Searches for vector-like quarks and other new physics in LHC data

Tomasz Procter

Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy

School of Physics and Astronomy
College of Science and Engineering
University of Glasgow

October 2024

Contents

Acknowledgements

Consistent with the acknowledgements section of every other PhD thesis, I must preface all the following thanks with the warning that there are many more people to thank than can be thanked in a single page: so if I missed you off, I apologise, and thank-you again.

The first vote of thanks is owed to my supervisor, Andy Buckley. Thanks for making sure I always could have several really interesting plates spinning; and knowing when to help me debug and when I would learn more by (enjoyably) bashing my head against the wall a bit. It has no doubt made me both a better programmer and a better scientist. Thanks also to all the other academics in Glasgow who welcomed me into the group even though I don't (merely by comparison, of course) care about the top-quark that much.

The work in this thesis also could not have been completed without invaluable advice and guidance from several academics beyond Glasgow: Anders Kvellestad and Tomas Gonzalo in GAMBIT; Jon Butterworth and Chris Gutschow in the RIVET/YODA/CONTUR "CEDAR" ecosystem; Quake Qin, Sergio Grancagnolo and Joe Haley for the ATLAS VLQ search; and Stephen Jiggins during my work on CARL. There are plenty of other academics who provided good questions, general help, and advice. I would like to particularly mention the LHC reinterpretation community – Sabine Kraml, Sezen Sekmen, Krzysztof Rolbiecki, Jack Araz and many more – who welcomed me into their scientific community and always made me feel like I was making a genuinely useful contribution.

My PhD was rudely interrupted by one of the occupational hazards of being a particle physicist (particularly one based in Geneva): a nasty skiing accident. Given the kindness and generosity of the entire LTA community at the time I cannot name everyone individually, but special shout-outs are deserved for Dwayne, for everything he did in Val Thorens; and my flat-mates, Seb and Yoran, who not only were very helpful in the aftermath of the operation, but put up with crutches clunking through the apartment for months on end with incredibly good humour. I also should thank the physios who (quite literally) got me back on my feet, particularly George in London; Damien, Théo and Anna in Geneva; and Chris in Glasgow.

Having had no office-mates at all for the first year and a half of my PhD, by my second year I would have been glad to share an office with anyone with a pulse. Instead (other than one French cheese of mysterious origin that *greatly* outstayed its welcome in the CERN office), I've had same fantastic office-mates: Elliot (reliably in the office before me with a conversation topic ready); Zef; Bruno (whose vouching for the quality of the `lwtnn` package led to investigations that would end up producing Section 5.6); Ethan; Fan; and Ivo (who I will, one day, beat at table football). I also greatly enjoyed the sharing the office with several summer students, as well as others with whom I've overlapped with more briefly. I never quite managed to share an office with Jamie or Harriet, but particularly in the isolated first year and a half, it was good to know I was working in parallel with someone, even if our paths only met at conferences, Collider Physics zoom lectures and online French classes. Also a huge thanks to my "virtual" office mates at Imperial in the "PhD struggles" group chat, (soon-to-be Dr.) Ben and (Dr.) Alex: what a journey.

Finally, my family. My sister (who, yes, got to call herself a doctor over a year before me) and my parents, who always encouraged me to study and ask questions, and helped me through all the highs and lows of the last four[1] years.

[1] Or rather, twenty-seven.

Declaration

The research presented in this thesis is the result from my own work within the Experimental Particle Physics group in the School of Physics and Astronomy at the University of Glasgow and has not been submitted for another qualification to this or any other university.

Part I is introductory, and therefore does not (unless explicitly stated otherwise) contain any novel research. All research in Part II was carried out by the author, unless otherwise explicitly specified. Chapters 7 and 8 in particular (though this is also true to a lesser extent of Chapters 5 and 6) were carried out within the contexts of the GAMBIT community and the ATLAS collaboration respectively.

The lone scientist locked in their ivory tower with a blackboard is a thing of the past in modern particle physics: in the twenty-first century, particle physics is a highly collaborative science. Research on the scale of the LHC would not be possible otherwise.

To particularly highlight some of the areas where others played a role, which are also made clear throughout the main text where relevant:

- The guidelines paper reprised in Section 5.8 was a community effort (as it had to be), though I was the co-lead of the group and listed as the corresponding author on the publication.
- As made clear in Section 7.3, the implementation of the GMSB SUSY-model, and the COLLIDERBIT-only results for this model were carried out by the wider GAMBIT community.
- Though it proved to be a sort of "Theseus's thread-safe projection system", with almost all aspects of the original code being replaced, the starting point of the design of the thread-safe RIVET projection system presented in Section 7.5 was inspired by a previously existing failed attempt to complete the same task.
- The ATLAS analysis presented in Chapter 8 was conducted as part of the wider ATLAS collaboration. I was the main analyser during the period of finalising the analysis strategy and unblinding. It makes use of the tools that have been developed by the ATLAS collaboration over the last twenty years: therefore, it may be more helpful to view the content of Sections 8.2 – 8.3 as additional introductory material. Furthermore, the analysis builds on earlier ATLAS studies in the same channel: this is mentioned throughout the Chapter where relevant.

Note that the formal use of the first person plural does *not* at any point imply that the work was done by someone other than the author.

Part I

Introduction

Chapter 1

The Standard Model and beyond

1.1 The Standard Model

With the discovery of the Higgs boson in 2012 [1, 2] all the particles predicted by the theory known as "The Standard Model of Particle Physics" (SM) – built around the symmetry group SU(3) × SU(2) × U(1) – have been discovered. These particles, as illustrated in Figure 1.1, consist of four spin-1 bosons, 12 spin-$1/2$ matter particles (split into 6 quarks and 6 leptons), and the spin-0 Higgs boson. Of the 12 matter particles, just three – the electron, the up quark, and the down quark – make up virtually all the matter that people see in their everyday lives.

The SU(N) symmetry groups represent the unitary (hence U) transformations of an N-dimensional sphere, without any reflections (which gives the S). These groups can be represented by $N \times N$ unitary, complex matrices of determinant one. In addition to SU(3) × SU(2) × U(1) symmetry, the SM also obeys constraints common to all quantum field theories (QFTs). Notably, this means it conserves a combination of three $\mathcal{Z}_2$ symmetries collectively known as CPT-symmetry [3]: C representing *charge-conjugation*, i.e. the "flipping" of all charges; P representing *parity*, the inversion (or "mirroring") of all spatial dimensions; and T representing *time reversal.* These symmetries need not be conserved individually – even if it may seem counter-intuitive that physics would, for example, run differently in a mirror. One particularly interesting case is combined CP-symmetry, which maps matter on to anti-matter.

1.1.1 The Higgs, symmetry-breaking and electroweak interactions

The Higgs potential

The Higgs-boson-only part of the SM Lagrangian is a quartic polynomial:

$$\mathcal{L}_{\text{Higgs}} = \partial_\mu \phi^\dagger \partial^\mu \phi - \mu^2 \phi^\dagger \phi + \frac{\lambda}{2} \left(\phi^\dagger \phi\right)^2 \ , \tag{1.1}$$

for a 2-component complex ϕ. When the coefficient of the quadratic term μ^2 is negative, as in the equation above, this distribution takes on a so-called "Mexican Hat" shape, a 2D slice of which is illustrated in Figure 1.2. In this scenario, rather than a single minimum corresponding to the vacuum state of the theory, there is a "ring" – more properly a manifold – of vacua at $|\phi| = v$. The value of v is the vacuum expectation value, or *vev*, which has been experimentally measured to be 246 GeV [5].

If we rewrite our theory around one of these minima (e.g. $\begin{bmatrix} v & 0 \end{bmatrix}$), then the U(1) symmetry of equation 1.1 (which will become a SU(2) × U(1) when the electroweak bosons are added) has been *spontaneously broken*, and rather than a pair of massless bosons, we have a massive boson – which we will eventually call the Higgs

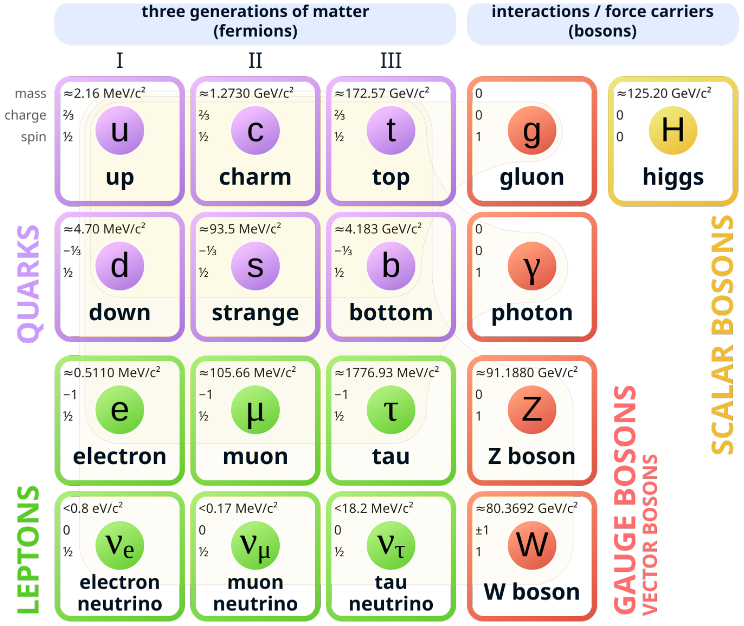


Figure 1.1: The fundamental particles of the Standard Model. Adapted from Reference [4].

boson, h – and a massless *Goldstone boson*, typically denoted χ.

Electroweak bosons and the Higgs Mechanism

The SM electroweak interactions are based on the symmetry group SU(2) × U(1), which has three SU(2) generators[1] T^i, and one U(1) generator Y. This corresponds to four *massless* gauge bosons: $SU(2)$ gauge bosons labelled W^1_μ, W^2_μ, W^3_μ, and B_μ, the $U(1)$ gauge boson[2]. This means that the covariant derivative of SU(2) × U(1), D_μ, is given by

$$D_\mu = \partial_\mu + ig_2 W^i_\mu T^i + ig_1 B_\mu Y \; , \tag{1.2}$$

where g_1 and g_2 are the $SU(2)$ couplings, and Y is the generator of the U(1) group (which takes value $1/2$ for the Higgs-boson) [6]. If we insert this covariant derivative into equation 1.1, then the spontaneous symmetry breaking affects the electroweak SU(2) × U(1) group also. This means that the four electroweak bosons we observe after symmetry breaking in fact emerge as

$$\begin{bmatrix} W^+_\mu \\ W^-_\mu \\ Z^0_\mu \\ A_\mu \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{g_2}{\sqrt{g_1^2+g_2^2}} & -\frac{g_1}{\sqrt{g_1^2+g_2^2}} \\ 0 & 0 & \frac{g_1}{\sqrt{g_1^2+g_2^2}} & \frac{g_1}{\sqrt{g_1^2+g_2^2}} \end{bmatrix} \begin{bmatrix} W^1_\mu \\ W^2_\mu \\ W^3_\mu \\ B_\mu \end{bmatrix} , \tag{1.3}$$

[1] For a Lie group (such as SU(2)), generators are members of the Lie algebra corresponding to the group. In more practical terms, they are a set of matrices T such that all elements of the group (and only elements of the group) can be generated via $\exp\left(i\phi_j T^j\right)$, for some phases ϕ_j.

[2] Traces of these bosons can be found in the names of their supersymmetric counterparts, such as the "bino" – see Section 1.4.

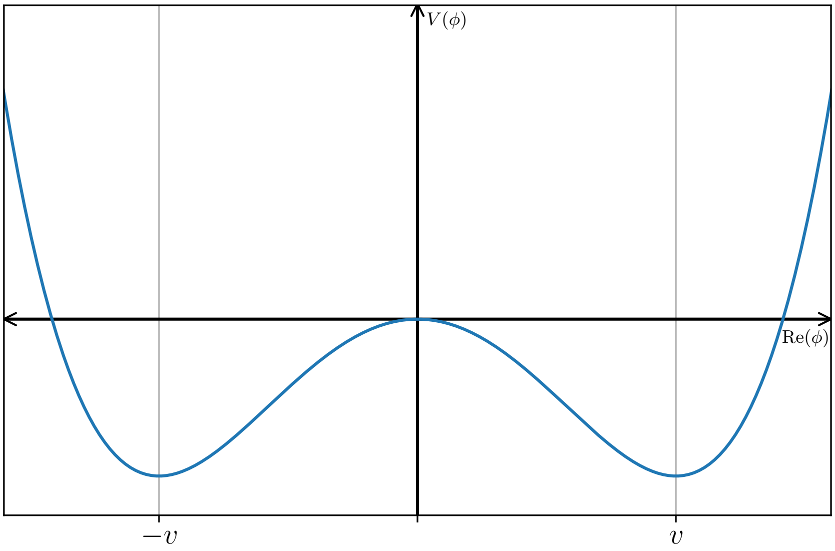


Figure 1.2: The 2D slice along the real plane of the "Mexican Hat" shape of the quartic Higgs potential, when the quadratic term is negative as in Equation 1.1. Note the two vacua, which if we could display the entire complex range of ϕ, would join to form a circular vacuum manifold.

where we can identify the Weinberg angle θ_W, defined by $\sin\theta_W = g_1/\sqrt{g_1^2+g_2^2}$. Because there is a residual U(1) symmetry the photon (A_μ) remains massless, but the remaining thee electroweak bosons have masses, which are related by $\cos\theta_W = M_W/M_Z$.

One final (and important) question is why we do not also observe the massless Goldstone boson. By introducing the gauge fields, we have an additional local symmetry which gives us a free gauge "choice". It is possible to fix this choice such that the Goldstone boson χ disappears [7]. This procedure of spontaneously breaking a symmetry and introducing a gauge field to provide a massive vector boson and eliminate the massless (and unobserved) Goldstone boson(s) is known as the Higgs mechanism [8–10].

Fermions and their masses

One of the unique features of the weak force is that it is not invariant under parity, the symmetry that flips spatial dimensions but not time: a phenomenon first observed in β decay [11]. This makes writing down the fermionic terms in the SM Lagrangian slightly more complicated than just copying in the Dirac equation. When decomposed into left- and right-handed spinors[3], the Dirac mass term becomes $m\bar{f}f = m\bar{f}_L f_R + m\bar{f}_R f_L$. However (once the gauge bosons are added) this term is not gauge invariant.

So is it impossible to write a fermionic mass term consistent with the gauge symmetries of the SM? Once again, the Higgs field comes to our rescue. There is a Yukawa interaction between fermions and the scalar Higgs field, which takes the generalised form [12]:

$$\mathcal{L} = \ldots - y_f\left(V_{ij}\bar{f}^i_L\phi f^j_R + U_{ij}\bar{f}^i_R\phi f^j_L\right)\ , \tag{1.4}$$

where y_f is the coupling strength for fermion f, and U and V are unitary matrices that allow for mixing across fermion generations i and j. This may look like a simple interaction vertex, but the existence of a

[3] Handedness here refers to the chirality of the fermion: in strict mathematical terms, which representation of the Poincaré group it transforms in. Parity symmetry (discussed earlier) makes left-handed spinors right-handed and vice-versa. For massless particles, chirality and helicity (the sign of the projection of a particles spin onto its momentum) are the same.

non-zero Higgs vev means that this can produce a mass term. As an example for first generation leptons (remembering that neutrinos are *technically* massless in the SM), we have:

$$\mathcal{L} = \ldots - \frac{y_e}{\sqrt{2}} \left(\begin{bmatrix} \bar{\nu}_{eL} & \bar{e}_L \end{bmatrix} \begin{bmatrix} 0 \\ v \end{bmatrix} e_R + \bar{e}_R \begin{bmatrix} 0 & v \end{bmatrix} \begin{bmatrix} \nu_{eL} \\ e_L \end{bmatrix} \right) = \ldots - \frac{y_e v}{\sqrt{2}} \left(\bar{e}_L e_R + \bar{e}_R e_L \right) \ , \tag{1.5}$$

where we can see the emergence of the electron mass term $\frac{y_e v}{\sqrt{2}}$. Similar terms exist for the other leptons and the quarks[4]. This feature of the Higgs boson – or more specifically, its vev – is why "popular science" presentations of the Higgs boson describe it as "giving all particles mass", or even more crudely, as the so-called "God Particle" [13].

Fermion mixing, the CKM matrix and $CP-$violation

The flavour eigenstates of the SM quarks are not necessarily exactly the same as the mass eigenstates obtained from the Yukawa coupling to the Higgs vev. To account for this, we use the mass eigenstates as the fields that we write down, and absorb the difference in the charged current electroweak interaction [3], which we write as:

$$-i\frac{g_W}{\sqrt{2}} \begin{bmatrix} \bar{u} & \bar{c} & \bar{t} \end{bmatrix} W_\mu \gamma^\mu \cdot \frac{1-\gamma^5}{2} \cdot \underline{\mathbf{V}}_{\mathrm{CKM}} \begin{bmatrix} d \\ s \\ b \end{bmatrix} , \tag{1.6}$$

where the Cabbibo-Kobayashi-Maskawa matrix (CKM matrix or $\underline{\mathbf{V}}_{\mathrm{CKM}}$) [14, 15] maps the mass eigenstates onto the flavour eigenstates (d', s', b') as

$$\begin{bmatrix} d' \\ s' \\ b' \end{bmatrix} = \underline{\mathbf{V}}_{\mathrm{CKM}} \cdot \begin{bmatrix} d \\ s \\ b \end{bmatrix} = \begin{bmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{bmatrix} \cdot \begin{bmatrix} d \\ s \\ b \end{bmatrix} . \tag{1.7}$$

That the CKM matrix is three-dimensional may seem merely incidental – a simple consequence of the existence of three generations in the quark sector. But the unitarity constraint on a 3×3 matrix allows for the presence of a complex phase parameter, which cannot be present for a 2×2 matrix. This complex phase, typically denoted δ, allows for a limited amount of CP-violation within the SM [6].

Once we allow for neutrinos to have mass (as discussed in Section 1.1.3), an equivalent role is played by the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix for lepton neutrinos [6].

1.1.2 Quantum Chromo-Dynamics

QCD is the part of the Standard Model that deals with "color", the three-dimensional "charge" that holds together protons and neutrons, carried by gluons and quarks. The three axes of this charge are denoted by *red*, *blue* and *green* which, while physically meaningless, allow easier discussion and depiction that just labelling them numerically.

The resulting phenomenon is known as the "strong nuclear force", and is one of the four known fundamental forces. Not only is it significantly stronger than electromagnetism, the weak force and gravity; but because of the self-coupling of the massless mediator (the gluon) it has the unique feature that the force between two bare point color-charges *increases* with their separation. This can be seen in the scaling behaviour of α_S, the coupling strength of the strong force:

$$\alpha_s\left(Q^2\right) = \frac{4\pi}{\left[11 - \frac{2}{3}n_f\right] \ln\left(\frac{Q^2}{\Lambda^2}\right)} \ , \tag{1.8}$$

[4] The up-type quarks are slightly more complicated, due to the "choice" that the vev is $\begin{bmatrix} 0 & v \end{bmatrix}$ not $\begin{bmatrix} v & 0 \end{bmatrix}$.

where Q is the energy-scale of the interaction, Λ is a cut-off for renormalisation, and n_f is the number of available quarks at a given Q [7]. Notably, as the energy-scale increases (and hence the length-scale decreases), as long as there are 16 or fewer quarks – which is always the case in the SM – the coupling decreases.

One consequence of this is *asymptotic freedom.* At very small distances, the strong force becomes sufficiently weak that quarks may be considered approximately free, and α_S is sufficiently small that it becomes possible to carry out calculations using perturbation theory [7].

Conversely, as quarks get further apart, α_S increases, and *confinement* becomes the most impactful consequence of QCD. Put crudely, this means it is impossible for free color-charges to exist below a confinement scale corresponding to approximately 1 fm. One consequence of this is the showering and hadronisation that will be discussed further in Section 3.2; it also explains why we do not see such phenomena for the top-quark, whose lifetime is shorter than the timescale implied by the confinement scale.

1.1.3 Neutrino masses

Technically, SM neutrinos are all massless. However, the observation of neutrino oscillations [16,17] – the fact that solar neutrinos change flavour as they travel through space – disproves this, as it shows that there are mass eigenstates of differing masses. The oscillation also allows for the violation of individual lepton-number conservation.

Although massive neutrinos are technically beyond-the-Standard-Model (BSM) particles – given their very low mass[5] and how they do not directly impact most of the higher energy theories studied at the LHC – it has become increasingly common to irrigorously refer to the SM + three very slightly massive neutrinos, as an adjusted form of "the Standard Model". This convention – which would not include the discovery of a heavy Majorana neutrino, as predicted by the See-Saw model to explain the very low mass of neutrinos [6] – will be adhered to throughout the rest of this thesis.

1.2 Why we need to go beyond the Standard Model

The SM is often cited as one of history's most precisely verified scientific theories – for example, the measured value of the electron's anomalous magnetic moment ($g_e - 2$) agrees with the SM theory prediction at more than 10 significant figures [19]. Despite its successes, however, there are several outstanding issues that the SM alone cannot explain, and therefore a new theory – a *Beyond-the-Standard-Model* (BSM) theory – is required.

Dark matter

The motion of galaxies (and other astrophysical and cosmological observations) suggests that the universe contains a significant amount of matter that does not interact with light – hence *dark matter* (DM). While there are explanations that could be compatible with the SM – such as large numbers of very small black holes, or modified theories of gravity [20] – astrophysical observations constrain these, suggesting a likely explanation of a new particle or particles which would have to be massive and not interact electromagnetically. Many of the more plausible theories have the DM particle interacting with the rest of the SM only via the weak force (leading to WIMPs, Weakly Interacting Massive Particles), or via the Higgs boson – "Higgs-portal" scenarios. Higgs-portal DM allows for a completely disjoint set of BSM gauge fields, coupled to the SM only

[5] The KATRIN experiment has placed an upper limit of just 0.45 eV on the electron neutrino mass, using precise measurements of the β-decay of tritium [18].

because both sets of fields have mass. While the DM candidate does not necessarily have to be completely stable, this would help it evade other cosmological constraints.

There are experiments that search directly for DM particles, such as LUX-ZEPLIN [21] or XENON [22]. These experiments typically rely on looking for the recoil of a nucleus in a detector substance as it interacts weakly with DM passing through the Earth. However, even with the existence of direct detection experiments, it is also possible the DM particle could be produced in a collider. Indeed, colliders are sensitive to possible types of DM – such as Higgs-portal scenarios – that are not accessible via traditional direct-detection experiments.

After being produced at or near the interaction point of a collider, DM particles will likely pass through the detector unnoticed. This means that the "smoking gun" signature for DM at a collider involves missing transverse energy (E_T^{miss}) – i.e. momentum in the plane perpendicular to the beamline appearing to not be conserved – in quantities much larger than can be expected from SM processes (e.g. any process that produces neutrinos).

The hierarchy problem and naturalness

Physical theories are often viewed with concern when their (dimensionless) parameters are spread over many orders of magnitude, or when they rely on parameters being very fine-tuned. In other words, very small changes to the parameters change the observables significantly. While some have criticised this as merely an "aesthetic" choice [23], many theorists claim this can be put on a sound basis using Bayesian statistics [24].

The most obvious application of this to the SM is in the (surprisingly small) mass of the Higgs boson, which is impacted by loop corrections to the Higgs-boson self-energy [25]. If we only assume the SM holds slightly beyond the energies we have been able to reach at colliders, this does not present fine-tuning problems – though it would imply the presence of some other new physics only slightly beyond our current energy frontier. However, if we assume that the SM is "all there is" (at least up to some very high energy indeed), then it must hold at energy scales where the loop corrections become very large, where it is very hard to keep the Higgs mass at 125 GeV without very careful fine-tuning of the corrections to ensure they cancel [3].

If we include neutrino masses in our definition of the SM, then it also becomes relevant to ask why the neutrino masses are so much lower than the rest of the fermion masses. In the minimal SM+massive neutrinos, this would imply that the Yukawa Higgs-neutrino couplings are orders of magnitude smaller than for the other fermions. One popular solution is the previously mentioned See-Saw model.

Gravity

Of the four fundamental forces, the only one not included in SM is gravity. In principle, the gravitational force could be mediated by a spin-2 *graviton*, though in practice treating this as tensor-boson in a QFT proves very difficult, because the theory is non-renormalizable; and the natural scale of the theory, the Planck scale, is so large. Among other approaches, string theory is one attempt to resolve these problems [26].

CP-violation

To explain the preponderance of matter over anti-matter in the universe, we need to break the exact matter vs anti-matter symmetry. As discussed in Section 1.1, this is CP symmetry, and while there is some CP-violation in the SM (in the CKM and PMNS matrices, as discussed above), there is not enough to explain the current dominance of matter in the universe [3].

The muon $g-2$ measurement

At the start of this section we noted how precisely the SM prediction of the electron's anomalous magnetic moment had been verified. However, the same cannot be said for the anomalous magnetic moment of its heavier sibling, the muon.

The best experimental measurements of $g_\mu - 2$ come from the Muon $g-2$ experiment at Fermilab [27], building – in some cases quite literally, given the re-use of experimental equipment – on the results from its predecessor at Brookhaven [28]. The measured value differs from the SM value calculated by the Muon $g-2$ Theory Initiative by 5.1σ [29,30] – typically the threshold used to announce "discoveries" in particle physics.

However, alternate theoretical approaches to calculating the SM value using lattice QCD result in a much lower tension [31]. Depending on whether and how the theoretical tension is resolved, there may be significant questions for the SM to answer.

Lepton universality and flavour anomalies

Lepton universality descibes the fact that the interaction of the SM leptons with the electroweak bosons are all the same – the only difference between the leptons being in their mass, which originates from their Yukawa coupling to the Higgs. For a brief period during 2021 and 2022 – though building on 2.5σ results dating back to LHC Run 1 [32] – lepton universality appeared to be broken. The LHCb experiment measured a series of greater than 3σ discrepancies [33] in a variable called R_{K^*}, which measures the ratio of the decay rate of the B^0 meson to muons over its decay rate to electrons, which should be unity if lepton universality holds. Combinations by theorists of all the 3σ anomalies suggested that the combined discrepancy was over 5σ [34].

This measurement was later corrected due to a missing background, which left the measured value of R_{K^*} consistent with the SM [35]. However for a number of other flavour-physics variables related to lepton universality there exists a growing tension with the SM when combining at LHCb, Belle-2 and BaBar [36]. This particularly motivates theories that introduce new quark-lepton interactions, such as leptoquarks.

What do solutions look like?

Individual BSM theories are typically put forward to resolve at least one of the above problems: for example, a new BSM particle could be a DM candidate; new processes involving BSM particles and vertices may allow for more CP violation; or higher-order Feynman diagrams involving new particles may make adjustments to anomalous SM measurements such as $g_\mu - 2$ or R_{K^*}. Not all such theories can be directly probed at the LHC, but the rest of this chapter will describe several that can, and that are pertinent to the rest of the thesis.

1.3 Vector-like quarks

Vector-like quarks (VLQs) are hypothetical particles predicted by a range of high-energy (HE) BSM physics models, for example composite- or little- Higgs models [37,38] or Grand Unified Theories (GUTs) built around the symmetry group E_6 [39,40]. Some of these HE theories resolve (at least partially) the hierarchy problem, and VLQs themselves can be sources of additional CP violation. They are "quarks" because they transform in the same QCD triplet that SM quarks do; and "vector-like" because their left-handed and right-handed components transform in the same way under gauge transformations.

Most models predict at least two additional quarks, the B- and T- quarks[6], and some models further predict the existence of the X and Y quarks (charges $^5/_3$ and $^{-4}/_3$) [41]. Given an SM Higgs field, the new

[6] These are sometimes denoted VLB and VLT to make the contrast with the SM b- and t-quarks more obvious.

VLQs can exist in singlets, doublets or triplets, and these scenarios have different implications for the best motivated branching fractions.

With their vector-like EW interactions, VLQs can couple to a SM boson (W, Z or Higgs), and in most cases the HE models suggest that VLQs should couple preferentially to third-generation SM quarks [42]. Accordingly, the majority of LHC searches have used simplified models with coupling exclusively to the third-generation quarks [43]. The space of models with couplings to first- and second- generation quarks has been much less explored; though large couplings to multiple generations would introduce a mechanism for (unobserved) flavour-changing neutral currents.

An effective Lagrangian for adding VLT's to the SM is

$$\begin{aligned}\mathcal{L}_{\mathrm{VLT}} = \kappa_T \Bigg\{ & \sqrt{\frac{\zeta_i \xi_W^T}{\Gamma_W^0}} \frac{g}{\sqrt{2}} \left[\bar{T}_{L/R} \not{W}^+ d^i_{L/R}\right] + \sqrt{\frac{\zeta_i \xi_Z^T}{\Gamma_Z^0}} \frac{g}{2c_W} \left[\bar{T}_{L/R} \not{Z} u^i_{L/R}\right] \\ & - \sqrt{\frac{\zeta_i \xi_H^T}{\Gamma_H^0}} \frac{g}{2c_W} \left[\bar{T}_{L/R} h u^i_{L/R}\right] \Bigg\} + h.c. - m_T \bar{T} T + i \bar{T} \not{D} T \ , \end{aligned} \tag{1.9}$$

with the same equation, *mutatis mutandis*, describing vector-like B quarks [44]. u^i and d^i represent the ith generation of SM up or down-like quarks; ζ, ξ and κ are constants of the theory, and Γ^0 represents the partial width of the decay to the given boson[7].

The first key feature to note is that the mass terms do not come from a Yukawa coupling to the Higgs as they do for SM quarks. This is important, as it allows them to avoid the otherwise strong constraints on the existence of a fourth generation of quarks from Higgs production [45].

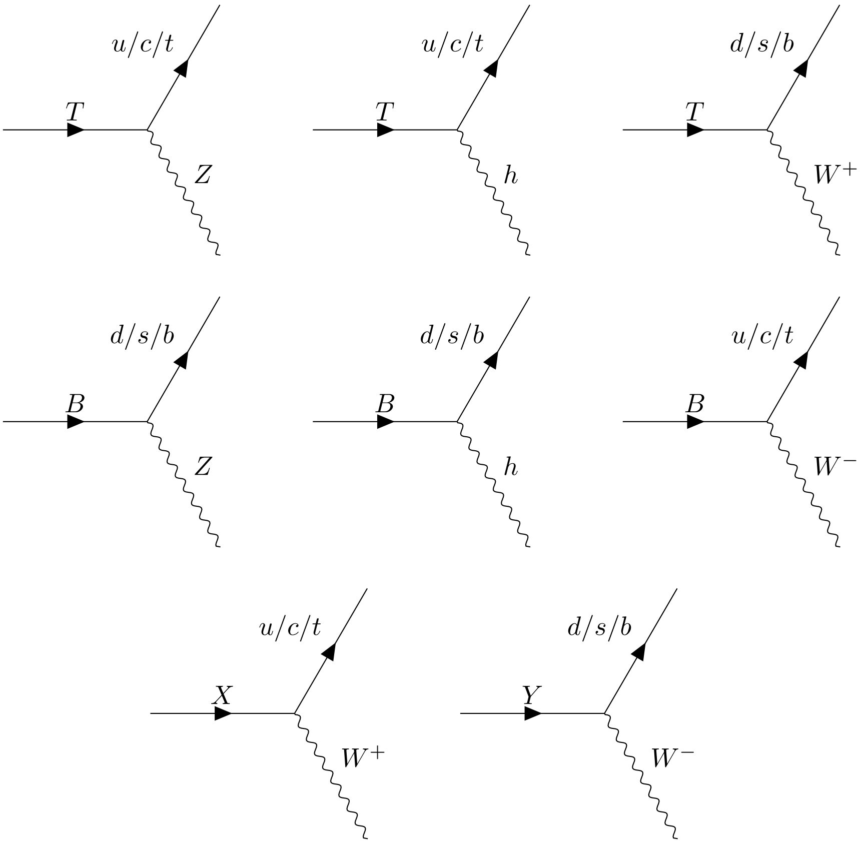


Figure 1.3: The "vector-like" vertices of VLQs, i.e. their couplings to the Z, W and Higgs bosons. Equivalent diagrams exist for all the anti-VLQs. These diagrams also illustrate the possible decay modes.

[7] The partial width is not an independent quantity, and is calculated by the event generator.

Another useful feature of this parametrisation is that ζ and ξ correspond directly to useful physical observables[8]. The $\xi^T_{W/Z/H}$ and $\xi^B_{W/Z/H}$ are the relative branching ratios of the VLT and VLB to W, Z, and Higgs bosons respectively, and $\zeta^{T/B}_i$, is the relative branching ratio to the ith generation of quarks. Due to the preferential coupling to third generation quarks, many phenomenological models set $\zeta_3 = 1$ and $\zeta_1 = \zeta_2 = 0$. These parameters are not independent: they must all be positive, real and in $[0, 1]$; and all branching fractions must sum to one. The κ parameters control the strength of the electroweak vertices, while the "quark-like" (quarky) QCD interactions of VLQs are hidden in the covariant derivative term, as for SM quarks.

A popular alternate reparametrisation of the Lagrangian is given by Reference [46]:

$$\begin{aligned}\mathcal{L}_{\mathrm{VLT}} = i\bar{T}\not{D}T - m_T\bar{T}T - h\left[\bar{T}\left(\hat{\kappa}^T_L P_L + \hat{\kappa}^T_R P_R\right) d_i + h.c.\right] \\ + \frac{g}{2c_w}\left[\bar{T}\not{Z}\left(\tilde{\kappa}^T_L P_L + \tilde{\kappa}^T_R P_R\right) u_i + h.c.\right] + \frac{\sqrt{2}g}{2}\left[\bar{T}\bar{\not{W}}\left(\hat{\kappa}^T_L P_L + \hat{\kappa}^T_R P_R\right) d_i + h.c.\right] \,, \qquad (1.10)\end{aligned}$$

where once again we have simplified by only showing the T quark, $P_{\mathrm{L/R}}$ are projection operators, and where various κ parameters are coupling parameters of the theory. While this formulation does not have the advantage of easily explaining parameters in terms of physical observables, unlike the previous formulation, it can handle renormalisation and hence be extended to NLO diagrams. For LO generation, the authors of Reference [46] provide conversion formulae back to the old model[9].

1.3.1 VLQ production

The two main production modes for VLQs which are accessible at the LHC are single- and pair-production. LO pair production (of a VLQ and an anti-VLQ) is dominated by 2→2 QCD diagrams which emphasise the "quarkiness" of VLQs – the same diagrams as shown in Figure 1.4 also exist for SM quarks. EM pair production via a photon instead of a gluon is also possible, but is negligible at leading order. Conversely, single VLQ production is dominated not by the "quarkiness" of VLQs, but by their coupling to the vector- and Higgs-bosons, as illustrated in Figure 1.5. Because $\alpha_{\mathrm{QCD}} \gg \alpha_{\mathrm{EWK}}$, 2→3 diagrams with an additional QCD vertex – such as in Figures 1.5d and 1.5e – cannot be ignored, and will in fact often dominate because they do not necessarily require a very off-shell SM boson.

1.3.2 VLQ event generation

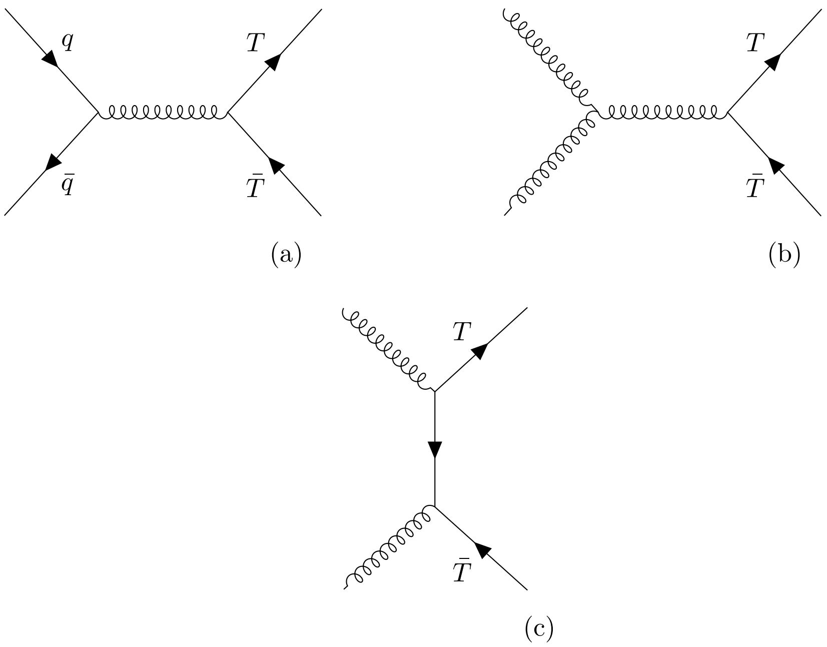
FeynRules [48] and UFO [49] files have been provided for both the Lagrangians in equations 1.9 and 1.10. This means that general-purpose event generators (which will be introduced more thoroughly in Chapter 3) such as MadGraph5_MC@NLO [50] (used in multiple Atlas analyses searching for singly-produced VLQs [51]) and Herwig [52, 53] (used by the Contur phenomenological study [54]) are able to produce events based on these models. The FeynRules files should also be compatible with Gambit's Gum [55], which will be introduced in Chapter 4. Custom implementations of pair-production of VLQs also exist in the Protos generator [56], which has been used to generate the signal samples in multiple Atlas pair-produced VLQ searches [57, 58].

1.3.3 Status of VLQ searches

LHC Run 1 searches for VLQs focussed primarily – though not exclusively [59] – on pair-production, and placed lower mass limits on primarily B or T production, in the 500 − 800 GeV range at both CMS [60] and

[8] At tree-level, given that this effective Lagrangian cannot be used for calculations at NLO or higher.

[9] As an introductory activity, I also wrote a FeynRules file using the input parameters from the first model, but implemented underneath with the couplings from the new – see Reference [47].



(a)

(b)

(c)

Figure 1.4: LO 2→2 VLQ pair production. Though the T VLQ is illustrated, the same diagrams exist for all four VLQs. EM diagrams exist, but are often neglected because they are negligible.

Atlas [61] depending on the branching fractions studied.

Run 2 searches were more extensive, and considered a wide range of final states for both single- and pair-production. Some searches gave mass limits; whereas others (noting that for single-production at any given mass the coupling κ can be tuned down to reduce the cross-section) give upper limits on the coupling as a function of mass. For pair-production, the Atlas partial-Run 2 pair-production combination placed lower limits of 1.31 TeV on the VLT mass and 1.03 TeV on the VLB mass across all possible branching fractions [62]. The full Run 2 CMS combination [43] of VLB pair-production searches extended the limit on the VLB mass to 1.5 TeV.

For single production of VLTs, the best limits come from the Atlas Run 2 single-production combination [63]: which excluded the singlet-VLT[10] at $\kappa = 0.5$ below 2 TeV; and the doublet-VLT[11] at the same value of κ below 1.4 TeV. 2D plots were also presented showing the mass exclusion as a function of κ. Notably, this combination did not include any searches targetting the $T \to Wb$ decay mode, which helps motivate the Atlas analysis presented in Chapter 8, targetting the one-lepton channel of this decay.

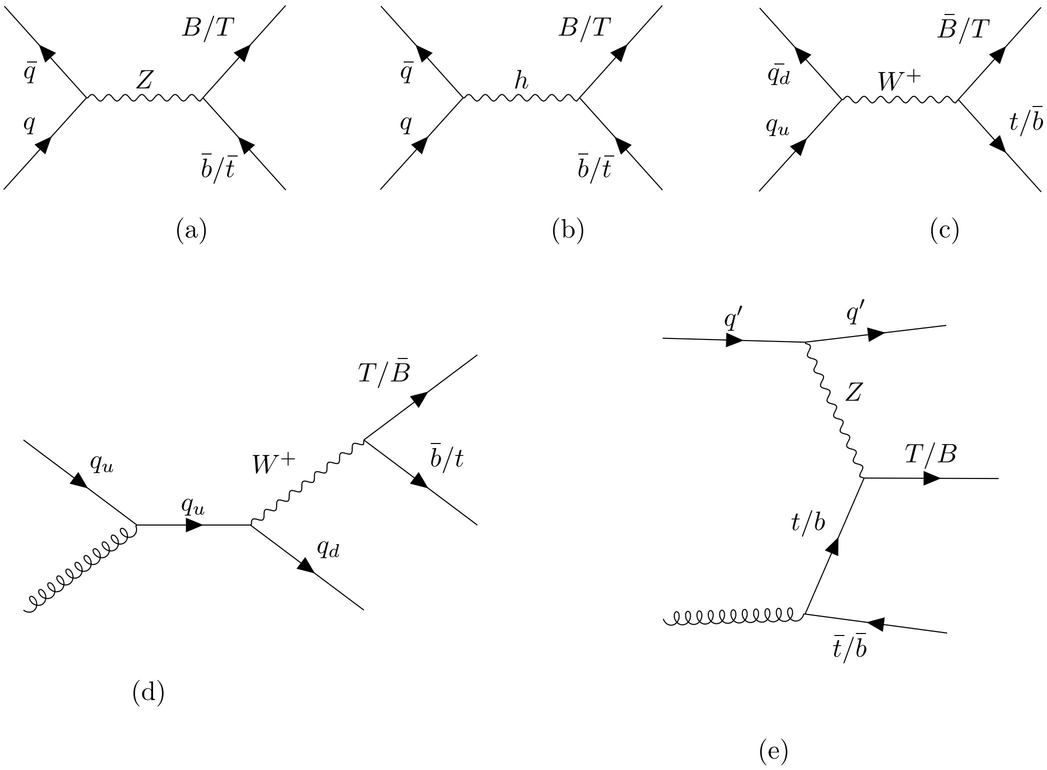
VLQs are also a popular model to examine phenomenologically. Notably, a series of scans using Contur [54] showed that SM LHC measurements have significant excluding power for regions of the VLQ parameter space, often complementary to those regions best excluded by direct searches.

1.3.4 More complex phenomenology and signatures

Unsurprisingly, as the LHC experiments have ruled out progressively more of the VLQ parameter space, phenomenologists have found new ways to extend the model to evade exclusion. A particularly popular development has been to add an additional scalar field in addition to one-to-four of the VLQs. If this scalar is lighter than the VLQ, it provides an additional decay channel to the typical vector- or Higgs-boson plus SM

[10] i.e. $\xi_W^T : \xi_h^T : \xi_Z^T = {}^1\!/\!_2 : {}^1\!/\!_4 : {}^1\!/\!_4$.

[11] i.e. $\xi_W^T : \xi_h^T : \xi_Z^T = 0 : {}^1\!/\!_2 : {}^1\!/\!_2$.



(a) (b) (c)

(d)

(e)

Figure 1.5: Examples (not exhaustive) of VLQ single production diagrams. We explicitly limit ourselves to the case where the VLQ only couples to third generation quarks.

quark channels, reducing – in some cases to zero – the branching fraction visible to existing LHC searches [64]. The scalar particle may even be a dark-matter candidate [65], a scenario that may be of particular relevance to global fitting tools like Gambit. These models, though very interesting, are beyond the scope of this thesis, but will likely be targetted by multiple searches during LHC Run 3.

1.4 Supersymmetry

Though it is not at the centre of this thesis, it is necessary to give a brief overview of Supersymmetry (SUSY), both as perhaps the most famous of BSM theories, and as it is at least tangentially relevant to some of the material in Chapters 5, 6, and 7. SUSY adds an additional symmetry to the Standard Model, with every SM boson receiving a fermionic *superpartner*, and every fermion receiving a bosonic *superpartner*. The bosonic fermion partners are given names prepended with an "s" – *selectron*, *smuon*, top-*squark etc*; and are denoted with the same symbol as their SM cousins with a tilde "˜": e.g. $\tilde{\nu}_e$ for the electron-sneutrino.

Meanwhile. the fermionic boson partners are given a *-ino* suffix, such as *higgsino* or *gluino*. Because of symmetry breaking, the mass states of the partners of the gauge- and higgs- bosons can mix [66]. This means that instead of single $\tilde{h}$, $\tilde{Z}$, $\tilde{W}^+$ and so forth mass eigenstates; we have electroweakinos: both *charginos* ($\tilde{\chi}_i^\pm$) and *neutralinos* ($\tilde{\chi}_i^0$) consisting of linear combinations of higgsino, wino, etc.

SUSY is attractive as a BSM theory because it solves many of the problems outlined in Section 1.2. For example the "Lightest Supersymmetric Particle" (LSP), most often the lightest neutralino $\tilde{\chi}_1^0$, is an excellent DM candidate [66]; and sparticle loop diagrams exactly cancel the contributions of their SM partners to the Higgs mass, "explaining" the hierarchy problem [25].

R-parity is a $\mathbb{Z}_2$ symmetry that prevents mixing between particles and sparticles, and means that sparticles can only be pair-produced [67]. Adding this symmetry is a simple and elegant way to prevent supersymmetry predicting proton decay, a process that has never been observed, and also has the benefit of making the LSP DM candidate stable. Not all SUSY searches assume that R-parity is conserved, but it is the first and most obvious simplification. Collider searches for R-parity violating models should be particularly careful that the model is not already completely ruled out on proton-lifetime or cosmological grounds, though the latter concern can also apply to R-parity conserving theories.

1.4.1 Searching for SUSY

A key observable in searches for (R-parity conserving) SUSY is the large missing-transverse energy contribution from a pair of LSPs that pass undetected through the entire detector. Significant additional jet activity is also a commonly used signature (for example, in the search reinterpreted in Section 5.5). In recent years, in part due to the succession of null results for more conventional signatures, SUSY searches for "long-lived particles" – i.e. particles that decay within the detector but significantly displaced from the IP – have also gained popularity. This includes so-called "stealth-SUSY" models that require a new non-supersymmetric scalar particle to "hide" the normal E_T^{miss} signature [68]; as well as R-parity violating models where the LSP lifetime means it decays within the detector [69].

1.4.2 Supergravity and GMSB

Supergravity is an extension of Supersymmetry that provides a spin-$^3/_2$ super-partner called the *gravitino* ($\tilde{G}$) to the spin-2 *graviton* (G), the force-mediator of the gravitational force [70]. One scenario for symmetry breaking in supergravity, relevant to Section 7.3, is Gauge-Mediated Supersymmetry Breaking (GMSB), which leads to suppressed gravitino masses of $\mathcal{O}$(1eV) [71]. Then the gravitino is the LSP, and the next-to-lightest supersymmetric particle (NLSP) is typically a neutralino [72], and may still be a DM candidate despite being unstable. The characteristic collider signature of this theory is the pair production of NLSP particles (the theory is still R-parity conserving), which decay into the gravitino (which becomes E_T^{miss} from the detector's perspective) and an SM particle, such as a Higgs-boson or a photon.

1.4.3 Simplified models (and their potential drawbacks)

One problem in searching for supersymmetry is the number of free parameters. Even the "simplest" complete form of SUSY, the Minimally Supersymmetric Standard Model (MSSM), has 105 new free parameters [66]. As this is a computationally intractable space, collider searches typically reduce this to just two or three parameters, using so called "simplified models" [73]. The most common parametrisation, as we will see in Chapters 5 and 6, is a pair of sparticle masses. Techniques used to make the simplifications include decoupling SUSY effects other than the process being studied and forcing some derived quantities (such as branching ratios) to fixed values. Most simplified models can be easily shared and reproduced using a SUSY Les Houches accord (SLHA) file [74].

For some sparticles in some simplified models, the mass exclusion limits (from LHC results) have now been pushed so high that the theories – if found to be true at higher masses – would still have some "naturalness" concerns [75].

Whether or not these simplified models give an accurate picture of the exclusion of the more general model is a question asked by phenomenological tools such as GAMBIT (see Chapter 4) and Mastercode [76], as well as ATLAS's own pMSSM scanning efforts [77]. While the full 105-dimensional MSSM space is still intractable, by sacrificing some accuracy and taking computational shortcuts not available to the original analyses, these tools can survey the $7-19$ dimensional space of further constrained versions of the MSSM such as the CMSSM or the pMSSM.

Chapter 2

The LHC, the Atlas detector, and object reconstruction

2.1 The Large Hadron Collider

The Large Hadron Collider (LHC), built inside the same 27km tunnel as the preceding Large Electron-Positron collider (LEP), is the world's most powerful particle collider, with centre-of-mass energies $\sqrt{s}$ reaching 13.6 TeV during Run 3. Around the LHC ring, there are four interaction points (IPs) where collisions may occur, and at each of these is situated a detector: CMS [78], LHCb [79], Alice [80], and Atlas [81]. All the experimental results in this thesis will come from the Atlas detector, and specifically from proton-proton collisions during Run 2 (2015 – 2018), when the centre-of-mass energy was 13 TeV.

All the protons involved in every collision during Run 2 started in a small bottle of hydrogen gas stored near the start of the linear accelerator Linac 2. The journey from this bottle to colliding at a small fraction below the speed of light is shown by the light-grey arrows in Figure 2.1. After being ionised and fed into Linac 2, the protons are further accelerated by the Proton Synchrotron (PS), and then the Super Proton-Synchrotron (SPS). The SPS injects a proton beam into the LHC in both directions.

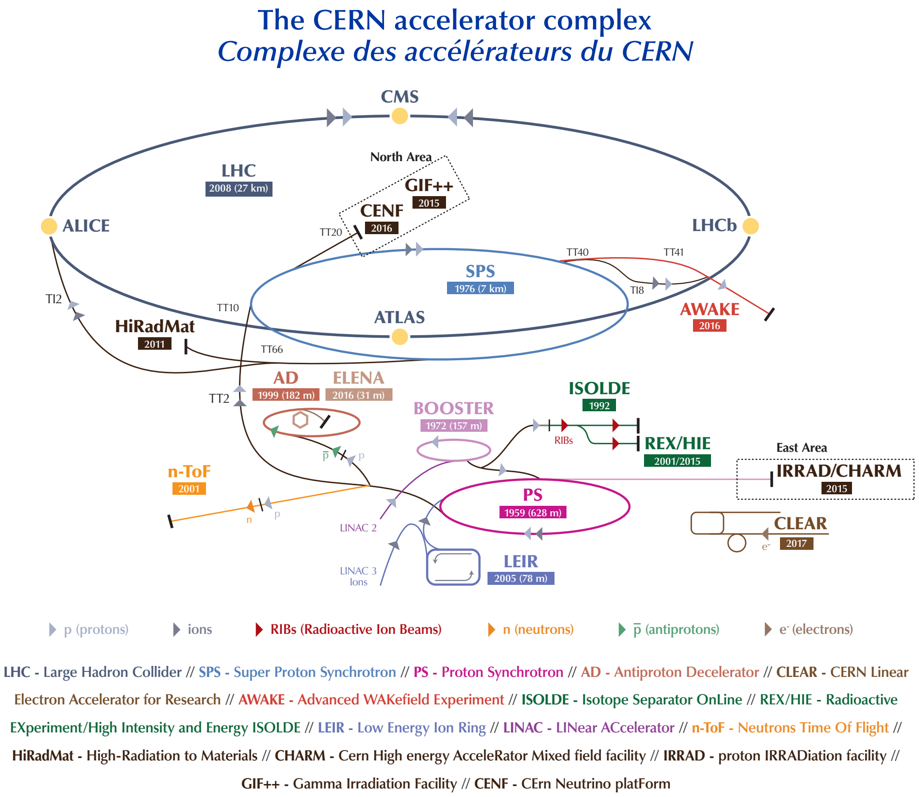
As Figure 2.1 shows, throughout this journey, accelerated protons may be diverted to other experiments at the CERN site, such as the various experiments at the Antiproton Decelerator (AD) [82]. The LHC, intermediate accelerators, and experiments all need extensive vacuum and cyrogenics facilities in order to keep the beam-line clear and maintain cooling for the super-conducting magnets [83].

Once in the LHC, protons are accelerated up to a final energy of 6.8 TeV (6.5 TeV during Run 2) each – corresponding to $\sqrt{s}$ =13.6 TeV (13.0 TeV during Run 2) for a head-on collision – and bunches of $1-2\times10^{11}$ protons [84] (depending on the Run period), separated by only 25 ns, collide at Interaction Point (IP) in the centre of the Atlas detector.

At any given instance, assuming the beams have a Gaussian profile of width σ_x and height σ_y, the activity delivered to the detector can be quantified by the luminosity, L, given by:

$$L = f\frac{N^2}{4\pi\sigma_x\sigma_y} \ , \tag{2.1}$$

where f is the frequency of the bunch-crossings, and N is the number of protons in each bunch [85]. This may seem an inconvenient choice of measurement, but when integrated over a whole LHC run, it has the benefit that multiplying the cross-section for a process by the integrated luminosity gives the Poisson mean expected number of events. By the end of Run 2, approximately 140 fb^{-1} of usable pp physics data was recorded by the ATLAS detector [86]. While multiple methods of measuring the luminosity are employed, the most important is the LUCID-2 detector [87]. LUCID-2 is a Cherenkov detector located approximately 17m beyond the ATLAS IP, based on radiation hardened photomultiplier tubes that record inelastic p-p collisions during bunch crossings.



The CERN accelerator complex
Complexe des accélérateurs du CERN
CMS
LHC
2008 (27 km)
North Area
GIF++
2015
CENF
2016
TT20
ALICE
LHCb
TT40
TT41
SPS
1976 (7 km)
TI8
TI2
TT10
ATLAS
AWAKE
2016
HiRadMat
2011
TT66
AD
1999 (182 m)
ELENA
2016 (31 m)
TT2
BOOSTER
1972 (157 m)
ISOLDE
1992
RIBs
REX/HIE
2001/2015
East Area
IRRAD/CHARM
2015
n-ToF
2001
p̄
p
n
p
LINAC 2
PS
1959 (628 m)
CLEAR
2017
e⁻
LINAC 3
Ions
LEIR
2005 (78 m)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive EXperiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n-ToF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials // CHARM - Cern High energy AcceleRator Mixed field facility // IRRAD - proton IRRADiation facility // GIF++ - Gamma Irradiation Facility // CENF - CErn Neutrino platForm

Figure 2.1: The CERN accelerator complex as of 2018, at the end of Run 2; taken from Reference [88]

2.2 ATLAS detector: co-ordinates and geometry

The ATLAS detector is a general-purpose detector which covers nearly the entire 4π solid angle around the IP. It is forward-backward symmetric – unlike, for example, LHCb [79] – and approximately cylindrical in shape. The terms "barrel" and "end-cap" are used to describe detector components that are parallel to, respectively, the curved side or flat ends of the cylinder.

By convention we use a three-dimensional co-ordinate system using (r_T, η, φ). This sits between classical spherical (r, ϑ, φ) and cylindrical (r_T, ϑ, z) co-ordinates, using the transverse radius r_T from cylindrical co-ordinates, but using an angular measurement instead of a cylindrical "z" co-ordinate. Figure 2.2 illustrates these variables, and how they relate to a conventional Euclidean (x, y, z) system. The diagram uses a displacement vector $\vec{r}$, though in practice this scheme is most commonly used to describe momenta.

The pseudo-rapidity η is preferred to the polar angle ϑ, though the two are related as

$$\eta \equiv -\ln\left[\tan\left(\frac{\vartheta}{2}\right)\right] , \tag{2.2}$$

meaning that η can take values from $-\infty$ to ∞; although in practice any particles emitted from the IP with $|\eta| \gtrsim 5$ will continue down the beampipe and fall out of detector acceptance. For massless particles, η becomes

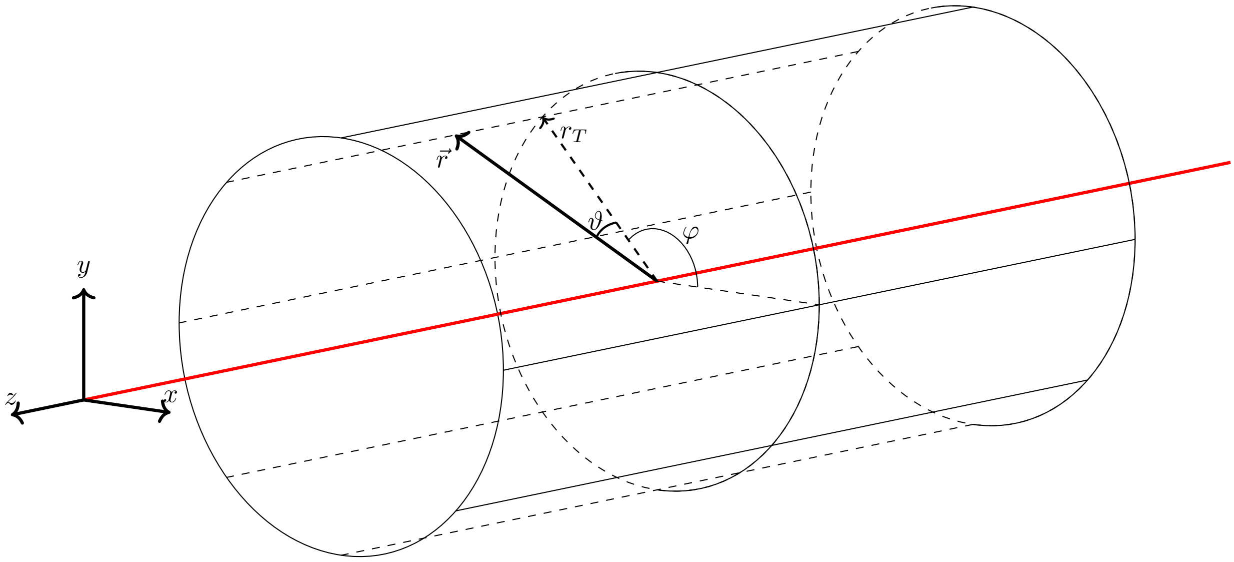


Figure 2.2: Illustrating the co-ordinates used to describe Atlas. By convention, the x axis points towards the centre of the LHC ring, and the y axis points directly up. The polar angle ϑ is converted to a pseudorapidity using equation 2.2.

equivalent to the rapidity

$$y = \frac{1}{2} \ln \left[\frac{E + p_z}{E - p_z} \right] \ , \tag{2.3}$$

where E is the total energy of the particle. Though it is a much harder quantity to measure than a simple angle, y is physically interesting, because rapidity differences between objects are invariant under Lorentz transformations along the beamline [85].

2.2.1 Angular differences and ΔR

Angular differences between momenta are often very physically interesting observables. Differences in φ or η alone are straightforward to calculate. However, an angular difference accounting for both η and ϕ is more complicated. The conventional measure in the LHC era is

$$\Delta R \equiv \sqrt{\Delta\varphi^2 + \Delta\eta^2} \tag{2.4}$$

which is explicitly not spherically symmetric. Occasionally, a similar quantity using the rapidity instead of the pseuodrapidity may be used

$$\Delta R \equiv \sqrt{\Delta\varphi^2 + \Delta y^2} \tag{2.5}$$

though we will use the former unless otherwise specified. The simple angle between two vectors in the plane they share – easily calculable via the inner product of the normalised vectors – is not used.

2.3 The inner detector

Some of the most interesting physics takes place close to the IP and, as such, the *inner detector* (ID) [81] contains the greatest density of detection capabilities. It covers the pseudorapidity range $|\eta| < 2.5$ and from the IP out, consists of silicon pixel detectors, the semiconductor tracker (SCT), and finally the transition radiation tracking detectors (TRT) – as illustrated in Figure 2.3. A 2 T magnetic field runs axially through the entire inner detector in order to ascertain the charges of particles that leave tracks.

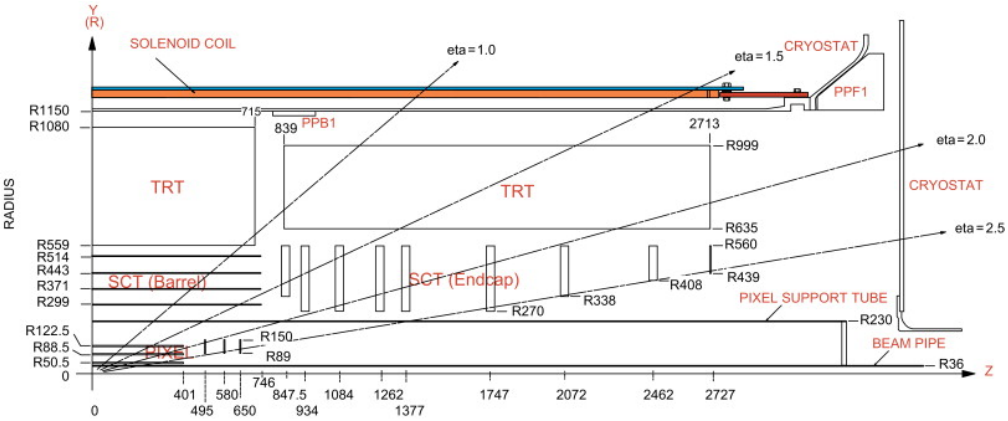


Figure 2.3: The components of the Atlas inner detector. Adapted from Reference [91].

The smallest pixels on the innermost layers of the *pixel detector* are just 40μm × 200μm, and outgoing particles pass through three pixel-layers [89]. The high-granularity of these detectors allows the identification of primary and secondary vertices for tracks with as little as 0.1 GeV of transverse momentum, which plays an important role in object identification – for example, the presence of a displaced decay vertex is a crucial element of b-tagging [90], which will be important for the analysis presented in Chapter 8.

The *SCT* [91] consists of layers – four in the barrel and nine in each endcap – of silicon strip detectors, each layer of which provides a 2D-location. Charged particles are detected as they create electron-hole pairs in the semiconductor material: under an electric field in the p-n junction formed from doped semiconductor layers, the electron and hole drift in opposite directions, and charge is collected on the opposite ends of the junction [85]. The SCT also contributes significantly to vertexing, which will be discussed further in Section 2.7.1.

The *TRT* [92] is made of approximately 300 000 4mm-diameter drift tubes, which are filled with Xenon gas. Charged particles (and photons close to a Xenon atomic transition energy) excite Xenon atoms in the tube, which create a charge that can be measured. By differentiating between low (300 eV) and high (6 keV) thresholds, the TRT also contributes to particle identification, as different particles at the same momentum have different (and known) energy deposition profiles.

All the components in the Inner Detector also have the additional engineering challenges of needing to be radiation resistant, maintaining performance for several years of run time despite lifetime radiation doses of up to 500 kGy [89], and doing all this with minimal matter, in order to minimise the number of extra particles that may end up being scattered. The radiation and interaction lengths (defined in the following section) of the ID are $0.1 - 1.0$ and $0.05 - 0.35$ respectively [93], increasing with pseudorapidity, as high-η particles need to travel through more material.

2.4 Calorimeters

The energy of most of the particles arising from a collision in the Atlas detector are measured in one of the two calorimeters. Calorimeters work by stopping (or at least slowing down as much as possible) the incoming particles and then measuring the energy released as they are stopped. This allows us to eventually reconstruct the full four-momentum of various physics objects. As particles are stopped, the electromagnetic or strong interactions cause cascades of additional particles ("showers") to be created, all of which in turn are eventually also stopped in the calorimeter and have their energy measured.

All the calorimeters used in the Atlas detector are *sampling calorimeters*, which measure the energy

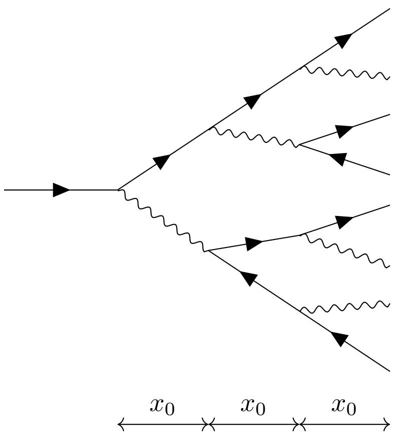


Figure 2.4: A simplified example of an EM shower. The initiating particle in this case is an electron, but a similar (albeit deeper penetrating) structure is formed for photon-initiated showers too. Note the definition of the radiation length x_0.

deposited by layering dense "stopping layers" with sensor layers that can measure the energy deposits [85]. ATLAS uses two calorimeters: Electromagnetic (EM), for electrons and photons; and hadronic for hadrons.

2.4.1 The Electromagnetic Calorimeter

Electrons are much lighter than hadrons; and photons are massless. This means the EM interactions they undergo with the electrons in the material stopping layer make them easier to stop than their hadronic counterparts, and so the EM calorimeter is the first detector after the ID as the particles head outwards. Energy loss for incoming electrons is dominated by the brehmstralung emission of photons, and electron-positron pair production dominates for incoming photons [85]. This lends itself to the simple model of an EM shower illustrated in Figure 2.4, which also illustrates the concept of a *radiation length*, x_0 – the expected distance through the stopping material that a particle travels in order to emit an additional showered particle.

The ATLAS EM Calorimeter uses liquid argon (LAr) as the sensing material and lead as the stopping material [94]. As illustrated in Figure 2.5, it is comprised of the barrel (EMB) calorimeter and endcap (EMEC) calorimeters, and the transition region between the two leads to a small η range where electrons may escape identification. The barrel section comprises of at least 22 interaction lengths, and there are at least 24 interaction lengths in the end-cap [81].

The barrel region uses a complex "accordion" structure to give complete φ-symmetry, apart from two very small gaps along the $z = 0$ plane [81]; and the granularity in $\Delta\varphi \times \Delta\eta$ is as little as 0.025×0.025, to ensure precise energy resolution for particles that can be matched to tracks in the ID. In the analysis presented in Chapter 8, this will allow the momentum of the leptonically decaying W-boson to be accurately reconstructed.

2.4.2 The Hadronic Calorimeter

Hadrons will pass through the EM calorimeter largely unhindered, but are eventually stopped in the hadronic calorimeter. Hadronic showers are more complicated than their EM counterparts: a veritable zoo of different hadrons can be involved, and they can all undergo a range of different strong-force related interactions with the nuclei in the stopping material. This leads to showers which are wider, which have an *interaction length* typically much longer the analogous radiation length for EM showers, and which leave behind a lower fraction of detectable energy [85], leading to a lower energy resolution for Hadronic Calorimeters as compared to EM

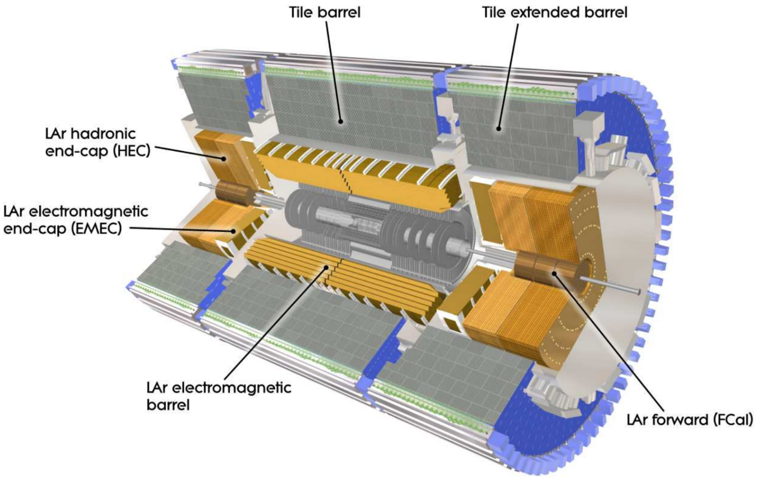

Tile barrel
Tile extended barrel
LAr hadronic end-cap (HEC)
LAr electromagnetic end-cap (EMEC)
LAr electromagnetic barrel
LAr forward (FCal)

Figure 2.5: The arrangement of the ATLAS calorimeters. Adapted from Reference [81], Figure 1.3.

Calorimeters.

ATLAS uses several hadronic calorimeters to cover different parts of the solid angle around the IP, all illustrated in Figure 2.5. The first of these is the tile calorimeter, with a central section covering approximately $|\eta| < 1$, and an extended barrel region covering approximately $0.8 < |\eta| < 1.7$ [95]. The stopping material is low-carbon steel, and the sensor material is a plastic scintillator. These are 14 mm and 3 mm thick respectively, and arranged in 3 layers of "tiles", with a $\Delta\varphi \times \Delta\eta$ granularity of 0.1×0.1 in the two inner layers, and $0.1{\times}0.2$ in the outermost layer.

The two other hadronic calorimeters use the same LAr sensor layer used by the EM calorimeter. The two hadronic endcap calorimeters (HECs) use a copper stopper layer [94], and cover an η range of $1.5 < \eta < 3.2$, providing some overlap with both the barrel and forward calorimeters. The forward calorimeter (FCal) also has some EM sensitivity, with a first stopper layer made of copper followed by two made of tungsten [81]. It covers a rapidity range of $3.1 < \eta < 4.9$. The density of the FCal also helps protect the muon wheels from the high levels of radiation that would otherwise seep through at these high $|\eta|$ values.

2.5 Muon systems

Given that electrons are stopped very easily in the EM calorimeter, it is perhaps a little surprising that we need to dedicate the entire outermost layer of the detector to sensing their heavier siblings, muons. The factor of approximately 200 mass difference with the electron makes brehmstralung rates much lower for muons, so they pass almost unhindered through the EM calorimeter; and, because they do not interact with the strong force, the hadronic calorimeter will not stop them either.

There are four ATLAS detectors dedicated to muon detection: two for precision tracking of muons, and two primarily for triggering and providing a second co-ordinate. Their positioning in the detector is illustrated in Figure 2.6, along with the magnet systems that, as in the ID, aid in charge reconstruction.

Precision tracking of muons is provided by the *Monitored Drift Tubes* (MDTs) and *Cathode Strip Chambers* (CSCs) [81]. In both the end-cap and the barrel, there are three layers of precise tracking sensors. These all consist of MDTs, apart from the innermost layer of the barrel end-cap, where CSCs are preferred, because they perform better with the higher background rates in this area of the detector. MDTs are 3 cm-wide tubes filled with a mixture of Argon and carbon dioxide pressurised to aproximately three atmospheres, and with a 50 μm-diameter wire running through the axis of the tube at a potential of over 3 000 V collecting ionised electrons, and therefore detecting the passage of a muon. CSCs are also based around high-voltage

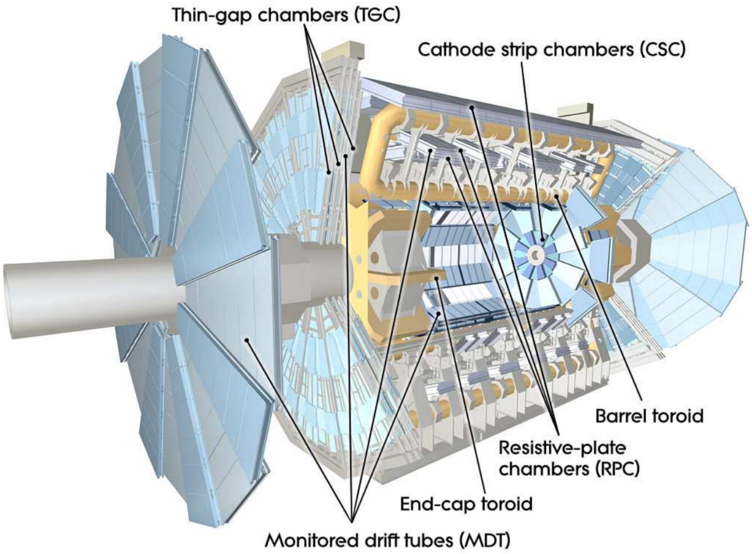

Thin-gap chambers (TGC)
Cathode strip chambers (CSC)
Barrel toroid
Resistive-plate chambers (RPC)
End-cap toroid
Monitored drift tubes (MDT)

Figure 2.6: The ATLAS muon system. From Reference [81], Figure 1.4.

wires suspended in an Argon-CO_2 dioxide mixture, but with many wires running parallel in one chamber.

MDTs and CSCs provide very precise spatial information, but their response time is too slow relative to the 25 ns gap between bunch-crossings to be useful for triggering decisions [85]. Lepton triggers are crucial to many analyses, including the one that will be described in Chapter 8. Therefore, two spatially less-precise but faster-in-response systems are also used: *Resistive Plate Chambers* (RPCs) in the barrel, and *Thin Gap Chambers* (TGCs) in the endcaps [81]. These also provide information about the position of the muon orthogonal to that provided by the precision tracking chambers.

The RPCs, like the MDTs in the barrel section, are also arranged in three concentric layers. An RPC detector consists of two resistive plates with a strong electric field between them, separated by a layer of gas[1]. Particles passing through the gas ionise electrons, which causes an "avalanche" between the plates, which is detected. The RPC's cover the pseudorapidity range $|\eta| < 1.05$.

There are up to seven layers of TGCs (but only two at the most forward positions). These chambers are similar in principle to the CSCs, but optimised for faster response and arranged to provide orthogonal information. The TGCs cover pseudorapidity range $1.05 < |\eta| < 2.7$, although triggering decisions can only be made on $1.05 < |\eta| < 2.4$.

2.6 Triggers

As mentioned in Section 2.1, proton bunches collide every 25 ns – 40 MHz in frequency – at the centre of the ATLAS detector. Unfortunately, if we were to save the full data-output of the detector every 25 ns, this would not only lead to the saving of a large amount of data consisting of proton-proton near-misses and low-energy glancing blows, but would also far outstrip the data-storage capacity of even the CERN Data Centre.

Instead ATLAS (and the other LHC detectors) rely on a "trigger" system that makes a very quick decision on whether or not to save the data from the collision. For Run 2, the trigger consisted of two stages: *L1*, a hardware-based trigger that reduced the event rate to 100 kHz, and the *HLT*, a software-based trigger that reduced this further to approximately 1kHz [96]. This means that the ATLAS experiment – before any analysis-specific preselection or other requirements – already rejected the data from more than 99.99% of the

[1] A 94.7/5/0.3 mixture of $C_2H_2F_4$/Iso-C_4H_{10}/SF_6 [81]

Run 2 bunch-crossings.

The L1 hardware trigger uses information from a subset of detectors: the calorimeters discussed in Section 2.4, forming the L1Calo system; and the two trigger-specific muon systems described in Section 2.5, forming the L1Muon system. From this information, the triggers seek to identify signatures corresponding to high-p_{T} leptons, photons or jets; as well as large amounts of missing- or total-transverse energy. It will always make a decision within 2.5 µs [81]. The information from L1Calo and L1Muon is used together to run topological algorithms in the L1Topo system [97]; and then a final decision is made by the Central Trigger Processor, which combines outputs from L1Calo, L1Muon and L1Topo using simple logical operations. The CTP is also responsible for applying "dead-time" – effectively stopping the recording of data when the detector read-out for a previous event is still ongoing and no more events can be accepted. During Run 2, the total inefficiency due to dead time was approximately 1% [98].

The HLT uses information from all the components of the detector subsystem, and may even carry out basic event-reconstruction – a coarser version of what will be outlined in the next section. The HLT runs fast algorithms to be able to reject some events very quickly, before resorting to more CPU-expensive checks. The HLT software is based on the same Athena [99] framework that is used in the other computationally expensive parts of the Atlas experiment, such as reconstruction (Section 2.7) and Monte-Carlo event generation (Chapter 3). The HLT aims to make a decision in $\mathcal{O}$(100ms).

At this point, the Atlas Data Acquisition (DAQ) system allows the splitting of events into multiple streams: some events go into the physics stream, but others can be written into specialist debugging or calibration streams. Within the physics stream, there is a "menu" of different trigger chains, that allow analyses to pick events that have been triggered on different physics objects; for example, events with at least one electron, or events with significant amounts of missing transverse energy. It is important to note that these triggers are not 100% efficient, particularly for objects with energies or momenta only slightly greater than the trigger requirement. For some analyses the trigger inefficiencies will make a significant difference to their final acceptances. For example, the muon trigger used for the 2016 data of the analysis in Chapter 8, which consisted of the 26 GeV and 50 GeV single muon triggers arranged in an "or" configuration[2], had efficiencies of only 60%–70% for barrel muons during Run 2, with worse performance at lower p_{T} [100].

2.7 Reconstruction

After data has passed through both levels of the trigger, the next stage in preparing it for analysis is reconstruction. This is the process of taking digitised hits from the detector and turning them into data on which statistical analyses can be performed: turning energy in the EM calorimeter into electrons or photons with known momentum and energy; using ID tracks and hadronic calorimeter energy to define hadronic jets; and so forth.

Unlike the trigger, which needs to run at the rate of event production (i.e. 40 MHz), after initial processing at the Tier 0 computing site at CERN, the reconstruction process can be run "offline". Like the Monte-Carlo event generation processes that will be outlined in the next chapter, this utilises the resources of the Worldwide LHC Computing Grid (WLCG) [101, 102] to distribute the computing workload all around the world, while ensuring that all Atlas users can still access all the relevant outputs [103, 104].

Reconstruction starts with the building of low-level objects such as tracks from the ID; useful physics objects such as jets or electrons are built out of these in turn. These steps often includes corrections for detector effects – a correction needs to be made between measuring the energy deposited in the calorimeter

[2] For 2015 data, a 20 GeV trigger was used instead of 26 GeV; and for 2017 and 2018 an additional, looser, 60 GeV trigger was placed in a logical "or" with those used in 2016.

by a particle and obtaining the total energy of the “particle-level” object.

2.7.1 Low-level objects

Tracks and vertexing

Charged particles move through the ID in a helical shape due to the magnetic field. Hits that meet the required voltage thresholds in the various layers of the ID are combined to form *tracks*.

First, hits in the detector are grouped into clusters, as even a single charged particle is likely to ionise multiple pixels: a neural net is used to estimate if clusters are likely to originate from individual or multiple particles. Even in a cluster, it is typically still possible to approximate the intersection point of the particle with the detector using the distribution of deposited charge.

Track “seeds” are formed from space-points in three different layers of the ID. Track candidates are then built by adding in other detector points using a combinatorial Kalman filter [105]. Because it is limited to $|\eta| < 2.0$, the TRT is not used to help form seeds or track candidates. Inevitably, several of the track candidates will use the same space-points: to resolve these ambiguities, a score is assigned to tracks incorporating the χ^2 of the fit of the track (from the Kalman filter), the expected cluster multiplicities of the clusters in the track, and a downweighting for tracks that contain holes, i.e. the track entirely absent from one intermediate tracking layer. Tracks are then iteratively rejected or adjusted until a final set of tracks is obtained. The full detail of the algorithms and selection criteria used in track reconstruction can be found in Reference [106].

An important initial task is identifying the “primary vertex” of the event, the point in space where the two protons collided. Vertices are found by extrapolating tracks back to the point where they meet; and the vertex with the highest associated $\sum\limits_{\text{tracks}} p_{\mathrm{T}}^2$ is categorised as the primary vertex. Secondary vertices may originate either from pile-up (to be discussed in Chapter 3), or else from long-lived particles that decay only after travelling a short distance through the detector – an effect that will be discussed again in Section 2.7.3 to help identify b-hadrons.

Topological Clusters

Before identifying higher-level physics objects, the energy deposited in the calorimeters is first organised into “topological clusters”, formed from neighbouring calorimeter cells. At this stage, EM and hadronic calorimeter cells are treated alike, and where they meet – the outermost layer of the EM calorimeter and the innermost of the hadronic calorimeter – clusters may cross between them.

For each calorimeter cell, a *cell signal significance* $\zeta_{\text{cell}}^{\text{EM}}$ is calculated according to

$$\zeta_{\text{cell}}^{\text{EM}} = \frac{E_{\text{cell}}^{\text{EM}}}{\sigma_{\text{noise, cell}}^{\text{EM}}} , \tag{2.6}$$

where $\sigma_{\text{noise, cell}}^{\text{EM}}$ is the root-mean-square of the expected pile-up and electrical noise for that specific calorimeter cell in the run-period in question. The EM superscript implies that this is calculated at the EM energy scale, i.e. interpreting the energy deposit in the calorimeter as for photons and electrons, and ignoring the effects discussed in Section 2.4, where hadronic jets deposit less detectable energy in the calorimeter.

“Protoclusters” are then seeded around cells with $|\zeta_{\text{cell}}^{\text{EM}}|$ greater than a threshold S – which has been optimised to 4.0; the protoclusters then grow cell-by-cell by the addition of cells with $|\zeta_{\text{cell}}^{\text{EM}}| > N$, with N chosen to be 2.0. There is an optional cut-off value P, but this is normally set to 0. Note that the use of the absolute value of the cell signal significance means that negative values of $\zeta_{\text{cell}}^{\text{EM}}$ are allowed to contribute to clusters: this allows for some noise cancellation [107]. To form the final clusters, a splitting algorithm is applied to divide overly-large clusters.

To account for various detector biases, a series of corrections needs to be applied to the clusters. These are different for EM and hadronic showers, so an estimate $\mathcal{P}^{\mathrm{EM}}$ of the cluster type is required to weight the corrections. The corrections include calibration terms for hadronic jets (because not all of their energy is detected); as well as out-of-cluster corrections for parts of the energy deposit that are likely to have been cut off from the cluster, and corrections for energy deposits from the physics object that caused the cluster that may have been left in dead material in the calorimeter. A full break-down of the corrections, as well as the noise-terms in equation 2.6 and the clustering algorithms, are given in Reference [107].

2.7.2 Jets

As will be discussed in Section 3.2, bare partons in the final state of diagrams such as Figure 1.5 produce a large "shower" of hadrons. Depending on the p_{T} of the original parton, these showers will be approximately cone-shaped, may be more- or less-collimated, and are described as "jets".

Jet clustering algorithms

The purpose of a jet-clustering algorithm is to take a set of momenta $\{p_i\}$, distributed in the 4π solid angle around the IP, and group these into a set of jets $\{j_i\}$. It is worth noting that exactly the same algorithms that are used to cluster detector signals are also used to cluster the particle-level information from Monte-Carlo event generators; this is largely due to the extensive calibration procedures we will shortly cover, but is also a testament to the granularity of modern detectors. Though Monte-Carlo processes will be covered fully in the next chapter, we will borrow some terminology for this section, as jet clustering is a topic relevant to both areas.

In the LHC era, almost all jets are constructed following some form of sequential recombination algorithm. These require the definitions of distance measures d_{iB} (the distance from jet i to the beamline) and d_{ij} (the distance between jets i and j). At each step, the two momenta with the lowest distance measure between them are merged (typically by summation), unless d_{iB} is smallest, in which case jet i is considered "final" and removed from further consideration. The most commonly used distance measures for LHC studies differ only in the value of a constant n:

$$d_{iB} = \left(p_{T_i}^2\right)^n \tag{2.7a}$$

$$d_{ij} = \min\left[\left(p_{T_i}^2\right)^n, \left(p_{T_j}^2\right)^n\right] \frac{\Delta R_{ij}}{R} \ , \tag{2.7b}$$

where ΔR is defined as in equation 2.5 (i.e. using rapidity not pseudorapidity), R is an approximation of the jet "radius" – increasing it will tend to give fewer, larger jets; and vice-versa. $n = 1$ is known as the "k_T algorithm", $n = 0$ the "Cambridge-Aachen algorithm", and $n = -1$ the "anti-k_T algorithm".

Some previous experiments – such as CDF during Run 1 of the Tevatron [108] – used algorithms that involved drawing cones around momenta or energy deposits. The overwhelming majority of these violated infrared-safety: they could produce different results based on a single additional infinitely-soft or infinitely-collinear splitting. This is dangerous, because IR divergences that cancel could end up in different jets, leading to final states whose cross-sections cannot be formally calculated correctly. Though IR-safe cone-based algorithms can be constructed, such as SISCone [109], the overwhelming majority of clustering in the LHC era have used the three algorithms based on equation 2.7.

Figure 2.7 compares the three jet-algorithms defined using equation 2.7 as well as the SISCone algorithm. Notably, anti-k_T gives very "circular" looking jets, which have practical benefits at LHC experiments, as they are easier to reproduce with the more approximate hardware-based algorithms used in the trigger [85], thus raising the trigger efficiencies.

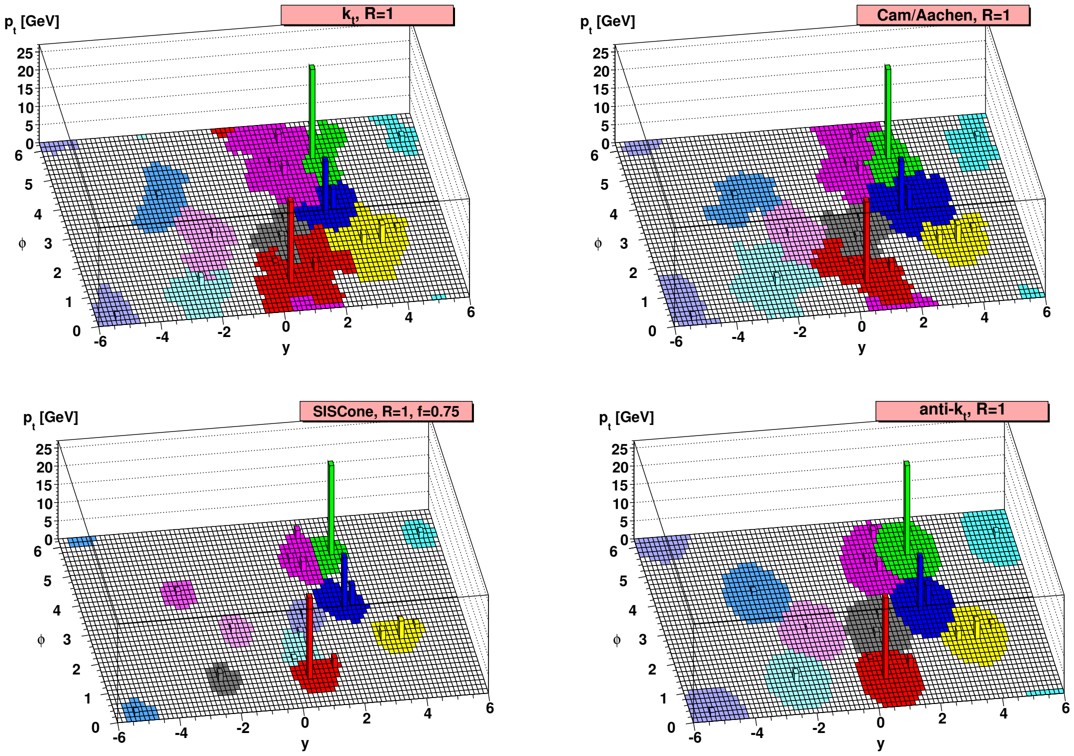


Figure 2.7: Comparison of four different jet clustering algorithms with the same radius parameter, from Reference [110]. Note that anti-k_T produces the most "circular" jets.

Especially in hadron-rich collisions with many inputs, jet-clustering has the potential to be very computationally expensive. In the contexts of LHC experiments, which also have to run other expensive reconstruction and detector simulation processes, this has been significantly ameliorated by the `FastJet` package [111], which implements all the common LHC algorithms at $\mathcal{O}(N^2)$ complexity [112]. For phenomenological codes – introduced *en masse* in Section 4.3 – such as Rivet or ColliderBit, with much lower reconstruction and detector simulation footprints, even $\mathcal{O}(N^2)$ complexity leaves jet-clustering as one of the most time-consuming parts of their workflow. Therefore, minimising the number of times jet clustering needs to be performed provides motivation for the Rivet projection system, which will be explained in Section 4.3.1 (and again in even more detail in Section 7.5).

Ghost-association is one way of taking physics objects that are not inputs to the jet clustering to be clustered and working out which jet they "belong" to. The magnitude of the object's momentum is shrunk to a very low number[3], but the direction is conserved and used as an input to the clustering. Exploiting the IR-safety of modern jet definitions, we know that the final jet will be unaffected by this very soft additional input; but we do see which jet it was clustered into. This is commonly used in Monte-Carlo to establish "true" values of jet tags before machine-learning training.

