PROCEEDINGS OF SCIENCE

CheckMATE: Checkmating new physics at the LHC

Jamie Tattersall[*]
Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University
E-mail: tattersall@physik.rwth-aachen.de

Daniel Dercks
II. Institute for Theoretical Physics, University of Hamburg
Physikalisches Institut der Universität Bonn
E-mail: daniel.dercks@desy.de

Nishita Desai
Laboratoire Charles Coulomb (L2C) & Laboratoire Univers et Particules de Montpellier (LUPM), CNRS-Université de Montpellier
E-mail: nishita.desai@umontpellier.fr

Jong Soo Kim
CTPU, Daejeon Institute for Basic Science
E-mail: jongsoo.kim@tu-dortmund.de

Frederic Poncza
Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University
E-mail: poncza@physik.rwth-aachen.de

Krzysztof Rolbiecki
University of Warsaw
E-mail: krzysztof.rolbiecki@fuw.edu.pl

Torsten Weber
Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University
E-mail: torsten.weber@rwth-aachen.de

In the first four years of running, the LHC has delivered a wealth of new data that is now being analysed. The two multi-purpose detectors, ATLAS and CMS, have performed many searches for new physics but theorists are eager to test their own particular model. We present the latest developments to the program CheckMATE (Check Models At Terascale Energies) that helps to automate this procedure so that new theories can easily be checked against the latest results. The backbone of CheckMATE is a library of over 50 LHC analyses that new physics models can be tested against. The user only needs to provide a model in SLHA format and CheckMATE will automatically decide if the model is ruled out or not. In addition, if new physics begins to appear, CheckMATE offers the possibility to quickly determine the model that best fits the data. Finally, a novel technique using kinematic transformations of existing events is presented that promises to speed up the determination of model best fit points by orders of magnitude.

38th International Conference on High Energy Physics
3-10 August 2016
Chicago, USA

[*] Speaker.

1. Overview of CheckMATE

CheckMATE[1] (Check Models At Terascale Energies) [1] is a well established publicly available program that allows for the easy testing of Beyond the Standard Model (BSM) theories against the latest Large Hadron Collider (LHC) data. The original version takes Monte-Carlo files produced by the user's own favourite event generator and in the default setting, CheckMATE states if the model is excluded or not at the 95% confidence level.

To make model testing even easier, an updated version version of CheckMATE will be available shortly[2] that now integrates the whole event generation chain via `MG5_aMC@NLO` [2] and `Pythia 8` [3]. Consequently the user now only has to specify the process of interest along with a model in the SLHA format and CheckMATE will automatically decide if the model is excluded.

The program flow for the new version is given in Fig. 1 where we see the multiple input options that CheckMATE now contains. Events can either be generated by `MG5_aMC@NLO` and then showered by `Pythia 8` or if the model and process is internally available in `Pythia 8`, the complete generation chain can be performed there. If the user wishes to use a specialised code to calculate the matrix element, a LHE input file can also be used and `Pythia 8` will then shower these events. We should also note that fully showered events in either `HepMC` or `STDHEP` format can still be given as input, just like the original CheckMATE.

Whilst ease of use was a clear motivation for including event generation within CheckMATE, the main reason for the development was the significant improvement in speed that results. This is due to large event files no longer being written and read from a hard disk and on a single core, the code is $\sim$2X faster. However, the real benefit is seen when running multiple instances of CheckMATE on a cluster where hard disk access was often the limiting factor. In this case, the program can now easily run an order of magnitude more quickly.

The new version of CheckMATE now also includes over 50 analyses of which more than 10 are using the latest 13 TeV LHC results (see e.g. [4]). Furthermore, analyses that target the high luminosity LHC run are now available for the first time and these have already been used in studies that estimate the exclusion and discovery reach of particular models (see e.g. [5]). In addition, the internal AnalysisManager [6] has also been significantly updated for the new release. This allows existing LHC analyses to be easily added to CheckMATE and also the quick and easy prototyping of new search strategies (see e.g. [7]).

2. Program Structure

The new input possibilities has meant that significant portions of CheckMATE have now be rewritten to make the code more flexible and the program flow is depicted in Fig. 1. As stated in the introduction, event generation can now be handled internally by CheckMATE which now links to `MG5_aMC@NLO` and `Pythia 8` via the newly developed C++ FRITZ (Flexible Rapid Interactive Tool Zipper). However, CheckMATE retains the feature that event generation can also be done externally if the user wishes.

