What we measure when we measure σ

Template fitting as a cross-section extraction technique at T2K

***Andrew Cudd**, Stephen Dolan, Ciro Riccio, Ka Ming Tsui*

Contact details for any questions
(or find us on the Nuisance Slack Channel!)

University of Colorado **Boulder**

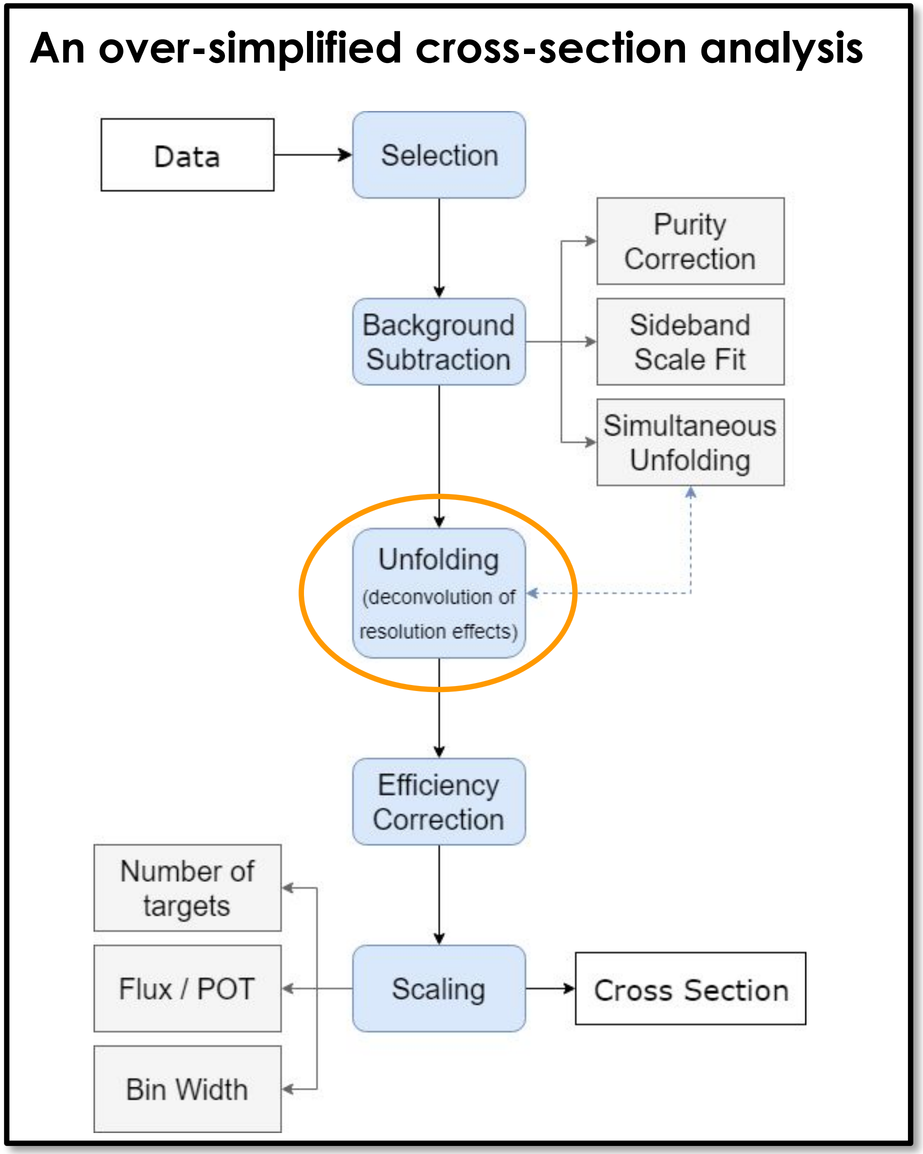
Overview

- This talk is meant to be a conceptual overview to how template fitting works as a cross section extraction method, and how it generally is used for T2K analyses
- Will also briefly cover the validation and diagnostics provided by the T2K implementation of template fitting
- This isn't going to be a tutorial on how to use T2K's code to do this (if there's demand, this is something we can consider in the future!)
- In this talk I also won't really explore in detail the caveats of unfolding (e.g. flux shape issues or PPP) or alternatives to it (e.g. ReMU).

PPP = Peelle's Pertinent Puzzle

Template fitting as a method of unfolding

Unfolding



- Unfolding is a key part of cross-section analyses
- It is the process of deconvolving detector resolution effects from data
- (Almost) all recent results which can be compared to theory/generator predictions are unfolded
- Unfolding without care can bias your result

Unfolding

- Measure ***selected*** number of **events** in bins of a ***reconstructed*** quantity

Efficiency correct

Bkg subtract

Unfolding

- Want the ***total*** number of **signal events** in bins of a ***true*** quantity

Assuming no background

Number of events in reco bin j

$$R_j = \sum_{True\ Bins,i} S_{ji} T_i$$

Number of events in true bin i

Smearing matrix

Number of events in true bin i

$$T_i = \sum_{Reco\ Bins,j} U_{ij} R_j$$

Number of events in reco bin j

Unsmearing matrix

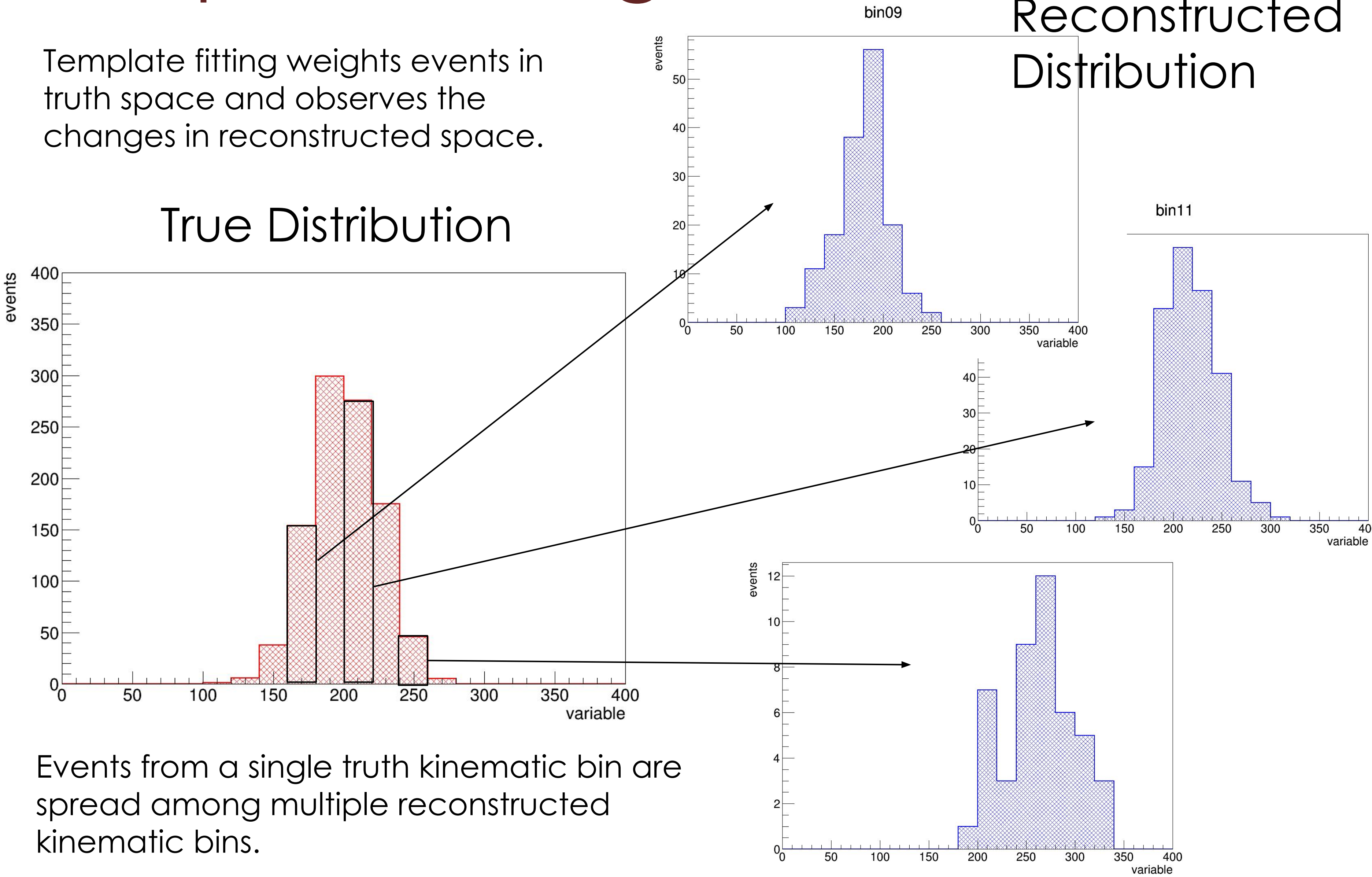
- Unfolding is finding the unsmearing matrix **U** from **S**
- Simplest method is to use the inverse of the smearing matrix **S**
- Reco (with detector effects) and true (without detector effects) bin refers to the event kinematics

Template fitting

Template fitting weights events in truth space and observes the changes in reconstructed space.

True Distribution

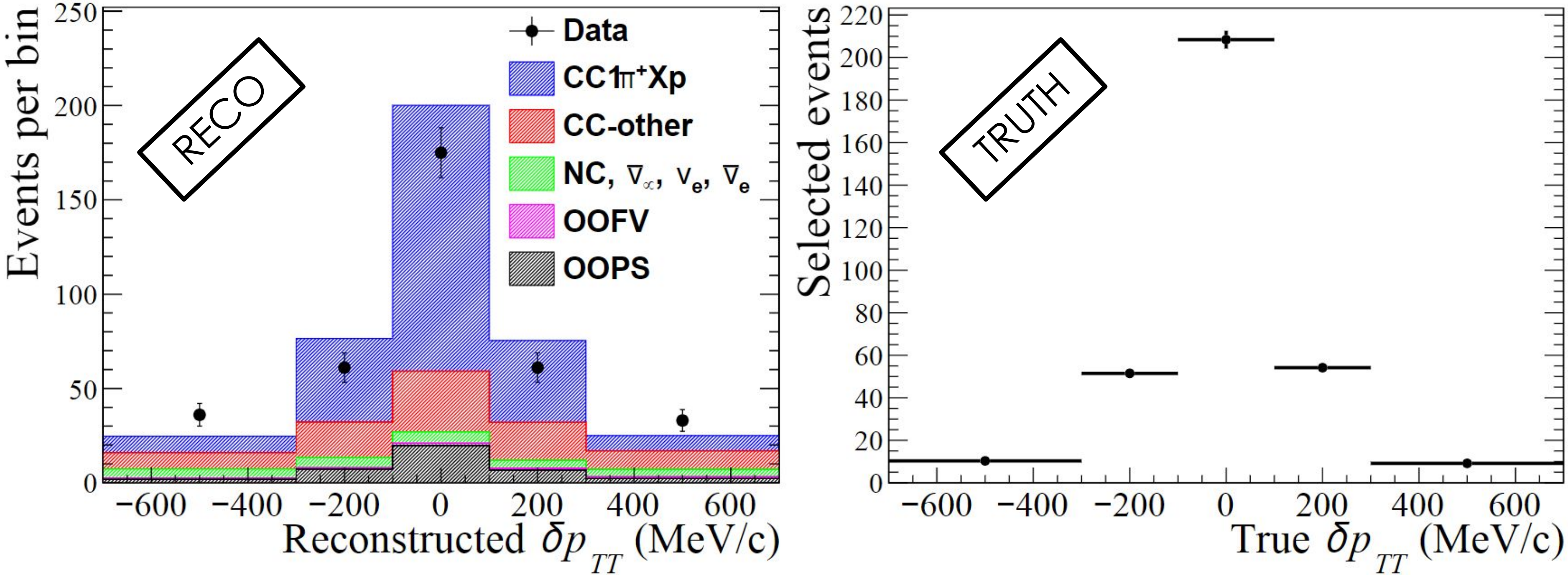
Reconstructed Distribution



Events from a single truth kinematic bin are spread among multiple reconstructed kinematic bins.

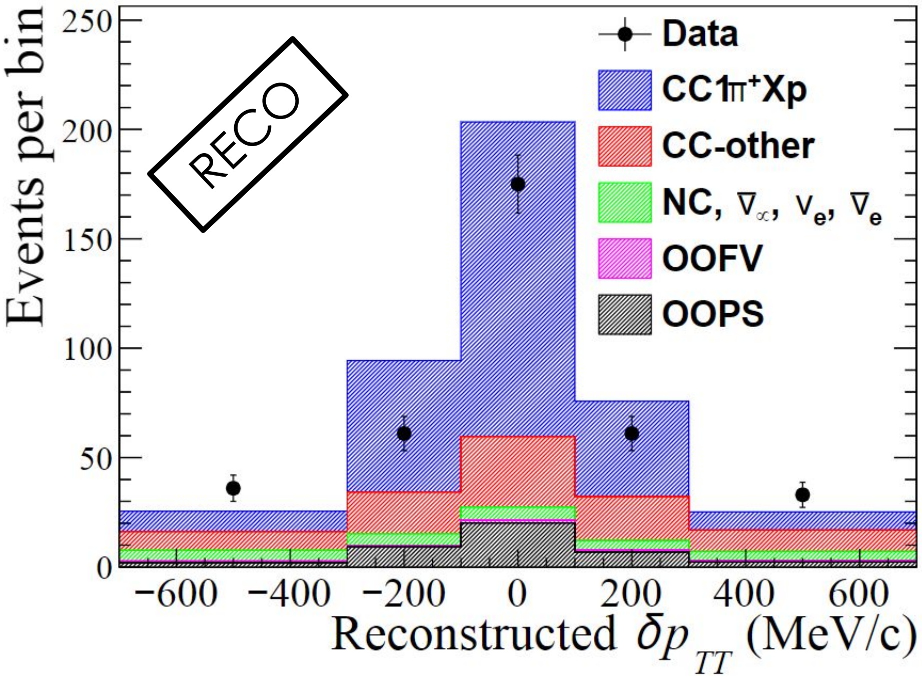
Template fitting

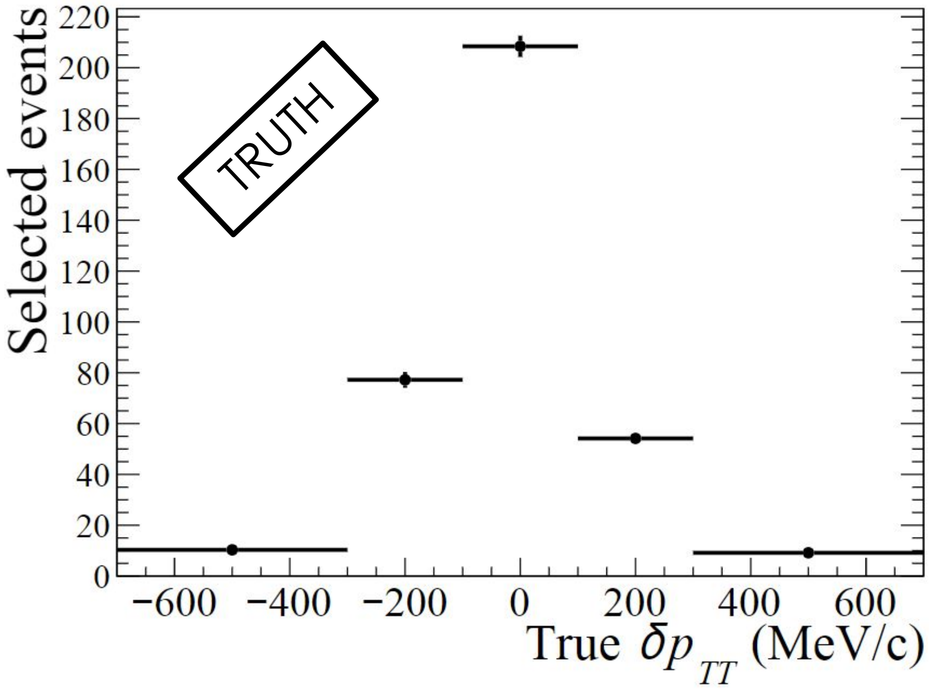
Example using the CC1pi+ plus protons transverse kinematic imbalance (TKI) analysis: https://arxiv.org/abs/2102.03346



Reconstructed event distribution on the left, true selected signal event distribution (in signal phase-space) on the right for the double transverse momentum.