Sometimes, we may wish to know which jet an object belongs to after jet-clustering has already occurred, and re-clustering may be impractical. In this case, ΔR*-association* can be used. This means associating any object within $\Delta R < R'$ of a jet to that jet. The most straightforward choice of R' is simply the jet radius used in the clustering algorithm; and for the notably circular jets produced by the anti-k_T algorithm, this

[3] How small a number depends on the implementation but, as an example, in Rivet – which uses `double` precision throughout – a factor of 10^{-7} is applied to the object's original momentum.

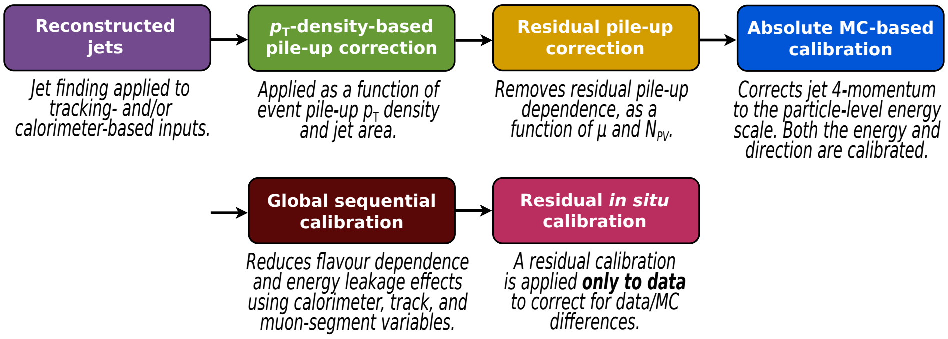

Reconstructed jets
Jet finding applied to tracking- and/or calorimeter-based inputs.
p_T-density-based pile-up correction
Applied as a function of event pile-up p_T density and jet area.
Residual pile-up correction
Removes residual pile-up dependence, as a function of μ and N_{PV}.
Absolute MC-based calibration
Corrects jet 4-momentum to the particle-level energy scale. Both the energy and direction are calibrated.
Global sequential calibration
Reduces flavour dependence and energy leakage effects using calorimeter, track, and muon-segment variables.
Residual *in situ* calibration
A residual calibration is applied **only to data** to correct for data/MC differences.

Figure 2.8: All the stages of jet calibration for EMtopo jets: perhaps the most important is the correction back to the particle-level energy scale. From Reference [114], Figure 2.

often produces results approximately identical to ghost-association (except in the case of overlapping jets). For cases where the associated objects are expected to provide all or almost all of the jet's momentum (e.g. heavy flavour tagging), a more conservative choice of R' (such as half the jet radius) may be used to reduce the rate of incorrectly tagged jets.

Jet reconstruction at Atlas

There are several possible sources of sets of momenta $\{p_i\}$ that can be used for jet clustering at Atlas. One simple option is to use the most basic momentum-like objects available, which are tracks from the ID. These are known as track-jets, and some Atlas analyses (e.g. Reference [113]) use anti-k_T, radius 0.2 track-jets for b-quark identification, as tracks effectively capture the displaced vertices that are the most important features in b-tagging. However, such jets do not leverage the information that comes from the calorimeter, and they cannot account for any electrically neutral particles.

The main option during the majority of Run 2 was to use "EMtopo" jets. These are jets formed by clustering clusters from the combined EM and hadronic calorimeters[4] with positive energy, explicitly without a correction from the EM energy scale to the hadronic scale. Each cluster is treated as a massless pseudo-particle, i.e the clustered 4-momentum is simply

$$p^{\mathrm{EM}} = E^{\mathrm{EM}} \begin{bmatrix} 1 & \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{bmatrix} , \tag{2.8}$$

where the angular location of the topological cluster $(\theta,\ \varphi)$ is calculated using a "centre-of-gravity" approach [114]. These jets then go through a rigorous additional calibration, as illustrated in Figure 2.8: these calibrations correct the jets to the particle-level energy scale (the "Jet Energy Scale" or JES correction) and also account for pile-up and data-Monte Carlo discrepancies. These corrections provide many systematic variations that need to be accounted for in Atlas analyses, and are discussed in full detail in Reference [114].

Improved performance for hadronic jets can be obtained using particle flow – "PFlow" – based jets, which will be used by the analysis presented in Chapter 8. These are constructed using information from both the calorimeters *and* the ID. In basic terms, the PFlow algorithm matches charged tracks in the ID to topological clusters in the calorimeter, and subtracts the energy from the charged tracks from those topoclusters (possibly removing the topocluster altogether). The set of charged tracks and remaining topoclusters are then passed to the jet clustering algorithms. A slightly more detailed illustration of the algorithm is shown in Figure 2.9, showing the modified procedure if the hadronic shower from a particular parton has been split into multiple

[4] Though the EM and hadronic calorimeters are optimised for EM particles and hadrons respectively, recall that the topological clusters are formed across both.

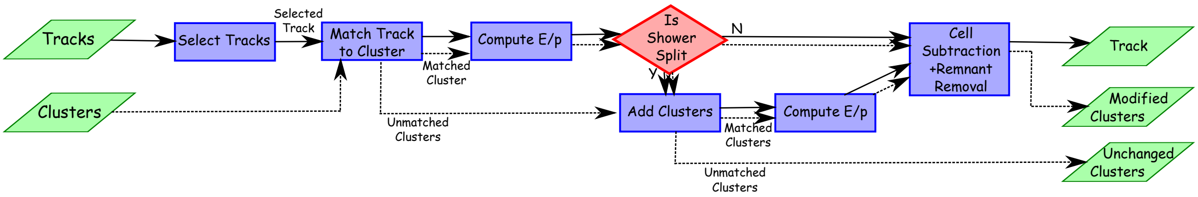


Figure 2.9: A flowchart explaining the PFlow algorithm used in ATLAS for obtaining jets, from Reference [115], Figure 2.

topoclusters; and the full explanation of each step in this diagram can be found in Reference [115]. PFlow jets also have to go through a similar calibration procedure to that discussed for EMtopo jets.

Because the charged components of hadronic jets are dominated by charged pions, the PFlow algorithm assumes that all the newly added momenta are not massless but have mass m_π. This helps PFlow jets perform better than EMtopo jets for tasks like the reconstruction of heavy particle masses. PFlow jets also have a better energy resolution than EMtopo jets, particularly at low-p_{T} and at central η values [115].

Several different jet algorithms and radii were used in analyses during ATLAS Run 2. The majority used the anti-k_T algorithm; and the most common radius parameters were 0.4 (often described simply "small" jets) and 1.0 (likewise "large"). The calibration procedures described above for EMtopo and PFlow jets are very labour-intensive, and need to be rederived for each jet-algorithm and radius parameter. So while in principle each physics analysis could optimise their jet-algorithm parameters like any other cut; in practice only a small subset of these combinations were made available.

To bridge the gap, some analyses (e.g. the analyses that will be reinterpreted in Section 5.6) use reclustered jets: these are made by applying a jet-clustering algorithm not on energy deposits or MC particles, but on already defined (and calibrated) small radius jets; and show similar performance to the original large-radius jets [116]. Because the calibration follows automatically from the small-radius constituents, analyses could pick a range of radii and clustering algorithms, for example variable-radius jets.

After the jets have been obtained, the Jet Vertex Tagger (JVT) [117] may also be used in order to remove low-p_{T} jets that originated from pile-up. Using properties of the tracks associated to the jet, the JVT calculates a probability which estimates whether the jet originated from signal or pile-up. Different JVT working points can be chosen, depending on how important pile-up suppression is to an analysis.

2.7.3 Jet tagging and substructure

Identifying the origin of jets can often add significant experimental sensitivity to an analysis. This works on multiple levels: some analyses may want to distinguish between QCD jets from quarks and gluons (for example Reference [118]); others may want to distinguish jets originating from b- or c- quarks from the light quarks (for example, the ATLAS search described in Section 5.5); and others still may pay less attention to the quark content (other than possibly b-quarks), and seek to distinguish whether the jet originated from a heavy SM particle decaying to quarks, for example a W- or Z-boson decaying into a quark pair.

Flavour-tagging of jets often relies heavily on vertexing information, as b-hadrons (and to a lesser-extent c-hadrons) typically have longer lifetimes, and therefore b- (and c-) jets will appear to originate from a displaced vertex. During Run 2, ATLAS used a series of b-tagging algorithms based on machine learning[5]. MV2c10 and its sibling MV2c20 [119] – which will be pertinent to Section 5.5 – are Boosted Decision Tree (BDT) taggers that use a mixture of vertexing information from the tracker as inputs. These were improved upon by the DL1 series of neural-net taggers, DL1r (used by the analysis presented in Chapter 8) and DL1d [120].

[5] See Section 4.2 for a fuller introduction to machine learning and associated vocabulary.

These took as inputs the outputs of several different, lower level algorithms, such as the secondary vertex tagger SV1 [121], which iterates over all tracks to find the best possible secondary vertex. Note that the inputs to these DL1 algorithms are all based on information that those outside the experiments will struggle to reproduce, given their dependence on the detector geometry and energy scales.

For Run 3, the Graph Neural Network algorithm GN1 [122] (and its imminent successor, GN2), offers an order-of-magnitude increase in performance again (measured by background rejection at fixed efficiencies) over the DL1 series, showing the benefits that using the latest machine learning technology can bring.

For tagging jets originating from heavier particles, typically larger jets – of radius one or reclustered jets of a similar size – are used and *jet substructure* becomes a very important discriminating variable. High-mass SM particles undergoing hadronic decay will decay to different numbers of partons: for example, the W-boson decay $W^+ \to q_u\bar{q}_d$ decays to two quarks; while for the top-quark decaying as $t \to W^+b \to q_u\bar{q}_d\, b$, there are three quarks in final state of the decay. These may of course end up producing separate jets; but, particularly when the decaying particle is produced at high-p_T – the so-called *boosted* regime – they are likely to be collimated. Then, we might expect the energy deposits within the jet to look more "two-pronged" or "three-pronged". This is the objective of jet substructure variables like n-subjettiness [123], whose τ_2 and τ_3 variables approximately measure the "two-prongedness" or "three-prongdness" of jets. Substructure variables are often used as inputs to boosted-object tagging neural nets or BDTs – for example, the tagger that will be discussed in Section 5.7. A similar, if cruder, effect can also be achieved by using a reclustered jet and comparing the transverse momentum and mass of the leading subjets – this approach is taken by the MCBOT tagger which will be described in Section 5.6.

2.7.4 Electrons and photons

Electrons and photons are constructed from a combination of topological clusters in the calorimeters, and tracks in the ID. The clusters must first be identified as EM clusters: this requires that more than 50% of the energy in the cluster is deposited inside the EM (not hadronic) calorimeter, and that the energy deposited in the EM calorimeter is greater than 400 MeV. Electron candidates are formed from ID tracks matched to EM-topoclusters: nearby EM topo-clusters may be added to account for the electron emitting bremsstrahlung radiation before reaching the EM calorimeter [124].

Photons would not be expected to leave tracks in the ID; however, it is estimated that 30% of photons "convert" to e^+e^- pairs in the ID before reaching the EM calorimeter [124]. "Converted" photon candidates are constructed from EM topo-clusters matched to "conversion" vertices in the ID; like for electrons, additional topo-clusters may be added to the candidate if they match the same conversion vertex. The "unconverted" photon candidates are constructed from EM topo-clusters that cannot be matched to ID tracks.

Electron and photon candidates become electron or photon objects based on how (un)likely the candidates where to have been produced by jets, rather than being "true" electrons and photons. This is estimated using a variety of variables which mainly describe the track properties, the shower's development in different layers of the EM calorimeter and the track to topo-cluster matching [125]. For both electrons and photons, several working points (WPs) are available, allowing analyses to trade a purer signal for the cost of reduced overall signal acceptance. The electron energy scale and resolution are calibrated on $Z \to ee$ events; and the photon energy scale and resolution are calibrated on $Z \to ee\gamma$ events.

2.7.5 Muons

The most important low-level objects for muon reconstruction are tracks from the ID and from the MS. The formation of MS tracks begins with short straight-line segments in an individual station of the muon system. After an extensive process of refinement to remove lower quality tracks that share hits with other tracks, combining information from both the precision and trigger muon systems (described in more detail in Reference [126]), a final set of tracks is obtained. These are used to form muon candidates in several ways, depending on the working point being used.

The "safest" form of candidate muons are combined or "CB" muons. These are formed from matching tracks in the MS to tracks in the ID, accounting for energy losses in the calorimeters; and performing a combined fit of a muon trajectory to these tracks. "IO" (inside-out) muon candidates are formed starting with ID tracks, extrapolating to the MS, and looking for MS hits, which may not have been incorporated into an MS track [126].

The `medium` and tighter muon WPs only use CB and IO muon candidates – and even at the `medium` WP, 98% of accepted candidates are CB muons. Other working points (specifically the `loose` and `lowpT`) use additional muon candidate types, for example CT muons, which are built from ID tracks and calorimeter deposits consistent with a minimally ionising particle. These are useful for selecting very low p_{T} muons (down to 3 GeV [126]), because such muons may deposit so much of their energy in the calorimeters that they will not leave a measurable track in the MS: but these WPs will not be pertinent to the remainder of this thesis.

Calibration and resolution measurements are made using $Z \to \mu\mu$ and $J/\psi \to \mu\mu$ events; and validated on $\Upsilon \to \mu\mu$ events (the J/ψ and the Υ mesons are the electrically neutral $c\bar{c}$ and $b\bar{b}$ mesons). The muon momenta are corrected to account for possible geometry errors, and systematics are estimated on the correction and the resolution for propagation into individual analyses [127].

2.7.6 Missing transverse energy

There are some objects that are invisible to detectors such as ATLAS or CMS. The most obvious SM candidates are SM neutrinos; but many BSM theories (not least SUSY, and most other LHC-accessible DM models) predict particles with long – possibly stable – lifetimes that do not couple to either the electromagnetic or strong forces, and so will not be directly detected at any point in the detector.

However, the conservation of momentum gives us a tool to probe even these particles. We can define the two-vector missing transverse momentum[6] $\vec{E}_T^{\mathrm{miss}}$ as the negative vectorial sum of all the physics objects in the event (charged leptons, photons and jets). Because particles with a large p_z component (including the beam remnants) are likely to fall outside the detector acceptance range, we typically do not include the z component of in the definition of $\vec{E}_T^{\mathrm{miss}}$, as it is unlikely to represent the actual z component of any invisible particles. This, combined with the φ-symmetry of the ATLAS detector, explains why the scalar $E_T^{\mathrm{miss}} = \sqrt{p_x^2 + p_y^2}$ is often the variable of most interest to analyses.

In practice, to the "hard" term defined above, ATLAS also adds a "soft" term corresponding to tracks in the ID associated with the primary vertex which cannot be matched to any of the reconstructed physics objects that have already been incorporated in the sum. There will also be calorimeter signatures that cannot be matched to either physics objects *or* the unmatched tracks contributing to the soft term (most likely from neutral particles or pile-up): however, by only using tracks (whose production vertex can be much more easily evaluated), the soft-term is much more resistant to pile-up.

ATLAS provides a set of working points for E_T^{miss} reconstruction. For Run 2, these were `loose`, `tight`, `tighter` and `tenacious`. The tightness is based on the criteria used for jets entering the calculation: tighter selections exclude jets that were more likely to originate from pile-up. This means that the tighter WPs are less likely to have pile-up jets affecting their definition of E_T^{miss}, but are also more likely to not include "real" jets from the event.

[6] Even though this is really missing transverse momentum, the term "missing transverse energy" is often preferred for historical reasons, and the consequential notation, $\vec{E}_T^{\mathrm{miss}}$, is almost universal.

$Z \to ll$[7] is used to measure the E_T^{miss} resolution for each WP: this varies from 11 GeV to 35 GeV depending on the amount of pile-up and the WP; the tighter WPs have a superior resolution in high pile-up conditions. The calibration scale and the resolution are further used to estimate uncertainties that will be propagated through to analyses that use E_T^{miss}. The uncertainties on the hard term will depend on the uncertainties of all the objects used to construct it, and will likely dominate in most cases, as the hard term should normally be larger than the soft. A full breakdown of the calibration and uncertainties procedure can be found in Reference [128].

One common procedure in analyses containing a single leptonically decaying W-boson – such as the one in Chapter 8 – is to use the x and y components of the $\vec{E}_T^{\text{miss}}$ along with the four-momentum of a lepton to reconstruct the four-momentum of the W-boson: the energy and z components can be found by enforcing that the neutrino is massless and that W is on shell (and hence has mass m_W) and then solving the quadratic equation.

2.7.7 Overlap removal

The procedures for obtaining physics objects outlined above are carried out independently: i.e. electron identification receives no information from jet-clustering, and muon reconstruction receives no information from photon identification, and so-forth. This could be problematic, as these physics objects can sometimes produce similar signals: electrons and muons will deposit some energy in the calorimeter that may get misidentified as a jet; a small number of low energy jets may be sufficently narrow and sufficiently stopped by the EM calorimeter that they also get misidentified as electrons; and so-forth. This would confuse analyses and also break the definition of E_T^{miss} and other event-level variables.

To mitigate this, ATLAS analyses use a sequential "overlap removal" procedure. The exact definition varies from analysis to analysis, depending on the objects used and the working points those objects are defined at. A common example would be:

1. Remove electrons that share an ID track with a muon: it is impossible for an electron to reach the MS, but a rare muon may deposit enough energy in the EM calorimeter to appear electron-like.

2. Remove jets whose total four-momentum lies within some cone (e.g. $\Delta R < 0.2$) of an electron: the calorimeter deposits that formed these jets are likely to come from the electron and the photons it emits via bremsstrahlung.

3. Remove electrons within some wider cone (e.g. $\Delta R < 0.4$) of left-over jets. As the electrons aren't central, they cannot be the only contributor to the jet. They are likely either electrons produced in the decay of various hadrons in the jets, or possibly charged π-mesons produced in hadronisation that are misidentified as electrons.

4. Remove jets within some cone of muons. Because muons shower less energy in the calorimeters, additional criteria are often added here, such as only removing jets associated with three or fewer tracks (which would correspond to a muon that emits two bremsstrahlung photons).

5. Remove remaining muons within a cone of jets: these muon objects could be caused by "punch-through" of some jet constituents into the MS.

The specific overlap procedures for the analyses that will be covered in Part II of the thesis will be discussed when needed.

[7] i.e. dimuon or dielectron.

2.7.8 Unfolding

The reconstructed objects that we have defined in this section – while having already undergone some correction for the effects of the detector (such as energy scaling) – are still not comparable one-to-one either with similar results from other detectors (e.g. CMS), or more importantly, with the results that theorists obtain following the procedures that will be outlined in Chapter 3. This is the difference between "detector-level" data which comes out of reconstruction and "particle-level" data from Monte-Carlo event generators, a comparison which will be made more clear in Figure 3.1.

Different collider analyses bridge this gap in different ways. Most searches for new physics use detector-simulation to turn particle-level theory predictions into detector-level information. This thesis concentrates on searches for new physics; as such, we will dedicate a large portion of Chapter 4 to describing detector simulation.

However, most LHC measurements (for example of fiducial cross-sections) use a different strategy: they *unfold* the detector information back to truth level: this could be viewed as a continuation of the reconstruction process [85]. All the measurements used by the Contur package (see Section 4.3.1), which will be used in Chapters 5 and 7, rely on unfolded data. This makes it much easier for theorists to reuse results from LHC measurements.

Typically, unfolding is carried out to reproduce a "truth-level" observable binned in a histogram from sets of "detector-level" bins, though other approaches have been developed [129]. The mapping from the detector-level bins to the truth-level bins is calculated using extensive MC generation, and a combination of technical matrix-manipulation and possibly Bayesian-statistics techniques. Any technical explanation of these lies beyond the scope of this thesis, but one point worth noting is that these techniques all require low statistical errors in each bin, and hence the signal regions need to contain a significant number of events.

Finally, if this procedure works for measurements, why not apply it to searches too? Firstly, because we expect any BSM signatures at the LHC to be small, signal regions optimised for high-BSM signal to background ratios are likely to contain very few events[8], which would place unfolding on a statistically unsound footing. Even more prohibitively however, what "theory" should we be unfolding back to? Do we need to carry out this computationally very complex task at every single point in BSM parameter grids? And would unfolding back to a simplified model even be of any use to theorists wanting to reuse this result? Therefore, as pointed out at the start of this section, almost all searches are carried out at detector-level.

[8] Indeed, in section 5.5, we will see an Atlas SUSY signal region where no events were observed.

Chapter 3

From Lagrangians to the detector – Monte Carlo simulation

In crude terms, the goal of a detector such as CMS or Atlas is to take short bursts of electromagnetic activity within in a detector, and turn these into information about the Lagrangian $\mathcal{L}$, that describes all the physics in our universe (apart from possibly the small matter of gravity). A key pillar of the scientific method is prediction and therefore, we also need a way of travelling in the opposite direction: starting with a Lagrangian $\mathcal{L}$, and arriving at predictions for what electromagnetic signals are seen at colliders as a result.

The two legs of this scientific journey are illustrated in Figure 3.1, along with many of its stopping points and shortcuts. This chapter will describe the upper arc of this figure, walking from Lagrangian to Detector; the various tools and techniques required for the return leg will either be covered in the next chapter or, in the case of reconstruction, have been covered already.

General-purpose Monte Carlo (MC) event generators (MCEGs) are software packages that traverse the first three boxes of Figure 3.1. The most common general-purpose generators are Herwig [52, 53],

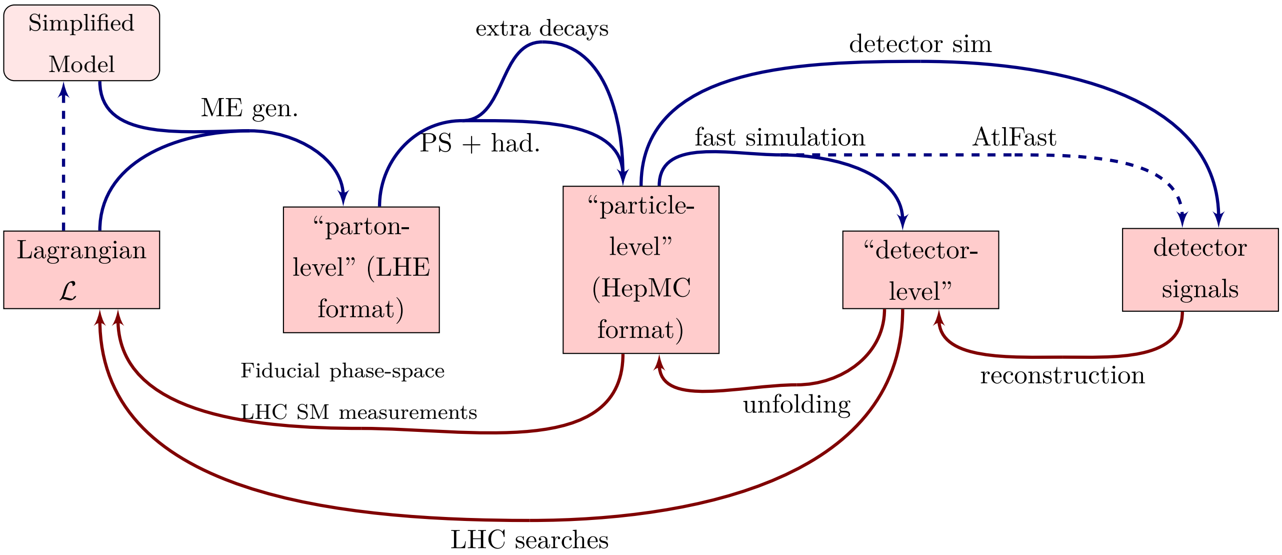


Figure 3.1: The steps and tools involved in moving from a Lagrangian, $\mathcal{L}$, to detector signals, and back again. "ME gen" represents matrix element generation, which we will cover in Section 3.1 and "PS + had." stands for parton showering and hadronisation, the topic of Section 3.2.

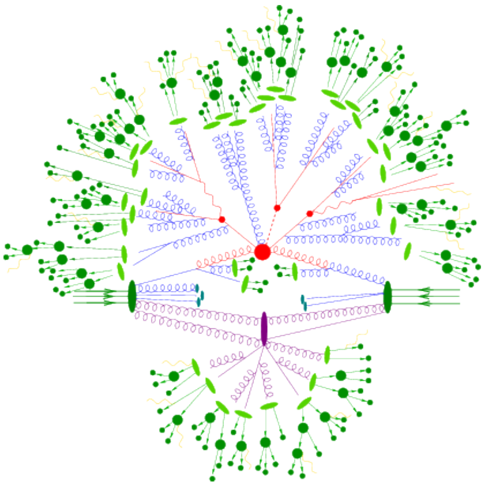


Figure 3.2: A visual representation of the work done by a general-purpose MCEG. The hard-process is depicted in red in the centre of the diagram – this will be covered in Section 3.1. The parton shower (the blue gluons) as well as hadronisation and decays (green, on the edge), will be described in Section 3.2. There is also some MPI – which will be covered in Section 3.3 – in purple at the bottom of the diagram.

Sherpa [130] and Pythia [131]; although because Pythia only implements most processes at 2 → 2 leading-order, it is primarily used for parton showers and hadronisation only. MadGraph5_MC@NLO [50] does not contain its own showering and hadronisation code, but has a seamless connection with Pythia. The procedures used within MCEGs are described in Sections 3.1 – 3.3, all of which are summarised in Figure 3.2.

3.1 Matrix elements and PDFs

The process of calculating what we expect to see at the LHC begins with the Feynman diagrams for the process: for example, the diagrams in Figure 3.3 show the LO 2→2 processes for QCD jet production.

For most processes, these matrix elements are calculated at fixed order in the strong coupling α_s, with other couplings (e.g. electromagnetic) included where necessary. While ideally we would calculate as many orders as possible, computational limits are restrictive. For example, most of the MC production used during Run 2 of the LHC experiments used results corrected to N^{3}LO for $pp \to h$ processes, and NNLO for $pp \to t\bar{t}$ processes [85]. At NLO and higher where there are divergences which need to be controlled by renormalisation, Q^2 (the momentum transfer of the event) is used as the renormalisation scale; but variations on this can be studied as systematic uncertainties. It is important to remember that an additional order of accuracy typically includes both new loops ("virtual" corrections) and new emissions ("real" corrections); cancellations from renormalisation often rely on the presence of both.

BSM theories are typically only calculated at tree-level, or occasionally NLO for more throroughly studied theories such as SUSY. Higher-order BSM event generation would be both impractical, given the size of BSM signal grids[1]; and less impactful, as very small differences in large SM backgrounds are typically much more significant for BSM searches than even moderately-sized changes to the signal modelling.

For some processes however, a fixed order-calculation may not be effective. This often occurs in processes that have had an infrared divergence removed by renormalisation, but where there are still large, finite effects near the divergence that make a perturbative expansion ineffective. One process where this happens is Drell-

[1] When studying BSM models, we may need to carry out event generation for many different variations of the BSM model – for example, different mass values for the SUSY LSP. Although points are not always distributed in a perfectly rectangular fashion, we refer to these as "signal grids": they will be of relevance to the research in Chapters 5, 6 and 8.

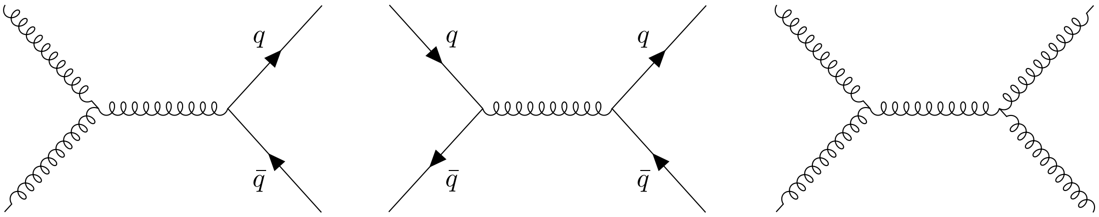


Figure 3.3: Three LO diagrams for QCD dijet production; QCD multijet production is the most common process at the LHC. Note that *none* of the incoming and outgoing legs can exist as bare particles.

Yan, the pair-production of leptons via a virtual photon. At NLO and higher, there is a divergence removed from the case an infinitely-soft or collinear gluon emission. In cases like these, there are large logarithms of scale ratios in the co-efficients that make a perturbative expansion inefficient: resummation techniques are used to extract these terms, and sum them [85, 132]. How many orders this summation covers is described by how far beyond the leading logarithm we have summed: "LL", "NLL", etc.

Unfortunately, because of confinement, it is impossible to accelerate individual quarks around the LHC and collide them. Fortunately however, asymptotic freedom means that when the proton is colliding with another proton at high energy, we can treat the quarks as approximately independent. For the first (but not the last) time in this section, we rely on the concept of factorisation – that we can treat physics at separated energy scales as approximately independent; and so we can write [85] the total *pp* cross-section as

$$\sigma = \sum_{i,j\in\{q,\bar{q},g\}} \int_0^1 \int_0^1 dx_1 dx_2\, f_i(x_1) f_j(x_2) \hat{\sigma}_{ij}\ , \tag{3.1}$$

where $\hat{\sigma}_{ij}$ is the cross-section evaluated for the specific combination of incoming quarks or gluons; and $f_{ij}(x)$ are the *parton distribution functions* (PDFs) for a given parton carrying a fraction x of the proton's momentum, which have an implicit dependence on Q^2. Although we commonly write the proton as *uud*, it is hopefully clear that simply writing down $f_u(x) = {}^2\!/_3$ and $f_d(x) = {}^1\!/_3$ would be very incomplete: we need gluons to hold the proton together, and there will also be radiative contributions. When we interact with a high energy proton, we could in fact be interacting not just with an up- or down-quark, but with any of the light quarks, light anti-quarks, or a gluon. The *b*-quark only contributes very slightly to the PDF; some schemes, known as four-flavour numbering schemes (4FNS) do not include it, but instead account for it possible presence using a gluon-splitting to $b\bar{b}$ in the matrix element. However five-flavour numbering schemes (5FNS), which include the *b* (albeit assumed massless) in the PDF, are also used[2].

A modern proton-PDF fit is shown in Figure 3.4 at two different scales. Modern PDFs have been built using global fits to experimental data primarily from deep inelastic scattering experiments like those at HERA [134, 135], where the structure of the proton could be probed in detail by colliding it with an electron or effective photon. However, lattice approaches to building PDFs from theory only, without the need for experimental fits, are being developed [136].

All event-generator software of note accesses PDFs via the LHAPDF package [138]. LHAPDF provides a repository for many different PDFs, which typically come in "sets" of several PDFs, providing a nominal along with variations which can be used to define systematic errors. LHAPDF also provides interpolation machinery so that the entire range of the PDFs can be used by the event generators.

Once the contributions of all the Feynman diagrams have been calculated, and the PDFs integrated and summed over, we have completed the high-energy "hard-process" step of event generation. At this point, a

[2] For an interesting comparison between the 4FNS and 5FNS schemes for single-top production, see Reference [133].

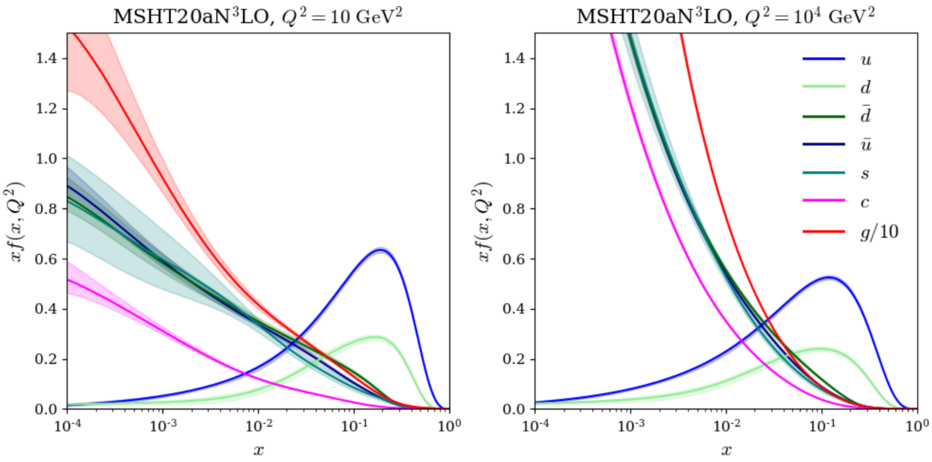


Figure 3.4: The proton PDF at Q^2=10 GeV^2 and at Q^2=10^4 GeV^2. Taken from Figure 34 of Reference [137], which describes the MSHT20aN^{3}LO PDF set depicted. As makes intuitive sense, at high x the intrinsic d and u contributions dominate, and the u contribution is approximately twice that of the d.

Monte-Carlo event generator can draw samples from the phase-space distributions in order to write simulated hard-events to an LHEF [139] file. This event record will typically contain $4 - 10$ particles, two of which will be the incoming protons. Particularly (though not exclusively) for BSM MC production of very heavy BSM particles with very short lifetimes, the LHEF format may also include the decays of the BSM particles. This is the typical output format for specialist event generators that do not carry out showering and hadronisation themselves, for example Protos [56].

If a matching or merging scheme is being applied (see next section), some additional information will need to be supplied to the showering and hadronisation software to inform it of which scheme and what cut-offs were used, to ensure there is no double-showering. One great advantage to users of general-purpose generators (or MadGraph5_MC@NLO) is that this process will be largely automated. However, even general-purpose event generators can be configured to still output LHEF files. This may be desirable for cross-checks – for example, supplying the same parton-level events to two different showering algorithms to compare them and potentially obtain a parton shower systematic uncertainty. For processes where the ME calculation is the most computationally expensive part, it can also be done to save disk space, as LHEF files can be an order of magnitude smaller than HepMC files that emerge after showering and hadronisation, and showering may be performed "on-the-fly" in memory at minimal performance cost – indeed, the significant decrease in disk I/O may even lead to performance improvements.

3.2 Parton showers and hadronisation

With PDFs, we took care of the problem of bare quarks or gluons on the incoming legs of our Feynman diagrams (e.g. Figure 3.3). However, there can still be bare quarks in the outgoing legs: these are removed by hadronisation, the process by which the bare quarks form colourless hadrons that we detect in the hadronic calorimeter. However, if we just magically converted the final state from Figures 3.3 directly into hadrons, we would obtain a very poor representation of the final state.

This is because many more partons will be emitted by the parton from the hard process before it hadronises (quarks will emit gluons and gluons will split into quark or gluon pairs). In principle, we could just keep adding more legs of ever-softer partons to the final state and give these effects the full QFT treatment; but this would become computationally intractable long before any reasonable results were obtained. Instead, we rely once again on factorisation, handling soft emissions in a physically motivated but more approximate process called the "parton shower" (PS).

3.2.1 Parton shower

The central mathematical object in parton showers is the the Sudakov Factor [85]:

$$\Delta_i(t_0, t) = \exp\left[-\sum_j \int_{t_0}^{t} \frac{dt'}{t'} \int_{z_{\min}}^{z_{\max}} dz P_{ij}(z) \right] , \tag{3.2}$$

where $P_{ij}(z)$ is the splitting function for partons i and j, obtained from the DGLAP equations[3]; this describes the probability that the parton j does *not* split between t_0 and t, factorised in the collinear emission limit.

There are several different algorithms for conducting the PS. These can be broadly split into "angular-ordered" algorithms, where t in the Sudakov factor is replaced by the angular separation of the emission θ, and all the emissions are calculated by emitting partons at successively narrower angles; and "p_T-ordered" algorithms, where progressively softer and softer partons are emitted by a pair of partons – hence this is sometimes known as a "dipole-shower" scheme. In both cases, for perturbative validity, a minimum-momentum cutoff needs to be imposed [85]. The 1 → 2 splittings in angular-ordered showers can never completely conserve Lorentz invariance whereas the 2 → 3 splittings in dipole-shower schemes can; while "real" parton-showers will not be completely on-shell, perfect momentum conservation at every stage does make matching and merging (described in the next section) more straightforward. However, there is a trade-off: angular schemes typically describe the color of the final states better.

An angular-ordered scheme is available in Herwig, while Pythia and Sherpa provide p_T-ordered dipole-shower algorithms. A comparison between the two schemes is included in most Atlas analyses as a theory uncertainty.

Before we complete our discussion of the parton shower, it is worth remarking that the relative "approximateness" of this process is a significant explanation of why QCD multi-jet processes like those begun in Figure 3.3 – which have a higher combined cross-section than any other process at the LHC – are so difficult to model accurately.

3.2.2 Matching and merging

One way to improve the description of showering is to include the hardest, most wide-angled showered partons in the matrix element, and then only use the parton shower for the remaining emissions. This effectively lowers the factorisation scale which separates the hard process from the PS. Such a procedure plays to the strengths of both the ME, which is very effective for a small number of hard partons; and the PS, which is least accurate for hard, wide-angled partons, but can treat many soft partons well [85]. When implementing such a procedure, it is very important to avoid double-counting emissions in both the ME and the PS. The algorithms most commonly used to treat extra emissions at tree-level LO are CKKW[4] matching, implemented in Sherpa; and MLM[5] matching, implemented in MadGraph5_MC@NLO [140]. The problem is even more complicated when we increase precision and emit partons using NLO diagrams due to the contributions of virtual corrections. Here, we avoid double-counting either with the Powheg method [141,142]; or using the MC@NLO prescription [143], as implemented (unsurprisingly) by MadGraph5_MC@NLO. Both methods

[3] The Dokshitzer-Gribov-Lipatov-Altarelli-Parisi functions $P_{ij}(z)$, where z is the fraction of the original particle's energy "taken" by the splitting particle, describe the evolution of the PDF for (light) quarks and gluons: for a fuller discussion and derivation, see Reference [6] or [26].

[4] Named for Catani-Krauss-Kuhn-Webber.

[5] Named for M.L. Mangano.

have their strengths and weaknesses: MC@NLO produces events with negative weights[6], which can be difficult to handle in practice, and Powheg does not work well for angular-ordered parton showers.

3.2.3 Hadronisation

The parton shower has taken from us a handful of partons to many partons: this is clearly a closer description of the QCD jets that are observed in LHC collisions. However, as discussed earlier, we still need to deal with the bare partons in *hadronisation*.

One model of hadronisation is the Lund string model [144]. The basic principle is that, $q\bar{q}$ pairs form, and due to differing momenta, begin to fly apart. Because the strong force gets stronger with separation, the "flux-tube" or "string" of QCD between them forms and contains more and more energy, until it becomes energetically favourable for a new $q\bar{q}$ to form in the "flux-tube" between them, breaking the string into two, as illustrated in Figure 3.5. A more rigorous treatment of this set-up, adding in gluon radiation from quarks and allowing for baryon formation by adding in diquarks, gives the full basis for string-based hadronisation models [85].

An alternative model is cluster hadronisation. This relies on preconfinement [145], the idea that after a PS, it is always possible to form colour-singlet clusters of all the partons that come out of the shower [85]; with a mass distribution that is calculable from the t_0 of the PS and the confinement scale Λ.

Both models are fundamentally phenomenological, and have a variety of non-physical parameters that have to be fitted to data in a process called tuning [146]. Pythia implements a string model, whereas Herwig and Sherpa use cluster models for hadronisation. The showers around the highest energy partons (whether originating in the hard process or the start of the PS) lead to cone-shaped "jets" of hadrons, which will be discussed further in Section 2.7.2.

After hadronisation, event generators will decay unstable particles (unless this is being handled by a specialist tool, as described in Section 3.2.4). Exactly what counts as "stable enough" to be considered a final-state particle varies from experiment to experiment and from analysis to analysis but, as a rough estimate, particles with a lifetime τ such that $c\tau > 10$mm will typically be considered stable. This cut-off will be much longer in searches for long-lived BSM particles (e.g. Reference [147]), that look for new particles that decay into the SM at some point significantly offset from the primary vertex, at any point from the ID to the very edge of the detector.

At this point, unless the event generator also handles some of the "other" effects discussed in Section 3.3, then the job of a general-purpose MC event generator is done. For LHC events, we are typically left with $100-1000$ final state particles, which will be a mixture of mesons, baryons, electrons, photons, and muons.

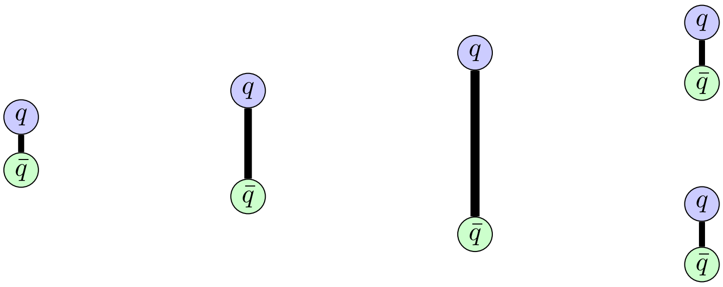


Figure 3.5: A very naïve depiction of the string model of hadronisation. $q\bar{q}$ pairs, with a string force flux-tube or "string" between them, move apart, increasing the energy in the flux-tube until it is energetically favourable for the flux tube to be broken by the creation of a new $q\bar{q}$ pair (and two new flux tubes).

[6] Technically, Powheg also can produce negatively weighted, but it produces many fewer such events and discards them.

These – along with all their precursor parent particles, all the way back to the incoming protons – will be written to a event-record format such as HepMC (or Root-based encodings thereof). Because so many particles need to be stored, HepMC files are typically very large: even when compressed, a few thousand events (far less than a second's data taking if they all occurred back to back) will take approximately 1 GB of disk space. As HepMC files effectively contain a more detailed, computational representation of Figure 3.2, it is tempting to think that the HepMC event record in some way represents "what-really-happened". This is *not* the case. The internal particles and vertices of the HepMC file are effectively just the scribbled "working out" done by the generator, and do not necessarily represent real, physical particles. For the same physics process, even if different MCEGs both agree almost exactly in all possible physical observables, the internal particles may be different, depending on parton shower or hadronisation algorithms, internal conventions, and so forth.

3.2.4 Additional decays

Although most unstable SM particles are decayed by the event generators themselves at the appropriate point, some heavy SM particles have complex decay kinematics that can be best handled by specialist tools. Therefore, dependning on the event-generator being used, for processes where this is likely to be significant an "afterburner" software may be applied to events. For example, EvtGen [148] handles the decays of B- and D-mesons; and Tauola [149, 150] deals with τ-leptons, which can decay hadronically.

3.3 The underlying event: the "other" effects during event generation

An LHC interaction point is a busy place. Typically, all effects other than the hard process, PS and hadronisation are collectively grouped together as the "underlying event".

The PDFs in equation 3.1 suggest that we just pluck an individual parton out of the proton to engage in our hard process and forget about the rest. In practice, the remains of both protons – described as *beam remnants* – cannot just be forgotten about. Any transverse momentum in the beam remnants will need to be offset by "intrinsic" transverse momentum in the colliding parton – an effect that should not be neglected.

If the transverse separation of the colliding protons (i.e. the impact factor) is small, it is possible that rather than just one parton from each proton interacting, we will have *multiple partonic interactions*, MPI. This is unlikely to contribute significant additional jets to the event [151]: instead it just adds to the relatively soft QCD "mess" of the event. A naïve first-order model may just overlay independent low-energy collisions, but in practice the partons produced from both collisions can interact and the processes will therefore not be independent [85].

Colour reconnection is the flow of colour between otherwise separated hadron clusters or separate partonic interactions from MPI. Most of our models do not account for this, but for some high-precision SM measurements it becomes an important effect [85].

3.4 Detector simulation

Once we have final-state particles, we need to know what effect they will have on the detector. We have just seen how difficult it is to model the interaction of two high energy protons: simulating the detector requires simulating (albeit at a much lower energy than pp collisions) the interaction of hundreds of final-state particles with an almost unimaginable number of atomic nuclei and electrons in a wide variety of complex materials and in a variety of magnetic field strengths and directions.

The first step in this process is having a very good understanding of which materials make up the detector, and an accurate map of where they lie. This means not just actual silicon sensing chips or liquid Argon calorimeter tiles, but also any wiring, cooling pipes or support struts, as particles can scatter off of these as well. In Atlas [103], the software description of the detector is based on the GeoModel library [152]. This also requires detailed measurements of the magnetic field strength and direction throughout the Atlas detector.

This, alongside the simulated event record, is then passed to Geant4 [153–155]. This software package simulates all the processes in the detector: the (curved) tracks of particles through the inner detector, the showering of photons and electrons in the EM calorimeters, the even more complex process of showering hadrons, and the motions of muons.

It is at this point in the simulation that we typically add *pile-up*. Because there are many protons in each bunch crossing, it is likely that we will measure some effects from other (lower energy) interactions during the same bunch crossing – this is referred to as "in-time" pile-up. Because 25ns is a very short time, it is also possible for effects from the preceding or succeeding bunch crossings to impact data-taking – referred to as "out-of-time" pile-up [85]. Pile-up is simulated separately, but added before digitisation, because it is easier to sum energy deposits than electrical responses.

Geant 4 provides a set of detector "hits", i.e. which particles struck which detector at what energy and with what momentum. The final step to complete the journey and have a simulated sample that is (ideally) indistinguishable from one obtained from data is digitisation. This is the process of turning hits into voltages, and then deciding if those voltages would trigger the detector to record or not.

3.5 Fast detector simulation and emulation

Full detector simulation, as described in the previous section, is a very computationally expensive process. Full simulation takes $\mathcal{O}$(minute) per event, and up to 90% of this time is in calorimeter modelling [156]. For the Atlas experiment, any reduction in the amount of full detector simulation that needs to be carried out would be a significant computational saving; and for smaller phenomenological groups seeking to carry out recasting (explained in detail in Chapter 3), not only is the full detector model not available, but it would also be completely impractical to run full simulation in an otherwise very simplified workflow.

For these reasons, faster, less-accurate methods of detector simulation can be very useful. As was illustrated in Figure 3.1, most fast-simulation software aims to convert particle-level truth information into an approximation of the reconstructed particles – hence skipping the digitisation and reconstruction steps – though some of the tools developed for internal use of the experiments also aim to reproduce the same "hits" that full simulation aims for.

3.5.1 The Atlfast series of tools

The most accurate of the fast simulation tools we will cover are the Atlfast series: AtlfastII (AF2) [157] and AtlFast3 (AF3) [158]. Because simulating the calorimeter is the slowest part of the Atlas detector simulation workflow, both Atlfast tools are faster simulations of the calorimeter only, with the normal Atlas simulation still being used for the other detector systems. AF2, used for some Atlas Run 2 samples in Chapter 8 of this thesis, gains its speed improvements – approximately one order-of-magnitude over full-simulation – by making various approximations, such as assuming all calorimeter cells are exact cuboids, and treating most hadrons exactly the same [156]. The next generation of Atlas fast-simulation software, AF3, has approximately the same CPU-cost as AF2, but replicates the Geant4 output much more closely. It uses a combination of a similar set of geometric simplifications to those used in AF2, with a neural net based approach using FastCaloGAN[7] [159]. The NN performs better that the geometric approximation approach at higher energies, and vice-versa; therfore AF3 interpolates between the two depending on the energy scale to obtain the best result. AF3 (unlike AF2) also includes a treatment of "muon punchthrough", the effect of the hadronic calorimeter failing to stop all of a hadronic shower, and some remnants of that shower "punching through" into the muon-system part of the detector.

Atlfast is useful for samples where the highest levels of accuracy are not required – for example large BSM signal grids, or systematic variations for samples where the systematic error is known to be large. However we should always be careful to compare apples with apples: an Atlfast nominal should also be generated to ensure a fair comparison with the systematic variation. Similarly, if a neural net is being trained to discriminate a signal sample from a background, it should not be trained to discriminate between an AtlFast signal and a full-simulation background, unless extensive studies have shown that all the NN inputs are well reproduced by the fast simulation.

The pace of developments in machine-learning since the publication of AF3 in 2021 – exemplified by the great successes of algorithms like GN2 (as discussed in Section 2.7.3) – suggest that there may be even greater performance (in speed or accuracy) improvements to come in the future. It would be fantastic for the (less-well resourced) reinterpretation community if some of these developments could bleed into the publicly available fast-simulation packages (some of which are described below) that target reco-level final states instead of hits.

3.5.2 Delphes

Perhaps the most common detector-simulation tool used by the reinterpretation community during the LHC era has been Delphes [160]. Delphes has an internal model of a "detector", consisting of a tracker, EM and hadronic calorimeters, and muon systems. The exact geometries of these, as well as various stopping efficiencies and magnetic field strengths are encoded in a "detector-card" input that defines the detector being simulated. Although they are not released by the experiments themselves, public cards corresponding to approximations of Atlas and CMS are available. However, it has also become common practice to adjust these for individual analyses either to account for poor modelling or to correct for tighter or looser than default identification efficiencies (see the discussion of MadAnalysis5 in Section 4.3.3).

There is no direct physics modelling of the interaction between final-state particles (hadrons, leptons, etc.) and different types of material in the detector, as there would be in Geant. Rather, a series of phenomenological parameters estimate the fractions of energy deposited in a generic calorimeter, or the tracks left behind in a generic inner tracker. The geometry modelling is much coarser than that described in Section 3.4: while the approximate positions and sizes of the individual detectors (tracker, calorimeter, etc.) are included, there is no direct accounting for smaller effects, such as scattering from support beams or bundles of electrical wiring. Nevertheless, the overall impact of these features will still be largely absorbed into the efficiencies and other phenomenological parameters that define the Delphes detector model.

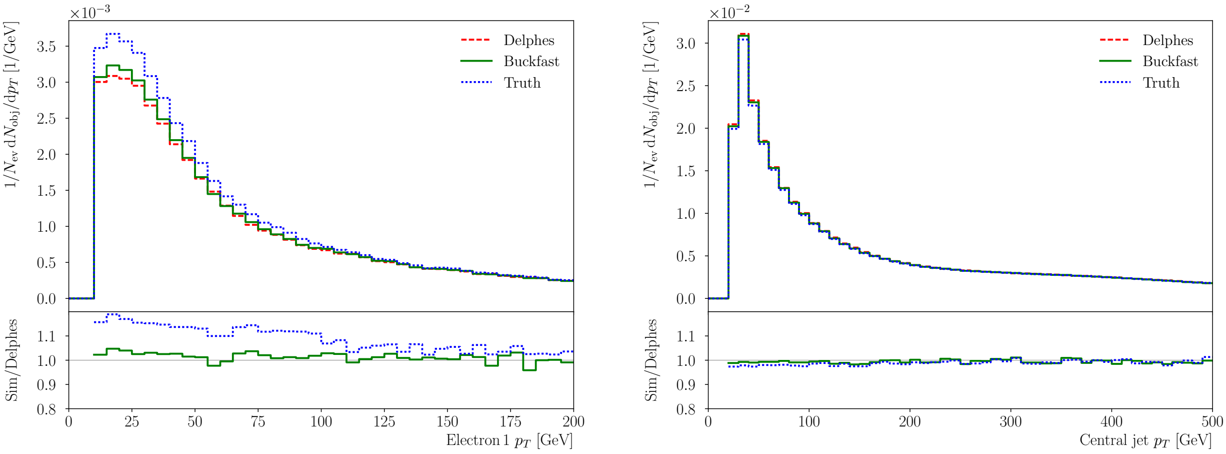
3.5.3 Four-vector smearing

The crudest – but also fastest – method in this section is detector simulation based on four-vector smearing. Here, the four-momenta of the relevant physics objects – jets, leptons, and so forth – are randomly smeared

[7] See Section 4.2 for an introduction to neural nets.

according to distributions based on the known, published detector resolution and identification efficiencies. These smearings are typically Gaussian, although for some frameworks – such as that in Rivet (to be introduced in Chapter 4) – it is possible to include customised distributions, such as the Crystal-Ball function [161]. Tagging is carried out using the efficiencies of the tagger, either uniformly across all objects, or according to an efficiency map – for example, b-tagging efficiency as a function of jet p_{T} – if one is available. Mistag rates – such as c-jets wrongly tagged as b-jets – are also considered when available.

Because no attempt is made to reconstruct the detector (other than a handful of pseudorapidity-based cuts), this is technically *emulation*, rather than simulation – though a similar comment could be made about Delphes. This technique was popularised by the BuckFast detector-emulator in ColliderBit (which will be introduced more fully in the next chapter); although it has since been adopted by several other tools as well. Some plots from an early version of BuckFast are shown in Figure 3.6: performance relative to Delphes is very good, especially given that Delphes is approximately an order of magnitude slower and due to its dependence on Root cannot be easily parallelised.



(a)

(b)

Figure 3.6: The performance of BuckFast detector-emulation, compared to a more detailed-yet-slower representation (Delphes), and the unsmeared, particle-level "truth" for the lead electron (a) and leading-jet (b) p_{T} distributions in pair-produced EWino events. The performance is very good. Reproduced from Reference [162], Figure 2.

Chapter 4

Tools for analysis and reinterpretation

The preceding chapters have covered how we collect data resulting from particle collisions; and how we can use a range of theoretical tools in order get from a Lagrangian to equivalent predictions. But how do we go from all this data to practical information, such as measuring a cross-section estimate or setting a statistical limit on a parameter of a BSM theory? And can theorists reuse published experimental results to provide additional information about other new theories? This is the realm of data analysis and reinterpretation.

4.1 Statistics

4.1.1 Binned likelihoods

Though the popularity of so-called “unbinned” fits has proliferated in recent years (see examples at both Atlas [163], CMS [164] and LHCb [165]), the overwhelming majority of traditional particle-physics analyses count events in bins, with bin edges dictated by some combination of event observables. These observables may be simple kinematics – e.g. the transverse momentum of the leading jet – or more complex features of the event, all the way up to highly non-linear artificial neural nets that may take over a hundred lower-level observables as inputs.

Typically, we are looking for some signal events, s, amongst the background events, b. Then, following Reference [166], we can write the expected number of events, n, in the ith bin as

$$\mathbb{E}\left(n_i\right) = b_i + \mu s_i \; , \tag{4.1}$$

where μ is a parameter that measures the “strength” of the signal, with $\mu = 1$ corresponding to the “nominal signal” – typically BSM physics exactly as the BSM theory predicts – and $\mu = 0$ corresponding to no signal at all, which typically corresponds to “just” the SM.

The values of s_i and b_i will be extracted from Monte-Carlo simulation as

$$b_i = b_{\text{total}} \cdot \int\limits_{\text{bin } i} f_b(x, \theta_b)\, dx \tag{4.2a}$$

and

$$s_i = s_{\text{total}} \cdot \int\limits_{\text{bin } i} f_s(x, \theta_s)\, dx \; , \tag{4.2b}$$

where f is a probability density function that we obtain from Monte-Carlo, x is the variable being measured, and θ is the set of nuisance parameters to profile over. These nuisance parameters will account for systematic uncertainties from theory (such as PDF variations), and experimental uncertainties (such as calorimeter resolution); as well as other terms that may be specific to the analysis in question.

Then, as the counts in each bin will be Poisson-distributed, the likelihood function for all N independent bins with observed counts $n = (n_1, n_2, ...)$ can be written as

$$L(n, \mu, \theta) = \prod_{i=1}^{N} \frac{(b_i + \mu s_i)^{n_i} \cdot e^{-(b_i + \mu s_i)}}{n_i!} \cdot \prod_{j=1}^{M} G(\theta_j) \ , \tag{4.3}$$

where we have kept the M nuisance parameters θ implicit in the first product. The nuisance parameter penalty terms $G(\theta_j)$ are typically Gaussian, and will be discussed in more detail later.

The likelihood is important, because many test statistics depend on the "profile likelihood ratio", $\lambda(\mu)$:

$$\lambda\left(n, \mu\right) = \frac{L\left(n, \mu, \hat{\hat{\theta}}(\mu)\right)}{L\left(n, \hat{\mu}, \hat{\theta}\right)} \tag{4.4}$$

where the single caret implies the values that maximise the likelihood (the "maximum-likelihood" parameters), and the double caret indicates the values of the nuisance parameters that maximise the likelihood, given a specific value of μ. As the denominator is the maximum possible value of the likelihood, and equation 4.3 is positive semi-definite, $0 \leq \lambda(\mu) \leq 1$. In simple terms, higher values of λ correspond to values of μ that give likelihoods that are close to the maximum and hence fit the data well, and conversely values close to zero show that the particular choice of μ is not a good representation of the data.

It is also worth mentioning two special cases. The first is $\lambda(n, \mu = 0)$, which is the "background-only" profile-likelihood ratio. This gives an understanding of how well the fit performs in the absence of any signal and may help diagnose mismodelling of backgrounds if present. The second is $\lambda(n = n_{\text{Asimov}}, \mu)$, where $n_{i,\text{Asimov}} = b_i + s_i$ is the "expected" number of events. Propagating this profile-likelihood into test statistics gives an idea of how well a study is likely to perform.

4.1.2 Test statistics and *p*-values

When it comes to placing precise limits on the model, while $\lambda(\mu)$ is used directly in some places, but we often prefer test statistics that are simple transformations of the profile-likelihood ratio. Again following Reference [166], and making the dependence on n implicit for brevity, three of the most common are

$$t_\mu = -2 \ln \lambda(\mu) \ , \tag{4.5}$$

$$q_0 = \begin{cases} -2 \ln \lambda(\mu) & \hat{\mu} \geq 0 \\ 0 & \hat{\mu} < 0 \ , \end{cases} \tag{4.6}$$

and

$$q_\mu = \begin{cases} -2 \ln \lambda(\mu) & \hat{\mu} \leq \mu \\ 0 & \hat{\mu} > \mu \ . \end{cases} \tag{4.7}$$

The first of these, t_μ, is used for the simple test of "how incompatible is the data with the given value of μ?"

The second, q_0, is a test of "discovery": how strongly does the data disagree with null hypothesis $\mu = 0$: as q_0 increases, the less compatible the data with an absence of signal. The change in behaviour for negative $\hat{\mu}$ means that we cannot "do better" than observing exactly the background, so the test statistic is set to zero for negative signal strengths.

Finally q_μ is a test statistic that allows us to place upper limits on values of μ. In this scenario, $\hat{\mu} > \mu$ should not place any constraint on the upper limit, so the test statistic is set to zero.

Note that the use of q in both Equations 4.6 and 4.7 is purely notational, and does not imply that q_0 is a limiting case of q_μ for $\mu \to 0$.

For both t_μ and q_μ, there exist variations labelled $\tilde{t}_\mu$ and $\tilde{q}_\mu$ designed for the case where physics enforces $\mu \geq 0$. In this case, we define

$$\tilde{\lambda}(\mu) = \begin{cases} \frac{L\left(\mu,\, \hat{\hat{\theta}}(\mu)\right)}{L\left(\hat{\mu},\, \hat{\theta}\right)} & \hat{\mu} \geq 0 \\ \frac{L\left(\mu,\, \hat{\hat{\theta}}(\mu)\right)}{L\left(\mu = 0,\, \hat{\hat{\theta}}(\mu = 0)\right)} & \hat{\mu} < 0 \end{cases} \tag{4.8}$$

and insert it into the relevant formulae for t_μ or q_μ:

$$\tilde{t}_\mu = -2 \ln \tilde{\lambda}(\mu) \tag{4.9}$$

and

$$\tilde{q}_\mu = \begin{cases} -2 \ln \tilde{\lambda}(\mu) & \hat{\mu} \leq \mu \\ 0 & \hat{\mu} > \mu \,, \end{cases} \tag{4.10}$$

noting that by using $\tilde{\lambda}$ in the definition we implictly have an additional case to consider.

For all these variables, we can convert from the test-statistic to the p-value via the integral

$$p = \int\limits_{q_{\mathrm{obs}}}^{\infty} f(q|\mu)\, dq \,, \tag{4.11}$$

where we substitute q for any one of t_μ, $\tilde{t}_\mu$, q_0, q_μ or $\tilde{q}_\mu$. f is the probability density function of the test statistic at a given value of μ. In physics, we often prefer to state the significance, Z, defined as

$$Z = \Phi^{-1}(1 - p) \tag{4.12}$$

where Φ is the cumulative distribution function of a unit Gaussian, and Φ^{-1} its inverse. Because Z can be equated to the number of standard deviations above the mean, it is often quoted as the "σ" value, with a global significance of 5σ being considered the benchmark for "discovery" during the LHC era.

Unfortunately, the integral of the PDF in equation 4.11 is not one we can look up in a table. Historically, the problem was solved with significant computational effort by running many MC "toy" experiments.

4.1.3 The Wald approximation and the Asimov dataset

Building on earlier work from Wilks for the special case $\mu = \mu'$ [167], the Wald approximation [168], in the case of one parameter interest, states that

$$t_\mu = -2 \ln \lambda(\mu) = \frac{(\mu - \hat{\mu})^2}{\sigma^2} + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right) \,, \tag{4.13}$$

where $\hat{\mu}$ is a Gaussian random variable with mean μ' and variance σ^2. σ^2 comes from the covariance matrix of all the nuisance parameters $V_{ij} = \mathrm{cov}(\vartheta_i \vartheta_j)$, where for convenience we absorb the POI as $\vartheta = (\mu, \theta)$. In the limit of a very large dataset – i.e. the *asymptotic limit* – this can be approximated as:

$$V_{ij}^{-1} = -\mathbb{E}\left[\frac{\partial^2 \ln L}{\partial \vartheta_i \partial \vartheta_j}\right] \,. \tag{4.14}$$

Using $\Lambda = \frac{(\mu - \mu')^2}{\sigma^2}$ as the *non-centrality parameter*, we can then write down a distribution for t_μ:

$$f(t_\mu|\Lambda) = \frac{1}{2\pi} \cdot \frac{1}{2\sqrt{t_\mu}} \cdot \left[\exp\left(-\frac{\left(\sqrt{t_\mu} + \sqrt{\Lambda}\right)^2}{2}\right) + \exp\left(-\frac{\left(\sqrt{t_\mu} - \sqrt{\Lambda}\right)^2}{2}\right)\right] \tag{4.15}$$

and similarly for any of the other test statistics from Section 4.1.2.

The Asimov dataset [166] – so named because of a science-fiction reference rather any direct involvement of the writer and biochemist – is an artificial dataset defined such that estimators always return the true values. In practice, this means that every measured quantity is equal to its expectation value; so, in the case of a single-binned experiment which counted how many heads were obtained after tossing a (fair) coin 99 times, the Asimov dataset would be $\{49.5\}$.

Using the Asimov dataset and equation 4.3, we can define an "Asimov Likelihood" λ_{A}:

$$\lambda_{\mathrm{A}}(\mu) = \frac{L_{\mathrm{A}}\left(\mu, \hat{\hat{\theta}}\right)}{L_{\mathrm{A}}\left(\hat{\mu}, \hat{\theta}\right)} = \frac{L_{\mathrm{A}}\left(\mu, \hat{\hat{\theta}}\right)}{L_{\mathrm{A}}\left(\mu', \hat{\theta}\right)} \ , \tag{4.16}$$

noting that a possibly fractional event count is not problematic because the factorial terms cancel. As pointed out by Reference [166], rather than obtaining an estimate of the variance from the covariance matrix analogously to equation 4.14, we can instead create an estimator σ_{A}^2 for the variance:

$$\sigma_{\mathrm{A}}^2 = \frac{(\mu - \mu')^2}{-2\ln\lambda_{\mathrm{A}}(\mu)} \ . \tag{4.17}$$

4.1.4 Approximate distributions

Combining the results from the previous section, we can (finally) write down easy-to-handle forms of the PDF in equation 4.11 for our test-statistics. Following Reference [166] one more time, and using $\delta(x)$ to represent the Dirac delta-function, we arrive at the following approximate PDFs:

$$f(t_\mu|\mu') = \frac{1}{2\sqrt{t_\mu}\sqrt{2\pi}} \cdot \left[\exp\left(-\frac{\left(\sqrt{t_\mu} + \frac{\mu-\mu'}{\sigma}\right)^2}{2}\right) + \exp\left(-\frac{\left(\sqrt{t_\mu} - \frac{\mu-\mu'}{\sigma}\right)^2}{2}\right)\right] \tag{4.18a}$$

$$f(q_0|\mu') = \left(1 - \Phi\left(\frac{\mu'}{\sigma}\right)\right)\delta(q_0) + \frac{1}{2\sqrt{2\pi}\sqrt{q_0}} \exp\left[-\frac{1}{2}\left(\sqrt{q_0} - \frac{\mu'}{\sigma}\right)^2\right] \tag{4.18b}$$

$$f(q_\mu|\mu') = \Phi\left(\frac{\mu' - \mu}{\sigma}\right)\delta(q_0) + \frac{1}{2\sqrt{2\pi}\sqrt{q_\mu}} \exp\left[-\frac{1}{2}\left(\sqrt{q_\mu} - \frac{\mu - \mu'}{\sigma}\right)^2\right] \tag{4.18c}$$

$$f(\tilde{t}_\mu|\mu) = \frac{1}{2\sqrt{\tilde{t}_\mu}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\sqrt{\tilde{t}_\mu} + \frac{\mu - \mu'}{\sigma}\right)^2\right] + \begin{cases} \frac{1}{2\sqrt{\tilde{t}_\mu}\sqrt{2\pi}} \exp\left[\frac{-1}{2}\left(\sqrt{\tilde{t}_\mu} - \frac{\mu-\mu'}{\sigma}\right)^2\right] \\ \qquad \tilde{t}_\mu \leq \mu^2/\sigma^2 \\ \frac{1}{2\sqrt{\tilde{t}_\mu}\sqrt{2\pi}} \exp\left[\frac{-1}{2}\frac{\left(\tilde{t}_\mu - (\mu^2 - 2\mu\mu')/\sigma^2\right)}{(2\mu/\sigma)^2}\right] \\ \qquad \tilde{t}_\mu > \mu^2/\sigma^2 \end{cases} \tag{4.18d}$$

$$f(\tilde{q}_\mu|\mu') = \Phi\left(\frac{\mu' - \mu}{\sigma}\right)\delta(\tilde{q}_\mu) + \begin{cases} \frac{1}{2\sqrt{2\pi}\sqrt{\tilde{q}_\mu}} \exp\left[-\frac{1}{2}\left(\sqrt{\tilde{q}_\mu} - \frac{\mu-\mu'}{\sigma}\right)^2\right] \\ \qquad 0 < \tilde{q}_\mu \leq \mu^2/\sigma^2 \\ \frac{1}{\sqrt{2\pi}(2\mu/\sigma)} \exp\left[-\frac{1}{2}\frac{\left(\tilde{q}_\mu - (\mu^2 - 2\mu\mu')/\sigma^2\right)^2}{(2\mu/\sigma)^2}\right] \\ \qquad \tilde{q}_\mu > \mu^2/\sigma^2 \ , \end{cases} \tag{4.18e}$$

which crucially come with the following analytic expressions for the cumulative density:

$$F(t_\mu|\mu') = \Phi\left(\sqrt{t_\mu} + \frac{\mu - \mu'}{\sigma}\right) + \Phi\left(\sqrt{t_\mu} + \frac{\mu - \mu'}{\sigma}\right) - 1 \tag{4.19a}$$

$$F(q_0|\mu') = \Phi\left(\sqrt{q_0} - \frac{\mu'}{\sigma}\right) \tag{4.19b}$$

$$F(q_\mu|\mu') = \Phi\left(\sqrt{q_\mu} - \frac{\mu - \mu'}{\sigma}\right) \tag{4.19c}$$

$$F(\tilde{t}_\mu|\mu') = \Phi\left(\sqrt{\tilde{t}_\mu} + \frac{\mu - \mu'}{\sigma}\right) + \begin{cases} \Phi\left(\sqrt{\tilde{t}_\mu} - \frac{\mu-\mu'}{\sigma}\right) - 1 & \tilde{t}_\mu \leq \mu^2/\sigma^2 \\ \Phi\left(\frac{\tilde{t}_\mu - (\mu^2 - 2\mu\mu')/\sigma^2}{2\mu/\sigma}\right) - 1 & \tilde{t}_\mu \leq \mu^2/\sigma^2 \end{cases} \tag{4.19d}$$

$$F(\tilde{q}_\mu|\mu') = \begin{cases} \Phi\left(\sqrt{\tilde{q}_\mu} - \frac{\mu-\mu'}{\sigma}\right) & 0 < \tilde{q}_\mu \leq \mu^2/\sigma^2 \\ \Phi\left(\frac{\tilde{q}_\mu - (\mu^2 - 2\mu\mu')/\sigma^2}{2\mu/\sigma}\right) & \tilde{q}_\mu > \mu^2/\sigma^2 \ , \end{cases} \tag{4.19e}$$

where we note there are significant simplifications for the case of $\mu = \mu'$ (i.e. the data matches the null hypothesis): for example, $t_\mu = -2\ln\lambda(\mu)$ approaches a chi-squared distribution.

4.1.5 The $\mathrm{CL_s}$ method

Unfortunately, the process of background modelling in particle physics is rarely perfect, and background models will suffer from up and down perturbations. Unfortunately, a particularly large up-variation may sufficiently exceed the observed data that even a signal strength of zero would be excluded by a test statistic based solely on fitting the combined signal and background model. In turn, this could lead to overly aggressive exclusion contours, with regions excluded not necessarily because the signal does not agree but because the background model fluctuates inconveniently.

The $\mathrm{CL_s}$ method [169, 170], is designed to mitigate this issue, by taking into account the goodness of fit of the background-only model, and downweighting exclusion where the background model also fits the data poorly. Most commonly the $\mathrm{CL_s}$ statistic is written as

$$\mathrm{CL_s} = \frac{\mathrm{CL_{s+b}}}{\mathrm{CL_b}} = \frac{p_{\mathrm{s+b}}}{1 - p_{\mathrm{b}}} \ , \tag{4.20}$$

though alternative formulations have been used, for example by the ALEPH collaboration at LEP [170, 171].

Exactly which test statistic to use to obtain the p-value is not consistently agreed upon, and nor is the exact definition of p_b. For example, the `pyhf` likelihoods package [172] by default uses the q_μ statistic from equation 4.7 (though $\tilde{q}_\mu$ is also available), and defines p_b as the integral from negative infinity to the observed value of q. This definition can be fully written down as:

$$\mathrm{CL_s^{\texttt{pyhf}}} = \frac{p_{\mathrm{s+b}}^{\texttt{pyhf}}}{1 - p_{\mathrm{b}}^{\texttt{pyhf}}} = \frac{\int_{q_{\mathrm{obs}}}^{\infty} f(q_\mu|\mu) dq_\mu}{1 - \int_{-\infty}^{q_{\mathrm{obs}}} f(q_\mu|\mu = 0) dq_\mu} = \frac{\int_{q_{\mathrm{obs}}}^{\infty} f(q_\mu|\mu) dq_\mu}{\int_{q_{\mathrm{obs}}}^{\infty} f(q_\mu|\mu = 0) dq_\mu} \ . \tag{4.21}$$

Conversely, the Contur reinterpretation tool – introduced in Section 4.3.1 – uses a χ^2 approximation to the t_μ test statistic from equation 4.5 and an inverted definition of p_b, which can be approximated as:

$$\mathrm{CL_s^{\textsc{Contur}}} = \frac{p_{\mathrm{s+b}}^{\textsc{Contur}}}{1 - p_{\mathrm{b}}^{\textsc{Contur}}} = \frac{\int_{t_{\mathrm{obs}}}^{\infty} f(t_\mu|\mu) dt_\mu}{1 - \int_{t_{\mathrm{obs}}}^{\infty} f(t_\mu|\mu = 0) dt_\mu} \ , \tag{4.22}$$

which means that, while in the `pyhf` convention a $\mathrm{CL_s}$ score close to zero implies exclusion, in the Contur convention a score close to one implies exclusion!

Reference [170], one of the two standard citations for this method, inverts *both* the p_b and p_{s+b} definitions relative to the `pyhf` conventions:

$$\mathrm{CL_s^{Read}} = \frac{p_{\mathrm{s+b}}^{\mathrm{Read}}}{p_{\mathrm{b}}^{\mathrm{Read}}} = \frac{\int_{-\infty}^{Q_{\mathrm{obs}}} f(Q|\mathrm{s+b}) dQ}{\int_{-\infty}^{Q_{\mathrm{obs}}} f(Q|\mathrm{b}) dQ} = \frac{1 - \int_{Q_{\mathrm{obs}}}^{\infty} f(Q|\mathrm{s+b}) dQ}{1 - \int_{Q_{\mathrm{obs}}}^{\infty} f(Q|\mathrm{b}) dQ} \ , \tag{4.23}$$

where Q is a different test statistic to those in equations 4.5 – 4.10; the exact details of which can be found in Reference [170] – though common at LEP and the Tevatron, it is no longer commonly used in the LHC era. Therefore, with the two inversions cancelling, a score close to zero implies exclusion and the oft-cited 95% exclusion contour lies at $\mathrm{CL_s} = 0.05$, as it does for `pyhf`. This definition is included for historical reasons, but neither $\mathrm{CL}_s^{\mathrm{Read}}$ nor Q are relevant to the rest of this thesis.

Throughout this thesis, because of the wide array of tools used, we will not be able to use only one convention: indeed for the test-statistic difference in particular, the best definition may depend on context. When dealing with results from CONTUR or `pyhf` in the reinterpretation context of Chapters 6 and 5, I will use each tools' own definition, though for ease of comparison the `pyhf` value will be subtracted from one, so that a score close to one implies exclusion.

The ATLAS analysis results in Chapter 8 will use the definition from the `trex-fitter` package, which is consistent with that used by `pyhf`, as both packages trace either dependence or a need for compatability with the ROOT `HistFactory` format [173].

4.1.6 Profile likelihood fits in experimental practice

The formulae of Sections 4.1.1 - 4.1.5 may seem very abstract. To try to bridge the gap between the statistical theory and the experiment, we will briefly walk through how they are implemented in practice, for a generic LHC search for new physics.

A simple LHC search will use a series of cuts on observables to define a signal region (SR), where the ratio of signal to background events is expected to be high; as well as a control region (CR), where the number of background events is small compared to the background population. In practice, searches will have multiple SRs and CRs; and they will often split each region into many bins by fitting to a histogram of some observable within the region. This can be particularly problematic if searches define overlapping signal regions – such as the ATLAS SUSY search reinterpreted in Section 5.5 – when great care must be taken to understand the correlations between the regions. The populations of these regions will follow equation 4.1, noting again that s_i and b_i will vary with the nuisance parameters θ originating from the systematic variations.

There are a variety of experimental, theoretical, and possibly *ad hoc* systematic errors that need to be profiled over. For example, there will likely be experimental systematics to describe the resolution of the detector in various variables; or theoretical systematics to cover uncertainties in the Monte-Carlo modelling. For practical and computational ease, systematics are assumed to have a nominal central value and (not necessarily symmetric) up and down variations corresponding to $\{+1\sigma, -1\sigma\}$ templates. During the fit, each nuisance parameter receives a Gaussian penalty term – of the form $\left(1/\sqrt{2\pi}\right)\exp\left(-\theta^2/2\right)$, multiplied into the likelihood – for deviation from its pre-fit value.

One particularly problematic case for this set-up is the so-called *two-point* systematic. This is used where our understanding of some uncertainty comes from having just two different versions of the same data – for example, MC simulation for one particular process carried out by two different event generators. In this case, we typically choose one sample as the nominal, the other as the “up” alternative, and complete the nuisance parameter distribution with an unphysical down variation obtained by symmetry. Because this is a rather *ad-hoc* approach, the up and down templates for two-point systematics are sometimes taken as $\{+2\sigma, -2\sigma\}$ in order to err on the side of conservatism.

Once the full likelihood – including the nuisance parameters – is constructed, a computational minimiser is used to profile over the nuisances and find the likelihoods that go into the profile likelihood ratio (equation 4.4). This is not a computationally trivial task – for example, the fit for the analysis in Chapter 8 contained over 250 nuisance parameters – though it is normally simplified by the absence of many strong correlations among the nuisance parameters.

The deviation of a nuisance parameter from its pre-fit nominal value – often referred to as a *pull* – is an interesting output that can often provide a lot of information about a fit. If any of the background related nuisance parameters are pulled significantly (e.g. over 2σ), this may suggest the background modelling is poor and needs to be re-examined. Similarly, if a 2-point systematic is strongly pulled towards the unphysical down variation, then something is likely awry with the modelling.

Finally, we can obtain p-values, significances and exclusions by integrating over the relevant distributions of test statistics as implied by equation 4.11 using the approximations in Equations 4.18 and 4.19. Note that this implies sampling over many μ values, necessitating multiple fits, and can therefore be a computationally expensive process; even before we repeat the procedure across multiple parameter points.

4.1.7 Preserving and publishing statistical models

An experimental analysis will typically publish mass or cross-section limits, or else a discovery significance. However, this is far from the only information that the analysis obtained, and often theorists and other users will require more information to make use of the result. Indeed the importance of sharing the full-likelihood function from CERN experiments was recognised as early as 2000 [174].

The simplest way to do this is to collapse all the information about a single signal region into an observed count, an expected (pre-fit) background contribution and a single error estimate on that background. This information is (nearly) always published within the main text of analyses from Atlas and CMS. This allows a crude rerunning of the likelihood calculation (which we will show later, in equation 4.41). However, it does not encapsulate important features such as the correlations between different regions, and it does not allow for any asymmetry in the statistical errors.

A step beyond this is the simplified likelihood model [175]. This uses as many background nuisance parameters as there are bins, to allow for correlations between bins to be partially understood. It does not allow for non-Gaussian nuisances, and naturally encodes less information than if all the nuisance parameters were kept. Simplified likelihood information has been published for several CMS analyses, for example Reference [176], which was used in the Gambit study that will be described in Section 7.3.

The `pyhf` format – with its Python package already discussed above – is a JSON format designed to encode all the information that can be included in likelihoods built in the Root-based HistFactory [173] format, without any dependence on Root or RooStats. Depending on the complexity of the analysis, this could effectively contain the likelihood of the entire workspace. Several Atlas SUSY analyses have published their likelihoods in `pyhf` format, including Reference [177], which we will use in Section 5.5.

A potentially even fuller description than that which `pyhf` might be able to provide comes from the full RooStats [178] workspace objects used by the experiments to store likelihoods internally. These are the full mathematical descriptions, and access to these would allow those using the results almost as much freedom as the original experimental teams. Unfortunately, these are entirely Root dependent and would be difficult for external users to interact with even if they were made public. The ongoing HS3 project [179] aims to create an independent framework for storing these likelihoods, allowing an even richer preservation format than `pyhf`, albeit still fully compatible with `pyhf` when `pyhf` can describe the likelihood.

The CMS experiment has also recently released Combine [180], which provides the full likelihood information for certain CMS analyses, using a docker image to allow access to the necessary CMS software. As Combine matures and HS3 nears release, it would be helpful if some degree of compatibility between the two could be co-ordinated.

4.2 Neural networks

As we will discuss further in Chapter 5, machine-learning (ML) techniques have a venerable history within particle physics. ML techniques can be broadly split into “supervised” and “unsupervised” problems. The supervised category aims to solve problems for which there is a known solution for each example in the training dataset, such as regressing to values of a function, or classifying between known categories. Conversely, unsupervised problems are not trying to learn a previously known answer, but rather carry out tasks such as clustering data into previously undefined classes.

The whole field of ML is far too broad to cover in a single thesis chapter. Rather, we will restrict ourselves to supervised learning, focussing on neural networks (NNs), and in particular where they are most pertinent to particle physics. As their name suggests, NNs were originally inspired by attempts to understand biological neural structures, but advances in both computing power – particularly the increasing availability of graphics processing units, GPUs – and statistical methodology have (largely) moved the field far beyond those origins.

Typically, we start with a large dataset of input features v_i and expected outputs o_j. The goal of the NN is to transform the inputs into as close as possible a representation of the outputs, based on a distance measure known as the *loss function*[1].

4.2.1 The simplest case: deep neural networks

One of the simplest forms of NN, still in widespread usage in physics, is the Deep Neural Network (DNN). A simplified depiction of a DNN is shown in Figure 4.1: this network takes three inputs; has one hidden layer with four nodes; and produces two outputs.

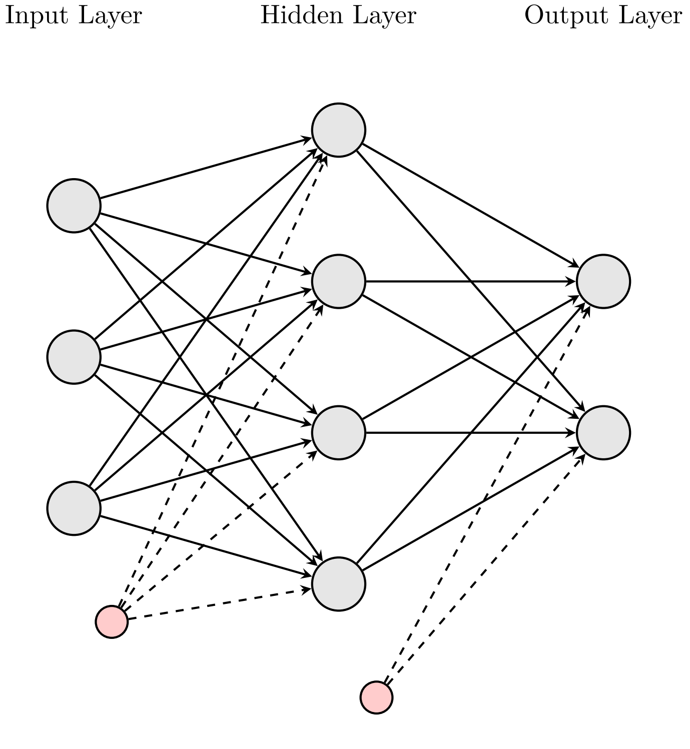


Figure 4.1: A simplified illustration of a DNN. The grey circles represent nodes, the solid lines weights, and the dashed lines biases. The small red circles are depicted just to give a start-point to the biases.

[1] The loss function can in fact take into account terms other than the distance between input and output; how this changes the goals and functions of NNs beyond the most basic cases will be described in the following sections.

To mathematically explain what the figure represents, we write this network in matrix form, using Einstein summation conventions familiar to physicists:

$$o_k = \sigma_2 \left[\sigma_1 \left(W^{(1)}_{ij} v_i + b^{(1)}_j \right) W^{(2)}_{jk} + b^{(2)}_k \right] \, , \tag{4.24}$$

where $v_i = (v_1, v_2, v_3)$ are the three input variables; $o_k = (o_1, o_2)$ are the two outputs; σ_i are the activation functions; and $W^{(x)}_{ij}$ and $b^{(x)}_i$ and the weights and biases for layer x. The biases are constant offset terms, and the activation functions are necessary in order to introduce non-linearity into the network [181]. Perhaps the most common activation function – though many others are used – is the ReLU (Rectified Linear Unit) function

$$\text{ReLU}(x) = \max(x, 0) \, , \tag{4.25}$$

which is displayed in Figure 4.2a. Particularly for classification problems the "softmax" function,

$$\sigma(n_i) = \frac{\exp(n_i)}{\sum_j \exp(n_j)} \tag{4.26}$$

displayed in Figure 4.2b, is used as the activation function of the final layer, as it ensures all the outputs can be (naïvely) interpreted as probability.

4.2.2 Losses

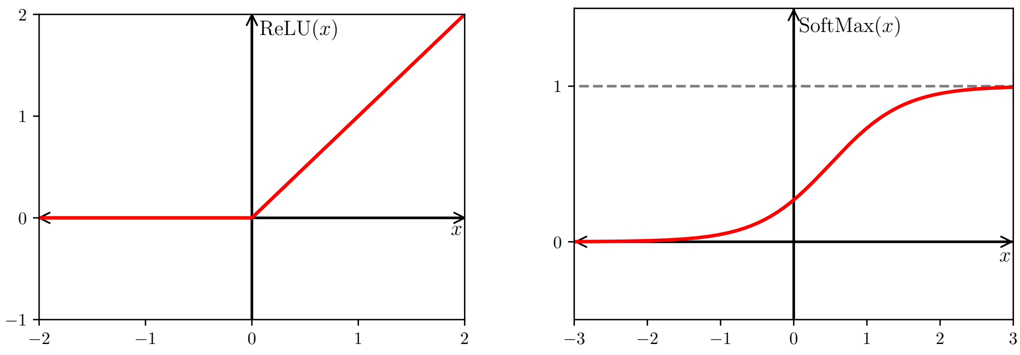
The loss function of a supervised machine learning problem is the function that measures how succesfully the network has replicated the "true" value for a given input. Therefore, the simplest conceivable loss function is just the absolute error:

$$L = \sum_i^N \left| y^i_{\text{true}} - y^i_{\text{pred}} \right| \, , \tag{4.27}$$

which has an obvious shortcoming: the magnitude function is potentially problematic for numerical differentiation, as its derivative is constant everywhere except from its minimum, where it is undefined. This can cause problems for the minimisers used in training, which will be discussed further in the next section.

An improved ansatz might be the mean-squared error:

$$L = \frac{1}{N} \sum_i^N \left(y^i_{\text{true}} - y^i_{\text{pred}} \right)^2 \tag{4.28}$$



(a)

(b)

Figure 4.2: Two common activation functions: ReLU (a), and SoftMax (b).

and indeed for many regression problems this may be a good choice.

For classification problems, where we know the true value is a unit column vector and that the NN will output a set of probabilities that sum to one, a natural choice is the cross-entropy:

$$L = -\frac{1}{N}\sum_{i}^{N}\left[y^{i}_{\text{true}}\ln\left(y^{i}_{\text{pred}}\right)\right] \ . \tag{4.29}$$

As NN training becomes more sophisticated, other terms may also be added to the loss function. For example, "regularisation" terms can be used to reduce over-fitting of the model (see next section) in exchange for a small trade-off in accuracy: the simplest of these, L1 and L2 regularisation, add in penalty to the loss function as either

$$L_{L1} = \ldots - \lambda\sum_{ij}|W_{ij}| \tag{4.30}$$

or

$$L_{L2} = \ldots - \lambda\sum_{ij}\left(W_{ij}\right)^2 \ ; \tag{4.31}$$

i.e. penalising the network if individual weights become large. These terms can be thought of as analagous to Lagrange multipliers in classical dynamics problems, enforcing a constraint by adding it to the term that needs to be minimised. In more complicated architectures, for example Generative Adversarial Networks (GANs), loss terms may also come from the relationship between outputs of different subnetworks within the model.

4.2.3 Training, over-training and validation

The ultimate goal of training a NN is to adjust the weights and biases such that the total loss across all possible inputs is minimised. In practice, we do not have access to an infinite set of possible inputs; rather we have a (hopefully) large dataset of inputs, with "true" values of the outputs in order to steer the training. Most NN training is done using some variety of gradient-descent minimiser and "back-propagation" method. First, a "forward-pass" runs the existing network on an input (or across several inputs) and calculates the loss. Then, in order to run a gradient based minimiser, we need to calculate the derivative of the loss – this is why the differentiability of loss terms and activation functions is essential.

Calculating derivatives: a chain rule example

Taking the simple example from equation 4.24, focussing on just the second layer of biases $b^{(2)}_k$, and assuming some loss function of the form $L(o_k)$, we can use the chain rule to obtain

$$\frac{\partial L}{\partial b^{(2)}_k} = \frac{\partial L}{\partial o_k}\cdot\frac{\partial o_k}{\partial b^{(2)}_k} = \frac{\partial L}{\partial o_k}\cdot\frac{\partial \sigma_2}{\partial b^{(2)}_k} \ . \tag{4.32}$$

For the final layer, this seems relatively straightforward, but the mathematics becomes rather onerous as we move backwards, unhelped by the somewhat cumbersome notation. Defining $z^{(x)}_i$, $a^{(x)}_i$ as the values of the nodes before and after the application of the activation function, and labelling the first layer (0) so that $o_k = a^{(2)}_k$, we rewrite as

$$\frac{\partial L}{\partial b^{(2)}_k} = \frac{\partial L}{\partial a^{(2)}_k}\cdot\frac{\partial a^{(2)}_k}{\partial z^{(2)}_k}\cdot\frac{\partial z^{(2)}_k}{\partial b^{(2)}_k} \ , \tag{4.33}$$

where we note that $\frac{\partial a^{(2)}_k}{\partial z^{(2)}_k}$ is just the derivative of the activation function σ_2.

