Machine Learning Based Reweighting Methods for Neutrino Physics

Roger Huang

BNL Particle Physics Seminar

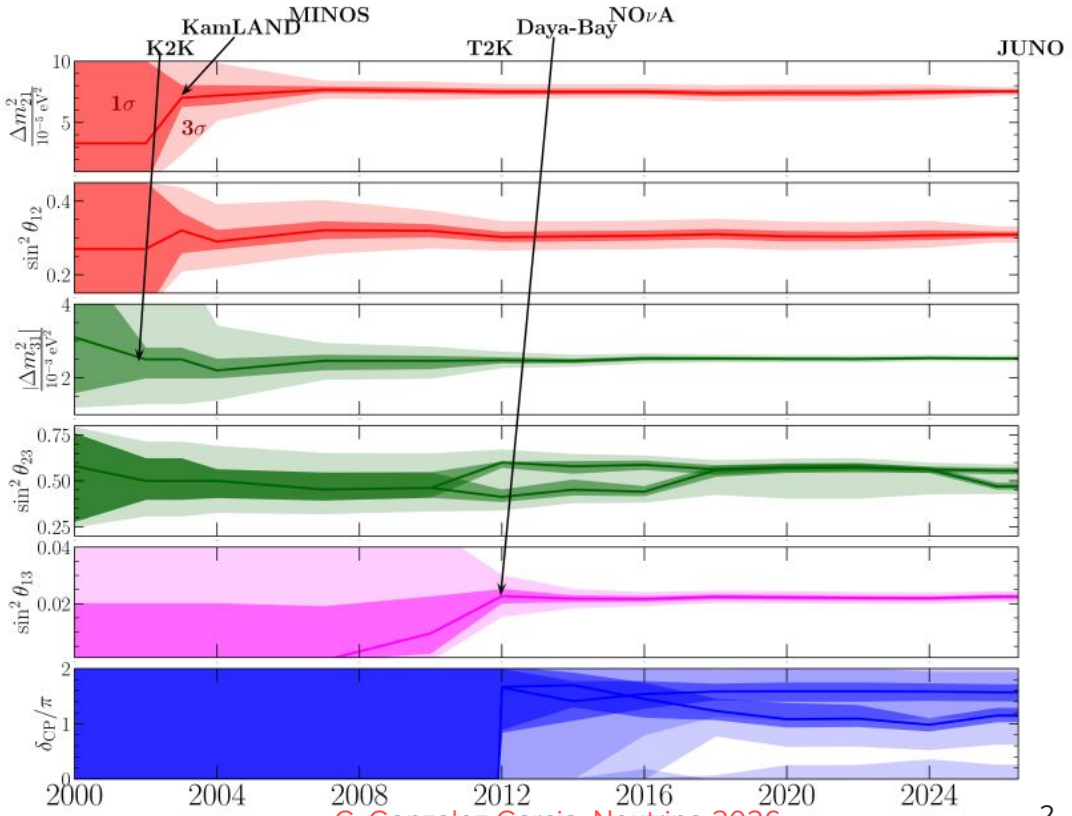
8/27/2026

$$\begin{bmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{bmatrix} \begin{bmatrix} c_{13} & 0 & s_{13}e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta_{CP}} & 0 & c_{13} \end{bmatrix} \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{bmatrix}$$

The Current State of Neutrino Oscillation Measurements

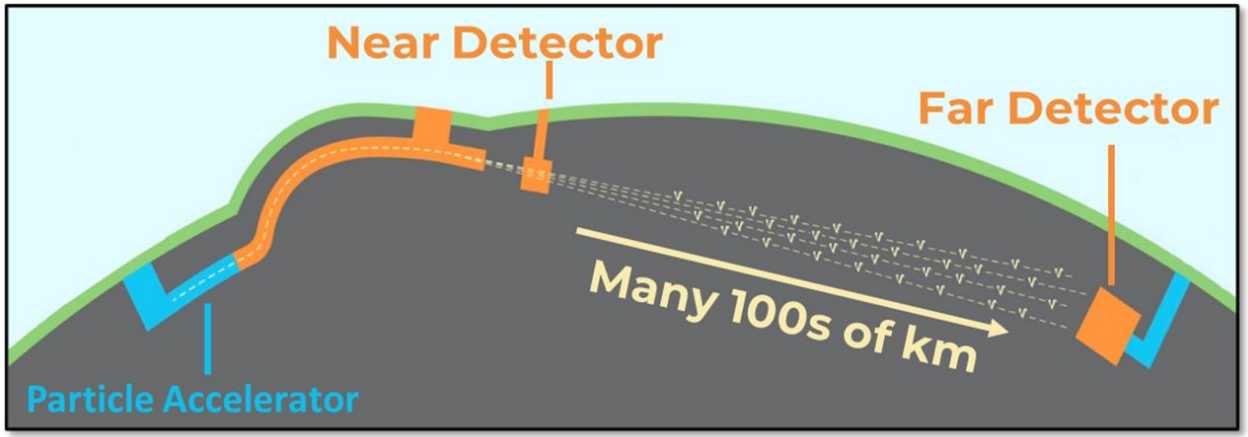
Increasing precision on the PMNS parameters from a range of experiments over the years

But mass ordering and δ_{CP} are still open questions



C. Gonzalez Garcia, Neutrino 2026

Upcoming Long-Baseline Oscillation Experiments



S. Dolan, Neutrino 2026

$$N_\ell(E_\nu) \propto P(\nu_\mu \to \nu_\ell)(E_\nu)\ \sigma(E_\nu)\ \Phi_\nu(E_\nu)\ \epsilon(E_\nu)$$

Far detector event rate

Oscillation Probability

Interaction Cross Section

Incoming neutrino flux

Detector Efficiency

The next-generation **long-baseline experiments** Hyper-K and DUNE plan to measure δ_{CP}

DUNE's long baseline will also let it measure the mass ordering

- JUNO will target this measurement by different means

Systematic Uncertainties

S. Dolan, Neutrino 2026

Uncertainty on Far Detector Event Rate N_L	T2K	NOvA	Hyper-K	DUNE
Stat. Uncertainty	6–10%	5–7%	1–2%	1–3%
Cross Sections	~4%	~3.5%	?	?
All Systematics	~5%	~3.5%	?	?

Systematic uncertainties in current long-baseline neutrino experiments are dominated by **interaction cross section uncertainties**, but they could hide under their statistical uncertainties

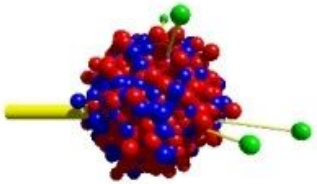
HyperK/DUNE will not be able to hide under statistics

Neutrino Event Generators

Neutrino event generators simulate the possible output products of any particular neutrino-nucleus interaction

- Underlie any simulation-based analysis

Several generators exist, taking different theoretical approaches

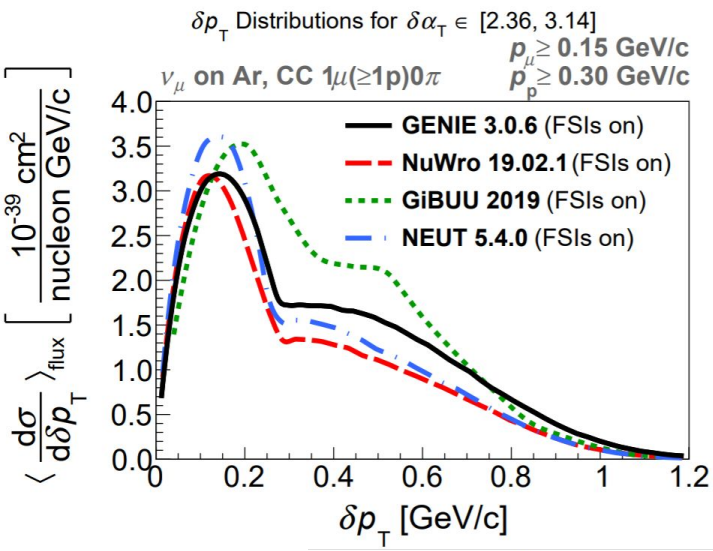


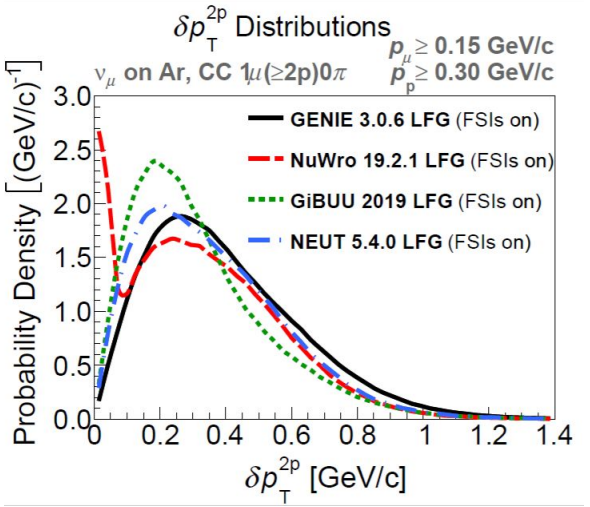
GiBUU
The Giessen Boltzmann-Uehling-Uhlenbeck Project

Neutrino Event Generators

Many approximations and assumptions are used in every existing generator, and they can be tuned for different purposes

Due to our still relatively poor understanding of neutrino-nucleus interactions, the **generators often disagree quite noticeably** with each other

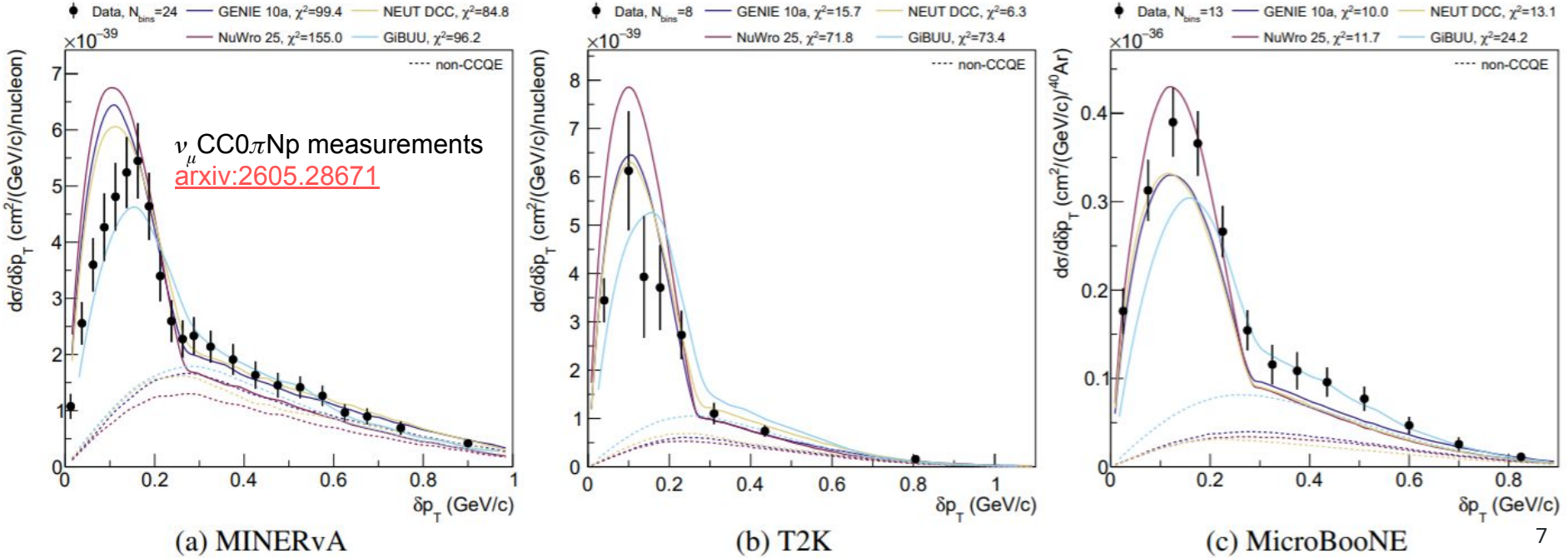




Plots: arXiv:2201.04664

Neutrino Event Generators

No individual generator is a global good fit to all existing neutrino measurements

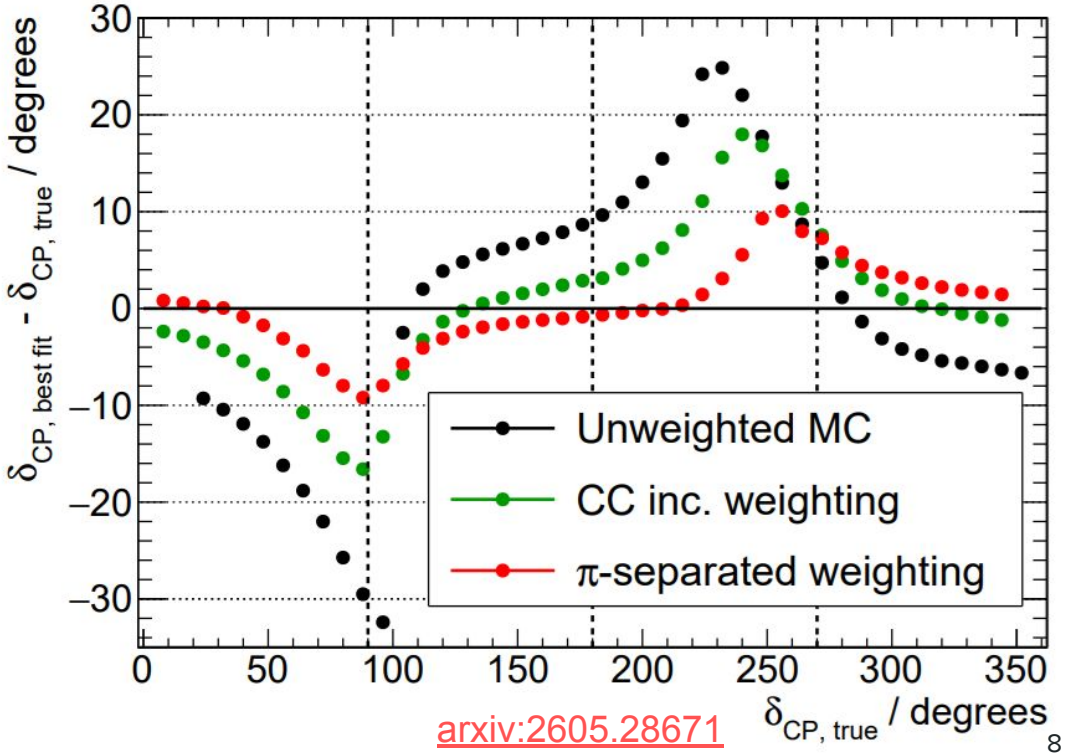


ν_μCC0πNp measurements
arxiv:2605.28671

Effects of Neutrino Cross Section Mismodeling on Oscillation Parameter Fits

Bias on δ_{CP} from a DUNE measurement fitting simulated data from one generator (NuWro) with another (GENIE)

- Improvements from “weighting” are constraints from a near detector fit



arxiv:2605.28671

Dealing with Generator Discrepancies

Long-term: theoretical effort to improve the interaction models

In the meantime, and to work towards that goal:

- Make detailed measurements in **multiple observables, with covariances** to better inform and constrain the models
- Ensure **analyses are not overly dependent on our choice of neutrino event generator**

The Likelihood Ratio “Trick”

A classifier trained to distinguish between two datasets **learns an approximation to their likelihood ratio**

Consider some loss **L** for a function **f** that classifies events from distributions parameterized by $\boldsymbol{\theta}_0$,$\boldsymbol{\theta}_1$ with **A(f(x))**, **B(f(x))** being rescalings that give the loss function

Minimizing the loss yields the likelihood ratio

$$L[f] = -\int \mathrm{d}x \Big(p(x \mid \theta_0) A(f(x)) + p(x \mid \theta_1) B(f(x)) \Big)$$

$$\begin{aligned} \frac{\delta L}{\delta f} &= -\frac{\partial}{\partial f}\Big(p(x \mid \theta_0) A(f(x)) + p(x \mid \theta_1) B(f(x)) \Big) \\ &= -\Big(p(x \mid \theta_0) A'(f(x)) \cdot f'(x) \\ &\qquad + p(x \mid \theta_0) B'(f(x)) \cdot f'(x) \Big) \\ &= 0 \iff -\frac{B'(f(x))}{A'(f(x))} = \frac{p(x \mid \theta_0)}{p(x \mid \theta_1)} = \mathcal{L}(x) \end{aligned}$$

J. High Energ. Phys. 2024, 136 (2024)

High-Dimensional Analysis

Likelihood ratio trick converts **multidimensional density estimation** (hard) into **classification** (easy$_{ish}$)

- ML-based classifiers are good at sorting through high-dimensional data

Dealing with detector effects in cross section measurements and studying effects of different neutrino event generators are both **problems of exploring high-dimensional distributions** that don't have analytic solutions, but where we can do forward simulation fairly well

Rest of this talk will be how we can use the likelihood ratio trick for:

1. **Unfolding detector effects**, in the case of making measurements
2. **Reweighting between neutrino event generators** to cover their differences, in the case of comparing them

Measurements and Unfolding

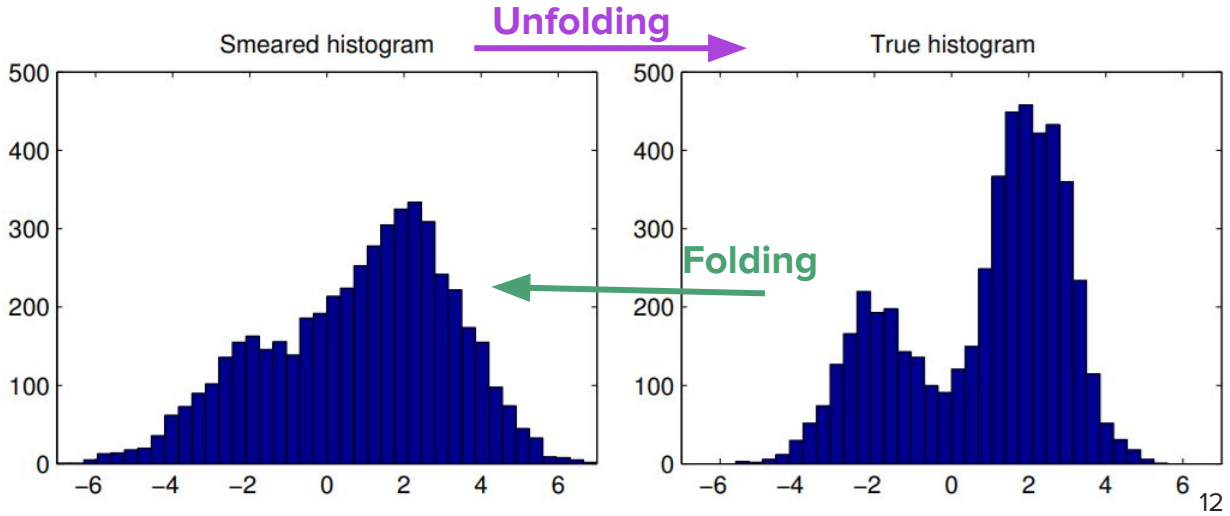
Measure **selected number** of **events** in a **reconstructed variable** with the detector.

