



City Research Online

City St George's, University of London

Citation: Araz, J. Y., Fuks, B. & Polykratis, G. (2021). Simplified fast detector simulation in MadAnalysis 5. The European Physical Journal C, 81(4), 329. doi: 10.1140/epjc/s10052-021-09052-5

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/36221/>

Link to published version: <https://doi.org/10.1140/epjc/s10052-021-09052-5>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).



Simplified fast detector simulation in MADANALYSIS 5

Jack Y. Araz^{1,2,a} , Benjamin Fuks^{3,4,b} , Georgios Polykratis^{3,c}

¹ School of Physics & Astronomy, University of Glasgow, Glasgow, UK

² Concordia University, 7141 Sherbrooke St. West, Montréal, QC H4B 1R6, Canada

³ Laboratoire de Physique Théorique et Hautes Energies (LPTHE), UMR 7589, Sorbonne Université et CNRS, 4 place Jussieu, 75252 Paris Cedex 05, France

⁴ Institut Universitaire de France, 103 boulevard Saint-Michel, 75005 Paris, France

Received: 23 June 2020 / Accepted: 15 March 2021 / Published online: 18 April 2021

© The Author(s) 2021

Abstract We introduce a new simplified fast detector simulator in the MADANALYSIS 5 platform. The PYTHON-like interpreter of the programme has been augmented by new commands allowing for a detector parametrisation through smearing and efficiency functions. On run time, an associated C++ code is automatically generated and executed to produce reconstructed-level events. In addition, we have extended the MADANALYSIS 5 recasting infrastructure to support our detector emulator, and we provide predefined LHC detector configurations. We have compared predictions obtained with our approach to those resulting from the usage of the DELPHES 3 software, both for Standard Model processes and a few new physics signals. Results generally agree to a level of about 10% or better, the largest differences in the predictions stemming from the different strategies that are followed to model specific detector effects. Equipped with these new functionalities, MADANALYSIS 5 now offers a new user-friendly way to include detector effects when analysing collider events, the simulation of the detector and the analysis being both handled either through a set of intuitive PYTHON commands or directly within the C++ core of the platform.

1 Introduction

The discovery of the last missing particle of the Standard Model at the Large Hadron Collider (LHC) at CERN has opened a new era in our understanding of the fundamental laws of nature. However, the concrete mechanism behind electroweak symmetry breaking is still today a mystery and there is no sign of phenomenon beyond the Standard Model, despite the wealth of data currently available. As a conse-

quence, the experimental LHC search results are interpreted as stronger and stronger constraints on a large set of new physics models, those constraints being obtained by comparing associated predictions with data.

Those reinterpretations can be achieved in two ways. First, they can rely on experimental data from which the detector effects have been unfolded, *i.e.* by considering LHC data as observed by a perfect detector with an infinite resolution and an ideal calibration. This requires an excellent understanding of the background and is a complex, ill-defined and time-consuming problem, as there is no unique solution to the inversion of the convolution of the detector response [1]. Consequently, reinterpretations are usually performed by adopting a second approach in which the detector effects are included, or folded forward, in the simulation of the new physics signals to be confronted to data. This folding is expected to appropriately capture the impact of the inner working of the detector and the inefficiency of the reconstruction of the event record. This has the advantage of being computationally much more acceptable.

The most accurate forward folding method relies on the implementation of the exact details of the detector functioning in a framework based on the GEANT 4 package [2]. The latter allows for the modelling of the detector geometry and material interactions, from which one could then deal with the simulation of the electronic response of the detector. As a final step, it is required to reproduce the impact of the reconstruction details associated with any given experiment, which yields appropriate definitions for the various physics objects used in an analysis. This is, however, in practice not achievable by anyone outside the collaborations, by virtue of the lack of publicly available information. In addition, even if feasible, running such a chain of tools for the plethora of new physics model potentially interesting is likely to be computationally unfeasible, as this requires several minutes of computing time for a single simulated event.

^a e-mail: jack.araz@glasgow.ac.uk

^b e-mail: fuks@lpthe.jussieu.fr (corresponding author)

^c e-mail: giopolykra@gmail.com

The resolution, reconstruction and identification efficiencies corresponding to all final-state objects relevant for a physics analysis are however often publicly available under the form of functions of standard object properties (like the transverse momentum or pseudo-rapidity). This, therefore, opens the door to a well-motivated and computationally much more efficient approach, as is implemented in packages like the DELPHES 3 software [3] or the RIVET toolkit [4].

In DELPHES 3, the simulation of the detector approximates the steps followed in a GEANT-based approach. It relies on approximate experiment geometries and particle propagation models, that are combined with tabulated reconstruction and identification efficiencies to yield the reconstructed physics objects to use in an analysis. The latter are obtained from hadron-level events clustered through one of the different algorithms available from the FASTJET package [5], together with potentially more complex techniques, such as particle-flow and energy-flow methods. Moreover, the programme allows for the optional simulation of pile-up. In RIVET, a lighter approach is implemented [6]. Effective transfer functions, including the smearing of the object kinematics properties and reconstruction efficiencies, are used to connect the Monte Carlo representation of any physics object (*i.e.* at the truth level) to their reconstructed representation.

In this paper, we report on the design of an extension of the capabilities of the MADANALYSIS 5 framework [7–9] to handle the simulation of the response of a detector. MADANALYSIS 5 is a general platform for beyond the Standard Model phenomenology. It can deal both with the development of an analysis of any given collider signal (together with its associated background) via a user-friendly PYTHON-based command-line interface and a developer-friendly C++ core. MADANALYSIS 5 can also be used for the (automated) reinterpretation of existing LHC results [10, 11].

The response of a typical detector can be emulated in MADANALYSIS 5 in two ways. The user has first the option to rely on the interface of the code to the FASTJET package [5]. Whilst fast and efficient, this way of proceeding is restricted to the simulation of a detector with an infinite resolution. It indeed leads to the sole application of a jet clustering algorithm (as available from FASTJET) to reconstruct events possibly together with flat parametrisations to model some specific reconstruction effects like *b*-tagging or tau-tagging. Nevertheless, this procedure is useful for studies dedicated to a particular effect or at the Monte Carlo truth level.

As a simple application of a jet algorithm is known to be quite unrealistic in many contexts for which detector effects matter, MADANALYSIS 5 has been interfaced to the DELPHES 3 package to enable a more detailed simulation of the detector effects. This interface drives the run of DELPHES 3, and additionally switches on the usage of new modules that are specific to MADANALYSIS 5. These are for instance related to a better handling of object isolation or

the skimming of the output ROOT file. The price to pay to employ DELPHES 3 instead of simply FASTJET is obviously a much slower run of the code, as stemming from the larger complexity of DELPHES 3 with respect to FASTJET.

In the following, we detail several improvements that have been added to the MADANALYSIS 5 interface to FASTJET. These allow the user to include a lightweight, therefore computationally cheap, and realistic detector simulation when Monte Carlo event reconstruction (with FASTJET) is in order.

Thanks to those developments, MADANALYSIS 5 offers now the flexibility to rely on any specific set of transfer functions to effectively map the Monte Carlo objects to their reconstructed counterparts (*i.e.* reconstruction efficiencies, tagging efficiencies and the corresponding mistagging rates). Moreover, the properties of the different objects can be smeared to mimic detector resolution degradations. MADANALYSIS 5 becomes thus the only publicly available high-energy physics package that offers the user the choice of either relying on lightweight smearing and efficiency functions or the heavier DELPHES 3 framework to model the response of a typical high-energy physics detector, easing hence the potential comparison of the pros and cons and of two methods within a single platform. Moreover, thanks to the MADANALYSIS 5 PYTHON interpreter and the associated intuitive meta-language, the parametrisation of the detector is user-friendly and the generation of the corresponding C++ code (either when a DELPHES 3-based or transfer-function-based simulation is used), its compilation and running happen behind the scenes in a fully automated fashion. Our changes in the C++ core of the programme have been combined with new functionalities at the level of the PYTHON command-line interface, so that any user can straightforwardly implement his/her own detector parametrisation in a simple and user-friendly manner. In addition, the code is shipped with predefined sets of commands allowing to automatically load detector configurations associated with the ATLAS and CMS detectors.

The new features introduced in this work are available from version 1.8.51 of the code, that can be downloaded from LAUNCHPAD.¹ As said above, this release of the programme allows for the simulation of the impact of a detector in a way that combines realism and efficiency. In addition, it allows for dedicated studies singling out specific detector effects, that could potentially be applied directly at the particle level. The information included in this paper is summarised on the MADANALYSIS 5 website.²

The rest of this paper has been prepared as follows. In Sect. 2, we briefly review how to reconstruct a sample of Monte Carlo events in MADANALYSIS 5 with the help of its interface to FASTJET. In Sect. 3, we detail the new features

¹ See the webpage <https://launchpad.net/madanalysis5>.

² See the webpage <http://madanalysis.irmp.ucl.ac.be/wiki/SFS>.

Table 1 Jet algorithms available in MADANALYSIS 5, shown together with the properties than can be changed by the user on run time. The default values are indicated. The jet radius default value of 1 has been replaced by 0.4 in MADANALYSIS 5 releases posterior to v1.9.11

Algorithm	ptmin	Exclusive_id	Radius	p	Overlap	npassmax	Input_ptmin
kt	5	True	1	–	–	–	–
Cambridge	5	True	1	–	–	–	–
Antikt	5	True	1	–	–	–	–
Genkt	5	True	1	1	–	–	–
Siscone	5	True	1	–	0.5	0	0

that have been developed in order to allow the user to combine event reconstruction with a fast and realistic detector simulation. In Sect. 4, we compare predictions obtained with our new method with those arising from the usage of DELPHES 3. We moreover quantitatively assess the differences between the results returned by both detector simulators relatively to the Monte Carlo truth. We then present, in Sect. 5, how the new MADANALYSIS 5 fast and simplified detector simulator can be used for the reinterpretation of the results of two specific LHC analyses. In this context, a comparison with a more usual method relying on DELPHES 3 is performed. We summarise our work and conclude in Sect. 6.

2 Event reconstruction with MADANALYSIS 5

2.1 Running FASTJET from MADANALYSIS 5

Thanks to its interface to FASTJET, MADANALYSIS 5 allows for the reconstruction of hadron-level Monte Carlo events through the application of a jet-clustering algorithm. To proceed, the programme has to be started in the reconstructed mode,

```
./bin/ma5 -R
```

and the FASTJET package has to be present on the user system. If this is not the case, it is sufficient to type, in the MADANALYSIS 5 command-line interface,

```
install fastjet
```

to trigger a local installation of FASTJET, in the subfolder `tools/fastjet`.

The MADANALYSIS 5 run has then to be configured following the specifications of the user. First, one must switch event reconstruction on by typing

```
set main.fastsim.package = fastjet
```

This turns on the usage of FASTJET for event reconstruction and indicates to the code that the input hadron-level event

sample(s) are encoded following the HEPMC event file format [12].³

By default, jets are reconstructed by making use of the anti- k_T algorithm [13], with a radius parameter set to $R = 1$ for MADANALYSIS 5 releases prior to version 1.9.11 and $R = 0.4$ for more recent versions.⁴ This behaviour can be modified by typing in

```
set main.fastsim.algorithm = <algo>
set main.fastsim.<property>= <value>
```

where `<algo>` represents the keyword associated with the adopted jet algorithm, and `<property>` generically denotes any of its property. MADANALYSIS 5 can employ the longitudinally invariant k_T algorithm [14, 15] (`kt`), the Cambridge/Aachen algorithm [16, 17] (`cambridge`), the anti- k_T algorithm (`antikt`), the generalised k_T algorithm [5] (`genkt`), as well as the seedless infrared-safe cone algorithm [18] (`siscone`)⁵. We refer to Table 1 for the list of available jet algorithms, the corresponding options and their default values.

Among all the jet algorithm properties that can be tuned, three of them are common to all algorithms. The user can define a transverse momentum threshold (`ptmin`) so that any softer jet is filtered out, and fix the jet radius parameter R (`radius`) that dictates how distant the constituents of a given jet can be. Moreover, he/she can decide whether the algorithm should be exclusive or inclusive (`exclusive_id`), *i.e.* whether the leptons and photons originating from hadron decays have to be included in their respective collections in addition to be considered as constituents of the reconstructed jets (`exclusive_id = false`), or not (`exclusive_id = true`).

Furthermore, the generalised k_T algorithm involves a distance measure depending on a continuous parameter p (`p`),

³ The deprecated STDHEP format is still supported in MADANALYSIS 5 and can possibly be used as well.

⁴ The $R = 1$ default choice is a remnant of the early days of the programme.

⁵ While other cone algorithms are interfaced, they are not infrared-safe and will therefore not be further discussed. We refer to Ref. [9] for more information.

and the *siscone* algorithm depends on the fraction of overlapping momentum above which two protojets are combined (*overlap*), on the maximum number of passes the algorithm should be carried out (*npassmax*), and on a threshold driving the removal of too soft objects (*input_ptmin*). Those protojets, on which jet reconstruction relies in general, are either final-state hadrons or objects that have been already combined. For all algorithms, the combination process obeys to the *E*-scheme [5], *i.e.* the momentum of the combined object equals the sum of the initial momenta.

In practice, the code starts by filtering the input particles (as provided in the event record), restraining the analyses list to all visible final-state particles. This corresponds to the entire set of objects not explicitly tagged as invisible. The tagging of any object as invisible can be achieved by typing in

```
define invisible = invisible <pdg-code>
```

where the *<pdg-code>* value corresponds to the Particle Data Group (PDG) identifier [19] of a new invisible state. The above command results in adding this new code to the list of invisible particles stored in the *invisible* container, which includes by default the Standard Model neutrinos and antineutrinos, as well as the supersymmetric lightest neutralino and gravitino.

Similarly, the user can inform the code about the existence of a new strongly-interacting particle, which hence participates to the hadronic activity in the event and has to be accounted for by the clustering algorithm. The information is provided by superseding the definition of the *hadronic* container,

```
define hadronic = hadronic <new-pdg-code>
```

After having defined the characteristics of the reconstruction, event files have to be imported, either one by one or simultaneously by using wildcards. This is achieved through the standard command

```
import <path-to-hepmc-files> as <set>
```

where the user-defined label *<set>* allows one to group several event files (assumed to describe the same physics process) into a single set. This line can be repeated as much as needed. The reconstructed events are saved on disk and stored in an event file encoded in the LHE event format [20,21] by typing, in the interpreter,

```
set main.outputfile = <name-of-an-LHE-file>
```

where *<name-of-an-LHE-file>* is a filename carrying an *.lhe* or *.lhe.gz* extension. This file will be created in the *<wdir>/Output/SAF/<set>/lheEvents0_0* directory during the MADANALYSIS 5 run (*<wdir>* denoting the run working directory).