Calculating derivatives: back propagation

Using our new notation, we can write the derivative of the loss with respect to the pre-activation nodes of the final layer (often called the error term) as

$$\frac{\partial L}{\partial z_k^{(2)}} = \frac{\partial L}{\partial a_k^{(2)}} \cdot \frac{\partial a_k^{(2)}}{\partial z_k^{(2)}} \, , \tag{4.34}$$

which we recognise as the first two terms from equation 4.33 – the derivatives with respect to all the weights and biases in the final layer can be calculated using this error term. Indeed, given the error term for an arbitrary layer x, $\partial L/\partial z_k^{(x)}$, we could use this to find the gradient of the loss with respect to the weights and biases that directly feed into that layer via

$$\frac{\partial L}{\partial W_{ij}^{(x)}} = \frac{\partial L}{\partial z_k^{(x)}} \cdot \frac{\partial z_k^{(x)}}{\partial W_{ij}^{(2)}} \, . \tag{4.35}$$

How do we get the error term for each layer? Given an error term at layer x, we can calculate

$$\frac{\partial L}{\partial z_k^{(x-1)}} = \left(\frac{\partial L}{\partial z_j^{(x)}} W_{jk}^{(x)} \right) \cdot \frac{\partial a_k^{(x-1)}}{\partial z_k^{(x-1)}} \, , \tag{4.36}$$

noting the implied summation over j inside the brackets.

From this equation, we can calculate the error terms backwards one layer at a time, thereby obtaining the gradient for all weights and biases, in a process called "back-propagation".

Minimisers

Armed with the loss and the gradients, we can feed these into a minimiser to obtain an improved set of weights and biases. Most NNs use some evolution of a basic gradient descent minimiser, i.e. taking "steps" in the direction of the steepest gradient to hopefully "walk" into the global minimum. This is quite unlike most other computational minimisation problems encountered in physics: the co-ordinates of the minimum (i.e. the actual weights and biases) are largely irrelevant, and indeed for many network structures including DNNs, there must by symmetry be many equivalent minima. Indeed, finding the true global minimum is not really required – as long as the loss is close to the best possible, how many other minima also have a similar loss is irrelevant.

Typically, optimisers will allow for a variable learning rate (i.e. step size), and will incorporate some form of "randomness" into the steps, in order to avoid getting stuck in local minima. The current default in Keras and PyTorch (see Section 4.2.6) is the ADAM optimiser [182].

Practicalities

The number of samples from the input dataset used for each update of the weights and biases is known as the *batch size*. A smaller batch size introduces additional randomness to the training, as a small sample of the training dataset is less likely to look like the original training dataset. This can be useful for avoiding local minima, however it also slows the training down and may lead to convergence problems, as successive batches move the weights and biases in completely contradictory directions.

One frequent danger in NN training is over-training. In very simple terms, this occurs when by (unfortunately very common) statistical misfortune, there arise some features (in the broadest possible sense) in the training dataset that are not actually representative of the original dataset. In this case, given another dataset drawn from the same distributions, the network that has been trained will perform significantly worse when tested on samples drawn from the original dataset.

To mitigate and quantify overtraining, rather than just using the entire dataset for training, it is common to split it into two or three: into “train”, “test” and “validate” samples (though “test” and “validate” may occasionally be merged). Unsurprisingly, the training sample is used to update the weights and biases as laid out above. Once the entire training sample has been run through once – typically described as an *epoch* – then the total (or mean) loss is evaluated over the entire test dataset. Because this data was not in the training sample, this is a test for overtraining. Though the exact procedure can vary, it is most common to discard the entire epoch of training *unless* the total loss for the test dataset is less than the total loss for the test dataset after the last succesful epoch of training. A common visual cue for overtraining is that when plotting the test and train losses as a function of the number of epochs, the test loss will level off earlier and at a significantly higher value than the train loss, as shown in Figure 4.3. The final data sample, the validation sample, is used as a further cross-check once training is complete, and can also be used to compare the performance off diferent NNs.

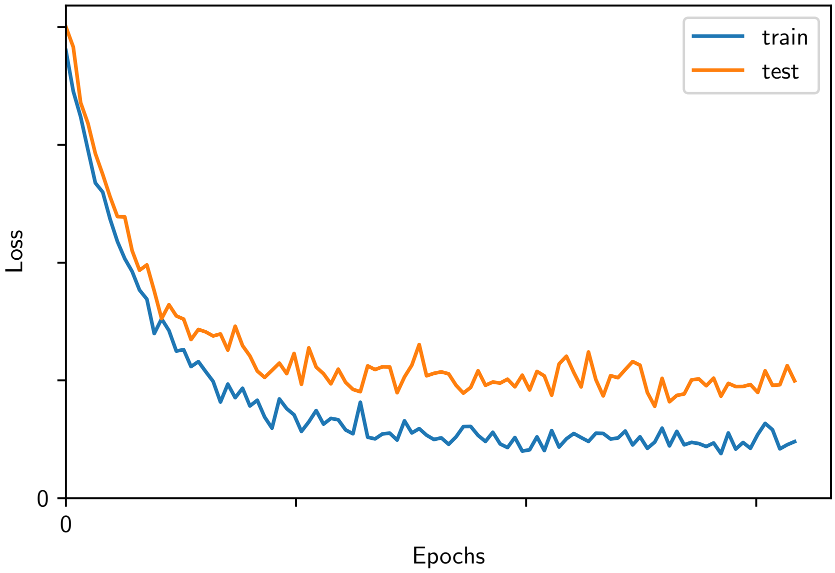


Figure 4.3: An illustration of how the total loss will evolve for the “test” and “train” samples during neural network training as a function of the number of epochs. A certain amount of overtraining is to be expected, and hence the “train” sample settles to a lower average value.

4.2.4 Other layers and architectures

In our discussion so far, we have only considered the networks with dense layers, layers where each node in the preceding layer is connected to all the nodes in the next. We will briefly discuss a small selection of other possible layers, though a full survey is far beyond the scope of this thesis.

Convolutional layers group together neighbouring nodes, outputting the sum, maximum or other function of all the nodes values. This is perhaps most useful in tasks like image processing where the data is naturally two-dimensional, and emphasising connections between adjacent points is intuitive.

Convolutional layers are just a special case of a Graph Neural Network layer. Graph Neural Networks are designed to operate on data that has a graph structure – i.e. some nodes are connected directly only to some others (e.g. only to neighbouring nodes in the convolutional example). This can be particularly useful in physics where our input data are not all alike and do have special relationships with one another: for example, jet kinematic variables have a special relationship with the other kinematic variables of the same jet; and ideally the NN should treat all jets equally, rather than resorting to artificial solutions like p_T or mass ordering, like we do for DNNs. One particularly effective use of GNN's in physics is the Atlas GN1 jet-flavour tagger [122], which shows very significant improvements over its DNN and BDT based predecessors.

Dropout layers – alongside various forms of regularisation – are one of the simplest tools for ML developers to combat overtraining. In the most basic case, a dropout layer depends on a single parameter, the dropout probability, and simply randomly "switches off" nodes or weights from the preceding layer, at a rate equal to the dropout probability, preventing the network from relying too heavily on any one feature.

4.2.5 Measuring NN performance

Although the loss function determines the direction in which the minimiser drives the network, the final value of the loss for the entire test or validation datasets does not necessarily give us a full picture of how well the network performs. We may wish to visualise the performance of our network in a more expressive way than a single number.

A popular way of evaluating the performance of classifiers is a receiver operating characteristic (ROC) curve. This plots the true-positive rate against the false-positive rate as a parametric function of the cut-value, as illustrated in Figure 4.4. A large area under the curve (AUC) generally implies a better trained classifier, and the ROC curve can also assist in deciding where to place the cuts that will be used – although once a classifier is finally put into practice, the signal-efficiency and background-rejection ratios at the chosen working point will remain the ultimate test of performance.

Of course, classifiers need not be binary: they can tag multiple categories, or they may be signal-vs-background classifiers that need to be resistant against many different backgrounds. In this case, we can plot multiple ROC curves to illustrate the discrimination capabilities of the classifier between each pair of possible samples.

Often, the end-goal of a NN is to replicate some sort of distribution: for example, in Chapter 6 we will aim to use a Neural Net to help reproduce various kinematic distributions, such as leading-jet mass or transverse momentum. Perhaps the simplest way of measuring how closely related two distributions are is by re-using the χ^2 test [85]:

$$\chi^2(P,Q) = \sum_x \frac{[P(x) - Q(x)]^2}{P(x)} \tag{4.37}$$

where P, Q are discrete distributions we are comparing, which share the same domain $\{x\}$. Given our aforementioned examples (such as the mass distribution of a jet) are continuous variables, the presence of a sum rather than an integral in equation 4.37 may be surprising. However, in practice we will (almost) always be operating on binned histograms, which discretises our data.

Another common distance measure between distributions is the Kullback-Leibler (KL) divergence:

$$D_{KL}(P,Q) = \sum_x \left[P(x) \ln \left(\frac{P(x)}{Q(x)} \right) \right] \tag{4.38}$$

which we note is explicitly not symmetric. The KL divergence is also a common choice in loss functions, though it will not be used as such in this thesis.

4.2.6 Tools for neural networks

Part of the reason for the explosion in popularity of NNs over the last decade has been the development of free, simple-yet-extensible tools for building and training them. These tools mean that many of the complex-but-repetitive steps – such as running a minimiser or automatic differentiation of activation functions and losses – are automatically taken care of. By hiding some of the most computationally expensive steps in precompiled libraries, most NN development can be carried out in Python, a much more convenient environment for data-analysis.

Two of the most popular tools are Keras [183] – which sits within the broader TensorFlow [184] ecosystem – and Pytorch [185]. Both have been used extensively in particle physics, and both are primarily Python-based tools – though they also have interfaces to other languages such as C++ – and provide many additional features. Both are also under constant development; as new developments arise and best practice changes, new versions of the packages are frequently released.

One key feature provided by both Pytorch and Keras is the ability to carry out training on graphical processing units (GPUs), instead of the central processing units (CPUs) that lie at the heart of most desktop PCs and laptops. In simple terms, GPUs are designed to carry out simple operations (matrix addition, multiplication, etc.) massively in parallel, as opposed to CPUs, which are better optimised for fewer though more complex tasks. Though the original aim was faster frame-rates in video games, this same set of features is also very useful for training NNs, where large matrix operations are a very significant part of the workflow.

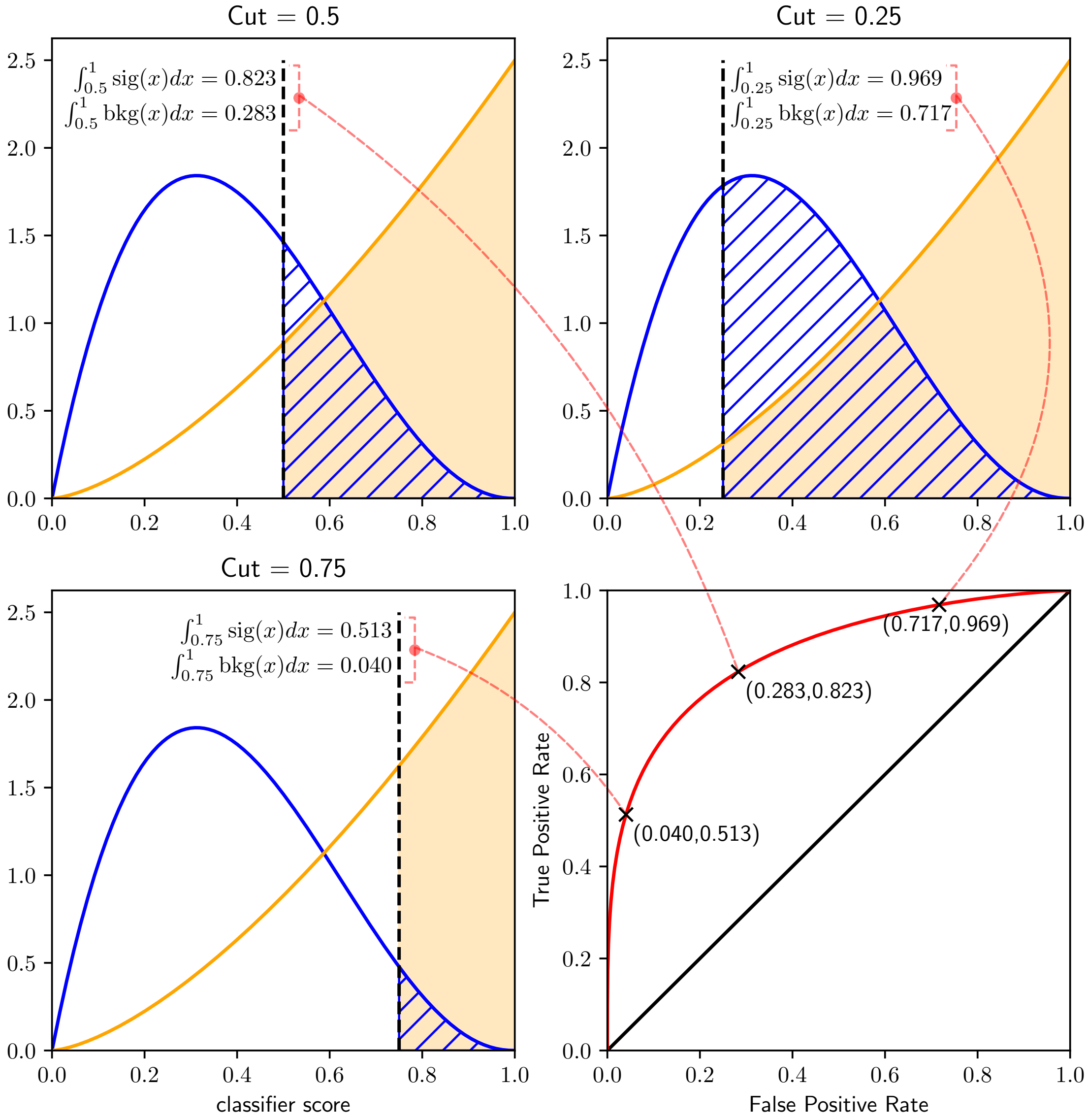


Figure 4.4: The construction of a ROC curve (bottom right), from the distribution of the classifier score for the signal (orange) and background (blue) samples, with snapshots at three values of the cut.

4.2.7 Practical use in collider physics

Perhaps the most common use of NNs in particle physics is in object-tagging. This can be either in reconstruction – for example, tagging reconstructed jets as b-jets, or tagging an EM calorimeter signature as photon or electron – or after reconstruction as part of the analysis, for example tagging an already reconstructed jet as originating from a W-boson or not. Analyses may also train a tagger to discriminate between signal and background given event-level information, and cut directly on this discriminant.

However, there are other roles for NNs outside of reconstruction and analysis. NNs can be used in reweighting procedures designed to make our MC workflows more efficient. Examples include the DCTR procedure [186], as well as the CARL method that will be discussed in Chapter 6.

One potential problem with using NNs in collider analyses is their "black-box" nature. When applying a cut on a "normal" observable, like a jet p_{T} or the aplanarity of an event, it is relatively intuitive what is being selected and what is being removed; and the behaviour of the cut if variables are shifted a small amount in any direction can be straightforwardly understood. When cutting on a NN-score, this is not the case: exactly what features and correlations the NN is putting into the score are not at all clear.

This can be partly ameliorated by studying feature importance – sequentially removing or adding input variables or sets of input variables, and quantifying how much better or worse the network performs. Other methods of "interpretable" machine-learning are also growing in popularity.

4.3 Tools for reinterpretation

In collider physics, "reinterpretation" describes the practice of using results from an experimental analysis to apply constraints in a new context, for example by using results from a search for one model of dark-matter to constrain a different model of dark matter. Currently, there are many at least moderately well-motivated BSM models, which often require many parameters to fully describe (such as the MSSM). However, even for 5 000-member experimental collaborations, there are only so many models that can be specifically targetted, and only so many corners of the parameter space that existing computing resources can cover. Our understanding of both the SM and BSM theories will also evolve over the ensuing decades; and it would be a great waste of the LHC's legacy if we were not able to combine (expensively obtained) LHC results with what will become cutting-edge theory. In this context, the ability to continue to be able to do tests on more varied and more complex models is essential to the future of particle physics.

The majority of reinterpretation tools rely on re-implementing a simplified version of the original analysis logic to obtain an approximation to their signal counts, though there are notable exceptions such as SModelS [187] or HiggsBounds [188]. The level of detail varies between reinterpretation tools. At one end, the Atlas-internal Recast tool [189], is effectively equivalent to rerunning the analysis within the experimental collaboration, with the commensurate CPU cost. New signal samples go through the same (AtlFast) detector simulation that the original signal grids would have gone through, and – thanks to preservation in Docker-image form – the software that makes signal region selections and performs statistical analysis should be near-identical. At the other end of the spectrum, tools like Gambit make multiple approximations in the event-generation and detector-simulation stages in the interests of computational efficiency.

A similar spectrum applies to the detail of the statistical model the tools use. Once again, Recast uses a full statistical model very close to that used in the original measurement. In contrast, independent recasting efforts may be limited to using a highly simplified setup consisting of a Poisson likelihood using the pre-fit background count and the signal count obtained by the re-implementation of the collider analysis as the counts, and the errors on these two counts as the two nuisance parameters.

4.3.1 A very brief introduction to Rivet, Yoda and Contur

Rivet and Yoda

Rivet (robust independent validation of theory and experiment [190]) is a software package for preserving the logic of collider analyses. In very basic terms, Rivet reads in HepMC events, runs over a set of user-specified analyses, and outputs key observables from those analyses – anything from invariant mass distributions to event counts in signal regions – stored in Yoda files.

Yoda [191] is a histogramming software, originally written for use with Rivet but capable of being used standalone. Written in `C++` with a Python API, Yoda provides a Root-free way of filling histograms of arbitrary datatypes in arbitrary dimensions, and provides both ASCII (`.yoda`) and binary (HDF5-based `.yoda5`) formats for saving and distributing the histograms.

Individual Rivet analyses consist of a single `C++` file, containing a `C++` class, which inherits from the `Analysis` class. The analysis needs to contain only three methods: `init`, for declaring which physics features[2] of the event will be needed by the analysis; `analyze`, for carrying out the analysis HepMC event by HepMC event; and `finalize` for carrying out any tasks – often based on normalisation or statistics – that need to be carried out after every event has been studied.

Historically Rivet was primarily used to preserve the logic of unfolded measurement analyses (for uses such as MC event generator tuning [146]), meaning that no detector simulation was required, and that particle-level events could be directly analysed and compared to the unfolded results published by the experiments. However, Rivet is also perfectly capable of – indeed, well suited to – the preservation of detector-level searches. The main reason for significantly more measurements than searches being preserved in the Rivet analysis library appears to be largely institutional inertia.

When required, in place of a full detector simulation, Rivet uses a suite of detector-smearing functions to approximate detector effects, in a similar manner and with similar results to Buckfast in Gambit (see Sections 3.5.3 and 4.3.2). Customised smearing functions can also be defined for individual analyses, if unconventional object definitions are required.

Analyses are often contributed by the authors of the original study; though others are written by either the Rivet authors or Rivet users particularly invested in reproducing certain results. Analyses are not limited to the LHC: there are many examples from smaller or older experiments such as at LEP, HERA and RHIC.

Projections - Rivet's secret sauce

For the average Rivet user or contributor, the section above probably constitutes a sufficient introduction. However, for the purposes of this thesis – particularly in preparation for Chapter 7 – it is necessary to examine the nuts and bolts[3] in closer detail.

In principle each individual analysis routine could interact directly with the HepMC event record, each analysis carrying out many computationally intense tasks on its own – each deciding which particles are long-lived enough to count as stable, each clustering hadrons into jets, and each identifying final-state leptons and photons by iterating through all the particles in the event.

Instead, as Rivet developed from the HZTool software used at HERA [192], a key design consideration was to minimise the number of repeated identical calculations in computationally intensive processes. A `Projection` is some physics object(s) that can be obtained from the event record that may be required in multiple places, either within a single analysis or across multiple analyses. Projections range from the very

[2] For those familiar with Rivet, read "projection" for "feature" – this will be addressed in the next section.

[3] Pun intended.

simple – such as `FinalState`, which simply returns all final-state particles, optionally with some kinematic cuts applied – to the very complex, such as the `Centrality` projection used for heavy-ion analyses. Projections often depend on other projections – for example, the `SmearedJets` projection takes an unsmeared set of Jets as an argument to its constructor, while any `FinalState` that includes kinematics will – automatically – work from a `FinalState` without any cuts applied.

In addition to the increase in computational efficiency, projections also significantly lower the technical bar to writing a Rivet analysis – there is no need to interact directly with HepMC or `FastJet` types (noting especially `FastJet`'s use of raw pointers); and complex objects that might require many lines of code to extract from the event record can be accessed in a single line.

Projections are `declare`d in the `init` method of an analysis – where they are given a name – and then can be accessed from the body of the `analyze` method using the `get` function and the previously assigned name. As part of the declaration in the `init` method, Rivet will work out which other projections are depended on, and whether or not an equivalent projection is already registered. This complex process, in the Rivet 3.X series, was carried out by a single global `ProjectionHandler` object for each Rivet run, which prevented Rivet from being easily parallelised. This will be discussed in even more excruciatingly technical detail in Section 7.5, where we examine how to eliminate the single-threaded `ProjectionHandler`.

Contur

If BSM physics is accessible at the LHC, then it is quite likely that it will make small but measurable changes to SM measurements, which we should have "already-seen". Contur – constraints on new theories using rivet [193, 194] – harnesses the vast library of SM measurements preserved by Rivet to place constraints on BSM theories.

In its simplest use case, BSM events are generated by some external generator and then run through a selection of Rivet analyses relating to precision SM measurements. Contur reads the resulting Yoda files and, based on the BSM population of SM signal regions, decides if the model would have been "already-seen" at the LHC.

Bins in unfolded SM measurements always contain large numbers of events – otherwise unfolding would not be possible. This is in stark contrast to many BSM searches, where some bins may not even observe any events during data-taking[4]; this feature is responsible for many of the differences between Contur and reinterpretation tools focussed on reinterpreting searches. In this large-population limit, the Poisson likelihood in equation 4.3 approaches a Gaussian, and the test statistic t_μ can be approximated for a single bin by the χ^2 statistic

$$\chi^2_\mu = \frac{((\mu s + b) - n)^2}{\sigma^2} - \frac{(b - n)^2}{\sigma^2} \,, \tag{4.39}$$

where σ is the standard deviation of the counting test, and b is a background estimate, which in the Contur case will be the SM prediction for the measurement in question. Here the systematic errors have been collapsed into σ, but if more detailed error breakdowns are provided, Contur also has some capability to include these. Early versions of Contur, instead of using the SM theory predictions, made the assumption that the data and the SM model matched exactly (not necessarily a bad assumption, given the lack of significant deviations observed so far). This meant that Contur could only exclude rather than favour BSM theories; but this is no longer the default treatment, and all significant Contur results used in this thesis use SM theory predictions as the background. Where systematics and correlations are available in the Yoda histograms that Contur reads in, these are multiplied into the likelihood.

[4] For example, in Section 5.5 it becomes a point of interest that the "SR-Gbb-2100-1600" region in the Atlas SUSY search [177] contains no observed events.

In order to avoid double counting the impact of individual BSM events that may change the results of multiple different SM measurements, Contur uses a "pool" system, grouping together measurements that are likely to inhabit an overlapping phase space and only counting the strongest contribution within each pool.

Running standalone, Contur uses this test statistic to output a $\mathrm{CL_s}$ statistic following equation 4.22; but it can also output the LLR for compatibility with Gambit[5], following

$$\mathrm{LLR} = \frac{-\chi^2(s)}{2} = -\frac{1}{2}\prod_i \left[\frac{b_i + s_i - n}{\sigma}\right]^2 . \tag{4.40}$$

A small number of BSM searches have been added to its database of results, and while likelihoods from these are not used by default, they can be used if requested.

Contur is also capable of running in more complex modes of operation than being fed one Yoda file at a time. It can also run simultaneously across a parameter grid, and can even be configured to steer certain Monte-Carlo event generators. The Contur-oracle [195], is a bolt-on to out-of-the-box Contur that uses ML methods to decide which sets of parameter points would contribute best to defining an exclusion contour. There is also an extensive API that should allow it to be included within other projects.

4.3.2 A very brief introduction to Gambit

Whatever is suggested in grant applications, the heart of the Atlas detector is not the only place where evidence for new physics may be observed. Astrophysical observations can tell us more about how dark matter behaves; atomic physics can place limits on fundamental quantities such as electron dipole moments (EDMs) of fundamental particles [196]; and direct DM detection experiments, even when not detecting anything, can place limits on the masses and nature of dark matter candidates [197, 198].

A global statistical fit comprises of creating a statistically-rigorous composite likelihood at each parameter point, and then intelligently moving around the parameter space to sample this likelihood [85]. This is a much more complete treatment than just overlaying exclusion curves published by experiments for similar models, or calculating exclusion $\mathrm{CL_s}$ values for a range of analyses independently and excluding the point if any fall above 95%. While it is possible to do global fits just combining collider analyses[6], the setup also naturally lends itself to incorporating likelihoods from other physics sources.

The aim of the "global and modular BSM inference tool", Gambit [199], is to perform global statistical fits that use results from a wide range of fields to both constrain new physics models, and find regions of these models' parameter spaces that are either under-constrained or even favoured relative to the SM.

Reflecting its eponymous modularity, Gambit contains many Bits with self-explanatory names such as FlavBit [200] for flavour physics, DarkBit [201] for direct and indrect dark-matter detection, or – most pertinently to this thesis – ColliderBit [162] for collider physics, primarily the recasting of LHC searches for BSM physics. Not all Bits directly calculate likelihoods – ScannerBit exists to co-ordinate the scanning of the multi-dimensional parameter spaces that Gambit interests itself in, and physics modules such as SpecBit exist to carry out specific calculations that will likely be useful to other physics modules in turn – see the roles of SpecBit and DecayBit in Figure 4.5. Bits also have access to multiple external codes, collectively known as "backends". In order to allow these previously developed tools to be dynamically loaded at run-time, Gambit contains a script called BOSS (Backend-on-a-Stick Script) [199], that defines abstract versions of external classes within the Gambit code so that external classes can then be dynamically loaded. "BOSSing" a backend code to ensure all its classes load correctly can be a non-trivial task.

[5] As will be described in Chapter 7, the ability to extract this directly from the API was added as part of work for this thesis.

[6] Indeed, Gambit has carried out such studies in the past.

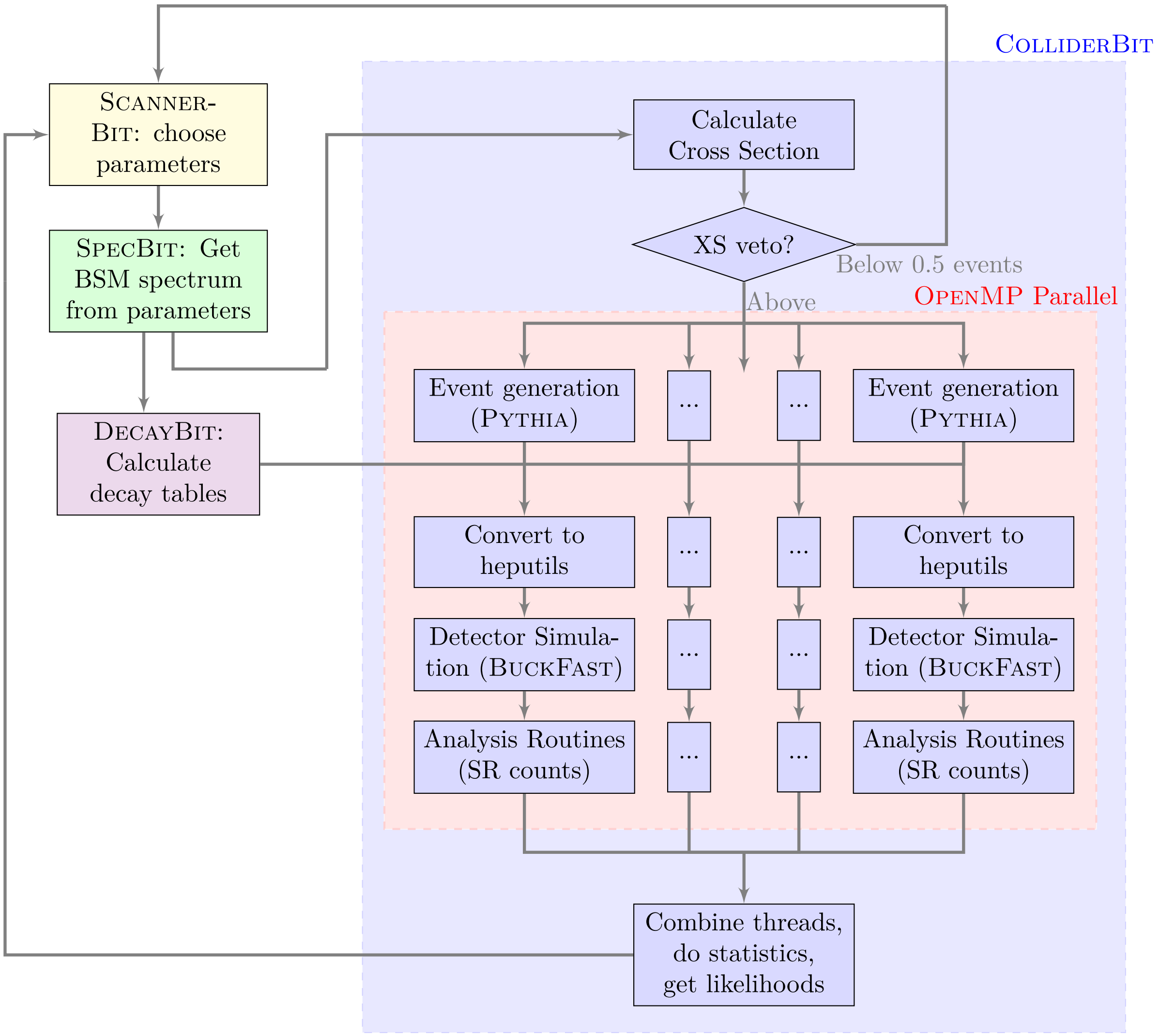

ColliderBit
ScannerBit: choose parameters
SpecBit: Get BSM spectrum from parameters
DecayBit: Calculate decay tables
Calculate Cross Section
XS veto?
Below 0.5 events
Above
OpenMP Parallel
Event generation (Pythia)
...
...
Event generation (Pythia)
Convert to heputils
...
...
Convert to heputils
Detector Simulation (BuckFast)
...
...
Detector Simulation (BuckFast)
Analysis Routines (SR counts)
...
...
Analysis Routines (SR counts)
Combine threads, do statistics, get likelihoods

Figure 4.5: Diagram describing the essential functions of ColliderBit and how they relate to other important Bit's within Gambit. It is not exhaustive: other possible running modes exist, some Bits are not shown, and some backends are excluded. Within the ColliderBit block only, the horizontal axis indicates concurrency.

Recent Gambit studies include large multi-dimensional tests of various SUSY models that rely on very intense Monte-Carlo event generation and collider likelihoods [202,203]; but also smaller studies that may not even harness collider likelihoods at all, such as placing cosmological constraints on Axion-Like Particles [204]. This illustrates the highly modular nature of the tool, which is achieved through its capability and dependency resolver system. A `capability` in Gambit describes the purpose of a module function; and multiple functions with the same objective (but possibly achieved through different methods) may exist. Each Gambit run is defined by a yaml file that defines which observables and likelihoods need to be calculated; and then the dependency resolver works out which modules and functions from Gambit will be required to obtain these results, and loads only these for the run. Where there are multiple possible functions for a single capability, the yaml file is also responsible for specifying exactly which function is needed. The result of this is the dependency resolution graph, one of which is illustrated – in part only – in Figure 4.6. A full description of the dependency resolution system and the steering yaml – sufficiently rich that they are almost a programming language in their own right – can be found in Reference [199].

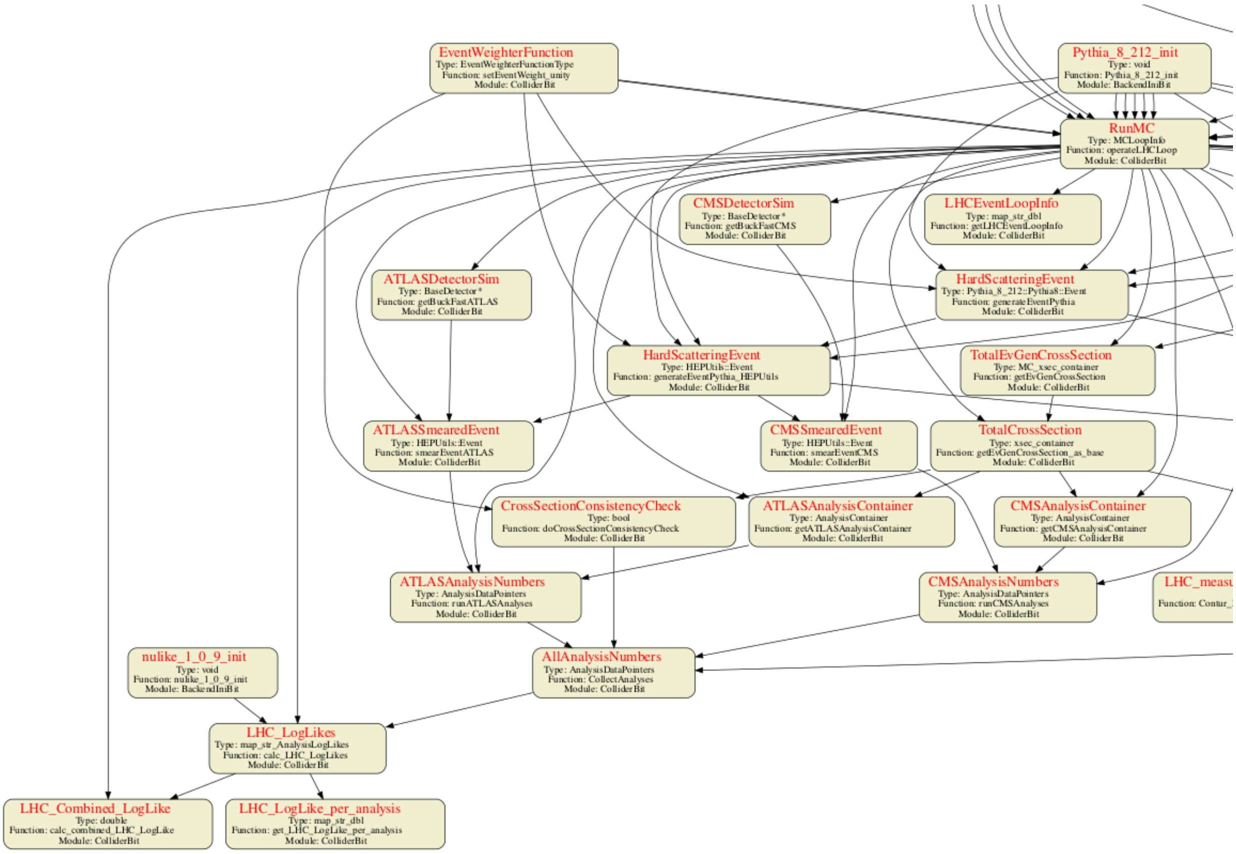


Figure 4.6: A small extract from a GAMBIT dependency graph, showing how various GAMBIT functions interrelate, and which capabilities provide them. Note how the observables at the bottom (e.g. the combined LogLike) depend on a long chain of other functions.

In more detail: ColliderBit

The central feature of ColliderBit is the reinterpretation of many LHC search analyses, primarily focussed on SUSY models, including 7, 8 and 13 TeV analyses from both Atlas and CMS. Figure 4.5 shows where the analyses sit within the ColliderBit workflow, and how ColliderBit interacts with the rest of Gambit. Particularly noteworthy are: the internal, in-memory event generation using the backended Pythia – necessary for performance reasons in the HPC context as Pythia, unlike other MC generators, operates with much less disk I/O; and the very-fast BuckFast detector-emulation, allowing the reinterpretation of detector-level searches without the performance penalty of Delphes (discussed in Section 3.5) or a similar tool. As shown in Section 3.5.3, BuckFast works by applying (detector-dependent) Gaussian smearing to the four-vectors of particle-level objects; and achieves similar results to Delphes in a fraction of the time.

Though ColliderBit does not contain an intricate projection system like Rivet, the conversion of events from HepMC to the heavily-simplified HEPUtils format offers some of the same performance benefits: most notably, HEPUtils events have jets already clustered, so jet-clustering only needs to be run once.

Statistically, unless the analysis provides a `pyhf` JSON file (as several Atlas analyses do) or a simplified likelihood (as some CMS analyses do), ColliderBit calculates likelihoods starting from the Poisson formula in equation 4.3. There is not yet any support for the CMS Combine framework, and the difficulties in having to interface to a docker container from many separate MPI processes may make it more practical to wait for the release of HS3.

For each bin, the entire likelihood calculation depends on the signal estimate s provided by the ColliderBit analysis, and the observed bin count and pre-fit background estimate – n and b – which will come from the original analysis. Associated with this are two nuisance parameters σ_s and σ_b, encoding all the systematic errors on the signal (from ColliderBit) and the background (from the original paper).

To reduce the dimensionality of the problem and further increase Gambit's computational efficiency, the two nuisance parameters are absorbed into one nuisance parameter ξ, which is assumed to follow a log-normal distribution (LN) with $\sigma_\xi = (\sigma_s^2 + \sigma_b^2)/(s+b)^2$. Then the final form of the likelihood, having marginalised over ξ, is:

$$L(n|s,b) = \int_0^\infty \frac{[\xi(s+b)]^n \exp\left(-\xi(s+b)\right)}{n!} \,\mathrm{LN}(\xi|\sigma_\xi)\, d\xi \;, \tag{4.41}$$

To steer the scanner and determine whether points are favoured or not, Gambit uses the log-likelihood ratio LLR, defined as

$$\mathrm{LLR} = \ln L(n|s = s_{\mathrm{BSM}}, b) - \ln L(n|s=0,b) \;, \tag{4.42}$$

where a high LLR implies that the BSM model is favoured. Note that other sources may include a factor of -2 in the LLR definition.

Where regions within an analysis are known to be orthogonal, the log-likelihoods can simply be added together, but where they are correlated (and a pyhf file or simplified likelihoods information is not provided), then the best available procedure is to pick the region with the highest *expected* limit on the new physics (i.e. most negative expected LLR). It is worth stating that the statistical benefits from being able to combine regions are significant, and that additional likelihood information is always very welcome to re-interpreters.

When combining likelihoods across analyses, Gambit by default assumes that the analyses are uncorrelated. Given the very small population of signal regions in most SUSY searches, this is a much more reasonable assumption than it would be for SM measurements. Particularly for SUSY analyses, there has been some effort by the experimental groups to design analyses that do not directly overlap.

Other useful but not integral features of ColliderBit include limits on various SUSY processes from LEP (where these are still competitive) and the ability to place BSM limits based on Higgs observables via the HiggsBounds [188] backend.

Scanning in many dimensions

New physics models – and even their phenomenological simplifications – are too large to be sampled efficiently by LHC experiments, which will typically sample a simplified model, considering only 1-3 parameters. Gambit is ideally positioned to take up the task of studying those higher-dimensional spaces that may be within the energy frontier of the LHC but lie beyond the computational frontier of the experimental collaborations. ColliderBit is able to run collider simulation far faster than the experiments due to massive parrelelism, performance-enhancing compromises (like BuckFast detector emulation or massively simplified systematics), and the ability to quickly veto points using constraints from other areas of physics before running expensive event generation.

However, most important in surveying these many-dimensional parameter spaces is the coupling of Gambit to the world-leading statistical scanning algorithms in ScannerBit. The algorithms efficiently search for points of maximum likelihood in the many dimensional parameter space, and in this process will also sample many points across the space. Scanners in ScannerBit include Markov-Chain Monte-Carlo (MCMC) scanners such as GreAT [205] and ensemble MCMC scanners like T-Walk [206]; however, Gambit studies [207] suggest that the most effective scanner for profile likelihood evaluation in the dimensionality ColliderBit scans are usually interested in is either the differential evolution scanner Diver or nested-sampler MultiNest [208]. Recent additions to ScannerBit mean that users can also interface to any Python-implemented scanner of their choice.

Gum

The Gambit universal model machine, Gum [55], is a tool designed to allow the simple importing of arbitrary BSM models into Gambit. Before Gum, adding a new model to Gambit was a complex procedure: to use the new model in ColliderBit, this potentially included writing out decay width expressions or Pythia matrix element code by hand.

Gum automates this procedure, taking a Lagrangian as input (in either FeynRules [209] or Sarah [210, 211] format). In addition to the Lagrangian, users provide a minimal configuration file, and then the rest of the procedure is handled by Gum. MadGraph5_MC@NLO is used to write matrix element code for Pythia, and CalcHep is used to calculate decay widths for DecayBit. If other Bits are used, other packages carry out additional tasks relevant to those Bits, such as Dark-Matter annihilation rates for DarkBit, also obtained via CalcHep. Although it is currently limited to $2 \to 2$ processes, Gum significantly expands the range of physics models that can be easily studied by Gambit.

Interface to Rivet and Contur

It is worth noting that, as of 2022, while ColliderBit did leverage the results of many LHC searches, it did not make any use of the results from LHC cross-section measurements. Indeed, constraints from such measurements had never before been used directly in a global BSM fit [203] – though in the interests of full-transparency, a small *post-hoc* study was carried out using Contur, examining the most excluded points from a previously published Gambit study [212]. Therefore, enabling Gambit to make use of Contur's constraints on new physics from measurements could potentially significantly increase constraining power without the requirement for generating a huge amount of new software.

4.3.3 Overview of similar reinterpretation tools

CheckMATE

CheckMATE (check models at terascale energies) [213] is a reinterpretation tool focussed on reinterpretting BSM (primarily but not exclusively SUSY) searches from Atlas and CMS. Compared to Gambit, it makes fewer sacrifices for the purpose of computing efficiency: it uses MadGraph5_MC@NLO for MC event generation (though users can also supply their own HepMC files), and Delphes detector simulation (introducing a Root dependency), which is much slower than BuckFast. This means it is potentially slightly more accurate at evaluating likelihoods for individual parameter points, but incapable of scanning the vast multi-dimensional spaces of interest to Gambit. CheckMATE also contains a large library of long-lived particle (LLP) searches for BSM particles that decay at significantly displaced vertices, which Rivet and ColliderBit do not (yet) have the ability to reproduce.

SimpleAnalysis and related tools

SimpleAnalysis [214] is an analysis preservation tool for Atlas SUSY analyses, written and maintained by the Atlas SUSY group. It bears many superficial similarities to Rivet: strong institutional support from a particular segment of the experimental community; a primary focus on preservation rather than statistical analysis; detector effects modelled by smearing (though Delphes is available); and a very similar analysis structure.

Being built on the Atlas software stack, SimpleAnalysis is only accessible to external (i.e. non-Atlas) users via a Docker image (making it difficult but not impossible to include within a larger non-Atlas workflow), and the externally available version does not contain any detector emulation at all, which for analyses that use tight object reconstruction definitions can be very problematic. For example, the ATLAS search [177] reinterpreted in Section 5.5 has some signal regions that require 4 b-jets. Given a 77% b-jet tagging efficiency, on the b-tagging efficiency alone the SimpleAnalysis implementation will obtain bin counts almost three times those obtained via the "correct" implementation[7].

Nearly all recently published ATLAS SUSY analyses have come with a SimpleAnalysis implementation. Even where it isn't useful within the SimpleAnalysis framework itself, the analysis `C++` code can be treated as a form of detailed pseudocode that documents the analysis procedure, and is often invaluable to reinterpreters – including for Section 5.5 of this thesis.

MaPyDe [215] is, in broad terms, to SimpleAnalysis what CONTUR is to RIVET: a framework that steers an event generator (in this case MADGRAPH5_MC@NLO, to be followed by PYTHIA for event generation and DELPHES for detector simulation) and uses the obtained signal counts together with the observed data and expected background to produce likelihoods and exclusions. Like SimpleAnalysis, MaPyDe needs to be run inside a Docker container. SimpleAnalysis is also used in ATLAS pMSSM scans [216], to provide a quick "first-check" of points before running the full RECAST framework.

MADANALYSIS5

MADANALYSIS5 [217] – from the same family of tools as the MC event generator MADGRAPH5_MC@NLO – is another preservation framework for BSM searches. Written in `C++` with a similar structure to RIVET; it boasts a very extensive set of SUSY and exotics BSM search analyses from the LHC. Because of its compatability with the other MAD programs, the most common way to run MADANALYSIS5 is directly from the MADGRAPH5_MC@NLO interface after event generation.

Historically, detector simulation has been done through DELPHES, though BUCKFAST-style smearing machinery is also available [218]. MADANALYSIS5 has produced some very accurate recasts: not just accurately reproducing bin-counts or cutflows at individual points, but also exclusion contours for the models considered in the analysis to incredible levels of detail. However, because this sometimes involves extensive tuning of the detector simulation [219], it is reasonable to ask how valid these recasts may be when applied to new signal models.

[7] $0.77^4 \approx 0.35$

Part II

Research

Chapter 5

Preservation and re-interpretation of collider analyses reliant on neural networks

5.1 Motivation

The field of particle physics has always stood at the forefront of Machine-Learning (ML) research. The same challenge – of having to deal with almost unimaginably large datasets – that led to the World Wide Web being developed at CERN [220], also meant that physicists had access to the scale of "big data" that required advanced statistical techniques like ML long before other fields. In 1989 – almost a decade before Google was founded and many decades before "AI" achieved its current buzzword status – Reference [221] considered the possible use of Neural Nets (NNs) in track reconstruction at the D0 detector at FermiLab. In the same year the plans for the eventually scrapped Superconducting SuperCollider (SSC) included the possible use of NNs for track reconstruction and photon vs electron identification in the calorimeter [222][1].

As Run 2 of the LHC progressed, machine learning technology became more widely available through openly available tools such as TensorFlow [184], Keras [183] and PyTorch [185]. The use of networks, not just in object reconstruction, but in defining observables central to the analysis flow – such as a signal vs background classifier – became increasingly common. By contrasting similar Run 2 analyses carried out at 36 fb^{-1} and 139 fb^{-1}, one will often notice that in order to improve the sensitivity, an ML based discriminant has been introduced: for example the Atlas search for gluino pair-production in multi-b-jet final states added the DNN which will be studied in Section 5.5, in order to improve the sensitivity further from the 36 fb^{-1} paper [223] to the full Run 2 139 fb^{-1} paper [177].

This potentially poses significant problems for those who wish to reuse these analyses. Firstly, there is the simple problem of data availability: unlike cut-and-count analyses where cuts can simply be described in the paper's text, NNs need to be saved and stored in a public location; and potentially they need file format conversions or disentanglement from experiment-internal code. For an experimental community that is already stretched very thin, these steps can easily be overlooked, leaving behind analyses that are highly resistant to reinterpretation.

A more fundamental question asks whether or not it is even valid to apply experimental neural nets,

[1] Though it should be noted, at this point in history, processing $\mathcal{O}(10\,000)$ events through a 3-layered dense NN in $\mathcal{O}(\text{hours})$ was considered very fast – a feat most modern laptops can handle in seconds.

trained on reco-level data, to the truth-level[2] (possibly smeared) outputs used by NNs. This will depend on the choice of input variables, and how well detector-emulation mimics these inputs; but even for well reconstructed inputs, it is possible that a simplified approach will not contain the correlation information that a reco-level neural net may depend on.

5.1.1 A short comment on efficiencies

Throughout the running of the LHC, the reinterpretation community have been approximating the outputs of complex multivariate algorithms, whose inputs are not available to them, using efficiencies. This includes processes such as jet flavour tagging, lepton identification, and isolation. It might be reasonable to ask if, as analyses use ML for increasingly complex tasks, it wouldn't be better to just keep using efficiencies, rather than directly reusing neural nets.

In some cases, the answer is yes – for example, Reference [225], an Atlas exotics search for long-lived particles, published both their neural net and a 6-dimensional parametrised efficiency map. The inputs to the neural net are specific to the hardware of the Atlas detector, and so are not accessible to those trying to reuse the analysis results: nevertheless, the efficiency means that useful results can be obtained.

However, in several situations this approach is likely to be less effective, or at least less effective without a great deal of additional care and effort.

Correlations and Region definitions

It is often unclear to reinterpreters within which region is the efficiency defined? Very often, if a paper refers to the efficiency of an ML-classifier, it will refer (sometimes only implicitly!) to its efficiency in a validation region used for network training, not inside the signal region itself. But if any of the other observables used to separate the training region from the signal regions has a strong correlation to the network output, then the efficiency will not be the same in the signal region. Often, networks will need to be applied in multiple

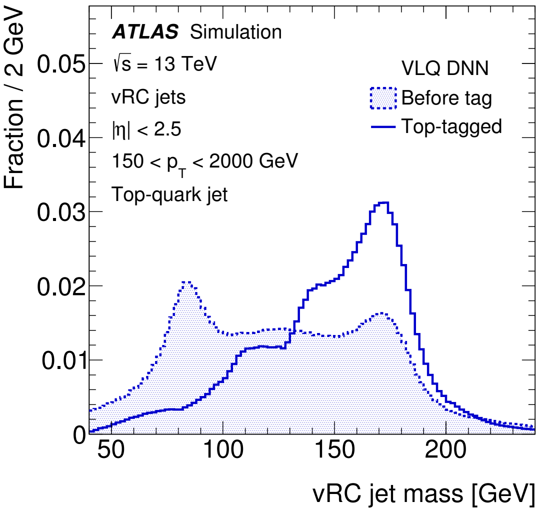


Figure 5.1: The jet mass distribution for "true" top-jets before and after the application of the MCBOT tagger. From Reference [224], Figure 4c.

[2] By "truth-level", we mean the hadronised and decayed output of a MCEG that has not undergone any detector simulation; as was represented by the middle box in Figure 3.1. Similarly, by "reco-level", we mean the reconstructed detector-level events, as represented by the fourth box in the same figure.

signal and control regions – whether or not a reliable definition of "efficiency" can be found is not necessarily a simple question.

As an example, consider Figure 5.1, taken from Reference [224]. This shows the distribution of the mass of "true" top-jets, before and after a neural network based cut has been applied, showing a clear bias towards high masses. If an additional cut on jet mass is applied, the stated efficiency in the training region clearly no longer applies.

These problems can be partially offset by the use of parametrised efficiencies – and for networks where inputs are hardware-dependent, very difficult to emulate or otherwise inaccessible, a highly parametrised efficiency map – for the moment – will remain the best option.

Truth-level ambiguities

When applying efficiencies to a b-jet or an electron, what constitutes a "true" b-jet or a "true" electron is well defined from the truth-level event record. This ceases to be the case for higher mass particles produced around resonances (such as the top-quark), where a lot of production may happen off-shell, and where different event generators will use different conventions about what a partonic top (for example) actually is. This may mean that these efficiencies become generator dependent, which would make them highly unreliable for reinterpretation.

Dependence on signal model

In the case of a ML model trained to discriminate signal from background – such as the ones covered in Sections 5.4 and 5.5 – efficiencies will only give the acceptances for the specific model being studied by the paper. Even if the analysis team provide extra information such as parametrising the efficiency by the mass of a BSM particle, we still have no information for what efficiencies to apply to other BSM particles, making the analysis near-useless for reinterpretation.

5.2 A brief survey of preservation methods

5.2.1 Pickle, `h5`, and other framework specific formats

Many of the common neural network training formats will include some sort of custom format for saving neural nets, often with the primary aim of stopping training so that it can be restarted again, or for allowing small-scale inference on a validation dataset. When training and inference are all carried out in Python, it is often also possible to store the neural network objects in Python `pickle` files. Proprietary NN software solutions (such as NeuroBayes [226], which has been used in published ATLAS studies [227]) often also use their own proprietary model file formats.

Almost without exception, these formats are a very poor choice for analysis preservation. Proprietary formats can only be used if other users have paid for access to the specific software. Pickled objects can be very sensitive to operating system, Python, and library versions, meaning that even simple pickled objects may be very difficult to read just a few years later. Framework specific formats may suffer from version inconsistency (particularly if default behaviour gets changed to match bleeding-edge industry standards), and most will eventually be deprecated in favour of newer, more efficient formats better suited to state-of-the-art architectures – for example, the recent deprecation of TensorFlow's "old" `.h5` format for storing weights [228]. If we want to create the best possible scientific legacy from the data we receive from the LHC, then we must ensure that we can rerun analyses for longer than $5-10$ years after publication.

Additionally, all these formats suffer from framework specificity, which in turn may also introduce language specificity. Most preservation codes use `C++`, while most ML training takes place in Python.

5.2.2 ONNX

`onnx` [229] is a binary file format designed to exchange NNs between different machine learning training and inference programs. For example, it allows a user to train a network using TensorFlow, and then evaluate it using Pytorch. Most ML training frameworks – including TensorFlow, Keras and Pytorch – now include functions to save most models in `onnx` format; though it should be noted that some of the most up-to-date generative model architectures are not yet supported.

The `onnx` project also maintains its own inference library, `ONNXRunTime`, which provides a method of evaluating `onnx` files from a variety of languages – API's exist for Python, `C/C++`, Julia, and others.

5.2.3 lwtnn

`lwtnn` was designed to allow networks trained in Python to be evaluated in `C++` – originally in the context of the Atlas trigger, though the code is now fully public. Compared to `onnx`, `lwtnn` is a smaller, lighter program with fewer dependencies, though the price for this is that it can handle input from a slightly smaller range of machine learning software, and has a more restricted set of possible network architectures. Unlike `onnx`, it is fully human-readable, using a JSON format.

Assuming all the layer types and activation functions are present in both `onnx` and `lwtnn`, it is possible – though strenuous – to convert an `onnx` file to `lwtnn`, using the `onnx2keras` [230] Python package as an intermediate step. Depending on the specifics of the ML model being converted, users may have to add a significant amount of boilerplate – such as dummy layers – during this conversion. Indeed, whilst trying to test this by converting the models described in Section 5.5, I had to go as far as contributing a new layer type to the `lwtnn` library [231]. There is currently no mechanism available for the reverse process (converting `lwtnn` to `onnx`), though it would surely be useful if developed.

5.3 Technical implementation

Once a network has been preserved and published in an easy-to-reuse format, the question turns to how inference using that model can be carried out within the reinterpretation frameworks. This requires the reinterpretation codes to have an interface to a project such as `lwtnn` or `onnxruntime`, ideally with some "syntactic sugar" to make use within analysis codes straightforward. In this section, we detail how this was achieved for Rivet and Gambit.

5.3.1 Rivet implementation

As part of the PhD, both `lwtnn` and `onnx` interfaces were added to Rivet. To minimise the number of additional dependencies for Rivet, while only a small number of analyses rely on these tools, each analysis must explicitly be linked against either `liblwtnn` or `libonnxruntime` when it is compiled. To facilitate this, the interface for each software framework consists of a single header file.

In keeping with Rivet's design principle that analyses should be clean, simple and easy to understand for those less familiar with `C++`, the analysis author will – in all normal cases – only need to write three neural net related lines in their code: calling `load` method in the analysis `init` function, calling the `compute` method in the `analyze` function, and declaring the neural network as a member object of the analysis.

In the case of lwtnn, we expose the native lwtnn neural network classes from the lwt namespace directly. Because of the additional complexity of initialisation – much of it just boilerplate – and the need to keep several variables alive for the whole analysis, the onnx interface uses its own RivetONNXrt wrapper class.

Correct usage of both interfaces is illustrated in the EXAMPLE_LWTNN and EXAMPLE_ONNX analyses in the Rivet analysis examples directory. The lwtnn interface has been available since Rivet 3.1.6 (though extended to more complex network structures in 3.1.7), and the onnx interface since 3.1.7 (with important bug-fixes in 3.1.8). Additional work meant that from version 4.0.0, Rivet could be configured to link against libonnxruntime automatically, which would trigger automated compilation of onnx dependant analyses – for now stored in their own pluginONNX directory – at compile time. The first analysis in the pluginONNX directory was the implementation of Reference [177] described in Section 5.5. In order to avoid bundling large, rarely-used onnx files into the Rivet repository, configuring rivet with the --enable-onnxrt would also trigger download of all the onnx files needed for the analyses in pluginONNX at build time.

5.3.2 Gambit implementation

Due to the structure of ColliderBit, Gambit's usual system of backending external codes so that they are not compile-time dependencies – as will be done for Rivet and Contur in Section 7.2 – does not work for libraries that need to be directly accessed during analysis execution. Therefore, in order to run analyses that depend on onnx, onnx needed to be made a compile time dependency inside Gambit's core (similar to the treatment of Yoda). Longer-term, the Gambit core developers are hoping that backends will be accessible from within analyses.

Similarly to the Rivet implementation, a wrapper class that contains all the necessary metadata and long-lived objects has been defined, and can now be accessed from any future analyses that depend on it.

5.4 Reinterpreting Atlas SUSY searches: A search for R-parity-violating supersymmetry in a final state containing leptons and many jets (SUSY-2019-04)

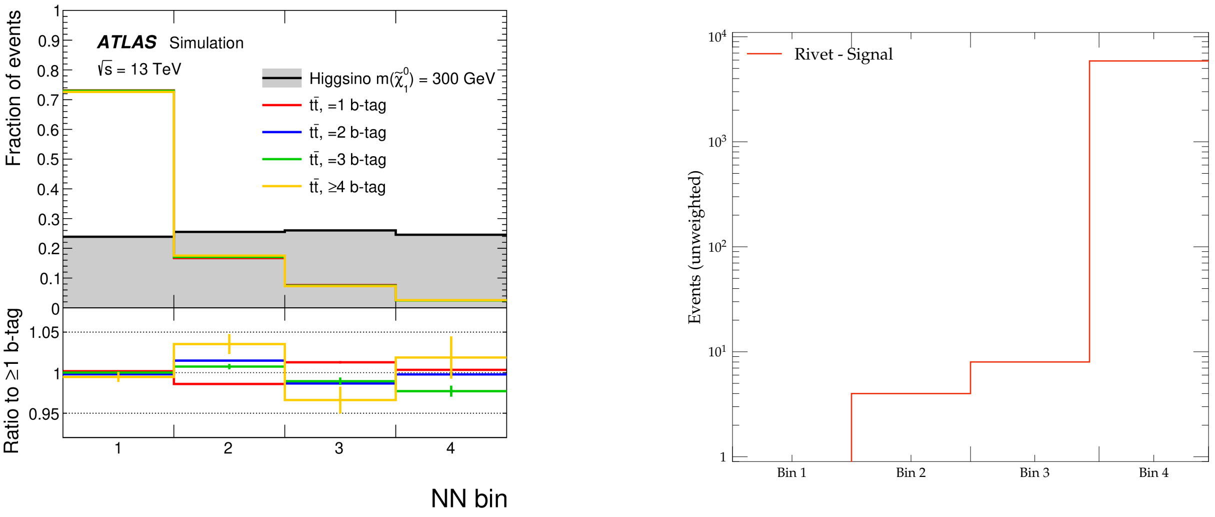
The Atlas SUSY group has led the way in publishing material required for reinterpreting analyses that depend on ML. Amongst the LHC experiments, both the first public Boosted Decision Tree (BDT) [232], and the first two public NN models [177,233] have come from this group.

This section concerns the first of these searches, for R-parity violating supersymmetry in final states containing a high jet-multiplicity, at least one lepton and either no or three b-jets [177]. This was the first Atlas or CMS analysis to provide their neural net on HEPData. The NN takes as inputs the kinematics of the leading ten jets, the kinematics of the leading lepton, and some event-level variables such as H_T[3]; and returns a straightforward signal vs background classification score, with a cut at approximately 0.75. As with many analyses from the Atlas SUSY group, writing a reinterpretation (in this case a Rivet analysis) was significantly aided by the analysis team providing a SimpleAnalysis code on HEPData. Also useful was the provision of several slha files, describing the SUSY models used in each cutflow table.

Unfortunately, it proved very difficult to replicate the results of the analysis. Figure 5.2 shows that the NN score was expected to have an approximately flat distribution for the signal model, but we obtained a much more peaked distribution. This suggests that in the Rivet context, the model was overwhelmingly more succesful at separating signal from background than it should have been[4]. One obvious explanation

[3] H_T is defined as the scalar sum of all the individual contribution to the E_T^{miss} hard-term.

[4] Although to state this with full confidence, we would also need to run on the background Monte-Carlo.



(a)

(b)

Figure 5.2: Comparing the expected output shape of the NN distribution published by the analysis (left, adapted from Reference [233], Figure 2) and that obtained in the RIVET reinterpretation (right). It is possible that differences in preceding cuts and Monte-Carlo generation account for some of the discrepancy. Note the logarithmic axis on the right-hand-side.

is that detector emulation in RIVET – using the standard RIVET detector-emulation functions – didn't adequately "confuse" the signal; although with the myriad other problems encountered in this analysis, and some ambiguity over the preceding cuts and the exact MC generator configuration used for this plot, that cannot be said with any certainty.

Tables 5.1 and 5.2 compare the cutflows obtained in this analysis for different RPV SUSY models, against the cutflows provided by the analysis-team. Both show significant discrepancies. Similar experiences were reported by other reinterpretation tools at the 7th general workshop of the LHC Reinterpretation Forum [234], including CheckMATE and MadAnalysis.

It is worth noting that the large discrepancies in Tables 5.1 and 5.2 occur several cuts before the NN is applied; and furthermore there are also slightly weaker discrepancies in the cutflow for a pure "cut-and-count" analysis region, as shown in Table 5.3. So the failure to reproduce the results of this analysis is not an indication that providing neural nets is unlikely to work; rather the most likely explanation for the failure is that, like so many conventional cut and count analyses before it, this analysis merely does not provide enough information to reproduce its kinematic distributions correctly.

Jet p_{T} threshold	20 GeV		40 GeV		60 GeV		80 GeV		100 GeV	
$1l$ channel	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet
≥ 4 jets	20.6%	9.9%	12.2%	6.4%	5.1%	3.2%	2.0%	1.3%	0.9%	0.62%
$= 4$ jets	4.6%	1.6%	6.7%	2.8%	3.5%	2.0%	2.0%	1.1%	0.9%	0.53%
$= 5$ jets	6.4%	2.7%	3.8%	2.3%	1.2%	0.91%	1.6%	0.2%	0.7%	0.07%
$= 6$ jets	5.2%	2.7%	1.3%	1.7%	0.3%	0.15%	0.4%	0.02%	0.1%	0.02%
$= 7$ jets	2.8%	1.9%	0.4%	0.22%	0.1%	0.07%	0.1%	0.01%	0.0%	0.0%
$= 8$ jets	1.1%	0.98%	0.1%	0.04%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
$= 4$ jets, ≥ 4 b-tags	0.06%	0.07%	0.09%	0.10%	0.04%	0.07%	0.02%	0.09%	0.01%	0.04%
$= 5$ jets, ≥ 4 b-tags	0.32%	0.41%	0.20%	0.28%	0.06%	0.15%	0.02%	0.02%	0.01%	0.0%
$= 6$ jets, ≥ 4 b-tags	0.43%	0.73%	0.11%	0.26%	0.02%	0.02%	0.00%	0.01%	0.00%	0.0%
$= 7$ jets, ≥ 4 b-tags	0.30%	0.54%	0.04%	0.07%	0.00%	0.04%	0.00%	0.01%	0.00%	0.0%
$= 8$ jets, ≥ 4 b-tags	0.13%	0.36%	0.01%	0.03%	0.00%	0.0%	0.00%	0.0%	0.00%	0.0%
$= 4$ jets, ≥ 4 b-tags, NN	0.02%	0.07%	-	-	-	-	-	-	-	-
$= 5$ jets, ≥ 4 b-tags, NN	0.09%	0.41%	-	-	-	-	-	-	-	-
$= 6$ jets, ≥ 4 b-tags, NN	0.11%	0.73%	-	-	-	-	-	-	-	-
$= 7$ jets, ≥ 4 b-tags, NN	0.07%	0.53%	-	-	-	-	-	-	-	-
$= 8$ jets, ≥ 4 b-tags, NN	0.03%	0.0%	-	-	-	-	-	-	-	-

Table 5.1: Cutflow comparison to Reference [233]; using data provided by auxiliary table 9, for $\tilde{\chi}_1^{\pm}\tilde{\chi}_1^0$ production in an RPV SUSY model. To minimise discrepancies due to Monte-Carlo, the data was generated using an LHE file generated by Atlas's `generate_tf` given the same DSID code as used by the original analysis. Agreement is poor before and after the application of the NN-based cut.

Jet p_{T} threshold	20 GeV		40 GeV		60 GeV		80 GeV		100 GeV	
$1l$ channel	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet
≥ 4 jets	30.8%	17.6%	19.7%	12.3%	9.0%	6.2%	3.8%	2.8%	1.7%	1.3%
$= 4$ jets	4.7%	1.7%	9.2%	5.1%	5.8%	3.9%	2.8%	2.1%	1.3%	1.06%
$= 5$ jets	7.6%	3.8%	6.4%	4.4%	2.4%	1.7%	0.8%	0.57%	0.3%	0.19%
$= 6$ jets	8.2%	4.9%	2.9%	2.1%	0.7%	0.57%	0.1%	0.10%	0.00%	0.03%
$= 7$ jets	5.8%	4.3%	0.9%	0.65%	0.1%	0.06%	0.00%	0.01%	0.00%	0.00%
$= 8$ jets	2.9%	2.9%	0.2%	0.06%	0.0%	0.01%	0.00%	0.00%	0.00%	0.00%
$= 4$ jets, ≥ 4 b-tags	0.05%	0.10%	0.08%	0.17%	0.05%	0.15%	0.02%	0.07%	0.01%	0.05%
$= 5$ jets, ≥ 4 b-tags	0.20%	0.49%	0.17%	0.56%	0.06%	0.25%	0.02%	0.09%	0.01%	0.03%
$= 6$ jets, ≥ 4 b-tags	0.39%	1.1%	0.15%	0.37%	0.03%	0.09%	0.01%	0.02%	0.00%	0.00%
$= 7$ jets, ≥ 4 b-tags	0.42%	1.2%	0.06%	0.23%	0.01%	0.02%	0.01%	0.00%	0.00%	0.00%
$= 8$ jets, ≥ 4 b-tags	0.27%	1.1%	0.02%	0.02%	0.00%	0.01%	0.00%	0.00%	0.00%	0.00%
$= 4$ jets, ≥ 4 b-tags, NN	0.02%	0.1%	-	-	-	-	-	-	-	-
$= 5$ jets, ≥ 4 b-tags, NN	0.06%	0.49%	-	-	-	-	-	-	-	-
$= 6$ jets, ≥ 4 b-tags, NN	0.12%	1.0%	-	-	-	-	-	-	-	-
$= 7$ jets, ≥ 4 b-tags, NN	0.13%	1.2%	-	-	-	-	-	-	-	-
$= 8$ jets, ≥ 4 b-tags, NN	0.10%	0.00%	-	-	-	-	-	-	-	-

Table 5.2: Cutflow comparison to Reference [233]; using data provided by auxiliary table 10, for $\tilde{\chi}_1^0\tilde{\chi}_2^0$ production in an RPV SUSY model. To minimise discrepancies due to Monte-Carlo, the data was generated using an LHE file generated by Atlas's `generate_tf` given the same DSID code as used by the original analysis. Agreement is poor before and after the application of the NN-based cut.

Jet p_T threshold	20 GeV		40 GeV		60 GeV		80 GeV		100 GeV	
$1l$ channel	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet
$\geq$ 6 jets	46.7%	53.4%	46.4%	55.0%	45.9%	54.2%	44.8%	52.9%	42.9%	50.5%
$\geq$ 8 jets	44.1%	52.4%	38.9%	45.7%	32.7%	37.8%	26.4%	29.0%	20.3%	21.3%
$\geq$ 10 jets	30.5%	35.7%	16.7%	17.8%	9.4%	8.8%	4.9%	4.15%	-	-
$\geq$ 11 jets	21.5%	23.9%	8.3%	8.1%	3.7%	2.8%	-	-	-	-
$\geq$ 12 jets	13.1%	14.0%	3.3%	2.9%	-	-	-	-	-	-
$\geq$ 15 jets	1.4%	1.3%	-	-	-	-	-	-	-	-
$\geq$ 6 jets (0 b-tags)	38.2%	47.2%	39.1%	44.9%	39.2%	42.7%	38.7%	40.8%	37.3%	38.2%
$\geq$ 8 jets (0 b-tags)	35.9%	45.1%	32.3%	38.2%	27.5%	30.6%	22.4%	23.8%	17.3%	17.4%
$\geq$ 10 jets (0 b-tags)	24.4%	31.6%	13.5%	15.6%	7.6%	7.9%	4.0%	3.7%	-	-
$\geq$ 11 jets (0 b-tags)	17.0%	21.6%	6.6%	7.3%	2.7%	2.5%	-	-	-	-
$\geq$ 12 jets (0 b-tags)	10.2%	12.9%	2.5%	2.7%	-	-	-	-	-	-
$\geq$ 15 jets (0 b-tags)	1.0%	1.2%	-	-	-	-	-	-	-	-
$2l$ same-charge channel	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet	Atlas	Rivet
$\geq$ 6 jets	6.9%	8.2%	6.8%	8.1%	6.8%	8.0%	6.7%	7.9%	6.4%	7.5%
$\geq$ 7 jets	6.9%	8.1%	6.6%	7.8%	6.2%	7.3%	5.7%	6.5%	-	-
$\geq$ 8 jets	6.5%	7.7%	5.7%	6.6%	-	-	-	-	-	-
$\geq$ 10 jets	4.5%	5.25%	-	-	-	-	-	-	-	-
$\geq$ 6 jets (0 b-tags)	5.7%	7.0%	5.7%	6.5%	5.8%	6.3%	5.7%	6.0%	5.5%	5.6%
$\geq$ 7 jets (0 b-tags)	5.6%	7.0%	5.5%	6.3%	5.3%	5.8%	4.9%	5.1%	-	-
$\geq$ 8 jets (0 b-tags)	5.3%	6.6%	4.8%	5.5%	-	-	-	-	-	-
$\geq$ 10 jets (0 b-tags)	3.6%	4.6%	-	-	-	-	-	-	-	-

Table 5.3: Cutflow comparison to Reference [233]; using data provided by auxiliary table 5, for an RPV SUSY model, for the shape analysis. These signal did not use a NN-based cut, yet agreement is still poor. To minimise discrepancies due to Monte-Carlo, the data was generated using an LHE file generated by Atlas's `generate_tf` given the same DSID code as used by the original analysis.

5.5 Reinterpreting Atlas SUSY searches: A search for supersymmetry in final states with missing transverse momentum and three or more *b*-jets (SUSY-2018-30)

Another analysis that made a neural network public was Reference [177]; a search for pair-produced gluinos decaying via third-generation squarks to final states with many b-jets and neutralinos. The models considered were classified as either “Gbb”, “Gtt”, or “Gtb”, depending on which SM quarks are produced in the decay of the gluino to the neutralino. The Feynman diagrams for the Gbb and Gtt decays are shown in Figure 5.3. It is (obviously) kinematically impossible for the neutralino to be more massive than the gluino it decayed from, and the region of parameter space where the neutralino mass is closest to the gluino mass is known as the “compressed” region. Here, the SM decay products are produced at lower p_T and are less likely to be collimated, as a result of which the final state is more complex. This makes both analysis and event generation more difficult – for example, additional gluon emissions (and therefore the choice of matching or merging scheme) become increasingly significant.

Each decay mode had its own set of specialised search regions. There were cut-and-count regions for all three decay modes; but the Gtt and Gbb models also had four NN-dependent regions each. These were defined by some relatively simple kinematic cuts, combined with a cut on a DNN classifier discriminant, trained to separate different signal models from different sources of background. The network inputs mainly consisted of object kinematics (small- and large-radius jets, leptons, E_T^{miss}), as well as some higher level observables (effective masses, etc.), and model parameters (Gbb vs Gtt, gluino and neutralino masses).

The exclusion limits obtained using the NN-based regions are significantly stronger – Figure 5.4 shows that the observed limit for the Gbb model is extended by approximately 200 – 300 GeV in both gluino and neutralino mass – so being able to reinterpret these regions would clearly be beneficial. The same figure also shows that the Contur calculated SM measurement exclusion of the model targetted by this search is very weak – likely due to a lack of measurements in the three- or four- b-jets + E_T^{miss} phase space – a fact that provides further motivation for succesfully reinterpreting this analysis.

In this instance, the network was not put on `HEPData`, but rather stored in a publicly accessible section of the SimpleAnalysis repository. SUSY searches form Gambit’s bread and butter, so in addition to the Rivet analysis, I also implemented the analysis in ColliderBit.

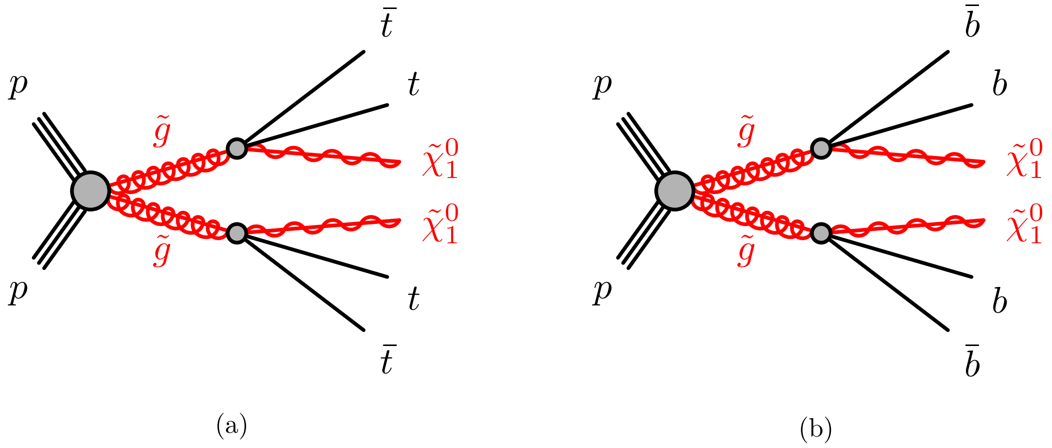


Figure 5.3: Feynman diagrams for the Gtt (a) and Gbb (b) processes studied in Reference [177]. Adapted from the same source.

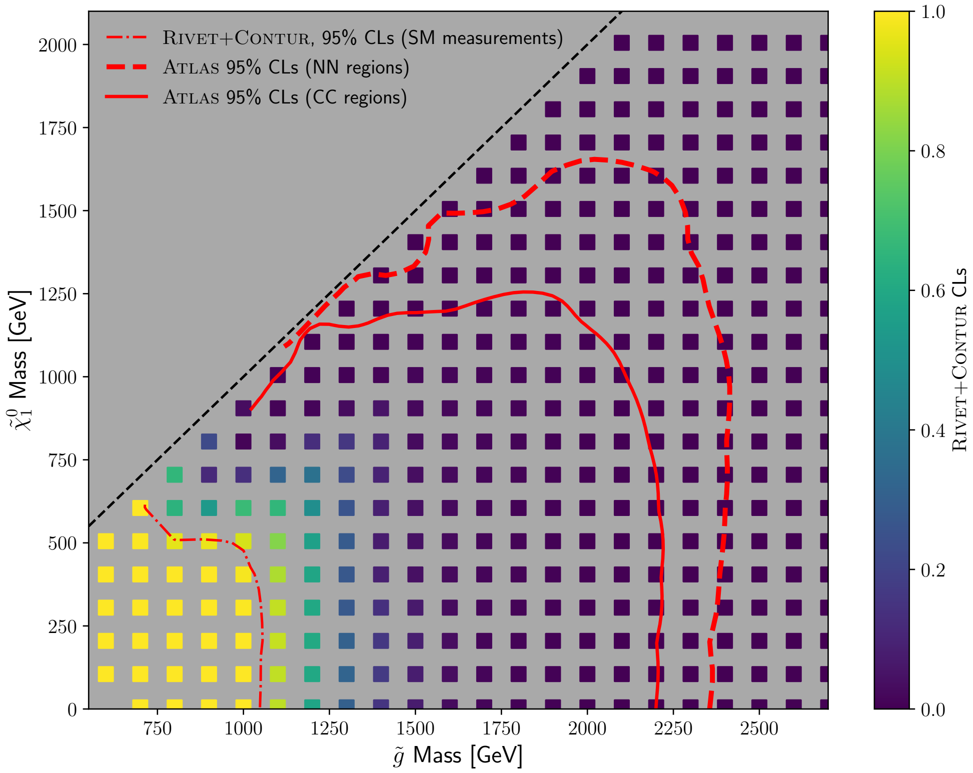


Figure 5.4: Exclusion curves for the Gbb model from Reference [177] for its neural net and cut-and-count (CC) based regions; alongside the much weaker exclusion curve produced when constraining the model using SM measurements in Contur. The Contur exclusion is dominated by the 3.2 fb^{-1} Atlas inclusive jet and dijet cross-section measurement [235], so it is possible the exclusion will be extended slightly if a full Run 2 version is published.

5.5.1 Reinterpretation challenges

Writing a successful reinterpretation required some specialisation beyond the normal detector-emulation tools previously available in Rivet. The most significant of these was in the application of b-tagging efficiency. Typically, working-points like Atlas's MV2c10 [119] are applied using a single b-tagging efficiency number to all jets (and c, τ, and light jet misidentification probabilities where appropriate). However, due to the high b-jet multiplicity of this analysis, it is very sensitive to mismodelling. Combined with the hardness of the leading b-jets predicted by the signal model, a more detailed description that understands that the efficiency is not constant in p_T, as illustrated in Figure 5.5, may give much better results.

Therefore, b-tagging emulation for this analysis was carried out used the binned efficiency taken from Figure 5.5. Unfortunately, this plot does not extend beyond 500 GeV, and a good estimate of a single overflow value is not obvious. A range of "reasonable" estimates is illustrated in Figure 5.6; and the impact on the yield of varying within this range was found to be up to $\mathcal{O}(10\%)$. Finally, an efficiency of 0.69 was chosen as a good compromise, though both this analysis and the reinterpretation of multi-b-jet searches in general would greatly benefit from the Atlas collaboration providing more detailed efficiency plots that cover the full range of interesting b-jet momenta.

Other similar changes included an additional filtering of jets with transverse momentum below 60 GeV to

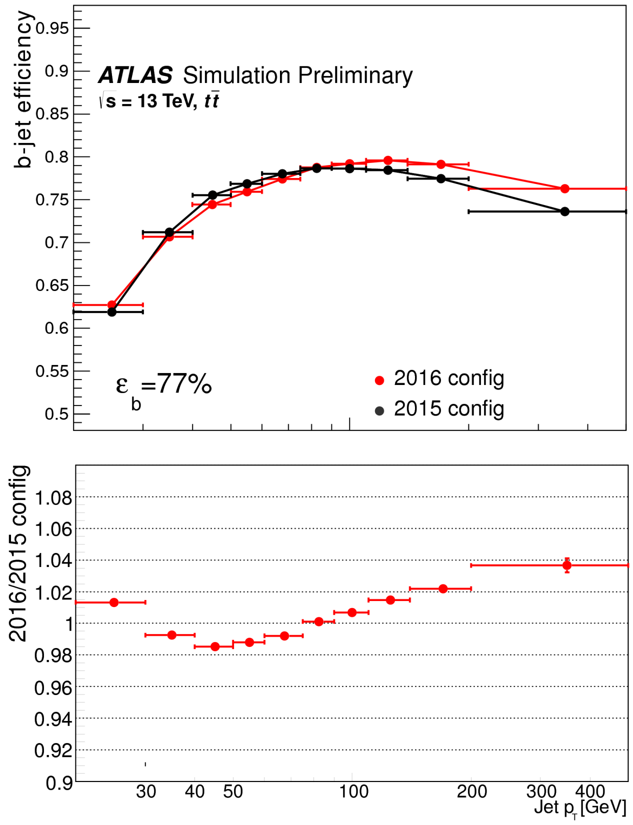


Figure 5.5: The ATLAS MV2c10 efficiency at the 77% working point, taken from Reference [119], Figure 12.

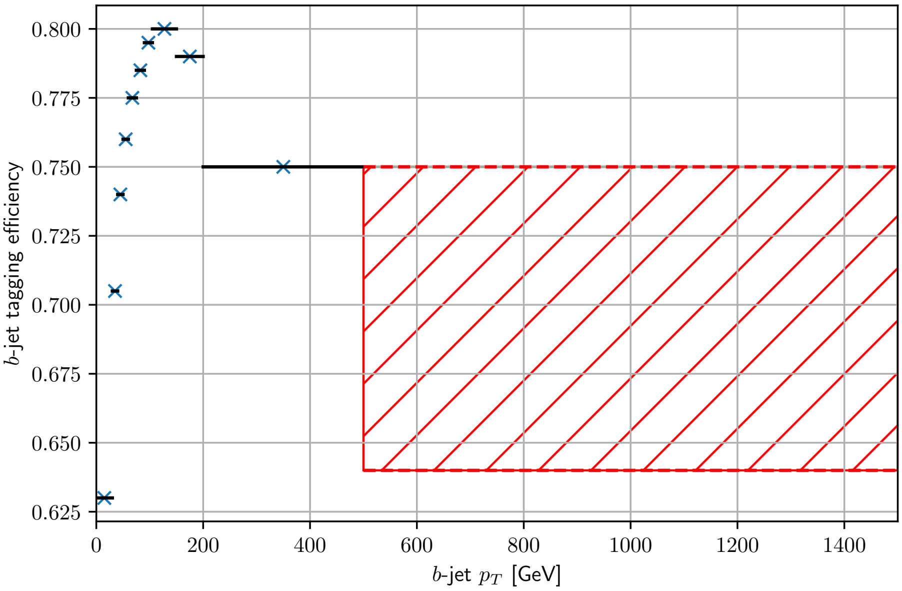


Figure 5.6: The ATLAS MV2c10 efficiency at the 77% working point, adapted from Reference [119], Figure 12 (reproduced here in Figure 5.5); the x-axis has been made linear in order to make visualising an extrapolation easier. The range of possible overflow-bin efficiency values considered is illustrated by the red-hatched area. Reference [119] explicitly states (though sadly does not show) that efficiency does fall slightly beyond 500 GeV, at least up to 2 TeV.

emulate “real” jets incorrectly removed by ATLAS’s Jet Vertex Tagger (JVT), as well as a customised logic for applying electron efficiencies in order to ensure a consistent set of both tight and loose tagged electrons. This analysis is only weakly dependent on soft-jets, but for analyses where they are more significant, JVT emulation is a challenge for reinterpreters, not least because the most recent public JVT efficiency plots date back to LHC Run 1 [236]. These JVT emulation functions were added to the RIVET library of smearing and efficiency functions, where they will be available to future RIVET analyses. Unifying the logic for consistent particle efficiency definitions into a single projection would probably also be useful further work.

Comparisons between the RIVET and GAMBIT implementations led to the discovery – and subsequent fixing – of bugs in the RIVET `LeptonFinder` Projection and in the RIVET emulation of ATLAS Run 2 electron identification efficiencies. More fine-grained muon efficiency maps were also added to COLLIDERBIT.

5.5.2 Cutflow results

Tables 5.4 – 5.8 compare the cutflow results from the original paper to those obtained by the RIVET and GAMBIT analyses, for both NN and cut-and-count signal regions. Event generation was carried out using PYTHIA 8.307 and the SLHA files provided by the analysis team on `HEPData`. The final counts from the NN signal regions are illustated in Figure 5.7. Errors are consistently below 15% which, for a re-interpretation using smearing-based detector-emulation, is very acceptable. Notably, the ratio columns in the tables show

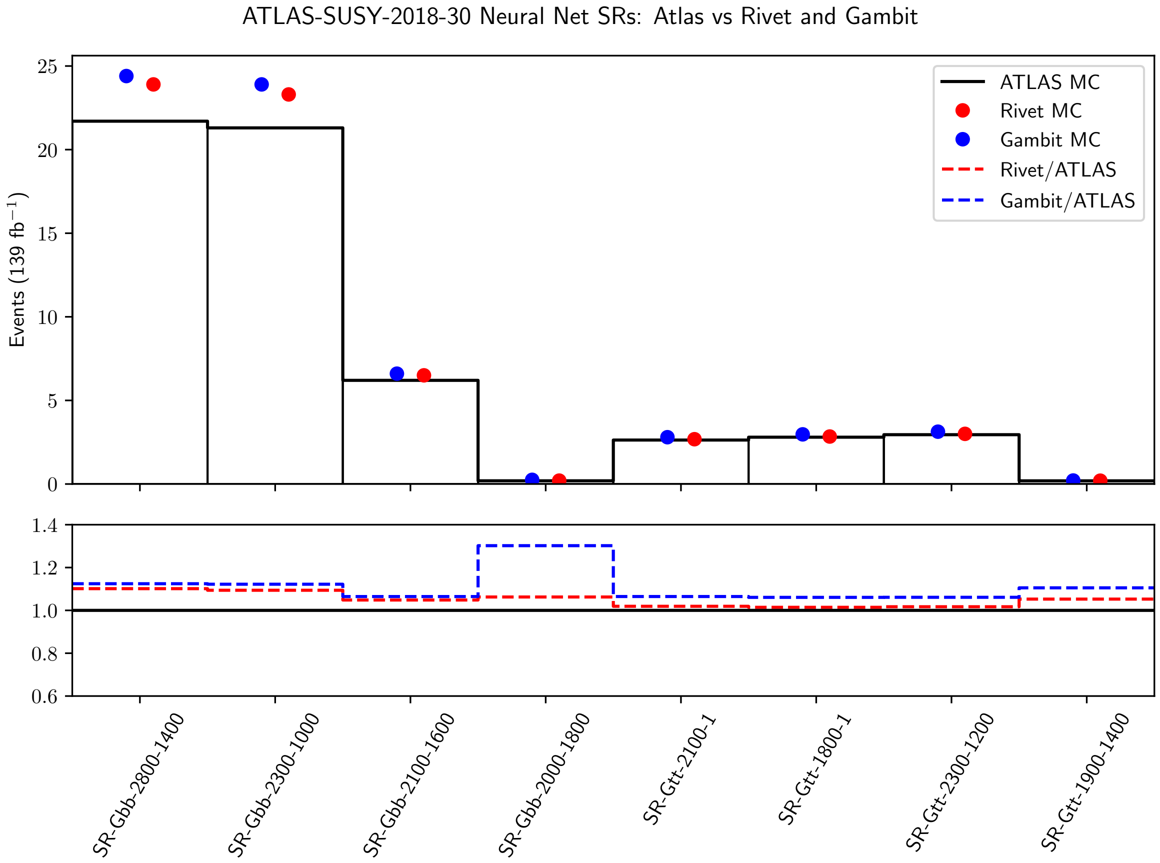


Figure 5.7: Final event counts at 139 fb^{-1} obtained by RIVET and GAMBIT for the eight NN-defined signal regions in Reference [177]. The signal model was the benchmark Gbb model for the Gbb regions and the benchmark Gtt model for the Gtt regions; both benchmark models were generated using the provided slha files. Errors of 15% or less are very satisfactory for detector-emulation-based reinterpretation.

