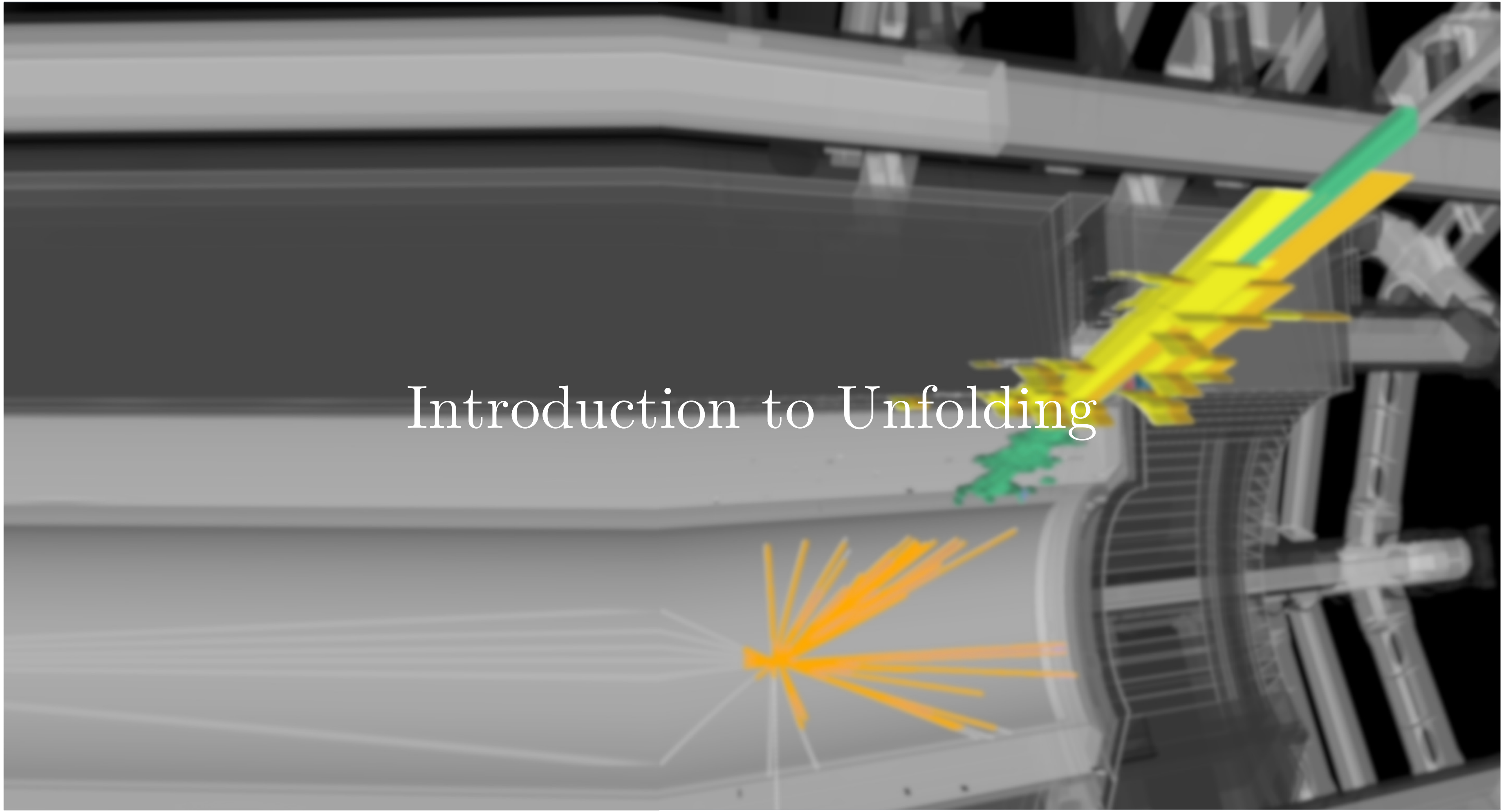
High-Dimensional Unfolding in Large Backgrounds

2507.06291

Adam Takacs (Heidelberg University)

with:
Alexandre Falcão

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



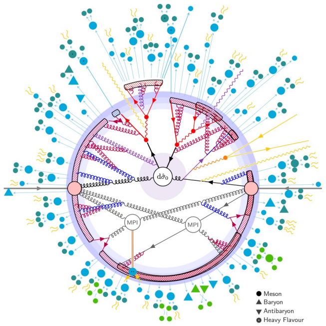
Introduction to Unfolding

What do experiments measure?

