Computing and Software for Big Science (2025) 9:12
https://doi.org/10.1007/s41781-025-00140-9

RESEARCH

SymbolFit: Automatic Parametric Modeling with Symbolic Regression

Ho Fung Tsoi[1] · Dylan Rankin[1] · Cecile Caillol[2] · Miles Cranmer[3] · Sridhara Dasu[4] · Javier Duarte[5] · Philip Harris[6,7] · Elliot Lipeles[1] · Vladimir Loncar[6]

Received: 3 March 2025 / Accepted: 16 May 2025

Abstract
We introduce SymbolFit (API: https://github.com/hftsoi/symbolfit), a framework that automates parametric modeling by using symbolic regression to perform a machine-search for functions that fit the data while simultaneously providing uncertainty estimates in a single run. Traditionally, constructing a parametric model to accurately describe binned data has been a manual and iterative process, requiring an adequate functional form to be determined before the fit can be performed. The main challenge arises when the appropriate functional forms cannot be derived from first principles, especially when there is no underlying true closed-form function for the distribution. In this work, we develop a framework that automates and streamlines the process by utilizing symbolic regression, a machine learning technique that explores a vast space of candidate functions without requiring a predefined functional form because the functional form itself is treated as a trainable parameter, making the process far more efficient and effortless than traditional regression methods. We demonstrate the framework in high-energy physics experiments at the CERN Large Hadron Collider (LHC) using five real proton-proton collision datasets from new physics searches, including background modeling in resonance searches for high-mass dijet, trijet, paired-dijet, diphoton, and dimuon events. We show that our framework can flexibly and efficiently generate a wide range of candidate functions that fit a nontrivial distribution well using a simple fit configuration that varies only by random seed, and that the same fit configuration, which defines a vast function space, can also be applied to distributions of different shapes, whereas achieving a comparable result with traditional methods would have required extensive manual effort.

Vladimir Loncar: Also at Institute of Physics Belgrade, Serbia.

✉ Ho Fung Tsoi
ho.fung.tsoi@cern.ch

1 University of Pennsylvania, Philadelphia, PA, USA

2 European Organization for Nuclear Research (CERN), Geneva, Switzerland

3 University of Cambridge, Cambridge, UK

4 University of Wisconsin-Madison, Madison, WI, USA

5 University of California San Diego, La Jolla, CA, USA

6 Massachusetts Institute of Technology, Cambridge, MA, USA

7 Institute for Artificial Intelligence and Fundamental Interactions, Cambridge, MA, USA

Introduction

Traditional parametric modeling methods, such as polynomial regression, require specifying and fixing an adequate functional form before fitting the data. Identifying suitable functional forms for distributions with arbitrary shapes is often challenging and time-consuming, as, in most cases, these functions cannot be derived from first principles and must be determined through trial and error. Instead, symbolic regression (SR) is a more flexible and powerful technique that performs a *machine-search* for functions that best fit the data. In SR, the functional form itself is treated as a trainable parameter that is dynamically adjusted throughout the fitting process, eliminating the need to predefine an exact function—an empirical task that is often challenging. We refer the reader to Refs. [1–18] for a review of the subject and some recent works.

Genetic programming [19] is a popular approach to SR [5, 15–18]. In this approach, a function is represented as an expression tree, where the building blocks—mathematical operators, variables, and constants—are denoted as nodes, connected to represent their algebraic relations. Different functional forms are generated through the evolution of these expression trees, where tree nodes are randomly selected and changed (mutation), and subtrees from different candidates are swapped to create new candidates (crossover), as illustrated in Fig. 1. As a result, the functional forms evolve during the fitting process, guiding the model toward convergence. Instead of predefining the final functional form, SR algorithms based on genetic programming need far less prior knowledge about the functions themselves. Only the constraints for constructing the expression trees need to be specified, such as the allowable mathematical operators ($+$, $\times$, $/$, pow, $\sin(\cdot)$, $\exp(\cdot)$, etc.). This flexibility eliminates the need to know the exact fitting function beforehand or to fine-tune one empirically.

Our primary application focus is the intermediary stage of data analysis in high-energy physics (HEP) experiments at the CERN Large Hadron Collider (LHC), where parametric functions are constructed to model data distributions and subsequently used for downstream statistical inference. In these analyses, uncertainty modeling is necessary, as the uncertainties associated with the parametric functions propagate to the final physics results. Standard SR algorithms generate best-fit functions but do not inherently provide uncertainty estimates. Our framework bridges this gap by automatically re-optimizing and estimating uncertainties for all candidate functions found by SR. This functionality is critical, as parametric models without well-defined uncertainties cannot be used in the statistical inference workflows within HEP. In the following section, we identify two analysis scenarios in which parametric modeling is traditionally used. We discuss the limitations of current methods in these contexts and how SR, by eliminating the need to predefine a functional form, can offer a more flexible and efficient alternative.

We also emphasize that, unlike SR applications in other domains, where the primary goal is often interpretability, such as direct extraction of physical laws from data (e.g., [5, 10]), our objective is different. We focus on constructing valid and flexible parametric models for statistical inference, rather than deriving interpretable expressions to uncover underlying physical laws from data. Our goal is to model arbitrary distribution shapes, as commonly encountered in LHC analyses, where no single underlying physical law governs the data. In such cases, interpretability is not a relevant requirement. Instead, the key criteria for a suitable parametric function in this setting are: (1) it should smoothly and accurately describe the shape of the distribution, and (2) its associated uncertainty should be well-behaved and capable of capturing the uncertainty in the data.

Challenges in Traditional Methods

Scenario 1: Signal and Background Modeling

When analyzing proton–proton collision data at the LHC in the search for new physics signatures, the data are typically binned and presented as histograms representing physical observables, such as the invariant mass of the final-state particles. Each bin records the observed or expected number of collision events with mass values within that range. To search for new physics signatures, which are often hypothesized as narrow and small peaks over a smoothly falling background in the invariant mass distribution, parametric functions are required to model both the signal and background based on these binned distributions. These models are then used to perform hypothesis testing.

In the traditional approach to parametric modeling, one typically relies on manually guessing the appropriate family of functions that might describe the shape of the distribution. Although these distributions often represent physical observables, they are usually obtained after applying a series of selection cuts on various variables, which can introduce arbitrary shape effects into the final distributions being modeled. As a result, these distributions generally do not have a known underlying true function, making it impossible to derive a suitable functional form from first principles and leaving empirical constructions as the only option. In some cases, when a suitable function cannot be found to describe the distribution after many trials, one is forced to

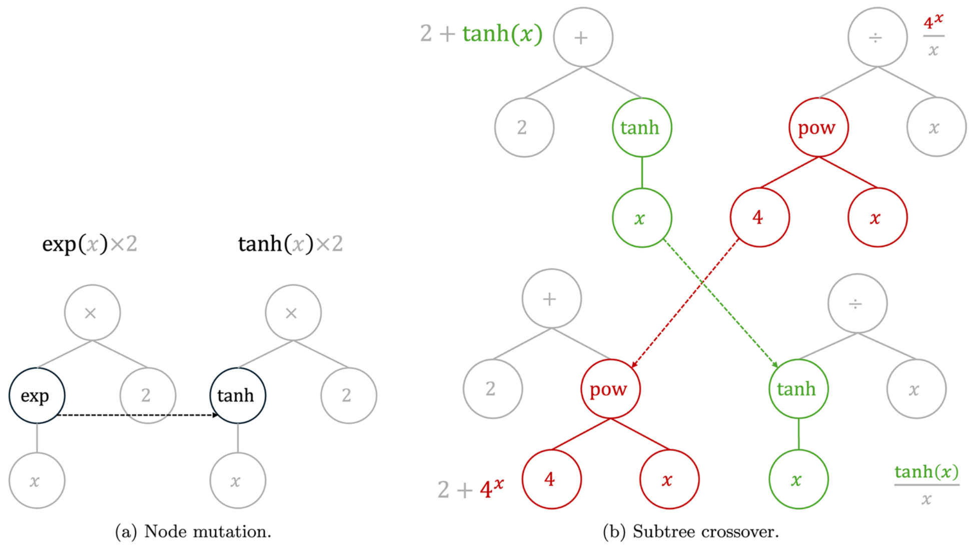


Fig. 1 Genetic programming approach to symbolic regression. Functions are represented by expression trees. New functions are generated through mutation of tree nodes (left) and crossover between subtrees (right)

compromise by adjusting the analysis strategy, say splitting the main distribution into multiple sub-distributions and fitting them separately, which could lead to a more complicated combined likelihood function. This empirical approach has been the standard strategy in the HEP community, requiring significant manual effort to craft a candidate function and iteratively fine-tune it.

For example, a search for new physics in high-mass trijet events performed by the CMS experiment [20] modeled the background by fitting the trijet invariant mass distribution, m_{jjj}, using three families of empirical functions. One of these functions takes the form:

$$f(x;N) = \frac{p_0(1-x)^{p_1}}{x^{\sum_{i=2}^{N} p_i \log^{i-2}(x)}}, \tag{1}$$

where $x = m_{\mathrm{jjj}}/\sqrt{s}$ is a dimensionless variable ($\sqrt{s}$ is the center-of-mass energy of the collisions), p_i are free parameters, and N is a hyperparameter for the function form. The function was fitted multiple times with different trial N values, and the optimal value was determined through a separate statistical test, such as an F-test [21].

Note that the functional form in Eq. 1 was constructed empirically, rather than derived from first principles, to reproduce the observed spectrum. This reflects the fact that the trijet distribution arises from events that have passed through multiple stages of selection, including triggering, reconstruction, and optimization, rather than being determined by a single underlying physics law. The same is true for other similar analyses at the LHC.

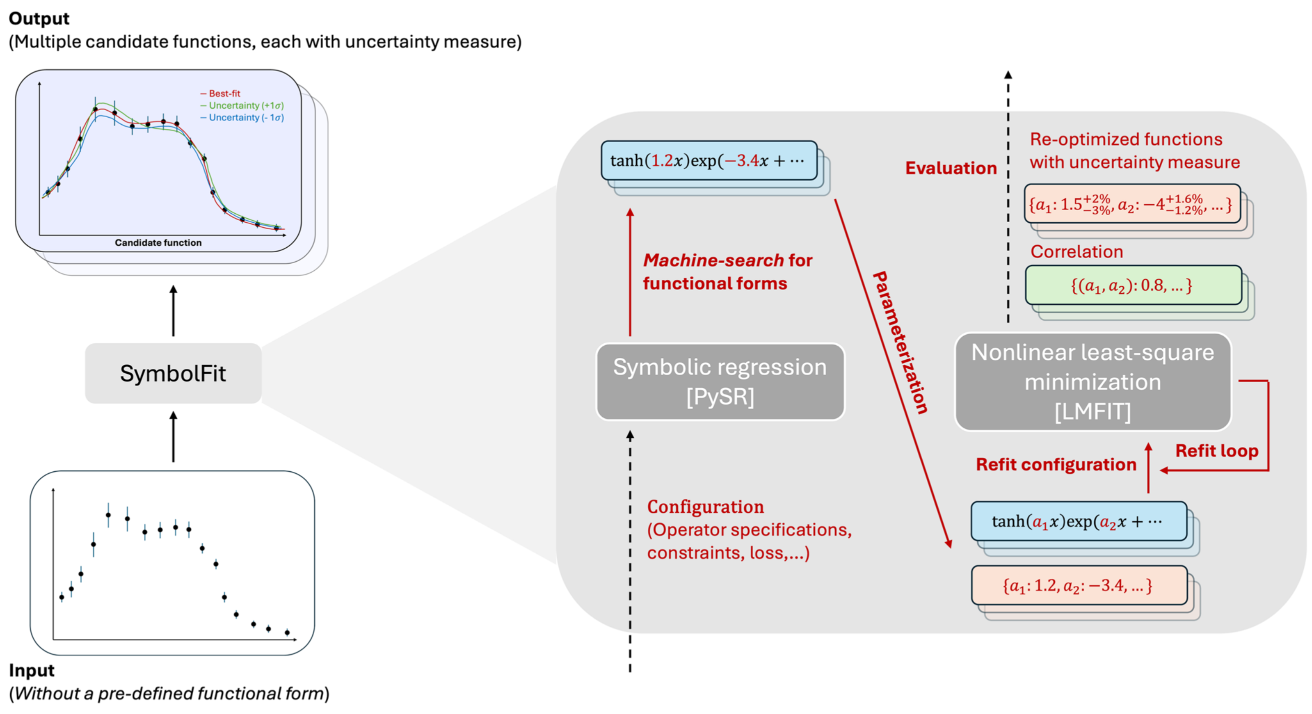
The challenge lies in the need to empirically craft a specific functional form, such as Eq. 1, for each individual distribution. These empirical functions are tailored to the particular distribution being fitted, making them rigid and potentially ineffective if there are slight changes in the data. For instance, variations in final-state objects, event selection strategies, and detector conditions during data collection can all introduce arbitrary modifications in the shape of the distribution. In such cases, function families that worked for past datasets may no longer be effective for future datasets, even within the same analysis channel, and an empirical searching for suitable functions must be repeated.

Analyses at the LHC have traditionally relied on this empirical fitting method when modeling signal and background processes from binned data. Examples include the milestone analyses that led to the discovery of the Higgs boson in 2012 [22–24], as well as some recent results from CMS searches for high-mass resonances in dijet [25], paired-dijet [26], trijet [20], diphoton [27], and dimuon [28] events.

In this context, SR has the potential to transform the approach to parametric modeling. By conducting a *machine-search* for suitable functional forms, SR significantly reduces the manual effort required in the modeling process, providing a more efficient and adaptive alternative to traditional methods.

Scenario 2: Derivation of Smooth Descriptions from Binned Data

When predicting signal and background processes using simulation, there is always some degree of mismatch with the observed data, which may result from inaccuracies in theoretical predictions, mis-modeling of detector effects, or measurements errors. These discrepancies are corrected by applying data-to-simulation scale factors (measured from isolated control regions) to the simulated events, ensuring that the simulation provides a more realistic representation



Output
(Multiple candidate functions, each with uncertainty measure)
Best-fit
Uncertainty (+1σ)
Uncertainty (- 1σ)
Candidate function
SymbolFit
Input
(Without a pre-defined functional form)
tanh(1.2x)exp(−3.4x + ⋯
Machine-search for functional forms
Symbolic regression [PySR]
Configuration (Operator specifications, constraints, loss,...)
Parameterization
Evaluation
Re-optimized functions with uncertainty measure
{a1: 1.5+2%−3%, a2: −4+1.6%−1.2%, … }
Correlation
{(a1, a2): 0.8, … }
Nonlinear least-square minimization [LMFIT]
Refit configuration
Refit loop
tanh(a1x)exp(a2x + ⋯
{a1: 1.2, a2: −3.4, … }

Fig. 2 A schematic sketch of the internal steps within the `SymbolFit` framework illustrates how it interfaces with `PySR` [5] and `LMFIT` [37] to automate parametric modeling using SR. The process begins with an input dataset that does not require a predefined functional form. Functional forms are generated using SR, parameterized, and then re-optimized through standard nonlinear least-square minimization. The output is a batch of candidate functions, each with associated uncertainty estimates

of the observed data. Examples include jet energy scale corrections parameterized by the jet p_T and η[1] [29], heavy-flavor jet tagging efficiency corrections parameterized by the jet p_T and η [30], hadronic tau identification efficiency corrections parameterized by the tau p_T, η, and decay modes [31].

These scale factors are typically derived from binned data and applied as binned weights, resulting in coarse-grained corrections. When smooth scale factors are desired, the process often follows the same empirical approach as described in Sec. "Scenario 1: signal and background modeling", facing the same limitations discussed earlier. In cases where the scale factor is parameterized by more than one variable, it becomes even more challenging to empirically construct an adequate functional form, forcing one to rely on coarse-grained corrections.

Another common scenario involves data-driven background estimation methods, where transfer factors are derived to estimate events in the signal region based on those in the sideband region. For example, in a search for a boosted Higgs boson decaying to b quarks performed by the CMS experiment [32], the QCD multijet background was estimated from observed data, where the transfer factor was parameterized and empirically constructed as the sum of products of Bernstein polynomials:

$$f(x_0,x_1)=\left(\sum_{\mu=0}^{n_{x_0}}\sum_{\nu=0}^{n_{x_1}} a_{\mu,\nu} b_{\mu,n_{x_0}}(x_0) b_{\nu,n_{x_1}}(x_1)\right)\times g(x_0,x_1), \tag{2}$$

where $b_{n,N}(x)$ is the nth Bernstein basis polynomial of degree N, $a_{\mu,\nu}$ are parameters to be extracted from a fit to observed data, and $g(x_0,x_1)$ is a function fitted separately to simulated events. The degrees of the Bernstein polynomials, n_{x_0} and n_{x_1}, are determined separately using an F-test.

By using SR, these empirical steps for deriving smooth scale factors can be significantly simplified into a single SR fit, without knowledge of the final functional form.

An Alternative Method: Gaussian Process Regression

An alternative fitting method is Gaussian process regression (GPR), which has been explored for these scenarios [33–35]. GPR models the dependent variable as following a Gaussian distribution at each point along the independent variable. The smoothness of the probability function is

[1] Common coordinate system used to define particle kinematics in collider physics: p_T is transverse momentum and η is the pseudorapidity angle.

controlled by a chosen covariance kernel between bins. As a result, GPR provides a probabilistic prediction, yielding both a smooth mean function and a variance function, which define a very generic distribution of functions that describe the data instead of a single exact function.

Fitting a GPR model to n data points requires inverting an $n \times n$ covariance matrix, which scales with a time complexity of $\mathcal{O}(n^3)$ [36]. This can become computationally prohibitive, especially for datasets with more than one independent variable. Additionally, integrating Bayesian GPR outputs into the standard HEP search framework requires subtle treatments [33], whereas SR directly provides explicit function templates that can be straightforwardly integrated into existing workflows.

Despite the potential of alternative methods like GPR, but due to the limitations described above, the empirical method remains the primary approach to parametric modeling within the HEP community. There is currently a lack of an efficient framework or a package based on an alternative method that can be readily used out-of-the-box.

Proposed Solution with Symbolic Regression

For the scenarios discussed above, we propose using SR to replace traditional methods, shifting the paradigm of parametric modeling in HEP.

In this paper, we introduce a Python API[2] for the `SymbolFit` framework, which interfaces with `PySR` [5] (a high-performance SR library) and `LMFIT` [37] (a nonlinear least-square minimization library), aimed at *automating* parametric modeling of binned data using SR. We demonstrate the effectiveness of the framework in two common HEP applications: parametric modeling of signal and background, and the derivation of smooth scale factors. These applications are validated using five real datasets from new physics searches at the CERN LHC, along with several toy datasets. The key features of the framework are summarized below:

- **Pre-determined functional forms are no longer needed.** With SR, only minimal *constraints* are required to define the function space, such as specifying the allowed mathematical operators (+, ×, /, pow, sin(·), exp(·), etc.). This does not demand extensive and prior knowledge of the final functions that describe the distribution. The search for suitable functions is automatically performed by machine, eliminating the empirical and manual process. We show that a simple SR configuration can flexibly fit a wide variety of distribution shapes.

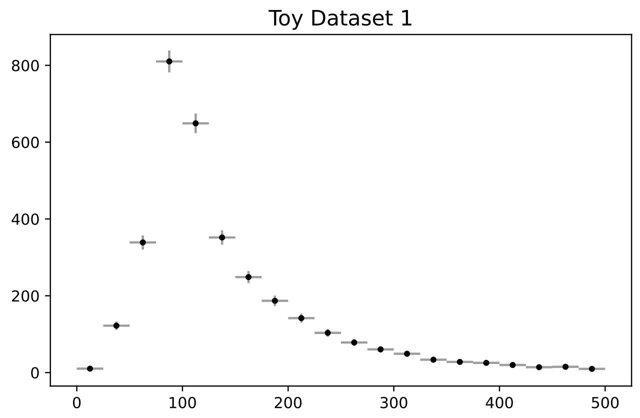
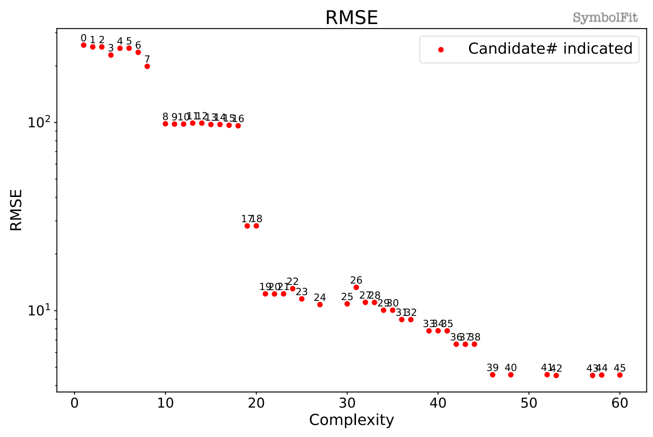


Fig. 3 Toy Dataset 1: a 1D binned dataset with uncertainties represented by vertical error bars. The data points are manually generated without reference to an underlying function

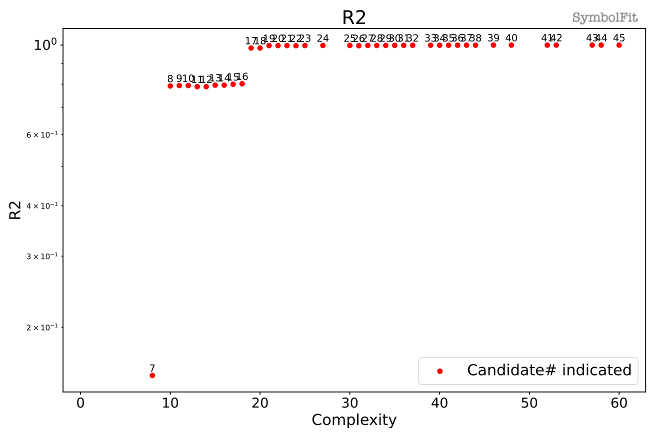
- **Generating multiple candidate functions per fit.** SR based on genetic programming generates and evolves successive generations of functions, producing a batch of candidate functions in each search iteration. The same search configuration can be repeated with different random seeds to explore different suitable functions from the vast function space. This flexibility in generating a variety of candidate functions allows for adaptability across different downstream tasks.
- **Inclusion of uncertainty measure.** While SR algorithms alone are dedicated to function searching and do not inherently provide any uncertainty estimation, our framework bridges the gap by incorporating a re-optimization process for the candidate functions. This step improves the best-fit models and generates uncertainty estimation needed to access the modeling reliability.
- **Modeling of multidimensional data.** The framework easily accommodates modeling data with multiple variables, which is particularly useful in HEP scenarios where scale factors are sometimes parameterized by more than one variable.

Moreover, the framework is designed to automate the process as much as possible, minimizing manual effort. Results are evaluated and plotted automatically, which are saved in readable formats such as CSV and PDF files, including diagnostic plots that allow users to visually assess the fit quality and data comparison. The candidate functions generated can be seamlessly integrated into downstream statistical inference tools commonly used in HEP, such as `Combine` [38] and `pyhf` [39, 40], as the output models are in identical

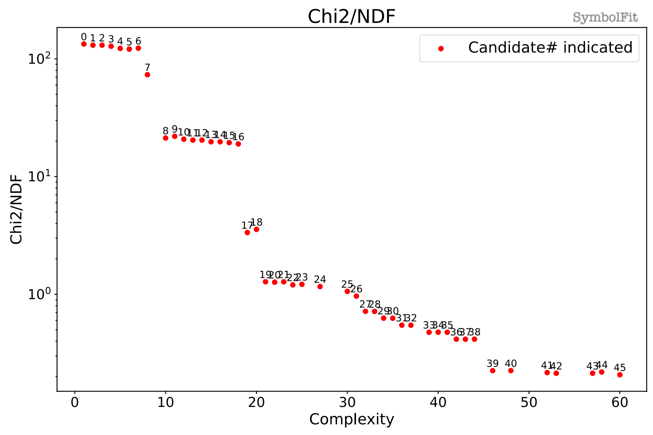
[2] https://github.com/hftsoi/symbolfit



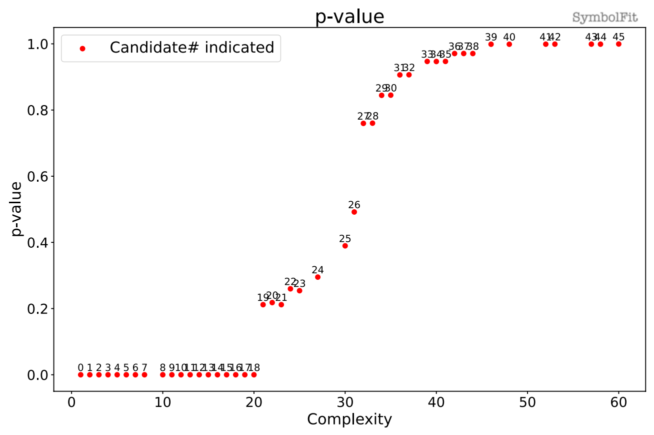
(a) Root-mean-square error (RMSE).



(b) Coefficient of determination (R2).



(c) χ^2/NDF.



(d) p-value.

Fig. 4 Goodness-of-fit scores vs. function complexity. A total of 46 candidate functions (labeled #0–#45) were obtained from a single fit on Toy Dataset 1 in Sec. "Toy Dataset 1 (1D) [signal modeling]". Candidate functions #10, #17, #27, and #38 are listed in Table 1, and their combined uncertainty coverage is presented in Fig. 7. Specifically, individual parameter variations for candidate function #38 are shown in Fig. 5, with its parameter correlation matrix shown in Fig. 6

representation to those from traditional methods—closed-form functions. The efficiency of SR in generating a wide range of well-fitted functions per fit also allows flexible modeling as the function choice can be treated as a source of systematic uncertainty through the discrete profiling method [41].

