
FAIR Universe HiggsML Uncertainty Dataset and Competition

Lisa Benato⁸, Wahid Bhimji¹, Paolo Calafiura¹, Ragansu Chakkappai^{2,7}, Po-Wen Chang¹, Yuan-Tang Chou³, Sascha Diefenbacher¹, Jordan Dudley^{1,4}, Ibrahim Elsharkawy⁵, Steven Farrell¹, Aishik Ghosh^{1,5}, Cristina Giordano⁸, Isabelle Guyon⁷, Chris Harris¹, Yota Hashizume⁹, Shih-Chieh Hsu³, Elham E Khoda^{1,3,10}, Claudius Krause⁸, Ang Li⁸, Benjamin Nachman¹, Peter Nugent¹, David Rousseau^{2,7}, Robert Schoefbeck⁸, Maryam Shooshtari⁸, Dennis Schwarz⁸, Benjamin Thorne¹, Ihsan Ullah⁷, Daohan Wang⁸, and Yulei Zhang³

¹Lawrence Berkeley National Laboratory

²Université Paris-Saclay, CNRS/IN2P3, IJCLab

³University of Washington, Seattle

⁴University of California, Berkeley

⁵University of Illinois Urbana-Champaign

⁶University of California, Irvine

⁷ChaLearn

⁸Institute for High Energy Physics, Vienna

⁹Kyoto University

¹⁰University of California, San Diego

Abstract

The FAIR Universe – HiggsML Uncertainty Challenge focused on measuring the physical properties of elementary particles with imperfect simulators. Participants were required to compute and report confidence intervals for a parameter of interest regarding the Higgs boson while accounting for various systematic (epistemic) uncertainties. The dataset is a tabular dataset of 28 features and 280 million instances. Each instance represents a simulated proton-proton collision as observed at CERN’s Large Hadron Collider in Geneva, Switzerland. The features of these simulations were chosen to capture key characteristics of different types of particles. These include primary attributes, such as the energy and three-dimensional momentum of the particles, as well as derived attributes, which are calculated from the primary ones using domain-specific knowledge. Additionally, a label feature designates each instance’s type of proton-proton collision, distinguishing the Higgs boson events of interest from three background sources. As outlined in this paper, the permanent release of the dataset allows long-term benchmarking of new techniques. The leading submissions, including Contrastive Normalising Flows and Density Ratios estimation through classification, are described. Our challenge has brought together the physics and machine learning communities to advance our understanding and methodologies in handling systematic uncertainties within AI techniques.

1 Introduction

1.1 Background and impact

For several decades, the discovery space in almost all branches of science has been accelerated dramatically due to increased data collection brought on by the development of larger, faster instruments.

More recently, progress has been further accelerated by the emergence of powerful AI approaches, including deep learning, to exploit this data. However, an unsolved challenge that remains, and *must* be tackled for future discovery, is how to effectively quantify and reduce uncertainties, including understanding and controlling *systematic* uncertainties (also named *epistemic* uncertainties in other fields). A compelling example is found in analyses to further our fundamental understanding of the universe through analysis of the vast volumes of particle physics data produced at CERN, in the Large Hadron Collider (LHC) [1]. Ten years ago, part of our team co-organised the Higgs Boson Machine Learning Challenge (HiggsML [2, 3], the most popular Kaggle challenge at the time attracting 1785 teams. This challenge has significantly heightened interest in applying Machine Learning (ML) techniques within High-Energy Physics (HEP) and, conversely, has exposed physics issues to the ML community. Whereas previously, the most effective methods predominantly relied on boosted decision trees, Deep Learning has since gained prominence (see, e.g., HEP ML living review [4]). While the LHC has not (yet) discovered new physics beyond the Higgs boson, it has accumulated vast data and will continue to accumulate more data well into the next decade. There is a discovery potential in very precise measurements of particle properties, particularly of the Higgs boson.

High-energy physics relies on statistical analysis of aggregated observations. Therefore, the interest in uncertainty-aware ML methods in HEP is nearly as old as the application of ML in the field. Advanced efforts that integrate uncertainties into the ML training include approaches that explicitly depend on nuisance parameters [5–14], that are insensitive to nuisance parameters [15–32], that use downstream test statistics in the initial training [33–43], and that use Bayesian neural networks for estimating uncertainties [44–47]. Many of these topics were covered in recent forward-looking review-type articles in Refs. [48, 49]. However, these developments all report technique performance on different ad-hoc datasets, so it is difficult to compare their merits. The Fair Universe HiggsML Uncertainty Challenge, an official NeurIPS 2024 competition, aimed to provide a common ground, with a dataset of sufficient complexity, equipped with systematic bias parameterisations, and a metric.

We aim to address the issue of systematic uncertainties within a specific domain. Yet, the techniques developed by the challenge participants will apply to identifying, quantifying, and correcting systematic uncertainties in other areas, particularly other science disciplines.

1.2 Novelty

This entirely new public competition has built on our experience running several competitions in particle physics and beyond. These include the original HiggsML challenge [2], the TrackML Challenges (NeurIPS 2018 competition) [50, 51], the LHC Olympics [52], AutoML/AutoDL [53, 54], and other competitions. Building on the foundation of the HiggsML challenge, this competition introduces a significant change by using simulated data that includes biases (or *systematic effects*). In addition, participants were asked to provide a confidence interval and not just a point estimate.

While there have been previous challenges focusing on meta-learning and transfer-learning, such as the NeurIPS 2021 and 2022 meta-learning challenges [55, 56], Unsupervised and Transfer Learning [57], challenges related to bias e.g. Crowd bias challenge [58], and those addressing distribution shifts, like the Shifts challenge[59] series, and CCAI@UNICT 2023 [60], this is the first challenge and dataset that requires participants to handle systematic uncertainty. Moreover, this project is connecting the Perlmutter system at NERSC [61], a large-scale supercomputing resource featuring over 7000 NVIDIA A100 GPUs, with Codabench [62], a new version of the renowned open-source benchmark platform CodaLab [63, 64]. Due to its complexity, the process of generating events was computationally intensive; use of the Perlmutter supercomputer allowed us to create a vast amount of data – hundreds of millions of events compared to less than a million events for the HiggsML competition, which will serve as a long-lasting benchmark.

2 Data

The dataset is publicly available on the Zenodo platform [65]. The data is saved as a tabular parquet [66] file of 16 GB and is accompanied by a Croissant JSON metadata file. The dataset comprises 280M simulated proton-proton collision events and is weighted to represent two weeks of LHC data taking. A separate 120M i.i.d dataset has been used for the final results in section 5 and is kept private for future over-training checks.

We are using a simulated particle physics dataset for this competition to produce data representative of high-energy proton collision data collected by the ATLAS experiment [67] at the LHC. The dataset [65] was created with two widely-used simulation tools, Pythia 8.2 [68] and Delphes 3.5.0 [69]; all the configuration and data pre-selection code is available from [70]. This required 1.8 million CPU core hours. We have organised the dataset into a tabular format where each row corresponds to a collision event and each of the 28 columns corresponds to a feature. The detailed dataset description is in [Appendix A](#), [Appendix B](#) and [Appendix C](#); it is mostly taken from the public unpublished Fair Universe whitepaper [71] which served as detailed documentation for the competition. Part of the features are primary features, essentially the energy and direction of a small set of particles, and the remainder are derived features, computed from the primary ones with domain knowledge. The events are divided into two categories (see [Table 1](#)): signal and background. The signal category includes collision events with a Higgs boson decaying into pairs of tau particles (see [Figure 7](#)) (one decaying, into, in addition to neutrino(s), a light lepton, the other one into a set of hadrons hence the name hadronic tau), while the background category includes other processes (subcategories) leading to a similar final state, but without an intermediate Higgs boson.

In addition, we provide a biasing script capable of manipulating a dataset by introducing six parameterised distortions as a function of six corresponding *Nuisance Parameters*¹ (the systematic biases); see details in [Appendix D](#). For example, a detector miscalibration can cause a bias in other features in a cascading way, or in another case, the magnitude of a particular background (e.g. the $t\bar{t}$) contribution can change so that the feature distributions can be different. In both cases, the inference would be done on a dataset not i.i.d. to the training dataset.

3 Tasks and application scenarios

The participant’s objective is to develop an estimator for the number of Higgs boson events in a dataset analogous to results from LHC experiments. Such a measurement is typical of those carried out at the LHC, which allows us to strengthen (or invalidate) our understanding of the fundamental laws of nature.

The primary metric is the *signal strength* (μ), which is the number of estimated Higgs boson events divided by the number of such events predicted by the Standard Model, which is the reference theory. The challenge involved estimating μ ’s true value, μ_{true} , which may vary from one (in practice for the challenge in the range 0.1 to 3) and is inherently unknown.

Participants were tasked with generating a 68.27% Confidence Interval (CI) for μ , incorporating both aleatoric (random) and epistemic (systematic) uncertainties rather than a single-point estimate. The six different systematic uncertainties are implemented in [Appendix D](#).

The primary simulation dataset assumes a μ of one. Participants receive a training subset, where events are labelled based on their event type (e.g. Higgs boson event). We provide a script to generate unlabelled pseudo-experiment datasets from the primary simulation dataset for any value of μ and the six systematic biases. The participant’s model should be able to reverse the process and provide a 68.27% CI on μ for any pseudo-experiment.

In a machine learning context, the task resembles a transduction problem with distribution shift: it requires constructing a μ interval estimator from labelled training data and biased unlabelled test data. One possibility is to train a classifier to distinguish Higgs boson from the background, with robustness against bias achieved possibly through data augmentation (or an adversarial approach, or black box optimisation or any other novel approach) via the provided script.

This challenge shifts focus from the qualitative discovery of individual Higgs boson events (which was the focus of our first challenge [2]) to the quantitative estimation of overall Higgs boson counts in test sets, akin to assessing disease impact on populations rather than diagnosing individual cases.

¹The name Nuisance Parameter, commonly used in the physics literature, refers to a parameter governing a specific parameterisation of a systematic bias. Nuisance Parameters can be in part constrained from the data itself. Still, the name implies that constraining them is only interesting as an auxiliary task in the process of determining a parameter of interest like the signal strength μ .

3.1 Metrics

Participants provided a model that can analyse a pseudo-experiment to determine (μ_{16}, μ_{84}) , the bounds of the 68.27% (approximately one standard deviation of a standard normal distribution) Confidence Interval (CI) for μ . The model is evaluated from the set of $[\mu_{16,i}, \mu_{84,i}]$ intervals obtained from N_{test} pseudo-experiments, see Figure 1a. The model's performance is assessed based on two criteria: **Average Interval Width** w (the smaller the better) computed as $w = \frac{1}{N_{\text{test}}} \sum_{i=1}^N |\mu_{84,i} - \mu_{16,i}|$ and the **Coverage**, the frequency with which μ_{truth} is covered by the CI (the closer to the standard 68.27% probability the better) computed as $c = \frac{1}{N_{\text{test}}} \sum_{i=1}^N 1$ if $\mu_{\text{true},i} \in [\mu_{16,i}, \mu_{84,i}]$.

A penalising function f has been defined to penalise the departure of c from the expected 68.27%, taking into account $\sigma_{68} = \sqrt{\frac{(1-0.6827)0.6827}{N_{\text{test}}}}$ the binomial statistical error on c :

$$f(c) = 1 + \mathbb{I}_{c < 0.6827 - 2\sigma_{68}} \cdot \left| \frac{c - (0.6827 - 2\sigma_{68})}{\sigma_{68}} \right|^4 + \mathbb{I}_{c > 0.6827 + 2\sigma_{68}} \cdot \left| \frac{c - (0.6827 + 2\sigma_{68})}{\sigma_{68}} \right|^3. \quad (1)$$

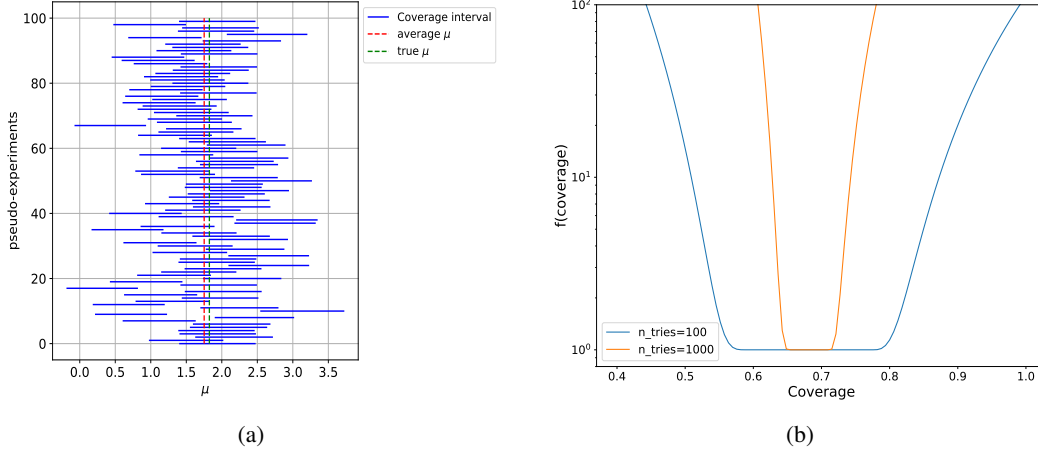


Figure 1: (1a) *Coverage plot*: all the predicted intervals (blue lines) for each pseudo experiment generated for a given μ_{true} (vertical dotted line). The coverage (here $70 \pm 5\%$) is determined by the fraction of time the horizontal blue lines intersects the vertical line. (1b) Penalising function as a function of the coverage value c , for two values of N_{test} , the number of pseudo-experiments.