Table 2 Properties defining the crude *b*-tagging algorithm available in older versions of MADANALYSIS 5, shown together with their default values. These properties can be modified by typing in the interpreter `set main.fastsim.bjet_id.<property> = <value>`

Properties	Default value	Type
Matching_dr	0.5	Float
Efficiency	1	Float in [0, 1]
Exclusive	True	Boolean
misid_cjet	0	Float in [0, 1]
misid_ljet	0	Float in [0, 1]

The clustering is finally started after the

```
submit
```

command is entered. This results in the automated generation of a C++ code representing the defined reconstruction process, its compilation and its execution on the input event sample(s).

2.2 Crude detector simulation

Already in its earlier versions (*i.e.* without the novelties introduced in this paper), MADANALYSIS 5 gives the option to include basic detector effects in the reconstruction process, such as simple tagging and mistagging efficiencies.

The platform first allows for the implementation of a simple *b*-tagging procedure, relying on flat efficiencies and mistagging rates. The decision behind the (mis)tagging of any given jet involves the angular distance between the reconstructed object and a true *B*-hadron (as read from the input event). Each property defining this procedure is entered in the interpreter as follows,

```
set main.fastsim.bjet_id.<property> = <value>
```

their list and default values being given in Table 2. Behind the scenes, the algorithm matches each *B*-hadron present in the analysed event to all reconstructed jets lying at an angular distance smaller than a threshold fixed by the user (*matching_dr*). Those jets are then considered as *b*-tagged with a probability entered by the user (*efficiency*). *b*-jet identification can be restricted to the closest jet (*exclusive* = *true*), or to all matched jets (*exclusive* = *false*). The mistagging of charm and light jets as *b*-jets is performed similarly, the decision being taken following flat probabilities that are provided by the user (*misid_cjet* and *misid_ljet* respectively).

Next, a simple tau identification procedure can be employed, its properties being set by typing, in the interpreter,

```
set main.fastsim.tau_id.<property> = <value>
```

The list of available options, together with their default values, is given in Table 3. The user has the possibility to fix

Table 3 Properties defining the crude tau-tagging algorithm available in older versions of MADANALYSIS 5, shown together with their default values. These properties can be modified by typing in the interpreter `set main.fastsim.tau_id.<property> = <value>`

Properties	Default value	Type
Efficiency	1	Float in [0, 1]
misid_ljet	0	Float in [0, 1]

the probability with which a jet originating from the hadronic decay of a tau lepton will be correctly tagged as a hadronic tau object (`efficiency`), as well as the probability with which a light jet will be incorrectly tagged as a hadronic tau (`misid_ljet`).

3 A simplified and realistic fast detector simulator

3.1 Generalities

In the present work, we have improved the way in which detector effects can be included when the interface of MADANALYSIS 5 with FASTJET is used for event reconstruction. Our modifications allow for the post-processing of the FASTJET output to model detector effects on the basis of transfer functions. Those functions are provided by the user directly in the PYTHON interpreter and could depend on various object kinematical properties. When the C++ code is generated, the transfer functions are converted into a new C++ module that is called at the end of the FASTJET run. This enables the modification of the properties of the reconstructed objects according to various experimental biases. In contrast to relying on a complex programme like DELPHES 3 that acts at the hadronic level, our setup acts at the time of the reconstruction, which results in a faster event reconstruction process and lighter output files.

We consider three classes of effects. Firstly, the kinematical properties of any given reconstructed objects could be smeared to account for the detector resolution. Secondly, each object has a given probability of being effectively reconstructed, depending on its kinematics. Finally, object identification can be more or less successful, depending again on the kinematics, and leads to a potential misidentification. Whilst the latter effects could already be accounted for in the previous version of the code (see Sect. 2.2), it was not possible to include any dependence on the object kinematics. This limitation has been alleviated.

In order to handle functions at the level of the command-line interface, we rely on abstract syntax trees to decode the information provided by the user and store it internally. Those trees can then be converted into C++ (or any other programming language) and merged with the code generated by the MADANALYSIS 5 interpreter, that is then compiled and run

on the input events. This allows for a very flexible definition of any transfer function. The latter is allowed to depend on any of the observables supported by MADANALYSIS 5, the complete list of them being available from the manual [7] or the normal mode reference card (see App. A of Ref. [9]). Moreover, they can involve a variety of standard mathematical functions like trigonometric, cyclometric or hyperbolic functions.

In the rest of this section, we discuss reconstruction efficiencies, smearing and object identification in Sects. 3.2, 3.3 and 3.4 respectively. The main commands to be typed in the command-line interface are collected in Table 4. At any time, the user can display the currently implemented detector simulator modules by means of the `display` command (see Table 4).

As the event file import, the generation of the working directory and the execution of the code are unchanged, we refer to Sect. 2.1. Details on the usage of the MADANALYSIS 5 simplified fast detector simulation in the expert mode of the programme are provided in Sect. 3.5, and Sect. 3.6 describes how to make use of the CMS and ATLAS detector parametrisation built in the MADANALYSIS 5 platform.

3.2 Reconstruction efficiencies

The granularity of a typical high-energy physics detector, together with the lack of precision in its data acquisition system, implies that it is not always possible to fully reconstruct every single object that leaves hits in it. This can be embedded in the MADANALYSIS 5 machinery by defining reconstruction efficiencies from the interpreter. The code will then generate on run time a probability distribution indicating whether an object should be reconstructed, according to its properties.

This is achieved by means of the `define` keyword,

```
define reco_efficiency <obj> <func> [<dom>]
```

the first argument (`reco_efficiency`) indicating that one deals with the definition of a new reconstruction efficiency. In the above syntax, `<obj>` stands for the object under consideration, `<func>` for the functional form of the efficiency and `<dom>` consists in an optional attribute relevant for piecewise functions. In the latter case, it is indeed necessary to define, through (in)equalities, the domain of application of each piece of the full function. In the case where the user would input an ill-defined function for which the different kinematics subdomains are not disjoint (which is not recommended), the code considers a weighted sum of the corresponding efficiency functions. In addition, if this domain `<dom>` is not provided, the efficiency is understood as applicable over the entire kinematical regime.

Table 4 The subroutines allowing for the definition of a simplified fast detector simulation in MADANALYSIS 5. Reconstruction efficiencies, smearing and object identification are further detailed in Sects. 3.2, 3.3 and 3.4 respectively

<code>define reco_efficiency <obj> <func> [<dom>]</code>
Defines the efficiency associated with the reconstruction of an object <obj>. The efficiency is provided as a piecewise function whose each component <func> has an optional domain of definition <dom>.
<code>define smearer <obj> with <obs> <func> [<dom>]</code>
Defines the standard deviation of a Gaussian of vanishing mean that is relevant for the smearing of the property <obs> of the object <obj>. The width of the Gaussian is provided as a piecewise function whose each component <func> has an optional domain of definition <dom>.
<code>set main.fastsim.jetrecomode = <value></code>
Allows to switch between jet-based jet smearing (<value> = jets, default) and constituent-based jet smearing (<value> = constituents).
<code>define tagger <obj> as <reco> <func> [<dom>]</code>
Defines the efficiency of tagging the Monte Carlo truth object <obj> as a reconstructed <reco> object. The efficiency is provided as a piecewise function whose each component <func> has an optional domain of definition <dom>.
<code>display reco_efficiency</code>
<code>display smearer</code>
<code>display tagger</code>
Displays the different modules of the implemented fast detector simulation.

The list of available reconstructed objects is given in Table 5 and can be referred to either through a dedicated label, or through a so-called ‘generalised PDG code’. The latter extends the traditional usage of PDG codes in high-energy physics software in the sense that the code refers here to reconstructed objects instead of the corresponding Standard Model particles. As shown in the table, reconstruction efficiencies can be defined for jets (j), hadronic taus (ta), electrons (e), muons (mu) and photons (a). At this stage of the detector simulation, the distinction between heavy-flavour and light jets has not been implemented. This is left for the particle identification module (see Sect. 3.4). Moreover, those object definitions always refer to objects that are not originating from hadronic decay processes. Any particle that would be related to a hadronic decay is instead used as input for jet clustering.

The function representing the efficiency and its corresponding domain of application (if relevant) have to be provided as valid formulas in PYTHON. They can involve any observable supported by the MADANALYSIS 5 interpreter, the mostly relevant ones being the transverse momentum p_T (PT), the x , y and z components of the momentum (PX, PY and PZ), the pseudo-rapidity η (ETA) or its absolute value $|\eta|$ (ABSETA), the energy E (E), the transverse energy E_T (ET) and the azimuthal angle $\varphi \in [0, 2\pi[$ (PHI). All these properties are collected in Table 6.

For instance, in the following (toy) snippet of code, we define a flat photon reconstruction efficiency of 99% provided that the photon energy E is larger than 2 GeV and its pseudo-rapidity satisfies $|\eta| \leq 0.88$. When the energy $E > 2$ GeV and $0.88 < |\eta| < 3$, the efficiency is of 98%, and it finally vanishes otherwise.

```
define reco_efficiency a 0 \
```

Table 5 List of reconstructed objects supported by the code. They are given together with their generalised PDG code and the corresponding MADANALYSIS 5 label through which they can be referred to in the code. b -jets and c -jets cannot be used for reconstruction efficiencies and smearing

Object	PDG code	Label
Electron	11	e
Muon	13	mu
Tau	15	ta
Jet	21	j
Photon	22	a
c -jet	4	c
b -jet	5	b

```
[E<=2. or ABSETA>=3.]
define reco_efficiency a 0.99 \
  [E>2. and ABSETA<=0.88]
define reco_efficiency a 0.98 \
  [E>2. and ABSETA>0.88 and ABSETA<3.]
```

As can be seen, the usage of the `or` and `and` keywords is supported (and needed) to define the different parts of the domain of definition of the piecewise function.

3.3 Smearing

The impact of the detector resolution is performed via the smearing formalism. The particle momenta, as returned by the Monte Carlo simulations, are smeared by MADANALYSIS 5 on the basis of Gaussian functions of vanishing mean and a width provided by the user.

The standard deviation of those Gaussian functions σ directly depends on the object properties. In the following, we

Table 6 Main object properties entering the different functions relevant for a simplified fast detector simulation in MADANALYSIS 5. The number of tracks n_{tr} can only be used for jet and hadronic tau (mis)tagging, when defining the domain of application of the corresponding efficiency piecewise function. The complete list of observables can be obtained from the manual [7] or the normal mode reference card (see App. A of Ref. [9])

Properties	Symbol	Label
x momentum component	p_x	PX
y momentum component	p_y	PY
z momentum component	p_z	PZ
Transverse momentum	p_T	PT
Transverse energy	E_T	ET
Energy	E	E
Pseudo-rapidity	η (or $ \eta $)	ETA or (ABSETA)
Azimuthal angle (in $[0, 2\pi]$)	φ	PHI
Number of tracks	n_{tr}	NTRACKS

rely on the example of energy smearing for calorimeter-based quantities. The discussion can however be straightforwardly adapted to any other class of energy-momentum smearing. In this example, energy smearing is often parametrised as originating from three distinct components [22],

$$\frac{\sigma(E, \eta, \varphi)}{E} = \frac{N(\eta, \varphi)}{E} \oplus \frac{S(\eta, \varphi)}{\sqrt{E}} \oplus C(\eta, \varphi) \quad (1)$$

$$= \sqrt{\frac{N^2(\eta, \varphi)}{E^2} + \frac{S^2(\eta, \varphi)}{E} + C^2(\eta, \varphi)}.$$

In this expression, N (in GeV) corresponds to the so-called noise term describing both the imperfections in the readout electronics, and the fluctuations arising from the simultaneous energy deposits of uncorrelated pile-up jets. This component of the resolution dominates for low-energy objects. The second term, S (in $\sqrt{\text{GeV}}$), represents the stochastic contribution related to statistical random fluctuations in the physical evolution of the shower in the detector, whereas the energy-independent last term C (therefore more relevant for high-energy objects) is associated with imperfections in the calorimeter geometry, anomalies in signal collection uniformities, as well as with inter-calibration errors and fluctuations in the longitudinal energy content.

Several examples for the energy resolution dependence on the object energy are shown in Fig. 1, for three set of noise, stochastic and constant factor values. We first consider a case in which the resolution is essentially dominated by its noise and stochastic terms (blue), so that it is especially poor in the low-energy regime. We then focus on a setup on which the noise factor is negligible with respect to a constant contribution (red), resulting to a quite similar overall resolution. Finally, a more realistic situation (green) in which all three

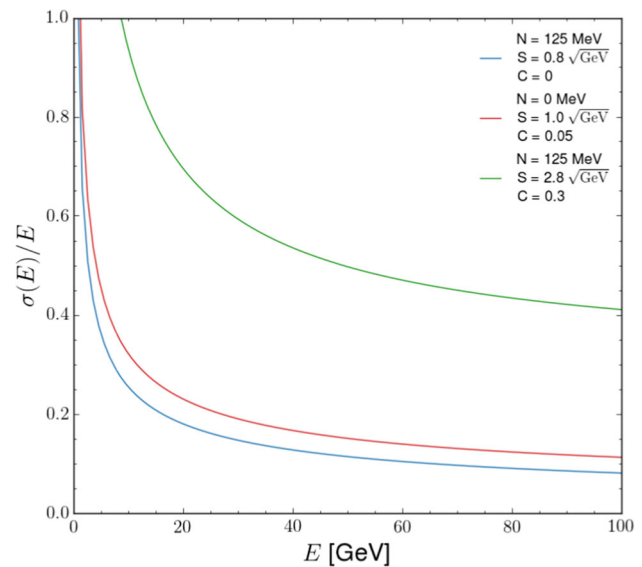


Fig. 1 Energy resolution, given as a function of the energy, for a few configurations of the noise, stochastic and constant factors introduced in Eq. (1)

components are contributing is presented, the total resolution being globally worse over the entire energy range.

