A response-matrix-centred approach to presenting cross-section measurements

L. Koch

Published version information

Citation: L Koch. "A response-matrix-centred approach to presenting cross-section measurements." Journal of Instrumentation, vol. 14, no. 09 (2019): P09013.

DOI: 10.1088/1748-0221/14/09/P09013

Prepared for submission to JINST

A response-matrix-centred approach to presenting cross-section measurements

L. Koch

RWTH Aachen University,
III. Physikalisches Insitut B,
Aachen, Germany
STFC Rutherford Appleton Laboratory,
Particle Physics Department,
Didcot, United Kingdom

E-mail: lukas.koch@stfc.ac.uk

Abstract: The current canonical approach to publishing cross-section data is to unfold the reconstructed distributions. Detector effects like efficiency and smearing are undone mathematically, yielding distributions in true event properties. This is an ill-posed problem, as even small statistical variations in the reconstructed data can lead to large changes in the unfolded spectra.

This work presents an alternative or complementary approach: the response-matrix-centred forward-folding approach. It offers a convenient way to forward-fold model expectations in truth space to reconstructed quantities. These can then be compared to the data directly, similar to what is usually done with full detector simulations within the experimental collaborations. For this, the detector response (efficiency and smearing) is parametrised as a matrix. The effects of the detector on the measurement of a given model is simulated by simply multiplying the binned truth expectation values by this response matrix.

Systematic uncertainties in the detector response are handled by providing a set of matrices according to the prior distribution of the detector properties and marginalising over them. Background events can be included in the likelihood calculation by giving background events their own bins in truth space.

To facilitate a straight-forward use of response matrices, a new software framework has been developed: the Response Matrix Utilities (`ReMU`). `ReMU` is a Python package distributed via the Python Package Index. It only uses widely available, standard scientific Python libraries and does not depend on any custom experiment-specific software. It offers all methods needed to build response matrices from Monte Carlo data sets, use the response matrix to forward-fold truth-level model predictions, and compare the predictions to real data using Bayesian or frequentist statistical inference.

Keywords: Analysis and statistical methods, Data reduction methods, Software architectures (event data models, frameworks and databases)

Contents

1. Motivation 1

2. Measurement strategy 4
- 2.1. Introduction 4
- 2.2. The detector model 5
- 2.3. The likelihood 7
- 2.4. Backgrounds 8
- 2.5. Matrix tests 10

3. Implementation 12
- 3.1. Building the detector response matrices 12
- 3.2. Binning 16
  - 3.2.1. General considerations 16
  - 3.2.2. Choosing the variables to bin in 17
  - 3.2.3. Bin widths 17
  - 3.2.4. Empty bins 18
- 3.3. Software and data format 19

4. Example analysis 20
- 4.1. Introduction 20
- 4.2. Preparation of the response matrix 21
- 4.3. Using the response matrix 21

5. Conclusions 25

References 27

A. Statistical methods 29
- A.1. Simple hypotheses and absolute maximum likelihood 29
- A.2. Likelihood ratio testing 29
- A.3. Composite hypotheses 30
- A.4. Parameter estimation 31
- A.5. Profile plug-in p-values 31
- A.6. Bayesian posterior sampling 32

1. Motivation

Cross-section measurements are an important tool for investigating possible manifestations of "new physics", i.e. phenomena beyond the currently accepted models. This is either done directly with the

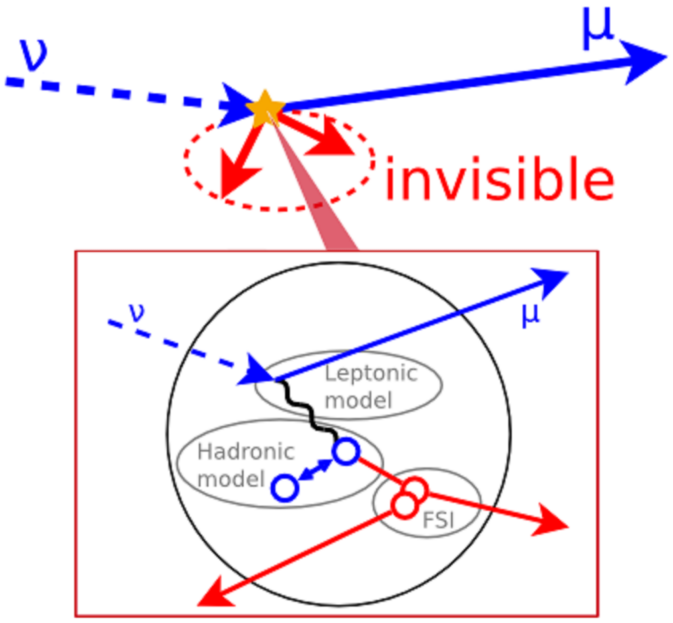

Figure 1. Neutrino cross sections for oscillation experiments. Neutrino oscillation experiments need to reconstruct the neutrino energy event by event. Since the neutrino is invisible, only the products of the interaction can be used for this. Exact reconstructions are made impossible by undetectable particles below the detection threshold of the respective detectors. Models for electroweak nuclear interactions are used to correct these effects. Currently there are quite large theoretical uncertainties on especially the hadronic model (initial state of the nucleus, nucleon form factors, etc.) and the *Final State Interactions* (FSI). Cross-section measurements play an important role in constraining these uncertainties (see e.g. [1]).

cross-section measurements, e.g. in beyond-the-standard-model searches at collider experiments, or by using cross-section measurements as inputs for other experiments, e.g. the use of neutrino cross-section measurements for constraining systematic uncertainties in oscillation experiments (see figure 1).

It is thus important to present these results in a way that allows re-interpretation of the data when new insights into the theoretical models are gained in the future. Especially in the case of neutrino cross-section measurements it can be difficult to disentangle the measured quantities from detector effects and (possibly poorly motivated) model assumptions (see [2] for an overview of challenges and possible solutions). Measurements that do not take care of these issues can end up being difficult to interpret and ultimately become useless for global data comparisons, fits, etc.

There is no single recipe that ensures that a measurement is free of model assumptions or detector effects. A couple of points are important to keep in mind though. For example, when trying to constrain a certain aspect of an electroweak nuclear interaction model, it is important to make sure that no assumptions of that model are influencing the measurement. In the worst case, one can end up publishing "data" that is really just a carbon copy of the model. Checks against these kinds of model bias are a common part of physics analyses.

It is equally important though, to ensure that (possibly implicit) assumptions about model effects one is *not* interested in do not affect the measurement. The detection efficiency and reconstruction

resolution of an event in a real detector can depend on a lot of variables. In principle, it depends on the type, momentum and direction of every single particle that leaves a vertex and can theoretically be detected. In practice, one is interested in the properties of only a few of those particles. Even when considering just one particle, it is often not fully characterised by its three momentum components, but the information is reduced to, for example, the magnitude of the momentum to avoid bins with very few events in n-dimensional histograms.

Unfortunately, not looking at the other variables does not mean that their influence on the detector efficiency goes away. Models that are well tuned to real data in the distribution of a certain quantity, like the total lepton momentum in a charged-current neutrino interaction, can differ wildly from reality in distributions that simply have not been looked at before, e.g. the second highest proton momentum in an event. Ignoring these quantities means that one uses the average efficiency of the events, obtained for a certain distribution of the ignored quantities.

This can lead to very different efficiencies and purities of event selections if two theories predict very different distributions. For example, all detectors have certain energy/momentum thresholds below which they are not sensitive to particles. If two theories (or a theory and reality for that matter) now predict different fractions of events/particles below that threshold, the resulting average efficiency of selecting the events will vary accordingly.

A lot of work is done on minimising or at least quantifying these effects. Strategies range from doing multi-dimensional differential cross-section measurements (to ensure all dependencies of the efficiency are modelled), to repeating the analysis with multiple theories and simply quoting how much the results depend on the used model. The former approach requires a lot of data to have a significant number of events in every bin, while the latter suffers from the uncertainty of whether all available models even cover reality at all.

The response-matrix-centred method described in this work aims to combine the model independence of the multi-dimensional approach with the ability to work with low number of events of the naive model test. This is achieved by de-coupling the binning of the reconstructed events from the description of the events at the generator level. The high dimensionality of variables is only needed in *truth space*, i.e. the description of the events at the generator level. The actual recorded data can be binned much coarser in *reco space*, i.e. with wider binning and/or fewer reconstructed variables. The response matrix is the connecting piece between the two, describing how likely an event in a particular truth space bin is going to end up in any of the reco space bins.

If the truth binning is chosen carefully, the response matrix should be (sufficiently[1]) independent of any assumed physics model of the interactions. That is, different models can predict different truth space distributions, but the values of the response matrix elements do not depend on the model that is used to build the matrix[2]. The real data and response matrix can then be used with arbitrary models to calculate a likelihood and extract cross sections.

This is so far not different from the naive model testing method. The advantage of the response matrix approach is realised when considering the matrix and the raw data as the main result of the measurement. They are (ideally) independent of any model assumptions and can be used to test any new model or model improvement that will be developed in the future. Furthermore, if the

[1] A certain dependence on the event distribution within the bins will always remain, but it can be reduced to the point where it does not matter compared to other uncertainties.

[2] Aside from statistical effects from the number of available simulated events in each truth bin.

raw data and response matrix are published, model developers can use them directly to test new models against old data. Compared to the classical approach, where the theories are developed by theorists and then tested within the experimental groups in dedicated analyses, this reduces the time of the development cycle considerably. In fact, a lot of work has been spent to make old experimental results available for easy model tuning, for example with the NUISANCE[3] or Rivet[4] frameworks. Results obtained with the response-matrix-centred approach would be very easy to include in such global fits.

It might seem like a shortcut for lazy experimental physicists to simply publish the raw data and response matrix to leave the rest to the model builders. This is not the case though, since the construction of the response matrix requires exactly the same understanding of the detector and care to cover all systematics as a classical, unfolding analysis. Also it is unlikely that any experimental group would publish the data and response matrix without also using them for their own model tests.

It is worth noting that model comparisons in reconstructed (or smeared) space are in general more powerful than equivalent model tests in truth (or unfolded) space [5]. The forward-folding approach might thus also be advantageous for analyses that *do* have enough statistics to do a multi-dimensional unfolding of the results. In any case, one is not restricted to do one or the other. If the data, time and person-power allow it, it might be the best choice to publish both an unfolded result, as well as the raw data with a response matrix. The additional work needed for doing an unfolding analysis on top of a forward folding one is probably less than it might seem. The response matrix can be used to do an unfolding analysis with it, e.g. using a likelihood fit or Markov Chain Monte Carlo with the bin-by-bin truth-level predictions as fit parameters. If the model-independence criterion of the forward-folding matrix leads to very underconstrained truth bins (i.e. a much finer truth binning than in reconstructed space), the dimensionality can be reduced by fitting templates of a theory. This would mean the result is no longer model-independent, but it could be argued that a purely unfolding analysis should suffer from the same problems.

The description of the mathematical model of the response-matrix-centred approach can be found in section 2. Details on how to build the matrix and how to contain the knowledge about the systematic uncertainties in it are given in section 3. The algorithms are implemented in a Python software library called `ReMU`, Response Matrix Utilities. It is intended to make the usage of the data and response matrix as easy as possible. More informations about the software and data formats are included in section 3.3.

2. Measurement strategy

2.1. Introduction

The response-matrix-centred approach is a way of presenting cross-section measurements (or any other kind of counting experiment) in a way that tries to be as model-independent as possible. Its main philosophy can be summarised in three main points:

1. There is a linear relationship between “true” physics expectation values, i.e. as described on the generator level, and expected number of measured events.

2. Our knowledge of that relationship is imperfect.

3. The data is the data is the data.

The linear relationship mentioned in the first point is the response matrix. It describes how likely it is to count an event that happened in the detector (efficiency) and in which reconstructed bin it is probably going to end up, i.e. what the reconstructed properties of the event will be (smearing). We know the elements of this matrix only to a certain precision. They are subject to uncertainties of evaluating them using Monte Carlo (MC) simulations of events in the detector..