We opted for an asymmetric penalty function because, within the High Energy Physics (HEP) field, overestimating uncertainty is deemed more acceptable than underestimating it [72, 73]. Hence, coverage exceeding 68.27% incurs a lesser penalty than coverage falling below 68.27%. The final **Quantile Score** (the larger the better) used to rank participants is calculated as follows:

$$\text{score} = -\ln((w + \epsilon)f(c)), \quad (2)$$

where w represents the average width of the Confidence Interval, c is the coverage, and $\epsilon = 10^{-2}$ is a regularisation term to guard against submissions that report unrealistically narrow CIs. To ensure efficient use of resources, each participant's model inference was executed across 100 pseudo-experiments times 10 trials, each with distinct values of μ_{truth} , with a time limit of 20s per inference on CPU or GPU. In the Final phase of the competition, each participant's best submission was evaluated over 100 pseudo-experiments, times 1000 trials, to minimize the statistical variance.

3.2 Limitations

The main limitation of the setup is that biases can be exactly parameterised: we are in the "known unknowns" regime. "Unknown unknowns", unexpected biases, are not covered.

The dataset has been produced using well-known standard software for event generation and detector simulation. However, a proper physics measurement would require more complex software, several

orders of magnitude slower, yielding marginally different simulated data. The methods developed on our dataset would perform equally well, provided they are fully retrained.

For each instance of the datasets, the features provided are essentially the energy and direction of a small set of particles, and derived quantities. A real physics measurement may also rely on additional quantities related to the quality of particle identification or to other particles in the same proton-proton collision. Nevertheless, the algorithms developed on our dataset should require limited added complexity to deal with additional features.

4 Software

Alongside the dataset, a GitHub repository [74] with the relevant code for reading and analysing it is made available. This includes a Jupyter notebook starting kit, simple baseline models, a small sample of the dataset, and code to compute the score.

The **Starting Kit** includes code for installing necessary packages, loading and visualising data, training and evaluating a model with the metrics described in subsection 3.1. The **Baseline** method estimates μ using standard techniques without directly addressing systematic uncertainties for simplicity. Initially, it utilises a classifier (based on an XGBoost Boosted Decision Tree) trained on a subset of the training data to enhance the signal event density and reduce the μ estimator variance. The classifier’s decision threshold is fixed heuristically. μ is then estimated from these filtered events, assuming a Poisson distribution, enabling interval maximum likelihood estimation. Further refinement involves binning events based on their classifier score and estimating μ per bin. A holdout dataset, is used to predict the amount of background and signal in each bin for $\mu = 1$. This calibration step then permits estimating μ (and the corresponding CI) on each pseudo-experiment. On Figure 2a, the alignment of maximum likelihood estimation (orange line) with unlabelled data (black line) indicates the method success, in the absence of any bias.

When unknown biases occur, the prediction on the amount of background and signal events per bin will be wrong, biasing the estimation of μ . To address the problem of systematic errors, we use the holdout dataset with biases by different amounts of the Nuisance Parameter (θ) and then build a calibration curve to estimate the signal and background in each bin. Figure 2b shows one such fit curve for the 24th bin (just as an example). Now, instead of μ depending on S and B , it will depend on fit functions $S(\theta)$ and $B(\theta)$. Finally, the minimisation function now regresses both μ and θ , thus making the model less susceptible to systematic bias. But this is only limited to one nuisance parameter; participants are encouraged to enhance the Baseline model, for instance, by modifying the architecture or training protocol to improve resilience against biases, attempting to directly model the biases, or refining the estimator through a bias-aware model.

Another way to see it is that, armed with the biasing script which can produce a dataset for any value of the six Nuisance Parameters and the signal strength μ , the participants could train a model which could regress the seven parameters for any pseudo-experiment and report the Confidence Interval on μ . This was actually done with different techniques by the winning trio (section 5).

5 Competition results and best submissions

At the end of the competition, a clear trio was at the top of the public leaderboard: HEPHY with a quantile score of 0.878, followed by Ibrahim (0.823) and Hzume (0.179). All submissions have been reevaluated on a new dataset (i.i.d. to the original one). The evaluation was done on 1000 trials of 100 pseudo-experiments (each trial with a given value of μ randomised between 0.1 and 3), instead of 10 trials for the public leaderboard. All submissions were run on the same pseudo-experiments, instead of separate pseudo-experiments for the public leaderboard.

Figure 3 shows the results for all trials for the trio. The CI width is seen falling at large values of μ : this is due to the clipping the Confidence Interval to a maximum value of 3 (which was not done in Figure 1a), which was the maximum value in this competition. Such clipping would be meaningless in the context of a real physics measurement where μ is truly unknown. This is the only "hack" specific to the competition context that could be identified. As far as the score is concerned, HEPHY and Ibrahim are very close. When merging all trials, the scores obtained by the top trio are: HEPHY

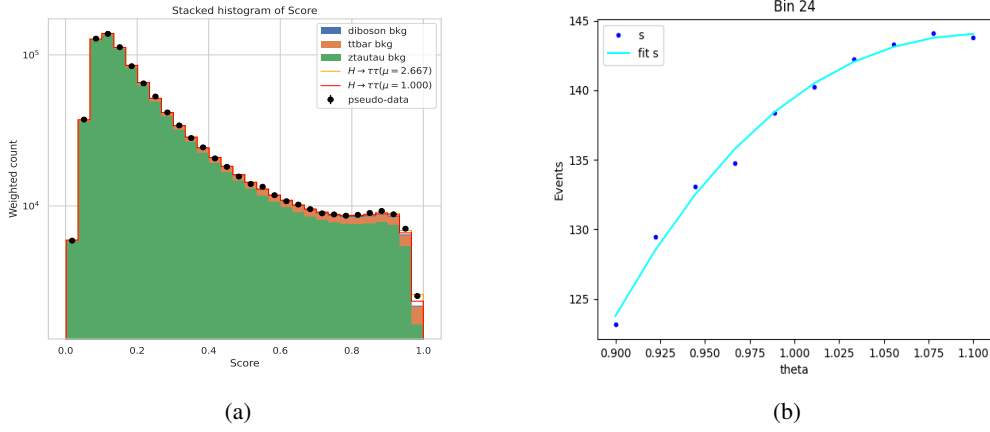


Figure 2: (a) classifier score for unlabelled test data (black points), and holdout data for (1) background events $Z \rightarrow \tau\tau$ (solid green), (2) background $t\bar{t}$ (solid orange) (3) background di-boson (solid blue, hardly visible) (4) signal events $H \rightarrow \tau\tau$ for $\mu = 1$ (red line), and (5) signal events fitted histogram to test data, leading to estimated $\mu = 2.667$ (orange line) (b) model of the bin content vs Nuisance Parameter θ for bin 24, as an example.

-0.582, Ibrahim -0.576 and HZUME -2.16. An additional bootstrap analysis of the variance of these results showed that HEPHY and Ibrahim cannot be reliably ranked, hence the final rankings :

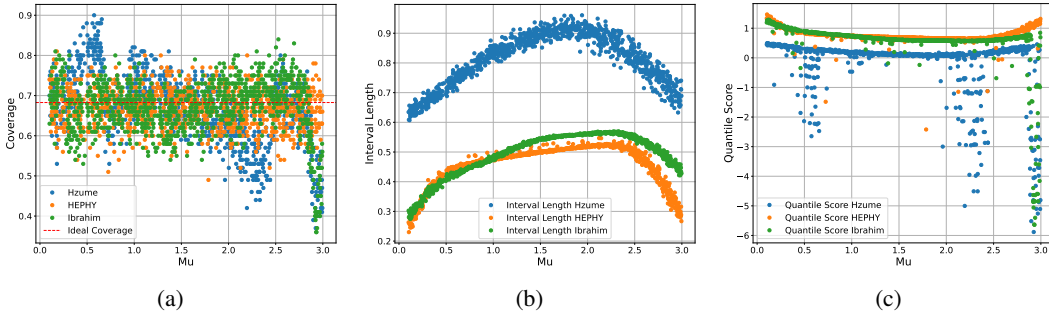


Figure 3: Comparative study of the three finalists (blue for Hzume, orange for HEPHY and green for Ibrahim's model) with 1000 trials of 100 pseudo-experiments (see subsection 3.1). 3a the coverage from each trial, 3b the average CI width and 3c the quantile score

- 1st tie: HEPHY (Lisa Benato, Cristina Giordano, Claudius Krause, Ang Li, Robert Schöfbeck, Maryam Shoostari, Dennis Schwarz, Daohan Wang) from Vienna's Institute of High Energy Physics (HEPHY) in Austria wins \$2000.
- 1st tie IBRAHIME (Ibrahim Elsharkawy) from University of Illinois at Urbana-Champaign, USA wins \$2000.
- 3rd HZUME (Hashizume Yota) from Kyoto University, Japan wins \$500

All three are co-authors of this paper and have provided a summary of their algorithms in the following sub-sections. HEPHY and Ibrahim's sub-sections also refer to their public full papers and their code.

5.1 HEPHY: Simulation-based inference with a calibrated multiclassifier and parametric regressors for learning systematics

We use simulation-based inference (SBI) to construct a flexible, unbinned, and refinable likelihood model [75] that captures the full high-dimensional event information for inference of the signal strength μ and the nuisance parameters ν via a multiclassifier and parametric regressors [76].

The code-base for “Guaranteed Optimal Likelihood-based Unbinned Method” (GOLLUM) is publicly available at Ref. [77]. Only a brief description is provided here. In the extended profiled likelihood-ratio test-statistic $q_\mu(\mathcal{D}) = -2 \log \frac{\max_{\nu} L(\mathcal{D}|\mu, \nu)}{\max_{\mu, \nu} L(\mathcal{D}|\mu, \nu)}$, we introduce a reference likelihood to the nominal (unvaried) hypothesis as $q_\mu(\mathcal{D}) = \min_{\nu} u(\mathcal{D}|\mu, \nu) - \min_{\mu, \nu} u(\mathcal{D}|\mu, \nu)$ where $-\frac{1}{2}u(\mathcal{D}|\mu, \nu) = -\mathcal{L}(\sigma(\mu, \nu) - \sigma(1, \mathbf{0})) + \sum_{i=1}^{N_{\text{obs}}} \log \left(\frac{d\sigma(\mathbf{x}_i|\mu, \nu)}{d\sigma(\mathbf{x}_i|1, \mathbf{0})} \right)$. We parametrise the inclusive yield ($\mathcal{L}(\sigma(\mu, \nu))$) (total number of expected events) and differential cross section ratio $\frac{d\sigma(\mathbf{x}|\mu, \nu)}{d\sigma(\mathbf{x}|1, \mathbf{0})}$ (density ratios) by surrogates in six disjoint selections, two of which are signal-enriched and the rest serve to constrain nuisance parameters.

A multiclass classifier is trained on nominal (i.e., unvaried) simulation data and predicts the class probabilities for the four processes: $H \rightarrow \tau\tau$, $Z \rightarrow \tau\tau$, $t\bar{t}$, and VV . The output class probability is scaled with $(1 + \alpha)^\nu$ for each of the nuisance parameters ν_{bkg} , $\nu_{t\bar{t}}$, and ν_{VV} , that control the normalization of background processes. The three accompanying constants α determine the pre-fit sizes of these uncertainties. A critical step is a dedicated and highly precise isotonic regression step to calibrate the classifier’s output.

