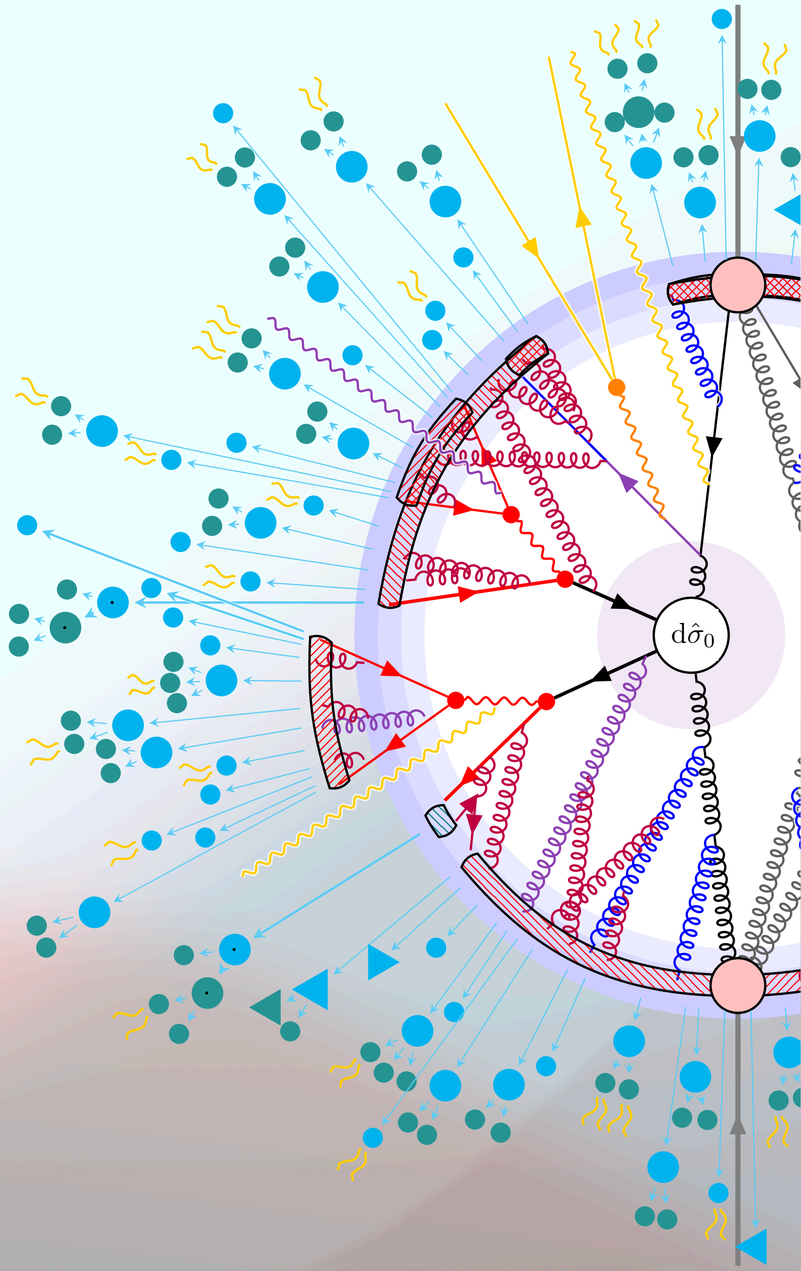
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

ITP

Neural models of hadronization

Ayodele Ore, *AI+HEP in East Asia 2026 @ KEK*



Based on 2509.03592, with:

Anja Butter, Sofia Palacios Schweitzer, Tilman Plehn, Benoît Assi,
Christian Bierlich, Philip Ilten, Tony Menzo, Stephen Mrenna,
Manuel Szewc, Michael K. Wilkinson, Ahmed Youssef and Jure Zupan

Figure courtesy of R. Winterhalder

Simulations at colliders

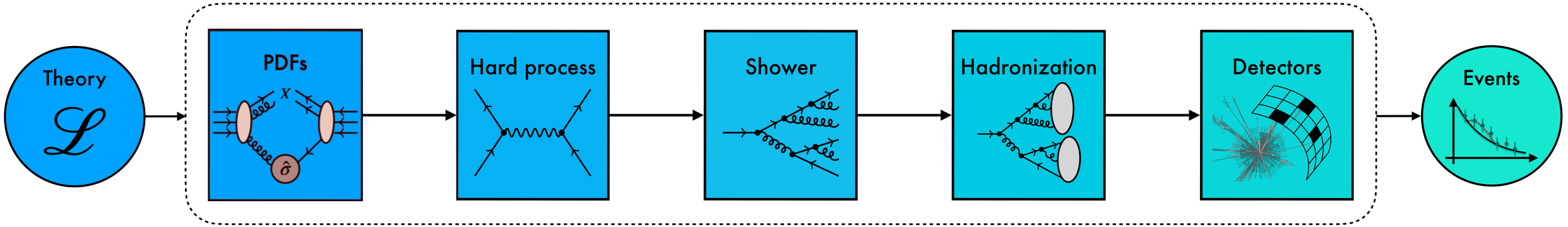
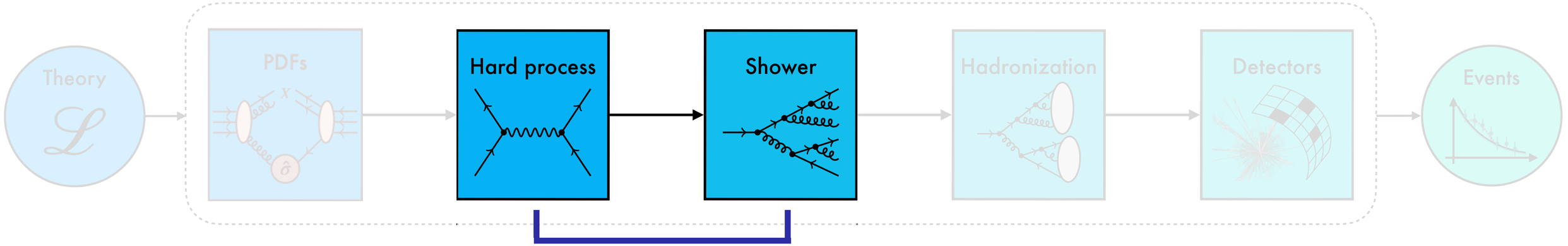


Figure courtesy of R. Winterhalder

Simulations at colliders



Calculable in perturbative Quantum Field Theory

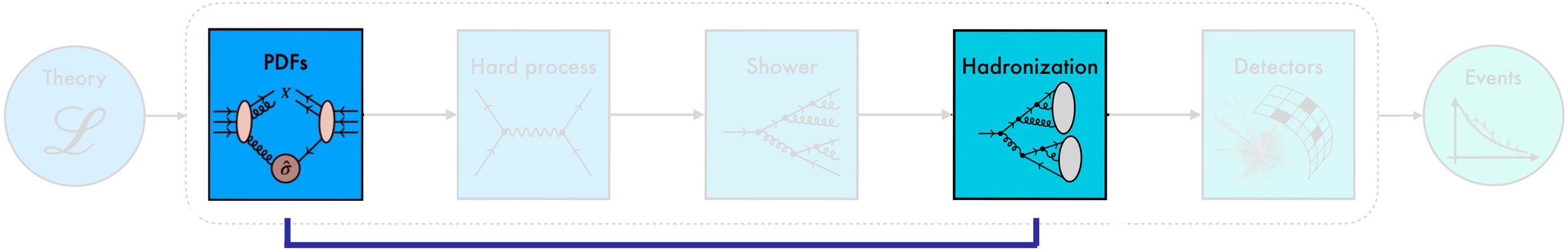
⇓

Highly-accurate

Figure courtesy of R. Winterhalder

Simulations at colliders

- Lund String model (Pythia)
- Cluster model (Herwig)



No first-principles predictions

⇓

Need empirical models

Wishlist for neural hadronization

1. ***Tunable to data*** *such that we actually improve predictions*
2. ***Physically principled*** *for universality and robustness*

} *Precludes end-to-end solutions*

3. ***Calibrated uncertainties***

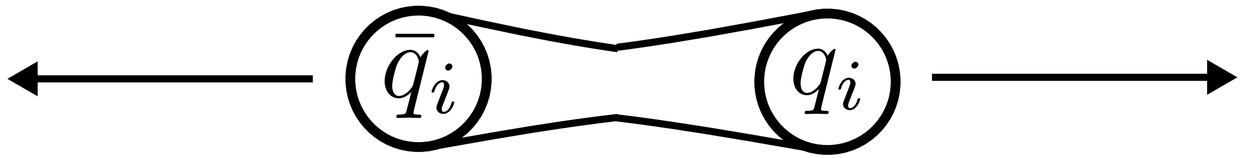
⇒ Bottom-up approach is best: Replace local modules within existing simulators

Two groups working in this direction:

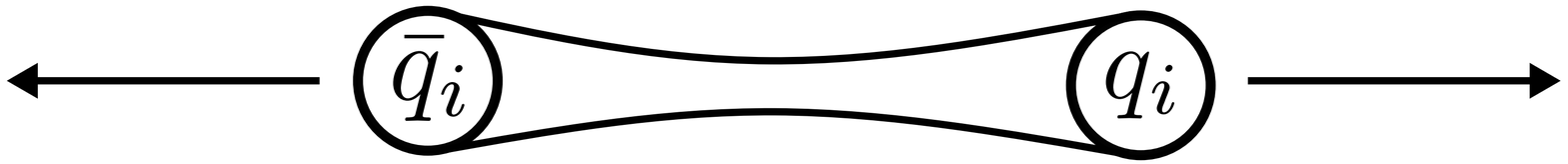
MLHAD *(Lund string model)*
[2203.04983], [2311.09296], [2410.06342], [2503.05667], [2509.03592]

