1 Carnegie Mellon University

2 UC Berkeley

3 名古屋大学

4 Stanford University

Unbinned Unfolding in the Presence of Nuisance Parameters

The 2nd "AI+HEP in East Asia" workshop

Joint work of Huanbiao (Richard) Zhu[1], Krish Desai[2], Mikael Kuusela[1], Vinicius Mikuni[3], Benjamin Nachman[4], Larry Wasserman[1]

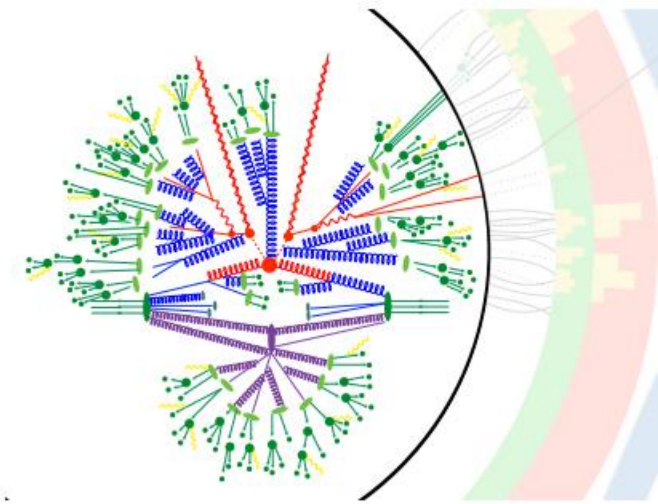
Based on arXiv:2512.07074

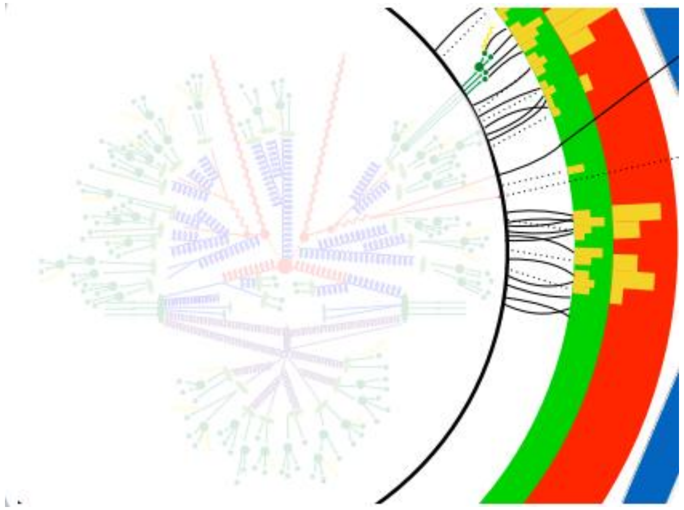
Outline

- Unfolding
- Machine-Learning based Method: **OmniFold**
- OmniFold in the Presence of Nuisance Parameters: **Profile OmniFold**
- Case Studies

Unfolding

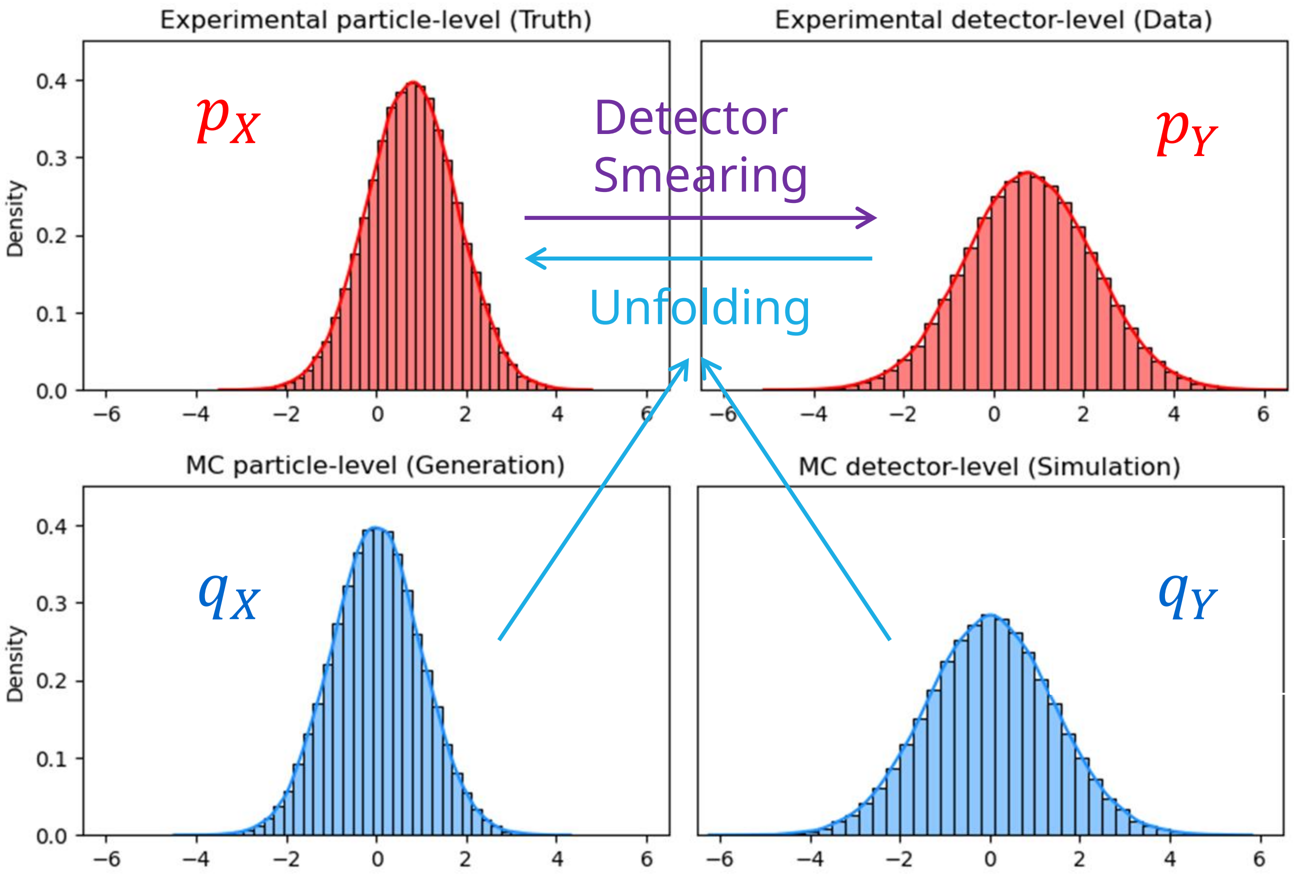
- Unfolding is the procedure of correcting measurement distortion due to smearing effects such as finite resolution of the detector