In all those examples, the energy E is smeared by a quantity distributed according to a Gaussian of vanishing mean, and of standard deviation given by Eq. (1),

$$E \rightarrow E' = E + \mathcal{N}_\varepsilon[E; 0, \sigma(E, \eta, \varphi)]. \quad (2)$$

The smeared energy E' is in addition enforced to be positive and the quantity $\mathcal{N}_\varepsilon$ is a random parameter sampled according to the standard normal distribution $\mathcal{N}$,

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]. \quad (3)$$

Smearing can be enabled in MADANALYSIS 5 through the `define` command, that takes as a first argument the `smeared` keyword indicating that a smearer is being defined. The complete syntax reads

```
define smeared <obj> with <obs> <func> [<dom>]
```

The object whose properties have to be smeared is denoted by `<obj>` and has to be provided either through its generalised PDG code, or its label (see Table 5). The property that has to be smeared is provided through the `<obs>` argument, the list of observables available being shown in Table 6. Finally, the function `<func>` (and its corresponding domain of application `<dom>`) represents the resolution σ , that can be provided as a piecewise function depending on any of the object properties.

After the smearing of some object properties, the missing energy is always recalculated accordingly.

As an example, we present below a snippet of code that could be relevant for the smearing of the jet energy at a typical LHC detector [23],

```
define smearer j with E \
  sqrt(E^2*0.05^2 + E*1.5^2) \
  [ABSETA <= 3]
define smearer j with E \
  sqrt(E^2*0.13^2 + E*2.7^2) \
  [ABSETA > 3 and ABSETA <= 5]
```

the piecewise function being only defined for cases in which the jets can effectively be reconstructed. We indeed implicitly assume that the jet reconstruction efficiency vanishes anywhere outside the considered domain. The resolution $\sigma(E)$ corresponds to

$$\sigma(E) = \begin{cases} 0.05 \oplus 1.5\sqrt{E} & \text{for } |\eta| \leq 3, \\ 0.13 \oplus 2.7\sqrt{E} & \text{for } 3 < |\eta| \leq 5. \end{cases} \quad (4)$$

Whilst the methods that are presented above hold for any class of object, jet smearing can be implemented in a second manner. Instead of smearing the properties of each reconstructed jet, treating it as a whole, the code offers the option to smear instead the properties of each constituent of the jet. The resulting jet four-momentum is then evaluated in a second step. For most cases, a jet-based smearing is sufficient to emulate most detector environments and there is no need to rely on any constituent-based smearing. However, this by definition renders the reconstruction process blind to any hadronic effect occurring inside a jet. The latter may be relevant for very specific studies, like when jet substructure is in order [6]. In this case, hard QCD radiation can for instance activate detector cells around those defining the jet and leads to the presence of two overlapping jets sharing the energy deposits of the surrounding cells. Such an effect can be covered by constituent-based jet smearing.

In order to activate constituent-based or jet-based jet smearing, it is sufficient to type, in the MADANALYSIS 5 command-line interface,

```
set main.fastsim.jetrecomode = <val>
```

where <val> can take either the jets value (default), or the constituents value.

3.4 Object identification - taggers

In its simplified detector simulation, MADANALYSIS 5 allows the user to input a large set of tagging efficiencies and related mistagging rates. The list of available pairs of reconstructed and truth-level objects is shown in Table 7.

Jets can be (correctly and incorrectly) tagged as *b*-jet, *c*-jet or lighter jets. Both light jets and true hadronic taus can be identified as taus. In addition, jets faking electrons and photons, as well as electrons, muons and photons faking each

other can be implemented. While the efficiencies relevant for *b*-tagging and tau-tagging can be provided as well (those consist in ‘taggers’ strictly speaking), the situation is slightly different for electrons, muons and photons. Here, the tagging efficiencies simply consist in the corresponding reconstruction efficiencies (see section 3.2) and should be implemented as such. The efficiency for correctly identifying a jet as a jet, a photon as a photon, an electron as an electron and a muon as a muon cannot thus be implemented as taggers, and the user has to enter them by means of the definition of reconstruction efficiencies.

A (mis)tager can be defined by typing, in the command-line interface,

```
define tagger <true> as <reco> <func> [<dom>]
```

Once again, this relies on the `define` command, that takes as a first argument the keyword `tagger` indicating to the code that a tagger is being defined. The keyword `<true>` represents the label of the truth-level object (see Table 5) that can be tagged as a reconstructed object denoted by `<reco>` (see Table 5 as well). As in the previous sections, the efficiency is provided as the function `<func>`, to be given as a valid PYTHON formula, and `<dom>` consists in its domain of application that is relevant when the (mis)tagging efficiency is a piecewise function. For any tagger of a jet or a hadronic tau as any given object, it is possible to make use of the number of tracks associated with the truth-level object to define the domain of application of the different elements of the piecewise function `<func>`.

The definition of a tagger leads to the generation of a probability distribution that indicates, at the time that the C++ code is executed, whether a reconstructed object (a photon, a jet, *etc.*) or one of its properties (*b*-tag, *c*-tag, *etc.*) has to be modified.

As an illustrative example, *b*-tagging performances that are typical of an LHC detector could be entered as

```
define tagger b as b \
  tanh(PT/400) * (85 / (4 + 0.252 * PT)) \
  [ABSETA <= 2.5]
define tagger b as b 0 [ABSETA > 2.5]
```

Table 7 List of tagging efficiencies and mistagging rates that are supported by the code

Reconstructed object	Truth object
<i>b</i> -jet	<i>b</i> -jet, <i>c</i> -jet, light jet
<i>c</i> -jet	<i>b</i> -jet, <i>c</i> -jet, light jet
Hadronic tau	Hadronic tau, light jet
Photon	Light jet, electron, muon
Electron	Light jet, muon, photon
Muon	Electron, photon

```
define tagger c as b \
  tanh(0.018*PT) * (1 / (4 + 0.0052*PT))
```

```
define tagger j as b 0.01 + 3.8e-05*PT
```

In this snippet of code, the b -tagging efficiency $\varepsilon_{b|b}$ is parametrised by [24]

$$\varepsilon_{b|b}(p_T) = \tanh[p_T/400] \frac{85}{4 + 0.252 p_T} \Theta[2.5 - |\eta|], \quad (5)$$

whilst the mistagging rate of a charmed jet ($\varepsilon_{b|c}$) as a b -jet and the one of a light jet as a b -jet ($\varepsilon_{b|j}$) are given by

$$\varepsilon_{b|c}(p_T) = \tanh[0.018 p_T] \frac{1}{4 + 0.0052 p_T}, \quad (6)$$

$$\varepsilon_{b|j}(p_T) = 0.01 + 3.8 \cdot 10^{-5} p_T.$$

3.5 Expert mode

Whereas the metalanguage defining the MADANALYSIS 5 interpreter is rich, it is limited by its own definition. The user can circumvent this inherent limitation by using the platform in its so-called expert mode. Analyses are here implemented directly in C++, bypassing the MADANALYSIS 5 command-line interface. In this way, the user can benefit from all capabilities of the programme (readers, observables, *etc.*), and focus on implementing only the non-standard routines necessary for his/her own purpose.

The analysis skeleton generated automatically when the expert mode is switched on can incorporate a simplified detector simulation as defined in the beginning of this section. To this aim, it is sufficient to initiate MADANALYSIS 5 by typing in a shell,

```
./bin/ma5 -Re <wdir> <label> <sfs>
```

where $\langle wdir \rangle$ stands for the working directory in which the analysis template is generated, $\langle label \rangle$ refers to the analysis name that is used throughout the entire analysis, and $\langle sfs \rangle$ is the (optional) configuration file, written as a set of normal MADANALYSIS 5 commands, defining how the simplified fast detector simulation should be run. Providing such a file is optional, so that if it is absent, the code runs as in previous versions of the programme. We refer to the manual [8] of the expert mode and Ref. [9] for more information.

3.6 Standard LHC detector parametrisation

MADANALYSIS 5 is shipped with two predefined detector parametrisations, that are respectively related to the ATLAS and CMS detectors. Those cards are available from the `madanalysis/input` subdirectory, and have been validated through a comparison with the standard ATLAS and CMS DELPHES 3 detector configuration files. The example of the CMS validation is presented in Sect. 4.

In order to load those cards when running MADANALYSIS 5, the user has to start the code by providing the path to the detector card as an argument of the `bin/ma5` command. This would give, from a shell,

```
./bin/ma5 -R madanalysis/input/<card>.ma5
```

where $\langle card \rangle$ has to be replaced by `CMS_default` and `ATLAS_default` for the CMS and ATLAS detector parametrisations respectively.

4 Comparison with DELPHES 3 and the Monte Carlo truth

In this section, we validate the implementation of our simplified fast detector simulation in MADANALYSIS 5. To this aim, we perform a comparison between predictions relying on our module and predictions relying instead on DELPHES 3, for a variety of Standard Model processes. Both results are moreover confronted to the expectation of the Monte-Carlo truth, where events are reconstructed as such (without any smearing and tagging). The general design of our simulation framework is depicted in Sect. 4.1, whilst Sects. 4.2, 4.3, 4.4 and 4.5 focus on jet, lepton, hadronic tau and photon properties respectively.

4.1 Simulation framework

In order to validate our implementation, we generate several samples of Monte Carlo events describing various Standard Model processes relevant for proton-proton collisions at a centre-of-mass energy of 14 TeV. Hard scattering events are produced with MG5_AMC [25] (version 2.7.3), that we use to convolute leading-order matrix elements with the leading-order set of NNPDF 2.3 parton densities [26]. Those events are matched with parton showering as modelled by the PYTHIA 8 package [27] (version 8.244), that is also employed to simulate hadronisation. As we focus on Standard Model processes, the typical energy scales under consideration are not so hard, ranging between a few tens of GeV to 100–200 GeV. Section 5 will be instead dedicated to new physics, and will thus be relevant for a study of the features of our fast detector simulation framework for much larger scales.

In our comparison, we include detector effects in four different ways. First, we consider an ideal detector, or the so-called Monte Carlo truth. In practice, reconstruction is performed as described in Sect. 2, with all detector effects being switched off. We rely on the anti- k_T algorithm with a radius parameter set to $R = 0.4$, as implemented in FASTJET (version 3.3.3). Second, we make use of the complex detector machinery as implemented in the DELPHES 3 package (version 3.4.2), using the standard CMS detector parametrisation shipped with the programme. In particular, this includes

object isolation and enforces the storage of all generator-level objects (as for the SFS for what concerns the hard-scattering process) and calorimetric and tracking information. Finally, we consider the simplified fast detector simulation presented in this work, that we run once with a jet-based jet smearing and once with a constituent-based jet smearing. The detector parametrisation has been designed such that it matches the standard CMS card from DELPHES 3.

By definition, object isolation cannot be implemented directly within the SFS framework, so that it needs to be performed at the analysis level. In this paper, we provide one example of a way to handle this, which relies on a simple method employing ΔR -based object separation. We consider two objects as overlapping if they lie at a distance ΔR in the transverse plane that is smaller than some threshold. An overlap removal procedure is then implemented, one of the two objects being removed. The details about which object is kept and which object is removed depend on what the user has in mind. We recall that in the entire SFS framework, those objects always refer to objects that are not originating from any hadronic decay processes, those latter objects being instead clustered into jets.

In Sects. 4.2, 4.4 and 4.5, we focus on jets, taus and photons. We therefore first remove from the jet collection any jet that would be too close to an electron. We then remove from the electron, muon and photon collections any electron, muon and photon that would be too close to any of the remaining jets. Finally, taus are required to be well separated from any jet. This strategy is similar to what is done, *e.g.* in the ATLAS analysis of Ref. [28], that examines the properties of hadronic objects. In Sect. 4.3, we focus instead on leptons and therefore implement a different isolation methodology inspired by the ATLAS analysis of Ref. [29]. Here, we firstly remove any lepton that would be too close to a jet, before secondly removing any jet that would be too close to any of the remaining leptons. More information about object isolation is provided, specifically for the different cases under study, in Sects. 4.2, 4.3, 4.4 and 4.5. It is important to bare in mind that this crude approximation (relatively to the DELPHES 3 capabilities) for isolation is sufficient in many useful physics cases, as testified by the results shown in Sect. 5 in the context of LHC recasting. Depending on what the user aims to do, this may however be insufficient.

As mentioned in the introduction to this paper, our new module has the advantage on DELPHES 3 to be cheaper in terms of CPU costs when its default configuration is compared with the default ones implemented in DELPHES 3 (that many users employ). This is illustrated in Table 8 for the reconstruction of different numbers of top-antitop events N_{events} . Such a process leads to hadron-level events including each about 1670 objects, that thus consist of the amount of inputs to be processed at the time of the simulation of the detector effects. It is interesting to note that in the context of

Table 8 Comparison of the reconstruction of N_{events} top-antitop events with DELPHES 3, using the built-in CMS detector parametrisation, and the simplified fast detector simulation introduced in this work. We consider jet smearing based either on the reconstructed jets themselves (SFS [Jets]) or on their constituents (SFS [Const.]). The results depict the size of the output file when the results are compressed (top panel) and the time needed for the run (lower panel), as obtained by using a machine with an Intel(R) Xeon(R) Core E3-1271v3 CPU with a 3.6 GHz clock speed. The time necessary for the input/output operations is negligible, so that the provided results can be considered as related to the detector simulation process only. In the upper table, we also provide information, included in parentheses, on the output size corresponding to a DELPHES 3 parametrisation in which the calorimetric, track, jet substructure and generator-level information is not written to the file

N_{events}	DELPHES 3	SFS [Jets]	SFS [Const.]
10,000	821.0 Mb (8.7 Mb)	6.7 Mb	6.7 Mb
50,000	4.1 Gb (43 Mb)	34.0 Mb	34.0 Mb
100,000	8.1 Gb (84 Mb)	67.0 Mb	68.0 Mb
200,000	17 Gb (166 Mb)	133 Mb	134 Mb
N_{events}	DELPHES 3	SFS [Jets]	SFS [Const.]
10,000	3 min 24 s	2 min 11 s	2 min 17 s
50,000	17 min 15 s	10 min 24 s	11 min 02 s
100,000	33 min 21 s	20 min 41 s	22 min 20 s
200,000	1 h 07 min	42 min 07 s	43 min 55 s

LHC recasting (as discussed in Sect. 5), the speed difference often reaches a factor of 2. This extra gain originates from the structure of the PAD itself. When relying on the SFS framework for the simulation of the detector, the analysis and the detector simulation can be performed directly, without having to rely on writing the reconstructed events on a file. In contrast, when relying on DELPHES 3 for the simulation of the detector, an intermediate ROOT file is first generated by DELPHES 3 and then read again at the time of the analysis. This results in a loss of efficiency in terms of speed.