Template fitting

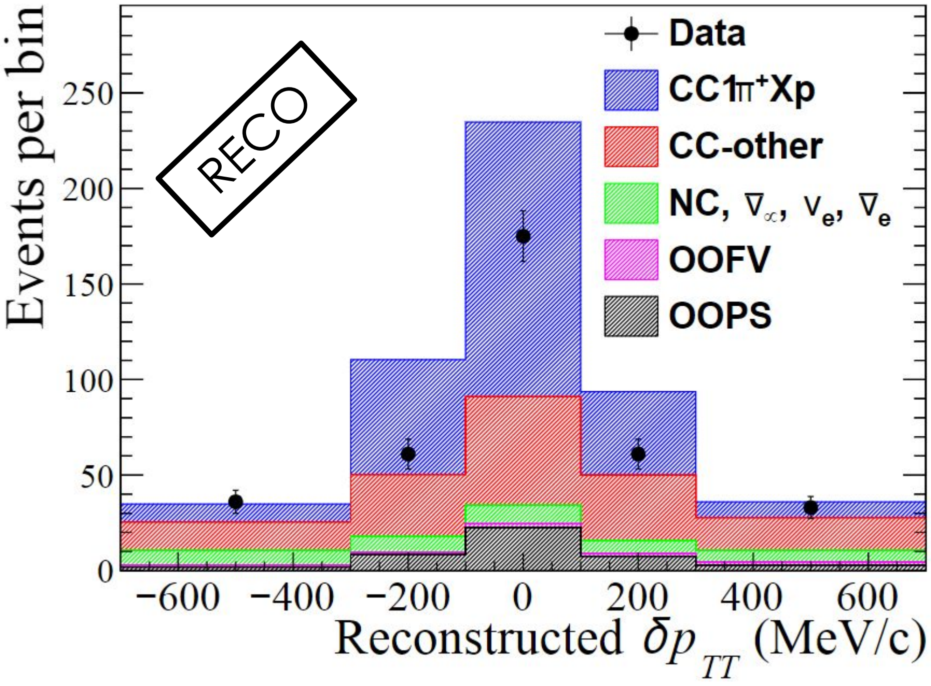


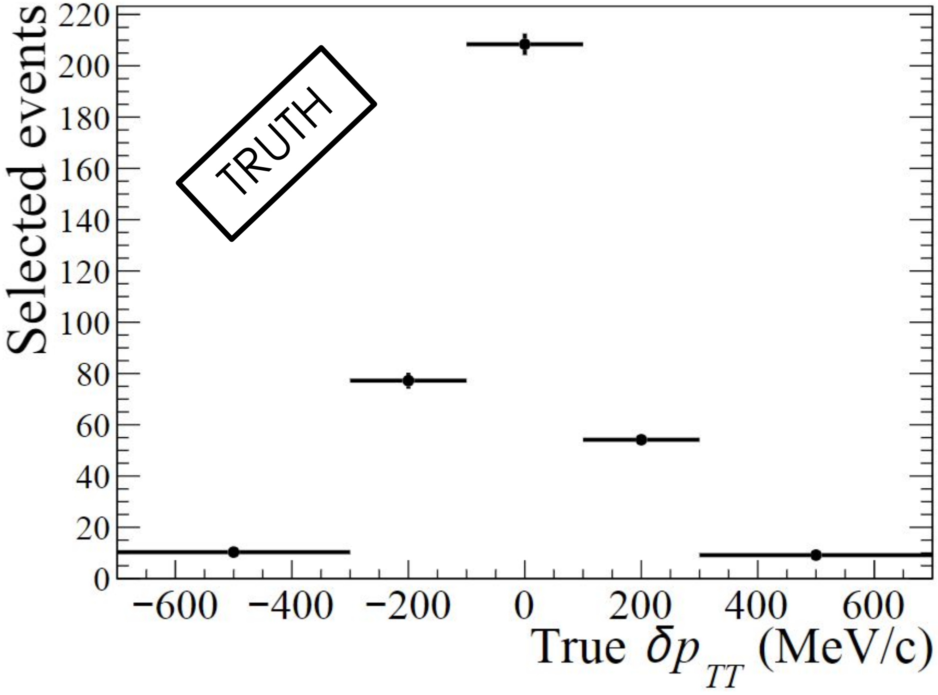


Adjust one template parameter in truth space and see how it affects reconstructed space. Note how only the signal events are affected.

One template parameter per bin in truth space.

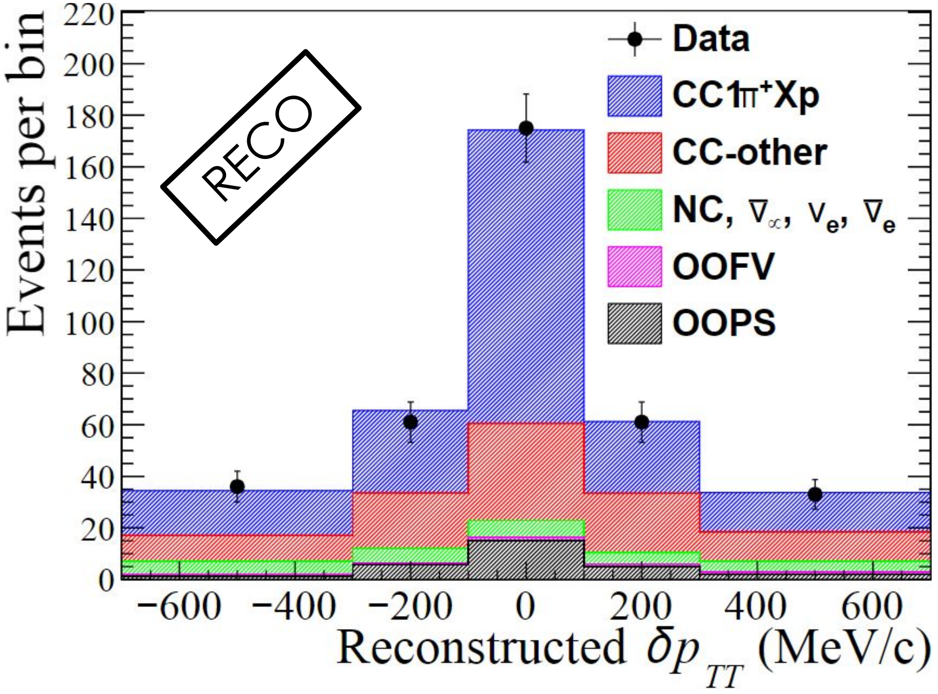
Template fitting

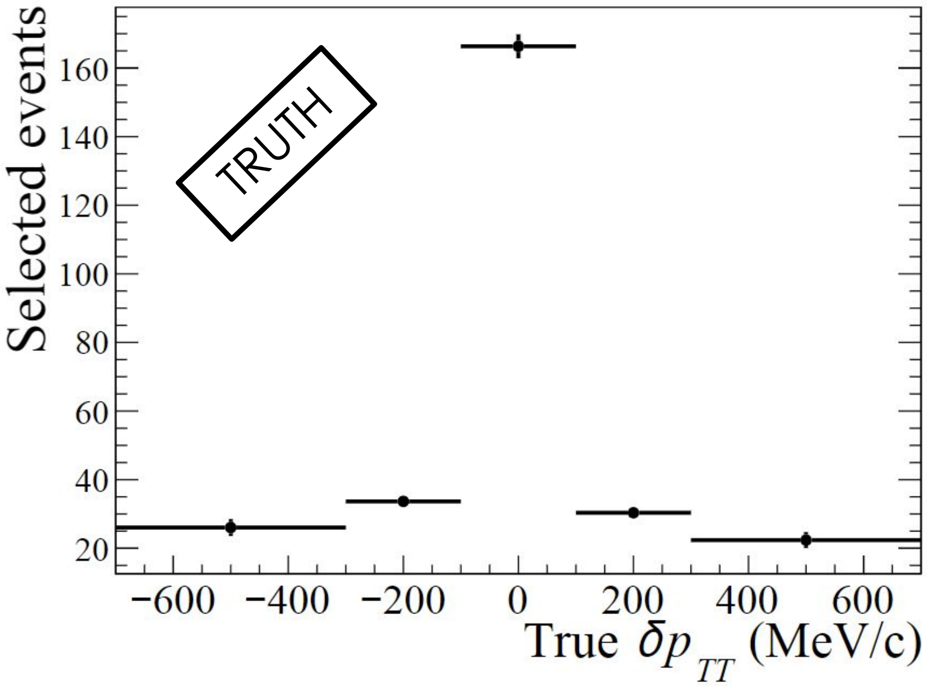




Adjust background contribution to event rate. Note how the true distribution of signal events is unaffected.

Template fitting





Adjust all parameters iteratively to achieve the best agreement between data and MC in reconstructed space.

For the T2K fit this is done by minimizing a chi-square metric via MINUIT.

Fit Parameters in a nutshell

For T2K analyses we generally have four types (or classes) or parameters:

Template parameters (~10 to 200, very analysis dependent)

- Apply weights to signal events as a function of the analysis "truth" bins
- One template parameter per truth bin

Flux parameters (10 to 40, depending on relevant fluxes or neutrino flavors)

- Normalization parameters/weights binned in true neutrino energy
- Highly correlated prior uncertainties

Neutrino interaction model parameters (10 to 30, depends on background events)

- Weights applied per event based on pre-calculated cubic splines
- Splines calculated for specific parameter values and interpolated

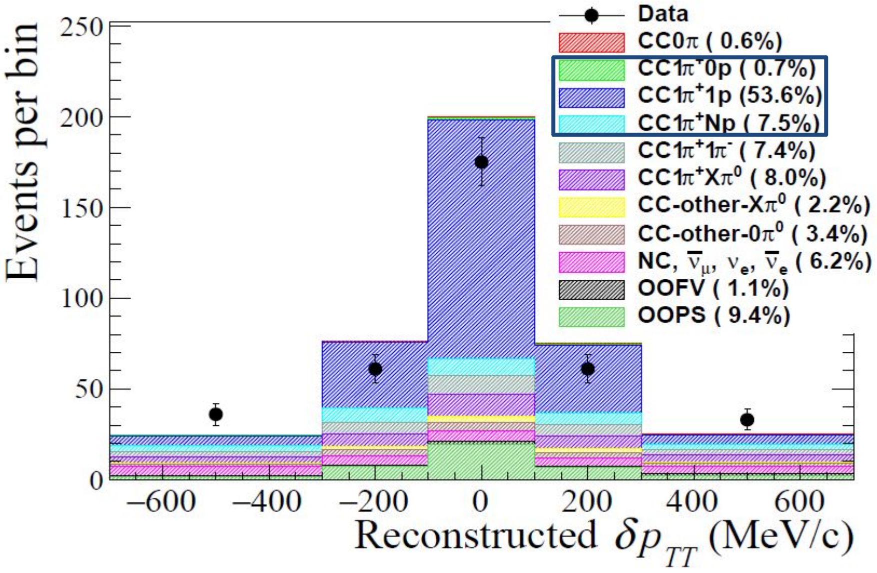
Detector parameters (50 to 1000, very analysis dependent)

- Normalization parameter/weight for every reconstructed bin in the analysis
- Large to highly correlated prior uncertainties
- Can apply Principal Component Analysis (PCA) to reduce the number of free detector parameters (can also be applied to any parameter set)

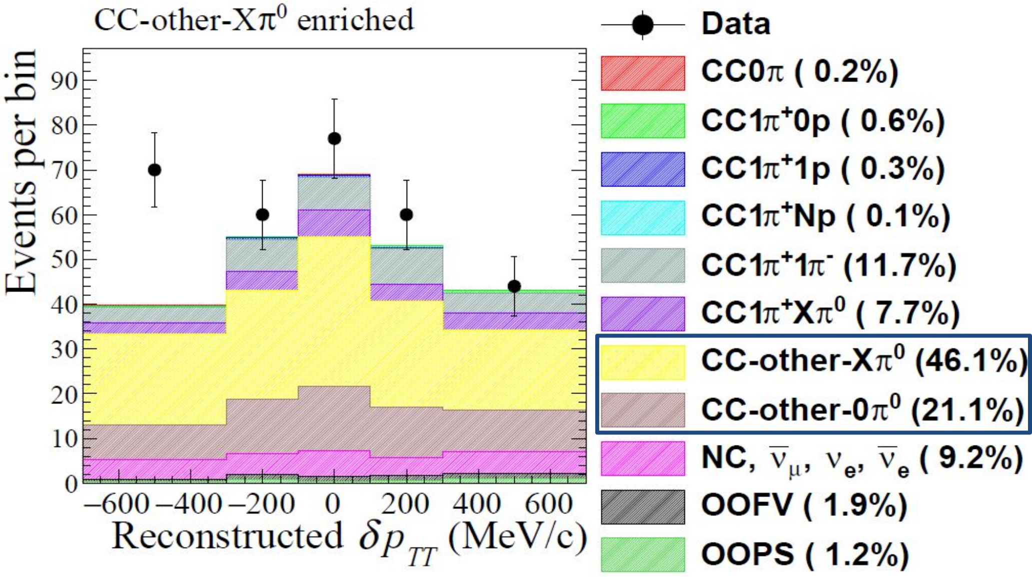
Fit Samples

- The fit is very free to alter the signal process via the template parameters by construction (e.g. no prior uncertainty)
- Signal dominated samples normally offer very little constraint on non-template parameters
- Other parameters (xsec model, flux, detector response) can be constrained by dedicated control samples*

CC1pi+Np Signal Sample



DIS/Pi-Zero Enriched Sample



Caveat: a control sample(s) is unlikely to contain identical physics to a background in the signal sample(s). Understanding how reliably constraints from the control region can be extrapolated into the signal region is crucial.

