

BROOD: Bilevel and Robust Optimization and Outlier Detection for Efficient Tuning of High-Energy Physics Event Generators

Wenjing Wang¹, Mohan Krishnamoorthy², Juliane Müller^{1*}, Stephen Mrenna^{3*}, Holger Schulz⁴,
Xiangyang Ju¹, Sven Leyffer², Zachary Marshall¹

1 Lawrence Berkeley National Laboratory, Berkeley, CA 94720

2 Argonne National Laboratory, Lemont, IL 60439

3 Fermi National Accelerator Laboratory, Batavia, IL 60510

4 Department of Computer Science, Durham University, South Road, Durham DH1 3LE, UK

* julianemueller@lbl.gov mrenna@fnal.gov

March 3, 2021

1 Abstract

The parameters in Monte Carlo (MC) event generators are tuned on experimental measurements by evaluating the goodness of fit between the data and the MC predictions. The relative importance of each measurement is adjusted manually in an often time-consuming, iterative process to meet different experimental needs. In this work, we introduce several optimization formulations and algorithms with new decision criteria for streamlining and automating this process. These algorithms are designed for two formulations: bilevel optimization and robust optimization. Both formulations are applied to the datasets used in the ATLAS A14 tune and to the dedicated hadronization datasets generated by the SHERPA generator, respectively. The corresponding tuned generator parameters are compared using three metrics. We compare the quality of our automatic tunes to the published ATLAS A14 tune. Moreover, we analyze the impact of a pre-processing step that excludes data that cannot be described by the physics models used in the MC event generators.

13

14 Contents

15	1 Introduction and Motivation	4
16	1.1 Notation and terminology	5
17	1.2 Mathematical formulation of the tuning problem	5

18	2 Finding the Optimal Weights for Each Observable	8
19	2.1 Bilevel optimization formulation	9
20	2.1.1 Formulation 1: Portfolio to balance mean and variance of errors	9
21	2.1.2 Formulation 2: Scoring of model fit and data uncertainty	10
22	2.1.3 Solving the bilevel optimization problem using surrogate models	11
23	2.2 Robust optimization formulation	12
24	3 Data Pre-processing: Filtering Observables or Bins	13
25	3.1 Filtering of observables by outlier detection	14
26	3.2 Filtering of bins by hypothesis testing	15
27	4 Numerical Experiments and Comparison of Different Tunes	16
28	4.1 Setup of the numerical experiments	16
29	4.2 Comparison metrics and optimal tuning parameters	17
30	4.3 The A14 dataset	19
31	4.4 The SHERPA dataset	21
32	4.5 Data pre-processing: filtering out observables and bins	22
33	4.6 Results for the A14 dataset	23
34	4.6.1 Comparison metric outcomes for the A14 dataset	23
35	4.6.2 Comparison of the cumulative distribution of bins at different variance levels	25
36	4.6.3 Optimal parameter values for the A14 dataset with rational approximation .	27
37	4.6.4 Comparison of optimal weights for the A14 dataset with rational approximation	29
38	4.6.5 Impact of data pre-processing by filtering on optimal results	30
39	4.6.6 Comparison of rational approximation and the MC simulator	30
40	4.7 Results for the SHERPA dataset	34
41	4.7.1 Comparison metric outcomes for the SHERPA dataset	35
42	4.7.2 Comparison of the cumulative distribution of bins at different variance levels	36
43	4.7.3 Optimal parameter values for the SHERPA dataset with rational approximation	36
44	4.8 A note on computation times	38
45	5 Eigentunes	40
46	6 Discussion	43
47	6.1 Implications of our results on physics	43
48	6.2 Observables with improved descriptions	44
49	6.3 Observables with worse descriptions	46

50	6.4 Results for SHERPA tuning	46
51	7 Conclusions	50
52	References	51
53	8 Online Supplement	55
54	8.1 Solving the outer problem with derivative-free surrogate optimization	55
55	8.2 Polynomial-time algorithm for filtering bins by hypothesis testing	59
56	8.3 A14 and SHERPA physics parameters	61
57	8.4 Selection of the best hyperparameter in robust optimization	62
58	8.5 Outlier observables in the A14 dataset	62
59	8.6 Outlier observables in the SHERPA dataset	62
60	8.7 Bin filtered data for A14 dataset	63
61	8.8 Bin filtered data for SHERPA dataset	63
62	8.9 Complete results from filtering out observables and bins	63
63	8.10 Comparison of the rational approximation with the MC generator	63
64	8.11 Optimal parameter values for the A14 dataset with the rational approximation	65
65	8.12 Results for using the cubic polynomial to approximate the MC simulation	65
66	8.12.1 Comparison metric outcomes for the A14 dataset using the cubic polynomial	
67	approximation	66
68	8.12.2 Optimal parameter values for the A14 dataset using the cubic polynomial	
69	approximation	66
70	8.12.3 Comparison of optimal weights for the A14 dataset with cubic polynomial	
71	approximation	70
72	8.12.4 Optimal parameter values for the SHERPA dataset with rational approximation	70
73	8.12.5 Optimal parameter values for the SHERPA dataset with the cubic polynomial	
74	approximation	70
75	8.12.6 Comparison metric outcomes for the SHERPA dataset with the cubic poly-	
76	nomial approximation	70
77	8.13 Weights assigned by different fitting methods	71
78	8.14 Eigentunes for the results obtained with the cubic polynomial approximation	72
79	8.15 Generator settings for PYTHIA and SHERPA	72

80

81

1 Introduction and Motivation

Monte Carlo (MC) event generators are simulation tools that predict the properties of high-energy particle collisions. Event generators are built from theoretical formulae and models that describe the probabilities for various sub-event phenomena that occur in a high-energy collision. They are developed by physicists as a bridge between particle physics perturbation theory, which is defined at very high energy scales, and the observed sub-atomic particles, which are low-energy states of the strongly-interacting full theory. This bridge is essential for interpreting event collision data in terms of the fundamental quantities of the underlying theory. See [1] for an overview of the event generators used for physics analysis at the Large Hadron Collider (LHC).

The description of particle collisions requires an understanding of phenomena at many different energy scales. At high energy scales (much larger than the masses of the sub-atomic particles), first principle predictions can be made in a perturbative framework based on a few universal parameters. At intermediate energy scales, an approximate perturbation theory can be established that introduces less universal parameters. At low energy, motivated, but subjective, models are introduced to describe sub-atomic particle production. These low-energy models introduce a large number of narrowly defined parameters. To make predictions or inferences, one must have a handle on the preferred models and the values of the parameters needed to describe the data. This process of adjusting the parameters of the MC simulations to match data is called *tuning*.

This tuning task is complicated by the fact that the phenomenological models cannot claim to be complete or scale-invariant. When compared to a large set of collider data collected in different energy regimes, the MC-models do not describe the full range of event properties equally well. Typically, the physicists demand a tune that describes a subset of the data very well, another subset moderately well, and a remainder that must only be described qualitatively. This distribution of subsets may well vary from one group of physicists to another and has led to the education of experts who subjectively select and weigh data to achieve some physics goal. Two such exercises are the Monash tune [2] and the A14 tune [3], though others exist in the literature. Both of these tunes are successful, in the sense that they have been useful in understanding a wide range of phenomena observed at particle colliders. However, the current approach to tuning remains inefficient and biased.

This work introduces a framework that, once agreed upon, greatly reduces the subjective element of the tuning process and replaces it with an automated way to select the data for parameter tuning.

1.1 Notation and terminology

The data used in the tuning process are in the form of observables, denoted by \mathcal{O} , and the set of observables is denoted by $\mathcal{S}_{\mathcal{O}}$. Observables are quantities constructed from the (directly or indirectly) measured sub-atomic particles produced in an event. In this case, each observable is presented as a *histogram* that shows the frequency that the observable is measured over a range of possible values. The range can be one or many divisions of the interval from the minimum to the maximum value that the observable can obtain. These divisions are called *bins*. In practice, the size of a bin is set by how well an observable can be measured. The number of bins of an observable \mathcal{O} is denoted as $|\mathcal{O}|$. We use \mathcal{R} to denote the reference data in the histograms, a subscript b to denote a bin, \mathcal{R}_b to denote the data value in a bin, and $\Delta\mathcal{R}_b$ to denote the corresponding measurement uncertainty.

The MC-model has parameters \mathbf{p} , a d -dimensional vector in the space Ω , $\mathbf{p} \in \Omega \subset \mathbb{R}^d$. The MC-based simulations are denoted by $\text{MC}(\mathbf{p})$ to emphasize that they depend on the physics parameters \mathbf{p} . The histograms computed from the MC simulation have the same structure as the histograms obtained from the measurement data \mathcal{R} , with a prediction per bin $\text{MC}_b(\mathbf{p})$ and an uncertainty associated with each bin $\Delta\text{MC}_b(\mathbf{p})$. The uncertainty on the MC simulation comes from the numerical methods used to calculate the predictions, and it typically scales as the inverse of the square root of the number of simulated events in a particular bin. **Say explicitly that theory/model uncertainties are not included**

Figure 1 shows a typical histogram. In this example, the observable, *Thrust*, has 17 bins. In the top pane, the black segments show the experimental data \mathcal{R} . The vertical error bars show the uncertainty associated with the data, i.e., $\Delta\mathcal{R}$. The red line shows the data obtained from the MC simulation $\text{MC}(\mathbf{p})$ with some parameter setting \mathbf{p} . The bottom pane shows the ratio of $\text{MC}(\mathbf{p})$ to the data in each bin. The black horizontal line shows the reference ratio value one, to make the visual inspection easier. When the red line is above the black line, it means $\text{MC}(\mathbf{p}) > \mathcal{R}$, and vice versa. The yellow region is defined by the range of the uncertainty on a measured value (usually the 68% confidence level on the reported value) relative to the measured value, i.e. $[1 - \Delta\mathcal{R}_b/\mathcal{R}_b, 1 + \Delta\mathcal{R}_b/\mathcal{R}_b]$. A “good” tune is one where the red line falls within the yellow band. In the example Figure 1, $\text{MC}(\mathbf{p})$ underpredicts the number of events with intermediate values of *Thrust* and overpredicts near the endpoints.

1.2 Mathematical formulation of the tuning problem

Our goal is to find an optimal set of physics parameters, \mathbf{p}^* , that minimizes the difference between the experimental data and the simulated data from an MC event generator. This difference is

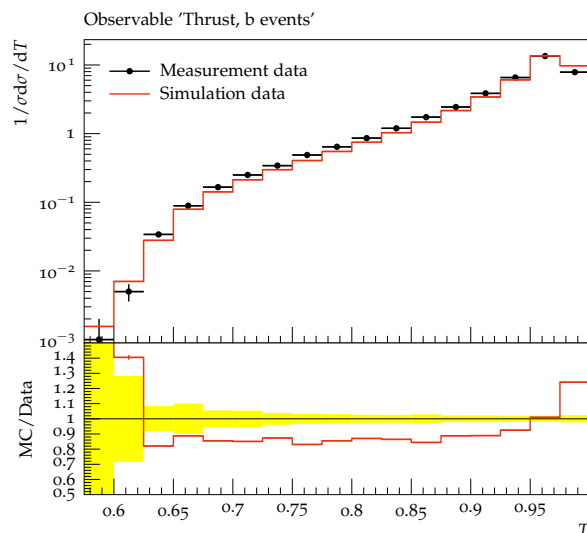


Figure 1: A histogram of a typical observable used in the tuning process. The top pane displays the measured (black) and predicted (red) data and their uncertainties. The bottom pane displays the ratio of predictions to measurements. The yellow band displays the measurement uncertainty on the reference data: $[1 - \Delta\mathcal{R}_b/\mathcal{R}_b, 1 + \Delta\mathcal{R}_b/\mathcal{R}_b]$.

Mention that observables are not some kind of basis for observed events, unlike phase space. So there will be correlations which are not considered here?

145 defined as follows:

$$\chi_{\text{MC}}^2(\mathbf{p}, \mathbf{w}) = \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}} \sum_{b \in \mathcal{O}} \frac{(\text{MC}(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta\text{MC}_b(\mathbf{p})^2 + \Delta\mathcal{R}_b^2}, \quad (1)$$

146 where $w_{\mathcal{O}}$ is the weight for an observable \mathcal{O} and \mathbf{w} is a vector of weights, $\mathbf{w} = [w_1, \dots, w_{|\mathcal{S}_{\mathcal{O}}|}]^T$. In
 147 general, the number of bins can be different for different observables. The weights $w_{\mathcal{O}} \geq 0$ reflect
 148 how much an observable contributes to the tune, i.e., if $w_{\mathcal{O}} = 0$ for some \mathcal{O} , then this observable
 149 will not influence the tuning of \mathbf{p} .

150 The MC simulation is computationally expensive (the generation of 1 million events for a given
 151 set of parameters consumes about 800 CPU minutes on a typical computing cluster), severely
 152 limiting the number of parameter choices \mathbf{p} that can be used in the tuning. To overcome these
 153 issues, we construct a parameterization of the MC simulation using an evolution of the PROFESSOR
 154 framework [4], named APPRENTICE. The code of APPRENTICE is available at <https://github.com/HEPonHPC/apprentice>. The function in Eq. (1) is not minimized directly. Instead, during the
 155 optimization over \mathbf{p} , the MC simulation is replaced by a surrogate model (here, a polynomial or a
 156 rational approximation to a number of MC simulations). For each bin b of each histogram, the central
 157 value and the corresponding uncertainty of the model prediction are parameterized independently as
 158

159 functions of the model parameters \mathbf{p} . This approach results in analytic expressions $f_b(\mathbf{p})$ and $\Delta f_b(\mathbf{p})$
 160 that approximate the central value and the uncertainty, respectively, and that can be evaluated in
 161 milliseconds. Thus, instead of Eq. (1), we minimize

$$\chi^2(\mathbf{p}, \mathbf{w}) = \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}} \sum_{b \in \mathcal{O}} \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}. \quad (2)$$

162 Because $f_b(\mathbf{p})$ and $\Delta f_b(\mathbf{p})$ are given analytically and provide a prediction for any choice of \mathbf{p}
 163 within our domain of interest, we can minimize Eq. (2) efficiently using numerical methods. Eq. (2)
 164 implicitly assumes that each bin b is completely independent of all other bins.

165 In practice, the weights $w_{\mathcal{O}}$ in Eq. (2) are adjusted manually, based on experience and physics
 166 intuition: the expert fixes the weights and minimizes the function in Eq. (2) over the parameters \mathbf{p} .
 167 If the fit is unsatisfactory, a new set of weights is selected, and the optimization over \mathbf{p} is repeated
 168 until the tuner is satisfied.¹ The selection of weights is time-consuming and different experts may
 169 have different opinions about how well each observable is approximated by the model. Our goal is
 170 to automate the weight adjustment, yielding a less subjective and less time-consuming process to
 171 find the optimal physics parameters \mathbf{p} that will then be used in the actual MC simulation. This
 172 problem was also considered in [5], where weights are assigned based on how influential data is on
 173 constraining parameters. Also related to this work is that of [6], which treats tuning as a black-box
 174 optimization problem within the framework of Bayesian optimization, but with no weighting of
 175 data.

Does that not
 come in through
 the errors?

176 For convenience, we summarize our notation in Table 1.

Table 1: Notation.

Notation	Definition
\mathcal{O}	observables that are constructed from data and MC-based simulations in the form of histograms
$ \mathcal{O} $	the number of bins in an observable \mathcal{O}
$\mathcal{S}_{\mathcal{O}}$	the set of observables used in the tune
$ \mathcal{S}_{\mathcal{O}} $	the number of observables
\mathcal{R}	the data in the histograms
b	a bin of a histogram \mathcal{O}
\mathcal{R}_b	the data value in a bin
$\Delta \mathcal{R}_b$	data uncertainty corresponding to the data value in a bin

¹For the A14 tune, this selection was based on looking at hundreds of histograms such as the one shown in Fig. 1.

Notation	Definition
\mathbf{p}	a d -dimensional vector of real-valued parameters
$\text{MC}(\mathbf{p})$	an MC simulation that depends on the physics parameters \mathbf{p}
$\text{MC}_b(\mathbf{p})$	the MC simulation in a bin b
$\Delta\text{MC}_b(\mathbf{p})$	an uncertainty associated with the MC simulation in a bin b
$f_b(\mathbf{p})$	central value of the model prediction parameterized independently as a function of the model parameters \mathbf{p}
$\Delta f_b(\mathbf{p})$	the uncertainty of the model prediction parameterized independently as a function of the model parameters \mathbf{p}
\mathbf{w}	an $ \mathcal{S}_{\mathcal{O}} $ -dimensional vector of real-valued weights
$w_{\mathcal{O}}$	the weight given to a histogram in constructing a tune (if $w_{\mathcal{O}} = 0$ for some \mathcal{O} , then this observable will not influence the tuning of \mathbf{p}).
$\hat{\mathbf{p}}_{\mathbf{w}}$	optimal physics parameters for a given choice for the weights
\mathbf{w}^*	an optimal set of weights for the observables
$\hat{\mathbf{p}}_{\mathbf{w}^*}$	the optimal set of simulation parameters corresponding to an optimal set of weights \mathbf{w}^* for the observables
g	the outer objective function of $\mathbb{R}^{ \mathcal{S}_{\mathcal{O}} \times d} \mapsto \mathbb{R}$ used in the bilevel optimization
μ	a hyperparameter that specifies the percentage of the observables used in the robust optimization
$\chi_{\mathcal{O}}^2(\mathbf{p})$	the per-observable error averaged over all bins in the observable \mathcal{O}
$\mathbf{p}_{\text{ideal}}^{\mathcal{O}}$	the <i>ideal</i> tune for an observable \mathcal{O} , i.e., the parameters that minimize Eq. (12) when using only observable \mathcal{O} for the tune

177 2 Finding the Optimal Weights for Each Observable

178 In this section, we describe two mathematical formulations for finding the optimal weights in Eq. (2)
 179 that determine how much influence each observable should have on the optimization over the physics
 180 parameters \mathbf{p} : bilevel and robust optimization.

181 2.1 Bilevel optimization formulation

182 We formulate a bilevel optimization problem as follows:

$$\min_{\mathbf{w} \in [0,1]^{|\mathcal{S}_{\mathcal{O}}|}, \hat{\mathbf{p}}_{\mathbf{w}} \in \Omega} g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}}) \quad (3a)$$

$$\text{s.t.} \quad \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}} = 1 \quad (3b)$$

$$\hat{\mathbf{p}}_{\mathbf{w}} \in \arg \min_{\mathbf{p} \in \Omega} \chi^2(\mathbf{p}, \mathbf{w}) \quad (3c)$$

183 where the function $g : \mathbb{R}^{|\mathcal{S}_{\mathcal{O}}| \times d} \mapsto \mathbb{R}$ describes a merit function to determine the goodness of weights
 184 (see below for the definitions we use in this work). The lower-level Eq. (3c) (same as Eq. (2))
 185 corresponds to finding the optimal parameters $\hat{\mathbf{p}}_{\mathbf{w}}$ for a given set of weights \mathbf{w} , and the upper-
 186 level Eq. (3a) provides a measure of how good the weights are. The weights are normalized to
 187 sum to unity, see Eq. (3b), in order to prevent the trivial solution where all weights are 0. Bilevel
 188 optimization problems have been studied extensively in the literature, see, e.g., [7–11].

189 In the following, we discuss two definitions of the outer objective function $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$. Other
 190 formulations are possible and our selection is driven by the goal to achieve reasonably good agree-
 191 ment between the simulated and the observed data for all observables (rather than fitting a few
 192 observables extremely well and others poorly).

193 2.1.1 Formulation 1: Portfolio to balance mean and variance of errors

194 The portfolio objective function is motivated by portfolio optimization in finance [12], where the
 195 goal is to maximize the expected return while minimizing the risk. Translated to our problem, we
 196 want to minimize the expected error over all observables while also minimizing the variance over
 197 these errors.

198 For a given set of weights \mathbf{w} , we obtain the “ \mathbf{w} -optimal” parameters $\hat{\mathbf{p}}_{\mathbf{w}}$. For each observable
 199 \mathcal{O} , an error term is averaged over the number of bins in the observable ($|\mathcal{O}|$):

$$e_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) = \frac{1}{|\mathcal{O}|} \sum_{b \in \mathcal{O}} \frac{(f_b(\hat{\mathbf{p}}_{\mathbf{w}}) - \mathcal{R}_b)^2}{\Delta f_b(\hat{\mathbf{p}}_{\mathbf{w}})^2 + \Delta \mathcal{R}_b^2}, \quad \mathcal{O} \in \mathcal{S}_{\mathcal{O}}, \quad (4)$$

This is becoming a little formula-heavy.
 Fine if the authors want
 to be precise, but the
 average reader will skip the details

200 where we write $e_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w})$ because the error value for each observable depends (implicitly) on the
 201 choice of the weights \mathbf{w} . Thus, we obtain a set of $|\mathcal{S}_{\mathcal{O}}|$ average error values from which we compute
 202 the following statistics:

$$\mu(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) = \frac{1}{|\mathcal{S}_{\mathcal{O}}|} \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} e_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}): \text{average error over all observables,} \quad (5a)$$

$$\sigma^2(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) = \frac{1}{|\mathcal{S}_{\mathcal{O}}|} \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} [e_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) - \mu(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w})]^2: \text{ empirical variance of errors over all observables.} \quad (5b)$$

203 The portfolio objective function for the outer optimization then becomes

$$g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}}) = \mu(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) + \sigma^2(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}), \quad \text{Any chance this key formula of 2.1.1 could be explained in two sentences?} \quad (6)$$

204 which represents a simultaneous minimization of the expected error *and* the variance of the errors.
 205 For problems in which minimizing the variance is of higher priority, one can introduce a multiplier
 206 λ before the variance term that reflects “risk aversion”. In that case, if λ is large, we are more
 207 risk-averse, since reducing the variance associated with the errors will drive the minimization. If λ
 208 is small, we are less risk-averse, and minimizing the mean of the errors is emphasized.

209 2.1.2 Formulation 2: Scoring of model fit and data uncertainty

210 We consider a second outer objective function formulation based on scoring schemes ([13, Eq. (27)]).
 211 The performance of a generic predictive model P at a point x is defined by a scoring rule, $S(P, x) =$
 212 $-\left(\frac{x - \mu_P}{\sigma_P}\right)^2 - \log \sigma_P^2$, where P has mean performance μ_P and variance σ_P^2 . A larger value for $S(P, x)$
 213 signifies better model performance. Thus, we minimize the negative of $S(P, x)$:

$$s(P, x) = -S(P, x) = \left(\frac{x - \mu_P}{\sigma_P}\right)^2 + \log \sigma_P^2. \quad (7)$$

214 For our application, x corresponds to the simulation prediction $f_b(\mathbf{p})$, μ_P to our observation data
 215 \mathcal{R}_b , and the variance σ_P^2 to our data uncertainty $\Delta \mathcal{R}_b$. For each bin b in an observable, we calculate
 216 the score based on Eq. (7). Then, we compute the median (and mean) of the scores over all bins
 217 to obtain the median (average) performance for each observable. In order to form the upper-level
 218 objective function, we sum up the median (mean) scores over all observables:

- 219 • Outer objective based on median score

$$g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}}) = \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \tilde{s}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}), \quad (8a)$$

220

$$\tilde{s}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}}|\mathbf{w}) = \text{median of } \left\{ \left(\frac{f_b(\hat{\mathbf{p}}_{\mathbf{w}}) - \mathcal{R}_b}{\Delta \mathcal{R}_b} \right)^2 + \log(\Delta \mathcal{R}_b^2), \forall b \in \mathcal{O} \right\}. \quad (8b)$$

- 221 • Outer objective based on mean score

$$g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}}) = \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \bar{s}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}} | \mathbf{w}), \quad (9a)$$

222

$$\bar{s}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}} | \mathbf{w}) = \frac{1}{|\mathcal{O}|} \sum_{b \in \mathcal{O}} \left\{ \left(\frac{f_b(\hat{\mathbf{p}}_{\mathbf{w}}) - \mathcal{R}_b}{\Delta \mathcal{R}_b} \right)^2 + \log(\Delta \mathcal{R}_b^2) \right\}. \quad (9b)$$

223 In our numerical experiments, we analyze and compare both the performance of the median
 224 score and the mean score. Both the median and the mean score outer objective functions take into
 225 account the deviation of the prediction of $f_b(\hat{\mathbf{p}}_{\mathbf{w}})$ from \mathcal{R}_b and the uncertainty in the data $\Delta \mathcal{R}_b$.
 226 Thus, if an observable has large uncertainties in the data or the model $f_b(\hat{\mathbf{p}}_{\mathbf{w}})$ does not approximate
 227 the data \mathcal{R}_b well, the score for this observable deteriorates. Ideally, both terms $\left(\frac{f_b(\hat{\mathbf{p}}_{\mathbf{w}}) - \mathcal{R}_b}{\Delta \mathcal{R}_b} \right)^2$ and
 228 $\log(\Delta \mathcal{R}_b^2)$ will be small.

229 2.1.3 Solving the bilevel optimization problem using surrogate models

230 Solving the inner optimization problem (3c) for each weight vector \mathbf{w} is generally computation-
 231 ally non-trivial and its computational demand increases with the number of physics parameters \mathbf{p}
 232 that have to be optimized and the number of observables present. Therefore, the goal is to try
 233 as few weights \mathbf{w} as possible. We interpret the solution of the inner optimization problem as a
 234 black-box function evaluation of $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$ for \mathbf{w} . Given an initial set of input-output data pairs
 235 $\{(\mathbf{w}_i, g(\mathbf{w}_i, \hat{\mathbf{p}}_{\mathbf{w}_i}))\}_{i=1}^I$, we fit a surrogate model² (here a radial basis function [14]) that allows us to
 236 predict the values of $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$ at untried \mathbf{w} . In each iteration of the optimization algorithm, these
 237 predictions are used to select the most promising weight vector for which the inner optimization
 238 problem should be solved next. Promising weight vectors have either low predicted values of $g(\cdot)$
 239 or are far away from already evaluated points [15, 16]. Each time a new weight vector has been
 240 evaluated, the surrogate model is updated. This iterative process repeats until a stopping criterion
 241 has been met, e.g., a maximal number of weight vectors has been evaluated or a maximal CPU time
 242 has been reached. Details about the surrogate model algorithm are given in the online supplement
 243 Section 8.1.

244 When solving the problem Eq. (3), the optimization algorithm for the outer problem selects a
 245 set of weights \mathbf{w} . Given a choice for the weights, we solve the inner optimization problem Eq. (3c)
 246 using APPRENTICE to obtain a set of optimal physics parameters $\hat{\mathbf{p}}_{\mathbf{w}}$. Given $\hat{\mathbf{p}}_{\mathbf{w}}$, we compute the
 247 corresponding function value of the outer objective function, $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$. Based on this value, the

²This surrogate model for the weights is independent of the one used to evaluate the MC-based predictions.

248 outer optimization algorithm selects a new set of weights, which will be used to solve the inner
 249 optimization problem again. This leads to a new solution for Eq. (3c), which in turn gives a new
 250 value for the outer objective function. This process repeats until the outer optimization converges to
 251 an optimal set of weights for the observables (denoted by $\mathbf{w}^* = [w_1^*, \dots, w_{|\mathcal{S}_{\mathcal{O}}|}^*]^T$) and a corresponding
 252 optimal set of simulation parameters (denoted by $\hat{\mathbf{p}}_{\mathbf{w}^*}$).

Not sure if this is relevant here, but this algorithm sounds very stable once it sits in a local minimum, so how does it ensure that it finds a good global minimum? Any kind of mutation or annealing?

253 2.2 Robust optimization formulation

254 As an alternative to the bilevel formulation, we developed a single-level robust optimization formu-
 255 lation for finding the optimal weights for Eq. (2). Robust optimization estimates the parameters \mathbf{p}
 256 that minimize the largest deviation $(f_b(\mathbf{p}) - \mathcal{R}_b)^2$ over all bins in an uncertainty set \mathcal{U}_b of bin b :

$$\underset{\mathbf{w} \in [0,1], \mathbf{p} \in \Omega}{\text{minimize}} \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \frac{w_{\mathcal{O}}}{|\mathcal{O}|} \sum_{b \in \mathcal{O}} \underset{\mathcal{R}_b \in \mathcal{U}_b}{\text{maximize}} (f_b(\mathbf{p}) - \mathcal{R}_b)^2. \quad (10)$$

257 Assuming that the experiment and the MC simulation are described using independent random
 258 variables with mean \mathcal{R}_b , the uncertainty set \mathcal{U}_b for each bin b is described by the interval $[\mathcal{R}_b -$
 259 $\Delta \mathcal{R}_b - \Delta f_b(\mathbf{p}), \mathcal{R}_b + \Delta \mathcal{R}_b + \Delta f_b(\mathbf{p})]$.

260 Introducing slack variables $\mathbf{t} = [t_1, t_2, \dots, t_{|\mathcal{O}|}]$, we rewrite (10) as:

$$\underset{\mathbf{t}, \mathbf{w} \in [0,1], \mathbf{p} \in \Omega}{\text{minimize}} \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \frac{w_{\mathcal{O}}}{|\mathcal{O}|} \sum_{b \in \mathcal{O}} t_b \quad (11a)$$

subject to

$$t_b \geq (f_b(\mathbf{p}) - (\mathcal{R}_b - \Delta \mathcal{R}_b - \Delta f_b(\mathbf{p})))^2 \quad \forall b \in \mathcal{O}, \forall \mathcal{O} \in \mathcal{S}_{\mathcal{O}} \quad (11b)$$

$$t_b \geq (f_b(\mathbf{p}) - (\mathcal{R}_b + \Delta \mathcal{R}_b + \Delta f_b(\mathbf{p})))^2 \quad \forall b \in \mathcal{O}, \forall \mathcal{O} \in \mathcal{S}_{\mathcal{O}}$$

$$\sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \frac{w_{\mathcal{O}}}{|\mathcal{O}|} \geq \frac{\mu}{100} \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \frac{1}{|\mathcal{O}|} \quad (11c)$$

261 where the constraint (11c) is enforced to avoid the trivial solution of all weights being zero. In this
 262 constraint, we bound the sum of the weights away from zero by a hyperparameter μ that specifies
 263 the percentage of the observables that should be used in the optimization. Problem (11) is attractive
 264 because it formulates the problem of finding optimal weights as a single-level optimization problem,
 265 which is easier to solve than the bilevel problem Eq. (3).

