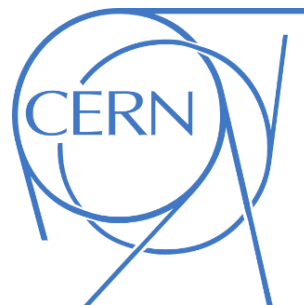# Unfolding and Regularisation
Experiences from neutrino cross-section measurements

# *Stephen Dolan*

stephen.joseph.dolan@cern.ch

Overview

- This talk is meant to be an overview of some experiences with unfolding and regularisation for neutrino cross-section measurements

- This is heavily based on talks I gave at NuInt and PhyStat-nu

- It is very likely that the difference in the expected statistics in ProtoDUNE measurements means that not all the examples are directly applicable

- However, hopefully some of the general considerations and concerns regarding unfolding are transferable

Unfolding

An over-simplified cross-section analysis
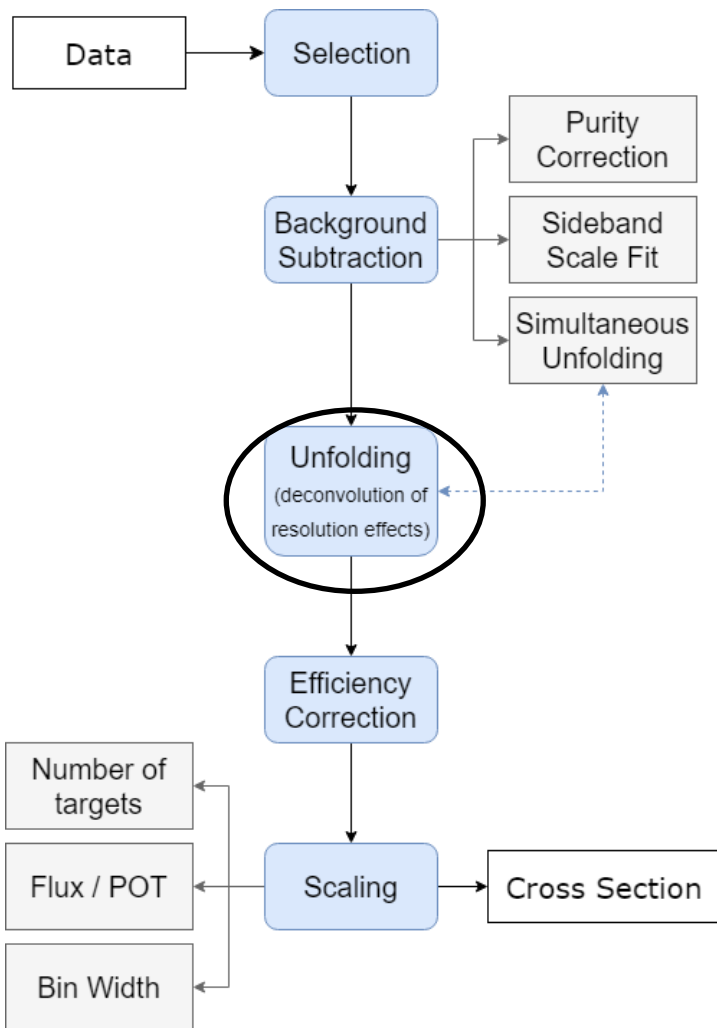
Smearing

Number of events in reco bin j (what we measure)

Number of events in true bin i (no detector smearing)

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

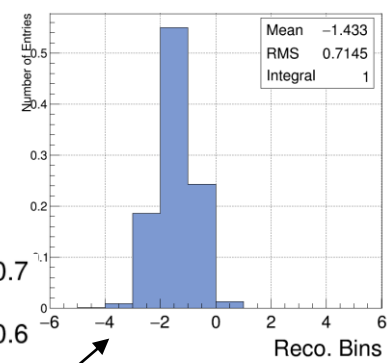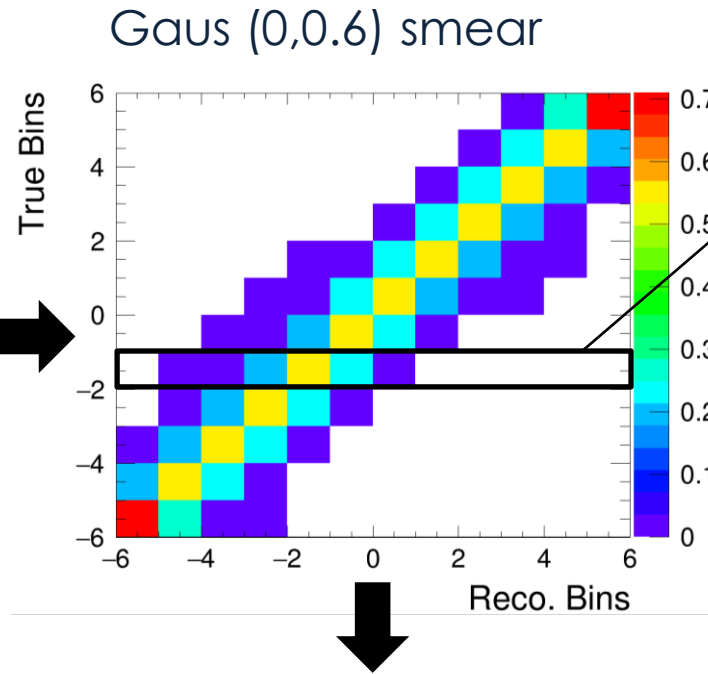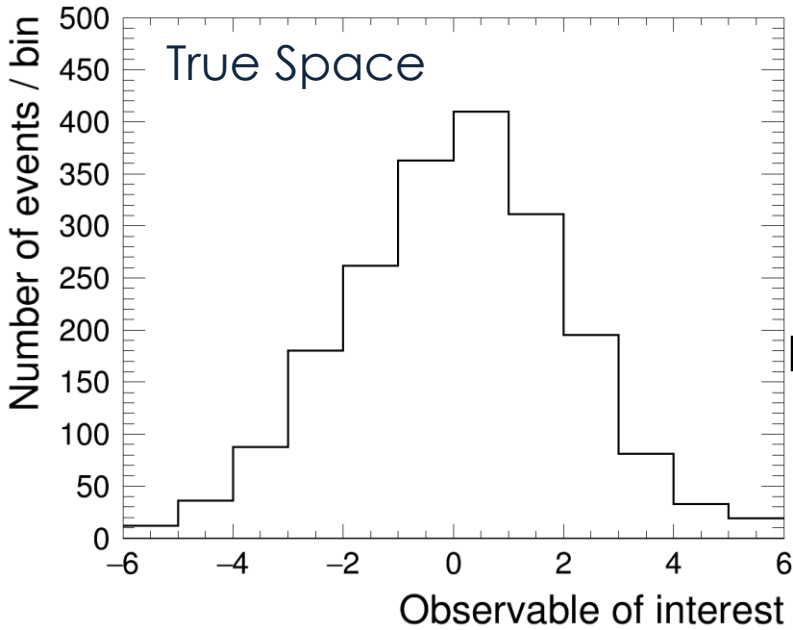Smearing matrix

Unfolding

Number of events in true bin i

Number of events in reco bin j

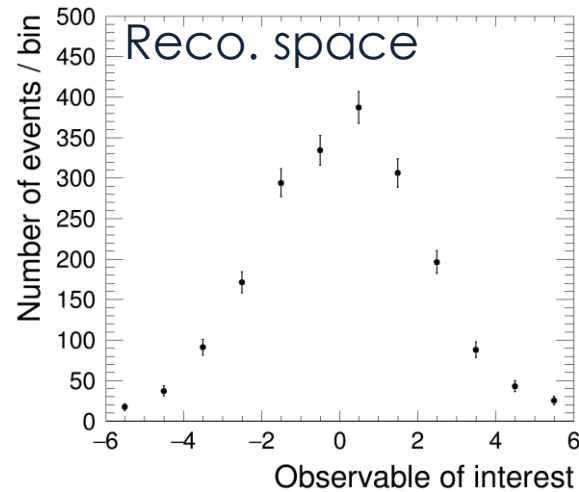$$T_i = \sum_{Reco\ Bins,j} U_{ij} R_j$$

Unsmearing matrix

- **The goal**: find U_{ij} from S_{ji}

- **Simplest method**: use S_{ji}^{-1}
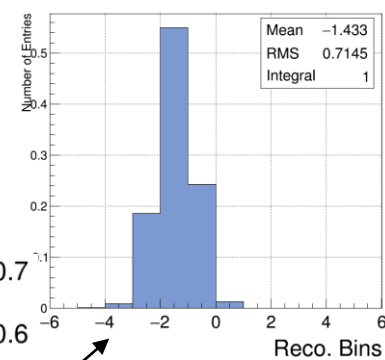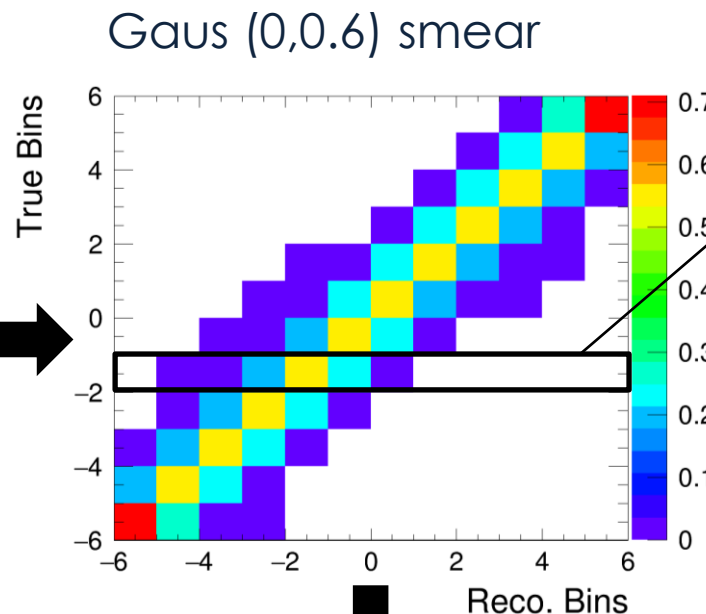 (unregularised result)

Toy example - smearing

True Space

Gaus (0,0.6) smear

Reco. space
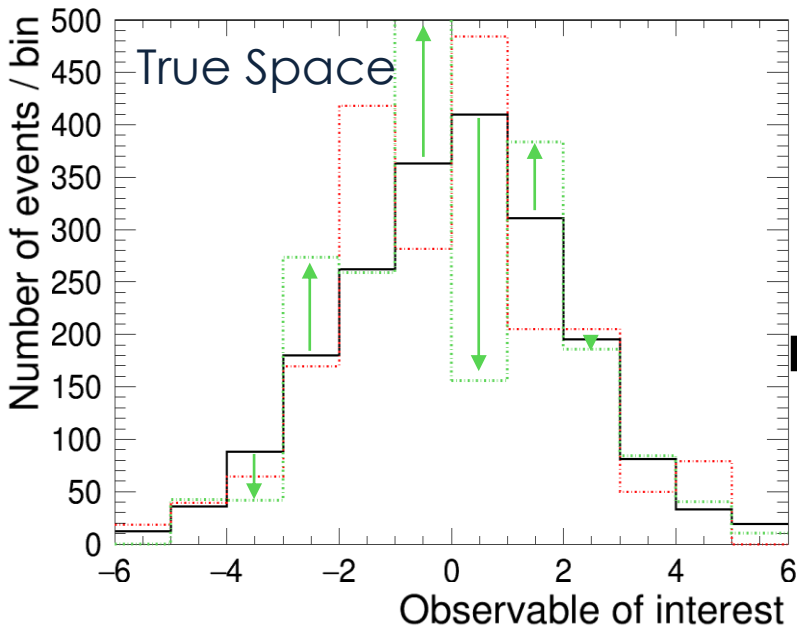
- 2000 events with bins of width 1.0 with a "resolution" of 0.6

- Quasi-realistic example for T2K neutrino scattering

Toy example - smearing
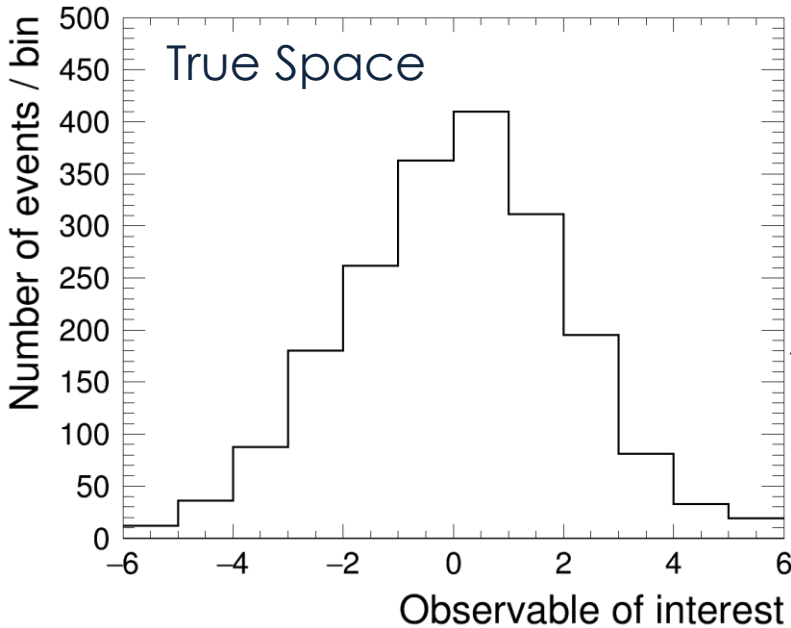

True Space

Gaus (0,0.6) smear

- Often free to move one bin down so long as we move the adjacent bins up to compensate*: see small changes in the reco. space

- True even with large variations

- Doesn't bode well for solving the inverse problem …
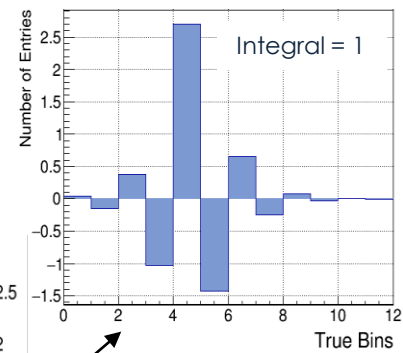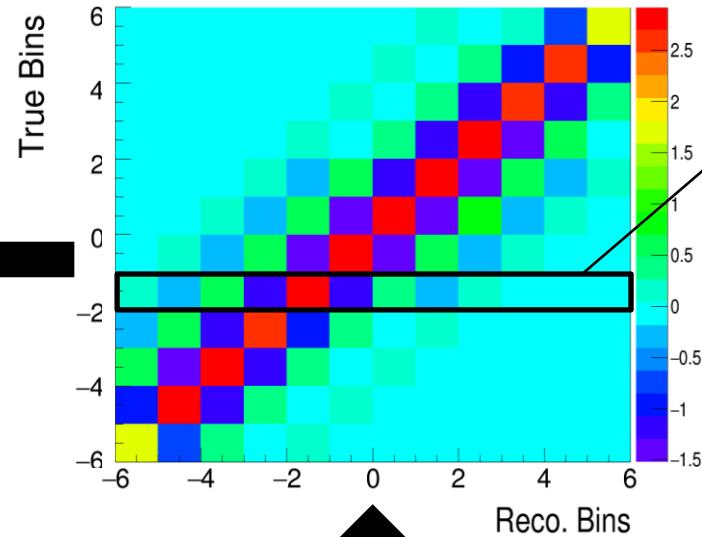
Reco. space
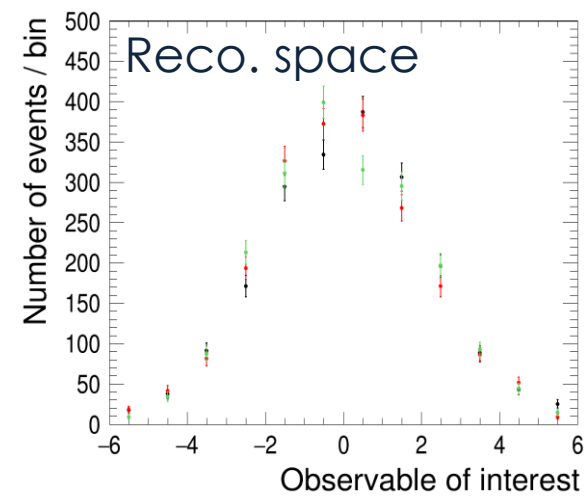
*This is intuitive: If you remove some events from a bin you can repopulate them with the "spill over" from the adjacent bins

Toy example - unfolding

True Space

Inverse (Gaus (0,0.6) smear)

Reco. space

- So the inverse looks fine provided that:

  - Response in MC is exactly the same as the real response

  - The reco. MC sim. is identical to the real data ...

Toy example - unfolding

True Space

Inverse (Gaus (0,0.6) smear)

Integral = 1
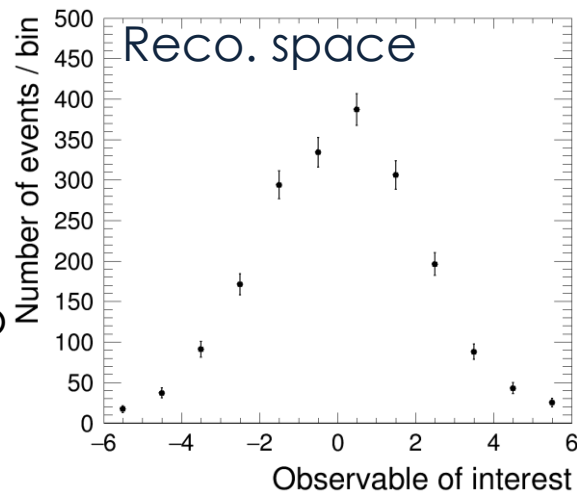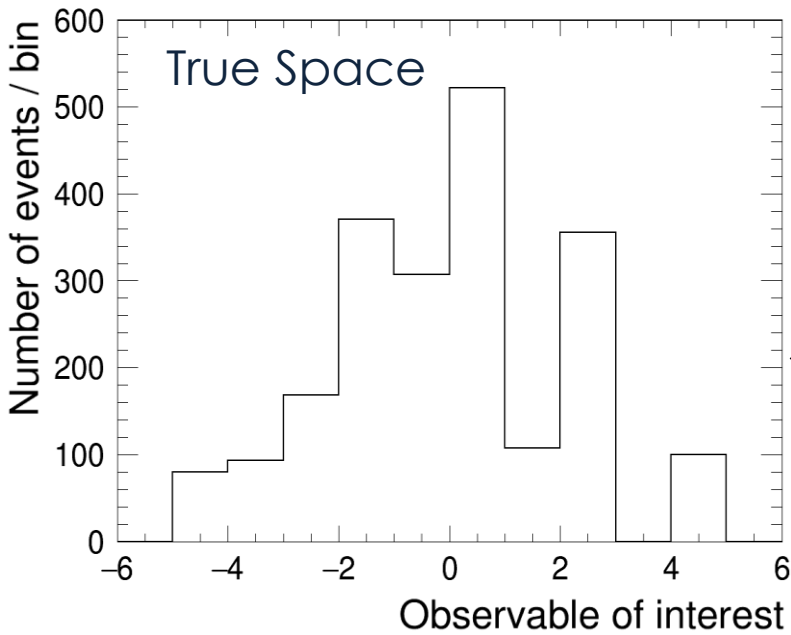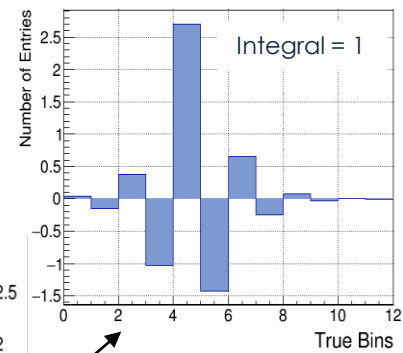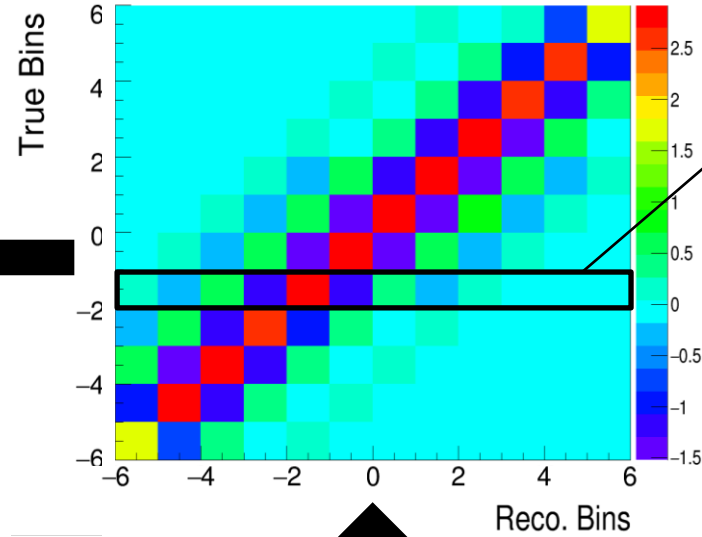
- Let's try a Poisson fluctuation of the reco. space!