The rest of the paper is structured as follows: Sec. "Method" describes the SymbolFit framework; Sec. "Demonstrations" presents demonstrations using a real LHC dataset as well as several toy datasets; Sec. "Summary" provides a summary of the work. More demonstrations are presented in Appendix A.

Method

The SymbolFit framework is illustrated in Fig. 2 and explained in the following:

1. **Input data.** We consider the input dataset $\{(\boldsymbol{x}^i, y^i, y^i_{\text{up}}, y^i_{\text{down}})\}_{i=1}^n$, where $\boldsymbol{x}^i$ represents one or more independent variables, y^i is the dependent variable with associated uncertainties $y^i_{\text{up/down}}$ at one standard deviation, and n is the number of data points. In the context of binned histograms, which are commonly used in HEP data analysis, there are n bins. Here, $\boldsymbol{x}^i$ represents the center of the i-th bin, and y^i is the bin content, represent-

Table 1 Nine examples from the 46 candidate functions obtained from a single fit to Toy Dataset 1. These functions were fitted to a scaled dataset (to enhance fit stability and prevent numerical overflow), which can be rescaled to describe the original dataset using the transformation: $f(x) \to 165 \times f(0.00211(x - 12.5))$. The comparison between the χ^2/NDF scores before and after the ROF step is presented. The total uncertainty coverage of candidate functions #10, #17, #27, #38 is shown in Fig. 7. Individual parameter variations for candidate function #27 are plotted in Fig. 5. The function complexity values, providing a rough estimate of the model size, are computed before algebraic simplification. Numerical values are rounded to three significant figures for display purposes

Complexity	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
12 (#10)	$0.101 + 23.3\text{gauss}(-3.74x)x$	2	374.3 / 18	374.3 / 18	10^{-68}
19 (#17)	$0.061 + (5.11x + 5.11\text{gauss}(-2.34 + 12.6x))\text{gauss}(2.52x)$	4	54.06 / 16	53.59 / 16	10^{-06}
22 (#20)	$0.0837 + (4.76\text{gauss}(-15.7x + 2.84) + 10.4\tanh(x))\text{gauss}(2.99x)$	6	48.47 / 14	17.76 / 14	0.218
27 (#24)	$(4.79\text{gauss}((-5.5 + 2x)(-0.538 + 3x)) + 10.2x)\text{gauss}(-3.03x) +0.0841$	6	16.88 / 14	16.31 / 14	0.2951
31 (#26)	$(4.9x + 4.9\text{gauss}(-2.79 + 15.4x) + 4.9\tanh(x))\text{gauss}(3x)+ 0.0789\text{gauss}(x)\exp(x)$	4	15.79 / 16	15.45 / 16	0.4919
32 (#27)	$(5.13\text{gauss}(-16.7x + 3.05) + 13.1x)\text{gauss}(x(-4.68 + x) + x) +0.0661$	6	12 / 14	10.04 / 14	0.760
37 (#32)	$(5.07\text{gauss}((-4.42 + 2x)(-0.724 + 4x)) + 5.07\tanh(x) + 7.79x)\times \text{gauss}(x(-4.65 + x) + x) + 0.066$	6	8.359 / 14	7.655 / 14	0.9065
44 (#38)	$(5.08\text{gauss}((-4.7 + 4x)(-0.719 + 4x)) + 12.7x)\times \text{gauss}(x(-4.66 + x) + x) + 0.0662$	6	6.278 / 14	5.826 / 14	0.971
52 (#41)	$0.0657 + (5\text{gauss}((-4.96 + 6x)(-0.712 + 4x)) + 12.4\tanh(x))\times \text{gauss}(x(-4.6 + x) + x) - 0.00624x$	6	3.564 / 14	3.032 / 14	0.999

ing the number of events within the bin. The associated uncertainties $y^i_{\text{up/down}}$ account for measurement errors or modeling inaccuracies.

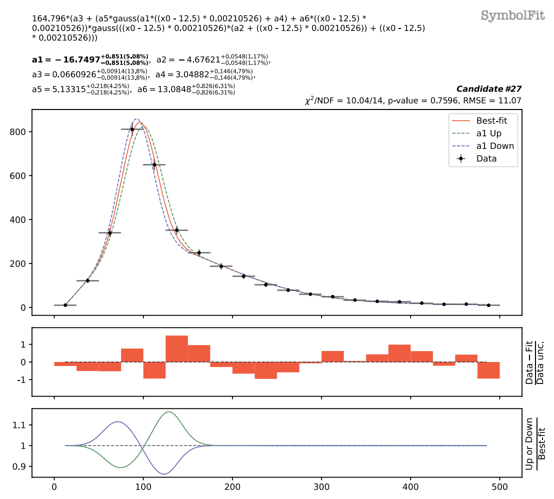
2. **Symbolic regression.** The core of the framework is to leverage SR to perform a *machine-search* for suitable functions to model the data, without predefining a functional form. We utilize PySR [5], a Python library for genetic programming-based SR, which is highly configurable in defining the function space for the search. The configuration process is highly simplified, requiring only the specification of allowed mathematical operators (+, ×, /, pow, sin(·), exp(·), etc.) and the constraints for the functional form. The objective of the search is to minimize:

$$\chi^2 \equiv \sum_{i=1}^{n} \left(\frac{f(\mathbf{x}^i) - y^i}{y^i_{\text{up}} \mathbf{1}_{f(\mathbf{x}^i)-y^i \geq 0} + y^i_{\text{down}} \mathbf{1}_{f(\mathbf{x}^i)-y^i < 0}} \right)^2, \tag{3}$$

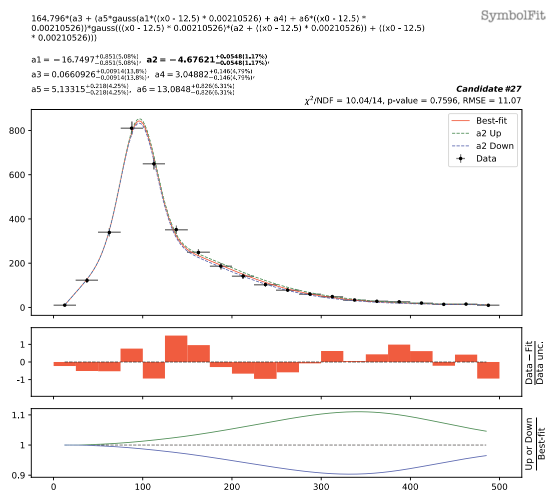
where f is the candidate function. Since PySR uses a multi-population strategy to evolve and select functions, each run generates a batch of candidate functions. These functions are then re-optimized in subsequent steps to improve the fit and provide uncertainty estimates.

3. **Parameterization.** SR algorithms search for exact functions but do not inherently provide any uncertainty measures. However, uncertainty estimation is essential in HEP data analysis to gauge the reliability of the observation and prediction. To address this, we freeze the functional forms found by SR and then re-optimize all constants in each function using standard nonlinear minimization techniques. The uncertainties in these re-optimized constants are used as the uncertainty measure for the candidate functions. First, within each candidate function, the constants are automatically identified and parameterized as $\{a_1, a_2, ...\}$, with the original values stored as initial values for the re-optimization process.

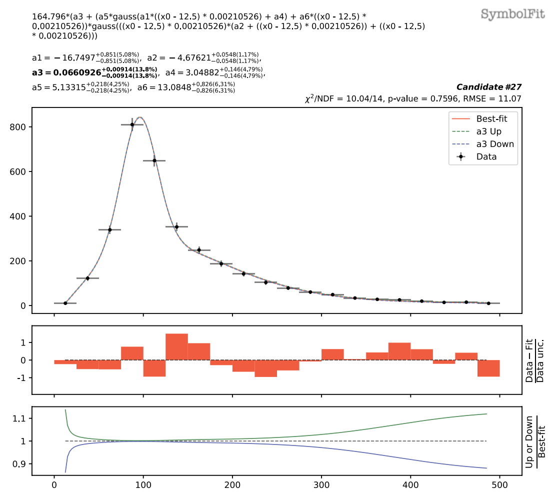
4. **Re-optimization fit (ROF).** To perform ROF of the candidate functions, we utilize LMFIT [37], a nonlinear least-square minimization library, to perform a second-fit for the parameters while keeping the functional forms fixed. The objective is to minimize χ^2 defined in Eq. 3. The parameterized functions are parsed to identify the set of parameters to be varied, and initially, all parameters are allowed to vary in the fit. In some cases, the minimization may fail to converge due to a too complex objective function. To handle these cases, a loop for ROF is implemented in the framework. This loop progressively reduces the number of degrees of freedom (NDF) by freezing more parameters to their initial values until the fit succeeds and all relative errors are below



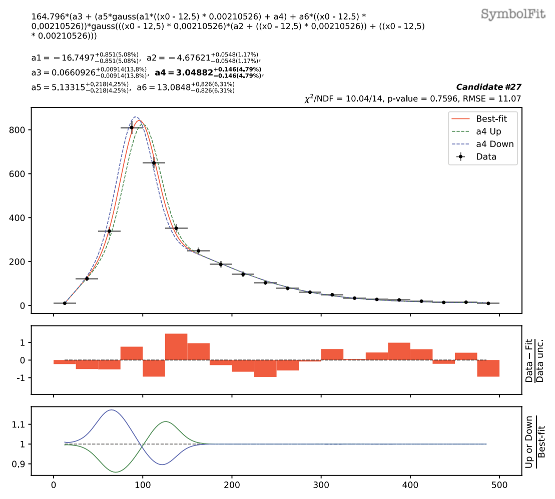
(a) ±1 std. variations of a_1.



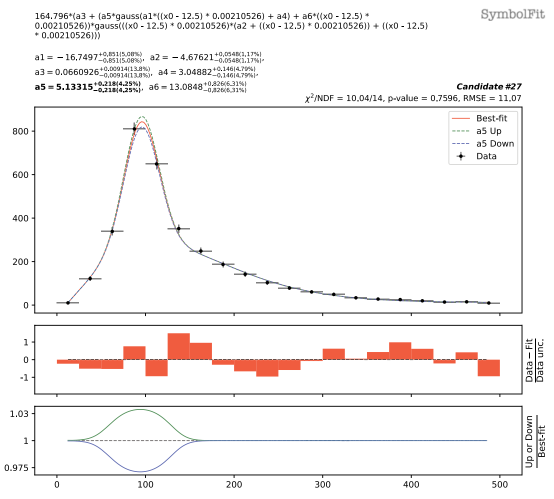
(b) ±1 std. variations of a_2.



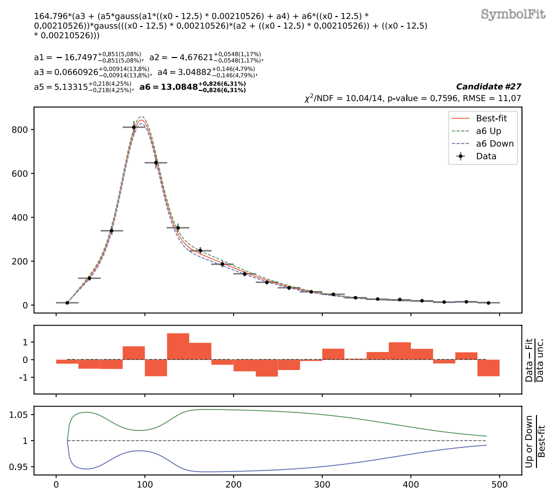
(c) ±1 std. variations of a_3.



(d) ±1 std. variations of a_4.



(e) ±1 std. variations of a_5.



(f) ±1 std. variations of a_6.

◂**Fig. 5** Individual parameter variations in candidate function #27 from a fit to Toy Dataset 1. The parameterized form of this function is shown at the top of each subfigure, along with the best-fit values of the parameters and their associated uncertainties. Each subfigure shows the same function, but with one parameter shifted by its ± 1 standard deviation (green/blue), while the other parameters remain fixed at their best-fit values. The function with all parameters held at their best-fit values is plotted in red and compared to the data, represented by black data points. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Fit}}{\text{Data unc.}}$. The bottom panel shows the ratio of the function with the uncertainty variations to the best-fit function

a predefined threshold. Finally, the candidate functions are evaluated and ranked in the outputs.

To summarize, `SymbolFit` automates all these steps in the modeling process, including the computation of various goodness-of-fit scores and the evaluation of correlations between the parameters. This integration streamlines the workflow and minimizes manual intervention while providing full information for downstream statistical analysis. The computation time of the workflow is primarily due to the search for functional forms, for which we utilize a highly optimized SR algorithm `PySR`. As a result, the process is not computationally intensive and can be flexibly configured. As the function space is usually huge even when under constrained, one can repeat the fit with the same configuration but with a different random seed to obtain a different batch of candidate functions.

Demonstrations

We demonstrate the effectiveness of our framework using five real LHC datasets from new physics searches published recently, as well as several toy datasets.

The LHC datasets consist of real proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV, collected by the CMS experiment during Run 2. These datasets cover various search channels: dijet [25] in Sec. "CMS dijet dataset (1D) [background modeling]", diphoton [27] in Appendix A.2, trijet [20] in Appendix A.3, paired-dijet [26] in Appendix A.4, and dimuon [28] in Appendix A.5. Each dataset consists of 1D binned data of the invariant mass of the respective objects, where smooth background predictions are obtained through parametric modeling and then tested for excess events indicative of new physics. In all these analyses, CMS reported no evidence of new physics is observed in the data. Therefore, for our demonstrations, we assume that each invariant mass spectrum contains no signal. We also perform experiments to validate the SR outputs for signal extraction in these LHC datasets, and details of these steps are given in Sec. "CMS dijet dataset (1D) [background modeling]".

In addition to the real LHC datasets, we also generate toy datasets for demonstration purposes. While SR has been shown to successfully identify the correct underlying function from noisy data [1, 5], here we focus on toy data generated by hand without an underlying function to illustrate SR's capability in modeling arbitrary distribution shapes. Toy Dataset 1, presented in Sec. "Toy Dataset 1 (1D) [signal modeling]", is a 1D binned distribution featuring a sharp peak and a high-end tail. Toy Dataset 2, presented in Sec. Appendix A.1, consists of three 1D binned distributions with various shapes. Toy Dataset 3, presented in Sec. "Toy dataset 3 (2D) [arbitrary shapes]", consists of three 2D binned distributions.

Toy Dataset 1 (1D) [Signal Modeling]

Figure 3 shows Toy Dataset 1, which consists of a 1D binned distribution. The distribution has a shape characterized by a sharp peak and a high-end tail, and it is generated by hand without reference to an underlying function.

In spite of the lack of a true functional form, we demonstrate that our framework using SR can replace the empirical process with minimal efforts. This can be seen from the simple `Python` snippet shown in List. 1[3], which defines the function space and configures `PySR` to perform a *machine-search* for functions to fit this dataset. In this example, the maximum function complexity is set to 60 to constrain the model size, ensuring that the total number of nodes in an expression tree does not exceed 60, with each operator equally weighted with a complexity of one. The allowed operators include two binary operators ($+$ and $\times$) and three unary operators ($\exp(\cdot)$, $\tanh(\cdot)$, and a custom-defined $\text{gauss}(\cdot) \equiv \exp(-(\cdot)^2)$). Constraints on operator nesting are imposed to prohibit scenarios like $\tanh(\tanh(\cdot))$. The loss function used is χ^2.

[3] The option definitions can be found at https://github.com/MilesCranmer/PySR and Ref. [5].

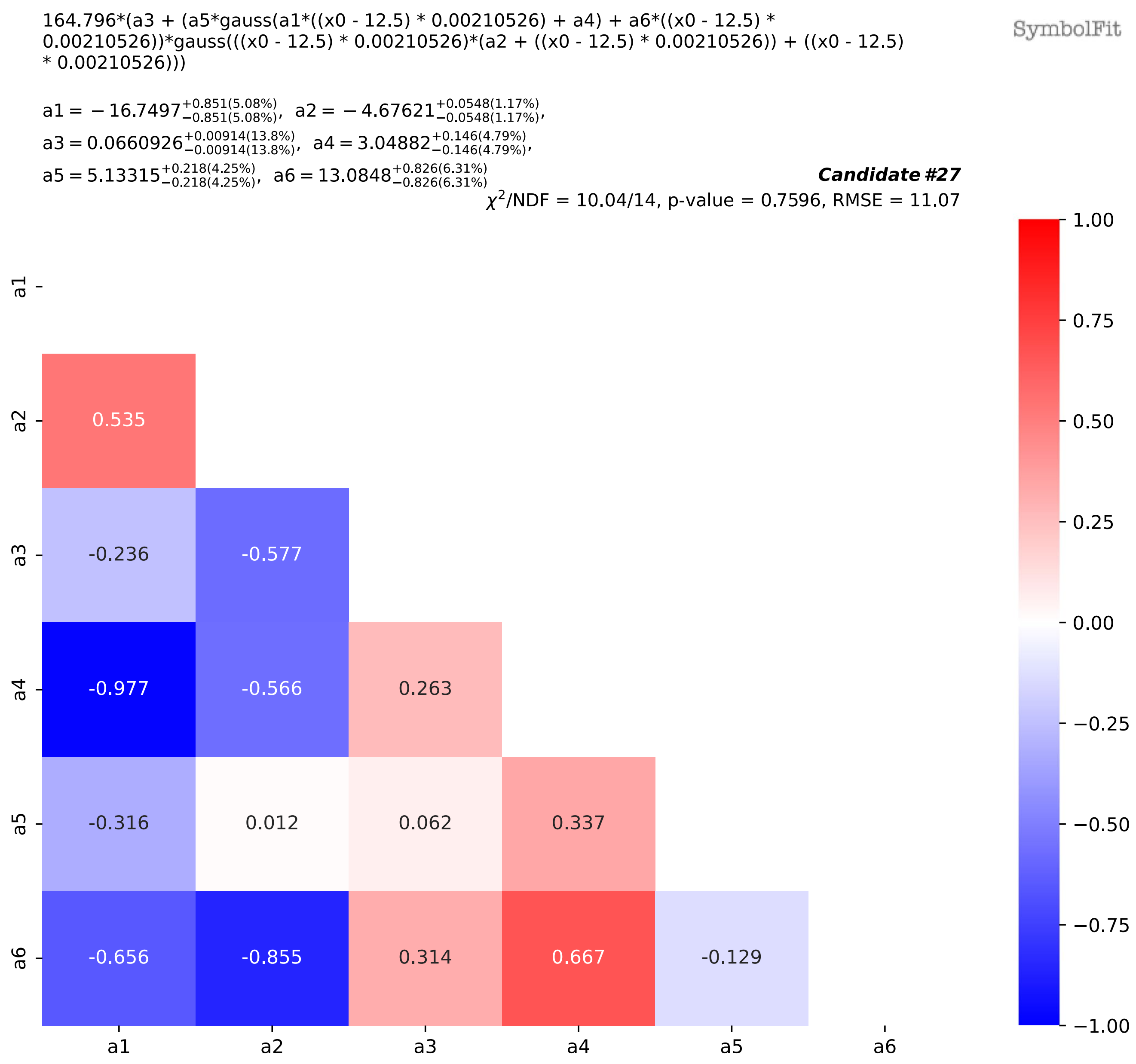
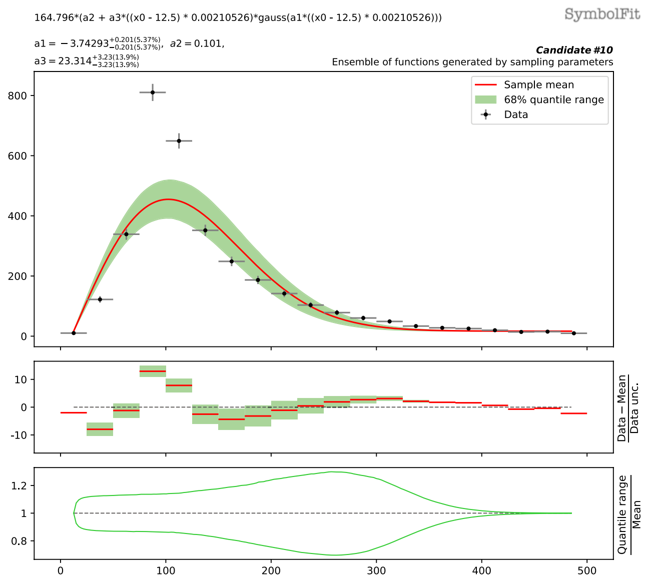
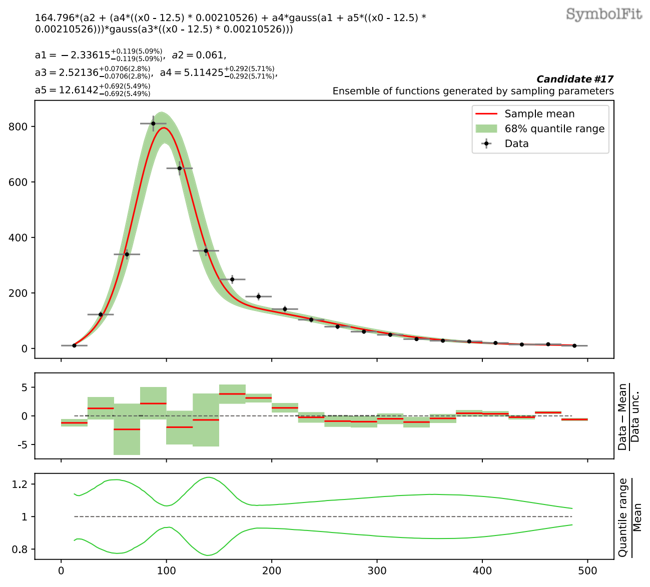


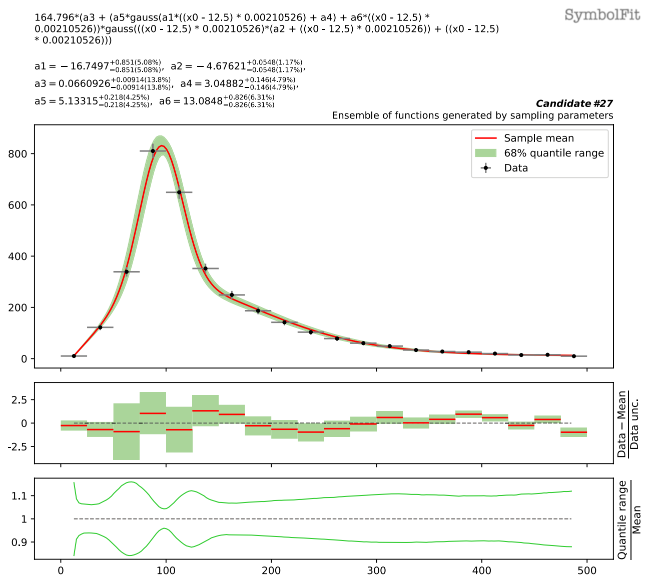
Fig. 6 Correlation matrix for the parameters of candidate function #27 from a fit to Toy Dataset 1 (see Table 1 and Fig. 5). The parameter uncertainties and their correlation define the uncertainty model associated with the function



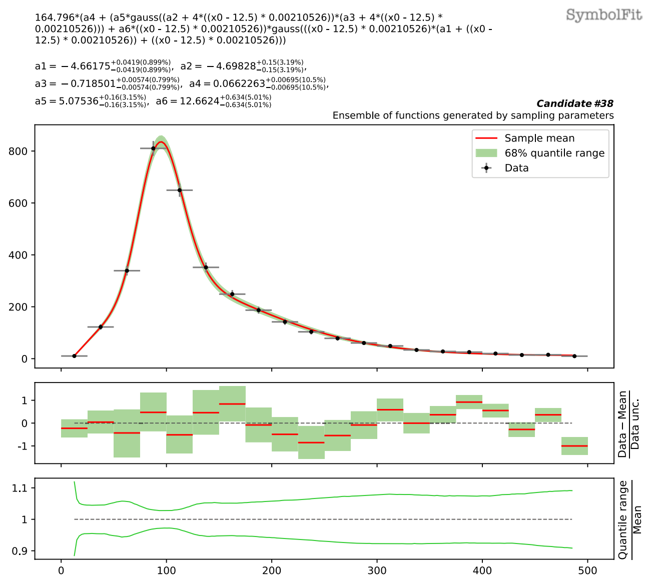
(a) Candidate function #10.



(b) Candidate function #17.



(c) Candidate function #27.



(d) Candidate function #38.

