RUTGERS

AI+HEP in East Asia workshop
21.01.2026

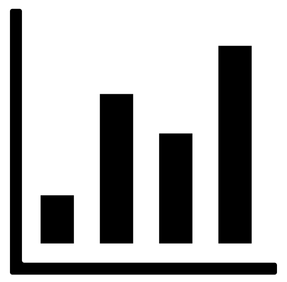
Status on Generative Unfolding

Sofia Palacios Schweitzer

How to unfold top decays, Favaro et al., arXiv:2501.12363
Analysis-ready Generative Unfolding, Butter et al., arXiv:2509.02708
Generative Unfolding of Jets and their Substructure, Petitjean et al., arXiv:2510.19906

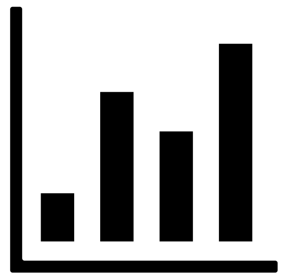
The Dream of Unfolding

What we have



Detector-level measurement

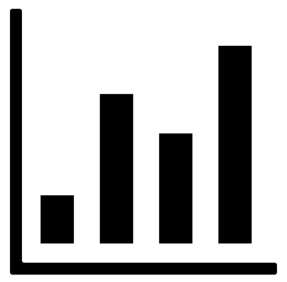
What we want



Particle-level distribution

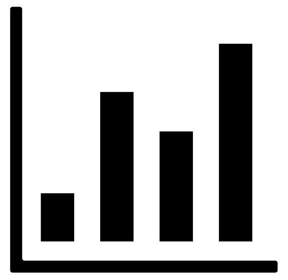
The Dream of Unfolding

What we have



Detector-level measurement

What we want



Particle-level distribution

Why?

Interesting physics at particle/parton-level

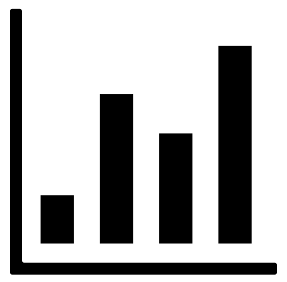
Efficiently testing new parameters

Direct access to theory parameters

…

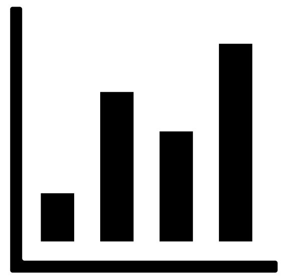
The Dream of Unfolding

What we have



Detector-level measurement

What we want



Particle-level distribution

How?

The Dream of Unfolding?

Target

$$p(x) = \int p(y)\, p(x\,|\,y)\, dy$$

Measurement

Forward model

Quest:

Find "the" **particle-level distribution** that reproduces the **detector-level measurement** under the given **forward model**.

The Dream of Unfolding?

Limited statistics

$$\boxed{p(x)} = \int \boxed{p(y)} \; \boxed{p(x\,|\,y)} \; dy$$

Measurement

Forward model

Loss of information

Quest:

Find “the” **particle-level distribution** that reproduces the **detector-level measurement** under the given **forward model**.

The Dream of Unfolding?

$$p(x) = \int \boxed{p_a(y)}\, p(x\,|\,y)\; dy = \int \boxed{p_b(y)}\, p(x\,|\,y)\; dy$$

Challenge:

Different particle-level distributions can give rise to the same detector-level distribution

Every particle-level distribution that can reproduce the detector-level measurement is a valid solution

The Dream of Unfolding?

$$p(x) = \int \boxed{p_a(y)}\, p(x\,|\,y)\, dy = \int \boxed{p_b(y)}\, p(x\,|\,y)\, dy$$

Challenge:

Different particle-level distributions can give rise to the same detector-level distribution

Every particle-level distribution that can reproduce the detector-level measurement is a valid solution

→ **introduce bias to physical solutions**

Unfolding — Two Avenues

$$
\begin{tikzcd}
p_{MC}(y) \arrow[d, "{p_d(x|y) = p_{MC}(x|y) \equiv p(x|y)}"'] & p_{\text{unfold}}(y) \\
p_{MC}(x) \arrow[r, leftrightarrow] & p_d(x) \arrow[u, "?"']
\end{tikzcd}
$$

Unfolding — Two Avenues

$$\begin{tikzcd}
p_{MC}(y) \arrow[d, "p_d(x \mid y) = p_{MC}(x \mid y) \equiv p(x \mid y)"'] & & p_{\text{unfold}}(y) \\
p_{MC}(x) \arrow[rr, leftrightarrow] & & p_d(x) \arrow[u, "?"']
\end{tikzcd}$$

Numerical inversion “$p(x \mid y)^{-1}$”

$$p_{unfold}(y) = \arg\min_{p(y)} \left[p_d(x) - \int p(y)\, p(x \mid y)\, dy \right]$$

Large variance, unless regularised

Matrix Unfolding, TUnfold, SpinUp

Unfolding — Two Avenues

$$\begin{tikzcd}
p_{MC}(y) \arrow[d, "p_d(x\,|\,y) = p_{MC}(x\,|\,y) \equiv p(x\,|\,y)"'] & & p_{\text{unfold}}(y) \\
p_{MC}(x) \arrow[rr, leftrightarrow] & & p_d(x) \arrow[u, "?"']
\end{tikzcd}$$

Numerical inversion "$p(x\,|\,y)^{-1}$"

$$p_{unfold}(y) = \arg\min_{p(y)} \left[p_d(x) - \int p(y)\, p(x\,|\,y)\, dy \right]$$

Large variance, unless regularised

Matrix Unfolding, TUnfold, SpinUp

Probabilistic inversion $p(y\,|\,x)$

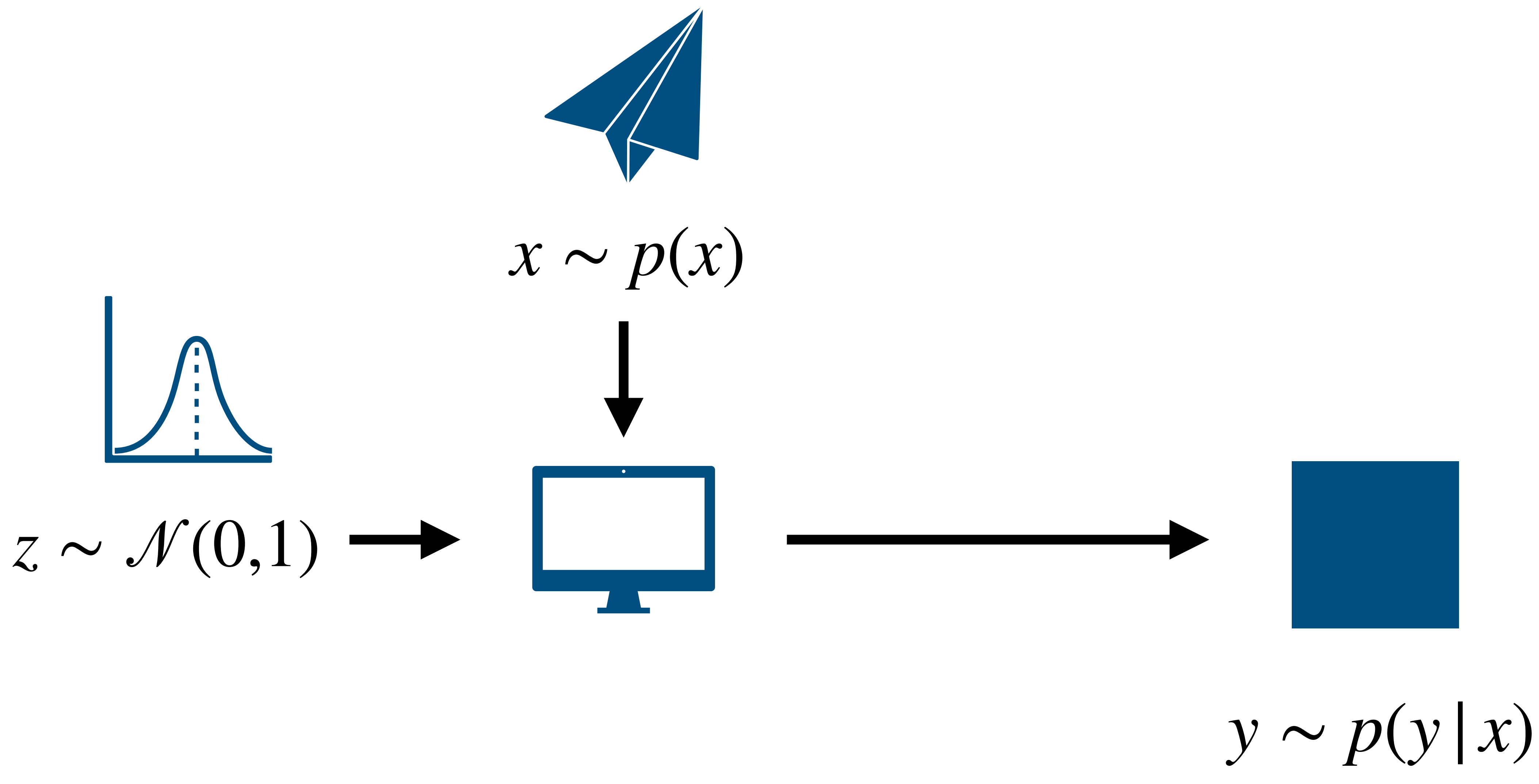
$$p_{unfold}(y) = \int p_d(x)\, p(y\,|\,x)\, dx$$

Only well-defined for a given prior $p(y)$

IBU, Omnifold, Generative Unfolding

Generative Unfolding

$$p_{unfold}(y) = \int p_d(x)\, p(y\,|\,x)\, dx$$



Learn to sample from conditional probability $p(y\,|\,x)$ with generative network

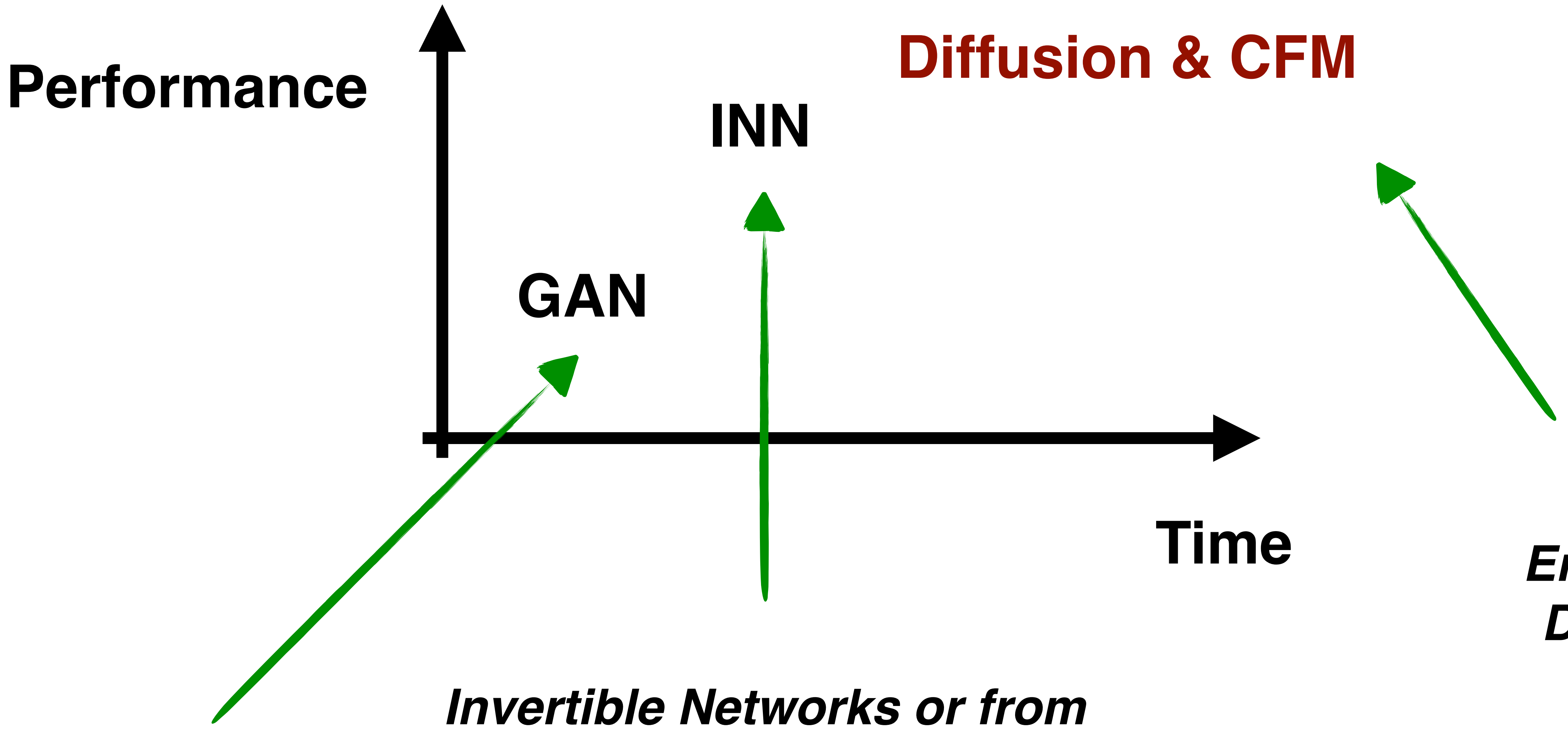
Training data:

$x, y \sim p_{MC}(x, y)$

→ per definition $p(y\,|\,x) \propto p_{MC}(y)$

Generative Unfolding

Which generative network?