It can also be seen, in Table 8, that regardless of the size of the event sample, our simplified fast detector simulation is up to 30% faster than DELPHES 3. As the time necessary for the input/output operations is negligible, the numbers provided in the lower panel of the table can be seen as what is needed to deal with the simulation of the detector itself. Independently of the event sample size, DELPHES 3 is found to process one event in about 0.0203 s, whilst the SFS framework requires 0.0130 s instead, in average and regardless of the jet smearing configuration. In addition, we have compared the SFS performance with those of DELPHES 3 when relying on a much simpler detector parametrisation that only includes elementary smearing, object reconstruction and identification functionalities. Isolation and tracking are thus turned off, so that we make use of DELPHES 3 in a way that is as close as possible as what the SFS does. We observe time performances

that are slightly closer to the SFS ones, DELPHES 3 requiring here only 0.0189 s to process a single event in average.

In terms of the disk space needed to store the outputted reconstructed events, we have found that DELPHES 3, when used in its default configuration, leads to output files that are about 100 times heavier than for an SFS-based detector simulation. However, most of the disk space is used to store track and calorimetric tower information. As this information is not included in the SFS framework, we investigate how the file sizes change when it is removed from the default CMS parametrisation in DELPHES 3. We also remove from the output file any information related to the generator-level collections and jet substructure. We consequently obtain results similar in the SFS and DELPHES 3 cases, as visible from the upper panel of Table 8.

4.2 Multijet production

In order to investigate the differences in the jet properties that would arise from using different reconstruction methods, we consider hard-scattering di-jet production, $pp \rightarrow jj$, in a five-flavour-number scheme. In our simulation process, we generate 500,000 events and impose, at the generator level, that each jet features a transverse momentum of at least 20 GeV and a pseudo-rapidity smaller than 5. Moreover, the invariant mass of the di-jet system is enforced to be larger than 100 GeV.

In our simplified fast detector simulation (SFS), the energy E of the jets is smeared according to the transfer functions provided in Ref. [23]. The resolution on the jet transverse momentum $\sigma(E)$ is given by Eq. (4). Moreover, we implement the b -tagging performance defined by Eqs. (5) and (6), keeping in mind that b -jet identification is ineffective for $|\eta| > 2.5$, by virtue of the CMS tracker geometry. Finally, we include the following jet reconstruction efficiency $\varepsilon_j(\eta)$, that depends on the jet pseudo-rapidity,

$$\varepsilon_j(\eta) = \begin{cases} 92.5\% & \text{for } |\eta| \leq 1.5, \\ 87.5\% & \text{for } 1.5 < |\eta| \leq 2.5, \\ 80\% & \text{for } |\eta| > 2.5. \end{cases} \quad (7)$$

This allows for the simulation of most tracker effects on the jets, with the limitation that charged and neutral jet components are equally considered. Our detector simulation also impacts all the other objects potentially present in the final state. However, we refer to the next sections for details on the simulation of the corresponding detector effects.

At the analysis level, we select as jet and lepton candidates those jets and leptons with a transverse momentum greater than 20 GeV and 10 GeV respectively. We moreover impose simple isolation requirements by ignoring any jet lying at an angular distance smaller than 0.2 of any reconstructed electron ($\Delta R_{ej} < 0.2$), and then ignore any lepton (electron

or muon) lying at a distance smaller than 0.4 of any of the remaining jets ($\Delta R_{\ell j} < 0.4$).

In Fig. 2, we present and compare various distributions when an ideal detector is considered (filled areas, truth), when a CMS-like detector is considered and modelled in DELPHES 3 (solid) and when the simplified fast detector simulation (SFS) introduced in this work, parametrised to model a CMS-like detector and run both when jet-based (dotted) and constituent-based (dashed) jet smearing is switched on. Consequently to the simple jet smearing studied in this section, there is no need to implement different efficiencies for constituent-based and jet-based smearing. However, this is not the case anymore for investigations relying, for example, on the jet spatial resolution like when jet substructure is involved.

In the upper line of the figure, we begin with studying the distribution in the number of jets N_j (upper left) and b -jets (upper right). As can be noticed by inspecting the N_j spectrum, the impact of the detector is especially important for the 0-jet bin, as well as when the number of jets is large. In this case, deviations from the Monte Carlo truth are the largest, regardless of the way the detector effects are simulated. However, DELPHES 3-based or SFS-based predictions agree quite well, at least when a jet-based smearing is used in the SFS case. A constituent-based smearing indeed yields quite significant differences, and predictions for the large-multiplicity bins are much closer to the Monte Carlo truth. The detector imperfections yield event migration from the higher multiplicity to the lower multiplicity bins independently of the exact details those imperfections are dealt with at the simulation level.

With the exception of the jet multiplicity spectrum, the impact of the used jet smearing method is however quite mild compared to the bulk of the detector effects. This is expected, as those effects are important only in very specific cases (not covered in this work). In the rest of the figure, we demonstrate this by investigating the shape of various differential spectra, and comparing them with predictions derived from the Monte Carlo truth. We study the transverse momentum and pseudo-rapidity spectrum of the leading jet and leading b -jet. All detector simulations agree with each other. The existing differences between DELPHES 3 and the other fast simulation results are found to originate from the reconstruction efficiency (see Eq. (7)). The latter, as implemented in the simplified fast detector simulation of MADANALYSIS 5, are expected to mimic, but only to some extent, the tracker effects included in DELPHES 3 that are much more complex and distinguish charged and neutral hadrons. Those differences however only impact the soft parts of the spectrum.

Finally, we consider a more inclusive variable in Fig. 3, namely the scalar sum of the transverse momentum of all reconstructed jets H_T . We again get a satisfactory level of agreement between the three detector simulations. The dif-

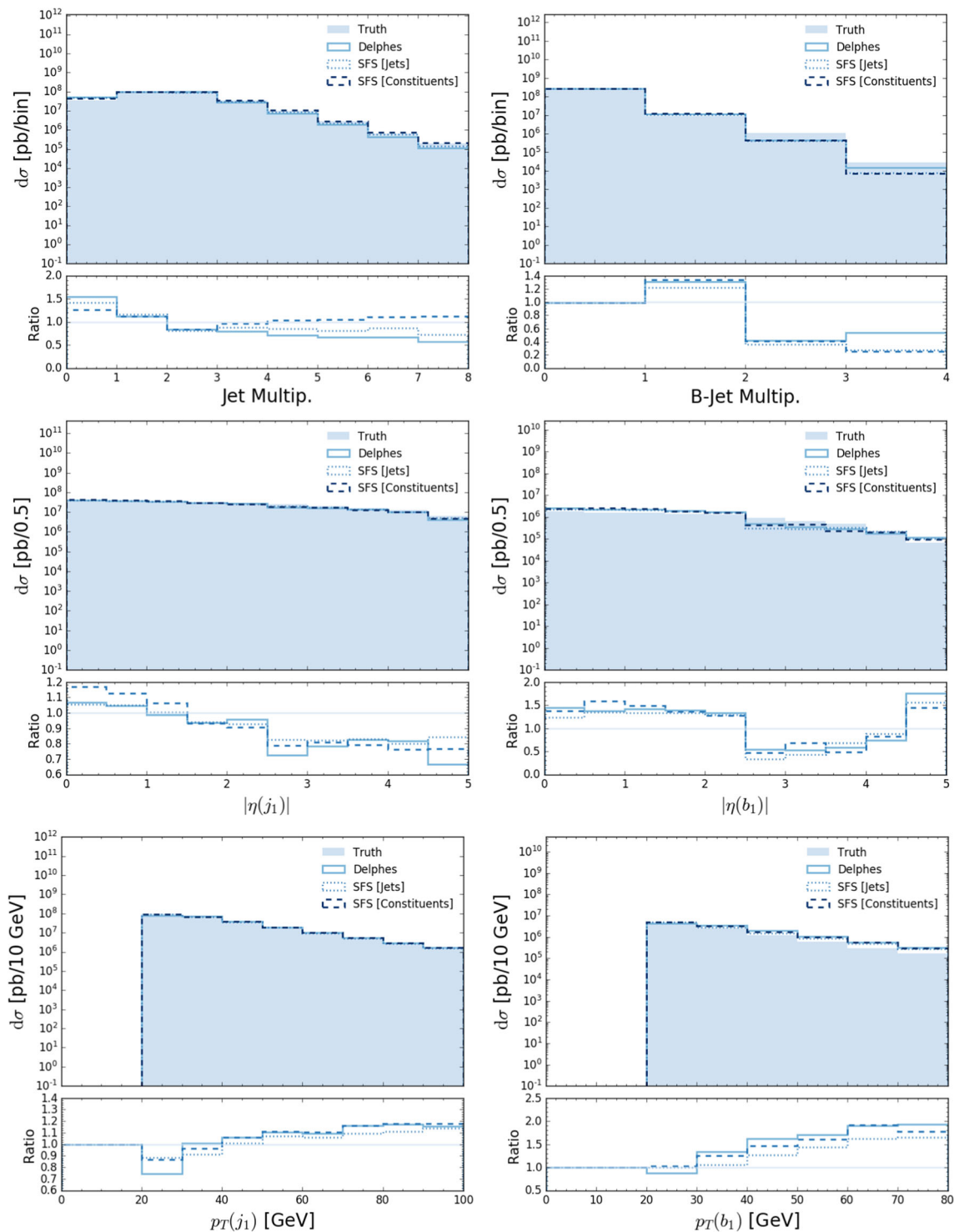


Fig. 2 Jet properties when reconstruction is considered with various options for the detector simulation: an ideal detector (truth, filled area), DELPHES 3 (solid) and the MADANALYSIS 5 simplified fast detector simulation with either jet-based (SFS [Jets], dotted) or constituent-based (SFS [Constituents], dashed) jet smearing. We show distributions in the

jet (upper left) and b -jet (upper right) multiplicity, leading light jet p_T (centre left) and $|\eta|$ (centre right), as well as leading b -jet p_T (lower left) and $|\eta|$ (lower right). In the lower insets, the distributions are normalised to the truth results

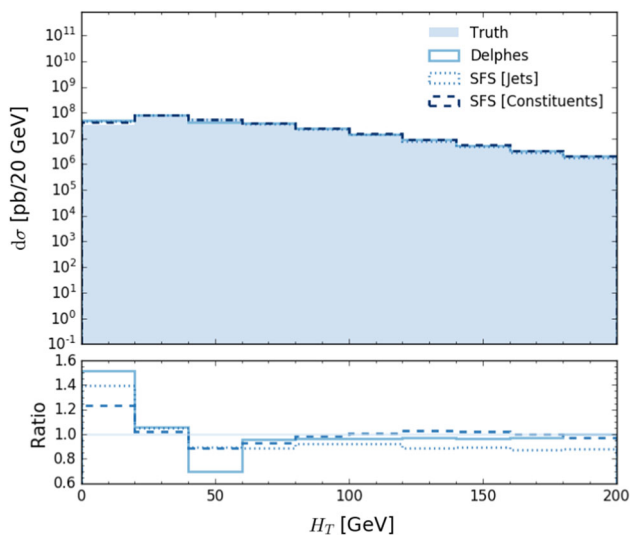


Fig. 3 Same as Fig. 2, but for the H_T spectrum

ferences between DELPHES 3-based and SFS-based predictions only affect the low- H_T bins in which the impact of the softest objects, that are most likely to be sensitive to the different treatment of the detector simulation, is the largest.

4.3 Electrons and muons

In this section, we perform on a comparison of electron and muon properties when these are reconstructed from the different methods considered in this work. To this aim, we generate a sample of 200,000 neutral-current and charged-current Drell-Yan events, $pp \rightarrow \ell^+\ell^- + \ell^-\bar{\nu}_\ell + \ell^+\nu_\ell$ with $\ell = e, \mu$. In our simulations, we impose, at the Monte Carlo generator level, the invariant mass of a same-flavour opposite-sign lepton pair to be of at least 50 GeV, and we constrain each individual lepton to feature a transverse momentum greater than 10 GeV.

The electron and muon reconstruction efficiencies ε_e and ε_μ are extracted from Ref. [30], and depend both on the electron and muon transverse momentum p_T and pseudo-rapidity η . They respectively read

$$\varepsilon_e(p_T, \eta) = \begin{cases} 0.7 & \text{for } p_T > 10 \text{ GeV and } |\eta| \leq 1.5, \\ 0.525 & \text{for } p_T > 10 \text{ GeV and } 1.5 < |\eta| \leq 2.5, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$\varepsilon_\mu(p_T, \eta) = \begin{cases} 0.891 \mathcal{B}(p_T) & \text{for } p_T > 10 \text{ GeV and } |\eta| \leq 1.5, \\ 0.882 \mathcal{B}(p_T) & \text{for } p_T > 10 \text{ GeV and } 1.5 < |\eta| \leq 2.4, \\ 0 & \text{otherwise,} \end{cases}$$

with

$$\mathcal{B}(p_T) = \Theta \left[1 - \frac{p_T}{\text{TeV}} \right] - e^{\left[\frac{1}{2}(1 - p_T/\text{TeV}) \right]} \Theta \left[\frac{p_T}{\text{TeV}} - 1 \right]. \quad (9)$$

In addition, we model the effects stemming from the electromagnetic calorimeter by smearing the electron energy in

a Gaussian way, with a standard deviation $\sigma_e(E, \eta)$ defined by [30,31]

$$\frac{\sigma_e(E, \eta)}{E} = \begin{cases} \left[1 + 0.64|\eta|^2 \right] \left[\frac{0.4}{E} \oplus \frac{0.11}{\sqrt{E}} \oplus 0.008 \right] & \text{for } |\eta| \leq 1.5 \\ \left[2.16 + 5.6(|\eta| - 2)^2 \right] \left[\frac{0.4}{E} \oplus \frac{0.11}{\sqrt{E}} \oplus 0.008 \right] & \text{for } 1.5 < |\eta| \leq 2.5, \\ \frac{2.08}{\sqrt{E}} \oplus 0.107 & \text{for } 2.5 < |\eta| \leq 5. \end{cases} \quad (10)$$

At the analysis level, we implement similar selections as in Sect. 4.2. We consider as jets and leptons those jets and leptons with $p_T > 20$ GeV and 10 GeV respectively, and require all studied objects to be isolated. Any lepton too close to a jet is discarded ($\Delta R_{\ell j} < 0.2$), and any jet too close to any of the remaining leptons is discarded too ($\Delta R_{\ell j} < 0.4$). Moreover, we impose a selection on the invariant mass of the lepton pair, $m_{\ell\ell} > 55$ GeV, for events featuring at least two reconstructed leptons of opposite electric charges.