- HUGE variation in the true space

Reco. space

Toy example - unfolding

True Space

Inverse (Gaus (0,0.6) smear)

Reco. space
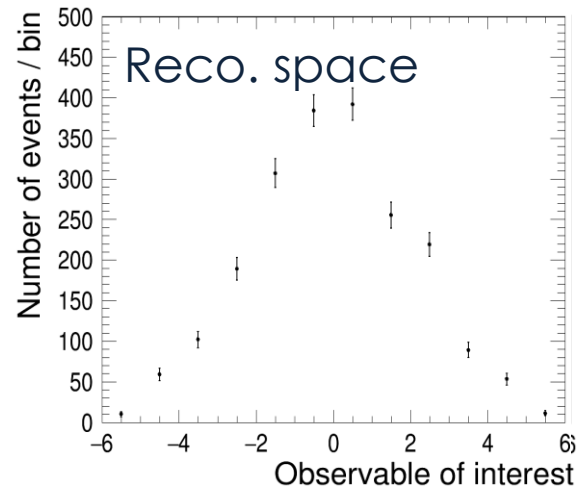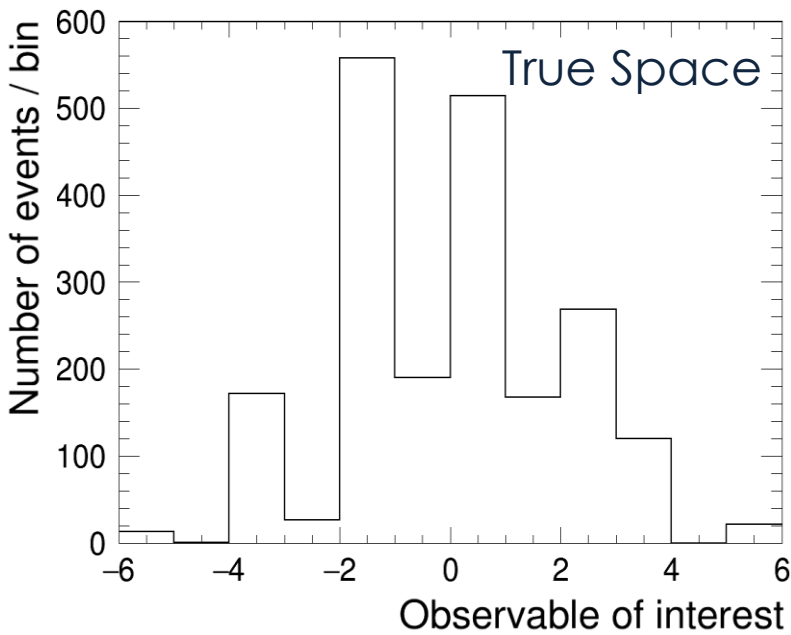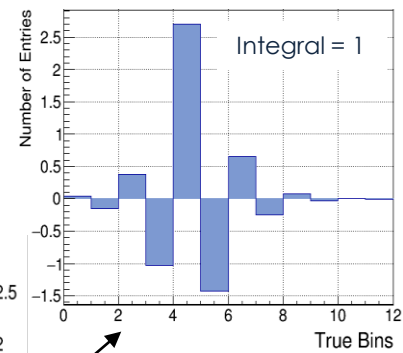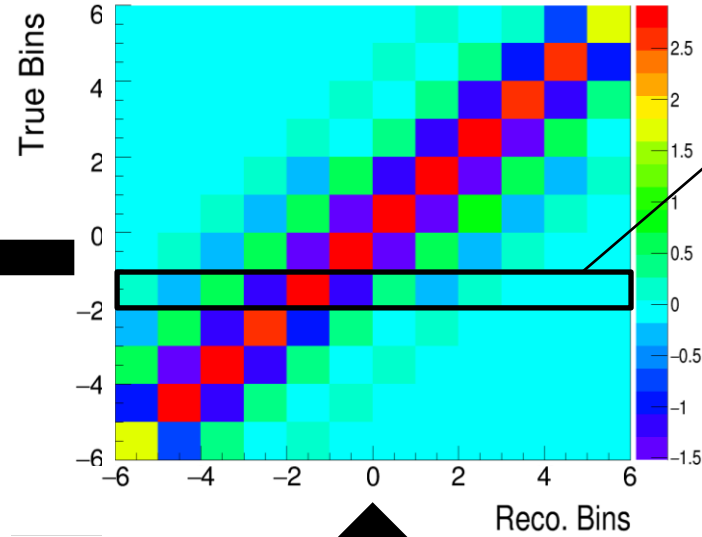
- Let's try another fluctuation

- Same thing …

Toy example - unfolding

True Space

Inverse (Gaus (0,0.6) smear)

Integral = 1

Reco. space
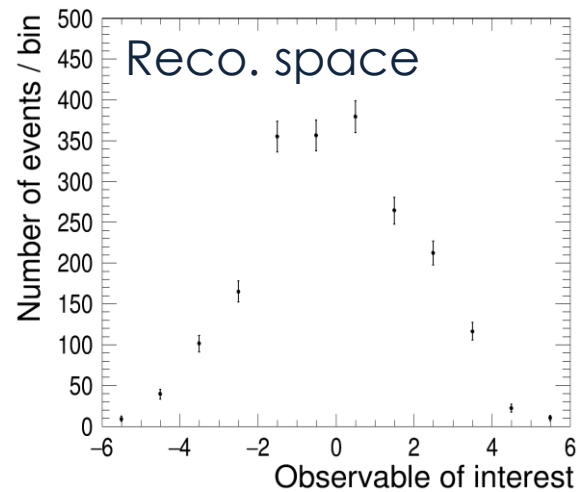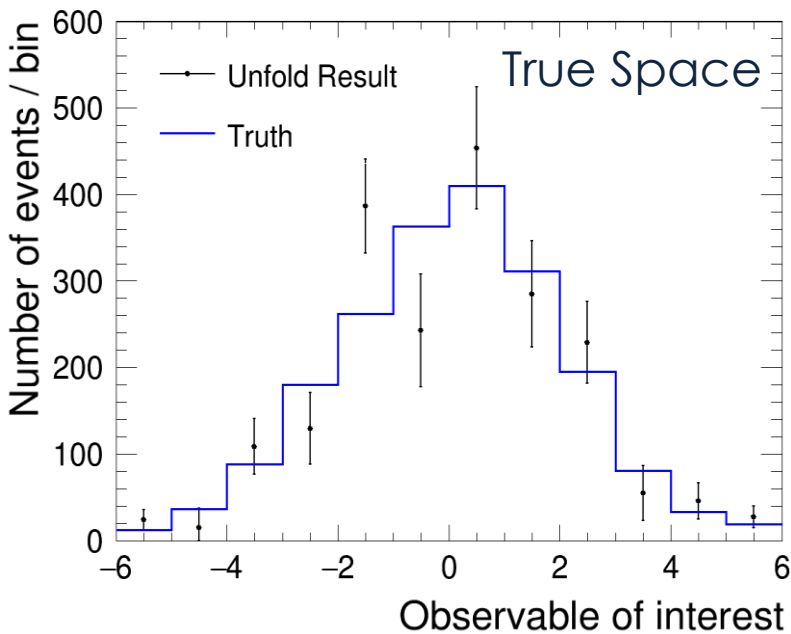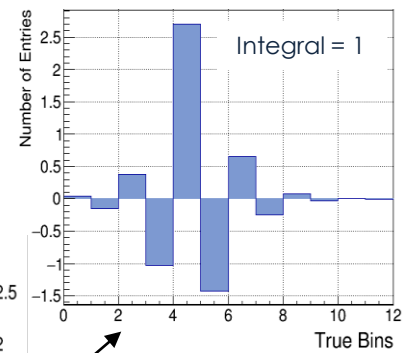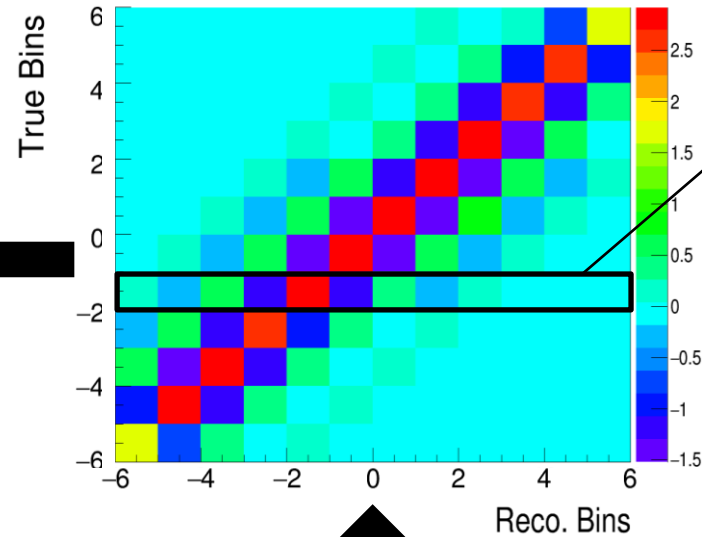
- Many fluctuations allow us to build a result with errors

- In most realistic circumstances we get a result which oscillates around the truth

- It doesn't *look* great
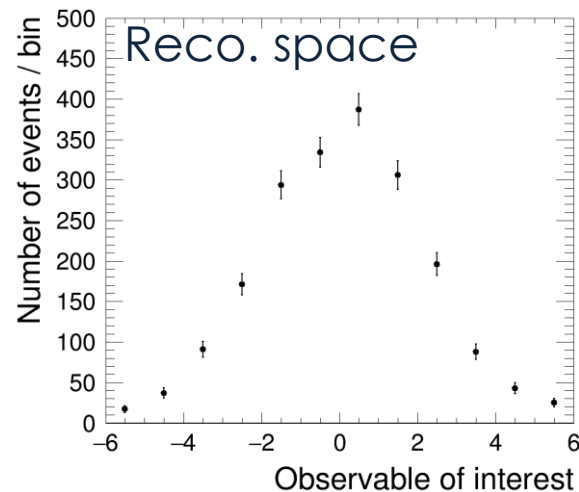
The unregularised result
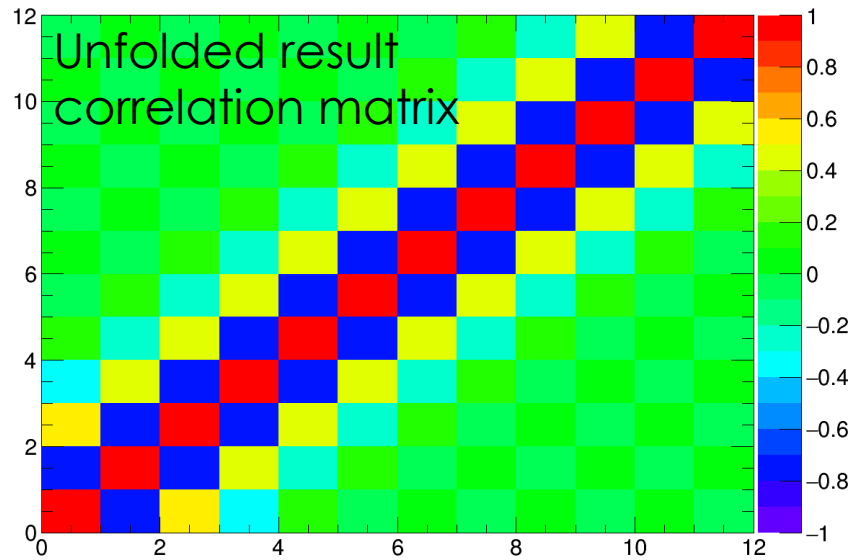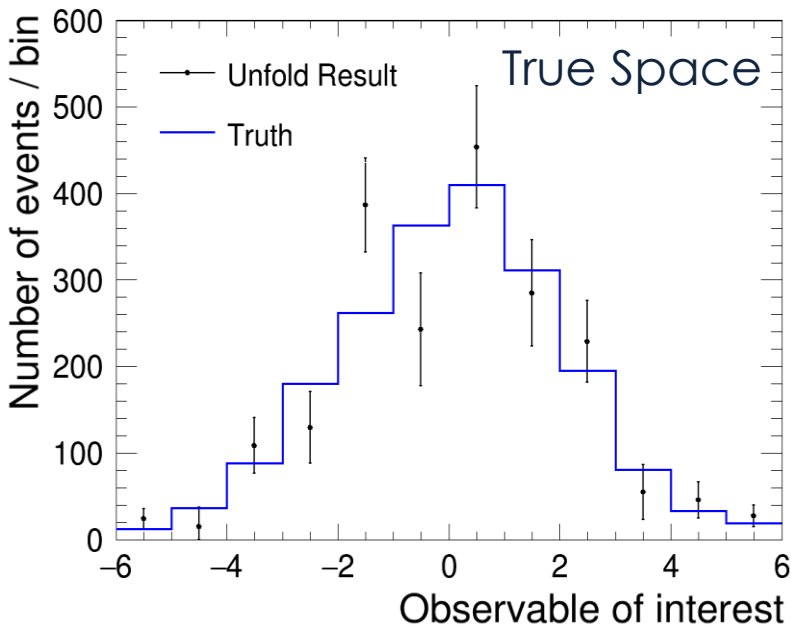
- The *unregularised result* typically oscillates around the truth
  - Small variations in reco. space cause large ones in truth space

- Although the agreement between the truth and result *looks* poor, the result is accompanied by large bin-to-bin anti-correlations

- A quantitative goodness of fit finds: $\frac{\chi^2}{NDOF} = 0.44$, pretty good!

- If we want to minimise bias this is probably the best thing we can do

The unregularised result
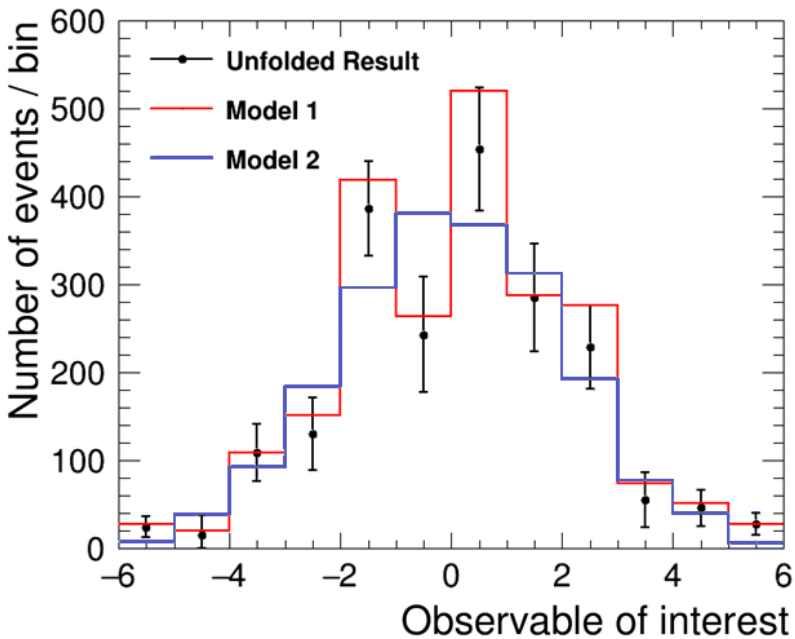
- Although the result is absolutely correct, it can be almost meaningless without the accompanying covariance matrix.

- Can you judge which of the models on the left fits the result best?

The unregularised result
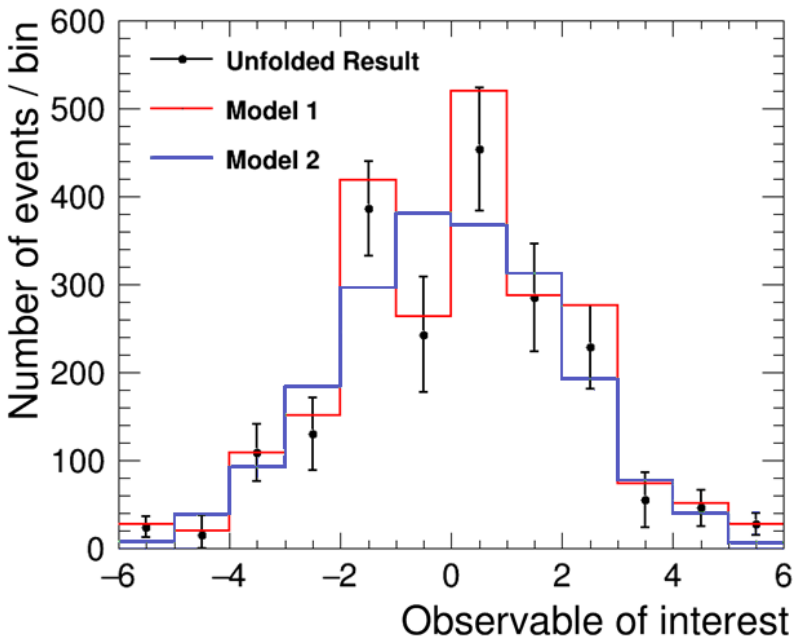
$$\chi_1^2/DOF = 2.7 \rightarrow p = 0.0013$$

$$\chi_2^2/DOF = 1.2 \rightarrow p = 0.25$$

- Although the result is absolutely correct, it can be almost meaningless without the accompanying covariance matrix.

- Can you judge which of the models on the left fits the result best?

- Even when we have the covariance matrix "chi-by-eye" is not very reliable …

- The unregularised result is **good for calculating χ^2**, but potentially very misleading for "by-eye" comparisons.

Case study: CC0π analysis

$0.98 < \text{true } \cos\theta_\mu < 1.00$

Phys. Rev. D **93**, 112012 (2016)

Stat. correlation matrix for these three bins

- Our *unregularised* ND280 $\nu_\mu CC0\pi$ analysis shows a dip in the momentum distribution for forward going muons at about 1 GeV

- Looks interesting! There has been some discussion about what this large "dip" could be

- But in reality the **large anti-correlations** between the pertinent bins make this result compatible with no dip.

- The "dip" may just be a statistical effect from the unregularised unfolding

The case for regularisation

Unregularised results with large anti-correlations are the best option for making χ^2 comparisons, but:

- *We might want to have an idea of the result's shape or to compare the model in a specific region of phase-space*

- *We can't accurately estimate the χ^2 from a plot in a paper or conference*

Perhaps producing a result which can be better interpreted by eye could be useful too …

Can we just re-bin it?

The oscillatory unfolded results are caused by a combination of:

- *Fine binning compared to the detector resolution*

- *Large statistical uncertainty in the reco. data*

Gaus (0,0.6) smear
1.0 bin width
2000 events
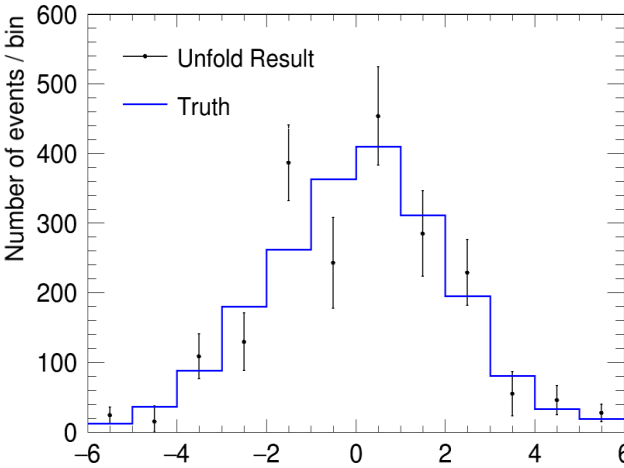
Gaus (0,0.6) smear
1.5 bin width
2000 events
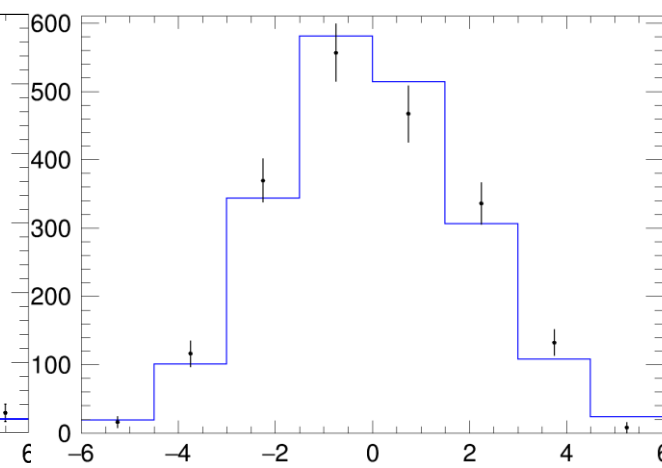
Gaus (0,0.6) smear
1.0 bins
10000 events
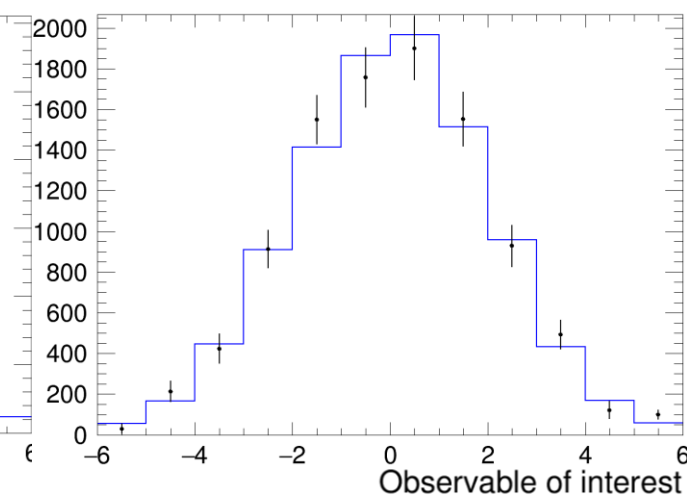
Can we just widen our bins or add more reco. bins for a fixed number of true bins until we get a smooth result?

Can we just re-bin it?

- **Yes!** Many of T2K's recent results do this

- This is largely unbiased unfolding where the **resolution of the detector is clearly shown by the width of the bins**

- Potential issues*:
  - Bin widths optimised on simulation, not on data
  - Coarser binning can give greater model dependencies
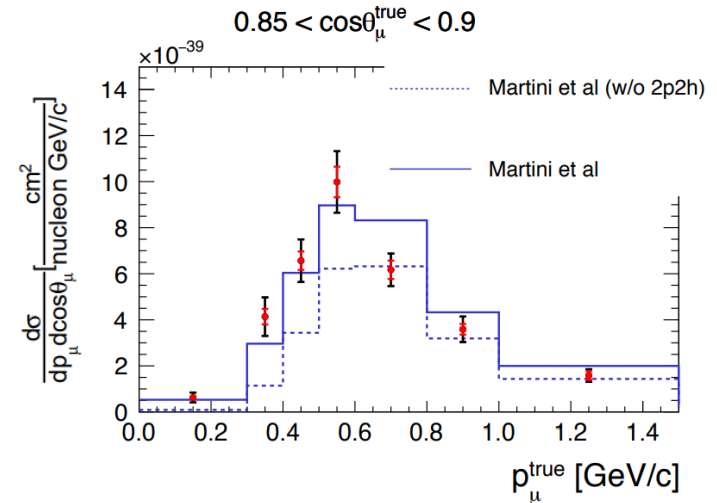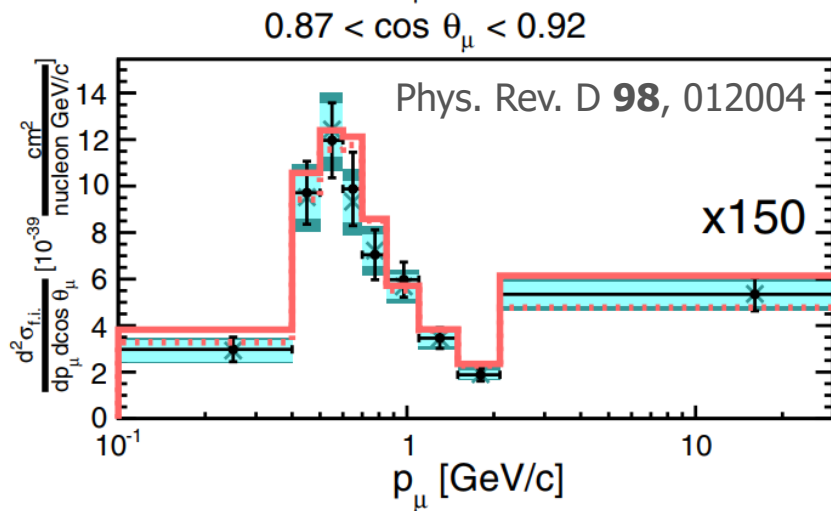
* Although these could be mitigated by first extracting a result in fine bins and then combining adjacent bins until the result is smooth

Direct regularisation

- Additional regularisation is often applied when there is insufficient constraints available from the data alone to reliably infer the desired result
  - E.g. if you have as many reco bins as true bins and a significant smearing

- It works by injecting some prior knowledge about the signal you are trying to measure into the cross section extraction with some strength
  - It can be "cross sections are usually smoothly changing"
  - It can also be "cross sections look like GEANT" …

- In the D'Agostini method, regularisation is gradually weakened by repeated iterations. For the likeihood fitting method there is no built in regularisation.

- Too strong regularisation can cause both bias and under-coverage

- All this can sometimes avoided by just having more reco bins than true bins

- But if you really insist on using it there are options …

Tikhonov regularisation

- Rather than combining bins completely, another option is to **loosely tie them together** with a penalty term (to be used in the likelihood fitting method of cross-section extraction – see backups)

$$\chi^2_{reg} = p_{reg} \sum_i (bin_i - bin_{i-1})^2$$

(this is just one potential penalty term, others are possible depending on how exactly you want to smooth your result)

- If p_{reg} is very large then this is equivalent to combining bins

- The inclusion of a penalty term means that the result moves away from the maximum likelihood solution and is therefore **at least a little biased.**

- How can we choose p_{reg} to give us a result we can better compare to by-eye but avoid excessive bias?

