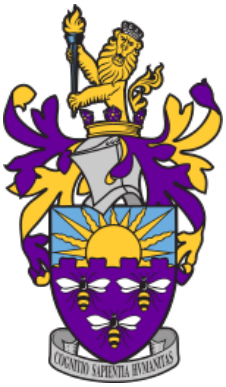


PHYS40181 MPHYS PROJECT

Using Fisher Information to enhance searches for Dark Matter at the LHC

Genís Marfà Martínez (9917395)

Supervisors: Dr. Darren Price & Dr. Stephen Menary

This experiment was performed in collaboration with Ahmad Alachkar.

Abstract:

Searches beyond the Standard Model are frequently characterised by low sensitivities when looking for direct evidence in high energy colliders. One such example is in searches for Dark Matter in events producing jets of high transverse momentum. Here we propose Fisher Information as a potential tool to discriminate how much we can learn about a model's parameters from a given distribution of measurements. When used in addition to the Crámer-Rao bound, lower bounds can be set on the uncertainty of any estimator of the model's parameters.

Here we sort out different variables and distributions of an EFT DM model, according to their FI to assist future experimental design. To do so, MadGraph5 at leading order was used to generate Monte Carlo samples in two different regions of phase space. ATLAS data was imported to draw 95% confidence level intervals to asses the lack of sensitivity, and to investigate what particular measurements drove the limit. Finally, likelihood fits on the effective field theory scale Λ_{EFT} were carried out to produce a per-bin prediction of the information content of 9 different distributions.

Our analysis concludes that missing momentum in the transverse direction, p_T^{miss}, is the preferred variable for this search. Bins with the most FI have been pointed out, to center focus in future experiments of this sort. We also propose variables like leading jet pseudorapidity for future collider experiments, due to their high predicted information.

1 Introduction:

Many modern particle physics search experiments are characterised by having large backgrounds and rather poor statistics [1]. Consequently, multivariate analysis (MVA) techniques have become increasingly popular in experiments with low sensitivity, such as explorations beyond the standard model. In addition, the increased popularity of machine learning tools has opened the gates to an overwhelming number of possibilities when it comes to experimental designing. With all these capabilities, it has become essential now more than ever, to effectively quantify how much information we can obtain from a particular data set or distribution. A possible metric to compute this is the Fisher information matrix which, in conjunction with the Cramér-Rao bound, establishes the maximum information on the model parameters given a particular observation [2]. Well aware that the scope of its applications is very broad, in this project the use of Fisher information will be examined in low sensitivity searches for dark matter signatures in an effective field theory (EFT). The analysis presented below builds on from an experiment carried out at ATLAS in 2017 [3], whose details are briefly summarised hereunder.

Despite the standard model's success at describing the fundamental particles in nature, it fails to explain the abundance of dark matter (DM) in the universe. Evidence for it's existence can be inferred from cosmological phenomena, such as measurements of rotation curves in spiral galaxies [4]. However, the constituents of DM on a microscopic level are far from known. This is one of the main goals at the large hadron collider (LHC), which searches for evidence of new physics following predictions from beyond standard model (BSM) theories. One such theory is suppersymmtery (SuSy), which provides candidate dark matter WIMP fermions in the lightest neutralino [5]. Their presence in a detector can be inferred from the so-called missing transverse momentum, defined as the momentum imbalance of the final state particles, in the plane transverse to the colliding beams. Two main WIMP production mechanisms, corresponding to different regions of phase space, will be considered:

1. **Vector Boson Fusion** $pp \to jj + p_T^{miss}$: Fusion of vector W bosons into mediator A, decaying into WIMP $\chi\bar{\chi}$ pair. Two well separated, high energy jets + p_T^{miss} are expected in the final state. The Feynman diagram for the process is shown in figure 1. The corresponding region of phase space will be referred to as the VBF region hereafter.

2. **Monojet signature** $pp \to j + p_T^{miss}$: Direct production of mediator A, decaying into WIMP $\chi\bar{\chi}$ pair. At least one gluon-initiated jet + p_T^{miss} is expected at the final state. The Feynman diagram for the process is shown in figure 2. The corresponding region of phase space will be referred to as the ≥ 1 jet region hereafter.

The dominant standard model counterparts for these processes yielding the same final states, are where the A mediator is replaced by the Z boson, which decays into a pair of neutrinos. Feynman diagrams for the two backgrounds are shown in figures 3 and 4.

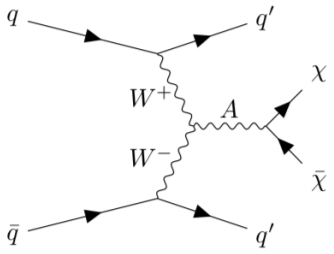


Figure 1: WIMP $\chi\bar{\chi}$ production along with two high energy jets.

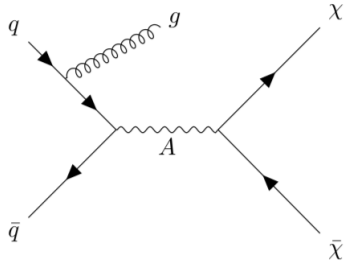


Figure 2: Direct WIMP $\chi\bar{\chi}$ production along with high energy, gluon initiated jet.

To mitigate systematics and detector inefficiencies, the quantity R^{miss} was measured at ATLAS, defined as the ratio:

$$R^{miss} = \frac{\sigma(p_T^{miss} + \text{jets})}{\sigma(l^+l^- + \text{jets})} \tag{1}$$

Where the numerator corresponds to the $Z \to \nu\bar{\nu} + \text{jets}$ events, and the denominator is where the Z boson decays into a pair of leptons (electrons or muons) instead. If the numerator cross-section is influenced by BSM physics, discrepancies should arise between the measured ratio and that predicted by the SM.

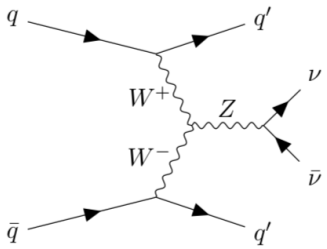


Figure 3: Electroweak $Z \to \nu\bar{\nu}$ background

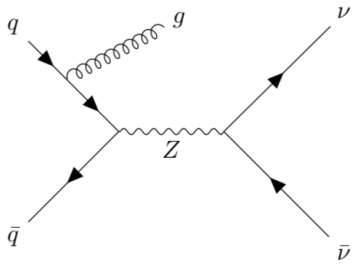


Figure 4: QCD $Z \to \nu\bar{\nu}$ background

The aim of this project is to use the standard model to predict the expected information in future experiments. Consequently and for simplicity, special focus will be placed on the numerator only. Different variables and distributions will be simulated using Monte Carlo methods, and tested against those in the original experiment. The expected SM predictions used at ATLAS will be used for cross-checks and validation of the Monte Carlo. These are available publicly in Ref. [6]. Furthermore, experimental sensitivity will be assessed by drawing 95% confidence level exclusion contours on the different dark matter model parameters.

2 The ATLAS detector and event selction:

A Toroidal LHC ApparatuS (ATLAS) is one of the seven detectors at the large hadron collider (LHC). It contains a series of trackers, calorimeters and muon chambers arranged in a layered design, making it a general-purpose detector.

The trackers are located within the inner detector (ID), located relatively close to the beam axis. It covers the pseudorapidity range $\mid \eta \mid < 2.5$. A 2T solenoidal magnet surrounds the ID. The extent to which charged particles' paths curve in the field determines their direction, charge and momentum [7]. Outside the magnet are the calorimeters, which "absorb" the particles' energy, bringing them to complete stoppage. The energy is deposited into the detector, which is recorded to reconstruct the particle's type. Hadronic and EM calorimeters cover the rapidity ranges $\mid \eta \mid < 4.9$ and $\mid \eta \mid < 3.2$ respectively [8].

The only known particles that can pass through the calorimeters are muons and neutrinos. Muons are identified at the muon spectrometer, the outermost layer of the detector. Up to 4000 chambers manage to construct tracks for these particles, which are then matched to those at the trackers. With the information provided by the two systems, and correcting for the energy deposited in the calorimeters, the muon's energy can be found. Once all the final state particles are gathered, the negative vector sum of all the transverse momenta yields the missing transverse momentum p^T_{miss}, associated to neutrinos. Tau leptons are identified from their decay products, by using MVAs such as boosted decision trees on the shapes of their calorimeter showers.

When colliding hardrons at high luminosities, large backgrounds can be observed. To maximise the number of signal events in the desired topologies, a series of kinematic cuts are imposed on the final state particles. Some of these are experimental requirements on the calorimeters or jet calibration but they mostly come from requirements of the phase-space distributions considered. This defines a fiducial region that corresponds to what is experimentally available from the detector. For the process $pp \to p_T^{miss} + \text{jet(s)}$ in the ≥ 1 jet and VBF regions, these are summarised in table 1.

The $p_T^{miss} > 200$ GeV cut is a requirement for the effective functioning of the trigger when selecting events. Furthermore, rapidity cuts are imposed to avoid the edges of the calorimeter. The cut on $\Delta\phi$ between the jet axis and p_T^{miss} is used to minimise backgrounds due to events with a high number of final sate jets. In these events, the p_T^{miss} of softer jets is often missread. Consequently, it is frequent that most of the p_T^{miss} lies in the direction of the jet. However, requiring the azimuthal separation between the two to be > 0.3 heavily suppresses this background.

Moreover, as a common trait of the VBF topology, the suppressed QCD activity in the dijet gap encourages a centrality veto [9]. Also, to minimise backgrounds where one of the bosons in the exchange may decay hadronically, a cut is placed on m_{jj}. Cuts on the jet p_T are required by the jet reconstruction algorithms. In

Selection Cut	$\geq$ 1 jet	VBF
p_T^{miss}	$>$ 200 GeV	
Jet $\mid y \mid$	$<$ 4.4	
Jet p_T	$>$ 25 GeV	
$\Delta\phi_{jet,p_T^{miss}}$	$>$ 0.4 for 4 leading jets with $pT >$ 30 GeV	
Lepton Cut	No e or μ with $p_T >$ 7 GeV and $\mid \eta \mid <$ 2.5	
Leading jet p_T	$>$ 120 GeV	$>$ 80 GeV
Subleading jet p_T	-	$>$ 50 GeV
Leading jet $\mid \eta \mid$	$<$ 2.4 GeV	-
m_{jj}	-	$>$ 200 GeV
Centrality		No jets with $p_T >$ 25 GeV

Table 1: Kinematic cuts on different variables defining fiducial phases spaces for ≥ 1 jet and VBF regions. m_{jj} is the invariant mass of the leading and subleading jets, and $\Delta\phi_{jet,p_T^{miss}}$ is the difference in azimuthal angle ϕ between the jet direction and p_T^{miss}.

the ≥ 1 region, at least one high energy jet is expected, so a cut is imposed on the leading jet p_T. Similarly, at least two high energy jets are expected in the VBF region, so cuts are imposed on the leading and subleading jet p_T. Finally an additional lepton veto discards any $W \to l +$ jets events, where the charged lepton might be outside calorimeter acceptance, or has not been successfully reconstructed.

Having now seen how the cuts on the kinematic variables may be used to maximise signal identification, all the preambles left to do are generating the Monte Carlo samples for posterior analysis.

3 Monte Carlo event simulation and generator level cuts:

It comes as no surprise that running an experiment at the LHC requires a lot of resources. For this reason, Monte Carlo (MC) simulations are widely used for calibration, cross-checks with theory predictions, limit setting and overall testing for new physics. These algorithms involve drawing a random sample from a probability distribution to simulate a desired output, which would otherwise be very complex to find analytically. In the context of particle physics, these are used to simulate parton showers under the appropriate PDFs, for a given hard process. The output after one such simulation will encode the four momenta of all the final state stable (or quasi-stable) particles. The underlying procedure in a MC simulation can be summarised in 3 steps [10]:

1. **Matrix Element Calculation:** This is where most of the physics occurs: PDF evaluation, amplitudes, phase spaces, spin correlations, etc.
2. **Parton Shower:** The hard process is simulated from the above calculated PDFs for the incoming partons. These then showered leading to the various many subprocesses. Hard particles such as the top quark decay before hadronisation.
3. **Harmonisation:** Outgoing gluons and quarks form hadronic jets. Unstable hadrons decay into detectable products.

Frequently, all three steps are ran independently through different software. Most generators will only execute the first step, and will not evolve the partons into final state particles. However, there a few multipurpose generators which will execute the full procedure. Examples of these include `PYTHIA`, `SHERPA` and `HERWIG`. An even further step is then to simulate how the outgoing products interact with detector materials. Available software to do so include `GEANT4` or `FLUKA`.

Often, each event in a MC sample is given a certain amplitude or weight, such that the sum over all such weights squared is (approximately) equal to the process' cross-section. The extent to which these agree will depend on the initial PDF calculations. A good way to correct for this discrepancy is to reweigh all the data by a scale factor defined as:

$$\text{Scale Factor} = \frac{\text{Cross-Section}}{\sum_{i=1}^{N_{tot}} w_i^2} \tag{2}$$

Where N_{tot} is the total number of data points and w_i their weights. If this factor is close to 1, it would be an indication that the matrix element calculation has been carried out very precisely by our generator.

Hence, for binned data, we can assume Poisson errors (err $\sim \sqrt{N}$) where the error at a particular bin, err_{bin} is given by:

$$err_{bin} = \sqrt{L_{int} \times \sigma_{bin}} = \sqrt{L_{int} \times \left(\sum_{i=1}^{N_{bin}} w_i^2 \right) \times \text{Scale Factor}} \tag{3}$$

Where L_{int} is the integrated luminosity, and we have used that fact that $N = \sigma L_{int}$. Here N_{bin} is the number of MC events in that bin, and w_i are the weights of these events.

For the project at hand, `MADGRAPH5@LO` v.2.6.7 interfaced to `PYTHIA` v.8.243 was used to produce MC samples for both the standard model backgrounds and the dark matter signals. A brief discussion of the generator is included below. The user manual is included in reference [11].

3.1 The `MADGRAPH` generator:

`MADGRAPH` is a general purpose event generator. The matrix element calculation makes use of the tree-level approximation. This is a low order approximation which treats Feynman diagrams with quantum loop corrections, arising from higher order terms in perturbation, as much simpler "tree" diagrams [12]. The generator includes a full description of the standard model, and allows importing BSM models at any collider. Given a certain hard process, `MADGRAPH` computes amplitudes for all the relevant subprocesses, which are then fed into the `MadEvent` matrix element calculator. Phase space integrals are then carried out for each subprocess to compute the cross-sections. After the event generation, information such as particle ids, momenta, spin, etc. is saved into a format which can be interfaced directly to a parton shower generator such as `PYTHIA`.

