Post talk comments

Comments

- Generally seemed positive.
- Main comments from Jake:
  - Consider dividing templates into energies too.
  - Some magic about the energy slice which might skip unfolding.
  - Potentially some confusion about MC/data discrepancies, still communicating.
- Started looking through tech note, still trying to understand the fit minimisation
- Planning to chat with Jake soon

Fitting discussion

Fitting method

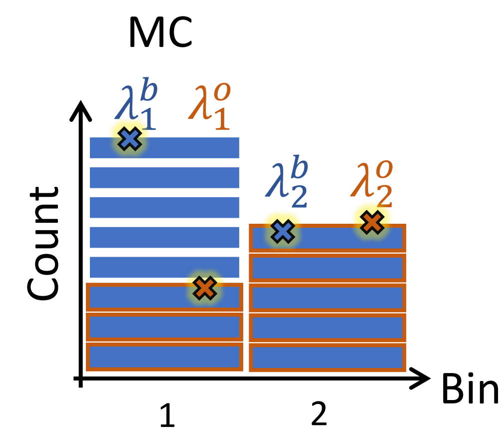
- The fit uses (python) Minuit's template fit, using Dembinski and Abdelmotteleb method.
- D. and A.'s method approximates the Beeston-Barlow method.
  - Henceforth, will discuss pure Beeston-Barlow, trusting the D. and A. method is sensible
- Methods can also deal with weighting the MC templates (no longer integer)
  - Currently only considering unweighted templates

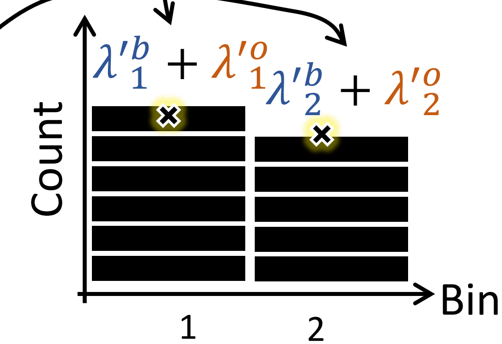
Fitting method

- Example fit – 2 bins, 2 channels
- MC sample has counts (8, 5)[b], (3, 5)[o].
- Data has counts (6, 5)
- From MC, create $\lambda_1^b, \lambda_2^b, \lambda_1^o, \lambda_2^o$
  - Note: $\lambda_2^{b/o} = N^{MC} - \lambda_1^{b/o}$

Eqs. 17 and 2 of BB

- Compare data:
  - $6 \sim \frac{N^D}{N^{MC}} P^b \lambda_1^b + \frac{N^D}{N^{MC}} P^o \lambda_1^o$
  - $5 \sim \frac{N^D}{N^{MC}} P^b \lambda_2^b + \frac{N^D}{N^{MC}} P^o \lambda_2^o$
- We want data yields $N^D P^b, N^D P^o$





Code

One data histogram to be fit (for multiple data histograms, combine multiple cost_func instances).
Shape: (N_e, N_b, N_b, N_b)
For: N_e energy bins,
N_b score bins
3 scores considered

```
cost_func = cost.Template(
            d_hist,
            generator.bin_edges,
            templates,
            name=generator.labels)
```

Histogram bin edges:
$(N_e + 1,) + (N_b + 1,) * 3$

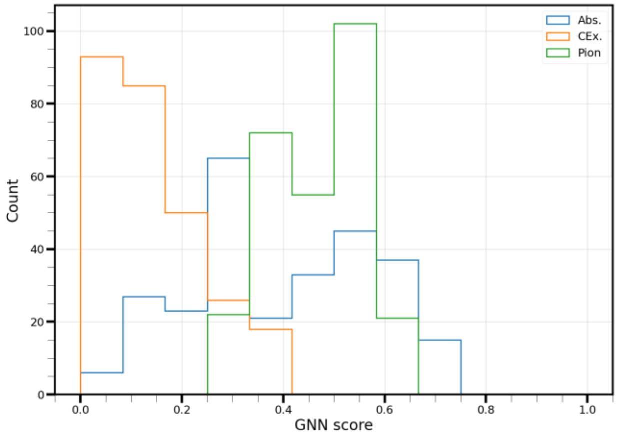
Labels for ID,
$N^{\text{labels}} = N^{\text{temps}}$

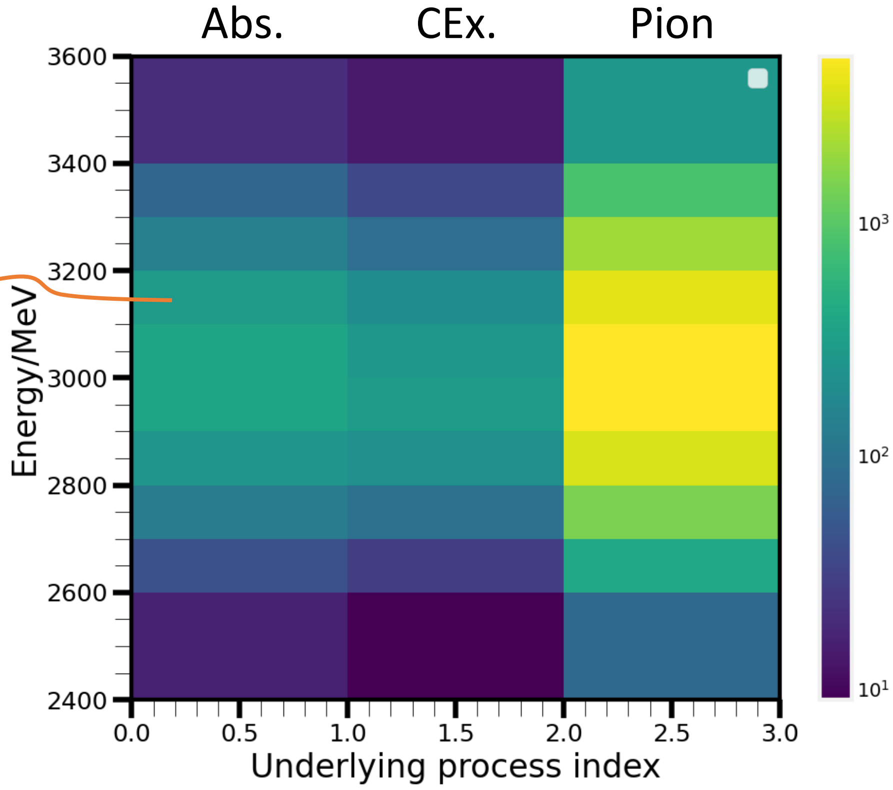
List of $\boldsymbol{N^{\text{temps}}}$ **histograms as templates**. There will be N^{temps} yields given by the fit, one for each templates
Shape: $[(N_e, N_b, N_b, N_b)] * N^{\text{temps}}$
(Each template has the same shape as the data histogram, but there are N^{temps} in the list)

Fitting options

- 2D histogram displays the total count of events as a function of energy and underlying process.
- Each of these points contains one (N_b, N_b, N_b) histogram.

Distribution (template) from this bin

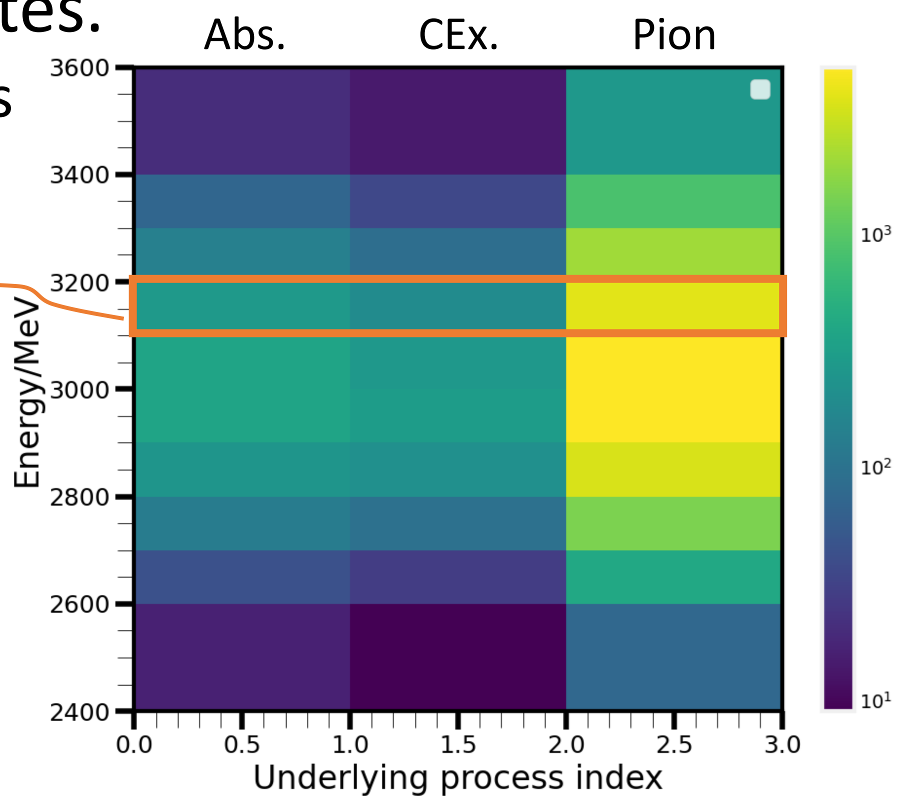




Fitting options – current

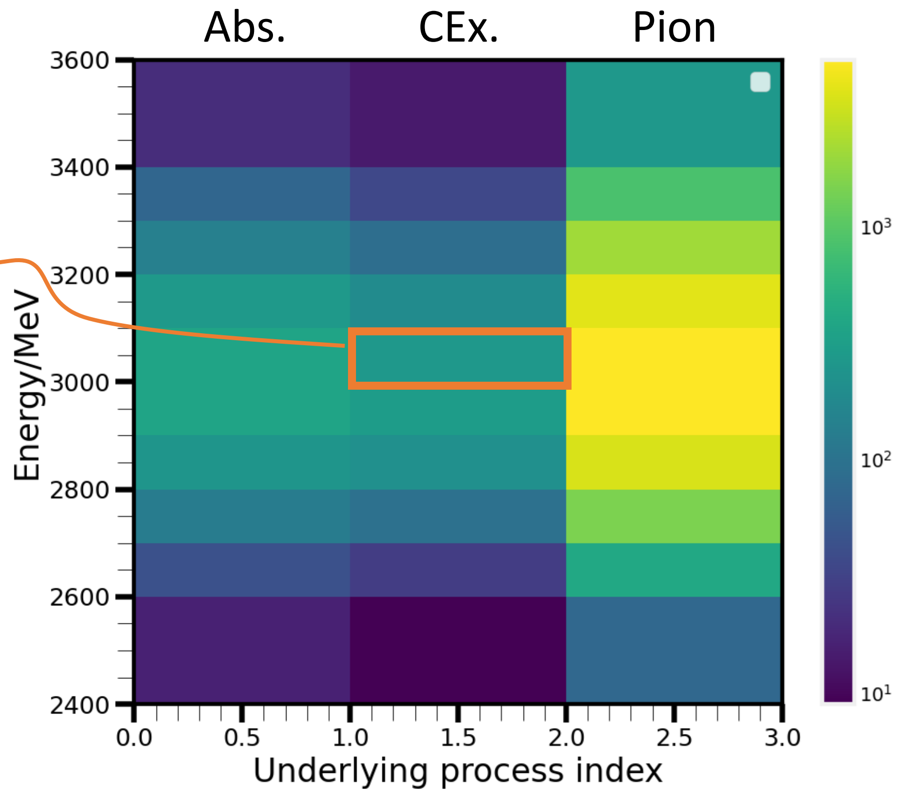
- Current idea, do N_e separate fits, each to one data histogram, shape (N_b, N_b, N_b).
- For each bin, get 3 templates.
  - For each bin, the templates are the three on the corresponding row of this histogram.

