BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

Machine Learning-Assisted Unfolding for Neutrino Cross Section Measurements

Roger Huang
Andrew Cudd, Masaki Kawaue, Tatsuya Kikawa, Ben Nachman, Callum Wilkinson

NPML 2024
ETH Zurich, Switzerland
June 27, 2024

Neutrino Cross-Section Measurements are Hard

- We aim to measure differential cross sections **(dσ/dx)$_i$** in some truth-level kinematic bins **x$_i$**
- What we actually measure is detector-reconstructed quantities are some number ***N*** signal events and ***B*** background events in each detector-level kinematic bin *j*, related by

$$\left(\frac{d\sigma}{dx}\right)_i = \frac{\sum_j \widetilde{U}_{ij}^{-1}\left(N_j - B_j\right)}{\Phi_\nu T \Delta x_i \epsilon_i}$$

for a response matrix **U**, total flux **Φ**, total target nuclei **T**, and detection efficiency ϵ

Unfolding is deciding how to invert this response matrix, deal with background subtraction, and do efficiency corrections, all of which can in general have **high-dimensional dependencies**

- More observables used in unfolding reduces marginalization over effective detector effects

Conventional methods struggle to unfold beyond 2 or 3 dimensions, as the number of required bins rapidly becomes unmanageable

Machine Learning Assistance - The Likelihood Ratio “Trick”

- Train a classifier using a weighted binary cross entropy loss function, where each event $\mathbf{x_i}$ with weight $\mathbf{w_i}$ has a true label $\mathbf{p_i} \subset \{0, 1\}$ and gets a network prediction of $\mathbf{q_i}$:

$$\mathbf{Loss(p_i, q_i) = -w_i * (p_i*log(q_i) + (1\text{-}p_i)*log(1\text{-}q_i))}$$

- If we train with dataset ***A*** with labels 1 and dataset ***B*** with labels 0, then we can reweight each of the events $\mathbf{x_i}$ in ***B*** by the likelihood ratio:

$$\mathcal{L}\,\mathbf{[A,B](x_i) = p_A(x_i) / p_B(x_i) \approx q_i / (1\text{-}q_i)}$$

- Converts the **difficult problem** of multidimensional density estimation into the **“easy” problem** of classification!

OmniFold - Concept

- With some given generator and detector simulation, we can train classifiers using the likelihood ratio trick to do **event-by-event reweighting** of the generated events to fit the observed data
  - Classifier is effectively unrestricted in the number of variables it uses in its decisions, allowing us to unfold in very high dimensional space
- ***Automatically*** get background subtraction and efficiency correction
- From the reweighted generator events, we can then extract **unbinned unfolded results of any observable**
  - But the usual warnings apply about extracting observables in phase regions of very low efficiency

For more info:

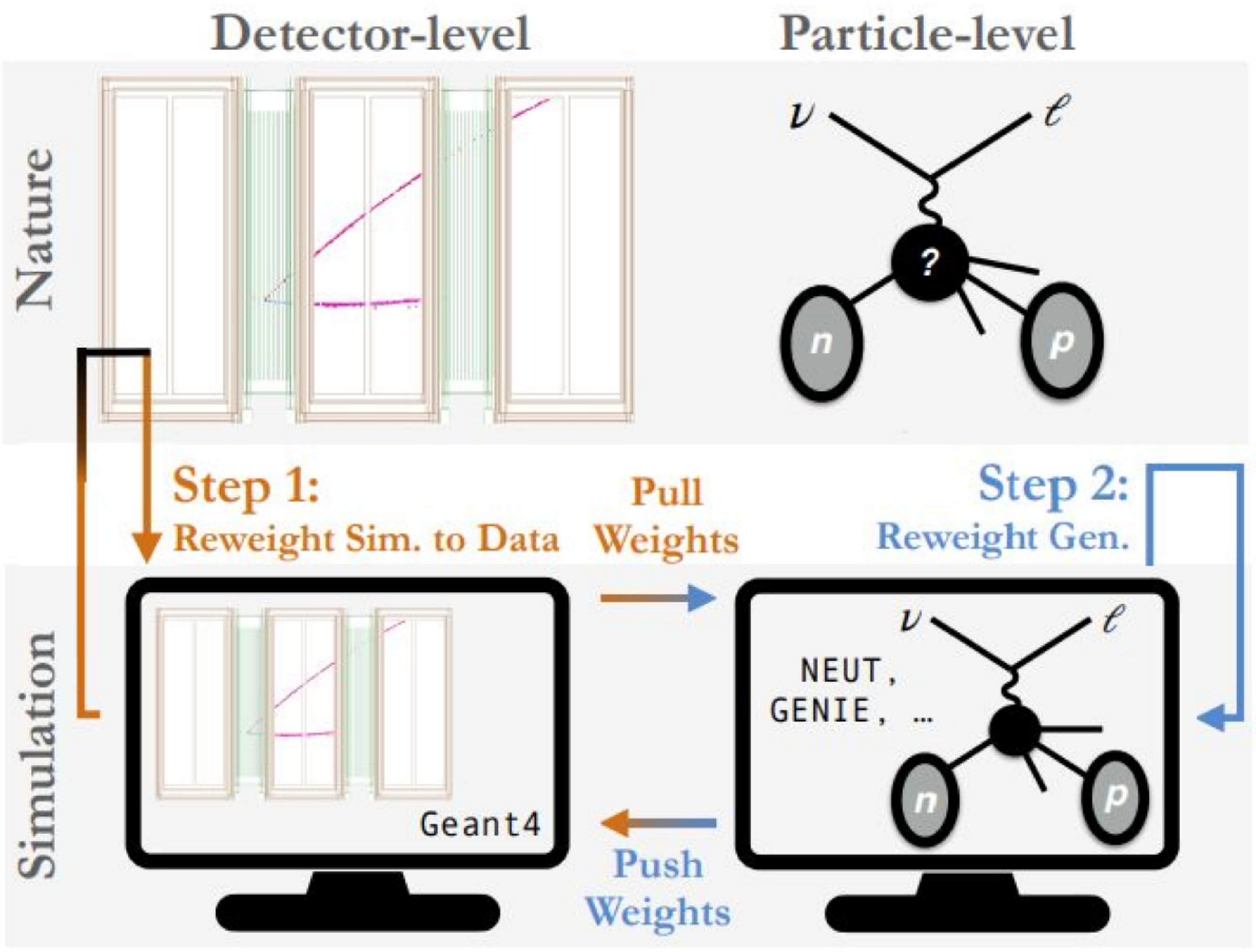
- Original paper https://doi.org/10.1103/PhysRevLett.124.182001
- Code release with implementation example https://github.com/hep-lbdl/OmniFold

OmniFold - Outline

1. $\omega_n(m) = \nu_{n-1}^{\text{push}}(m)\, L[(1, \text{Data}), (\nu_{n-1}^{\text{push}}, \text{Sim.})](m),$

2. $\nu_n(t) = \nu_{n-1}(t)\, L[(\omega_n^{\text{pull}}, \text{Gen.}), (\nu_{n-1}, \text{Gen.})](t).$

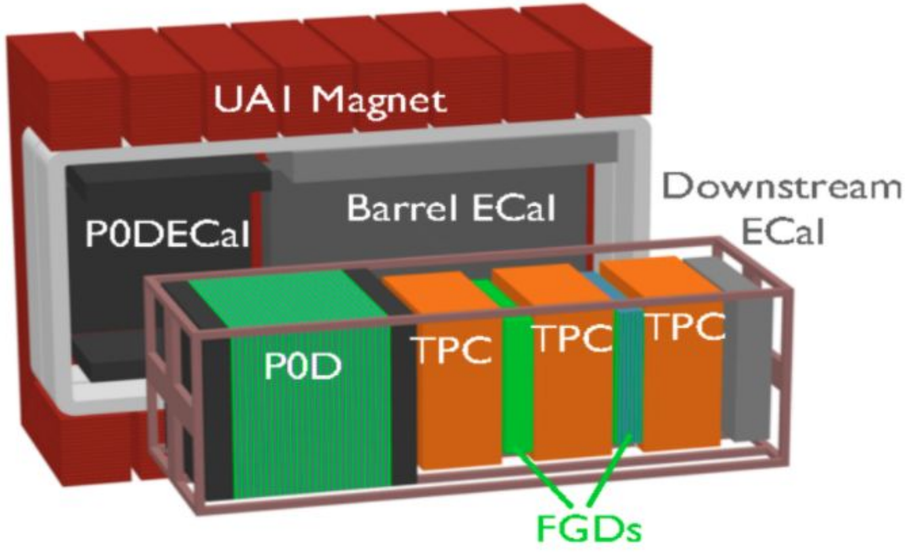
- OmniFold iterates on the above 2 steps to obtain *pull weights* $\boldsymbol{\omega_n}$**(m)** and *push weights* $\boldsymbol{\nu_n}$**(t)** for each simulated event with reconstructed value **m** and true value **t**
- A set of final push weights reweights the entire set of generated events to provide our unfolded result