266 Selecting the best μ among all the 100 runs of robust optimization is determined using a cumu-
 267 lative density curve of the number of observables satisfying $\frac{\chi_{\mathcal{O}}^2(\mathbf{p}^*, \mathbf{w})}{|\mathcal{O}|} \leq \tau$, where \mathbf{p}^* is the optimal
 268 parameter obtained from the robust optimization run, $\mathbf{w} = \mathbf{1}$, $\tau \in \mathbb{R}^+$ and $\mathcal{O} \in \mathcal{S}_{\mathcal{O}}$. Hence, in the

269 plot of this curve (e.g., see Figure 12), the number of observables on the y-axis is monotonically
 270 increasing as τ increases on the x-axis. Then, the area between the cumulative density curve for
 271 each robust optimization run and the ideal cumulative density curve is computed. To build the
 272 ideal cumulative density curve, the \mathbf{p}^* in $\frac{\chi_{\mathcal{O}}^2(\mathbf{p}^*, \mathbf{w})}{|\mathcal{O}|} \leq \tau$ is obtained by considering only observable
 273 \mathcal{O} in Eq. (2). The best run is then chosen to be the one whose area to the ideal cumulative density
 274 curve is the smallest. An example plot of the cumulative density curve and an illustration of the
 275 procedure to find the best run is included in Section 8.4 of the online supplement.

276 3 Data Pre-processing: Filtering Observables or Bins

277 We also investigate the question of how to detect and exclude observables or bins whose data \mathcal{R}_b
 278 cannot be explained by the MC simulation model. One special choice of weight for an observable is
 279 $w_{\mathcal{O}} = 0$, which corresponds to excluding (filtering out) the observable \mathcal{O} from our parameter tune.
 280 This is driven by a significant discrepancy between the simulation and data. Such discrepancies can
 281 arise for at least two reasons: (1) a mistake has been made in the experimental analysis; and/or (2)
 282 the observable is out of the domain of predictions that can be made reliably with the simulation.
 283 For our studies, we assume that the source of discrepancies is from (2). Because the simulation
 284 is a metamodel constructed from many smaller models, it is difficult to make *a priori* statements
 285 about the domain of its predictions. Important physics may be missing from the metamodel and/or
 286 a model can describe the mean behavior but not the rarer fluctuations around the mean. The
 287 simulation should be able to describe the physics, but some observables worsen the description.
 288 Thus, it is quite reasonable to exclude these observables.

Along the same line, what about observables where
 I know that I cannot go beyond a given level of precision?

289 In our discussion to this point, we have assumed that each *observable* has a given weight.
 290 However, in those situations where the model can describe the mean behavior, it can be beneficial
 291 to filter out individual bins b of the observable. In the observables considered in this study, and
 292 typical of the high energy physics phenomenon, the models can have difficulties in describing the
 293 rise and/or fall of a distribution (consider the example in Figure 1 where there is a rise from the first
 294 to the second bin and a fall from the penultimate to the last bin and the corresponding predicted
 295 data are far away from the measured, indicated by the red line in the lower pane.).

3.1 Filtering of observables by outlier detection

Using the surrogate model $f_b(\mathbf{p})$ to approximate the expensive MC simulation, we can efficiently minimize the per-observable- χ^2 function:

$$\chi_{\mathcal{O}}^2(\mathbf{p}) = \frac{1}{|\mathcal{O}|} \sum_{b \in \mathcal{O}} \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}, \mathcal{O} \in \mathcal{S}_{\mathcal{O}} \quad (12)$$

for each observable $\mathcal{O} \in \mathcal{S}_{\mathcal{O}}$, separately. $\chi_{\mathcal{O}}^2(\mathbf{p})$ represents the average per-bin error for the observable and the best possible fit of the model for this single observable. If we used only one observable for the tune, the parameters $\mathbf{p}_{\text{ideal}}^{\mathcal{O}}$ that minimize Eq. (12) would represent the *ideal* tune. The corresponding ideal objective function value $\chi_{\mathcal{O}}^2(\mathbf{p}_{\text{ideal}}^{\mathcal{O}})$ is the best possible result for each individual observable \mathcal{O} . Because the ideal parameter values will be different for each observable, we will not be able to obtain one parameter set that minimizes Eq. (12) for all observables simultaneously. Therefore, we obtain a set \mathcal{X} of length $|\mathcal{S}_{\mathcal{O}}|$ of ideal objective function values of Eq. (12): $\mathcal{X} := \{\chi_1^2(\mathbf{p}_{\text{ideal}}^1), \chi_2^2(\mathbf{p}_{\text{ideal}}^2), \dots, \chi_{|\mathcal{S}_{\mathcal{O}}|}^2(\mathbf{p}_{\text{ideal}}^{|\mathcal{S}_{\mathcal{O}}|})\} = \{\chi_i\}_{i=1}^{|\mathcal{S}_{\mathcal{O}}|}$. If the ideal error is large for some observables, it means that the model is not able to fit the data of these observables well at all, even with the freedom of not having to fit any other observables. Therefore, the inclusion of these data in optimizing Eq. (2) may negatively impact the overall optimization because large errors might drive the optimization.³

To address this issue, we use the distribution of the values in \mathcal{X} to identify outliers (observables with values for Eq. (12) “that deviate so much from other observations as to arouse suspicions that it was generated by a different mechanism.” [17]). We exclude the outlier observables from the optimization of Eq. (2) by setting their corresponding weights to zero, $w_{\mathcal{O}} = 0$.

There are multiple methods that can be used for outlier detection, such as scatter plots [18], Z-score [18, Section 1.3.5.17], interquartile range [19], generalized extreme studentized deviate [20], Grubb’s test [21, 22], Dixon’s Q test [23], Thompson tau test [24], Pierce’s Criterion [25], and Tietjen-Moore test [26], to name a few. We obtained reasonable results using the Z-score. For the set $\mathcal{X} = \{\chi_i\}_{i=1}^{|\mathcal{S}_{\mathcal{O}}|}$, the Z-score of an observation χ_i is defined as $z_i = (\chi_i - m)/s$ where m is the mean of the observation set \mathcal{X} and s is the standard deviation. We calculate the Z-score for each data point i in \mathcal{X} and define an outlier as $z_i \geq 3$. In other words, any ideal fit with a residual outside of 3 standard deviations is classified as an outlier. **Is the limit on z motivated by a number of observations included?**

The benefit of performing the outlier detection is that the computational cost of minimizing Eq. (2) is reduced. In addition, the optimization will not be biased by observables that the underlying model cannot describe well.

³We address later the fidelity of the surrogate model.

3.2 Filtering of bins by hypothesis testing

We explore a second and more refined approach that allows us to identify and exclude bin data from the optimization of Eq. (2). Instead of eliminating whole observables, we identify a subset of bins for each observable that cannot be approximated well by the MC simulator model and we exclude only those bins from the optimization [27]. The motivation behind this idea is driven by physics. In many cases, the physics observables are constructed such that one or both extrema of the observables' bins in the histogram represent high or low energy regimes. At low energy, a phenomenological model may be entirely inadequate. At high energy, genuine perturbative corrections not included in our calculations may be needed. It would be non-physical to adjust the model parameters to explain these extremes.

At some level, the separation into bins and observables is also ad-hoc, because each bin constitutes a statistically independent measurement, no?

To this end, we use the χ^2 test, which is a hypothesis test performed when the test statistic is χ^2 -distributed under the null hypothesis [28]. Note that the χ^2 test statistic is different from the $\chi^2(\mathbf{p}, \mathbf{w})$ objective function introduced earlier. We first compute the χ^2 test statistic for a subset \mathcal{B} of the bins in an observable \mathcal{O} using the computationally cheap approximation model $f_b(\mathbf{p})$:

$$\chi_{\mathcal{B}}^2(\mathbf{p}) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B} \subset \mathcal{O}} \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}. \quad (13)$$

For this statistic, we hypothesize that:

Null hypothesis H_0 : There is no significant difference between $f_b(\mathbf{p})$ and \mathcal{R}_b , in other words, the data \mathcal{R}_b can be appropriately described by $f_b(\mathbf{p})$.

Alternate hypothesis H_1 : H_0 is rejected, i.e., there is a significant difference between $f_b(\mathbf{p})$ and \mathcal{R}_b .

In (13), we have a sample of size $|\mathcal{B}|$ based on which we compute the χ^2 test statistic. However, the degrees of freedom of the χ^2 distribution is not $|\mathcal{B}|$ because the samples $f_b(\mathbf{p})$, $b \in \mathcal{B} \subset \mathcal{O}$ are not independent and they are related to each other through the parameters \mathbf{p} . Due to this relationship, the number of degrees of freedom is reduced (see [29] for a similar argument). Hence the resulting degrees of freedom of the χ^2 distribution for the set \mathcal{B} is given by

$$\rho_{\mathcal{B}} = |\mathcal{B}| - d, \quad (14)$$

where d is the dimension of \mathbf{p} .

We now choose a value for the significance level α (commonly used values are 0.01, 0.05, or 0.1). From a χ^2 distribution table, we then obtain the critical value $\chi_{c,\mathcal{B}}^2$ for bins in \mathcal{B} as a function of

353 the significance level α and degrees of freedom $\rho_{\mathcal{B}}$. More formally, we say that if the probability
 354 $P_{H_0}(T \leq \chi_{c,\mathcal{B}}^2) = \alpha$, then under $H_0 : T \sim \chi^2(\rho_{\mathcal{B}})$. Let us assume a random variable $Z \sim \chi^2(\rho_{\mathcal{B}})$,
 355 then $P(Z \leq \chi_{c,\mathcal{B}}^2) = \alpha$. Thus, to find $\chi_{c,\mathcal{B}}^2$, we need to compute the inverse of the cumulative
 356 distribution function (CDF) of the χ^2 distribution with $\rho_{\mathcal{B}}$ degrees of freedom and at level α . Then
 357 we compare the test statistic with the critical value to decide whether H_0 is accepted or not, i.e., if
 358 $\chi_{\mathcal{B}}^2 \leq \chi_{c,\mathcal{B}}^2$, we keep the bin subset \mathcal{B} ; otherwise, we cannot keep this bin subset.

359 We mainly intend to exclude bins at the extremes of the observables, and hence we require that
 360 the bins we keep are contiguous. For some observables all bins may pass the χ^2 test, for others, all
 361 bins may be excluded, or a subset of contiguous bins is kept.

362 The problem is then to find the largest contiguous subset of bins \mathcal{B} such that $\chi_{\mathcal{B}}^2 \leq \chi_{c,\mathcal{B}}^2$. This
 363 is equivalent to solving the mixed-integer program

$$\begin{aligned} \max_{s,e \in \{1,2,\dots,|\mathcal{O}|\}} \quad & e - s \\ \text{s.t.} \quad & \chi_{\mathcal{B}}^2 \leq \chi_{c,\mathcal{B}}^2, \quad \mathcal{B} = \{s, \dots, e\}, \end{aligned} \tag{15}$$

364 where s and e are the start and end indices of contiguous bins in observable \mathcal{O} . This problem can
 365 also be solved using a polynomial-time algorithm based on the maximum sub-array problem [30].
 366 This algorithm is described in Section 8.2 in the online supplement. In some cases, the bins to keep
 367 may not be unique, i.e., there may be multiple ranges of $\{s, \dots, e\}$ that are of the same maximum
 368 length and satisfy the null hypothesis (or satisfy the constraint in Eq. (15)). In practice, this is not
 369 a problem, since selecting any one of these bin subsets does not change the outcome of the filtering
 370 or the optimization in Eq. (2).

371 4 Numerical Experiments and Comparison of Different Tunes

372 In this section, we describe the setup of our numerical experiments, the datasets we use in our study,
 373 and the results. Additional information can be found in the online supplement.

374 4.1 Setup of the numerical experiments

375 We compare the results of using the methods shown in Table 2 for adjusting the weights of the
 376 observables in our datasets. The performance of each method is evaluated with and without data
 377 pre-processing (observable-filtering and bin-filtering approaches, see Sections 3.1 and 3.2), and when
 378 using a cubic polynomial (results presented in the online supplement) versus a rational approxima-

Table 2: Optimization methods used in this study.

Name	Methodology	Reference
“Bilevel-portfolio”	bilevel optimization with portfolio outer objective function	Section 2.1.1.
“Bilevel-medianscore”	bilevel optimization with median score outer objective function	Section 2.1.2.
“Bilevel-meanscore”	bilevel optimization with mean score outer objective function	Section 2.1.2.
“Robust optimization”	single level robust optimization approach	Section 2.2.
“Expert”	weight adjustment done by the expert (only for the A14 dataset, see Section 4.3)	[3]
“All-weights-equal”	no optimization is used and all observable weights are set to 1	

379 tion for $f_b(\mathbf{p})$ in APPRENTICE. We found relatively good performance using the degrees 3 and 1 for
 380 the numerator and denominator polynomial, respectively, for the rational approximation.

381 For the bilevel optimization formulation (see Eq. (3)), we made the following choices: The initial
 382 experimental design for the outer optimization has $|\mathcal{S}_{\mathcal{O}}| + 1$ points, where $|\mathcal{S}_{\mathcal{O}}|$ is the number of
 383 observables (number of weights to be adjusted) included. The total number of allowed outer objec-
 384 tive function evaluations (number of weight vectors tried) is 1000. Because the inner optimization
 385 function is multimodal, we use 100 multi-starts with APPRENTICE to solve it. The bilevel optimiza-
 386 tion with each method (portfolio, meanscore, medianscore) is repeated three times with different
 387 random seeds and we report the results of the best run.

388 For the robust optimization formulation (Eq. (11)), a total of 100 random values of $\mu \in (0, 100]$
 389 are used when evaluating Eq. (11c) and, for each μ , the algorithm is run once. The best run amongst
 390 these is returned as the best μ for the robust optimization. The procedure to select the best μ is
 391 described in Section 2.2.

Are the results for bi-level and robust optimization produces with a comparable
 computing time? Ah, found the table, please mention here that it exists.

392 4.2 Comparison metrics and optimal tuning parameters

393 There are many ways to assess the quality of a tune. In many cases, the domain experts visually
 394 inspect a potentially large number of histograms (see, e.g., Figure 1) to make a judgment. As an
 395 objective measure, we propose three metrics, each represented as a single number for each tuning
 396 method, that can be used to compare the quality of the model fits obtained by the different methods
 397 in a more objective fashion:

- 398 1. *Weighted χ^2* : the sum over all χ^2 at the best $\hat{\mathbf{p}}_{\mathbf{w}^*}$,

$$\sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}}^* \sum_{b \in \mathcal{O}} \frac{(f_b(\hat{\mathbf{p}}_{\mathbf{w}^*}) - \mathcal{R}_b)^2}{\Delta f_b(\hat{\mathbf{p}}_{\mathbf{w}^*})^2 + \Delta \mathcal{R}_b^2}$$

399 where $w_{\mathcal{O}}^*$, the weight of observable \mathcal{O} , is scaled such that $w_{\mathcal{O}}^* \in [0, 1]$ and $\sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}}^* = 1$.

400 2. *A-optimality*:

$$\text{Tr}(\mathbf{\Gamma}_{\text{post}}(\hat{\mathbf{p}}_{\mathbf{w}^*}, \mathbf{w}^*)) = \sum_{i=1}^d \lambda_i$$

401 3. *log D-optimality*:

$$\log \det(\mathbf{\Gamma}_{\text{post}}(\hat{\mathbf{p}}_{\mathbf{w}^*}, \mathbf{w}^*)) = \sum_{i=1}^d \log \lambda_i,$$

402 where λ_i are the eigenvalues of $\mathbf{\Gamma}_{\text{post}}$, $\mathbf{\Gamma}_{\text{post}}$ is the weighted posterior covariance matrix in the
 403 Bayesian formulation of the inverse problem, d is the dimension of $\hat{\mathbf{p}}_{\mathbf{w}^*}$. To find $\mathbf{\Gamma}_{\text{post}}$, we compute
 404 the optimal parameter point $\hat{\mathbf{p}}_{\mathbf{w}^*}$, which is also referred to as the maximum a posteriori probability
 405 estimate in the context of Bayesian inverse problems [31]. Given the optimal parameters, we can
 406 find a linearization of the model as

$$\mathbf{F}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}^*}) = \left[\frac{\partial f_1(\hat{\mathbf{p}}_{\mathbf{w}^*})}{\partial \mathbf{p}}, \frac{\partial f_2(\hat{\mathbf{p}}_{\mathbf{w}^*})}{\partial \mathbf{p}}, \dots, \frac{\partial f_{|\mathcal{O}|}(\hat{\mathbf{p}}_{\mathbf{w}^*})}{\partial \mathbf{p}} \right]^\top$$

407 for each observable \mathcal{O} . Then the weighted posterior can be approximated by a Gaussian $\mathcal{N}(\hat{\mathbf{p}}_{\mathbf{w}^*}, \mathbf{\Gamma}_{\text{post}})$.

408 Here

$$\mathbf{\Gamma}_{\text{post}}(\hat{\mathbf{p}}_{\mathbf{w}^*}, \mathbf{w}^*) = \left(\sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}}^* \mathbf{F}_{\mathcal{O}}^\top(\hat{\mathbf{p}}_{\mathbf{w}^*}) \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{F}_{\mathcal{O}}(\hat{\mathbf{p}}_{\mathbf{w}^*}) \right)^{-1} \quad (16)$$

409 where $\mathbf{\Gamma}_{\text{noise}}[\mathcal{O}] = \text{diag}(\Delta f_1(\hat{\mathbf{p}}_{\mathbf{w}^*})^2 + \Delta \mathcal{R}_1^2, \Delta f_2(\hat{\mathbf{p}}_{\mathbf{w}^*})^2 + \Delta \mathcal{R}_2^2, \dots, \Delta f_{|\mathcal{O}|}(\hat{\mathbf{p}}_{\mathbf{w}^*})^2 + \Delta \mathcal{R}_{|\mathcal{O}|}^2)$ and $w_{\mathcal{O}}^*$
 410 is the weight of observable \mathcal{O} obtained from the methods and is scaled such that $w_{\mathcal{O}}^* \in [0, 1]$ and
 411 $\sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} w_{\mathcal{O}}^* = 1$.

412 The $\mathbf{\Gamma}_{\text{post}}(\hat{\mathbf{p}}_{\mathbf{w}^*}, \mathbf{w}^*)$ calculated at the optimal parameters and the optimal weights in (16) are
 413 used here to describe the confidence region around the tuned parameters $\hat{\mathbf{p}}_{\mathbf{w}^*}$. In order to summarize
 414 the multidimensional nature of $\mathbf{\Gamma}_{\text{post}}$ into a scalar quantity, we use the A- and log D-optimality
 415 criteria. A graphical representation of the optimality criteria is shown in Figure 2. The A-optimality
 416 criterion computes the trace of $\mathbf{\Gamma}_{\text{post}}$, which is equivalent to the sum of its eigenvalues. This metric
 417 is proportional to the sum of the semiaxis lengths of the confidence ellipsoid of the parameters (lower
 418 is better), which corresponds to the average sum of the variances of the estimated parameters for
 419 the model [32]. The log D-optimality criterion computes the log of the determinant of $\mathbf{\Gamma}_{\text{post}}$, which
 420 is equivalent to the sum of the log of the eigenvalues of $\mathbf{\Gamma}_{\text{post}}$. This metric is proportional to the
 421 (log) volume of the confidence ellipsoid of the parameters (lower is better) [33]. It can be interpreted
 422 in terms of Shannon information.

Would a Gaussianity test of the output be interesting for those metrics?

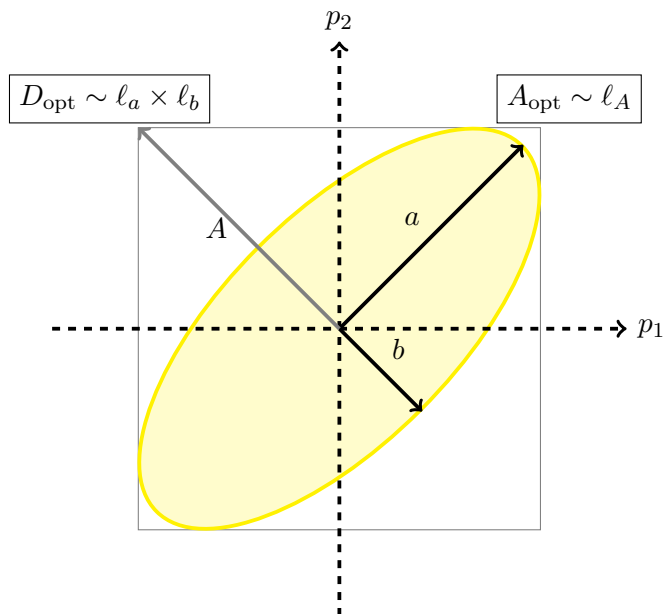


Figure 2: Graphical interpretation of optimal design criteria. The D-criterion is the determinant of the covariance matrix or the volume of the joint confidence region. The A-criterion is the trace of the covariance matrix or the dimension of a box that bounds the joint confidence region.

423 4.3 The A14 dataset

424 We chose the A14 tune [3] of the PYTHIA⁴ event generator [34] as one benchmark for developing
 425 and testing the methods proposed in this work. This tune has been widely used for Large Hadron
 426 Collider (LHC) simulations, the methodology was reasonably well documented, and some of the
 427 simulation data were available to us. The generator settings are available in Section 8.15 of the
 428 online supplement.

429 The A14 dataset contains 406 observables (thus, 406 weights to optimize) and there are 10
 430 tunable physics parameters \mathbf{p} . In our studies, we use the RIVET [35] package to compare our
 431 predictions to data. The definition and ranges of the parameters \mathbf{p} used to construct the original
 432 A14 tune are shown in Table 22 in Section 8.3 of the online supplement.

433 Because the coefficients of the cubic interpolation used in [3] were not available to us, we start
 434 by reproducing the hand-tuned parameter values with the NNPDF published in Table 3 in [3],
 435 which we refer to as *NNPDF*. In particular, we use the weights given in Table 2 in [3], use their

⁴To match the original study, we used version v8.186.

436 optimal parameter values as a starting point for the χ^2 minimization, and apply our optimizer
 437 to Eq. (2). The resulting parameter values are reassuringly close to the values reported in [3]
 438 as showed in Table 3 where we label the original parameters as NNPFD, and the re-optimized
 439 parameter values as Expert. We observe that most of the NNPFD parameter values lie within
 440 the confidence interval derived from eigentunes (see Section 5) for the re-optimized Expert values.
 441 Additionally, to check whether the parameter \mathbf{p} reported in [3] is within the confidence ellipsoid
 442 centered on the parameter $\hat{\mathbf{p}}_{\mathbf{w}}$ obtained from the χ^2 minimization (i.e., Expert parameter values),
 443 we calculate $s \equiv \|\mathbf{L}^T(\mathbf{p} - \hat{\mathbf{p}}_{\mathbf{w}})\|_2$, where \mathbf{L} is the Cholesky factor of $\mathbf{\Gamma}_{\text{post}}(\hat{\mathbf{p}}_{\mathbf{w}}, \mathbf{w})$ from Eq. (16)
 444 with weights \mathbf{w} given in [3]. Since $s = 2.73 \times 10^{-3}$ is less than one, we say that the parameter \mathbf{p} is
 445 covered within the confidence ellipsoid centered on $\hat{\mathbf{p}}_{\mathbf{w}}$ [36].

446 In the remainder of this paper, we use the Expert parameter values for comparison, rather than
 447 the NNPFD values, and we refer to this tune as the *Expert* tune in our comparisons. This change
 448 provides a better comparison, because we found that the original NNPFD parameter values did not
 449 correspond to a minimizer of the χ^2 optimization, Eq. (2), and thus using the original values would
 450 unfairly disadvantage the NNPFD tune in our comparisons. The main reason for this discrepancy is
 451 the fact that we use a better optimization routine, and that we did not have access to the coefficients
 452 of the cubic interpolation used in [3].

Table 3: Parameter values for A14 published tune (left), and A14 corrected expert tune and corresponding eigentune confidence intervals (right).

Parameter name	A14 published expert tune	A14 corrected expert tune		
	NNPDF	Expert	min	max
<code>SigmaProcess:alphaSvalue</code>	0.140	0.143	0	0.299
<code>BeamRemnants:primordialKThard</code>	1.88	1.904	1.899	1.908
<code>SpaceShower:pT0Ref</code>	1.56	1.643	1.621	1.667
<code>SpaceShower:pTmaxFudge</code>	0.91	0.908	0.897	0.917
<code>SpaceShower:pTdampFudge</code>	1.05	1.046	1.041	1.052
<code>SpaceShower:alphaSvalue</code>	0.127	0.123	0.120	0.127
<code>TimeShower:alphaSvalue</code>	0.127	0.128	0	0.355
<code>MultipartonInteractions:pT0Ref</code>	2.09	2.149	1.055	3.442
<code>MultipartonInteractions:alphaSvalue</code>	0.126	0.128	0	0.289
<code>BeamRemnants:reconnectRange</code>	1.71	1.792	1.784	1.802

453 The 10 tunable parameters are primarily related to the production of additional jets in the
 454 collisions, the distribution of energy within those jets, and the kinematics (angles and momenta)

455 of the jets. They also relate to the sharing and spread of energy in the soft portion of the event,
 456 the portion that is less dependent on the hard process (e.g., top-quark production or Z -boson
 457 production). As a physics reader I would really want to know those tuning parameters and their impact. Can
 you add an appendix with them, I really think that would be cool to have.

458 The A14 observables are measurements of properties of proton-proton collisions at $\sqrt{s} = 7$ TeV
 459 performed by the ATLAS collaboration. These include event properties (e.g., the Z -boson transverse
 460 momentum, or the opening angles between the highest transverse momentum jets in the event) and
 461 properties of jets (e.g., the spread of energy within a jet, or the momentum of particles within a jet).
 462 In the publication [3], the 406 observables are categorized into 10 groups (see Table 11), namely
 463 *Track jet* properties (200 observables), *Jet shapes* (59 observables), *Dijet decorr* (9 observables),
 464 *Multijets* (8 observables), p_T^Z (fit range < 50 GeV, 20 observables), *Substructure* (36 observables), $t\bar{t}$
 465 *gap* (4 observables), $t\bar{t}$ *jet shapes* (20 observables), *Track-jet UE* (8 observables), and *Jet UE* (42
 466 observables). The highest weights in [3] are assigned to observables that relate to the production of
 467 additional high-momentum partons (the ratios of 3-jet to 2-jet events, and the fraction of top-quark
 468 production events that do not have an additional central jet). On the other hand, low weights are
 469 assigned to observables that measure the same physical phenomenon in several kinematic regimes.
 470 The weighting of these observables ensures that the additional radiation and soft part of the events
 471 are consistent and well-modeled for all hard processes. In addition, these parameters are difficult or
 472 impossible to constrain using data from e^+e^- collision events, and they must be tuned using data
 473 from the LHC.

474 4.4 The SHERPA dataset

475 As a second benchmark, we tune a set of parameters for the SHERPA event generator [37]. To
 476 our knowledge, the default parameters were not optimized by weighting data, and thus serve as an
 477 unbiased cross-check of our results. In contrast to the A14 dataset used to tune PYTHIA, the data
 478 are confined to observables at e^+e^- colliders. The data includes event shapes and charged particle
 479 inclusive spectra from Z -boson decays, differential and integrated jet rates, measurements of B -
 480 hadron fragmentation, and the multiplicity of various hadrons [38–41]. Accordingly, the parameters
 481 are limited to those of the SHERPA hadronization model.

482 The SHERPA dataset contains 88 observables, hence 88 weights to optimize. The number 88 is
 483 significantly less than the set of observables available in the RIVET analyses (126) for the following
 484 reasons. First, we reduce the number of observables to 114 by removing those that measure more
 485 than 3 jets, since this is beyond the scope of the physics simulation. Then, we apply a pre-filter
 486 step that removes distributions where *none* of the data bins fall within the envelope of predictions
 487 from our surrogate model. These all correspond to single-bin particle counts (such as the number

488 of f_0 mesons) that the SHERPA hadronization model either grossly under- or over-estimates. There
 489 are 13 tunable physics parameters whose definition and ranges are shown in Table 23 in Section 8.3
 490 of the online supplement. These parameters are all part of the cluster model that produces physical
 491 particles from quarks and gluons. **Reading this part, the SHERPA data set seems to be much less interesting than
 the A14 set. What is the reason to include it? Some specific strength?**

492 4.5 Data pre-processing: filtering out observables and bins

493 In this subsection, we present the results of applying the filtering methods described in Sections 3.1
 494 and 3.2. First, we consider the outlier detection method described in Section 3.1. We find that
 495 the filtering results differ based on the choice of surrogate function (cubic polynomial versus a
 496 rational approximation). Based on the comparison of surrogate function predictions to the full
 497 MC simulations, we believe that the rational approximation yields a more faithful representation.
 498 Therefore, we present our main results using only the rational approximation. The names of the
 499 outlier observables in the A14 and the SHERPA dataset using a cubic polynomial and a rational
 500 approximation, respectively, are shown in the online supplement in Sections 8.5 and 8.6. Table 4
 501 shows a distribution of the $\chi_{\mathcal{O}}^2$ values obtained for each observable \mathcal{O} from Eq. (12) for A14 (left)
 502 and SHERPA (right) when using the rational approximation. We find that the per-observable ideal
 503 parameters yield mostly small $\chi_{\mathcal{O}}^2$ values (in $[0, 1)$), but outliers are present in both datasets. Using
 504 the rational approximation, 9 and 3 outlier observables are filtered from the A14 and SHERPA
 505 datasets, respectively.

Table 4: Distribution of the $\chi_{\mathcal{O}}^2$ values for A14 (left) and SHERPA (right). 2.0438 and 2.0177 correspond to the values where the Z-score equals 3 (see Section 3.1). The observables with $\chi_{\mathcal{O}}^2 \geq 2.0438$ for A14 and $\chi_{\mathcal{O}}^2 \geq 2.0177$ for SHERPA are the outliers. There are 9 outliers (6+2+1) in A14 and 3 outliers (1+2+0) in SHERPA.

A14		SHERPA	
$\chi_{\mathcal{O}}^2$ range	Number of observables	$\chi_{\mathcal{O}}^2$ range	Number of observables
[0, 1)	367	[0, 1)	82
[1, 2.0438)	30	[1, 2.0177)	3
[2.0438, 3)	6	[2.0177, 3)	1
[3, 4)	2	[3, 4)	2
[4, 5)	1	[4, 5)	0

506 Figure 3 shows the outcomes of the bin-filtering approach described in Section 3.2 for each ob-
 507 servable \mathcal{O} in A14 (top) and SHERPA (bottom) when using the rational approximation. In both

508 datasets, multiple bins are removed. More specifically, most bins are removed in the *Track jet prop-*
509 *erties* and p_T^Z groups of the A14 dataset. The patterns in the A14 plot result from the partitioning
510 of the data. For *Tracked jet properties* (labeled A), the observables are replicated for two values of
511 jet cone size ($R = 0.4, 0.6$), explaining the similarities between bins (1, 100) and (101, 200). Fur-
512 thermore, 4 types of observables are considered, and each is sliced into different ranges of transverse
513 momentum and rapidity.

