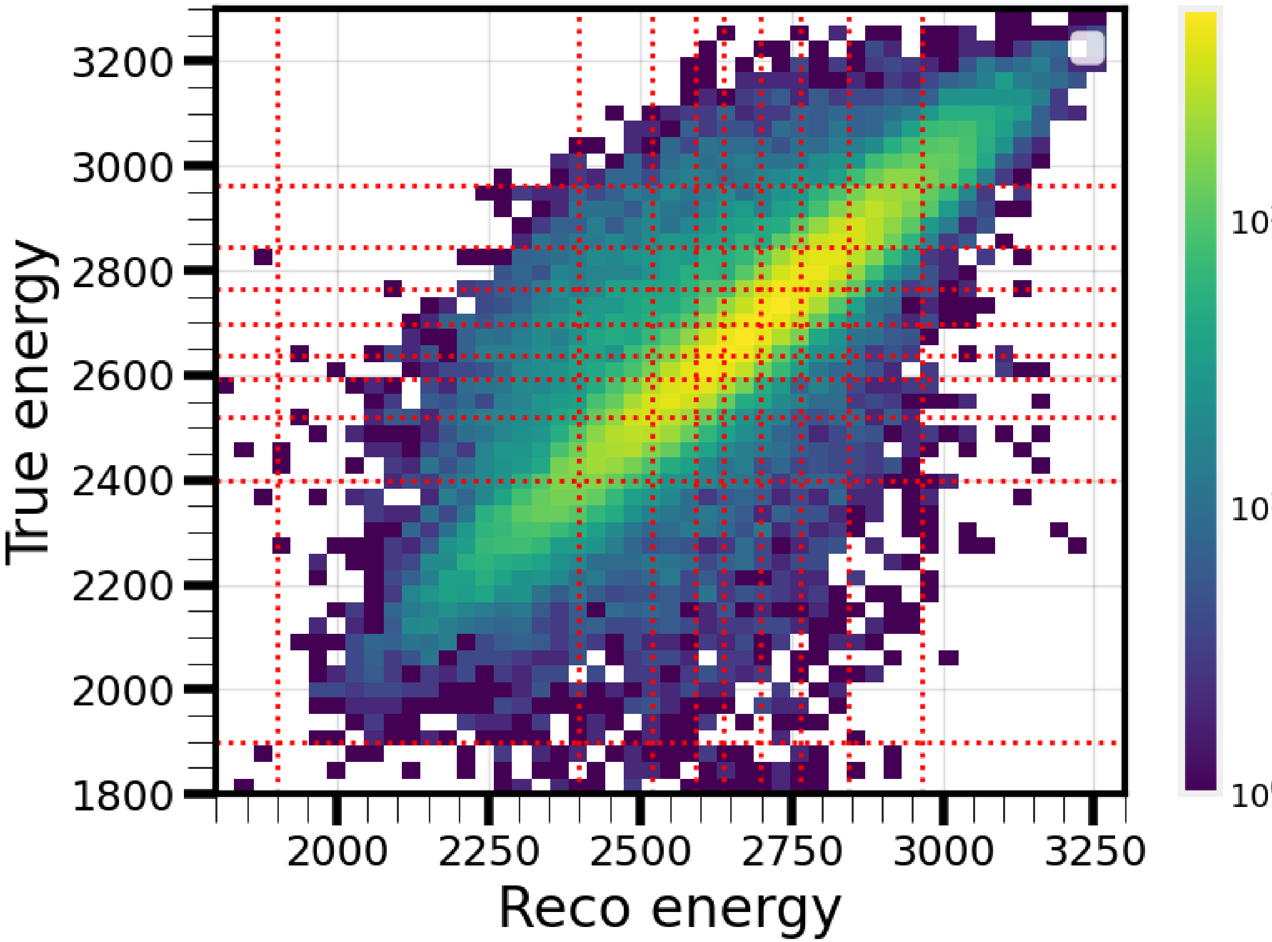
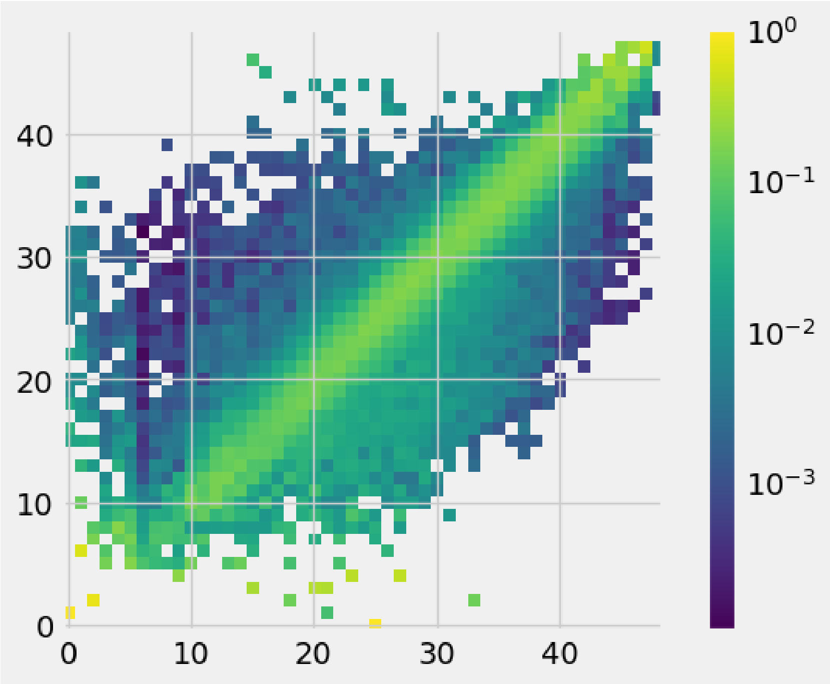
Unfolding tests