The actually measured data on the other hand is the only thing we can be 100% sure about. It consists of exact numbers, and systematic or even statistical errors only apply if one interprets the actual data as expectation values for future measurements. For example. if we do a cross-section measurement and measure 16 events of a certain type, we measure *exactly* 16 events, not something between 12 and 20. Once we try to predict future repetitions of the experiment, we have to interpret this number as measurement of the expectation value, so we get an uncertainty on that: the expectation value is 16 ± 4. In general, there is no one-to-one correspondence between data and the physics variables we are interested in, so the response matrix must be used to translate between the two.

Ideally, a measurement should remain useful not only for the current interaction model, but also for all possible future models. This can be achieved if:

- Arbitrary models can be checked for compatibility with the published data.
- The publication contains all tools and information to do this.
- These tools do not depend on the currently favoured model.

All of this is possible with the response-matrix-centred approach (see figure 2).

The idea of bringing truth-space expectation values into the reconstructed space is not new. This is regularly done in most experiments. Unfortunately this usually requires expert knowledge and access to the (simulation) software stack of the experiment, which is often only available to collaborators. Simplifying the process by providing a response matrix could enable non-collaborators to bring their models into the reco space of the experiment, as well as speed up the process of testing models within a collaboration.

The main result of any measurement presented in this way consists of the *raw* reconstructed data (without any systematic errors) and the response matrix including all uncertainties on the matrix elements. These two objects are everything that is needed to test arbitrary physics models against the data in a consistent way. The tests then produce "traditional" results in truth space.

2.2. The detector model

We categorise all events by their true properties and sort them into a set of truth bins. For simulated events these properties are directly accessible, while for real data they remain hidden. Selected events are also binned according to their reconstructed properties. The Poisson expectation value for the number of events in the j-th truth bin is μ_j. It is determined by the underlying physics models and the experimental setup, e.g. target material, mass and integrated neutrino flux.

If an event happens in truth bin j, it has a certain probability $P(j \rightarrow i)$ to be selected and reconstructed in the i-th reconstruction bin. This probability can be calculated from MC samples

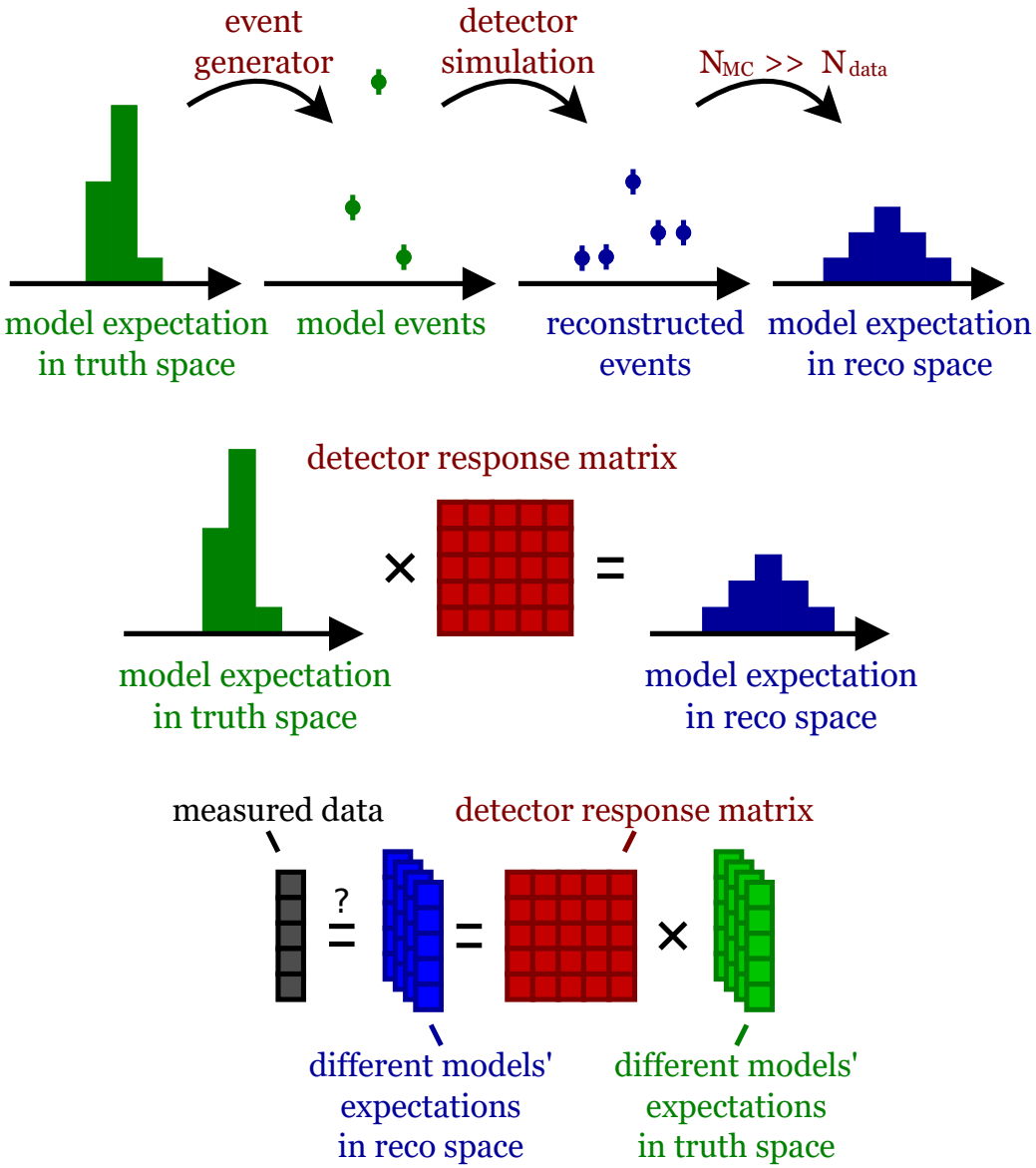

Figure 2. The response-matrix-centred approach. The aim of the approach presented here is to replace the computationally intensive full detector simulation (top) with a much simpler matrix multiplication (centre). This would allow a much faster test of different models against the data (bottom).

with known true and reconstructed properties:

$$P(j \to i) = \lim_{N\to\infty} \frac{N(\text{truth} = j, \text{reco} = i)}{N(\text{truth} = j)}. \tag{2.1}$$

It should depend *only* on the detector properties and *not* on the interaction model. This can be achieved by choosing an appropriate binning in truth space (see section 3.2). The expectation value for the i-th reconstruction bin ν_i is then

$$\nu_i = \sum_j P(j \to i)\mu_j. \tag{2.2}$$

This can be expressed as a matrix product (using Einstein notation)

$$\nu_i = R_{ij}\mu_j, \tag{2.3}$$

where R is the detector response matrix. Please note that this matrix models both the selection efficiency and reconstruction smearing.

Truth and reconstructed space will generally be binned in multiple variables, which might give the impression that the response matrix needs to be an n-dimensional object. This is not the case, as both binnings can be linearised, i.e. assigning each bin an identifying integer.[3] The response matrix should now be seen as a regular, two-dimensional matrix that translates between the two one-dimensional vectors of bins.

Since we need to know the truth information, R can only be built from MC samples. Unfortunately the simulated detector does not mirror the real detector perfectly. The differences are parametrised in a set of systematic uncertainties, e.g. an uncertainty on the momentum resolution or the track reconstruction efficiency. Their effect on the response matrix can be evaluated by producing lots of "toy simulations",[4] in which the same dataset is processed, but the detector properties are sampled from their uncertainty distribution.

Because it is often impractical to run the full detector simulation hundreds of times, these toy simulations are commonly created by modifying a single full "baseline" simulation. Depending on the type of systematic uncertainty, this can be done by assigning weights to events, or by varying the reconstructed properties of the events. This yields a set of N_{toy} response matrices R^t, each describing one possible true detector and its reconstruction expectation values:

$$\nu_i^t = R_{ij}^t \mu_j. \tag{2.4}$$

2.3. The likelihood

One way to measure the compatibility of a given hypothesis and the measured data is the likelihood L. For a discrete counting experiment, it describes the probability of getting exactly the measured result $\boldsymbol{n}$, given the tested hypothesis $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta}) = P(\boldsymbol{n}|\boldsymbol{\theta}). \tag{2.5}$$

In our framework, the hypothesis is described by the expectation values of the truth bins $\boldsymbol{\mu}$:

$$L(\boldsymbol{\mu}) = P(\boldsymbol{n}|\boldsymbol{\mu}). \tag{2.6}$$

We can expand this expression to explicitly include the possibility of different detector responses:

$$P(\boldsymbol{n}|\boldsymbol{\mu}) = \int_R P(\boldsymbol{n}|\boldsymbol{\mu}, R) f(R) \mathrm{d}R, \tag{2.7}$$

[3] In fact, the binning does not even have to be "regular" in any way. The bins can have arbitrary shapes in arbitrary dimensions. The only thing that is demanded of the bins is that they do not overlap, so each event is assigned exactly one truth bin and at most one reco bin.

[4] Depending on the collaboration, these are also called "universes".

where the integral is over all possible detectors R and the probability density $f(R)$ of them being true. This is impractical, but we can replace the infinite, high-dimensional integral with a random sample of toy detectors R^t:

$$P(\boldsymbol{n}|\boldsymbol{\mu}) = \frac{1}{N_{\text{toy}}} \sum_t P(\boldsymbol{n}|\boldsymbol{\mu}, R^t). \tag{2.8}$$

The sample is drawn from the uncertainty distributions of the detector properties f, so more-probable matrices will appear more often than unlikely ones. Within the set of toy matrices, each one is equally likely.

The remaining probability term is just that of a multi-bin Poisson counting experiment:

$$\begin{aligned} P(\boldsymbol{n}|\boldsymbol{\mu}, R^t) &= P_{\text{Poisson}}(\boldsymbol{n}|\boldsymbol{\nu} = R^t \cdot \boldsymbol{\mu}) \\ &= \prod_i \frac{(R^t_{ij}\mu_j)^{n_i}}{n_i!} \exp(-R^t_{ij}\mu_j) \end{aligned} \tag{2.9}$$

So ultimately the total marginal likelihood of a tested hypothesis, given the measured data, is

$$L(\boldsymbol{\mu}) = P(\boldsymbol{n}|\boldsymbol{\mu}) = \frac{1}{N_{\text{toy}}} \sum_t \prod_i \frac{(R^t_{ij}\mu_j)^{n_i}}{n_i!} \exp(-R^t_{ij}\mu_j). \tag{2.10}$$

Alternatively one can also choose to use the profile likelihood

$$L_{\text{profile}}(\boldsymbol{\mu}) = \max_t \prod_i \frac{(R^t_{ij}\mu_j)^{n_i}}{n_i!} \exp(-R^t_{ij}\mu_j), \tag{2.11}$$

which just selects the toy migration matrix with the highest resulting likelihood.

Using the profile likelihood with a discrete set of toy matrices can lead to results that are very dependent on the number of simulated toys, e.g. when the parameters of the systematics are not strictly bound. If a parameter of the matrix is distributed without strict limits[5] and the maximum likelihood is achieved for very extreme matrices, the achieved likelihood will depend a lot on the number of toy matrices (see figure 3). The more matrices are sampled from the unlimited distribution, the more extreme the most extreme matrix will become. If, on the other hand, all parameters are sampled from bounded distributions[6], the extremeness of the most extreme matrix will tend to a limiting value instead of rising towards infinity with the number of toy matrices.

Likelihoods calculated with the response matrix can then be used in standard frequentist or Bayesian inference methods. Some examples and explanations are given in section A.1 and onwards.

2.4. Backgrounds

In general, data will contain background events that are not part of the process one wants to investigate. Within the forward-folding approach, there are three ways to deal with these background events. Simply subtracting them from the data vector is not an option, even if the background contamination is perfectly known. This would break the Poissonian assumptions in the likelihood function.

We can roughly divide the background in three categories:

[5] E.g. with a normal distribution, which is not bounded in either direction.

[6] E.g. uniform distributions.