Fig. 7 Convergence of candidate functions to the data (Toy Dataset 1), from lower to higher function complexity values. To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range ($\pm 1\sigma$) of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

Table 2 The candidate functions are obtained from three separate fits using the same fit configuration but with different random seeds, fitted to the pseudodata of the dijet spectrum with the (injected) signal region blinded

	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
SR model 1	$(570x(x(-0.423\exp(2x)+\exp(x))+x)+149)\times(0.00328+0.0304\tanh(x))^{4.87x}$	5	400.5 / 30	29.21 / 30	0.507
SR model 2	$(145(0.958+x)^{\tanh(-0.711+4.32x)}+145\tanh(x))\times(0.00591+0.0522\tanh(x))^{5.48x}$	5	103.8 / 30	29.91 / 30	0.47
SR model 3	$\mathrm{pow}(149(0.0101x+0.0101\tanh(0.171+0.724x)), x+(2.38x\tanh(-0.71+x)+2.39)\tanh(x)+\tanh(x))$	5	214.8 / 30	30.93 / 30	0.419

The fits were performed on a scaled dataset (to enhance fit stability and prevent numerical overflow), and the functions can be transformed back to describe the original spectrum using the transformation:x $\rightarrow$ 0.000145(x $-$ 1568.5). These functions are plotted and compared with the blinded pseudodata in Fig. 9.Numerical values are rounded to three significant figures for display purposes

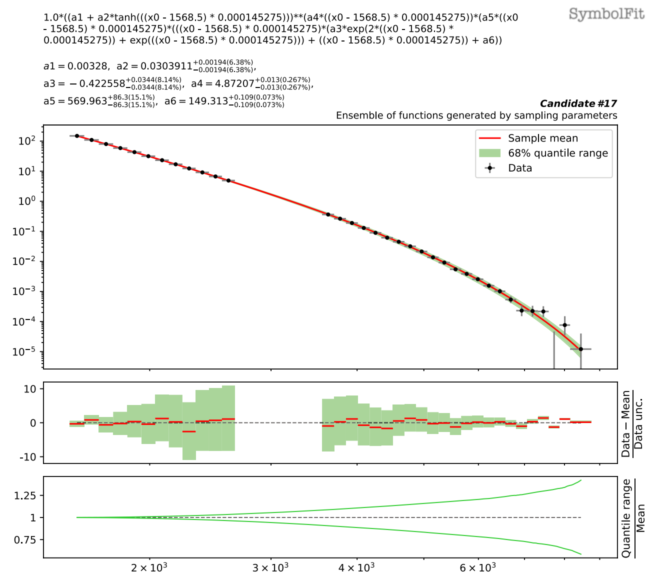
```python
from pysr import PySRRegressor
import sympy

pysr_config = PySRRegressor(
    model_selection = "accuracy",
    timeout_in_seconds = 60*100,
    niterations = 200,
    maxsize = 60,
    binary_operators = ["+", "*"],
    unary_operators = [
        "exp",
        "gauss(x) = exp(-x*x)",
        "tanh"
    ],
    nested_constraints = {
        "tanh":   {"tanh": 0, "exp": 0, "gauss": 0, "*": 2},
        "exp":    {"tanh": 0, "exp": 0, "gauss": 0, "*": 2},
        "gauss":  {"tanh": 0, "exp": 0, "gauss": 0, "*": 2},
        "*":      {"tanh": 1, "exp": 1, "gauss": 1, "*": 2}
    },
    extra_sympy_mappings={"gauss": lambda x: sympy.exp(-x*x)},
    loss = "loss(y, y_pred, weights) = (y - y_pred)^2 * weights"
)
```

Listing 1 The Python code snippet that configures PySR to search for candidate functions for Toy Datasets 1 and 3

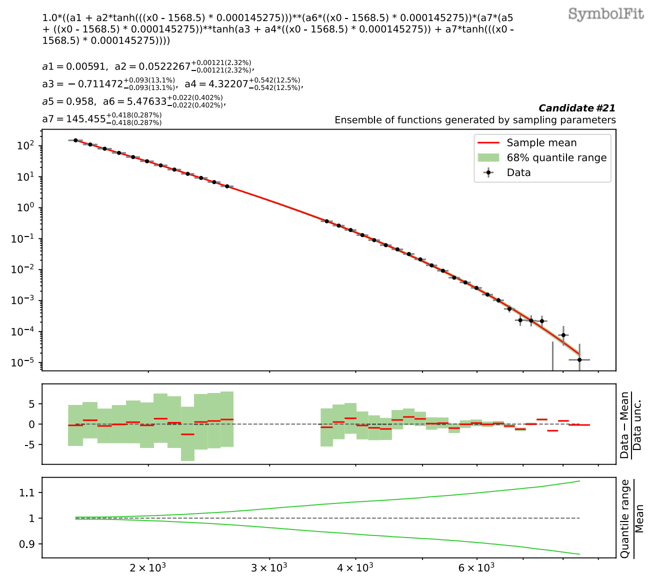
We run `SymbolFit` with the `PySR` configuration shown in List. 1. In particular, the parameter "maxsize" controls the maximum complexity of the functions, which is up to the user to choose if simpler or more complex functions are desired. This single fit generates 46 candidate functions (labeled #0 to #45), with function complexity values ranging from 1 to 60. These complexity values provide a rough estimate of the model size and are computed before any algebraic simplification. Four goodness-of-fit scores, including the χ^2/NDF and p-value, are plotted against function complexity in Fig. 4. As expected, more complex functions tend to offer better fits to the data, as their higher complexity makes them more expressive. The variety of functions produced provides flexibility, allowing one to choose a candidate function based on the desired fit quality for downstream tasks. For example, nine of the 46 candidate functions are listed in Table 1, showing χ^2/NDF values above 1, around 1, and below 1, offering a range of fit qualities.

In general, the χ^2/NDF score improves significantly after ROF compared to the original function found from SR. This is because SR algorithms are typically focused on finding optimal functional forms rather than fine-tuning a specific function. As a result, the ROF step improves the constants in each function to achieve a better fit and provides uncertainty estimates for the parameters.

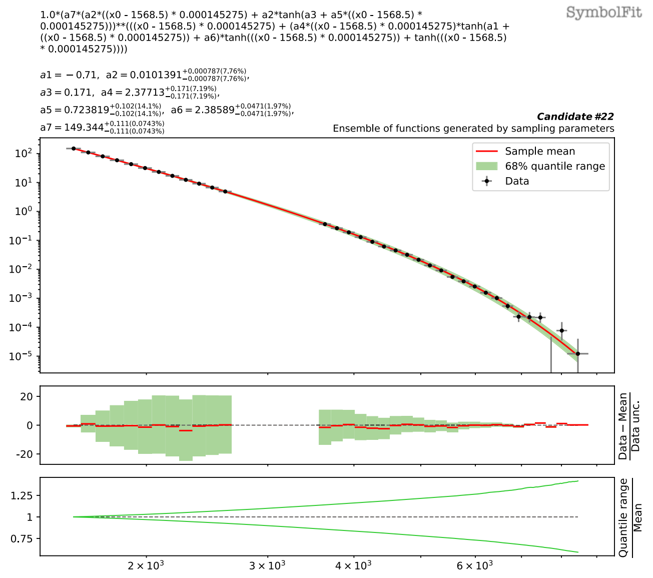
As an example, Fig. 5 shows candidate function #27. This candidate function has six parameters, with uncertainty variations for each plotted separately. The correlation matrix for these parameters is shown in Fig. 6. The correlation matrix, along with the parameter uncertainties,



(a) SR model 1.



(b) SR model 2.



(c) SR model 3.

Fig. 8 The three SR models fitted to the pseudodata of the dijet spectrum with the signal region blinded (see Table 2). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

defines the uncertainty model associated with the function and is required for propagating uncertainties to the final statistical inference results.

To compare various candidate functions obtained from a single fit in the framework, Fig. 7 shows four candidate functions with a range of fit scores from very low to very high. This demonstrates that within the variety of functions generated per fit, there is a convergence from poorer to better-fitted functions, providing flexibility in choices. To illustrate the uncertainty coverage for a candidate function, an ensemble of functions is generated by sampling parameters from a multidimensional normal distribution, considering their best-fit values and covariance matrix. The 68% quantile range of the function ensemble is shown for each candidate function, representing the total uncertainty coverage by simultaneously accounting for uncertainties in all parameters. These computations are all automatically performed within the SymbolFit framework.

CMS Dijet Dataset (1D) [Background Modeling]

CMS performed a search for high-mass dijet resonances using proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV and reported no significant deviations from the Standard Model prediction [25]. The dataset for the dijet spectrum is publicly available on HEPDATA at Ref. [42]. In the analysis, CMS used an empirical 4-parameter function to model the background contribution in the distribution of the dijet invariant mass, m_{jj}:

$$f(x) = \frac{p_0(1-x)^{p_1}}{x^{p_2+p_3\ln(x)}}, \tag{4}$$

where $x = m_{\mathrm{jj}}/\sqrt{s}$ is dimensionless and $p_{\{0,1,2,3\}}$ are free parameters. While this function fits the current dijet spectrum reasonably well, it may be too rigid to accommodate potential future changes in the dijet spectrum due to modifications in analysis strategies or detector performance.

For our experiments, we use the PySR configuration shown in List. 2 to fit the dijet dataset. The main difference between List. 2 and List. 1 used in Sec. "Toy Dataset 1 (1D) [signal modeling]" is that it does not explicitly include a Gaussian operator, as the mass spectrum assumes no peaks in the background. This same SR configuration is also applied to the other LHC datasets as well as Toy Dataset 2, generating a range of well-fitted functions for each case despite their very different distribution shapes, demonstrating the flexibility of the SR approach.

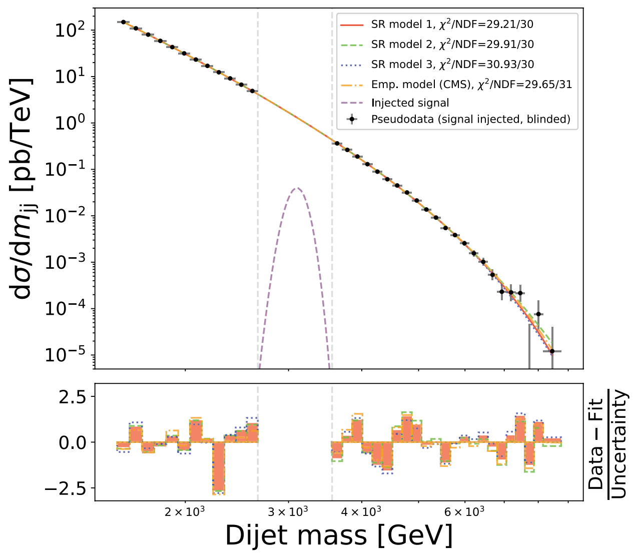
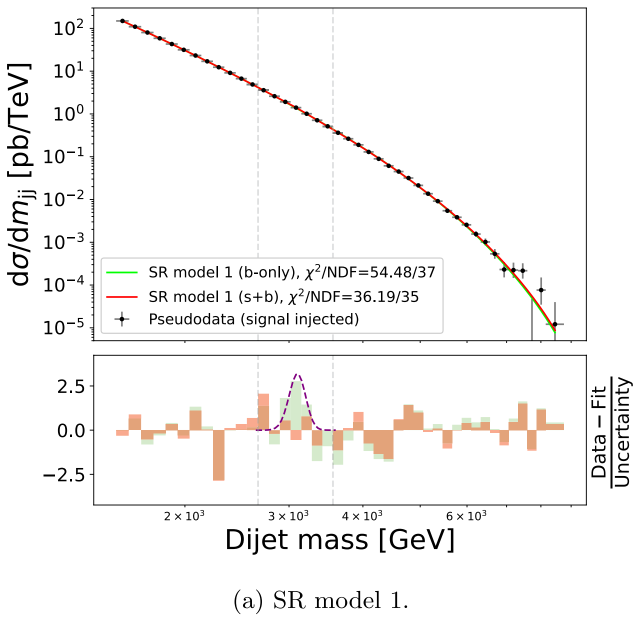


Fig. 9 Pseudodata of the dijet spectrum with the injected signal shown in the blinded signal region. The three SR models (see Table 2) are compared against the empirical model used by CMS. The lower panel shows the residual error per bin, measured in units of the data uncertainty. It can be seen that the three SR models, generated easily from three separate fits using the same simple fit configuration with different random seeds, yield results that are readily comparable to the CMS empirical model that would have required extensive manual effort to obtain

```
from pysr import PySRRegressor

pysr_config = PySRRegressor(
    model_selection = "accuracy",
    timeout_in_seconds = 60*100,
    niterations = 200,
    maxsize = 80,
    binary_operators = [
        "+", "*", "/", "^"
                      ],
    unary_operators = [
        "exp",
        "tanh",
    ],
    nested_constraints = {
        "exp":    {"exp": 0, "tanh": 0, "*": 2, "/": 1, "^": 1},
        "tanh":   {"exp": 0, "tanh": 0, "*": 2, "/": 1, "^": 1},
        "*":      {"exp": 1, "tanh": 1, "*": 2, "/": 1, "^": 1},
        "^":      {"exp": 1, "tanh": 1, "*": 2, "/": 1, "^": 0},
        "/":      {"exp": 1, "tanh": 1, "*": 2, "/": 0, "^": 1},
    },
    loss="loss(y, y_pred, weights) = (y - y_pred)^2 * weights",
)
```

Listing 2 The Python code snippet configures PySR to search for candidate functions for the dijet dataset. This same configuration is used for the other four LHC datasets and Toy Dataset 2 with variations in the maximum complexity values



(a) SR model 1.

(b) SR model 2.

(c) SR model 3.

(d) Empirical model (CMS).

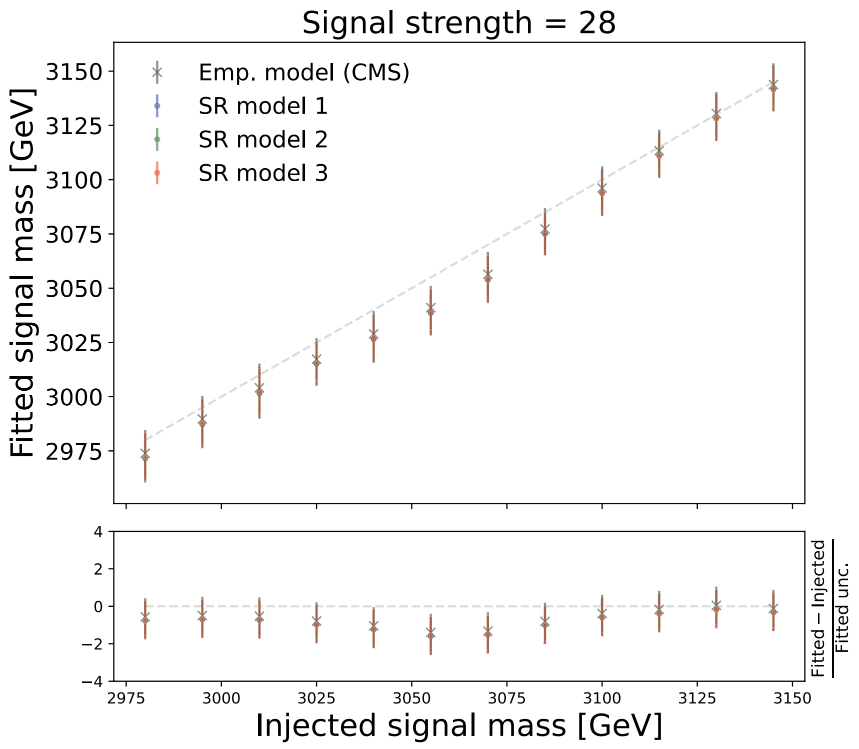
Fig. 10 Comparison of the b-only fits and the s+b fits to the unblinded pseudodata of the dijet spectrum. The lower panel shows the residual error per bin, measured in units of the data uncertainty. The shape of the injected signal is also shown

Table 3 Comparison of the χ^2/NDF scores from three types of fits to the dijet dataset: the b-only fits to the blinded pseudodata, the b-only fits to the unblinded pseudodata, and the s+b fits to the unblinded pseudodata

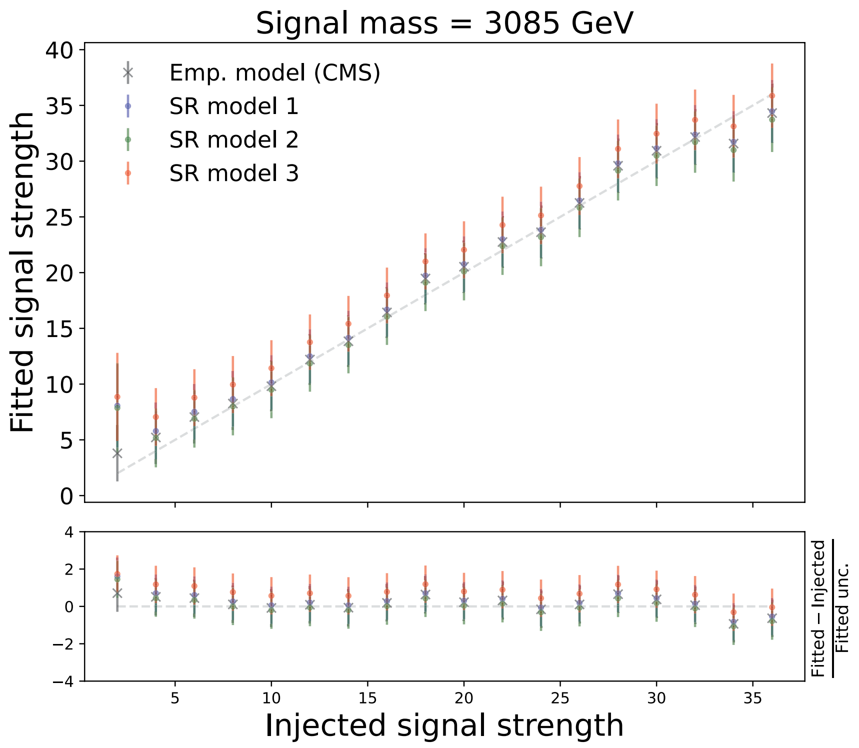
	χ^2/NDF (b-only, blinded)	χ^2/NDF (b-only, unblinded)	χ^2/NDF (s+b, unblinded)
SR model 1	29.21 / 30 = 0.974	54.48 / 37 = 1.47	36.19 / 35 = 1.03
SR model 2	29.91 / 30 = 0.997	51.25 / 37 = 1.39	36.28 / 35 = 1.04
SR model 3	30.93 / 30 = 1.03	65.33 / 37 = 1.77	39.98 / 35 = 1.14
Emp. model (CMS)	29.65 / 31 = 0.956	54.63 / 38 = 1.44	35.07 / 36 = 0.974

The background models used for the fits are listed in Table. 2, and the fits are shown in Fig. 9 (blinded) and Fig. 10 (unblinded)

Fig. 11 Fitted values vs. the true values of the parameters of the injected signal in the dijet dataset. The bottom panels show the residual error in units of the fitted uncertainty



(a) Fitted vs. injected signal mass at a specified signal strength value.



(b) Fitted vs. injected signal strength at a specified signal mass value.

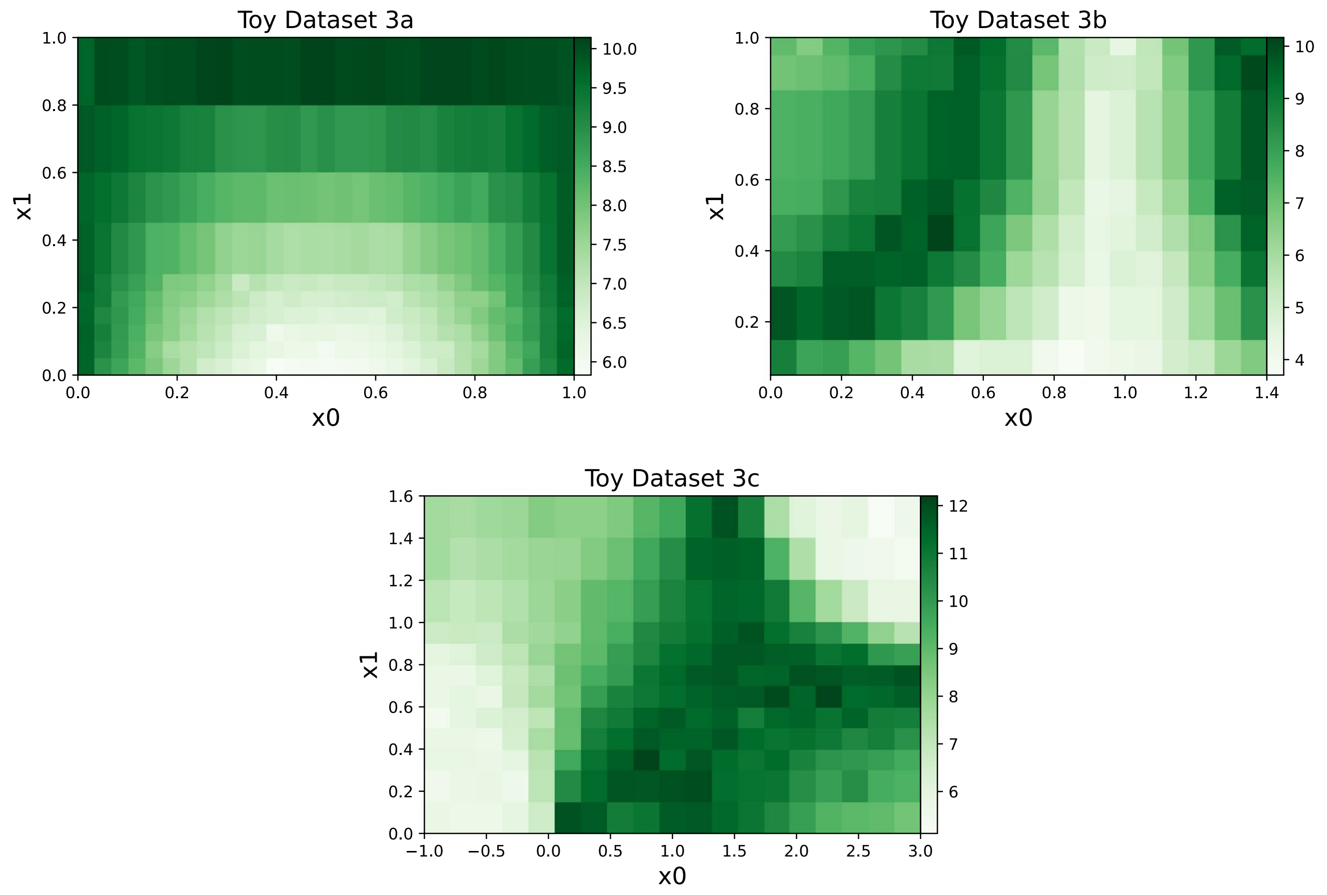


Fig. 12 Toy Dataset 3: three 2D binned sub-datasets manually generated without reference to an underlying function

To validate the SR approach in background modeling and signal extraction, we start with the original dijet spectrum and generate pseudodata by injecting a small and narrow Gaussian signal, $s_0 \frac{1}{\sqrt{2\pi}s_2} \exp\left(-\frac{(x-s_1)^2}{2s_2^2}\right)$, centered at $m_{\mathrm{jj}} = 3.1$ TeV (s_1), with a width of 0.2 TeV ($2s_2$) and a signal strength of $s_0 = 10$. The injected signal intensity per bin is perturbed by 10% random noise.

To model the background, we first blind the signal region by masking the m_{jj} bins near the injected signal peak, specifically between 2.7 and 3.5 TeV, and then perform fits to this blinded pseudodata. We conduct three separate SymbolFit runs on the blinded pseudodata, using the same fit configuration but with different random seeds. This demonstrates that a variety of well-fitted functions can be obtained from the same fit configuration, given the vast function space in the SR search. From each of the three fits, one candidate function is selected, referred to as “SR model 1”, “SR model 2”, and “SR model 3”, respectively. These three background models are then compared with the empirical function in Eq. 4 used by CMS, referred to as the “empirical model (CMS)”.

Table 2 lists the three SR models, each obtained from a fit initialized with a different random seed. The χ^2/NDF scores improve significantly after the ROF step compared to the original functions returned by PySR, with the final scores close to 1. The three background models fit the blinded pseudodata well, as shown in Fig. 8 for the total uncertainty coverage and Fig. 9 for a comparison with the empirical model used by CMS.

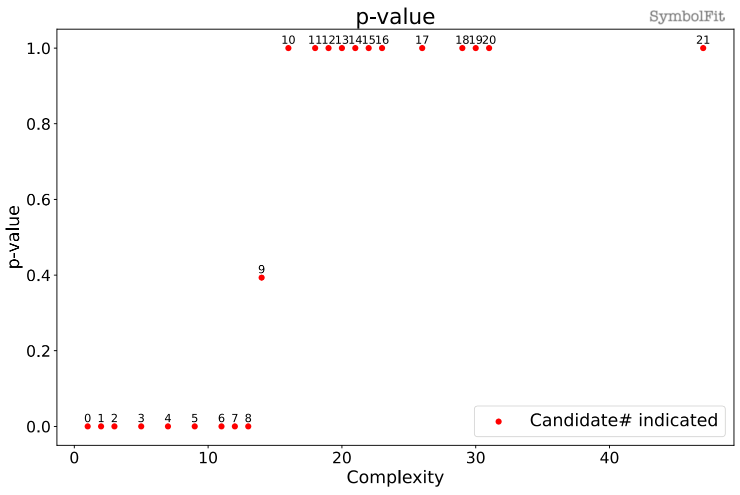
Once the background models are established, we incorporate a parameterized Gaussian signal template into each model $f(x)$:

$$f(x) + s_0 \frac{1}{\sqrt{2\pi}s_2} \exp\left(-\frac{(x-s_1)^2}{2s_2^2}\right). \quad (5)$$

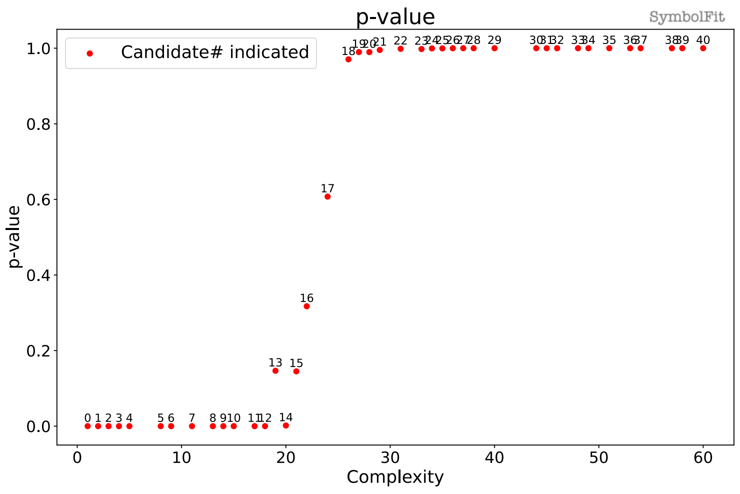
In the following analysis, when the model is fitted to the unblinded pseudodata with $s_0 = 0$ held fixed, it is referred to as a background-only (b-only) fit. When s_0 is allowed to vary, it is referred to as a signal-plus-background (s+b) fit.