How to GAN away Detector Effects

Bellagente et al. arXiv:1912.00477

Invertible Networks or from Partons to Detector and back again

Bellagente et al. arXiv:2006.06685

An unfolding method based on Conditional Invertible Neural Networks (cINN) using iterative training

Backes et al. arXiv:2212.08674

End-To-End Latent Variational Diffusion Models for Inverse Problems in High Energy Physics

Shmakov et al. arXiv:2305.10399

The landscape of unfolding with machine learning

Huetsch et al. arXiv:2404.18807

…

Updating your prior

$$p_{MC}(y) \xrightarrow{\text{update}} p_{\text{unfold}}(y)$$

$$p_{MC}(x, y) \xrightarrow{\text{update}} ?$$

Classifier-based: $w(y) = \dfrac{p_{\text{unfold}}(y)}{p_{MC}(y)}$

Generator-based: $p_{MC}(x) \xrightarrow{\text{update}} \int dy \, \underbrace{p(x \,|\, y)}_{\text{Generative network}} \, p_{\text{unfold}}(y)$

Updating your prior

Iterations

$$
\begin{tikzcd}
p^{(i)}_{MC}(y) \arrow[d, "p(x\,|\,y)"'] & & p^{(i)}_{\text{unfold}}(y) \arrow[ll] \\
p^{(i)}_{MC}(x) \arrow[rr, leftrightarrow] & & p_d(x) \arrow[u, "p^{(i)}(y\,|\,x) \,\propto\, p^{(i)}_{MC}(y)"']
\end{tikzcd}
$$

Application to Z + jets

What we have

Detector-level measurement
(6d jet observables)

Prior-dependence
(two different Pythia simulations for data and MC)

What we want

Particle-level distribution
(6d jet observables)

Prior-independence

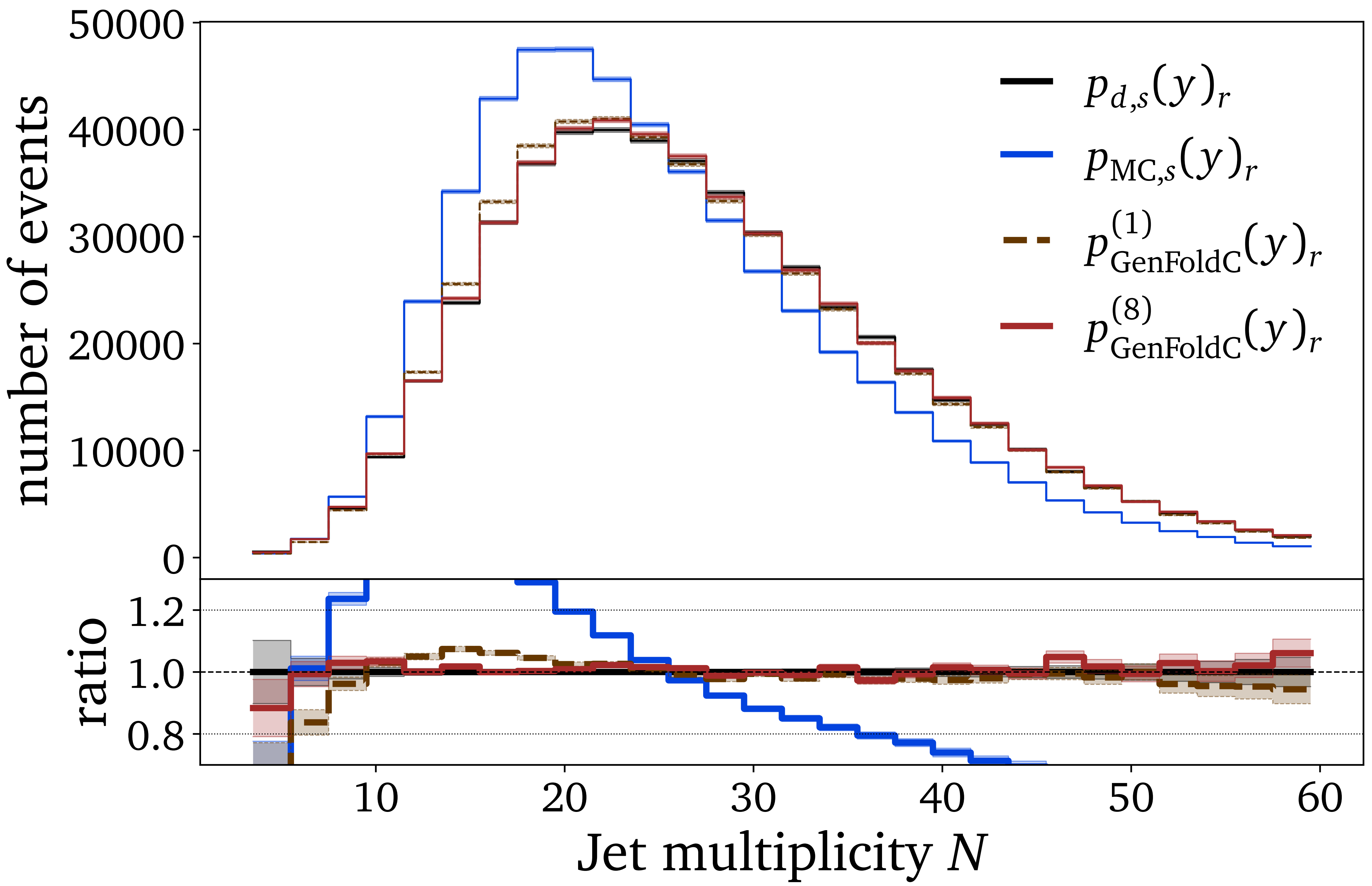
Application to Z + jets

What we have

Detector-level measurement
(6d jet observables)

Prior-dependence
(two different Pythia simulations for data and MC)

What we want



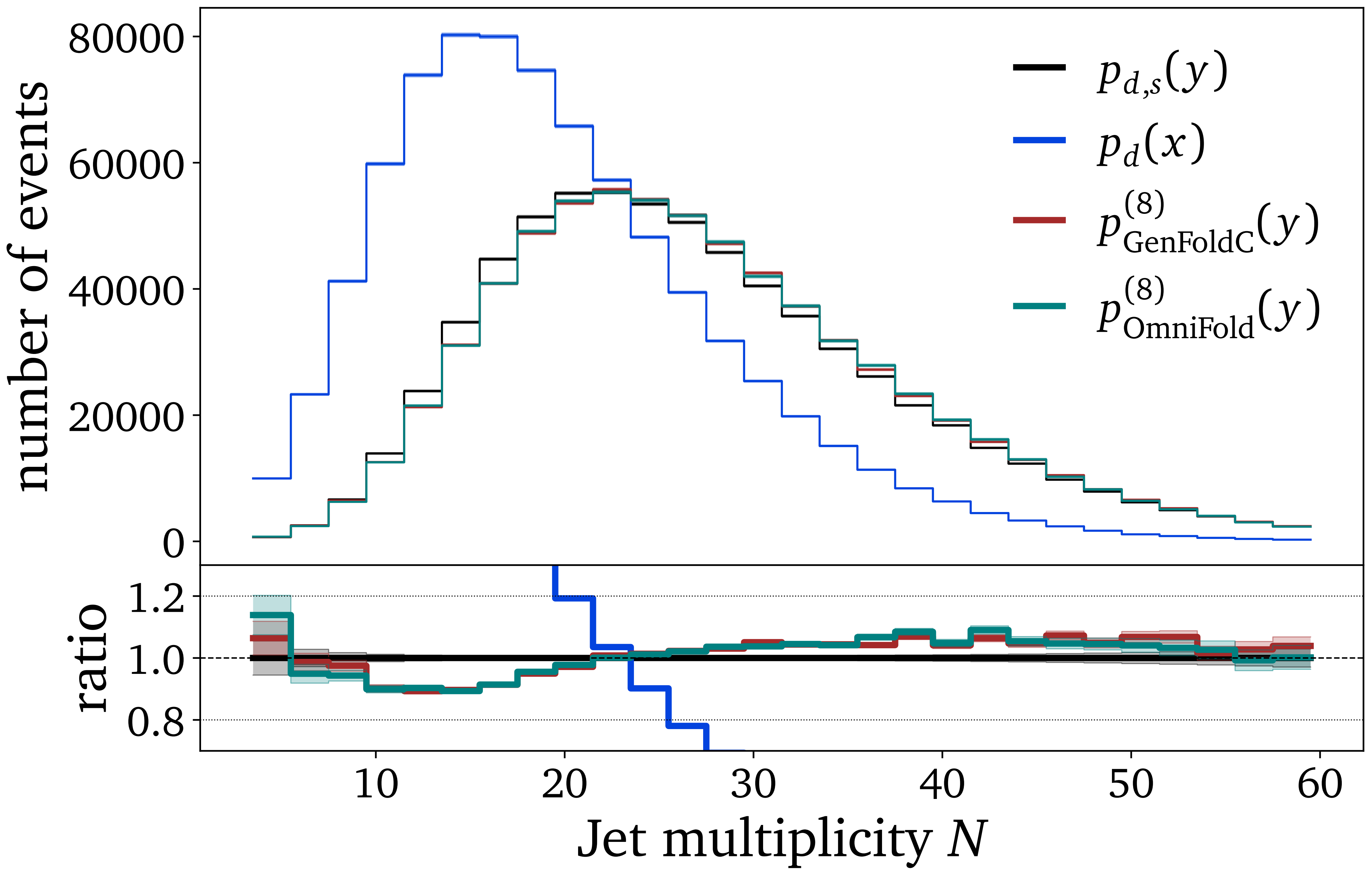
Application to Z + jets

What we have

Detector-level measurement
(6d jet observables)

Prior-dependence
(Herwig and Pythia for data and MC)

What we want?



Application to Z + jets

Main assumption (for any unfolding algorithm):

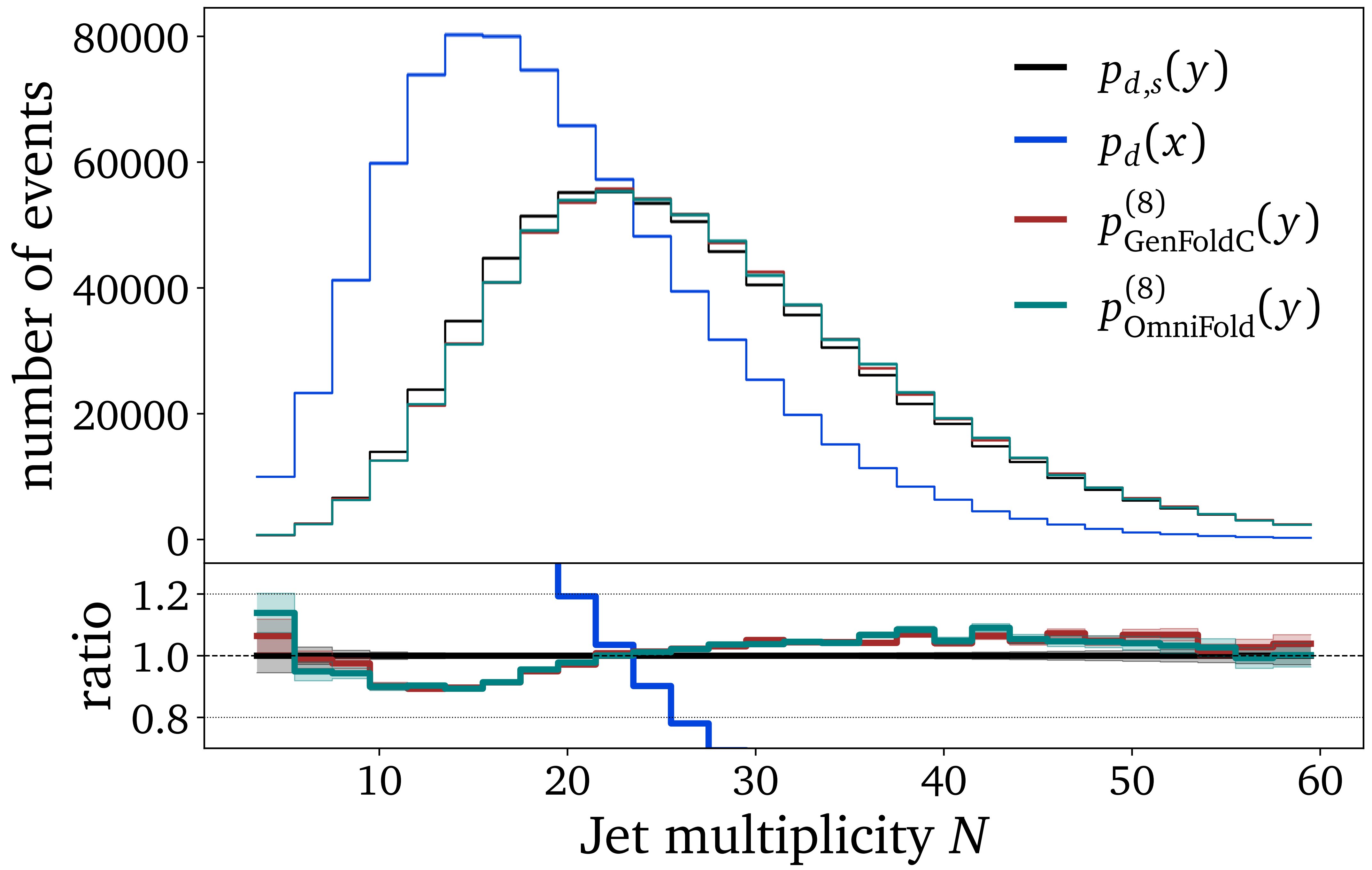
$$p_d(x\,|\,y) = p_{MC}(x\,|\,y) = p(x\,|\,y)$$

What about summary statistics, like jets?

$$p(x\,|\,J) = \int dy\; p(x\,|\,y)p(y\,|\,J)$$

Generally, $p_d(y\,|\,J) \neq p_{MC}(y\,|\,J)$

What we want?



Option 1: Treat non-closure as systemic uncertainty

ATLAS arXiv:2405.20041

Option 2: Scale things up?

Unfolding ~$\mathcal{O}(100)$ dimensional phase spaces

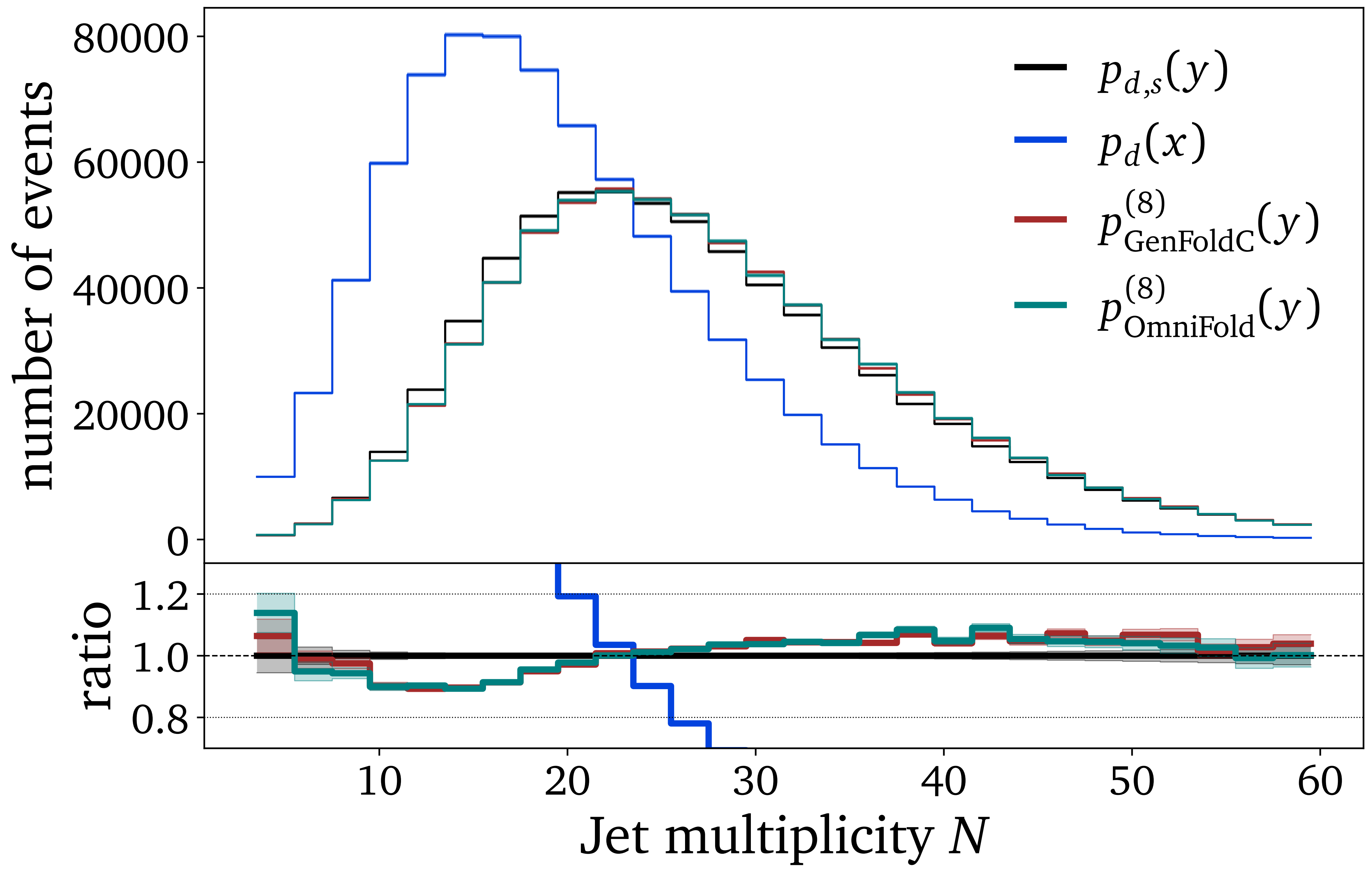
=

Training a generative network to reproduce phase space of varying multiplicity to %-level precision

+

Conditioned on a second high-dimensional phase space distribution of varying multiplicity

What we want?