The details of the event generation can be modified in the `run` and `parameter` cards. The former includes things like number of events, beam energies, renormalisation scales and kinematic cuts. The latter contains all the different free parameters in the model. For the standard model, for example, this includes things like particle masses, coupling strengths, Cabibbo mixing angles, etc.

`PYTHIA` contains a broad library of hard processes, parton showers, event merging methods, particle decays and many other useful tools to evolve the hard process into a showered multi-particle state [13]. This software is purposely designed for high energy processes, as the allowed final states are approximated by a continuum. At lower energies we enter the regime for hadronic resonance, where this approximation fails. All components of the cross-section are described, including softer processes due to elastic scattering and non-diffractive topologies. The main showering algorithms are ISR and FSR, which shower particles in order according to their p_T, until the final state partons reach the hadronisation energy scale ($\sim$ 1GeV) [14]. The shower evolution for both algorithms is based on the DGLAP splitting kernel, which describes the probability that a quark emits a gluon $P(q \to qg)$. After all events undergo showering, they are combined and collected in the standard event record HepMC format.

3.2 The EFT D7a model:

To simulate Monte Carlo events for BSM physics, models can be imported into the MadGraph generator. In this analysis, the D7a model by N. Christensen, C. Duhr and B. Fuks [15] will be used. This model relies on electroweak gauge bosons as the primary source of interaction between the standard model constituents and dark matter. The latter is described under the framework of an effective field theory (EFT), which attempts to describe new physics without knowing the exact details of the theory. This is achieved by ignoring all effects beyond a certain length or energy threshold. EFTs are self-consistent field theories themselves, in the sense that matrix elements can be computed from the EFT Lagrangian alone. They also contain renormalisation schemes that make matrix elements finite. Overall, they behave jut like any other quantum field theory.

Frequently, EFTs are used as a low energy limit approximation to some *full theory*, since they allow calculations of measurable quantities within some finite error. This error is determined by the number of terms expanded in perturbation about the counting parameter δ. This means that including more terms of the expansion reduces the error, although this might be complicated sometimes. The expansion parameter δ is commonly chosen as the ratio of some low-energy scale, such as external momentum, to some short-distance

scale Λ, such that $\delta = \frac{p}{\Lambda}$ [16].

The main task when describing an EFT is choosing an appropriate Lagrangian, which usually requires three steps:

1. Finding the degrees of freedom, and thus the field content.
2. Renormalisation: Which involves mitigating singularities caused by self-interactions.
3. Choosing appropriate couplings.

Many popular BSM models are a form of standard model EFT (SMEFT) which are built upon multidimensional versions the basic fields in the standard model. Dimension-4 terms yield the SM Lagrangian, but higher dimensional operators can be constructed to describe BSM physics. The dimension of these operators may determine some physical quantities, such as their CP property.

In the D7a EFT, dark matter constituents are assumed to be electrically neutral, and belong to some multiplet χ. Thus, the only light degrees of freedom considered include the matter content of the SM plus the additional dark matter χ, which is either Majorana or Dirac fermionic. A dimension 7 operator V couples χ to the different SM bosons: γ, Z^0 and $W^{\pm}$. This operator satisfies invariance under the electromagnetic unitary group, but is not required for the electroweak $SU(2)_L \times U(1)_Y$, to allow for symmetry breaking.

A short-distance scale Λ is used in the perturbation expansion of the EFT. This scale can be understood more generally in UV completion models as:

$$\frac{1}{\Lambda^2} \sim \frac{g_\chi g_{SM}}{M^2}, \tag{4}$$

where g_χ and g_{SM} are the couplings of the dark matter and SM particles to the hypothetical mediator A, and M it's mass. Therefore, each dark matter model is completely determined by two parameters: the EFT scale Λ_{EFT} and the dark matter mass m_χ. These are modifiable in the `MADGRAPH parameter` card. Limits on dark matter production have been previously set on the $m_\chi - \Lambda_{EFT}$ plane in Ref. [15].

The Lagrangian form of the D7a operator is $\chi\bar{\chi}V^{\mu\nu}V_{\mu\nu}$, which is CP-conserving. The chosen canonical normalisation is Λ^{-3}. This implies that for a given process, the dark matter only cross-section should satisfy $\sigma_{DM} \propto \Lambda^{-6}$. Therefore, given the distribution for some dark matter cross-section at a particular $\Lambda = \Lambda_1$, it can be re-normalised to some other Λ_2, following:

$$\sigma_{DM}^{\Lambda_1} = \left(\frac{\Lambda_2}{\Lambda_1}\right)^6 \sigma_{DM}^{\Lambda_2} \tag{5}$$

Hence, given a set of high precision Monte Carlo events at a certain Λ, they can easily extended to the whole line of constant m_χ by the application of equation 5.

The D7a model only describes a few subprocesses, namely those including vertices of AA, AZ, ZZ or WW and hence, the direct production channel illustrated in figure 2 cannot be simulated. The VBF process containing AW vertices was simulated, along with the two SM backgrounds depicted in figures 3 and 4.

Once the three MC samples are collected, posterior data analysis is carried out in two phases:

1. Apply selection cuts on a per-event basis: Events failing to pass the cuts described in table 1 are discarded. This is done via `Rivet` v.2.7.0. A toolkit designed to unpack HepMC files and validate MC events. A routine can be written following an object oriented C++ framework, which provides the tools for particle-level analysis [17]. A rivet routine implementing the above described cuts is provided by C. Gutschow as part of the Rivet standard library [18]. This routine was used (albeit slightly modified) to apply cuts and write the different variables such as p_T^{miss}, m_{jj}, $\Delta\phi_{jj}$, etc. for the surviving events to a `.csv` file. See appendix 1 for the full code.
2. Posterior analysis: These files are then imported into a `PYTHON3` script, where libraries like `matplotlib` can be used to produce histograms of the distributions. Limit setting, likelihood fits and Fisher information calculations can also be carried out using scientific python libraries.

3.3 Event Selection Validation and Generator Level Cuts:

While imposing cuts on the kinematic variables may be of good practice to better identify signal events, there are times where these might be too aggressive and leave very few surviving events, leading to poor statistics. Defining efficiency as the ratio:

$$\text{Eff} = \frac{\text{Events Passing Cuts}}{\text{Events Simulated}} \cdot 100(\%), \tag{6}$$

allows for a numerical validation of the cuts outlined in table 1 for the three processes simulated. A very low efficiency would suggest that the cuts are potentially too restrictive.

Efficiencies were calculated after a first round of MC simulations for each of the three samples. Low efficiencies ($< 1\%$) were observed for the QCD $Z \to \nu\nu$ background in both phase space regions. Cut flow diagrams for this process are included in figures 5 and 6 for the two regions of phase space. These reveal that the final state jets do not posses enough p_T to pass through the cuts, since the p_T^{miss} cut killed approximately 99.7% of the events.

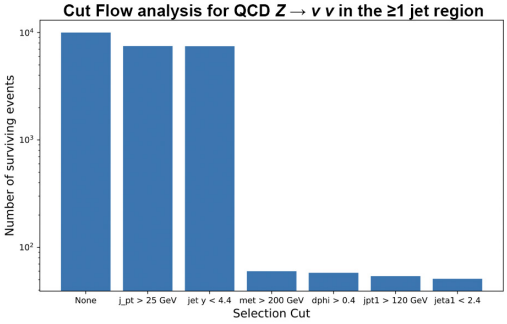


Figure 5: QCD $Z \to \nu\bar{\nu}$ cut-flow analysis for the ≥ 1 jet region.

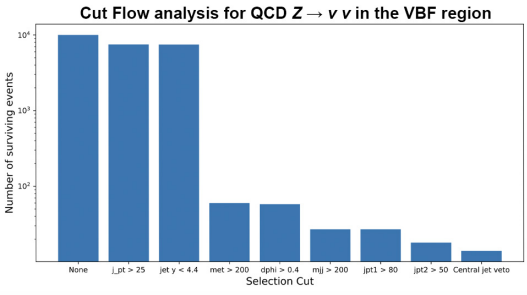


Figure 6: QCD $Z \to \nu\bar{\nu}$ cut-flow analysis for the VBF region

In order to be consistent with the experiment at ATLAS, rather than modifying the event selection and thus alter the fiducial regions, cuts were imposed on the minimum leading jet p_T at generator level. This was implemented by modifying the `ptj1min` paramater in the `MADGRAPH` run card. In doing so, the overall p_T of the final state is boosted, thus increasing the efficiency. This introduces a trade-off between efficiency improvement and bias introduced to the experiment.

`ptj1min` (GeV)	Cross-Section (pb)	Efficiencies (%)		Efficiency Increase Factor:	
		≥ 1 jet	VBF	≥ 1 jet	VBF
20	2250 ± 3	0.45	0.13	Benchmark	Benchmark
50	556.8 ± 0.7	1.93	0.53	4.31	4.01
60	379.7 ± 0.4	2.84	0.79	6.35	5.95
70	265.9 ± 0.3	4.07	1.07	9.08	8.06
80	189.8 ± 0.2	5.56	1.49	12.4	11.2

Table 2: Testing the effect of imposing generator level cuts on the minimum leading jet pT (`ptj1min`) as a way to increase efficiencies.

Distributions at different minimum leading jet p_T were considered. The cross-sections and efficiency increases are summarised in table 2. By default, `MADGRAPH` generates jets of at least $p_T > 20$ GeV, as seen in the `ptj` parameter in the run card, so this was taken as the benchmark. The full distributions with no kinematic cuts applied are shown in figure 7 for different values of `ptj1min`.

From figure 7, one can note that:

1. The curves peak at their respective `ptj1min` value and then decay smoothly, converging at high energies. Therefore, the bias between the curves is minimal at high p_T.

2. All events with leading jet $p_T < 80$ GeV are discarded in the VBF region in the posterior analysis. The cyan line indicates where this cut is imposed and hence, biases below this line can be ignored.

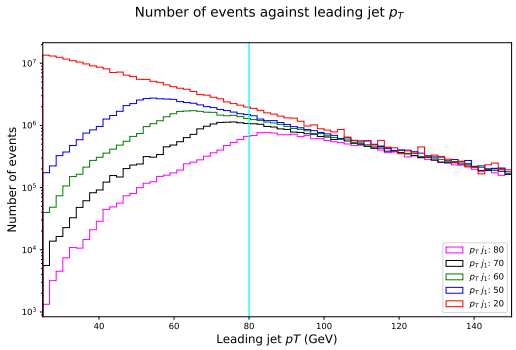


Figure 7: QCD $Z \to \nu\bar{\nu}$ distributions for leading jet p_T at different `ptj1min`.

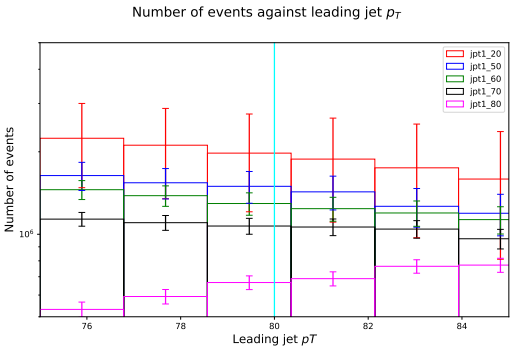


Figure 8: Distributions magnified around region of the cut (Leading jet $p_T > 80$). Everything below the cyan line is ignored posteriorly.

Consequently, only the bias around the region of the cut has to be considered. This region is magnified in figure 8. It follows that up to and including `ptj1min`: 70, the distributions lie within the Monte Carlo uncertainties of the benchmark case. For minimal biases and to retain a larger cross-section, the `ptj1min`: 60 case was chosen for the analysis. By imposing this generator cut, the same Monte Carlo precision is obtained as if (roughly) 6 times as many events were generated in the benchmark case. Updated cut flows diagrams comparing the two scenarios are included in figures 9 and 10.

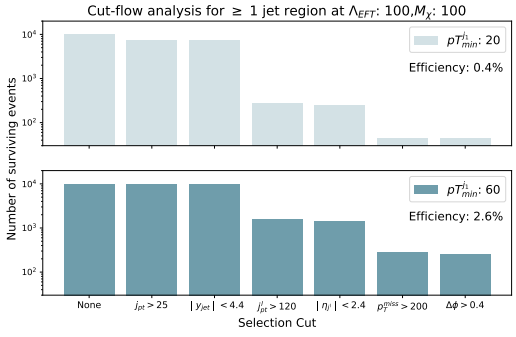


Figure 9: QCD $Z \to \nu\bar{\nu}$ cut-flow analysis before (top) and after (bottom) the `ptj1min` generator cut, in the ≥ 1 jet region.

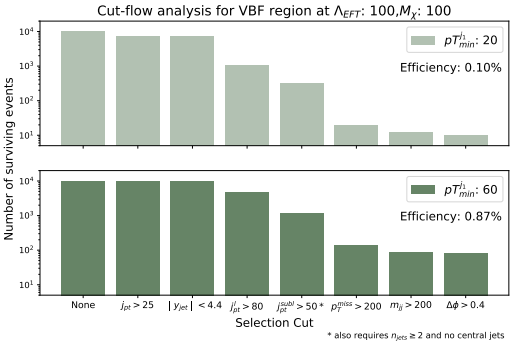


Figure 10: QCD $Z \to \nu\bar{\nu}$ cut-flow analysis before (top) and after (bottom) the `ptj1min` generator cut, in the VBF region.

4 SM Monte Carlo Validation:

As discussed in the previous section, frequently the matrix element calculation, parton shower and hadronisation stages of MC generators are ran independently. This was the case for the SM predictions in the original ATLAS experiment, where a series of tools were used to maximise the precision of the MC sample. The details are shortly summarised below:

- `SHERPA` v2.2.0 was used for the event generation.
- `Comix` and `OpenLoops` were used for the matrix element calculations at next-to-leading order (NLO) and merged with he `SHERPA` parton shower. The samples were then normalised to next-to-next-to leading order (NNLO).
- The merging of the multi-parton events was enhanced using `MEPS@NLO`.
- Alternative samples were simulated using `MG5@NLO` v2.2.2 at leading order, interfaced to `PYTHIA8` v1.8.6. for cross-checks and calculations of systematics.
- Furthermore, all samples where passed though `GEANT4` for detector simulations. Which included correction factors to account for the trigger.

Altogether, this produces a very precise MC sample which accounts for NLO and NNLO effects, available publicly at HepData, an online portal that collects data from high energy physics experiments. This sample will be referred to hereafter as the HepData SM prediction. In contrast, using `MADGRAPH` samples only contemplates LO effects. At ATLAS, four differential cross-sections were measured, with respect to: p_T^{miss} in both regions, m_{jj} and $\Delta\phi_{jj}$ in the VBF region only.