Likelihood components

The best fit parameters are those that maximize the likelihood, or equivalently minimize the following chi-square approximation (χ2 ~ -2ln(L)):

Total chi-square: the test statistic minimized by the fit.

$$\chi^2 = \chi^2_{\text{stat}} + \chi^2_{\text{syst}} = -2\ln\mathcal{L}_{stat} - 2\ln\mathcal{L}_{syst}$$

Poisson likelihood: how well the MC matches the data.

$$-2\ln\mathcal{L}_{stat} = \sum_{j}^{\text{bins}} 2\left(N_j^{\text{exp}} - N_j^{\text{obs}} + N_j^{\text{obs}}\ln\frac{N_j^{\text{obs}}}{N_j^{\text{exp}}}\right)$$

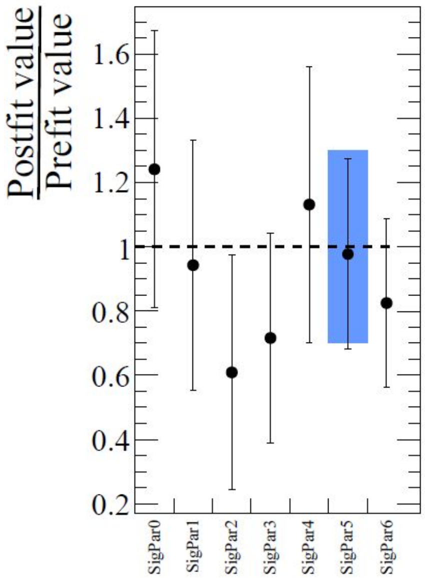
Penalty term: penalizes fit for moving systematic parameters away from nominal.

$$-2\ln\mathcal{L}_{syst} = \sum_{p} \left(\vec{p}-\vec{p}_{\text{prior}}\right)\left(V^{\text{syst}}_{\text{cov}}\right)^{-1}\left(\vec{p}-\vec{p}_{\text{prior}}\right)$$

Further details:

- The T2K fitter includes the option to modify the likelihood to include the MC statistical uncertainty following a variation of the Barlow-Beeston method
- Optional regularization terms/contributions can be included

The Fit Output

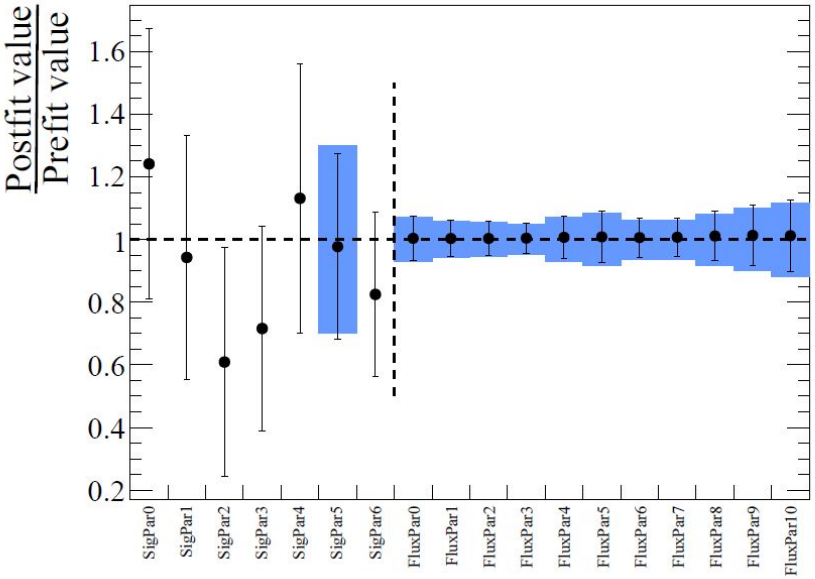


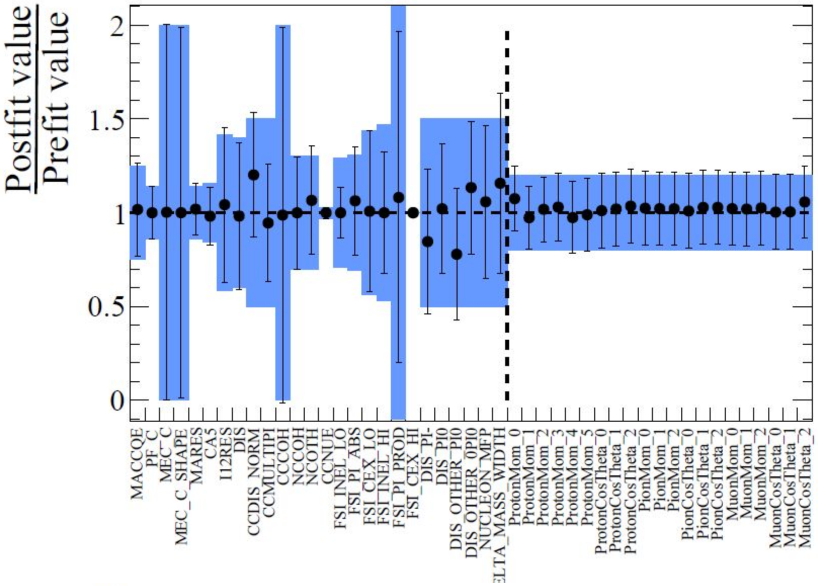
Fit output for a mock data study fitting different versions of T2K MC (e.g. different NEUT versions).

Blue band shows prior uncertainty. Error bars indicate post-fit uncertainty.

Template parameters (in general) have no prior uncertainty.

The Fit Output

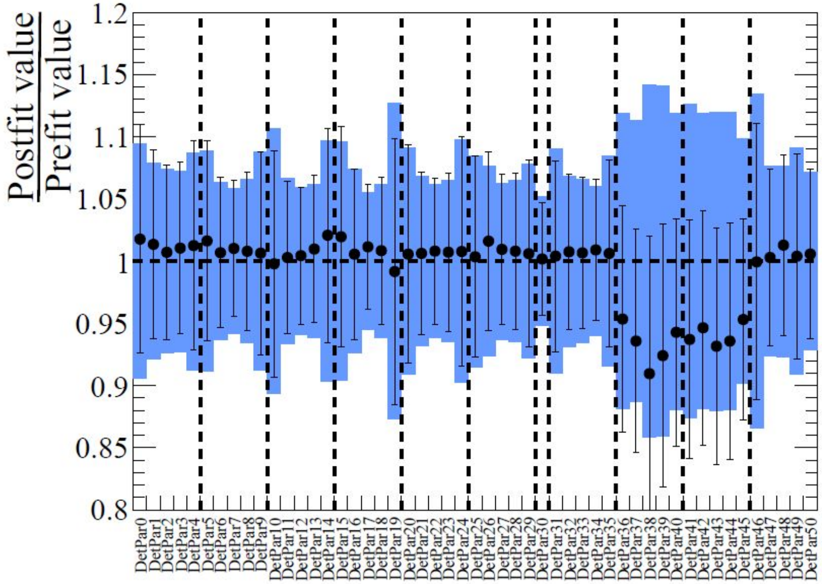




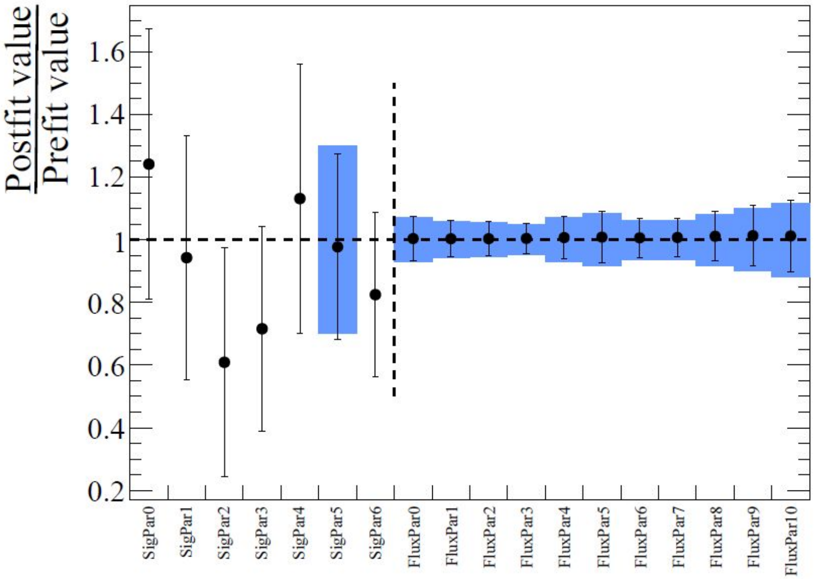
In addition to the template parameters, the pre/post-fit values and uncertainties for each parameter are available.

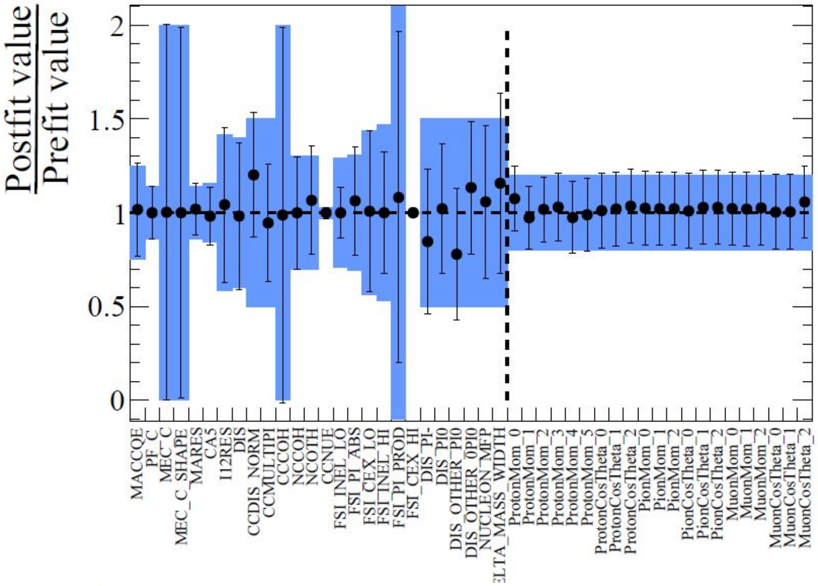
Systematic/nuisance parameters have some prior uncertainty, and are further constrained by control samples.

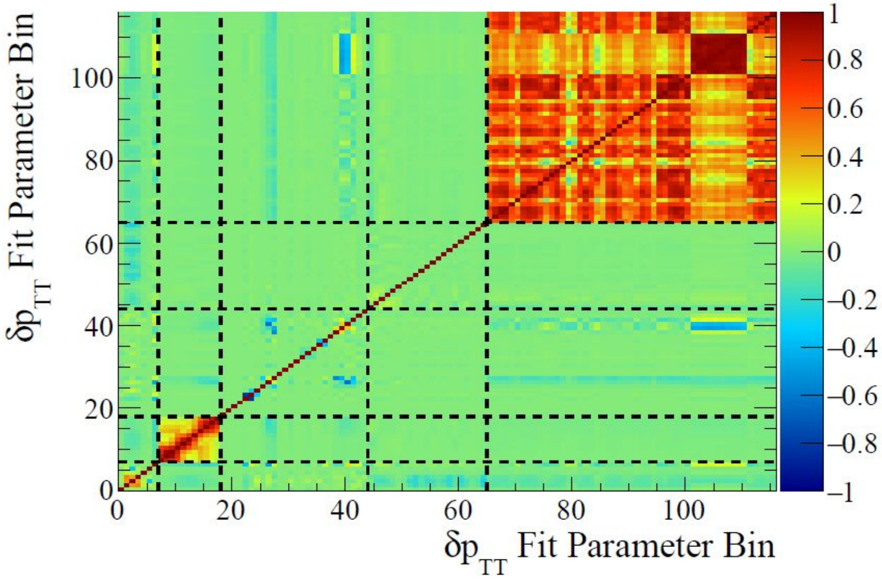
The main fit output or "result" consists of the parameter values, uncertainties, and their correlations.

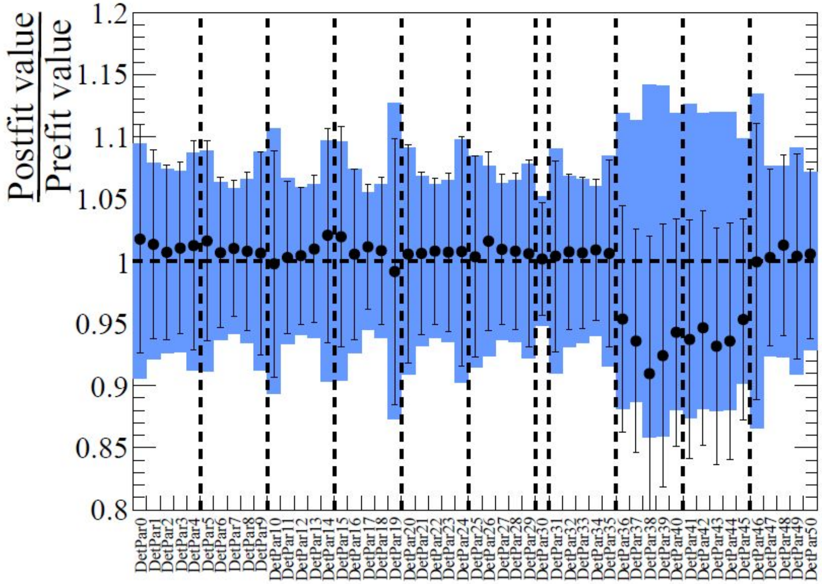


The Fit Output









Cross-section extraction following a fit

From a fit result to a cross section

- The fit parameters by themselves are not a cross section -- need to use full fit result to calculate the cross section and uncertainty.
- The uncertainty on the fit parameters directly corresponds to an uncertainty on the number of signal events

$$\left(\frac{d\sigma_{flux\ integrated}}{dx}\right)_i = \frac{\boxed{N_i^{signal}}}{\boxed{\epsilon_{i,sim}^{signal}}\boxed{\Phi}N_{targets}^{FV}} \times \frac{1}{\Delta x_i}$$

(*i* is the bin index)

- The post-fit systematic parameters also predict the **flux integral** and **efficiency** (with associated uncertainties)
- Need a method to propagate the uncertainties of fit parameters to the cross section distribution...

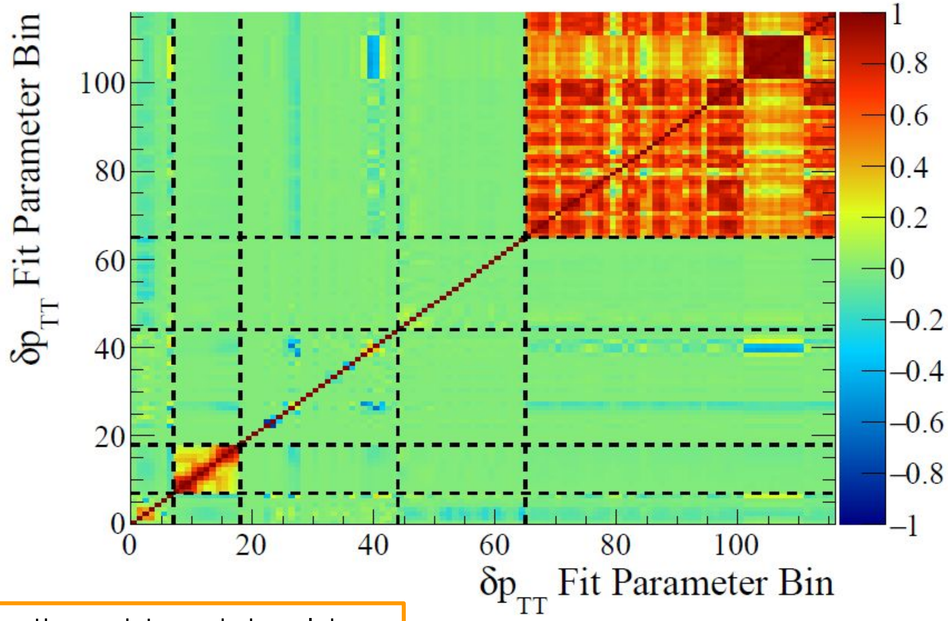
Caution surrounding the efficiency correction is another whole can of worms that I won't discuss here. Suffice to say the efficiency correction should ideally be made in all interaction model-related variables which characterise the detectors acceptance.

From a fit result to a cross section

Numerical propagation method of the fit parameter uncertainties.

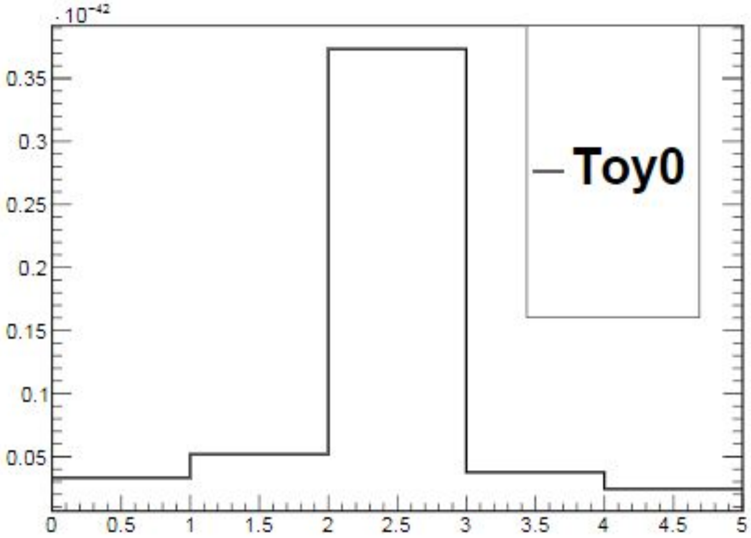
1. Generate a random sample of the parameters using the post-fit covariance matrix
2. Generate random values for additional parameters not included in the fit according to their priors (e.g. number of targets)
3. Calculate the cross section using the sampled parameters*
4. Repeat thousands of times.