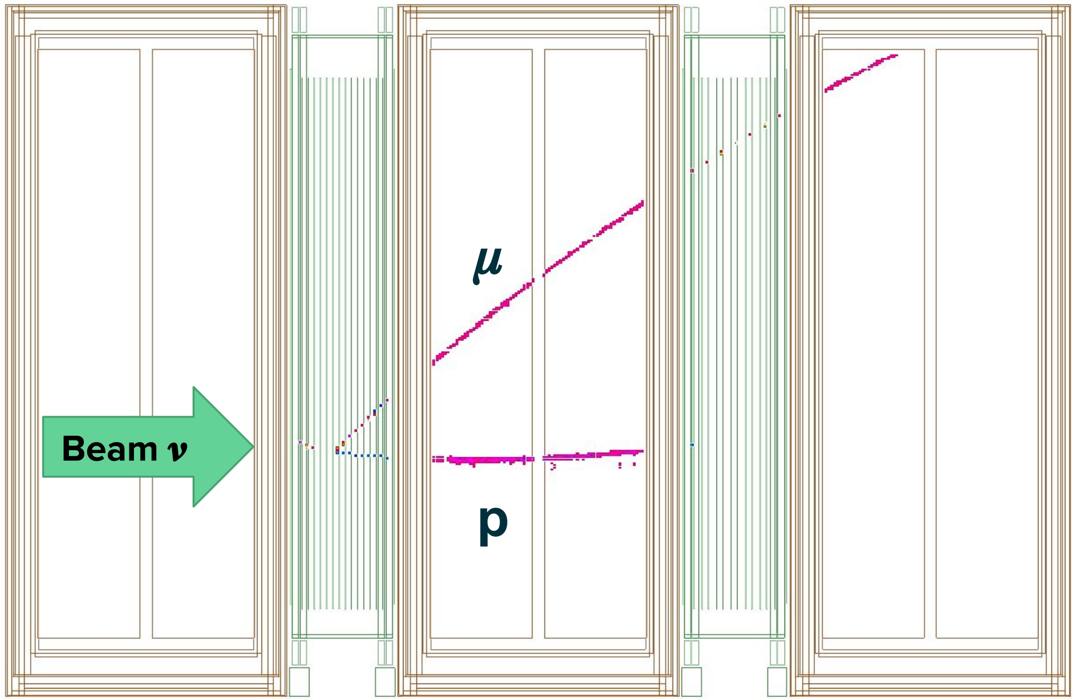


*Work funded by US-Japan grant

Test Setup - T2K Public Dataset

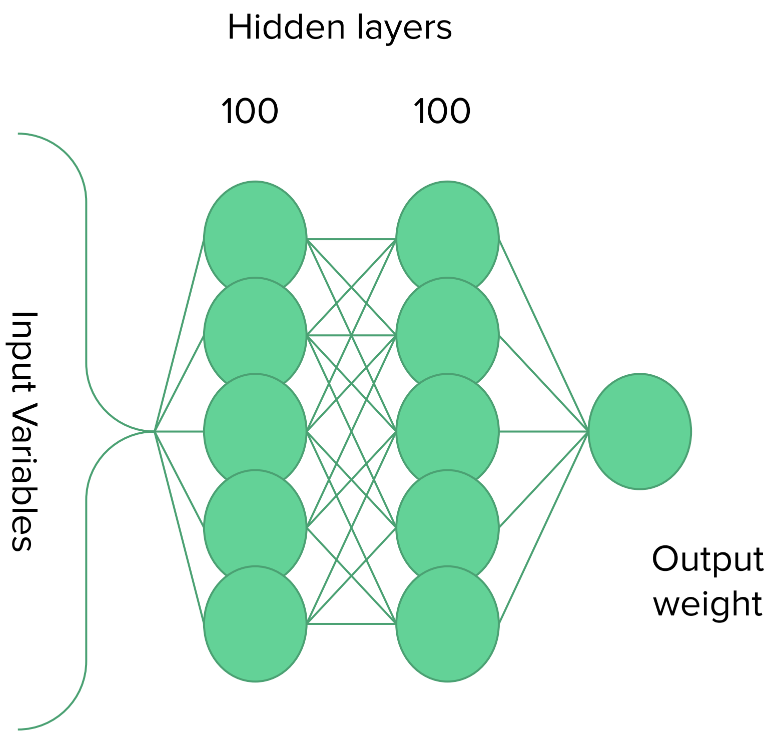
- Using public dataset of ~1.2 million T2K simulated near-detector events that was used in a ν_μ CC0π 2D differential cross-section analysis [1]
- Dataset includes muon and leading proton kinematics for each event
  - Also info about which subdetector the muon and proton were each reconstructed in
- We randomly divide the dataset in half and **set aside one half as "data"** for the unfolding procedure
- Create sets of **fake data** by applying various reweights to the "data"
  - Compare against the truth values after unfolding to evaluate performance
- From the remaining half that is considered simulation, generate 100 throws (pseudo-experiments) varying systematic and statistical uncertainties
  - Systematics include flux, cross-section, detector response uncertainties





