

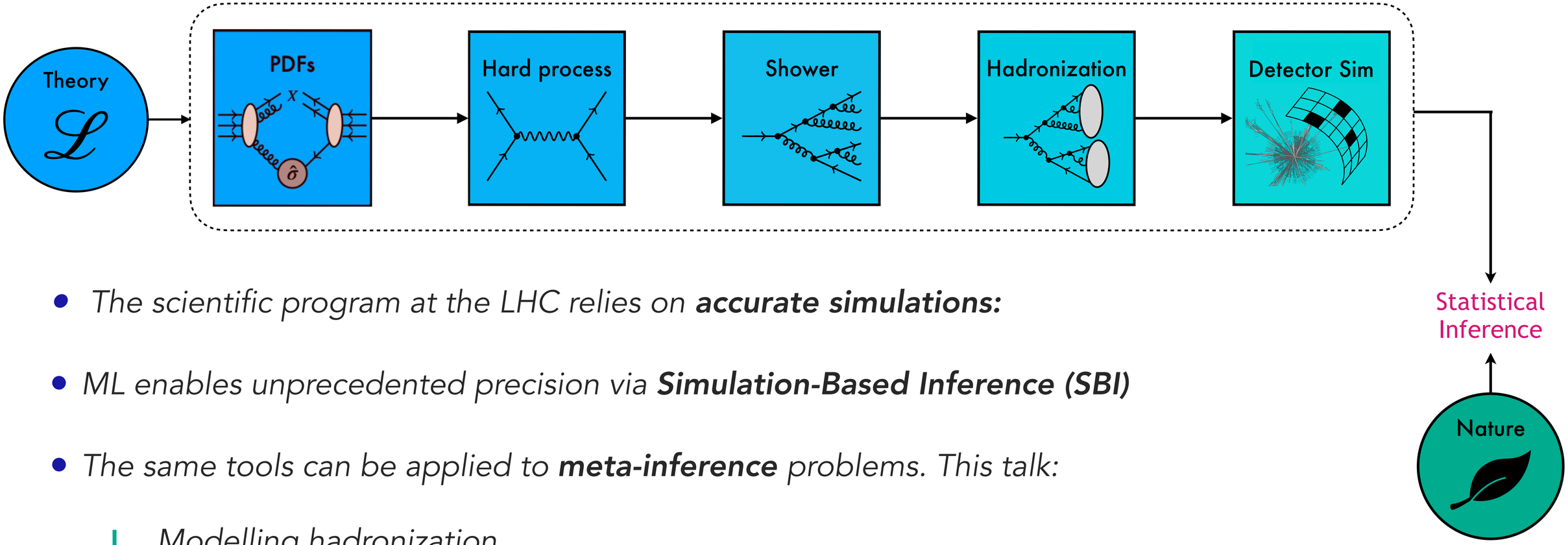
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

ITP

Machine Learning for Hadronization and Unfolding at Colliders

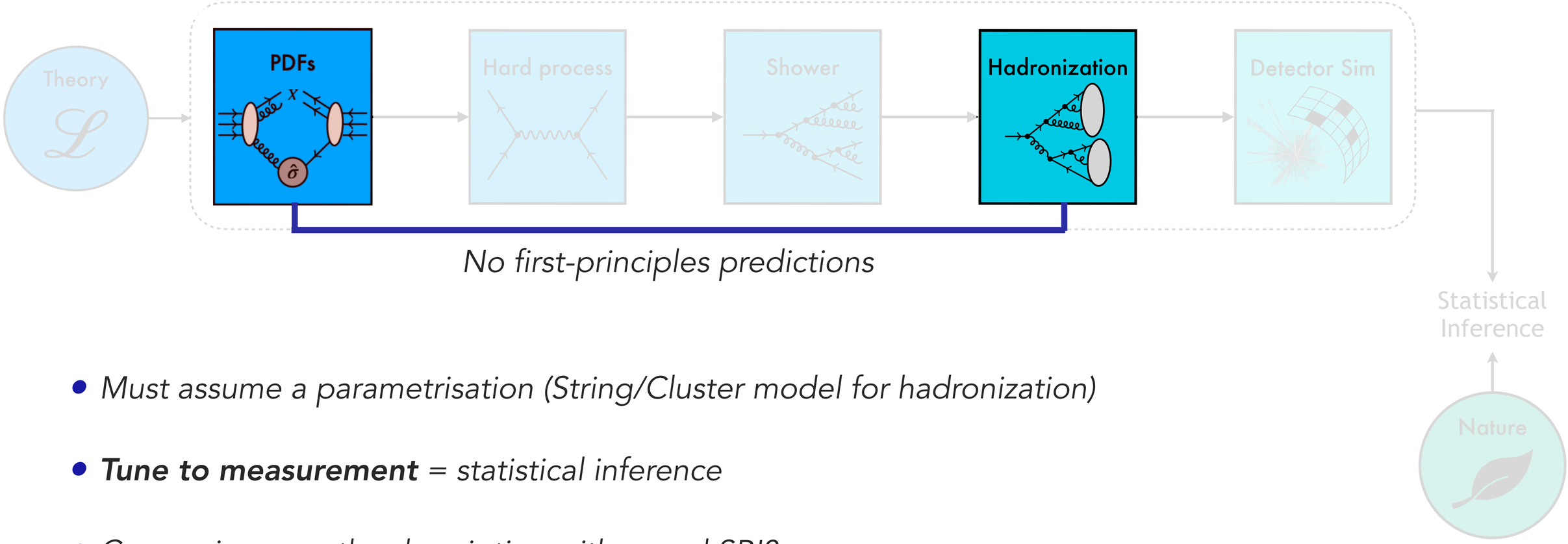
***Ayodele Ore**, NPTwins 2026 @ ECT*, Trento*

Simulations and inference at colliders



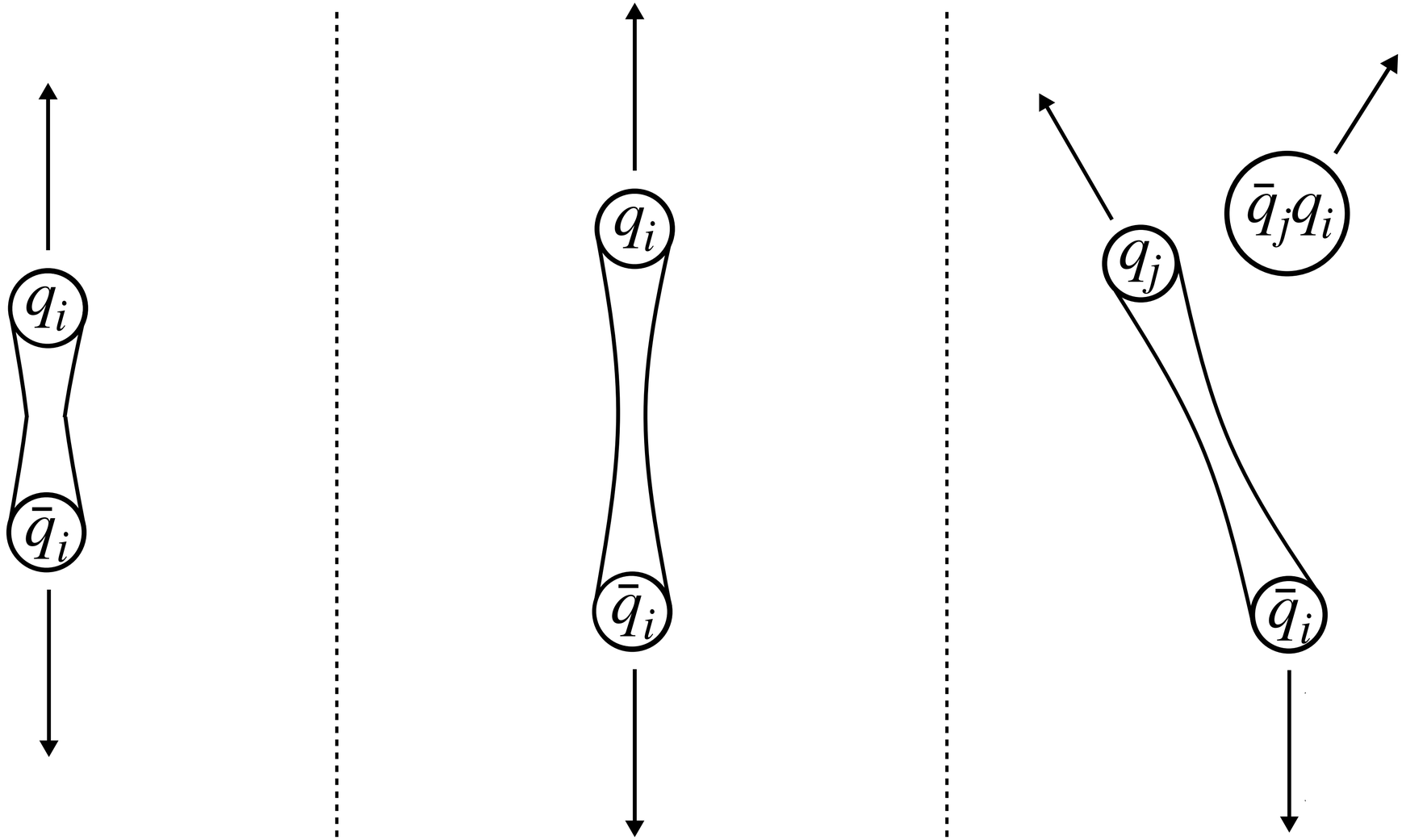
- *The scientific program at the LHC relies on* ***accurate simulations:***
- *ML enables unprecedented precision via* ***Simulation-Based Inference (SBI)***
- *The same tools can be applied to* ***meta-inference*** *problems. This talk:*
  - I. *Modelling hadronization*
  - II. *Unfolding data*

Modelling hadronization



- *Must assume a parametrisation (String/Cluster model for hadronization)*
- ***Tune to measurement** = statistical inference*
- *Can we improve the description with neural SBI?*

String model hadronization (simplified)

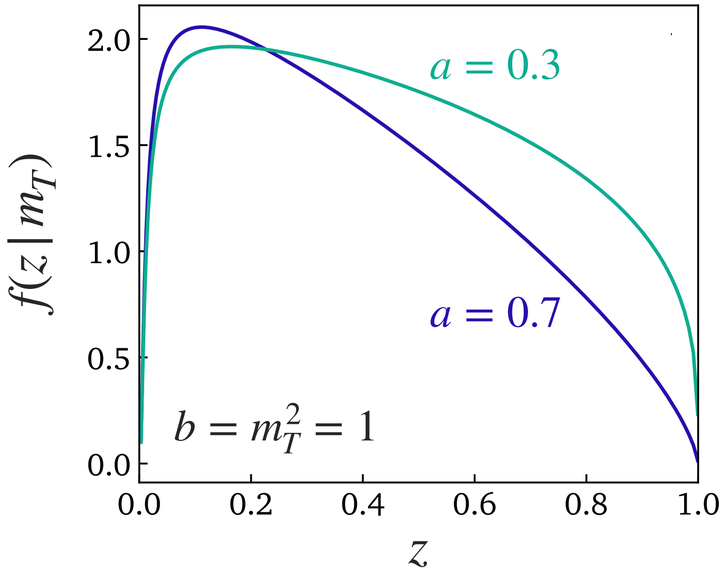


String model hadronization (simplified)

- *Characterise string* ***breaks*** *by* $s \equiv \{z, m_T\}$
  - z: *Lightcone momentum fraction*
  - m_T: *Transverse mass*