In Fig. 4, we compare various lepton-related distributions after using the different considered options for the simulation of the detector effects. We start by considering the distributions in the electron (upper left) and muon (upper right) multiplicity, which show that the loss in leptons relatively to the Monte-Carlo truth is similar for all three detector simulations. This is not surprising as the efficiencies have been tuned accordingly.

For similar reasons, the detector effects impacting the leading electron (centre left) and muon (centre right) pseudo-rapidity distributions are similarly handled in all three setups. Relatively to the Monte-Carlo truth, not a single electron and muon is reconstructed for pseudo-rapidities larger than 2.5 and 2.4 respectively, and the fraction of lost leptons is larger for $|\eta| > 1.5$ than for $|\eta| < 1.5$. This directly stems from the detector geometry and design, that make it impossible to reconstruct any non-central lepton and lead to degraded performance for larger pseudo-rapidities (as implemented in all detector simulator reconstruction efficiencies).

The various options for modelling the detector effects however yield important differences for the leading electron (lower left) and leading muon (lower right) transverse momentum distributions. DELPHES 3 predicts a smaller number of leptons featuring $p_T < 40$ GeV than in the SFS case, and a larger number of leptons with transverse momenta $p_T > 40$ GeV. The difference ranges up to 20% at the kinematical selection threshold of $p_T \sim 10$ GeV, and for the largest p_T bins. This originates from the inner machineries implemented in the various codes. In DELPHES 3, hadron-level objects are first converted into tracks and calorimetric deposits, which involves efficiencies provided by the user. Next, momenta and energies are smeared, before that the

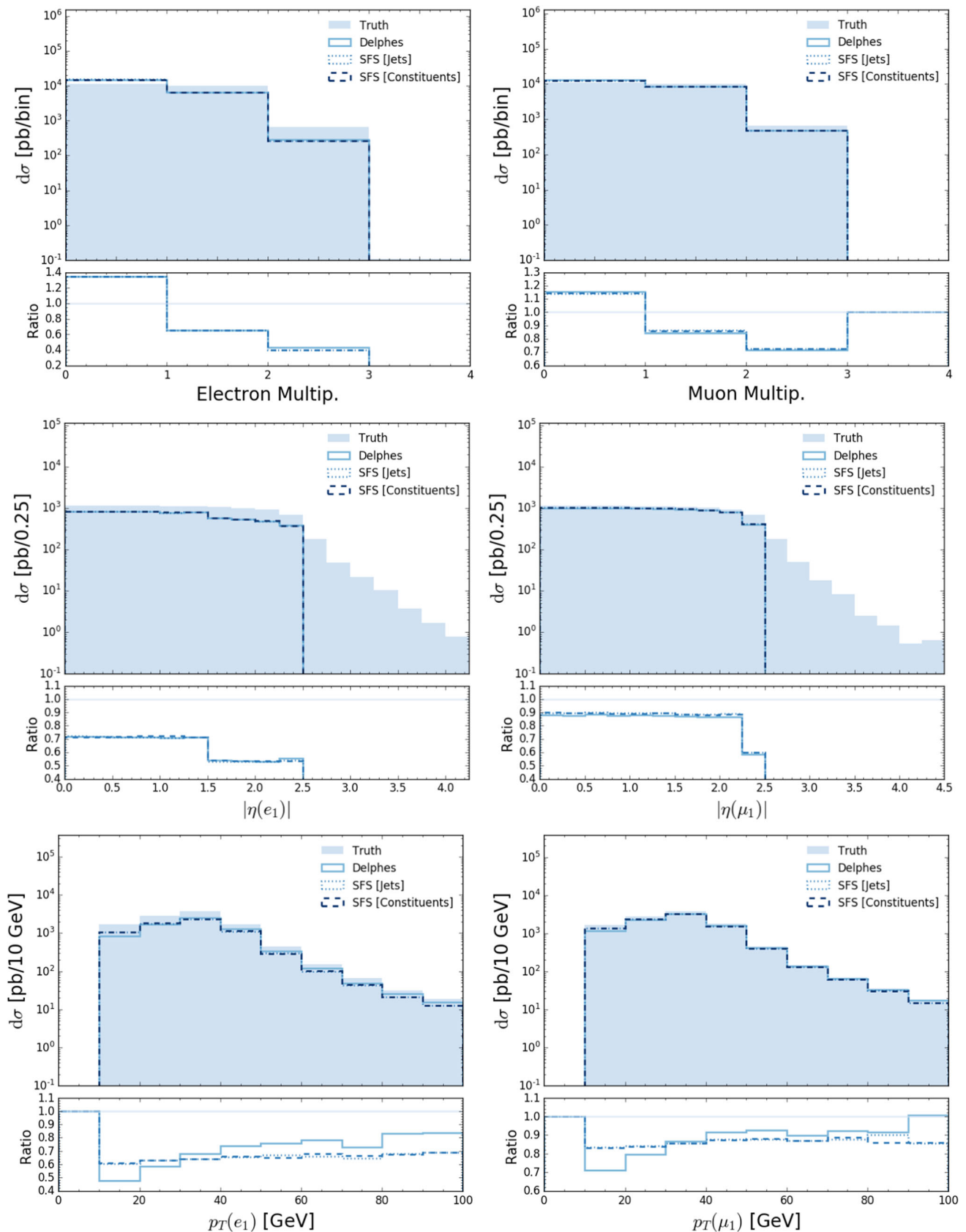


Fig. 4 Same as Fig. 2, but for the electron and muon multiplicity distribution (upper row), and the pseudo-rapidity (central row) and transverse momentum (lower row) spectrum of the leading electron (left) and muon (right)

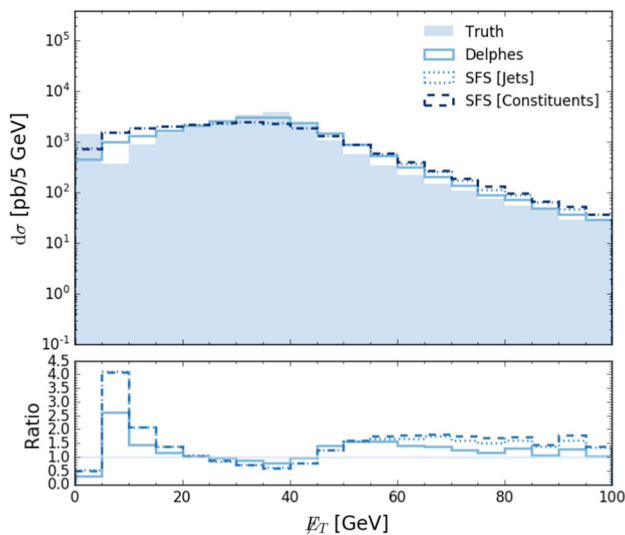


Fig. 5 Same as Fig. 2, but for the missing transverse energy spectrum

code reconstructs the objects to be used at the analysis level. This latter step relies again on user-defined reconstruction efficiencies, this time specific to each class of reconstructed objects. In the MADANALYSIS 5 SFS, such a splitting of the reconstruction efficiency into two components has not been implemented, for the purpose of keeping the detector modelling simple. Such a two-component smearing would however introduce significant shifts in the distributions describing the properties of the reconstructed objects, especially for objects for which tracking effects matter.

This feature consequently impacts any observable, potentially more inclusive, that would depend on the properties of the various leptons. This is illustrated in Fig. 5 where we present the missing transverse energy ($\cancel{E}_T$) spectrum. In this case, the differences are even more pronounced than for the lepton p_T spectra, due to the particle flow method used in DELPHES 3 to reconstruct the missing energy of each event. In contrast, MADANALYSIS 5 only sums vectorially the momentum of all visible reconstructed objects. Most differences occur in the low-energy region of the distribution, that is largely impacted by the differently modelled softer objects, but the effects are also significant for large $\cancel{E}_T$ values.

4.4 Tau pair-production

We now move on with a comparison of the detector performance for the reconstruction of hadronic taus, for which DELPHES 3 and our simplified detector simulation follow quite different methods. DELPHES 3 uses a cone-based algorithm to identify hadronic taus from the jet collection, whereas in the SFS approach, hadronic taus are tagged through the matching of a reconstructed jet with a hadron-

level object. Such a matching imposes that the angular distance, in the transverse plane, between a reconstructed tau and a Monte Carlo hadron-level (and thus un-decayed) tau is smaller than some threshold. Such differences between the DELPHES 3 and SFS approaches can cause large variations in the properties of the reconstructed taus, as shown in the rest of this section.

To quantify the impact of this difference, we produce a sample of 250,000 di-tau events, $pp \rightarrow \tau^+ \tau^-$, that includes a 10 GeV selection on the tau transverse momentum at the generator level, as well as a minimum requirement of 50 GeV on the invariant mass of the di-tau system. In the SFS case, we reconstruct jets, electrons and muons as described in Sects. 4.2 and 4.3. We moreover implement a tau reconstruction efficiency $\varepsilon_{\text{tracks}}$ aiming at reproducing tracker effects,

$$\varepsilon_{\text{tracks}}(p_T) = 0.7 \Theta \left[\frac{p_T}{\text{GeV}} - 20 \right], \quad (11)$$

as well as a tau tagging efficiency $\varepsilon_{\tau|\tau}$ that is identical to the one embedded in the standard DELPHES 3 CMS detector parametrisation, and that is independent of the tau transverse momentum,

$$\varepsilon_{\tau|\tau}(\eta) = 0.6 \Theta [2.5 - |\eta|]. \quad (12)$$

The corresponding mistagging rate of a light jet as a tau $\varepsilon_{\tau|j}$ is flat and independent of the kinematics,

$$\varepsilon_{\tau|j} = 0.01. \quad (13)$$

In addition, the properties of each reconstructed tau object have been smeared as for jets (see Sect. 4.2).

We follow the same analysis strategy as in Sect. 4.2. We first select as jet and lepton candidates those jets and leptons with a transverse momentum greater than 20 GeV and 10 GeV respectively. Then, we remove from the jet collection any jet lying at $\Delta R_{ej} < 0.2$ of any reconstructed electron, and any electron or muon lying at $\Delta R_{\ell j} < 0.4$ from any of the remaining jets. In the SFS framework, the (true) hadronic tau collection is first extracted from the event history. In contrast, in DELPHES 3, hadronic taus are defined from the jet collection. To account for this difference, we implement identical selections on taus and jets at the level of the analysis. Additionally, we restrict the collection of tau candidates and select only taus whose transverse momentum is larger than 20 GeV, and remove any potential overlap between the tau and the jet collection by ignoring any jet that is too close to a tau ($\Delta R_{\tau j} < 0.2$).

In Fig. 6, we compare predictions obtained with the SFS approach, DELPHES 3 and the Monte Carlo truth (*i.e.* for an ideal detector).

We start by considering the tau multiplicity spectrum (upper left). As expected from the imperfections of the tagger of Eqs. (12) and (13), a large number of true tau objects are not tagged as such, for all three detector simulation setups.

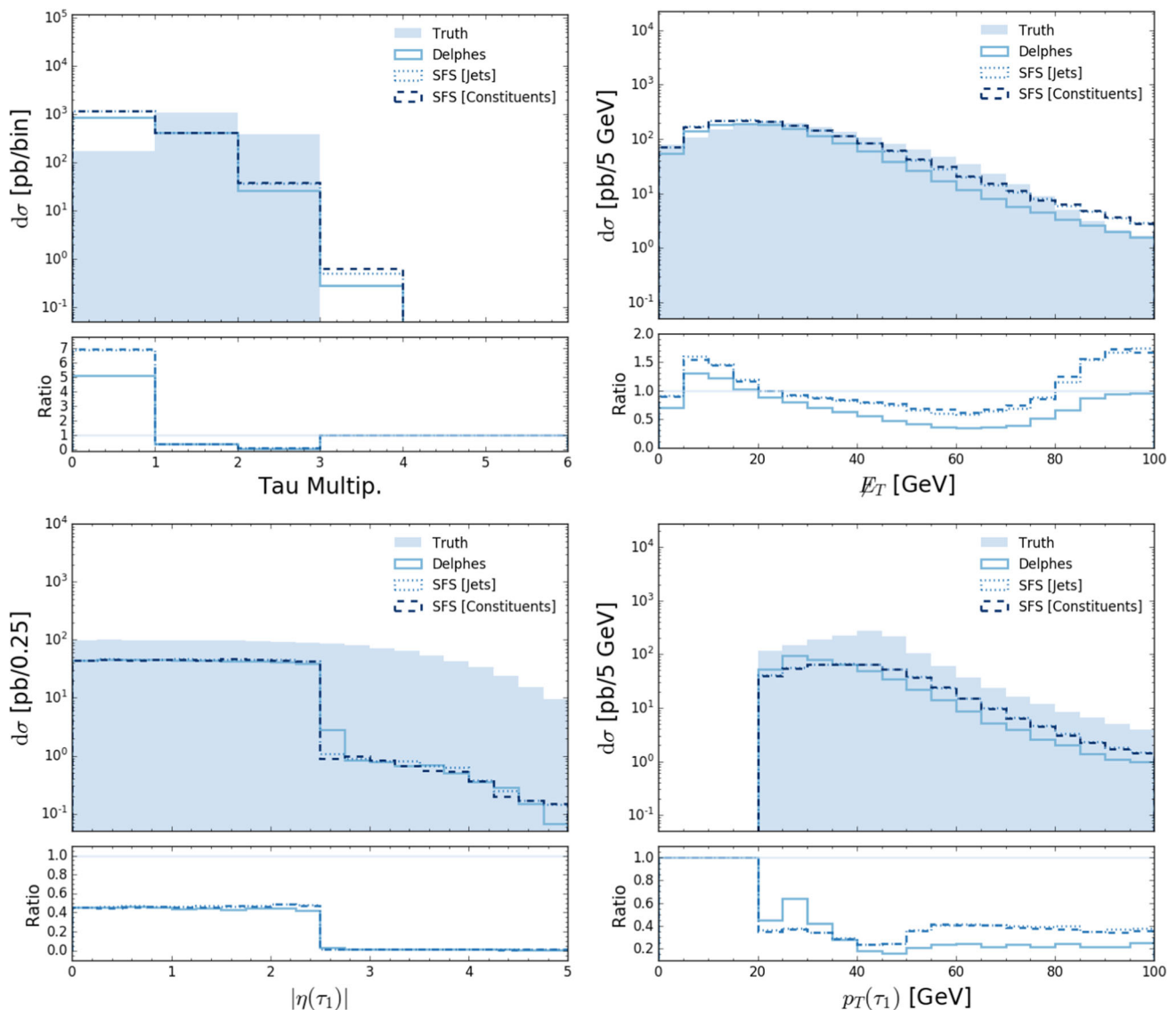


Fig. 6 Same as Fig. 2, but for the reconstructed hadronic tau multiplicity distribution (upper left), and the pseudo-rapidity (lower left) and transverse momentum (lower right) spectrum of the leading tau. We additionally include the missing transverse energy spectrum (upper right)

We observe a quite good agreement between SFS-based and DELPHES 3-based results, despite the above-mentioned differences between the two approaches for tau tagging.

