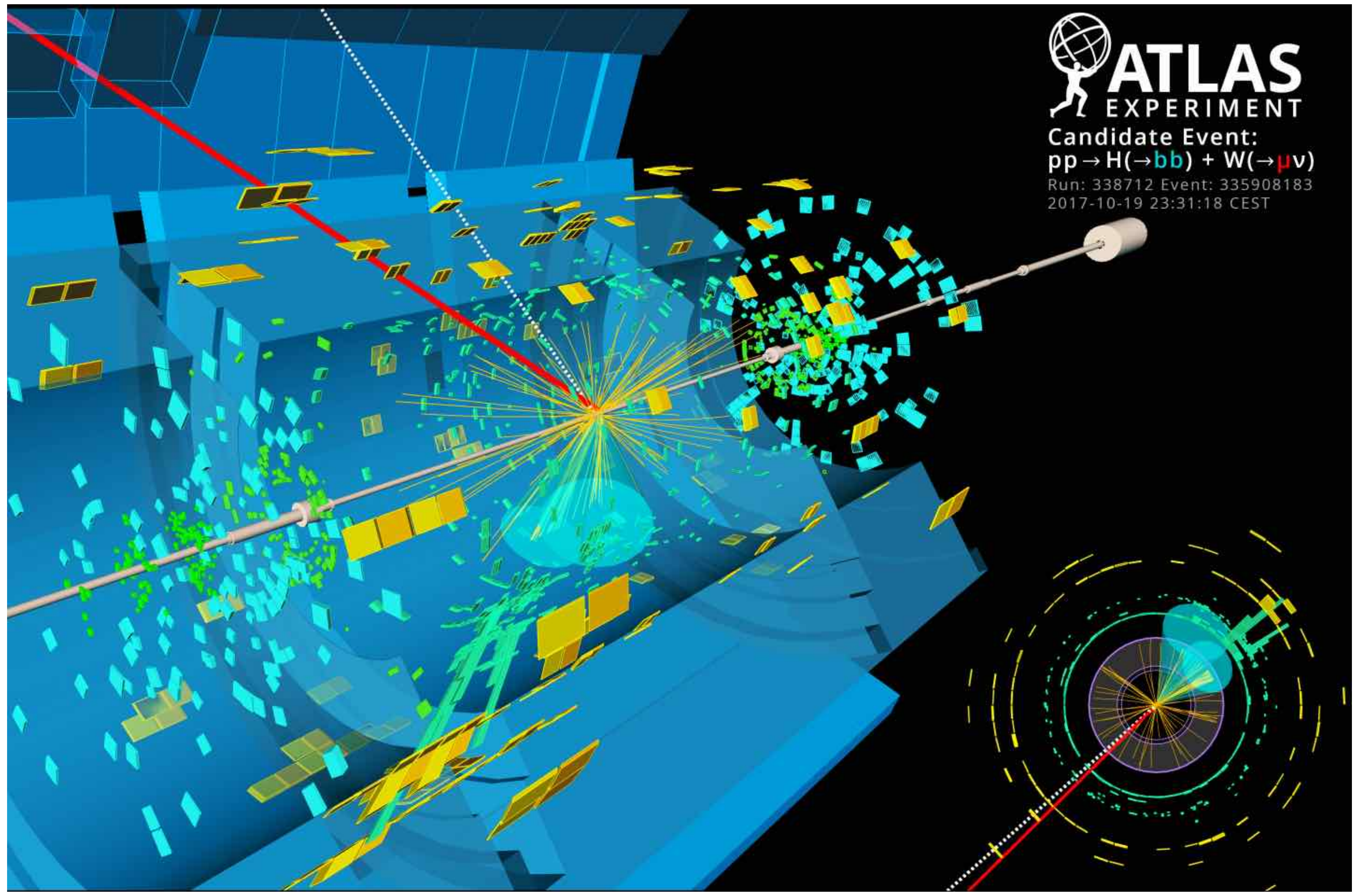
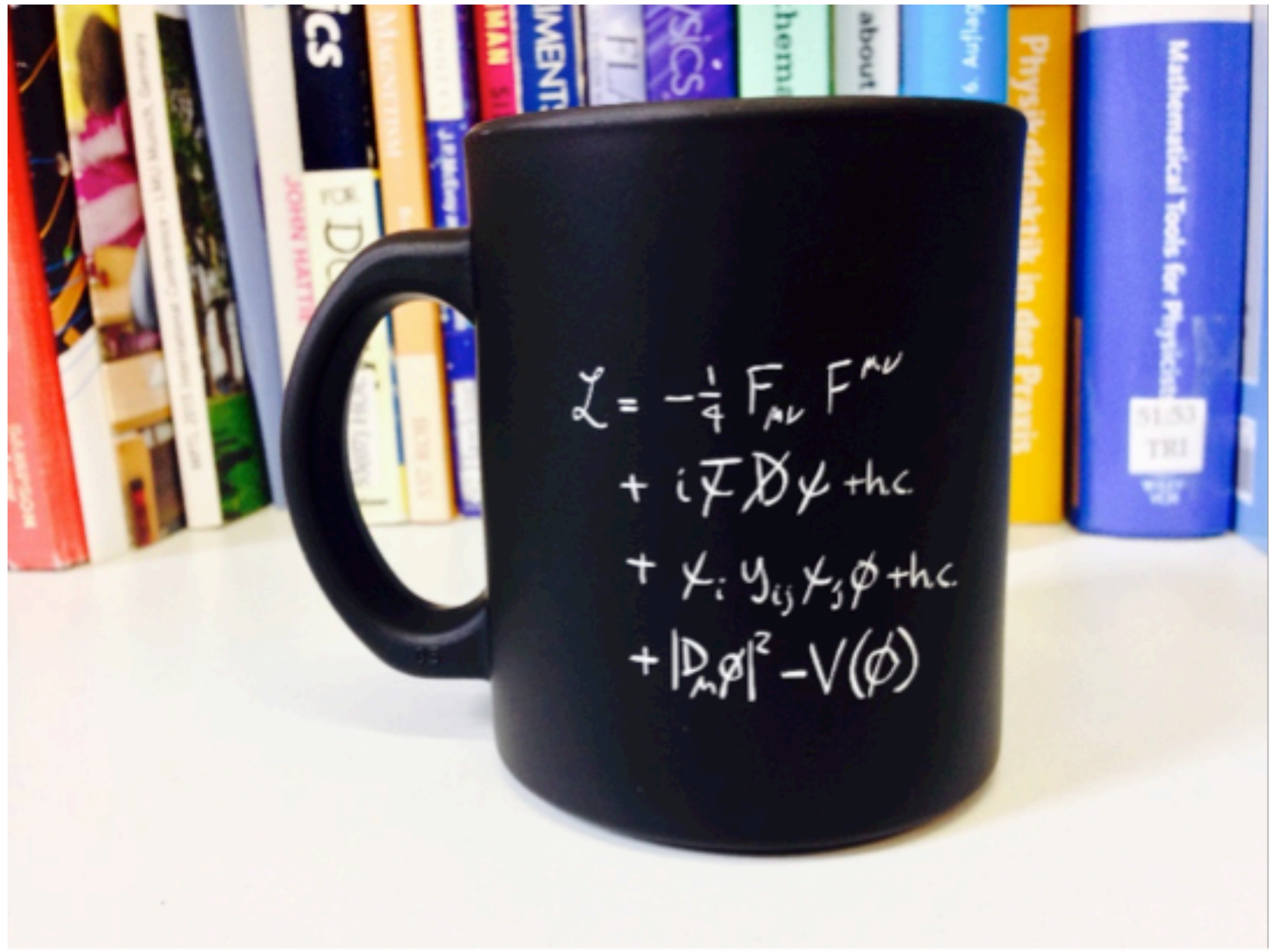
Boosting Loop Amplitudes and Event Generation with Precision Networks

Probing the frontiers of Nuclear physics with AI at EIC
26 September 2023

Anja Butter, LPNHE/ITP

From Theory to Data and Back



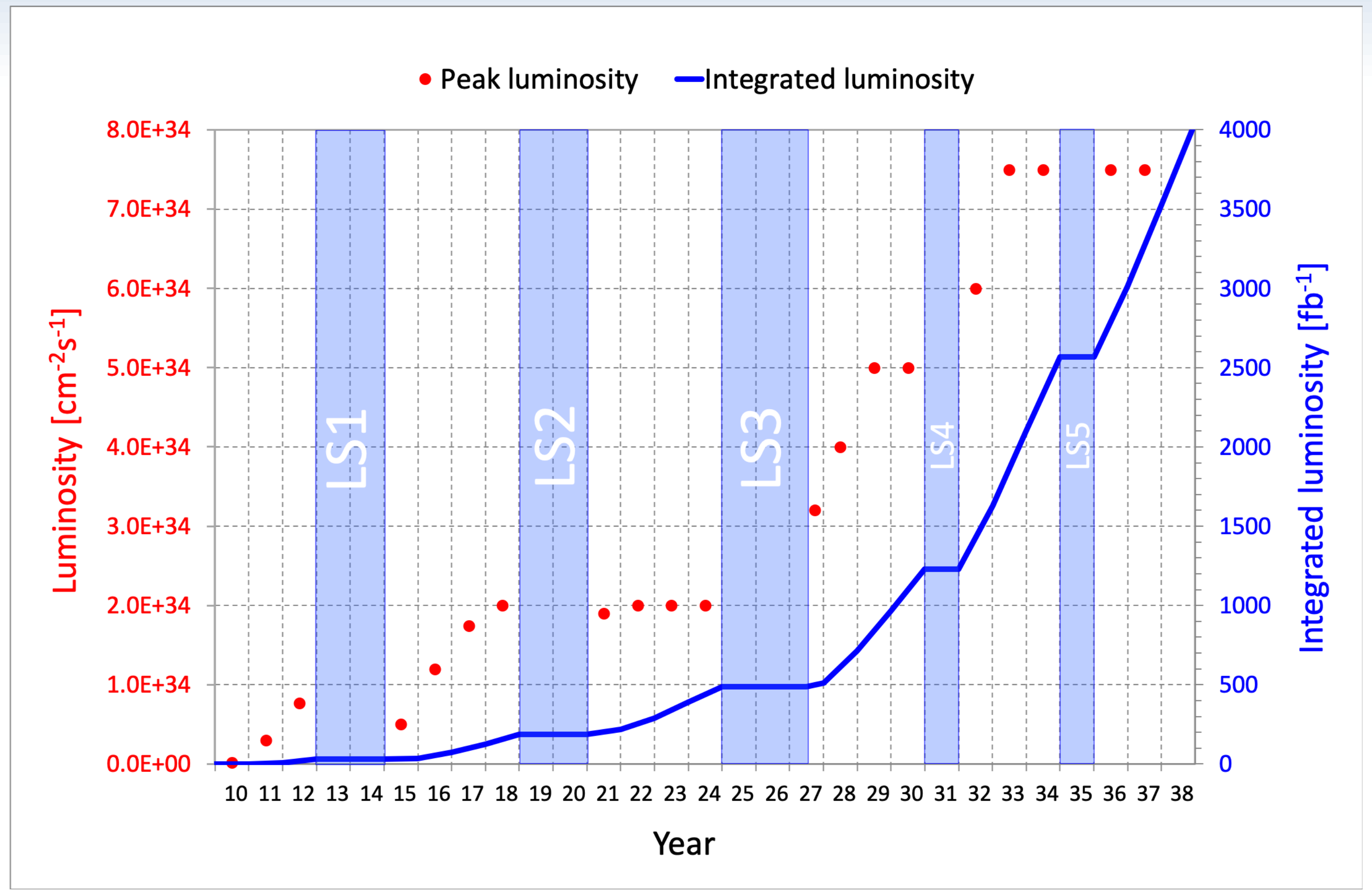


Goal

- Understand full dataset from **1st principles**
- Precision measurements of the SM
- Find signs of new physics (eg dark matter)
- **Huge** dataset ~1Pb/s before trigger selection

Open questions

- Facing **25 times** the amount of data
- What do we need to understand the data? (*read:* find new physics)



- **Precision predictions**
  - Higher order amplitudes
  - Event generation
  - Shower
  - Detector simulation

- **Optimized analysis for high-dimensional data**
  - Unfolding
  - MEM
  - Anomaly detection
  - Uncertainty treatment for ML methods

How can ML help?

Monte carlo event generation

1. Generate phase space points

$\rightarrow$ set of four-momenta p_i

2. Calculate event weight

$$w_{\text{event}} = f(x_1, Q^2) f(x_1, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots, p_n) \times J(p_i(r))$$

PDF

Matrix element

Phase space mapping

3. Unweighting

keep events with $\dfrac{w_i}{w_{\max}} > r \in [0,1]$

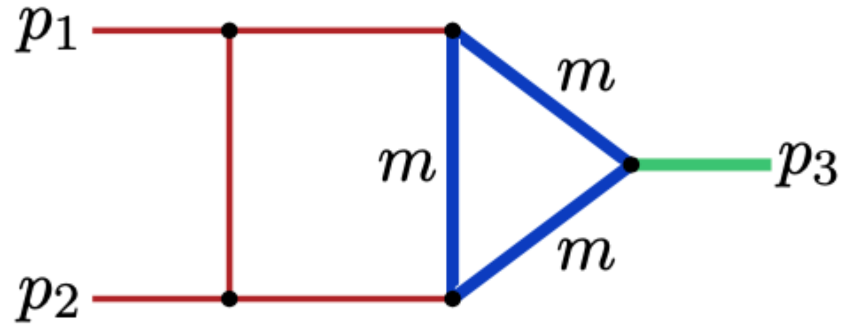
Bottlenecks

1. Slow **matrix element** calculation
   - Complexity grows exponentially with
     - # final state particles
     - Precision (LO, NLO, NNLO, ...)
2. Low **unweighting** efficiency
   - Discard most events if $w_i \ll w_{\max}$
   - Optimize phase space mapping
     ➡ $J(p_i(r)) = (f \times \mathcal{M})^{-1}$

ML for Amplitudes
-
Reducing variance in exact calculations

Multi-loop calculations with NNs

Precision predictions based on loop diagrams



Analytic expression for loop amplitude

$$G = \int_{-\infty}^{\infty} \left(\prod_{l=1}^{L} \frac{\mathrm{d}^D k_l}{i\pi^{\frac{D}{2}}} \right) \prod_{j=1}^{N} \frac{1}{(q_j^2 - m_j^2 + i\delta)^{\nu_j}}$$

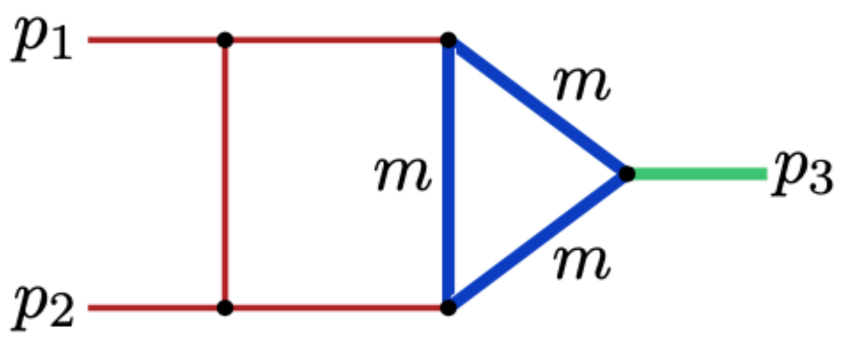
$$= \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \, x_j^{\nu_j - 1} \, \frac{U^{\nu-(L+1)D/2}}{F^{\nu-LD/2}} = \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \; I(\vec{x})$$

Rewrite with
Feynman parameters

Still contains singularities

Multi-loop calculations with NNs

Precision predictions based on loop diagrams



Analytic expression for loop amplitude

$$G = \int_{-\infty}^{\infty} \left(\prod_{l=1}^{L} \frac{\mathrm{d}^D k_l}{i\pi^{\frac{D}{2}}} \right) \prod_{j=1}^{N} \frac{1}{(q_j^2 - m_j^2 + i\delta)^{\nu_j}}$$

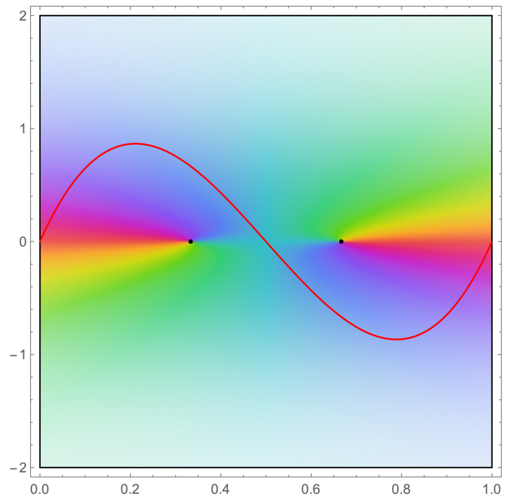
$$= \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \, x_j^{\nu_j - 1} \frac{U^{\nu-(L+1)D/2}}{F^{\nu - LD/2}} = \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \; I(\vec{x})$$

Rewrite with
Feynman parameters

Still contains singularities

Solved by contour deformation due to Cauchy's theorem

$$\int_0^1 \prod_{j=1}^{N} \mathrm{d}x_j \, I(\vec{x}) = \int_0^1 \prod_{j=1}^{N} \mathrm{d}x_j \; \det\left(\frac{\partial \vec{z}(\vec{x})}{\partial \vec{x}} \right) I(\vec{z}(\vec{x}))$$