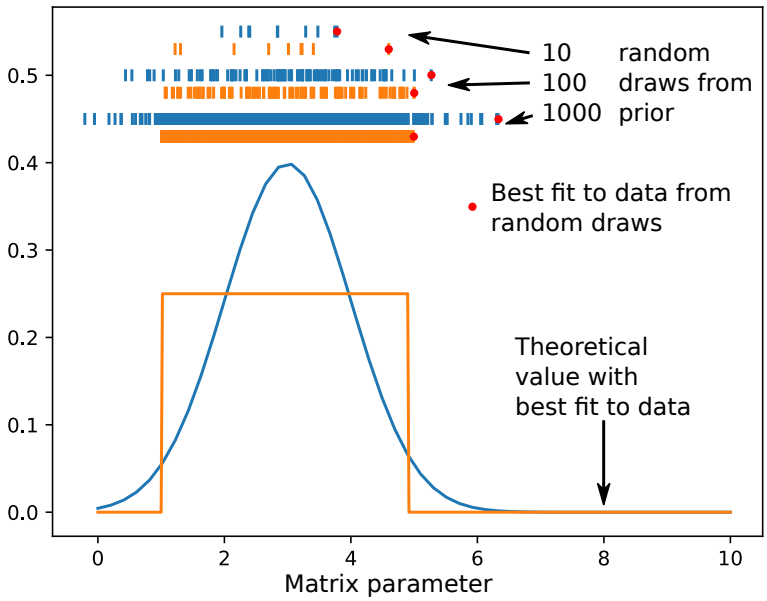

Figure 3. Profile likelihood and unbounded detector parameters. If the best possible fit value for a detector parameter lies outside the expected range, this can led to unwanted effects in combination with the use of a profile likelihood. The best fit value (red) keeps increasing with the number of random evaluations in the case of the normally distributed parameter (blue). When the parameter uncertainty is assumed to be a bound uniform distribution, the value approaches a limiting value much quicker (orange).

1. irreducible background,
2. "physics-like" background,
3. detector specific background.

Irreducible background produces exactly the same (measurable) signal in the detector as signal events. Since they are identical, as far as the detector is concerned, they occupy the same truth bins, and it is not possible to tell them apart on an event-by-event basis. The signal purity within a truth bin is always determined by the tested models, since those also determine any connections between different truth bins (background shapes, etc.). Please note the distinction between the definition of the truth bins – which is dictated by what can be measured with the detector – and the definition of the "signal of interest", which depends on the tested models.

"Physics-like" background does in principle produce detector signatures that are different from the signal. When it ends up in the final selection, it is usually due to some sort of reconstruction failure, like a misidentified or missed particle. The shape and amount of physics-like background depends on physics models that might be not well constrained at the moment, so it should be treated like the signal, with its own bins in truth space and response matrix columns (see figure 4). These truth bins are *separate* from the truth bins of the signal events. Since the physics-like backgrounds produce different kinds of events, the detector response to these events is also different from the

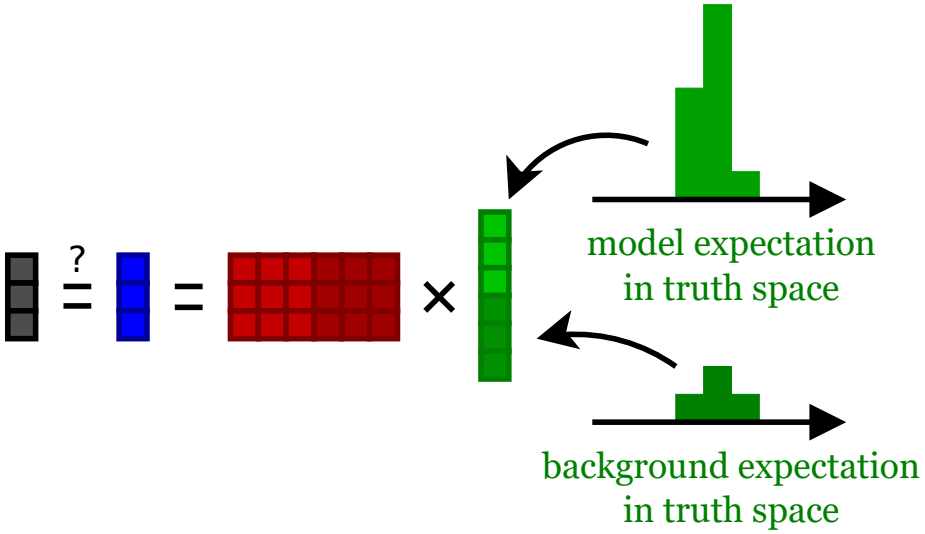

Figure 4. Forward-folded background. Background processes get their own separated binning in truth space. Future changes in the modeling of the background are possible. Data releases can include templates of the background distributions, so users of the response matrix will not have to provide their own background estimates.

response to the signal events. It can be seen as its own response matrix, but the toy variations have to be consistent with those of the signal events. This allows the data to be used consistently with evolving signal and background physics model predictions. Since users of the data and response matrix release might not care about the background model, the publication can include such models, e.g. as templates in truth space.

If the background is very detector specific and does not depend (much) on unconstrained physics models, it could be simpler to encode the reconstructed background shape in the columns of the response matrix (see figure 5). Each such column corresponds to a single bin in truth space, which would then decide the strength/amount of that background in the sample. This would make it impossible to change the background shape in the future, but it can reduce the complexity of the response matrix considerably.

For both physics-like and detector specific background it is possible to give the data the power to constrain their contributions. This can be done by including one or more control regions in the reco data vector, each being enriched in different types of background events. Reco-space model predictions in the control regions and the signal region are correlated by the response matrix, but the data points remain statistically independent Poissonian samples.

2.5. Matrix tests

All methods described in this work depend on the model-independence of the response matrix, so this needs to be ensured with dedicated tests. In general, no response matrix will be completely model-independent, but the aim is to reduce the model dependence to a level where it can be neglected. The easiest way to test the model dependence of the response matrix is to use multiple different event generators, i.e. models, to generate the matrix and then compare the matrices with one another.

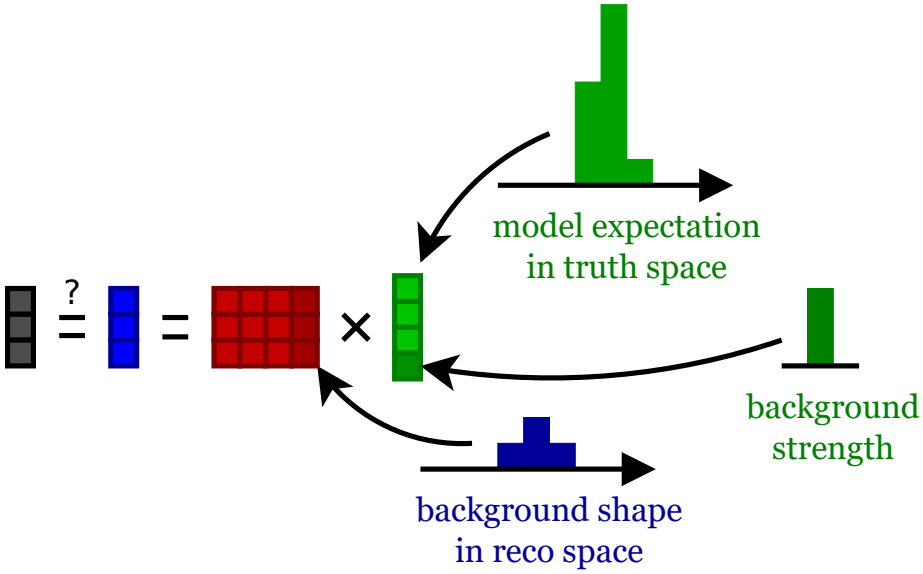

Figure 5. Template background. The reco-space shape of the background is stored as columns in the response matrix. The corresponding truth bins decide the strength/weight of the background. Future changes in the modeling of the background are *not* possible. Only the weights can be varied.

All matrices are only known to a certain precision. This uncertainty is expressed as a Bayesian posterior probability distribution of the true matrix parameters (see section 3). For the purpose of the comparison of two matrices, we can interpret the set of matrix elements R_{ij} as a multivariate random variable $\boldsymbol{R}$. The two posterior distributions of the compared matrices ($\boldsymbol{R}$ and $\boldsymbol{R}'$) define a combined distribution of matrix differences $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{R} - \boldsymbol{R}' \tag{2.12}$$

The two matrices can be considered compatible with one another if the point signifying that the matrices are identical $\boldsymbol{X}^0 = (0, 0, \ldots)$, i.e. $R_{ij} = R'_{ij}\ \forall i, j$, is a reasonable part of the distribution of $\boldsymbol{X}$.

A way to test this, is using the Mahalanobis distance of $\boldsymbol{X}^0$. The Mahalanobis distance D_M is a generalised standard score[7] for correlated, multivariate random variables [6]:

$$D_M(\boldsymbol{X}) = \sqrt{(\boldsymbol{X} - E[\boldsymbol{X}])^T \cdot \Sigma^{-1} \cdot (\boldsymbol{X} - E[\boldsymbol{X}])}, \tag{2.13}$$

where Σ^{-1} is the inverted covariance matrix of the distribution of $\boldsymbol{X}$. It can be calculated from a randomly generated set of matrices according to their uncertainties (see section 3). The higher the Mahalanobis distance, the more unlikely a point is under the assumption of the given random distribution.[8]

The "extremeness" of $\boldsymbol{X}^0$ can be measured by the probability included in the area of the distribution where $D_M(\boldsymbol{X}) < D_M(\boldsymbol{X}^0)$. Equivalently one can define the "compatibility" C of the two matrices as the probability included in the area where $D_M(\boldsymbol{X}) > D_M(\boldsymbol{X}^0)$. In the case of normally distributed $\boldsymbol{X}$, the squared Mahalanobis distance is distributed like a chi-squared distribution:

[7] I.e. the distance of a point from the centre of a distribution, measured in standard deviations.

[8] Assuming a sufficiently symmetric and unimodal distribution.

$D^2_M(\boldsymbol{X}) \sim \chi^2_k$, with the number of degrees of freedom k being the number of response matrix elements. This means the compatibility can be calculated as:

$$C = 1 - F(D^2_M(\boldsymbol{X}^0), k), \tag{2.14}$$

with the cumulative chi-squared distribution function F. Since it is not given that the differences of the two matrices are normally distributed,[9] the more general approach is to numerically integrate the probability:

$$C = \int_{D_M(\boldsymbol{X}) > D_M(\boldsymbol{X}^0)} f(\boldsymbol{X}) \mathrm{d}\boldsymbol{X}, \tag{2.15}$$

with the probability density function f.

A high compatibility C means that it is reasonable to assume that the two matrices describe the same true matrix *within their statistical uncertainties*. Deciding on a critical value for C, below which the two matrices are considered to be too different, is not straight forward. At 0.5, the matrices are ensured to be identical more likely than not. This is rather conservative though, as it corresponds to a difference between the two matrices of "less than 1σ" in the one-dimensional case. If one is to decide whether to combine two independent measurements of the same variable into one weighted average, it seems like one would (in general) still combine measurements that differ up to 2σ (or even 3σ). This corresponds to a critical C of 0.05 (0.003). Since the model-independence is so important for this analysis method though, and a modification of the truth binning is a relatively cheap operation, it might be justified to be conservative here.

It is important to note that passing this check is a necessary condition for model-independence, but not a sufficient one. The available models might not be different enough to reveal any hidden model dependencies given the available MC statistics. Ideally, one should also test the generation of the response matrix with completely random input vectors for the detector simulation. That is, instead of using a physics model to generate the events, one uses particle guns with varying underlying event distributions. If such data is not available with sufficient statistics, the next best thing to step outside the bounds of the available models is to re-weight those models with arbitrary weight functions. Of course, in that case the usual limitation of re-weighting applies, i.e. that only parts of the phase-space that have been simulated to begin with can be re-weighted.

No matter what is done, it remains impossible to prove conclusively that the response matrix is truly model-independent. In the end, the analyst will have to decide at which point enough has been done to show that the possible remaining model dependence is small enough for the given data.

3. Implementation

3.1. Building the detector response matrices

The detector response matrix is built from Monte Carlo simulations. Events are first categorised by their truth information and assigned a truth bin number j. Then events that end up being selected and thus have reconstructed properties get assigned a reco bin number i.