This builds a distribution of cross sections toys/universes that represents the central value and the uncertainty band.

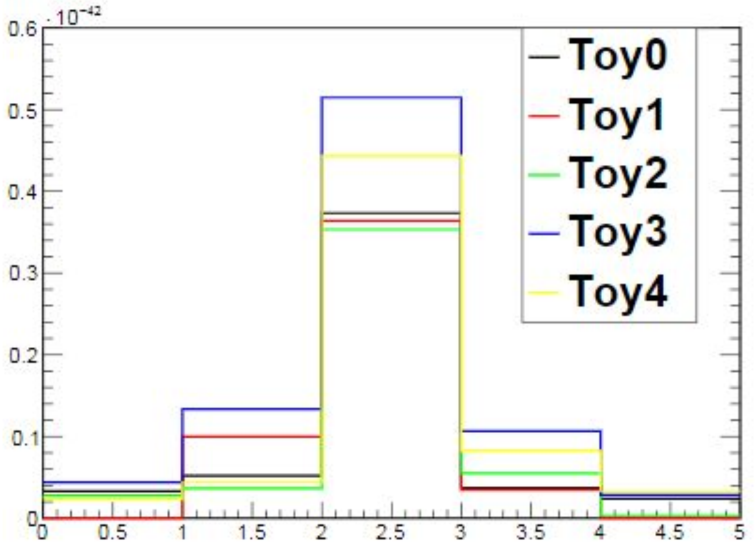


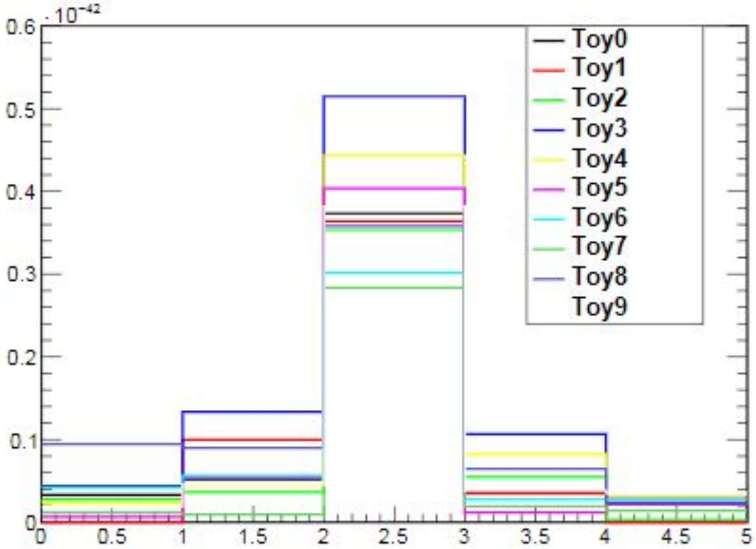
*Caveat: The treatment of the flux parameters needs some thought and should depend on what exactly you intend to measure (Phys. Rev. D **102**, 113012)

From a fit result to a cross section

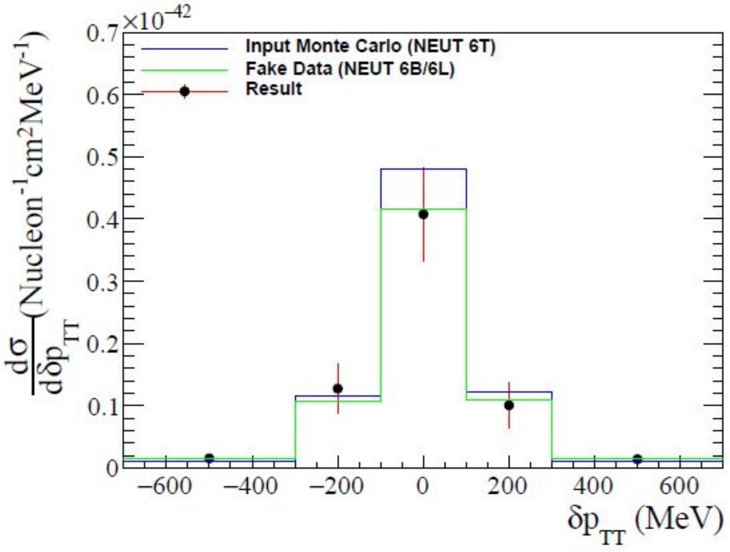


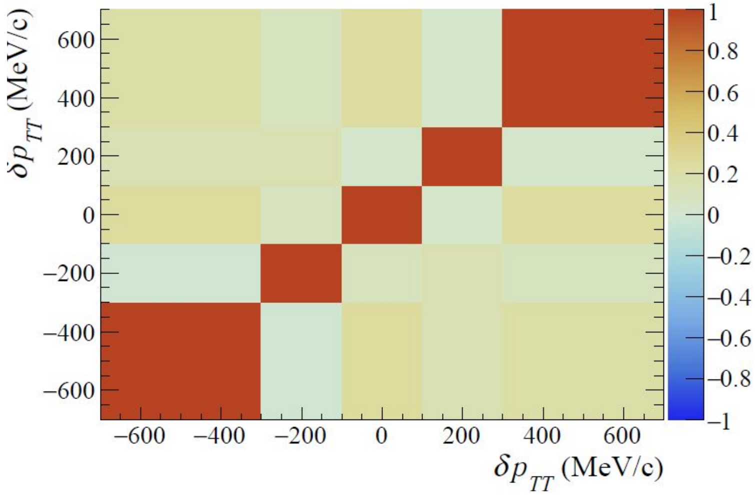
Example of cross section toys or universes for the double transverse momentum.





From a fit result to a cross section





Over many toy sample produce cross section result and uncertainties

To summarize:

- For each sample (toy or universe) we calculate the cross section.
- With many toy cross section results we can construct uncertainties.
- This can be expressed as a best fit cross section and an accompanying covariance matrix*

*Caveat: this a strong Gaussian error approximation that may not always be valid! If it’s not, then best to publish the toys too.

So far so good ...

Template fitting is a conceptually straightforward unfolding and cross section extraction method with a number of benefits:

- Similarities with neutrino oscillation fits
- Simple method of constraining backgrounds
- No “built-in” regularisation: the unfolding part is model independent
    - *N.B. Model bias can still enter via integration over kinematic variables with non-flat efficiency or via background constraints*
    - *In the no-background case with the same reco and true binning, the fit is equivalent to D’Agostini with infinite iterations (or just matrix inversion)*
    - *Can still include a regularization scheme in the fit (not discussed here)*

Key outstanding question: **how do I know I can trust my fit result?**

- Lots of diagnostics available! See next slides.

Fit Diagnostics

Fit diagnostics

The fit output (for real or mock data) contains a variety of information which can be used to assess the validity of the fit, including:

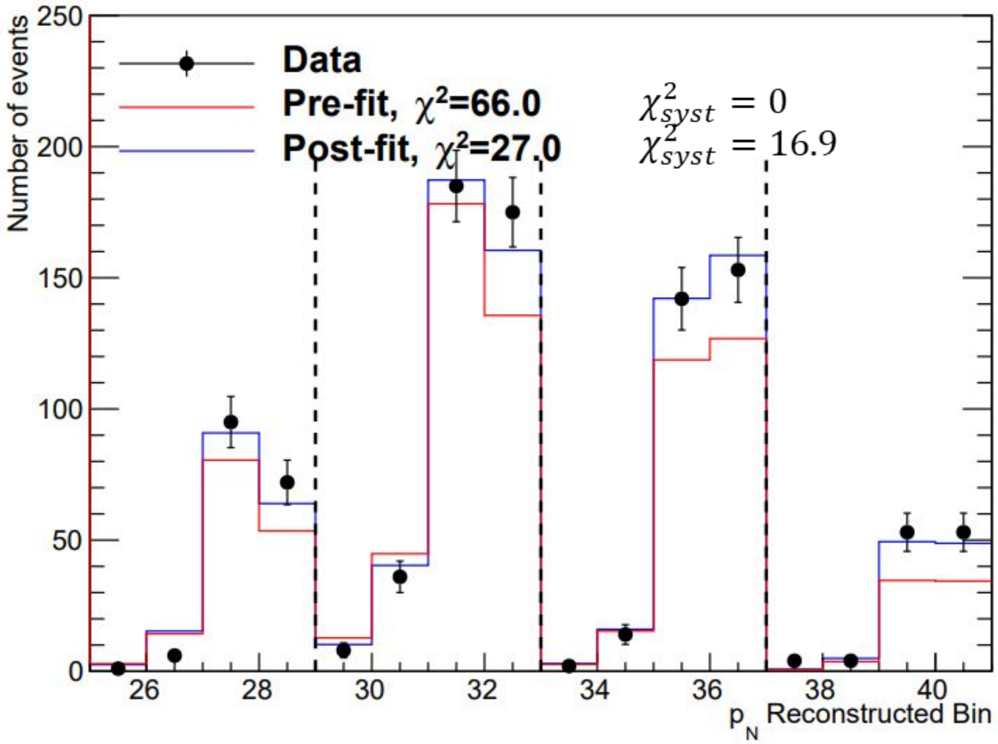
- **The post-fit set of parameter values and covariance**
  - Can be used to judge if the pulls are reasonable given the input model
  - Can be used to calculate p-values for each parameter set
  - Behavior of model parameters can be studied with mock data studies
- **Likelihood scans of each parameter around the best fit point**
  - Can check how Gaussian the likelihood is for a given parameter
- **Pre/post-fit reconstructed distributions and the chi-square contribution per sample**
  - Can be used to see how well the post-fit agrees with the (mock) data
  - Can be used to calculate a p-value for the (mock) data fit
- **Error/log output from the fit routine e.g. MINUIT output**
  - Will indicate errors with the minimization, such as if the covariance was forced positive definite or if the fit failed to converge

P-Value Tests

We would like to be sure that:

1. The post-fit model is compatible with the data
2. The post-fit model is plausible, given the input model's prior uncertainties

Example from the CC1πNp analysis



- The post-fit model seems to describe the data and the chi-square is clearly reduced.
- But can we be more quantitative? (Spoiler: yes)

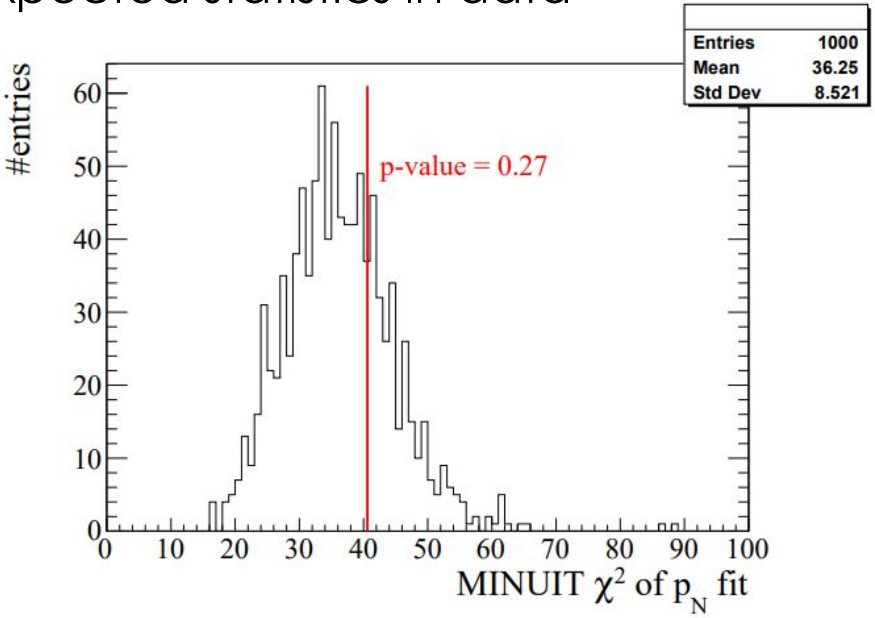
P-Value Tests

We would like to be sure that:

1. The post-fit model is compatible with the data
2. The post-fit model is plausible, given the input model's prior uncertainties

Calculate a p-value:

1. Produce an ensemble of systematic variations of your input model (given your prior uncertainties) and take a further random poisson variation of each according to the expected statistics in data
2. Fit each throw (or toy, or universe) in the ensemble
3. Compare the data best-fit chi-square with the distribution from the ensemble of throws
4. Ratio of throws which exceed the data chi-square is the p-value.



P-Value Tests

We would like to be sure that:

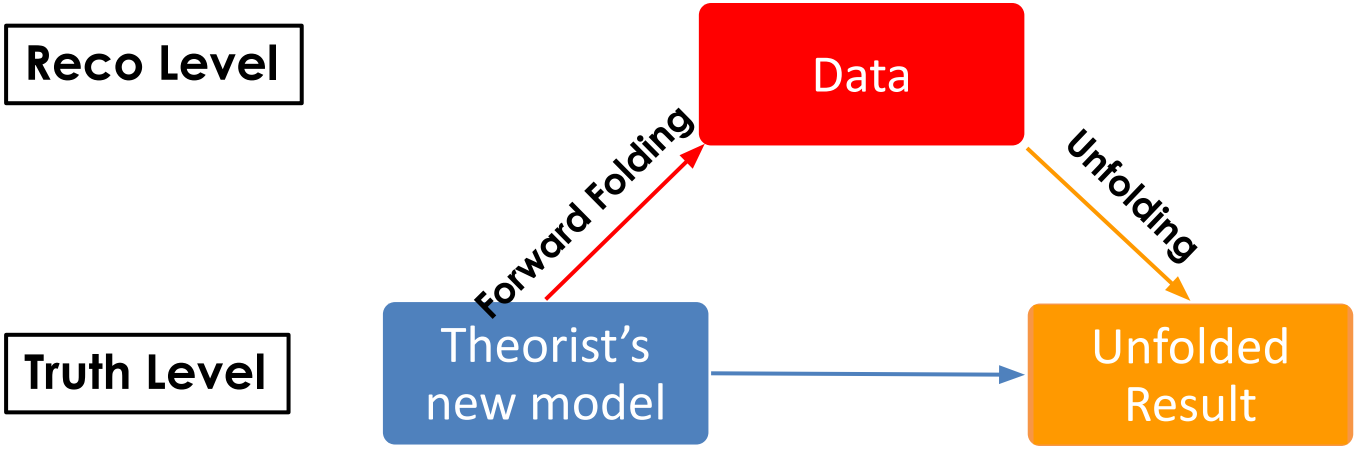
1. The post-fit model is compatible with the data
2. The post-fit model is plausible, given the input model's prior uncertainties

- Can look at the spread of only the statistical contribution to the chi-square (addressing the first point above)
- Can repeat the procedure considering a single fit sample, or a limited set of fit samples, to get a sample specific p-value*
- Can also calculate a p-value for individual parameter sets* (e.g. flux) and see how the distribution compares to the total systematic contribution.
- Investigating single parameters equivalent to a "pulls" study.

* But beware of the "look-elsewhere" effect!

Unfolding is still unfolding …

- The fitter is a good tool for unfolding, offering diagnostics to avoid many of its typical pathologies
- But even perfectly signal-model independent unfolding is not as sensitive as a comprehensive forward folding analysis -- c.f. arXiv:1607.07038



doi.org/10.1088/1748-0221/14/09/P09013

Summary

Template fitting is a conceptually straightforward unfolding and cross section extraction method with a number of benefits:

- Similarities with neutrino oscillation fits
- Simple method of constraining backgrounds
- Regularisation: none built-in but data-driven schemes can be added
- Lots of diagnostics available to check you can trust your result!

