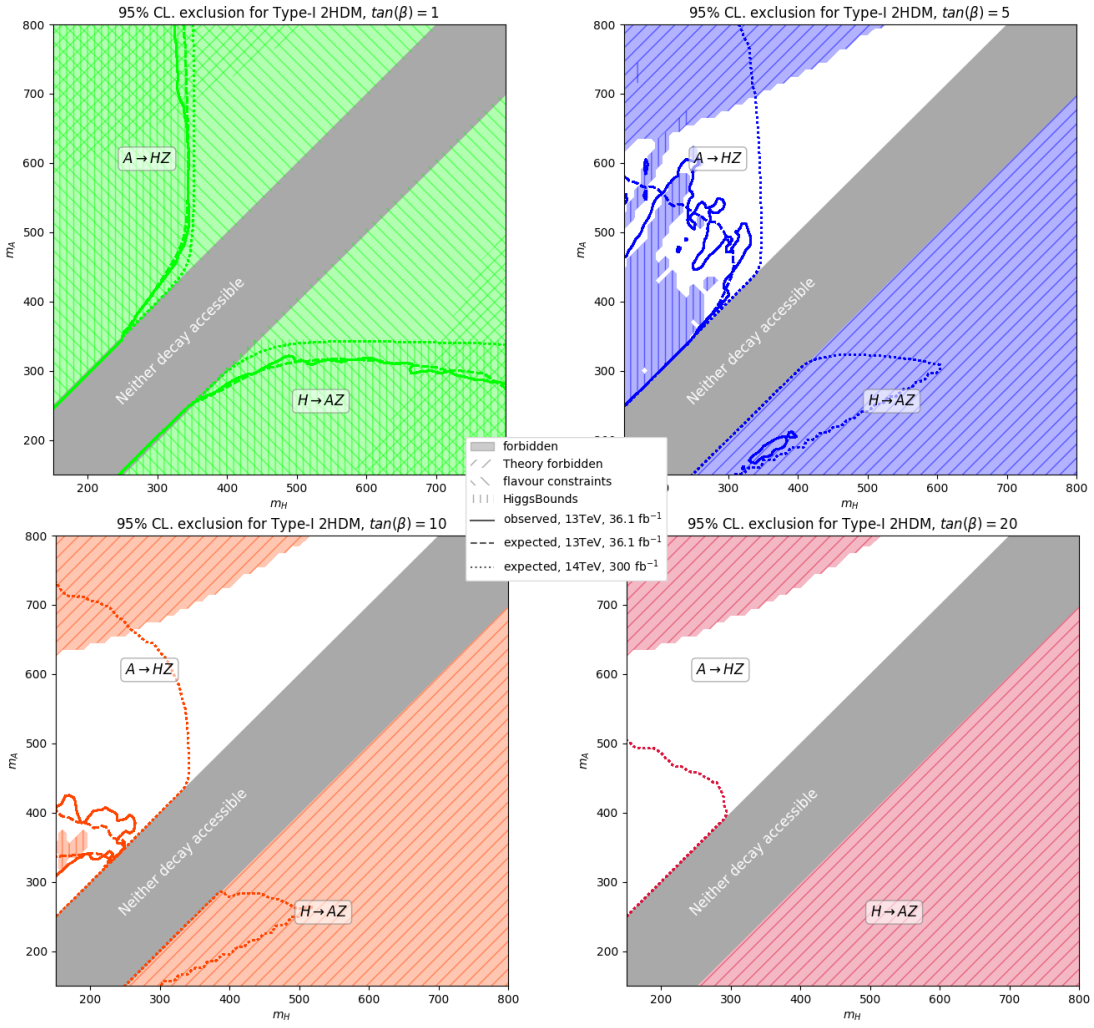


FIGURE 3.3: Exclusion limits at 95% CL in Type-I. The lines denoting expected and observed exclusion limits do not appear at all on some plots when the prediction never exceeds the expected or observed limit. The asymmetry in both constraints and sensitivity is expected, see the discussion in section 3.2.2 and the branching ratios in Figure 3.7.

If a combination of parameters is forbidden by theory, `HiggsBounds` or flavour constraints then the corresponding area is filled with solid colour, conversely, white areas pass all these checks and so are of interest. The hatching over the solid colour is used to indicate which of the checks causes the corresponding parameter combination to fail. There are three boundary lines drawn over the plots: these are the observed and expected 95% CLs for the ATLAS detector in its present state, 13 TeV and 36.1 fb^{-1}, plus the expected 95% CL for an upgraded LHC and ATLAS detector at 14 TeV and 300 fb^{-1}[6]. The model predictions exceed the 95% CL inside the curve.

In Figure 3.3 the parameter space with Type-I Yukawa couplings is shown. The upper left plot shows that $\tan(\beta) = 1$ is always forbidden by flavour constraints. The upper

[6]The case of $\sqrt{s} = 13$ TeV and $L \approx 140$ fb^{-1} is neglected here, as it only improves marginally the present situation yet it would be make the plots far too crowded.

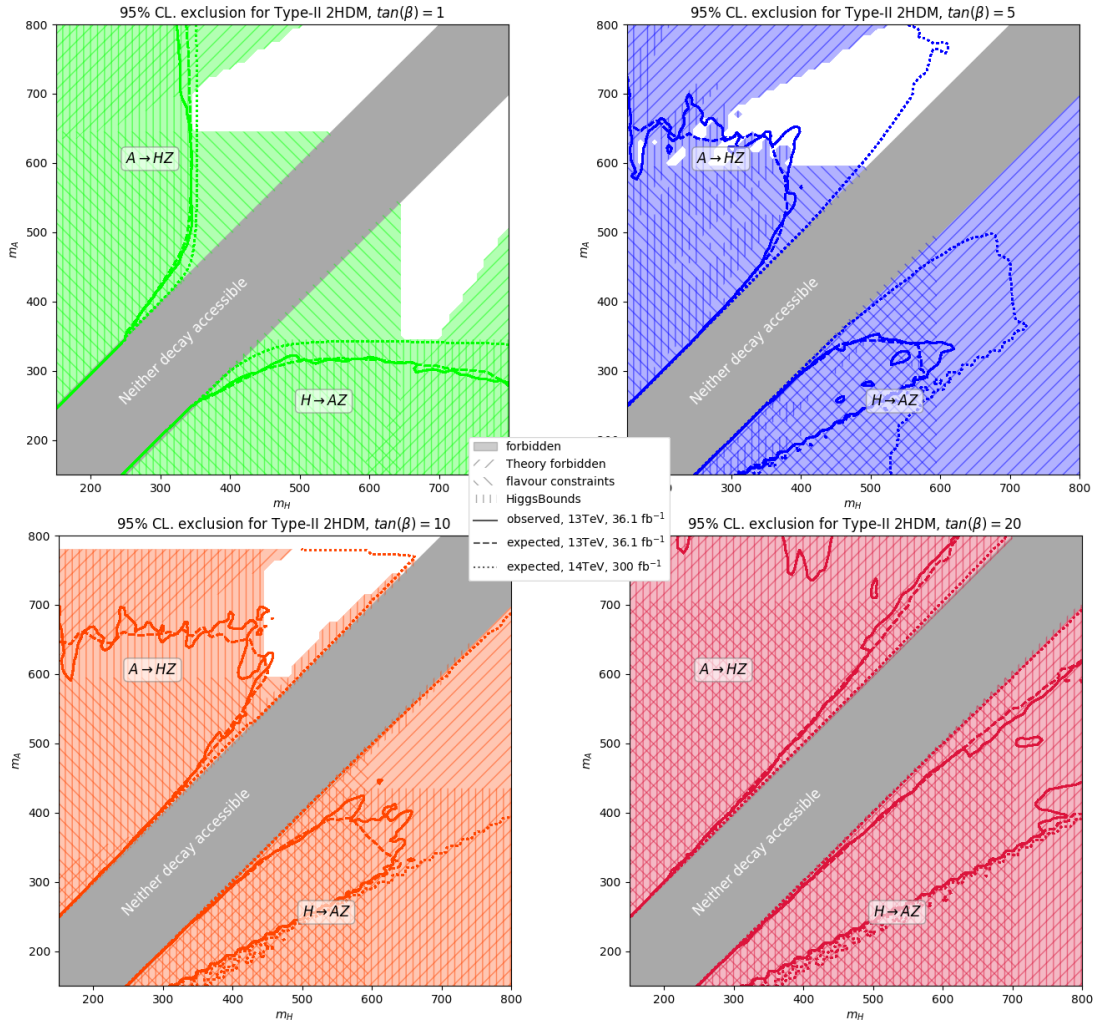


FIGURE 3.4: Like in Fig. 3.3 but for Type-II. The asymmetry in both constraints and sensitivity is expected, see the discussion in section 3.2.2 and the branching ratios in Figure 3.7.

right plot shows that there are many mass combinations that do not prevent the decay $A \to ZH$ for $\tan(\beta) = 5$, but theory constraints forbid all mass combinations relevant to $H \to ZA$. At $\tan(\beta) = 5$ for 13 TeV (and 36.1 fb^{-1}) the area of sensitivity (inside the expected curve) that is not excluded by observation (inside the observed curve) is very limited. It is also seen that the $H \to AZ$ signal has reduced sensitivity when $\tan(\beta)$ is 5 or more. This is due to $H \to AA$ competing with $H \to AZ$, as shown in Figure 3.7. The branching ratio $H \to AA$ becomes significant because of the enhancement of the trilinear coupling λ_{HAA} at large $\tan(\beta)$. At 14 TeV and 300 fb^{-1}, however, many mass combinations are expected to be testable that have not yet been excluded. The lower left plot shows the behaviour at $\tan(\beta) = 10$ to be similar to $\tan(\beta) = 5$, i.e., everything is forbidden for $H \to ZA$ by theory while for $A \to ZH$ most combinations for which there is sensitivity have been excluded at 13 TeV but 14 TeV offers even more possible parameter space than seen at $\tan(\beta) = 5$. Finally, in the lower right frame of Figure 3.3, the parameter space for $\tan(\beta) = 20$ is shown. The state of $H \to ZA$ is unchanged, but

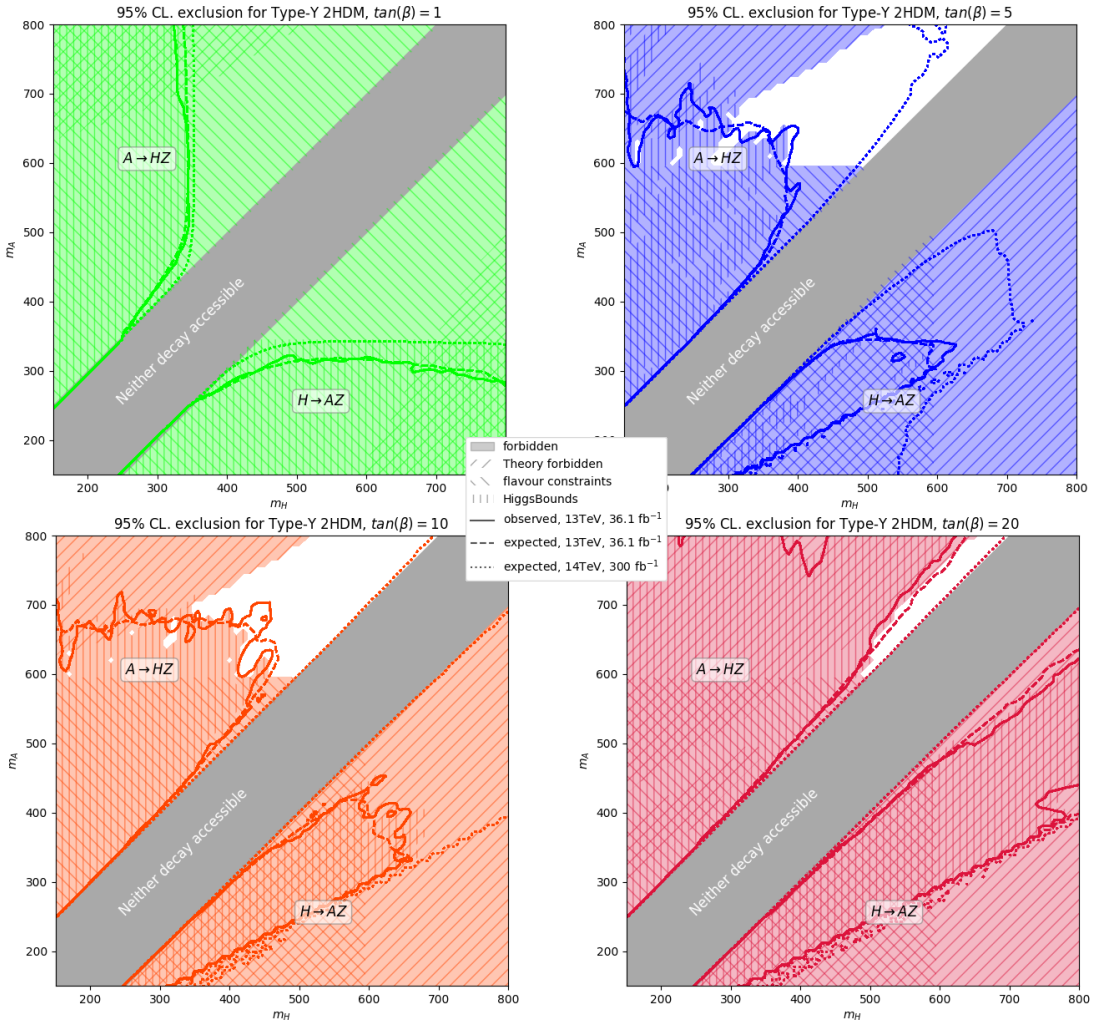


FIGURE 3.5: Like in Fig. 3.3 but for Type-Y (Flipped). The asymmetry in both constraints and sensitivity is expected, see the discussion in section 3.2.2 and the branching ratios in Figure 3.7.

now $A \to ZH$ has no expected or observed exclusion at 13 TeV, i.e., these parameters are harder to probe. With the upgrade to 14 TeV and 300 fb^{-1} there is some sensitivity to $A \to ZH$ at $\tan(\beta) = 20$.

As might be expected, the behaviour of Type-II, shown in Figure 3.4 and Type-Y, shown in Figure 3.5, is remarkably similar. The upper left plot shows that $\tan(\beta) = 1$ is forbidden by flavour constraints in all areas where there is sensitivity. At 13 TeV and 36.1 fb^{-1} the upper right plot shows that the same can be said for $\tan(\beta) = 5$, however, after Run 3, at 14 TeV and 300 fb^{-1}, there are many permitted mass combinations for $A \to ZH$. However, $H \to ZA$ is excluded by theory. The behaviour at $\tan(\beta) = 10$, shown in the lower left plot, is much the same as for $\tan(\beta) = 5$, except more of the exclusion at 13 TeV and 36.1 fb^{-1} is from observations provided by `HiggsBounds`. Finally, in the lower right plot, $\tan(\beta) = 20$ is shown to be excluded for almost all mass choices, by multiple constraints.

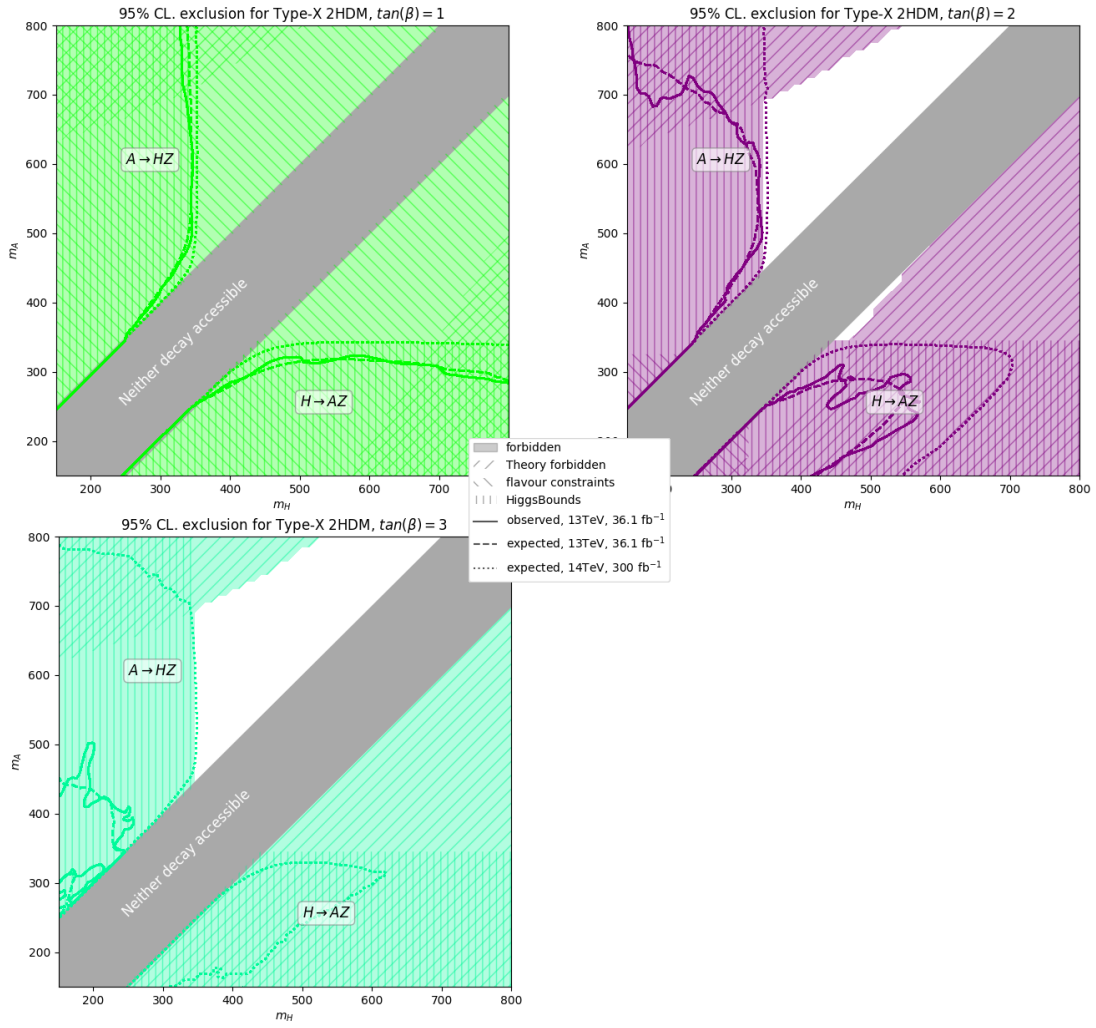


FIGURE 3.6: Like in Fig. 3.3 but for Type-X (Lepton specific). The asymmetry in both constraints and sensitivity is expected, see the discussion in section 3.2.2 and the branching ratios in Figure 3.7.

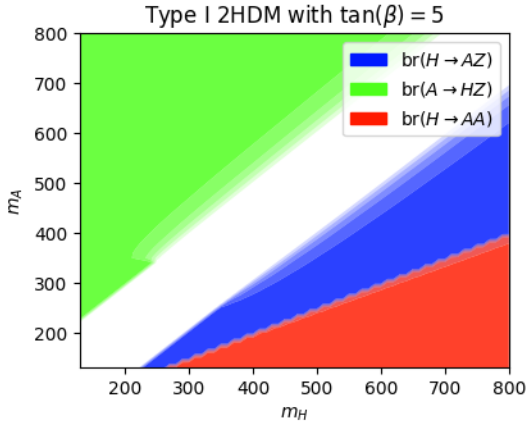


FIGURE 3.7: The branching ratio $H \to AZ$ is suppressed by the branching ratio $H \to AA$. This effect occurs for all types, but does not occur at small $\tan(\beta)$.

$\tan(\beta)$	1	5	10	20
Type-I	Flavour constraints	Some masses	Many masses	Low sensitivity
Type-II	Flavour constraints	Some masses after upgrade	Some masses after upgrade	Theory constraints
Flipped	Flavour constraints	Some masses after upgrade	Some masses after upgrade	Theory constraints
$\tan(\beta)$	1	2	3	
Lepton specific	Flavour constraints	Excluded by `HiggsBounds`	Excluded by `HiggsBounds`	

TABLE 3.1: Table summarising the findings in Figs. 3.3 to 3.6. An overview of the possibility of each Yukawa type and value of $\tan(\beta)$ is given. Entries in red indicate that the combination has little or no mass combinations that are not forbidden while those in blue represent available parameter space accessible presently at Run 2 or after the upgrade of Run 3.

In Figure 3.6 the behaviour of the Type-X 2HDM is shown, at a set of $\tan(\beta)$ values that differs from those previously considered. The change is made because the parameter space in Type-X shrinks more rapidly with increasing $\tan(\beta)$ compared to the other Yukawa types. For these choices `HiggsBounds` excludes all areas inside the expected limits. This remains true even after the end of Run 3.

Finally, Table 3.1 summarises the findings, highlighting that sensitivity only really exists for $5 < \tan(\beta) < 10$ and limitedly to the 2HDM Type-I, both at Run 2 and 3, and -II and -Y (or Flipped), but only at Run 3. The case of Type-X (or Lepton specific) is never accessible.

3.4 Conclusions

In summary, this work has revisited an experimental analysis of the ATLAS Collaboration of the production and decay process $gg, b\bar{b} \to A \to ZH \to l^+l^-b\bar{b}$ performed at Run 2 with 36.1 fb^{-1} of luminosity, which had been interpreted in terms of exclusion limits over the parameter space of the four types of the 2HDM, wherein the lightest Higgs state is identified with the SM-like Higgs boson discovered during Run 1 at the LHC with mass 125 GeV. Upon validating the ATLAS interpretation in this framework, though, it was discovered that their (fixed) choice of m_{12}, a mass parameter in the 2HDM Lagrangian that softly breaks an underlying Z_2 symmetry of the 2HDM to avoid FCNCs, yields parameter space configurations which are ruled out by theoretical requirements of model consistency. Hence, values of m_{12}^2 have been reselected, subject to the aforementioned theoretical constraints. Thus, redrawing the actual sensitivity of such an experimental search to all four Yukawa types of the 2HDM, according to both $\tan\beta$ and the masses, m_H and m_A.

In addition, this work has also forecast the potential sensitivity of this channel to the 2HDM parameter space at the end of Run 3, assuming increased energy to 14 TeV and luminosity to 300 fb^{-1}. This revealed some extended coverage of the 2HDM Type-I, -II and -Y (but not -X), especially for intermediate $\tan(\beta)$ values (say, between 5 and 10), with m_A up to 800 GeV and m_H up to 700 GeV. This is somewhat beyond what is presently covered, i.e., up to 150 GeV or so in mass of either Higgs state, so as to justify further searches for this signature at the next stage of the LHC.

Finally, sensitivity of this analysis has been recast onto that of the channel $gg, b\bar{b} \to H \to ZA \to l^+l^-b\bar{b}$. However, this finds that the complementary parameter space accessible this way (i.e., $m_H \geq m_A + m_Z$) is actually entirely excluded already by existing theoretical and/or experimental constraints, so as to conclude that it is not warranted to pursue further this channel at the LHC, at least, not with a view to interpret it in the context of the standard four Yukawa types of the 2HDM[7].

[7] Finally while it is true that analyses similar to [46] performed by the CMS Collaboration exist [59, 60]. These where not used here for two reasons. On the one hand, they did not convey all the information necessary to make extrapolations to higher energies. On the other hand, they did not afford one with significantly different sensitivity to the 2HDM at present energies than what achieved by the ATLAS analysis [46] that was adopted here as benchmark.

Chapter 4

Jet Physics

Jets are a pervasive and crucial signal in high energy colliders. Resolving and identifying them plays a major role in the search for massive particles such as Higgses. This chapter will review a series of concepts that illustrate the requirements for jets, then review key aspects of jets themselves. Concepts will be discussed in a time ordered sequence;

1. This sequence begins with a description of the hard event, collision of two protons to form particles of interest [1].
2. The output of the hard event will then undergo showering and hadronisation. Furthermore, there are a number of other processes are generating noise, such as pile-up and initial state radiation (ISR). Collectively, these things present the most significant hurdles to accessing the hard event, which jets are designed to overcome.
3. The detection of the hadronised objects in the detector will be discussed, along with some technical limitations on reconstructing these signals as particles.
4. Finally the process of gathering the reconstructed particles into jets, and classifying these jets can be described.

Figure 4.1 offers a symbolic depiction of item 1 and item 2. Primarily, item 1 to item 3 should give background knowledge required for investigation of new jet clustering methods. Jets simplify the initial reconstruction of the detector output, which often contains hundreds of particles, and provide a handle to further process the data.

Beyond that, this review should also illustrate that jets are better seen as concept, rather than a definition. For a particular hard interaction, say Higgs produced by gluon fusion, there is one clean definition; jets do not have this privilege. Many algorithms for

[1] Interesting things happened before this point of course; *"If you want to make a pie from scratch, you must first create the universe." –Carl Sagan*. Alas, I'm no cosmologist.

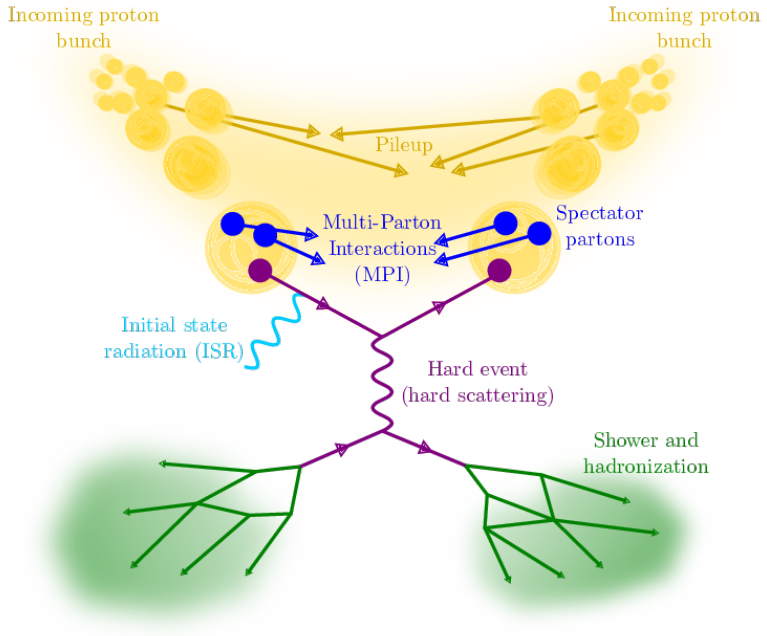


FIGURE 4.1: Symbolic depiction of the physics processes occurring in an event. The hard event is where new physics might be found, and the end points of the shower are the detectable remnants of this. Three types of noise are depicted; MPI, pileup, and ISR.

jets are available, and they each lead to a subtly different end product. Indulging in a bit of philosophy, one might say that while gluon fusion is a natural kind [72], jets are not. Natural kinds being things that benefit from absolute definitions, such as "gold", a counterexample of which might be a "chair" [2].

Throughout this review, each section will also include a description of the computational tools relevant to that step. These are of great practical importance to this thesis. Computational tools are almost all highly modular; each tool is built by combining several more specific tools, which were written to deal with a single step of the process being carried out. For practicality, this review limits its scope to the highest level tools, the ones the user directly interacts with.

4.1 Hard Event

The word 'hard' in physics is synonymous to 'high energy density'. At the time of writing, the LHC collisions have center-of-mass energy 13 TeV [73]. While 'hard' is a relative description, particle collisions at this energy would be consistently described as 'hard'.

[2] "Gold" is the element with 79 protons. A "chair" has a back, some legs and can be sat on; all of which can also be said of a horse.

The word 'event', in this context, refers to the happenings that take place when a particular pair of particles interact. The event includes the things those particles themselves become, plus other things that get mixed up with them, forms of noise, or background, such as pileup.

Taken together, the words 'hard event' mean only the high energy, initial, parts of the particle interaction, a phrase that came into use sometime in the 1950's [3]. The hard event is a very small subsection of the whole event. A hard event is typically a short enough series of interactions that it can be described in a single feynman diagram. It might begin with two quarks exchanging a W boson to form a Higgs, and end with the Higgs decaying to a pair of b-quarks.

Due to asymptotic freedom of the strong force, at small distances and high energies (or equivalently, high energy densities) particles which would usually be confined to colour neutral systems may behave to first order as if they were free, these are referred to as partons [74]. The behaviour of these partons at high energy densities is predicted by QCD, and can be calculated perturbatively. This holds up to distances of approximately 10^{-17} m [75].

Probability of obtaining a particular final state of the hard event is found by calculating the inclusive cross section. For high energy collisions, short range behaviour can be calculated using the asymptotic freedom in QCD, the long range behaviour can be measured experimentally and summarised as Parton Distribution Functions (PDFs). Both the long and the short range behaviour contribute to the inclusive cross section. The factorisation theorem gives the means to combine both factors [76]. The objective of the whole particle collider system is to test the short range predictions by comparison with observations. This process is referred to as unfolding.

A second reason for interest in the hard event over other potential interactions is that the high energy densities make the production of massive particles, such as Higgses, possible. This thesis aims to investigate the possibility of more than one type of Higgs, and if they exist, these additional Higgses will only be produced at high energies. For a full description of the particular extended Higgs sector of interest, see chapter 2.

4.2 Simulating the Hard Process

`MadGraph` [71] is the most popular Monte Carlo program for simulating high energy interactions. Monte Carlo (MC) programs [4] are optimal for modelling systems with

[3] Exactly when the phrase 'hard event' became popular in physics is somewhat obfuscated by the poet Horace, who liked to use it for describing battles in about 8BC.

[4] Monte Carlo computer programs started out somewhat ingloriously. They were first used for developing nuclear weapons, and are named after the creator's uncle's favourite casino [77].

large parameter space. It may be difficult, or even impossible to simulate every possible outcome of a system. Further, the curse of dimensionality makes a sampling grid inefficient for systems with many parameters. Randomly sampling the large number of parameters is often sufficient to build a representative picture of the system, and is less sensitive to the number of parameters needed.

Alternatives to `MadGraph` do exist, for example `CompHEP` [78] or `ALPGEN` [79] are both also MC programs for modelling the hard event. This review will focus on `MadGraph`.

Using `MadGraph` without any extension, it is possible to generate hard events for any SM process. The input particles and the output particles must be described. For example, `generate p p > h`; this will sample from all lowest order processes where a pair of protons generates a SM Higgs. Optionally, subsequent decays may be specified, for example it is possible to require that the SM Higgs decays to b-quarks, `generate p p > h, (h > b b~)`. To simulate the 2HDM processes described in section 2.1.4 requires importing additional models.

A further refinement of `MadGraph` is quite important to this work; the ability to impose kinematic cuts on all parts of the process. This is very flexible indeed, there can be cuts on the direction, or momentum of the particles and cuts on the relative angle between particles. It is also possible to put cuts on global properties of the event. This is primarily useful for focusing the simulation on only the events with kinematics that would be detectable, thus reducing the space that must be sampled.

4.3 Showering and Hadronisation

As the products of the hard event fly apart, they leave the region of asymptotic freedom. Quarks will radiate gluons, mostly into a narrow cone in their direction of travel [75]. Gluons also create more gluons, quickly generating many particles. This process is called showering. Showering explains why jets undergo transverse broadening with increasing hard parton energy [80], as higher energies lead to more Final State Radiation (FSR).

As two coloured objects are separated, a gluonic flux tube stretches between them [75]. As such, separating, colour charged, objects continuously interact. This is important, because the eventual goal is to individually identify the products of the hard interaction, but their decay products will not be possible to cleanly separate. Instead, the detectable particles include decay products from the interactions between coloured objects.

Eventually energetically favourable states will be found. The gluonic flux tubes have such a high energy density that it becomes possible to generate more quark-antiquark

pairs, and the tube fragments into colour neutral hadrons. Most of theses decay products will be pions and photons, which have sufficient stability to reach the detector. This process is called hadronisation.

Background, or noise, is also produced alongside this process. Eight kinds of noise are theoretically distinct, however there isn't much consistency in the literature on the terms used to refer to them. Here, the convention used is;

1. **Out-of-time pileup**; timing resolution in the detector is not always adequate to correctly associate the tracks with the proton bunches. This leads to tracks from previous and future bunches contaminating the event. Often the vertices for these tracks will be displaced from the primary vertex, although the effects of timing can still make isolating this noise challenging [81]. This is sometimes simply referred to as pileup.

2. **In-time pileup**; each bunch in the beam contains $\mathcal{O}(10^{11})$ protons [82], so it is not surprising if more than one of them interacts. These also normally have displaced vertices. This is also sometimes just called pileup.

3. **Multi-Parton Interactions**; within the protons that interact, only one out of three quarks will directly interact, the remaining quarks are known as spectator quarks. These spectator quarks produce another type of noise referred to as MPI [83]. These are likely very soft, and so may be removed by kinematic cuts.

4. **Initial State Radiation** (ISR); prior to actually interacting, the partons that annihilate in the hard interaction may radiate. This radiation is not always soft, ISR can take a considerable fraction of the collision energy before the hard interaction occurs [82].

5. **Final state radiation** (FSR); radiation from the products of the hard interaction. This is not clearly distinct from the process of showering, however, when radiation is emitted at a wide angle it may not be included in the jet and so lower the reconstructed mass. In this sense FSR is also noise.

6. **Cavern background**; particles in the cavern may create small background contributions simply by merit of their own decay. These are normally small enough to be omitted [84].

7. **Beam halo events**; protons may just scrape against collimators on their way into the detector. This creates a beam halo. This signal is distinctive and easy to remove [84].

8. **Beam gas events**; protons from the beam may collide with gas in the detector. Interactions from these collisions create another form of noise. Again, this signal is distinctive and easy to remove [84].

Together, all these kinds of noise may be referred to as the underlying event, or pileup. They will act to obscure the signal event. The greatest contributions come from item 1 to item 5, so these will be subject to most investigation.

4.4 Simulating the Shower

There are many programs which can simulate the shower, known as event generators. The most famous three are `Pythia8` [85], `Herwig` [86] and `Sherpa` [87]. Calling these event generators makes sense as they are also capable of simulating the hard process, but the norm is to simulate the hard process in a dedicated to tool, then use an event generator to deal with the showering and hadronisation.

Showering is simulated as a step-wise Markov chain; starting from the contents of the hard event, an addition to the current state is selected based on probability distributions. These probability distributions are perturbative at high energies.

Details of this algorithm vary between event generators. For example, `Pythia8` and `Sherpa` are dipole-type showers; colour-anticolour dipoles are put between pairs of partons, one of these partons emits a new parton and the other, the spectator, is used to conserve momentum locally. `Herwig` takes a different approach, with branching gluons off heavy quarks using angular ordering. Each approach has different benefits and drawbacks, in terms of what aspects it is best able to model.

The underling event can also be simulated by event generators. The output of the event generator aims to completely replicate everything that can be detected, including the noise. This is very valuable for designing object reconstruction algorithms, such as jet finding algorithms.

Another aspect of event generators that can be of great use is the ability to rerun only the showing, hadronisation and underlying event while keeping the hard event fixed. As the hard event is kept the same, any object reconstruction on these repetitions should aim to produce as little variation as possible. This is a useful handle for investigating the behaviour of object reconstruction algorithms.

4.5 Detectors

After hadronisation the particles are stable enough to travel to the detector. A great diversity of equipment exists for detecting particles, different devices have different sensitivities and limitations. Some of the requirements on the detectors for a HEP collider are;

- Sensitivity to both charged and neutral particles which interact electromagnetically. While it would be very exciting to also measure particles such as neutrinos which only interact via the weak force, this is typically outside of the scope of a standard detector. Currently it is only done at dedicated experiments, but there are plans to add neutrino detection to the LHC; the ForwArd Search ExpeRiment (FASER) [88] at the LHC is an example of this [5].

- Sensitivity to the momentum and direction of particles. This can often be achieved by many layers of sensors, each with good spatial resolution, such that a path is reconstructed.

- Sensitivity to the curvature of a charged particles path. With this, when a detector is bathed in a magnetic field the direction of curvature will indicate the sign of a particle's charge.

- Short time resolution, to distinguish between events. The greater the uncertainty on the time resolution, the more out-of-time pileup is expected, see item 1, section 4.3.

- Short deadtime, to minimise the number of particles that are missed when two particles hit the same sensor in quick secession.

- Radiation harness; the beam radiates lots of high energy radiation. Sensors will eventually be damaged by this radiation, and need replacing. See, for example [89].

In order to meet these requirements many varieties of sensors will be needed. Different types of sensor are stacked together in shells, so that they can each provide good coverage of the majority of the angular area of the detector. Each shell, with a different type of sensor, is known as a subsystem. The beam itself must enter and leave the detector volume, this happens in what is known as the 'forward' area, and at the forward area there will be some gaps in the sensors coverage.

Composition and geometry of the detectors varies between experiments. Even for one single particle beam, often several different detector designs operate. For the LHC there are seven detectors; ALICE, ATLAS, CMS, LHCb, LHCf, TOTEM and MoEDAL. Of these, ATLAS, CMS and LHCb are significantly larger. Physics goals of the detectors vary, for example TOTEM [90] is designed to help measure the complete proton-proton cross section by detecting particles emitted in the forward area. TOTEM requires specialist equipment to achieve this called Roman Pots.

The physics goals of ATLAS, CMS and LHCb are broad and relatively similar, however they all have different configurations. This is a great advantage, as it allows results for

[5] And also, perhaps an example of the impact of GUTs, as noted in footnote 3, chapter 2.

each detector to be cross checked with the others. Energy of the beam at the LHC is the highest in the world; the maximum beam energy of the LHC being 6.5 TeV, with the next most energetic being TEVATRON with 0.98 TeV [91]. As such, there is no other place all results could be verified.

In this thesis we take a particular interest in the Compact Muon Solenoid (CMS) detector on the LHC. The capacity, and physics goals, of CMS are broad, so all work and conclusions remain very generalisable. Professor Claire Shepherd-Themistocleous, who supervised the project, is CMSUK deputy PI, and so the choice was made to focus on CMS in order to best utilise this in depth knowledge.

4.5.1 CMS Hardware

CMS has particular coordinate conventions. These are depicted in Figure 4.2. The z axis goes along the beam-line, in a westward direction, anticlockwise when the LHC ring is seen from above. The x axis points horizontally in towards the centre of the ring, the y axis points vertically up to the sky. Two angles are also used: $0 < \phi < 2\pi$, which goes around the roll of the barrel, starting at the x axis, and rotating in the x-y plane[6]; $0 < \theta < \pi$ which measures the angle from the z axis [93].

Two other common coordinates in use are rapidity, often denoted y (not to be confused with the linear y coordinate[7]), and pseudorapidity, often denoted η. Their definitions are respectively;

$$y \equiv \frac{1}{2} \ln \frac{E + p_z}{E - p_z} \tag{4.1}$$

and

$$\eta \equiv -\ln \left(\tan \left(\frac{\theta}{2} \right) \right). \tag{4.2}$$

Both of these coordinates provide a means of measuring how much of an object's momentum is in the z direction. As such they are often used as substitutes for a p_z coordinate. Pseudorapidity is the massless approximation of rapidity.

The detector itself is shaped as a thick cylindrical shell. The flat, circular disks on each end are referred to as endcaps, while the curved walls are referred to as the barrel.

When the shower is complete and the products have left the interaction region, they will encounter a series of detector systems depicted in Figure 4.3. The first sensor they encounter is a pixel detector. The distance that they cross to reach the pixel detector is

[6]This use of ϕ as the azimuth angle is a popular convention in physics; it has been suggested that mathematicians should try it too [92].

[7]This rarely causes issue in practice because it is unusual to need the linear x and y coordinates in particle physics, due to the symmetry of rotation in ϕ.

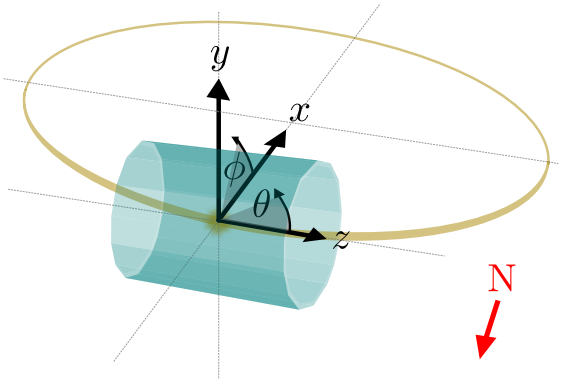


FIGURE 4.2: Coordinate system used to specify locations with respect to the CMS detector.

about 5.3 cm [94]. Although the deadtime for a silicon pixel is short, $\lesssim 50$ ns [93], each pixel can only register one binary hit per collision. As the flux on this pixel detector is very high, the pixels must be small, each pixel is 150×150 μm^2. Outside the silicon pixel detector, the flux is lower, and so a silicon strip detector with larger areas can be used. Together, the pixels and the strip make up the inner tracker.

Silicon pixels or strips have low stopping power, so they do not significantly reduce the momentum of the particles passing through them. They can only detect charged particles, such as leptons and protons. Neutral particles can only be detect by stopping them, causing a messy shower, and given the tracker is designed for precision it is not desirable to capture photons [8].

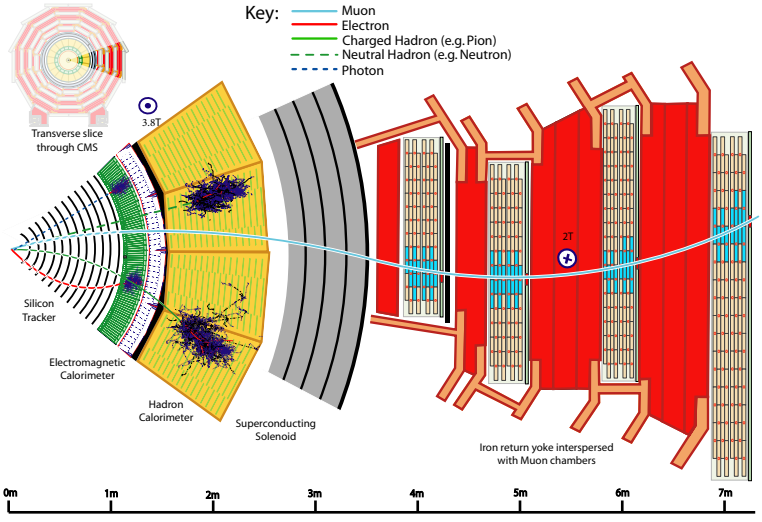


FIGURE 4.3: Depiction of the various subsystems of CMS [95].

[8] This might seem a little surprising, since a standard phone camera is also made of silicon pixels, and if the pictures it takes are a mess, that has more to do with the photographer than the silicon. However, a phone camera is detecting photons with about a billionth of the energy, so perhaps it's reasonable that those photons are somewhat neater.

At a radius of 1.29 m the electromagnetic calorimeter (ECAL) begins [96]. This is made of lead tungstate ($PbWO_4$), which has a large electromagnetic cross section, while having sufficiently low density, 8.28 g cm^{-3}, to present a small hadronic cross section. Most things that interact electromagnetically will shower here, in a Molière cone. One of the desirable properties of lead tungstate is that the Molière radius is only 2.2 cm. This gives the detector a fine granularity. Once a particle has showered on the lead tungstate, the resulting photons are detected and used to measure the energy that the particle had.

Particles that do not have a significant electromagnetic cross section may pass straight through the 23 cm of the ECAL. After that there is a hadronic calorimeter (HCAL), which is designed to capture and measure as much of the energy as possible from the remaining particles. The HCAL is composed of alternating layers of brass absorber and plastic scintillator [97]. The brass absorber is not magnetic and has a short interaction length [93].

Beyond the hadronic calorimeter is the solenoid magnet. This superconducting magnet creates a 4 T magnetic field [93], causing the path of charged particles in the detector to curve. A strong magnetic field is needed because many charged particles have very high momentum, so their path in the detector might appear straight in a moderate magnetic field. The fidelity of the detector puts a limit of how shallow a curve can be detected. A strong magnetic field creates deeper curves, improving the accuracy of charge reconstruction.

Outside the magnet's solenoid coil is another scintillator, the outer hadron calorimeter, which uses the coil of the magnet itself as an absorber [98]. This improves the energy containment a little further.

Finally, at $\approx$ 6.5 m, the muon system begins. A combination of three complimentary detector technologies track the muons which escape the solenoid [99]. These are; drift tube chambers (in the barrel), cathode strip chambers (in the endcaps) and resistive plate chambers. Differing conditions on the barrel and endcaps necessitate different sensor choices. This system aims to reconstruct the energy and charge of muons from their curvature in the magnetic field.

All together, the varying response of different varieties of particle to each subsystem enables them to be clearly identified. A summary of this is presented in Figure 4.4. The direction of the particle, and its energy, are also often possible to reconstruct. In the case of charged particles, the sign of the particle is identifiable from the direction of its curvature in the magnetic field.

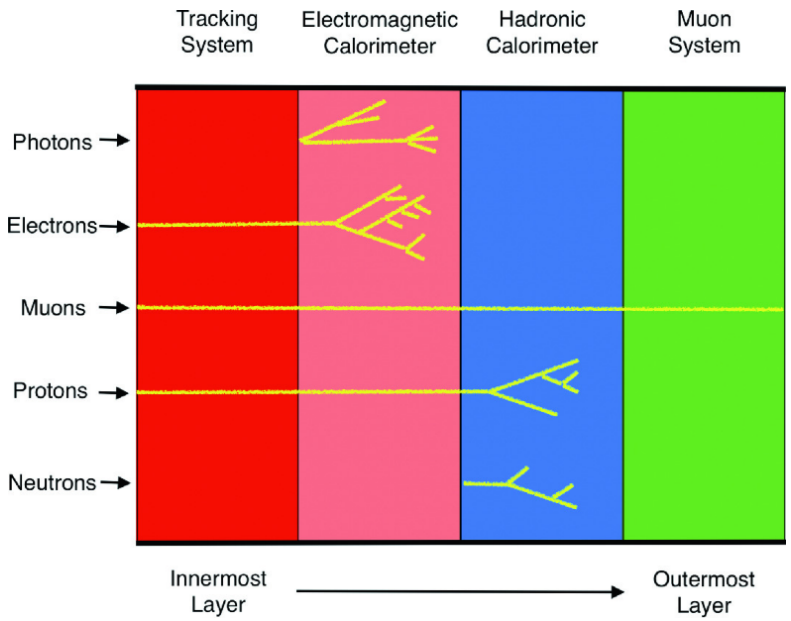


FIGURE 4.4: An illustration of the subsystems in which various types of particle may be detected [100].

4.5.2 CMS Trigger

The current luminosity of the LHC is 2.1×10^{34} cm^{-2} s^{-1} [91], that is, every second 2.1×10^{34} protons pass through a window of a centimetre squared. Not all of these will result in collisions, but many will. They are delivered in bunches, each bunch containing 11×10^{10} particles, the bunches pass through every 24.95 ns [91]. This is roughly equivalent to 1 GHz of information, a phenomenal rate of data, more than could be stored, or even transmitted away from the detector. Only the most useful parts can be kept, and those useful parts need to be selected in real time by the detector itself. This is the role of the trigger.

For CMS, the trigger is composed of two parts: the Level 1 (L1) trigger, which acts first from the hardware; the High Level Trigger, which is implemented in software and uses more complex decision making algorithms [101].

The decisions made by these trigger systems define the data that is preserved. This is the only data that object reconstruction algorithms can access, as such, the specifics of the triggers are of interest.

The L1 trigger selects based on detecting the presence of simple objects. Many smaller sections can be identified, each with separate tasks, which report up to other sections. On the lowest levels, there are trigger primitives, then many segments have their own track finders, finally, gathered information is combined in the global trigger system.

Together the L1 trigger reduces the 1 GHz of data to about 100 kHz.

If an event is accepted by the L1 trigger then it will be passed to the high level trigger. The high level trigger is composed of filter-builder units, where events are pieced together and reconstructed, before being filtered by various cuts. The reconstruction here is very similar to the offline reconstruction.

For the most part, only promising events, with clear tracks, are retained by this system.

4.5.3 Simulating the Detector

Naturally the detector does not perfectly reconstruct particles that enter it. Energy is smeared out, particles may enter regions with no sensors, sensors have deadtime and error rates. Simulating these inaccuracies is key to understand their impact on object reconstruction, and planing around this. The most prevalent simulation program is called `Delphes` [102]. Other simulations exist, `PGS` [103], short for Pretty Good Simulation, is a much older simulation written in `FORTRAN77` without any use of the library `GEANT` [104][9]. Some experiments have their own internal simulations, CMS has a simulation package known as `OSCAR`, which stands for Object oriented Simulation for CMS Analysis and Reconstruction. However access to `OSCAR` [93] is limited to CMS collaboration members[10].

`Delphes` is attractive because it is open source and well integrated with `MadGraph`. It can take files in the `.hepmc` format as input, for example, the `.hepmc` files produced by `Pythia8`. `Delphes` is not designed to be as accurate as the full detector simulations produced by the experiments themselves, however it is flexible and carries configuration files to mimic all the detectors at the LHC. Newer versions of `Delphes` also include some particle reconstruction algorithms that mimic the 'particle flow' algorithm used in CMS [102]. Further, `Delphes` is also capable of handling pileup subtraction, provided data representing pileup is supplied [102].

The complexity of the detector is such that its simulation can require more time than the simulation of both the hard event and the shower. Hence, there is good incentive to use a simple approximation for the detector simulation.

Particles with higher momentum are more likely to be reconstructed, as are particles that enter the silicon tracker. The silicon tracker has coverage of pseudorapidities -2.5 to 2.5, where pseudorapidity is as defined in Equation 4.2. Given this, it would be sensible to use particles with $p_T > 0.5$ and $|\eta| < 2.5$ as a very rough approximation of what can be reconstructed in the detector, given the selections made in the trigger.

[9] The website for `PGS` includes various enticing claims including "For many analyses you will find (we hope!) that the answer from `PGS` agrees within a factor of two of the answer you might obtain with a full-fleged [sic] detector simulation."

[10] With approximately 4000 CMS members, and 3400 seats in The Dolby Theatre where the Oscars are hosted, the two varieties of oscar are approximately equally exclusive.