1 set of fit templates is one row of this histogram.
Each row of the histogram is fitted to unique, non-overlapping data histograms



Fitting options – free-for-all

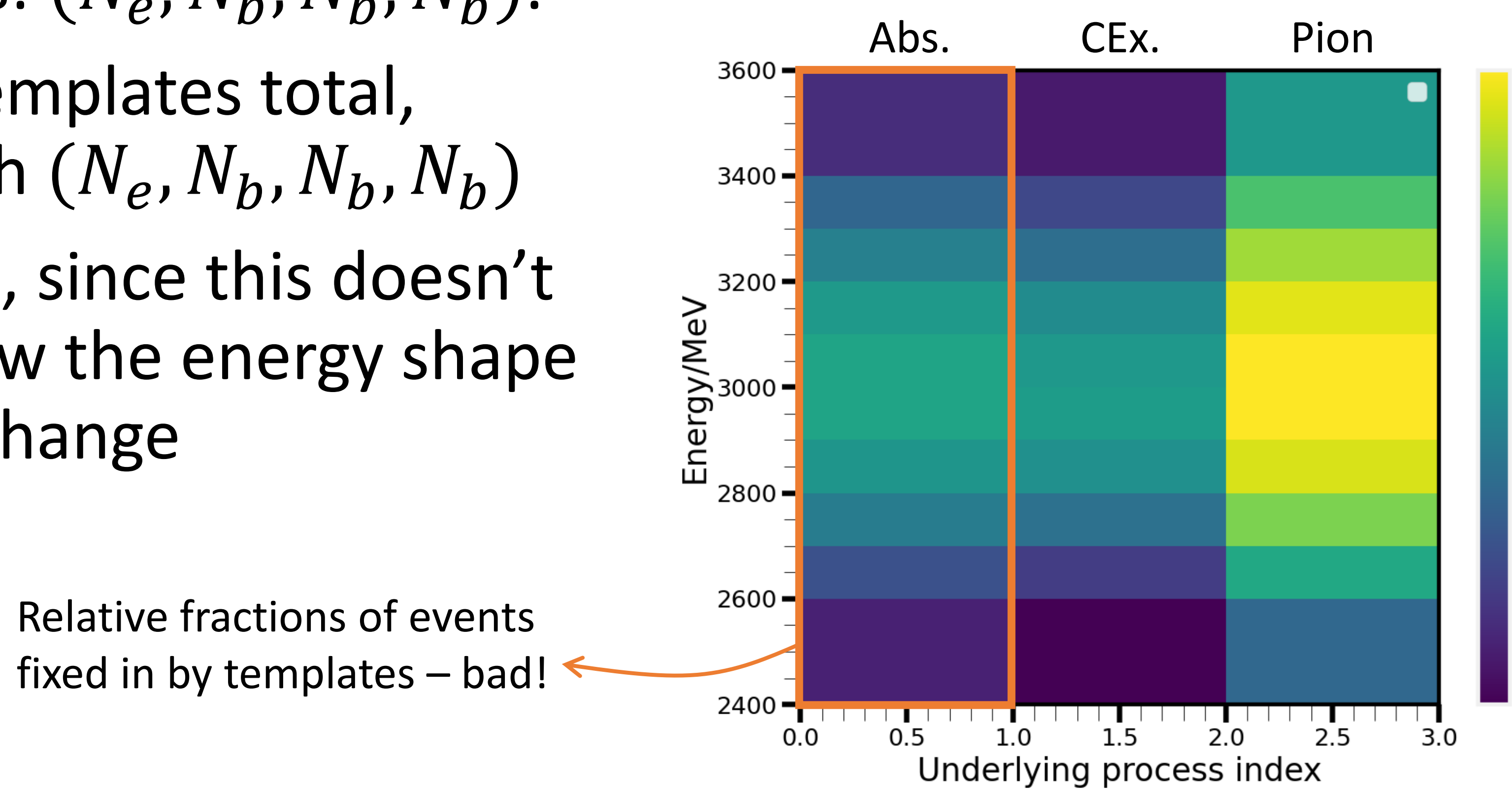
- A valid (but poor) fitting option would be to do one fit to all data (N_b, N_b, N_b), where each energy and process gets its own template.
- $3 \times N_e$ templates total, each (N_b, N_b, N_b)



Each point is passed as a (N_b, N_b, N_b) template.
Fit to one data histogram which includes all energies.
Fit predicts a count for each template.

Fitting options – energy fixed

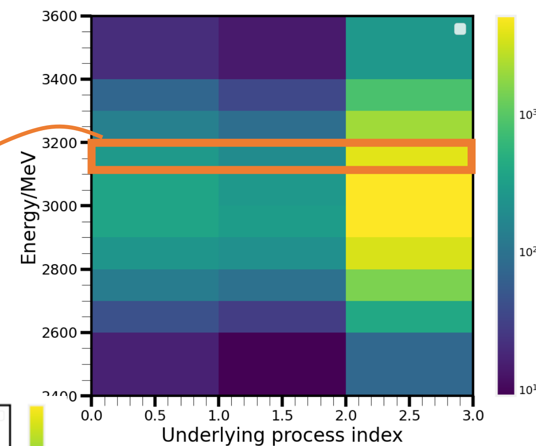
- An attempt at simultaneous energy fitting could use one data histogram, which includes energy bins: (N_e, N_b, N_b, N_b).
- 3 templates total, each (N_e, N_b, N_b, N_b)
- Bad, since this doesn't allow the energy shape to change

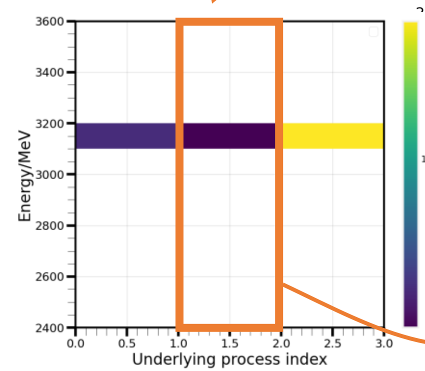


Fitting options – energy binned

- Use one data histogram, including energy bins: (N_e, N_b, N_b, N_b). But separate templates for each energy bin.
- $3 \times N_e$ templates total, each (N_e, N_b, N_b, N_b)
- Each template has non-zero values in exactly one of the indices across the first dimension (N_e).

Extract template binned in energy, but with only one non-zero bin.



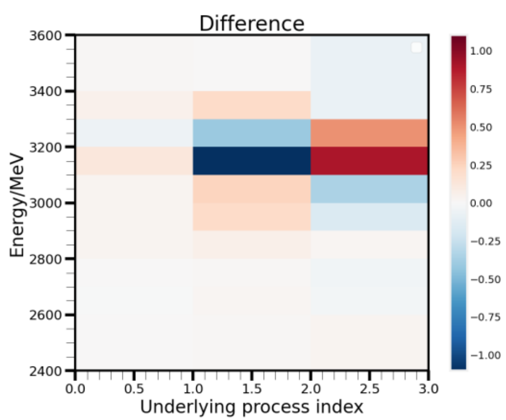


Each process is then given as its own template, for $3 \times N_e$ total.

Energy binned vs. separate fits

- Use 50% MC as template, 50% as “data”.
  - Not done any energy weighting.
- Performed current fit (separate fits for each energy)
- Perform the energy binned (final option mentioned).
- Investigated the difference between the two:

Current – E. binned

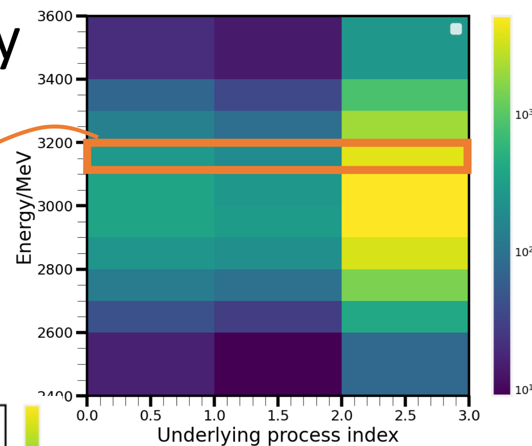


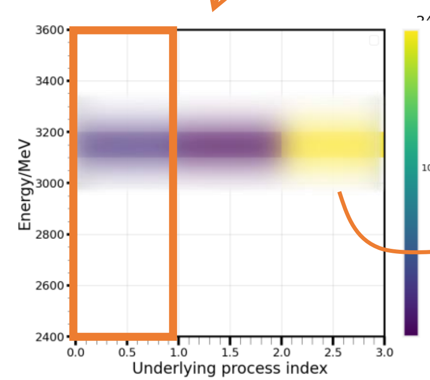
Current / E. binned - 1

```
[[ 3.62256938e-05  2.47275383e-03  2.55603376e-04]
 [-1.63322683e-04  1.58220131e-03 -7.34721345e-05]
 [ 3.74767953e-05  3.63326015e-04 -2.24963122e-05]
 [ 9.47627484e-05  3.42696054e-04  5.75051774e-06]
 [ 8.28801557e-05  7.26401418e-04 -2.93631169e-05]
 [ 1.00984362e-04  1.45238574e-03 -6.34326830e-05]
 [ 4.74770175e-04 -8.51413460e-03  2.29216628e-04]
 [-4.59448680e-04 -6.49467485e-03  2.36643514e-04]
 [ 9.60154333e-04  2.06738156e-03 -9.37032226e-05]
 [ 4.94567951e-04  3.48525444e+01 -2.39613045e-04]]
```

Other options – energy unfolding

- In the energy fitting method, templates are picked by the same binning as the y-axis
  - In this case, beam instrumentation energy rather than interaction energy.
- The histogram could be produced from MC truth interaction energy, but split into templates via reco. Interaction energy



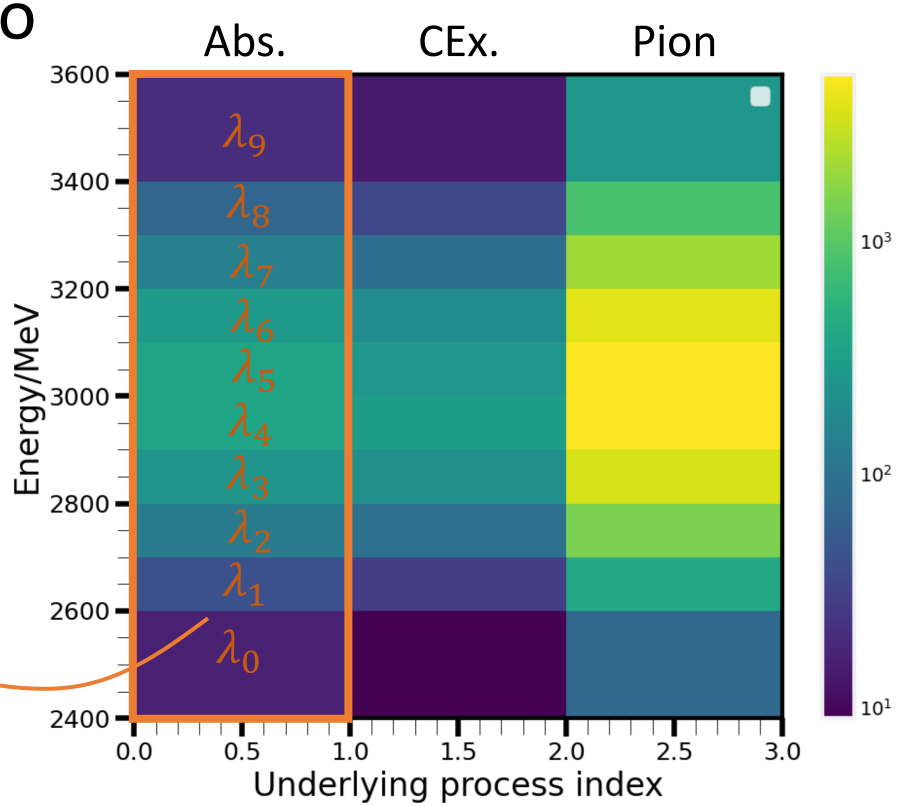


“Artistic representation” of binning in reco. energy, but selecting in MC true energy

Other options – using the nuisances

- The fit must produce nuisances per bin of the fit for each template.
- In principle, we could try to extract these and “manually” reconstruct the energy binning