Region	Selection	Event yield (139fb^{-1})		% of previous yield	
		Paper (Atlas)	Rivet	Paper (Atlas) %	Rivet %
SR-Gtt-0L-B	$N_{\text{jet}} \geq 5$	3.92	4.8*	-	-
	$E_T^{\text{miss}} \geq 600$	2.60	3.2	66%	67%*
	$m_{\text{eff}} \geq 2900$	1.17	1.4	45%	45%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	1.07	1.3	91%	92%
	$M_J^{\Sigma} \geq 300$	0.997	1.25	93%	94%
SR-Gtt-0L-B	$N_{\text{jet}} \geq 9$ & $N_{b\text{-jets}} \geq 3$	2.34	2.6*	-	-
	$E_T^{\text{miss}} \geq 600$	1.53	1.7	65%	65%*
	$m_{\text{eff}} \geq 1700$	1.53	1.7	100%	100%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	1.36	1.6	89%	90%
	$M_J^{\Sigma} \geq 300$	1.17	1.34	86%	87%
SR-Gtt-0L-B	$N_{\text{jet}} \geq 10$ & $N_{b\text{-jets}} \geq 3$	1.49	1.7*	-	-
	$E_T^{\text{miss}} \geq 500$	1.13	1.3	76%	76%*
	$m_{\text{eff}} \geq 1100$	1.13	1.3	100%	100%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	0.990	1.1	88%	89%
	$M_J^{\Sigma} \geq 300$	0.973	1.13	98%	98%
SR-Gtt-0L-B	$N_{\text{jet}} \geq 10$	1.49	1.7*	-	-
	$N_{b\text{-jets}} \geq 4$	0.722	0.9*	48%	51%*
	$E_T^{\text{miss}} \geq 500$	0.638	0.8	88%	87%*
	$m_{\text{eff}} \geq 1100$	0.638	0.8	100%	100%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	0.526	0.65	82%	86%
	$M_J^{\Sigma} \geq 300$	0.526	0.65	100%	100%

Table 5.4: Cutflow comparison to Reference [177], auxiliary Table 2. N.B. due to the analysis structure in Rivet, the N_{jet} and E_T^{miss} cuts cannot be fully disentangled, so the Rivet entries marked with a * cannot be considered true "apples-to-apples" comparison. Note the excellent performance of the ratios in the final two columns.

that deviations are consistent across all cuts, suggesting that even the small remaining errors may either come from cross-section or normalisation differences, or else some discrepancy in the understanding of pre-selection cuts. The consistent slight undercounts in the 1-lepton cut-and-count regions compared to the consistent slight overcounts in the 0-lepton cut-and-count regions may indicate that Rivet's detector emulation and identification/isolation efficiencies do not produce enough leptons: perhaps because we do not currently have the capability to misidentify jets as leptons (although we do implement the inverse).

Unfortunately, the exceptions to this general positivity were the three regions (of 20 total) designed to probe the "Gtb" model, shown in Table 5.7. In each case, while the ratios between different cuts was broadly consistent with the provided cutflows, there was an overall normalisation factor discrepancy of about 1.8. This suggests that there may be a pre-selection cut missing, or else that there was some problem with the Monte-Carlo or cross-section estimate: these three regions' cutflows all use the same `slha` file that wasn't used anywhere else. Notably, these three regions do not use any NN based cuts, showing that relying on a neural needn't make the analysis unreinterpretable – indeed, **the greatest threat to reinterpretability comes not from using ML, but from insufficient or inaccurate documentation.**

Region	Selection	Event yield (139fb^{-1})		% of previous yield	
		Paper (Atlas)	Rivet	Paper (Atlas) %	Rivet %
	$N_{\text{jet}} \geq 4$ & $N_{b\text{-jets}} \geq 3$	4.34	4.0*	-	-
	$E_T^{\text{miss}} \geq 600$	2.55	2.3	59%	59%*
SR-Gtt-1L-B	$m_{\text{eff}} \geq 600$	1.98	1.8	78%	76%
	$m_T \geq 150$	1.73	1.5	87%	85%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	1.47	1.3	85%	87%
	$M_J^{\Sigma} \geq 200$	1.36	1.21	93%	92%
	$N_{\text{jet}} \geq 5$ & $N_{b\text{-jets}} \geq 3$	4.26	3.9*	-	-
	$E_T^{\text{miss}} \geq 600$	2.51	2.3	59%	59%*
SR-Gtt-1L-B	$m_{\text{eff}} \geq 2000$	2.32	2.1	92%	92%
	$m_T \geq 150$	1.93	1.7	83%	79%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	1.65	1.4	86%	87%
	$M_J^{\Sigma} \geq 200$	1.47	1.29	89%	89%
	$N_{\text{jet}} \geq 8$ & $N_{b\text{-jets}} \geq 3$	2.34	2.0*	-	-
	$E_T^{\text{miss}} \geq 500$	1.67	1.4	71%	70%*
SR-Gtt-1L-B	$m_{\text{eff}} \geq 1100$	1.67	1.4	100%	100%
	$m_T \geq 150$	1.37	1.1	82%	77%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	1.15	0.9	84%	85%
	$M_J^{\Sigma} \geq 200$	1.14	0.89	99%	99%
	$N_{\text{jet}} \geq 9$ & $N_{b\text{-jets}} \geq 3$	1.44	1.2*	-	-
	$E_T^{\text{miss}} \geq 300$	1.32	1.1	92%	92%*
SR-Gtt-1L-B	$m_{\text{eff}} \geq 800$	1.32	0.8	100%	100%
	$m_T \geq 150$	1.13	0.8	86%	79%
	$m_{\text{T,min}}^{b\text{-jets}} \geq 120$	0.906	0.69	80%	81%

Table 5.5: Cutflow comparison to Reference [177], auxiliary Table 3. N.B. due to the analysis structure in Rivet, the N_{jet} and E_T^{miss} cuts cannot be fully disentangled, so the Rivet entries marked with a * cannot be considered true “apples-to-apples” comparison. Note the excellent performance of the ratios in the final two columns.

Region	Selection	Event yield (139fb^{-1})		% of previous yield	
		Paper (Atlas)	Rivet	Paper (Atlas) %	Rivet %
Common	$N_{\text{lep,base}} = 0$	80.0	83.7*	-	-
	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	$m^{b\text{-jets}}_{\text{T,min}} \geq 130$	56.6	59.4	93%	93%
SR-Gbb-B	$E_T^{\text{miss}} \geq 550$	35.69	37.5	63%	63%*
	$p_T^{\text{jet}} \geq 65$	35.69	37.5	100%	100%*
	$m_{\text{eff}} \geq 2600$	10.13	12.5	28%	33%
SR-Gbb-M	$E_T^{\text{miss}} \geq 550$	35.69	37.5	63%	63%*
	$m_{\text{eff}} \geq 2000$	28.30	30.2	79%	80%
SR-Gbb-C	$E_T^{\text{miss}} \geq 550$ †	35.69	37.5	63%	63%*
	$m_{\text{eff}} \geq 1600$ †	34.71	36.6	97%	97%

† N.B. these two cuts appear the other way round in the original copy of the cutflow, but consistency with other regions, as well as the actual results, suggests this was a typo.

Table 5.6: Cutflow comparison to Reference [177], auxiliary Table 4. N.B. due to the analysis structure in Rivet, the $N_{\text{lep, base}}$ and $\Delta\phi^{4j}_{\text{min}}$ cuts cannot be fully disentangled, so the Rivet entries marked with a * cannot be considered true "apples-to-apples" comparison. Note the excellent performance of the ratios in the final two columns.

Region	Selection	Event yield (139fb^{-1})		% of previous yield	
		Paper (Atlas)	Rivet	Paper (Atlas) %	Rivet %
Common	$N_{\text{lep,base}} = 0$	15.93	23.3	-	-
	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	12.08	17.4	76%	75%
	$m^{b\text{-jets}}_{\text{T,min}} \geq 130$	10.95	16.1	91%	92%
SR-Gtb-B	$m_{\text{eff}} \geq 2600$	7.96	11.9	73%	74%
	$E_T^{\text{miss}} \geq 550$	6.81	10.1	86%	85%
	$M_J^{\Sigma} \geq 200$	6.13	9.2	90%	91%
SR-Gtb-M	$N_{\text{jet}} \geq 6$	9.54	13.6	87%	85%
	$N_{b\text{-jets}} \geq 4$	3.82	6.3	39%	46%
	$m_{\text{eff}} \geq 2000$	3.52	5.8	92%	93%
	$E_T^{\text{miss}} \geq 550$	2.76	4.5	78%	77%
	$M_J^{\Sigma} \geq 200$	2.50	4.1	91%	91%
SR-Gtb-C	$N_{\text{jet}} \geq 7$	7.55	10.5	69%	65%
	$N_{b\text{-jets}} \geq 4$	3.18	5.0	42%	48%
	$m_{\text{eff}} \geq 1300$	3.17	5.0	100%	100%
	$E_T^{\text{miss}} \geq 500$	2.52	3.9	79%	78%
	$M_J^{\Sigma} \geq 50$	2.52	3.9	100%	100%

Table 5.7: Cutflow comparison to Reference [177], auxiliary Table 5. Note the excellent performance of the ratios in the final two columns.

Region	Selection	Event yield (139fb^{-1})		% of previous yield	
		Paper (Atlas)	Rivet	Paper (Atlas) %	Rivet %
Common Gbb	$N_{\text{lep,base}} = 0$	80.0	83.7*	-	-
SR-Gbb-2800-1400	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) $\geq$ 0.999	21.7	23.9	41%	44%
SR-Gbb-2300-1000	$\Delta\phi^{4j}_{\text{min}} \geq 0.6$	52.5	54.6	66%	65%*
	P(Gbb) $\geq$ 0.9994	21.3	23.3	41%	43%
SR-Gbb-2100-1600	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) $\geq$ 0.9993	6.20	6.50	10%	10%
SR-Gbb-2000-1800	$\Delta\phi^{4j}_{\text{min}} \geq 0.4$	61.1	63.8	76%	76%*
	P(Gbb) $\geq$ 0.997	0.192	0.204	3.1%	3.2%
Common Gtt	$N_{\text{lep,sig}} = 1$ or ($N_{\text{lep,base}} =$ 1 and $\Delta\phi^{4j}_{\text{min}} \geq 0.4$)	7.66	8.1*	-	-
SR-Gtt-2100-1	P(Gtt) $\geq$ 0.9997	2.63	2.68	34%	33%
SR-Gtt-2100-1	P(Gtt) $\geq$ 0.9997	2.80	2.84	37%	35%
SR-Gtt-2100-1	P(Gtt) $\geq$ 0.9997	2.95	3.00	39%	37%
SR-Gtt-2100-1	P(Gtt) $\geq$ 0.9997	0.19	0.20	2.5%	2.5%

Table 5.8: Cutflow comparison to Reference [177], auxiliary Tables 6 and 7. N.B. due to the analysis structure in Rivet, the $N_{\text{lep, base}}$ and $\Delta\phi^{4j}_{\text{min}}$ cuts cannot be fully disentangled, so the Rivet entries marked with a * cannot be considered true “apples-to-apples” comparison. The P(Gbb)) and P(Gtt)) cuts are the NN-based cuts, which perform as well as any other cut in this analysis. Note particularly the excellent performance of the ratios in the final two columns.

5.5.3 Exclusion contour study

We can further extend the validation by trying to reproduce the exclusion contours provided by the experiment. To do this, we needed to generate a large number of events – eventually 100 000 were used – at many points across the parameter grid. The original analysis used 169 points, sampling (approximately) every 200 GeV in gluino mass and every 100 GeV in neutralino mass. At this scale, event generation “by-hand” – a custom-written Pythia main function importing the SLHA file provided by the analysis team on `HEPData`, being executed manually to produce HepMC output at each parameter point – would be completely impractical.

Instead, we built a custom workflow to generate events for each parameter point in parallel using HTCondor. This was controlled by a Python script which would, at each parameter point, make the necessary edits to the SLHA file, and then run a customised Pythia executable using this SLHA file. Using Pythia also allowed us to use Pythia’s Rivet interface, and run the analysis on in-memory HepMC events on the fly, rather than having to save 50 000 HepMC events at every one of at least 169 parameter points. Instead, we only had to write a Yoda file at each parameter point – a filetype whose size is effectively independent of the number of events it contains. This not only saves disk-space, but also the large amount of CPU usage that would otherwise have been spent on disk I/O. The hard-process was defined by the Pythia `SUSY:gg2gluinogluino` and `SUSY:qqbar2gluinogluino` flags; and the event generation used the default

NNPDF2.3 PDF set [237], the default Monash tune [238], and PYTHIA's default p_T-ordered parton shower, with ISR, FSR and MPI simulation all switched on. Cross-sections were corrected based on the values given in the public LHC SUSY Cross Section Working Group Twiki for gluino pair production [239], which in turn are based on the higher-order cross-sections calculated at NLO in the strong coupling with resummation of soft gluon emissions at NLL, as presented in References [240, 241].

While various parameters associated with the Gbb model in this study were hard-coded into the workflow, it should be relatively straightforward to adjust the procedure for any other simplified SUSY model. The process is not limited to two dimensions – indeed, a brief test was carried out in which the top-squark mass was also varied. All the relevant code, including the PYTHIA executable and the HTCondor steering scripts, are available in Reference [242], which includes build instructions.

Two approaches were taken to convert the observed signal region event counts in the YODA files from the RIVET analysis into $\mathrm{CL_s}$ scores. The simpler approach is adding this search to the CONTUR database, alongside the expected SM background contribution and the observed value. For this analysis, this comes with the additional ambiguity that the errors on the pre-fit background model are not given, so as a best approximation we had to propagate from the post-fit background errors. It should also be noted that CONTUR uses a χ^2 approximation that is always valid for LHC measurements with high statistics, but may be less applicable to search signal regions where single-digit events are recorded.

A richer description of the likelihood than that encapsulated by just the three numbers given to CONTUR comes in the `pyhf` files provided by the analysis team on `HEPData`, which include correlations with relevant control regions and individual error breakdowns for many of the systematic errors that enter into the fit. To facilitate the `pyhf` approach, a program was written that could read YODA files from across a signal grid, write the appropriate JSON patch for each region, and then combine the patch with the background `pyhf` file and run the hypothesis test[5]. Had the recently released SPEY package [243] – and its `pyhf` interface – been available at the time, it would have allowed a much simpler implementation of this procedure.

Figure 5.8 shows the 95% $\mathrm{CL_s}$ exclusion contour for the Gbb model. Both the CONTUR and `pyhf` approaches produce reasonably similar exclusion contours, consistently within 100 GeV of each-other. That the likelihood calculations differ is not surprising – as discussed in Section 4.1.5, CONTUR and `pyhf` use different statistical definitions of $\mathrm{CL_s}$, and CONTUR is operating with less information than is contained in the `pyhf` file. The `pyhf` result is also consistently within 100 GeV of the original ATLAS exclusion plot – impressive given that the coarseness of the original ATLAS signal grid was 200 GeV[6]. The CONTUR result provides a slightly weaker exclusion, in some places by as much as 200 GeV. Both perform worse in the compressed region along the diagonal, where the physics is more complicated, and our LO PYTHIA event generation may perform worse.

To reinforce the point that this demonstrates NNs can be succesfully reinterpreted, the contour we obtain from reinterpreting the cut-and-count search with `pyhf` in Figure 5.11 differs from the published result by a similar amount, even though it lies significantly further from the difficult-to-model compressed diagonal.

One final question about the likelihood contour obtained in Figure 5.8 is the origin of the "bite" taken out of the RIVET+`pyhf` likelihood at approximately (2200 GeV, 1450 GeV), with a similar but less pronounced effect in the RIVET+CONTUR likelihood. Figure 5.9 shows which signal region is providing the exclusion (i.e. has the highest expected exclusion) at each parameter point, and we can see that the top edge of the "bite" very clearly occurs at the transition between the Gbb-2300-1000 and Gbb-2100-1600 signal regions. Then, as also illustrated in Figure 5.8, because the observed exclusion from the Gbb-2100-1600 is stronger than the expected (i.e. $\mu = 0$) exclusion, this leads to a sharp jump in exclusion, explaining the bump.

[5] Available via Reference [47].

[6] The grid was sampled more-finely than 200 GeV at some points in the compressed region: see Figure 5.10.

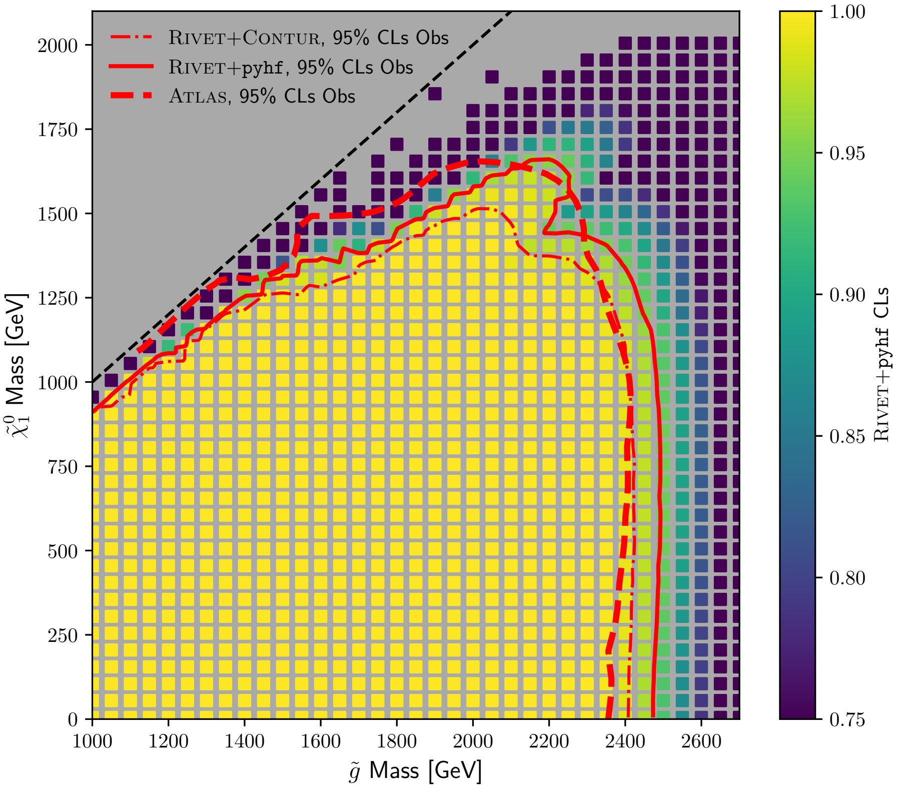


Figure 5.8: Exclusion of the Gbb model, comparing the Atlas result – from Reference [177], Figure 10b – to the reinterpretation.

The fact that the observed exclusion is higher for SR-Gbb-2300-1000 than the expected makes sense given that the analysis observed no data events in this region. Comparing to the regions the original analysis used for exclusion – illustrated in Figure 5.10 – we see that a sharp transition is avoided due to an additional "island" of SR-Gbb-2800-1400 dominated exclusion between the 2300-1000 and 2100-1600 regions which we do not observe in the reinterpretation. It is not entirely clear why this is not observed in the reinterpretation.

In discussing Figure 5.10, it is important to point out that, with the exception of the aforementioned "island", the limiting region in the Atlas analysis is largely consistent with the excluding region we obtained *in the vicinity of the exclusion contours.* The disagreements in the bulk of the contour are not problematic: these points lie at the extreme end of the likelihood distribution, and it is unsurprising that even with pyhf full likelihoods that there are discrepancies. It is also of no physical interest: whether a point is excluded at 10σ by one region or at 12σ by another is completely irrelevant.

We also obtained an exclusion contour for the NN case of the Gtt model, as shown in Figure 5.12. Here, event generation is slightly more complex (as the top-quarks produced in some cases may be quite off-shell); and as such this model was not studied in as much detail. However, even with these difficulties, the contour is broadly in agreement with the Atlas result; always within 200 GeV, and typically within 100 GeV. As for the Gbb case, the exclusion is over-estimated in the bottom left (high gluino mass, low neutralino mass); and improves at higher neutralino masses. Because the exclusion contour lies further from the compressed region, the reproduction difficulties observed in the compressed region for the Gbb model are not repeated.

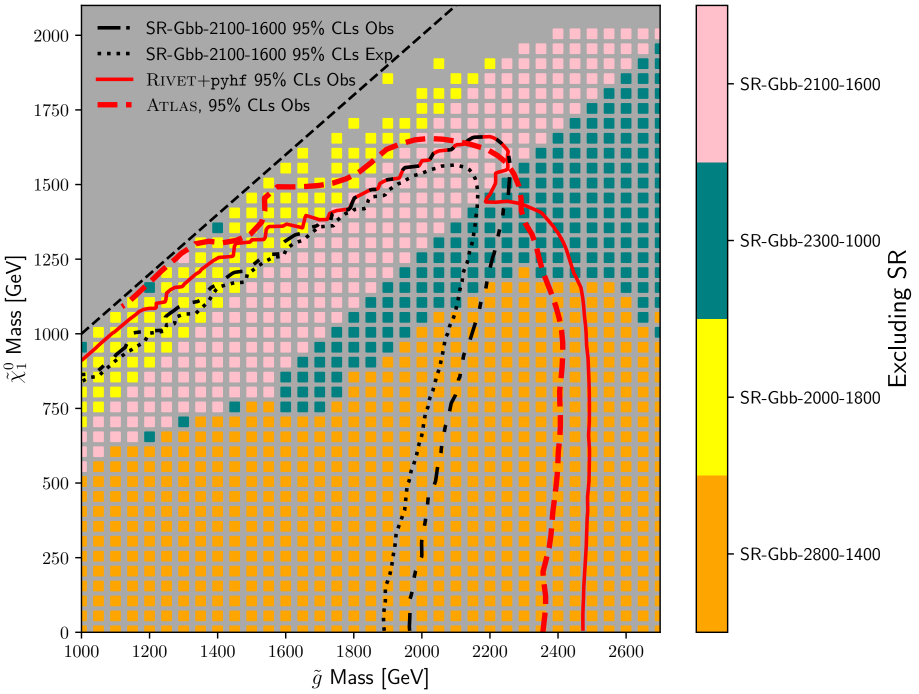


Figure 5.9: Region breakdown of the Gbb exclusion illustrated in Figure 5.8; the "excluding SR" is the signal region with the strongest expected exclusion. The boundary between the Gbb-2300-1000 and Gbb-2100-1600 dominated regions combined with the stronger-than-expected exclusion from the Gbb-2100-1600 region explains the sharp turn in the RIVET+**pyhf** 95% exclusion contour (solid red line).

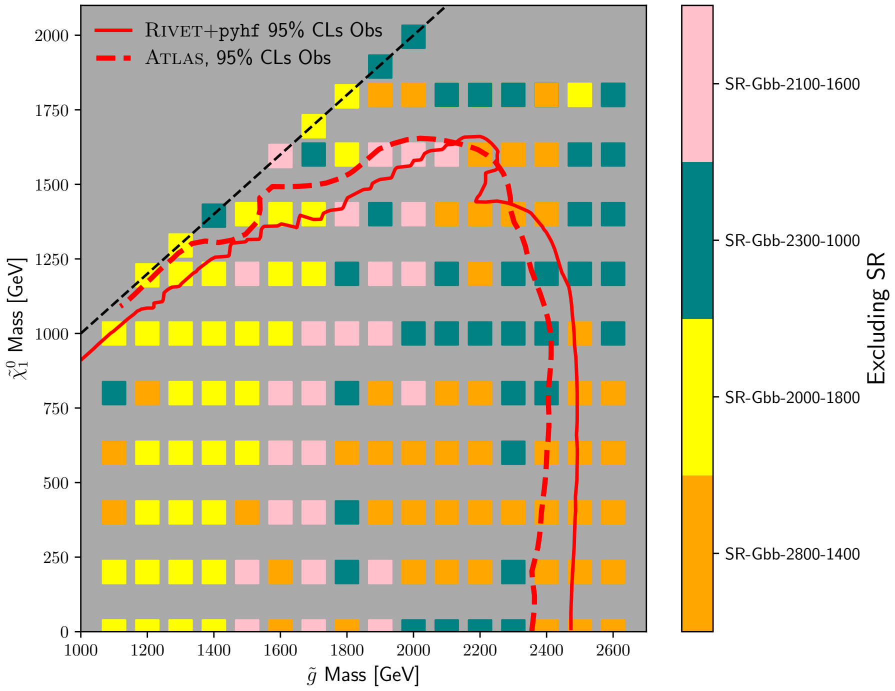


Figure 5.10: The reported most excluding region for the Gbb model from Reference [177], created using inferences from auxiliary Figures 4b and 6b. The main discrepancy is the additional "island" of SR-Gbb-2800-1400 at approximately (2300 GeV, 1500 GeV).

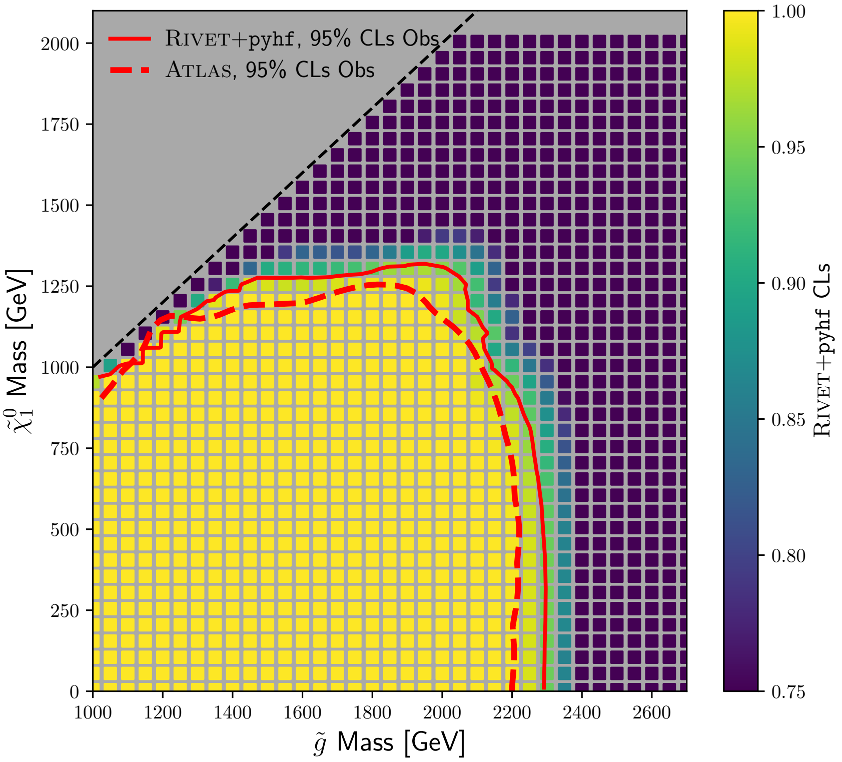


Figure 5.11: Exclusion of the Gbb model using only the cut-and-count regions, comparing the Atlas result – from Reference [177], auxiliary Figure 3b – to the reinterpretation. Performance is very comparable to the NN case in Figure 5.8, especially as the cut-and-count contour lies further from the compressed diagonal where we expect there to be greater modelling discrepancies.

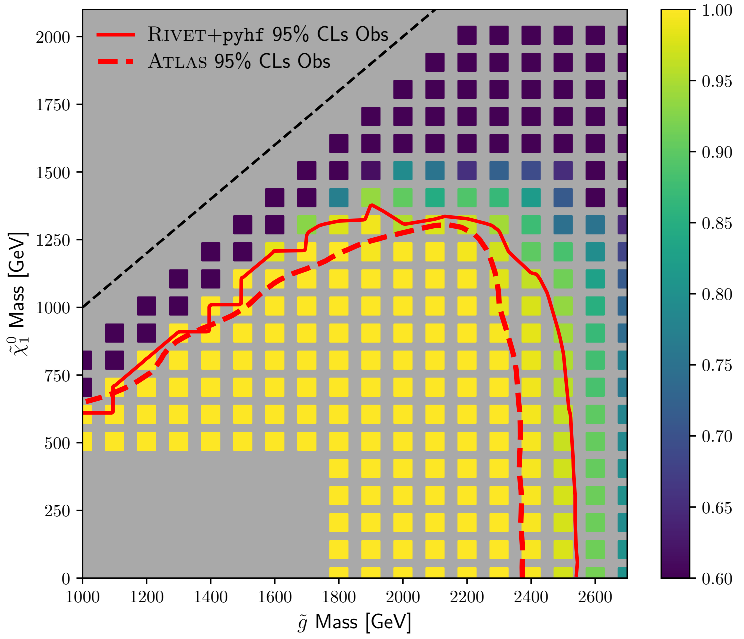


Figure 5.12: Exclusion of the Gtt model, comparing the Atlas result – from Reference [177], Figure 10a – to the reinterpretation. Given the additional difficulties of event generation with this model, the performance is still very good. The "missing" points in the bottom-left corner were not generated in order to economise on computational resources – they are clearly already excluded.

5.6 Reinterpreting MCBOT

Several Run 2 Atlas VLQ searches rely on a neural net called MCBOT (Multi-Class Boosted Object Tagger) [244]. This is a Deep Neural Net (DNN) designed to distinguish the origin of large-radius jets: whether from vector bosons, Higgs bosons, or top quarks – all possible decay products of VLQs (see Section 1.3) – or the background QCD.

While it was not originally made public, the MCBOT neural net was already preserved within Atlas as a `lwtnn` file, making it easy to re-use in Rivet using the tools described in Section 5.3.

This section will focus on work done with the goal of reinterpreting Reference [245], an Atlas search for pair-produced VLQs in final states with a leptonically decaying Z-boson. MCBOT was used to tag hadronically-decaying SM bosons originating from the decay of the "other" VLQ[7], as well as hadronically decaying top-quarks, which could be produced by the decay of either of the pair-produced VLQs. Before validating the analysis as a whole, we also specifically validated the performance of the neural network by reproducing the distribution of the NN output score for the three signal outputs, shown in Figure 5.13. Given the difference in signal model Monte-Carlo, and difficulties in matching the original input distributions – the exact cuts that went into the input were not clear – the re-interpretation performs well, with all the major peaks matching. The only significant discrepancy is a slight bias left in the top-tagging distribution, and undercounts in the zeroth bin of all distributions.

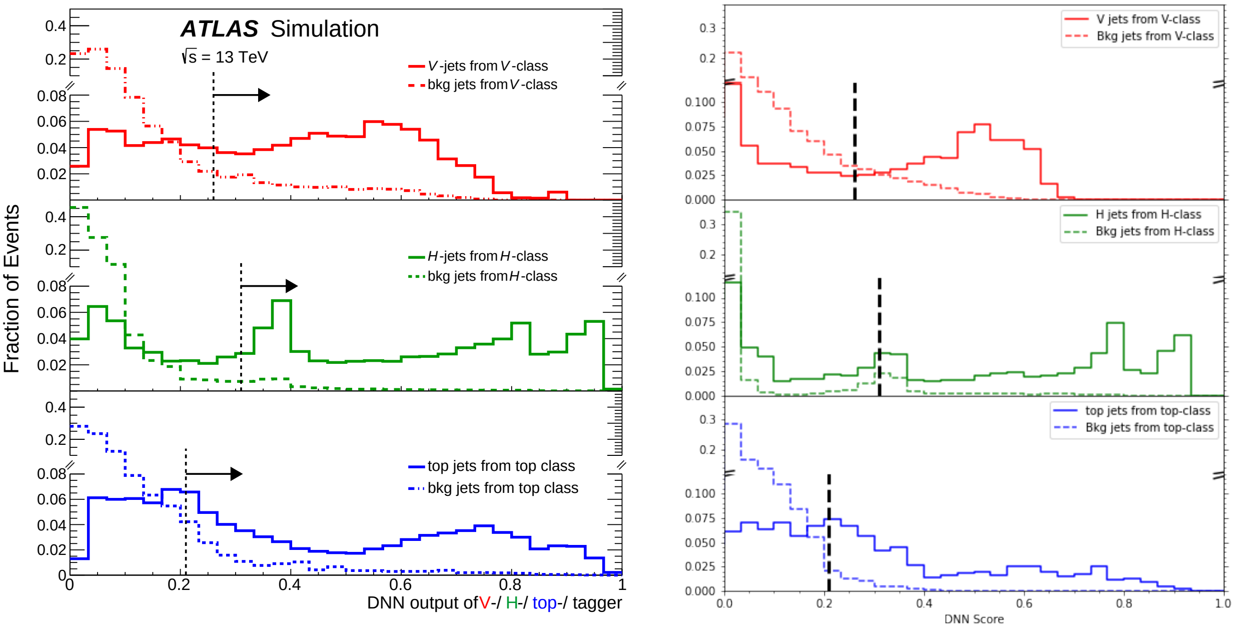
A Rivet implementation was also written for Reference [224], which also used MCBOT. Many interesting studies were carried out on this analysis routine, for example testing the effects of different types of smearing function. One issue identified here was the usage of the pseudo-continuous b-tagging score directly in the network, which is a much harder variable to emulate than a simple yes/no b-tagging boolean. Unfortunately, it used an older version of the tagger for which the DNN weights were not preserved, and as such an accurate re-interpretation was not possible.

5.6.1 Truth-level validation

As we sought to make the `lwtnn` JSON file containing the neural net public, concerns were raised that due to the black box nature of neural nets, it is possible that MCBOT, having been trained on fully-reconstructed Monte-Carlo data (and then evaluated on both real data and Monte-Carlo), may be using reco-level information. This would mean that feeding in truth-level information – or smeared truth-level information – may not accurately replicate the original network. *A priori*, for an arbitrary reco-trained network, this is not a bad assumption, particularly if the network input features include variables which can be hard to replicate at truth-level, such as jet substructure variables.

Therefore, we carried out an additional study to verify that MCBOT was consistent between reco-level and truth-level on an event-by-event basis. The event selection was designed to try to replicate the set of jets used in the validation of the network, rather than the final signal regions of the analysis, in order to increase the number of events in our testing sample. While the original VLQ samples were no longer preserved in the Atlas metadata interface (AMI), we used very similar samples that were preserved. We extracted re-clustered anti-k_T jets following the approach in the original paper, while also extracting the truth-level information using a customised EventLoop algorithm. Additional code was written to allow us to apply Rivet's smearing based detector-emulation without having to rewrite HepMC files for each event. Basic checks were also carried out to ensure the comparison was fair – for example, if the Atlas reconstruction caused the leading truth-level jet to become the second leading reco-level jet, we compared these jets, based

[7] i.e. the VLQ that did not produce the leptonically decaying Z-boson.



(a)

(b)

Figure 5.13: Comparing the MCBOT output distribution produced by the original ATLAS study (a) – taken from Reference [245], Figure 3 – and the RIVET reinterpretation (b). Note that due to differences between the MC generators used for the signal model and some ambiguity about the kinematic cuts for the events in this plot, this is not an exact “apples to apples” comparison.

on a broad $\Delta R \leq 2.5$ requirement, and very broad matching requirements were also applied on p_{T}.

Figure 5.14 compares reco-level vs smeared truth (using the RIVET apparatus) for the four MCBOT outputs. While there is some (symmetric) spread, there is clearly no bias introduced and the results are consistent. This is further validated by Figure 5.15, which shows the same events plotted on a single histogram axis. It is interesting that there is no significant difference in agreement with reco between the RIVET smeared truth values and the original truth values. This probably reflects the stability of the original inputs to reconstruction. For readers particularly interested in the inputs, detailed plots comparing these can be found in Appendix A.

For yet further validation, we repeated this procedure for a non-VLQ BSM model (specifically a Z') and once more the results – shown in Figure 5.16 – show strong consistency between the reco- and truth-level values. In this test, we also examined the small mistag[8] populations, which appear visually distinct from the overwhelming majority of events distributed along the diagonal. The majority of the mistag population for vector-boson and top tagging comes from reclustered jets with a different number of sub-jets (either due to sub-jets merging or due to small sub-jets being reclustered into different jets): this emphasises the importance of “prongedness” in top and vector-boson tagging. Conversely, the majority of the mistags for Higgs-boson tagging come from events were the leading sub-jets were differently b-tagged, showing that MCBOT was sensitive to the (dominant) $h \to bb$ decay mode. Even for both these classes of events (differing number of sub-jets and differing lead b-tags), more than half of events were still tagged consistently.

[8] In this context, by “mistag” we imply that the jet was tagged differently between the full detector simulation and RIVET, independently of which tag was correct, i.e. the top-left and bottom-righ quadrants in Figure 5.16.

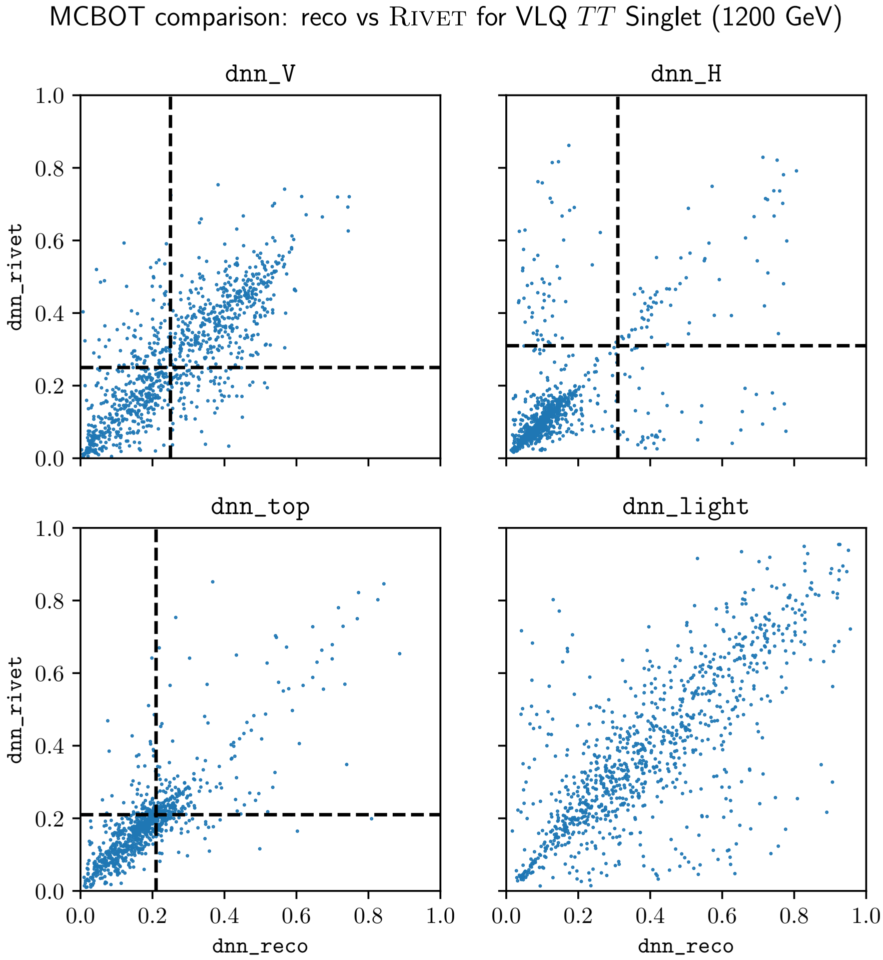


Figure 5.14: All four MCBOT output scores, with the score obtained on the fully reconstructed RC jet score plotted against that for the RIVET-smeared truth RC jet for VLQ signal events. The dashed lines show the NN cut values used for tagging during the analysis.

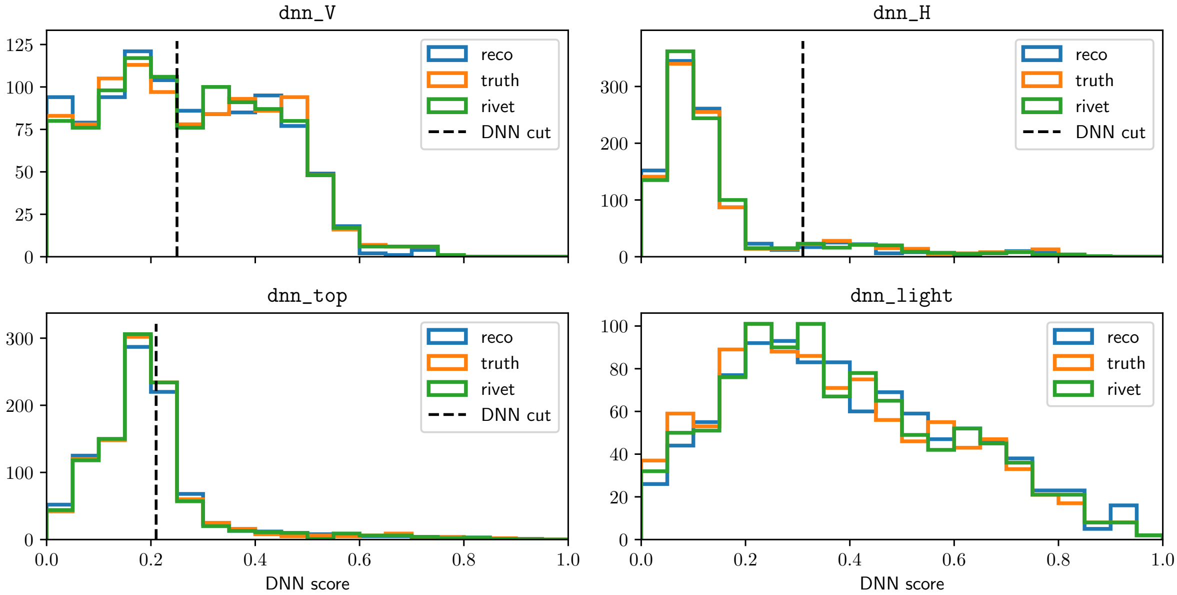


Figure 5.15: All four MCBOT output scores, plotting the overall number of jets in each bin for reco-jets, RIVET-smeared truth-jets, and unsmeared truth-jets. The dashed lines show the NN cut values used for tagging during the analysis. The reproduction of the distribution is excellent.

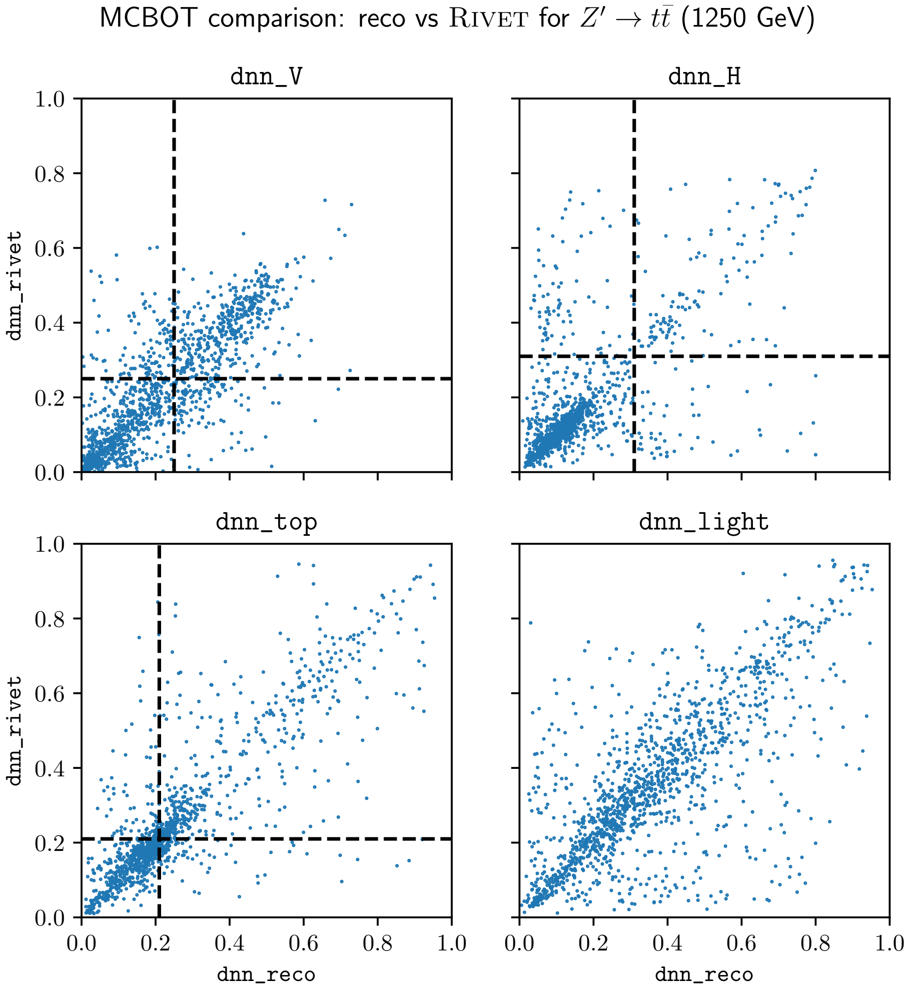


Figure 5.16: All four MCBOT output scores, with the score obtained on the fully reconstructed RC jet score plotted against that for the RIVET-smeared truth RC jet for Z' signal events. The dashed lines show the NN cut values used for tagging during the analysis.

5.7 Reinterpreting an ATLAS top-tagger

A tagger with surface-level similarities to MCBOT – an `lwtnn`-implemented, ATLAS DNN that can tag top-jets – is the top-tagger presented in Reference [113]. The most significant difference is that this tagger uses jet-substructure information such as n-subjettiness [123] for some of its inputs. Substructure variables are significantly more detector-dependent than simple jet-kinematics, and so the tagger may produce quite different results on truth- (even crudely smeared truth-) level data. In order to try to account for this additional uncertainty, we used consituent-based rather than jet-based four-vector smearing, following the approach in Section 6 of Reference [246]. Rather than just applying a Gaussian uncertainty to the clustered jet momenta, this alternate method applies a smearing to all the truth-level particles before they are clustered, and the results in Reference [246] suggest it could significantly improve the reproduction of jet-substructure variables. The raw, unsmeared particle-level data was also studied as a cross-check.

5.7.1 Tagger details

The tagger consists of a simple DNN with 13 inputs, and is designed to tag large-radius ($R = 1.0$) anti-k_T jets as originating from a top-quark (or not). Of the inputs, two are simple kinematic variables: the mass and the transverse momentum of the jet[9], which should be relatively easy for a recasting to reproduce accurately. The remaining 11 inputs are all based on jet substructure information: the n-subjettiness scores τ_1, τ_2, τ_3; their ratios τ_{21} and τ_{32}; various energy correlation function (ECF) ratios [247]; the jet splitting scales $\sqrt{d_{12}}$, $\sqrt{d_{23}}$; and Q_w, the mass of the third-leading subjet when the jet constituents are reclustered with a k_T-algorithm. The tagger outputs a single top-tagging score in the range $[0, 1]$; and comes with 50% and 80% working points, defined with a cut on the score that is a function of the jet p_T.

This tagger has already been used in multiple Atlas analyses, including searches for vector-like T quarks [248], W' resonances [249], and top-squarks [250]. All of these searches would be very interesting to have preserved in Rivet or ColliderBit. The VLQ analysis in particular would offer an interesting complement to the results in Sections 5.6 and 7.4 of this thesis; and an additional top-squark search would be helpful for future Gambit studies.

5.7.2 Analysis and Monte-Carlo details

Reference [113] not only introduced the tagger, but also published several plots illustrating the tagger's performance on real data compared with the expected MC performance. This is effectively a detector-level measurement, and so can be implemented in Rivet. The analysis considered a region rich in $t\bar{t}$ jets, and two regions dominated by important backgrounds: a QCD-rich region and γ+jets rich region (although we did not implement the γ+jets region in the Rivet analysis). The $t\bar{t}$ region required (among other conditions): a large-R jet with $p_\mathrm{T} > 350$ GeV, to ensure the top-quark was fully contained; an exclusive one-lepton cut, to ensure only one top decayed fully hadronically; and various angular requirements between the lepton, the large-R jet, and an additional small-radius b jet. The exact cuts defining the regions are described in Reference [113].

The QCD region was dominated almost entirely by QCD events (>98%). The analysis provided comparisons to Pythia and Herwig with different (but clearly published) scaling factors applied to account for the difficulties of accurate dijet modelling. Because LO Pythia dijet samples are so fast to generate, this was the natural choice to use in the comparison between Rivet and the original analysis. For the purposes of the comparison, 6.3 million events HepMC events were generated.

The $t\bar{t}$ region also included some contamination from single-top and vector-boson + jets samples, though once again we will compare Rivet against the MC results (instead of the data); and therefore only use the $t\bar{t}$ sample. Because it is impractical for a small reinterpretation effort to use the same higher-order MC event generation as the original Atlas analysis, we used MadGraph5_MC@NLO at leading order, and normalised all plots to the same number of events.

Not every jet in a $t\bar{t}$ event is a top-jet. Given the dangers, particularly for heavy resonances, of relying too heavily on the inner details of event generator records described in Chapter 3, what exactly constitutes a true top-jet is not a trivial question. However, for the purpose of training and evaluating this tagger, jets in $t\bar{t}$ events were split into three categories: signal jets, $t\bar{t}$(top); and two background categories, $t\bar{t}(W)$ and $t\bar{t}$(other); defined precisely in Table 5.9. The analysis only evaluated the score on the leading jet, and so the category of the leading jet defines the category of the event. The $t\bar{t}(W)$ category mainly consists of events where the top-quark is produced at sufficiently low transverse momentum that the b-quark and the hadronic products of the W end up in different jets. This was implemented in Rivet by functions that recursed through the HepMC event[10].

[9] The original tagger used a slightly unconventional method of reconstructing the jet-mass from the calorimeter data [113] but, at the level of recreation possible with four-vector smearing, the two should be indistinguishable. This was confirmed in correspondence with one of the original developers.

[10] Yes, this behaviour was actively discouraged in Chapter 3, but sometimes it is impossible to avoid.

Category	Definition
$t\bar{t}$(top)	$\Delta R(j,t) < 0.75$ and $\Delta R(j,b) < 0.75$ and $\Delta R(j,q_{ij}) < 0.75$
$t\bar{t}(W)$	$\Delta R(j,W) < 0.75$ and $\Delta R(j,b) > 0.75$ and $\Delta R(j,q_{ij}) < 0.75$
$t\bar{t}$(other)	Any jet not qualifying for the above.

Table 5.9: How $t\bar{t}$ events are classified, given a top-quark decaying as $t \to Wb \to q_i q_j b$, in the vicinity of the leading jet j. The original analysis added an additional requirement for matching detector-level jets to particle-level jets with a $\Delta R < 0.75$ cut off.

5.7.3 Results

Figure 5.17a compares the output of the tagger from the original study and its reinterpretation, for a $t\bar{t}$(top) signal sample. The reproduction of the signal score distribution appears quite promising. However, the performance for the background samples in the other plots in Figure 5.17 is worse, particularly for the $t\bar{t}(W)$ sample. For all the backgrounds, the top-tagging score is biased towards higher values, implying that in a "real" physics analysis, the false-positive rate would be too high. The QCD sample, shown in Figure 5.17d, is an important cross-check, as the plot is obtained in a very different phase-space, using a different event generator, and without any dependence on parton-matching procedures; but still shows a bias towards false-positives.

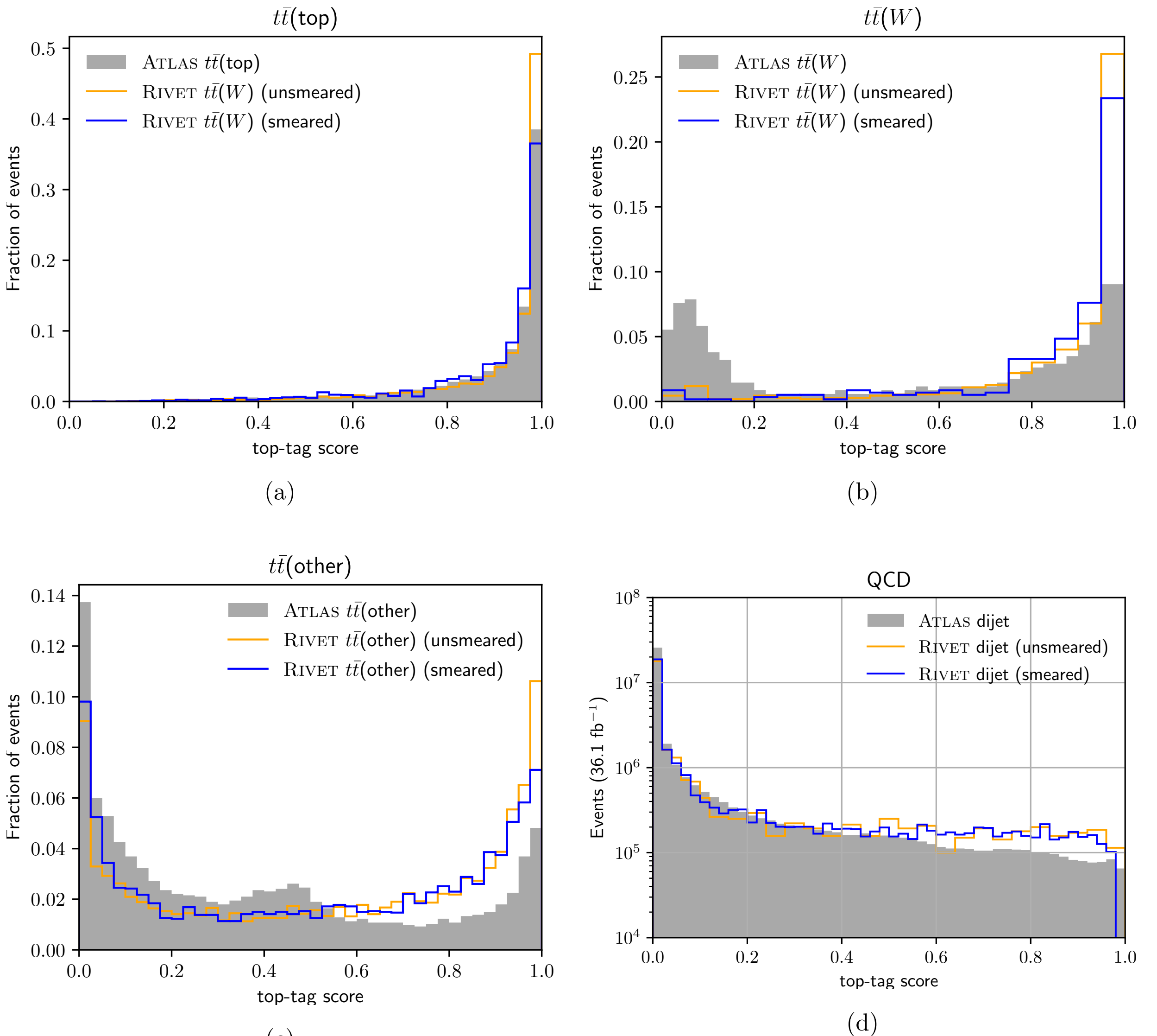
The p_{T}-dependent cut value makes it difficult to estimate exactly how much higher the false-positive rate would be given the information made public by the analysis. However, a good approximation can be found in Figure 5.18, which presents ROC curves for this tagger, with mistag rates presented for all three backgrounds separately. Performance is worse for all three, but most significantly for the $t\bar{t}(W)$ category: the reinterpreted tagger is almost completely unable to distinguish this background from $t\bar{t}$(top).

An earlier description of the same tagger [251] provided some feature-importance information, identifying the two most important features as the jet mass and the τ_{32} n-subjettiness ratio. Reference [113] – in an example of helpful practice – provided plots of these variables. The jet mass – a variable used successfully in many Rivet and ColliderBit analyses – is relatively well reproduced; but τ_{32} is not, both for particle-smeared and unsmeared data, as shown in Figure 5.19. Interestingly, given the successes reported by Reference [246], the particle-smeared datasets tended to perform slightly worse than the unsmeared, though neither performed particularly well. Notably, the best reproduced τ_{32} distribution – $t\bar{t}$(top) – led to the best reproduced top-tagging score.

However, it should also be noted that for $t\bar{t}$(other) and QCD, the mismodelling is towards high τ_{32} (i.e. "two-prongedness"), so the higher false-positive rate for top-tags (which should be negatively correlated with τ_{32}) in those samples cannot be entirely blamed on the τ_{32} distribution. It is possible that the mismodelling of τ_{32} is symptomatic of all the n-subjettiness variables – or even all the substructure variables – being poorly modelled; and that the mismodelling of these other variables contributes to the high false-positive rate.

5.7.4 Conclusion

Based on the results presented above, the Rivet implementation of this tagger cannot yet be inserted into recasts of analyses that use it. However, the performance is encouraging enough to suggest further work may yield positive results. Notably, since the release of the original study, a full Run 2 set of jet substructure measurements focussing on $t\bar{t}$ dominated phase spaces [252] has been published, which may allow for the derivation of better particle-smearing functions. Implementations of the γ+jets sample; or of the related



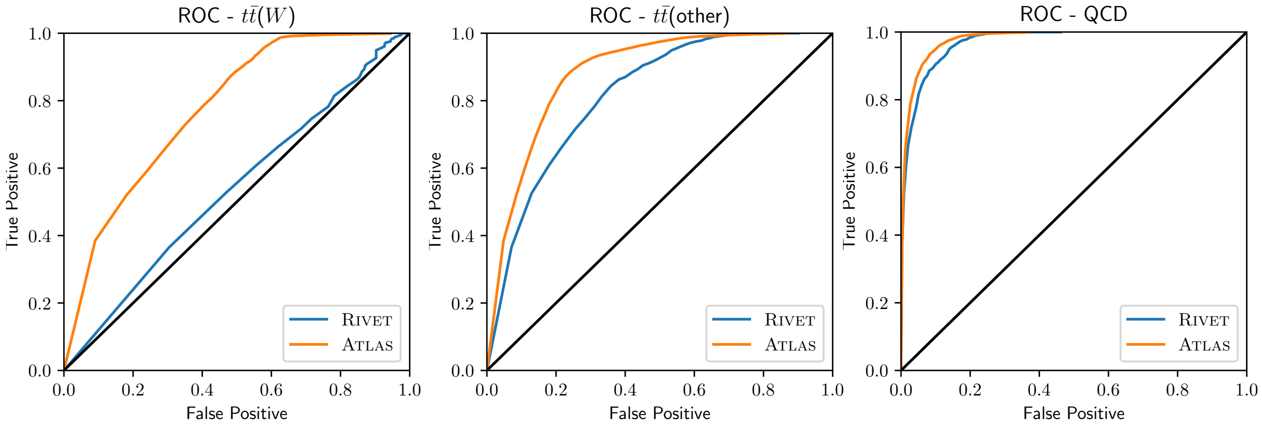
(a)

(b)

(c)

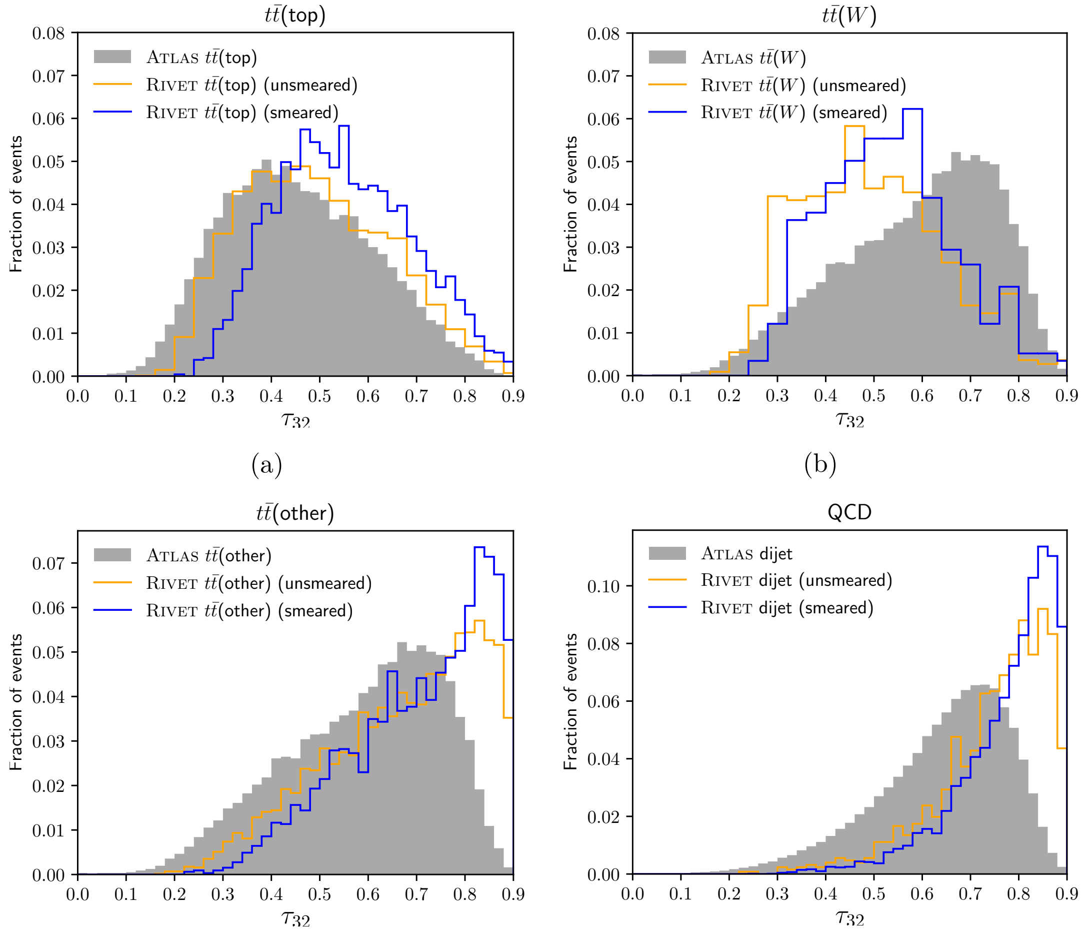
(d)

Figure 5.17: The top-tagging scores for several different MC samples. The ATLAS results have been extracted from Figure 12b (a-c) and Figure 27b (d) of Reference [113]. The reproduction is acceptable for the $t\bar{t}$(top) sample, but poor for the others, with significant biases towards producing false positives. Note the logarithmic axis in Figure (d).



(a) (b) (c)

Figure 5.18: ROC curves compared between the original ATLAS implementation and the (smeared) RIVET one. Unsurprisingly, the RIVET implementation has less discrimination power and, for the $t\bar{t}(W)$ sample, is almost completely unable to distinguish between signal and background jets. ATLAS results were obtained from the values in Figure 5.17.



(a) (b)

(c) (d)

Figure 5.19: The n-subjettiness ratio τ_{32} for several different MC samples. A value close to one implies a more "two-pronged" jet, and a value close to zero a more "three-pronged" one. The ATLAS results have been extracted from Figure 11b (a-c) and Figure 26b (d) of Reference [113]. The reproduction is quite poor for all samples, though best for the $t\bar{t}$(top) sample, which also had the best reproduction of the neural net score.

W-tagger presented in the same paper, may also give further useful information. Any results that improved the re-interpretability of detector-level searches that rely on substructure variables would be beneficial far beyond the recasting of the handful of analyses that rely on this specific tagger.

5.8 Guidelines and Outlook

Building on these experiences, and in collaboration with other academics who had worked on similar challenges in their own frameworks, I put together a set of guidelines for how LHC analyses can ensure their ML-dependent analyses can be more easily reinterpreted [253].

We first reviewed the existing analyses with public ML reinterpretation material: the two ATLAS SUSY searches in Sections 5.4 and 5.5 [177, 233]; a handful of ATLAS SUSY searches based on BDTs [232]; and one NN model from a (ATLAS exotics) Long-Lived Particle (LLP) search[11], which also came with a six-dimensional efficiency map [225]. Notably, no examples have yet come from CMS – we are hopeful that the examples in this thesis, the successes reported by CHECKMATE in the guidelines, and the advice in the guidelines may help CMS – and indeed other groups in ATLAS – to start releasing this information.

A central point in the document was that many of the issues with reinterpreting NNs – as discussed throughout this chapter – can be addressed by thinking about reinterpretation early on in the analysis design process. Most obviously, as noted in Section 5.2, not all ML training frameworks support outputting to a easy-to-reuse format like `onnx`. Given that popular choices such as Keras and Pytorch support `onnx`, it should be straightforward to choose such a framework at the start of the analysis. Another choice that should be considered is restricting the set of input variables to only those that can be well reconstructed at truth-level. While this cannot apply to all analyses, a network that uses only one or two variables that are inaccessible to re-interpreters may be shooting its analysis in the foot, and either eliminating these variables or replacing them with similar proxies could be the best way forward.

Finally – as emphasised throughout this chapter – good quality validation material is absolutely essential to any succesful reinterpretation. In the ML context this must include a description of the network inputs and outputs with their units and conventions; cutflows before and after any ML-based cuts and plots of output variables are also very useful. Often, the simplest way to implement this will be with snippets of analysis code, such as SimpleAnalysis or RIVET.

Of course, without significant advances in publicly available fast simulation, some ML implementations trained at detector- or reco- level will never be reinterpretable at truth level. One solution to this is the continued publication of (increasingly detailed) efficiency maps; but another solution anticipated by the guidelines is the use of *surrogate models*. These are ML models trained to simulate the output of a detector- or reco-level algorithm using truth-level data. For example, a surrogate to a detector-level b-tagger would be trained not to classify jets as b-tagged or not, but to regress to the score the original tagger gave that jet, given truth-level information (e.g. the truth jet kinematics, truth secondary vertex displacement and whether or not the jet contained a truth-level b-hadron). Reference [254] has put forward a suggested framework for these networks at LHC experiments; and we hope that when the ATLAS GN2 tagger is released, it will be published alongside truth-level information to allow for the training of surrogates.

[11] RIVET is not currently well suited to recasting LLP searches, so this analysis was not investigated, but this is an area where future developments in RIVET may prove intriguing.

Chapter 6

Machine-learning-based parametrisation of BSM signal grids

6.1 Reweighting: motivation

Atlas BSM searches typically require a large amount of Monte-Carlo event generation, because the analysis needs to sample a multi-dimensional space of many BSM parameters. They then run into the problem computer scientists describe as the "curse of dimensionality" [255, 256][1]. The more complicated the model, the more parameters there are: for example, the pMSSM has nineteen free parameters [257]. Any technique that can reduce the number of points that need to be sampled would be very useful in easing the strain on computing resources within the experiment.

The objective of the work in this chapter was to examine whether or not machine-learning-based reweighting methods could serve this purpose, particularly focussing on event generation for supersymmetry (SUSY) models as a testing ground. The principle behind reweighting is that, rather than generating a set of events for a parameter point, we take an already existing sample at another parameter point and generate a set of weights for each event in said sample, that allows us to recreate the expected distribution of observables at the new parameter point, using the events generated at the first parameter point. In this chapter, we will be using the machine-learning-based CARL approach [258] to generate these weights. Because we will be using (primarily) particle-level data to generate the weights, we may also be able to make additional efficiencies by minimising the required detector simulation.

6.1.1 Example: reweighting a toy model

As an illustrative example, consider the very simple case where we have tossed an unbiased coin 100 times; and suppose we would like to estimate the head/tail distribution for a coin that has a 75% chance of landing on heads, *without* having to go through the time- and effort-draining procedure of actually tossing this biased coin another 100 times. In this setup, the parameter space of the model consists of one parameter, the "biasedness" of the coin, expressed as the probability that it lands on heads; and there is one experimental observable, the percentage of experimentally observed flips that land on heads.

To reweight the original unbiased dataset to a biased one, we simply assign a weight of 1.5 to all observed heads, and 0.5 to all tails – the outcome of such a test is shown in Figure 6.1. While in this case the solution is so obvious that the reweighting approach seems a distraction – the model parameter, $P(\text{head})$, is effectively

[1] For example, sampling every point in a 10×10 grid is relatively easy; in a 10^{19} hyper-grid, less so.

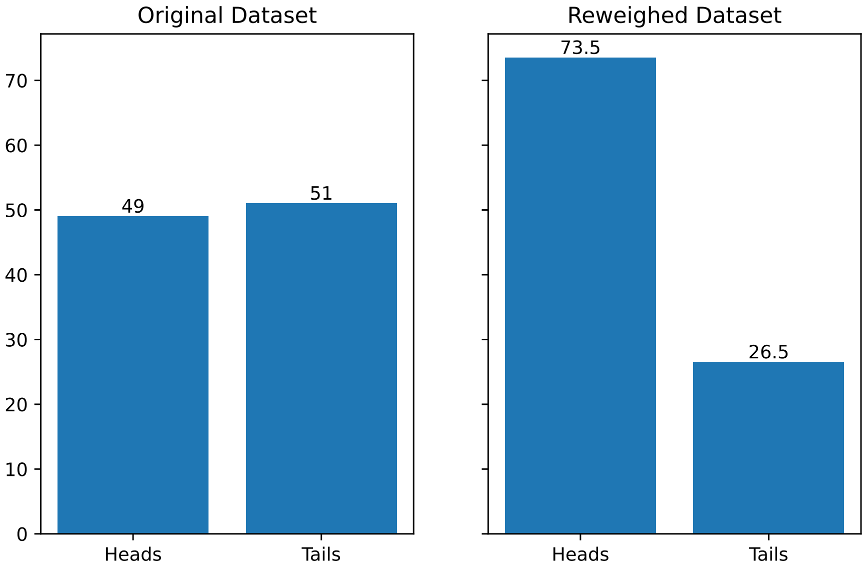


Figure 6.1: Reweighting a toy model: on the left we have the original distribution, obtained by simulating 100 coin flips with a random number generator, and on the right, we reweight to a different model where the coin is biased with $P(\text{heads}) = 0.75$.

the same as the observable – for a Monte-Carlo event, assigning a weight based on all the relevant physical observables and model parameters is a complex problem, for which we will use the CARL machinery described in the next section.

Of course, reweighting will never be a perfect solution: aside from possible statistical issues such as possibly reduced statistical power, we cannot expect the neural network to be able to reweight to physics not present in the nominal. For example, in the pMSSM, as the sparticle mass hierarchy changes, various decays switch on and off, and the network will not be able to reweight across this boundary in the parameter space. Similarly, if the nominal distribution has no values in a certain range, it will never be able to reweight the alternative to these values, as infinite weights aren't possible – in the toy coin model, if you bias a coin such that it has a non-negligible probability of landing on its side, you will never be able to accurately reweight to this setup from a sample where the coin had a curved edge such that it could never land on its side. Mathematically speaking, the domain of the target distribution must be a subset of the domain of the nominal; i.e. the nominal must "support" the target.

6.2 Technical introduction: CARL

The CARL method, built on the statistical underpinnings of Reference [258], is a method that uses a neural network to parametrise the probability density ratio of two distributions, and uses this ratio to "learn" how to weight the two distributions on to each other.

The method is built around a neural-network classifier. Trying to reweight from a *nominal* probability distribution $p_0(\vec{x})$ to an *alternate* $p_1(\vec{x})$, the neural network learns to classify the probability that an arbitrary point $\vec{x}$ in the parameter space belongs to one distribution or the other. Formally, this means the network estimates $s(\vec{x})$, defined as:

$$s(\vec{x}) = \frac{p_0(\vec{x})}{p_0(\vec{x}) + p_1(\vec{x})} , \tag{6.1}$$

which varies from zero if the network "thinks" the event definitely came from the alternate, to one if it definitely came from the nominal. Within this equation, the presence of the density ratio of the two probability distributions, $r(\vec{x}) = p_0(\vec{x})/p_1(\vec{x})$, is clear. Therefore, by re-arranging, we can extract an estimate of the density ratio, which is called the Carl weight:

$$r(\vec{x}) = \frac{s(\vec{x})}{1 - s(\vec{x})} \; . \tag{6.2}$$

Therefore, once the network has been trained, it can be used to produce a set of weights for another sample from the nominal distribution that reweight it onto the alternate.

At first glance, this may suggest that the network can only reweight between two points at a time, requiring training data at both points in the parameter space – which we call "point-to-point" reweighting. This would be useless for our final goal since, if we needed to generate events to provide training data at a parameter point, then we would already have a sample at that point and we would not need to reweight to it.

However, if the network has multiple parameter points in the alternate sample, and the parameter values are included in the data fed to the network, then the network can also learn to reweight to an arbitrary point. This is demonstrated by the toy example in Figure 6.2. Here we trained a network to reweight from a Gaussian, to a set of skewnorm distributions with skewness parameters α taking integer and half-integer values in the range [-3, 3]. The trained network succesfully reweighted to a distribution with $\alpha = 0.75$, verifying that interpolation to new points is possible. However, it is worth noting that reweighting to a point outside the range that was trained on did not work, as shown in Figure 6.2c.

Nevertheless, this approach requires a lot more training data and time, and so for initial tests we will often first carry out point-to-point reweighting – if this proves impossible, then reweighting to an arbitrary parameter point will also not work.

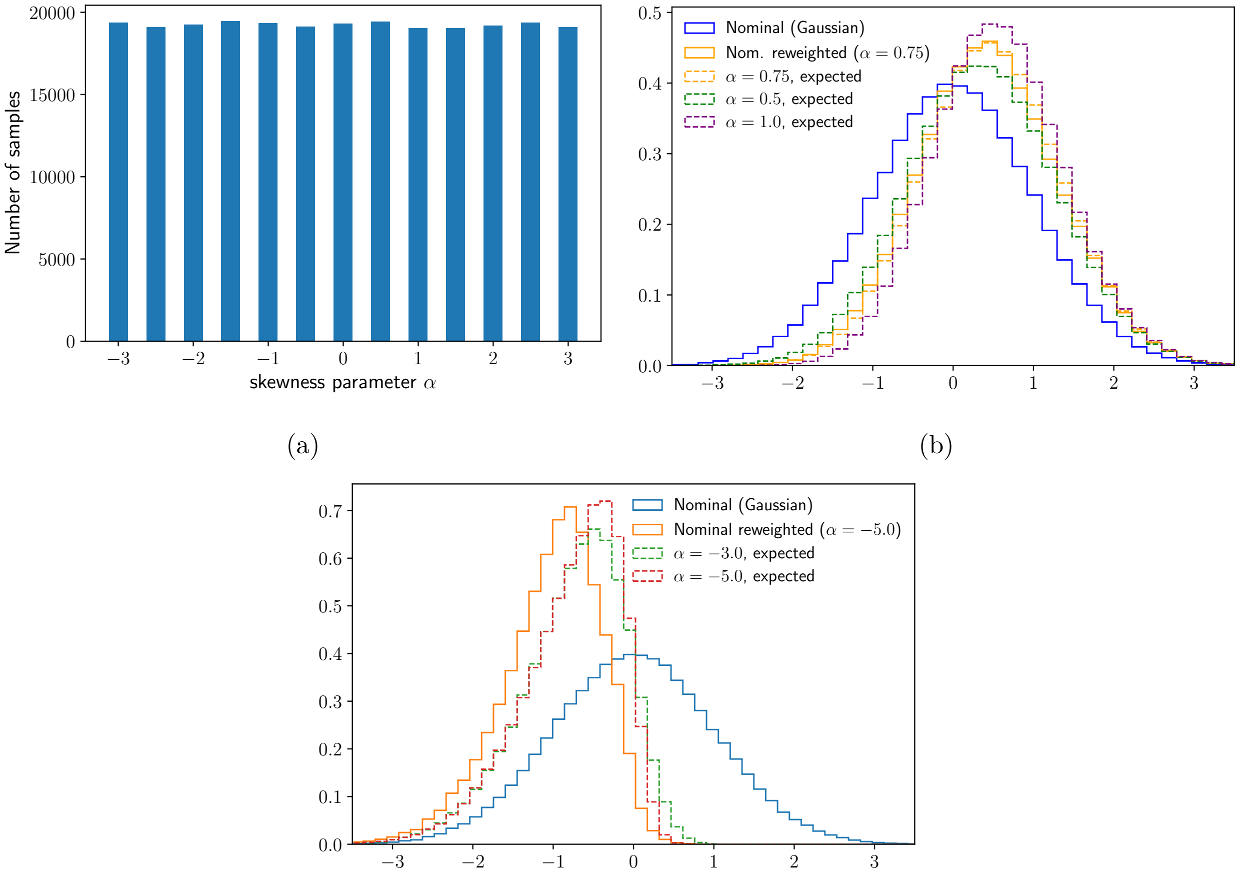
The `carl-torch` package [259], is a Python toolbox that automates much of the process outlined above. It uses the Pytorch package to train and evaluate the classifier neural network, by default using the `adam` optimiser and a binary cross-entropy loss – which we used throughout this Task – although other options are available. It also reads in `ROOT` datafiles and automates the process of parsing them and passing data to Pytorch.

6.3 Initial tests

For initial testing, we re-used the signal grids generated for an Atlas SUSY study searching for squarks and gluinos [260]. The final state for both squark and gluino pair-production in the models considered was a pair of LSPs (observed as E_T^{miss}), a pair of W-bosons (targetting particularly the case where one W decayed leptonically and the other hadronically), and spectator quarks. This final state was attractive because it contained jets, leptons, and E_T^{miss}; so we would be able to test the reweighting on a wide variety of observables. The grids generated for this analysis sampled a three-dimensional SUSY parameter space, defined by the gluino, neutralino, and squark masses ($m_{\tilde{g}}$, $m_{\tilde{\chi}}$, $m_{\tilde{q}}$). In order to simplify the test cases, we only used samples from the 2D slice at a squark mass of 60 GeV. We also generated data at a few additional points to "fill in" the grid and provide more training data. The distribution of all points to which we had access is shown in Figure 6.3.

In order to prepare the data for reweighting, we used the CARLAthenaONNX framework [261]. CARLAthenaONNX is a framework for parsing input DAOD files[2] and extracting only the parameters that the

[2] An Atlas data-format containing events that have undergone some level of processing and filtering.



(a)

(b)

(c)

Figure 6.2: Tests carried out on a skewnormal toy model. The network learned to reweight between different distributions, characterised by the skewness parameter α.
The network used was a simple DNN with three hidden layers, each composed of 10 neurons. The distribution of α in the alternate training sample is shown in Figure 6.2a.
Figure 6.2b shows successful reweighting from the nominal sample to a skewness of 0.75, which was not included in the training sample. Also illustrated are the expected distributions for skewness 0.5, 0.75 and 1.
Figure 6.2c shows unsuccesful reweighting to a skewness of $\alpha = -5.0$. The expected results for $\alpha = -5.0$, as well as $\alpha = -3.0$, the most negative α used in the training, are also shown.

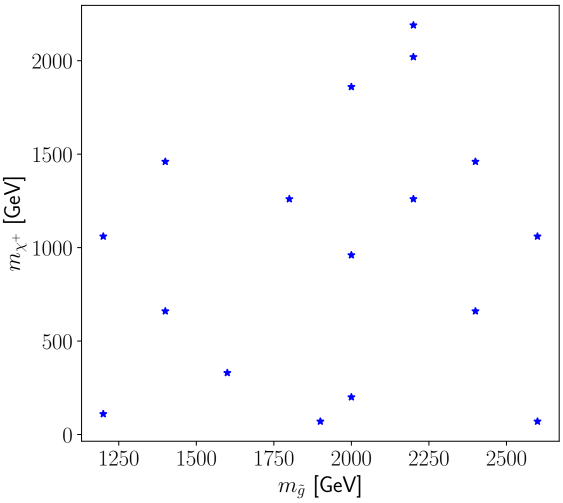


Figure 6.3: All the points in the gluino-neutralino mass plane for which training data was obtained, for the signal model used in Section 6.3.

CARL neural network needs – including possible calculations of event-level quantities such as the effective mass m_{eff} – before writing simplified output ROOT files that the CARL machinery can read. This procedure is also very useful for merging input files together. A specific new CARLAthena algorithm was written for this project: the observables this algorithm selected for the tests included event-level variables such as the effective and transverse masses (m_{eff}, $m_{\text{transverse}}$), the missing transverse momentum, as well as simple kinematics[3] for the four leading jets and leptons. Not all tests used all the variables – indeed, for the very first tests we trained the network using just one distribution as an input, effectively reproducing the simple test from Figure 6.2, but for a slightly more complex probability distribution.

Initial tests were carried out by training the network to reweight a small subset of the observables (for example, the m_{eff} and the E_T^{miss}) from one fixed point to another, using 10 000 events from each point. The results – shown in Figure 6.4 – were poor. The "spikiness" of the distributions shows that a small number of very large weights were dominating the results.

However, on expanding the training dataset to 100 000 events per point, results were more positive. We succesfully demonstrated "point-to-point" reweighting for a wide range of variables: for example, Figure 6.5 shows a network trained to carry out point-to-point reweighting across almost the entire range of the parameter space. The network was always a DNN with three hidden layers of 50 nodes each. At this early stage, there was no benefit to fine-tuning the hyperparameters, although it was noticed that adding score-supression regularisation – i.e. effectively punishing particularly large weights – significantly improved performance.

However, not all tests were as succesful. Figure 6.6 shows a very poor reweighting of the second jet p_T. This is a clear example of the scenario described in Section 6.1.1 – the domain of the alternate distribution is wider than that of the nominal, and therefore it is impossible to find a finite weight that would reweight the nominal to the alternate in the non-overlapping region. It is further noteworthy that when there is such a non-overlap, the reweighting is poor even in the overlapping regions. Indeed, there is support in all but the three left-most bins (i.e. over 90% of the domain is shared), but the performance of the reweighting is poor everywhere, even where there is significant support.

Therefore, as we cannot simply assign some sort of "domain of validity" based on the domain of the nominal sample, this means we always need to try to reweight from a wider to a narrower distribution.

[3] Transverse momentum, mass, pseudorapidity, and the azimuthal angle φ.

Similarly, there will be cases where the two distributions have a similar range but, due to an offset there are nevertheless regions where there is no support for the alternate – for example, Figure 6.7 shows effective mass distributions of similar widths which, due to an offset, cannot be effectively reweighted between in either direction. Similar problems have been seen in other attempts to use neural networks in reweighting problems, for example in Reference [262].

This problem becomes more serious the more observables we try to reweight, as we require total overlap between all the distributions – i.e., a region of no support in one parameter of a multi-parameter reweighting will negatively impact reweighting across all parameters. If there are many variables to consider, it is likely that either there will be a significant offset between the two points in one distribution; or one of the pair will be significantly wider in one variable, while the other is significantly wider in another variable. This means that reliable reweighting in *either* direction would become impossible.

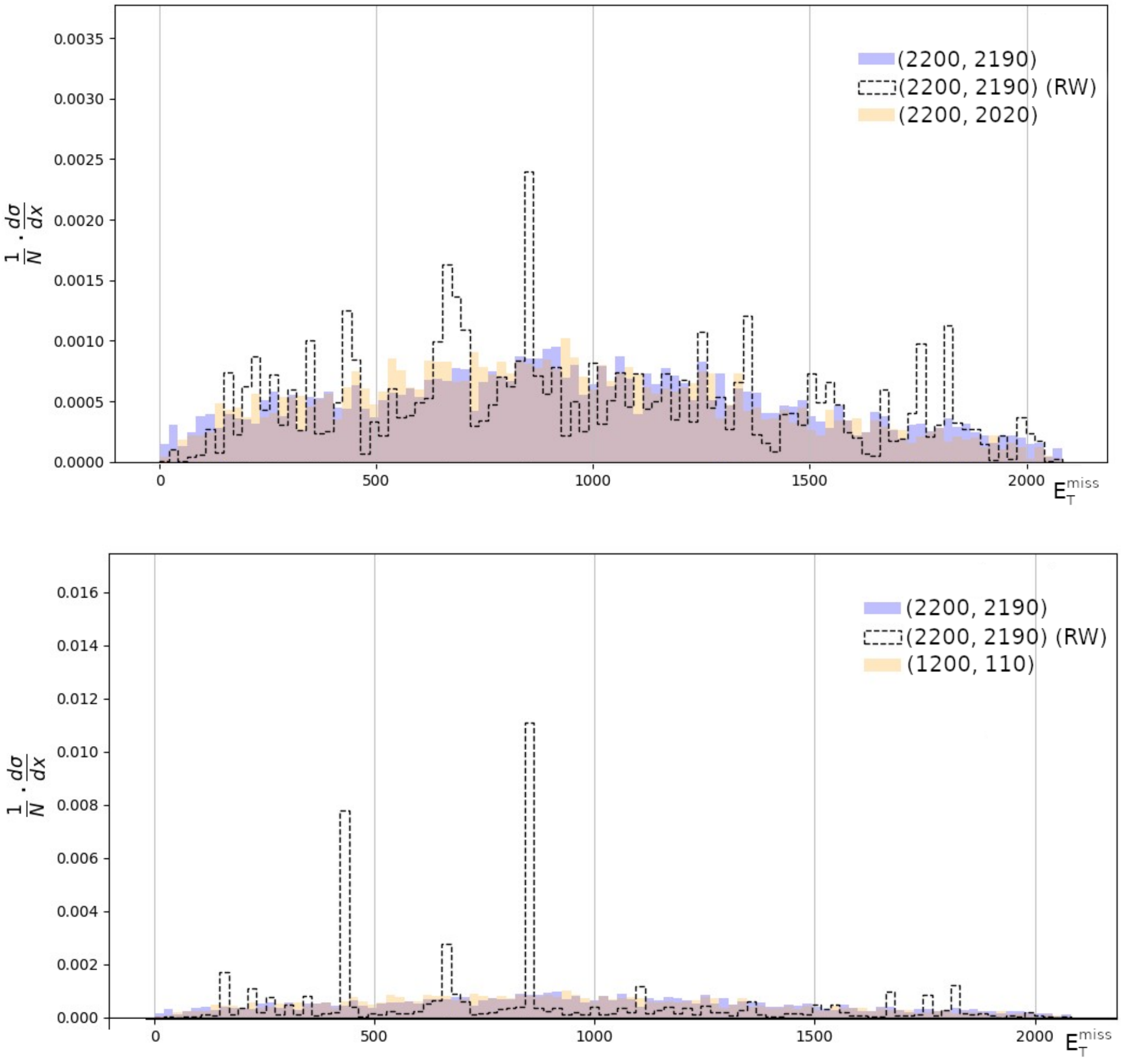


Figure 6.4: Reweighting the E_T^{miss}distribution, from (2200 GeV, 2190 GeV) to (2200 GeV, 2020 GeV) (above) and (1200 GeV, 110 GeV) (below). In both plots, the nominal is in blue, the alternative (target) in yellow, and the reweighting is dashed. Training was done on 10 000 Monte-Carlo events. Unphysical spikes appear at the same values of E_T^{miss} for all reweighting attempts from this point, although the results are worse the greater the reweighted distance.