- *Sample z from* ***fragmentation function****:*

$$f(z \,|\, m_T^2) = \frac{(1-z)^a}{z} \exp\left(-\frac{b\, m_T^2}{z}\right)$$

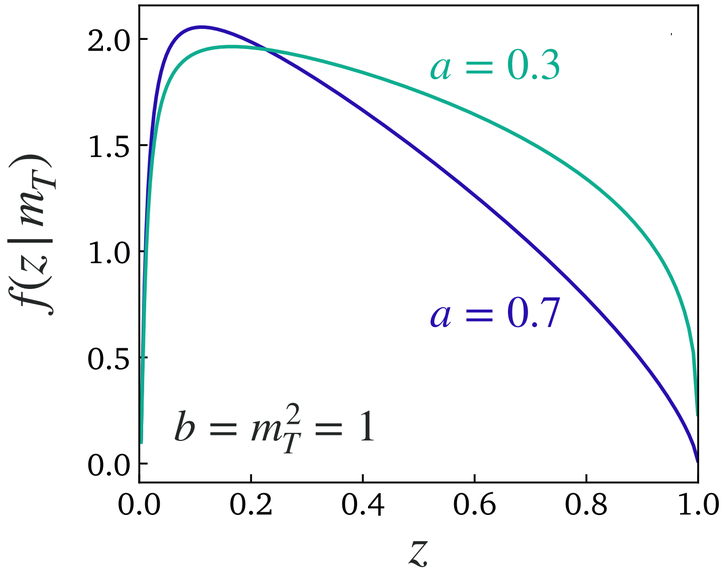


String model hadronization (simplified)

- *Characterise string* ***breaks*** *by* $s \equiv \{z, m_T\}$
  - z: *Lightcone momentum fraction*
  - m_T: *Transverse mass*

- *Sample z from* ***fragmentation function****:*

$$f(z\,|\,m_T^2) = \frac{(1-z)^a}{z} \exp\left(-\frac{b\,m_T^2}{z}\right)$$



- *Simulate fragmentation* ***history*** *as a sequence of breaks*

$$\zeta \xrightarrow{s_1 \sim f(z\,|\,m_T)\,p(m_T)} \zeta' \xrightarrow{s_2 \sim f(z\,|\,m_T)\,p(m_T)} \zeta'' \xrightarrow{s_3 \sim f(z\,|\,m_T)\,p(m_T)} \cdots \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} \equiv H$$

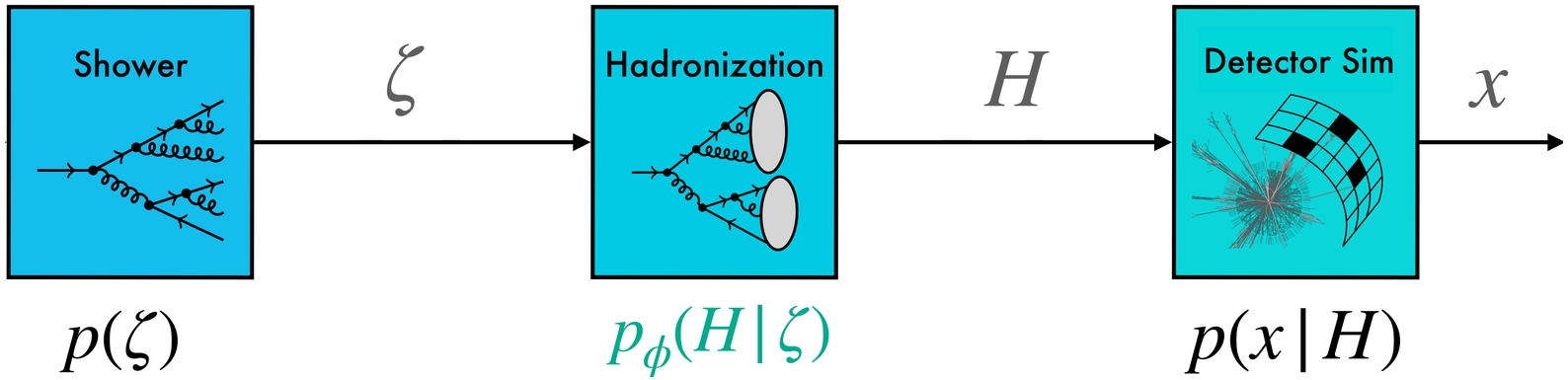
String state — *'History'*

A latent variable problem

String model:

$$p_\phi(H \mid \zeta) = \prod_{s \in H} f_\phi(z \mid m_T)$$

- *The full probabilistic model is*



- *Only x is observable in data. Best you can do is* ***match marginal likelihood***

$$p_{\text{data}}(x) \quad \xleftrightarrow{\text{Match}} \quad p_\phi(x) = \int dH d\zeta \, p(x \mid H) \, p_\phi(H \mid \zeta) \, p(\zeta)$$

Choices for matching objective

- $D_{\mathrm{KL}}[p_{\mathrm{data}}, p_\phi] = \mathbb{E}_{p_{\mathrm{data}}(x)}\left[\log p_\phi(x)\right]$ $\quad p_\phi(x) = \int dH d\zeta\, p(x\,|\,H)\, p_\phi(H\,|\,\zeta)\, p(\zeta)$

  - x *Need likelihoods*
  - x *High-dim integral* $\Rightarrow$ *Requires detector surrogate*
  - x *Need gradients of detector sim*

Choices for matching objective

- $D_{\mathrm{KL}}[p_{\mathrm{data}}, p_\phi] = \mathbb{E}_{p_{\mathrm{data}}(x)}\left[\log p_\phi(x)\right]$ $\qquad p_\phi(x) = \int dH d\zeta\, p(x\,|\,H)\, p_\phi(H\,|\,\zeta)\, p(\zeta)$

  - x *Need likelihoods*
  - x *High-dim integral* $\Rightarrow$ *Requires detector surrogate*
  - x *Need gradients of detector sim*

- $D_{\mathrm{JS}}[p_{\mathrm{data}}, p_\phi] = \mathbb{E}_{p_{\mathrm{data}}(x)}\left[\log C(x;\phi)\right] + \mathbb{E}_{p_\phi(x)}\left[\log\left(1 - C(x;\phi)\right)\right]$ $\qquad C(x;\phi) \equiv \dfrac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_\phi(x)}$

  - ✓ *Likelihood-free*
  - ✓ *Auto-marginalization* $\Rightarrow$ *Requires Generative Adversarial Network (GAN)*
  - x *Need gradients of detector sim*

Matching by reweighting

$$p_\phi(H\,|\,\zeta) \longrightarrow w_\phi(H)\, p_{\text{ref}}(H\,|\,\zeta)$$

- *Solves main difficulties:*

  ✓ *Likelihood-free* ✓ *Bypass detector gradients* ? *Marginalisation*

- *Bonus:*

  ✓ *Physics prior* ✓ *Easy uncertainty propagation* ✓ *Seamless integration with event generators*

- *Idea implemented by MLhad collaboration:*

  [MLhad 2410.06342] ***H****istories and* ***O****bservables for* ***M****onte-Carlo* ***E****vent* ***R****eweighting (****HOMER****)*

  [MLhad, AO et al. 2509.03592] ***iHOMER****: cheap marginalisation + uncertainties*

$$w_\phi(H) = \prod_{s\in H} w_\phi(s) \qquad\qquad f_{\text{HOMER}}(z\,|\,m_T) \equiv w_\phi(s)\, f_{\text{ref}}(z\,|\,m_T)$$

HOMER approach

- ***Ingredients***: *Measurements* $x \sim p_{\text{data}}(x)$ *and simulated pairs* $(x, H) \sim p_{\text{ref}}(x, H)$

- *Distribution matching objective, now in terms of weights:*

$$\frac{p_{\text{data}}(x)}{p_{\text{ref}}(x)} \quad \xleftrightarrow{\text{Match}} \quad \int dH\, p_{\text{ref}}(H \mid x) \prod_{s \in H} w_\phi(s)$$

HOMER approach

- ***Ingredients***: *Measurements* $x \sim p_{\text{data}}(x)$ *and simulated pairs* $(x, H) \sim p_{\text{ref}}(x, H)$

- *Distribution matching objective, now in terms of weights:*

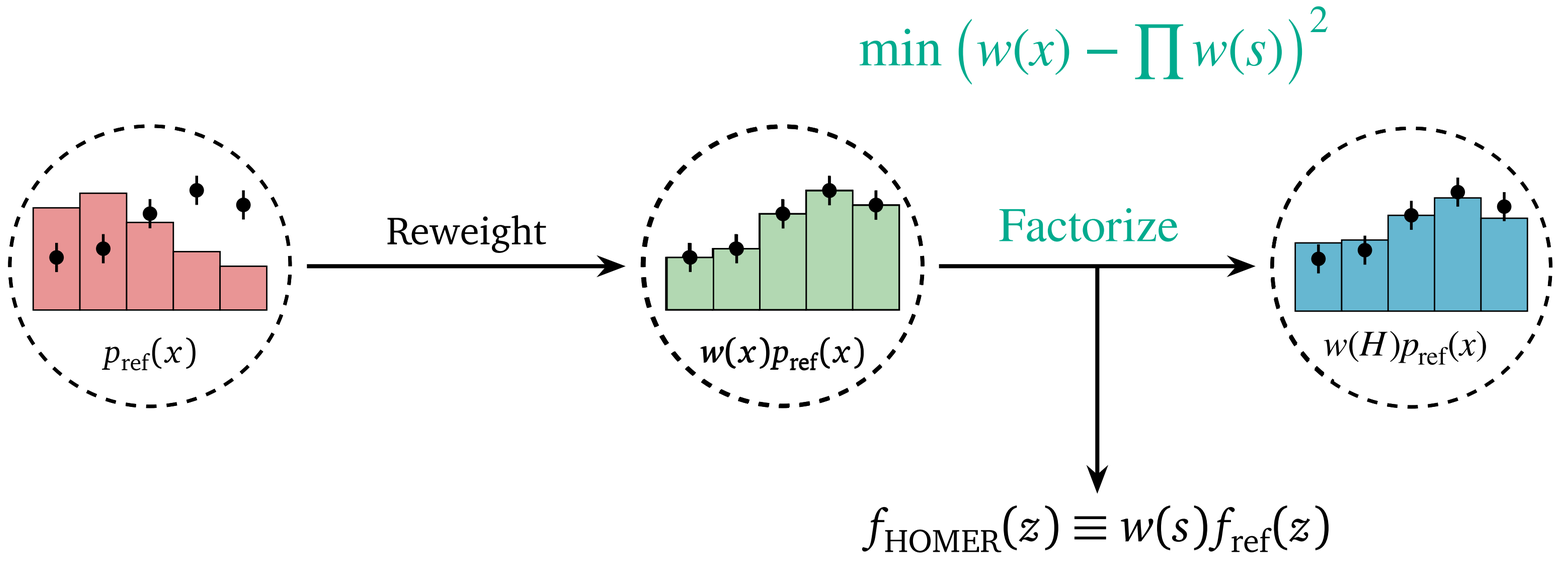
$$\frac{p_{\text{data}}(x)}{p_{\text{ref}}(x)} \quad \text{Match} \quad \int dH\, p_{\text{ref}}(H \mid x) \prod_{s \in H} w_\phi(s)$$

