Chapter 9

Preservation and reusability

Today's particle physics experiments are designed to collect physics data over a span over several decades. They thus operate at scales that makes it impossible for the experiments to be repeated in the foreseeable future. The data taken at these experiments and physics results derived are thus extremely valuable and major problems arise from a scientific reproducibility point of view. In the following, the reusability problems directly connected to an individual analysis are discussed, and approaches taken in view of analysis preservation and analysis reinterpretation are presented. In section 9.1, a motivation for reinterpretations is given and the necessary ingredients are described. The remaining sections discuss three efforts related to ingredients especially necessary for large-scale reinterpretations.

9.1 The case for reinterpretations

9.1.1 Motivation

Designing and executing searches for BSM physics requires a substantial amount of person-power and computing resources. As laid out in detail in part II of this thesis, an analysis generally aims to search regions in which a given BSM model can be efficiently discriminated against SM background. Although the careful design of such regions already requires significant amount of resources, it constitutes only a fraction of the work necessary for concluding the search. Contributions in the search regions from SM processes need to be estimated, usually requiring expensive MC simulation and the development of background estimation strategies. Systematic uncertainties arising from numerous sources need to be considered and estimated. For the BSM signal, a similar processing pipeline involving MC simulation, event reconstruction and event selection needs to be executed. Finally, recorded data also needs to be reconstructed and processed through the event selection. As shown in fig. 9.1, an analysis can thus be divided into three main processing pipelines; a *background pipeline*, *signal pipeline* and *data pipeline*. Only after all three processing pipelines are concluded can the next analysis step be performed—the *statistical inference*, producing the final analysis results, like e.g. limits on model parameters.

Due to the substantial amount of resources necessary for developing and performing an analysis, it is not feasible to develop dedicated searches for every possible BSM scenario. Instead, analyses are typically only interpreted in a finite set of BSM models that only have a small number of

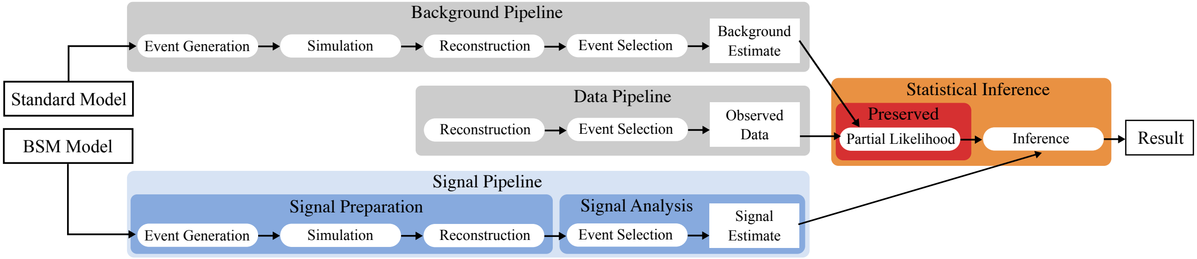

Figure 9.1: Full analysis workflow including the three main processing pipelines for deriving background and signal estimates as well as observed data rates. The outputs of the three processing pipelines are combined into a likelihood forming the basis for statistical inference. In a Recast setup, the background and data paths are archived (e.g. by preserving the partial likelihood created from the background estimates and the observed data), and the signal path is fully preserved such that it can be re-run at any time. Figure recreated from Ref. [249].

free parameters that need to be varied. Still, it is very likely that a given analysis is sensitive to a variety of different BSM scenarios not considered in the original publication.

Consequently, it is not surprising that there is significant interest in the HEP community to reinterpret BSM searches in different signal models. Reinterpretations of published ATLAS searches for SUSY are performed both within as well as outside of the ATLAS collaboration. For the HEP community outside of the experimental collaborations, the results published by the analyses performed by the collaborations represent the only available windows into the dataset recorded, in the context of BSM scenarios like SUSY. Reinterpretations of reproducible analyses are thus the only possibility to determine the implications of LHC data for a variety of models [248]. Likewise, within the experimental collaborations, reinterpretations can additionally serve as a powerful tool for shaping the search program. Reinterpretations of ATLAS SUSY searches in more complete SUSY models like the pMSSM not only allow to state a combined sensitivity of ATLAS to more realistic SUSY models, but also enable the collaboration to identify potential blind spots and parameter regions still uncovered by existing analyses. Such reinterpretations have been done in the past [76, 73] for the Run 1 dataset. Similar efforts aiming to reinterpret the current ATLAS search for SUSY in the pMSSM using the full Run 2 dataset are currently ongoing.

9.1.2 Approaches for reinterpretations

As the event selection of an analysis is fixed, the pre-fit background estimates and observed data in the regions of interest targeted do not change. Hence, the data and background pipelines, shown in fig. 9.1 and entering the statistical inference only through event rates, can be archived in a reusable format significantly smaller than the original input data. Reinterpreting a search in the light of a new signal model consequently only requires to re-execute two of the main analysis ingredients with (partially) new inputs; the signal pipeline and the statistical inference.

Recently, it has become technically possible to directly preserve the partial analysis likelihood built from the background estimates and observed data, including all auxiliary data and details of the statistical model used for inference. Once the signal estimates are known, a new full analysis likelihood can be built, and the viability of the new signal model can be tested. The publication of likelihood of the 1ℓ search is further discussed in section 9.2.