Optimal parametrization = minimal variance

Turn it into an ML Problem

Integration with normalizing flows

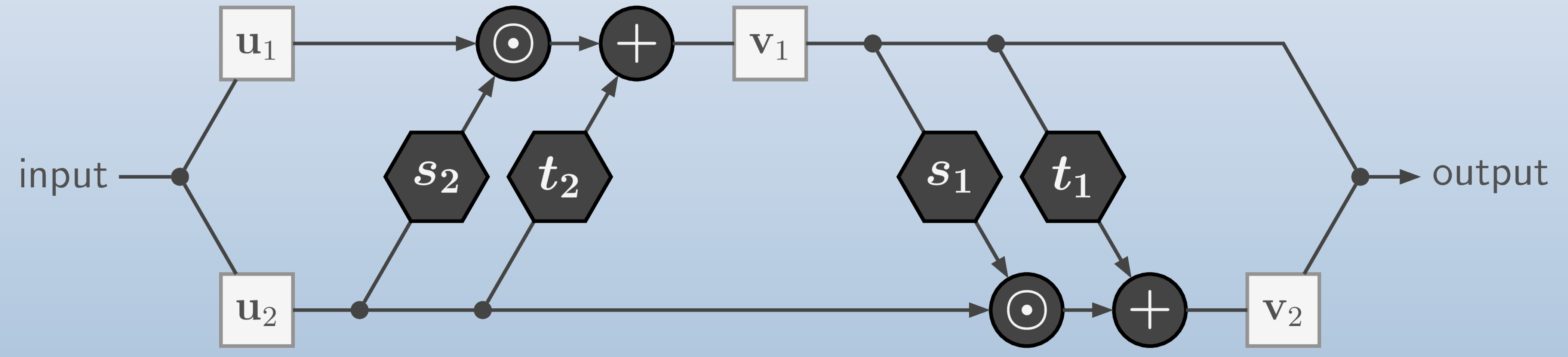
Numeric evaluation of integral $G = \int_0^1 \mathrm{d}x_j \ \det\left(\frac{\partial \vec{z}(\vec{x})}{\partial \vec{x}}\right) I(\vec{z}(\vec{x}))$

Normalizing flow networks

+ Bijective mapping
+ Tractable Jacobian
+ Combine many blocks

Parametrization $\rightarrow z = \mathrm{INN}(x)$

$\vec{x}$



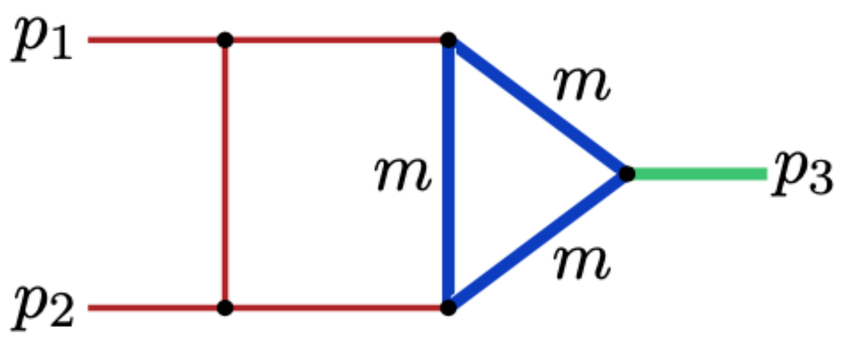
$\vec{z}(\vec{x})$

Minimize **variance** $\rightarrow$ loss $\mathscr{L} = \sigma_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left| \det\left(\frac{\partial \vec{z}(\vec{x}_{(i)})}{\partial \vec{x}_{(i)}}\right) I(\vec{z}(\vec{x}_{(i)})) - \langle I \rangle \right|^2$

Multi-loop calculations with INNs

Profiting from the Jacobian

Precision predictions based on loop diagrams



Analytic expression for loop amplitude

$$G = \int_{-\infty}^{\infty} \left(\prod_{l=1}^{L} \frac{\mathrm{d}^D k_l}{i\pi^{\frac{D}{2}}} \right) \prod_{j=1}^{N} \frac{1}{(q_j^2 - m_j^2 + i\delta)^{\nu_j}}$$

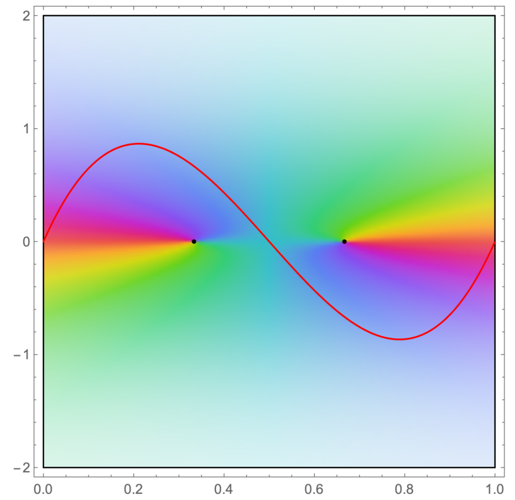
$$= \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \, x_j^{\nu_j - 1} \frac{U^{\nu-(L+1)D/2}}{F^{\nu-LD/2}} = \int_0^1 \prod_{j=1}^{N-1} \mathrm{d}x_j \, I(\vec{x})$$

Rewrite with
Feynman parameters

Still contains singularities

Solved by contour deformation due to Cauchy's theorem

$$\int_0^1 \prod_{j=1}^{N} \mathrm{d}x_j \, I(\vec{x}) = \int_0^1 \prod_{j=1}^{N} \mathrm{d}x_j \, \det\left(\frac{\partial \vec{z}(\vec{x})}{\partial \vec{x}} \right) I(\vec{z}(\vec{x}))$$

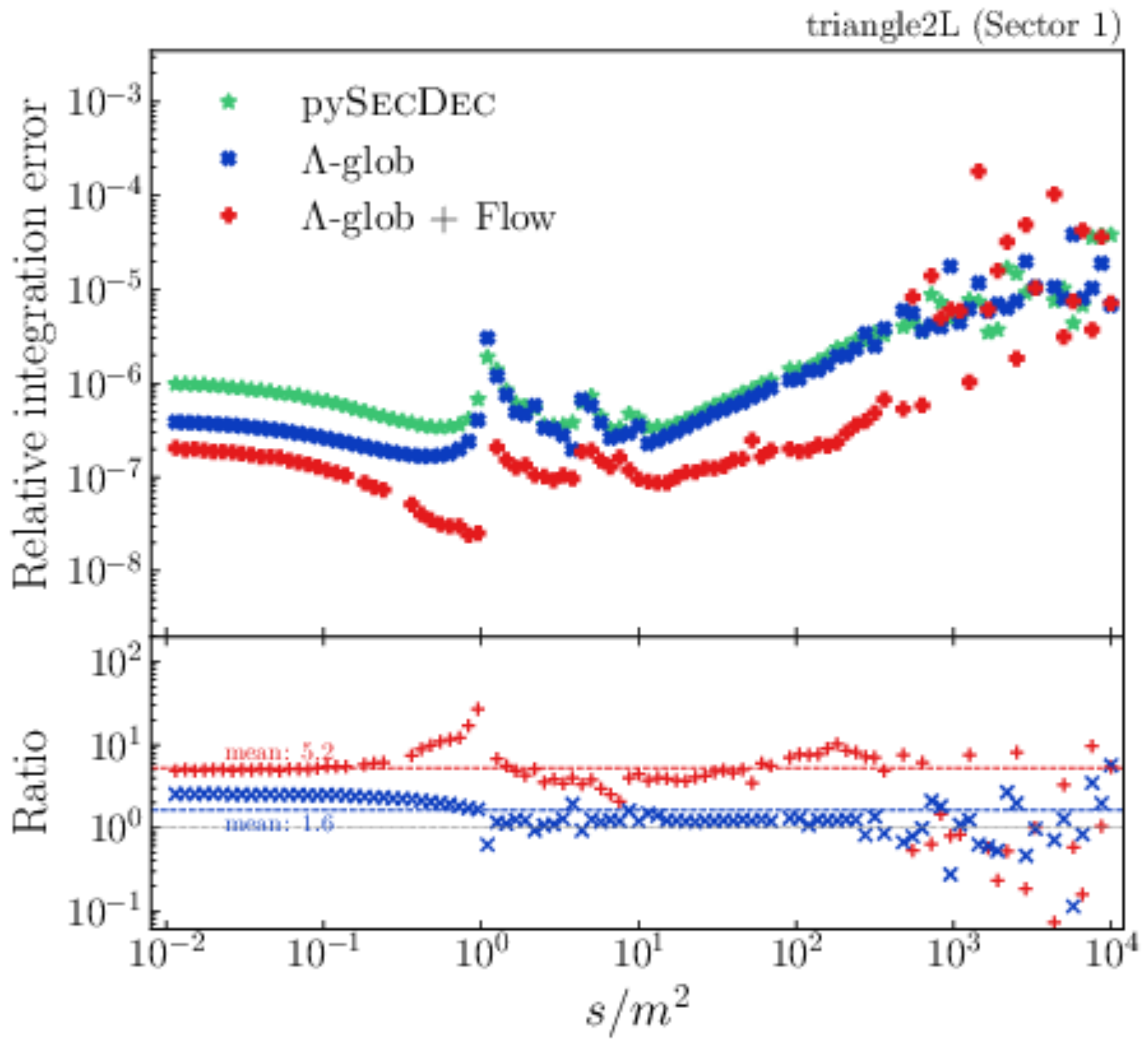


Optimal parametrization = minimal variance

Turn it into an ML Problem

Parametrization $\rightarrow z = \mathrm{INN}(x)$

Variance $\rightarrow \mathscr{L}$



R. Winterhalder, et al. [2112.09145]

ML for Amplitudes
-
Approximation with Regression

Limitations of a standard network

Example
$gg \to \gamma\gamma g(g)$ @LO

90k training amplitudes
870k test amplitudes

Standard approach

Training data
$T =$ (phase space points x, Amplitudes $A'(x)$)

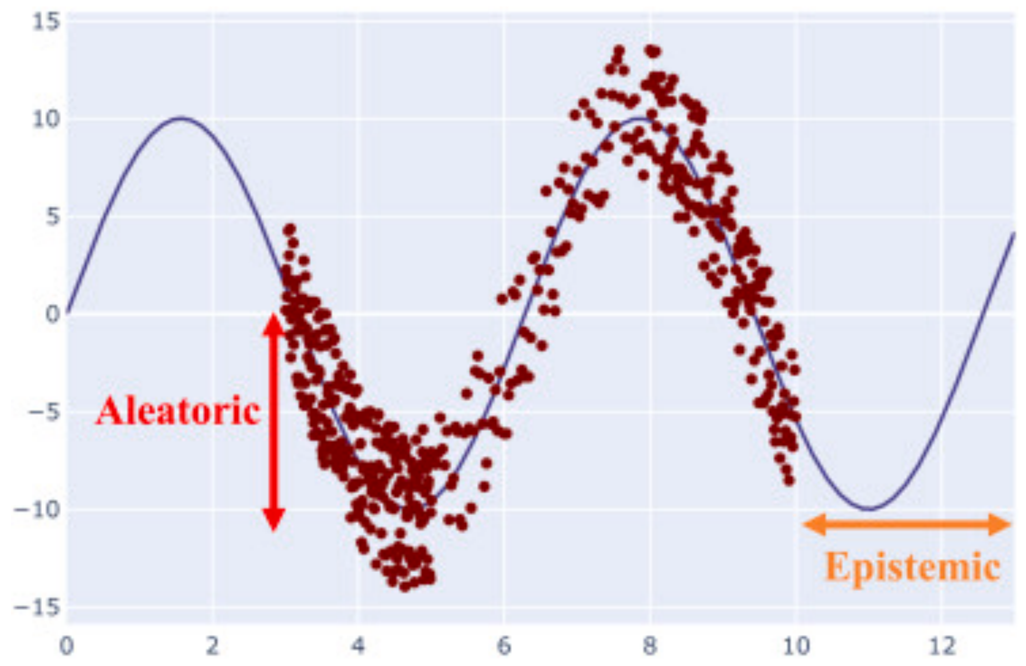
Loss
$\mathscr{L} = (A'(x) - NN(x))^2$

PROBLEM: For limited data there is **no unique solution**

→ Need better formulation of the problem

→ Find $p(A \mid x, T)$ (from now on x is implicit)

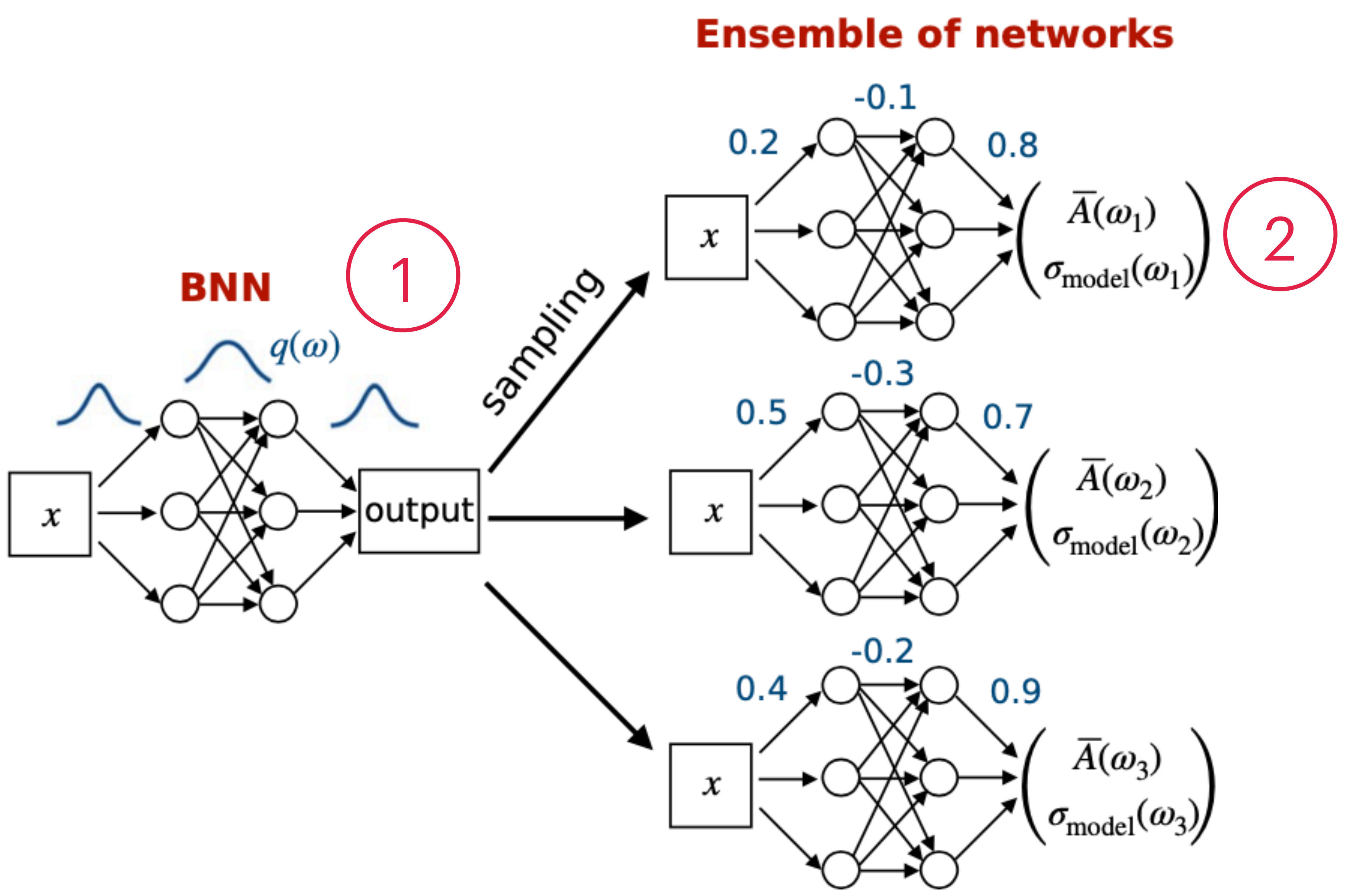
$$\rightarrow p(A) = \int \mathrm{d}w \; p(A \mid w) p(w \mid T)$$



Capturing probabilities with Bayesian networks

$$\boxed{p(A) = \int dw\ \overset{2}{p(A\,|\,w)}\,\overset{1}{p(w\,|\,T)} \approx \int dw\ p(A\,|\,w)q(w)}$$

Bayesian network



Building the loss function

Approximate $q(w)$ by minimizing KL divergence

$$\mathscr{L}_{BNN} = \mathrm{KL}[q(w), p(w\,|\,T)]$$

$$= \int dw\ q(w)\ \log \frac{q(w)}{p(w\,|\,T)}$$

$$= \int dw\ q(w)\ \log \frac{q(w)p(T)}{p(w)p(T\,|\,w)}$$

$$= \mathrm{KL}[q(w), p(w)] - \int dw\ q(w)\ \log p(T\,|\,w)$$

1 Gaussian prior

$$\boxed{\frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \log \frac{\sigma_p}{\sigma_q}}$$