[1] CheckMATE can be downloaded at https://checkmate.hepforge.org/

[2] A β-version is already available online for testing purposes

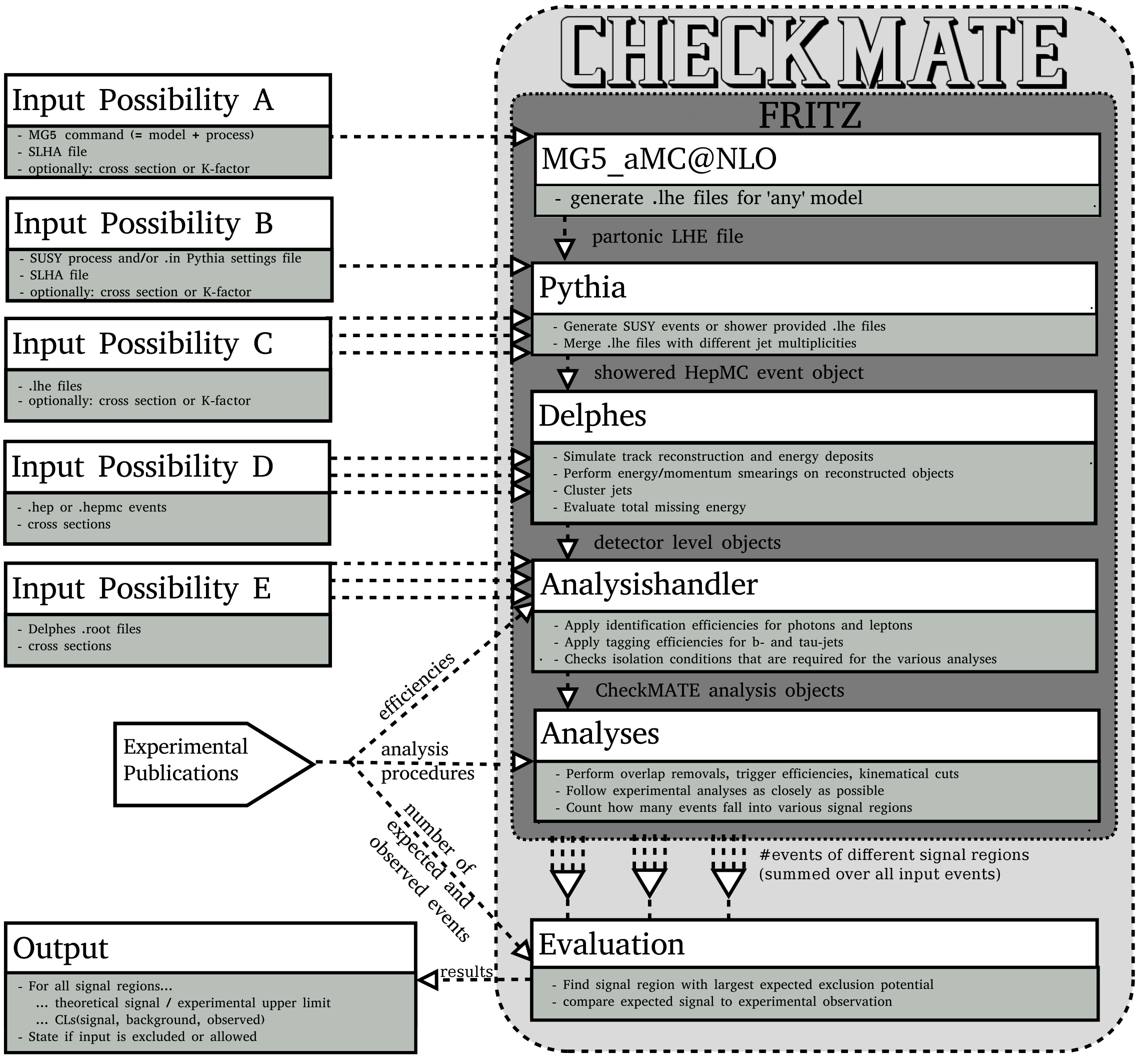
CHECKMATE
FRITZ
Input Possibility A
- MG5 command (= model + process)
- SLHA file
- optionally: cross section or K-factor
Input Possibility B
- SUSY process and/or .in Pythia settings file
- SLHA file
- optionally: cross section or K-factor
Input Possibility C
- .lhe files
- optionally: cross section or K-factor
Input Possibility D
- .hep or .hepmc events
- cross sections
Input Possibility E
- Delphes .root files
- cross sections
MG5_aMC@NLO
- generate .lhe files for 'any' model
partonic LHE file
Pythia
- Generate SUSY events or shower provided .lhe files
- Merge .lhe files with different jet multiplicities
showered HepMC event object
Delphes
- Simulate track reconstruction and energy deposits
- Perform energy/momentum smearings on reconstructed objects
- Cluster jets
- Evaluate total missing energy
detector level objects
Analysishandler
- Apply identification efficiencies for photons and leptons
- Apply tagging efficiencies for b- and tau-jets
- Checks isolation conditions that are required for the various analyses
CheckMATE analysis objects
Analyses
- Perform overlap removals, trigger efficiencies, kinematical cuts
- Follow experimental analyses as closely as possible
- Count how many events fall into various signal regions
#events of different signal regions
(summed over all input events)
Experimental Publications
efficiencies
analysis procedures
number of expected and observed events
Evaluation
- Find signal region with largest expected exclusion potential
- compare expected signal to experimental observation
results
Output
- For all signal regions...
... theoretical signal / experimental upper limit
... CLs(signal, background, observed)
- State if input is excluded or allowed