Efficiency Correction → **Background Subtraction** → **Unfolding**

Want the **total number** of **signal events** in a **true variable** -- what physically happened.

Unfolding is the process of removing detector effects to recover the underlying "true" distributions

- But **very different true histograms can map to very similar smeared (reconstructed) distributions**



“Traditional” Unfolding

Several common unfolding techniques currently used for neutrino physics are:

- Iterative Bayesian Unfolding (aka D’Agostini)
- SVD Unfolding (including Wiener SVD)
- Template Likelihood Unfolding (recent T2K analyses)

These methods (generally) **require that the distributions are binned**, and work best with a **small set of variables** (around 1 to 4)

However many of the **corrections (e.g. efficiency) can have high-dimensional dependence**, and this is difficult to capture with only a few variables

In all cases the reconstructed MC distribution is reweighted to better match the data, and this is propagated to the truth MC distribution

OmniFold: ML-assisted Unfolding

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

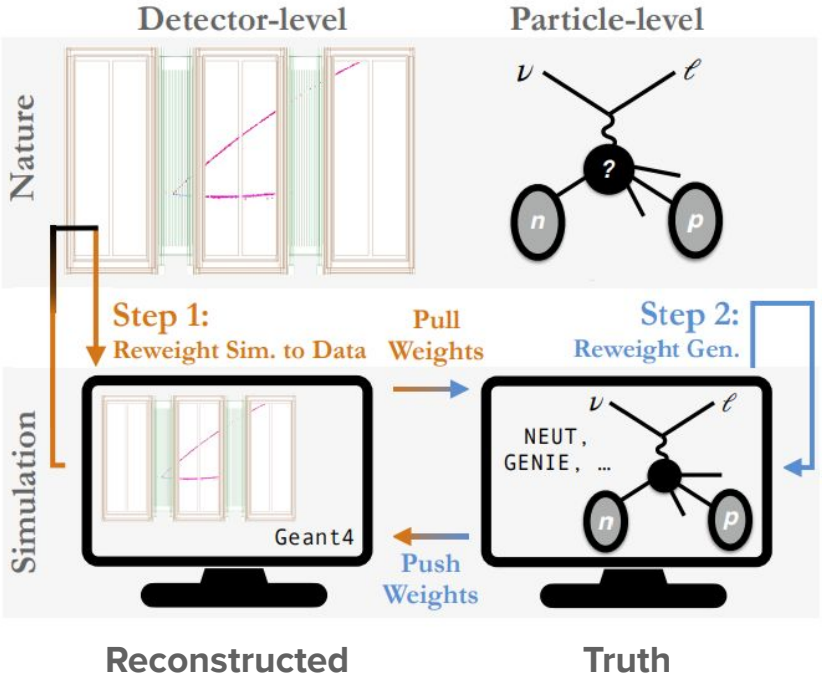
$$2.\ \nu_n(t) = \nu_{n-1}(t)\, L[(\omega_n^{\text{pull}}, \text{Gen.}), (\nu_{n-1}, \text{Gen.})](t).$$

OmniFold is an **iterative unfolding procedure** performed in two steps:

1. Reweight reconstructed MC distribution to (better) match data, yielding **pull weights $\omega_n(m)$** for each reconstructed **m**
2. Reweight nominal truth MC distribution to incorporate information from step 1, yielding **push weights $\nu_n(t)$** for each true value **t**

This is one iteration, and the method repeats until some convergence criteria is satisfied

Each step uses a separately trained classifier, and applies **per-event weights**



OmniFold details: PRL 124 (2020) 182001

ML-based Reweighting in OmniFold

For binned inputs, OmniFold is equivalent to IBU

Training ML-based classifiers and applying the likelihood ratio trick gives high-dimensional **event-by-event reweighting** of simulated events to fit the observed data

- If the ML classifier uses many variables in its reweighting, we effectively unfold in very high dimensional space
- Result: a set of reweighted generator events from which we can extract **unbinned unfolded results of any observable**

Efficiency corrections are also unbinned, from the reweighting's interpolation

Background estimates naturally included in reweighting as well

OmniFold Applications

OmniFold has been used for several collider-based measurements already*

OmniFold is useful anywhere there is **a lot of (reliable) information that is relevant** to the unfolding problem, or there are **many observables of interest**.

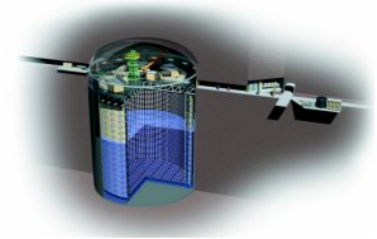
In neutrino physics, it offers:

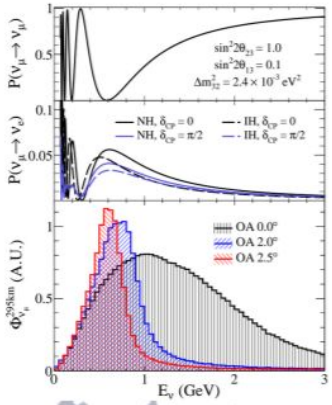
- Improving the smearing/efficiency corrections by being unbinned and high-dimensional
- Simultaneous unfolding of several correlated samples
- Probing samples where many observables are useful proxies for underlying physics, and we'd like to project our results into many directions at once

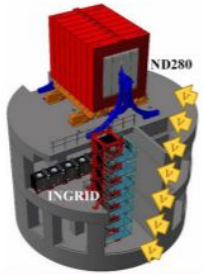
*Summarized in Eur. Phys. J. C 86, 106 (2026)

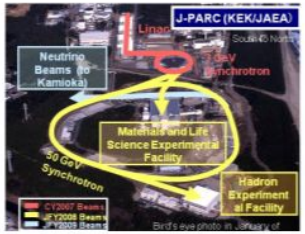
The T2K Experiment

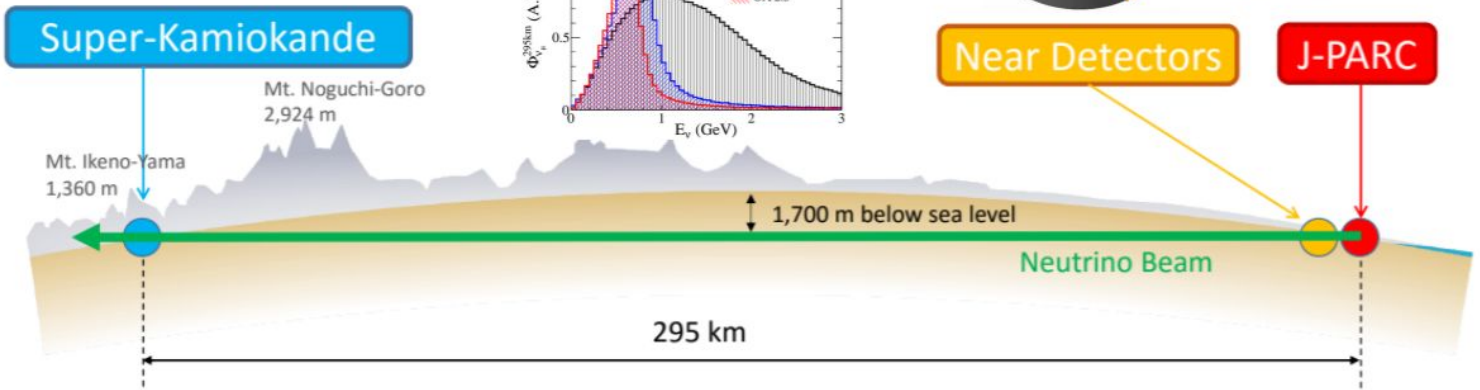
T2K is a long-baseline neutrino oscillation experiment in Japan that has been accumulating data since 2010



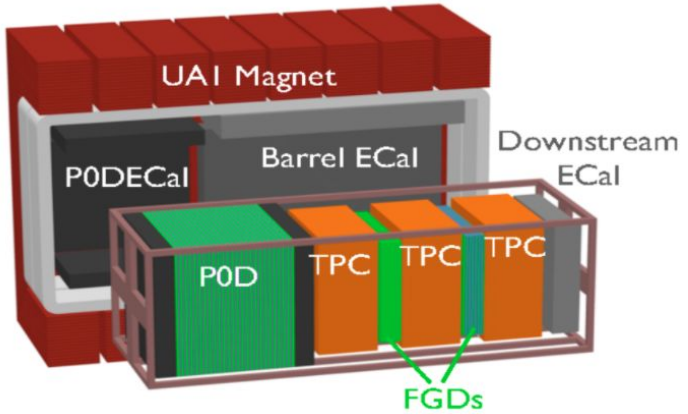








Test Setup - ND280 Public Dataset

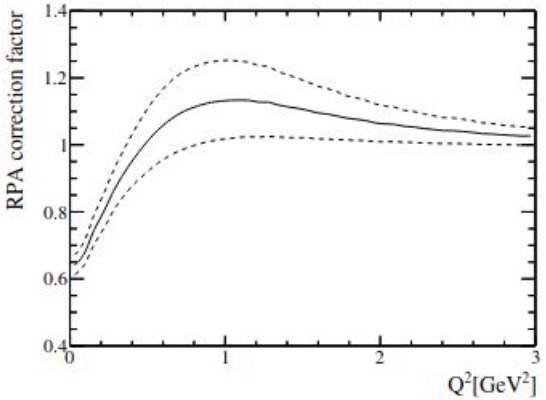


Public dataset of **~1.2 million simulated ND280 events** intended for ν_μ CC0π 2D differential cross-section

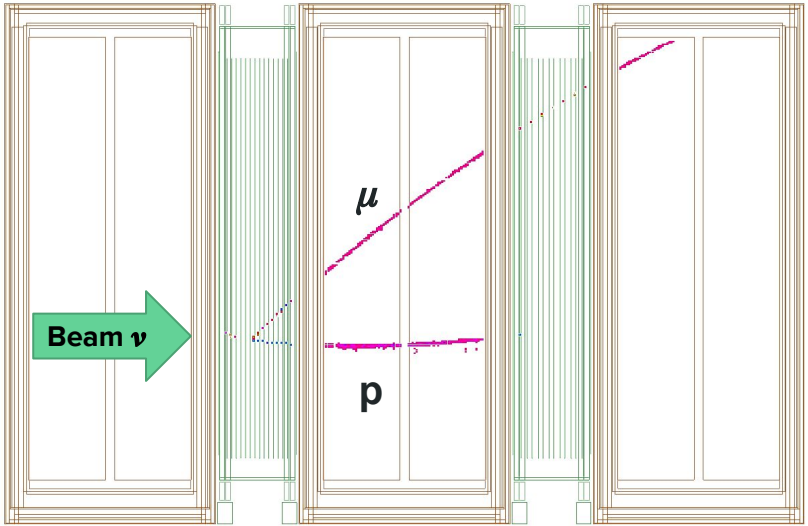
- Corresponds to about **20k measured events**

Dataset includes **muon and leading proton info** per event

Create **fake data** with a reweighting as a function of Q^2



- Extract CC0π differential xsecs like with real data and compare against the data-truth values after unfolding to evaluate



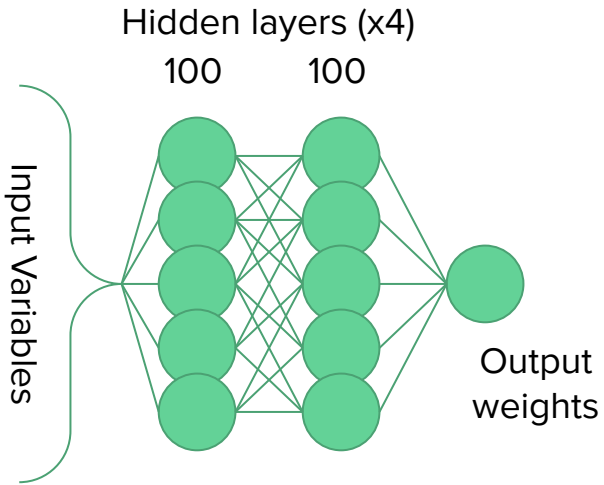
OmniFold Inputs & Network

OmniFold is agnostic to the choice of classifier

These results use a densely connected neural network with 4 hidden layers of 100 nodes each

Input variables:

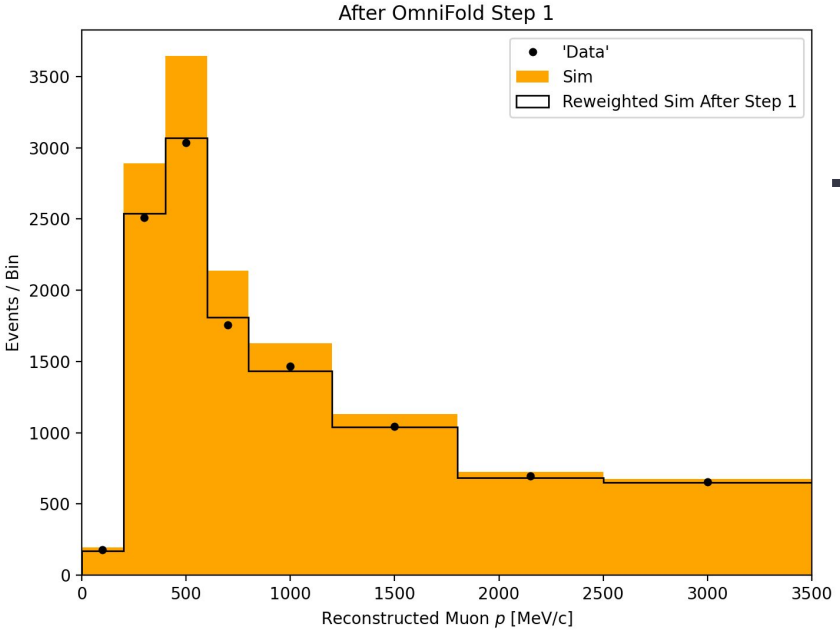
- Various **kinematic observables** (details on following slides). Generally standardized to mean 0 and unit variance, with values of 0 when the variable does not exist
- For detector space only: **Detector sample ID** (1-hot encoded, out of 8 possible sample IDs)
- For truth space only: **Interaction topology** (1-hot encoded, out of 5 possibilities: CC0π0p, CC0π1p, CC0πNp, CC1π, CCOther)



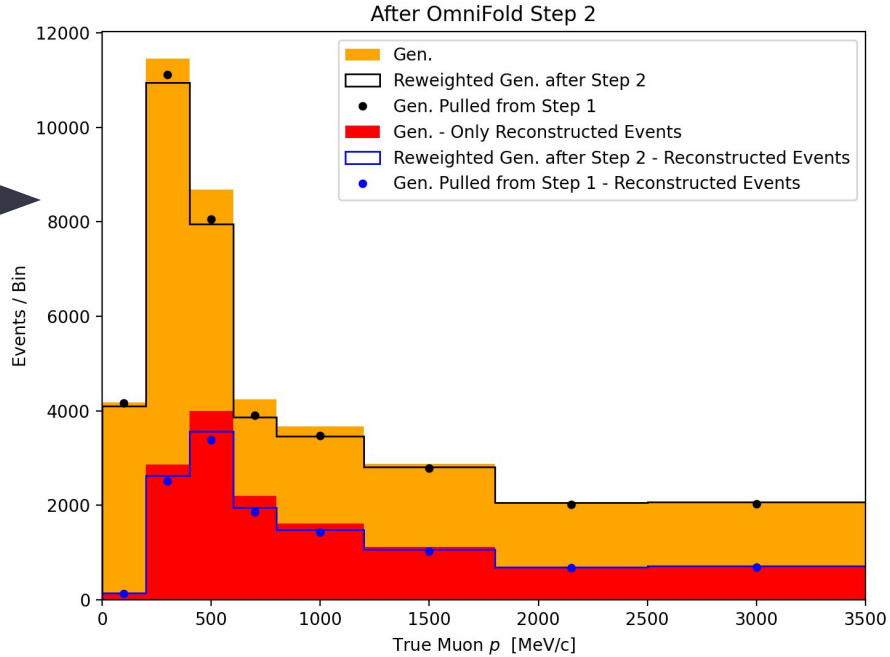
Using one NVIDIA A100 on a NERSC Perlmutter node, << 1 minute to train one network

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!