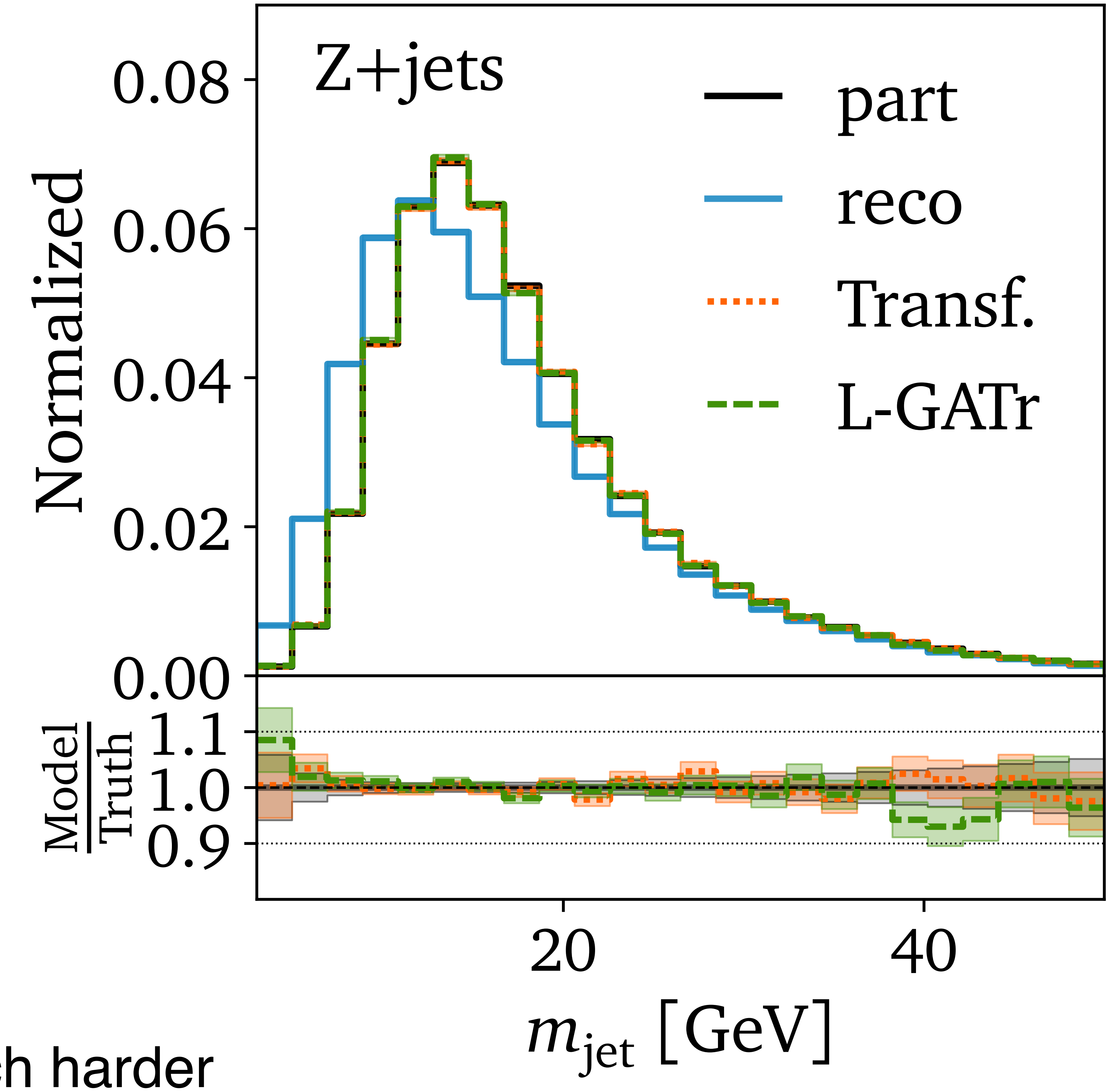
ML task becomes much harder

Option 2: Scale things up!

1. Step: Closure on MC-MC unfolding

Unfolding ~$\mathcal{O}(100)$ dimensional phase spaces

High dimensional correlations are unfolded to %-level



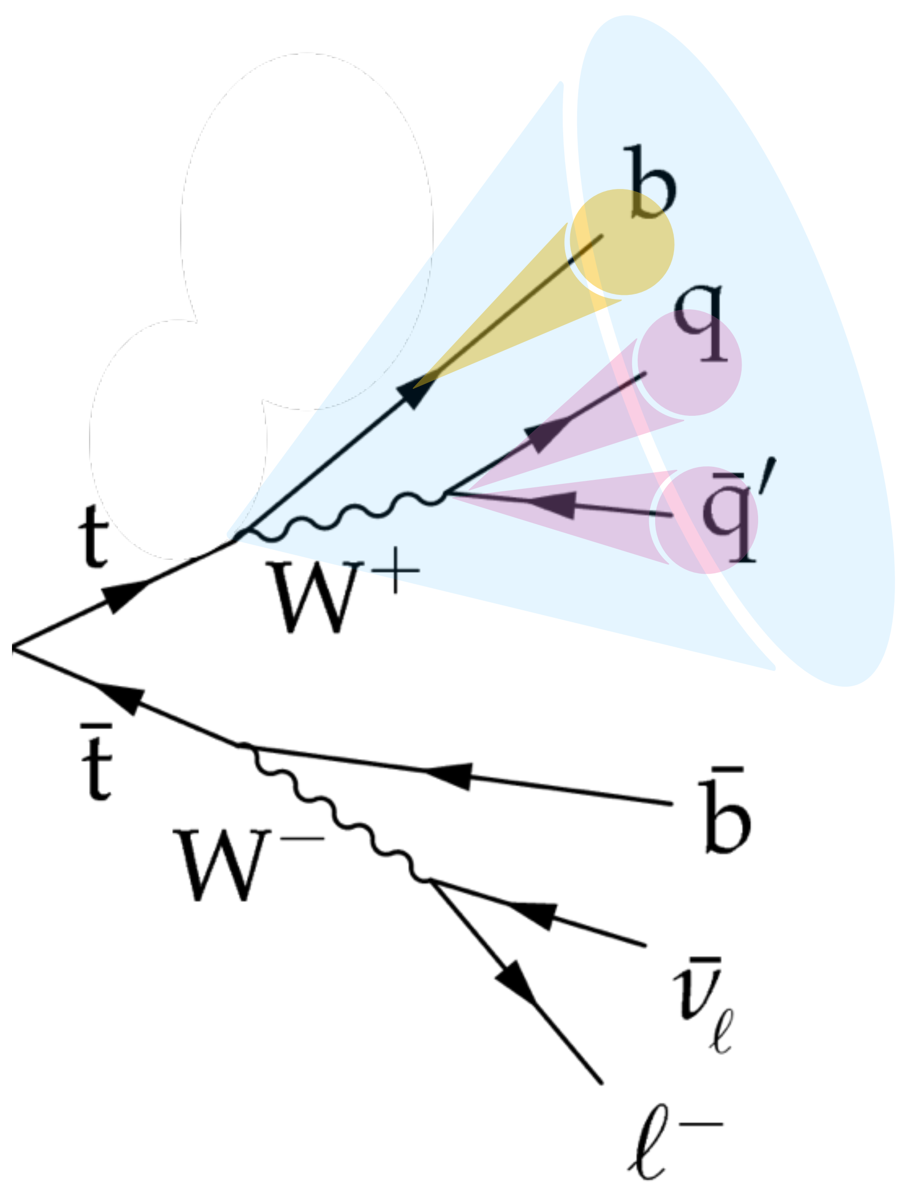
ML task becomes much harder

Option 2: Scale things up!

2. Step: Closure on data-MC

Boosted top decays

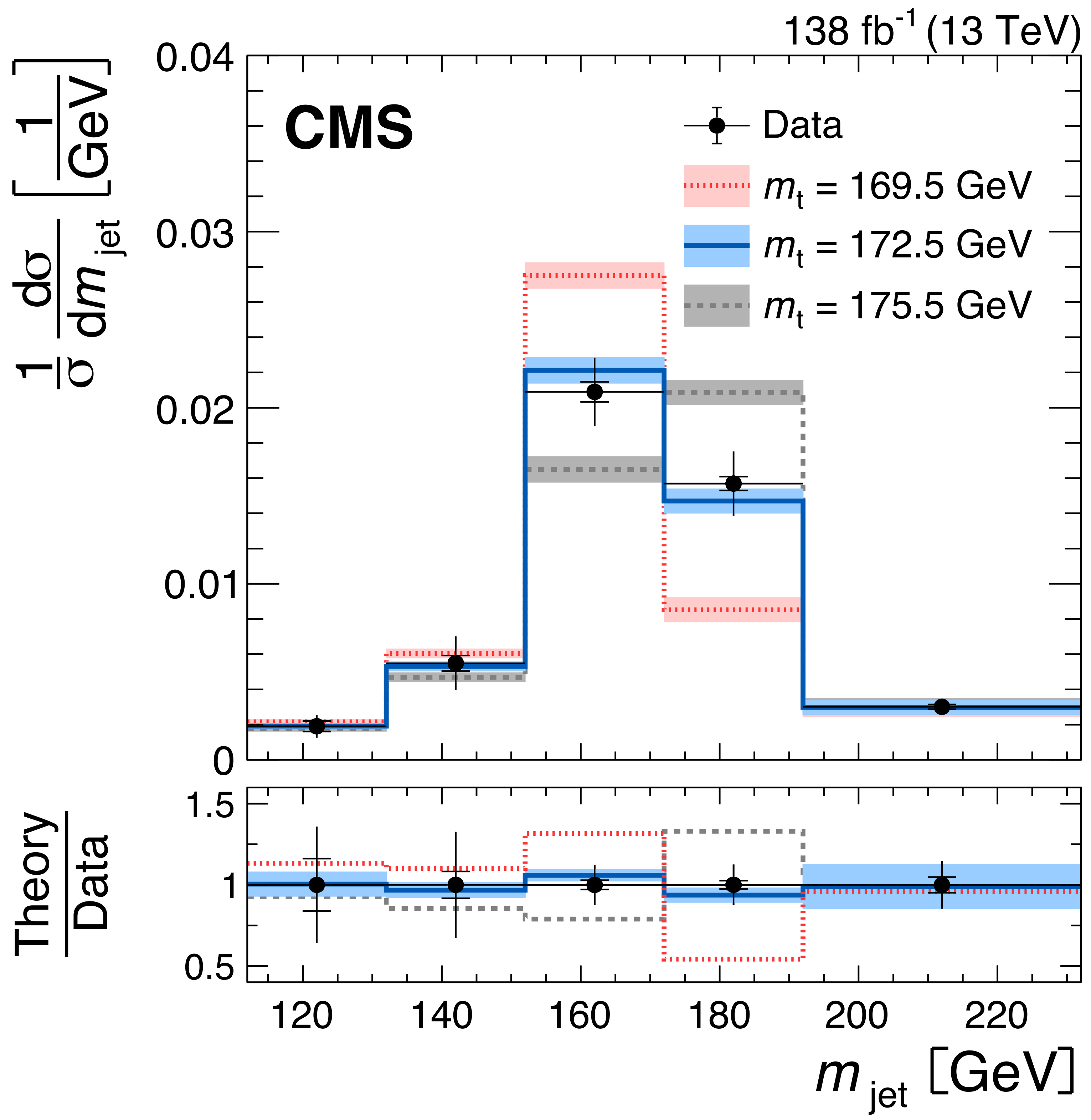
$p_{T,J} > 400$ GeV



Reconstruct triple jet mass M_{jjj} to measure m_t

Tag side

Previously done in CMS with TUnfold
(classical binned unfolding algorithm)



CMS **2211.01456**

BUT leading uncertainty: choice of m_t in simulation

→ **Could generative unfolding help?**

Application to Boosted Tops

What we have

Detector-level measurement
(3 jets 4-momenta)

Prior-dependence
(Different top masses for data and MC)

What we want

Particle-level distribution
(3 jets 4-momenta)

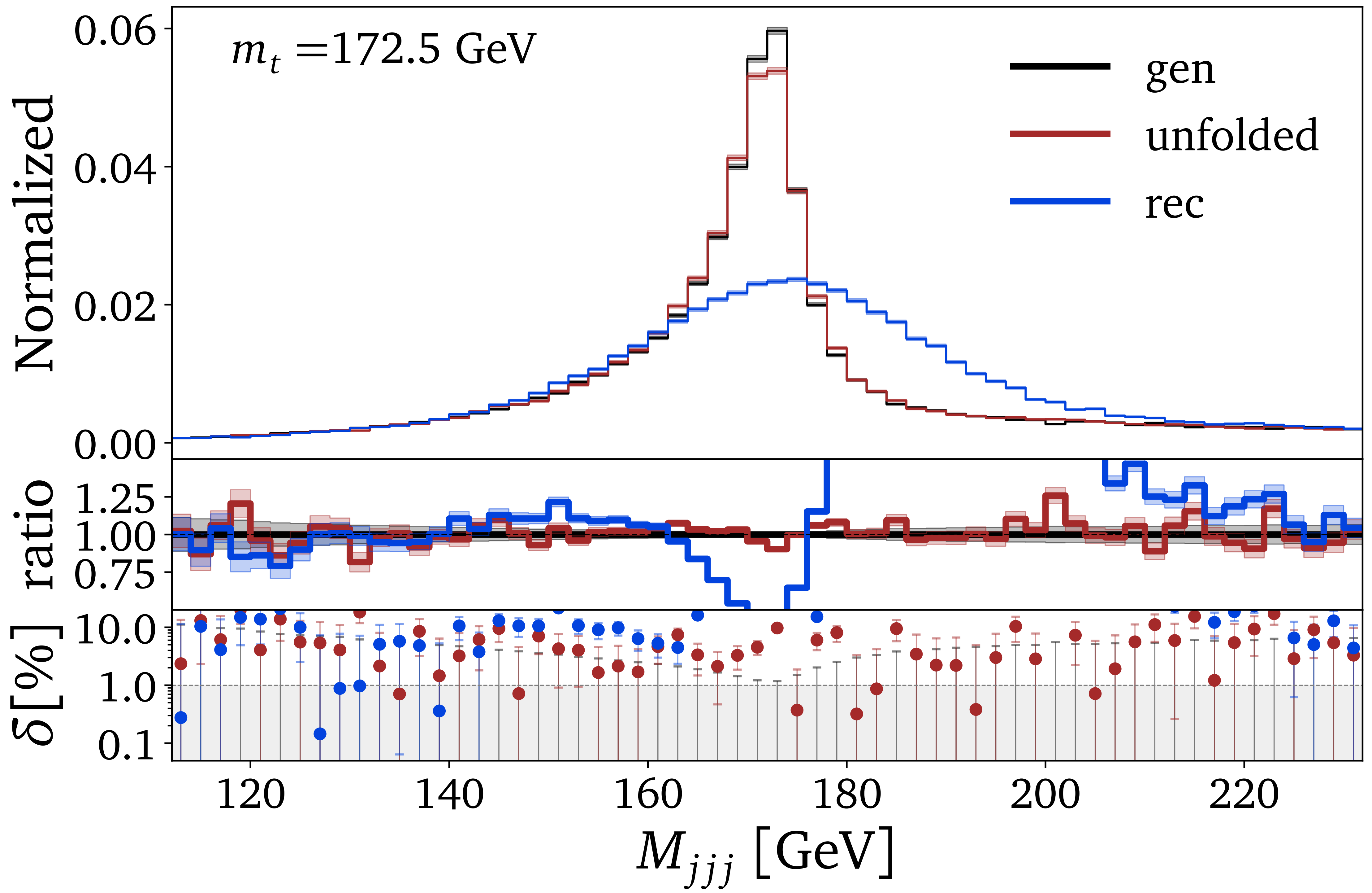
Prior-independent top mass measurement

Model-Dependence?