FIGURES SOURCE: NACHMAN "UNFOLDING: MACHINE LEARNING APPROACHES"
NUXTRACT, CERN (HYBRID), OCT. 2023.

Unfolding: A Simple Example

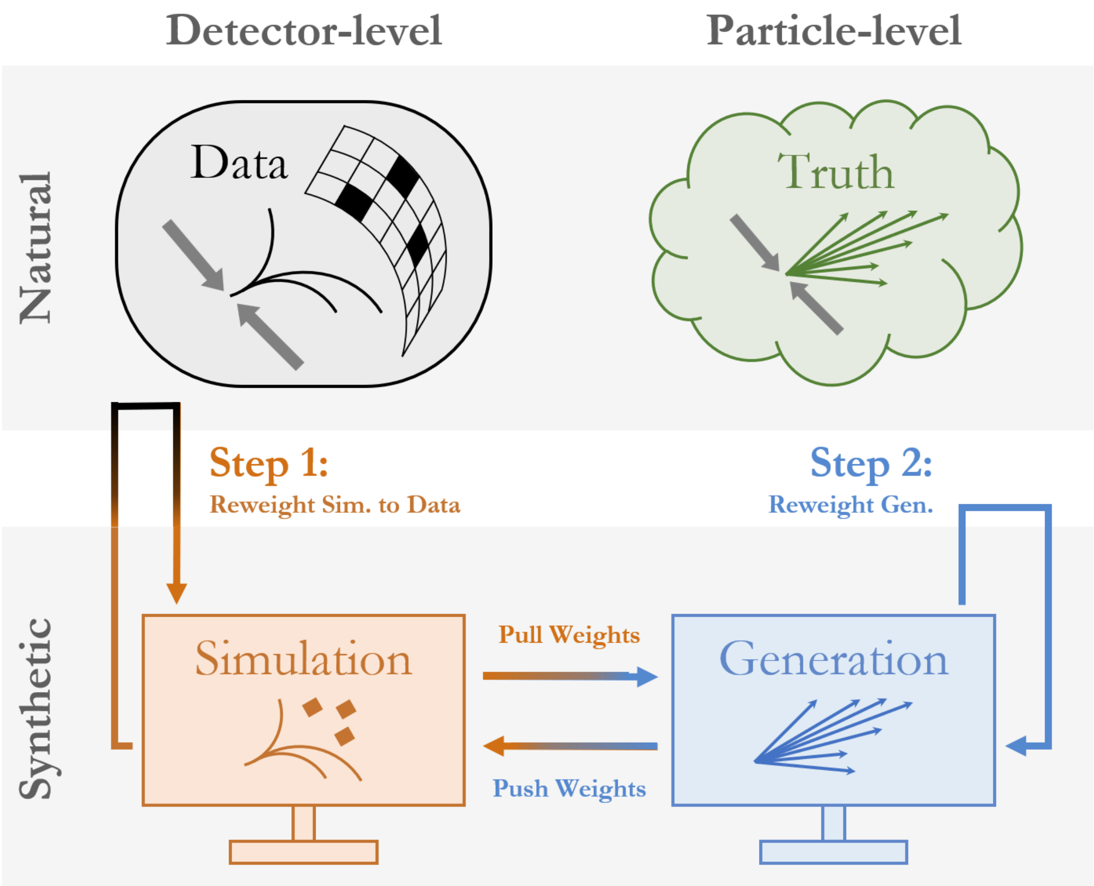


$$p_Y(y) = \int_{x \in X} k(y,x) p_X(x) dx \,,$$

$$k(y,x) = p(\text{smeared observation } y \mid \text{true event } x)$$

$$q_Y(y) = \int_{x \in X} k(y,x) q_X(x) dx$$

OmniFold [Andreassen et al. (2020)]