[1] https://doi.org/10.1103/PhysRevD.101.112001

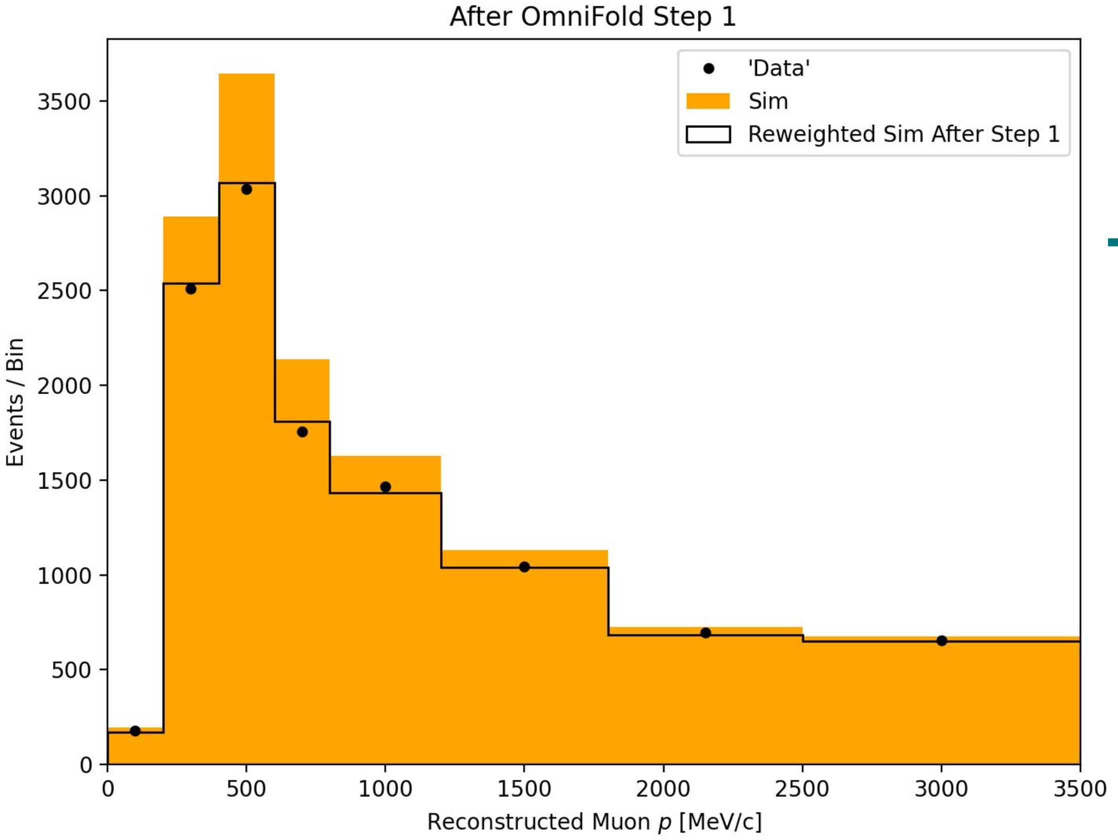
OmniFold Classifier Setup



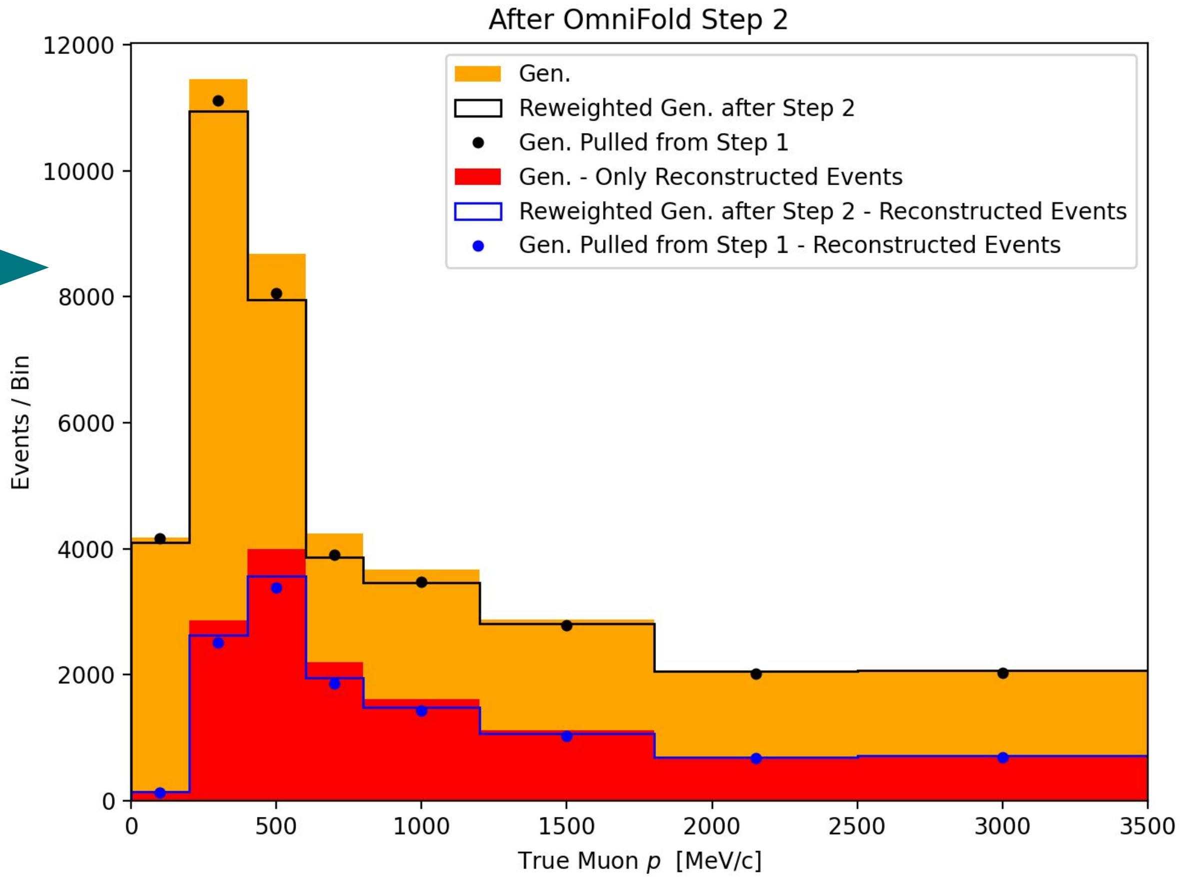
- Neural network structure: simple MLP with 2 hidden layers of 100 nodes each
- Input variables (everything in the public dataset):
  - **Primary muon** total momentum, cos θ (forward angle), ϕ (transverse angle)
  - **Leading proton** total momentum, cos θ, ϕ. Values set to 0 when no proton present
  - For detector space only: **Detector sample ID** (1-hot encoded, out of 8 possible sample IDs)
- Using one NVIDIA A100 on a NERSC Perlmutter node, takes 0.5-1 hours to run 15 OmniFold iterations on one set of data/MC
  - Multiply by desired number of throws and neural network trials

OmniFold Procedure Example

Step 1: reweight simulated reconstructed MC to observed data



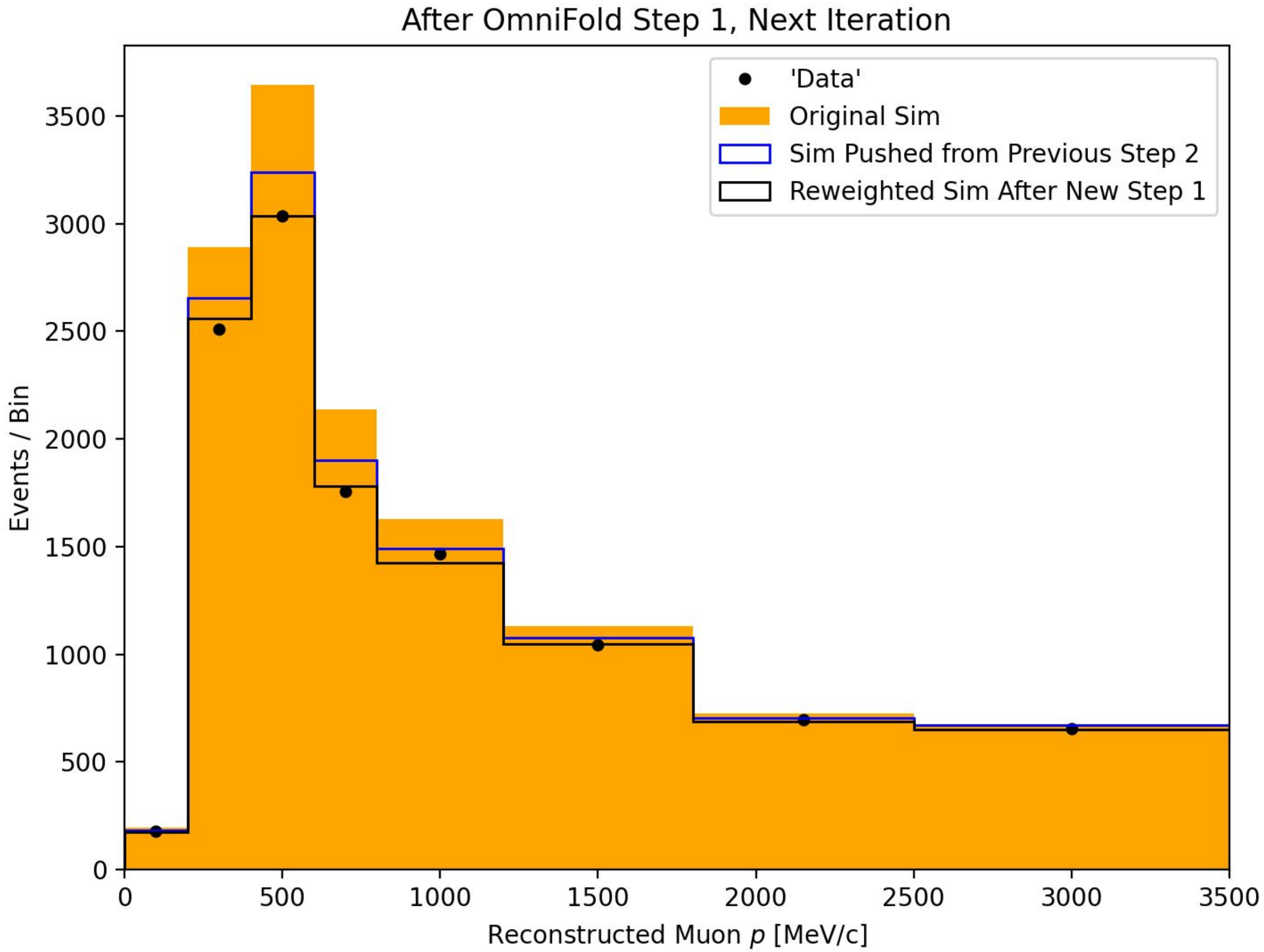
Step 2: reweight generator-level MC to itself with pulled reweighting factors from step 1



Reminder: the reweighting procedure is unbinned!