HADML *(Cluster model)*
[2203.12660], [2305.17169], [2312.08453]

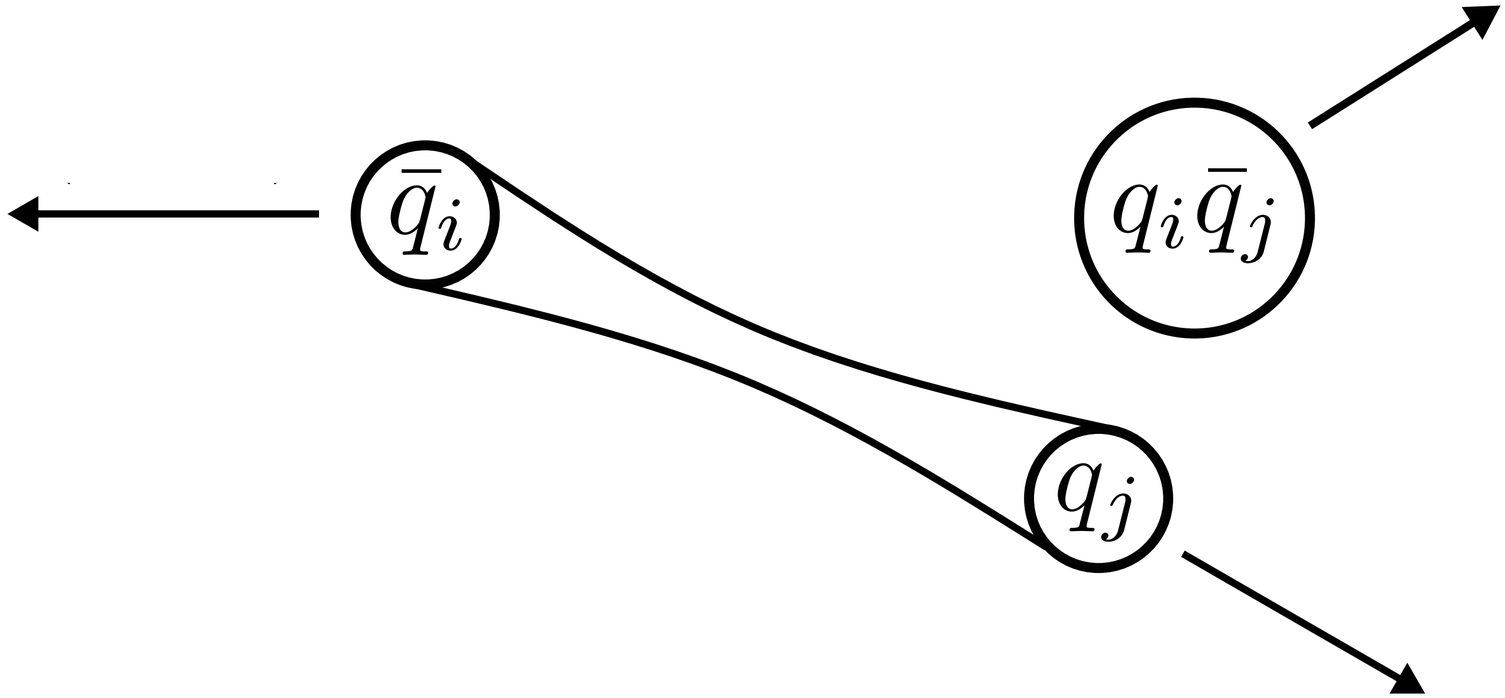
The (simplified) string model



The (simplified) string model



The (simplified) string model



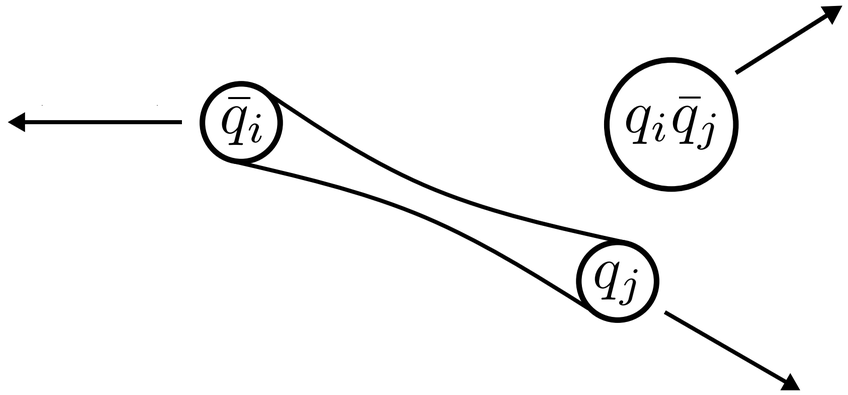
The (simplified) string model

- Characterise string ***breaks*** by

Lightcone momentum fraction

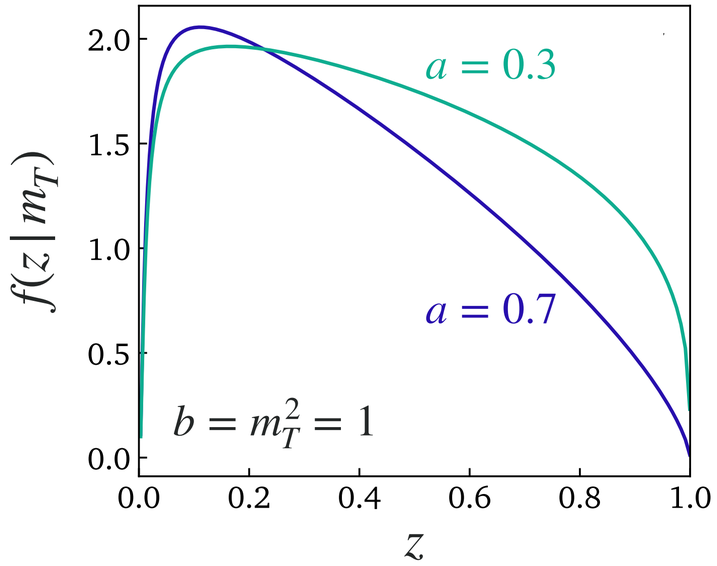
$$s \equiv \{z, m_T\}$$

Transverse mass



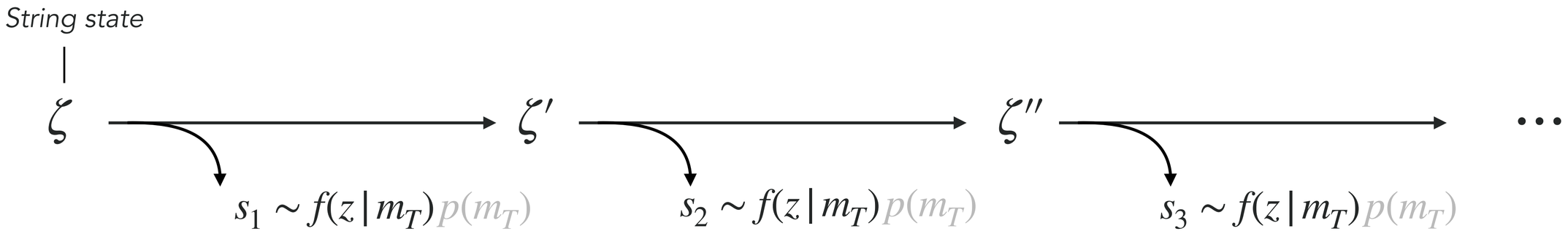
- Sample z from ***fragmentation function***:

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

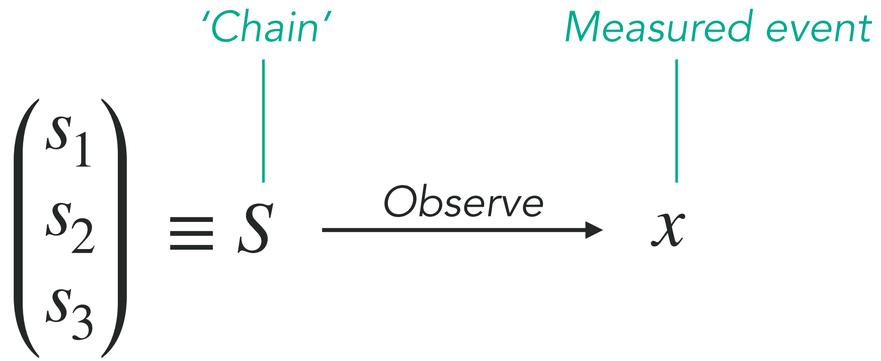


The (simplified) string model

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



- Full chain (over) specifies the final set of hadrons



A latent variable problem

- The full probabilistic model is

Hadronization generator

$$p_\phi(x, S, \zeta) \equiv p(x \mid S)\, p_\phi(S \mid \zeta)\, p(\zeta)$$

Detector simulation

Parton shower

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

- But only x is observable in data… Best you could do is maximise marginal likelihood

$$\max_\phi \sum_{x \sim p_{\text{data}}} p_\phi(x)$$

$$p_\phi(x) = \int dS d\zeta\, p(x \mid S)\, p_\phi(S \mid \zeta)\, p(\zeta)$$

Three challenges

$$\max_{\phi} \sum_{x \sim p_{\text{data}}} p_\phi(x) \qquad\qquad p_\phi(x) = \int dS d\zeta \, p(x \mid S) \, p_\phi(S \mid \zeta) \, p(\zeta)$$

Performing the fit directly requires:

- Tractable likelihoods of each distribution
- Marginalization over S and ζ
- Differentiable detector simulation (or unfolded data)

A GAN-based solution

$$p_\phi(x) = \int dS d\zeta \, p(x \mid S) \, p_\phi(S \mid \zeta) p(\zeta)$$

- Maximum likelihood is just KL divergence. But, ***Jensen-Shannon divergence***…

$$D_{\mathrm{JS}}[p_{\mathrm{data}}, p_\phi] = \frac{1}{2} D_{\mathrm{KL}} \left[p_{\mathrm{data}}, \frac{p_{\mathrm{data}} + p_\phi}{2} \right] + \frac{1}{2} D_{\mathrm{KL}} \left[p_\phi, \frac{p_{\mathrm{data}} + p_\phi}{2} \right]$$

$$= \mathbb{E}_{x \sim p_{\mathrm{data}}} \log C(x; \phi) + \mathbb{E}_{x \sim p_\phi} \log \left(1 - C(x; \phi) \right) + \mathrm{const}. \qquad C(x; \phi) \equiv \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_\phi(x)}$$

- So $p_\phi(S \mid \zeta)$ can be trained in a GAN-like setup. [Louppe et al. 1707.07113, *Chan et al. 2305.17169*]

✓ Likelihood-free

✓ Auto-marginalization

x Need gradients of detector sim (or unfolded data)

x GAN training can be unstable

Reweighted hadronization

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

- Can solve all the difficulties:

  ✓ Likelihood-free ✓ Bypass detector gradients ✓ Physical prior ? marginalisation

- Bonus:
  - Weights integrate seamlessly with event generators
  - Easy to propagate uncertainties

- Two implementations by MLHAD:

  [Bierlich et al. 2305.17169] based on ratio's of normalising flows

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

- This talk: ***iHOMER****:* cheap marginalisation + uncertainties *[Butter, AO et al. 2509.03592]*

HOMER Overview

Goal: Find string-break weights that correct observables in reference set

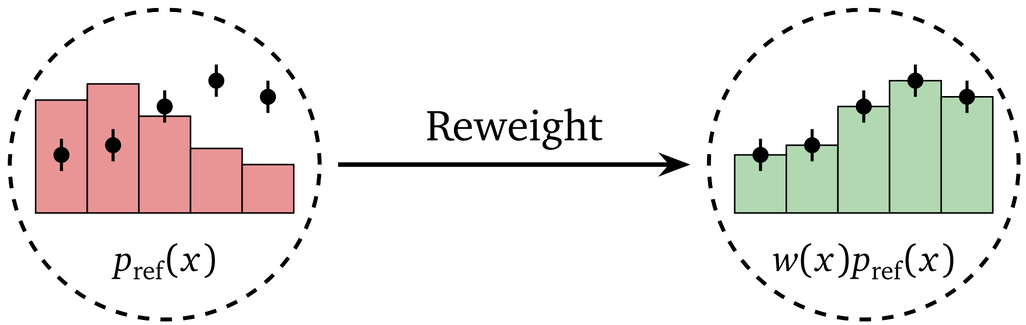
$$p_{\text{ref}}(x) \xrightarrow{w(s)} p_{\text{data}}(x)$$

Result: A data-driven fragmentation function:

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

HOMER: Step 1

$$w(x) \approx p_{\text{data}}(x)/p_{\text{ref}}(x)$$



- *Likelihood ratio trick*:
  - Binary cross-entropy loss

$$\mathscr{L} = \Big\langle \log C(x) \Big\rangle_{p_{\text{data}}(x)} + \Big\langle \log\big(1 - C(x)\big) \Big\rangle_{p_{\text{ref}}(x)}$$

  - Optimum at

$$\delta\mathscr{L} = \int dx \left[\frac{p_{\text{data}}(x)}{C(x)} - \frac{p_{\text{ref}}(x)}{1 - C(x)} \right] \delta C \stackrel{!}{=} 0$$

  - Yields

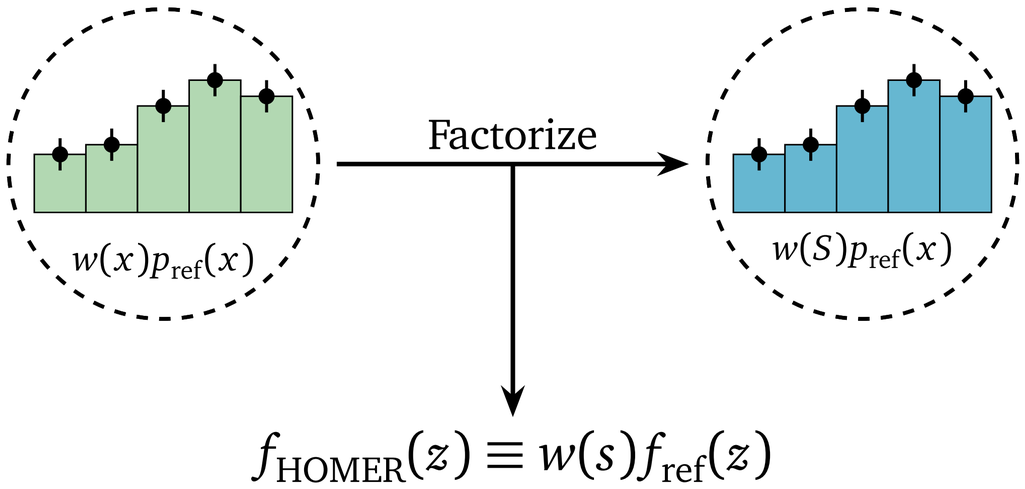
$$\frac{p_{\text{data}}(x)}{p_{\text{ref}}(x)} = \frac{C(x)}{1 - C(x)} \equiv w(x)$$

HOMER: Step 2

- Distill the reference $\longrightarrow$ data correction $w(x)$ into $w(s)$

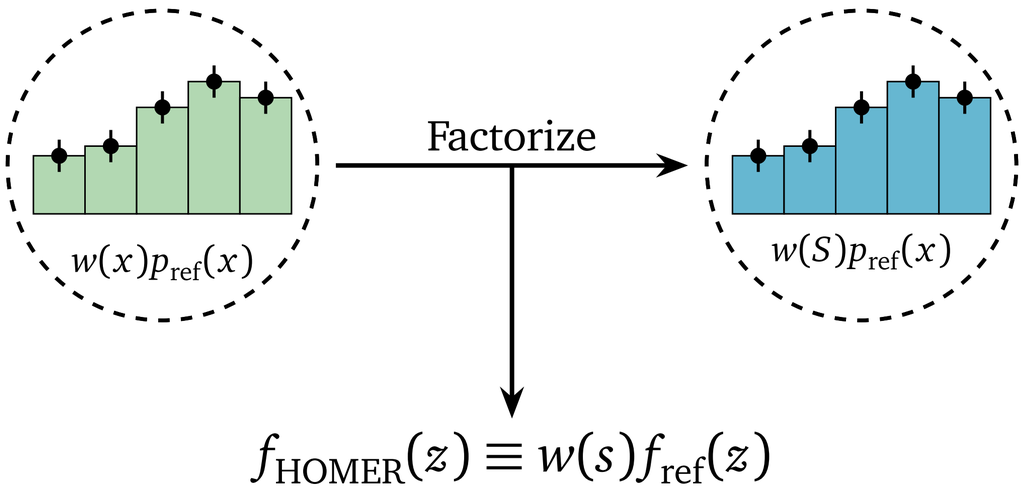
- Exploit *joint* samples from reference set

$$\begin{aligned} x, \quad & S = (s_1, s_2, s_3, s_4) \\ \vdots \quad & \vdots \\ x, \quad & S = (s_1, s_2, s_3, s_4, s_5) \\ x, \quad & S = (s_1, s_2, s_3) \end{aligned} \quad \sim p_{\text{ref}}(x, S)$$



- Assign $w(s)$ to each string break and 'match'

HOMER: Step 2



- Three options to factorise event weight:
  - Minimise error 1-to-1 between chains and events
    [Bierlich et al. 2410.06342]