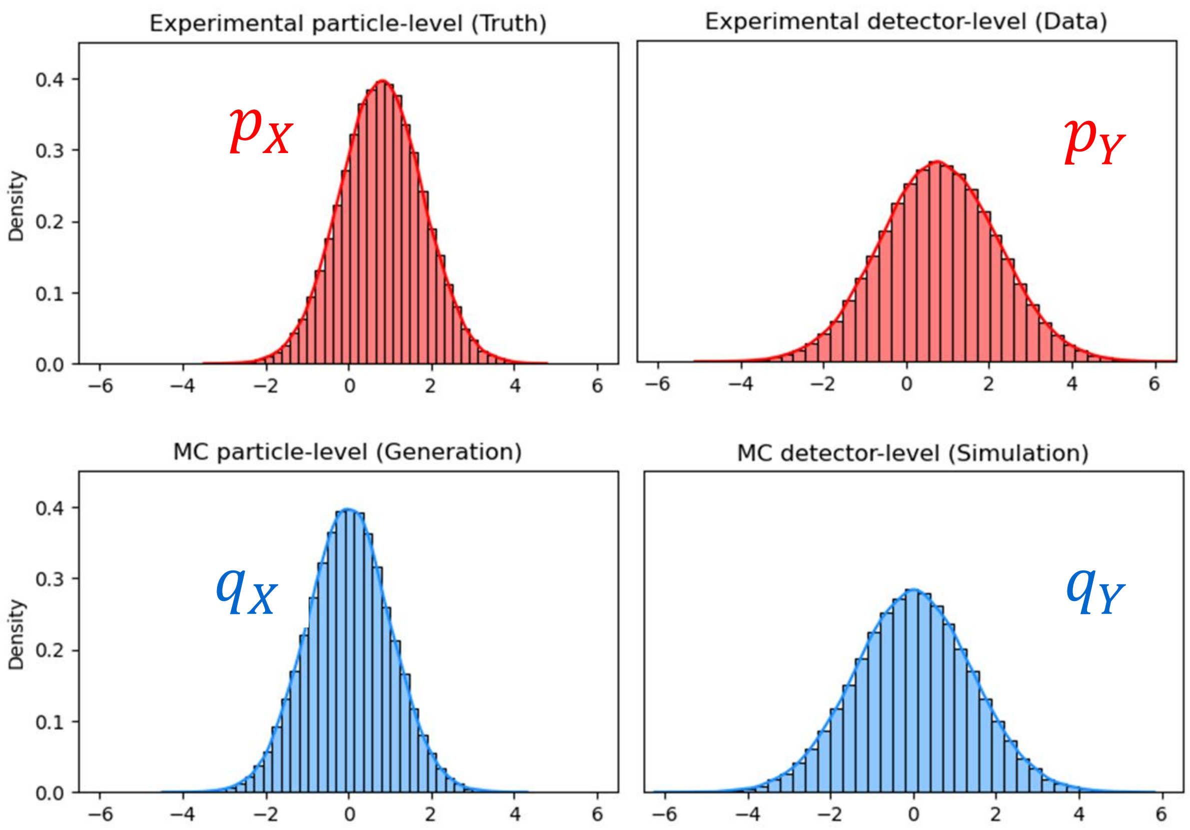


- OmniFold is one of the earliest machine learning-based approaches that have been proposed to perform **unbinned unfolding**

- Successfully applied to experimental data from the Large Hadron Collider (LHC) at CERN and other particle and nuclear physics experiments (H1 Collaboration, 2022a,b; Komiske, Kry hin and Thaler, 2022; LHCb Collaboration, 2023a; H1 Collaboration, 2023b,c; Song, 2023; Pani, 2024; CMS Collaboration, 2024a; ATLAS Collaboration, 2024b,c; H1 Collaboration, 2024d; ATLAS Collaboration, 2025a; H1 Collaboration, 2025b; Badea et al., 2025; Huang et al., 2025; Canelli et al., 2025).

ANDREASSEN ET AL. (2020). OMNIFOLD: A METHOD TO SIMULTANEOUSLY UNFOLD ALL OBSERVABLES. PHYSICS REIVEW LETTER, 124.