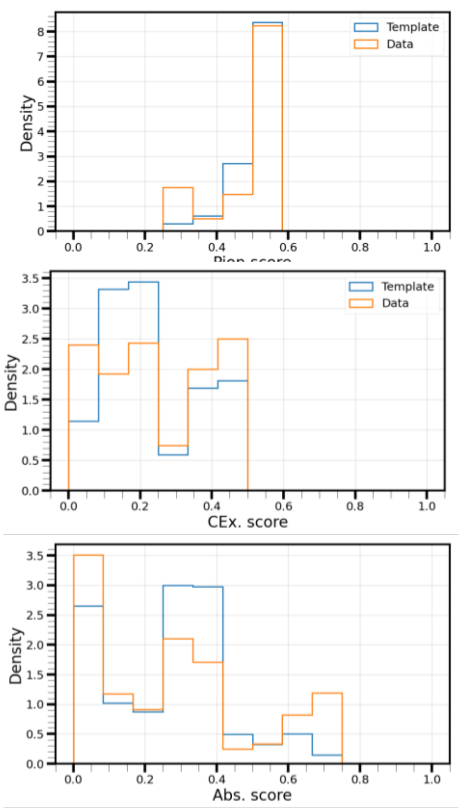
Extract nuisances to reconstruct shape. Probably possible, but definitely complicated… (e.g. correlations between overall norm and the nuisances)

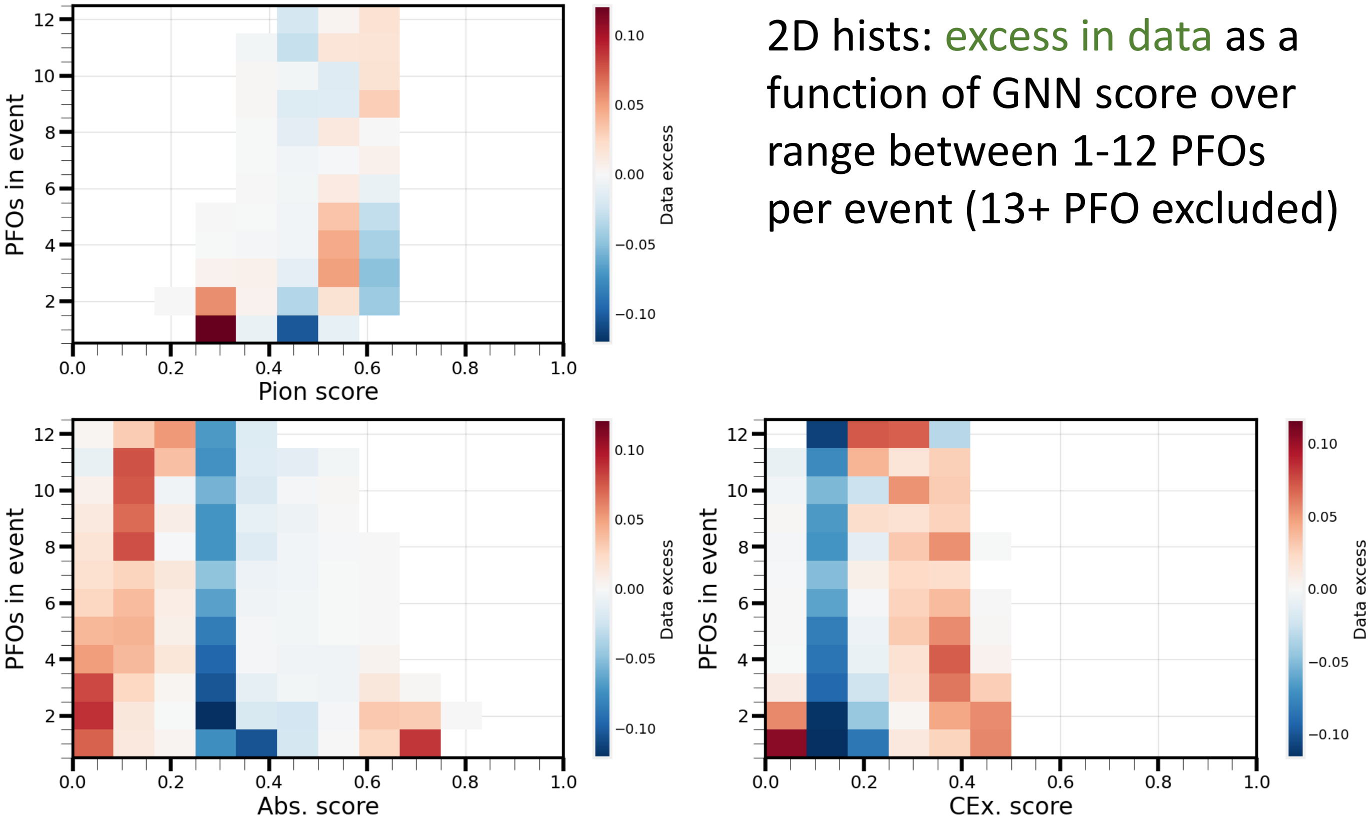


PFO count variation - comparison

- Plots compare all MC events (not split by true process) vs. data events.

1 PFO in event scores

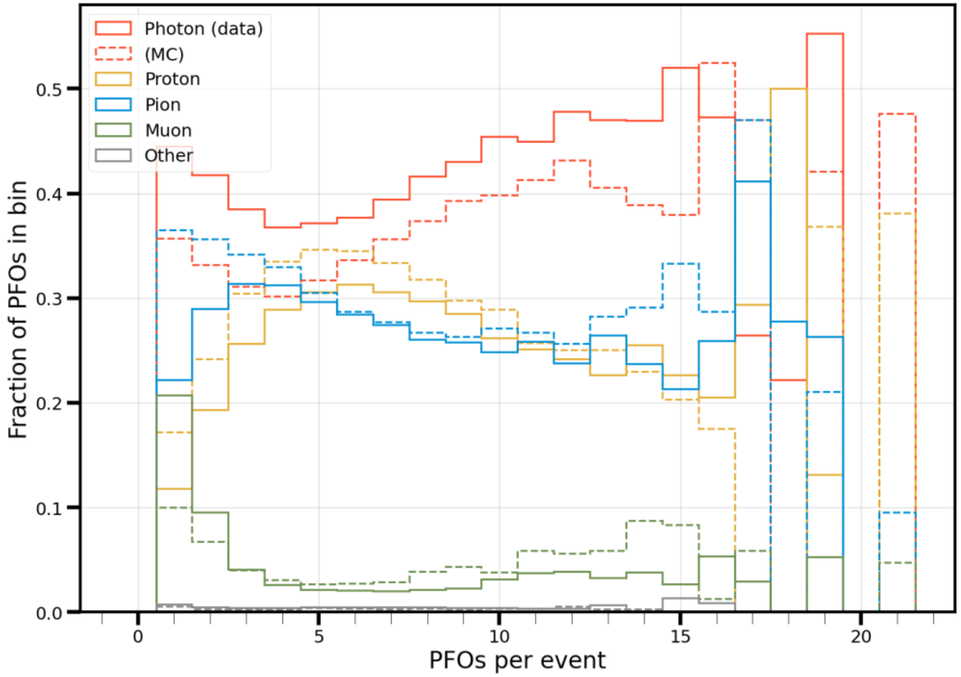




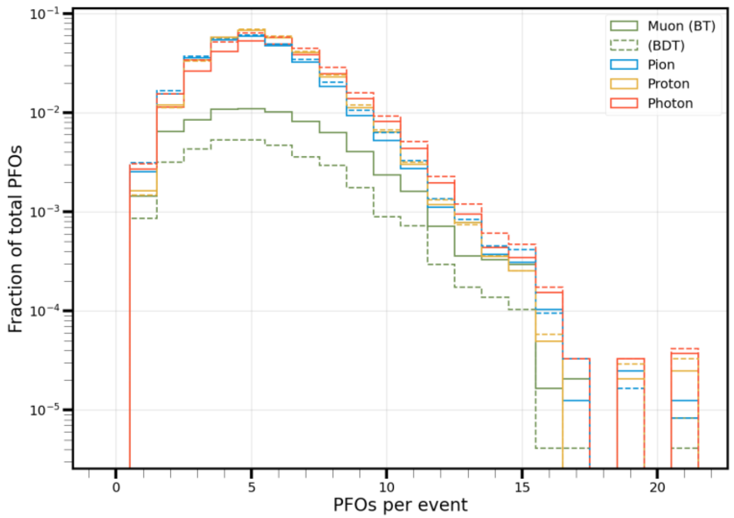
2D hists: excess in data as a function of GNN score over range between 1-12 PFOs per event (13+ PFO excluded)

Particle content

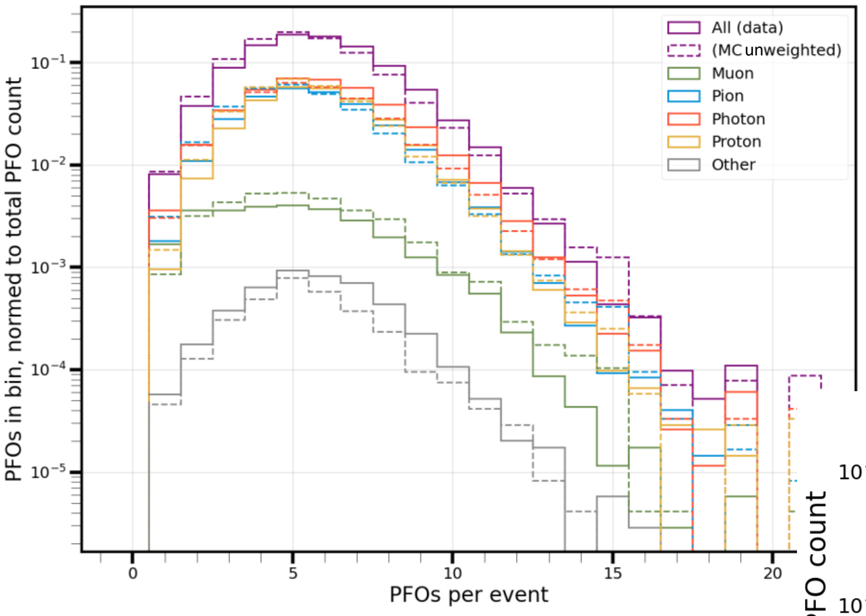
- MC vs. data discrepancy could be caused by mismodelling of the species expected from nuclear events.
- Use a simple BDT (same BDT used for PID in the full network) to estimate proportions of particles in MC vs. data.

