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

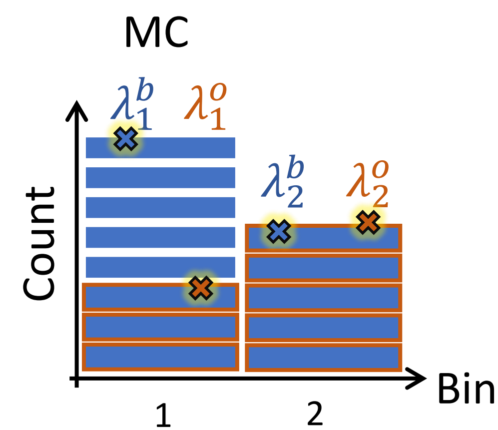
- The fit uses (python) Minuit's template fit, using Dembinski and Abdelmotteleb method.
- D. and A.'s method approximates the Beeston-Barlow method.
  - Henceforth, will discuss pure Beeston-Barlow, trusting the D. and A. method is sensible
- Methods can also deal with weighting the MC templates (no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

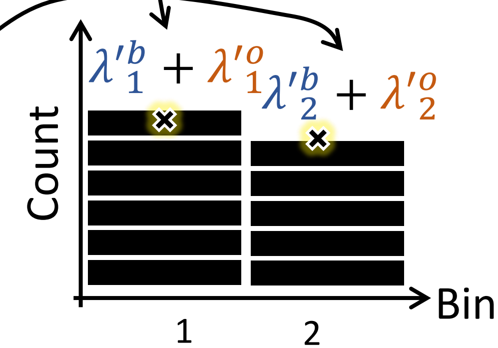
Fitting method

- 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

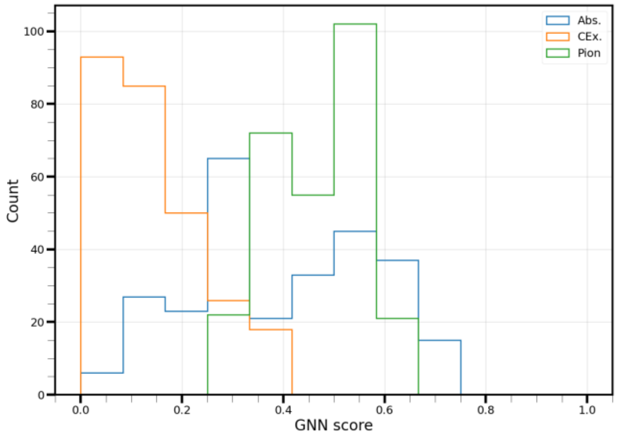
Labels for ID,
$N^{\text{labels}} = N^{\text{temps}}$

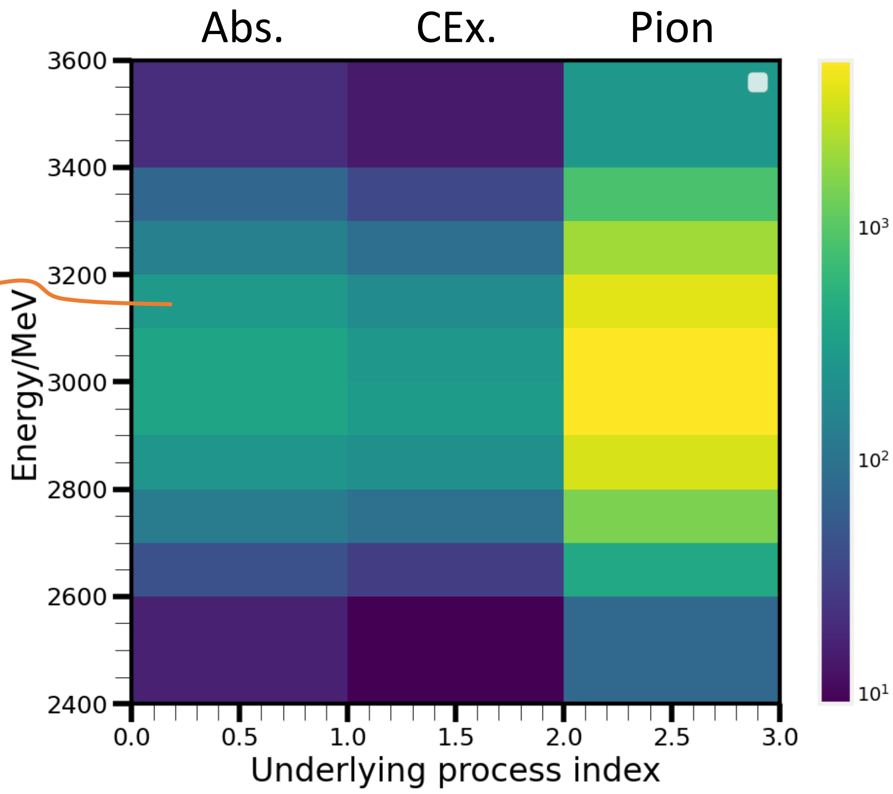
List of $\boldsymbol{N^{\text{temps}}}$ **histograms as templates**. There will be N^{temps} yields given by the fit, one for each templates
Shape: $[(N_e, N_b, N_b, N_b)] * N^{\text{temps}}$
(Each template has the same shape as the data histogram, but there are N^{temps} in the list)