OmniFold – Iterative Reweighting



Goal: Find $\nu^*(x)$ such that $\nu^*(x)q(x) = p(x)$

Initialization: $\nu^{(0)}(x)q(x) = q(x)$

Step 1: Detector-level reweighting

$$r^{(k)}(y) = \frac{p(y)}{\tilde{q}^{(k)}(y)},$$

where $\tilde{q}^{(k)}(y) = \int \nu^{(k)}(x)q(x,y)dx$

Step 2: Particle-level reweighting

$$\nu^{(k+1)}(x) = \nu^{(k)}(x)\frac{\tilde{q}^{(k)}(x)}{q(x)},$$

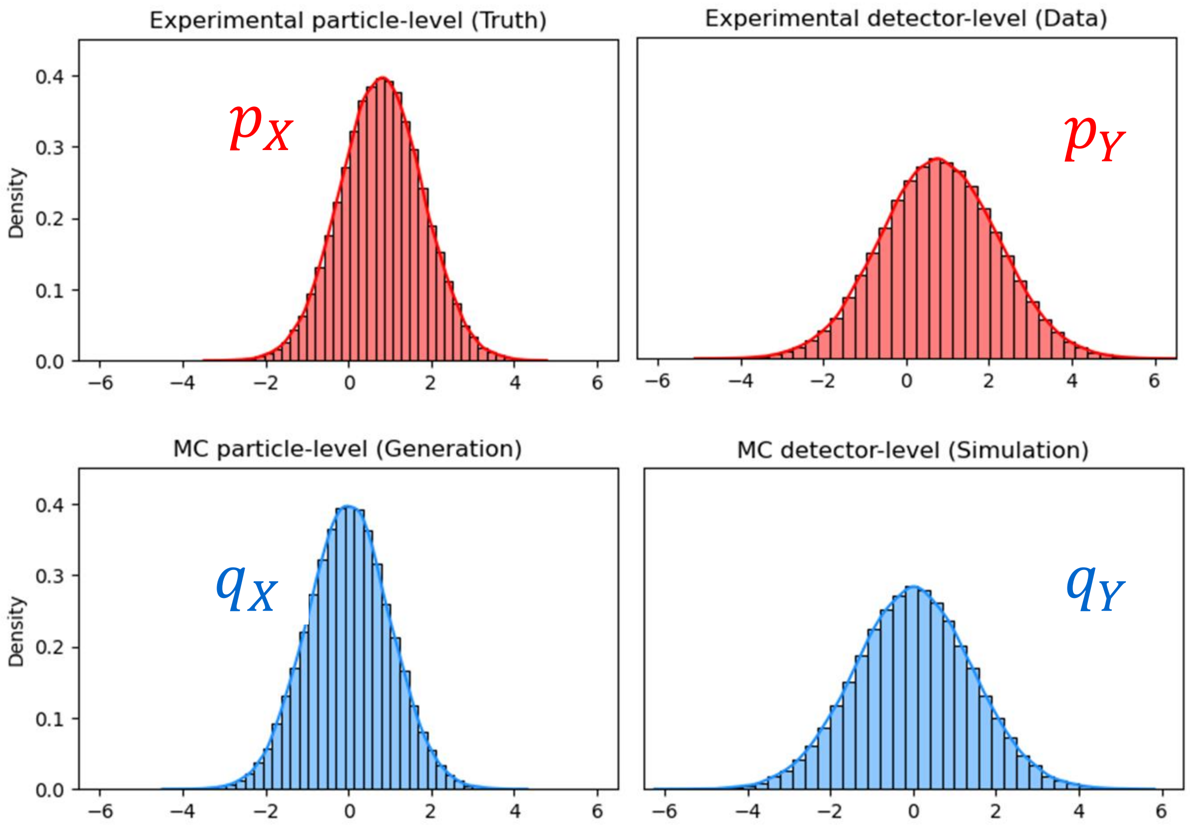
where $\tilde{q}^{(k)}(x) = \int r^{(k)}(y)q(x,y)dy$

Classifier Trick!

Train a classifier f to distinguish samples from $\mathcal{P}_1$ to $\mathcal{P}_2$

Then we have $\frac{\mathcal{P}_1}{\mathcal{P}_2} \propto \frac{f}{1-f}$

OmniFold – Iterative Reweighting



Goal: Find $\nu^*(x)$ such that $\nu^*(x)q(x) = p(x)$

Initialization: $\nu^{(0)}(x)q(x) = q(x)$

Step 1: Detector-level reweighting

$r^{(k)}(y$

where $\tilde{q}^{(k)}(y) =$

$\nu^{(\infty)}(x)q(x) \rightarrow p(x)$?

Step 2: Particle-level reweighting

$$\nu^{(k+1)}(x) = \nu^{(k)}(x)\frac{\tilde{q}^{(k)}(x)}{q(x)},$$

where $\tilde{q}^{(k)}(x) = \int r^{(k)}(y)q(x,y)dy$

<u>Classifier Trick!</u>

Train a classifier f to distinguish samples from $\mathcal{P}_1$ to $\mathcal{P}_2$

Then we have $\frac{\mathcal{P}_1}{\mathcal{P}_2} \propto \frac{f}{1-f}$

OmniFold – EM Algorithm

- Maximize the population-level log-likelihood

$$\ell(\nu) = \int p(y) \log \int p(y|x)\nu(x)q(x)dxdy$$
$$\text{subject to} \int \nu(x)q(x)dx = 1$$