4.6 Jets

After the detector has reconstructed what it can, each event may have $\mathcal{O}(100)$ particles. Those particles have come from a chain of events described in the flow chart depicted in Figure 4.1. The hard scattering contains the information of interest, it is likely to involve $\mathcal{O}(10)$ partons. The reconstructed particles are the only information we have experimental access too. This is quite a shift from $\mathcal{O}(100)$ reconstructed particles. This increase in multiplicity comes from both the backgrounds and from the showering and hadronisation. Jet clustering is designed to bridge the gap between these two multiplicities; firstly by gathering together particles that carry information about the same parton in the hard event, and secondly by separating particles with information about the hard event from particles that are mostly influenced by the background[11]. As such, each signal jet attempts to represent part of the hard event, sometimes referred to as the jet's parent object [105].

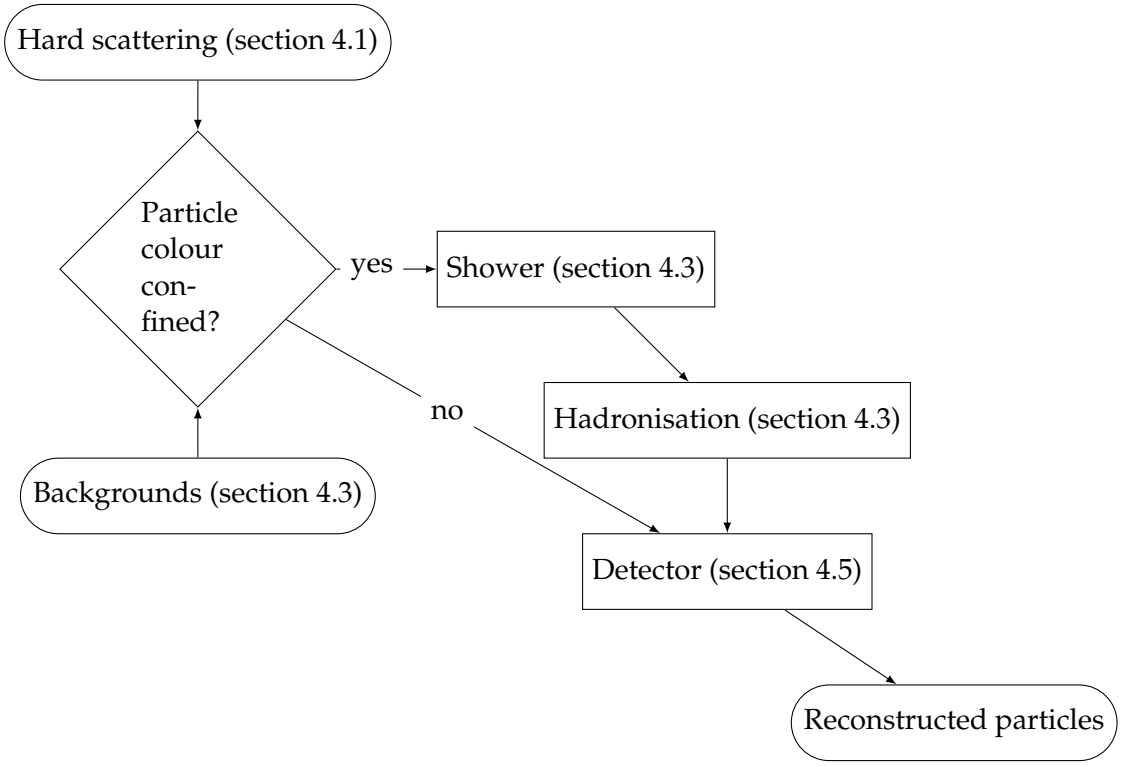


FIGURE 4.5: Overview of the sequence that relates the hard scattering to the reconstructed particles. Each iteration of this sequence constitutes one event.

A good jet algorithm should create a representation the parent object while fulfilling the criteria set out in the Snowmass accords [106];

1. It is simple to implement in experimental analysis.

[11] It has been said that *"this is an art that is similar to reading tea leaves."* –*quantumdiaries.org*

2. It is simple to implement in theoretical calculation.
3. It is defined at any order of perturbation theory.
4. It yields a finite cross sections at any order of perturbation theory.
5. It yields a cross section that is insensitive to hadronisation.

A jet is a purely conceptual construct, it is composed of a group of physical objects gathered by an algorithm, rather than being an innate physical phenomenon. Despite this, it is possible make inferences about particle models from jets and predict jet behaviour from models. Often the analysis done on jets is to view a spectrum of their masses, which can be calculated as the invariant mass of all the constituents of the jet. Resonances in the spectrum of jet masses give evidence of the existence of particles with a corresponding mass. Measurements of the number of jets in a mass range can be compared to predicted results, and provide strong constraints on models [107].

4.6.1 Infra-Red Safety

Insensitivity to soft or collinear emissions is an important attribute for jets to possess; this is referred to as IR safety. Formally, a soft emission is the emission of a massless particle with very low energy. Tending to the limit of 0 energy, there is a divergence in the probability calculation, which means that is it not possible to perform a perturbative calculation. This is not a problem, as a massless 0 energy particle is not observable. In a similar manner, there is a singularity associated with a particle generating two particles with the same momentum direction, a collinear splitting. Again, this decay is not detectable, as two particle in exactly the same place with the same combined momentum will be measured as identical to the original particle [108].

There are three key reasons for wanting an algorithm that will not allow the measurable quantities of the jets to change in response to IR radiation [105].

1. If the algorithm to form a jet was, in principle, sensitive to soft or collinear splittings it would not be possible to make predictions about jet quantities from QCD. Without these predictions many interesting comparisons to theory are spoilt. This is means IR safety is important to item 4 of the Snowmass accords.
2. The Monte Carlo simulations are also based on probabilities calculated from QCD, and they can only be tuned and compared to experimental data using measurable quantities. It cannot be determined if these simulations are accurate in the IR limit. As the IR limit is approached, the accuracy of the simulation will suffer [109]. Hence, jets designed using MC data cannot depend on the IR behaviour of the simulation.

3. The exact energy at which a soft particle becomes invisible, or a collinear splitting becomes indistinguishable depends on the detector being used. This makes predictions of behaviour of an IR unsafe algorithm unique to the detector, further complicating matters.

The comparison between safe and unsafe behaviour is illustrated in Figure 4.6 and Figure 4.7.

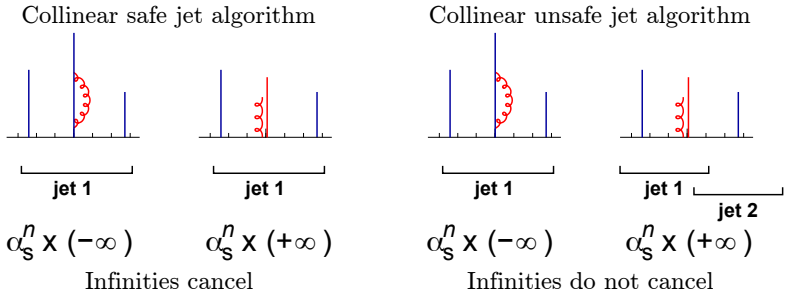


FIGURE 4.6: This is a comparison of an algorithm with collinear safety to one without. On the left the allocation of non-soft particles to jets is not influenced by the presence of a collinear splitting, this is collinear safe. On the right the allocation of non-soft particles to jets changes after the collinear splitting, this is not collinear safe. [105]

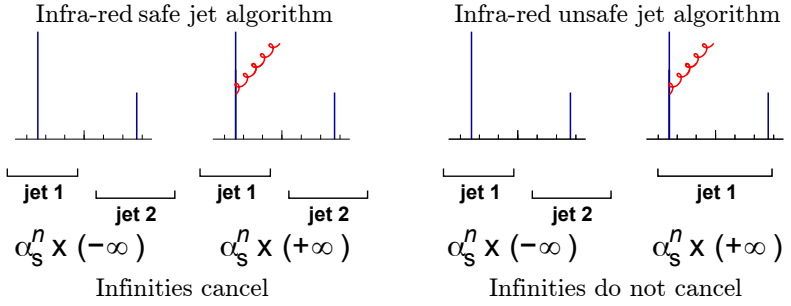


FIGURE 4.7: This is a comparison of an algorithm with infra-red safety to one without. On the left the allocation of non-soft particles to jets is not influenced by the presence of a soft emission, this is infra-red safe. On the right the allocation of non-soft particles to jets changes after the soft emission, this is not infra-red safe. [105]

As will be seen in section 4.6.3, there are jet definitions that are not IR safe, however most modern jet formation algorithms are IR safe, and this is strongly preferred.

4.6.2 Shape Variables

It was alluded to in the previous section, section 4.6.1, that there are predictions that can be made for the distributions of jets from QCD. Event shape variables are an example of this. Comparisons between various MC event generation algorithms show the shape variables measured on jets to be relatively insensitive to the details of the MC simulation used [110]. Thus, these variables are good comparisons, even including the uncertainties of simulation.

Here, 5 common shape variables are described;

1. **Jet mass**; a jet's momentum is the combined 4-momentum of all reconstructed particles assigned to the jet. The invariant jet mass spectrum is simply the invariant mass of the momentum of one or more of the jets. It can be calculated for more than one jet in each event, or just of the highest p_{T} jet of each event.

2. **Thrust**; is a description of how much of the jet momentum goes along the dominant axis. It is calculated as

$$T = \min_{\hat{n}_t} \frac{2\sum_i p_i \cdot \hat{n}_t}{\sum_i |p_i|}, \tag{4.3}$$

   where i sums over all jets, or sometimes only a subset of the jets, in the event. The factor of 2 being customary, but sometimes omitted.

3. **Thrust major and minor**; these measure thrust in other directions. Thrust major is defined as

$$T_M = \min_{\hat{n}_M \ni \hat{n}_t \cdot \hat{n}_M = 0} \frac{2\sum_i p_i \cdot \hat{n}_M}{\sum_i |p_i|}, \tag{4.4}$$

   so it is the same as thrust, only its axis, $\hat{n}_M$, is required to be perpendicular to the thrust axis, $\hat{n}_t$. Thrust minor has its axis, $\hat{n}_m$ perpendicular to both $\hat{n}_t$ and $\hat{n}_M$. Its axis is $\hat{n}_m = \hat{n}_t \times \hat{n}_M$, thus no minimisation is needed. It is calculated as;

$$T_m = \frac{2\sum_i p_i \cdot \hat{n}_m}{\sum_i |p_i|}. \tag{4.5}$$

4. **Oblateness**; this is a property also used in earth science to describe the shape of the earth[12]. It is calculated from the thrust major and the thrust minor as [112]

$$O_b = T_M - T_m. \tag{4.6}$$

   Conceptually, this measures how squished the event is.

5. **Sphericity**; heuristically this can be seen as a measure of how far the event deviates from a spherical configuration. It is calculated by first constructing the momentum tensor;

$$S^{\alpha\beta} = \frac{\sum_i p_i^\alpha p_i^\beta}{\sum_i |\vec{p}_i|^2} \tag{4.7}$$

   where α, β are x, y or z, thus $S^{\alpha\beta}$ is a 3 by 3 tensor, and the sum over i sums over all the momentum vectors of the jets (or some subset). By calculating the eigenvalues of the momentum tensor, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, the event sphericity can be written as

[12] The earth is currently an oblate sphere, however, a number of other possibilities have been studied. Figure 1 of [111], the shape the earth might take if it had a lot more water, is a striking example.

$S = \frac{3}{2}(\lambda_2 + \lambda_3)$ [113]. This is equivalent to calculating

$$S = \min_{\hat{n}_s} \frac{3\sum_i (p_i - p_i \cdot \hat{n}_s)^2}{2\sum_i (p_i)^2}. \tag{4.8}$$

The vector $\hat{n}_s$ is referred to as the sphericity axis [114].

6. **Spherocity**; this has a definition that at first glance appears similar to sphericity [114];

$$S' = \min_{\hat{n}_{s'}} \left(\frac{4\sum_i (p_i - p_i \cdot \hat{n}_{s'})}{\pi \sum_i (p_i)} \right)^2. \tag{4.9}$$

However, on closer inspection, it become clear that taking the square out of the sum means that jet momentum vectors going in opposite directions will cancel. This is now measuring how spherical and how well balanced the event is.

In some ways thrust, sphericity and spherocity describe a similar property, but the actual values obtained differ. Some cartoons illustrating this conceptually are given in Figure 4.8, and plots of a particular jet configuration are shown in Figure 4.9. Transverse variants of these shape variables exist; those are not the same quantities.

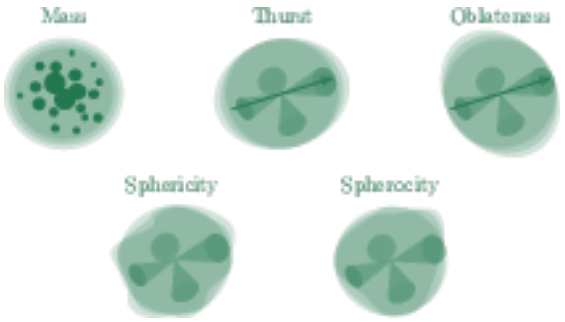


FIGURE 4.8: Conceptual illustration of various shape variables.

These quantities are sensitive to IR behaviour, and so having IR safe jets is required to make predictions about their distributions.

4.6.3 Existing Definitions

Now that the intention of, and the requirements for, jets have been established, the chapter is completed by presenting some common jet algorithms. This collection is guided by the collection of common algorithms described in [105].

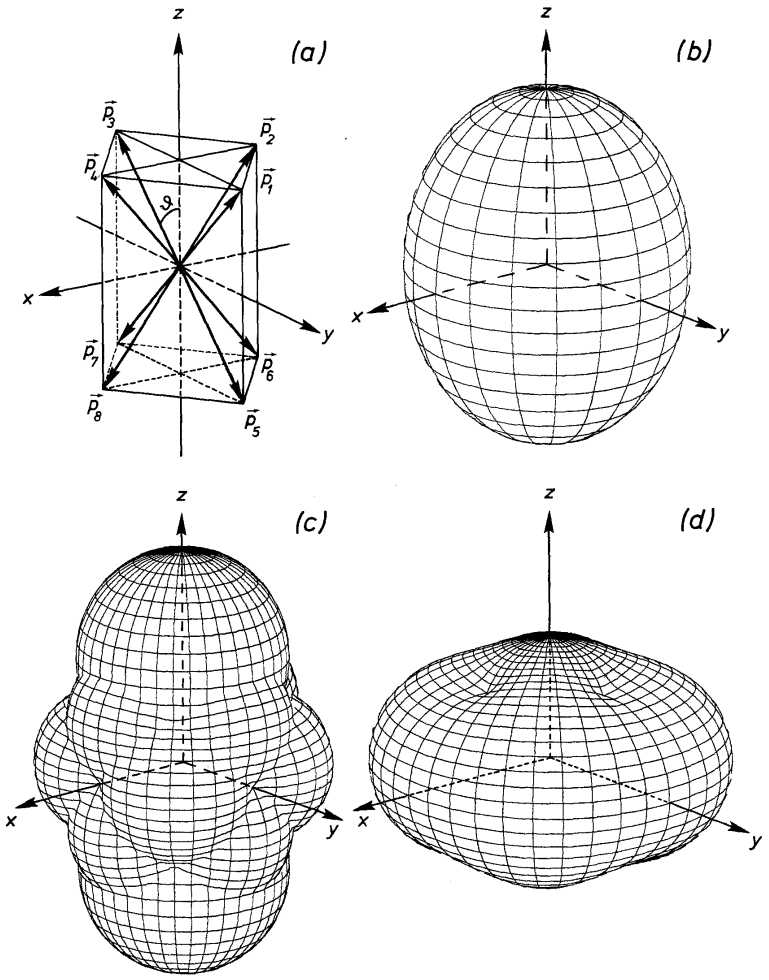


FIGURE 4.9: A plot of the magnitude of various shape axis with changing angles, published in [114]. The momentum vectors of 8 jets are given in (a), then the magnitude of various shape variable axis are calculated as the direction of the axis is changed. Thrust axis (b), sphericity axis (c) and spherocity axis (d).

4.6.3.1 First Cone Algorithm

The first jet algorithm [115] was used in a low background, electron positron collision. This algorithm attempts to creates exactly 2 jets, with a radius specified by the parameter δ. The jets are optimised to contain as much energy as possible. If it is possible to chose these jets such that they contain at least $1 - \epsilon$ of the events energy, then the jets are declared successful.

This algorithm is mostly only suitable for events with no boost factor, because it assumes that almost all the energy of the event is captured. This is not the case in hadron colliders.

4.6.3.2 Iterative Cone Algorithms

In many events more than two jets are expected. An advancement from the first cone algorithm is the concept of a iterative cone algorithm [116]. To form a jet using an iterative cone algorithm;

1. A seed particle is chosen. The momentum vector of the seed is the initial jet direction.
2. All particles within a radius δ of the jet direction are selected. Distances are usually measured using the metric $\sqrt{\delta y^2 + \delta\phi^2}$, where δy is the difference in rapidity and $\delta\phi$ is the difference in angle.
3. The combined momentum of the selected particles is the new jet direction.
4. If the jet direction has changed, return to item 2. Otherwise, this jet is now a stable cone, all currently selected particles are the final jet constituents.

This algorithm requires two more details to be complete; a procedure to select the seed particle and a procedure to handle overlapping jets. Overlapping jets are usually prevented by removing the particles assigned to each complete jet when it is formed. Seed particles may be chosen randomly, or may be the highest p_{T} particles. Either choice has problems: if the seed particles are chosen randomly then a soft emission is capable of being a seed, and potentially changing the final jet configuration; if seed particles are those with highest p_T then a collinear splitting could change the seeds, and potentially the final configuration. Thus, this form of iterative cone is not IR safe.

There are different cone algorithms that do have the property of IR safety. SISCone [117] is an example of this, it achieves IR safety by avoiding seed particles all together, and finding every possible stable cone.

4.6.3.3 Agglomerative Algorithms

Agglomerative clustering algorithms are algorithms that repeatedly join single units or clusters to other single units or clusters until desired clusters are obtained. This results in a hierarchy of elements. The inverse of an agglomerative clustering algorithm is a divisive clustering algorithm.

In the context of jet formation, this proceeds as follows [118];

1. At the start, declare all particles to be pseudojets. A pseudojet being an incomplete jet. Each pseudojet has the same momentum as the particle.

2. Calculate the pairwise distance between all pseudojets, $d_{i,j}$. Also calculate the beam distance of each pseudojet, d_{iB}.

3. Find the smallest distance from both the $d_{i,j}$ or the d_{iB};

   (a) If the smallest distance is a pairwise distance, join those two closest pseudojets to form a new pseudojet, so that there is now one less pseudojet in the event. The momentum vector of the new pseudojet is determined by the recombination scheme.

   (b) If the smallest distance is a beam distance, promote the corresponding pseudojet to a jet and remove it from consideration.

4. If any pseudojets remain, return to step item 2.

Two things have been left unspecified here; the nature of the distance metrics, $d_{i,j}$ and d_{iB}, and the nature of the recombination scheme. The most common recombination scheme is exactly what one might expect; the 4-momentum of the two pseudojets are simply added together. This is called E-scheme recombination [105]. Other schemes are possible.

Distance metrics are far more varied. A common choice is a distance metric from the generalised k_{T} scheme where;

$$d_{i,j} = \min({p_{\mathrm{T}_i}}^{2q}, {p_{\mathrm{T}_i}}^{2q})(\delta y_{i,j}^2 + \delta \phi_{i,j}^2) \tag{4.10}$$

and

$$d_{iB} = {p_{\mathrm{T}_i}}^{2q} \cdot R^2 \tag{4.11}$$

where R and q are constants.

The value of R influences the size of the jet, the larger R is the wider the jets can become. Values of R between 0.4 and 0.8 are common.

The maximum possible width of a jet with $q = 0$ is $3R$. This can occur like so;

- Place particles a, b, c and d on a straight line at $\phi = 0$, with rapidities 0, $R - 2\epsilon$, $2R - 3\epsilon$ and $3R - 5\epsilon$, where ϵ is a small number. Let particles a and d have momentum proportional to $\epsilon/4$, and particles b and c have momentum proportional to 1.

- Initially, the closest parings are a, b and c, d, both being $R - 2\epsilon$ apart. These will merge first, call the new particles created b' and c'.

- As a and d have small momentum $\epsilon/4$, b' and c' will each be shifted for the original locations of b and c by less than $\epsilon/4$; thus they are guaranteed to still be within R of each other.

- b' and c' merge, so that all four original particles are in one jet.
- This jet has a width $3R - 5\epsilon$, so as ϵ tends to zero, the width of the jet tends to $3R$.

Should there be any particle before a or after d, merging that particle with a or d respectively would move the result out of range of b or c. Thus, the jet can get no wider.

In general, let the jet width be the maximum value of $\sqrt{\delta y_{i,j}^2 + \delta\phi_{i,j}}$ for any i and j corresponding to particles in the jet, where $\delta y_{i,j} = (y_i - y_j)$ and $\delta\phi_{i,j}$ is the distance between ϕ_i and ϕ_j, taking into account the cyclic properties of the coordinate. It is also true that the mean jet width will be greater that R. This can be seen in Figure 4.10. So

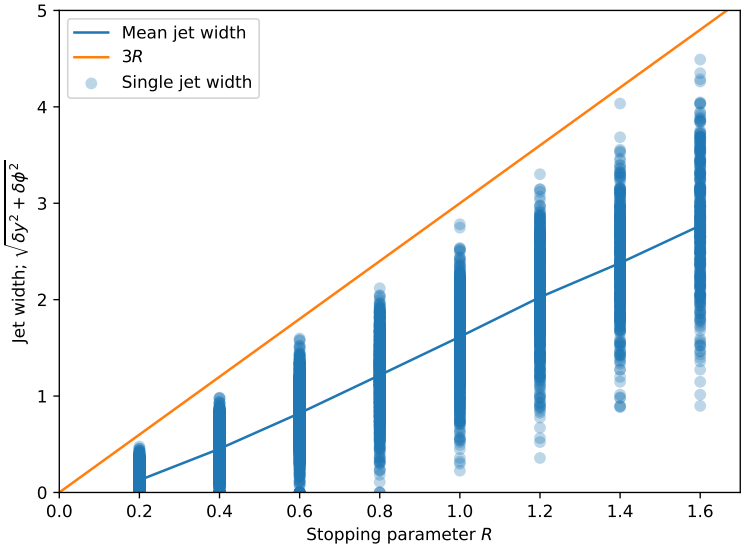


FIGURE 4.10: A small sample of 24987 Cambridge-Aachen jets ($q = 0$), formed on MC data, are plotted against their stopping parameter, R. This demonstrated that the stopping parameter is proportionate to the width, but that it is not equal to the average or maximum width.

while R is often referred to as a jet width, or jet opening angle, it is not strictly either of these things. There is a strong, linear, correlation between the jet width and R.

The value of q determines how the algorithm treats hard emissions. When $q = 0$ the algorithm treats all emissions the same, this algorithm is known as a Cambridge-Aachen algorithm [119, 120]. When $q = 1$ the algorithm tends to gather soft particles first, this algorithm is known as a k_T algorithm [121, 122]. When $q = -1$ the algorithm tends to gather about hard centres, and produces particularly round jets, this algorithm is known as an anti-k_T algorithm [123, 124, 125].

When $q \neq 0$ then the radius of the jets has a more complex relationship to R. It remains true that the greatest distance over which two pseudojets can merge is R. If $q < 0$, the 'min' in Equation 4.10 picks out the larger p_T pseudojet. Taking i as the higher p_T object, then, $d_{i,j} = {p_T}_i^{2q}(\delta y_{i,j}^2 + \delta\phi_{i,j}^2)$, $d_{iB} = {p_T}_i^{2q} R^2$ and $d_{jB} = {p_T}_j^{2q} R^2$. It is clear that $d_{iB} < d_{jB}$,

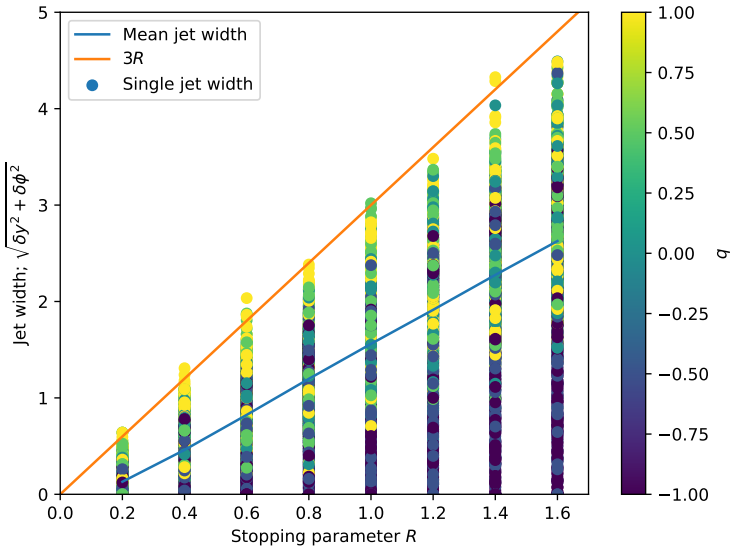


FIGURE 4.11: A sample of 123507 generalised k_T jets, with $-1 < q < 1$, formed on MC data, are plotted against their stopping parameter, R. Again, the stopping parameter is proportionate to the width, but is not equal to the average or maximum width. Also, note that $3R$ is no longer a hard limit on jet width.

so the next step will not be to remove j. Provided $\delta y_{i,j}^2 + \delta\phi_{i,j}^2 < R$ then i and j will merge, otherwise, if no other measure is smaller i will be removed. On the other hand, if $q > 0$, the 'min' in Equation 4.10 picks out the lower p_T pseudojet. Keeping the same convention, j will be the lower p_T object, then, $d_{i,j} = {p_T}_j^{2q}(\delta y_{i,j}^2 + \delta\phi_{i,j}^2)$, $d_{iB} = {p_T}_i^{2q} R^2$ and $d_{jB} = {p_T}_j^{2q} R^2$. Provided $\delta y_{i,j}^2 + \delta\phi_{i,j}^2 < R$ then i and j will merge.

What changes, in the case of $q \neq 0$, is that pseudojets may not merge with the next closest pseudojet in terms of $\delta y_{i,j}^2 + \delta\phi_{i,j}^2$. This is an important requirement for the hard limit of $3R$. Now the maximum possible jet radius is at least $3R$. Finding this maximum is non trivial, but that it may be greater than $3R$ can be seen in Figure 4.11.

Generalised-k_T algorithms are all IR safe. All collinear splitting will result in pseudojets with $d_{i,j} = 0$, and so they will be merged immediately. Any pseudojet that merges with a soft pseudojet will result in a new pseudojet in exactly the same location. As the clustering has no memory of previous steps, it will then proceed as if the soft particle or collinear splitting had never existed.

Chapter 5

Revisiting Jet Clustering Algorithms For 2HDM Signals

This section is drawn from the work published in [3]. This work was co-authored with Amit Chakraborty, Srinandan Dasmahapatra, Billy G. Ford, Shubhani Jain, Stefano Moretti, Emmanuel Olaiya and Claire H. Shepherd-Themistocleous.

Decisions made about direction and content were primarily driven by Professor Amit Chakraborty, Professor Stefano Moretti and Billy Ford, with Professor Claire H. Shepherd-Themistocleous and Dr Emmanuel Olaiya offering guidance from a up to date experimental understanding. Billy Ford and Shubhani Jain ran the primary data pipeline, which used `MadGraph`, `Pythia8`, `Delphes`, `FASTJET` and `Madanalysis`, to perform the calculations required. I constructed a complementary pipeline, using the same data drawn from `MadGraph` and `Pythia8` but then processing and forming jets with custom built `python3` programs. This custom pipeline could not process the same volume of data processed by the primary pipeline, but it served two supplementary purposes. Firstly, as the secondary pipeline was all handwritten, it facilitated obtaining information about individual events or intermediate points of the process. So it served to verify the source of imperfections in the data, such as edge effects, missing b-jets, and incorrect mass reconstructions. Some of these could be remedied with adjusted parameters, others only accounted for. Secondly, having a point of comparison, even one with somewhat limited statistics, allowed us to replicated the distributions created by the primary pipeline, aiding debugging, and increasing confidence in the results. Billy Ford and Shubhani Jain constructed the plots from the data generated by the primary data pipeline. The group collectively analysed the findings, and the text of the original publication [3] was a collective effort.

5.1 Introduction

Continuing on the theme that was opened in chapter 2, this study explores the means of resolving signals from the 2HDM. In particular, the question of whether different jet clustering techniques might be more or less suited to resolving topologies involving Higgs particles from the 2HDM. In such scenarios, as will be explained in more detail, high b-jet multiplicity final states are expected and a point worth addressing is which current experimental jet reconstruction is in fact optimal for these types of searches.

As covered in section 2.1.4, the 2HDM introduces 5 new Higgs states, labelled as h, H (which are CP even with, conventionally, $m_h < m_H$), A (which is CP-odd) and a pair of charged states with mixed CP properties, $H^{\pm}$. In section 3.2, the SM Higgs was associated with the lighter, CP even Higgs, h. This section will consider one example of each case, that of $m_h = 125$ GeV and $m_H > 125$ GeV and it's inverse $m_h < 125$ GeV and $m_H = 125$ GeV.

In a study that investigates limits due to statistics and collision energy, such as that presented in chapter 3, heavy particles are the greater challenge. They require more energy to produce and are produced less frequently. In a study that is focused on mass peak identification, such as the one presented in this chapter, a lighter particle poses a greater challenge. The lighter particle's mass peak is more likely to be lost in pileup, or hidden by the numerous backgrounds. Hence, this chapter offers two points for comparison. These chosen benchmark points will be described in section 5.2.1.

When $m_h < m_H/2$ or $m_A < m_H/2$, the decays $H \to hh$ and/or $H \to AA$ (respectively) may occur. These processes are often referred to as Higgs cascade decays, see section 2.1.4.3. Then, taking H as the SM-like 125 GeV Higgs boson, for a h state with a mass of order 60 GeV or less, the dominant decay mode in a 2HDM is bottom-antibottom quark pairs [55, 56], i.e., $h \to b\bar{b}$. In which case, the final state emerging from the hard scattering $pp \to H \to hh$ is made up, at the partonic level, of four (anti)quarks[1], see Figure 5.1. However, due to the confinement properties of Quantum Chromo-Dynamics (QCD), the partonic stage is not accessible by experiment, only the stable particles the end of the parton shower and hadronisation phase are seen. Jet clustering algorithms gather these stable particles into exclusive sets known as jets, and these exclusive sets aim to represent partons emerging from the hard interaction; a review of this is given in chapter 4.

Alongside the most common jet clustering algorithms described in section 4.6.3, there are a number of lesser known alternatives. The purpose of this study is to determine whether alternative jet reconstruction tools, in particular a modification to traditional sequential combinations algorithms employing a variable inter-jet distance measure [126] (so-called 'variable-R' algorithms, where R represents a typical cone size

[1] Notice that the same argument can be made for the case of $pp \to H \to AA \to b\bar{b}b\bar{b}$ when $m_A < m_H/2$.

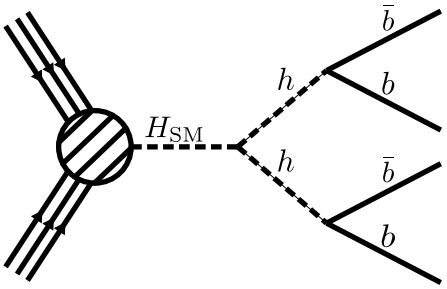


FIGURE 5.1: The 2HDM process of interest, where the SM-like Higgs state ($m_H = 12$ GeV) produced from gluon-gluon fusion decays into a pair of lighter scalar Higgs states, hh, each in turn decaying into $b\bar{b}$ pairs giving a four-b final state.

characterising the jet), might be better suited to the four-b final states coming from 2HDMs. Furthermore, the four-b final state that is invoked here is an ubiquitous signal of BSM Higgs boson pairs which are lighter than the SM one so that they can be produced from it[2]. Crucially, such states give access (through the extraction of the h state properties) to key features of the underlying BSM scenario, e.g., in the form of the shape of the Higgs potential, hence, of the vacuum stability and perturbative phases of it.

While the above outlines that the problem of optimal jet reconstruction is clearly an experimental endeavour, it is stressed that this study is undertaken at a theoretical level. It aims are to employ a simplified analysis in order to compare the relative performance of traditional fixed-R jet clustering, as described in section 4.6.3, against a variable-R method, which will be described here. A comprehensive, more realistic, experimental investigation is left to a future study. For example, another key feature of the hadronic final state initiated by b-quarks that will be studied is that the emerging jets can be "tagged" as such, unlike the case of lighter (anti)quarks and gluons, which are largely indistinguishable from each other. Here, a simplified method of tagging is implemented using Monte Carlo (MC) truth information on the b-partons, along with a probabilistic implementation of inefficiencies. For a more detailed discussion on b-tagging at detectors, please refer to [127]. A short replication study of some taggers used by CMS can be seen in Appendix A.

[2]Here, ubiquitous refers to the fact that this signal is very typical of a variety of BSM scenarios, so that in effect the 2HDM can be used for illustration purposes. These results can therefore be applied to the case of other new physics models.

Label	m_h (GeV)	m_H (GeV)	m_A (GeV)	$m_{H^\pm}$ (GeV)	$\tan\beta$	$\sin(\beta-\alpha)$	m_{12}^2
BP1	125	700	847.2	744.2	2.355	-0.999	1.46×10^5
BP2	60	125	620	400	1.6	0.1	4×10^3

TABLE 5.1: The 2HDM, Yukawa type II, parameters for the benchmark points used here. Note that, in both cases, $\lambda_6=\lambda_7=0$ is chosen.

Label	m_H (GeV)	BR($H\to hh$)	m_h (GeV)	BR($h\to b\bar{b}$)	σ (pb)
BP1	700	6.128×10^{-1}	125	6.164×10^{-1}	1.870×10^{-1}
BP2	125	6.764×10^{-1}	60	8.610×10^{-1}	6.688

TABLE 5.2: The 2HDM, Yukawa type II, branching ratios and cross sections for the process in Figure 5.1. Masses are repeated in grey, to clarify the source of the differences between the rows.

5.2 Methodology

This section begins by defining the 2HDM parameter values that are used, two benchmark points. It then introduces the variable-R jet clustering method, and offers some justification for the unusual choice. Following that, it describes the data and analysis pipeline used to create the results in the next section.

5.2.1 2HDM Benchmarks

Two parameter value sets from the 2HDM are used in this study. The lighter cascade has $m_h = 60$ GeV and $m_H = 125$ GeV, so m_H plays the role of the SM Higgs. The heavier cascade has $m_h = 125$ GeV and $m_H = 700.668$ GeV, this time m_h plays the role of the SM Higgs.

These two choices have been tested (and pass as not currently excluded) against theoretical and experimental constraints by using `2HDMC` [24], `HiggsBounds` [128] and `HiggsSignals` [129], as described in section 3.2. Flavour constraints were checked with SuperISO [130]. Using SuperISO, the following flavour constraints on b-meson decay Branching Ratios (BRs) and mixings are tested to a 2σ level: BR($b\to s\gamma$), BR($B_s\to\mu\mu$), BR($D_s\to\tau\nu$), BR($D_s\to\mu\nu$), BR($B_u\to\tau\nu$), $\frac{\text{BR}(K\to\mu\nu)}{\text{BR}(\pi\to\mu\nu)}$, BR($B\to D_0\tau\nu$) and $\Delta_0(B\to K^*\gamma)$.

The production and decay rates for the subprocesses $gg, q\bar{q}\to H\to hh\to b\bar{b}b\bar{b}$ are presented in Table 5.1, alongside the 2HDM-II input parameters. Notice that the H and h decay widths are of order MeV, hence much smaller than the detector resolutions in two- and four-jet invariant masses, respectively, so that the Higgs states can essentially be treated as on-shell. In the calculation of the overall cross section, the renormalisation and factorisation scales were both set to be $H_T/2$, where H_T is the sum of the

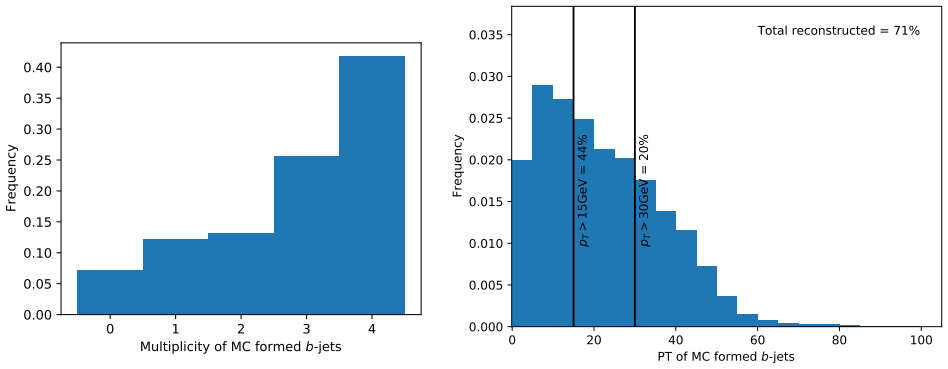


FIGURE 5.2: A sketch of the behaviour of a highly idealised, MC based, clustering algorithm on the 40 GeV Higgs cascade decay. On the left, the multiplicity of b-jets in each event is shown, on the right, the p_T of those b-jets is shown. Percentages given are percentages of the total b-quarks produced, which are represented as jets, both in total, and after various cut possibilities. Cuts have been applied to the input particles.

transverse energy of each parton. The Parton Distribution Function (PDF) set used was `NNPDF23_lo_as_0130_qed` [131].

In the development of this study a third point was considered, with $m_h = 40$ GeV and $m_H = 125$ GeV. It was not retained because this configuration will be hidden from almost any plausible analysis, no matter how well the clustering algorithm performs. To illustrate this, an idealised clustering algorithm can be used.

Say, using information from a MC simulation, particles were allocated to jets based on which particle in the hard event had started the chain of decays that created them. The parton at the top of that decay chain is referred to here as the ancestor of the particle. There would be a little ambiguity in such a scheme, because in the hadronisation step particles created by one parton will interact with particles created by another. This results in an end state particle that is related to more than one parton, this particle might be said to have multiple parton ancestors. In order to turn this information into a jet, a somewhat subjective decision must be made to allocate that particle to a parton. A reasonable criteria for this might be to assign particles with multiple ancestor partons to the parton with the closest direction of travel, which again, appeals to information only known in MC simulation. Applying this criteria for dealing with these multiple parton ancestor particles, and then assigning the particles to jet according to their ancestors, gives an unrealistically good jet clustering algorithm. Using such an unrealistic algorithm allows us to put limits on the potential of any realistic algorithm.

Using the smaller secondary data pipeline, such an jet formation algorithm was designed. Basic cuts were applied to the particles, requiring each particle have $p_T > 0.5$ GeV and $|\eta| < 2.5$. Even this algorithm, with unrealistic access to MC information, cannot perfectly locate all the b-quarks. This can be seen in Figure 5.2. There are many events whose multiplicity is less than 4, because one or more of the b-quarks left no

trace that passed the aforementioned cuts. As seen in the right hand plot, 71% of the b-quarks could be identified as a jet, but, if that jet must have at least $p_T > 30$ GeV then only 20% of the b-quarks are represented by a jet. Given this clustering was performed with the aid of MC truth, any real clustering algorithm can be expected to fair worse. A 30 GeV cut should be expected to eliminate a sizeable majority of the signal.

While this may not be a perfect analysis, the data set used is only $\mathcal{O}(10^3)$, the picture it paints is very decisive. Reconstructing such a benchmark is certainly implausible.

5.2.2 Jet Clustering Algorithms

While jet clustering algorithms have included a great diversity of designs, the ones of interest are agglomerative algorithms, briefly reviewed in section 4.6.3.3.

The first algorithm that is used here is the generalised k_T algorithm. Repeating the relevant points from section 4.6.3.3; this algorithm has a distance metric of

$$d_{i,j} = \min({p_\mathrm{T}}_i^{2q}, {p_\mathrm{T}}_i^{2q})(\delta y_{i,j}^2 + \delta\phi_{i,j}^2) = \min({p_\mathrm{T}}_i^{2q}, {p_\mathrm{T}}_i^{2q})\Delta R^2 \tag{5.1}$$

where $\delta y_{i,j} = (y_i - y_j)$ and $\delta\phi_{i,j}$ is the distance between ϕ_i and ϕ_j, taking into account the cyclic properties of the coordinate. This algorithm also makes use of the 'beam distance', which is the separation between object i and the beam B,

$$d_{iB} = {p_\mathrm{T}}_i^{2q} \cdot R^2 \tag{5.2}$$

where R and q are constants. Note that this notation mimics that of [126], where R^2 is included in the definition of d_{Bi}. (An alternative convention is to embed R^2 into the definition of d_{ij} such that $d_{ij} = \min(p_{Ti}^{2q}, p_{Tj}^{2q})\frac{\Delta R_{ij}^2}{R^2}$, leaving $d_{Bi} = p_{Ti}^{2q}$, like in [123].) For a set of particles, all possible d_{ij}'s and d_{Bi}'s are calculated and the minimum is taken. If the minimum is a d_{ij}, objects i and j are combined and the process is repeated. If, instead, a d_{Bi} is the minimum, then i is declared a jet and removed form the sample. This procedure is then repeated until all objects are classified into jets.

In d_{Bi}, R is a fixed input variable which dictates the size of the jet, and acts as the cut-off for any particle pairing. Considering some pair of particles i and j, with i having lower p_T (and hence being selected in d_{ij}), then for $q \geq 0$

$$d_{ij} = \Delta R_{ij}^2 p_{Ti}^{2q} = \frac{\Delta R_{ij}^2}{R^2} d_{Bi}. \tag{5.3}$$

So long as, for some j, the ratio $\frac{\Delta R_{ij}^2}{R^2} < 1$ then i will undergo further merges, rather than forming a new jet. If $q = 0$, then the maximum jet radius becomes $3R$, as demonstrated in section 4.6.3.3[3]. From this general formulation, the main two jet clustering algorithms currently in use at the LHC are the Cambridge-Aachen (CA) [120, 119] one and the anti-k_T [123] one, which use the above expressions with $q = 0$ and -1, respectively [105].

5.2.3 Jet Clustering with Variable-R

There has, in fact, been a more recent development to these techniques. One notices that the above algorithms require as input a fixed parameter, R, which in the case of the anti-k_T algorithm is correlated with jet radius, as discussed and illustrated in section 4.6.3.3. Recall this acts as a cut off for combining hadrons and can therefore guides the size of the jets.

From the point of view of the shower, the angular spread of the final constituents has a dependence on the initial partons p_T. For higher p_T objects the decay products will be more tightly packed into a more collimated cone, whereas for low p_T objects one would expect the resulting shower constituents to be spread over some wider angle. One can therefore see a potential advantage in selecting the R value used for clustering depending on the p_T of the final state jets. It is this advantage the variable-R [126] seeks to obtain.

A modification to the distance measure, d_{ij}, is made, by replacing the fixed input parameter R with a p_T dependent $R_{\text{eff}}(p_T) = \frac{\rho}{p_T}$, where ρ is a chosen dimensionful constant (taken to be) $\mathcal{O}(\text{jet } p_T)$. With this replacement, the beam distance measure becomes

$$d_{Bi} = p_{Ti}^{2q} R_{\text{eff}}(p_{Ti})^2. \tag{5.4}$$

When the distance measures are calculated, d_{Bi} will therefore be suppressed for objects with larger p_T and hence these objects become more likely to be classified as jets. For low p_T objects, d_{Bi} is enhanced and so these are more likely to be combined with a near neighbour, thus increasing the spread of constituents in the eventual jet.

This work hypothesises that, in multijet signal events where one might expect signal b-showers with a wide spread of different p_T's, a variable-R reconstruction procedure could improve upon the performance of traditional fixed-R routines. In particular, using a variable-R alleviates the balancing act of finding a single fixed cone size that suitably engulfs all of the radiation inside a jet, without sweeping up too much outside 'junk'.

[3] For $q \neq 0$ this maximum no longer strictly holds, although it does still tend to be true.

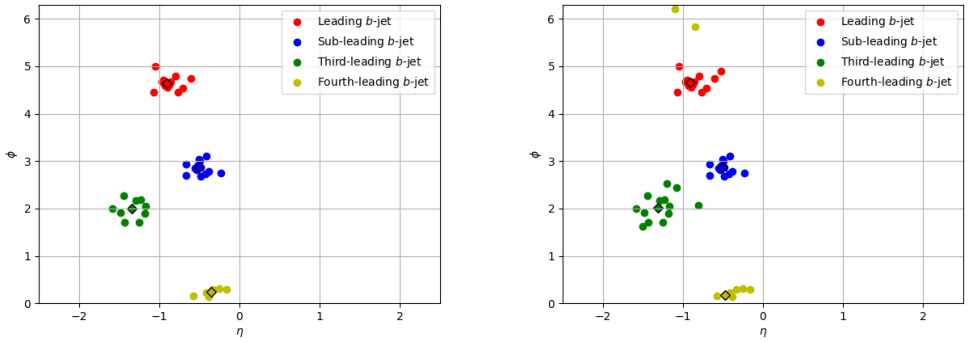


FIGURE 5.3: The same MC event in (η, ϕ) space. Tracks have been clustered with (left) a fixed $R = 0.4$ and (right) variable-R algorithm. The coloured points are the constituents of the corresponding b-jet in the legend and black outlined diamonds are at the overall (η, ϕ) coordinates of the formed b-jet. The anti-k_{T} algorithm is used in both cases.

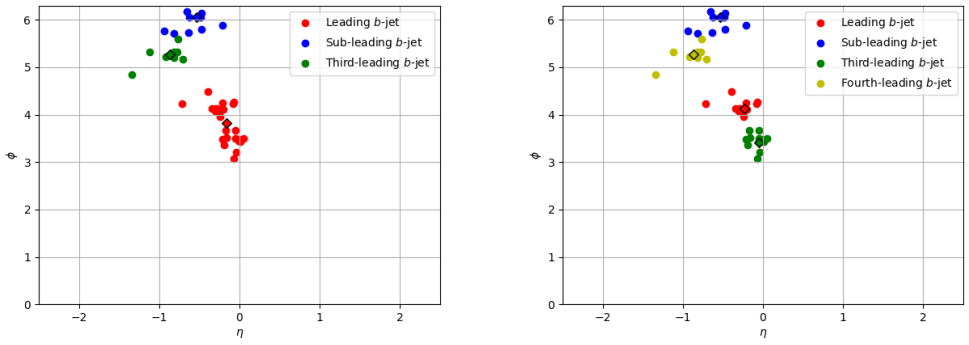


FIGURE 5.4: Same plot as in Figure 5.3, however, here, the given event is clustered into three b-jets when a fixed $R = 0.8$ is used (left) and four b-jets when a variable-R approach is used (right).

As a brief visualisation, the constituents of b-tagged jets (herafter, b-jets for short) in the same event can be displayed, which have been clustered using both a variable-R and fixed $R = 0.4$ scheme, as seen in Figure 5.3. Notice that, for the leading and sub-leading b-jets, the jet content is roughly the same. For the lower p_T jets, however, the variable-R jets gather a wider cone of constituents. Provided all jets are resolved, a wider cone increases the chances that all decay products of the signal are captured, improving the ability to accurately reconstruct Higgs masses when analysing b-jets. Figure 5.4 shows a case where using a larger fixed cone ($R = 0.8$), to try and gather all of the constituents, only resolves three b-jets. Variable-R however 'finds' all four b-jets expected from the signal. It can be seen that fixed-R sweeps radiation from a nearby jet into the leading b-jets, whereas variable-R is able to resolve both due to the larger p_T (and hence smaller R_{eff}) of the leading b-jet, while also having a large enough cone to suitably reconstruct the lower p_T jets.

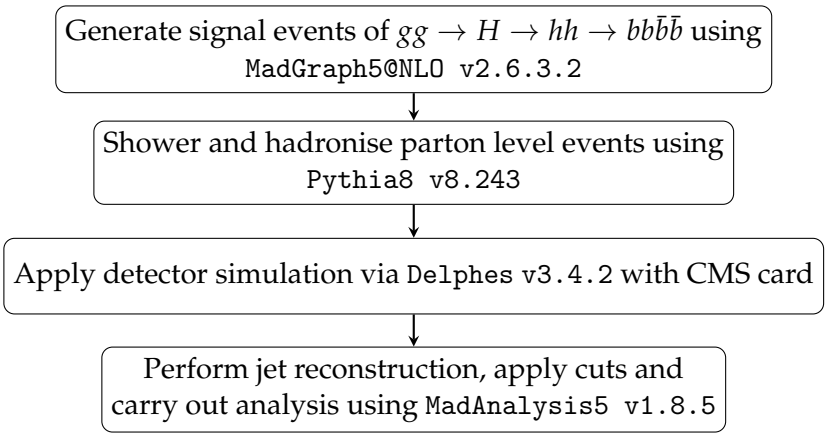


FIGURE 5.5: Description of the procedure used to generate and analyse MC events.

5.2.4 Implementation of b-Tagging

For this study, a simplified MC informed b-tagger is implemented. For events clustered using a fixed-R cone size, jets within angular distance R from each parton level b-(anti)quark are searched for and tagged as appropriate. For scenarios where multiple jets are found, the closest is taken as the assignee for the b-tag. When the variable-R approach is used, the size of the tagging cone is taken as the effective size of the jet, R_{eff}, defined above.

In addition, when the signal and background rates are considered, the finite efficiency of identifying a b-jet as well as the non-zero probability that c-jets and light-flavour plus gluon jets are mistagged as b-jets is accounted for. After the jets are run through the MC b-tagger described above, the mistag rates found in the `Delphes` CMS card are applied.

5.2.5 Data Generation

In order to carry out a realistic MC simulation, the toolbox described in Figure 5.5 is used to generate and analyse events [71, 85, 132, 133][4]. This is used to create simulation of observations for both BP1 and BP2, as described in section 5.2.1. Samples of $\mathcal{O}(10^5)$ events are generated, with $\sqrt{s} = 13$ TeV. The Parton Distribution Function (PDF) set used was `NNPDF23_lo_as_0130_qed` [131].