The framework for template fitting is adaptable and expandable

- *Want to avoid Gaussian-errors approximations in the fit result?* Swap the minimiser to a MCMC and use the steps.
- *Need to avoid PPP?* Publish the toys used to build the cross-section as well as the covariance.
- *Want to tune a model (e.g. for an oscillation analysis) instead of extracting a cross-section?* Just turn off the template parameters!
- *Want to turn a cross section analysis into a (simplistic) search for SBL oscillations?* Replace the template parameters with oscillation parameters.
    - *Can be used for BSM searches in general.*

PPP = Peelle's Pertinent Puzzle

Contact details

Andrew Cudd - andrew.cudd@colorado.edu

Stephen Dolan - stephen.joseph.dolan@cern.ch

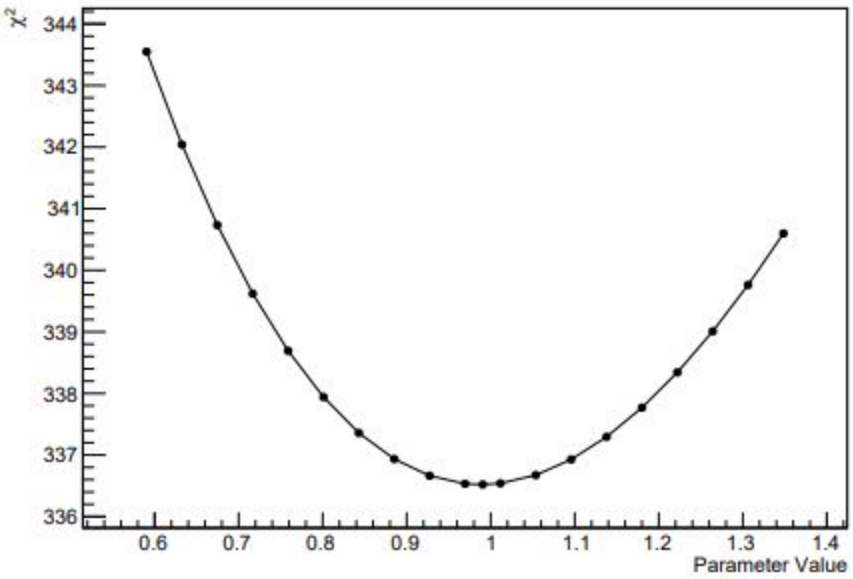
Ciro Riccio - ciro.riccio@stonybrook.edu

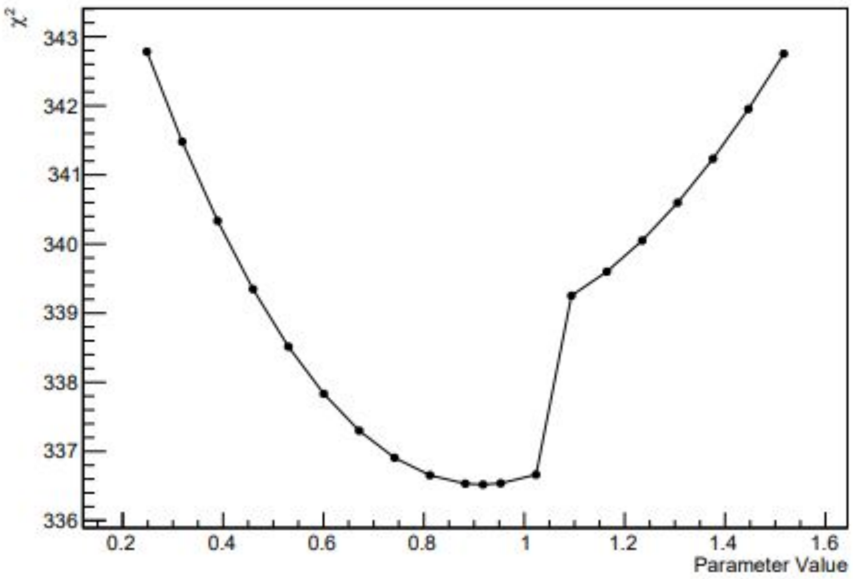
Ka Ming Tsui - k.m.tsui@liverpool.ac.uk

Backups

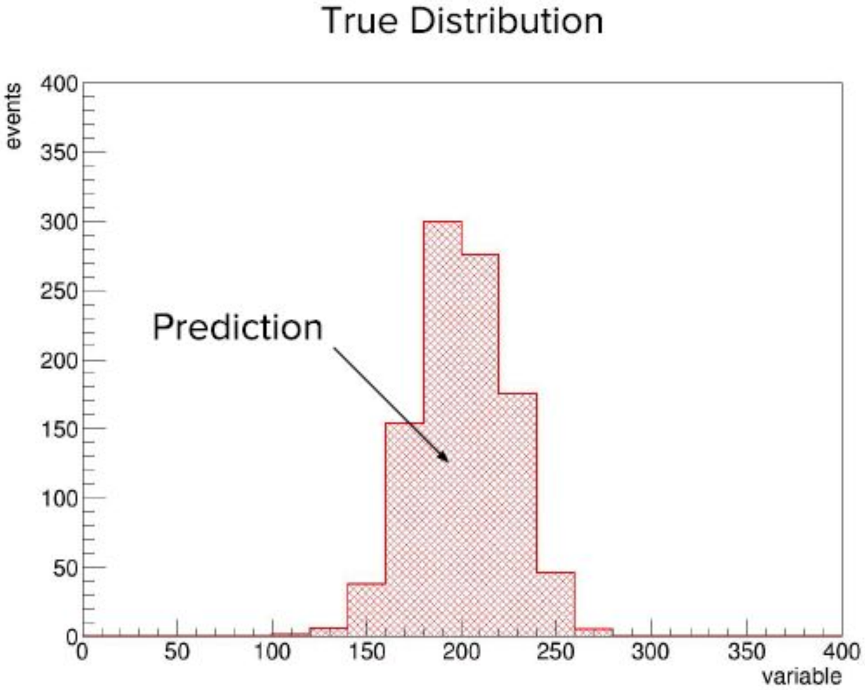
Likelihood Scans

- In addition to p-value tests we can consider likelihood scans to check our fit parameters have Gaussian(ish) responses close to the best point (as we assume in order to get our post-fit covariance)
- If we find non Gaussian responses the offending parameters need to be carefully studied (e.g. analysing whether they actually have much impact on the fit and why they have a non Gaussian response)
- *Beware: a parameter with a Gaussian response does not necessarily cause Gaussian variations on the cross section (see warning on slide 26!)*