Train with full CMS simulation with
$m_t = 172.5$ GeV

Unfolded distribution of triple jet mass
within $\mathcal{O}(1\%)$ of truth particle-level

BUT: Test data also simulation with
$m_t = 172.5$ GeV

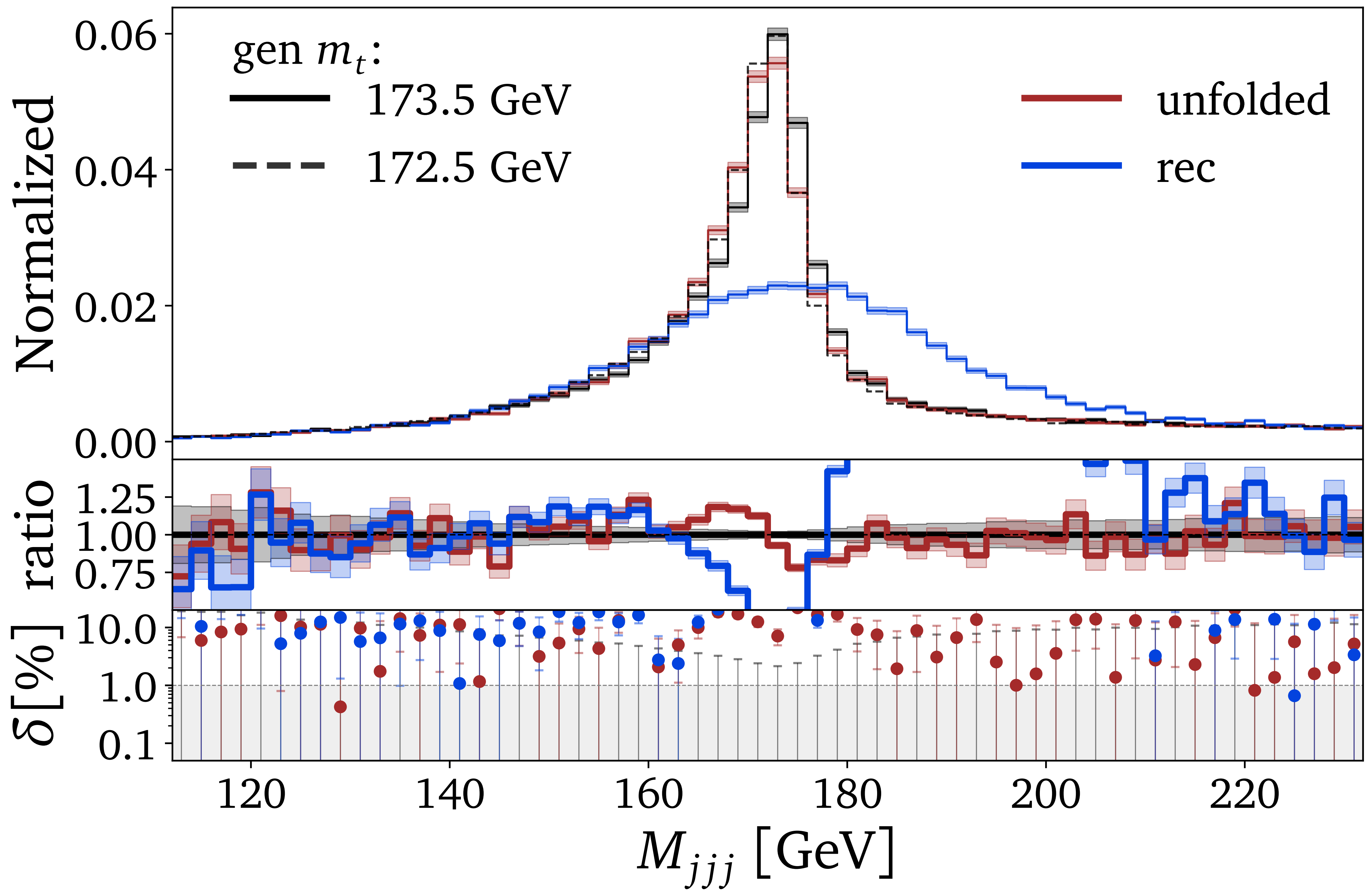


Model-Dependence!

Train with full CMS simulation with
$m_t = 172.5$ GeV

Unfolded distribution of triple jet mass
within $\mathcal{O}(1\%)$ of truth particle-level

BUT: Test data also simulation with
$m_t = 172.5$ GeV



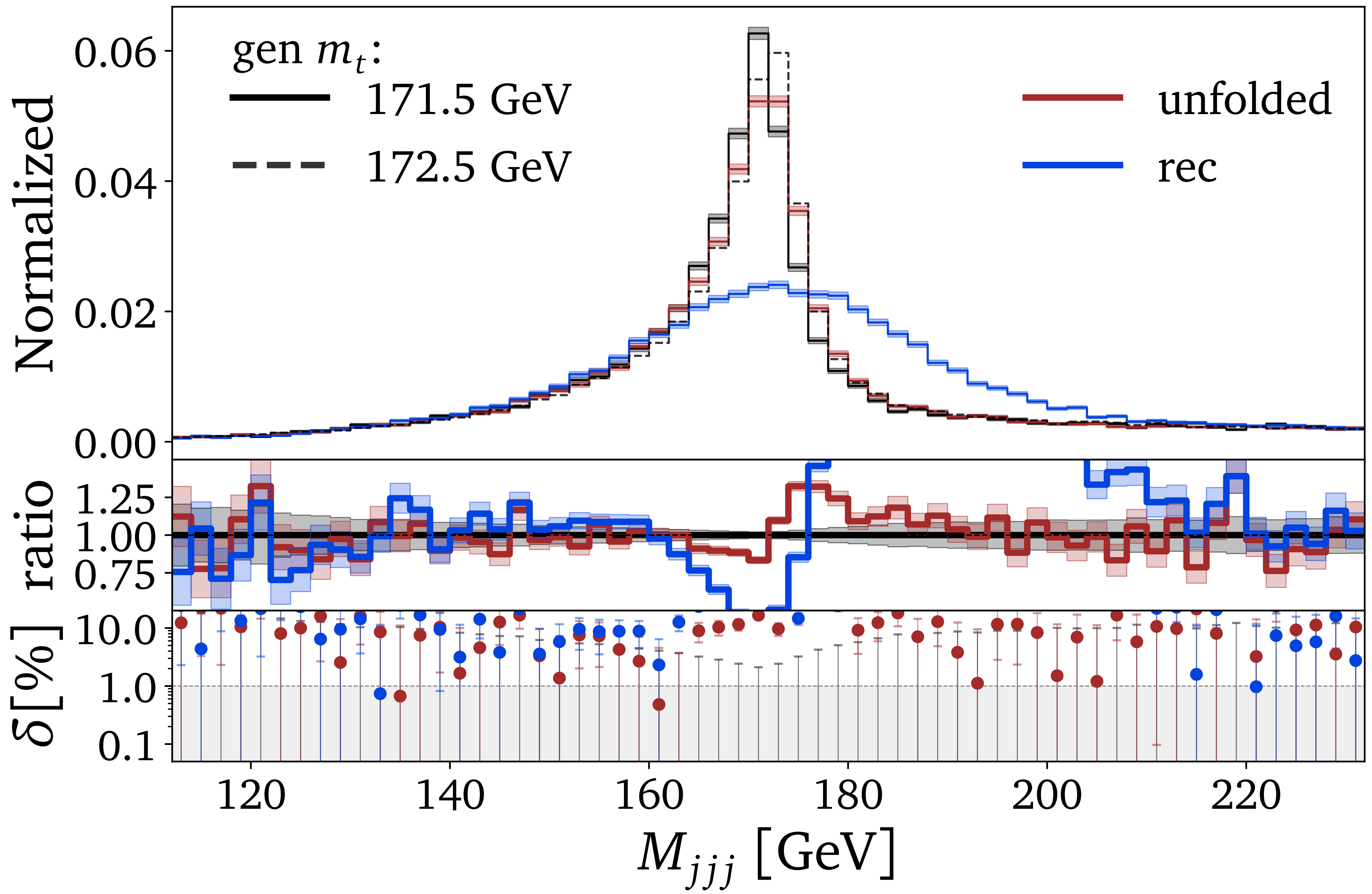
For pseudo-data with different top masses :
Algorithm falls back to prior ($m_t = 172.5$ GeV)

Model-Dependence!

Train with full CMS simulation with
$m_t = 172.5$ GeV

Unfolded distribution of triple jet mass
within $\mathcal{O}(1\%)$ of truth particle-level

BUT: Test data also simulation with
$m_t = 172.5$ GeV



For pseudo-data with different top masses :
Algorithm falls back to prior ($m_t = 172.5$ GeV)

Removing Model-Dependence

$$
\begin{tikzcd}
p_{MC}(y\,|\,m_s) \arrow[d, "p(x\,|\,y)"'] & & p_{\text{unfold}}(y\,|\,m_s, m_d) \\
p_{MC}(x\,|\,m_s) \arrow[rr, leftrightarrow] & & p_d(x\,|\,m_d) \arrow[u, "p_{model}(y\,|\,x, m_s)"]
\end{tikzcd}
$$

Removing Model-Dependence

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
p_{MC}(y \mid m_s) \arrow[d, "p(x \mid y)"'] & & p_{\text{unfold}}(y \mid m_s, \cancel{m_d}) \\
p_{MC}(x \mid m_s) \arrow[rr, leftrightarrow] & & p_d(x \mid m_d) \arrow[u, "p_{model}(y \mid x, m_s)"]
\end{tikzcd}
$$

→ **Solution: Strengthen m_d dependence, but how?**

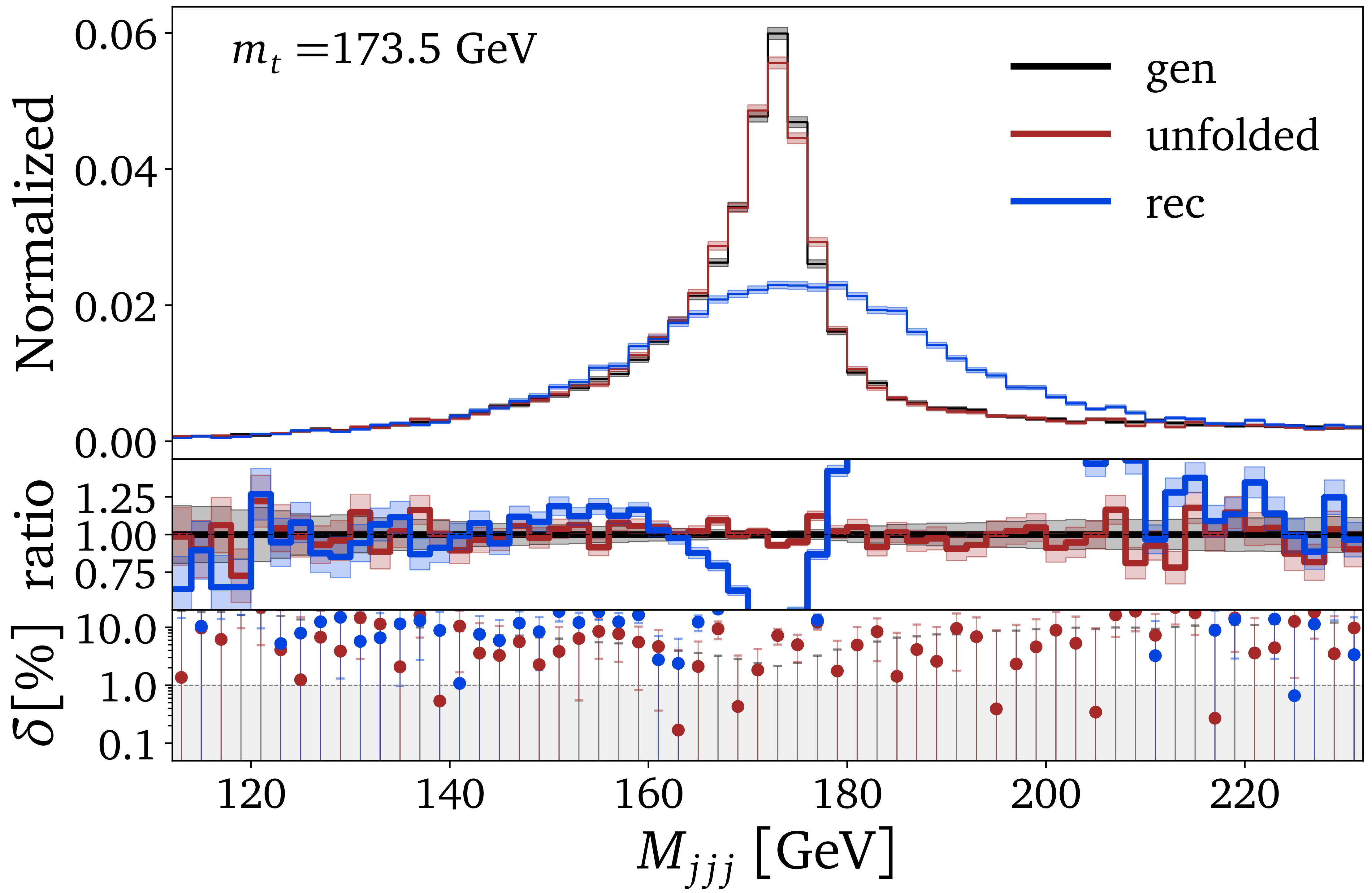
1. Augment training data with simulation from different top masses
2. Estimate batch-wise $m_d \approx$ weighted-median(M_{jjj}^{batch}) on detector-level

Removing Model-Dependence!