To account for the dependence of the likelihood on the remaining systematic uncertainties, a second set of networks estimates the relative variation in the differential cross section as a function of $\nu_{\text{calib}} = \{\nu_{\text{tes}}, \nu_{\text{jes}}, \nu_{\text{met}}\}$. These nuisances control uncertainties in the calibration of the data and enter training data via the biasing script. We fit an exponential ansatz parameterised by a neural network for each of the four processes (labeled by p) and separately in each region: $\frac{d\sigma_p(\mathbf{x}|\mu, \nu)}{d\sigma_p(\mathbf{x}|1, \mathbf{0})} \simeq \hat{S}_p(\mathbf{x}|\nu_{\text{calib}}) = \exp(\nu_A \hat{\Delta}_{r,p,A}(\mathbf{x}))$, where ν_A is a multi-index that labels three linear, three quadratic, and three mixed terms of the three calibration-type nuisances. The $\hat{\Delta}_{r,p,A}(\mathbf{x})$ are functions learned by the network and specific to the selection r and the process p . Based on the cross-entropy loss, the ansatz leads to the loss function

$$L[\hat{\Delta}_A] = \sum_{\nu \in \mathcal{V}} \left[\int d\sigma(\mathbf{x}|\mathbf{0}) \text{Soft}^+(\nu_A \hat{\Delta}_A(\mathbf{x})) + \int d\sigma(\mathbf{x}|\nu) \text{Soft}^+(-\nu_A \hat{\Delta}_A(\mathbf{x})) \right]. \quad (3)$$

This architecture allows the surrogate to interpolate continuously in both feature and nuisance parameter space. The complete likelihood can then be computed from the surrogate for the differential cross-section ratio with the closed-form expression

$$\begin{aligned} \frac{d\sigma(\mathbf{x}|\mu, \nu)}{d\sigma(\mathbf{x}|1, \mathbf{0})} &\simeq \mu \hat{g}_H(\mathbf{x}) \hat{S}_H(\mathbf{x}|\nu_{\text{calib}}) + (1 + \alpha_{\text{bkg}})^{\nu_{\text{bkg}}} \left(\hat{g}_Z(\mathbf{x}) \hat{S}_Z(\mathbf{x}|\nu_{\text{calib}}) \right. \\ &\quad \left. + (1 + \alpha_{t\bar{t}})^{\nu_{t\bar{t}}} \hat{g}_{t\bar{t}}(\mathbf{x}) \hat{S}_{t\bar{t}}(\mathbf{x}|\nu_{\text{calib}}) + (1 + \alpha_{VV})^{\nu_{VV}} \hat{g}_{VV}(\mathbf{x}) \hat{S}_{VV}(\mathbf{x}|\nu_{\text{calib}}) \right) \end{aligned} \quad (4)$$

where $\hat{g}_p(\mathbf{x})$ is the (calibrated) output of the multiclassifier. The surrogate is efficient in evaluating and differentiable with respect to all parameters. For the inclusive cross-section component of the extended likelihood, we introduce a spline-based interpolation scheme that reduces numerical instabilities and speeds up the evaluation during profiling.

We train one multiclass classifier and one systematic network per selection. Closure tests show that the surrogates reproduce the shapes and normalisations of the simulated distributions across many kinematic observables and several orders of magnitude. The unbinned surrogate could be further refined in a modular way: new systematics or background processes can be added without retraining the entire model, mirroring the workflow of traditional HEP data analyses. This “refinable” modelling is crucial for scalability in real LHC analyses where hundreds of nuisance parameters are typical.

We profile the nuisance parameters using the MINUIT package [78] and determine the 68% CI by evaluating the profiled likelihood as a function of μ . The gain from the unbinned model becomes evident when performing inference, where the unbinned surrogate model improves the expected 1σ confidence interval on the signal strength by 20% compared to a traditional binned analysis using classifier-based templates. The unbinned model also leads to significantly stronger constraints on nuisance parameters, especially for calibration-related systematics like ν_{tes} and ν_{jes} , reducing their impact on μ by up to 65% when compared with the binned case [76].

We assess the performance with $5 \cdot 10^4$ toys in Figure 4. The signal strength μ is reconstructed stably over the whole range of relevant μ_{true} . We severely constrain $\nu_{t\bar{t}}$ and ν_{jes} , reducing the impact of the corresponding uncertainties. The total training time for the model was 200 CPU core hours.

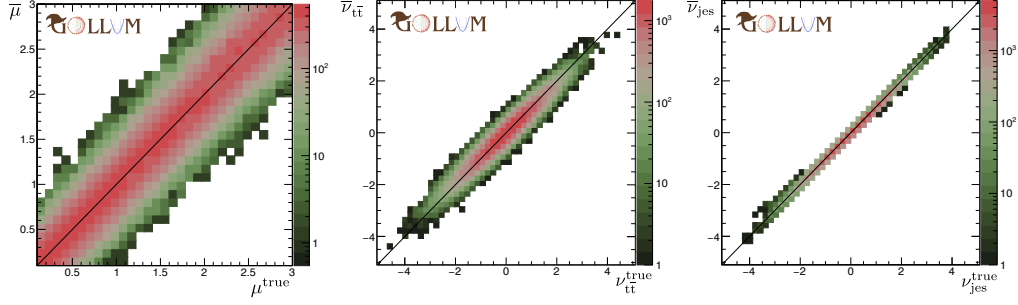


Figure 4: Scatter plot of the true value of the $H \rightarrow \tau\tau$ signal strength parameter μ (left) and the MLE $\bar{\mu}$ for $5 \cdot 10^4$ toys showing stability over the whole range of relevant μ_{true} . The normalisation-type nuisance parameter ν_{tt} (middle) and the calibration-type nuisance parameter ν_{jes} (right) are severely constrained, reducing the impact of the corresponding uncertainties.

5.2 ibrahime: Contrastive Normalizing Flows for Uncertainty-Aware Parameter Estimation

The full description of the method can be found in the method paper [79]. The code used to train and evaluate the method is available at [80].

Motivation A binary classifier can, in principle, estimate any model parameter Θ_i by learning a monotonic approximation of the likelihood ratio $r(\mathbf{x}, \{\Theta_i, \nu_i\}, \{\Theta'_i, \nu'_i\}) \propto \frac{P(\mathbf{x}|\{\Theta_i, \nu_i\})}{P(\mathbf{x}|\{\Theta'_i, \nu'_i\})}$ [5], where \mathbf{x} are the data features and ν_i are nuisance parameters. In practice, this classifier approach can be impractical; if the number of model parameters k_Θ or nuisance parameters k_ν is large, the dimensionality prevents sufficient sampling of parameter space for many choices of $\{\Theta_i, \nu_i\}$. For the challenge, $\Theta \equiv \mu \propto f_s$, where f_s is the signal fraction, and ν_i are the six HiggsML nuisance parameters. Given $\mu \propto f_s$ we can attempt to learn instead the likelihood ratio $r(\mathbf{x}, \{\nu_i\}, \{\nu'_i\}) \propto \frac{p_s(\mathbf{x}|\{\nu_i\})}{p_b(\mathbf{x}|\{\nu'_i\})}$, where p_s and p_b are the signal and background distributions, by training on class labels and then determining μ with maximum likelihood estimation. To remedy the curse of dimensionality, we then replace the raw nuisance parameters ν_i with some discrimination functions $\Phi_{s,b}[\mathbf{x}; \{\nu_i\}]$ such that $r(\mathbf{x}, \{\nu_i\}, \{\nu'_i\}) \propto \frac{p_s(\mathbf{x}|\Phi_s[\mathbf{x}; \{\nu_i\}])}{p_b(\mathbf{x}|\Phi_b[\mathbf{x}; \{\nu'_i\}])}$. If these discrimination functions are relatively insensitive to nuisance parameters and take very different values for $\mathbf{x} \sim p_s$ compared to $\mathbf{x} \sim p_b$, a classifier trained on these features will more accurately approximate the desired likelihood with less data. We argue that Contrastive Normalising Flows (CNFs) are especially suitable for these functions $\Phi_{s,b}[\mathbf{x}; \{\nu_i\}]$.

Contrastive Normalising Flows (CNFs) A CNF is a normalising flow trained with a contrastive objective that simultaneously *maximises* the likelihood of one class and *suppresses* the likelihood of the other. Starting from the standard NF loss, and training on labelled data $\mathbf{x}_s \sim p_s$ and $\mathbf{x}_b \sim p_b$, we insert a term $c \log p_\theta^{(s)}(\mathbf{x}_b)$ so that

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_s, \mathbf{x}_b \in \mathcal{D}} \left\{ -\log p_\theta^{(s)}(\mathbf{x}_s) + c \log p_\theta^{(s)}(\mathbf{x}_b) \right\} \quad (5)$$

thereby causing the learned density $p_\theta^{(s)}$ to concentrate probability mass in regions characteristic of the signal and unlike background. CNFs have been used in anomaly detection settings [81]. We generalize with c and develop a novel architecture and training procedure empirically required for accurate learning [79]. Exchanging the roles of \mathbf{x}_s and \mathbf{x}_b gives a loss function \mathcal{L}_b and a learned function $p_\theta^{(b)}$ that concentrates in background regions. Transforming these probabilities as $\Phi_{s,b}(\mathbf{x}) = p_\theta^{(s,b)}(\mathbf{x}) / [1 + p_\theta^{(s,b)}(\mathbf{x})]$ gives us our monotonic discrimination functions that retain the full shape of each class. Because the model learns a class distribution, not just a decision boundary, its scores are more stable under systematic shifts than those of a purely discriminative network. Tuning c lets us trade off coverage versus stability under systematic shifts seen in Figure 5.

The Method The total training time is 10 GPU hours.

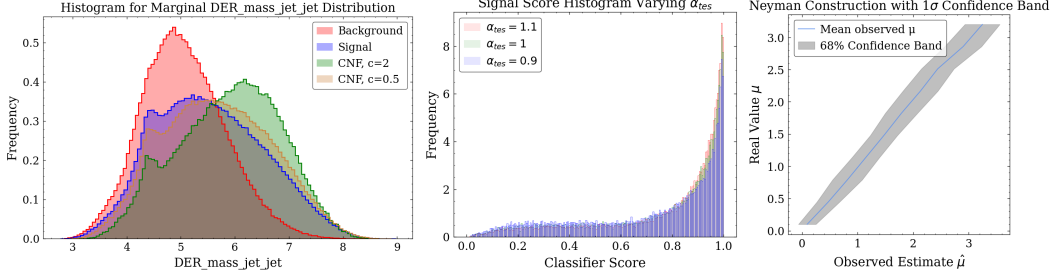


Figure 5: CNF distributions for various c (left), DNN score histograms for signal varying the nuisance parameter α_{tes} (center panel), the Neyman confidence belt (right)

Step 1. Pre-processing. Events are split into 1-jet and 2-jet categories (empirically, 0-jet events hurt performance). We take the log of features which peak near zero and then standardise all features.

Step 2. CNF density learning. For each jet category we fit two CNFs ($p_{\theta,c}^{(s)}, p_{\theta,c}^{(b)}$) for $c \in \{0.5, 2.0\}$. $c > 1$ sharpens signal-rich regions and is empirically shift-robust, while $c < 1$ preserves coverage.

Step 3. DNN Classifier For any event \mathbf{x} we compute $\Phi^{(s,b)}(\mathbf{x}) = \frac{p_{\theta,c}^{(s,b)}(\mathbf{x})}{1 + p_{\theta,c}^{(s,b)}(\mathbf{x})}$ for $c \in \{0.5, 2.0\}$ yielding four CNF scores per jet category. Together with the primary and derived features, these are fed to a two-headed DNN (shared trunk, jet-specific heads) whose binary-cross-entropy loss is minimised on just 1,000 shifted mixtures uniformly sampling each ν_i . We highlight the efficacy of CNF features with the relative invariance of the score histogram in Figure 5.

Step 4. Maximum likelihood estimation and the Neyman Construction. After training, the classifier scores are histogrammed for a given test set, and maximum likelihood estimation is performed to find point estimates for μ , α_{jes} , and α_{tes} given spline-interpolated signal and background template histograms. The point estimate for μ , $\hat{\mu}$, is used to build a Neyman confidence belt, where for each value of real μ we estimate $\hat{\mu}$ and compute the 68% spread as can be seen in Figure 5. This confidence belt can then be inverted at evaluation time to find the 1σ error bars on μ given a $\hat{\mu}$ value.

5.3 hzume: Decision-Tree Aggregated Features and Hybrid Bin-Classifer/Quantile-Regressor

We build a two-stage model composed of an Aggregation stage and an Estimation stage. Total training time is one CPU core hour.

Aggregation Stage: Feature Engineering : (i) For each event (x_{ij}) a decision tree estimates the class label y_{ij} (signal vs. background), yielding a probability p_{ij} . From the set $\{p_{ij}\}$ we compute and aggregate mean, variance, skewness, kurtosis, and the empirical quantiles at levels 0–255. (ii) For each feature x_{ij} its mean and variance across events, is fed into a second decision tree that predicts the Nuisance Parameters (e.g. TES, JES). These predictions are appended as additional features.

Estimation Stage: Two Models & Merging Strategy (i) A decision tree classifier partitions the interval $[0.1, 3]$ into five equal-width bins and predicts the bin containing μ . The resulting probability is converted into the narrowest CI covering 68 % of the total probability. (ii) A quantile-regression model directly predicts the lower and upper quantiles, providing an alternative CI for μ .

Model Selection Rule. Empirically, the quantile regressor loses accuracy when μ is near the end-points (0.1 or 3). Therefore, we adopt the bin classifier in the edge regions and the quantile regressor in the central region to produce the final CI.

6 Conclusions and Outlook

We have prepared a dataset [65] (with relevant software [74]), challenge, and platform for developing and comparing machine learning methods that quantify uncertainties in addition to providing point estimates. With the growing size of datasets in high-energy physics, the sophistication of tools, and the precision requirements to explore new phenomena, uncertainty quantification will be an

essential part of machine learning in the future. The two winning approaches, [subsection 5.1 \[76\]](#) and [subsection 5.2 \[79\]](#), show two alternative techniques on how the treatment of systematic uncertainties can be incorporated successfully in experimental analyses.

The two techniques have very similar performances, however their results are not very correlated which implies the optimum has not been reached yet. Beyond this specific metric, we expect that this unique large dataset equipped with a biasing script will be the basis of future studies, for example: (i) the precise parametrisation of density and density ratios over several order of magnitudes which is fundamental to precision physics (ii) development of morphing/Optimal Transport techniques to parameterise multidimensional non-parametric biases (iii) the same studies but with a focus on learning with a limited number of instances.