Now, we unblind the pseudodata and perform b-only fits and s+b fits on the full dijet spectrum. Since the pseudodata

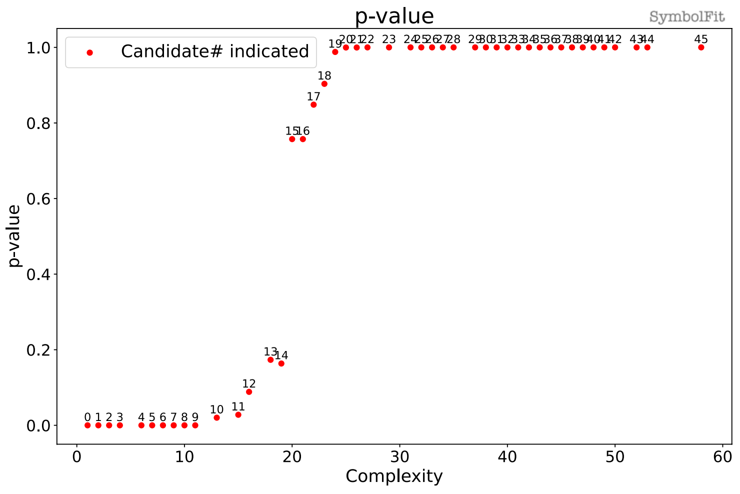
Fig. 13 p-value vs. function complexity. A total of 22, 41, and 46 candidate functions (labeled #0–#21, #0–#40, and #0–#45) were obtained from a single fit to Toy Dataset 3a, 3b, and 3c, respectively



(a) Toy Dataset 3a.



(b) Toy Dataset 3b.



(c) Toy Dataset 3c.

Table 4 Example candidate functions for Toy Dataset 3 are listed

Complexity	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
Toy Dataset 3a					
14 (#9)	$8.08\text{gauss}(x_0(-2.76 + x_1)) + 9.46x_0 + x_1$	3	292.9 / 287 = 1.02	292.9 / 287 = 1.02	0.393
19 (#12)	$9.58 + 14.4x_0(-0.994 + x_0)(\text{gauss}(x_1))^2 + x_1$	3	41.06 / 287 = 0.1431	32.75 / 287 = 0.1141	1.0
20 (#13)	$9.92 + 16.0x_0(-1 + x_0)(-0.541x_1 + \text{gauss}(x_1))$	4	15.8 / 286 = 0.0552	13.22 / 286 = 0.0462	1.0
Toy Dataset 3b					
22 (#16)	$1.38\exp(x_0^2) + 7.08\text{gauss}(-0.736x_1 + x_0^2 + x_0) + \tanh(x_1)$	3	156.7 / 149 = 1.052	156.7 / 149 = 1.052	0.3172
24 (#17)	$1.3\exp(x_0^2) + 6.83\text{gauss}(-0.726x_1 + x_0^2 + x_0) + \tanh(2.48x_1)$	3	144.4 / 149 = 0.969	143.7 / 149 = 0.964	0.607
26 (#18)	$0.345 + 5.73x_1\text{gauss}(x_1) + 5.73\text{gauss}(-0.809x_1 + 2.62x_0^2) + \exp(x_0^2)$	3	118.4 / 149 = 0.7945	118.1 / 149 = 0.7928	0.971
57 (#38)	$4.83\text{gauss}(0.571 + 1.18x_0(2x_0 + x_1) + x_1(-3.07 + x_1)) + 2\text{gauss}(x_1) - \tanh(-4.87 + 4.67x_0 + 2.74x_1) + 0.129\exp(x_0(1.54 + x_0)) + 4.83\tanh(x_1) + 0.335x_1$	8	52.8 / 144 = 0.3667	52.06 / 144 = 0.3615	1.0
Toy Dataset 3c					
20 (#15)	$-0.886x_0(\text{gauss}(x_0x_1) + \exp(x_1)) + 6.74\tanh(x_0) + 8.24$	3	214.2 / 225 = 0.9521	209.9 / 225 = 0.9328	0.7573
23 (#18)	$1.9x_0(-0.296 + \text{gauss}(x_1^2)) + 4.38\text{gauss}(-1.14 + x_0) + 6.56$	5	198.1 / 223 = 0.8881	196 / 223 = 0.8789	0.9035
25 (#20)	$4.33\text{gauss}(-1.14 + x_0) + 6.57 + x_0(-0.238 + \text{gauss}(x_1^2))(1.33 + x_1)$	5	160.3 / 223 = 0.7188	157.3 / 223 = 0.7054	0.9997
42 (#34)	$5.16 + x_1 + (-0.229 + \text{gauss}(-0.481 + x_1^3(-1.25 + x_0)))(1.48x_0 + 1.48\tanh(4.66x_0) + 2.35\text{gauss}(x_1)) + 3.53\text{gauss}(-0.907 + x_0)$	8	59.38 / 220 = 0.2699	54.89 / 220 = 0.2495	1.0

The example candidate functions—#12 for Toy Dataset 3a, #38 for 3b, and #34 for 3c—are plotted in Fig. 14, Fig. 15, and Fig. 16, respectively. Numerical values are rounded to three significant figures for display purposes

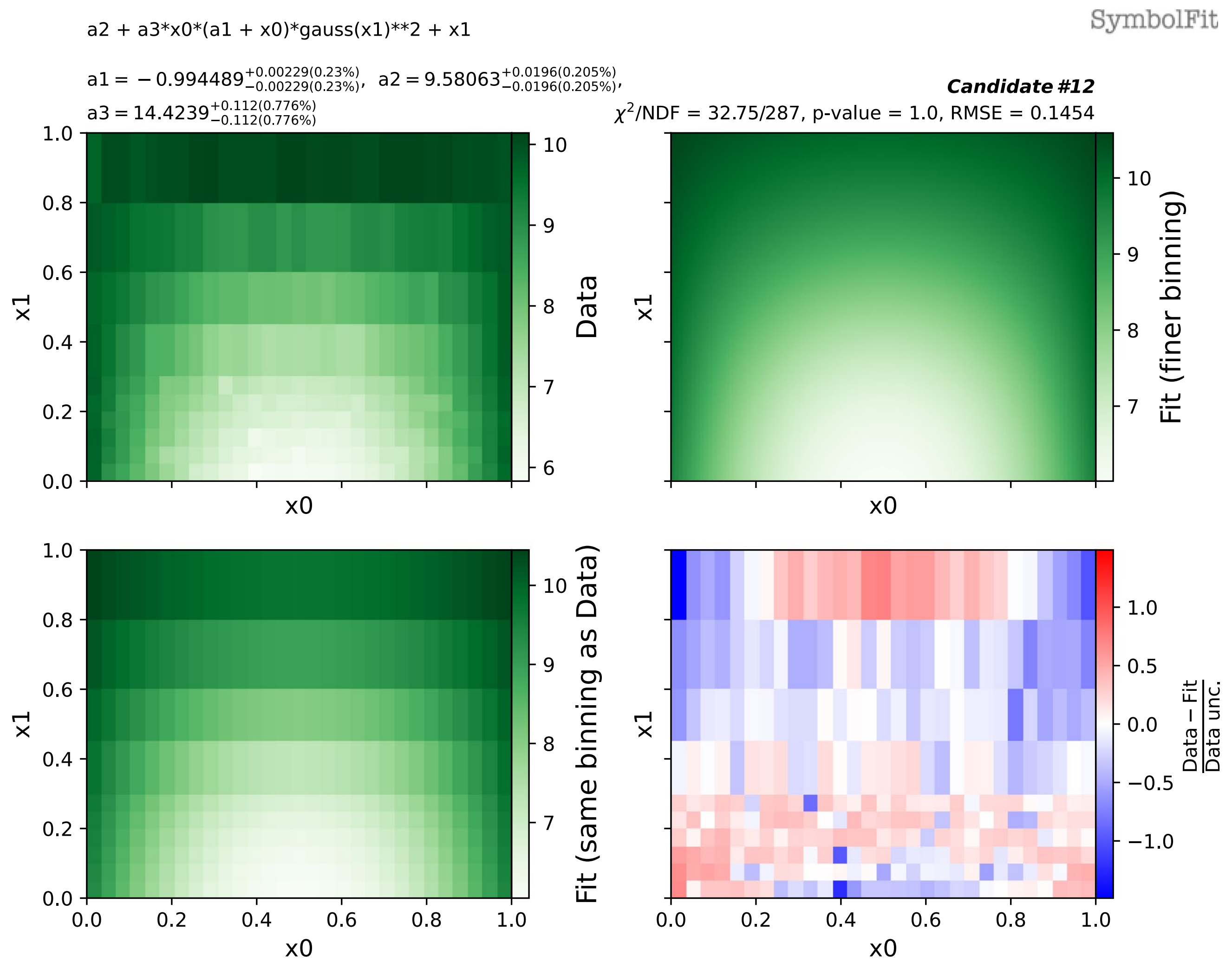


Fig. 14 Candidate function #12 for Toy Dataset 3a (see Table 4). The parameterized form of this function is shown at the top of the figure, along with the best-fit values and associated uncertainties. Upper left: the binned data being fitted. Lower left: the candidate function plotted with the same binning as the fitting data. Upper right: the candidate function plotted with a finer binning. Lower right: the residual error, $\frac{\text{Data}-\text{Fit}}{\text{Uncertainty}}$, in units of the data uncertainty

contain an injected signal, we expect to observe an excess of events over the background model around the signal location, provided the background is properly modeled and not overly fitted. When performing the s+b fits, we expect that the excess of events observed in the b-only fits will diminish as the signal is accounted for by the model template. The results of the b-only and s+b fits for each model are compared and shown in Fig. 10. In all three SR models, as well as the CMS empirical model, the excess of events over the background around the injected signal location observed in the b-only fits is reduced in the s+b fits, demonstrating that the models are sensitive to the injected signal. Table 3 lists the χ^2/NDF scores for each model, indicating the fit performance in response to the presence of the injected signal.

Additionally, to assess whether the SR models can extract the injected signals, we generate multiple sets of pseudodata by injecting Gaussian signals with varying mean values between 2980 to 3150 GeV and signal strengths ranging from 2 to 38. We then perform the s+b fits to extract the corresponding signal parameters. Figure 11 shows the extracted signal parameters (mass and strength) plotted against their injected values. All three SR models are capable of extracting the correct signal parameter values within reasonable uncertainties. They perform comparably to the empirical model used by CMS and, in some cases, yield more accurate fitted values, demonstrating that functions obtained from SR are effective for such tasks.

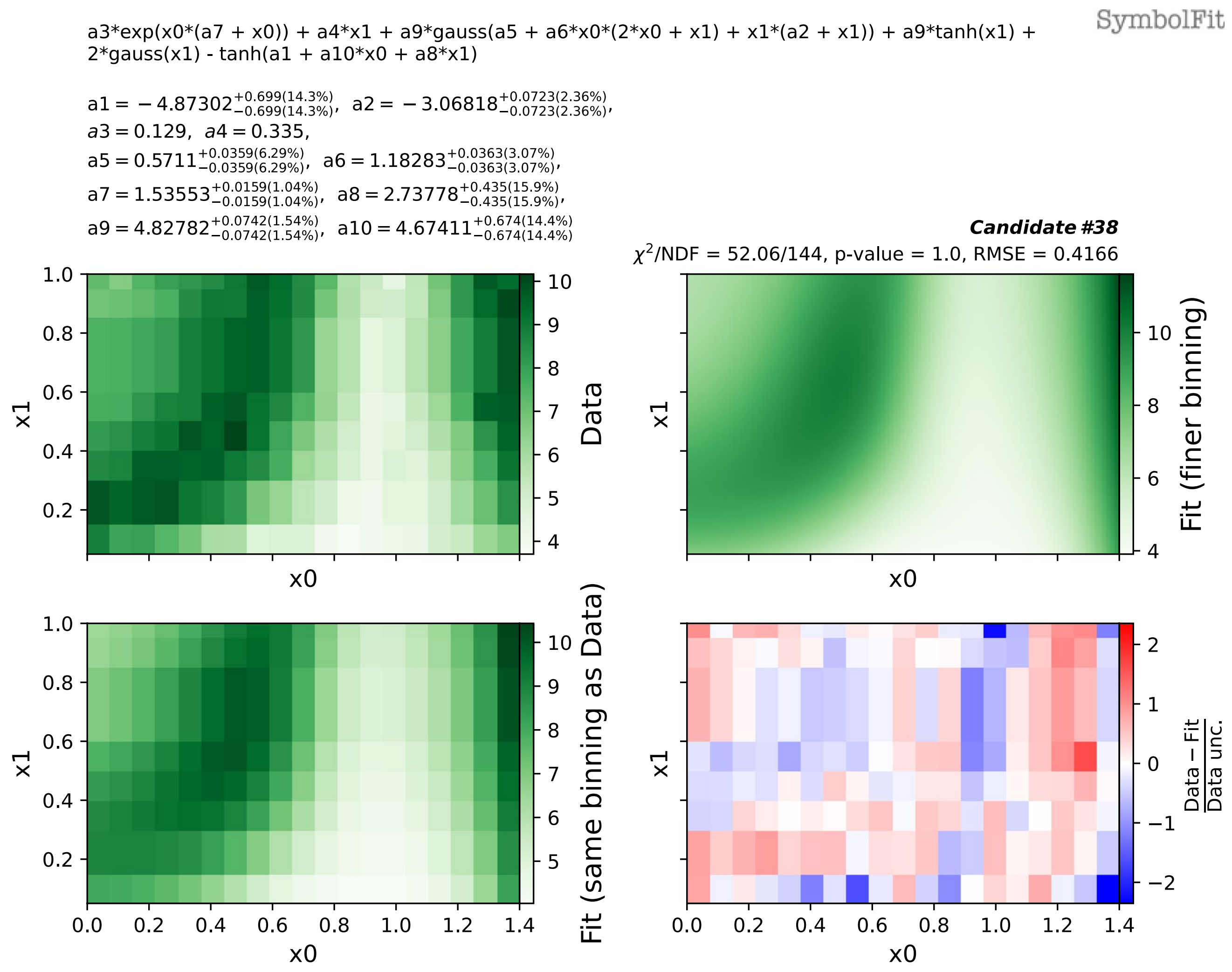


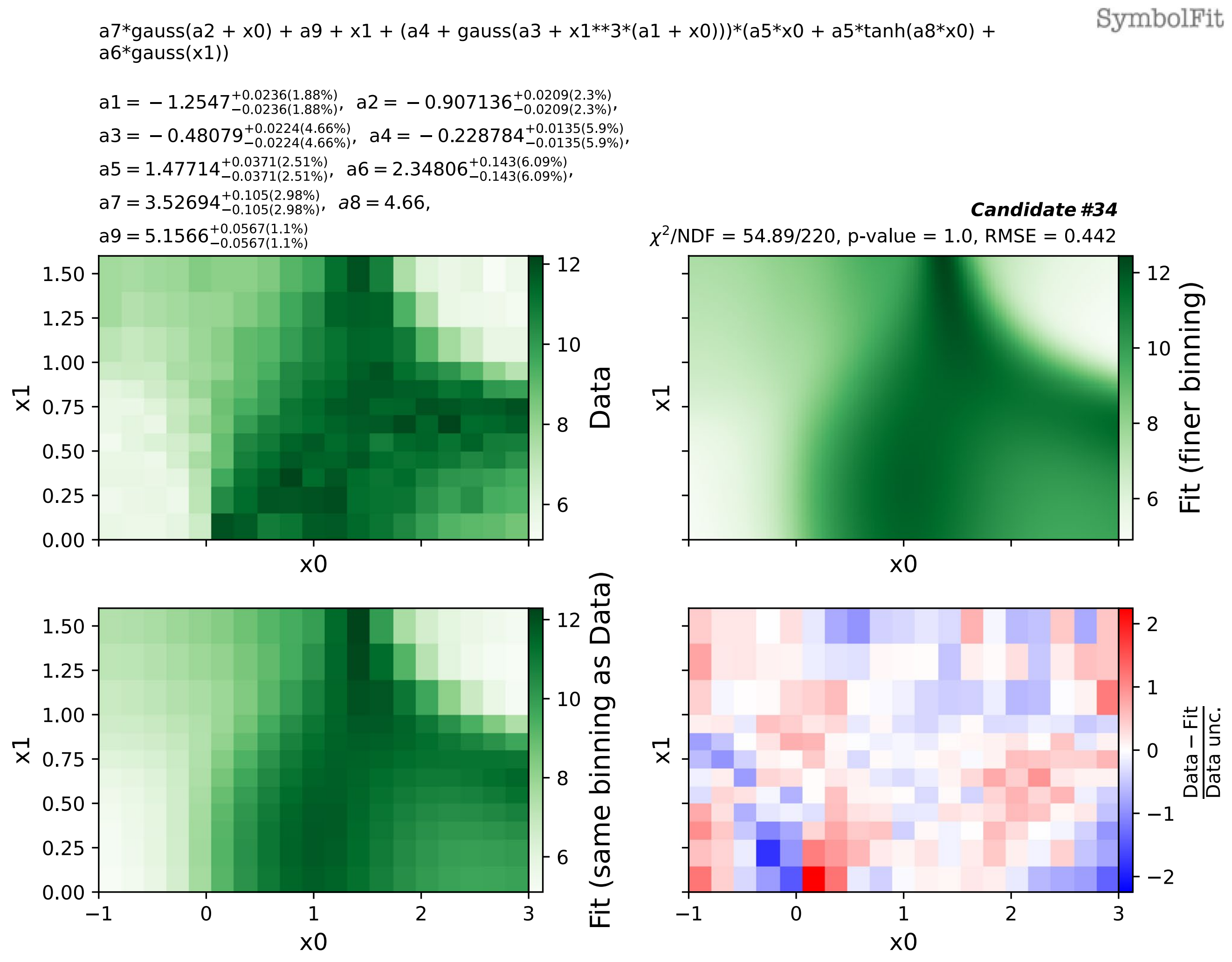
Fig. 15 Candidate function #38 for Toy Dataset 3b (see Table 4). The parameterized form of this function is shown at the top of the figure, along with the best-fit values and associated uncertainties. Upper left: the binned data being fitted. Lower left: the candidate function plotted with the same binning as the fitting data. Upper right: the candidate function plotted with a finer binning. Lower right: the residual error, $\frac{\text{Data}-\text{Fit}}{\text{Uncertainty}}$, in units of the data uncertainty

We demonstrated that our framework can easily generate multiple suitable candidate functions using a very simple fit configuration. By simply changing the random seed, additional candidate functions can be obtained as needed, and these functions are readily comparable to the empirical function obtained by CMS, which required extensive manual and iterative effort. This shows that our method is preferable to traditional methods, as the labor-intensive step of finding adequate functional forms is now automated using machine learning.

Toy Dataset 3 (2D) [Arbitrary Shapes]

In Toy Dataset 3, we consider three 2D binned sub-datasets, labeled 3a, 3b, and 3c, as shown in Fig. 12. These datasets are manually generated without reference to an underlying function and are used to demonstrate applications such as deriving smooth scale factors from binned data with more than one independent variable. The framework can be easily extended to datasets with multiple independent variables.

We use the same PySR configuration applied to Toy Dataset 1, as shown in List. 1, to fit these 2D binned datasets. For each sub-dataset, a single run of SymbolFit is performed to generate a batch of candidate functions. Figure 13 shows the p-value plotted against function complexity. Several



a7*gauss(a2 + x0) + a9 + x1 + (a4 + gauss(a3 + x1**3*(a1 + x0)))*(a5*x0 + a5*tanh(a8*x0) + a6*gauss(x1))

$a1 = -1.2547^{+0.0236(1.88\%)}_{-0.0236(1.88\%)}$, $a2 = -0.907136^{+0.0209(2.3\%)}_{-0.0209(2.3\%)}$,
$a3 = -0.48079^{+0.0224(4.66\%)}_{-0.0224(4.66\%)}$, $a4 = -0.228784^{+0.0135(5.9\%)}_{-0.0135(5.9\%)}$,
$a5 = 1.47714^{+0.0371(2.51\%)}_{-0.0371(2.51\%)}$, $a6 = 2.34806^{+0.143(6.09\%)}_{-0.143(6.09\%)}$,
$a7 = 3.52694^{+0.105(2.98\%)}_{-0.105(2.98\%)}$, $a8 = 4.66$,
$a9 = 5.1566^{+0.0567(1.1\%)}_{-0.0567(1.1\%)}$

Fig. 16 Candidate function #34 for Toy Dataset 3c (see Table 4). The parameterized form of this function is shown at the top of the figure, along with the best-fit values and associated uncertainties. Upper left: the binned data being fitted. Lower left: the candidate function plotted with the same binning as the fitting data. Upper right: the candidate function plotted with a finer binning. Lower right: the residual error, $\frac{\text{Data}-\text{Fit}}{\text{Uncertainty}}$, in units of the data uncertainty

candidate functions for each sub-dataset are selected and listed in Table 4.

A candidate function is shown for each of the three sub-datasets: #26 for sub-dataset 3a in Fig. 14, #39 for sub-dataset 3b in Fig. 15, and #37 for sub-dataset 3c in Fig. 16.

Summary

We have developed a framework called SymbolFit that automates parametric modeling without the need for a priori specification of a functional form to fit data. The framework utilizes symbolic regression to machine-search for suitable functional forms and incorporates a re-optimization step to improve the candidate functions and provide uncertainty estimates. Due to the nature of genetic programming, each symbolic regression fit generates a batch of candidate functions with a variety of forms that can potentially model the data well. This offers flexibility and allows users to select the most suitable candidates for their downstream tasks.

Our primary focus is on applications in high-energy physics data analysis, specifically in signal and background modeling, as well as the derivation of smooth scale factors from binned data. There is no reason it cannot be applied to other fields where parametric modeling is needed. We have demonstrated our framework using five real proton–proton collision datasets from new physics searches at the CERN LHC, as well as several toy datasets, including two-dimensional binned data. Our framework has been shown to easily generate a variety of suitable candidate functions for non-trivial distributions using a very simple fit configuration, even without prior knowledge of the final functional forms. The candidate functions obtained from our framework are readily comparable to the empirical functions derived from traditional methods, which would have required extensive manual and iterative effort. This suggests that our method is preferable to traditional methods since the labor-intensive process of finding adequate functional forms is now automated using machine learning.

Since the fit outputs in this approach are parametric closed-form functions, the resulting model representation is identical to that from traditional parametric modeling methods. This allows seamless integration with established downstream statistical tools used in LHC experiments such as `Combine` and `pyhf` for hypothesis testing. Furthermore, the ease of generating a wide range of well-fitted functions within this framework facilitates flexible modeling, as the choice of functions can be treated as a source of systematic uncertainty using well-established techniques, such as the discrete profiling method.

We have developed an API for the framework, designed for easy use by the high-energy physics community. This API automates and streamlines the process of finding suitable functions with uncertainty estimates for modeling binned data, significantly reducing manual effort. Our goal is to transform the approach to parametric modeling in high-energy physics experiments, moving away from traditional fitting methods that rely on manually determining empirical functions on a case-by-case basis, which is both time-consuming and prone to bias.

Appendix A: More Examples

Toy Dataset 2 (1D) [Arbitrary Shapes]

In Toy Dataset 2, we consider three 1D binned sub-datasets, labeled 2a, 2b, and 2c, as shown in Fig. 17. These datasets are manually generated without reference to an underlying function and are used to demonstrate applications such as deriving smooth scale factors from binned data.

We use the same `PySR` configuration applied to the five LHC datasets, as shown in List. 2, to fit these 1D binned datasets. For each sub-dataset, a single run of `SymbolFit` is performed to generate a batch of candidate functions. Figure 18 shows the p-value plotted against function complexity. Several candidate functions for each sub-datasets are selected and listed in Table 5.

Figure 19 shows candidate functions #13, #21, and #21 with uncertainty coverage, for sub-datasets 2a, 2b, and 2c, respectively.

CMS High-Mass Diphoton Dataset (1D) [Background Modeling]

CMS performed a search for high-mass diphoton resonances using proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV and reported no significant deviations from the Standard Model prediction [27]. The dataset for the diphoton spectrum is publicly available on HEPDATA at Ref. [43]. In the analysis, CMS considered four empirical functions to model the background contribution in the distribution of the diphoton invariant mass, $m_{\gamma\gamma}$, and one of them is:

$$f(x) = p_0 x^{p_1 + p_2 \log(x)}, \tag{A.1}$$

where $x = m_{\gamma\gamma}/\sqrt{s}$ is dimensionless and $p_{\{0,1,2\}}$ are free parameters.