Different approaches can be taken for deriving signal estimates for a new SUSY scenario of interest. Manifestly the most precise approach involves running the original analysis software, but using a different BSM model as input. As this requires to preserve the entirety of the original software environment including workflows used in the analysis, this is arguably the most involved approach, especially since it involves executing the computationally expensive detector simulation. A framework designed to facilitate such an effort, called Recast and originally proposed in Ref. [250], is under development and aims to provide the cyber-infrastructure necessary for reinterpretations as a service. Through a web interface, physicists would provide an alternative BSM model and request a reinterpretation of a search, triggering a computational workflow executing the original analysis and delivering the *recasted* results. Section 9.3 discusses an attempt to fully preserve the 1ℓ search presented in this thesis using the Recast paradigm.

As the details of the existing Recast implementations of ATLAS searches for SUSY are not publicly available in their entirety but only meant to be interacted with through a Recast request, the exact original implementation of the analysis selection is in general not readily available. For this reason, a number of public tools aiming to reimplement an approximated version of the event selections of a number of BSM searches are available. Prominent examples include CheckMate [251, 252] and MadAnalysis5 [253]. ATLAS has internally maintained a similar catalogue of its SUSY analyses and is publishing event selection snippets in C++ for many SUSY searches on HEPData [254]. Recently, this package maintained by ATLAS, called SimpleAnalysis [255], has been made publicly available, allowing the C++ snippets published to be executed outside the collaboration.

A crucial step necessary for achieving a reliable reimplementation of the signal pipeline is the detector simulation. Executing the full detector simulation requires access to the collaborations's detector description and is computationally expensive, making it unfeasible to be used in the context of large-scale reinterpretations. For this reason, it is often approximated using simplified detector geometries and granularities. The most commonly used package for fast detector simulation outside of the ATLAS collaboration is Delphes [256], which is used in e.g. CheckMate and MadAnalysis5. Other packages like e.g. Rivet [257, 258] approximate the detector response using dedicated four-vector smearing techniques, assuming that the detector response roughly factorises into the responses of single particles. Internally, ATLAS also uses a dedicated framework for four-vector smearing, used in scenarios where other fast simulation techniques are still too expensive. Section 9.4.2 discusses these dedicated smearing functions further.

Finally, instead of trying to estimate the signal rates of a new signal model using MC simulation and (reimplemented) analysis event selections, some reinterpretation efforts like e.g. SModelS [259, 260] use *efficiency maps* encoding the selection efficiency of the analysis as a function of some of the analysis observables (typically the sparticle masses). Such efficiency maps are routinely published by the ATLAS SUSY searches on HEPData, and allow for efficient reinterpretations as long as the signal efficiencies mostly depend on the signal kinematics and are largely independent from the specific details of the signal model [259]. For the analysis presented in the previous part of this work, the efficiency maps and further analysis data products are available at Ref. [261].

9.2 Public full likelihood

The likelihood is arguably one of the most information-dense, and thus valuable, data products of an analysis. In reinterpretation efforts, approximations need to be made for the statistical inference, in terms of e.g. correlations between event rate estimates as well as the treatment of uncertainties, if the exact likelihood function of the original analysis is not known. Recently, in an extraordinary step towards open and reproducible science, ATLAS has started to publish full analysis likelihoods built using the HistFactory pdf template introduced in chapter 3 [147]. This effort has been facilitated by the development of `pyhf` in conjunction with the introduction of a `JSON` specification fully describing the HistFactory template. As a pure-text format, the `JSON` likelihoods are human- and machine-readable, highly compressible and can easily be put under version control, all of which are properties that make them ideal for long-term preservation, which is crucial for reinterpretations.

The full likelihood of the 1ℓ search has been published [262] and is not only heavily used in the following chapters, but also in various analysis reinterpretation and combination efforts currently ongoing in ATLAS. Furthermore, several efforts outside of the ATLAS collaboration have already included the analysis likelihood into their reinterpretations. The SModelS [263] and MadAnalysis5 [264, 265] efforts have both reported significant precision improvements through the use of the full likelihood. Furthermore, the full likelihood of the search presented herein has recently been used to demonstrate the concept of scalable distributed statistical inference on high-performance computers (HPCs) [266]. Through the `funcX` package [267], `pyhf` is used as a highly scalable *function as a service* to fit the entire signal grid of 125 signal points with a wall time of 156 s using 85 available worker nodes[†].

9.3 Full analysis preservation using containerised workflows

For an analysis to be fully re-usable under the Recast paradigm, the signal pipeline of the original analysis (cf. fig. 9.1) needs to be preserved such that it can be re-executed on new inputs. As typically only the processing steps after the event reconstruction are analysis-specific, it is sufficient to preserve this part of the signal pipeline. Processing steps preceding the calibration and selection of physics objects only involve the central ATLAS production system and result in a *derived analysis object data* format[§] that are used by analyses. These processing steps are preserved using centrally provided infrastructure.

In the following, the term *signal analysis* will refer to the analysis-specific processing steps that are not handled by the central ATLAS production system, typically starting with selection of events in the *derived analysis object data* format that have passed the reconstruction step in fig. 9.1. Preserving the signal analysis not only needs preservation of the full software environment for the different processing steps, but also knowledge of the correct usage of the software through parameterised job templates together with a workflow graph connecting the

[†] Theses benchmarks use `pyhf`'s NumPy backend and SciPy optimiser, a combination that has a slower log-likelihood minimisation time than e.g. PyTorch coupled with SciPy, as will be shown in section 10.3.