Approximate with a classifier

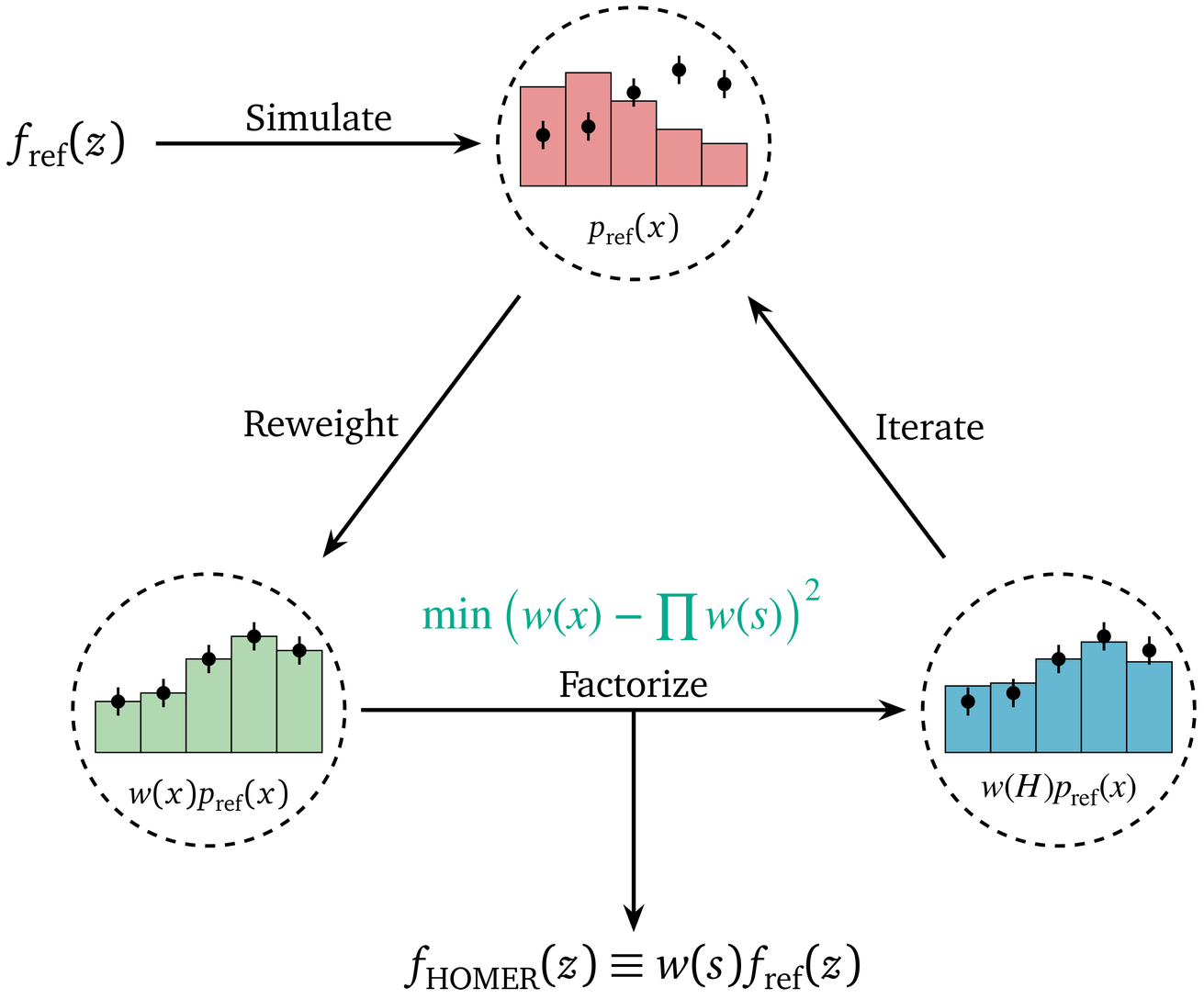
Assume p(Hlx) "narrow". Take H from pair with x

$$w_\theta(x) \quad \text{Match} \quad \prod_{s \in H} w_\phi(s)$$

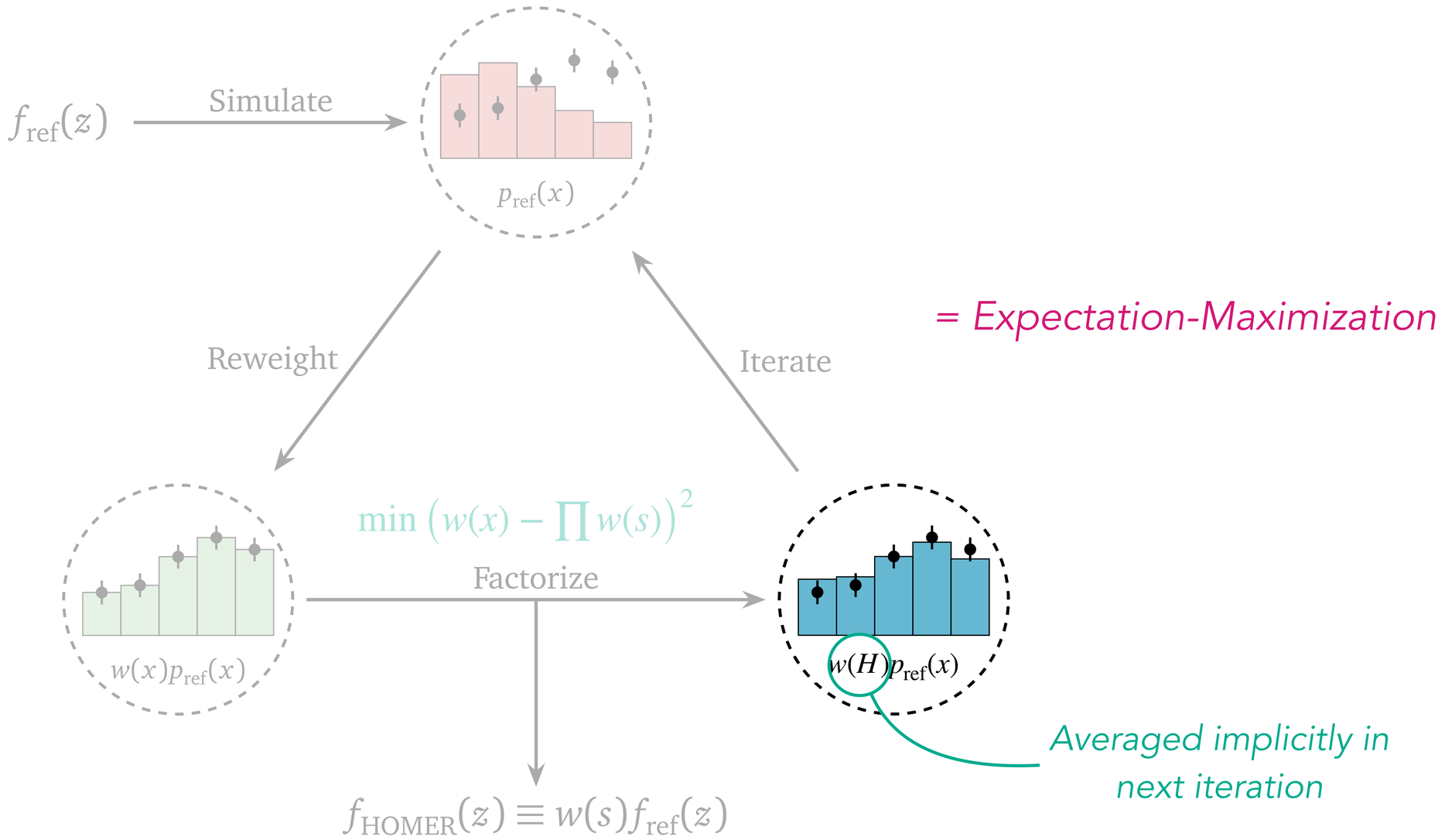
HOMER Steps 1 and 2



iHOMER: Iterations



iHOMER: Iterations



= Expectation-Maximization

Averaged implicitly in next iteration

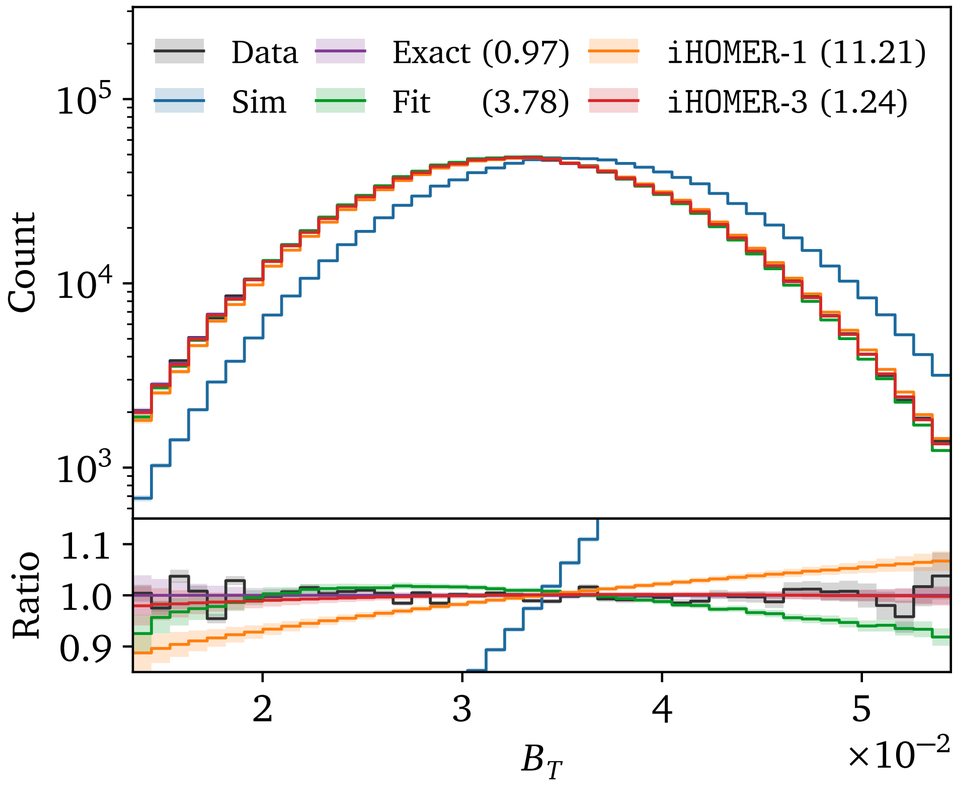
Reweighting accuracy: Observables

Cannot fit with standard form

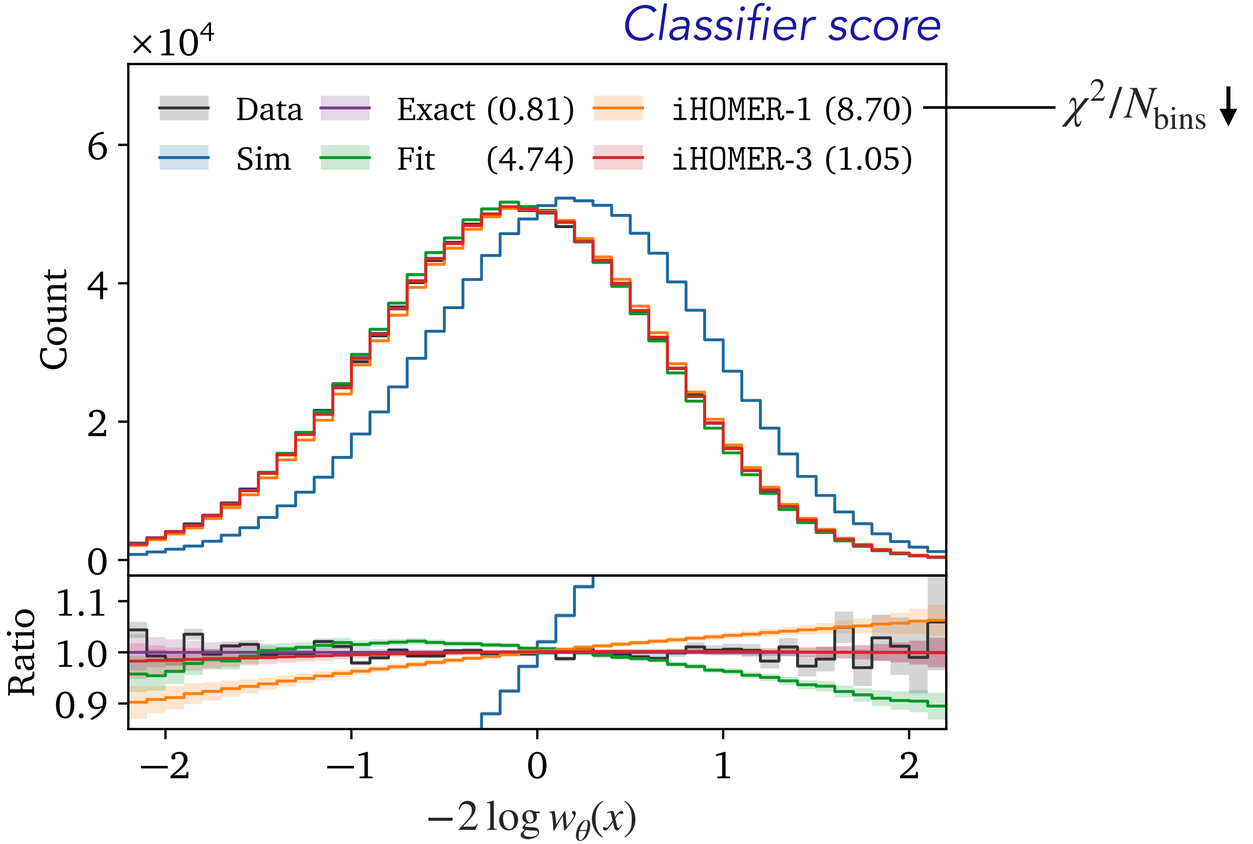
- *Example case: Hadronize $q\bar{q}$ strings at $\sqrt{s} = 90$ GeV*