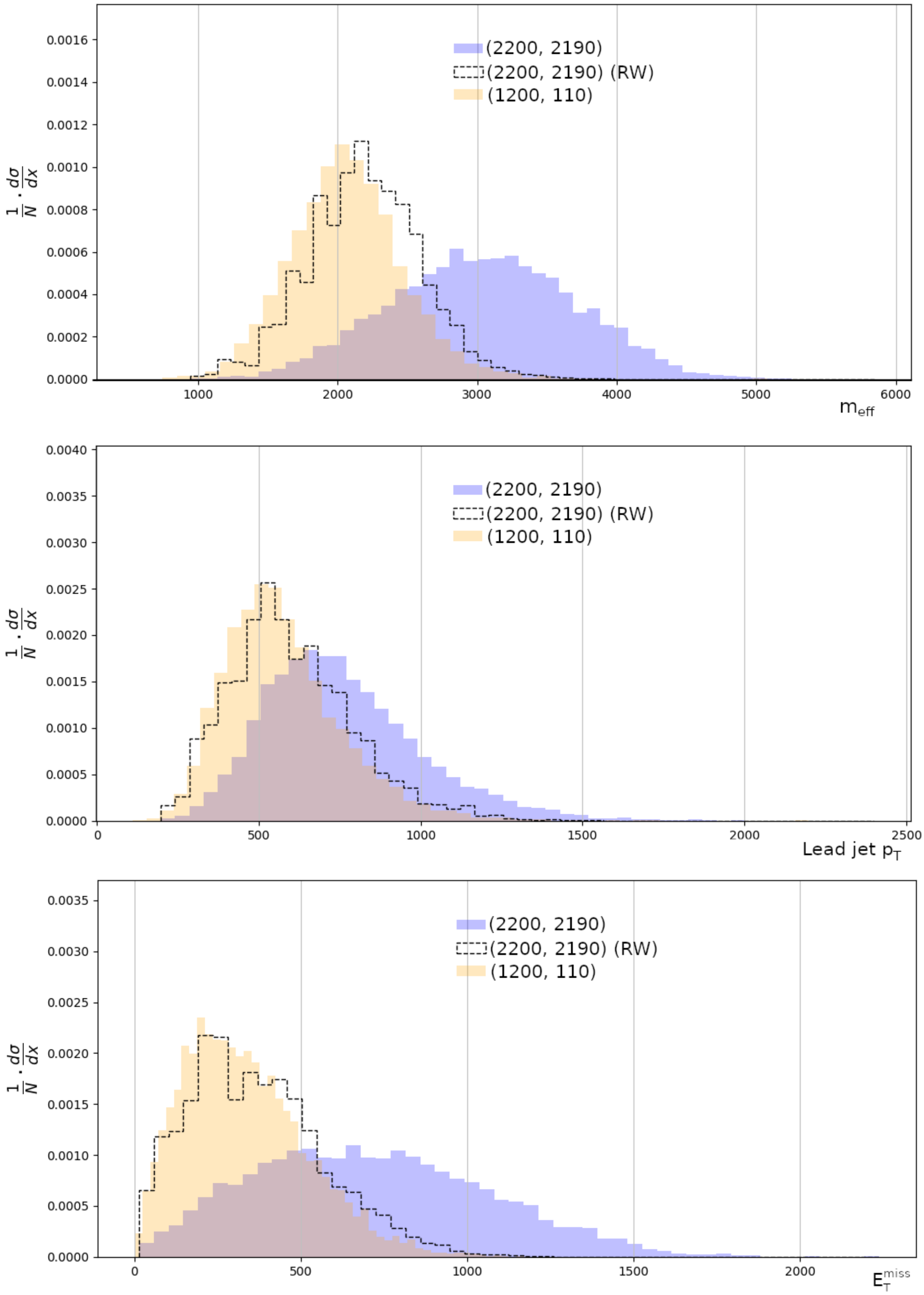


Figure 6.5: Reweighting the m_{eff}, leading-jet p_T and E_T^{miss} (top to bottom) distributions from the point at (2200 GeV, 2190 GeV) to the point at (1200 GeV, 110 GeV) of Figure 6.4 using all 100 000 points in the training, which was performed using the full set of 33 possible input parameters. The performance is still relatively good, even though we are traversing almost the full range of the parameter space.

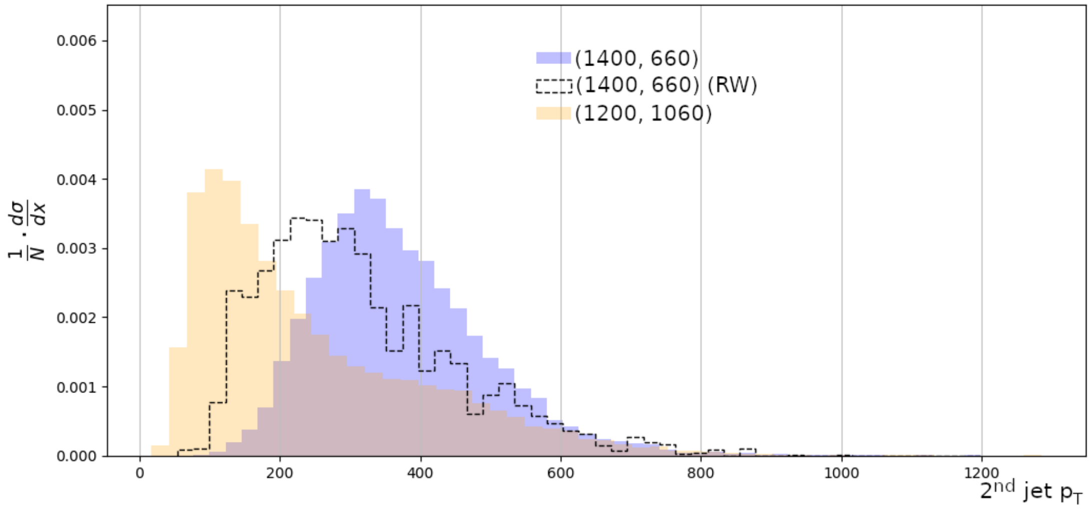


Figure 6.6: Reweighting the second jet p_T from (1400 GeV, 660 GeV) to (1200 GeV, 1060 GeV). There are only three bins where the nominal does not support the target, but the reweighting still performs poorly everywhere.

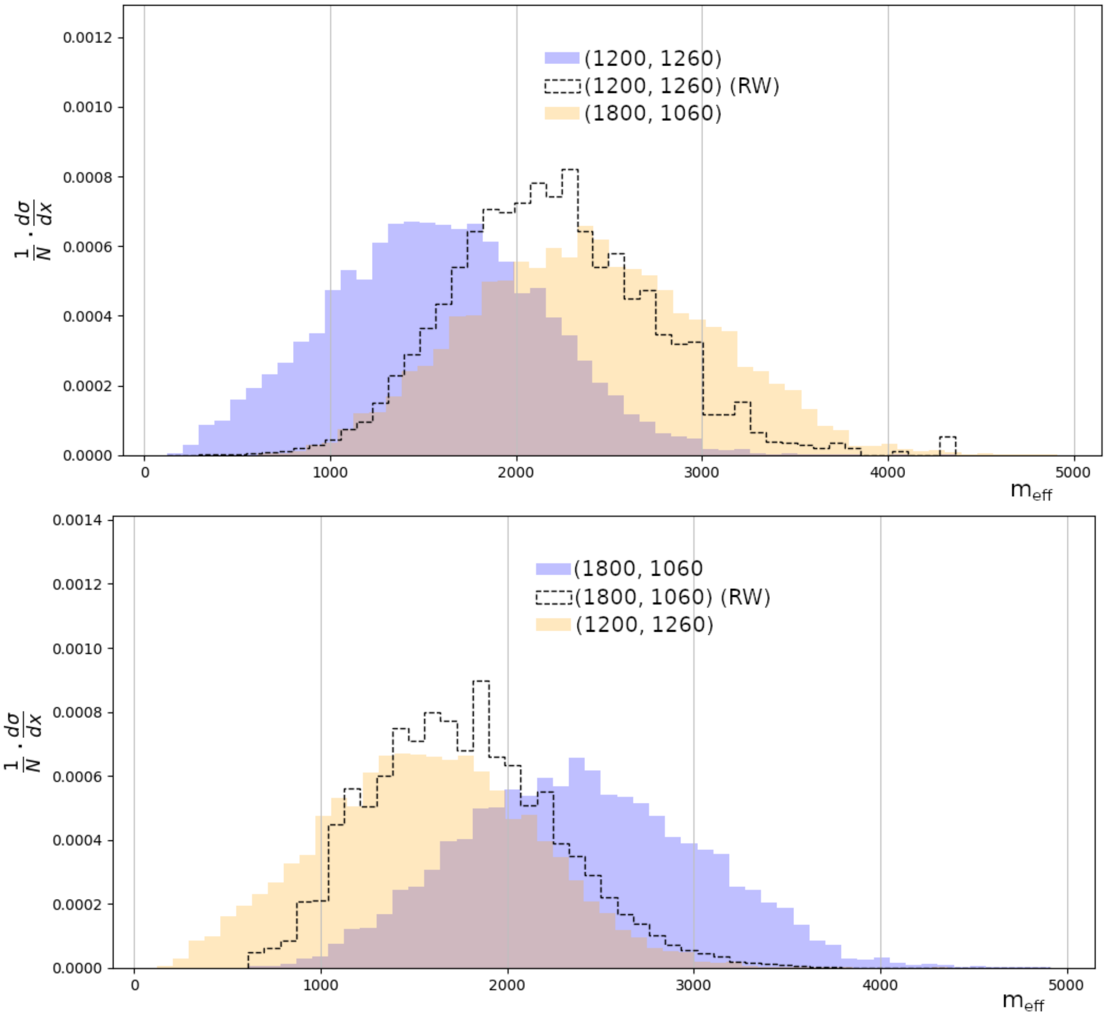


Figure 6.7: Reweighting the m_{eff} distribution from (1200 GeV, 1260 GeV) to (1800 GeV, 1060 GeV) in the upper panel; and in the opposite direction in the lower panel; for the model described in Section 6.3. Because neither distribution fully supports the other, the reweighting performance is poor in both directions.

As we look forward to interpolated reweighting to completely new points, the problem becomes even more severe: *a priori*, we have no way of telling what the domain of the observables at a new point will be, so we have no way of knowing if the reweighting procedure was valid.

6.4 Ensuring support: the "metapoint"

There is actually no requirement that the nominal distribution be physically valid, as long as the alternate distribution it gets reweighted to is. Therefore, rather than reweighting from a single point, we decided to try reweighting from a "metapoint", with a sample formed from the amalgamation of several points from across the parameter space. By choosing to merge distributions from all the way across the extremes of the space, our nominal distributions should be as wide as possible, meaning that the problem of lack of overlap should be greatly reduced, if not eliminated.

This approach also solves another possible conundrum for interpolating the reweighting. When reweighting to a completely new point, how do we decide which parameter point to reweight from? Should we use only the nearest neighbour? Should we combine the results from reweighting all the points? Should we penalise reweighting from more distant points? In this approach, we reweight from all the points amalgamated into the metapoint at once, and the neural network learns whether or not to prioritise nearby points on its own.

To enable the network to reweight to an arbitrary point, the alternative points should provide their model parameters – in this case, $m_{\tilde{\chi}}$ and $m_{\tilde{g}}$ – to the network as training features for each event, in the same way as observables such as m_{eff} or leading-jet p_T are used as network inputs. Assuming there are enough alternate points, the network learns how to adjust the weights as a function of the target parameters, and this enables it to reweight to points that are not included in the alternate sample.

Which values the nominal file should display for the model parameters is a more difficult question. If we provide the "true" parameter values in the nominal, then the classifier will be able to distinguish perfectly between nominal and alternate using these values, and equation 6.2 would break down. Rather, the nominal should get randomly assigned parameter values from the possible values in the alternate – this way the classifier will not be able to use the parameter values as a shortcut. This does mean that the network does not directly "know" which point it is reweighting from, although it can access this information indirectly because of correlations between the parameter values and the other observables used in training.

To recapitulate what this means for neural net training: we train our NN to classify events – summarised by a feature vector containing physics object kinematics, event-level variables and the two model parameters – as belonging to either the nominal (i.e. metapoint) sample or the alternate sample. In training, the model-parameter entries for the nominal sample are drawn at random from the possible parameters in the alternate sample. Then, to obtain weights for a particular parameter point, we use the same nominal sample as in training, but replace the model-parameter entries with those of the target point.

6.5 Testing the metapoint

6.5.1 Machinery

Although valuable lessons were learned from the tests in Section 6.3, carried out on the signal grid shown in Figure 6.3; for testing the metapoint workflow laid out above, it was clear we needed a different approach to producing our nominal, alternative and validation samples. We needed more events at each parameter point; we needed more parameter points (especially for validation); and we needed to do this with a minimal memory footprint because, if generating more than 100 000 events at more than 100 parameter points, having to store full HepMC files (or Atlas EVNT equivalent) would realistically become impossible for a proof-of-concept study.

Fortunately, the extensive machinery developed to validate the results of Section 5.5 can be reused with fairly minimal modifications to solve this problem. We ran on the same “Gbb” physics model: for details, refer back to Section 5.5. The publicly available Rivet implementation of the analysis is replaced by a similar analysis code[4] which, instead of obtaining observables and cutting on them to produce Yoda objects with the final event counts, obtains the same observables and dumps them into a Root ntuple. The observables dumped include all 84 inputs into the DNN described in Section 5.5, including small- and large-jet kinematics, as well as event-level variables such E_T^{miss} and $\Delta\phi_{\text{min}}^{4j}$. Using this procedure, we would end up with a grid of approximately 300 Root files, each containing up to 400 000 events. Another notable advantage of this workflow (over, for example, an Atlas signal request) was the responsiveness: when 100 000 events per parameter point proved too few, we could move to 400 000; when it was realised a new variable was of interest, the ntuple-dumping Rivet analysis could be tweaked and new Root files produced. Wholesale regeneration would probably take weeks or months using a more formal system: for us, it required just 8 – 36 hours on the University of Glasgow batch system.

To make the actual Root files that would be fed to CARL for training, additional scripts were written that, given a set of ntuple files from multiple parameter points, would pick random events to include in the nominal and alternate samples. These scripts also ensured the reweighting parameters – the gluino and neutralino masses – were properly inserted into the training data, i.e. with the true values for the alternate sample and with a randomly drawn parameter point from the alternate set for the nominal.

6.5.2 Training and evaluation workflow

Tests used 10 – 40 different variables dumped by Rivet for each event. However, several observables – such as the kinematics of the reclustered large-radius jets[5] – were deliberately excluded from training. This allowed us to test if the event–by-event weights produced by CARL would be able to reweight observables not included in the training data. Excluding all large-jet variables was probably an aggressive choice: a production rather than proof-of-concept implementation would probably include them and gain some additional performance as a result.

Training was carried out on 1.2 – 4.8 million nominal and alternate points in each sample, and typically 200 – 600 epochs were required. Given the available computational resources, this led to training times in the range of 4 – 24 hours. The classifier was always a DNN with three hidden layers, each of fifty nodes: the combination of the slow training time, limited computational resources and limited time meant that there was much less opportunity for hyper-parameter optimisation than a project such as this often enjoys. While this is obviously dissapointing, it should also be remembered that this means the (already very promising) results presented below could likely be improved significantly further with relatively little effort.

After training was completed (based on either a fixed 600 epochs or no improvement in the last 50 epochs), the quality of the reweighting was evaluated by reweighting the metapoint to all parameter points we had obtained data for, and comparing against the original distributions. Due to the number of parameter points involved, this step had to be parallelised using custom-written HTCondor scripts. For each observable, we also plotted the χ^2 statistic and KL-divergence between the reweighted distribution and the target. This procedure produced more than 12 500 plots for each training setup[6]. We strongly considered trying to

[4] Available, along with the rest of the code from Section 5.5, in the repository at Reference [242].

[5] Yes, these should be somewhat correlated with the included small-jet kinematics (albeit some tests only used the four leading small-jets), but this would likely be true for most sets of physically interesting variables in zero-lepton phase spaces.

[6] (40 observables × 312 parameter points) + (2 statistical distance measures × 40 observables) = 12560.

condense the information further (for example, making a single 2D plot of the $\sum \chi^2$ over all observables), but this often ended up hiding interesting results (such as different variables being reweighted better or worse in different regions). Unsurprisingly therefore, the following results can only supply a (hopefully representative and informative) snapshot of all the data obtained.

6.5.3 Results

An illustration of how the metapoint method reweights from many samples at once is shown in Figure 6.8. This shows the nominal sample (the metapoint) split into its constituents, and how they each contribute in different corners of the parameter space. We see as we would expect that nearby points tend to contribute more significantly: this is most clear for the "bulk" point illustrated on the bottom-right, where the four-leading contributions are the four nearest nominal points from the edge. Similarly for the more-poorly reproduced point in the top-right plot, the three most significant contributions (by a large margin) were the three points on the compressed diagonal: unsurprisingly, physics at points in the compressed region is best reproduced by physics at other parameter points in the compressed region.

Perhaps more surprising is the lower-left plot in Figure 6.8. In this particular scenario, the network is trying to reweight back from the nominal to one of the points in the nominal. We might naïvely assume that the optimal weights would be zero for all the $N - 1$ "other" members of the metapoint, and N for all the events from the member of the metapoint being reweighted to. But in fact the plot shows almost the opposite: the point at (1000 GeV, 5 GeV) is not even in the four leading contributions to its own reweighting! That it is not even in the top four is likely to be due to imperfections in the training[7]; but to set all other contributions to zero would be even more undesirable. Consider the case that this point were not in the metapoint: we would still hope that the network would learn to reweight the network to that point. Then, once we add the point itself back in, should we discard the reweighting from all the other points, at a stroke reducing our statistics by a factor of as much as N? As shown consistently in all panels of Figure 6.8, the reweighting performance is better the more points in the metapoint contribute significantly.

A related early lesson learned about the metapoint setup was that performance was improved if the alternate sample, previously made up of events split evenly between a set of $8-16$ alternate parameter points (see for example Figure 6.9a), also included events from points in the nominal set. It may seem strange that the network would need additional information to train the model in order to reweight points back to themselves; however, we must remember that we explicitly chose our nominal parameter points in order to contain the most extreme points in the phase-space, to ensure that the domain of all observables is fully covered. If we do not include these points in the alternate sample, then the NN will not be able to learn (efficiently) the density ratio in these extreme corners – because $p_1(\vec{x})$ from equation 6.1 will be estimated very poorly. This is illustrated in Figure 6.9, which shows the improvements from adding the nominal points into the set of points sampled for the alternate. It is very clear that it is the extreme corners that perform worse with the no-nominal-in-alternate network: in Figure 6.9a it is the corner at the top of the diagonal (2200 GeV, 2005 GeV) with the largest difference KL-divergence; in Figures 6.9b and 6.9c the difference in KL-divergence is greatest at the bottom of the diagonal at (1000 GeV, 805 GeV); and Figure 6.9b also shows issues in the top right corner at (2600 GeV, 2005 GeV).

Another important lesson was that the score-suppressed regularisation, which was found to significantly improve training in Section 6.3, was found to be a hindrance here, as illustrated by Figure 6.10. In retrospect, this makes sense: suppressing large weights was helpful when the training dataset was small, because they likely originated from statistical fluctuations; however, as the training dataset in this section is approximately

[7] Perhaps either a bias against this point because of the distribution of the alternate samples in the parameter space, or a simple statistical insufficiency.

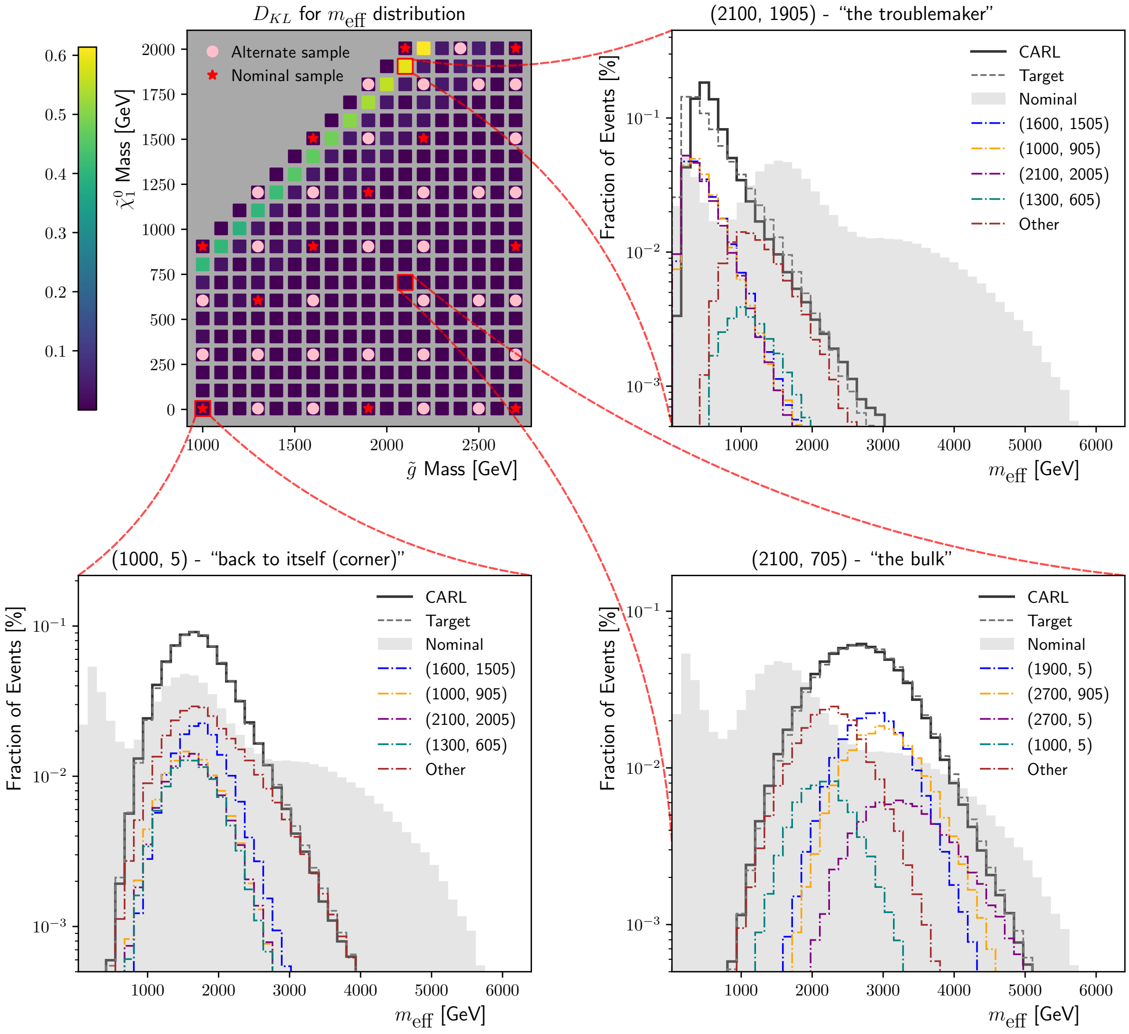


Figure 6.8: Top-left, the post-training KL-divergence in the 2D gluino-neutralino mass plane. Surrounding it, we have examples of how the nominal, target, and CARL-reweighted distributions look at three of the parameter points. To illustrate how reweighting from "the metapoint" works, the contribution to the total CARL reweighted distribution from the nominal points is also shown: the most significant four points (by total weight) are plotted individually and the rest absorbed into "other".

Even the point with a high KL-divergence, by eye reproduces the distribution quite closely. Also, the well reproduced points, both in the bulk and the corner, have significant contributions from many points (to the extent that the "other" contribution is larger than all of the leading four individual contributions); whereas the point with the largest KL-divergence has significant contributions only from three nominal points on the diagonal, with all other contributions at least an order of magnitude smaller.

Training was carried out using 4.8 million events in each of the nominal and alternate samples, and with the kinematics of the four leading-jets and some event-level variables ($E_T^{\rm miss}$, $m_{\rm eff}$, and so forth) as the input features.

50 times larger, statistical fluctuations will be smaller, and larger weights are more likely to be caused by "genuine" effects.

The consistent feature of almost all the plots in this section has been that the worst performance has come in the compressed region – and since the introduction of nominal samples in the alternate as outlined above, particularly in the second most compressed diagonal, i.e. the first diagonal not containing any training data. Various follow up studies showed that similar patterns emerged if all points were shifted 50 GeV up or down, or if the top one or two diagonals were removed from training (with the points in the nominal shifted down accordingly).

This can be partly explained by the fact that the compressed region is where the nominal sample is most different to the target, and so CARL has the most "work" to do here, as shown in Figure 6.11. This is not necessarily a fixed feature of the dataset, and our choice of metapoint will play a role. However, because of the reduced phase-space available to SM particles in the compressed region, they are likely to have narrower distributions and hence will typically be more "different" to the nominal samples which are deliberately chosen to maximise the distribution breadth.

However, this cannot be the only cause of the poor reweighting, as Figure 6.11 also shows several other areas – most obviously the top-most compressed diagonal, but also the bottom left corner – where there is a large KL-divergence between the nominal and the target, but the reweighting (also illustrated in Figure 6.11) performs much better. The result in the upper panel of Figure 6.8 reveals further insight. Points along this diagonal are largely (more than 80%) dominated by the nominal points along the top-most diagonal; and just 100 GeV below, their contribution drops to less than 40%. CARL appears to be struggling to interpolate between these compressed and uncompressed regimes. This could possibly be alleviated with a smarter choice of nominal and alternate points, which we will discuss in more detail in the next section; although notably similar features were visible in one place or another even for set-ups that sampled more densely around the compressed region. Ultimately, we cannot include every point in the alternate sample and there will always be some points between which interpolation will be harder.

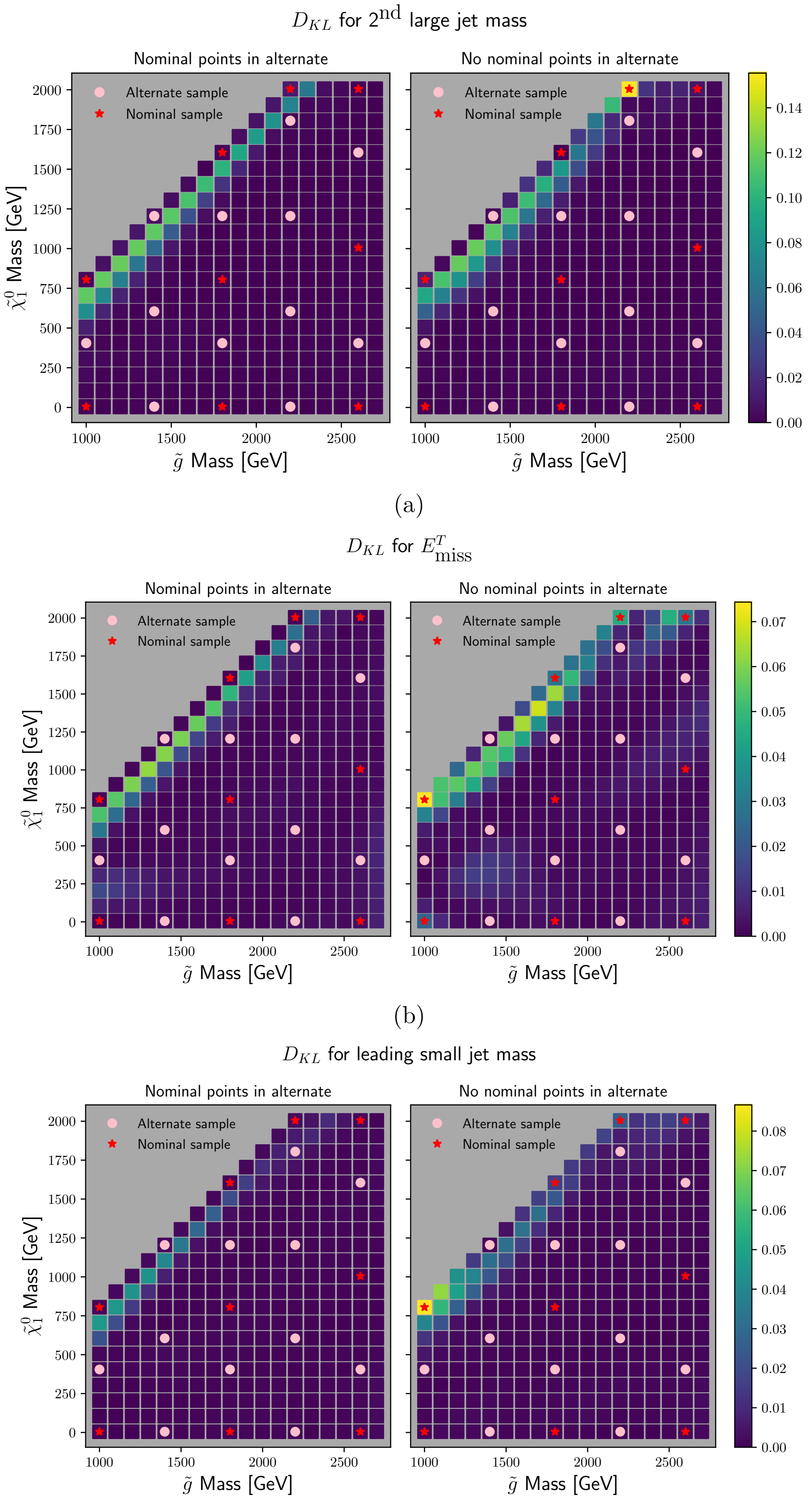
One final comment must be made on the results in this section: we witnessed no saturation as we moved from training on 1.2 million to 4.8 million events per sample; and training even on 4.8 million events with a wider set of input features actually performed worse than a more limited feature-set. This suggests that just adding more training events to the existing set-up may lead to immediate improvements.

Optimising the points in the nominal and alternate samples

As discussed earlier, there is still a lot of scope for hyper-parameter optimisation that could further improve the performance. One set of hyper-parameters (using a very loose definition of the term) that are somewhat unique to this task are the numbers and distributions of parameter points that go into the nominal and alternate samples: though before any particularly complicated schemes that require sampling the entire grid are suggested, recall that for production purposes, the main goal is to avoid doing more event generation than necessary. Nevertheless, there are likely many interesting follow-up problems relating to ML methods that would aim to pick the optimal next point or set of points to do event generation at in order to maximise the performance of the reweighting across the entire grid.

6.6 Scanning a parameter space with a single event-set

In the section above we have obtained a network that, given a set of events from the metapoint, can produce a set of weights that reweight the metapoint events to any point in the parameter space. This means that in principle we can run a RIVET analysis to determine bin counts (and hence statistics such as LLR or $\mathrm{CL_s}$)

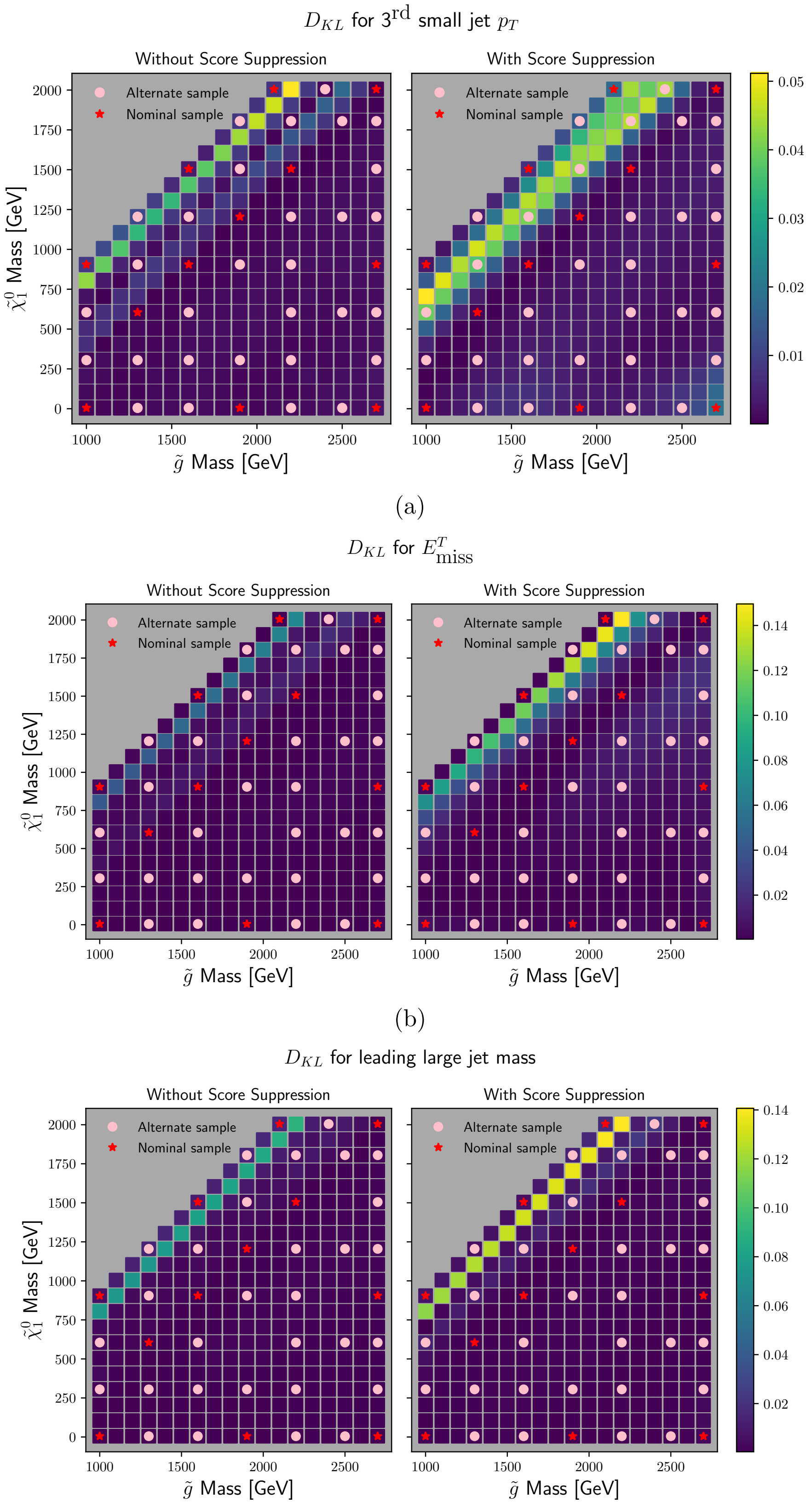


(a)

(b)

(c)

Figure 6.9: The effect on including samples from the metapoint, displayed across the parameter grid for three different observables. Training performs worse when the nominal points are not included in the alternate sample, particularly in the corners around nominal points – e.g. (1000 GeV, 805 GeV) in (a). Training used samples of 2 million nominal and alternate events each, using only a small subset of the observables as inputs (the kinematics of the four leading small jets and some event-level variables).


D_{KL} for 3$^{\text{rd}}$ small jet p_T

(a)

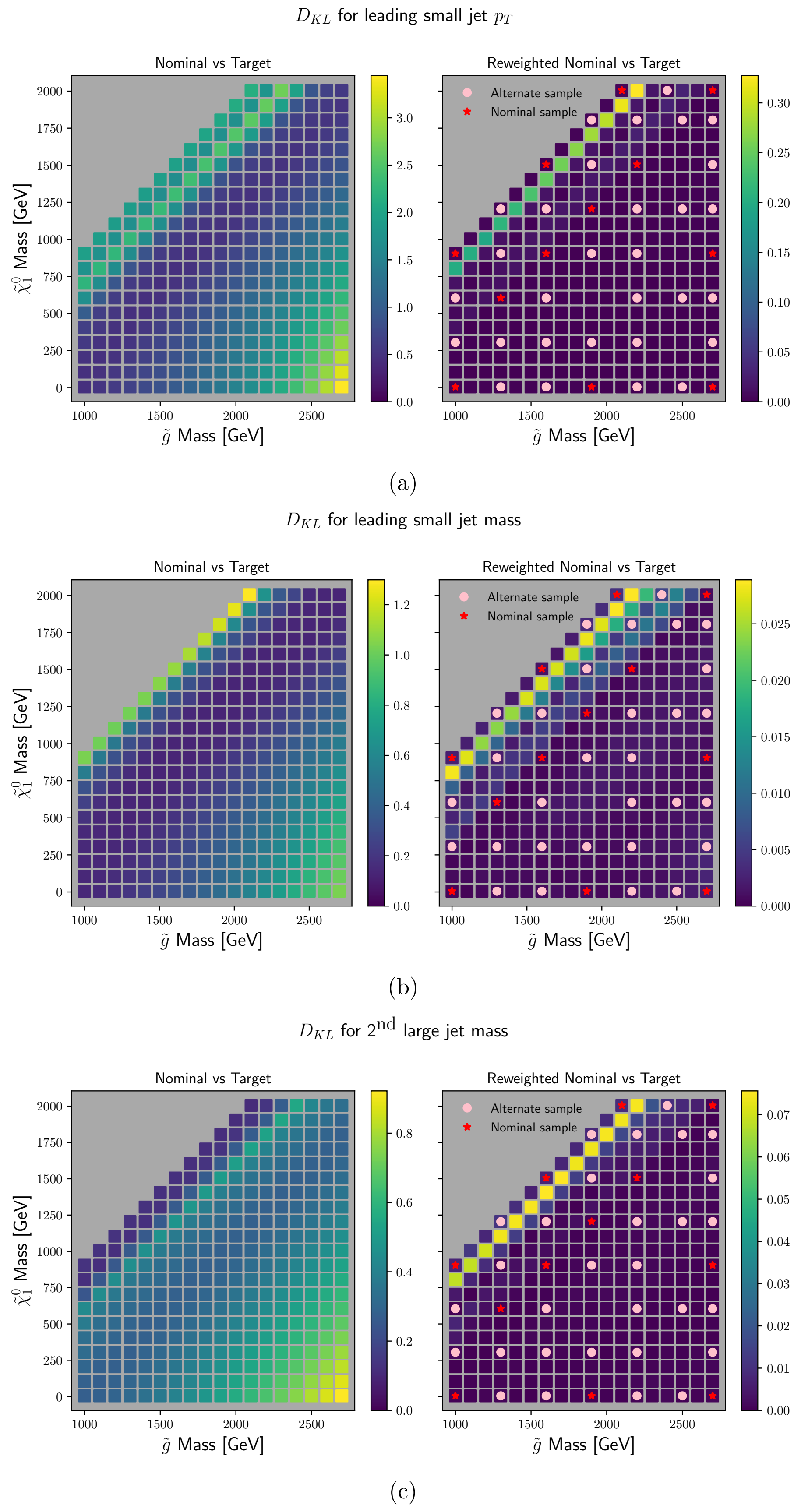
D_{KL} for E^T_{miss}

(b)

D_{KL} for leading large jet mass

(c)

Figure 6.10: The effect of removing the score suppression regularisation: once we were training on a large enough dataset, it became more hindrance than help. Only three observables are shown (note the large jet mass was not included in the training), but similar effects are seen for all the others as well. Results here come from training on 4.8 million events, using only a small subset of the observables as inputs (the kinematics of the four leading small jets and event-level variables).



(a)

(b)

(c)

Figure 6.11: The KL-divergence to the target distribution for the nominal (before training, left) and the reweighted nominal (after training, right). For the majority of variables and the majority of choices of the nominal dataset, the compressed region is likely to have larger discrepancies, because the SM particles produced in the SUSY decay have a reduced phase space and hence narrower distributions.

at any point in the space, on a grid as fine or as coarse as desired, using a single set of events from the metapoint. This is a useful additional validation of the reweighting because we are testing not only that the distributions are well reproduced, but specifically that they are well reproduced in the extreme corners of the phase space that are used to define BSM signal regions. This becomes a proxy for actually using the reweighted data in an Atlas analysis across multiple grid points, albeit without the added difficulties of detector effects and systematic variations.

This test also hints at potential uses of this technique not just for Atlas signal grids, but also for phenomenological workflows: many Gambit and Contur studies' main computational task is running the same set of collider analyses on freshly generated events at many points in the parameter space. We can imagine a scenario where the tool scans as normal for the first few iterations, before using the data obtained in the scan to train a CARL network and interpolate across the parameter space to obtain better-defined exclusion contours.

6.6.1 Technical steps

The CARL machinery already saves its DNN as an Onnx file after training is complete. Using the Rivet machinery for using `ONNXRunTime` to carry out inference introduced in Section 5.3, the CARL network can be loaded; and then, for each event, the relevant weights can be obtained by extracting the required inputs from the events.

Rivet has a very sophisticated handling of weights in HepMC files – analysis authors and users alike can remain blissfully unaware that their code is simultaneously running on hundreds or even thousands of weight-streams, with an almost imperceptible performance penalty. However, this machinery only applies to weights defined within the HepMC file: during the `init` stage of analysis, Rivet reads the first event to identify how many weights there are and what their names are. For the purposes of scanning an entire grid in one run, we need a separate weight-stream for each parameter point we are interested in; and these weights are generated on the fly based on the events kinematics (i.e. *after* they are read in).

One solution to this issue would be to use a two-pass system: write one Rivet analysis whose sole purpose is to obtain the CARL weights for each parameter point event by event; dump this information into some intermediate storage format (e.g. an ASCII csv file); then write a script to add this information to the HepMC files – recalling that HepMC files are typically $\mathcal{O}(10)$ GB in size and often stored in a compressed format – and finally run the "normal" Rivet analysis on the second pass. Such a solution would be slow and inefficient, both in computational time and human effort.

Another possible solution would be to adjust the Rivet weight-handling system to allow Rivet analyses to declare additional weights they would like added to those from the HepMC file during their `init` methods – perhaps booking weights in the same way that analyses presently book projections. A benefit of this approach would be that even though the weights are only calculated in one analysis, if other analyses are included in the same Rivet run, they would have access to the same weights, which could be very helpful if we were trying to scan a signal grid with a tool like Contur. However, although an interesting idea, and a possible avenue for future Rivet development, this solution was deemed too technically involved a project given the limited time frame.

Instead, we made use of several new features from version 2 of Yoda. Specifically, because histograms can now be indexed using arbitrarily many objects of almost any continuous or discrete type, we can add the 2D mass grid as an additional two dimensions to every data object in the analysis.

Working from the original Rivet implementation of Reference [177] introduced in Section 5.5 (an Atlas search for pair produced gluinos in a multi b-jet final state), all the zero-dimensional `Counter` objects that stored the event counts in the signal and control regions have been upgraded to `Histo<int, int>`, i.e. 2D histograms where the axes are the two SUSY masses[8]. Likewise, all 1D distributions plotted as `Histo<double>` (perhaps more familiar to Rivet users as under the typedef `Histo1D`) were promoted to `Histo<int,int,double>`, i.e. a 2D discrete histogram of 1D continuous histograms.

For the purposes of our test, the mass-grid (in terms of minima, maxima and spacing) was hard-coded into the analysis, but if desired it would be relatively easy to make this configurable at run time based on a command line argument or configuration file. The CARL weights were calculated as soon as enough physics information (jet kinematics, transverse momenta and so forth) had been extracted, and were stored as a member variable of the derived analysis class.

Unlike all the work so far, where we could consider distributions normalised to unity, here the distributions needed to be corrected based on the cross-section and luminosity. The correction to 139 fb^{-1} using the higher-order cross-section calculations (discussed in Chapter 5) requires the sum of all weights at each parameter point, so no events could be vetoed before the CARL weights were calculated and stored in a `Hist<int,int,double>`; these would then be used in the analyses' `finalise` method. Note that because the "destination" parameter point is an input to the network, the neural net had to be evaluated once per grid point. Though this was not a bottleneck in this use case, if in future work it became one – either due to much more complex network structures, much finer or more-dimensional signal grids, or much simpler analysis code – this could probably be partly by ameliorated using the batching capabilities of `ONNXRunTime`.

An analysis specific "fill" method was defined: given a histogram and a fill-value, and with access to all the CARL weights for the current event via the aforementioned analysis-owned weights histogram, the new "fill" function would automatically fill the entire grid. This kept the code very readable and minimised the changes with respect to the original code. The adjusted Rivet analysis can be found in Reference [242].

6.6.2 Results

Figure 6.12 compares the reweighted yields to those obtained by Rivet run on events generated specifically at that parameter point, for several regions of the analysis. Given the huge variation in expected counts, this is impressive performance – though we should acknowledge that the higher-order cross-section correction is helping here, too. As expected, performance is weakest along the compressed diagonal, for broadly the same reasons as in Section 6.5.3.

Performance is also slightly better for the cut-and-count signal regions than the NN-based signal regions; and perhaps even better still in the NN control regions. For the cut-and-count regions, this can be explained by the fact that the cut-and-count regions typically contain more events, and so enjoy better statistics. For the NN-based control regions, this is a little harder to explain, as these contain close to an order-of-magnitude fewer signal events than the NN SRs, but still perform much better. However, even though they contain fewer events, they are still sampled more from the bulk of the distributions[9]. This likely means that the neural net gets a much better estimate of the density ratio here due to all the surrounding data, than it would in the tails of the distributions.

As perhaps the most relevant cross-check for actual physics performance, Figures 6.13 and 6.14 compare the exclusion contours made using reweighted data against those made from data generated at each point individually. These plots combine results from all thirteen analysis regions significantly populated by the signal model, and reproduce what is ultimately the most important output of the analysis.

[8] The masses were recorded as integers for easier equality checking and to allow us to consider individual mass points: even with the increased efficiency of surveying a parameter grid using this method, studying a grid so fine-grained that it requires decimal-point precision on the sparticle masses would be a pointless endeavour.

[9] In a crude 1D analogy, this is like sampling a unit Gaussian between in [0,0.1] vs sampling it in [1,2].

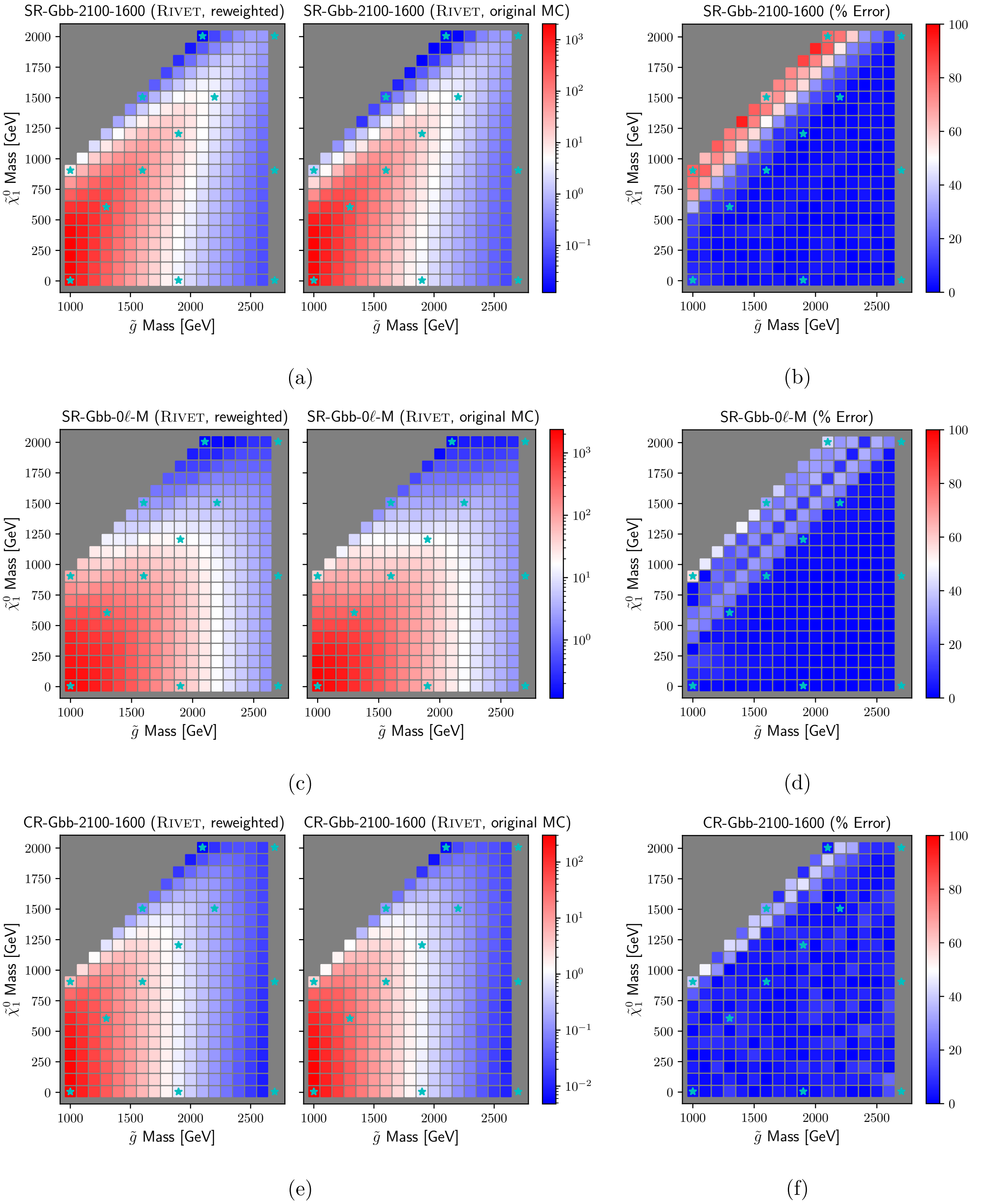
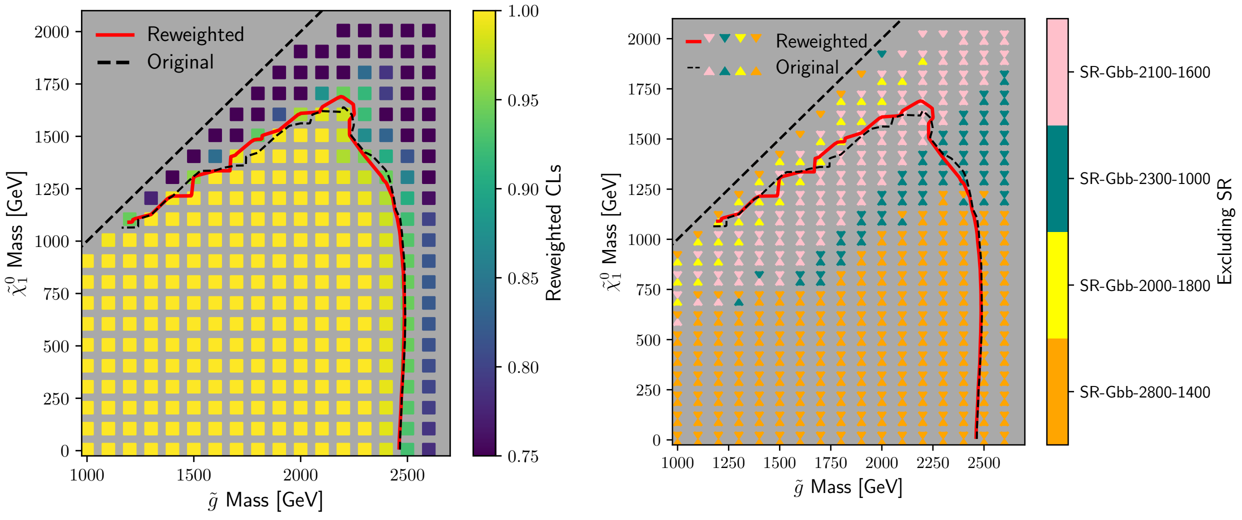


Figure 6.12: Comparison of the event count in each of the various regions, comparing the reweighted counts (left) to those obtained by event generation at that point (centre), with the absolute percentage error between the two relative to the original MC on the right. The parameter points used in the nominal sample (i.e. the "source" of the reweighting) are marked with cyan stars. One of the four NN signal regions (top), one of the three cut-and-count signal regions (middle), and one of the four NN control regions (bottom), are displayed, and they are each broadly representative of their type of region. The full set of thirteen regions is displayed in Appendix B. Performance is consistently worse along the diagonal.



(a)

(b)

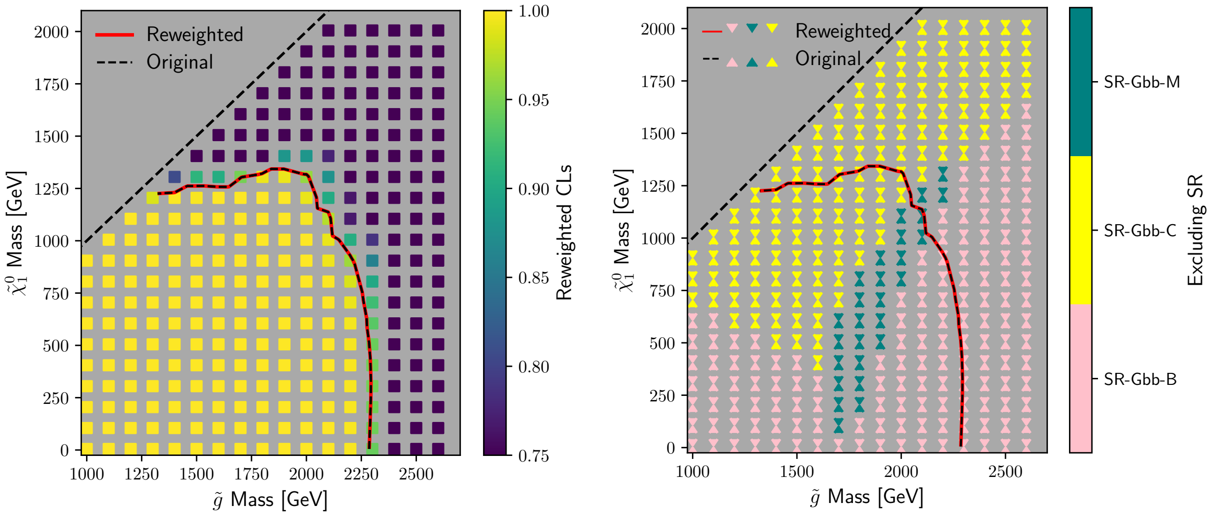
Figure 6.13: Comparing the exclusion contour obtained using the reweighted event counts for the NN-based regions to the "original" values obtained in Chapter 5 by running Rivet on the entire "true" signal grid. Which region is providing the exclusion at each point is shown for both cases in (b). The contour itself is very close indeed to the original target – indeed, the difference is much smaller than that between the Rivet and Atlas contours in Figure 5.8; and, with the exception of the under-utilisation of the "SR-Gbb-2000-1800" pool, the excluding region is almost always the same.
The "original" Rivet exclusion does not exactly match that in Figure 5.8 because we have re-interpolated it on the same coarser (100 GeV spaced, not 50 GeV spaced) grid that the reweighting ran on, in order to ensure an apples-to-apples comparison. For consistency with the plots in Chapter 5, we use the Contur CL_s convention, even though only `pyhf` was used in this case.

The reproduction of the contours is excellent for the NN regions, and effectively perfect for the cut-and-count regions – where we not only reproduce the exact exclusion contour, but do so based on exactly the same signal region at every single parameter point, even close to the diagonal where Figure 6.12d had suggested there might be some discrepancies.

For the NN-based exclusion contour, the reproduction is almost perfect away from the diagonal (as we would expect from Figure 6.12), both for the exclusion contour itself and for which region defines it. In the compressed region close to the diagonal, there is some mismatch in the excluding region, suggesting a consistent under-estimate in the "Gbb-2000-1800" bin; however the exclusion contour itself is still consistently within 100 GeV of the original.

It should not be underestimated how promising these results are. Although the network used in this particular test was trained on 11 nominal and 29 alternate points, which would be a lot of data for Atlas production (though still smaller than, for example, the 169 points used in the original study), it is important to remember that the network barely underwent any hyperparameter optimisation, and it is likely it would work just as well with fewer training points given only small additional improvements required elsewhere. The network also had no b-jet inputs, even though b-jet related cuts are very important to this analysis.

The fact that the reproduction of likelihoods appears to be a task that CARL is carrying out particularly succesfully also bodes very well for possible integration into phenomenological tools: even if the 11-nominal and 29-alternate points that went into this training are on the large side for Atlas production requests,



(a)

(b)

Figure 6.14: Comparing the exclusion contour obtained using the reweighted event counts for the cut-and-count regions to the "original" values obtained in Chapter 5 by running Rivet on the entire "true" signal grid. Which region is providing the exclusion at each point is shown in (b) for both cases. The contours themselves are visually identical; and the excluding region is the same at every parameter point.
As in the preceding figure, the "original" Rivet exclusion is not identical to that in Figure 5.11, because we have re-interpolated it on the same coarser (100 GeV spaced) grid that the reweighting ran on, in order to ensure an apples-to-apples comparison.

these are very small numbers compared to scans carried out with Gambit or Contur. The next step in implementing CARL in that context would be testing on three- or more-dimensional models, potentially using multiple analyses.

Chapter 7

Contur, Rivet and Yoda in Gambit

7.1 Philosophy and hurdles

In principle, the populations of signal regions of dedicated BSM searches and the populations of histogram-bins in unfolded SM measurement analyses should not be strongly correlated. While a handful of systematics such as luminosity are shared, these are rarely dominant; and BSM signal regions typically (albeit not always) contain an order-of-magnitude (or even multiple orders of magnitude) fewer events than fiducial phase-space measurements. Therefore, it should be possible to directly combine the likelihoods from searches and measurements at the LHC by summing log-likelihoods – in an analagous way to how Gambit already directly combines searches from Atlas and CMS; or how Contur directly combines measurements from Atlas, CMS and LHCb.

As mentioned in Section 4.3.2, Gambit as of 2022 did not have a functioning interface to Contur: indeed, likelihoods from SM cross-section measurements at Atlas and CMS had not been used in any previous BSM global fit. To do this, although the likelihood calculation itself will come from Contur, we also need access to Rivet and Yoda in order to provide the Yoda histograms that Contur takes as input.

Figure 7.1 paints in the broadest possible brushstrokes the way the three new software packages need to interact: HepMC events are produced internally by Gambit; Rivet analyses these, producing Yoda objects to summarise the results, and these are finally used by Contur to obtain a likelihood.

7.2 Technical implementation and validation

7.2.1 File-free Yoda files

In normal operation, Rivet writes histograms to Yoda files on disk, and Contur reads these files back in to obtain Yoda objects. In the HPC context that Gambit runs in, with hundreds or even thousands of threads running at the same time, this is unacceptable for performance reasons.

Instead, Yoda objects need to be handed between Rivet and Contur in memory. Naïvely, this may appear simple: both Contur and Rivet already use Yoda types internally. However, Contur is a Python based package, so expects Python objects from Yoda's Python API, whereas Rivet produces Yoda `C++` objects; and does not give access to their Python wrappings in its Python interface. To add complexity, the two programs will not interact directly – objects will need to be passed through Gambit's internal "pipes" system. The strict modularity of Gambit requires that backends cannot directly pass data to each-other: instead, the dependency resolver pipes data between dependencies as needed using the pipes system.

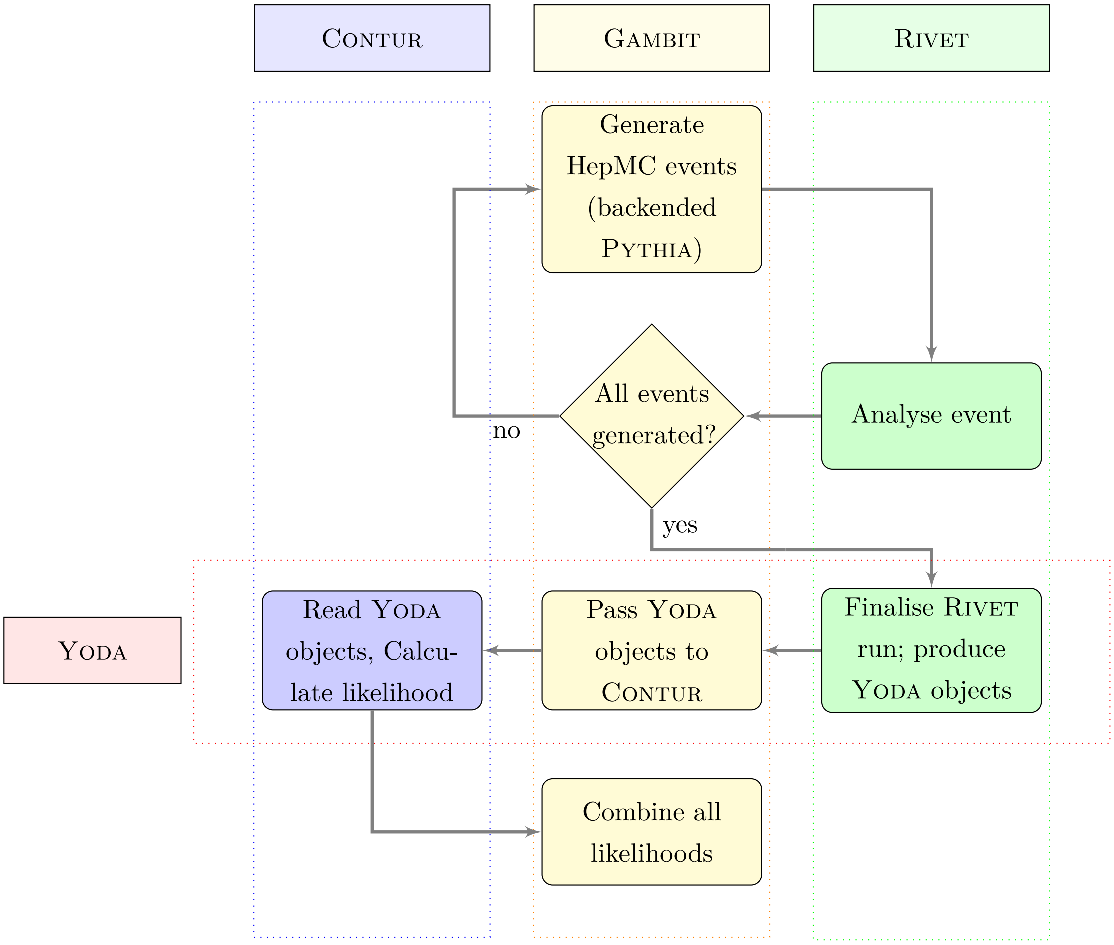

Contur
Gambit
Rivet
Generate HepMC events (backended Pythia)
Analyse event
All events generated?
no
yes
Yoda
Read Yoda objects, Calculate likelihood
Pass Yoda objects to Contur
Finalise Rivet run; produce Yoda objects
Combine all likelihoods

Figure 7.1: Figurative description of how we would like Gambit, Rivet, Yoda, and Contur to interact. All possible technical difficulties and pitfalls have been removed from the diagram for illustrative purposes.

As Yoda files are most commonly produced in ASCII format, the solution arrived at was to write Rivet output to a `std::stringstream`, a C++ STL type that consists of a buffer of characters. There are two obvious benefits to this format. Firstly, there is an easily convertible Python equivalent in the form of the `StringIO` class from the (standard) Python `io` library, and conversion between the two can be easily carried out in the cython code that defines the Yoda Python API. Secondly, it inherits from the same base-class – `std::iostream` – as C++ `filestream` objects, so the amount of new code that needs to be written (and maintained) should be minimal, as the `filestream` and `stringstream` objects can be treated in a very similar manner, with many functions taking pointers or references to the base `iostream`, `istream` or `ostream` class.

As a simple test-bed for handling Yoda objects via stream, we developed a Python script that was intended to carry out the entire Rivet→Contur process in one file, without any file-writing. The file ran Rivet using the `run` class from the Python API, and outputted results to a Yoda StringIO object, which was read in by Contur to calculate the likelihood[1].

In order to implement this, the Yoda Python API simply needed an explicit new method to read from `stringstream`s. The changes to Contur were more extensive, and included restructuring the main function so it could directly take a dictionary of arguments, rather than just an `argparse` object. This made it much easier to steer from an external code.

In addition to relevant pre-existing arguments (such as `quiet`, to reduce the level of printing while

[1] This script is publicly available via Reference [47], and is also reproduced in Appendix C

running), the input dictionary could also contain new arguments specific to running on StringIO objects. Most obviously, this includes the `YODASTREAM` argument, which contains the Yoda file Contur should run on, in StringIO form. The new `LOGSTREAM` argument allows the Contur log to be directed to any stream. For usage in Gambit, this meant the Contur log file could be integrated into the ColliderBit log, which simplified debugging, and also prevented unnecessary disk I/O from writing the Contur log to file.

In normal running, Contur prints results to the terminal and also saves them to an output file; this is less useful when Contur is being used by another program. Therefore another important new argument was `YODASTREAM_API_OUTPUT_OPIONS`, which allows users to specify which statistical outputs they would like returned, in memory, from Contur. For Gambit, this was the LLR (and possibly also the per-pool LLR and other meta-data); though $\mathrm{CL_s}$ is also available. New methods were added in Contur to make returning log-likelihood based statistics simpler.

While these changes were made specifically with the Gambit use-case in mind, they could also be used in other contexts where researchers would like to interact with Contur more programmatically[2]. As this mode of operation is somewhat different to how Contur normally runs, new CI tests were added to the Contur suite in order to ensure its stability.

7.2.2 Integration into the Gambit build system

Contur

Pure Python backends are relatively simple to integrate into Gambit. A backend convenience function is defined that explictly uses the Contur types and method in `C++` code, using Gambit's embedded Pybind11 interpreter [263]. This backend convenience function is then used by the `Contur_measurements_from_stream`[3] function, which belongs to the `LHC_measurments` capability, which can be requested in the steering yaml file like any other Gambit capability. An example of how to include Rivet and Contur in the steering file can be found in `ColliderBit_CMSSM.yaml`, packaged in Gambit from version 2.4 onwards.

A utility function was also provided for getting the list of available analyses from Contur – to help steer Rivet – and other small changes were also made, such as the ability to redirect Contur's log, in order to avoid additional writing to disk.

Rivet

As discussed in Section 4.3.2, for Gambit to be interfaced to a `C++` backend, the backend in question needs to go through the occasionally complex procedure of being BOSSed. The only core class of Rivet that was BOSSed was the `AnalysisHandler` class, along with some global functions for keeping track of analysis library objects. To minimise the need to patch future Rivet release for compatibility with BOSS, Rivet was updated to remove calls to `exit`, which were replaced with `exception`s that would eventually be caught by the even higher-level Rivet `Run` class, which is not required by Gambit.

Yoda

Instead of being backended, Yoda was integrated into Gambit's `contrib` directory. This comes with the downside of Yoda having to be built every time ColliderBit is built – and would be a totally impractical technique to apply to every one of Gambit's hundreds of backends. However, because we know Yoda exists before compiling ColliderBit, there is no need to BOSS Yoda to access its members and methods.

[2] An example of how to go about this is the script in Appendix C.

[3] `Contur_measurements_from_file` was also included for debugging purposes but, as described earlier, this is highly unperformant and should not be used in production.

7.2.3 Gambit workflow and the single-thread problem

Figure 7.2 extends the flowchart in Figure 4.5 to include an outline of how Rivet and Contur functions interact with other parts of Gambit: Rivet analyses take HepMC events from ColliderBit's backended Pythia implementation, and then Gambit feeds the Yoda output to Contur. One thing that immediately stands out is the fact that Rivet analyses – unlike their ColliderBit counterparts – do not run concurrently. The versions of Rivet first interfaced to Gambit were in the `3.1.X` series, within which the analysis handler was a single global object.

In order to avoid corrupted results or crashes, a `#pragma omp critical` block was placed around the call to `AnalysisHandler->analyze`, ensuring multiple threads could not analyse events simultaneously, and thus that the analysis handler only processed one event at a time. However, introducing a system of locks which requires communication between threads introduced a significant performance overhead – demonstrated indirectly in Figure 7.18. This meant that production runs involving Rivet `3.1.X` could only practically run with a single `OpenMP` thread – partly compensated for by also increasing the number of MPI processes. Section 7.5 will lay out how this problem was fixed, but this was not ready for production of any of the physics results displayed later in this chapter.

7.3 Physics use case: Gambit gravitino study

7.3.1 Background

Building on previous Gambit scans for electroweakinos [202]; the Gambit community studied a similar parameter space but with the addition of a light gravitino [203]. The resultant model was based on GMSB, as described in Section 1.4.2. Gambit scanned a 4D parameter space consisting of three mass parameters M_1, M_2, and μ; and the dimensionless $\tan\beta$ angle[4]. The gravitino mass was fixed at 1 eV, the gravitino sufficiently decoupled from the other sparticles that even significant variations would not substantially alter the physics.

Likelihoods during the scanning stage were evaluated using ColliderBit's extensive library of LHC SUSY searches. SpecBit and DecayBit were respectively used for spectrum-generation and computing electroweakino decays; and ScannerBit – using the differential evolution sampler Diver 1.0.4 – controlled the scan. No other Bit's were used for likelihood computation. The scan consisted of approximately 300 000 successful parameter points and, at each of these, 16 million events were generated with Pythia 8.2.

The results after this main scanning step are shown in Figures 7.3, in the ($m_{\tilde{\chi}_2^0}$, $m_{\tilde{\chi}_1^0}$) plane. There is a small favoured region along the $m_{\tilde{\chi}_2^0} = m_{\tilde{\chi}_1^0}$ diagonal, with a best fit point at $m_{\tilde{\chi}_1^0}$ = 170 GeV, $m_{\tilde{\chi}_2^0}$= 179 GeV and $m_{\tilde{\chi}_1^\pm}$ = 177 GeV, the surroundings of which we will describe as the "low-mass favoured region". There is also a secondary, local maximum at approximately $m_{\tilde{\chi}_1^0}$ = 320 GeV, the "high-mass favoured region".

The compressed region avoids many of the strong constraints in the rest of the parameter space because the branching fraction to $\gamma\tilde{G}$ is lower; and the support over the SM is obtained by fitting simultaneously to small excesses in the Atlas $b-$jets + E_T^{miss} search [264] and Atlas [265] and CMS [266] leptons + E_T^{miss} searches.

7.3.2 Contur strategy

Because of the issues with multi-threaded processing with the Rivet `3.X` series described in Section 7.2.3, it was not computationally practical to use Rivet and Contur in the initial scans of the parameter space.

[4] These are parameters from the Neutralino mass matrix: for a full formal definition, see Reference [202].

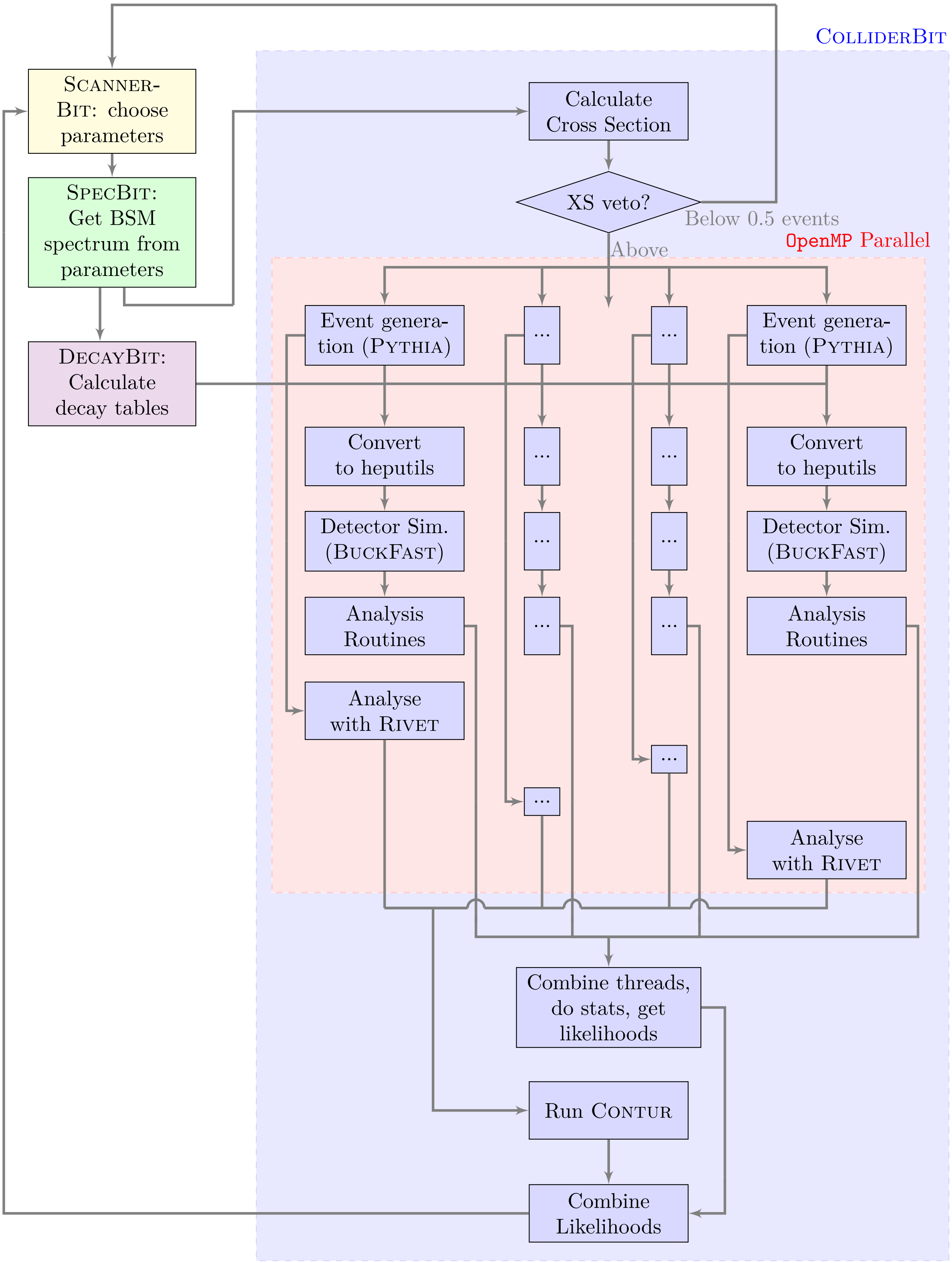


Figure 7.2: Extending Figure 4.5 to include Rivet. As before, the flowchart is illustrative and not comprehensive and the vertical axis implies concurrency within the `OpenMP` block exclusively. Note that Rivet does not need a thread combination block, because each Rivet thread uses the same global analysis handler.

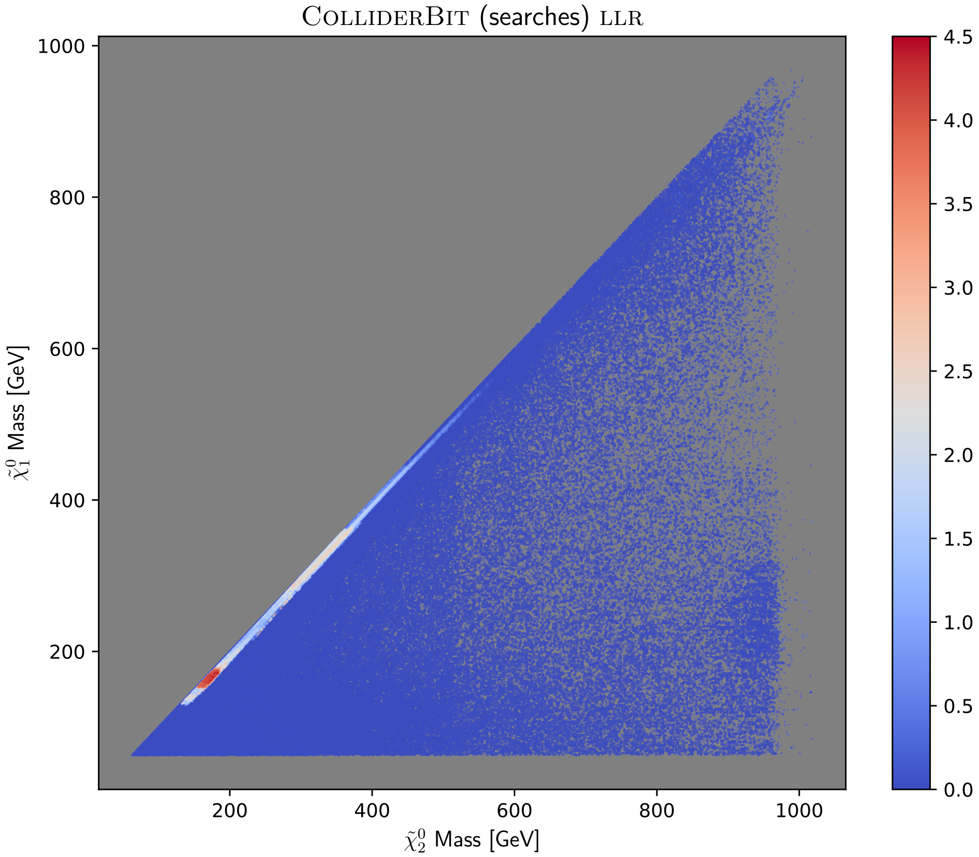


Figure 7.3: The log-likelihood ratio (defined by equation 4.42) from ColliderBit-implemented LHC searches. The colour-bar minimum is capped at zero in order to best illustrate the two favoured regions along the diagonal.

Instead, Rivet and Contur were used as a post-processing step, providing additional collider likelihoods to the approximately 300 000 most favoured points, based on LLR's from the ColliderBit implementations of 13 TeV Atlas and CMS SUSY searches. This workflow also played to another strength of Contur: because LHC measurements typically have much larger acceptances than search signal regions that are focussed on an extreme phase-space, while ColliderBit searches needed to be run on as many as 16 million events per parameter point, good results can be obtained with Contur with only 100 000 events. Although OpenMP parallelisation was impossible, separate processes could still run `AnalysisHandler`'s in parallel, so some parallelisation with OpenMPI was possible.

7.3.3 Contur results

The Contur log-likelihood contribution across the parameter space is shown in Figure 7.4. The exclusion is strongest at low mass points, because the NLSP pair-production cross-section is higher. The range of Contur pools that contribute to the exclusion across the parameter space is illustrated in Figure 7.5a. These can largely be explained by examining the dominant electroweakino decay modes (Figures 7.5b and 7.5c). For example, where $\tilde{\chi}_1^0 \to \gamma\tilde{G}$ has the largest branching fraction – in the bulk, for NLSP masses below 200 GeV – Contur exclusion is dominated by photon final state measurements in the `ATLAS_13_LL_GAMMA` pool [267, 268]; and for the majority of the region where the dominant NLSP decay is to $Z\tilde{G}$ – also in the bulk, but for NLSP masses above 200 GeV – the dominant pool is `ATLAS_13_METJET`, which consists of a

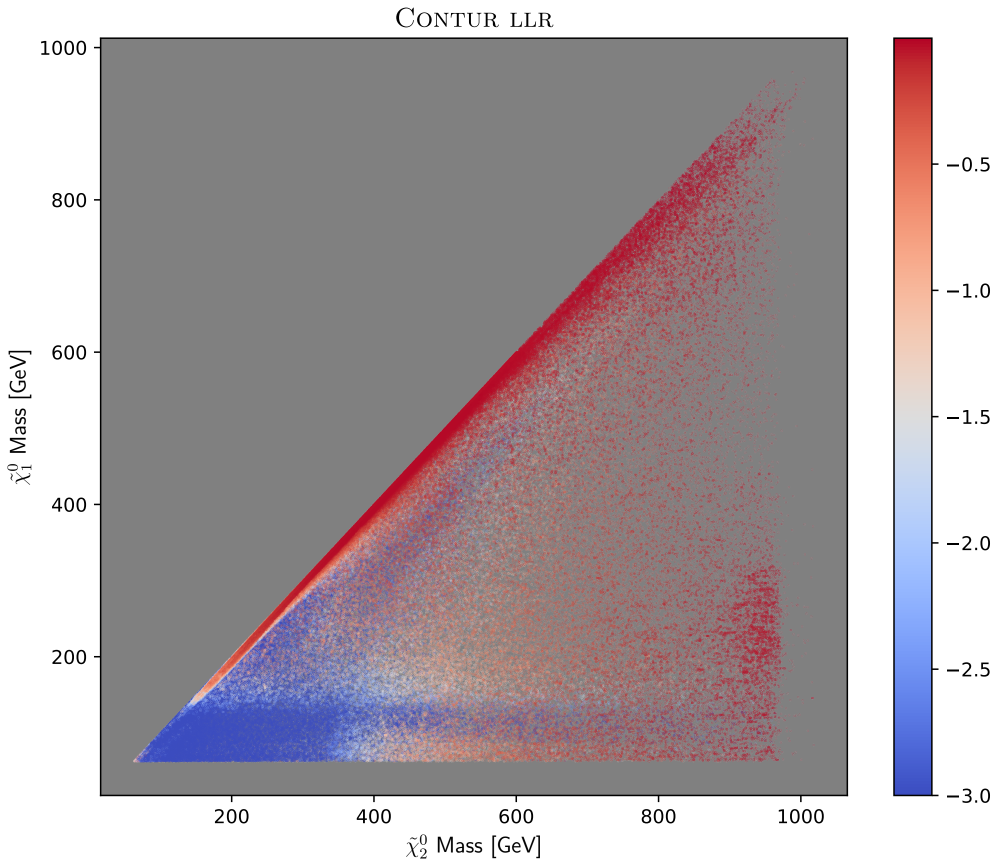


Figure 7.4: The log-likelihood ratio from SM measurements as calculated by Contur: this is still defined by equation 4.42, but with the individual contributions coming from equation 4.40, as laid out in Section 4.3. The exclusion is strongest at low masses. Where points overlap, the point with the highest combined Contur and ColliderBit likelihood is put on top.

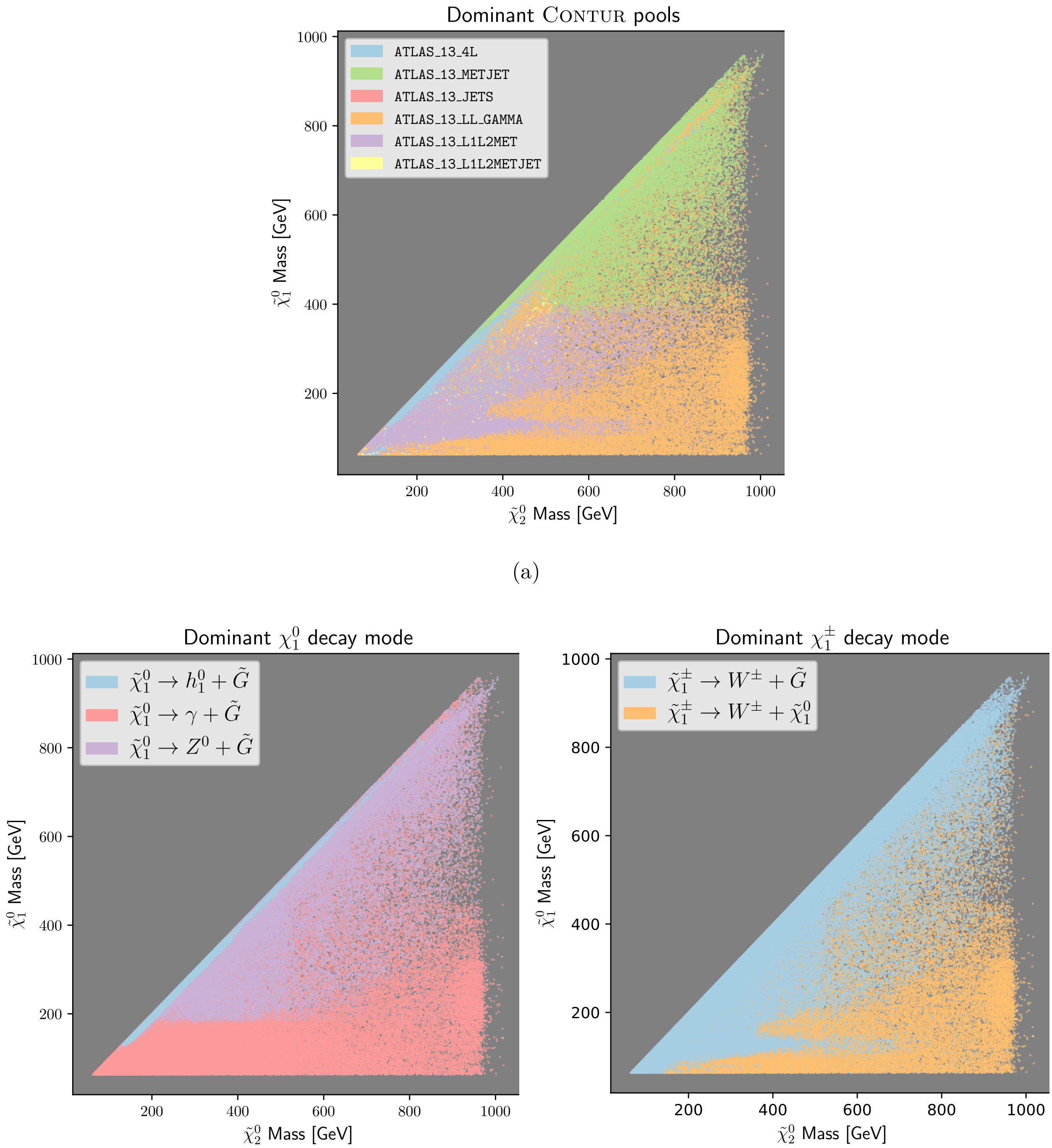
single analysis sensitive to Z bosons [269].

Interestingly however, the horizontal "spike" in exclusion at (200 − 700 GeV, 100 GeV) of Figure 7.4 is dominated by Atlas WW production measurements [270], which cannot be explained by the dominant NLSP decay to $\gamma\tilde{G}$. Instead in this region, Contur is likely most sensitive to the pair production of the lightest chargino – even though it is more massive than the NLSP – and its subsequent direct decay to $W^{\pm}\tilde{G}$ which as shown in Figure 7.5c, occurs in this specific region.

The Contur exclusion in only those regions that ColliderBit had already identified as most likely – the "low-mass" and "high mass" maxima along the compressed diagonal – is shown in Figure 7.6. In these regions the Contur exclusion is dominated by the Atlas 4-lepton measurement [271], which includes $h \to 4l$ regions that are sensitive to the $\tilde{\chi}_1^0 \to h\tilde{G}$ decay – the largest contribution to the NLSP branching width here. There is little impact on the "high-mass" region, but some constraint on the "low-mass" region.

7.3.4 Final combined results

When combined with ColliderBit likelihoods, the effect of the stronger SM exclusion at low masses is to reduce the LLR of the "low-mass" by approximately one unit while leaving the "high-mass" region largely unchanged, bringing two maxima much closer to each-other. This is illustrated in Figure 7.7, and again even more clearly in the projection onto the $m_{\tilde{\chi}_1^0}$ axis in Figure 7.8.



(a)

(b)

(c)

Figure 7.5: The most excluding Contur pool (a), presented alongside the leading $\tilde{\chi}_1^0$ (b) and $\tilde{\chi}_1^\pm$ (c) decay modes, which are essential in understanding (a). For example, the exclusion along the diagonal from the `ATLAS_13_4L` pool (which includes $hh \rightarrow 4l$ regions) can be explained by the equivalent region in (b) where the NLSP decays predominantly to a Higgs boson.