[4]Note that the Leading Order (LO) normalisation for the signal cross sections is used here, for consistency with the fact that most of the backgrounds in the forthcoming analysis are only implemented at LO. While this affects the final results on event rates and significances, the results are sufficient for the purposes of this study; to assess the jet clustering performance, rather then the exact values of signal and backgrounds rates.

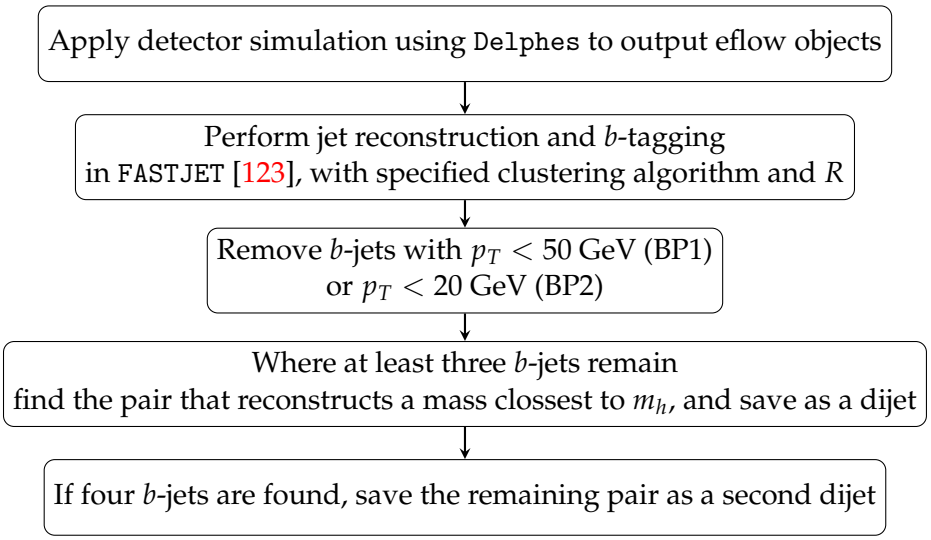


FIGURE 5.6: Description of our initial procedure for jet clustering, b-tagging and selection of jets.

5.2.6 Cutflow

Before introducing the full sequences of cuts adopted here, some discussions on their possible choice are needed. In existing four b-jet analyses by the ATLAS and CMS collaborations, seeking to extract chain decays of Higgs bosons like the ones considered here from the background, comparatively restrictive cuts have been used for the ensuing fully hadronic signature. Taking CMS as an example, upon enforcing the same p_T cuts on b-jets as in [134] on BP2, the signal selection efficiency was too low. It would not be possible for any jet clustering algorithm to create high enough multiplicities to yield a useful sample, assuming luminosities of run 2 and 3. This issue was already touched on in section 5.2.1.

For BP1, the cuts used in [135] are suitable. These require b-jets in the event have a p_T of at least 50 GeV. For BP2, all jets are required to have at least 20 GeV. It remains to be seen if this is viable at the LHC, but for the purposes of comparing the behaviour of different jet clustering algorithms it yields samples of a useful size.

Then, having removed jets with insufficient p_T, those events that are left with at least 3 jets, are used to reconstruct at least one m_h. Events that have kept 4 jets can reconstruct both m_h, and therefore m_H as well.

As the h decays into two b-quarks, two b-jets are selected, and the energy and momentum of these two b-jets is added. The mass of this dijet object, m_{bb} is considered to be the reconstructed estimate of m_h. When all 4 b-quarks are identified with separate jets in an event, there are 6 possible pairings. Using information in the MC, it is possible to identify which of these pairings is the correct choice; the two b-jets which have been tagged by b-quarks which decayed from the same h should be matched. This MC pairing

would not be possible in an experimental setup, so it might be seen as a somewhat idealised method. It is also not readily attainable information when suing `Madanalysis`. For these reasons, the primary pipeline did not seek to implement a matching based on MC pairings, that is, information about which b-quarks had decayed form which light Higgs. Instead, an estimate based on information that would be experimentally available is used.

Many different approaches have been taken in other work. When an experimental tagger is used, there will be varying confidence in each jet's classification, and this maybe used to form dijet pairs [136, 137]. That isn't possible when using an MC tagger. Sometimes the highest p_T pair is considered a dijet [138, 139]. This study did attempt this method, but it was not very successful.

Alternatively, many studies involving four jets in the final state select dijet pairs that minimise the mass difference between the dijet pairs [140, 141, 142]. This study chose to include events containing only 3 b-tagged jets, and so this method was not applicable.

Another common method, is to select the dijet pair who mass best matches the target mass, that is, minimise the difference between the mass of the dijet and the mass of the particle being reconstructed [143, 144, 145]. This method is followed in this study. It could be seen as an optimistic choice, so its accuracy is evaluated here, by comparison to MC ground truth.

Only events with at least 3 b-jets are used, so unless one b-jet is in fact the merging of two b-quarks, it is guaranteed that there is at least one correct (as in, representing two b-quarks from the same light Higgs) pairing available. In cases where two b-quarks have merged in to one jet they are quite likely to be from the same h, particularly in the $m_h = 60$ GeV case, as seen in Figure 5.8. In that case, the remaining two b-jets are still a correct pairing. However, occasionally, two b-quarks from different Higgs, may by chance merge. This would be more likely for $m_h = 125$ GeV, where the separation between the light Higgs is on average smaller, and the separation between the b-quarks is on average larger (again, see Figure 5.8). In this case there would not be any correct pairing available. This is only a small minority of events.

Having established that at least one correct pairing should be available in events with three or four b-jets, a decision must be made based on experimentally accessible information; the pair minimising $|m_{bb} - m_h|$ is chosen. In the case where all four b-jets are reconstructed, the left-over $b\bar{b}$ pair is also included so as to account for the presence of the second light Higgs state.

Using the secondary data pipeline, where MC variables are more readily available, the accuracy of this scheme is then evaluated. This is shown in Figure 5.7. It can be seen that while minimising $|m_{bb} - m_h|$ frequently does not find the same pairings as would be found by MC, the masses produced are not much changed. There is a slight trend to

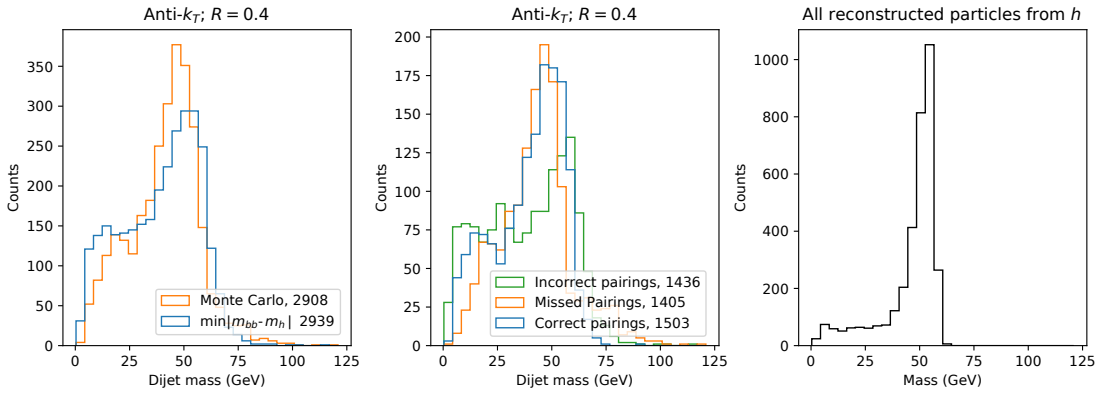


FIGURE 5.7: Evaluation of the performance of allocating b-jets to dijet pairs, such that $|m_{bb} - m_h|$ is minimised. Left panel; the comparison between the mass peak obtained with MC matching, and minimisation matching. Centre panel; dijets sorted according to matching outcome. A dijet pairing is counted as correct when it is joined by both MC and mass minimisation. Pairings made by mass minimisation that do not match the MC pairing are labelled as incorrect, and pairings made by MC information that are not found by mass minimisation are labelled as missed. Right panel; the combined mass of all particles created by the h, which can be reconstructed. This shows the mass loss expected due to detector sensitivities. The clustering algorithm used is the anti-k_T algorithm, with $R = 0.4$, the data used is generated according to BP2.

create higher masses by minimising $|m_{bb} - m_h|$. Overall, this could be seen as a realistic source of error.

It would be possible to reduce this error, by matching to a slightly lower mass than the target mass, thus accounting for particles lost due to detector limitations. There is some precedent for this [146], but the more conventional method is retained here, to aid comparisons with other work.

5.3 Results

This section presents the results for the signal at both the parton and detector level. In the latter case, it also discusses the dominant backgrounds, due to QCD $4b$ production, $gg, q\bar{q} \to Zb\bar{b}$ and $gg, q\bar{q} \to t\bar{t}$[5].

5.3.1 Parton Level Analysis

At the Matrix Element (ME) level, all the events have four b-quarks originating from the decay of the two light Higgs bosons (h). In the upper panel of Figure 5.8, the R separation between the b-quarks coming from the same light Higgs state is plotted. The two distributions corresponding to BP1 ($m_h = 125$ GeV) and BP2 ($m_h = 60$ GeV) are markedly different. This can be understood as follows. In general, the angular separation between the decay products a and b in the resonant process $X \to ab$ can be approximated as $\Delta R(a,b) \sim \frac{2m_X}{p_T^X}$. Hence, the in the lower left panel of Figure 5.8 the transverse momentum of each of the h bosons is plotted.

For $m_h = 60$ GeV, in BP2, the light Higgs boson has less p_T than in BP1, owing to the smaller $m_H - m_h$ mass difference. Therefore, the b-quarks are more widely separated in this case, compared to $m_h = 125$ GeV. In the light of this, it is concluded that there is a strong correlation between the lightest Higgs boson mass and the cone size of the jet clustering algorithm that might be used without risking merging jets. Two key ways to improve multiplicities are preventing pairs of quarks being merged into the same jet, and ensuring each individual jet gains enough p_T to pass kinematic cuts. In order to maximise the number of jets for different choices of the light Higgs boson mass, the jet radius parameter ought to be varied. That is, a fixed jet radius parameter may not be suitable here for all m_h choices.

Finally, in the lower right panel of Figure 5.8, the ΔR separation between the two light Higgs states is plotted. For BP1, with $m_h = 125$ GeV, it is clear (since $\Delta R \approx \pi$) that the $H \to hh$ decay is dominantly back-to-back (in the laboratory frame). However, for BP2, with $m_h = 60$ GeV, there is a double peak structure. This occurs due to a recoil effect from ISR, which only becomes apparent at the mass boundary where $m_H \simeq 2m_h$. The inability of the two emerging h states to fly apart implies some overlapping of the b-quark momenta. This overlapping momentum increases the risk that the b-jets are merged if the jets become too large, lowering the b-jet multiplicity. Low mass is also expected for b-jets in BP2, potentially reducing the number of jets that pass kinematic cuts.

[5]For this study it was also verified that the additional noise due to $t\bar{t}b\bar{b}$ events as well as hadronic final states emerging from $W^+W^-, W^\pm Z$ and ZZ production and decay are negligible, once mass reconstruction around m_h and m_H is enforced.

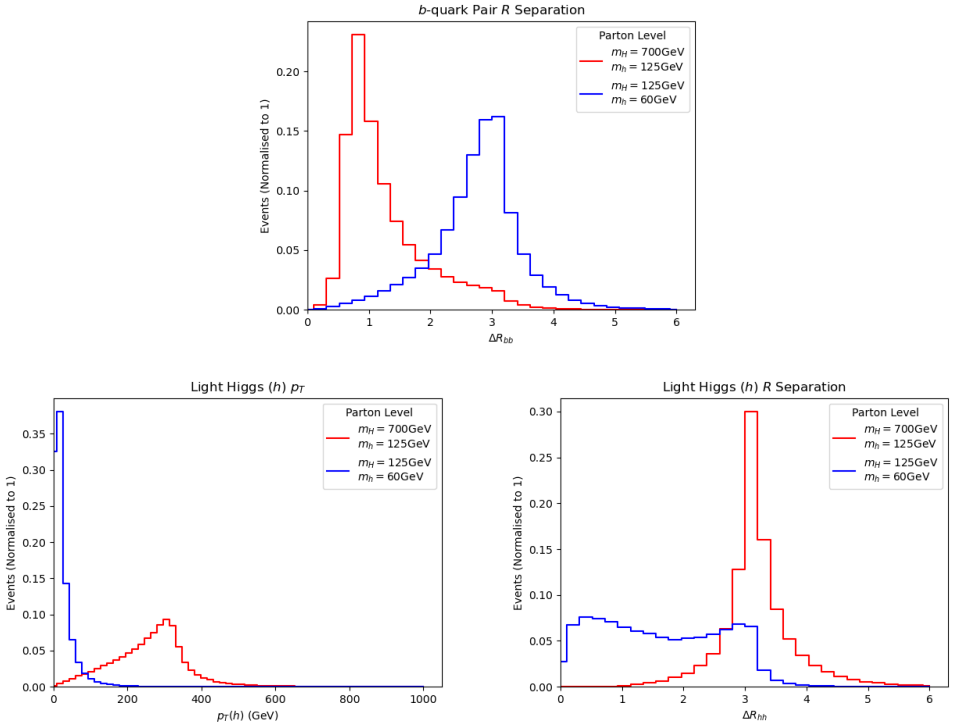


FIGURE 5.8: Upper panel: the ΔR distribution between the two b-partons originating from the same h. Lower left panel: the p_T distribution of the light Higgs boson h originating from H decay. Lower right panel: the ΔR distribution between the two h states originating from the H decay. No (parton level) cuts have been enforced here.

As a final study, in fact, the p_T of the b-quarks is plotted. This is done in Figure 5.9. From the top histogram it can be seen that in both mass configurations the b-quarks have a wide range of p_T's and hence one would expect the resulting jets to have a similar spread of p_T's. In particular, there are plots of the highest and lowest p_T's amongst the b-quarks in a given event (lower frames). Further to the discussion in section 5.2.3, one would therefore expect the resulting spread of radiation from each signal b-quark to vary in solid angle and hence the resulting jets to be of differing sizes. This thus motivates the need for a jet reconstruction sequence that behaves sensibly for jets of various cone sizes. Therefore, the next section firstly tests how jet clustering with fixed-R input behaves and then introduces the variable-R algorithm.

5.3.2 Jet Level Analysis

After establishing some expectations from the behaviour of the partons, the various jet formation algorithms can be compared. MC events that have been showered, hadronised, passed through detector simulation and then undergone particle reconstruction are used as the input to various jet clustering algorithms. This is done twice, for both BP1 and BP2, to provide examples of behaviour in different scenarios.

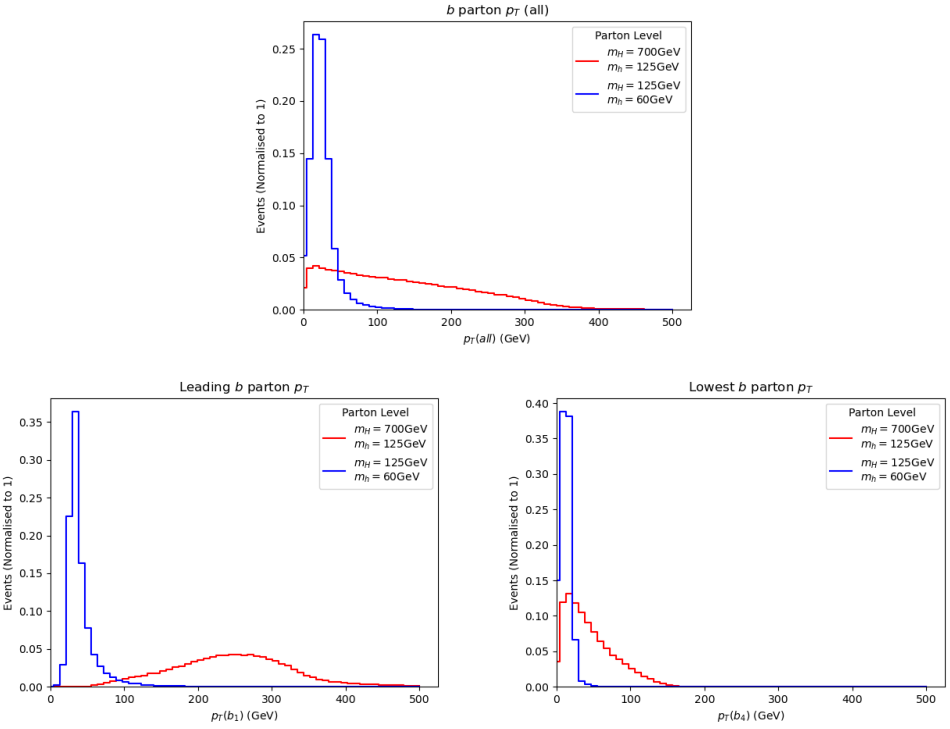


FIGURE 5.9: Upper panel: the p_T distribution for all b-quarks. Lower left panel: highest p_T amongst the b-quarks. Lower right panel: lowest p_T amongst the b-quarks. No (parton level) cuts have been enforced here.

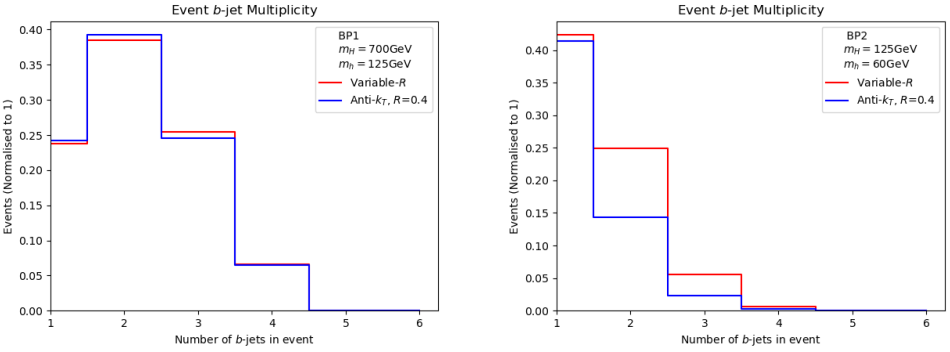


FIGURE 5.10: Left panel: the b-jet multiplicities for BP1. Right panel: the b-jet multiplicities for BP2. All cuts are enforced.

With the jets that are formed, two key indicators of success will be considered; the jet multiplicity, and the reconstructed mass spectrum. Higher jet multiplicities improve the statistics that are gained from any analysis. Also requiring an accurate jet mass spectrum prevents arbitrary manipulations that might improve multiplicity at the cost of allocating junk to the jet, or being overcautious to limit jet merging. Both of these must be avoided to gain a clear signal reconstruction.

For both the anti-k_T and variable-R algorithms, some parameter choices are needed. A

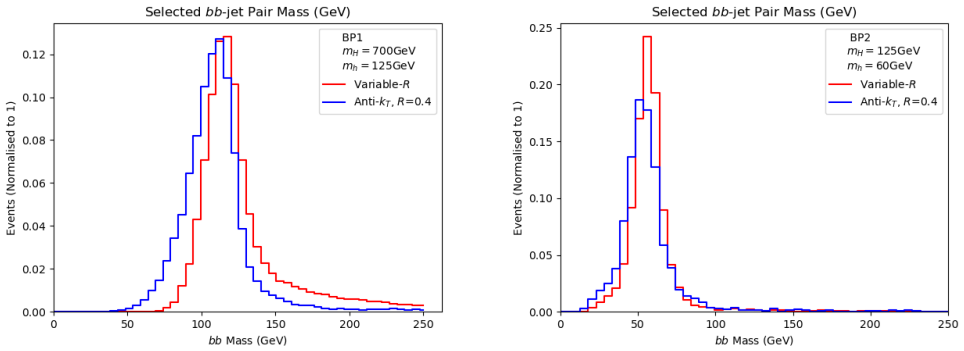


FIGURE 5.11: Left panel: the b-dijet invariant masses for BP1. Right panel: the b-dijet invariant masses for BP2. All cuts are enforced.

value of $R = 0.4$ is always used for the anti-k_T algorithm. The CA algorithm was also investigated, but the results are very similar, so it is not presented to avoid cluttering the plots. For variable-R, different parameter values are used for each of the benchmarks; for BP1, $\rho = 100$ GeV, for BP2, $\rho = 20$ GeV. These values are inferred from the p_T scale of the fixed cone b-jets. Finally, $R_{\text{min}} = 0.4$ and $R_{\text{max}} = 2.0$ are used throughout wherever variable-R is used.

To begin with, the b-jet multiplicities can be seen in Figure 5.10.

The stark difference between the two plots is due to the relative kinematics of the final state b-jets. Due to the different mass configurations, b-jets from BP2 have significantly lower p_T than those from BP1, and so significantly more are lost to the trigger, as well as from the (p_T dependent) b-tagging efficiencies. This is in line with the expectations in section 5.3.1. As for the effect of variable-R, in the BP1 case there is very small increase toward events with higher b-jet multiplicities. This shift becomes more significant in BP2, as BP2 benefits more from increased jets radius at small p_T, when this helps jets pass the kinematic cuts.

Having seen good results in multiplicity, another central objective of jet clustering should be examined; the capacity to reconstruct mass peaks.

In Figure 5.11 the advantage of the variable-R becomes more apparent. For both BP1 and BP2, the variable-R algorithm is able to get closer to the mass of the parton than the anti-k_T algorithm, and creates a better representation of the light Higgs mass. The same behaviour is seen in the four b-jet masses in Figure 5.12. The capacity of reconstruct a wider range of jet widths results in less signal loss.

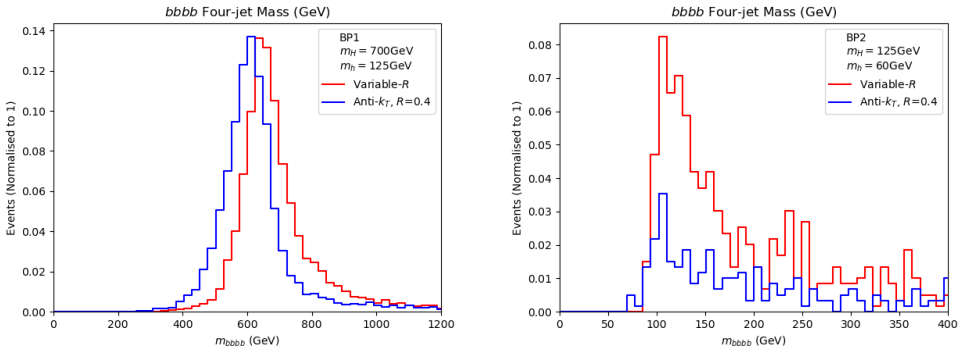


FIGURE 5.12: Left panel: the four b-jet invariant masses for BP1. Right panel: the four b-jet invariant masses for BP2. All cuts are enforced.

5.3.3 Signal-to-Background Analysis

It is also important that any jet algorithm proposed does not sculpt the backgrounds. To check this, a calculation of the signal-to-background rates is given. This is needed to properly compare the various jet reconstruction procedures described in this study, in connection with their performance in dealing with events not coming from the target BSM process. In order to do so, the selection procedure described in Figure 5.16 are applied, which bolts on the discussed reduced p_T cuts. Again, anti-k_T has been used throughout but conclusions would not change if CA was used instead. One more cut will be used for this analysis, detailed below.

5.3.3.1 Jet Quality Cuts

As per the process in the original variable-R paper [126], jet quality cuts are used to remove weak reconstructions. These are of course applied to all jet construction algorithms. They are well motivated, the concept is that a well constructed jet will have the same direction of flow in energy and transverse momentum. Jets that do not have this quality are likely not symmetric, and so ill formed.

This requires the following definitions;

$$P_E = \sum_i E_i \hat{p}_i, \quad P_{p_T} = \sum_i p_{T_i} \hat{p}_i \tag{5.5}$$

where i labels the jet constituents of the jet being assessed, $\hat{p}_i$ is the four-momentum of the ith constituent, normalised to unity and E_i and p_{T_i} are its energy and transverse momentum respectively. The quality cut applied then reads;

$$\Delta R(P_E, P_{p_T}) < \delta \tag{5.6}$$

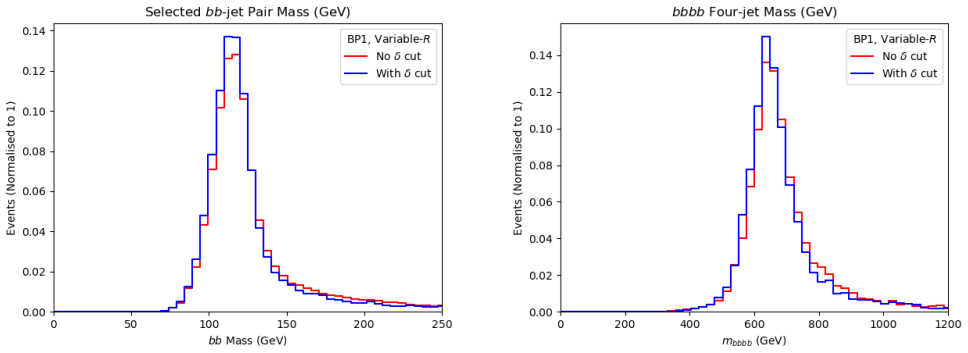


FIGURE 5.13: Left panel: the b-dijet invariant masses for BP1, with and without quality cuts. Right panel: the four b-jet invariant masses for BP1, with and without quality cuts. Here a value of $\delta = 0.05$ is used.

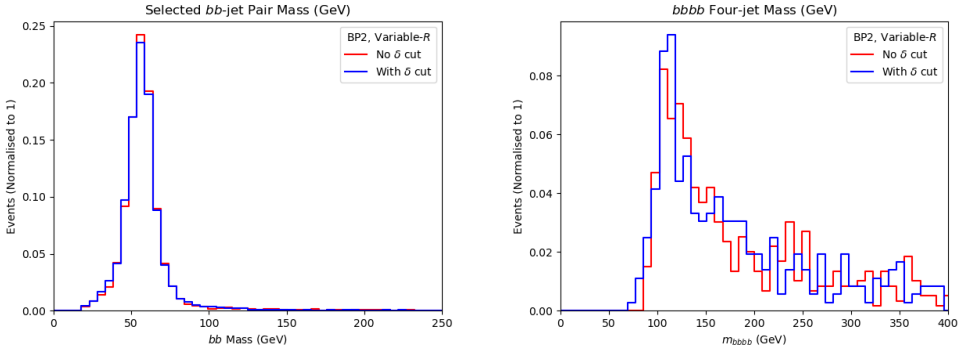


FIGURE 5.14: Left panel: the b-dijet invariant masses for BP2, with and without quality cuts. Right panel: the four b-jet invariant masses for BP2, with and without quality cuts. Here a value of $\delta = 0.1$ is used.

where δ is a user defined cut-off. To demonstrate the usefulness of these cuts, the b-dijet mass, and the four jet invariant mass, are plotted with and without these quality cuts. This can be seen in Figure 5.13 and Figure 5.14.

These cuts have improved the height of the signal peaks a little. This is particularly apparent for the heavier shower, for BP1, in Figure 5.13.

This is not the intended benefit of these quality cuts, however, the intention of the quality cuts is to reduce the impact of backgrounds.

5.3.3.2 Signal Selection

To carry out this exercise, $pp \to b\bar{b}b\bar{b}$ and $pp \to Zb\bar{b}$ and $pp \to t\bar{t}$ background processes are generated using the toolbox described in Figure 5.15[71, 85, 132, 133]. Table 5.3 contains the cross sections in pb for signal and the various background processes upon applying the aforementioned cuts and mass selections.

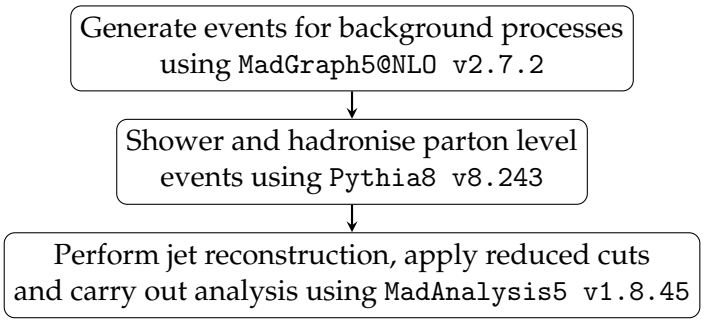


FIGURE 5.15: Description of the procedure used to generate and analyse MC events for background processes.

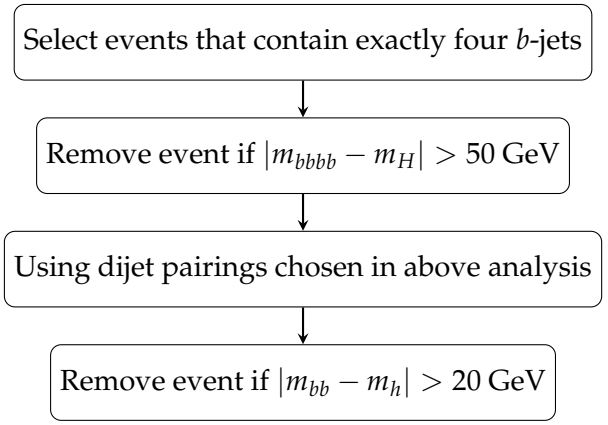


FIGURE 5.16: Event selection used to compute the signal-to-background rates.

It is clear from the data obtained that the QCD-induced $pp \to b\bar{b}b\bar{b}$ process is the dominant background channel[6], followed by $pp \to Zb\bar{b}$ and $pp \to t\bar{t}$. The next step is then to calculate the event rates in order to get the significances for two values of (integrated) luminosity, e.g., $\mathcal{L} = 140$ and 300 fb^{-1}, corresponding to full Run 2 and 3 data samples, respectively. The event rate (N) for the various processes is given by:

$$N = \sigma \times \mathcal{L}. \tag{5.7}$$

After the event rates have been calculated, the significance can be evaluated, Σ, which is given by (as a function of signal S and respective background B rates)

$$\Sigma = \frac{N(S)}{\sqrt{N(B_{b\bar{b}b\bar{b}}) + N(B_{Zb\bar{b}}) + N(B_{t\bar{t}})}}. \tag{5.8}$$

It is then clear from Table 5.4 to Table 5.5 that the variable-R approach works better than fixed-R one also in providing the best significances, no matter the choices of R for the latter. The improvement in the final significances is indeed very significant. This should not be surprising, given the ability of the former in outperforming the latter

[6]In fact, this study computed the full four-jet sample produced by QCD, i.e., including all four-body partonic final states, yet, in presence of the described kinematical selections and b-tagging performances, the number of non-$b\bar{b}b\bar{b}$ events surviving is negligible [147, 142, 148].

Process	variable-R		$R = 0.4$	
	BP1	BP2	BP1	BP2
$pp \to H \to hh \to b\bar{b}b\bar{b}$	2.077×10^{-4}	8.962×10^{-3}	1.254×10^{-4}	3.210×10^{-3}
$pp \to t\bar{t}$	3.798×10^{-3}	2.131	1.651×10^{-3}	9.470×10^{-1}
$pp \to b\bar{b}b\bar{b}$	7.973×10^{-4}	2.850×10^{-2}	1.595×10^{-3}	2.217×10^{-2}
$pp \to Zb\bar{b}$	9.689×10^{-6}	2.627×10^{-2}	3.876×10^{-6}	9.695×10^{-3}

TABLE 5.3: Cross sections (in pb) of signal and background processes upon enforcing the reduced cuts plus the mass selection criteria $|m_{bbbb} - m_H| < 20$ GeV and $|m_{bb} - m_h| < 15$ GeV for the various jet reconstruction procedures.

	variable-R	$R = 0.4$
BP1	1.145	0.823
BP2	2.268	1.214

TABLE 5.4: Final Σ values calculated for signal and backgrounds for $\mathcal{L} = 140$ fb^{-1} upon enforcing the reduced cuts plus the mass selection criteria $|m_{bbbb} - m_H| < 20$ GeV and $|m_{bb} - m_h| < 15$ GeV for the various jet reconstruction procedures.

	variable-R	$R = 0.4$
BP1	1.676	1.205
BP2	3.320	1.777

TABLE 5.5: Final Σ values calculated for signal and backgrounds for $\mathcal{L} = 300$ fb^{-1} upon enforcing the reduced cuts plus the mass selection criteria $|m_{bbbb} - m_H| < 20$ GeV and $|m_{bb} - m_h| < 15$ GeV for the various jet reconstruction procedures.

from the point of view of kinematics. Again, while the signal-to-background analysis has been performed for the anti-k_T algorithm, the same conclusions are reached for the CA case.

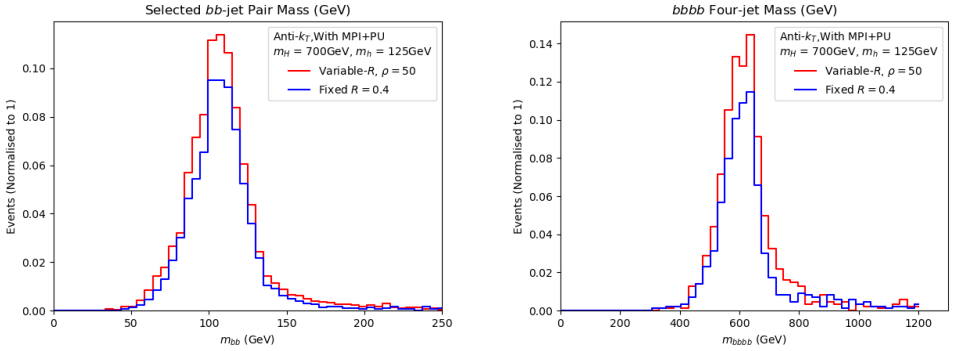


FIGURE 5.17: Mass peaks comparing variable-R and fixed R clustering, acting on simulation that includes MPI and pileup. Using the parameters of BP1. Left panel: b-dijet mass peak. Right panel: four b-jet mass peak.

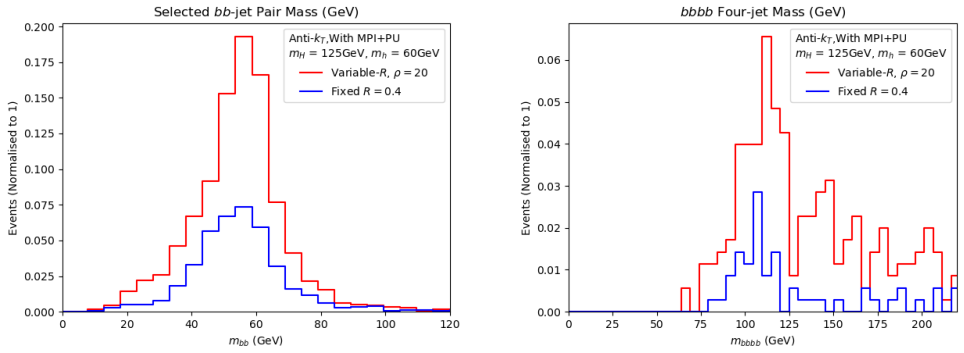


FIGURE 5.18: Mass peaks comparing variable-R and fixed R clustering, acting on simulation that includes MPI and pileup. Using the parameters of BP2. Left panel: b-dijet mass peak. Right panel: four b-jet mass peak.

5.3.4 Variable-R and Pile-Up

It has been noted that the nature of variable-R, combined with reduced p_T restrictions, allow for wider cone signal b-jets. A quick study of the impact of pile-up and multiple parton interactions (MPI), using variable-R, is offered next. As briefly mentioned, in order to perform such a study a more sophisticated detector simulation is required. `Delphes` is employed for this. The hadronised events (and pile-up, simulated in `Pythia8`) are passed through the CMS card (with the same b-tagging efficiencies and c/light-jet mistag rates as before). The same exercise is conducted with a fixed cone of $R = 0.4$ to compare.

In Figure 5.17 and Figure 5.18 the m_{bb} and m_{bbbb} spectra, as described earlier, are presented with pileup and MPI. This compares $R = 0.4$ and variable-R jet clustering. With the addition of pileup, in BP1, a different value of the ρ parameter are used; $\rho = 50$ GeV. Furthermore, no jet quality cuts are used here. The primary purpose of the jet quality cuts is to mitigate background, and as background is not considered here, the quality cuts are omitted in case the obscure the comparison. It is clear that, with pileup added, a variable-R algorithm retains significantly more events in its selection procedure.

As a final point, note that a further pile-up mitigation technique is possible in variable-R. This is in the values chosen for the $R_{\text{min/max}}$ variables. Clearly if, for some particular process, one discovers that using a variable-R sweeps in too much extra 'junk' into the jets, a simple reduction of R_{max} is always possible.

5.3.5 Other Variable-R Studies

Before concluding, a short review of other literature utilising a variable-R reconstruction procedure is offered.

While the leading b-jet has an R_{eff} roughly in line with expected values ($R_{\text{eff}} \simeq 0.5$), the lowest p_T b-jets have large cone sizes ($R_{\text{eff}} > 1.0$). These wide low p_T jets risk potential contamination from additional radiation. This effect is discussed in [149]. This study does not implement any vetoes to remedy this effect, yet, despite this, the results still suggest that the variable-R approach displays an improvement over traditional methods.

There have been other studies utilising variable-R methods for physics searches. For example, in highly boosted object tagging of $hh \to bb\bar{b}\bar{b}$ decays in [150]. Furthermore, in [149] mentioned above, a variable-R algorithm is deployed in the context of heavy particle decays. In both examples, an improvement over current fixed-R methods is present when using variable-R, which is in line with the findings of this study.

Finally, the relation of variable-R jet reconstruction in experiments to b-tagging performance has been considered. In particular, the studies of [151, 152] explore the possibility of Higgs to b-jet tagging at ATLAS using variable-R techniques. Specifically, these studies deal with boosted topologies, focusing on fat b-jet substructure, so the validity of applying these techniques in a non-boosted regime is to be determined.

5.4 Conclusions

This study has assessed the potential scope of the LHC experiments (from mainly a theoretical perspective) in accessing BSM Higgs signals induced by cascade decays of the 125 GeV SM-like Higgs state discovered in July 2012. It considers the following prototypical production and decay channel: $gg, q\bar{q} \to H \to hh$. Two benchmarks are considered, in BP1 H is a heavy BSM Higgs state and h is the SM-like Higgs state, while in BP2 H is the SM-like Higgs state and h is a lighter BSM Higgs state, with mass less than $m_H/2$. These mass choices induce resonant production and decay, thereby enhancing the overall rate. Any such Higgs boson, largely independently of the BSM construct hosting it, would decay to $b\bar{b}$ pairs, eventually leading to a four b-jet signature. The latter is extremely difficult to establish at the LHC, owing to the substantial hadronic background. Therefore, b-tagging techniques are to be exploited in order to make such a signal visible. However, this poses the problem that the latter are most efficient at large transverse momentum of the b-jets, say at least 20 GeV, which in turn corresponds to a significant loss of signal events if the BSM Higgs mass is in the sub-60 GeV range. Hence, if one intends to maximise sensitivity to this benchmark signature of BSM physics, a thorough reassessment of the current Run 2 approaches is mandated for and especially so in view of the upcoming Run 3.

With current p_T cuts on final state b-jets, using a fixed-R jet reconstruction and tagging procedure, will lead to a poor signal visibility. A majority of signal b-jets would be lost.

A variable-R approach shows a significant improvement in signal yield as well as signal-to-background rates. In final states of this kind, the signal b-jets have a wide range of p_T and hence varied spread of constituents. Using a fixed cone of a standard size ($R = 0.4$) constructs well higher p_T jets in an event but does not capture much of the wider angle radiation from lower p_T jets. This leads to two issues. Firstly, it will prove difficult to accurately construct m_h and m_H in the two- and four-jet invariant masses. Secondly, these jets will more often be lost due to kinematic cuts. A larger cone ($R = 0.8$), conversely, will gather up too much 'junk' in the higher p_T jets, which again will contaminate the signal. Hence a variable-R jet reconstruction algorithm offers a significant improvement for this search.

All of the above has been obtained in presence of a sophisticated MC event simulation. This employs exact scattering MEs, state-of-the-art parton shower, hadronisation and B-hadron decays as well as a simplified detector simulation. Given the results of this analysis, undertaking a more thorough detector level analysis is recommended. This ought to be done for a variety of different high b-jet multiplicity scenarios, to explore whether a shift to variable-R jet clustering, on the one hand, could be implemented and, on the other hand, would improve upon current signal significance limitations using fixed-R jet reconstruction. In fact, although this study is quantitatively based on the example of the 2HDM-II, this procedure can identically be used in other BSM

constructs featuring a range of (pseudo)scalar states emerging from decays of the SM-like Higgs state and in turn decaying into $b\bar{b}$ pairs.

Chapter 6

Existing Machine Learning in Jet Physics

ML emerges smoothly out of other disciplines, primarily statistics, and so there is not a strict boundary on what should be considered as ML [153]. Textbooks will frequently begin by looking at linear regression problems [153, 154], which would be well understood as a traditional statistics approach, before moving out of the domain of problems that could ever plausibly be tackled with a pen and paper. In this light, it is hard to say when ML was first used in particle physics, or even jet physics in particular.

Perhaps there is a strong claim for the iterative cone algorithm [116] to be the first example of ML for jet physics (see section 4.6.3.2 for a short description). It utilises a form of gradient decent, which is a common feature in ML algorithms. It was not referred to as ML by its creators, but the phrase ML was not in common use at that time. Subsequent sources have made the link between clustering algorithms and ML explicitly; *"A jet is defined by a clustering algorithm, which is an example of an unsupervised machine learning technique" – [155].*

For the rest of this chapter, that sticky question is put aside, and the focus is instead on ML in jet physics today. Here, a narrative review, covering only the most popular applications of ML, will be presented[1]. For a much more substantial review, see the living review of ML for particle physics [156].

6.1 Specific Challenges for ML

In order to properly discuss modern ML in jet physics, a couple of challenges that are particular to ML need to be outlined. These are the issue of overfitting, and working

[1] A full review of such a broad topic would be as long as a book, and that particular book would be out of date almost as soon at it had been written.

with "black box" analysis techniques.

Historically, vanishing gradient problems would also have appeared on this list, at least for Deep Neural Networks (DNNs), but these problems have been largely solved. The gradient descent used to train a DNN involves taking many partial derivatives in a row, and limits of precision would sometimes cause this gradient to appear exactly equal to zero, and hence vanish. With zero gradient, there was no way to progress the training of the DNN. Certain activation functions (such as leaky ReLU [157]), along with more modern compute power, have alleviated issues with vanishing gradients. A longer discussion of this challenge is offered in section 6.2.2.4.

Overfitting is a challenge that is not strictly limited to ML. When an algorithm is trained, it is normal to reserve a limited amount of data, perhaps 20%, to investigate its behaviour on data that has not been seen in training. The terminology for this is training data, and test data. If the algorithm performs significantly better on the training data compared to the test data, then the algorithm is described as overfitting the training data.

Essentially, this occurs because the data will contain information about the signal distribution, and also statistical noise. The information about the signal distribution is what the user hopes the algorithm will learn, and map to the correct outputs. This signal distribution will be common to both the test and train data. With enough parameters, however, the algorithm may become capable of identifying individual items in the training dataset, using unique variations that amount to noise. Being able to identify individual items in the training set allows the algorithm to classify them perfectly, however it is not generalisable. The noise characteristics that are used to identify each item have no pattern, and the same items will not be found in the test set, and so these associations only serve to confuse the algorithm when it encounters new data.

A common solution to this is to limit the complexity of the fit to align with the complexity of the signal. The details of this process are specific to the model being trained. To take the example of a Neural Network (NN), this can be done with weight decay, early stopping, or adding noise [158].

The second issue raised here is working with black-box tools. This is more subjective, and perhaps situational. The nature of the issue is that understanding the reasons for a decision made by an ML algorithm can be difficult. Standard algorithms normally take steps that are inspired by a human understanding of the problem. For example, a classical decision tree makes binary partitions of the data according to the scientist's understanding of what features best indicate the data's true classification. An example of this construction is given in Figure 6.1. This won't always give the right classification, but understanding the original intentions behind the decisions the algorithm makes may help predict problems, or resolve them when they arise.

By contrast, many machine learning algorithms operate on much looser assumptions. The algorithm is capable of recreating a great variety of functions, often referred to as the function space. It is assumed that there will be at least one close approximation to the signal function in that function space, and also, that the training process can locate one of these close approximations. If there is to be any understanding of why a particular function was selected from this function space, it must be found retrospectively. This is often complicated by how complex the chosen function may be; it will often contain literally thousands of parameters. There has been some success in unpacking meaning from such functions, this can be done by highlighting which parts of the input were most relevant to the decision [159], or by converting to a more comprehensible model [160].

Why though, should the user care why the training process picked a particular function? Is it enough to find a function that performs well on the test data? In physics, one might be interested in why a decision was made in the hope that understanding will provide physical intuition. In any field, there is the danger that occasionally, this ML tool may come unstuck[2]. A classic example of this, is that an image classifier may make classification decisions based on the image background rather than the subject of the image [161, 162]. Physicists might find this especially worrying, as given the complexity of the data, and the extensive modelling with complicated uncertainties, an error like this might not be obvious. It should be noted that not all ML tools are black-box processes.

6.2 Mini Review of Contemporary ML in Jet Physics

As mentioned above, [156] is a living review of ML in all particle physics. One point that is made clear by the contents of that review is that the bulk of ML applications relating to jets aim to classify. At the time of writing the review contains $\mathcal{O}(100)$ articles relating to classifying jets or events, as compared to 3 articles relating to jet formation. Track construction, which is closely tied to jet construction, is a popular ML topic, with $\mathcal{O}(50)$ entries.

The same focus on classification can be seen in [163, 164]. The series of publications [163, 164] discuss ML challenges, or bounties, on particle physics data. Those discussions emphasise the importance of open data, to provide 'standard candles' for comparing algorithms, which becomes increasingly more important with the complexity of the algorithms involved. Defining such challenges also illustrates the nuance of the questions being asked[3]. The methods to evaluating the proposed solutions needed to be highly detailed. The challenges often elicited solutions that were both ingenuous and simple.

[2] Occasionally, in a very amusing way; https://youtu.be/vppFvq2quQ0?t=280.

[3] In many ways getting results from ML is like asking a genie for wishes; often, you will get exactly what you asked for, and sometimes, learn that wasn't what you wanted.

The solutions proposed also reflects the more general shift from boosted decision trees (BTDs) to NNs.

The rise of machine learning techniques for classification in particular has opened some questions about the correct statistical treatment of the predictions. How can the statistical and systematic uncertainties of (an often unknown) function be estimated? These questions are addressed in [165]. That paper covers both the source and the means of determining uncertainties, along with a description of the best treatment of nuisance features.

6.2.1 Jet Formation

Agglomerative algorithms, such as generalised k_T (for a description see section 4.6.3.3), are now such a ubiquitous default that jet formation is often discussed as if there was no other possibility; *"As jets are constructed from* $2 \to 1$ *clustering algorithms"* – *[155]*. As seen in section 4.6 other established options do exist, and indeed recent examples of their use can be found [166], but the quote still illustrates the prevalence of agglomerative methods.

While there is not a great focus on jet formation using ML, some ideas have been put forward.

In the paper [167], the idea of an agglomerative clustering, where the order of merges has some probabilistic element was explored. These are dubbed Q-Jets. The study assumes that this probabilistic element is small enough that it normally results in approximately the same jets. Rather that focusing on which particle a jet belongs to, the objective is to create many alternative clustering orders[4]. The same jet with an altered clustering order will have the same 'raw' kinematics; however, as the study points out, jet pruning may have different outcomes. This allows a distribution to be generated for the jet's pruned mass, for example. The authors observe that this distribution gives them a means to estimate the true uncertainty. They also explore the idea of variables that could be constructed by comparing variations in the clustering order, to give an idea of the variability of the jet.

A different approach would be to focus on the uncertainly associated with assigning a particle to a jet. As mentioned in section 7.1, there are some fundamental reasons for feeling that jets might be more naturally described as fuzzy sets rather than discrete sets. In [168], this idea is properly developed and dubbed fuzzy jets. These are inspired

[4]The justification given for this is that a jet's clustering order is intended to be a reflection of the splitting order in the shower, which is perhaps a slightly dubious assertion. The splitting of the shower was the inspiration for the initial agglomerative k_T algorithms, but the extension to the full class of generalised k_T algorithms has a different motivation. An anti-k_T jet's clustering order does not, in practice, look much like a refection of the shower.

by Gaussian mixture models (see section 7.1.1.9). The approach used assumes a Euclidean metric between all particles; each particle contributes its rapidity linearly to the location of the Gaussian. This scheme proves to be good at recreating kinematic variables, and may also supply extra information in the form of the relationship between the learned standard deviation, σ, of the Gaussian and the jet mass.

An earlier exploration of the maximum likelihood principle to define jets can be found in [169]. Although this takes a somewhat different approach, there is overlap in the central premise, however, it does not offer any numeric analysis of the method.

A common point to all these suggestions is an interest in acknowledging the probabilistic nature of jets.

Arguably, some convolutional neural networks (CNNs) also create jets, but without a clustering process. By binning the hits in 2 or more dimensions, a whole event can be treated as a image to be processed. In particular, some CNNs will identify which pixels are considered Higgs signal [170]. This is closer to a classification activity on the pixels than a clustering actively on reconstructed particles. This is discussed in more depth in section 6.2.2.5.

6.2.2 Jet Classification

As mentioned in earlier sections, much of the focus on machine learning in jet physics has been on classification. Various assignments are made. A jet might be classified simply as signal or background. The quark that created the particles in the jet might be guessed, this is known as tagging, the jet is said to be tagged by the quark. The hard event (see section 4.1) itself might be directly classified. There are a myriad of excellent classification tools available from ML.

When considering ML classifiers [171] and [172] both emphasise the importance of simplicity, that is, reducing the number of free parameters in the model, particularly if there is to be training on simulated data. Despite the great accuracy with which events are simulated, simulation is complex and requires some approximation and tuning. Given these conditions, it's difficult to know how faithfully the finer details of the simulation will reproduce the physics. While many features are certainly well matched between simulated and observed data, it's not possible to be sure that every feature that could be extracted will be accurate in simulation, because the number of features that could be extracted is not finite. The more parameters a model has the greater the concern that it may use some obscure artefact of the simulation to improve its performance, this problem is similar to overfitting, as described in section 6.1. In particular, [172] makes the point that given sufficient data and parameters most machine learning models can achieve the same benchmarks on particle data, so the objective becomes the capacity to

meet the benchmark with the simplest model. Furthermore, semi-supervised or unsupervised training on real data is very advantageous where possible.