'***Sim'***: *Standard reference simulation* '***Data'***: *sampled from mixture model:* $z \sim \frac{1}{2} f_1(z \mid m_T^2) + \frac{1}{2} f_2(z \mid m_T^2)$

Event shape observable



Classifier score

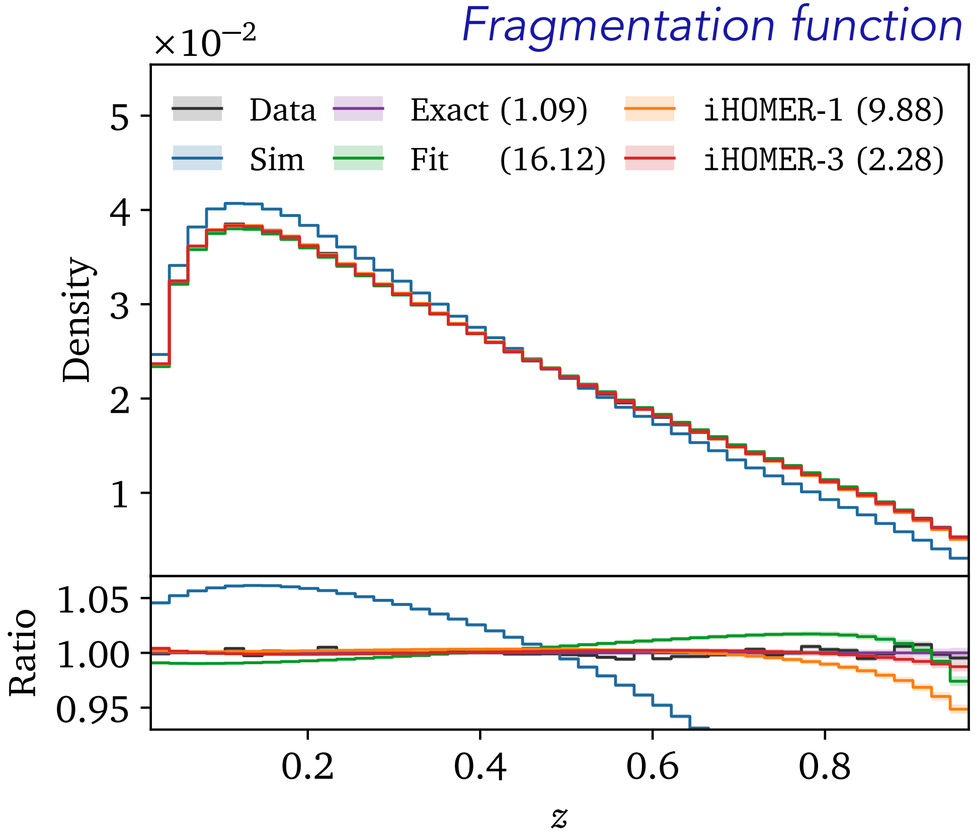


[Butter, AO et. al 2509.03592]

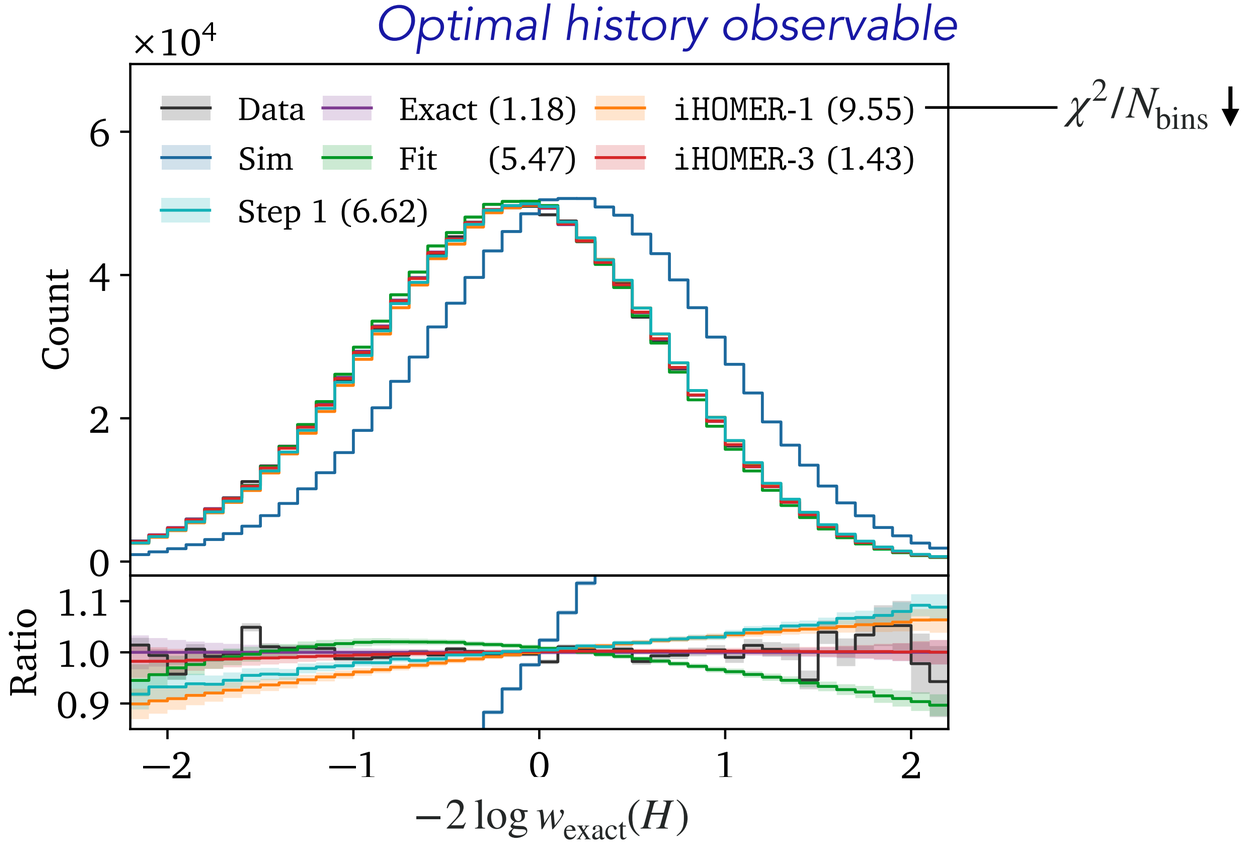
Reweighting accuracy: Fragmentations

- *iHOMER also corrects fragmentation!* ⇒ *Extracts underlying model*

Fragmentation function

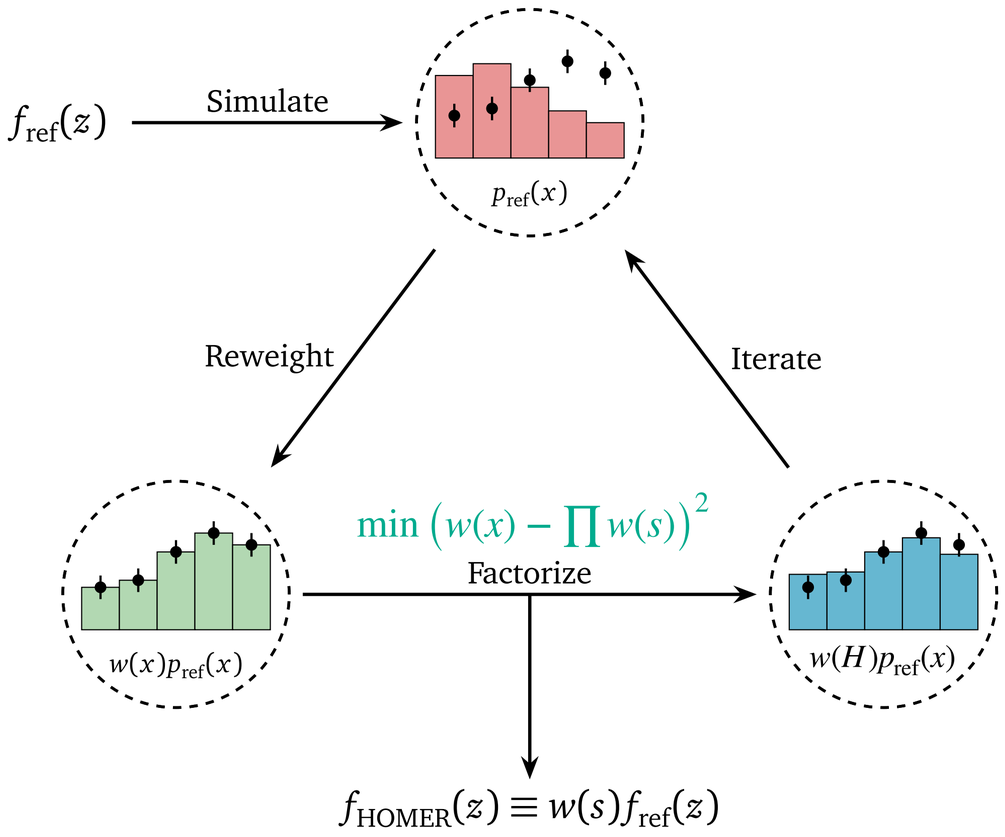


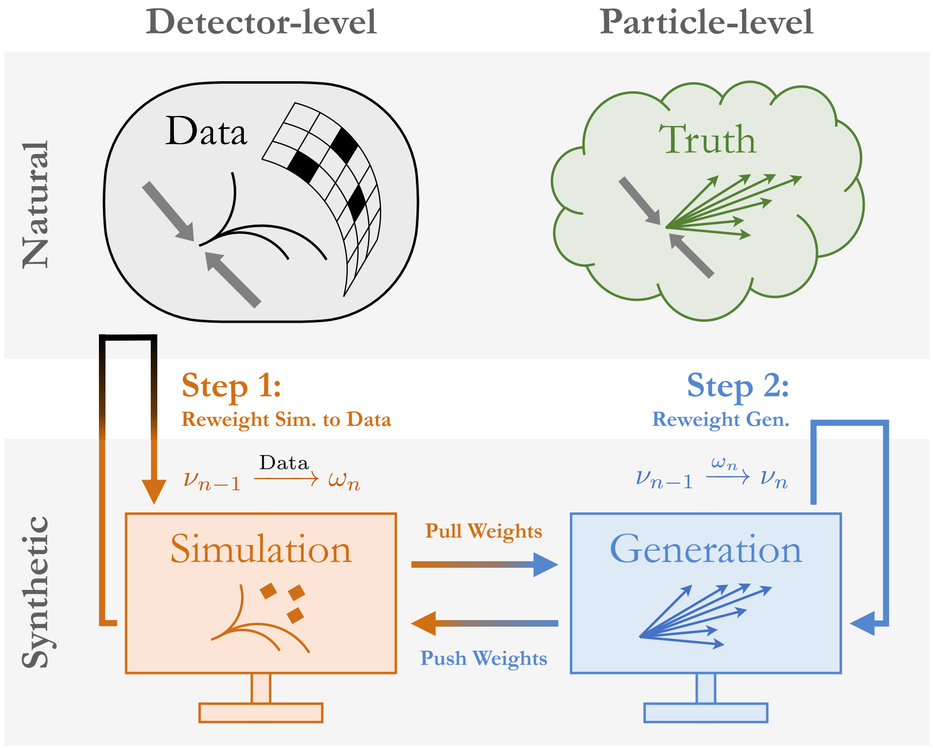
Optimal history observable



[Butter, AO et. al 2509.03592]

Isn't that OmniFold?