- Tried some unfolding with PyUnfold
- Aim to investigate coverage



Unfolding tests

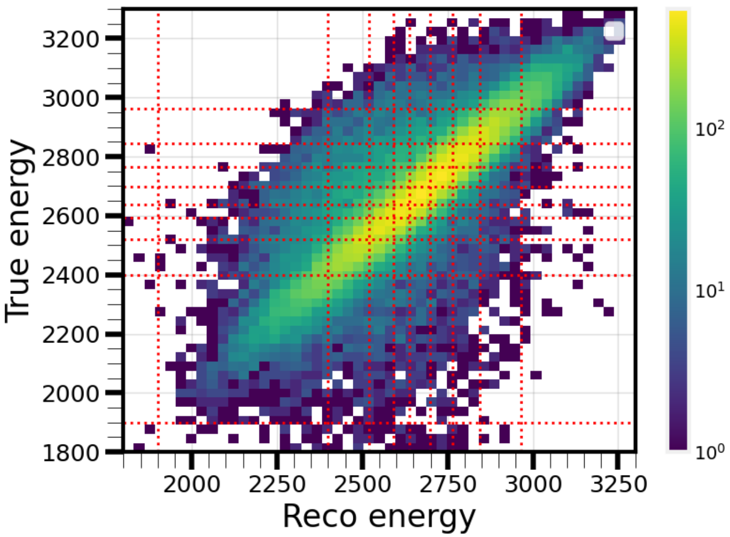
Unfolding matrix, iteration 30



- PyUnfold returns:

	unfolded	stat_err	sys_err	num_iterations	unfolding_matrix	ts_iter	ts_stopping
0	[1218.232532836397, 1234.6625135098056, 1058.1...	[7.696001794129402, 6.92093518151426, 6.724283...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	1	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	0.084544	1.000000e-09
1	[1228.1974196922615, 1243.6429697847236, 959.0...	[11.703983184421455, 9.48220859162182, 9.20765...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	2	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	0.066882	1.000000e-09
2	[1178.3509615608675, 1178.9590022673947, 846.5...	[15.868461241287548, 12.497441508648512, 11.84...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	3	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	0.042333	1.000000e-09
3	[1117.2439953474968, 1099.4230157304628, 749.9...	[19.62412857443053, 15.418507728940634, 14.023...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	4	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	0.025893	1.000000e-09
4	[1057.4211215470127, 1023.4812790766808, 670.6...	[22.933526582259, 18.162496958398275, 15.88832...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	5	[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...	0.016563	1.000000e-09
...	...	...	...	...	...	...	...