The probability for an event in truth bin j ending up in reco bin i is

$$P(j \to i) = \lim_{N_j \to \infty} \frac{N_{ij}}{N_j}, \tag{3.1}$$

[9] Especially for bins with only very few entries.

as defined in section 2. Here N_j is the number of events in truth bin j, including the events that do not get assigned a reco bin, and N_{ij} the number of events in truth bin j and reco bin i. Since the number of Monte Carlo events is limited by the available computing resources, this value can only be approximated:

$$R_{ij} = \frac{N_{ij}}{N_j}. \tag{3.2}$$

The simulated detector does not reproduce the behaviour of the real one perfectly. We parametrise the estimated difference as a set of systematic uncertainties that get propagated as weights and variations in the selection (see section 2.2). This event weighting and variation is done as a replacement for full simulations of varied detectors to save computing time. The systematic parameters are sampled from their assumed distributions (e.g. normal or uniform) and the events are weighted and varied accordingly. We call each sampling of the parameter space a toy simulation. Each toy simulation t yields its own response matrix

$$R^t_{ij} = \frac{W^t_{ij}}{W_j} = \frac{N^t_{ij} w^t_{ij}}{N_j w_j}, \tag{3.3}$$

where W_j and W^t_{ij} are the sum of weights, and w_j and w^t_{ij} the average weights of all events in the respective bins. Since the detector variations do not affect the events on the generator level, the sum of weights in the truth bin W_j is not affected by the toys.

Another important uncertainty in the matrix comes from the statistical uncertainty due to the finite number of simulated MC events. This means the values of R^t_{ij} will also suffer from statistical variations from the true MC value. These fluctuations are not represented in the systematic toys, as those do not vary the generated events. To effectively incorporate this effect in a coherent way, we decompose the effects of efficiency and smearing in the matrix and build a model that we can draw toy matrices from. We estimate the statistical uncertainties in a "Bayesian inspired" three-step process.

The first two uncertainties stem from the multinomial sampling of N^t_{ij}. For the purpose of statistical error estimation, we split the multinomial process in two parts:

- A binomial chance of being reconstructed at all (i.e. efficiency) ϵ^t_j
- A multinomial probability of ending up in a certain reco bin (i.e. smearing) p^t_{ij}

$$\epsilon^t_j = \lim_{N_j \to \infty} \frac{\sum_i N^t_{ij}}{N_j}, \tag{3.4}$$

$$p^t_{ij} = \lim_{N_j \to \infty} \frac{N^t_{ij}}{\epsilon^t_j N_j}, \tag{3.5}$$

$$\epsilon^t_j \cdot p^t_{ij} = \lim_{N_j \to \infty} \frac{N^t_{ij}}{N_j}. \tag{3.6}$$

We do not know the true values of these parameters, so we can treat them as Bayesian random variables. Treating the efficiency separately from the smearing makes it easier to find fitting prior parameters for the distributions (see below).

If we assume a beta distribution[10] as a prior for the distribution of ϵ^t_j, we can use the simulated

[10] The beta distribution is the conjugate prior for binomial distributed likelihoods. See [7].

number of events directly to update the parameters of the prior, β'_{*j} and $\beta'_{\dagger j}$, to get the parameters of the posterior:[11]

$$\epsilon_j^t \sim \text{Beta}(\beta_{*j}^t, \beta_{\dagger j}^t), \tag{3.7}$$

$$\begin{aligned}\beta_{*j}^t &= \beta'_{*j} + \sum_i N_{ij}^t \\ &= \beta'_{*j} + N_{*j}^t,\end{aligned} \tag{3.8}$$

$$\begin{aligned}\beta_{\dagger j}^t &= \beta'_{\dagger j} + (N_j - \sum_i N_{ij}^t) \\ &= \beta'_{\dagger j} + N_{\dagger j}^t.\end{aligned} \tag{3.9}$$

Here N_{*j}^t is the number of selected and $N_{\dagger j}^t$ the number of "lost", i.e. not selected, events in truth bin j.

We can do the same for the smearing uncertainty if we assume a Dirichlet distribution[12] as a prior for the distribution of p_{ij}^t. Again we can use the simulated number of events directly to update the prior's parameters α'_{ij}:

$$\boldsymbol{p}_j^t \sim \text{Dir}(\boldsymbol{\alpha}_j^t), \tag{3.10}$$

$$\alpha_{ij}^t = \alpha'_{ij} + N_{ij}^t. \tag{3.11}$$

The variances of the resulting posterior distributions are

$$\sigma^2(\epsilon_j) = \frac{\beta_{*j}^t \beta_{\dagger j}^t}{(\beta_{*j}^t + \beta_{\dagger j}^t)^2(\beta_{*j}^t + \beta_{\dagger j}^t + 1)}, \tag{3.12}$$

$$\sigma^2(p_{ij}^t) = \frac{\alpha_{ij}^t(\sum_{i' \neq i} \alpha_{i'j}^t)}{(\sum_{i'} \alpha_{i'j}^t)^2((\sum_{i'} \alpha_{i'j}^t) + 1)}, \tag{3.13}$$

and the expectation values

$$\begin{aligned}\hat{\epsilon}_j^t &= \frac{\beta_{*j}^t}{\beta_{*j}^t + \beta_{\dagger j}^t} \\ &= \frac{\beta_{*j}^t}{N_j + \beta'_{*j} + \beta'_{\dagger j}},\end{aligned} \tag{3.14}$$

$$\hat{p}_{ij}^t = \frac{\alpha_{ij}^t}{\sum_{i'} \alpha_{i'j}^t}. \tag{3.15}$$

As prior parameters we set

$$\beta'_{*j} = \beta'_{\dagger j} = 1 \tag{3.16}$$

and

$$\alpha'_{ij} = \min(1, 3^{N_{\text{reco variables}}}/N_{\text{reco bins}}). \tag{3.17}$$

[11] Usually the parameters of the beta function are denoted as α and β. To avoid confusion with the parameters of the Dirichlet distribution α_i, we decided to use β_* and $\beta_\dagger$ respectively instead.

[12] The Dirichlet distribution is the conjugate prior for multinomial distributed likelihoods. See [7].

The choice of prior parameters in eq. 3.16 ensures that the overall reconstruction (in)efficiency of the truth bins is uniformly distributed apriori. The prior parameters in eq. 3.17 assume that the reconstruction probabilities are concentrated on a few reco bins (~ 3 per reco variable), while being completely agnostic about *which* reco bins those are.[13] Please note that a flat Dirichlet prior ($\alpha'_{ij} = 1$) is generally not suitable for the description of event smearing with many reco bins. It biases the matrix towards very strong smearing (all truth bins are smeared to all reco bins equivalently in the prior).

The resulting variances of the posterior distributions are consistent with the standard frequentist approach in the limit of high statistics. Especially the binomial case corresponds to an experiment where we added a "pseudo-observation" of two simulated events, of which one was successfully reconstructed, to the actual data.

The third step is to evaluate the statistical uncertainty of the weight correction. The true weight correction

$$m^t_{ij} = \lim_{N_j \to \infty} \frac{w^t_{ij}}{w_j} \tag{3.18}$$

is estimated from the sum of of weights:

$$\hat{m}^t_{ij} = \frac{w^t_{ij}}{w_j} = \frac{W^t_{ij}/N^t_{ij}}{W_j/N_j}. \tag{3.19}$$

For the purpose of the variance estimation, we treat w^t_{ij} *independently* from ϵ^t and p^t_{ij} as arithmetic means of samples with given sizes.

We apply the usual standard error of the mean formula for each average weight. The sample variance is estimated from the sum of squared weights. To be able to estimate variances even for bins with only one entry, we add a pseudo-observation event with an expected weight of 1:

$$\sigma^2(w^t_{ij}) = \left(\left(\frac{V^t_{ij} + 1^2}{N^t_{ij} + 1} \right) - \left(\frac{W^t_{ij} + 1}{N^t_{ij} + 1} \right)^2 \right) \frac{1}{N^t_{ij} + 1}, \tag{3.20}$$

where V^t_{ij} are the sums of the squared weights in the respective bins. The pseudo-observation represents our prior knowledge of the weights and has no effect in the limit of high statistics. The variance of the weight correction is then

$$\sigma^2(m^t_{ij}) = \frac{\sigma^2(w^t_{ij})}{(w_j)^2} + \left(\frac{w^t_{ij}}{(w_j)^2} \right)^2 \sigma^2(w_j). \tag{3.21}$$

Here $\sigma^2(w_j)$ is the statistical uncertainty on the average weight in truth bin j, defined analogously to $\sigma^2(w^t_{ij})$.

All that is left now, is to combine the variances of the multinomial sampling and the weight correction:

$$\sigma^2_{\text{MC stat}}(R^t_{ij}) = (\hat{\epsilon}^t_j \hat{m}^t_{ij})^2 \sigma^2(p^t_{ij}) + (\hat{\epsilon}^t_j \hat{p}^t_{ij})^2 \sigma^2(m^t_{ij}) + (\hat{p}^t_{ij} \hat{m}^t_{ij})^2 \sigma^2(\epsilon^t_j) \tag{3.22}$$

[13] Dirichlet distributions with $\alpha < 1$ favour "extreme" sets of p, where most of the probability is concentrated in few categories, over flat sets, where the probability is more uniformly distributed. The corresponding reco bins do *not* have to be contiguous.

If the statistical variance is much smaller than the systematic detector variation,

$$\sigma^2_{\text{MC stat}}(R^t_{ij}) \ll \sigma^2_{\text{syst}}(R_{ij}) \approx \frac{1}{N_{\text{toy}} - 1} \sum_t (R^t_{ij} - \bar{R}_{ij})^2, \tag{3.23}$$

for all toy experiments, we can neglect it. In practice there will almost certainly be bins where this is not the case, e.g. (almost) empty matrix elements.

To deal with these non-negligible statistical uncertainties, we generate random toy matrices from every systematic toy matrix according to the three step process described above: First we draw a set of efficiencies and multinomial probabilities from the posterior beta/Dirichlet distributions, and then we modify these with weight factors calculated from normal distributed mean weights.

$$\epsilon^{t*}_j \sim \text{Beta}(\beta^t_{*j}, \beta^t_{\dagger j}), \tag{3.24}$$

$$\boldsymbol{p}^{t*}_j \sim \text{Dir}(\boldsymbol{\alpha}^t_j), \tag{3.25}$$

$$w^{t*}_{ij} \sim \text{Norm}(w^t_{ij}, \sigma^2(w^t_{ij})), \tag{3.26}$$

$$R^{t*}_{ij} = \frac{w^{t*}_{ij}}{w^{t*}_j} \epsilon^{t*}_j p^{t*}_{ij}. \tag{3.27}$$

These toy matrices are then handled just like the systematic toy matrices to calculate marginal or profile likelihoods.

To further limit the influence of the statistical uncertainties, we check whether the truth bin expectation values exceed the number of simulated events:

$$\mu_j \overset{!}{<} N_j. \tag{3.28}$$

Hypotheses that predict more events in a given truth bin than were simulated are outside the testable scope of the response matrix. If the tested hypotheses (e.g. in a likelihood fit or Bayesian posterior sampling) are close to this limit, it could lead to model dependence of the results. Therefore it is necessary to check whether this is the case.

3.2. Binning

3.2.1. General considerations

The properties of the detector response matrix depend first and foremost on the chosen binning in truth and reco space. The binning has to balance the following (contradictory) aims:

- Ensure the independence of the interaction model → large number of truth bins.
- Maximise the separation power, i.e. how well well different models can be told apart on the reco level → large number of reco bins.
- Minimise the influence of statistical errors → large number of events per bin.

The following sections describe the general methodology of choosing the binning.

3.2.2. Choosing the variables to bin in

The response matrix can only be model-independent if it is binned in the right variables. Variables close to the actual observables are more suited than those that describe the event in a more fundamental way, which have to be inferred from the measurement. For example, the lepton momentum of a charged-current neutrino interaction is a good variable, as the detector can directly measure it. The neutrino energy on the other hand is a bad choice, because the translation of neutrino energy to observables in the detector depends on the physics model (FSI, etc).

But even when binning in direct observables only, one has to take care not to introduce hidden model dependencies. The distribution of events in variables that we do not bin in, might still have an effect on the average detector response. If different models predict different angular distributions, which in turn change the detector efficiency, binning only in the muon momentum will *not* be model independent. One would have to bin in *all* truth variables that affect the detector performance to be truly model independent. In practice this is not possible, as the available computing power and thus the number of Monte Carlo events is limited. In any case, it is not necessary to expend lots of time and energy to push the model-dependence to infinitesimally low values if the measurement is already limited by the amount of real data statistics. Compromises have to be made.