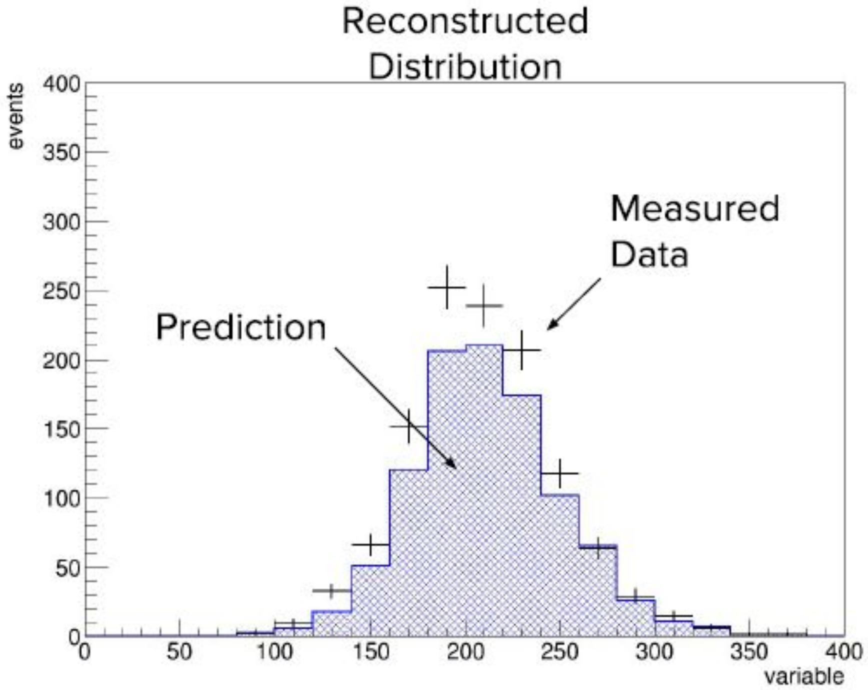




Template fitting


True Distribution

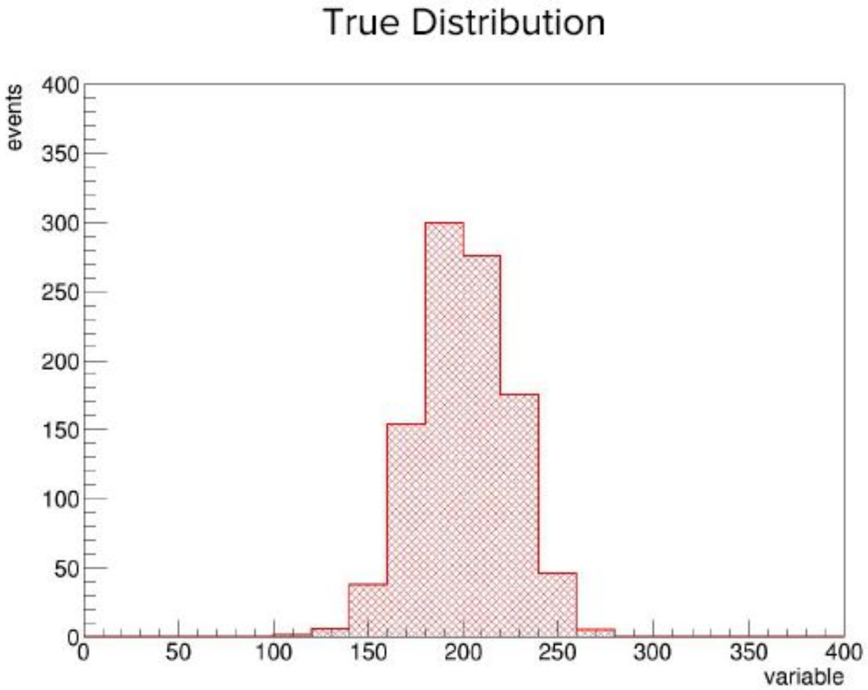
Prediction

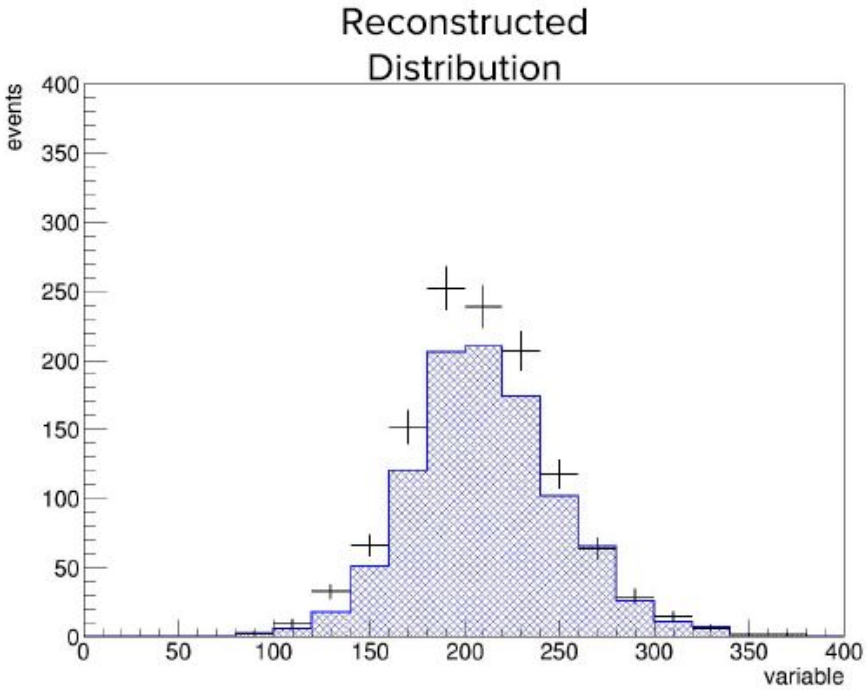

Reconstructed Distribution

Measured Data

Prediction

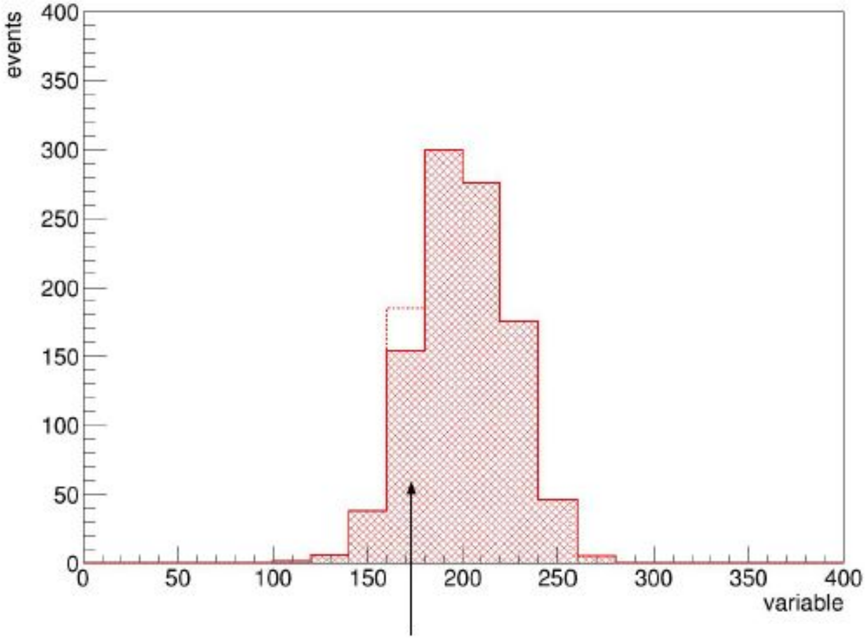
Template fitting





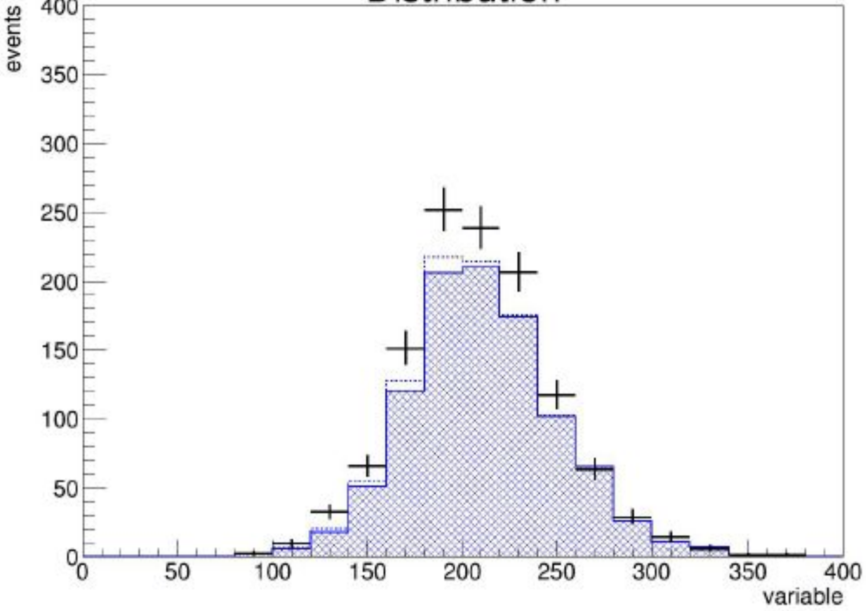
Template fitting

True Distribution



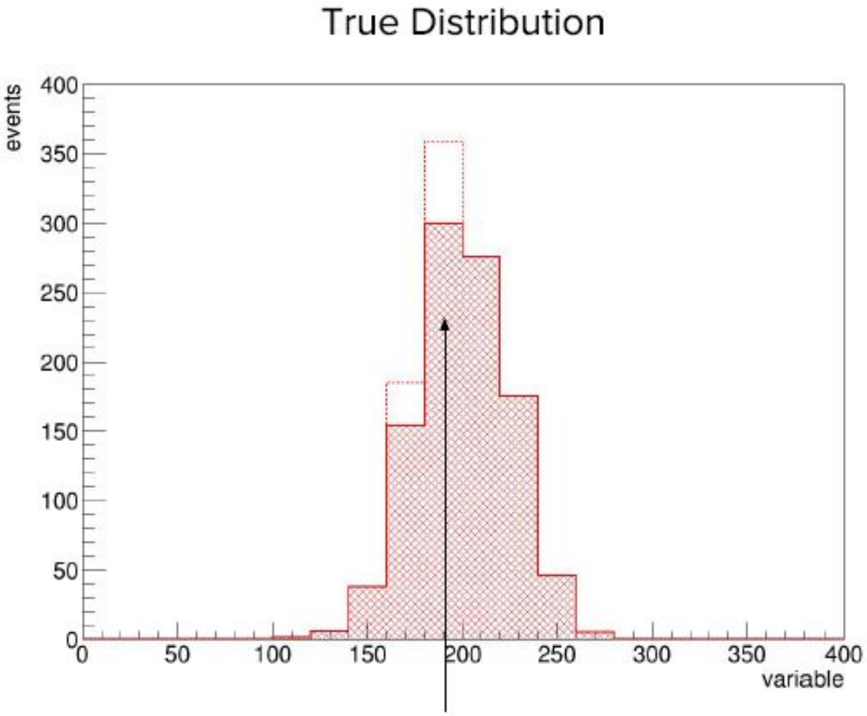
Change template weight for bin

Reconstructed Distribution

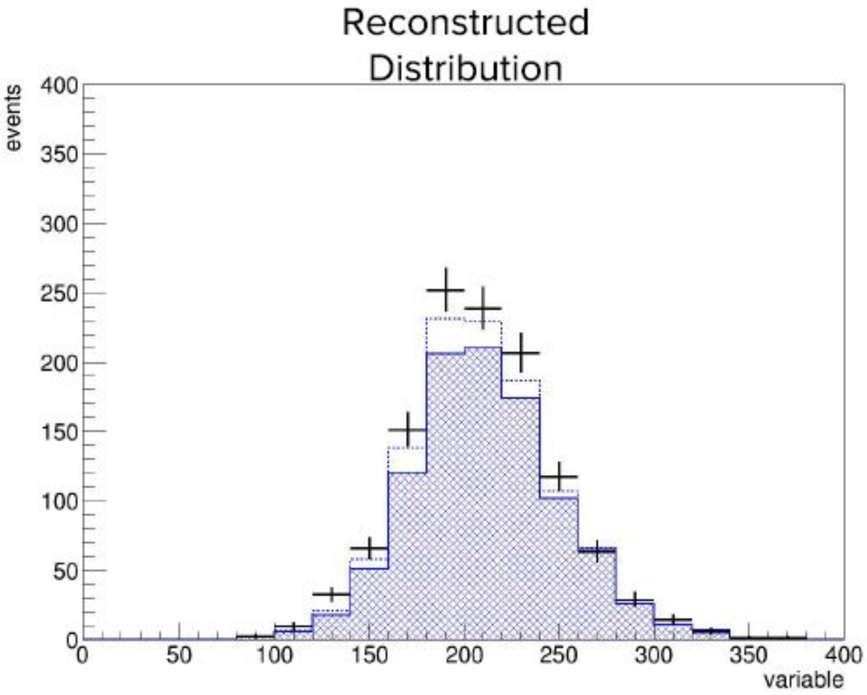


Observe changes in reco. space.

Template fitting

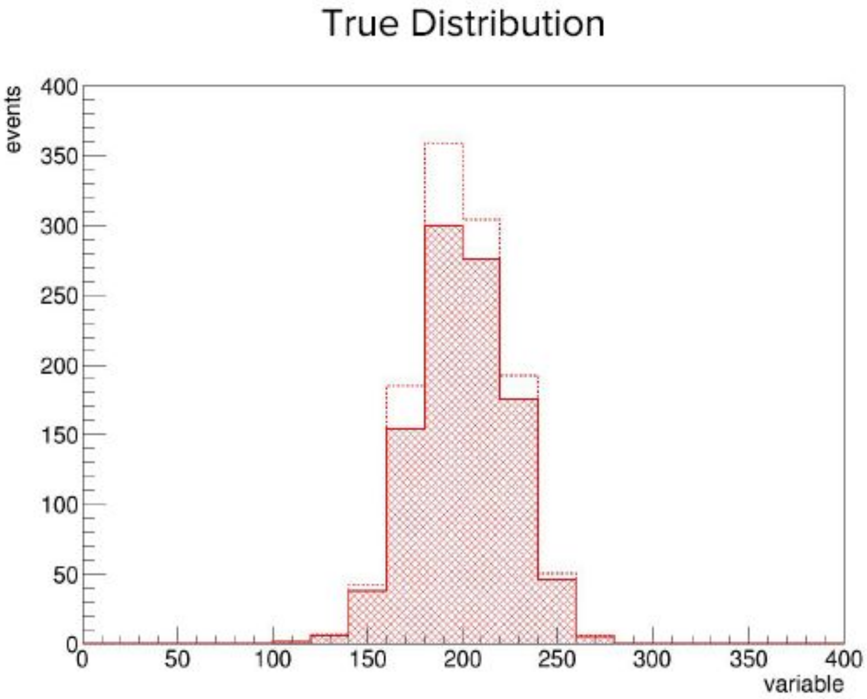


Change template weight for bin

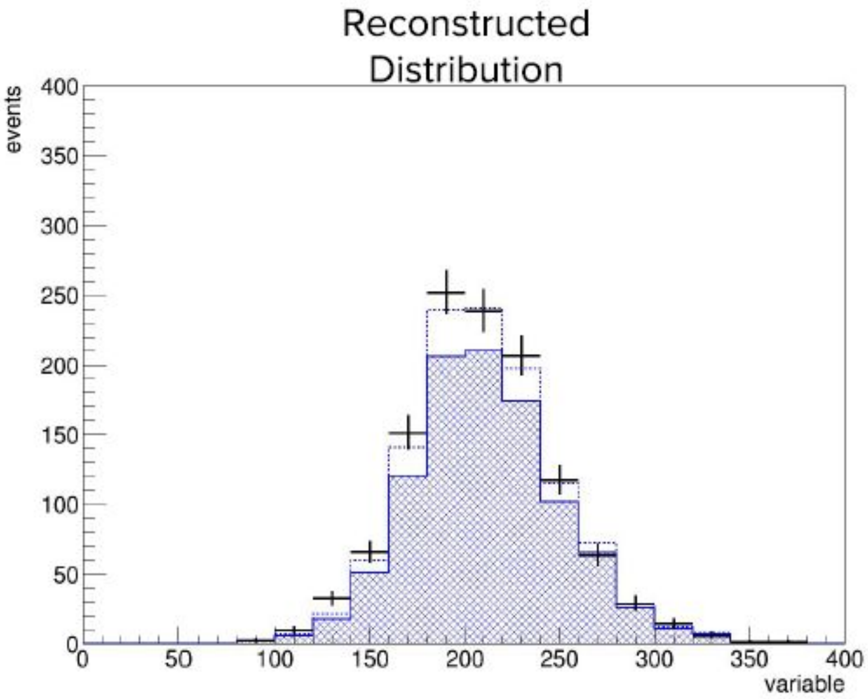


Observe changes in reco. space.

Template fitting



Increase total flux weight by 10%.



Observe changes in reco. space.

Applying the fit parameters

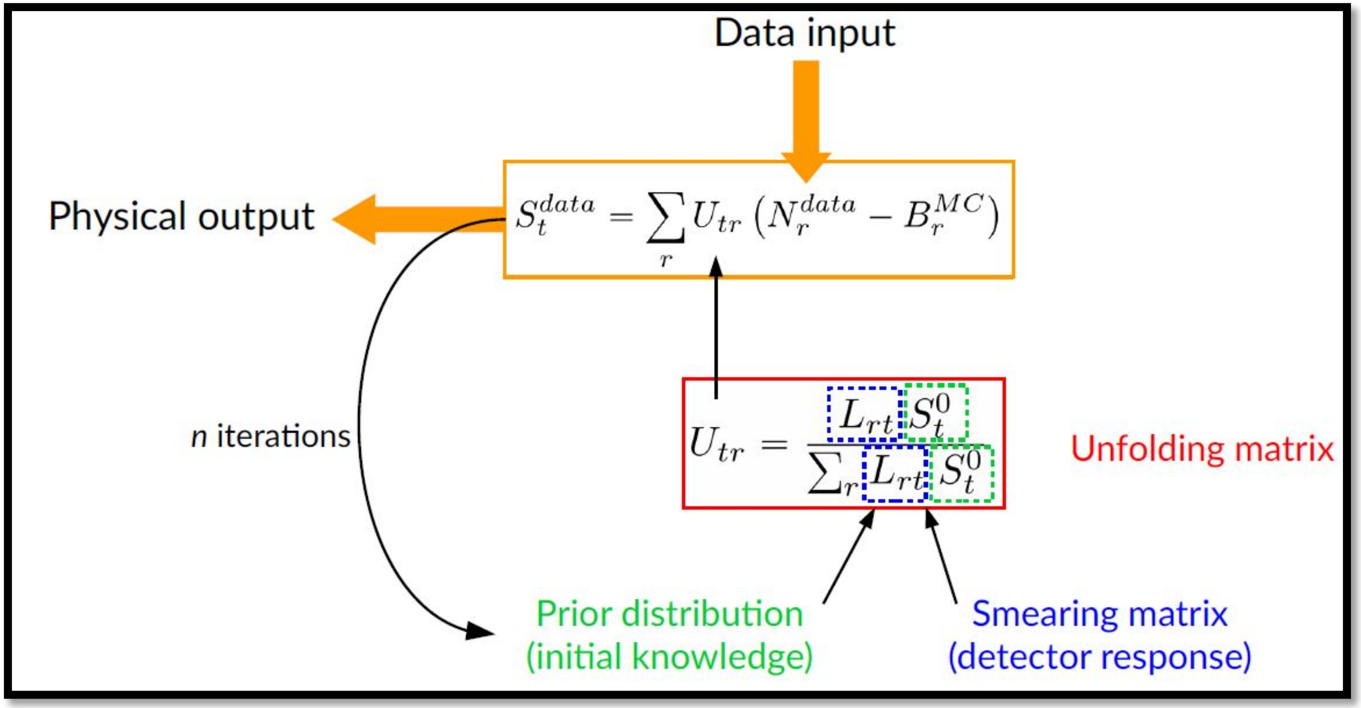
$$N_{ij}^{\mathrm{MC}} = \underbrace{d_{ij}}_{\text{Detector parameters}} \cdot \sum_{k}^{\substack{\nu\text{ energy bins,}\\ \text{right/wrong sign}}} \underbrace{f_k}_{\text{Flux parameters}} \cdot \begin{cases} \displaystyle\sum_{l}^{\text{truth bins}} \overbrace{c_l}^{\text{Template parameters}} \cdot \sum_{m}^{\substack{\text{Events within}\\ \text{sample } i,\\ \text{reco bin } j,\\ \text{energy bin } k,\\ \text{truth bin } l}} \overbrace{\sigma_m}^{\text{Cross section parameters}} \cdot \overbrace{w_m^{HL}}^{\text{Event weights}} & \text{if signal} \\ \displaystyle\sum_{m}^{\substack{\text{Events within}\\ \text{sample } i,\\ \text{reco bin } j,\\ \text{energy bin } k,}} \sigma_m \cdot w_m^{HL} & \text{if background} \end{cases}$$

Recalculated for every iteration of the fitter.

D'Agostini's method

*Although this method uses Bayes' theorem, ***it is not a Bayesian technique*** (in fact it's equivalent to the widely-used "Expectation-maximisation algorithm") [M.Kuusela]*

- Using Bayes' theorem* to form unsmearing matrix:



Data input

$$S_t^{data} = \sum_r U_{tr} \left(N_r^{data} - B_r^{MC}\right)$$

Physical output

n iterations

$$U_{tr} = \frac{L_{rt} S_t^0}{\sum_r L_{rt} S_t^0}$$

Unfolding matrix

Prior distribution (initial knowledge)

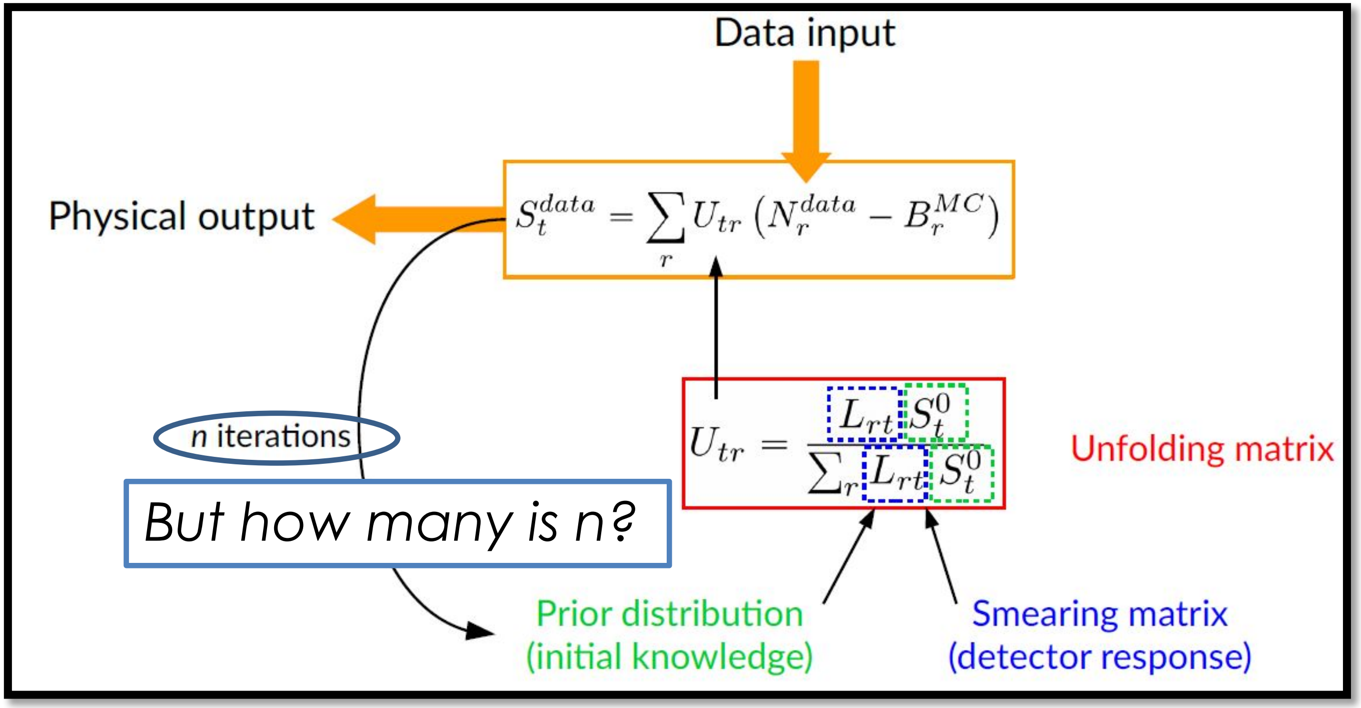
Smearing matrix (detector response)

- Most commonly used method (MINERvA, MiniBooNE, T2K)
- If prior formed from MC - **model dependence is explicit**
- Mitigate by updating prior with unfolded result and iterating
- **Many iterations** (typically hundreds / thousands) → **unregularised result**

D'Agostini's method

Although this method uses Bayes' theorem, **it is not a Bayesian technique (in fact it's equivalent to the widely-used "Expectation-maximisation algorithm") [M.Kuusela]*

- Using Bayes' theorem* to form unsmearing matrix:



Data input

$$S_t^{data} = \sum_r U_{tr}\left(N_r^{data} - B_r^{MC}\right)$$

Physical output

$$U_{tr} = \frac{L_{rt} S_t^0}{\sum_r L_{rt} S_t^0}$$

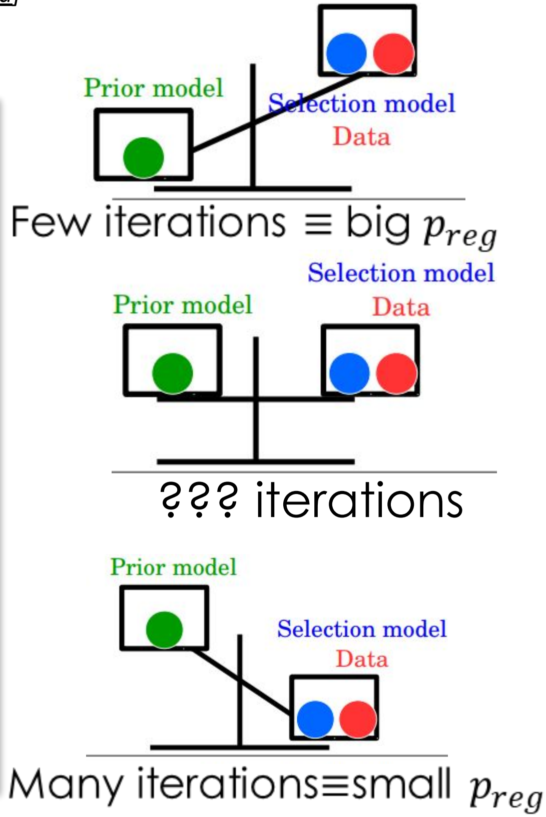
Unfolding matrix

n iterations

But how many is n?