We perform the same experiments conducted on the dijet dataset, as detailed in Sec. "CMS dijet dataset (1D) [background modeling]". Starting from the original diphoton spectrum, we generate pseudodata by injecting a perturbed Gaussian signal centered at $m_{\gamma\gamma} = 1320$ GeV (s_1), with a width of 74 GeV ($2s_2$) and a signal strength of $s_0 = 15$. To model the background, we blind the signal region by masking the $m_{\gamma\gamma}$ bins between 980 and 1400 GeV in the pseudodata and perform the fits.

Three `SymbolFit` runs using different random seeds are carried out, applying the same `PySR` configuration as used for the dijet dataset (see List. 2), except that the maximum complexity is set at 20 instead of 80, since the diphoton distribution shape is less complex. Table 6 lists the three SR models, each obtained from a fit initialized with a different random seed. The χ^2/NDF scores improve significantly after the ROF step compared to the original functions returned by `PySR`. The three background models fit the blinded pseudodata well, as shown in Fig. 20 for the total uncertainty coverage and Fig. 21 for a comparison with the empirical model used by CMS.

Next, we unblind the pseudodata and perform b-only fits and s+b fits on the full pseudodata spectrum. These results are shown in Fig. 22. In all three SR models, as well as the

CMS empirical model, the excess of events over the background around the injected signal location observed in the b-only fits is reduced in the s+b fits, demonstrating that the models are sensitive to the injected signal. Table 7 lists the χ^2/NDF scores for each model, showing the fit performance in response to the presence of the injected signal.

To assess whether the SR models can accurately extract the injected signals, we generate multiple sets of pseudodata by injecting Gaussian signals with different mean values ranging from 1080 to 1120 GeV and varying signal strengths between 5 and 50. We then perform the s+b fits to extract the corresponding signal parameters. Figure 23 shows the extracted signal parameters plotted against their injected values. All three SR models are capable of extracting the correct signal parameter values within reasonable uncertainties and are comparable to the empirical model used by CMS.

CMS Trijet Dataset (1D) [Background Modeling]

CMS performed a search for high-mass trijet resonances using proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV and reported no significant deviations from the Standard Model prediction [20]. The dataset for the trijet spectrum is publicly available on HEPDATA at Ref. [44]. In the analysis, CMS considered four empirical functions to model the background contribution in the distribution of the trijet invariant mass, m_{jjj}, and one of them is:

$$f(x) = \frac{p_0(1-x)^{p_1}}{x^{p_2+p_3\log(x)}}, \tag{A.2}$$

where $x = m_{\mathrm{jjj}}/\sqrt{s}$ is dimensionless and $p_{\{0,1,2,3\}}$ are free parameters. Equation A.2 corresponds to Eq. 1 with $N = 3$ determined by an F-test.

We perform the same experiments conducted on the dijet dataset, as detailed in Sec. "CMS dijet dataset (1D) [background modeling]". Starting from the original trijet spectrum, we generate pseudodata by injecting a perturbed Gaussian signal centered at $m_{\mathrm{jjj}} = 4000$ GeV (s_1) with a width of 400 GeV ($2s_2$) and a signal strength of $s_0 = 50000$. To model the background, we blind the signal region by masking the m_{jjj} bins between 3000 and 5000 GeV in the pseudodata and perform the fits.

Three SymbolFit runs using different random seeds are carried out, applying the same PySR configuration as used

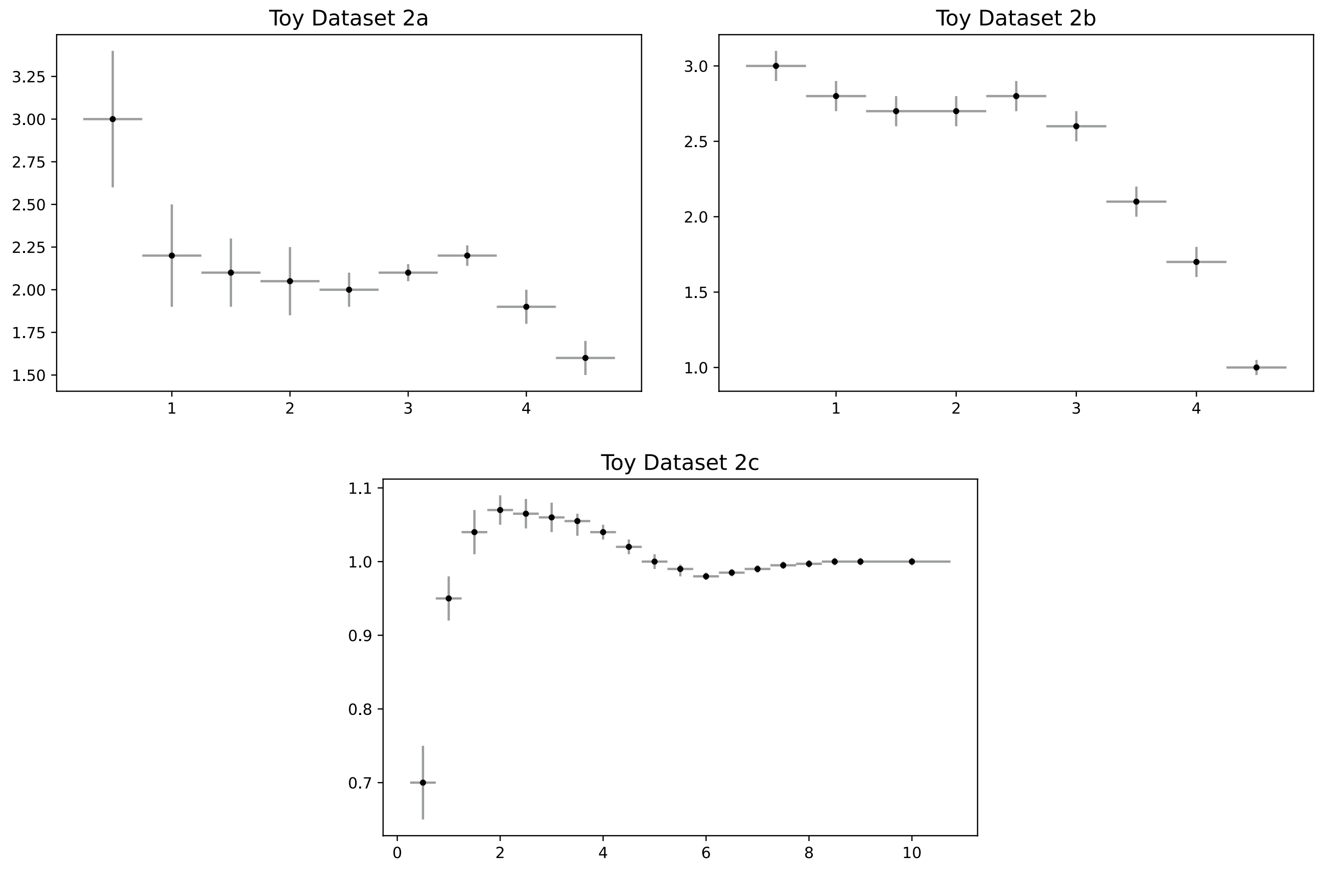
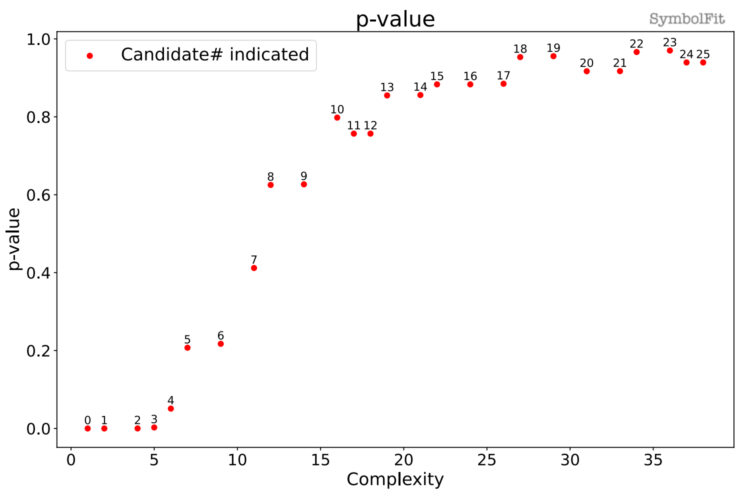
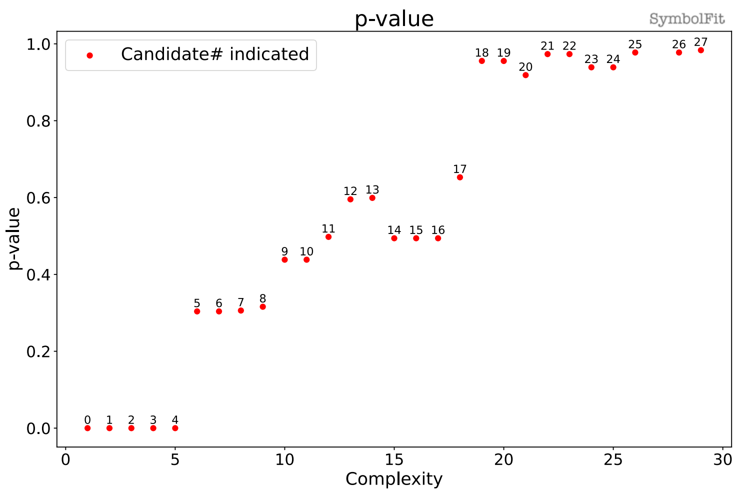


Fig. 17 Toy Dataset 2: three 1D binned sub-datasets manually generated without reference to an underlying function

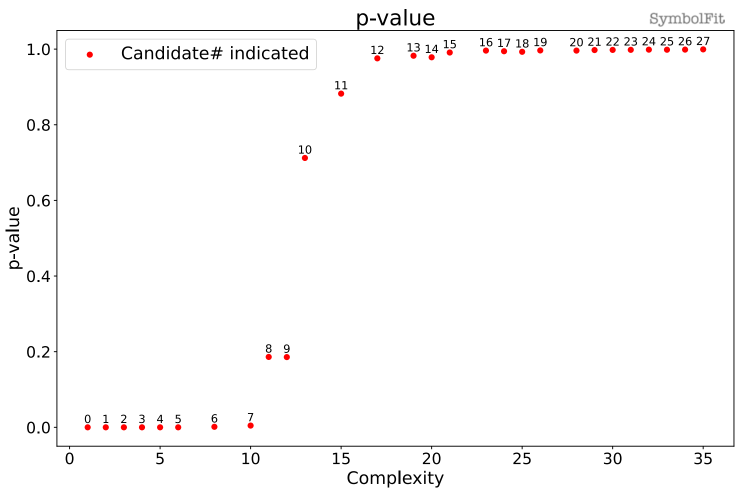
Fig. 18 *p*-value vs. function complexity. A total of 26, 28, and 28 candidate functions (labeled #0–#25, #0–#27, and #0–#27) were obtained from a single fit to Toy Dataset 2a, 2b, and 2c, respectively



(a) Toy Dataset 2a.



(b) Toy Dataset 2b.



(c) Toy Dataset 2c.

Table 5 Example candidate functions for Toy Dataset 2 are listed

Toy Dataset 2a					
Complexity	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
12 (#8)	$0.667 \times 0.991^{\exp(x)} x + 1.44/x$	3	5.273 / 6 = 0.8789	4.382 / 6 = 0.7303	0.6251
19 **(#13)**	$0.697 \times 0.991^{\exp(x)} x + 0.697^{\exp(x)} x + 1.16/x$	3	2.863 / 6 = 0.4772	2.619 / 6 = 0.4365	0.8549
38 (#25)	$0.907^{0.0993 \exp(x)} x(0.0609 + \tanh(0.596^{0.632/x}))+ 0.948/x + 0.948^{\exp(1.63x)} - 0.115$	3	1.776 / 6 = 0.2961	1.771 / 6 = 0.2951	0.9395
Toy Dataset 2b					
Complexity	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
14 (#13)	$-0.0219 \exp(x) + 1.96 + \tanh(x)^{-0.586+x}$	2	5.613 / 7 = 0.8018	5.501 / 7 = 0.7858	0.5991
19 (#18)	$-0.0161 \exp(x) + 0.177 + 3.02 \times 0.825^x + \tanh(0.177x^x)$	4	2.092 / 5 = 0.4184	1.084 / 5 = 0.2169	0.9555
22 **(#21)**	$-0.0149 \exp(x) - 0.00659 + 4.11 \tanh(0.787^x) + \tanh(0.16x^x)$	4	0.857 / 5 = 0.1714	0.8561 / 5 = 0.1712	0.9733
Toy Dataset 2c					
Complexity	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
13 (#10)	$0.626x \exp(-1.38^x) + \tanh(x^{0.502})$	3	16.75 / 16 =1.047	12.45 / 16 = 0.7784	0.7122
15 (#11)	$x/(2x + \exp(1.41^x)) + \tanh(x^{0.505})$	2	10.67 / 17 = 0.6276	10.48 / 17 = 0.6165	0.8823
29 **(#21)**	$(1.02x + 1.02 \tanh(x^2)) \tanh(x)/(2.58x^{1.39} + x+ \exp(1.39^x)) + 1.02 \tanh(x^{0.389})$	4	7.767 / 15 = 0.5178	4.079 / 15 = 0.2719	0.9975

The example candidate functions—#13 for Toy Dataset 2a, #21 for 2b, and #21 for 2c—are plotted in Fig. 19. Numerical values are rounded to three significant figures for display purposes

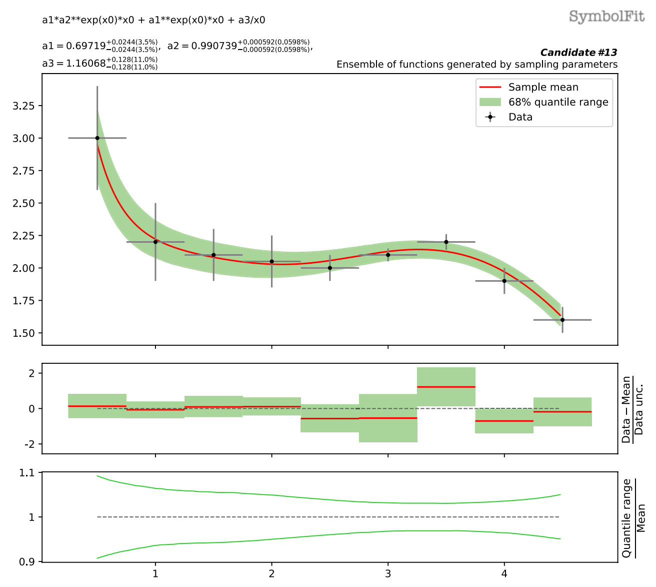
for the dijet dataset (see List. 2). Table 8 lists the three SR models, each obtained from a fit initialized with a different random seed. The χ^2/NDF scores improve significantly after the ROF step compared to the original functions returned by PySR. The three background models fit the blinded pseudodata well, as shown in Fig. 24 for the total uncertainty coverage and Fig. 25 for a comparison with the empirical model used by CMS.

Next, we unblind the pseudodata and perform b-only fits and s+b fits on the full pseudodata spectrum. These results are shown in Fig. 26. In all three SR models, as well as the CMS empirical model, the excess of events over the background around the injected signal location observed in the b-only fits is reduced in the s+b fits, demonstrating that the models are sensitive to the injected signal. Table 9 lists the χ^2/NDF scores for each model, showing the fit performance in response to the presence of the injected signal.

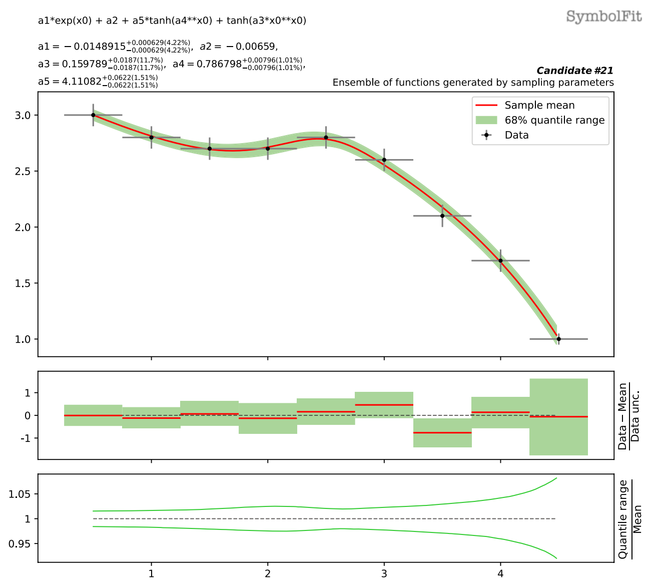
To assess whether the SR models can accurately extract the injected signals, we generate multiple sets of pseudodata by injecting Gaussian signals with different mean values ranging from 3600 to 4500 GeV and varying signal strength between 25000 and 100000. We then perform the s+b fits to extract the corresponding signal parameters. Figure 27 shows the extracted signal parameters plotted against their injected values. All three SR models are capable of extracting the correct signal parameter values within reasonable uncertainties and are comparable to the empirical model used by CMS.

CMS Paired-Dijet Dataset (1D) [Background Modeling]

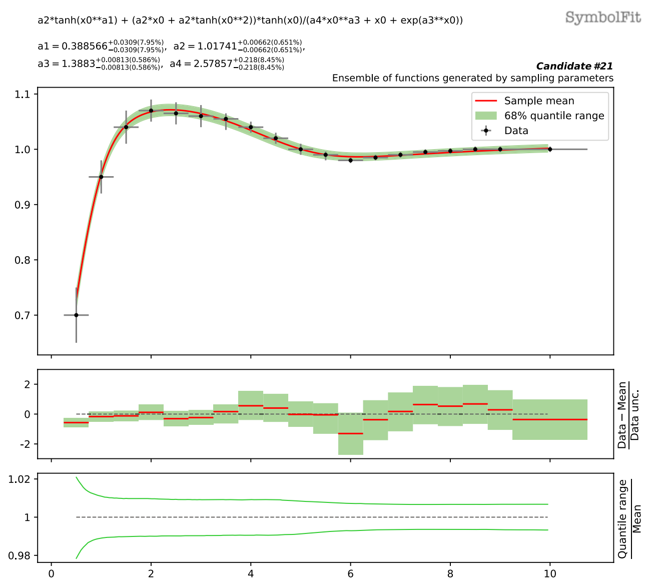
CMS performed a search for high-mass four-jet resonances using proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV and reported no significant deviations from the Standard Model prediction [26]. The dataset for the four-jet spectrum is publicly available on HEPDATA at Ref. [45]. In the analysis, CMS considered four empirical



(a) Candidate function #13 for Toy Dataset 2a.

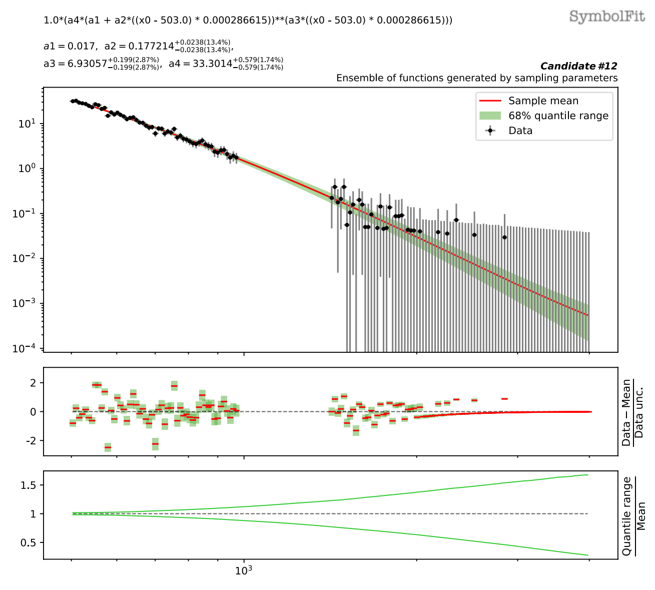


(b) Candidate function #21 for Toy Dataset 2b.

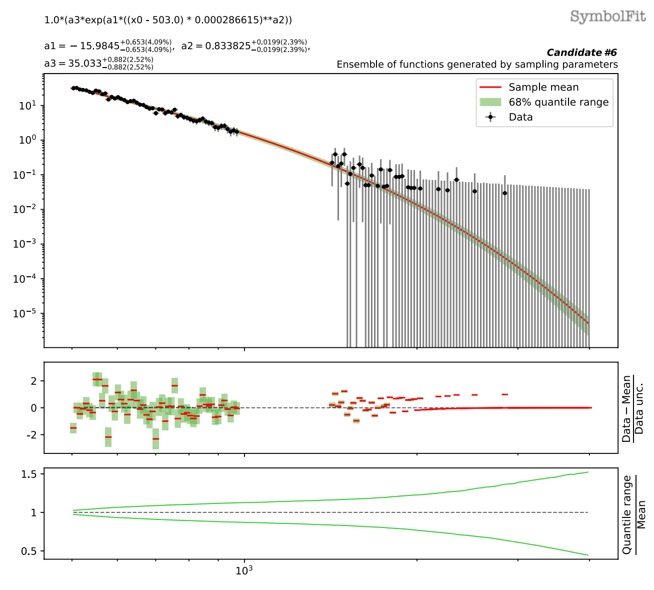


(c) Candidate function #21 for Toy Dataset 2c.

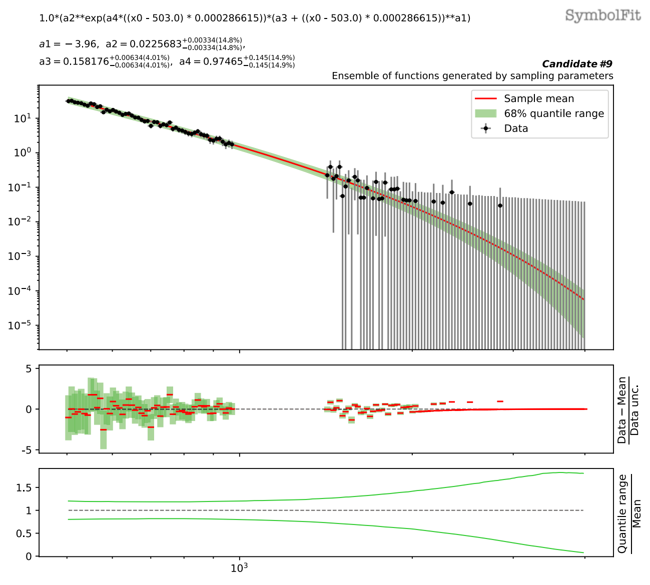
Fig. 19 Example candidate functions for Toy Dataset 2 (see Table 5). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multi-dimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean



(a) SR model 1.



(b) SR model 2.



(c) SR model 3.

Fig. 20 The three SR models fitted to the pseudodata of the diphoton spectrum with the signal region blinded (see Table 6). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

Table 6 The candidate functions are obtained from three fits using different random seeds, fitted to the pseudodata of the diphoton spectrum with the (injected) signal region blinded

	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
SR model 1	$33.3(0.017 + 0.177x)^{6.93x}$	3	47.12 / 136 = 0.3465	46.83 / 136 = 0.3443	1.0
SR model 2	$35.0\exp(-16.0x^{0.834})$	3	63.44 / 136 = 0.4665	53.37 / 136 = 0.3925	1.0
SR model 3	$0.0226\exp(0.975x)(0.158 + x)^{-3.96}$	3	48.94 / 136 = 0.3598	47.22 / 136 = 0.3472	1.0

The fits were performed on a scaled dataset (to enhance fit stability and prevent numerical overflow), and the functions can be transformed back to describe the original spectrum using the transformation: x → 0.000287(x − 503). These functions are plotted and compared with the blinded pseudodata in Fig. 21. Numerical values are rounded to three significant figures for display purposes

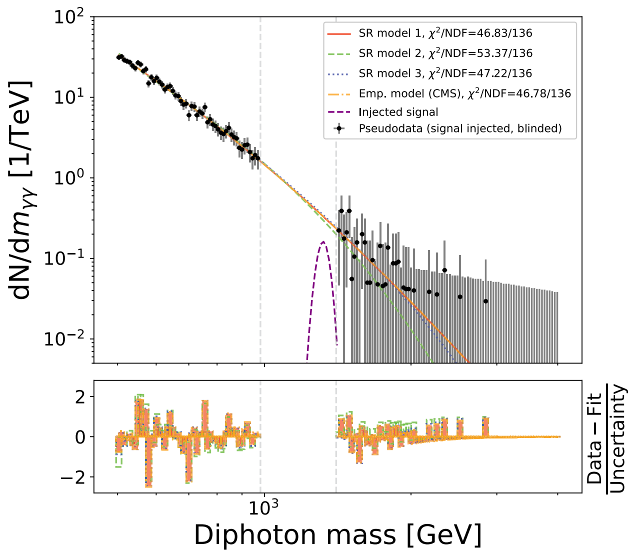


Fig. 21 Pseudodata of the diphoton spectrum with the injected signal shown in the blinded signal region. The three SR models (see Table 6) are compared against the empirical model used by CMS. The lower panel shows the residual error per bin, measured in units of the data uncertainty. It can be seen that the three SR models, generated easily from three separate fits using the same simple fit configuration with different random seeds, yield results that are readily comparable to the CMS empirical model that would have required extensive manual effort to obtain

functions to model the background contribution in the distribution of the four-jet invariant mass, m_{jjjj}, and one of them is:

$$f(x) = \frac{p_0(1 - x^{1/3})^{p_1}}{x^{p_2 + p_3 \log x + p_4 \log^2 x}}, \tag{A.3}$$

where $x = m_{\mathrm{jjjj}}/\sqrt{s}$ is dimensionless and $p_{\{0,1,2,3,4\}}$ are free parameters.