Aside from detector performance considerations, one of course also has to bin in the variables of interest. The reco binning is dictated by the physics goals of the measurement. Again it is important to choose variables as close to the actual observables as possible. If a variable of interest is the function of other more basic observables, a binning in those observables would be less susceptible to hidden model dependencies.[14] Unfortunately the number of events per bin decreases exponentially with the number of binning dimensions.

3.2.3. Bin widths

As seen in section 3.1, the efficiencies of the truth bins are estimated – and toy matrices generated – using a Bayesian approach that adds two pseudo-measurements as prior information. In order to not be biased too much towards that prior, we would like the actual number of events per truth bin to be much larger than the number of pseudo observation. If the models used for the matrix building are similar to the models that will be tested with the matrix, the average number of events per truth bin is a good measure for this. If the building models and tested models vary widely, it might be better to use another number as figure of merit. The median number of events per bin could be used, or even a lower percentile. The exact details of this are not that important, as the number is merely a guideline to use when optimising the binning. The properties of the matrix will be tested independently of this anyway. For now, let us demand that $\text{mean}(N_j) \overset{!}{>} 50$.

To maximise the number of events per bin, one could choose a very wide binning. This can lead to model dependences though, if the detector performance varies considerably within a truth bin. Since model independence is a primary goal of this analysis, this defines an upper limit for the truth bin sizes.

We estimate the response variation within one bin ΔR_{ij} from the variation between neighbouring bins:

$$\Delta R_{ij} = \max_{j'} |R_{ij} - R_{ij'}|, \tag{3.29}$$

[14] A perfect truth binning would of course prevent any model dependencies.

with the neighbouring bins j'. Ideally, one would like this variation to be not much higher[15] than the uncertainties on the matrix elements:

$$\Delta' R_{ij} = \max_{j'} \left| \frac{R_{ij} - R_{ij'}}{\sqrt{\sigma^2(R_{ij}) + \sigma^2(R_{ij'})}} \right| \overset{!}{<} l, \tag{3.30}$$

with the normalised in-bin variation $\Delta' R_{ij}$ and a limit $l \sim O(5)$. Unfortunately this aim is contradictory to the need to fill each truth bin with sufficiently many events to reduce the influence of the priors (see above). Also, small scale variations might be hidden within the bins, so care has to be taken on a variable by variable base to optimise the binning.

If the detector response is sufficiently flat, the truth bin widths should be adjusted to include the necessary MC statistics. Other than that, they should be made as small as possible. The reco bin width is mostly dictated by the physics goals of the analysis, data and MC statistics, and the resolution of the detector. If the truth binning is chosen in a way to ensure model independence, no reco binning will introduce additional model dependence. A fine reco binning might expose model dependencies, but the cause is solely in the truth binning. Conversely, a coarse binning can hide dependencies, so one should aim for as fine a binning as MC and data statistics permit. This also ensures the best hypothesis testing power. Reco bins should not be finer than their truth counterparts.

One possible algorithm to decide on the final bin widths is as follows:

1. Set reco binning according to expected statistics and physics goals.
2. Set truth binning very fine.
3. Merge truth bins until $\text{mean}(N_j) \overset{!}{>} 50$
   a) Set limit for in-bin variation l.
   b) Merge neighbouring bins with lowest number of entries until limit is reached.
   c) Merge neighbouring bins with lowest in-bin variation $\Delta' R_{ij}$ until limit is reached.
   d) If necessary, increase l and repeat.
4. Fine-tune binning by hand.

After this, the resulting matrix must be checked for sufficient model-independence (see section 2.5). If it fails, the binning has to be adjusted.

3.2.4. Empty bins

The Monte Carlo samples used to generate the response matrix use a physics model $\boldsymbol{\mu}'$. In that model, certain areas of the truth phase space are very unlikely to be realised and the corresponding truth bins will not be filled with a sufficient number of events during the response matrix construction. This means that we have not enough information about how the detector would react to these kinds

[15] Ideally one would like the variation within the bins to be lower than the statistical uncertainty, but if there is no actual in-bin variation, the statistical uncertainty will dominate this estimate.

of events. Ideally one would like to build the response matrix with simulation data that covers all possible phase space, but this is computationally difficult.

Since we cannot predict how those events behave in the detector, we remove those bins from the vector of truth expectation values $\boldsymbol{\mu}$. This is equivalent to setting those expectation values to 0 in all considered hypotheses, and reduces the dimensionality of $\boldsymbol{\mu}$. It means that we *cannot* test hypotheses that predict any events in these bins.

There might also be reconstruction bins that never get filled during the construction of the response matrix. The expectation value in those bins will be close[16] to 0 for all possible hypotheses. Finding events in these bins would necessitate further investigation and possibly the generation of more Monte Carlo data.

To judge how well the simulated data covers the real measurement and tested hypotheses, we can compare the number of simulated events to the number of measured/predicted events:

$$\xi_{\text{reco},i}(\boldsymbol{n}) = \max_t \frac{n_i}{N_i^t} \tag{3.31}$$

$$\xi_{\text{truth},j}(\boldsymbol{\mu}) = \frac{\mu_j}{N_j}. \tag{3.32}$$

Numbers close to or above one indicate that the simulated phase space is not sufficient and should be extended. More specifically, $\boldsymbol{\xi}_{\text{truth}}(\boldsymbol{\mu})$ indicates how well the given hypothesis $\boldsymbol{\mu}$ is covered by the simulation, while $\boldsymbol{\xi}_{\text{reco}}(\boldsymbol{n})$ shows whether the actual measurement is covered at all.

3.3. Software and data format

A lot of particle physics experiments rely on the ROOT analysis framework developed at CERN [8] as their main data format for storing event data. These data containers can be very specific to the experiment and often require specialised software tools to read them. Those tools are often only available within a collaboration and are not intended for external users.

The ultimate goal of the response-matrix-centred approach is to enable people who are not intimately familiar with the experiment to compare event generators with the measured data. To this end, it was decided to develop the software that deals with the response matrix independently from collaboration-internal frameworks, and that can be used for both building response matrices as well as using them to test models against published data. The result of this effort is the Response Matrix Utilities framework `ReMU` [9].

`ReMU` is written in pure Python and thus able to run on any system that supports the scripting language. Numerical calculations are handled by the NumPy [10], SciPy [11], and PyMC [12] packages to take advantage of the performance gains of compiled code. `ReMU`'s source code is publicly available on the code-sharing platform GitHub [13], and releases of the software are distributed via the Python Package Index (PyPI) [14]. This means, installing the framework on systems supporting PyPI can be done with a single command:

```
pip install remu
```

Data is stored and exchanged with standard file formats. The binning of the response matrix is saved in YAML files [15], a text format that is both human readable and easy to parse by machines.

[16] It will not be exactly 0 due to the generation of statistically varied matrices as described in section 3.1.

The response matrix is saved as a binary NumPy file. To save disk and RAM space, the matrix is saved as a "sparse" matrix, i.e. only the rows of the matrix corresponding to truth bins that were filled during the matrix creation are saved. The information which bins were filled (and how many events were simulated in each) is saved in another binary NumPy file. The data itself (reco or truth space) can either be provided as binned histograms with binary NumPy files, or event-by-event with Comma-Separated Values (CSV) files. Other file formats are supported via the python data analysis library "pandas" [16]. ROOT files can be read in directly using the uproot library [17]. It does *not* require ROOT to be installed to be usable.

For long term data storage, it is also planned to implement an "archival" file format that stores all information as text files. Text files have the highest chance of remaining readable in the future, as they are generally considered to be the lowest common denominator in data exchange. Even if `ReMU` (or even Python in general) should stop working at some point, the data and response matrix could be re-used by different programs in this form relatively easily.

A publication following the response-matrix centred approach would include at least these elements:

- Response matrix binning in reco space ("reco-binning.yml")
- Response matrix binning in truth space ("truth-binning.yml")
- The systematically and statistically varied sparse response matrices ("response.npy")
- A truth space histogram of how many events were simulated in each bin ("generator-truth.npy")
- Reco histogram of data ("data.npy") or CSV file of reco properties of all data events ("data.csv")
- Optionally, truth space background templates (background.npy)

Users of the publication could then provide their own signal predictions to calculate likelihoods. `ReMU` provides many functions to make this as easy as possible. This includes the definition of composite hypotheses and the likelihood maximisation over their parameter spaces.

4. Example analysis

4.1. Introduction

This section is intended as a rough outline on how `ReMU` and the response-matrix-centred approach could be used in practice. Since the actual software is subject to active development, we will concentrate on the principles rather than the actual implementation here. The example is taken from the documentation of `ReMU`, and the reader should refer to it for the full implementation details and additional information [18].

The mock experiment that is handled here is quite simple. It records events with only two properties: x and y. Only x is smeared by the detector (Gaussian blur with $\sigma = 1$). The efficiency of detecting an event depends only on y ($\epsilon = 0.9\frac{1}{2}(1+\mathrm{erf}(y/\sqrt{2}))$). There are no background events. Let us assume we are interested in a measurement of the distribution of x.

4.2. Preparation of the response matrix

The response matrix must be prepared by the detector experts within an experiment's collaboration. Only they have the necessary knowledge about the detector response and its uncertainties, as well as access to the full detector simulation framework. Care must be taken to ensure that the response matrix is actually as model-independent as desired. `ReMU` offers a few methods to test the matrices for that property.

Response matrix objects are created by specifying the binning in reco and truth space. They are then filled with simulated events that were processed with the full detector simulation and analysis chain:

```
respA = migration.ResponseMatrix(reco_binning, truth_binning)
respA.fill_from_csv_file("modelA_data.txt")
```

A model-independent response matrix should not depend on the model that was used to populate the response matrix. `ReMU` offers a method to calculate the Mahalanobis distance between two matrices and compare the result with the expected distribution assuming that the two matrices are random variations of the same matrix (see section 2.5):

```
respB = migration.ResponseMatrix(reco_binning, truth_binning)
respB.fill_from_csv_file(["modelB_data.txt"])
respA.plot_compatibility("compatibility.png", respB)
```

See figure 6 for how these plots might look for compatible and incompatible matrices. Note that passing this test is a necessary condition for model-independent matrices, but not a sufficient one. The available models for this test might be too similar to show any intrinsic model-dependencies of the response matrix. It is up to the detector experts to make sure that they are covering the necessary response variations in the truth binning.

In the case of this example, it is necessary to bin the truth both in x (because this is the variable of interest) and in y (because the detection efficiency depends on this variable). Note that if the response matrix is model-independent, it can actually be populated by all available simulated data combined:

```
resp.fill_from_csv_file(["modelA_data.txt", "modelB_data.txt"])
```

Figure 7 shows a plot of the 2D projections of the final response matrix. It was generated by one of the many methods in `ReMU` that are intended to help gaining insights into the properties of the response matrices when building them:

```
resp.plot_values("optimised_response_matrix.png", variables=(None, None))
```

4.3. Using the response matrix

Once the response matrix (or the set of response matrices, see section 2.3) is prepared and published with the data vector, it can be used to do statistical tests. This can be done both inside the experiment's

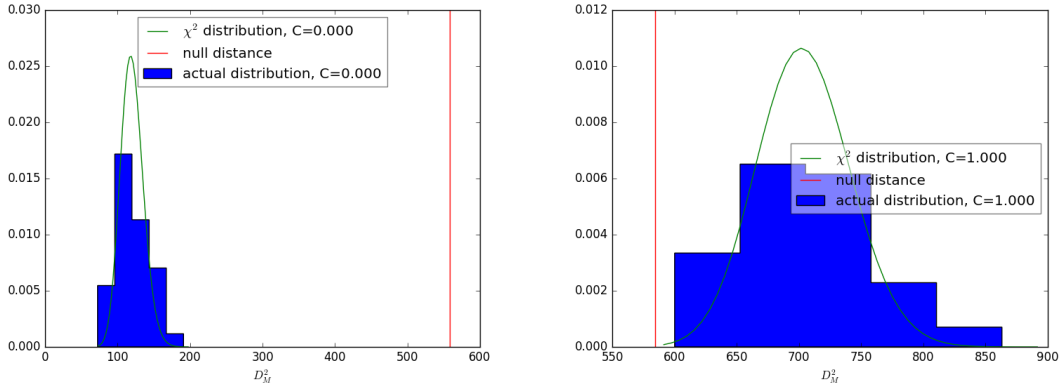

Figure 6. Matrix compatibility plots. The squared Mahalanobis distance (see section 2.5) of two matrices populated with simulated events from different models. When the truth binning cannot cover the varying detector response between the models (left), the distance (vertical line) will be larger than expected from purely random variation (blue histogram). In the Gaussian limit, this variation should be chi-squared distributed (green curve). If the binning covers the model differences (right), the distance should fall within the expected distribution, or be smaller. The distance can be smaller than the expected distribution, because the parametrisation of the uncertainties of the matrix elements starts with a prior (or pseudo observations) that are common to both compared matrices (see section 3.1). So two matrices with no data in them will be perfectly identical, despite large expected statistical uncertainties.

collaboration, as well as outside of it. The usage of the response matrix does not require expert knowledge of the detector.