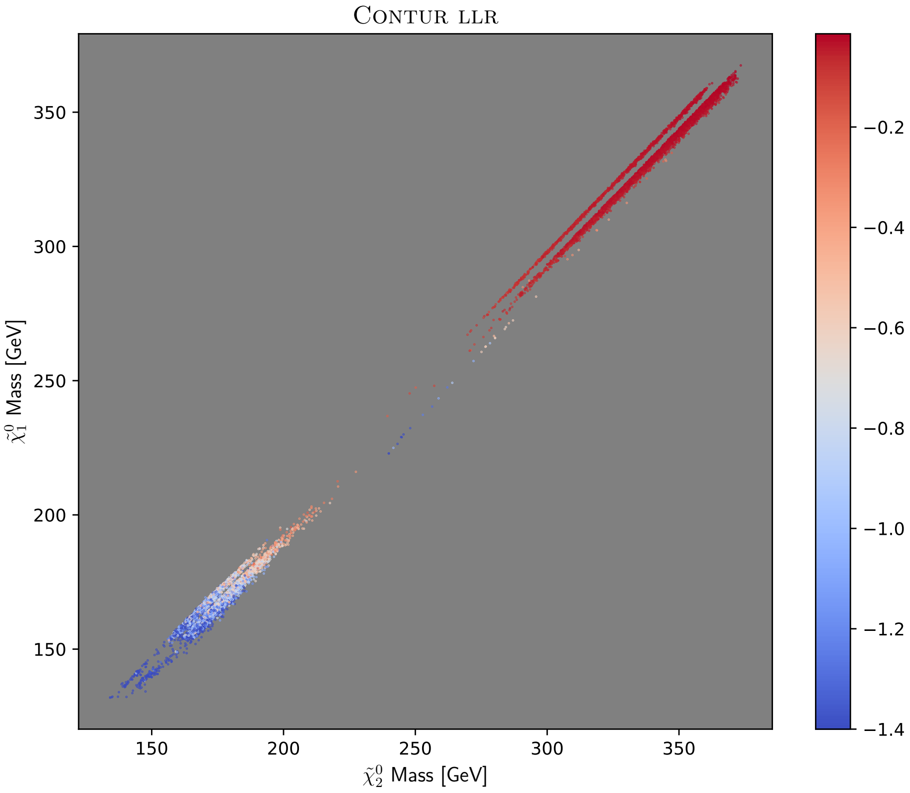


Figure 7.6: The log-likelihood ratio from SM measurements as calculated by CONTUR, plotted only for the approximately 10 000 parameter points with a COLLIDERBIT search-based LLR greater than 1.8. The exclusion is more significant in the "low-mass favoured region".

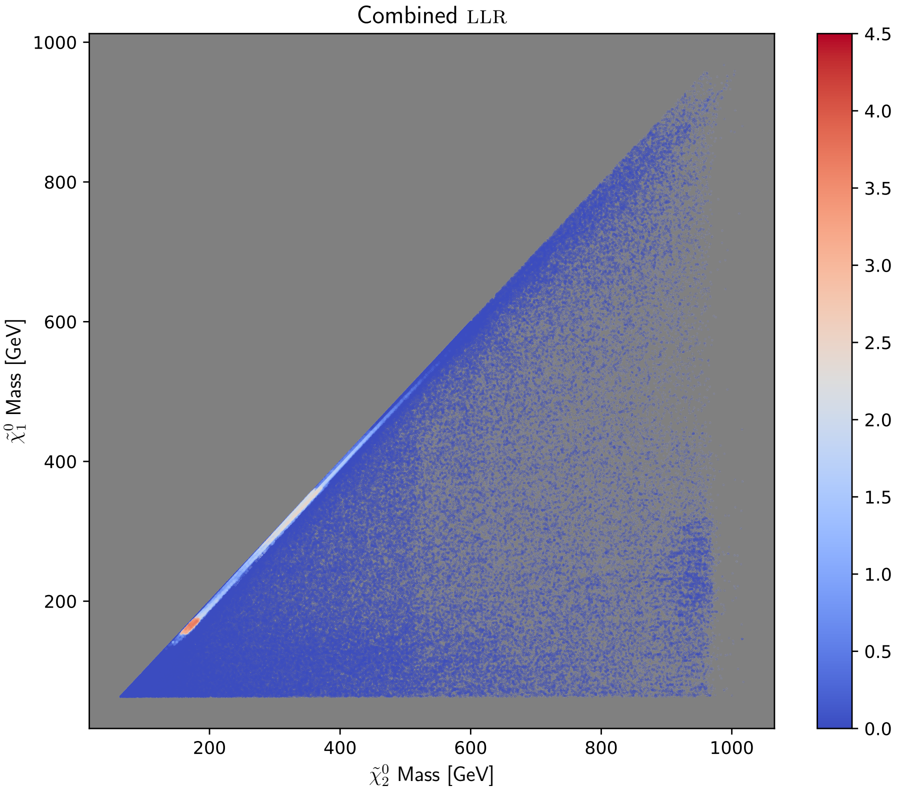


Figure 7.7: The total LLR after adding the contribution from LHC measurements. For easier comparison to Figure 7.3, the colourbar uses the same scale; though the comparison is clearest in Figure 7.8, where the colourbar axis becomes the y-axis.

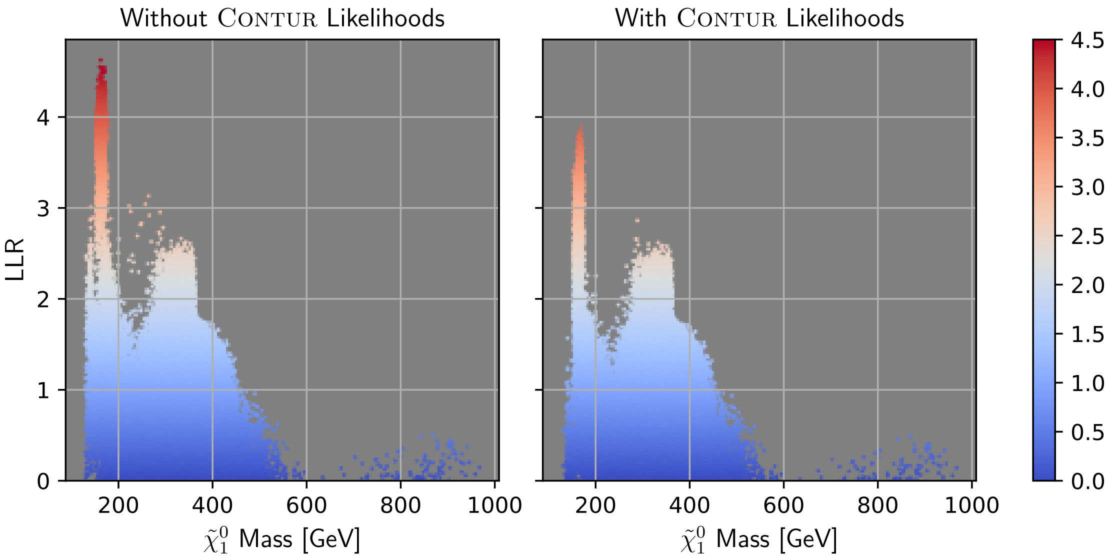


Figure 7.8: The total LLR before and after adding the contribution from LHC measurements, projected onto the $m_{\chi_1^0}$ plane.

The continued presence of – in several cases significantly – unexcluded points, primarily along the uncompressed diagonal, suggests that there remain areas of the electroweakino parameter space that warrant further examination from the LHC experiments. It would be enlightening to repeat this study with the full complement of full Run 2 searches and measurements as they are published, as these may provide additional constraints.

These results also show there is complementarity between search and measurement exclusions, and that there are clear and tangible benefits to including LHC measurements in global fits going forward, both within GAMBIT and for others interested in surveying the unexplored expanses of many-dimensional, less-simplified SUSY models. Perhaps one obvious potential external beneficiary could be the ATLAS pMSSM scanning effort, which should be able to benefit from RIVET and CONTUR already being included in the ATLAS software stack, substantially lowering some of the technical barriers to including CONTUR likelihoods.

7.3.5 8 TeV study

The low-mass favoured points (below 200 GeV) are sufficiently light that there would have been non-negligible NLSP production during the 8 TeV portion of LHC Run 1. Although COLLIDERBIT does contain a significant number of LHC Run 1 SUSY searches from ATLAS and CMS, given the lower sensitivity of most Run 1 searches, the computational cost-benefit analysis suggested that generating several million 8 TeV events, even at a select subset of more interesting points, was unlikely to be beneficial.

However, because LHC measurements have much higher acceptances than LHC searches, the number of events that need to be generated to obtain a meaningful likelihood are an order-of-magnitude lower. Therefore, as a supplementary cross-check, we generated 100 000 events at the 100 points with the highest (uncapped) log-likelihood ratio. As these all fell in the low-mass favoured region – with $m_{\tilde{\chi}_1^0} \in (160 \text{ GeV}, 175 \text{ GeV})$ – we also generated the twenty most favoured points from the local maximum in the $m_{\tilde{\chi}_1^0} \in (280 \text{ GeV}, 360 \text{ GeV})$ range.

The results of this test are shown in Figure 7.9. As expected, because the Run 1 production cross section of NLSPs is much smaller in the high-mass region, the addition of 8 TeV CONTUR results is negligible there. In the low-mass favoured region, there is a minor increase in exclusionary power, primarily from ATLAS 8 TeV

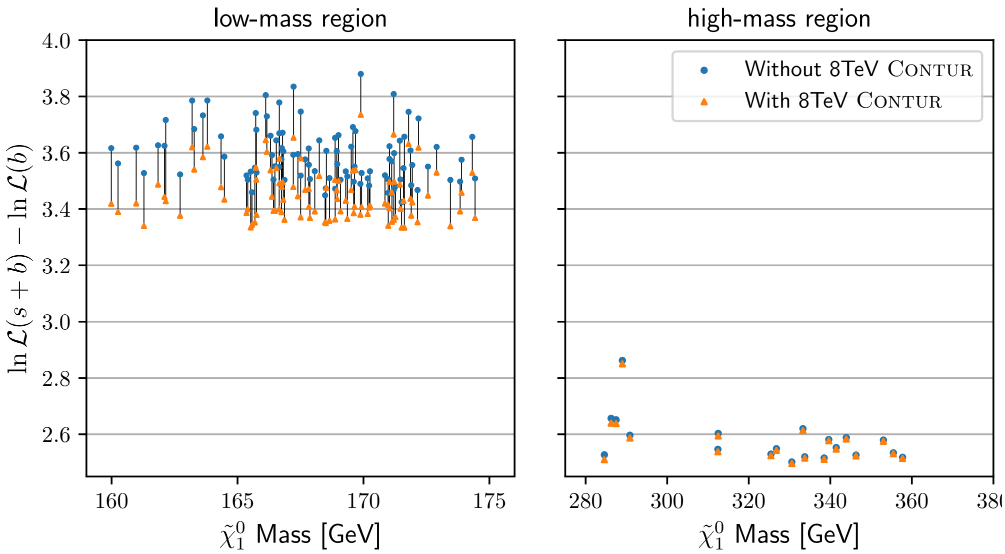


Figure 7.9: The impact of adding in 8 TeV likelihoods obtained by Contur on the 100 most favoured data points (left panel) and the 20 most favoured data points in the higher-mass local maximum. The impact is more significant at lower masses due to the higher production cross-section at 8 TeV.

Higgs to diphoton measurements [272]: but this is still modest compared to the total combined likelihood around the best-fit points. Because of this, the 8 TeV event generation and exclusion was not propagated to the whole fit as an additional post-processing step.

Contur pool overlap

As an aside, during the debugging of very strong over-exclusion for a small subset of points in the low mass region of Figure 7.9, it was discovered that there was a narrow overlap between the `ATLAS_8_GAMMA` and `ATLAS_8_GAMMA_MET` pools in Contur, which allowed a very small number of events to contribute to the likelihood twice. To ensure similar problems do not occur again as more analyses are added, using a method like Taco [273] – a computational method for identifying expected maximally performant combinations of mutually orthogonal SRs – to check the full set of histograms used by Contur for unexpected overlaps – perhaps even integrated into the Rivet or Contur CI – could be beneficial.

7.4 Physics use case: preliminary VLQ scans

7.4.1 Motivation

Given their relevance to the rest of this thesis, and the fact that Contur is known to constrain them [54], VLQs would also be an interesting model to study with Gambit. Even before considering extensions that may contain a scalar DM candidate (discussed in Section 1.3.4), there are still opportunities to interface with other Gambit modules, such as FlavBit.

Before studying VLQs with ColliderBit, the most significant task that needs to be completed is writing ColliderBit implementations of direct Atlas and CMS searches for VLQs[5]. However, another interesting preliminary study is to examine whether any of the searches currently implemented in Gambit – almost

[5] As a contribution to this effort, I wrote and validated a ColliderBit implementation of Reference [274], an Atlas search for single-production of VLTs decaying to Zt final states.

exclusively SUSY searches, of which a significant majority focus on electroweakino pair production – have any sensitivity to VLQs, and could provide useful additional likelihood contributions in the full scan.

A point frequently made by those defending the continued importance of SUSY searches within the LHC program is that the final states and signatures of SUSY (including R-parity violation, to allow for single-production) are so varied, that any comprehensive set of SUSY searches – such as those carried out by Atlas and CMS during LHC Runs 1 and 2 – should be able to exclude (or detect) many other types of new physics [275]. The short study in this section will not comprehensively validate or invalidate this claim, because ColliderBit contains only a subset of LHC SUSY searches and VLQs are only one non-SUSY BSM model among hundreds: but it will be a (somewhat rare) opportunity for this claim to be confronted by real data.

Because ColliderBit (primarily) contains searches for R-parity conserving SUSY, we do not expect it to be sensitive to single-VLQ production; therefore, we will concentrate on pair-production. This allows for better comparison to the Contur study, which also only considered pair-production; and simplifies event generation, by restricting us to $2 \to 2$ processes.

The natural comparison to this study is the Atlas Run 2 VLQ pair-production combination, which studied both $B\bar{B}$ and $T\bar{T}$ production. Notably, all the analyses used in this combination considered only the first 36.1 fb^{-1} of LHC Run 2 data; and therefore it is likely a full Run 2 combination would provide even more sensitivity. There is a more recent full Run 2 CMS combination [43], but this only considered $B\bar{B}$ production.

7.4.2 Event generation

To generate VLQ events inside Gambit, we used Gum, introduced in Section 4.3.2. Our input to Gum is the FeynRules file discussed in Section 1.3, based on the Lagrangian in equation 1.10; but with internal conversions to allow constants of the theory to be provided in the convention defined by equation 1.9 (i.e. ζ_i and $\xi_{W/Z/h}$).

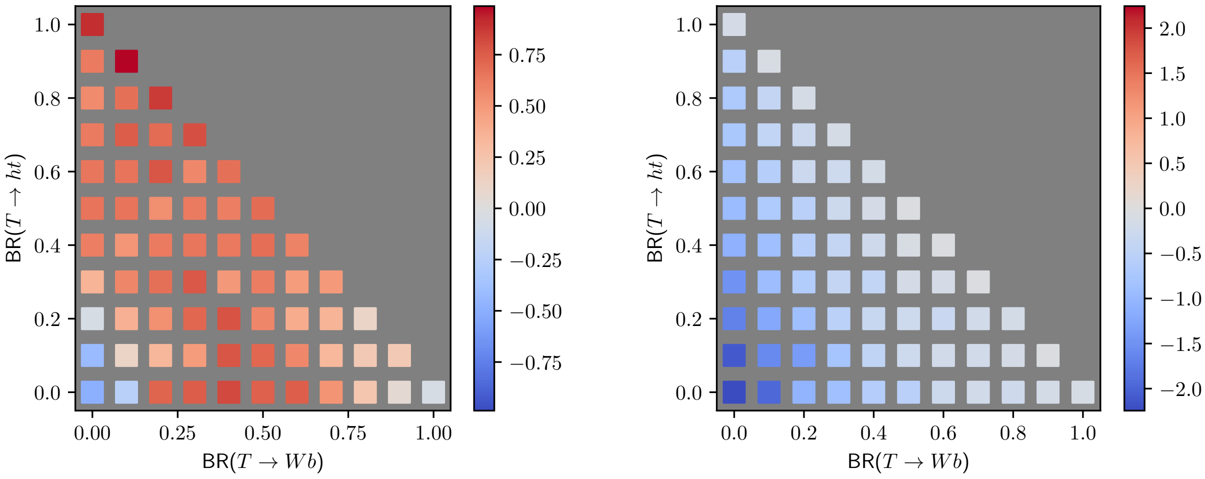
Following the normal procedure, Gum processed the FeynRules file to generate a UFO file; and then provided MadGraph5_MC@NLO with that UFO file in order to write matrix-element code that enabled event generation to be carried within Pythia. Unfortunately, initial tests showed that partial widths for the $T \to Zt$ and $T \to Wb$ calculated by CalcHep[6] disagreed strongly with the expected values. This was noticed because the branching ratios implied by ξ_W and ξ_Z were not obtained; and the error confirmed by comparing to the decay widths calculated by FeynRules. Both FeynRules and CalcHep consistently agreed on the partial width of $T \to ht$.

To work around this issue, the analytic expressions for decay width produced by FeynRules in the UFO file were coded directly into DecayBit, circumventing the necessity of using CalcHep. To validate the new set up, the Gambit decay-table entries were checked against MadGraph5_MC@NLO+MadSpin at a series of mass and coupling points. In addition, a customised Rivet analysis was written which would walk through the event record to ensure that events were being produced proportionally to the stated decay widths.

7.4.3 Preliminary results

For the purposes of this small preliminary study, we generated only pair-produced $T\bar{T}$ events across the entire branching-ratio plane at 1.3 TeV, which is approximately the largest mass that the Atlas combination could

[6] In fairness, the origin of the discrepancy has not yet been identified and it could also originate in either the Gum code that feeds into CalcHep, or the DecayBit code that is responsible for interpreting CalcHep's output.



(a)

(b)

Figure 7.10: The (a) uncapped and (b) capped log-likelihood ratio from ColliderBit SUSY searches for VLT pair production at 1.3 TeV. The contributions, whilst not amounting to an outright exclusion, are significant enough to warrant inclusion in future Gambit studies. Note the the two colour-bars are not the same, in order to make the plot on the left clearer.

exclude for all BRs [62]. The set of ColliderBit analyses was the same as for the gravitino study in the previous section. We will discuss some of the more interesting contributions below, but for a full description of all the ColliderBit analysis implementations, see Reference [203].

Both the full log-likelihood ratio and the capped log-likelihood ratio[7] for 1.3 TeV VLTs are shown in Figure 7.10. While the LLR is not sufficient for us to begin discussing exclusion of the model, the contributions across the majority of the plane are clearly large enough that it would be worth including SUSY ColliderBit analyses in future Gambit studies on VLQs.

The two most significantly contributing analyses – whose likelihood contributions are illustrated in Figure 7.11 – are the Atlas search for supersymmetry in four-lepton final-states [265], which provides the majority of the positive LLR contribution; and the CMS search for new physics in events with jets and two same-sign or at least three leptons [276], which provides the majority of the exclusion.

The Atlas search targetted pair production of electroweakinos, in final states with four leptons. The models considered include RPV-models where the electroweakinos are still pair-produced, but are permitted to decay to SM particles. One signal region targetting these final states was the $\texttt{SR0}_{\mathrm{breq}}$ region, the requirements for which included at least one b-jet and a veto on lepton-pairs with an invariant mass close to the Z-mass. Crucially for this study, because it targetted decaying RPV LSPs, there was no minimum cut on missing transverse momentum. The LLR from this region provided effectively all of the LLR contribution from the analysis as a whole, shown in Figure 7.11a. $1.19^{+0.30}_{-0.28}$ SM events were predicted for this region, but three were observed, explaining why a relatively large positive LLR can be obtained. That the LLR is largest in the top-left, at high BR($T \to ht$) makes sense: the leptonic-Z veto supresses $T \to Zt$ events (though off-shell or hadronically decaying Z-bosons may still be selected); and if both VLTs decay as $T \to Wt$, then it is impossible to attain the preselection requirement of four charged leptons.

The CMS search is optimised for several different SUSY models for squark and gluino pair-production.

[7] I.e. capping the LLR contribution from any analysis at zero, on the basis that anything that supports BSM over the SM must be a random fluctuation.

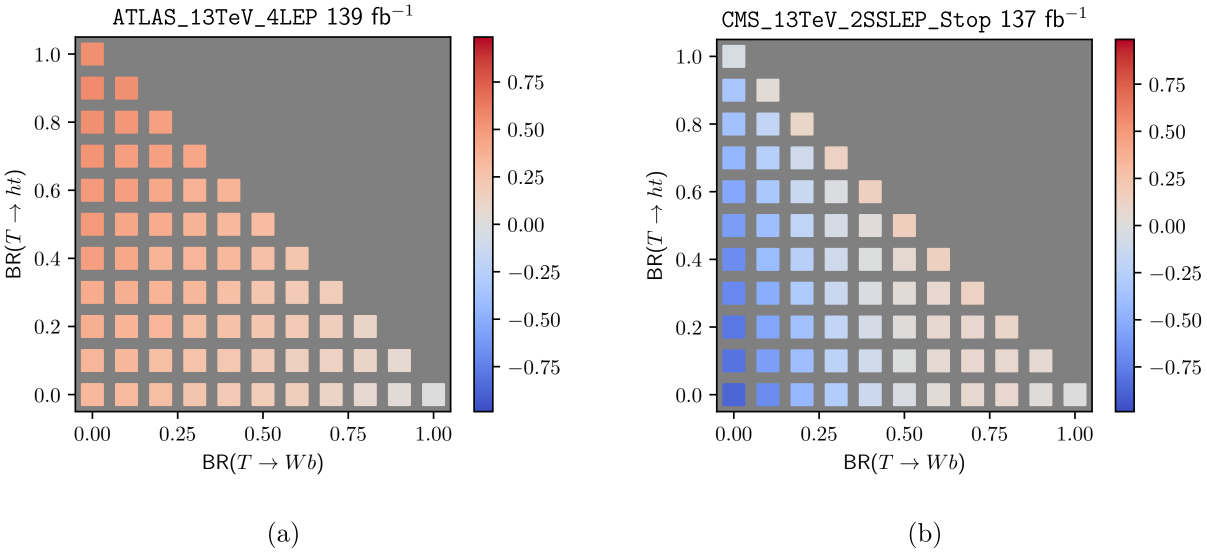


Figure 7.11: Two of the most significant likelihood contributions came from the ATLAS four-lepton SUSY search (a); and the CMS two same-sign or three lepton search (b). The ATLAS search provides the largest LLR in the Higgs-dominated top-left corner; whereas the CMS search is weakest in the W-boson-dominated bottom-right corner because this final state cannot produce the three charged leptons required for the excluding signal region.

Again, this included RPV-SUSY models where sparticles are still pair-produced, but can decay to SM particles and so do not contribute to missing transverse momentum. For reinterpretation purposes, the analysis provided a set of seventeen "simplified" inclusive signal regions (ISRs), which do not target specific models as tightly, but aim to provide *some* information about a more general set of models. For this study at least, this proved to be a very helpful decision.

The exclusion from the CMS search, shown in Figure 7.11b, comes from ISR17: a region that among other conditions, requires three leptons, at least two b-jets[8], and – presumably to include the pair-produced RPV models – no missing momentum requirement. The three-lepton requirement explains why the exclusion is weakest in the high BR($T \to Wb$) region in the bottom-right of the BR-plane, as if both VLQs follow this decay, the event cannot produce three leptons.

Smaller but nevertheless signficant contributions come from the CMS search for SUSY in two oppositely-charged lepton final states [277], and the ATLAS search for squarks and gluinos in final states with same-sign leptons [278]. This ATLAS search is particularly intriguing because it appears that several of its signal regions are sensitive to VLQs, and even though the analysis team have provided a `pyhf` full-likelihood for it, it has not yet been incorporated into ColliderBit. This suggests that adding the full-likelihood information could allow us to combine information from all the regions, improving the sensitivity.

7.4.4 Conclusions

The likelihoods obtained from ColliderBit SUSY searches were not large enough to exclude pair-produced VLTs at 1.3 TeV (which the ATLAS $T\bar{T}$ combination did): however, they do provide LLR information on a scale similar to that provided by Contur for the study in the previous section. Therefore, as MC events at 13 TeV will already be being produced for dedicated VLQ ColliderBit analyses, and it is unlikely that

[8] Recall that all pair production of VLQs that couple exclusively to the third-generation of SM quarks should produce at least two b-jets, as even when the VLQ decays to a top-quark, this in turn will decay to a W-boson and a b-quark.

ColliderBit will be able to include all the studies that went into the original combination without losses in information (due to unreproducible regions or region-combinations), it does make sense to include these analyses in future Gambit VLQ studies.

It is noteworthy that the signal regions that contributed most to the likelihood – unusually among the set of all SUSY signal regions – did not have any missing momentum requirements. In the context of the simple VLQ model we tested here, this makes sense: there are no sources of missing transverse momentum[9] other than SM processes such as $W \to \ell\nu_\ell$ or $Z \to \nu\nu$, which normally are much smaller in magnitude than BSM sources such as LSPs. However, it does highlight the importance of the inclusion of RPV-searches if we want to use a comprehensive SUSY-search program to look for more general "new physics".

7.5 The Rivet projection system: a thread-safe redesign

Section 7.3.2 showed that not being able to run with multiple Rivet `AnalysisHandler` objects in parallel has a significant negative impact on the usefulness of the Gambit-Contur interface. While post-processing jobs clearly have provided useful scientific information to the Gambit study, LHC measurement likelihoods could have even more impact if they could help steer the scan.

This would also likely provide a computational benefit as, because measurements have higher acceptances, we would need to generate fewer events to get a meaningful likelihood with Contur. Gambit already has machinery to terminate event generation if the existing data is sufficient to exclude the point, but this would be even more powerful if we could veto a parameter point after evaluating just 40 000 – 100 000 events with Contur, as opposed to 100 000 – 5 000 000 with ColliderBit searches.

7.5.1 Preamble: an even more technical look at the Rivet projection system

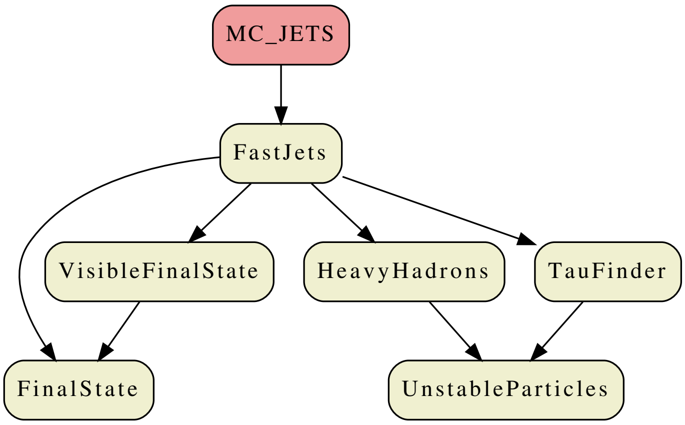


Figure 7.12: An example of a Rivet projection "tree", using the simple Rivet `MC_JETS` analysis. `MC_JETS` has only one child projection (`FastJets`), which in turn has four child projections. The `ProjectionHandler` has identified that the `FinalState` that is a child of `FastJets` is identical to the child of `VisibleFinalState`; likewise it has identified that both `UnstableParticles` children of `HeavyHadrons` and `TauFinder` are in fact the same projection.

As noted in Chapter 4, as Rivet developed from the HZTool software used at HERA [192], a key design consideration was to minimise the number of repeated identical calculations in computationally intensive

[9] If the VLQ model is expanded to include additional particles, then it is possible to add BSM sources of missing transverse momentum.

processes via the projection system.

In technical terms, a projection in Rivet is an object that inherits from the Rivet `Projection` class. Calling the projection's `project` method returns some physical observable(s) from the event being analysed. Examples include `FinalState`, which returns final-state particles satisfying a provided cut, and `Fastjets`, which returns clustered jets for a given jet algorithm. Effective caching of repeated projections for observables that are required in multiple places (either in the same analysis or across different analyses) is what allows Rivet to operate at a high level of efficiency even when conducting a large number of analyses. For example, many LHC analyses will carry out jet-clustering with the same algorithm and parameters: caching the `FastJets` projection ensures clustering is only carried out once per event. Projections will often declare and call other projections (rather than working on the HepMC events directly): this maximises the amount of common calculation that can be cached. However, this also means that care must be taken when building up the "tree" of required projections – the example in Figure 7.12 shows that even with just one simple analysis, projection caching can still avoid some duplicated effort.

Rivet, before version 4.0, solved this with a single global projection handler object, with which projections had to be registered as analyses were initialised. The handler was responsible for catching duplicate projections, and ensuring that the right projection was returned to parent projections and analyses.

We should also mention two other objects here. The first is an "analysis": in computational rather than physics terms, an analysis in Rivet inherits from the `Analysis` class, and provides `init`, `analyze` and `finalize` methods. `analyze` analyses HepMC events one by one, and `finalize` completes operations that need all the events before they can be executed. Of most interest here however is `init`, which constructs and registers the projections the analysis will require (as well as booking what output data the analysis will provide).

The second object to highlight is the analysis handler, which effectively controls a Rivet run: one adds the analyses one wants to use to it; it is the analysis handler that is responsible for taking each event and ensuring it is analysed by every analysis; the analysis handler that finalises each run; and the analysis handler that writes the output to Yoda files or streams.

Why Rivet wasn't thread-safe

Rivet was originally designed with a single, global projection handler. This design cannot be thread-safe – with multiple analysis handlers on multiple threads, each handler needs to be able to register its own projections, as each handler will need to project quantities from different sets of events.

One work-around attempted in recent versions of Rivet was to use `thread_local` – a `C++11` keyword denoting a static variable stored separately for each thread – projection handlers protected by a `std::mutex`[10]. But even though this gives us one analysis handler per thread, it still has significant drawbacks. Firstly, it is very sensitive to the exact order analysis handlers are initialised in – a basic test program running inside a simple OpenMP loop will still crash without additional preventative measures being put in place. Secondly, using `thread_local` variables means that one can still never have more than one analysis handler on each thread. This is particularly problematic for initialising and finalising runs – processes normally carried out on a single thread – although there may also be justifications for using two handlers in one thread during normal runs. Finally, although a `std::mutex` generates less overhead than, for example, the OpenMP `critical` blocks described in Section 7.2.3, any system of locks will always generate some overhead, which should be avoided where possible.

[10] A `mutex` is an object that can only be owned by one thread at once, allowing us to "lock" out other threads when a thread-unsafe step is being carried out.

7.5.2 Making Rivet thread-safe

The solution to the issues outlined above is to allow each Rivet analysis handler to have its own projection handler, as opposed to a single global projection handler. Unfortunately, if we constrain ourselves to having a minimal impact on the rest of Rivet, this is a deceptively complicated task.

Rivet's projection comparison functions very often rely at least in part on a recursive comparison – i.e. A and B are the same only if all their children are the same. This means that before a projection can be registered with the analysis handler, all its children (and their children, and their children's children, etc.) must be registered. This makes passing information about which projection handler "owns" the current projection recursively down to children a complex task, as the parent projection will not be registered, and hence will not know its projection handler.

One solution would be to add a pointer or reference to a projection handler as an argument to the constructor of every projection and to the `init` method of every analysis, so they could be manually passed from projection to projection. However, this would be a very bad idea: it would require changes to over a thousand Rivet analyses, as well as over a hundred projection constructors. It would also fundamentally change Rivet's principle that analyses – and to a lesser extent projections – should be simple to write, without requiring an understanding of advanced `C++` concepts. A lot of additional effort – and possibly yet more additional arguments – would be needed to ensure that children were always registered first, even in complex cases.

A better solution, which was implemented, is to give both the projection and analysis classes a new member called a `_declQueue`[11]. This is a double-ended queue – i.e. a `std::deque` – of pointers to constructed (but unregistered) child projections: the projections pointed to may in turn have child projections in their own queues. Then, after the `AnalysisHandler` has initialised the analysis, it calls `_syncDeclQueue`[12] on it, which recursively traverses the queues, registering projections from the bottom of the tree up. Letting the analysis handler call `_syncDeclQueue` after an analysis was initialised was found to be a much simpler solution than the initial plan for this section of code, which involved a complex system – that became ever-more complex as more special cases were found – of projections automatically detecting when they became owned, and then registering themselves with the projection handler. Particularly problematic for automatic detection were cases where projections took as arguments to their constructors references to projections that may or may not have already been registered.

Using this method means that all analyses and projections still work without modifications[13] – so the ordinary user should be totally unaware of these changes, unless they wish to utilise the thread-safety in their own Rivet runs.

The flowcharts in Figures 7.14 and 7.15 demonstrate how the new and old systems would work for a toy set-up (illustrated in 7.13): one analysis, in which one projection ("parent") is declared; the "parent" projection in turn declares one further projection, "child". The most obvious difference is that we now have a "two-pass" system rather than "one-pass" system – before, we could recurse through all an analyses' child projections just once, declaring on the way "back up". However, this is now no longer possible, as we need to be able to select the correct `ProjectionHandler` to declare with, so we need to make a second pass through the tree.

Far more complex cases than this two projection set-up are not only possible, but occur often across the

[11] In fact, the `_declQueue` is actually a member of the `ProjectionApplier` class, which is inherited from by both `Projection` and `Analysis` – for the full Rivet inheritance structure, see Reference [279].

[12] Also a method of `ProjectionApplier`.

[13] In the interests of full transparency, one projection did actually need to be changed: but this was because it incorrectly called `declareProjection` instead of `declare`, but was fortunate to "get away with it" under the old system.

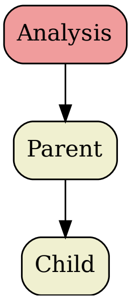


Figure 7.13: The projection tree for the very basic analysis used by Figures 7.14 and 7.15. By looking at Figure 7.15 and comparing the tree in this Figure to e.g. that in Figure 7.16, we can appreciate how complex the registration flowchart becomes for real use cases.

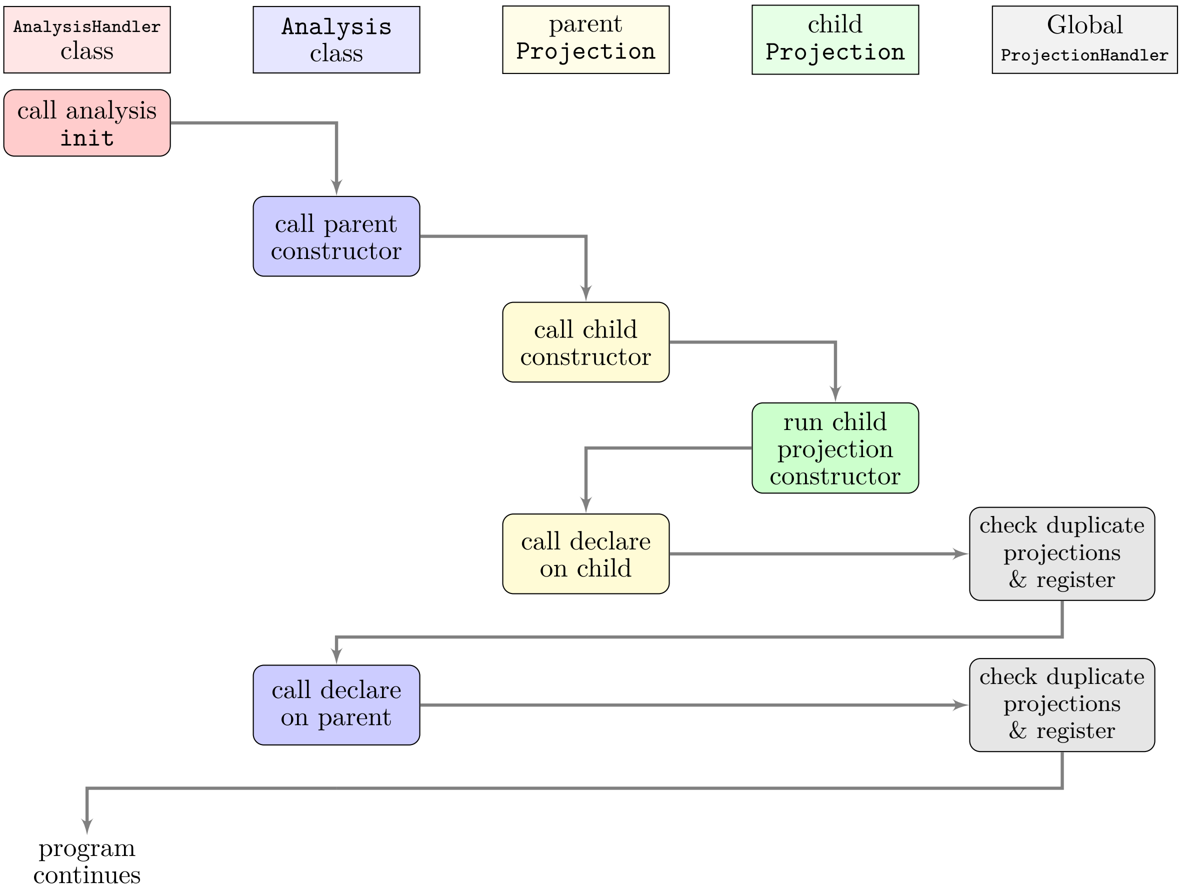


Figure 7.14: Flowchart illustrating how the old Rivet projection registration system used to work, for a simple case with one analysis and two projections, one of which is the child of the other. The columns illustrate which class the step is being carried out in.

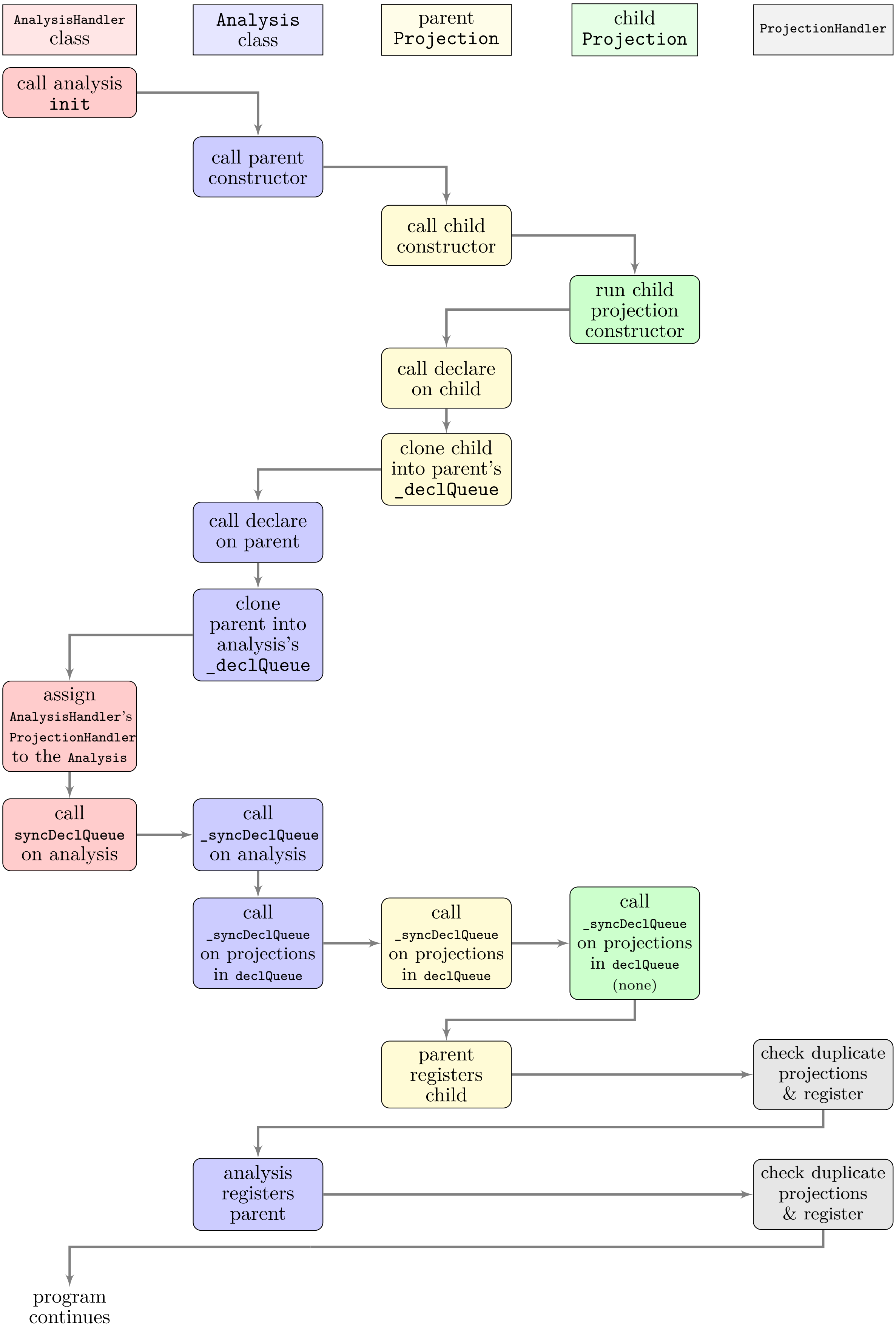

AnalysisHandler class
Analysis class
parent Projection
child Projection
ProjectionHandler
call analysis init
call parent constructor
call child constructor
run child projection constructor
call declare on child
clone child into parent's _declQueue
call declare on parent
clone parent into analysis's _declQueue
assign AnalysisHandler's ProjectionHandler to the Analysis
call syncDeclQueue on analysis
call _syncDeclQueue on analysis
call _syncDeclQueue on projections in declQueue
call _syncDeclQueue on projections in declQueue
call _syncDeclQueue on projections in declQueue (none)
parent registers child
check duplicate projections & register
analysis registers parent
check duplicate projections & register
program continues

Figure 7.15: Flowchart illustrating how the new Rivet projection registration works, for the same case as Figure 7.14. The columns illustrate which class the step is being carried out in, even though the methods themselves may belong to a base class. Note that calling the `_syncDeclQueue` method from a class passes on information about that class's `ProjectionHandler`.

set of all RIVET analyses.

The new version passed all of RIVET's internal continuous-integration (CI) checks, as well as the full RIVET regression suite designed to check consistency of results from version to version. Several large parallel runs were completed in order to ensure it was truly thread-safe. In this process several small bugs in the new system were uncovered and fixed, for example: a thread-safety issue with the RIVET logging system; and ensuring that the return value of the `declare` method was valid for cases where analysis authors had chained methods together.

This process also uncovered a thread-safety issue in the N-subjettiness `fastjet` plugin [280], used in several RIVET analyses. Two threads calculating the N-subjettiness almost simultaneously could affect each-others' axes definitions, because the arrays of jets and axes were declared as `static` in an effort to minimise re-initialisation time. This would corrupt the calculation leading to unphysical results that cause a crash. After correspondence with the package authors, I carried out several tests to verify that a fix would not come with a performance penalty. The fix has been included in `fj_contrib` since version 1.049.

7.5.3 An aside: a projection tree debugging tool for RIVET

In the process of developing and debugging thread-safe RIVET – where identifying discrepancies and debugging crashes in projection registration accounted for the overwhelming majority of the work – it quickly became apparent that trying to draw projection trees by hand based on chains of printed hexadecimal memory addresses was not a sustainable way of working. Therefore, I also developed a tool to help debug projection-trees. This tool is responsible for the plots in this section (Figures 7.12, 7.13, and 7.16). Two new classes are defined – a `ProjectionTreeNode` to track the individual projections, and the `ProjectionTreeGenerator` which, given an already initialised analysis handler, starts with the analyses and walks down the projection tree, calling `getChildren` until all projections are childless. Each time we reach a child projection, we add a node for it if it has not yet been encountered, and whether or not it is new, add an edge from its parent to

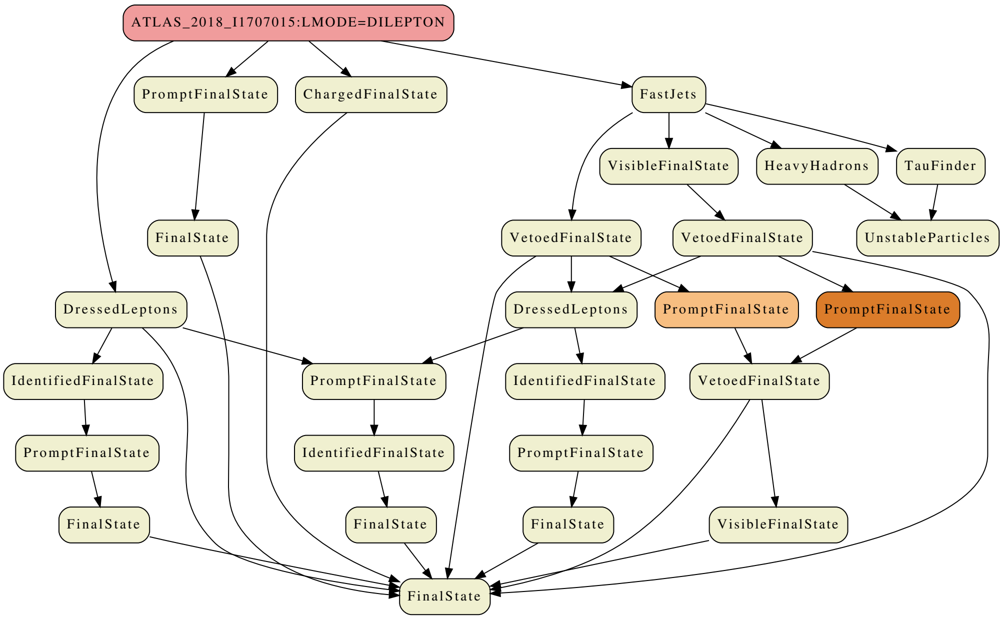


Figure 7.16: Debugging the `ATLAS_2018_I1707015` projection tree. This demonstrates a real use-case of the debugger. In moving from thread-unsafe to thread-safe RIVET, we found a discrepancy in this analysis – the extra `PromptFinalState` projection highlighted in dark-orange, which should be equivalent to the one in light-orange. This led to an improvement in the `compare` method of the `VetoedFinalState` projection.

itself.

A good example of where the projection tree generator proved its worth was whilst debugging differences in projection registration between the new and old systems for the `ATLAS_2018_I1707015` analysis (based on Reference [281], though the physics details are irrelevant here). Because the projection `compare` method guarantees that the ordering of projections within a ProjectionHandler will always be the same, for two identical projection trees, the graphviz `.gv` files outputted by the ProjectionTreeGenerator should be ASCII character-for-character the same, allowing easy comparison with the `diff` utility. Therefore, to ensure that the old and new methods produced the same results, it was straightforward to generate trees for all analyses and compare them. This procedure identified a discrepancy in the aforementioned analysis, which was traced to a flaw in the `compare` method of the `PromptFinalState` projection, as illustrated in Figure 7.16. Fixing the flaw not only brought the new and old projection registration systems into agreement, but also led to performance benefits across Rivet as other unnecessarily duplicated `PromptFinalState` projections – and possibly their parents, differentiated only by recursive comparison – were removed. The total speed-up when running across the full set of LHC measurements – the common use case for Contur and Gambit – was about 5%.

The projection tree generator has been included in Rivet from version `4.0.0`, and can be run from the command line with Rivet by passing the `--print-projection-tree` flag when running. In order to avoid adding a dependency on the graphviz package [282] to Rivet, the tool outputs a plain-text `.gv` file, and prints instructions on how to convert this to a graphics file if the user has graphviz installed.

The debugging tool can also be accessed via the Rivet Python API. This means that projection trees can still be debugged when Rivet isn't being run directly from the command line, but rather steered by Python scripts. It also allows users to create even more advanced debugging metrics by using the raw tree-network data in Python – for example, by interfacing with a graph-theory focussed package like `networkx` [283].

7.5.4 Impact on Gambit

The comparison of Figure 7.17 with Figure 7.2, shows how thread-safe Rivet will simplify the ColliderBit workflow, and suggests that we might expect significant performance improvements. While the full suite of Yoda `2.0`, Rivet `4.0`, and Contur `3.0` was released too late for a full and official integration into Gambit, preliminary results testing on a single machine were obtained. A single desktop – typically with four to sixteen cores – will not see the same increase in performance as an HPC machine that can run hundreds or thousands of threads simultaneously, but nevertheless there should be a clear indication of the improvement.

Test results are shown in Figure 7.18, comparing the "old", thread-unsafe setup described in Section 7.2.3 and illustrated in Figure 7.2 with the new setup illustrated in Figure 7.17. 10 000 events were generated by ColliderBit, analysed by Rivet only (no ColliderBit analyses), and then statistics were calculated using Contur. This test was repeated five times for each number-of-threads to obtain a meaningful error estimate.

As expected, performance gains from multithreading for the "old" setup are minimal. Even though some time will be saved in parallel event generation, the additional overhead of locks to ensure the analysis handler is only being accessed by one thread at a time leads to a slow-down for more than four threads. With thread-safe Rivet, and one analysis handler per `OpenMP` thread, there was a significant improvement in performance, with results a lot closer to the naïve multi-threading model (where the total run time is simply divided by the number of threads).

There is still some saturation after four threads, but this is not unexpected in this context, and is not a red-flag for performing Gambit scans with $\mathcal{O}(100)$ `OpenMP` threads. Real scans using HPC resources would

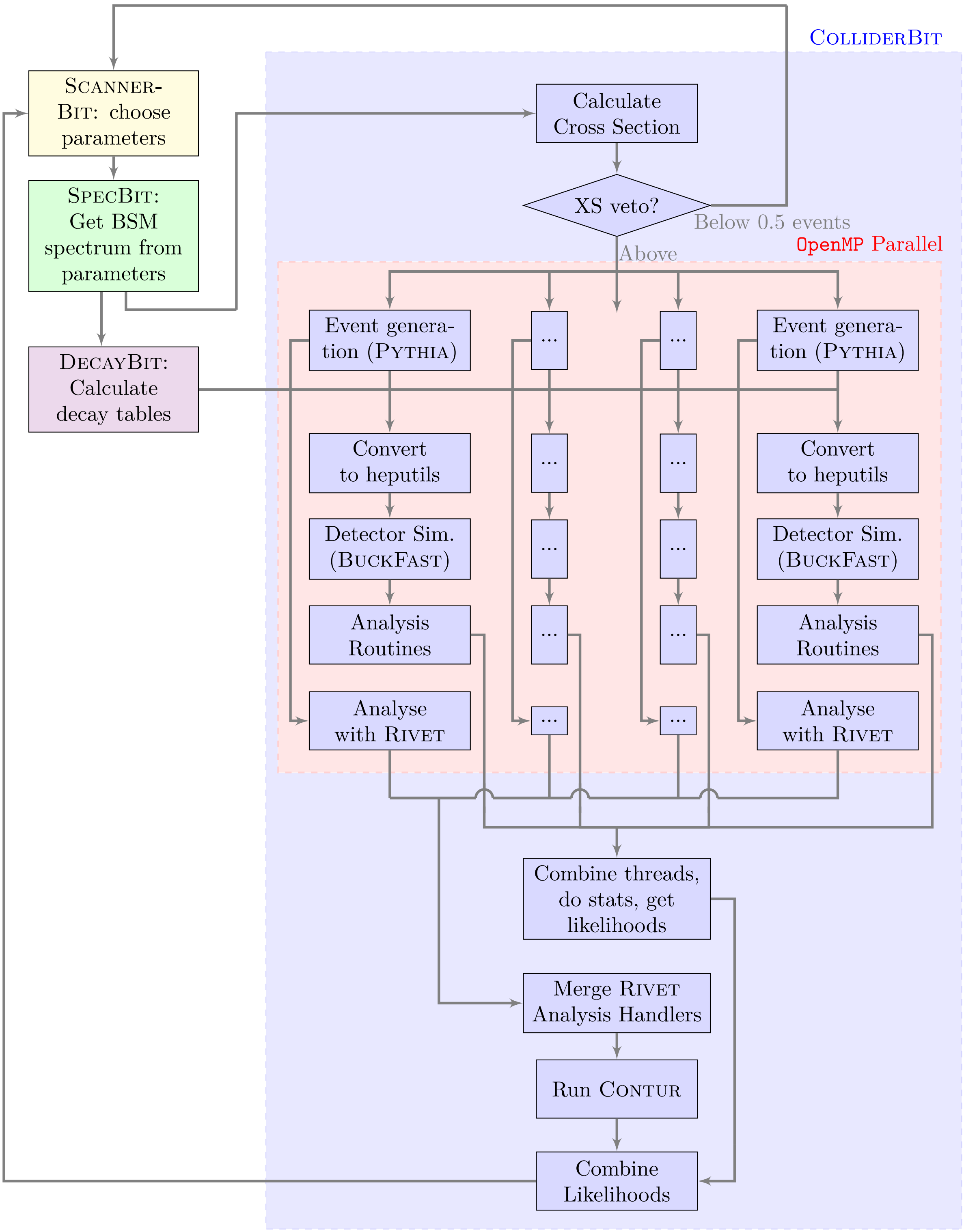

ColliderBit
Scanner-Bit: choose parameters
SpecBit: Get BSM spectrum from parameters
DecayBit: Calculate decay tables
Calculate Cross Section
XS veto?
Below 0.5 events
Above
OpenMP Parallel
Event generation (Pythia)
...
Convert to heputils
Detector Sim. (BuckFast)
Analysis Routines
Analyse with Rivet
Combine threads, do stats, get likelihoods
Merge Rivet Analysis Handlers
Run Contur
Combine Likelihoods

Figure 7.17: Extending Figure 4.5 to show how ColliderBit can interact with thread-safe Rivet. In contrast to the thread-unsafe version in Figure 7.2, the diagram is simplified by allowing multiple analysis handlers to run at once. As before, the flowchart is illustrative and not comprehensive, and the vertical axis implies concurrency within the OpenMP block exclusively.

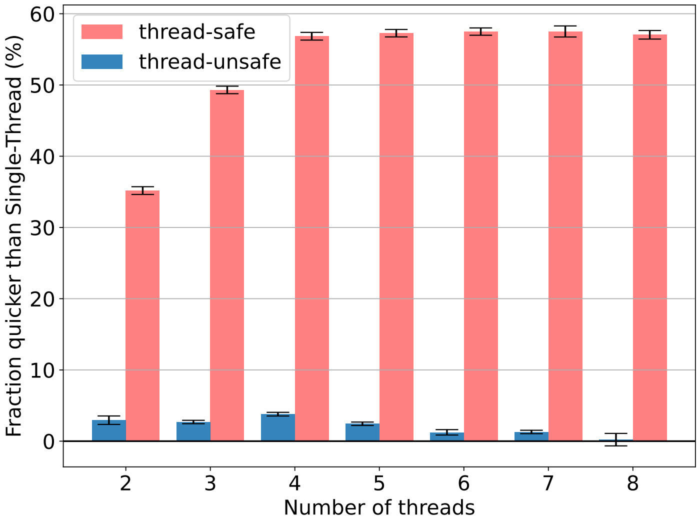


Figure 7.18: Gambit+Rivet performance as a function of number of OpenMP threads. Performance actually does increase with the number of OpenMP threads when using thread-safe Rivet inside Gambit. All tests were carried out on an Intel i7 processor with four cores and eight logical processors.

simulate between 100 000 and 16 million Monte-Carlo events at each parameter point, whereas in order to complete this test locally with reliable statistics, only 10 000 were used. For this number of events, when using five or more threads, close to fifty percent of the total program execution was in single threaded initialisation or finalisation steps, such as running Contur or the Gambit dependency resolution system. Therefore, the possible performance improvements at this point quickly become negligible. It is also worth noting that the test machine only had eight logical processors, so background and operating system tasks likely also had an impact.

Overall, the results show that adding thread-safety to Rivet does add a significant performance boost when running Rivet inside Gambit; and that therefore it will soon be practical to include likelihood contributions not just as post-processing steps to global fits, but in driving the scanning steps too.

Chapter 8

ATLAS search for single-produced vector-like *T*- or *Y*-quarks decaying to a *W*-boson and a *b*-quark in the one-lepton channel

8.1 Analysis philosophy

The analysis in this chapter aims to discover – or, in the case of a null result, place limits on – the single production of vector-like T- or Y-quarks decaying to a W-boson and a b-quark, using 139fb^{-1} of LHC Run 2 data recorded by the ATLAS experiment. As laid out in Section 1.3, both the vector-like T and Y quarks can decay to a Wb final state (as illustrated in Figure 1.3), so both particles can be targetted by a single search.

For the case of VLQ single-production, one of the Feynman diagrams for the process is reproduced in Figure 8.1. A key feature to note is the spectator quark q', which will typically produce a "forward" (i.e. $|\eta| > 2.5$) jet, a distinctive signature of VLQ single production [43]. The W-boson can further decay either hadronically or leptonically and, in the latter case, the emission of a neutrino will also produce missing momentum. Although the branching fraction is larger for hadronic decay, previous studies have shown that the leptonic channel is more sensitive, and it has been studied by ATLAS on multiple previous occasions, with less data and at different energies [284–286].

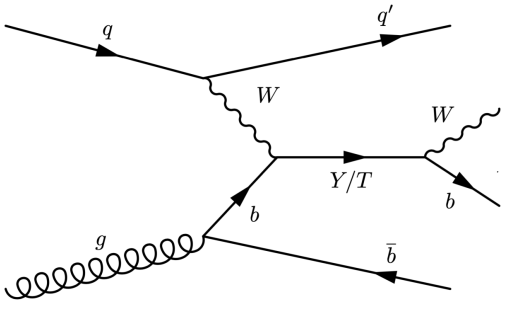


Figure 8.1: One diagram for the single-production of vector-like T or Y quarks, decaying to the $W + b$ final state.

Based on the Feynman diagram, for the leptonic decay channel (henceforth, “1-lepton”), we note some key signatures we wish to target: a forward jet from the spectator quark, a lepton and E_T^{miss} from a leptonically decaying W-boson, multiple b-quarks[1], and no other light central jets. These observations will inspire the definition of our signal region (and indirectly, the control regions), introduced in Section 8.4.

This signature has already been studied once during Run 2 by ATLAS, using 36 fb^{-1} of data [286]. No significant deviation from the SM was observed, and limits were placed on the coupling strengths at a selection of given mass points.

8.2 Object definitions and selections

In Section 8.1 we laid out the rationale for the final state we hope to observe: a leptonically decaying W-boson, as well as light and heavy-flavour (HF) jets. Hence the physics objects that we need to extract from LHC data are electrons, muons, jets (including b-tagging information) and E_T^{miss}. Photons and tau-leptons are not required or reconstructed.

8.2.1 Leptons

Electrons are constructed from the EM topo-clusters in the calorimeter matched to ID tracks, as described in Section 2.7.4, and muons using the muon systems as laid out in Sections 2.5 and 2.7.5. Both electrons and muons use the `tight` working points. Electrons are required to have transverse energy $E_T > 27$ GeV; and muons are required to have $p_T > 27$ GeV. As described in Chapter 2, the layout of the detector systems constrains the range in which we observe leptons: for electrons, this means $|\eta_e| < 2.47$ and $|\eta_e| \not\in [1.37, 1.52]$; and for muons $|\eta_\mu| < 2.5$.

8.2.2 Jets

This analysis uses only “small” PFlow jets, constructed using an $R = 0.4$ anti-k_T algorithm from calorimeter topo-clusters and tracks, as described in Section 2.7.2. The JVT (with a cut of 0.59, a medium WP) is applied to filter out jets with $p_T < 60$ GeV and $|\eta_j| < 2.4$ which are likely to have originated from pile-up. Jets selected for the analysis required transverse momentum $p_T > 25$ GeV, and were split into central jets, $|\eta_j| < 2.5$, and forward jets $2.5 < |\eta_j| < 4.5|$. Jets with $|\eta_j| < 2.5$ were tagged using the DL1r algorithm, discussed in Section 2.7.3, which aims to tag jets containing b-hadrons. The loose 85% efficiency working point was used, and the subset of jets passing this tagging requirement will be labelled “b-jets”.

8.2.3 Missing transverse energy

The E_T^{miss} is defined as the negative vectorial sum of all the physics objects in the event (i.e. the jets and lepton in the preceding section), combined with a soft-term, as described in Section 2.7.6. The `tight` working point is used in the definition of the soft term: the E_T^{miss} contribution should be dominated by the neutrino from leptonic W-boson decay, and it is important that pile-up jets do not interfere with our ability to reconstruct the W-boson.

[1] It should be noted that the b-quark not originating from the VLQ decay may be too soft to observe, primarily due to PDF bias.

8.2.4 Overlap removal

Following a logic very similar to the detailed explanation in Section 2.7.7, the overlap removal procedure for the analysis is as follows:

1. Remove any muons tagged in the calorimeter that share an ID track with an electron.
2. Remove any electrons that share an ID track with a muon.
3. Remove any jets within a cone of $\Delta R < 0.2$ of any electron candidates.
4. Remove any electrons within a cone of $\Delta R < 0.4$ of any remaining jets.
5. Remove any muon within a cone of $\Delta R < \min\{0.4, 0.04 + 10\text{ GeV}/p_T^\mu\}$ of any jets.
6. Remove any remaining muons within a cone of $\Delta R < 0.2$ of jets ghost associated to three or more ID tracks.
7. Remove any remaining jets, ghost associated to fewer than three ID tracks, that are within a cone of $\Delta R < 0.2$ of any surviving muons.

8.2.5 Reconstruction of higher-level variables

This analysis targets events that contain a leptonically decaying W-boson. We can "reconstruct" the four-momentum of the W-boson via a procedure similar to that outlined in Section 2.7.6: we obtain the p_x and p_y components by summing the momenta of the lepton and the $\vec{E}_T^{\text{miss}}$; and then, by insisting that the neutrino was massless and that the reconstructed W is on-shell, we are left with a quadratic equation that provides the remaining components[2]. Both of these assumptions – that all the E_T^{miss} comes from the neutrino and that the W is perfectly on shell – are approximations, but they do normally produce consistent results. If this equation provided multiple real solutions, the smaller value of p_z^ν was chosen; if instead there were no real solutions, the magnitude of the missing transverse energy was varied minimally until there was exactly one solution. The reconstructed transverse momentum of the W-boson, p_T^W, will be an inportant variable in this analysis.

Assuming we are looking at a signal event, we can go one step further and reconstruct the VLQ mass (m_{VLQ}) by summing the four-momenta of the W-boson and leading b-tagged jet, and extracting the mass component of the result. If we are looking at the decay of a "real" VLQ, the distribution of the reconstructed four-vector mass should be centred around the "true" mass of the VLQ (as will be shown in Figure 8.4). However, SM sources of W-bosons and b-jets will not produce such a distribution – which gives us a variable that provides excellent discrimination between VLQ signal processes and the SM background.

8.3 Monte-Carlo simulation

8.3.1 SM backgrounds and generator systematic uncertainties

The most significant SM backgrounds for this analysis are $t\bar{t}$, W+jets, and single-top production; though diboson (WW, WZ, ZZ), Z+jets, $t\bar{t}V$, $t\bar{t}H$, and QCD multi-jet production also need to be considered. Table 8.1 shows the exact generator versions used for all the nominal background samples used in the

[2] The equation can be re-arranged in terms of any of the remaining unknown variables, but for the Rivet implementation (see later sections) we used p_z^ν.

statistical fit. Where Sherpa was used to conduct the parton shower, it used the default Sherpa tune; where Pythia was used, it used Atlas's A14 tune [287].

For $t\bar{t}$ producton, the h_{damp} parameter[3] was set to $1.5m_t$. Varying this to $3m_t$, combined with the variation of the renormalisation and factorisation scales μ_R, μ_F by factors of two, constituted an initial-state radiation (ISR) uncertainty. A final-state radiation uncertainty (FSR) was provided by variations on the α_S parameter in Pythia. Generator and parton shower uncertainties were estimated as two-point systematics using an additional two samples: a MadGraph5_MC@NLO+Herwig sample and a Powheg Box+Herwig sample. Given the scale of these uncertainties, AtlFast II detector-simulation was sufficient, though notably this meant we also required an AtlFast version of our nominal $t\bar{t}$ sample to give a true "apples-to-apples" comparison. These generator uncertainties were also decorrelated (i.e. allowed to vary independently) between low-, mid-, and high-p_T^W regions (which will be discussed further in Section 8.4.1). To further account for mismodelling in the phase-space of our signal region, the overall normalisation of $t\bar{t}$ was allowed to float freely (i.e. a flat distribution with no penalty term), which will be discussed further in Section 8.6.2.

The same approach to ISR and FSR uncertainties was used for the single-top sample. The approach for generator and parton-shower uncertainties was also similar to that used for $t\bar{t}$: an MadGraph5_MC@NLO + Pythia sample was used to estimate a two-point generator uncertainty; and a Powheg Box + Sherpa sample was used to estimate a 2-point shower uncertainty. As for $t\bar{t}$, these uncertainties all made use of AtlFast II, and were decorrelated based on p_T^W.

There is overlap between final states in $t\bar{t}$ and Wt processes. To avoid double-counting such events, the "diagram removal" (DR) scheme [288, 289] was used. The alternate "diagram subtraction" scheme [289, 290] is used to provide a two-point uncertainty on the choice of scheme.

W+jets and Z+jets samples were produced with Sherpa 2.2.11. Sherpa 2.2.11 showed an improvement in performance for our analysis over older samples produced using Sherpa 2.2.1, possibly because the newer version has improvements which allow better population of low cross-section areas of the phase space [291] (e.g. BSM signal regions). This also allowed us to incorporate into our samples NLO EW corrections, which are available as "on-the-fly" weights. The "exponentiated" scheme [292] was used as the nominal, and the difference between the exponentiated and multiplicative schemes was used to estimate an (albeit negligible) uncertainty on this correction. The W+jets events were further split into two samples: "W+HF", for events containing at least one jet ghost-associated to a b- or c-hadron; and "W+light" for the remaining events. Almost all generator systematics were considered independently for the two samples. Both samples were given an independent free-floating normalisations in the final fit.

Errors on the PDFs used to generate the $t\bar{t}$, single-top and W+jets samples were included using the 30-point ($t\bar{t}$ and single-top) and 100-point (W+jets) variations for the given PDF set. For all samples, the PDF errors were negligible – less than 0.5% in every fitted bin.

Unless otherwise specified, full detector simulation as described out in Chapter 3 was applied to all samples; including all the nominal SM events actually included in the fit.

8.3.2 VLQ production

Signal samples were generated at LO using MadGraph5_MC@NLO 2.6.5, using Pythia 8 for the parton shower and hadronisation. The VLQ model used was that described in equation 1.9, provided by Reference [44]. A 4FNS was used, and the PDFs came from the LO NNPDF2.3 set. For the VLT, the samples used only included the $T \rightarrow Wb$ decay, as earlier internal studies showed this analysis was only negligibly

[3] In top-quark MC production, h_{damp} is a matching/merging related parameter that controls the scale of the first high-p_{T} emission.

Process	Generator + parton showering/hadronisation	PDF set: Gen. + show./had.	Inclusive cross-section order in pQCD
$\boldsymbol{t\bar{t}}$	Powheg-Boxv2 + Pythia 8.230	NNPDF3.0 + NNPDF2.3	NNLO
Single top	Powheg-Boxv2 + Pythia 8.230	NNPDF3.0 + NNPDF2.3	NNLO
Dibosons WW, WZ, ZZ	Sherpa 2.2.2	NNPDF3.0	NNLO
$\boldsymbol{Z}$**+jets**	Sherpa 2.2.1	NNPDF3.0	NNLO
$\boldsymbol{W}$**+jets**	Sherpa 2.2.11	NNPDF3.0	NNLO
$\boldsymbol{t\bar{t}V}$	Madgraph5_aMC@NLO 2.2.3 + Pythia 8.210	NNPDF2.3 + NNPDF2.3	NLO
$\boldsymbol{t\bar{t}H}$	Powheg-Box 2.0 + Pythia 8.230	NNPDF3.0 + NNPDF2.3	NLO
QCD multijet	Pythia 8.186	NNPDF2.3	LO

Table 8.1: Generator information for all the nominal background samples used in the analysis.

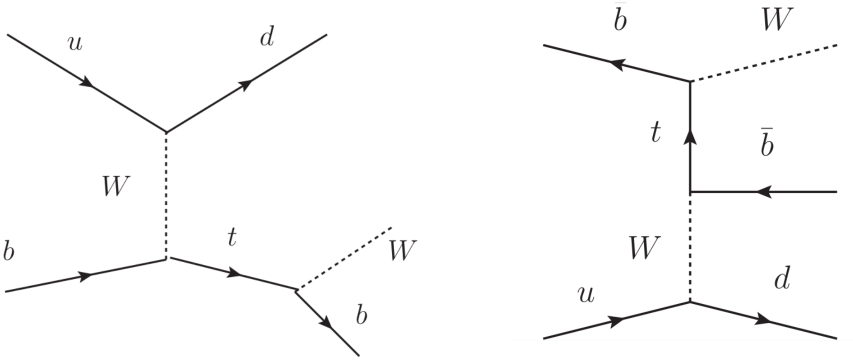
sensitive to the other decay modes of the T. The LO cross-sections from MadGraph5_MC@NLO were corrected to NLO using the higher-order benchmark results in Reference [293]. Notably, these NLO results use the narrow-width approximation[4]; therefore, to ensure all results displayed are valid, we will not display any mass points where the ratio of the VLQ width to its mass (Γ/M) is greater than 50%.

On multiple occasions throughout this thesis we have discussed problems related to the computational cost of carrying out full MC simulation across multi-dimensional BSM signal grids; and this chapter is no exception. Ideally, for this study, we would like dedicated samples of singlet-T[5], doublet-Y (right-handed) and triplet-Y (left-handed) across the mass-κ plane. However to save computational resources, the 2D grid was collapsed into a 1D grid, with samples being generated every 200 GeV at $\kappa = 1$ in the mass range [1100 GeV, 2700 GeV]. These events were then reweighted to other values of mass and κ using a matrix-element event-by-event reweighting, based on the difference in kinematic distributions at primary-parton level. A 2.5% uncertainty is included to account for possible discrepancies between the reweighted samples and an equivalent sample that went through full processing.

Furthermore, because we do not consider other T decay modes and because EM effects are negligible, the T and Y quarks are effectively indistingushable – other than interference effects (discussed below) and constant factors of $1/2$ from production and decay which are easily added *post-hoc*. Other internal studies have also shown (again excluding intereference effects) that there is only a negligible difference between the two chiralities of the Y. Therefore, we only need to fully generate one set of VLT events at $\kappa = 1$ in order to survey all three signal models across the grid.

[4] I.e. the Breit-Wigner function in the VLQ propagator is collapsed to a dirac-δ function in the cross-section calculation; this topic is explored at length in Reference [294].

[5] For the singlet-T scenario, the T is left-handed, and BR($T \rightarrow Wb$) = 0.5.



(a)

(b)

Figure 8.2: Processes that can interfere with single VLQ production: (a) off-shell single t production interferes with T production; (b) electroweak Wbq production interferes with (left-handed) single-Y production.

8.3.3 Signal production: interference effects

There can be non-negligible interference effects between VLQ production and the SM, which a search for VLQs cannot completely ignore. Notably, these will be different for the different signal scenarios. For the singlet-T (where the T only has left-handed couplings), the interference comes from single-top production with a very off-shell top-quark, as illustrated in Figure 8.2a.

If the Y is predominantly left-handed, then interference is possible with Electroweak Wbq production, as illustrated in Figure 8.2b. This is the case for the $\begin{pmatrix} T & B & Y \end{pmatrix}$ triplet model[6] in which the right-handed component of the Y is heavily suppressed [40]. However, in the case of a Y in a $\begin{pmatrix} B & Y \end{pmatrix}$ doublet, the LH coupling of the Y is suppressed; and due to the parity-violating nature of the weak force, interference effects with the RH Y components are minimal.

A completely rigorous MC treatment of interference would require the production of a dedicated sample of VLQ + SM processes for the matrix element

$$|\mathcal{M}|^2 = |\mathcal{M}_{\mathrm{SM}}|^2 + |\mathcal{M}_{\mathrm{VLQ}}|^2 + 2\,\mathrm{Re}\left(\mathcal{M}_{\mathrm{SM}}{}^* \cdot \mathcal{M}_{\mathrm{VLQ}}\right) \ , \tag{8.1}$$

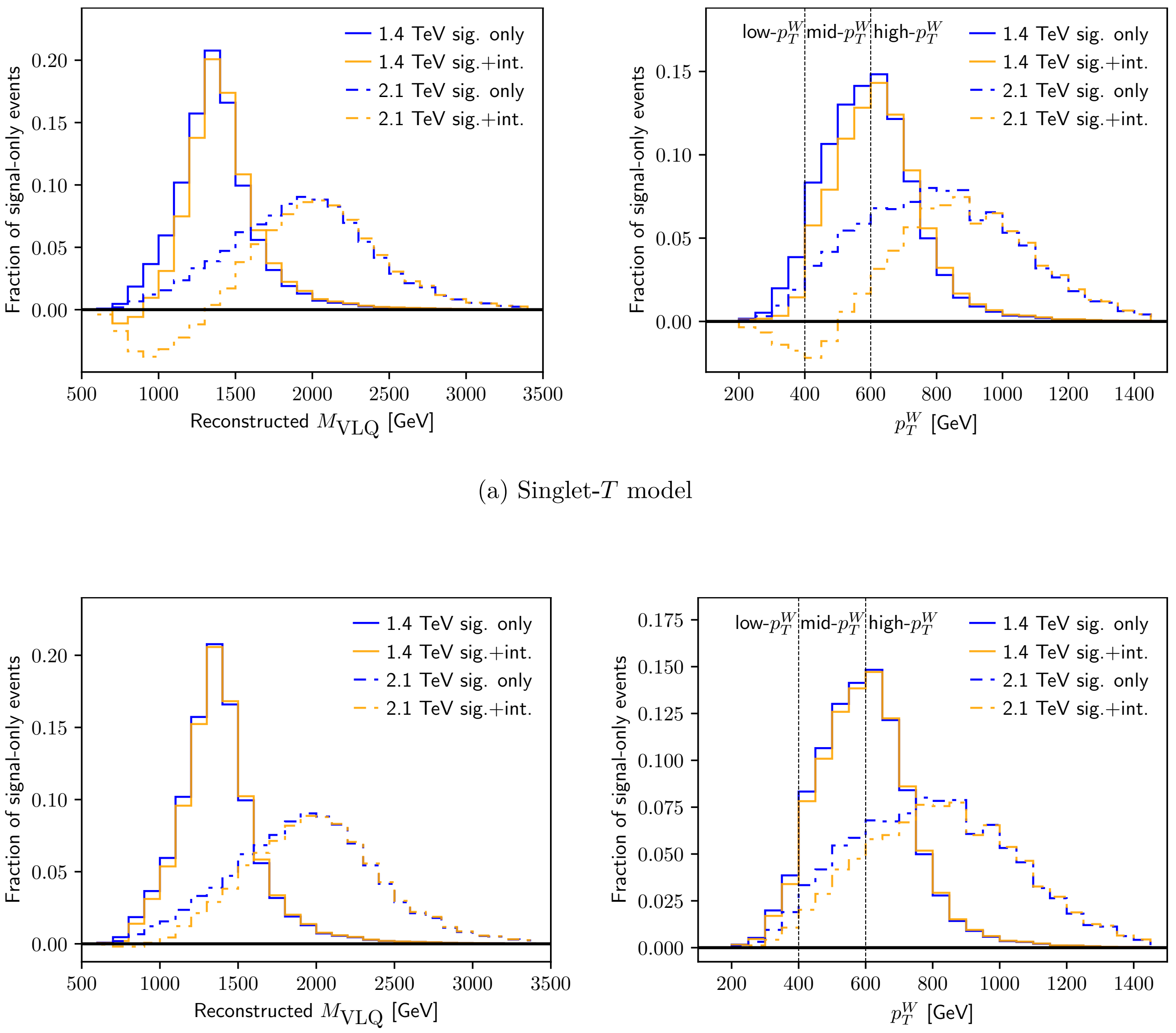
which would be computationally impractical. Instead, these interference effects are included by integrating them into the matrix-element reweighting scheme outlined above for reweighting across κ. In the full combined treatment, first introduced in the 36.1 fb^{-1} version of this analysis [286], the particle-level distributions of Wb invariant mass $f(m_{Wb})$ are used to obtain interference inclusive reweighting factors r from κ_0 to κ as

$$r(m_{Wb}\,;\kappa,\kappa_0) = \frac{K_{\mathrm{VLQ}} f_{\mathrm{VLQ}}(m_{Wb}\,;\kappa) + \sqrt{K_{\mathrm{SM}} \cdot K_{\mathrm{VLQ}}}\, f_{\mathrm{I}}(m_{Wb}\,;\kappa)}{f_{\mathrm{VLQ}}(m_{Wb}\,;\kappa_0)} \ , \tag{8.2}$$

where the K-factors K_{SM}, K_{VLQ} are the constant terms that normalise the LO cross-section for a given process to the NLO cross-section. This can be split into two terms to allow the signal-only and interference components to be treated separately.

Because they interfere with different SM processes, the effect of the interference on our final observables is different for the singlet-T and triplet-Y scenarios. This is illustrated in Figure 8.3, which shows the impact of interference at two different mass points for the m_{VLQ} and p_T^W distributions. Overall, the interference has more impact on the singlet-T than the triplet-Y – there is almost no negative “dip” in the distributions of the Y even for heavy VLQs; and also becomes more significant the higher the mass of the VLQ.

[6] Note we do not consider the T in the triplet scenario for this analysis, because BR($T_{\mathrm{triplet}} \rightarrow Wb$)= 0.



(a) Singlet-T model

(b) Triplet-Y model

Figure 8.3: The effect of interference on two important observable distributions in our (inclusive) signal region – defined formally in Section 8.4 – for the singlet-T (a) and triplet-Y (b) models, at a low and a high signal mass. Interference has a greater effect for high-mass VLQs – though for a given signal model is more impactful at lower values of the reconstructed VLQ mass – and is more significant for the singlet-T than for the triplet-Y.

8.4 Analysis strategy

After applying the preselection set out in Table 8.2, a signal region (SR) is defined as shown in Table 8.3, based on the signatures described at the start of this chapter. As discussed in Section 8.2.5, the reconstructed VLQ mass has a very different shape for signal and background samples and so, as illustrated in Figure 8.4, the fitted variable within the signal region (and indeed all regions) is the reconstructed VLQ mass, m_{VLQ}.

Also visible in Figure 8.4 are the Monte-Carlo simulations of the background. The largest backgrounds in the SR are $t\bar{t}$ and W+jets, which unfortunately are sensitive to mismodelling in extreme, BSM-sensitive phase-spaces. Therefore, in addition to control-regions – defined in Table 8.3 and chosen for having a similar phase-space to the SR but being signal-depleted and dominated by their respective background samples –

Requirement	Preselection	W+jets reweighting
Leptons	1	1
E_T^{miss}	> 120 GeV	> 120 GeV
Central jets ($p_{\text{T}} > 25$ GeV)	≥ 1	≥ 1
b-tagged jets	≥ 1	0
Leading-jet is b-tagged	Yes	-
Leading-jet p_{T}	> 200 GeV	> 200 GeV
$\left\lvert \Delta\phi \left(\text{leading jet}, \vec{E}_T^{\text{miss}} \right) \right\rvert$	≥ 2	≥ 2

Table 8.2: Preselection cuts for the analysis regions and the W+jets reweighting region (which requires no further cuts).

Region Requirement	SR	$t\bar{t}$ VR	$t\bar{t}$ CR	W+jets CR	W+jets VR 1	W+jets VR 2
b-tagged jets	≥ 1	≥ 1	≥ 2	1	1	1
Leading-jet p_{T}	> 350 GeV	> 200 GeV	> 200 GeV	> 250 GeV	> 250 GeV	> 250 GeV
$\lvert\Delta\phi(\text{lepton, leading jet})\rvert$	> 2.5	> 2.5	> 2.5	> 2.5	[1.5, 2.5]	[1.5, 2.5]
Additional hard central jets	0	≥ 1	0	–	–	–
Forward jets ($p_{\text{T}} > 40$ GeV)	≥ 1	≥ 1	0	0	≥ 1	0

Table 8.3: Definition of the signal, control and validation regions. The signal and control regions are used in the fit, whereas the validation regions are only used as a check on the modelling. The preselection cuts in Table 8.2 must also be satisfied, and the regions displayed here will further be split by p_T^W into low, mid- and high based on cuts at 400 GeV and 600 GeV.

we also define background-reweighting regions for these two processes, to allow us to perform a reweighting to correct them. Finally, we also define two W+jets and one $t\bar{t}$ validation region, in order to ensure that the adjustments made by the fit are not overly-tuned to the control regions.

These region definitions and broader strategy were already in place from earlier ATLAS efforts in this channel, and the only adjustments made to it were a swapping of the $t\bar{t}$ control- and validation-regions[7], as well as the p_T^W-based splitting set out below.

8.4.1 p_T^W splitting

The signal, control and validation regions defined in the section above were further split into three based on the value of the reconstructed p_T^W, with "low", "medium" and "high" regions based on p_T^W cuts at 400 GeV and 600 GeV. How this splitting divides the signal distribution is illustrated in Figure 8.3.

This was originally carried out only in the W+jets control region and the signal regions, motivated by a desire to improve the post-fit modelling of the p_T^W distribution in the W+jets control region by moving to a coarsely binned two-dimensional fit; but, even though this was significantly ameliorated instead by improvements to the reweighting (described in Section 8.5), this splitting was kept – and indeed propagated to the other regions – because it was found to provide other improvements to the analysis.

[7] The "old" control region contained too much single-top production to effectively control the $t\bar{t}$ population of the SR.

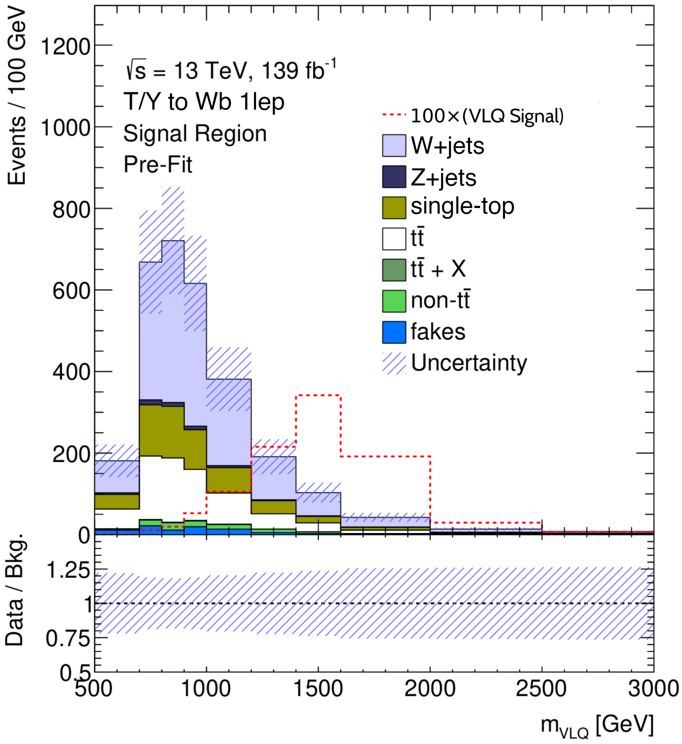


Figure 8.4: The Monte-Carlo predictions for the population of the signal region defined in Table 8.3. The VLQ signal model – in this case a 1600 GeV singlet-T is multiplied by a factor of 100 to make it easier to see. It is nevertheless clear that the shape of the m_{VLQ} distribution is very different for background and BSM samples.

Many of the modelling systematics – which were discussed in Section 8.3 – were decorrelated according to the p_T^W cuts. Several systematics were observed to behave differently between the slices, and a better description of the data was obtained when decorrelating them. For example, in background-only fits which used Asimov data in the blinded regions[8], the parton-shower uncertainty on the $t\bar{t}$ MC sample was pulled strongly, but only for the low-p_T^W regions.

This change also improved the search's sensitivity by replacing one inclusive signal region with two regions with a comparable or even higher signal-to-background (S/B) ratio (the "mid" and "high" p_T^W slice), and just one with a lower S/B (the "low" region). This is illustrated in Figure 8.5 for a variety of parameter points. In practical terms, the relatively signal-depleted low-p_T^W region also gave a good intermediate region for partial unblinding, to ensure that the post-fit SM modelling was reasonable beyond the control and validation regions once a final fit-configuration had been agreed. For consistency, we will continue to call this region the low-p_T^W signal region, despite its very low S/B ratio.

After the split, for some lower-mass and higher-κ signal scenarios the high-p_T^W $t\bar{t}$ control region also had a high signal-to-background ratio (over 10%). As a precaution, this region – and likewise the high-p_T^W $t\bar{t}$ validation region – were initially blinded.

[8] To avoiding biasing, the measured data in regions with $S/B > 10\%$ (for a 1600 GeV T-singlet) was initially blinded while the analysis design was being finalised. The blinded regions were the mid- and high-p_T^W SR, the high-p_T^W $t\bar{t}$ CR, and the mid- and high-p_T^W $t\bar{t}$ VR; for historical reason the low-p_T^W SR was also blinded. While the data was blinded, to test the full fit setup, the Asimov prediction was used in these regions.

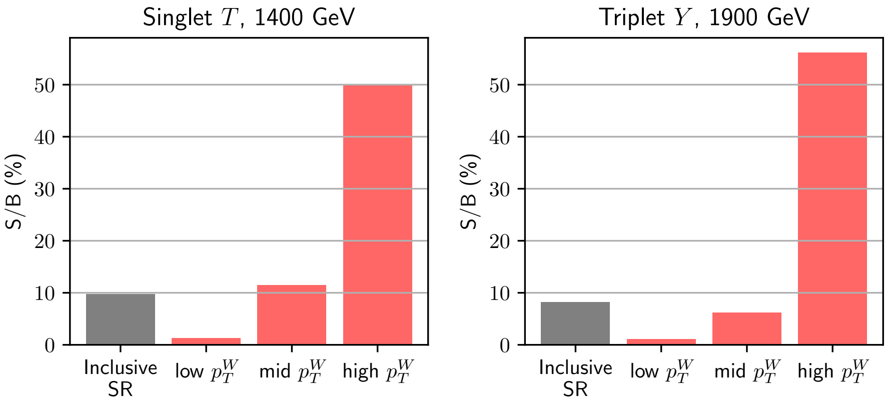


Figure 8.5: An example of the effect of the p_T^W splitting for two different VLQ models (both with $\kappa = 1/2$). After splitting, the high-p_T^W region always had a significantly higher signal-to-background (S/B) ratio than the old inclusive region, and the mid-p_T^W region was typically comparable very similar to the old inclusive region.