514 In the SHERPA dataset, **all bins are removed from some observables** whereas from two observ- *which? discuss!*
515 ables, we remove only two and five bins. Additionally, since the number of degrees of freedom of
516 the χ^2 distribution is reduced by the number of parameters that the bins share in each observable
517 (see Eq. (14)), the bin filter is not applied to any observable with fewer than 10 and 13 bins in the
518 A14 and the SHERPA datasets, respectively. The names of the observables from which the bins have
519 been filtered and their χ^2 test statistic and critical χ^2 values are given in Sections 8.7 and 8.8 of the
520 online supplement. The single bin observables correspond to counts of a particular type of particle.

521 4.6 Results for the A14 dataset

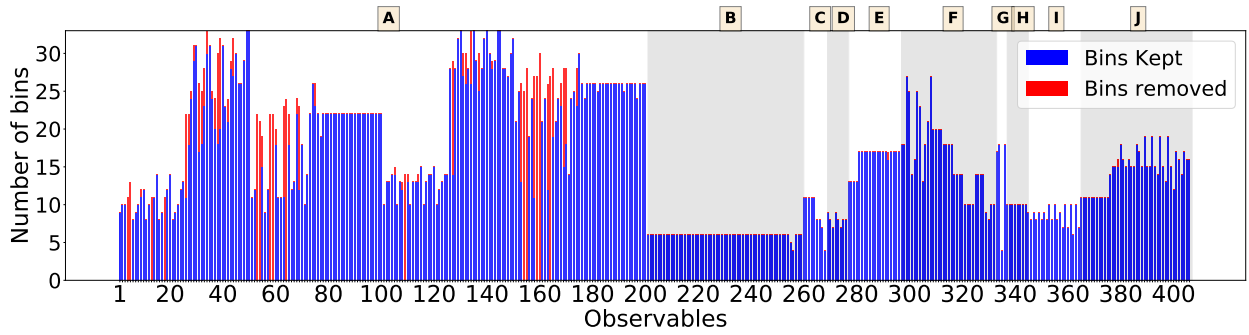
522 In this section, we present a detailed analysis of our results with the A14 dataset.

523 4.6.1 Comparison metric outcomes for the A14 dataset

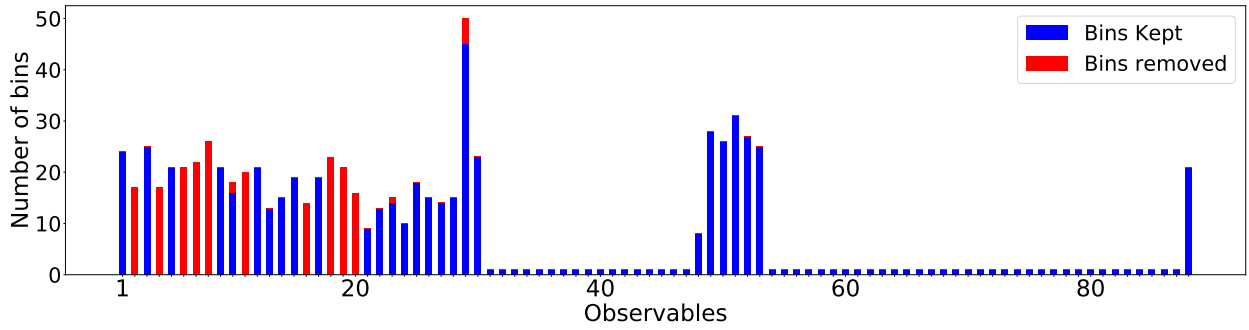
524 In this section, we consider the three metrics introduced in Section 4.2 to compare various tunes.
525 For the A14 dataset, Tables 5-7 show the results when using the rational approximation for the full
526 data, the observable-filtered data, and the bin-filtered data, respectively. The results when using
527 the cubic polynomial approximation are shown in the online supplement in Section 8.12.1. Note
528 that smaller numbers indicate better performance. We bold the smallest number of each metric
529 for better visualization. For our comparison metrics, we take into account all observables and bins,
530 respectively, but we do not use the filtered out observables and bins when determining the optimal
531 parameters.

532 Based on these results we can see that no method performs the best for all metrics in all
533 cases. In fact, for the full dataset, the *Expert* tune has the best score for two of our three metrics.
534 Nonetheless, the automated methods do produce comparable results in those cases. The robust
535 optimization consistently achieves the best performance under the Weighted χ^2 criterion.

536 The Bilevel-portfolio method performs the best under the A-optimality criteria, and the *Expert*
537 tune performs the best under the D-optimality criteria for the observable-filtered datasets. The
538 Bilevel-portfolio method performs the best under the A- and D-optimality criteria for the bin-
539 filtered datasets. In comparison to the results obtained with the cubic polynomial approximation



(a) Bins kept and removed by the bin filter in all A14 observables organized by the observable group. Group A is *Track jet properties*, group B is *Jet shapes*, group C is *Dijet decorr*, group D is *Multijets*, group E is p_T^Z , group F is *Substructure*, group G is $t\bar{t}$ gap, group H is *Track-jet UE*, group I is $t\bar{t}$ jet shapes, and group J is *Jet UE*.



(b) Bins kept and removed by the bin filter in all SHERPA observables.

Figure 3: Illustration of the bin filtering results.

540 (see Section 8.12.1 of the online supplement), the rational approximation yields better results for
 541 all methods under the Weighted χ^2 criterion.

542 When comparing across Tables 5-7, we see that in most cases, results with the observable-filtered
 543 data and bin-filtered data provide smaller values compared with those using the full dataset. We
 544 observe that by filtering out the observables and bins that cannot be well explained by the model,
 545 the quality of the model fits can be improved.

Please do not just state the comparison results, but explain them or at least put them into context.

Table 5: A14 results with the *full dataset* when using the *rational* approximation. Lower numbers are better. The best results are in bold.

Method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1119	0.8513	-63.6805
Bilevel-medscore	0.1320	0.7673	-63.3846
Bilevel-portfolio	0.1224	0.9425	-61.1694
Expert	0.0965	0.5705	-68.4091
All-weights-equal	0.0815	0.7673	-64.0008
Robust optimization	0.0402	1.0526	-65.7547

Maybe combine Tabs.5-7 into one, with a layout that makes them easy to compare?

Table 6: A14 results with the *observable-filtered* data when using the *rational* approximation. Lower numbers are better. The best results are indicated in bold.

Method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.0671	0.6793	-65.1939
Bilevel-medscore	0.0823	0.7008	-64.3410
Bilevel-portfolio	0.1372	0.5130	-68.0382
Expert	0.0965	0.5705	-68.4091
All-weights-equal	0.0843	0.7110	-64.4546
Robust optimization	0.0388	1.1086	-65.7182

546 4.6.2 Comparison of the cumulative distribution of bins at different variance levels

547 In this section, we introduce a new summarized graphical comparison of the results that is motivated
 548 by the bottom pane in the histogram plot of Figure 1. We study the distribution of the χ^2 values
 549 per bin obtained using different tuning approaches. For each parameter set, we compute the ratio

Table 7: A14 results with the *bin-filtered* data when using the *rational* approximation. Lower numbers are better. The best results are in bold. 1811 out of 7010 bins were filtered out and not used during the optimization. All numbers are computed over all 7010 bins.

method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.0677	0.7098	-66.1484
Bilevel-medianscore	0.1198	0.7448	-64.2754
Bilevel-portfolio	0.1464	0.3747	-70.5889
Expert	0.0965	0.5705	-68.4091
All-weights-equal	0.0835	0.7185	-65.3507
Robust optimization	0.0439	0.8332	-67.4820

550 $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$ of the residual between the data and the prediction divided by the variance
 551 per bin. The r_b values are sorted from the smallest to largest, and the cumulative distribution is
 552 formed.

553 The cumulative distribution plot for all bins in the A14 dataset is shown in Figure 4 and for
 554 the bins in each category in Figure 5. The more bins that reside on the bands of variance levels
 555 less than 1 the better, as this indicates smaller deviations of the model from the experimental data.
 556 When analyzing these results it is important to note that even though all the category plots have a
 557 scale between 0 and 1 on the y axis, the number of bins in one category of A14 is very different from
 558 the other. For e.g., more than 50% of all bins in the A14 dataset belong to *Track Jet Properties*.
 559 Hence, we see that the trend of the curves in the plot for *Track Jet Properties* in Figure 5 follows
 560 more closely to the trend of the curves when all A14 bins are considered as in Figure 4.

561 It can be seen from Figure 4 that there is a small difference among the approaches when all
 562 A14 bins are considered. Near the variance boundary, the difference between the approaches is
 563 even smaller. Additionally, at the variance boundary, all approaches perform better than the *Expert*
 564 tune. Figure 5 shows that these differences become more prominent when considering individual
 565 categories of the A14 data. For instance, the parameters obtained from the robust optimization
 566 perform well for *Jet shapes* and *Track-jet UE*. We also see that near the variance boundary, the
 567 parameters obtained from the *Expert* tune perform better for *Multijets* and $t\bar{t}$ *gap* whereas the
 568 parameters obtained from the other approaches perform better for *Substructure*. These plots also
 569 show that there is a trade-off in fitting among the different approaches, which enables the physicist
 570 to use these results as guidance for selecting the most appropriate tuning method depending on the
 571 categories that are of greater significance.

Not sure I understand this right, how would this kind of distribution look for a Gaussian statistics sample or for some kind of systematics or anything like that? Maybe I could derive this somehow, but you are the authors :)

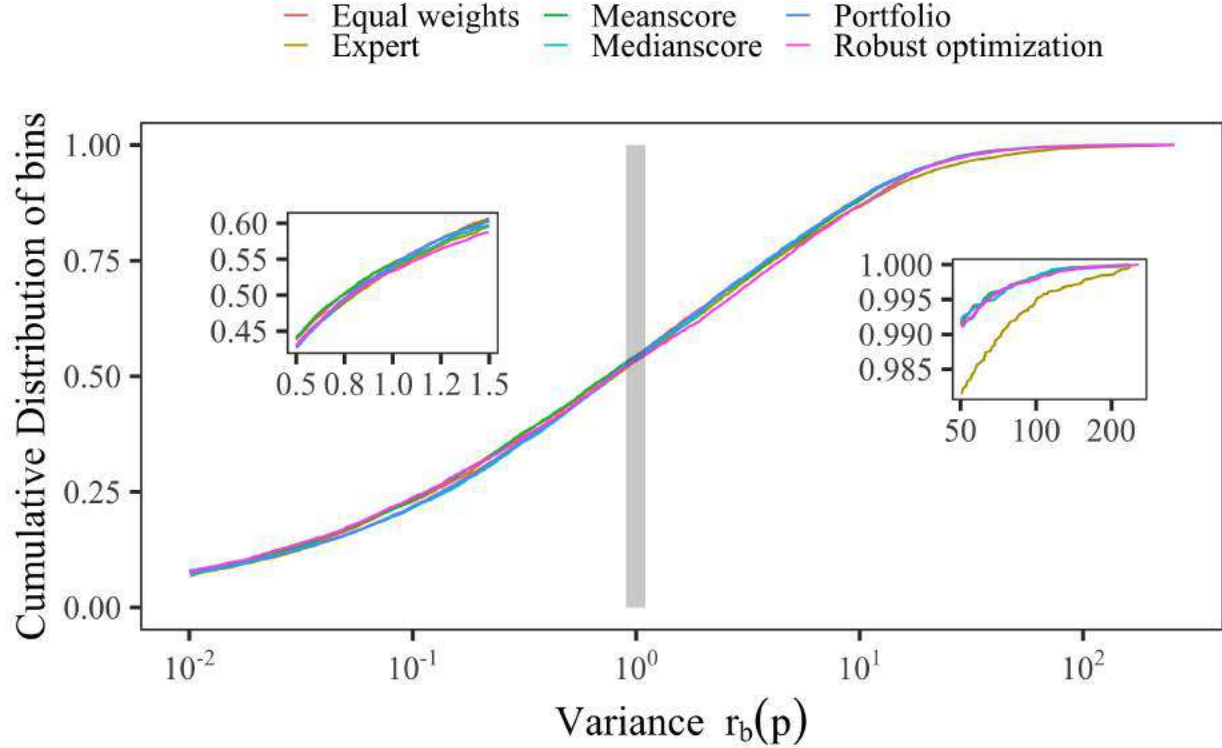


Figure 4: Cumulative distribution of all bins (y-axis) in the A14 dataset at different bands of variance levels (x-axis) given by $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$.

572 4.6.3 Optimal parameter values for the A14 dataset with rational approximation

573 The optimal parameter values for the A14 dataset when using the full dataset, the outlier-filtered
 574 dataset, and the bin-filtered dataset are shown in Tables 8, 9, and 10, respectively. For a better
 575 visual comparison of the different solutions obtained with our methods, we illustrate the [0,1]-scaled
 576 optimal values in the online supplement Section 8.11. We have also computed the Euclidean distance
 577 between the *Expert* tune and our tunes after normalizing the parameter values to [0,1].

578 In Table 8, we can see that there are differences between the optimal parameters obtained with
 579 different methods. In particular, the results of the Bilevel-meanscore method tend to be further away
 580 from the expert's solution than the other methods. The robust optimization and All-weights-equal
 581 results are very similar to each other as well as to the *Expert's* solution.

Table 8: Optimal parameter values for the A14 dataset obtained when using all observables in the optimization and the rational approximation. Euclidean distance is calculated based on the normalized parameter values.

ID	Parameter name	Expert	Bil.-meanscore	Bil.-medianscore	Bil.-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.138	0.133	0.136	0.139	0.137
2	<code>BeamRemnants:primordialKThard</code>	1.904	1.855	1.723	1.796	1.883	1.851
3	<code>SpaceShower:pT0Ref</code>	1.643	1.532	1.184	1.322	1.588	1.493
4	<code>SpaceShower:pTmaxFudge</code>	0.908	1.014	1.083	1.041	1.025	1.026
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.071	1.084	1.061	1.084	1.067
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.128	0.129	0.128	0.127	0.128
7	<code>TimeShower:alphaSvalue</code>	0.128	0.130	0.129	0.128	0.132	0.129
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	2.033	1.883	1.937	2.052	2.076
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.124	0.118	0.120	0.126	0.125
10	<code>BeamRemnants:reconnectRange</code>	1.792	2.082	1.914	1.987	2.602	1.980
Euclidean distance from the expert solution			0.290	0.664	0.475	0.268	0.301

Table 9: Optimal parameter values for A14 when using the rational approximation with all methods after outlier detection to filter out observables that cannot be approximated well by the model. Euclidean distance is calculated based on the normalized parameter values.

ID	Parameter name	Expert	Bilevel-meanscore	Bilevel-medianscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.140	0.138	0.141	0.138	0.139
2	<code>BeamRemnants:primordialKThard</code>	1.904	1.865	1.839	1.861	1.879	1.843
3	<code>SpaceShower:pT0Ref</code>	1.643	1.574	1.603	1.593	1.614	1.550
4	<code>SpaceShower:pTmaxFudge</code>	0.908	0.953	0.906	0.984	1.006	0.950
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.076	1.081	1.060	1.075	1.062
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.128	0.128	0.129	0.128	0.127
7	<code>TimeShower:alphaSvalue</code>	0.128	0.123	0.123	0.118	0.132	0.124
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	2.064	2.017	2.095	2.022	2.039
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.126	0.125	0.129	0.125	0.126
10	<code>BeamRemnants:reconnectRange</code>	1.792	1.852	1.903	1.801	2.719	1.937
Euclidean distance from the expert solution			0.227	0.293	0.273	0.291	0.254

Again, maybe there is a way to combine Tabs. 8-10 into something easier to read and compare and get the point?

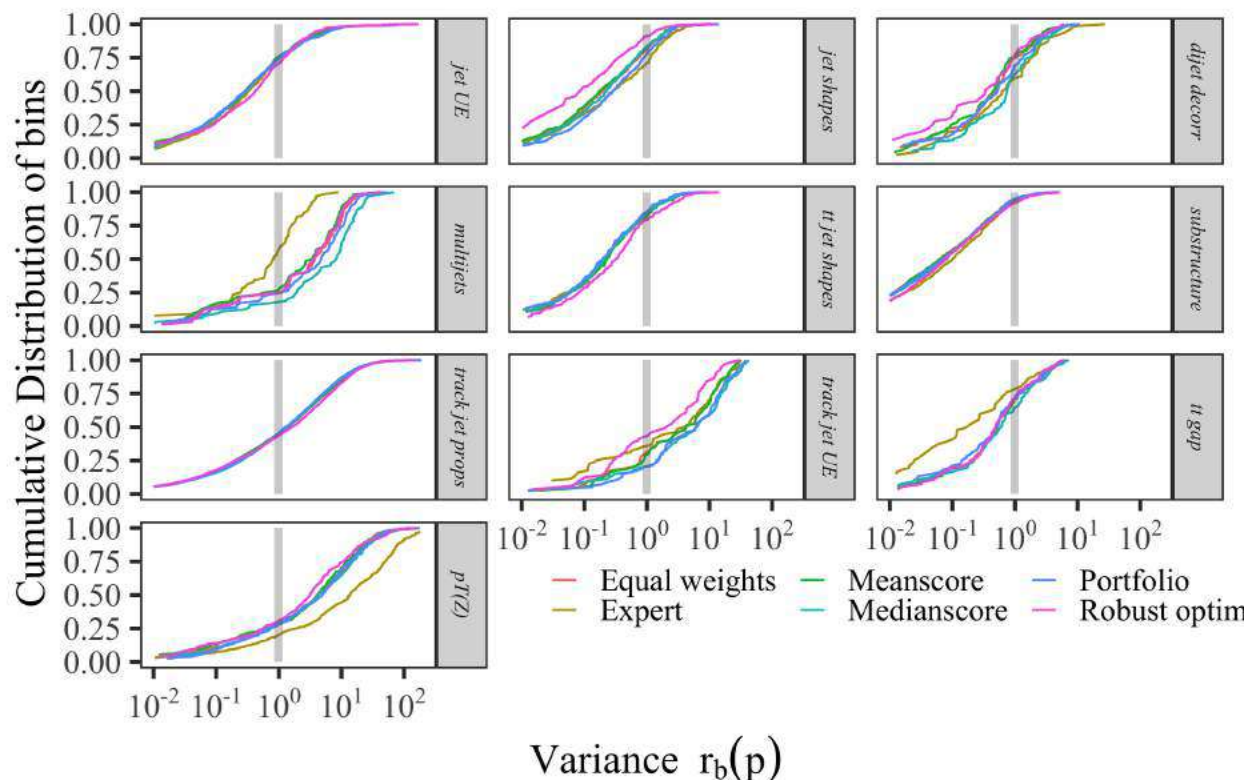


Figure 5: Cumulative distribution of bins (y-axis) in each category of the A14 dataset at different bands of variance levels (x-axis) given by $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$.

582 4.6.4 Comparison of optimal weights for the A14 dataset with rational approximation

583 We compare the optimal weights obtained by the different tuning methods in Table 11. We normalize
 584 the weights obtained to match the scale of weights assigned by *Expert* published in [3]. In each
 585 group, we report the average weight of observables in that group. The *Expert* tune assigned the
 586 highest weights to the categories *Multijets* and *t \bar{t} gap*. The robust optimization approach sets some
 587 of the weights for *Track jet properties* to zero. The four *Track-jet properties* classes of observables
 588 are nearly dependent resulting in redundant components of least-square residuals. Because the
 589 robust optimization approach can be viewed as minimizing the maximum residual, it detects this
 590 redundancy of observables, and sets the weights accordingly to zero. We observe in Figure 5 that
 591 setting these weights to zero does not degrade the residuals of these observables, confirming that
 592 redundant information is presented.

Table 10: Optimal parameter values obtained for A14 with the rational approximation with all methods after using the bin-filtering approach that excludes individual bins from the optimization 1811 bins out of 7010 total bins were filtered out. Euclidean distance is calculated based on the normalized parameter values.

ID	Parameter name	Expert	Bilevel-meanscore	Bilevel-medianscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.139	0.140	0.131	0.137	0.140
2	<code>BeamRemnants:primordialKThard</code>	1.904	1.877	1.885	1.811	1.822	1.876
3	<code>SpaceShower:pT0Ref</code>	1.643	1.572	1.561	2.227	1.426	1.627
4	<code>SpaceShower:pTmaxFudge</code>	0.908	0.964	0.968	0.869	0.948	0.943
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.056	1.053	1.481	1.053	1.068
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.128	0.128	0.136	0.128	0.128
7	<code>TimeShower:alphaSvalue</code>	0.128	0.128	0.129	0.126	0.136	0.130
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	2.028	2.175	2.338	1.931	2.080
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.124	0.128	0.135	0.120	0.126
10	<code>BeamRemnants:reconnectRange</code>	1.792	2.047	1.854	1.820	2.404	2.001
Euclidean distance from the expert solution			0.232	0.179	1.076	0.426	0.194

4.6.5 Impact of data pre-processing by filtering on optimal results

In Table 12, we show the number of filtered and unfiltered bins in the A14 and SHERPA datasets that lie within a one σ variance level. A large number of bins within a one σ level indicates smaller deviations of the model from the experimental data. The cumulative distribution plot with the parameters obtained from the robust optimization approach for filtered and unfiltered data for the different categories is shown in Figure 6 (the plots for the other methods are shown in Section 8.9 of the online supplement).

From these results, we observe that there is no significant difference in the number of bins within the one σ variance level between the optimal parameters \mathbf{p}_a^* obtained when all bins were used for tuning and the optimal parameters \mathbf{p}_b^* and \mathbf{p}_o^* obtained when only the bin filtered and observable filtered bins are used for tuning, respectively. Additionally, when comparing across Tables 5-7, we see that in most cases, the results with the observable-filtered data and bin-filtered data provide smaller values in the proposed criteria compared with those using the full dataset. These observations indicate that the MC generator cannot explain the bins removed by the filtering approaches well and that the information contained in these bins does not add significant information to the tune.

4.6.6 Comparison of rational approximation and the MC simulator

Similar to the analysis conducted in Section 4.6.2, we compare the cumulative distribution of bins at different bands of variance levels computed using the approximation model as $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$ and the MC generator model as $\widetilde{r_b(\mathbf{p})} = \frac{(\text{MC}_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta \text{MC}_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$, where \mathbf{p} are the parameters obtained

Table 11: Comparison of the optimal weights obtained by each method using the rational approximation. The observable grouping corresponds to the same grouping as in [3].

	expert	Bilevel- meanscore	Bilevel- medianscore	Bilevel- portfolio	robustopt
Track jet properties					
Charged jet multiplicity (50 distributions)	10	11.41	11.92	11.43	17.85
Charged jet z (50 distributions)	10	11.01	10.00	10.28	0.00
Charged jet p_T^{rel} (50 distributions)	10	9.47	10.20	13.11	1.62
Charged jet $\rho_{ch}(r)$ (50 distributions)	10	10.63	12.72	12.19	0.00
Jet shapes					
Jet shape ρ (59 distributions)	10	12.46	8.49	9.69	17.85
Dijet decorr					
Decorrelation $\Delta\phi$ (Fit range: $\Delta\phi > 0.75$) (9 distributions)	20	18.82	10.32	18.50	15.87
Multijets					
3-to-2 jet ratios (8 distributions)	100	15.06	11.18	11.06	17.85
p_T^Z (Fit range: $p_T^Z < 50\text{GeV}$)					
Z-boson p_T (20 distributions)	10	12.16	11.85	9.25	17.85
Substructure					
Jet mass, $\sqrt{d_{12}}, \sqrt{d_{23}}, \tau_{21}, \tau_{23}$ (36 distributions)	5	10.71	12.75	14.23	17.85
$t\bar{t}$ gap					
Gap fraction vs Q_0, Q_{sum} for $ y < 0.8$	100	24.56	5.05	1.97	17.85
Gap fraction vs Q_0, Q_{sum} for $0.8 < y < 1.5$	80	23.73	47.01	4.01	17.85
Gap fraction vs Q_0, Q_{sum} for $1.5 < y < 2.1$	40	2.39	14.20	7.35	17.85
Gap fraction vs Q_0, Q_{sum} for $ y < 2.1$	10	5.47	19.00	12.82	17.85
Track-jet UE					
Transverse region N_{ch} profiles (5 distributions)	10	13.01	24.18	7.46	17.85
Transverse region mean p_T profiles for $R = 0.2, 0.4, 0.6$ (3 distributions)	10	7.91	16.89	9.68	17.85
$t\bar{t}$ jet shapes					
Jet shapes $\rho(r), \psi(r)$ (20 distributions)	5	10.44	11.47	10.29	15.17
Jet UE					
Transverse, trans-max, trans-min sum p_T incl. profiles (3 distributions)	20	12.11	5.32	10.51	17.85
Transverse, trans-max, trans-min N_{ch} incl. profiles (3 distributions)	20	6.16	14.42	6.56	17.85
Transverse sum E_T incl. profiles (2 distributions)	20	5.11	2.71	7.72	17.85
Transverse sum $ET/\text{sum } p_T$ ratio incl., excl. profiles (2 distributions)	5	11.94	10.81	11.65	17.85
Transverse mean p_T incl. profiles (2 distributions)	10	12.47	7.28	10.45	17.85
Transverse, trans-max, trans-min sum p_T incl. distributions (15 distributions)	1	10.54	14.44	8.27	17.85
Transverse, trans-max, trans-min sum N_{ch} incl. distributions (15 distributions)	1	11.62	10.33	11.48	17.85

Table 12: Number of bins in the A14 and SHERPA datasets within the one σ variance level. Larger numbers are better. The variance level for each bin is calculated as $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$. *Test data type* specifies the data over which $r_b(\mathbf{p})$ is calculated, where *All* means that all bins are used, *Not Filtered* refers to only the bins that remain after filtering, and *Filtered* refers to the bins that were filtered out by the respective filter specified in the *Filtering Method* as well as the envelope filter. For each data type, the number of bins in the corresponding dataset is also specified. *Parameters* specify the type of optimal parameters used in $r_b(\mathbf{p})$ where \mathbf{p}_a^* are the parameters obtained when all bins were used during tuning whereas \mathbf{p}_b^* and \mathbf{p}_o^* are the parameters obtained when only the bin filtered and observable filtered data are used, respectively.

Dataset	Filtering method	Test data type	Parameters	Robust optimization	Bilevel-meanscore	Bilevel-medianscore	Bilevel-portfolio
A14	Bin Filtered	All (# 7010)	\mathbf{p}_a^*	3730	3724	3687	3693
			\mathbf{p}_b^*	3625	3775	3765	3573
		Not filtered (# 5199)	\mathbf{p}_a^*	3350	3317	3265	3273
			\mathbf{p}_b^*	3248	3365	3342	3185
		Filtered (# 1811)	\mathbf{p}_a^*	380	407	422	420
			\mathbf{p}_b^*	377	410	423	388
	Observable Filtered	All (# 7010)	\mathbf{p}_a^*	3730	3724	3687	3693
			\mathbf{p}_o^*	3732	3734	3695	3509
		Not filtered (# 6707)	\mathbf{p}_a^*	3675	3660	3624	3630
			\mathbf{p}_o^*	3679	3672	3629	3444
		Filtered (# 303)	\mathbf{p}_a^*	55	64	63	63
			\mathbf{p}_o^*	53	62	66	65
SHERPA	Bin Filtered	All (# 792)	\mathbf{p}_a^*	320	337	371	256
			\mathbf{p}_b^*	343	328	345	243
		Not filtered (# 588)	\mathbf{p}_a^*	272	283	317	214
			\mathbf{p}_b^*	282	270	292	200
		Filtered (# 204)	\mathbf{p}_a^*	48	54	54	42
			\mathbf{p}_b^*	61	58	53	43
	Observable Filtered	All (# 792)	\mathbf{p}_a^*	320	337	371	256
			\mathbf{p}_o^*	286	348	386	252
		Not filtered (# 727)	\mathbf{p}_a^*	304	319	355	237
			\mathbf{p}_o^*	271	331	370	235
		Filtered (# 65)	\mathbf{p}_a^*	16	18	16	19
			\mathbf{p}_o^*	15	17	16	17

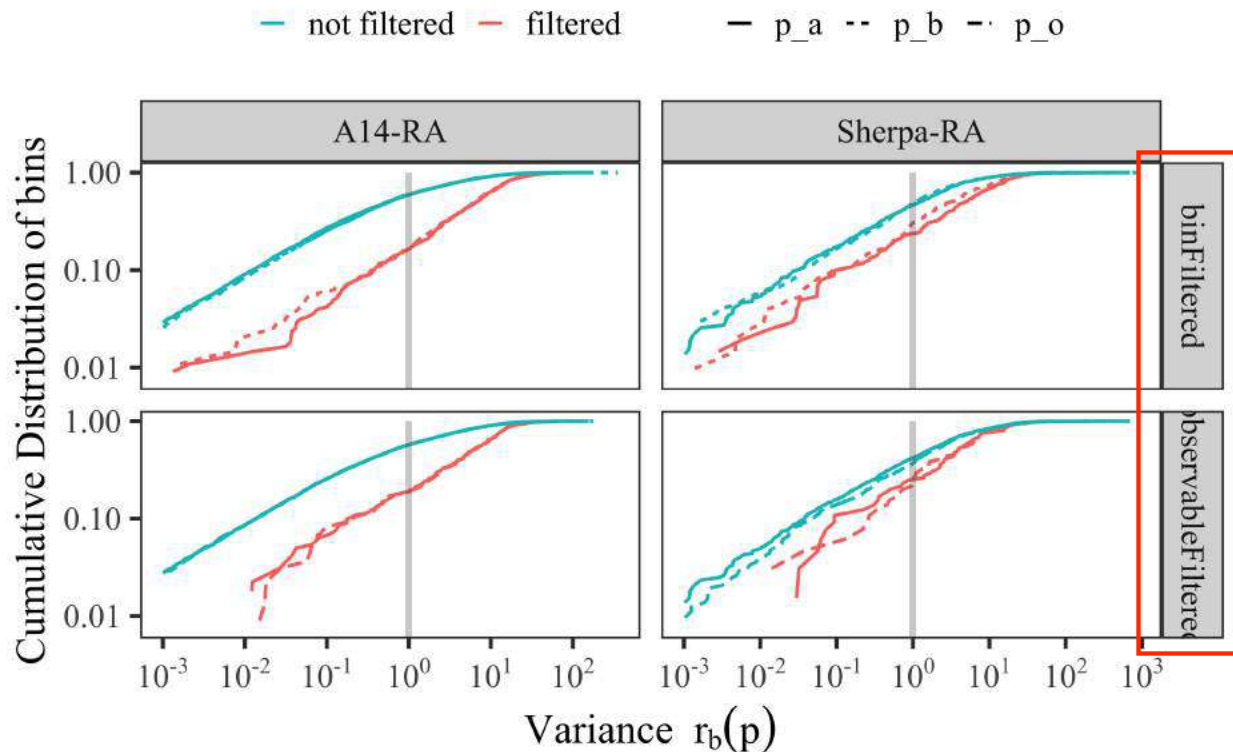


Figure 6: Cumulative distribution of bins remaining after filtering (*not filtered*) and of those filtered out (*filtered*) on the y-axis at different bands of variance levels on the x-axis. The variance level for each bin is calculated as $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$ with parameters \mathbf{p}_a^* , which is obtained when all bins were used, and parameters \mathbf{p}_b^* and \mathbf{p}_o^* , which are obtained when only the bin filtered and observable filtered data are used, respectively.