- Expectation-Maximization (EM) algorithm: in each iteration, maximize

$$Q\left(\nu; \nu^{(k)}\right) = \int p(y) \int p\left(x \middle| y, \nu^{(k)}\right) \log[p(y|x)\nu(x)q(x)]dxdy$$

$$\Longrightarrow \quad \nu^{(k+1)}(x) = \text{argmax}_\nu Q\left(\nu(x)\,; \nu^{(k)}\right)$$
$$\nu^{(k+1)}(x) = \nu^{(k)}(x) \int \frac{p(y)}{\int p(y|x')\nu^{(k)}(x')q(x')dx'} p(y|x)dy$$

OmniFold – EM Algorithm

$$\nu^{(k+1)}(x) = \nu^{(k)}(x) \int \frac{p(y)}{\int p(y|x')\nu^{(k)}(x')q(x')dx'} p(y|x)dy$$

$$= \nu^{(k)}(x) \cdot \frac{1}{q(x)} \int \frac{p(y)}{\int \nu^{(k)}(x')q(x',y)dx'} q(x,y)dy$$

KEY ASSUMPTION

$p(y|x) = q(y|x)$

$\nu^{(\infty)}(x)q(x) \to p(x)$?

Step 1:

$$r^{(k)}(y) = \frac{p(y)}{\tilde{q}^{(k)}(y)},$$

where $\tilde{q}^{(k)}(y) = \int \nu^{(k)}(x)q(x,y)dx$ (reweighted q(y) at kth iteration)

Step 2:

$$\nu^{(k+1)}(x) = \nu^{(k)} \frac{\tilde{q}^{(k)}(x)}{q(x)},$$

where $\tilde{q}^{(k)}(x) = \int r^{(k)}(y)q(x,y)dy$ (reweighted q(x) at kth iteration)

Nuisance Parameters

- **One of the key assumptions** in OmniFold (and other ML methods):

$$p(y|x) = q(y|x)$$

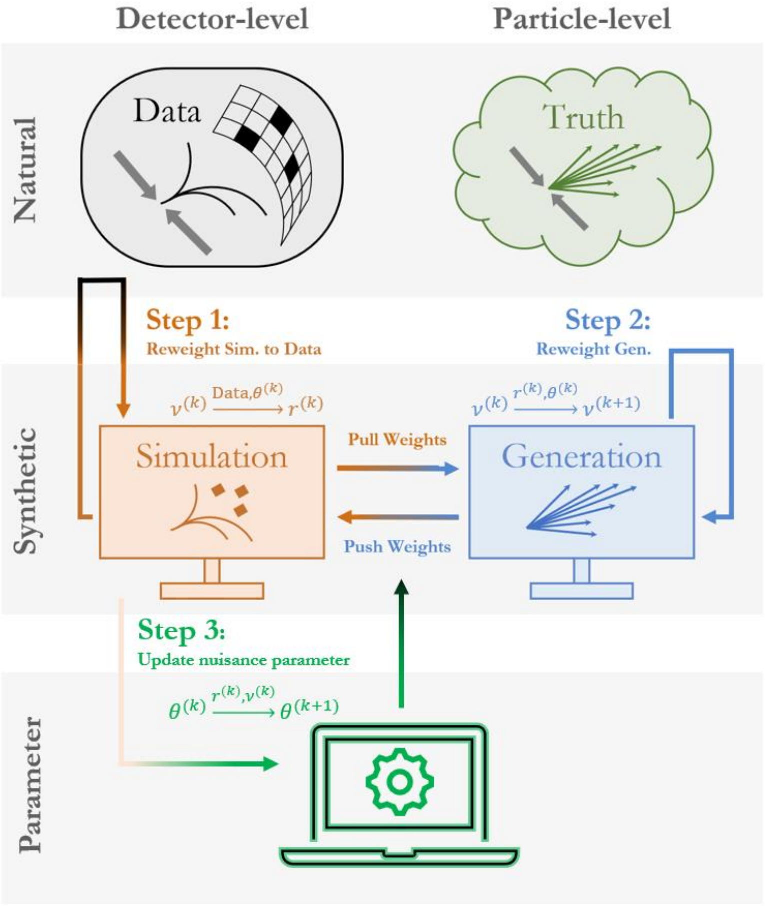
- Only approximately true, with the simulation depending on a number of **nuisance parameters** θ
- Forward model: $p_{Y,\theta}(y) = \int_{x \in X} p(y|x,\theta) p_X(x) dx$
- Log-likelihood

$$\ell(\nu,\theta) = \int p(y) \log \int w(y,x,\theta) q(y|x) \nu(x) q(x) dx dy$$