6.2.2.1 Features

Many approaches to classification use some so-called 'expert features'[5]. These frequently include kinematic features of tracks, jets and jet-vertices [174, 175, 176, 147]. A simple example would be the mass of a jet,

$$m_{\text{jet}} = \sqrt{\left(\sum_i E_i\right)^2 - \left(\sum_i p_{x_i}\right)^2 - \left(\sum_i p_{y_i}\right)^2 - \left(\sum_i p_{z_i}\right)^2}, \tag{6.1}$$

where i is a sum over the particles contained in the jet. A slightly more nuanced example might be the angle between the jet axis, defined by the direction of the sum of the momentum of all jet constituents, and the highest p_T track in the jet. The secondary vertex of a jet, where a particle inside the jet underwent further decay, is expected to be a particularly good discriminator because it gives a measure of the lifetime of the particle that created the jet.

Jet substructure variables, such as n-subjettiness and energy correlation functions, are more complex examples of expert features [177, 178]. They are also often used as inputs for clustering functions, although that is not their only application. It is possible to ascribe them with a clear physical interpretation, and so like the event shape variables discussed in section 4.6.2, they provide a higher level description of the events.

A comprehensive collection of jet substructure variables was constructed in [179], dubbed energy flow polynomials. They are a linear basis containing all IR-safe jet substructure variables. Some very successful classifiers have been built with these inputs [180].

Related to this is the Lund plane [181]. As the name suggests, the Lund plane is not a single variable, but a diagram that captures key information about jets. This diagram is a heatmap, populated by the series of merges from the root of the jet, to a particle, each time following the highest p_T subjet. That is, starting with the split at the root of the jet, the split is added to the heatmap, then following the highest p_T subjet (or pseudojet) find another split to add, and repeat until a single particle is reached. The variable on the x axis of this heatmap is $\ln(1/\Delta)$, where Δ is the angle between the subjets in the split, the variable on the y axis is $\ln(k_T)$, where k_T is the transverse momentum of the lower p_T subjet, relative to the higher p_T subjet. The colour of the heatmap indicates the local density. This heatmap offers a terse and interpretable description of a single jet, or a collection of jets. It is a popular input for classification tasks [182, 183].

Three arguments might be given in favour of using expert features;

[5] Sometimes also referred to as high-level features [173].

1. Physical intuition may be easier to obtain when designing a system based on variables with clear interpretation. This is sometimes cited as a motive for using expert features [180, 184].

2. Expert features typically have a higher information density when compared to other data representations [147]. Dense information allows the classifier to function with fewer parameters, which improves interpretation and reduces room for overfitting.

3. The accuracy of simulations, in particular parton showers, is checked for specific variables. This is done via 'tunes', which adjust various parameters of the simulation. Often these include jet shape variables [185]. So there are stronger guarantees on the accuracy of these variables, at least in the energy ranges used to create the tunes.

And perhaps two clear counter arguments;

1. Evidently any choice of expert feature will exclude some information. It is possible that the excluded information could improve performance. There are no guarantees however, as expert features are engineered to minimise noise, and anything using the data 'raw' must be able to exclude this noise by other means. Comparative analysis of methods that do and do not use expert features have been done on the same dataset [147], and the results are mixed.

2. Different features may be optimal for different signals. Devising features that carry the right information, and are not highly correlated is a challenging task, which may not be transferable to other problems [173, 184].

These advantages and disadvantages are to some extent common to other classification tasks. Both approaches are still taken, although there is currently a trend to move away from expert features. This can be seen in the differing approaches taken in an open kaggle contest in 2014 [186], which primarily involved using expert features for classification, to the more recent kaggle contest in [164], where classifiers were typically trained on lower level data.

6.2.2.2 Established Techniques

Prior to the meteoric rise of DNNs, the favourite technique for jet classification was Boosted Decision Trees (BDTs). They didn't actually come first, DNNs emerged first [186], but took longer to be favoured in particle physics. In part, they where originally favoured because BDTs offer much more transparency than NNs or DNNs.

The BDT is really the conjunction of two algorithms, one algorithm to construct trees, and a second 'boosting' algorithm which can improve any model. The first algorithm, to construct the decision tree, aims to construct a single decision tree, which could be used to classify a chosen data point. A illustration of this idea is shown in Figure 6.1. Various algorithms exist for constructing these trees. To name a few; the ID3 algo-

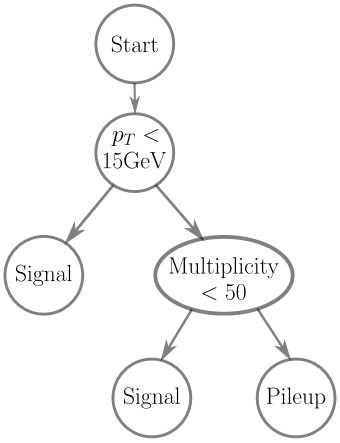


FIGURE 6.1: A short mock-up of what a decision tree for classifying jets as signal or pileup jets might look like.

rithm [187], the successor of the ID3 algorithm, the C4.5 algorithm [188][6], and Classification and Regression Trees (CART) [189, 190]. These algorithms can construct deep trees with many decisions. However, for the purposes of a BDT, only shallow trees are formed, also known as 'stumps'.

CART makes a good general example of this construction process; it is essentially a class of divisive recursive splitting algorithms. Steps to create the tree using a simple version of CART are as follows;

1. Start with a root node that contains all points.
2. While there is at least one node containing both a minimum number of points, and points from more than one category, split that node;
   (a) For each possible split, j, calculate the Gini impurity;

$$\text{Gini}(j) = 1 - \sum_i P(i|j)^2 \tag{6.2}$$

   where i is a class, and $P(i|j)$ is the relative frequency of i in j.
   (b) Chose the split with the lowest Gini impurity, use it to create two new nodes, each containing the points assigned by this split.

This creates a single decision tree, as in Figure 6.1. Then the boosting algorithm is used to improve the tree. Again, various boosting algorithms are available, but this

[6]Which itself has a successor, C5.0, however, that one is proprietary.

time there is one that is by far the most prevalent, AdaBoost [191]. AdaBoost works by repeating the creation of the classification tree, and assigning weights to the points that are proportional to how frequently the last tree misclassified that class. This very simple process focuses the next tree on the points that are hardest to classify.

BDTs have been used for a variety of classification tasks. During the open Higgs Machine Learning Challenge many contestants attempted BDTs [186]. The performance of BDTs and NNs in top-tagging has been compared, and found to be similar [192], a further study from the same authors repeated the comparison for a convolutional neural networks (CNNs), again, finding that the performance of the CNN and the BDT was similar [173]. Still hunting for Higgs particles, the study [193] used BDTs to identify whole events, when Higgs decays are accompanied by $t\bar{t}$ pairs. The study trained two BDTs, one for identifying events where both tops decay hadronically, the other for the semileptonic top decay. Using these BDTs, the total number of observed counts in various bins was calculated. Comparison with expectations confirmed the agreement with standard model predictions. A variation on BDTs, with a focus on systematic uncertainty, is presented in [194][7]. This is proposed as a method designed specifically for particle physics problems. In [170], a combination of a CNN and a BDT is used to locate and identify Higgs jets. The CNN is used to locate areas that appear to be candidate Higgs jets, the BDT is then used to discriminate between actual Higgs jets, and similar background jets. Despite the complexity of the CNN used, the BDT still improves the discrimination between the signal and the background.

When advocating NNs, a feature that is often emphasised is that they are universal approximators [195][8]. It is not quite so well known that BDTs are also universal approximators, for more or less the same reasons as NNs are [196], so in that respect they are on equal footing.

Despite having the same theoretical limits[9] , the actual performance of a BTD and a NN on the same dataset will not be equal. There is no universal trend as to which is the better choice, the performance is depended on the dataset [197, 198].

6.2.2.3 Neural Networks

Neural networks (NNs) are machine learning algorithms capable of classification or regression. These are not a recent conception, they were first written about in 1943, and

[7] The author of [194] also echos the observation that NNs and BDTs are the two favourite methods of classification in jet physics.

[8] Actually, the claim is that any Borel measurable function can be approximated by a NN with one hidden layer, so long as the hidden layer uses a squashing function as an activation function. This caveat explains why a NN cannot be expected to, say, predict digits of π, or predict prime numbers. Those things are not Borel measurable.

[9] These limits could be reached with an infinite computer, a device that has similar uses and limitations to a spherical cow.

have been rediscovered many times since [199]. There are a great variety of structures that go under the name NN, the common feature is that they are built of components known as neurons. These neurons take a vector of values as input and deterministically produce an output value. These neurons are inspired by the neurons found in the animal brain, and the NN is sometimes described as resembling the brain [200][10].

Neural networks (NNs), gained popularity once progress was made on the practical issues of training deeper networks. This issue is known as the vanishing gradient problem, and explaining it requires a brief overview of what a NN is, and how it is trained. The remainder of this chapter will explore NNs of increasing complexity. Starting with a basic linear feed forward network, and then looking at other popular architectures. At each stage, some notes on the applications in physics are given.

This will include a minimum of technical explanation. An excellent pedagogical introduction to NNs can be found in [154].

6.2.2.4 Linear Feed Forward Networks

The basic unit of a neural network is a neuron. A neuron is a function that takes multiple input values, makes a linear combination, applies a function known as the activation function, and outputs one value. In symbols this is

$$f(\vec{x}) = g\left((\sum_i w_i x_i) + b\right) \tag{6.3}$$

where $\vec{x}$ is a vector if input values, x_i, w_i is the weight associated with the ith input, b is a constant known as the bias of the neuron[11], and g is the activation function. The popular visualisation of this is shown in Figure 6.2.

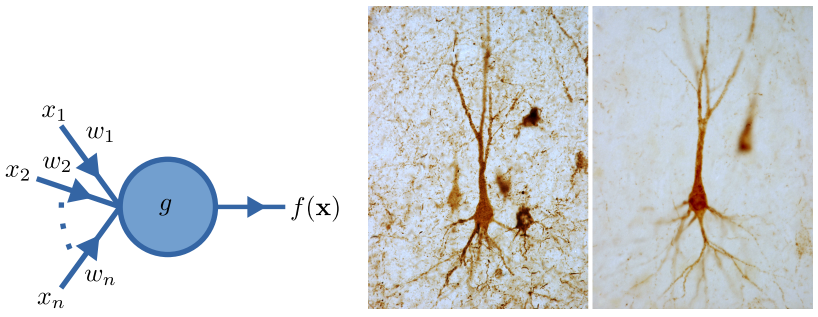


FIGURE 6.2: To the left; a single neuron, with activation function g, and inputs 1 to n. This visual representation of Equation 6.3 emphasises the inspiration of the biological neuron, which are shown in two photographs to the right. Photographs from [201].

[10] Or alternatively, they are described as the *"subject of exaggerated claims regarding their biological plausibility" – [199].*

[11] No relation to statistical bias.

Often the activation function will be chosen to be a leaky ReLU, function;

$$\mathrm{ReLU}_\epsilon(y) = \begin{cases} \epsilon y, & \text{if } y < 0 \\ y & \text{otherwise,} \end{cases} \tag{6.4}$$

where ϵ is a small constant. 'ReLU' is an abbreviation of Rectified Linear Unit. The reasons for this choice will be easier to cover when discussing training the NN.

For now, it is enough to notice that by subtracting the output of one neuron from another it is possible to create a step like function. To take a one dimensional example; let the neurons be labelled a and b. Chose their weights to be, $w_a = +1$ and $w_b = +1$, and their bias' be $b_a = 0$ and $b_b = -1$. The difference of f_a and f_b is then

$$f_a(x) - f_b(x) = \mathrm{ReLU}_\epsilon(x) - \mathrm{ReLU}_\epsilon(x-1) = \begin{cases} \epsilon & x < 0 \\ \epsilon + (1-\epsilon)x & 0 \leq x < 1 \\ 1 & \text{otherwise} \end{cases} \tag{6.5}$$

This is a step like function, which meets the requirements for a squashing function, as set out in [195]. Therefore, a linear superposition of these neurons is a universal approximator. A single linear superposition of neurons is often referred to as a layer, this is depicted in Figure 6.3.

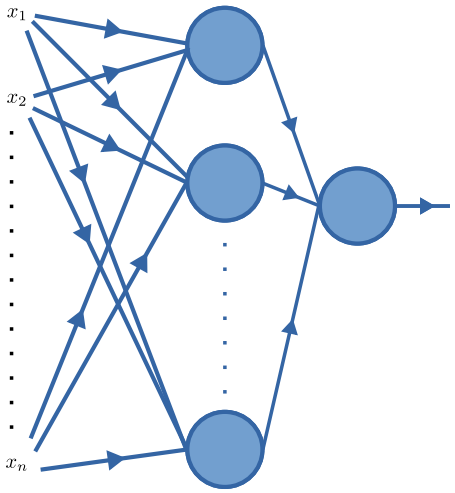


FIGURE 6.3: A layer of neurons linked together, with a single neuron creating a linear superposition of their outputs.

This layer is the most basic form of NN, and with sufficient neurons it can behave as any function. The trick is to find the best parameter values (the parameters being the weights, w_i and biases, b_i), to approximate the function needed. A loss function is needed, which measures the distance between the NN's current output and the ideal output. Once the loss function is defined, the NN can be 'trained' to better mimic the ideal output.

A simplistic version of training might proceed as follows;

1. The input data is normalised.
2. Make two sections of the data, a larger segment for training on, and a smaller section for testing against.
3. All weights and bias' are initialised from a random uniform distribution between 0 and 1.
4. It is now time to train the parameters. While the performance on the test data set tends to improve;
   (a) Take a random example from the training dataset, use the NN to get an output value.
   (b) Obtain the partial derivative of each parameter (weights and biases) with respect to the loss function. This is known as back propagation.
   (c) Using the partial derivates as guidance, make a small alteration to each parameter in the direction of reduced loss.

In the beginning, the performance on the test data should improve, despite it not being directly used in the training process. This occurs as the NN learns the signal function shape, which is common to the test and the train data. Often there is some averaging in evaluating the stopping condition. After some training, the performance on the test set will actually start to degrade. This is because the NN is now learning to replicate noise in the training data, and the shape of the noise differers between the test and the train data. The performance on the test set may randomly degrade on occasion, and this is not sufficient to stop training. If the performance on the test set gets worse on average over 10 steps, then the training should be abandoned. Other methods are possible.

There are many variations and improvements taken on this process. Most NNs contain more than one layer of neurons, this makes it possible to better approximate a function with fewer parameters. NNs with multiple layers are often described as deep (and therefore named Deep Neural Networks, DNNs). It is here that the problem of vanishing gradient can arise. With many layers of activation function, the gradients calculated in item 4b can become very small, even to the point of being obscured by computationally indistinguishable from zero. If a non-zero gradient cannot be found then the NN cannot be improved. The leaky ReLU activation function is less susceptible to this than the previously preferred sigmoid function, as the gradient of a single function never tends to zero. Using modern computational power also allows better precisions, and so smaller numbers remain distinct from zero.

For a more complete, practical approach to training NNs, see [154]. However, these steps capture the essence of training an NN.

In [202], comparisons are made between a fully connected, feed forward, NN, and more complex architectures. The aim is to differentiate between b, c and light jets. Changing the architecture does not lead to significantly different accuracies. The same study also looks at the impact of changing the input features. They try expert features, vertex features and track features. This does have an impact on accuracy, with expert features preforming best when considering only one category. However, supplying all 3 categories at once improves the performance still further.

A pair of NNs designed to distinguish b, c and light jets, named CSVv2 and DeepCSV [203]. Both were used at different points by the CMS experiment. The classifiers take expert features as input. It was found that increasing the number of layers, from CSVv2 with 1 hidden layer, to DeepCSV with 4 hidden layers, improved performance, although this was done alongside increasing the features available, so the conclusion is not straightforward. A replication study is available in Appendix A.

Some classifiers have more complex tasks; in [204] the same classifier is used to distinguish signal jets from background, and to identify which part of the signal process each jet corresponds to. The process considered is $t\bar{t} \rightarrow W^+bW^-\bar{b}$, for which may be as many as 180 permutations of the signal to consider. Various training schemes are compared, along with variations in the size of the NN. These are also compared to a maximum likelihood approach, and found to be a consistent improvement.

6.2.2.5 Convolutional Neural Networks

The principle behind Convolutional Neural Networks (CNNs) is that images contain patterns that should be recognised as the same at any location in the image. These patterns may exist on many levels. For example, at lower levels one might expect edges at different angles, above that, corners and curves, and above that eyes and faces[12]. At each level the features are identified if they activate the corresponding kernel matrix when that kernel matrix is applied to the pixels containing the feature; an image of this can be seen in Figure 6.4. Each kernel is applied to all points of the image, and so the pattern that the kernel is designed to identify will be recognised in any part of the image. Moving the kernel across the image is called convolving the kernel with the image. This process may happen several times with different kernels, generating multiple output features from the same image. Often this is accompanied with pooling the output features of nearby pixels.

Eventually, the final image should contain a very regular representation of the image. The pixels of this become inputs for a regular NN. This last stage converts the reduced feature map into a classification or regression.

[12] This visual hierarchy is the inspiration for the layers in a CNN. It has often been noted that the actual layers of a CNN do not always see anything that is possible to interpret [205].

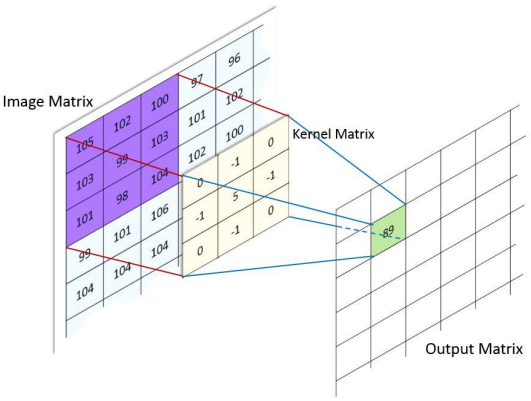


FIGURE 6.4: A CNN kernel acting on pixels in an image [206], The kernel is aligned with a set of pixels of the same size, and the corresponding entries are multiplied together and then the values are summed. This may happen many times over.

Preprocessing is important for good CNN performance. Image data is sparse, and often obeys many trivial symmetries. If preprocessing can remove the obvious symmetries, the CNN will not waste parameters learning these symmetries.

In the case of jet physics, data will take the form of a image with coordinates in rapidity (or possibly pseudorapidity) and ϕ, (see Figure 4.2 in section 4.5). The pixels of the image are often coloured according to energy deposits, possibly with a distinct colour for each subsystem of the detector. CNNs for this task can be split into two common categories: the first kind skips the jet clustering step, classifying with a whole event at a time; the second kind act after the jets have been formed, classifying one jet at a time. In both these challenges there are some key symmetries and some transformations that are not as safe as they might appear:

- Almost all detectors are symmetric in the ϕ coordinate (see Figure 4.2 in section 4.5). So events should be rotated in ϕ to line up at least one key feature (for example, the direction of greatest energy flow).
- Equally, the event should be symmetric in the x and y axes, so if a flip in a plane perpendicular to the beam can align a second moment of the energy, that should also be applied.
- For CMS, the positive and negative z axes (or rapidity directions) are identical. So events should be flipped in z axes such that all events have the majority of their energy flow in the same z direction.
- Often events are normalised in energy, or transverse momentum. Clearly these aspects will carry real information about the event, but it is actually not desirable to classify based on these criteria. This is because the results of the classification produced by the CNN will often be used to analyse a mass spectrum[13], and so

[13] "Bump hunting".

it is very important that the performance of the CNN is not correlated with the mass of the event or jet being classified. Both transverse momentum and energy are strongly correlated with mass.

- One imperfect symmetry is a rotation in the rapidity – ϕ plane. This is not a proper symmetry because it is not possible to rotate a jet in a way that preserves both the n-subjettiness and jet mass [155, p. 92]. So one or the other must be altered.

Early attempts to apply a CNN to the problem of jet tagging include ANN [207], a top tagger, an general flavour tagger [208], and two classifiers that identify boosted W bosons [209, 210]. All 3 attempts note an improved performance in comparison to expert feature techniques. Only the last one, [210], takes note of the complications involving rotations in rapidity – ϕ space. In some respects, [209] goes rather further than the others; that work investigates the influence that the MC showering algorithm responsible for generating the training samples has on the classifier, which is of particular interest when working with MC data. It is difficult to determine how accurate the representation of the MC data is, so by obtaining similar performance on different showering simulations, the confidence in the classifications of the tagger is increased. The CNN of [211] addressed the simulator dependence of previous studies and showed invariance between `Pythia8` and `Herwig`.

A different approach to this concern would be to train, in part or in full, on unlabelled, real data. This tactic has only been more recently explored for architectures as complex as CNNs; in [212] it was applied to distinguish Z + jet and dijet events. The study [213] also explores classifying on unlabelled data, but attempts to find examples of currently unknown particles. Both works demonstrate that classification on data is possible, even for models as complex as a CNN.

6.2.2.6 Recurrent Neural Networks

Recurrent Neural Networks (RNNs)[14] are a modification of the NN concept designed to take an input in the form of an ordered sequence of indeterminate length. They are popular for text processing because sentences, paragraphs and corpora normally do have a strict order and variable lengths.

In the most general sense, a RNN is any NN which is designed to be applied once to each step of a sequence, and takes both values from the data at the current step and the output of the NN from the previous step as an input. In practice, the most common configuration for an RNNs is the Long Short Term Memory (LSTM) network [215].

[14] RNNs are related to, but not the same as Recursive Neural Networks (RecNNs, or sometimes RvNNs), see [214, 184].

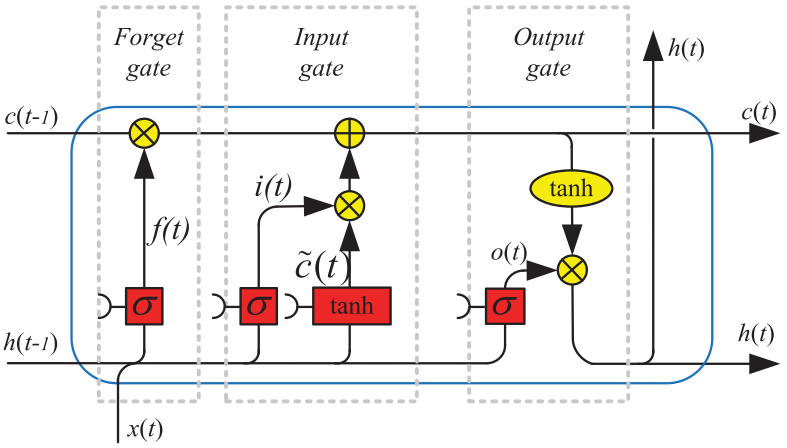


FIGURE 6.5: The internal structure of a LSTM [215]. Black lines represent a vector of data. When two lines come together, two vectors have been concatenated, when two lines diverge, the vectors have been copied. Red rectangles represent NN units, with the activation function named in the rectangle. The half moon out to the left of the NN units represents their bias. These NN units will transform the vector of data. The yellow circles represent point-wise operations, to add or multiply all elements of the incoming vectors.

The core concept of an LSTM is that there should be three inputs to each step, the first is from the new data in that step, the second is the output from the previous step (short term memory), and the third represents information distilled over many previous steps (long term memory). Three components of this algorithm can be identified [215];

1. A forget gate, to erase some of the long term memory;
   (a) Concatenate a vector of input from data at step t, $x(t)$, with output from the last step $h(t-1)$.
   (b) Pass this through a NN with a sigmoid activation function, which will generate a vector $f(t)$ which indicates how much of the long term memory should be removed.
2. An input gate, to update the long term memory;
   (a) Concatenate a vector of input from data at step t, $x(t)$, with output from the last step $h(t-1)$.
   (b) Pass this through two parallel NNs; one with a tanh activation function, which will generate new information, $\tilde{c}(t)$, to be stored in long term memory, and the other with a sigmoid activation function, which will generate a vector $i(t)$ which indicates what parts of the new information should be retained.
   (c) Using $f(t)$, $\tilde{c}(t)$ and $i(t)$, the previous long term memory $c(t-1)$ can be updated;

$$c(t) = f(t)c(t-1) + i(t)\tilde{c}(t) \tag{6.6}$$

3. An output gate, to generate a classification or regression for this step, and the short term memory for the next step, $h(t)$;
   (a) Concatenate a vector of input from data at step t, $x(t)$, with output from the last step $h(t-1)$.
   (b) Pass this through a NN with a sigmoid activation function, which will generate a vector $o(t)$ which represents the information of this step.
   (c) Generate the output, $h(t)$ using $o(t)$ and $c(t)$;

$$h(t) = o(t)\tanh(c(t)) \tag{6.7}$$

The LSTM can either be used generatively, by looking at $h(t)$ for each step, or one value can be ascribed to the whole sequence by looking at $c(t)$ at the end of the sequence. An alternative depiction of this can be seen in Figure 6.5. While many other varieties of RNN exist, the LSTM is generally very successful.

LSTMs have been applied as taggers; in [216] an LSTM as used for top tagging. Each track creates three numerical inputs; (p_T, η, ϕ). Finding a natural order in which to parse the tracks in the RNN can be challenging. In [216], the tracks are ordered using the anti-k_T clustering order. If the clustering order is considered as a binary tree, the leaves of the tree would be the inputs that need to be ordered for the RNN. Each node of that tree has a join distance, which is the separation between the two nodes that were merged to form it. The leaves are sorted by performing a depth first search, which prioritises nodes with smaller join distances. In the preprocessing steps, care is taken to preserve jet mass using rotation, in line with the challenges discussed with respect to CNNs. The study found that LSTMs operating in this way could outperform NN's working on expert features.

The more challenging task of tagging jets from strange quarks has also been considered as an LSTM task in [217]. The tracks are ordered by simple kinematic variables, such as energy, or distance from the jet axis. This is attempted with data that contains different levels of realism, and the impact of considering the challenges of the real experimental environment is illustrated.

6.2.2.7 More Complex Network Structures

There are many other possible ways to connect together neurons, or whole NNs, train them, and extract estimates from them. In [184] and [218], a variant of an RNN, known as a Recursive Neural Network was chosen for its tree like structure that mimics that of jet clustering.

A great deal of work has also been carried out on NNs in the form of sets or graphs, for example [219] and [220]. As the field is explored, more specialised configurations,

which reflect the underlying physical structures seem likely to gain popularity. At present, there is no clear winner, and more work will be required for any one method to see wide adoption.

Chapter 7

Clustering Methods and Spectral Clustering

An elegant and fascinating method of clustering points is known as spectral clustering. The word 'spectral' is in reference to the eigen*spectrum*, as spectral clustering's key component is an eigenvalue calculation. This chapter will set the scene by exploring the wider world of clustering algorithms before providing an in-depth review of the Machine Learning (ML) technique of spectral clustering. Following this, subsequent chapters will draw on this chapter to describe the specialised algorithm that was adopted for jet physics, and the results this algorithm can generate.

7.1 Clustering in ML

Clustering can be seen as an unsupervised form of classification [221]. In most cases, clustering is performed based on the assumption that the data is drawn from an unknown set of groups, and these groups are the ground truth. The role of the clustering algorithm is to recreate these unknown, ground truth groups. In other cases, the objects are grouped based on a subjective measure of similarity, with the aim of creating "interesting" clusters[1]. Arguably, jet formation walks the line between these two situations; on the one hand, there is a ground truth objective to create jets that match the momentum of the partons. On the other hand objects (particles in this case) may not originate from one parton exclusively, rather, they have often been generated by interactions between the output of the partons. Finally, there is some irreducible uncertainty in the momentum of the partons being reproduced. This materialises in the width of the Breit-Wigner distribution. So while the problem does possess a ground truth, it may only have imperfect answers, and while one complete solution can be

[1] One might claim that is isn't necessary for a clustering to be subjective to be interesting, but that would be subjective.

declared as better or worse than another, it is not possible to give the best allocation for a single particle. Given this, fuzzy set theory might arguably be a better fit for jet formation [222], however, a hard partition, also known as crisp clustering [223], is normally demanded for jets. The reasoning for this may in part be that old habits die hard; the first jet formation algorithms considered only 2 back to back jets [115], and in that setup, two exclusive jets is natural. Jets as fuzzy sets might also mean reconsidering many other factors, from background subtraction to comparison to predictions. There are a few studies that break this trend, in [168], a method of constructing jets using maximum likelihood is applied, which is dubbed fuzzy jets. This thesis will retain the concept of a jet being a discrete set.

There is no one agreed upon taxonomy of clustering algorithms, but a distinction that is normally considered of great importance is the difference between a partition based or a hierarchical clustering algorithm [221, 224, 225]. A partition based algorithm finds some means to allocate points to clusters, which does not depend on forming internal sub-clusters. By contrast, a hierarchical algorithm places the points at the leaves of a hierarchy which takes the form of a simple graph (that is, it has no loops), known as a dendrogram. Each level of the hierarchy dictates a set of clusters, each of which may contain smaller clusters. There are two ways to generate this hierarchy: the hierarchy can be generated divisively, beginning with all points in one cluster and repeatedly splitting; alternatively, the clustering may proceed agglomeratively, each point starting in a separate cluster, the clusters being progressively joined.

If this taxonomy was represented in a list it would have the form;

- Partition based.
- Hierarchy based.
  - Divisive.
  - Agglomerative.

While various other taxonomies exist [223], this one is most relevant to the work of this thesis, so no others will be considered[2].

Within the hierarchical clustering methods, the way in which similarity is measured is also key to the algorithm's behaviour. There are three fundamental categories [226];

1. **Single-linkage clustering**; also known as the connectedness, the minimum method or the nearest neighbour method. The distance between two clusters is considered to be the distance between the pair of elements, one from each cluster, which are closest.

[2] And besides, this direction has a real danger of leading to a taxonomy of taxonomies.

2. **Complete-linkage clustering**; also known as the diameter, the maximum method or the furthest neighbour method. The distance between two clusters is considered to be the distance between the pair of elements, one from each cluster, which are furthest apart.

3. **Average-linkage clustering**; also known as minimum variance method. The distance between two clusters is considered to be the average distance between each pair of elements, one from each cluster.

These three categories may all be applied with either divisive or agglomerative hierarchical methods. There are two common further options. The first is known as exponential-linkage [227], in which all three linkage options are available, and weighted against each other using a learned parameter. The second is known as Ward hierarchical clustering, and will be described further in section 7.1.1.4.

7.1.1 Algorithms and Characteristics of Common Clustering Methods

In Figure 7.1 the grouping formed by 10 common clustering procedures are displayed. A brief description of each of these methods, besides spectral clustering, will now be given, along with their place in the taxonomy. This provides an overview of the alternatives that might be considered for this task, and gives a stronger context in which to justify the choice of spectral clustering. Spectral clustering is then discussed in depth, starting from section 7.2.

7.1.1.1 Minibatch KMeans

This algorithm is a speed optimisation of the traditional K-Means algorithm. In the K-Means algorithm the objective is to minimise the cost function [226]

$$J = \sum_j \sum_i \left\| x_i^{(j)} - c_j \right\|^2 \tag{7.1}$$

where c_j is the centre of the jth cluster, $x_i^{(j)}$ is the location of the ith element in the jth cluster and $\|\bullet\|$ indicates the distance measure of choice. The convention is to use $\|\bullet\| = \|\bullet\|_2$, in which case the cost function is proportional to the variance of the cluster.

Normally K-Means is achieved by the following steps:

1. Chose locations for the c_j at random. This is often done by selecting a random set of k points, known as seeds, and setting the c_j to their location.

2. Assign each element to the closest c_j.

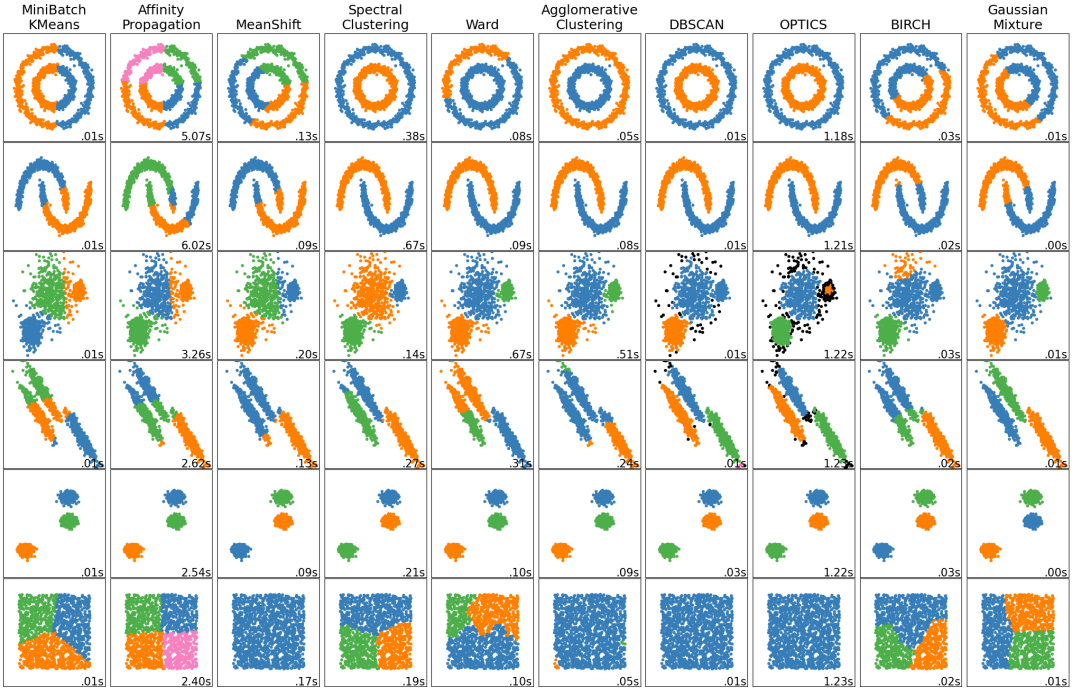


FIGURE 7.1: Comparison between 10 common cluster formation methods. Calculated and plotted by `scikit-learn` [228]. Each of these algorithms uses input parameters, which may not be optimised for the data given, results are only illustrative of potential behaviour.

3. Recalculate c_j to be the average location of all points assigned.
4. If any of the c_j have moved return to item 2.

This cost function is not convex, and may produce a slightly different result each time the algorithm is carried out, due to the random initial placement of c_j in item 1. Sometimes it is run multiple times with different seeds to increase the odds of locating the global minimum.

Conceptually, the update step is very similar to the iterative cone algorithms described in section 4.6.3.2. The distinction being that iterative cone algorithms assign particles to clusters in a greedy manner, and use a fixed cluster area.

Minibatch KMeans is a modification to optimise this algorithm for large datasets [229]. It batches the data, and converts the update step to a gradient descent step

1. Chose locations for the c_j at random. Often done by selecting a random set of k points, known as seeds, and setting the c_j to their location.
2. Create a vector, v_j to keep track of how many points have been used to move each c_j. Initially all $v_j = 0$
3. Chose a batch of b points, x_i, from the full dataset.

4. For each x_i;
   (a) Find the nearest centroid, c_j.
   (b) Update the number of moves for that centroid; $v_j \leftarrow v_j + 1$
   (c) Calculate the learning rate for this move; $\eta = \frac{1}{v_j}$
   (d) Move the centroid; $c_j \leftarrow (1 - \eta)c_j + \eta x_j$
5. Return to item 3 for another batch, until sufficient batches have been processed, or another stopping condition is reached.

Both variants of the KMeans algorithm are examples of partition based clustering. The iterative cone algorithm (section 4.6.3.2) is also a partition based clustering. While KMeans creates a predetermined number of clusters, the iterative cone algorithm does not.

7.1.1.2 Affinity Propagation

A means of clustering that aims to chose a small number of 'exemplar' points, then associate all remaining points to exactly one exemplar point [230]. This views the points as nodes of a graph, which must pass messages to each other with magnitude proportionate to their similarity. Between each two nodes a similarity is defined;

$$s_{i,k} = -\|x_i - x_k\|^2, \tag{7.2}$$

where $\|\bullet\|$ is the distance measure of choice. The similarity of a particle to itself, $s_{k,k}$, should reflect the initial degree of belief that particle k should be an exemplar. This can be set to a constant if this prior knowledge is not available.

The messages that are passed fall into two categories: responsibility of i to k, $r_{i,k}$, indicates how well suited k is to serve as exemplar to i, taking into account other potential exemplars for i; availability of i to k, $a_{i,k}$, indicates how well suited k is to serve as exemplar to i, taking into account the support from other points for using k as and exemplar.

An algorithm for this might go like;

1. All availabilities start as zero; $a_{i,k} = 0$.
2. Responsibilities are computed from $r_{i,k} \leftarrow s_{i,k} - max_{k' \neq k}\left(a_{i,k'}, +s_{i,k'}\right)$. Following this, $r_{k,k}$ is called the self responsibility of the point and its magnitude contributes to the belief that this point should be an exemplar at this stage of the algorithm.
3. Availabilities are now updates using $a_{i,k} \leftarrow \min\left(0, r(k,k) + \sum_{i' \notin i,k} \max\left[0, r(i'k)\right]\right)$. Likewise, $a_{k,k}$ is called the self availability of the point and its magnitude contributes to the belief that this point should be an exemplar at this stage of the algorithm.

4. Unless a stopping condition has been reached, return to item 2. Stopping conditions might be change falling below a chosen threshold, no alteration of the indicated exemplars for a determined number of iterations, or simply a fixed number of iterations.

Despite the special status of the exemplar, this algorithm is still divisive rather than hierarchical. It does not scale well with number of points.

7.1.1.3 Mean Shift

If the distribution of points is viewed as a landscape of variable density, there will be a finite number of peaks in density. The objective of mean shift clustering is to assign each point to the peak that its local gradient vector converges towards [231]. Such a landscape is depicted in Figure 7.2. This goal can be accomplished without directly computing the density.

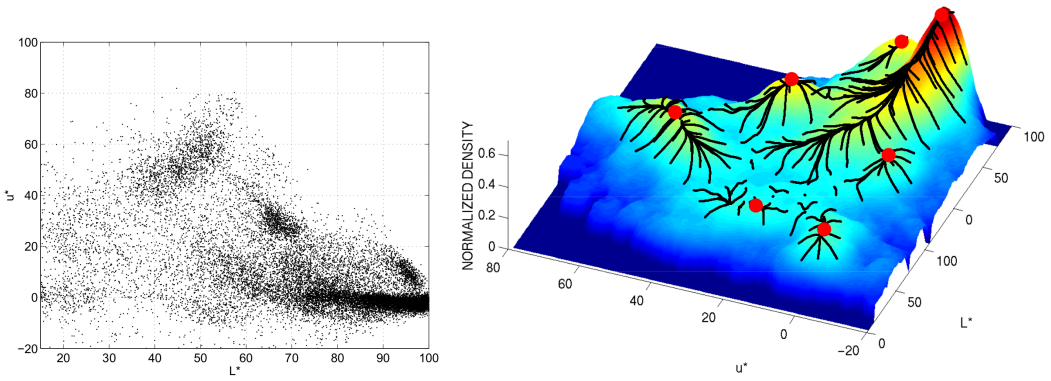


FIGURE 7.2: On the left is a set of points, and to the right is the height map for the density of those points [231]. Black lines on the height map indicate local gradient vectors, and the red dots indicate the points at which the gradient vectors converge.

A general, multivariate, kernel density estimator can be written;

$$\hat{f}_{\tilde{\mathrm{H}},K}(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\tilde{\mathrm{H}}}(\vec{x} - \vec{x}_i) \tag{7.3}$$

where $\vec{x}$ is a d dimensional vector representing the location of evaluation of the kernel, $\vec{\mathrm{H}}$ is a $d \times d$ bandwidth parameter matrix and $\vec{x}_i$ is the location of the ith data point. If the kernel is restricted to being spherically symmetric, and the bandwidth is restricted to a single parameter, h, this can be rewritten as

$$\hat{f}_{\tilde{\mathrm{H}},K}(\vec{x}) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{\vec{x} - \vec{x}_i}{h}\right\|^2\right). \tag{7.4}$$

The clustering will depend not in the density of the kernel itself, but on the gradient of that density. Provided k is differentiable, this may be computed directly [231];

$$\nabla \hat{f}_{\tilde{\mathrm{H}},K}(\vec{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\vec{x} - \vec{x}_i) k' \left(\left\| \frac{\vec{x} - \vec{x}_i}{h} \right\|^2 \right). \tag{7.5}$$

Now for notational simplicity, let $g(x) = -k'(x)$. This gives a new kernel which can be used to find the local gradient;

$$G(\vec{x}) = c_{g,d}(-f'(\|\vec{x}\|^2)) = c_{g,d} g(\|\vec{x}\|^2) \tag{7.6}$$

Now what is desired is to know which direction this gradient tends in, from the current location of the kernel, $\vec{x}$. This direction is the equivalent to the average value of g on the local points. It is desirable to normalised this with the magnitude of the local gradient, such that larger steps are taken in areas with low gradient (far from a peak), and smaller steps are taken in areas of high gradient (near a peak). So the iterative step becomes;

$$\vec{y}_{s+1} = \frac{\sum_i \vec{x}_i g\left(\left\| \frac{\vec{y}_s - \vec{x}_i}{h} \right\| \right)}{\sum_i g\left(\left\| \frac{\vec{y}_s - \vec{x}_i}{h} \right\| \right)}. \tag{7.7}$$

From this, a clustering procedure may be written;

1. For each point $\vec{x}_i$, find a peak location by;
   (a) Start with $\vec{y}_{s=0} = \vec{x}_i$.
   (b) Iterate with Equation 7.7, until $\|\vec{y}_s - \vec{y}_{s+1}\| < \epsilon$, where ϵ is some tolerance.
   (c) Return a peak location of $\vec{y}_{s+1}$.
2. Identify all peaks within 2ϵ as being the same peak, and put the corresponding points into the same cluster.

In some cases, for example, if K is the Epanechnikov kernel, Equation 7.7 converges in a finite number of steps. In that case ϵ may be infinitely small.

This is another divisive clustering algorithm.

7.1.1.4 Ward Hierarchical Clustering

It has been highlighted in [232] that the "Ward hierarchical clustering" method has acquired several inequivalent meanings. In all common interpretations, the morphology

of the dendrogram produced is the same, so with an appropriate stopping criteria, all variations are equal.

The objective of the original version of the algorithm was to minimise the error of the sum of the squares [233];

$$\mathrm{ESS}_{\mathrm{group}} = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2. \tag{7.8}$$

This error represents some loss of information that arises from combining points into clusters.

Ward is an agglomerative algorithm, which greedily combines clusters to minimise the ESS. An algorithm for this would be;

1. Begin by placing each point in its own cluster.
2. For each pair of clusters, calculate the EES that would be the result of combining that pair, according to Equation 7.8.
3. Combine the two clusters that result in the smallest EES.
4. Check the stopping condition, if the stopping condition is not reached, return to item 2.

Ward hierarchical clustering differs from single, complete and average linking in that is tends to produce the most even cluster sizes. However, a strong drawback, particularly for physics, is that ward clustering cannot operate with non-Euclidean metrics.

7.1.1.5 Agglomerative Clustering

As has already been alluded too, agglomerative clustering is actually a relatively broad category. The particular form of agglomerative clustering used in Figure 7.1 is average linkage, with a city-block distance. The stopping condition was the number of clusters.

For completeness the algorithm for this is given here;

1. Begin by placing each point in its own cluster.
2. For each pair of clusters, c_i, c_j, calculate the mean city-block distance between all possible pairs of points;

$$d(c_i, c_j) = \frac{1}{n_i n_j} \sum_i \sum_j \left\| \vec{x}_i - \vec{x}_j \right\|_1, \tag{7.9}$$

where $\|\bullet\|_1$ is the l1 norm, or the city-block distance and n_i (n_j) is the number of particles in c_i (c_j).

3. Combine the two clusters that result in the smallest $d(c_i, c_j)$.
4. Check the stopping condition, if the stopping condition is not reached, return to item 2.

This algorithm does work with non-Euclidean distances. It is not quite equivalent to the generalised k_T algorithm, section 4.6.3.3, due to the recombination step in generalised k_T algorithms, but there are distinct similarities.

7.1.1.6 DBSCAN

The premise of DBSCAN is that clusters have a minimum density, and should the density of points drop below this level, the edge of the cluster has been reached [234, 235].

Some definitions will make the algorithm neater. Let $N_\epsilon(x_i)$ be the list of all the points within range ϵ of x_i and $|N_\epsilon(x_i)|$ be the number of points within range ϵ of x_i.

Algorithmically, this can be written as;

1. Chose a minimum cluster density, $d_{\min}$. For a given number of points this defines a minimum number of points in a circle radius ϵ; $n_{\min}$.
2. Chose a point that has not been labelled noise, or assigned to a cluster, x_i.
3. If the point does not meet density criteria, $|N_\epsilon(x_i)| \geq n_{\min}$, label the x_i noise and return to item 2.
4. Start a new empty cluster, c_j.
5. Find $N_\epsilon(x_i)$ (all points within distance ϵ of x_i, including x_i itself), and start a stack from these points.
6. While there are still points in the stack;
   (a) Remove a point from the stack, call it x_s.
   (b) If x_s is already in a cluster return to item 6a.
   (c) Add x_s to the cluster created in item 4, c_j.
   (d) If x_s passes the density criteria, $|N_\epsilon(x_s)| \geq n_{\min}$, add all points within ϵ to the stack.
7. If points outside clusters, which are not noise remain, return to item 2.

This algorithm is fully deterministic, provided the clusters are unordered. It is a divisive algorithm and it has the unusual property of labelling noise as a separate category.

7.1.1.7 OPTICS

OPTICS bares many similarities to DBSCAN (section 7.1.1.6), but rather than having a fixed minimum density, it orders points by the density they require to be captured by a previous point [225]. Each item in the list defined by this ordering is associated with the minimum density required to make it reachable from another point in a DBSCAN algorithm. This ordered list contains sufficient information to reproduce the clusterings of DBSCAN, but also can be used to identify and produce clusters of varying minimum density, while still discarding noise.

Before defining the algorithm, defining the rules determining the core distance of a point will keep the algorithm legible. As previously, let $N_\epsilon(x_i)$ be the list of all the points within range ϵ of x_i and $|N_\epsilon(x_i)|$ be the number of points within range ϵ of x_i. Given a minimum number of points, $n_{\min}$, and a maximum core distance $\epsilon_{\max}$, the core distance for point x_i is;

$$\mathcal{C}(x_i, n_{\min}, \epsilon_{\max}) = \min_\epsilon(|N_\epsilon(x_i)| \geq n_{\min}) \begin{cases} \epsilon \leq \epsilon_{\max} & \epsilon \\ \epsilon > \epsilon_{\max} & \text{undefined} \end{cases} \tag{7.10}$$

So either the core distance is the minimum ϵ containing $n_{\min}$ points, or if that ϵ would be greater than $\epsilon_{\max}$ the core distance is undefined. One further idea needed is a reachability distance of x_i form x_j; this is the minimum ϵ needed from any other point for that point to capture this point, and that point to meet the $n_{\min}$ requirement;

$$\mathcal{R}(x_i, x_j, n_{\min}, \epsilon_{\max}) = \max\left(\mathcal{C}(x_j, n_{\min}, \epsilon_{\max}), \left\|x_i - x_j\right\|\right) \tag{7.11}$$

This is only defined if $\mathcal{C}(x_j, n_{\min}, \epsilon_{\max})$ is not undefined.

The first part of the algorithm is to construct a list of points, which will later inform clustering;

1. Chose a minimum number of points for an object's core distance $n_{\min}$.
2. Optionally, chose a maximum core distance $\epsilon_{\max}$; if not desired, this can simply be considered $\epsilon_{\max} = \infty$, this will increase the run time needed.
3. Create an empty ordered list; this will eventually contain all points and be used to identify clusters.
4. Chose a point that has not yet been added to the list, x_i.
5. Create a temporary stack, which will is used to sort more points, so they can be added to the list. This stack will be ordered by the smallest reachability distance, $\mathcal{R}$ found for each point in the stack.

6. Find $\mathcal{C}(x_i, n_{\min}, \epsilon_{\max})$ according to Equation 7.10.
7. Place x_i in the ordered list. Its reachability will be calculated later.
8. If its core distance $\mathcal{C}(x_i, n_{\min}, \epsilon_{\max})$ is undefined return to item 4.
9. For each x_n in $N_\epsilon(x_i)$;
   (a) If x_n is already in the list go on to the next x_n.
   (b) Add x_n to the stack if it is not already there.
   (c) Find the reachability distance with respect to x_i, $\mathcal{R}(x_n, x_i, n_{\min}, \epsilon_{\max})$ as in Equation 7.11. If this is less than the previous reachability distance update x_n's smallest reachability distance in the stack (and the list) and resort the stack.
10. Take the first item from the stack, x_s.
11. Add x_s to the list, along with its smallest reachability so far.
12. If the core distance of x_s, $\mathcal{C}(x_s, n_{\min}, \epsilon_{\max})$ is not undefined then find its neighbours $N_\epsilon(x_s)$, and for each x_n in $N_\epsilon(x_s)$;
   (a) If x_n is already in the list go on to the next x_n.
   (b) Add x_n to the stack if it is not already there.
   (c) Find the reachability distance with respect to x_s, $\mathcal{R}(x_n, x_s, n_{\min}, \epsilon_{\max})$ as in Equation 7.11. If this is less than the previous reachability distance update x_n's smallest reachability distance in the stack (and the list) and resort the stack.
13. If the stack is not empty return to item 10, else return to item 4.

This should yield a list of all points that are reachable from any other point, along with their smallest reachability distance from any other object. This list plots as a series of troughs as can be seen in Figure 7.3.

Now the challenge is to use this list to form clusters. If all the clusters can be restricted to the same density it sufficient to simply cut the list of points into segments every time the value of the reachability distance rises above a chosen constant. This is equivalent to DBSCAN. However OPTICS offers the possibility of variable sized clusters by considering any trough in the list which contains at least $n_{\min}$ point. Both alternatives are depicted in Figure 7.3.