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

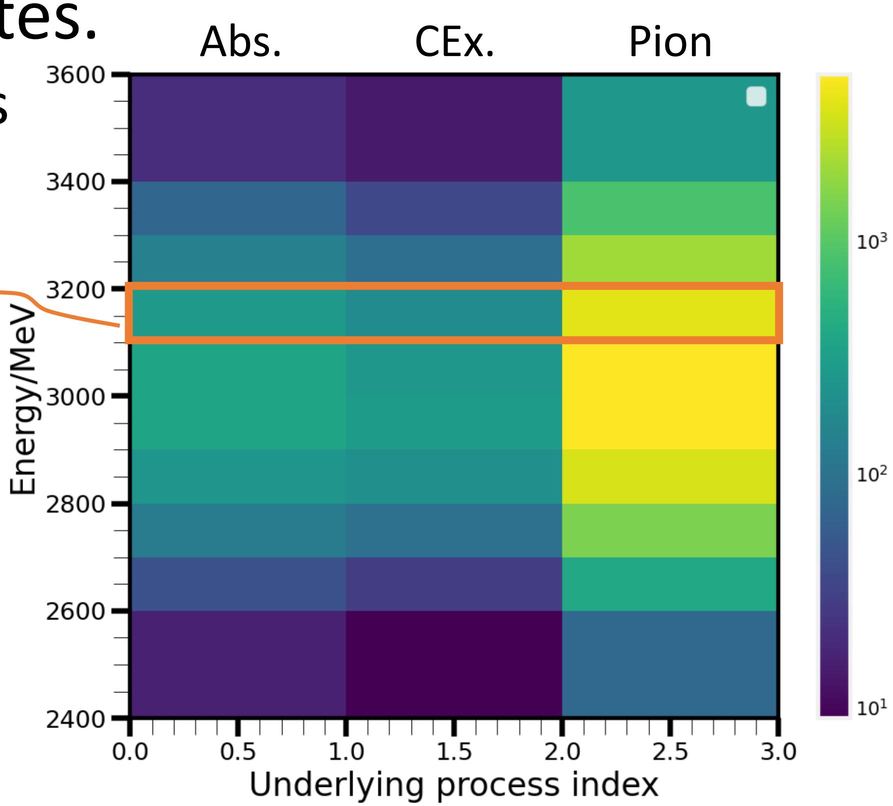




Fitting options – current

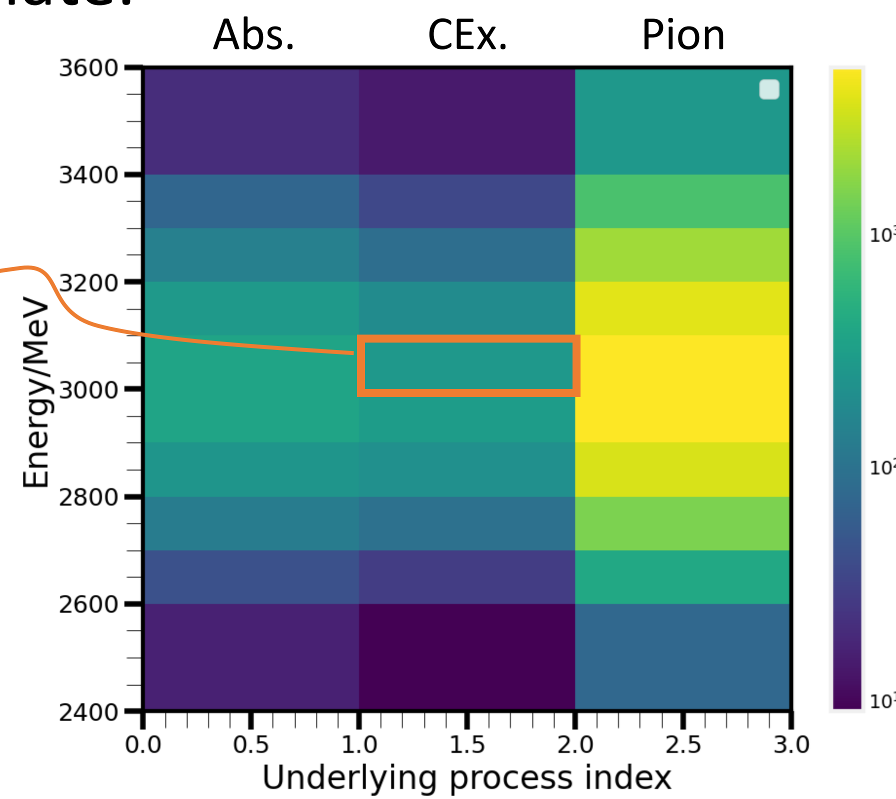
- Current idea, do N_e separate fits, each to one data histogram, shape (N_b, N_b, N_b).
- For each bin, get 3 templates.
    - For each bin, the templates are the three on the corresponding row of this histogram.