The discrepancy between the two standard model background simulations for these distributions is shown in figure 11. Each histogram shows the fiducial cross-section for the $\sigma(pp \to \nu\bar{\nu} + \text{jets})$ backgrounds, normalised to bin width.

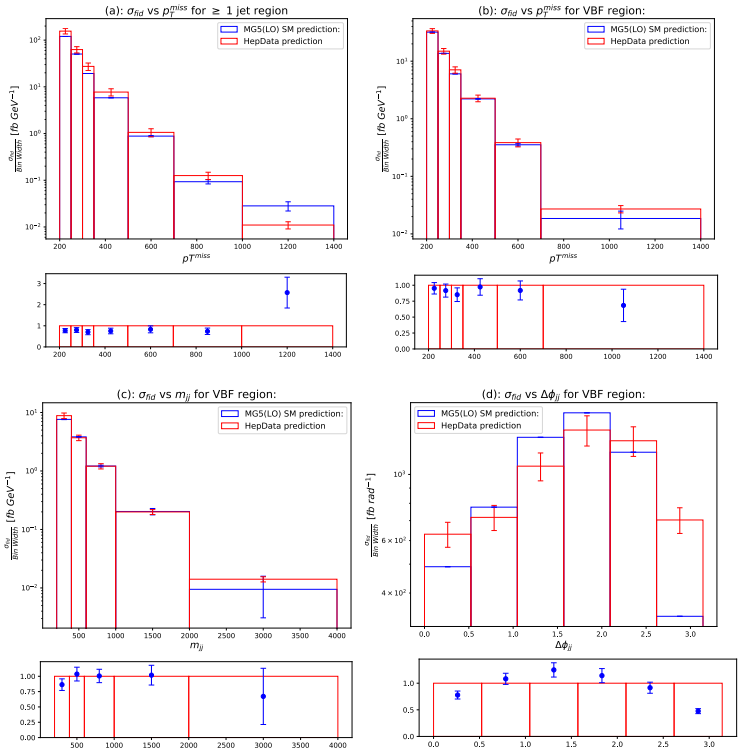


Figure 11: Comparison between HepData standard model MC samples, and own MG5(LO) samples, in the four measured distributions. Plots for residuals are also included.

Two conclusions can be drawn immediately:

1. There is in general good agreement between the two data sets. The major discrepancies occur at the last bins in each of the distributions, as a consequence of the small efficiencies in the MADGRAPH generation. Very few events went into those bins, specially for the QCD $Z \to \nu\bar{\nu}$ which, despite imposing generator cuts, still yielded a poor efficiency. After simulating 100,000 events for each process, the number of events going into the last bins of the distributions are summarised in table 3. The fractional errors for the two backgrounds combined are included.

Distribution	Events into last bin		Combined Fractional Error (%)
	QCD $Z \to \nu\bar{\nu}$	EWK $Z \to \nu\bar{\nu}$	
p_T^{miss} for ≥ 1 jet	3	3	40.8
p_T^{miss} for VBF	3	16	22.9
m_{jj} for VBF	3	63	12.3
$\Delta\phi_{jj}$ for VBF	44	115	7.9

Table 3: SM background events going into last bin for each distribution, after 100,000 events simulated. The combined fractional errors are included.

Therefore, the discrepancy in those bins is justified by the lack of precision.

2. Residuals for the $\Delta\phi_{jj}$ distribution show a clear difference in shapes. The MADGRAPH simulation presents a higher a narrower peak around $\Delta\phi_{jj} = \frac{\pi}{2}$, indicating more events had the two leading jets orthogonal in the momentum - beam axis plane. This is because this variable is very sensible to Lorentz structure of the interaction. At NLO and NNLO, quantum loop corrections flatten out the distribution, as is observed in the HepData SM prediction [19].

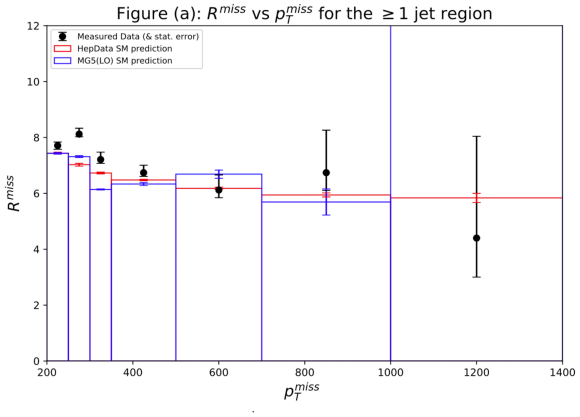
A possible way improve the MADGRAPH predictions, is to reweigh the samples by the ratio of the more optimal HepData total process cross section, to the suboptimal MADGRAPH cross sections. This way, our MC samples become closer to the "best" SM prediction used at ATLAS. For a given distribution, we can define a correction factor as:

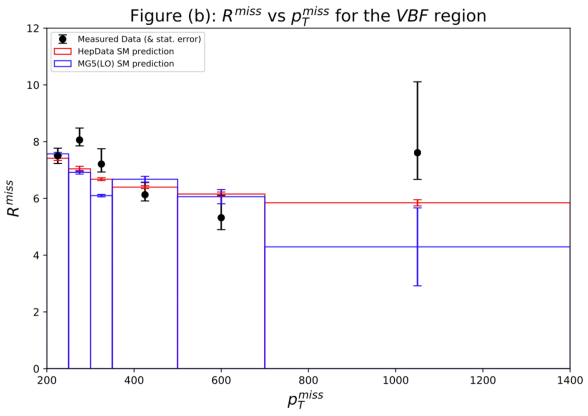
$$\text{Correction} := \frac{\text{HepData total cross section}}{\text{MADGRAPH total cross section}}$$

For the four distributions, the correction factors are summarised in table 4.

Distribution	Total Cross Sections (pb)		Correction factor
	HepData	MADGRAPH	
p_T^{miss} for ≥ 1 jet	13.7	10.6	1.30
p_T^{miss} for VBF	3.22	3.00	1.07
m_{jj} for VBF	3.22	3.00	1.07
$\Delta\phi_{jj}$ for VBF	3.05	3.00	1.02

Table 4: Correction of MADGRAPH sample to match more optimal HepData SM prediction.





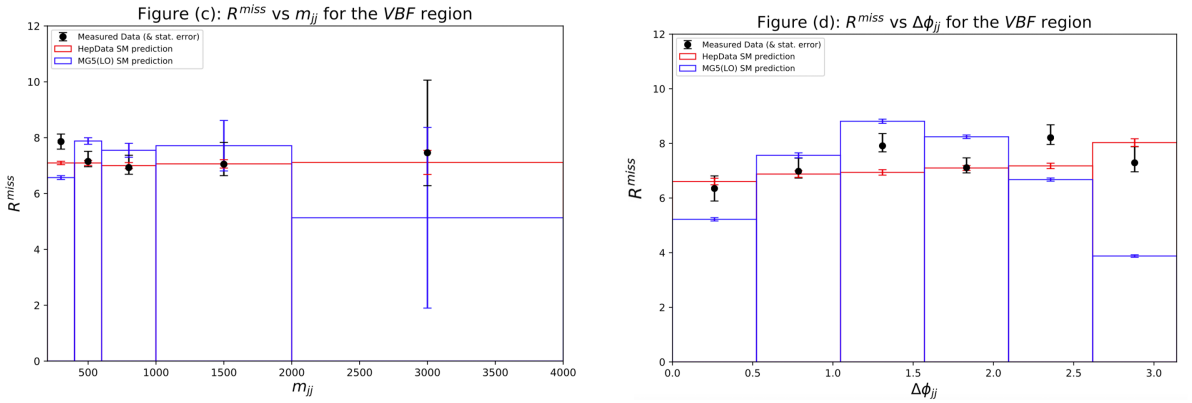


Figure 12: Validating MADGRAPH-generated Monte Carlo samples for the four measured distributions. The more optimal HepData SM backgrounds are overlaid, along with the experimental data.

While this does not account for shape differences, it does slightly improve the expected magnitude of the cross section. Finally, denominator information for $\sigma(l^+l^- + \text{jets})$ can be obtained from HepData tables to construct R^{miss} for the `MADGRAPH` corrected samples. Plots overlaying the two SM predictions, along with the experimental data collected at ATLAS are provided in figure 12.

5 Limit setting:

Having gathered and corrected the Monte Carlo samples, limits can be set using χ^2 test statistics to determine 95% (frequentist) confidence level intervals (CLIs). This is defined as a proposed range of values $[x_-, x_+]$ for a certain model parameter x, such that x is expected to lie in the interval with a certain coverage probability. I.e. For 95% coverage, there is is a 0.95 probability that $x \in [x_-, x_+]$. Confidence regions are hence defined such that:

$$P(x_- \leq x \leq x_+) = \int_{x_-}^{x_+} f(x)dx = \text{Coverage } (= 0.95) \tag{7}$$

Where $f(x)$ is the underlying probability density function for the model parameter x. A frequent choice is the χ^2 distribution, given by:

$$f(z;n) = \frac{1}{2^{n/2}\Gamma\left(\frac{n}{2}\right)} z^{n/2-1} e^{-z/2} \tag{8}$$

where n is the number of degrees of freedom. Most search experiments are commonly interested in the upper limit only, so in these cases, x_- is set to $-\infty$ [20]. The resultant interval is of the form $[-\infty, x_+]$, where x^+ is chosen such that:

$$\int_{-\infty}^{x_+} f(z;n)\ dz = \text{Coverage} \iff \int_{x^+}^{+\infty} f(z;n)\ dz = 1 - \text{Coverage} \tag{9}$$

The confidence-level (p-value) of a χ^2 test statistic can be found from the cumulative distribution [21]:

$$\text{CL (p-value)} = \int_{\chi^2}^{\infty} f(z;n)\ dz = 1 - \int_{-\infty}^{\chi^2} f(z;n)\ dz = 1 - f_{cum}(\chi^2) \tag{10}$$

Where the χ^2 value is defined as:

$$\chi^2 = (\mathbf{x}_{data} - \mathbf{x}_{pred})^T Cov^{-1} (\mathbf{x}_{data} - \mathbf{x}_{pred}) \tag{11}$$

Here, Cov^{-1} is the inverse of the covariance matrix associated to the measured data $\mathbf{x}_{data}$, and $\mathbf{x}_{pred}$ is the SM+DM prediction for the 4 distributions under study. By calculating the CL associated to a sample of x and interpolating, by equation 9 the upper limit x_+ is found at the value of x where the p-value is equal to $1 -$ Coverage.

Furthermore, the CLs method was used in the calculations of the limits. This is a statistical tool used in setting upper limits where, rather than equating the confidence level to the coverage probability, the ratio [22]:

$$\frac{CL_{s+b}}{CL_b} = \text{Coverage}, \tag{12}$$

is equated instead. Here CL_{s+b} is the signal+background confidence level, and CL_b represents the confidence level for the background only hypothesis.

This method lacks frequentist coverage in places of low signal sensitivity, but this it not necessarily disadvantageous: When the measured data fluctuates below the expected background, if the signal is small both the $s+b$ and b hypothesis are disfavoured. This will lead to small p-values for the $s+b$ hypothesis, and hence a possible false exclusion of that model. Therefore, the CL_s method is more conservative in that, it avoids excluding models where the sensitivity is not sufficient to completely rule out that model.

5.1 Covariances:

Covariance matrices encode the various sources of error in the experimental set-up. The diagonal entries represent the variances in a particular vector, and the off-diagonals the correlation:

$$\mathrm{Cov}_{ij} = \sigma_i \sigma_j \mathrm{Corr}_{ij} \tag{13}$$

A distinction can be made between theory covariances, due to uncertainty in the modelling, and statistical covariances, arising due to random fluctuations in the data. For binned Monte Carlo data, one may assume statistical independence between the different bins of the distribution. Consequently, we can write:

$$\mathrm{Corr}_{ij}^{MC} = \delta_{ij} \implies \mathrm{Cov}_{ij}^{\mathrm{MC}} = \delta_{ij}\sigma_i\sigma_j = I\boldsymbol{\sigma}^2 \tag{14}$$

$\mathrm{Cov}^{\mathrm{MC}}$ denotes the statistical covariance due to fluctuations in the MC sample, with a vector of variances $\boldsymbol{\sigma}^2$. For simplicity, theory covariances will be neglected throughout the analysis, since they are hard to predict and require precise knowledge on the BSM model. The different covariances used in the limit setting are discussed in table 5.

Covariance	**Descprition**	**Source**
Measured Covariance	Covariance associated to the measurement of R^{miss} at ATLAS. Quadrature sum of both statistical and systematic uncertainties.	HepData tables
Expected SM Covariance	Expected experimental covariance for running the experiment at ATLAS, under the SM only hypothesis. (Obtained from Monte Carlo estimations). Again, accounts for both statistical fluctuation and systematics.	HepData tables
Theory Covariance	Covariance associated to theory (modelling) in SM and BSM predictions. Set to zero for simplicity.	-
MG5 SM Covariance	Statistical covariance of the MG5@LO MC sample for the SM-only prediction.	MG5 errors for EWK & QCD backgrounds
MG5 DM Covariance	Statistical covariance of the MG5@LO MC sample for the DM-only prediction.	MG5 errors for $pp \to \chi\bar{\chi} + jj$

Table 5: Discussion of covariances considered for the limit setting. MG5@LO statistical covariances were calculated assuming bin independence following equation 14.

5.2 Experimental Systematics:

A brief discussion of the systematics identified in the measurement of R^{miss} at ATLAS is included. For the full description see Ref. [3].

The main source of systematic uncertainties was due to the limited efficiency in isolating and reconstructing the leptonic events in the denominator, which affected the direct measurements but also the background predictions. Furthermore, jet energy scales and resolution uncertainties were also relevant. These were mainly due to differences in the selection criteria for the numerator and denominator events. Further systematics were due to background estimations of multijet and $W \to \tau\nu$ events.

Three main sources of theoretical uncertainties were identified, which included PDF calculations, extrapolations in W background predictions and errors in the cross-section of events involving the top quark quark. Statistical uncertainties were described by Poisson errors in the number of counts at each bin.

5.3 Observed and Expected limits on Λ_{EFT}:

95% CL limits were drawn in the $m_\chi - \Lambda_{EFT}$ plane using both the suboptimal MG5@LO SM prediction and the "better" HepData SM. Two different limits were found:

- **Expected Limit:** Predicts where the limit should be under the SM-only (null) hypothesis.
- **Observed Limit:** Draws limits from ATLAS collected data, using MC predictions for BSM physics.