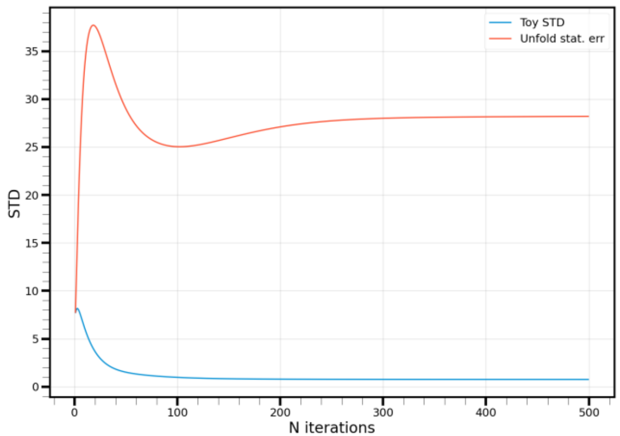
Unfolding tests

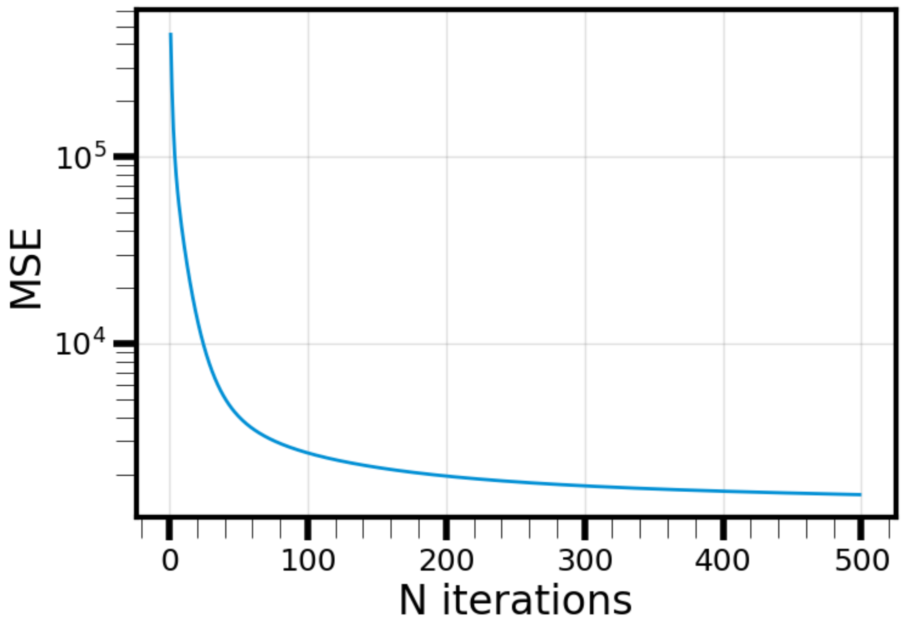


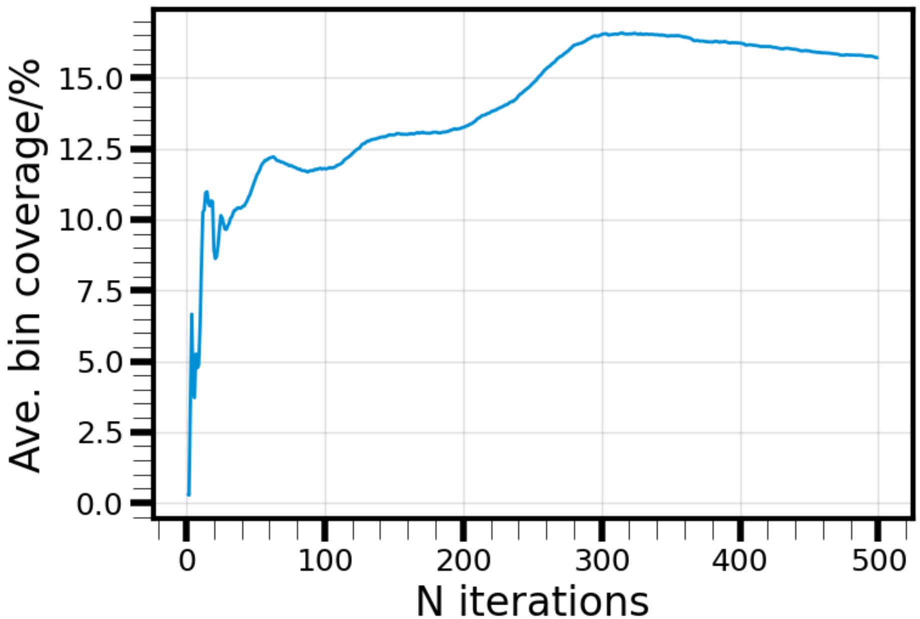
- Generate 1000 toys by Poisson drawing from the reconstructed counts.
  - Maybe bad way? I should probably Poisson drawn the true counts, and then applied the response matrix
- Generate the unfolding results by applying the unfolding matrix of each iteration.
- From 1000 toys, calculate bias and variance
  - Note with respect to the unchanged truth, hence I should have drawn from changed truth not reco.

Unfolding tests

- MSE is variance + bias^2
- Coverage underestimated
  - Likely poor calculation
- Uncertainties from PyUnfold >> calculated variances:







Unfolding

- Less information from PyUnfold (i.e. errors of the unfolding matrix).
- Fewer methods available
  - I would like to be able to try a simpler method (i.e. SVD)
- Installed RooUnfold with python bindings
- It doesn’t work with the current environment
- Need to export information, perform the unfolding in the ROOT env., then import back to original python env.

Iterative Bayesian Unfolding

- I do not believe we should accept any data input when constructing the unfolding:
- Result ($\hat{n}(C_i)$) is simply the unfolding matrix on the observed energies
- This is calculated from some prior $P_0(C_i)$

$$\hat{n}(\mathrm{C}_i) = \sum_{j=1}^{n_\mathrm{E}} M_{ij} n(\mathrm{E}_j),$$

where