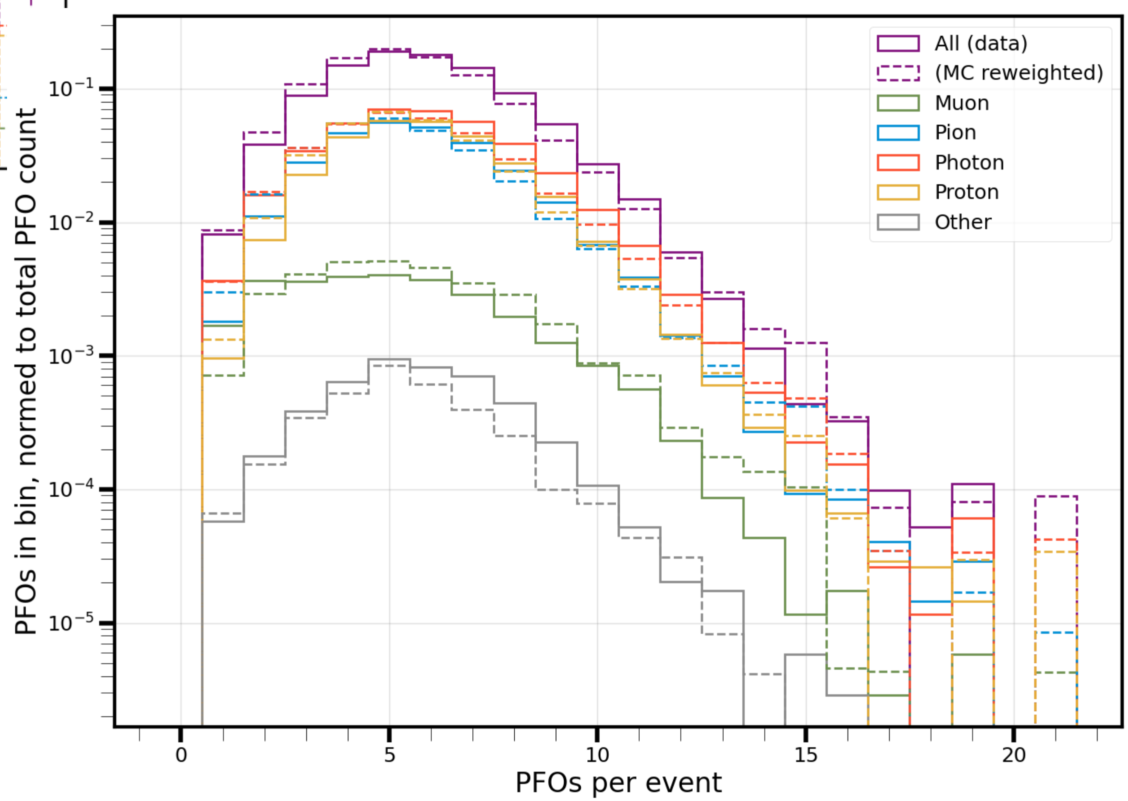


BDT classification count (solid) vs. back-tracked classification count (dashed)



Reweighting events





$$w_p = \frac{N_p^{data}}{N_{tot}^{data}} \div \frac{N_p^{MC}}{N_{tot}^{MC}}$$

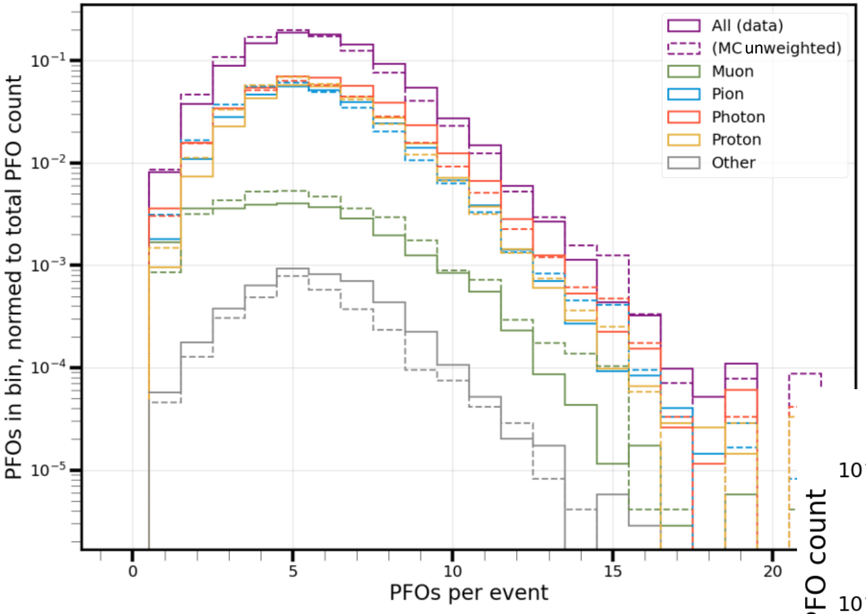
$$w_{evt} = \prod_p w_p^{\,n_p/n_{tot}}$$

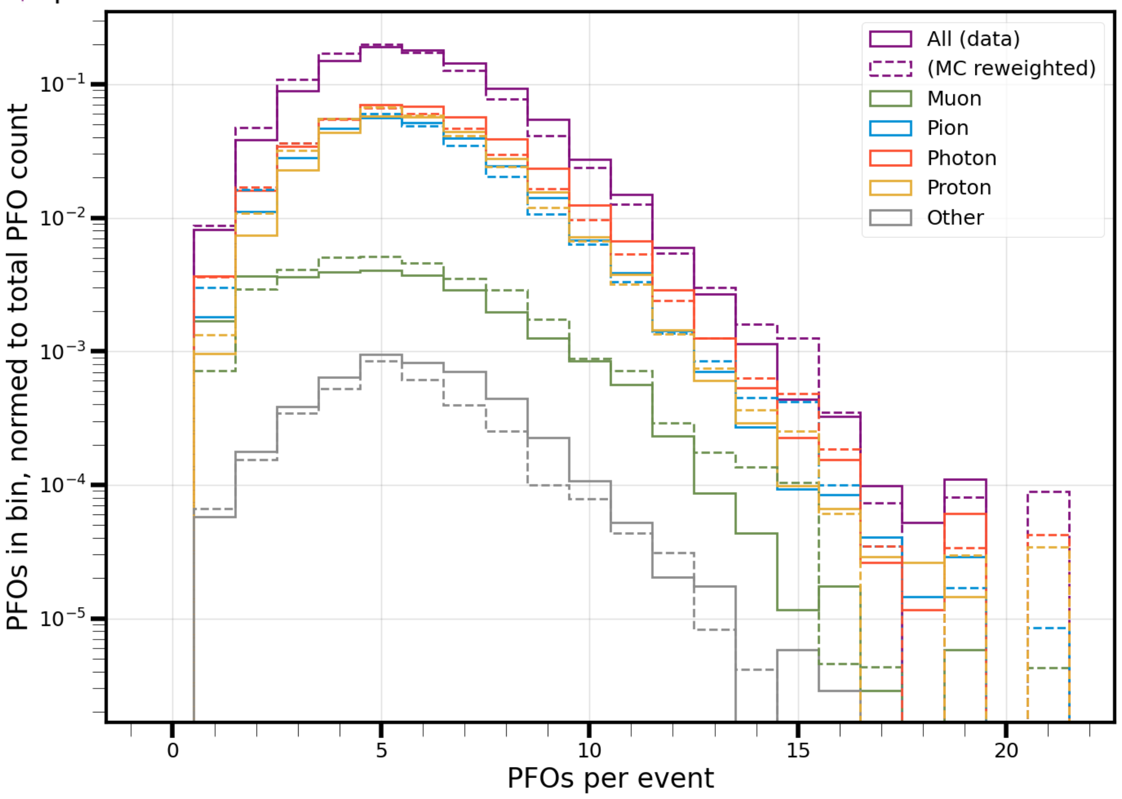
N: all events

n: particular event

p: particle species

Reweighting events





$$w_p = \frac{N_p^{MC}}{N_{tot}^{MC}} \times \frac{N_{tot}^{data}}{N_p^{data}}$$

$$w_{evt} = \sum_p n_p^{\,w_p/n_{tot}}$$

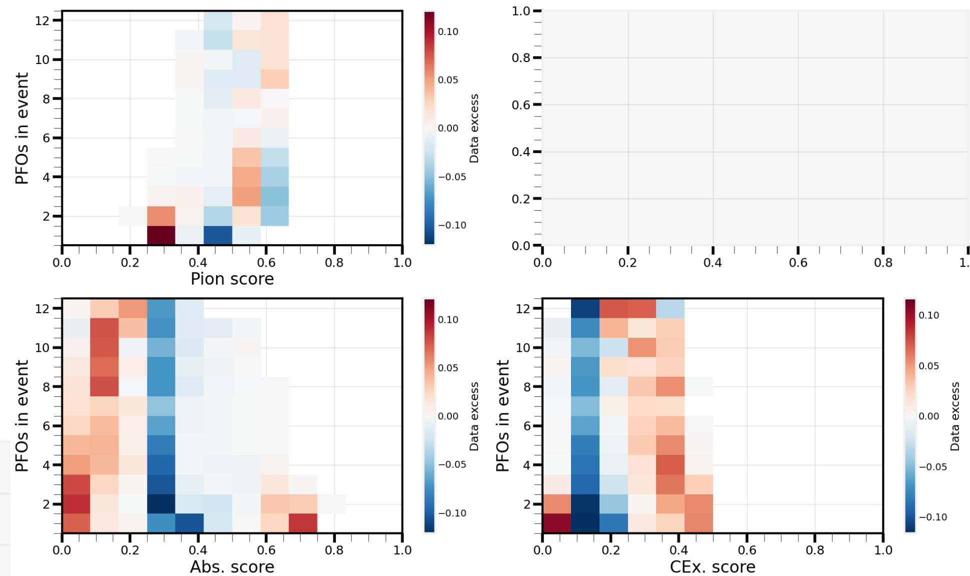
N: all events

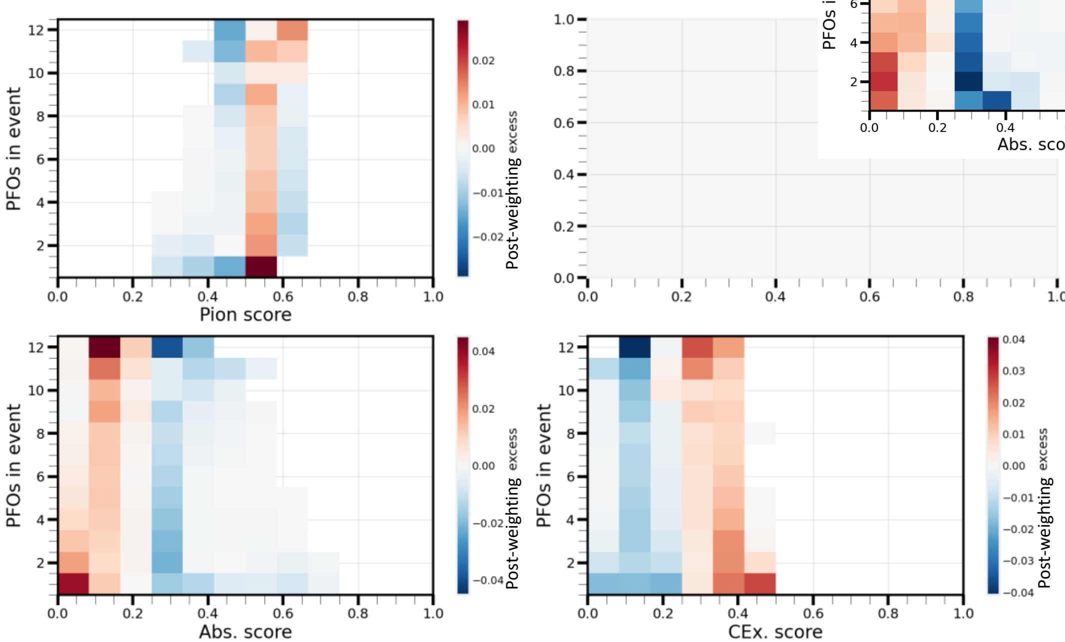
n: particular event

p: particle species

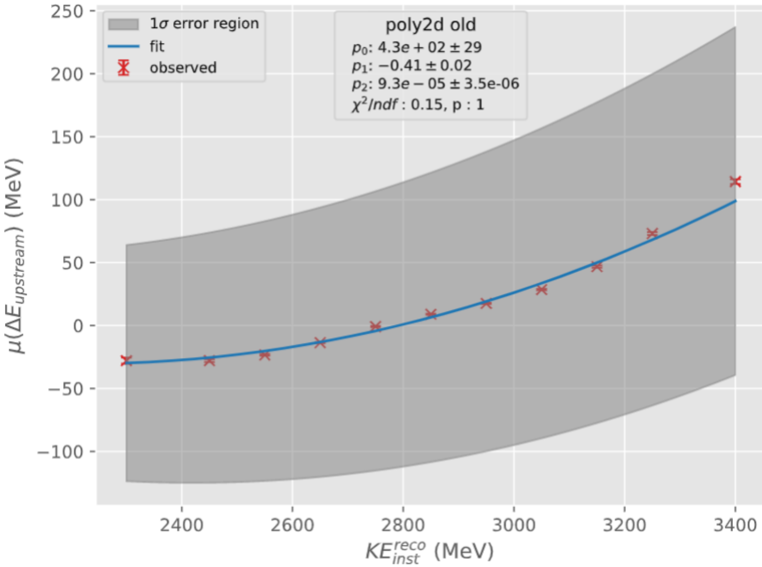
After weighting

If the re-weighting accounts to the MC/data discrepancy, the MC/reweighted difference should match the MC/data difference.