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

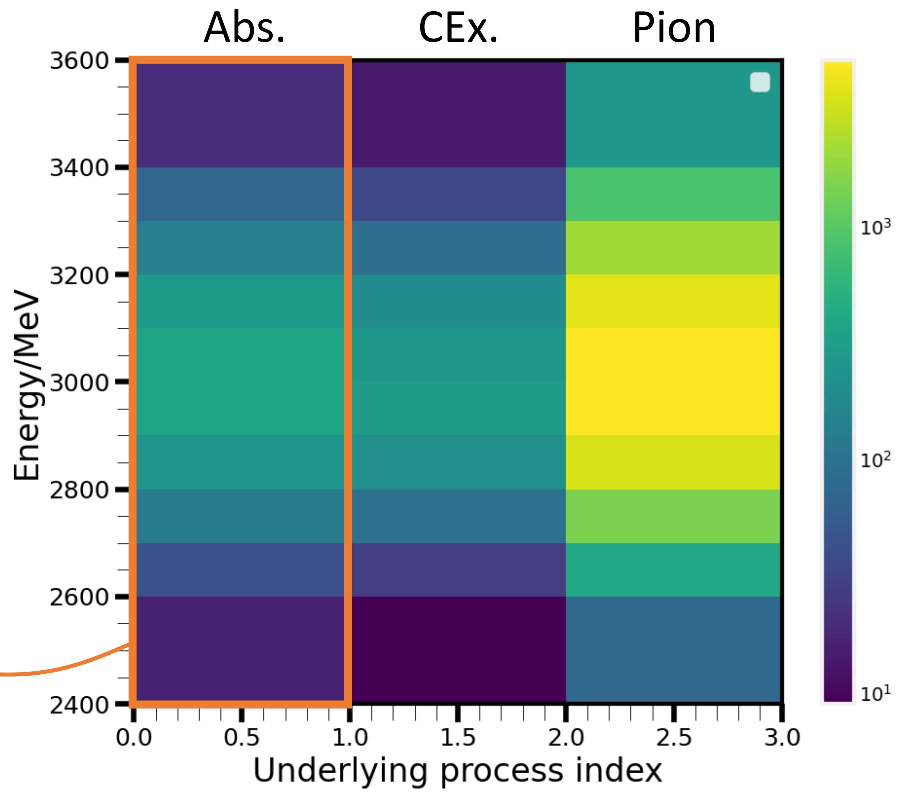
- A valid (but poor) fitting option would be to do one fit to all data (N_b, N_b, N_b), where each energy and process gets its own template.
- $3 \times N_e$ templates total, each (N_b, N_b, N_b)



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

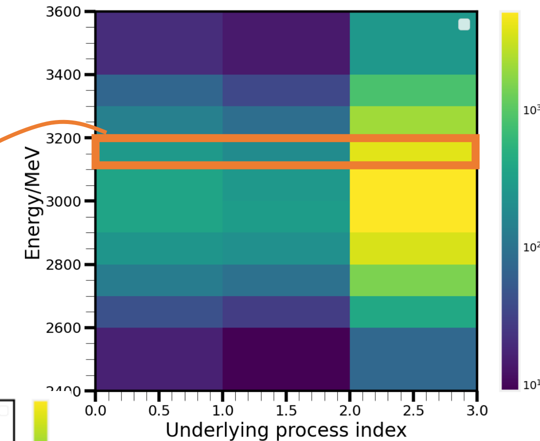


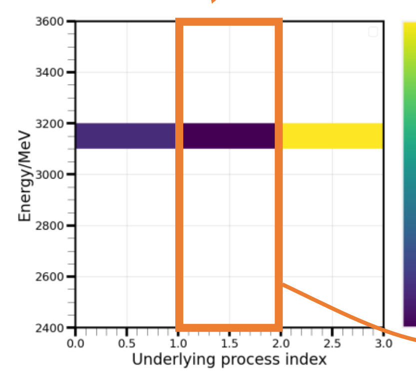
Relative fractions of events fixed in by templates – bad!

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.



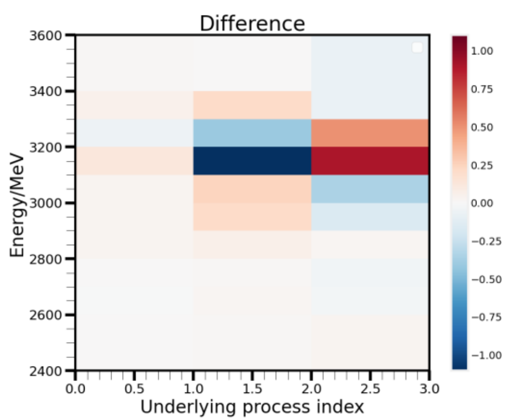


Each process is then given as its own template, for $3 \times N_e$ total.