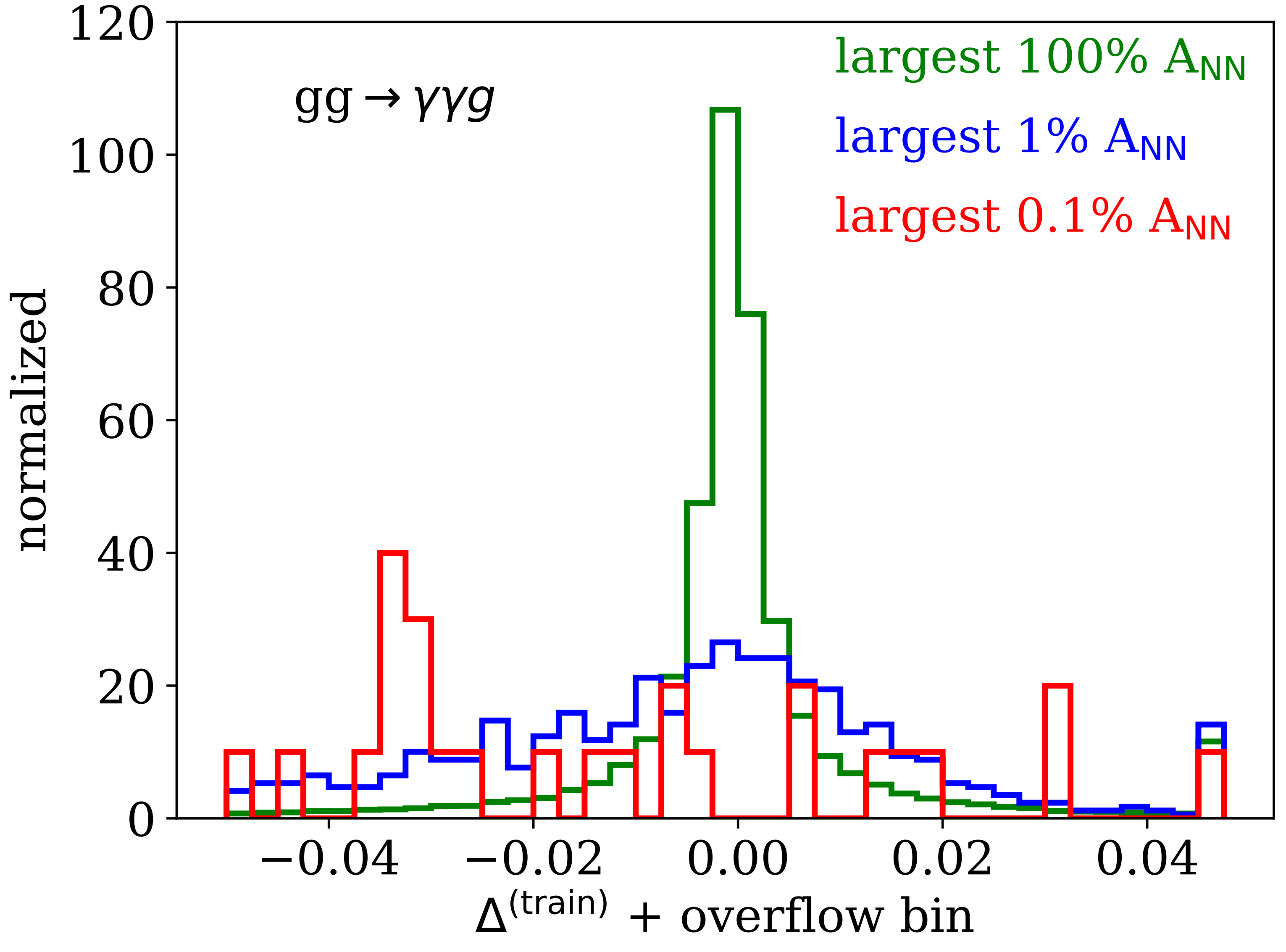
2 Gaussian uncertainty

$$\boxed{\frac{\left|\bar{A}_j(\omega) - A_j^{(\mathrm{truth})}\right|^2}{2\sigma_{\mathrm{model},j}(\omega)^2} + \log \sigma_{\mathrm{model},j}(\omega)}$$

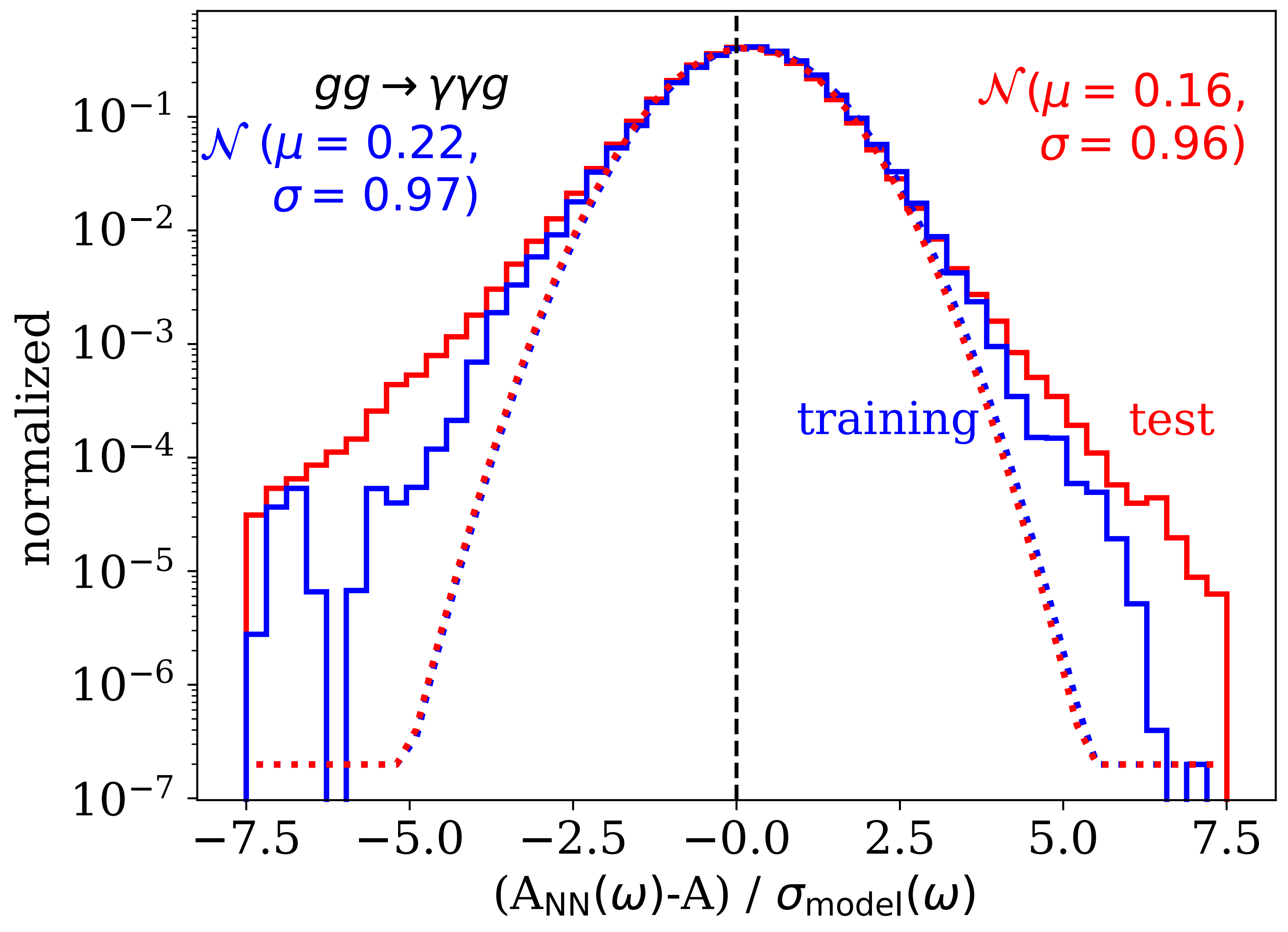
Results - out of the box

+ Deviations at 1 percent level

Precision $\Delta^{(train)} = \frac{A_{NN} - A_{train}}{A_{NN}}$



Calibration $\Delta^{(train)} = \frac{A_{NN} - A_{train}}{A_{NN}}$



Performance worse for rare points with large amplitudes (collinear)

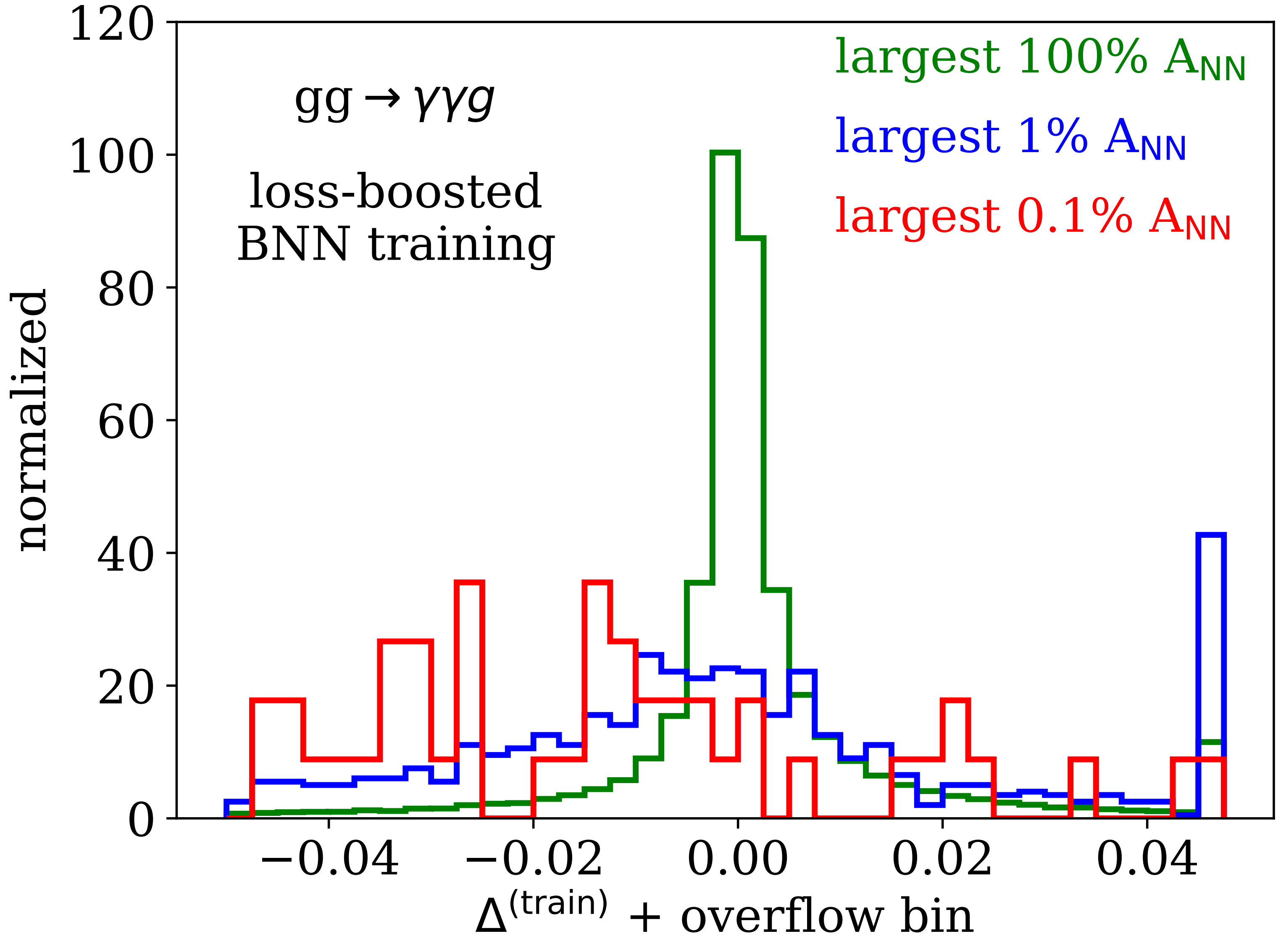
Roughly Gaussian but enhanced tails

Loss boosting

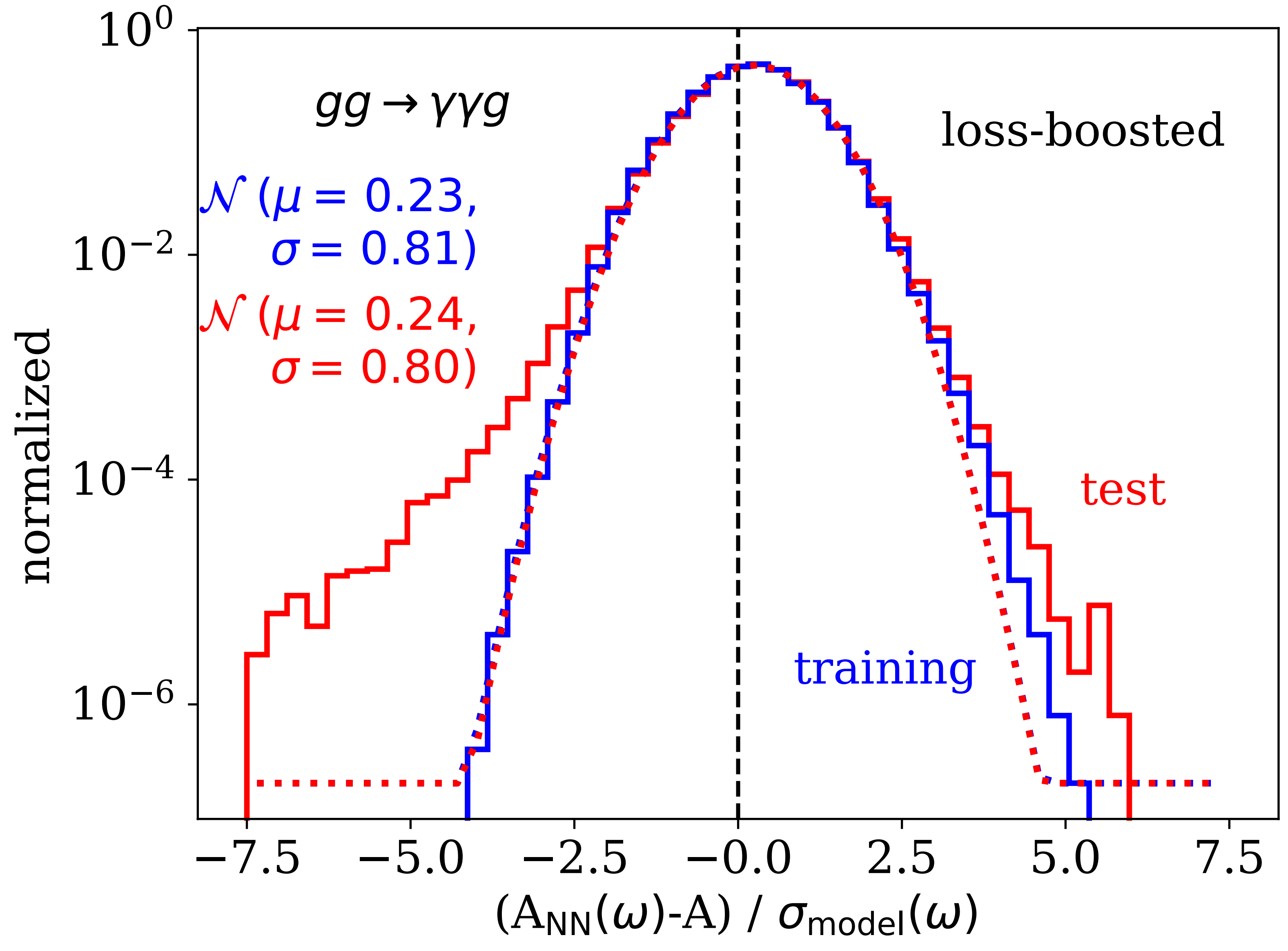
Enforce training on samples with $\Delta A > 2\sigma$