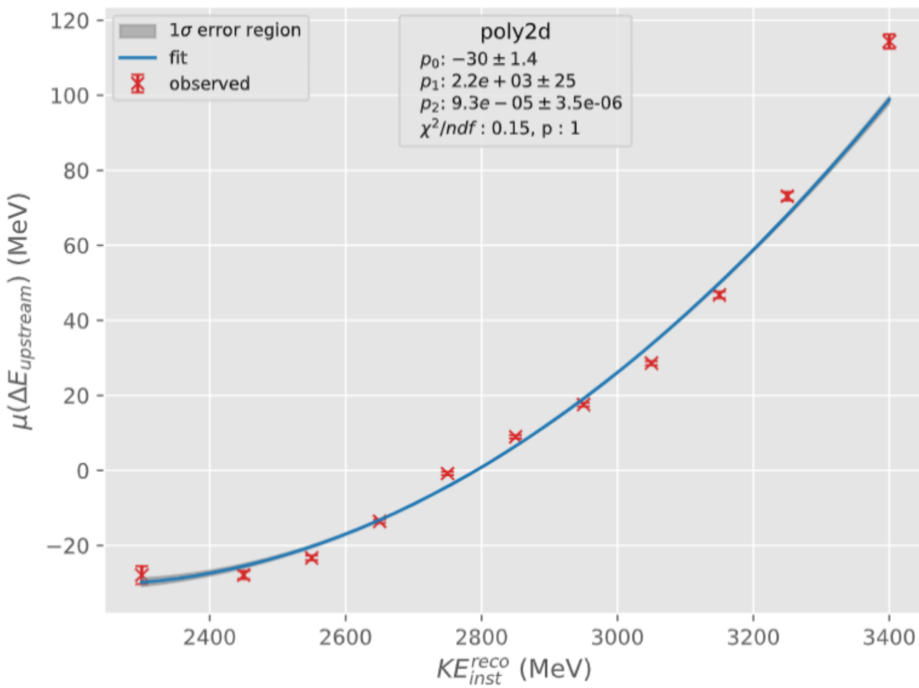
Upstream correction fit



Recall previously, fit of upstream energy correction had these excessive errors (left). Changing the equation fixes this:

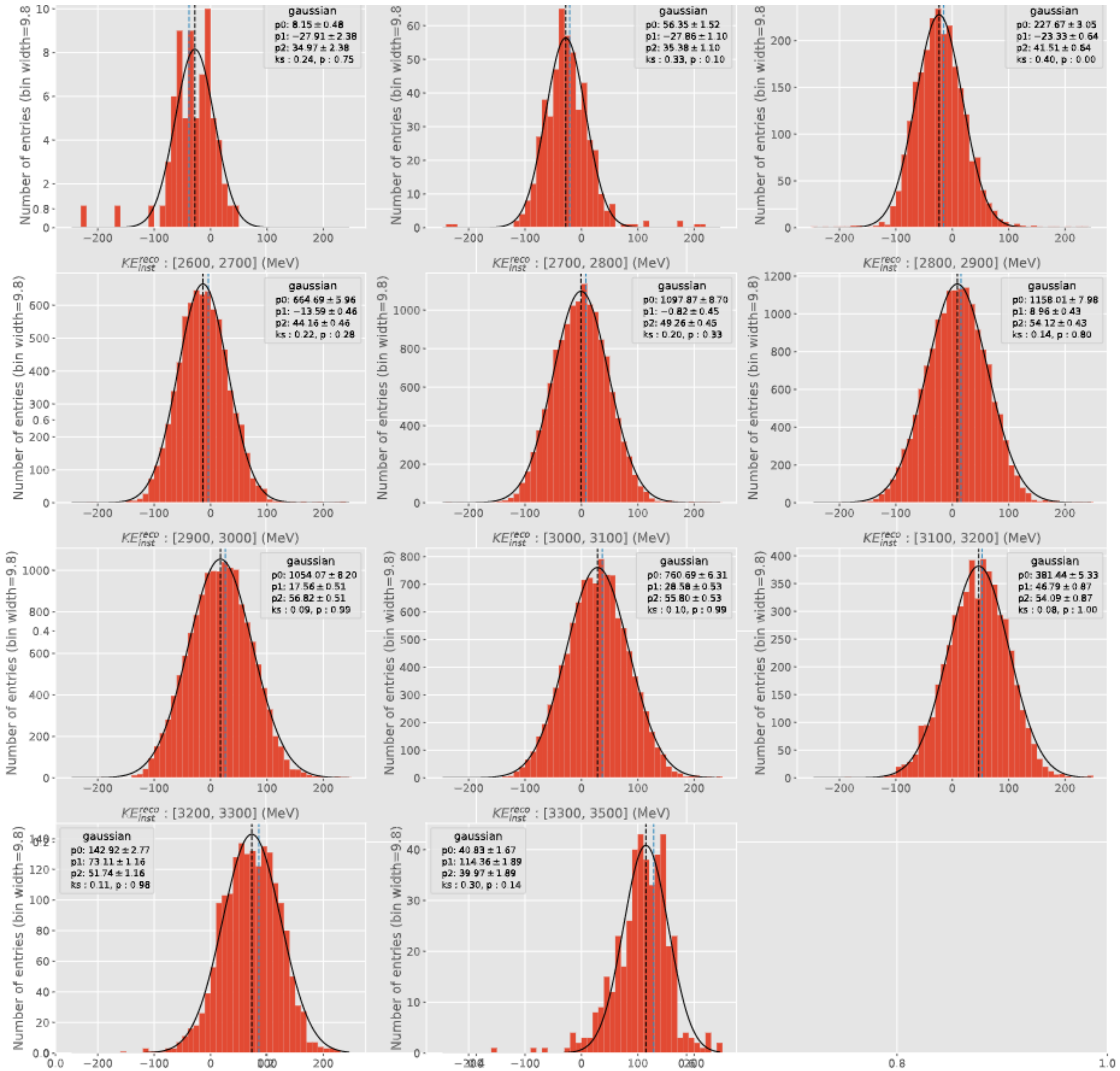
Left: $p_2 x^2 + p_1 x + p_0$

Below: $p_2(x - p_1)^2 + p_0$

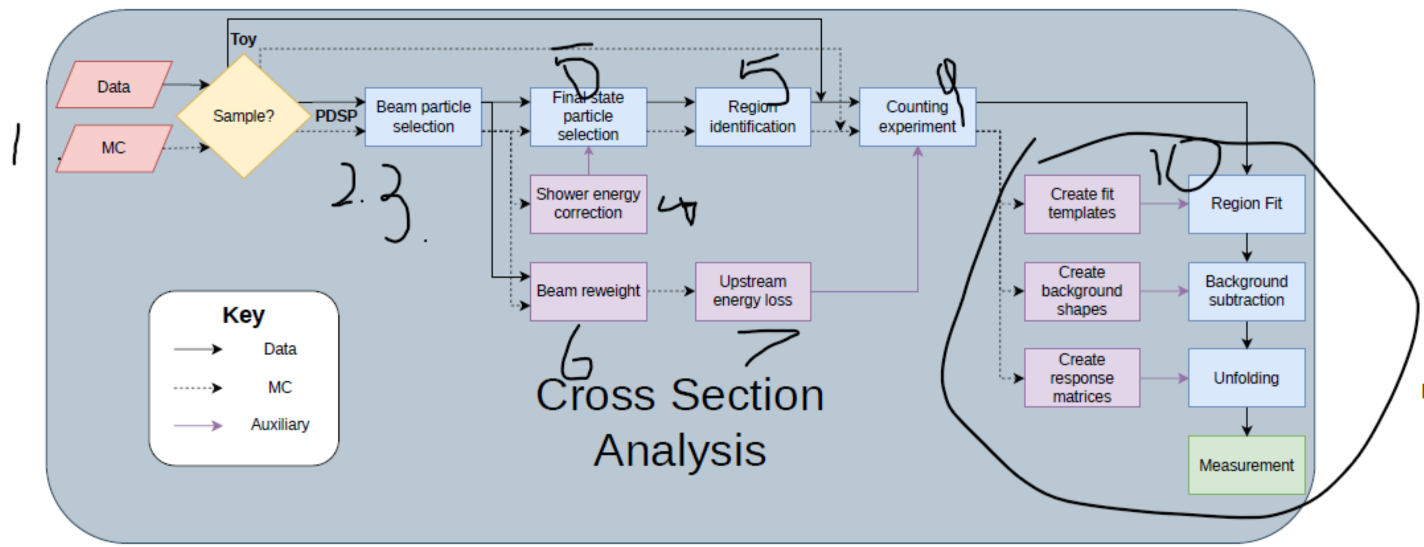


Upstream correction

- Systematic offset between Gaussian mean (black) and arithmetic mean (blue)
- Not seen in 2GeV
- Scrapers?



Code updates



App order:
1. Normalisation
2. Beam quaility
3. Beam scraper
4. Photon correction
5. Selection
6. Reweight
7. Upstream correction
8. Toy parameters
9. Analysis inputs
10. Analyse

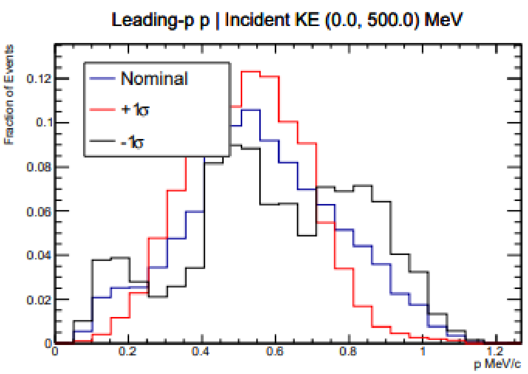
NEW order:
1. Normalisation
2. Beam quality
3. Beam scraper
4. (Per event) Photon correction
5. Event selection (maybe swap before photon correction?)
6. GNN results/PFO selection
7. Reweight
8. Upstream correction
9. Toy parameters
10. Analysis inputs
11. Analyse

Weighting schemes

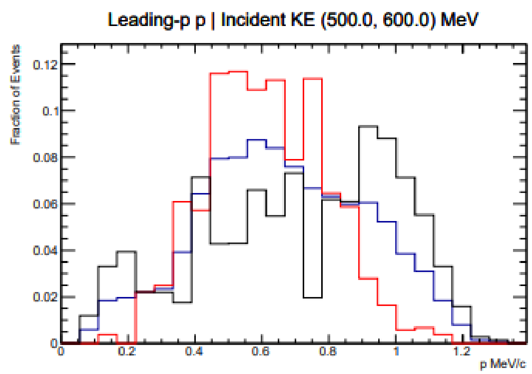
- Chatted with Jake
- Recommended start with simple event by event weights, not Geat4Reweight
- 6.6-6.8 in his technote show a series of ideas
- Find some distribution about i.e. the leading energy proton
- Create weights from these histograms

Weighting schemes

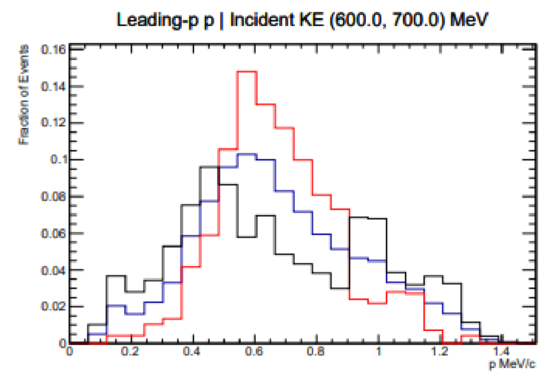
Figure 36: Efficiencies of other events (to be selected as other) as functions of leading-momentum π^+ momentum in various incident π^+ kinetic energy regions.