Prior distribution (initial knowledge)

Smearing matrix (detector response)



Few iterations ≡ big p_{reg}

??? iterations

Many iterations≡small p_{reg}

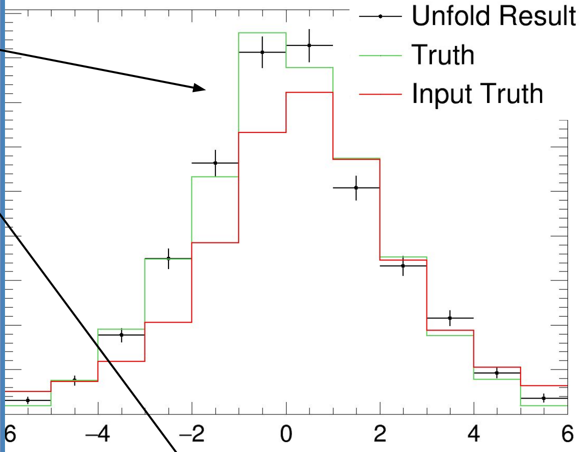
- **Changing the number of iterations can change physics conclusions**
- Typically select number of iterations **based on mock-data studies**
- If real data looks different, can select "wrong" number

How many iterations? – Choose via MC

Gaus (0.1,1.0) smear,1.0 bin width, 2000 events, Truth is a Gaus(0,2.0), Input is a BW(0.3,2.5)

- Too few iterations can give a result which *looks* **okay but is actually biased to the shape of the input**
- Adjacent bins are correlated, even though we binned close to our detector resolution
- Early termination of D'Agostini can give unrealistically small errors

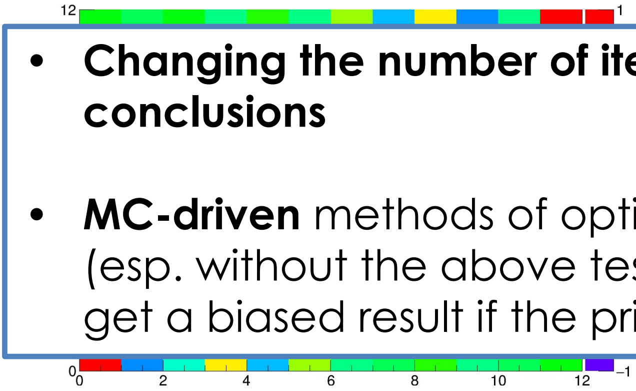
First test for this: **Check that the χ^2 preference in model comparisons is similar to the unregularised results**

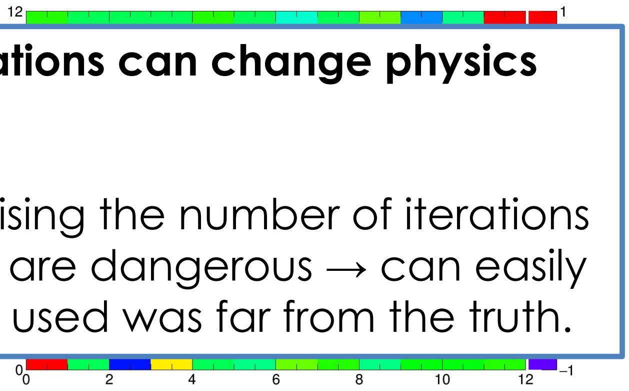


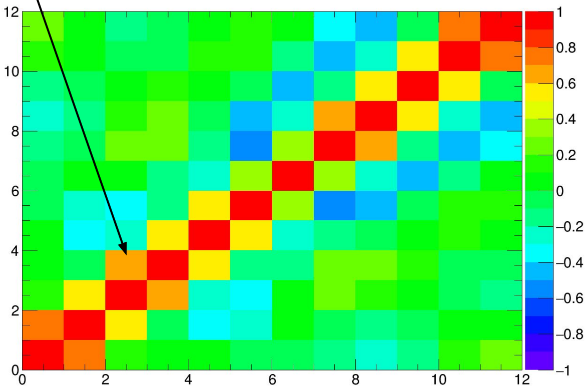
2000 iterations (~unreg)
$\chi^2_{truth} = 2.9, \chi^2_{input} = 134$

50 iterations
$\chi^2_{truth} = 32, \chi^2_{input} = 254$

4 iterations
$\chi^2_{truth} = 9606, \chi^2_{input} = 1568$

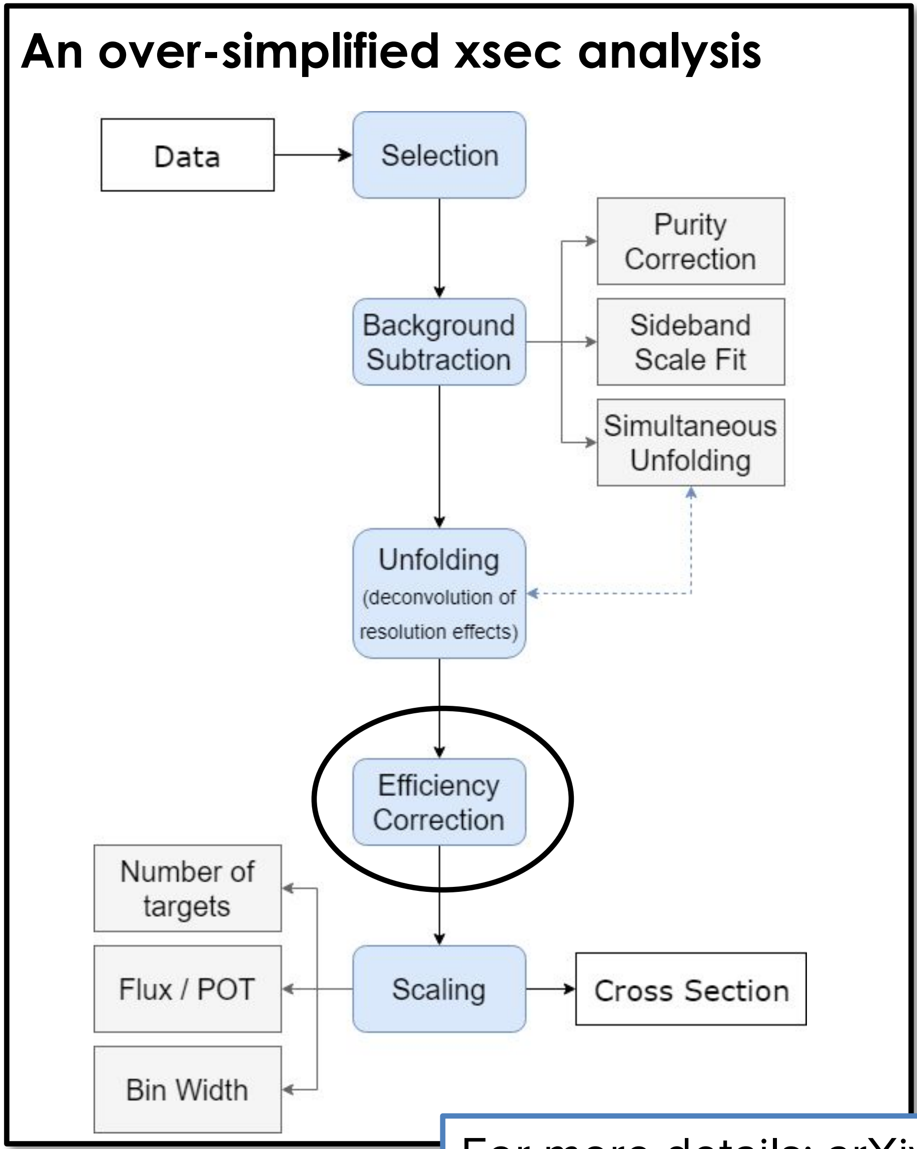






- **Changing the number of iterations can change physics conclusions**
- **MC-driven** methods of optimising the number of iterations (esp. without the above test) are dangerous → can easily get a biased result if the prior used was far from the truth.

Efficiency corrections

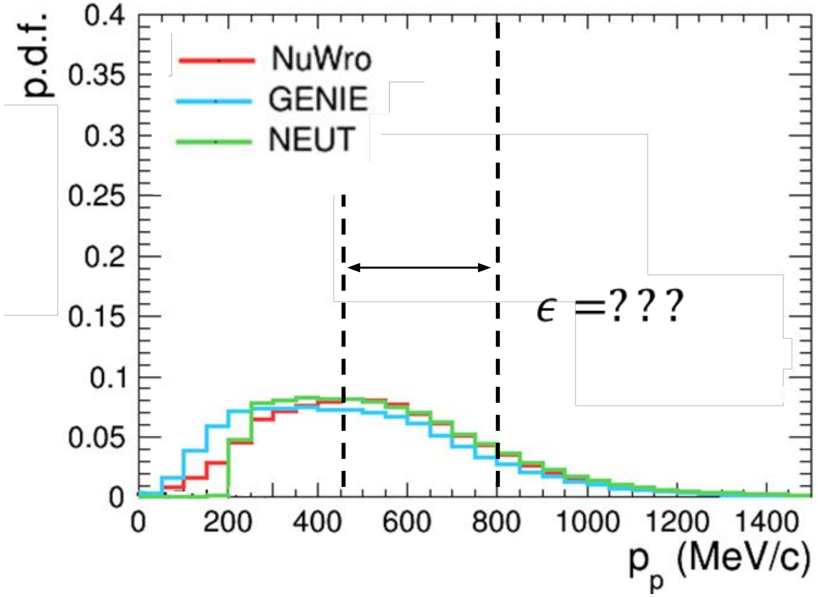


- After unfolding we have the a measure of the true number of *selected* signal events
- To get to a cross section, we need to correct for our detectors acceptance
- It's also easy to add bias here …
- Not entirely separate from unfolding
  - Unfolding in too few variables can give bias here

For more details: arXiv 1805.07378 (TENSIONS Workshop 2016)

Toy example

- I want to measure a cross section in some range of proton momentum

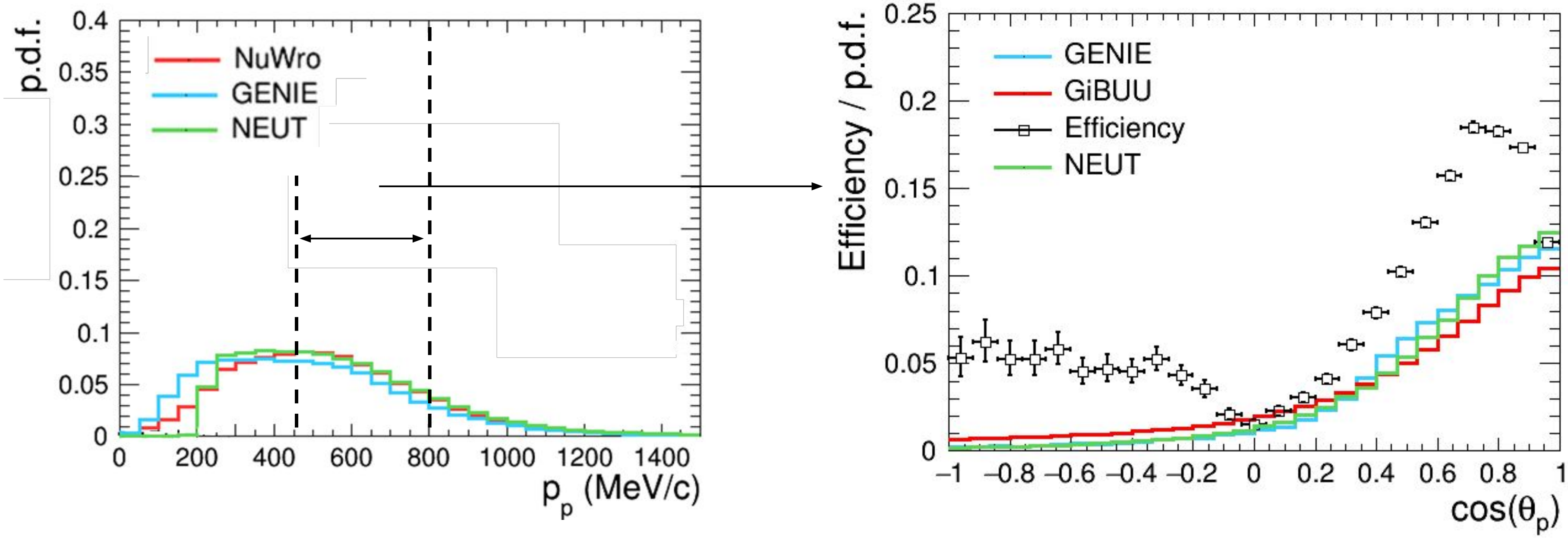


- But my detection efficiency depends on both proton momentum and angle (and on other particles, but let's focus on the angle for the moment!)