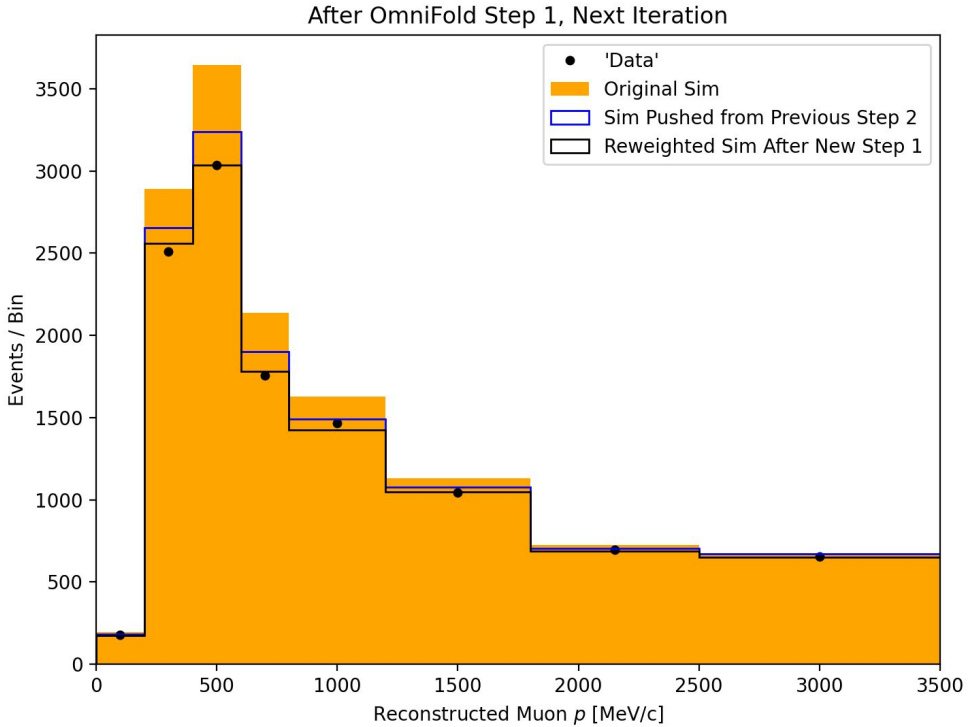
OmniFold Procedure Example

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

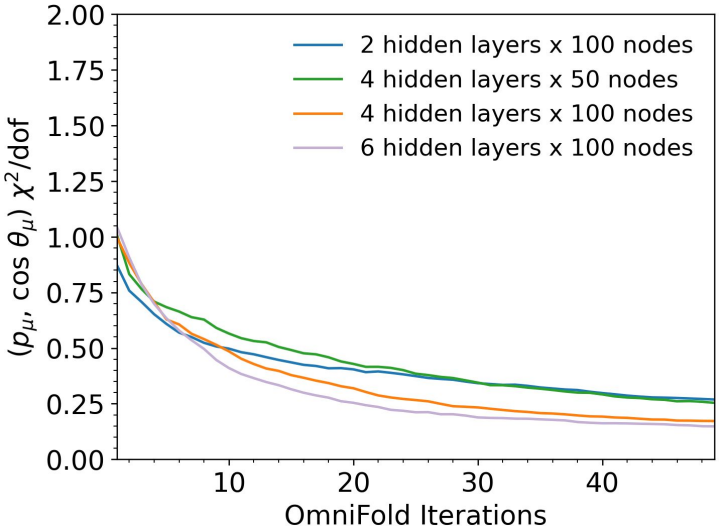


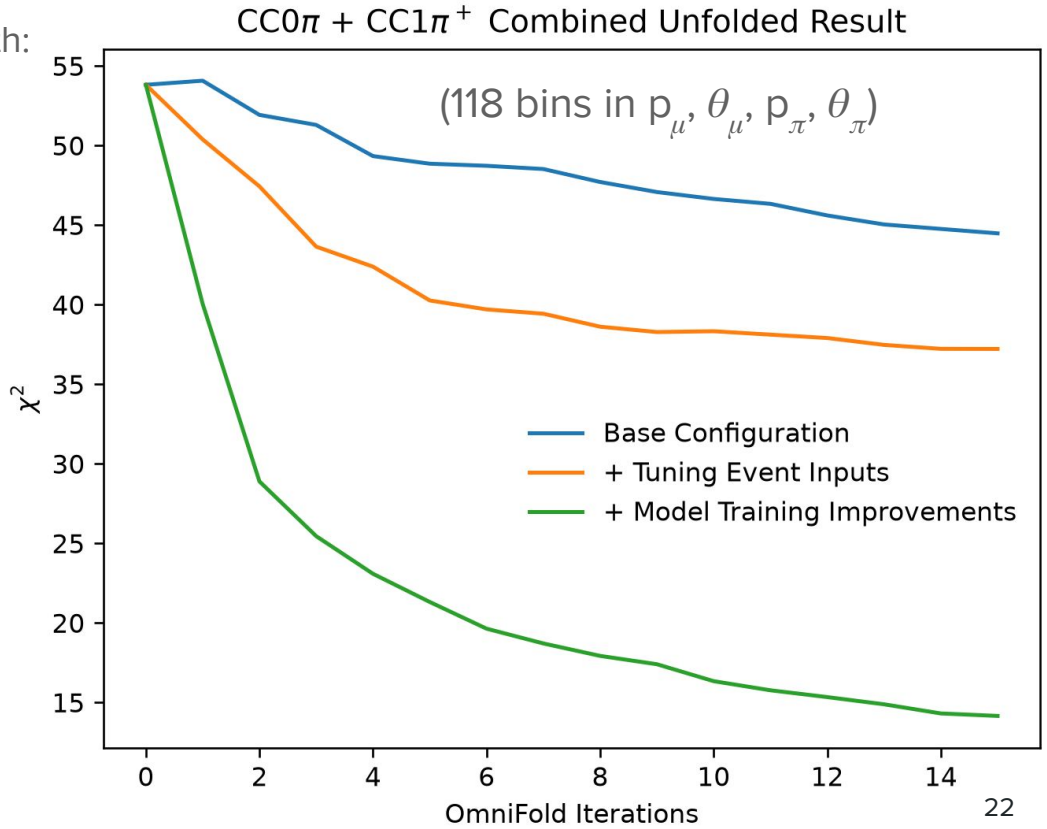
Tuning OmniFold

Unfolding performance can vary significantly with:

- Choice of inputs
- Network structure and training hyperparameters

Must be tuned before final result!

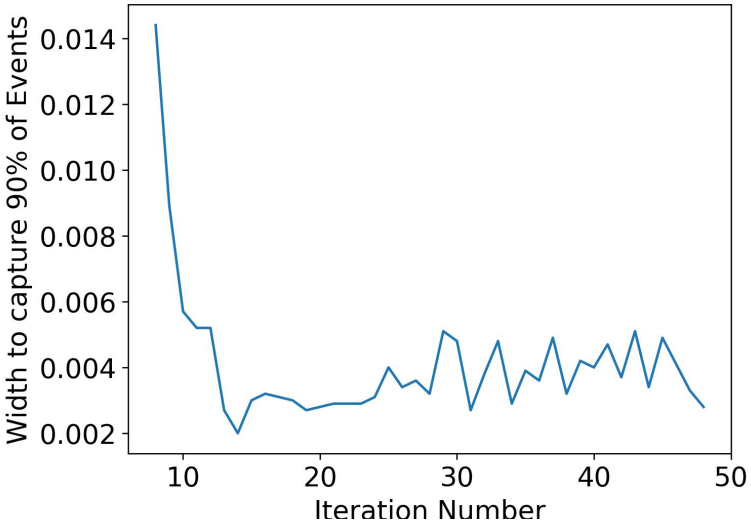


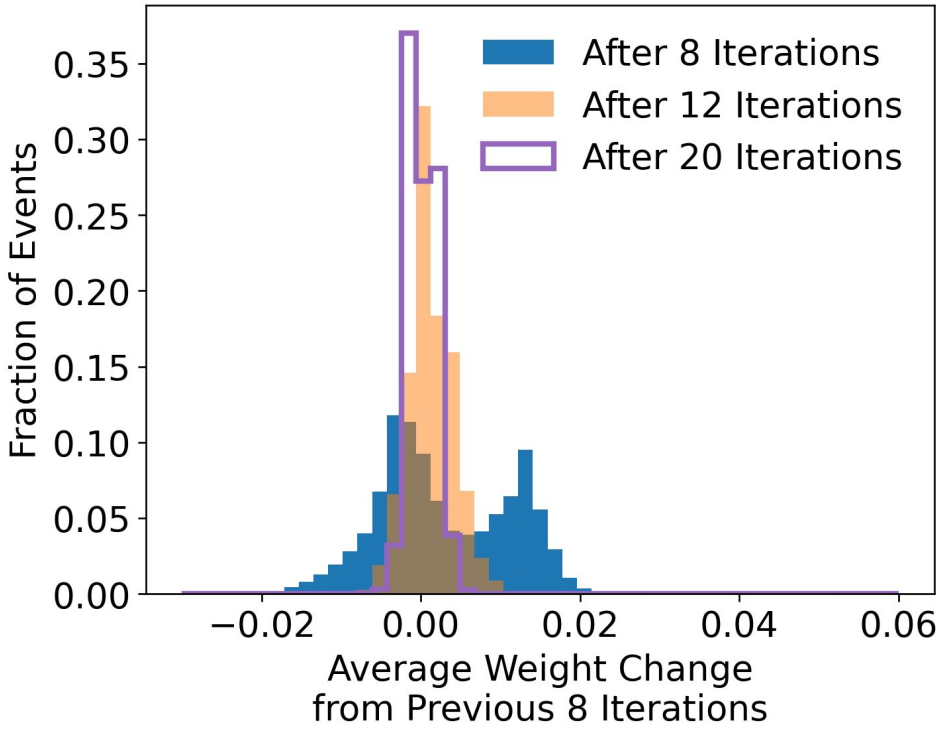


Unbinned Convergence Metrics

A convergence metric for OmniFold is ideally independent of binning and observable choices

Plot average **weight change of each event** over the last N iterations, which should cluster around 0 as it converges



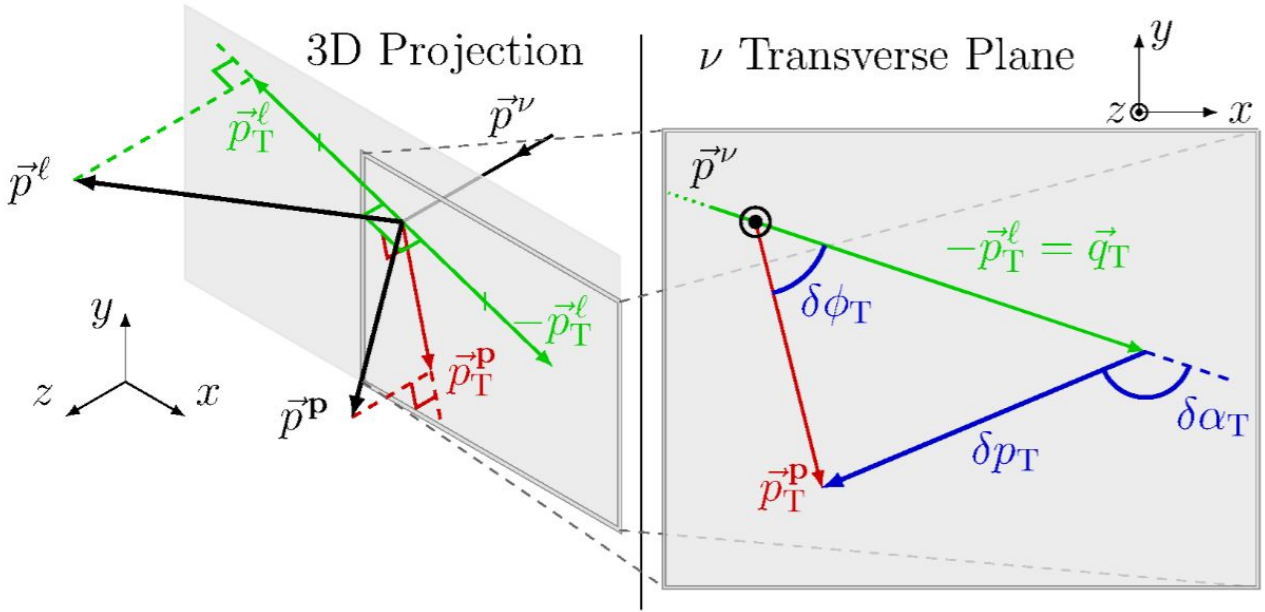


Variables of Interest

Evaluate performance of unfolding methods with 4 variables of interest:

- (p_μ, $\cos\theta_\mu$): muon momentum and forward angle
- $$\delta\vec{p}_{\mathrm{T}} \equiv \vec{p}^{\,\mathrm{T}}_{\ell'} + \vec{p}^{\,\mathrm{T}}_{\mathrm{h}'}$$
  $$\delta\phi_{\mathrm{T}} \equiv \arccos\left(\frac{-\vec{p}^{\,\mathrm{T}}_{\ell'}\cdot\delta\vec{p}_{\mathrm{T}}}{p^{\mathrm{T}}_{\ell'}\delta p_{\mathrm{T}}}\right)$$
  $$\delta\alpha_{\mathrm{T}} \equiv \arccos\left(\frac{-\vec{p}^{\,\mathrm{T}}_{\ell'}\cdot\vec{p}^{\,\mathrm{T}}_{\mathrm{h}'}}{p^{\mathrm{T}}_{\ell'}p^{\mathrm{T}}_{\mathrm{h}'}}\right)$$

These last 3 are the single transverse variables (STVs)



Testing OmniFold

Setup	Kinematic Inputs	Description
IBU-UniFold	Binned, 1-hot-encoded	Equivalent to IBU
Binned UniFold	Binned, using kinematic centers	Equivalent to IBU, but structured poorly for a NN
UniFold	Unbinned, single observable	Still unfolding each observable separately, but unbinned
MultiFold	$(p_\mu, \cos\theta_\mu, p_p, \delta p_T, \delta\alpha_T, \delta\phi_T)$	All observables of interest provided as direct input
OmniFold	$(p_\mu, \cos\theta_\mu, \phi_\mu, p_p, \cos\theta_p, \phi_p)$	Muon/proton kinematics as input, from which observables of interest can be derived

Each method is run **500 times**, to get syst/stat uncertainties with a toy-universe approach

Results

- UniFold is worse than IBU-UniFold
  - Little gain from going unbinned without additional info (with good bin choices)
  - Evidence of imperfections in NN training
- For χ^2, MultiFold is similar to IBU and better than OmniFold
- For bias (triangular discriminator), MultiFold/OmniFold are both better than IBU

	χ^2			
Method	$(p_\mu, \cos\theta_\mu)$ DoF=58	δp_T DoF=8	$\delta \alpha_T$ DoF=8	$\delta \phi_T$ DoF=8
Prior	298.2	2.3	5.9	4.9
IBU-UniFold	2.1	0.2	0.4	0.1
Binned UniFold	21.4	1.4	0.9	0.5
UniFold	27.1	1.1	0.6	1.1
MultiFold	3.1	0.3	0.2	0.3
OmniFold	10.0	0.8	1.1	0.4

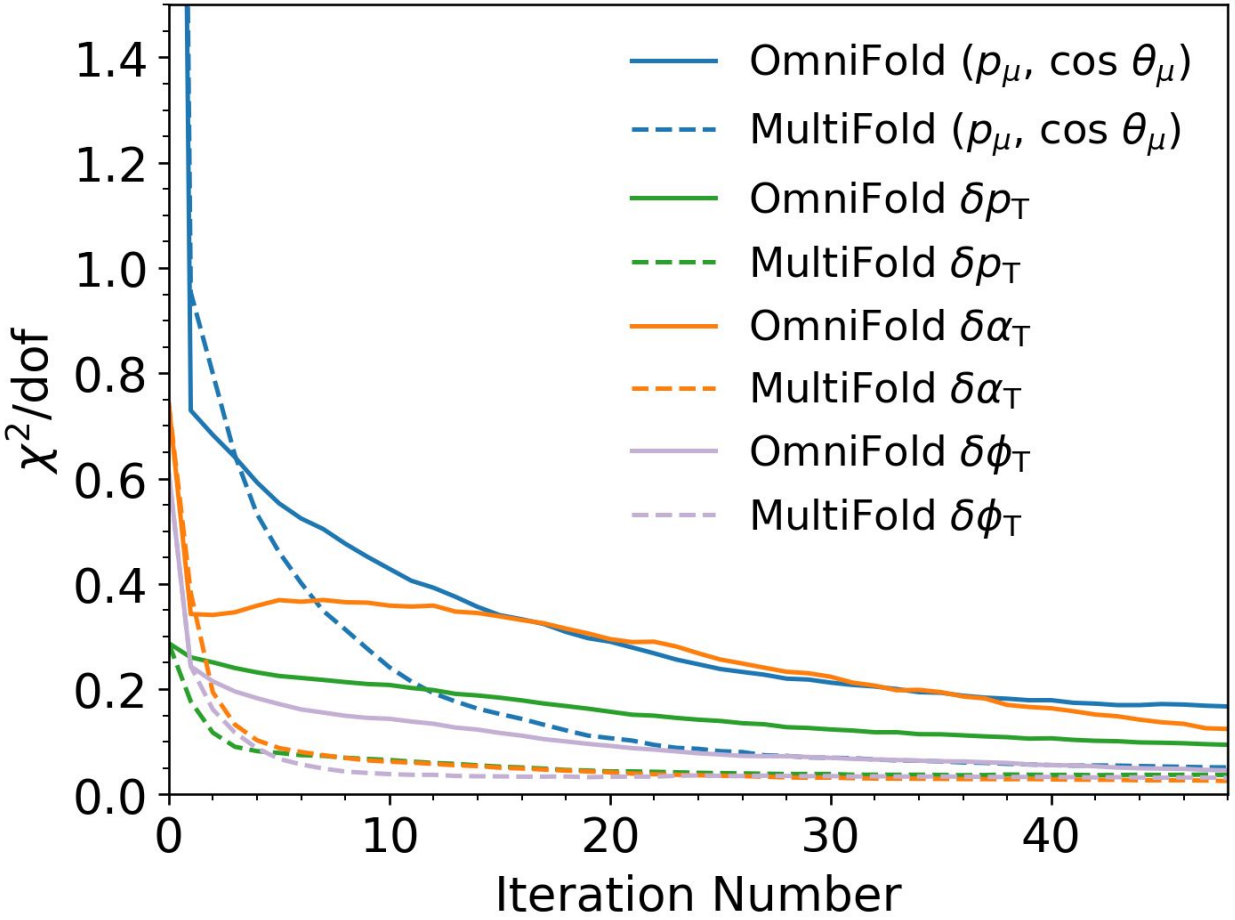
	Triangular Discriminator			
Method	$(p_\mu, \cos\theta_\mu)$	δp_T	$\delta \alpha_T$	$\delta \phi_T$
Prior	545.6	27.5	31.2	26.7
IBU-UniFold	17.1	1.9	3.4	0.8
Binned UniFold	29.9	2.8	6.0	1.9
UniFold	17.3	5.7	5.8	1.7
MultiFold	2.7	0.7	0.6	0.6
OmniFold	9.4	1.7	3.0	2.1