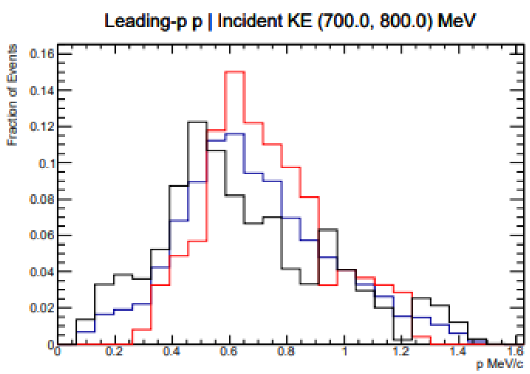
(a)



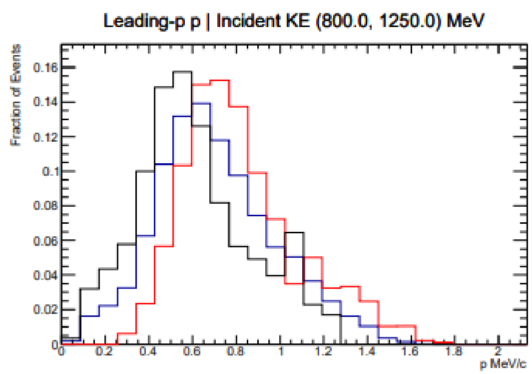
(b)



(c)



(d)



(e)

Analysis fitting!

MC-MC fits (Asimov)

- In principle, an Asimov fit fits the data with itself.
- There are still some effects which change the values used:
  - Energy binning by reconstructed vs. true interaction energies
  - Beam reweighting (this shouldn’t happen in a true Asimov fit, but we can test a small effect from difference between beam with and without the MC training sample)

Reconstructed slicing, no weights

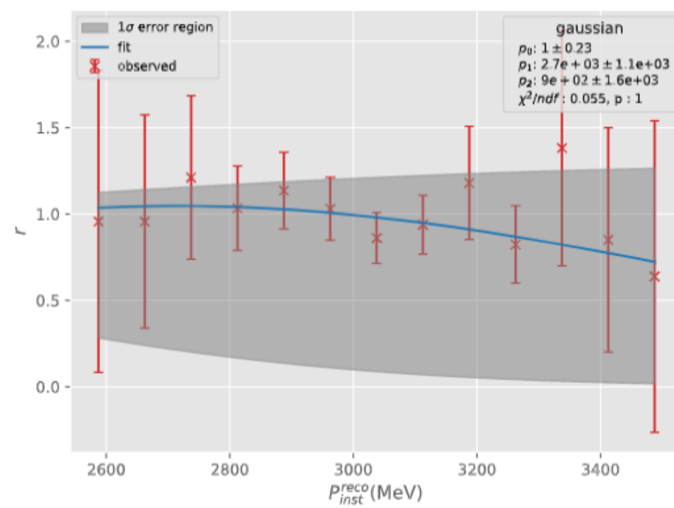
	3.1-2.825 GeV			2.825-2.55 GeV			2.55-2.275 GeV			2.275-2.0 GeV		
	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion
Init. pred	52	0	1804	525	0	16132	646	0	13304	155	0	1964
True yield	97	83	1676	1001	763	14893	1203	686	12061	263	107	1749
Fit yield	97.0	82.8	1676.5	1001.2	765.2	14892.7	1203.0	686.2	12060.5	263.0	107.0	1749.1
Fit unc.	21.8	66.2	89.4	72.9	176.3	250.9	72.8	160.6	229.2	29.8	55.7	82.8

- When using reconstructed slicing with no weighting, the templates and data exactly match
- Excellent agreement expected

Reconstructed slicing, weighted

	3.1-2.825 GeV			2.825-2.55 GeV			2.55-2.275 GeV			2.275-2.0 GeV		
	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion
Init. pred	52	0	1804	525	0	16132	646	0	13304	155	0	1964
True yield	97	83	1676	1001	763	14893	1203	686	12061	263	107	1749
Fit yield	155.1	55.7	1642.1	2437.0	797.8	13456.6	2698.9	773.1	10522.6	425.6	108.8	1588.2
Fit unc.	47.7	65.4	94.0	240.0	175.7	315.5	240.0	161.1	296.7	75.0	53.0	90.1
Pull	1.22	-0.42	-0.36	5.98	0.20	-4.55	6.23	0.54	-5.19	2.17	0.03	-1.78

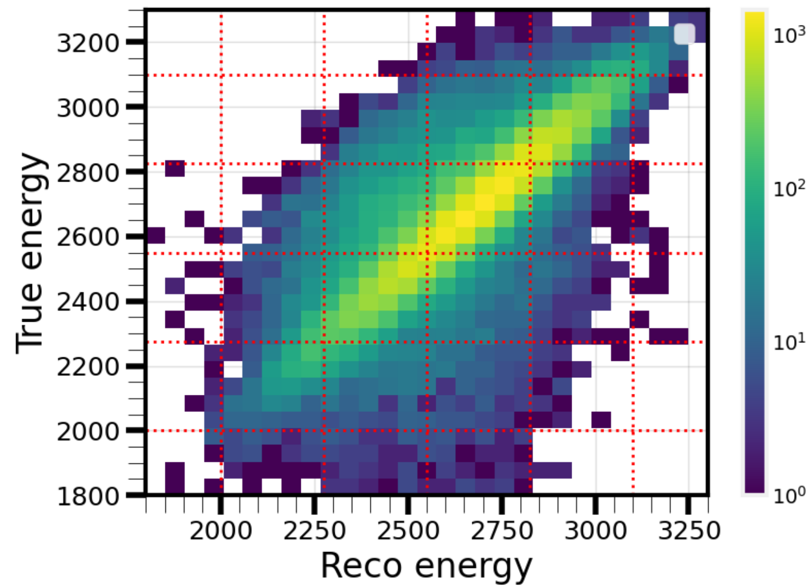
- Same samples for each, but slight differences due to beam weights applied to the templates.



True slicing, no weights

	3.1-2.825 GeV			2.825-2.55 GeV			2.55-2.275 GeV			2.275-2.0 GeV		
	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion
Init. pred	52	0	1804	525	0	16132	646	0	13304	155	0	1964
True yield	97	83	1676	1001	763	14893	1203	686	12061	263	107	1749
Fit yield	220.2	147.1	1491.7	2797.1	457.5	13535.1	3337.7	1166.3	9639.4	678.8	289.5	1222.3
Fit unc.	63.9	65.1	98.1	292.1	193.2	346	320.5	178	346.4	142.7	72	116.9
Pull	1.93	0.98	-1.88	6.15	-1.58	-3.92	6.66	2.70	-6.99	2.91	2.53	-4.51

- Now templates are constructed from true energies, data from reco.
- No reweighting



True slicing, no weights

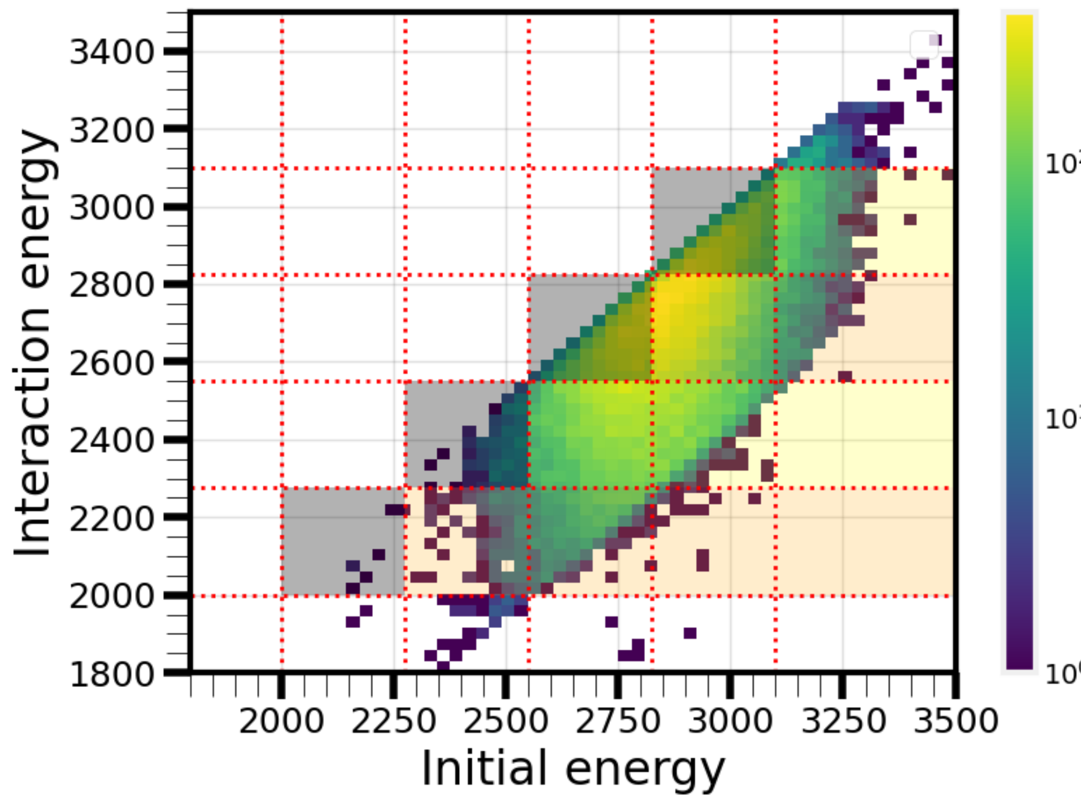
	3.1-2.825 GeV			2.825-2.55 GeV			2.55-2.275 GeV			2.275-2.0 GeV		
	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion	Abs.	CEx.	Pion
Init. pred	52	0	1804	525	0	16132	646	0	13304	155	0	1964
True yield	97	83	1676	1001	763	14893	1203	686	12061	263	107	1749
Fit yield	164.2	134.4	1559.3	2852.1	453.6	13517	3485	1179.8	9522.7	614.6	303.6	1274.1
Fit unc.	51.1	58.7	88.3	308.2	188.6	350.7	341.6	177.5	357.9	135.4	67.7	113
Pull	1.32	0.88	-1.32	6.01	-1.64	-3.92	6.68	2.78	-7.09	2.60	2.90	-4.20

- Templates binned in true energy
- Templates reweighted

Energy binning

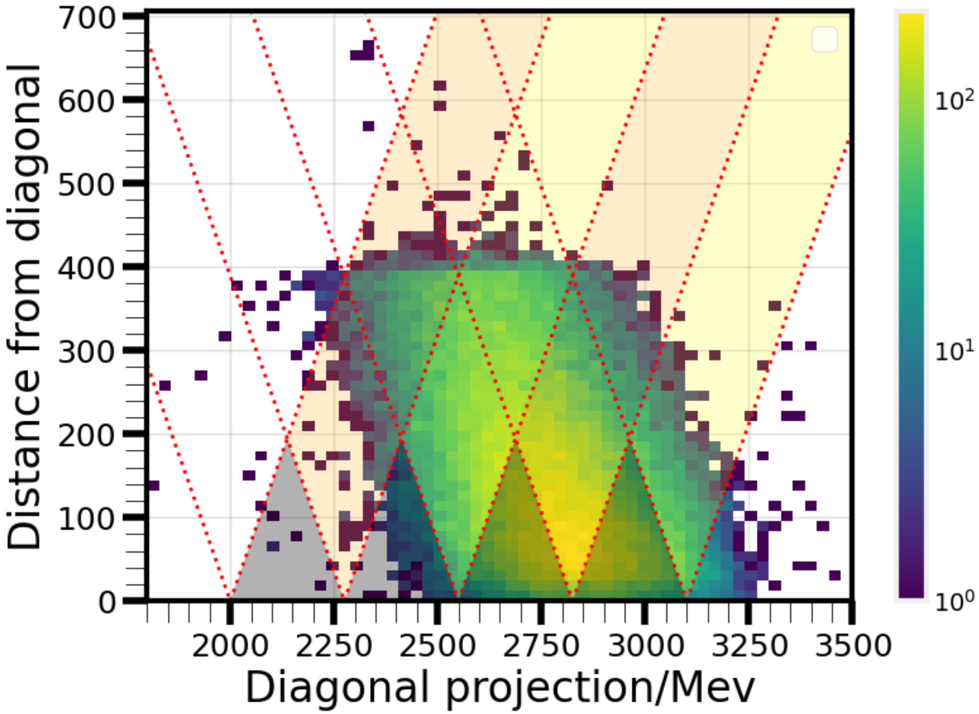
Choose set of fixed bins

- Plan to construct a set of bins before running more robustness type tests
  - Fix the bins now, for consistency
- Update required for variable widths
- We want at least *N* events per bin, and minimise the number of events this makes invalid