$\rightarrow$ include them 5 times in each epoch

$\rightarrow$ Repeat 4 times



No change in performance



Tails reproduced for training data

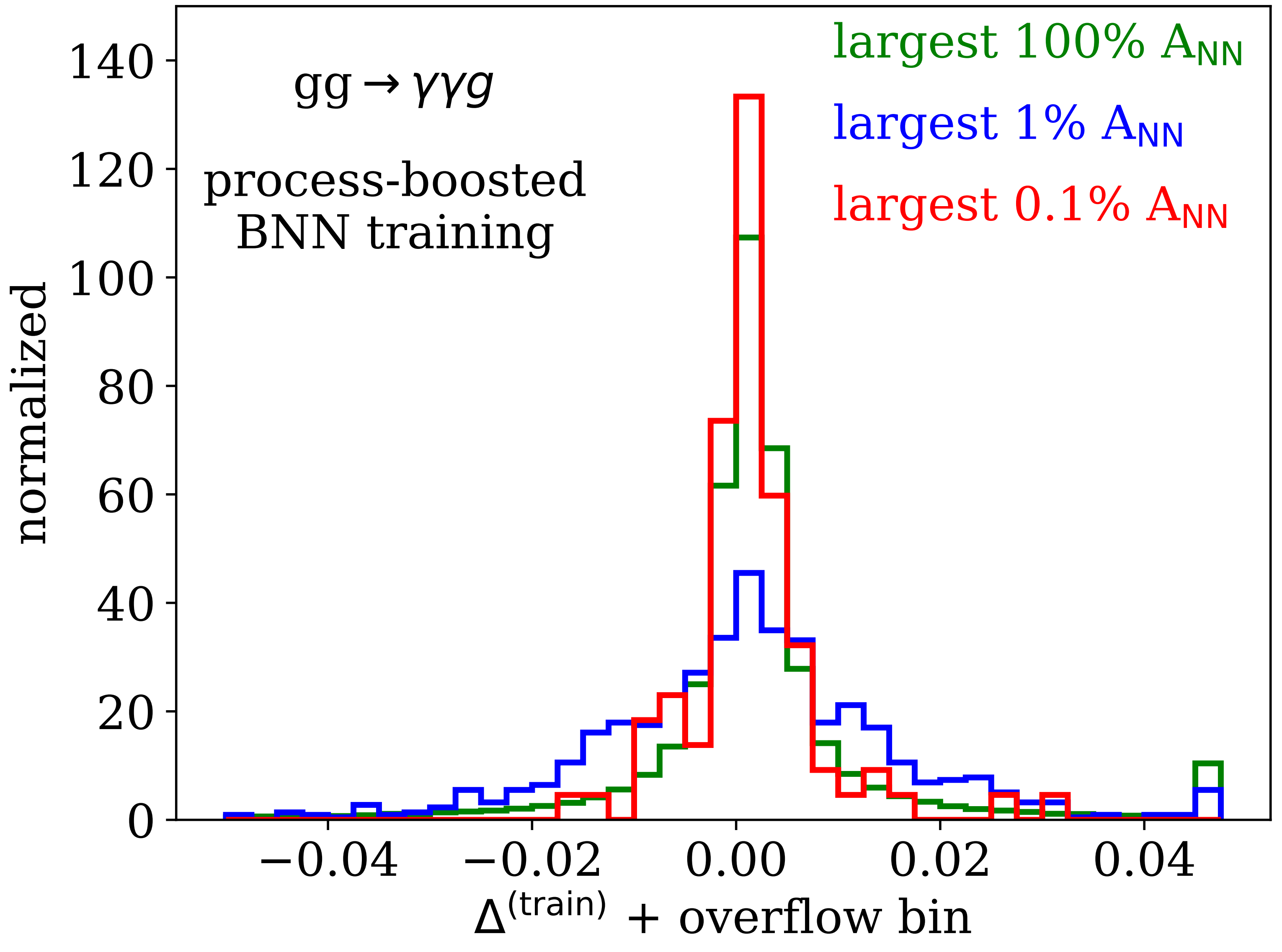
Improvement for test data

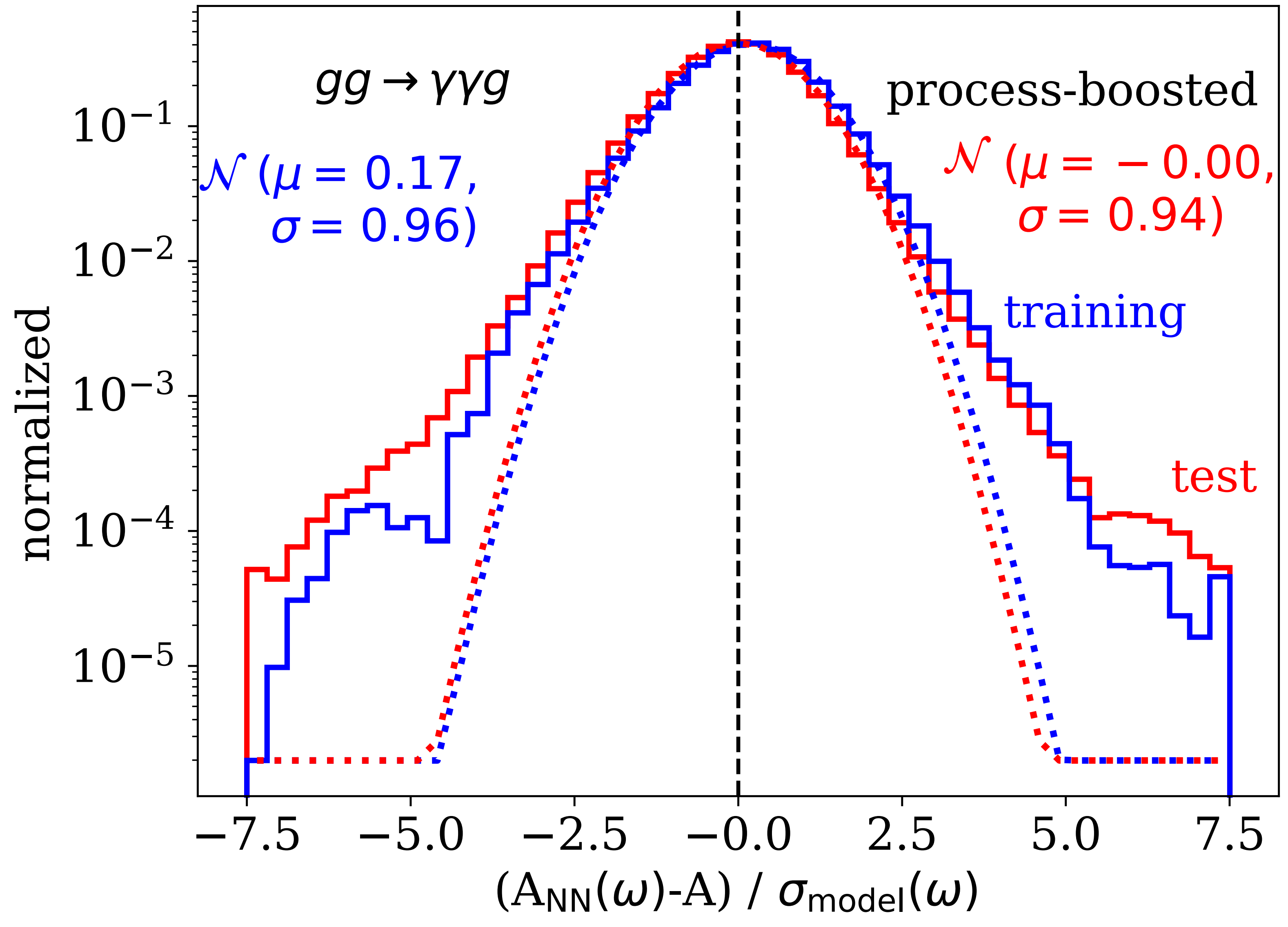
Performance boosting

Enforce training on 200 samples with largest uncertainty σ_{tot}

→ include them +3 times in each epoch

→ Repeat 20 times





Significant improvement in performance

Precision networks for loop amplitudes

1. Normalizing flows can **reduce variance of integration** in exact loop calculations

2. Aproximating amplitudes:
Bayesian networks provide **uncertainty estimates**
Boost network performance for precision or calibration

Monte carlo event generation

1. Generate phase space points

$\rightarrow$ set of four-momenta p_i

2. Calculate event weight

$$w_{\text{event}} = f(x_1, Q^2)f(x_1, Q^2) \times \mathscr{M}(x_1, x_2, p_1, \ldots, p_n) \times J(p_i(r))$$

PDF

Matrix element

Phase space mapping

3. Unweighting

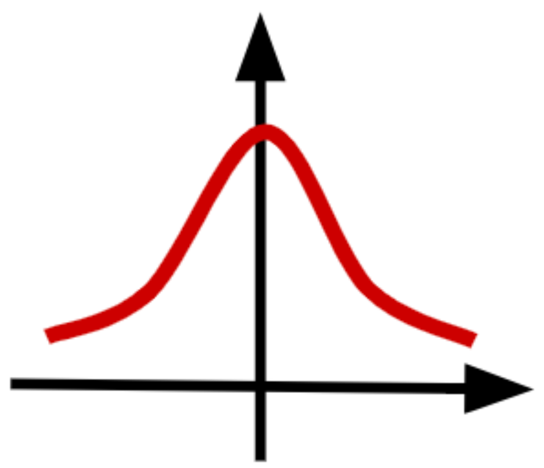
keep events with $\dfrac{w_i}{w_{\max}} > r \in [0,1]$

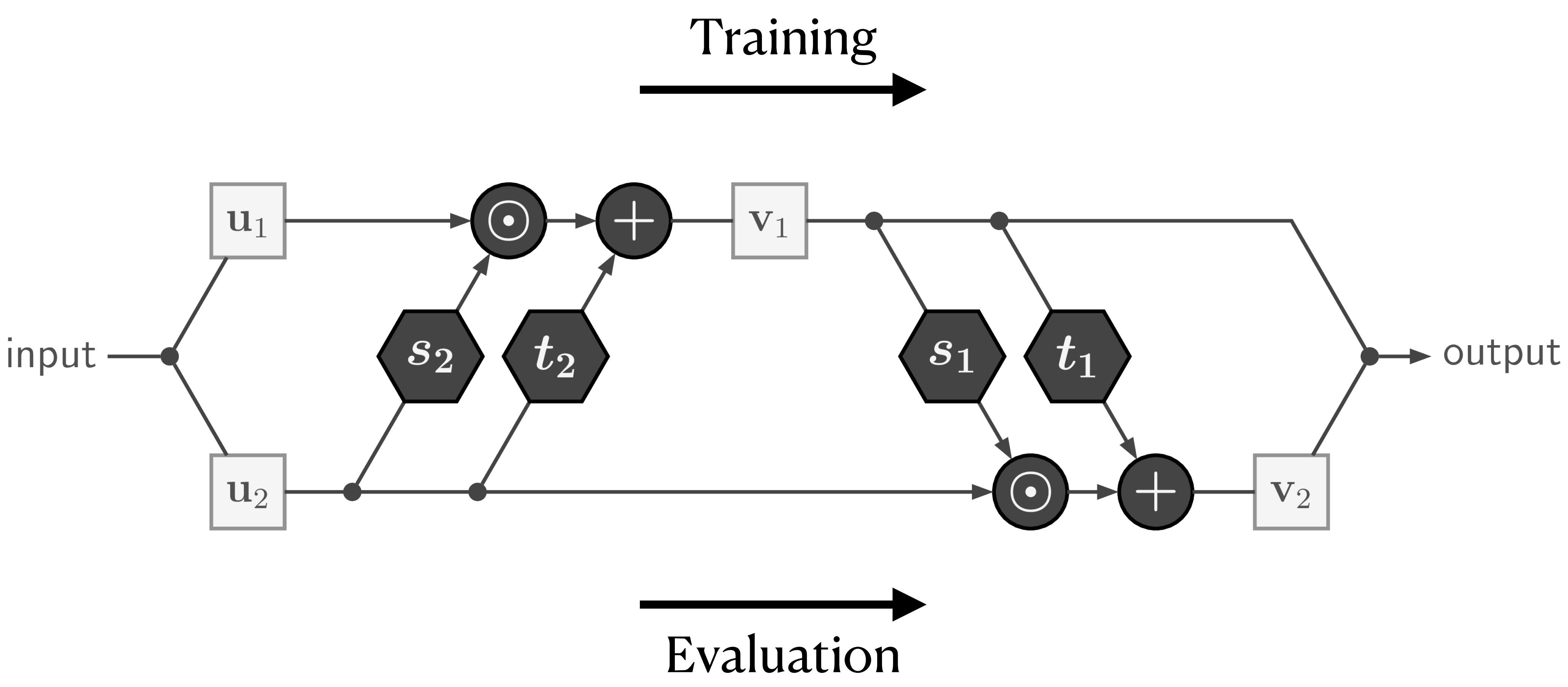
ML for phase space generation

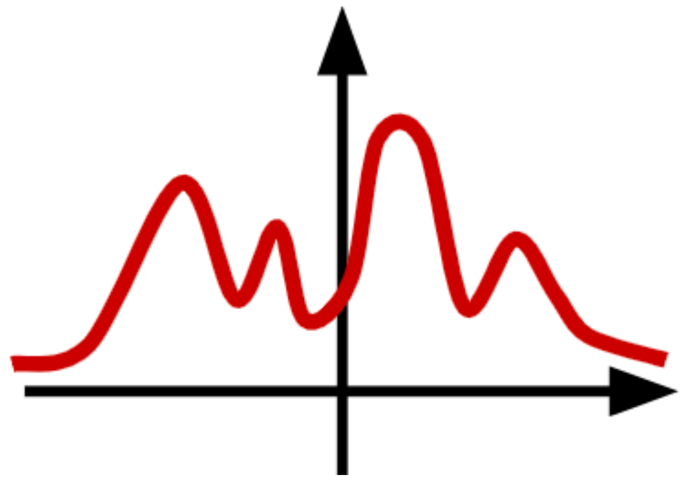
Normalizing flows

Invertible networks for complex transformations

+ Bijective mapping
+ Tractable Jacobian $\rightarrow p_x(x) = p_z(z) \cdot J_{NN}$
+ Fast evaluation in both direction







Training on density $t(x)$

$\rightarrow$ Minimize difference

$$\mathscr{L} = \log p_x(x)/t(x)$$
$$= \log p_z(z(x))\, J_{NN} / t(x)$$

Training on samples x

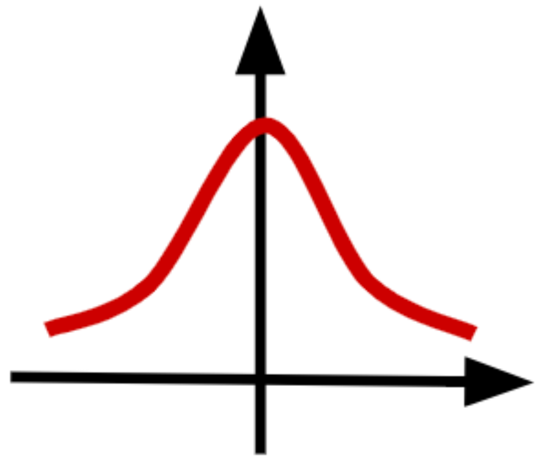
$\rightarrow$ Maximize the log-likelihood

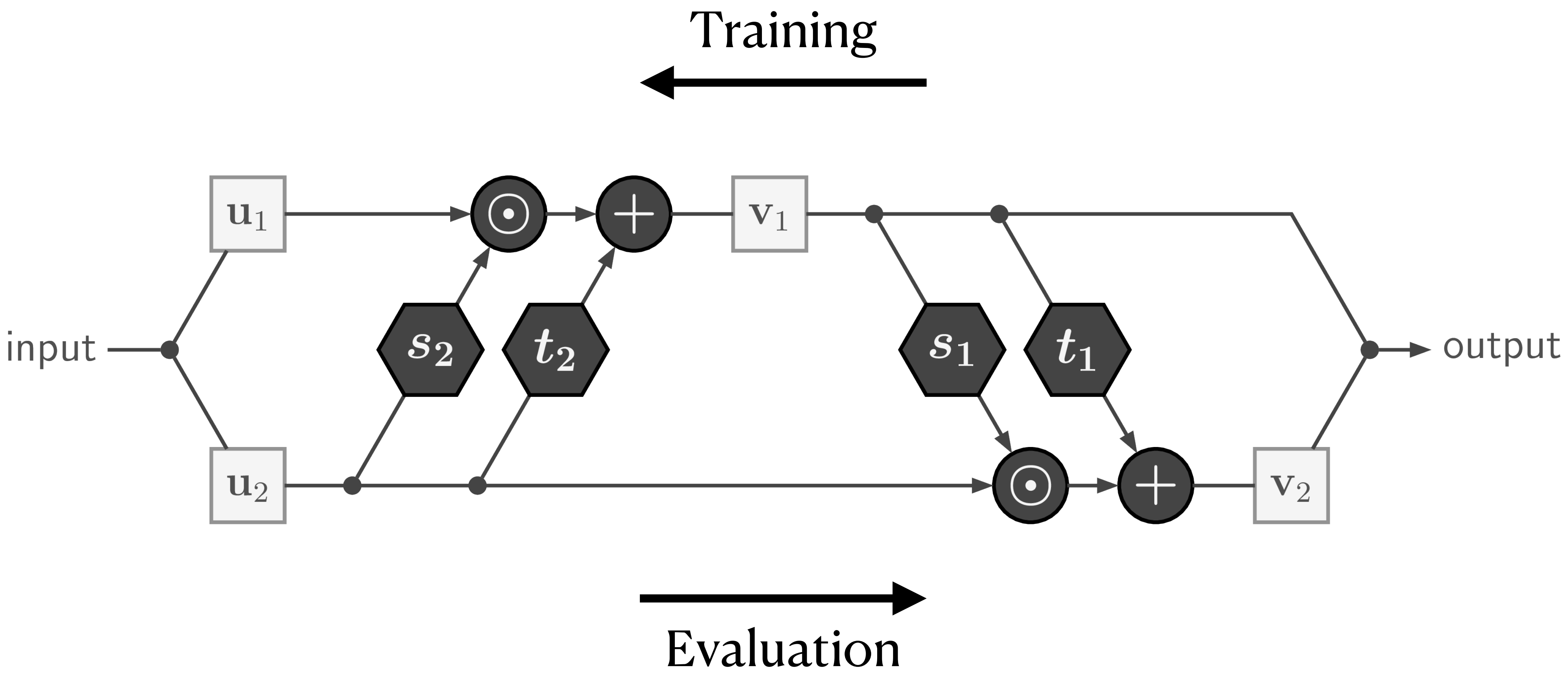
$$\mathscr{L} = \log p(\theta \,|\, x)$$
$$= \log p(z \,|\, \theta) + \log J_{NN} + p(\theta)$$

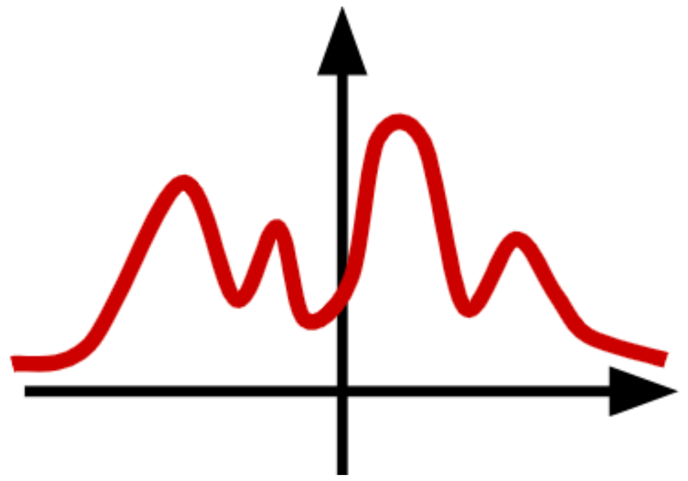
Normalizing flows

Invertible networks for complex transformations

+ Bijective mapping
+ Tractable Jacobian → $p_x(x) = p_z(z) \cdot J_{NN}$
+ Fast evaluation in both direction