Within `ReMU` the data and response matrices are combined into `LikelihoodMachine` objects. These then provide methods to do different statistical tests on model predictions. It does not matter what exactly the data looks like or how many matrices make up the set, the interface to the user stays the same.

The simplest kind of test that can be done is comparing the likelihoods of different models. For this, all that is needed is a model prediction for the events in truth space:

```
truth_binning.fill_from_csv_file("modelA_truth.txt")
modelA = truth_binning.get_values_as_ndarray()
```

The `LikelihoodMachine` can then calculate (log-)likelihoods of the measured data, given this prediction and marginalising over the detector uncertainties encoded in the set of response matrices:

```
lm.log_likelihood(modelA)
```

In case models have free parameters, it is also possible to maximise the likelihood over the allowed parameter space. For example, one can use the (area normalised) shape of models as templates and let the template weight (i.e. the number of true events) be fitted to the data:

```
modelA_shape = TemplateHypothesis([modelA / np.sum(modelA)])
lm.max_log_likelihood(modelA_shape)
```

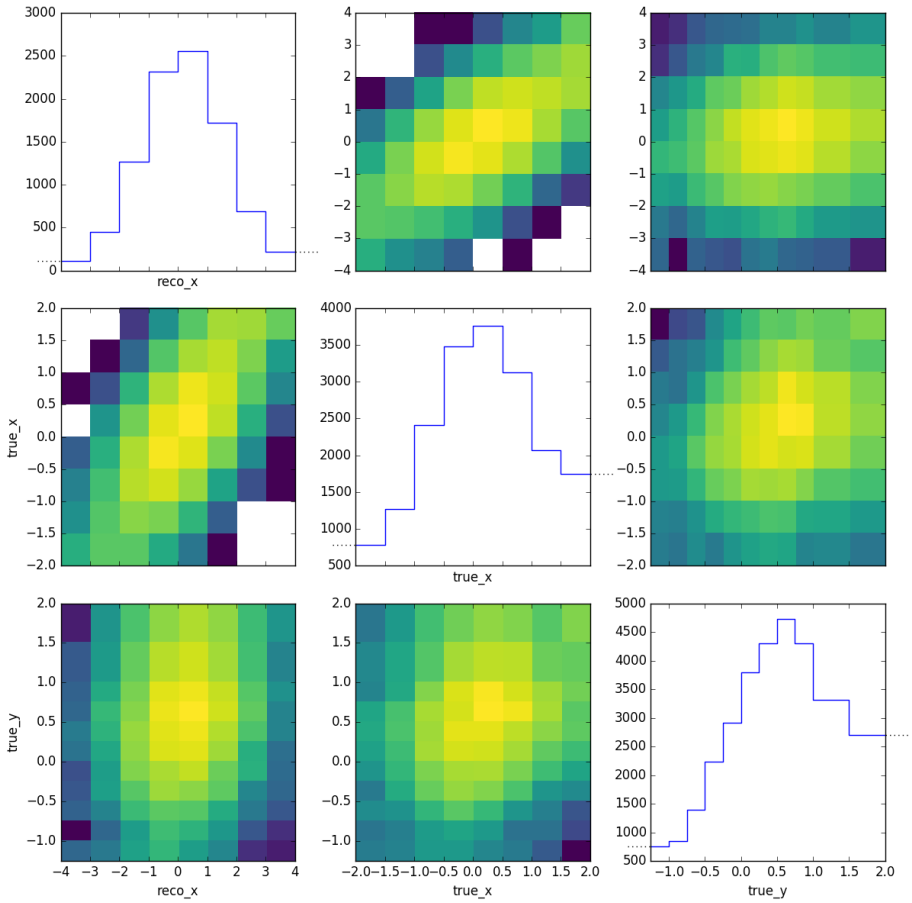

Figure 7. Response matrix projection. 1D and 2D projections of the distribution of events that populate the response matrix. Only events that have been reconstructed are included, i.e. this shows the smearing/migration of events, but not the efficiency. Dotted lines outside the plot axes indicate that the corresponding bin is not constrained in that direction, i.e. it behaves like on over- or underflow bin.

This will return both the maximised (log-)likelihood and the parameter values of that point.

Since likelihood values alone are hard to interpret, `ReMU` also offers several methods to calculate p-values:

```
lm.likelihood_p_value(modelA)
lm.max_likelihood_p_value(modelA_shape)
lm.max_likelihood_ratio_p_value(model0, model1)
```

These respectively calculate the probability of

- measuring data that yields a lower likelihood than the actual one, assuming the provided model is true,
- measuring data that yields a lower maximum likelihood than the actual one, assuming the best fit point of the provided model is true,

Table 1. Example models.

	$\mathrm{E}(x)$	$\mathrm{E}(y)$	$\mathrm{var}(x)$	$\mathrm{var}(y)$	$\mathrm{cov}(x, y)$
Model A	0.1	0.2	1.0	1.0	0.0
Model B	0.0	0.0	1.0	1.0	0.5

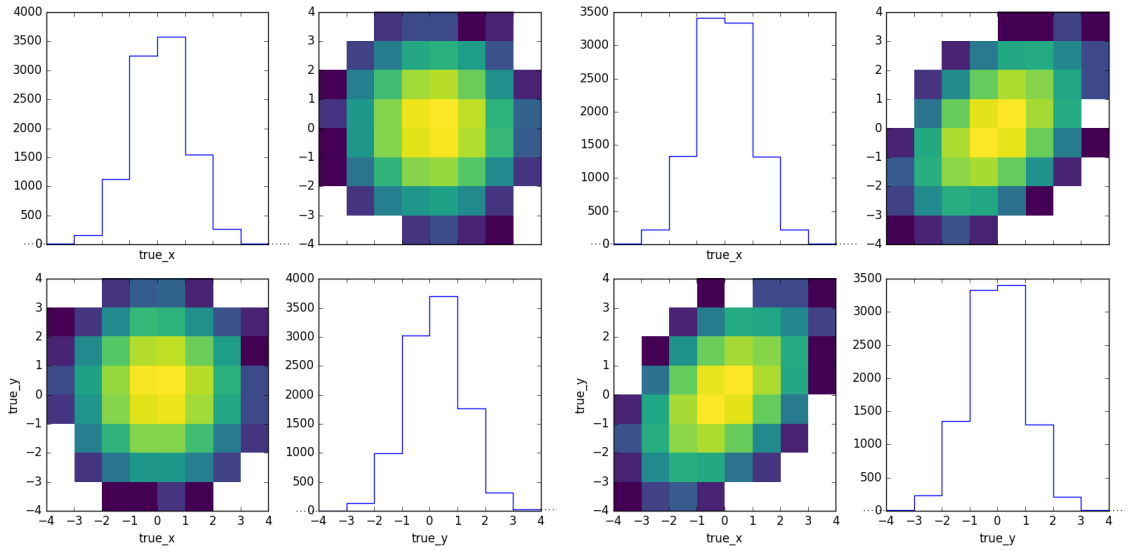

Figure 8. Example models. True distribution of events in model A (left) and model B (right). Dotted lines outside the plot axes indicate that the corresponding bin is not constrained in that direction, i.e. it behaves like on over- or underflow bin.

- measuring data that yields a lower ratio of maximised likelihoods between the two specified models than the actual one, assuming the best fit point of `model0` is true.

These p-values can then be used to check goodness of fit, to construct different confidence intervals, or to do frequentist hypothesis tests.

Let us assume we have two models we want to compare to the data. Model A assumes that the true properties x and y of events are uncorrelated, normal distributed. Model B assumes a correlation between x and y (see figure 8). They also feature slightly different means of the distribution. See table 1 for a summary of the model parameters. Each model only predicts the shape of the event distribution, but not the total number of events. Note that even though we are only interested in a measurement of x, the different behaviours in y lead to different average detection efficiencies between the models.

Maximising the likelihood over the free normalisation parameter of the models yields two maximum likelihood solutions that both fit the data reasonably well (see figure 9). A look at their respective `max_likelihood_p_values` tells us that model A is slightly disfavoured (p-value ~ 0.1).

Instead of globally excluding a hypothesis, it can be useful to look at the local p-values in the parameter space. Figure 10 shows this for the two models. Again model A is disfavoured. This is not useful to construct confidence intervals under the assumption that each model is true, though.

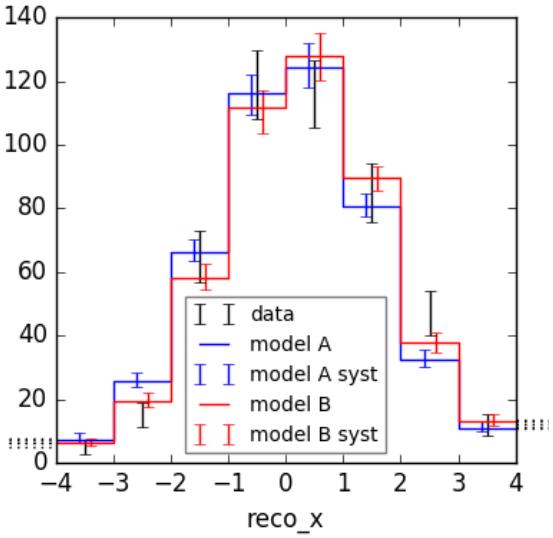

Figure 9. Reco-space model comparison. The model predictions are shown with mean and standard deviation due to detector systematics. The data is shown with $\sqrt{N}$ "error bars" for visualisation purposes only.

This is done in figure 11 using the `max_likelihood_ratio_p_value` of the fixed normalisation prediction vs the floating normalisation models. By construction, this p-value is 1 at the maximum likelihood fit point of the parameter space. Model A and B yield different confidence intervals for the total number of events, because their average detection efficiencies are different.

This kind of model-dependent result is a good illustration for the advantage of the response-matrix-centred forward-folding approach of sharing data. Had the data of this mock experiment been shared as an unfolded distribution, it would have had to include the different average efficiencies of possible models in the systematic uncertainties of the result. This would lead to inflated errors compared to the specific model tests done here. But even those inflated errors could not guarantee that the coverage of the true value is as expected if the true model is not among the considered ones. Alternatively the result could have been unfolded in both x and y, but that approach is not always feasible. Depending on the available data statistics, the number of variables that influence the detector response, and how well the detector can measure those in the first place, the data might not be able to constrain all relevant truth bins. This is not an issue in the forward-folding approach.

5. Conclusions

The response-matrix-centred approach to presenting cross-section measurements promises to be a useful addition to the set of tools available to cross-section analysts. It combines the fine-grained model-independence of a multi-dimensionally unfolded differential cross-section measurement with the ability to work with low-statistics, coarsely binned real data. Even for measurements with enough real data for very fine reco-binning, the comparison in reco-space can offer superior model-separation power over comparisons in (unfolded) truth-space [5].

A method was presented how to handle systematic detector uncertainties by providing a set of possible response matrices and calculating the marginal likelihood of their reco-space predictions.