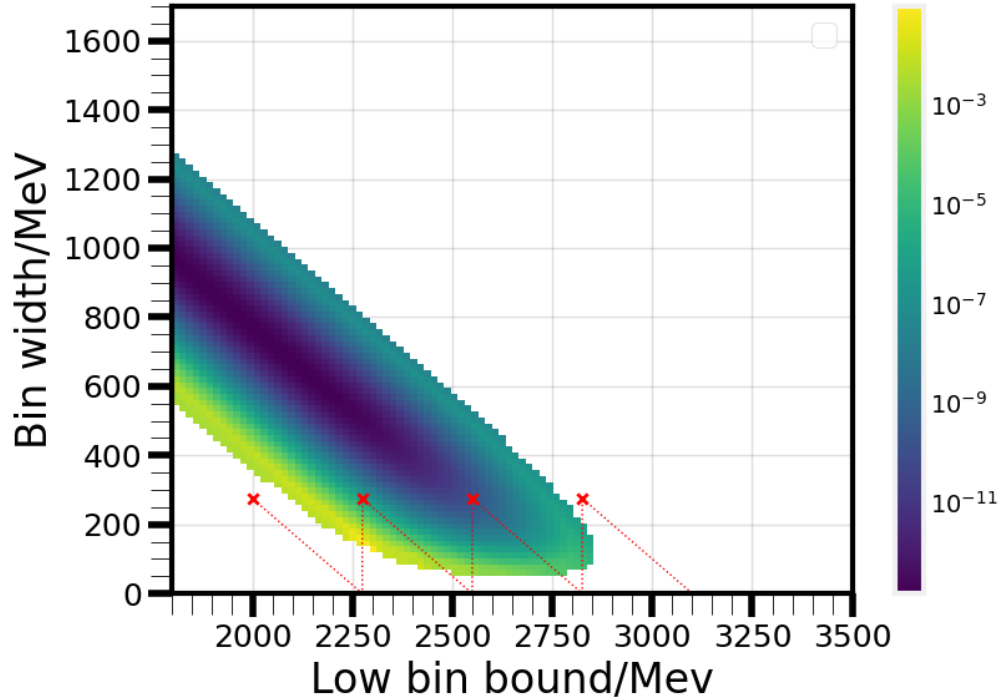
Energy width optimisation

- I spent way too much effort on this…
  - But it was too fun a challenge to ignore!
- Consider projecting along the $E_{init} = E_{int}$ line
- The perpendicular distance is proportional to the bin width
- Let's calculate the number of events in/excluded by some bin edges…



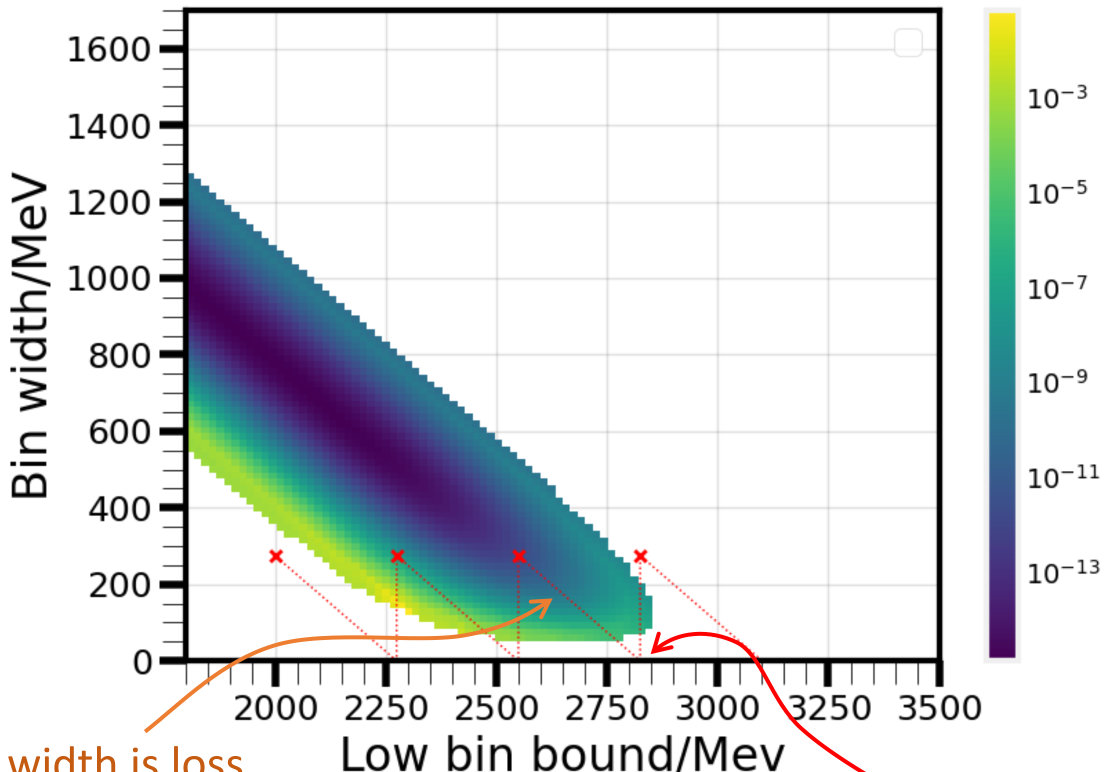
Optimisation strategy

- We can calculate a loss:
- $$L = \begin{cases} m^{1.5} \times e^{\frac{5*(i-6000)}{6000}} & , \ i \geq 4000 \\ \infty & , \text{otherwise} \end{cases}$$
    - i=number of valid events in bin
    - m=number of events excluded from E_{init}, E_{int} in same bin
- View current bins against this loss



Optimisation strategy

- Given some upper bin edge
- Optimum lower bin edge is minimum along line of const. upper bin ($y = -x$)