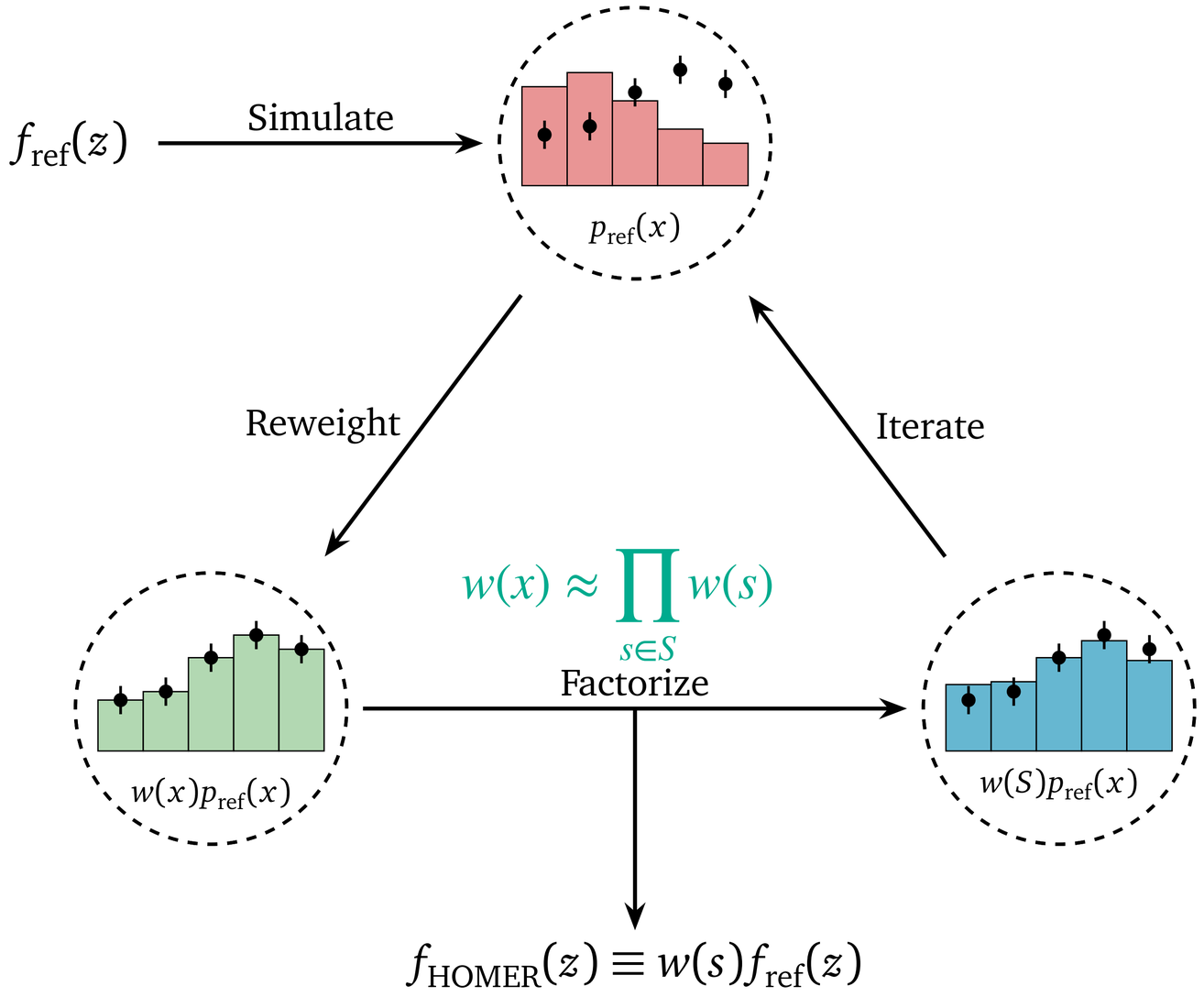
$$w(x) \leftarrow \prod_{s \in S} w(s)$$

  - Explicitly integrate over chains *[Assi et al. 2503.05667]*

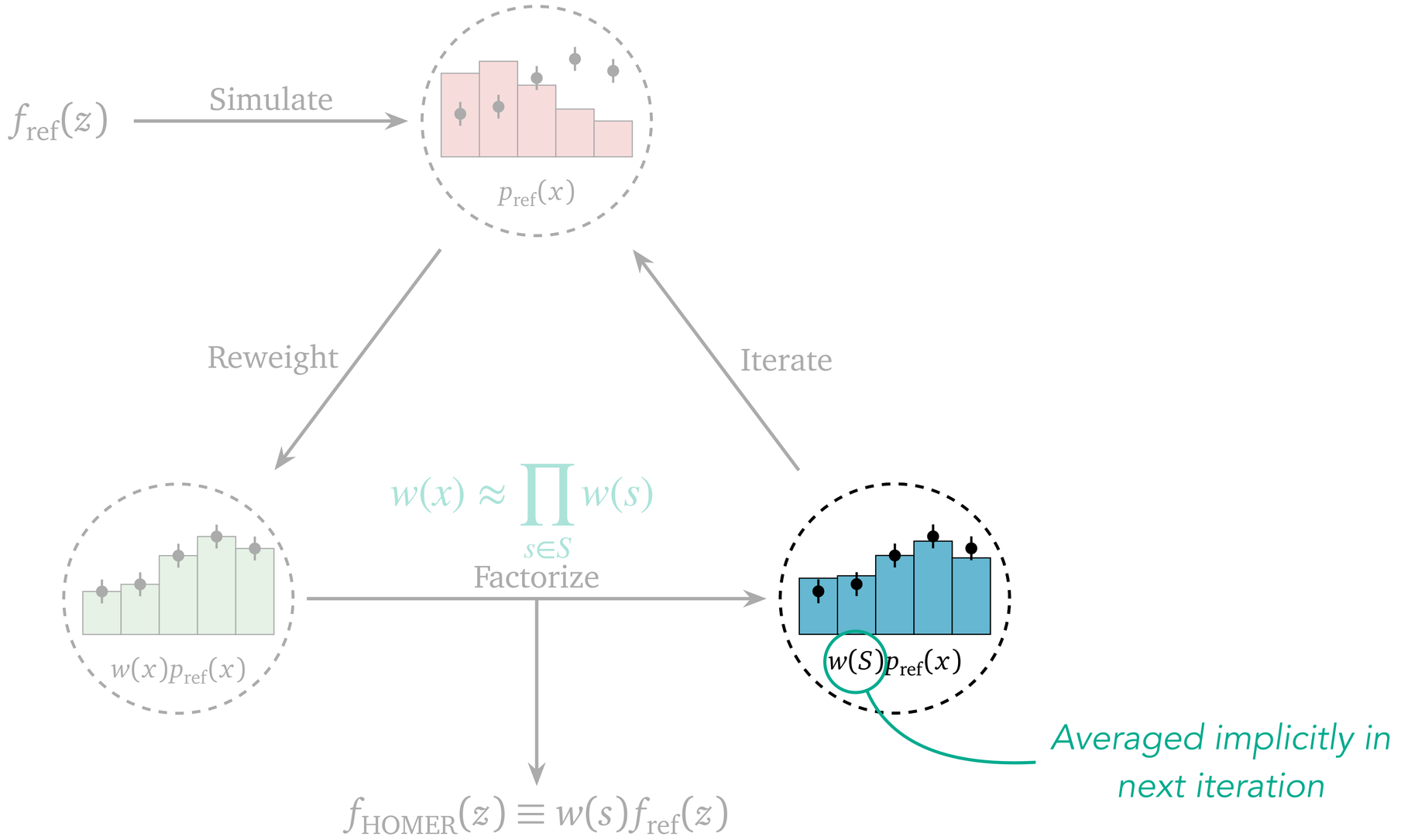
$$w(x) \leftarrow \int dS\, p(S \mid x) \prod_{s \in S} w(s)$$

  - Ignore and ***iterate!*** *(Expectation-Maximisation)*
    [Butter, AO et al. 2509.03592]
    Also OmniFold [1911.09107]

iHOMER: Iterations



iHOMER: Iterations



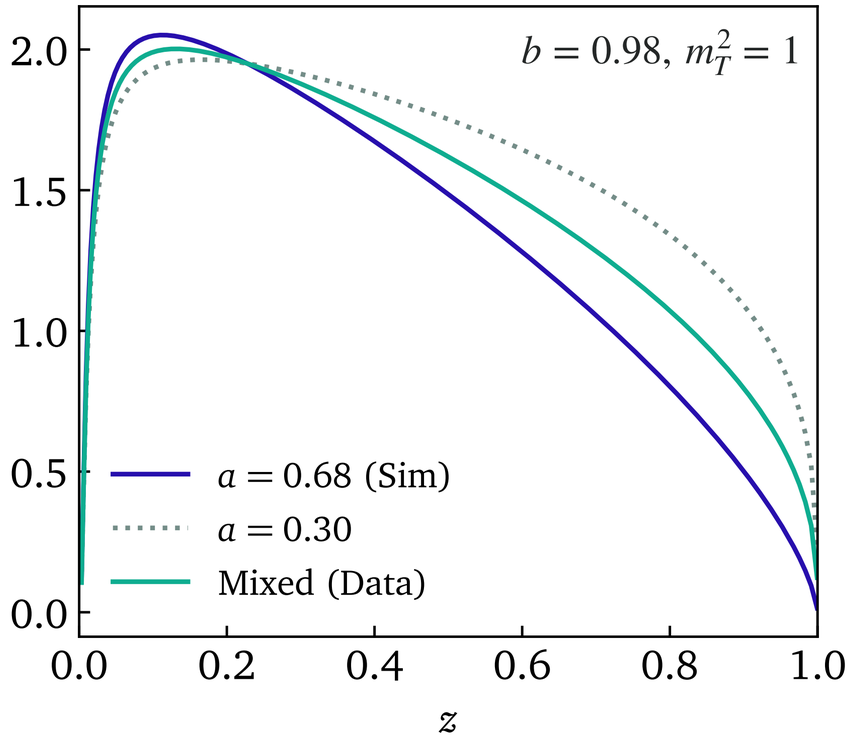
Simulated Datasets

- $q\bar{q}$ string at sqrt(s) = 90GeV
- ***Sim*:** Standard fragmentation

  $z \sim f(z \mid m_T^2)$ w/ **a=0.68**, b=0.98

- ***Data:*** *Mixture fragmentation* *(Cannot fit with standard form)*

  $z \sim \frac{1}{2} f_1(z \mid m_T^2) + \frac{1}{2} f_2(z \mid m_T^2)$ *w/* ***a_1=0.68***, ***a_2=0.30***, *b=0.98*