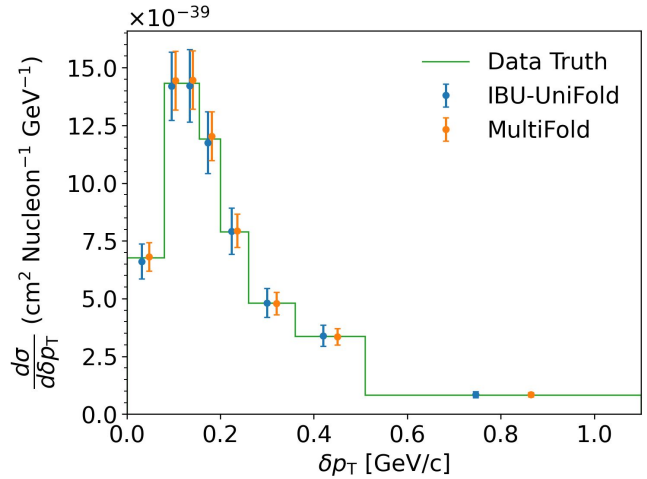
Input Comparison

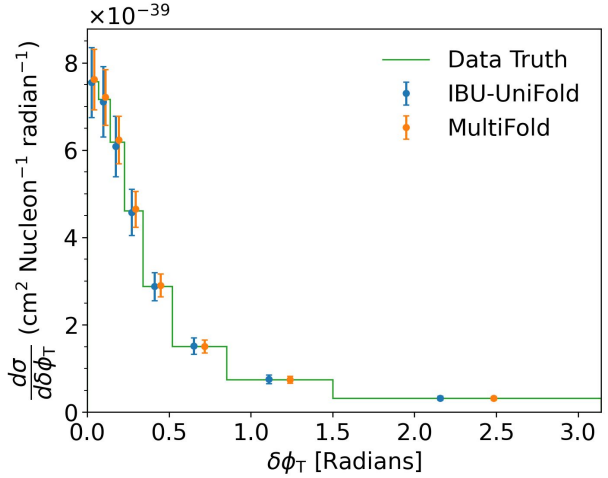
MultiFold (using unfolding variables as direct input) **outperforms OmniFold** (general muon/proton kinematics input)

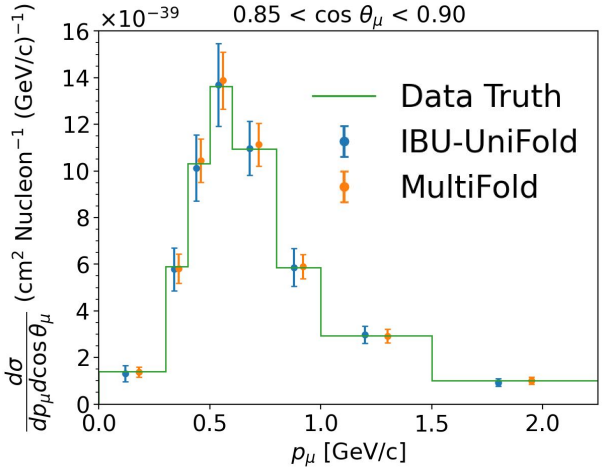
Again indicates imperfect neural network training

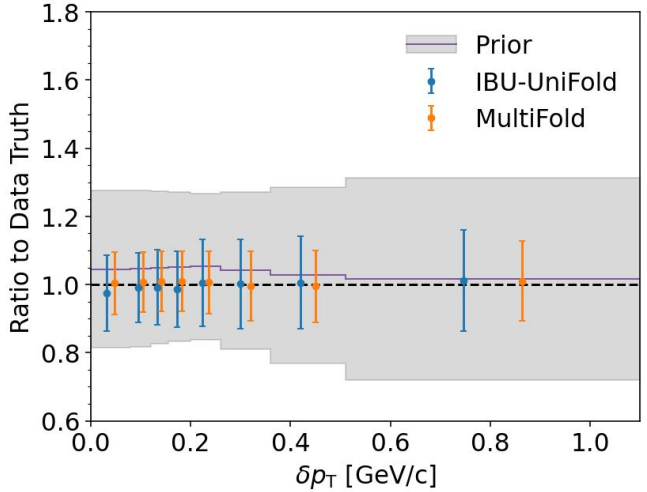
- Our statistics are low: 20k data events

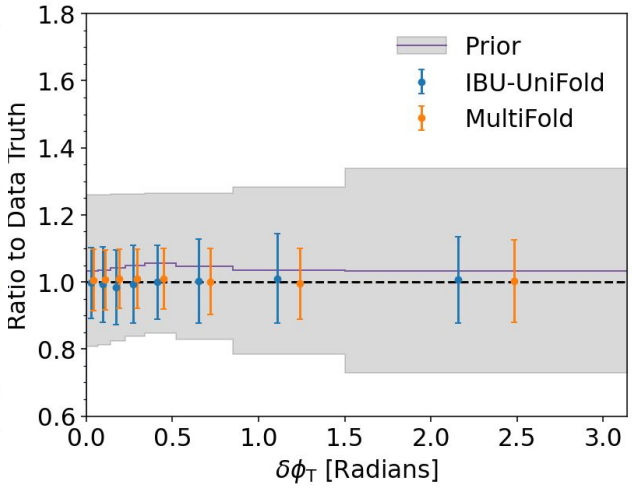


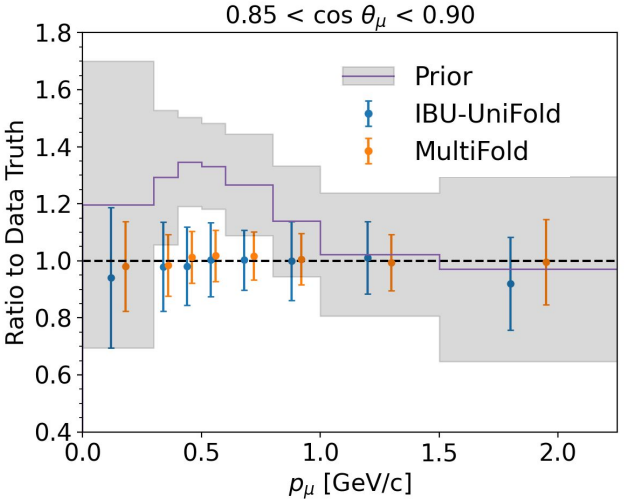

×10⁻³⁹
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{d\delta p_T}$ (cm² Nucleon⁻¹ GeV⁻¹)
δp_T [GeV/c]


×10⁻³⁹
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{d\delta\phi_T}$ (cm² Nucleon⁻¹ radian⁻¹)
$\delta\phi_T$ [Radians]


×10⁻³⁹
0.85 < cos θ_μ < 0.90
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{dp_\mu d\cos\theta_\mu}$ (cm² Nucleon⁻¹ (GeV/c)⁻¹)
p_μ [GeV/c]


Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
δp_T [GeV/c]


Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
$\delta\phi_T$ [Radians]


0.85 < cos θ_μ < 0.90
Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
p_μ [GeV/c]

OmniFold Results with T2K: Summary

More details on this T2K study with OmniFold: Phys. Rev. D 112, 012008 (2025)

- **OmniFold approach obtains similar χ^2 and less bias** than IBU, but we get to unfold everything at once and in an unbinned way
  - OmniFold is able to make use of additional information!
- **Low statistics** limits the performance, since the classifiers must be trained against actual data
  - Data augmentations may help
  - New neutrino datasets may not be so statistics-limited anymore

Now looking towards more measurements with T2K, MINERvA, SBND, ProtoDUNE...

(Back to) Neutrino Event Generators

What about when we want to check the dependence of the measurement we're attempting on the input model we use for our simulation?

Detector simulation is computationally very expensive

- Cannot create a full statistics detector-reco-level MC dataset for every generator
- Most generators provide ways to reweight their results so that modeling uncertainties can be studied without re-running detector simulation

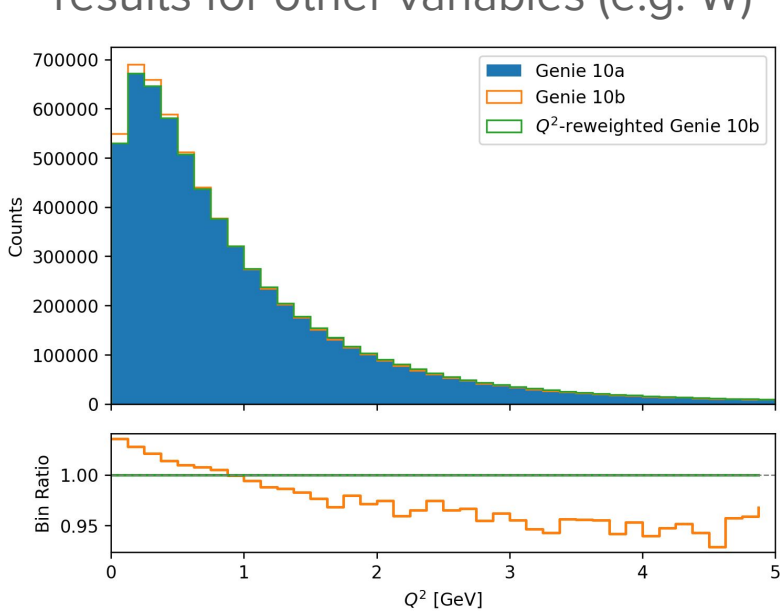
Running just the generators is computationally cheap

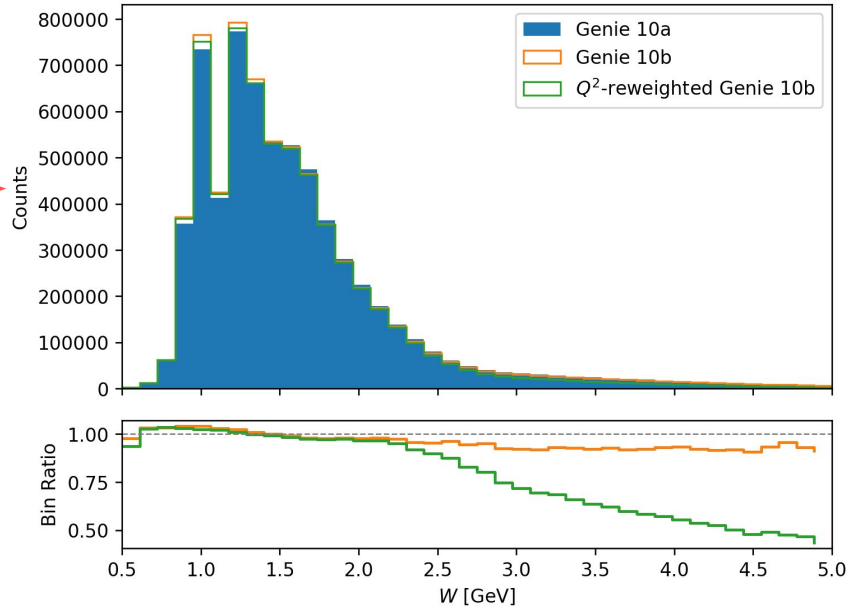
- We can generate lots of samples to compare in truth space

Bin-based Reweighting

Traditional reweighting methods were bin-by-bin or with splines

But binned reweighting in one variable (e.g. Q^2) does not generally give valid results for other variables (e.g. W)





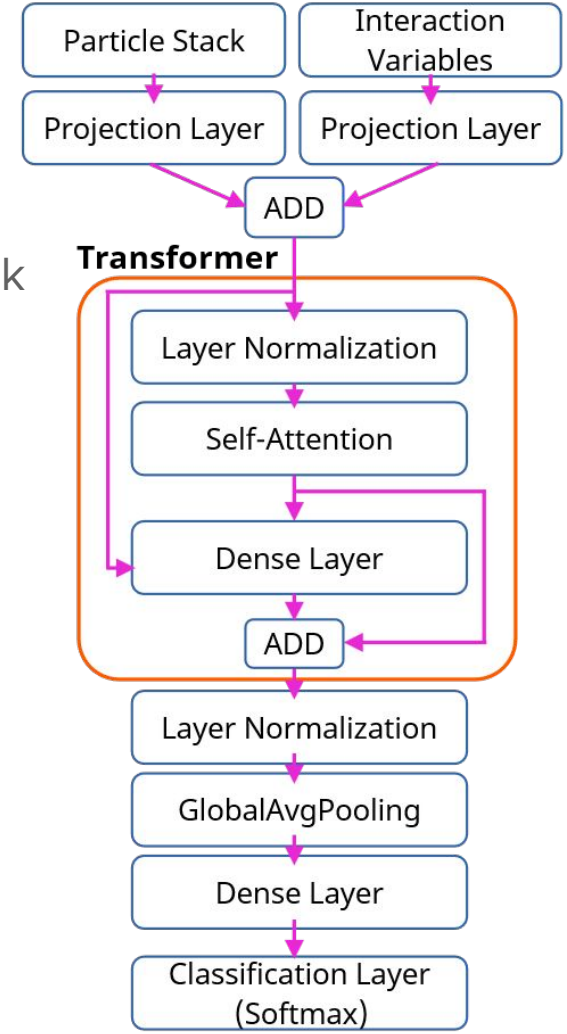
Transformer-based Reweighting

Training a classifier and applying the likelihood ratio trick again gives a way to reweight generator outputs as a function of many of their outputs simultaneously!

Ideal reweighting would be based on the **full particle stack** output of the generators

- All observables of interest are derived from this particle stack
- A set of generator events can be **run through detector simulation once** and then **reweighted based on the generator-level outputs**

Transformer architecture can learn the inter-particle relations and handle variable-length particle stacks.



Training Details

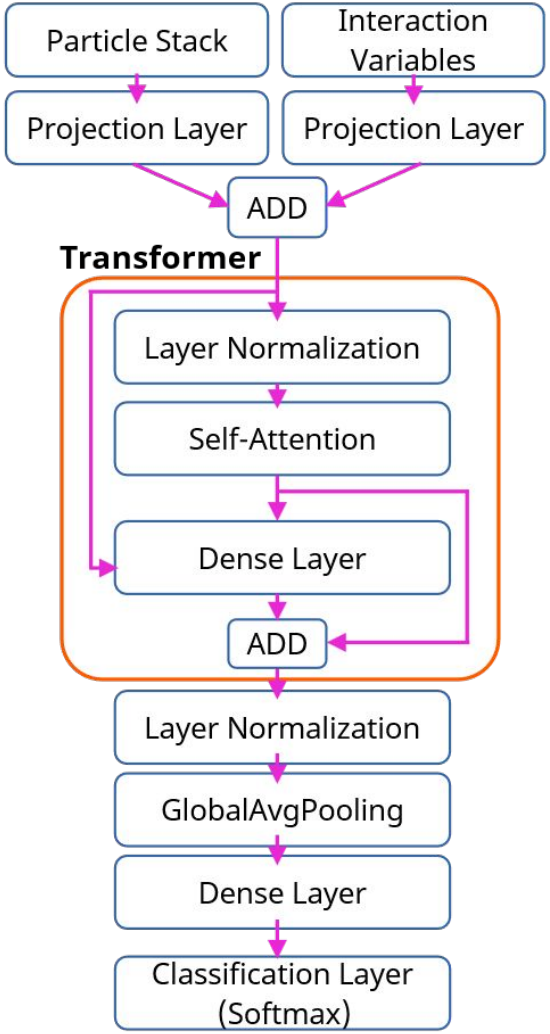
Training data: ~50M Genie10a ν_μ CC events and ~50M Genie10b ν_μ CC events, generated with flat neutrino flux

Model inputs:

- Up to 10 particles (muons, protons, neutrons, pions) per event, with 4-momentum and PID for each
- Event-level variables q_0, q_3, Q^2, W, E_ν
- ~1M model parameters

Training time: ~4 hours on 6x4 A100 GPUs

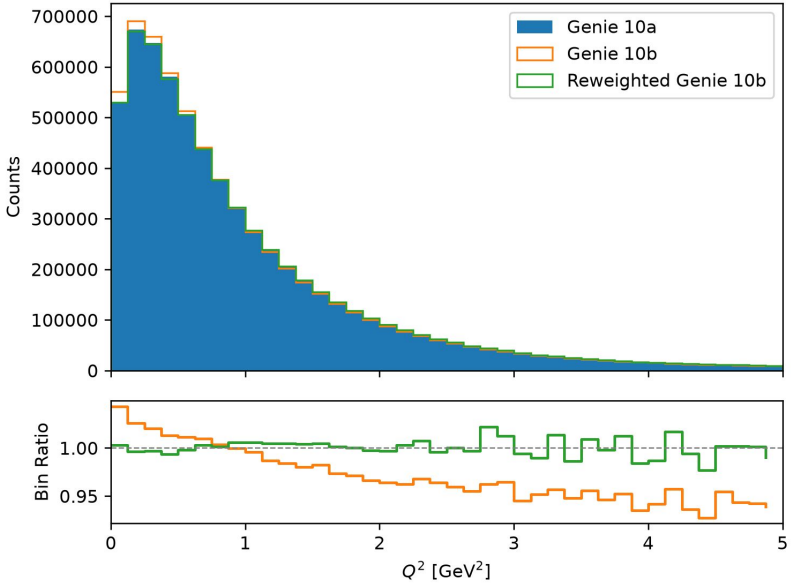
Test data: ~10M Genie10a and Genie10b ν_μ CC events generated with DUNE near detector neutrino flux