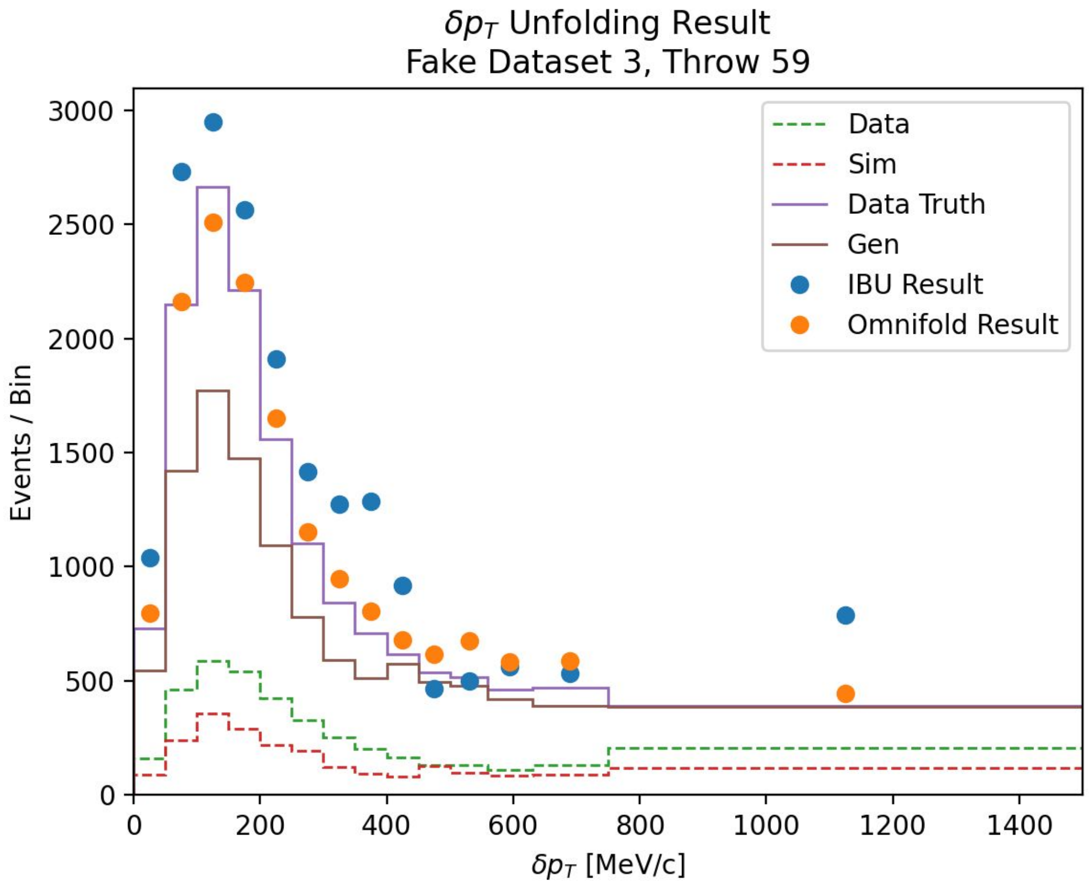
OmniFold Procedure Example

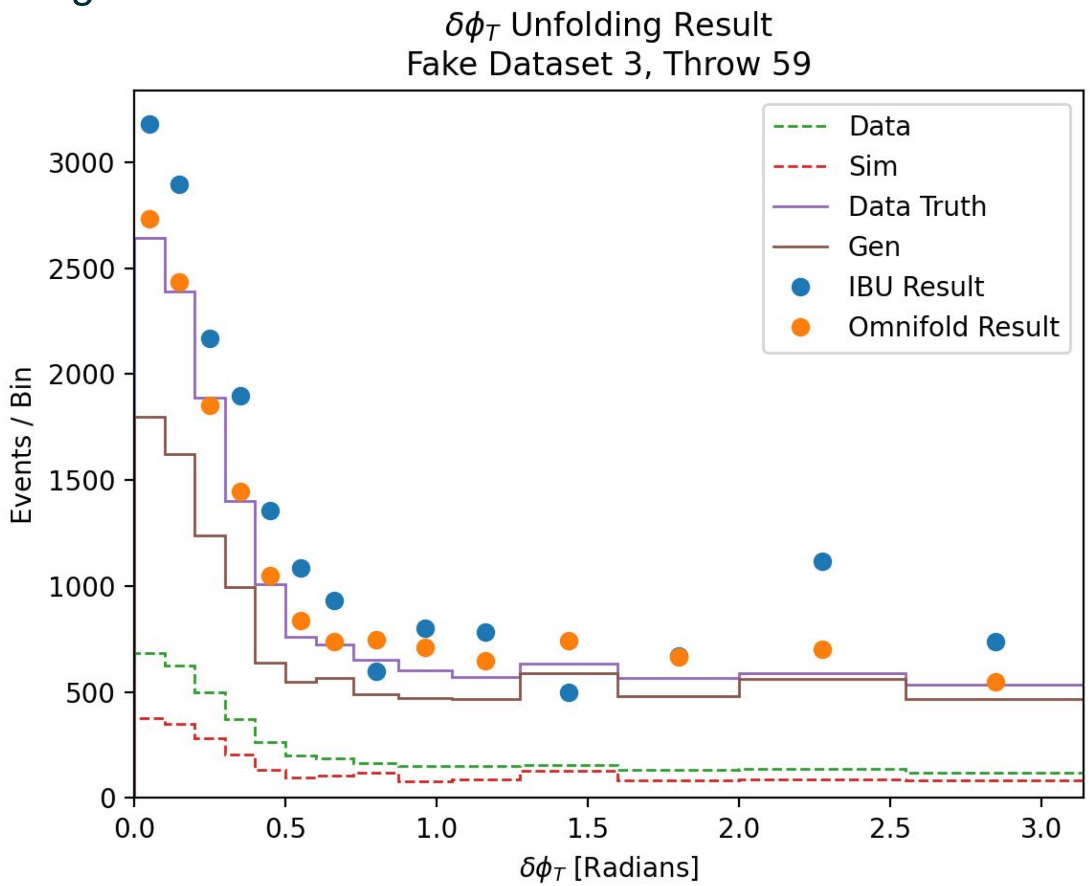
- **One iteration** is a step 1 + step 2 combo
- On a new iteration, go to step 1 again, but starting with pushed reweighting factors on the simulated reconstructed events from the previous iteration's step 2 result
- Repeat for any number of iterations
  - **Regularization** comes from a cutoff on the number of iterations, based on some chosen convergence criterion