Figure 1: Diagram illustrating the data flow through CheckMATE along with the various input and output possibilities.

After an event is generated (or read from file), it is passed to `Delphes 3` [8] which simulates the calorimeter and track reconstruction of either ATLAS or CMS. As a change from the original CheckMATE version, all identification, tagging and reconstruction efficiencies are now handled internally by CheckMATE which allows for a better tuning of these parameters.

The fully reconstructed event is then passed to the core of CheckMATE which are the library of experimental analyses. Here we attempt to reproduce all features of the experimental analyses as closely as possible to accurately calculate the experimental acceptances for each signal region. For every validated analysis we publish a document online where we test the CheckMATE results against the publicly available cutflows and exclusions provided by the experimental collaborations.

As the final step, the user can then choose various statistical measures to decide whether or not the model is ruled out. The simplest metric available is given by,

$$r \equiv \frac{S - 1.64\Delta S}{S^{95}_{\text{obs}}}\,, \tag{2.1}$$

where S is the number of signal events predicted in the signal region expected to be most constraining by the BSM theory, ΔS is the 1 σ uncertainty on this prediction and S^{95}_{obs} is the experimentally determined 95% confidence limit on the signal. In this case, a model can be considered as excluded to the 95% confidence level, if $r > 1$. Alternatively, the option is also available to calculate the full CLs with the signal uncertainty treated as a nuisance parameter if extra accuracy is required or the user wishes to know the actual CLs value. Finally, as a new option, CheckMATE can also calculate the combined likelihood of all considered signal regions. This allows the user to calculate a p-value for the model being tested and is especially useful when trying to fit signal excesses.

3. Fast parameter scans

Whilst CheckMATE is able to test a huge range of BSM theories, the program does suffer from a significant issue, namely the computing time required. Depending on the model under test and the number of Monte-Carlo events that are required for adequate statistics, this time can range from a few minutes to even a few hours on a single CPU.

Such a computationally intensive task becomes a severe issue when we wish to explore models with many degrees of freedom (e.g. PMSSM-19). Consequently, there is a significant incentive to develop algorithms that can substantially increase the speed of individual parameter point testing with the aim of reducing the computational time to under a second. However the method should be general enough so that it can work automatically with most of the current BSM theories.

Progress was made towards such an algorithm in [9] where a method is detailed of reusing already generated and analysed Monte-Carlo events. The idea is that the couplings and spin structure of the new physics model can be arbitrarily changed and accounted for simply by reweighting the events according to the corresponding matrix element. However, the algorithm is unable to account for changes in the masses of the new physics particles which is a significant drawback since it is these parameters that are often most important was assessing the LHC phenomenology.