Acknowledgements

We are grateful to the US Department of Energy, Office of High Energy Physics, and the subprogram on Computational High Energy Physics, for sponsoring this research, as well as to the ANR Chair of Artificial Intelligence HUMANIA (ANR-19-CHIA-0022). Seminal discussions contributing to this work took place at the workshop “Artificial Intelligence and the Uncertainty Challenge in Fundamental Physics,” sponsored by the CNRS AISSAI Center and the DATAIA Institute, and hosted at Institut Pascal at Université Paris-Saclay. The DATAIA Institute and Institut Pascal are respectively funded by the “Investissements d’Avenir” programs ANR-17-CONV-003 and ANR-11-IDEX-0003-01. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award HEP-ERCAP0032917. The computational results of [subsection 5.1 \[76\]](#) were obtained using the CLIP cluster.

References

- [1] L. Evans and P. Bryant, *LHC machine*, *JINST* **3** (aug, 2008) S08001.
- [2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, *The Higgs boson machine learning challenge*, in *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, eds. PMLR, Montreal, Canada, 13 Dec, 2015. <http://proceedings.mlr.press/v42/cowa14.html>.
- [3] “Higgs boson machine learning challenge.” <https://www.kaggle.com/c/higgs-boson>, 2014.
- [4] HEP ML Community, “A Living Review of Machine Learning for Particle Physics.” <https://iml-wg.github.io/HEPML-LivingReview/>.
- [5] K. Cranmer, J. Pavez, and G. Louppe, *Approximating Likelihood Ratios with Calibrated Discriminative Classifiers*, *arXiv:1506.02169 [stat.AP]*.
- [6] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, *Parameterized neural networks for high-energy physics*, *Eur. Phys. J. C* **76** (2016) 235, *arXiv:1601.07913 [hep-ex]*.
- [7] J. Brehmer, F. Kling, I. Espejo, and K. Cranmer, *MadMiner: Machine learning-based inference for particle physics*, *Comput. Softw. Big Sci.* **4** (2020) 3, *arXiv:1907.10621 [hep-ph]*.
- [8] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, *Mining gold from implicit models to improve likelihood-free inference*, *Proc. Nat. Acad. Sci.* (2020) 201915980, *arXiv:1805.12244 [stat.ML]*.
- [9] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, *Constraining Effective Field Theories with Machine Learning*, *arXiv:1805.00013 [hep-ph]*.
- [10] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, *A Guide to Constraining Effective Field Theories with Machine Learning*, *arXiv:1805.00020 [hep-ph]*.
- [11] B. Nachman, *A guide for deploying Deep Learning in LHC searches: How to achieve optimality and account for uncertainty*, *arXiv:1909.03081 [hep-ph]*.

- [12] A. Ghosh, B. Nachman, and D. Whiteson, *Uncertainty-aware machine learning for high energy physics*, *Phys. Rev. D* **104** (2021) 056026, [arXiv:2105.08742 \[physics.data-an\]](#).
- [13] F. Rozet and G. Louppe, *Arbitrary Marginal Neural Ratio Estimation for Simulation-based Inference*, in *_*, 10, 2021. [arXiv:2110.00449 \[cs.LG\]](#).
- [14] ATLAS Collaboration, *An implementation of neural simulation-based inference for parameter estimation in ATLAS*, [arXiv:2412.01600 \[hep-ex\]](#).
- [15] A. Blance, M. Spannowsky, and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, *JHEP* **10** (2019) 047, [arXiv:1905.10384 \[hep-ph\]](#).
- [16] C. Englert, P. Galler, P. Harris, and M. Spannowsky, *Machine Learning Uncertainties with Adversarial Neural Networks*, *Eur. Phys. J. C* **79** (2019) 4, [arXiv:1807.08763 \[hep-ph\]](#).
- [17] G. Louppe, M. Kagan, and K. Cranmer, *Learning to Pivot with Adversarial Networks*, [arXiv:1611.01046 \[stat.ME\]](#).
- [18] J. Dolen, P. Harris, S. Marzani, S. Rappoccio, and N. Tran, *Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure*, *JHEP* **05** (2016) 156, [arXiv:1603.00027 \[hep-ph\]](#).
- [19] I. Mout, B. Nachman, and D. Neill, *Convolved Substructure: Analytically Decorrelating Jet Substructure Observables*, *JHEP* **05** (2018) 002, [arXiv:1710.06859 \[hep-ph\]](#).
- [20] J. Stevens and M. Williams, *uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers*, *JINST* **8** (2013) P12013, [arXiv:1305.7248 \[nucl-ex\]](#).
- [21] C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, and A. Sogaard, *Decorrelated Jet Substructure Tagging using Adversarial Neural Networks*, [arXiv:1703.03507 \[hep-ex\]](#).
- [22] L. Bradshaw, R. K. Mishra, A. Mitridate, and B. Ostdiek, *Mass Agnostic Jet Taggers*, [arXiv:1908.08959 \[hep-ph\]](#).
- [23] ATLAS Collaboration, *Performance of mass-decorrelated jet substructure observables for hadronic two-body decay tagging in ATLAS*, ATL-PHYS-PUB-2018-014 (2018) . <http://cds.cern.ch/record/2630973>.
- [24] G. Kasieczka and D. Shih, *DisCo Fever: Robust Networks Through Distance Correlation*, [arXiv:2001.05310 \[hep-ph\]](#).
- [25] S. Wunsch, S. Jörger, R. Wolf, and G. Quast, *Reducing the dependence of the neural network function to systematic uncertainties in the input space*, *Comput. Softw. Big Sci.* **4** (2020) 5, [arXiv:1907.11674 \[physics.data-an\]](#).
- [26] A. Rogozhnikov, A. Bukva, V. V. Gligorov, A. Ustyuzhanin, and M. Williams, *New approaches for boosting to uniformity*, *JINST* **10** (2015) T03002, [arXiv:1410.4140 \[hep-ex\]](#).
- [27] CMS Collaboration, *A deep neural network to search for new long-lived particles decaying to jets*, *Machine Learning: Science and Technology* (2020) , 1912.12238.
- [28] J. M. Clavijo, P. Glaysheer, and J. M. Katzy, *Adversarial domain adaptation to reduce sample bias of a high energy physics classifier*, [arXiv:2005.00568 \[stat.ML\]](#).
- [29] G. Kasieczka, B. Nachman, M. D. Schwartz, and D. Shih, *ABCDiCo: Automating the ABCD Method with Machine Learning*, [arXiv:2007.14400 \[hep-ph\]](#).
- [30] O. Kitouni, B. Nachman, C. Weisser, and M. Williams, *Enhancing searches for resonances with machine learning and moment decomposition*, [arXiv:2010.09745 \[hep-ph\]](#).
- [31] V. Estrade, C. Germain, I. Guyon, and D. Rousseau, *Systematic aware learning - A case study in High Energy Physics*, *EPJ Web Conf.* **214** (2019) 06024.

- [32] A. Ghosh and B. Nachman, *A cautionary tale of decorrelating theory uncertainties*, *Eur. Phys. J. C* **82** (2022) 46, [arXiv:2109.08159 \[hep-ph\]](#).
- [33] S. Wunsch, S. Jörger, R. Wolf, and G. Quast, *Optimal statistical inference in the presence of systematic uncertainties using neural network optimization based on binned Poisson likelihoods with nuisance parameters*, *Comput. Softw. Big Sci.* **5** (2021) 4, [arXiv:2003.07186 \[physics.data-an\]](#).
- [34] CMS Collaboration, *Development of systematic uncertainty-aware neural network trainings for binned-likelihood analyses at the LHC*, [arXiv:2502.13047 \[hep-ex\]](#).
- [35] L. Heinrich, *Learning Optimal Test Statistics in the Presence of Nuisance Parameters*, [arXiv:2203.13079 \[stat.ME\]](#).
- [36] A. Elwood, D. Krücker, and M. Shchedrolosiev, *Direct optimization of the discovery significance in machine learning for new physics searches in particle colliders*, *J. Phys. Conf. Ser.* **1525** (2020) 012110.
- [37] L.-G. Xia, *QBDT, a new boosting decision tree method with systematical uncertainties into training for High Energy Physics*, *Nucl. Instrum. Meth. A* **930** (2019) 15, [arXiv:1810.08387 \[physics.data-an\]](#).
- [38] P. De Castro and T. Dorigo, *INFERNO: Inference-Aware Neural Optimisation*, *Comput. Phys. Commun.* **244** (2019) 170, [arXiv:1806.04743 \[stat.ML\]](#).
- [39] T. Charnock, G. Lavaux, and B. D. Wandelt, *Automatic physical inference with information maximizing neural networks*, *Physical Review D* **97** (Apr, 2018) .
<http://dx.doi.org/10.1103/PhysRevD.97.083004>.
- [40] J. Alsing and B. Wandelt, *Nuisance hardened data compression for fast likelihood-free inference*, *Mon. Not. Roy. Astron. Soc.* **488** (2019) 5093, [arXiv:1903.01473 \[astro-ph.CO\]](#).
- [41] N. Simpson and L. Heinrich, *neos: End-to-End-Optimised Summary Statistics for High Energy Physics*, *J. Phys. Conf. Ser.* **2438** (2023) 012105, [arXiv:2203.05570 \[physics.data-an\]](#).
- [42] P. Feichtinger et al., *Punzi-loss: a non-differentiable metric approximation for sensitivity optimisation in the search for new particles*, *Eur. Phys. J. C* **82** (2022) 121, [arXiv:2110.00810 \[hep-ex\]](#).
- [43] L. Layer, T. Dorigo, and G. Strong, *Application of Inferno to a Top Pair Cross Section Measurement with CMS Open Data*, [arXiv:2301.10358 \[hep-ex\]](#).
- [44] G. Kasieczka, M. Luchmann, F. Otterpohl, and T. Plehn, *Per-Object Systematics using Deep-Learned Calibration*, [arXiv:2003.11099 \[hep-ph\]](#).
- [45] S. Bollweg, M. Haußmann, G. Kasieczka, M. Luchmann, T. Plehn, and J. Thompson, *Deep-Learning Jets with Uncertainties and More*, *SciPost Phys.* **8** (2020) 006, [arXiv:1904.10004 \[hep-ph\]](#).
- [46] J. Y. Araz and M. Spannowsky, *Combine and Conquer: Event Reconstruction with Bayesian Ensemble Neural Networks*, *JHEP* **04** (2021) 296, [arXiv:2102.01078 \[hep-ph\]](#).
- [47] M. Bellagente, M. Haußmann, M. Luchmann, and T. Plehn, *Understanding Event-Generation Networks via Uncertainties*, [arXiv:2104.04543 \[hep-ph\]](#).
- [48] T. Dorigo and P. De Castro Manzano, *Dealing with Nuisance Parameters using Machine Learning in High Energy Physics: a Review*, [arXiv:2007.09121 \[stat.ML\]](#).
- [49] T. Y. Chen, B. Dey, A. Ghosh, M. Kagan, B. Nord, and N. Ramachandra, *Interpretable Uncertainty Quantification in AI for HEP*, in *Snowmass 2021*. 8, 2022. [arXiv:2208.03284 \[hep-ex\]](#).