612 from the tuning approaches. The more bins that are on the bands of variance levels less than
 613 one, the better. Figure 7 shows the plot of this comparison for bins in each category of the A14
 614 dataset.⁵ To avoid making the plot too busy, we show the results using the parameters from three
 615 approaches. A similar plot showing the results with parameters from the remaining approaches is
 616 given in Section 8.10 in the online supplement.

617 We observe in Figure 7 that the *Dijet decorr*, *Jet shapes*, $\widetilde{p_T^Z}$, *Track-jet UE*, and *t \bar{t} gap* categories
 618 show differences in the performance between $r_b(\mathbf{p})$ and $\widetilde{r}_b(\mathbf{p})$ for each approach. Additionally, for

⁵The *Jet UE* comparison is missing from this figure because the internal ATLAS analysis is not available to us.

619 the *robust optimization* and *Bilevel-meanscore* approaches, this difference in the performance is not
 620 as wide as that of the *Expert* (for e.g., see p_T^Z , *Track-jet UE* categories). This suggests that (a)
 621 there are categories where the approximations are not able to capture the MC generator perfectly,
 622 and (b) in general, the rational approximation is a better surrogate for the MC generator than
 623 the polynomial approximation, i.e., the rational approximation gives better predictions of the MC
 624 generator than the polynomial approximation.

Again, I would argue that in a physics paper this would be where we enter some discussion of these observables and their challenges etc.

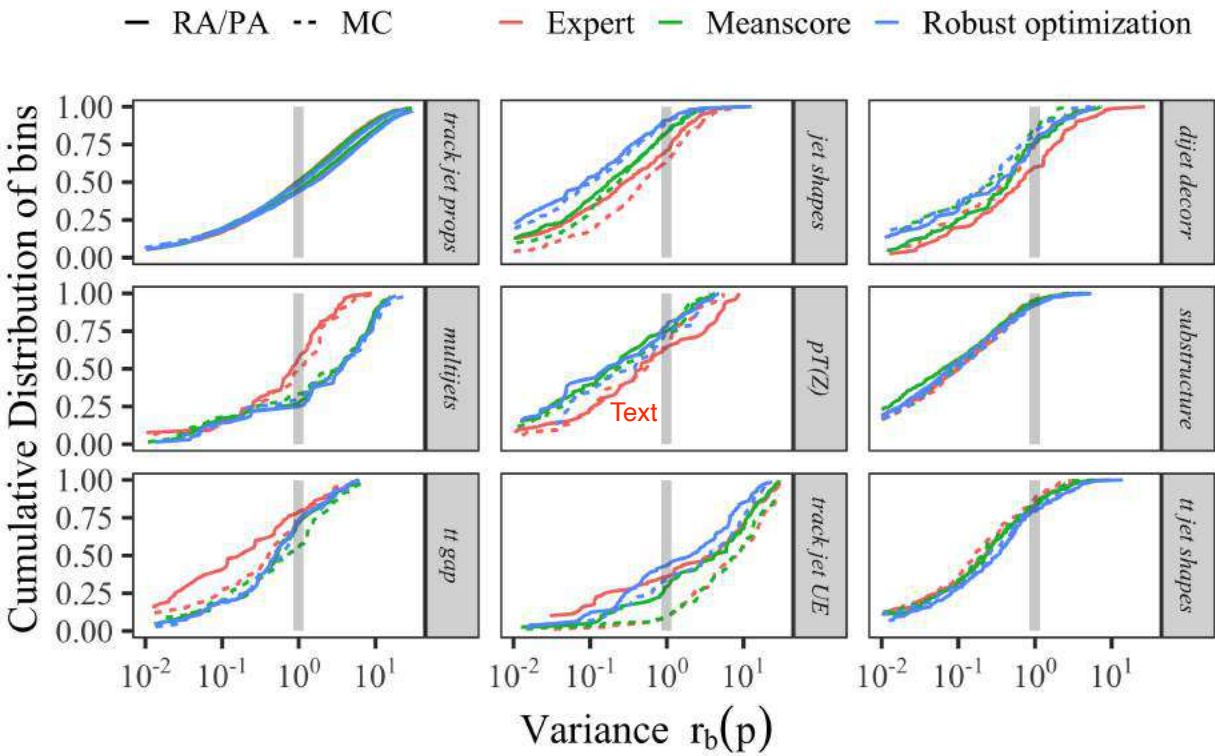


Figure 7: Cumulative distribution of bins (y-axis) in each category of the A14 dataset at different bands of variance levels (x-axis) computed with cubic polynomial approximation (PA) or rational approximation (RA) and the MC simulation.

625 **4.7 Results for the SHERPA dataset**

626 In this section, we present the detailed results for the SHERPA dataset.

627 **4.7.1 Comparison metric outcomes for the SHERPA dataset**

628 Tables 13-15 show the results when using the rational approximation (results for the cubic polyno-
 629 mial approximation are in the online supplement Section 8.12.6). Smaller numbers indicate better
 630 performance. The smallest number of each metric is bold for better visualization. Similar to A14, we
 631 find that the robust optimization approach achieves the best performance in terms of the Weighted
 632 χ^2 criterion. Assigning All-weights-equal to all observables yields the best results in terms of A-
 633 and D-optimality for the full and the bin-filtered dataset. The portfolio approach yields the best
 634 A- and D-optimality values when using the observable-filtered dataset.

635 Compared with the results of A14, we see that the magnitudes of the numbers obtained for the
 636 SHERPA dataset for the Weighted χ^2 , A- and D-optimality criteria are much larger, indicating that
 637 **we are not certain about the optima found by the methods.**

Is this a problem with the global structure of the parameter space?

Table 13: Results for the comparison metrics for the full SHERPA dataset using the rational approximation.

method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.2201	9.0147	-39.3957
Bilevel-medscore	0.2249	43.2031	-25.7164
Bilevel-portfolio	0.1510	11.9869	-35.7488
All-weights-equal	0.2794	6.8428	-42.0325
Robust optimization	0.0603	55.8079	-22.0884

Table 14: Results for comparison metrics for the observable-filtered SHERPA dataset using the rational approximation. Three observables were filtered out and not used during the optimization.

method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.3621	11.1570	-36.5249
Bilevel-medscore	0.2315	13.0679	-35.3498
Bilevel-portfolio	0.4728	8.5578	-38.6042
All-weights-equal	0.4587	59.7043	-19.1257
Robust optimization	0.0509	32.9470	-30.5536

Table 15: Results for the bin-filtered SHERPA dataset using the rational approximation. 204 out of 5426 bins were filtered out and not used during the optimization.

method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1406	16.5417	-33.3334
Bilevel-medscore	0.1352	16.9715	-33.7009
Bilevel-portfolio	0.2792	15.2932	-35.8314
All-weights-equal	0.2105	8.9591	-38.3039
Robust optimization	0.0869	17.4497	-34.0525

638 4.7.2 Comparison of the cumulative distribution of bins at different variance levels

639 Similar to the analysis conducted in Section 4.6.2, we compare the cumulative distribution of bins at
 640 different bands of variance level computed using the optimal parameters \mathbf{p} obtained from the tuning
 641 approaches. Figure 8 shows the plot of this comparison for all bins. The results show that fewer bins
 642 lie within the variance boundary of one when using the parameters of the bilevel-portfolio approach.
 643 On the other hand, the bilevel-medianscore approach finds parameters that yield the most bins at
 644 lower bands of variance levels.

645 4.7.3 Optimal parameter values for the SHERPA dataset with rational approximation

646 The optimal parameter values for the SHERPA dataset are shown in Tables 16, 17 and 18. For a
 647 different visualization of the different solutions obtained with our methods, we illustrate the $[0,1]$ -
 648 scaled optimal parameters in the online supplement Section 8.12.4.

649 We see that most of the parameters are on the boundaries of the parameter space (indicated
 650 in the table in bold), except for `KT_0` and `STRANGE_FRACTION`. This observation indicates that we
 651 might need to change the size of the parameter domain to avoid model extrapolation.

652 Note that for the SHERPA dataset, we do not have an “expert” solution for benchmark compar-
 653 ison. Instead, we compare the solutions to the chosen reasonable default setting. The parameter
 654 range is constructed by multiplying the default value by 0.5 and 1.5 to obtain the lower and the
 655 upper bound of its range respectively, i.e., the default values lie in the middle of the parameter
 656 range. We see that there are differences between the optimal parameters obtained with the different
 657 methods, in particular, bilevel-medianscore gives a very similar solution to the default setting.

658 The distribution of weights from the different methods has a similar pattern as for the tunes
 659 based on the A14 dataset. These patterns are displayed in Fig. ?? in the online supplement. Robust
 660 optimization selects only one of the event shape observables as relevant, while applying the same

Table 16: Optimal parameter values obtained with all methods using rational approximation when no filtering was applied before optimization (88 observables). The parameter values on the boundaries of the parameter space are indicated in bold.

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.888	0.789	0.919	0.909	0.872
2	ALPHA_G	1.25	0.626	1.500	0.626	1.874	0.626
3	ALPHA_L	2.50	3.749	1.890	3.749	1.252	3.749
4	BETA_L	0.10	0.150	0.050	0.087	0.150	0.150
5	GAMMA_L	0.50	0.274	0.339	0.750	0.683	0.293
6	ALPHA_H	2.50	3.400	2.897	1.251	2.841	3.440
7	BETA_H	0.75	0.827	0.536	0.783	0.540	0.795
8	GAMMA_H	0.10	0.148	0.050	0.082	0.150	0.150
9	STRANGE_FRACTION	0.50	0.517	0.498	0.583	0.508	0.546
10	BARYON_FRACTION	0.18	0.100	0.175	0.106	0.136	0.090
11	P_QS_by_P_QQ_norm	0.48	0.720	0.419	0.572	0.613	0.720
12	P_SS_by_P_QQ_norm	0.02	0.010	0.015	0.030	0.030	0.010
13	P_QQ1_by_P_QQ0	1.00	1.499	1.206	0.948	1.190	1.499
Euclidean distance from the default solution			1.513	0.984	1.244	1.289	1.531

Table 17: Optimal parameter values for the SHERPA dataset obtained with all methods using rational approximation after observable filtering (3 observables were filtered out). The parameter values on the boundaries of the parameter space are indicated in bold.

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.867	0.744	0.952	0.876	0.886
2	ALPHA_G	1.25	0.775	0.626	0.626	0.626	0.957
3	ALPHA_L	2.50	3.749	1.252	3.749	1.252	2.424
4	BETA_L	0.10	0.109	0.050	0.050	0.150	0.113
5	GAMMA_L	0.50	0.250	0.437	0.413	0.750	0.460
6	ALPHA_H	2.50	3.053	2.318	1.251	2.826	3.132
7	BETA_H	0.75	0.827	0.625	0.750	0.375	0.969
8	GAMMA_H	0.10	0.050	0.134	0.094	0.050	0.131
9	STRANGE_FRACTION	0.50	0.479	0.580	0.651	0.506	0.511
10	BARYON_FRACTION	0.18	0.270	0.137	0.090	0.137	0.180
11	P_QS_by_P_QQ_norm	0.48	0.720	0.469	0.495	0.470	0.601
12	P_SS_by_P_QQ_norm	0.02	0.010	0.030	0.030	0.030	0.019
13	P_QQ1_by_P_QQ0	1.00	0.500	1.499	1.499	1.499	0.958
Euclidean distance from the default solution			1.408	1.249	1.372	1.446	0.637

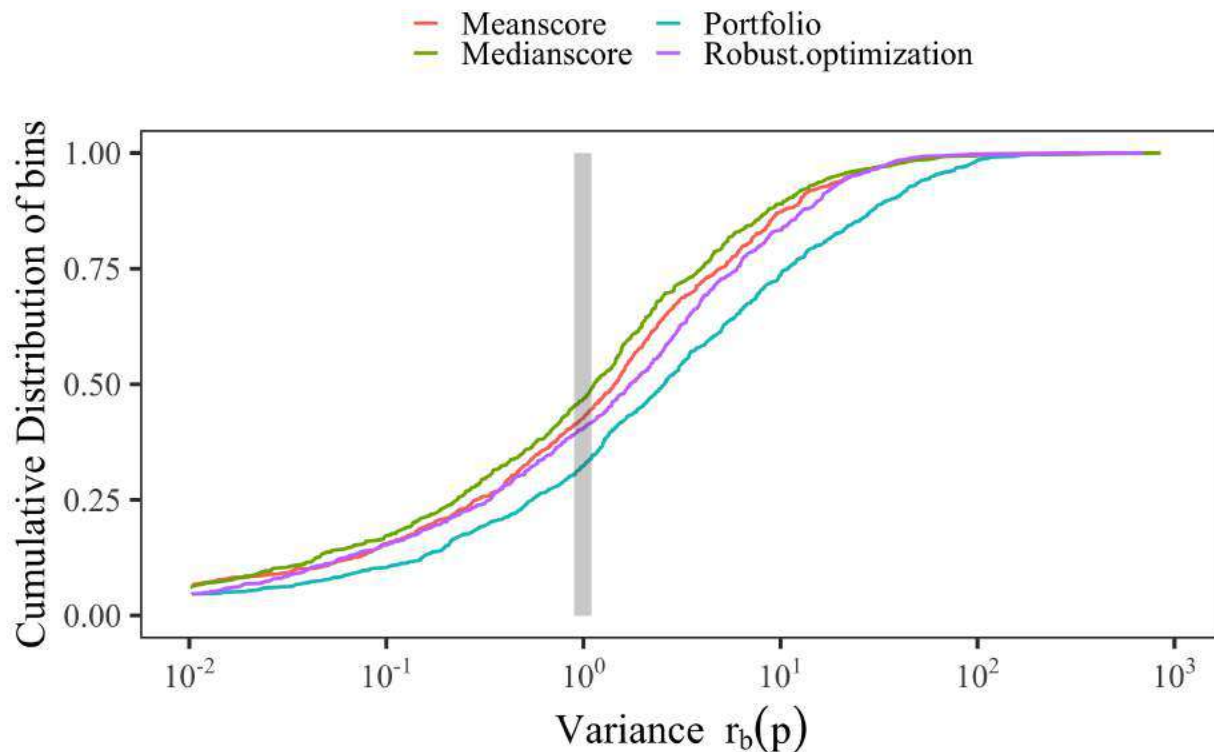


Figure 8: Cumulative distribution of all bins (y-axis) in the SHERPA dataset at different bands of variance levels (x-axis) given by $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$.

661 equal weight to most of the particle multiplicity (one bin) distributions. The other methods have
 662 weights that are more widely distributed among the observables with a small number of weights far
 663 from the average.

664 4.8 A note on computation times

665 The bilevel optimization approaches of medianscore, meanscore, and portfolio are run on a 4-core,
 666 32 GB RAM machine running at 1.1 GHz. For the results of robust optimization presented in this
 667 paper, 100 values for μ are used that are run on 100 threads in parallel on a server with 64 Intel
 668 Xeon Gold CPU cores running at 2.30 GHz. There are two threads per core, but each run of robust
 669 optimization is run on a single thread. Additionally, this server is equipped with 1.5TB DDR4 2666
 670 MHz of memory. A simple comparison to find the best μ takes one minute. The all-weights-equal

Table 18: Optimal parameter values for the SHERPA dataset obtained with all methods using rational approximation after bin-filtering (7 bins were filtered out). The parameter values on the boundaries of the parameter space are indicated in bold.

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.895	0.821	0.948	0.820	0.899
2	ALPHA_G	1.25	0.893	1.483	0.626	1.874	0.626
3	ALPHA_L	2.50	3.749	2.334	2.567	3.749	3.749
4	BETA_L	0.10	0.050	0.150	0.074	0.050	0.067
5	GAMMA_L	0.50	0.390	0.250	0.750	0.250	0.454
6	ALPHA_H	2.50	1.251	3.670	1.251	1.969	1.251
7	BETA_H	0.75	0.715	0.534	0.739	1.125	0.715
8	GAMMA_H	0.10	0.119	0.142	0.105	0.050	0.089
9	STRANGE_FRACTION	0.50	0.556	0.542	0.570	0.531	0.559
10	BARYON_FRACTION	0.18	0.122	0.120	0.124	0.138	0.124
11	P_QS_by_P_QQ_norm	0.48	0.595	0.720	0.492	0.497	0.577
12	P_SS_by_P_QQ_norm	0.02	0.030	0.030	0.030	0.030	0.030
13	P_QQ1_by_P_QQ0	1.00	1.499	1.499	1.499	1.499	1.499
Euclidean distance from the default solution			1.266	1.377	1.201	1.462	1.242

671 approach is run on a 4-core, 32 GB RAM machine running at 1.1 GHz.

672 The time taken by all the tuning approaches for unfiltered (*All data*) as well as for bin filtered
673 and observable filtered A14 data is given in Table 19. In the unfiltered data case, the bilevel
674 optimization approaches of medianscore, meanscore, and portfolio take approximately 14.5 hours
675 and each run (i.e., one μ) of robust optimization takes an average of about 0.8 hours. Since all 100
676 values of μ were run in parallel, the total time to complete all 100 runs of robust optimization is
677 approximately two hours. In comparison, campaigns to tune weights by hand takes many weeks
678 or months. Given our results, we can see that the automated weight adjustment by optimization
679 is significantly faster than hand-tuning. The all-weights-equal approach took less than 10 minutes,
680 however, this approach leads to worse results.

681 The observable filtering method requires a single-tune to obtain the χ^2 values per observable
682 which takes 1647 seconds (0.45 hours) for all observables in the A14 dataset, which is followed by
683 applying the Z-score method to filter out outliers (see Section 3.1) and this takes about 10 seconds.
684 Once the single-tune to obtain the χ^2 values per observable is performed, the bin-filtering method
685 takes an additional 300 seconds to filter out the bins from the A14 dataset. Thus, the total pre-
686 processing time required for observable filtering is 1657 seconds (0.46 hours) and for bin-filtering is
687 1947 seconds (0.54 hours).

688 From Table 19, we observe that the time taken to tune parameters in the observable filtered
689 and bin filtered data case is significantly smaller than for the unfiltered data case. For the bilevel
690 optimization approaches, the time required per iteration for the observable- and bin-filtered cases

691 is 6%, and 55% less, respectively, and for each run of robust optimization, it is 9% and 36% less,
 692 respectively. Also, the overhead of performing observable and bin filtering is small compared to
 693 the time it takes to tune parameters. Since the results from Section 4.6.5 show that the bins
 694 filtered by bin and observable filtering do not add significant information to the tune, we can claim
 695 that using filtered data provides a significant improvement in compute-time performance for tuning
 696 parameters.

Table 19: CPU time (in seconds) and time per iteration (in seconds) taken by all approaches when using all, the observable-filtered, and the bin-filtered A14 data. The robust optimization approach converges after 69, 105, and 83 iterations, respectively. The bilevel-medianscore, -meanscore, and -portfolio approaches are all run for 1000 iterations.

Method	All data		Bin filtered		Observable filtered	
	CPU time	Time per iteration	CPU time	Time per iteration	CPU time	Time per iteration
Robust optimization	3035	44	2989	28	3327	40
Bilevel-medianscore	52326	52	23600	24	49057	49
Bilevel-meanscore	52169	52	23600	24	49018	49
Bilevel-portfolio	52366	52	23609	24	49084	49

697 5 Eigentunes

698 We use the eigentune approach to calculate confidence intervals for the optimal parameters. We
 699 note that the A- and D-optimality criteria provide the size of confidence ellipsoid around the optimal
 700 parameters. Here, we expand this information by scanning generator parameters along the principal
 701 axes of this ellipsoid. Details of this method are described in [4] and a similar approach is used in
 702 estimating the uncertainties of predictions from the parton distribution functions [42]. The interval
 703 defines a boundary beyond which the value of the objective function is larger than the objective
 704 function value at the minimum by a criterion. The criterion is normally chosen to be the number
 705 of degrees of freedom n , which is defined as the total number of bins of all observables minus the
 706 number of generator parameters, d , i.e., $n = \sum_{\mathcal{O} \in \mathcal{S}_O} |\mathcal{O}| - d$. However, to properly take into account
 707 the weights assigned to observables, we use the scaled effective sample size as the criteria, which is

708 calculated as follows:

$$n = \gamma \times \left(\frac{(\sum_i w_i)^2}{\sum_i w_i^2} - d \right)$$

709 The weights are normalized so that the sum of weights associated with all observables equals one.
 710 γ is iteratively tuned and chosen to be 0.01. The interval would represent the uncertainties of
 711 the parameters assuming that the objective function follows a χ^2 distribution. Smaller intervals
 712 associated with the tuned parameters indicate that the parameters are better constrained by the
 713 experimental data.

714 Given the non-linearity of the objective function and parameter correlations, a reliable approach
 715 to find the 68% confidence interval is to evaluate the objective function for all possible parameter
 716 values. However, this poses a computational challenge. Instead, we project the multidimensional
 717 parameter space into two directions defined by the eigenvectors $u_{1,2}$ associated with the largest
 718 and smallest eigenvalues of the covariance matrix of the parameters, which are calculated using the
 719 inverse of Eq. (16). Then we find an offset α such that the sum of all χ^2 satisfies

$$\chi^2(\mathbf{p}'_{1,2}) = \chi^2(\mathbf{p}^*) + n \quad (17)$$

720 where $\mathbf{p}'_{1,2} = \mathbf{p}^* \pm u_{1,2} \times \alpha$. For each eigenvector, we obtain two vectors \mathbf{p}' from Eq. (17). Finally, the
 721 procedure results in a matrix of sizes of 4 times d . Each column represents a generator parameter;
 722 the minimum and maximum in each column are used to define the eigentune as shown in Tables 20
 723 and 21 for the A14 and the SHERPA dataset, respectively, using the rational approximation. The
 724 same surrogate model is used for all methods. It is possible that the determined intervals go beyond
 725 the predefined parameter range. In this case, the MC predictions are extrapolated by the surrogate
 726 model. When the lower part of the interval goes negative, we force the value to be zero.

727 For the A14 data, different optimization methods result in similar intervals for all paramete-
 728 ters. The beam remnants (e.g. `BeamRemnants:reconnectRange`) and space-like showering param-
 729 eters (e.g. `SpaceShower:pT0Ref`) are better constrained; their intervals are within 1% of their
 730 optimized parameters. However, the strong coupling constant α_s in hard scattering processes
 731 (`SigmaProcess:alphaSvalue`) and time-like showering (`TimeShower:alphaSvalue`) are less con-
 732 strained. Again a physics question, what does alphaSvalue mean here? It sounds a little weird...

733 For the SHERPA data, different optimization methods produce quite different intervals. Overall,
 734 the bilevel-meanscore method results in relatively small intervals for all parameters. The heavy
 735 quark fragmentation parameters (e.g. `ALPHA_H`) are well-constrained thanks to the B -hadron frag-
 736 mentation measurements, but the light quark fragmentation parameters are not.

Table 20: Eigentune results for the A14 data using the rational approximation for different optimization methods.

Parameters	Expert		Bilevel-meanscore		Bilevel-mediansocre		Bilevel-portfolio		Robust optimization	
	min	max	min	max	min	max	min	max	min	max
SigmaProcess:alphaSvalue	0.075	0.193	0.079	0.192	0.079	0.190	0.074	0.195	0.085	0.183
BeamRemnants:primordialKThard	1.903	1.906	1.805	1.910	1.674	1.769	1.744	1.850	1.876	1.892
SpaceShower:pT0Ref	1.636	1.653	1.516	1.547	1.142	1.228	1.298	1.344	1.586	1.591
SpaceShower:pTmaxFudge	0.905	0.912	1.012	1.016	1.069	1.096	1.037	1.046	1.025	1.026
SpaceShower:pTdampFudge	1.044	1.048	1.064	1.076	1.082	1.086	1.058	1.064	1.078	1.091
SpaceShower:alphaSvalue	0.121	0.124	0.125	0.131	0.127	0.130	0.124	0.133	0.123	0.129
TimeShower:alphaSvalue	0.043	0.197	0.044	0.192	0.039	0.213	0.030	0.213	0.051	0.198
MultipartonInteractions:pT0Ref	1.665	2.543	1.649	2.562	1.780	1.979	1.160	2.829	1.461	2.528
MultipartonInteractions:alphaSvalue	0.068	0.177	0.072	0.161	0.115	0.121	0.062	0.186	0.094	0.151
BeamRemnants:reconnectRange	1.788	1.795	2.065	2.105	1.912	1.915	1.972	2.000	2.589	2.618

Table 21: Eigentune results for the SHERPA data using the rational approximation for different optimization methods. Parameters with negative values are set to zero.

Parameters	Bilevel-meanscore		Bilevel-mediansocre		Bilevel-portfolio		Robust optimization	
	min	max	min	max	min	max	min	max
KT_0	0.815	0.970	0.688	0.957	0.524	1.254	0.491	1.273
ALPHA_G	0.438	0.792	1.325	1.604	0.571	0.691	1.597	2.115
ALPHA_L	3.683	3.824	1.309	2.863	3.525	3.939	0.291	2.088
BETA_L	0	0.460	0.043	0.062	0	0.440	0	0.387
GAMMA_L	0.175	0.362	0.330	0.352	0.688	0.823	0.220	1.087
ALPHA_H	3.245	3.537	2.843	2.988	1.200	1.311	2.289	3.475
BETA_H	0.747	0.898	0.484	0.585	0.623	0.972	0.350	0.759
GAMMA_H	0.059	0.249	0	0.080	0.013	0.133	0	0.469
STRANGE_FRACTION	0.496	0.556	0.395	0.595	0.415	0.706	0.440	0.567
BARYON_FRACTION	0	0.459	0.129	0.218	0.018	0.170	0	0.342
P_QS_by_P_QQ_norm	0.552	0.809	0.319	0.524	0.552	0.588	0.594	0.629
P_SS_by_P_QQ_norm	0.	0.031	0.	0.103	0	0.081	0	0.068
P_QQ1_by_P_QQ0	1.492	1.512	1.202	1.210	0.945	0.952	1.167	1.210

6 Discussion

The results presented in the previous sections demonstrate that automated tuning methods can produce better fits of the generator predictions to data. Several figures of merit for comparing different tunes were considered. The automation of the process means that tuning can be performed in less time and with less subjective bias. In this section, we discuss the physics impact of various tuning results.

6.1 Implications of our results on physics

Physics event generators are imperfect tools. They contain a mixture of solid physics predictions, approximations, and *ad hoc* models. The approximations and models are expected to be incomplete, and thus are unlikely to describe the full range of observables accessible by the experiment. Despite this fact, for a certain choice of parameters, a model may be able to describe part of the data. This agreement would be accidental and would likely compromise predictions of this model for different parts of the data. The weighting of data by an expert is a primitive attempt to force the model to agree with data in a region of interest to the physicist – which, most of the time, corresponds to a region where a model should be applied. It is equivalent to adding a large systematic uncertainty to the data that is de-emphasized by the weighting.

The physics discussion is coming too late for the typical SciPost reader. Expand and integrate in the text?

Here, we address whether the automated methods accomplish this weighting of data without explicit input from the physicist. First, we should state our expectations for a tune to the A14 dataset. The features of the expert tune were previously discussed in Section. 2.2.1 of the A14 publication [3]. The A14 data is all of interest to the physicist, but some of those observables are expected *a priori* to be described better by the event generator than others. The parton shower and hadronization model are expected to describe well *Tracked jet properties* and *Jet shapes*. The description of jets is essential for all hadron collider analyses and is the *raison d'être* for event generators. *t \bar{t} jet shapes* emphasize the final state parton shower, and is critical to be described well when making precision predictions that are sensitive to the top quark mass. *Dijet decorr* and p_T^Z observables provide constraints on initial state parton shower and intrinsic transverse momentum parameters free from most other parameters, and are generically important to be described well. Additional properties, such as the number of jets produced in di-jet or *Z* events or the production of jets at extreme angles, are beyond the scope of the PYTHIA predictions. *Track-jet UE* and *Jet UE* observables are sensitive to PYTHIA's multi-parton-interaction model, which describes most of the particles produced in a high-energy collision. The addition of *Multijets* observables is biasing the parton shower to describe a next-to-leading order observable, while the leading-logarithm par-

769 ton shower includes only an approximation to the full result. Experience shows that this biasing
 770 provides a globally better description of many observables of interest to the physicist with little
 771 effort and without significantly impacting other predictions. This feature was built into the *Expert*
 772 tune by applying a large weight to this dataset. Finally, adding the $t\bar{t}$ *gap* category is asking for
 773 the description of an exclusive observable, which has very strong requirements in its construction,
 774 whereas the PYTHIA prediction here is valid for more inclusive observables. Including this data
 775 in the tune is a very specific physics requirement that may be beyond the scope of the PYTHIA
 776 approximations.

777 6.2 Observables with improved descriptions

778 Examples of observable predictions with a lower $\chi^2_{\mathcal{O}}$ value than the *expert* tune are displayed in
 779 Figures 9a-9c. These reflect an improvement in a class of observables and are indicative of all the
 780 comparisons between predictions and data.

781 All of our methods produce a better description of the data than the expert tune for the category
 782 *Jet shapes*, though the expert prediction is mainly differing in only the first bin. This observable
 783 is expected to be described well, in general, since it lies in a physics regime compatible with the
 784 PYTHIA approximations.

785 The predictions for the p_T^Z and *Dijet decorr* categories are also improved. We note that the
 786 weights found for these analyses are not substantially different than for the expert tune, but that
 787 other categories have their weights reduced (see Table 11 for reference). This implies some tension
 788 between these observables and the *Multijets* category (to be discussed below).