The role of regularisation

Flat input sim.
(truth and reco)

The role of regularisation

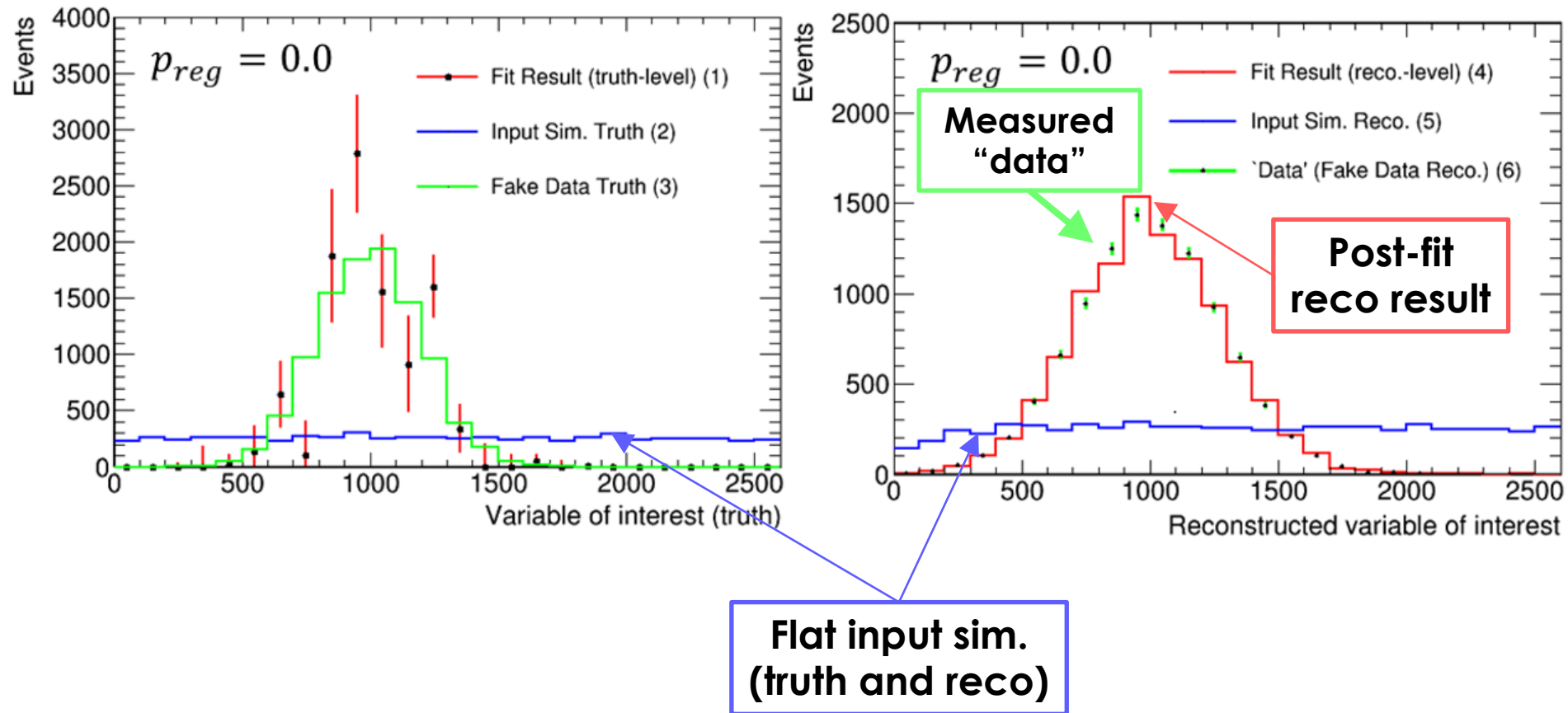Left plot: $p_{reg} = 0.0$ — Events vs Variable of interest (truth)
- Fit Result (truth-level) (1)
- Input Sim. Truth (2)
- Fake Data Truth (3)

Right plot: $p_{reg} = 0.0$ — Events vs Reconstructed variable of interest
- Fit Result (reco.-level) (4)
- Input Sim. Reco. (5)
- `Data' (Fake Data Reco.) (6)

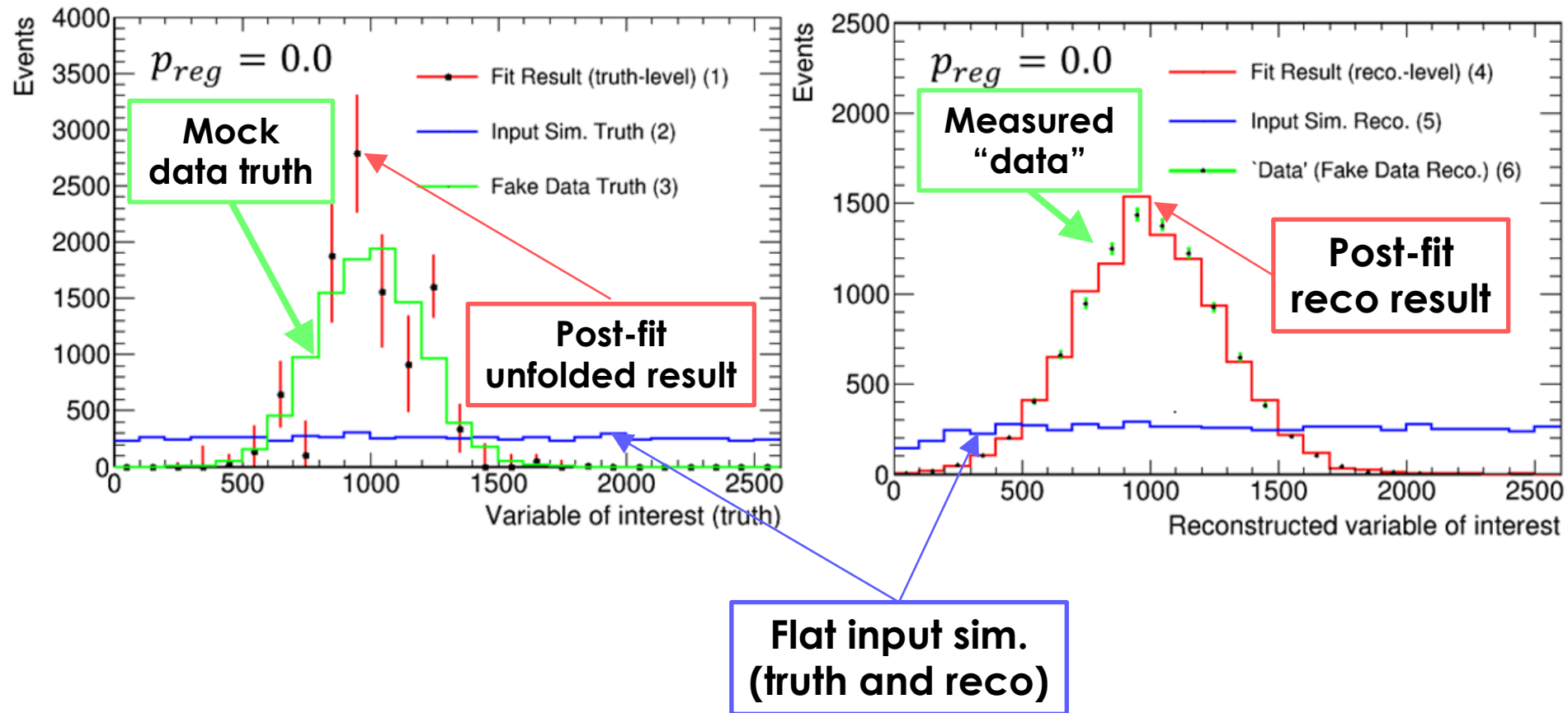**Measured "data"**
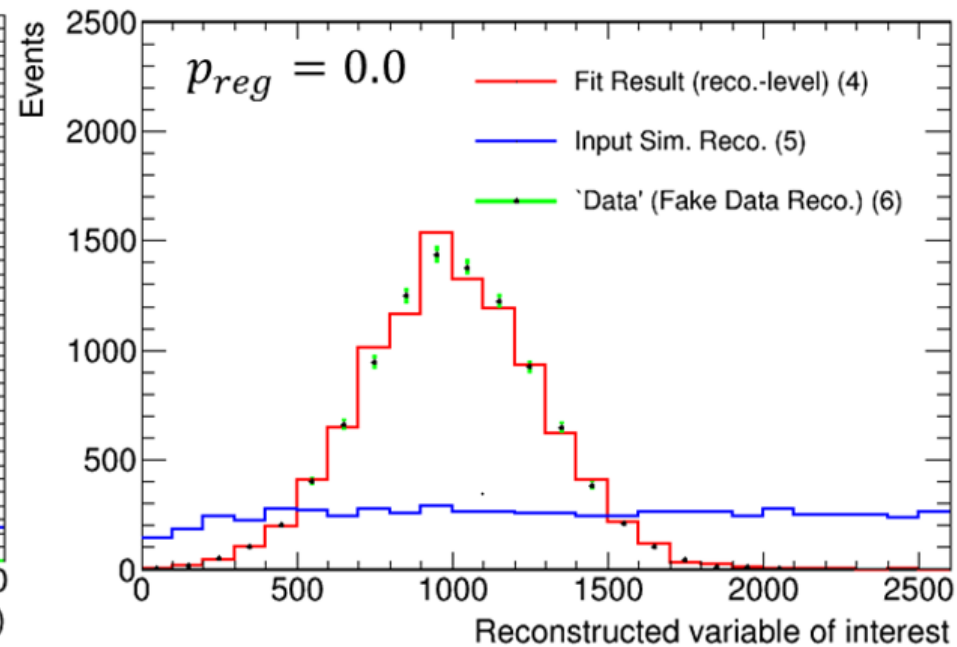
Post-fit reco result
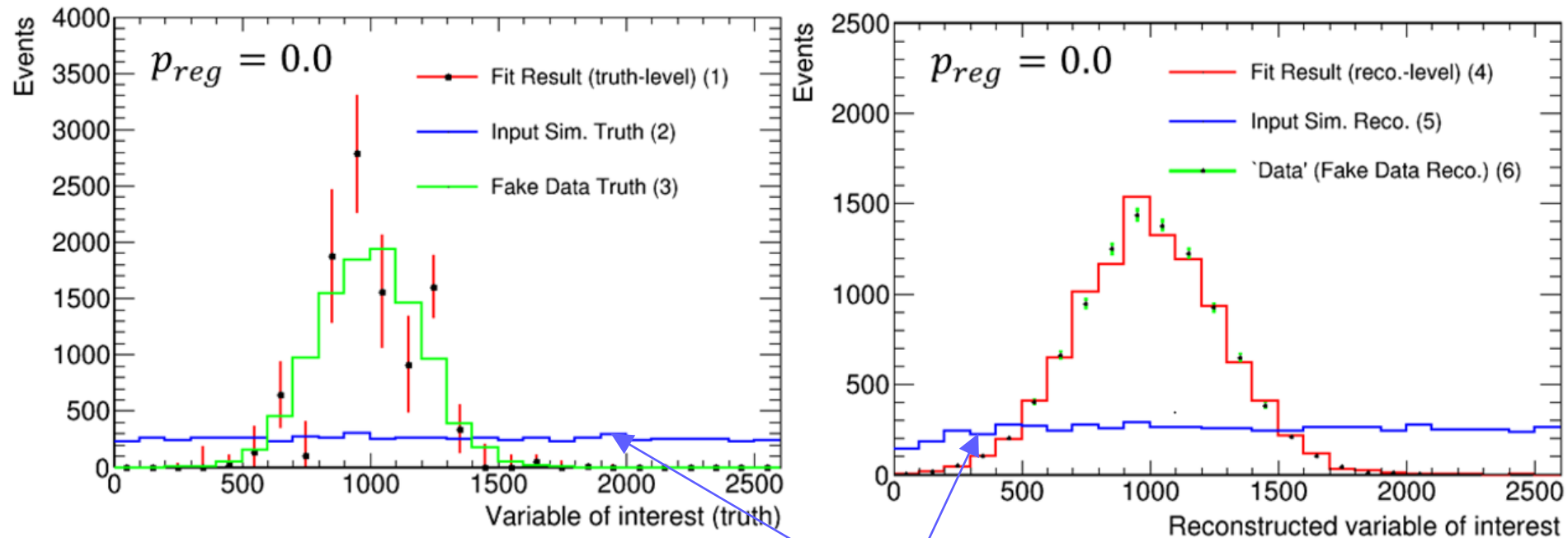
Flat input sim. (truth and reco)

The role of regularisation

The role of regularisation
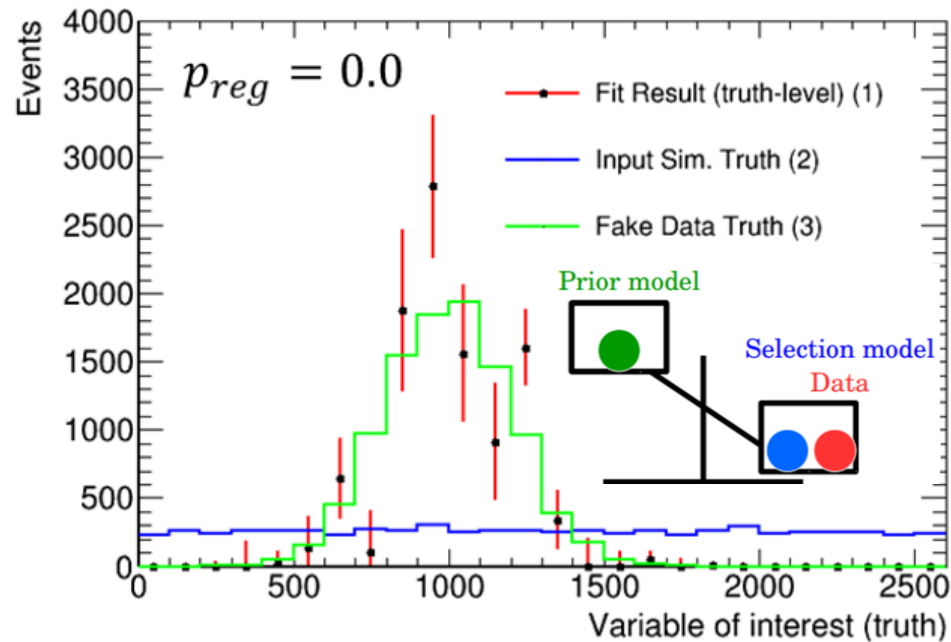
Read p_{reg} as regularisation strength

The role of regularisation

Read p_{reg} as regularisation strength

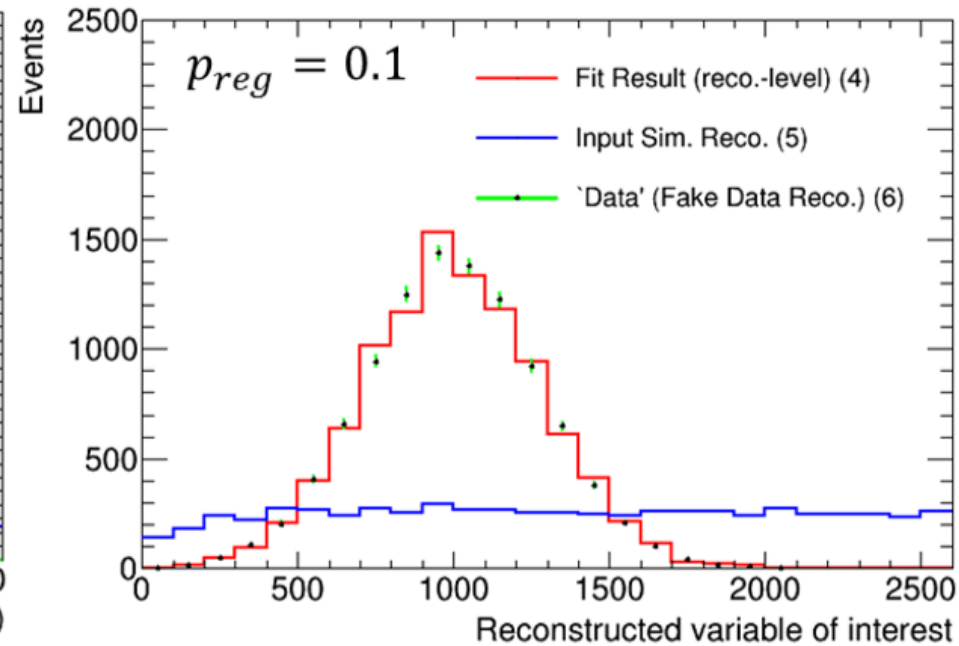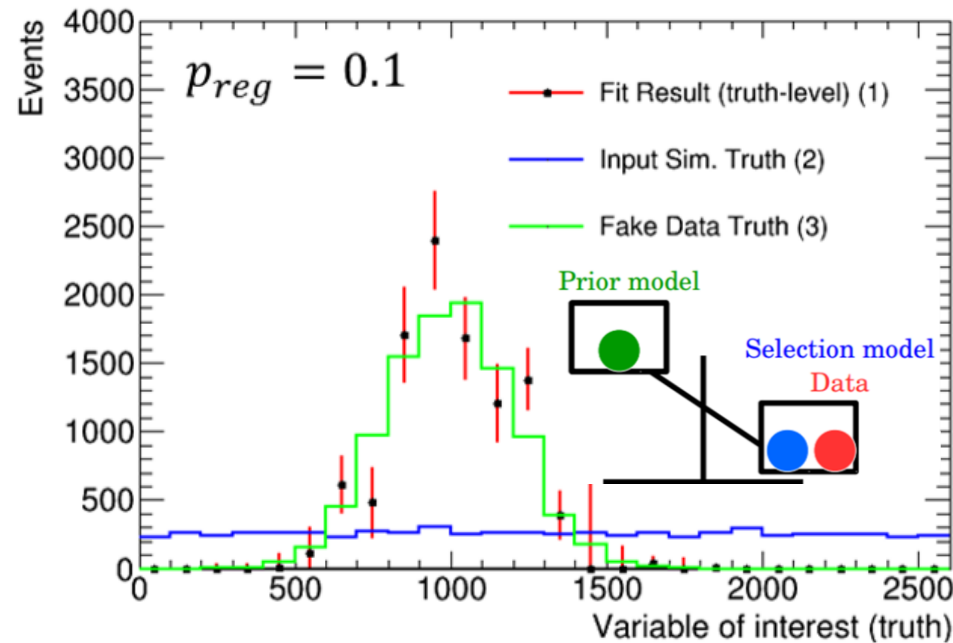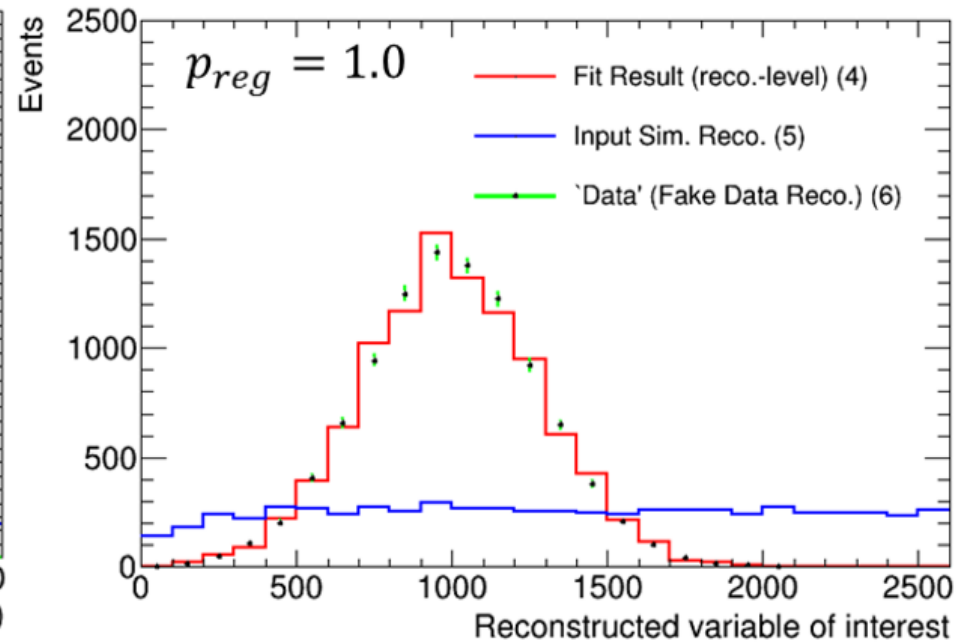# The role of regularisation

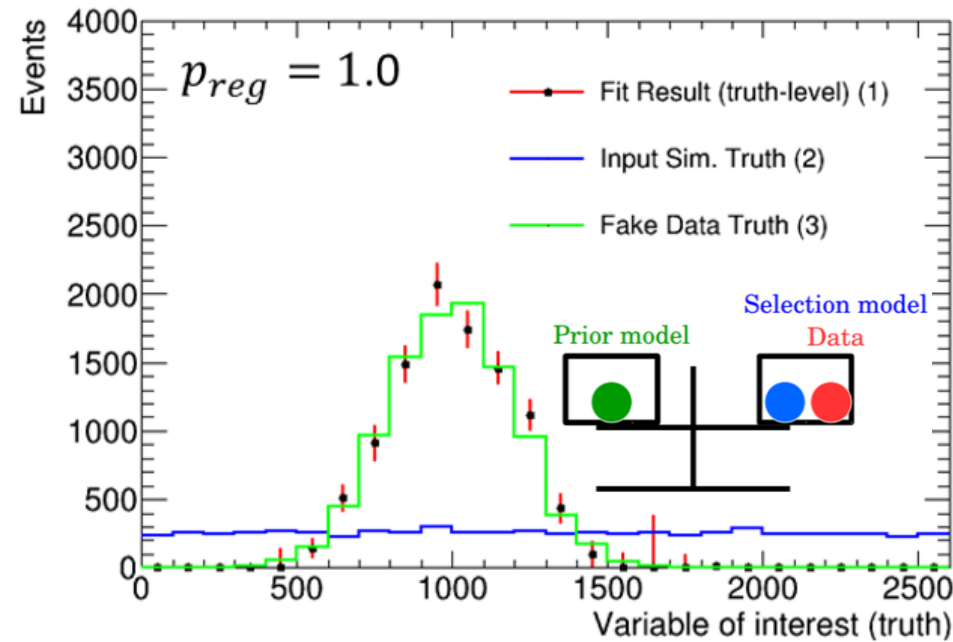Read p_{reg} as regularisation strength

The role of regularisation
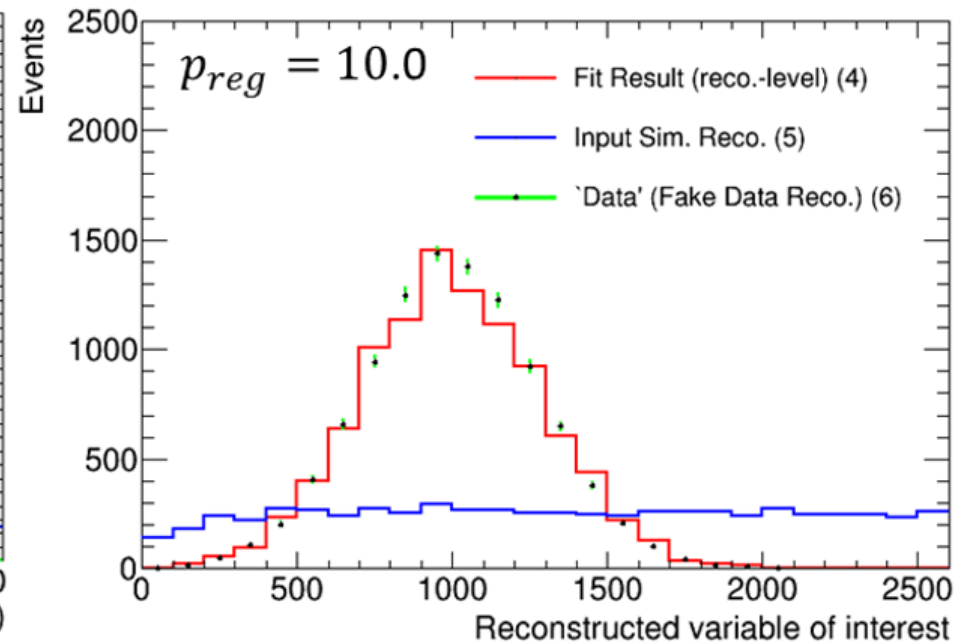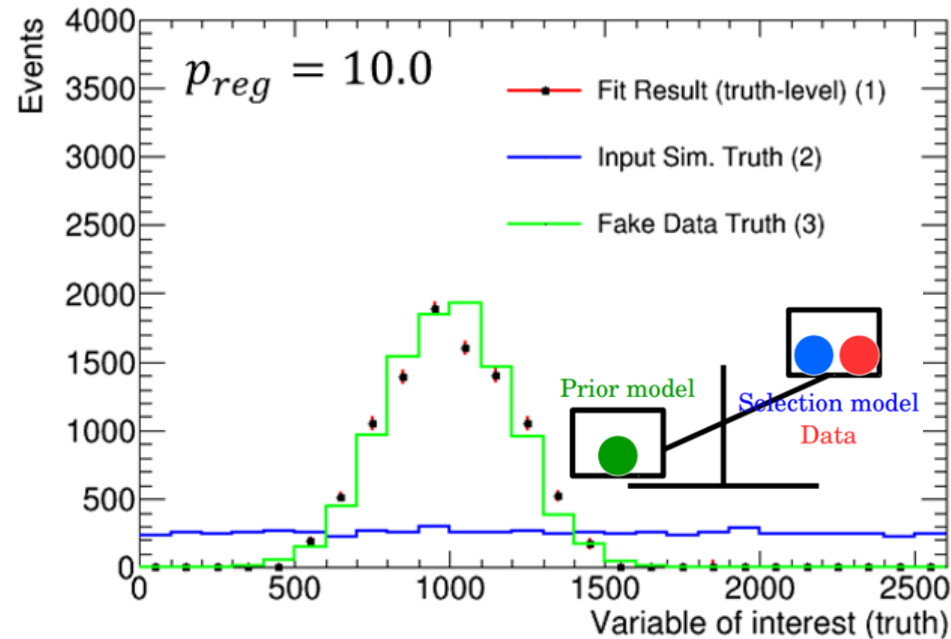
Read p_{reg} as regularisation strength