[§] The derived analysis object data formats are generated from the original reconstructed datasets recorded or simulated. Their main goal is to reduce the input size that needs to be read by any given analysis by including only a minimal set of objects needed in the analysis. Many such derived formats exist in ATLAS, each tailored to suit the needs of a certain set of analyses.

different processing steps. A full graph representation of the entire analysis, implemented in RECAST is shown in fig. 9.2.

9.3.1 Software preservation

As much of the software is only tested, validated and deployed on a narrow set of architectures and platforms, the full software environment defining an analysis pipeline not only includes the original analysis-specific code used for object definitions, calibrations, event selection and statistical inference, but also the operating system used and a number of low-level system libraries that the applications depend upon. Preserving the full software environment can be achieved through the use of *Docker containers* [268, 269] that—except for the operating system kernel—are able to package the full software environment in a portable data format, including a layered file system, the operating system as well as the actual application and all of its dependencies. As opposed to full virtualisation, Docker containers do not rely on hardware virtualisation but share the operating system kernel with the host and thus only interact with the host through system calls to the Linux kernel [269], offering a highly stable interface. This makes Docker containers a well-suited solution for deploying isolated applications on a heterogeneous infrastructure.

Due to the software structure of the 1ℓ search, a containerisation requires a total of three container images spanning the following processing steps:

- Event selection and physics object calibration: this step reads events in the *derived analysis object data* format and produces flat ROOT files.
- Generation of expected signal rates: the histogram-building features of HISTFITTER are exploited to generate the necessary signal histograms in the relevant selections including all systematic variations. The histograms are subsequently converted into a JSON patch file that can be used to patch the partial likelihood and create a full analysis likelihood function.
- Statistical inference: although the original analysis used HISTFITTER for the statistical inference, the RECAST implementation uses the `pyhf`-implementation of the HISTFACTORY models in order to benefit from the possibility of using a partial `JSON` likelihood to preserve background and data rates. Studies have shown that the HISTFITTER and `pyhf` implementations of the statistical inference have been shown to produce exactly the same results up to machine precision (cf. e.g. Ref. [147]).

The first Docker image is based on a *base image* providing a fixed ATLAS software release including all dependencies, expanded with the relevant analysis software. The second docker image uses the ROOT installation version originally used in the analysis, provided as part of a suitable ATLAS software release. The last image is based on a `pyhf` base image containing the `pyhf` release version used when validating the two HISTFACTORY implementations against each other in the context of the 1ℓ search. All docker images are subject to version control and continuous integration, such that changes to the underlying software environment can be tracked and tagged. This allows for a consistent preservation of multiple versions of the analysis pipeline.

9.3.2 Processing steps preservation

Preserving the software environment is not sufficient, as detailed instructions on how to use it have to be given to the user. This is achieved through parameterised job templates that specify the precise commands and arguments required to re-execute the analysis code for specific processing steps. As re-executing the analysis pipeline using different signal models involves varying input parameters, all job template parameters are exposed to the user. Within RECAST, the job templates are formulated using the YAML syntax. In fig. 9.2, the parameterised job templates including their output are shown as dashed boxes, connected together through a workflow specification represented via black arrows.

User-specifiable arguments and inputs to the event selection and physics object calibration step include the actual reconstructed MC events in *derived analysis object data* format, obtained through the central ATLAS production system, as well as corresponding files necessary for the pile-up correction in MC. In addition, the signal process cross section as well as MC generator-level efficiencies need to be given for correct normalisation the estimated signal rates to the integrated luminosity of the full Run 2 dataset. For each new signal model to be tested, three MC samples need to be provided, generated with specific pile-up profiles close to the pile-up profile in data during the 2015–2016, 2017 and 2018 data-taking periods, respectively[†]. In all three jobs, the events processed are weighted according to the integrated luminosity of the data-taking period they represent within the full Run 2 dataset. A subsequent *merging* step uses the same docker image as the previous processing step, and serves to merge the three produced outputs into a single ROOT file that can be read by the subsequent step.

Apart from the merged ROOT output file produced in the previous step, the generation of the expected signal rates in a JSON patch format requires only one additional input—a JSON file containing theory uncertainties on the expected signal rates. These are optional and do not have to be specified if deemed to be negligible for the signal model to be tested.

The statistical inference steps requires the signal JSON patch from the previous step as well as the archived partial likelihood containing observed data as well as expected background rates including systematic variations thereof.

9.3.3 Workflow preservation

Finally, the preserved processing steps need to be linked together, creating a parameterised workflow completely defining the analysis pipeline from centrally produced MC datasets to the statistical inference results. Within RECAST, this is achieved using the workflow description language yadage [270], capturing the full workflow in YAML format. The workflow uses the job templates and defines their processing order and dependencies.

The RECAST implementation of the analysis presented in this work has been validated against original analysis inputs. The expected and observed CL_s values derived in the original analysis could be re-derived using the containerised workflow implementation. On a non-isolated CPU, the full preserved analysis pipeline for a single signal model can be executed within 1 hour. Due to the highly portable nature of the containerised workflow, the pipeline can easily be run in a distributed setup, allowing scalable reinterpretations at full analysis precision.