Measurement = True signal + Background + Detector effects

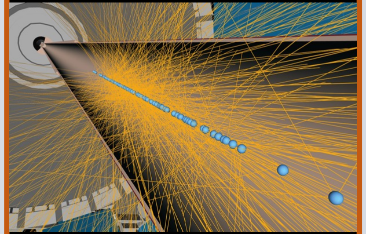
True signal

- Higgs.
- HI bulk
- **jets**

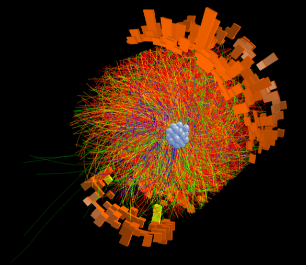


Background

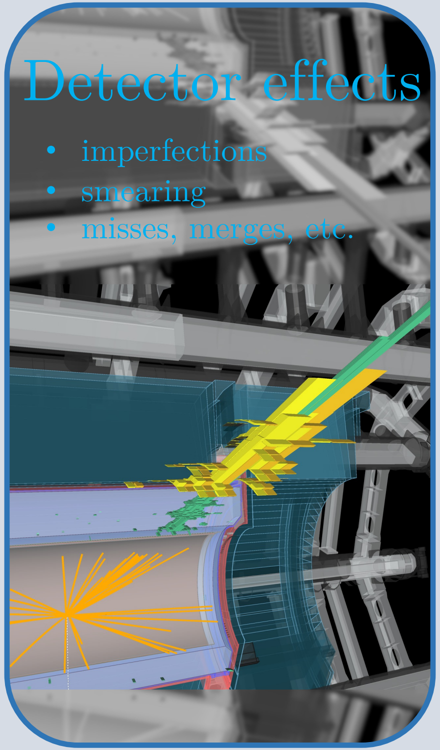
- pile-up



- **underlying event**



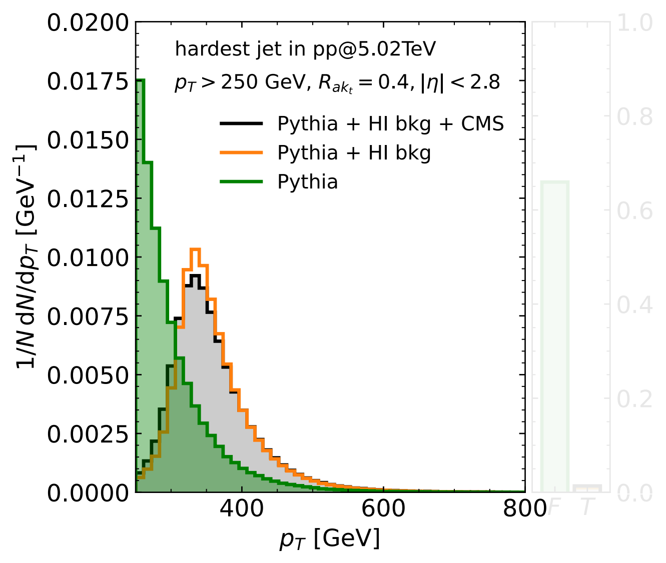
Detector effects

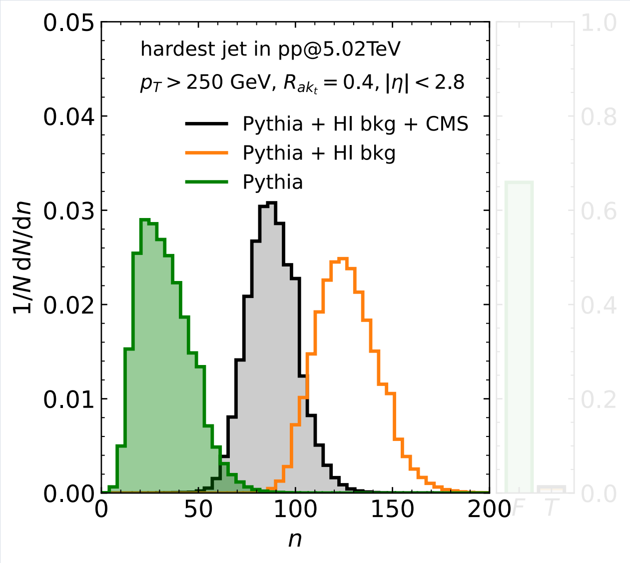


- imperfections
- smearing
- misses, merges, etc.

What do experiments measure?

Example: Pythia jets + HI background + CMS detector simulation





- HI bkg. adds to the jet energy.
- Detectors lose energy.

- HI bgk. adds to the jet multiplicity.
- Detectors lose particles.

Unfolding algorithm

[(Iterative) Bayesian Unfolding, D'Agostini 1994,2010]

Unfolding = correcting for bkg. and detector effects

Inference task: $$\underbrace{x(t_i)}_{\text{truth count in } t_i \text{ bin}} = \sum_j \underbrace{p(t_i|m_j)}_{\text{prob. of } m_j \to t_i} \underbrace{x(m_j)}_{\text{measured count in } m_j \text{ bin}}$$ (likelihood-free inference)

Bayesian inference:

$$p(t_i|m_j) = \frac{p(m_j|t_i)\,p_0(t_i)}{p_0(m_j)}$$

Forward simulation: prob. of $t_i \to m_j$

priors

Bayesian → Maximum likelihood unfolding:

$$x(t_i) = \lim_{n\to\infty} x_n(t_i) = \sum_j \frac{p(m_j|t_i)\,p_{n-1}(t_i)}{p_{n-1}(m_j)}\,x(m_j)$$

Unfolding algorithm

[(Iterative) Bayesian Unfolding, D'Agostini 1994,2010]

Unfolding = correcting for bkg. and detector effects

Inference task: $$x(t_i) = \sum_j p(t_i|m_j)\, x(m_j)$$ (likelihood-free inference)

truth count in t_i bin

prob. of $m_j \to t_i$

measured count in m_j bin

Bayesian inference:

$$p(t_i|m_j) = \frac{p(m_j|t_i)p_0(t_i)}{p_0(m_j)}$$

Forward simulation: prob. of $t_i \to m_j$

priors

what we have:

- MC samples
- Det simulations

Bayesian → Maximum likelihood unfolding:

$$x(t_i) = \lim_{n\to\infty} x_n(t_i) = \sum_j \frac{p(m_j|t_i)p_{n-1}(t_i)}{p_{n-1}(m_j)} x(m_j)$$

Unfolding algorithm

[(Iterative) Bayesian Unfolding, D'Agostini 1994,2010]

Unfolding = correcting for bkg. and detector effects

Inference task: $x(t_i) = \sum_j p(t_i|m_j)\, x(m_j)$ (likelihood-free inference)

truth count in t_i bin

prob. of $m_j \to t_i$

measured count in m_j bin

Bayesian inference:

$$p(t_i|m_j) = \frac{p(m_j|t_i)p_0(t_i)}{p_0(m_j)}$$

Forward simulation: prob. of $t_i \to m_j$

priors

what we have:

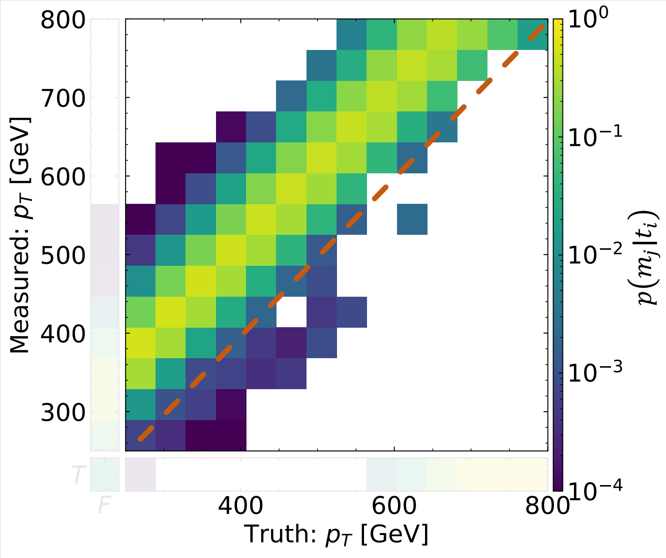
- MC samples
- Det simulations

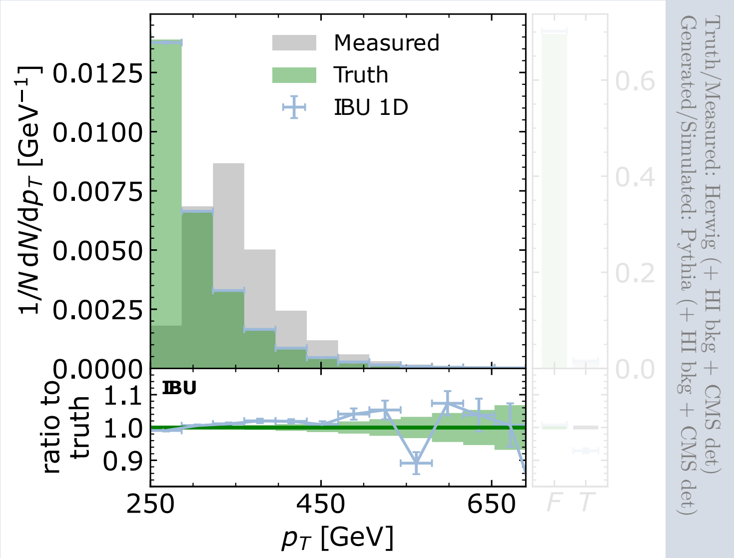
Bayesian → Maximum likelihood unfolding:

$$x(t_i) = \lim_{n\to\infty} x_n(t_i) = \sum_j \frac{p(m_j|t_i)p_{n-1}(t_i)}{p_{n-1}(m_j)} x(m_j)$$

Unfolding algorithm

Example: Pythia jets + HI background + CMS detector simulation





- Response mx: Pythia (+ HI bkg + CMS det)
- Unfolding Herwig (+ HI bkg + CMS det)

- Important statistical uncertainty.
- Deviations here contribute to sys. unc!

High-Dimensional Unfolding

Unfolding algorithm in high-dimension

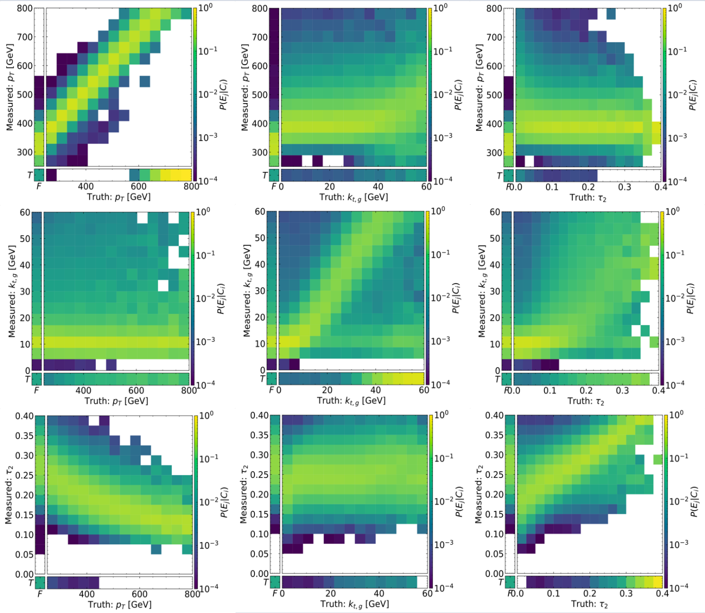
Same inference procedure:

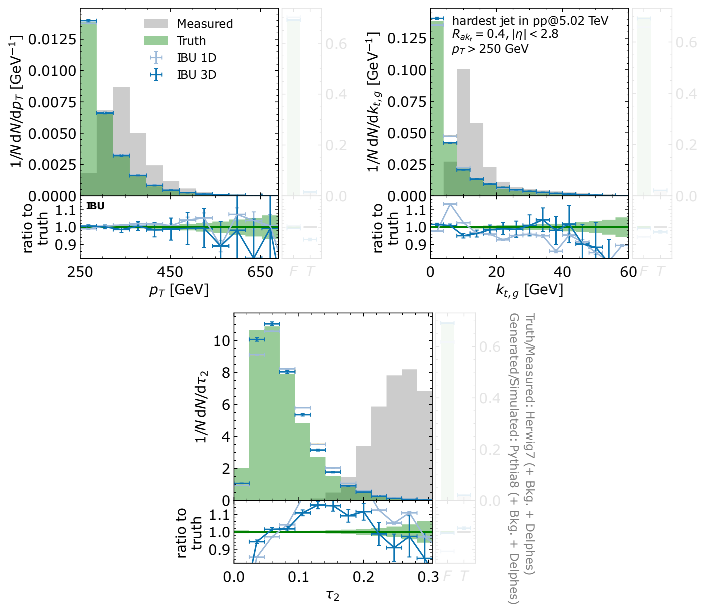
$$x([t_{i_1}, t_{i_2}, \dots]) = \sum_j p([t_{i_1}, \dots] | [m_{j_1}, \dots])\, x([m_{j_1}, m_{j_2}, \dots])$$

prob. of
$[m_j, \dots] \to [t_i, \dots]$

Unfolding algorithm in high-dimension

Example:





- Response mx of Pythia vs measured.
- Cross correlations $p_T, k_{t,g}, \tau_2$!

- Simultaneous unfolding of 3 observables.
- Unfolding improves from 1d → 3d.

Unfolding algorithm in high-dimension

Same inference procedure:

$$x([t_{i_1}, t_{i_2}, \ldots]) = \sum_j p([t_{i_1}, \ldots] | [m_{j_1}, \ldots]) \, x([m_{j_1}, m_{j_2}, \ldots])$$