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

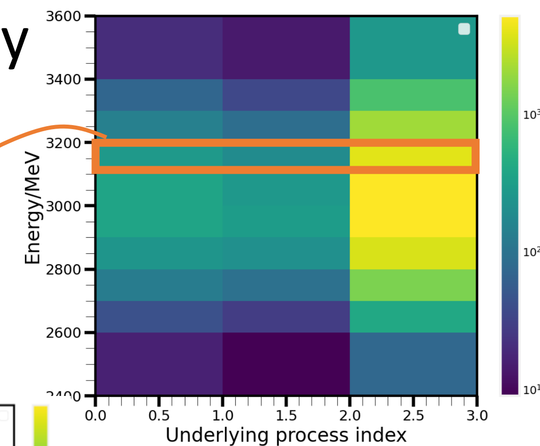


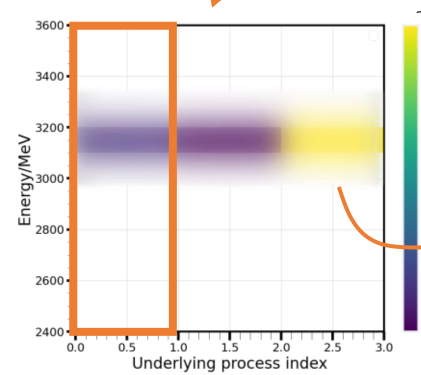
Current / E. binned - 1

```
[[ 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

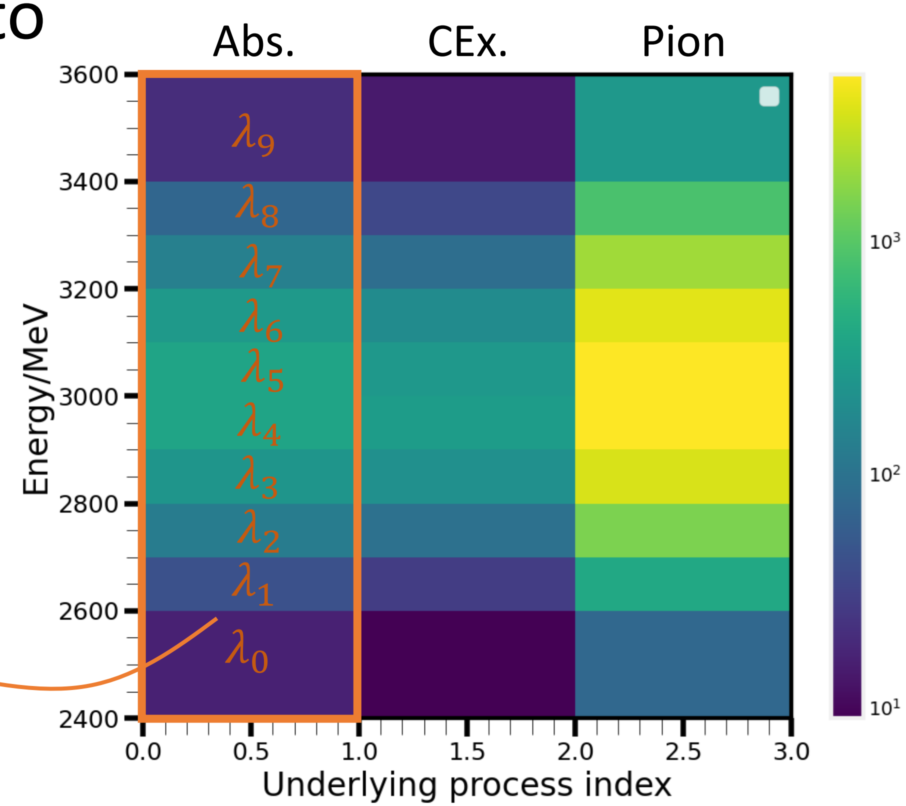




“Artistic representation” of binning in reco. energy, but selecting in MC true energy

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

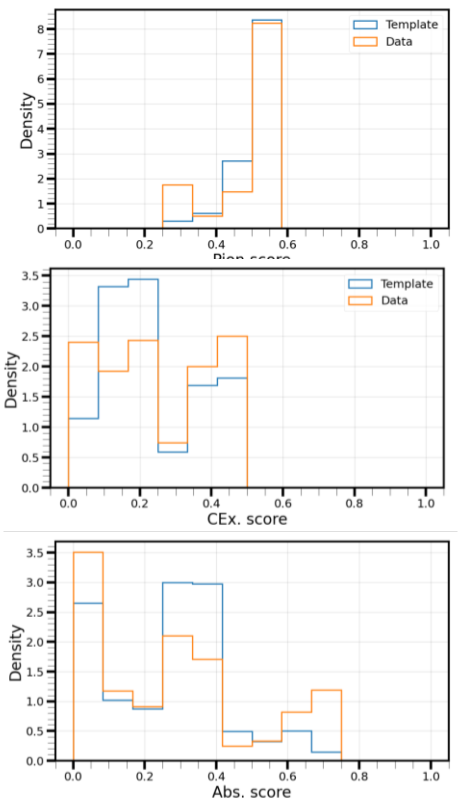