[†] This allows to have pile-up weights relatively close to unity, avoiding unnecessary statistical dilution.

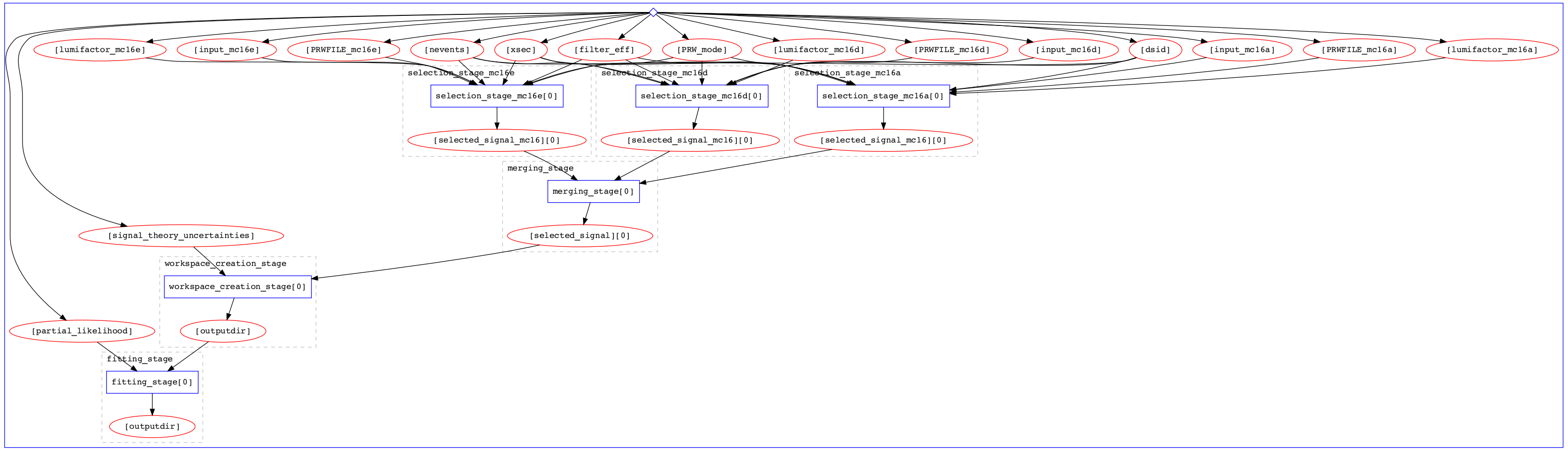

Figure 9.2: Graph of the workflow as specified for the analysis pipeline. The containerised processing steps are represented as blue rectangular nodes, while input parameters, input files and outputs are shown as red oval nodes. The workflow is comprised of four processing steps: `selection_stage_mc16(a,d,e)`, `merging_stage`, `workspace_creation_stage` and `fitting_stage`. The first two steps perform the object calibration, event selection and merging of the three MC datasets representing the three data-takin periods 2015–2016, 2017 and 2018. The latter two steps implement the generation of the signal `JSON` patch as well as the final statistical inference. Compared to fig. 9.1 the first two steps implement the *signal analysis* part, while the latter two steps implement the *statistical inference* deriving the final results.

9.4 Truth-level analysis

A full preservation of the entire analysis pipeline is highly desirable as it allows for a maximum precision reinterpretation of the original analysis in a new, promising signal model. As the full detector simulation needs a significant amount of CPU resources in addition to the non-negligible wall time of the actual preserved analysis pipeline, this approach can only be used on a limited set of models. In large-scale reinterpretations over high-dimensional parameter spaces, the amount of unique models that need to be sampled and investigated using the analysis is too high to employ the fully preserved analysis pipeline. In order to significantly reduce the number of models that need to be passed through the full analysis pipeline, a pre-sorting of the models needs to be performed, filtering models for which (non-)exclusion based on a simplified analysis implementation is uncertain.

In the following, two major, complementary approaches to analysis simplifications are discussed, targeting both the *signal pipeline* as well as the *statistical inference* blocks in fig. 9.1. This section discusses the SimpleAnalysis implementation of the analysis, an approach implementing the signal pipeline at *truth-level*, i.e. using the generator-level objects without running a detector simulation. An approximation of the detector response is performed using four-vector smearing.

Chapter 10 introduces a procedure for building simplified likelihoods out of the published full likelihoods of ATLAS SUSY searches in order to significantly lower the wall time needed for running statistical fits in an analysis. In chapter 11, both approximations are combined into a *simplified analysis pipeline* and applied on a set of SUSY models sampled from the pMSSM.

9.4.1 Truth-level selection

All signal and control regions considered in the original 1-lepton search are implemented at truth-level using the publicly available framework SimpleAnalysis. The exact implementation is publicly available at Ref. [261] and was already used in chapter 7 for the derivation of some of the theory uncertainties in the full analysis.

The truth-level implementation fully specifies all object definitions introduced in section 4.4 even though some of them, like e.g. lepton isolation, are technically not well-defined at truth-level. The four-vector smearing subsequently described is, however, in many cases implemented as a function of said object definitions and thus allows to consider them nonetheless. Additionally, as discussed in section 9.1, the full specification of the original analysis event selection including all object definitions allows for simpler reinterpretations by efforts outside of the ATLAS collaboration that generally do not have access to the original analysis software.