Monte Carlo events for the $pp \to \chi\bar{\chi} + jj$ signal were produced at a small sample of different m_χ using MG5@LO. The observed and expected limits on Λ_{EFT} for both SM predictions are summarised in table 6.

m_χ [GeV]	**HepData SM prediction**		**MG5@LO SM prediction**	
	Exp. Limit	**Obs. Limit**	**HepData SM**	**MG5 SM**
100	721.1	875.5	711.0	993.5
300	701.5	852.1	691.7	963.5
500	670.2	815.4	660.5	931.75
700	632.9	770.9	623.9	881.8
900	590.7	717.8	581.8	813.5

Table 6: Observed and Expected 95% CLs limits on Λ_{EFT} at different m_χ for MG5@LO SM prediction, and HepData SM for validation. The former accounts the statistical covariances due to the MC. Both limits use MG5@LO predictions for the D7a Dirac dark matter fermion. Everything is in units of GeV.

These contours were plotted in figures 13 and 14. In figure 13, where the HepData SM prediction is used, the limits are consistent with those drawn in previous studies (Ref. [3], Figure 6 (top)). The observed limits drawn using the MG5@LO SM, suggest a larger confidence level interval, favouring the rejection of the null hypothesis. A possible explanation for this is that the MG5 SM prediction is smaller on average compared to the HepData SM, as seen in table 4 and figure 12. Consequently, a higher signal to background ratio is observed which, in the CLs method, yields smaller p-values in regions of higher Λ_{EFT}.

Furthermore, distinguishing between the two SM prediction gives a good visual signature to evaluate how MadGraph performs compared to other generators. As one of the most frequently used generators, these plots may be of some value to future researchers willing to use MG5@LO to set limits. These plots also confirm that accounting for NLO and NNLO effects in the SM prediction pushes back the observed limit.

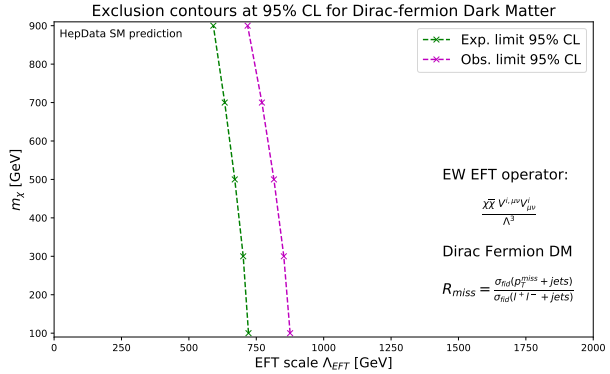


Figure 13: HEP SM Limits

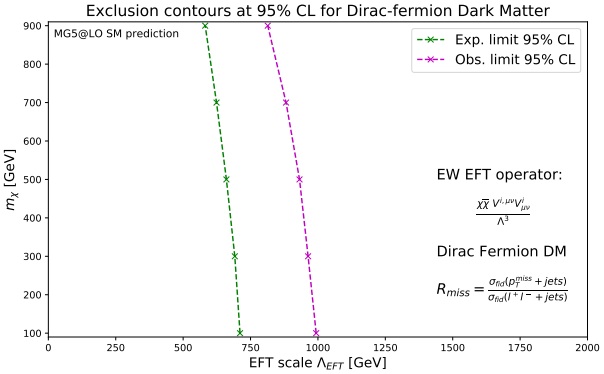


Figure 14: MG5@LO SM Limits

Since all covariance matrices are of the same dimension and contain information about the variance σ^2, quadrature addition of uncertainties is equivalent to simple element-by-element addition of the matrices. In both SM predictions, the measured, theory and MG5 DM covariances were combined for the observed limit, and the expected SM, theory and MG5 DM covariances for the expected limit. In the limits using the MG5@LO SM prediction, the MG5 SM covariance was also included into the error. The covariances used for the limits are shown in table 7.

	HepData SM prediction		**MG5@LO SM prediction**	
	Exp. Limit	**Obs. Limit**	**Exp. Limit**	**Obs. Limit**
Covariances Used	Expected SM MG5 DM Theory	Measured MG5 DM Theory	Expected SM MG5 SM MG5 DM Theory	Measured MG5 SM MG5 DM Theory

Table 7: Covariances used in calculations of the limits for the two SM predictions. All covariances are referred to as in their definition in table 6.

Neglecting the MG5 DM covariance had a small effect on the limits. However, ignoring the MG5 SM covariances in the MG5@LO SM limit calculation gave infinite limits. Recall these errors were rather large due to the small generator efficiencies in some particular bins (outlined in table 3) and must be included in the analysis to find somewhat "reasonable" limits. The python script used to compute the limits is included in appendix 2.

5.4 Limit Discussion:

The above plots show the contours where the p-value is 0.05, which corresponds to a significance Z=1.64. An interesting question one may ask now is what particular bins of the four distributions drive the observed limit. To answer this, two approaches are carried out:

1. Removing each bin individually and hence, the corresponding row & column in the covariance matrices, and recomputing the limit. The difference should tell how much that particular bin contributed to the limit.

2. Disregarding the correlations due to each bin with all the rest, and recomputing the limit. To do so, the off-diagonal entries of the covariances matrices due to each bin will be set to zero. This complements the above point in that, we study if the change in the limits is due to the data point itself and its variance, or if it's due to the correlations with other measurements.

Results are shown in tables 8 and 9.

Effect after bin removal				
Bin Number	p_T^{miss} in ≥ 1 jet	p_T^{miss} in VBF	m_{jj} in VBF	$\Delta\phi_{jj}$ in VBF
1	-0.44	-1.34	-5.12	-4.33
2	-23.0	+0.78	+0.28	-0.45
3	-0.10	+0.29	+0.67	-0.83
4	-0.33	-4.32	+1.54	-1.51
5	+0.11	-12.8	+0.58	-2.41
6	-2.34	+0.57	-	-2.94
7	-67.2	-	-	-

Table 8: Studying the effect on the limits after removing each bin. Bin numbers increase along the distribution, e.g. for p_T^{miss} in ≥ 1 jet, bin 1 is the range 200-250 GeV and bin 7, 1000-1400 GeV.

The collective removal of all the bins which had a positive (green) shift on limit, pushes it from 875.46 $\rightarrow$ 877.51 GeV. Conversely, collectively removing the bins which shift the limit negatively by at least 4 units (highlighted in yellow), push the limit from 875.46 $\rightarrow$ 741.95 GeV, which is very close to the expected limit (721.1 GeV).

Effect after disregarding correlations due to particular bin:				
Bin Number	p_T^{miss} in ≥ 1 jet	p_T^{miss} in VBF	m_{jj} in VBF	$\Delta\phi_{jj}$ in VBF
1	+0.33	-1.36	-1.36	-1.04
2	-10.15	+5.02	+0.22	-0.53
3	+1.21	+0.32	+1.90	+0.63
4	-1.23	-0.76	+3.68	-1.40
5	+4.46	+5.17	+1.01	-0.48
6	-0.38	+0.75	-	-1.23
7	-1.26	-	-	-

Table 9: Impact on the observed limit after removing correlations in the data for each bin in all 4 distributions.

Since all further studies will be carried on a per-bin basis, where all correlations are disregarded completely, we can combine the two effects and see the impact on the observed limit. To do so, we first disregard all correlations by zeroing all of-diagonal entries in the covariances and then remove the data points and variances and see it's effect on the limit.

Combined Effect:				
Bin Number	p_T^{miss} in ≥ 1 jet	p_T^{miss} in VBF	m_{jj} in VBF	$\Delta\phi_{jj}$ in VBF
1	-0.77	0.00	-4.45	-2.59
2	-14.13	-6.01	0.00	-0.08
3	-1.61	-0.53	-0.95	-3.90
4	+0.24	-2.90	-1.79	-0.10
5	-3.90	-15.59	-0.57	-3.14
6	-4.48	-4.05	-	-1.48
7	-59.32	-	-	-

Table 10: Combined effect of disregarding all correlations and then removing each bin, on the observed limit.

From table 10, we see that the p_T^{miss} distribution on the ≥ 1 jet region is the most influential on the limit, specially, the last bin corresponding to the range $p_T^{miss} \in [1000, 1400]$ GeV. Then follow the p_T^{miss}, $\Delta\phi_{jj}$ and m_{jj} in the VBF region, in that order. The more influential bins in each distribution have been highlighted, thus providing an answer to the question: what measurements drive the observed limit?. One can now study if there is any correspondence whatsoever, between this "influence" on the observed limit, and the observed Fisher Information in that bin.

6 Likelihood fits and Fisher information:

Now that all the data is gathered, we wish to deduce certain information about the model parameters m_χ and Λ_{EFT}. A common technique is to make a likelihood fit for the measured sample. Likelihoods are defined as a probability function for the model parameters θ, given a sample of measurements $\boldsymbol{x}$.

In other words, likelihoods address the question: what was the probability to obtain the data set I measured on a certain event, at a certain value of the model parameters. Therefore, it is often denoted by $L(\boldsymbol{\theta} \mid \boldsymbol{x})$. Note that $\boldsymbol{\theta}$ can be a vector when there is more than one model parameter, like in this case $\boldsymbol{\theta} = (m_\chi, \Lambda_{EFT})$. Ratios of likelihood functions are used widely as a frequentist test statistic in hypothesis testing. In fact, the Neyman-Pearson lemma states that for a null hypothesis H_0 with $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ and an alternative H_1 with $\boldsymbol{\theta} = \boldsymbol{\theta}_1$ the likelihood ratio:

$$\frac{L(\boldsymbol{\theta}_0 \mid \boldsymbol{x})}{L(\boldsymbol{\theta}_1 \mid \boldsymbol{x})}$$

is the most powerful test statistic at any given significance level. Power in this context refers to the probability of rejecting the null hypothesis in favour of the alternative [23].

The outcome of the measurements provides estimators for the model parameters. These are said to be unbiased if its expected value is equal to the true value, and it is called biased otherwise. Often, the values of $\boldsymbol{\theta}$ that maximise the likelihood are used in the so-called maximum likelihood estimation (MLE). ML estimators

are not necessarily unbiased, but the discrepancy between the estimator and the true value is minimal for a large sample size [24]. Therefore, if $\boldsymbol{\theta}$ is a MLE, the observed data $\hat{\boldsymbol{x}}$ is the most probable. For biased estimators, the covariance matrix can be used as measure the accuracy of the sample .

Given the likelihood function, the Fisher matrix describes the information content of the measured sample $\boldsymbol{x}$. By weighting the sensitivity at each possible outcome of the distribution, with the probability for that value according to the likelihood function, the matrix finds the overall sensitivity of the distribution to changes in the parameter space. It can be computed by:

$$I_{ij}(\boldsymbol{\theta}) = -E\left[\frac{\partial^2 \log L(\boldsymbol{\theta} \mid \boldsymbol{x})}{\partial\theta_i \partial\theta_j} \;\middle|\; \boldsymbol{\theta}\right] \tag{15}$$

The Crámer-Rao bound then establishes that the minimal uncertainty obtainable in the sample is given by the inverse of the information matrix. This places a lower bound on the entries of the covariance matrix:

$$\mathrm{Cov}_{ij} \geq (I^{-1})_{ij} \tag{16}$$

Therefore, the larger the entries of the information matrix, the larger this lower bound will be, meaning that experimental data can be measured effectively within the model space. One can define the **score** as the gradient of the log-likelihood function. If evaluated at a particular point θ of the parameter space Θ, the score then indicates the sensitivity to changes in that parameter [25]. A score of zero occurs whenever θ equals its true value θ_0, so there is "no information left" about θ in the data sample. The Fisher Information then measures the variance of the score. I.e. A large Fisher information signifies that the score can change largely from sample to sample, so with each measurement you learn more about the parameter space.

For model parameters $\boldsymbol{\theta} = (m_\chi,\ \Lambda_{EFT})$, the Information matrix would be of the form:

$$I(m_\chi,\ \Lambda) = \begin{pmatrix} -\left\langle \frac{\partial^2 \log L}{\partial m_\chi^2} \right\rangle & -\left\langle \frac{\partial^2 \log L}{\partial m_\chi \partial \Lambda} \right\rangle \\ -\left\langle \frac{\partial^2 \log L}{\partial \Lambda \partial m_\chi} \right\rangle & -\left\langle \frac{\partial^2 \log L}{\partial \Lambda^2} \right\rangle \end{pmatrix}, \tag{17}$$

which requires explicit knowledge on how the likelihood function depends on the model parameters. Unfortunately, only the term involving the second derivative in Λ can be done analytically and is outlined in section 6.1 below. Hence, only the information due to the fit to Λ_{EFT} is considered. The terms involving derivatives in m_χ could be computed using numerical gradient methods, and is left open to further study. This would encode the information due to the simultaneous fit of both model parameters.