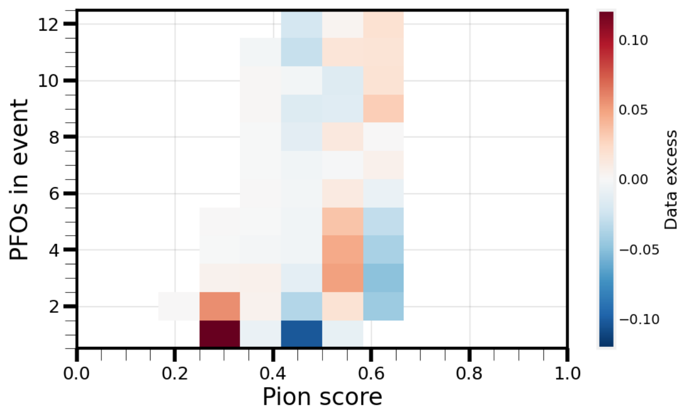
Extract nuisances to reconstruct shape. Probably possible, but definitely complicated… (e.g. correlations between overall norm and the nuisances)

PFO count variation - comparison

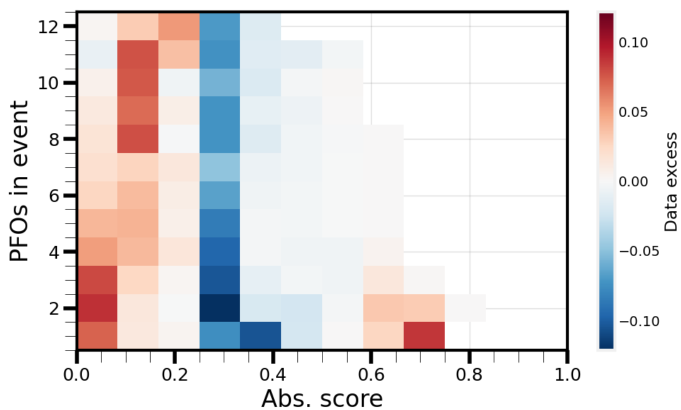
- Plots compare all MC events (not split by true process) vs. data events.

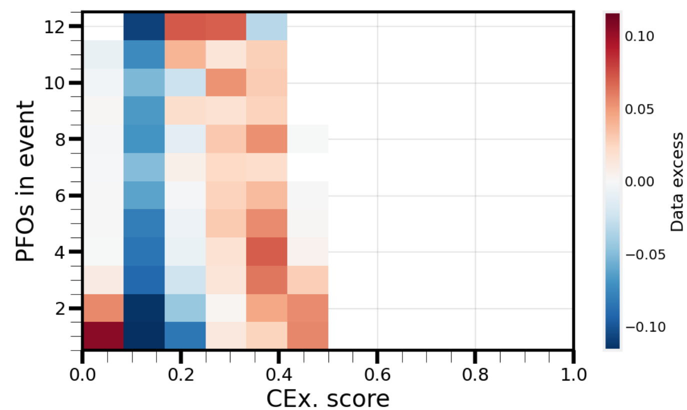
1 PFO in event scores





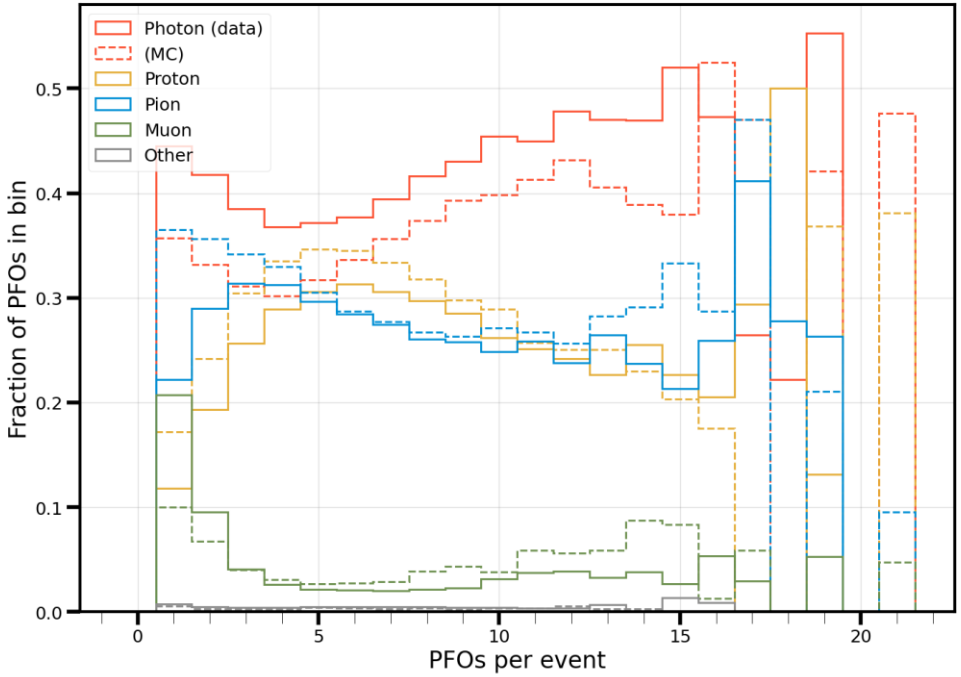