Following the object definitions, an overlap removal procedure following the same prescription as described for the reconstruction-level analysis is performed, i.e. especially also using the same shrinking cone definitions introduced in section 4.5. The overlap removal step removing electrons sharing a track with a muon is approximated by using a distance parameter of $\Delta R = 0.01$ between the objects. Although often neglected[†] in reinterpretation efforts outside of the collaboration, the correct implementation of the overlap removal procedure employed in the original analysis is crucial to reproduce the signal estimates of the original analysis. Figure 9.3

[†] The overlap removal procedures in ATLAS SUSY searches tend to be quite intricate, making them non-trivial to re-implement without ATLAS and analysis-specific knowledge.

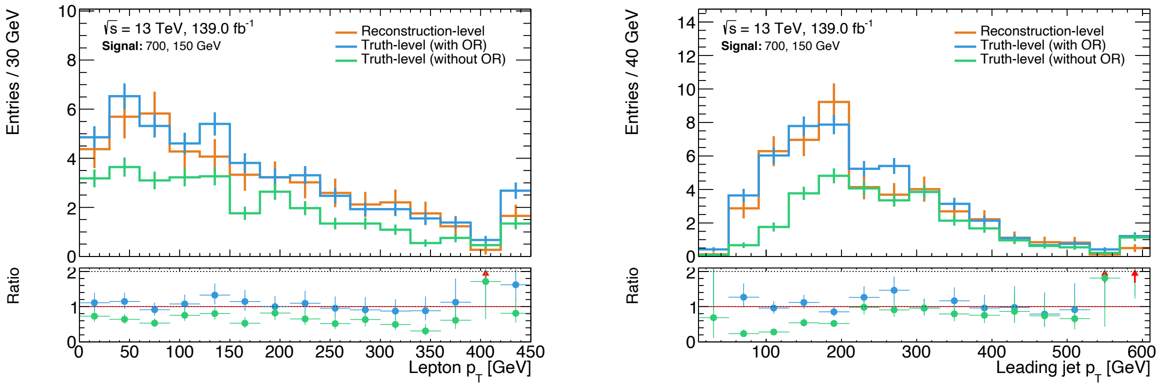

Figure 9.3: Impact of the overlap removal (OR) procedure at truth-level illustrated in the lepton and leading jet transverse momenta distributions. The truth-distribution without overlap removal (green) generally underestimates the number of signal events at reconstruction-level (orange) due to a number of events not meeting the final state criteria. Correct overlap removal procedure at truth-level (blue) improves the agreement. The exemplary benchmark signal point with $m(\tilde{\chi}_1^\pm/\tilde{\chi}_2^0), m(\tilde{\chi}_1^0) = 700, 150\,\text{GeV}$ is shown in both plots (at truth- and reconstruction-level). All distributions are shown in a loose preselection requiring exactly one lepton, $E_\text{T}^\text{miss} > 50\,\text{GeV}$, $m_\text{T} > 50\,\text{GeV}$, and 2–3 jets, two of which need to be b-tagged.

illustrates that, not implementing the overlap removal procedure of the original 1ℓ search results in many truth-level events not passing the analysis selections because of too many objects in the final state.

Finally, the exact implementation of all analysis observables is explicitly given in the SimpleAnalysis implementation, followed by the definition of all control and signal regions.

9.4.2 Truth smearing

The general assumption of the truth smearing applied in the following is that the detector response roughly factorises into the responses of single particles. This allows to use the ATLAS detector performance results in order to construct detector response maps parameterised in different observables for each physics object. Detector response maps include object reconstruction and identification efficiencies as well as scale factors to correct for differences between MC and observed data. Likewise, effects from the finite resolution of energy measurements in the detector are modelled through energy resolution maps. In the following, the four-vector components of electrons, muons, jets and E_T^miss are smeared.

In the case of truth electrons, the identification efficiencies considered are parameterised in η and p_T as well as the identification working point used. In η, nine fixed-width bins are used. In p_T, six bins are implemented and a linear interpolation between two adjacent p_T-bins is used to get the efficiency for the given p_T of each truth electron. The probability of finding a fake electron in a truth jet is estimated through a similar two-dimensional map depending on the truth jet η and p_T, again using fixed-width bins in η and a linear interpolation in p_T. The range of the p_T interpolation for identification efficiencies and fake rates extends from 7 GeV to 120 GeV. If the truth p_T of the electron is outside of that range, the identification efficiency and fake rate from the respective bound of the corresponding η-bin are used. The probability for misidentifying an electron as a photon is estimated using different fixed values for the barrel and end-cap regions. Finally, the transverse energy of the electron is smeared using a random number drawn from a Gaussian distribution with standard deviation corresponding to the η- and p_{T}-dependent energy resolution.

For truth muons, the identification efficiencies are also parameterised in η and p_{T} as well as the identification working point used. Similar to truth electrons, the p_{T} of the muon is smeared using a Gaussian distribution with standard deviation corresponding to the momentum resolution. The momentum resolution of combined truth muons, σ_{CB}, is computed from the measured resolutions in the ID, σ_{ID}, and MS, σ_{MS}, as

$$\sigma_{\mathrm{CB}} = \frac{\sigma_{\mathrm{ID}}\sigma_{\mathrm{MS}}}{\sqrt{\sigma_{\mathrm{ID}}^2 + \sigma_{\mathrm{MS}}^2}}, \tag{9.1}$$

where σ_{ID} and σ_{MS} are parameterised in η and p_{T}.