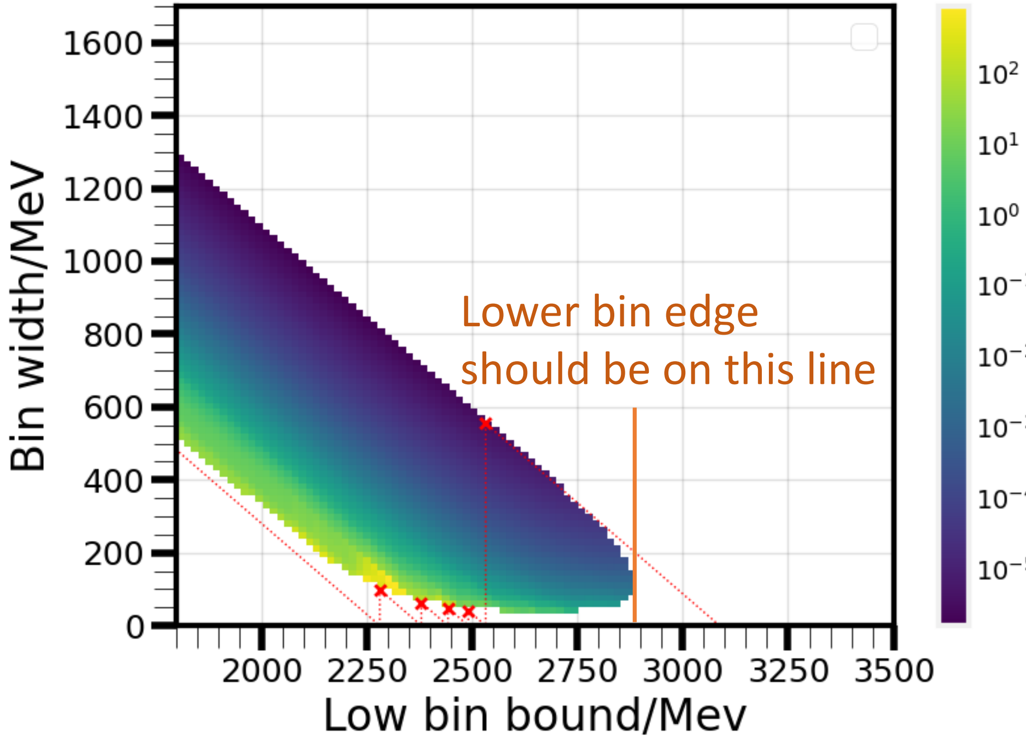


Optimum width is loss minimum along this line

Upper edge fixed

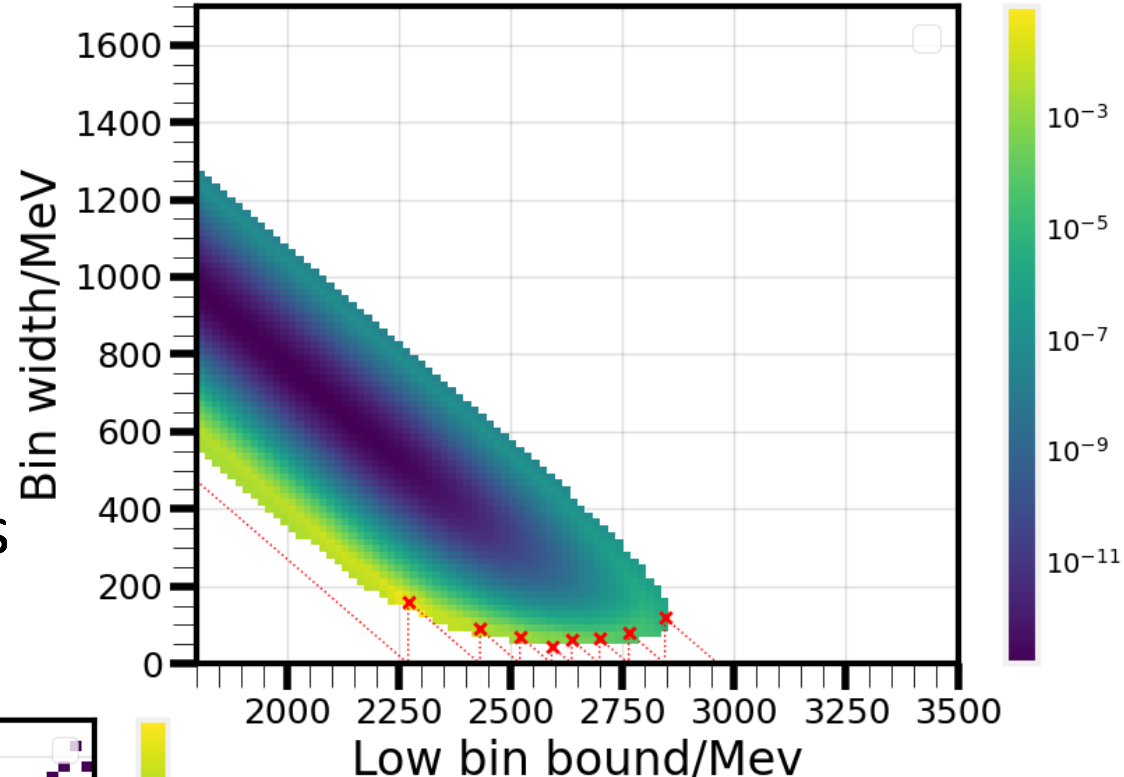
First bin optimisation

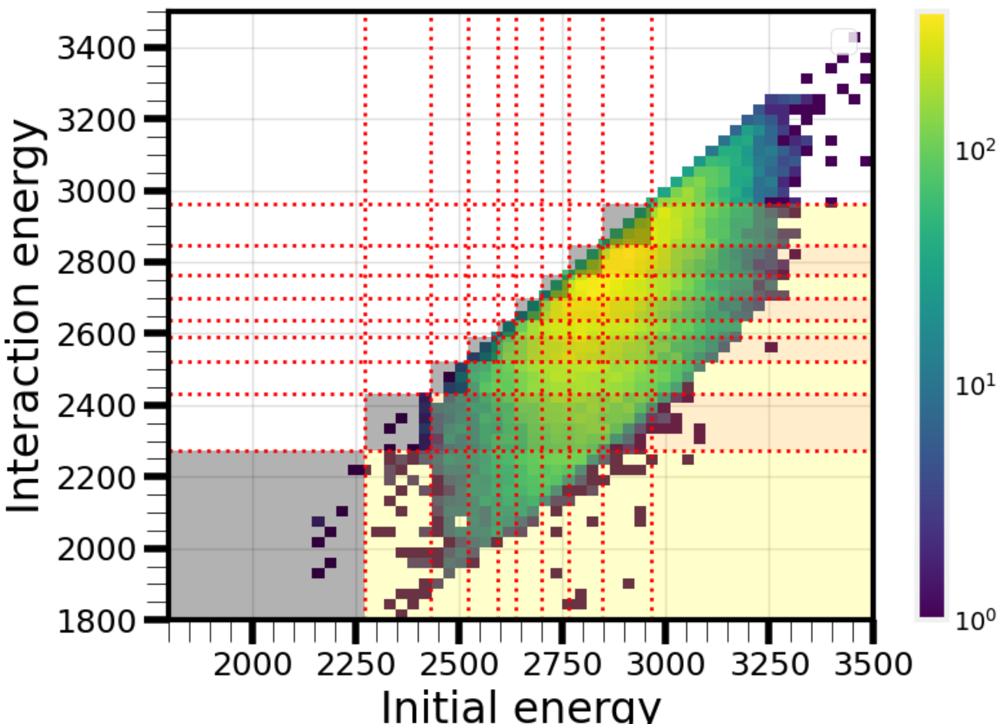
- Naïvely, I applied the same rules to the first bin.
- Picks out the tangent of -1 gradient
- Actually want tangent of infinite gradient.
  - I was aiming for this from the start, but I didn't account for silly coding!



Results

- 9 bins
- 5981 missed events (11.49%)





- Occupancies:
  - [4001, 5875, 5935, 5993, 4228, 5998, 5919, 6000, 2118]

Results

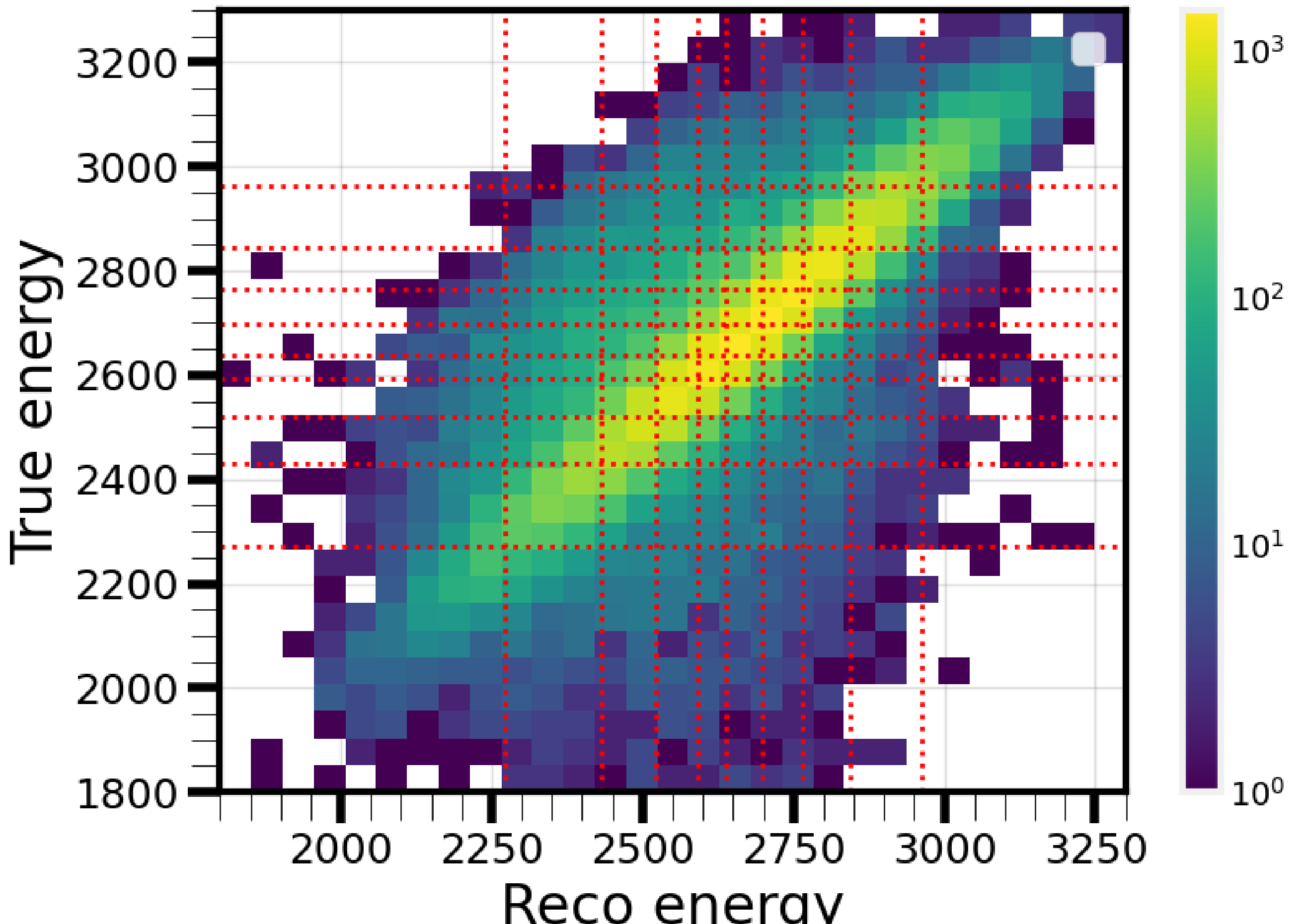
- Can tune the parameters
- Target number of bins via the target count per bin.
- First bin occupancy
- Importance of missing events vs. having many
- Probably should relax the target count addition to make the minimum less deep (do some calculus!)

Other considerations

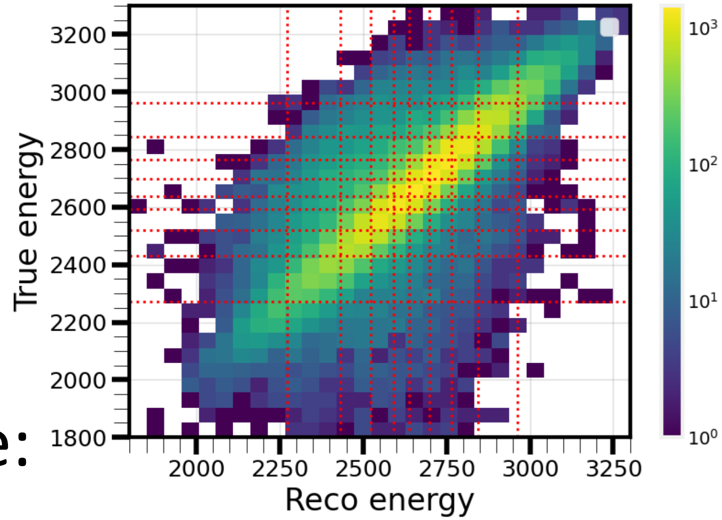
- Missing events bias against high cross-section interactions in energy slice version
  - $P(\text{selected event})\ \alpha \int_{\Delta E} P(\Delta E) P(\text{interact in } \Delta E)\ \alpha\ 1/\sigma$
  - $\Delta E(E_{init}; \text{bins})$ runs over possible energy range before first bin boundary
- Likely less important for thin slice version of the analysis (particles probably start interacting instantly, can add an arbitrarily small initial bin)
- Thin slice method should be reassessed (but perhaps post-thesis worthy result…)

Unfolding – total or per process

- Current binning has too many bins for nice unfolding.
- Small bins to fit, large bins to unfold?

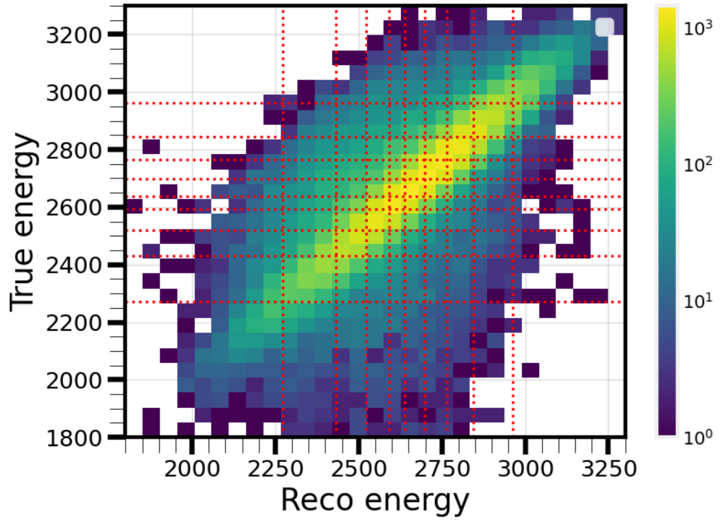


Unfolding – total or per process



- Assume detector has process invariant response.
- $P(E_{reco}|E_{true})$
- Unfolding inverts this to estimate: $P(E_{true}|E_{reco})$
- This is performed on histograms. Either:
  - Interacting yields produced *post*-fit
  - Incident yields *pre*-fit (one could try constructing this via moving fractions of histograms around)

Unfolding – total or per process



- We want to find $E_{true,i}$ which is the true energy of the process, i we want to measure
- As such we now have multiple "causes", dependent on the $\{\sigma_i\}$
- $P(E_{true,i}|E_{reco,j})$ to unfold *post*-classification
- Assume detector-independent: $P(E_{true}|E_{reco})$
  $P\left(E_{true,i}\middle|E_{reco,j}\right)$

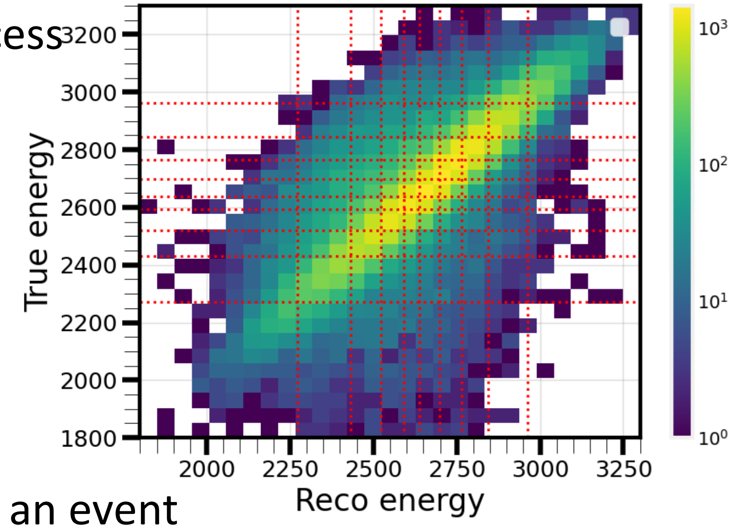
$$= \int_{E_{true}} \frac{P(E_{true}|E_{reco}) \times}{P(E_{true}\,|\,i)\;\; P(E_{reco,j} \in i)}$$

Unfolding – total or per process

Unfolded with no knowledge of process

$$\int_{E_{true}} P(E_{true}|E_{reco}) \times P(E_{true}|i) P(E_{reco,j} \in i)$$

Ignore this term!



- $P(E_{true}|i)$ depends on σ_i.
- This cannot be determined from MC alone, since it is based on the actual cross-section.
- Factorising out means this could be iteratively improved, in principle.

- Probability that an event classified as j is actually process i
- For unfolding *pre*-fitting, this is summed over, so can be ignored:

$$P(E_{true,i}|E_{reco}) = \sum_j P(E_{true,i}|E_{reco,j})$$

- For unfolding *post*-fitting, we assume $P(E_{reco,j} \in i) = \delta_{i,j}$