OmniFold [Andreassen, et al.1911.09107]

- *Yes! Unfolding also seeks to **infer a latent distribution** compatible with data*
- *But can we do better than iterations?*

Unfolding without Iterations, Adversaries or Surrogates

Goal

$$w_\theta(x) \xleftrightarrow{\text{Match}} \overline{w}_\phi(x) \equiv \int dH\, p_{\text{ref}}(H\,|\,x)\, w_\phi(H)$$

OmniFold

$$\min \Big(w_\theta(x) - w_\phi(H) \Big)^2 \quad \textit{+ Iterations}$$

[AO and T. Plehn, 2602.24282]

Unfolding without Iterations, Adversaries or Surrogates

Goal

$$w_\theta(x) \xleftrightarrow{\text{Match}} \overline{w}_\phi(x) \equiv \int dH\, p_{\text{ref}}(H\,|\,x)\, w_\phi(H)$$

OmniFold

$$\min \Big(w_\theta(x) - w_\phi(H) \Big)^2 \quad \textit{+ Iterations}$$

- *We define a weighted* ***residual function:*** $G(x) \equiv \Big(w_\theta(x) - \overline{w}_\phi(x) \Big)\, p_{\text{ref}}(x)$

[AO and T. Plehn, 2602.24282]

Unfolding without Iterations, Adversaries or Surrogates

Goal

$$w_\theta(x) \xleftrightarrow{\text{Match}} \overline{w}_\phi(x) \equiv \int dH\, p_{\text{ref}}(H\,|\,x)\, w_\phi(H)$$

OmniFold

$$\min \Big(w_\theta(x) - w_\phi(H) \Big)^2 \quad \text{+ Iterations}$$

- *We define a weighted* ***residual function:*** $G(x) \equiv \Big(w_\theta(x) - \overline{w}_\phi(x) \Big)\, p_{\text{ref}}(x)$

- G *itself is not tractable, but its* ***kernel norm*** *is! Minimize:*

$$\|G\|_K^2 \equiv \int dx dx' G(x) K(x,x') G(x')$$

Just require K 'characteristic' (e.g. Gaussian)

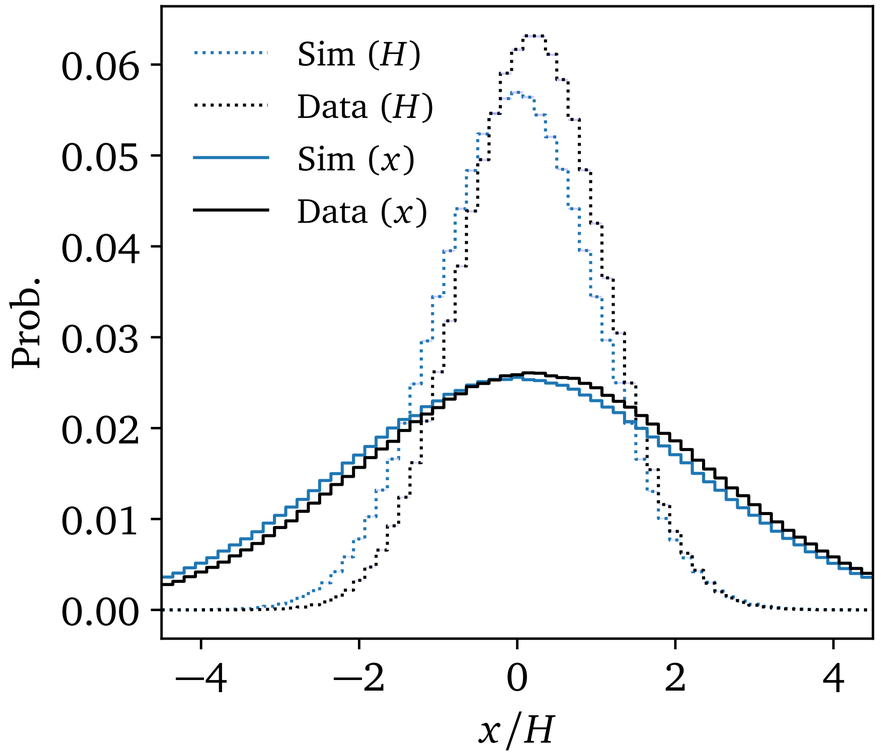
$$= \mathbb{E}_{p_{\text{ref}}(x,H) p_{\text{ref}}(x',H')} \Big[\Big(w_\theta(x) - w_\phi(H) \Big) K(x,x') \Big(w_\theta(x') - w_\phi(H') \Big) \Big]$$

- ***No Iterations, no surrogate, no GAN:*** ***A****dversary-free* ***U****nfolding* ***S****an****S*** ***I****teration or* ***E****mulation (****AUSSIE****)*

[AO and T. Plehn, 2602.24282]

Gaussian example

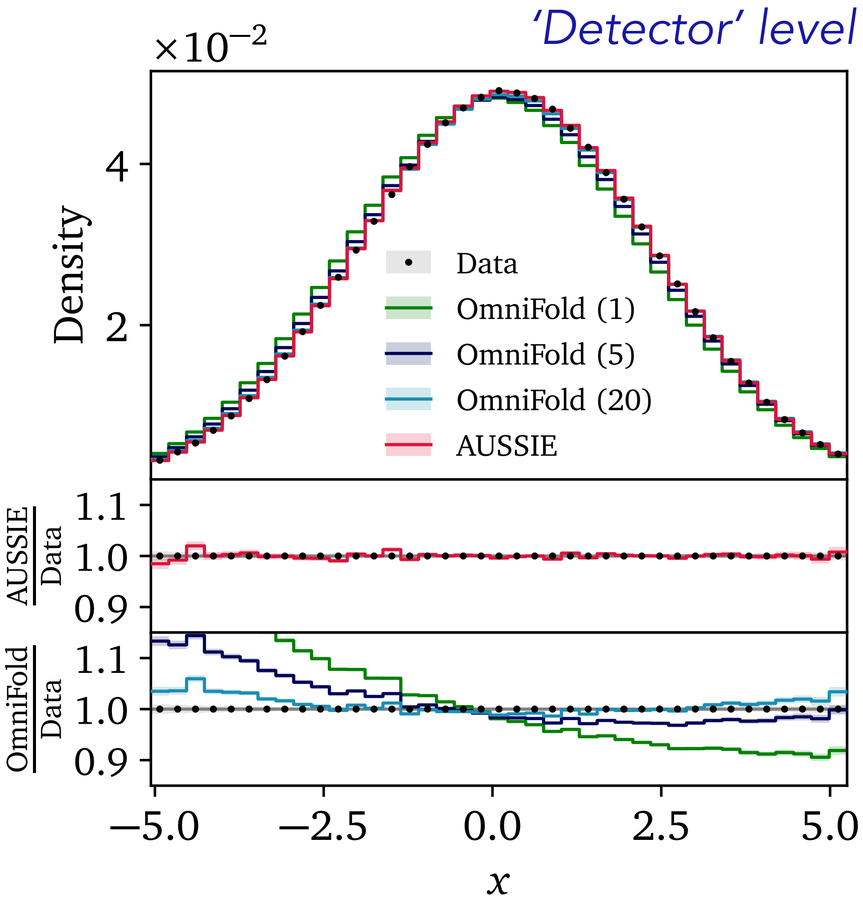
- *Toy model with large smearing exemplifies advantage of AUSSIE*

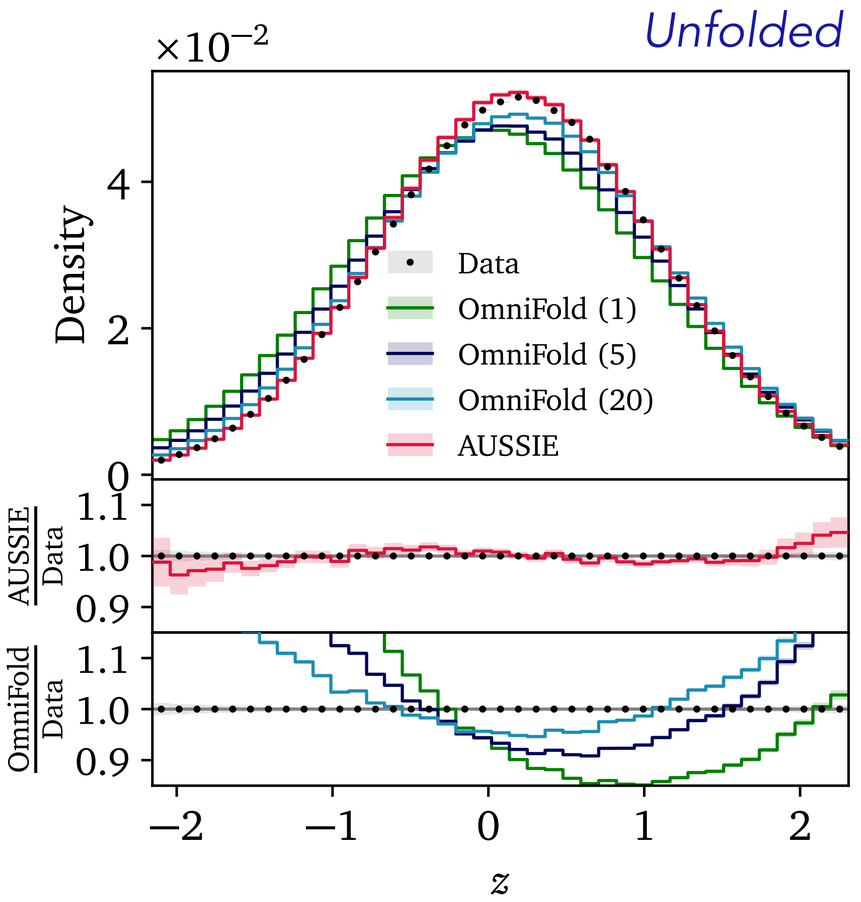


[AO and T. Plehn, 2602.24282]