The transverse momentum of truth jets is smeared using a Gaussian with standard deviation equal to the JER, provided in a map parameterised in five bins in $|\eta|$, ranging from $|\eta| = 0$ to $|\eta| = 4.5$. Following Ref. [216], jet energy resolutions are provided using parameterisations of a noise N, stochastic S and constant C term for each of the seven bins in $|\eta|$, such that the resolution can be computed as

$$\frac{\sigma(p_{\mathrm{T}})}{p_{\mathrm{T}}} = \frac{N}{p_{\mathrm{T}}} \oplus \frac{S}{\sqrt{p_{\mathrm{T}}}} \oplus C. \tag{9.2}$$

Only truth jets with $10\,\mathrm{GeV} < p_{\mathrm{T}} < 1.5\,\mathrm{TeV}$ are smeared. For truth jets with $p_{\mathrm{T}} > 20\,\mathrm{GeV}$, the flavour tagging efficiency is considered using efficiencies parameterised in η, p_{T} and the MV2c10 working point (introduced in section 4.4) used, measured in fully reconstructed simulated $t\bar{t}$ events [222].

Finally, the smeared missing transverse energy is computed using the transverse momenta of all smeared truth objects in the event, including an approximation for the track soft term. The latter is approximated using resolution measurements from $Z \to \ell\ell$ events [225], allowing to infer a distribution of the mean soft term projected in the direction longitudinal to the total transverse momentum of all hard objects in an event, $\boldsymbol{p}_{\mathrm{T}}^{\mathrm{hard}}$. The measured resolution parallel and perpendicular to $\boldsymbol{p}_{\mathrm{T}}^{\mathrm{hard}}$ is then used to smear the nominal soft track value.

9.5 Validation of the truth-level analysis

9.5.1 Validation in loose preselection

The performance of the truth smearing is illustrated in a loose preselection for a single exemplary benchmark signal point in fig. 9.4. The loose preselection applied requires exactly one lepton, $E_{\mathrm{T}}^{\mathrm{miss}} > 50\,\mathrm{GeV}$, $m_{\mathrm{T}} > 50\,\mathrm{GeV}$, and 2–3 jets, two of which need to be b-tagged. The *reconstruction-level* distributions, i.e. the distributions obtained with MC simulated inputs for which the full detector simulation and reconstruction has been run, are compared with the truth-level distributions before and after truth smearing. It can clearly be observed that the