- [50] S. Amrouche, L. Basara, P. Calafiura, V. Estrade, S. Farrell, D. R. Ferreira, L. Finnie, N. Finnie, C. Germain, V. V. Gligorov, T. Golling, S. Gorbunov, H. Gray, I. Guyon, M. Hushchyn, V. Innocente, M. Kiehn, E. Moyse, J.-F. Puget, Y. Reina, D. Rousseau, A. Salzburger, A. Ustyuzhanin, J.-R. Vlimant, J. S. Wind, T. Xylouris, and Y. Yilmaz, *The Tracking Machine Learning Challenge: Accuracy Phase*, in The NeurIPS 2018 Competition, pp. 231–264. Springer International Publishing, Nov., 2019. [arXiv:1904.06778 \[hep-ex\]](https://arxiv.org/abs/1904.06778).
- [51] S. Amrouche, L. Basara, P. Calafiura, D. Emeliyanov, V. Estrade, S. Farrell, C. Germain, V. V. Gligorov, T. Golling, S. Gorbunov, H. Gray, I. Guyon, M. Hushchyn, V. Innocente, M. Kiehn, M. Kunze, E. Moyse, D. Rousseau, A. Salzburger, A. Ustyuzhanin, and J.-R. Vlimant, *The Tracking Machine Learning Challenge: Throughput Phase*, *Comput. Softw. Big Sci.* **7** (2023) 1, [arXiv:2105.01160 \[cs.LG\]](https://arxiv.org/abs/2105.01160).
- [52] G. Kasieczka, B. Nachman, D. Shih, O. Amram, A. Andreassen, K. Benkendorfer, B. Bortolato, G. Brooijmans, F. Canelli, J. H. Collins, B. Dai, F. F. De Freitas, B. M. Dillon, I.-M. Dinu, Z. Dong, J. Donini, J. Duarte, D. A. Faroughy, J. Gonski, P. Harris, A. Kahn, J. F. Kamenik, C. K. Khosa, P. Komiske, L. Le Potier, P. Martín-Ramiro, A. Matevc, E. Metodiev, V. Mikuni, C. W. Murphy, I. Ochoa, S. E. Park, M. Pierini, D. Rankin, V. Sanz, N. Sarda, U. Seljak, A. Smolkovic, G. Stein, C. M. Suarez, M. Szwec, J. Thaler, S. Tsan, S.-M. Udrescu, L. Vaslin, J.-R. Vlimant, D. Williams, and M. Yunus, *The LHC Olympics 2020: a community challenge for anomaly detection in high energy physics*, *Reports on Progress in Physics* **84** (Dec., 2021) 124201. [http://dx.doi.org/10.1088/1361-6633/ac36b9](https://doi.org/10.1088/1361-6633/ac36b9).
- [53] I. Guyon, L. Sun-Hosoya, M. Boullé, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W.-W. Tu, and E. Viegas, *Analysis of the AutoML Challenge Series 2015–2018*, pp. 177–219. Springer International Publishing, Cham, 2019. https://doi.org/10.1007/978-3-030-05318-5_10.
- [54] Z. Liu, A. Pavao, Z. Xu, S. Escalera, F. Ferreira, I. Guyon, S. Hong, F. Hutter, R. Ji, J. C. S. J. Junior, G. Li, M. Lindauer, Z. Luo, M. Madadi, T. Nierhoff, K. Niu, C. Pan, D. Stoll, S. Treguer, J. Wang, P. Wang, C. Wu, Y. Xiong, A. Zela, and Y. Zhang, *Winning solutions and post-challenge analyses of the ChaLearn AutoDL challenge 2019*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020) 17.
- [55] A. E. Baz, I. Ullah, and etal, *Lessons learned from the NeurIPS 2021 MetaDL challenge: Backbone fine-tuning without episodic meta-learning dominates for few-shot learning image classification*, PMLR (2022, to appear) .
- [56] D. Carrión-Ojeda, H. Chen, A. E. Baz, S. Escalera, C. Guan, I. Guyon, I. Ullah, X. Wang, and W. Zhu, *NeurIPS’22 Cross-Domain MetaDL competition: Design and baseline results*, [arXiv:2208.14686 \[cs.LG\]](https://arxiv.org/abs/2208.14686).
- [57] I. Guyon, G. Dror, V. Lemaire, D. L. Silver, G. Taylor, and D. W. Aha, *Analysis of the IJCNN 2011 UTL challenge*, *Neural Networks* **32** (2012) 174.
- [58] M. L. Danula Hettiachchi, *Crowd Bias Challenge*, 2021. <https://kaggle.com/competitions/crowd-bias-challenge>.
- [59] A. Malinin, A. Athanasopoulos, M. Barakovic, M. B. Cuadra, M. J. F. Gales, C. Granziera, M. Graziani, N. Kartashev, K. Kyriakopoulos, P.-J. Lu, N. Molchanova, A. Nikitakis, V. Raina, F. L. Rosa, E. Sivena, V. Tsarsitalidis, E. Tsompoulou, and E. Volf, “Shifts 2.0: Extending the dataset of real distributional shifts.” <https://arxiv.org/abs/2206.15407>, 2022.
- [60] S. P. Federica Proietto, Giovanni Bellitto, *Ccai@unict 2023*, 2023. <https://kaggle.com/competitions/ccaiunict-2023>.
- [61] “Nersc: Perlmutter.” <https://www.nersc.gov/systems/perlmutter/>, 2022.
- [62] Z. Xu, S. Escalera, A. Pavão, M. Richard, W.-W. Tu, Q. Yao, H. Zhao, and I. Guyon, *Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform*, *Patterns* **3** (2022) 100543. <https://www.sciencedirect.com/science/article/pii/S2666389922001465>.

- [63] H. Carlens, *State of machine learning competitions in 2024*, ML Contests Research (2025) .
<https://mlcontests.com/state-of-machine-learning-competitions-2024>.
- [64] A. Pavao, I. Guyon, A.-C. Letournel, D.-T. Tran, X. Baro, H. J. Escalante, S. Escalera, T. Thomas, and Z. Xu, *Codalab competitions: An open source platform to organize scientific challenges*, Journal of Machine Learning Research **24** (2023) 1.
<http://jmlr.org/papers/v24/21-1436.html>.
- [65] W. Bhimji, P. Calafiura, R. Chakkappai, P.-W. Chang, Y.-T. Chou, S. Diefenbacher, J. Dudley, S. Farrell, A. Ghosh, I. Guyon, C. Harris, S.-C. Hsu, K. Elham E, B. Nachman, P. Nugent, D. Rousseau, B. Thorne, I. Ullah, and Y. Zhang, “Fair universe - higgsml uncertainty challenge public dataset.” <https://zenodo.org/doi/10.5281/zenodo.15131565>, 2025.
- [66] D. Vohra, *Apache Parquet*, pp. 325–335. Apress, Berkeley, CA, 2016.
https://doi.org/10.1007/978-1-4842-2199-0_8.
- [67] ATLAS Collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*, **JINST 3** (2008) S08003.
- [68] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, *An introduction to PYTHIA 8.2*, **Comput. Phys. Commun.** **191** (2015) 159, [arXiv:1410.3012 \[hep-ph\]](https://arxiv.org/abs/1410.3012).
- [69] DELPHES 3, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, **JHEP 02** (2014) 057, [arXiv:1307.6346 \[hep-ex\]](https://arxiv.org/abs/1307.6346).
- [70] W. Bhimji, P. Calafiura, R. Chakkappai, P.-W. Chang, Y.-T. Chou, S. Diefenbacher, J. Dudley, S. Farrell, A. Ghosh, I. Guyon, C. Harris, S.-C. Hsu, E. E. Khoda, B. Nachman, P. Nugent, D. Rousseau, B. Thorne, I. Ullah, and Y. Zhang, “Fair universe generation.”
<https://github.com/FAIR-Universe/genHEPdata>, 2025.
- [71] W. Bhimji et al., *FAIR Universe HiggsML Uncertainty Challenge Competition*,
[arXiv:2410.02867 \[hep-ph\]](https://arxiv.org/abs/2410.02867).
- [72] B. Nachman and T. Rudelius, *Evidence for conservatism in LHC SUSY searches*, **Eur. Phys. J. Plus** **127** (2012) 157, [arXiv:1209.3522 \[stat.AP\]](https://arxiv.org/abs/1209.3522).
- [73] B. Nachman and T. Rudelius, *A Meta-analysis of the 8 TeV ATLAS and CMS SUSY Searches*, **JHEP 02** (2015) 004, [arXiv:1410.2270 \[hep-ph\]](https://arxiv.org/abs/1410.2270).
- [74] W. Bhimji, P. Calafiura, R. Chakkappai, P.-W. Chang, Y.-T. Chou, S. Diefenbacher, J. Dudley, S. Farrell, A. Ghosh, I. Guyon, C. Harris, S.-C. Hsu, E. E. Khoda, B. Nachman, P. Nugent, D. Rousseau, B. Thorne, I. Ullah, and Y. Zhang, “Fair universe dataset.”
https://github.com/FAIR-Universe/FAIR_Universe_dataset, 2025.
- [75] R. Schöfbeck, *Refinable modeling for unbinned SMEFT analyses*, **Mach. Learn. Sci. Tech.** **6** (2025) 015007, [arXiv:2406.19076 \[hep-ph\]](https://arxiv.org/abs/2406.19076).
- [76] L. Benato, C. Giordano, C. Krause, A. Li, R. Schöfbeck, D. Schwarz, M. Shooshtari, and D. Wang, *Unbinned inclusive cross-section measurements with machine-learned systematic uncertainties*, [arXiv:2505.05544 \[hep-ph\]](https://arxiv.org/abs/2505.05544).
- [77] L. Benato, C. Giordano, C. Krause, A. Li, R. Schöfbeck, D. Schwarz, M. Shooshtari, and D. Wang, “GOLLUM Code repository.” <https://github.com/HephyAnalysisSW/GOLLUM>, 2025.
- [78] F. James and M. Roos, *Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations*, **Comput. Phys. Commun.** **10** (1975) 343.
- [79] I. Elsharkawy and Y. Kahn, *Contrastive Normalizing Flows for Uncertainty-Aware Parameter Estimation*, [arXiv:2505.08709 \[physics.data-an\]](https://arxiv.org/abs/2505.08709).
- [80] I. Elsharkawy, “CNF for Parameter Estimation.” Github repository, 2025.
<https://github.com/ibrahimEls/CNFParameterEstimation>.

- [81] R. Schmier, U. Köthe, and C.-N. Straehle, “Positive difference distribution for image outlier detection using normalizing flows and contrastive data.”
<https://arxiv.org/abs/2208.14024>, 2023.
- [82] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, *Learning to discover: the Higgs boson machine learning challenge - Documentation*, .
<http://opendata.cern.ch/record/329>.
- [83] ATLAS Collaboration, *Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector*, **JHEP** **04** (2015) 117, [arXiv:1501.04943](https://arxiv.org/abs/1501.04943) [hep-ex].
- [84] P. Baldi, P. Sadowski, and D. Whiteson, *Searching for Exotic Particles in High-Energy Physics with Deep Learning*, **Nature Commun.** **5** (2014) 4308, [arXiv:1402.4735](https://arxiv.org/abs/1402.4735) [hep-ph].
- [85] P. Baldi, P. Sadowski, and D. Whiteson, *Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning*, **Physical Review Letters** **114** (Mar., 2015) .
<http://dx.doi.org/10.1103/PhysRevLett.114.111801>.

A Proton collisions and detection

This appendix gives details on how the data was generated.

The LHC collides bunches of protons every 25 nanoseconds within each of its four experiments. Two colliding protons produce a small firework in which part of the kinetic energy of the protons is converted into new particles. Most resulting particles are very unstable and decay quickly into a cascade of lighter particles. The ATLAS detector measures properties of these surviving particles (the so-called *final state*): the type of the particle (electron, photon, muon, etc.), its *energy*, and the 3D *direction* of the particle. Based on these properties, the decayed parent particle's properties can be inferred, and the inference chain continues until the heaviest primary particles are reached.

An online trigger system discards most of the bunch collisions containing uninteresting events. The trigger is a three-stage cascade classifier which decreases the event rate from 40 000 000 to about 400 per second. The selected 400 events are saved on disk, producing about one billion events and three petabytes of raw data per year.

The different types of particles or pseudo-particles of interest for the challenge are electrons, muons, hadronic tau, jets, and missing transverse energy. Electrons, muons, and taus are the three leptons² from the standard model.

Electrons and muons live long enough to reach the detector, so their properties (energy and direction) can be measured directly. Conversely, Taus decay almost immediately after their creation into either an electron and two neutrinos, a muon and two neutrinos, or a bunch of hadrons (charged particles) and a neutrino. The bunch of hadrons can be identified as a pseudo-particle called the hadronic tau. Jets are pseudo particles rather than real particles; they originate from a high-energy quark or gluon and appear in the detector as a collimated energy deposit associated with charged tracks. The primary information provided for the challenge is the measured momenta (see [Appendix B](#) for a short introduction to special relativity) of all the particles of the event.

We are using the conventional 3D direct reference frame of ATLAS throughout the document (see [Figure 6](#)): the z axis points along the horizontal beam line, and the x and y axes are in the transverse plane with the y axis pointing towards the top of the detector. θ is the polar angle and ϕ is the azimuthal angle. Transverse quantities are quantities projected on the $x - y$ plane, or, equivalently, quantities for which the z component is omitted. Instead of the polar angle θ , we often use the *pseudorapidity* $\eta = -\ln \tan(\theta/2)$; $\eta = 0$ corresponds to a particle in the $x - y$ plane ($\theta = \pi/2$), $\eta = +\infty$ corresponds to a particle traveling along the z -axis ($\theta = 0$) direction and $\eta = -\infty$ to the opposite direction ($\theta = \pi$). Particles can be identified in the η range in $[-2.5, 2.5]$. For $|\eta| \in [2.5, 5]$, their momentum is still measured but they cannot be identified. Particles with $|\eta|$ beyond 5 escape detection along the beam pipe.

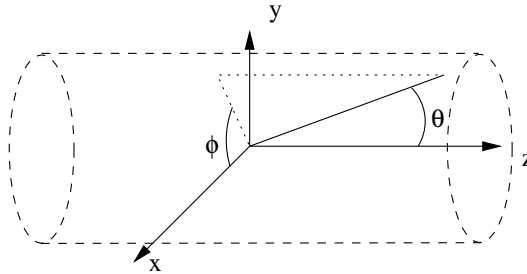


Figure 6: ATLAS reference frame

The missing transverse energy is a pseudo-particle which deserves a more detailed explanation. The neutrinos produced in the decay of a tau escape detection entirely. We can nevertheless infer their properties using the law of momentum conservation by computing the vectorial sum of the momenta of all the measured particles and subtracting it from the zero vector. In practice, measurement errors for all particles make the sum poorly estimated. Another difficulty is that many particles are lost

²For the list of elementary particles and their families, we refer the reader to <http://www.sciencemag.org/content/338/6114/1558.full>.