As for the electron and muon case, predictions for the pseudo-rapidity spectrum of the leading tau (lower left) are comparable, regardless of the adopted detector simulator. The main differences arise in the $|\eta| > 2.5$ forward regime. For SFS-based results, forward taus are stemming from the misidentification of a light jet as a tau, whereas in the DELPHES 3 case, the entire jet collection (including c -jets and b -jets) is used. This different treatment, on top of the above-mentioned differences inherent to the whole tau-tagging method as well as the statistical limitations of the generated event sample in the forward regime (that actually

dominate), leads to the observed deviations between the predictions for $|\eta| > 2.5$.

In the lower right panel of Fig. 6, we present the distribution in the transverse momentum of the leading tau. As for the other considered observables, the detector effects are pretty important when compared with the Monte Carlo truth. For hard taus with $p_T \gtrsim 50$ GeV, DELPHES 3 and SFS-based predictions are in very good agreement. However, the different treatment in the two simulators significantly impacts the lower p_T regime, leading in a very different behaviour.

In the upper right panel, we show how those effects impact a more global observable. This is illustrated with the missing transverse energy $\cancel{E}_T$ (that is reconstructed with a particle flow algorithm in DELPHES 3). The discrepancies between DELPHES 3 and SFS results are not so drastic as for the

$p_T(\tau_1)$ distribution in the soft regime, but impact instead the entire spectrum with a shift of $\mathcal{O}(10)\%$ in one way or the other. When $\cancel{E}_T > 100$ GeV (not shown on the figure), however, we enter the hard regime where the exact details of the detector simulation matter less and a good agreement is obtained between all predictions.

4.5 Photon production

This subsection is dedicated to the last class of objects that could be reconstructed in a detector, namely photons. To compare the expectation from predictions made with DELPHES 3 to those made with the simplified fast detector simulator of MADANALYSIS 5, we generate a sample of 250,000 di-photon events, $pp \rightarrow \gamma\gamma$. In our simulations, we impose a generator level selection of 10 GeV on the photon p_T , as its pseudo-rapidity is constrained to be below 5 in absolute value.

The SFS simulation includes a reconstruction efficiency ε_γ given by

$$\varepsilon_\gamma(p_T, \eta) = \begin{cases} 0.96 & \text{for } p_T > 10 \text{ GeV and } |\eta| \leq 1.5, \\ 0.875 & \text{for } p_T > 10 \text{ GeV and } 1.5 < |\eta| \leq 2.5, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

and the photon energy is smeared as in Eq. (10), this smearing originating from the same electronic calorimeter effects as in the electron case.

At the analysis level, we follow the same strategy as in the previous sections. In practice, we begin with rejecting all leptons and photons with a transverse momentum smaller than 10 GeV, and all jets with a transverse momentum smaller than 20 GeV. Then, we remove from the jet collection any jet that lies at a distance in the transverse plane $\Delta R_{ej} < 0.2$ from any electron candidate, and next remove from the lepton collection any lepton that would lie at a distance $\Delta R_{\ell j} < 0.4$ of any of the remaining jets. We finally remove, from the photon collection, any photon lying at a distance $\Delta R_{aj} < 0.4$ of a jet.

In order to assess the impact of the detector on the photons, we compare in Fig. 7 the Monte Carlo truth predictions (ideal detector) with the results obtained with the MADANALYSIS 5 SFS simulation and with DELPHES 3. In all cases, the detector capabilities in reconstructing the photons are quite good, only a few photons being lost, and their properties are nicely reproduced. Some exceptions are in order, in particular for the photon multiplicity spectrum. Predictions using DELPHES 3 and those relying on the SFS framework are quite different for bins associated with a number of photons larger than 2. Those extra photons (relatively to the two photons originating from the hard process) come the hadronic decays following the hadronisation process. In some rare cases, they are pro-

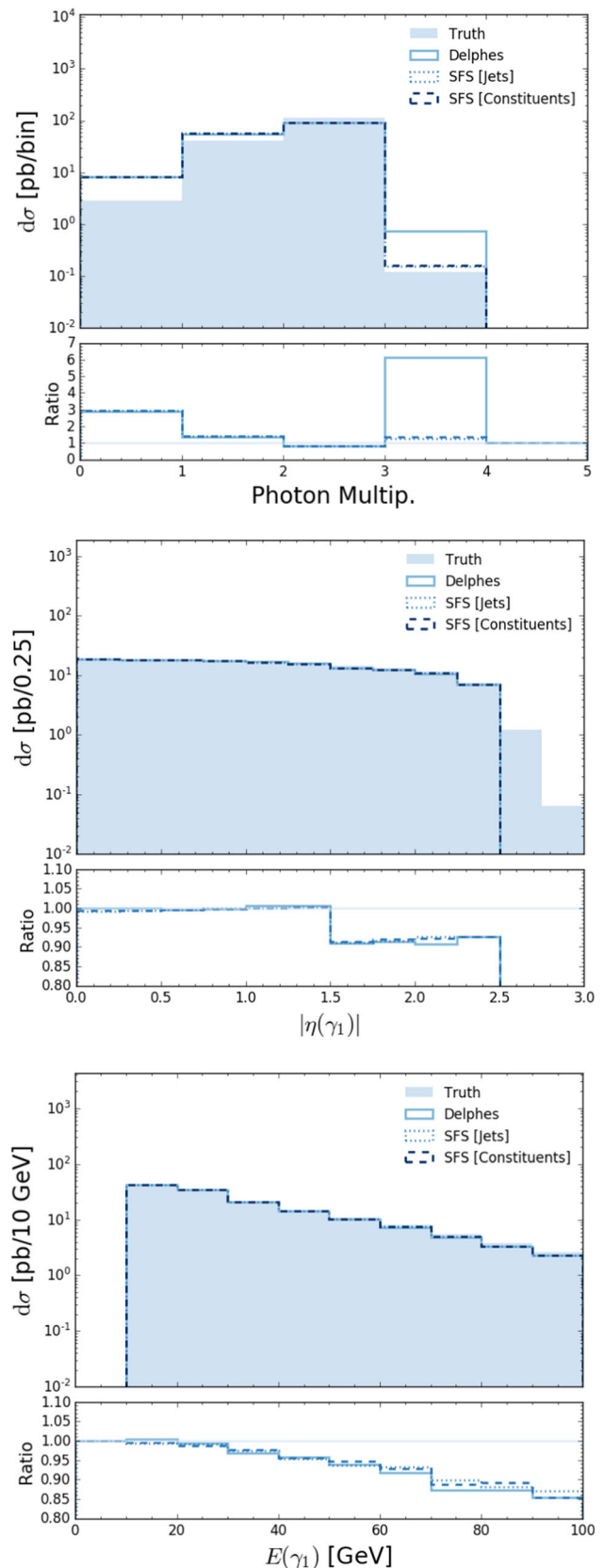


Fig. 7 Same as Fig. 2, but for the reconstructed photon multiplicity distribution (upper), and for the pseudo-rapidity (centre) and energy (lower) spectrum of the leading reconstructed photon

duced at wide angles and are thus not clustered back by the jet algorithm. As photon isolation and reconstruction are treated quite differently in the SFS framework and in DELPHES 3, differences are expected, as visible from the photon multiplicity figure.

5 Reinterpreting the results of the LHC

5.1 Generalities

The LHC collaborations usually interpret their results for a specific set of selected models. This hence leaves the task of the reinterpretation in other theoretical frameworks to studies that have to be carried out outside the collaborations. The most precise method that is available to theorists in this context relies on the Monte Carlo simulation of any new physics signal of interest. The resulting events are first reconstructed by including a detector simulation mimicking the ATLAS or CMS detector, and next studied in order to see to which extent the signal regions of a given analysis are populated. From those predictions, it is then possible to conclude about the level of exclusion of the signal, from a comparison with data and the Standard Model expectation.

Such an analysis framework is available within MADANALYSIS 5 for half a decade [9, 10]. In practice, it makes use of the MADANALYSIS 5 interface to DELPHES 3 to handle the simulation of the detector.

In this work, we have extended this infrastructure, so that the code offers, from version 1.8.51 onwards, the choice to employ either DELPHES 3 or the SFS to deal with the simulation of the detector response. Similarly to what is done in RIVET [4] or COLLIDERBIT [32], the simulation of the detector can now be handled together with a simple event reconstruction to be performed with FASTJET, through transfer functions embedding the various reconstruction and tagging efficiencies (see Sect. 3).

The estimation of the LHC potential relatively to any given new physics signal can be achieved by typing, in the MADANALYSIS 5 interpreter (after having started the programme in the reconstruction-level mode),

```
set main.recast = on
import <events.hepmc.gz>
submit
```

The above set of commands turns on the recasting module of the platform and allows for the reinterpretation of the results of all implemented LHC analyses available on the user system. The signal to test is described by the `<events.hepmc.gz>` hadron-level event sample. After submission, MADANALYSIS 5 generates a recasting card requiring to switch on or off any of all implemented analyses, regardless that they use DELPHES 3 or the SFS for the

simulation of the detector response. On run time, the code automatically chooses the way to handle it, so that the user does not have to deal with it by themselves. We refer to refs. [9, 11] for more information on LHC recasting in the MADANALYSIS 5 context, the installation of the standard Public Analysis Database analyses (PAD) that relies on DELPHES 3, as well as for a detailed list with all available options.

All analyses that have been implemented and validated within the SFS context can be downloaded from the internet and locally installed by typing, in the command-line interface,

```
install PADForSFS
```

Up to now, this command triggers the installation of four analyses, namely the ATLAS-SUSY-2016-07 search for gluinos and squarks in the multi-jet plus missing transverse energy channel [33] and its ATLAS-CONF-2019-040 full run 2 update [34], the ATLAS-SUSY-2018-31 search for sbottoms when their decay gives rise to many b -jets (possibly originating from intermediate Higgs boson decays) and missing transverse energy [28] and the CMS-SUS-16-048 search for charginos and neutralinos through a signature comprised of soft leptons and missing transverse energy [35]. Details about the validation of the SFS implementations of the ATLAS-SUSY-2016-07 and CMS-SUS-16-048 analyses are provided in Sects. 5.2 and 5.3 respectively. The implementation of the ATLAS-SUSY-2018-31 and its validation have been documented in refs. [36, 37], and we refer to the MADANALYSIS 5 website⁶ for information about the implementation and validation of the last analyses. This webpage will maintain an up-to-date list with all validated analyses available for LHC recasting with an SFS detector simulation, in addition to those that could be used with DELPHES 3 as a detector simulator.

In the future, MADANALYSIS 5 aims to support both DELPHES 3-based and SFS-based LHC recasting. On the one hand, this strategy prevents us from having to re-implement, in the SFS framework, any single PAD analysis that is already available when relying on DELPHES 3 as a detector simulator. Second, this leaves more freedom to the user who would like to implement a new analysis in the PAD for what concerns the choice of the treatment of the detector effects. It should however be kept in mind that new functionalities that are currently being developed will extend the built-in SFS capabilities of MADANALYSIS 5, and not add any new feature to DELPHES 3. For instance, methods to deal with long-lived particles going beyond what DELPHES 3 could do are already available from the version 1.9.10 of MADANALYSIS 5 [38].

In the rest of this section, we compare the SFS-based predictions with those resulting from the usage of the DELPHES 3

⁶ See the webpage <http://madanalysis.irmp.ucl.ac.be/wiki/PublicAnalysisDatabase>.

software for the simulation of the response of the LHC detectors for the two considered analyses. Additionally to a direct comparison of a DELPHES 3-based and transfer-function-based approach for LHC recasting, this allows one to assess the capabilities of the SFS approach as compared with DELPHES 3 in the case of events featuring very hard objects that are typical of most searches for new physics at the LHC. Very hard jets are in particular considered in the ATLAS-SUSY-2016-07 analysis, which contrasts with the study of the jet properties achieved in Sect. 4 that solely covers objects featuring a moderate transverse momentum of 10–100 GeV.

5.2 Recasting a multi-jet plus missing energy ATLAS search for squarks and gluinos

In the ATLAS-SUSY-2016-07 analysis, the ATLAS collaboration searches for squarks and gluinos through a signature comprised of 2 to 6 jets and a large amount of missing transverse energy. A luminosity of 36.1 fb^{-1} of proton-proton collisions at a centre-of-mass energy of 13 TeV is analysed. This search includes two classes of signal regions. The first one relies on the effective mass variable $M_{\text{eff}}(N)$, defined as the scalar sum of the transverse momentum of the N leading jets and the missing transverse energy, and the second one on the recursive jigsaw reconstruction technique [39]. However, only the former region can be recasted due to the lack of public information associated with the signal regions relying on the jigsaw reconstruction technique.