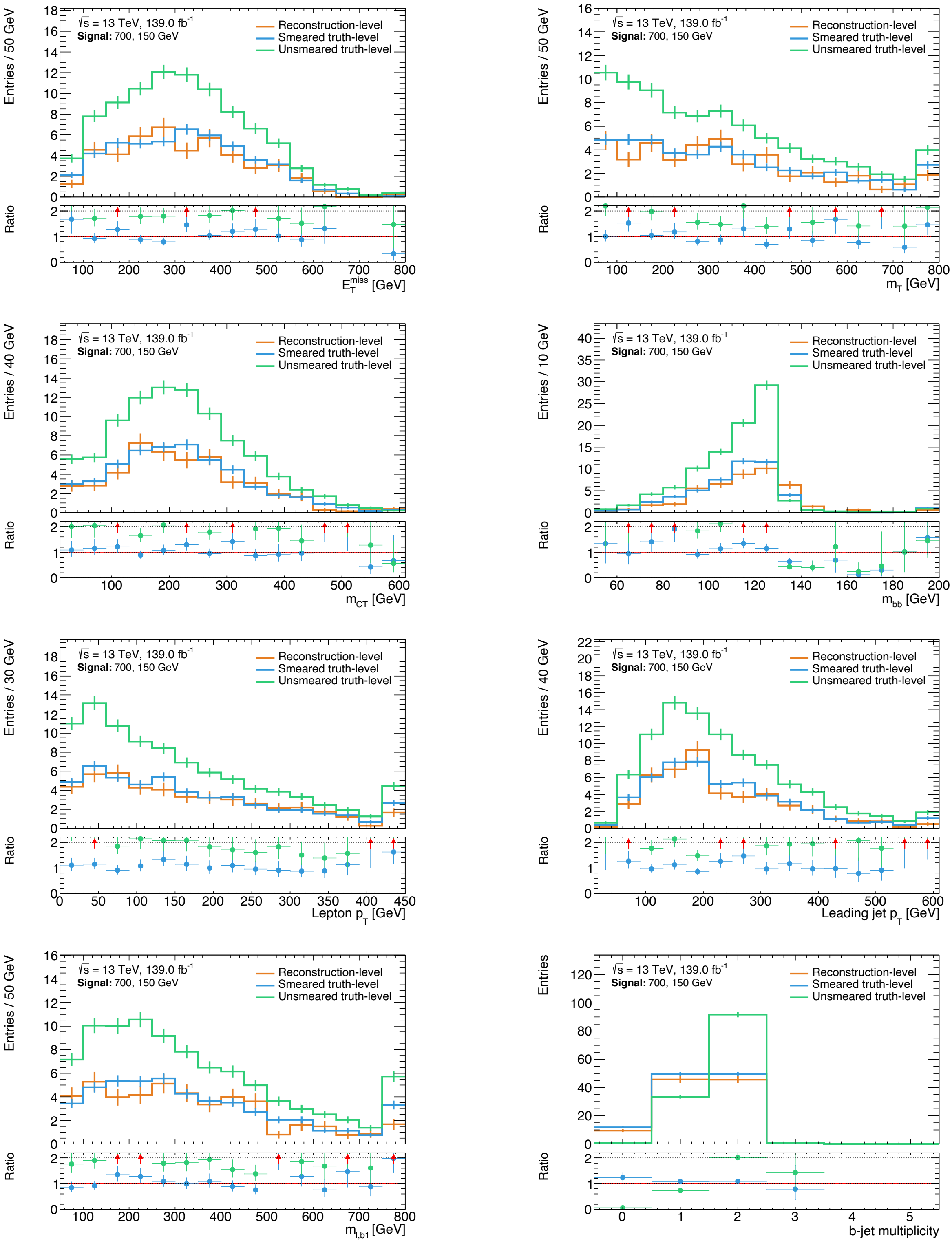

Figure 9.4: Comparisons of the kinematic distributions of key observables at (smeared) truth- and reconstruction-level. The exemplary benchmark signal point with $m(\tilde{\chi}_1^{\pm}/\tilde{\chi}_2^0), m(\tilde{\chi}_1^0) = 700, 150\,\mathrm{GeV}$ is shown. The ratio pad shows the ratio between smeared and unsmeared truth-level distributions (blue and green) to reconstruction-level distributions (orange). Only MC statistical uncertainty is included in the error bars. All distributions are shown in a loose preselection requiring exactly one lepton, $E_{\mathrm{T}}^{\mathrm{miss}} > 50\,\mathrm{GeV}$, $m_{\mathrm{T}} > 50\,\mathrm{GeV}$, and 2–3 jets, two of which need to be b-tagged. The latter requirement is dropped for the b-jet multiplicity distribution.

truth smearing noticeably improves the agreement between the truth- and reconstruction-level distributions. While the lepton and jet reconstruction and identification efficiencies are—due to their dependence on η, p_{T} and individual working points—crucial for the overall agreement in shape, the inclusion of flavour-tagging efficiencies significantly improves the overall agreement in normalisation.

Although some minor differences remain, overall a good agreement is observed across all relevant kinematic distributions at loose preselection level. Most of the differences between smeared truth-level and reconstruction-level distributions in individual bins are well within the MC statistical uncertainties arising from the relatively limited MC statistics available.

9.5.2 Validation in signal regions

As the expected signal rates in the signal regions are ultimately what is entering the statistical inference, it is important that the good agreement observed at preselection is still present in the kinematically tighter selections of the signal regions. Additionally, it is worth investigating the agreement across all signal models considered in the original analysis, as opposed to only validating specific benchmark points. A comparison of the reconstruction-level and truth-level event rates before and after smearing in the signal regions SR-LM, SR-MM and SR-HM is shown in fig. 10.5 for all signal models considered in the 1-lepton analysis. For the sake of conciseness, only the cumulative m_{CT} bins are shown in each SR in fig. 10.5. The agreement in the individual m_{CT} bins in each SR-LM, SR-MM and SR-HM is provided in figs. C.1 to C.3.

The truth smearing drastically improves the agreement in event rate estimates at truth- and reconstruction-level across all SR bins considered. While the event rates are generally overestimated at truth-level before smearing, compared to reconstruction-level, both tend to agree well within statistical uncertainties after smearing.

9.5.3 Validation using likelihood

Using the nominal expected event rates at smeared truth-level for every signal model in the original signal grid considered in the 1ℓ search, expected and observed CL_s values can be computed and exclusion contours can be derived. Figure 10.5 compares the expected and observed exclusion contours obtained using the full likelihood and reconstruction-level signal inputs with those obtained using the full likelihood and truth-level signal inputs before and after truth smearing. While all theory and systematic uncertainties on the signal are included in the reconstruction-level contours, no signal uncertainties are considered when obtaining both the smeared and unsmeared truth-level contours. As expected from the previous validation steps in the signal regions, the sensitivity using unsmeared truth-level signal inputs is significantly overestimated compared to the published analysis exclusion limit using reconstruction-level inputs. The smeared truth-level inputs, however, yield exclusion contours with an acceptable match compared to the reconstruction-level results.

In summary, this validation process at multiple selection levels of the analysis shows that the signal pipeline can be approximated using a truth-level analysis and dedicated smearing functions, allowing to produce signal event rate estimates with high computational efficiency. In large-scale reinterpretations, this approach can be used as a basis for an efficient classification