789 The comparisons between predictions and data shown in our figures are based on runs of the
 790 MC event generator for the parameter values derived using the surrogate model. Before continuing,
 791 we should comment on the differences in Figure 7 (and in Figure 17 in Section 8.9 of the online
 792 supplement) between the surrogate model (RA) and explicit runs of the event generator (MC)
 793 at the output tuned parameters. The surrogate model would be unreliable if the output tune
 794 parameters were outside or near the boundary of the parameter range used to derive the inputs for
 795 the surrogate. A comparison of the parameter values relative to the expert tune and Figure 18 shows
 796 the distribution of parameter values normalized to the sampling range: $r_{\text{param}} = \frac{p - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}}$. All
 797 of the central values for the parameters are well within the sampling range. Only the parameters
 798 `SpaceShower:pTdampFudge` and `BeamRemnants:reconnectRange` come near the boundaries. For
 799 the former, the minimum sampling value was 1.0, and the tuning results only indicate that this
 800 parameter should be near 1.0. For the latter, the maximum sampling value was chosen quite large
 801 so that all results appear to be close to the minimum value.

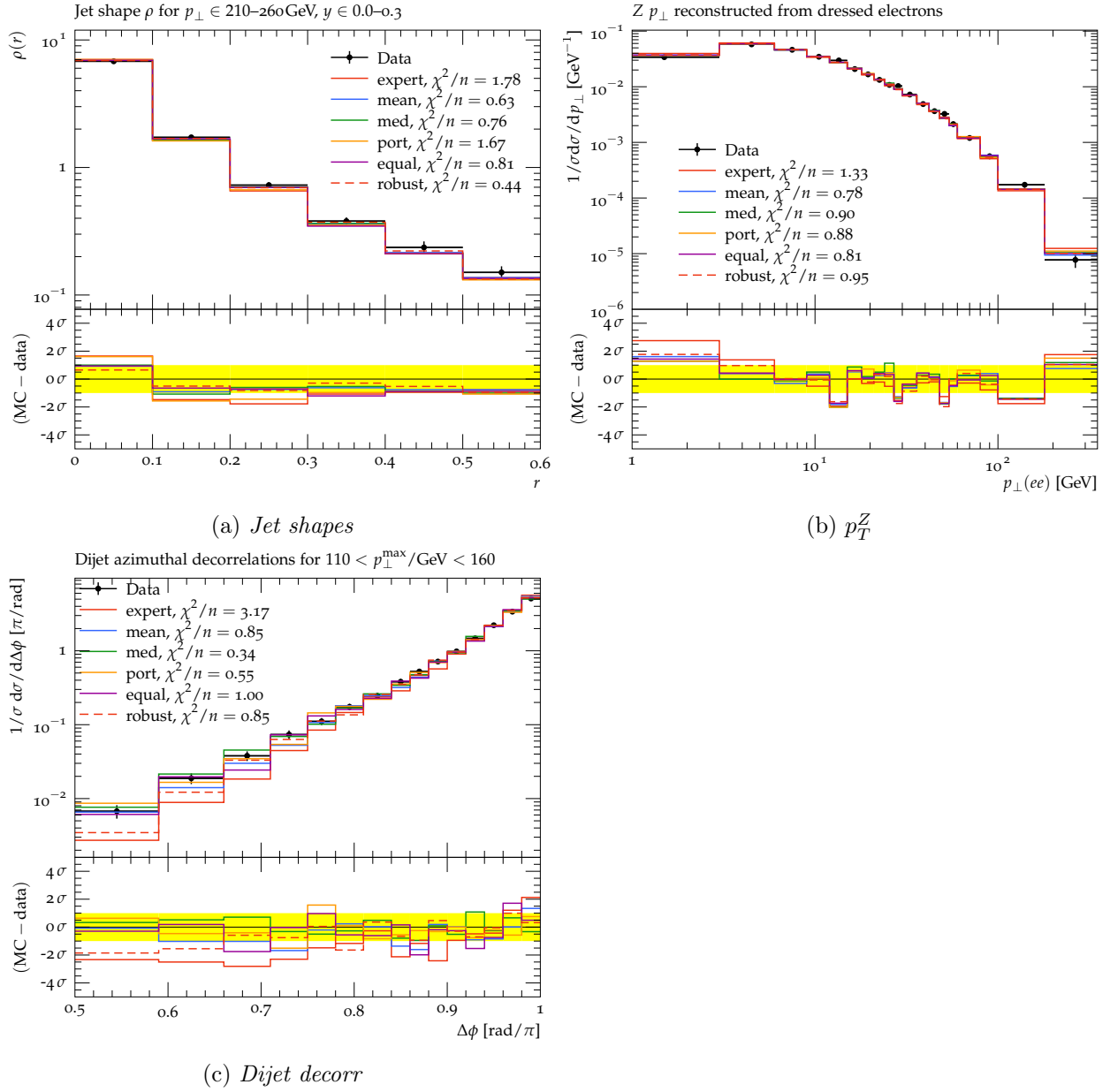


Figure 9: Examples of A14 observables and their $\chi^2_{\mathcal{O}}$ values for which the automated tuning leads to better fits than the expert's hand tuning.

802 Furthermore, the most noticeable differences between the RA surrogate predictions and MC
803 occur for rather small values of the variance between the data and predictions. These values have
804 a negligible impact on the full χ^2 , and are within the expected range of validity of the surrogate
805 model.

806 6.3 Observables with worse descriptions

807 The predictions for *Track jet properties* and *Substructure* are not significantly improved, but also
808 not degraded. Most of the observables in these categories were designed to tune and test the
809 multi-parton interaction model, and thus it is no surprise that they are described well.

810 Two categories stand out as being better described by the expert tune. These are the *Mul-*
811 *tijets* and *t \bar{t} gap* categories that were given a particularly large weight in the expert tune. Some
812 examples can be seen in Figure 10a-10c. It is no surprise that these categories are not described
813 well as the expert tune. It is surprising that the parameters sensitive to this observable, namely
814 `TimeShower:alphaSvalue` and `SpaceShower:alphaSvalue` are actually somewhat larger than the
815 expert tune values, see Table 8. Larger values for these parameters should mean forcing the predic-
816 tion to look *more* like a higher-order calculation. Clearly, other data, such as *Dijet decorr* and p_T^Z
817 prefer larger values for these parameters than the *Multijets* category alone.

818 It is worth noting that without expert input our automated methods do not emphasize these
819 observables. However, the results of the eigentunes are that the “error bands” for these parameters
820 contain the values for the A14 tune. It is understood *a priori* that the physics models inside PYTHIA
821 behind the description of *Multijets* and *t \bar{t} gap* need theory corrections. **In fact, even at the time**
822 **of the A14 tune, methods existed to better describe the *Multijets* category using PYTHIA, but**
823 **it requires a process-dependent correction.** The explicit aim of the A14 tune was to be process-
824 agnostic and applied to physics predictions for which the corrections were not readily available or
825 easily applicable. However, if the goal is to provide a tune that can be used in association with
826 process-dependent corrections, then those provided in this study are more appropriate.

Please say it,
rather than
talking around
it...

827 6.4 Results for SHERPA tuning

828 Some of the results of the SHERPA tuning are shown in Figure 11. In general, all of the parameter
829 selection methods applied here yield an improved global $\chi^2_{\mathcal{O}}$ over the default values. The param-
830 eters varied in this tuning exercise are all related to the formation of physical particles. This is
831 a phenomenon that occurs at a low-energy scale and cannot be described realistically (currently,
832 at least) from theory. The model employed in SHERPA is a cluster model that fissions colorless
833 blobs of energy into particles using a parameterized probability distribution. Despite the fact that

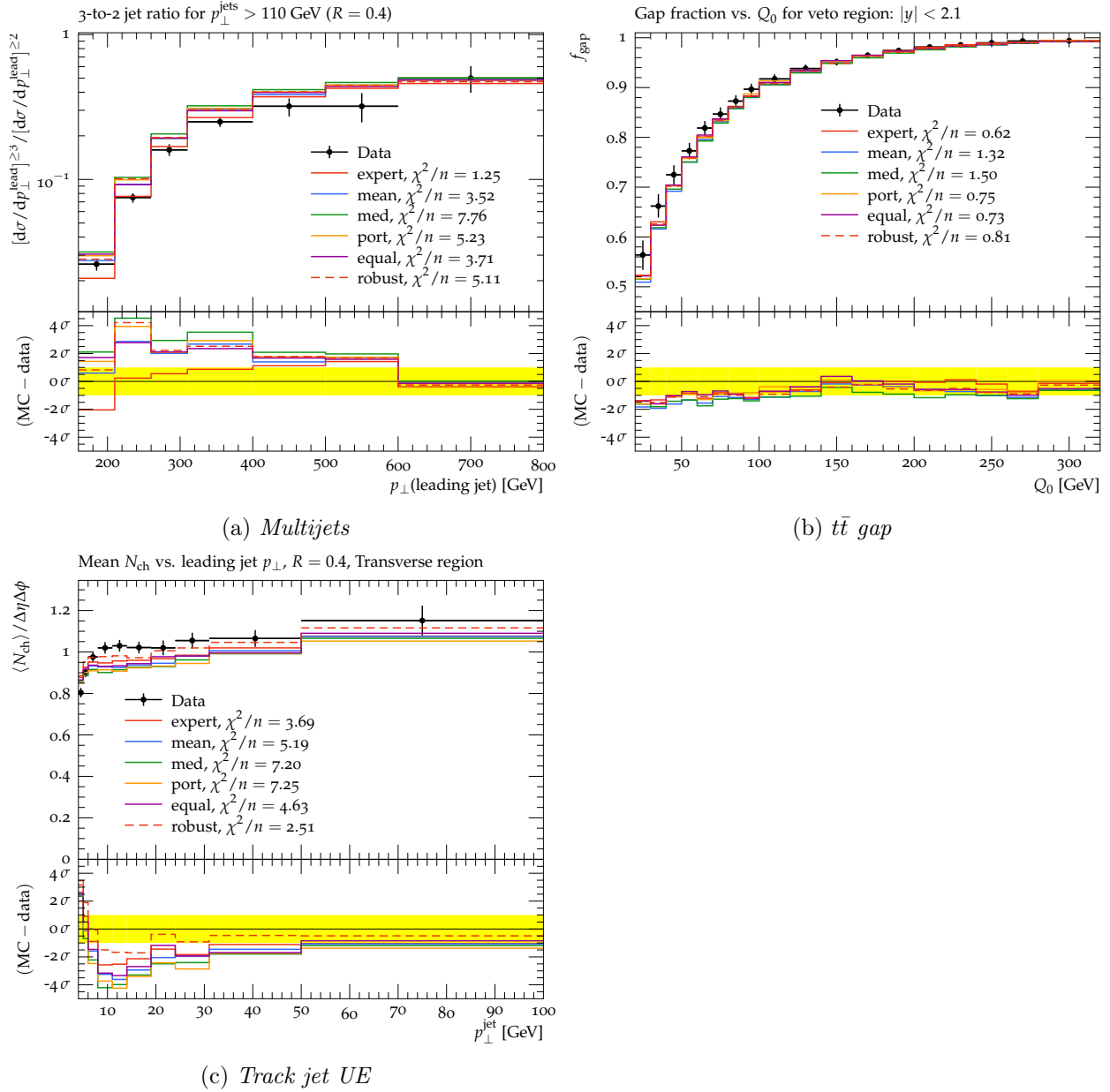
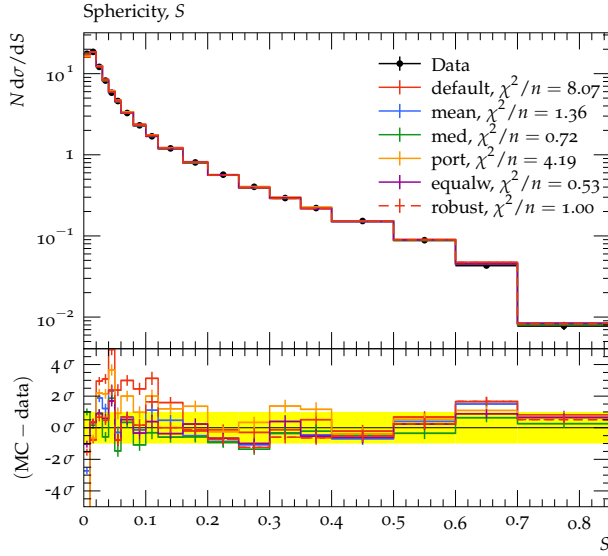
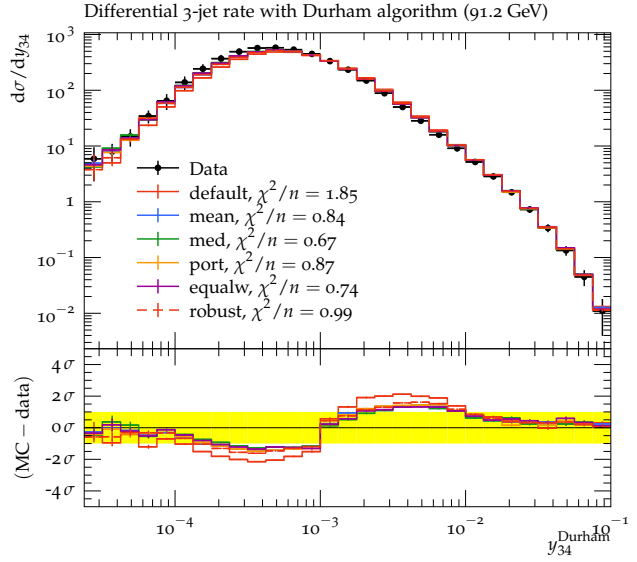


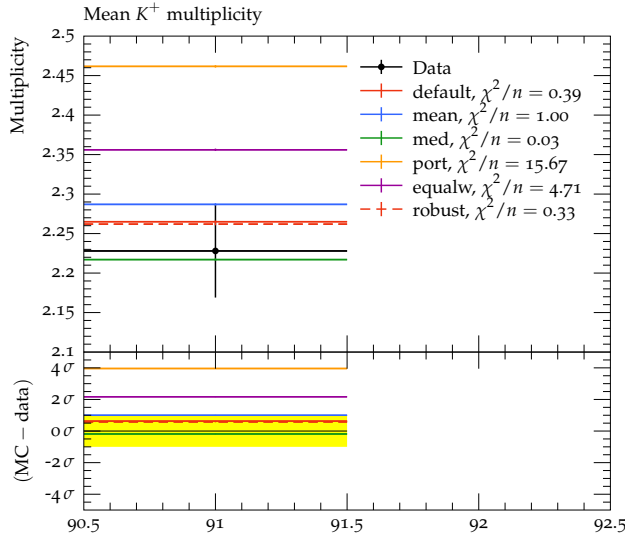
Figure 10: Examples of A14 observables and their $\chi^2_{\mathcal{O}}$ values for which the automated tuning approach performs worse than the expert's hand tuning.



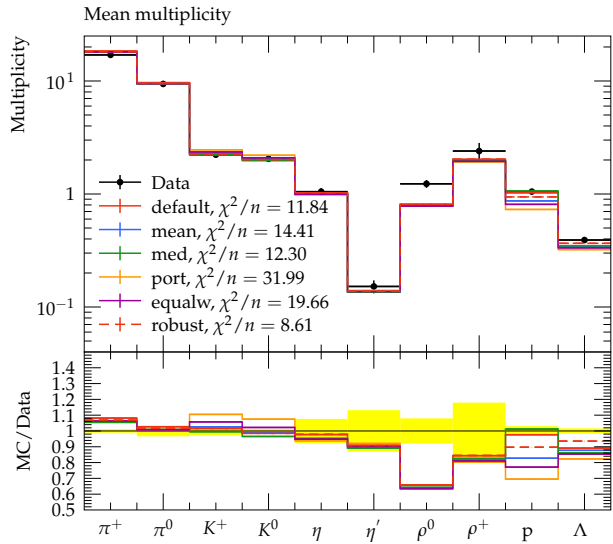
(a) Jet shapes



(b) Jet rates



(c) Particle count



(d) Several particle counts side-by-side

Figure 11: Examples of histogram plots of the $\chi^2_{\mathcal{O}}$ values for the SHERPA tune.

834 hadronization occurs at a low-energy scale, it has an impact on observables that are used to test
835 perturbative predictions at relatively high-energy scales. For these observables, it is impossible to
836 entirely disentangle the perturbative prediction from the non-perturbative hadronization model pre-
837 diction. Figures 11a-11b show comparisons of our tunes to the default, demonstrating a significant
838 improvement in most cases. Figure 11c shows mixed results for the production of one particular
839 species of particle. Figure 11d is an example of an inclusive observable that counts the number of
840 particles produced without any direct reference to their energy or position in the detector.

841 All of these results are for a certain precision of perturbation theory. There are both technical
842 and mathematical reasons to truncate perturbation theory in a certain order. These calculations
843 were based on the lowest order perturbation theory with an improved parton shower approximation
844 to simulate additional perturbative effects. The lowest order prediction produces 2 jets using exact
845 perturbation theory and any additional jets using the parton shower approximation. Figure 11b is
846 an observable that counts the number of 3-jet events as a function of the jet definition. While our
847 results are improved over the default, this indicates higher-order perturbative calculations might
848 improve the description even more (e.g., 3 jets calculated in exact perturbation theory and 4 or
849 more jets from the parton shower approximation).

850 Table 16 shows the parameters values for the various tunes. The simplest comparison is between
851 the default values and “All-weights-equal.” The all-weights-equal method yields the tune that would
852 result if only the data considered in this study were used. One result is that several of the parameters
853 take on the extremum of the values considered here. Without any additional direction to choose
854 the range for our parameter scan, we chose 1/2 of the default value to define our sampling window.

855 One surprising result is that the parameter `P_QQ1_by_P_QQ0`, which represents the ratio of spin-1
856 to spin-0 diquarks, is driven to a value > 1 . While there is no obvious reason that the cluster model
857 breaks down, spin-1 diquark production is usually expected to be suppressed. The fact that the
858 parameter `BARYON_FRACTION` is driven to its minimal value compensates for this large value.

859 While the type of large scale parameter tuning we have in mind here can only be performed prac-
860 tically using surrogate models, the fact that some tuned parameters are pushed to the boundaries
861 suggests another direction of algorithmic development. In particular, we would like our algorithm
862 to have the capability to recognize a trust region and update the surrogate model with dedicated
863 simulations when necessary.

So here is one thing I am missing - we have these eigentunes, so the authors could take some kind of
process and observable not in the tune and illustrate what we can learn from the new tunes in terms
of uncertainties. Or not? That is what I would have expected as the outcome of the study...

864 7 Conclusions

865 In this paper, we propose several algorithms for automating weighting the importance of data used
866 in the tuning process for Monte Carlo event generators. We performed two studies. The first used
867 particle collider data and predictions are from the Large Hadron Collider (LHC) and had an *expert*
868 selection of analysis weights as a benchmark. The second used data and predictions are from the
869 Large Electron-Positron (LEP) Collider and had only the default parameter choices as a reference.
870 The algorithms considered included a bilevel optimization based on several scoring procedures and
871 a single-level robust optimization. We find that our automatic methods produce parameter tunes
872 that are comparable to labor-intensive, by-hand tunes. For the LHC tuning, filtering of hard-to-
873 describe observables can lead to tunes of superior quality by identifying observables or subsets of
874 observables that cannot be described by the event generator. For the LEP tuning, many of the
875 tuned parameters were driven to the extremum of our sampling range, suggesting that the current
876 models are missing some important physics.

877 First, the results show that the parameter values we found agree with and have the potential
878 to improve the physicists' hand-tuned results. Second, since we automate the weight adjustment
879 for the tune-relevant observables, physicists do not need to hand-tune the weights for observables
880 anymore; we propose several methods for adjusting the weights, so physicists are not involved in
881 the subjective re-weighting anymore. Third, by filtering out and excluding observables and bins,
882 we can save computational time during optimization and improve the parameter values. Fourth, we
883 derived new metrics to easily compare different tunes, and it shows that our methods can perform
884 better than the physicists' hand-tuning approach.

885 For the SHERPA data, most of the optimal parameters are on the boundaries of the parameter
886 space, indicating that we might need to change the size of the parameter domain to avoid model
887 extrapolation. One possible solution to this problem is to build an outer loop for moving the center
888 of the parameter search space and apply the trust region method. We leave this to future research.

889 In this work, we assume that each bin is completely independent of all the other bins. Taking
890 into account bin correlations will be left for future research work.

891 Acknowledgements

892 SM thanks Stefan Hoeche for discussions about our SHERPA results.

893 **Funding information** This work was supported by the U.S. Department of Energy, Office of
894 Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. Support
895 for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC)
896 program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing
897 Research. This work was also supported by the U.S. Department of Energy through grant DE-
898 FG02-05ER25694, and by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359
899 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This work
900 was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Sci-
901 entific Computing Research and Office of Nuclear Physics, SciDAC program through the FASTMath
902 Institute under Contract No. DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory.

903 References

- 904 [1] A. Buckley *et al.*, *General-purpose event generators for LHC physics*, Phys. Rept. **504**, 145
905 (2011), doi:10.1016/j.physrep.2011.03.005, 1101.2599.
- 906 [2] P. Skands, S. Carrazza and J. Rojo, *Tuning PYTHIA 8.1: the Monash 2013 Tune*, Eur. Phys.
907 J. **C74**(8), 3024 (2014), doi:10.1140/epjc/s10052-014-3024-y, 1404.5630.
- 908 [3] ATLAS Collaboration, *ATLAS PYTHIA 8 tunes to 7 TeV data*, Tech. Rep. ATL-PHYS-PUB-
909 2014-021, CERN, Geneva (2014).
- 910 [4] A. Buckley, H. Hoeth, H. Lacker, H. Schulz and J. von Seggern, *Systematic event generator*
911 *tuning for the LHC*, The European Physical Journal C **65**, 331 (2010), doi:10.1140/epjc/s10052-
912 009-1196-7.
- 913 [5] J. Bellm and L. Gellersen, *High dimensional parameter tuning for event generators*, Eur. Phys.
914 J. C **80**(1), 54 (2020), doi:10.1140/epjc/s10052-019-7579-5, 1908.10811.
- 915 [6] P. Ilten, M. Williams and Y. Yang, *Event generator tuning using Bayesian optimization*, JINST
916 **12**(04), P04028 (2017), doi:10.1088/1748-0221/12/04/P04028, 1610.08328.
- 917 [7] Y. Chen and M. Florian, *The nonlinear bilevel programming problem: Formulations, regularity*
918 *and optimality conditions*, Optimization **32**, 193 (1995).
- 919 [8] P. Marcotte and D. L. Zhu, *Exact and inexact penalty methods for the generalized bilevel*
920 *programming problem*, Mathematical Programming **74**(2), 141 (1996).

- 921 [9] J. J. Ye, D. L. Zhu and Q. J. Zhu, *Exact penalization and necessary optimality conditions for*
922 *generalized bilevel programming problems*, SIAM J. Optimization **7**(2), 481 (1997).
- 923 [10] B. Colson, P. Marcotte and G. Savard, *An overview of bilevel optimization*, Annals of operations
924 research **153**(1), 235 (2007).
- 925 [11] J. F. Bard, *Practical bilevel optimization: algorithms and applications*, vol. 30, Springer Science
926 & Business Media (2013).
- 927 [12] H. Markowitz, *Portfolio selection*, The Journal of Finance **7**, 77 (1952), doi:10.2307/2975974.
- 928 [13] T. Gneiting and A. Raftery, *Strictly proper scoring rules, prediction, and estimation*, Journal
929 of the American Statistical Association **102**(477), 359 (2007).
- 930 [14] M. Powell, *Advances in Numerical Analysis, vol. 2: wavelets, subdivision algorithms and radial*
931 *basis functions. Oxford University Press, Oxford, pp. 105-210*, chap. The Theory of Radial
932 Basis Function Approximation in 1990, Oxford University Press, London (1992).
- 933 [15] J. Müller and C. Shoemaker, *Influence of ensemble surrogate models and sampling strategy on*
934 *the solution quality of algorithms for computationally expensive black-box global optimization*
935 *problems*, Journal of Global Optimization **60**, 123 (2014).
- 936 [16] J. Müller and J. Woodbury, *GOSAC: global optimization with surrogate approximation of*
937 *constraints*, Journal of Global Optimization doi:10.1007/s10898-017-0496-y (2017).
- 938 [17] D. M. Hawkins, *Identification of outliers*, vol. 11, Springer (1980).
- 939 [18] C. Croarkin, P. Tobias, J. Filliben, B. Hembree, W. Guthrie *et al.*, *NIST/SEMATECH e-*
940 *handbook of statistical methods*, NIST/SEMATECH, July. Available online: [http://www.itl.](http://www.itl.nist.gov/div898/handbook)
941 [nist.gov/div898/handbook](http://www.itl.nist.gov/div898/handbook) (2006).
- 942 [19] G. Upton and I. Cook, *Understanding statistics*, Oxford University Press (1996).
- 943 [20] B. Rosner, *Percentage points for a generalized esd many-outlier procedure*, Technometrics
944 **25**(2), 165 (1983).
- 945 [21] F. E. Grubbs, *Procedures for detecting outlying observations in samples*, Technometrics **11**(1),
946 1 (1969).
- 947 [22] W. Stefansky, *Rejecting outliers in factorial designs*, Technometrics **14**(2), 469 (1972).

- 948 [23] W. Dixon, *Processing data for outliers*, Biometrics **9**(1), 74 (1953).
- 949 [24] R. Thompson, *A note on restricted maximum likelihood estimation with an alternative outlier*
950 *model*, Journal of the Royal Statistical Society: Series B (Methodological) **47**(1), 53 (1985).
- 951 [25] C. Dardis, *Peirce's criterion for the rejection of non-normal outliers; defining the range of*
952 *applicability*, J Stat Softw **10**, 1 (2004).
- 953 [26] G. L. Tietjen and R. H. Moore, *Some Grubbs-type statistics for the detection of several outliers*,
954 Technometrics **14**(3), 583 (1972).
- 955 [27] O. Etzioni, *Hypothesis filtering: A practical approach to reliable learning*, In J. Laird, ed.,
956 *Machine Learning Proceedings 1988*, pp. 416 – 429. Morgan Kaufmann, San Francisco (CA),
957 ISBN 978-0-934613-64-4, doi:<https://doi.org/10.1016/B978-0-934613-64-4.50047-5> (1988).
- 958 [28] W. G. Cochran, *The χ^2 test of goodness of fit*, Ann. Math. Statist. **23**(3), 315 (1952),
959 doi:10.1214/aoms/1177729380.
- 960 [29] A. Shapiro, Y. Xie and R. Zhang, *Goodness-of-fit tests on manifolds* (2019), 1909.05229.
- 961 [30] J. Bentley, *Programming pearls: Algorithm design techniques*, Commun. ACM **27**(9), 865–873
962 (1984), doi:10.1145/358234.381162.
- 963 [31] L. Pronzato and A. Pázman, *Design of experiments in nonlinear models*, Lecture notes in
964 statistics **212**, 1 (2013).
- 965 [32] B. Crestel, A. Alexanderian, G. Stadler and O. Ghattas, *A-optimal encoding weights for non-*
966 *linear inverse problems, with application to the helmholtz inverse problem*, Inverse problems
967 **33**(7), 074008 (2017).
- 968 [33] E. Kuram, B. Ozcelik, M. Bayramoglu, E. Demirbas and B. T. Simsek, *Optimization of cutting*
969 *fluids and cutting parameters during end milling by using d-optimal design of experiments*, Jour-
970 *nal of Cleaner Production* **42**, 159 (2013), doi:<https://doi.org/10.1016/j.jclepro.2012.11.003>.
- 971 [34] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel,
972 C. O. Rasmussen and P. Z. Skands, *An introduction to PYTHIA 8.2*, Comput. Phys. Commun.
973 **191**, 159 (2015), doi:10.1016/j.cpc.2015.01.024, 1410.3012.
- 974 [35] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lonnblad, J. Monk, H. Schulz
975 and F. Siegert, *RIVET user manual*, Comput. Phys. Commun. **184**, 2803 (2013),
976 doi:10.1016/j.cpc.2013.05.021, 1003.0694.

- 977 [36] S. B. Pope, *Algorithms for ellipsoids*, Cornell University Report No. FDA pp. 08–01 (2008).
- 978 [37] E. Bothmann *et al.*, *Event Generation with Sherpa 2.2*, SciPost Phys. **7**(3), 034 (2019),
979 doi:10.21468/SciPostPhys.7.3.034, 1905.09127.
- 980 [38] P. Pfeifenschneider *et al.*, *QCD analyses and determinations of α_s in e^+e^- annihilation at ener-*
981 *gies between 35-GeV and 189-GeV*, Eur. Phys. J. C **17**, 19 (2000), doi:10.1007/s100520000432,
982 hep-ex/0001055.
- 983 [39] P. Abreu *et al.*, *Tuning and test of fragmentation models based on identified particles and*
984 *precision event shape data*, Z. Phys. C **73**, 11 (1996), doi:10.1007/s002880050295.
- 985 [40] K. Abe *et al.*, *Measurement of the b quark fragmentation function in Z^0 decays*, Phys. Rev.
986 D **65**, 092006 (2002), doi:10.1103/PhysRevD.66.079905, [Erratum: Phys.Rev.D 66, 079905
987 (2002)], hep-ex/0202031.
- 988 [41] C. Amsler *et al.*, *Review of Particle Physics*, Phys. Lett. B **667**, 1 (2008),
989 doi:10.1016/j.physletb.2008.07.018.
- 990 [42] J. Pumplin, D. Stump, R. Brock, D. Casey, J. Huston, J. Kalk, H. L. Lai and W. K. Tung,
991 *Uncertainties of predictions from parton distribution functions. 2. The Hessian method*, Phys.
992 Rev. D **65**, 014013 (2001), doi:10.1103/PhysRevD.65.014013, hep-ph/0101032.
- 993 [43] A. Booker, J. Dennis Jr, P. Frank, D. Serafini, V. Torczon and M. Trosset, *A rigorous framework*
994 *for optimization of expensive functions by surrogates*, Structural Multidisciplinary Optimization
995 **17**, 1 (1999).
- 996 [44] M. Powell, *Recent Research at Cambridge on Radial Basis Functions*, New Developments in
997 Approximation Theory, pp. 215-232. Birkhäuser, Basel (1999).
- 998 [45] S. Kotz, N. Balakrishnan and N. L. Johnson, *Continuous multivariate distributions, Volume*
999 *1: Models and applications*, John Wiley & Sons (2004).
- 1000 [46] D. M. Blei, A. Y. Ng and M. I. Jordan, *Latent Dirichlet allocation*, the Journal of machine
1001 Learning research **3**, 993 (2003).
- 1002 [47] K. W. Ng, G.-L. Tian and M.-L. Tang, *Dirichlet and related distributions: Theory, methods*
1003 *and applications* (2011).
- 1004 [48] R. Regis and C. Shoemaker, *A stochastic radial basis function method for the global optimization*
1005 *of expensive functions*, INFORMS Journal on Computing **19**, 497 (2007).