Example Result - Single MC Throw

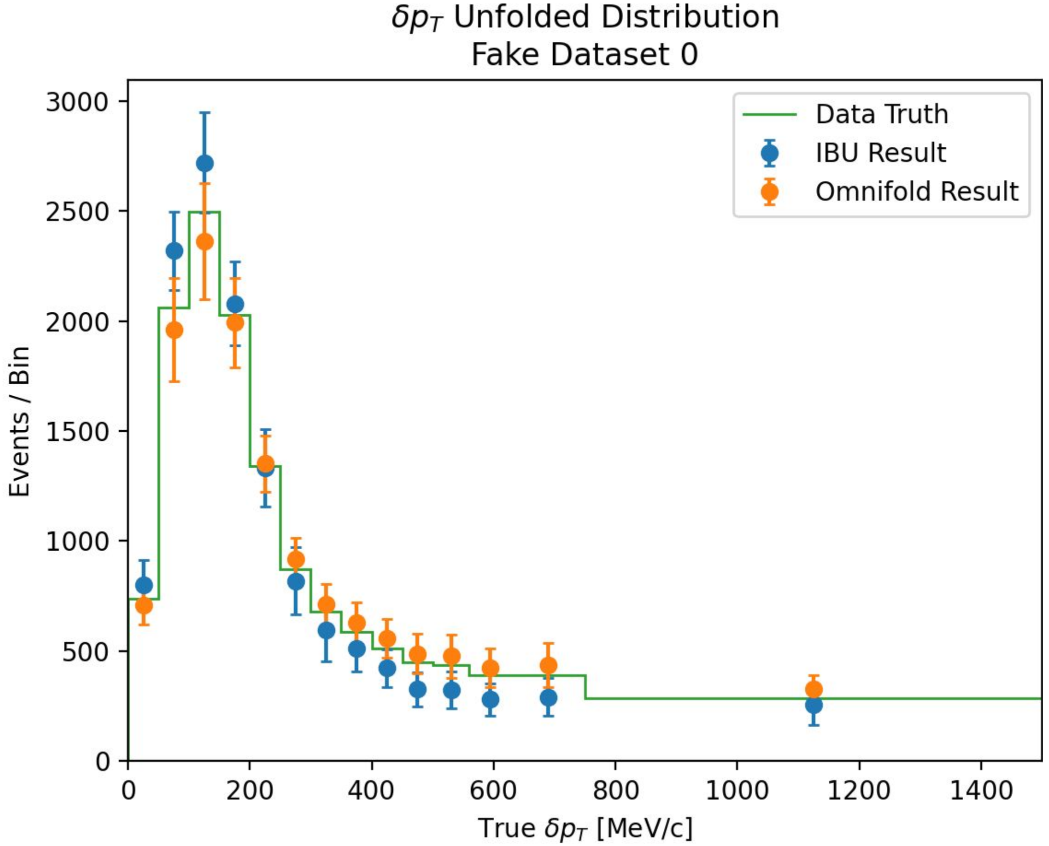
- Compare against result of Iterative Bayesian Unfolding (**IBU**) (d'Agostini unfolding) as a conventional benchmark
  - Note: IBU result for each variable comes from a separate unfolding, whereas OmniFold gets all observables from one unfolding
- OmniFold result looks less sensitive to fluctuations in the single variable distribution