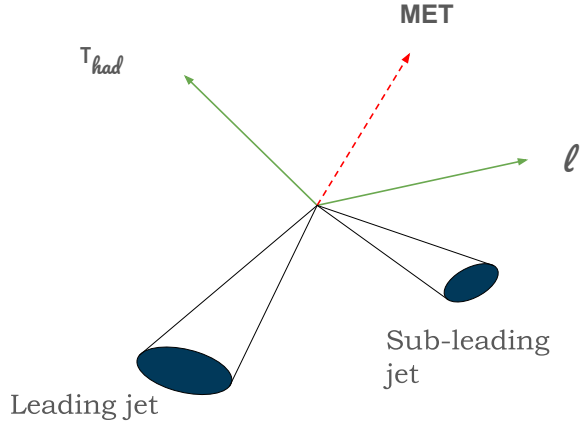


Figure 7: Diagram of the particles in the final state chosen: one lepton, one tau hadron, up to two jets, and the missing transverse momentum vector, see text for details.

Table 1: Summary of the dataset for each category and subcategory. "Number Generated" is the number of events available in the dataset. In contrast, "LHC events" is the average number in this category in a pseudo-experiment corresponding to running of the Large Hadron Collider for 10 fb^{-1} , corresponding to approximately 800 billion inelastic proton collisions, or 2 weeks in summer 2024 conditions.

Process	Number Generated	LHC Events	Label
Higgs	52 040 227	1 015	signal
Z Boson	160 383 358	1 002 395	background
Di-Boson	605 118	3 783	background
$t\bar{t}$	7 070 398	44 192	background

in the beam pipe along the z axis, so the information on momentum balance is lost in the direction of the z axis. Thus, we can carry out the summation only in the transverse plane, hence the name missing transverse energy, which is a 2D vector in the transverse plane.

For this competition, we selected only events with exactly one electron or exactly one muon, and with exactly one hadronic tau. These two particles should be of opposite electric charge. [Figure 7](#) shows the particles in the selected final state, whose parameters are provided in the data.

To summarise, for each event, we produce a list of momenta for an electron or muon, a tau hadron, up to two jets, plus the missing transverse energy.

Table 1 details the number of events of each category in the dataset.

B Special relativity

This appendix gives a very minimal introduction to special relativity for a better understanding of how the Higgs boson search is performed and what the extracted features mean (taken mainly from [82]).

B.1 Momentum, mass, and energy

A fundamental equation of special relativity defines the so-called 4-momentum of a particle,

$$E^2 = p^2 c^2 + m^2 c^4, \quad (6)$$

where E is the energy of the particle, p is its momentum, m is the rest mass and c is the speed of light. When the particle is at rest, its momentum is zero, and so Einstein's well-known equivalence between mass and energy, $E = mc^2$, applies. In particle physics, we usually use the following units: GeV for energy, GeV/ c for momentum, and GeV/ c^2 for mass. 1 GeV (10^9 electron-Volt) is one billion times the energy acquired by an electron accelerated by a field of 1 V over 1 m, and it is also approximately the energy corresponding to the mass of a proton (more precisely, the mass of the proton is about 1 GeV/ c^2). When these units are used, Equation 6 simplifies to

$$E^2 = p^2 + m^2. \quad (7)$$

To avoid the clutter of writing GeV/ c for momentum and GeV/ c^2 for mass, a shorthand of using GeV for all the three quantities of energy, momentum, and mass is usually adopted in most of the recent particle physics literature (including papers published by the ATLAS and the CMS experiments). We also adopt this convention throughout this document.

The momentum is related to the speed v of the particle. For a particle with non-zero mass, and when the speed of the particle is much smaller than the speed of light c , the momentum boils down to the classical formula $p = mv$. In special relativity, when the speed of the particle is comparable to c , we have $p = \gamma mv$, where

$$\gamma = \frac{1}{\sqrt{1 - (v/c)^2}}.$$

The relation holds both for the norms v and p and for the three dimensional vectors \vec{v} and \vec{p} , that is, $\vec{p} = \gamma m \vec{v}$, where, by convention, $p = |\vec{p}|$ and $v = |\vec{v}|$. The factor γ diverges to infinity when v is close to c , and the speed of light cannot be reached or surpassed. Hence, momentum is a concept more frequently used than speed in particle physics. The kinematics of a particle is fully defined by the momentum and energy, more precisely, by the 4-momentum (p_x, p_y, p_z, E) . When a particle is identified, it has a well-defined mass³, so its energy can be computed from the momentum and mass using Equation 6. Conversely, the mass of a particle with known momentum and energy can be obtained from

$$m = \sqrt{E^2 - p^2}. \quad (8)$$

Instead of specifying the momentum coordinate (p_x, p_y, p_z) , the parameters ϕ , η , and $p_T = \sqrt{p_x^2 + p_y^2}$, explained in Appendix A are often used.

B.2 Invariant mass

The mass of a particle is an intrinsic property of a particle. So, for all events with a Higgs boson, the Higgs boson will have the same mass. To measure the mass of the Higgs boson, we need the 4-momentum $(p_x, p_y, p_z, E) = (\vec{p}, E)$ of its decay products. Take the simple case of the Higgs boson H decaying into a final state of two particles, A and B , which are measured in the detector. By conservation of energy and momentum (which are fundamental laws of nature), we can write $E_H = E_A + E_B$ and $\vec{p}_H = \vec{p}_A + \vec{p}_B$. Since the energies and momenta of A and B are measured in the detector, we can compute E_H and $p_H = |\vec{p}_H|$ and calculate $m_H = \sqrt{E_H^2 - p_H^2}$. This is called the invariant mass because (with a perfect detector) m_H remains the same even if E_H and p_H differ from event to event. This can be generalised to more than two particles in the final state and to any number of intermediate states.

³neglecting the particle width

In our case, the final state for particles originating from the Higgs boson is a lepton, a hadronic tau, and three neutrinos. The lepton and hadronic tau are measured in the detector, but for the neutrinos, all we have is the transverse missing energy, which estimates the sum of the momenta of the three neutrinos in the transverse plane. Hence, the mass of the $\tau\tau$ can not be measured; we have to resort to different estimators which are only correlated to the mass of the $\tau\tau$. For example, the visible mass (feature DER_mass_vis) which is the invariant mass of the lepton and the hadronic tau, hence deliberately ignoring the unmeasured neutrinos. The possible jets in the events are not originating from the Higgs boson itself, but can be produced in association with it.

B.3 Other useful formulas

The following formulas are useful to compute DERived features from PRImary features (in [Appendix C](#)). For tau, lep, leading_jet, and subleading_jet, the momentum vector can be computed as

$$\vec{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} p_T \times \cos \phi \\ p_T \times \sin \phi \\ p_T \times \sinh \eta \end{pmatrix},$$

where p_T is the transverse momentum, ϕ is the azimuth angle, η is the pseudo rapidity, and \sinh is the hyperbolic sine function. The modulus of p is

$$p_T \times \cosh \eta, \quad (9)$$

where \cosh is the hyperbolic cosine function. The mass of these particles is neglected, so $E = p$.

The missing transverse energy \vec{E}_T^{miss} is a two-dimensional vector

$$\vec{E}_T^{\text{miss}} = \begin{pmatrix} |\vec{E}_T^{\text{miss}}| \times \cos \phi_T \\ |\vec{E}_T^{\text{miss}}| \times \sin \phi_T \end{pmatrix},$$

where ϕ_T is the azimuth angle of the missing transverse energy.

The invariant mass of two particles is the invariant mass of their 4-momentum sum, that is (still neglecting the mass of the two particles),

$$m_{\text{inv}}(\vec{a}, \vec{b}) = \sqrt{\left(\sqrt{a_x^2 + a_y^2 + a_z^2} + \sqrt{b_x^2 + b_y^2 + b_z^2} \right)^2 - (a_x + b_x)^2 - (a_y + b_y)^2 - (a_z + b_z)^2}. \quad (10)$$

The transverse mass of two particles is the invariant mass of the vector sum, but this time the third component is set to zero, which means only the projection on the transverse plane is considered. That is (still neglecting the mass of the two particles),

$$m_{\text{tr}}(\vec{a}, \vec{b}) = \sqrt{\left(\sqrt{a_x^2 + a_y^2} + \sqrt{b_x^2 + b_y^2} \right)^2 - (a_x + b_x)^2 - (a_y + b_y)^2}. \quad (11)$$

The pseudorapidity separation between two particles, A and B , is

$$|\eta_A - \eta_B|. \quad (12)$$

The R separation between two particles A and B is

$$\sqrt{(\eta_A - \eta_B)^2 + (\phi_A - \phi_B)^2}, \quad (13)$$

where $\phi_A - \phi_B$ is brought back to the $]-\pi, +\pi]$ range. A good intuition for the R separation is that it behaves like the 3D angle in radians between the two particles.

C The detailed description of the features

In this section, we explain the list of features that describe the events.

Prefix-less variables `Weight`, `Label`, `DetailedLabel`, have a special role and should not be used as regular features for the model⁴:

`Weight` The event weight w_i . Not to be used as a feature. Not available in the test sample.

`Label` The event label (integer) y_i 1 for signal, 0 for background . Not to be used as a feature. Not available in the test sample.

`DetailedLabel` The event detailed label (string) "htautau" for signal (when `Label==1`), "ztautau", "ttbar" and "diboson" for the three background categories (when `Label==0`). Not to be used as a feature. Not available in the test sample. This feature is used to implement some systematic biases; see [Appendix D](#). It could be used to train a multi-category classifier.

The variables prefixed with PRI (for PRImitives) are “raw” quantities about the bunch collision as measured by the detector, essentially parameters of the momenta of particles (see [Figure 8](#), [Figure 9](#) and [Figure 10](#) for their distributions).

In addition:

- Features are float unless specified otherwise.
- All azimuthal ϕ angles are in radian in the $]-\pi, +\pi]$ range.
- Energy, mass, and momentum are all in GeV
- All other features are unitless.
- Features are indicated as “undefined” when it can happen that they are meaningless or cannot be computed; in this case, their value is -25 , which is outside the normal range of all variables.
- The mass of particles has not been provided, as it can safely be neglected for the challenge.

`PRI_had_pt` The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the hadronic tau.

`PRI_had_eta` The pseudorapidity η of the hadronic tau.

`PRI_had_phi` The azimuth angle ϕ of the hadronic tau.

`PRI_lep_pt` The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the lepton (electron or muon).

`PRI_lep_eta` The pseudorapidity η of the lepton.

`PRI_lep_phi` The azimuth angle ϕ of the lepton.

`PRI_met` The missing transverse energy E_T^{miss} .

`PRI_met_phi` The azimuth angle ϕ of the missing transverse energy vector.

`PRI_jet_num` The number of jets.

`PRI_jet_leading_pt` The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the leading jet, that is the jet with the largest transverse momentum (undefined if `PRI_jet_num` = 0).

`PRI_jet_leading_eta` The pseudorapidity η of the leading jet (undefined if `PRI_jet_num` = 0).

`PRI_jet_leading_phi` The azimuth angle ϕ of the leading jet (undefined if `PRI_jet_num` = 0).

`PRI_jet_subleading_pt` The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the sub leading jet, that is, the jet with the second largest transverse momentum (undefined if `PRI_jet_num` \leq 1).

`PRI_jet_subleading_eta` The pseudorapidity η of the subleading jet (undefined if `PRI_jet_num` \leq 1).

⁴In the starting kit, they are split away in separate numpy arrays while the regular features are stored in a Dataframe

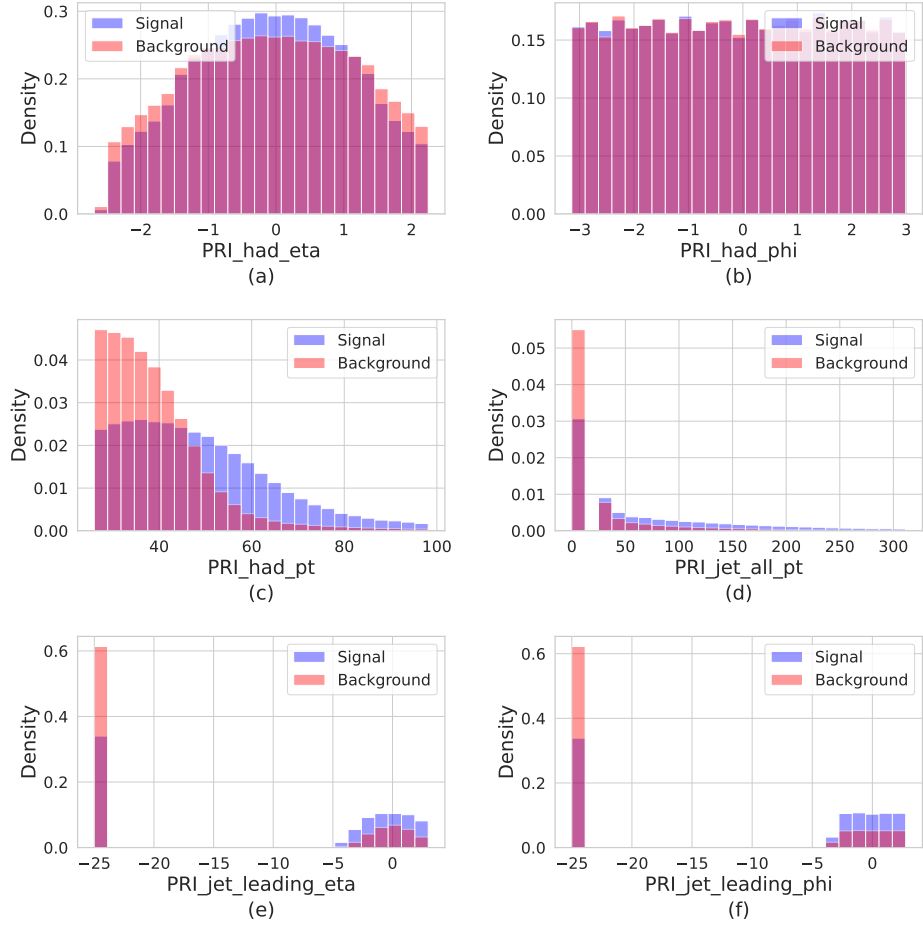


Figure 8: Distributions of: (a) hadron η , (b) hadron ϕ , (c) hadron p_T , (d) all jets p_T , (e) leading jet η , and (f) leading jet ϕ . For jet quantities, the left most bin is the default value in the absence of jets.