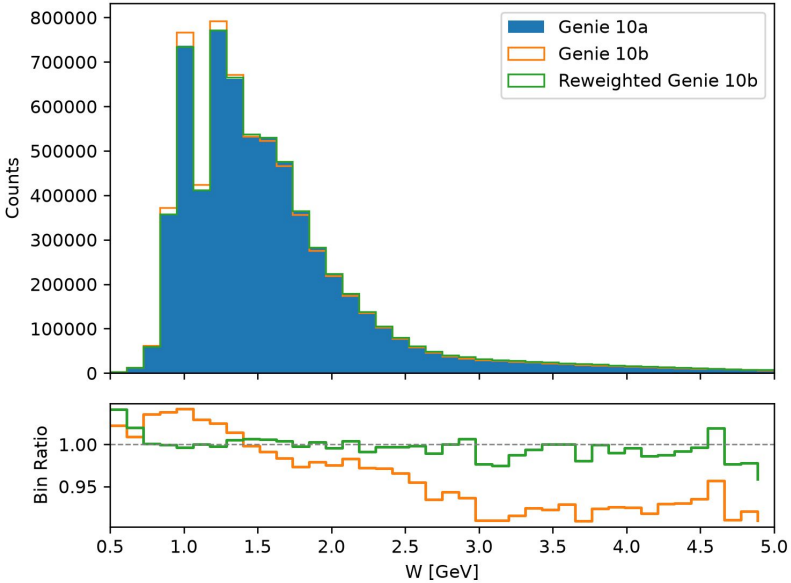


1D Results

Test setup: Reweighting Genie 10b ν_μ CC events to match Genie 10a

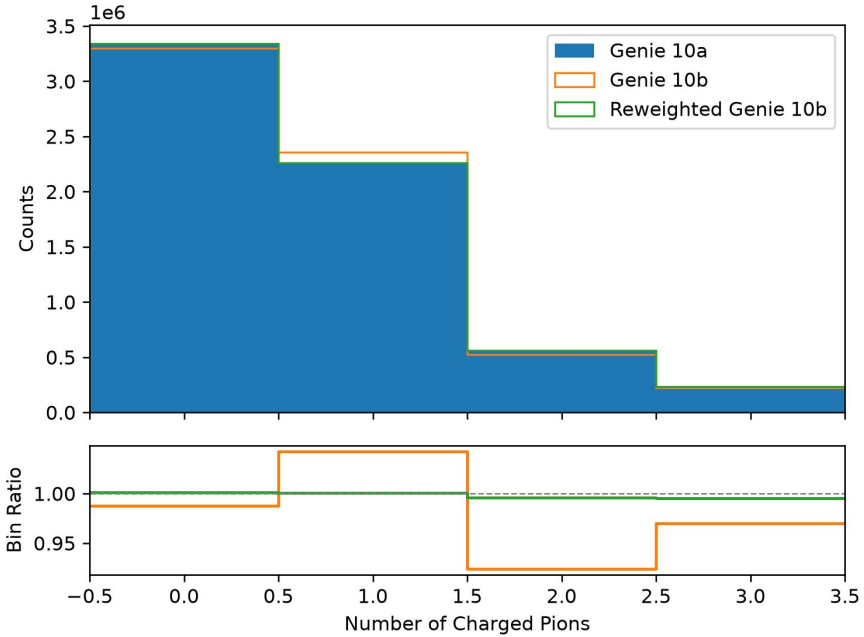
Projecting the results into a few 1D comparisons shows the reweighted Genie 10b outputs look reasonably similar to the Genie 10a outputs

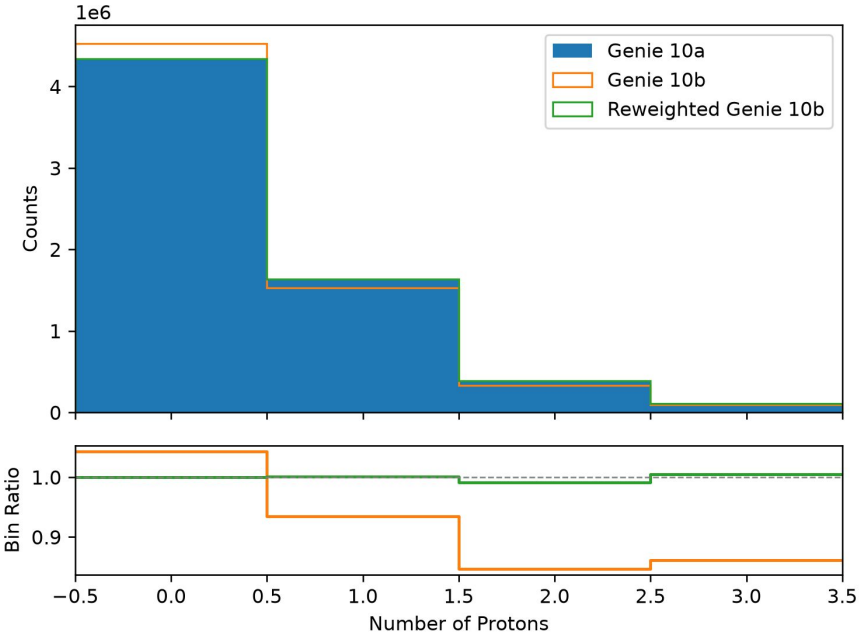




1D Results

The number of $\pi^{\pm}$ and protons over some detector threshold (p>100 MeV for pions, p>50 MeV for protons) is not an explicit input, but is learned quite well too





Metrics - Effective Sample Size

What result is **good enough**? Ideally checked with an actual analysis, but what if we want to evaluate our general usefulness?

In the limit of infinite statistics, we expect to always be able to distinguish between the reweighted distribution and the target distribution

Real analyses do not have infinite statistics in their MC, so what we actually care about is an **effective sample size**

- A level of statistics where the reweighted distribution and target distribution are identical within statistical uncertainties

Many standard tests exist for this on binned distributions, but we want an **unbinned comparison**, since our method is performing unbinned reweighting

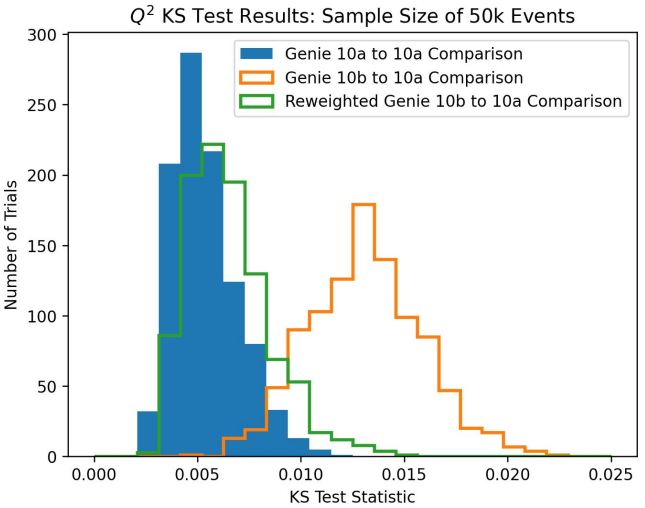
1D Metrics - KS Test

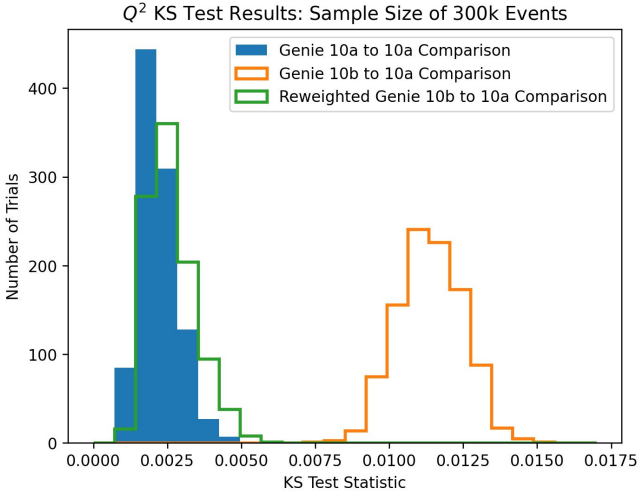
KS test provides a standard unbinned 1D comparison metric

For any given sample size, we can bootstrap resample our events to build a distribution of KS test results given the available statistics

Larger test statistics gives greater separating power as expected

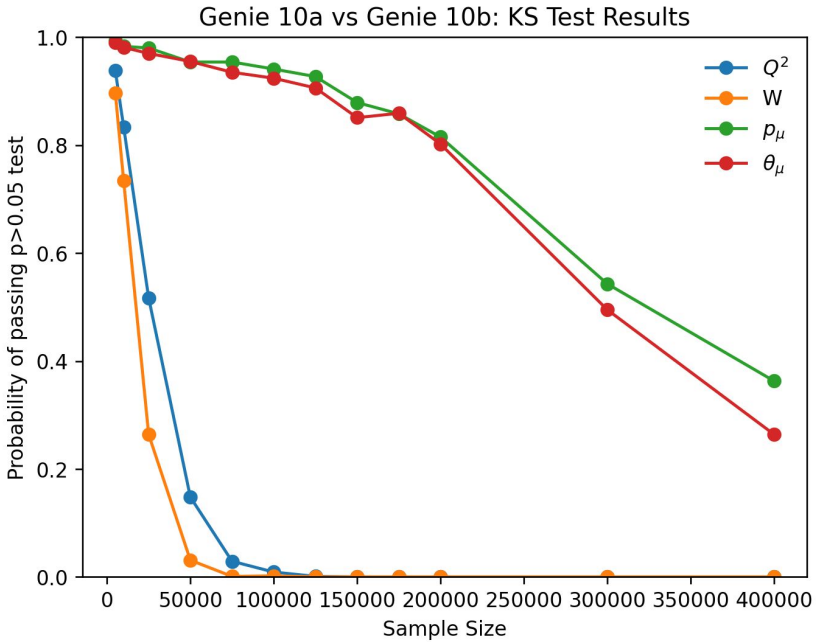
When **comparing our reweighted result against the target distribution** (Genie 10a), what percentage of KS test results are within the 95th percentile of KS test results you get from a **Genie 10a self-comparison**?



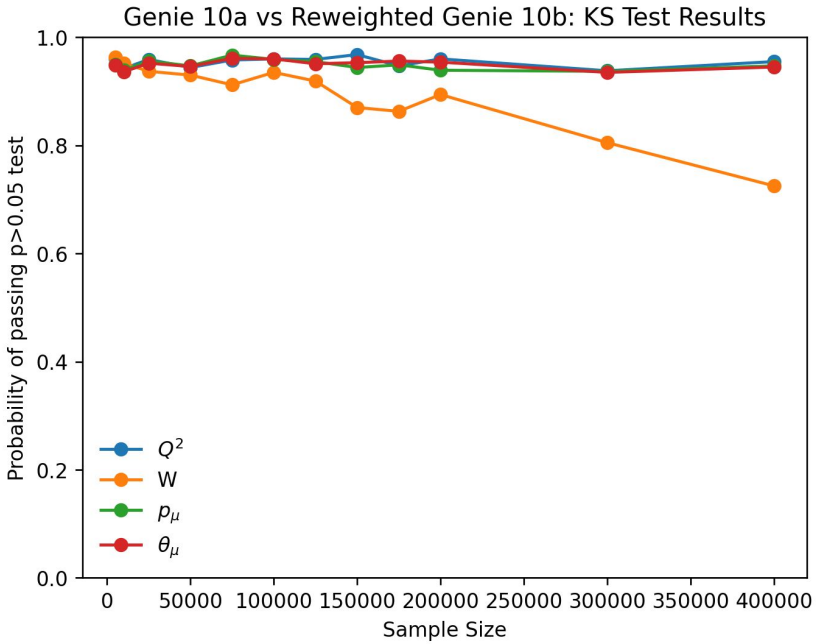


1D Metrics - KS Test Results

Without reweighting: sanity check that the KS test distributions are very distinguishable at small sample sizes



Transformer-based reweighting: reasonable up to fairly large sample sizes



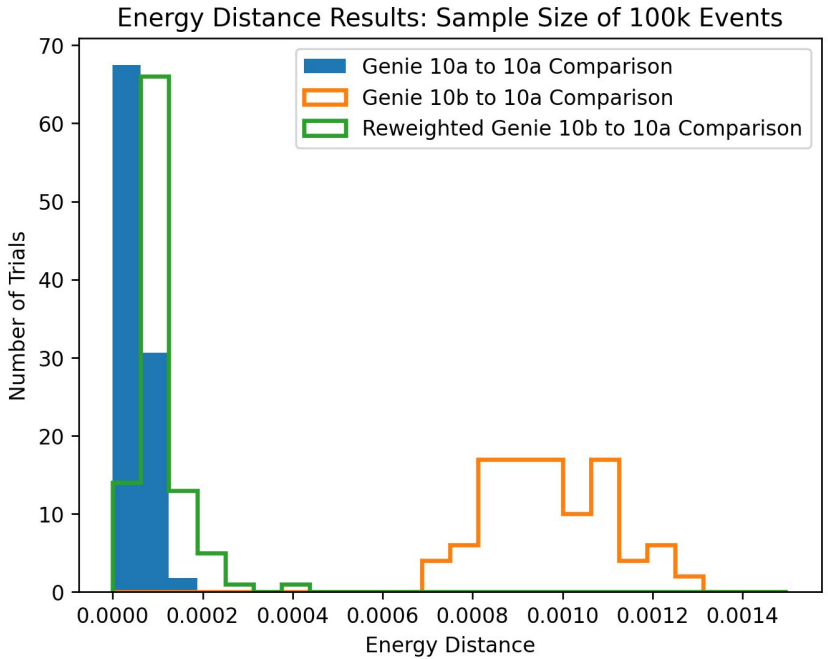
Multidimensional Metrics - Energy Distance

$$D^2(F,G) = 2\,\mathrm{E}\,\|X-Y\| - \mathrm{E}\,\|X-X'\| - \mathrm{E}\,\|Y-Y'\|$$

Ideally want an **unbinned, multidimensional test** to evaluate our reweighted result

Energy distance provides this, for some choice of observables:

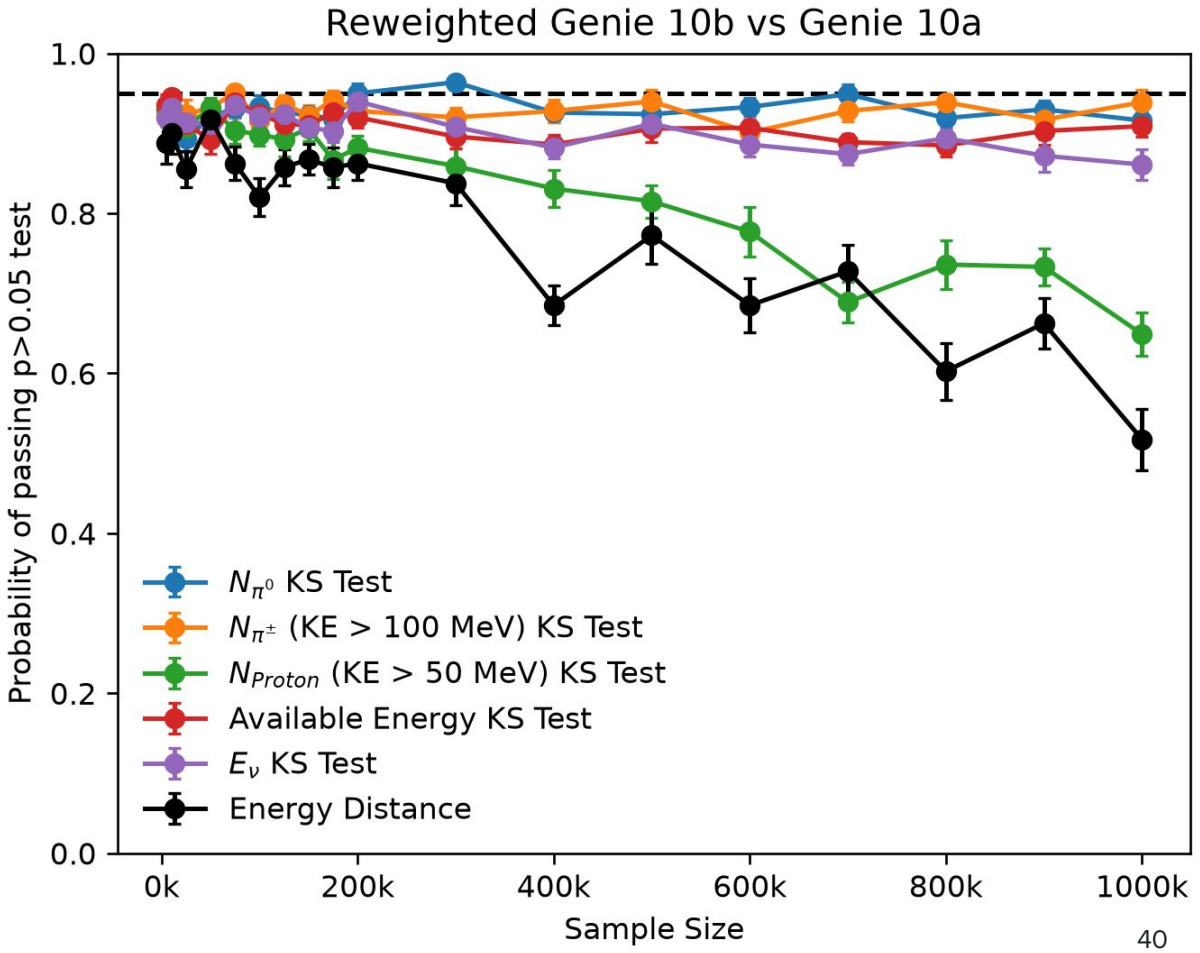
- E_ν, $E_{available}$, Number of π^0, Number of $\pi^\pm$ with KE > 100 MeV, Number of protons with KE > 50 MeV