$$M_{ij} = \frac{P(\mathrm{E}_j|\mathrm{C}_i) P_0(\mathrm{C}_i)}{\left[\sum_{l=1}^{n_\mathrm{E}} P(\mathrm{E}_l|\mathrm{C}_i)\right]\left[\sum_{l=1}^{n_\mathrm{C}} P(\mathrm{E}_j|\mathrm{C}_l) P_0(\mathrm{C}_l)\right]}.$$

M_{ji} can be seen as the terms of the *unfolding matrix* **M**, which is clearly *not* the mathematical inverse of the smearing matrix **S**. Let us examine the various contribu-

D'Agostini

Iterative Bayesian Unfolding

- Prior indicates our *lack of knowledge* of the true values.
- If we run MC with the true energy dist. As the prior, prior is not updated.
- Prior should not be overtrained, we need accurate uncertainty estimation – coverage is good!

$$\hat{n}(\mathrm{C}_i) = \sum_{j=1}^{n_\mathrm{E}} M_{ij} n(\mathrm{E}_j),$$

where

$$M_{ij} = \frac{P(\mathrm{E}_j|\mathrm{C}_i)P_0(\mathrm{C}_i)}{\left[\sum_{l=1}^{n_\mathrm{E}} P(\mathrm{E}_l|\mathrm{C}_i)\right]\left[\sum_{l=1}^{n_\mathrm{C}} P(\mathrm{E}_j|\mathrm{C}_l)P_0(\mathrm{C}_l)\right]}.$$

M_{ji} can be seen as the terms of the *unfolding matrix* **M**, which is clearly *not* the mathematical inverse of the smearing matrix **S**. Let us examine the various contribu-

D'Agostini

Iterative Bayesian Unfolding

- Would like to try a prior in which the theoretical expected interaction energies is used based on:
  - Measured beam energy distribution
  - Current total LAr-Pion cross section knowledge
- In addition, explore the output vs. input priors