$$\text{subject to} \int \nu(x) q(x) dx = 1$$

$$w(y,x,\theta) = \frac{p(y|x,\theta)}{q(y|x)}$$

Profile OmniFold – EM Algorithm



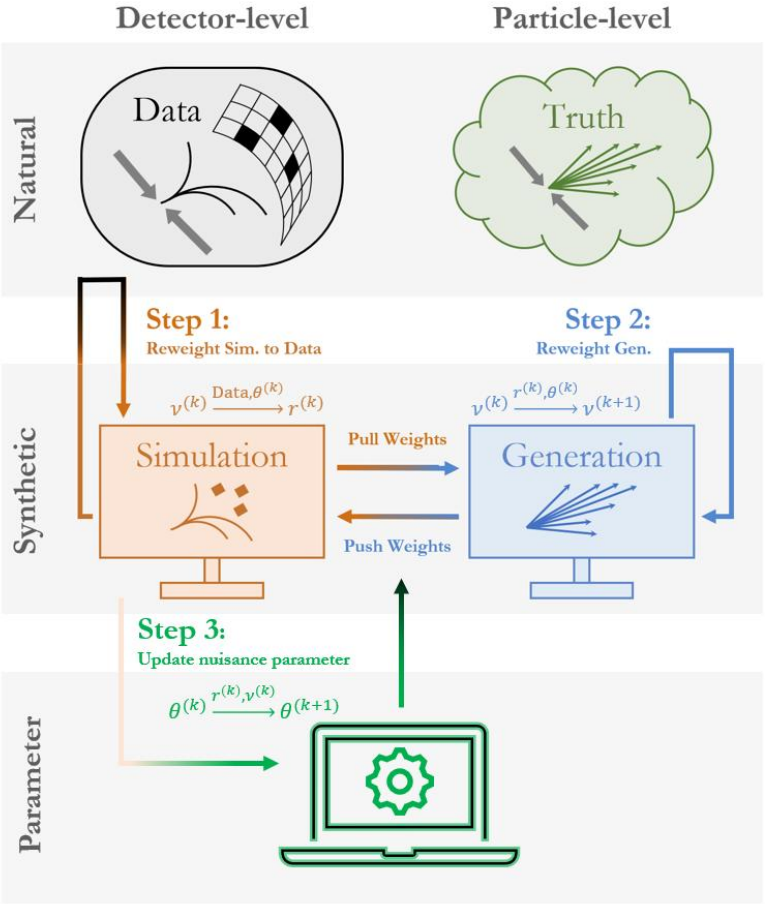
- Q function in the presence of nuisance parameters

$$Q\left(\nu,\theta;\nu^{(k)},\theta^{(k)}\right)$$
$$= \int p(y)\int p\left(x\middle|y,\nu^{(k)},\theta^{(k)}\right)\log[w(y,x,\theta)q(y|x)\nu(x)q(x)]dxdy$$
$$\text{subject to} \int \nu(x)q(x)dx = 1$$

- In each iteration, compute

$$\nu^{(k+1)},\theta^{(k+1)} = \text{argmax}_{\nu,\theta}Q\left(\nu,\theta;\nu^{(k)},\theta^{(k)}\right)$$

Profile OmniFold – EM Algorithm



Step 1: Detector-level reweighting

$$r^{(k)}(y) = \frac{p(y)}{\tilde{q}^{(k)}(y)},$$

where $\tilde{q}^{(k)}(y) = \int w\big(y, x, \theta^{(k)}\big)\nu^{(k)}(x)q(x,y)dx$

Step 2: Particle-level reweighting

$$\nu^{(k+1)}(x) = \nu^{(k)} \frac{\tilde{q}^{(k)}(x)}{q(x)},$$

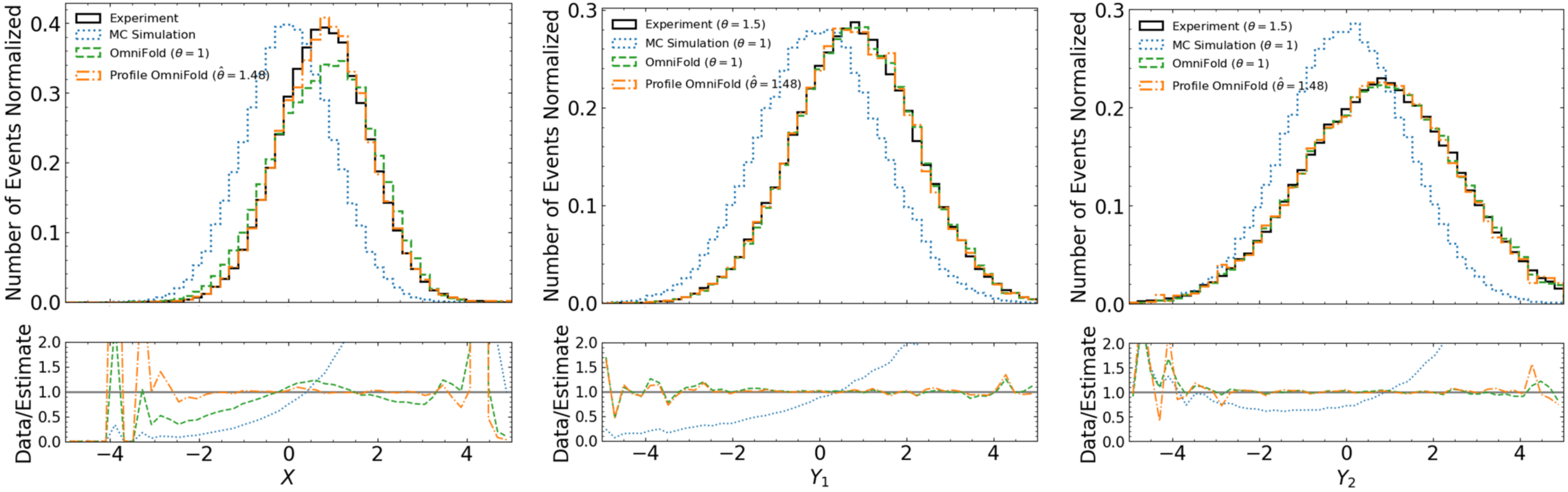
where $\tilde{q}^{(k)}(x) = \int w\big(y, x, \theta^{(k)}\big)r^{(k)}(y)q(x,y)dy$