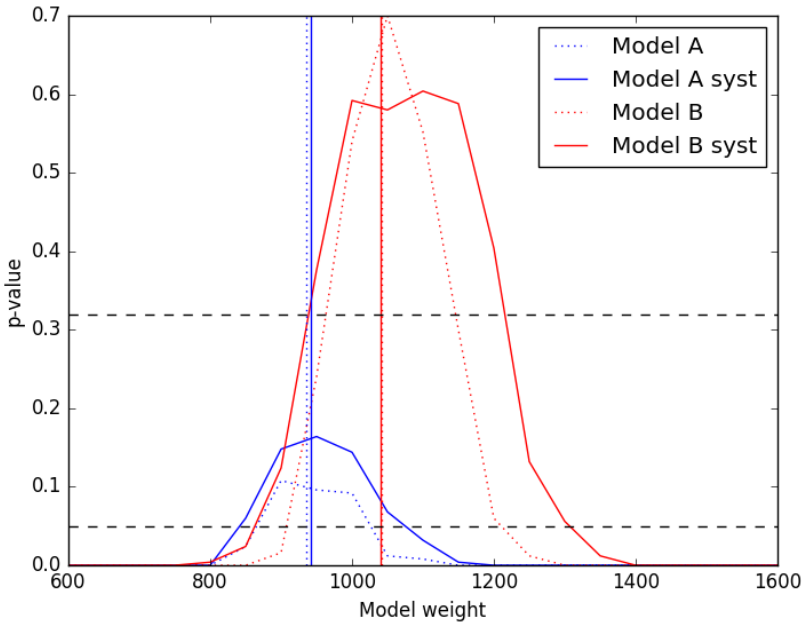

Figure 10. Local p-values, as a function of the template weight (number of true events). Vertical lines show the maximum likelihood solutions. The dotted lines show the results when not applying any detector systematics, i.e. using only a single (nominal) response matrix.

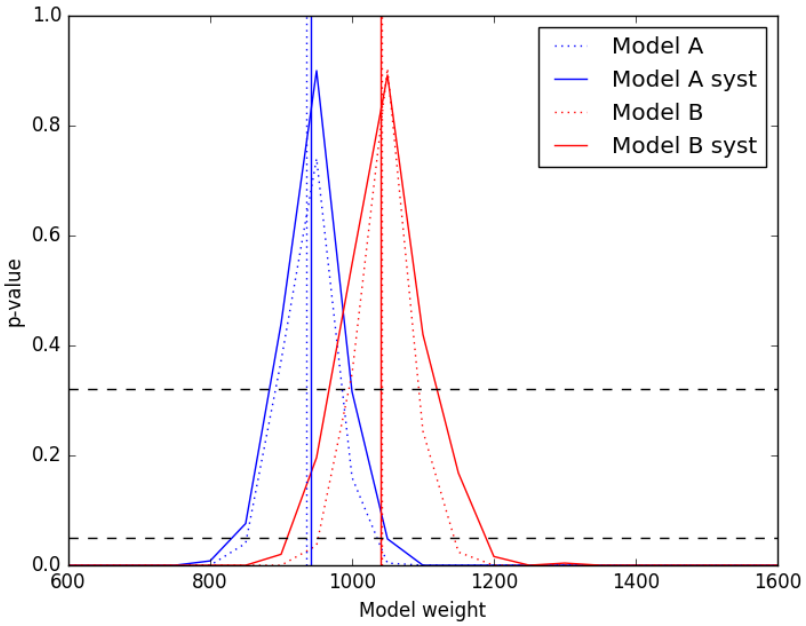

Figure 11. Likelihood ratio p-values, as a function of the template weight (number of true events). Vertical lines show the maximum likelihood solutions. The dotted lines show the results when not applying any detector systematics, i.e. using only a single (nominal) response matrix.

Statistical uncertainties of the matrix elements stemming from finite Monte Carlo statistics are handled in a similar way, by quantifying the uncertainties and creating random variations of the response matrices accordingly.

Three methods to deal with backgrounds were presented, avoiding subtracting any events from the original data vector. Irreducible background must be added to the signal truth bins by the (background) model. "Physics-like" background can be handled just like signal events, with its own set of truth bins and corresponding response parametrisation in the response matrix. Detector-specific background templates can be put directly into the response matrix, with a single truth bin determining their strength.

The biggest challenge for the analyst creating the matrix is to choose an appropriate truth binning. The detector response typically depends on a multitude of variables, but the number of bins grows exponentially with the number of binning dimensions. This leads to a very high demand for Monte Carlo statistics to build the response matrices.

Once the matrix is available though, it is relatively easy to test various physics models against the data, without having to re-evaluate the detector response and its uncertainties for each model. This will be useful for the NUISANCE[3] and Rivet[4] frameworks, for example.

Acknowledgments

I want to thank my colleagues of the T2K collaboration for supporting me during the genesis of this paper, including – but not limited to – Morgan Wascko, Kendall Mahn and Stephen Dolan. Especially Stephen has been very helpful, with many fruitful discussion about the finer points of (un-)folding, and feedback on paper drafts. I also want to thank my colleagues at the STFC Rutherford Appleton Laboratory for their feedback and for providing a "collider physics perspective". This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) [grant number RO3625-1].

References

[1] L. Alvarez-Ruso et al. "NuSTEC white paper: status and challenges of neutrino–nucleus scattering". *Progress in Particle and Nuclear Physics* 100 (2018), pp. 1–68. DOI: 10.1016/j.ppnp.2018.01.006.

[2] Yoshi Uchida et al. "PhyStat-ν 2016 at the IPMU: summary of discussions" (June 28, 2018). arXiv: http://arxiv.org/abs/1806.10913v1 [hep-ex].

[3] P. Stowell et al. "NUISANCE: a neutrino cross-section generator tuning and comparison framework". *Journal of Instrumentation* 12.01 (Jan. 2017), P01016–P01016. DOI: 10.1088/1748-0221/12/01/p01016.

[4] Andy Buckley et al. *Rivet user manual*. Mar. 2, 2010. arXiv: http://arxiv.org/abs/1003.0694v8 [hep-ph].

[5] Robert D. Cousins, Samuel J. May, and Yipeng Sun. "Should unfolded histograms be used to test hypotheses?" *arXiv preprint* (2016). arXiv: http://arxiv.org/abs/1607.07038v1 [physics.data-an].

[6] P. C. Mahalanobis. “On the generalised distance in statistics”. *Proceedings National Institute of Science, India*. Vol. 2. 1. Apr. 1936, pp. 49–55. URL: https://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol02_1936_1_Art05.pdf.

[7] Daniel Fink. *A Compendium of Conjugate Priors*. Tech. rep. Environmental Statistics Group, Department of Biology, Montana State University, Jan. 1997.

[8] Rene Brun and Fons Rademakers. “ROOT — An object oriented data analysis framework”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1-2 (Apr. 1997), pp. 81–86. DOI: 10.1016/s0168-9002(97)00048-x.

[9] Lukas Koch. *ReMU - Response Matrix Utilities*. en. 2019. DOI: 10.5281/zenodo.1217572.

[10] Travis E. Oliphant. *Guide to NumPy*. USA: Trelgol Publishing, 2006.

[11] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: http://www.scipy.org/.

[12] C. Fonnesbeck et al. *PyMC: Bayesian Stochastic Modelling in Python*. URL: http://github.com/pymc-devs/pymc.

[13] *GitHub*. 2018. URL: http://github.com/.

[14] *PyPI - the Python Package Index*. 2018. URL: http://pypi.python.org/pypi.

[15] *YAML: YAML Ain’t Markup Language*. 2018. URL: http://yaml.org/.

[16] *pandas: Python Data Analysis Library*. URL: https://pandas.pydata.org/.

[17] Jim Pivarski et al. *scikit-hep/uproot*. 2019. DOI: 10.5281/zenodo.1173083.

[18] Lukas Koch. *ReMU – Response Matrix Utilities documentation*. 2019. URL: https://remu.readthedocs.io/.

[19] J. Neyman and E. S. Pearson. “On the Problem of the Most Efficient Tests of Statistical Hypotheses”. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 231.694-706 (Jan. 1933), pp. 289–337. DOI: 10.1098/rsta.1933.0009.

[20] S. S. Wilks. “The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses”. *The Annals of Mathematical Statistics* 9.1 (Mar. 1938), pp. 60–62. DOI: 10.1214/aoms/1177732360.

[21] Luv Demortier. *P Values and Nuisance Parameters*. eng. 2008. DOI: 10.5170/cern-2008-001.23.

[22] Paul Kabaila and Chris J. Lloyd. “A computable confidence upper limit from discrete data with good coverage properties”. *Statistics & Probability Letters* 47.2 (Apr. 2000), pp. 189–198. DOI: 10.1016/s0167-7152(99)00156-x.

[23] Murray Aitkin. “Posterior Bayes Factors”. *Journal of the Royal Statistical Society: Series B (Methodological)* 53.1 (Sept. 1991), pp. 111–128. DOI: 10.1111/j.2517-6161.1991.tb01812.x.

[24] I. Smith and A. Ferrari. "Generalizations related to hypothesis testing with the Posterior distribution of the Likelihood Ratio". *arXiv preprint* (2014). arXiv: 1406.1023v1 [physics.data-an].

[25] H. Jeffreys. "An Invariant Form for the Prior Probability in Estimation Problems". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 186.1007 (Sept. 1946), pp. 453–461. DOI: 10.1098/rspa.1946.0056.

A. Statistical methods

A.1. Simple hypotheses and absolute maximum likelihood

A *simple hypothesis* is completely characterised by the vector of truth expectation values $\boldsymbol{\mu}$. It has no free parameters. Each expectation value must be a non-negative real number, $\mu_j \in \mathbb{R}_{\geq 0}$. This defines the set of all conceivable hypotheses Ω:

$$\Omega = \mathbb{R}^d_{\geq 0}, \tag{A.1}$$

where $d = \dim(\boldsymbol{\mu})$ is the number of truth bins. We can thus define a maximum likelihood hypothesis $\boldsymbol{\mu}_{\max L}$ such that

$$L_{\max}(\Omega) = L(\boldsymbol{\mu}_{\max L}) = \max_{\boldsymbol{\mu}\in\Omega} L(\boldsymbol{\mu}). \tag{A.2}$$

This hypothesis and its likelihood value can then be used as a baseline to compare other hypotheses to it.

A.2. Likelihood ratio testing

The agreement between the data and any given hypothesis can be evaluated with the likelihood ratio λ:

$$\lambda(\boldsymbol{\mu}) = \frac{L(\boldsymbol{\mu})}{L_{\max}(\Omega)}. \tag{A.3}$$

By construction, this value is in the range $[0, 1]$. A high value shows good agreement, while low values indicate disagreement.

According to the *Neyman–Pearson lemma* [19], a hypothesis test using λ as a test statistic is the most powerful[17] test possible. So we define a critical value η and reject hypotheses where $\lambda < \eta$. The choice of η depends on the desired significance[18] of the test α and the expected distribution of the likelihood ratio $f(\lambda)$ given that $\boldsymbol{\mu}$ is true. It must be chosen such that

$$P(\lambda < \eta|\boldsymbol{\mu}) = \int_0^\eta f(\lambda|\boldsymbol{\mu})\,\mathrm{d}\lambda \overset{!}{\leq} \alpha. \tag{A.4}$$

The distribution of λ must be evaluated for each tested hypothesis separately, for example by doing a sufficient number of MC experiments. The critical value is thus a function of the hypothesis $\eta(\boldsymbol{\mu})$. This complicates direct comparisons of different hypotheses.

[17] The power of a test $1 - \beta$ describes the probability of rejecting a false hypothesis.
[18] The significance of a test α describes the probability of rejecting a true hypothesis.

To get a value that is comparable between hypotheses, one can use the likelihood ratio p-value as test statistic directly. It is the probability of measuring a likelihood "worse" than the actually measured one λ_0, assuming that the tested hypothesis is true:

$$p_\lambda(\boldsymbol{\mu}) = P(\lambda < \lambda_0|\boldsymbol{\mu}) = \int_0^{\lambda_0} f(\lambda|\boldsymbol{\mu})\,\mathrm{d}\lambda. \tag{A.5}$$

By construction, this value is uniformly distributed for the true hypothesis, so the critical value is just the significance, and hypotheses are rejected if

$$p_\lambda(\boldsymbol{\mu}) < \alpha. \tag{A.6}$$

A.3. Composite hypotheses

Often a tested hypothesis will have some free (nuisance) parameters. Those are called *composite hypotheses*. They will define the truth expectation values $\boldsymbol{\mu}$ as a function of these free parameters $\boldsymbol{\mu}(\boldsymbol{\theta})$, with the number of free parameters $d' = \dim(\boldsymbol{\theta}) < d$. The possible values of $\boldsymbol{\theta}$ define the set of *simple* hypotheses Θ, which is a subset of all conceivable hypotheses:

$$\boldsymbol{\mu}(\boldsymbol{\theta}) \in \Theta \subset \Omega \quad \Leftrightarrow \quad \boldsymbol{\theta} \in \omega, \tag{A.7}$$

where ω is the set of allowed values of $\boldsymbol{\theta}$. For example, if all parameters are unrestricted real values, we have

$$\omega = \mathbb{R}^{d'}. \tag{A.8}$$

We consider a composite hypothesis *true* if it contains the true simple hypothesis $\boldsymbol{\mu}_{\text{true}}$, and *false* otherwise.