6.1 Analytic solution for Fisher Information with respect to Λ_{EFT}:

The goal is to find the corresponding entry of the information matrix by computing: $-\left\langle \frac{\partial^2 \log L}{\partial \Lambda^2} \right\rangle$. To this end, a Gaussian approximation is used, where the likelihood is claimed to be of the form:

$$L(\Lambda \mid x) = C \exp\left\{ \frac{-(\mathbf{x} - \mathbf{p}(\Lambda))^2}{2(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \right\},$$

where C is a normalisation constant, $\mathbf{x}$ is the expected experimental outcome and p is the predicted value (for SM+DM) as a function of the model parameter Λ. This approximation holds as a consequence of the central limit theorem in regimes of high statistics. However, this statement will fail in bins with a small number of events, but it is sufficient for our current studies.
The log-likelihood for the above Gaussian is:

$$\log L(\Lambda \mid x) = \log C - \frac{(\mathbf{x} - \mathbf{p}(\Lambda))^2}{2(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)}$$

Therefore, the first derivative of the log-likelihood with respect to Λ follows by the chain rule:

$$\frac{\partial \log L}{\partial \Lambda} = \frac{(\mathbf{x} - \mathbf{p}(\Lambda))}{(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \cdot \left(\frac{\partial p(\Lambda)}{\partial \Lambda} \right). \tag{18}$$

And the second derivative then follows by the product rule:

$$\frac{\partial^2 \log L}{\partial \Lambda^2} = \frac{(\mathbf{x} - \mathbf{p}(\Lambda))}{(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \cdot \left(\frac{\partial^2 p(\Lambda)}{\partial \Lambda^2} \right) - \frac{1}{\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2} \cdot \left(\frac{\partial p(\Lambda)}{\partial \Lambda} \right)^2 \tag{19}$$

Now from equation 5, one uses the fact that:

$$p(\Lambda) = \mathrm{SM} + \frac{\mathrm{DM}^{\Lambda=1}}{\Lambda^6} \tag{20}$$

Where SM is the predicted SM background and $\mathrm{DM}^{\Lambda=1}$ is the dark matter prediction at Λ_{EFT}=1 GeV. The derivatives $\frac{\partial p}{\partial \Lambda}$ and $\frac{\partial^2 p}{\partial \Lambda^2}$ can be computed analytically:

$$\frac{\partial p}{\partial \Lambda} = \frac{-6\ \mathrm{DM}^{\Lambda=1}}{\Lambda^7} \qquad \text{and} \qquad \frac{\partial^2 p}{\partial \Lambda^2} = \frac{42\ \mathrm{DM}^{\Lambda=1}}{\Lambda^8} \tag{21}$$

And hence, substituting into equation 16 gives:

$$\frac{\partial^2 \log L}{\partial \Lambda^2} = \frac{42\ \mathrm{DM}^{\Lambda=1}(\mathbf{x} - \mathbf{p}(\Lambda))}{\Lambda^8(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} - \frac{36\ (\mathrm{DM}^{\Lambda=1})^2}{\Lambda^{14}(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \tag{22}$$

Therefore, the Fisher information with respect to the EFT scale Λ is given by:

$$I = -\left\langle \frac{42\ \mathrm{DM}^{\Lambda=1}(\mathbf{x} - \mathbf{p}(\Lambda))}{\Lambda^8(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} - \frac{36\ (\mathrm{DM}^{\Lambda=1})^2}{\Lambda^{14}(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \right\rangle_{\mathbf{x}}. \tag{23}$$

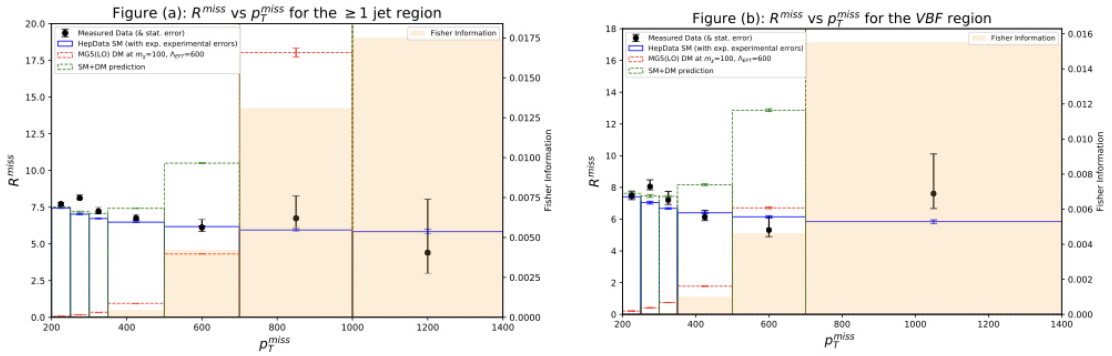
Where the average can be found by throwing toys around the central values of $\mathbf{x}$. It should be noted that the uncertainty in the predicted value σ_p, can also be treated as a function of Λ by:

$$\sigma_p(\Lambda) = \sqrt{\sigma_{SM}^2 + \left(\frac{\sigma_{DM^{\Lambda=1}}}{\Lambda^6}\right)^2} \tag{24}$$

However, this effect was small and was neglected for simplicity.

6.2 Observed Information

From equation 23, it can be noted how, at large Λ, the first term in the sum dominates and $I \propto -\langle x - p(\Lambda) \rangle$. Hence, it follows that the larger the discrepancy between the measured data and the SM+DM hypothesis (within experimental errors), the larger the information. This is seen in figure 15, which finds the measured Fisher Information due to Λ_{EFT} for a particular D7a dark matter model at $m_\chi = 100$ GeV and $\Lambda_{EFT} = 600$ GeV. Unfortunately, no direct correspondence was identified with the results from section 5.4. for the most "influential" bins on the CLs limit. In the regions of phase space with large FI, the data is more sensitive to changes in the model parameter Λ_{EFT}. Furthermore, the Crámer-Rao bound places a smaller lower bound on the uncertainties achievable in any estimator of Λ_{EFT}.



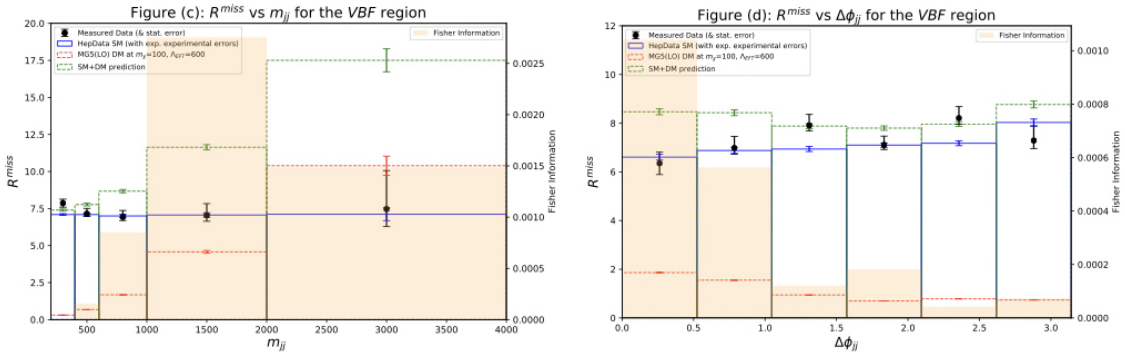


Figure 15: Measured Fisher Information with respect to Λ_{EFT}, using the ATLAS measured data and the HepData SM prediction. A Gaussian likelihood fit was made to describe the samples. The FI was found by equation 23, and the average was computed after throwing 5,000 toys drawn from a normal distribution $\sim N(x, \sigma_x)$. As this is a per-bin analysis, σ_x and σ_p were obtained from the diagonal entries in the respective covariance matrices.

The information is additive and hence, the total information due to a single distribution can be found by summing over all bins. The total information due to figures (a), (b), (c) and (d) was 35.2×10^{-3}, 21.2×10^{-3}, 5.23×10^{-3} and 2.02×10^{-3} respectively. This confirms that p_T^{miss} is the preferred variable for this search. Furthermore, it can be noted that the in this particular model, the higher the p_T^{miss}, the higher the observed information.

This is because the DM prediction grows relatively fast towards the end of the p_T^{miss} spectrum compared to the rather flat shape of the SM-only curve. Wherever the discrepancy between the measured data and the BSM prediction is large (weighted by experimental errors), the more information is contained. In contrast, the DM prediction for $\Delta\phi_{jj}$ does not depart from the SM hypothesis sufficiently to yield a high information. In this distribution, the regions with small $\Delta\phi_{jj}$ provide the most information. Furthermore, for m_{jj}, the experimental uncertainties in the last bin prevent it from having a large information, and it is in fact the second to last bin that contributes the most.

Decreasing θ in either direction of the parameter space increases the production for dark matter in the D7a model. Hence, due to the proportionality $I \propto -\langle x - p(\Lambda)\rangle$, at small m_χ and/or Λ_{EFT} the prediction p increases and consequently, so does the FI . This is shown in Table 11, where the Fisher Information for the p_T^{miss} distribution in the ≥ 1 jet region is provided at different points in parameter space.

		m_χ		
		100	500	900
Λ_{EFT}	400	3967	2410	812
	600	35.2	14.9	3.27
	800	0.613	0.25	0.05

Table 11: Fisher Information ($\times10^{-3}$) of the p_T^{miss} distribution at ≥ 1 jet space, at different regions of parameter space. All parameters are in GeV. Similar tables of total FI for the remaining distributions are provided in appendix 5.

6.3 Expected Information:

At this point, the Madgraph simulations at leading order will be used to study further variables like rapidity, leading jet p_T and subleading jet p_T in the different regions of phase space. Since these have not been measured at ATLAS, the expected information will be computed as predicted by the SM. If you recall equation 23 for the information due to the EFT scale:

$$I = -\left\langle \frac{42\ \mathrm{DM}^{\Lambda=1}(\boldsymbol{x} - \mathbf{p}(\Lambda))}{\Lambda^8(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} - \frac{36\ (\mathrm{DM}^{\Lambda=1})^2}{\Lambda^{14}(\boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_p^2)} \right\rangle_{\boldsymbol{x}},$$

where previously $\boldsymbol{x}$ was the measured data. This time however, the MadGraph SM prediction will be used instead. This will give a feel for how much information the SM predicts there to be for each of the

distributions. This conclusion will be valuable to determining if there are any other variables that may be interesting, aside from the four originally measured.

There is however, one limitation in that, the expected experimental covariances are unknown, so the value for σ_x has to be chosen arbitrarily. However, the aim of this study is to compare the information content of different potential experiments, so as long as they are all under the same framework, however limited this may be, adequate conclusions can be drawn. Therefore the MadGraph Monte Carlo errors will be used for the SM expectation. Also, since only the numerator of R^{miss} was simulated on MadGraph, that is, the processes corresponding to $\sigma(pp \to p_T^{miss} + \text{jets})$, these are the values upon which the information will be calculated. All events were weighted by the average of the correction factors from table 4.

The results for a DM prediction at $m_\chi = 100$ GeV, $\Lambda_{EFT} = 600$ GeV are summarised in table 12. A few distributions are included in figure 16. For all of the distributions, see appendix 4.

Variable	Phase Space Region	Total Differential Cross Section (fb)	Total Expected FI ($\times 10^{-5}$)	Region of Maximum FI
p_T^{miss}	≥ 1 jet	11,793	65.3	[700, 1000] GeV
	VBF	3,342	46.1	[700, 1400] GeV
m_{jj}	VBF	3,342	27.9	[1000, 2000] GeV
$\Delta\phi_{jj}$	VBF	3,342	17.3	$[0, \frac{\pi}{6}]$
Leading jet η	≥ 1 jet	11,793	23.18	[-0.5, +0.5]
	VBF	3,342	18.41	[-0.5, +0.5]
Leading jet p_T	≥ 1 jet	7,199	49.6	[600, 1000] GeV
	VBF	2,010	27.1	[600, 1000] GeV
Subleading jet p_T	VBF	3.333	16.5	[200, 300] GeV

Table 12: Expected information under MG5@LO SM hypothesis, for EFT DM production at m_χ : 100 GeV, Λ_{EFT}: 600 GeV. The processes under study are $pp \to p_T^{miss}$+jets, which have been weighted with an average correction factor based on NLO and NNLO corrections.

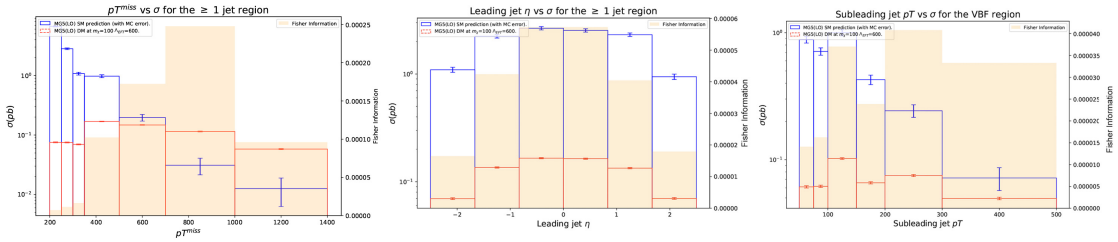


Figure 16: A few of the distributions for the expected FI. For all 9, see appendix 4.

6.4 Discussion

Table 12 proposes 5 new variables for the search, and predicts the regions of higher sensitivity according to their Fisher Information. We can see again how the distributions of p_T^{miss} contain the most total information, confirming this is the preferred variable for these searches.

With this intuition in mind, the p_T of the leading and subleading jets individually was studied. This was quite fruitful for the leading jet, where both regions combined gave a total information of 74.0×10^{-5}, most of which came from the last bin. In contrast, the information due the subleading jet studies was the smallest of all. However, the regions of highest FI were [100, 150] and [200, 300] GeV. Therefore, this distribution could be used on operations at lower energies.

For m_{jj}, the information content peaks at the second to last bin. For $m_{jj} \geq 2000$, the DM and expected SM production are in fact quite low, giving small sensitivity. In addition, the newly proposed leading jet rapidity gave a quite substantial total information in both regions combined of 41.6×10^{-5}. It comes as no surprise that the events with $\eta \sim 0$, encode the maximum information, since these corresponds to the

leading jet flying along the transverse direction.

Finally the distribution of $\Delta\phi_{jj}$, one of the four originally measured at ATLAS, gave a rather poor expected information. Based on FI alone, we conclude this distribution is not the most optimal for this search. There may be however, other reasons why measuring it may be of interest, since it's shape can vary a lot for different BSM models, and could help distinguishing them.

In addition, one can note how in general, there is a good agreement in the shapes of the measured and expected information for the 4 measured distributions. The only region of major discrepancy is in the last bin of the p_T^{miss} distribution for ≥ 1 jet, where the experimental uncertainties were in fact smaller than the MadGraph prediction, due to the limited efficiency of the generator in that region. This means that the expected information was in fact smaller than what was actually measured. However, we can conclude that in this case, the MadGraph SM prediction encodes quite accurately the information content measured at ATLAS.

7 Conclusion

In this project, we provide a first tool to estimate the expected experimental sensitivity by means of the Fisher Information, which finds a lower bound for any estimator of the EFT scale Λ_{EFT}. This is provided by means of a python3 code attached in appendix 3. 5 new distributions are tested in the different regions of phase space, on top of the 4 originally measured at ATLAS in 2017. These studies confirm that p_T is the preferred variable for this search, as it encodes the most information. Further knowledge on the model parameter can be gained by looking at the leading jet p_T^{miss} alone, and it's pseudorapidity distribution. We also argue that judging on FI alone, $\Delta\phi_{jj}$ is not the most optimal variable for this search.

Furthermore, 95% CL limits on the parameter space were provided which match that of previous research. By individually removing each data point of the distribution, conclusions were drawn on the influence each bin has on the observed limit. An interesting analysis we open to further study is to examine the relationship, if any, between the measured information and the influence on the limits. This would further explore the uses of FI in hypothesis testing. Samples generated by MG5@LO were also used to compute limits, thus providing an estimation of the effect of NLO and NNLO corrections. Since MadGraph is one of the most popular multipurpose generators to date, this may be to some value for future research which use this tool to gather MC samples.

Finally, we outline the limitations of our information calculations, and propose ways in which these can be overcome:

1. A likelihood fit on only one of the two model parameters was found. The information due to the WIMP mediator mass m_χ was not explored, since it could not be done analytically. Numerical gradient methods could be used to estimate the derivatives of the log-likelihood with respect to m_χ, and hence, span the entire parameter space.

2. A Gaussian likelihood approximation was used to fit the MC samples. While this was sufficient for these studies, a more accurate fit could be used alternatively. One possibility is to use MINUIT, a numerical tool by CERN used for likelihood estimation [26].