Train with full CMS simulation with $m_t = [172.5$ GeV, 169.5 GeV, 175.5 GeV$]$

Test by unfolding simulation with $m_t = 171.5$ GeV & 173.5 GeV

Unfolded distribution of triple jet mass within $\mathcal{O}(1\%)$ of truth particle-level **without** bias



ML task becomes much harder

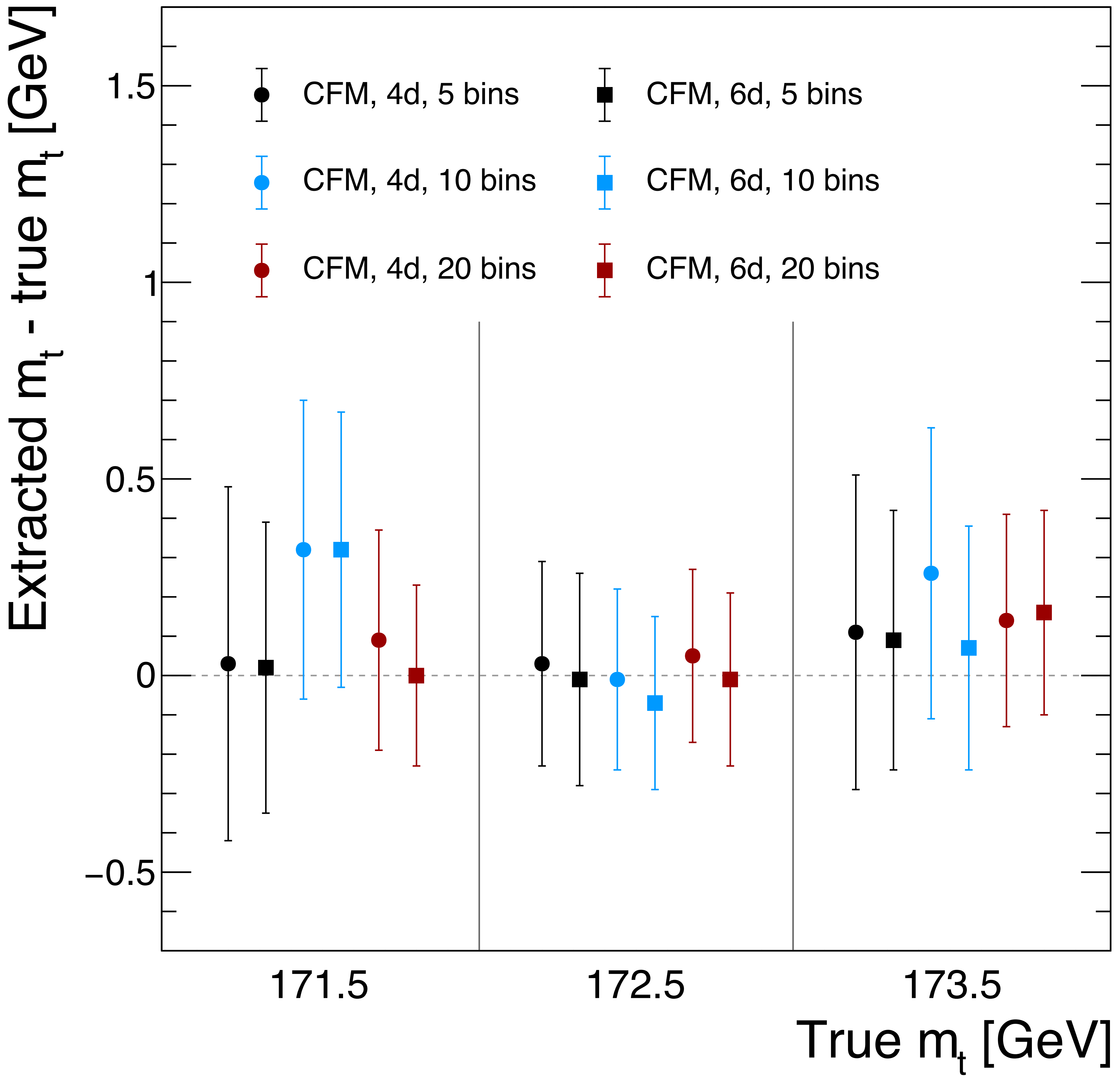
Mass Measurement

For a fixed top mass:

Choose subset of test data of 41000 detector-level events

Unfolded 1000 bootstrapped replicas

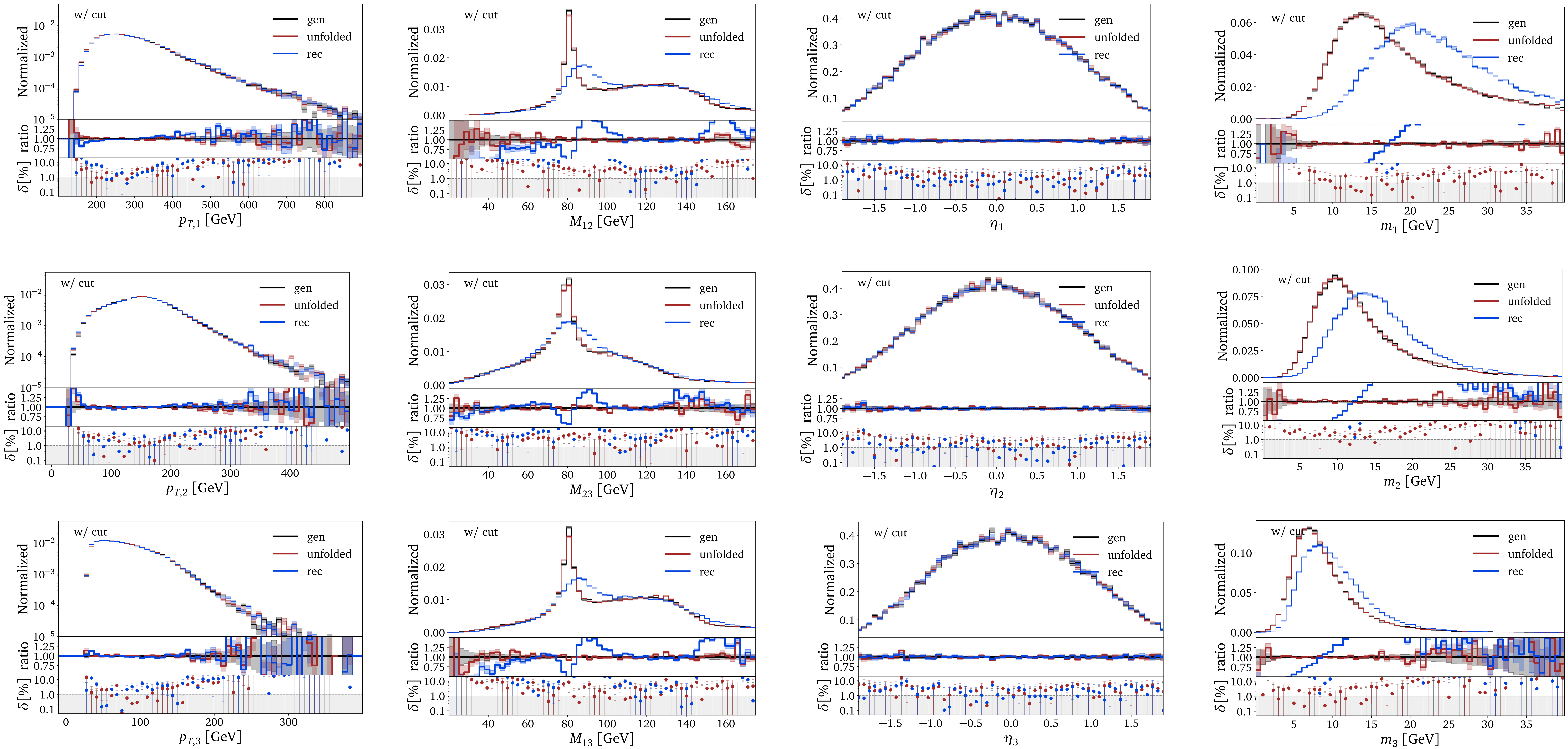
Estimate covariance matrix and mean by 1000 different unfolded distributions



→ Reliably measure top mass without bias

→ Reducing systematic uncertainty by **60%** and statistical by up to **36%**

Full Phase Space Unfolding (12d)

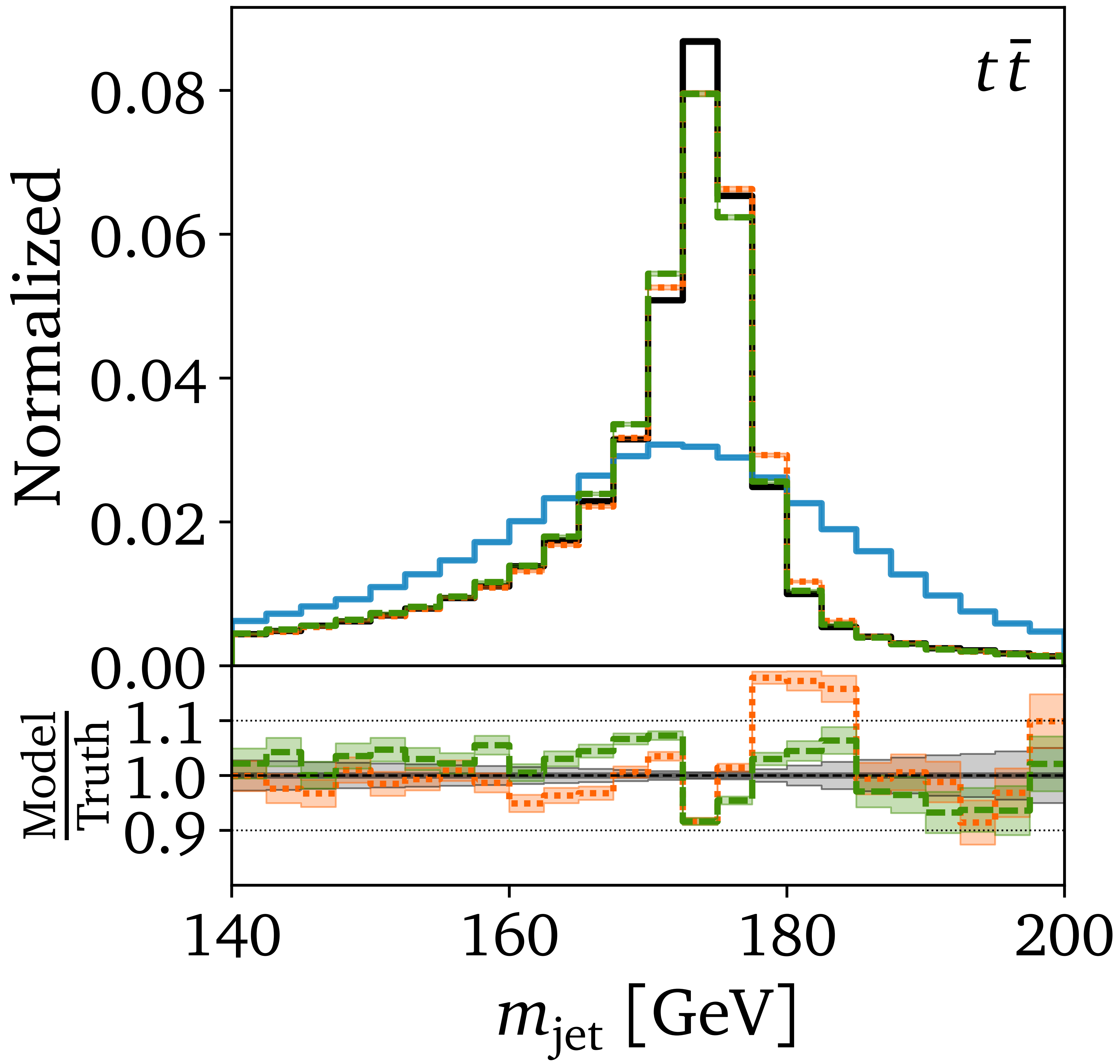


Full Phase Space Unfolding $\mathcal{O}(100)$ d?

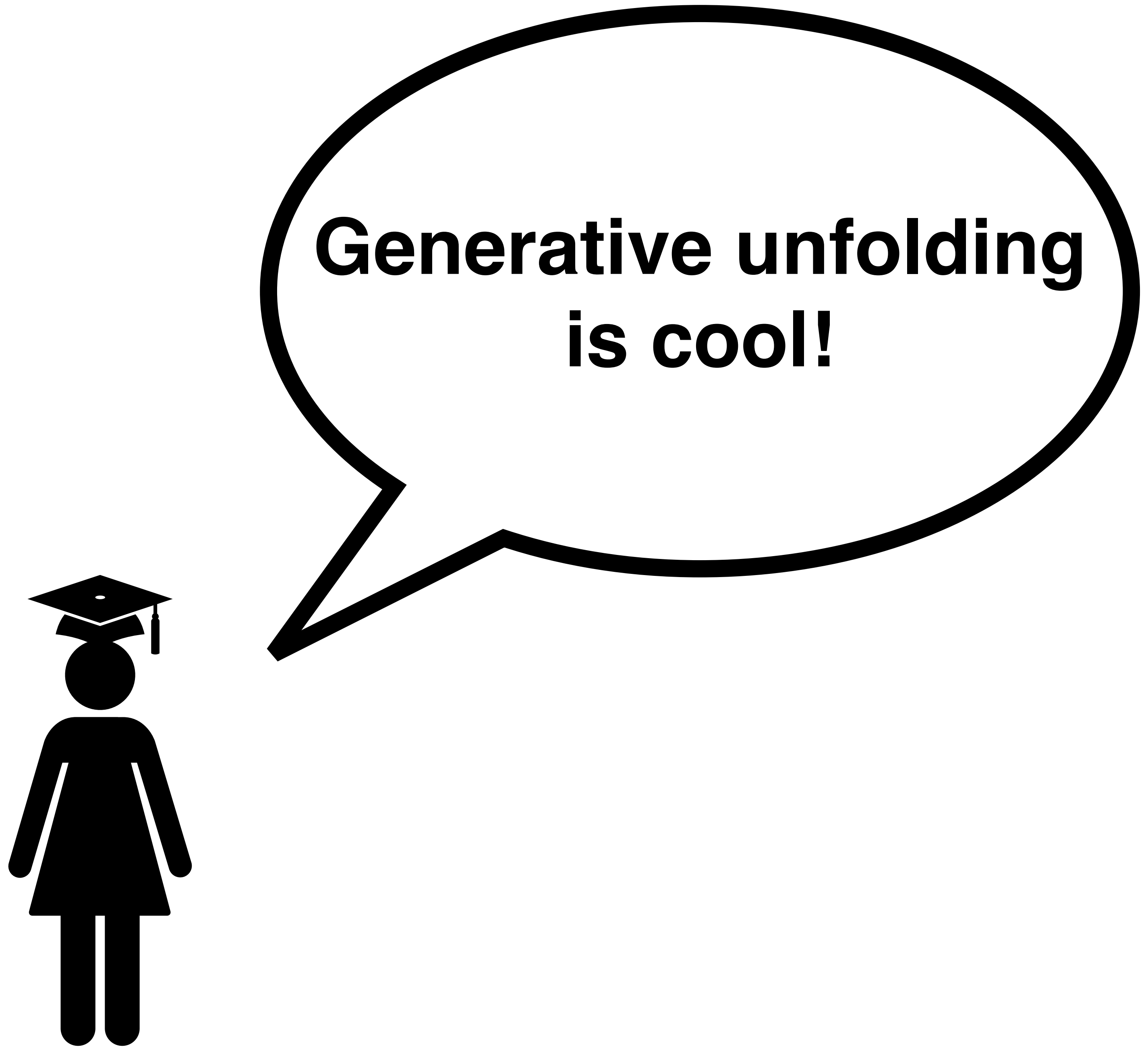
1. Step: MC-MC unfolding

Unfolding ~$\mathcal{O}(100)$ dimensional phase spaces

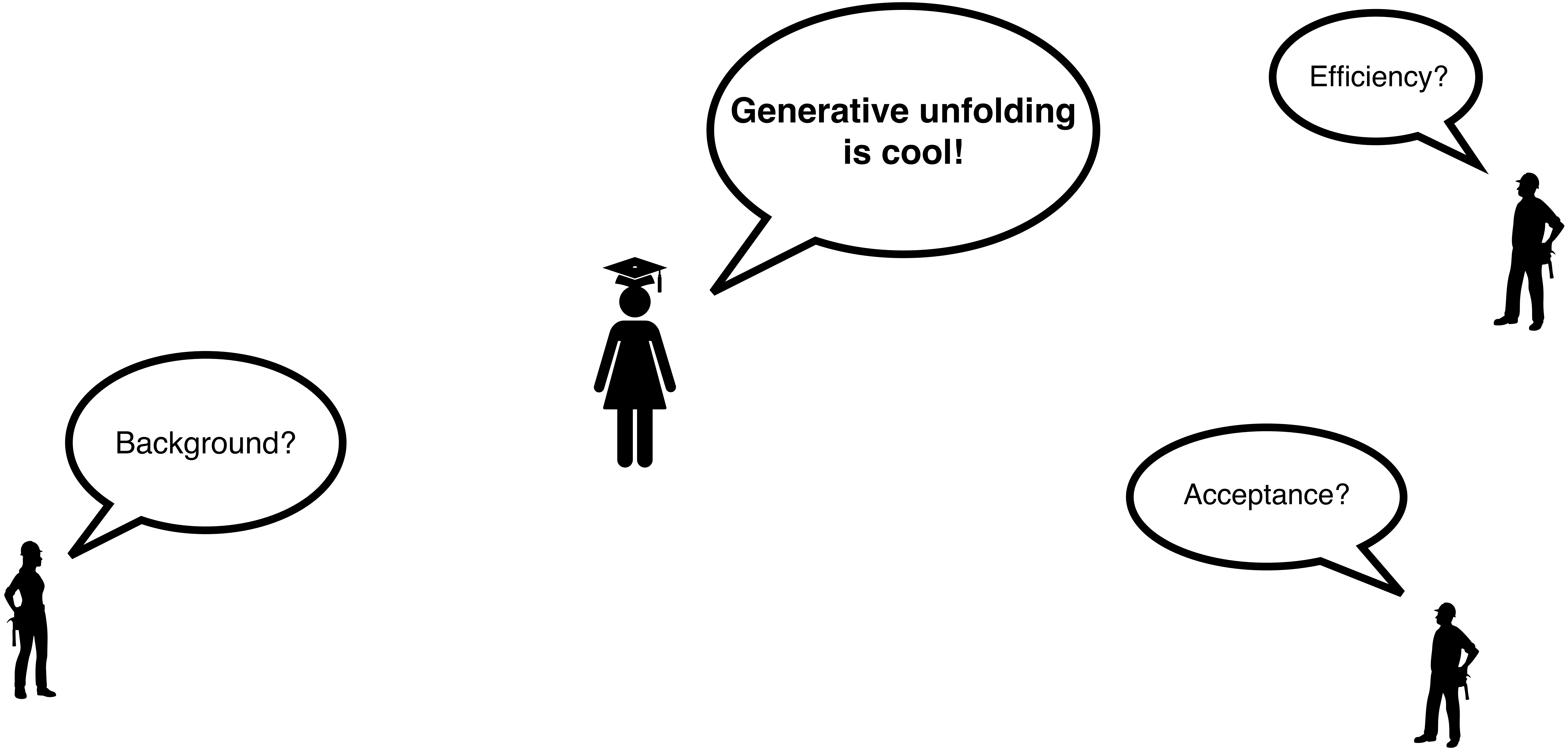
High dimensional correlations are unfolded to $\mathcal{O}(1)$ %-level with equivariant network