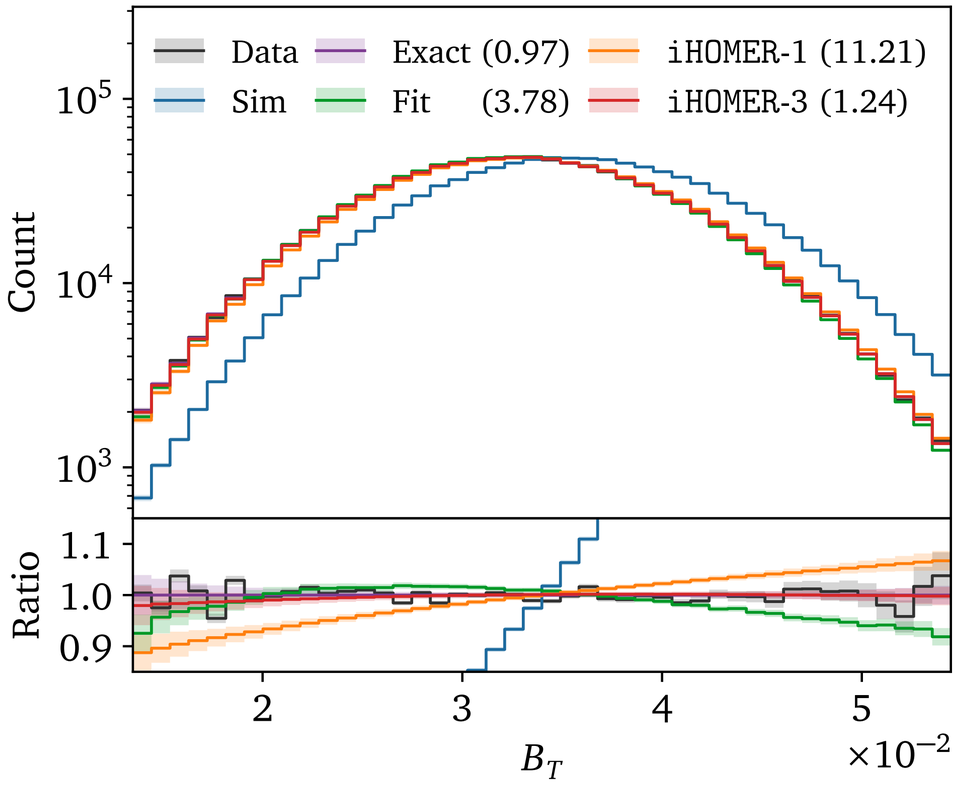
★ Events represented as high-level observables (event shapes, multiplicities, etc…)

★ Simplified scenario: only pions, no gluons, fixed initial state

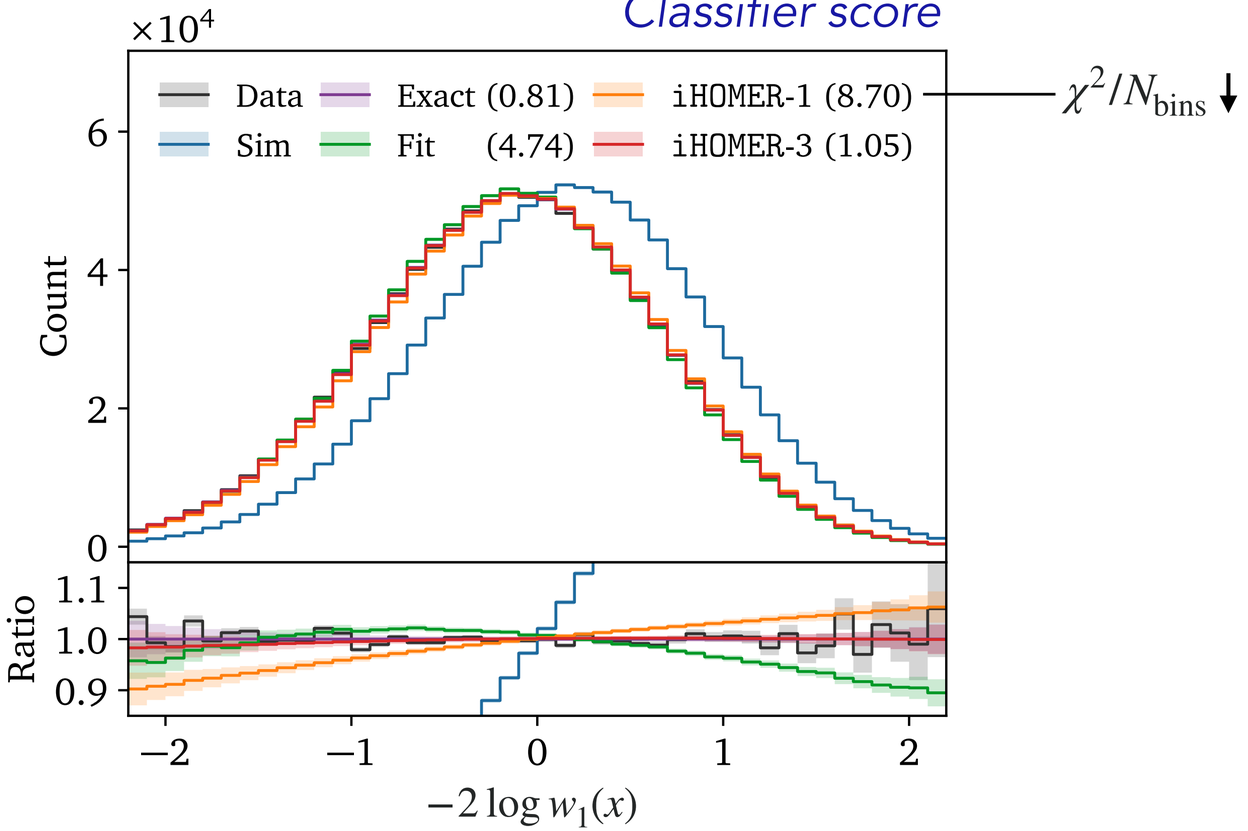
Reweighting accuracy: Observables

- Iterations mitigate bias!
- iHOMER improves on ‘naive’ fit that assumes standard fragmentation