- I can't know the efficiency (ϵ) without knowing the distribution of proton angle within the bin

Toy example

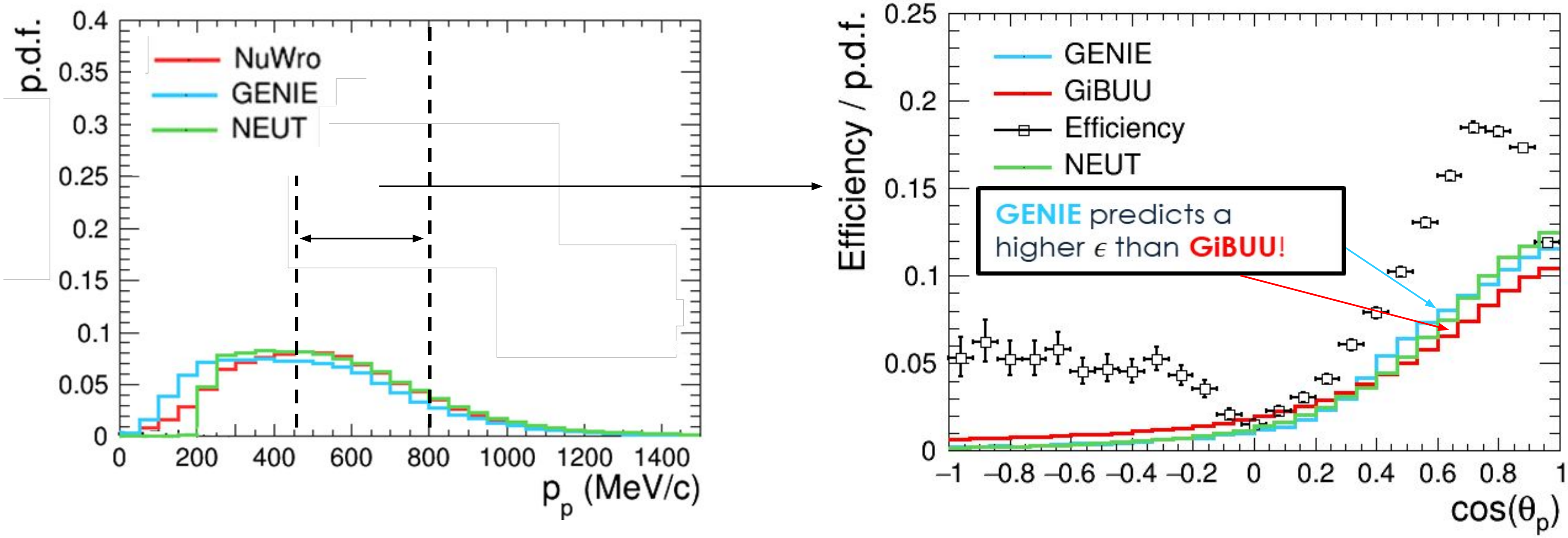
- I want to measure a cross section in some range of proton momentum



- The efficiency in the momentum bin a convolution of the efficiency **and the predicted cross section**

Toy example

- I want to measure a cross section in some range of proton momentum

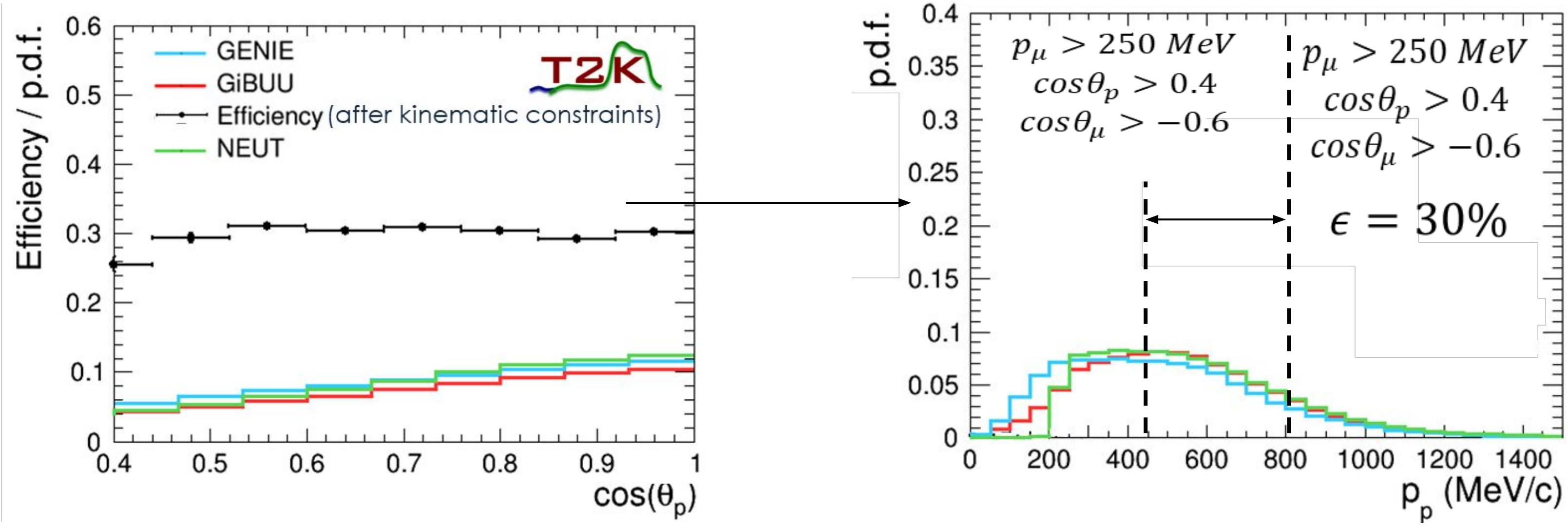


GENIE predicts a higher ϵ than GiBUU!

- The efficiency in the momentum bin a convolution of the efficiency **and the predicted cross section**
- Compared to GiBUU, GENIE predicts a higher cross section ***in the high efficiency region*** → GENIE predicts a higher (~5-10%) efficiency
- **Efficiency depends on the input model** → Bias

Kinematic constraints

- Placing kinematic constraints on outgoing particles $(p_{\mu,p}, \theta_{\mu,p})$ can leave us with a relatively flat efficiency in a specific region of $\cos\theta_p$



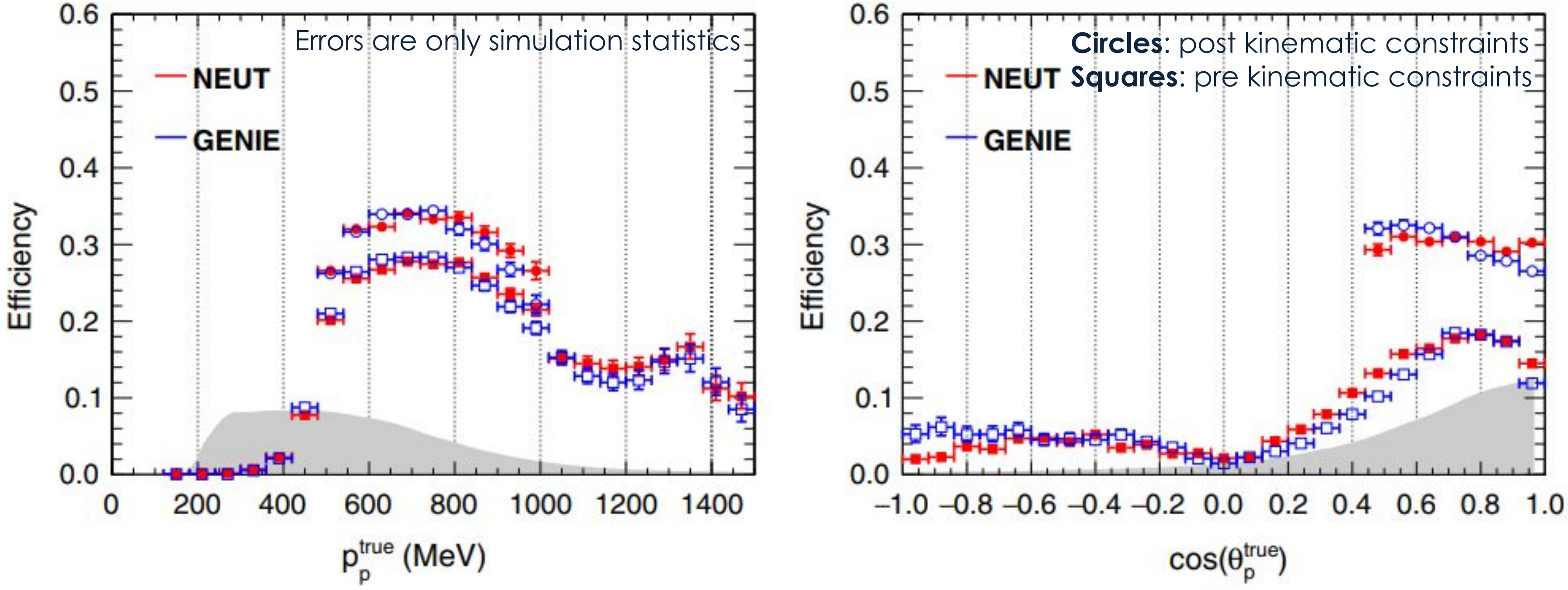
- In this case the shape of the input model doesn't alter the efficiency → **model independent correction**!

- Our analyses try to ensure integration only over flat-efficiencies in observables where simulations have poor predictive power

Efficiency correction example

Phys. Rev. D **98**, 032003

- Measuring δp_T relies on integrating the efficiency over $p_{\mu,p}, \theta_{\mu,p}$
- We set kinematic constraints in each to keep efficiency relatively flat, especially in regions of phase space where models have low predictive power (proton kinematics)
- Still not perfect, **ideally should efficiency correct in all relevant kinematics**

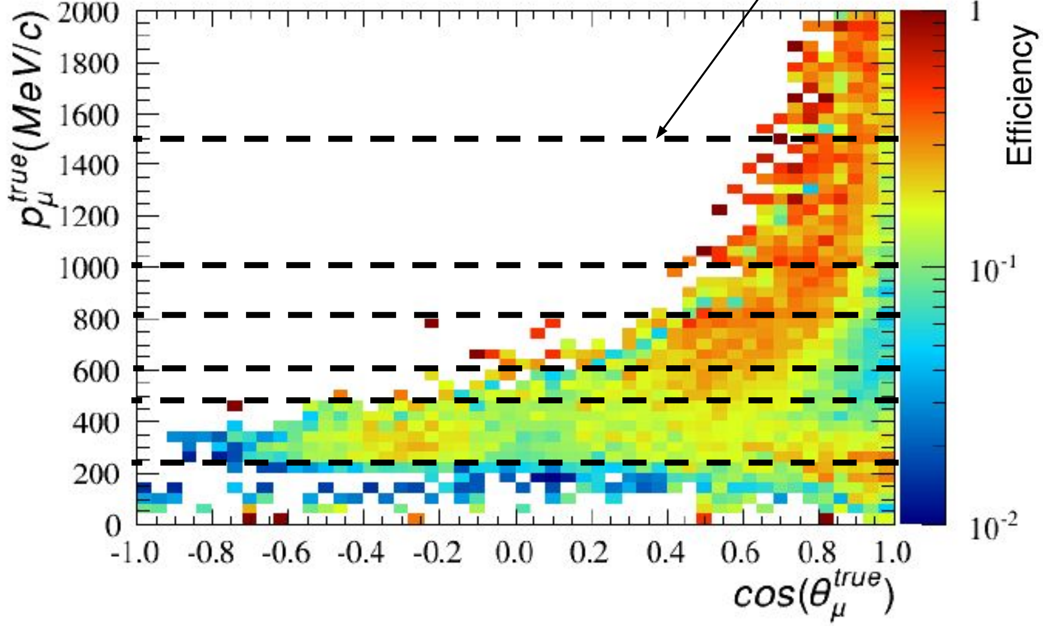


Efficiency Corrections

Want to measure single-differential p_μ

- Consider 2D efficiency

1D desired p_μ binning

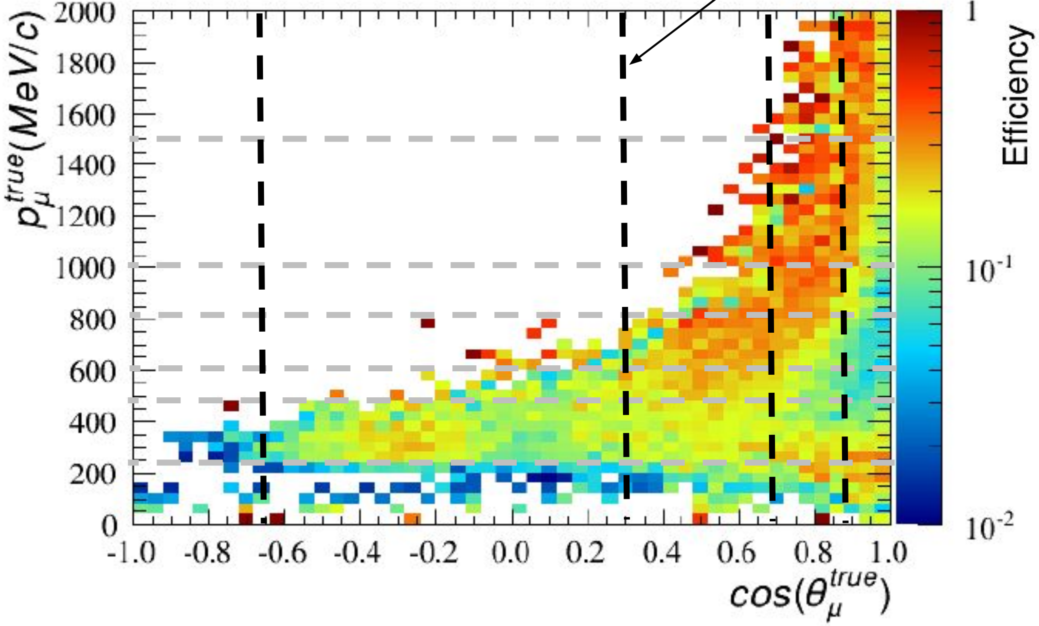


Efficiency Corrections

Want to measure single-differential p_μ

- Consider 2D efficiency
- Bin just fine enough in $\cos(\theta_\mu)$ such that ϵ~flat

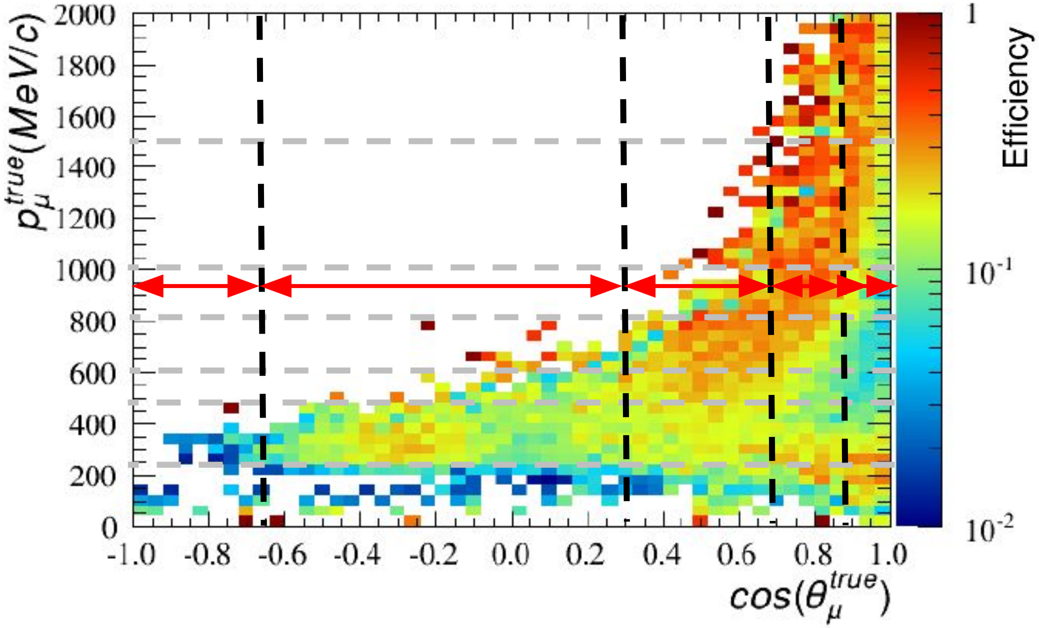
Coarse $\cos(\theta_\mu)$ binning



Efficiency Corrections

Want to measure single-differential p_μ

- Consider 2D efficiency
- Bin just fine enough in $\cos(\theta_\mu)$ such that ϵ~flat
- Extract 2D cross-section
- Marginalise over $\cos(\theta_\mu)$ bins


p_μ^{true}(MeV/c)
$\cos(\theta_\mu^{true})$
10^{-1}
10^{-2}
Efficiency