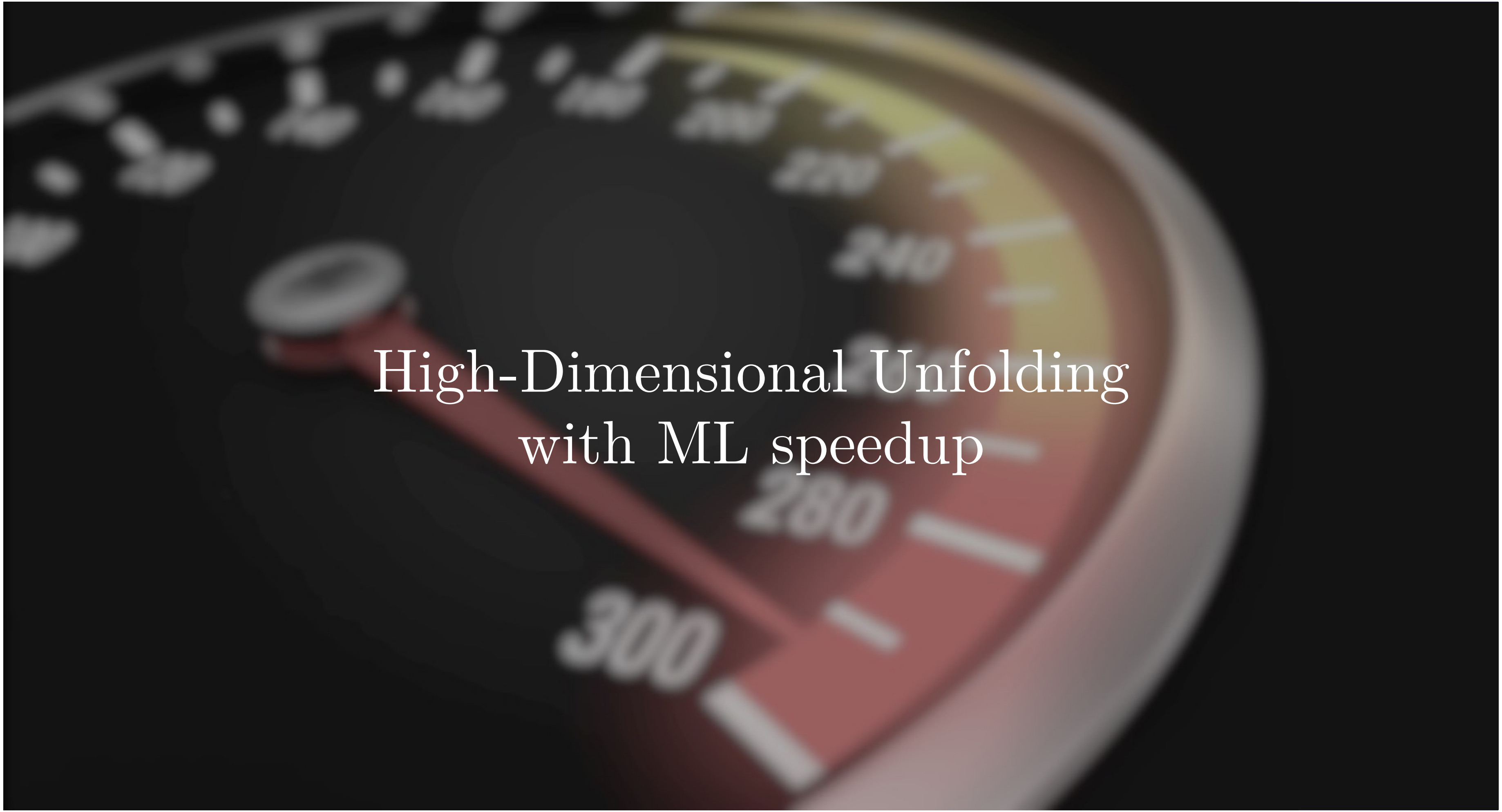
prob. of
$[m_j, \ldots] \to [t_i, \ldots]$

Going to higher dimensions?

1. Statistical uncertainties?
2. Convergence?
3. Computation power? 1M events, 8 obs, 10 bins:

1 PiB = 1000 TB!

```
numpy.core._exceptions.MemoryError: Unable to allocate 326. PiB for
an array with shape (45949729863572161,) and data type float64
```



High-Dimensional Unfolding with ML speedup

An algorithm for high-dimensions

[OmniFold:Andreassen,Komiske,Metodiev, Nachman,Thaler,1911.09107, 2105.04448]

- Rewrite with likelihood ratios:

$$p_n(t_i) = \sum_j \frac{p(m_j|t_i)p_{n-1}(t_i)}{p_{n-1}(m_j)} p(m_j) \longrightarrow \underbrace{\frac{p_n(t_i)}{p_{n-1}(t_i)}}_{\text{probability ratios}} = \sum_j p(m_j|t_i) \underbrace{\frac{p(m_j)}{p_{n-1}(m_j)}}_{\text{probability ratios}}$$

- Likelihood ratios = optimum problem: $\text{argmin}_{c(m)}(L(c(m))) \leftrightarrow \frac{p(m)}{p_{n-1}(m)}$

$$L[c(m)] = -\int dm\ [p(m)\log(c(m)) + p_{n-1}(m)\log(1-c(m))]$$

- Using events: $p(m) = \sum_i \delta(m - m_i)$

$$L[c(m)] = -\sum_i \delta_{i=meas} \log(c(m_i)) + \delta_{i=sim_{n-1}} \log(1-c(m_i))$$

- Optimum problem ≡ training a NN classifier (OmniFold)

OmniFold-HI: efficiency, bkg counts, Trash, and Fake events, stat and sys uncertainties.

An algorithm for high-dimensions

[OmniFold:Andreassen,Komiske,Metodiev, Nachman,Thaler,1911.09107, 2105.04448]

- Rewrite with likelihood ratios:

$$p_n(t_i) = \sum_j \frac{p(m_j|t_i)p_{n-1}(t_i)}{p_{n-1}(m_j)} p(m_j) \longrightarrow \frac{p_n(t_i)}{p_{n-1}(t_i)} = \sum_j p(m_j|t_i) \frac{p(m_j)}{p_{n-1}(m_j)}$$

probability ratios

- Likelihood ratios = optimum problem: $\mathrm{argmin}_{c(m)}\big(L(c(m))\big) \leftrightarrow \frac{p(m)}{p_{n-1}(m)}$

$$L[c(m)] = -\int dm \, [p(m)\log(c(m)) + p_{n-1}(m)\log(1-c(m))]$$

- Using events: $p(m) = \sum_i \delta(m - m_i)$

$$L[c(m)] = -\sum_i \delta_{i=meas}\log(c(m_i)) + \delta_{i=sim_{n-1}}\log(1-c(m_i))$$

- Optimum problem ≡ training a NN classifier (OmniFold)

OmniFold-HI: efficiency, bkg counts, Trash, and Fake events, stat and sys uncertainties.