OPTICS is a complex algorithm, but is is only mildly more computationally expensive than DBSCAN [225]. It is essentially a divisive algorithm, although as troughs may be found within troughs, it has the capacity to produce some hierarchical structure.

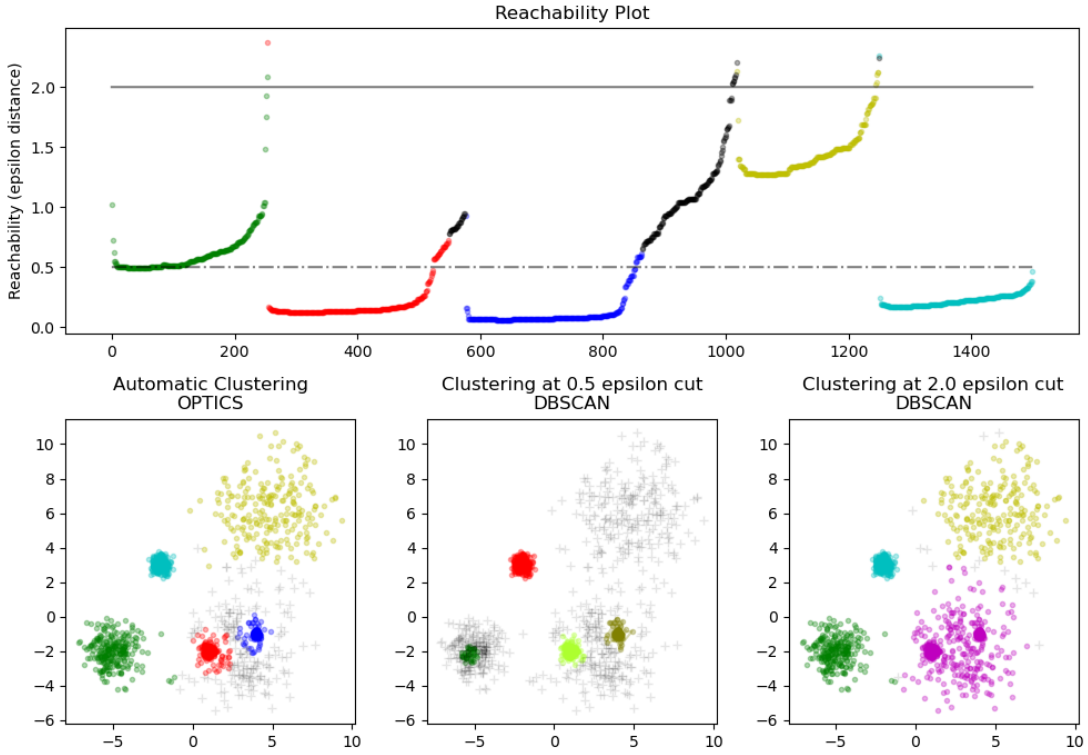


FIGURE 7.3: At the top is the list created by the OPTICS, showing troughs for each cluster. Below the various clusters that can be formed from this plot using the OPTICS algorithm or DBSCAN are shown. Plot created by `scikit-learn` [228].

7.1.1.8 BIRCH

The Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is an algorithm that boasts excellent use of both compute time and memory for large datasets [236]. This algorithm uses an initial pass over the data to form an acceptable clustering, referred to as a CF tree, which can be further refined with additional passes if time allows. At no point do all the data points need to be loaded into memory, and the algorithm is reported to scale well compared to its competitors.

The comparisons to competitor algorithms in the original paper are based on somewhat pessimistic assumptions about the performance of the competitors. Such as the claim that *"the best time complexity of a practical HC [hierarcical clustering] algorithm is* $\mathcal{O}(N^2)$*"* –*[236]*, where as the generalised k_T algorithm is now famously $N \log N$ [237][3].

In this context, the abbreviation 'CF' stands for Clustering Feature. These CF are properties which can be used to describe any group of two or more points. Every node of the dendrogram will be described by the CF properties of all points beneath that node. The CF properties chosen by the original authors of the algorithm are [236];

[3] In defence of the authors of [236] the very title of [237] makes it clear that the efficiency of generalised k_T was not widely understood.

- The number of points beneath the node.
- The linear sum of the vectors representing points beneath the node.
- The sum of the squares of the vectors representing the points beneath the node.

Whatever set of features are chosen as the CF, two attributes are important: they must be cumulative, when nodes combine, the new nodes CF should be a simple sum of the combining nodes' CF; and they must be all the information needed to calculate the distance between two nodes. So there would be many possibilities for the CF, and the best choice would depend on the distance metric used.

The CF tree itself has three parameters;

- **A branching factor,** B; each non-leaf node of the tree can have between 1 and B children. This factor should be set such that all the immediate children of a node can be loaded into memory.
- **A leaf branching factor,** L; each leaf node of the tree can have between 1 and L points assigned to it. This factor should be set such that all the points of a leaf node can be loaded into memory.
- **A threshold,** T; the maximum distance between a point and its assigned leaf-node. The larger T the smaller the tree is, as each leaf node represents a subcluster, and the size of T dictates the size of the subcluster.

The tree is depicted in Figure 7.4. Each node of the tree has CF properties which can be calculated using only the CF properties of its direct children. This is due to the cumulative nature of CF properties.

The steps to create this tree are as follows.

1. The tree begins by creating one root node which has one leaf node containing the first point in the data set.
2. Chose a point that has not yet been added to the tree.
3. Starting from the root node, recursively descend the tree, at each step choosing the closest node, using a distance between the node and the point defined on the CF properties.
4. When a leaf node is reached check if it is possible to place this point in that leaf. That is, does this point increase the radius of the leaf beyond T, or increase the number of particles in the leaf beyond L? If not, place the particle in the leaf, and return to item 2.

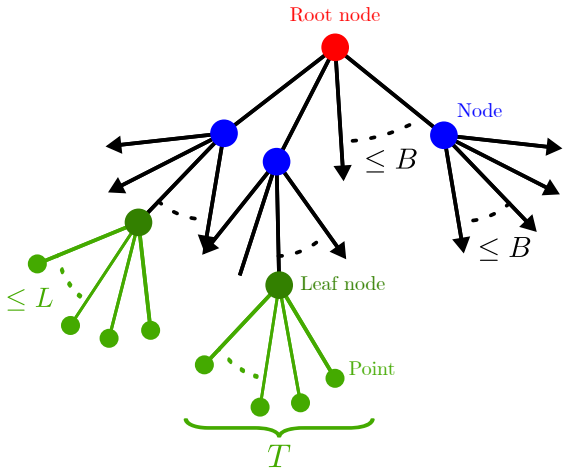


FIGURE 7.4: Depiction of the CF tree. This is a dendrogram where nodes represent potentials clusters. Non-leaf nodes (including the root node) may have up to B children. The leaf nodes may contain up to L points, but those points must not span a radius greater than T.

5. If the point does not fit in the leaf, then the leaf must be split. Chose the farthest pair of points in the leaf as seeds for new leaves, assign all remaining points according to which leaf seed they are closest to. The new leaf nodes need their CF properties recalculating from their points. Should this split cause the node above the leaf to have more than B children that node must also be split.

   (a) Splitting a node proceeds in the same manner as splitting the leaf; chose the farthest pair of children in the node as seeds for new nodes, assign all remaining children according to which node seed they are closest to.

   (b) The new nodes need their CF properties recalculating from their child nodes.

   (c) This split then propagates up the tree until either a node does not need to be split, because after splitting, its child it still has less than B children, or the root node is reached. If the root node is reached, it too is split, moving the root up another level.

   (d) In the final node that does not need to be split, check if the two closes children correspond to the new split. If not merge those children, and resplit if this generates a node with more than B children.

6. Once the splitting has finished, return to item 2.

These steps constitute the initial pass of the data, and create the CF tree. It is possible to stop here, defining the nodes separated by some suitable distance as the clusters, or treating the nodes as input points to another clustering algorithm.

A second optional step is to make the tree a little more compact by increasing T and reallocating the points to this new tree. The objective being to obtain a tree with fewer leaves. This can be done by considering each leaf to be the destination at the end of a

unique path from the root. For each node the children are assigned a number between 1 and L. Then the path from the root to the leaf can be described as a sequence of numbers, each number being the number assigned to the child that must be followed to find the leaf.

While there are still particles left in the old tree;

1. Chose a path to a leaf in the old tree, call this path p_{old}.
2. Create a path in the new tree with all the same node numbers as p_{old}, and call this p_{new}.
3. For each point in the leaf of p_{old} try to place it in the leaf that is closest to it in the new tree, if that is not possible because that leaf is full, or the closes leaf is at p_{new} place it in p_{new}.
4. The leaf at p_{old} should now be empty and can be deleted. Some nodes in the old tree now may also have no children and can be deleted. This prevents to memory requirements from expanding too far.
5. Depending on where leaves have been placed in the new tree there may also be some empty leaves and unneeded nodes in the new tree. Remove these too.

These steps can make the CF tree smaller.

Optionally some steps to remove outliers may be employed at this point. However, if more time is available, improving the quality of the clusters is also possible. There are a variety of optional steps for this. These include reducing the size of the tree by increasing T and clustering points within the leaves.

Finally, global clustering completes the algorithm. The leaf nodes are each treated as a single data point, and their group allocation defines the allocation of the points within them [4].

The algorithm is hierarchical and agglomerative. This algorithm is a powerful scalable algorithm. From the description given here it is clear that is it very complicated, more complicated that anything else considered in this chapter so far. Whether this is a drawback is very much a matter of perspective.

A further limitation of BIRCH is that it performs poorly when the points exist in high dimensions [228].

[4] Interestingly, this is only implied in the original definition of the BIRCH algorithm [236], but it is explicitly stated in later works [238].

7.1.1.9 Gaussian Mixture

This algorithm attempts to find a good fit for the data points, by treating them as samples drawn from a number of Gaussian distributions [239]. The most basic version specifies a number of clusters, and which parameters of the Gaussian distributions chosen to represent those clusters may vary. Initially the Gaussian are randomly generated. They are then tuned to maximise the expectation of the points with an iterative algorithm.

This is a partition based algorithm. It has the advantage of simplicity, and the capacity to generate fuzzy sets.

7.1.2 Comparative Conclusions on Algorithms Described

In this section, 9 out of the 10 algorithms depicted in Figure 7.1, have been described in depth and placed in the basic taxonomy of partition based or hierarchical. These are all implemented in the popular and widely available `python3` package `scikit-learn` [228]. So they constitute a range of algorithms that see wide practical use.

It has been seen that 6 out of 9 algorithms are partition based algorithms. Of the 3 a that were hierarchical, all of them were agglomerative rather than divisive. This may be the result of the computational challenge of considering splits on a large dataset, as opposed to considering the combination of a single point with another, as much as anything else. Partitional algorithms typically require the number of clusters to be specified. There are ways round this, such as trying the algorithm multiple times and varying the number of clusters required.

A number of the algorithms involved assume the particles exist in a Euclidean space;

- The KMeans algorithm is arguably intrinsically defined in Euclidean space, because it relies on minimising the variance of the data points about each mean, which so happens to correspond to a sum of the Euclidean distances from the centroid. This process is guaranteed to converge. Adapting this for other metrics is possible, with convergence still guaranteed. The most general form would be clustering with Bregman divergences [240].
- Ward clustering is specific to points that can be mapped into Euclidean space.
- Gaussian mixture models do assume a Euclidean metric, there has been some work to generalise this [241], but it is not often done.
- Mean shift uses a kernel to assess density, which is normally defined in Euclidean space. Here there is a slight additional complication, because while a different

kernel could be defined, mean shift may not converge in a finite number of steps for any kernel, but provided a cut off can be chosen, this is not a deal breaker.

This assumption of Euclidean space forms part of the objective of those clustering functions. The remaining algorithms rely on some distance, but do not require a specific metric. For a physical application, where points may be relativistic particles, algorithms that assume a Euclidean metric may require some adaptation. Particles are often represented as points in rapidity (or pseudorapidity) – ϕ space. Pairwise distance between these particles would often be expressed as Euclidean. However, when two (or more) particles merge, the resulting particle would not naturally be at the Euclidean geometric mean, because the rapidity (or pseudorapidity) coordinate is not cumulative. So, if an algorithm requires a centroid for a set of particles, the centroid cannot be found with a Euclidean geometric mean if it is to have a physical interpretation. This is not an insurmountable problem, in many cases it would be as simple as locating the centroid using the particle's four-momentum, rather that rapidity. Even if an algorithm ignored this non-euclidean distance behaviour, it might well still find sensible results.

Conceptual complexity of the algorithms varies greatly as well. A more complex algorithm may obscure the physical interpretation of the clusters formed. This is a significant loss; as clustering is frequently done without a ground truth, defining an optimal solution requires some, context specific, interpretation. This is broadly the case in jet formation. A clear interpretation of the relationship between the physics and the clustering criteria offers greater insight into the cluster algorithms suitability.

This is not to be confused with computational cost, which also varies, but has different implications. The events we deal with are likely to have $\mathcal{O}(100)$ particles; 100 points is not an excessive number for a clustering algorithm. Computational time is still of major interest, because of the sheer number of events, but loading a whole event, or set of points to be clustered, into memory is not the difficulty. The points themselves do not have a great number of dimensions either.

From this discussion, it can be seen that some clustering methods will be unsuitable because they are designed for a Euclidean metric, which doesn't apply to out data. Other clustering algorithms are designed for a fixed number of clusters, which would need to be remedied. Some algorithms are designed to mitigate issues we do not expect to encounter, such as having so many points to cluster that memory became restricted.

From what we have seen so far agglomerative clustering passes all these checks, and this is the principle of the excellent generalised k_{T} algorithm, that is already used for jet formation with great success. OPTICS might also be a promising choice, although it is a complex algorithm, which would need careful physical interpretation.

7.2 Objective of Spectral Clustering

All clustering algorithms have some function of the clusters that they seek to optimise. This can be explicit, as in K-Means, where we are minimising variance in the cluster, or implicit, such as affinity propagation, which seeks to minimise distances of all points and a common central point. If the objective of the clustering function does not align with the motivation for clustering the data, then no matter how well the algorithm performs, the output will be undesirable.

Spectral clustering is a class of clustering algorithms, all of which have an explicit objective function with a common format, and all of which use the same mechanism, the eigenspectrum, to optimise their objective function [242]. This is known as the Laplacian eigenmap [243]. Originally, this construction was conceived as a means to set a lower bound on the number of edges that would cross between groups when partitioning an unweighted graph [244]. It is a procedure often employed in applications such as image segmentation [245]. There are now a great wealth of variations of algorithms which use this technique, on various types of graphs, and various objective functions.

To give a concrete example, a valid objective function for spectral clustering could be defined as follows. Define an affinity measure between all pairs of points. This affinity[5], $a_{i,j}$, should be positive and reflect the degree of belief that point i and point j belong in the same cluster. This way, our set of points becomes a graph, with each point being a vertex, and the edges of the graph being weighted by the affinities between the corresponding vertices. The affinities are not distances, and as such they do not represent any assumption about the space that the points occupy.

A naïve objective would be to be minimise

$$\sum_K \left(\sum_{i \in K, j \notin K} a_{i,j} \right) \tag{7.12}$$

where the first sum is a sum over all chosen clusters; $K = 1 \dots n$. This makes sense, a clustering that minimises this avoids placing particles with high affinity in separate groups, which follows perfectly from the definition of affinity. Unfortunately, this normally results in the majority of clusters only containing a single point. A denominator is needed to require some balance in the cluster sizes. The simplest choice is the number of points in the cluster, so let $|K| = \sum_{i \in K} 1$ be the number of points assigned to cluster K. Then a suitable objective function, called RatioCut, can be written as [246];

$$\text{RatioCut} = \sum_K \left(\frac{\sum_{i \in K, j \notin K} a_{i,j}}{|K|} \right) \tag{7.13}$$

[5]Sometimes also referred to as similarity [244].

It penalises clusters with varied numbers of points in them, thus disfavouring solutions that have isolated points.

Unfortunately, minimising ratio cut is NP hard [246], so it is not normally done directly. In section 7.3 remedies for this hurdle will be described.

There are many variants on the objective function in Equation 7.13 used for spectral clustering. The most common of these is the normalised cut [245];

$$\text{NCut} = \sum_K \left(\frac{\sum_{i \in K, j \notin K} a_{i,j}}{\sum_{i \in K} \sum_j a_{i,j}} \right). \tag{7.14}$$

Rather than judging a group's size to be the number of points in the group this variant considers a group's size to be the sum of degree of the vertices in the group. The degree of a node is the sum of all affinities associated with the edges that connect to that node.

A general expression can be written that captures any idea of any desired point size. This was first considered in [244] where it was named penalised cut. The objective function of penalised cut is;

$$\text{PCut} = \sum_K \left(\frac{\sum_{i \in K, j \notin K} a_{i,j}}{\sum_{i \in K} \pi_i} \right), \tag{7.15}$$

where π_i is any size that has been assigned to point i, which would be contributed to whichever group point i is placed in. This formulation allows great freedom to define a clustering function that suits the problem at hand.

7.3 Relaxation to Solve Spectral Clustering

These objective functions are no use without a tractable means to minimise them. To this end, a simple relationship to the eigenvectors of a matrix known as the graph Laplacian can be shown.

An excellent starting point is too look at minimising the RatioCut criteria for a set of N clusters. So to begin with, the Laplacian should again be taken to be $L = D - A$. This will be done using some number of indicator vectors, each of which have as many entries as there are points to be clustered. The entries of the indicator vectors are piecewise constant, having the same value for all points to be allocated to the same group. The case of $N = 1$ is trivial, and doesn't need any indicator vectors. The case of $N = 2$ turns out to be a special case, needing only one indicator vector, this was shown in [247] and it is given in detail in Appendix B.

Here, the general case for ratio cut is presented, as indicated in [247]. Let the clusters be K_n, where $n = 1 \ldots N$. Let $\overline{K_n}$ be everything not in cluster K_n; $\overline{K_n} = K_1 \cup \cdots \cup K_{n-1} \cup$

$K_{n+1} \cup \cdots \cup K_N$. Also let $|K_n|$ be equal to the number of points in K_n. In order to record the allocation of points to the clusters K_n, a set of indicator vectors, $f(n)$, can be defined such that

$$f(n)_i = \begin{cases} 1/\sqrt{|K_n|}, & \text{if } i \in K_n \\ 0, & \text{if } i \notin K_n. \end{cases} \tag{7.16}$$

So each indicator vector indicates membership of one group by its positive members, and all indicator vectors are perpendicular.

The target is to find an equation that constructs the $f(n)_i$ such that the ratio cut, Equation 7.13, is minimised. The keystone to this will be the unnormalised graph Laplacian; let the unnormalised graph Laplacian be

$$L = D - A \tag{7.17}$$

where A is a matrix of affinities, such that $A_{i,j} = a_{i,j}$ and $A_{i,i} = 0$, and D is a diagonal matrix with $D_{i,i} = \sum_j a_{i,j}$.

Now a link can be drawn by taking the product of these things;

$$\begin{aligned} f(n)'Lf(n) =& \sum_{i,j} f(n)_i L_{i,j} f(n)_j \\ =& \sum_{i,j} f(n)_i \left(\delta_{i,j} \sum_p a_{i,p} - a_{i,j} \right) f(n)_j \\ =& \sum_i \left(f(n)_i^2 \sum_p a_{i,p} - \sum_j f(n)_i f(n)_j a_{i,j} \right) \\ =& \sum_{i,j} a_{i,j} \left(f(n)_i^2 - f(n)_i f(n)_j \right) \\ =& \frac{1}{2} \sum_{i,j} a_{i,j} \left(f(n)_i - f(n)_j \right)^2 \end{aligned} \tag{7.18}$$

which uses $a_{i,i} = 0$ and Equation 7.17 between lines one and two. Then using Equation 7.16, substitutions can be made for $f(n)_i$ and $f(n)_j$. If i and j are both in cluster K_n, or both not in K_n, then this leads to $f(n)_i - f(n)_j = 0$, so we only need to consider $i \in K_n, j \in \overline{K_n}$ or $i \in \overline{K_n}, j \in K_n$.

$$\begin{aligned} f(n)'Lf(n) =& \frac{1}{2} \sum_{i \in K_n, j \in \overline{K_n}} a_{i,j} \left(\frac{1}{\sqrt{|K_n|}} \right)^2 + \frac{1}{2} \sum_{i \in \overline{K_n}, j \in K_n} a_{i,j} \left(-\frac{1}{\sqrt{|K_n|}} \right)^2 \\ =& \sum_{i \in K_n, j \in \overline{K_n}} \frac{a_{i,j}}{|K_n|} \end{aligned} \tag{7.19}$$

Now notice that this is a single term from the RatioCut objective, Equation 7.13.

To obtain all the terms, let F be the matrix that is constructed by stacking $f(1) \dots f(N)$. This matrix will have dimensions (N, N). Now, if L is pre and post multiplied by this matrix, a matrix with every possible product of $f(n_1)'Lf(n_2)$ is produced. The only ones of relevance are when $n_1 = n_2$, which are found in the trace of this matrix. So taking this trace;

$$\mathrm{Tr}(F'LF) = \sum_n \frac{\sum_{i \in K_n, j \in \overline{K_n}} a_{i,j}}{|K_n|} \tag{7.20}$$

gives exactly the right objective.

There is another use that can be made of Equation 7.16; $\sum_i f(n)_i^2 = |K_n| \frac{1}{\sqrt{|K_n|}}^2 = 1$. As the indicator vectors are perpendicular, this leads to $f(m_1)'f(m_2) = \delta_{m_1,m_2}$, and therefore $F'F = \mathbb{1}$. So the cost function can be expressed neatly in terms of the indicator vectors, F, and the Laplacian, L.

$$\text{RatioCut} = \frac{\mathrm{Tr}(F'LF)}{F'F} \tag{7.21}$$

The aim is to minimise this. This is still not directly solvable, not with the requirements of Equation 7.17. But if those requirements are relaxed, and $f(n)_i$ is allowed to take any value, provided that $f(n)$ remain perpendicular to each other and $\mathbb{1}$, then the right hand side becomes the Rayleigh-Ritz quotient. Minimising this is done by finding eigenvector associated with the smallest eigenvalue. After the relaxation, these vectors are no longer called indicator vectors, they are now simply the eigenvectors of the Laplacian.

Given the form of this Laplacian, the very smallest eigenvalue will be 0, and the corresponding eigenvector will be a constant vector. This could be seen as representing the trivial solution of all points in the same cluster. The following eigenvectors, in order of accenting eigenvalue, should be closely related to the indicator vectors that would optimally partition this graph.

So with a small relaxation, a solution to Equation 7.13 has been found. Following on from that, solutions to the alternative objectives are wanted. A solution for the penalised cut objective, Equation 7.15, is sufficient to cover any other case.

$$\text{PCut} = \sum_K \left(\frac{\sum_{i \in K, j \notin K} a_{i,j}}{\sum_{i \in K} \pi_i} \right), \tag{7.22}$$

This solution follows the method described in [242]. The indicator vectors should be initially imagined as;

$$f(n)_i = \begin{cases} 1/\sqrt{\sum_{p\in K_n} \pi_p}, & \text{if } i \in K_n \\ 0, & \text{if } i \notin K_n. \end{cases} \tag{7.23}$$

Then Equation 7.17 and Equation 7.18 remain as before. The next change arrives at Equation 7.19;

$$\begin{aligned} f(n)'Lf(n) =& \frac{1}{2} \sum_{i\in K_n, j\in \overline{K_n}} a_{i,j} \left(\frac{1}{\sqrt{\sum_{p\in K_n} \pi_p}} \right)^2 + \frac{1}{2} \sum_{i\in \overline{K_n}, j\in K_n} a_{i,j} \left(\frac{1}{\sqrt{\sum_{p\in K_n} \pi_p}} \right)^2 \\ =& \frac{\sum_{i\in K_n, j\in \overline{K_n}} a_{i,j}}{\sum_{i\in K_n} \pi_i} \end{aligned} \tag{7.24}$$

This looks like the corresponding element of the penalised cut objective. So then the whole objective can be recreated by stacking this new $f(n)$ into a F and applying the same trace trick, $\mathrm{Tr}(F'LF)$. Unfortunately, $F'F \neq \mathbb{1}$, so a further step is needed to use the Rayleigh-Ritz theorem. Instead, $f(n)'f(n) = \sum_{i\in K_n} \pi_i$, and so let us define an N by N diagonal matrix, Z, with $Z_{m,n} = \delta_{m,n} \sum_{i\in K_n} \pi_i$. Notice that;

$$F'ZF = \begin{pmatrix} \frac{\sum_{i\in K_0} \pi_i}{\sqrt{\sum_{i\in K_0} \pi_i}^2} & 0 & \dots & 0 \\ 0 & \frac{\sum_{i\in K_1} \pi_i}{\sqrt{\sum_{i\in K_1} \pi_i}^2} & & \\ \vdots & & \ddots & \\ 0 & & & \frac{\sum_{i\in K_N} \pi_i}{\sqrt{\sum_{i\in K_N} \pi_i}^2} \end{pmatrix} = \mathbb{1} \tag{7.25}$$

Then let $T = Z^{\frac{1}{2}}F$, which makes a matrix T such that $T'T = \mathbb{1}$. Effectively this is just a rescaling of each of the $f(n)$. Then;

$$\mathrm{PCut} = \frac{\mathrm{Tr}(T'Z^{-\frac{1}{2}}LZ^{-\frac{1}{2}}T)}{T'T} \tag{7.26}$$

Allowing for the same relaxation as before, this yields a form which can be minimised according to the Rayleigh-Ritz theorem. Instead of solving the eigenvalue equation for $L = D - A$, the eigenvalue equation of $L_{\mathrm{PCut}} = Z^{-1/2}(D - A)Z^{-1/2}$ must be solved.

This time the smallest eigenvalue is again 0, and the corresponding eigenvector simply reflects the π_i. The following eigenvectors, in order of accenting eigenvalue, should be closely related to the indicator vectors that would optimally partition this graph.

7.4 After the Relaxation

Of course, this is not yet a clustering algorithm. The relaxation prevents the eigenvectors from providing clean separations. Instead, the eigenvectors are used to form what is known as an embedding space [242]. Each point in the original problem is allocated a set of coordinates in the embedding space by taking one coordinate value from each eigenvector.

To put this in more concrete terms, say that the N eigenvalue equations, corresponding to the $2 \dots N+1$ smallest eigenvalues have the form;

$$Lx(n) = \lambda_n x(n), \tag{7.27}$$

with $n = 1 \dots N$

Then the coordinates for the ith point in the embedding space will be;

$$y_i = (x(0)_i, x(1)_i, \dots x(N)_i) \tag{7.28}$$

There are a variety of methods for resolving points in this embedding space into clusters. A common first step is to normalised the lengths of the embedding space vectors, y_i, so that they are placed on the surface of a unit hypersphere [248, 249, 250]. The reasons for this are best explained in [248]; clusters in the embedding space are expected to be orthogonal to one another, thus absolute distance from the centre should not separate points.

Following the normalisation of y_i, they should be subjected to a clustering algorithm, such as k-means. The cluster allocation of the points in the embedding space determines their allocation in the real space.

7.5 Spectral Clustering Algorithms in Other Works

Although there are no previous attempts to apply spectral clustering for jet formation, Spectral clustering has also had success in other physics contexts. One such example is in identifying the motion of vortices [251] in fluid dynamics, to determine the correct number of clusters to contain the vortices.

Furthermore, another successful application was found in organising power grids. To reduce the risk of blackouts, power grids may be subdivided into 'islands', which are electromechanically stable regions with minimum load shedding. The ideal location of such islands is found by minimising the power flow between them using spectral clustering as shown in [252].

7.6 Potential for Spectral Clustering in Jet Formation

Having seem many clustering methods and considered spectral clustering in depth, all that remains is to synthesise this information to motivate the choice of spectral clustering for jet formation. There are a number of quality of spectral clustering that are desirable in a candidate for jet formation;

- The explicit objective function makes it easier to relate the clustering goals to the physics goals. In addition, the algorithm itself is not excessively complex. These are not uncommon traits for clustering algorithms, none the less they are very important in this application.
- Flexibility of the affinity measure; the affinity measure, which indicates how likely two points are to belong to the same group, has few constraints on it. This could be inspired by existing successful jet clustering algorithms. This is somewhat unusual, most algorithms put some constrain on the relationship between points.
- Flexibility of the sizes of the points, π_i, a physically inspired choice is possible. Some form of weighting is possible with many algorithms.
- The final step, where the points are clustered in the embedding space, is not determined by the main mechanism in spectral clustering. This is another point of flexibility, which may be used to form a variable number of clusters. Most algorithms would not permit this without fundamentally altering the objective function.

While each point individually can be fulfilled with a different choice of algorithm, the combination is somewhat unusual. However, there are some not insignificant hurdles to this application;

- Eigenvalue equations are expensive operations. There are ways to mitigate this, particularly if only a limited subset of eigenvectors are needed.
- It is not immediately obvious that the resulting clusters will be IR safe[6]. This is a very important property, and the algorithm must be engineered carefully to obtain it.

These challenges do not seem insurmountable. Spectral clustering is a choice worth exploring. If its behaviour was favourable, one might hope to obtain additional information, useful for classifying jets and events from the embedding space created in the clustering.

[6] See section 4.6.1.

Chapter 8

Spectral Clustering for Jet Physics

This section is drawn from the work published in [2]. This work was co-authored with Srinandan Dasmahapatra, Billy G. Ford, Stefano Moretti, and Claire H. Shepherd-Themistocleous.

Decisions made about the direction and content were primary driven by Srinandan Dasmahapatra, Stefano Moretti and myself, with Professor Claire Shepherd offering guidance from an up to date experimental understanding. I combined existing tools, such as `MadGraph`, `Pythia8` and `FASTJET`, with a considerable repository of custom `python3` code. Stefano Moretti also provided some `Fortran77` scripts for the calculation of jet shape variables. Together, I used these tools to generate and pre-process data, cluster the data with existing and new clustering algorithms, evaluate the performance and behaviour of many potential choices. For much of the preprocessing, and some of the existing clustering algorithms, Billy Ford provided validation, by undertaking the same operations without using any of my custom python code. This validation was done using `MadGraph`, `Pythia8`, `FASTJET` and `MadAnalysis`. The group collectively analysed the findings, and the text of the original publication, [2], was a collective effort.

8.1 Introduction

As discussed in section 4.6.3.3, the preferred choice for jet clustering in the context of hadron collider physics tends to be a simple agglomerative algorithm. There are three common choices, the anti-k_T [123, 124, 125], the Cambridge-Aachen [119, 120] or the k_T one [122]. They have been the default choice for some time because they have a number of desirable properties. They are IR safe, excellent implementations of them are publicly available (see `FASTJET` [253]) and they are flexible enough to capture many different jet signals with minimal parameter changes. These algorithms are recursive

(or iterative) and agglomerative. A recursive algorithm is well suited to clustering objects when the number of groups is not known from the outset. Agglomerative algorithms create jets by grouping objects, starting from individual particles, and continuing to combine the groups of particles into larger groups, until the desired jet size is reached. Creating jets that are IR safe can be achieved by ensuring that pairs of particles emerging from collinear emissions, combine at the start of this process. Once these IR splittings have been recombined they cannot influence the rest of the clustering process.

Jet definition precedes further algorithmic methods to extract useful physical quantities. Finding an alternative clustering method that compares favourably to these popular jet algorithms, and which offers additional features for further analysis, is the objective of this work. Success in obtaining clusters based on informative transformations of the data offers the possibility of exploiting such representations. This work uses spectral clustering, as described in section 7.2, to allocate reconstructed particles to jets.

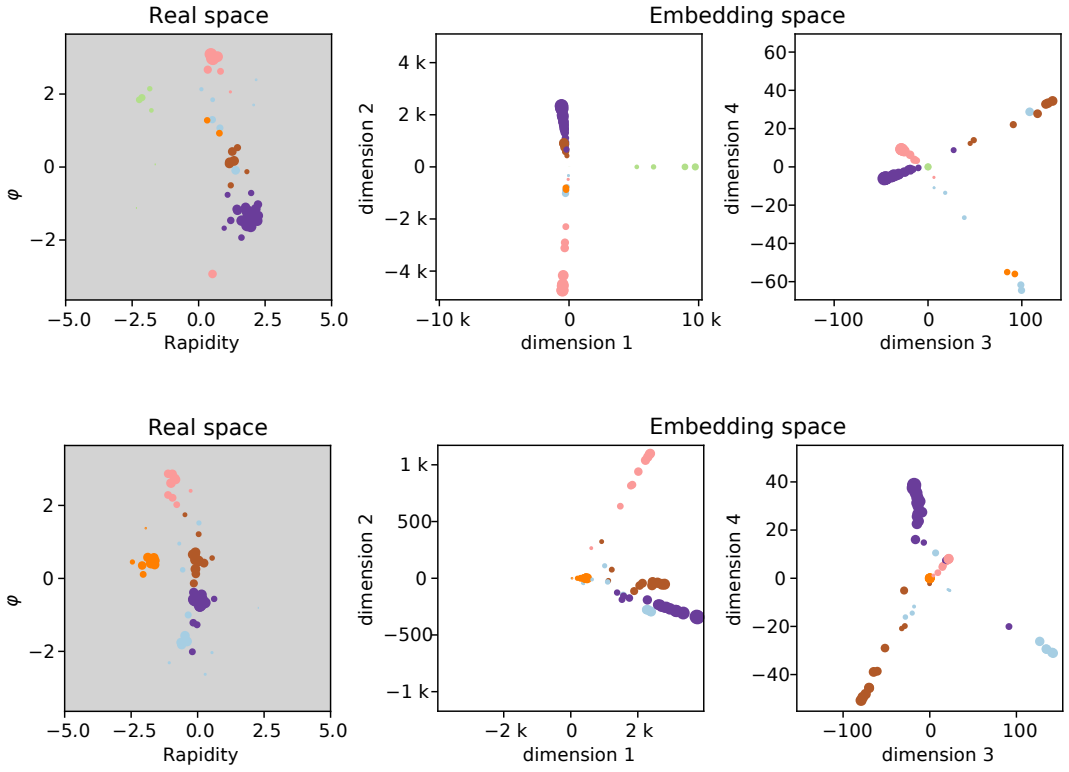


FIGURE 8.1: Two events and their embedding space, as created by spectral clustering. To the left the grey plot shows the particles in the event as points on the unrolled detector barrel. The colour of each point indicates the shower it came from. On the right, two plots show the first 4 dimensions of the embedding space and the location of the points within the embedding space. The event in the first row is cleaner than the one in the second row, the second row will be more challenging to correctly cluster.

8.2 Method

In this section the methodology is covered in five parts. First, some general principles of working with a spectral embedding are discussed. Second, the algorithm chosen in this work for applying spectral clustering is given. Third, choices and interpretations for the variable parameters in this algorithm are given. Fourth, the datasets against which this method will be measured are specified. Fifth, the procedure for checking IR sensitivity is described.

8.2.1 Working in the Embedding Space

An example of an embedding space constructed from eigenvectors is shown in Figure 8.1. It is formed following the logic set out in section 7.3. The precise details of the algorithm used will be given in section 8.2.2 The image illustrates how the embedding space highlights the clusters, with which some key aspects of the embedding space can be better visualised.

8.2.1.1 Distance in the Embedding Space

When the spectral clustering algorithm is used to create an embedding space from a set of points, the points are distributed in the embedding space according to their ideal group. Each point can be seen as a vector, its direction indicating the group to which this point should be assigned. Changes in magnitude of the vectors cause the Euclidean distance between the corresponding points to grow, however, this is not an indicator of the correct grouping. An angular distance is invariant to changes in magnitude, therefore it is a suitable measure to use.

8.2.1.2 Information in the Eigenvalues

When the clusters in the data are well separated, the affinities between groups are close to 0 and the eigenvalues will also be closer to 0. Should a group exist which has exactly 0 affinity with the rest of the graph, then the Laplacian would be possible to put in block diagonal form, and the corresponding eigenvalue would be exactly zero. So a small eigenvalue means that the corresponding eigenvector is separating the particles cleanly according to the affinities. It is possible to make use of this information.

In a traditional application of spectral clustering, the number of clusters desired, s, is predetermined. If an embedding space is created by taking c eigenvectors, corresponding to the smallest eigenvalues, excluding the trivial eigenvector. Then the usual choice is $c = s$. The embedding space then has $c = s$ dimensions.

When forming jets we do not know from the outset how many clusters to expect in the dataset, so the number of eigenvectors to keep is not clear. In this case, it's not possible to set $c = s$. While one could choose a fixed, arbitrary number of eigenvectors, this is suboptimal. A better approach is to take all non-trivial eigenvectors corresponding to eigenvalues smaller than some limiting number, λ_{limit}. For a symmetric Laplacian the eigenvalues are $0 \leq \lambda_{\mathbf{1}} \leq \lambda_{\mathbf{2}} \leq \cdots \lambda_{\mathbf{n}} \leq 2$, and $\lambda_{\mathbf{k}}$ is related to the quality of forming $\mathbf{k}$ clusters [254]. Removing eigenvectors with eigenvalues close to 0 would result in discarding useful information, while retaining eigenvectors whose eigenvalues are close to 2 would increase the noise. Values of $0 < \lambda_{\text{limit}} < 1$ are sensible choices, and within this range the choice is not critical. Then, the number of dimensions in the embedding space will vary, according to the number of non-trivial eigenvectors with corresponding $\lambda < \lambda_{\text{limit}}$.

There is one more manipulation from the information in the eigenvalues. The dimensions of this embedding space are not of equal importance. This can be accounted for by dividing the eigenvector by some power, β, of the eigenvalue.

Let the eigenvectors for which $\lambda < \lambda_{\text{limit}}$ be

$$\sum_j L_{i,j} x_{\mathbf{n}\,j} = \lambda_{\mathbf{n}} x_{\mathbf{n}\,i}. \tag{8.1}$$

Then, the coordinates of the j^{th} point in the c dimensional embedding space become $m_j = \left(\lambda_{\mathbf{1}}^{-\beta} h_{\mathbf{1}\,j}, \ldots \lambda_{\mathbf{c}}^{-\beta} h_{\mathbf{c}\,j},\right)$. In effect, the magnitude of the vectors, m_j, in the n^{th} dimension are compressed by a factor $\lambda_{\mathbf{n}}^{\beta}$, so the larger $\lambda_{\mathbf{n}}$ the greater the compression.

8.2.1.3 Stopping Conditions

If a recursive algorithm is to be chosen, like the generalised k_{T} algorithm, a stopping condition is needed. A stopping condition based on smallest distance between points in the embedding space was attempted in this study but this was not found to be stable. Choosing an acceptable value for all events was not possible.

Distance between the last two points to be joined before the desired jets have been formed varies significantly between events, so minimum separation is not a good stopping condition. The average distance between points before this last joining is more stable because it is balanced by two opposing influences. When points are joined together in a fixed number of dimensions the average distance between points rises. If this were used in physical space it would be roughly proportional to the number of points remaining. So, in physical space, if clustering stopped when average distance exceeded some cut-off, it would produce roughly the same number of jets in each event. However, the embedding space has a variable number of dimensions. When lots of clustering still remains to be done the lower eigenvalues mean that the embedding space has more dimensions, as described in section 8.2.1.2. When the number of dimensions in the embedding space falls, the mean distance between points will also fall.

As points combine the mean distance will rise, but when fewer combinations with higher affinity remain the number of dimensions in the embedding space falls, counteracting the rise in mean distance. In short, the mean distance in the embedding space makes a natural cut-off. The assertions made here are evidenced in Appendix C.

8.2.2 Spectral Clustering Algorithm

For every simulated event, the following process is used to identify the jets. To begin with, relevant cuts are applied to the particles to simulate the detector reconstruction capability. (These are described in detail in section 8.2.4.) Then all particles are declared pseudojets and given an index, $j = 1 \ldots n$, with no particular order. The algorithm is agglomerative, recursively selecting pairs of pseudojets to merge, hence, the first iteration step is labelled $t = 1$.

When the two pseudojets to be merged, i and j, have been identified, they are combined using the E-scheme. The E-scheme forms a new pseudojet by summing the 4-momenta of the two joined pseudojets, $p(t+1)_l = p(t)_i + p(t)_j$. The steps used to select two pseudojets to merge proceed as follows.

1. The pseudojets are used to form the nodes of a graph, the edges of which will be weighted by some measure of proximity between the particles called affinity. To obtain an affinity, first a distance is obtained. Between pseudojets i and j this is

$$d(t)_{i,j} = \sqrt{(y(t)_i - y(t)_j)^2 + (\phi(t)_i - \phi(t)_j)^2}, \tag{8.2}$$

   where $y(t)_j$ is the rapidity of pseudojet j at step t and $\phi(t)_j$ is the angle in the transverse plane, likewise for i. No p_T (transverse momentum) dependence is used, unlike in many traditional jet clustering methods.

2. The affinity must increase as pseudojets become more similar, whereas the distance, $d(t)_{i,j}$, will shrink. Affinity is defined as

$$a(t)_{i,j} = \exp(-d(t)^{\alpha}_{i,j}/\sigma_v), \tag{8.3}$$

   where $\alpha = 2$ is the standard Gaussian kernel as used in [243]. Distances much larger than σ_v are only allowed very small affinities, thus less influence over the clustering.

3. Pseudojets that are far apart have low affinity, hence are unlikely to be good candidates for combination. Removing these affinities reduces noise. A fixed number, k_{NN}, of neighbours of each pseudojet is preserved while all other affinities are set to zero. Thus, when there are more than k_{NN} pseudojets, each pseudojet has at least k_{NN} non-zero affinities with other pseudojets.

4. These affinities allow the construction of the Laplacian, which is proportional to $-a(t)_{i,j}$ in the i^{th} row and j^{th} column. For ease of notation, let $z(t)_j$ be the sum of all affinity connected to particle j at step t; $z(t)_j = \sum_i a(t)_{i,j}$. Also, let $w(t)_j$ be a measure of the size a pseudojet j contributes to a cluster, beginning with the same value as $z(1)_j$; $w(1)_j = \sum_k a_{j,k}$. Define square matrices $A(t)_{i,j} = (1 - \delta_{i,j})a(t)_{i,j}$, $Z(t)_{i,j} = \delta_{i,j} z(t)_i$ and $W(t)_{i,j} = \delta_{i,j} w(t)_i$.

   Then, the Laplacian used is written

$$L(t) = W(t)^{-\frac{1}{2}}(Z(t) - A(t))W(t)^{-\frac{1}{2}}. \tag{8.4}$$

   After each step this Laplacian shrinks by one row and column. When two pseudojets have been combined, instead of calculating w_j as the sum of the affinities of the combined pseudojet, the new w_j is the sum of the two previous w_j's. And so,

if pseudojets i and i from step t are to be combined to make pseudojet i at $t+1$, then $w(t+1)_i = w(t)_i + w(t)_j$ rather than $w(t+1)_1 = \sum_k a(t+1)_{i,k}$.

5. The eigenvectors of $L(t)$ ($\mathbf{q}$ being the eigenvalue index)

$$L(t)h(t)_{\mathbf{q}} = \lambda(t)_{\mathbf{q}} h(t)_{\mathbf{q}},\ \mathbf{q} = 1, \ldots c \tag{8.5}$$

are used to create the embedding of the pseudojets. The eigenvector corresponding to the smallest eigenvalue represents the trivial solution, which places all points in the same cluster (see section 7.3). All non-trivial eigenvectors, corresponding to eigenvalues less than an eigenvalue limit, $\lambda(t)_{\mathbf{c}} < \lambda_{\text{limit}} < \lambda(t)_{\mathbf{c+1}}$, are retained (see section 8.2.1.2). If no eigenvectors are retained by this, the clustering ends here.

6. An eigenvector is divided by the corresponding eigenvalue raised to β. To prevent zero division errors, the smallest eigenvalues are clipped to 0.001, such that $\lambda'_{\mathbf{q}} = \min(\lambda_{\mathbf{q}}, 0.001)$. This acts to compress the dimensions that hold less information, again, see section 8.2.1.2. The embedding space can now be formed. The eigenvectors have as many elements as there are pseudojets and the coordinates of the j^{th} pseudojet at step t are defined to be $m(t)_j = \left(\lambda'_{\mathbf{1}}(t)^{-\beta} h_{\mathbf{1}}(t)_j, \ldots \lambda'_{\mathbf{c}}(t)^{-\beta} h_{\mathbf{c}}(t)_j\right)$.

7. A measure of distance between all pseudojets in the embedding space is calculated. In the embedding space angular distances are most appropriate (see section 8.2.1.1):

$$d'(t)_{i,j} = \arccos\left(\frac{m(t)_i \cdot m(t)_j}{\|m(t)_i\| \|m(t)_j\|}\right). \tag{8.6}$$

where $\|m\|$ is the (Euclidean) length of m.

8. A stopping condition, based on the parameter R, is now checked. Provided the mean of the distances $d'(t)_{i,j}$ is less than the value of R, that is,

$$\frac{2}{c(c-1)} \sum_{i \neq j} \sqrt{d'(t)_{i,j}} < R, \tag{8.7}$$

then the two pseudojets that have the smallest embedding distance are combined. (Reasons for this stopping condition are given in section 8.2.1.3.)

When the mean of the distances in the embedding space rises above R, then all remaining pseudojets are promoted to jets. Jets with less than 2 tracks are removed and their contents considered noise. Further cuts may then be applied as described in section 8.2.4.

These steps will form a variable number of jets from a variable number of particles.

8.2.3 Tunable Parameters

Unlike most deep learning methods currently used in particle physics, spectral clustering does not have large arrays of learnt parameters. The parameters for the clustering are a small, interpretable set. Appropriate values were chosen by performing scans and observing the influence of changes to the parameters on jets formed.

In section 8.2.2, 6 parameters are named: σ_v, α, k_{NN}, λ_{limit}, β and R. While these are more parameters than in generalised k_{T}, for example, we find that the parameters do not need to take precise values to obtain good performance.

The interpretation of these parameters is as follows.

- σ_v: introduced in step 2, this is a scale parameter in physical space. The value indicates an approximate average distance for particles in the same shower, or alternatively, the size of the neighbourhood of each particle. It is closely tied to the stopping parameter for the generalised k_{T} algorithm, R_{k_T}, and they both relate to the width of the jets formed. It should take values on the same order of magnitude as R_{k_T}.
- α: also introduced in step 2, this changes the shape of the distribution used to describe the neighbourhood of a particle. Higher values reduces the probability of joining particles outside σ_v. $\alpha = 2$ defines a Gaussian kernel.
- k_{NN}: introduced in step 3, it dictates the minimum number of non-zero affinities around each point. Lower values create a sparser affinity matrix, reducing noise at the potential cost of lost signal. Values above 7 are seen to have little impact.
- λ_{limit}: introduced in step 5, it is a means of limiting the number of eigenvectors used to create dimensions in the embedding space. Only eigenvectors corresponding to eigenvalues less than λ_{limit} are used. Thus, the number of dimensions in the embedding space can be increased with a larger λ_{limit}. However, as the eigenvalues will be influenced by the number of clear clusters available, there will not be the same number of dimensions in each event. Values of $0 < \lambda_{\mathrm{limit}} < 1$ are sensible choices, see discussion in section 8.2.1.2.
- β: introduced in step 6, it accounts for variable quality of information in the eigenvectors, as given by their eigenvalues, in such a way that the dimensions of the embedding spaces corresponding to higher eigenvalues are compressed, as they contain lower quality information. (This is discussed in section 8.2.1.2.)
- R: introduced in step 8, it determines the expected spacing between jets in the embedding space. As the number of dimensions in the embedding space grows with increasing number of clear clusters, it will not result in the same or similar number of clusters each time.

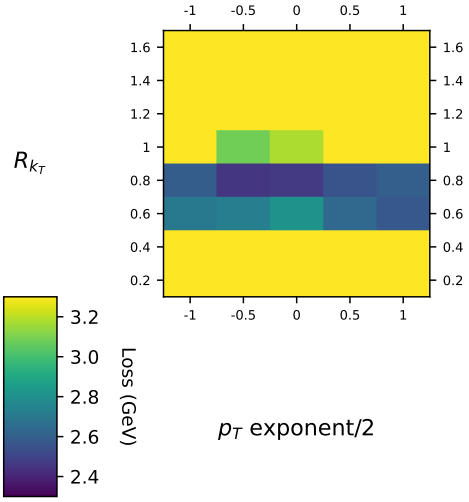


FIGURE 8.2: The generalised k_T algorithm has 2 parameters that can be varied. The stopping condition, R_{k_T}, and a multiple for the exponent of the p_T factor. When the exponent of the p_T factor is -1 the algorithm becomes the anti-k_T algorithm. Here, the "Loss", as described in Equation 8.8, is shown as a colour gauge for a number of parameter combinations.

To investigate the behaviour of the clustering when the parameters change, scans where performed. On a small sample of 2000 events the clustering is performed with many different parameter choices.

With the aid of MC truth information a metric of success can be created. For each object that must be reconstructed (e.g., a b-quark) the MC truth can reveal which of the particles that are visible to the detector have been created by that object. In many cases, a particle seen in the detector will have been created by two objects, such as a particle coming from an interaction between a $b\bar{b}$ pair, in these cases both objects are considered together. The complete set of visible particles that came from these objects could be referred to as their descendants. The aim in jet clustering is to capture only all of the descendants in the same number of jets as there were objects that created them. So the descendants of a $b\bar{b}$ pair should be captured in exactly 2 jets. The use of MC information has also been pursued in [255], for a jet originating from a colour singlet hard particle, namely a W boson. In contrast, this study seeks to find quark jets. Allowing the descendants of groups of interacting showers to be clustered in any configuration that results in the correct number of jets avoids the need to associate each descendant to one object (e.g. b-quark) uniquely, which is not possible when the objects in question are colour charged [255].