Step 3: Nuisance parameter update

$$\theta^{(k+1)} = \operatorname{argmax}_\theta \mathbb{E}_{X,Y\sim q}\left[\nu^{(k)}(X)w\big(Y, X, \theta^{(k)}\big)r^{(k)}(Y)\log[\mathrm{w}(\mathrm{Y}, X, \theta)]\right]$$

Gaussian Example

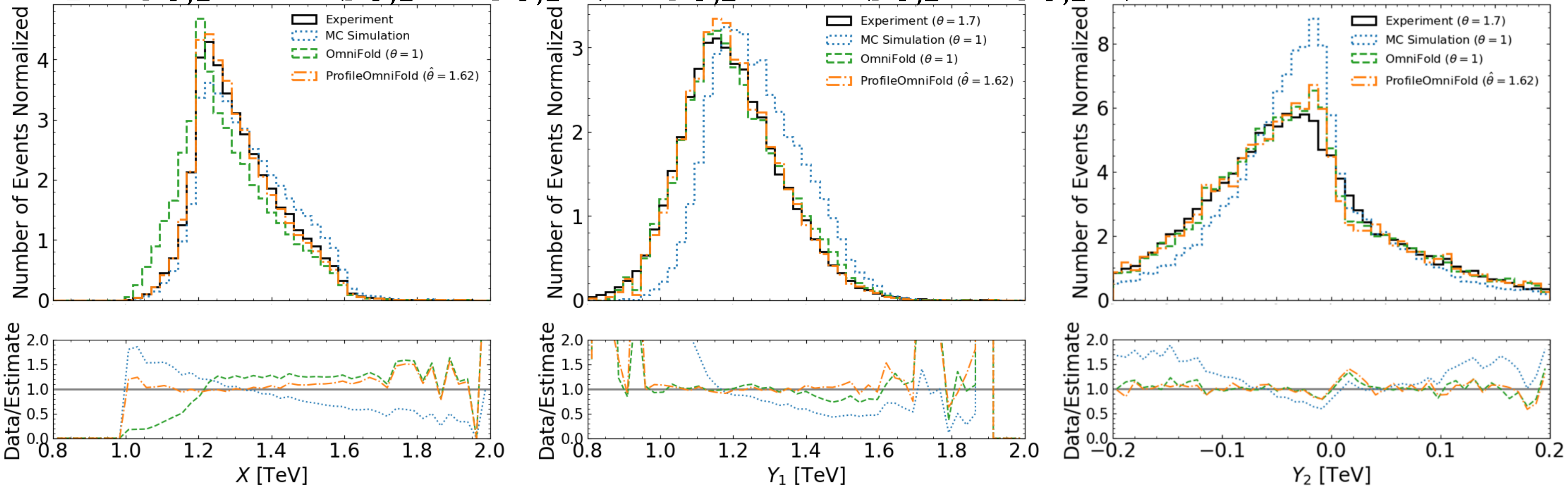
$X \sim N(\mu, \sigma^2), Z_1 \sim N(0,1), Z_2 \sim N(0, \theta^2),$
$Y_1 = X + Z_1,\ Y_2 = X + Z_2$



True $\theta = 1.5$, MC $\theta = 1.0$

CMS Open Data

$X = p_{T,1}^{\text{truth}} + p_{T,2}^{\text{truth}},\ Y_1 = p_{T,1}^{\text{truth}} + \theta(p_{T,1}^{\text{reco}} - p_{T,1}^{\text{truth}}) + p_{T,2}^{\text{truth}} + \theta(p_{T,2}^{\text{reco}} - p_{T,2}^{\text{truth}}),$
$Y_2 = p_{T,1}^{\text{truth}} + \theta(p_{T,1}^{\text{reco}} - p_{T,1}^{\text{truth}}) - p_{T,2}^{\text{truth}} - \theta(p_{T,2}^{\text{reco}} - p_{T,2}^{\text{truth}})$



True $\theta = 1.7$, MC $\theta = 1.0$

Summary

- We have developed a new machine learning-based unfolding algorithm, Profile OmniFold (POF), which extends the original OmniFold (OF) algorithm to the case where the forward model is not completely specified

- POF simultaneously updates the reweighting function and nuisance parameters, and shares similar steps as in OF, which allows for easy implementation while preserving many of its benefits