Imagine a scenario…



Imagine a scenario…



The Dream of Unfolding

What we have

Detector-level measurement

=

Signal + Background inside detector-level fiducial region

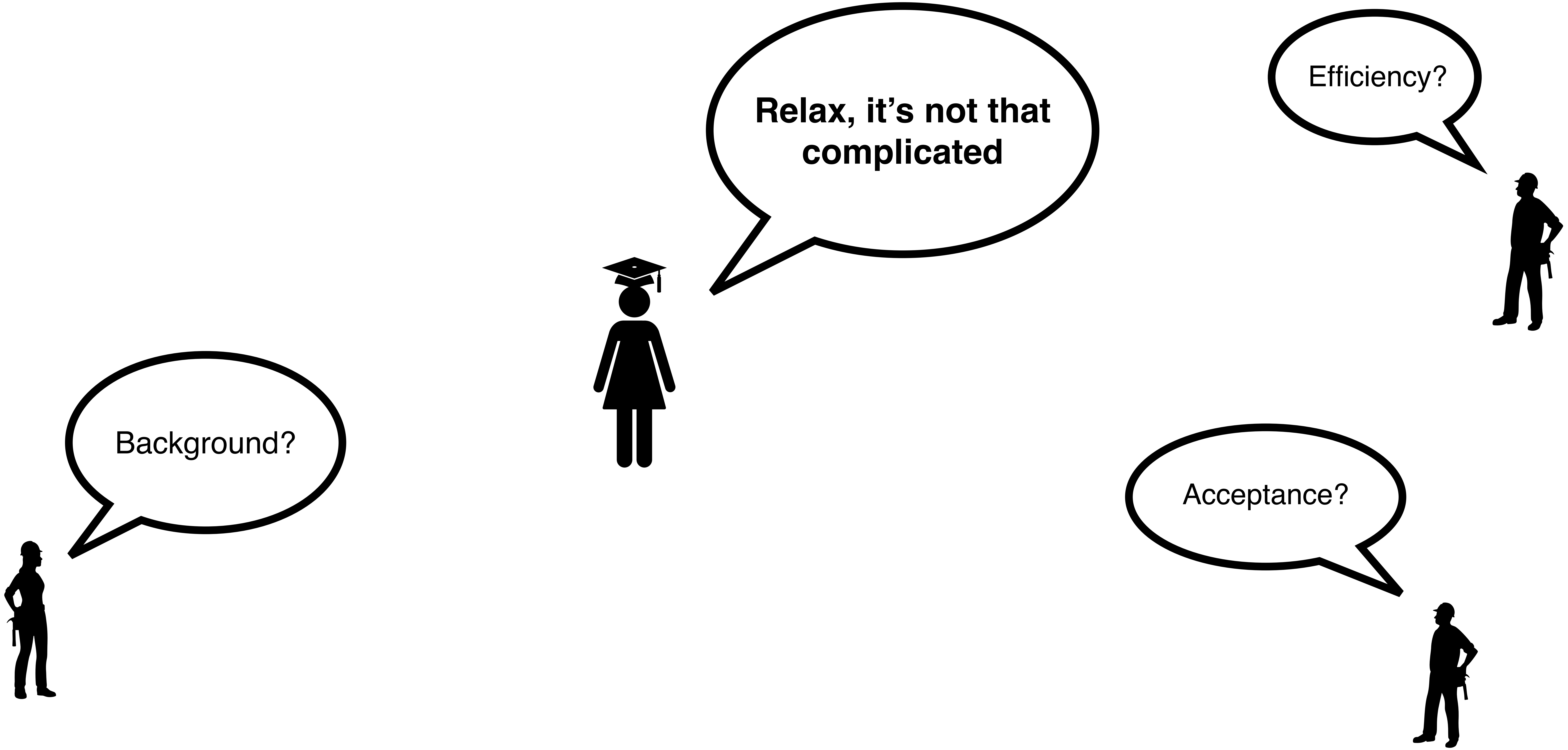
What we want

Particle-level distribution

=

Signal inside particle-level fiducial region

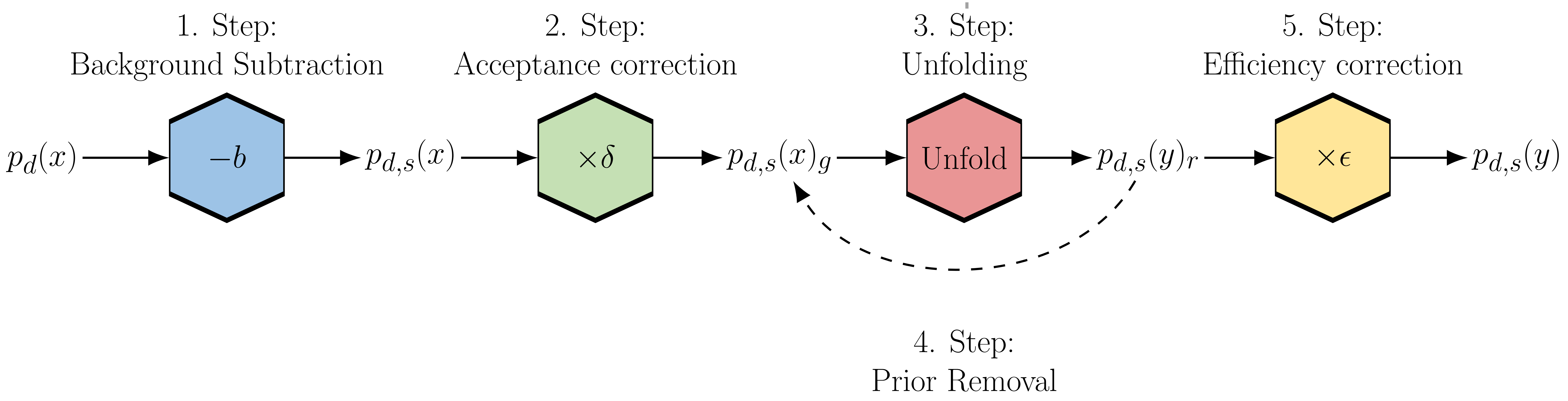
Imagine a scenario…



Getting there

Detector-level measurement

Particle-level distribution



Background subtraction

$$p_{d,s}(x) = p_d(x) - p_{d,b}(x) \stackrel{!}{=} p_d(x) - p_{MC,b}(x) \stackrel{!}{>} 0$$

1. Assumption:

Data background = MC background

2. Assumption:

No local background-only region

Classifier-based: $\nu(x) = \dfrac{C(x)}{1 - C(x)} \approx \dfrac{p_d(x) - p_{MC,b}(x)}{p_d(x)}$

Generator-based: $p_{\text{model},s}(x) \approx p_d(x) - p_{MC,b}(x)$

arXiv:2105.04448,
arXiv:1912.08824

Acceptance Correction

Local acceptance probability $p_{d,s}(g\,|\,x) = \dfrac{p(g)p_{d,s}(x)_g}{p(g)p_{d,s}(x)_g + p(\bar{g})p_{d,s}(x)_{\bar{g}}} \stackrel{!}{=} p_{MC,s}(g\,|\,x)$

with global acceptance probability $p(g) = \dfrac{N_g}{N_{\text{total}}}$

1. Assumption:

Data acceptance = MC acceptance

Classifier-based: $\delta(x) = \dfrac{p(g)p_{MC,s}(x)_g}{p(g)p_{MC,s}(x)_g + p(\bar{g})p_{MC,s}(x)_{\bar{g}}}$

1. Multiply $\delta(x)p_{d,s}(x)$

2. Hit or miss with probability $\delta(x)$

arXiv:2310.07752

Efficiency Correction

Local efficiency probability $p_{d,s}(r\,|\,y) = \dfrac{p(r)p_{d,s}(y)_r}{p(r)p_{d,s}(y)_r + p(\bar{r})p_{d,s}(y)_{\bar{r}}} \overset{!}{=} p_{MC,s}(r\,|\,y)$

with global efficiency probability $p(r) = \dfrac{N_r}{N_{\text{total}}}$

1. Assumption:

Data efficiency = MC efficiency

Classifier-based: $\epsilon(y) = \dfrac{p(r)p_{MC,s}(y)_r + p(\bar{r})p_{MC,s}(y)_{\bar{r}}}{p(r)p_{MC,s}(y)_g}$

1. Multiply $\epsilon(y)p_{d,\text{unfold}}(y)$

2. Oversample from $p_{d,\text{unfold}}(y)$
Hit or miss with with probability $\epsilon(y)$

arXiv:2310.07752

Application to Z + jets

What we have

Detector-level measurement
(6d jet observables)

=

Signal **(Z + jets)**

\+ Background **(Z + V)**

inside detector-level fiducial region

$p_{T,r} > 150$ GeV

Prior-dependence
(two different Pythia simulations for data and MC)

What we want

Particle-level distribution
(6d jet observables)