Training on density $t(x)$

→ Minimize difference

$$\begin{aligned}\mathscr{L} &= \log p_x(x)/t(x) \\ &= \log p_z(z(x))\, J_{NN} \,/\, t(x)\end{aligned}$$

Training on samples x

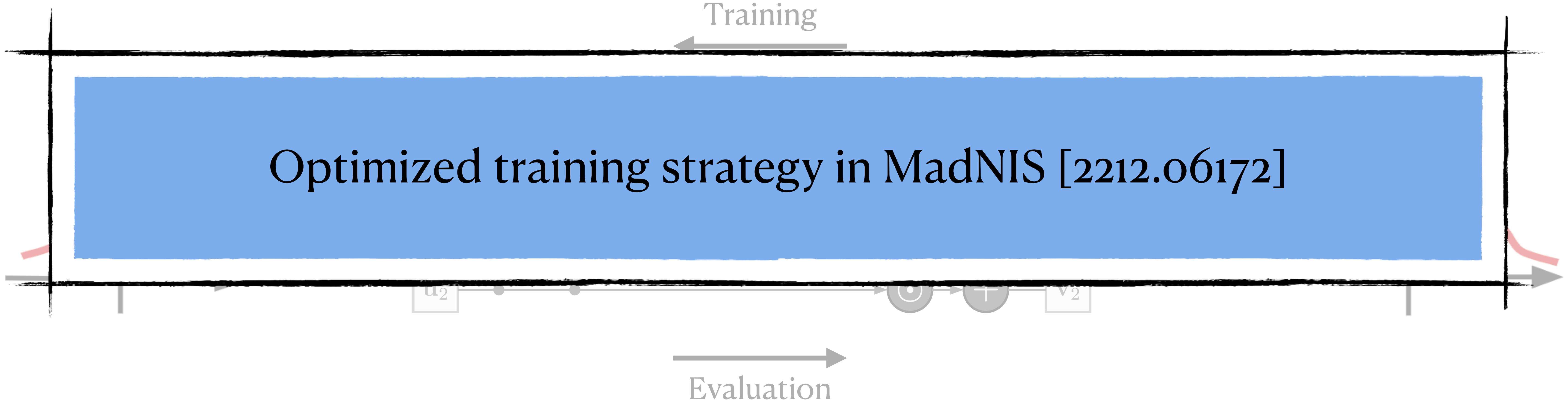
→ Maximize the log-likelihood

$$\begin{aligned}\mathscr{L} &= \log p(\theta\,|\,x) \\ &= \log p(z\,|\,\theta) + \log J_{NN} + p(\theta)\end{aligned}$$

Normalizing flows

Invertible networks for complex transformations

+ Bijective mapping
+ Tractable Jacobian $\rightarrow p_x(x) = p_z(z) \cdot J_{NN}$
+ Fast evaluation in both direction



Optimized training strategy in MadNIS [2212.06172]

Training on density $t(x)$

$\rightarrow$ Minimize difference

$$\mathscr{L} = \log p_x(x)/t(x)$$
$$= \log p_z(z(x))\, J_{NN}\, /\, t(x)$$

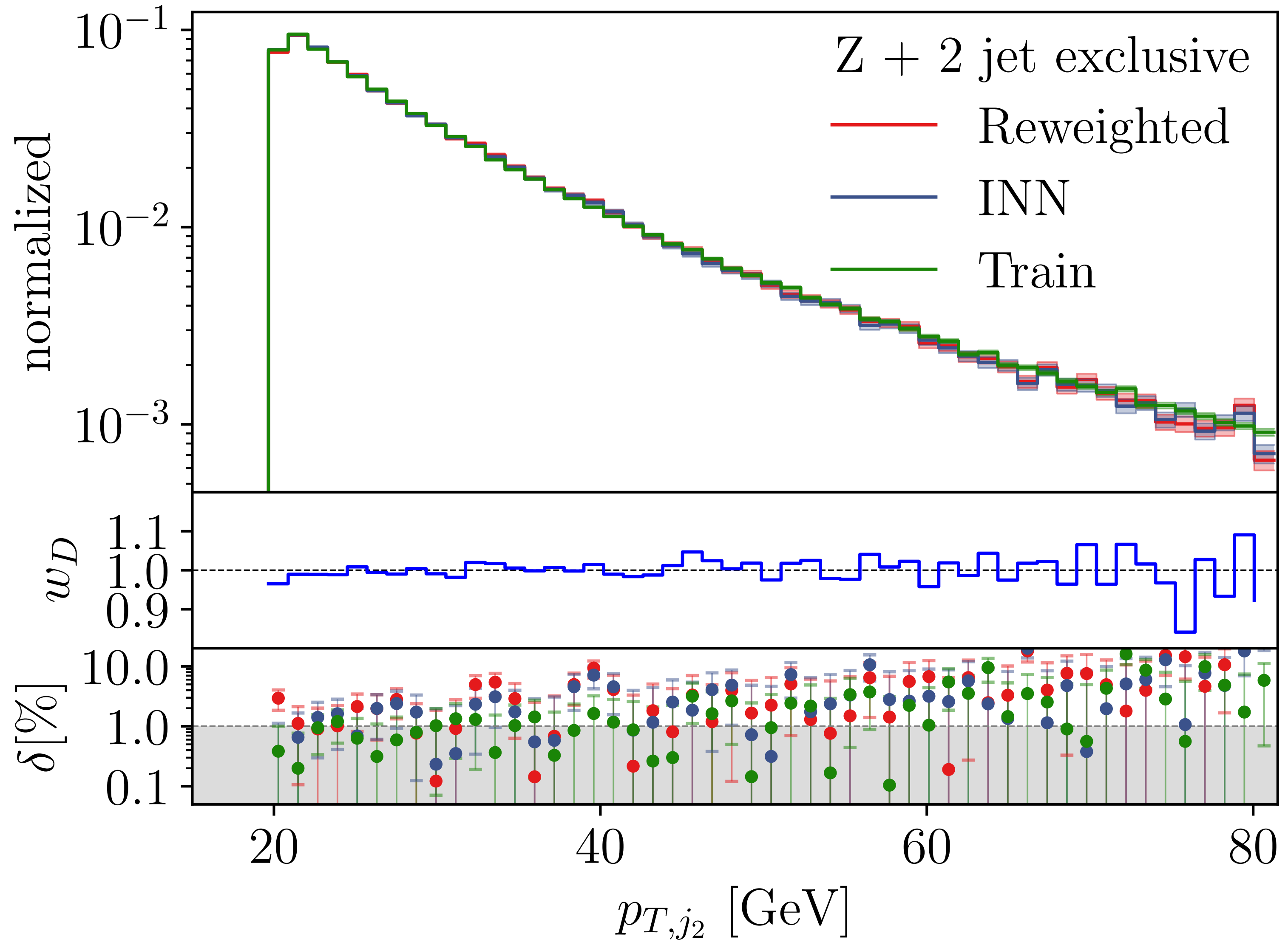
Training on samples x

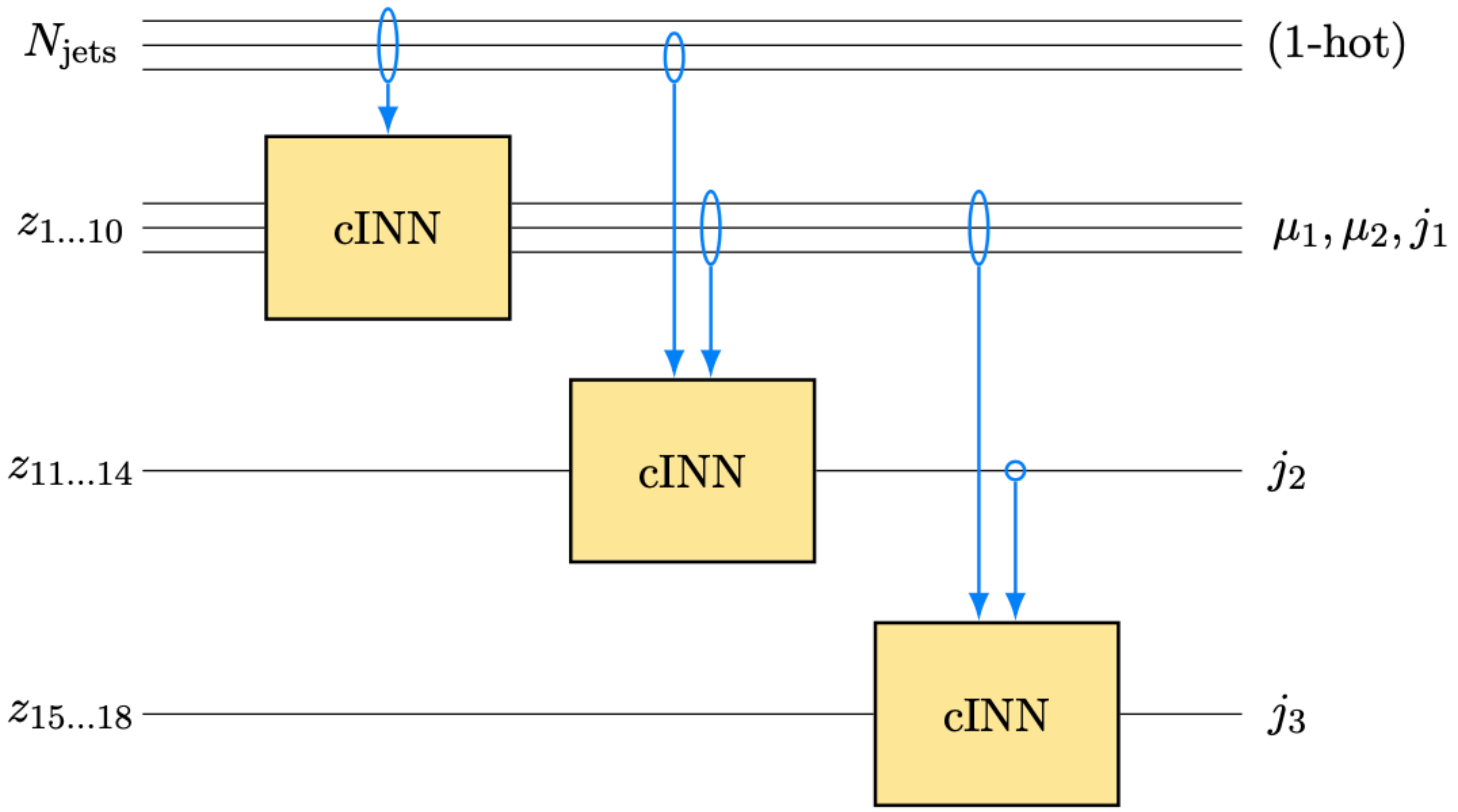
$\rightarrow$ Maximize the log-likelihood

$$\mathscr{L} = \log p(\theta \,|\, x)$$
$$= \log p(z \,|\, \theta) + \log J_{NN} + p(\theta)$$

Putting flows to work

Event generation



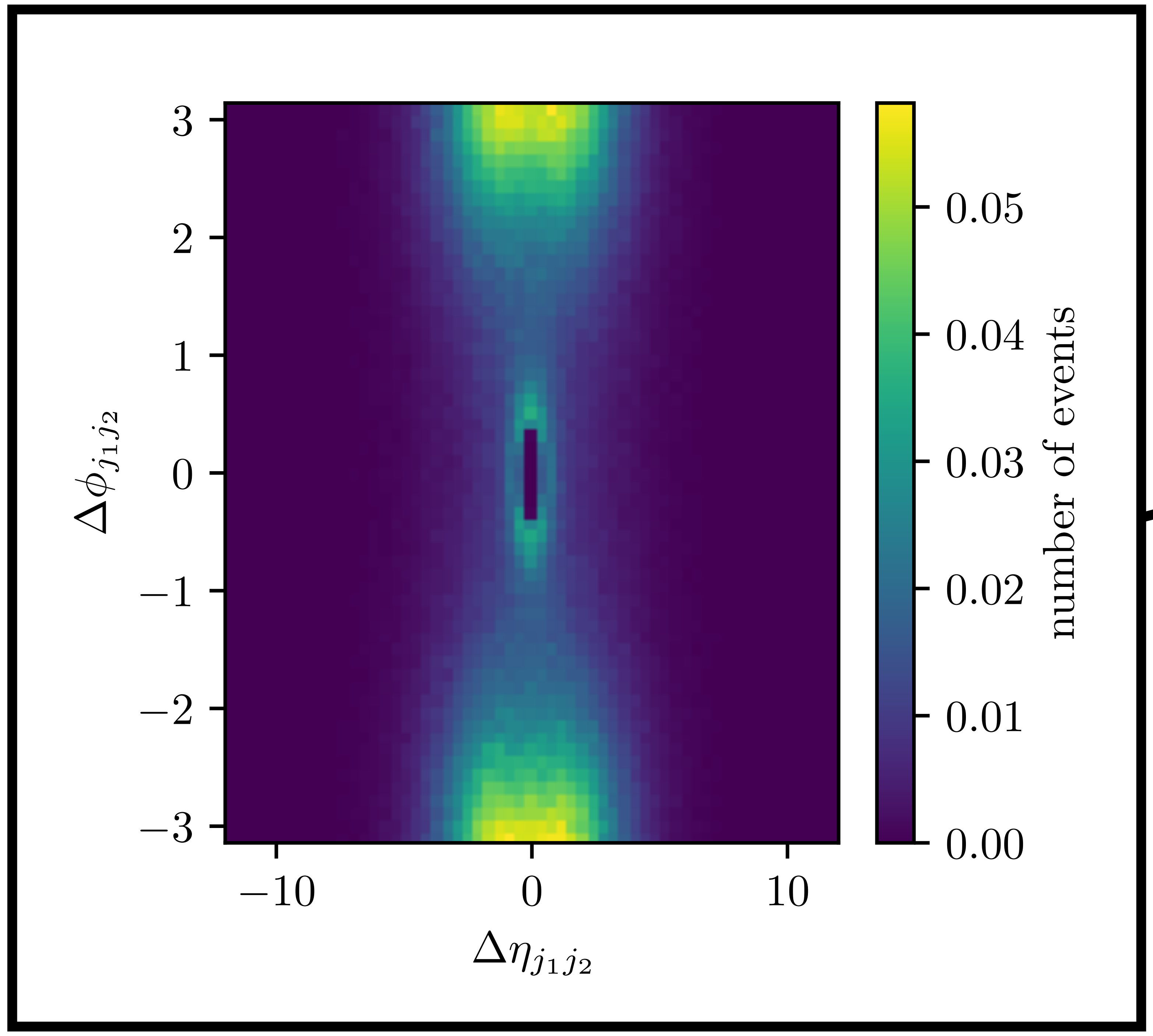


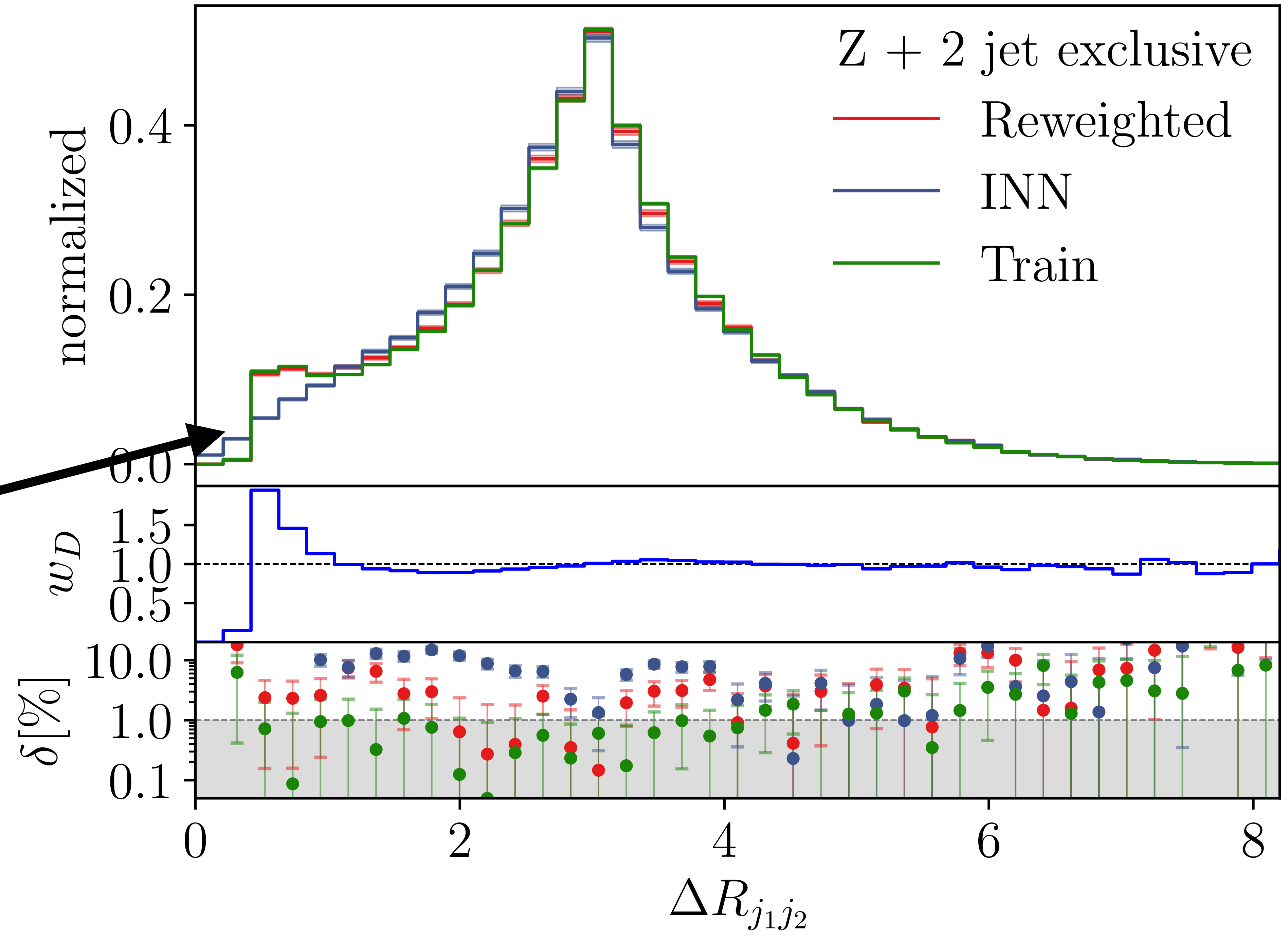
- Train normalizing flow on 4-momenta
- Include symmetries in feature representation
- Excellent performance for direct output