- In both Gaussian example and CMS open data, POF is able to accurately estimate the true distribution, whereas OF fails due to misspecfied forward model

References

- ANDREASSEN, A., KOMISKE, P. T., METODIEV, E. M., NACHMAN, B. and THALER, J. (2020). OmniFold: A Method to Simultaneously Unfold All Observables. Physical Review Letters 124.
- H1 COLLABORATION (2022a). Measurement of Lepton–Jet Correlation in Deep-Inelastic Scattering with the H1 Detector Using Machine Learning for Unfolding. Physical Review Letters 128.
- H1 COLLABORATION (2022b). Machine learning-assisted measurement of multi-differential lepton-jet correla tions in deep-inelastic scattering with the H1 detector. H1prelim-22-031.
- LHCB COLLABORATION (2023a). Multi differential Study of Identified Charged Hadron Distributions in Z Tagged Jets in Proton–Proton Collisions at $\sqrt{s}$ = 13 TeV. Physical Review D 108.
- H1 COLLABORATION(2023b).Unbinned Deep Learning Jet Substructure Measurement in High Q2 ep collisions at HERA. Physics Letters B 844 138101.
- H1 COLLABORATION (2023c). Machine learning-assisted measurement of azimuthal angular asymmetries in deep-inelastic scattering with the H1 detector. H1prelim-23-031.
- CMS COLLABORATION (2024a). Measurement of event shapes in minimum bias events from pp collisions at 13 TeV Technical Report, CERN, Geneva.
- ATLAS COLLABORATION (2024b). Simultaneous unbinned differential cross section measurement of twenty four Z+jets kinematic observables with the ATLAS detector. Physical Review Letters 133.
- ATLAS COLLABORATION (2024c). Measurement of Track Functions in ATLAS Run 2 Data.
- H1 COLLABORATION (2024d). Machine Learning-Assisted Measurement of Lepton–Jet Azimuthal Angular Asymmetries in Deep-Inelastic Scattering at HERA.
- ATLAS COLLABORATION (2025a). Measurement of Jet Track Functions in pp Collisions at $\sqrt{s}$ = 13 TeV with the ATLAS Detector. 22
- H1 COLLABORATION (2025b). Towards Unfolding All Particles in High Q2 DIS Events. H1 preliminary result.

arXiv:
2512.07074

Link to arXiv

Questions

Backup

Binned Unfolding

- D'Agostini Iteration[1]

$$\max_{\boldsymbol{\lambda}} \mathrm{L}(\boldsymbol{\lambda}; \boldsymbol{y}) = \sum_i \frac{\left(\sum_j K_{ij}\lambda_j\right)^{y_i}}{y_i!} e^{-\sum_j K_{ij}\lambda_j}$$

- E-Step

$$Q\left(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(\boldsymbol{k})}\right) = \mathbb{E}\left[\ell(\boldsymbol{\lambda}; \boldsymbol{x}, \boldsymbol{y}) \middle| \boldsymbol{y}, \boldsymbol{\lambda}^{(\boldsymbol{k})}\right]$$

- M-Step

$$\boldsymbol{\lambda}^{(\boldsymbol{k+1})} = argmax_{\boldsymbol{\lambda}}\, Q\left(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(\boldsymbol{k})}\right)$$

$$\lambda_j^{(k+1)} = \frac{\lambda_j^{(k)}}{\sum_i K_{ij}} \sum_i \frac{K_{ij} y_i}{\sum_l K_{il} \lambda_l^{(k)}}$$

Density Ratio Estimation through probabilistic classifier

- Let $x_1, \dots, x_n \sim p_1, x'_1, \dots, x_m{}' \sim p_2$ and assign $\mathrm{c} = 1$ to x and $\mathrm{c} = 0$ to x'

$$\frac{p_1(x)}{p_2(x)} = \frac{p(x|c=1)}{p(x|c=0)}$$

$$\propto \frac{p(c=1|x)}{p(c=0|x)}$$

- Train a classifier f to distinguish between $\{x_i\}_{i=1}^{n}$ and $\{x'_i\}_{i=1}^{m}$, then estimate

$$\frac{p_1(x)}{p_2(x)} \propto \frac{f(x)}{1 - f(x)}$$

W Function Training

- By definition,

$$w(y, x, \theta) = \frac{p\,(y|x, \theta)}{q(y|x)} = \frac{p(y, x|\theta)}{q(y, x)} \cdot \frac{q(x)}{p(x|\theta)}`$$

- Train two classifiers to learn two ratios.
- Data:

$$D_1 = \{X_i, Y_i, \theta_i\}, D_2 = \{X'_i, Y'_i, \theta'_i\}$$
$$Y_i \sim p(\cdot\,|X_i, \theta_i), Y_i' \sim q(\cdot\,|X'_i)$$

- Classifier 1: Distinguish $\{X_i, Y_i, \theta_i\}$ from $\{X'_i, Y'_i, \theta'_i\}$
- Classifier 2: Distinguish $\{X_i, \theta_i\}$ from $\{X'_i, \theta'_i\}$