We perform the same experiments conducted on the dijet dataset, as detailed in Sec. "CMS dijet dataset (1D) [background modeling]". Starting from the original four-jet spectrum, we generate pseudodata by injecting a perturbed Gaussian signal centered at $m_{\mathrm{jjjj}} = 3500$ GeV (s_1) with a width of 400 GeV ($2s_2$) and a signal strength of $s_0 = 2$. To model the background, we blind the signal region by masking the m_{jjjj} bins between 3000 and 4000 GeV in the pseudodata and perform the fits.

Three SymbolFit runs using different random seeds are carried out, applying the same PySR configuration as used for the dijet dataset (see List. 2). Table 10 lists the three SR models, each obtained from a fit initialized with a different random seed. The χ^2/NDF scores improve significantly after the ROF step compared to the original functions returned by PySR. The three background models fit the blinded pseudodata well, as shown in Fig 28 for the total uncertainty coverage and Fig. 29 for a comparison with the empirical model used by CMS.

Next, we unblind the pseudodata and perform b-only fits and s+b fits on the full pseudodata spectrum. These results are shown in Fig. 30. In all three SR models, as well as the CMS empirical model, the excess of events over the background around the injected signal location observed in the b-only fits is reduced in the s+b fits, demonstrating that the models are sensitive to the injected signal Table 11 lists the χ^2/NDF scores for each model, showing the fit performance in response to the presence of the injected signal.

To assess whether the SR models can accurately extract the injected signals, we generate multiple sets of pseudodata by injecting Gaussian signals with different mean values ranging from 3350 to 3750 GeV and varying signal strength between 0.5 and 10. We then perform the s+b fits to extract the corresponding signal parameters. Figure 31 shows the extracted signal parameters plotted against their injected values. All three SR models are capable of extracting the correct signal parameter values

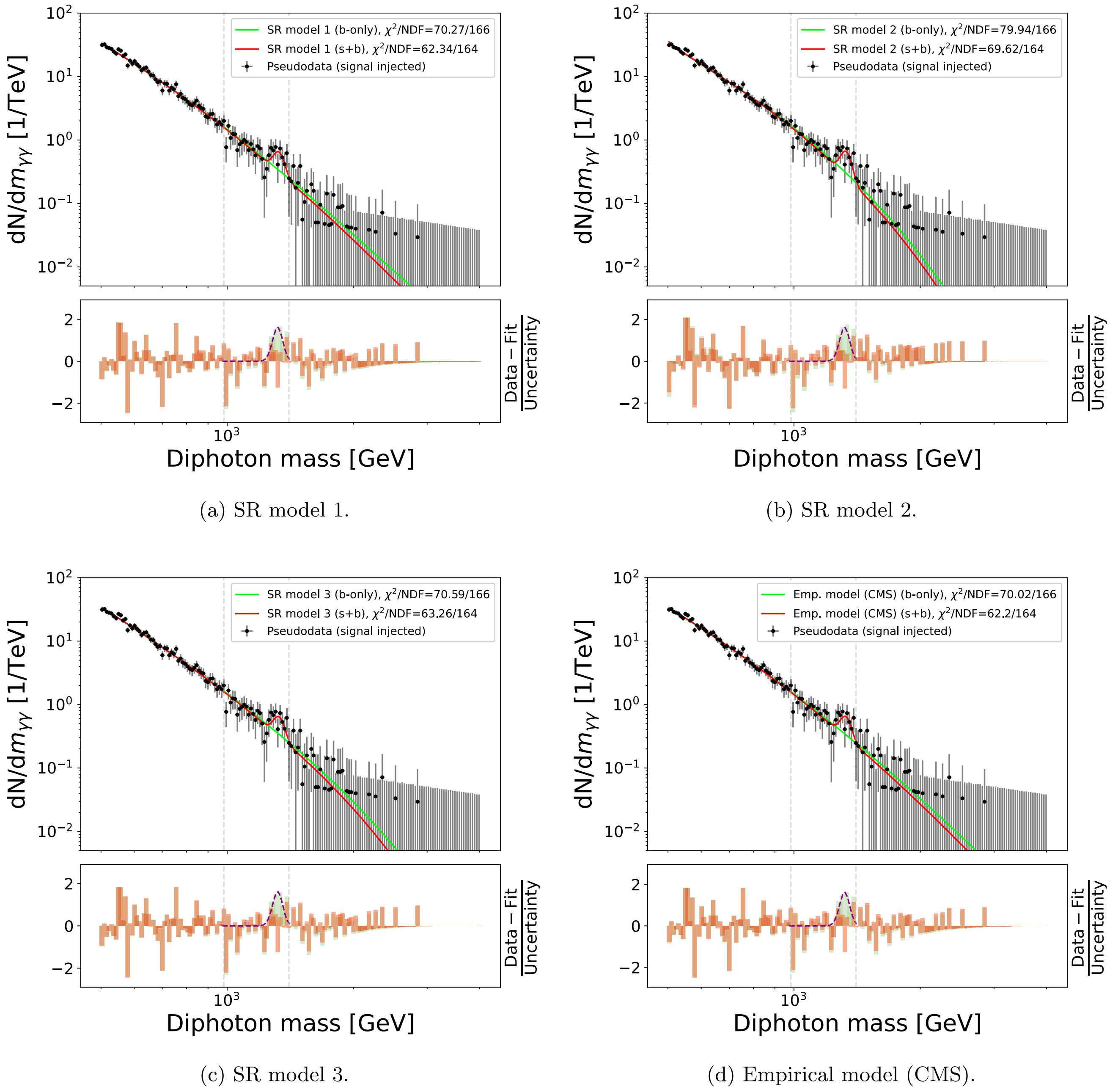


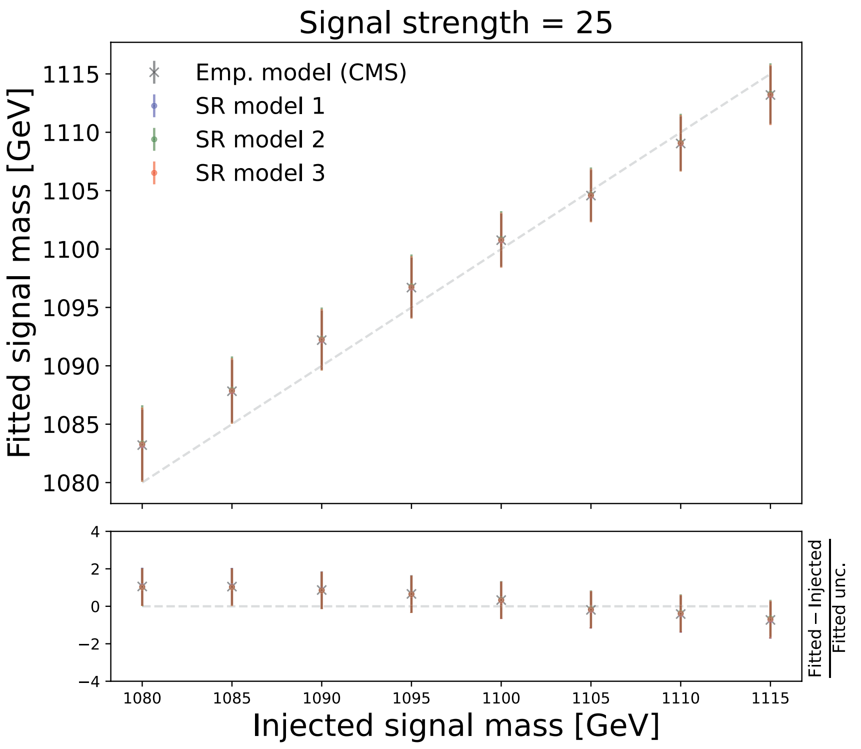
Fig. 22 Comparison of the b-only fits and the s+b fits to the unblinded pseudodata of the diphoton spectrum. The lower panel shows the residual error per bin, measured in units of the data uncertainty. The shape of the injected signal is also shown

Table 7 Comparison of the χ^2/NDF scores from three types of fits to the diphoton dataset: the b-only fits to the blinded pseudodata, b-only fits to the unblinded pseudodata, and s+b fits to the unblinded pseudodata

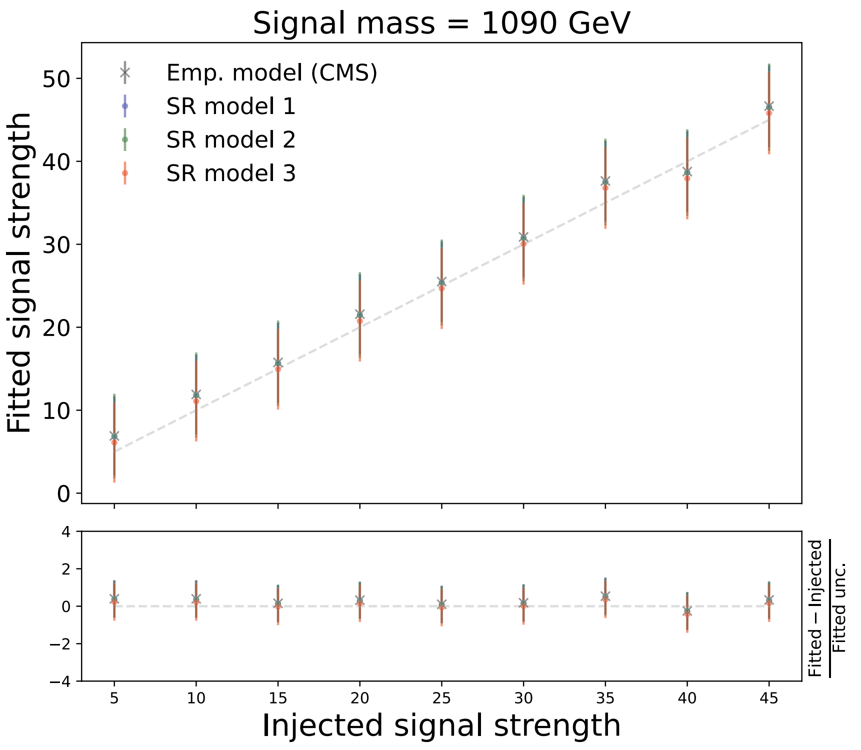
	χ^2/NDF (b-only, blinded)	χ^2/NDF (b-only, unblinded)	χ^2/NDF (s+b, unblinded)
SR model 1	46.83 / 136 = 0.3443	70.27 / 166 = 0.4233	62.34 / 164 = 0.3801
SR model 2	53.37 / 136 = 0.3924	79.94 / 166 = 0.4816	69.62 / 164 = 0.4245
SR model 3	47.22 / 136 = 0.3472	70.59 / 166 = 0.4252	63.26 / 164 = 0.3857
Emp. model (CMS)	46.78 / 136 = 0.344	70.02 / 166 = 0.4218	62.2 / 164 = 0.3793

The background models used for the fits are listed in Table 6, and the fits are shown in Fig. 21 (blinded) and Fig. 22 (unblinded)

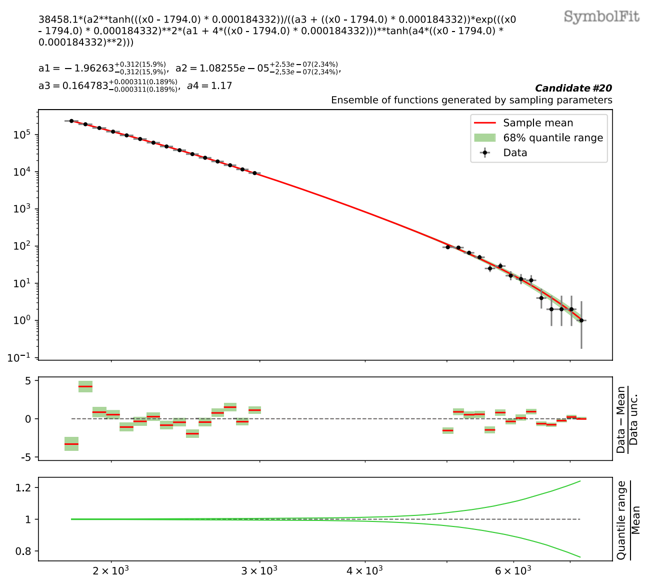
Fig. 23 Fitted values vs. the true values of parameters of the injected signal in the diphoton dataset. The bottom panels show the residual error in units of the fitted uncertainty



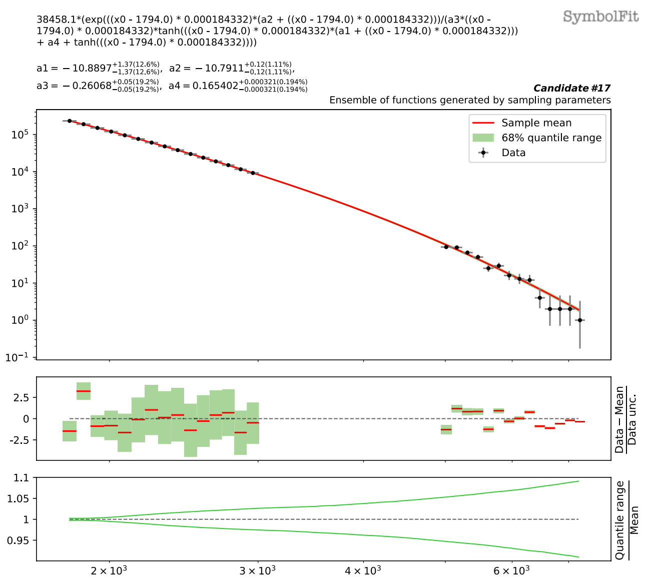
(a) Fitted vs. injected signal mass at a specified signal strength value.



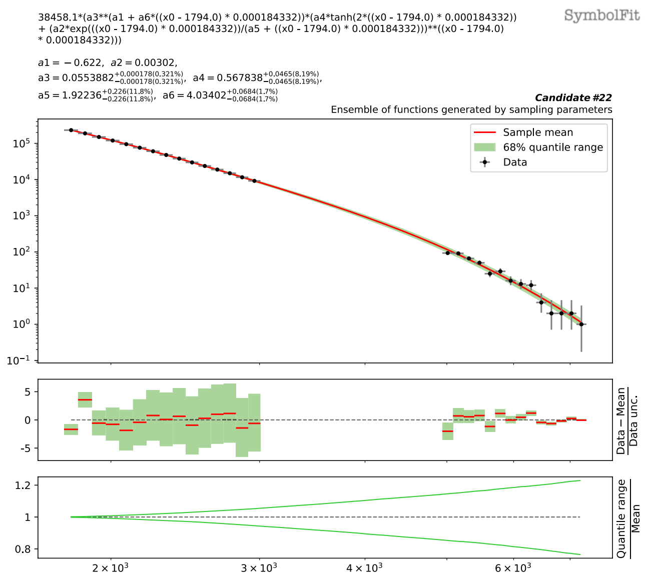
(b) Fitted vs. injected signal strength at a specified signal mass value.



(a) SR model 1.



(b) SR model 2.



(c) SR model 3.

Fig. 24 The three SR models fitted to the pseudodata of the trijet spectrum with the signal region blinded (see Table 8). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

Table 8 The candidate functions are obtained from three fits using different random seeds, fitted to the pseudodata of the trijet spectrum with the (injected) signal region blinded

	Candidate function (After ROF)	# param.	χ^2/NDF (Before ROF)	χ^2/NDF (After ROF)	p-value (After ROF)
SR model 1	$(1.08\times10^{-5})^{\tanh(x)}/((0.165+x)\times \exp(x^2(-1.96+4x))^{\tanh(1.17x^2)})$	3	50.46 / 26 = 1.941	49.04 / 26 = 1.886	0.00408
SR model 2	$\exp(x(-10.8+x))/(-0.261x\tanh(x\times (-10.9+x))+0.165+\tanh(x))$	4	39.49 / 25 = 1.58	33.15 / 25 = 1.326	0.1273
SR model 3	$0.0554^{-0.622+4.03x}(0.568\tanh(2x)+ (0.00302\exp(x)/(1.92+x))^x)$	4	38.4 / 25 = 1.536	37.31 / 25 = 1.492	0.05395

The fits were performed on a scaled dataset (to enhance fit stability and prevent numerical overflow), and the functions can be transformed back to describe the original spectrum using the transformation: $f(x) \rightarrow 38458 \times f(0.000184(x-1790))$. These functions are plotted and compared with the blinded pseudodata in Fig. 25. Numerical values are rounded to three significant figures for display purposes

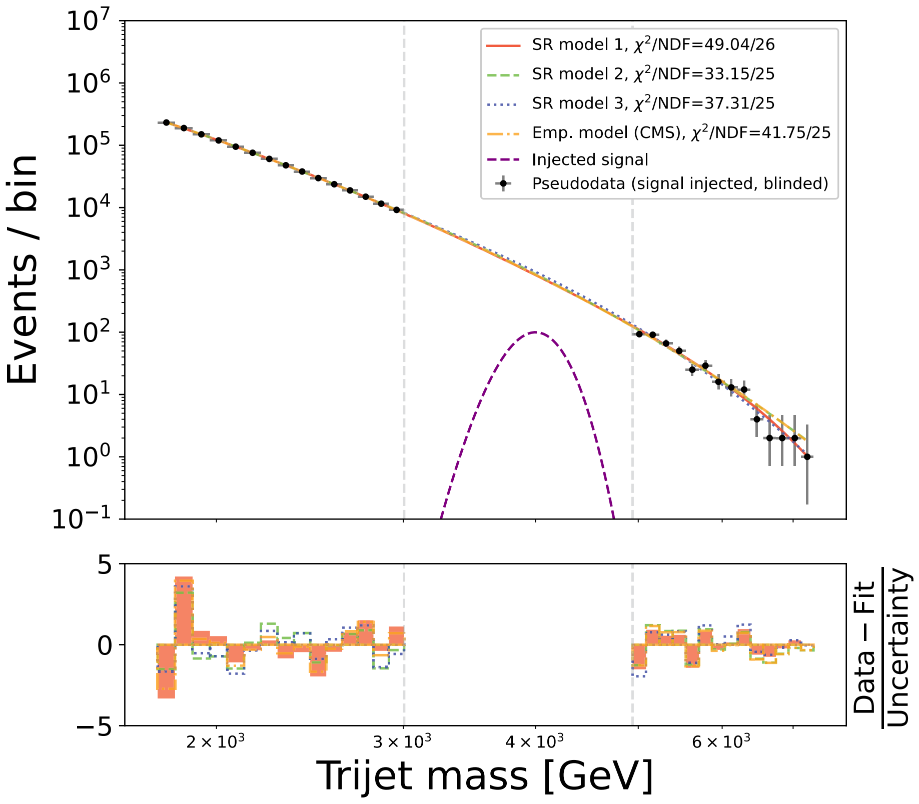


Fig. 25 Pseudodata of the trijet spectrum with the injected signal shown in the blinded signal region. The three SR models (see Table 8) are compared against the empirical model used by CMS. The lower panel shows the residual error per bin, measured in units of the data uncertainty. It can be seen that the three SR models, generated easily from three separate fits using the same simple fit configuration with different random seeds, yield results that are readily comparable to the CMS empirical model that would have required extensive manual effort to obtain

within reasonable uncertainties and are comparable to the empirical model used by CMS.

CMS High-Mass Dimuon Dataset (1D) [Background Modeling]

CMS performed a search for high-mass dimuon resonances using proton–proton collision data at a center-of-mass energy of $\sqrt{s} = 13$ TeV and reported no significant deviations from the Standard Model prediction [28]. The dataset for the dimuon spectrum is publicly available on HEPDATA at Ref. [46]. In the analysis, CMS considered three different functions to model the background contribution in the distribution of the dimuon invariant mass, $m_{\mu\mu}$. These functions include a simple exponential, a power-law, and a first-order Bernstein polynomial. Since the dimuon distribution in the

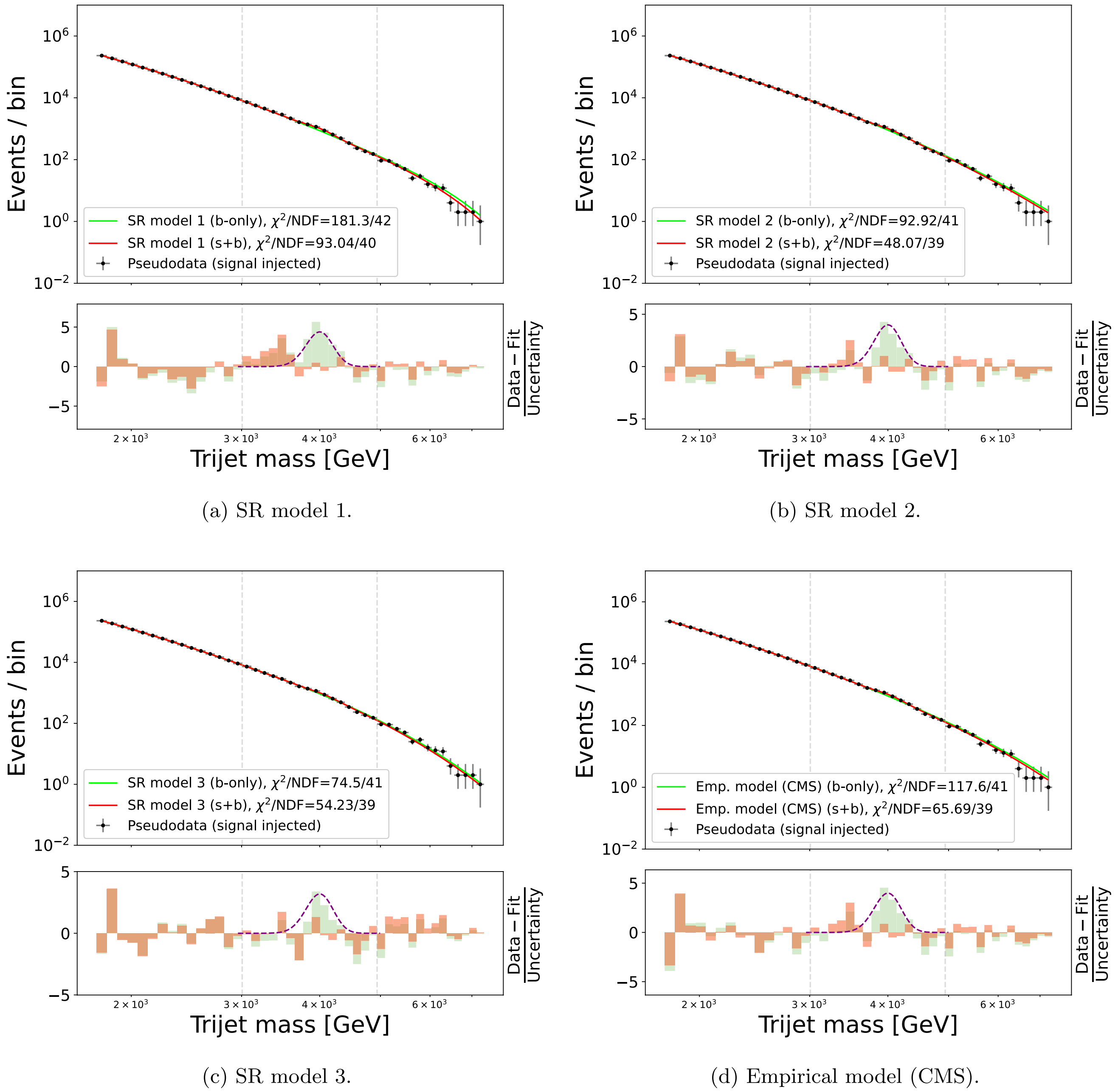


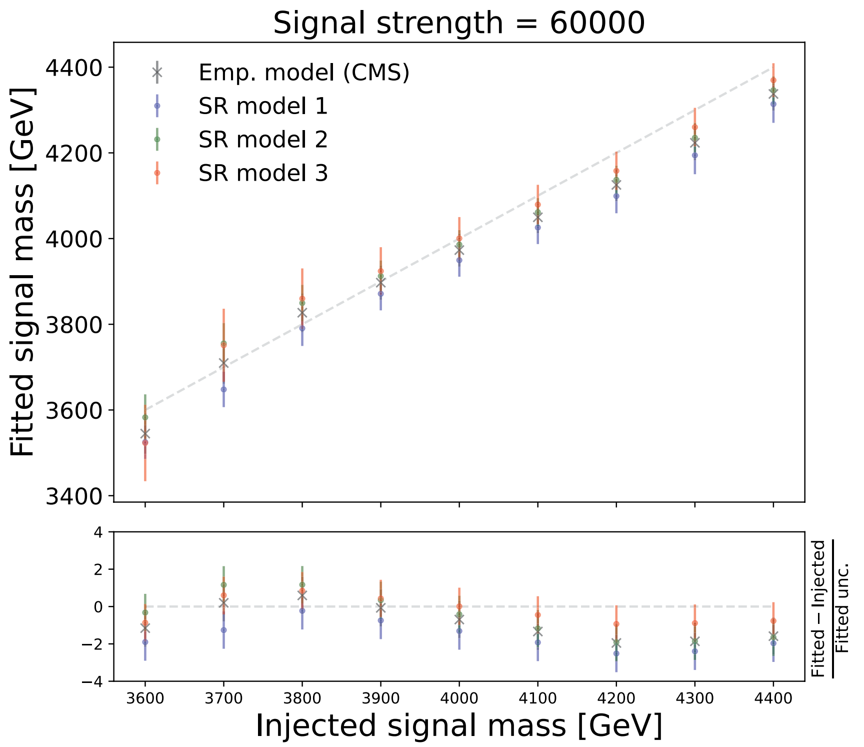
Fig. 26 Comparison of the b-only fits and the s+b fits to the unblinded pseudodata of the trijet spectrum. The lower panel shows the residual error per bin, measured in units of the data uncertainty. The shape of the injected signal is also shown