An algorithm for high-dimensions

[OmniFold:Andreassen,Komiske,Metodiev, Nachman,Thaler,1911.09107, 2105.04448]

- Rewrite with likelihood ratios:

$$p_n(t_i) = \sum_j \frac{p(m_j|t_i)p_{n-1}(t_i)}{p_{n-1}(m_j)} p(m_j) \longrightarrow \frac{p_n(t_i)}{p_{n-1}(t_i)} = \sum_j p(m_j|t_i) \frac{p(m_j)}{p_{n-1}(m_j)}$$

probability ratios

- Likelihood ratios = optimum problem: $\mathrm{argmin}_{c(m)}\big(L(c(m))\big) \leftrightarrow \frac{p(m)}{p_{n-1}(m)}$

$$L[c(m)] = -\int dm\ [p(m)\log(c(m)) + p_{n-1}(m)\log(1-c(m))]$$

- Using events: $p(m) = \sum_i \delta(m - m_i)$

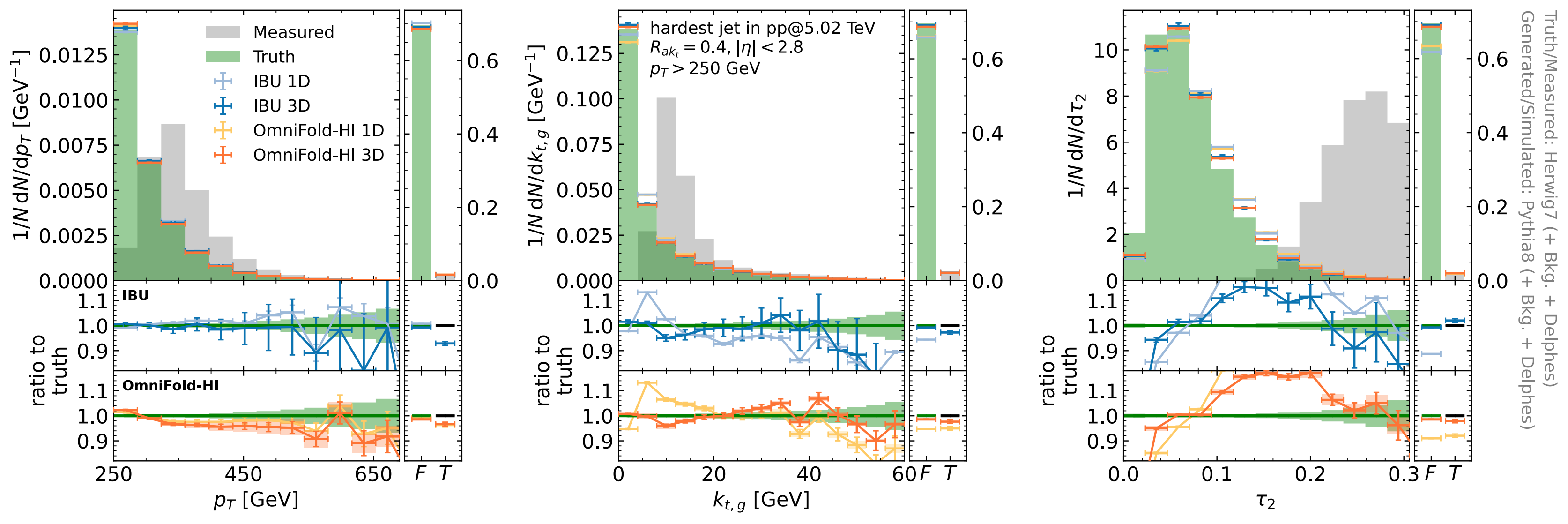
$$L[c(m)] = -\sum_i \delta_{i=meas}\log(c(m_i)) + \delta_{i=sim_{n-1}}\log(1-c(m_i))$$

- Optimum problem ≡ training a NN classifier (OmniFold)

OmniFold-HI: efficiency, bkg counts, Trash, and Fake events, stat and sys uncertainties.

An algorithm for high-dimensions

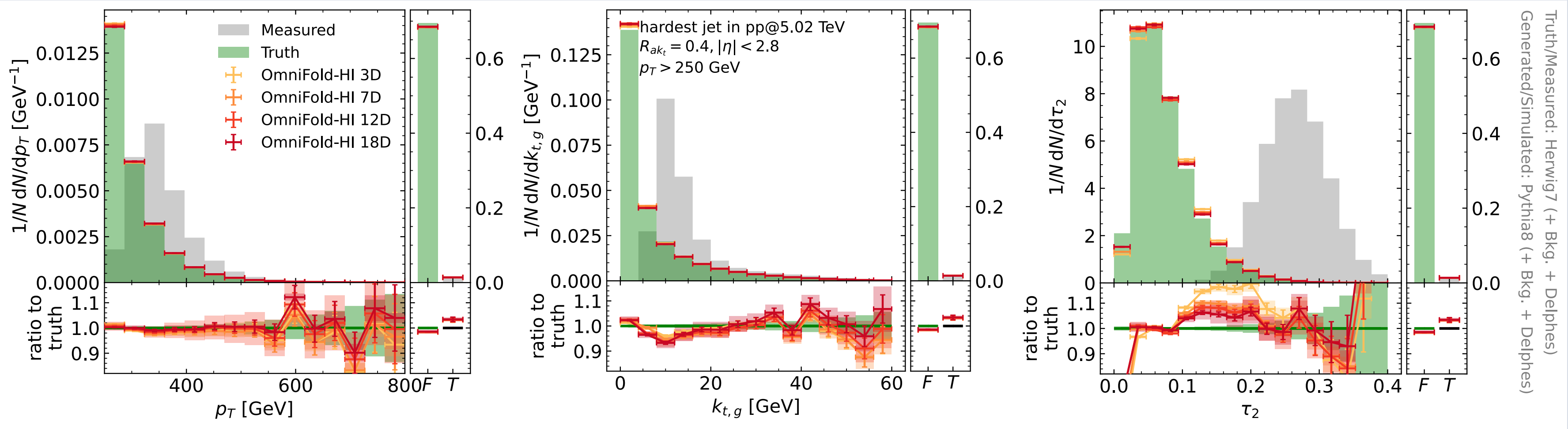
Example:



- IBU and OmniFold-HI agrees in 1d and 3d!
- Understanding stat. and sys uncertainties for NN!