3. In doing a per-bin analysis on one of the parameters only, we fail to account for the correlations between bins of the samples. This should however be corrected if the two above limitations are mitigated.

In doing so, a valuable framework for experiment designing can be established, whose uses go far beyond the scope of this search, and could be similarly implemented across many areas of high energy physics research.

8 References:

[1] Presentation of search results: the CLs technique. (2002). Journal of physics G: Nuclear and particle physics. https://indico.cern.ch/event/398949/attachments/799330/1095613/The_CLs_Technique.pdf [Accessed 12 Dec. 2019].

[2] Brehmer, J., Cranmer, K. and Kling, F. (2017). Better Higgs boson measurements through information geometry. Physical Review D, 95(7). https://arxiv.org/abs/1612.05261 [Accessed 26 Sep. 2019].

[3] ATLAS Collaboration (2017). Measurement of detector-corrected observables sensitive to the anomalous production of events with jets and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector. https://arxiv.org/abs/1707.03263 [Accessed 28 Sep. 2019].

[4] Baudis, L. (2006). Dark matter searches. International Journal of Modern Physics A, 21(08n09). https://arxiv.org/abs/1801.08128v1 [Accessed 29 Dec. 2019].

[5] Catena, R. and Covi, L. (2014). SUSY dark matter(s). The European Physical Journal C, 74(5). https://arxiv.org/abs/1310.4776 [Accessed 29 Oct. 2019].

[6] The ATLAS collaboration (2017). Measurement of detector-corrected observables sensitive to the anomalous production of events with jets and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector. Hepdata. https://www.hepdata.net/record/ins1609448 [Accessed 9 Nov. 2019]

[7] ATLAS (n.d.). The Inner Detector. ATLAS Experiment at CERN. https://atlas.cern/discover/detector/inner-detector [Accessed 26 Dec. 2019].

[8] Puzo, P. (2002). ATLAS Calorimetry. Nuclear Instruments and Methods in Physics Research, A 494, pp.340–345.

[9] Rubbo, F. (2016). VBF $H \rightarrow bb$ (& more). Indico.cern.ch. https://indico.cern.ch/event/540653/contributions/2196046/attachments/1290433/1921606/vbfhbb_Jamboree2016_theorysession.pdf [Accessed 23 Dec. 2019].

[10] Srimanobhas, N. (2010). Introduction to Monte Carlo for Particle Physics Study. Indico.cern.ch. https://indico.cern.ch/event/92209/contributions/2114409/attachments/1098701/1567290/CST2010-MC.pdf [Accessed 14 Dec. 2019].

[11] Alwall, J. (2007). MadEvent, Minimal user guide. https://cp3.irmp.ucl.ac.be/projects/madgraph/attachment/wiki/ManualAndHelp/Manual-March-2007.pdf [Accessed 12 Oct. 2019].

[12] Kumericki, K. (2016). Feynman Diagrams for Begginers. Department of Physics, Faculty of Science, University of Zagreb, Croatia. https://arxiv.org/abs/1602.04182 [Accessed 24 Oct. 2019].

[13] PYTHIA8 (2019). Online manual. http://home.thep.lu.se/~torbjorn/pythia82html/Welcome.html [Accessed 31 Oct. 2019].

[14] Tsiskaridze, S. (2017). Search for flavor-changing neutral current top quark decays. Barcelona: SPRINGER INTERNATIONAL PU, Chap 3, p. 64.

[15] Cotta, R., Hewett, J. and Le, M. (2013). Bounds on dark matter interactions with electroweak gauge bosons. Physical Review D, 88(11). https://arxiv.org/abs/1210.0525 [Accessed 20 Dec. 2019].

[16] Introduction to Effective Field Theories. (2017). Lecture Notes, Les Houches Summer School. https://arxiv.org/abs/1804.05863 [Accessed 29 Dec. 2019].

[17] Buckley, A., Butterworth, J. and Grellscheid, D. (2013). Rivet user manual. Computer Physics Communications, 184(12). https://arxiv.org/abs/1003.0694 [Accessed 19 Oct. 2019].

[18] Gutschow, C. (2017). ATLAS_2017_I1609448. Rivet analyses reference – Hepforge. https://rivet.hepforge.org/analyses/ATLAS_2017_I1609448 [Accessed 30 Sep. 2019].

[19] Behring, A., Czakon, M. and Mitov, A. (2019). Higher Order Corrections to Spin Correlations in Top Quark Pair Production at the LHC. Physical Review Letters, 123(8). https://arxiv.org/abs/1901.05407 [Accessed 25 Dec. 2019].

[20] Barlow, R. (2019). Practical Statistics for Particle Physics. Lecture Notes, The University of Huddersfield. https://arxiv.org/abs/1905.12362 [Accessed 29 Sep. 2019].

[21] Cowan, G. (1998). Statistical data analysis. 1st ed. Oxford University press, p.62.

[22] Eilam, G. (2020). LHC Statistics for Pedestrians. Weizmann Institute, Rehovot, Israel. https://cds.cern.ch/record/1099994/files/p205.pdf [Accessed 17 Dec. 2019].

[23] Cowan, G., Cranmer, K., Gross, E. and Vitells, O. (2011). Asymptotic formulae for likelihood-based tests of new physics. The European Physical Journal C, 71(2). https://arxiv.org/abs/1007.1727 [Accessed 20 Dec. 2019].

[24] Blocker, C. (2004). Likelihood Fits. Brandeis University. http://physics.bu.edu/neppsr/2004/Talks/Likelihood-Blocker.pdf [Accessed 24 Nov. 2019].

[25] Ly, A., Marsman, M., Verhagen, J., Grasman, R. and Wagenmakers, E. (2017). A Tutorial on Fisher information. Journal of Mathematical Psychology, [online] 80, pp.40-55. https://arxiv.org/abs/1705.01064 [Accessed 12 Dec. 2019].

[26] CERN. (2008). Minuit. https://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/ [Accessed 1 Jan. 2020].

9 Appendices:

I. Rivet Routines (v.2.7.0) used in the analysis:

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/DressedLeptons.hh"
#include "Rivet/Projections/VetoedFinalState.hh"
#include "Rivet/Projections/PromptFinalState.hh"
#include "Rivet/Projections/MissingMomentum.hh"
#include <vector>
#include <cmath>

namespace Rivet {
  // Define classes and constructor:
  class pass_mono :

  public Analysis {
  public:
    /// Constructor
    pass_mono(const string name="pass_mono",
                        const string ref_data="pass_mono") : Analysis(name) {
      setRefDataName(ref_data);
    }
/// Initialize
    void init() {

      // Get options from the new option system
      _mode = 0;
      if ( getOption("LMODE") == "NU" ) _mode = 0; // using Z -> nunu channel by
    default

// Work in detector range eta=+- 4.9. (abolute value pseudorap. (abseta) < 4.9)
      // Prompt photons. Declare final state photons
      PromptFinalState photon_fs(Cuts::abspid == PID::PHOTON && Cuts::abseta <
    4.9);
      // Prompt electrons. Declare final state electrons.
      PromptFinalState el_fs(Cuts::abseta < 4.9 && Cuts::abspid == PID::ELECTRON);
      // Prompt muons. Declare final state muons.
      PromptFinalState mu_fs(Cuts::abseta < 4.9 && Cuts::abspid == PID::MUON);

      // Dressed leptons. Leptonic cut. Select leptons with pT> 7 GeV and detector
     range eta=+- 2.5
      Cut lep_cuts = Cuts::pT > 7*GeV && Cuts::abseta < 2.5;
      DressedLeptons dressed_leps(photon_fs, (_mode == 2 ? el_fs : mu_fs), 0.1,
    lep_cuts);
      declare(dressed_leps, "DressedLeptons");

      // In-acceptance leptons for lepton veto
      PromptFinalState veto_lep_fs(Cuts::abseta < 4.9 && (Cuts::abspid == PID::
    ELECTRON || Cuts::abspid == PID::MUON));
      veto_lep_fs.acceptTauDecays();
      veto_lep_fs.acceptMuonDecays();
      DressedLeptons veto_lep(photon_fs, veto_lep_fs, 0.1, lep_cuts);
      declare(veto_lep, "VetoLeptons");

      // MET
      VetoedFinalState met_fs(Cuts::abseta > 2.5 && Cuts::abspid == PID::MUON); //
     veto out-of-acceptance muons
      if (_mode) met_fs.addVetoOnThisFinalState(dressed_leps);
```

```
53        declare(MissingMomentum(met_fs), "MET");
54
55        // Jet collection
56        FastJets jets(FinalState(Cuts::abseta < 4.9), FastJets::ANTIKT, 0.4, JetAlg
      ::NO_MUONS, JetAlg::NO_INVISIBLES);
57        declare(jets, "Jets");
58          MSG_INFO("data: event,weight,njets,met,jpt1,jeta1,jpt2,jpt3,mjj,dphi,
      pass_vbf,pass_mono");
59      }
60
61      bool isBetweenJets(const Jet& probe, const Jet& boundary1, const Jet&
      boundary2) {
62        const double y_p = probe.rapidity();
63        const double y_b1 = boundary1.rapidity();
64        const double y_b2 = boundary2.rapidity();
65        const double y_min = std::min(y_b1, y_b2);
66        const double y_max = std::max(y_b1, y_b2);
67        return (y_p > y_min && y_p < y_max);
68      }
69
70      int centralJetVeto(Jets& jets) {
71        if (jets.size() < 2) return 0;
72        const Jet bj1 = jets.at(0);
73        const Jet bj2 = jets.at(1);
74
75        // Start loop at the 3rd hardest pT jet
76        int n_between = 0;
77        for (size_t i = 2; i < jets.size(); ++i) {
78          const Jet j = jets.at(i);
79          if (isBetweenJets(j, bj1, bj2) && j.pT() > 25*GeV)  ++n_between;
80        }
81        return n_between;
82      }
83
84        double sum_weight=0.0;
85        double sur_events=0.0;
86
87      /// Perform the per-event analysis
88      void analyze(const Rivet::Event& event) {
89
90      std::cout << "Starting event " << numEvents() << std::endl ;
91      const double weight = event.weight(); // SoW counter
92
93      sum_weight += weight;
94
95        // Require 0 (Znunu) or 2 (Zll) dressed leptons
96        bool isZll = bool(_mode);
97        const vector<DressedLepton> &vetoLeptons = applyProjection<DressedLeptons>(
      event, "VetoLeptons").dressedLeptons();
98        const vector<DressedLepton> &all_leps = applyProjection<DressedLeptons>(
      event, "DressedLeptons").dressedLeptons();
99          if (!isZll && vetoLeptons.size())    vetoEvent;
100         if ( isZll && all_leps.size() != 2)  vetoEvent;
101         //isZll = 0 for all events. (Should be in mode 0)
102
103       // Get jets:
104         Jets jets = applyProjection<FastJets>(event, "Jets").jetsByPt(Cuts::pT >
      25*GeV && Cuts::absrap < 4.4);
105
106       const size_t njets = jets.size();
107
108         FourMomentum PtMiss;
109         for (unsigned i=0; i<njets; i++) {
110             PtMiss+= jets[i].momentum();
111         }
```

```
112         double etmiss=PtMiss.pT();
113
114         if (!njets)  vetoEvent;
115       const int njets_gap = centralJetVeto(jets);
116
117       double jpt1 = jets[0].pT(); // transverse momentum for 1st jet.
118       double jeta1 = jets[0].eta(); // preseudorapidity for 1st jet.
119       double mjj = 0., jpt2 = 0., jpt3=0., dphijj = 0.;
120
121       if (njets >= 2) {
122         mjj = (jets[0].momentum() + jets[1].momentum()).mass();
123         jpt2 = jets[1].pT();
124         dphijj = deltaPhi(jets[0], jets[1]);
125       }
126       if (njets >=3) {jpt3 = jets[2].pT();}
127
128       // MET
129       Vector3 met_vec = apply<MissingMomentum>(event, "MET").vectorMPT();
130       double met = met_vec.mod();
131
132 // Cut on deltaPhi between MET and first 4 jets, but only if jet pT > 30 GeV
133       bool dphi_fail = false;
134       for (size_t i = 0; i < jets.size() && i < 4; ++i) {
135         dphi_fail |= (deltaPhi(jets[i], met_vec) < 0.4 && jets[i].pT() > 30*GeV);
136       }
137
138      // Cuts & Event Selection:
139       const bool pass_met_dphi = met > 200*GeV && !dphi_fail;
140       const bool pass_vbf = pass_met_dphi && mjj > 200*GeV && jpt1 > 80*GeV &&
      jpt2 > 50*GeV && njets >= 2 && !njets_gap;
141       const bool pass_mono = pass_met_dphi && jpt1 > 120*GeV && fabs(jeta1) < 2.4;
142
143         if (pass_mono) {
144             sur_events+=1;
145         }
146
147     if(pass_mono)
148     // Or alternatively, if pass_vbf, depending on what region of phase space we
      are considering.
149     // Print out a few variables.
150     {MSG_INFO("data: " << numEvents() << ", " << weight << ", " << njets << ", "<<
       met << ", " << jpt1 << ", " << jeta1 << ", " << jpt2 << ", " << mjj << ", " <<
       dphijj << ", " << pass_vbf << ", " << pass_mono);}
151     }
152
153     /// Normalise, scale and otherwise manipulate histograms here
154       void finalize() {
155           std::cout << "Sum of weights: " << sum_weight << std::endl;
156           std::cout << "Events surviving: " << sur_events << std::endl;
157       const double sf(crossSection() / femtobarn / sumOfWeights());
158
159         MSG_INFO(sf);
160       }
161   protected:
162     // Analysis-mode switch
163     size_t _mode;
164   };
165   // Hooks for the plugin system
166 DECLARE_RIVET_PLUGIN(pass_mono);
167 }
```

II. PYTHON3 script for calculating limits:

```
1 # Get CL limits:
2 import sys
3 import matplotlib.pyplot as plt
```

```
4  import numpy              as np
5  import scipy.stats        as stats
6  import pickle
7  #  ====================================================================
8  #      GLOBAL: READ DATA, SM AND BSM PREDICTIONS:
9  #  ====================================================================
10 inputs = pickle.load(open("paper_values_and_DMEFT_D7a_m10_l400_prediction.pickle",
       "rb"))
11
12 meas         = inputs["meas_values"]   #  This is the measurement (central values)
13 meas_cov     = inputs["meas_cov"]      #  This is the measurement (covariance)
14 meas_cov_inv = np.linalg.inv(meas_cov) #  This is the inverse covariance
15 n_dof        = len(meas)               #  This is the number of bins (=24)
16
17 SM           = inputs["SM_model_values"]  #  This is the SM expectation (central
       values)
18 SM_exp_cov   = inputs["SM_exp_cov"]       #  This is the expected experimental
       covariance
19 SM_thry_cov  = inputs["SM_model_cov"]     #  This is the covariance on the central
       values due to theory modelling
20 SM_thry_cov  = np.zeros(shape=(n_dof, n_dof))               # Uncomment this line
       if you want to ignore theory uncertainty on SM
21 SM_total_cov = SM_exp_cov + SM_thry_cov           #  This is the sum of the theory
       covariance with the expected experimental covariance
22 SM_exp_cov_inv   = np.linalg.inv(SM_exp_cov)
23 SM_total_cov_inv = np.linalg.inv(SM_total_cov)
24
25 # MadGraph Generated Data goes here:
26 BSM_400     = inputs["BSM400_values"]             #  This is Ste's MG5 (LO)
       expectation for DMEFT D7a: m_chi = 10 GeV, lambda = 400 GeV
27 BSM_400_cov = inputs["BSM400_cov"]                #  This is the covariance on the
       expectation
28 BSM_400_cov = np.zeros(shape=(n_dof, n_dof))
29
30 #  ============================================================
31 #         LOAD OWN MG5 DATA (PICKLE):
32 #  ============================================================
33 with open('MG_Data_m100.pickle', 'rb') as f:
34     inputs = pickle.load(f)
35
36 mchi         = inputs[0]    # mchi used
37 lda          = inputs[1]    # Lambda used
38 Rmiss_SM     = inputs[2]    # MG5(LO) Rmiss for SM prediction
39 Rmiss_DM     = inputs[3]    # MG5(LO) Rmiss for DM only prediction.
40 cov_stat_SM = inputs[4]    # Statistical Covariance Matrix for SM prediction
41 cov_stat_DM = inputs[5]    # Statistical Covariance Matrix for DM prediction
42
43 Rmiss_SM=np.asarray(Rmiss_SM)
44 Rmiss_DM=np.asarray(Rmiss_DM)
45 Rmiss_BSM=Rmiss_SM+Rmiss_DM
46
47
48 cov_SM=cov_stat_SM+SM_exp_cov;
49 #  Experimental covariance (cov_stat_SM) + theory covariance (MC modelling)
50
51 SM_thry   =[];
52 SM_thry = np.zeros(shape=(n_dof, n_dof))
53
54 if True:
55 # TESTS FOR SM THEORY ERRORS:
56     k = [];
57     k[:]=[0.005 for i in range (24)]
58     SM_thry   = np.diag(k)
59
60 cov_DM=cov_stat_DM;
```

```
61 MG_SM_cov_thry=cov_stat_SM
62 # Theory covariances (from MC modelling). These are the statistical covariances
63 # from the MG simulution. (Diagonal matrices of variances.)
64
65 # Set this to true to print comparison HepData-MG simulated data.
66 data_compare=False;
67
68 if data_compare:
69     print('Mass used:', mchi)
70     print('SM HepData prediction:')
71     print('----------------------')
72     print(SM)
73     print('')
74     print('SM MG(LO) Data prediction:')
75     print('-------------------------')
76     print(np.around(Rmiss_SM, decimals=5))
77     print('')
78     print("HepData EFT DM prediction at mass",str(masses[3]),'lambda 400:')
79     print('-----------------------------------------------')
80     print(BSM_400_allmasses[3])
81     print('')
82     print('Madgraph EFT DM at mass ', mchi,' lambda ', lda,':')
83     print('-----------------------------------------------')
84     print(np.around(SM+Rmiss_DM, decimals=6))
85     print('')
86
87 #  ============================================================================
88 #      REQUIRED FUNCTIONS:
89 #  ============================================================================
90 #  Calculate chi2 between two distributions with a specified covariance:
91 def get_chi2 (meas, pred, cov) :
92   cov_inv = np.linalg.inv(cov) # Computes covariance matrix inverse.
93   res = meas - pred;
94   return np.matmul(res, np.matmul(cov_inv, res))
95
96 #  Calculate the confidence level (p-value) of a chi2 test-statistic assuming that
       it follows the usual distribution
97 def get_frequentist_CL (chi2) :
98   return 1.0-stats.chi2.cdf(chi2, n_dof) #cdf: Cummulative Distribution Function
99 # Stats paper: Practical stats for particle physics, Barlow: p. 40, point 4: Upper
       Limit.
100
101 #  Calculate the CLs limit associated with a given measurement
102 #  - method: profile lambda, create a BSM prediction at each lambda value, add it
      to the SM and calculate CLs. Return value of lambda where profile crosses (1-
      coverage)
103
104 def get_CLs_limit_MG (this_meas, this_cov, coverage=0.95, pred='MG'):
105     if coverage <= 0 or coverage >= 1 : raise ValueError("get_CLs_limit():
      provided coverage {coverage} is out of the allowed range [0, 1]")
106
107     lambda_linspace=np.linspace(500, 1000, 51);
108     CLs=[];
109     if pred=='MG':
110
111         for this_lambda in lambda_linspace:
112             scale_factor=(400./this_lambda) ** 6;
113             RmissDM = Rmiss_DM * scale_factor;
114             cov_DM=cov_stat_DM * scale_factor * scale_factor;
115             chi2_DM= get_chi2(this_meas, Rmiss_SM + RmissDM, this_cov+ cov_DM +
      MG_SM_cov_thry);
116             chi2_SM= get_chi2(this_meas, Rmiss_SM, this_cov + MG_SM_cov_thry)
117             CLs.append( get_frequentist_CL(chi2_DM) / get_frequentist_CL(chi2_SM)
      );
118         this_CLs=np.interp([1.0 - coverage], CLs, lambda_linspace)[0];
```

```
119         return(this_CLs)
120
121     elif pred =='HEP':
122         for this_lambda in lambda_linspace:
123
124             scale_factor=(400./this_lambda) ** 6;
125             RmissDM = Rmiss_DM * scale_factor;
126             cov_DM=cov_stat_DM * scale_factor * scale_factor;
127
128             chi2_DM= get_chi2(this_meas, SM + RmissDM, this_cov+ cov_DM);
129             chi2_SM= get_chi2(this_meas, SM, this_cov)
130
131             CLs.append( get_frequentist_CL(chi2_DM) / get_frequentist_CL(chi2_SM)
    );
132         this_CLs=np.interp([1.0 - coverage], CLs, lambda_linspace)[0];
133         return(this_CLs)
134
135 def remove_bin(plot, n): # nth bin in specific plot (starting at 0th bin)
136     n=n-1
137     if plot=='b':
138         n+=7
139     if plot=='c':
140         n+=13
141     if plot=='d':
142         n+=18
143
144     global cov_stat_SM, cov_stat_DM, MG_SM_cov_thry, meas_cov, SM_exp_cov,
    Rmiss_DM, Rmiss_SM, SM, meas;
145
146     # Remove corresponding row & column from covariance matrices:
147     cov_stat_SM=np.delete(cov_stat_SM, n, axis=0)
148     cov_stat_SM=np.delete(cov_stat_SM, n, axis=1)
149
150     cov_stat_DM=np.delete(cov_stat_DM, n, axis=0)
151     cov_stat_DM=np.delete(cov_stat_DM, n, axis=1)
152
153     MG_SM_cov_thry=np.delete(MG_SM_cov_thry, n, axis=0)
154     MG_SM_cov_thry=np.delete(MG_SM_cov_thry, n, axis=1)
155
156     meas_cov=np.delete(meas_cov, n, axis=0)
157     meas_cov=np.delete(meas_cov, n, axis=1)
158
159     SM_exp_cov=np.delete(SM_exp_cov, n, axis=0)
160     SM_exp_cov=np.delete(SM_exp_cov, n, axis=1)
161
162     # Remove entries in data set:
163     Rmiss_DM=np.delete(Rmiss_DM,n)
164     Rmiss_SM=np.delete(Rmiss_SM,n)
165     SM=np.delete(SM,n)
166     meas=np.delete(meas,n)
167
168     if plot=='a':
169         print('Removed bin '+str(n+1)+' from plot a')
170     if plot=='b':
171         print('Removed bin '+str(n+1-7)+' from plot b')
172     if plot=='c':
173         print('Removed bin '+str(n+1-13)+' from plot c')
174     if plot=='d':
175         print('Removed bin '+str(n+1-18)+' from plot d')
176
177 # Diagonalise covariance matrices: Remove correlations from other bins.
178 def remove_corr(plot, n):
179     # Meas Cov and SM_exp_cov... But really only need meas cov.
180     n=n-1
181     if plot=='b':
```

```
182             n+=7
183         if plot=='c':
184             n+=13
185         if plot=='d':
186             n+=18
187 
188         global meas_cov; global SM_exp_cov
189         var=np.zeros((n_dof, n_dof))
190         var[n][n]=meas_cov[n][n]
191 
192         meas_cov[n] = 0   # zeroes out row n
193         meas_cov[:,n] = 0 # zeroes out column n
194 
195         meas_cov = meas_cov + var; # Add variance back.
196 
197         if plot=='a':
198             print('Removed correlations due to bin '+str(n+1)+' from plot a')
199         if plot=='b':
200             print('Removed correlations due to bin '+str(n+1-7)+' from plot b')
201         if plot=='c':
202             print('Removed correlations due to bin '+str(n+1-13)+' from plot c')
203         if plot=='d':
204             print('Removed correlations due to bin '+str(n+1-18)+' from plot d')
205 
206     def diag_meas_cov():
207         global meas_cov;
208         diag=np.diag(meas_cov)
209         meas_cov=np.diag(diag)
210         print('Zeroed off-diagonals of meas_cov')
```

III. PYTHON3 script to predict the FI for some distribution:

```
# Calculating Fisher information from d^2 ln L / d lda^2:
# NEW MG ONLY DATA:
import numpy              as np
import scipy.stats        as stats
import Functions_library as fl
import matplotlib.pyplot as plt

save_to='/Users/Genis/mphys/Rivet/DM/HNoE/plots/plots_dec_13';
# ----------------------------------------------------------------------
def throw_toys(mean, std_dev, ntoys, nbins=30, plot=True):
    toys=[]; weights=[];
    for i in range(ntoys):
        ith_toy=np.random.normal(loc=mean, scale=std_dev)
        toys.append(ith_toy)
    mean_toys=np.sum(toys)/ntoys

    if plot==True:
        figt, axt = plt.subplots(1,figsize=(9,6),sharex=True)

        # Create temporary histogram to calculate normalisation:
        h_temp=plt.hist(toys, bins=nbins, histtype='bar', label='Toys')
        bin_width=h_temp[1][1]-h_temp[1][0]
        areas=bin_width*h_temp[0]
        # Normalise to area = 1
        norm_factor=1.0/np.sum(areas);
        weights[:]=[norm_factor for i in range(ntoys)]

        plt.close('all')
        del figt, axt, h_temp

        fig, ax = plt.subplots(1,figsize=(9,6),sharex=True)

        ax.set_title('Toys arround measured data',fontsize=16)
        ax.set_yscale('linear')

        x = np.linspace(mean - 3*std_dev, mean + 3*std_dev, 200)
        ax.plot(x, stats.norm.pdf(x, mean, std_dev), label='Expected Gaussian')
        ax.hist(toys, bins=nbins, weights=weights, histtype='bar', label='Toys')
        ax.text(0.03, 0.93, 'ntoys = '+str(ntoys), fontsize=15, transform=ax.
    transAxes)
        ax.text(0.03, 0.87, 'Toys mean = '+str(np.round(mean_toys, decimals=2)),
    fontsize=15, transform=ax.transAxes)
        ax.set_ylabel('Probability', fontsize=15)
        ax.set_xlabel('Toys',fontsize=15)
        ax.legend(loc="upper right", fontsize=13)
    plt.show()
    return(toys)


def dlogLsquared(x, lda, p, BSM1, sigma_x, sigma_p):
    # return d^2 ln L / d lda^2 assuming Gaussian L.
    return(((x-p)/(sigma_x**2+sigma_p**2))*(42*BSM1/lda**8)-(36*BSM1*BSM1/(lda
    **14*(sigma_x**2+sigma_p**2))))
    # Throw toys to find average.

def get_info(x, lda, p, BSM1, sigma_x, sigma_p, ntoys=5000, plot_dist=True):
    toys=throw_toys(mean=x, std_dev=sigma_x, ntoys=ntoys, nbins=50, plot=False)

    All_dlogLsquared=[]
    for i in range(ntoys):
        x=toys[i];
        dlogLsquared_pertoy = dlogLsquared(x, lda, p, BSM1, sigma_x, sigma_p)
        All_dlogLsquared.append(dlogLsquared_pertoy)
```

```
61      mean=np.sum(All_dlogLsquared)/ntoys
62
63      if plot_dist==True:
64          fig, ax = plt.subplots(1,figsize=(9,6),sharex=True)
65
66          ax.set_title('Toys arround measured data',fontsize=16)
67          ax.set_yscale('linear')
68
69          ax.hist(All_dlogLsquared, bins=50, histtype='bar', label='Toys')
70          ax.text(0.03, 0.93, 'ntoys = '+str(ntoys), fontsize=15, transform=ax.
        transAxes)
71          ax.text(0.03, 0.87, 'Toys mean = '+str(np.round(mean, decimals=5)),
        fontsize=15, transform=ax.transAxes)
72          ax.set_ylabel('Toy Frequency', fontsize=15)
73          ax.set_xlabel(r'$\frac{d^2 ln\ L}{d \Lambda_{EFT}^2}$',fontsize=15)
74          ax.legend(loc="upper right", fontsize=13)
75          plt.show()
76
77      return (-1.0*mean)
78
79
80  def get_fisher_var(var, lda_new, bins_, sample, scale_factors, x_label):
81  # lda_new: Rescale data to this lambda, to find fisher information.
82      m_chi=100;              # Mass of MC simulations.
83      l=100;                  # Lambda fof MC simulations.
84      rescale_factor=1.114;   # This is the ratio of cross-sections HepData/MG
85
86      if sample=='vbf':       # These are for the histogram titles
87          tit = 'VBF';
88      elif sample == 'mono':
89          tit = r'$\geq$ 1 jet';
90      else:
91          raise ValueError("Please choose sample 'mono' or 'vbf'")
92
93      data=fl.read_data(sample);  # Read CSV files
94      sig=data[0]; bkg_2=data[2]; jpt160=data[3];
95
96      # Choose al data within some upper limit upper_lim:
97      upper_lim=bins_[len(bins_)-1]
98      sig=sig.query("{}{}{}".format((var),"<", upper_lim))
99      jpt160=jpt160.query("{}{}{}".format((var),"<",upper_lim))
100     bkg_2=bkg_2.query("{}{}{}".format((var),"<", upper_lim))
101
102     fig, ax0 = plt.subplots(1,figsize=(8,6),sharex=True)
103     ax0.set_title('$\sigma$ vs '+var+' for the '+sample+' region',fontsize=16)
104     ax0.text(0.03, 0.93, '$\Lambda_{EFT}$ = '+str(lda_new)+' GeV', fontsize=15,
        transform=ax0.transAxes)
105     ax0.text(0.03, 0.88,  '$M_{\chi}$   = '+str(m_chi)+' GeV', fontsize=15,
        transform=ax0.transAxes)
106
107     h=ax0.hist([bkg_2[var],jpt160[var],sig[var]], weights = [bkg_2['weight']
108     *scale_factors[2],jpt160['weight']*(scale_factors[1]),sig['weight']
109     *scale_factors[0]*pow(1/lda_new,6.0)], bins=bins_, color=['green','red','blue'
        ], stacked=False,
110     fill=False, histtype='step', label=[r'EWK $Z \rightarrow \nu \overline{\nu} +
        j j$ ', r'QCD $Z \rightarrow \nu \overline{\nu} + j$ ' ,r'EFT  $ A \rightarrow
        \chi \overline{\chi}$ + jj'])
111
112     ax0.set_ylabel('$\sigma$ (pb)$', fontsize=14)
113     ax0.set_xlabel(var, fontsize=14)
114     ax0.legend(loc="upper right", fontsize=12)
115
116     errors=fl.get_error2(h, sample, var, lda_new, scale_factors)
117     peaks_sig=[]; peaks_bkg_1=[]; peaks_bkg_2=[];
118     for i in range(len(bins_)-1):
```