8.5 Data-driven reweighting

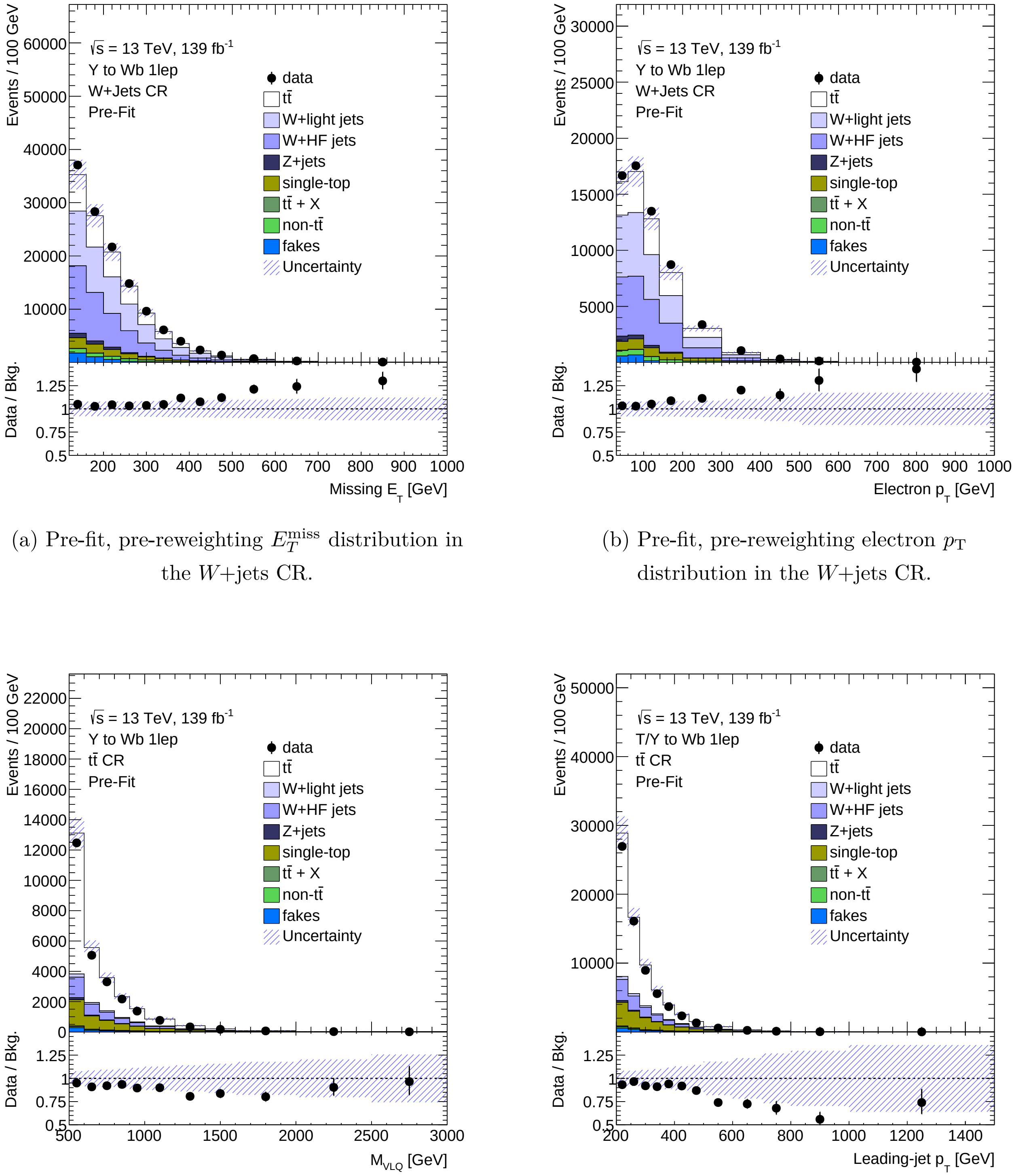
As shown in Figure 8.6, the pre-fit modelling in the W+jets and $t\bar{t}$ control regions is poor, suggesting some mismodelling in the W+jets, $t\bar{t}$, and possibly single-top MC samples. This is not entirely surprising, as we are using these large SM samples in an extreme corner of their phase-space. We will attempt to ameliorate this mismodelling with a two-step kinematic reweighting process.

The principle behind reweighting is to obtain a set of weights, as a binned function of some observables, that correct the event yields in the signal and control regions. These weights are calculated using a (relatively) pure reweighting region, where we can assume all data-MC discrepancies originate in the modelling of the sample we are trying to correct. The simplest reweighting is a single correction-factor that would normalise the MC yield in the reweighting region to the data yield, although this would fail to capture different physics effects across the phase space. Instead, we use a 2D reweighting for both $t\bar{t}$ and W+jets, which achieves a good compromise between, on the one hand, capturing the variety of physics across the region; and on the other, guarding against over-fitting to the reweighting region, and avoiding bins with very low populations.

We have already defined our reweighting regions in Table 8.2: note that the preselection region doubles as the $t\bar{t}$ reweighting region[9]; while this does mean that the SR is technically a subset of the reweighting region, it is such a small subset that it is highly unlikely that this would bias the results of the analysis. The W+jets reweighting region is significantly purer in W+jets than the $t\bar{t}$ reweighting region is in $t\bar{t}$ and single-top (over 90% vs approximately 80%); and therefore the W+jets reweighting is carried out first. Weights are obtained for the W+jets sample – noting that the light and HF contributions are combined into one sample for the purpose of reweighting – and then these are applied to the W+jets samples in the the preselection region. Then the $t\bar{t}$ reweighting is carried out on this already partly-corrected data. Because the W+jets contribution to the mismodelling has already been corrected, this means that "effective purity"[10] of the $t\bar{t}$ reweighting region also approaches 90%.

[9] Though for brevity we will refer to "the $t\bar{t}$ reweighting region" or to "$t\bar{t}$ reweighting", single-top is also included in this calculation and one reweighting factor is calculated for the combination of the two samples.

[10] i.e. ($t\bar{t}$ and single top total count) / (all uncorrected samples total count).



(a) Pre-fit, pre-reweighting E_T^{miss} distribution in the W+jets CR.

(b) Pre-fit, pre-reweighting electron p_{T} distribution in the W+jets CR.

(c) Pre-fit, pre-reweighting m_{VLQ} distribution in the $t\bar{t}$ CR.

(d) Pre-fit, pre-reweighting leading jet p_{T} distribution in the $t\bar{t}$ CR.

Figure 8.6: Pre-fit data-MC comparisons in the $t\bar{t}$ and W+jets control regions for several observables, *without* the application of any reweighting. Agreement is poor, particularly at higher energies.

Variable 1	Variable 2	
n_{jets}	p_T^W	✓
n_{jets}	E_T^{miss}	
n_{jets}	m_{VLQ}	
$p_T^{\text{leading-jet}}$	p_T^W	

Table 8.4: Possible variable combinations studied for W+jets reweighting. The ordering of Variable 1, Variable 2 is irrelevant. As indicated by the checkmark (✓), the (n_{jets}, p_T^W) combination was selected.

8.5.1 *W*+jets reweighting

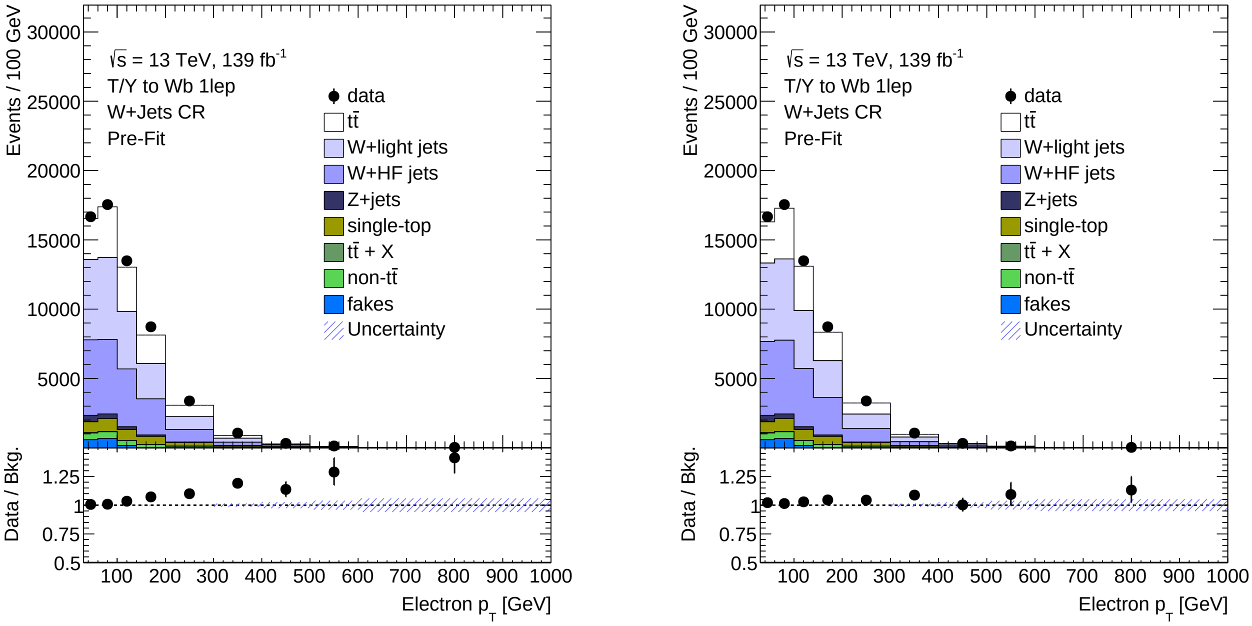
Before we can derive weights, we need to decide which variables we will use to bin the reweighting. For W+jets, several different variable pairings were considered, as listed in Table 8.4. Though splitting the reweighting region by m_{VLQ} is potentially problematic because it is also the fitted variable, the $(n_{\text{jets}}, m_{\text{VLQ}})$ combination was still considered because this had been used in the in the workflow of the original analysis.

As this reweighting procedure guarantees perfect closure in the reweighting regions, effectiveness is best tested in the relevant control regions – though all control and validation regions were examined to ensure no unphysical behaviour. Unfortunately, the small populations of the two W+jets validation regions made these less helpful in determining the reweighting strategy. Validation plots in the W+jets control region showed that reweighting in (n_{jets}, p_T^W) was the most effective. Although the two were otherwise broadly very similar, it outperformed (n_{jets}, E_T^{miss}) in reproducing lepton transverse momenta, as shown in Figure 8.7.

Intuitively ($p_T^{\text{leading-jet}}$, p_T^W) seemed to be an excellent choice because it involved two variables that contribute to the reconstruction of the VLQ mass and are also tied to easily identifiable physics objects in the W+jets final state. However, it actually performed very slightly worse in reconstructing $p_T^{\text{leading-jet}}$ itself, more strongly under-predicting the data in the 600 GeV – 1000 GeV range, as shown in Figure 8.8. While this very small difference alone would not be enough to reject this choice of variables, closer examination showed that this unexpected result occurred because in the W+jets reweighting region, the leading-jet p_{T} distribution is slightly harder in the data than in the MC, but the reverse is true in the preselection region. No orthogonal reweighting region will ever have exactly the same kinematic trends as the preselection region, but using variables with such obvious discrepancies to calculate weights that will be propagated to the preselection region would obviously be bad practice. This trend is not present in the p_T^W distribution.

The old $(n_{\text{jets}}, m_{\text{VLQ}})$ combination as expected performed poorly for many variables in the W+jets CR. Typically, it significantly underpredicted event counts at large energies and momenta, leading to large apparent "excesses" which, if also present in the SR[11], would be exactly where we would expect to see a signal if there was one. An example for E_T^{miss} is shown in Figure 8.9, which is broadly representative of several other similar distributions.

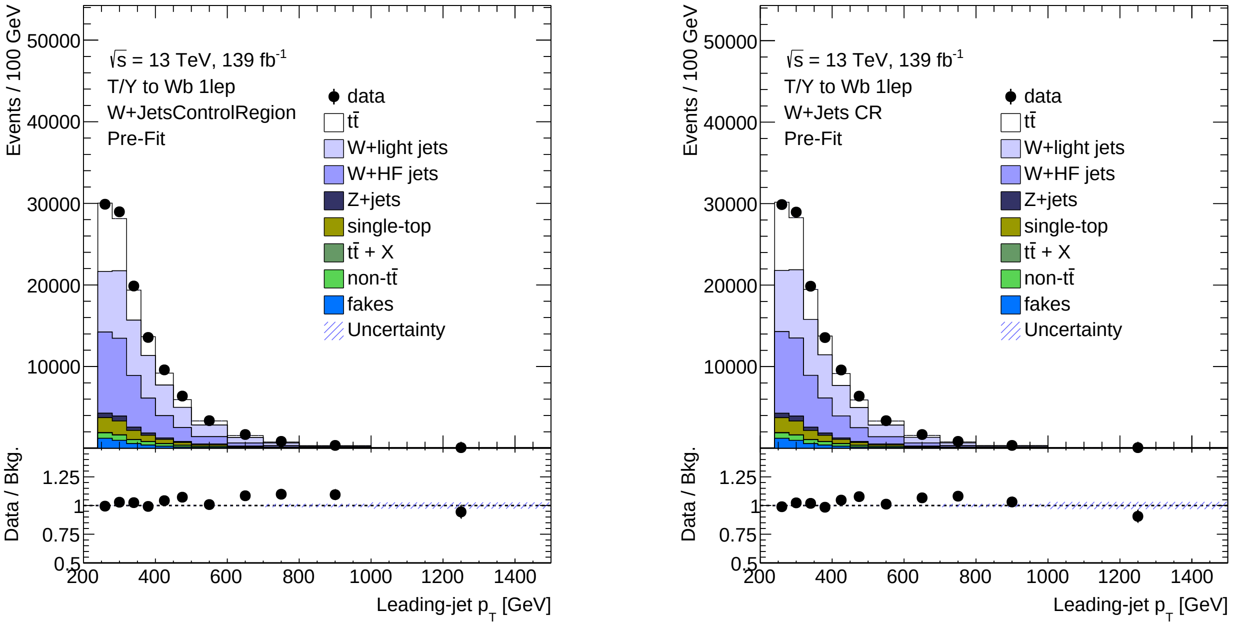
[11] Because of the blinding of the SR, this could not be checked directly.



(a)

(b)

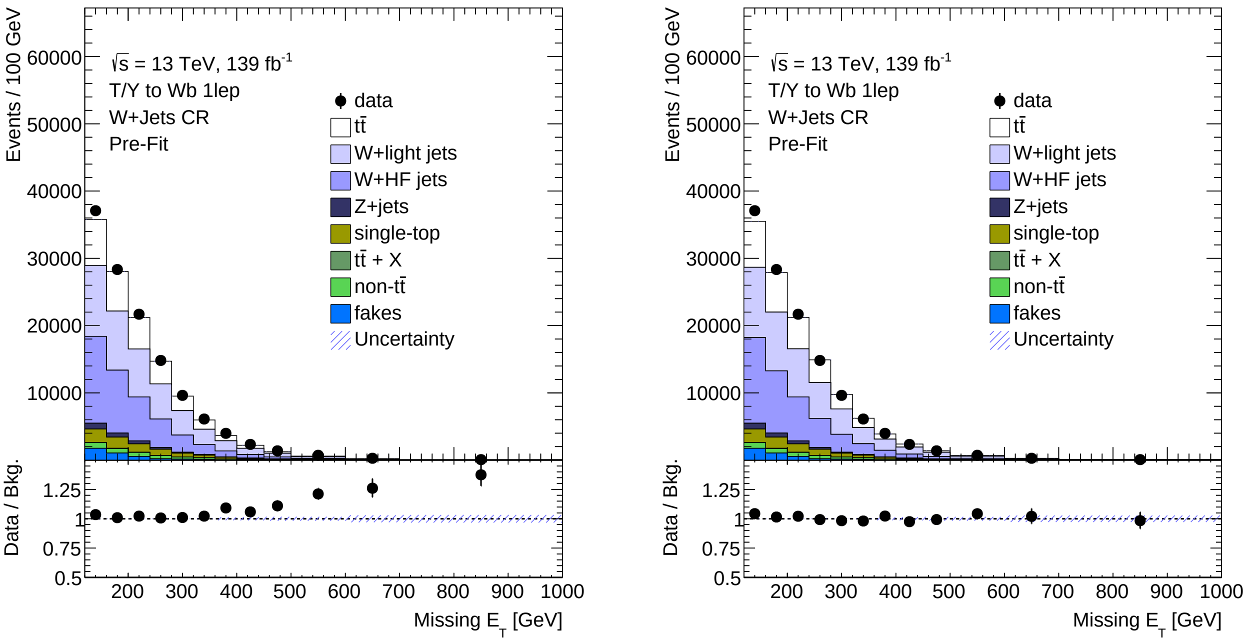
Figure 8.7: Comparing the results from (a), a 2D reweighting obtained using (n_{jets}, E_T^{miss}) to (b), a reweighting obtained using (n_{jets}, p_T^W) (b); plotting the electron p_{T} in the inclusive W+jets control region. This is one of the distributions where the weights obtained in a (n_{jets}, p_T^W) binning outperform those obtained with the (n_{jets}, E_T^{miss}) binning. Systematic errors are not included in this plot.



(a)

(b)

Figure 8.8: Comparing the results from a 2D reweighting obtained using ($p_T^{\text{leading-jet}}$, p_T^W) (a) to a reweighting obtained using (n_{jets}, p_T^W) (b); plotting the leading-jet p_{T} in the inclusive W+jets control region. Slightly surprisingly, the weights obtained in ($p_T^{\text{leading-jet}}$, p_T^W) perform very slightly worse, under-estimating the data in the 600 GeV − 1000 GeV range. Systematic errors are not included in this plot.



(a)

(b)

Figure 8.9: Comparing the results from a 2D reweighting obtained using (n_{jets}, m_{VLQ}) (a) to the reweighting obtained using (n_{jets}, p_T^W) (b); plotting the E_T^{miss} in the inclusive W+jets control region. The (n_{jets}, m_{VLQ}) scheme performs poorly for most observables, including this one, where the reweighted MC significantly under-predicts the data at high E_T^{miss}. Systematic errors are not included in this plot.

8.5.2 $t\bar{t}$ reweighting

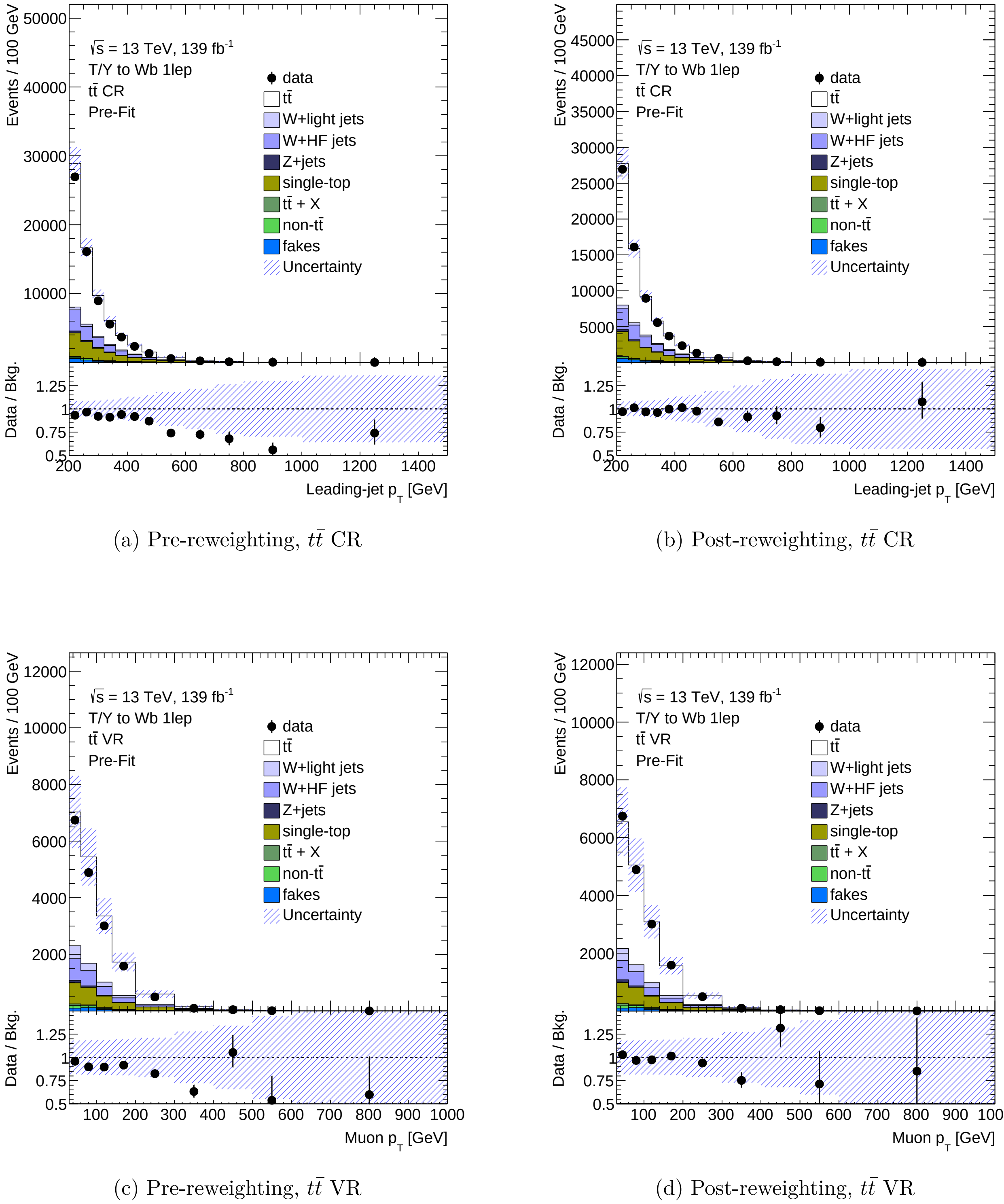
For $t\bar{t}$ reweighting, an earlier version of the analysis had used a 2D reweighting in leading-jet p_{T} and the number of jets. This was found to perform well, so no time was spent on re-optimising the reweighting variables from scratch, and the exisiting setup was used. Nevertheless, the weights still had to be recalculated given the new W+jets weights; satisfactory performance in the $t\bar{t}$ CR and VR – with the reweighted MC consistently within the error range – is illustrated in Figure 8.10.

8.5.3 Final configuration and systematic uncertainties

The final 2D weights for W+jets reweighting and $t\bar{t}$ reweighting are shown in Figure 8.11. Both are relatively smooth distributions, suggesting that we are capturing real physical effects rather than errors or random fluctuations. All bins contained a minimum of 200 events, which should minimise the statistical error on the reweighting. As a closure test, the weights were applied to events from the reweighting regions, and as expected perfect data-MC agreement was observed.

The 2D weights were stored as a Root macro (a series of if-statements dedicated to finding the correct bin and then returning the appropriate weight) and applied on-the-fly as the fit was conducted. This may sound like a small detail, but the increase in efficiency of this approach over inserting an additional weight into every analysis ntuple was significant, and without it the analysis may not have been completed.

These reweightings will not be perfect, and to account for this each reweighting incurred an additional systematic error. This is (overly) conservatively set at 100% of the reweighting, i.e. a 1σ deviation corresponds to not applying the reweighting. Although the same reweighting factors are applied to both $t\bar{t}$ and single-t production, exactly how the reweighting fails to fully capture the data is not likely to be the same for both



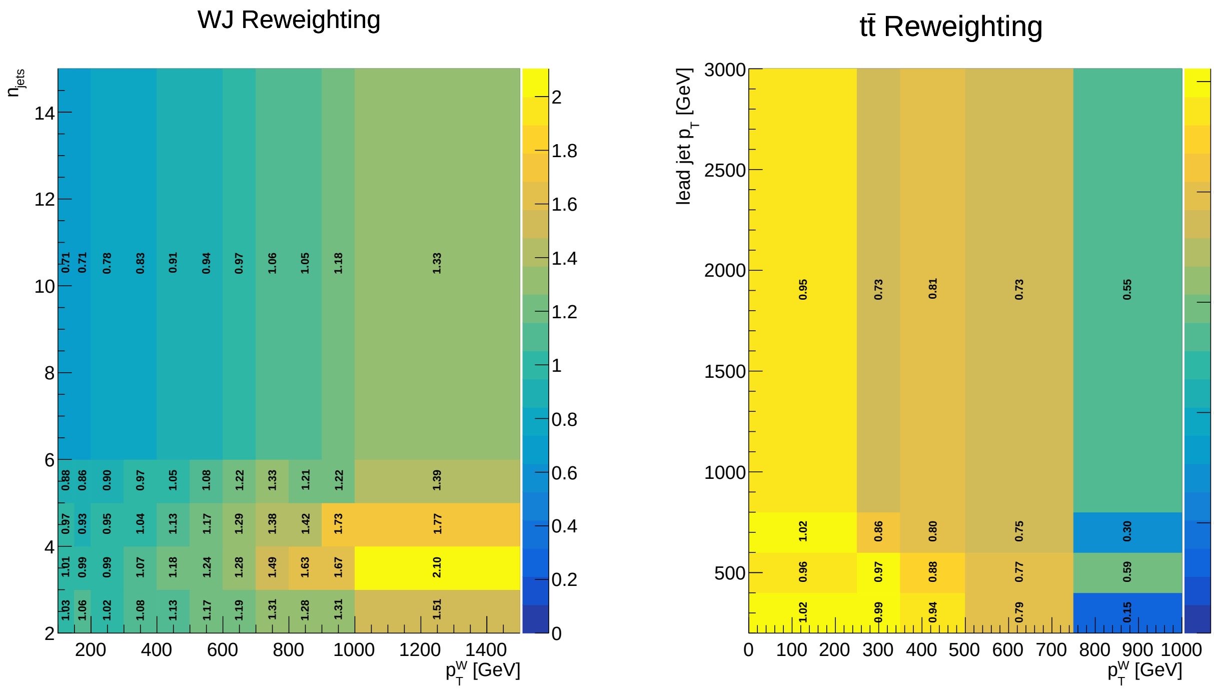
(a) Pre-reweighting, $t\bar{t}$ CR

(b) Post-reweighting, $t\bar{t}$ CR

(c) Pre-reweighting, $t\bar{t}$ VR

(d) Post-reweighting, $t\bar{t}$ VR

Figure 8.10: Pre- (left) and post- (right) $t\bar{t}$ reweighting results for two observables, in the $t\bar{t}$ CR (leading-jet transverse momentum, top) and VR (muon transverse momentum, bottom). The reweighting clearly improves the modelling.



(a) (b)

Figure 8.11: The final weights for W+jets (a) and $t\bar{t}$ (b) samples on their respective 2D grids. That the weight distribution is smooth (there is no significant "zig-zagging" in colour) is an indicator that we are capturing real physical effects.

samples and therefore these errors are not correlated in our fit. As for many other systematic uncertainties, these were also decorrelated across the three p_T^W slices.

8.6 Systematic uncertainties and statistical setup

We have already discussed some systematic uncertainties where appropriate above: for example, the generator-systematics and the reweighting uncertainties. In this section, we will introduce the experimental uncertainties and give additional information about other sources of systematic error where more detail is required.

8.6.1 Experimental systematic uncertainties

Experimental uncertainties in this analysis arise from the reconstruction and measurement of jets, leptons, and E_T^{miss}. The nature and physical origins of these terms have already been discussed in Chapter 2. The many different physical sources of error in the jet energy scale (JES) and jet energy resolution (JER) are separated into independent components to be varied in the fit, following the procedure laid out in Reference [295]. There are 30 independent components for the JES, and eight for the JER.

Errors are also included to account for mis-tagging during flavour tagging. As discussed in Section 2.7.3, correction factors between data and MC tagging rate are measured in $t\bar{t}$ and D^* meson events, and uncertainties on these measurements provide p_{T} and η dependent errors on the tagging, with six-independent components for b-jets and four independent components for c-jets.

Because the correction factors for heavy flavour jets are only valid up to 300 GeV, an additional systematic error is included to account for the uncertainty of extrapolating these factors to higher momenta. Because the analysis construction depends on a high-p_{T} b-jet, there were concerns that this systematic may be very significant. This is an interesting recapitulation of a theme we previously discussed in Section 5.5.1, where the limited transverse-momentum range studied in b-tagging calibration and efficiency studies (on that occasion for the MV2c10 tagger) was problematic for the reinterpretation of a SUSY search that depended on b-jets. Although very high-p_{T} b-jets are rare, they clearly can be signature of a variety of BSM physics models and as such, it would be useful going forward if calibration and efficiency studies considered a greater range. Nevertheless, fortunately for this analysis, this did not end up being significant in the final fit.

For many of the previously discussed theoretical uncertainties, decorrelating the systematic errors between the p_T^W regimes could be physically motivated. However, these arguments do not hold for experimental errors. Therefore, all the errors that have been introduced in this section vary consistently across the entire p_T^W range.

8.6.2 Free-floating normalisations, rescaling, and dedicated reweightings

As mentioned in Section 8.3, in order to help alleviate the modelling difficulties for $t\bar{t}$ and W+jets samples, the fit included three free-floating normalisation factors for the three samples, also described as K-factors[12]. This potentially created a multiple-redundancy in the fit, as several of the sample-specific systematic uncertainties (e.g. those originating from the generator and parton shower) could be decomposed to have a component whose only effect was the overall normalisation of the sample. While in principle this would not affect the minimum likelihood obtained by the fit, it may make numerical convergence in the minimiser more difficult, and it could also impact several goodness-of-fit metrics. To account for this, individual samples that provided up and down systematic variations that had a significant component parallel to the free-floating factors (summarised in Table 8.5) were normalised such that they had the same total yield across signal and control regions as the nominal they were being compared to. This does not leave these as “shape-only” systematics: varying them can still change the relative yields between different signal and control regions, just not the total yield across all the fitted regions.

Another effect that could have potentially affected theory systematics specific to the $t\bar{t}$ and W+jets samples was biasing from the reweighting described in Section 8.5. Applying weights calculated using the nominal sample could artificially increase the size of the uncertainty, and is not a fair comparison of the difference that would occur if that alternate sample had been used as the nominal. Therefore dedicated reweightings, using the same final choice of binning variables as described in Section 8.5, were calculated and applied for a certain subset of significant systematic variations potentially impacted by the reweighting, as shown in Table 8.5.

We also considered including a free-floating normalisation for the single-top sample. However, even with the promotion of the $t\bar{t}$ VR (which is approximately 15% single-top) to a control region, the background-only fits using Asimov data in the control regions could not reasonably constrain the normalisation, which had a post-fit uncertainty of over 80%[13].

Nevertheless, it was still noticeable that several single-top systematics (primarily those listed in Table 8.5) were encountering similar problems: they were all responsible for a large, pure-normalisation effect that again gave the fit a large amount of additional redundancy. This was resolved by normalising the three most significant single-top systematics, and then adding an additional normalisation correction systematic, equal to the largest normalisation correction – approximately 50%, from the choice of DR/DS scheme[14]. This

[12] They mimic the K-factors used to correct SM measurements to higher-order cross-sections, but in this case are data- rather than theory-derived.

[13] Compare to uncertainties of 8% – 15% for the W+jets and $t\bar{t}$ K-factors.

[14] This is not the same as a free-floating normalisation, because it has a pre-fit Gaussian penalty term to constrain it, the scale of which is dictated by a “real” systematic.

decorrelated the overall normalisation uncertainty of these systematic variations from their region-to-region and intra-region effects.

Uncertainty	Rescaling	Single-top rescaling	Dedicated reweighting
$t\bar{t}$ generator comparison	✓	–	✓
$t\bar{t}$ parton-shower comparison	✓	–	✓
$t\bar{t}$ FSR/ISR	✓	–	✓
$t\bar{t}$ μ_R/μ_F	✓	–	✓
Single-top generator comparison	–	✓	✓
Single-top parton shower comparison	–	✓	✓
Single-top Wt DR/DS	–	✓	✓

Table 8.5: List of the uncertainties with the normalisation rescaling and/or dedicated data-driven reweighting applied.

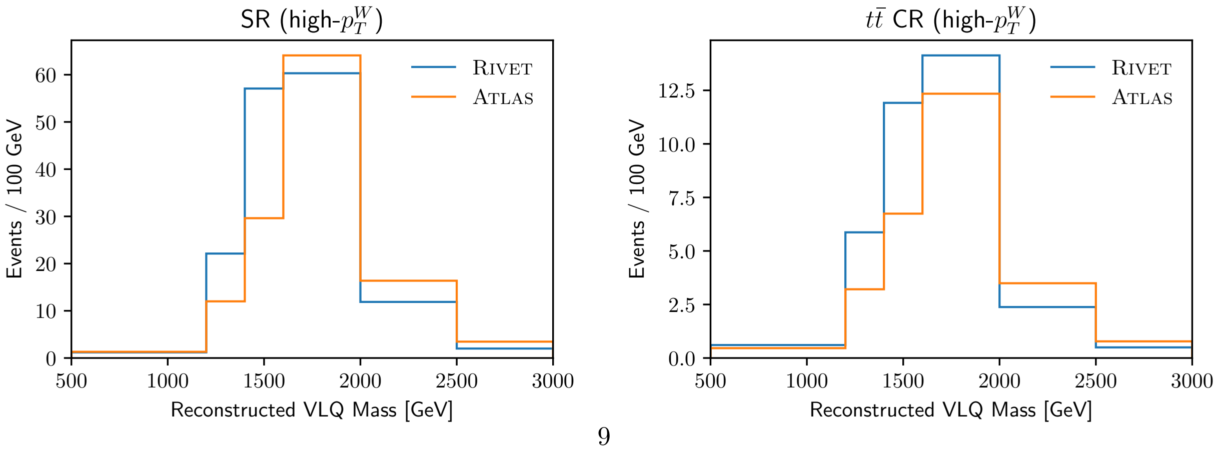
8.6.3 CKKW and QSF uncertainties for the W+jets sample

For the W+jets background samples generated with Sherpa 2.2.11, there are two MC-generator uncertainty contributions that we did not discuss in Section 8.3, in order to give a fuller account here. These come from the choice of two event-generator variables: the CKKW factor, a merging-related parameter that is approximately equivalent to the cut-off in GeV below which a jet is handled by the parton shower; and the resummation scale, QSF. Atlas evaluates the impact of these uncertainties using MC samples generated using alternate values either side of the nominal: 15 GeV and 30 GeV (either side of 20 GeV) for the CKKW parameter, and 0.25 and 4 (either side of 1) for the QSF.

Chapter 3 laid out the computational costs associated with detector simulation. To avoid incurring these costs, Atlas only generated the alternate CKKW and QSF samples at particle-level, with no simulation (even AtlFast) applied. Therefore, to evaluate the impact of these uncertainties, we must evaluate their impact on a particle-level approximation of our analysis regions, and propagate it back to our detector-level fit.

The natural tool for evaluating differences at particle level is Rivet. Therefore, we wrote a Rivet implementation of the analysis. Figure 8.12 shows validation plots for the high-p_T^W signal and $t\bar{t}$ control regions, confirming that the Rivet implementation is a reasonable, albeit imperfect, reproduction of the full detector-level search. Once the analysis is published, this Rivet code will also provide useful reinterpretation material that ensures the results from this search will be re-usable for future phenomenologists. Brief investigations suggested that the approximately 100 GeV offset between the Rivet and Atlas m_{VLQ} distributions most likely originates from the simplified lepton reconstruction and smearing in Rivet, although b-jet emulation may also play a role. Ideally, for reinterpretation purposes in particular, this discrepancy would be better understood and made consistent.

The original plan for propagating these uncertainties from the Rivet analysis was to replicate the m_{VLQ} histogram for each of our regions, and propagate the up and down variations as relative errors bin-by-bin. Unfortunately however, due to additional efforts on the part of Atlas to optimise computational efficiency, the alternate samples contained only 50 million MC events each (compared to 2 billion in the nominal sample). While this is likely more than sufficient for SM searches with high acceptances, it is inadequate for our search: in the high-p_T^W signal region, we observed as few as 65 total MC events. As Figure 8.13 shows, this means that the bin-by-bin relative uncertainty in all three fitted regions is entirely dominated by the statistical error



(a)

(b)

Figure 8.12: Validation of the Rivet analysis using a 1700 GeV sample at $\kappa = 0.5$, displaying the reconstructed VLQ mass distribution in the high-p_T^W slices of the signal region and $t\bar{t}$ control region. The reproduction is satisfactory, though in both cases the Rivet implementation is shifted lower by approximately 100 GeV, albeit still producing very "signal-like" distribution shapes.

on the MC samples, rather than by the systematic uncertainty itself.

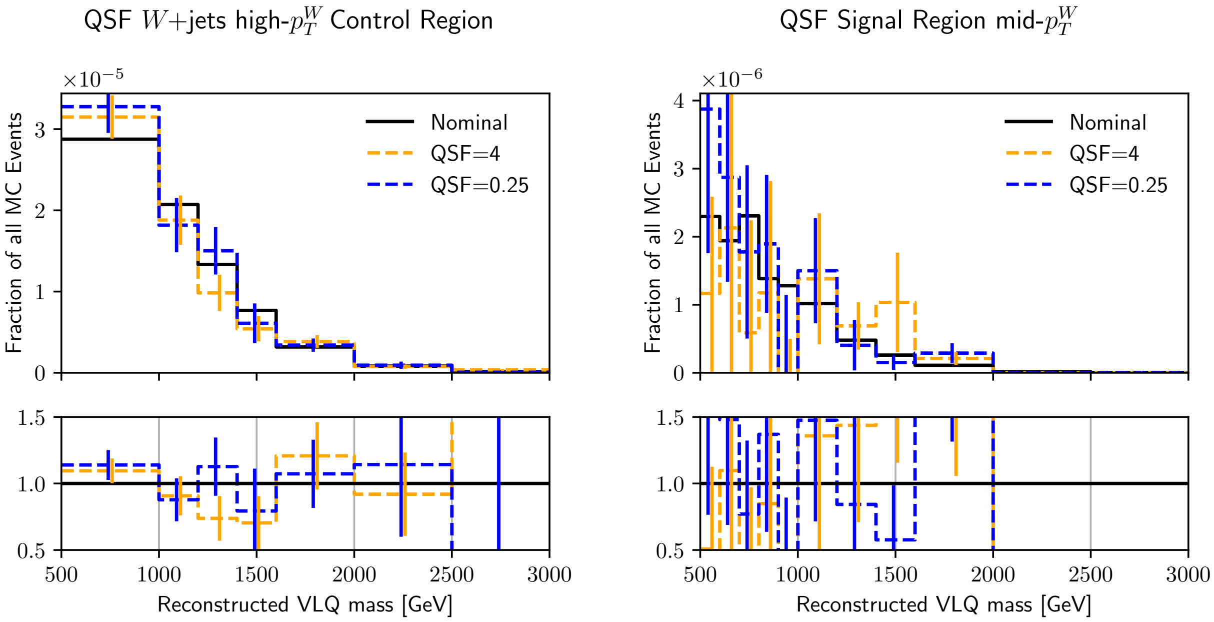
Instead we decorrelated the systematics into "shape" and "acceptance" components. The acceptance components control the region-to-region shifts. There are (just about) sufficient statistics to evaluate this for all three p_T^W-inclusive signal and control regions. Because the analysis already includes a free-floating normalisation constant for the W+jets component, these shifts were normalised so that there was no change to the total number of events across the combination of all signal and control regions. There were not sufficient statistics to calculate these separately for the low-, mid-, and high-p_T^W slices[15]; but just in case there was some variation with p_T^W, the acceptance systematics were decorrelated between the slices, to allow the fit to change this if necessary.

The shape component of the CKKW and QSF uncertainties controls the movement across the m_{VLQ} distribution within each signal region. In order to obtain a usably small statistical error in each bin, this was evaluated in the QSF/CKKW shape region – a new region, defined to be the same as the preselection region, but without the b-tag requirement. Because not all events in this new region contain b-jets, where necessary the VLQ candidate was reconstructed from the W and the leading jet. This effectively adds the W+jets reweighting region to the preselection region and gives us a much higher acceptance of W+jets events. Therefore, as shown in Figure 8.14, the statistical errors are sufficiently small that a meaningful estimate of the theoretical shape uncertainty can be obtained for the QSF.

Indeed, we had sufficient statistics for the QSF variations that we could obtain shape variation separately for the W+light and W+HF samples, as these had already been decorrelated elsewhere in the fit. The QSF variations were approximately a factor of two larger than the CKKW (i.e. the difference between the nominal and variation is consistently about twice as large). This gave us more leeway for a higher MC statistical error. We exploited this to obtain the CKKW shape variation separately for the W+light and W+HF samples, as these had already been decorrelated elsewhere in the fit.

For the CKKW variations, the shape uncertainty was (after normalisation) statistically indistinguishable from zero in every bin, and therefore a shape term was not included.

[15] Though the results were entirely statistically consistent with there being no change as a function of p_T^W.



(a)

(b)

Figure 8.13: The statistics of the available MC samples for the QSF and CKKW systematics variations were totally inadequate to apply the uncertainty bin-by-bin. As representatives of all the others, we illustrate the QSF (the larger, more statistically significant uncertainty) for two regions: the mid-p_T^W W+jets CR (a) – noting that the W+jets sample is the largest contributor to this region – and the mid-p_T^W SR (b).

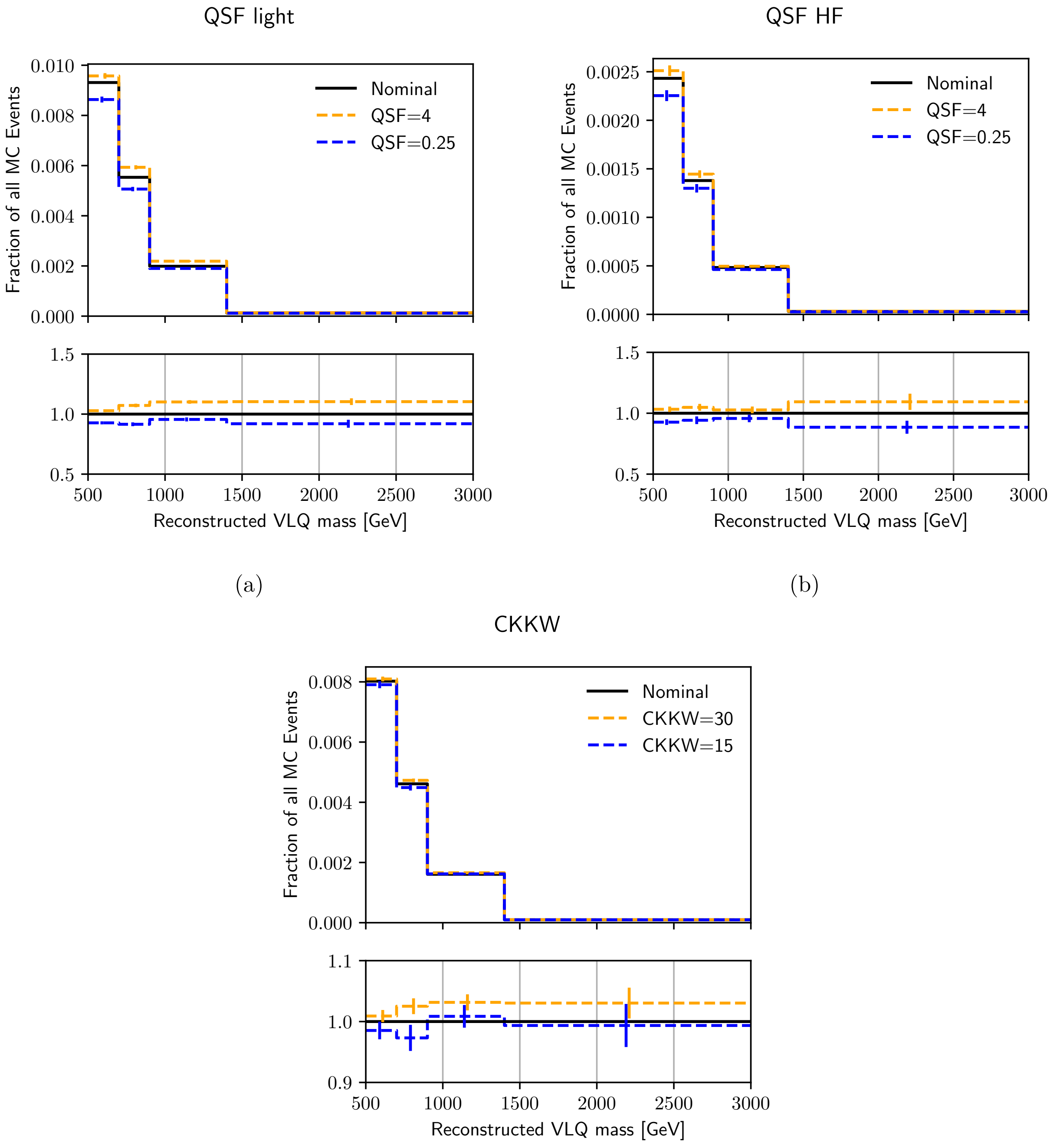
Ideally, going forward for Run 3, these uncertainties could be better evaluated for BSM searches. While forcing search analyses to go to the additional effort of producing a Rivet implementation is arguably good from a reinterpretation perspective[16], providing only 50 million alternate events is likely to be inadequate for most BSM searches, and possibly may even have led to the over-estimation of this uncertainty in previous analyses.

8.6.4 Final fit setup

The statistical core of the analysis is a profile-likelihood fit, as described in Section 4.1.6. There are nine fitted regions (3 p_T^W slices across the SR, $t\bar{t}$ CR, and W+jets CR, as defined in Table 8.3), and in each region we fit the reconstructed VLQ mass (m_{VLQ}) distribution. The tests that will be presented in Section 8.7.2 for discovery used the q_0 statistic from equation 4.6; cross-section limits that will be presented in Section 8.7.3 used the $\tilde{q}_\mu$ statistic from equation 4.7; and the 2D-exclusion contours used a CL$_s$-value calculated for the $\mu = 1$ case, calculated using the HistFactory conventions laid out in Section 4.1.5. To reduce the computational complexity of the fit, systematic errors smaller than 0.5% were "pruned" to save the minimiser spending a lot of CPU-resources on optimising the value of an almost irrelevant nuisance parameter.

Preliminary tests using Asimov data in blinded regions suggested that the nuisance parameters associated with several modelling uncertainties would be very strongly constrained in the final fit. To reduce this, most $t\bar{t}$, single-top and W+jets nuisance parameters were decorrelated into "shape" and "acceptance" components, in a process similar to that described in Section 8.6.3. Tests fitting to control regions only suggested that this

[16] Though for it to be useful, these analyses actually need to be made *public* after they have been written and used internally – the Atlas searches that have published Rivet code in some form is a small subset of those that claim to use QSF or CKKW uncertainties on Sherpa 2.2.11 samples.



(a)

(b)

(c)

Figure 8.14: The QSF shape uncertainty for the light (a) and HF (b) W+jets samples, as well as the CKKW shape uncertainty for the inclusive sample (c). The normalisation has not been applied; after normalisation both the up and down uncertainties for the CKKW uncertainty are indistinguishable from zero.

gave the fit the freedom to model the data without introducing very strict constraints on, for example, the choice of $t\bar{t}$ generator. This has become a common procedure in ATLAS analyses when systematic uncertainties from modelling become strongly constrained [296]. For fits involving interference terms, the amplitude of the interference term I will be affected by the signal strength μ relative to the signal S and background B as

$$\mu \cdot S + \sqrt{\mu} \cdot I + B \, . \tag{8.3}$$

In practice, this means that the parameter-of-interest that the minimiser will vary is $\sqrt{\mu}$, not μ itself, and therefore care must be taken when propagating errors on μ. This also means that we are explicitly constraining ourselves to positive μ, and indeed positive $\sqrt{\mu}$, as equation 8.3 is not physically valid if the coefficient of $\sqrt{\mu}$ undergoes a sign-flip.

8.7 Results

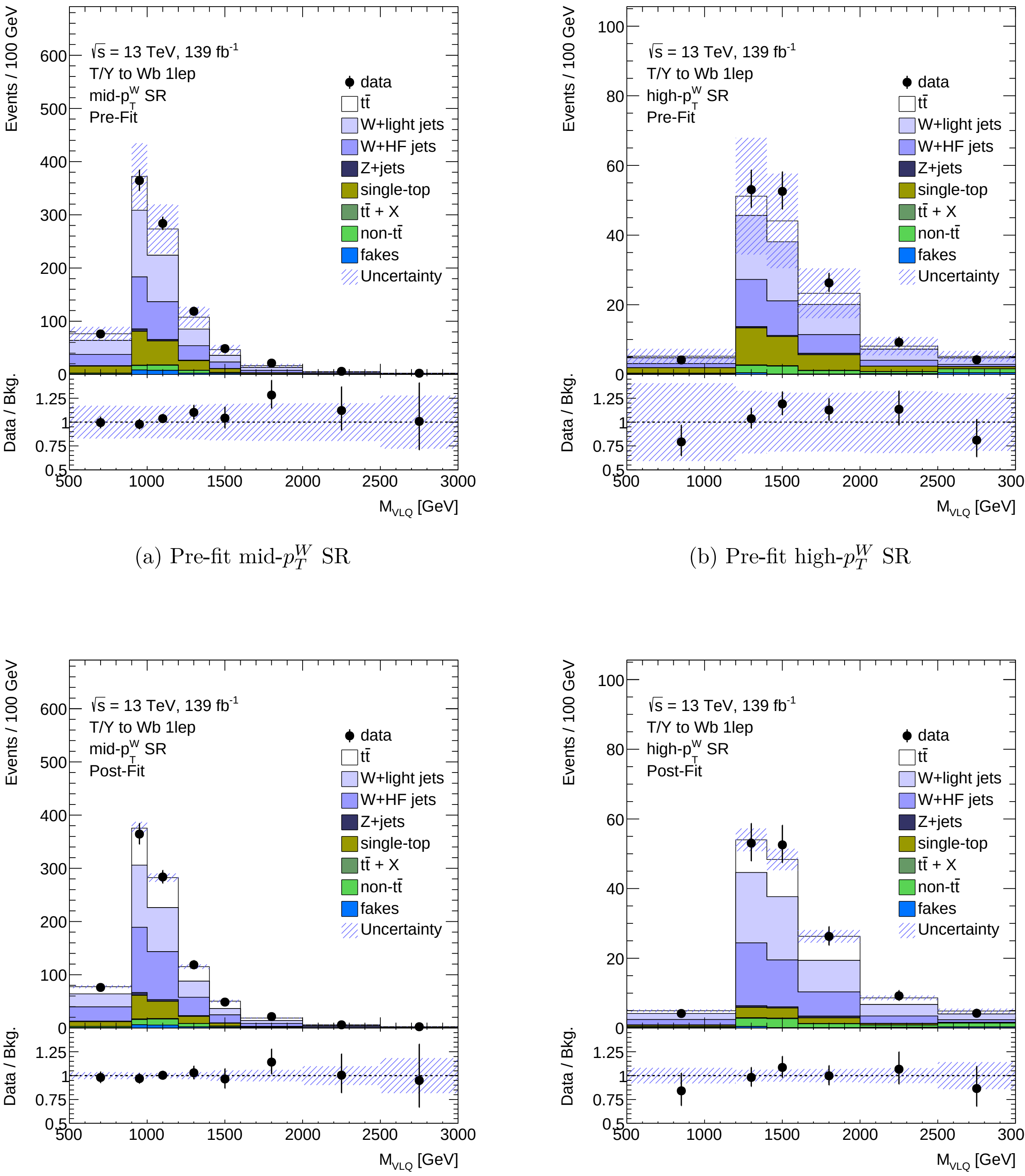
8.7.1 Background-only and benchmark-signal statistical fits

To explore the quality and features of the fit before drawing firm physics conclusions, we will consider a background-only fit, and a fit to 1.6 TeV and 2.4 TeV singlet-T VLQ models, as examples to examine more closely. These were chosen to give a comparison across the mass range, as the fit characteristics were similar for the T and the Y. The background-only pre- and post-fit distributions of m_{VLQ} in the mid- and high-p_T^W signal regions, are illustrated in Figure 8.15. The background-only fit does not suggest any excesses which cannot be reasonably explained by the SM. In the 1.6 TeV signal fit, shown in Figure 8.16, we see that the signal contribution has been significantly reduced, but not to zero – the post-fit value of μ was 0.24 ± 0.15. The remaining signal is permitted due to a very small excess in the 1600 GeV $< m_{\mathrm{VLQ}} <$ 2000 GeV bin of the mid-p_T^W SR, and a series of even less significant excesses in the 1400 GeV−2500 GeV range of the high-p_T^W SR.

For the fit to a 2.4 TeV singlet-T model – illustrated in Figure 8.17 – the signal is reduced much more signficantly, albeit it with a very large uncertainty – post-fit, $\mu = 0.04 \pm 2$. Because the reconstructed m_{VLQ} distribution peaks at higher masses for heavier signal models, they cannot be fitted to the small excess in the middle of the reconstructed mass range of the mid-p_T^W SR as effectively. The large uncertainty on μ is not concerning: it merely reflects that it is hard to place a limit on a very small pre-fit signal contribution; and, as shown particularly clearly for the mid-p_T^W region in Figure 8.17c, the single-production cross-section at 2.4 TeV leads to very low SR populations.

The only nuisance parameter pulled beyond 1σ in the background-only fit was the acceptance component of the $t\bar{t}$ parton shower systematic in the low-p_T^W slice, which was pulled to 1.4σ. This suggests that in the low-p_T^W region, the region-to-region migration effects are captured best by a very HERWIG like shower. This seems to reduce progressively with p_T^W – the pull is 0.5 for the mid-p_T^W slice and 0.2 for the high-p_T^W slice – and is therefore unlikely to have much impact on the fitting of the signal, which is most significant in the high-p_T^W slice.

For the signal models, we can illustrate the effects of the various nuisance parameters on the fit using a “ranking-plot”. This measures the “impact” of each systematic uncertainty by measuring how much a 1σ shift would change the measured value of the parameter-of-interest, i.e. μ, or in this case $\sqrt{\mu}$. It overlays this information with the pulls (measured on the lower x-axis), to give an intuitive picture of which nuisance parameters made the most difference in the fit. We provide ranking plots of the ten most impactful parameters for the fits to the 1.6 TeV and 2.4 TeV signal samples in Figure 8.18. These show that the majority of the most impactful nuisance parameters are related to $t\bar{t}$ and single-top theory uncertainties, though W+jets



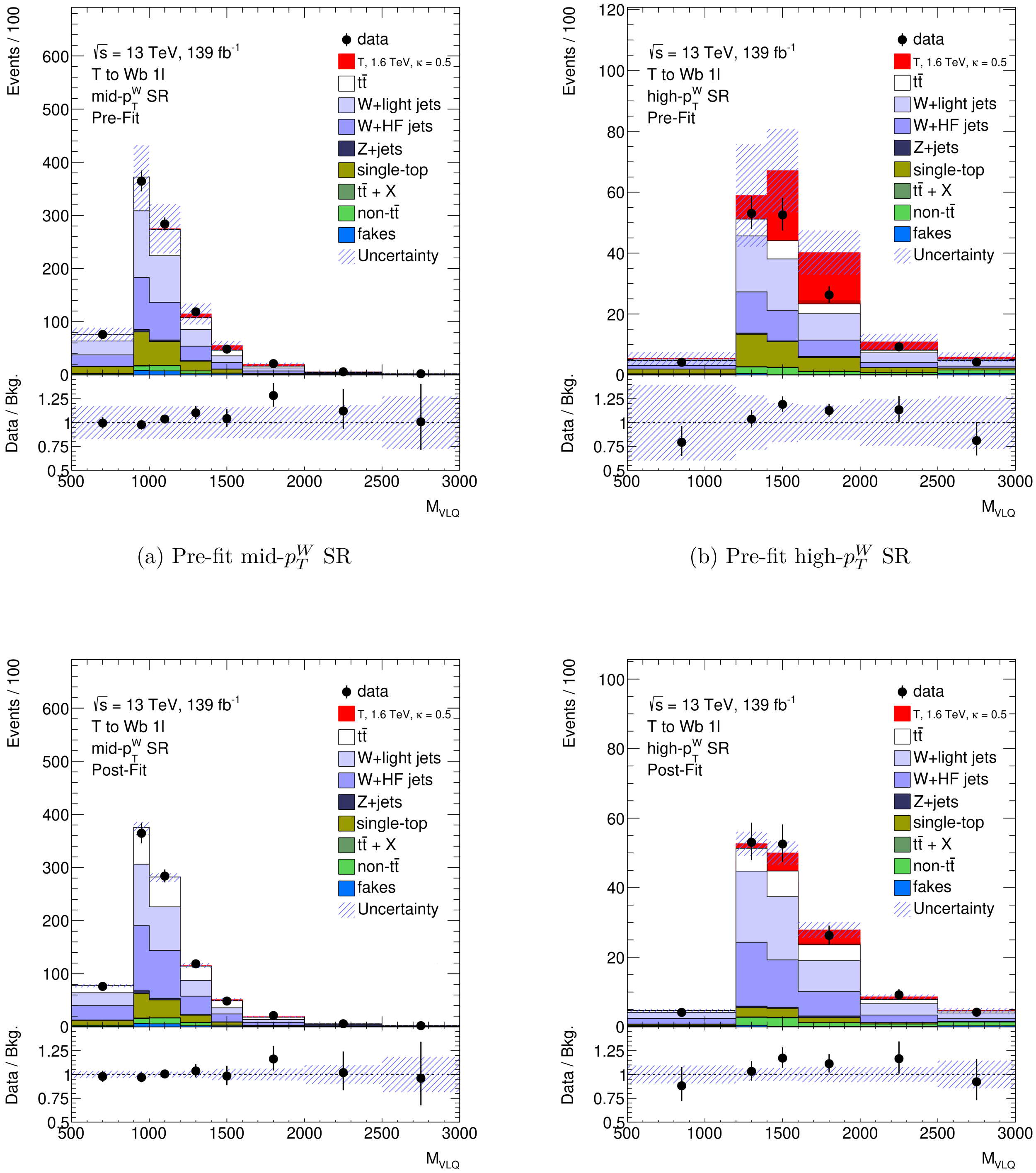
(a) Pre-fit mid-p_T^W SR

(b) Pre-fit high-p_T^W SR

(c) Post-fit mid-p_T^W SR

(d) Post-fit high-p_T^W SR

Figure 8.15: Pre- and post-fit m_{VLQ} distribution in the mid- and high-p_T^W signal regions for a background-only fit. There are no significant excesses, despite the hint in the 1600 GeV $-$ 2000 GeV bin in the mid-p_T^W SR.



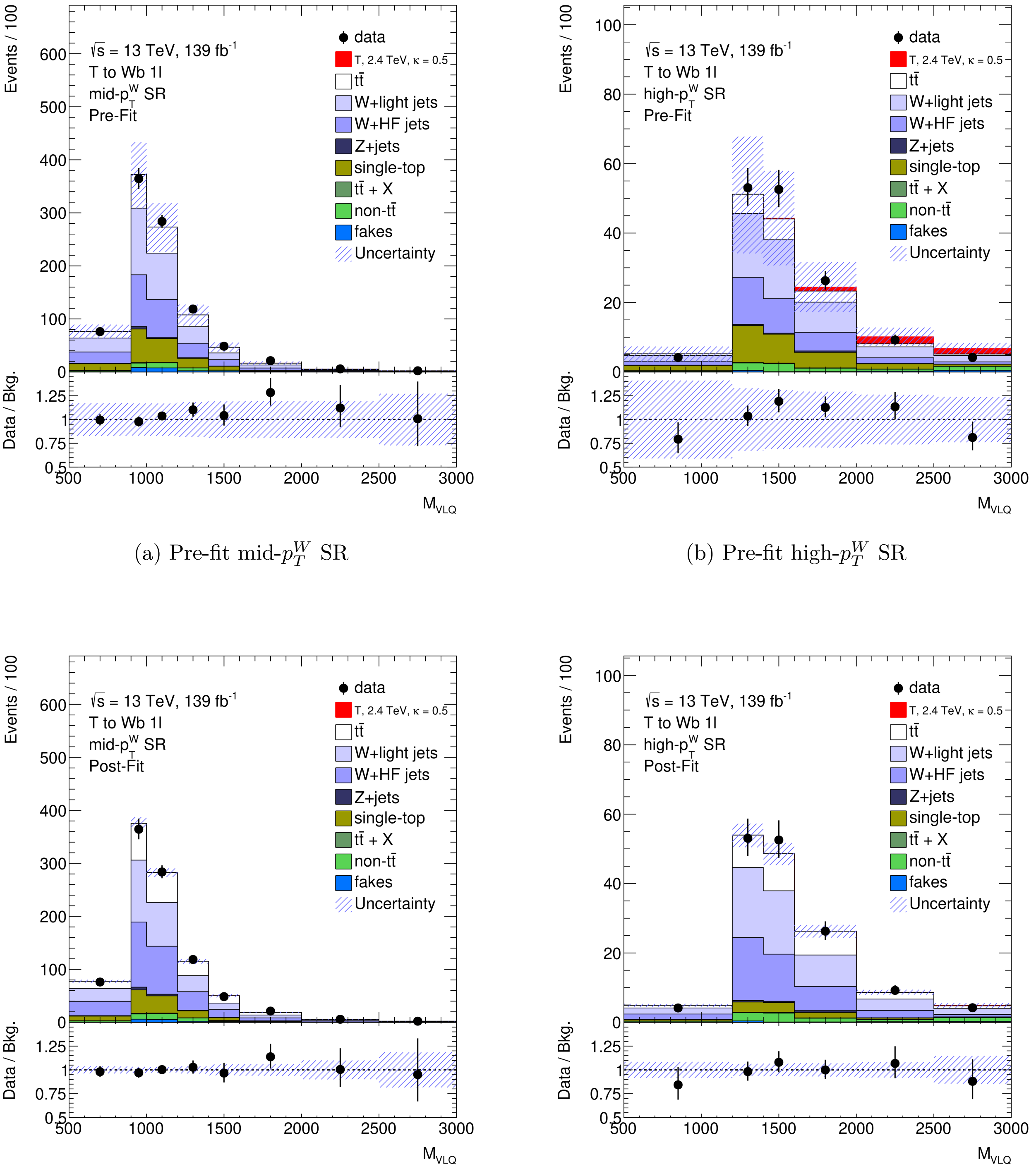
(a) Pre-fit mid-p_T^W SR

(b) Pre-fit high-p_T^W SR

(c) Post-fit mid-p_T^W SR

(d) Post-fit high-p_T^W SR

Figure 8.16: Pre- and post-fit m_{VLQ} distribution in the mid- and high-p_T^W signal regions for a fit to a 1600 GeV singlet-T model (including interference effects). Note that the ratio plots are data over background, i.e. do not include the signal contribution, in order to make any excesses more visible. The fit reduces the signal contribution, but not to zero (post-fit, μ was 0.24). The remaining signal is allowed due to a very small excess in the 1600 GeV − 2000 GeV mid-p_T^W bin, and even smaller and less statistically significant excesses in 1400 GeV − 2500 GeV bins in the high-p_T^W bin.



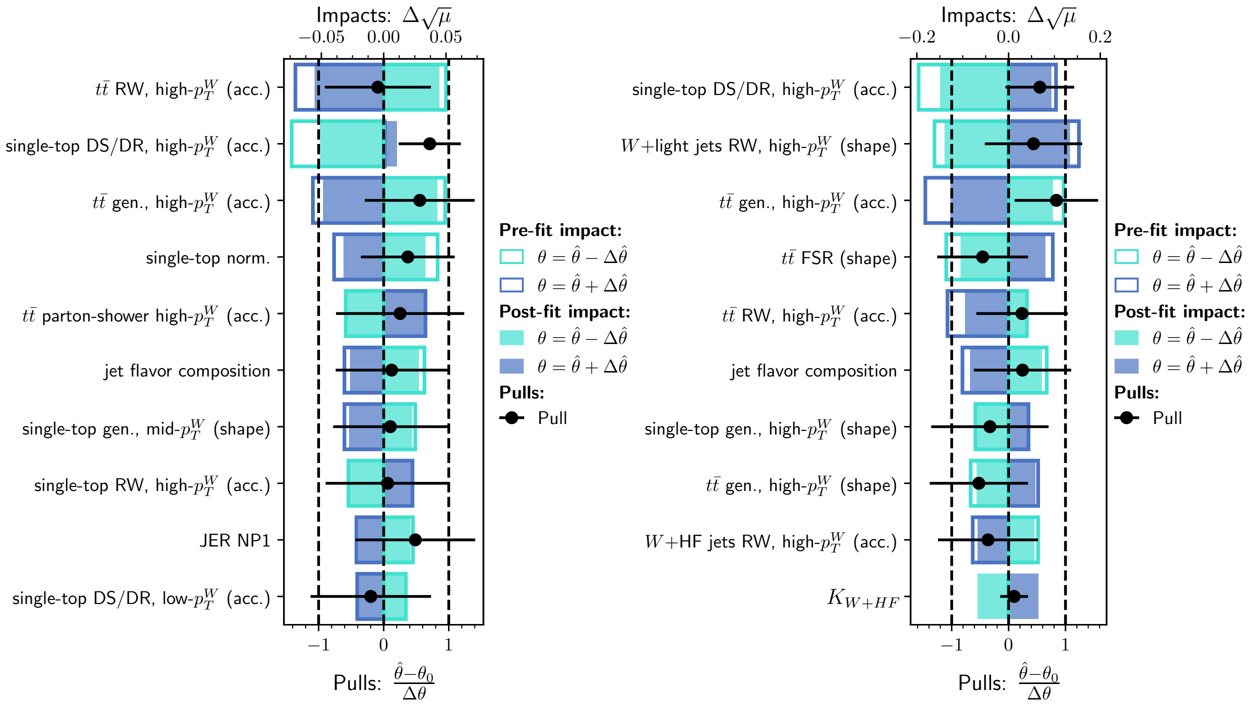
(a) Pre-fit mid-p_T^W SR

(b) Pre-fit high-p_T^W SR

(c) Post-fit mid-p_T^W SR

(d) Post-fit high-p_T^W SR

Figure 8.17: Pre- and post-fit m_{VLQ} distribution in the mid- and high-p_T^W signal regions for a fit to a 2.4 TeV singlet-T model (including interference effects). Note that the ratio plots are data over background, i.e. do not include the signal contribution, in order to make any excesses more visible. The fitted value of μ was 0.04.



(a) Ranking plot for a 1.6 TeV T-singlet model.

(b) Ranking plot for a 2.4 GeV T-singlet model.

Figure 8.18: Ranking plots for a 1600 GeV (a) and 2400 GeV (b) T-singlet model, showing the ten most impactful systematics. The rectangular blocks measured on the upper axis shows the "impact": how much a 1σ variation in the nuisance parameter changes the value of the parameter-of-interest. Meanwhile, the black dots and whiskers illustrate the pull: how much the best-fit value $\hat{\theta}$ deviates from the pre-fit value θ_0 (normalised by the width of the penalty term $\Delta\theta$). "JER NP1" is the first of the eight independent jet energy resolution nuisance parameters; $K_{W+\mathrm{HF}}$ is the free-floating normalisation for W+HF jets; "shape" and "acc" in brackets refer to the shape and acceptance decorrelation scheme discussed in Section 8.6.4. The most impactful systematic uncertainties are theory-based $t\bar{t}$ and single-top uncertainties. The apparent tight constraint on $K_{W+\mathrm{HF}}$ is just a visualisation issue, because this nuisance parameter had a uniform pre-fit penalty distribution, so any post-fit constraint at all appears very strong.

theory uncertainties begin to play a larger role at higher mass points. This is not necessarily unexpected: $t\bar{t}$ and single-top modelling are known to be difficult to model [297].

Figure 8.18 also shows that where systematics were decorrelated across the three p_T^W-slices, the high-p_T^W nuisance parameter was typically the most impactful. This makes sense – as Figure 8.5 shows, these are acting in the slice with the highest S/B ratio, and so a small change in the background counts will have a larger effect on the fitted value of μ.

Overall, none of the nuisance parameters in Figure 8.18 are very strongly constrained – the apparent strong constraint of the W+HF jets K-factor is just a consequence of its pre-fit uniform distribution. In fact, all three K-factors in all three fits were within a standard deviation of unity, showing that the freedom to fit this variable without a penalty term has not been "abused" by the fit. Unsurpisingly, there was a relatively strong negative correlation between the W+HF and W+light K-factors. Of the impactful systematics, the strongest constraint is on the high-p_T^W DS/DR nuisance parameter; but this is not too surprising given the size of the difference between the nominal and the variation in the high-p_T^W region. The absence of any highly-impacting but over-constrained nuisance parameters suggests that the fit is stable.

Also worth remarking on is the highly asymmetric impact of the high-p_T^W DS/DR uncertainty in Figure 8.18a. This should not be troubling: recall that this is a "two-point" systematic with an unphysical down variation: it is not surprising that this unphysical variation might have strange and significant effects on the fitted value of μ. As we would hope, however, the fit does not go in this direction – the high-p_T^W DS/DR nuisance parameter is pulled in the up-direction to almost $+1\sigma$. This merely implies that at high-p_T^W the diagram-subtraction scheme provides a better representation of the data than the diagram-reduction scheme. In fact, a similar (albeit less pronounced) effect can also be seen in Figure 8.18b. In an example of one of the weaknesses of the two-point variation approach, the influence of the unphysical down variation (which, all being well, the fit will never "choose") causes the ranking plot to overemphasise the impact of the DS/DR uncertainty.

8.7.2 Discovery

The first goal of a BSM search is to test for the discovery of a BSM particle. The results at the example point above (particularly at 1600 GeV), where the fitted signal was clearly non-zero, provide further motivation. Therefore, we carried out a test of local discovery significance – based on the test statistic $\tilde{q}_0$ defined in equation 4.10. The results across the mass range are shown in Figure 8.19. The largest local significance is 1.4σ, with similar values occuring for a range of κ values at 1500 GeV. The observed local significance is similar for all three signal models, though the Y-triplet is consistently very slightly higher, perhaps suggesting that the signal shape without interference can be better fitted to the data.

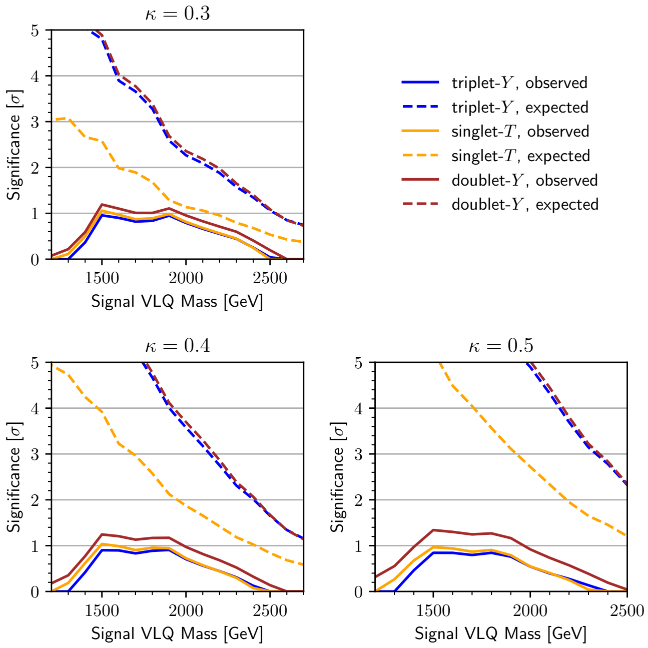


Figure 8.19: The expected (i.e. if the data corresponded to $\mu = 1$) and observed local significance for all three signal models across the entire valid mass range for κ values of 0.3, 0.4 and 0.5. Other than at the lowest κ value, the observed significance is consistently much lower than the expected significance. The expected significance for vector-like Y quarks is larger because their production cross-section is approximately twice as large. κ-values greater than 0.5 are not plotted because the Γ/M constraint (visualised in Figures 8.21 and 8.22) excludes the majority of the mass range; values below 0.3 are not plotted because the expected sensitivity is so low that the results are meaningless.

However, Figure 8.19 also shows that, with the exception of low-κ regions where the expected sensitivity is minimal, the observed significance is consistently and significantly lower than the "expected" significance for $\mu_{\mathrm{VLQ}} = 1$. Therefore, in the next section we expect to be able to exclude even many of the parameter points with a local significance greater than one.

8.7.3 Exclusion and comparison to other studies

A 1D-limit on the cross-section of the Y-doublet (which decays 100% to Wb) is shown in Figure 8.20 for $\kappa = 0.5$. Because the Y-doublet is unaffected by interference, it is easy to adjust this plot for either VLQ (without interference) at any value of κ and for any branching ratio to W, simply by multiplying the cross-section through by a constant factor. This makes this a very useful plot for phenomenologists to reuse.

The most glaring feature of all the limit plots – Figures 8.20 – 8.24 – is the extended section of less-than-expected exclusion between 1500 GeV and 2000 GeV, which corresponds to the region of non-zero discovery significance in Figure 8.19. Given the mass range, the under-exclusion likely originates with the same very small excesses in the middle of the m_{VLQ} range seen for the 1.6 TeV fit in Figure 8.16; comparing to the 2.4 TeV-fit then also explains why the exclusion returns to the expected value for VLQ masses above 2 TeV.

2D limits in the mass-kappa plane are shown in Figure 8.21 for the singlet-T VLQ, and in Figure 8.22 for the triplet-Y VLQ. Unsurprisingly we again see that the exclusion is weaker than expected, by slightly more than one σ, in the same region as for the doublet-Y. Because event generation is unreliable when the decay-width of the VLQ is more than half its mass, ATLAS by convention excludes these regions from the 2D limits.

Figure 8.23 also presents a 2D-limit plot for the singlet-T in the $(\Gamma_T/m_{\mathrm{VLQ}})$ vs m_{VLQ} plane. Reparametrising to include the total width is useful for phenomenologists studying models where the VLQ can decay to particles other than the three massive SM bosons, such as the scalar DM model mentioned in Section 1.3.4. As more of the "conventional" VLQ phase space is excluded, such models are likely to become increasingly of interest, and so it is important for the experimental community to provide the results in a

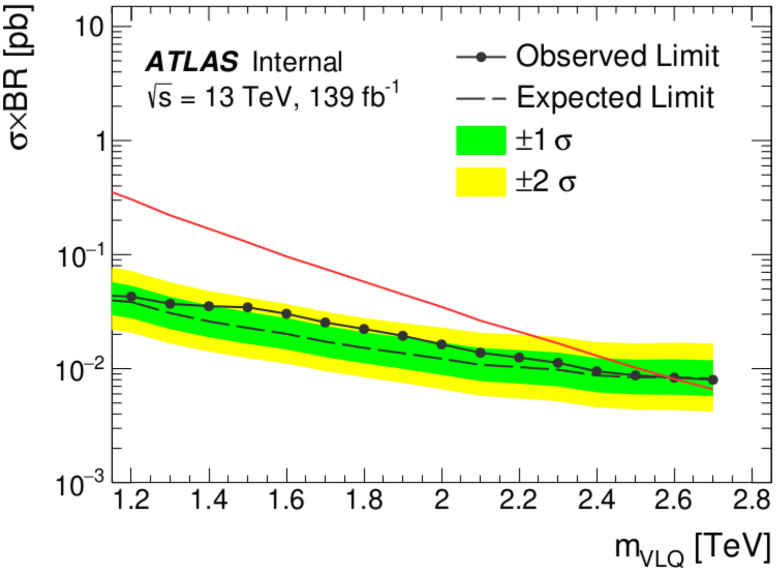


Figure 8.20: Cross-section limits on the Y-doublet as a function of mass for $\kappa = 0.5$. There is a section of under-exclusion (by more than 1σ), at approximately 1.5 TeV – 2.0 TeV. The predicted ($\mu = 1$) cross-section is shown in red.

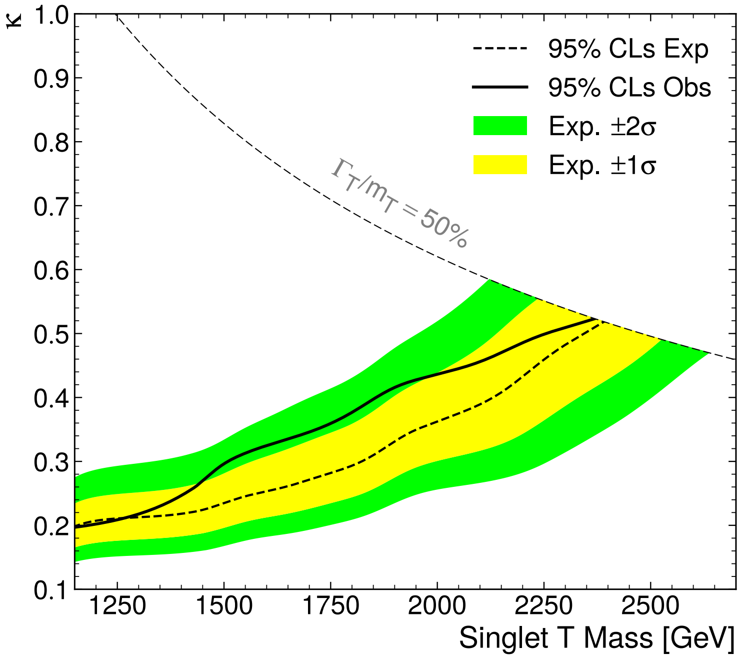


Figure 8.21: The 2D exclusion contour for the singlet-T model including interference effects. The less-than-expected exclusion in the 1500 GeV – 2000 GeV range corresponds to the similar effects in the other exclusion and discovery plots.

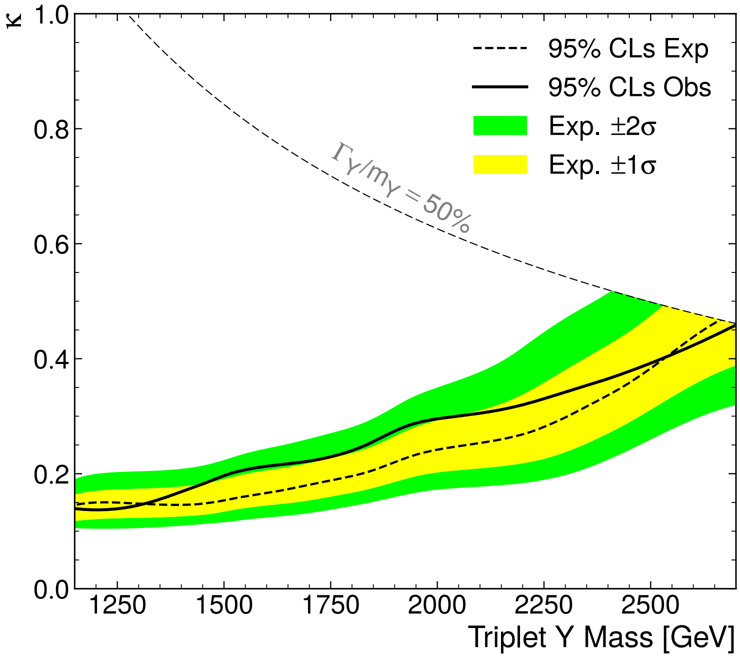


Figure 8.22: The 2D exclusion contour for the triplet-Y model including interference effects. The less-than-expected exclusion in the 1500 GeV – 2000 GeV range corresponds to the similar effects in the other exclusion and discovery plots.

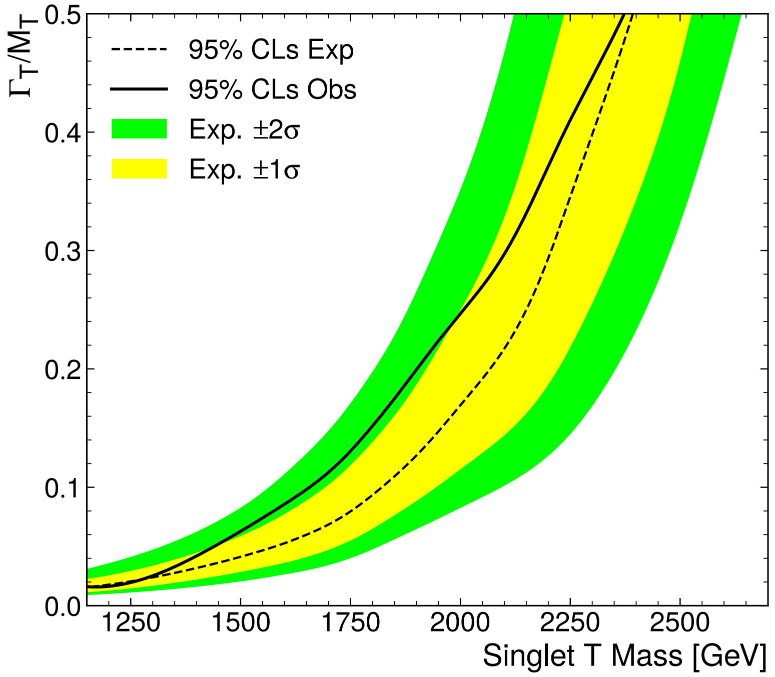


Figure 8.23: The 2D exclusion contour for the singlet-T model including interference effects in the total width over VLQ mass to mass plane. As for the κ-m_{VLQ} plot (Figure 8.21), the less-than-expected exclusion in the 1500 GeV − 2000 GeV range corresponds to the similar effects in the other exclusion and discovery plots.

relevant format.

In terms of mass limits at $\kappa = 0.5$ – a common comparison point with other analyses – the lower mass-limit on a singlet-T is 2.25 TeV. Even more impressively, both Y-quark models are completely excluded for the entire mass range satisfying $\Gamma/M < 50\%$. To better this limit, future studies will either need to use Monte-Carlo samples that do not make the narrow-width approximation; or else aim to only push the exclusion contour to progressively lower values of κ. At $\kappa = 0.3$ – perhaps a good choice for a new benchmark point – the lower limit is 2 TeV for the triplet-Y, and 1.5 TeV for the singlet-T.

The 2D-limit for the doublet-Y (i.e. no interference) is shown in Figure 8.24, alongside a comparison to the exclusion contour obtained by the recently published ATLAS all-hadronic channel search [298], which targets the same final state but with a hadronically decaying W-boson. As suggested by previous results, the one-lepton channel is significantly more sensitive, particularly for higher VLQ-masses: the mass limit obtained for the $\kappa = 0.5$ Y-doublet is 2 TeV, whereas this analysis excludes the entire 0 − 2.5 TeV range allowed by the narrow width approximation. The all-hadronic channel does provide a slightly stronger observed exclusion in the narrow 1.85 TeV − 1.95 TeV mass range where, as discussed previously, the one-lepton analysis does exclude less powerfully than expected.

Given that they are orthogonal, it is slightly intriguing that both the one-lepton and fully-hadronic analyses see a small (1 − 1.5 σ) excess – visible in the exclusion contour as an under-exclusion – in a similar same 1.5 TeV − 1.8 TeV region, albeit with the fully-hadronic analysis observing this for a slightly narrower mass range. A Run 2 combination of these channels is unlikely to be worthwhile given how much stronger the 1-lepton limit is across most of the parameter space; however, it does provide additional motivation to study this final state again during Run 3.

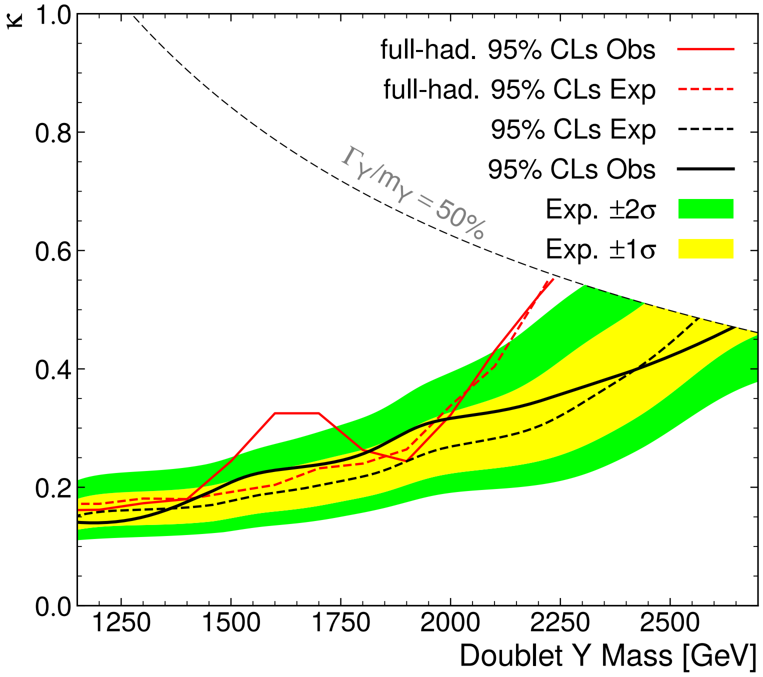


Figure 8.24: The 2D exclusion contour for the doublet-Y model including a comparison to the ATLAS zero-lepton (fully-hadronic) search for the same BSM process. The zero-lepton channel is more sensitive in almost the entire mass range, especially at higher VLQ masses. The fact that both under-exclude in a similar mass-range is intriguing.

The most recent CMS VLQ search in a comparable phase space dates to 2016, and uses just 2.3 fb^{-1} of LHC Run 2 data [299] – making the search presented in this chapter the first full Run 2 search for T or Y VLQ decaying to Wb in the one-lepton channel; and so unsurprisingly the mass limits in this study significantly exceed those obtained by CMS – at $\kappa = 0.5$, the limit on m_Y was just 1.4 TeV. The signal model in the CMS study is subtly different (the study does not include interference effects), although the phase space requirements – large missing-transverse momentum, one lepton, b- and forward-jets – were almost identical. It is probably unnecessary to pour additional water on any notion that the 1.4σ excess observed in the 1500 GeV – 2000 GeV region may represent a serious hint of new physics, but it is still worth noting that the CMS analysis in fact saw an almost 2σ *over-exclusion* in this region.

As mentioned in Section 1.3, ATLAS recently published a statistical combination of three searches for single-production of vector-like T quarks, considering both the singlet-T and doublet-T scenarios [63]. While this search is not sensitive to the doublet scenario[17], it does provide a very interesting comparison to the combination's results for singlet-T. As shown in Figure 8.25, the mass-κ limit placed by this search outperforms all the individual searches in the combination, which while pleasing is somewhat unsurprising, given that the branching ratio to the Wb final state targetted in this search is twice that of the Ht or Zt final states targetted by the searches in the combination.

Unsurprisingly, this present search provides less exclusion than the combination in the majority of the mass range; however, it is interesting to note that at higher masses (approximately 2.05 TeV – 2.3 TeV) it extends the exclusion from the combination. This allows us to extend the combinations's limit on singlet-T

[17] This is because the branching ratio of doublet-T to a Wb final state is zero.

mass at $\kappa = 0.5$ – the previous strongest – from 2.1 TeV to 2.25 TeV. Nevertheless, for κ values below 0.45, the combination still provides the best exclusion limit.

Intriguingly, of the three searches included in the combination, we only expect significant overlap with one: as the mono-top search [300] is in the zero-lepton channel; and the opposite-sign multi-lepton (for $T \to Zt$ where the Z decays leptonically) [274] is in the two-lepton channel. The one search in the combination which is likely to at least partially overlap with this one is the search for $T \to (H/Z)t$ in the one-lepton channel [51]. This suggests that the 95% exclusion contour could be extended even further by adding this study to the statistical combination.

Six years after the end of Run 2, it is unlikely to be a good use of resources to carry out a fresh Run 2 combination of VLQ searches simply to add in the results of this study. However, looking ahead to the Run 3 VLQ search program, hopefully the channel from this analysis has proved its importance, both for a specific analysis and for combination with other VLQ searches.

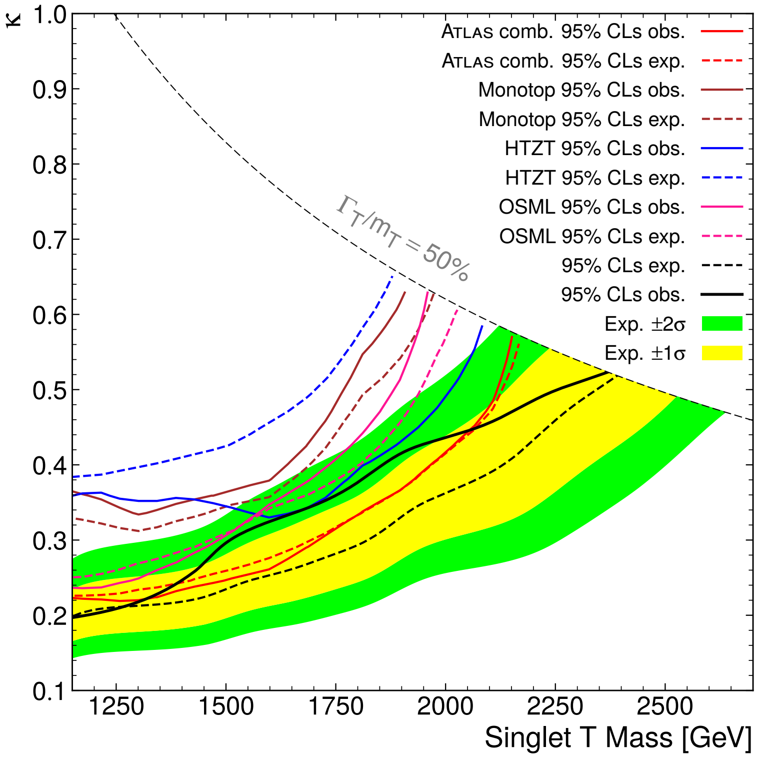


Figure 8.25: The 2D exclusion contour for the singlet-T model including interference effects, as shown in Figure 8.21. It is overlayed with the expected and observed exclusions from the ATLAS combination [63], as well as the individual exclusion contours from the three analyses included in the combination: the mono-top [300]; the opposite-sign multi-lepton (OSML) [274]; and the $T \to (H/Z)t$ (HTZT) [51] – which is the only analysis included in the combination which uses the one-lepton channel.
The analysis described in this chapter provides a stronger exclusion than any of the individual analyses in the combination. Although the ATLAS single-T combination [63] still provides the stronger exclusion limit in the majority of the mass range, this analysis provides better exclusion for masses above 2.05 TeV.