Consequently, only signal regions depending on the $M_{\text{eff}}(N)$ quantity have been implemented in the MADANALYSIS 5 framework, as detailed in Ref. [40]. In this implementation, the simulation of the detector is handled with DELPHES 3 and an appropriately tuned parameter card. This DELPHES 3-based recast code has been validated by reproducing public information provided by the ATLAS collaboration, so that we use it as a reference below. In the following, we denote predictions obtained with it as ‘PAD’ results, the acronym PAD referring to the traditional Public Analysis Database of MADANALYSIS 5 that relies on DELPHES 3 for the simulation of the detector. In contrast, results obtained by using an SFS detector simulation are tagged as ‘SFS’ or ‘PADForSFS’ results.

In order to illustrate the usage of the SFS framework for LHC recasting, we have modified the original ATLAS-SUSY-2016-07 analysis implementation and included it in the PADForSFS database, together with an appropriate ATLAS SFS detector parametrisation. The user has thus the choice to use either DELPHES 3 or the SFS framework for the reinterpretation of the results of this ATLAS analysis.

The validation of our implementation has been achieved by comparing predictions for a well-defined benchmark scenario with those obtained with the reference version of the

implementation based on DELPHES 3. We have adopted a simplified model setup inspired by the Minimal Supersymmetric Standard Model in which all superpartners are decoupled, with the exception of the gluino and the lightest neutralino. Their masses have been fixed to 1 TeV and 825 GeV respectively, and the gluino is enforced to decay into a di-jet plus neutralino system with a branching ratio of 1.

We have generated 200,000 new physics Monte Carlo events by matching leading-order matrix elements convoluted with the leading order set of NNPDF 2.3 parton densities [26] as generated by MG5_AMC[25], with the parton shower machinery of PYTHIA 8[27]. Gluino decays are handled with the MADSPIN [41] and MADWIDTH [42] packages, and we have relied on PYTHIA 8 for the simulation of the hadronisation processes. Those events have then been analysed automatically in MADANALYSIS 5, both in the ‘PAD’ context with a DELPHES 3-based detector simulation and in the SFS context with an SFS-based detector simulation.

We have compared the two sets of results and found that they deviate by at most 10% for all signal regions populated by at least 20 events (out of the 200,000 simulated events). In addition, we have verified that enforcing a constituent-based or jet-based jet smearing had little impact on the results. We have observed that this choice indeed leads to a modification of the SFS predictions of about 1%, so that the jet-smearing choice is irrelevant. Jet-based jet smearing is therefore used below. Furthermore, in terms of performance, the run of DELPHES 3 has been found 50% slower than the SFS one.

We present a subset of the results in Table 9, focusing on three of the ATLAS-SUSY-2016-07 signal regions that are among the most populated ones by the considered 1 TeV gluino signal. We show predictions obtained by using DELPHES 3 for the simulation of the detector (PAD), and depict them both in terms of the number of events n_i surviving a cut i and of the related cut efficiency

$$\varepsilon_i = n_i / n_{i-1} . \quad (15)$$

We additionally display SFS-based predictions (SFS), showing again both the number of events and the various cut efficiencies.

The deviations δ_i between the DELPHES 3 and SFS predictions are evaluated at the level of the efficiencies,

$$\delta_i = \left| 1 - \frac{\varepsilon_i^{\text{SFS}}}{\varepsilon_i^{\text{PAD}}} \right| . \quad (16)$$

As above-mentioned, the deviations at any cut-level are smaller than 10%, and often lie at the level of 1%. This is in particular the case for the cuts relevant to the other 19 (not shown) signal regions. A more complete set of results, including predictions for all signal regions, is available online.⁷

⁷ See the PADForSFS webpage <http://madanalysis.irmp.ucl.ac.be/wiki/SFS>.

Table 9 Cut-flow charts associated with three signal regions of the ATLAS-SUSY-2016-07 analysis, among the most populated ones by a 1 TeV gluino signal. We focus on regions relevant for events featuringat least 3 jets (top), 4 jets (middle) or 6 jets (lower). In our notation, j_k denotes any of the three leading jets, and j_l any jet from the third one

Cuts	n_i^{PAD}	$\varepsilon_i^{\text{PAD}}$	n_i^{SFS}	$\varepsilon_i^{\text{SFS}}$	δ_i
Initial	7262.2	—	7262.2	—	—
Preselection	1129.8	0.156	1092.0	0.150	3.3%
At least 2 jets	1125.0	0.996	1087.5	0.996	0
At least 3 jets	1073.0	0.954	1032.9	0.950	0.4%
$\Delta\varphi(j_k, \not{p}_T) > 0.4$	848.5	0.791	821.6	0.796	0.6%
$\Delta\varphi(j_3, \not{p}_T) > 0.2$	767.6	0.905	743.9	0.905	0.1%
$p_T(j_1) > 700$ GeV	53.9	0.070	54.8	0.074	4.9%
$p_T(j_2) > 50$ GeV	53.9	1	54.8	1	0
$p_T(j_3) > 50$ GeV	53.9	1	54.8	1	0
$\cancel{E}_T/\sqrt{H_T} > 16\sqrt{\text{GeV}}$	43.0	0.798	44.4	0.812	1.7%
$M_{\text{eff}}(3) > 1300$ GeV	43.0	1	44.4	1	0
Cuts	n_i^{PAD}	$\varepsilon_i^{\text{PAD}}$	n_i^{SFS}	$\varepsilon_i^{\text{SFS}}$	δ_i
Initial	7262.2	—	7262.2	—	—
Preselection	1129.8	0.156	1092.0	0.150	3.3%
At least 2 jets	1125.0	0.996	1087.5	0.996	0
At least 4 jets	857.4	0.762	821.2	0.755	0.9%
$\Delta\varphi(j_k, \not{p}_T) > 0.4$	677.1	0.790	652.3	0.794	0.6%
$\Delta\varphi(j_l, \not{p}_T) > 0.4$	522.0	0.771	500.8	0.768	0.4%
$p_T(j_4) > 100$ GeV	113.5	0.218	112.5	0.225	3.3%
$ \eta(j) < 2$	91.1	0.803%	89.0	0.791	1.5%
Aplanarity > 0.04	54.5	0.598	52.9	0.594	0.7%
$\cancel{E}_T/M_{\text{eff}}(4) > 0.25$	49.2	0.902	48.2	0.911	1.0%
$M_{\text{eff}}(4) > 1400$ GeV	27.0	0.549	27.5	0.570	3.8%
Cuts	n_i^{PAD}	$\varepsilon_i^{\text{PAD}}$	n_i^{SFS}	$\varepsilon_i^{\text{SFS}}$	δ_i
Initial	7262.2	—	7262.2	—	—
Preselection	1129.8	0.156	1092.0	0.15	3.3%
At least 2 jets	1125.0	0.996	1087.5	0.996	0
At least 6 jets	244.2	0.217	225.2	0.207	4.6%
$\Delta\varphi(j_k, \not{p}_T) > 0.4$	194.0	0.795	179.5	0.797	0.3%
$\Delta\varphi(j_l, \not{p}_T) > 0.2$	154.9	0.798	143.3	0.799	0
$ \eta(j) < 2$	101.4	0.655	93.0	0.649	0.8%
$\cancel{E}_T/M_{\text{eff}}(6) > 0.25$	85.9	0.847	78.2	0.841	0.8%
$M_{\text{eff}}(6) > 1200$ GeV	67.6	0.787	61.6	0.788	0.1%

5.3 Recasting a CMS search for compressed electroweakinos with soft leptons and missing energy

The CMS-SUS-16-048 analysis is an unusual search for the supersymmetric partners of the Standard Model gauge and Higgs bosons. It relies on the reconstruction of soft leptons with a transverse momentum smaller than 30 GeV, and mainly targets the associated production of a neutralino and chargino pair $\tilde{\chi}_1^\pm \tilde{\chi}_2^0$ in a setup in which the supersymmetric spectrum is compressed and features small mass splittings

between these two states and the lightest neutralino $\tilde{\chi}_1^0$. This search investigates 35.9 fb^{-1} of proton-proton collisions at a centre-of-mass energy of 13 TeV.

The production process under consideration ($pp \rightarrow \tilde{\chi}_1^\pm \tilde{\chi}_2^0$) is thus followed by a decay of both superpartners into the lightest neutralino $\tilde{\chi}_1^0$ and an off-shell gauge boson,

$$pp \rightarrow \tilde{\chi}_1^\pm \tilde{\chi}_2^0 \rightarrow (W^* \tilde{\chi}_1^0) (Z^* \tilde{\chi}_1^0). \quad (17)$$

The considered signal is thus potentially comprised of a pair of opposite-sign (OS) soft leptons (arising from the off-shell

gauge bosons), possibly carrying the same flavour (SF), and some missing transverse momentum carried away by the two produced lightest neutralinos.

The CMS collaboration has also designed a series of signal regions dedicated to the search for electroweakino production from stop decays,

$$pp \rightarrow t\bar{t}^* \rightarrow (\chi_1^\pm b) (\chi_1^\mp \bar{b}) \rightarrow (W^* \tilde{\chi}_1^0 b) (W^* \tilde{\chi}_1^0 \bar{b}). \quad (18)$$

As for the process of Eq. (17), such a signal also gives rise to soft leptons, produced this time in association with b -jets. The top squark is however compressed with the other states, so that those b -jets are in most cases not identified. The resulting signature is therefore very similar to the one originating from Eq. (17).

This CMS-SUS-16-048 analysis has been implemented in the MADANALYSIS 5 framework and validated in the context of the last Les Houches workshop on TeV colliders [43], both for a simulation of the detector effects relying on DELPHES 3 and on the SFS infrastructure. This analysis is therefore both included in the standard PAD and new PADForSFS database.

As in Sect. 5.2, the recast code based on DELPHES 3 has been first validated against publicly available material provided by the CMS collaboration. The SFS-based version of the analysis has been developed next, using the DELPHES 3-based one as a reference for the validation procedure. Whereas both implementations were validated in Ref. [43], we report in the following on the comparison of DELPHES 3-based and SFS-based predictions for a given signal, as this fits in the spirit of this work aiming at documenting the performance of the new SFS module of MADANALYSIS 5 and comparing with a more standard DELPHES 3-based detector simulation.

For our comparison, we again consider a simplified model inspired by the Minimal Supersymmetric Standard Model. The Standard Model is extended by the three electroweakino states relevant for the signal of Eq. (17). We make use of one of the next-to-minimal simplified scenarios proposed in Ref. [44] and that is studied in the considered CMS analysis, and set the mass of the lightest neutralino to 142.5 GeV and the ones of the other states to 150 GeV.

We generate 300,000 signal events in the same simulation chain as the one described in the previous section. When comparing the recasting results, we observe a good agreement between DELPHES 3-based and SFS-based predictions. Ignoring signal regions featuring a poor statistics (less than 10 events out of the 300,000 generated ones), the cut efficiencies are found to differ by at most 50%, although the deviations are found to lie at the percent level for a large majority of cuts from all signal regions. In addition, using jet-based or constituent-based jet smearing does not change the conclusions, the deviations δ_i from DELPHES 3 being impacted by about 1%.

We illustrate our results in Table 10 for two of the CMS-SUS-16-048 signal regions that are among the most populated ones by the considered compressed electroweakino signal. The complete set of results can be obtained online, on the SFS wiki page.

6 Conclusions

We have discussed the implementation of a new detector emulator in the MADANALYSIS 5 framework. This extends the capacities of the platform when FASTJET is used for event reconstruction.

Our work includes two components. First, the PYTHON-like interpreter of MADANALYSIS 5 has been extended by new commands. Those allow for an intuitive parametrisation of a high-energy physics detector through user-defined smearing, reconstruction and identification/mistagging efficiency functions. Next, the C++ core of the programme has been modified so that it could handle those user-defined functions, that are converted from the PYTHON-like MADANALYSIS 5 metalanguage to C++ by the interpreter.

On run time, MADANALYSIS 5 automatically generates a full C++ code that allows for event reconstruction with FASTJET and that incorporates the input detector effects. The code is then automatically compiled and run.

Correspondingly, we have extended the recasting infrastructure of MADANALYSIS 5. The user has now the option to rely on our fast simplified detector simulator instead of DELPHES 3 to simulate the ATLAS and CMS detectors. Four LHC analyses are currently available and have been validated. Moreover, the programme is shipped with two configuration scripts reproducing the response of the ATLAS and CMS detectors.

In order to validate our implementation, we have compared SFS-based predictions with those obtained when the DELPHES 3 package is used instead for the simulation of the detector effects. We have considered Standard Model processes as well as new physics signals, the latter being investigated in the context of LHC recasting. In all our comparisons, we have found that the variations between the two approaches are most of the time of about 10%. However, there are cases where larger discrepancies are found, as could be expected by virtue of the different strategies followed by the DELPHES 3 detector simulator and the SFS approach. The main advantages of our emulator is that it allows the user to deal with any custom detector in a very simple and user-friendly manner, either through intuitive PYTHON-like commands to be cast in the MADANALYSIS 5 interpreter, or directly by programming within the expert mode of the platform. In addition, our simplified detector simulation is generally faster than DELPHES 3, as it is more light-weight. With such a machinery at hand, it becomes thus possible to use

Table 10 Cut-flow charts associated with two signal regions of the CMS-SUS-16-048 analysis, among the most populated ones by the considered compressed electroweakino scenario (see the text). We focus on a region dedicated to the search for electroweakino production from stop decays (upper) and a region dedicated to direct electroweakino pro-

duction (lower). In our notation, $\cancel{E}_T^{\text{cor.}}$ denotes the transverse momentum resulting from the vector sum of the missing momentum and the momenta of the two leading leptons, and $M_{\tau\tau}$ represents the invariant mass of the di-tau system that would stem from considering the two leptons as originating from tau decays