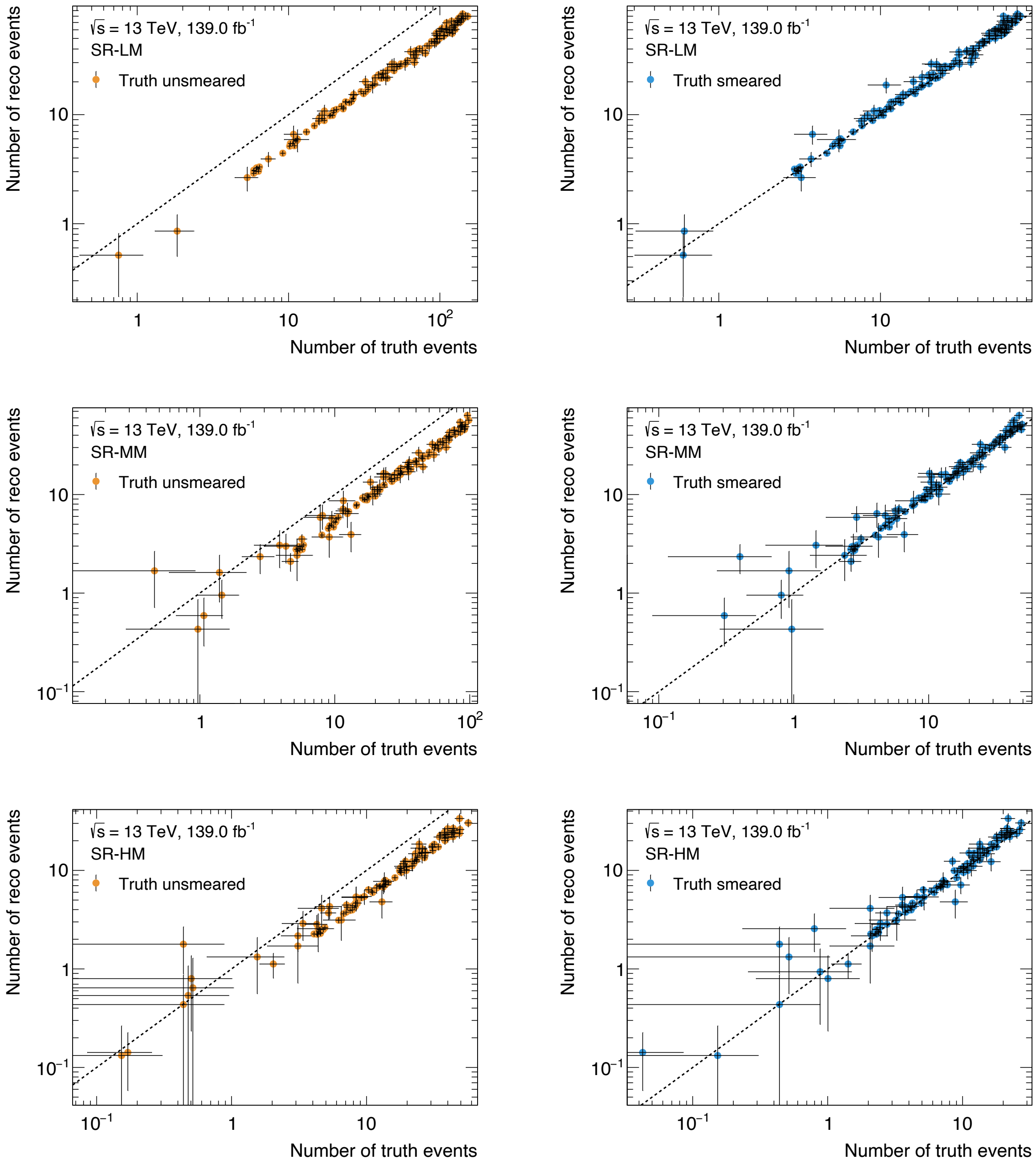

Figure 9.5: Comparison of the event rates at truth- and reconstruction-level before (left) and after (right) truth smearing. From top to bottom, the SR-LM, SR-MM and SR-HM signal regions are shown, with cumulative (integrated) m_{CT} bins. Every single point in the scatter plots represents a single signal model considered in the original 1-lepton analysis. Uncertainties include MC statistical uncertainties.

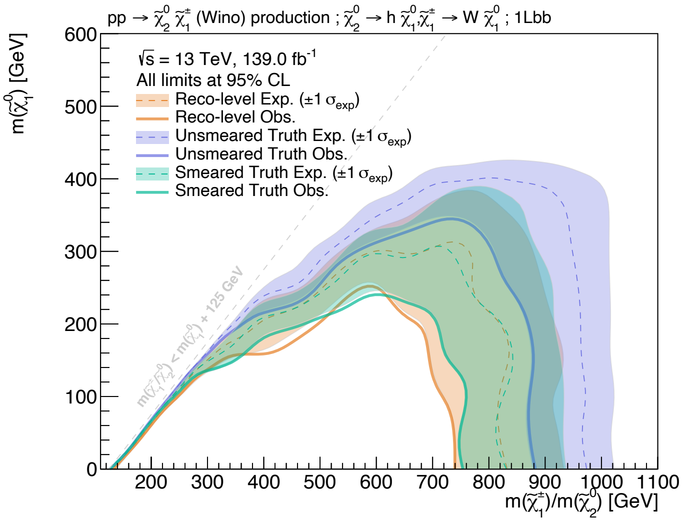

Figure 9.6: Expected and observed exclusion contours obtained with the full and likelihood using reconstruction-level inputs (orange) as well as truth-level inputs before (purple) and after (green) smearing. Uncertainties include all statistical and systematic uncertainties on the background and signal for the reconstruction-level contours, but only statistical and systematic uncertainties on the background for truth-level signal inputs.

of models into safely excluded and non-excluded models as well as models where exclusion is in doubt and where the full analysis pipeline using Recast is needed.