Regularisation optimisation: The L-curve

$$\sum_i (bin_i - bin_{i-1})^2 = \frac{\chi^2_{reg}}{p_{reg}}$$

The quantity we want to minimise with our regularisation
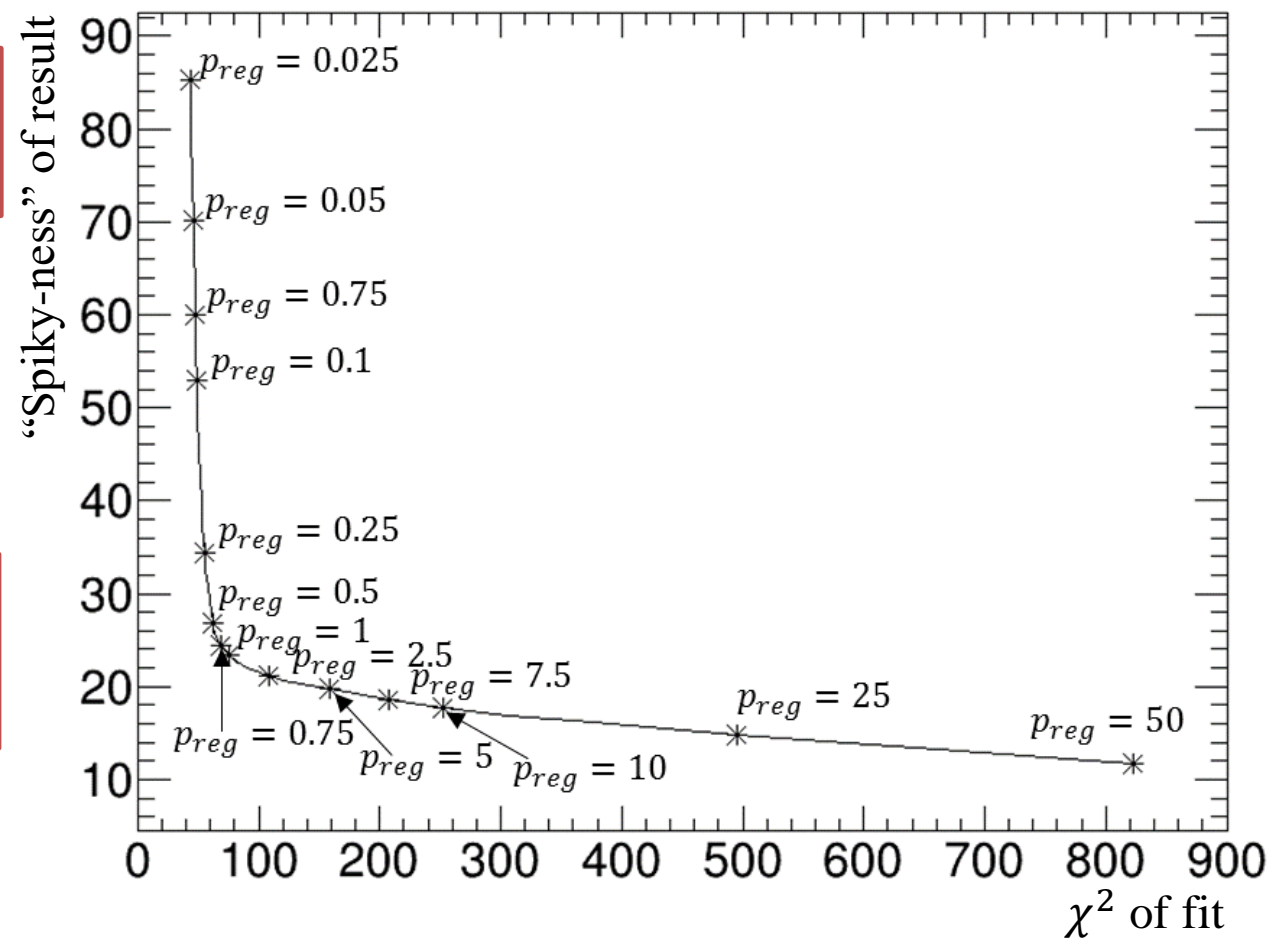
This is a measure of bias – basically the deviation from the unregularised result

Regularisation optimisation: The L-curve

- Balance regulation with bias by **choosing the "kink" in the curve**

- **L-curve can be formed on real data** – data driven regularisation
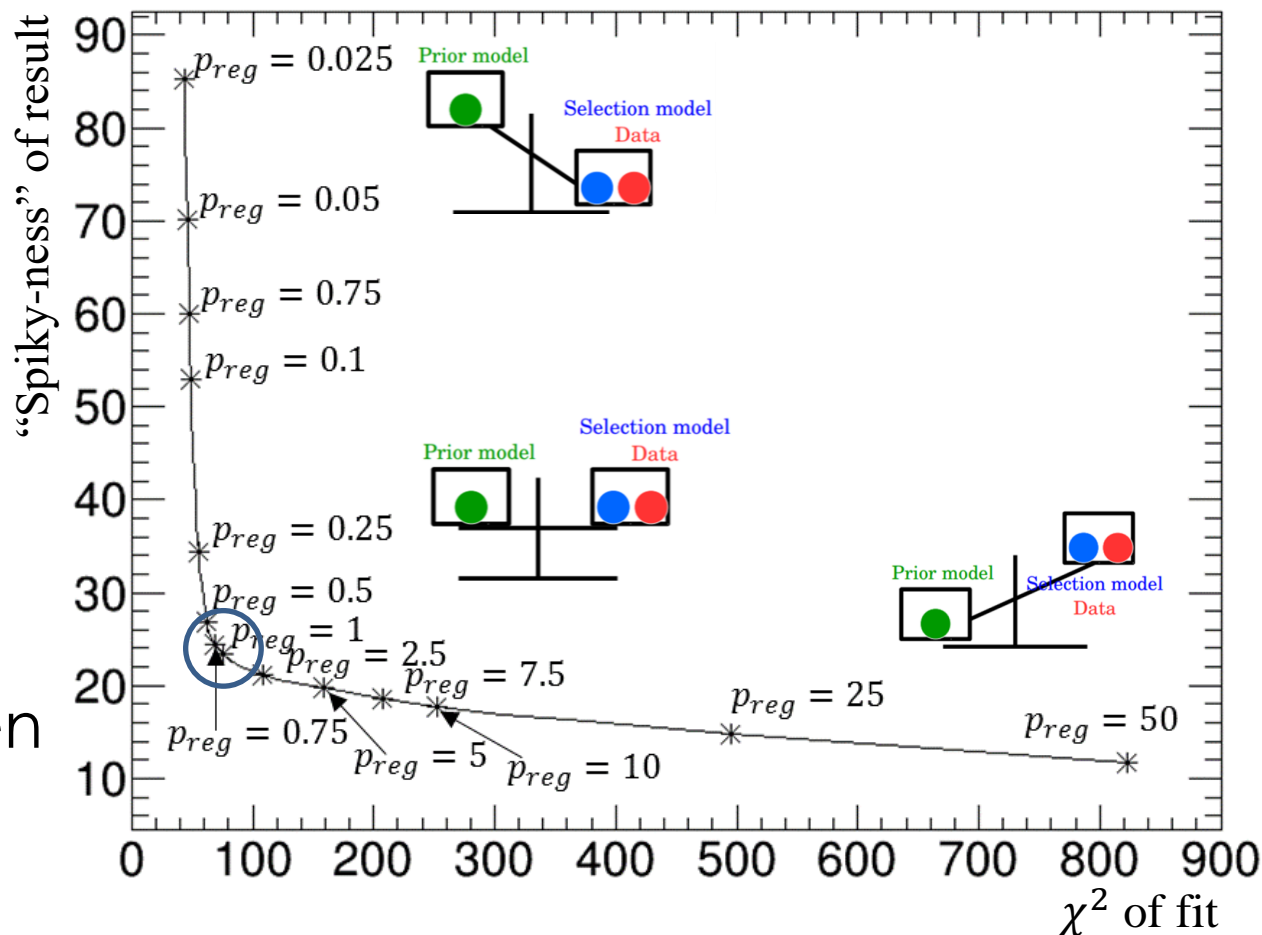
- Well established method to **select the smoothest of many almost degenerate solutions**:

http://epubs.siam.org/doi/abs/10.1137/1034115
http://epubs.siam.org/doi/abs/10.1137/0914086   http://arxiv.org/pdf/1205.6201v4.pdf - use in TUnfold

Regularisation optimisation: The L-curve

- **Caveat:** regularisation can lead to under-coverage

- **Simple solution:** provide the unregularised result.

- **Better solution:** consider providing "undersmoothed" result [Kussela, 2016] – effectively move up the L-Curve to ensure good coverage.
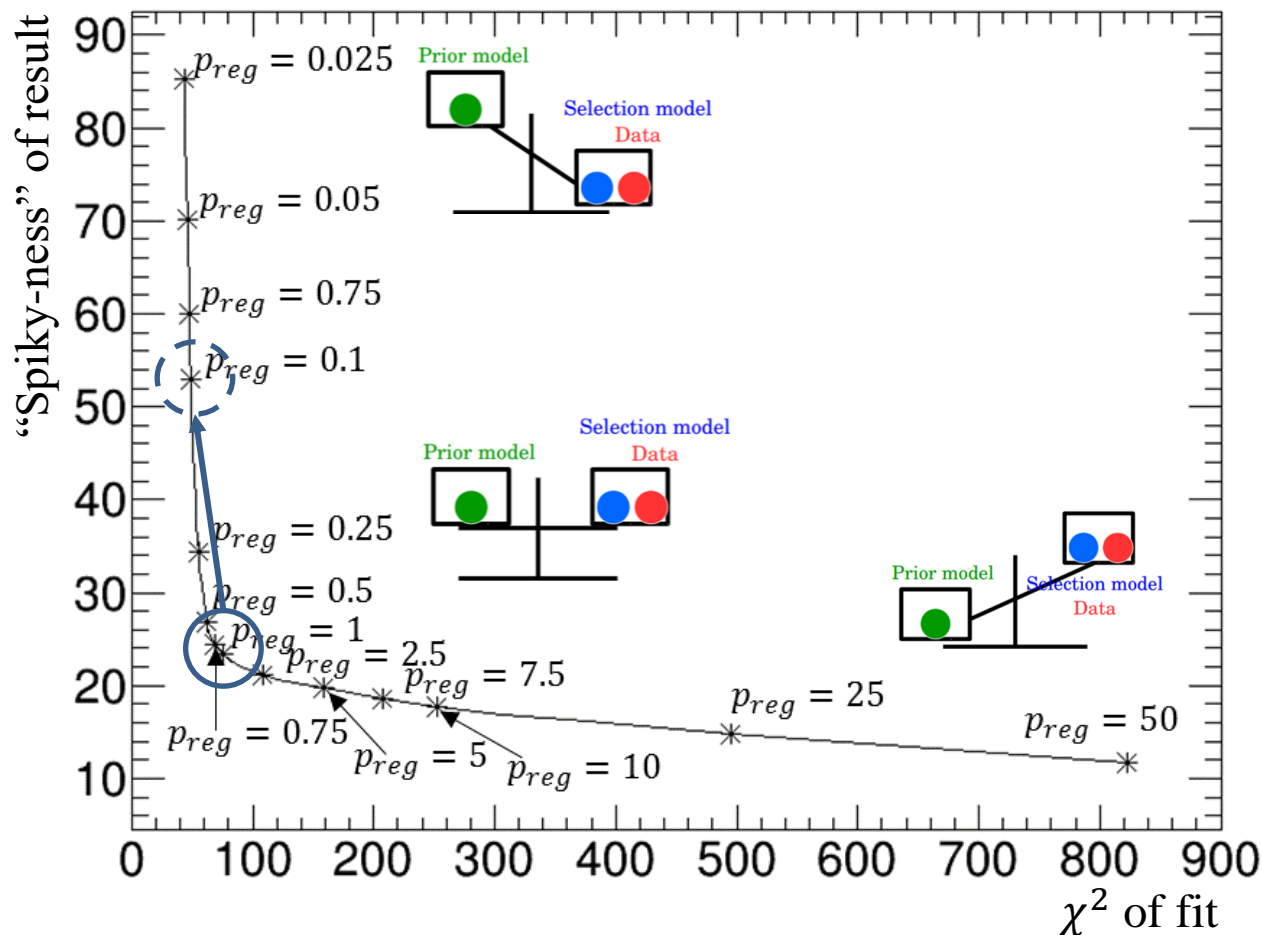
Case study: CC0π in δp_T

- Measure CC0π+protons cross section in missing transverse momentum (δp_T)

  Phys. Rev. D **98**, 032003 (2018)

- Unregularised best for χ^2, regularised best for actually showing anywhere

Case study: CC0π in δp_T

- Measure CC0π+protons cross section in missing transverse momentum (δp_T)

  Phys. Rev. D **98**, 032003 (2018)

- Unregularised best for χ^2, regularised best for actually showing anywhere

Unregularised

L-Curve
$p_{reg} = 3.0$

Regularised

Correlation matrix
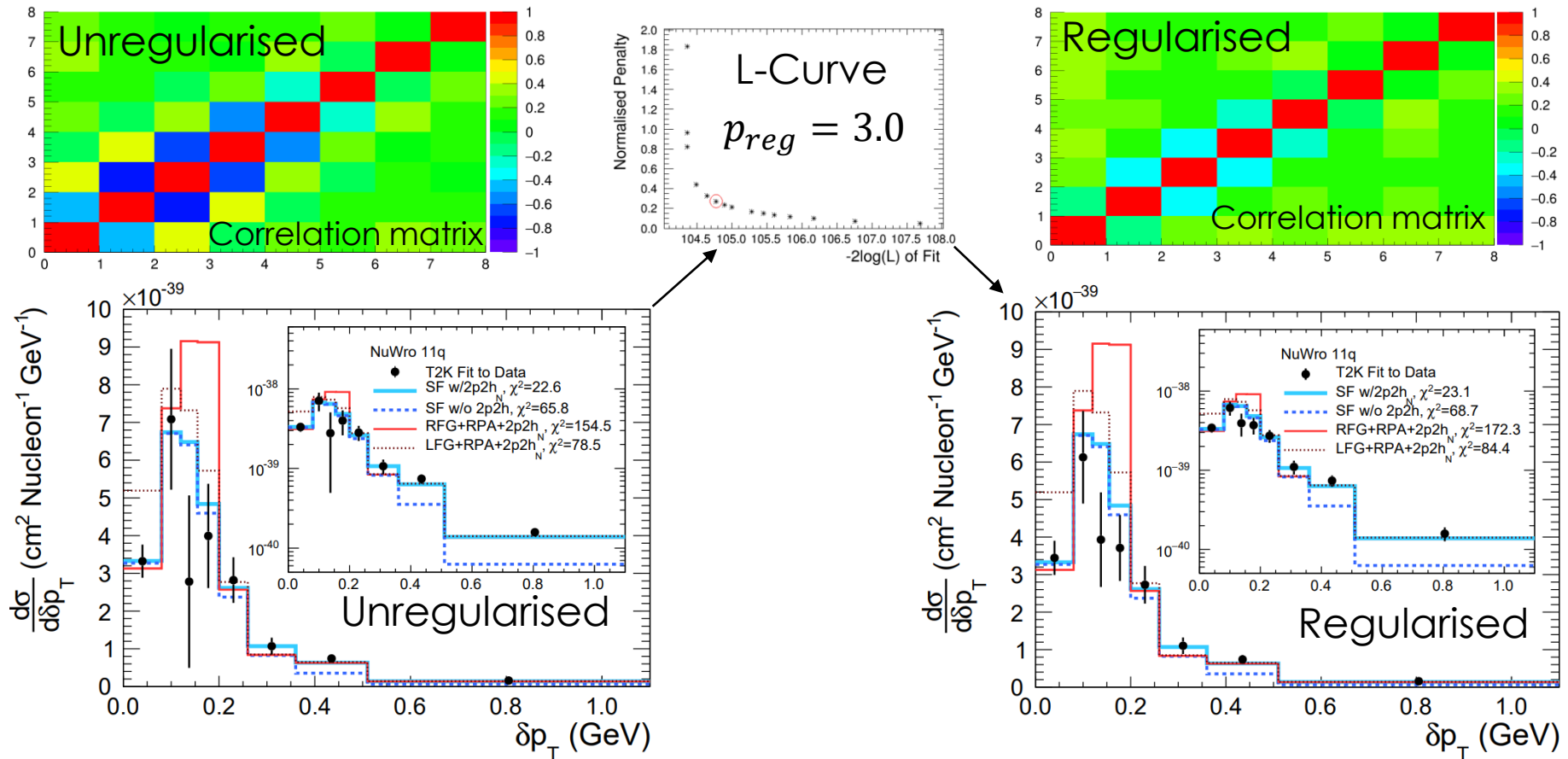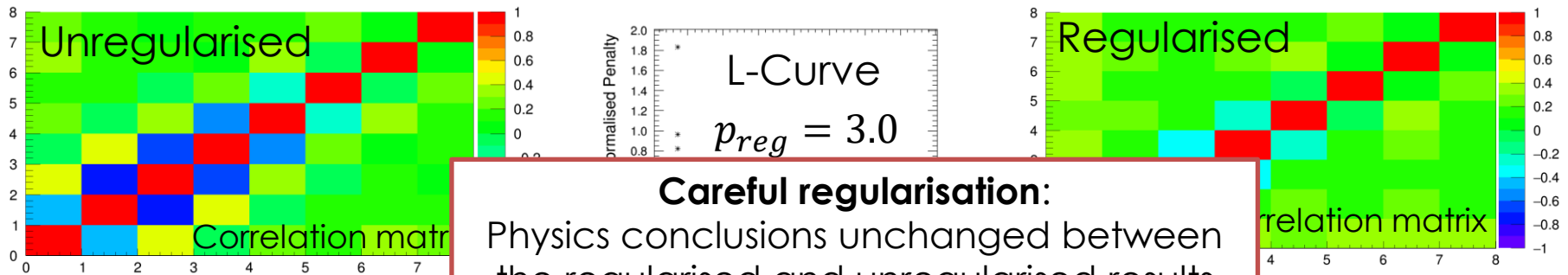
Careful regularisation:
Physics conclusions unchanged between the regularised and unregularised results

NuWro 11q
- T2K Fit to Data
- SF w/2p2h, χ^2=22.6
- SF w/o 2p2h, χ^2=65.8
- RFG+RPA+2p2h$_N$, χ^2=154.5
- LFG+RPA+2p2h$_N$, χ^2=78.5

NuWro 11q
- T2K Fit to Data
- SF w/2p2h, χ^2=23.1
- SF w/o 2p2h, χ^2=68.7
- RFG+RPA+2p2h$_N$, χ^2=172.3
- LFG+RPA+2p2h$_N$, χ^2=84.4
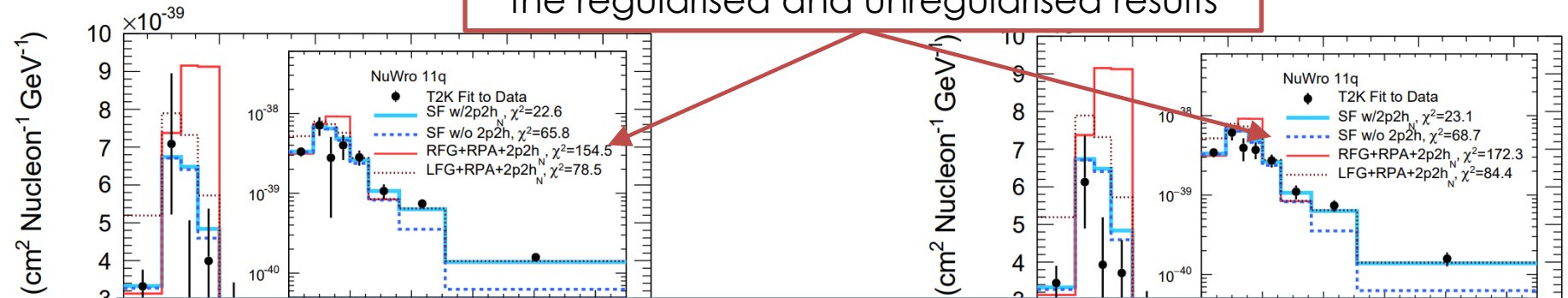
- **CAUTION:** here we can see that whilst the unregularised result has reduced correlations, but interpreting a goodness of fit by eye is still challenging.