=

Signal **(Z + jets)**

inside particle-level fiducial region

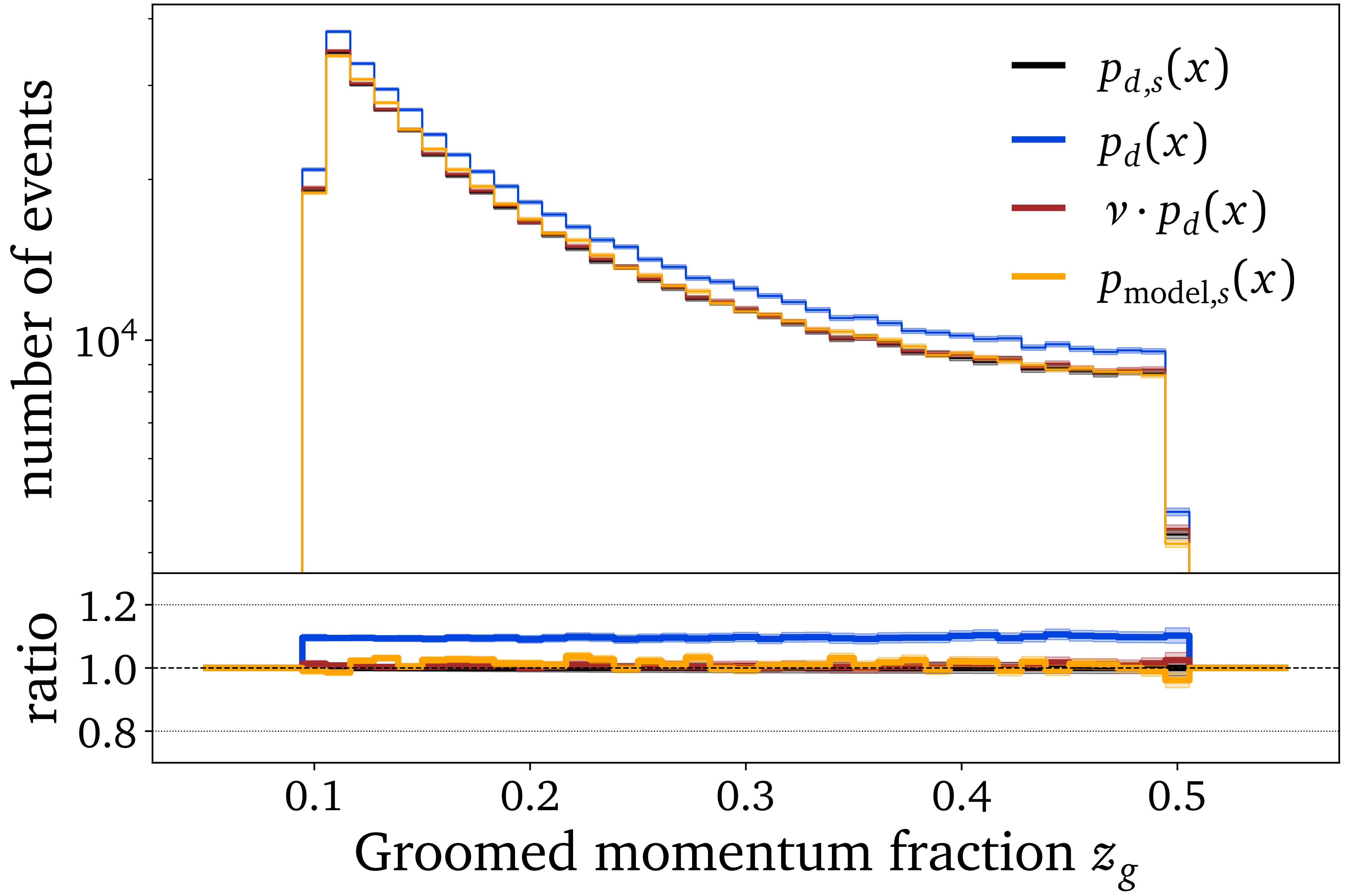
$p_{T,g} > 150$ GeV

Prior-independence

Subtracting (Z + V)-bkg

8 % background contamination

Able to correct global background effects

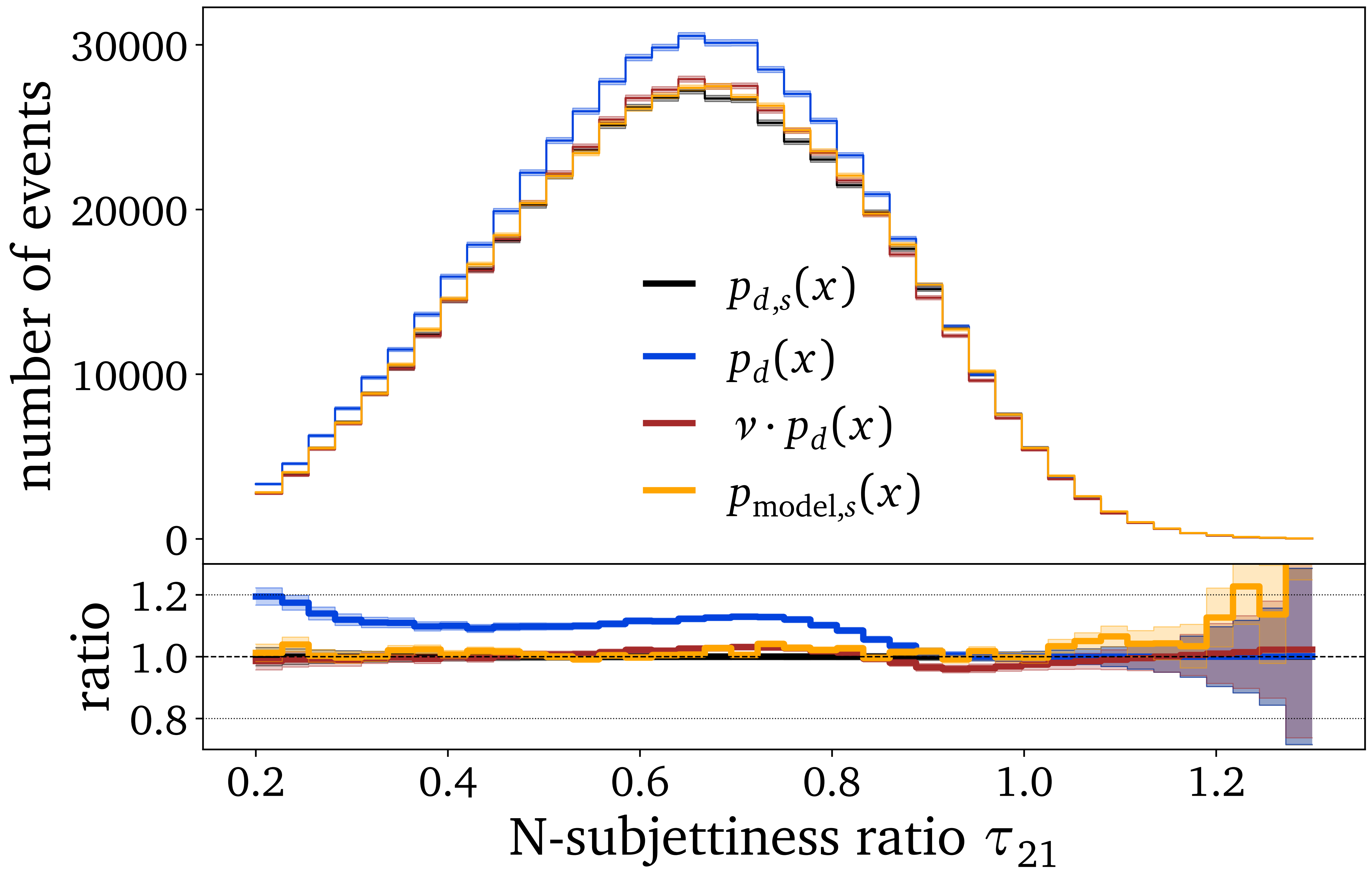


Subtracting (Z + V)-bkg

8 % background contamination

Able to correct global background effects

AND local background effects

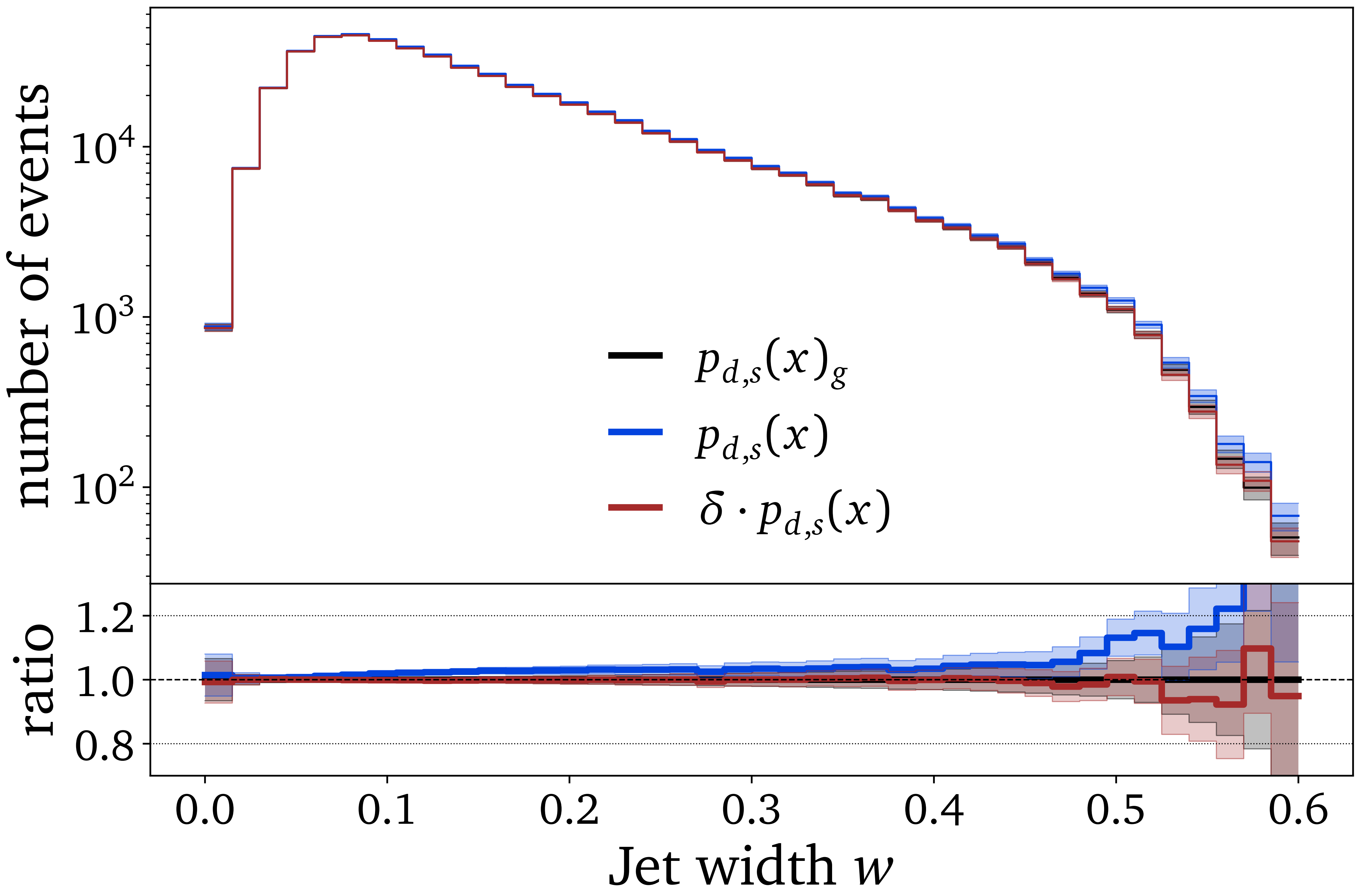


Acceptance correction

Acceptance effects are rather small (2%)

Detector-level and particle-level cut strongly correlated

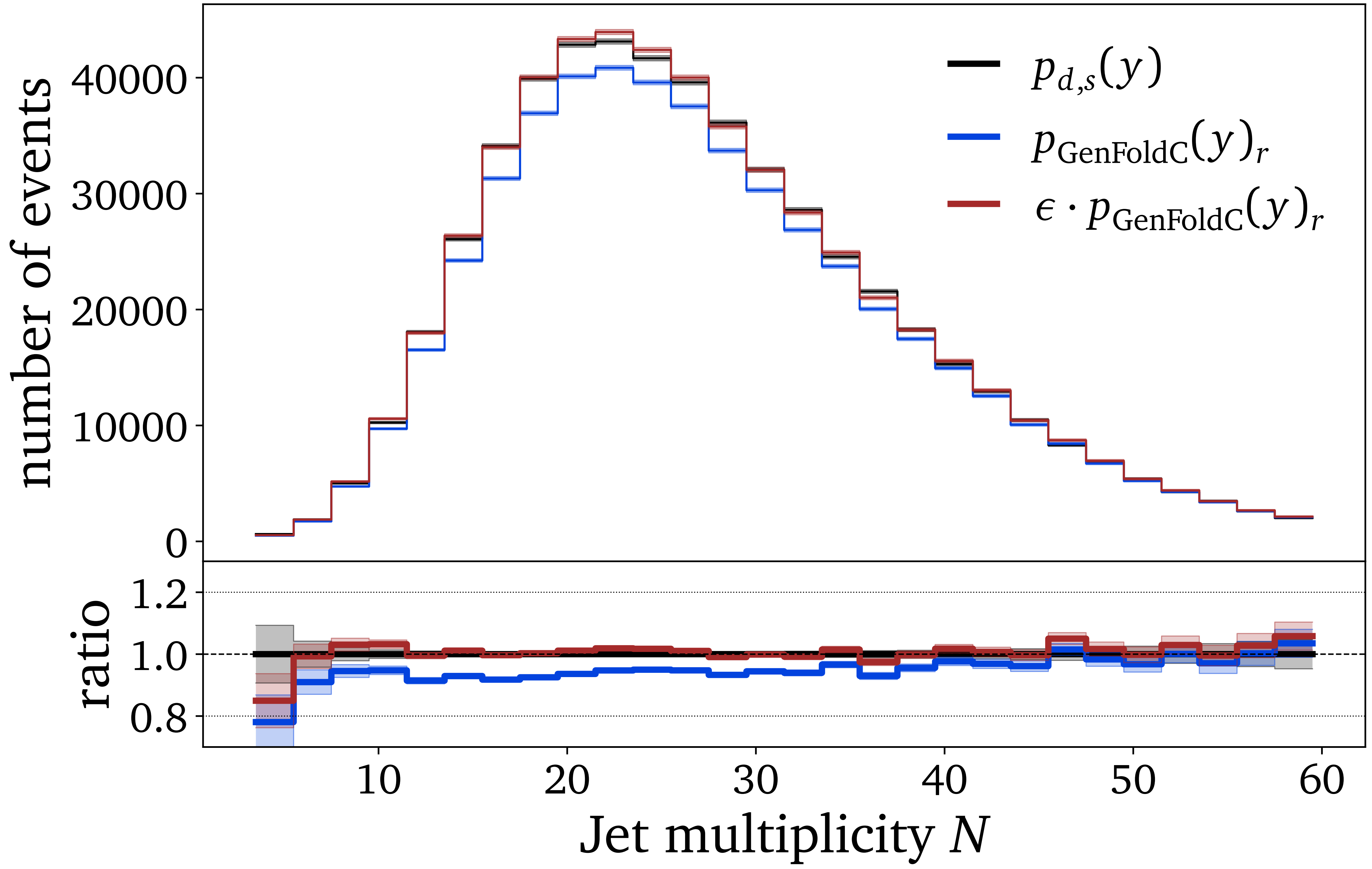
Able to correct for local and global effects



Efficiency

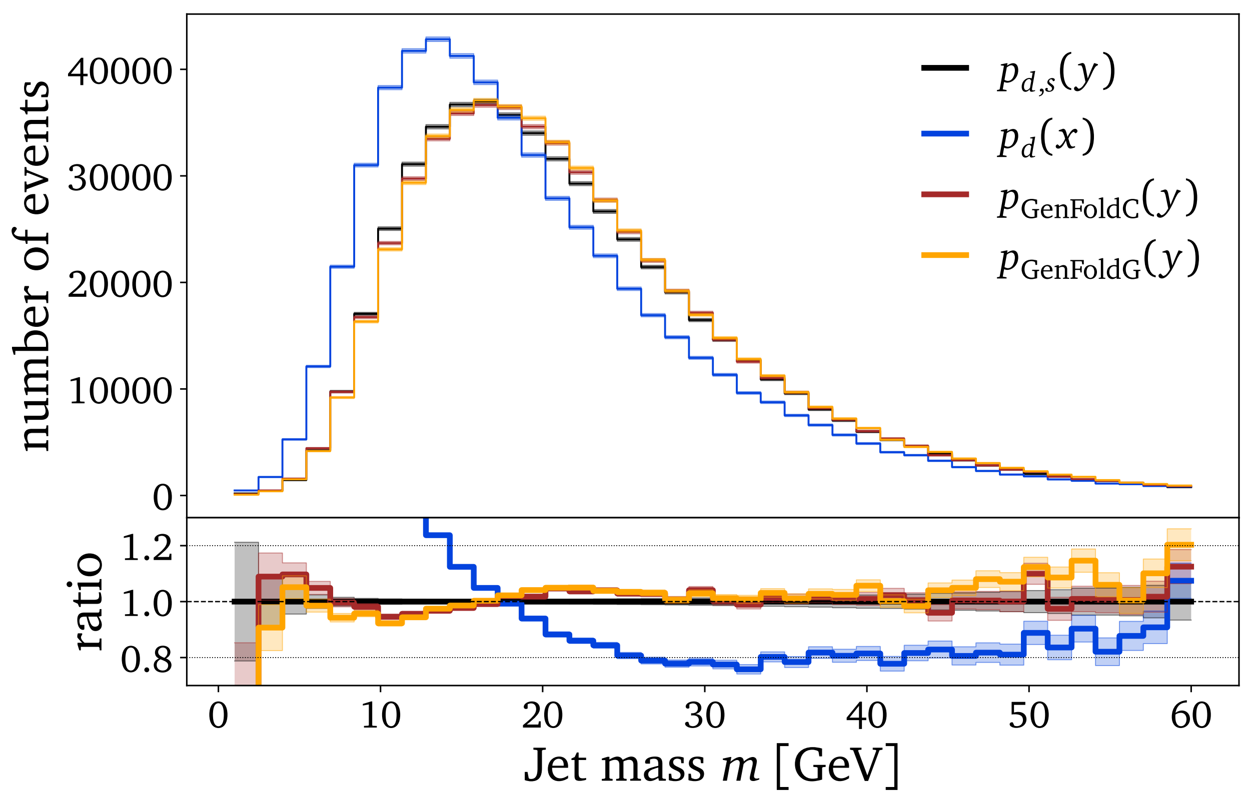
Efficiency effects are larger (6%)

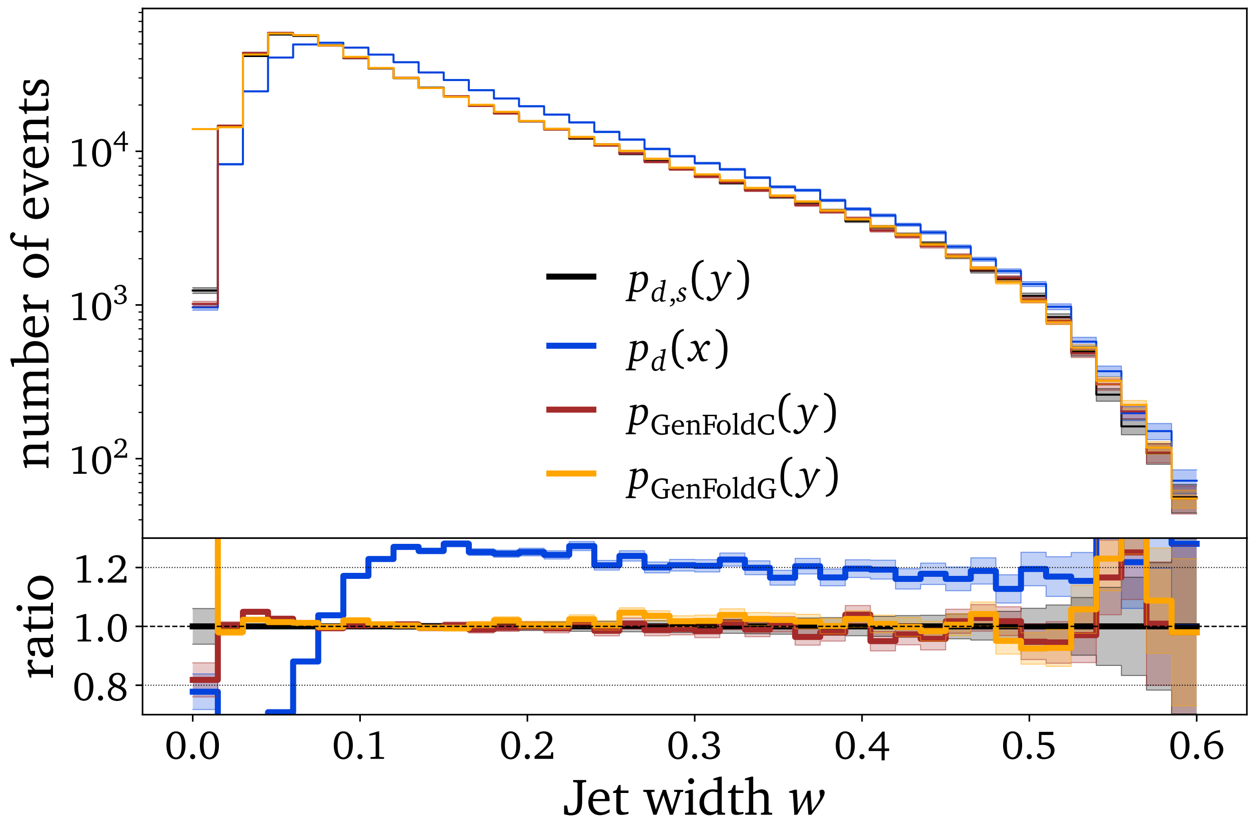
Able to correct for local and global effects