Gaussian example

- *Toy model with large smearing exemplifies advantage of AUSSIE*



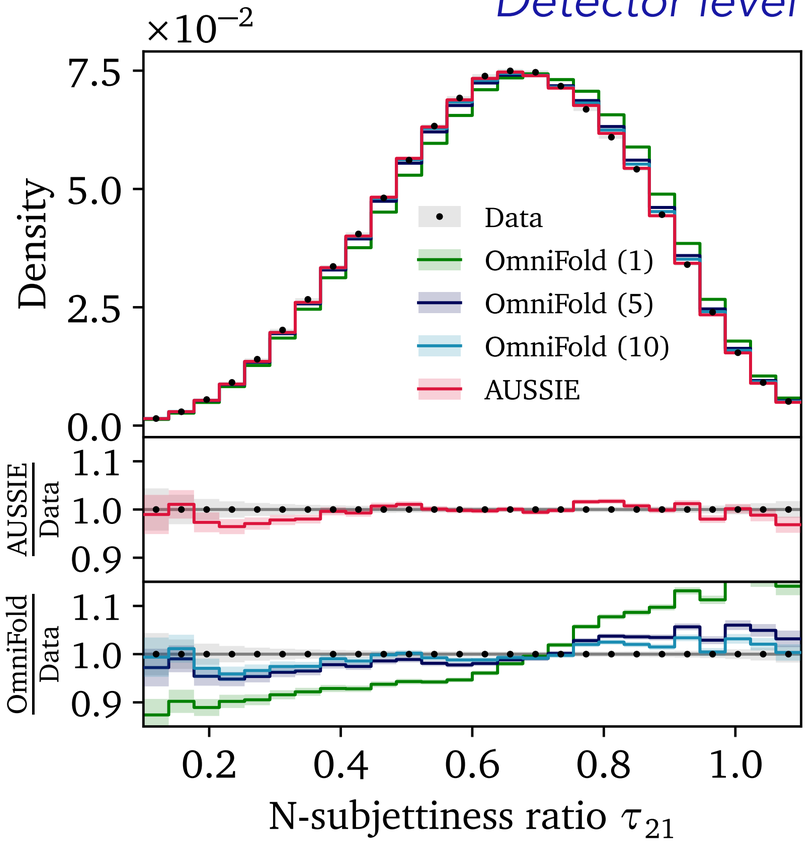


[AO and T. Plehn, 2602.24282]

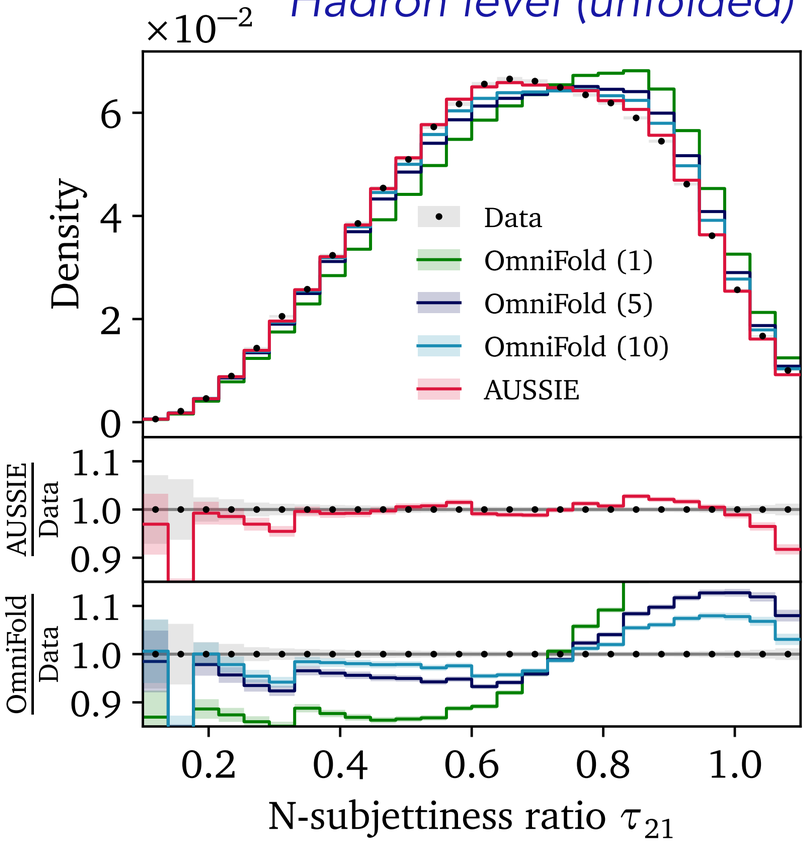
Jet substructure (high-level)

- ***Classic benchmark dataset:*** *6D unfolding problem*
- *Unit Gaussian kernel after preprocessing (no tuning)*
- *AUSSIE matches while OmniFold struggles*

Detector level

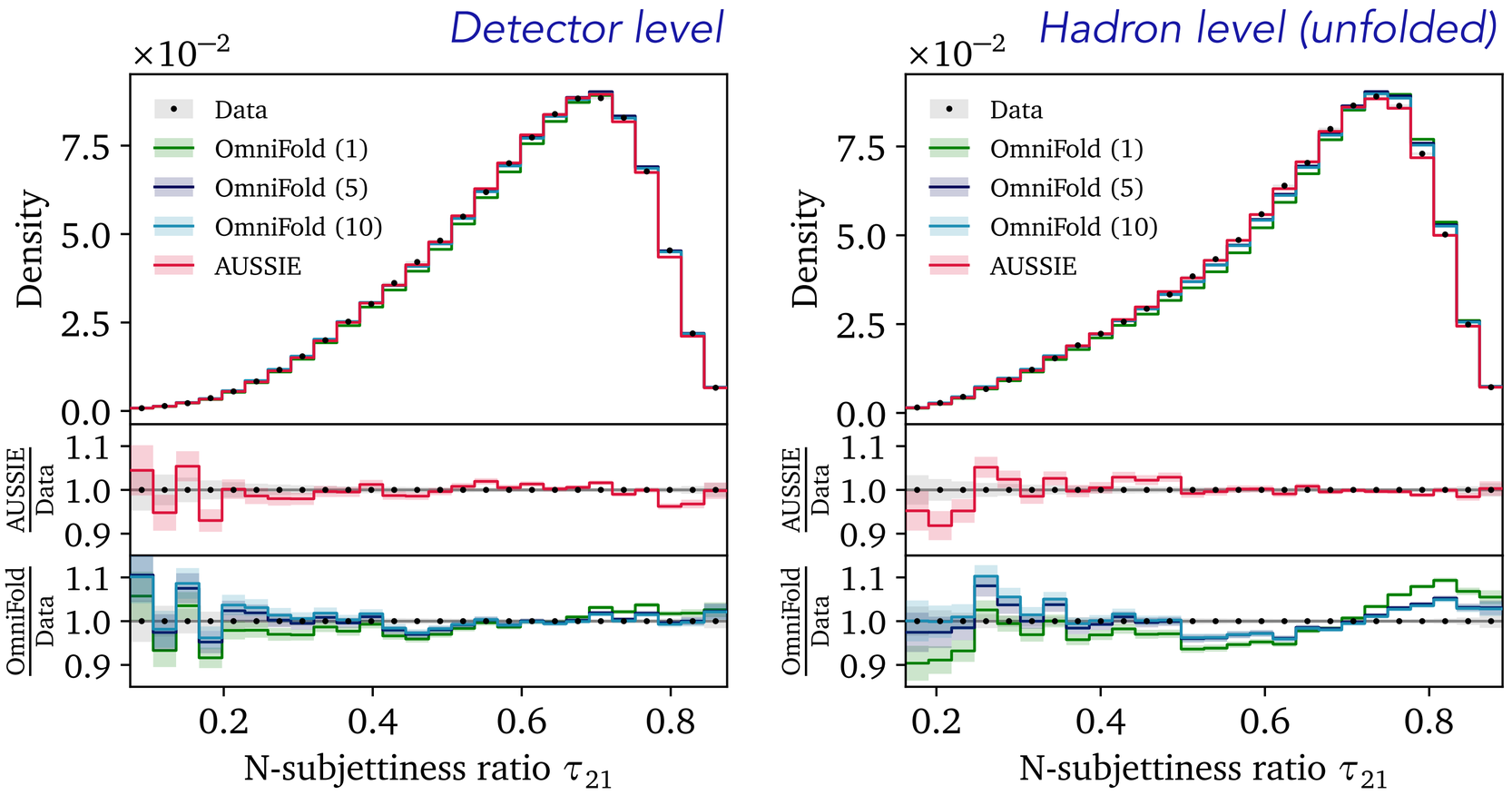


Hadron level (unfolded)



[AO and T. Plehn, 2602.24282]

Jet substructure (constituent-level)



- ***Full phase space unfolding:*** *particle cloud up to N=150*
- *Adaptive kernel from* ***Neural Tangent Kernel*** *of step-1 classifier:*

$$K(x, x') = \left(\nabla_\theta w_\theta(x) \right)^T \nabla_\theta w_\theta(x')$$

- *AUSSIE avoids residual bias toward simulated dataset*

[AO and T. Plehn, 2602.24282]

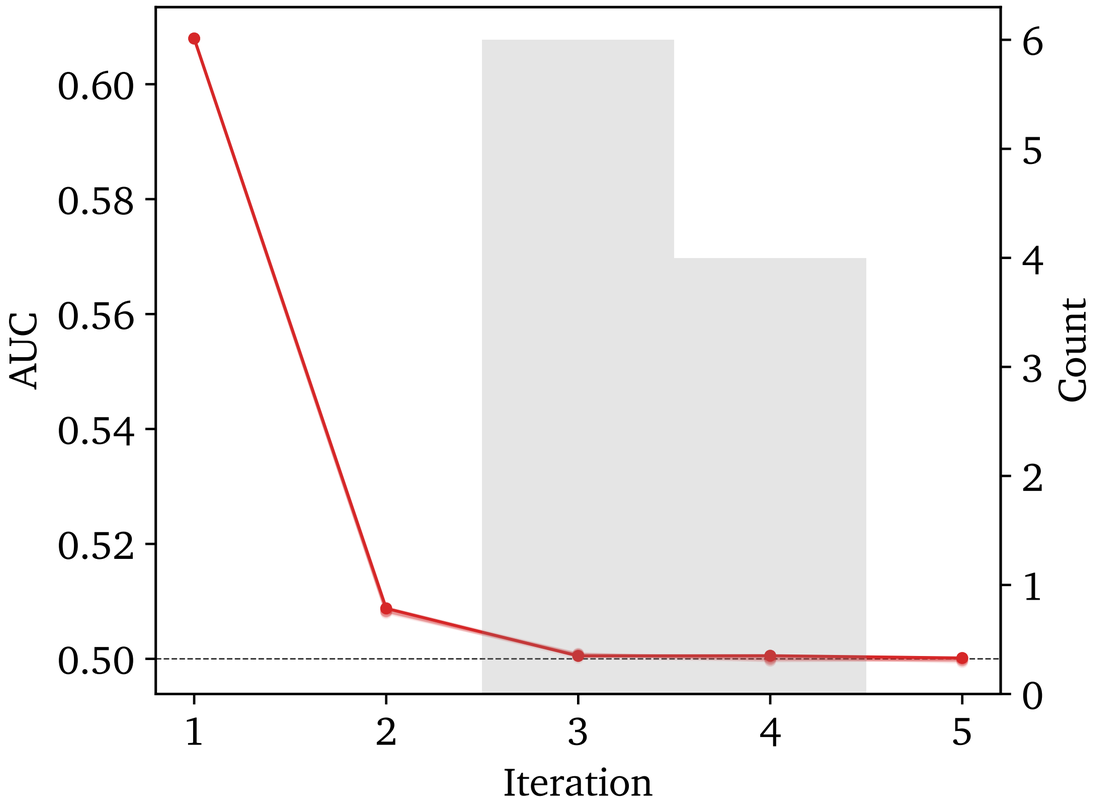
Summary

- *Reweighting is a strong solution for inferring latent distributions*
  - ✓ *Likelihood-free*
  - ✓ *Bypasses detector gradients*
  - ✓ *Leverage structure of reference simulation*
  - ✓ *Marginalisation by iteration or **AUSSIE***
- ***HOMER** extracts string fragmentation functions from data* [2509.03592]