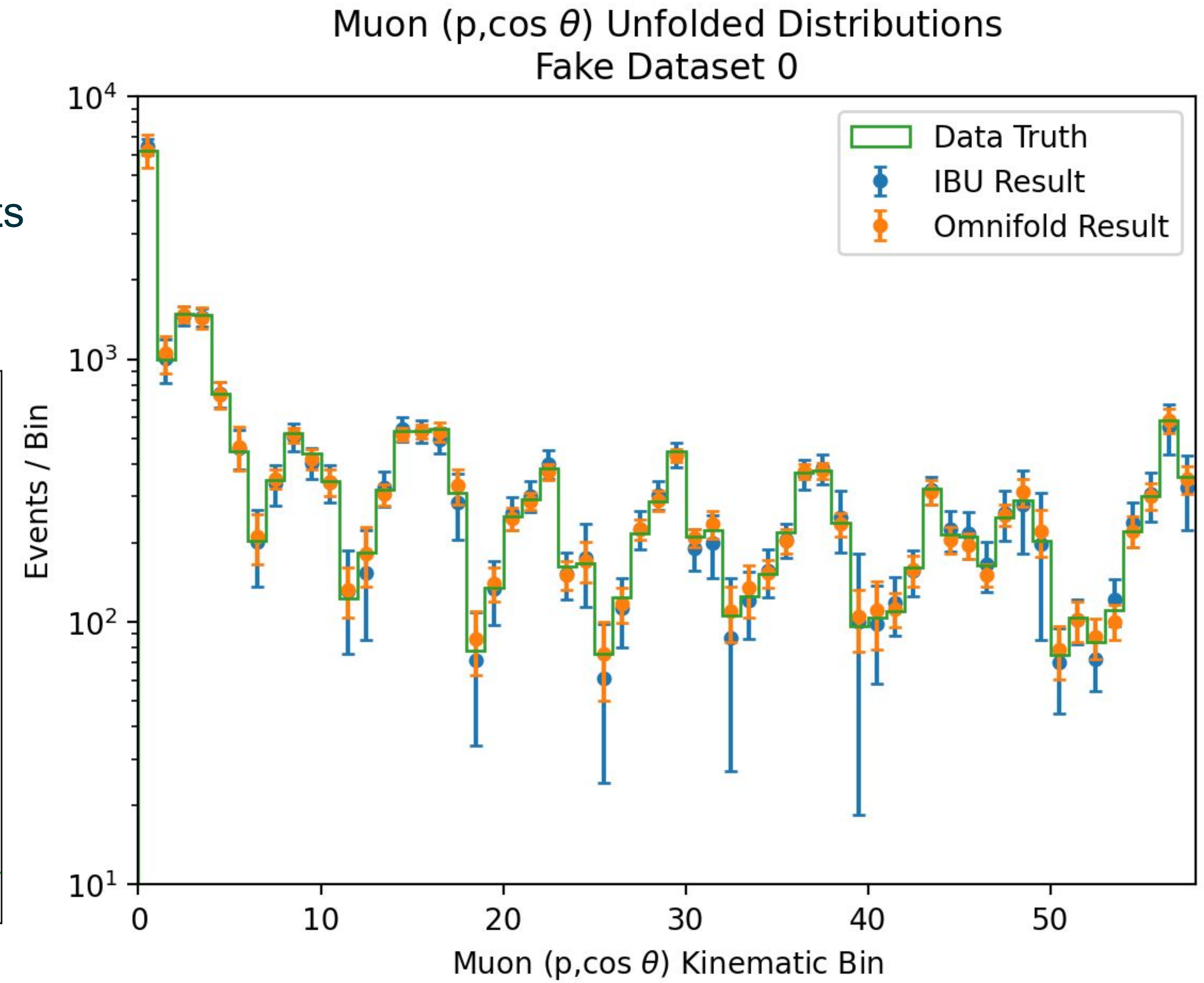




Overall Result with Syst./Stat. Variations

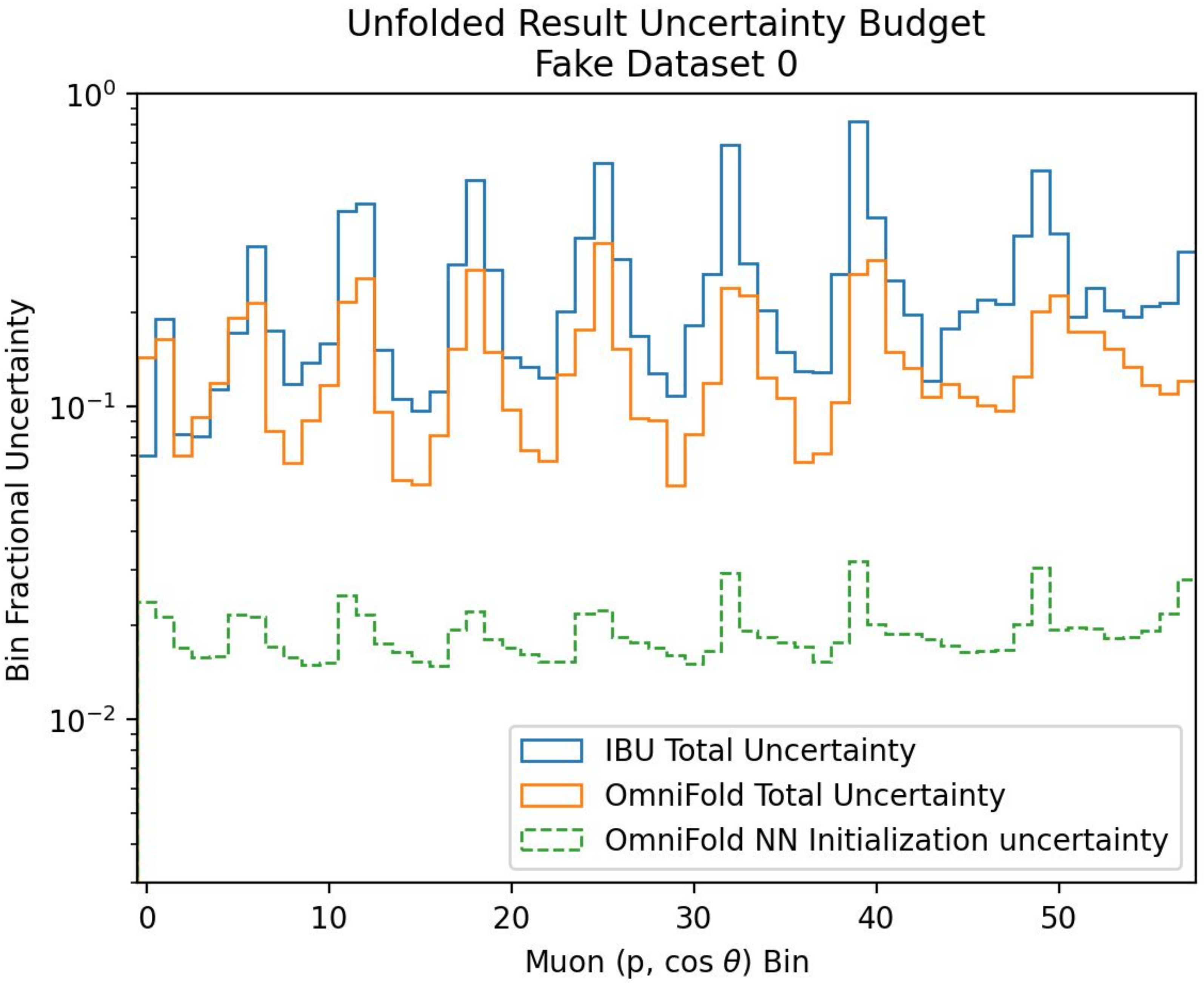
- Error bars show spread from results of 100 syst./stat. variations





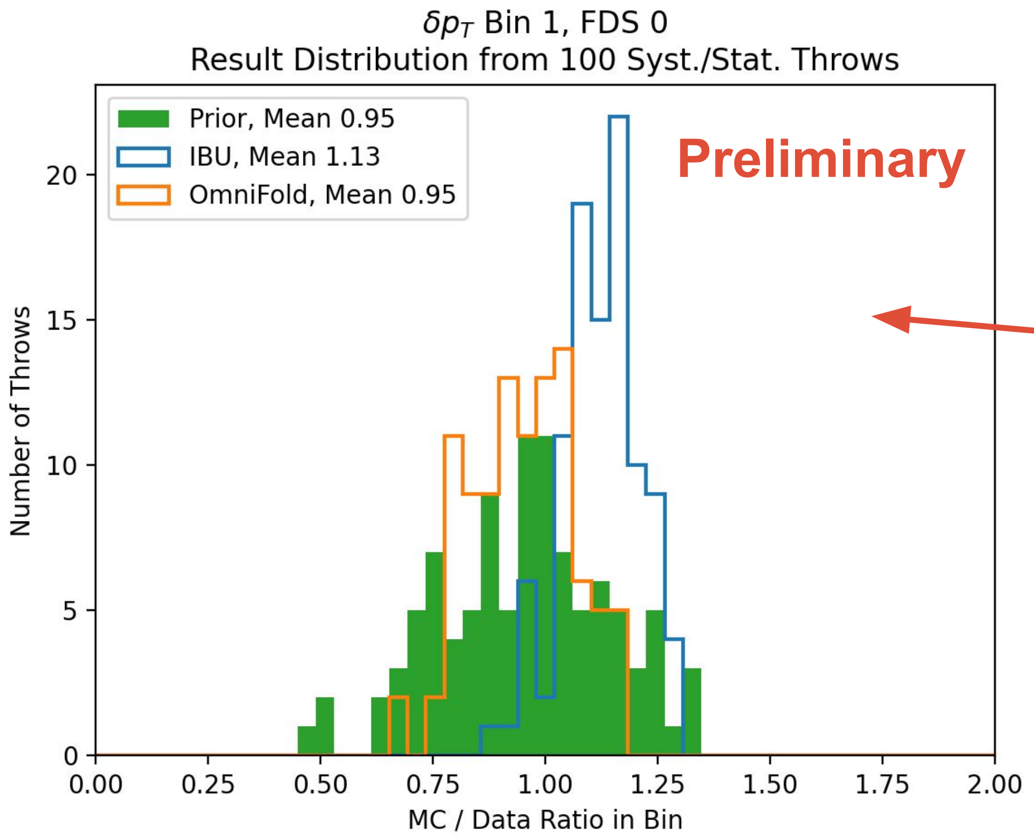
Uncertainty Budget

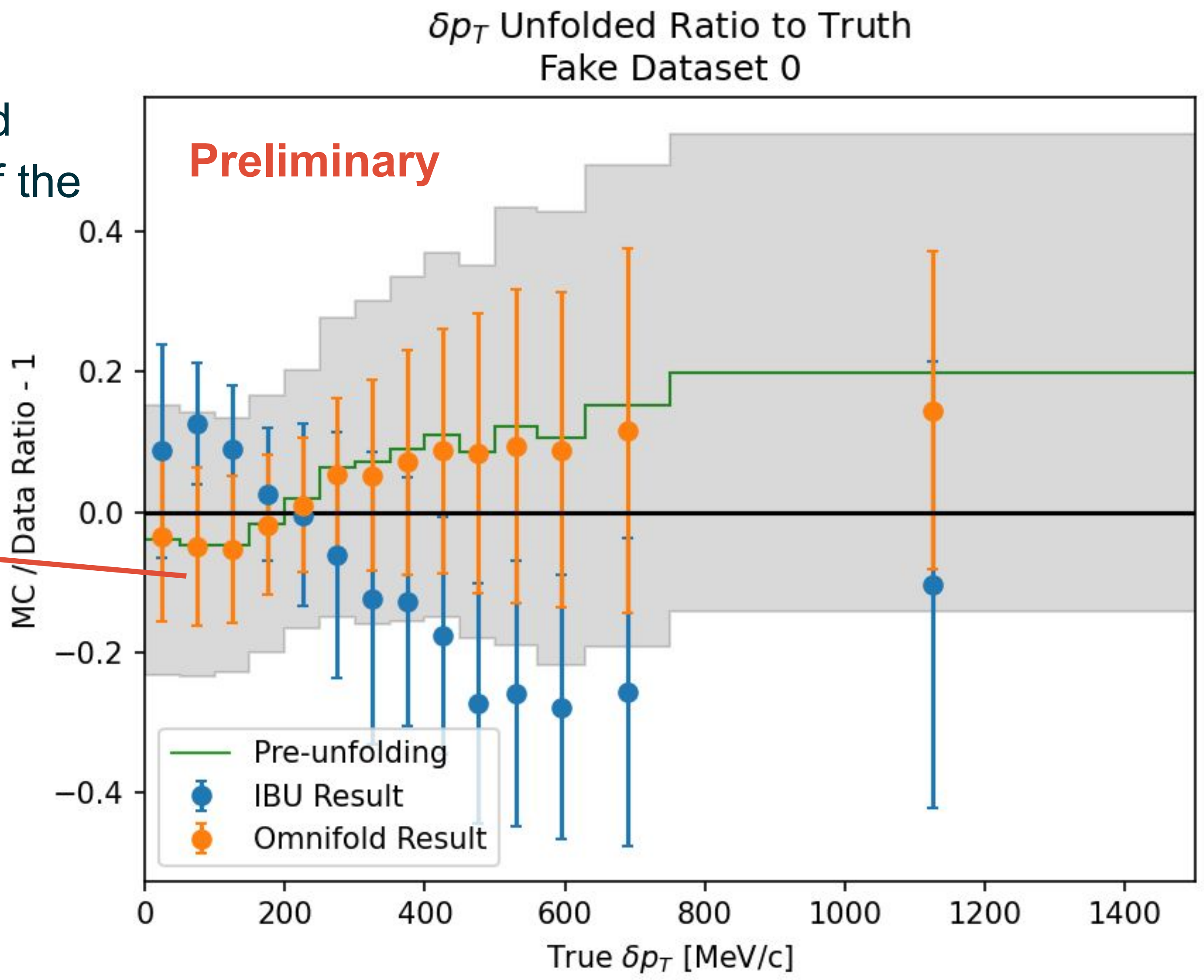
- **Total uncertainty** is spread in results from adding systematic + statistical variations to MC
  - OmniFold uncertainties notably lower than conventional IBU
- **Aleatoric uncertainty** from randomness of network initialization and training should be rendered negligible by averaging the results of multiple trials
  - NN initialization uncertainties here are from averaging 5 trials