To solve this outstanding issue, we propose a set of quick transformations that take existing Monte-Carlo events generated with a particular model parameter point and return the kinematics at some other parameter point. For brevity we explain for the simplified case of a new s-channel resonance with mass $\tilde{M}$ where the kinematics are transformed from events that were generated using a mass M. In the simplifying case that the resonance has no transverse momentum, the transformed four vector is given by,

$$\tilde{\mathbf{P}} = \left(\sqrt{\tilde{x}_1\tilde{x}_2 s}\,, 0\,, 0\,, \frac{\sqrt{s}}{2}(\tilde{x}_1 - \tilde{x}_2)\right)\,, \tag{3.1}$$

where s is the collider energy and $\tilde{x}_i$ are the transformed parton momentum fractions which are calculated using,

$$\tilde{x}_i = a x_i\,, \qquad a = \frac{\tilde{M} + b\tilde{\Gamma}}{M + b\Gamma}, \qquad b = \frac{\sqrt{x_1 x_2 s} - M}{\Gamma}\,. \tag{3.2}$$

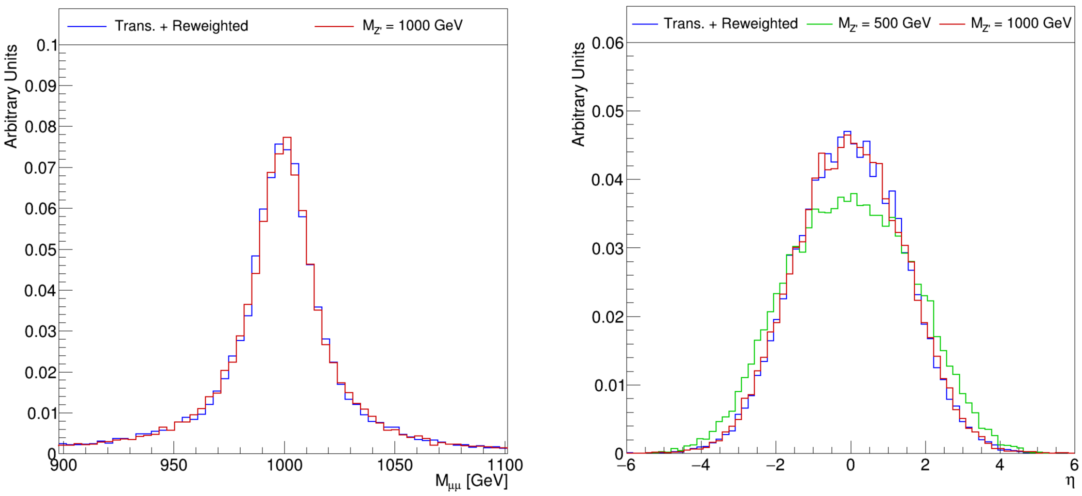

Figure 2: Distributions showing Monte-Carlo events for the process $pp \to Z' \to \mu^+\mu^-$ at 13 TeV. The red curve shows a model with $M_{Z'} = 1000$ GeV, the green $M_{Z'} = 500$ GeV and the red events that have been transformed from $M_{Z'} = 500$ GeV to $M_{Z'} = 1000$ GeV. Left: invariant mass of the muons, right: rapidity of the muons.

Here x_i are the original parton momentum fractions, $\Gamma(\tilde{\Gamma})$ is the original (transformed) width of the resonance. The above equations ensure that both the peak position and the width of resonance are correctly reproduced for the transformed events.

In order that the longitudinal components of the four momentum also follow the appropriate distribution, reweighting by the PDFs is particularly important. This is easily done by simply applying the weight, w, to each event using,

$$w = \frac{f_a(\tilde{x}_1)f_b(\tilde{x}_2)}{f_a(x_1)f_b(x_2)}, \tag{3.3}$$

where f_i are the parton distribution functions of the relevant incoming partons of the process.

The above equations specify the four-momentum of the s-channel mediator in our example. The momentum of the decay products now follow from two very simple principles. Firstly we demand that each decay vertex respects four-momentum conservation. Secondly, we define the direction of the individual decay products in the rest frame of the decaying particle is the same as the original event. Such a procedure uniquely defines the four-momentum of all particles produced by the resonance.

The results of the algorithm can be seen in Fig. 2 where we display the results of transforming events in a model with a hypothetical Z' resonance when the mass is doubled from $M_{Z'} = 500$ GeV to $M_{Z'} = 1000$ GeV. We see that within the statistical uncertainties, the transformed events successfully reproduce the correct distributions.

As a second example, we present a toy model where we double the Standard Model top mass. We see that as we go from $M_t = 173$ GeV to $M_t = 346$ GeV, the distributions are accurate for the final state leptons even though these events contain a complicated multi-stage decay.

The above algorithm will be integrated within the CheckMATE framework so that the parameter space of a model can be constrained far faster than is feasible at the present time.