Cuts	n_i^{PAD}	$\varepsilon_i^{\text{PAD}}$	n_i^{SFS}	$\varepsilon_i^{\text{SFS}}$	δ_i
Initial	48572.7	—	48572.7	—	—
At least 1 lepton	17028.2	0.351	17369.5	0.358	2.0%
$p_T(\ell_1) \in [5, 30]$ GeV	13142.0	0.772	13266.2	0.764	1.0%
$p_T(\ell_2) < 30$ GeV	1677.8	0.128	1688.4	0.127	0.3%
OS dilepton	1042.6	0.621	1041.4	0.617	0.7%
$p_T(\ell_1\ell_2) > 3$ GeV	961.6	0.922	963.9	0.926	0.4%
$\cancel{E}_T > 125$ GeV	148.9	0.155	156.2	0.162	4.6%
$\cancel{E}_T^{\text{cor.}} > 125$ GeV	90.5	0.608	93.2	0.597	1.9%
$\cancel{E}_T > 300$ GeV	27.9	0.308	25.8	0.277	10.0%
At least 1 jet	27.8	0.997	25.7	0.994	0.3%
$H_T > 100$ GeV	27.2	0.979	25.0	0.974	0.5%
$\cancel{E}_T/H_T \in [0.6, 1.4]$	25.2	0.924	23.7	0.948	2.5%
b -jet veto	15.3	0.609	14.6	0.616	1.0%
$M_{\tau\tau} \notin [0, 160]$ GeV	12.1	0.788	11.7	0.799	1.5%
$p_T(\ell_1) < 12$ GeV	5.1	0.426	5.7	0.484	13.7%
Cuts	n_i^{PAD}	$\varepsilon_i^{\text{PAD}}$	n_i^{SFS}	$\varepsilon_i^{\text{SFS}}$	δ_i
Initial	48572.7	—	48572.7	—	—
At least 1 lepton	17028.2	0.351	17369.5	0.358	2.0%
$p_T(\ell_1) \in [5, 30]$ GeV	13142.0	0.772	13266.2	0.764	1.0%
$p_T(\ell_2) < 30$ GeV	1677.8	0.128	1688.4	0.127	0.3%
$p_T(\ell_2) > 5$ GeV	922.7	0.550	923.1	0.547	0.6%
SF dilepton	597.8	0.648	600.0	0.650	0.3%
OS dilepton	360.2	0.603	359.0	0.598	0.7%
$p_T(\ell_1\ell_2) > 3$ GeV	331.4	0.920	328.3	0.914	0.6%
$M_{\ell_1\ell_2} \in [4, 50]$ GeV	245.0	0.739	245.6	0.748	1.2%
Hadronic res. veto	237.8	0.971	238.1	0.969	0.1%
$\cancel{E}_T > 125$ GeV	40.0	0.168	44.6	0.187	11.5%
$\cancel{E}_T^{\text{cor.}} > 125$ GeV	26.1	0.654	29.1	0.652	0.3%
$\cancel{E}_T > 250$ GeV	13.1	0.500	14.5	0.499	0.1%
At least 1 jet	12.8	0.981	14.3	0.983	0.2%
$H_T > 100$ GeV	12.7	0.987	13.9	0.977	1.1%
$\cancel{E}_T/H_T \in [0.6, 1.4]$	11.3	0.889	13.0	0.929	4.4%
b -jet veto	7.5	0.666	8.9	0.684	2.7%
$M_{\tau\tau} \notin [0, 160]$ GeV	5.5	0.738	7.0	0.788	6.8%
$M_T(\ell, \cancel{p}_T) < 70$ GeV	4.5	0.821	6.1	0.871	6.1%
$M_{\ell_1\ell_2} < 9$ GeV	3.8	0.837	4.8	0.785	6.2%

the code to single out a particular detector effect and hence determine the performance of any given (future) detector.

With this work, we have hence augmented the capabilities of the MADANALYSIS 5 package, offering a new fast and efficient way to emulate the response of a typical high-energy physics detector. The user is now allowed to choose between a DELPHES 3-based or a transfer-function-based simulation of the detector response, both methods being implemented through an intuitive set of user-friendly methods included in the MADANALYSIS 5 command-line interface. Predictions compatible with those obtained when DELPHES 3 is used have been found, the difference lying at the level of the uncertainties inherent to a fast detector simulation carried out outside the experimental collaborations.

In a future release, we plan to further extend both our package and the normal mode of MADANALYSIS 5 to include the handling of vertex position and impact parameter variables. Those heavier changes will consequently allow for an appropriate modelling of long-lived particles in the context of our simplified fast detector simulation [38].

Acknowledgements The authors are grateful to Eric Conte, Mariana Frank and Sabine Kraml for useful comments on the manuscript. JYA acknowledges the hospitality of the LPTHE (Sorbonne University), IPPP (Durham University) and LNF (INFN) where parts of this work have been completed during his visits. JYA has received funding from the European Union's Horizon 2020 research and innovation programme as part of the Marie Skłodowska-Curie Innovative Training Network MCnetITN3 (Grant Agreement no. 722104).

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: There is no data associated with this paper.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. Funded by SCOAP³.

References

1. F. Spano, Unfolding in particle physics: a window on solving inverse problems. EPJ Web Conf. **55**, 03002 (2013)
2. GEANT4 collaboration, S. Agostinelli et al., GEANT4: A Simulation toolkit, Nucl. Instrum. Meth. A **506** (2003) 250–303
3. DELPHES 3 collaboration, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens et al., DELPHES 3, A modular framework for fast simulation of a generic collider experiment, JHEP **02** (2014) 057, [arXiv:1307.6346](https://arxiv.org/abs/1307.6346)
4. A. Buckley, J. Butterworth, L. Lonnblad, D. Grellscheid, H. Hoeth, J. Monk et al., Rivet user manual. Comput. Phys. Commun. **184**, 2803–2819 (2013). [arXiv:1003.0694](https://arxiv.org/abs/1003.0694)
5. M. Cacciari, G.P. Salam, G. Soyez, FastJet User Manual. Eur. Phys. J. C **72**, 1896 (2012). [arXiv:1111.6097](https://arxiv.org/abs/1111.6097)
6. A. Buckley, D. Kar, K. Nordström, Fast simulation of detector effects in Rivet. SciPost Phys. **8**, 025 (2020). [arXiv:1910.01637](https://arxiv.org/abs/1910.01637)
7. E. Conte, B. Fuks, G. Serret, MadAnalysis 5, A User-Friendly Framework for Collider Phenomenology. Comput. Phys. Commun. **184**, 222–256 (2013). [arXiv:1206.1599](https://arxiv.org/abs/1206.1599)
8. E. Conte, B. Dumont, B. Fuks, C. Wymant, Designing and recasting LHC analyses with MadAnalysis 5. Eur. Phys. J. C **74**, 3103 (2014). [arXiv:1405.3982](https://arxiv.org/abs/1405.3982)
9. E. Conte, B. Fuks, Confronting new physics theories to LHC data with MADANALYSIS 5. Int. J. Mod. Phys. A **33**, 1830027 (2018). [arXiv:1808.00480](https://arxiv.org/abs/1808.00480)
10. B. Dumont, B. Fuks, S. Kraml, S. Bein, G. Chalons, E. Conte et al., Toward a public analysis database for LHC new physics searches using MADANALYSIS 5. Eur. Phys. J. C **75**, 56 (2015). [arXiv:1407.3278](https://arxiv.org/abs/1407.3278)
11. J.Y. Araz, M. Frank, B. Fuks, Reinterpreting the results of the LHC with MadAnalysis 5: uncertainties and higher-luminosity estimates. Eur. Phys. J. C **80**, 531 (2020). [arXiv:1910.11418](https://arxiv.org/abs/1910.11418)
12. M. Dobbs, J.B. Hansen, The HepMC C++ Monte Carlo event record for High Energy Physics. Comput. Phys. Commun. **134**, 41–46 (2001)
13. M. Cacciari, G.P. Salam, G. Soyez, The Anti-k(t) jet clustering algorithm. JHEP **04**, 063 (2008). [arXiv:0802.1189](https://arxiv.org/abs/0802.1189)
14. S. Catani, Y.L. Dokshitzer, M.H. Seymour, B.R. Webber, Longitudinally invariant K_T clustering algorithms for hadron hadron collisions. Nucl. Phys. B **406**, 187–224 (1993)
15. S.D. Ellis, D.E. Soper, Successive combination jet algorithm for hadron collisions. Phys. Rev. D **48**, 3160–3166 (1993). [arXiv:hep-ph/9305266](https://arxiv.org/abs/hep-ph/9305266)
16. Y.L. Dokshitzer, G.D. Leder, S. Moretti, B.R. Webber, Better jet clustering algorithms. JHEP **08**, 001 (1997). [arXiv:hep-ph/9707323](https://arxiv.org/abs/hep-ph/9707323)
17. M. Wobisch, T. Wengler, Hadronization corrections to jet cross-sections in deep inelastic scattering, in Monte Carlo generators for HERA physics. Proceedings, Workshop, Hamburg, Germany **1998–1999**, 270–279 (1998). [arXiv:hep-ph/9907280](https://arxiv.org/abs/hep-ph/9907280)
18. G.P. Salam, G. Soyez, A Practical Seedless Infrared-Safe Cone jet algorithm. JHEP **05**, 086 (2007). [arXiv:0704.0292](https://arxiv.org/abs/0704.0292)
19. Particle Data Group collaboration, M. Tanabashi et al., Review of Particle Physics, Phys. Rev. D **98** (2018) 030001
20. E. Boos et al., Generic User Process Interface for Event Generators, in *2nd Les Houches Workshop on Physics at TeV Colliders*, 9, 2001. [arXiv:hep-ph/0109068](https://arxiv.org/abs/hep-ph/0109068)
21. J. Alwall et al., A Standard format for Les Houches event files. Comput. Phys. Commun. **176**, 300–304 (2007). [arXiv:hep-ph/0609017](https://arxiv.org/abs/hep-ph/0609017)
22. C. Fabjan, F. Gianotti, Calorimetry for particle physics. Rev. Mod. Phys. **75**, 1243–1286 (2003)
23. CMS collaboration, S. Chatrchyan et al., Description and performance of track and primary-vertex reconstruction with the CMS tracker, JINST **9** (2014) P10009, [arXiv:1405.6569](https://arxiv.org/abs/1405.6569)
24. CMS collaboration, S. Chatrchyan et al., Identification of b-Quark Jets with the CMS Experiment, JINST **8** (2013) P04013, [arXiv:1211.4462](https://arxiv.org/abs/1211.4462)
25. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. JHEP **07**, 079 (2014). [arXiv:1405.0301](https://arxiv.org/abs/1405.0301)
26. R.D. Ball et al., Parton distributions with LHC data. Nucl. Phys. B **867**, 244–289 (2013). [arXiv:1207.1303](https://arxiv.org/abs/1207.1303)

27. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., An Introduction to PYTHIA 8.2. *Comput. Phys. Commun.* **191**, 159–177 (2015). [arXiv:1410.3012](#)
28. ATLAS collaboration, G. Aad et al., Search for bottom-squark pair production with the ATLAS detector in final states containing Higgs bosons, b-jets and missing transverse momentum, *JHEP* **12** (2019) 060, [arXiv:1908.03122](#)
29. ATLAS collaboration, G. Aad et al., Search for electroweak production of charginos and sleptons decaying into final states with two leptons and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions using the ATLAS detector, *Eur. Phys. J. C* **80** (2020) 123, [arXiv:1908.08215](#)
30. CMS collaboration, V. Khachatryan et al., Performance of Electron Reconstruction and Selection with the CMS Detector in gProton-Proton Collisions at $\sqrt{s} = 8$ TeV, *JINST* **10** (2015) P06005, [arXiv:1502.02701](#)
31. CMS collaboration, S. Chatrchyan et al., Energy Calibration and Resolution of the CMS Electromagnetic Calorimeter in pp Collisions at $\sqrt{s} = 7$ TeV, *JINST* **8** (2013) P09009, [arXiv:1306.2016](#)
32. GAMBIT collaboration, C. Balázs et al., ColliderBit: a GAMBIT module for the calculation of high-energy collider observables and likelihoods, *Eur. Phys. J. C* **77** (2017) 795, [arXiv:1705.07919](#)
33. ATLAS collaboration, M. Aaboud et al., Search for squarks and gluinos in final states with jets and missing transverse momentum using 36 fb^{-1} of $\sqrt{s} = 13$ TeV pp collision data with the ATLAS detector, *Phys. Rev. D* **97** (2018) 112001, [arXiv:1712.02332](#)
34. ATLAS collaboration, The ATLAS collaboration, Search for squarks and gluinos in final states with jets and missing transverse momentum using 139 fb^{-1} of $\sqrt{s} = 13$ TeV pp collision data with the ATLAS detector, ATLAS-CONF-2019-040
35. CMS collaboration, A. M. Sirunyan et al., Search for new physics in events with two soft oppositely charged leptons and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV, *Phys. Lett. B* **782** (2018) 440–467, [arXiv:1801.01846](#)
36. J.Y. Araz et al., Proceedings of the second MadAnalysis 5 workshop on LHC recasting in Korea. *Mod. Phys. Lett. A* **36**, 2102001 (2021). [arXiv:2101.02245](#)
37. J.Y. Araz, B. Fuks, Implementation of the ATLAS-SUSY-2018-31 analysis in the MadAnalysis 5 framework (sbottoms with multi-bottoms and missing transverse energy; 139 fb^{-1}). *Mod. Phys. Lett. A* **36**, 2141010 (2021)
38. J. Y. Araz, B. Fuks, M. D. Goodsell and M. Utsch, in preparation,
39. P. Jackson, C. Rogan, Recursive Jigsaw Reconstruction: HEP event analysis in the presence of kinematic and combinatoric ambiguities. *Phys. Rev. D* **96**, 112007 (2017). [arXiv:1705.10733](#)
40. G. Chalons and H. Reyes-Gonzalez, MadAnalysis 5 implementation of ATLAS-SUSY-16-07 [arXiv:1712.02332](#), <https://doi.org/10.7484/INSPIREHEP.DATA.56DC.PPE2>
41. P. Artoisenet, R. Frederix, O. Mattelaer, R. Rietkerk, Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations. *JHEP* **03**, 015 (2013). [arXiv:1212.3460](#)
42. J. Alwall, C. Duhr, B. Fuks, O. Mattelaer, D.G. Öztürk, C.-H. Shen, Computing decay rates for new physics theories with FeynRules and MadGraph5_aMC@NLO. *Comput. Phys. Commun.* **197**, 312–323 (2015). [arXiv:1402.1178](#)
43. G. Brooijmans et al., *Les Houches 2019 Physics at TeV Colliders: New Physics Working Group Report*, in *11th Les Houches Workshop on Physics at TeV Colliders: PhysTeV Les Houches*, 2, 2020. [arXiv:2002.12220](#)
44. B. Fuks, M. Klasen, S. Schmiemann and M. Sunder, Realistic simplified gaugino-higgsino models in the MSSM, *Eur. Phys. J. C* **78** (2018) 209, [arXiv:1710.09941](#)