Event shape observable



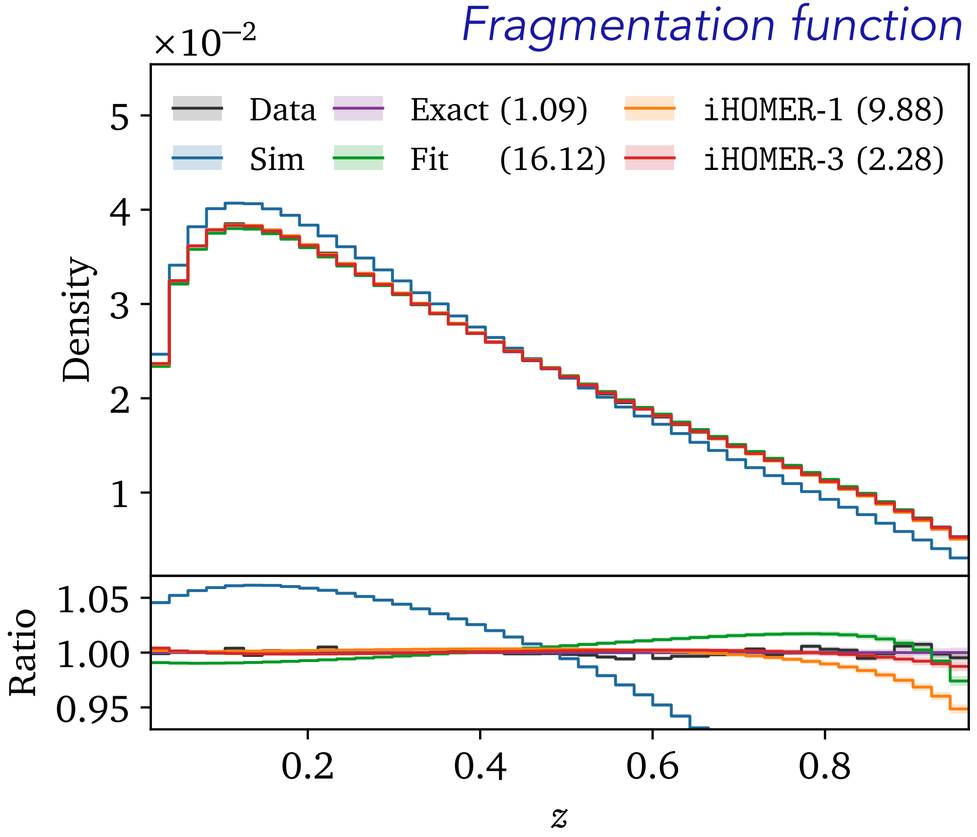
Classifier score



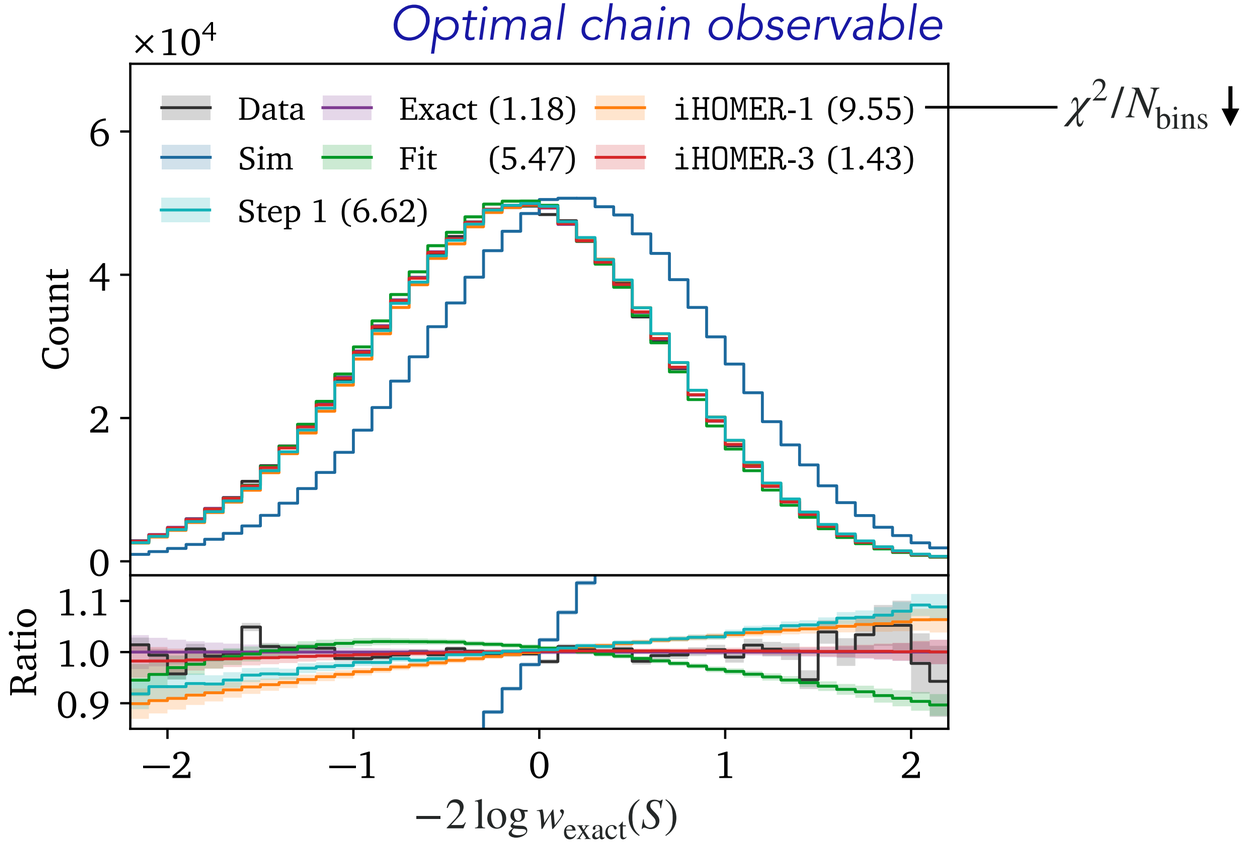
Reweighting accuracy: Fragmentations

- iHOMER also corrects fragmentation! ⇒ Extracts universal physics

Fragmentation function



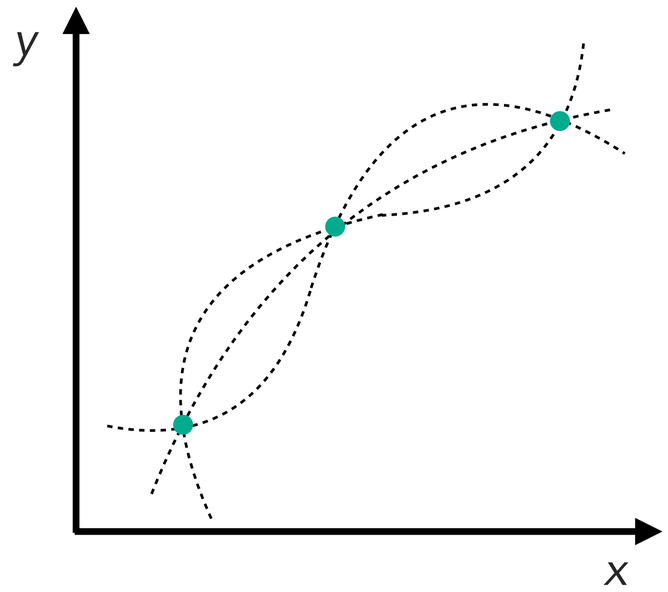
Optimal chain observable



Uncertainties

- Two ways in which ‘well-trained’ models can make errors:

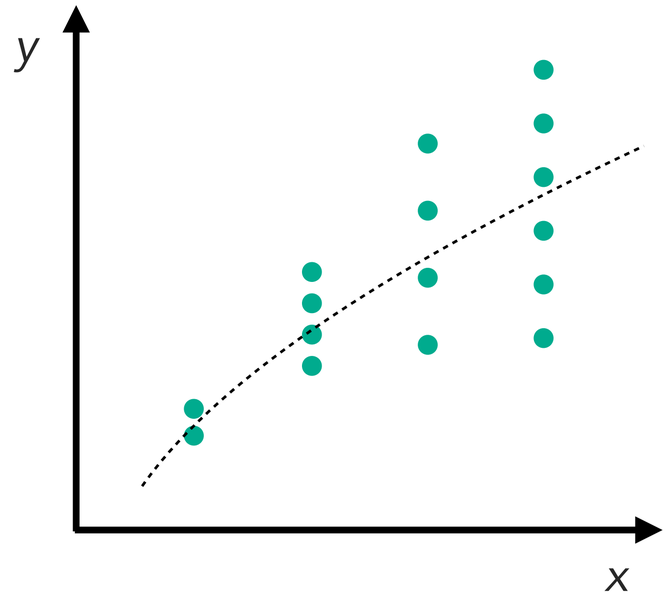
Epistemic *or* ***Statistical*** *uncertainty*



Vanishes given infinite data

To capture: Infer *distribution on network parameters*

Alleatoric *or* ***Systematic*** *uncertainty*



Persists despite infinite data

To capture: Infer *distribution on network outputs*

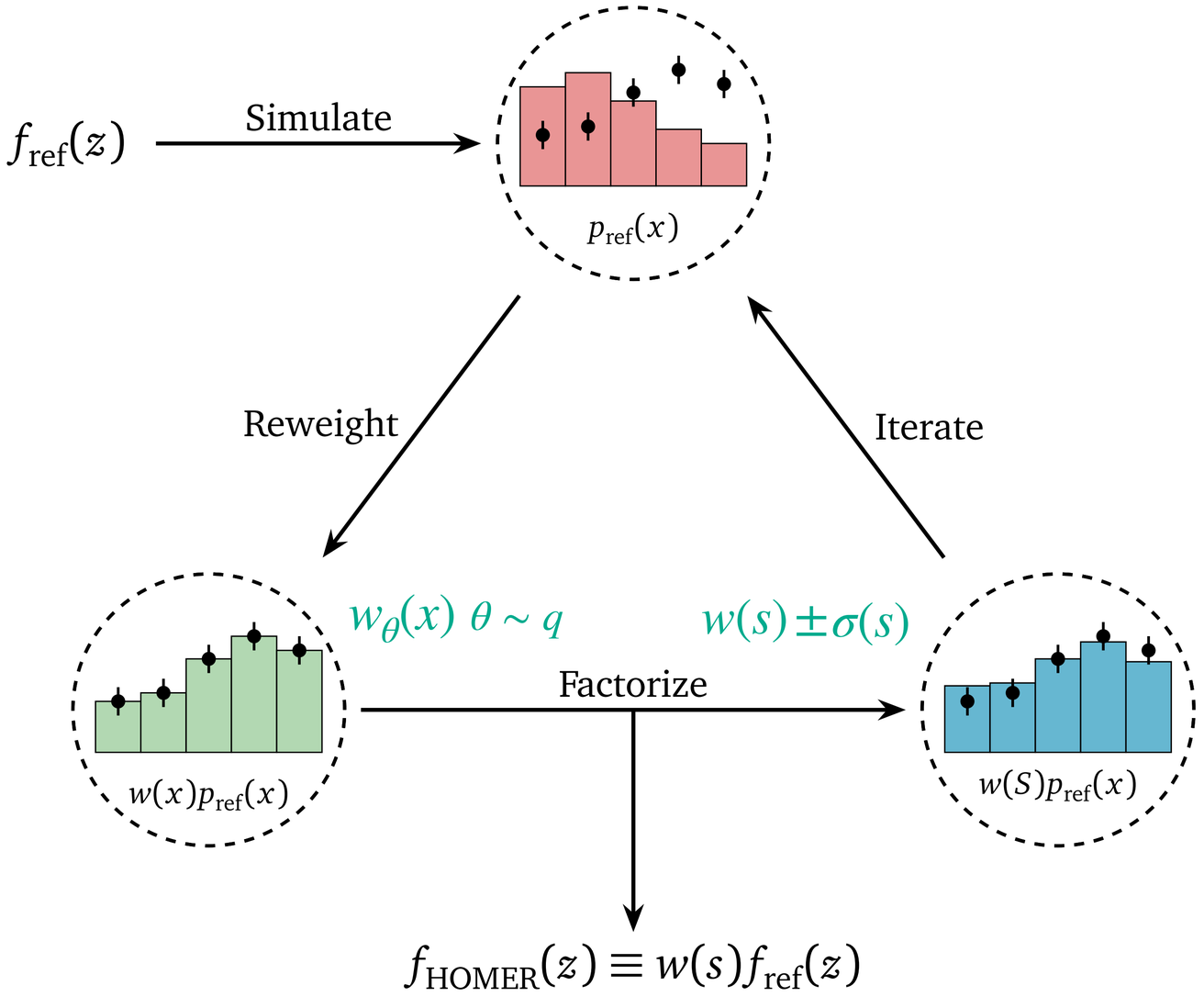
iHOMER: Uncertainties

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

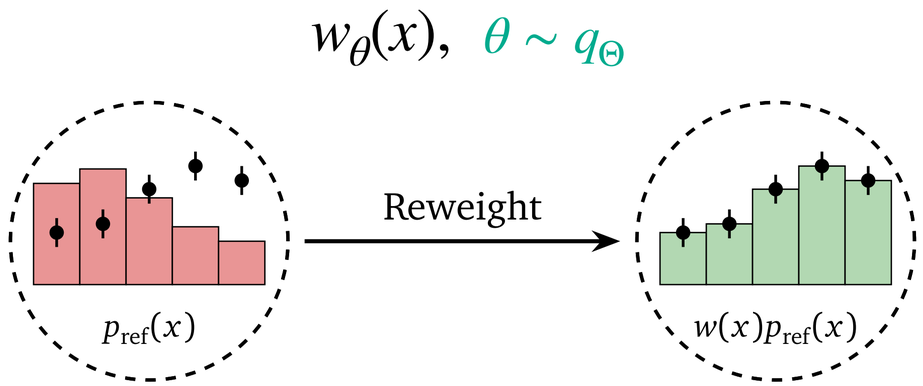
  Bayesian Neural Network

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

  Gaussian regression



iHOMER: Step 1 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:

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

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

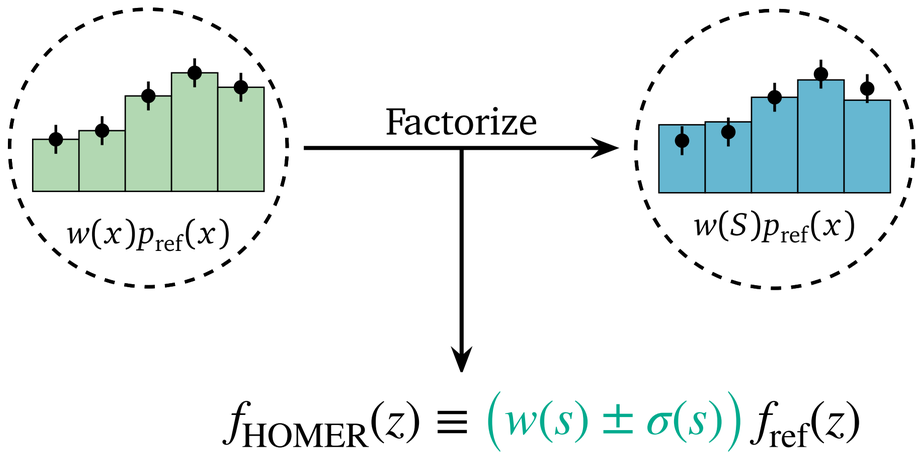
e.g. Classification loss

- After training, we can estimate uncertainties by sampling

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

where f is any function of the network.

iHOMER: Step 2 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:

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

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

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

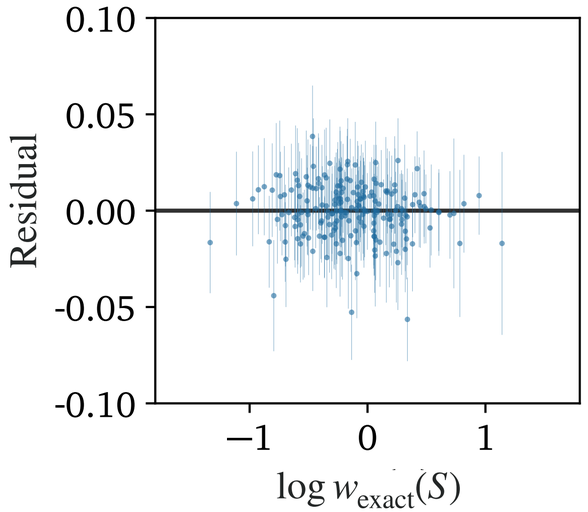
Uncertainty Calibration

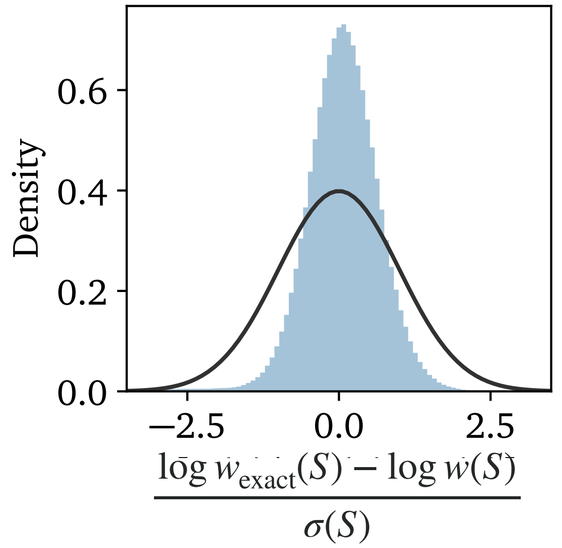
- Centred but narrow pulls on log-weights
- Gaussian for chains, but wilder for breaks
- Actually expected:

  High-level observables cannot fully constrain fragmentation

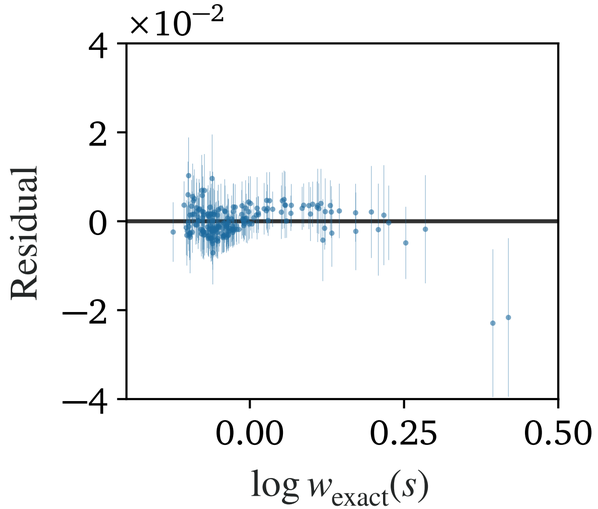
  +

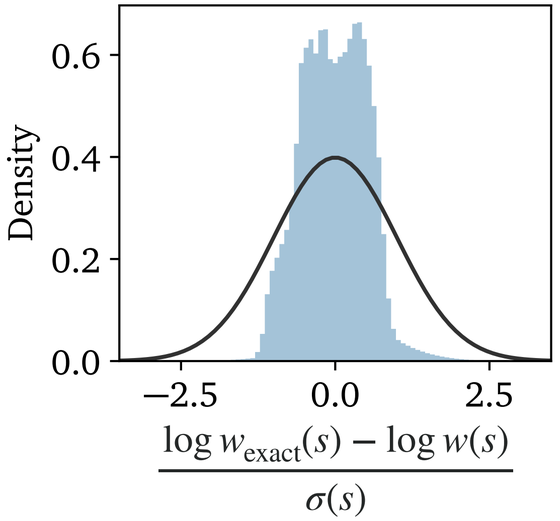
  We evaluate against just one compatible fragmentation model.





Chain pull





Break pull

Summary

- Machine learning is a natural upgrade to empirical models at colliders

  … but there are many constraints!

- ***iHOMER*** ticks all the boxes for a neural hadronization model

  ✓ Likelihood-free ✓ Physical prior from string model ✓ Uncertainties

  ✓ Bypasses detector gradients ✓ Marginalisation by iteration

- *Lots more to be studied:*

  - Validating beyond simplified scenarios

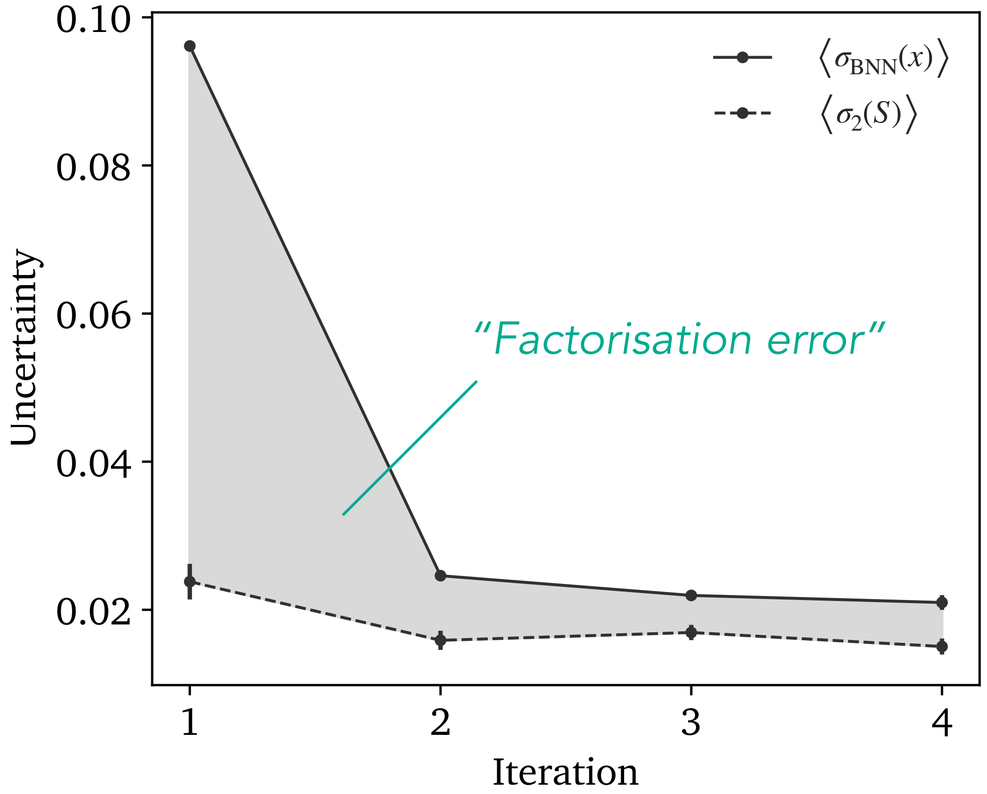
  - How much is HOMER limited by the string picture?