There are two ways a jet finding algorithm can make mistakes in this task: the first is to omit some of the descendants of the objects being reconstructed, causing the jet to have less mass than it should; the second is to include particles that are not in the descendants of the objects being reconstructed, such as initial state radiation or particles from other objects, causing the jet to have more mass than it should. The effects of these mistakes might cancel in the jet mass, but they are both still individually undesirable,

so separate metrics are made for each of them. The first is "Signal mass lost", the difference between the mass of the jets that were formed, and the mass they would have had if they had successfully captured all descendants of the object being reconstructed. The second is "Background contamination", the difference between the mass of jets that were formed, and the mass they would have if they did not contain anything but descendants they captured from the objects being reconstructed. A "Loss" function is then constructed as a weighted Euclidean combination of these two,

$$\text{Loss} = \sqrt{w\,(\text{Background contamination})^2 + (\text{Signal mass lost})^2}, \tag{8.8}$$

where w is a weight used to alter the preference for suppressing "Signal mass lost" versus reducing "Background contamination". When applying an anti-k_T algorithm, increasing R_{k_T} will result in lower "Signal mass lost", in exchange for a higher "Background contamination". This has been chosen to make a comparison to $R_{k_T} = 0.8$ as our sample dataset has well separated jets and low background. This value of R_{k_T} slightly prefers suppressing "Signal mass lost" over "Background contamination", to create the clearest mass peaks. To make the "Loss" reflect this we choose $w = 0.73$.

An example of this scan for the generalised k_T algorithm is given in Figure 8.2. It can be seen that, while good results are possible with many values of the p_T exponent, R_{k_T} must fall in a narrow range. We thus deem this choice of stopping condition, $R_{k_T} = 0.8$, to be rather fine-tuned.

For spectral clustering there are more than 2 variables to deal with, so a set of two dimensional slices are extracted. These slices have been chosen to include the best performing combination. As can be seen in Figure 8.3, the parameters choices are not fine-tuned, as many values can be chosen to achieve good results. For example, it can be seen that some parameters, such as α, k_NN, β and λ_limit, are relatively unconstrained, yielding good results for a wide range of numerical choices. Even when R and, especially, σ_v yield some large signal "Loss", say, for $R = 1.22$ or 1.3 and $\sigma_v = 0.05$, this happens in very narrow ranges. For definiteness, the parameters used in the remainder of this work are $\alpha = 2.$, $k_\text{NN} = 5$, $R = 1.26$, $\beta = 1.4$, $\sigma_v = 0.15$ and $\lambda_\text{limit} = 0.4$.

8.2.4 Particle Data

To evaluate the behaviour of the spectral clustering method four datasets are used,[1] all produced for the Large Hadron Collider (LHC).

[1] The first two uses a 2-Higgs Doublet Model (2HDM) setup as described in chapter 2 while the last two are purely Standard Model (SM) processes. Notice that all unstable objects are rather narrow, including the Beyond the SM (BSM) Higgs states [55, 56], so interference effects with their irreducible backgrounds have been neglected.

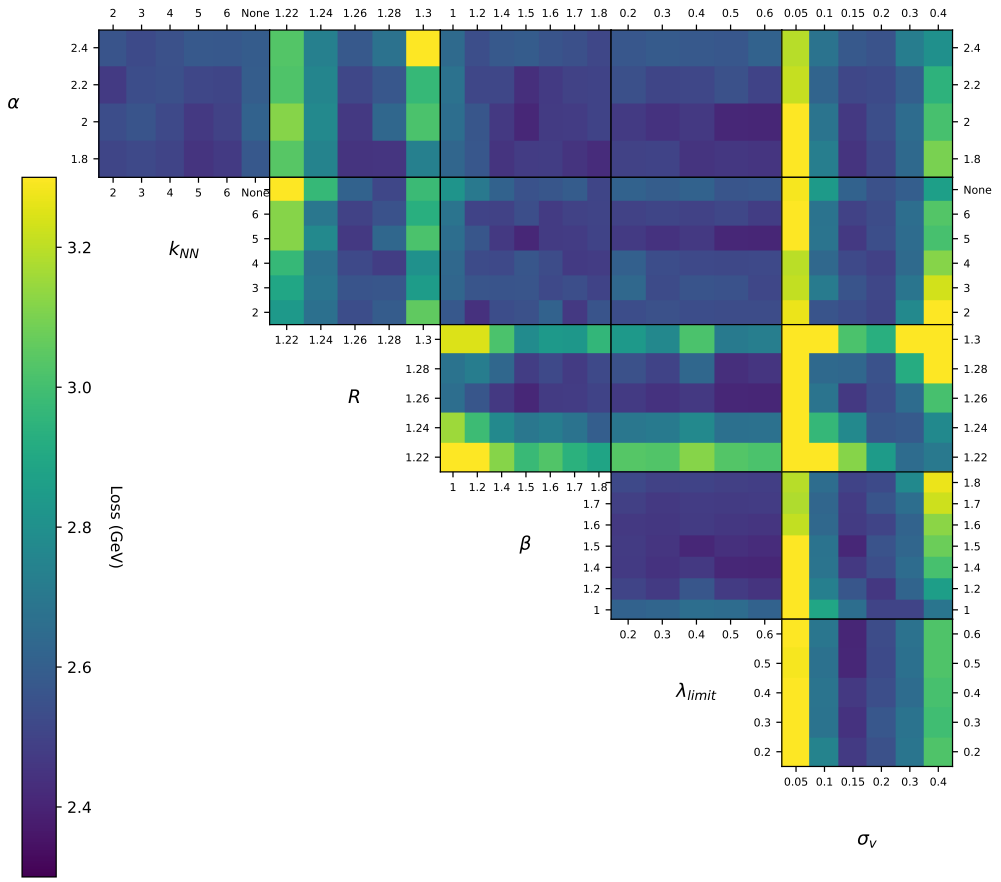


FIGURE 8.3: The spectral clustering algorithm has 6 parameters that can be varied (described in the text). Here, the "Loss", as described in eq. (8.8), is shown as a colour gauge for reasonable parameter ranges chosen either by convention (e.g., α is typically 1 or 2) or according to physical scales (e.g., σ_v is of order 0.1).

1. Light Higgs: A SM-like Higgs boson with a mass 125 GeV decays into two light Higgs states with mass 40 GeV, which in turn decay into $b\bar{b}$ quark pairs. That is, the process is $pp \to H_{125\text{ GeV}} \to h_{40\text{ GeV}} h_{40\text{ GeV}} \to b\bar{b}b\bar{b}$, simulated at Leading Order (LO).

2. Heavy Higgs: A heavy Higgs boson with a mass 500 GeV decays into two SM-like Higgs states with mass 125 GeV, which in turn decay into $b\bar{b}$ quark pairs. That is, the process is $pp \to H_{500\text{ GeV}} \to h_{125\text{ GeV}} h_{125\text{ GeV}} \to b\bar{b}b\bar{b}$, simulated at LO.

3. Top: A $t\bar{t}$ pair decays semileptonically, i.e., one $W^{\pm}$ decays into a pair of quark jets jj and the other into a lepton-neutrino pair $\ell\nu_\ell$ ($\ell = e, \mu$). That is, the process is $pp \to t\bar{t} \to b\bar{b}W^+W^- \to b\bar{b}jj\ell\nu_\ell$, simulated at LO. (Note that, here, $m_t = 172.6$ GeV and $m_W = 80.4$ GeV.)

4. 3-jets: For the purpose of checking IR sensitivity, we have used 3-jet events, this being a rather simple configuration where IR singularities could be observed. That is, the process is $pp \to jjj$, simulated at both LO and Next-to-LO (NLO).

Using `MadGraph` [71] to generate the partonic process and `Pythia8` [85] to shower, $\mathcal{O}(10^5)$ events for each of these processes are generated. A full detector simulation is not used, instead, cuts on the particles are imposed to approximate detector resolution, as detailed below.

The Center-of-Mass (CM) energy used is $\sqrt{s} = 13$ TeV.

Each event also contains (hard) Initial State Radiation (ISR) and soft QCD dynamics from beam remnants, i.e., the Soft Underlying Event (SUE). There is no pileup nor multiparton interactions in the datasets.

Each of these datasets requires different cuts, both at the particle level, to simulate detector coverage, and at the jet level, to select the best reconstructed events. The cuts on each dataset are as follows.

1. The reconstructed particles are required to have pseudorapidity $|\eta| < 2.5$ and transverse momentum $p_T > 0.5$ GeV. These cuts are likely to remove the majority of the radiation from beam remnants and reduce the radiation from ISR. The b-jets are required to have $p_T > 15$ GeV, which is possibly lower than is realistic [3], but it leaves a larger number of events to compare the behaviour of jet clustering algorithms.

2. The reconstructed particles are required to have $|\eta| < 2.5$ and $p_T > 0.5$ GeV. The b-jets are required to have $p_T > 30$ GeV, which is realistic for efficient b-tagging performance and further reduces ISR and the SUE. As the average jet p_T is higher we can afford this higher p_T cut.

3. The reconstructed particles are required to have $|\eta| < 2.5$ and $p_T > 0.5$ GeV. The event is required to have $p_T^{\text{miss}} > 50$ GeV, where p_T^{miss} is the missing transverse momentum due to the neutrino. The lepton in the event must have $|\eta| < 2.4$. If the lepton is a muon then its p_T must be > 55 GeV. If the lepton is an electron and it is isolated (as defined in [256]) then its p_T must be > 55 GeV, if it is not isolated then $p_T > 120$ GeV. The reconstructed jets must have $p_T > 30$ GeV and $|\eta| < 2.4$. Finally, the lepton must be separated from the closest jet by at least $\sqrt{\Delta\eta^2 + \Delta\phi^2} > 0.4$ or $p_T^{\text{relative}} > 40$ GeV. These cuts are copied from [166].

4. The only restriction on the particles is that the pseudorapidity must be < 2.5. There are no cuts on the jets. While unrealistic, since issues of IR sensitivity are emphasised at low p_T, to highlight this, all p_T cuts are abandoned.

The Higgs boson cascade datasets have the desirable property of creating b-jets with different kinematics: while in case 1 some slim jets may be expected (as on average they are rather stationary, because of the small mass difference between $H_{125\text{ GeV}}$ and $h_{40\text{ GeV}}$), in case 2 the jets are mainly fat jets (owing to the boost provided by the large

mass difference between $H_{500\ \mathrm{GeV}}$ and $h_{125\ \mathrm{GeV}}$). Mass reconstruction requirements for the Light Higgs and Heavy Higgs follow the same logic. In order to reconstruct a Higgs boson decaying directly to a pair of b-quarks, a separate jet tagged by each b-quark is required. In other words, the mass reconstruction of a Higgs is only attempted when there are two jets that were tagged by the b-quarks from that Higgs state. To reconstruct a Higgs boson that decays into a pair of (child) Higgs particles, requires both child Higgs boson to have been reconstructed, that is, all four b-jets are found.

In the case of the Top events three masses can be reconstructed from jets, the hadronic W, the hadronic top and the leptonic top. The hadronic W is reconstructed if both of the quarks it decayed to have tagged jets: they are permitted to tag the same jet, so the hadronic W can be reconstructed from one or two jets. The hadronic top is reconstructed if the b-quark from it has tagged a jet, so the correct b-jet is required in addition to the requirements on the W. The leptonic top is reconstructed if the b-quark from the top decay tags a jet and the missing momentum calculation which reconstructs the leptonic W yields a real mass. If the mass calculation for the leptonic W yields two real masses, the one closest to the true W mass is selected.

Now, spectral is compared to anti-k_{T} clustering to test IR sensitivity of the former, while this is a well-known feature of the latter. Following that, Higgs boson and top quark events can be studied.

8.2.5 Determining IR Sensitivity

It would be optimal to demonstrate IR safety analytically, however, that is beyond the scope of the current work. As the environment required for clustering on MC data is already set up, it is rather efficient for this study to prove that in practice the algorithm is not sensitive to IR considerations in simulated data. This can be done by showing that an IR sensitive variable, for example, the jet mass spectrum, is stable between a LO dataset with no IR singularities and a NLO one which will instead contain IR singularities.

This is a very important property, as the algorithm must not be modified by any approximation used for the IR limit in MC simulation.

Showing the jet mass spectrum at LO and NLO for a particular configuration, that is, a particular selection of clustering parameters, would allow a comparison that would highlight any differences caused by IR sensitivity. This will be done for illustrative purposes, however, since even an IR unsafe algorithm, such as the iterative cone one [123], has some configurations for which these singularities are avoided.

To provide a more global view, a scan of parameter configurations must be compared. Thus, for an unsafe algorithm (such as the iterative cone) the unsafe configuration will

be found. It would be cumbersome to compare all these jet mass spectra by eye, however. Instead, a summary statistic representing the divergence between two distributions is introduced; the Jensen-Shannon score [257].

The Jensen-Shannon score is a value computed between two distributions that increases in magnitude the more these distributions differ. It is a symmetrised variant of the Kullback-Leibler divergence [257]. The Kullback-Leibler divergence between probability densities p and q can be written as

$$D_{\mathrm{KL}}(p|q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx, \tag{8.9}$$

from which the Jensen-Shannon divergence can be written as

$$D_{\mathrm{JS}}(p,q) = \frac{1}{2} D\left(p|\frac{1}{2}(p+q)\right) + \frac{1}{2} D\left(q|\frac{1}{2}(p+q)\right). \tag{8.10}$$

Here, D_{JS} treats p and q symmetrically and will grow as they become more different. The spectrum of Jensen-Shannon scores will be plotted for a known IR safe clustering algorithm, generalised k_{T}, a known unsafe clustering algorithm, iterative cone, and the spectral algorithm. If the Jensen-Shannon scores for spectral are consistently small, then it is not sensitive to IR.

8.3 Results

Before the behaviour of the algorithms is analysed, some plots of kinematic variables are shown in Figure 8.5. It can be seen that the algorithms do not greatly differ on the kinematics of the events. In particular, spectral clustering does not appear to sculpt any distributions in any of the datasets involving Higgs bosons and top (anti)quarks. Some edge effects can be seen in the rapidity plots on the central column. When a particle shower is spread over the boundary of the calorimetric at $|\text{rapidity}| = \pm 2.5$ then with a wider jet radius the detectable particles ($|\text{rapidity}| \leq 2.5$) may be merged into another jet sitting at a more central rapidity, or if the jet cone is too narrow for that, they will form a "half jet" on their own. These half jets only contain the parts of the shower that have $|\text{rapidity}| \leq 2.5$, and so the rapidity of the jet is offset towards lower rapidity than those of the shower. This creates the edge effect.

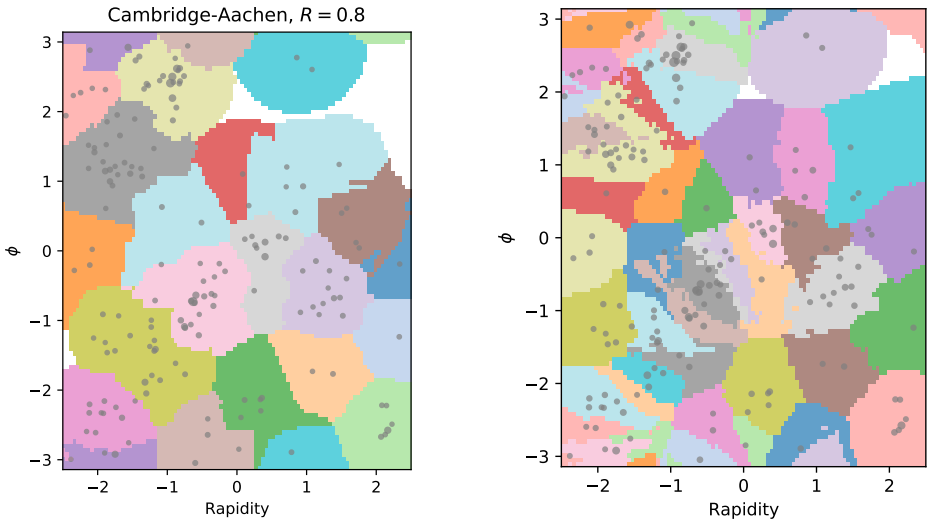


FIGURE 8.4: Images comparing the shape of jets produced by generalised k_T to spectral. The filled area represents all locations at which an additional particle would be included into the jet, if it were present. For discussion of edge effects in rapidity, see section 8.3.

8.3.1 IR Sensitivity

Shape variables (see section 4.6.2) such as jet mass, thrust, sphericity, spherocity and oblateness, are sensitive to IR divergences. For each configuration of the clustering algorithm we expect an IR safe algorithm to present a stable transition in a shape variable from the LO to NLO datasets, as significant changes in the spectra would indicate sensitivity to soft and collinear radiation. The clustering and evaluation here is done using the 3-jets dataset, as described in section 8.2.4. Shape variables are calculated from the

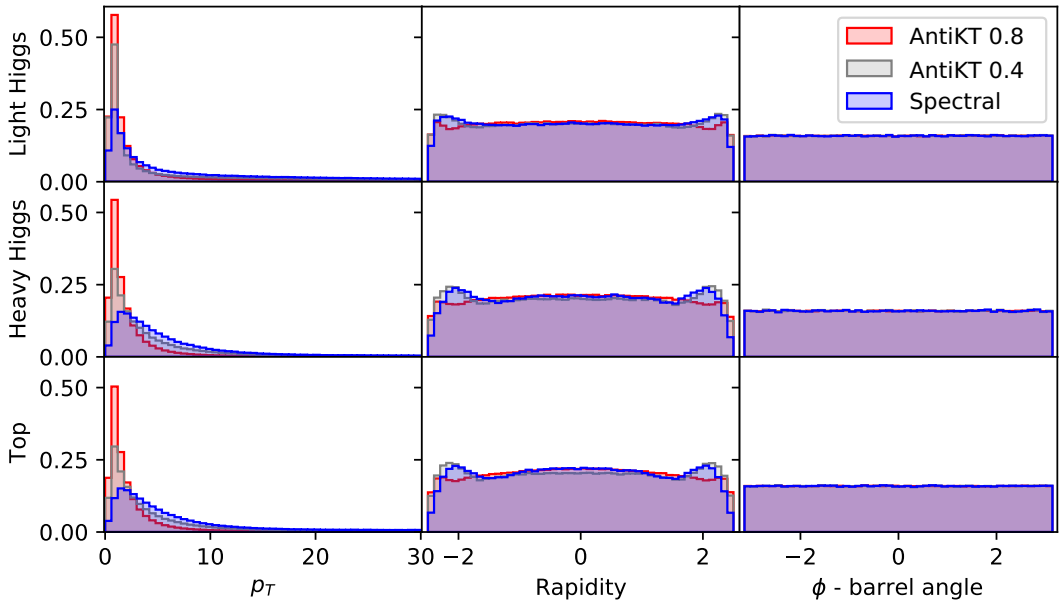


FIGURE 8.5: Basic jet variables for each of the analysis datasets and three clustering algorithms. In the first column there are some noticeable differences in the transverse momentum. In the second column the rapidity shows that the algorithms cluster jets at the edge of the barrel slightly differently. In the third column the barrel angle shows no noticeable changes.

total momentum of the 4 jets with highest p_T in each event. This comparison is made in Figure 8.6. It can be seen in this figure that little difference exists between generalised k_T and spectral clustering, so as to reinforce that they are both insensitive to IR.

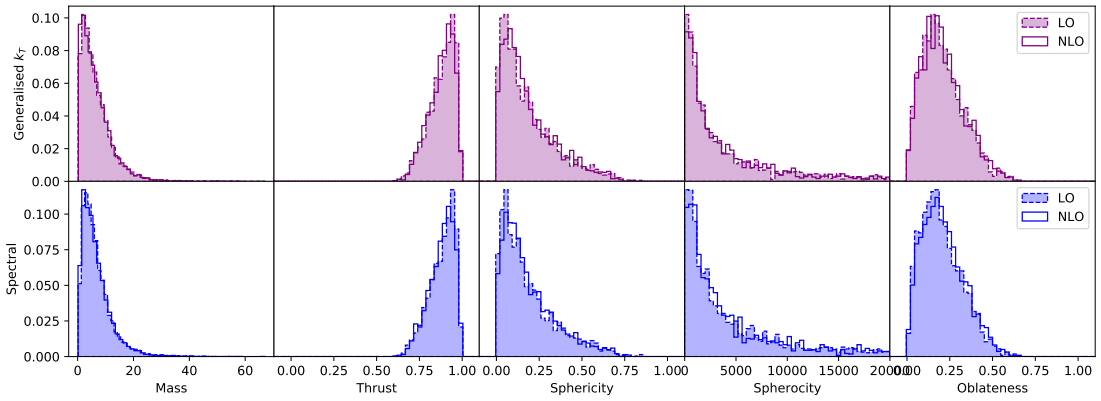


FIGURE 8.6: Spectra for jet properties created with LO and NLO datasets. The 4 jets with highest p_T from each event are used in aggregate as an average to form these plots. The columns from left to right are: the jet mass, thrust, sphericity, spherocity and oblateness. Algorithms were configured (i.e., the settings of R chosen) to give sensible results on this dataset, therefore distributions may not represent worst case scenarios.

However, this method of establishing IR sensitivity only looks at one parameter configuration and could be accused of cherry-picking. As described in section 8.2.5, this

can be systematically compared for many parameter configurations by calculating a Jensen-Shannon score for each LO and NLO pair of jet mass spectra. If the Jensen-Shannon metric is low, then the two distributions are similar and appear IR safe. To further clarify the result an algorithm known to be IR unsafe, the iterative cone algorithm, is included. The spectral method produces Jensen-Shannon scores very similar to generalised k_T methods. Only the iterative cone algorithm produces high Jensen-Shannon scores thus indicating significant changes between the LO and NLO spectra. This can be seen in Figure 8.7.

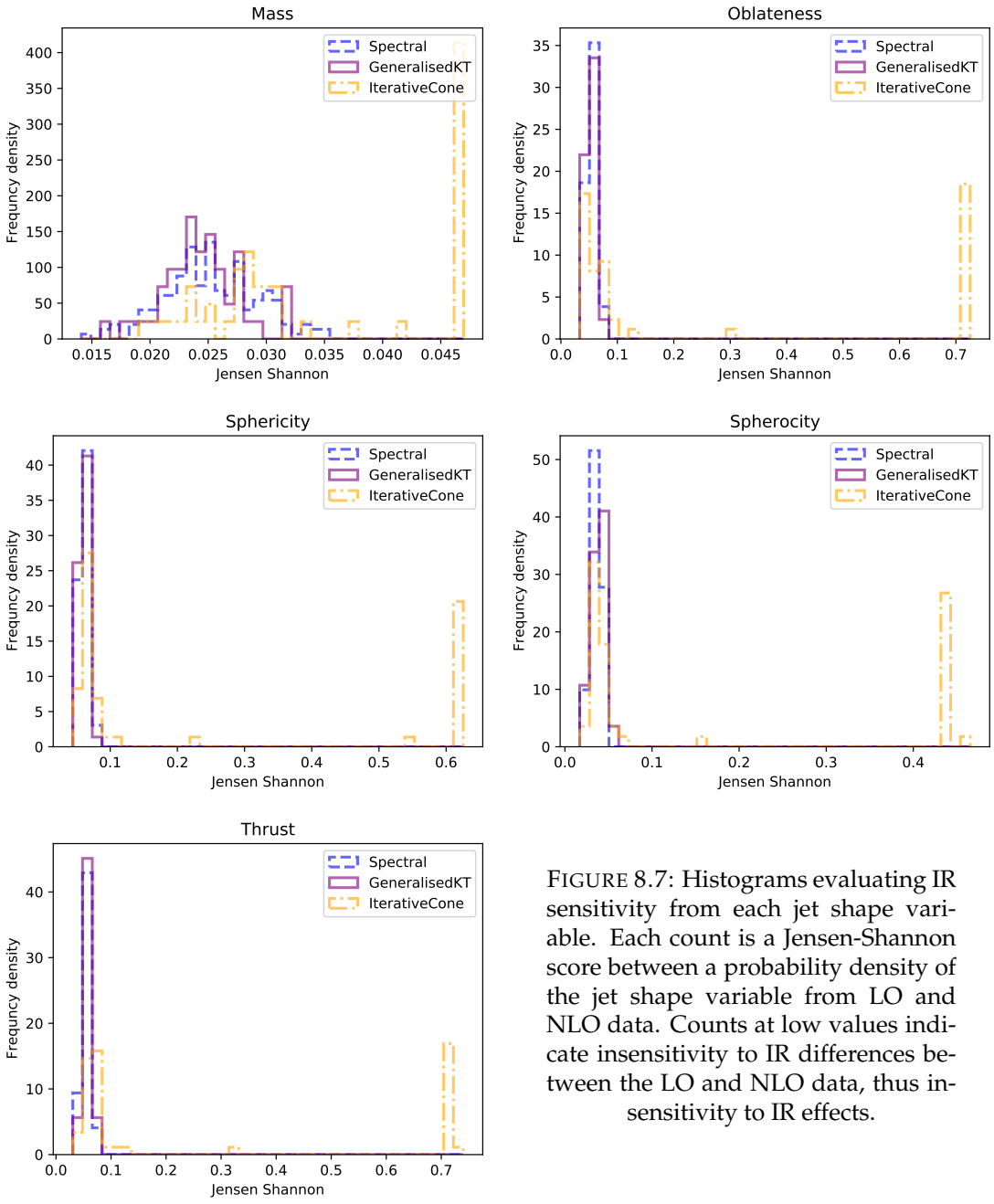


FIGURE 8.7: Histograms evaluating IR sensitivity from each jet shape variable. Each count is a Jensen-Shannon score between a probability density of the jet shape variable from LO and NLO data. Counts at low values indicate insensitivity to IR differences between the LO and NLO data, thus insensitivity to IR effects.

From these figures it is clear that spectral clustering is insensitive to IR effects in MC data, at least, as much as generalised k_T algorithms are. This contrasts with the iterative

cone algorithm, for which the jet mass spectra at LO and NLO differ significantly for many configurations. This is not unexpected, as the inputs to the spectral clustering algorithm are the same as for the Cambridge-Aachen one, which is itself IR safe, and the iterative cone has been proved to produce kinematic configurations which are IR unsafe [117]. However, it is crucial to have such a verification in data, as has been done.

8.3.2 Mass Peak Reconstruction

In this section, the anti-k_{T} algorithm setups with jet radius $R_{k_T} = 0.4$ and $R_{k_T} = 0.8$ are compared to the spectral algorithm specified in section 8.2.3. The jets are tagged using MC truth. Each of the b-quarks created by a signal particle (either a Higgs boson or a top (anti)quark) tag the closest jet (by using the distance metric $\sqrt{(y_{\text{quark tag}} - y_{\text{jet}})^2 + (\phi_{\text{quark tag}} - \phi_{\text{jet}})^2}$), provided that the separation between the jet and the quark is no greater than 0.8 according to the distance metric. In the case of a W decay, the procedure is the same applied to light quark states. From this point on, only jets tagged this way are considered.

Firstly, jet multiplicities, that is, the number of reconstructed jets found per event, are given for both the anti-k_{T} and spectral clustering algorithms. These can be seen for the first three datasets described in section 8.2.4 in Figure 8.8. Herein, it is seen that spectral clustering produces the best multiplicity (i.e., most events where 4 jets are found) for Light Higgs events while for the Heavy Higgs and Top MC samples it creates a multiplicity closer to that of anti-k_{T} with $R_{k_T} = 0.4$ than $R_{k_T} = 0.8$, the first of these being the best performer of the two. As a result of this study, we remark upon the adaptability of spectral clustering to the different final states without requiring adjusting its parameters, unlike the anti-k_{T} one. The latter may seem to indicate that 0.4 is the best choice for all datasets, but this is in tension with the fact that different masses from different datasets do require the anti-k_{T} algorithm to be adjusted, as will be seen in the mass peaks.

Mass peaks are constructed from the reconstructed jets as well as, for the top sample only, from the lepton and neutrino. Again, the anti-k_{T} results with $R_{k_T} = 0.4$ and 0.8 are given for comparison.

In Figure 8.9 three selections are plotted for the Light Higgs MC sample. Firstly, events here all four b-jets were found are combined for total invariant mass of the event, thus reconstructing the mass of the SM Higgs boson. Each event also contains two light Higgs states, though. These are differentiated by the mass of the particles (generated by them) that pass the particle cuts, as follows. The light Higgs boson reconstructed from the $2b$-jet system with more mass visible to the detector is called the "Light Higgs with stronger signal" while the one reconstructed with less mass visible in the detector is called the "Light Higgs with weaker signal". The correct jets for each Higgs mass reconstruction are identified using MC truth, so the correct pairings are always made. (If

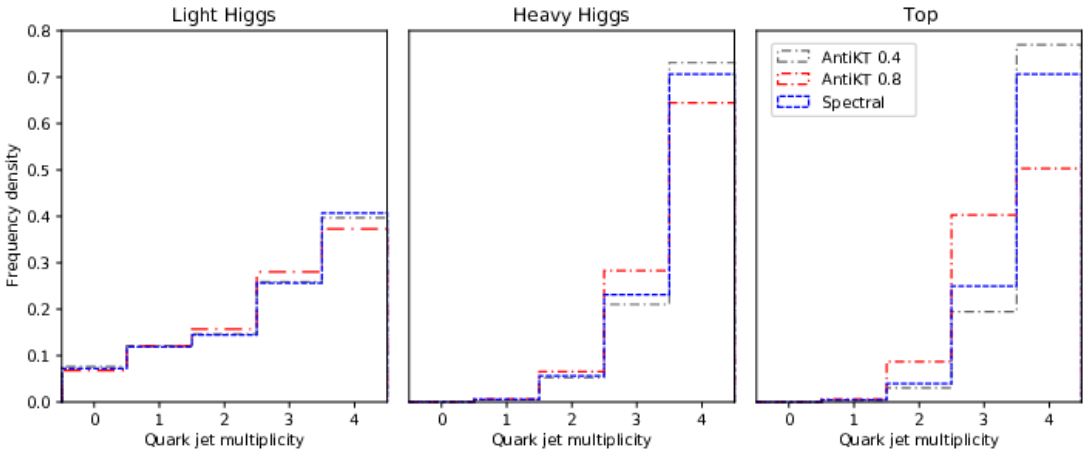


FIGURE 8.8: Jet multiplicities for the anti-k_T (for two jet radius choices) and spectral clustering algorithms on the Light Higgs, Heavy Higgs and Top MC samples. For all such datasets, the hard scattering produces 4 partons in the final state, so maximising a multiplicity of 4 jets indicates good performance.

two such dijet systems are not found the event is not included in the plots). Altogether, it can be seen that spectral clustering forms the sharpest peaks and such peaks are all very close to the correct mass. All peaks are shifted down (to the left) of the mass of the object reconstructed, this is because of the particle cuts described in section 8.2.4. These particle cuts replicate the way that some particles created by the decay in the hard event would never be detectable, and thus the reconstructed objects are missing a little mass. In fact, the performance of spectral is comparable to that of anti-k_T with jet radius 0.8 and is clearly better than the 0.4 option.

In Figure 8.10 the exercise is repeated for the Heavy Higgs MC dataset. All the parameters of spectral clustering are the same as in the Light Higgs MC sample yet we note that its performance is still excellent, with very sharp peaks at the correct masses, although the three clustering algorithms are overall much closer in performance. However, recall that, in Figure 8.8, it was seen that spectral clustering achieved better multiplicity than anti-k_T with $R_{k_T} = 0.8$ on this dataset. Furthermore, while the multiplicity of anti-k_T with $R_{k_T} = 0.4$ is a little better, the location of all Higgs mass peaks for anti-k_T with $R_{k_T} = 0.4$ is slightly worse. So, again it is concluded that spectral clustering is probably the best performer overall with the added benefit of not requiring any adjustment of its parameters to achieve this.

Finally, in Figure 8.11, the W and t mass peaks for semileptonic $t\bar{t}$ decays are shown. Three mass reconstructions are given. Firstly, the hadronic W is reconstructed from the jets that come from the quarks it decayed to. Correct decisions about which quarks correspond to which particle in the hard process are made by using information in the MC, this is to prevent any mismatching from causing additional complication in evaluating the performance of the clustering. To tag a jet with a quark a distance measure

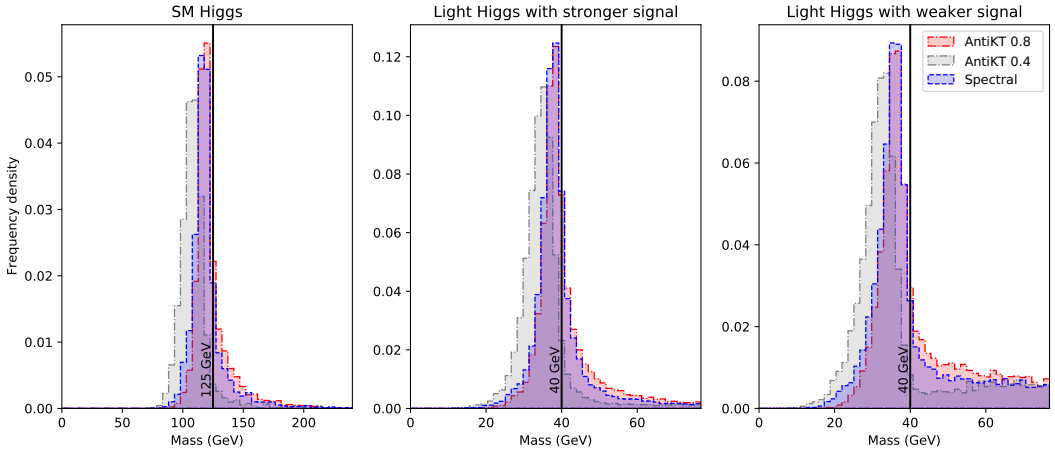


FIGURE 8.9: Three mass selections are plotted for the Light Higgs dataset. From left to right: the invariant mass of the $4b$-jet system, of the $2b$-jet system with heaviest invariant mass and of the $2b$-jet system with lightest invariant mass (as defined in the text). Three jet clustering combinations are plotted as detailed in the legend. The spectral clustering algorithm is consistently the best performer in terms of the narrowest peaks being reconstructed and comparable to anti-k_T with $R_{k_T} = 0.8$ in terms of their shift from the true Higgs mass values, with anti-k_T with $R_{k_T} = 0.4$ always being the outlier. For further discussion see section 8.3.2.

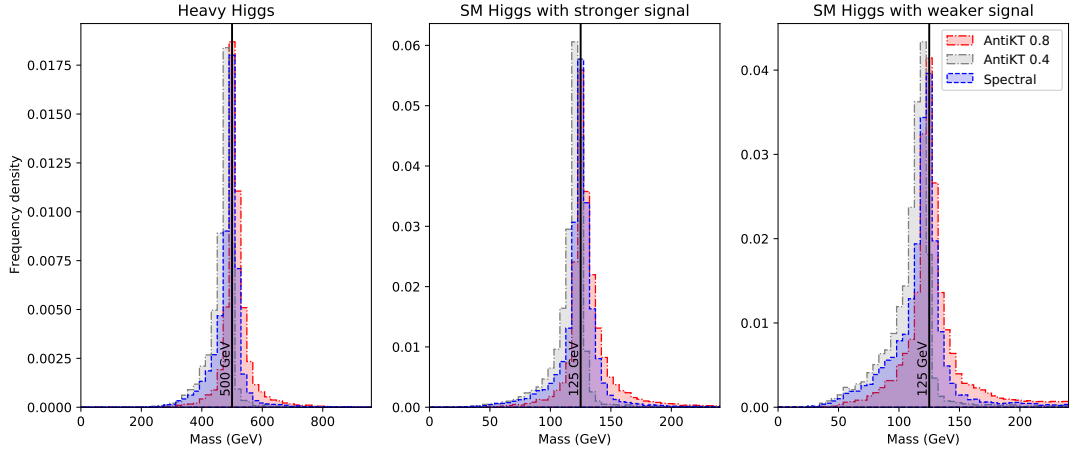


FIGURE 8.10: Same as Figure 8.9 for the Heavy Higgs dataset. Here, the performance of the spectral clustering and anti-k_T (with both 0.4 and 0.8 as jet radii) clustering algorithms is much closer to each other. For further discussion see section 8.3.2. Note that the scale here is significantly larger than in Figure 8.9, and so the mass lost due to particle cuts is too small to be visible.

$\sqrt{(y_{\text{quark tag}} - y_{\text{jet}})^2 + (\phi_{\text{quark tag}} - \phi_{\text{jet}})^2}$ is used and, if the distance from the quark to the closest jet is less than 0.8, that jet is tagged by that quark. The W will always decay to a pair of quarks, but both these quarks may be captured in one jet or separate jets. If either of the these quarks are too far away from the closest jet to tag it, that is $\sqrt{(y_{\text{quark tag}} - y_{\text{jet}})^2 + (\phi_{\text{quark tag}} - \phi_{\text{jet}})^2} > 0.8$, then it is not associated with any jet and the hadronic W is not reconstructed. The mass of the hadronic top is then reconstructed in events where the hadronic W could be reconstructed and the b-jet from the hadronic

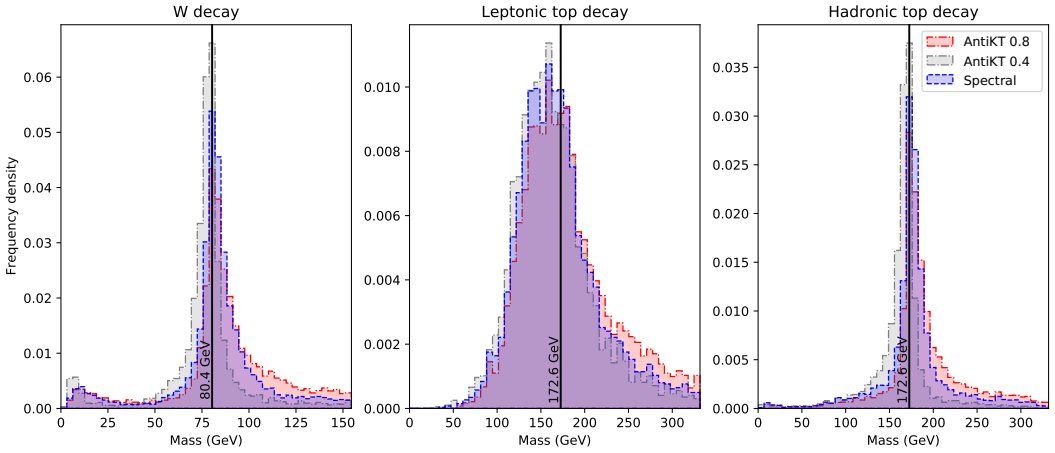


FIGURE 8.11: Three mass selections are plotted for the Top dataset. From left to right: the invariant mass of the light jet system, of the reconstructed leptonic W (as described in the text) combined with a b-jet and of the hadronic W combined with the other b-jet. Three jet clustering combinations are plotted as detailed in the legend. The spectral clustering algorithm consistently outperforms the anti-k_{T} one with jet radius 0.8 and is slightly worse than the anti-k_{T} one with jet radius 0.4, but only in terms of sharpness, not location. For further discussion see section 8.3.2.

top is also found. The leptonic top is then reconstructed in events where a b-jet from the top is combined with the reconstructed W that decayed leptonically. The leptonic reconstruction of the W uses the momentum of the electron p_ℓ, the missing transverse momentum p_T^{miss} (identified with that of the neutrino) and the longitudinal neutrino momentum (p_L^ν, which is unknown) in a quadratic equation, $(p_\ell + p_T^{\text{miss}} + p_L^\nu)^2 = m_W^2$, of which only the real solutions are plotted. In this case, it can be seen that spectral clustering is adapting to jets of a different radius. In fact, while before its behaviour had mostly resembled anti-k_T with $R_{k_T} = 0.8$, it has now moved closer to the case with $R_{k_T} = 0.4$. (Semileptonic top events would typically be processed using anti-k_T with $R_{k_T} = 0.4$.) The peaks of spectral clustering are not quite as narrow as those from anti-k_T with $R_{k_T} = 0.4$, but they improve on $R_{k_T} = 0.8$ and their location is substantially correct.

8.3.3 Run Time

Given the requirement for an eigenvalue calculation, an $\mathcal{O}(n^2)$ operation, it's clear that this algorithm will have longer run-times than the generalised k_{T} algorithm, which boasts $\mathcal{O}(n \log(n))$ [237]. The initial steps of the spectral algorithm require similar calculations to generalised k_{T}, so would be expected to have the same runtime. The implementation used in this work actually neglects the improvements that took generalised k_{T} from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log(n))$, so initial steps should run in $\mathcal{O}(n^2)$. Then the eigenvector calculation would add a further $\mathcal{O}(n^2)$. So with a naïve implementation, one would expect the spectral algorithm to require $\mathcal{O}(n^4)$.

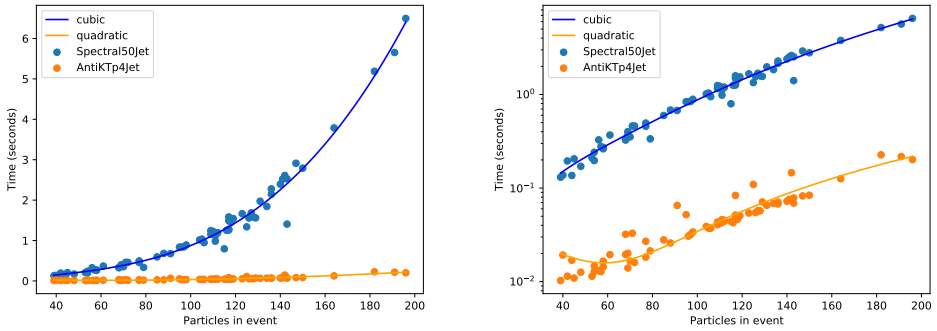


FIGURE 8.12: The run time of spectral, compared to a naïve implementation of generalised k_T (without the performance refinements in [237]), on datasets of varying size. Cubic and quadratic fits are shown for each dataset respectively. This shows that spectral runs in $\mathcal{O}(n^3)$.

This reasoning makes the results in Figure 8.12 a little surprising. In Figure 8.12, it is seen that spectral in fact runs in $\mathcal{O}(n^3)$, not $\mathcal{O}(n^4)$. No particular optimisations were used to achieve this, the implementation of spectral is a basic pythonic implementation of the algorithm set out in section 8.2.2. Specifically, no effort was made to take advantage of the sparse Laplacian matrix when performing the eigenvector calculation. In fact, if anything the implementation contains more branches than required, because it was designed to facilitate investigating variations, such as those shown in Figure 8.3.

So the improvements that render the algorithm $\mathcal{O}(n^3)$ rather than $\mathcal{O}(n^4)$ must be attributed to intelligently designed libraries. The eigenvector calculation was performed by `scipy`'s [258], function; `scipy.linalg.eigh`. This function optimises the calculation by using two criteria. Firstly, it requires the input be a Hermitian matrix, this holds for the Laplacian. Secondly, it allows the desired range of eigenvalues to be specified. Step 5, in section 8.2.2, determines that this spectral algorithm requires only the eigenvectors corresponding to a pre-specified range of eigenvalues. These two optimisations appear to buy an $\mathcal{O}(n)$ runtime improvement.

None the less, further improvements to the run time would be needed to render this a practical algorithm. That is outside the scope of this study.

8.4 Conclusions

Spectral clustering is a popular ML algorithm, wherein complex datasets are transformed to clarify groupings in a new space. In performing this transformation, it makes use of the spectrum (eigenvalues/eigenvectors) of the Laplacian matrix, which is constructed from localised information. At no point in the process are large matrices of learnt parameters, common to deep learning methods, needed. As such, spectral clustering is a transparent, simple to implement, algorithm using standard linear algebra methods. Owing to these features, this study has found it to be a promising new method for jet formation in high energy particle physics events.

For a start, it is robust to the IR distinctions between LO and NLO simulation and creates jets with the expected kinematics, as dictated by QCD dynamics. Furthermore, while it has many parameters, they do not appear to be as finely tuned as those of more standard tools, such as sequential (or iterative) generalised k_{T} algorithms. This can be seen in both parameter scan stability and its adaptability to various datasets, each capturing physics signals embedding heavy objects decaying into lighter ones in very different patterns, all yielding complicated hadronic signatures at the LHC.

The adaptability between datasets is remarkable as a spectral clustering parameter choice tuned on a light Higgs boson cascade gave excellent performance on both a heavy Higgs boson cascade and that of top-antitop pairs decaying semileptonically. In the case of the Light Higgs dataset, spectral clustering gave the correct mass peak positions, the narrowest resonant distributions and a jet multiplicity mapping well the partonic one. This would not be surprising as it was tuned for that dataset in the first place. In the case of the Heavy Higgs dataset only anti-k_{T} with $R_{k_T} = 0.8$ and the spectral algorithm gave correct mass peaks but spectral clustering offers considerably better multiplicity rates. This demonstrates that its performance is not dependent on fine tuning its parameters and hence that the algorithm is adaptable to the same final state with different masses involved. Finally, spectral clustering was applied to a Top dataset with a different final state and for which the ideal jet radius differed, semileptonic decays of top-antitop pairs. Its equivalent parameter σ_v was not allowed to vary to account for this, instead it was applied again with no parameter changes. The algorithm again proved to be adaptable and modified its behaviour to follow that of anti-k_{T} with $R_{k_T} = 0.4$, the standard choice for this kind of analyses.

In short, spectral clustering is a novel and promising approach to jet formation, which initial development already demonstrates flexibility and excellent performance for numerical analyses at the forefront of collider physics.

Chapter 9

Conclusions

Three studies are combined in this work. The first being a study that aimed to locate the available parameter space of the 2HDM where the signals of the cascade decays $A \to HZ$ and $H \to AZ$ would be detectable. The second was a comparative investigation, considering the relative merits of two jet formation algorithms, anti-k_{T} and variable-R, for finding jets produced in Higgs cascades. The third study experimented with the adoption of a novel jet formation algorithm, the spectral algorithm, with a particular interest in finding the jets produced in Higgs cascade decays. Together, they each contribute to an aspect of the search for 2HDM signals.

Chapter 3 contains the first study. It began with the results of ATLAS experimental analysis of the production and decay process $gg, b\bar{b} \to A \to ZH \to l^+l^-b\bar{b}$ performed at run 2. The data from this study was combined with predictions, and additional checks for exclusions, to establish the parameters that are not excluded, but for which there is experimental sensitivity. This process was extended in two ways, firstly, an alternative decay chain was considered; $gg, b\bar{b} \to H \to ZA \to l^+l^-b\bar{b}$. This additional decay chain did not yield any additional parameter space, as all mass combinations that would have left it kinematically open were excluded by either theoretical constraints, or flavour physics constraints. The second extension was a projection of the detector sensitivity to the conditions of run 3. This yielded more exciting results; for Yukawa types I, II and Y, and for moderate $\tan(\beta)$ values, those around 5 - 10, parameter space will be available. This is a promising window of opportunity for run 3 at the LHC.

The second study is in chapter 5. This study considered jet formation on another Higgs cascade decay; $H \to hh \to b\bar{b}b\bar{b}$. The four jet final state provides particular challenges for jet reconstruction, and would be common to many other 2HDM signal processes. Two variations of this decay were considered, one in which the H has a mass of 500 GeV and the h corresponded to the observed standard model Higgs. The other assigned H to be the standard model Higgs at 125 GeV and selected h to be a light Higgs at 60 GeV. The study illustrated that this second configuration would be challenging to spot, as

the light Higgs would not create a large shower, and typically have low transverse momentum. A significant fraction of this signal will be lost to kinematic cuts, even with excellent jet reconstruction. The reconstruction of both signals was considerably enhanced by choosing the unusual variable-R method, rather than the prevalent anti-k_{T} algorithm. Variable-R clustering was able to improve jet multiplicities, and mass peak locations, by use of its adaptability to differing jet p_T.

Finally, the study in chapter 8 investigated the use of spectral clustering for jet formation. This is a novel approach, which treats the particles to be clustered as nodes of a graph, then uses the simple and powerful features of the Laplacian eigenmap of the graph. This work demonstrates that a spectral algorithm accurately forms jets on a range of events, without requiring parameter adjustment. This was investigated using both Higgs cascade decays, and semileptonic $t\bar{t}$ decays. It is insensitive to the influence of differing IR effects between LO and NLO simulation. Furthermore, the algorithm has a limited number of parameters, each of which have clear intuitive meaning, something of a rarity in a ML technique. Overall, this algorithm shows great promise, and facilitates better jet formation techniques.

Appendix A

Replication Study of CSVv2 and DeepCSV

In order to investigate the practicalities of using ML in jet physics some replication studies were performed. The subject chosen was CVSv2 and DeepCSV, both of which are jet taggers which have been used in practice by the CMS collaboration [176]. Recreating the taggers using a different set of libraries offers verification of their expected behaviour, and explores the challenges of training ML tools in a realistic setting.

Finding the best configuration to address a specified problem may require training and evaluating many versions of a tool, particularly so for NNs like CSVv2 and DeepCSV. Therefore factors influencing the speed of training are of particular interest.

To begin with the origin and nature of the data used will be described. The preprocessing of the data, in particular the reweighting, will be discussed. The two NNs, CSVv2 and DeepCSV, are geometrically distinct. Both of them are described. The framework in which the NNs are trained is described, as this study is interested in factors that influence the speed with which they are trained. The training of NNs required various hyperparameters, the way in which these were selected is described, and some indication of the effects of varying them is given.

All of the code used to read and process the data, and perform and evaluate the training is available at `https://bitbucket.org/tidefall/jettagging`.

A.1 Input Data

The input data is MC simulated data designed to emulate possible observations at the LHC from the CMS detector. To be as realistic as possible, this data includes noise from

MPI, ISR and pileup. Detector precision is also simulated. For a more general overview of these processes, see section 4.5.

In order to train a NN labelled data is needed. As the data is generated in MC, the ground truth is known, and can be used for these labels.

The MC pipeline is as follows; GEANT 4 [104] is used to simulate interactions between particles and the detector material. POWHEG 2.0 [259] and MadGraph5_aMC@NLO 2.2.2 [71] are used to generate signal events. PYTHIA 8.2 [85] is used to simulate the background, parton showering and hadronizsation. Data produced in this way has been kindly provided by Dr Emmanuel Olaiya.

In this data sample the signal event used is a $t\bar{t}$ fully leptonic decay as depicted in Figure A.1.