Bin-by-bin Results - δp

- Looking at per-bin results, OmniFold does a better job of bringing each of the MC throws closer to data truth





Performance Comparison

- Calculate χ^2 for each binned variable of interest using full covariance matrix from unfolding fake data to 100 MC throws with systematic/statistical variations
  - Note: for a comparison in 4 variables for a single fake dataset, IBU had to be run 4 separate times while OmniFold gets all of them from a single unfolding pass-through
- OmniFold achieves comparable or better performance than conventional IBU in almost all cases

Fake Dataset	Unfolding Method	Muon (p,θ) χ^2 (DoF=58)	δp_T χ^2 (DoF=14)	$\delta\alpha_T$ χ^2 (DoF=18)	$\delta\phi_T$ χ^2 (DoF=14)
Shape only $\delta\phi_T$	IBU	**13.6**	7.3	1.9	6.4
	Omnifold	15.1	**1.3**	**1.6**	**1.7**
Shape + Norm. $\delta\phi_T$	IBU	64.8	18.3	14.1	20.9
	Omnifold	**23.9**	**1.7**	**2.5**	**2.9**

Summary

- OmniFold provides a general method to perform high-dimensional unfolding of several (all) observables simultaneously
  - This has been used in collider physics already, but neutrino experiments can similarly benefit from its advantages
- We have demonstrated for the first time OmniFold's application to a neutrino cross-section measurement using a public T2K near-detector simulated dataset
  - OmniFold has comparable or better performance than a conventional approach across several observables with several fake data tests
  - Paper and code release in preparation!
- Next steps for application to real data:
  - Improvements to information included in MC
  - Possible improvements to classifier choice
  - Dealing with low statistics of real neutrino data

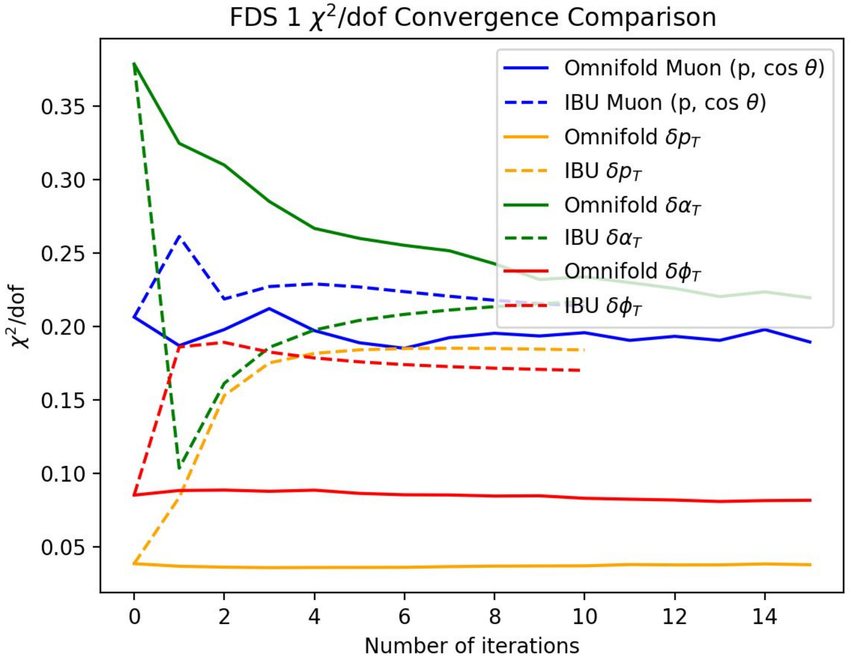
BERKELEY LAB
Bringing Science Solutions to the World

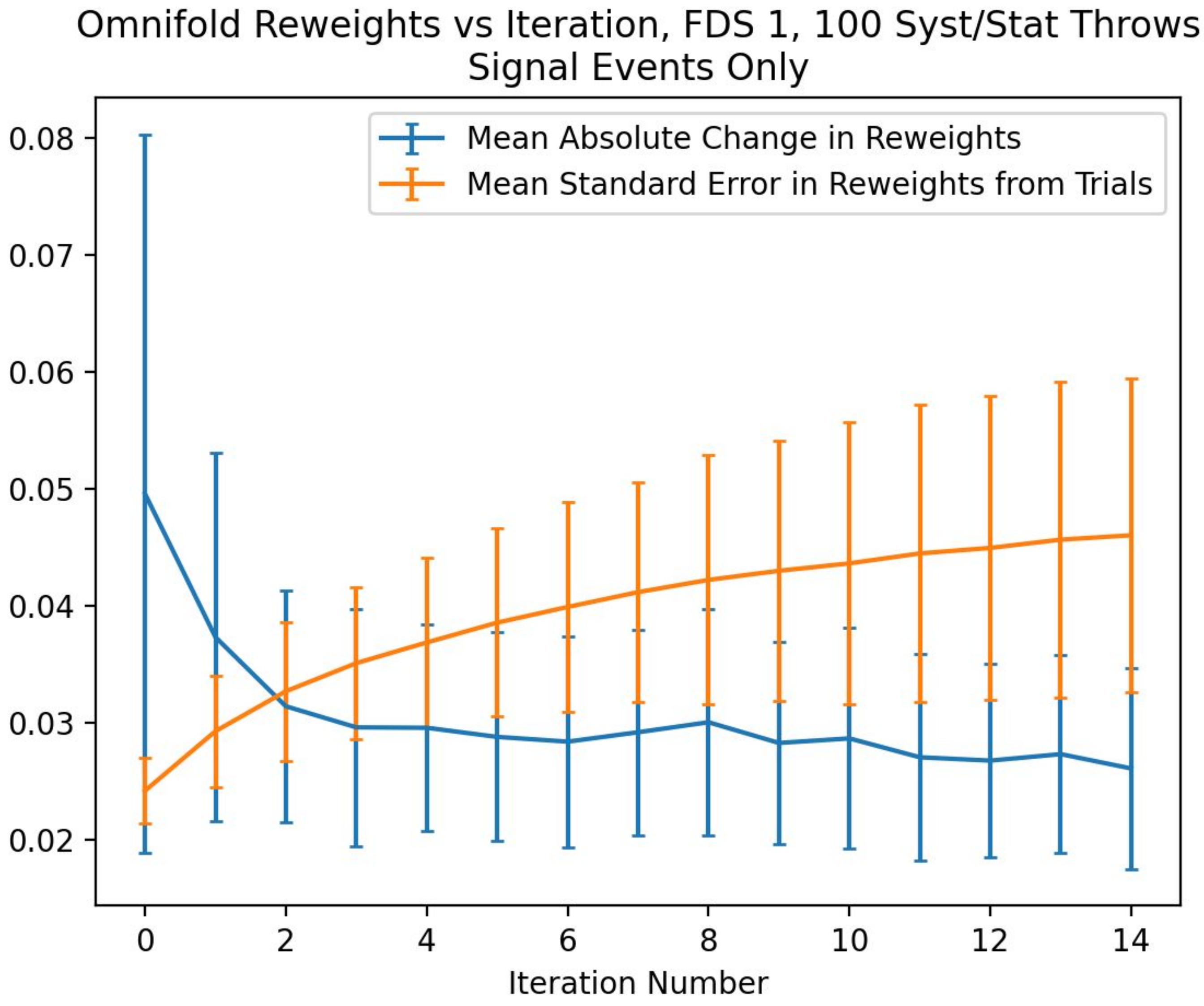
U.S. DEPARTMENT OF ENERGY
Office of Science