- Extend setup vor variable jet multiplicity

Challenges for normalizing flows

- Narrow features
- Topological holes (eg ΔR cuts)
  - no bijective mapping possible
  - can only be approximated





Reweighting for Precision

- Classifier loss

$$\mathscr{L} = -\sum_{x\sim p_{data}} \log(D(x)) - \sum_{x\sim p_{INN}} \log(1 - D(x))$$

$$= -\int \mathrm{d}x\, p_{data}(x)\, \log(D(x)) + p_{INN}(x)\, \log(1 - D(x))$$

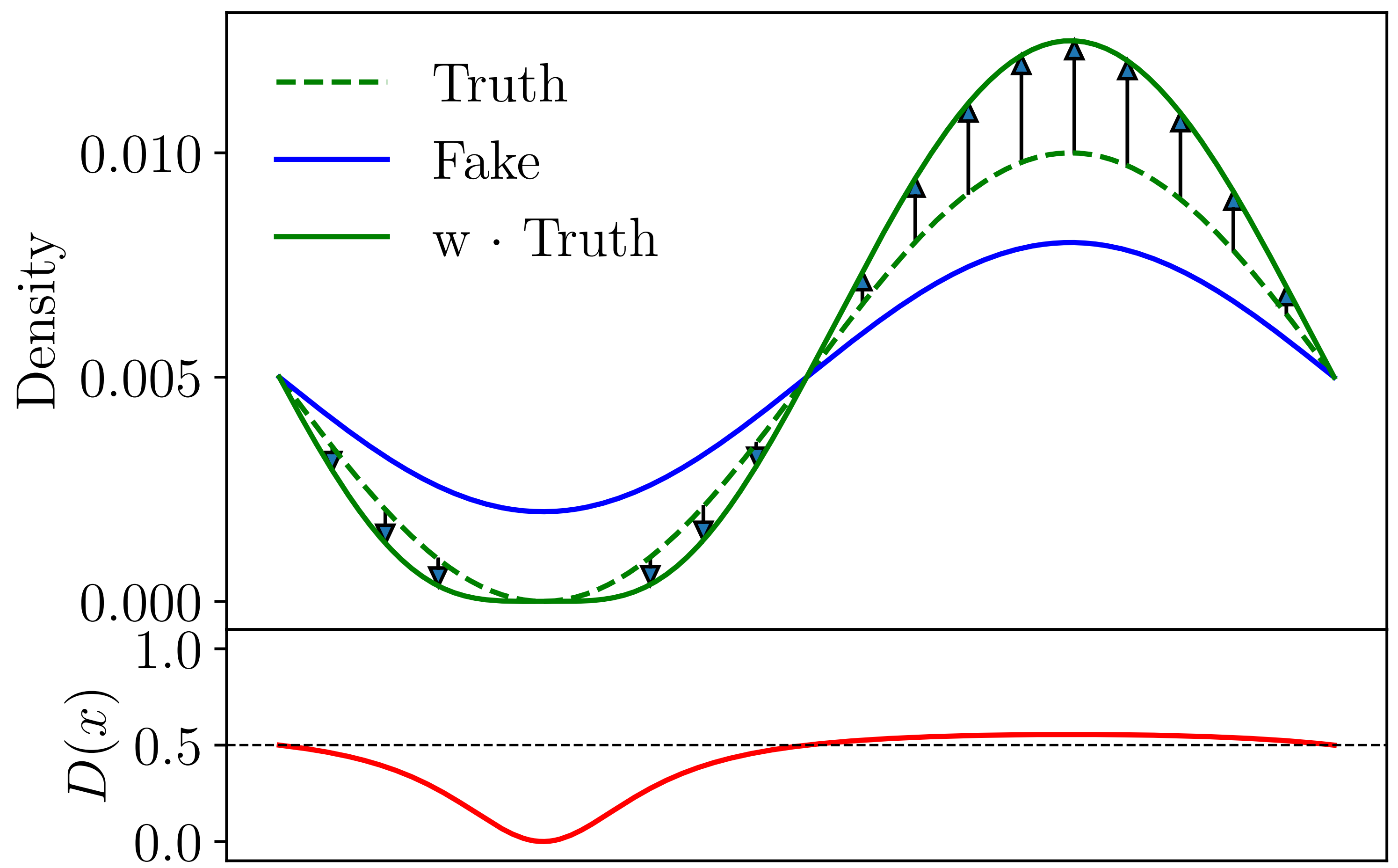
- Upon convergence obtain **reweighting factor**

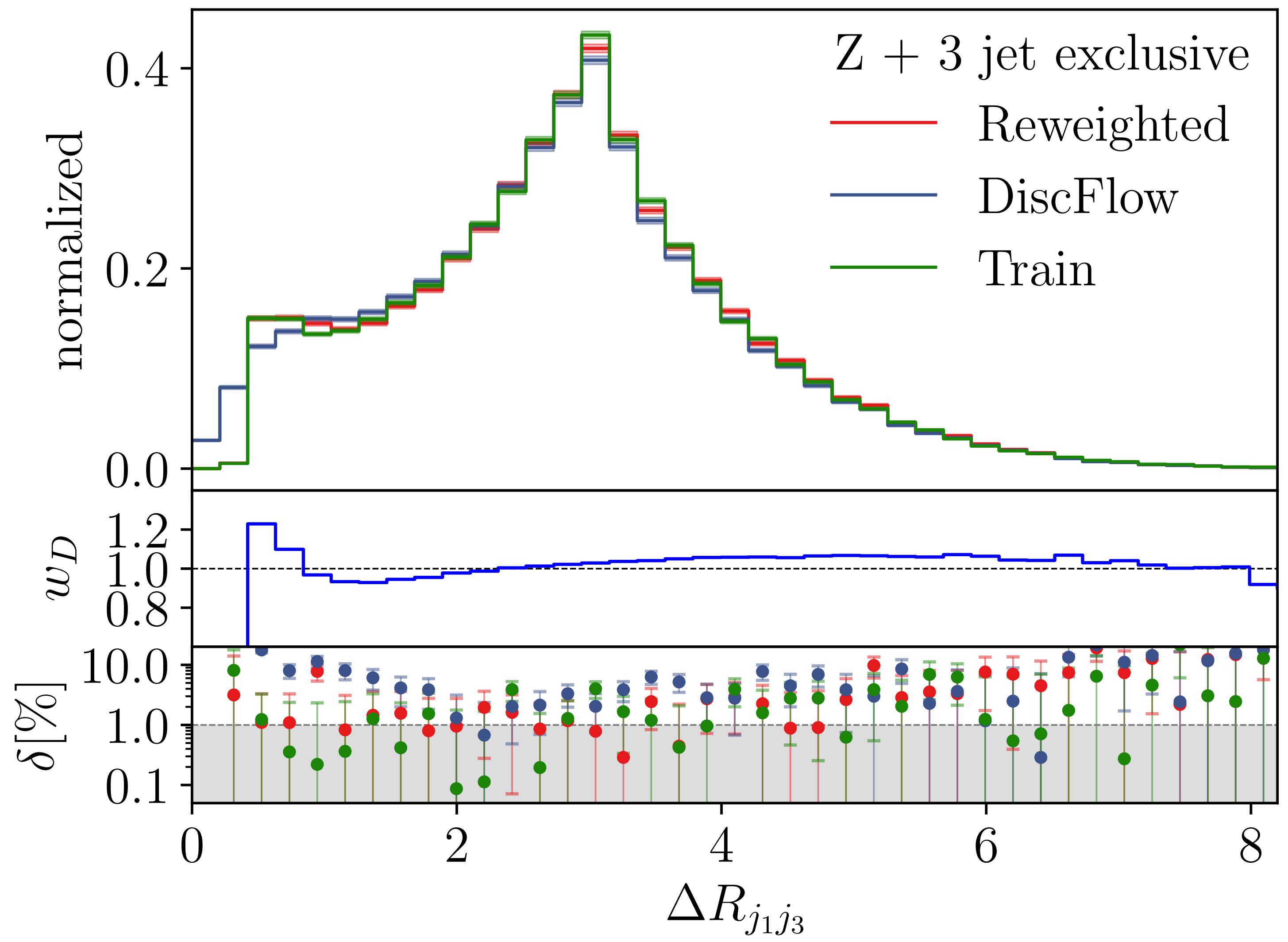
$$\Rightarrow \frac{p_{data}(x)}{p_{INN}(x)} = \frac{D(x)}{1 - D(x)} = w_D$$

- Use classifier feedback to enhance gradients

$$\mathscr{L}_{\text{DiscFlow}} \approx \int \mathrm{d}x \quad \underbrace{w_D(x)^\alpha P(x)}_{\text{reweighted truth}} \left(\frac{\psi(x;c)^2}{2} - \log J(x) \right)$$

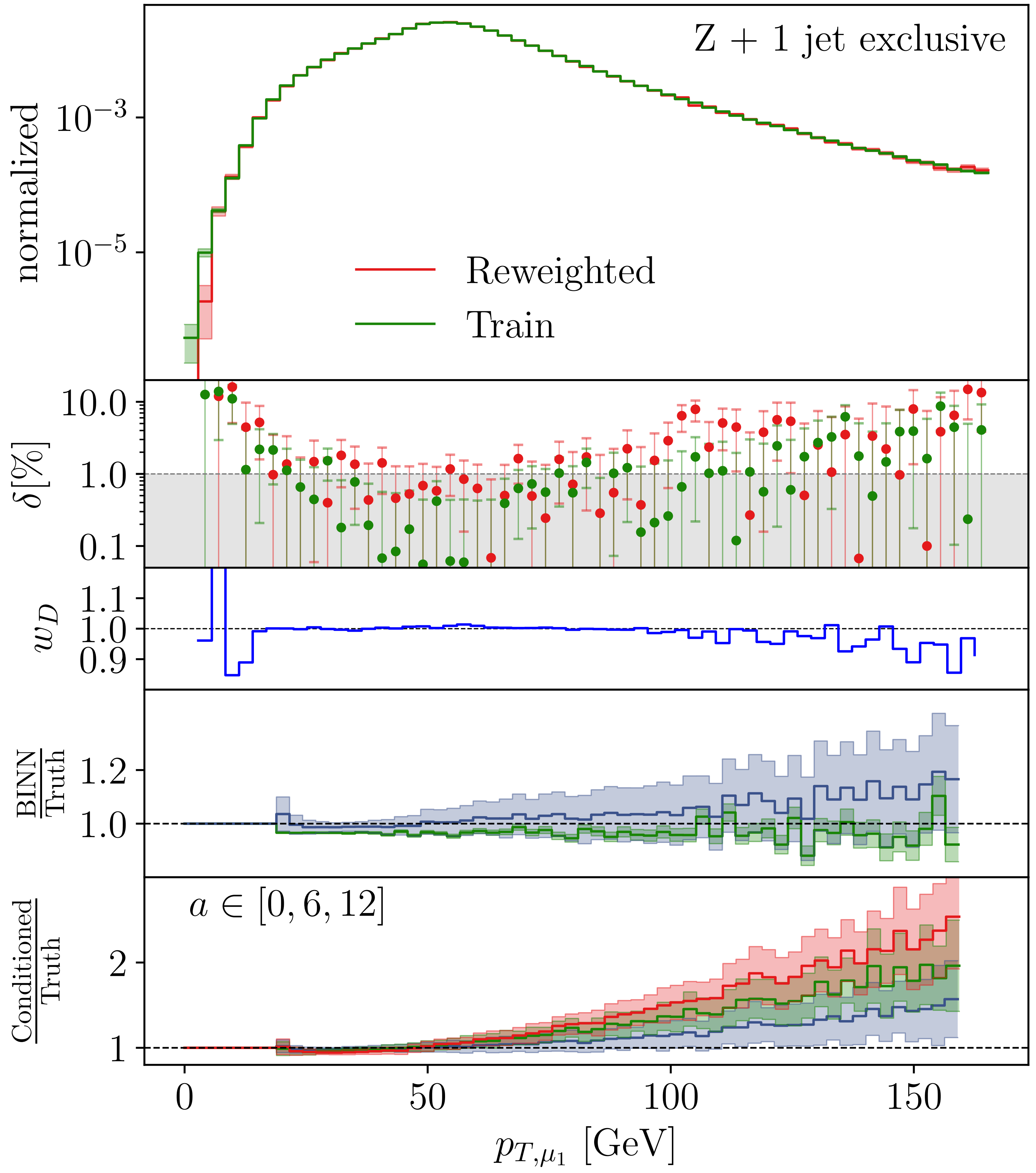
$\Rightarrow$ Reduces range of reweighting factors





Putting flows to work

Event generation

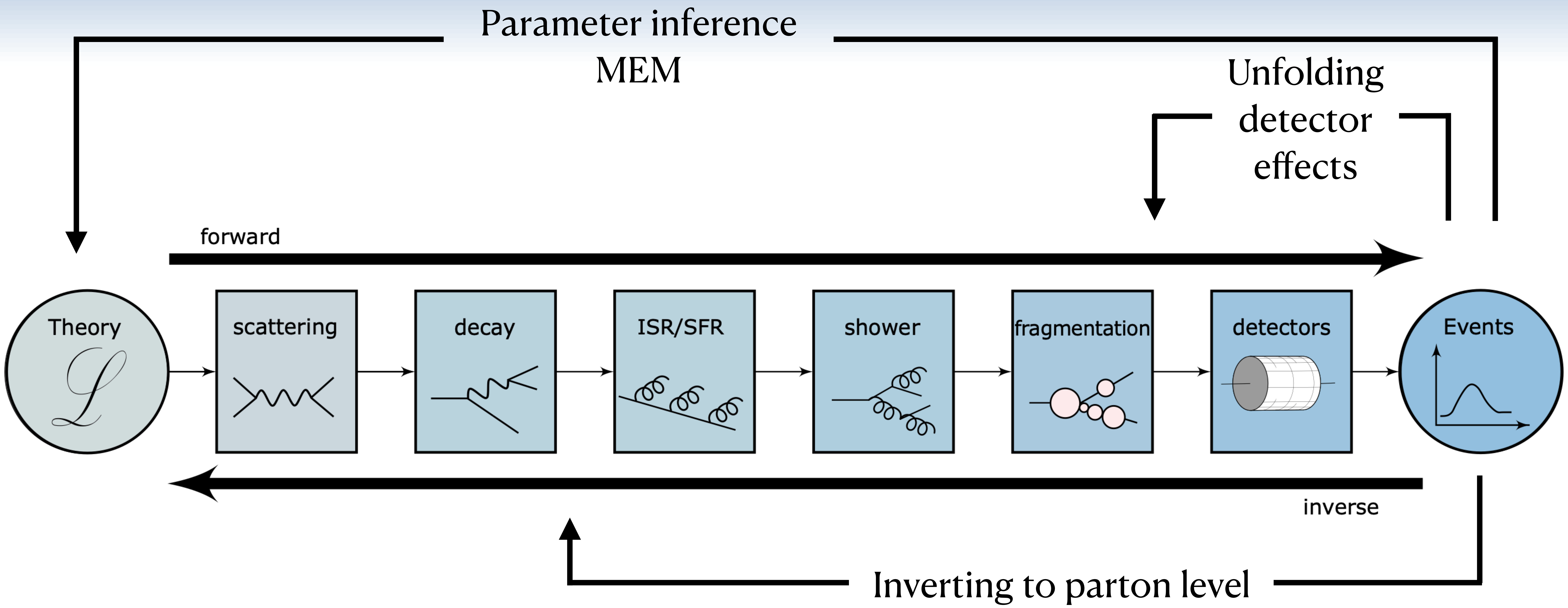


- Basis: INN
  - Phase space symmetries in architecture
- Control via classifier D
  - $$\frac{p_{\text{truth}}(x)}{p_{\text{INN}}(x)} = \frac{D(x)}{1 - D(x)}$$
- Precision via reweighting
  - Correct deviations of p_{INN}