2D hists: excess in data as a function of GNN score over range between 1-12 PFOs per event (13+ PFO excluded)



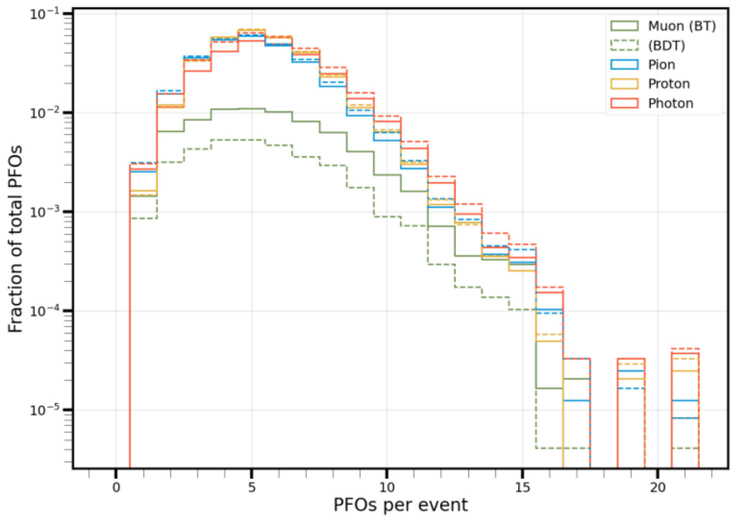


Particle content

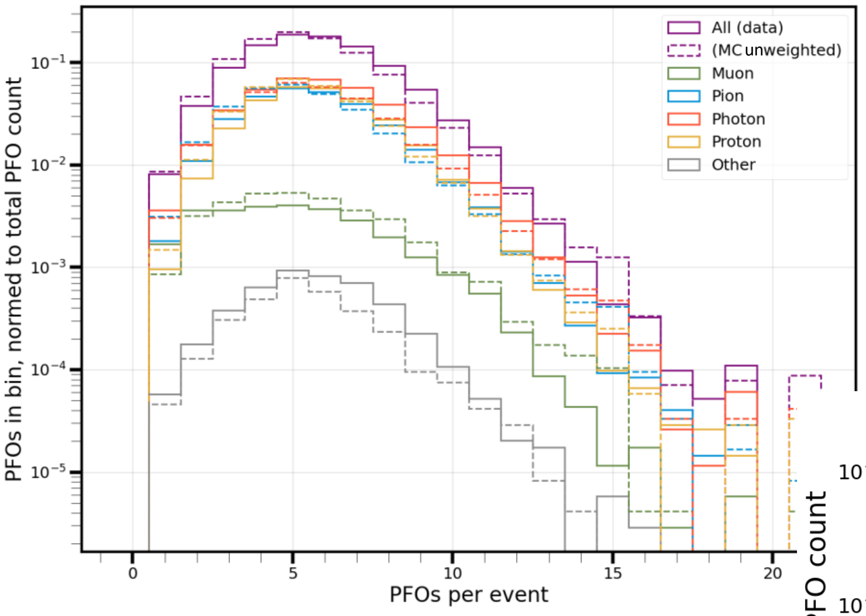
- MC vs. data discrepancy could be caused by mismodelling of the species expected from nuclear events.
- Use a simple BDT (same BDT used for PID in the full network) to estimate proportions of particles in MC vs. data.



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