Again, we would like to test hypotheses with the highest possible power at a given significance. To reject a composite hypothesis Θ, we must reject all contained simple hypotheses $\boldsymbol{\mu}(\boldsymbol{\theta})$:

$$\lambda(\boldsymbol{\mu}) < \eta(\boldsymbol{\mu}) \quad \forall \quad \boldsymbol{\mu} \in \Theta. \tag{A.9}$$

As an approximation with lower than ideal power, we can consider the maximum likelihood and minimum critical value:

$$\lambda_{\max}(\Theta) = \max_{\boldsymbol{\mu}\in\Theta} \lambda(\boldsymbol{\mu}) \overset{!}{<} \eta_{\min}(\Theta) = \min_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu}). \tag{A.10}$$

Depending on the variation of $\eta(\boldsymbol{\mu})$ within Θ,[19] the significance of the test will be less than or equal to the nominal value α.

To increase the power of the test, we can use the p-values of the likelihood ratios directly. With this test statistic, the critical value is identical for all simple hypotheses, and we reject a composite hypothesis if

$$p_\lambda(\boldsymbol{\mu}) < \alpha \quad \forall \quad \boldsymbol{\mu} \in \Theta. \tag{A.11}$$

That means we can exclude a composite hypothesis by checking whether

$$p_{\max}(\Theta) = \max_{\boldsymbol{\mu}\in\Theta} p_\lambda(\boldsymbol{\mu}) < \alpha \tag{A.12}$$

[19] In the limit of large sample sets, the distribution of $\lambda_{\max}$ will approach a χ^2-distribution [20] and an exact value for η can be chosen accordingly.

with maximum power. This is called the "supremum method" [21] and not computationally harder than finding $\eta_{\min}$. In both cases $f(\lambda|\boldsymbol{\mu})$ has to be calculated for each evaluation in the minimisation/maximisation process.

A.4. Parameter estimation

If a composite theory Θ is not rejected, one might want to quote a set of "best fit" parameters and/or a range of allowed values, i.e. confidence intervals. The maximum likelihood point estimator for the parameters $\hat{\boldsymbol{\theta}}$ is straight forward. It is the set of parameters that produce the highest likelihood:

$$L(\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})) = L_{\max}(\Theta). \tag{A.13}$$

Confidence intervals for the parameters can be calculated by rejecting part of the possible parameter space analogously to the general composite hypothesis test in section A.3. For this, we split the parameters into interesting parameters $\boldsymbol{\theta}$, where we want to quote the intervals, and nuisance parameters $\boldsymbol{\phi}$. We then interpret the set of all $\boldsymbol{\mu}(\boldsymbol{\theta}, \boldsymbol{\phi})$ with a fixed $\boldsymbol{\theta}$ as a new composite hypothesis $\Theta(\boldsymbol{\theta})$:

$$\boldsymbol{\mu}(\boldsymbol{\theta}, \boldsymbol{\phi}) \in \Theta(\boldsymbol{\theta}) \subset \Theta \quad \Leftrightarrow \quad \boldsymbol{\phi} \in \Phi(\boldsymbol{\theta}), \tag{A.14}$$

where $\Phi(\boldsymbol{\theta})$ is the set of allowed values of $\boldsymbol{\phi}$ given a specific $\boldsymbol{\theta}$. Now we can exclude values of $\boldsymbol{\theta}$ by checking whether

$$p_{\max}(\Theta(\boldsymbol{\theta})) < \alpha. \tag{A.15}$$

Those values of $\boldsymbol{\theta}$ that have not been rejected define the confidence region.

It might be useful to construct confidence intervals for parameters of composite hypotheses whether or not they have been excluded. In these cases, one is usually only interested in the allowed parameter range within the context of the analysed hypotheses. This can easily be achieved by replacing the absolute maximum likelihood $L_{\max}(\Omega)$ with the maximum likelihood of the hypothesis $L_{\max}(\Theta)$. The likelihood ratio is then

$$\lambda(\boldsymbol{\mu}) = \frac{L(\boldsymbol{\mu})}{L_{\max}(\Theta)}, \tag{A.16}$$

and the construction of the confidence interval only ever compares the nested hypothesis $\Theta(\boldsymbol{\theta})$ directly with the enveloping hypothesis Θ. This can reduce the number of parameters considerably, as no evaluation of the absolute maximum likelihood needs to be done. A lower number of free parameters decreases the computational load considerably.

A.5. Profile plug-in p-values

Even when only comparing two hypotheses with moderate number of parameters, finding $p_{\max}(\Theta(\boldsymbol{\theta}))$ is a computationally intensive task. Calculating the p-value for a single $\boldsymbol{\mu}$ takes the generation of $O(100)$ toy data sets from the reco predictions of that hypothesis, and then maximising the likelihoods of both compared composite hypotheses for each data set. Maximising the p-value with a typical optimisation algorithm means that it has to be evaluated at least $O(10000)$ times, depending on the difficulty of finding the global(!) likelihood maxima. This quickly escalates into millions upon millions necessary fits and a corresponding demand of computing power.

A drastic reduction can be achieved when using the "profile plug-in" p-value instead of the maximum p-value. Instead of maximising the p-value over all possible hypotheses $\boldsymbol{\mu} \in \Theta$, one only evaluates the p-value of the most likely hypothesis $\hat{\boldsymbol{\mu}}$:

$$p_{\text{plug}}(\Theta) = p_\lambda(\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})), \tag{A.17}$$

or in the context of parameter estimation:

$$p_{\text{plug}}(\Theta(\boldsymbol{\theta})) = p_\lambda(\boldsymbol{\mu}(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})). \tag{A.18}$$

Here $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$ are the maximum likelihood estimates of the (nuisance) parameters:

$$L(\boldsymbol{\mu}(\hat{\boldsymbol{\theta}})) = L_{\max}(\Theta), \tag{A.19}$$

$$L(\boldsymbol{\mu}(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})) = L_{\max}(\Theta(\boldsymbol{\theta})). \tag{A.20}$$

The calculation of this value requires only a single optimisation of the likelihood. The p-value itself is then computed with toy data assuming the truth of the estimate.

This is called "profile plug-in" p-value as we plug-in the profile maximum likelihood estimate for the nuisance values as an estimate for the distribution of the likelihood ratios of the true hypothesis. The method has certain advantages over other approximation methods [22], but it is still an approximation. It is thus important to check the coverage properties of any analysis using this method.

A.6. Bayesian posterior sampling

The exact frequentist approaches described above need a prohibitive amount of computing power when the number of parameters of the tested composite hypothesis is large. Those models are better handled by a Bayesian approach. Using a Markov Chain Monte Carlo (MCMC) method, it is relatively easy to sample parameter sets $\boldsymbol{\theta}$ from the posterior probability

$$P(\boldsymbol{\theta}|\boldsymbol{n}) \propto L(\boldsymbol{\mu}(\boldsymbol{\theta}))\, P(\boldsymbol{\theta}), \tag{A.21}$$

with the prior probability $P(\boldsymbol{\theta})$. These sets can then be used to infer information about the parameters, e.g. point estimates or credible intervals, and to compare different hypotheses with one another.

Hypothesis comparisons are usually done with the *Bayes factor* K:

$$B = \frac{\int L(\boldsymbol{\mu}(\boldsymbol{\theta}_0)) P(\boldsymbol{\theta}_0) \mathrm{d}\boldsymbol{\theta}_0}{\int L(\boldsymbol{\mu}(\boldsymbol{\theta}_1)) P(\boldsymbol{\theta}_1) \mathrm{d}\boldsymbol{\theta}_1}, \tag{A.22}$$

i.e. the ratio of prior mean likelihoods. To Bayesian posterior odds K are then just the Bayes factor multiplied by the prior odds:

$$K = B \frac{P(\Theta_0)}{P(\Theta_1)}. \tag{A.23}$$

When averaging the likelihood over the posterior distributions of the parameters, one gets the *posterior Bayes factor* B_{post} [23]:

$$B_{\text{post}} = \frac{\int L(\boldsymbol{\mu}(\boldsymbol{\theta}_0)) P(\boldsymbol{\theta}_0|\boldsymbol{n}) \mathrm{d}\boldsymbol{\theta}_0}{\int L(\boldsymbol{\mu}(\boldsymbol{\theta}_1)) P(\boldsymbol{\theta}_1|\boldsymbol{n}) \mathrm{d}\boldsymbol{\theta}_1}. \tag{A.24}$$

Using the posterior Bayes factor reduces the influence of the priors of the parameters on the result, but it is sometimes criticised for "using the data twice": once for determining the posterior distributions of the parameters $P(\boldsymbol{\theta}|n)$ and once for calculating the likelihood $L(\boldsymbol{\mu}(\boldsymbol{\theta}))$ to be averaged over those distributions.

Another possible method is using the *Posterior distribution of the Likelihood Ratio* (PLR) to infer the data preference of one model over another. The PLR is defined as the posterior probability of the likelihood ratio of the compared hypotheses being below or equal to a certain threshold value:

$$\mathrm{PLR}_{\Theta_0,\Theta_1}(\boldsymbol{n},\zeta) = P\left(\frac{L(\boldsymbol{\mu}(\boldsymbol{\theta}_0))}{L(\boldsymbol{\mu}(\boldsymbol{\theta}_1))} \leq \zeta \,\middle|\, \boldsymbol{\theta}_0 \sim P(\boldsymbol{\theta}_0|\boldsymbol{n}), \boldsymbol{\theta}_1 \sim P(\boldsymbol{\theta}_1|\boldsymbol{n})\right), \tag{A.25}$$

with $\boldsymbol{\mu}(\boldsymbol{\theta}_0) \in \Theta_0$ and $\boldsymbol{\mu}(\boldsymbol{\theta}_1) \in \Theta_1$ the (completely independent) parametrisations of the tested hypotheses. If the threshold value ζ is set to 1, the PLR is equivalent to a frequentist p-value under certain circumstances [24]. But even when this is not the case, the interpretation is straight forward: $\mathrm{PLR}_{\Theta_0,\Theta_1}(\boldsymbol{n},\zeta = 1)$ is the posterior probability of the data being more likely under Θ_1 than under Θ_0.

For all Bayesian analyses, it is important to choose suitable priors $P(\boldsymbol{\theta})$. There is no single "correct" way to do this, but one useful "non informative" prior is the *Jeffreys prior* [25]. Its main advantage is that its probability density – and especially the posterior probability density resulting from using this prior – is *invariant* under variable transformations. This means that the results of the analysis do *not* depend on the particular parametrisation of $\boldsymbol{\mu}(\boldsymbol{\theta})$. A drawback of Jeffreys priors is that they are not necessarily proper, i.e. they cannot always be normalised. This is not necessarily a problem here though, as long as the posterior is well defined.

The Bayesian approach treats all unknown parameters equal. It is therefore natural to also include the detector uncertainties in the MCMC sampling. We simply treat the detector toy index as additional (nuisance) parameter of the model. The posterior probability thus also includes information about how likely or unlikely the different toy detectors are:

$$\begin{aligned} P(\boldsymbol{\theta}, t|\boldsymbol{n}) &\propto L^t(\boldsymbol{\mu}(\boldsymbol{\theta}))\, P(\boldsymbol{\theta}, t) \\ &= P(\boldsymbol{n}|\boldsymbol{\mu}(\boldsymbol{\theta}), R^t)\, P(\boldsymbol{\theta}, t). \end{aligned} \tag{A.26}$$