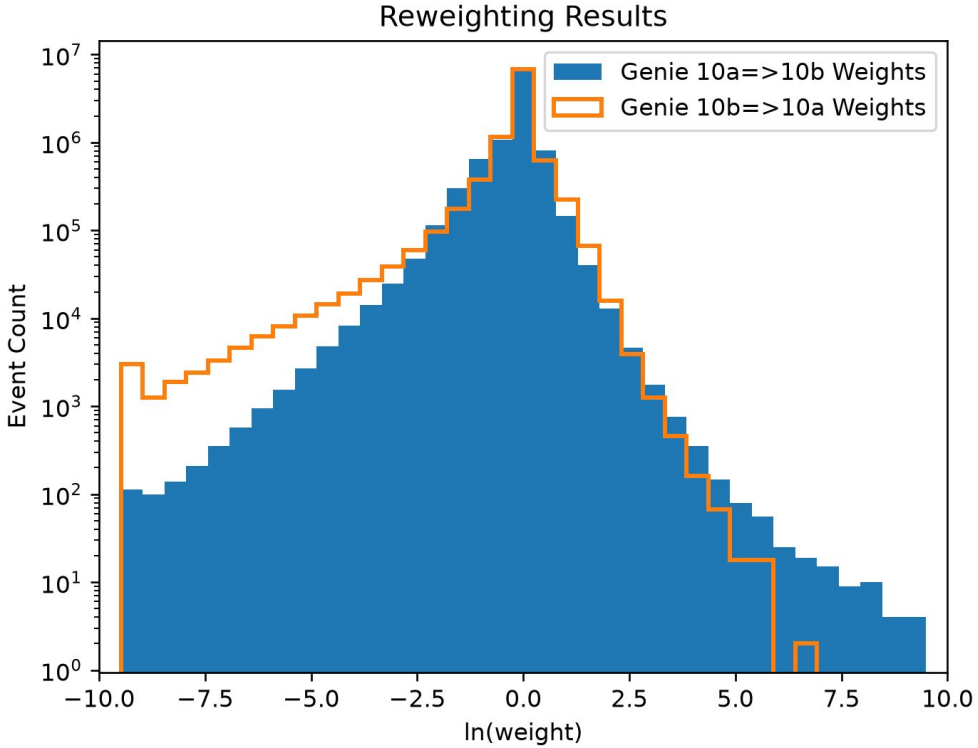
Combined Metrics

Energy distance with all observables combined **shows more discriminatory power** than the individual KS tests

The cutoff for “good enough” will still be arbitrary, but this provides a fair comparison between different reweighting results

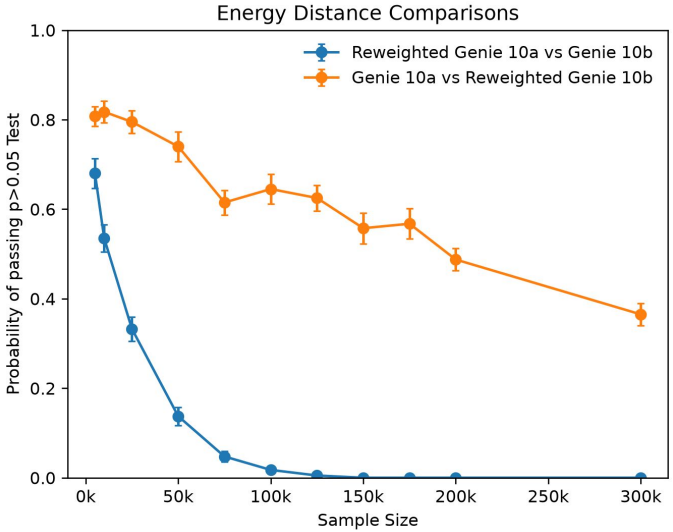


Notes on General Reweighting



Reweighting is problematic when models have disparate phase spaces

- **Weights near 0:** source model covers a phase space that's not in the target model
- If target model includes phase space not in the source model, **these will never be covered**



A Surrogate Neutrino Event Generator?

What if we could create a surrogate neutrino event generator whose events can be reweighted to replicate the output of any existing generator?

$$w(\vec{x}) = \frac{G_{\mathrm{B}}(\vec{x})}{G_{\mathrm{A}}(\vec{x})}$$

This can be 0

This cannot be 0
(for any model of interest)

This model would not be physical on its own, but anyone could pass it through their detector simulation once and then apply simple weights to use it like any other event generator

Currently planning an **open data challenge** on this topic under the US-Japan DORAEMON initiative - stay tuned!

Conclusions

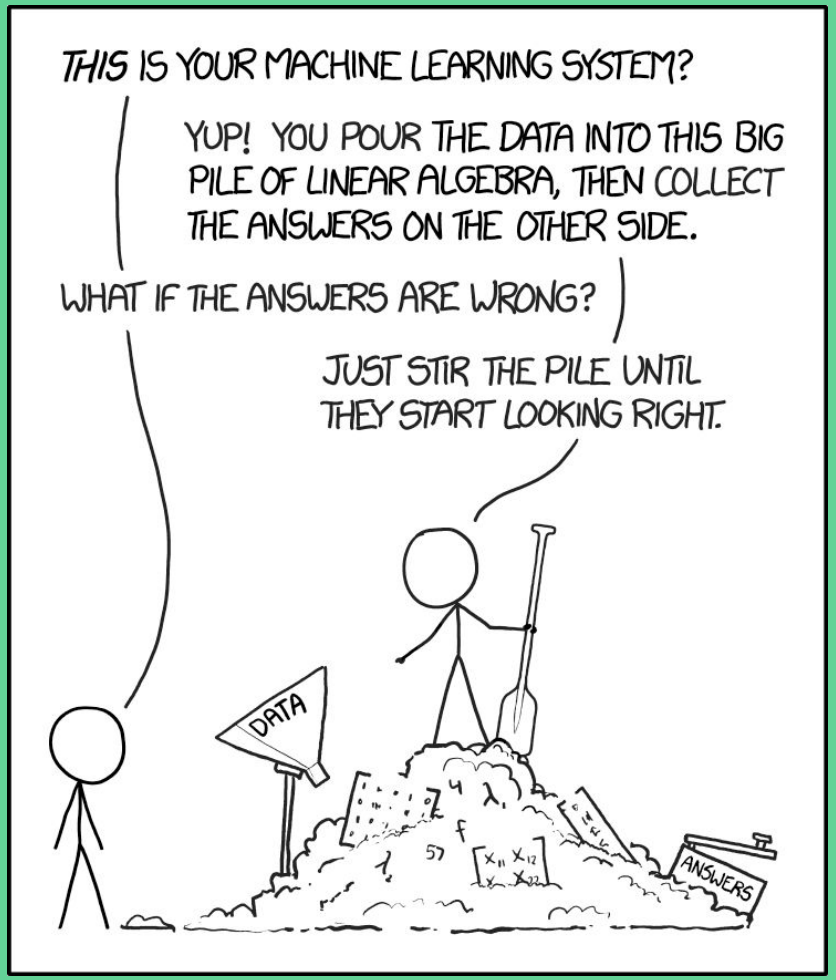
ML-based classifiers can be used to **estimate the likelihood ratio between arbitrary distributions** as a function of any variables we use as input. We've seen how this can be used for:

1. Unbinned unfolding using many observables, with better accuracy
2. Saving computational resources by reweighting truth-level outputs from different event generators, with validity over more of the phase space

But caution: this lets us use more of the information available to us, but it's up to us to make sure the information we use is valid! For instance:

- Reconstructed objects without proper detector systematics
- Weird outputs from neutrino event generators

Backup

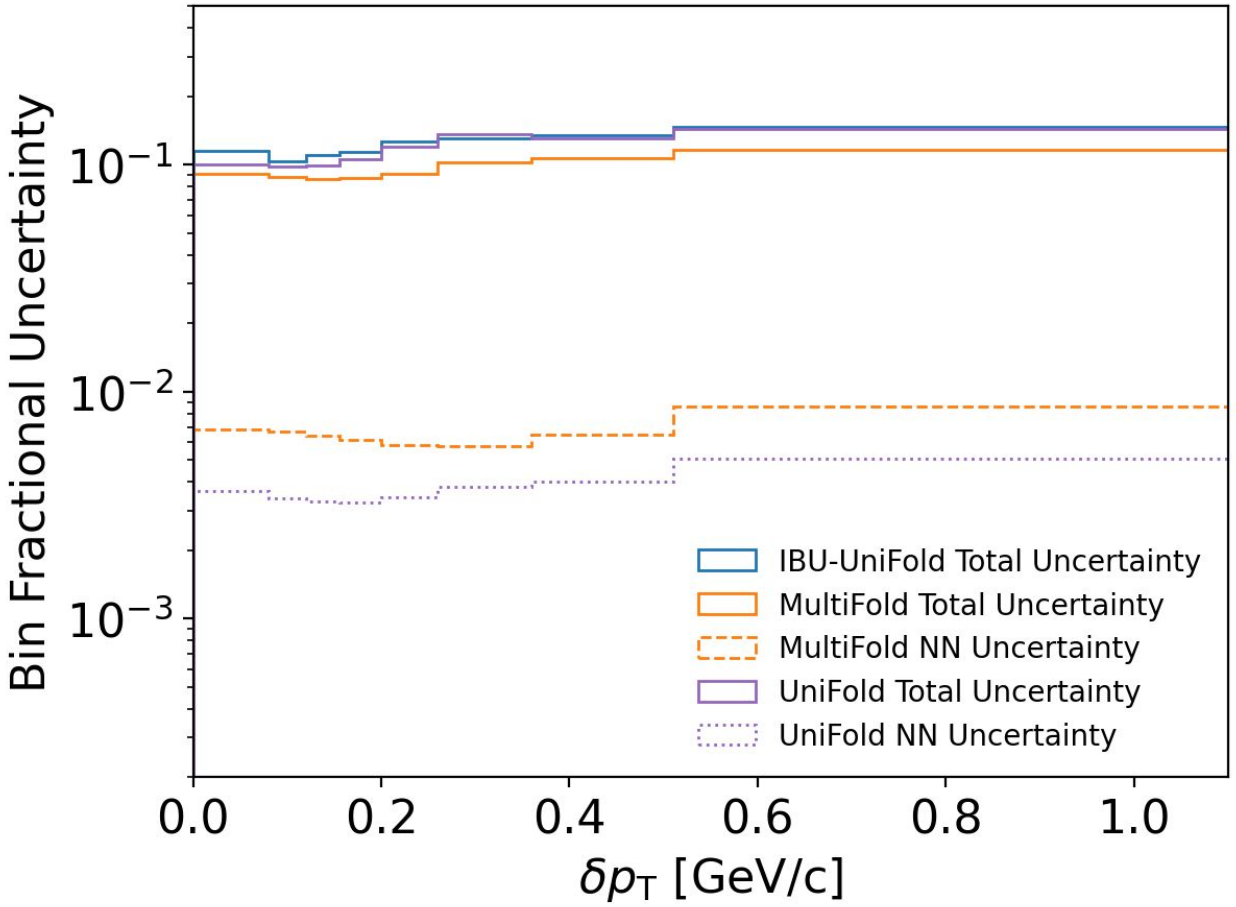

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Uncertainties

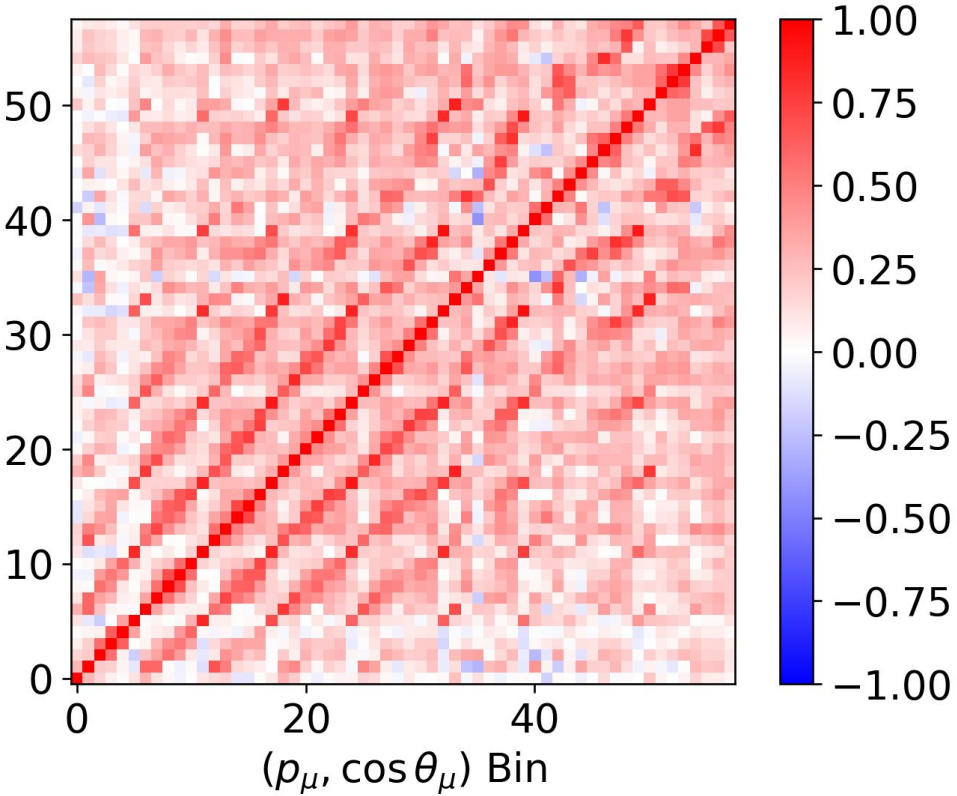
MultiFold achieves smaller bias and lower uncertainties than IBU

- Neural network training is a form of regularization

For the STVs, UniFold has smaller NN uncertainty than MultiFold, due to the relative simplicity of the network



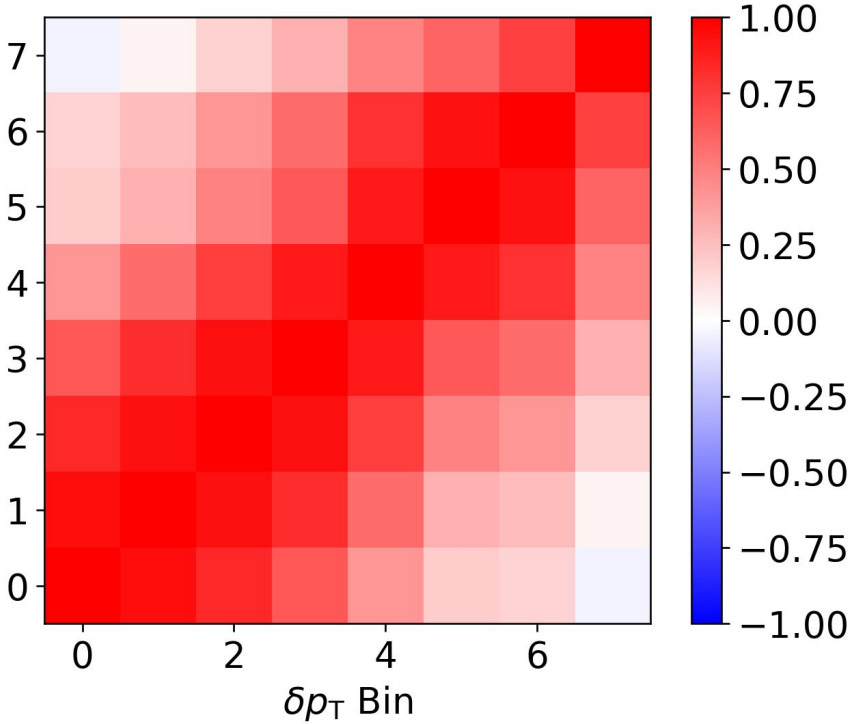
MultiFold Correlation Matrices

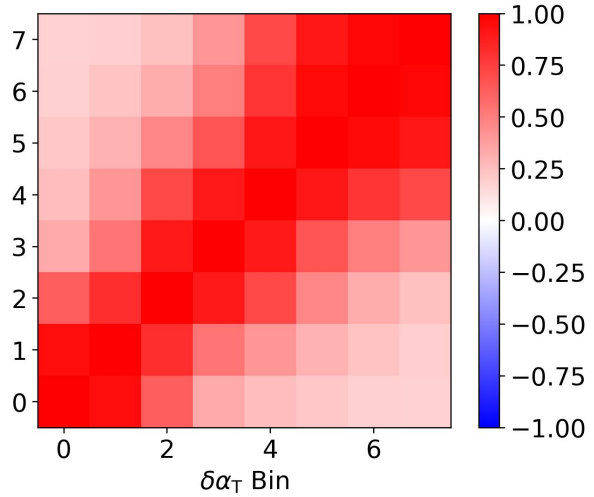


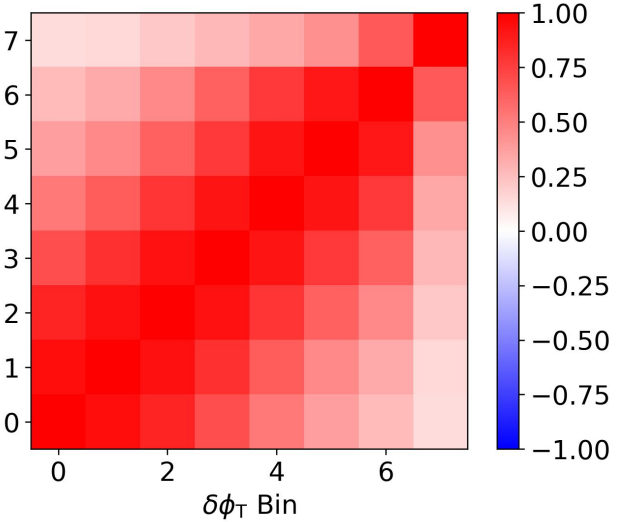
Correlation matrix excluding flux uncertainty