Reweighting events



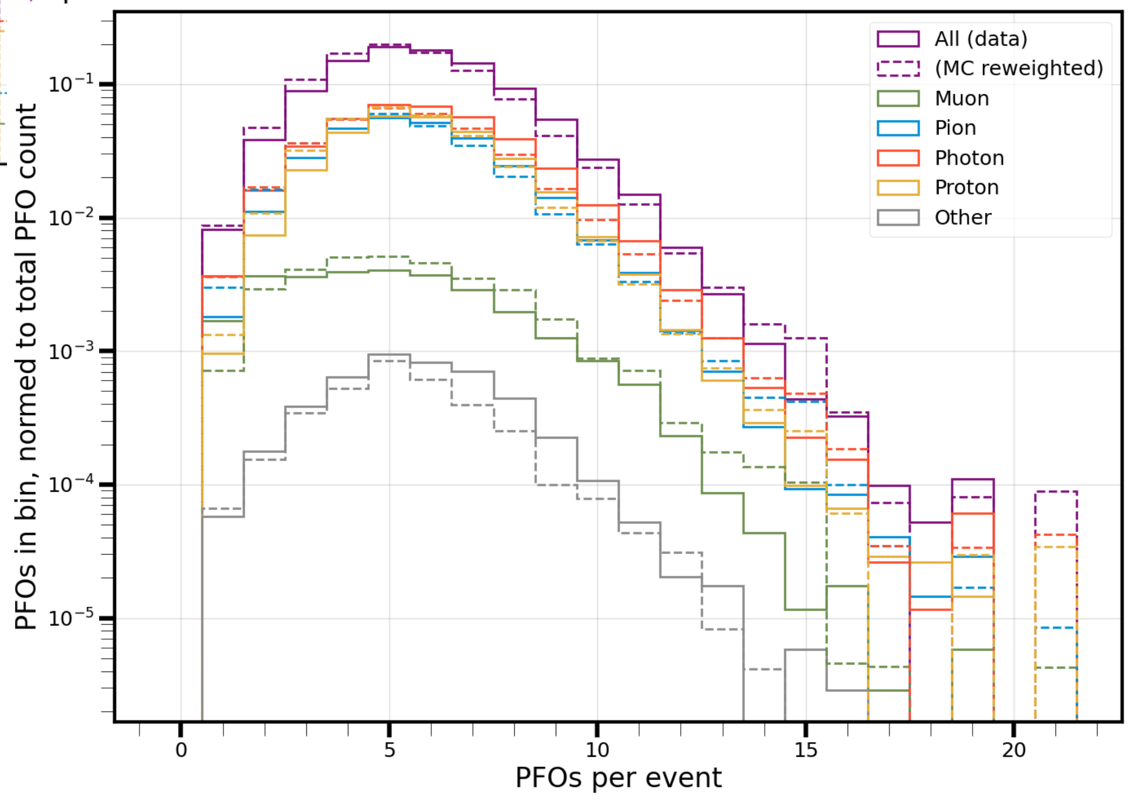
$$w_p = \frac{N_p^{data}}{N_{tot}^{data}} \div \frac{N_p^{MC}}{N_{tot}^{MC}}$$

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

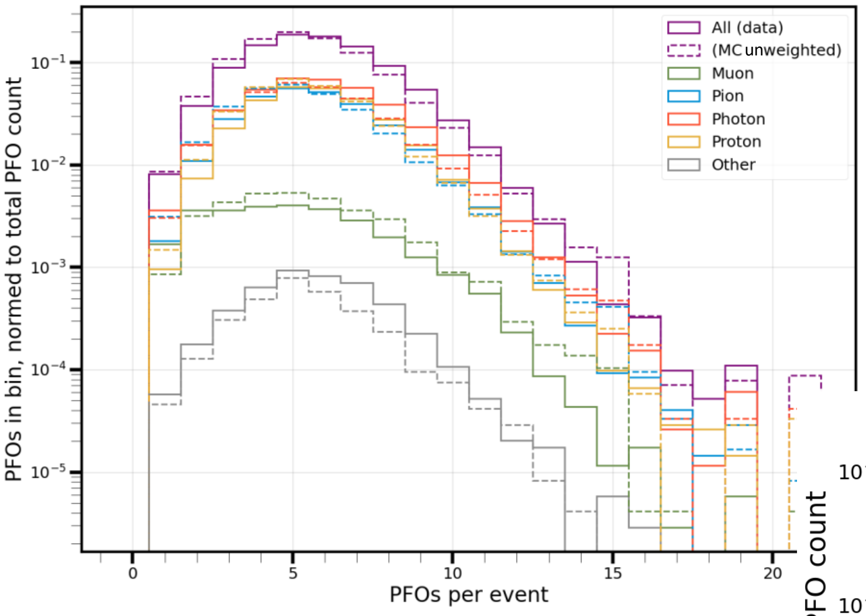
N: all events

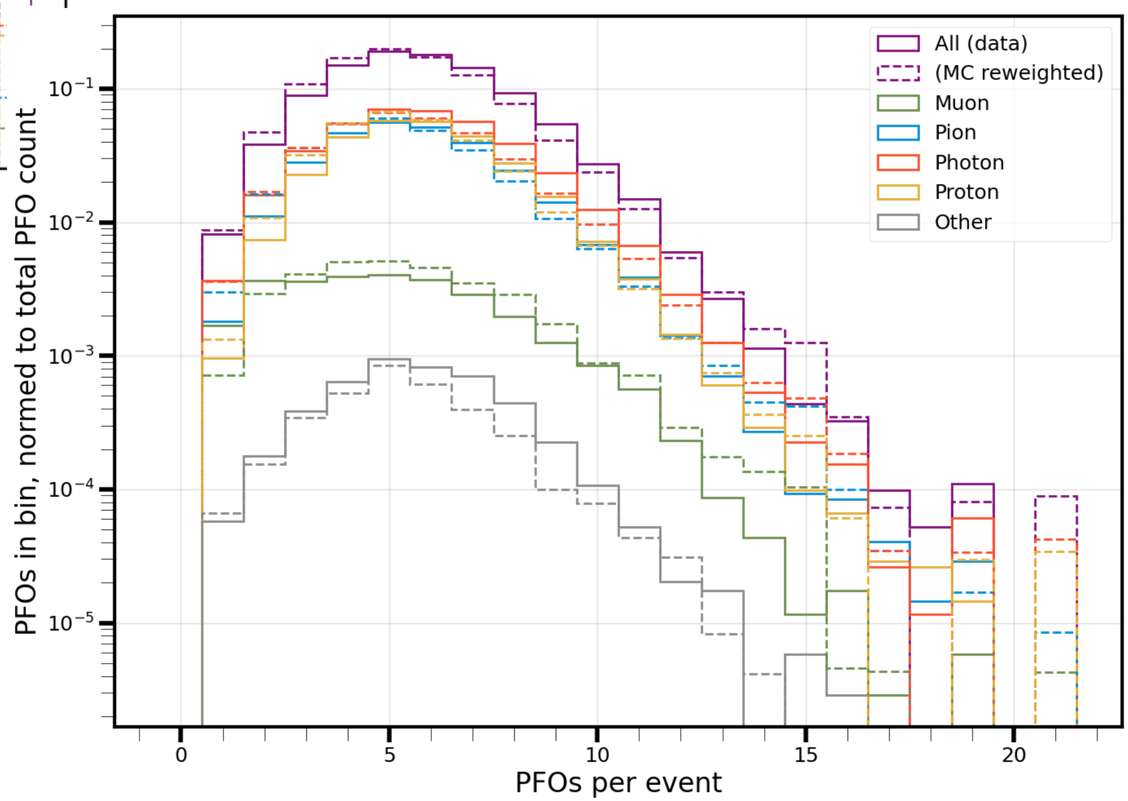
n: particular event

p: particle species



Reweighting events





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

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