An algorithm for high-dimensions

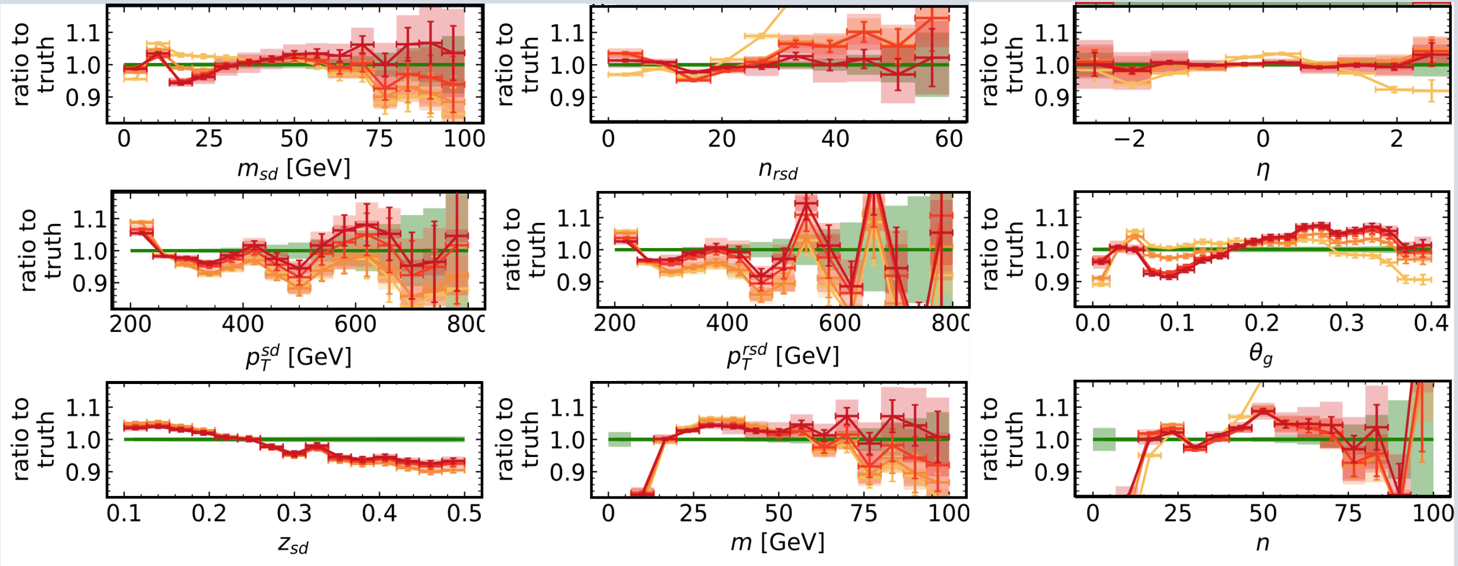
Example:



- Unfolding in higher dimensions.
- Going to 18d improves performance!
- Improvement in difficult observables: m_{jet}, n_{jet}

An algorithm for high-dimensions

Example:



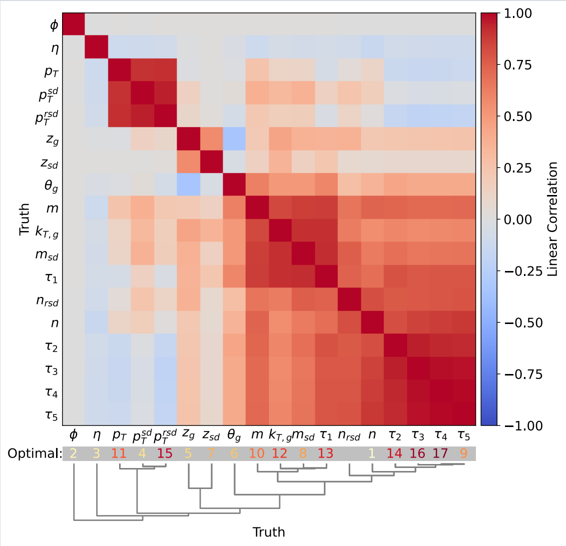
- Unfolding in higher dimensions.
- Going to 18d improves performance!
- Improvement in difficult observables: m_{jet}, n_{jet}

Choosing Auxiliary Observables

How to choose observables?

Auxiliary observables

(selection based on correlation distances)



Integrate to calibration

Jet calibration:

1. Correct the jet 4-momentum:

$$\boxed{p_{jet}^{\mu,reco}, \text{many event obs.}} \rightarrow \boxed{p_{jet}^{\mu,true}}$$

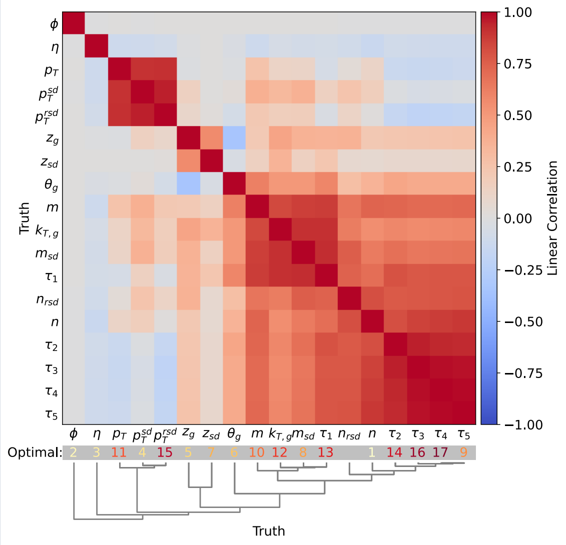
2. Unfold the rest.

Calibration is an inference task!
Combine with unfolding.

How to choose observables?

Auxiliary observables

(selection based on correlation distances)



Integrate to calibration

Jet calibration:

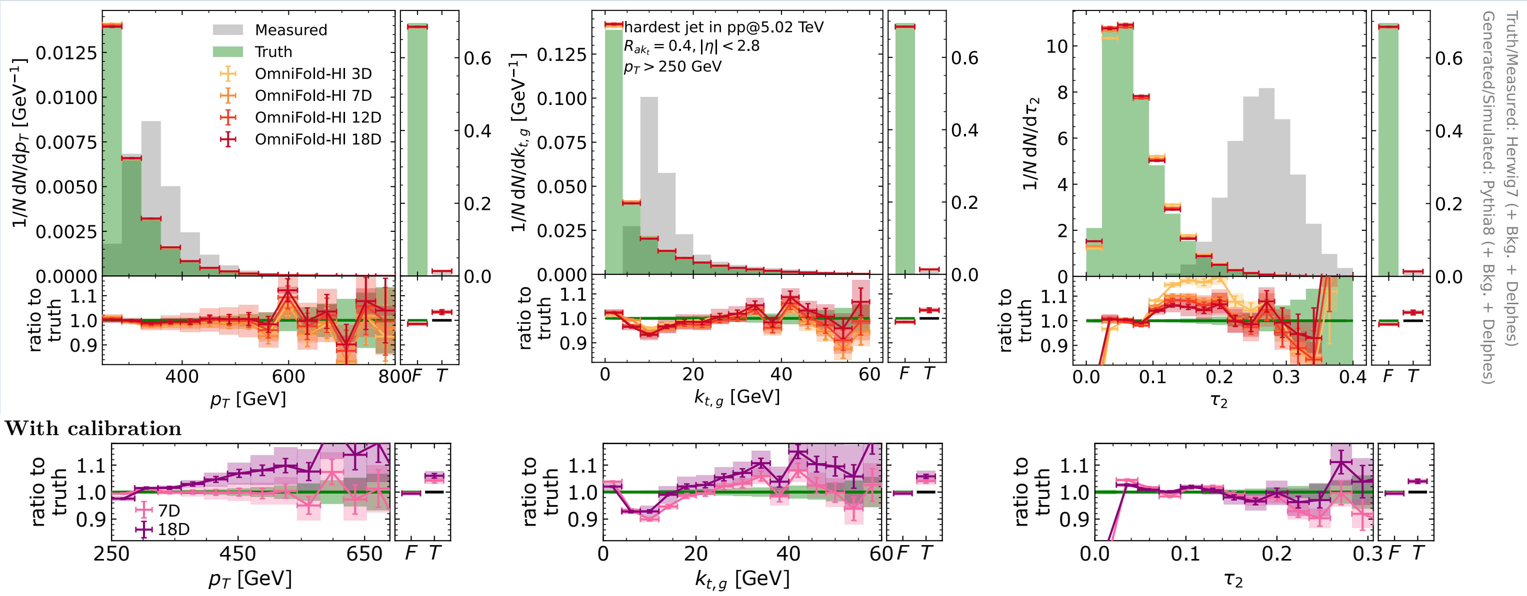
1. Correct the jet 4-momentum:

$$\boxed{p_{jet}^{\mu,reco}\text{, many event obs.}} \rightarrow \boxed{p_{jet}^{\mu,true}}$$

2. Unfold the rest.

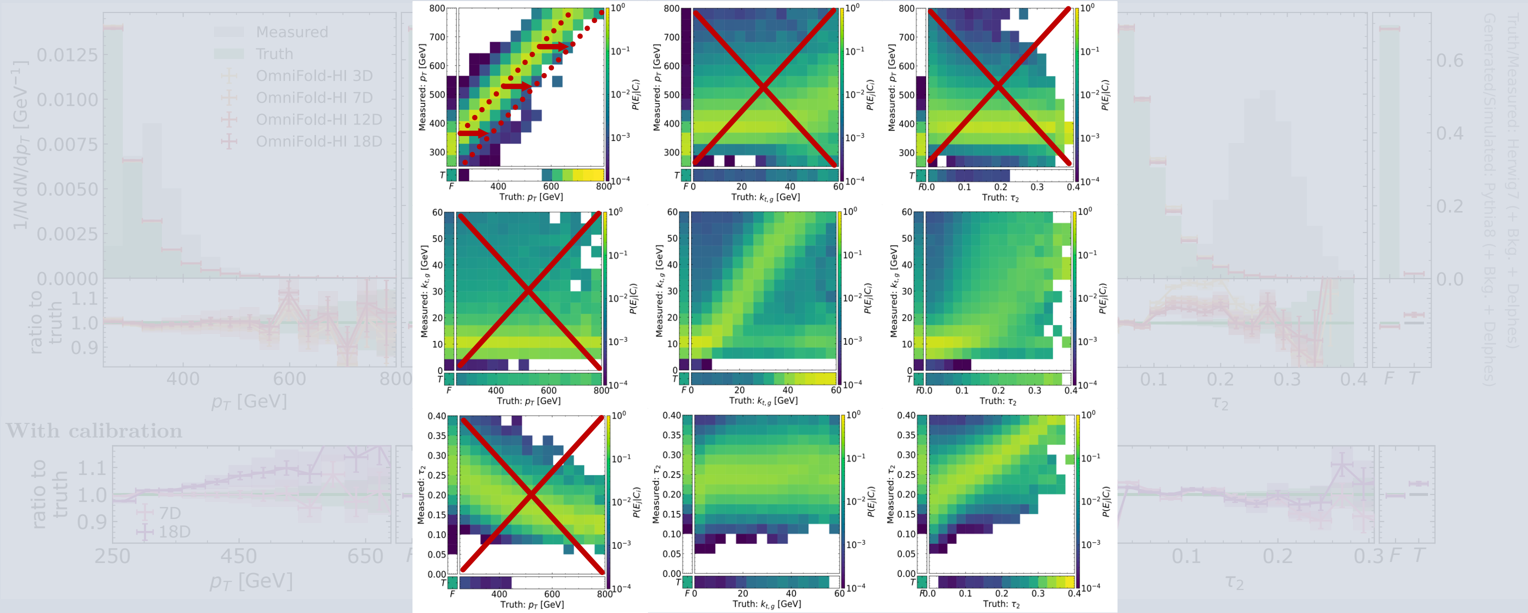
Calibration is an inference task!
Combine with unfolding.

How to choose observables?



- Calibration worsens the performance.

How to choose observables?



- Calibration worsens the performance in high dims.

Summary: unfolding in high-dimensions

1. Introduction to unfolding (likelihood-free inference)

2. OmniFold to unfold in high dimensions

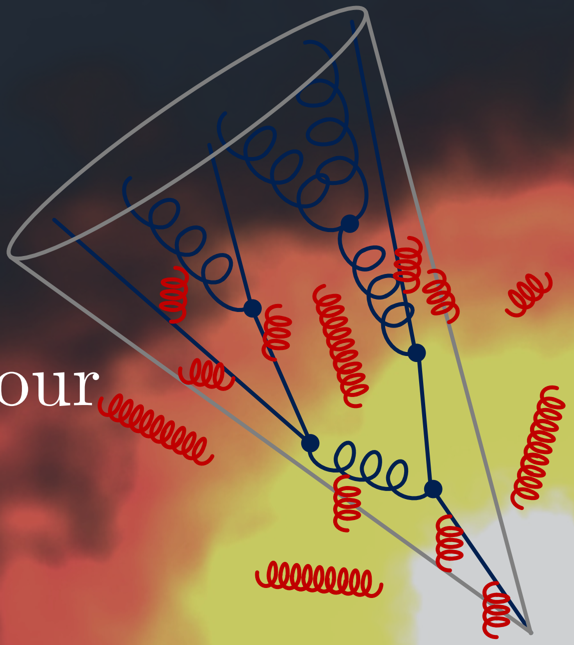
   Better performance is already in use in pp: [OmniFold 1911.09107, 2105.04448]
   [H1 2108.12376,2303.13620]
   [LHCb 2208.11691]
   [STAR 2307.07718, 2403.13921]
   [ATLAS 2405.20041, 2502.02062]
   [CMS 2505.17850]