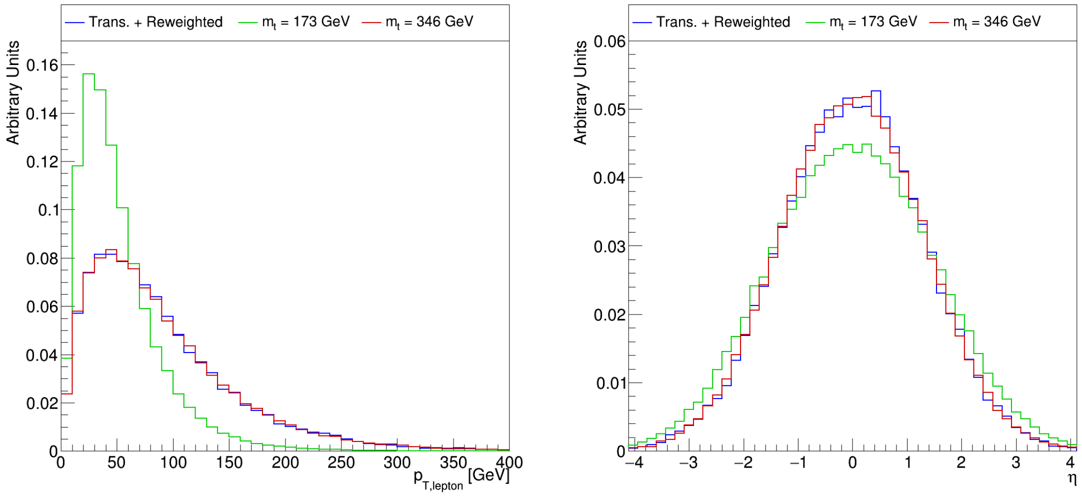

Figure 3: Distributions showing Monte-Carlo events for the process $pp \to t\bar{t} \to bl^+\nu\bar{b}l^-\bar{\nu}$ at 13 TeV. The red curve shows a model with $M_t = 346$ GeV, the green $M_t = 173$ GeV and the red events that have been transformed from $M_t = 173$ GeV to $M_t = 346$ GeV. Left: p_T of the leptons, right: rapidity of the leptons.

4. Conclusion

In this proceedings we have presented the latest developments to the CheckMATE program that allows new physics models to be tested against the latest LHC data. We also propose a new algorithm that will allow model scans to be carried out far quicker than is currently possible and will be publicly available soon.

References

[1] M. Drees, H. Dreiner, D. Schmeier, J. Tattersall and J. S. Kim, Comput. Phys. Commun. **187**, 227 (2015) doi:10.1016/j.cpc.2014.10.018 [arXiv:1312.2591 [hep-ph]].

[2] J. Alwall *et al.*, JHEP **1407**, 079 (2014) doi:10.1007/JHEP07(2014)079 [arXiv:1405.0301 [hep-ph]].

[3] T. Sjöstrand *et al.*, Comput. Phys. Commun. **191**, 159 (2015) doi:10.1016/j.cpc.2015.01.024 [arXiv:1410.3012 [hep-ph]].

[4] A. Biekötter, J. L. Hewett, J. S. Kim, M. Krämer, T. G. Rizzo, K. Rolbiecki, J. Tattersall and T. Weber, arXiv:1608.01312 [hep-ph].

[5] J. S. Kim, K. Rolbiecki, R. Ruiz, J. Tattersall and T. Weber, arXiv:1606.06738 [hep-ph].

[6] J. S. Kim, D. Schmeier, J. Tattersall and K. Rolbiecki, Comput. Phys. Commun. **196**, 535 (2015) doi:10.1016/j.cpc.2015.06.002 [arXiv:1503.01123 [hep-ph]].

[7] K. Rolbiecki and J. Tattersall, Phys. Lett. B **750**, 247 (2015) doi:10.1016/j.physletb.2015.09.028 [arXiv:1505.05523 [hep-ph]].

[8] J. de Favereau *et al.* [DELPHES 3 Collaboration], JHEP **1402**, 057 (2014) doi:10.1007/JHEP02(2014)057 [arXiv:1307.6346 [hep-ex]].

[9] J. S. Gainer, J. Lykken, K. T. Matchev, S. Mrenna and M. Park, JHEP **1410**, 078 (2014) doi:10.1007/JHEP10(2014)078 [arXiv:1404.7129 [hep-ph]].