- **You still need a covariance for a regularised result!**

D'Agostini's method

- Using Bayes' theorem* to form unsmearing matrix:

Data input

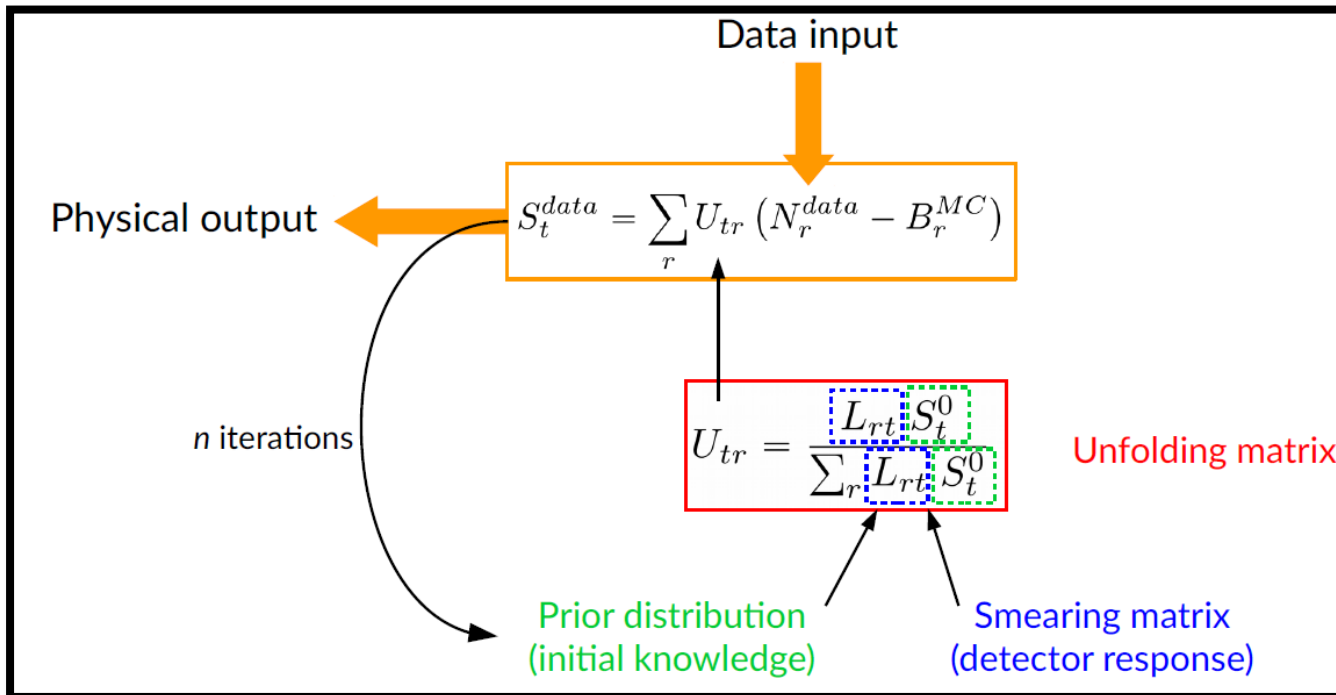$$S_t^{data} = \sum_r U_{tr} \left(N_r^{data} - B_r^{MC} \right)$$

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)

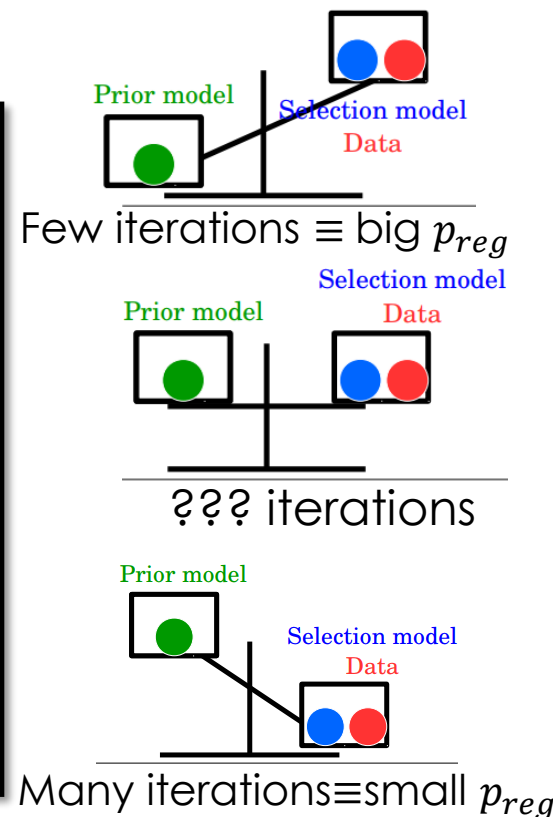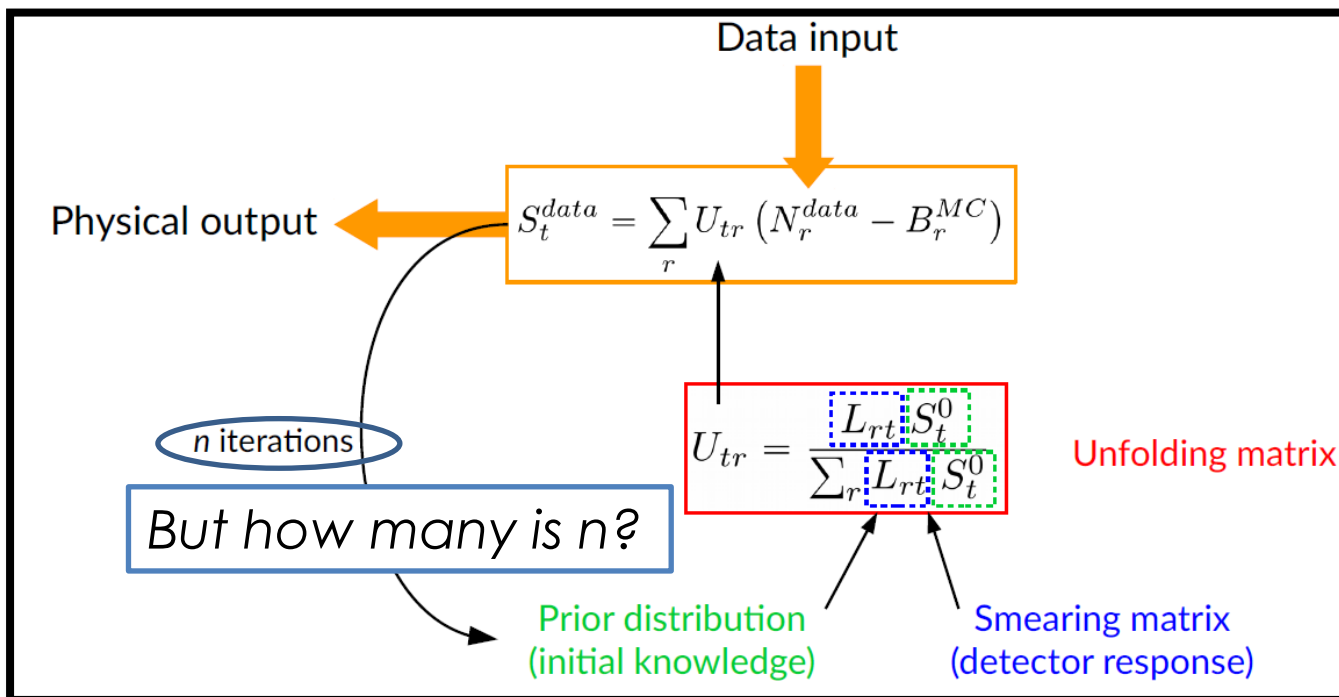Smearing matrix (detector response)

- 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** (problem dependent, typically >>4) → **unregularised result**

D'Agostini's method

- 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

But how many is n?

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

Unfolding matrix

Prior distribution (initial knowledge)

Smearing matrix (detector response)

Prior model   Selection model   Data

Few iterations $\equiv$ big p_{reg}

Selection model   Prior model   Data

??? iterations

Prior model   Selection model   Data

Many iterations$\equiv$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 (toy example in backups)

Efficiency corrections

An over-simplified xsec analysis
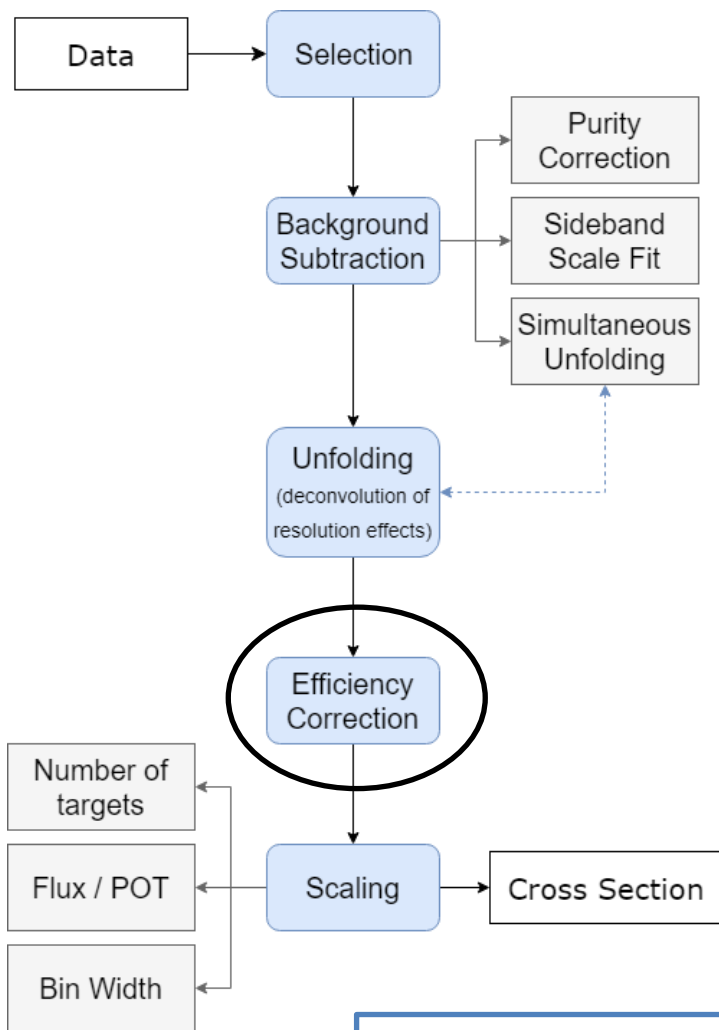
- After unfolding we have the a measure of the true number of *selected* signal events

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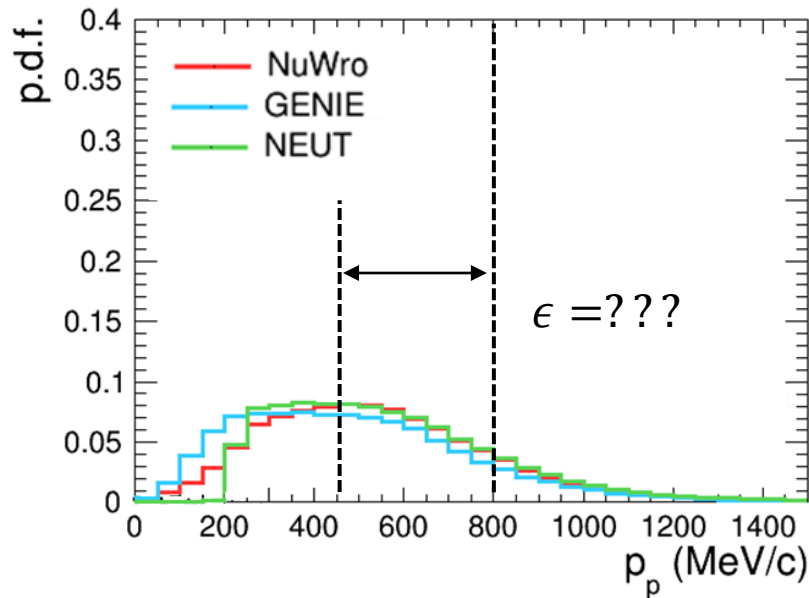
- But my detection efficiency depends on both proton momentum and angle (and on other particles, but let's focus on the angle for the moment!)

- 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
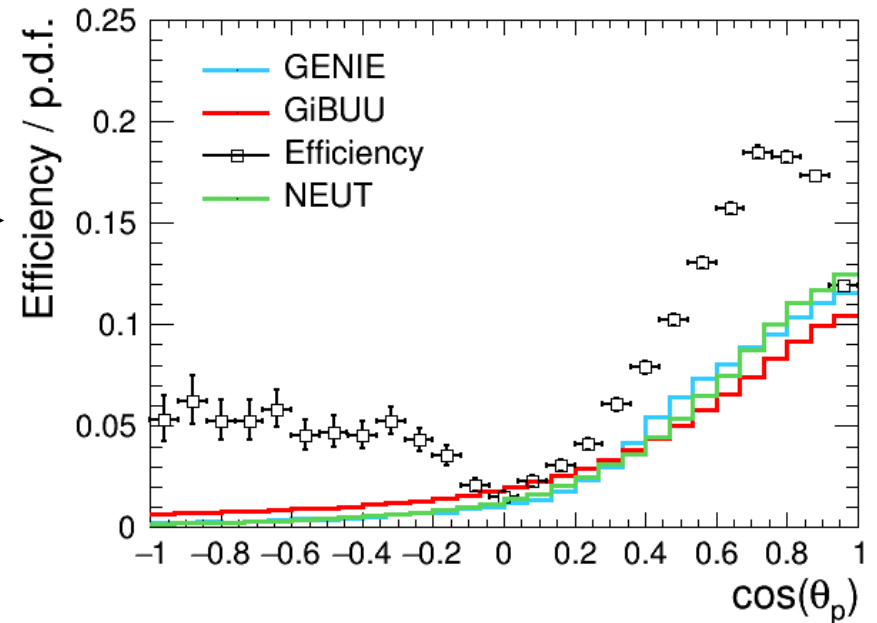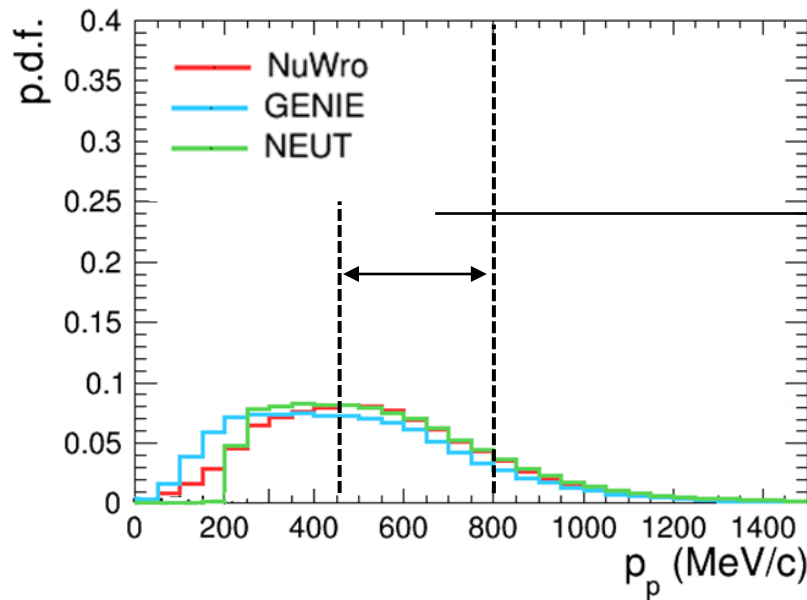
- The efficiency in the momentum bin a convolution of the efficiency **and the predicted cross section** in proton angle
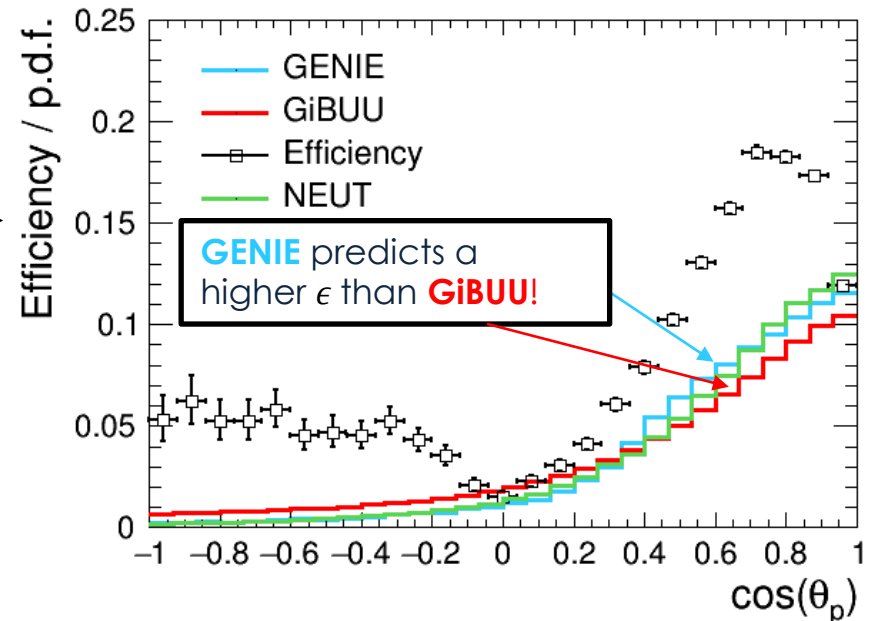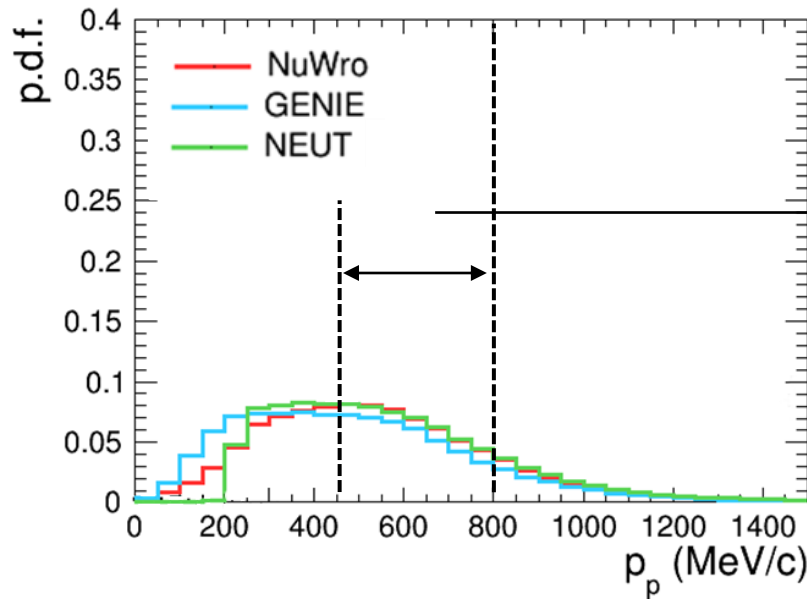
Toy example

- 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** in proton angle

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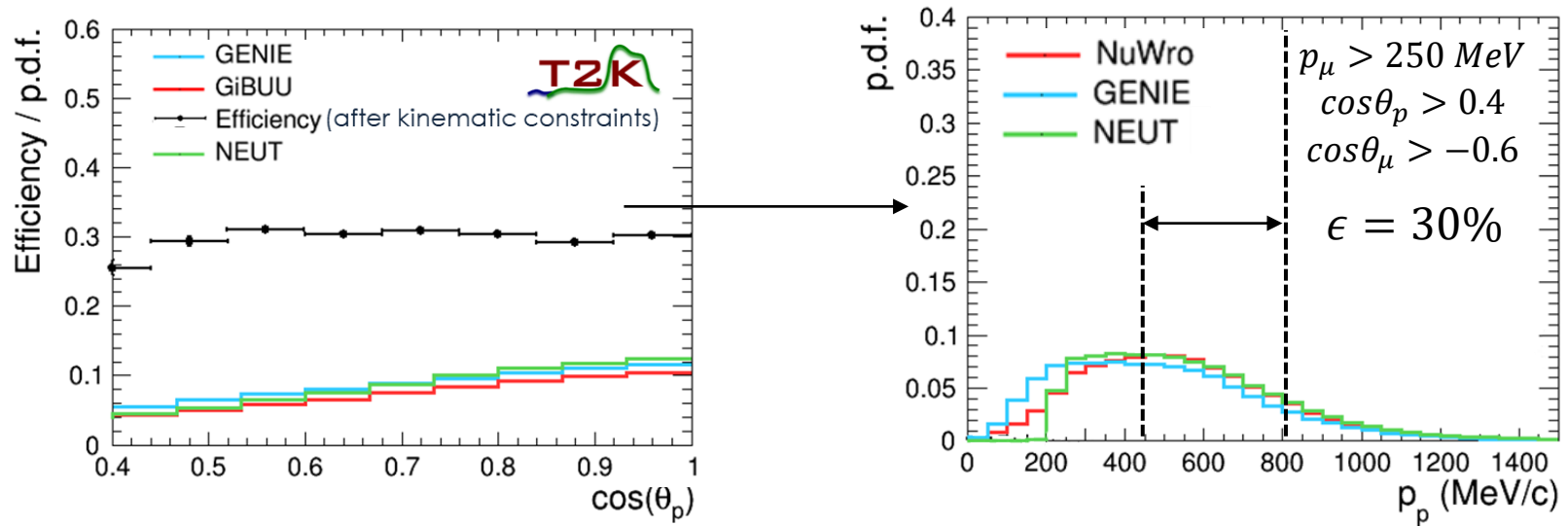
- In this case the shape of the input model doesn't alter the efficiency → **model independent correction**!

- Analyses should try to ensure integration only over flat-efficiencies in observables where simulations have poor predictive power (Example in backups)

Summary

- Unfolding / efficiency correcting without bias is hard – but there are a variety of ways to try and mitigate this

- All methods give results with important correlations between bins
  - χ^2 (or similar) is usually essential to validate physics conclusions

Unregularised Result

✓ Gives correct χ^2 with no unfolding bias

X Sometimes not for anything other than χ^2, can't trust "chi-by-eye".
  o However can still get a smooth result by rebinning (or more reco bins than true bins)

➤ Useful part of data release and as a reference to check bias of regularised results

Direct Regularisation

✓ Smoother results, easier to interpret

X Adds at least some bias – worse for getting reliable χ^2

➤ Not easy to choose a regularisation strength that suits data based on simulation → Use **data-driven** methods

Just don't unfold!

- Producing a good *unfolded* result that can be interpreted by-eye with is **hard**! But maybe there's another way …

Reco Level

Truth Level

Data

Forward Folding

Unfolding

Theorist's new model

Unfolded Result
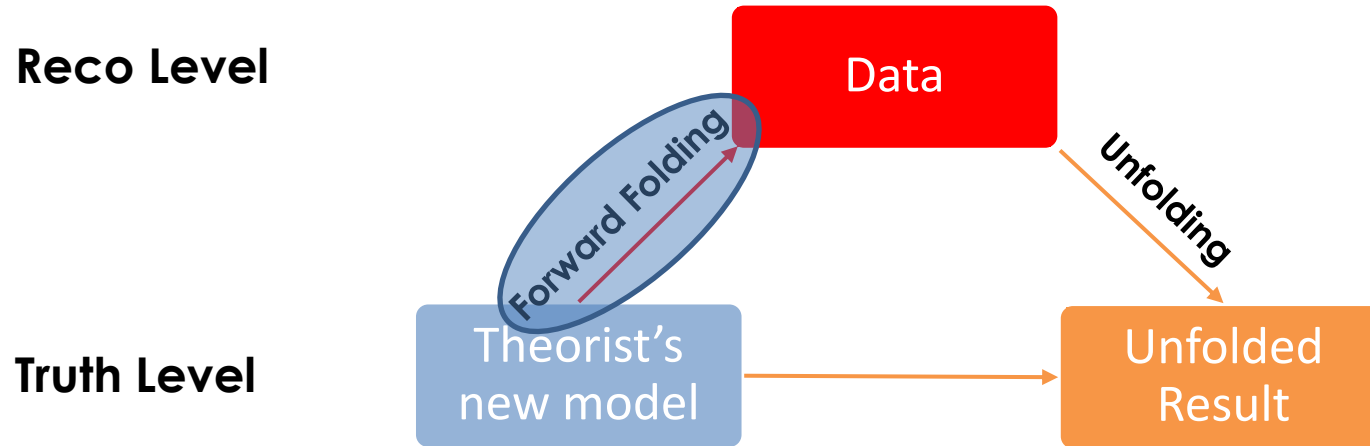
- No inherent bias

- No data point correlations: much easier to judge model comparisons "by eye"

- Best model separation is in the reco. space
 (Cousins, May, Sun – arXiv:1607.07038)

Thank you for listening

Providing the response matrix

How it works

data → reco expectation $\overset{?}{=}$ = (red matrix) × truth expectation
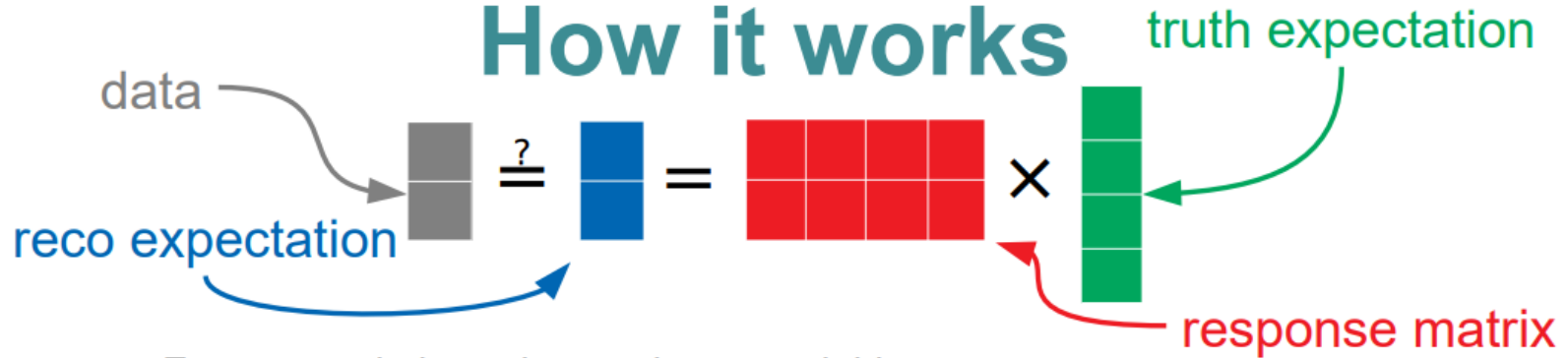
response matrix

- The challenge for the analyser is building and quantifying uncertainties on the response matrix

- Easy to just supply a reco to truth mapping in some interesting physics variable(s): but this may be highly model-dependent – defeats the purpose of not unfolding

- Ideally response matrix should contain all truth dimensions that effect reconstruction

Example: Gas Interactions analysis

- Reco. binning: 16 bins
    - 4 samples (main + 3 control samples). In each:
    - 2 bins in primary track angle
    - 2 bins in particle multiplicity (1 track vs. n tracks)

- Truth. binning: 11,760 bins (limited by simulation stats.)
    - 7 in momentum
    - 7 in angle
    - 5 in forward separation of track
    - 6 in backward separation of track
    - 8 bins to categorise events (4 FV, 4 OOFV)
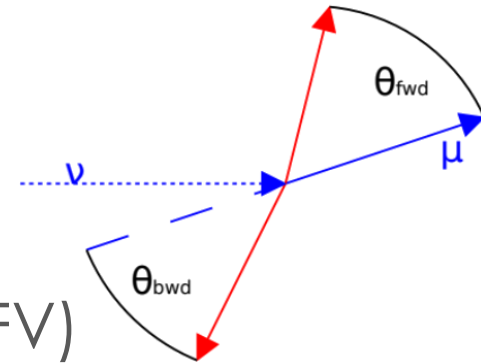
T2K Work in Progress

Response Matrix Utils (Lukas Koch)

- A tool for forward folding analyses
  - Builds response matrix
  - Tests model dependence
  - Evaluates uncertainties
  - Compare model to data (likelihoods, p-values, MCMC)

- More information: https://remu.readthedocs.io/

Unfolding

Forward folding

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 U_{ij} from S_{ji}.