   OmniFold-HI improvements: derivation, bkg, efficiency, uncertainties → ready for HI pheno!
   [2507.06291, https://github.com/OmniFoldHI]

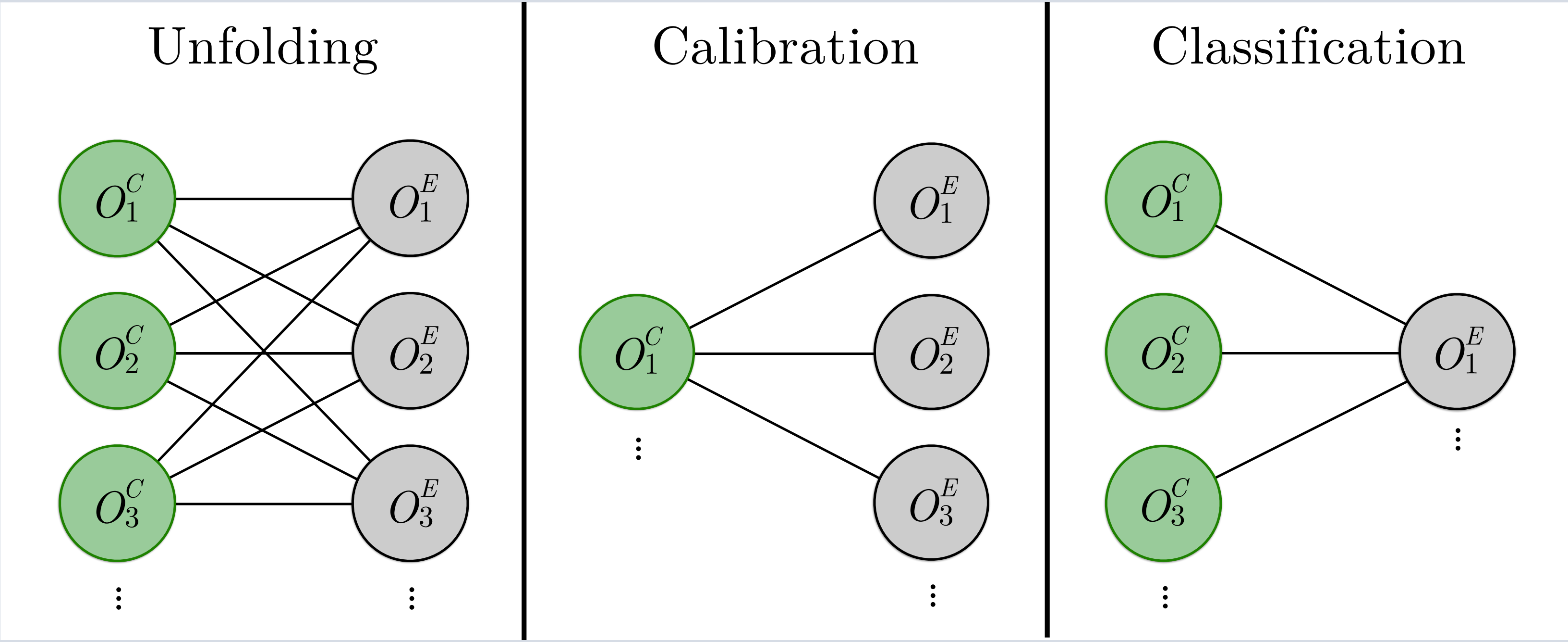
3. Strategy to choose auxiliary observables

4. Improved workflow by unifying: calibration + bkg subtraction + unfolding.

Thank you for your attention!

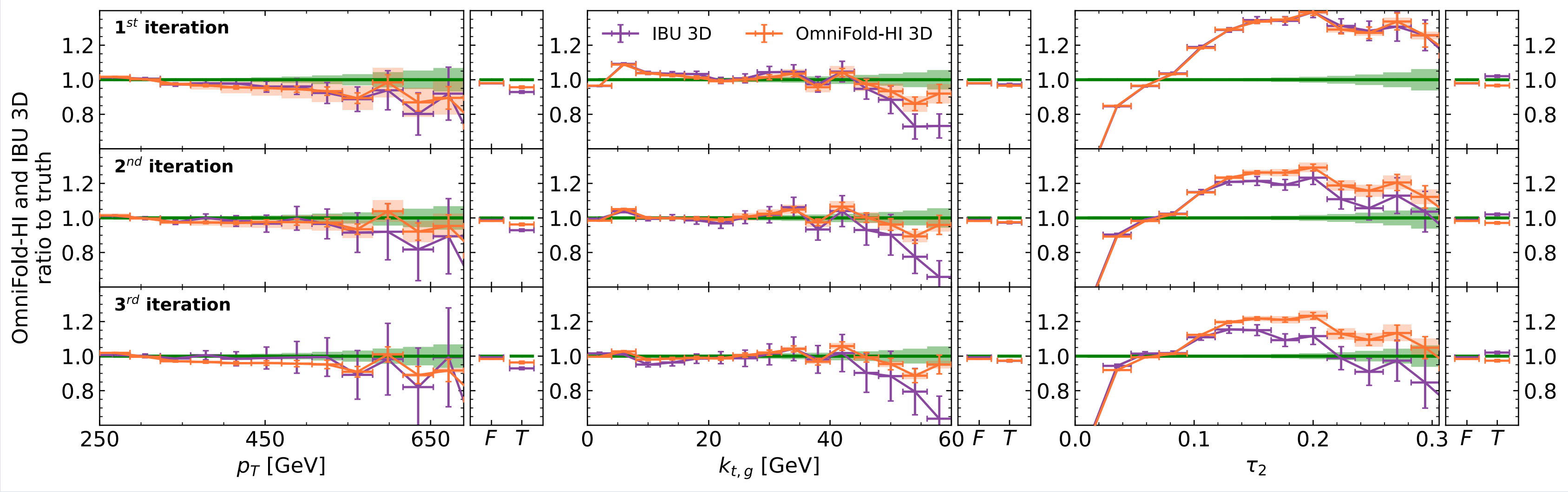


Unfolding as a general inference



IBU vs OmniFold

Example:



- Unfolding in 3d.
- IBU and OmniFold-HI agrees!
- Understanding stat. and sys uncertainties