Backup

OmniFold Convergence

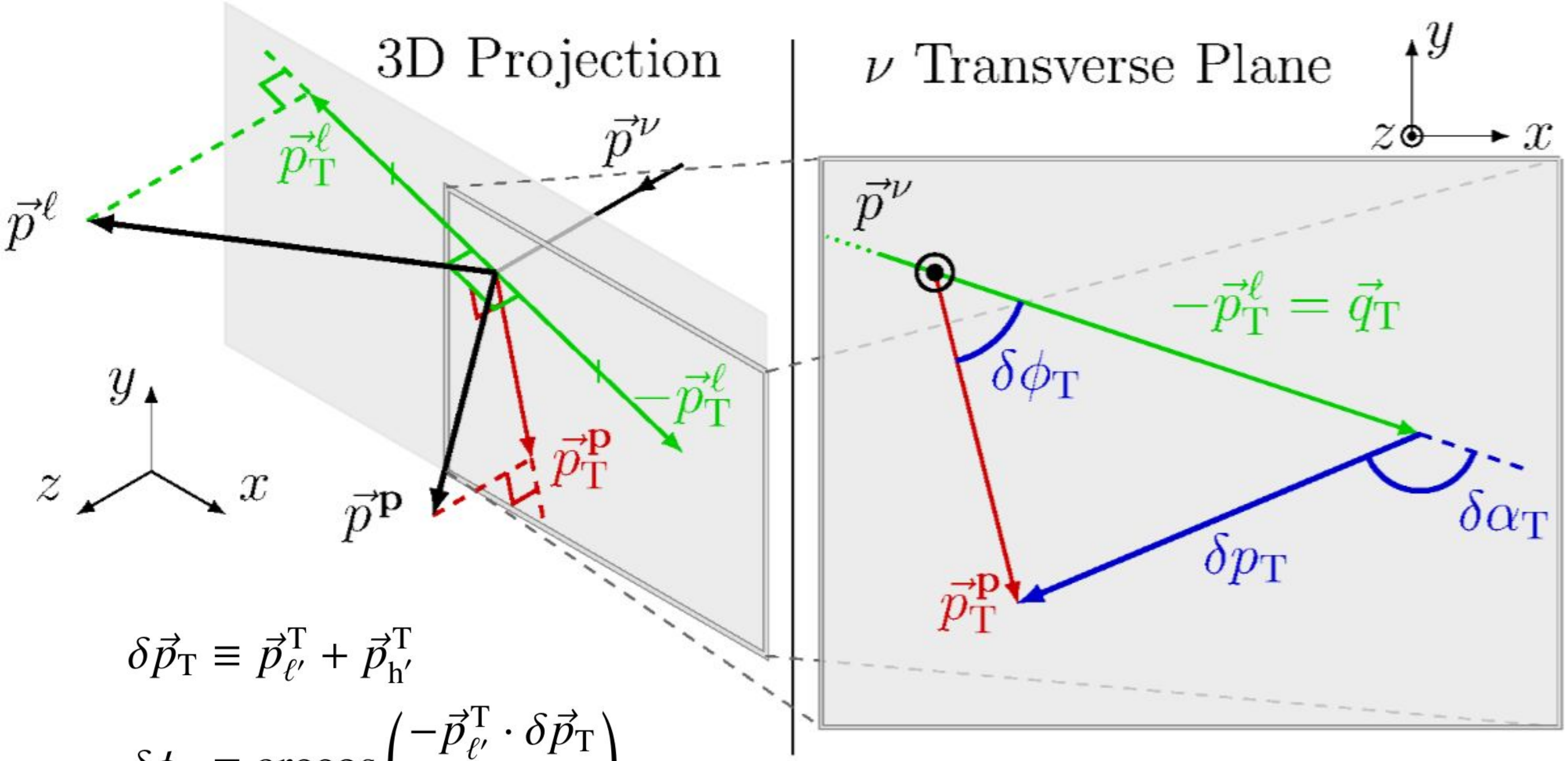
- **Data-driven convergence criterion** for OmniFold: when reweighting variation from NN initialization/training effects is larger than overall change in reweighting
- Resulting cutoff seems reasonable on the χ^2 curves





Single-Transverse Kinematic Variables

- Single-transverse variables (**STVs**) are measures of the kinematic imbalance between the primary lepton and hadron in a CC interaction
- These variables serve as a good proxy for studying various nuclear effects
  - No imbalances in the absence of nuclear effects



$$\delta\vec{p}_{\mathrm{T}} \equiv \vec{p}^{\,\mathrm{T}}_{\ell'} + \vec{p}^{\,\mathrm{T}}_{\mathrm{h}'}$$

$$\delta\phi_{\mathrm{T}} \equiv \arccos\left(\frac{-\vec{p}^{\,\mathrm{T}}_{\ell'} \cdot \delta\vec{p}_{\mathrm{T}}}{p^{\mathrm{T}}_{\ell'}\,\delta p_{\mathrm{T}}}\right)$$

$$\delta\alpha_{\mathrm{T}} \equiv \arccos\left(\frac{-\vec{p}^{\,\mathrm{T}}_{\ell'} \cdot \vec{p}^{\,\mathrm{T}}_{\mathrm{h}'}}{p^{\mathrm{T}}_{\ell'}\,p^{\mathrm{T}}_{\mathrm{h}'}}\right)$$

Iterative Bayesian Unfolding (IBU)

- Also goes by other names: d'Agostini unfolding, Lucy-Richardson deconvolution
- Given data, response matrix, and prior (all binned): iteratively perform Bayesian updates on the prior
- OmniFold is mathematically an unbinned version of this approach

In the IBU approach for our tests:

- Estimate binned background from MC and subtract it from the data
- Perform bin-by-bin efficiency corrections

Simplified:

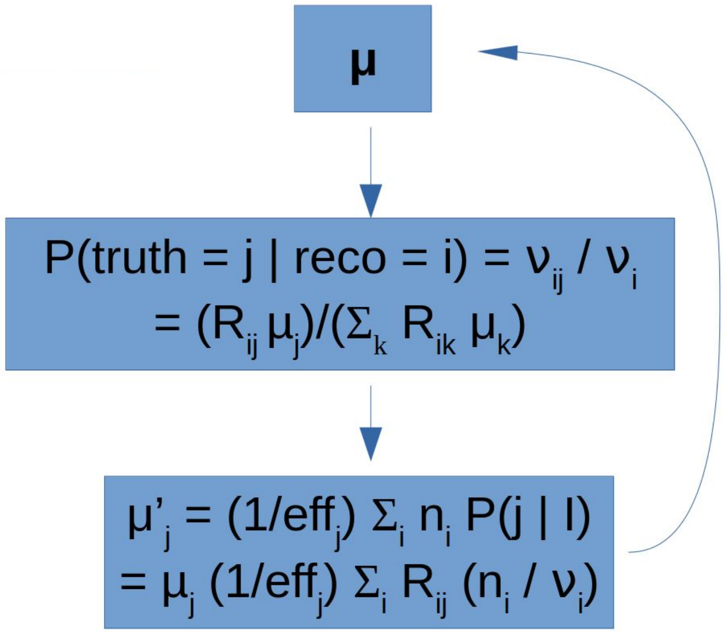


Diagram source