- Simplest method: use S_{ji}^{-1}

Efficiency correction example

- 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
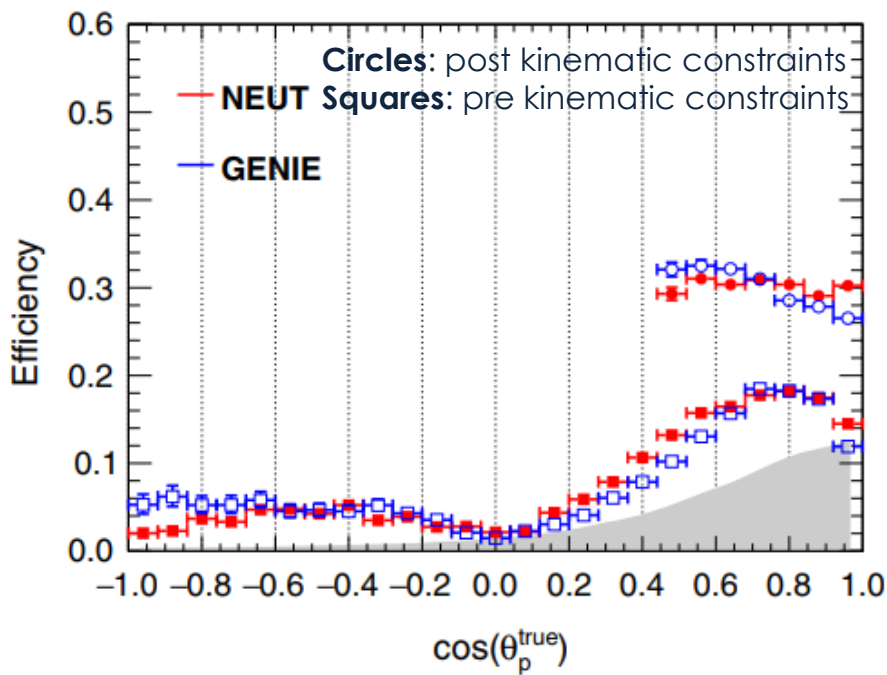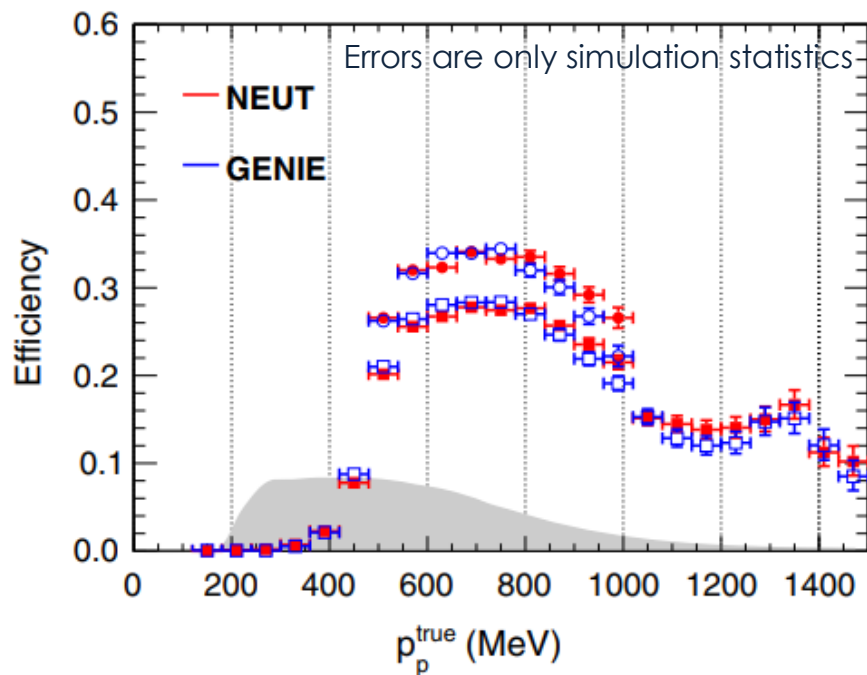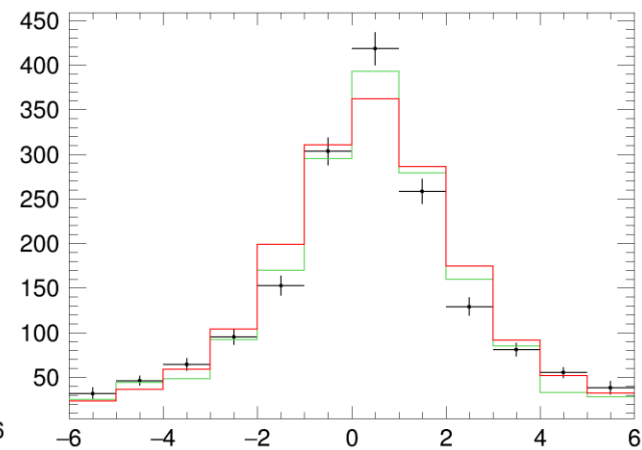
How many iterations? – Choose via MC
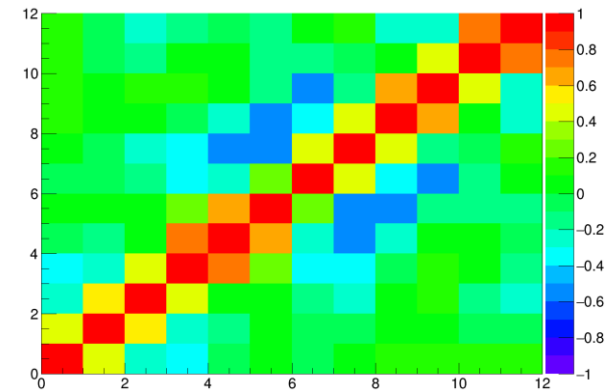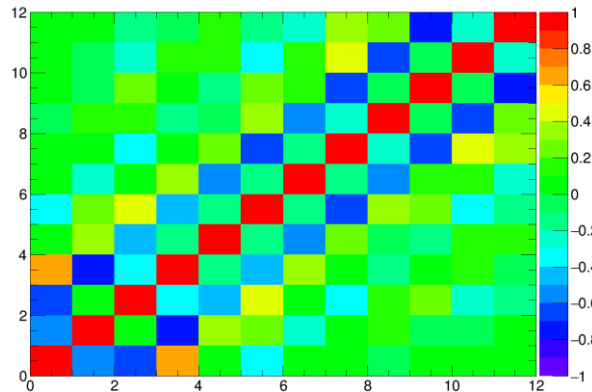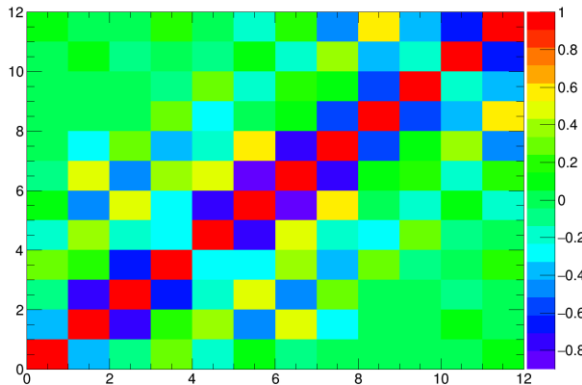
Gaus (0.1,1.0) smear,1.0 bin width, 2000 events, Truth is a BW(0.4,3.0), Input is a BW(0.3,2.5)

500 iterations (~unreg)
$\chi^2_{truth} = 12, \chi^2_{input} = 26$
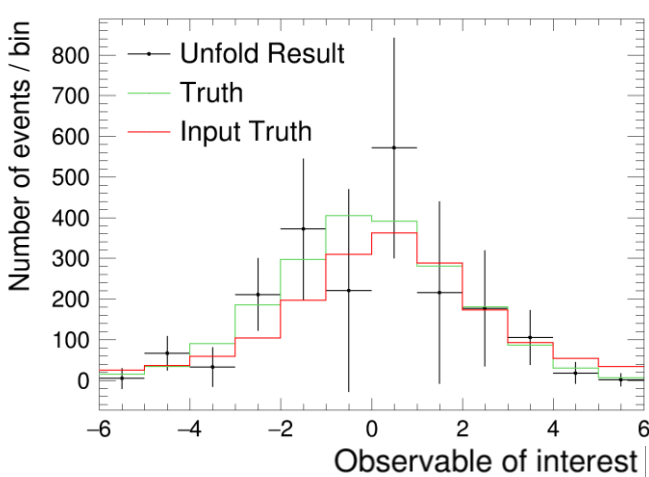
50 iterations
$\chi^2_{truth} = 18, \chi^2_{input} = 27$
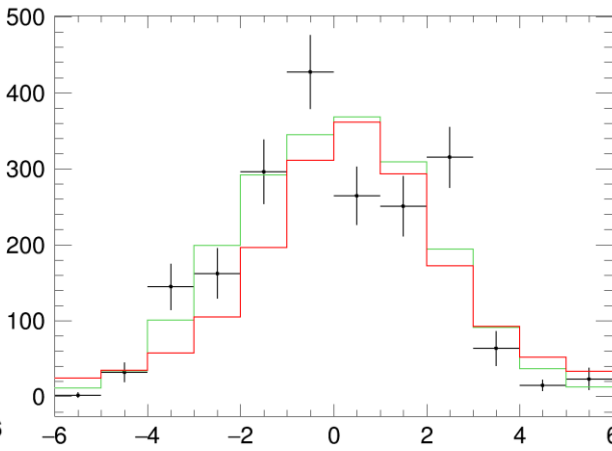
4 iterations
$\chi^2_{truth} = 299, \chi^2_{input} = 212$
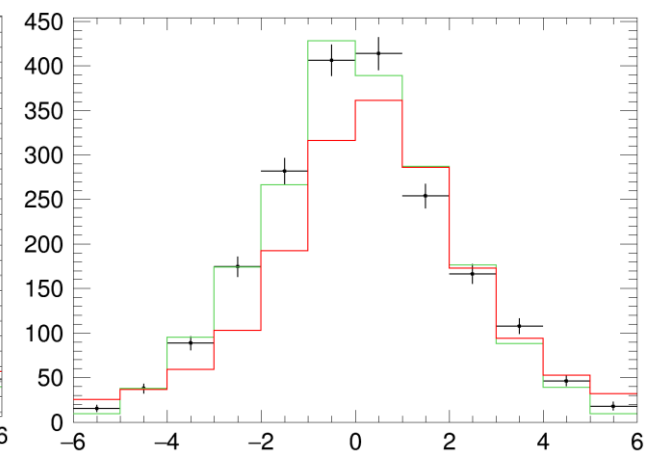
How many iterations? – Choose via MC

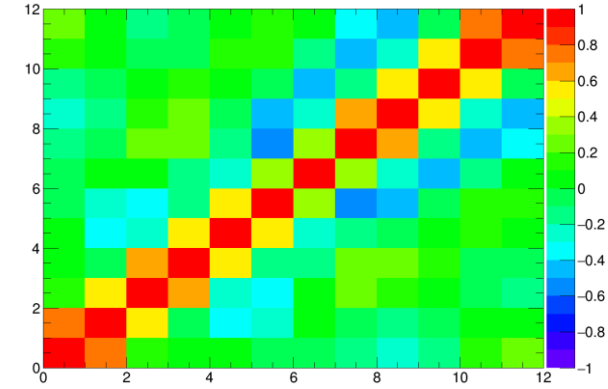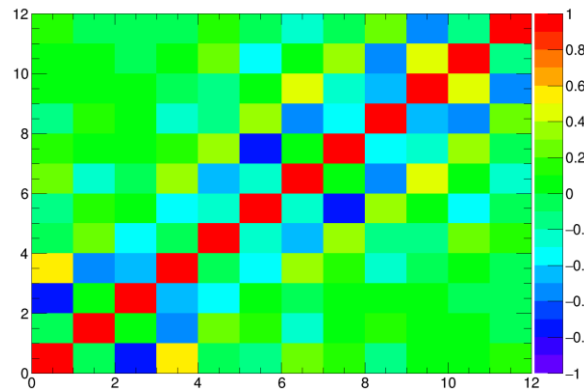Gaus (0.1,1.0) smear,1.0 bin width, 2000 events, Truth is a BW(0.4,3.0), Input is a BW(0.3,2.5)

500 iterations (~unreg)
$\chi^2_{truth} = 12, \chi^2_{input} = 26$

50 iterations
$\chi^2_{truth} = 18, \chi^2_{input} = 27$

4 iterations
$\chi^2_{truth} = 299, \chi^2_{input} = 212$

Too many iterations and we get an oscillating (but correct) result, **too few** and we bias to the input

• 50 iterations seems like a good choice here.

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)
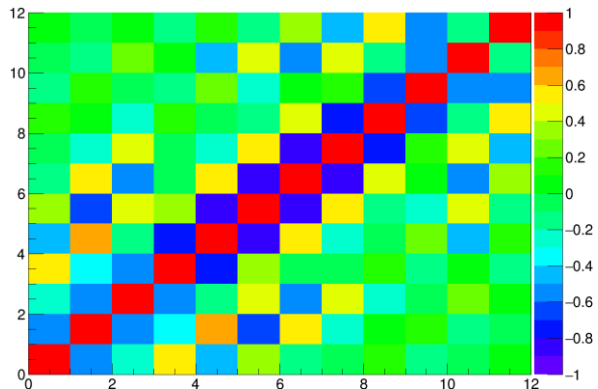

2000 iterations (~unreg)
$$\chi^2_{truth} = 2.9, \chi^2_{input} = 134$$

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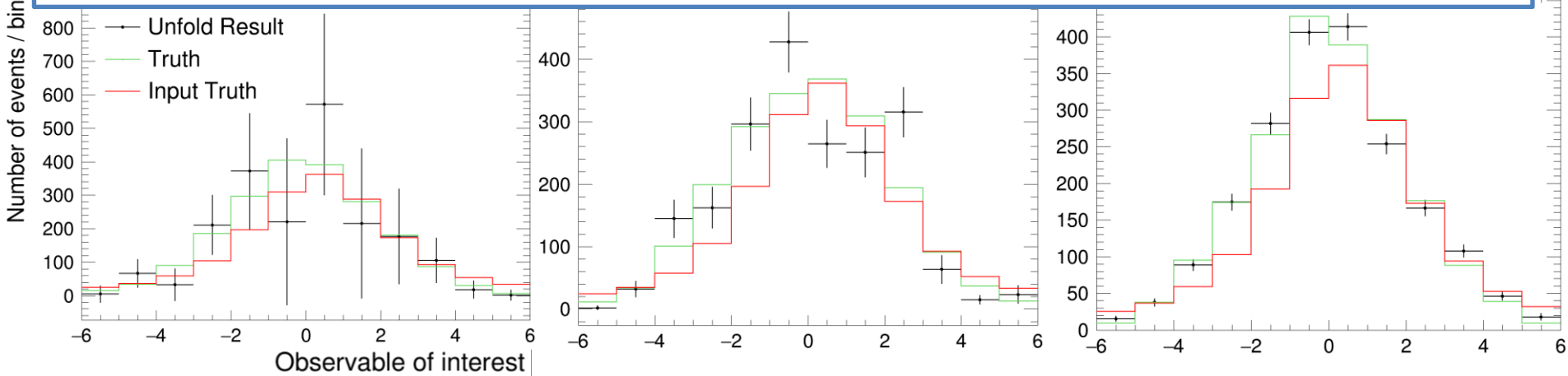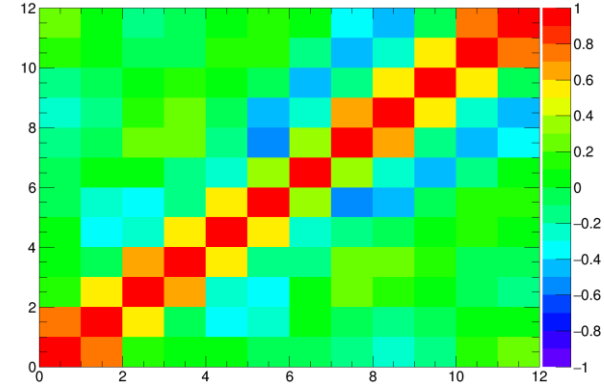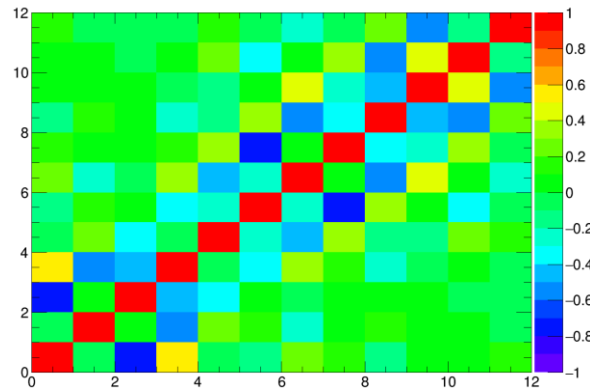$$\chi^2_{truth} = 32, \chi^2_{input} = 254$$

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

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)

A (small) change to the modelled truth and 50 iterations doesn't seem so good!
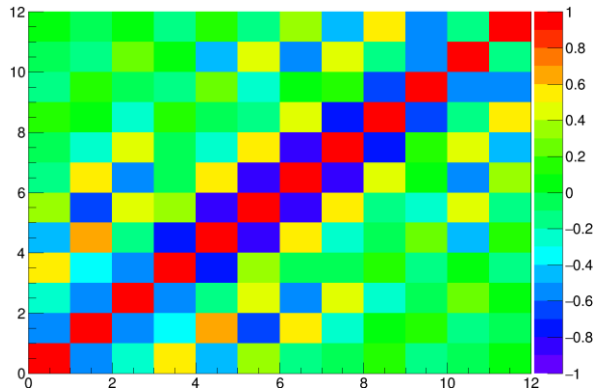

2000 iterations (~unreg)
$\chi^2_{truth} = 2.9, \chi^2_{input} = 134$

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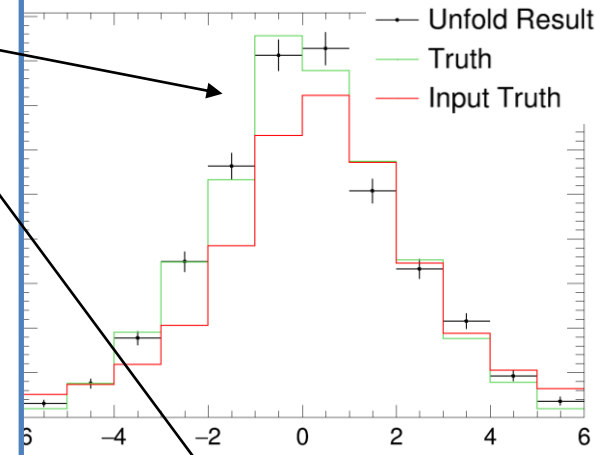$\chi^2_{truth} = 32, \chi^2_{input} = 254$

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

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

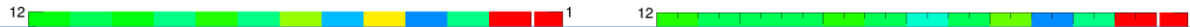- Early termination of D'Agostini can give unrealistically small errors

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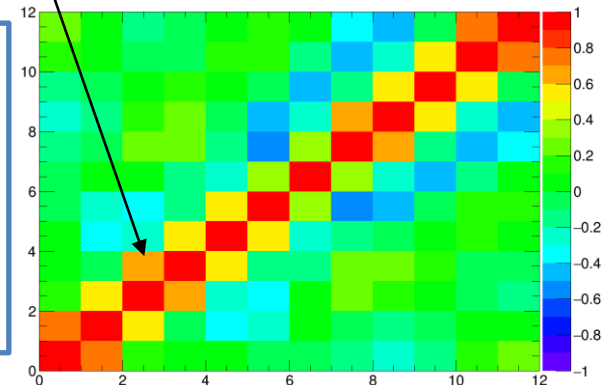

2000 iterations (~unreg)
$\chi^2_{truth} = 2.9, \chi^2_{input} = 134$

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.

# How many iterations? – Choose via data

- To mitigate this issue two recent T2K analyses utilising D'Agostini's method employ a **data-driven** regularisation

- Can think of as similar approach to L-Curve:
  - Balance the impact of the smoothing (Y)
  - With the <u>distance</u> to the unregularised result (X)

$$\chi_n^2 = \sum_{bin\ i,j} (N_i^n - N_i^\infty)(Cov^n)^{-1}(N_j^n - N_j^\infty)$$

(Use some suitably large number in place of ∞)

n – num. iterations Cov – covariance matrix N – num. unfolded signal events

Too many iterations

About right

Too few

χ_n^2
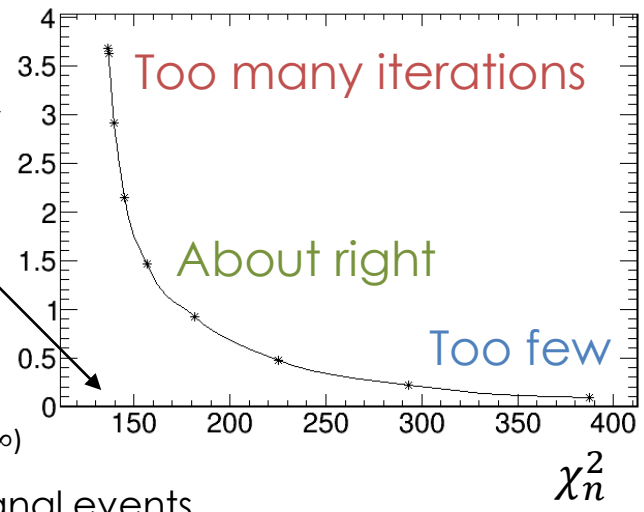
- Actual implementation is a little different (see main slides)

Case Study: On-axis CC1π measurement

- Number of iterations chosen via fake data: **3 – 7**

- Number of interactions chosen via data: **16**

See talk from Benjamin Quilain at NuInt2018

What if my L-curve isn't L-shaped?

Is the condition on the Y-axis reasonable?