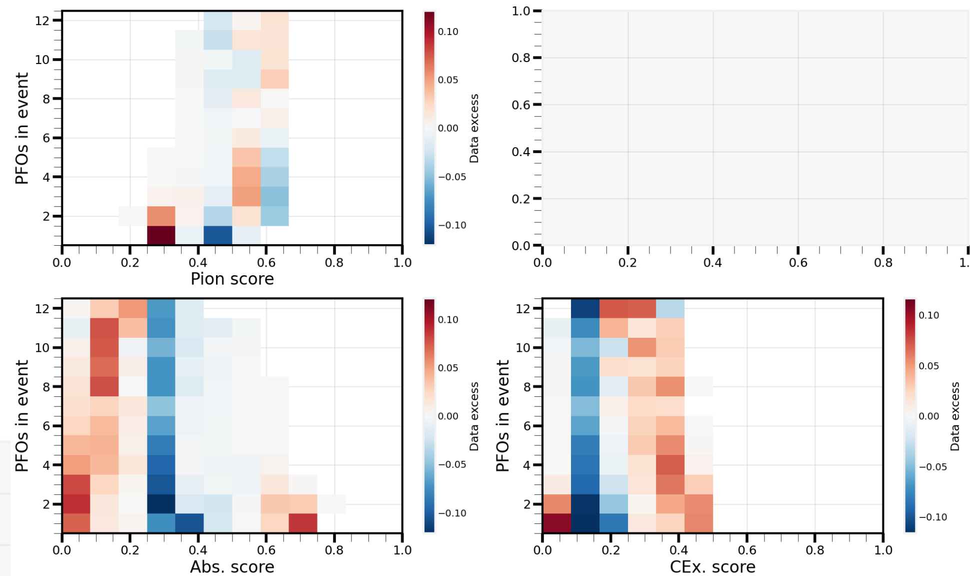
N: all events

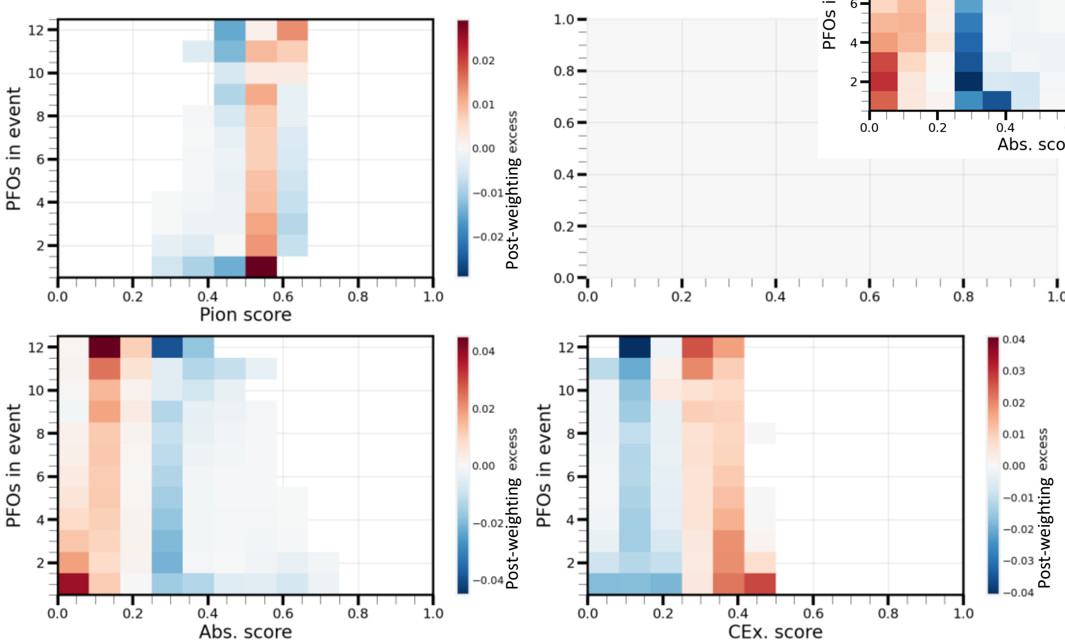
n: particular event

p: particle species

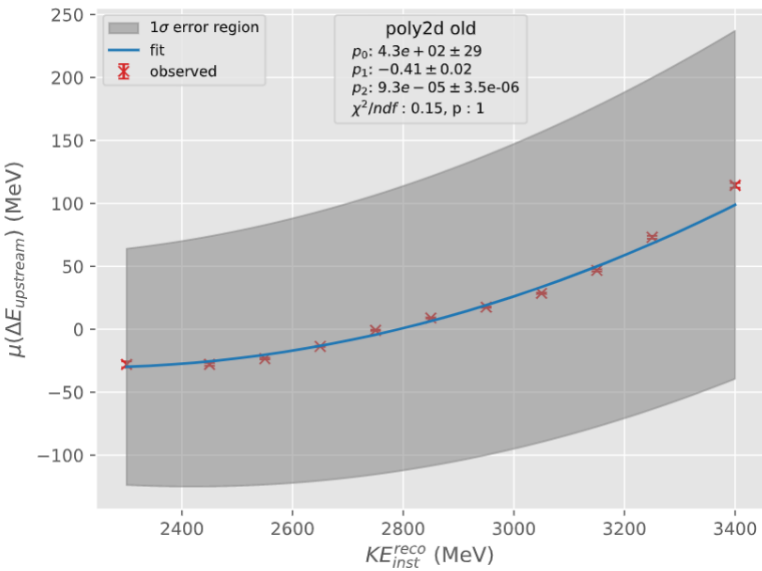
After weighting

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





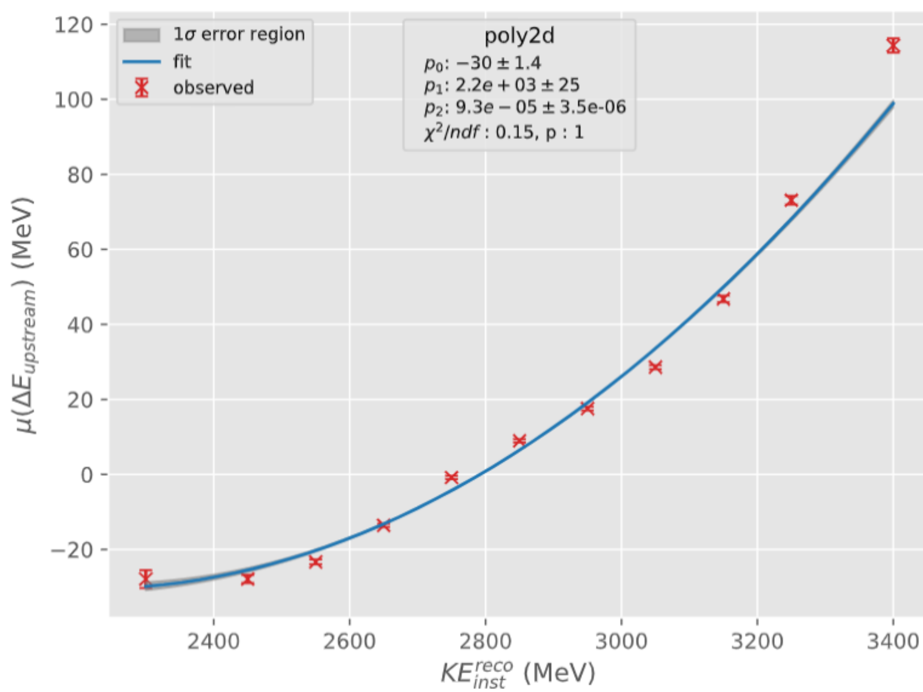
Upstream correction fit



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

Left: $p_2x^2 + \ p_1x + p_0$

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?