  Still to explore:
  - *How much is HOMER limited by the string picture?*
  - *To what degree can non-fragmentation effects be absorbed?*
- ***AUSSIE** provides a direct, non-iterative unfolding with minimal bias from simulation* [2602.24282]
  - *Readily extendible to account for efficiency / acceptance effects*

Bonus: iHOMER Stopping condition

- Track AUC (with BNN uncertainty) over iterations
- ***Stop once AUC=0.5 is within uncertainty***



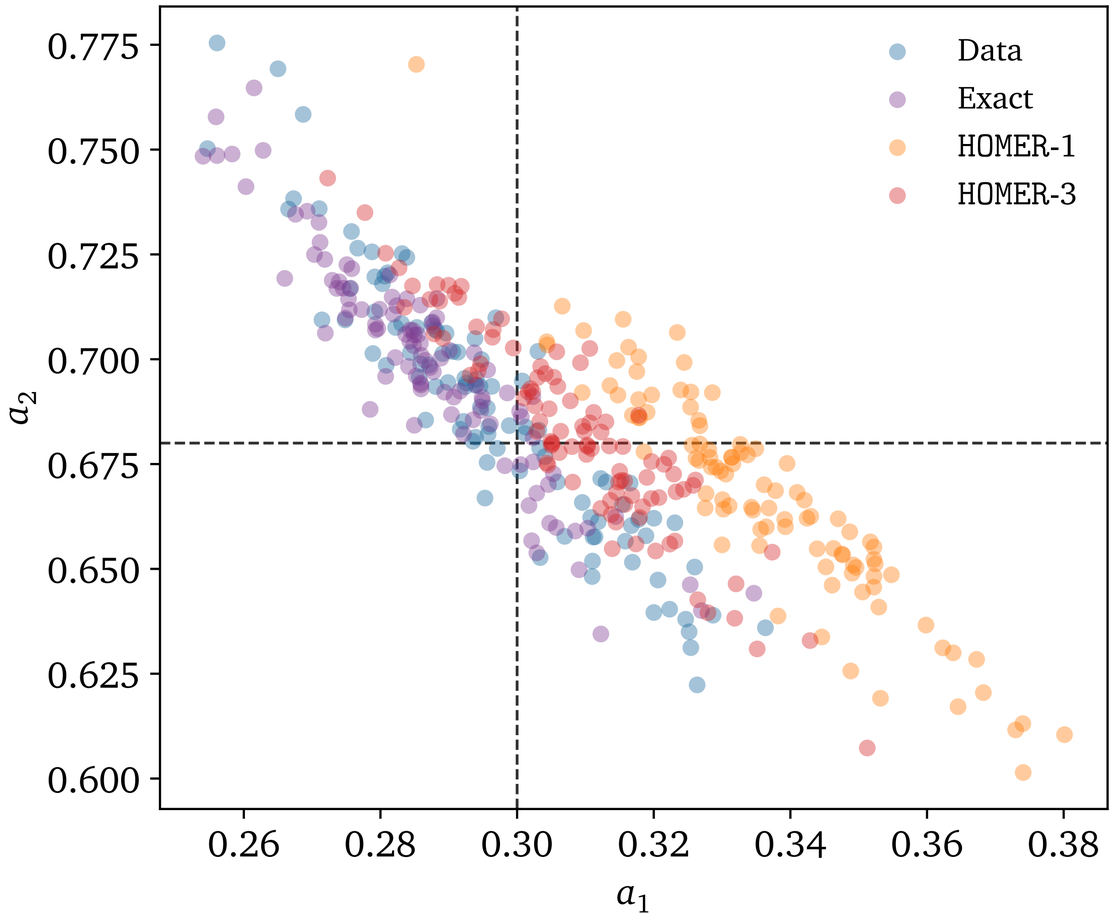
Bonus: iHOMER Parametric fit

- In addition to matching event observables, HOMER defines a fragmentation function:

$$f_{\text{HOMER}}(z \,|\, m_T) \equiv w(s) f_{\text{ref}}(z \,|\, m_T)$$

- Using an ensemble of bootstrapped MLEs, we find that a fit recovers parameters consistent with the true mixture model

- More generally, $f_{\text{HOMER}}(z \,|\, m_T)$ can be used to directly test some fragmentation model of interest



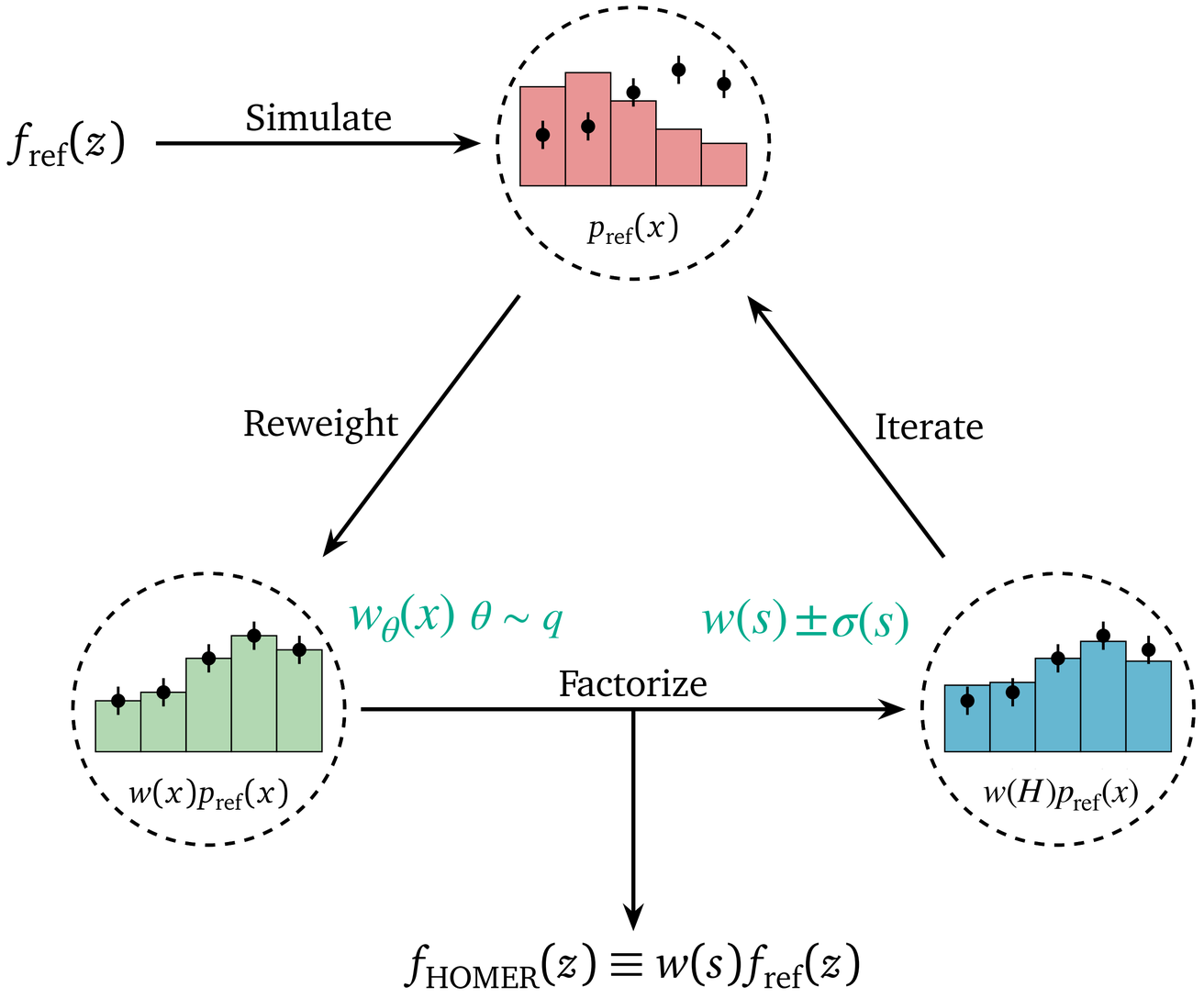
Bonus: iHOMER Uncertainties

- Step 1 already predicts probabilities
  ⇒ Only treat statistical uncertainty:

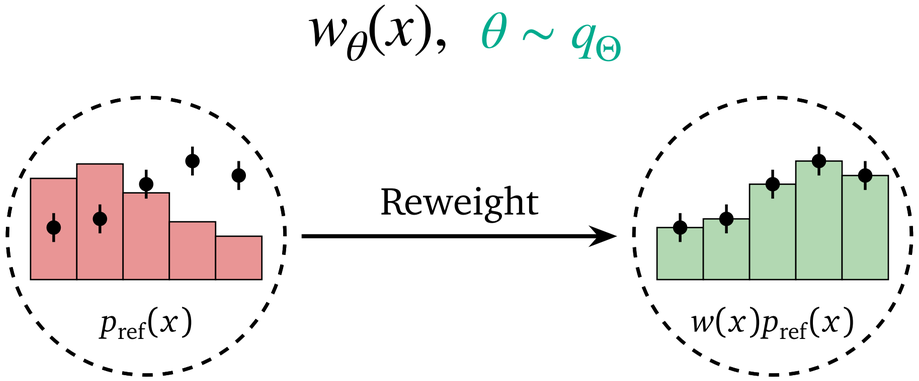
  Bayesian Neural Network

- Step 2 only uses simulation.
  ⇒ Can ignore statistical uncertainty:

  Gaussian regression



Bonus: iHOMER Bayesian Neural Network



- Make variational approximation $q_\Theta(\theta) \approx p_{\text{posterior}}(\theta \,|\, \mathscr{D}_{\text{train}})$
  (Typically independent Gaussians)
- Train above condition by KL divergence:

$$\mathscr{L}(\Theta) = D_{\text{KL}}\left[q_\Theta, p_{\text{posterior}}\right]$$

$$= D_{\text{KL}}\left[q_\Theta, p_{\text{prior}}\right] - \left\langle p(\mathscr{D}_{\text{train}} \,|\, \theta)\right\rangle_{\theta \sim q_\Theta}$$

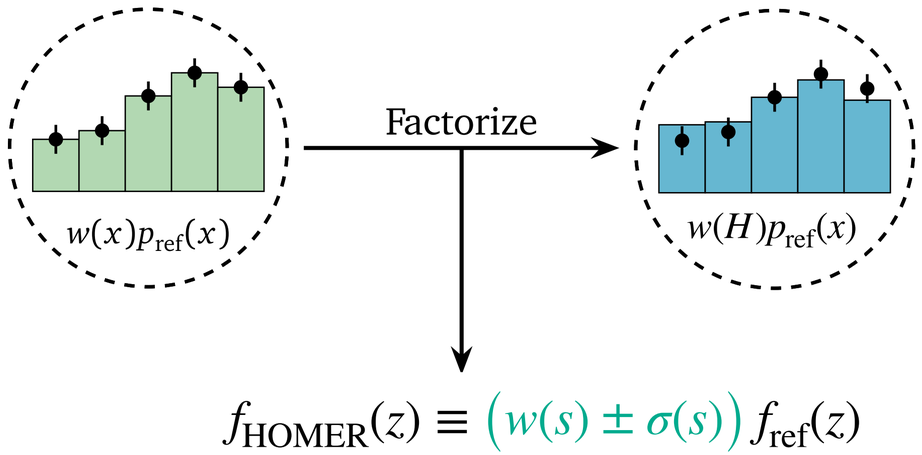
e.g. Classification loss

- After training, we can estimate uncertainties by sampling

$$\left\{f(\theta)\right\}, \theta \sim q_\Theta$$

where f is any function of the network.