- If the form of the penalty pushes the result somewhere that is incompatible with the no regularisation case, the drop on the y-axis can be limited to very small values of p_{reg}
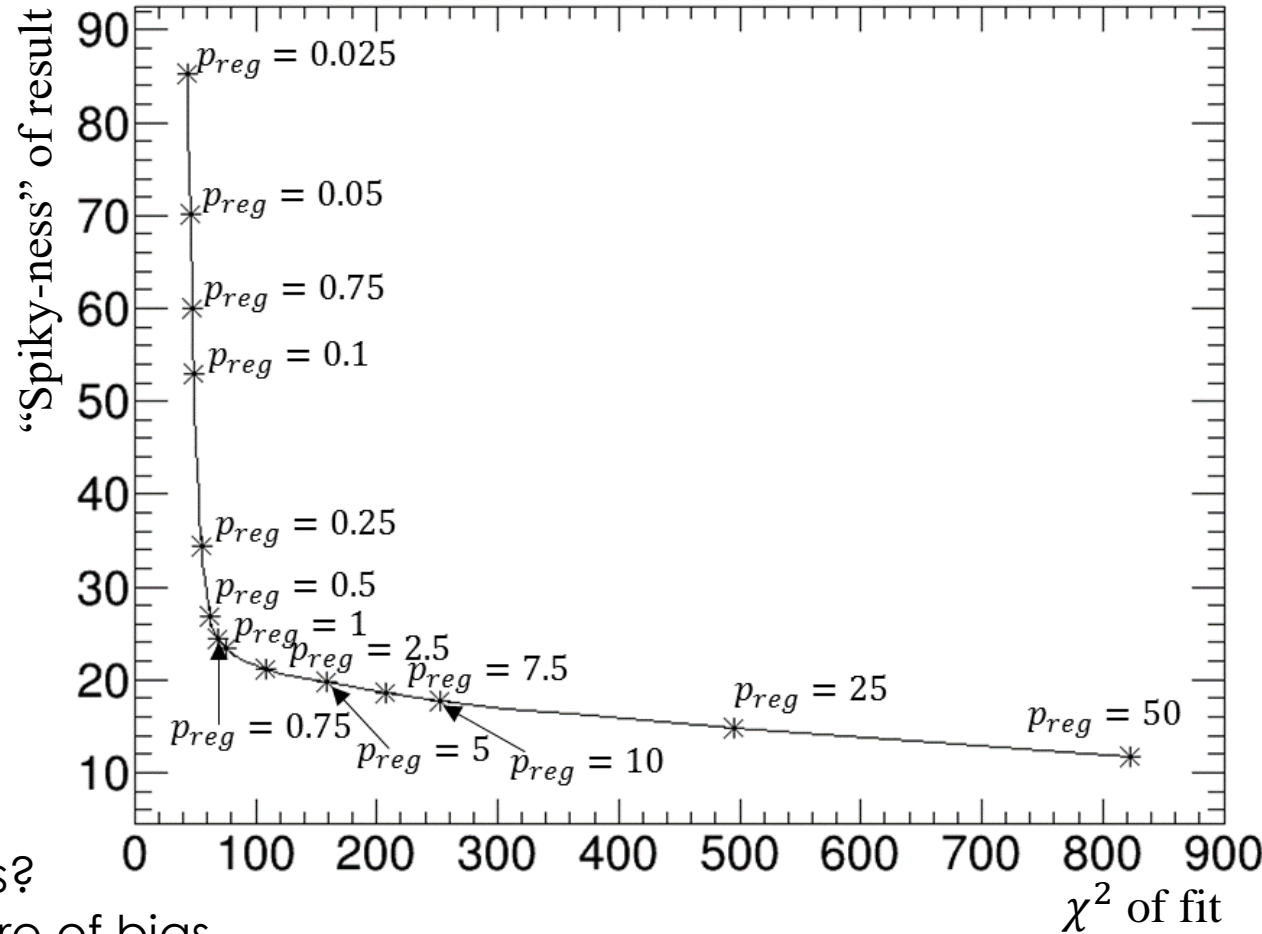

How did you form the x-axis?
- This needs to be a measure of bias.

What is the first value on the x-axis?
- How does this compare with the x-axis value of the unregulairsed result? If they're very different consider smaller p_{reg}

Aesthetic regularisation

- A result with a carefully chosen regularisation strength shouldn't significantly alter the physics conclusions with respect to the unregularised case – it's just aesthetic.
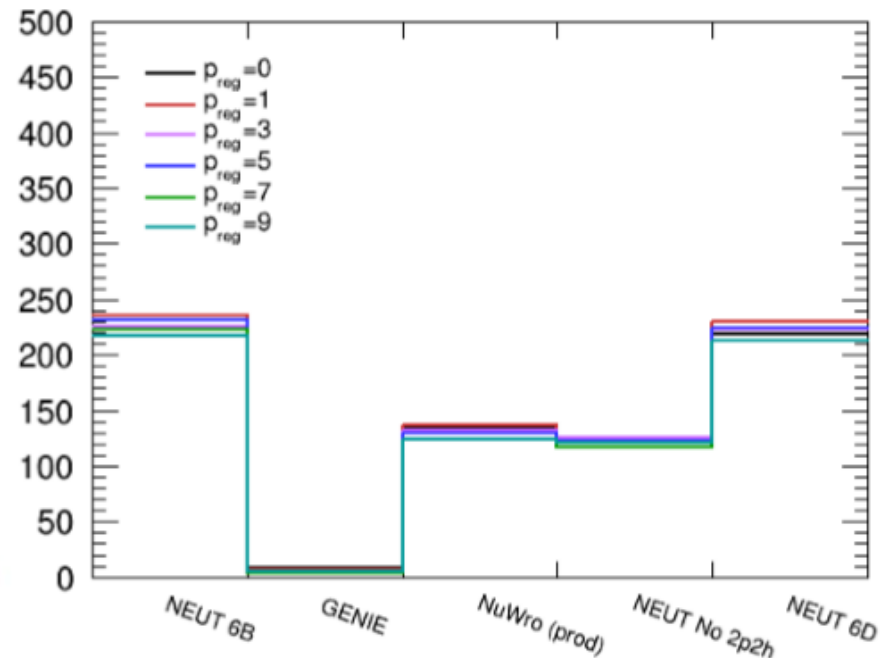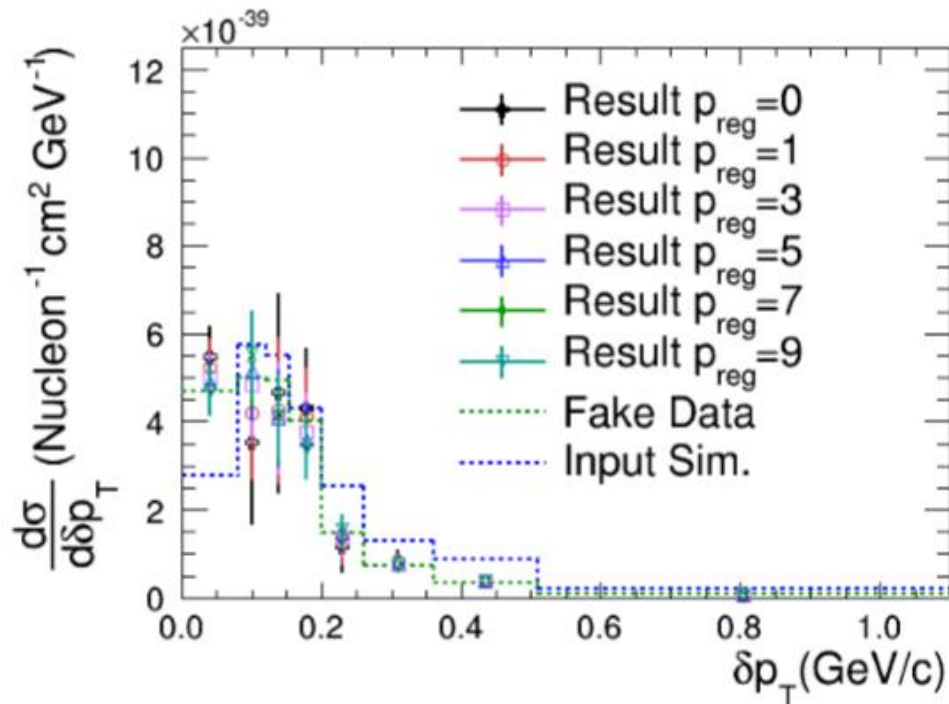
Unregularised result as a reference for regularised result bias

TABLE IX. The full and shape-only χ^2 comparisons to the δp_T result with nominal and no regularization. The table is ordered by the size of the no-regularization shape-only χ^2. More details of these models can be found in Sec. IV A.

Generator	Full		Shape Only	
	No Reg.	Nom. Reg.	No Reg.	Nom. Reg.
NEUT 5.4.0 ($LFG_N + 2p2h_N$)	31.6	30.4	3.38	2.60
NEUT 5.3.2.2 ($SF + 2p2h_N + 2 \times FSI$)	15.9	14.8	11.0	10.1
NEUT 5.3.2.2 ($SF + 2p2h_N$)	31.9	30.3	16.6	15.5
NuWro 11q ($SF + 2p2h_N$)	22.6	23.1	16.8	15.6
NuWro 11q ($LFG + 2p2h_N$)	81.5	81.7	39.0	15.6
NuWro 11q ($LFG + RPA + 2p2h_N$)	78.5	84.4	39.9	36.3
NEUT 5.3.2.2 ($SF + 2p2h_N + No\ FSI$)	114	112	42.9	41.4
GENIE 2.12.4 ($RFG + 2p2h_E$)	92.9	92.4	47.9	47.7
NuWro 11q ($SF\ w/o\ 2p2h$)	65.8	68.7	55.4	54.8
NEUT 5.3.2.2 ($SF\ w/o\ 2p2h$)	93.3	91.5	61.2	59.6
GiBUU 2016 ($LFG + 2p2h_G$)	77.0	78.9	66.1	59.6
NuWro 11q ($RFG + 2p2h_N$)	150	155	67.2	69.0
NuWro 11q ($RFG + RPA + 2p2h_N$)	155	172	68.6	70.4
GENIE 2.12.4 ($RFG\ w/o\ 2p2h$)	94.6	97.8	74.1	76.2

- These numbers are very similar → No change of physics conclusions form regularisation. Important test.
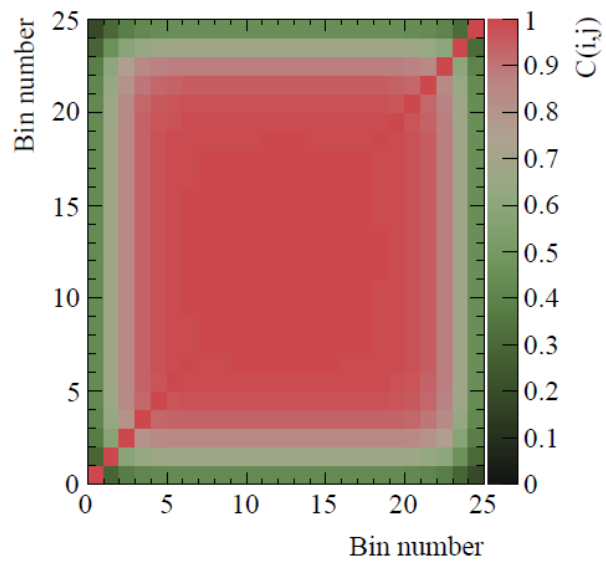
Phys. Rev. D **98**, 032003

Do I really need a covariance?

Example by L. Pickering

Guess which MC fits each data better better?

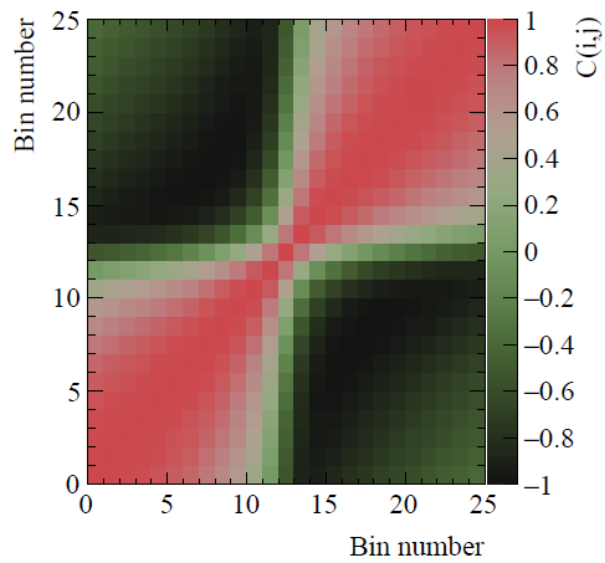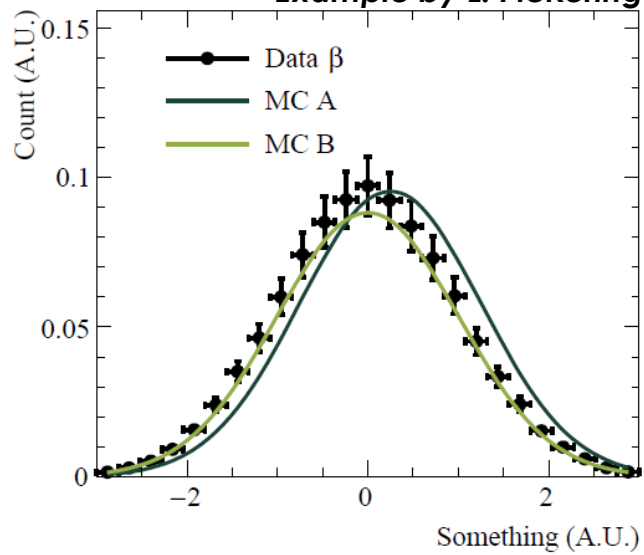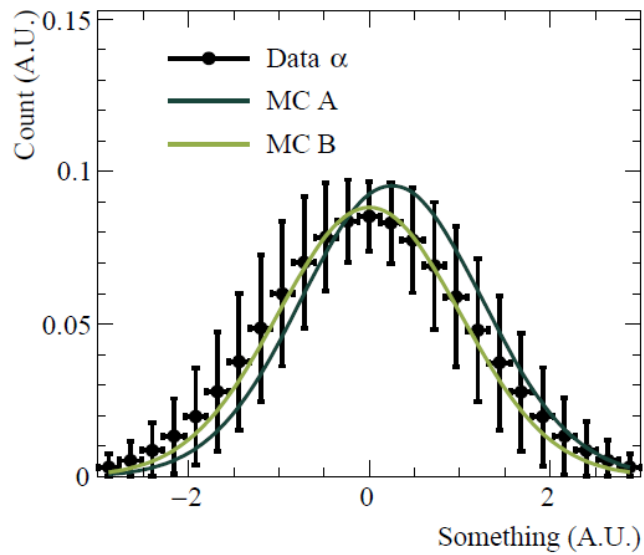# Do I really need a covariance?
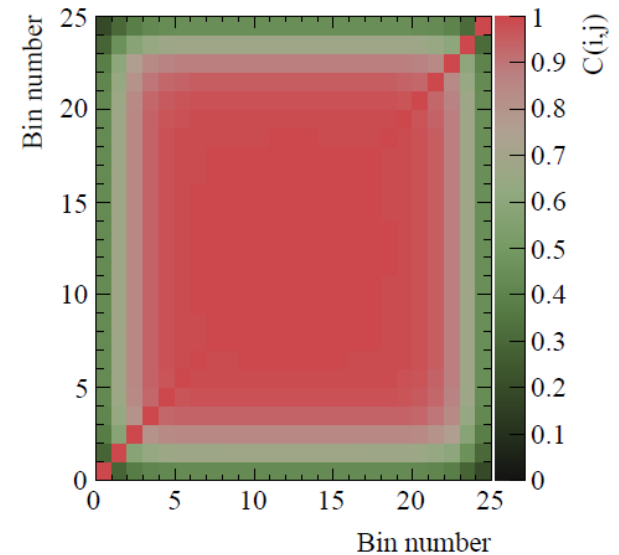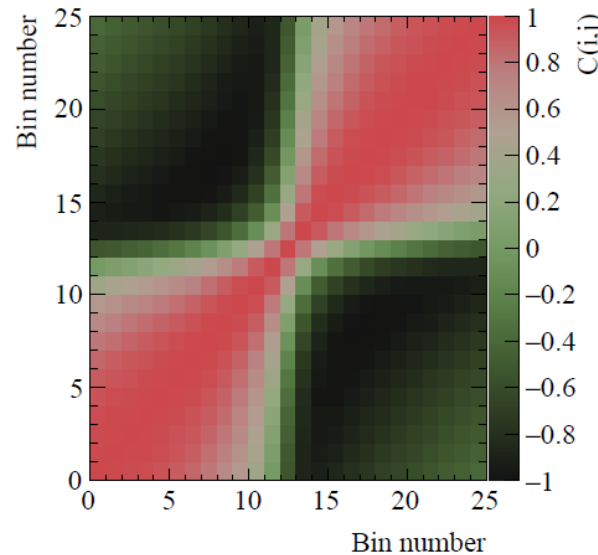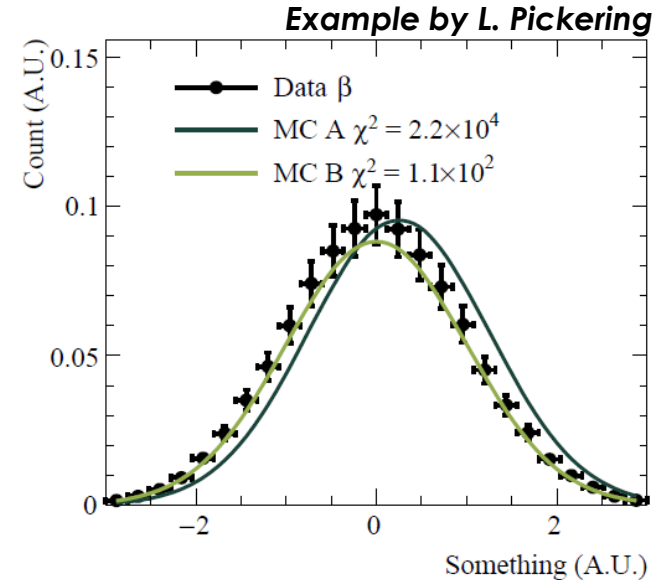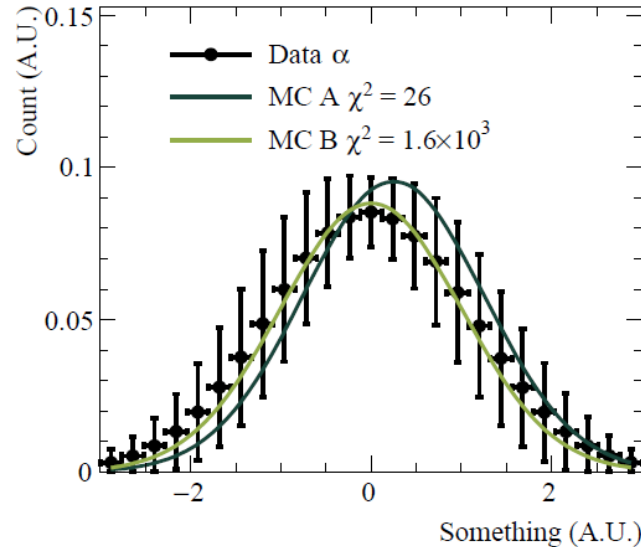
How about now?

Can you do chi-by-eye?

Chi-by-eye?

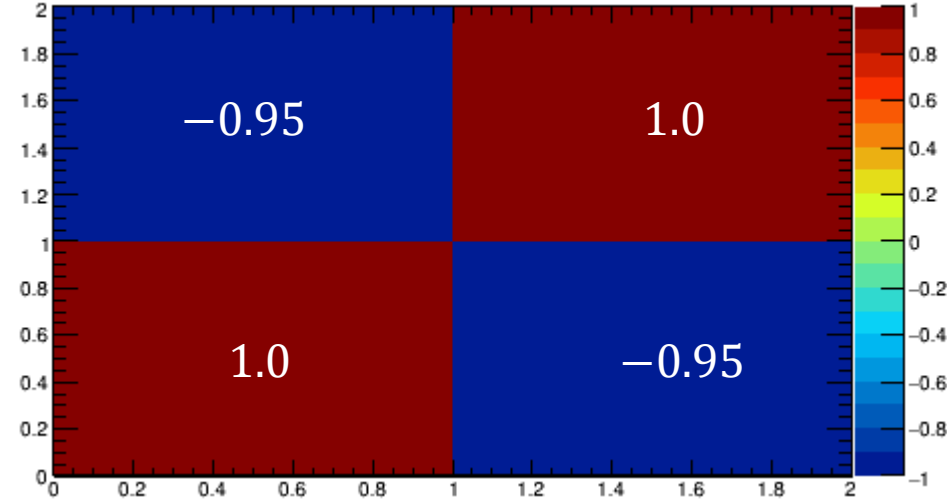Interpreting any result-simulation comparison without a covariance matrix and a goodness of fit is dangerous.
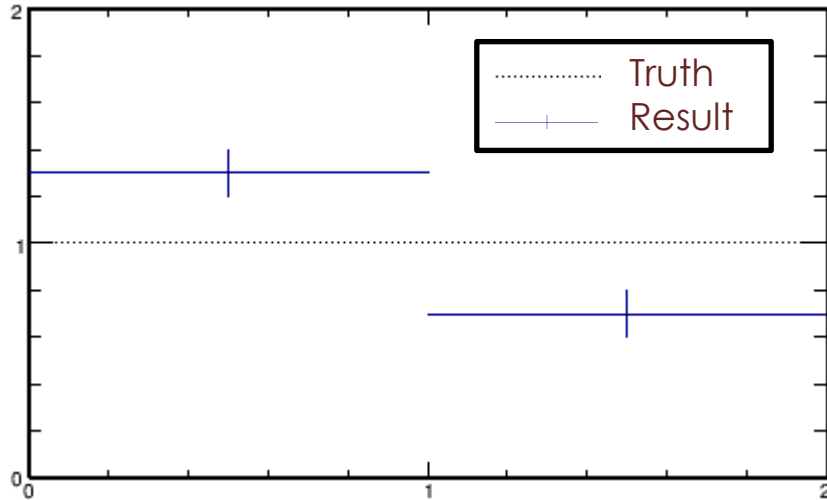
If you really must, then the regularised result is better, but may still be misleading.

If you make a conclusion by eye, check the χ^2 tell the same story.

But the result looks awful!?