1006 8 Online Supplement

1007 Online supplement for “BROOD: Bilevel and Robust Optimization and Outlier Detection for Effi-
1008 cient Tuning of High-Energy Physics Event Generators”.

1009 8.1 Solving the outer problem with derivative-free surrogate optimization

1010 Solving the inner optimization problem can become computationally demanding as it depends on
1011 the number of observables involved, the number of bins per observable (and therefore the number
1012 of parameters), and the starting guess (and therefore the number of iterations needed). Thus, the
1013 goal is to determine the optimal weights \mathbf{w}^* within as few iterations of the outer loop as possible
1014 since this number determines how often we have to solve the inner optimization problem. We do not
1015 have a full analytic expression of $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$ (black box) since computing this value involves solving
1016 the inner optimization problem. Thus, also derivatives of $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$ are not available. A widely used
1017 approach for optimizing computationally expensive black-box functions is to use computationally
1018 cheap approximations (surrogates, metamodels) of the expensive function and to use the approxima-
1019 tion throughout the optimization to make iterative sampling decisions [43]. Here, we approximate
1020 $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}})$ with a radial basis function (RBF) [44], although in general any approximation model
1021 could be used. An RBF interpolant is defined as follows:

$$s(\mathbf{w}) = \sum_{i=1}^n \gamma_i \phi(\|\mathbf{w} - \mathbf{w}_i\|_2) + q(\mathbf{w}), \quad (18)$$

1022 where $s : \mathbb{R}^{|\mathcal{S}_O|} \mapsto \mathbb{R}$, \mathbf{w}_i , $i = 1, \dots, n$, are the weight vectors for which we have already evaluated the
1023 objective function of the outer optimization problem, γ_i are parameters that must be determined,
1024 $\phi(\cdot)$ is the radial basis function (here, we choose the cubic, $\phi(r) = r^3$, but other options are possible),
1025 $\|\cdot\|_2$ denotes the Euclidean norm, and $q(\cdot)$ is a polynomial tail whose order depends on the choice
1026 of ϕ . When using the cubic RBF, the polynomial tail must be at least linear ($q(\mathbf{w}) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{w}$)
1027 in order to uniquely determine the RBF parameters ($\gamma_i, i = 1, \dots, n, \beta_0, \boldsymbol{\beta} = [\beta_1, \dots, \beta_{|\mathcal{S}_O|}]^\top$).
1028 The RBF interpolant $s(\mathbf{w})$ then predicts the value of the objective function at the point \mathbf{w} . It is
1029 interpolating, and thus the prediction at an already evaluated point \mathbf{w}_i will agree with the observed
1030 function value. Using the RBF, we thus have $g(\mathbf{w}, \hat{\mathbf{p}}_{\mathbf{w}}) = s(\mathbf{w}) + e(\mathbf{w})$, where $e(\mathbf{w})$ denotes the
1031 difference between the RBF and the true function value and it is 0 at already evaluated vectors \mathbf{w}_i .

1032 The values of the RBF parameters are determined by solving a linear system of equations:

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta}' \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix}, \quad (19)$$

1033 where the elements of $\mathbf{\Phi}$ are $\Phi_{\ell\nu} = \phi(\|\mathbf{w}_\ell - \mathbf{w}_\nu\|_2)$, $\ell, \nu = 1 \dots n$, $\mathbf{0}$ is a matrix with all entries 0 of
1034 appropriate dimension, and

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^\top & 1 \\ \vdots & \vdots \\ \mathbf{w}_n^\top & 1 \end{bmatrix} \quad \boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix} \quad \boldsymbol{\beta}' = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{|\mathcal{S}_\mathcal{O}|} \\ \beta_0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} g(\mathbf{w}_1, \widehat{\mathbf{p}}_{\mathbf{w}_1}) \\ g(\mathbf{w}_2, \widehat{\mathbf{p}}_{\mathbf{w}_2}) \\ \vdots \\ g(\mathbf{w}_n, \widehat{\mathbf{p}}_{\mathbf{w}_n}) \end{bmatrix}. \quad (20)$$

1035 The linear system in Eq. (19) has a solution if and only if $\text{rank}(\mathbf{W}) = |\mathcal{S}_\mathcal{O}| + 1$ [14]. During the
1036 optimization, we use the RBF prediction at unsampled points to determine a new vector \mathbf{w} for which
1037 we solve the inner optimization problem. It is important that at this step only weights that sum up
1038 to 1 are chosen. The steps of the iterative sampling algorithm are summarized in Algorithm 8.1.

1039 The inputs that must be supplied to the algorithm are the number of points n_0 to be used in the
1040 initial experimental design and the maximum number n_{\max} of outer objective function evaluations
1041 (i.e., the number of inner optimizations) one is willing to allow. The number n_0 should in our case
1042 be at least $|\mathcal{S}_\mathcal{O}| + 1$, since this is the minimum number of points we need to fit the RBF model.
1043 n_{\max} should depend on how long the inner optimization takes and the time budget of the user.

1044 When creating the initial experimental design in Step 1, we have to ensure that the con-
1045 straint (3b) is satisfied. Also, we have the condition that the weights lie in $[0, 1]$ and are uniform
1046 in their support. This means that the weights follow the Dirichlet distribution, i.e., the set of
1047 points are uniformly distributed over the open standard $(|\mathcal{S}_\mathcal{O}| - 1)$ -simplex. To achieve this, we
1048 generate an initial design where all weights are drawn from the symmetric Dirichlet distribution,
1049 $\text{Dir}(\alpha_1 = \alpha_2 = \dots = \alpha_{|\mathcal{S}_\mathcal{O}|} = 1)$ [45–47].

1050 We evaluate the outer objective function at these points, i.e., we solve the inner optimization
1051 problem at each point and we obtain \mathbf{G} in Eq. (20). With the sum-one-scaled initial experimental
1052 design, however, the rank of the matrix \mathbf{W} is now only $|\mathcal{S}_\mathcal{O}|$ (and not the required $|\mathcal{S}_\mathcal{O}| + 1$). Thus,
1053 we solve the problem as one of dimension $|\mathcal{S}_\mathcal{O}| - 1$, i.e., for fitting the RBF model, we only use the

Algorithm 8.1: Derivative-free optimization of the outer equality-constrained optimization problem

Input: Number of initial experimental design points n_0 ; the maximum number of evaluations

n_{\max}

Output: The best weight vector \mathbf{w}^* and corresponding $\hat{\mathbf{p}}_{\mathbf{w}^*}^*$

- 1: Create an initial experimental design with n_0 points; ensure that Eq. (3b) is satisfied for all points;
 - 2: Compute the value of the outer optimization objective function at all points in the initial design;
 - 3: Fit an RBF model to the sample data pairs $\{(\mathbf{w}_i, g(\mathbf{w}_i, \hat{\mathbf{p}}_{\mathbf{w}_i}))\}_{i=1}^{n_0}$
 - 4: Set $n = n_0$
 - 5: **while** $n < n_{\max}$ **do**
 - 6: Use the RBF to determine a new point \mathbf{w}_{new} and ensure that Eq. (3b) is satisfied;
 - 7: Solve the inner optimization problem for \mathbf{w}_{new} and obtain $\hat{\mathbf{p}}_{\mathbf{w}_{\text{new}}}$;
 - 8: Compute the value of the outer optimization objective function for $(\mathbf{w}_{\text{new}}, \hat{\mathbf{p}}_{\mathbf{w}_{\text{new}}})$;
 - 9: Update the RBF model with the new data;
 - 10: $n \leftarrow n + 1$;
 - 11: **end while**
 - 12: **return** the best parameter values $(\mathbf{w}^*, \hat{\mathbf{p}}_{\mathbf{w}^*}^*)$;
-

1054 first $|\mathcal{S}_O| - 1$ values of each sample point (the “reduced” sample points). Thus, we use

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,|\mathcal{S}_O|-1} & 1 \\ w_{2,1} & w_{2,2} & \cdots & w_{2,|\mathcal{S}_O|-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,|\mathcal{S}_O|-1} & 1 \end{bmatrix} \quad (21)$$

1055 and the coefficient vector for the polynomial tail thus becomes $[\beta_1, \dots, \beta_{|\mathcal{S}_O|-1}, \beta_0]^\top$. The vector γ
 1056 and the matrix \mathbf{G} do not change. The elements of Φ are computed from the $(|\mathcal{S}_O| - 1)$ -dimensional
 1057 sample vectors. Note, however, that when we evaluate the objective function in Eq. (3a), we always
 1058 evaluate it for the full-dimensional vectors, as we can simply compute $w_{j,|\mathcal{S}_O|} = 1 - \sum_{i=1}^{|\mathcal{S}_O|-1} w_i$ for
 1059 each $j = 1, \dots, n$.

1060 In the iterative sampling procedure (Steps 5-11), we use the RBF model to determine a new
 1061 vector \mathbf{w}_{new} at which we will do the next evaluation of Eq. (3a). Since we do not know whether the
 1062 objective function is multimodal, we have to balance local and global search steps, i.e., we have to
 1063 balance our sample point selection such that we select points with low predicted function values but
 1064 also points that are far away from already evaluated points. Moreover, the new sample point must
 1065 satisfy Eq. (3b). In order to do so, we generate a large set of candidate points from the Dirichlet
 1066 distribution. We use the RBF to predict the function values at the candidate points. Since the RBF
 1067 is defined over the $(|\mathcal{S}_O| - 1)$ -dimensional space, we use only the first $|\mathcal{S}_O| - 1$ parameter values
 1068 of each candidate point. We denote the $(|\mathcal{S}_O| - 1)$ -dimensional candidate points by $\mathbf{x}_1, \dots, \mathbf{x}_{N_{\text{cand}}}$,
 1069 where we choose N_{cand} large (for example, $500|\mathcal{S}_O|$). For each candidate point, we use the RBF to
 1070 predict its function value using (18) and we obtain $s(\mathbf{x}_k), k = 1, \dots, N_{\text{cand}}$. We scale these values
 1071 to $[0,1]$ according to

$$V_s(\mathbf{x}_k) = \frac{s(\mathbf{x}_k) - s_{\min}}{s_{\max} - s_{\min}}, k = 1, \dots, N_{\text{cand}}, \quad (22)$$

1072 where

$$s_{\min} = \min\{s(\mathbf{x}_k), k = 1, \dots, N_{\text{cand}}\} \text{ and } s_{\max} = \max\{s(\mathbf{x}_k), k = 1, \dots, N_{\text{cand}}\}. \quad (23)$$

1073 We also compute the distances $d(\mathbf{x}_k, S)$ of each candidate point to the set of already evaluated points
 1074 S (in the $(|\mathcal{S}_O| - 1)$ -dimensional Euclidean space), and we scale these distances to $[0,1]$ according
 1075 to

$$V_d(\mathbf{x}_k) = \frac{d_{\max} - d(\mathbf{x}_k)}{d_{\max} - d_{\min}}, k = 1, \dots, N_{\text{cand}}, \quad (24)$$

1076 where

$$d_{\min} = \min\{d(\mathbf{x}_k, S), k = 1, \dots, N_{\text{cand}}\} \text{ and } d_{\max} = \max\{d(\mathbf{x}_k, S), k = 1, \dots, N_{\text{cand}}\}. \quad (25)$$

1077 The ideal new sample point \mathbf{w}_{new} will have a large distance to the set of already evaluated points
 1078 S and a low predicted objective function value. Using the two criteria defined above, we compute
 1079 a weighted sum of both (following [48])

$$V(\mathbf{x}_k) = \nu V_s(\mathbf{x}_k) + (1 - \nu)V_d(\mathbf{x}_k), k = 1, \dots, N_{\text{cand}}, \quad (26)$$

1080 where $\nu \in [0, 1]$ is a parameter that determines how much emphasis we put on either criterion.
 1081 If ν is large, it means we put most emphasis on V_s , and we favor candidate points that have low
 1082 predicted objective function values. This also means that the search is more local as low function
 1083 values are usually predicted around the best point found so far. If ν is small, we put more emphasis
 1084 on V_d and we favor points that are far away from the set of already evaluated points, and thus the
 1085 search is more global. By varying the weights ν between different values in $[0, 1]$, we can achieve a
 1086 repeated transition between local and global search, and therefore we can avoid becoming stuck in
 1087 a local optimum. The candidate point with the lowest V value will become the new sample point
 1088 \mathbf{w}_{new} . We evaluate the objective function (inner optimization) at the new point (augmented with
 1089 the missing parameter value), and given the new data, we update the RBF model. The algorithm
 1090 iterates until the maximum number of function evaluations n_{max} has been reached.

1091 8.2 Polynomial-time algorithm for filtering bins by hypothesis testing

1092 In this section, we describe the polynomial-time algorithm to solve the problem of finding the largest
 1093 contiguous subset of bins $\mathcal{B} \subset \mathcal{O}$ to be kept for tuning, i.e., finding the largest contiguous subset of
 1094 bins $\mathcal{B} \subset \mathcal{O}$ such that $\chi_{\mathcal{B}}^2 \leq \chi_{c, \mathcal{B}}^2$, where $\chi_{c, \mathcal{B}}^2$ is the critical value for bins in \mathcal{B} . This algorithm is
 1095 described in Algorithm 8.2 and it is based on the maximum subarray problem [30].

1096 In this algorithm, we first find the critical value for each bin in line 1 as described in Section 3.2.
 1097 The degree of the freedom is given by $\rho_{\mathcal{B}} = |\mathcal{B}| - d$ and since $\rho_{\mathcal{B}}$ cannot be negative, the critical
 1098 values for only the bin index $b > d$ is calculated in line 1. Then the χ^2 test statistic is computed for
 1099 each bin in \mathcal{O} in lines 2-3. Then, while iterating through the bins in \mathcal{O} , in lines 6, we check whether
 1100 the current bin b can be added to \mathcal{B} and if so, we update the counters and add the current bin b to
 1101 the end of \mathcal{B} (via e) in lines 7-10. If the current bin b cannot be added to \mathcal{B} , then in lines 12-13 we
 1102 shift the start s of \mathcal{B} (through τ) such that the start is now at the bin index where the condition
 1103 in line 6 could be satisfied in future iterations. Finally, in lines 14-19, we perform a sanity check to
 1104 make sure that \mathcal{B} contains the set of bins that yield the lowest $\chi_{\mathcal{B}}^2$ test statistic.

Algorithm 8.2: Algorithm to find bins \mathcal{B} in observable \mathcal{O} to keep for tuning

Input : $f_b, \mathcal{R}_b, \Delta f_b, \Delta \mathcal{R}_b, \forall b \in \mathcal{O}$; significance level α

Output: start index s and end index e of bins, i.e., $\mathcal{B} = \{s, \dots, e\}$ to keep in \mathcal{O}

1 Calculate the critical values for each bin:

$$k_b = \begin{cases} \chi_{c,b}^2 = f(\rho_b, \alpha), & \text{if } b > d \\ \infty, & \text{otherwise} \end{cases}, \quad \forall b \in \{1, 2, \dots, |\mathcal{O}|\}, \mathbf{p} \in \Omega \subset \mathbb{R}^d$$

2 Find \mathbf{p}^* by minimizing $\chi_{\mathcal{O}}^2$ in Eq. (12)

3 Calculate the test statistic values for each bin:

$$\chi_b^2(\mathbf{p}^*) = \frac{(f_b(\mathbf{p}^*) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p}^*)^2 + \Delta \mathcal{R}_b^2}, \quad \forall b \in \{1, 2, \dots, |\mathcal{O}|\}$$

4 Initialize $\Sigma \leftarrow 0, \hat{b} \leftarrow 0, s \leftarrow 0, e \leftarrow 0, \tau \leftarrow 0$

5 **for** $b \in \{1, 2, \dots, |\mathcal{O}|\}$ **do**

6 **if** $\Sigma + \chi_b^2 \leq k_{\hat{b}+1}$ **then**

7 $\Sigma \leftarrow \Sigma + \chi_b^2$

8 $\hat{b} \leftarrow \hat{b} + 1$

9 $s \leftarrow \tau$

10 $e \leftarrow b$

11 **else if** $\Sigma \neq 0$ **then**

12 $\Sigma \leftarrow \Sigma - \chi_{b-\hat{b}}^2 + \chi_b^2$

13 $\tau \leftarrow b - \hat{b} + 1$

14 **if** $s > 0$ **and** $\chi_{s-1}^2 < \chi_e^2$ **then**

15 $e \leftarrow e - 1$

16 $s \leftarrow s - 1$

17 **else if** $e < |\mathcal{O}|$ **and** $\chi_s^2 > \chi_{e+1}^2$ **then**

18 $e \leftarrow e + 1$

19 $s \leftarrow s + 1$

20 **return** $\mathcal{B} = \{s, \dots, e\}$.

1105 **8.3 A14 and SHERPA physics parameters**

1106 The A14 tunable physics parameters, their definitions and tuning ranges are shown in Table 22.

1107 The SHERPA parameters, their definitions and tuning ranges are shown in Table 23.

Table 22: A14 physics parameters, their definitions and tuning ranges (min, max).

Parameters	Definition	min	max
<code>SigmaProcess:alphaSvalue</code>	The α_S value at scale $Q^2 = M_Z^2$	0.12	0.15
<code>BeamRemnants:primordialkTthard</code>	Hard interaction primordial k_{\perp}	1.5	2.0
<code>SpaceShower:pT0Ref</code>	ISR p_T cutoff	0.75	2.5
<code>SpaceShower:pTmaxFudge</code>	Mult. factor on max ISR evolution scale	0.5	1.5
<code>SpaceShower:pTdampFudge</code>	Factorization/renorm scale damping	1.0	1.5
<code>SpaceShower:alphaSvalue</code>	ISR α_S	0.10	0.15
<code>TimeShower:alphaSvalue</code>	FSR α_S	0.10	0.15
<code>MultipartonInteractions:pT0Ref</code>	MPI p_T cutoff	1.5	3.0
<code>MultipartonInteractions:alphaSvalue</code>	MPI α_S	0.10	0.15
<code>BeamRemnants:reconnectRange</code>	CR strength	1.0	10.0

Table 23: SHERPA physics parameters, their definitions and tuning ranges (min, max).

Parameters	Definition	min	max
<code>KT_0</code>	generic parameter for non-perturbative transverse momentum	0.5	1.5
<code>ALPHA_G</code>	gluon fragmentation	0.62	1.88
<code>ALPHA_L</code>	light quark fragmentation z power	1.25	3.75
<code>BETA_L</code>	light quark fragmentation $1 - z$ power	0.05	0.15
<code>GAMMA_L</code>	light quark fragmentation exp power	0.25	0.75
<code>ALPHA_H</code>	heavy quark fragmentation z power	1.25	3.75
<code>BETA_H</code>	heavy quark fragmentation $1 - z$ power	0.375	1.125
<code>GAMMA_H</code>	heavy quark fragmentation exp power	0.05	0.15
<code>STRANGE_FRACTION</code>	suppression of s quarks	0.25	0.75
<code>BARYON_FRACTION</code>	suppression of baryons	0.09	0.27
<code>P_QS_by_P_QQ_norm</code>	fraction of di-quarks with one strange quark	0.24	0.72
<code>P_SS_by_P_QQ_norm</code>	fraction of di-quarks with two strange quarks	0.01	0.03
<code>P_QQ1_by_P_QQ0</code>	fraction of di-quarks with spin-1 to spin-0	0.5	1.5

1108 8.4 Selection of the best hyperparameter in robust optimization

1109 In order to find the best value for μ in the robust optimization, we first build for each run (each
 1110 μ) a cumulative density curve of the number of observables for which $\frac{\chi_{\mathcal{O}}^2(\mathbf{p}^*, \mathbf{w})}{|\mathcal{O}|} \leq \tau$, where \mathbf{p}^* is
 1111 the optimal parameter obtained from the robust optimization run, $\mathbf{w} = \mathbf{1}$, $\tau \in \mathbb{R}^+$ and $\mathcal{O} \in \mathcal{S}_{\mathcal{O}}$.
 1112 Then, we construct the “ideal” cumulative density curve, for which \mathbf{p}^* in $\frac{\chi_{\mathcal{O}}^2(\mathbf{p}^*, \mathbf{w})}{|\mathcal{O}|} \leq \tau$ is obtained
 1113 by optimizing for each observable \mathcal{O} separately. An example plot showing the cumulative density
 1114 curve from the ideal case to some of the robust optimization runs is shown in Figure 12.

1115 Then, the area between the cumulative density curve for each robust optimization run and the
 1116 ideal cumulative density curve is computed. For the A14 dataset and all runs completed for robust
 1117 optimization, the area between the curve is given in Table 24 (smaller values are better). Finally,
 1118 for completeness, the best values of μ found for both the A14 and SHERPA datasets are given in
 1119 Table 25.

1120 8.5 Outlier observables in the A14 dataset

1121 There are 12 outlier observables using the cubic polynomial approximation and 9 outlier observables
 1122 using the rational approximation in the A14 dataset.

Cubic Polynomial Model	Rational Approximation Model
/ATLAS_2011_I919017/d01-x02-y02	/ATLAS_2011_I919017/d01-x02-y02
/ATLAS_2011_I919017/d01-x02-y03	/ATLAS_2011_I919017/d01-x04-y04
/ATLAS_2011_I919017/d01-x03-y02	/ATLAS_2011_I919017/d02-x04-y03
/ATLAS_2011_I919017/d01-x03-y07	/ATLAS_2011_I919017/d02-x04-y04
/ATLAS_2011_I919017/d01-x04-y07	/ATLAS_2011_I919017/d02-x04-y05
/ATLAS_2011_I919017/d01-x04-y08	/ATLAS_2011_I919017/d02-x04-y09
/ATLAS_2011_I919017/d01-x04-y09	/ATLAS_2011_I919017/d02-x04-y10
/ATLAS_2011_I919017/d02-x04-y04	/ATLAS_2011_I919017/d02-x04-y14
/ATLAS_2011_I919017/d02-x04-y10	/ATLAS_2011_I919017/d02-x04-y15
/ATLAS_2011_I919017/d02-x04-y13	
/ATLAS_2011_I919017/d02-x04-y14	
/ATLAS_2011_I919017/d02-x04-y15	

Table 24: Area between ideal cumulative density curve and the cumulative density curves of the robust optimization runs for various hyperparameters μ for the A14 full dataset (smaller values are better). The data are organized in ascending order of the area between the curves.

rank	μ	Area	rank	μ	Area	rank	μ	Area	rank	μ	Area
1	80	7.51e+02	26	79	1.05e+03	51	24	1.32e+03	76	38	1.66e+03
2	78	7.93e+02	27	81	1.12e+03	52	35	1.32e+03	77	29	1.72e+03
3	76	7.95e+02	28	71	1.13e+03	53	93	1.35e+03	78	51	1.72e+03
4	77	8.15e+02	29	95	1.14e+03	54	89	1.39e+03	79	49	1.73e+03
5	73	8.53e+02	30	10	1.15e+03	55	45	1.40e+03	80	57	1.73e+03
6	70	8.91e+02	31	11	1.17e+03	56	42	1.41e+03	81	50	1.74e+03
7	90	8.96e+02	32	12	1.17e+03	57	41	1.43e+03	82	43	1.74e+03
8	26	9.09e+02	33	18	1.18e+03	58	68	1.43e+03	83	44	1.76e+03
9	88	9.11e+02	34	3	1.19e+03	59	67	1.44e+03	84	55	1.79e+03
10	74	9.14e+02	35	21	1.19e+03	60	46	1.44e+03	85	47	1.87e+03
11	86	9.42e+02	36	20	1.19e+03	61	39	1.46e+03	86	60	1.93e+03
12	72	9.43e+02	37	16	1.19e+03	62	30	1.48e+03	87	37	1.94e+03
13	27	9.44e+02	38	69	1.20e+03	63	63	1.48e+03	88	59	1.95e+03
14	83	9.47e+02	39	22	1.20e+03	64	40	1.51e+03	89	33	1.96e+03
15	75	9.53e+02	40	23	1.21e+03	65	64	1.52e+03	90	54	1.97e+03
16	87	9.59e+02	41	19	1.21e+03	66	28	1.55e+03	91	58	1.99e+03
17	8	9.61e+02	42	13	1.22e+03	67	61	1.56e+03	92	53	2.08e+03
18	82	9.72e+02	43	25	1.23e+03	68	98	1.56e+03	93	94	2.13e+03
19	2	9.77e+02	44	97	1.24e+03	69	62	1.58e+03	94	56	2.14e+03
20	84	9.80e+02	45	7	1.25e+03	70	66	1.58e+03	95	52	2.23e+03
21	5	9.90e+02	46	15	1.27e+03	71	32	1.58e+03	96	99	2.74e+03
22	85	9.99e+02	47	17	1.28e+03	72	48	1.59e+03	97	36	3.00e+03
23	1	1.01e+03	48	14	1.29e+03	73	31	1.60e+03	98	34	3.02e+03
24	9	1.03e+03	49	92	1.30e+03	74	65	1.61e+03	99	91	3.05e+03
25	6	1.04e+03	50	4	1.31e+03	75	96	1.64e+03	100	100	3.89e+03

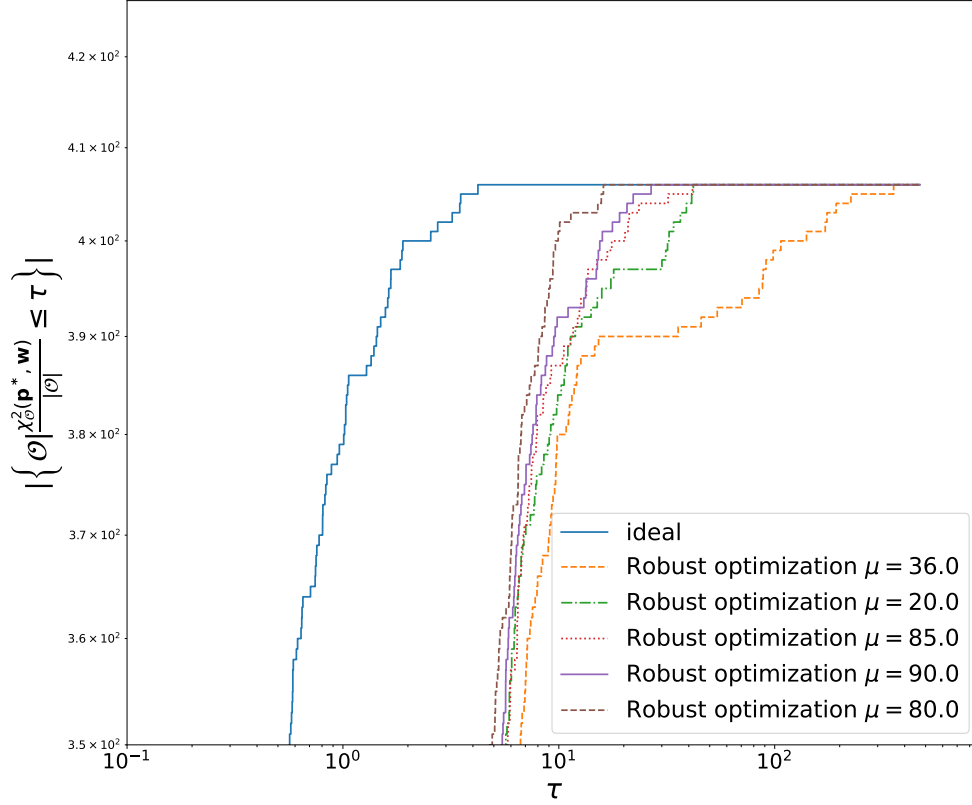


Figure 12: Ideal cumulative density curve and the cumulative density curves of robust optimization runs with selected hyperparameter values μ for the A14 dataset.

Table 25: Best μ obtained for A14 and SHERPA datasets when unfiltered (*All data*), bin filtered and observable filtered data are used for parameter tuning.

Dataset	All data	Bin filtered	Observable filtered
A14	80	76	80
SHERPA	82	71	73

1123 **8.6 Outlier observables in the SHERPA dataset**

1124 There are 2 outlier observables using the cubic polynomial approximation and 3 outlier observables
 1125 using the rational approximation in the SHERPA dataset.

Cubic Polynomial Model	Rational Approximation Model
/DELPHI_1996_S3430090/d07-x01-y01	/DELPHI_1996_S3430090/d02-x01-y01
/DELPHI_1996_S3430090/d08-x01-y01	/DELPHI_1996_S3430090/d07-x01-y01
	/DELPHI_1996_S3430090/d08-x01-y01

1124 8.7 Bin filtered data for A14 dataset

1125 In Table 28, we give the names of the A14 observables from which bins have been filtered, the
 1126 number of bins filtered out, critical χ^2 value, and χ^2 test statistic before and after filtering the bins.

1127 8.8 Bin filtered data for SHERPA dataset

1128 In Table 29, we give the names of the SHERPA observables from which bins have been filtered, the
 1129 number of bins filtered out, critical χ^2 value, and χ^2 test statistic before and after filtering the bins.

1130 8.9 Complete results from filtering out observables and bins

1131 In Figures 13 and 14, the cumulative distribution plots for parameters obtained after bin filtering
 1132 and observable filtering for the A14 data are presented. In Figures 15 and 16, the cumulative
 1133 distribution plots for parameters obtained after bin filtering and observable filtering for the SHERPA
 1134 data are presented. From these figures, we observe that there is no significant difference in the
 1135 number of bins within the 1σ variance level between the optimal parameters \mathbf{p}_a^* obtained when
 1136 all bins were used for tuning and the optimal parameters \mathbf{p}_b^* and \mathbf{p}_o^* obtained when only the bin
 1137 filtered and observable filtered bins are used for tuning, respectively. This indicates that the MC
 1138 generator cannot explain the bins removed by the filtering approaches very well. Hence, removing
 1139 these bins from the tuning process does not reduce the information required to achieve a good tune
 1140 as it performs very similarly to when all bins are used for tuning.