➡ Uncertainty estimation via Bayesian NN

➡ Uncertainty propagation via conditioning

Inverting the simulation chain

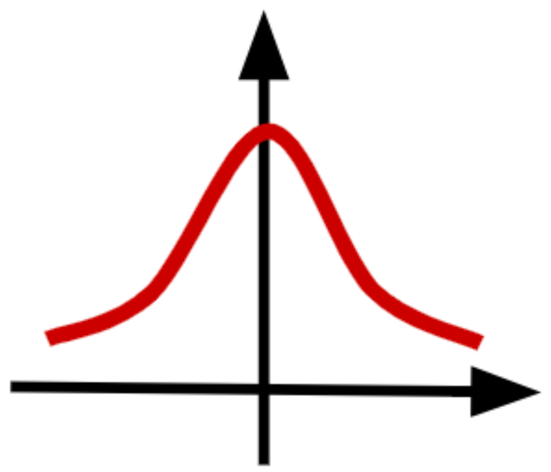


Requirements

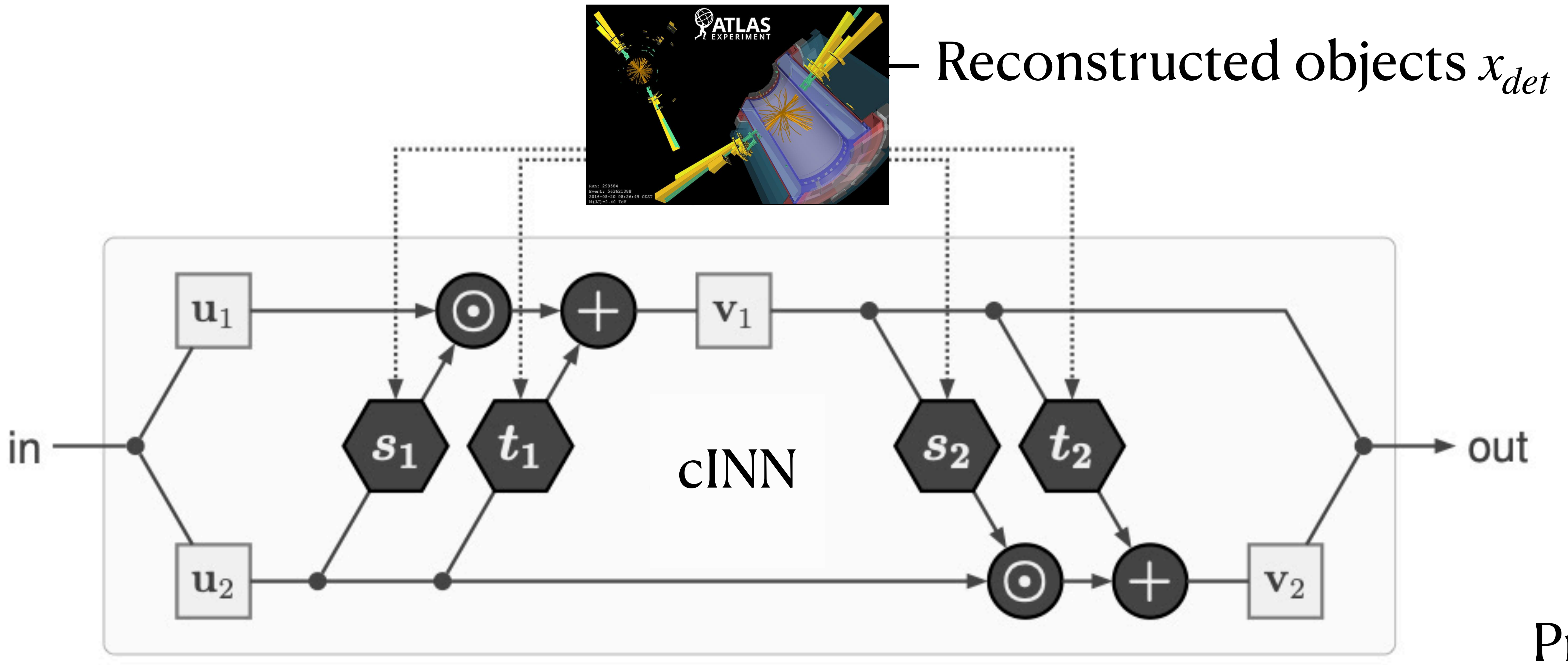
- [ ] Highdimensional
- [ ] Bin independent
- [ ] Statistically well defined

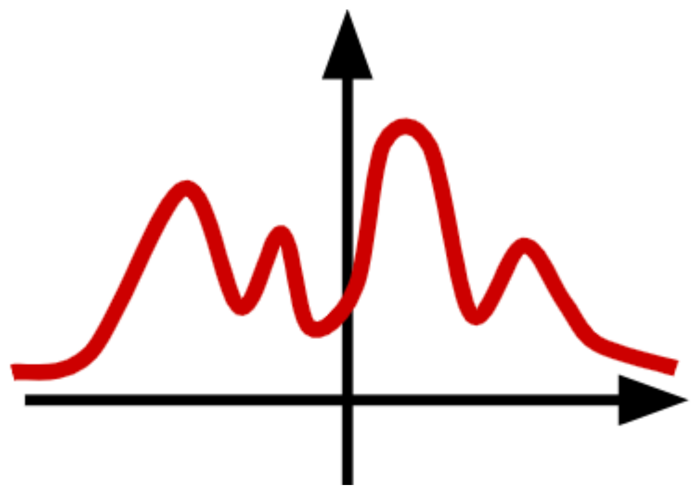
cINN unfolding

Given a reconstructed event:
What is the probability distribution at particle level?



Normal distribution
$z \sim \mathcal{N}$





Probability distribution
$x \sim P_{\text{parton}}$

Training

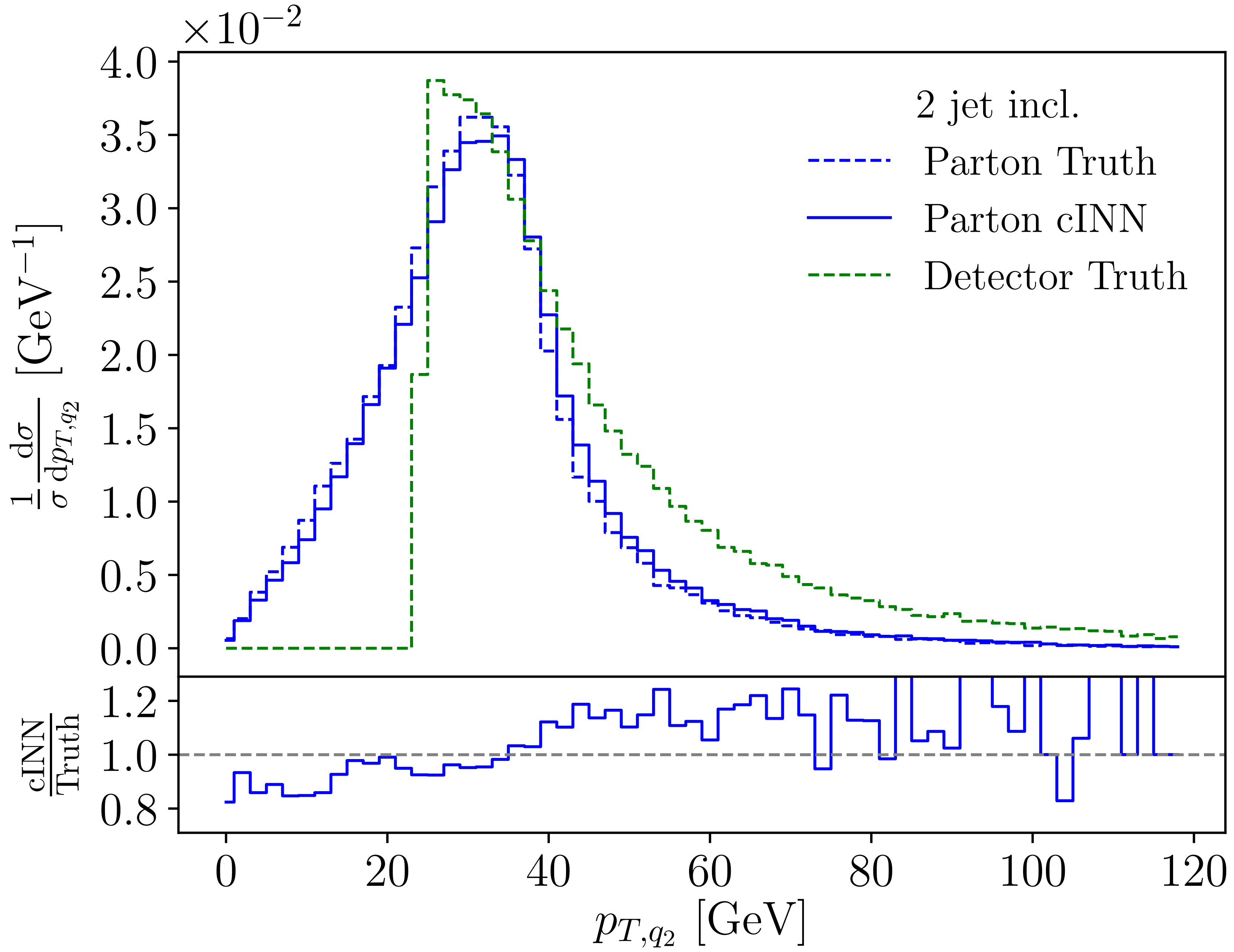
Unfolding

$$\begin{aligned}\mathscr{L} &= \log p(\theta \,|\, x_{part}, x_{det}) \\ &= \log p(z(x) \,|\, \theta) + \log J_{NN} + p(\theta)\end{aligned}$$