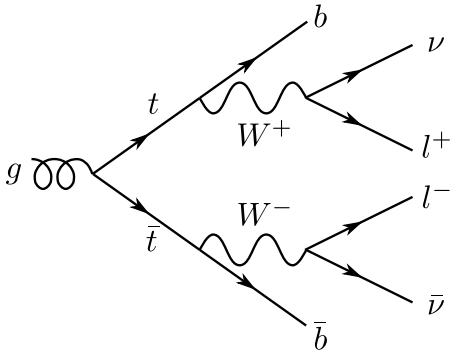


FIGURE A.1: The signal event found in the Monte Carlo data used. This is a decay of top quarks into bottom quarks and other products. The bottom quarks will then hadronize to form distinctive b jets. The other products are two neutrinos, which will appear as missing momentum in the event, and two leptons. If there is sufficient energy the leptons may be muons. Muons produce particularly distinctive tracks, and so are a great advantage in reconstructing events.

The input vector for both NNs is a set of high level variables. To obtain these high level variables the output of the simulation requires preprocessing. A global event reconstruction is performed to identify the particle tracks. Various cuts are applied to concentrate the data on the area of interest. These are as detailed in reference [176]. Then the anti-k_T jet clustering algorithm [237] with distance parameter $\Delta R = 0.4$ is applied to select the jets.

The reconstruction of jets has varying levels of success. On this bases the jets can be divided into 3 categories; [176, p. 15]

- **RecoVertex:** The jet contains one or more Secondary Vertices (SVs).
- **PseudoVertex:** No SVs, but the tracks fit these conditions;
    1. At least 2 tracks with a 2D Impact Parameter (IP) significance above 2. '2D' refers to the IP in the transverse plane.

2. The combined invariant mass of the tracks selected by the first condition is at least 50 MeV away from the K_S^0 mass.

- **NoVertex:** All jets that don't meet the requirements for either of the first two categories.

For the NN called CSVv2 only the best reconstructions (RecoVertex) were used in this study. Partial reconstructions are discarded. The NN called DeepCSV uses all reconstructions. As they are using different selections of the data the classification results of the two NNs are not directly comparable. DeepCSV is at a disadvantage because it will be used to classify all events, including those whose reconstruction did not meet the RecoVertex level. 19 high level variables are used, which can also be found here [176, p. 15]. They are listed here, following the target variable.

1. The target variable **parton flavour**. Partons generated by the simulation that are within $\Delta R < 0.3$ of the jet axis may give the jet flavour. If there is more than one candidate then the order of preference is b, c then light partons ('udsg') [260, p. 3].

2. **SV 2D flight distance significance:** The SV flight distance is the distance between the primary and secondary vertices. '2D' refers to movement in the transverse plane. This variable is recorded per jet and if there are multiple SVs in the jet then the one with the smallest uncertainty in flight distance is used.

3. **Number of SVs:** The number of SVs found in the jet.

4. **Track $\eta_{\mathbf{rel}}$:** This is the pseudorapidity (see section 4.5) of the track relative to the jet axis. It is recorded per track and the track with the smallest uncertainty in flight distance is used.

5. **Corrected SV mass:** This variable is defined per jet and its definition depends on the jet category.

   In the RecoVertex category it is the corrected mass of the secondary vertex with the smallest uncertainty in flight distance. The correction is "$\sqrt{M_{\mathrm{SV}}^2 + p^2 \sin^2\theta} + p\sin\theta$, where M_{SV}^2 is the invariant mass of the tracks associated with the SV, p is the SV momentum obtained from the tracks associated with it and θ is the angle between the secondary vertex momentum and the vector pointing from the primary vertex to the secondary vertex." [176, p. 10]

   In the PseudoVertex category this variable is the invariant mass obtained from the total summed four momentum vector of all the tracks in the jet.

   In the NoVertex category this variable is not used.

6. **Number of tracks from the SV:** This variable is defined per jet and its definition depends on the jet category.

   In the RecoVertex category it is the number of tracks associated with with the smallest uncertainty in flight distance.

   In the PseudoVertex category it is the total number of tracks in the jet.

   In the NoVertex category this variable is not used.

7. **SV energy ratio:** this is the energy of the SV with the smallest uncertainty on its flight distance divided by the combined energy of all the tracks in the jet.

8. $\Delta R(\mathbf{SV}, \mathbf{jet})$**:** This is a measure of angular separation in η-ϕ space. It is defined per jet and its definition depends on the jet category.

   In the RecoVertex category it is the angular distance between the jet axis and the SV with with the smallest uncertainty in flight distance.

   In the PseudoVertex category it is the angular distance between the jet axis and the summed four momentum of all the tracks in the jet.

   In the NoVertex category this variable is not used.

9. **Track 3D IP significance:** This variable is defined per track. The values of the four tracks with the highest 2D IP significance are used, so this variable provides 4 values as input data.

10. **Track** $p_{\mathbf{T,\,rel}}$**:** This is the track momentum perpendicular to the jet axis, p_{T} relative to the jet axis. The variable is defined per track and the track with the highest 2D IP significance will be chosen as an input value.

11. $\Delta R(\mathbf{track}, \mathbf{jet})$**:** This is a measure of angular separation in η-ϕ space (see section 4.5). It is the angular separation between the track and the jet axis. The variable is defined per track and the track with the highest 2D IP significance will be chosen as an input value.

12. **Track** $p_{\mathbf{T,\,rel}}$ **ratio:** This is the track momentum perpendicular to the jet axis divided by the magnitude of the track momentum. The variable is defined per track and the track with the highest 2D IP significance will be chosen as an input value.

13. **Track distance:** This variable is defined per track, it is the minimal distance between the track and the jet axis. The track with the highest 2D IP significance will be chosen as an input value.

14. **Track decay length:** This variable is defined per track, it is the distance between the primary vertex and the point of closest approach between the track and the jet axis. Note the somewhat unusual definition. The track with the highest 2D IP significance will be chosen as an input value.

15. **Summed tracks $E_{\mathbf{T}}$ ratio:** This variable is defined per jet. It is the summed transverse energy of all tracks in the jet divided by the transverse energy of the jet.

16. ΔR(**summed tracks**, **jet**)**:** This is a measure of angular separation in η-ϕ space. It is defined per jet and it is the angular separation between the summed for momentum of the tracks and the jet axis.

17. **First track 2D IP significance above c threshold:** This variable is defined per jet. It is the 2D IP significance of a chosen track. The track is chosen by adding the four momenta of the tracks in order of least uncertainty in flight distance until the combined four momentum vector has a mass greater than 1.5 GeV. The last track added, the one that pushed the sum over this threshold, is chosen. The value of 1.5 GeV comes from the mass of the c quark.

18. **Number of selected tracks:** This is the number of tracks in the jet.

19. **Jet** $p_{\mathbf{T}}$

20. **Jet** η

21. The discriminating variable **track 2D IP significance:** This variable is defined per track. It is used to sort the tracks and in deciding the jet category.

The distributions of these variables are shifted to centre about 0 and rescaled so that the standard deviation is 1. Then all unreconstructed values are set to 0.

Some of the variables that are used are likely to exhibit a strong dependence on the jet mass. After the classification is complete, the identified signal is often used to study a mass spectrum, and so it is very important that the performance of the classifier is not correlated with jet mass.

On a practical level this means reweighting the samples produced from Monte Carlo to ensure that signal dependent variables have the same distribution between flavours. That is, when the NN is being trained variables will be selected at random from the training data (Monte Carlo labelled data). The distribution of a variable is then the distribution of the values that belong to all the randomly selected jets. So if the random distribution from which the jets are drawn is not confined to be uniform it can be used to shape the distribution of variables seen in the selected jets. Changes to this distribution are parametrised by 'weights' on each jet. The higher the weight of a given jet the more likely it is to be selected. The aim is then to find a set of weights such that when jets are drawn at random and then split by flavour the distribution of signal dependent variables looks identical for each flavour. This prevents the classifier from directly discriminating on those variables, which would result in a decision correlated with jet mass and the signal process.

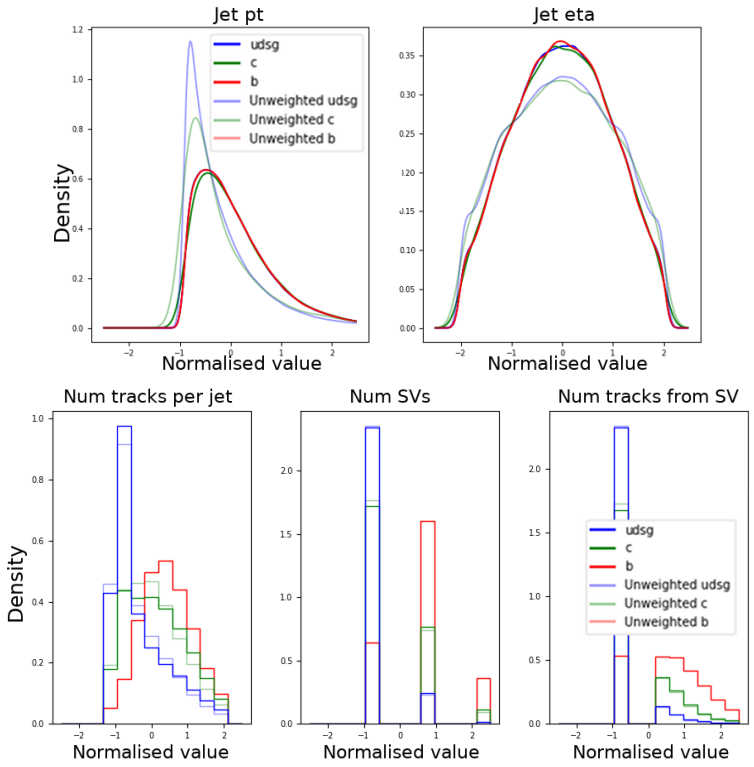


FIGURE A.2: The distribution of variables jet p_T, jet η, number of selected tracks, number of SVs and number of tracks from secondary vertices. They have been normalised and are plotted before and after reweighting.

The correlations between variables are not lost in the reweighting, and a NN is able to use these in its classification. However, the reweighting may warp useful physical properties of the training sample. All distributions will be effected by the reweighting, not just those signal dependant variables for which it is needed. This changes may make it more difficult for the NN to identify flavours.

Two variables that are known to depend on the signal event, and not the jet flavour, are the jet p_T and the jet η. This is because their energy scale corresponds to that of the signal event and not that of a quark mass. Simultaneous reweighting of two variables presents a challenge. This is because changes in weight that correct the distribution of one variable will alter the distributions of all other variables. An iterative algorithm can be used to improve the distributions each in turn, until they are within a suitable margin of each other. Here, the algorithm described in [261][1] has been used to match the distributions.

The results of the preprocessing and reweighting can been seen in Figure A.2 to Figure A.4. In figure A.2 it can be seen that although the reweighted distributions of jet p_T and jet η are not quite identical they are close enough to be within the level of the natural variations in the data. For the most part this reweighting has had only a marginal

[1] Available at [262].

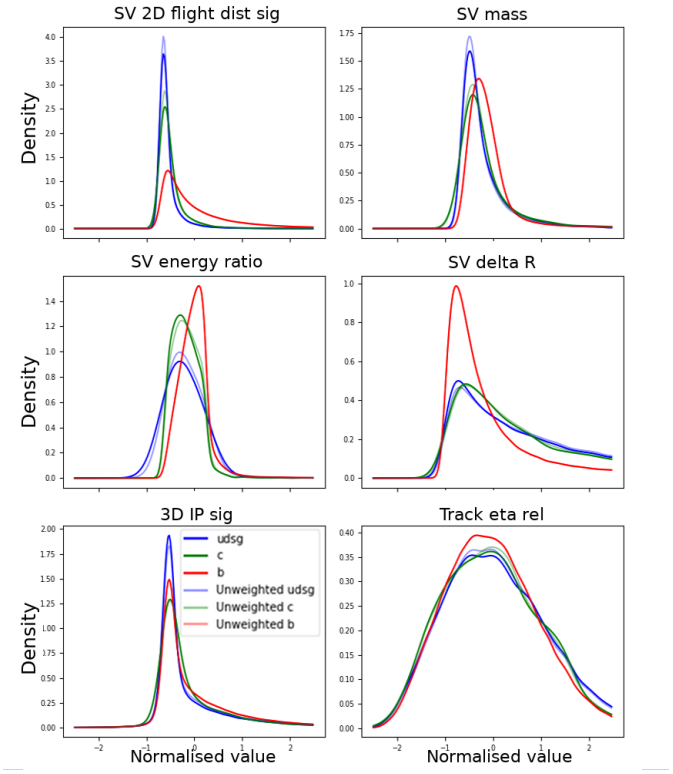


FIGURE A.3: The distribution of jet variables secondary vertex 2D flight distance significance, corrected SV mass SV energy ratio, $\Delta R(\text{SV}, \text{jet})$, track 3D IP significance and track η_{rel}. They have been normalised and are plotted before and after reweighting.

impact on the other distributions, however in figures A.3 and A.4 it can bee seen that the SV energy ratio and the summed tracks E_{T} ratio are both somewhat affected. This could reduce the efficiency of the jet tagger.

A.2 NN Architectures

There are two NNs considered in this study. The first is called CSVv2, and the second is called DeepCSV. They were both designed by the CMS collaboration to tag b and c jets during the second (13 TeV) run of the LHC [176].

CSVv2 is the smaller of the two NNs. It is shown graphically in Figure A.5. It was superseded by DeepCSV. DeepCSV is a larger NN that is shown in Figure A.6. This NN is the larger of the two, its size will increase the compute time required to train it.

Both of the networks are fully connected, feed forward, neural networks. The training process itself has a number of hyperparameters. These do not appear in the finalised network, but will change the time required to train the network, and the maximum performance achieved. Three key hyperparameters are batch size, learning rate and weight decay.

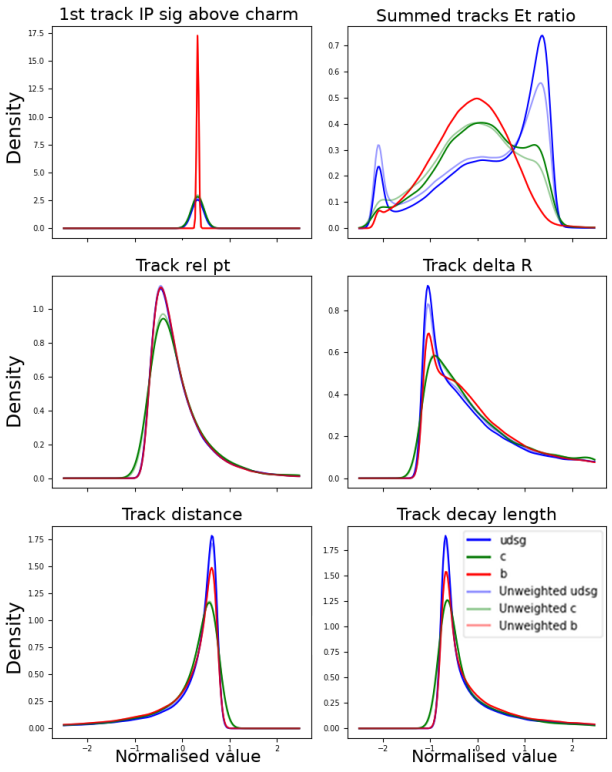


FIGURE A.4: The distribution of jet variables first track 2D IP significance above c threshold, summed tracks E_T ratio, track $p_{T,rel}$, $\Delta R(\text{track}, \text{jet})$, track distance and track decay length. They have been normalised and are plotted before and after reweighting.

To begin with, the batch size; when adjusting the parameters of the NNs, back propagation can be used to find a direction that would improve the output for each individual event. This is a generic optimisation problem. It is also possible to consider and average distance that would improve the performance on a batch of events. Larger batches enable more vectorisation in the process, and often speed up the process. However, smaller batches introduce a stochastic element to the process, which can help the training process escape local minima, therefore avoid getting stuck. Batch sizes between 400 and 20000 where tested; for both CSVv2 and DeepCSV a batch size around 5000 was found to be optimal.

Moving on, the learning rate; when the optimiser is working on the NN the magnitude of the changes made with each batch is adjusted by the learning rate. When the NN is far from the minima a high learning rate will improve the speed of the training. When the NN gets close to the minima, if the learning rate is too high then the optimiser may over adjust the NN and miss the minima. The batch size also has an influence on the rate at which the NN is changing, and so influences the ideal learning rate.

Unlike the other hyperparameters, learning rate will correct itself to some degree; as the NN trains it uses an adaptive learning rate scheduler to chose an optimum learning

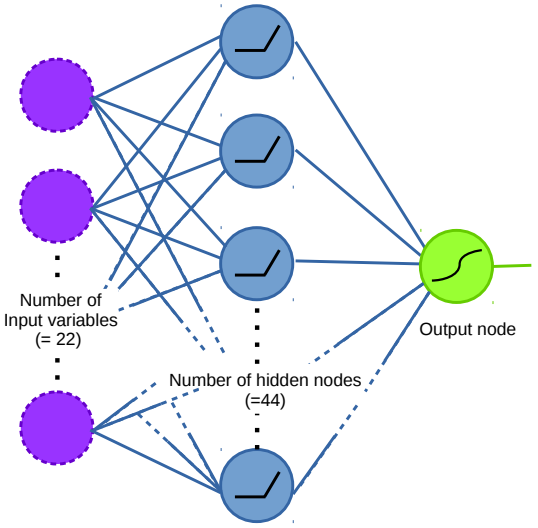


FIGURE A.5: The topology of CSVv2. There are 22 input nodes, one hidden layer with 44 hidden nodes and one output node. The activation function of the nodes in the hidden layer is ReLU, and the activation function of the output node is a sigmoid. The output node indicates a degree of belief that the input jet is a b jet.

rate. At the end of each epoch the scheduler is called, if the loss over time of the NN appears to have plateaued then the scheduler will reduce the learning rate.

For this reason it is only necessary to pick acceptable starting values of the learning rate, if they are poor choices they will be fixed soon after the start of the training. A good initial learning rate for CSVv2 is 0.1. A good initial learning rate for DeepCSV is 0.01.

Finally, the weight decay; Overfitting is a problem that is encountered when the NN has become too complex for the problem, and is fitting noise instead of the signal distribution. As the noise is different in the test and the train datasets, the effect is that although the performance of the NN appears to be improving in the training dataset, the performance in the test dataset will degrade. To prevent this, measures are taken to limit the complexity of the NN. There are a number of ways to achieve this; reducing the number of nodes in the hidden layers will reduce the complexity, however, this is only well correlated to the complexity of the NN when few neurons are used [200]. The overall complexity of a NN remains proportionate to the magnitude of the weights of the NN, no matter how many hidden nodes it has. Weight decay utilises this by adding an L_2 penalty to the optimiser that increases with the magnitude of the weights. The optimiser will be penalised for making the weights larger as this increases the complexity of the NN and therefore its opportunities to overfit.

It has also been observed that NNs with some weight decay train faster, even if the problem does not require it.

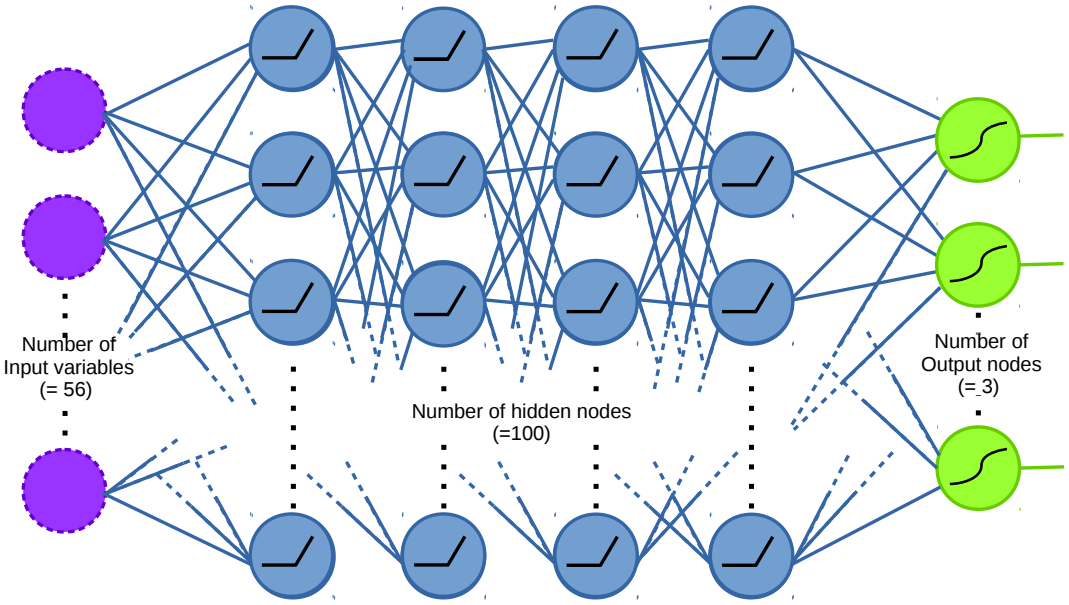


FIGURE A.6: The topology of DeepCSV. There are 56 input nodes, 4 hidden layers with 100 hidden nodes and one output node. The activation function of the nodes in the hidden layers is ReLU, and the activation function of the output node is a sigmoid. The first output node indicates the degree of belief that the output is a b jet, the second corresponds to c jets and the third to light jets (udsg). Note that in the original version of DeepCSV there were 5 output nodes, the two additional nodes indicated fat jets, or collimated jets. There are none of these jets in the data sample used in this paper, so those outputs were removed.

The weight decay has one parameter, this scales the penalty from weight decay relative to the loss as seen by the optimiser. A range of values from 2×10^{-7} to 0.0005 were tried, and for CSVv2 a value of 0.0002 achieved the right complexity, while for DeepCSV a value of 10^{-6} performed best.

Using these training hyperparameters, both NNs were trained on the same MC data sample.

A.3 Results

The output of the trained networks is compared to those trained by CMS in Figure A.7. While the general distribution of the response is similar, the performance of the replicas in this study falls short of those trained by CMS. The relative rate of error between jets with different ground truth flavours matches well between versions, *c*-jets are more likely than light jets to be mistaken for *b*-jets.

There are more factors that influence training progress than the three hyperparameters discussed in section A.2. Other factors include;

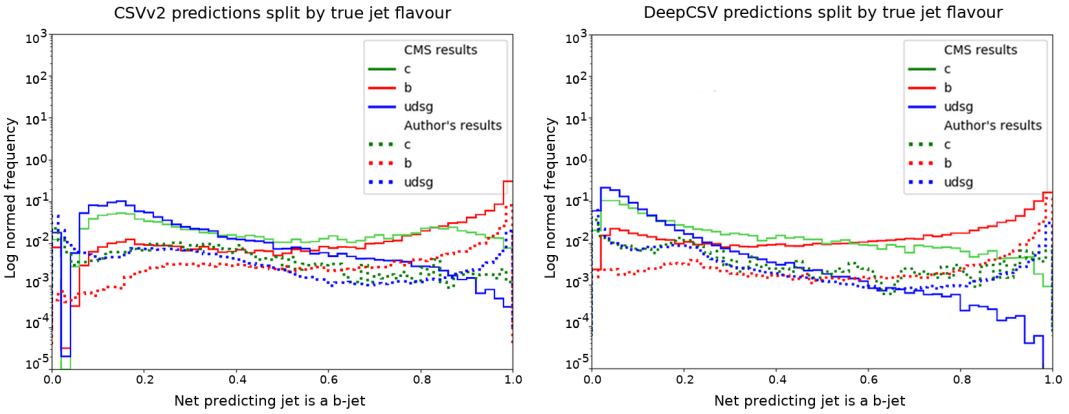


FIGURE A.7: The behaviour of the trained CSVv2 and DeepCSV replicas, as compared to the originals trained by the CMS collaboration. The author's NN produces a similar output, but does not perform as well as the NN trained by the CMS collaboration. For discussion of this see section A.3.

- The exact loss function used by CMS in training is not noted in the publication. A better choice of loss function might be less susceptible to false plateaus in training. This includes the shape of the weight penalty term.
- In a similar vein, the distribution from which the events where drawn, or the weights of the events, in other words, might have been better chosen in the CMS study. This study simply sought to prevent the NNs from becoming correlated with jet mass; more advances techniques such as bootstrapping might have been used to improve the performance of the NNs on the hardest events.
- The optimiser chosen to control the updates will have an impact.

Further investigation would be needed to uncover the exact process used be CMS to achieve their performance.

A.3.1 Time Required to Train

It is also interesting to know the total time required to train a NN using different computational resources. Training time is compared on three different cards[2];

- Nvidia Tesla V100 enterprise graphics cards, released June 21st, 2017. These are currently retailing for approximately £8000.
- Nvidia GTX 1080Ti consumer graphics cards, released March 10th 2017. These are currently retailing for approximately £600.
- Intel Xeon Gold 6138 CPU card, July 11th, 2017. These are currently retailing for approximately £2000.

[2] All recent cards at the time of testing; September 2018

Prices will only be accurate to about ±20% as they fluctuate and can be altered by purchasing at different points in the supply chain. For example the Nvidia Tesla v100 used in this study was recently acquired by the University of Southampton for £7580 and the GeForce GTX 1080Ti was acquired for £679.

This test requires identifying the time for completion of training. Several candidates were tested for this. The most successful was a ratio of standard deviation of the loss function, comparing the points before each time step to the points after that time step. This quantity is referred to as the standard deviation ratio;

$$R_\sigma(t) = \frac{\sigma(l(t'|t' > t))}{\sigma(l(t''|t'' < t))}. \tag{A.1}$$

This was chosen based on the principle that while the NN is training the output of the loss function will be varying rapidly, and so its standard deviation will be high. When training finishes the loss function will not be static, some stochastic movement is expected, but it will not move far from its minimum and it will obtain a low standard deviation. This minimum is described as the plateau. Thus if the graph of the loss is split in two at some epoch, and the standard deviation of all epochs ahead is divided by the standard deviation of all epoch behind, this ratio will reach a minima at the epoch when loss reaches a plateau. The minima of $R_\sigma(t)$ is then selected as the epoch when the training was completed.

This is plotted for CSVv2 in Figure A.8. It can be seen that the NN is reliably trained within 300 seconds regardless of the card used. The same plot for DeepCSV is presented in Figure A.9. Here the considerable advantage of GPU cards can be seen. When the NN is trained on the GPU card it is completed on a similar time scale as for CSVv2, however, training on the CPU takes about 7 times as long.

These numbers highlight something else quite unexpected; the considerably more expensive Tesla v100 (retailing at around £8000) is outperformed by the GTX 1080 Ti (£600) in this task.

The GTX 1080 Ti benefits from having good serial and parallel capacity. If larger NNs such as deep flavour had been included in this study it is possible that the Tesla v100 would have been able to better display its capacity for vectorised computation, however there are other aspects at play. The speed of the two cards is informally compared in [263, 264], and in both cases the Tesla v100 is found to be only marginally faster of the two. It is also speculated that the underperformance of the Tesla v100 is in part due to the relative immaturity of the software tailoring for the Tesla v100. The Tesla v100 was released in June, 2017 and while the GeForce GTX 1080 Ti was released March 2017 it has much in common with the GeForce GTX 1080, released May 2016. Furthermore, due to their more attainable price-point, it would be understandable if the GTX 1080 Ti had benefited from more development attention.

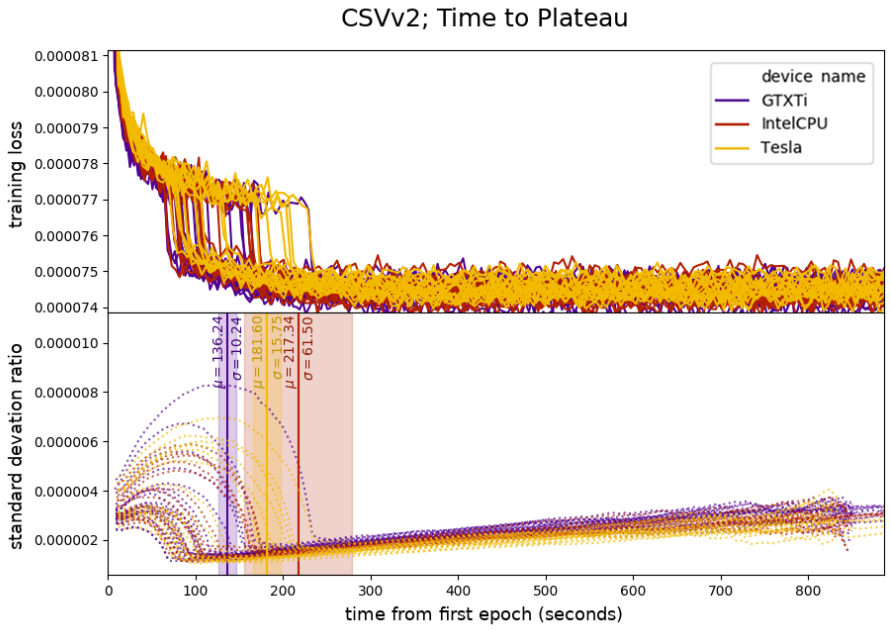


FIGURE A.8: The time for CSVv2 to reach a plateau. The upper plot is the training loss against time in seconds for the training of many NNs. When the loss stops descending and plateaus the NN is no longer learning. The colours are split by the card that the NN was trained on. The lower plot is the ratio of training loss standard deviation before and after each point. This ratio reaches a minimum when the NN first plateaus, see section A.3.1 for an explanation. The plateau time, and standard error are marked on the lower plot as vertical bars. It can be seen that the time to plateau is close for all 3 cards when training CSVv2.

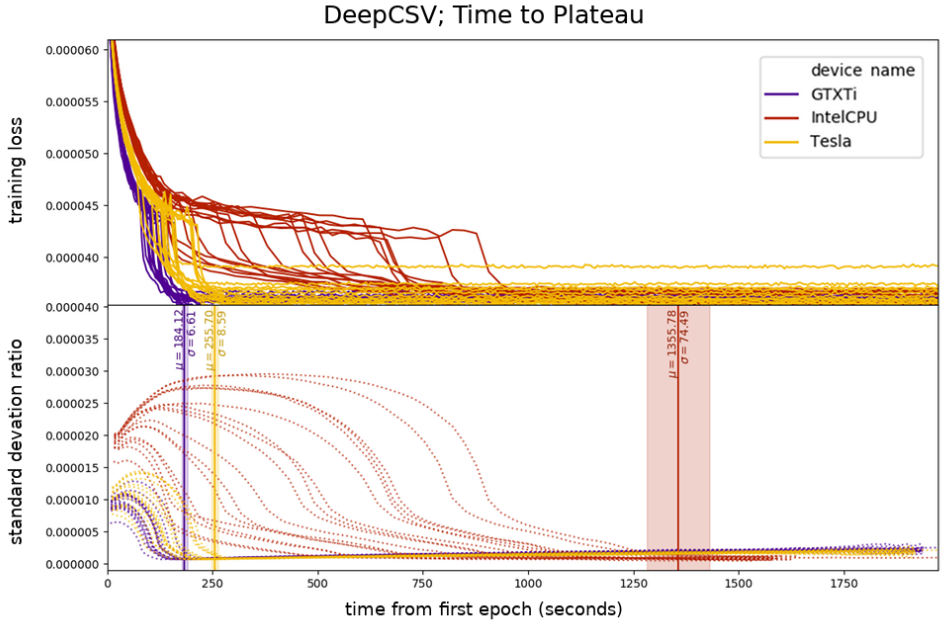


FIGURE A.9: The time for DeepCSV to reach a plateau. The plots are as in Figure A.8. It can be seen here that the GPUs confer a significant advantage, reaching plateau approximately 7 times faster than the CPU.

Comparing between Figure A.8 and Figure A.9, it is clear that the more parameters the network has to train, the greater the advantage gained by the graphics cards over the CPU.

A.4 Conclusions

From this replication study, two points should be noted. Firstly, the performance of a NN may be quite sensitive to its training conditions. Achieving the same results requires replicating not just the architecture, but also the training input distribution, optimiser and loss function.

Secondly, although it is no surprise that a GPU can train the network faster than a CPU, it is surprising that the cheaper of the two GPUs tested proved to be the fastest option. This highlights the importance of the software maturity in achieving the best speeds.

Appendix B

Two Cluster Spectral Clustering

Spectral clustering allows a slight optimisation when minimising the ratio cut criteria for the simplest, non-trivial, case of two clusters. Only a single indicator vector is needed; this is given in [247], and will be shown in detail here.

Let the first chosen cluster be K, and the second be $\bar{K}$, which is naturally equivalent to all points not in K. In order to record the allocation of points to K or $\bar{K}$, an indicator vector can be defined such that

$$f_i = \begin{cases} \sqrt{|\bar{K}|/|K|}, & \text{if } i \in K \\ -\sqrt{|K|/|\bar{K}|}, & \text{if } i \in \bar{K}. \end{cases} \tag{B.1}$$

The target is to find an equation that constructs f_i such that the ratio cut, Equation 7.13, is minimised. The keystone to this will be the unnormalised graph Laplacian; let the unnormalised graph Laplacian be

$$L = D - A \tag{B.2}$$

where A is a matrix of affinities, such that $A_{i,j} = a_{i,j}$ and $A_{i,i} = 0$, and D is a diagonal matrix with $D_{i,i} = \sum_j a_{i,j}$.

Now a link can be drawn by taking the product of these things;

$$\begin{aligned} f'Lf =& \sum_{i,j} f_i L_{i,j} f_j \\ =& \sum_{i,j} f_i \left(\delta_{i,j} \sum_p a_{i,p} - a_{i,j} \right) f_j \\ =& \sum_i \left(f_i^2 \sum_p a_{i,p} - \sum_j f_i f_j a_{i,j} \right) \\ =& \sum_{i,j} a_{i,j} \left(f_i^2 - f_i f_j \right) \\ =& \frac{1}{2} \sum_{i,j} a_{i,j} \left(f_i - f_j \right)^2 \end{aligned} \tag{B.3}$$

which uses $a_{i,i} = 0$ and Equation B.2 between lines one and two. Then using Equation B.1, substitutions can be made for f_i and f_j. If i and j are both in the same cluster then this leads to $f_i - f_j = 0$, so we only need to consider $i \in K, j \in \bar{K}$ or $i \in \bar{K}, j \in K$.

$$\begin{aligned} f'Lf =& \frac{1}{2} \sum_{i \in K, j \in \bar{K}} a_{i,j} \left(\sqrt{\frac{|\bar{K}|}{|K|}} + \sqrt{\frac{|K|}{|\bar{K}|}} \right)^2 + \frac{1}{2} \sum_{i \in \bar{K}, j \in K} a_{i,j} \left(-\sqrt{\frac{|K|}{|\bar{K}|}} - \sqrt{\frac{|\bar{K}|}{|K|}} \right)^2 \\ =& \sum_{i \in K, j \in \bar{K}} a_{i,j} \left(\frac{|\bar{K}|}{|K|} + \frac{|K|}{|\bar{K}|} + 2 \right) \end{aligned} \tag{B.4}$$

Now notice that the first term of this sum is the numerator from Equation 7.13. The remaining terms do not depend on the index.

$$\begin{aligned} f'Lf =& (|\bar{K}| + |K|) \left(\frac{\sum_{i \in K, j \in \bar{K}} a_{i,j}}{|K|} + \frac{\sum_{i \in K, j \in \bar{K}} a_{i,j}}{|\bar{K}|} \right) \\ =& (|\bar{K}| + |K|) \, \text{RatioCut} \end{aligned} \tag{B.5}$$

There is another use that can be made of Equation B.2; $\sum_i f_i^2 = |K| \sqrt{|\bar{K}|/|K|}^2 + |\bar{K}| \left(-\sqrt{|K|/|\bar{K}|} \right)^2 = |\bar{K}| + |K|$. Which takes the equation to;

$$\text{RatioCut} = \frac{f'Lf}{f'f} \tag{B.6}$$

So the cost function can be expressed neatly in terms of the indicator vectors, f, and the Laplacian, L. The aim is to minimise this. This is still not directly solvable, not with the requirements of Equation B.2. But if those requirements are relaxed, and f_i can take any values, provided that they are perpendicular to $\mathbb{1}$. then the right hand side becomes the Rayleigh-Ritz quotient. Minimising this is done by finding eigenvector associated with the smallest eigenvalue.

Appendix C

Stopping Condition

To offer some evidence for the assertions made in section 8.2.1.3, the behaviour of the mean distance during clustering is shown in Figure C.1.

Clustering is performed on the dataset described in section 8.2.4 called Light Higgs. The parameters used for the spectral algorithm are the ones given at the end of section 8.2.3. First, the upper panel of Figure C.1 shows the mean distance between pseudojets for 2000 events, plotted against the number of pseudojets remaining. Each line is shown in yellow until its value first exceeds $R = 1.26$, the stopping condition, after which the line becomes green. When finding jets with spectral clustering, the algorithm would normally be stopped at the end of the yellow section, as the stopping condition has been reached, the green section is shown here to illustrate what happens beyond this point. It can be seen that the transition from yellow to green happens with approximately 3 to 13 pseudojets remaining. This supports the assertion that a mean distance stopping condition will not force the same number of jets in each event. It can also be seen that the mean distance does rise smoothly for most of the clustering sequence, becoming erratic only when less than 5 pseudojets remain.

Second, in the lower panel, the factors that alter the mean distance are plotted. Again, each of the 2000 events is represented as a single solid line. In blue, change of mean distance due to merging pseudojets is shown. Normally merging two pseudojets causes the mean distance to rise, as the embedding space is becoming sparser, however, there are some configurations in which this does not hold. Occasionally, two points that merge will lower the mean distance, and the blue line will dip below zero. It can be seen from the plot that such configurations are less common than those that increase mean distance.

The second panel also shows change of mean distance due to a reduction in the number of dimensions in the embedding space in red. This universally decreases mean distance, the red lines remain below or at zero. Not every step of the algorithm will reduce the number of dimensions, and so the red line for an event is frequently zero.

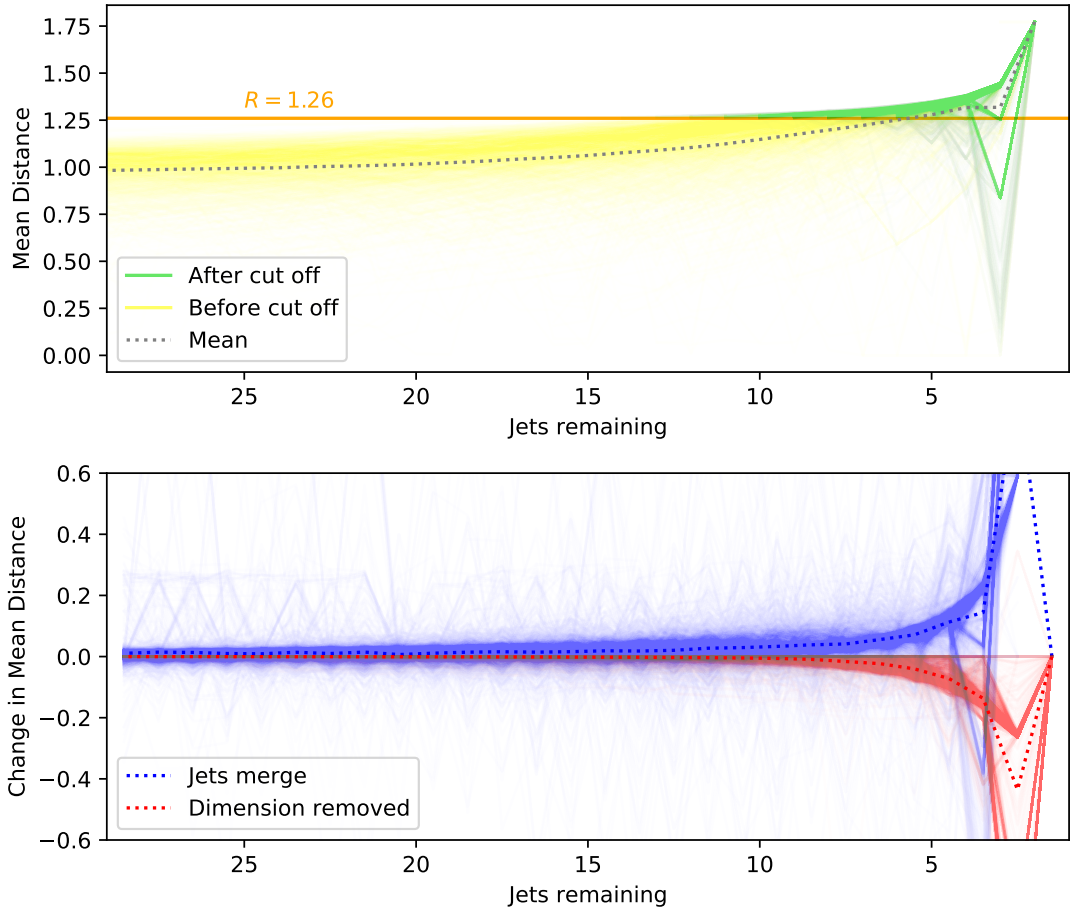


FIGURE C.1: In the upper panel, the mean distance between pseudojets for 2000 events is plotted against the number of pseudojets remaining. Each line is shown in yellow until its value first exceeds $R = 1.26$, the stopping condition, after which the line becomes green. A dotted line shows the average mean distance across all 2000 events. In the lower panel, the factors that alter the mean distance are plotted. Again, each of the 2000 events is represented as a single line, and the average is given as a dotted line. In blue, change of mean distance due to merging pseudojets is shown. In red, change of mean distance due to a reduction in the number of dimensions in the embedding space is shown.

It can be seen that these two factors balance each other to produce a steady trend in mean distance.

There is a third possibility, very rarely the number of dimensions in the embedding space will increase. This is not pictured, as it is not possible to visually distinguish the line from $y = 0$ and it would clutter the plot.

Bibliography

[1] Henry Day-Hall et al. "Mapping $pp \to A \to ZH \to l^+l^-b\bar{b}$ and $pp \to H \to ZA \to l^+l^-b\bar{b}$ current and future searches onto 2HDM parameter spaces". In: *Physics Letters B* 810 (Nov. 2020), p. 135819. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2020.135819.

[2] Henry Day-Hall et al. "Spectral Clustering for Jet Physics". In: (2021). arXiv: 2104.01972 [hep-ph].

[3] Henry Day-Hall et al. "Revisiting Jet Clustering Algorithms for New Higgs Boson Searches in Hadronic Final States". In: (2020). arXiv: 2008.02499 [hep-ph].

[4] Joseph John Thomson. "Discovery of the electron". In: *Philosophical Magazine* 44 (1897), p. 93.

[5] Michael E Peskin. *An introduction to quantum field theory*. CRC press, 2018.

[6] Christian Grefe. "Detector Optimization Studies and Light Higgs Decay into Muons at CLIC". In: (2014). arXiv: 1402.2780 [physics.ins-det].

[7] Abdelhak Djouadi. "The anatomy of electroweak symmetry breaking". In: *Physics Reports* 457.1-4 (Feb. 2008), pp. 1–216. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2007.10.004.

[8] Michael Spira. "Higgs boson production and decay at hadron colliders". In: *Progress in Particle and Nuclear Physics* 95 (July 2017), pp. 98–159. ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2017.04.001.

[9] Rei Tanaka. *SM Higgs production cross sections at $\sqrt{s}$ = 13-14 TeV (CERN Report 3)*. URL: https://twiki.cern.ch/twiki/bin/view/LHCPhysics/CERNYellowReportPageAt1314T (visited on 12/13/2016).

[10] G.C. Branco et al. "Theory and phenomenology of two-Higgs-doublet models". In: *Physics Reports* 516.1-2 (July 2012), pp. 1–102. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2012.02.002.

[11] CERN. *New results indicate that particle discovered at CERN is a Higgs boson*. Mar. 14, 2013. (Visited on 06/04/2021).

[12] G. Aad et al. "Combined measurements of Higgs boson production and decay using up to 80 fb^{-1} of proton-proton collision data at $s = 13$ TeV collected with the ATLAS experiment". In: *Physical Review D* 101.1 (Jan. 2020). ISSN: 2470-0029. DOI: 10.1103/physrevd.101.012002.

[13] Margarete Mühlleitner. "Beyond the Standard Model Physics". In: (Sept. 2014), p. 144. URL: https://www.itp.kit.edu/~rauch/Teaching/WS1415_BSMHiggs/bsm.pdf.

[14] Karl R. Popper and George Weiss. "The Logic of Scientific Discovery". In: *Physics Today* 12.11 (1959), pp. 53–54. DOI: 10.1063/1.3060577. eprint: https://doi.org/10.1063/1.3060577.

[15] S. Rivat. "On the Heuristics of the Higgs Mechanism." In: *Journal for General Philosophy of Science* 45.2 (2014), pp. 351–367. ISSN: 15728587.

[16] Simon Friederich, Robert Harlander, and Koray Karaca. "Philosophical perspectives on ad hoc hypotheses and the Higgs mechanism". In: *Synthese* 191.16 (2014), pp. 3897–3917. ISSN: 00397857, 15730964.

[17] Giuseppe Degrassi et al. "Higgs mass and vacuum stability in the Standard Model at NNLO". In: *Journal of High Energy Physics* 2012.8 (Aug. 2012). ISSN: 1029-8479. DOI: 10.1007/jhep08(2012)098.

[18] Mu-chun Chen and K. T. Mahanthappa. "Fermion masses and mixing and CP-violation in SO(10) models with family symmetries". In: *International Journal of Modern Physics A* 18.32 (Dec. 2003), pp. 5819–5888. ISSN: 1793-656X. DOI: 10.1142/s0217751x03017026.

[19] N. Jeffrey et al. "Dark Energy Survey Year 3 results: curved-sky weak lensing mass map reconstruction". In: *Monthly Notices of the Royal Astronomical Society* 505.3 (May 2021), pp. 4249–4277. ISSN: 0035-8711. DOI: 10.1093/mnras/stab1515. arXiv: 2105.13539 [astro-ph.CO].

[20] S. Abdollahi et al. "Fermi Large Area Telescope Fourth Source Catalog". In: *Astrophysical Journal Supplement* 1, 33 (247), p. 33. DOI: 10.3847/1538-4365/ab6bcb. arXiv: 1902.10045 [astro-ph.HE].

[21] Andrei D Sakharov. "Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe". In: *Soviet Physics Uspekhi* 34.5 (May), pp. 392–393. DOI: 10.1070/pu1991v034n05abeh002497.

[22] T. Albahri et al. "Measurement of the anomalous precession frequency of the muon in the Fermilab Muon g-2 Experiment". In: *Physical Review D* 103.7 (Apr. 2021). ISSN: 2470-0029. DOI: 10.1103/physrevd.103.072002.

[23] Joel Oredsson and Johan Rathsman. "Z_2 breaking effects in 2-loop RG evolution of 2HDM". In: *Journal of High Energy Physics* 2, 152 (2019), p. 152. DOI: 10.1007/JHEP02(2019)152. arXiv: 1810.02588 [hep-ph].

[24] David Eriksson, Johan Rathsman, and Oscar Stål. "2HDMC - two-Higgs-doublet model calculator". In: *Computer Physics Communications* 181.1 (2010), pp. 189–205. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2009.09.011.

[25] Renato Guedes, Rui Santos, and Miguel Won. "Limits on strong flavor changing neutral current top couplings at the LHC". In: *Physical Review D* 88.11 (Dec. 2013). ISSN: 1550-2368. DOI: 10.1103/physrevd.88.114011.

[26] Gauhar Abbas et al. "Flavour-changing top decays in the aligned two-Higgs-doublet model". In: *Journal of High Energy Physics* 2015.6 (2015), pp. 1–26. arXiv: 1503.06423 [hep-ph].

[27] A. E. Cárcamo Hernández, S. F. King, and H. Lee. "Fermion mass hierarchies from vector-like families with an extended 2HDM and a possible explanation for the electron and muon anomalous magnetic moments". In: *Physical Review D* 103.11 (2021), p. 115024. arXiv: 2101.05819 [hep-ph].

[28] Mayumi Aoki et al. "Models of Yukawa interaction in the two Higgs doublet model, and their collider phenomenology". In: *Physical Review D* 80.1 (July 2009). ISSN: 1550-2368. DOI: 10.1103/physrevd.80.015017.

[29] M. Carena et al. "Reconciling the two-loop diagrammatic and effective field theory computations of the mass of the lightest -even Higgs boson in the MSSM". In: *Nuclear Physics B* 580.1-2 (July 2000), pp. 29–57. ISSN: 0550-3213. DOI: 10.1016/s0550-3213(00)00212-1.

[30] S. Dimopoulos, S. Raby, and Frank Wilczek. "Supersymmetry and the scale of unification". In: *Physical Review D* 24 (6 Sept. 1981), pp. 1681–1683. DOI: 10.1103/PhysRevD.24.1681.

[31] Andrew Fowlie et al. "Dark matter and collider signatures of the MSSM". In: *Physical Review D* 88.5 (Sept. 2013). ISSN: 1550-2368. DOI: 10.1103/physrevd.88.055012.

[32] Yingchuan Li, Stefano Profumo, and Michael Ramsey-Musolf. "A comprehensive analysis of electric dipole moment constraints on CP-violating phases in the MSSM". In: *Journal of High Energy Physics* 2010.8 (Aug. 2010). ISSN: 1029-8479. DOI: 10.1007/jhep08(2010)062.

[33] ATLAS collaboration. "Search for Supersymmetry at the LHC in Events with Jets and Missing Transverse Energy". In: *Physical Review Letters* 22, 221804 (107), p. 221804. DOI: 10.1103/PhysRevLett.107.221804. arXiv: 1109.2352 [hep-ex].

[34] S. Chatrchyan et al. "Search for Supersymmetry at the LHC in Events with Jets and Missing Transverse Energy". In: *Physical Review Letters* 107.22 (Nov. 2011). ISSN: 1079-7114. DOI: 10.1103/physrevlett.107.221804.

[35] M. Shifman. "Reflections and Impressionistic Portrait at the Conference "Frontiers Beyond the Standard Model," FTPI, Oct. 2012". In: (2012). arXiv: 1211.0004 [physics.pop-ph].

[36] Shinya Kanemura, Takahiro Kubota, and Eiichi Takasugi. "Lee-Quigg-Thacker bounds for Higgs boson masses in a two-doublet model". In: *Physics Letters B* 313.1-2 (Aug. 1993), pp. 155–160. ISSN: 0370-2693. DOI: 10.1016/0370-2693(93)91205-2.