```
119         peaks_sig.append(h[0][2][i])
120         peaks_bkg_1.append(h[0][1][i])
121         peaks_bkg_2.append(h[0][0][i])
122 
123     DM_var      = peaks_sig
124     DM_var_err  = errors[0]
125     SM_var      = [(x+y)*rescale_factor for x,y in zip (peaks_bkg_1,peaks_bkg_2)]
126     SM_var_err  = [np.sqrt(x**2+y**2)   for x,y in zip (errors[1],  errors[2])]
127 
128     plt.close('all')
129     del fig, ax0, h
130 
131     # FISHER INFORMATION:
132 
133     # SM PREDICTION & EXPERIMENTAL ERROR:
134     exp_frac_err=0.1;      # Use 10% expected experimental covariance for now...
135 
136     x           = SM_var                        # SM hypothesis
137     sigma_x     = [exp_frac_err*x for x in SM_var]  # Expected Experimental
    Covariance.
138 
139     # BSM PREDICTION & (MADGRAPH) ERRORS:
140     p           = [x+y for x,y in zip(SM_var, DM_var)]                  # SM+DM
    prediction at l=400.
141     sigma_p     = [np.sqrt(x+y) for x,y in zip(SM_var_err, DM_var_err)] # Errors
    for SM+DM at l=400.
142 
143 
144     # DM ONLY PREDICTION &  ERRORS:
145     DM          = DM_var
146     sigma_DM    = DM_var_err
147 
148     BSM1        = [lda_new**6*DM for DM in DM_var] # DM at lambda=1: Used to calc.
    F.I.
149 
150     info_all_bins=[]
151 
152     for i in range(len(x)):
153         info_bin_i=get_info(x[i], lda_new, p[i], BSM1[i], sigma_x[i], sigma_p[i],
    plot_dist=False)
154         info_all_bins.append(info_bin_i)
155 
156     fig, ax = plt.subplots(1,figsize=(9,6),sharex=True)
157     ax.set_title(x_label+' vs $\sigma$ for the '+tit+' region',fontsize=16)
158 
159     ax.set_ylabel('$\sigma (pb)$', fontsize=14)
160     ax.set_xlabel(x_label, fontsize=14)
161 
162     # Get Bin Centres and Widths:
163     bin_centres=[]; bin_widths=[]
164     for i in range(len(bins_)-1):
165         ith_centre = 0.5*(bins_[i+1]-bins_[i])+bins_[i]
166         ith_width  = (bins_[i+1]-bins_[i])
167         bin_centres.append(ith_centre)
168         bin_widths.append(ith_width)
169 
170     ax.bar(bin_centres,  x, fill=False, width=bin_widths, yerr=sigma_x,   ecolor='
    blue', edgecolor='blue', capsize=4, label='MG5(LO) SM prediction (with '+str(
    exp_frac_err*100)+'% error).')
171     ax.bar(bin_centres, DM, fill=False, width=bin_widths, linestyle='--', yerr=
    sigma_DM, ecolor='red', edgecolor='red', capsize=4, label='MG5(LO) DM at $m_{\
    chi}$='+str(m_chi)+' $\Lambda_{EFT}$='+str(lda_new)+'.')
172 
173     ax_two=ax.twinx()
174     ax_two.bar(bin_centres, info_all_bins, fill=True, alpha=0.2, width=bin_widths,
```

```
        facecolor='orange', label='Fisher Information')
175
176     ax_two.set_ylabel('Fisher Information')
177     ax_two.set_ylim(0)
178
179     ax.legend(loc="upper left", fontsize=8)
180     ax_two.legend(loc="upper right", fontsize=8)
181     plt.savefig(save_to+'/Fisher_Info_'+var+'_'+sample+'_m'+str(m_chi)+'_l'+str(
        lda_new)+'.pdf')
182     plt.show()
183
184     print('Information:', info_all_bins)
185
186
187 #-------------------------------------------------------------------------------
188
189 #Plot:
190 sum_of_weights=[29715.6,2354.22,6.33457];
191 cross_sections=[4.28E+04,2.26E+03,6.334571];
192 scale_f=[xs / sw for xs,sw in zip(cross_sections,sum_of_weights)];
193
194 # FROM THE PAPER:
195 # a. MET FOR MONO REGION
196 bins = [200, 250, 300, 350, 500, 900, 1500]
197 get_fisher_var('met', 400, bins, 'mono', scale_f, r'$pT^{miss}$')
198
199 # b. MET FOR VBF REGION
200 bins = [200, 250, 300, 350, 500, 700, 1400]
201 get_fisher_var('met', 400, bins, 'vbf', scale_f, r'$pT^{miss}$')
202
203 # c. MJJ FOR VBF REGION:
204 bins = [200,400,600,1000,2000,4000]
205 get_fisher_var('mjj', 400, bins, 'vbf', scale_f, r'$m_{jj}$')
206
207 # d. DPHI FOR VBF REGION:
208 bins = [0, np.pi/6, np.pi/3, np.pi/2, 2*np.pi/3, 5*np.pi/6, np.pi]
209 get_fisher_var('dphi', 400, bins, 'vbf', scale_f, r'$\Delta \phi_{jj}$')
210
211 # New Plots with Fisher Information: Pseudorapidity / jpt1 / jpt2.
212
213 # Pseudorapidity:
214 bins= [0, 0.5, 1.0, 1.5, 2, 4]
215 get_fisher_var('jeta1', 400, bins, 'mono', scale_f, 'Leading jet $\eta$')
216 get_fisher_var('jeta1', 400, bins, 'vbf', scale_f, 'Leading jet $\eta$')
217
218 # Leading Jet pT:
219 bins=[200, 250, 300, 400, 600, 1000]
220 get_fisher_var('jpt1',  400, bins, 'mono', scale_f, 'Leading jet $pT$')
221 get_fisher_var('jpt1',  400, bins, 'vbf', scale_f, 'Leading jet $pT$')
222
223
224 # Subleading Jet pT:
225 bins=[50, 75, 100, 150, 200, 300, 500]
226 get_fisher_var('jpt2', 400, bins, 'vbf', scale_f, 'Subleading jet $pT$')
```

IV. Measured Fisher Information ($\times 10^{-3}$) for VBF distributions:

		m_χ		
		100	500	900
Λ_{EFT}	400	1909	998	378
	600	21.2	8.75	1.95
	800	0.38	0.15	0.03

Table 13: p_T^{miss}

		m_χ		
		100	500	900
Λ_{EFT}	400	856	432	115
	600	5.22	2.02	0.42
	800	0.09	0.03	0.01

Table 14: m_{jj}

		m_χ		
		100	500	900
Λ_{EFT}	400	607	239	54.8
	600	2.02	0.74	0.14
	800	0.02	0.00	0.00

Table 15: $\Delta\phi_{jj}$

V. Distributions of expected information:

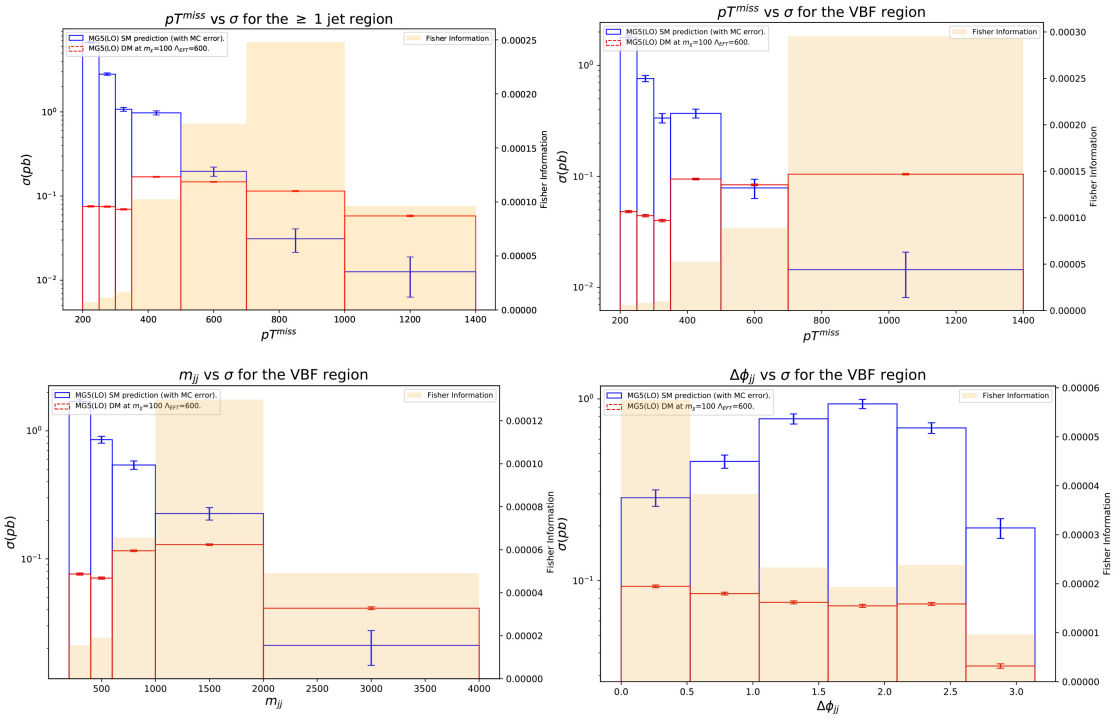


Figure 17: Expected Information for the 4 distributions originally measured at ATLAS.

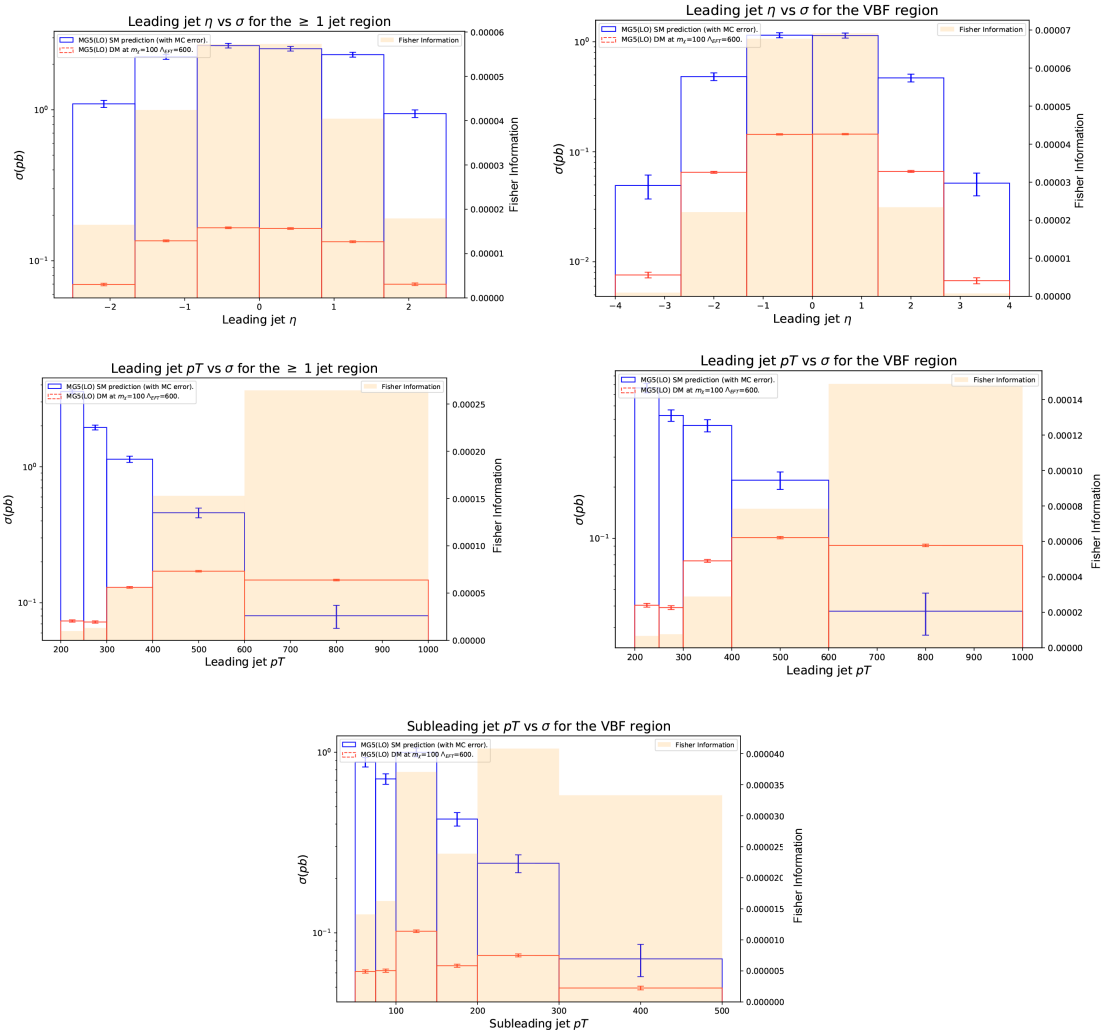


Figure 18: Expected Information for 5 new distributions.