PRI_jet_subleading_phi The azimuth angle ϕ of the subleading jet (undefined if $\text{PRI_jet_num} \leq 1$).

PRI_jet_all_pt The scalar sum of the transverse momentum of all the jets of the events (not limited to the first 2).

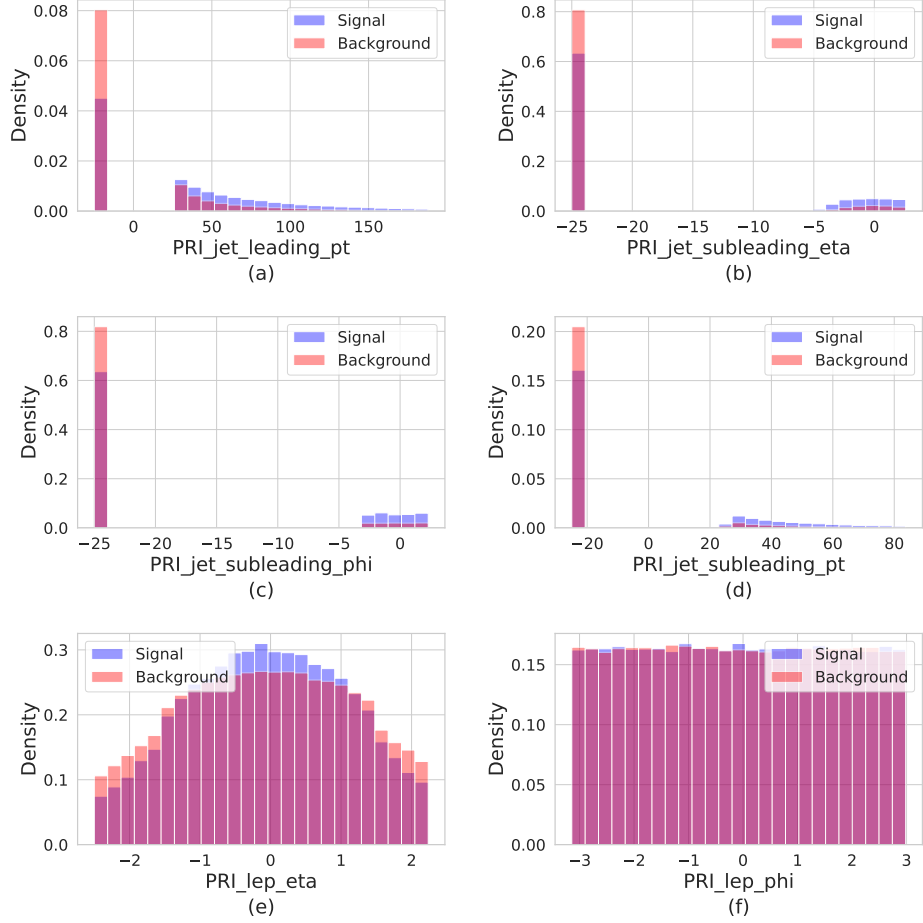


Figure 9: Distributions of: (a) leading jet p_T , (b) subleading jet η , (c) subleading jet ϕ , (d) subleading jet p_T , (e) lepton η , and (f) lepton ϕ . For jet quantities, the left most bin is the default value in no jet, or only one jet.

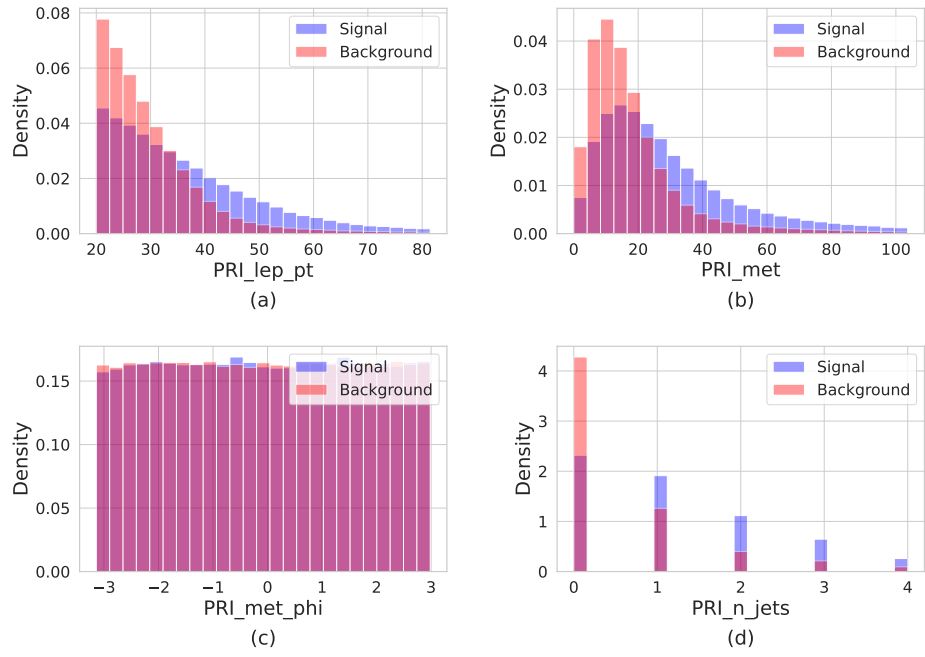


Figure 10: Distributions of: (a) lepton p_T , (b) MET, (c) MET ϕ , and (d) number of jets.

Variables prefixed with DER (for DERived) are quantities computed from the primitive features on the fly from PRImary features (including possible systematics shifts)⁵(see Figure 11 and Figure 12 for their distributions). These quantities were selected by the physicists of ATLAS in the reference document [83] either to select regions of interest or as features for the Boosted Decision Trees used in this analysis in order to enhance signal Higgs boson events separation from background events. DERived features were already present in the HiggsML dataset [82]⁶. The DERived features correspond to feature engineering; an ideal model to be trained on infinite statistics should not need these features. This distinction between primary and derived features (or "low-level" and "high-level" or "raw variables" and "human-assisted variables") is rather standard in the AI for HEP literature, see for example [84, 85]. There is no guarantee that all DERived features are useful for this challenge (they could even be detrimental in the context of systematics). The challenge participant is free to keep these DERived features, remove them altogether, keep a few, or do more feature engineering.

DER_mass_transverse_met_lep The transverse mass (Equation 11) between the missing transverse energy and the lepton.

DER_mass_vis The invariant mass (Equation 10) of the hadronic tau and the lepton.

DER_pt_h The modulus (Equation 9) of the vector sum of the transverse momentum of the hadronic tau, the lepton, and the missing transverse energy vector.

DER_deltaeta_jet_jet The absolute value of the pseudorapidity separation (Equation 12) between the two jets (undefined if PRI_jet_num ≤ 1).

DER_mass_jet_jet The invariant mass (Equation 10) of the two jets (undefined if PRI_jet_num ≤ 1).

DER_prodelta_jet_jet The product of the pseudorapidities of the two jets (undefined if PRI_jet_num ≤ 1).

DER_deltar_had_lep The R separation (Equation 13) between the hadronic tau and the lepton.

DER_pt_tot The modulus (Equation 9) of the vector sum of the missing transverse momenta and the transverse momenta of the hadronic tau, the lepton, the leading jet (if PRI_jet_num ≥ 1) and the subleading jet (if PRI_jet_num = 2) (but not of any additional jets).

DER_sum_pt The sum of the moduli (Equation 9) of the transverse momenta of the hadronic tau, the lepton, the leading jet (if PRI_jet_num ≥ 1) and the subleading jet (if PRI_jet_num = 2) and the other jets (if PRI_jet_num ≥ 3).

DER_pt_ratio_lep_tau The ratio of the transverse momenta of the lepton and the hadronic tau.

DER_met_phi_centrality The centrality of the azimuthal angle of the missing transverse energy vector w.r.t. the hadronic tau and the lepton

$$C = \frac{A + B}{\sqrt{A^2 + B^2}},$$

where $A = \sin(\phi_{\text{met}} - \phi_{\text{lep}}) * \text{sign}(\sin(\phi_{\text{had}} - \phi_{\text{lep}}))$, $B = \sin(\phi_{\text{had}} - \phi_{\text{met}}) * \text{sign}(\sin(\phi_{\text{had}} - \phi_{\text{lep}}))$, and ϕ_{met} , ϕ_{lep} , and ϕ_{had} are the azimuthal angles of the missing transverse energy vector, the lepton, and the hadronic tau, respectively. The centrality is $\sqrt{2}$ if the missing transverse energy vector $\vec{E}_{\text{T}}^{\text{miss}}$ is on the bisector of the transverse momenta of the lepton and the hadronic tau. It decreases to 1 if $\vec{E}_{\text{T}}^{\text{miss}}$ is collinear with one of these vectors and it decreases further to $-\sqrt{2}$ when $\vec{E}_{\text{T}}^{\text{miss}}$ is exactly opposite to the bisector. The logic behind this feature is that if the neutrinos are colinear to the lepton and the hadronic tau (which is a good approximation), then the missing transverse energy vector should be between the lepton and the hadronic tau.

DER_lep_eta_centrality The centrality of the pseudorapidity of the lepton w.r.t. the two jets (undefined if PRI_jet_num ≤ 1)

$$\exp \left[\frac{-4}{(\eta_1 - \eta_2)^2} \left(\eta_{\text{lep}} - \frac{\eta_1 + \eta_2}{2} \right)^2 \right],$$

⁵The code to compute DERived features from PRImitive features can be seen at https://github.com/FAIR-Universe/FAIR-Universe_dataset/blob/main/hep_challenge/derived_quantities.py

⁶The notable exception of DER_mass_MMC which was in the HiggsML dataset but is deliberately absent from the Fair-Universe dataset because it was the result of a complex and lengthy Monte-Carlo Markov Chain integration which is not practical to rerun.

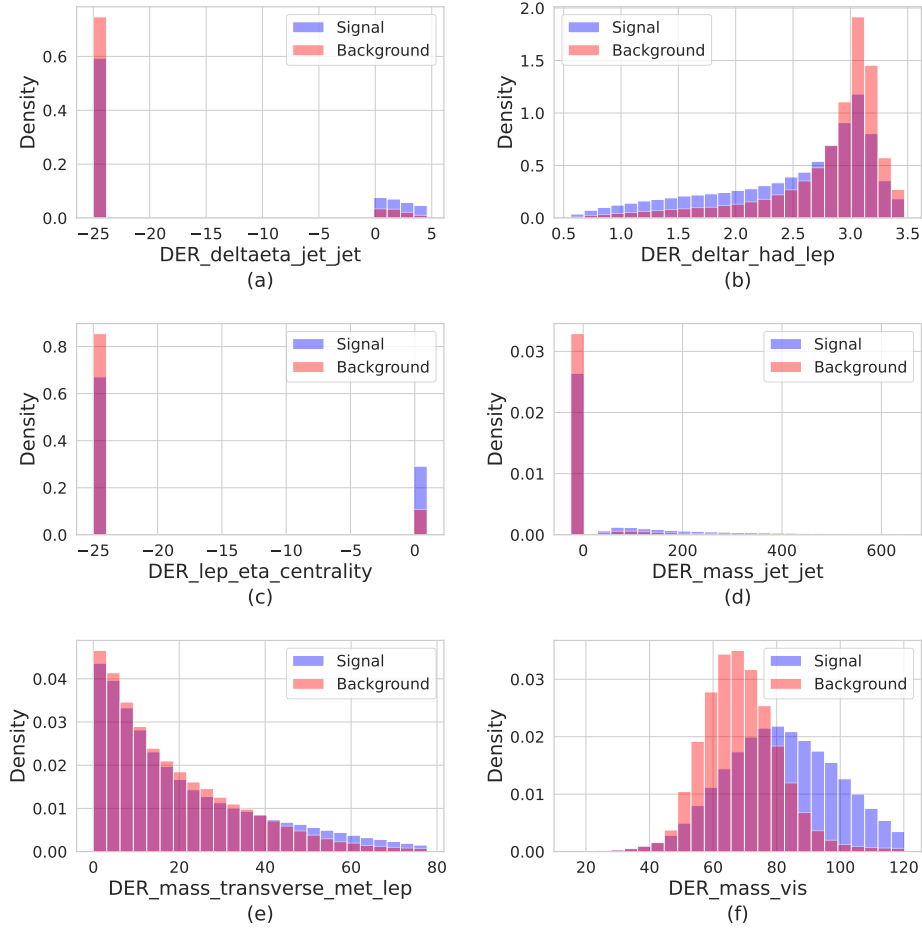


Figure 11: Distributions of kinematic variables: (a) $\Delta\eta(jet-jet)$, (b) $\Delta R(had-lep)$, (c) $lep \eta$ centrality, (d) $m(jet-jet)$, (e) $m_T(MET-lep)$, and (f) visible mass.

where η_{lep} is the pseudorapidity of the lepton and η_1 and η_2 are the pseudorapidities of the two jets. The centrality is 1 when the lepton is on the bisector of the two jets, decreases to $1/e$ when it is collinear to one of the jets, and decreases further to zero at infinity. The logic behind this feature is that if the two jets are emitted together with the Higgs boson, then the Higgs decay product should be in average between the two jets.