1141 8.10 Comparison of the rational approximation with the MC generator

1142 We compare the cumulative distribution of bins at different bands of variance levels computed using
 1143 the rational approximation (RA) model as $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$ and the MC generator model as
 1144 $\widetilde{r}_b(\mathbf{p}) = \frac{(\text{MC}_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta \text{MC}_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$, where \mathbf{p} are the parameters obtained from the different tuning approaches.
 1145 In Figure 7, we showed the plot of this comparison for bins in each category of the A14 dataset

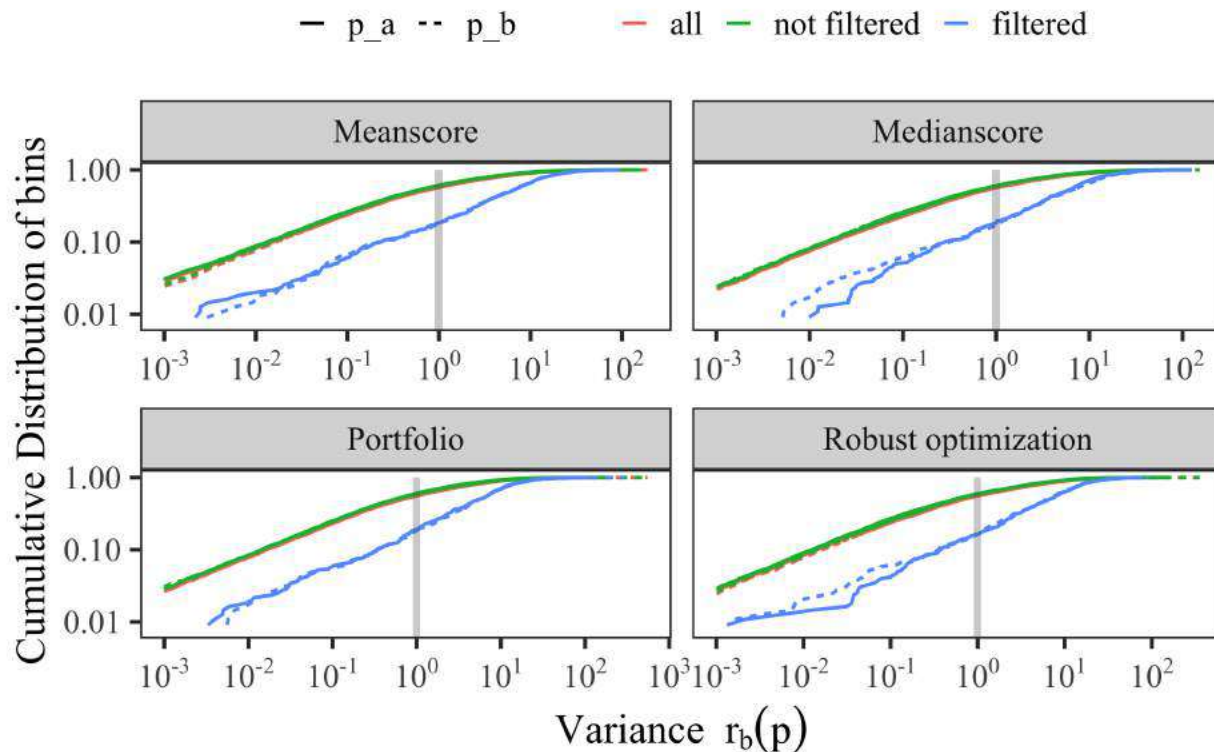


Figure 13: Cumulative distribution of bins for the A14 dataset at different bands of variance levels using different approaches. Results are shown using the parameters \mathbf{p}_a^* obtained using all bins during optimization, and the parameters \mathbf{p}_b^* obtained when only the bin filtered bins are used during optimization.

1146 using the parameters from three approaches. For completeness, in Figure 17, we show the plot of
 1147 this comparison for the remaining three approaches.

1148 We observe in Figure 17 that around the variance boundary, except for in the *Track-jet UE* and
 1149 *Multijets* categories, there is no significant difference in performance between $r_b(\mathbf{p})$ and $\widetilde{r_b(\mathbf{p})}$ for
 1150 each approach. In the case of *Track-jet UE* and *Multijets* categories, the number of bins that lie
 1151 within the variance boundary is quite low compared to other categories. This suggests that many
 1152 bins in these categories cannot be explained well by either the MC generator or the approximation
 1153 for the optimal tuning parameters reported by the approaches. Additionally, we observe in these
 1154 categories that the approximations are not able to capture the MC generator perfectly.

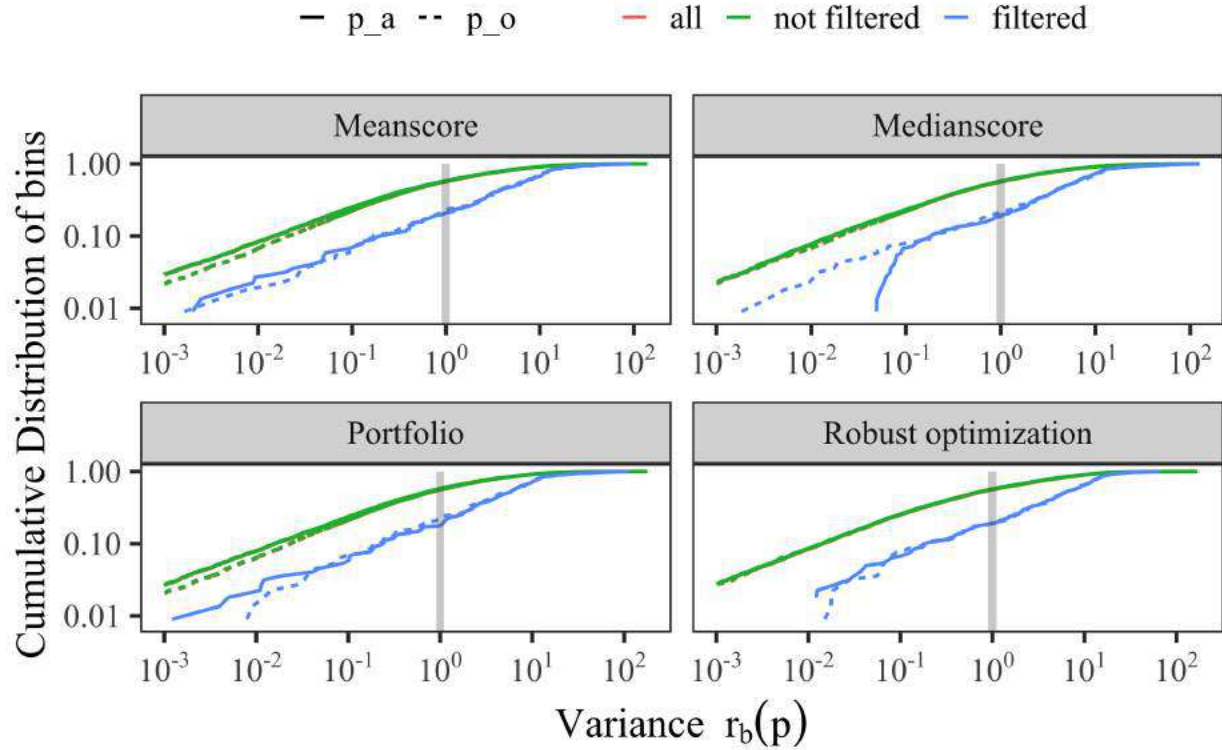


Figure 14: Same as Figure 13, but using observable filtering.

1155 8.11 Optimal parameter values for the A14 dataset with the rational approxi- 1156 mation

1157 To better visually compare the different solutions obtained with our optimization methods, we show
1158 the [0,1]-scaled optimal parameter values in Figure 18.

1159 8.12 Results for using the cubic polynomial to approximate the MC simulation

1160 In the main paper, we showed the numerical results when using a rational approximation of the
1161 MC simulation during tuning. In the A14 publication [3], a cubic polynomial was used. Thus, in
1162 this section, we present the results obtained with our optimization methods when using a cubic
1163 polynomial instead of a rational approximation.

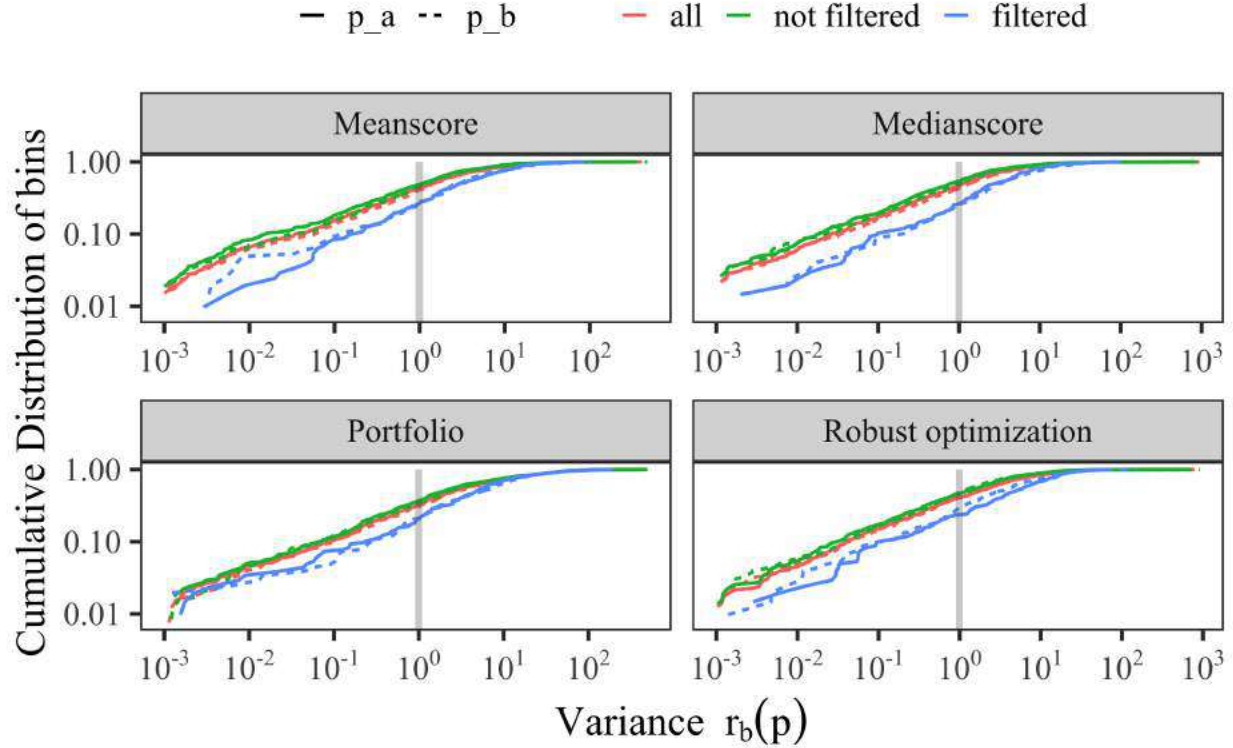


Figure 15: Same as Figure 13 , but for the SHERPA dataset.

1164 8.12.1 Comparison metric outcomes for the A14 dataset using the cubic polynomial 1165 approximation

1166 Tables 30-32 show the comparison metrics we introduced in the main paper in Section 4.2 when
1167 using the cubic polynomial approximation for the full data, the observable-filtered data, and the
1168 bin-filtered data, respectively. We see that for all three cases and most criteria (except for the
1169 D-optimality in the observable-filtered case), our automated methods for adjusting the observable
1170 weights perform better than the expert solution (i.e., using the parameters published in [3]).

1171 8.12.2 Optimal parameter values for the A14 dataset using the cubic polynomial 1172 approximation

1173 Table 33 shows the optimal values for the tuned parameters obtained by all methods for the A14
1174 dataset when using all observables in the tune. For Bilevel-meanscore, -medianscore and -portfolio,

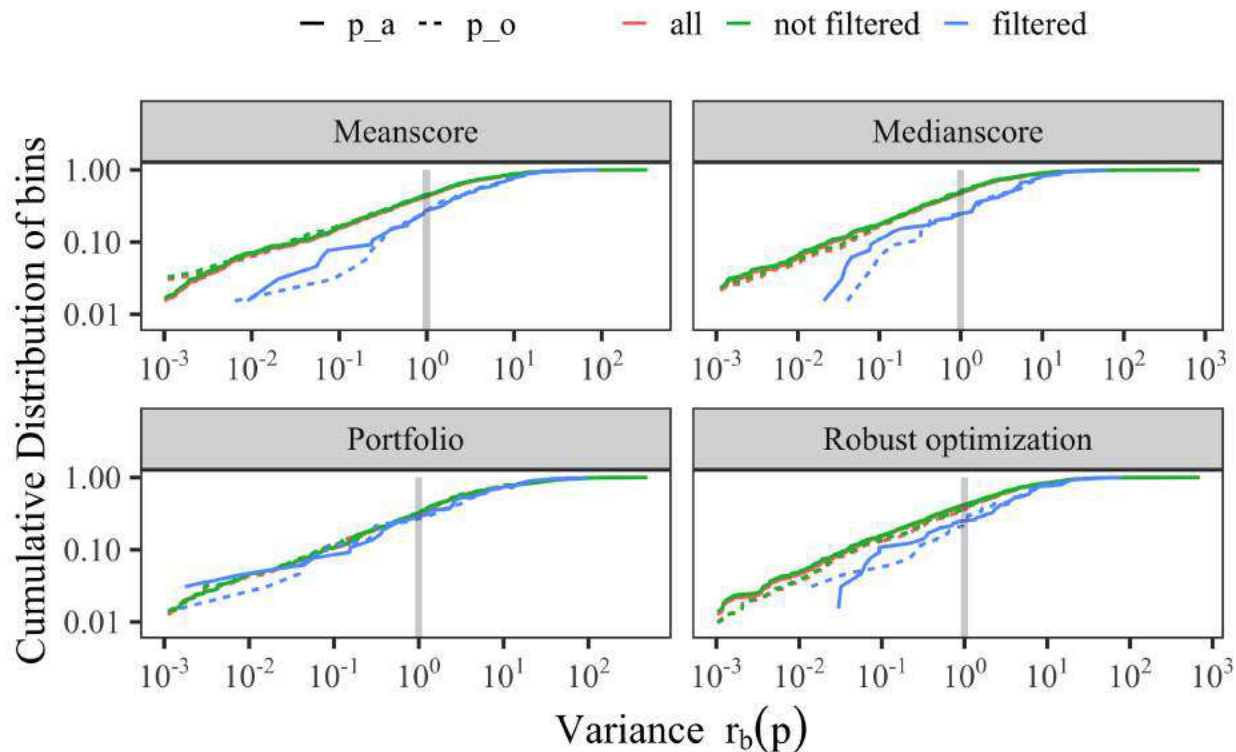


Figure 16: Same as Figure 15, but using observable filtering.

1175 we repeated the experiments three times using different random number seeds and we report the
 1176 best results among the three trials based on their respective objective functions. From these results,
 1177 we can see that the Bilevel-medianscore method leads to a solution that is closest to the expert's
 1178 solution.

1179 To better visually compare the different solutions obtained with our methods, we show the
 1180 $[0,1]$ -scaled optimal values in Figure 18. We can see that there are differences between the optimal
 1181 parameters obtained with the different methods, in particular, the results of the robust optimization
 1182 method tend to be further away from the expert's solution for parameters 1, 2, 3, 7, 8, 9 and 10.
 1183 The results of the portfolio optimization differ from the expert tune in particular for parameters 1,
 1184 2, 3, 4 and 7. The mean- and medianscore results are very similar to each other as well as to the
 1185 expert's solution.

1186 We conducted a similar analysis on the observable- and bin-filtered data. Table 34 shows the
 1187 optimal parameter values that we obtain with the automated optimization methods after filtering

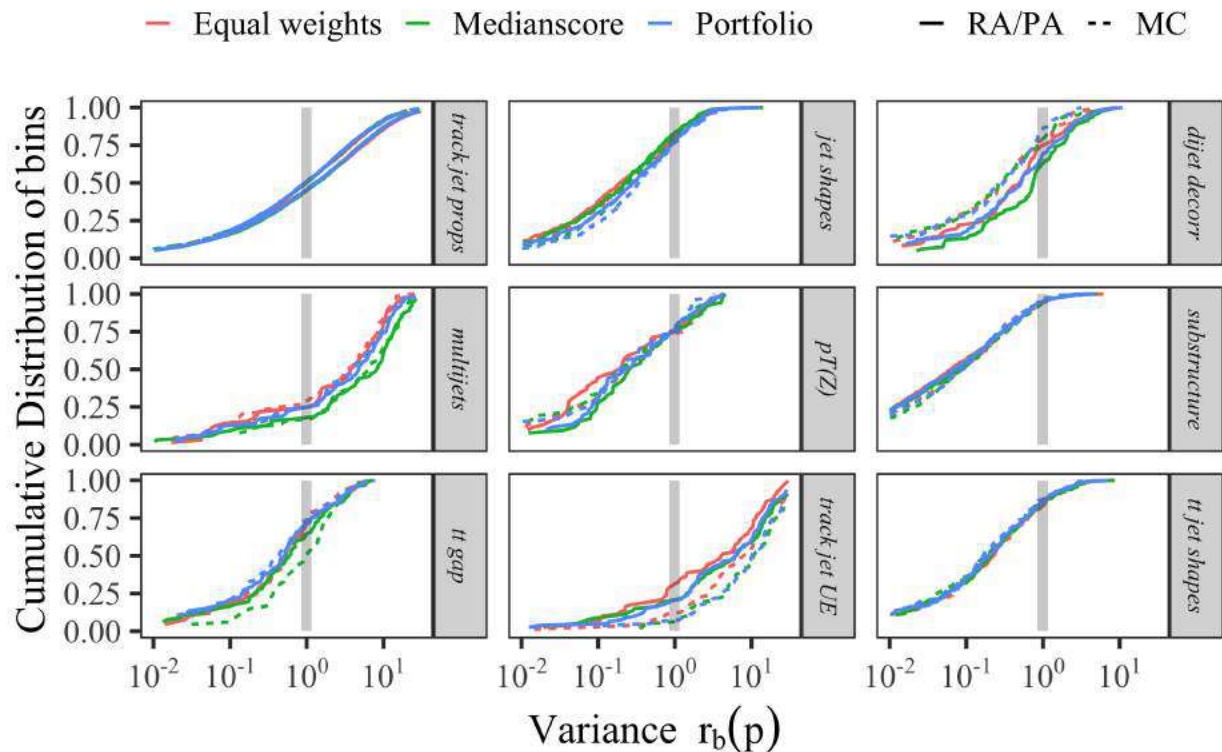


Figure 17: Cumulative distribution of bins in each category of the A14 dataset at different bands of variance levels computed with the rational approximation (RA) given by, $r_b(\mathbf{p}) = \frac{(f_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta f_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$ and the MC simulation given by $\widetilde{r_b(\mathbf{p})} = \frac{(\text{MC}_b(\mathbf{p}) - \mathcal{R}_b)^2}{\Delta \text{MC}_b(\mathbf{p})^2 + \Delta \mathcal{R}_b^2}$

1188 out the 12 observables that the model cannot explain (see also Section 3.1). The expert solution is
 1189 the same as before and based on all observables. We include it for easier comparison. With only a
 1190 few exceptions, all parameters obtained with the automated optimizations change (as compared to
 1191 using the full dataset). Figure 18 shows the optimal parameter values obtained with each method
 1192 scaled to $[0,1]$. In comparison to when using the full dataset, we see that the results of the robust
 1193 optimization now agree better with the expert's tune for parameters 3, 4, and 8, but less agreement
 1194 is achieved for parameter 10. Of the three bilevel methods, the medianscore objective function leads
 1195 to optimal parameters that are most similar to the expert tune.

1196 In Table 35 and Figure 18 we show the optimal parameter values obtained with our methods after

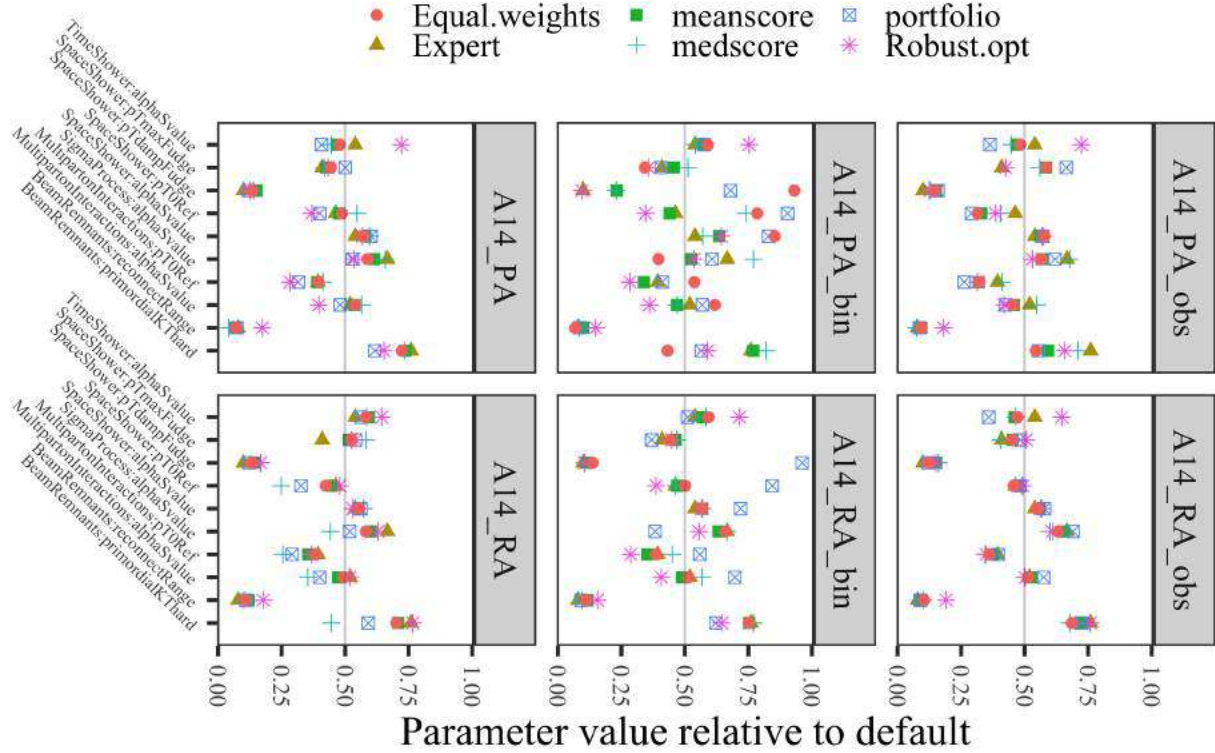


Figure 18: Optimal parameter values for the A14 dataset obtained when using all, bin-filtered (`_bin`) and observable-filtered (`_obs`) data in the optimization and the polynomial approximation (PA) and rational approximation (RA). Values are normalized to $[0,1]$.

1197 applying the bin-filtering approach described in Section 3.2 in the main document. In comparison
 1198 to our results that do not use any filtering, we can see a much larger disagreement in the optimal
 1199 parameters for all methods. In fact, all methods yield optimal parameters that are significantly
 1200 further away from the expert’s solution, except for parameters 7 and 10. The Euclidean distance
 1201 between the optimal parameters obtained by our proposed methods and the expert solution shows
 1202 that the bilevel-medianscore method leads to the most similar parameter values while all the other
 1203 methods lead to very different tunes.

1204 **8.12.3 Comparison of optimal weights for the A14 dataset with cubic polynomial**
1205 **approximation**

1206 In Table 36 we present the optimal weights assigned to each observable group by each method
1207 following the presentation style in [3]. The weights reported for our method are averages of the
1208 weights over all observables that belong to the same group. We scaled the weights such that they
1209 are on equal footing (all add up to 4580).

1210 The largest differences between the expert-adjusted values and the values determined by our
1211 methods are for *Multijets*, *t \bar{t} gap* and *Jet UE*, while for the remaining groups, the values are very
1212 similar. These results, together with our analysis above let us conclude that an automated method
1213 for adjusting the weights of observables for tuning parameters is a viable approach and can lead to
1214 better results than hand-tuning.

1215 **8.12.4 Optimal parameter values for the SHERPA dataset with rational approximation**

1216 For a better visual comparison of the different solutions obtained with our methods, we show the
1217 [0,1]-scaled optimal values in Figure 19. Compared to the results for the A14 dataset, we see that
1218 there are significant differences between the optimal parameters obtained with the different methods.

1219 **8.12.5 Optimal parameter values for the SHERPA dataset with the cubic polynomial**
1220 **approximation**

1221 The physics parameters \mathbf{p} and their optimization ranges are shown in Table 23. Tables 37, 38
1222 and 39 shows the optimal values for the physics parameters obtained by all methods when no
1223 filtering was applied before optimization, after using outlier detection to remove observables from
1224 the optimization, and after using the bin-filtering approach that excludes individual bins from
1225 the optimization, respectively. For an illustrative comparison, we show the [0,1]-scaled optimal
1226 parameter values in Figure 19. The default values lie right in the middle of the parameter range.

1227 **8.12.6 Comparison metric outcomes for the SHERPA dataset with the cubic polynomial**
1228 **approximation**

1229 Tables 40, 41, and 42 show the comparison metrics of our experiments when using the cubic poly-
1230 nomial approximation for the full data, the observable-filtered data, and the bin-filtered data, re-
1231 spectively. Smaller numbers indicate better performance. The smallest number of each metric is
1232 bold for better visualization.

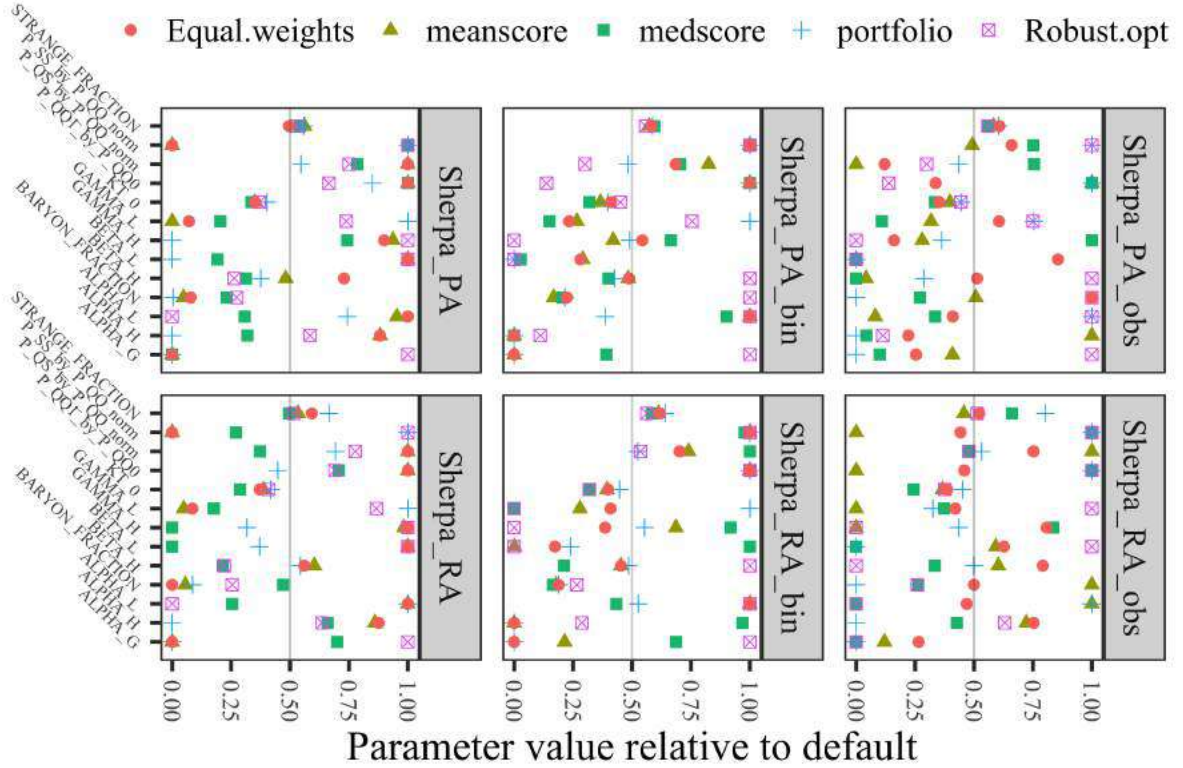


Figure 19: Comparison of the optimal parameter values for SHERPA obtained with the different optimization methods when no, observable, and bin data filtering was applied and the *rational* and polynomial approximation was used. Values are normalized to [0,1].

1233 Based on these results, we can see that the all-weights-equal method (i.e. not adjusting any
 1234 weights) has the best performance for the full dataset under the A- and D-optimality. The bilevel-
 1235 portfolio method performs best under the A- and D-optimality for both the observable- and bin-
 1236 filtered datasets. The robust optimization method performs best in all three cases under the
 1237 Weighted χ^2 criterion.

1238 8.13 Weights assigned by different fitting methods

1239 Figure ?? shows the weights per observable obtained from the tune to SHERPA using the methods
 1240 described in this paper.

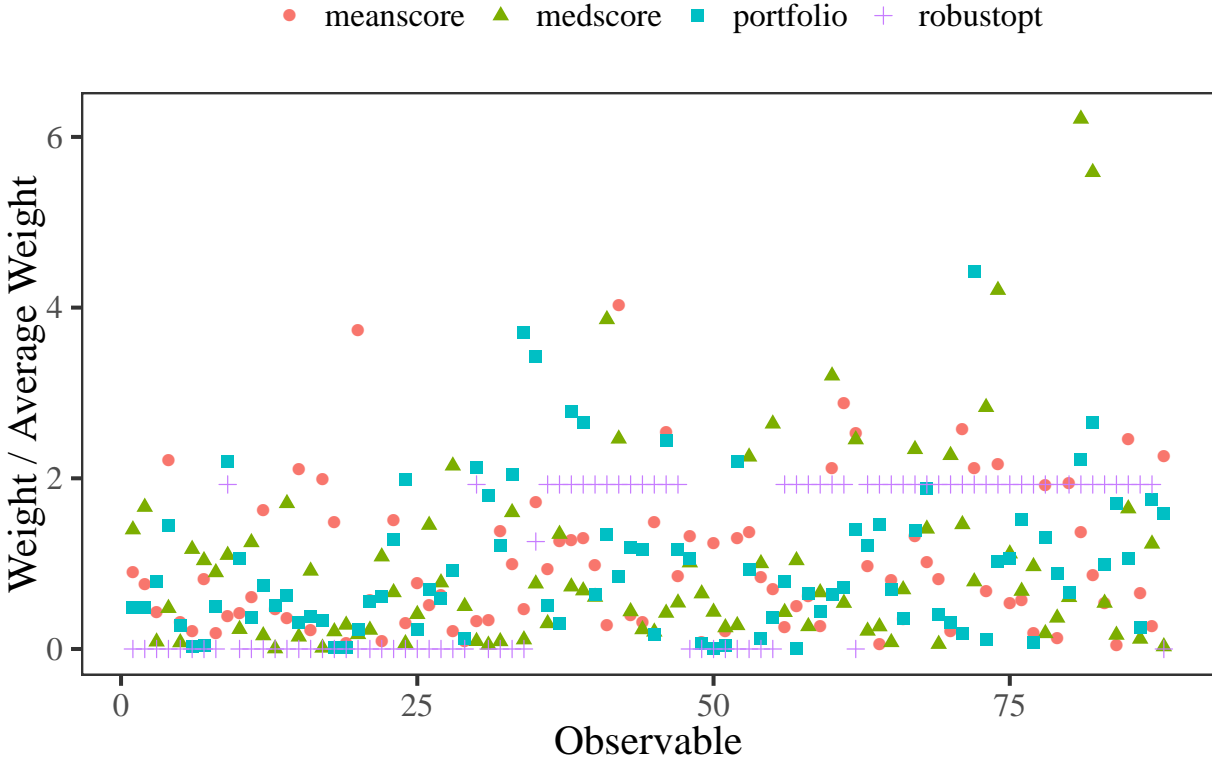


Figure 20: Distribution of weights assigned to observables for the different fitting methods described in the paper. Observables to the left are based on kinematic properties of events, while those to the right are particle multiplicities.