Part III

Conclusions and bibliography

Chapter 9

Conclusions

Despite its many successes, the Standard Model of particle physics is unable to account for several significant observations of the universe, such as dark matter or the “suspiciously” low mass of the Higgs-boson (i.e. the hierarchy problem). Unfortunately, there are a very wide variety of different physics models that could fix (some) of the shortcomings of the SM, and these produce a very wide array of different collider signatures. Despite the best effort of kilo-member collaborations, there is insufficient person-power (and for some models, computing-power) to carry out a dedicated search for every model that has ever graced a phenomenologist’s dreams. In this context, ensuring that analyses can be reused and reinterpreted in the context of different BSM models (without making them even more resource- or time-intensive to carry out in the first place) is essential to making full use of the data that the LHC produces.

Chapters 5 and 7 were both focussed on tools for analysis preservation and phenomenology. Chapter 5 focussed on the preservation of ML-dependent LHC searches: demonstrating the succesful preservation of one such analysis in both Rivet and ColliderBit; highlighting (with real examples) issues that may impact the preservation of ML-dependent searches going forward; and hopefully providing useful advice (and possibly even inspiration) to the wider community. Going forward, we eagerly anticipate the first publication of NN or BDT weights from CMS; and hope for more to come from Atlas. The integration of inference machinery within Rivet and ColliderBit should ensure that this procedure is not overly taxing on experimental teams. Chapter 7 showed the integration of Rivet and Contur into ColliderBit, arming Gambit with another tool with which to study new physics. Section 7.3 showed that this has already played a useful role in Gambit’s GMSB gravitino study; and the development of thread-safe Rivet described in Section 7.5 will allow Rivet and Contur to potentially play an even greater role in future studies, as Contur likelihoods can now be used to help steer ScannerBit.

Chapter 6 developed the use of CARL for reweighting samples in BSM signal grids. This can potentially greatly improve the efficiency of MC sample production for BSM signal models, an issue likely to only increase in importance to experiments as BSM models become more complex and the strain on computational resources increases. The work in this chapter answered questions about how to ensure the target distribution is supported across the parameter space, and how to choose which points to reweight from. Tests on a Rivet example (Figures 6.13 and 6.14) showed the exclusion contour is very well reproduced; the next test would ideally be in the context of a “real” Atlas signal sample being produced for a “real” Atlas analysis. The method may also have applications in phenomenology: tests on the outputs of small Gambit or Contur scans would also be of interest.

Vector-like quarks are a popular BSM scenario. The analysis in presented in Chapter 8 is the first full Run 2 search for vector-like T or Y quarks decaying to Wb in the one-lepton final state. It extended the existing limit on singlet VLTs at $\kappa = 0.5$ from 2.1 TeV to 2.25 TeV, and also placed new mass limits on the triplet and doublet Y VLQ. Had the result been ready in time for the ATLAS single-VLT combination, it would have increased the sensitivity of that study.

However, given the focus of this thesis on reinterpretation, as we draw together our conclusions it would be remiss of us not to note that while the analysis concluded after the publication of the ATLAS Run 2 combination, a "private" combination using reinterpretation tools is still eminently feasible. To do this, the RIVET implementation introduced in Section 8.6.3 would likely need to be refined, to fix the previously discussed shift in the m_{VLQ} distribution in Figure 8.12. Implementations would also be need to be written for the three analyses in the ATLAS single combination, and the $T/Y \to Wb$ fully-hadronic channel analysis (noting this also observed a small excess in a similar region): writing these for COLLIDERBIT would harmonise well with the ongoing GAMBIT VLQ project mentioned in Section 7.4. The "monotop" analysis [300] from the combination relies on a BDT (albeit one that has not yet been made public), so the experiences from Chapter 5 are likely to prove useful.

It is also important to note that unless full-likelihoods for all the analyses are published, it is unlikely a meaningful combination could be achieved, as we would be limited to using just one signal region from each analysis, leading to significantly reduced sensitivity. Full-likelihood information would also allow us to consider the impact of common systematics, which would give us greater confidence in the validity of our statistical procedure.

If we wanted to tie things together in a particularly elegant bow, we could even design a hypothetical follow-up study that makes use of all four research chapters in the thesis: We would generate a large VLQ signal grid using the CARL machinery developed in Chapter 6 (instead of the matrix-element reweighting method); the analysis from Chapter 8 would be reused, alongside the analyses from the single-combination (where as mentioned, the BDT-dependent monotop search will benefit from the experience in Chapter 5); and we could include Chapter 7 by carrying out the study in GAMBIT while also considering the impact of SM measurements via CONTUR. While this example is perhaps a little contrived, it should illustrate the fact that all the research in Part II either already has, or will likely go on to play, a role in the ongoing hunt for new physics: in direct searches at the LHC, as well as in their preservation and reuse.

Looking forward to Run 3, the analysis in Chapter 8 has shown the importance of the one-lepton channel targetting the Wb decay for single vector-like quark searches. A Run 3 analysis studying this channel again with more data (and at slightly higher centre-of-mass energies) would not only likely be even more sensitive on its own, but would also hopefully be a cornerstone of any Run 3 (or even Run 2 + Run 3) combination of searches for singly produced VLQs. Due to the increased integrated luminosity of Run 3, the "shelf-life" of Run 3 searches will be longer[1], and so – combined with the ever-increasing reliance of searches on ML-methods and the ever-growing variety and complexity of BSM models – the reinterpretation topics of this thesis are only going to be of more importance going forward.

[1] As future runs will need even more time to significantly exceed the integrated luminosity of Run 3, the period of time before new results have sufficient sensitivity to render Run 3 searches uncompetitive will be even longer.

Chapter 10

Bibliography

[1] G. Aad *et al.*, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Phys. Lett. B*, vol. 716, pp. 1–29, 2012.

[2] S. Chatrchyan *et al.*, "Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC," *Phys. Lett. B*, vol. 716, pp. 30–61, 2012.

[3] M. Thomson, *Modern Particle Physics.* Cambridge University Press, 2013.

[4] W. Commons, "File:standard model of elementary particles.svg — wikimedia commons, the free media repository," 2024. [Online; accessed 5-August-2024].

[5] A. Tumasyan *et al.*, "A portrait of the Higgs boson by the CMS experiment ten years after the discovery.," *Nature*, vol. 607, no. 7917, pp. 60–68, 2022. [Erratum: Nature 623, (2023)].

[6] C. Burgess and G. Moore, *The Standard Model: A Primer.* Cambridge University Press, 2007.

[7] W. Cottingham and D. Greenwood, *An Introduction to the Standard Model of Particle Physics.* Cambridge University Press, 1998.

[8] P. W. Higgs, "Broken Symmetries and the Masses of Gauge Bosons," *Phys. Rev. Lett.*, vol. 13, pp. 508–509, 1964.

[9] F. Englert and R. Brout, "Broken Symmetry and the Mass of Gauge Vector Mesons," *Phys. Rev. Lett.*, vol. 13, pp. 321–323, 1964.

[10] G. S. Guralnik, C. R. Hagen, and T. W. B. Kibble, "Global Conservation Laws and Massless Particles," *Phys. Rev. Lett.*, vol. 13, pp. 585–587, 1964.

[11] C. S. Wu, E. Ambler, R. W. Hayward, D. D. Hoppes, and R. P. Hudson, "Experimental Test of Parity Conservation in β Decay," *Phys. Rev.*, vol. 105, pp. 1413–1414, 1957.

[12] A. Rajantie, "Unification: Lecture notes," 2019. Accessed December 2019.

[13] L. Lederman and D. Teresi, *The God Particle: If the universe is the answer, what is the question?* Dell Publishing, 1993.

[14] N. Cabibbo, "Unitary Symmetry and Leptonic Decays," *Phys. Rev. Lett.*, vol. 10, pp. 531–533, 1963.

[15] M. Kobayashi and T. Maskawa, "CP Violation in the Renormalizable Theory of Weak Interaction," *Prog. Theor. Phys.*, vol. 49, pp. 652–657, 1973.

[16] Y. Fukuda *et al.*, "Evidence for oscillation of atmospheric neutrinos," *Phys. Rev. Lett.*, vol. 81, pp. 1562–1567, 1998.

[17] Q. R. Ahmad *et al.*, "Direct evidence for neutrino flavor transformation from neutral current interactions in the Sudbury Neutrino Observatory," *Phys. Rev. Lett.*, vol. 89, p. 011301, 2002.

[18] M. Aker *et al.*, "Direct neutrino-mass measurement based on 259 days of KATRIN data," 6 2024.

[19] D. Hanneke, S. F. Hoogerheide, and G. Gabrielse, "Cavity Control of a Single-Electron Quantum Cyclotron: Measuring the Electron Magnetic Moment," *Phys. Rev. A*, vol. 83, p. 052122, 2011.

[20] M. Milgrom, "A Modification of the Newtonian dynamics as a possible alternative to the hidden mass hypothesis," *Astrophys. J.*, vol. 270, pp. 365–370, 1983.

[21] B. J. Mount *et al.*, "LUX-ZEPLIN (LZ) Technical Design Report," 3 2017.

[22] E. Aprile *et al.*, "The XENON dark matter search experiment," *New Astron. Rev.*, vol. 49, pp. 289–295, 2005.

[23] S. Hossenfelder, "Screams for explanation: finetuning and naturalness in the foundations of physics," *Synthese*, vol. 198, no. Suppl 16, pp. 3727–3745, 2021.

[24] S. Fichet, "Quantified naturalness from Bayesian statistics," *Phys. Rev. D*, vol. 86, p. 125029, 2012.

[25] I. Aitchison, *Supersymmtery in Particle Physics: An Elementary Introduction.* Cambridge University Press, 2007.

[26] M. Peskin and D. Shroeder, *An introduction to Quantum Field Theory.* Westview Press, 1995.

[27] J. Grange *et al.*, "Muon (g-2) Technical Design Report," 1 2015. arXiv:1501.06858.

[28] G. Bunce, "The Brookhaven muon g-2 experiment," *AIP Conf. Proc.*, vol. 343, pp. 328–336, 2008.

[29] T. Aoyama *et al.*, "The anomalous magnetic moment of the muon in the Standard Model," *Phys. Rept.*, vol. 887, pp. 1–166, 2020.

[30] P. Girotti, "The precision measurement of the muon $g-2$ at Fermilab," *Nuovo Cim. C*, vol. 47, no. 3, p. 87, 2024.

[31] S. Borsanyi *et al.*, "Leading hadronic contribution to the muon magnetic moment from lattice QCD," *Nature*, vol. 593, no. 7857, pp. 51–55, 2021.

[32] R. Aaij *et al.*, "Test of lepton universality with $B^0 \to K^{*0}\ell^+\ell^-$ decays," *JHEP*, vol. 08, p. 055, 2017.

[33] R. Aaij *et al.*, "Test of lepton universality in beauty-quark decays," *Nature Phys.*, vol. 18, no. 3, pp. 277–282, 2022. [Addendum: Nature Phys. 19, (2023)].

[34] J. Aebischer, W. Altmannshofer, D. Guadagnoli, M. Reboud, P. Stangl, and D. M. Straub, "B-decay discrepancies after Moriond 2019," *Eur. Phys. J. C*, vol. 80, no. 3, p. 252, 2020.

[35] R. Aaij *et al.*, "Measurement of lepton universality parameters in $B^+ \to K^+\ell^+\ell^-$ and $B^0 \to K^{*0}\ell^+\ell^-$ decays," *Phys. Rev. D*, vol. 108, no. 3, p. 032002, 2023.

[36] S. Iguro, T. Kitahara, and R. Watanabe, "Global fit to $b \to c\tau\nu$ anomalies 2022 mid-autumn," 10 2022. arXiv:2210.10751.

[37] L. Randall and R. Sundrum, “Large mass hierarchy from a small extra dimension,” *Physical review letters*, vol. 83, no. 17, p. 3370, 1999.

[38] K. Agashe, R. Contino, and A. Pomarol, “The minimal composite higgs model,” *Nuclear Physics B*, vol. 719, no. 1-2, pp. 165–187, 2005.

[39] J. L. Hewett and T. G. Rizzo, “Low-energy phenomenology of superstring-inspired e6 models,” *Physics Reports*, vol. 183, no. 5-6, pp. 193–381, 1989.

[40] J. A. Aguilar-Saavedra, R. Benbrik, S. Heinemeyer, and M. Pérez-Victoria, “Handbook of vectorlike quarks: Mixing and single production,” *Phys. Rev. D*, vol. 88, no. 9, p. 094010, 2013.

[41] B. Fuks and H.-S. Shao, “QCD next-to-leading-order predictions matched to parton showers for vector-like quark models,” *Eur. Phys. J. C*, vol. 77, no. 2, p. 135, 2017.

[42] J. A. Aguilar-Saavedra, “Mixing with vector-like quarks: constraints and expectations,” in *EPJ Web of Conferences*, vol. 60, p. 16012, EDP Sciences, 2013.

[43] A. Hayrapetyan *et al.*, “Review of searches for vector-like quarks, vector-like leptons, and heavy neutral leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV at the CMS experiment,” 5 2024. arXiv:2405.17605.

[44] M. Buchkremer, G. Cacciapaglia, A. Deandrea, and L. Panizzi, “Model Independent Framework for Searches of Top Partners,” *Nucl. Phys. B*, vol. 876, pp. 376–417, 2013.

[45] O. Eberhardt, G. Herbert, H. Lacker, A. Lenz, A. Menzel, U. Nierste, and M. Wiebusch, “Impact of a Higgs boson at a mass of 126 GeV on the standard model with three and four fermion generations,” *Phys. Rev. Lett.*, vol. 109, p. 241802, 2012.

[46] B. Fuks and H.-S. Shao, “Qcd next-to-leading-order predictions matched to parton showers for vector-like quark models,” *The European Physical Journal C*, vol. 77, no. 135, pp. 1–21, 2017.

[47] T. S. R. Procter, “TPThesis-Snippets.” `https://github.com/tprocter46/TPThesis-Snippets`, 2024.

[48] N. D. Christensen and C. Duhr, “FeynRules - Feynman rules made easy,” *Comput. Phys. Commun.*, vol. 180, pp. 1614–1641, 2009.

[49] L. Darmé *et al.*, “UFO 2.0: the ‘Universal Feynman Output’ format,” *Eur. Phys. J. C*, vol. 83, no. 7, p. 631, 2023.

[50] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations,” *JHEP*, vol. 07, p. 079, 2014.

[51] G. Aad *et al.*, “Search for single production of vector-like T quarks decaying into Ht or Zt in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *JHEP*, vol. 08, p. 153, 2023.

[52] M. Bahr *et al.*, “Herwig++ Physics and Manual,” *Eur. Phys. J. C*, vol. 58, pp. 639–707, 2008.

[53] J. Bellm *et al.*, “Herwig 7.0/Herwig++ 3.0 release note,” *Eur. Phys. J. C*, vol. 76, no. 4, p. 196, 2016.

[54] A. Buckley, J. M. Butterworth, L. Corpe, D. Huang, and P. Sun, “New sensitivity of current LHC measurements to vector-like quarks,” *SciPost Phys.*, vol. 9, no. 5, p. 069, 2020.

[55] S. Bloor, T. E. Gonzalo, P. Scott, C. Chang, A. Raklev, J. E. Camargo-Molina, A. Kvellestad, J. J. Renk, P. Athron, and C. Balázs, "The GAMBIT Universal Model Machine: from Lagrangians to likelihoods," *Eur. Phys. J. C*, vol. 81, no. 12, p. 1103, 2021.

[56] J. A. Aguilar-Saavedra, "PROTOS, a PROgram for TOp Simulations," tech. rep. `http://jaguilar.web.cern.ch/jaguilar/protos/`.

[57] G. Aad *et al.*, "Search for production of vector-like quark pairs and of four top quarks in the lepton-plus-jets final state in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector," *JHEP*, vol. 08, p. 105, 2015.

[58] M. Aaboud *et al.*, "Search for pair production of up-type vector-like quarks and for four-top-quark events in final states with multiple b-jets with the ATLAS detector," *JHEP*, vol. 07, p. 089, 2018.

[59] The ATLAS Collaboration, "Search for single production of vector-like quarks coupling to light generations in 4.64 fb^{-1} of data at $\sqrt{s} = 7$ TeV," 9 2012. ATLAS-CONF-2012-137, preliminary.

[60] S. Bhattacharya, "Search for exotic top partners at $\sqrt{s} = 8$ TeV," in *Meeting of the APS Division of Particles and Fields*, 10 2013.

[61] A. Succurro, "Search for pair-produced vector-like quarks with the ATLAS detector," *EPJ Web Conf.*, vol. 60, p. 20037, 2013.

[62] M. Aaboud *et al.*, "Combination of the searches for pair-produced vector-like partners of the third-generation quarks at $\sqrt{s} = 13$ TeV with the ATLAS detector," *Phys. Rev. Lett.*, vol. 121, no. 21, p. 211801, 2018.

[63] G. Aad *et al.*, "Combination of searches for singly produced vector-like top quarks in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," 8 2024.

[64] A. Banerjee *et al.*, "Phenomenological aspects of composite Higgs scenarios: exotic scalars and vector-like quarks," 3 2022. arXiv:2203.07270.

[65] A. S. Cornell, A. Deandrea, T. Flacke, B. Fuks, and L. Mason, "Top partners and scalar dark matter: A nonminimal reappraisal," *Phys. Rev. D*, vol. 107, no. 7, p. 075004, 2023.

[66] B. Allanach and H. E. Haber, "Supersymmetry, Part I (Theory)," 1 2024. arXiv:2401.03827.

[67] S. P. Martin, "A Supersymmetry primer," *Adv. Ser. Direct. High Energy Phys.*, vol. 18, pp. 1–98, 1998.

[68] M. Aaboud *et al.*, "Search for long-lived particles produced in pp collisions at $\sqrt{s} = 13$ TeV that decay into displaced hadronic jets in the ATLAS muon spectrometer," *Phys. Rev. D*, vol. 99, no. 5, p. 052005, 2019.

[69] G. Aad *et al.*, "Search for long-lived, massive particles in events with displaced vertices and multiple jets in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *JHEP*, vol. 2306, p. 200, 2023.

[70] S. D. Deser and B. Zumino, "Consistent supergravity," *Phys. Lett. B*, vol. 62, no. CERN-TH-2164, pp. 335–337, 1976.

[71] M. Dine, A. E. Nelson, and Y. Shirman, "Low-energy dynamical supersymmetry breaking simplified," *Phys. Rev. D*, vol. 51, pp. 1362–1370, 1995.

[72] N. E. Bomark, A. Kvellestad, S. Lola, P. Osland, and A. R. Raklev, "Long lived charginos in Natural SUSY?," *JHEP*, vol. 05, p. 007, 2014.

[73] J. Alwall, P. Schuster, and N. Toro, "Simplified Models for a First Characterization of New Physics at the LHC," *Phys. Rev. D*, vol. 79, p. 075020, 2009.

[74] B. C. Allanach *et al.*, "SUSY Les Houches Accord 2," *Comput. Phys. Commun.*, vol. 180, pp. 8–25, 2009.

[75] G. Aad *et al.*, "The quest to discover supersymmetry at the ATLAS experiment," 3 2024. arXiv:2403.02455.

[76] K. J. de Vries, "SUSY fits with full LHC Run I data," *Nucl. Part. Phys. Proc.*, vol. 273-275, pp. 528–534, 2016.

[77] M. Donadoni, M. Feickert, L. Heinrich, Y. Liu, A. Mečionis, V. Moisieienkov, T. Šimko, G. Stark, and M. V. García, "Scalable ATLAS pMSSM computational workflows using containerised REANA reusable analysis platform," *EPJ Web Conf.*, vol. 295, p. 04035, 2024.

[78] S. Chatrchyan *et al.*, "The CMS Experiment at the CERN LHC," *JINST*, vol. 3, p. S08004, 2008.

[79] A. A. Alves, Jr. *et al.*, "The LHCb Detector at the LHC," *JINST*, vol. 3, p. S08005, 2008.

[80] K. Aamodt *et al.*, "The ALICE experiment at the CERN LHC," *JINST*, vol. 3, p. S08002, 2008.

[81] G. Aad *et al.*, "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST*, vol. 3, p. S08003, 2008.

[82] M. Hori and J. Walz, "Physics at CERN's Antiproton Decelerator," *Prog. Part. Nucl. Phys.*, vol. 72, pp. 206–253, 2013.

[83] "LHC Machine," *JINST*, vol. 3, p. S08001, 2008.

[84] S. Dubourg, M. Schaumann, and D. Walsh, eds., *Proceedings of the 2019 Evian Workshop on LHC Beam Operations*, (Geneva, Switzerland), 2019.

[85] A. Buckley, C. White, and M. White, *Practical Collider Physics.* IOP, 12 2021.

[86] G. Aad *et al.*, "Luminosity determination in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC," *Eur. Phys. J. C*, vol. 83, no. 10, p. 982, 2023.

[87] G. Avoni *et al.*, "The new LUCID-2 detector for luminosity measurement and monitoring in ATLAS," *JINST*, vol. 13, no. 07, p. P07017, 2018.

[88] E. Mobs, "The CERN accelerator complex - August 2018. Complexe des accélérateurs du CERN - Août 2018," 2018. General Photo.

[89] G. Aad *et al.*, "ATLAS pixel detector electronics and sensors," *JINST*, vol. 3, p. P07007, 2008.

[90] A. Duperrin, "Flavour tagging with graph neural networks with the ATLAS detector," in *30th International Workshop on Deep-Inelastic Scattering and Related Subjects*, 6 2023.

[91] A. A. et al, "The silicon microstrip sensors of the atlas semiconductor tracker," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 578, no. 1, pp. 98–118, 2007.

[92] B. Mindur, “ATLAS Transition Radiation Tracker (TRT): Straw tubes for tracking and particle identification at the Large Hadron Collider,” *Nucl. Instrum. Meth. A*, vol. 845, pp. 257–261, 2017.

[93] M. Aaboud *et al.*, “Study of the material of the ATLAS inner detector for Run 2 of the LHC,” *JINST*, vol. 12, no. 12, p. P12009, 2017.

[94] H.-Q. Zhang, “The ATLAS Liquid Argon calorimeter: Overview and performance,” *J. Phys. Conf. Ser.*, vol. 293, p. 012044, 2011.

[95] G. Aad *et al.*, “Operation and performance of the ATLAS tile calorimeter in LHC Run 2,” 1 2024.

[96] R. Abreu, “The upgrade of the ATLAS High Level Trigger and data acquisition systems and their integration,” in *19th Real Time Conference*, p. 7097414, 2014.

[97] G. Aad *et al.*, “Performance of the ATLAS Level-1 topological trigger in Run 2,” *Eur. Phys. J. C*, vol. 82, no. 1, p. 7, 2022.

[98] G. Aad *et al.*, “Operation of the ATLAS trigger system in Run 2,” *JINST*, vol. 15, no. 10, p. P10004, 2020.

[99] ATLAS Collaboration, “Athena,” 4 2019. `https://doi.org/10.5281/zenodo.2641997`.

[100] G. Aad *et al.*, “Performance of the ATLAS muon triggers in Run 2,” *JINST*, vol. 15, no. 09, p. P09015, 2020.

[101] “LHC computing Grid. Technical design report,” 6 2005.

[102] I. Bird *et al.*, “Update of the Computing Models of the WLCG and the LHC Experiments,” 4 2014.

[103] G. Aad *et al.*, “Software and computing for Run 3 of the ATLAS experiment at the LHC,” 4 2024.

[104] F. Barreiro Magino, D. Cameron, A. Di Girolamo, A. Filipcic, I. Glushkov, F. Legger, T. Maeno, and R. Walker, “Operation of the ATLAS Distributed Computing,” *EPJ Web Conf.*, vol. 214, p. 03049, 2019.

[105] R. Fruhwirth, “Application of Kalman filtering to track and vertex fitting,” *Nucl. Instrum. Meth. A*, vol. 262, pp. 444–450, 1987.

[106] M. Aaboud *et al.*, “Performance of the ATLAS Track Reconstruction Algorithms in Dense Environments in LHC Run 2,” *Eur. Phys. J. C*, vol. 77, no. 10, p. 673, 2017.

[107] G. Aad *et al.*, “Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1,” *Eur. Phys. J. C*, vol. 77, p. 490, 2017.

[108] B. Andrieu, “Jet finding algorithms at Tevatron,” *Acta Phys. Polon. B*, vol. 36, pp. 409–415, 2005.

[109] G. P. Salam and G. Soyez, “A Practical Seedless Infrared-Safe Cone jet algorithm,” *JHEP*, vol. 05, p. 086, 2007.

[110] G. P. Salam, “Towards Jetography,” *Eur. Phys. J. C*, vol. 67, pp. 637–686, 2010.

[111] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet User Manual,” *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[112] M. Cacciari and G. P. Salam, “Dispelling the N^3 myth for the k_t jet-finder,” *Phys. Lett. B*, vol. 641, pp. 57–61, 2006.

[113] M. Aaboud *et al.*, “Performance of top-quark and W-boson tagging with ATLAS in Run 2 of the LHC,” *Eur. Phys. J. C*, vol. 79, no. 5, p. 375, 2019.

[114] G. Aad *et al.*, “Jet energy scale and resolution measured in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Eur. Phys. J. C*, vol. 81, no. 8, p. 689, 2021.

[115] M. Aaboud *et al.*, “Jet reconstruction and performance using particle flow with the ATLAS Detector,” *Eur. Phys. J. C*, vol. 77, no. 7, p. 466, 2017.

[116] B. Nachman, P. Nef, A. Schwartzman, M. Swiatlowski, and C. Wanotayaroj, “Jets from Jets: Reclustering as a tool for large radius jet reconstruction and grooming at the LHC,” *JHEP*, vol. 02, p. 075, 2015.

[117] G. Aad *et al.*, “Tagging and suppression of pileup jets,” 5 2014. ATLAS-CONF-2014-018.

[118] G. Aad *et al.*, “Constituent-Based Quark Gluon Tagging using Transformers with the ATLAS detector,” 2023.

[119] G. Aad *et al.*, “Optimisation of the ATLAS b-tagging performance for the 2016 LHC Run,” 2016.

[120] G. Aad *et al.*, “ATLAS flavour-tagging algorithms for the LHC Run 2 pp collision dataset,” *Eur. Phys. J. C*, vol. 83, no. 7, p. 681, 2023.

[121] “Secondary vertex finding for jet flavour identification with the ATLAS detector,” 6 2017.

[122] “Graph Neural Network Jet Flavour Tagging with the ATLAS Detector,” 2022.

[123] J. Thaler and K. Van Tilburg, “Identifying Boosted Objects with N-subjettiness,” *JHEP*, vol. 03, p. 015, 2011.

[124] G. Aad *et al.*, “Electron and photon efficiencies in LHC Run 2 with the ATLAS experiment,” *JHEP*, vol. 05, p. 162, 2024.

[125] G. Aad *et al.*, “Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton-proton collision data,” *JINST*, vol. 14, no. 12, p. P12006, 2019.

[126] G. Aad *et al.*, “Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at $\sqrt{s} = 13$ TeV,” *Eur. Phys. J. C*, vol. 81, no. 7, p. 578, 2021.

[127] G. Aad *et al.*, “Studies of the muon momentum calibration and performance of the ATLAS detector with pp collisions at $\sqrt{s} = 13$ TeV,” *Eur. Phys. J. C*, vol. 83, no. 8, p. 686, 2023.

[128] G. Aad *et al.*, “The performance of missing transverse momentum reconstruction and its significance with the ATLAS detector using 140 fb^{-1} of $\sqrt{s} = 13$ TeV pp collisions,” 2 2024.

[129] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, “OmniFold: A Method to Simultaneously Unfold All Observables,” *Phys. Rev. Lett.*, vol. 124, no. 18, p. 182001, 2020.

[130] E. Bothmann *et al.*, “Event Generation with Sherpa 2.2,” *SciPost Phys.*, vol. 7, no. 3, p. 034, 2019.

[131] C. Bierlich *et al.*, “A comprehensive guide to the physics and usage of PYTHIA 8.3,” *SciPost Phys. Codeb.*, vol. 2022, p. 8, 2022.

[132] N. Kidonakis, “NNLL resummation for s-channel single top quark production,” *Phys. Rev. D*, vol. 81, p. 054028, 2010.

[133] R. Frederix, E. Re, and P. Torrielli, “Single-top t-channel hadroproduction in the four-flavour scheme with POWHEG and aMC@NLO,” *JHEP*, vol. 09, p. 130, 2012.

[134] G. Wolf *et al.*, “THE ZEUS DETECTOR: TECHNICAL PROPOSAL,” 3 1986.

[135] G. E. Wolf, “HERA: Physics, Machine and Experiments,” *NATO Sci. Ser. B*, vol. 164, pp. 375–449, 1987.

[136] C. Zimmermann and A. Schäfer, “Valence Quark PDFs of the Proton from Two-Current Correlations in Lattice QCD,” 5 2024.

[137] J. McGowan, T. Cridge, L. A. Harland-Lang, and R. S. Thorne, “Approximate N^{3}LO parton distribution functions with theoretical uncertainties: MSHT20aN^{3}LO PDFs,” *Eur. Phys. J. C*, vol. 83, no. 3, p. 185, 2023. [Erratum: Eur.Phys.J.C 83, 302 (2023)].

[138] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, “LHAPDF6: parton density access in the LHC precision era,” *Eur. Phys. J. C*, vol. 75, p. 132, 2015.

[139] J. Alwall *et al.*, “A Standard format for Les Houches event files,” *Comput. Phys. Commun.*, vol. 176, pp. 300–304, 2007.

[140] S. Hoeche, F. Krauss, N. Lavesson, L. Lonnblad, M. Mangano, A. Schalicke, and S. Schumann, “Matching parton showers and matrix elements,” in *HERA and the LHC: A Workshop on the Implications of HERA for LHC Physics: CERN - DESY Workshop 2004/2005 (Midterm Meeting, CERN, 11-13 October 2004; Final Meeting, DESY, 17-21 January 2005)*, pp. 288–289, 2005.

[141] P. Nason, “A New method for combining NLO QCD with shower Monte Carlo algorithms,” *JHEP*, vol. 11, p. 040, 2004.

[142] S. Frixione, P. Nason, and C. Oleari, “Matching NLO QCD computations with Parton Shower simulations: the POWHEG method,” *JHEP*, vol. 11, p. 070, 2007.

[143] S. Frixione and B. R. Webber, “Matching NLO QCD computations and parton shower simulations,” *JHEP*, vol. 06, p. 029, 2002.

[144] T. Sjostrand, “The Lund Monte Carlo for Jet Fragmentation,” *Comput. Phys. Commun.*, vol. 27, p. 243, 1982.

[145] D. Amati and G. Veneziano, “Preconfinement as a Property of Perturbative QCD,” *Phys. Lett. B*, vol. 83, pp. 87–92, 1979.

[146] A. Buckley, H. Hoeth, H. Lacker, H. Schulz, and J. E. von Seggern, “Systematic event generator tuning for the LHC,” *Eur. Phys. J. C*, vol. 65, pp. 331–357, 2010.

[147] G. Aad *et al.*, “Search for neutral long-lived particles that decay into displaced jets in the ATLAS calorimeter in association with leptons or jets using pp collisions at $\sqrt{s} = 13$ TeV,” 7 2024.

[148] A. Ryd, D. Lange, N. Kuznetsova, S. Versille, M. Rotondo, D. P. Kirkby, F. K. Wuerthwein, and A. Ishikawa, “EvtGen: A Monte Carlo Generator for B-Physics,” 5 2005.

[149] Z. Was, “TAUOLA the library for tau lepton decay, and KKMC / KORALB / KORALZ /... status report,” *Nucl. Phys. B Proc. Suppl.*, vol. 98, pp. 96–102, 2001.

[150] Z. Was, “TAUOLA for simulation of tau decay and production: perspectives for precision low energy and LHC applications,” *Nucl. Phys. B Proc. Suppl.*, vol. 218, pp. 249–255, 2011.

[151] A. Buckley *et al.*, “General-purpose event generators for LHC physics,” *Phys. Rept.*, vol. 504, pp. 145–233, 2011.

[152] J. Boudreau and V. Tsulaia, “The GeoModel toolkit for detector description,” in *14th International Conference on Computing in High-Energy and Nuclear Physics*, pp. 353–356, 2005.

[153] S. Agostinelli *et al.*, “GEANT4–a simulation toolkit,” *Nucl. Instrum. Meth. A*, vol. 506, pp. 250–303, 2003.

[154] J. Allison *et al.*, “Geant4 developments and applications,” *IEEE Trans. Nucl. Sci.*, vol. 53, p. 270, 2006.

[155] J. Allison *et al.*, “Recent developments in Geant4,” *Nucl. Instrum. Meth. A*, vol. 835, pp. 186–225, 2016.

[156] M. Beckingham, M. Duehrssen, E. Schmidt, M. Shapiro, M. Venturi, J. Virzi, I. Vivarelli, M. Werner, S. Yamamoto, and T. Yamanaka, “The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim,” 10 2010.

[157] W. Lukas, “Fast Simulation for ATLAS: Atlfast-II and ISF,” *J. Phys. Conf. Ser.*, vol. 396, p. 022031, 2012.

[158] G. Aad *et al.*, “AtlFast3: The Next Generation of Fast Simulation in ATLAS,” *Comput. Softw. Big Sci.*, vol. 6, no. 1, p. 7, 2022.

[159] M. P. Heath, “The new ATLAS Fast Calorimeter Simulation,” *PoS*, vol. EPS-HEP2017, p. 792, 2018.

[160] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi, “DELPHES 3, A modular framework for fast simulation of a generic collider experiment,” *JHEP*, vol. 02, p. 057, 2014.

[161] J. E. Gaiser, “Charmonium Spectroscopy From Radiative Decays of the J/ψ and ψ',” 8 1982.

[162] C. Balázs *et al.*, “ColliderBit: a GAMBIT module for the calculation of high-energy collider observables and likelihoods,” *Eur. Phys. J. C*, vol. 77, no. 11, p. 795, 2017.

[163] G. Aad *et al.*, “Searches for exclusive Higgs boson decays into $D^*\gamma$ and Z boson decays into $D^0\gamma$ and $K_s^0\gamma$ in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” 2 2024.

[164] A. Hayrapetyan *et al.*, “Observation of the J/$\psi \rightarrow \mu^+\mu^-\mu^+\mu^-$ decay in proton-proton collisions at s=13 TeV,” *Phys. Rev. D*, vol. 109, no. 11, p. L111101, 2024.

[165] R. Aaij *et al.*, “Search for CP violation in $\Xi_b^- \rightarrow pK^-K^-$decays,” *Phys. Rev. D*, vol. 104, no. 5, p. 052010, 2021.

[166] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, “Asymptotic formulae for likelihood-based tests of new physics,” *Eur. Phys. J. C*, vol. 71, p. 1554, 2011. [Erratum: Eur.Phys.J.C 73, 2501 (2013)].

[167] S. S. Wilks, “The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses,” *Annals Math. Statist.*, vol. 9, no. 1, pp. 60–62, 1938.

[168] A. Wald, “Tests of statistical hypotheses concerning several parameters when the number of observations is large,” *Transactions of the American Mathematical society*, vol. 54, no. 3, pp. 426–482, 1943.

[169] T. Junk, “Confidence level computation for combining searches with small statistics,” *Nucl. Instrum. Meth. A*, vol. 434, pp. 435–443, 1999.

[170] A. L. Read, “Presentation of search results: The CL_s technique,” *J. Phys. G*, vol. 28, pp. 2693–2704, 2002.

[171] S. Jin, “The signal estimator limit setting method,” *Nucl. Phys. A*, vol. 675, pp. 88C–91C, 2000.

[172] L. Heinrich, M. Feickert, G. Stark, and K. Cranmer, “pyhf: pure-Python implementation of HistFactory statistical models,” *J. Open Source Softw.*, vol. 6, no. 58, p. 2823, 2021.

[173] K. Cranmer, G. Lewis, L. Moneta, A. Shibata, and W. Verkerke, “HistFactory: A tool for creating statistical models for use with RooFit and RooStats,” 6 2012.

[174] F. James, Y. Perrin, and L. Lyons, eds., *Workshop on confidence limits, CERN, Geneva, Switzerland, 17-18 Jan 2000: Proceedings*, CERN Yellow Reports: Conference Proceedings, 5 2000.

[175] A. Buckley, M. Citron, S. Fichet, S. Kraml, W. Waltenberger, and N. Wardle, “The Simplified Likelihood Framework,” *JHEP*, vol. 04, p. 064, 2019.

[176] A. M. Sirunyan *et al.*, “Search for new physics in final states with an energetic jet or a hadronically decaying W or Z boson and transverse momentum imbalance at $\sqrt{s} = 13$ TeV,” *Phys. Rev. D*, vol. 97, no. 9, p. 092005, 2018.

[177] G. Aad *et al.*, “Search for supersymmetry in final states with missing transverse momentum and three or more b-jets in 139 fb^{-1} of proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Eur. Phys. J. C*, vol. 83, no. 7, p. 561, 2023.

[178] L. Moneta, K. Belasco, K. S. Cranmer, S. Kreiss, A. Lazzaro, D. Piparo, G. Schott, W. Verkerke, and M. Wolf, “The RooStats Project,” *PoS*, vol. ACAT2010, p. 057, 2010.

[179] C. Burgard, “HEP statistics serialization standard.” Talk at the 8th general workshop of the *Forum on the interpretation of the LHC results for BSM studies*, Ref. [301].

[180] A. Hayrapetyan *et al.*, “The CMS Statistical Analysis and Combination Tool: Combine,” *Comput. Softw. Big Sci.*, vol. 8, no. 1, p. 19, 2024.

[181] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[182] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 12 2014. arXiv:1412.6980.

[183] F. Chollet *et al.*, “Keras.” `https://keras.io`, 2015.

[184] M. A. et al, “TensorFlow: Large-scale machine learning on heterogeneous systems.” Software available from tensorflow.org.

[185] PyTorch Team, “Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation,” in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS ’24)*, 4 2024.

[186] A. Andreassen and B. Nachman, “Neural Networks for Full Phase-space Reweighting and Parameter Tuning,” *Phys. Rev. D*, vol. 101, no. 9, p. 091901, 2020.

[187] G. Alguero, J. Heisig, C. K. Khosa, S. Kraml, S. Kulkarni, A. Lessa, H. Reyes-González, W. Waltenberger, and A. Wongel, "Constraining new physics with SModelS version 2," *JHEP*, vol. 08, p. 068, 2022.

[188] H. Bahl, T. Biekötter, S. Heinemeyer, C. Li, S. Paasch, G. Weiglein, and J. Wittbrodt, "HiggsTools: BSM scalar phenomenology with new versions of HiggsBounds and HiggsSignals," *Comput. Phys. Commun.*, vol. 291, p. 108803, 2023.

[189] K. Cranmer and I. Yavin, "RECAST: Extending the Impact of Existing Analyses," *JHEP*, vol. 04, p. 038, 2011.

[190] C. Bierlich, A. Buckley, J. Butterworth, C. Gutschow, L. Lonnblad, T. Procter, P. Richardson, and Y. Yeh, "Robust Independent Validation of Experiment and Theory: Rivet version 4 release note," 4 2024.

[191] A. Buckley, L. Corpe, M. Filipovich, C. Gutschow, N. Rozinsky, S. Thor, Y. Yeh, and J. Yellen, "Consistent, multidimensional differential histogramming and summary statistics with YODA 2," 12 2023.

[192] B. M. Waugh, H. Jung, A. Buckley, L. Lonnblad, J. M. Butterworth, and E. Nurse, "HZTool and Rivet: Toolkit and Framework for the Comparison of Simulated Final States and Data at Colliders," in *15th International Conference on Computing in High Energy and Nuclear Physics*, 5 2006.

[193] J. M. Butterworth, D. Grellscheid, M. Krämer, B. Sarrazin, and D. Yallup, "Constraining new physics with collider measurements of Standard Model signatures," *JHEP*, vol. 03, p. 078, 2017.

[194] A. Buckley *et al.*, "Testing new physics models with global comparisons to collider measurements: the Contur toolkit," *SciPost Phys. Core*, vol. 4, p. 013, 2021.

[195] J. Rocamonde, L. Corpe, G. Zilgalvis, M. Avramidou, and J. Butterworth, "Picking the low-hanging fruit: testing new physics at scale with active learning," *SciPost Phys.*, vol. 13, no. 1, p. 002, 2022.

[196] M. Verma, A. M. Jayich, and A. C. Vutha, "Electron electric dipole moment searches using clock transitions in ultracold molecules," *Phys. Rev. Lett.*, vol. 125, no. 15, p. 153201, 2020.

[197] D. S. Akerib *et al.*, "First direct detection constraint on mirror dark matter kinetic mixing using LUX 2013 data," *Phys. Rev. D*, vol. 101, no. 1, p. 012003, 2020.

[198] V. A. Kudryavtsev, "Recent Results from LUX and Prospects for Dark Matter Searches with LZ," *Universe*, vol. 5, no. 3, p. 73, 2019.

[199] P. Athron *et al.*, "GAMBIT: The Global and Modular Beyond-the-Standard-Model Inference Tool," *Eur. Phys. J. C*, vol. 77, no. 11, p. 784, 2017. [Addendum: Eur.Phys.J.C 78, 98 (2018)].

[200] F. U. Bernlochner *et al.*, "FlavBit: A GAMBIT module for computing flavour observables and likelihoods," *Eur. Phys. J. C*, vol. 77, no. 11, p. 786, 2017.

[201] J. M. Cornell, "An overview of DarkBit, the GAMBIT dark matter module," *J. Phys. Conf. Ser.*, vol. 1342, no. 1, p. 1, 2020.

[202] P. Athron *et al.*, "Combined collider constraints on neutralinos and charginos," *Eur. Phys. J. C*, vol. 79, no. 5, p. 395, 2019.

[203] V. Ananyev *et al.*, “Collider constraints on electroweakinos in the presence of a light gravitino,” *Eur. Phys. J. C*, vol. 83, no. 6, p. 493, 2023.

[204] C. Balázs *et al.*, “Cosmological constraints on decaying axion-like particles: a global analysis,” *JCAP*, vol. 12, p. 027, 2022.

[205] A. Putze and L. Derome, “The Grenoble Analysis Toolkit (GreAT)—A statistical analysis framework,” *Phys. Dark Univ.*, vol. 5-6, pp. 29–34, 2014.

[206] J. A. Christen and C. Fox, “A general purpose sampling algorithm for continuous distributions (the t-walk),” *Bayesian Analysis*, vol. 5, 2010.

[207] G. D. Martinez, J. McKay, B. Farmer, P. Scott, E. Roebber, A. Putze, and J. Conrad, “Comparison of statistical sampling methods with ScannerBit, the GAMBIT scanning module,” *Eur. Phys. J. C*, vol. 77, no. 11, p. 761, 2017.

[208] F. Feroz, M. P. Hobson, and M. Bridges, “MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics,” *Mon. Not. Roy. Astron. Soc.*, vol. 398, pp. 1601–1614, 2009.

[209] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr, and B. Fuks, “FeynRules 2.0 - A complete toolbox for tree-level phenomenology,” *Comput. Phys. Commun.*, vol. 185, pp. 2250–2300, 2014.

[210] F. Staub, “SARAH 4 : A tool for (not only SUSY) model builders,” *Comput. Phys. Commun.*, vol. 185, pp. 1773–1790, 2014.

[211] F. Staub, “Exploring new models in all detail with SARAH,” *Adv. High Energy Phys.*, vol. 2015, p. 840780, 2015.

[212] S. Amoroso *et al.*, “Les Houches 2019: Physics at TeV Colliders: Standard Model Working Group Report,” in *11th Les Houches Workshop on Physics at TeV Colliders: PhysTeV Les Houches*, 3 2020.

[213] D. Dercks, N. Desai, J. S. Kim, K. Rolbiecki, J. Tattersall, and T. Weber, “CheckMATE 2: From the model to the limit,” *Comput. Phys. Commun.*, vol. 221, pp. 383–418, 2017.

[214] The ATLAS collaboration, “SimpleAnalysis: Truth-level Analysis Framework,”

[215] G. Stark, C. A. Ots, and M. Hance, “Reduce, Reuse, Reinterpret: an end-to-end pipeline for recycling particle physics results,” 6 2023.

[216] G. Aad *et al.*, “ATLAS Run 2 searches for electroweak production of supersymmetric particles interpreted within the pMSSM,” *JHEP*, vol. 05, p. 106, 2024.

[217] E. Conte, B. Fuks, and G. Serret, “MadAnalysis 5, A User-Friendly Framework for Collider Phenomenology,” *Comput. Phys. Commun.*, vol. 184, pp. 222–256, 2013.

[218] J. Y. Araz, B. Fuks, and G. Polykratis, “Simplified fast detector simulation in MADANALYSIS 5,” *Eur. Phys. J. C*, vol. 81, no. 4, p. 329, 2021.

[219] J. Lim, C.-T. Lu, J.-H. Park, and J. Park, “Implementation of the ATLAS-SUSY-2018-04 analysis in the MadAnalysis 5 framework (staus in the di-tau plus missing transverse energy channel; 139 fb^{1}),” *Mod. Phys. Lett. A*, vol. 36, no. 01, p. 2141009, 2021.

[220] T. J. Berners-Lee, R. Cailliau, J. F. Groff, and B. Pollermann, “World Wide Web: An Information infrastructure for high-energy physics,” *Conf. Proc. C*, vol. 9201131, pp. 157–164, 1992.

[221] B. H. Denby and S. L. Linn, “Status of HEP Neural Net Research in the USA,” *Comput. Phys. Commun.*, vol. 57, pp. 297–300, 1989.

[222] R. Donaldson, “Proceedings of the workshop on triggering and data acquisition for experiments at the supercollider,” tech. rep., Lawrence Berkeley Lab., CA (United States). SSC Central Design Group, 1989.

[223] M. Aaboud *et al.*, “Search for supersymmetry in final states with missing transverse momentum and multiple b-jets in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *JHEP*, vol. 06, p. 107, 2018.

[224] M. Aaboud *et al.*, “Search for pair production of heavy vector-like quarks decaying into hadronic final states in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Phys. Rev. D*, vol. 98, no. 9, p. 092005, 2018.

[225] G. Aad *et al.*, “Search for neutral long-lived particles in pp collisions at $\sqrt{s} = 13$ TeV that decay into displaced hadronic jets in the ATLAS calorimeter,” *JHEP*, vol. 06, p. 005, 2022.

[226] M. Feindt and U. Kerzel, “The NeuroBayes neural network package,” *Nucl. Instrum. Meth. A*, vol. 559, pp. 190–194, 2006.

[227] G. Aad *et al.*, “Search for pair-produced vector-like top and bottom partners in events with large missing transverse momentum in pp collisions with the ATLAS detector,” *Eur. Phys. J. C*, vol. 83, no. 8, p. 719, 2023.

[228] N. Kovelamudi and F. Chollet, “Save, serialize, and export models.” `https://www.tensorflow.org/guide/keras/serialization_and_saving` (Accessed 01-09-2024), 2023.

[229] ONNX Runtime developers, “ONNX Runtime.” `https://onnxruntime.ai/`, 2021.

[230] G. Malivenko *et al.*, “onnx2keras.” `https://github.com/gmalivenko/onnx2keras`, 2021.

[231] T. Procter, “Pull request: Keras add layer.” `https://github.com/lwtnn/lwtnn/pull/171`, 2022.

[232] G. Aad *et al.*, “Search for squarks and gluinos in final states with jets and missing transverse momentum using 139 fb^{-1} of $\sqrt{s}$ =13 TeV pp collision data with the ATLAS detector,” *JHEP*, vol. 02, p. 143, 2021.

[233] G. Aad *et al.*, “Search for R-parity-violating supersymmetry in a final state containing leptons and many jets with the ATLAS experiment using $\sqrt{s} = 13TeV$ proton–proton collision data,” *Eur. Phys. J. C*, vol. 81, no. 11, p. 1023, 2021.

[234] “7th LHC BSM Reinterpretation Forum Workshop.” `https://indico.cern.ch/event/1197680/`, 1215 Dec 2022, CERN. Accessed: 2023-12-13.

[235] M. Aaboud *et al.*, “Measurement of inclusive jet and dijet cross-sections in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *JHEP*, vol. 05, p. 195, 2018.

[236] G. Aad *et al.*, “Performance of pile-up mitigation techniques for jets in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector,” *Eur. Phys. J. C*, vol. 76, no. 11, p. 581, 2016.

[237] R. D. Ball *et al.*, “Parton distributions with LHC data,” *Nucl. Phys. B*, vol. 867, pp. 244–289, 2013.

[238] P. Skands, S. Carrazza, and J. Rojo, “Tuning PYTHIA 8.1: the Monash 2013 Tune,” *Eur. Phys. J. C*, vol. 74, no. 8, p. 3024, 2014.

[239] LHC SUSY Cross Section Working Group, “LHC SUSY Cross Section Working Group Twiki,” 2024.

[240] C. Borschensky, M. Krämer, A. Kulesza, M. Mangano, S. Padhi, T. Plehn, and X. Portell, “Squark and gluino production cross sections in pp collisions at $\sqrt{s} = 13$, 14, 33 and 100 TeV,” *Eur. Phys. J. C*, vol. 74, no. 12, p. 3174, 2014.

[241] W. Beenakker, C. Borschensky, M. Krämer, A. Kulesza, and E. Laenen, “NNLL-fast: predictions for coloured supersymmetric particle production at the LHC with threshold and Coulomb resummation,” *JHEP*, vol. 12, p. 133, 2016.

[242] T. S. R. Procter, “TP-thesis-susygridgenerator.” `https://github.com/tprocter46/TP-thesis-susygridgenerator`, 2024.

[243] J. Y. Araz, “Spey: Smooth inference for reinterpretation studies,” *SciPost Phys.*, vol. 16, no. 1, p. 032, 2024.

[244] E. M. Freundlich, *Searches for vector-like quarks with 13 TeV at the ATLAS experiment and development of a boosted-object tagger using a deep neural network.* PhD thesis, Tech. U., Dortmund (main), Tech. U., Dortmund (main), 2020.

[245] G. Aad *et al.*, “Search for pair-production of vector-like quarks in pp collision events at s=13 TeV with at least one leptonically decaying Z boson and a third-generation quark with the ATLAS detector,” *Phys. Lett. B*, vol. 843, p. 138019, 2023.

[246] A. Buckley, D. Kar, and K. Nordström, “Fast simulation of detector effects in Rivet,” *SciPost Phys.*, vol. 8, p. 025, 2020.

[247] A. J. Larkoski, G. P. Salam, and J. Thaler, “Energy Correlation Functions for Jet Substructure,” *JHEP*, vol. 06, p. 108, 2013.

[248] J. H. Foo, “Search for single production of a vector-like T quark decaying into a Higgs boson and top quark with fully hadronic final states using the ATLAS detector,” *PoS*, vol. LHCP2022, p. 321, 2023.

[249] G. Aad *et al.*, “Search for vector-boson resonances decaying into a top quark and a bottom quark using pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *JHEP*, vol. 12, p. 073, 2023.

[250] “A search for top-squark pair production, in final states containing a t-quark, c-quark and missing transverse momentum, using the full Run 2 dataset collected by the ATLAS detector,” 2023.

[251] G. Aad *et al.*, “Performance of Top Quark and W Boson Tagging in Run 2 with ATLAS,” 8 2017.

[252] G. Aad *et al.*, “Measurement of jet substructure in boosted tt¯ events with the ATLAS detector using 140 fb-1 of 13 TeV pp collisions,” *Phys. Rev. D*, vol. 109, no. 11, p. 112016, 2024.

[253] J. Y. Araz *et al.*, “Les Houches guide to reusable ML models in LHC analyses,” 12 2023.

[254] S. Bieringer, G. Kasieczka, J. Kieseler, and M. Trabs, “Classifier Surrogates: Sharing AI-based Searches with the World,” 2 2024.

[255] R. Bellman, R. Corporation, and K. M. R. Collection, *Dynamic Programming.* Rand Corporation research study, Princeton University Press, 1957.

[256] M. Köppen, “The curse of dimensionality,” in *5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1, pp. 4–8, 2000.

[257] W. J. Fawcett, "pMSSM studies with ATLAS and CMS," *PoS*, vol. LHCP2016, p. 146, 2016.

[258] K. Cranmer, J. Pavez, and G. Louppe, "Approximating Likelihood Ratios with Calibrated Discriminative Classifiers," 6 2015.

[259] S. Jiggins and L. Vesterbacka, "CARL-TORCH." https://github.com/sjiggins/carl-torch, 2022.

[260] G. Aad *et al.*, "Search for squarks and gluinos in final states with one isolated lepton, jets, and missing transverse momentum at $\sqrt{s} = 13$ with the ATLAS detector," *Eur. Phys. J. C*, vol. 81, no. 7, p. 600, 2021. [Erratum: Eur.Phys.J.C 81, 956 (2021)].

[261] S. Jiggins and L. Vesterbacka, "carlAthenaOnnx." https://github.com/sjiggins/carl-torch, 2022.

[262] A. Butter, "Machine learning and LHC event generation." `https://indico.cern.ch/event/1180220/`, 2022.

[263] W. Jakob, J. Rhinelander, and D. Moldovan, "pybind11 – seamless operability between c++11 and python," 2017. https://github.com/pybind/pybind11.

[264] M. Aaboud *et al.*, "Search for pair production of higgsinos in final states with at least three b-tagged jets in $\sqrt{s} = 13$ TeV pp collisions using the ATLAS detector," *Phys. Rev. D*, vol. 98, no. 9, p. 092002, 2018.

[265] G. Aad *et al.*, "Search for supersymmetry in events with four or more charged leptons in 139 fb^1 of $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector," *JHEP*, vol. 07, p. 167, 2021.

[266] A. Tumasyan *et al.*, "Search for electroweak production of charginos and neutralinos in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 04, p. 147, 2022.

[267] G. Aad *et al.*, "Measurement of the $Z(\rightarrow \ell^+\ell^-)\gamma$ production cross-section in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *JHEP*, vol. 03, p. 054, 2020.

[268] G. Aad *et al.*, "Measurement of $Z\gamma\gamma$ production in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *Eur. Phys. J. C*, vol. 83, no. 6, p. 539, 2023.

[269] M. Aaboud *et al.*, "Measurement of detector-corrected observables sensitive to the anomalous production of events with jets and large missing transverse momentum in pp collisions at $\sqrt{\mathbf{s}} = \mathbf{13}$ TeV using the ATLAS detector," *Eur. Phys. J. C*, vol. 77, no. 11, p. 765, 2017.

[270] M. Aaboud *et al.*, "Measurement of fiducial and differential W^+W^- production cross-sections at $\sqrt{s} =$ 13 TeV with the ATLAS detector," *Eur. Phys. J. C*, vol. 79, no. 10, p. 884, 2019.

[271] G. Aad *et al.*, "Measurements of differential cross-sections in four-lepton events in 13 TeV proton-proton collisions with the ATLAS detector," *JHEP*, vol. 07, p. 005, 2021.

[272] G. Aad *et al.*, "Measurements of fiducial and differential cross sections for Higgs boson production in the diphoton decay channel at $\sqrt{s} = 8$ TeV with ATLAS," *JHEP*, vol. 09, p. 112, 2014.

[273] J. Y. Araz, A. Buckley, B. Fuks, H. Reyes-Gonzalez, W. Waltenberger, S. L. Williamson, and J. Yellen, "Strength in numbers: Optimal and scalable combination of LHC new-physics searches," *SciPost Phys.*, vol. 14, no. 4, p. 077, 2023.

[274] G. Aad *et al.*, "Search for singly produced vectorlike top partners in multilepton final states with 139 fb-1 of pp collision data at s=13 TeV with the ATLAS detector," *Phys. Rev. D*, vol. 109, no. 11, p. 112012, 2024.

[275] R. Franceschini, "SUSY overview and new directions for run 3," in *ATLAS SUSY workshop 2023 (Oslo)*, `https://indico.cern.ch/event/1322199/`, 9 2023.

[276] A. M. Sirunyan *et al.*, "Search for physics beyond the standard model in events with jets and two same-sign or at least three charged leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV," *Eur. Phys. J. C*, vol. 80, no. 8, p. 752, 2020.

[277] A. M. Sirunyan *et al.*, "Search for supersymmetry in final states with two oppositely charged same-flavor leptons and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 04, p. 123, 2021.

[278] G. Aad *et al.*, "Search for squarks and gluinos in final states with same-sign leptons and jets using 139 fb^{-1} of data collected with the ATLAS detector," *JHEP*, vol. 06, p. 046, 2020.

[279] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lönnblad, J. Monk, H. Schulz, and F. Siegert, "Rivet user manual," *Computer Physics Communications*, vol. 184, no. 12, pp. 2803–2819, 2013.

[280] J. Thaler and K. Van Tilburg, "Identifying boosted objects with n-subjettiness," *Journal of High Energy Physics*, vol. 2011, no. 3, pp. 1–28, 2011.

[281] M. Aaboud *et al.*, "Measurements of inclusive and differential fiducial cross-sections of $t\bar{t}\gamma$ production in leptonic final states at $\sqrt{s} = 13$ TeV in ATLAS," *Eur. Phys. J. C*, vol. 79, no. 5, p. 382, 2019.

[282] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *Software: practice and experience*, vol. 30, no. 11, pp. 1203–1233, 2000.

[283] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.

[284] G. Aad *et al.*, "Search for single production of vector-like quarks decaying into Wb in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector," *Eur. Phys. J. C*, vol. 76, no. 8, p. 442, 2016.

[285] G. Aad *et al.*, "Search for single production of vector-like quarks decaying into Wb in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," 8 2016. ATLAS-CONF-2016-072, arXiv:1602.05606.

[286] M. Aaboud *et al.*, "Search for single production of vector-like quarks decaying into Wb in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *JHEP*, vol. 05, p. 164, 2019.

[287] "ATLAS Pythia 8 tunes to 7 TeV data," 11 2014. ATLAS-PHYS-PUB-2014-021.

[288] S. Frixione, E. Laenen, P. Motylinski, and B. R. Webber, "Single-top production in MC@NLO," *JHEP*, vol. 03, p. 092, 2006.

[289] S. Frixione, E. Laenen, P. Motylinski, B. R. Webber, and C. D. White, "Single-top hadroproduction in association with a W boson," *JHEP*, vol. 07, p. 029, 2008.

[290] C. D. White, S. Frixione, E. Laenen, and F. Maltoni, "Isolating Wt production at the LHC," *JHEP*, vol. 11, p. 074, 2009.

[291] G. Aad *et al.*, "Modelling and computational improvements to the simulation of single vector-boson plus jet processes for the ATLAS experiment," *JHEP*, vol. 08, p. 089, 2022.

[292] S. Kallweit, J. M. Lindert, P. Maierhöfer, S. Pozzorini, and M. Schönherr, "NLO electroweak automation and precise predictions for W+multijet production at the LHC," *JHEP*, vol. 04, p. 012, 2015.

[293] G. Cacciapaglia, A. Carvalho, A. Deandrea, T. Flacke, B. Fuks, D. Majumder, L. Panizzi, and H.-S. Shao, "Next-to-leading-order predictions for single vector-like quark production at the LHC," *Phys. Lett. B*, vol. 793, pp. 206–211, 2019.

[294] D. O'Brien, *Large Hadron Collider phenomenology of vector-like quarks beyond the Narrow Width Approximation.* PhD thesis, Southampton U., 2018.

[295] M. Aaboud *et al.*, "Jet energy scale measurements and their systematic uncertainties in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *Phys. Rev. D*, vol. 96, no. 7, p. 072002, 2017.

[296] S. Amoroso, C. Gutschow, R. Hawkings, Z. Marshall, B. Malaescu, and W. Verkerke, "Recommendations on the treatment of theoretical systematic uncertainties in statistical analysis of ATLAS data," tech. rep., CERN, Geneva, 2020.

[297] G. Aad *et al.*, "Measurement of differential cross-sections in $t\bar{t}$ and $t\bar{t}$+jets production in the lepton+jets final state in pp collisions at $\sqrt{s} = 13$ TeV using 140 fb^{1} of ATLAS data," *JHEP*, vol. 08, p. 182, 2024.

[298] G. Aad *et al.*, "Search for single-production of vector-like quarks decaying into Wb in the fully hadronic final state in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," 9 2024. arXiv:2409.20273.

[299] A. M. Sirunyan *et al.*, "Search for single production of vector-like quarks decaying into a b quark and a W boson in proton-proton collisions at $\sqrt{s} = 13$ TeV," *Phys. Lett. B*, vol. 772, pp. 634–656, 2017.

[300] G. Aad *et al.*, "Search for new particles in final states with a boosted top quark and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *JHEP*, vol. 05, p. 263, 2024.

[301] "8th LHC BSM Reinterpretation Forum Workshop." `https://conference.ippp.dur.ac.uk/event/1178/`, 29 Aug. – 1 Sep. 2023, Durham Univ., Durham, UK. Accessed: 2023-12-13.

Appendices

Appendix A

Additional MCBOT plots

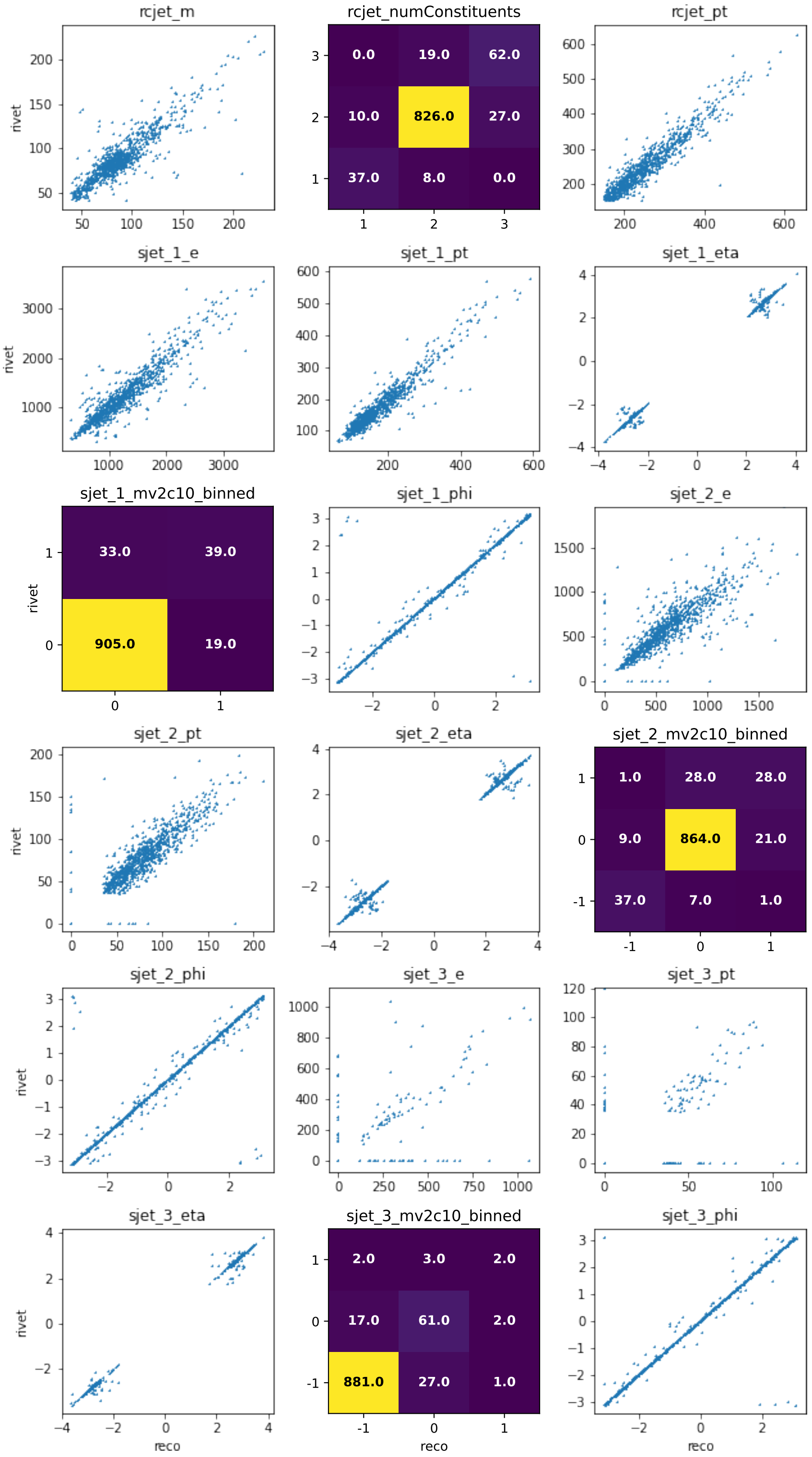


Figure A.1: All 18 MCBOT inputs compared between Rivet detector emulation and full Atlas reconstruction of the same truth-level events.

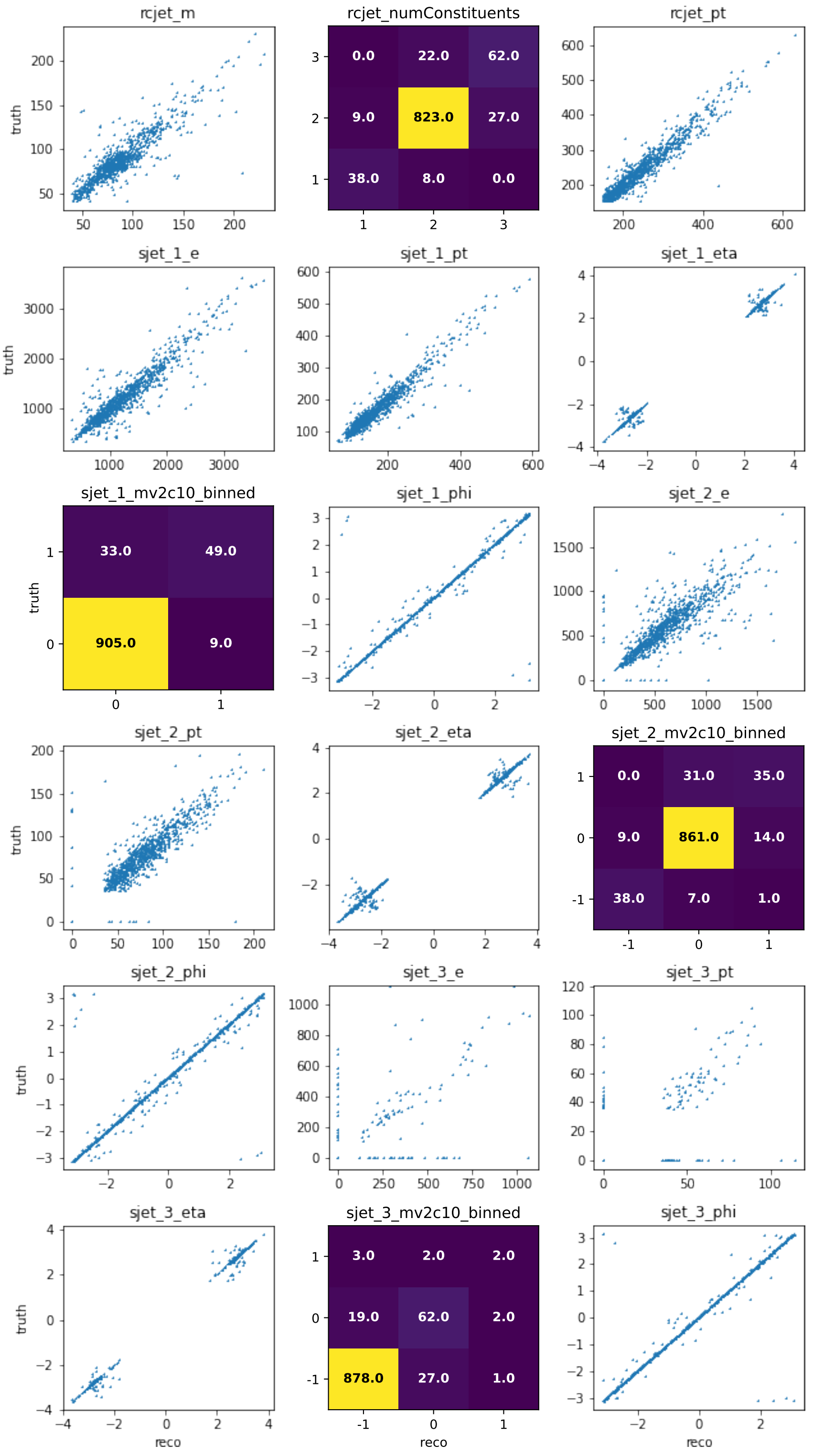


Figure A.2: All 18 MCBOT inputs compared between the original truth-level events, and full Atlas reconstruction applied to the same truth-level events.

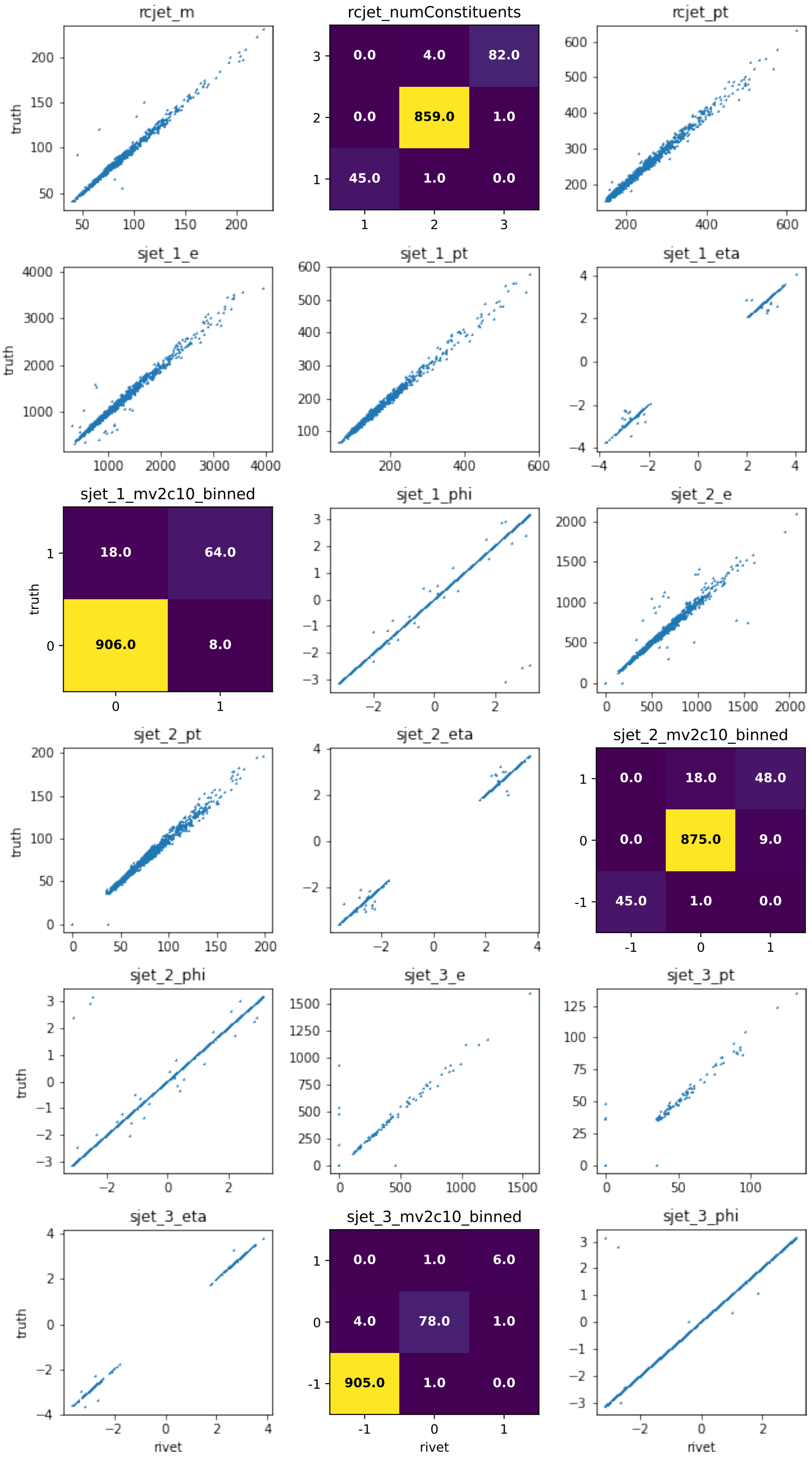
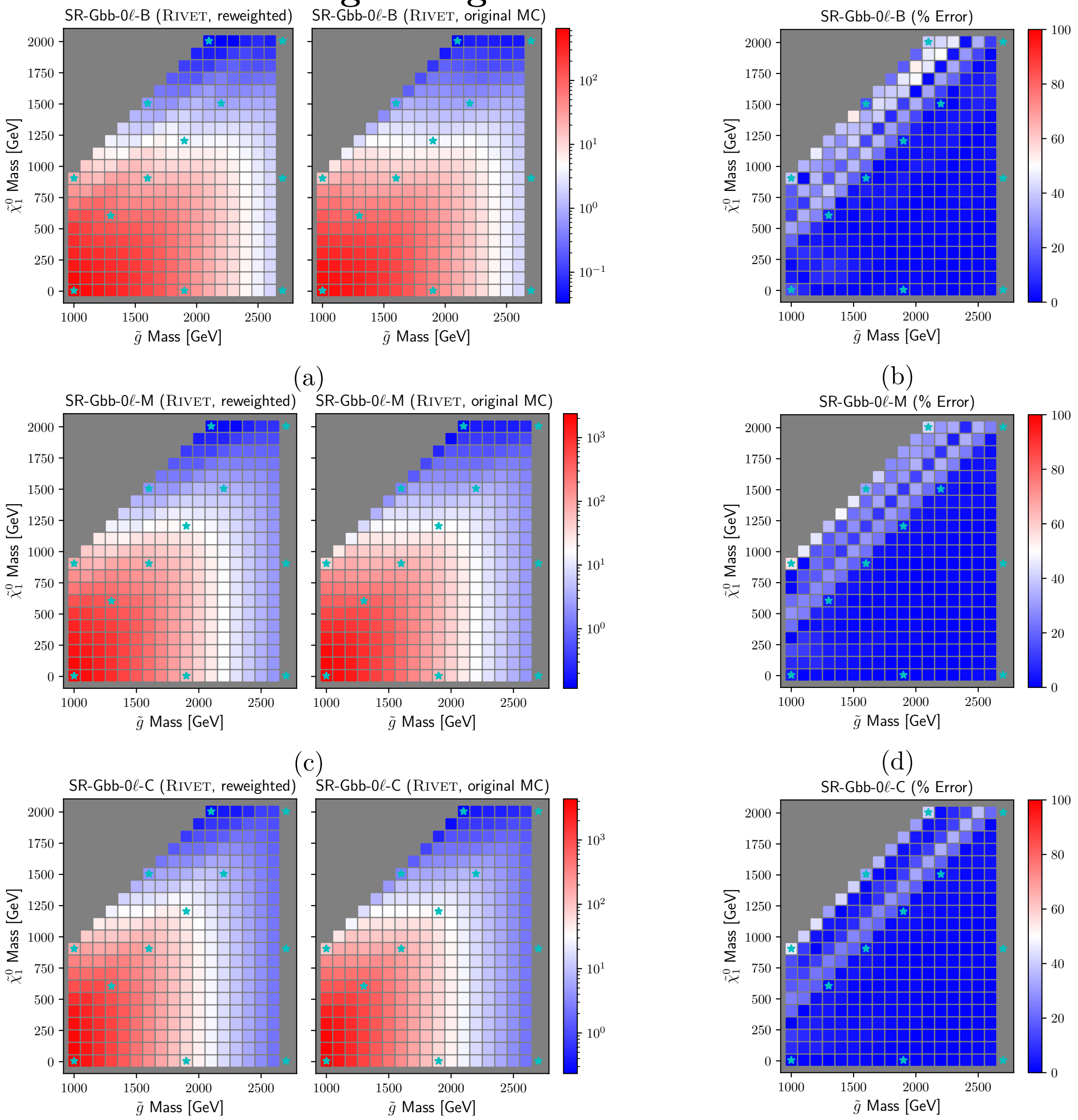


Figure A.3: All 18 MCBOT inputs compared between the original truth-level events and rivet detector emulation of those same truth-level events.

Appendix B

Using CARL to reweight RIVET results: additional plots

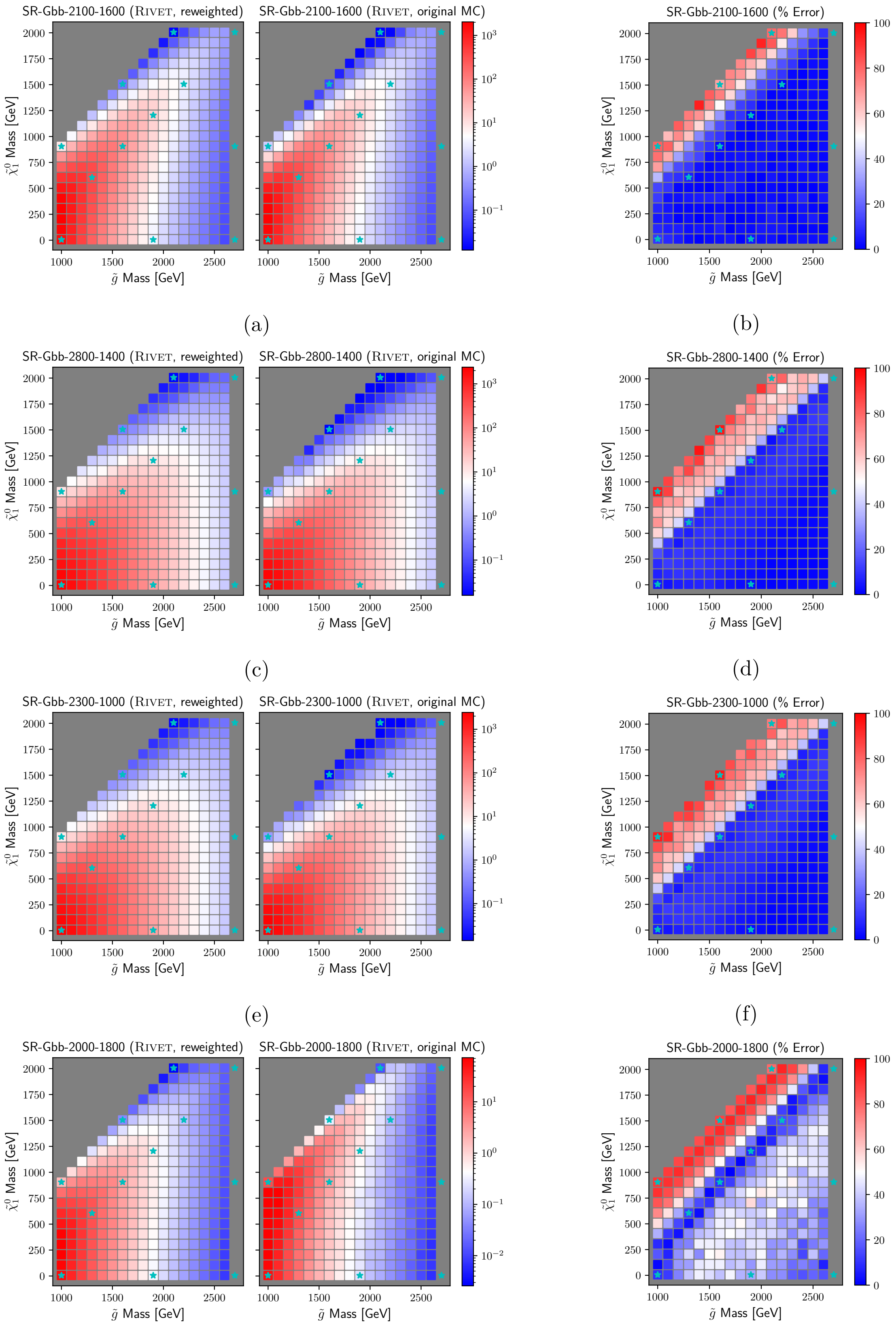
B.1 Cut-and-count signal regions



(a) (b) (c) (d) (e) (f)

Figure B.1: Comparison of the event counts in the cut-and-count signal regions, comparing the reweighted counts (left), to those obtained by event generation at each point (centre), with the percentage error between the two on the right. Cyan stars mark points in the nominal sample.

B.2 Neural-net signal regions



(a) (b)

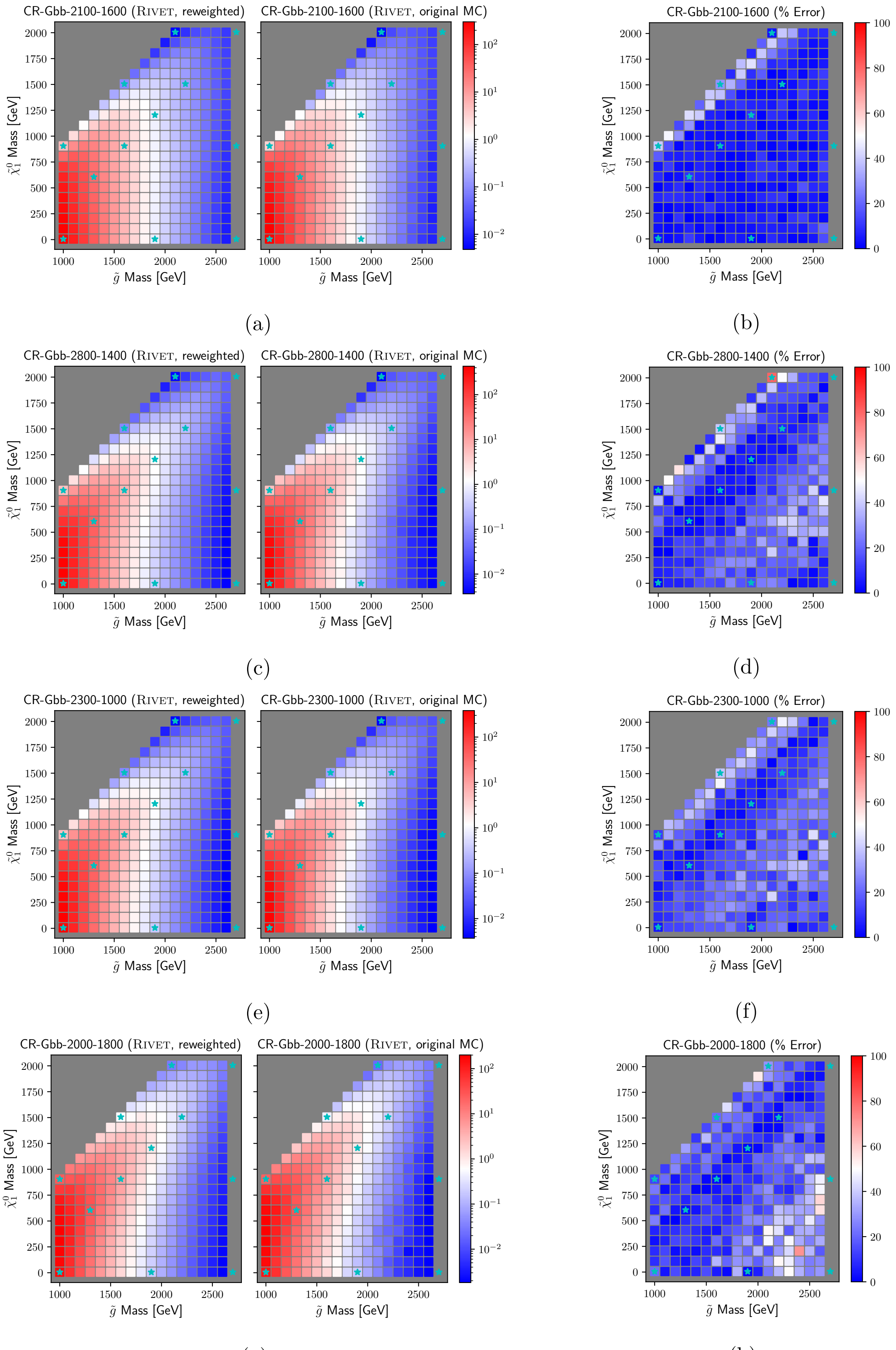
(c) (d)

(e) (f)

(g) (h)

Figure B.2: Comparison of the event count in each of the NN-based signal regions, comparing the reweighted counts (left) to those obtained by event generation at each point (centre), with the percentage error between the two on the right. Cyan stars mark points in the nominal sample.

B.3 Neural-net control regions



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure B.3: Comparison of the event count in each of the NN-based control regions, comparing the reweighted counts (left) to those obtained by event generation at each point (centre), with the percentage error between the two on the right. Cyan stars mark points in the nominal sample.

Appendix C

New Rivet + Contur API example

```python
#! /usr/bin/env python

# Tomasz Procter 2022.
# (Significant) evolution of a script I got from Andy Buckley in early 2021

from contur.run.arg_utils import get_argparser
from contur.run.run_analysis import main
from io import StringIO
import os
import rivet

# Options that might need to be frequently changed

### Rivet options
hepmcfiles = [os.path.abspath("/path/to/your/hepmcfile.hepmc.gz")]
MAX_EVTS=100
analyseslist = ["ATLAS_2021_I1849535", "ATLAS_2017_I1614149"]

### Contur options: more available! See contur argparser.
ConturArgs = {
  "STAT_OUTPUT_TYPE": "CLs",
  ’YODASTREAM_API_OUTPUT_OPTIONS’: ["LLR", "Pool_LLR", "Pool_tags"]
}

##################################
# Real code starts here

# Deal with rivet options:
if (type(hepmcfiles) is not list):
  hepmcfiles = [hepmcfiles]

# Let’s run rivet
ah = rivet.AnalysisHandler()
for analysis in analyseslist:
ah.addAnalysis(analysis)

run = rivet.Run(ah)
for hepmcfile in hepmcfiles:
  run.openFile(hepmcfile)
  eventcount = 0
  while True:
    ok = run.readEvent()
    if not ok:
      break
    ok = run.processEvent()
    if not ok:
      break
    eventcount+=1
    if eventcount > MAX_EVTS:
      break
run.finalize()

#################################
# Transform output and run Contur

sio = StringIO()
# Write yoda to stringstream
ah.writeData(sio)
# For debugging and cross-checks, print the yoda
ah.writeData("test.yoda")
# We could also reload a yoda file:
# sio = open("test.yoda", "r")
sio.seek(0)

# Setup argparser
args = get_argparser('analysis')
args = args.parse_args()
args = vars(args)
args['YODASTREAM'] = sio

for arg in ConturArgs:
  args[arg]=ConturArgs[arg]

output = main(args)
poolsdict = output["Pool_LLR"]
pooltags = output["Pool_tags"]

print("Output Likelihood = ", output['LLR'])
print("POOLS:")
for pool in poolsdict:
  print(pool,pooltags[pool], ": ",poolsdict[pool])
```