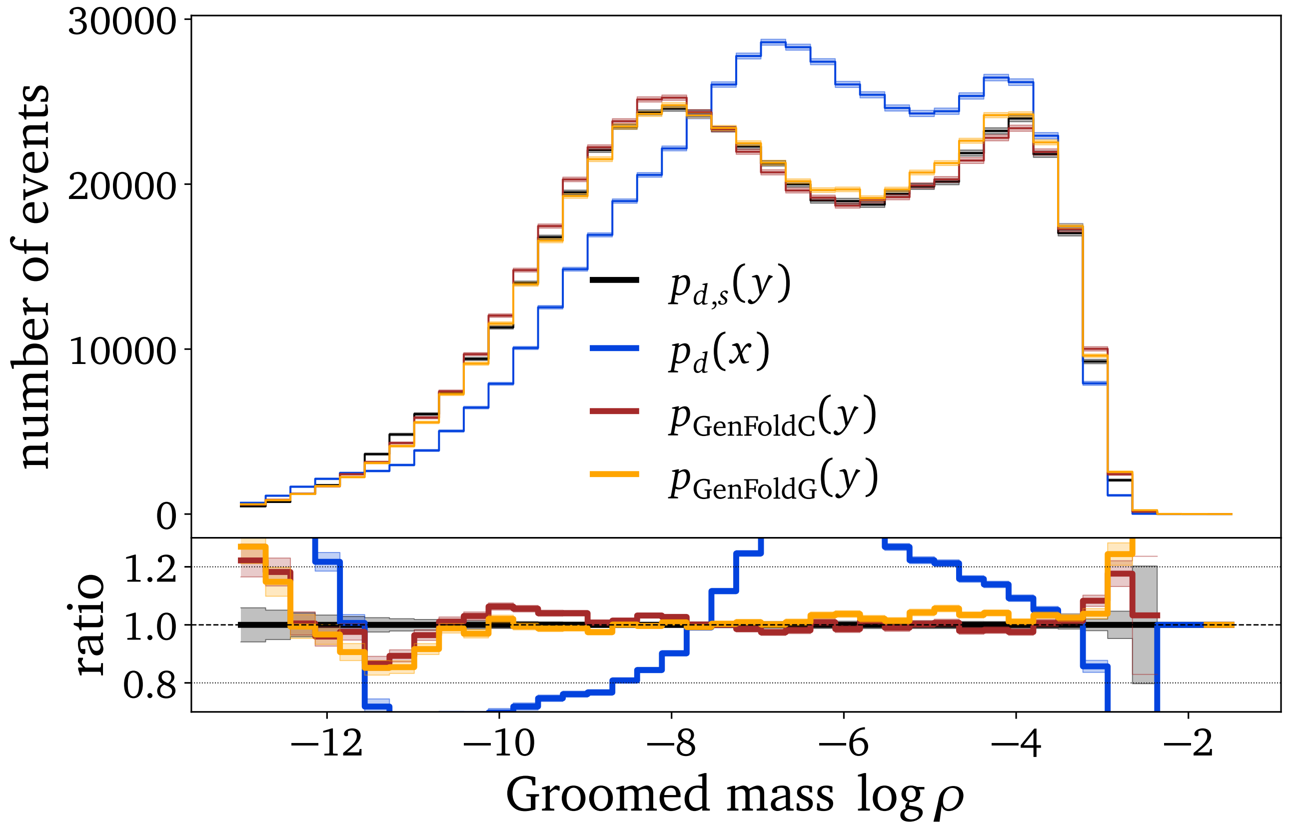


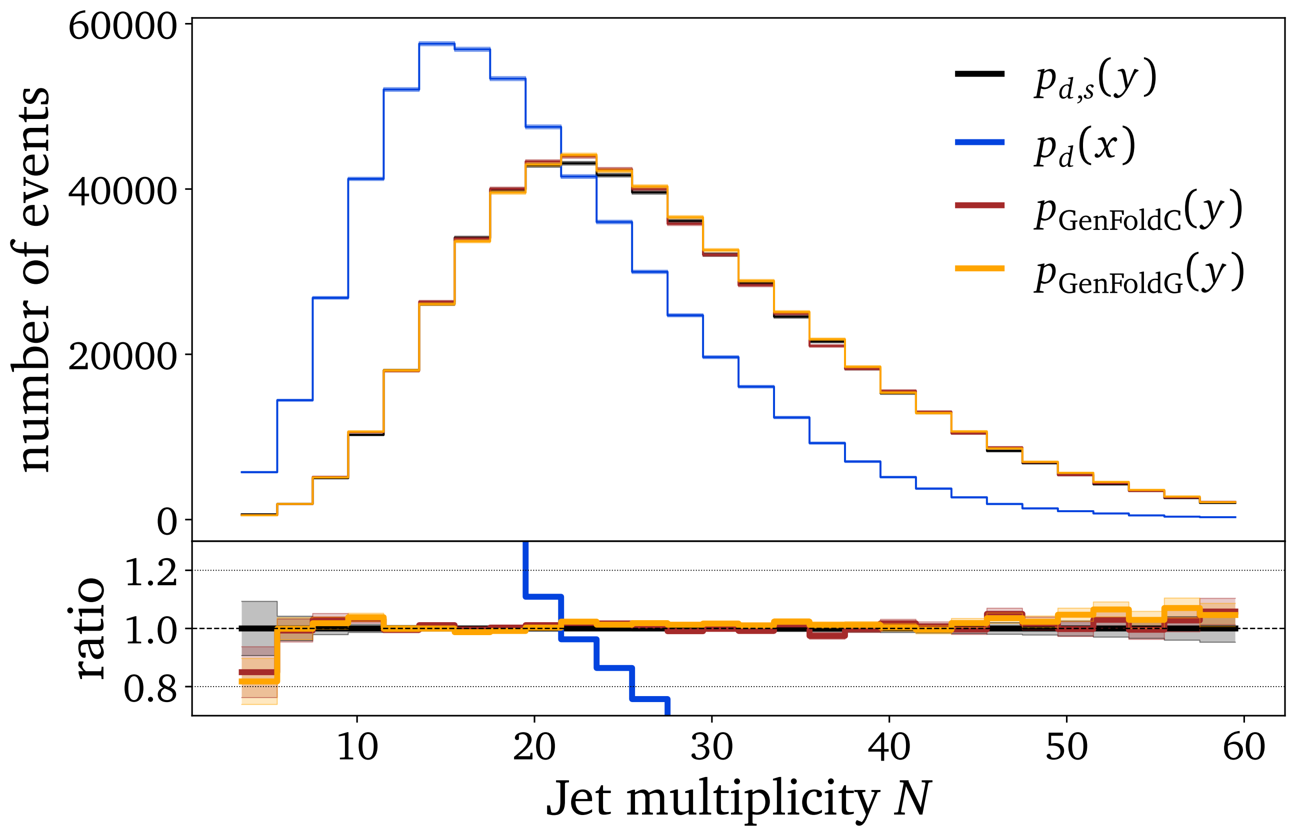
Final Performance

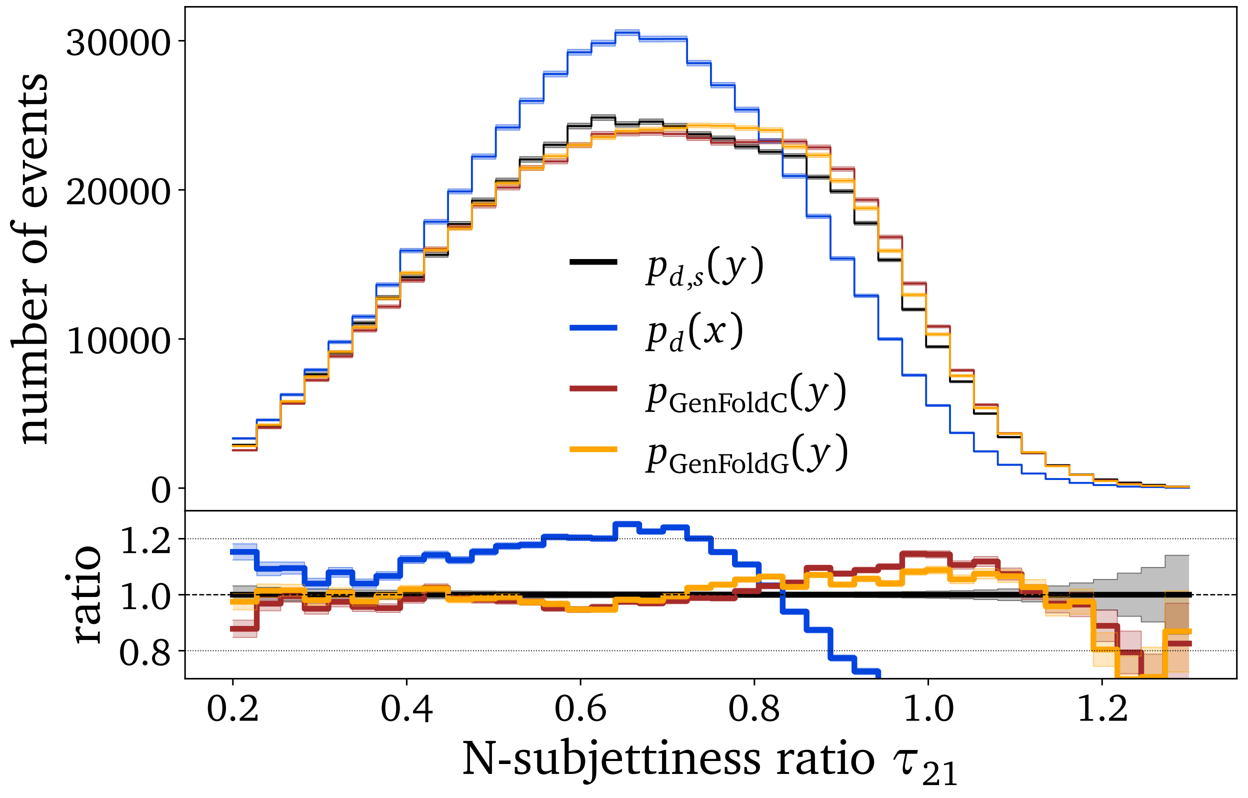
Detector-level measurement → **Particle-level unfolding**

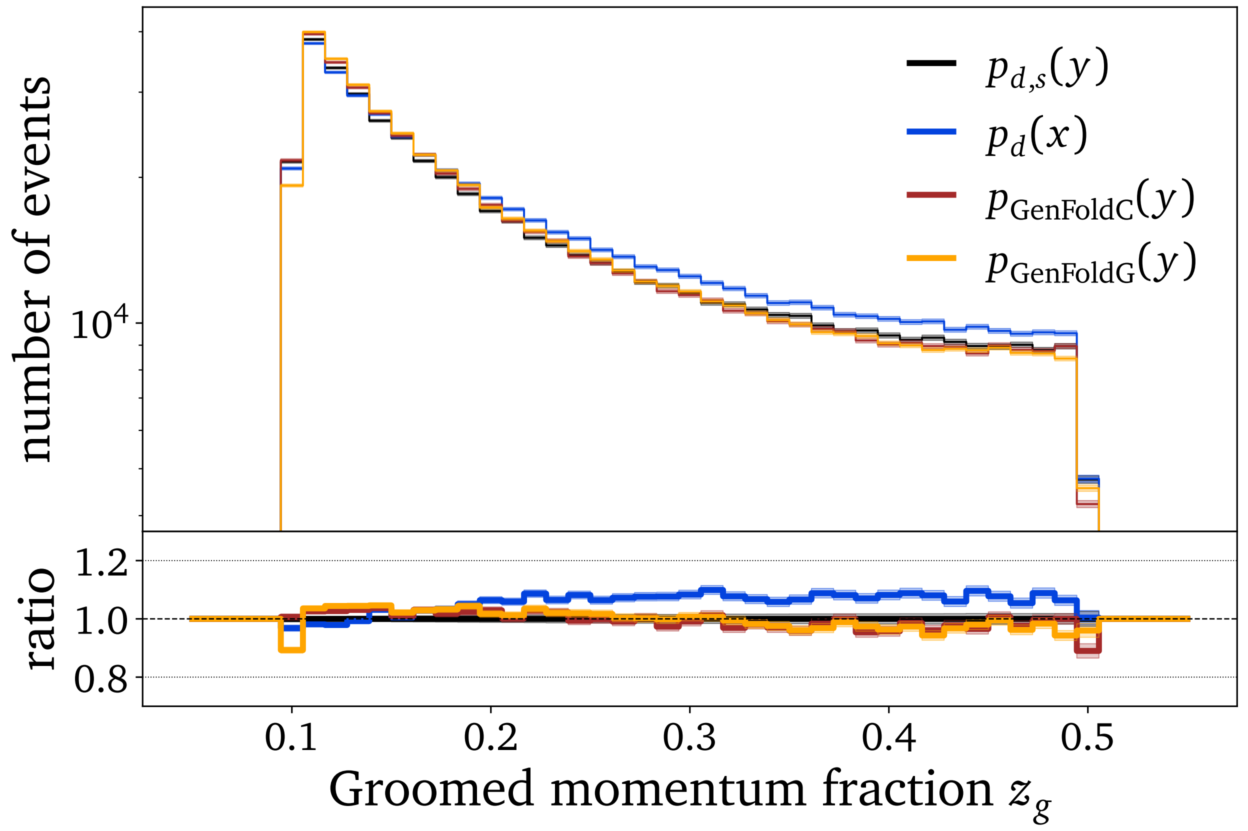










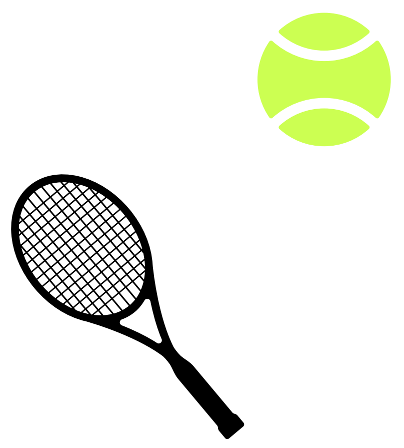


And now what?

Generative unfolding is cool and **analysis-ready**

Correcting for **background, efficiency and acceptance** is easy peasy

Correcting for **prior-dependence** can be tricky, but possible



Let's use it!