1241 **8.14 Eigentunes for the results obtained with the cubic polynomial approxima-** 1242 **tion**

1243 Tables 43 and 44 shows the eigentune results for the A14 and SHERPA datasets, respectively, when
1244 using the cubic polynomial approximation.

1245 **8.15 Generator settings for PYTHIA and SHERPA**

1246 Typical run card for A14 studies using PYTHIAV8.186.

Tune:pp = 14

Tune:ee = 7

```

PDF:useLHAPDF = on
PDF:LHAPDFset = NNPDF23_lo_as_0130_qed
PDF:LHAPDFmember = 0
PDF:extrapolateLHAPDF = off

! 3) Beam parameter settings. Values below agree with default ones.
Beams:idA = 2212           ! first beam, p = 2212, pbar = -2212
Beams:idB = 2212           ! second beam, p = 2212, pbar = -2212
Beams:eCM = 7000.         ! CM energy of collision

# uncomment for QCD
PhaseSpace:pTHatMin = 10.0
HardQCD:all = on
PhaseSpace:bias2Selection = on
PhaseSpace:bias2SelectionRef = 10.0
# uncomment for t-tbar
#Top:qqbar2ttbar = on
#Top:gg2ttbar = on
#SpaceShower:pTmaxMatch = 2
#SpaceShower:pTmaxFudge = 1
#SpaceShower:pTdampMatch = 1
# uncomment for Z
#WeakSingleBoson:ffbar2gmZ = On
#23:onMode = off
#23:onIfAny = 11 13 15 5 4 3
#SpaceShower:pTmaxMatch = 2
#SpaceShower:pTmaxFudge = 1
#SpaceShower:pTdampMatch = 1

# Example set of tuning parameters
SigmaProcess:alphaSvalue           0.1343
BeamRemnants:primordialKThard      1.711
SpaceShower:pT0Ref                 1.823
SpaceShower:pTmaxFudge             1.047
SpaceShower:pTdampFudge            1.492
SpaceShower:alphaSvalue            0.1302
TimeShower:alphaSvalue             0.1166
MultipartonInteractions:pT0Ref     2.953
MultipartonInteractions:alphaSvalue 0.127

```

BeamRemnants:reconnectRange 4.747

ParticleDecays:limitTau0 = on
ParticleDecays:tau0Max = 10

1247 We used these settings to reproduce the original results when necessary and to make full predictions
1248 for parameters selected using the surrogate function. Some of the original data using in the A14
1249 study was private at that time and was only made public later. In a relatively small number of
1250 cases, the public data was in a different form than that used for the original study, so we were
1251 unable to reproduce those predictions.

1252 Typical run card for SHERPA studies using v3.0.0.

```
# general settings
```

```
SHOWER_GENERATOR: CSS  
ANALYSIS: Rivet  
FRAGMENTATION: Ahadic  
INTEGRATION_ERROR: 0.02
```

```
# model parameters
```

```
ALPHAS(MZ): 0.1188  
ORDER_ALPHAS: 2
```

```
# collider setup
```

```
BEAMS: [11, -11]  
BEAM_ENERGIES: 45.6
```

```
# hadronization parameters
```

```
AHADIC:
```

```
KT_0 : 0.9088969039427998  
ALPHA_G : 1.8736652396525728  
ALPHA_L : 1.2518697247467987  
BETA_L : 0.14989272155179253  
GAMMA_L : 0.6832145156132761  
ALPHA_H : 2.840868263919124  
BETA_H : 0.5404054759080933  
GAMMA_H : 0.14984034099619253
```

```
STRANGE_FRACTION : 0.5075082631730515
BARYON_FRACTION  : 0.1357479921139296
P_QS_by_P_QQ_norm : 0.612797404412154
P_SS_by_P_QQ_norm : 0.029994467832440565
P_QQ1_by_P_QQ0   : 1.1896505751927051
```

PARTICLE_DATA:

```
4: {Massive: true}
5: {Massive: true}
```

PARTICLE_CONTAINER:

```
1098: {Name: C, Flavours: [4, -4]}
1099: {Name: B, Flavours: [5, -5]}
```

PROCESSES:

```
- 11 -11 -> 93 93:
    Order: {QCD: 0, EW: 2}
- 11 -11 -> 4 -4:
    Order: {QCD: 0, EW: 2}
- 11 -11 -> 5 -5:
    Order: {QCD: 0, EW: 2}
```

RIVET:

ANALYSES:

```
- SLD_2002_S4869273
- DELPHI_1996_S3430090
- JADE_OPAL_2000_S4300807
- PDG_HADRON_MULTIPLICITIES
```

1253 We used these settings to reproduce the data for our surrogate function and to make full predictions
1254 for parameters selected using the surrogate function.

This manuscript has been authored by an author at Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges, that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>.

Table 28: Bin filtering of A14 data: Shown are the observables from which bins were removed and the number of bins removed. We also show the critical χ^2 values and the χ^2 test statistic before and after bin filtering. If all the bins were removed from the observable then the number of bins removed is shown in bold font and the χ^2 test statistic before and after bin filtering is the same.

Observable Name	No. of filtered bins	$\chi_{c,\mathcal{B}}^2$	$\chi_{\mathcal{B}}^2$ before filtering bins	$\chi_{\mathcal{B}}^2$ after filtering bins
/ATLAS_2011_I919017/d01-x02-y04	11	3.84	9.77	9.77
/ATLAS_2011_I919017/d01-x02-y05	13	7.81	13.51	13.51
/ATLAS_2011_I919017/d01-x02-y13	11	3.84	9.43	9.43
/ATLAS_2011_I919017/d01-x02-y18	11	3.84	6.20	6.20
/ATLAS_2011_I919017/d01-x03-y01	11	21.03	24.00	3.57
/ATLAS_2011_I919017/d01-x03-y02	4	21.03	48.57	19.72
/ATLAS_2011_I919017/d01-x03-y03	2	25.00	28.99	24.72
/ATLAS_2011_I919017/d01-x03-y04	2	32.67	35.36	32.19
/ATLAS_2011_I919017/d01-x03-y06	10	26.30	59.81	26.13
/ATLAS_2011_I919017/d01-x03-y07	7	25.00	58.78	23.91
/ATLAS_2011_I919017/d01-x03-y08	5	28.87	36.98	28.27
/ATLAS_2011_I919017/d01-x03-y09	6	35.17	41.10	35.10
/ATLAS_2011_I919017/d01-x03-y12	3	23.68	31.51	21.33
/ATLAS_2011_I919017/d01-x03-y13	15	31.41	58.77	26.18
/ATLAS_2011_I919017/d01-x03-y14	12	33.92	69.51	32.38
/ATLAS_2011_I919017/d01-x03-y17	3	23.68	30.48	22.60
/ATLAS_2011_I919017/d01-x03-y18	1	30.14	30.65	26.75
/ATLAS_2011_I919017/d01-x03-y19	12	33.92	43.45	6.13
/ATLAS_2011_I919017/d01-x04-y03	22	21.03	45.11	45.11
/ATLAS_2011_I919017/d01-x04-y04	21	19.68	93.99	93.99
/ATLAS_2011_I919017/d01-x04-y05	4	16.92	22.81	16.74
/ATLAS_2011_I919017/d01-x04-y08	22	21.03	65.21	65.21
/ATLAS_2011_I919017/d01-x04-y09	22	21.03	71.99	71.99
/ATLAS_2011_I919017/d01-x04-y10	2	18.31	25.18	18.27
/ATLAS_2011_I919017/d01-x04-y13	12	22.36	49.09	2.36
/ATLAS_2011_I919017/d01-x04-y14	24	23.68	71.30	71.30
/ATLAS_2011_I919017/d01-x04-y15	4	21.03	27.53	20.80
/ATLAS_2011_I919017/d01-x04-y18	2	23.68	23.77	22.57
/ATLAS_2011_I919017/d01-x04-y19	8	22.36	36.75	16.78
/ATLAS_2011_I919017/d01-x04-y25	3	26.30	29.14	24.98
/ATLAS_2011_I919017/d02-x02-y05	1	11.07	13.84	9.87
/ATLAS_2011_I919017/d02-x02-y09	1	9.49	12.32	8.52
/ATLAS_2011_I919017/d02-x02-y14	1	9.49	12.19	9.18
/ATLAS_2011_I919017/d02-x03-y02	15	30.14	40.31	7.63
/ATLAS_2011_I919017/d02-x03-y06	3	31.41	36.64	28.59
/ATLAS_2011_I919017/d02-x03-y07	4	31.41	55.51	29.12
/ATLAS_2011_I919017/d02-x03-y12	7	31.41	45.41	30.04
/ATLAS_2011_I919017/d02-x03-y17	1	30.14	30.64	28.11
/ATLAS_2011_I919017/d02-x04-y03	10	26.30	46.87	19.20
/ATLAS_2011_I919017/d02-x04-y04	25	25.00	136.83	136.83
/ATLAS_2011_I919017/d02-x04-y05	28	28.87	74.75	74.75
/ATLAS_2011_I919017/d02-x04-y08	16	27.59	82.23	25.29
/ATLAS_2011_I919017/d02-x04-y09	27	27.59	156.13	156.13
/ATLAS_2011_I919017/d02-x04-y10	30	31.41	126.00	126.00
/ATLAS_2011_I919017/d02-x04-y13	14	26.30	71.23	23.47
/ATLAS_2011_I919017/d02-x04-y14	27	27.59	103.20	103.20
/ATLAS_2011_I919017/d02-x04-y15	9	28.87	70.47	26.49
/ATLAS_2011_I919017/d02-x04-y18	3	26.30	32.01	25.80
/ATLAS_2011_I919017/d02-x04-y19	13	28.87	67.53	23.91
/ATLAS_2011_I919017/d02-x04-y20	10	28.87	57.69	28.31
/ATLAS_2011_I919017/d02-x04-y24	10	28.87	43.46	28.24
/ATLAS_2011_I919017/d02-x04-y25	3	31.41	39.98	28.20
/ATLAS_2011_ZPT/d02-x01-y01	1	14.07	15.77	14.06
/ATLAS_2011_ZPT/d02-x02-y02	2	14.07	16.94	13.93
/ATLAS_2011_ZPT/d03-x01-y01	1	14.07	15.32	13.85
/ATLAS_2013_JETUE/d08-x01-y03	1	12.59	19.97	11.51

Table 29: Bin filtering of SHERPA data: Shown are the observables from which bins were removed and the number of bins removed. We also show the critical χ^2 values and the χ^2 test statistic before and after bin filtering. If all the bins were removed from the observable then the number of bins removed is shown in bold font and the χ^2 test statistic before and after bin filtering is the same.

Observable Name	No. of filtered bins	$\chi_{c,B}^2$	χ_B^2 before filtering bins	χ_B^2 after filtering bins
/DELPHI_1996_S3430090/d02-x01-y01	17	9.49	35.19	35.19
/DELPHI_1996_S3430090/d04-x01-y01	17	9.49	24.89	24.89
/DELPHI_1996_S3430090/d06-x01-y01	21	15.51	41.59	41.59
/DELPHI_1996_S3430090/d07-x01-y01	22	16.92	83.91	83.91
/DELPHI_1996_S3430090/d08-x01-y01	26	22.36	80.11	80.11
/DELPHI_1996_S3430090/d10-x01-y01	2	11.07	15.90	10.59
/DELPHI_1996_S3430090/d11-x01-y01	20	14.07	94.30	94.30
/DELPHI_1996_S3430090/d16-x01-y01	14	3.84	17.63	17.63
/DELPHI_1996_S3430090/d18-x01-y01	23	18.31	101.31	101.31
/DELPHI_1996_S3430090/d19-x01-y01	21	15.51	59.12	59.12
/DELPHI_1996_S3430090/d20-x01-y01	16	7.81	20.48	20.48
/DELPHI_1996_S3430090/d33-x01-y01	5	52.19	75.18	50.31

Table 30: A14 results for the *full data set* when using the *cubic polynomial* approximation. Smaller numbers are better. The best results are indicated in bold.

Method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1290	0.5358	-66.0364
Bilevel-medianscore	0.1645	0.4114	-70.0545
Bilevel-portfolio	0.1900	0.6590	-63.0378
Expert	0.1306	0.5466	-68.6511
All-weights-equal	0.1034	0.5553	-65.6099
Robust optimization	0.0697	0.9749	-66.7931

Table 31: A14 results for the *observable-filtered data* when using the *cubic polynomial* approximation (12 observables were filtered out and not used during the optimization). Smaller numbers are better. Best results are indicated in bold. All values are computed over all 406 observables.

Method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1079	0.8082	-61.9210
Bilevel-medianscore	0.1702	0.4955	-66.6920
Bilevel-portfolio	0.1764	0.7408	-61.3839
Expert	0.1306	0.5466	-68.6511
All-weights-equal	0.1049	0.6689	-63.6502
Robust optimization	0.0829	1.0574	-66.3665

Table 32: A14 results for the *bin-filtered data* when using the *cubic polynomial* approximation. 1811 out of 7010 total bins were filtered out and not used during the optimization. Smaller numbers are better. Best results are indicated in bold. All results are computed over all bins.

Method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1244	0.7147	-64.7848
Bilevel-medianscore	0.2171	0.5433	-69.7202
Bilevel-portfolio	0.1159	0.5205	-70.1573
Expert	0.1306	0.5466	-68.6511
All-weights-equal	0.1406	0.4122	-69.2732
Robust optimization	0.1234	0.8075	-67.1015

Table 33: Optimal parameter values for the A14 dataset obtained when using all observables in the optimization and the *cubic polynomial* approximation.

ID	Parameter name	Expert	Bil.-meanscore	Bil.-medianscore	Bil.-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.139	0.141	0.140	0.136	0.138
2	<code>BeamRemnants:primordialkThard</code>	1.904	1.867	1.884	1.866	1.826	1.862
3	<code>SpaceShower:pT0Ref</code>	1.643	1.632	1.735	1.651	1.395	1.603
4	<code>SpaceShower:pTmaxFudge</code>	0.908	0.939	0.904	0.988	0.933	0.944
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.079	1.069	1.047	1.063	1.067
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.129	0.130	0.130	0.128	0.129
7	<code>TimeShower:alphaSvalue</code>	0.128	0.123	0.124	0.121	0.136	0.124
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	2.083	2.065	2.039	1.925	2.092
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.127	0.127	0.126	0.120	0.127
10	<code>BeamRemnants:reconnectRange</code>	1.792	1.531	1.405	1.591	2.567	1.636
Euclidean distance from the expert solution		0.246	0.235	0.428	0.451	0.259	

Table 34: Optimal parameter values for A14 when using the *cubic polynomial* approximation with all methods after outlier detection to filter out observables that cannot be approximated well by the model.

ID	Parameter name	Expert	Bilevel-meanscore	Bilevel-medianscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.136	0.141	0.137	0.136	0.137
2	<code>BeamRemnants:primordialkThard</code>	1.904	1.793	1.853	1.754	1.829	1.772
3	<code>SpaceShower:pT0Ref</code>	1.643	1.329	1.369	1.218	1.425	1.301
4	<code>SpaceShower:pTmaxFudge</code>	0.908	1.079	1.088	1.223	0.926	1.085
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.069	1.053	1.101	1.065	1.074
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.129	0.128	0.129	0.129	0.129
7	<code>TimeShower:alphaSvalue</code>	0.128	0.124	0.123	0.116	0.136	0.124
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	1.971	2.098	1.870	1.971	1.983
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.122	0.126	0.120	0.121	0.123
10	<code>BeamRemnants:reconnectRange</code>	1.792	1.812	1.614	1.714	2.632	1.851
Euclidean distance from the expert solution			0.447	0.279	0.553	0.432	0.480

Table 35: Optimal parameter values obtained for A14 with the *cubic polynomial* approximation with all methods after using the bin-filtering approach that excludes individual bins from the optimization.

ID	Parameter name	Expert	Bilevel-meanscore	Bilevel-medianscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	<code>SigmaProcess:alphaSvalue</code>	0.143	0.141	0.143	0.136	0.136	0.132
2	<code>BeamRemnants:primordialkThard</code>	1.904	1.919	1.918	1.575	1.794	1.716
3	<code>SpaceShower:pT0Ref</code>	1.643	1.802	2.284	2.300	1.355	2.123
4	<code>SpaceShower:pTmaxFudge</code>	0.908	0.968	1.014	0.920	0.856	0.843
5	<code>SpaceShower:pTdampFudge</code>	1.046	1.071	1.147	1.442	1.047	1.465
6	<code>SpaceShower:alphaSvalue</code>	0.123	0.130	0.130	0.144	0.132	0.143
7	<code>TimeShower:alphaSvalue</code>	0.128	0.129	0.127	0.131	0.138	0.130
8	<code>MultipartonInteractions:pT0Ref</code>	2.149	2.059	1.800	2.228	1.925	2.306
9	<code>MultipartonInteractions:alphaSvalue</code>	0.128	0.126	0.120	0.131	0.118	0.131
10	<code>BeamRemnants:reconnectRange</code>	1.792	1.860	1.922	1.807	2.340	1.622
Euclidean distance from the expert solution			0.376	0.354	0.848	0.525	1.111

Table 36: Comparison of the optimal weights obtained by each method using the *cubic polynomial* approximation. The observable grouping corresponds to the same grouping used in [3].

	expert	Bilevel- meanscore	Bilevel- medianscore	Bilevel- portfolio	robustopt
Track jet properties					
Charged jet multiplicity (50 distributions)	10	10.74	14.98	10.64	19.38
Charged jet z (50 distributions)	10	11.29	8.66	13.71	0.00
Charged jet p_T^{rel} (50 distributions)	10	11.20	10.39	10.99	0.00
Charged jet $\rho_{ch}(r)$ (50 distributions)	10	11.57	10.58	12.55	0.00
Jet shapes					
Jet shape ρ (59 distributions)	10	11.57	11.06	10.20	19.38
Dijet decorr					
Decorrelation $\Delta\phi$ (Fit range: $\Delta\phi > 0.75$) (9 distributions)	20	12.39	8.37	9.39	15.07
Multijets					
3-to-2 jet ratios (8 distributions)	100	12.99	27.19	5.88	19.38
p_T^Z (Fit range: $p_T^Z < 50\text{GeV}$)					
Z-boson p_T (20 distributions)	10	12.78	14.53	6.71	19.38
Substructure					
Jet mass, $\sqrt{d_{12}}, \sqrt{d_{23}}, \tau_{21}, \tau_{23}$ (36 distributions)	5	10.55	9.91	9.74	15.61
$t\bar{t}$ gap					
Gap fraction vs Q_0, Q_{sum} for $ y < 0.8$	100	0.18	2.10	3.88	19.38
Gap fraction vs Q_0, Q_{sum} for $0.8 < y < 1.5$	80	0.75	9.52	5.71	19.38
Gap fraction vs Q_0, Q_{sum} for $1.5 < y < 2.1$	40	7.93	8.31	39.20	19.38
Gap fraction vs Q_0, Q_{sum} for $ y < 2.1$	10	18.19	13.43	11.05	19.38
Track-jet UE					
Transverse region N_{ch} profiles (5 distributions)	10	15.87	13.45	13.53	19.38
Transverse region mean p_T profiles for $R = 0.2, 0.4, 0.6$ (3 distributions)	10	7.56	11.72	10.30	19.38
$t\bar{t}$ jet shapes					
Jet shapes $\rho(r), \psi(r)$ (20 distributions)	5	10.86	10.91	12.25	10.66
Jet UE					
Transverse, trans-max, trans-min sum p_T incl. profiles (3 distributions)	20	12.76	22.51	9.65	19.38
Transverse, trans-max, trans-min N_{ch} incl. profiles (3 distributions)	20	15.57	9.65	6.01	19.38
Transverse sum E_T incl. profiles (2 distributions)	20	12.71	12.75	25.03	3.73
Transverse sum $ET/\text{sum } p_T$ ratio incl., excl. profiles (2 distributions)	5	7.53	18.29	28.35	19.38
Transverse mean p_T incl. profiles (2 distributions)	10	7.65	7.45	13.34	19.38
Transverse, trans-max, trans-min sum p_T incl. distributions (15 distributions)	1	9.39	5.50	11.04	19.38
Transverse, trans-max, trans-min sum N_{ch} incl. distributions (15 distributions)	1	11.92	9.85	14.52	19.38

Table 37: Optimal parameter values for the SHERPA dataset obtained with all methods using the *cubic polynomial* approximation when no filtering was applied before optimization (88 observables).

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.850	0.837	0.903	0.870	0.853
2	ALPHA_G	1.25	0.626	0.626	0.626	1.874	0.626
3	ALPHA_L	2.50	3.634	2.022	3.108	1.252	3.749
4	BETA_L	0.10	0.150	0.069	0.050	0.150	0.150
5	GAMMA_L	0.50	0.250	0.353	0.750	0.619	0.286
6	ALPHA_H	2.50	3.455	2.047	1.251	2.712	3.454
7	BETA_H	0.75	0.736	0.610	0.657	0.573	0.922
8	GAMMA_H	0.10	0.144	0.124	0.050	0.150	0.140
9	STRANGE_FRACTION	0.50	0.531	0.521	0.529	0.514	0.497
10	BARYON_FRACTION	0.18	0.099	0.132	0.091	0.139	0.104
11	P_QS_by_P_QQ_norm	0.48	0.720	0.617	0.502	0.601	0.720
12	P_SS_by_P_QQ_norm	0.02	0.010	0.030	0.030	0.030	0.010
13	P_QQ1_by_P_QQ0	1.00	1.499	1.499	1.349	1.164	1.499
Euclidean distance from the default solution			1.508	1.130	1.400	1.236	1.497

Table 38: Optimal parameter values for the SHERPA dataset obtained with all methods using the *cubic polynomial* approximation after using outlier detection to remove observables from the optimization (3 observables removed).

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.898	0.834	0.946	0.945	0.853
2	ALPHA_G	1.25	1.136	0.751	0.626	1.874	0.942
3	ALPHA_L	2.50	1.454	2.088	3.749	3.749	2.275
4	BETA_L	0.10	0.050	0.050	0.050	0.050	0.136
5	GAMMA_L	0.50	0.409	0.305	0.627	0.626	0.553
6	ALPHA_H	2.50	3.748	1.358	1.251	1.533	1.804
7	BETA_H	0.75	0.406	0.375	0.591	1.125	0.760
8	GAMMA_H	0.10	0.078	0.150	0.086	0.050	0.066
9	STRANGE_FRACTION	0.50	0.541	0.528	0.552	0.529	0.553
10	BARYON_FRACTION	0.18	0.181	0.139	0.090	0.270	0.270
11	P_QS_by_P_QQ_norm	0.48	0.240	0.602	0.449	0.384	0.298
12	P_SS_by_P_QQ_norm	0.02	0.020	0.025	0.030	0.030	0.023
13	P_QQ1_by_P_QQ0	1.00	1.499	1.499	1.499	0.639	0.837
Euclidean distance from the default solution			1.222	1.327	1.378	1.463	0.937

Table 39: Optimal parameter values for the SHERPA dataset obtained with all methods using the *cubic polynomial* approximation after using the bin-filtering approach that excludes individual bins from the optimization (204 bins out of 5246 total bins were removed).

ID	Parameter name	Default	Bilevel-meanscore	Bilevel-medscore	Bilevel-portfolio	Robust opt	All-weights-equal
1	KT_0	1.00	0.866	0.820	0.897	0.950	0.911
2	ALPHA_G	1.25	0.626	1.114	0.626	1.874	0.626
3	ALPHA_L	2.50	3.749	3.502	2.216	3.749	3.749
4	BETA_L	0.10	0.079	0.053	0.050	0.050	0.078
5	GAMMA_L	0.50	0.383	0.325	0.750	0.627	0.367
6	ALPHA_H	2.50	1.251	1.251	1.251	1.527	1.251
7	BETA_H	0.75	0.738	0.675	0.694	1.125	0.741
8	GAMMA_H	0.10	0.092	0.116	0.099	0.050	0.104
9	STRANGE_FRACTION	0.50	0.536	0.547	0.543	0.529	0.541
10	BARYON_FRACTION	0.18	0.120	0.127	0.129	0.270	0.130
11	P_QS_by_P_QQ_norm	0.48	0.636	0.578	0.472	0.384	0.569
12	P_SS_by_P_QQ_norm	0.02	0.030	0.030	0.030	0.030	0.030
13	P_QQ1_by_P_QQ0	1.00	1.499	1.499	1.499	0.637	1.499
Euclidean distance from the default solution			1.263	1.215	1.272	1.464	1.224

Table 40: Comparison metrics for the full SHERPA dataset when using the *cubic polynomial* approximation: Weighted χ^2 , A-optimality, D-optimality of different tuning methods. Smaller numbers are better and the best results are in bold. Note that we do not have an expert solution for this dataset.

method	Weighted χ^2	A-optimality	D-optimality (log)
Bilevel-meanscore	0.1777	9.0959	-39.9863
Bilevel-medscore	0.2370	13.3943	-37.1420
Bilevel-portfolio	0.3409	8.7863	-39.6956
All-weights-equal	0.2305	6.8732	-42.0678
Robust optimization	0.0507	56.9168	-21.9561

Table 41: Comparison metrics for the SHERPA dataset after filtering out outlier observables and using the *cubic polynomial* approximation: Weighted χ^2 , A-optimality, D-optimality of different tuning methods. 3 observables were filtered out and not used during the optimization and results are computed based on all observables. Smaller numbers are better and the best results are indicated in bold. Note that we do not have an expert solution for this dataset.

method	Weighted χ^2	A-optimality	D-optimality(log)
Bilevel-meanscore	0.4740	14.3374	-35.2608
Bilevel-medscore	0.4786	13.6299	-36.6594
Bilevel-portfolio	0.2139	10.4481	-36.8254
All-weights-equal	0.4789	28.2419	-28.1536
Robust optimization	0.0093	94.7811	-23.5723

Table 42: Comparison metrics for the SHERPA dataset after bin-filtering and using the *cubic polynomial* approximation: Weighted χ^2 , A-optimality, D-optimality of the different tuning methods. 204 out of 5426 bins were filtered out and not used during the optimization, and results were computed over all bins. Smaller numbers are better and the best results are indicated in bold. Note that we do not have an expert solution for this dataset.

method	Weighted χ^2	A-optimality	D-optimality(log)
Bilevel-meanscore	0.2504	17.2334	-32.7683
Bilevel-medscore	0.1835	16.9248	-32.1289
Bilevel-portfolio	0.2906	13.3500	-36.3598
All-weights-equal	0.1928	10.4897	-37.0305
Robust optimization	0.0364	72.5601	-26.8516

Table 43: Eigentune results for the A14 dataset using the optimal physics parameters \mathbf{p}^* obtained with the different optimization methods when using the *cubic polynomial* approximation.

Parameters	Expert		Bilevel-meanscore		Bilevel-medianscore		Bilevel-portfolio		Robust optimization	
	min	max	min	max	min	max	min	max	min	max
SigmaProcess:alphaSvalue	0.072	0.196	0.071	0.197	0.079	0.190	0.076	0.191	0.079	0.187
BeamRemnants:primordialKThard	1.899	1.904	1.849	1.888	1.877	1.894	1.855	1.881	1.764	1.895
SpaceShower:pT0Ref	1.616	1.633	1.622	1.640	1.733	1.737	1.631	1.667	1.377	1.411
SpaceShower:pTmaxFudge	0.904	0.914	0.938	0.940	0.884	0.923	0.986	0.990	0.932	0.935
SpaceShower:pTdampFudge	1.039	1.047	1.059	1.102	1.053	1.085	1.045	1.049	1.061	1.064
SpaceShower:alphaSvalue	0.116	0.128	0.128	0.130	0.118	0.141	0.129	0.131	0.128	0.129
TimeShower:alphaSvalue	0.076	0.199	0.034	0.223	0.046	0.205	0.083	0.145	0.042	0.198
MultipartonInteractions:pT0Ref	1.749	2.666	1.533	2.707	1.536	2.621	1.989	2.116	1.866	1.965
MultipartonInteractions:alphaSvalue	0.045	0.186	0.095	0.154	0.114	0.140	0.044	0.180	0.100	0.133
BeamRemnants:reconnectRange	1.719	1.719	1.523	1.541	1.390	1.420	1.589	1.595	2.565	2.568

Table 44: Eigentune results for the SHERPA dataset using the optimal physics parameters \mathbf{p}^* obtained with the different optimization methods when using the *cubic polynomial* approximation.

Parameters	Bilevel-meanscore		Bilevel-medianscore		Bilevel-portfolio		Robust optimization	
	min	max	min	max	min	max	min	max
KT_0	0.572	1.845	0.818	0.884	0.798	1.002	0.350	1.021
ALPHA_G	0.113	0.769	0.472	0.690	0.612	0.639	1.288	2.044
ALPHA_L	3.468	4.227	1.956	2.181	2.917	3.309	0	1.697
BETA_L	0	0.255	0	0.487	0	0.305	0	0.233
GAMMA_L	0.064	0.915	0.226	0.405	0.746	0.755	0.328	1.625
ALPHA_H	2.981	3.587	2.000	2.162	1.235	1.268	2.427	2.898
BETA_H	0.662	0.771	0.582	0.677	0.637	0.675	0	0.741
GAMMA_H	0.045	0.190	0.070	0.255	0	0.134	0	0.652
STRANGE_FRACTION	0.068	0.749	0.446	0.655	0.501	0.558	0.413	0.546
BARYON_FRACTION	0	0.335	0.117	0.166	0	0.186	0.030	0.516
P_QS_by_P_QQ_norm	0.669	0.828	0.576	0.715	0.458	0.549	0.537	0.619
P_SS_by_P_QQ_norm	0	0.087	0	0.105	0	0.076	0	0.050
P_QQ1_by_P_QQ0	1.496	1.508	1.498	1.500	1.348	1.349	1.153	1.200