The feature list and event sample are primarily inspired from [83]. One crucial difference is that the dataset was produced with a more straightforward (leading-order) event generator (Pythia), and the detector effect was simulated with a more straightforward detector simulation (Delphes rather than Geant4 ATLAS Simulation). These simplifications allowed us to provide to participants a large sample allowing the development of sophisticated models while preserving the complexity of the original problem.

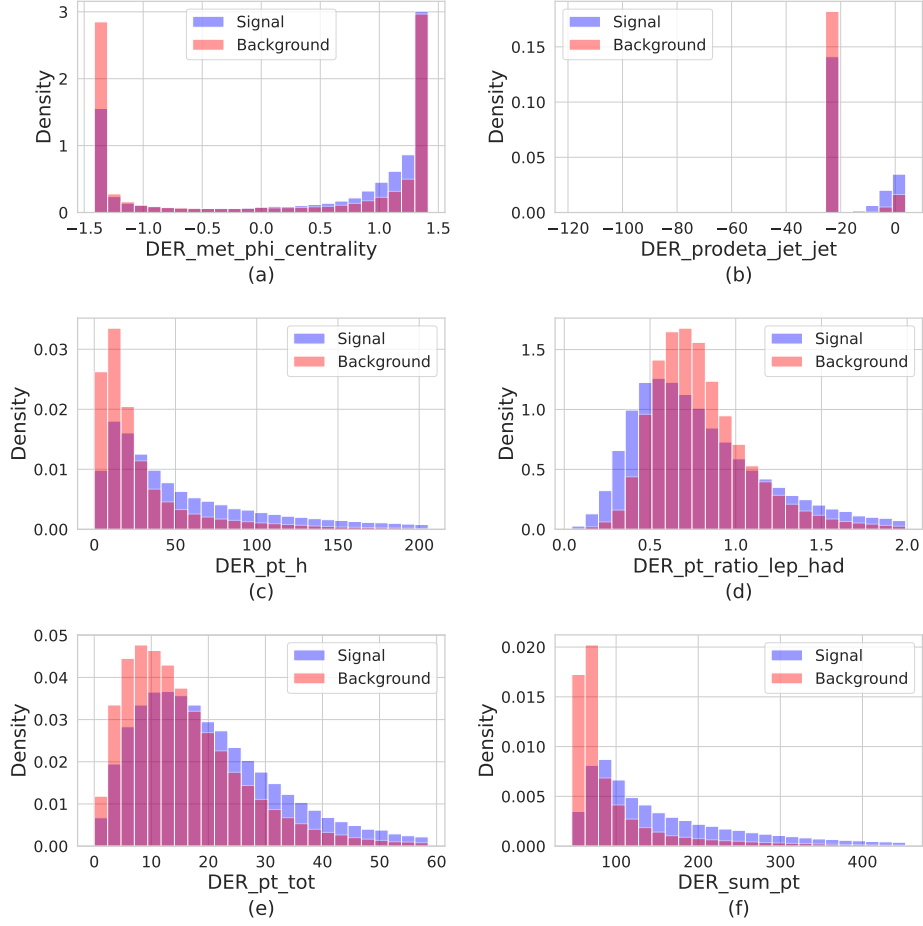


Figure 12: Distributions of: (a) MET ϕ centrality, (b) $\text{prod } \eta(\text{jet-jet})$, (c) p_T^h , (d) $p_T(\text{lep/had})$ ratio, (e) p_T^{tot} , and (f) $\sum p_T$.

Variable	Mean	Sigma	Range
α_{tes}	1.	0.01	[0.9, 1.1]
α_{jes}	1.	0.01	[0.9, 1.1]
$\alpha_{\text{soft_met}}$	0.	1.	[0., 5.]
$\alpha_{\text{ttbar_scale}}$	1.	0.02	[0.8, 1.2]
$\alpha_{\text{diboson_scale}}$	1.	0.25	[0., 2.]
$\alpha_{\text{bkg_scale}}$	1.	0.001	[0.99, +1.01]

Table 2: List of six systematic bias Nuisance Parameters defined in the challenge, with the mean and sigma of their Gaussian (Log-normal for $\alpha_{\text{soft_met}}$) distribution and their range. The corresponding α is set to the Mean value whenever a systematic bias is switched off. "No systematics" means all α are set to their Mean value.

D Systematic biases

This appendix details the implementation of the systematic biases Nuisance Parameters⁷.

D.1 Systematic bias definition

Table 2 lists the different Nuisance Parameters with their Gaussian distribution and the range to which they are clipped. α_{tes} , α_{jes} , and $\alpha_{\text{soft_met}}$ impacts some PRImary features, and then DERived features in cascade. α_{tes} and α_{jes} also impact which events make it to the final dataset. $\alpha_{\text{ttbar_scale}}$, $\alpha_{\text{diboson_scale}}$ and $\alpha_{\text{bkg_scale}}$ only impact the Weight of some background categories, that is to say, the composition of the background (for $\alpha_{\text{ttbar_scale}}$ and $\alpha_{\text{diboson_scale}}$) or the overall level of the background $\alpha_{\text{bkg_scale}}$. The Gaussian distributions parameterise our ignorance of the exact value of the biases. We think their value is 1 (or zero for $\alpha_{\text{soft_met}}$) while their real value is slightly different, as parameterised by their width, thus biasing our measurement by an unknown amount, which can be simulated.

D.2 Impact of biases on features

To detail the impact of the systematics, we need to detail first how the 4-momenta from the final state particles can be reconstructed from the PRImary features, following Appendix B. The four parameters (P_x, P_y, P_z, E) of the four-vector of each particle in the final state can be reconstructed from the PRImary features as follows (using the hadronic tau as an example, and reminding that the mass is neglected so that $E = P$),

$$P_{\text{had}} = \begin{pmatrix} \text{PRI_had_pt} * \cos(\text{PRI_had_phi}) \\ \text{PRI_had_pt} * \sin(\text{PRI_had_phi}) \\ \text{PRI_had_pt} * \sinh(\text{PRI_had_eta}) \\ \text{PRI_had_pt} * \cosh(\text{PRI_had_eta}) \end{pmatrix}$$

(where \sinh and \cosh are the hyperbolic sine and cosine functions), and similarly for P_{lep} , $P_{\text{leading jet}}$ and $P_{\text{subleading jet}}$.

The Missing ET vector is, by definition, in the transverse plane, so we have:

$$P_{\text{MET}} = \begin{pmatrix} \text{PRI_met} * \cos(\text{PRI_met_phi}) \\ \text{PRI_met} * \sin(\text{PRI_met_phi}) \\ \text{PRI_met} \end{pmatrix}$$

α_{tes} is meant to describe the fact that the detector is not calibrated correctly for the measurement of the hadron momentum, meaning when the detector reports a momentum P_{had} it really is :

$$P_{\text{had}}^{\text{biased}} = \alpha_{\text{tes}} P_{\text{had}}$$

And similarly, for the jets momentum (when they are defined)

$$P_{\text{jet_leading}}^{\text{biased}} = \alpha_{\text{jes}} P_{\text{jet_leading}}$$

⁷See also https://github.com/FAIR-Universe/FAIR_Universe_dataset/blob/main/hep_challenge/systematics.py

$$P_{\text{jet_subleading}}^{\text{biased}} = \alpha_{\text{jes}} P_{\text{jet_subleading}}$$

α_{tes} and α_{jes} also have an impact on P_{MET} : P_{MET} is obtained from the opposite of the sum of all visible objects in the event so that changing one of the visible objects (like P_{had} , $P_{\text{leading jet}}$ or $P_{\text{subleading jet}}$) has a correlated impact on P_{MET} (this calculation is performed on the first two coordinates and E_{MET} is recalculated from their modulus):

$$P_{\text{MET}}^{\text{biased}} = P_{\text{MET}} + (1 - \alpha_{\text{tes}})P_{\text{had}} + (1 - \alpha_{\text{jes}})P_{\text{leading jet}} + (1 - \alpha_{\text{jes}})P_{\text{subleading jet}}$$

$\alpha_{\text{soft_met}}$ has a different role; it expresses an additional noise source in the measurement of the missing ET vector, which is not present in the simulation. A random 2D vector of norm $ET_{\text{soft}} = \text{Lognormal}(\alpha_{\text{soft_met}})$ is added to P_{MET} (with different values event by event, by contrast with $\alpha_{\text{soft_met}}$, which has a fixed value for a given pseudo-experiment) (this calculation is performed on the first two coordinates and E_{MET} is recalculated from their modulus):

$$P_{\text{MET}}^{\text{biased}} = P_{\text{MET}} + \begin{pmatrix} \text{Gauss}(0, ET_{\text{soft}}) \\ \text{Gauss}(0, ET_{\text{soft}}) \end{pmatrix}$$

The corresponding modified PRImary features are then recomputed to new biased values: PRI_had_pt , $\text{PRI_leading_jet_pt}$, $\text{PRI_leading_jet_pt}$, PRI_met , and PRI_met_phi .

In addition,

$$\text{PRI_jet_all_pt}^{\text{biased}} = \alpha_{\text{jes}} \times \text{PRI_jet_all_pt}$$

If the number of jets is three or more, the impact of α_{jes} on missing ET cannot be calculated, given that detailed information on the additional jets (beyond two) is not available; this is a legitimate approximation as the total jet transverse momentum would be in most cases dominated by the first two leading.

DERived features are also impacted if they depend on these PRImary features (see [Appendix C](#)). Thus, for each of α_{tes} , α_{jes} and $\alpha_{\text{soft_met}}$, different features are impacted in a correlated way.

D.3 Weight impacting bias implementation

$\alpha_{\text{bkg_scale}}$, $\alpha_{\text{ttbar_scale}}$ and $\alpha_{\text{diboson_scale}}$ only impact the Weight of background events, more precisely:

- events with `DetailedLabel="ztautau"`:

$$\text{Weight}^{\text{bias}} = \alpha_{\text{bkg_scale}} \times \text{Weight}$$

- events with `DetailedLabel="ttbar"`:

$$\text{Weight}^{\text{bias}} = \alpha_{\text{bkg_scale}} \times \alpha_{\text{ttbar_scale}} \times \text{Weight}$$

- events with `DetailedLabel="diboson"`:

$$\text{Weight}^{\text{bias}} = \alpha_{\text{bkg_scale}} \times \alpha_{\text{diboson_scale}} \times \text{Weight}$$

So $\alpha_{\text{bkg_scale}}$ only affects the overall level of the background but leaves the background distributions unchanged. $\alpha_{\text{ttbar_scale}}$ and $\alpha_{\text{diboson_scale}}$ impacts only the proportion of the smaller backgrounds (see [Table 1](#)), thus distorting the overall background distribution.

D.4 Event selection

Hadronic tau (and also the jets) can only be identified in the detector above a certain transverse momentum threshold ("low threshold" in the following) so that the raw dataset PRI_had_pt , $\text{PRI_jet_leading_pt}$, $\text{PRI_jet_subleading_pt}$ have clear thresholds. When applying α_{tes} and α_{jes} , these thresholds move so that if nothing else is done, the threshold position would be an obvious giveaway of the value of α_{tes} and α_{jes} .

To alleviate this, "high thresholds" (see [Table 3](#)) have been defined, which should systematically be applied after the calculation of the biased PRImary parameters, so that the thresholds to be observed on PRI_had_pt , $\text{PRI_jet_leading_pt}$, $\text{PRI_jet_subleading_pt}$ are independent of α_{tes} and α_{jes} . The

Variable	Low threshold (GeV)	High threshold (GeV)
P_{had}^T	$\simeq 23$	26
$P_{\text{leading_jet}}^T$ and $P_{\text{subleading_jet}}^T$	$\simeq 23$	26

Table 3: Low and high threshold of hadronic tau and jet transverse momentum.

ranges in Table 2 are such that the thresholds should also be applied when no systematics bias is used⁸.

⁸In practice, function `systematics` in https://github.com/FAIR-Universe/FAIR_Universe_dataset/blob/main/hep_challenge/systematics.py should always be used, even in the no systematics case.