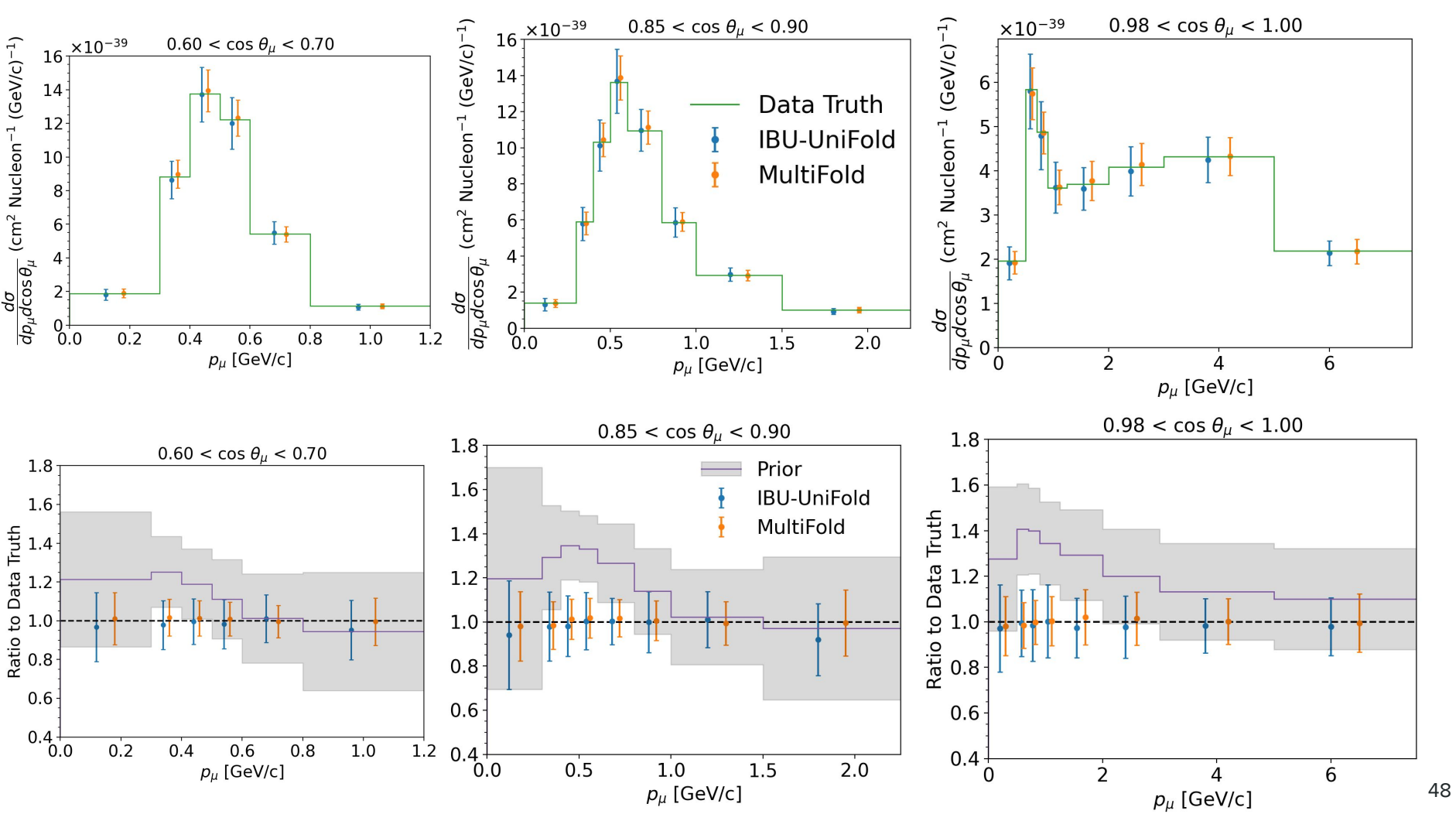
- Actual χ^2 is evaluated including effects of flux uncertainty in the covariance matrix

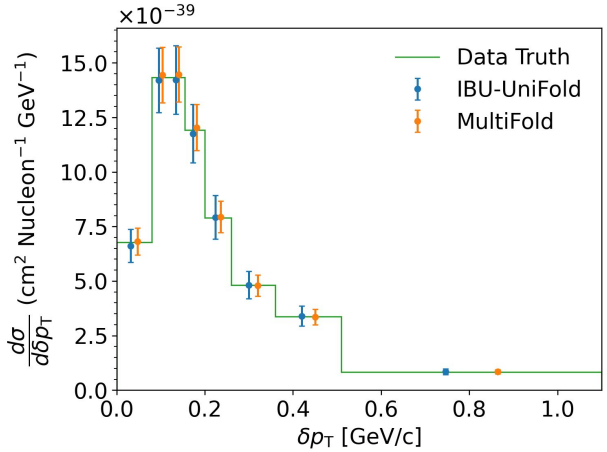
MultiFold Correlation Matrices

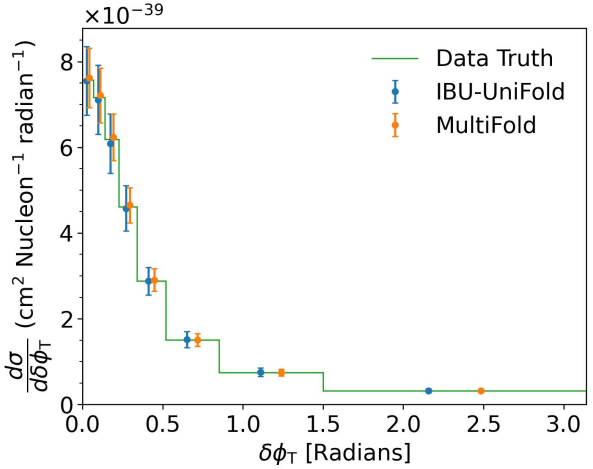


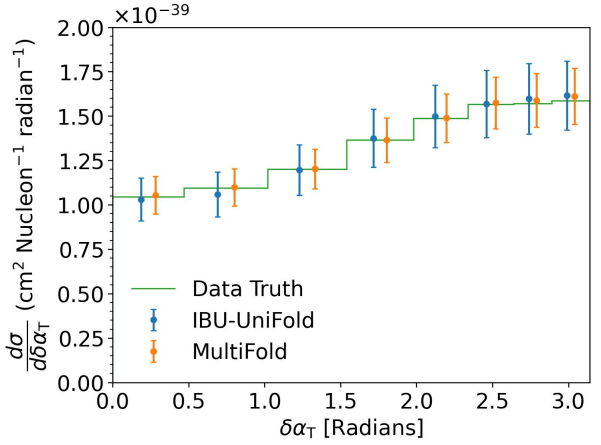


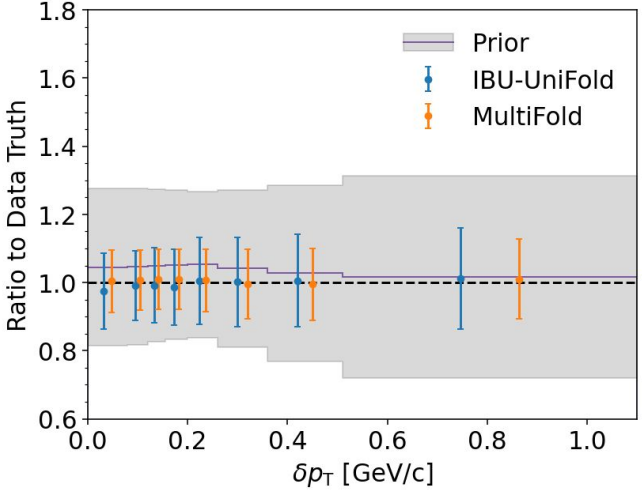


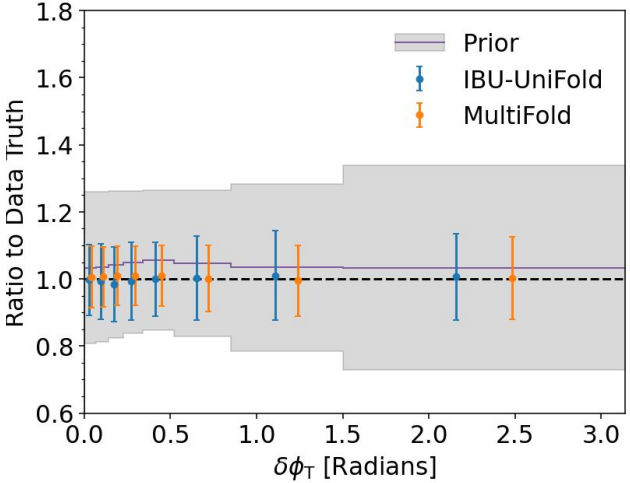

0.60 < cos θμ < 0.70
0.85 < cos θμ < 0.90
0.98 < cos θμ < 1.00
×10−39
dσ/dpμdcosθμ (cm² Nucleon⁻¹ (GeV/c)⁻¹)
pμ [GeV/c]
Data Truth
IBU-UniFold
MultiFold
Prior
Ratio to Data Truth

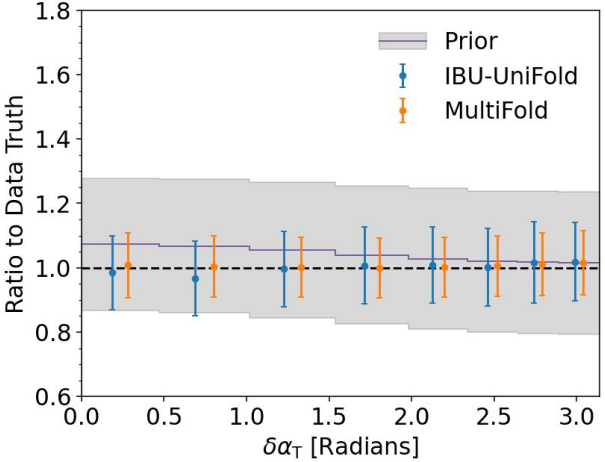

×10⁻³⁹
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{d\delta p_T}$ (cm² Nucleon⁻¹ GeV⁻¹)
δp_T [GeV/c]


×10⁻³⁹
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{d\delta\phi_T}$ (cm² Nucleon⁻¹ radian⁻¹)
$\delta\phi_T$ [Radians]


×10⁻³⁹
Data Truth
IBU-UniFold
MultiFold
$\frac{d\sigma}{d\delta\alpha_T}$ (cm² Nucleon⁻¹ radian⁻¹)
$\delta\alpha_T$ [Radians]


Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
δp_T [GeV/c]


Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
$\delta\phi_T$ [Radians]

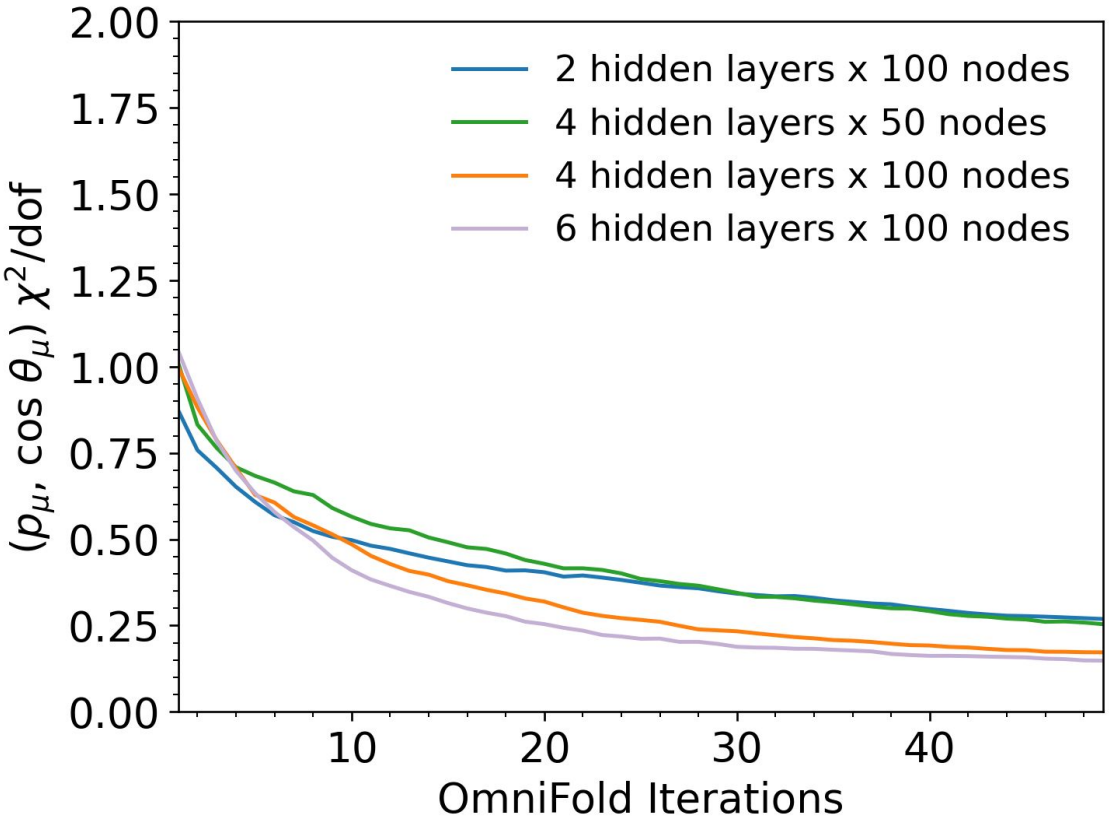

Prior
IBU-UniFold
MultiFold
Ratio to Data Truth
$\delta\alpha_T$ [Radians]

Neural Network Sizes

Networks that are too small limit the performance

- The likelihood ratio approximation in each step becomes too inaccurate

But no gain from further size increases at a certain point

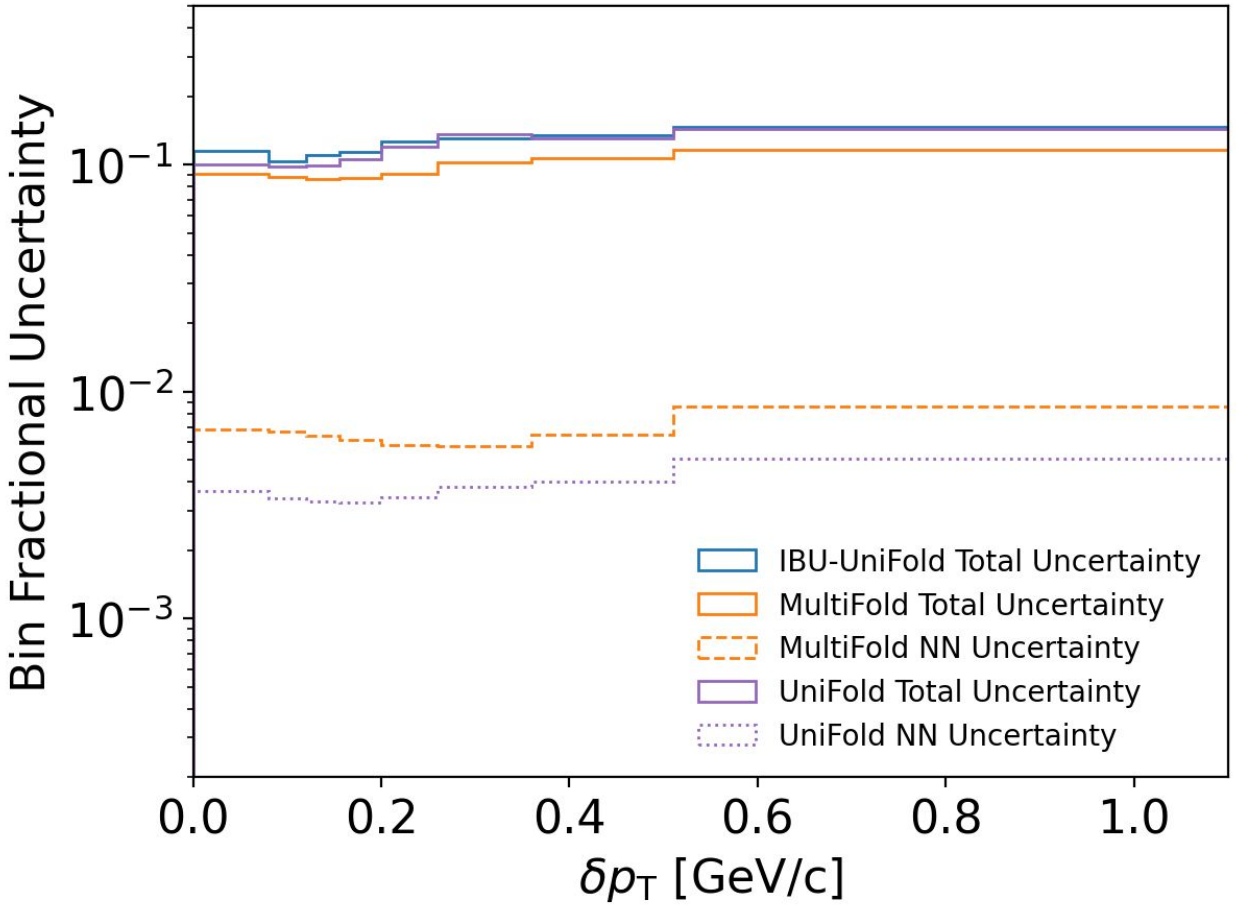


Uncertainties

MultiFold achieves smaller bias and lower uncertainties than IBU

- Neural network training is a form of regularization

For the STVs, UniFold has smaller NN uncertainty than MultiFold, due to the relative simplicity of the network