- Consider a two bin result:

$$pull_i = \frac{N_{fit} - N_{true}}{Error}$$
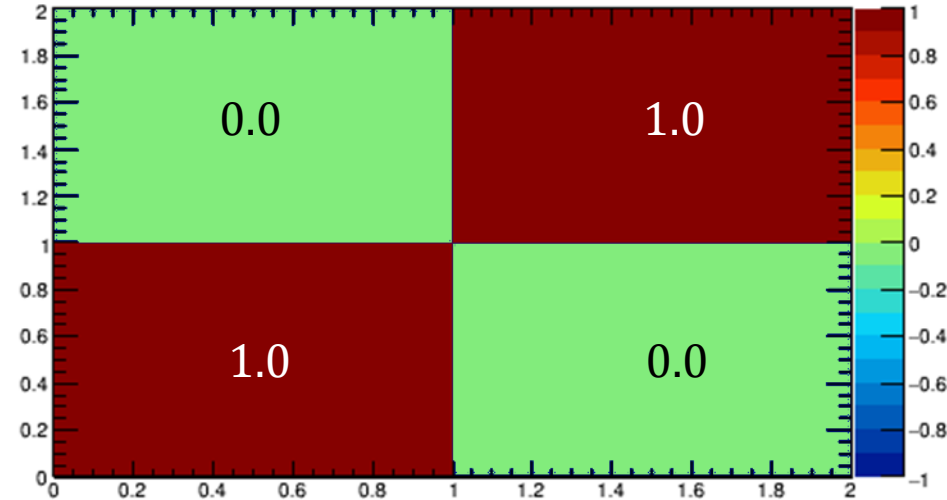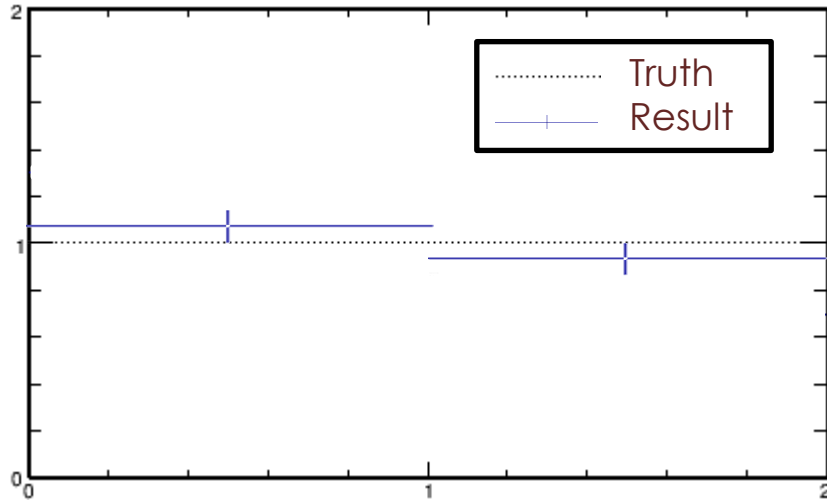
$pull_0 = 3$ ⎱ **Fairly awful**
$pull_1 = 3$ ⎰ **pull**

$$\chi^2 = (\overline{N_{fit}} - \overline{N_{true}})(V_{cov})^{-1}(\overline{N_{fit}} - \overline{N_{true}})$$

$\chi^2 = 1.69$ ⎱ **Good χ^2**

- Need to see the correlation matrix to tell whether the result is good or not.

But the result looks awful!?

- Consider a two bin result:

$$pull_i = \frac{N_{fit} - N_{true}}{Error}$$

$pull_0 = 1$
$pull_1 = 1$ } **Better pull**

$$\chi^2 = \left(\overline{N_{fit}} - \overline{N_{true}}\right)(V_{cov})^{-1}\left(\overline{N_{fit}} - \overline{N_{true}}\right)$$
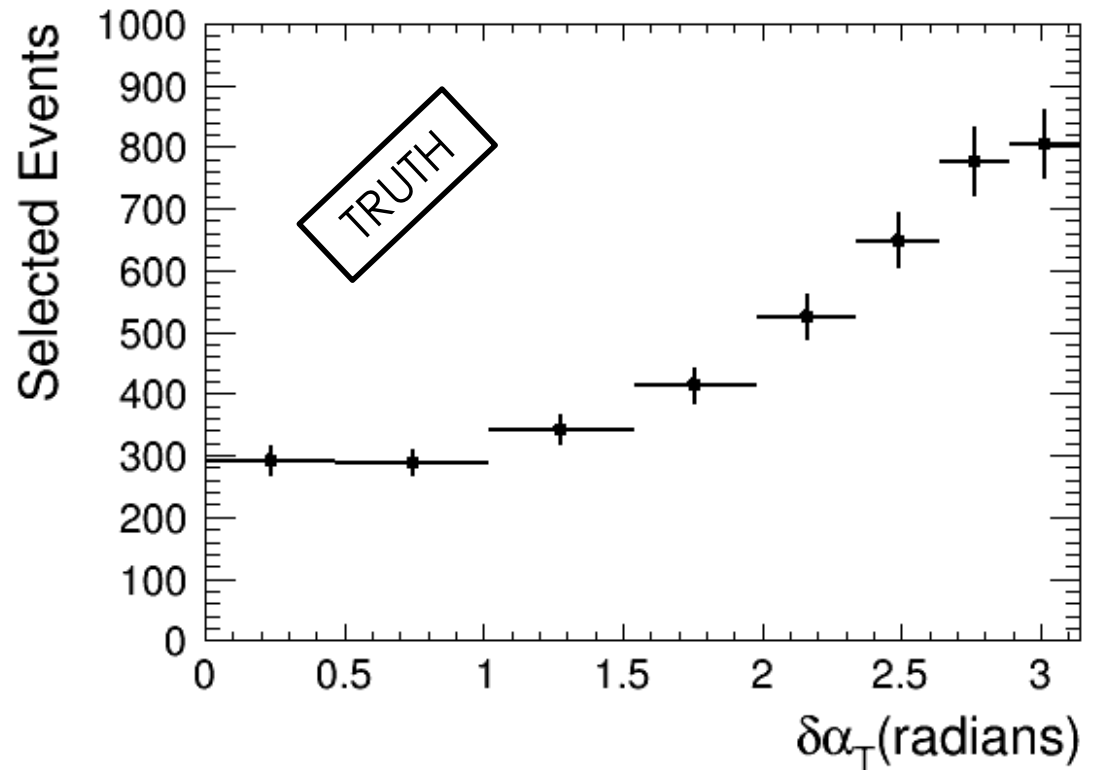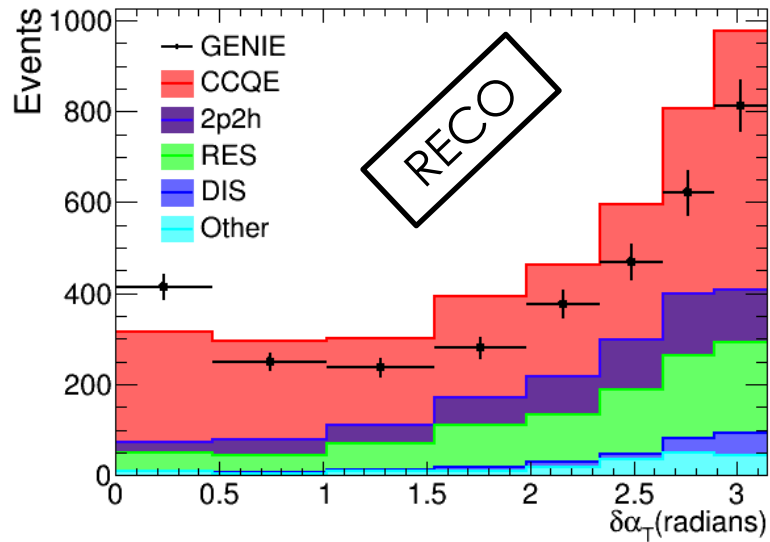
$\chi^2 = 2.0$ } **Worse** χ^2

- Pulls/bin-to-bin bias doesn't tell the whole story
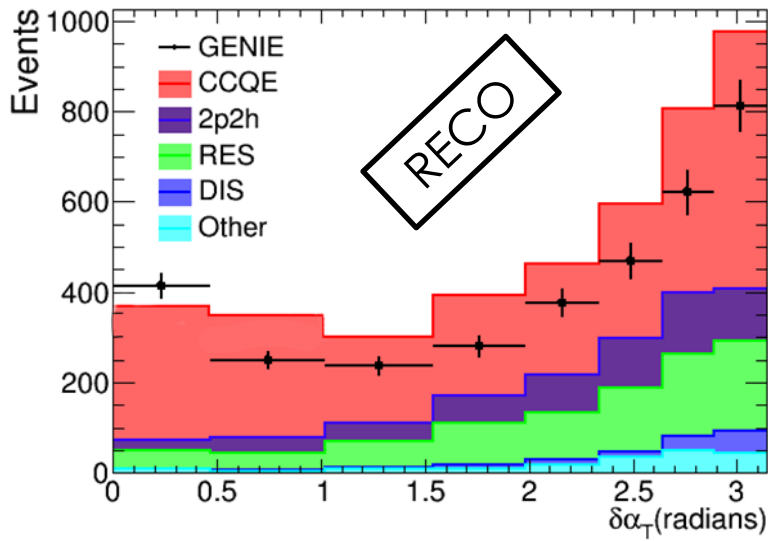
Unfolding at T2K: likelihood fitting

- True bin → Reco. template

- Vary MC template norms (c_i) and compare to data

- Maximise Poisson likelihood + syst. penalty term
  (using max. gradient decent)
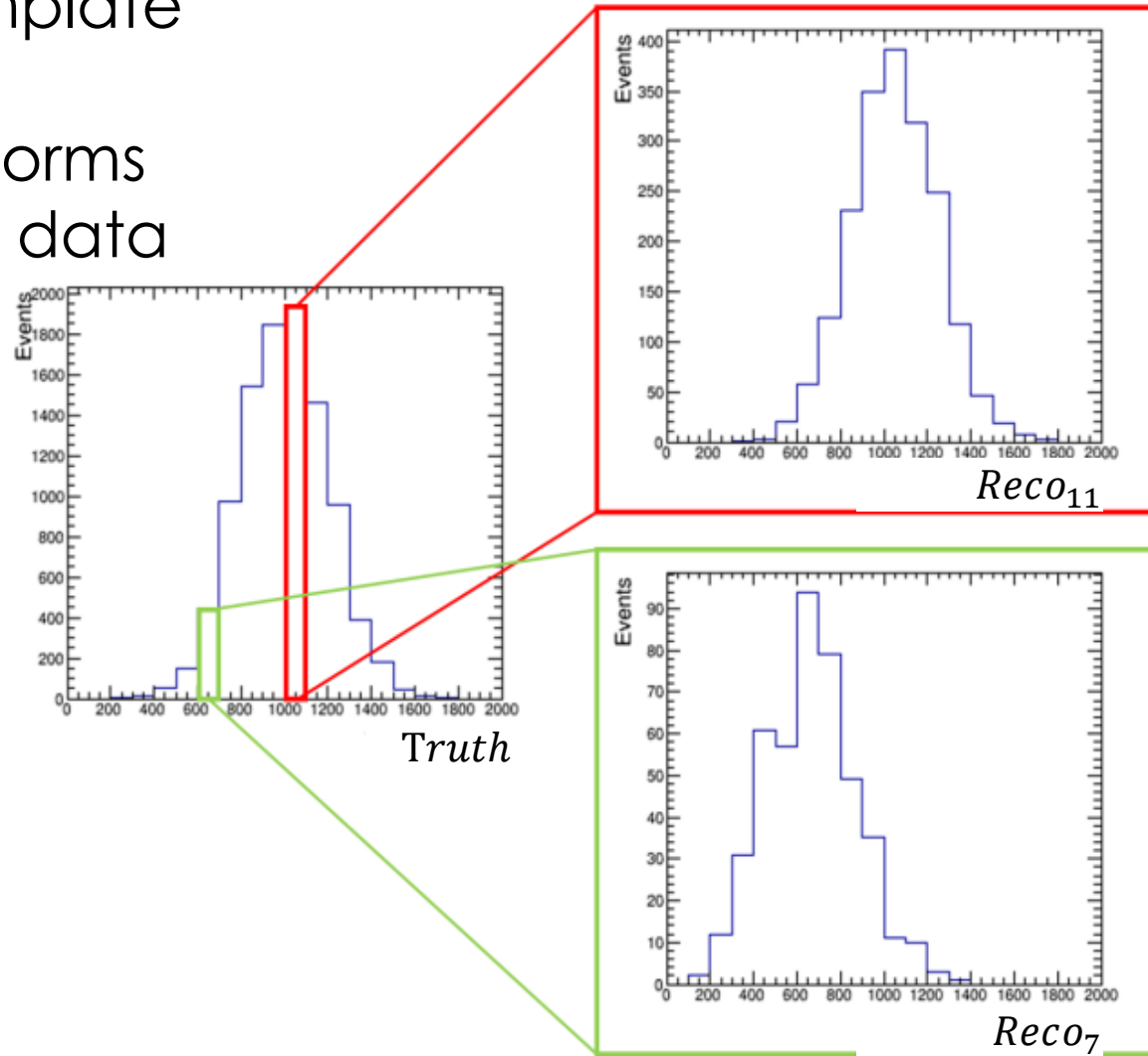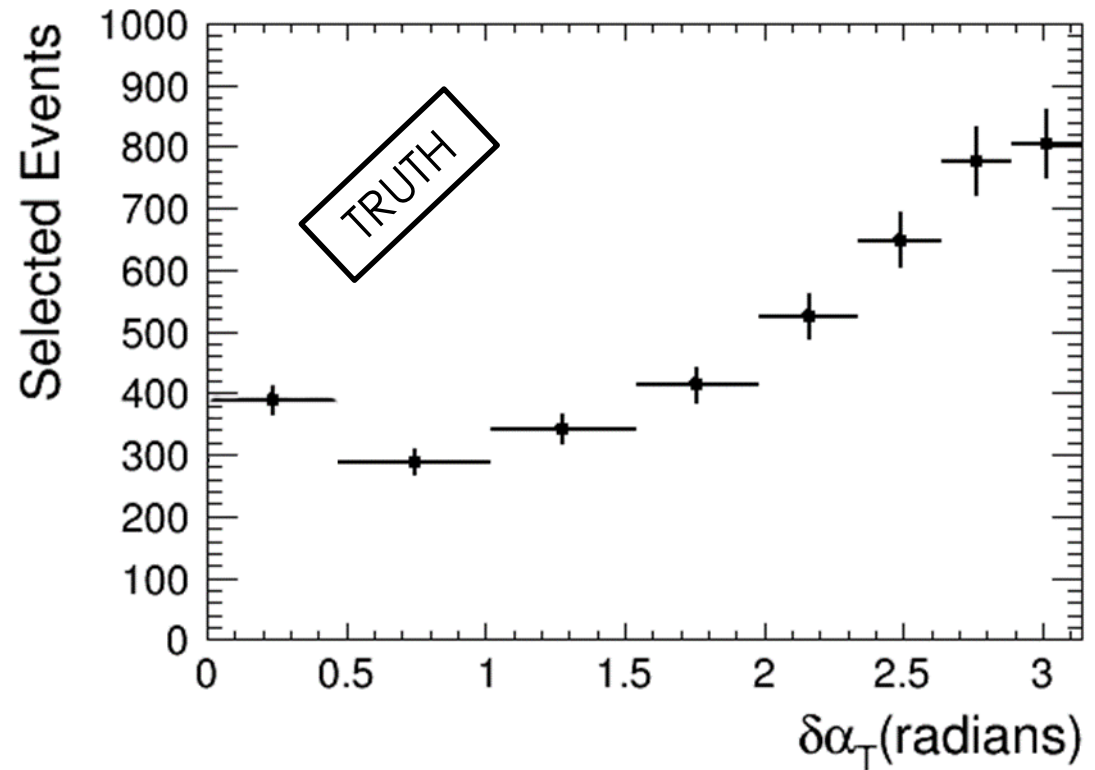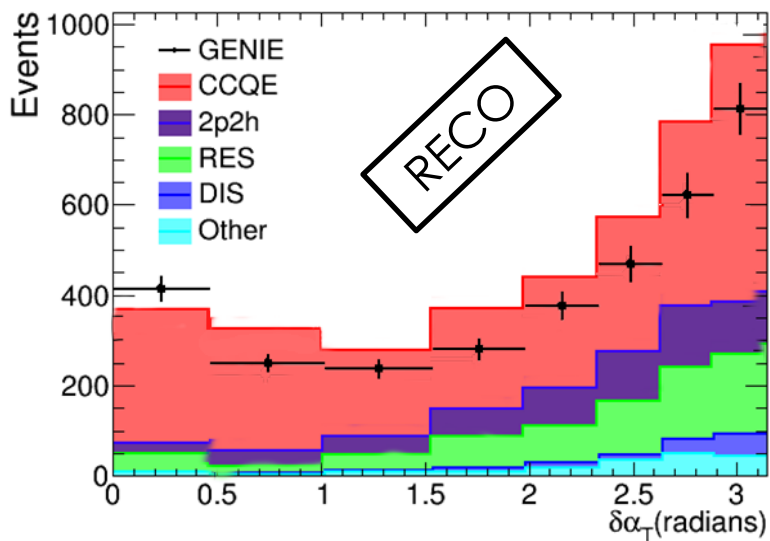
How does it work?

How does it work?
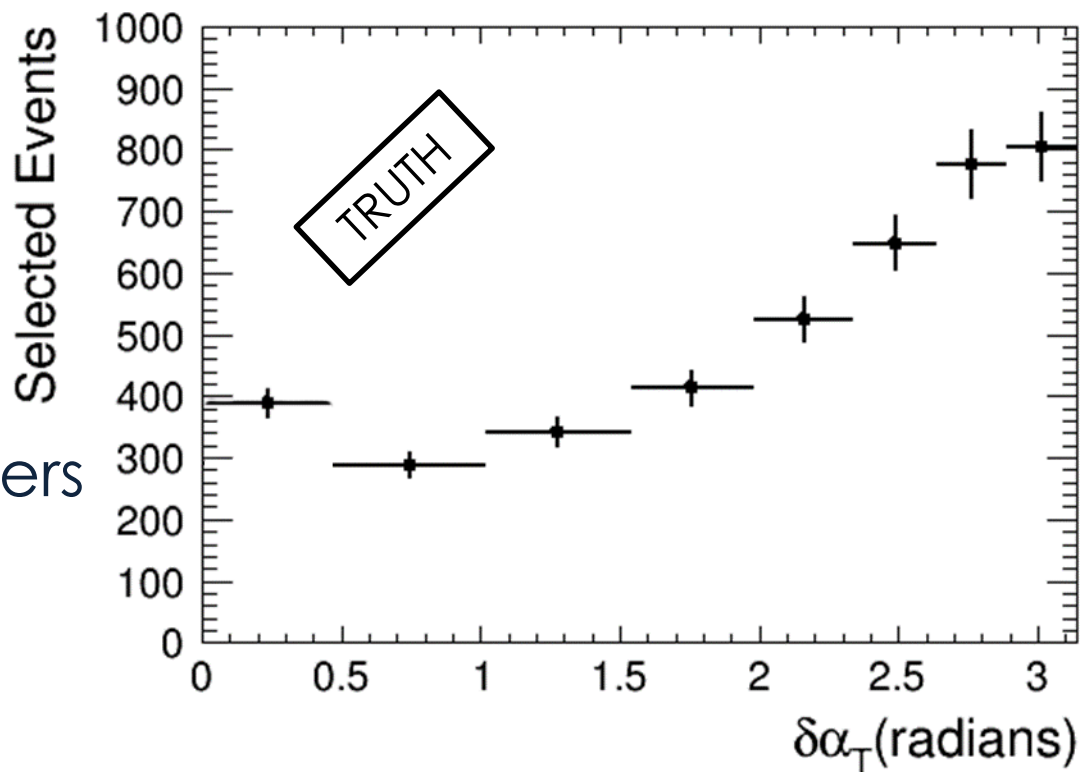

- Scale template weights

How does it work?

- Alter background systematic parameters

- These should ideally be constrainable by control regions

- Overall can alter:
  - Template weights
  - BG Model parameters
  - Flux
  - Detector response

How does it work?

- Keep iterating to maximize the likelihood
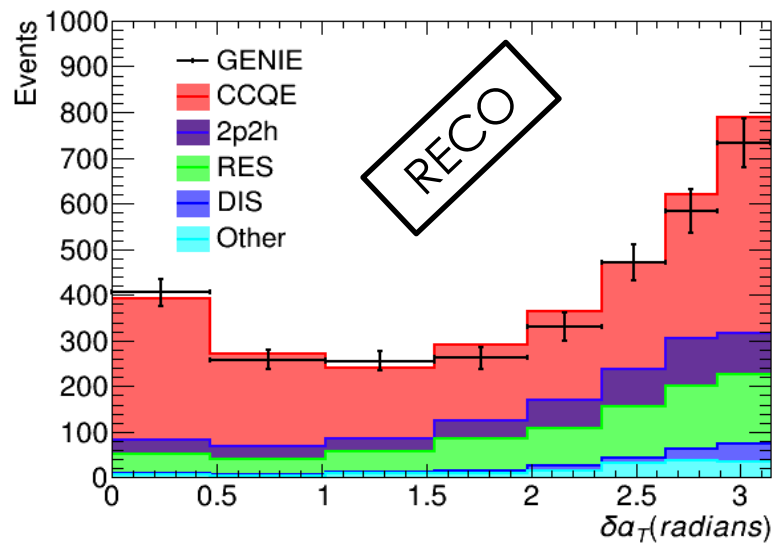
- Maximise likelihood / minimise $-2\ln(L) \approx \chi^2$
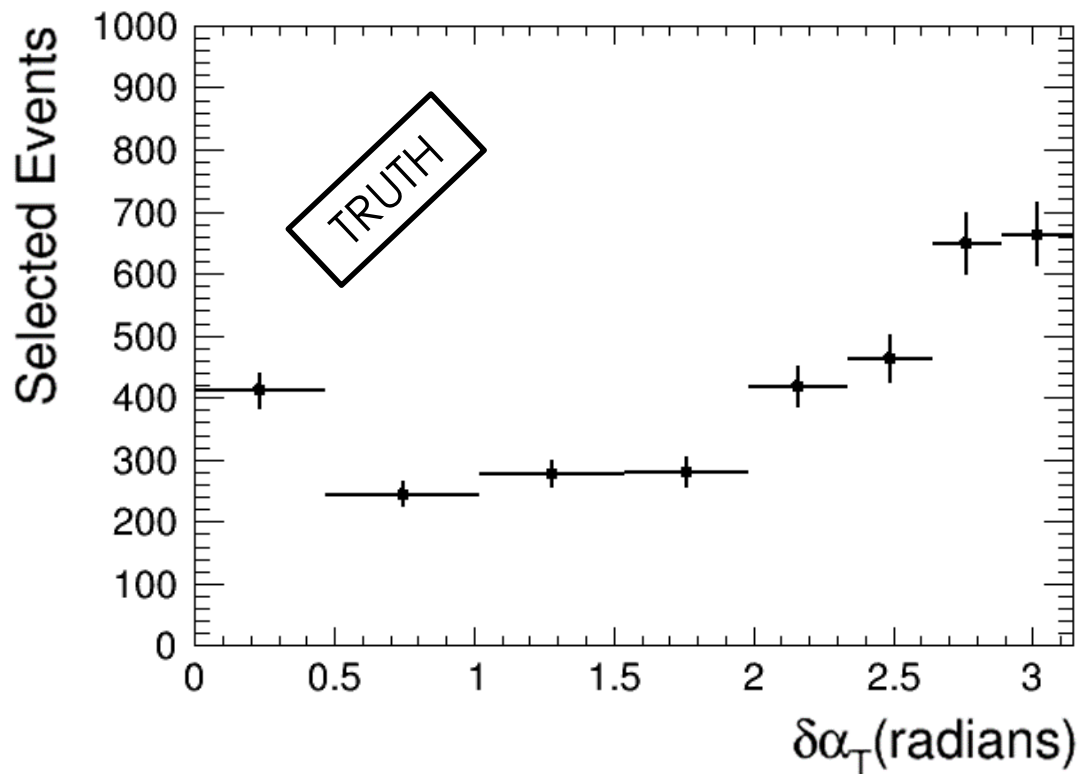
Regularisation in the likelihood fitter

- The best fit parameters are those that minimise the following:

$$\chi^2 = \boxed{\chi^2_{stat(fit\ goodness)}} + \boxed{\chi^2_{syst(penalty)}} + \boxed{\chi^2_{reg}}$$

$$\chi^2_{stat} = \sum_j^{recobins} 2\left(N_j^{MC} - N_j^{obs} + N_j^{obs} ln\frac{N_j^{obs}}{N_j^{MC}}\right)$$

$$\chi^2_{syst} = (\vec{a}^{syst} - \vec{a}^{syst}_{prior})(V_{cov}^{syst})^{-1}(\vec{a}^{syst} - \vec{a}^{syst}_{prior})$$

- With an optional regularisation term (other terms are possible, exact choice of term is beyond the scope of this talk):

$$\chi^2_{reg} = p_{reg} \sum_i (c_i - c_{i-1})^2 = p_{reg}(\boldsymbol{c} - \boldsymbol{c}_{prior})V_{cov}^{reg}(\boldsymbol{c} - \boldsymbol{c}_{prior})$$