Table 9 Comparison of the χ^2/NDF scores from three types of fits to the trijet dataset: the b-only fits to the blinded pseudodata, b-only fits to the unblinded pseudodata, and s+b fits to the unblinded pseudodata

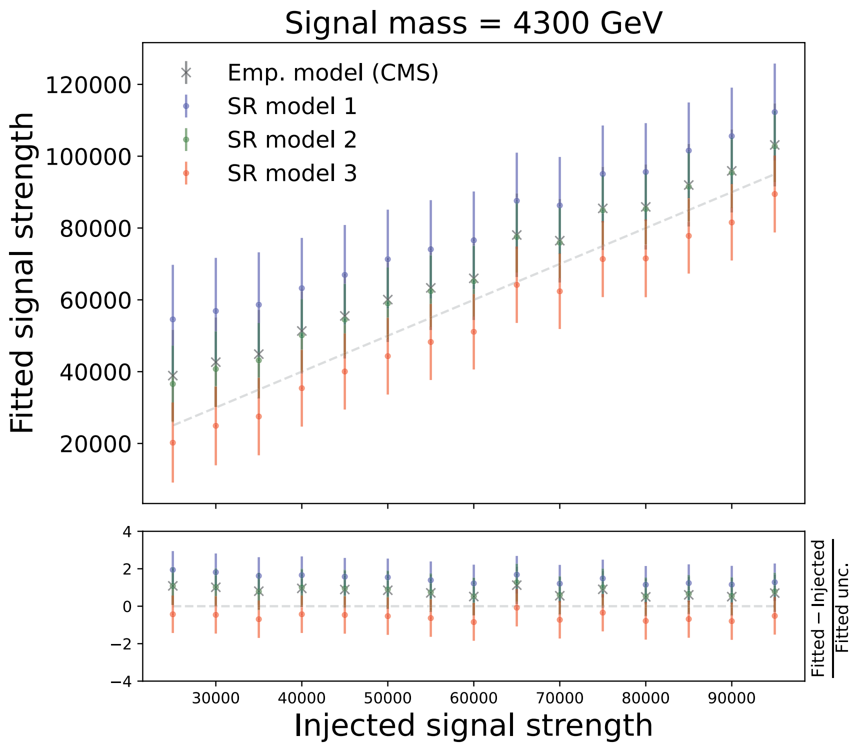
	χ^2/NDF (b-only, blinded)	χ^2/NDF (b-only, unblinded)	χ^2/NDF (s+b, unblinded)
SR model 1	49.04 / 26 = 1.886	181.3 / 42 = 4.317	93.04 / 40 = 2.326
SR model 2	33.15 / 25 = 1.326	92.92 / 41 = 2.266	48.07 / 39 = 1.233
SR model 3	37.31 / 25 = 1.492	74.5 / 41 = 1.817	54.23 / 39 = 1.391
Emp. model (CMS)	41.75 / 25 = 1.67	117.6 / 41 = 2.868	65.69 / 39 = 1.684

The background models used for the fits are listed in Table 8, and the fits are shown in Fig. 25 (blinded) and Fig. 26 (unblinded)

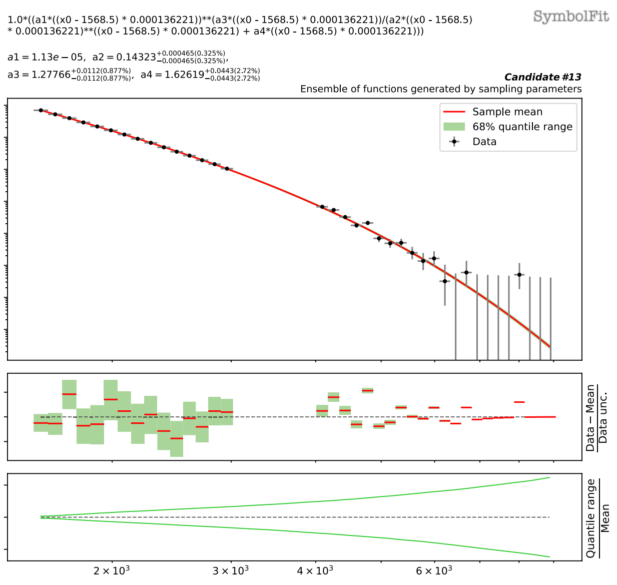
Fig. 27 Fitted values vs. the true values of parameters of the injected signal in the trijet dataset. The bottom panels show the residual error in units of the fitted uncertainty



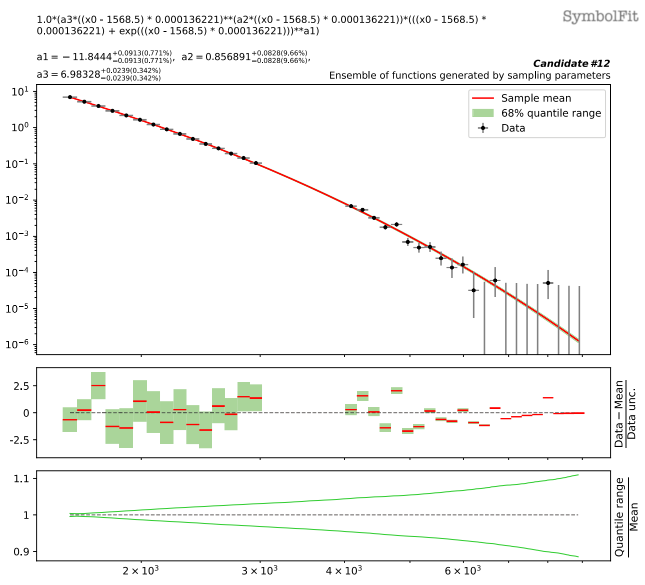
(a) Fitted vs. injected signal mass at a specified signal strength value.



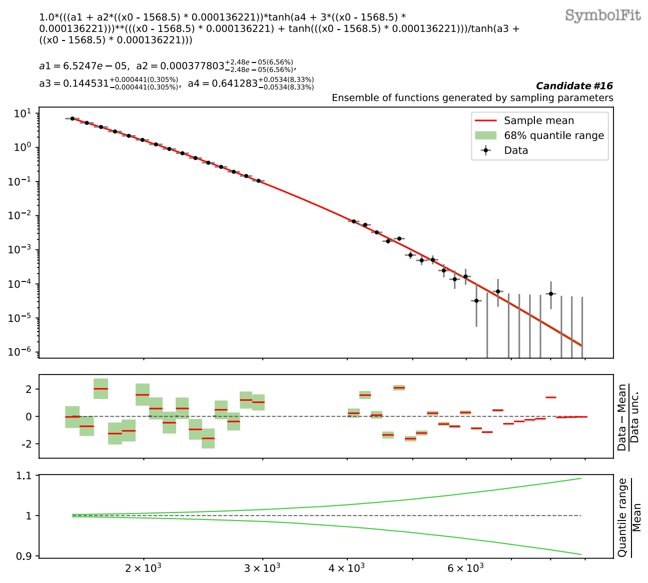
(b) Fitted vs. injected signal strength at a specified signal mass value.



(a) SR model 1.



(b) SR model 2.



(c) SR model 3.

Fig. 28 The three SR models fitted to the pseudodata of the paired-dijet spectrum with the signal region blinded (see Table 10). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

signal region is statistically limited, simpler functions are preferred to avoid over-fitting the background. For our comparison, we take the first-order Bernstein polynomial as the empirical model used by CMS.

We perform the same experiments conducted on the dijet dataset, as detailed in Sec. "CMS dijet dataset (1D) [background modeling]". Starting from the original dimuon spectrum, we generate pseudodata by injecting a Gaussian

Table 10 The candidate functions are obtained from three fits using different random seeds, fitted to the pseudodata of the four-jet spectrum with the (injected) signal region blinded

	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	*p*-value (after ROF)
SR model 1	$(1.13 \times 10^{-5}x)^{1.28x}/(0.143x^{x} + 1.63x)$	3	47.95 / 34 = 1.41	39.07 / 34 = 1.149	0.2524
SR model 2	$6.98x^{0.857x}(x + \exp(x))^{-11.8}$	3	47.36 / 34 = 1.393	39.83 / 34 = 1.171	0.2267
SR model 3	$((6.52 \times 10^{-5} + 0.000378x)\tanh(0.641 + 3x))^{x+\tanh(x)} / \tanh(0.145 + x)$	3	71.24 / 34 = 2.095	35.57 / 34 = 1.046	0.3942

The fits were performed on a scaled dataset (to enhance fit stability and prevent numerical overflow), and the functions can be transformed back to describe the original spectrum using the transformation: $x \rightarrow 0.000136(x - 1568.5)$. These functions are plotted and compared with the blinded pseudodata in Fig. 29. Numerical values are rounded to three significant figures for display purposes

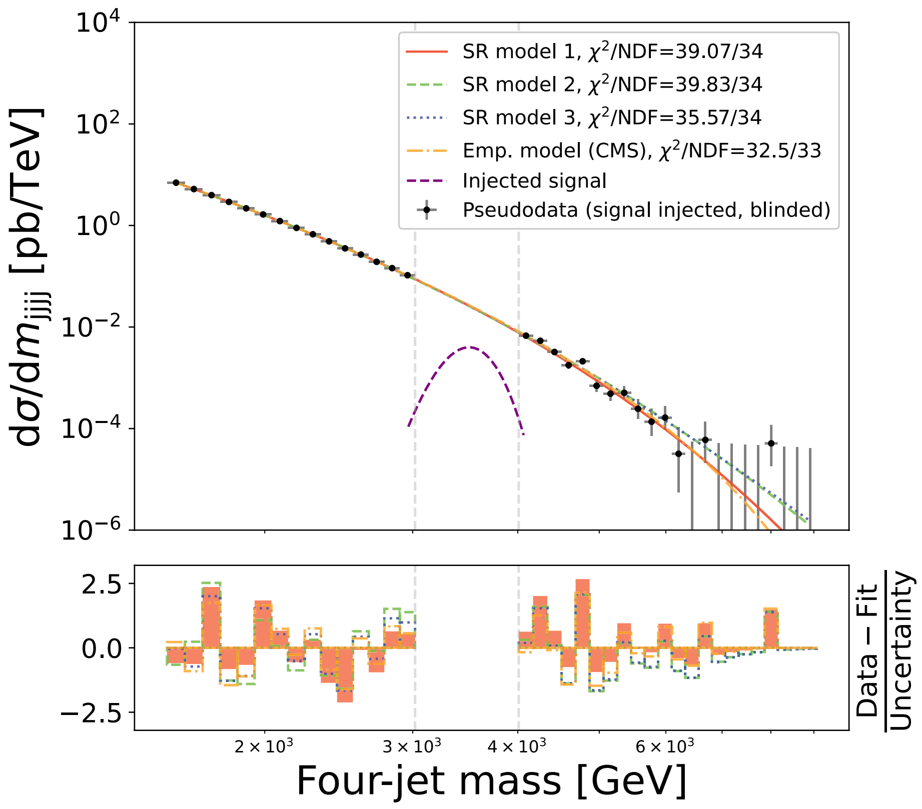
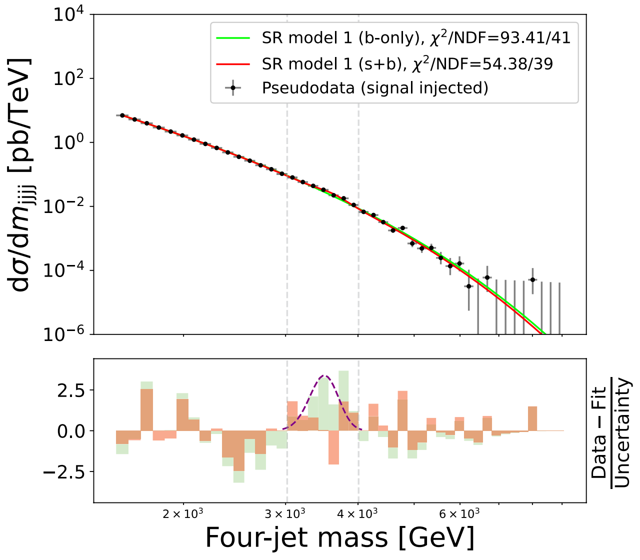


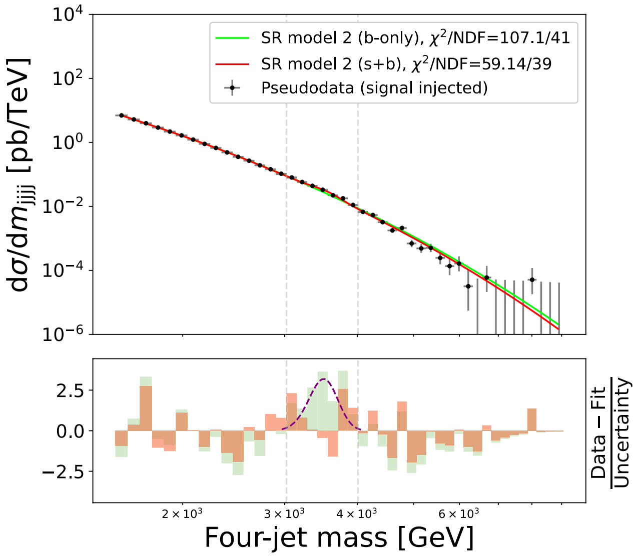
Fig. 29 Pseudodata of the four-jet spectrum with the injected signal shown in the blinded signal region. The three SR models (see Table 10) are compared against the empirical model used by CMS. The lower panel shows the residual error per bin, measured in units of the data uncertainty. It can be seen that the three SR models, generated easily from three separate fits using the same simple fit configuration with different random seeds, yield results that are readily comparable to the CMS empirical model that would have required extensive manual effort to obtain

signal centered at $m_{\mu\mu} = 500$ GeV (s_1), with a width of 20 GeV ($2s_2$) and a signal strength of $s_0 = 350$. To model the background, we blind the signal region by masking the $m_{\mu\mu}$ bins between 450 and 550 GeV in the pseudodata and perform the fits.

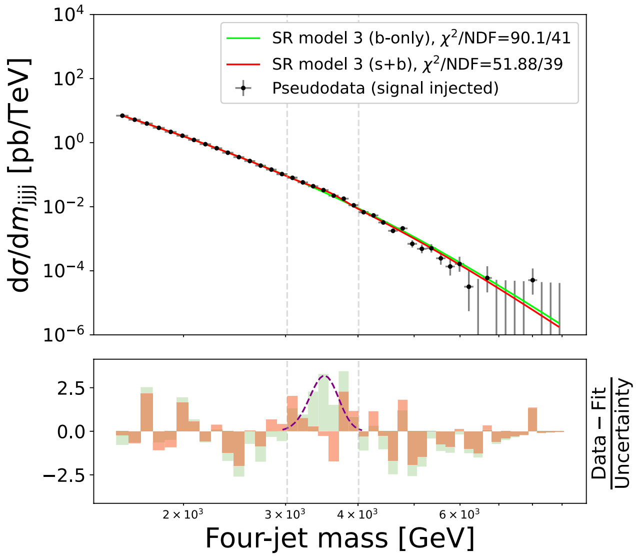
Three `SymbolFit` runs using different random seeds are carried out, applying the same `PySR` configuration as used for the dijet dataset (see List. 2), except that the maximum complexity is set at 20 instead of 80, since the dimuon distribution shape is less complex. Table 12 lists the three SR models, each obtained from a fit initialized with a different random seed. The χ^2/NDF scores improve significantly after the ROF step compared to the original functions returned by `PySR`. The three background models fit the blinded pseudodata well, as shown in Fig. 32 for the total



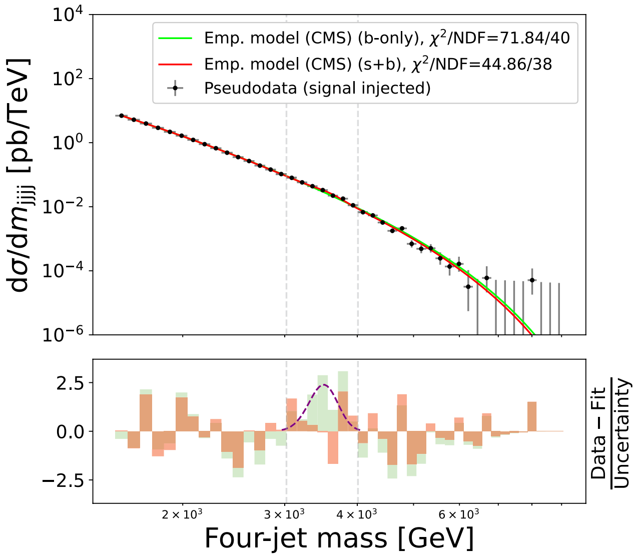
(a) SR model 1.



(b) SR model 2.



(c) SR model 3.



(d) Empirical model (CMS).

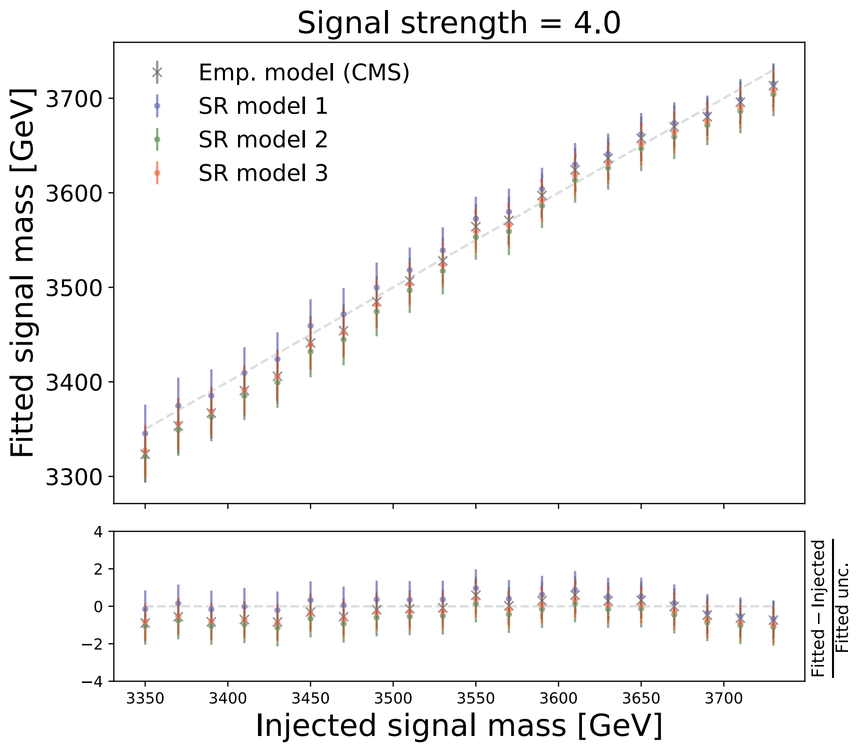
Fig. 30 Comparison of the b-only fits and the s+b fits to the unblinded pseudodata of the four-jet spectrum. The lower panel shows the residual error per bin, measured in units of the data uncertainty. The shape of the injected signal is also shown

Table 11 Comparison of the χ^2/NDF scores from three types of fits to the paired-dijet dataset: the b-only fits to the blinded pseudodata, b-only fits to the unblinded pseudodata, and s+b fits to the unblinded pseudodata

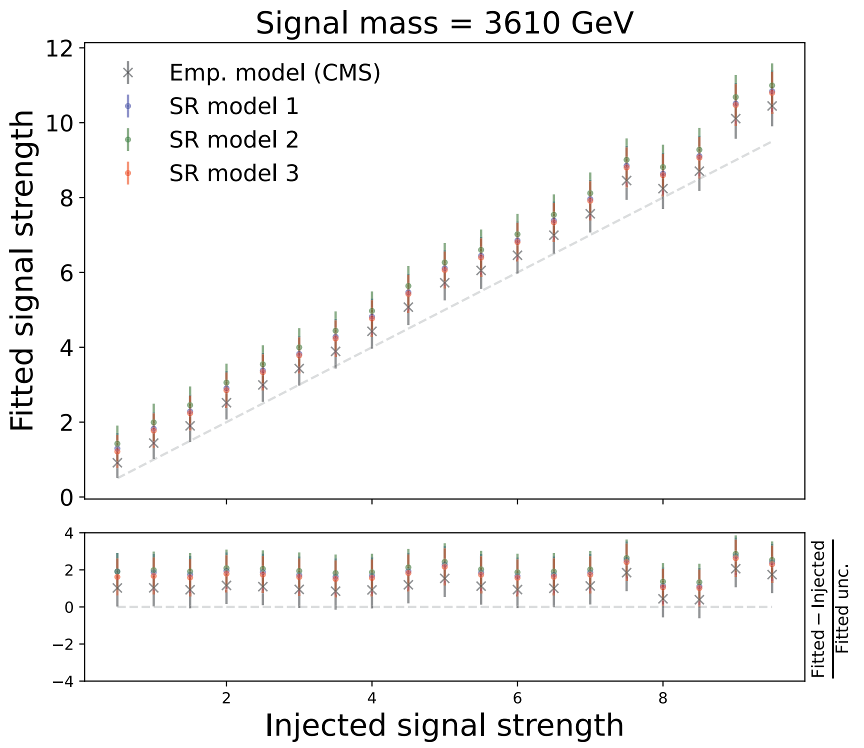
	χ^2/NDF (b-only, blinded)	χ^2/NDF (b-only, unblinded)	χ^2/NDF (s+b, unblinded)
SR model 1	39.07 / 34 = 1.149	93.41 / 41 = 2.278	54.38 / 39 = 1.394
SR model 2	39.83 / 34 = 1.171	107.1 / 41 = 2.612	59.14 / 39 = 1.516
SR model 3	35.57 / 34 = 1.046	90.1 / 41 = 2.198	51.88 / 39 = 1.33
Emp. model (CMS)	32.5 / 33 = 0.985	71.84 / 40 = 1.796	44.86 / 38 = 1.181

The background models used for the fits are listed in Table 10, and the fits are shown in Fig. 29 (blinded) and Fig. 30 (unblinded)

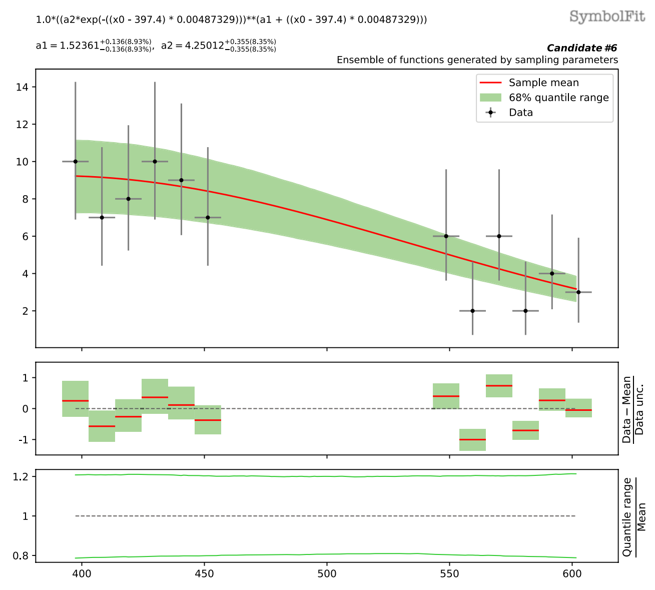
Fig. 31 Fitted values vs. the true values of parameters of the injected signal in the paired-dijet dataset. The bottom panels show the residual error in units of the fitted uncertainty



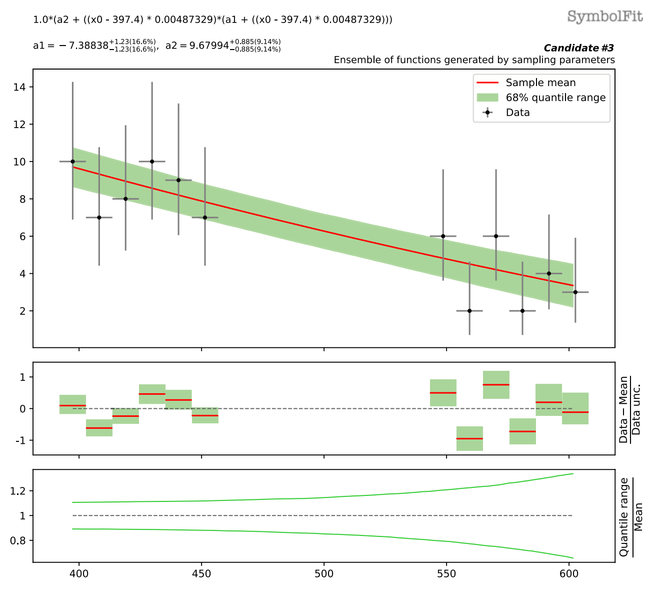
(a) Fitted vs. injected signal mass at a specified signal strength value.