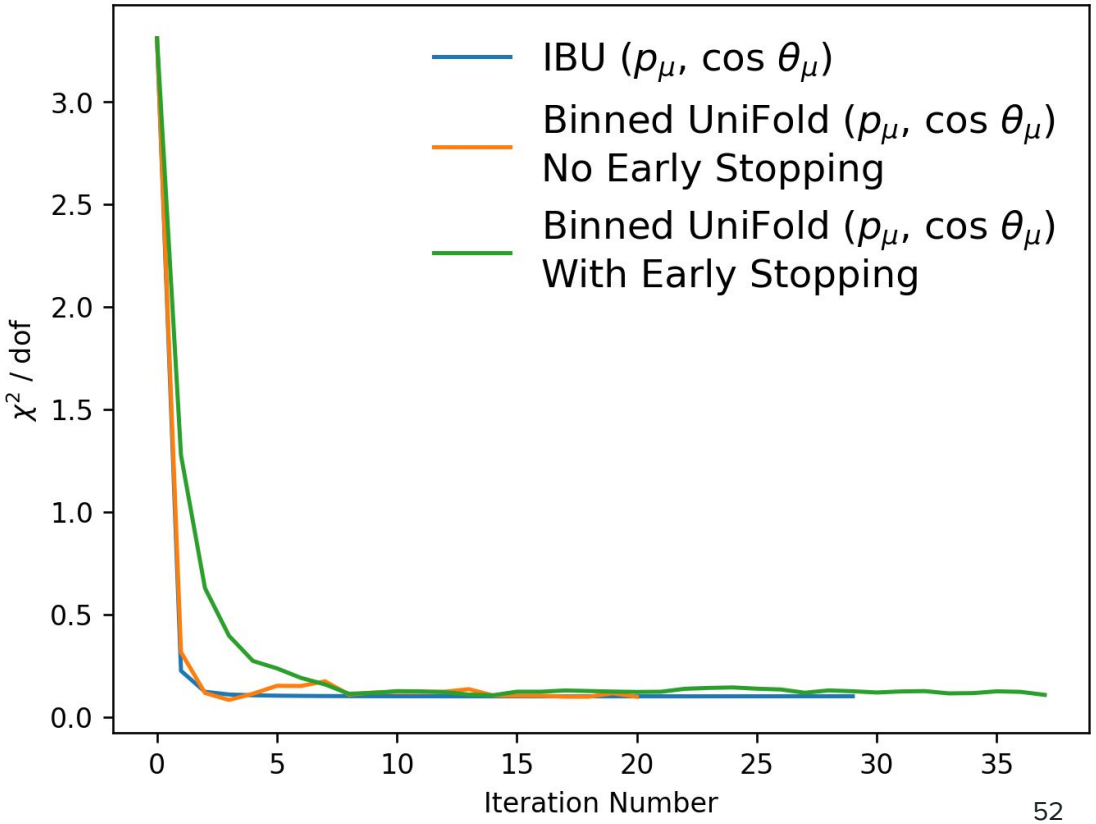


IBU-UniFold Equivalence to IBU

In the absence of any regularization on the NN training procedure, IBU-UniFold gives the same result as an off-the-shelf IBU algorithm

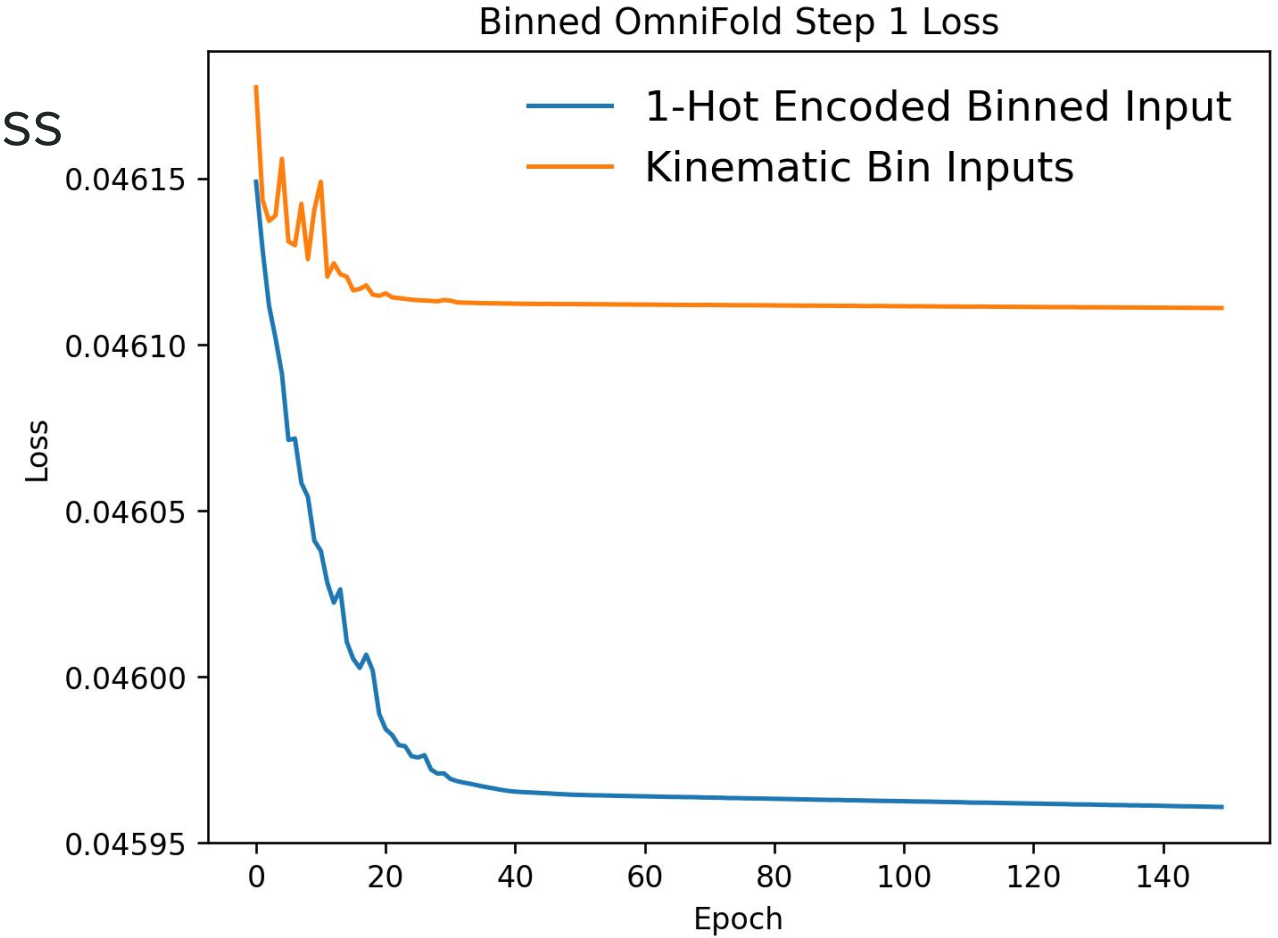
Even with regularization, they converge to the same values

Further studies on IBU/UniFold equivalence: JINST 20 (2025) P05034

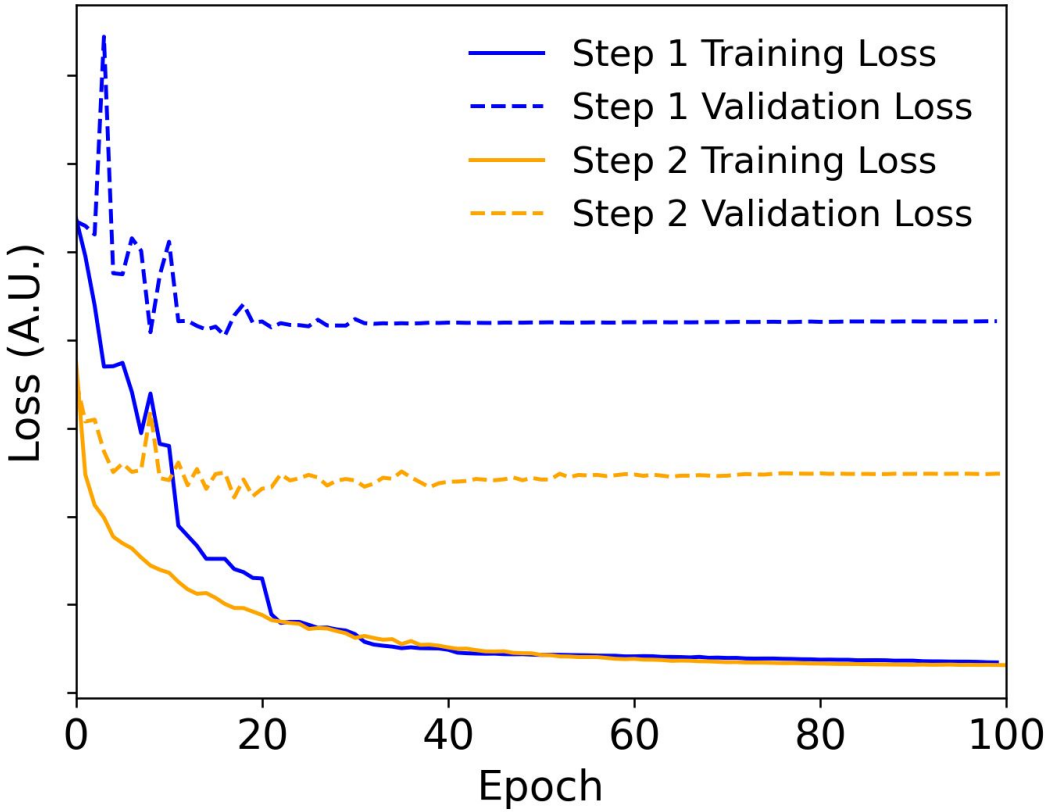


Binned Training Loss

Difference in performance for IBU-UniFold (1-hot encoded) vs Binned-UniFold (kinematic bins) visible in training loss curves



Training/Validation Loss



Example of training/validation loss curves for a step 1 + step 2 iteration of OmniFold

- Arbitrary offset on y-axis for each curve for plotting convenience

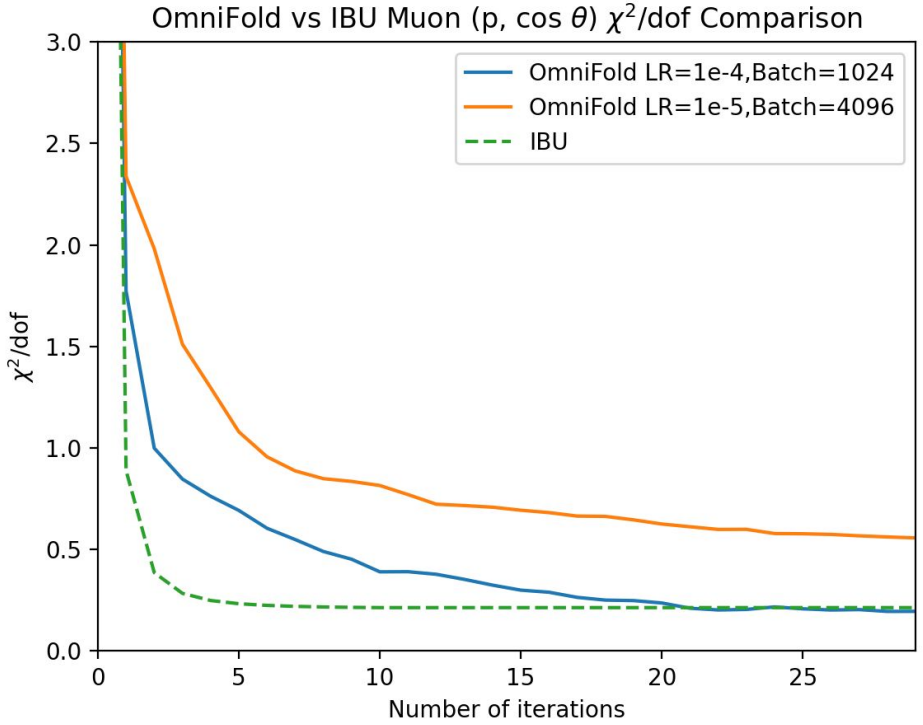
Actual result would be terminated after 15 epochs of no improvement in validation loss

Training Parameters

Our final results have been with the “best” NN training settings

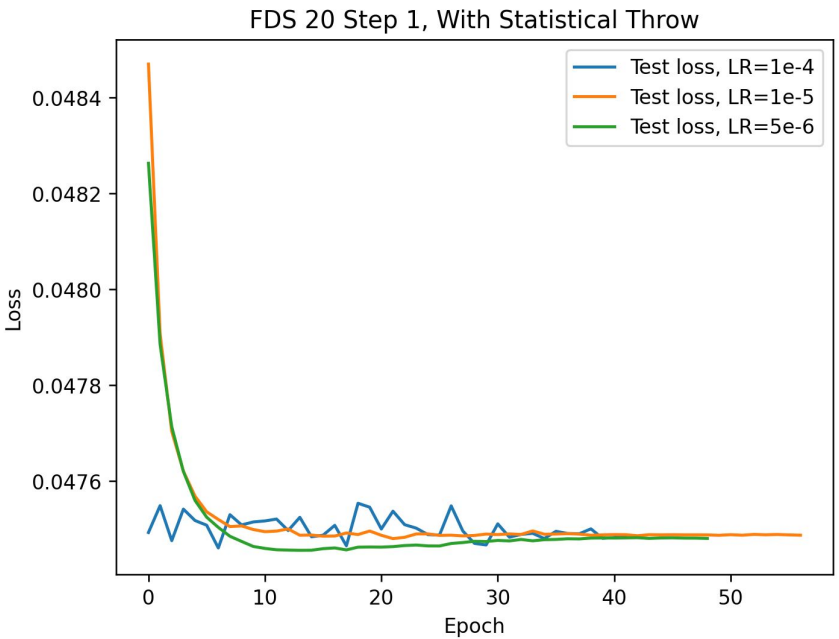
But in our scans we have found many settings that yield worse performance

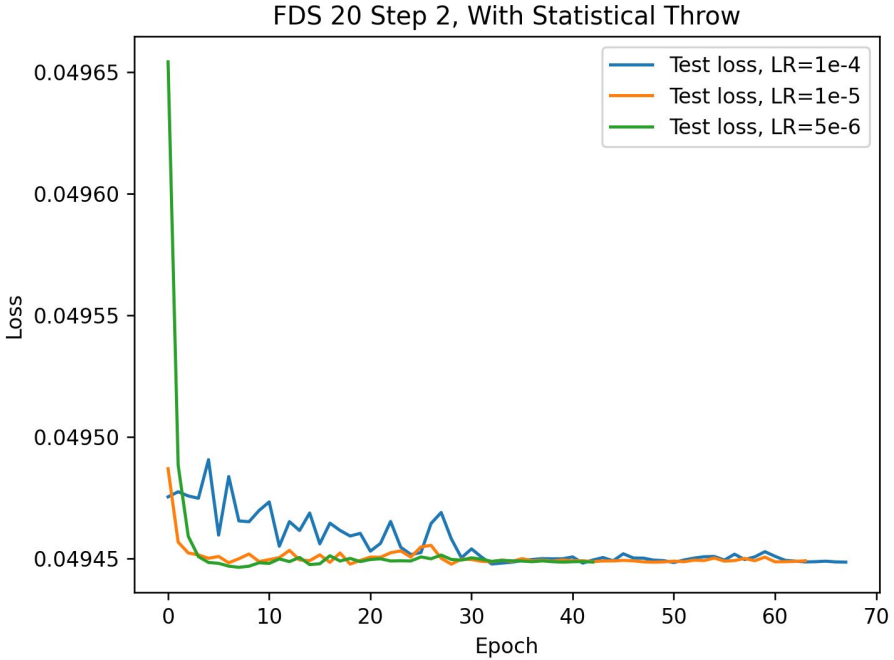
- Must be careful about tuning parameters before applying to real data!



Learning Rates

Absent other changes, different learning rates converge on the same loss

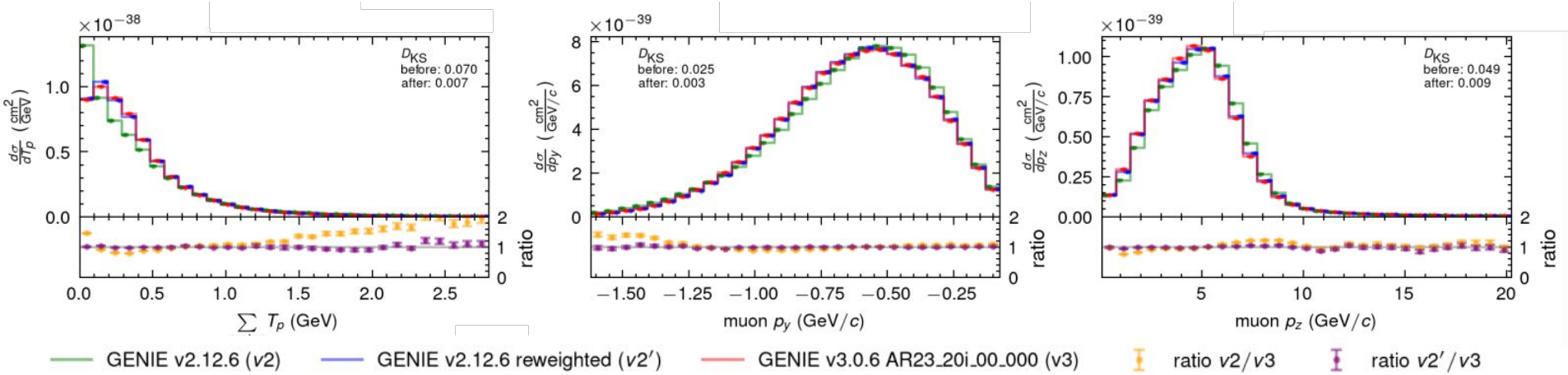




BDT Reweighting for Neutrino Generators

Prior work using **BDTs for multivariate reweighting** of neutrino generator outputs has already been done in MINERvA (pictured) and DUNE

But BDTs are generally not powerful enough to cover the entire possible phase space of interest from generators



arXiv:2510.07463

Phase Space Restrictions

Theoretically, we can reweight anywhere the phase spaces overlap

In practice, some areas will be hard and should be cut

The Genie10a/10b results worsen significantly on events with > 2 protons

- Results shown in these slides **eliminated those events from comparison**