Bonus: iHOMER Gaussian regression



- Predict mean and variance for break-level log-weight:

$$\ln w(s) \to \ln w(s) \pm \sigma(s)$$

- Propagate mean and variance to chain level:

$$\ln w(H) \to \ln w(H) \pm \sigma(H) \qquad \text{with} \qquad \sigma^2(H) \equiv \sum_{s \in H} \sigma^2(s)$$

- Fit step-one log-weight with Gaussian likelihood:

$$\mathscr{L} = -\left\langle \ln \mathcal{N}\left(\underbrace{\ln w_\theta(x)}_{\text{Target}} ; \underbrace{\ln w(H)}_{\text{Mean}}, \underbrace{\sigma(H)}_{\text{Uncertainty}}\right)\right\rangle_{p_{\text{ref}}(x,H),\, q_\Theta(\theta)}$$

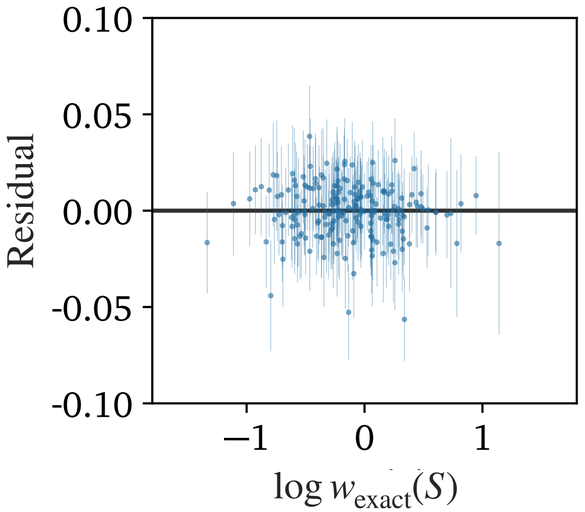
Bonus: Uncertainty Calibration

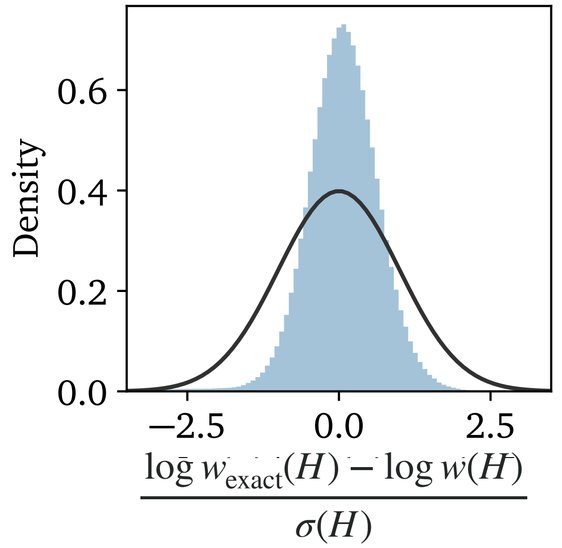
- Centred but narrow pulls on log-weights
- Gaussian for histories, but wilder for breaks
- Actually expected:

  High-level observables cannot fully constrain fragmentation

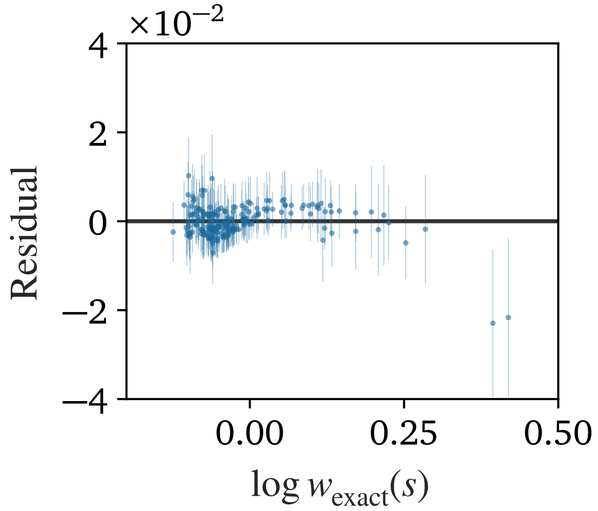
  +

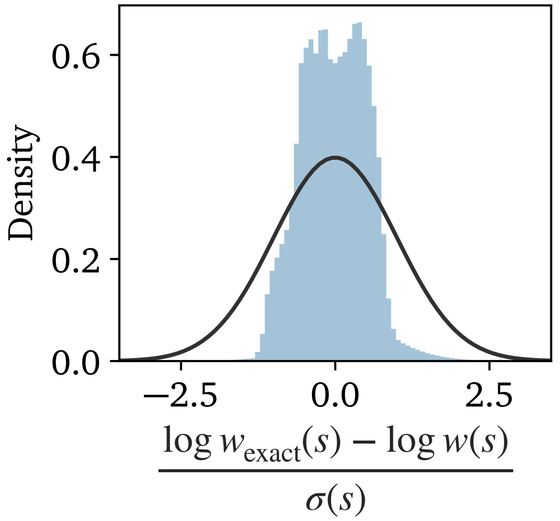
  We evaluate against just one compatible fragmentation model.





History pull

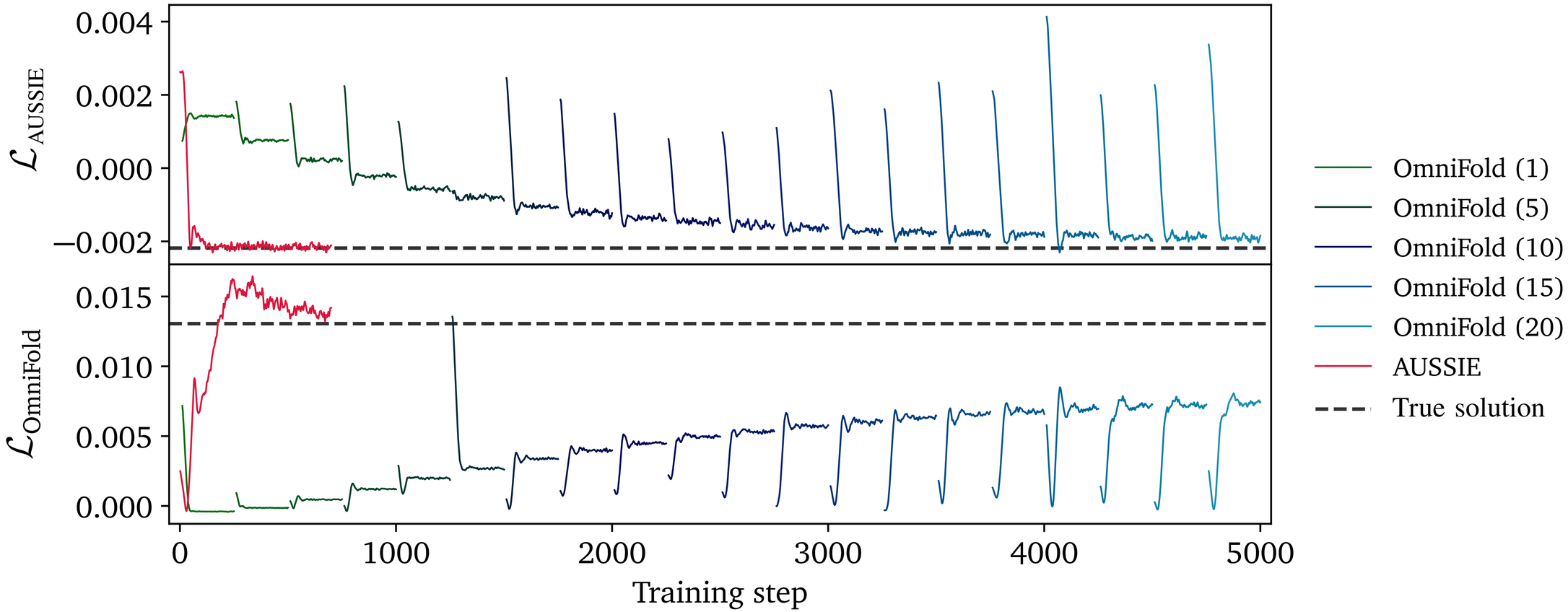




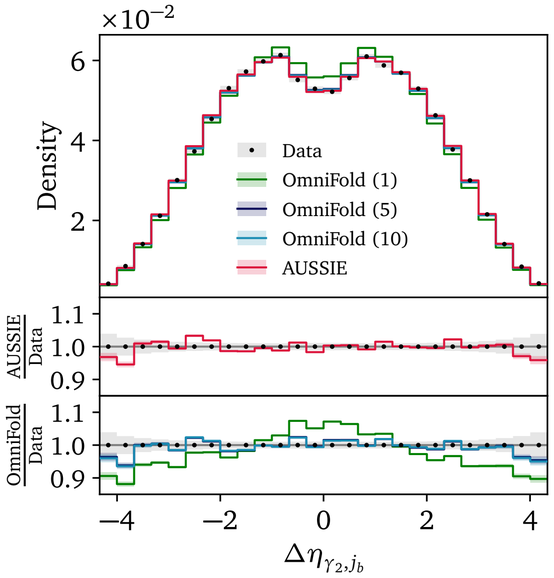
Break pull

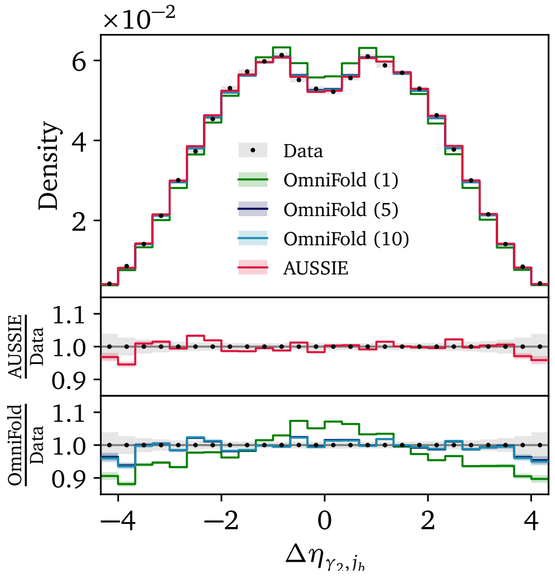
Bonus: Loss traces for Gaussian example

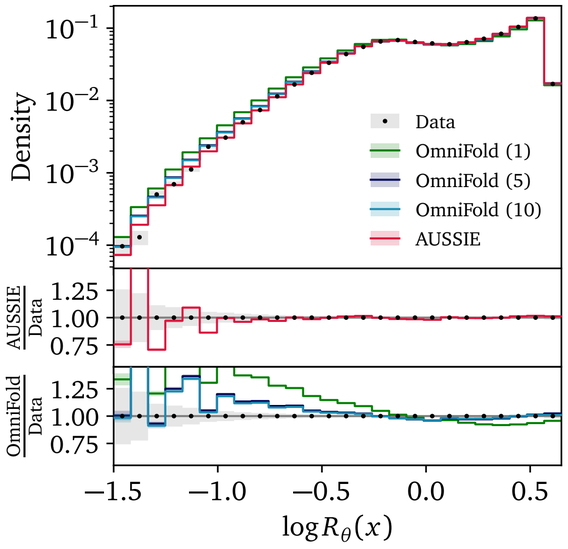
- *AUSSIE loss is minimised by true solution, whereas each iteration of OmniFold is misspecified*



Bonus: Parton-level unfolding







← Detector level

Parton level (unfolded) →