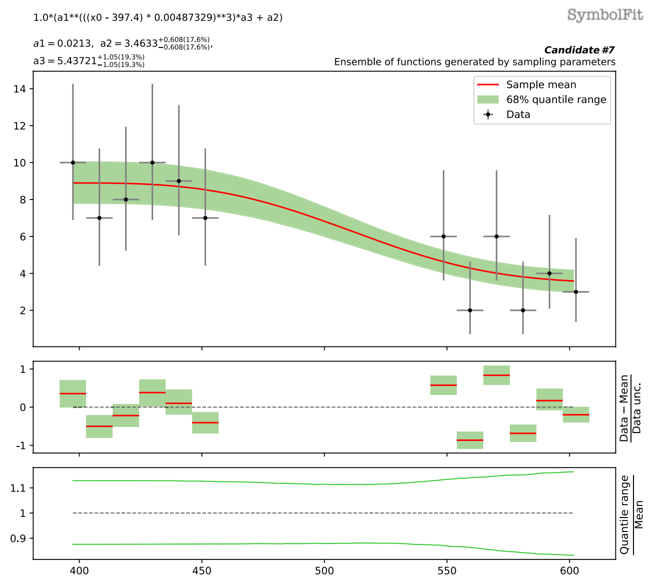
(b) Fitted vs. injected signal strength at a specified signal mass value.



(a) SR model 1.



(b) SR model 2.



(c) SR model 3.

Fig. 32 The three SR models fitted to the pseudodata of the dimuon spectrum with the signal region blinded (see Table 12). To visualize the total uncertainty coverage of each candidate function, the green band in each subfigure represents the 68% quantile range of functions obtained by sampling parameters, taking into account the best-fit values and the covariance matrix within a multidimensional normal distribution. The red line denotes the mean of the function ensemble. At the top of each subfigure, the candidate function and the fitted parameters are shown. The middle panel shows the weighted residual error: $\frac{\text{Data}-\text{Mean}}{\text{Data unc.}}$. The bottom panel shows the ratio of the 68% quantile range to the mean

Table 12 The candidate functions are obtained from three fits using different random seeds, fitted to the pseudodata of the dimuon spectrum with the (injected) signal region blinded

	Candidate function (after ROF)	# param.	χ^2/NDF (before ROF)	χ^2/NDF (after ROF)	p-value (after ROF)
SR model 1	$(4.25\exp(-x))^{1.52+x}$	2	3.469 / 10 = 0.3469	3.007 / 10 = 0.3007	0.9813
SR model 2	$9.68 + x(-7.39 + x)$	2	3.484 / 10 = 0.3484	3.075 / 10 = 0.3075	0.9796
SR model 3	$0.0213^{x^3}5.44 + 3.46$	2	3.456 / 10 = 0.3456	3.066 / 10 = 0.3066	0.9798

The fits were performed on a scaled dataset (to enhance fit stability and prevent numerical overflow), and the functions can be transformed back to describe the original spectrum using the transformation: $x \to 0.00487(x - 397.4)$. These functions are plotted and compared with the blinded pseudodata in Fig. 33. Numerical values are rounded to three significant figures for display purposes

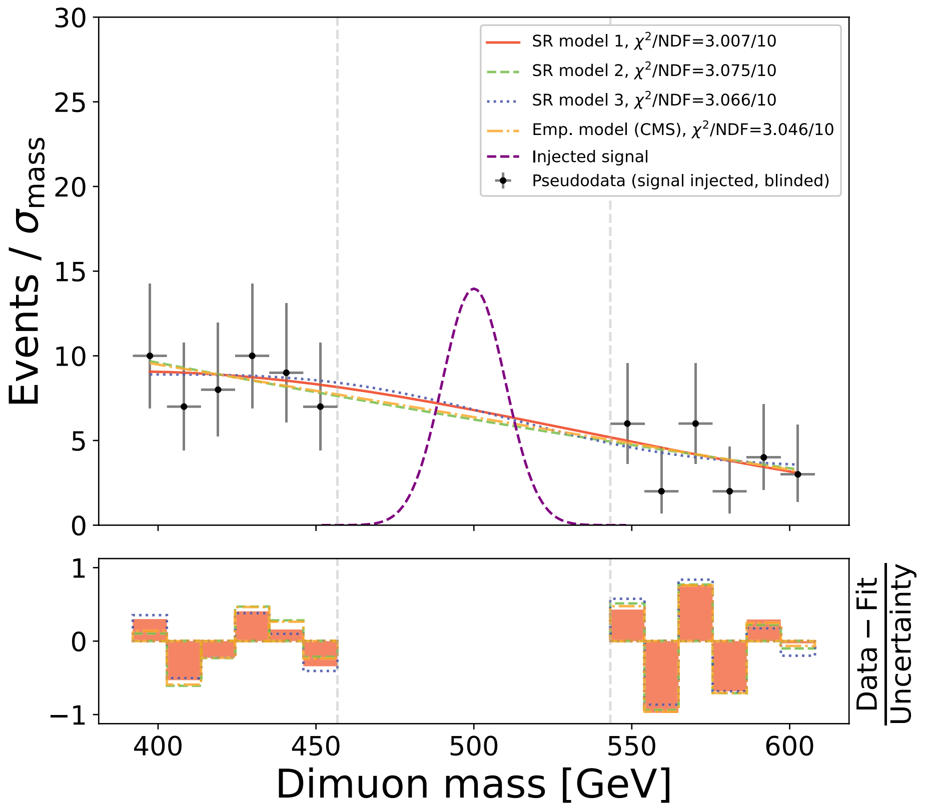
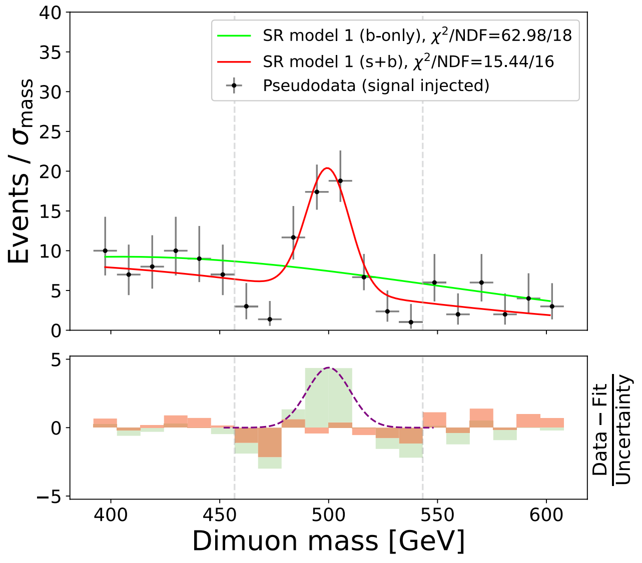


Fig. 33 Pseudodata of the dimuon spectrum with the injected signal shown in the blinded signal region. The three SR models (see Table 12) are compared against the empirical model used by CMS. The lower panel shows the residual error per bin, measured in units of the data uncertainty

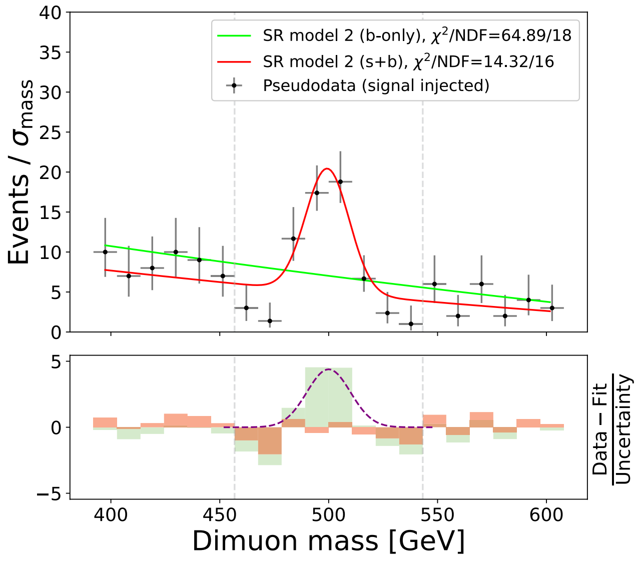
uncertainty coverage and Fig. 33 for a comparison with the empirical model used by CMS.

Next, we unblind the pseudodata and perform b-only fits and s+b fits on the full pseudodata spectrum. These results are shown in Fig. 34. In all three SR models, as well as the CMS empirical model, the excess of events over the background around the injected signal location observed in the b-only fits is reduced in the s+b fits, demonstrating that the models are sensitive to the injected signal. Table 13 lists the χ^2/NDF scores for each model, showing the fit performance in response to the presence of the injected signal.

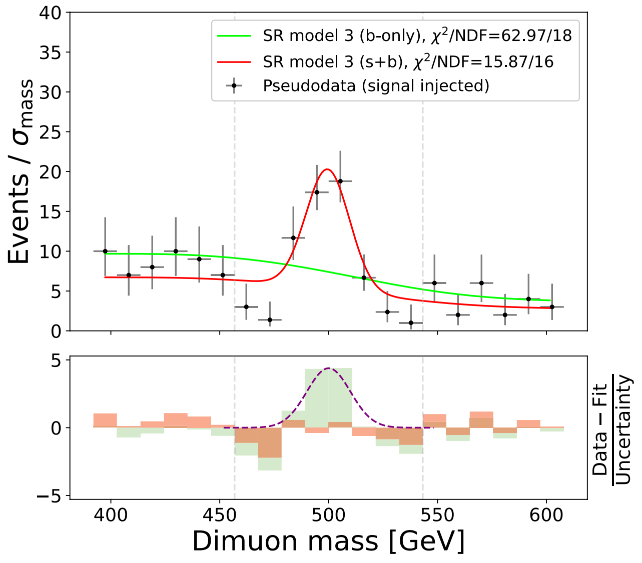
To assess whether the SR models can accurately extract the injected signals, we generate multiple sets of pseudodata by injecting Gaussian signals with different mean values ranging from 490 to 510 GeV and varying signal strength between 350 and 600. We then perform the s+b fits to extract the corresponding signal parameters. Figure 35 shows the extracted signal parameters plotted against their injected values. All three SR models are capable of extracting the correct signal parameter values within reasonable uncertainties and are comparable to the empirical model used by CMS.



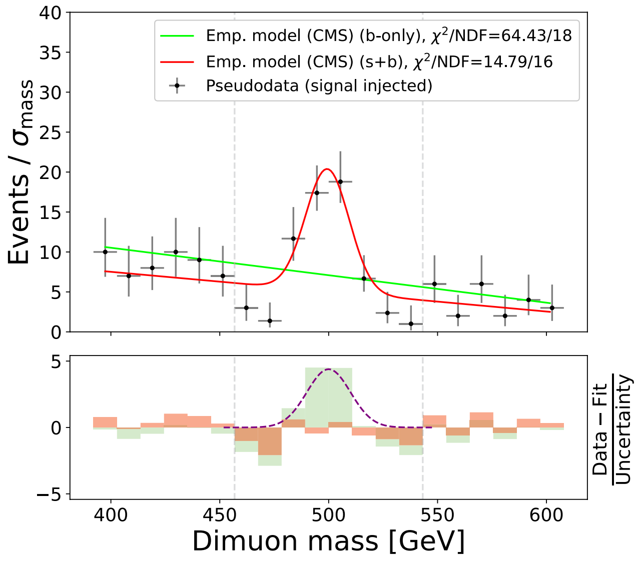
(a) SR model 1.



(b) SR model 2.



(c) SR model 3.



(d) Empirical model (CMS).

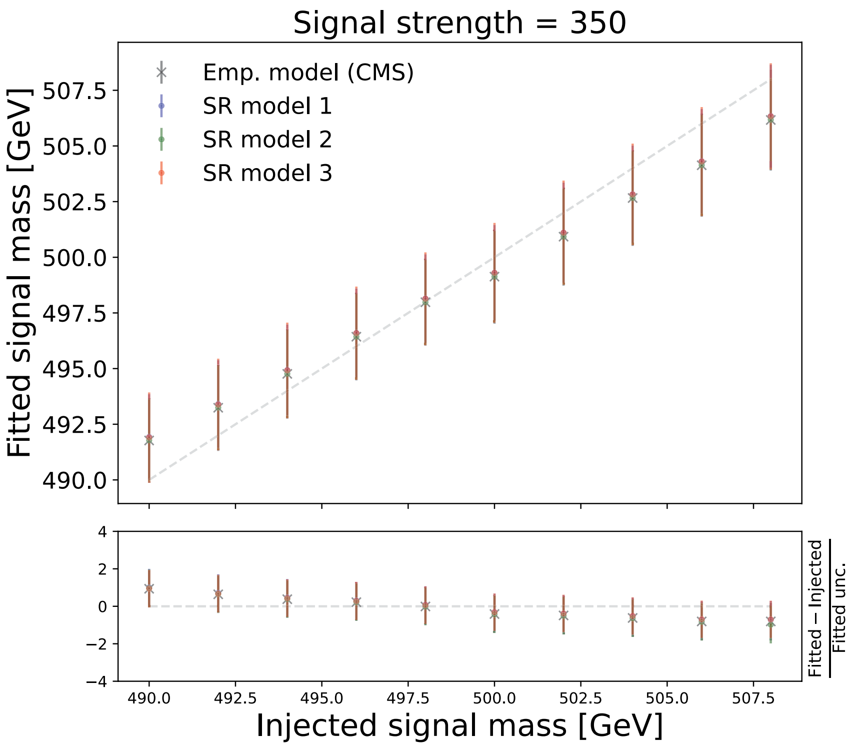
Fig. 34 Comparison of the b-only fits and the s+b fits to the unblinded pseudodata of the dimuon spectrum. The lower panel shows the residual error per bin, measured in units of the data uncertainty. The shape of the injected signal is also shown

Table 13 Comparison of the χ^2/NDF scores from three types of fits to the dimuon dataset: the b-only fits to the blinded pseudodata, b-only fits to the unblinded pseudodata, and s+b fits to the unblinded pseudodata

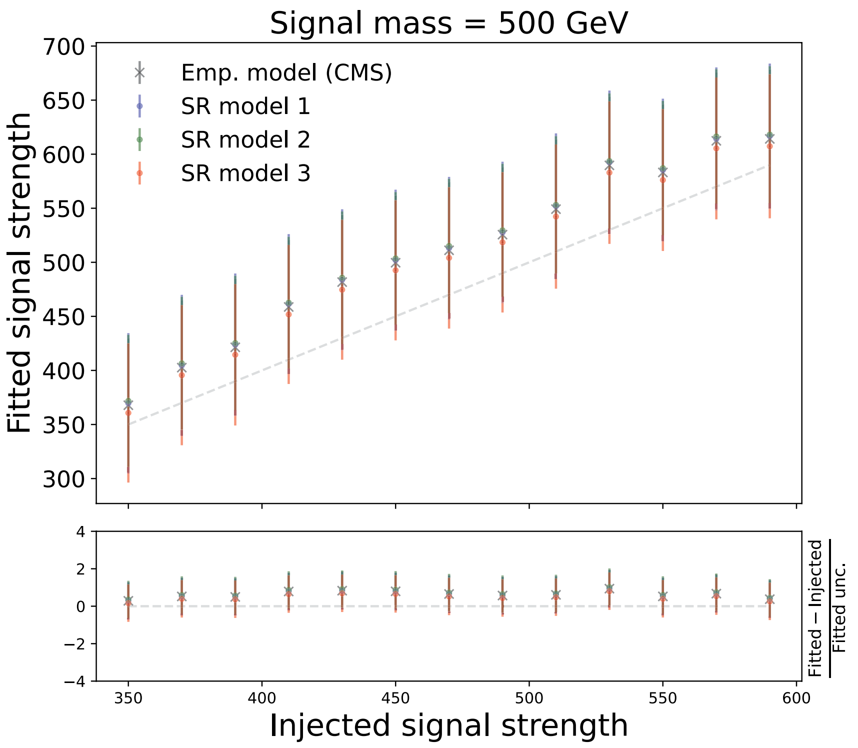
	χ^2/NDF (b-only, blinded)	χ^2/NDF (b-only, unblinded)	χ^2/NDF (s+b, unblinded)
SR model 1	3.007 / 10 = 0.3007	62.98 / 18 = 3.499	15.44 / 16 = 0.965
SR model 2	3.075 / 10 = 0.3075	64.89 / 18 = 3.605	14.32 / 16 = 0.895
SR model 3	3.066 / 10 = 0.3066	62.97 / 18 = 3.498	15.87 / 16 = 0.9919
Emp. model (CMS)	3.046 / 10 = 0.3046	64.43 / 18 = 3.579	14.79 / 16 = 0.9244

The background models used for the fits are listed in Table 12, and the fits are shown in Fig. 33 (blinded) and Fig. 34 (unblinded)

Fig. 35 Fitted values vs. the true values of parameters of the injected signal in the dimuon dataset. The bottom panels show the residual error in units of the fitted uncertainty



(a) Fitted vs. injected signal mass at a specified signal strength.



(b) Fitted vs. injected signal strength at a specified signal mass value.

Acknowledgements DR is supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0025324. SD is supported by the U.S. DOE, Office of Science, Office of High Energy Physics, under Award No. DE-SC0017647. JD is supported by the Research Corporation for Science Advancement (RCSA) under grant #CS-CSA-2023-109, Alfred P. Sloan Foundation under grant #FG-2023-20452, U.S. DOE, Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187. JD, PH, and VL are supported by the U.S. National Science Foundation (NSF) Harnessing the Data Revolution (HDR) Institute for Accelerating AI Algorithms for Data Driven Discovery (A3D3) under Cooperative Agreement PHY-2117997. PH is also supported by the Institute for Artificial Intelligence and Fundamental Interactions (IAIFI), under the NSF grant #PHY-2019786. EL is supported by the U.S. DOE, Office of Science, Office of High Energy Physics, under Award No. DE-SC0007901.

Funding Open access funding provided by CERN (European Organization for Nuclear Research).

References

1. La Cava W (2021) *et al.* Contemporary symbolic regression methods and their relative performance. In Vanschoren, J. & Yeung, S. (eds.) *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1. https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/c0c7c76d30bd3dcaefc96f40275bdc0a-Paper-round1.pdf
2. Udrescu S-M, Tegmark M (2020) Ai feynman: a physics-inspired method for symbolic regression. Sci Adv 6:2631
3. Keren LS, Liberzon A, Lazebnik T (2023) A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge. Sci Rep. https://doi.org/10.1038/s41598-023-28328-2
4. Cava WL (2021) *et al.* Contemporary symbolic regression methods and their relative performance. https://arxiv.org/abs/2107.14351. arXiv:2107.14351
5. Cranmer M (2023) Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. arXiv:2305.01582
6. Tsoi HF, Loncar V, Dasu S, Harris P (2025) SymbolNet: neural symbolic regression with adaptive dynamic pruning for compression. Mach Learn Sci Tech 6:015021 arXiv:2401.09949
7. Davis BL, Jin Z (2023) Discovery of a planar black hole mass scaling relation for spiral galaxies. Astrophys J Lett 956:L22. https://doi.org/10.3847/2041-8213/acfa98
8. Mengel T, Steffanic P, Hughes C, da Silva ACO, Nattrass C (2023) Interpretable machine learning methods applied to jet background subtraction in heavy-ion collisions. Phys Rev C 108:L021901. https://doi.org/10.1103/PhysRevC.108.L021901
9. Wadekar D et al (2023) The sz flux-mass (y-m) relation at low-halo masses: improvements with symbolic regression and strong constraints on baryonic feedback. Monthly Notices Royal Astron Soc 522:2628–2643. https://doi.org/10.1093/mnras/stad1128
10. Lemos P, Jeffrey N, Cranmer M, Ho S, Battaglia P (2023) Rediscovering orbital mechanics with machine learning. Mach Learn Sci Tech 4:045002 arXiv:2202.02306
11. Delgado AM et al (2022) Modelling the galaxy-halo connection with machine learning. Mon Not Roy Astron Soc 515:2733–2746 arXiv:2111.02422
12. Tsoi HF et al (2024) Symbolic regression on FPGAs for fast machine learning inference. EPJ Web Conf 295:09036 arXiv:2305.04099
13. Butter A, Plehn T, Soybelman N, Brehmer J (2024) Back to the formula - LHC edition. Sci Post Phys 16:037 arXiv:2109.10414
14. Shao H et al (2022) Finding universal relations in subhalo properties with artificial intelligence. Astrophys. J. 927:85 arXiv:2109.04484
15. Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. Science 324:81–85. https://doi.org/10.1126/science.1165893
16. Stephens T (2016) Genetic programming in Python, with a scikit-learn inspired API: gplearn. https://gplearn.readthedocs.io/en/stable/
17. Burlacu B, Kronberger G & Kommenda M (2020) Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, 1562–1570 (publisherAssociation for Computing Machinery, addressNew York, NY, USA, 2020). https://doi.org/10.1145/3377929.3398099
18. Virgolin M, Alderliesten T, Witteveen C, Bosman PAN (2021) Improving model-based genetic programming for symbolic regression of small expressions. Evol Comput 29:211–237
19. Koza J (1994) Genetic programming as a means for programming computers by natural selection. Statist Comput 4:87–112
20. Hayrapetyan A et al (2024) Search for narrow trijet resonances in proton-proton collisions at s=13 TeV. Phys Rev Lett 133:011801 arXiv:2310.14023
21. Fisher RA (1922) On the interpretation of chi^2 from contingency tables, and the calculation of p. J Royal Statis Soc 85:87–94 (http://www.jstor.org/stable/2340521)
22. Aad G et al (2012) Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC. Phys Lett B 716:1–29 arXiv:1207.7214
23. Chatrchyan S et al (2012) Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. Phys Lett B 716:30–61 arXiv:1207.7235
24. Chatrchyan S et al (2013) Observation of a new boson with mass near 125 GeV in *pp* Collisions at $\sqrt{s} = 7$ and 8 TeV. JHEP 06:081 arXiv:1303.4571
25. Sirunyan AM et al (2020) Search for high mass dijet resonances with a new background prediction method in proton-proton collisions at $\sqrt{s} = 13$ TeV. JHEP 05:033 arXiv:1911.03947
26. Tumasyan A et al (2023) Search for resonant and nonresonant production of pairs of dijet resonances in proton-proton collisions at $\sqrt{s} = 13$ TeV. JHEP 07:161 arXiv:2206.09997
27. Hayrapetyan A et al (2024) Search for new physics in high-mass diphoton events from proton-proton collisions at $\sqrt{s} = 13$ TeV. JHEP 08:215 arXiv:2405.09320

28. Hayrapetyan A et al (2023) Search for a high-mass dimuon resonance produced in association with b quark jets at $\sqrt{s} = 13$ TeV. JHEP 10:043 arXiv:2307.08708
29. Khachatryan V et al (2017) Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV. JINST 12:P02014 arXiv:1607.03663
30. Sirunyan AM et al (2018) Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. JINST 13:P05011 arXiv:1712.07158
31. Sirunyan AM et al (2018) Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV. JINST 13:P10005 arXiv:1809.02816
32. Sirunyan AM et al (2020) Inclusive search for highly boosted Higgs bosons decaying to bottom quark-antiquark pairs in proton-proton collisions at $\sqrt{s} = 13$ TeV. JHEP 12:085 arXiv:2006.13251
33. Frate M, Cranmer K, Kalia S, Vandenberg-Rodes A, Whiteson D (2017) Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes 1709:05681
34. Gandrakota A, Lath A, Morozov AV, Murthy S (2023) Model selection and signal extraction using Gaussian Process regression. JHEP 02:230 arXiv:2202.05856
35. Xu R (2024) A Measurement of Boosted Dibosons with Gaussian Process Background Modeling at the ATLAS Detector. https://cds.cern.ch/record/2901208. Presented 21 May 2024
36. Swiler LP, Gulian M, Frankel AL, Safta C, Jakeman JD (2020) A survey of constrained gaussian process regression: approaches and implementation challenges. J Machine Learn Model Comput 1:119–156. https://doi.org/10.1615/JMachLearnModelComput.2020035155
37. Newville M, Stensitzki T, Allen DB & Ingargiola A (2015) LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python. https://doi.org/10.5281/zenodo.11813
38. Hayrapetyan A et al (2024) The CMS Statistical Analysis and Combination Tool: COMBINE 2404:06614
39. Heinrich L, Feickert M, Stark G. pyhf: v0.7.6. https://doi.org/10.5281/zenodo.1169739. Https://github.com/scikit-hep/pyhf/releases/tag/v0.7.6
40. Heinrich L, Feickert M, Stark G, Cranmer K (2021) pyhf: pure-python implementation of histfactory statistical models. J Open Source Softw 6:2823. https://doi.org/10.21105/joss.02823
41. Dauncey PD, Kenzie M, Wardle N, Davies GJ (2015) Handling uncertainties in background shapes: the discrete profiling method. JINST 10:P04015 arXiv:1408.6865
42. CMS Collaboration. Search for high mass dijet resonances with a new background prediction method in proton-proton collisions at $\sqrt{s} = 13$ TeV. HEPData (collection) (2019). https://doi.org/10.17182/hepdata.91059
43. CMS Collaboration. Search for new physics in high-mass diphoton events from proton-proton collisions at $\sqrt{s} = 13$ TeV. HEPData (collection) (2024). https://doi.org/10.17182/hepdata.150677
44. CMS Collaboration. Search for narrow trijet resonances in proton-proton collisions at $\sqrt{s} = 13$ TeV. HEPData (collection) (2023). https://doi.org/10.17182/hepdata.144165
45. CMS Collaboration. Search for resonant and nonresonant production of pairs of dijet resonances in proton-proton collisions at $\sqrt{s} = 13$ TeV. HEPData (collection) (2022). https://doi.org/10.17182/hepdata.130817
46. CMS Collaboration. Search for a high-mass dimuon resonance produced in association with b quark jets at $\sqrt{s}$=13 TeV. HEPData (collection) (2023). https://doi.org/10.17182/hepdata.141455