[37] A. G Akeroyd, A. Arhrib, and E. Naimi. "Note on tree-level unitarity in the general two Higgs doublet model". In: *Physics Letters B* 490.1-2 (Sept. 2000), pp. 119–124. ISSN: 0370-2693. DOI: 10.1016/s0370-2693(00)00962-x.

[38] A. Arhrib. "Unitarity constraints on scalar parameters of the Standard and Two Higgs Doublets Model". In: (2000). arXiv: hep-ph/0012353.

[39] I. F. Ginzburg and I. P. Ivanov. "Tree-level unitarity constraints in the 2HDM with CP-violation". In: (Dec. 2003). arXiv: hep-ph/0312374 [hep-ph].

[40] Agnieszka Ilnicka, Maria Krawczyk, and Tania Robens. "Constraining the Inert Doublet Model". In: (2015). arXiv: 1505.04734 [hep-ph].

[41] John F. Gunion and Howard E. Haber. "CP-conserving two-Higgs-doublet model: The approach to the decoupling limit". In: *Physical Review D* 67.7 (Apr. 2003). ISSN: 1089-4918. DOI: 10.1103/physrevd.67.075019.

[42] J. Haller et al. "Update of the global electroweak fit and constraints on two-Higgs-doublet models". In: *The European Physical Journal C* 78.8 (Aug. 2018). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-6131-3.

[43] Michael E. Peskin and Tatsu Takeuchi. "Estimation of oblique electroweak corrections". In: *Physical Review D* 46 (1 July 1992), pp. 381–409. DOI: 10.1103/PhysRevD.46.381.

[44] W. Grimus et al. "The oblique parameters in multi-Higgs-doublet models". In: *Nuclear Physics B* 801.1-2 (Sept. 2008), pp. 81–96. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2008.04.019.

[45] Jérémy Bernon et al. "Scrutinizing the alignment limit in two-Higgs-doublet models. Part 2. $m_H = 125$ GeV". In: *Physical Review D* 93.3 (Feb. 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.93.035027.

[46] Morad Aaboud et al. "Search for a heavy Higgs boson decaying into a Z boson and another heavy Higgs boson in the $\ell\ell bb$ final state in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: *Physics Letters B* 783 (2018), pp. 392–414. DOI: 10.1016/j.physletb.2018.07.006. arXiv: 1804.01126 [hep-ex].

[47] Tilman Plehn. "Charged Higgs boson production in bottom-gluon fusion". In: *Physical Review D* 67.1 (Jan. 2003). ISSN: 1089-4918. DOI: 10.1103/physrevd.67.014018.

[48] Abdesslam Arhrib et al. "Enhanced charged Higgs production through W -Higgs fusion in W -b scattering". In: *Journal of High Energy Physics* 2016.5 (May 2016). ISSN: 1029-8479. DOI: 10.1007/jhep05(2016)093.

[49] E. Eichten et al. "Supercollider physics". In: *Review of Modern Physics* 56 (4 Oct. 1984), pp. 579–707. DOI: 10.1103/RevModPhys.56.579.

[50] Prasenjit Sanyal. "Limits on the charged Higgs parameters in the two Higgs doublet model using CMS $\sqrt{s} = 13$ TeV results". In: *The European Physical Journal C* 79.11 (Nov. 2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7431-y.

[51] Mayumi Aoki et al. "Light charged Higgs bosons at the LHC in two-Higgs-doublet models". In: *Physical Review D* 84.5 (Sept. 2011). ISSN: 1550-2368. DOI: 10.1103/physrevd.84.055028.

[52] A. A. Barrientos Bendezú and B. A. Kniehl. "Squark loop correction to $W^{\pm}H^{\mp}$ associated hadroproduction". In: *Physical Review D* 63.1 (Dec. 2000). ISSN: 1089-4918. DOI: 10.1103/physrevd.63.015009.

[53] Baradhwaj Coleppa et al. "Seeking heavy Higgs bosons through cascade decays". In: *Physical Review D* 97 (7 Apr. 2018), p. 075007. DOI: 10.1103/PhysRevD.97.075007.

[54] A. G. Akeroyd et al. "Prospects for charged Higgs searches at the LHC". In: *The European Physical Journal C* 77.5 (May 2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-4829-2.

[55] S Moretti and W.J Stirling. "Contributions of below-threshold decays to Higgs branching ratios". In: *Physics Letters B* 347.3-4 (Mar. 1995), pp. 291–299. ISSN: 0370-2693. DOI: 10.1016/0370-2693(95)00088-3.

[56] Abdel Djouadi, Jan Kalinowski, and Peter M Zerwas. "Two-and three-body decay modes of susy higgs particles". In: *Zeitschrift f"ur Physik C Particles and Fields* 70.3 (1996), pp. 435–447. arXiv: hep-ph/9511342 [hep-ph].

[57] A. Djouadi, J. Kalinowski, and M. Spira. "HDECAY: a program for Higgs boson decays in the Standard Model and its supersymmetric extension". In: *Computer Physics Communications* 108.1 (Jan. 1998), pp. 56–74. ISSN: 0010-4655. DOI: 10.1016/s0010-4655(97)00123-9.

[58] Marcel Krause, Margarete Mühlleitner, and Michael Spira. "2HDECAY-A program for the calculation of electroweak one-loop corrections to Higgs decays in the Two-Higgs-Doublet Model including state-of-the-art QCD corrections". In: *Computer Physics Communications* 246 (Jan. 2020), p. 106852. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2019.08.003.

[59] V. Khachatryan et al. "Search for neutral resonances decaying into a Z boson and a pair of b jets or *tau* leptons". In: *Physics Letters B* 759 (Aug. 2016), pp. 369–394. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2016.05.087.

[60] A. M. Sirunyan et al. "Search for new neutral Higgs bosons through the H $\rightarrow$ ZA $\rightarrow$ $\ell^+\ell^-$b$\bar{\text{b}}$ process in pp collisions at $\sqrt{s}$ = 13 TeV". In: *Journal of High Energy Physics* 2020.3 (Mar. 2020). ISSN: 1029-8479. DOI: 10.1007/jhep03(2020)055.

[61] Baradhwaj Coleppa, Felix Kling, and Shufang Su. "Exotic decays of a heavy neutral Higgs through HZ/AZ channel". In: *Journal of High Energy Physics* 2014.9 (Sept. 2014). ISSN: 1029-8479. DOI: 10.1007/jhep09(2014)161.

[62] A. M. Sirunyan et al. "Search for a heavy pseudoscalar boson decaying to a Z and a Higgs boson at $\sqrt{s}$ = 13,TeV". In: *The European Physical Journal C* 79.7 (July 2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7058-z.

[63] M. Aaboud et al. "Search for heavy resonances decaying into a W or Z boson and a Higgs boson in final states with leptons and b-jets in 36fb^{-1} of $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector". In: *Journal of High Energy Physics* 2018.3 (Mar. 2018). ISSN: 1029-8479. DOI: 10.1007/jhep03(2018)174.

[64] A. M. Sirunyan et al. "Search for a heavy pseudoscalar Higgs boson decaying into a 125 GeV Higgs boson and a Z boson in final states with two tau and two light leptons at $\sqrt{s}$ = 13 TeV". In: *Journal of High Energy Physics* 2020.3 (Mar. 2020). ISSN: 1029-8479. DOI: 10.1007/jhep03(2020)065.

[65] Pedro M. Ferreira, Stefan Liebler, and Jonas Wittbrodt. "$pp \rightarrow A \rightarrow AZh$ and the wrong-sign limit of the two-Higgs-doublet model". In: *Physical Review D* 97.5 (Mar. 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.055008.

[66] G. C. Dorsch et al. "Echoes of the Electroweak Phase Transition: Discovering a Second Higgs Doublet through $A_0 \rightarrow H_0$". In: *Physical Review Letters* 113.21 (Nov. 2014). ISSN: 1079-7114. DOI: 10.1103/physrevlett.113.211802.

[67] Robert V. Harlander, Stefan Liebler, and Hendrik Mantler. "SusHi: A program for the calculation of Higgs production in gluon fusion and bottom-quark annihilation in the Standard Model and the MSSM". In: *Computer Physics Communications* 184 (2013), pp. 1605–1617. DOI: 10.1016/j.cpc.2013.02.006. arXiv: 1212.3249 [hep-ph].

[68] Robert V. Harlander, Stefan Liebler, and Hendrik Mantler. "SusHi Bento: Beyond NNLO and the heavy-top limit". In: *Computer Physics Communications* 212 (2017), pp. 239–257. DOI: 10.1016/j.cpc.2016.10.015. arXiv: 1605.03190 [hep-ph].

[69] Robert V. Harlander and William B. Kilgore. "Next-to-next-to-leading order Higgs production at hadron colliders". In: *Physical Review Letters* 88 (2002), p. 201801. DOI: 10.1103/PhysRevLett.88.201801. arXiv: hep-ph/0201206.

[70] Robert V. Harlander and William B. Kilgore. "Higgs boson production in bottom quark fusion at next-to-next-to leading order". In: *Physical Review D* 68 (2003), p. 013001. DOI: 10.1103/PhysRevD.68.013001. arXiv: hep-ph/0304035.

[71] Johan Alwall et al. "MadGraph 5: going beyond". In: *Journal of High Energy Physics* 2011.6 (June 2011). ISSN: 1029-8479. DOI: 10.1007/jhep06(2011)128.

[72] Alexander Bird and Emma Tobin. "Natural Kinds". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2018. Metaphysics Research Lab, Stanford University, 2018.

[73] Waleed Abdallah et al. "Reinterpretation of LHC Results for New Physics: Status and recommendations after Run 2". In: *SciPost Physics* 9.2 (Aug. 2020). ISSN: 2542-4653. DOI: 10.21468/scipostphys.9.2.022.

[74] Richard P. Feynman. "The Behavior of Hadron Collisions at Extreme Energies". In: *Special Relativity and Quantum Theory: A Collection of Papers on the Poincaré Group*. Ed. by M. E. Noz and Y. S. Kim. Dordrecht: Springer Netherlands, 1988, pp. 289–304. ISBN: 978-94-009-3051-3. DOI: 10.1007/978-94-009-3051-3_25.

[75] A. Ali and G. Kramer. "JETS and QCD: a historical review of the discovery of the quark and gluon jets and its impact on QCD". In: *The European Physical Journal H* 36.2 (Sept. 2011), pp. 245–326. ISSN: 2102-6467. DOI: 10.1140/epjh/e2011-10047-1.

[76] John C. Collins, Davison E. Soper, and George F. Sterman. "Factorization of Hard Processes in QCD". In: *Advanced Series on Directions in High Energy Physics* 5 (1989), pp. 1–91. DOI: 10.1142/9789814503266_0001. arXiv: hep-ph/0409313.

[77] N Metropolis. "The Beginning of the Monte Carlo Method". In: *Los Alamos Science* 15 (1987), pp. 125–130.

[78] E. Boos et al. "CompHEP 4.4 - Automatic Computations from Lagrangians to Events". In: *Nuclear Instruments & Methods in Physics Research Section A-accelerators Spectrometers Detectors and Associated Equipment* 534 (2004), pp. 250–259.

[79] Michelangelo L Mangano et al. "ALPGEN, a generator for hard multiparton processes in hadronic collisions". In: *Journal of High Energy Physics* 2003.07 (July 2003), pp. 001–001. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2003/07/001.

[80] Andy Buckley et al. "General-purpose event generators for LHC physics". In: *Physics Reports* 504.5 (July 2011), pp. 145–233. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2011.03.005.

[81] R. Reed et al. "Vertex finding in pile-up rich events for p+p and d+Au collisions at STAR". In: *Journal of Physics: Conference Series* 219 (Apr. 2010). DOI: 10.1088/1742-6596/219/3/032020.

[82] Simone Marzani, Gregory Soyez, and Michael Spannowsky. "Looking Inside Jets". In: *Lecture Notes in Physics* (2019). ISSN: 1616-6361. DOI: 10.1007/978-3-030-15709-8.

[83] T Sj strand and P Skands. "Multiple Interactions and the Structure of Beam Remnants". In: *Journal of High Energy Physics* 2004.03 (mar), pp. 053–053. DOI: 10.1088/1126-6708/2004/03/053.

[84] Z Marshall. *Simulation of Pile-up in the ATLAS Experiment*. Tech. rep. Geneva: CERN, Oct. 2013. DOI: 10.1088/1742-6596/513/2/022024.

[85] Torbjörn Sjöstrand et al. "An introduction to PYTHIA 8.2". In: *Computer Physics Communications* 191 (June 2015), pp. 159–177. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2015.01.024.

[86] Manuel Bähr et al. "Herwig++ physics and manual". In: *The European Physical Journal C* 58.4 (Nov. 2008), pp. 639–707. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-008-0798-9.

[87] Enrico Bothmann et al. "Event generation with Sherpa 2.2". In: *SciPost Physics* 7.3 (Sept. 2019). ISSN: 2542-4653. DOI: 10.21468/scipostphys.7.3.034.

[88] Jonathan L. Feng et al. "ForwArd Search ExpeRiment at the LHC". In: *Physical Review D* 97.3 (Feb. 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.035001.

[89] Michael Moll. "Displacement Damage in Silicon Detectors for High Energy Physics". In: *IEEE Transactions on Nuclear Science* 65.8 (2018), pp. 1561–1582. DOI: 10.1109/TNS.2018.2819506.

[90] G. Anelli et al. "The TOTEM experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3 (2008), S08007. DOI: 10.1088/1748-0221/3/08/S08007.

[91] P.A. Zyla et al. "Review of Particle Physics". In: *Progress of Theoretical and Experimental Physics* 2020.8 (2020), p. 083C01. DOI: 10.1093/ptep/ptaa104.

[92] Tevian Dray and Corinne A Manogue. "Conventions for spherical coordinates". In: *Oregon State University* (2002).

[93] G. L. Bayatian et al. "CMS Physics: Technical Design Report Volume 1: Detector Performance and Software". In: (2006).

[94] Danek Kotliński. "The CMS pixel detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 465.1 (2001). SPD2000, pp. 46–50. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(01)00345-X.

[95] A.M. Sirunyan et al. "Particle-flow reconstruction and global event description with the CMS detector". In: *Journal of Instrumentation* 12.10 (Oct. 2017), P10003–P10003. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/10/p10003.

[96] Marco Paganoni. "The CMS electromagnetic calorimeter". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 535.1 (2004). Proceedings of the 10th International Vienna Conference on Instrumentation, pp. 461–465. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2004.07.174.

[97] Seth I. Cooper. "Phase I Upgrade of the CMS Hadron Calorimeter". In: *Nuclear and Particle Physics Proceedings* 273-275 (2016). 37th International Conference on High Energy Physics (ICHEP), pp. 1002–1007. ISSN: 2405-6014. DOI: https://doi.org/10.1016/j.nuclphysbps.2015.09.157.

[98] Bannaje Sripathi Acharya et al. *The CMS Outer Hadron Calorimeter*. Tech. rep. Geneva: CERN, June 2006. URL: http://cds.cern.ch/record/973131.

[99] I. Vila. "The CMS muon system". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 518.1 (2004). Frontier Detectors for Frontier Physics: Proceedin, pp. 91–93. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2003.10.032.

[100] Chiara Rizzi. "LHC and ATLAS". In: *Searches for Supersymmetric Particles in Final States with Multiple Top and Bottom Quarks with the Atlas Detector*. Cham: Springer International Publishing, 2020, pp. 33–67. ISBN: 978-3-030-52877-5. DOI: 10.1007/978-3-030-52877-5_3.

[101] Vardan Khachatryan et al. "The CMS trigger system". English. In: *Journal of Instrumentation* 12.1 (2017). Replaced with the published version. Added the journal reference and DOI. All the figures and tables can be found at http://cms-results.web.cern.ch/cms-results/public-results/publications/TRG-12-001/index.html. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/01/P01020.

[102] J. de Favereau et al. "DELPHES 3: a modular framework for fast simulation of a generic collider experiment". In: *Journal of High Energy Physics* 2014.2 (Feb. 2014). ISSN: 1029-8479. DOI: 10.1007/jhep02(2014)057.

[103] John Conway. *PGS 4*. Apr. 2012. URL: http://conway.physics.ucdavis.edu/research/software/pgs/pgs4-general.htm (visited on 06/24/2021).

[104] S. Agostinelli et al. "Geant4 – a simulation toolkit". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8.

[105] Gavin P. Salam. "Towards jetography". In: *The European Physical Journal C* 67.3-4 (May 2010), pp. 637–686. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-010-1314-6.

[106] Stephen D. Ellis, Zoltan Kunszt, and Davison E. Soper. "The One Jet Inclusive Cross-section at Order $\alpha^{-}s^{3}$: Gluons Only". In: *Physical Review Letters* 62 (1989), p. 726. DOI: 10.1103/PhysRevLett.62.726.

[107] Mario Martinez. "Experimental review of jet physics in hadronic collisions". In: *European Physical Journal C* 61 (2009), pp. 637–647. DOI: 10.1140/epjc/s10052-008-0849-2.

[108] Mark Srednicki. *Quantum field theory*. Cambridge University Press, 2007.

[109] S. Joseph et al. "HERWIRI1.0: MC realization of IR-improved DGLAP-CS parton showers". In: *Physics Letters B* 685.4 (2010), pp. 283–292. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2010.02.007.

[110] G. Dissertori, F. Moortgat, and M. A. Weber. "Hadronic Event-Shape Variables at CMS". In: *34th International Conference on High Energy Physics*. Oct. 2008. arXiv: 0810.3208 [hep-ph].

[111] Aleš Bezděk and Josef Sebera. "Matlab script for 3D visualizing geodata on a rotating globe". In: *Computers & Geosciences* 56 (2013), pp. 127–130. ISSN: 0098-3004. DOI: https://doi.org/10.1016/j.cageo.2013.03.007.

[112] S. Chekanov et al. "Effect of PYTHIA8 tunes on event shapes and top-quark reconstruction in e^+e^- annihilation at CLIC". In: (2017). arXiv: 1710.07713 [hep-ph].

[113] G. Aad et al. "Measurement of charged-particle event shape variables in inclusive $\sqrt{s} = 7$ TeV proton-proton interactions with the ATLAS detector". In: *Physical Review D* 88.3 (Aug. 2013). ISSN: 1550-2368. DOI: 10.1103/physrevd.88.032004.

[114] S. Brandt and H. D. Dahmen. "Axes and scalar measures of two-jet and three-jet events". In: *Zeitschrift fur Physik C Particles and Fields* 1.1 (Mar. 1979), pp. 61–70. DOI: 10.1007/BF01450381.

[115] George Sterman and Steven Weinberg. "Jets from Quantum Chromodynamics". In: *Physical Review Letters* 23 (39), pp. 1436–1439. DOI: 10.1103/PhysRevLett.39.1436.

[116] F. Abe et al. "Topology of three-jet events in $\overline{p}p$ collisions at $\sqrt{s} = 1.8$ TeV". In: *Physical Review D* 45 (5 Mar. 1992), pp. 1448–1458. DOI: 10.1103/PhysRevD.45.1448.

[117] Gavin P Salam and Grégory Soyez. "A practical seedless infrared-safe cone jet algorithm". In: *Journal of High Energy Physics* 05 (2007), pp. 086–086. DOI: 10.1088/1126-6708/2007/05/086. arXiv: 0704.0292 [hep-ph].

[118] Erik Gerwick et al. "QCD Jet Rates with the Inclusive Generalized kt Algorithms". In: *Journal of High Energy Physics* 04 (2013), p. 089. DOI: 10.1007/JHEP04(2013)089. arXiv: 1212.5235 [hep-ph].

[119] Yu.L Dokshitzer et al. "Better jet clustering algorithms". In: *Journal of High Energy Physics* 1997.08 (Aug. 1997), pp. 001–001. ISSN: 1029-8479. DOI: 10.1088/1126-6708/1997/08/001.

[120] Markus Wobisch and Thorsten Wengler. "Hadronization corrections to jet cross sections in deep-inelastic scattering". In: (1999), hep–ph/9907280. arXiv: hep-ph/9907280 [hep-ph].

[121] Yu L Dokshitzer. "contribution cited in Report of the Hard QCD Working Group". In: *Proc. Workshop on Jet Studies at LEP and HERA, Durham*. 1990.

[122] Stephen D. Ellis and Davison E. Soper. "Successive combination jet algorithm for hadron collisions". In: *Physical Review D* D48 (1993), pp. 3160–3166. DOI: 10.1103/PhysRevD.48.3160. arXiv: hep-ph/9305266 [hep-ph].

[123] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. "The anti-k_T jet clustering algorithm". In: *Journal of High Energy Physics* 2008.04 (Apr. 2008), pp. 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/04/063.

[124] S. Catani et al. "Longitudinally invariant k_T clustering algorithms for hadron hadron collisions". In: *Nuclear Physics B* 406 (1993), pp. 187–224. DOI: 10.1016/0550-3213(93)90166-M.

[125] Stefano Moretti, Leif Lonnblad, and Torbjorn Sjostrand. "New and old jet clustering algorithms for electron - positron events". In: *Journal of High Energy Physics* 08 (1998), p. 001. DOI: 10.1088/1126-6708/1998/08/001. arXiv: hep-ph/9804296 [hep-ph].

[126] David Krohn, Jesse Thaler, and Lian-Tao Wang. "Jets with variable R". In: *Journal of High Energy Physics* 2009.06 (June 2009), pp. 059–059. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2009/06/059.

[127] Luca Scodellaro. "b tagging in ATLAS and CMS". In: (2017). arXiv: 1709.01290 [hep-ex].

[128] Philip Bechtle et al. "HiggsBounds-4: improved tests of extended Higgs sectors against exclusion bounds from LEP, the Tevatron and the LHC". In: *The European Physical Journal C* 74.3 (Mar. 2014). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-013-2693-2.

[129] Philip Bechtle et al. "HiggsSignals: Confronting arbitrary Higgs sectors with measurements at the Tevatron and the LHC". In: *The European Physical Journal C* 74.2 (Feb. 2014). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-013-2711-4.

[130] F. Mahmoudi. "SuperIso v2.3: A program for calculating flavor physics observables in supersymmetry". In: *Computer Physics Communications* 180.9 (Sept. 2009), pp. 1579–1613. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2009.02.017.

[131] Richard D. Ball et al. "Parton distributions for the LHC run II". In: *Journal of High Energy Physics* 2015.4 (Apr. 2015). ISSN: 1029-8479. DOI: 10.1007/jhep04(2015)040.

[132] Eric Conte, Benjamin Fuks, and Guillaume Serret. "MadAnalysis 5, a user-friendly framework for collider phenomenology". In: *Computer Physics Communications* 184.1 (Jan. 2013), pp. 222–256. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2012.09.009.

[133] Eric Conte and Benjamin Fuks. "Confronting new physics theories to LHC data with MADANALYSIS 5". In: *International Journal of Modern Physics A* 33.28 (Oct. 2018), p. 1830027. ISSN: 1793-656X. DOI: 10.1142/s0217751x18300272.

[134] A. M. Sirunyan et al. "Search for nonresonant Higgs boson pair production in the $b\bar{b}b\bar{b}$ final state at $\sqrt{s} = 13$ TeV". In: *Journal of High Energy Physics* 2019.4 (Apr. 2019). ISSN: 1029-8479. DOI: 10.1007/jhep04(2019)112.

[135] A. M. Sirunyan et al. "Search for nonresonant Higgs boson pair production in final states with two bottom quarks and two photons in proton-proton collisions at $\sqrt{s} = 13$ TeV". In: *Journal of High Energy Physics* 2021.3 (Mar. 2021). ISSN: 1029-8479. DOI: 10.1007/jhep03(2021)257.

[136] M. Aaboud and et. al. "Search for pair production of Higgs bosons in the $b\bar{b}b\bar{b}$ final state using proton-proton collisions at $\sqrt{s} = 13$TeV with the ATLAS detector". In: *Physical Review D* 94 (5 Sept. 2016), p. 052002. DOI: 10.1103/PhysRevD.94.052002.

[137] CMS Collaboration. "Search for resonant and nonresonant Higgs boson pair production in the *bblvlv* final state in proton-proton collisions at sqrt(s) = 13 TeV". In: *Journal of High Energy Physics* 1 (2018). arXiv: 1708.04188 [hep-ex].

[138] V. Khachatryan et al. "Studies of inclusive four-jet production with two b - tagged jets in proton-proton collisions at 7 TeV". In: *Physical Review D* 94.11 (Dec. 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.94.112005.

[139] Andrés Flórez et al. "Probing axionlike particles with $\gamma\gamma$ final states from vector boson fusion processes at the LHC". In: *Physical Review D* 103.9 (May 2021). ISSN: 2470-0029. DOI: 10.1103/physrevd.103.095001.

[140] Maxime Gouzevitch et al. "Scale-invariant resonance tagging in multijet events and new physics in Higgs pair production". In: *Journal of High Energy Physics* 2013.7 (July 2013). ISSN: 1029-8479. DOI: 10.1007/jhep07(2013)148.

[141] M. Aaboud et al. "A search for pair-produced resonances in four-jet final states at $\sqrt{s} = 13$TeV with the ATLAS detector". In: *The European Physical Journal C* 78.3 (Mar. 2018). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-5693-4.

[142] J. Katharina Behr et al. "Boosting Higgs pair production in the $b\bar{b}b\bar{b}$ final state with multivariate techniques". In: *The European Physical Journal C* 76.7 (July 2016). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4215-5.

[143] CDF Collaboration. "Search for a two-Higgs-boson doublet using a simplified model in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV". In: *Physical review letters* 110.12 (2012), p. 121801. arXiv: 1212.3837 [hep-ex].

[144] G. Aad et al. "Search for a multi-Higgs-boson cascade in $W^+W^-b\bar{b}$ events with the ATLAS detector in pp collisions at $\sqrt{s} = 8$TeV". In: *Physical Review D* 89.3 (Feb. 2014). ISSN: 1550-2368. DOI: 10.1103/physrevd.89.032002.

[145] T. Aaltonen et al. "First simultaneous measurement of the top quark mass in thelepton+jetsand dilepton channels at CDF". In: *Physical Review D* 79.9 (May 2009). ISSN: 1550-2368. DOI: 10.1103/physrevd.79.092005.

[146] G. Aad et al. "Search for the $HH \to b\bar{b}b\bar{b}$ process via vector-boson fusion production using proton-proton collisions at $\sqrt{s} = 13$TeV with the ATLAS detector". In: *Journal of High Energy Physics* 2021.1 (Jan. 2021). ISSN: 1029-8479. DOI: 10.1007/jhep01(2021)145.

[147] Benjamin Tannenwald et al. "Benchmarking Machine Learning Techniques with Di-Higgs Production at the LHC". In: (2020). arXiv: 2009.06754 [hep-ph].

[148] Jacob Amacker et al. "Higgs self-coupling measurements using deep learning in the $b\bar{b}b\bar{b}$ final state". In: *Journal of High Energy Physics* 2020.12 (Dec. 2020). ISSN: 1029-8479. DOI: 10.1007/jhep12(2020)115.

[149] T. Lapsien, R. Kogler, and J. Haller. "A new tagger for hadronically decaying heavy particles at the LHC". In: *The European Physical Journal C* 76.11 (Nov. 2016). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4443-8.

[150] *Boosted Object Tagging with Variable-R Jets in the ATLAS Detector*. Tech. rep. Geneva: CERN, July 2016. URL: https://cds.cern.ch/record/2199360.

[151] *Variable Radius, Exclusive-k_T, and Center-of-Mass Subjet Reconstruction for Higgs($\to b\bar{b}$) Tagging in ATLAS*. Tech. rep. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-010. Geneva: CERN, June 2017. URL: http://cds.cern.ch/record/2268678.

[152] G. Aad et al. "Measurement of the associated production of a Higgs boson decaying into b-quarks with a vector boson at high transverse momentum in pp collisions at $s = 13$ TeV with the ATLAS detector". In: *Physics Letters B* 816 (May 2021), p. 136204. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2021.136204.

[153] Nils J Nilsson. "Introduction to machine learning. An early draft of a proposed textbook". In: (1996).

[154] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, Massachusetts: MIT Press, Nov. 1, 2016. 800 pp. ISBN: 978-0-262-03561-3.

[155] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. "Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning". In: *Physics Reports* 841 (Jan. 2020), pp. 1–63. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2019.11.001.

[156] HEP ML Community. *A Living Review of Machine Learning for Particle Physics*. URL: https://iml-wg.github.io/HEPML-LivingReview/.

[157] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[158] Richard M. Zur et al. "Noise injection for training artificial neural networks: A comparison with weight decay and early stopping". In: *Medical Physics* 36.10 (2009), pp. 4810–4818. DOI: https://doi.org/10.1118/1.3213517. eprint: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.3213517.

[159] Graziani M. et al. "Concept attribution: Explaining CNN decisions to physicians". In: *Computers in Biology and Medicine* 123 (2020), p. 103865. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2020.103865.

[160] Quanshi Zhang et al. "Interpreting CNNs via decision trees". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6261–6270. arXiv: 1802.00121 [cs.CV].

[161] Kai Xiao et al. "Noise or Signal: The Role of Image Backgrounds in Object Recognition". In: (2020). arXiv: 2006.09994 [cs.CV].

[162] Xu Luo et al. "Rectifying the Shortcut Learning of Background: Shared Object Concentration for Few-Shot Image Recognition". In: (2021). arXiv: 2107.07746 [cs.CV].

[163] Alexander Radovic et al. "Machine learning at the energy and intensity frontiers of particle physics". In: *Nature* 560.7716 (2018), pp. 41–48. DOI: 10.1038/s41586-018-0361-2.

[164] David Rousseau and Andrey Ustyuzhanin. "Machine Learning scientific competitions and datasets". In: (2020). arXiv: 2012.08520 [physics.data-an].

[165] Benjamin Nachman. "A guide for deploying Deep Learning in LHC searches: How to achieve optimality and account for uncertainty". In: *SciPost Physics* 8.6 (June 2020). ISSN: 2542-4653. DOI: 10.21468/scipostphys.8.6.090.

[166] A. M. Sirunyan et al. "Measurement of the Jet Mass Distribution and Top Quark Mass in Hadronic Decays of Boosted Top Quarks in pp Collisions at $\sqrt{s} = 13$ TeV". In: *Physical Review Letters* 124.20 (May 2020). ISSN: 1079-7114. DOI: 10.1103/physrevlett.124.202001.

[167] Stephen D. Ellis et al. "Nondeterministic Approach to Tree-Based Jet Substructure". In: *Physical Review Letters* 108.18 (May 2012). ISSN: 1079-7114. DOI: 10.1103/physrevlett.108.182003.

[168] Lester Mackey et al. "Fuzzy jets". In: *Journal of High Energy Physics* 2016.6 (June 2016). ISSN: 1029-8479. DOI: 10.1007/jhep06(2016)010.

[169] Fyodor V. Tkachov. "A Theory Of Jet Definition". In: *International Journal of Modern Physics A* 17.21 (2002), pp. 2783–2884. DOI: 10.1142/S0217751X02009977. eprint: https://doi.org/10.1142/S0217751X02009977.

[170] Jinmian Li, Tianjun Li, and Fang-Zhou Xu. "Reconstructing boosted Higgs jets from event image segmentation". In: *Journal of High Energy Physics* 2021.4 (Apr. 2021). ISSN: 1029-8479. DOI: 10.1007/jhep04(2021)156.

[171] M Andrews et al. "End-to-End Event Classification of High-Energy Physics Data". In: *Journal of Physics: Conference Series* 1085 (Sept. 2018), p. 042022. DOI: 10.1088/1742-6596/1085/4/042022.

[172] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. "Energy flow networks: deep sets for particle jets". In: *Journal of High Energy Physics* 2019.1 (Jan. 2019). ISSN: 1029-8479. DOI: 10.1007/jhep01(2019)121.

[173] Sebastian Macaluso and David Shih. "Pulling out all the tops with computer vision and deep learning". In: *Journal of High Energy Physics* 2018.10 (Oct. 2018). ISSN: 1029-8479. DOI: 10.1007/jhep10(2018)121.

[174] Pierre Baldi et al. "Parameterized neural networks for high-energy physics". In: *The European Physical Journal C* 76.5 (Apr. 2016). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4099-4.

[175] Kaustuv Datta and Andrew Larkoski. "How much information is in a jet?" In: *Journal of High Energy Physics* 2017.6 (June 2017). ISSN: 1029-8479. DOI: 10.1007/jhep06(2017)073.

[176] A.M. Sirunyan et al. "Identification of heavy-flavour jets with the CMS detector in *pp* collisions at 13 TeV". In: *Journal of Instrumentation* 13.05 (May 2018), P05011–P05011. ISSN: 1748-0221. DOI: 10.1088/1748-0221/13/05/p05011.

[177] A Altheimer et al. "Jet substructure at the Tevatron and LHC: new results, new tools, new benchmarks". In: *Journal of Physics G: Nuclear and Particle Physics* 39.6 (May 2012), p. 063001. ISSN: 1361-6471. DOI: 10.1088/0954-3899/39/6/063001.

[178] D. Adams et al. "Towards an Understanding of the Correlations in Jet Substructure". In: *The European Physical Journal C* 75.9 (2015), pp. 1–52. arXiv: 1504.00679 [hep-ph].

[179] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. "Energy flow polynomials: a complete linear basis for jet substructure". In: *Journal of High Energy Physics* 2018.4 (Apr. 2018). ISSN: 1029-8479. DOI: 10.1007/jhep04(2018)013.

[180] Amit Chakraborty, Sung Hak Lim, and Mihoko M. Nojiri. "Interpretable deep learning for two-prong jet classification with jet spectra". In: *Journal of High Energy Physics* 2019.7 (July 2019). ISSN: 1029-8479. DOI: 10.1007/jhep07(2019)135.

[181] Frédéric A. Dreyer, Gavin P. Salam, and Grégory Soyez. "The Lund jet plane". In: *Journal of High Energy Physics* 2018.12 (Dec. 2018). ISSN: 1029-8479. DOI: 10.1007/jhep12(2018)064.

[182] Frédéric A. Dreyer and Huilin Qu. "Jet tagging in the Lund plane with graph networks". In: *Journal of High Energy Physics* 2021.3 (2021), pp. 1–23. arXiv: 2012.08526 [hep-ph].

[183] Charanjit K. Khosa and Simone Marzani. "Higgs boson tagging with the Lund jet plane". In: *Physical Review D* 104 (5 Sept. 2021), p. 055043. DOI: 10.1103/PhysRevD.104.055043. arXiv: 2105.03989 [hep-ph].

[184] Taoli Cheng. "Recursive Neural Networks in Quark/Gluon Tagging". In: *Computing and Software for Big Science* 2.1 (June 2018). ISSN: 2510-2044. DOI: 10.1007/s41781-018-0007-y.

[185] Judith M. Katzy. "QCD Monte-Carlo model tunes for the LHC". In: *Progress in Particle and Nuclear Physics* 73 (2013), pp. 141–187. ISSN: 0146-6410. DOI: https://doi.org/10.1016/j.ppnp.2013.08.002.

[186] C Adam-Bourdarios et al. "The Higgs Machine Learning Challenge". In: *Journal of Physics: Conference Series* 664.7 (dec), p. 072015. DOI: 10.1088/1742-6596/664/7/072015.

[187] Ian H Witten and Eibe Frank. "Data mining: practical machine learning tools and techniques with Java implementations". In: *ACM Sigmod Record* 31.1 (2002), pp. 76–77.

[188] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1558602380.

[189] Leo Breiman et al. *Calssification and Regression Trees*. Chapman and Hall, 1984. ISBN: 9780412048418.

[190] Stuart L. Crawford. "Extensions to the CART algorithm". In: *International Journal of Man-Machine Studies* 31.2 (1989), pp. 197–217. ISSN: 0020-7373. DOI: https://doi.org/10.1016/0020-7373(89)90027-8.

[191] Yoav Freund, Robert E Schapire, et al. "Experiments with a new boosting algorithm". In: *icml*. Vol. 96. Citeseer. 1996, pp. 148–156.

[192] Gregor Kasieczka et al. "Deep-learning top taggers or the end of QCD?" In: *Journal of High Energy Physics* 2017.5 (May 2017). ISSN: 1029-8479. DOI: 10.1007/jhep05(2017)006.

[193] M. Aaboud et al. "Observation of Higgs boson production in association with a top quark pair at the LHC with the ATLAS detector". In: *Physics Letters B* 784 (2018), pp. 173–191. DOI: 10.1016/j.physletb.2018.07.035. arXiv: 1806.00425 [hep-ex].

[194] Li-Gang Xia. "QBDT, a new boosting decision tree method with systematical uncertainties into training for High Energy Physics". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 930 (June 2019), pp. 15–26. ISSN: 0168-9002. DOI: 10.1016/j.nima.2019.03.088.

[195] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257.

[196] Rob Potharst, Jan C Bioch, and Thijs Petter. "Monotone decision trees". In: *Department of Computer Science, Erasmus University Rotterdam* (1997).

[197] Stephen P. Curram and John Mingers. "Neural Networks, Decision Tree Induction and Discriminant Analysis: an Empirical Comparison". In: *Journal of the Operational Research Society* 45.4 (1994), pp. 440–450. DOI: 10.1057/jors.1994.62. eprint: https://doi.org/10.1057/jors.1994.62.

[198] Piotr Płoński and Aleksandra Płońska. *Decision Tree vs Neural Network*. 2021. URL: https://mljar.com/machine-learning/decision-tree-vs-neural-network/.

[199] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., Feb. 1, 2006. 758 pp. ISBN: 978-0-387-31073-2.

[200] David J.C. MacKay. *Information theory, inference and learning algorithms*. 7.2. Cambridge university press, Mar. 28, 2015. 640 pp. ISBN: 978-0-521-67051-7.

[201] Lidia Blazquez-Llorca, Virginia Garcia-Marin, and Javier DeFelipe. "Pericellular innervation of neurons expressing abnormally hyperphosphorylated tau in the hippocampal formation of Alzheimer's disease patients". In: *Frontiers in Neuroanatomy* 4 (2010), p. 20. ISSN: 1662-5129. DOI: 10.3389/fnana.2010.00020.

[202] Daniel Guest et al. "Jet flavor classification in high-energy physics with deep neural networks". In: *Physical Review D* 94.11 (Dec. 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.94.112002.

[203] Markus Stoye and. "Deep learning in jet reconstruction at CMS". In: *Journal of Physics: Conference Series* 1085 (sep), p. 042029. DOI: 10.1088/1742-6596/1085/4/042029.

[204] Johannes Erdmann et al. "From the bottom to the top—reconstruction of $t\bar{t}$ events with deep learning". In: *Journal of Instrumentation* 14.11 (2019), P11015. DOI: 10.1088/1748-0221/14/11/P11015. arXiv: 1907.11181 [hep-ex].

[205] Y. Le Cun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. NIPS'89. Cambridge, MA, USA: MIT Press, 1989, pp. 396–404.

[206] Nameer Hirschkind et al. *Convolutional Neural Network*. URL: https://brilliant.org/wiki/convolutional-neural-network/ (visited on 09/18/2021).

[207] Leandro G. Almeida et al. "Playing Tag with ANN: Boosted Top Identification with Pattern Recognition". In: *Journal of High Energy Physics* 2015.7 (2015), pp. 1–21. arXiv: 1501.05968 [hep-ph].

[208] Josh Cogan et al. "Jet-images: computer vision inspired techniques for jet tagging". In: *Journal of High Energy Physics* 2015.2 (Feb. 2015). ISSN: 1029-8479. DOI: 10.1007/jhep02(2015)118.

[209] James Barnard et al. "Parton shower uncertainties in jet substructure analyses with deep neural networks". In: *Physical Review D* 95.1 (Jan. 2017). ISSN: 2470-0029. DOI: 10.1103/physrevd.95.014018.

[210] Luke de Oliveira et al. "Jet-images - deep learning edition". In: *Journal of High Energy Physics* 2016.7 (July 2016). ISSN: 1029-8479. DOI: 10.1007/jhep07(2016)069.

[211] Patrick T. Komiske, Eric M. Metodiev, and Matthew D. Schwartz. "Deep learning in color: towards automated quark/gluon jet discrimination". In: *Journal of High Energy Physics* 2017.1 (Jan. 2017). ISSN: 1029-8479. DOI: 10.1007/jhep01(2017)110.

[212] Patrick T. Komiske et al. "Learning to classify from impure samples with high-dimensional data". In: *Physical Review D* 98.1 (2018), p. 011502. DOI: 10.1103/PhysRevD.98.011502. arXiv: 1801.10158 [hep-ph].

[213] Jack H. Collins, Kiel Howe, and Benjamin Nachman. "Extending the search for new resonances with machine learning". In: *Physical Review D* 99.1 (2019), p. 014038. DOI: 10.1103/PhysRevD.99.014038. arXiv: 1902.02634 [hep-ph].

[214] Zhou Cheng et al. "TreeNet: Learning Sentence Representations with Unconstrained Tree Structure". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 4005–4011. ISBN: 9780999241127.

[215] Yong Yu et al. "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures". In: *Neural Computation* 31.7 (July 2019), pp. 1235–1270. ISSN: 0899-7667. DOI: 10.1162/neco_a_01199.

[216] Shannon Egan et al. "Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC". In: (2017). arXiv: 1711.09059 [hep-ex].

[217] J. Erdmann, O. Nackenhorst, and S.V. Zeißner. "Maximum performance of strange-jet tagging at hadron colliders". In: *Journal of Instrumentation* 16.08 (Aug. 2021), P08039. ISSN: 1748-0221. DOI: 10.1088/1748-0221/16/08/p08039.

[218] Gilles Louppe et al. "QCD-aware recursive neural networks for jet physics". In: *Journal of High Energy Physics* 2019.1 (Jan. 2019). ISSN: 1029-8479. DOI: 10.1007/jhep01(2019)057.

[219] Xiangyang Ju et al. "Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors". In: *33rd Annual Conference on Neural Information Processing Systems*. Mar. 2020. arXiv: 2003.11603 [physics.ins-det].

[220] Huilin Qu and Loukas Gouskos. "Jet tagging via particle clouds". In: *Physical Review D* 101.5 (Mar. 2020). ISSN: 2470-0029. DOI: 10.1103/physrevd.101.056019.

[221] Rui Xu and D. Wunsch. "Survey of clustering algorithms". In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678. DOI: 10.1109/TNN.2005.845141.

[222] L.A. Zadeh. "Fuzzy sets". In: *Information and Control* 8.3 (1965), pp. 338–353. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X.

[223] Olivier Cailloux, Claude Lamboray, and Philippe Nemery. "A taxonomy of clustering procedures". In: *66th Meeting of the European Working Group on MCDA*. Marrakech, Morocco, 2007, N/A. URL: https://hal.archives-ouvertes.fr/hal-00985860.

[224] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data Clustering: A Review". In: *ACM Computing Surveys* 31.3 (Sept. 1999). ISSN: 0360-0300. DOI: 10.1145/331499.331504.

[225] Mihael Ankerst et al. "OPTICS: Ordering Points to Identify the Clustering Structure". In: *SIGMOD Rec.* 28.2 (1999). ISSN: 0163-5808. DOI: 10.1145/304181.304187.

[226] Amit Saxena et al. "A review of clustering techniques and developments". In: *Neurocomputing* 267 (2017), pp. 664–681. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.053.

[227] Nishant Yadav et al. "Supervised Hierarchical Clustering with Exponential Linkage". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 6973–6983. URL: http://proceedings.mlr.press/v97/yadav19a.html.

[228] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[229] D. Sculley. "Web-Scale k-Means Clustering". In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, pp. 1177–1178. ISBN: 9781605587998. DOI: 10.1145/1772690.1772862.

[230] Brendan J. Frey and Delbert Dueck. "Clustering by Passing Messages Between Data Points". In: *Science* 315.5814 (2007), pp. 972–976. ISSN: 0036-8075. DOI: 10.1126/science.1136800. eprint: https://science.sciencemag.org/content/315/5814/972.full.pdf.

[231] D. Comaniciu and P. Meer. "Mean shift: a robust approach toward feature space analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619. DOI: 10.1109/34.1000236.

[232] Fionn Murtagh and Pierre Legendre. "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" In: *Journal of Classification* 31.3 (Oct. 2014), pp. 274–295. ISSN: 1432-1343. DOI: 10.1007/s00357-014-9161-z.

[233] Joe H. Ward Jr. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244. DOI: 10.1080/01621459.1963.10500845. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500845.

[234] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[235] Erich Schubert et al. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Transactions on Database Systems* 42.3 (Aug. 2017). ISSN: 0362-5915. DOI: 10.1145/3068335.

[236] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". In: vol. 25. SIGMOD '96 2. Montreal, Quebec, Canada: Association for Computing Machinery, June 1996. DOI: 10.1145/235968.233324.

[237] Matteo Cacciari and Gavin P. Salam. "Dispelling the N^3 myth for the k_T jet-finder". In: *Physics Letters B* 641.1 (2006), pp. 57–61. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2006.08.037.

[238] Boris Lorbeer et al. "Variations on the Clustering Algorithm BIRCH". In: *Big Data Research* 11 (2018). Selected papers from the 2nd INNS Conference on Big Data: Big Data & Neural Networks, pp. 44–53. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2017.09.002.

[239] Douglas A Reynolds. "Gaussian mixture models." In: *Encyclopedia of biometrics* 741 (2009), pp. 659–663.

[240] Arindam Banerjee et al. "Clustering with Bregman divergences." In: *Journal of machine learning research* 6.10 (2005).

[241] Marek 'Smieja and Jacek Tabor. "Spherical wards clustering and generalized voronoi diagrams". In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2015, pp. 1–10. arXiv: 1705.02232 [cs.LG].

[242] Zhihua Zhang and Michael I. Jordan. "Multiway Spectral Clustering: A Margin-Based Perspective". In: *Statistical Science* 23.3 (2008), pp. 383–403. DOI: 10.1214/08-STS266.

[243] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation". In: *Neural Comput.* 15.6 (June 2003). ISSN: 0899-7667. DOI: 10.1162/089976603321780317.

[244] W. E. Donath and A. J. Hoffman. "Lower Bounds for the Partitioning of Graphs". In: *IBM Journal of Research and Development* 17.5 (1973), pp. 420–425. DOI: 10.1147/rd.175.0420.

[245] Jianbo Shi and Jitendra Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. DOI: 10.1109/34.868688.

[246] P.K. Chan, M.D.F. Schlag, and J.Y. Zien. "Spectral K-way ratio-cut partitioning and clustering". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 13.9 (1994), pp. 1088–1096. DOI: 10.1109/43.310898.

[247] Ulrike von Luxburg. "A Tutorial on Spectral Clustering". In: (2007). arXiv: 0711.0189 [cs.DS].

[248] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an Algorithm". In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. NIPS'01. Vancouver, British Columbia, Canada: MIT Press, 2001, pp. 849–856.

[249] Stephen Shum, Najim Dehak, and J. Glass. "On the Use of Spectral and Iterative Methods for Speaker Diarization". In: *INTERSPEECH*. 2012.

[250] Nicolas Tremblay et al. "Compressive Spectral Clustering". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1002–1011. URL: http://proceedings.mlr.press/v48/tremblay16.html.

[251] Alireza Hadjighasem et al. "Spectral-clustering approach to Lagrangian vortex detection". In: *Physical Review E* 93 (6 June 2016), p. 063107. DOI: 10.1103/PhysRevE.93.063107. eprint: 1506.02258.

[252] Hao Li et al. "Strategic Power Infrastructure Defense". In: *Proceedings of the IEEE* 93.5 (2005), pp. 918–933. DOI: 10.1109/JPROC.2005.847260.

[253] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "FastJet user manual". In: *The European Physical Journal C* 72.3 (Mar. 2012). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-012-1896-2.

[254] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. "Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities". In: *Journal of ACM* 61.6 (Nov. 2014). ISSN: 0004-5411. DOI: 10.1145/2665063.

[255] Xiangyang Ju and Benjamin Nachman. "Supervised Jet Clustering with Graph Neural Networks for Lorentz Boosted Bosons". In: *Physical Review D* 102.7 (2020), p. 075014. DOI: 10.1103/PhysRevD.102.075014. arXiv: 2008.06064 [hep-ph].

[256] A.M. Sirunyan et al. "Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV". In: *Journal of Instrumentation* 13.06 (2018). ISSN: 1748-0221. DOI: 10.1088/1748-0221/13/06/p06015. arXiv: 1804.04528 [hep-ex].

[257] J. Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151. DOI: 10.1109/18.61115.

[258] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[259] Simone Alioli et al. "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX". In: *Journal of High Energy Physics* 2010.6 (June 2010), pp. 1–58. ISSN: 1029-8479. DOI: 10.1007/JHEP06(2010)043.

[260] The CMS collaboration. "Identification of b-quark jets with the CMS experiment". In: *Journal of Instrumentation* 8.4 (Apr. 15, 2013), p. 65. ISSN: 1748-0221. DOI: 10.1088/1748-0221/8/04/P04013.

[261] Alex Rogozhnikov. "Reweighting with boosted decision trees". In: *Journal of Physics: Conference Series*. Vol. 762. 1. IOP Publishing. Aug. 20, 2016, p. 012036. DOI: 10.1088/1742-6596/762/1/012036.

[262] Alex Rogozhnikov. *Reweighting algorithms - hep_ml 0.6.0 documentation*. URL: https://arogozhnikov.github.io/hep_ml/reweight.html (visited on 09/12/2018).

[263] Alejandro Baldominos. *A Comparison between NVIDIA's GeForce GTX 1080 and Tesla P100 for Deep Learning*. Medium. Oct. 5, 2017. URL: https://medium.com/@alexbaldo/a-comparison-between-nvidias-geforce-gtx-1080-and-tesla-p100-for-deep-learning-81a918d5b2c7 (visited on 09/17/2018).

[264] Yusaku Sako. *Titan V vs 1080 Ti - Head-to-head battle of the best desktop GPUs on CNNs*. Medium. Dec. 21, 2017. URL: https://medium.com/@u39kun/titan-v-vs-1080-ti-head-to-head-battle-of-the-best-desktop-gpus-on-cnns-d55a19866b7c (visited on 09/17/2018).