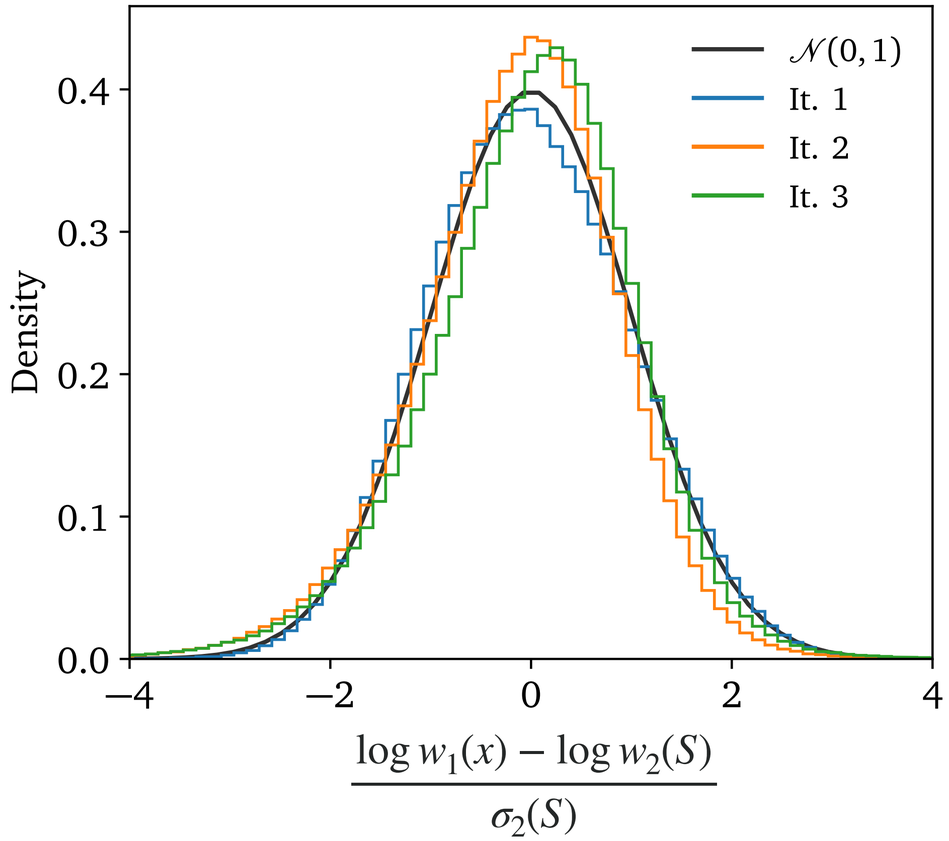
  - What if detector simulation / parton shower are uncertain? (see 2409.10421)

  - To what degree can non-fragmentation effects be absorbed? (see 2307.10370)

- *Might see an ML tune in coming years…*

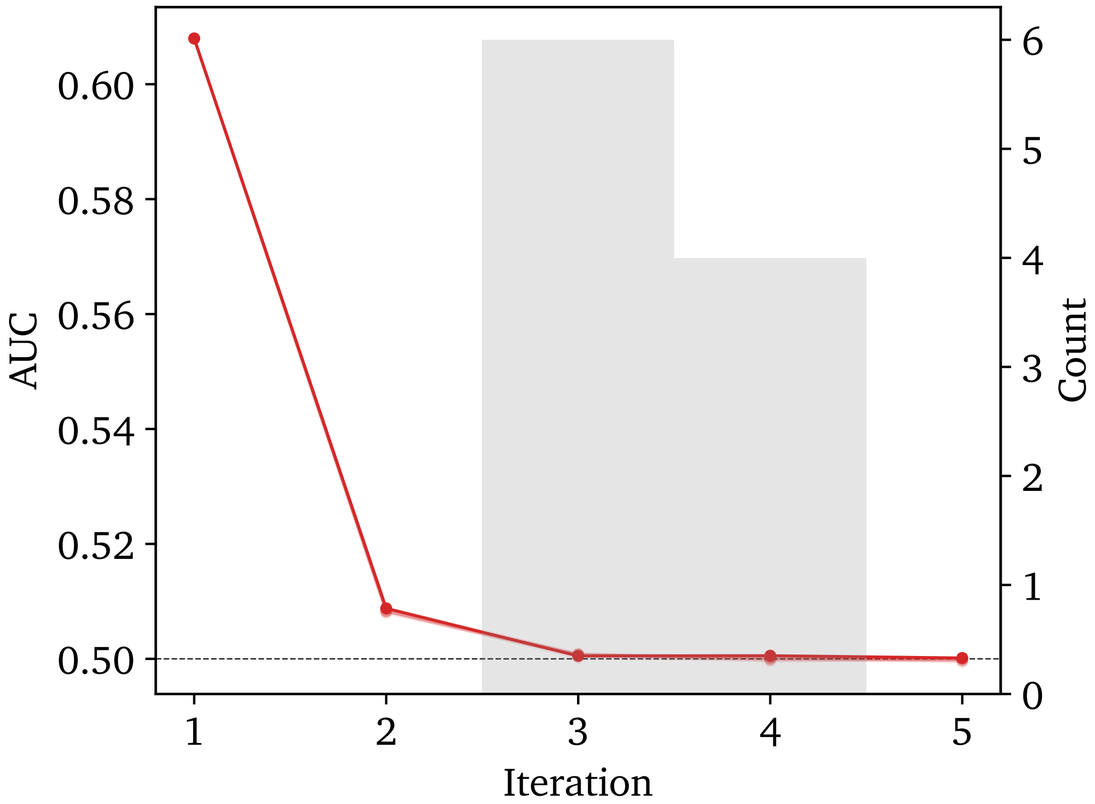
Backup: Step 1 vs Step 2 Uncertainties





Backup: Stopping condition

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



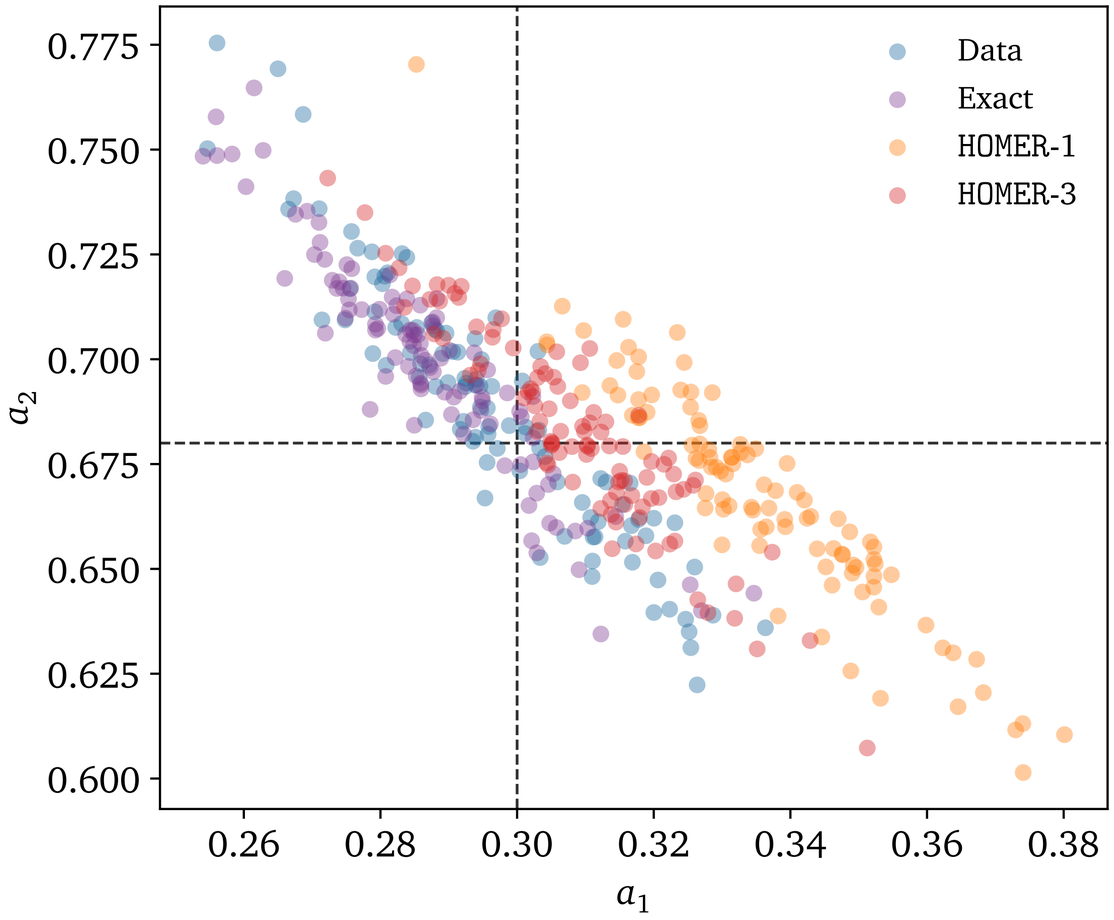
Backup: Parametric fit

- 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



Backup: Acceptance effects

- Some chains cannot be resolved into a physical hadronic system → Rejected by Pythia

$$A(S) = \begin{cases} 1 & S \text{ accepted} \\ 0 & S \text{ rejected} \end{cases}$$

- Consequently, probability for events and chains are related by the *acceptance rate*:

$$p(x) = \frac{1}{\alpha} p(S) \left| \frac{dS}{dx} \right| \quad \text{with} \quad \alpha \equiv \int dS A(S) p(S)$$

- Factorisation condition should also be amended, including an inferred acceptance rate:

$$w(x) \approx \frac{\alpha_{\text{ref}}}{\alpha(\phi)} \prod_{s \in S} w_\phi(s) \quad \text{with} \quad \alpha(\phi) \equiv \int dS \prod_{s \in S} w_\phi(s) A(S) p(S)$$