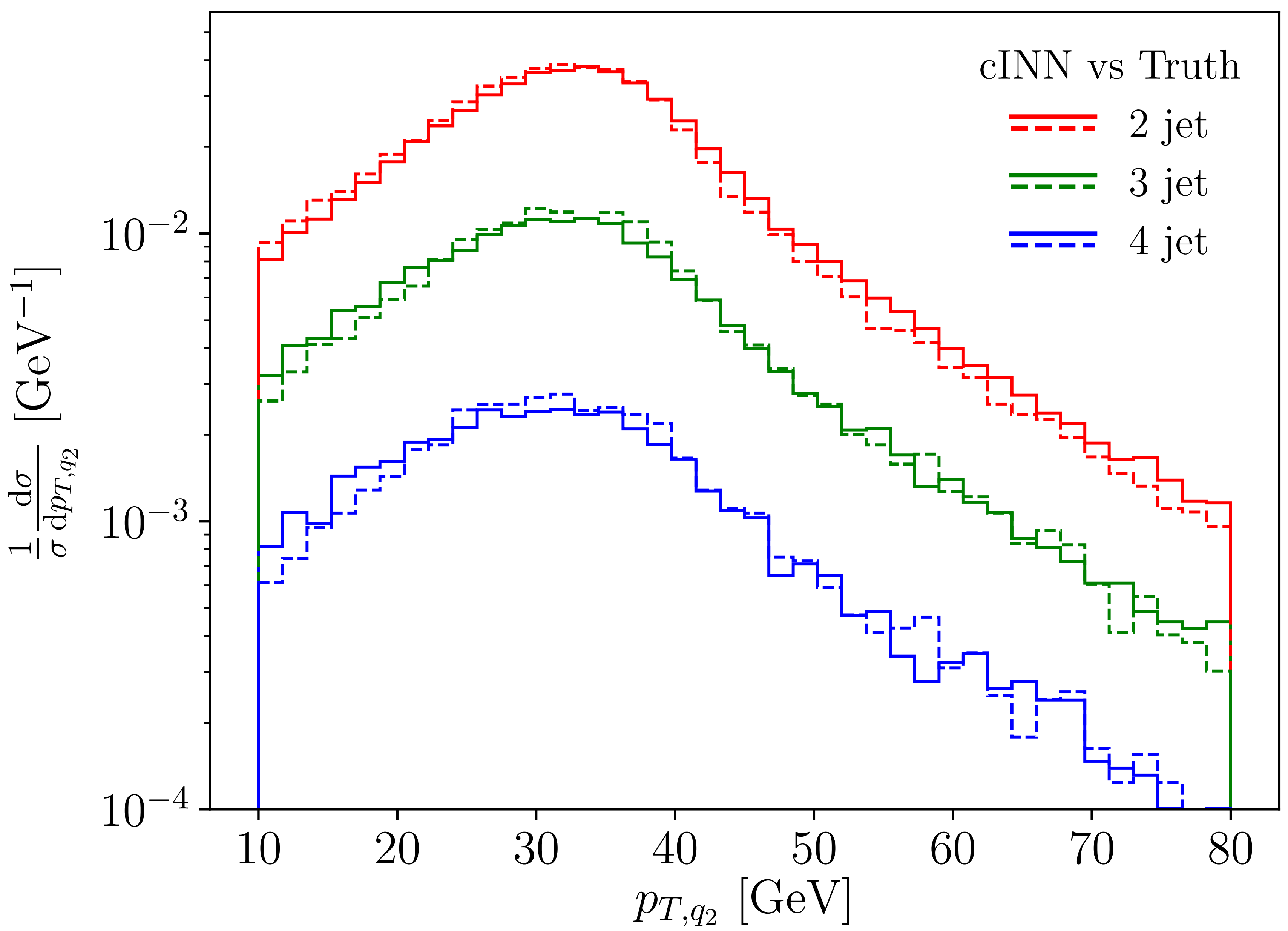
Inverting inclusive distributions

$pp > WZ > q\bar{q}l^+l^- +$ ISR $\rightarrow$ 2/3/4 jet events

Training on inclusive dataset



Evaluate exclusive 2/3/4 jet events

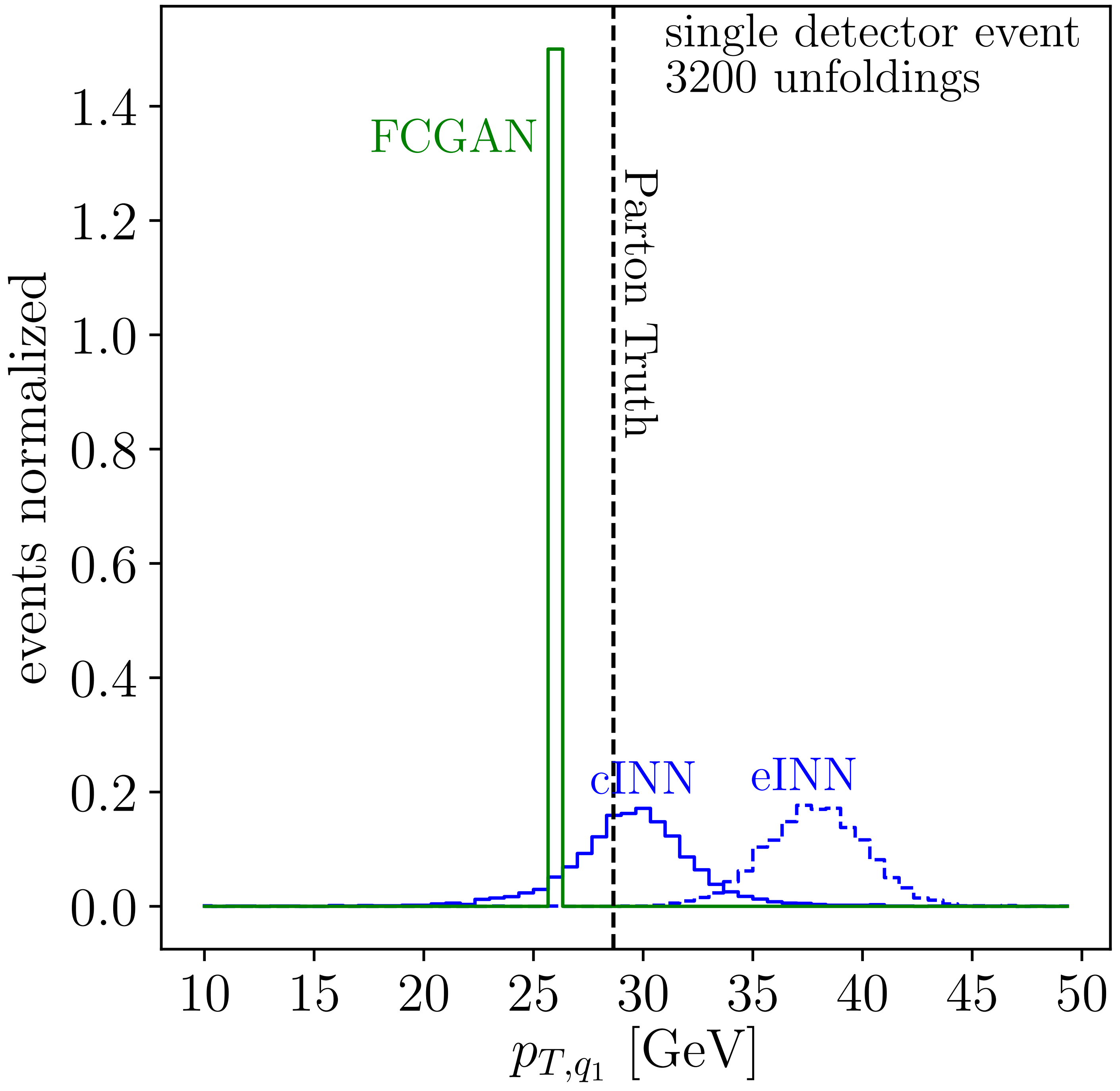


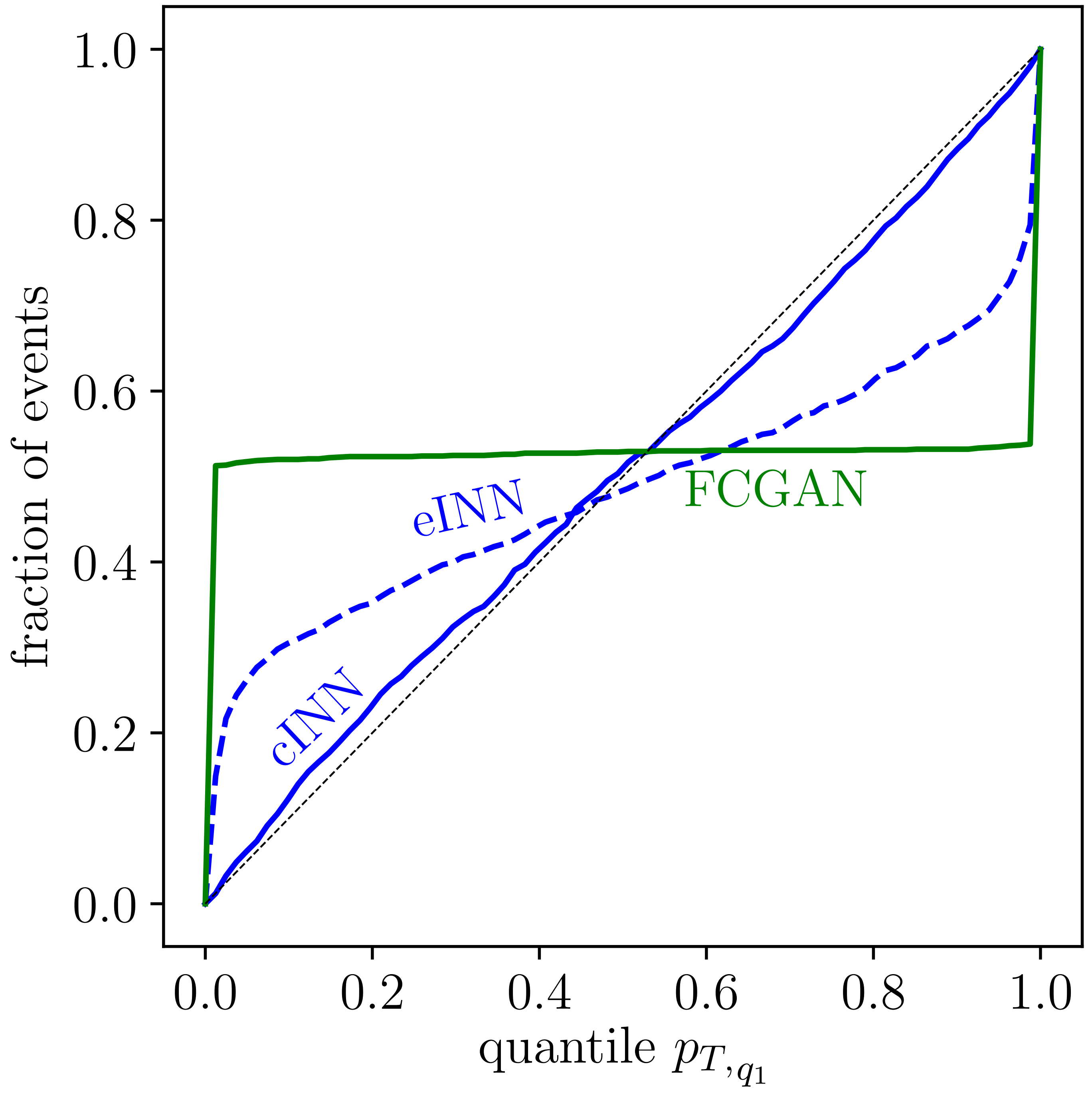
- ☑ High-dimensional
- ☑ Bin-independent
- ☐ Statistically well defined ?

M. Bellagente et al. [2006.06685]

Event-wise unfolding

No deterministic mapping!
Check calibration of probability density for individual event unfolding





- ☑ High-dimensional
- ☑ Bin-independent
- ☑ Statistically well defined

M. Bellagente et al. [2006.06685]

Beyond unfolding: Enabling the MEM

2210.00019

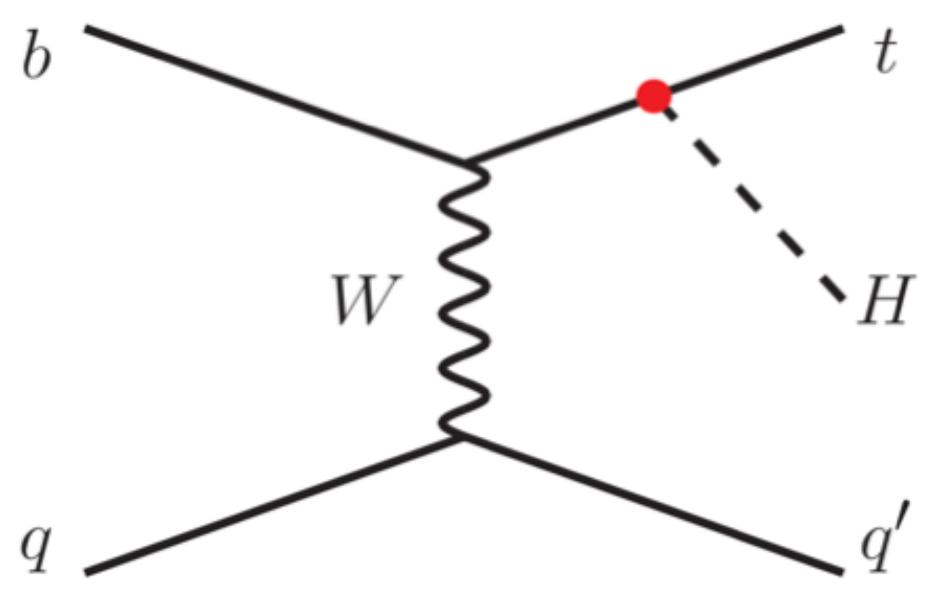
Matrix element method is based on untractable likelihood

$$p(x_{\text{reco}}|\alpha) = \int \mathrm{d}x_{\text{hard}}\ \underbrace{p(x_{\text{hard}}|\alpha)}_{\text{diff. CS}}\ \underbrace{p(x_{\text{reco}}|x_{\text{hard}}, \alpha)}_{\text{estimate with network}}$$

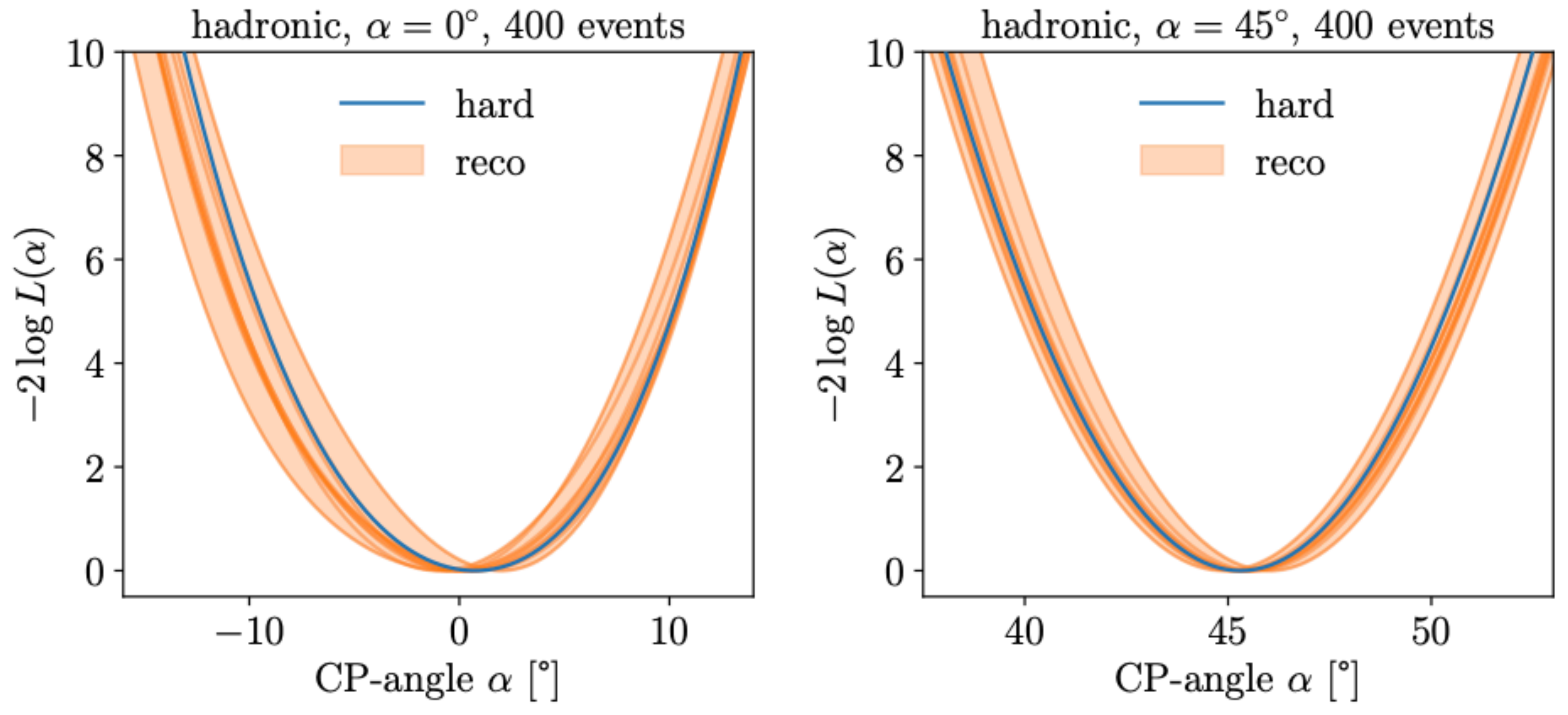
Problem: integration over full phase space of the hard scattering

Solution: Use unfolding cINN to sample x_{hard}

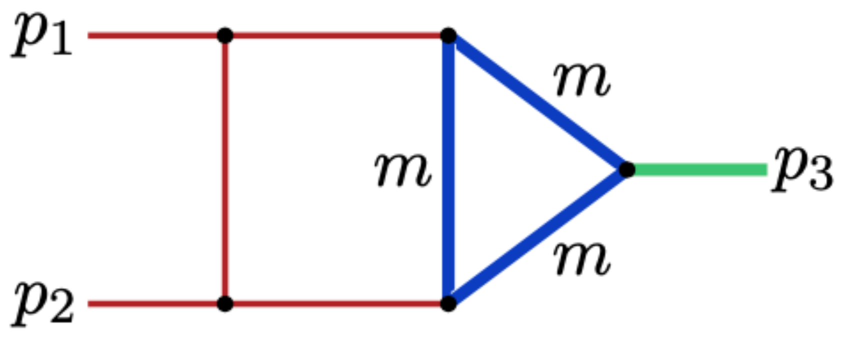
$$p(x_{\text{reco}}|\alpha) = \left\langle \frac{1}{q(x_{\text{hard}})}\ p(x_{\text{hard}}|\alpha)\ p(x_{\text{reco}}|x_{\text{hard}}, \alpha) \right\rangle_{x_{\text{hard}} \sim q(x_{\text{hard}})}$$

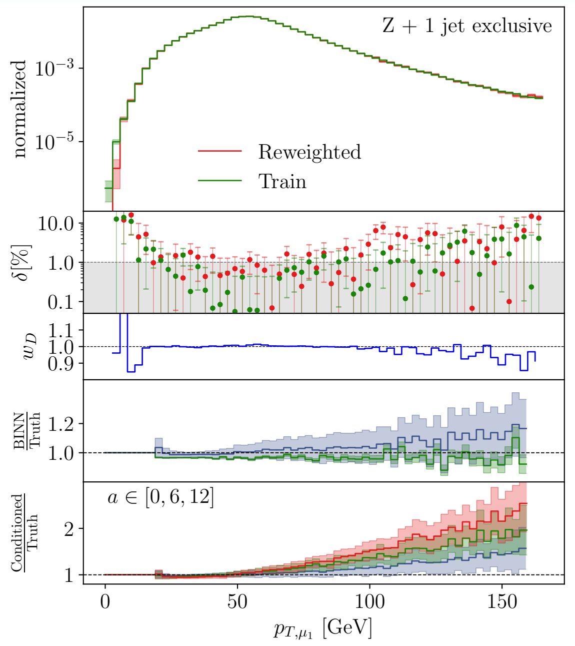


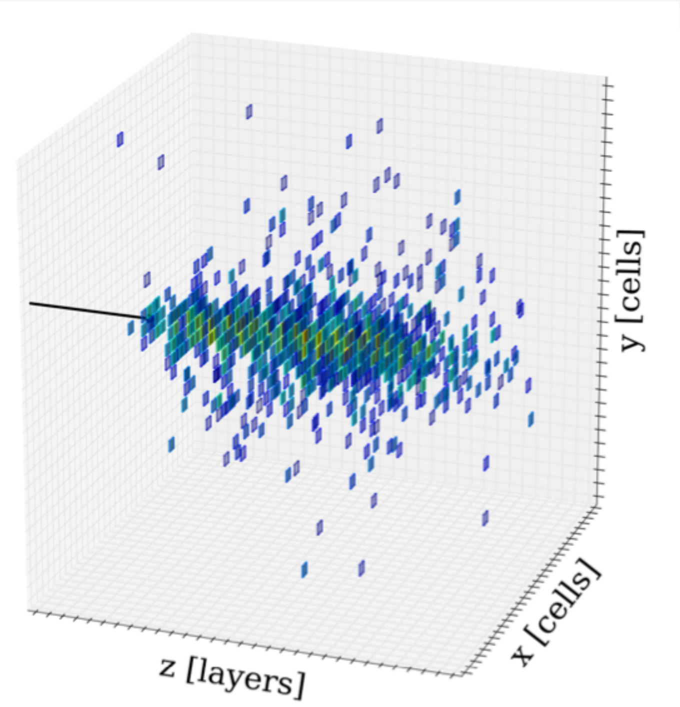
Single Higgs production
with anomalous non-CP conserving Higgs coupling

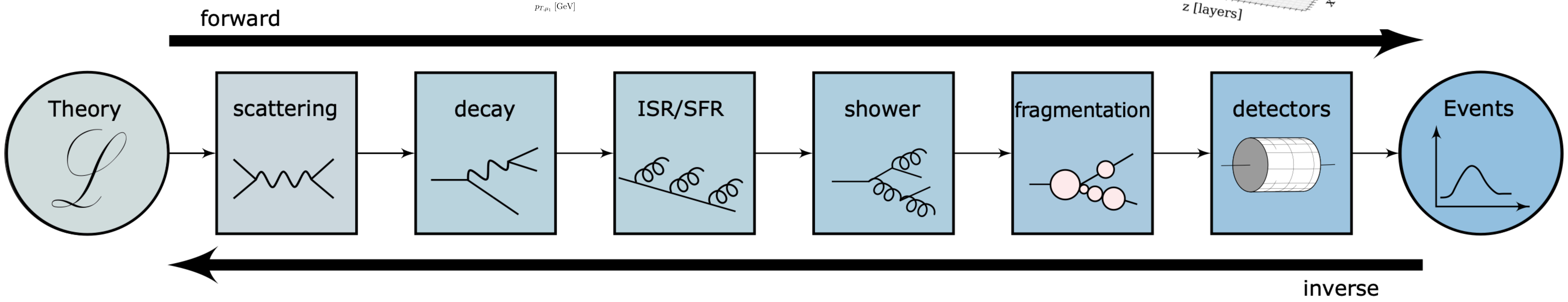


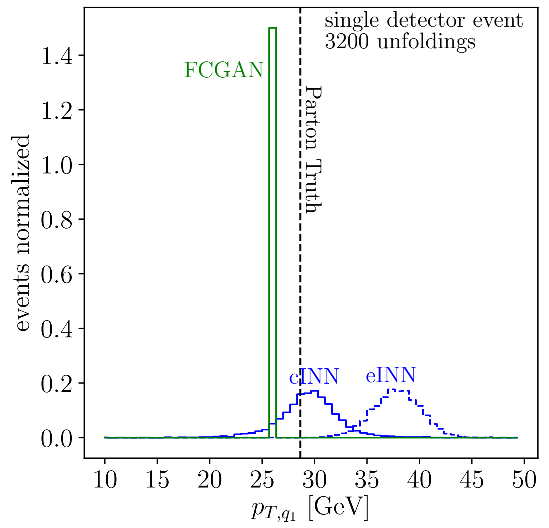
ML 4 LHC Event generation

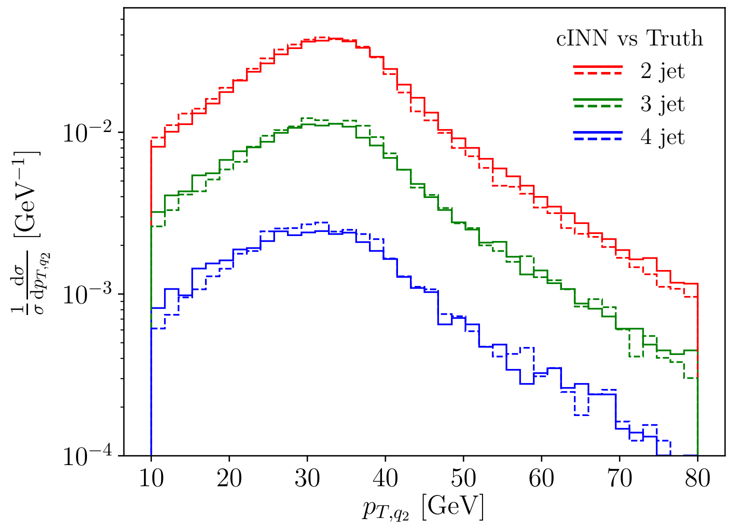


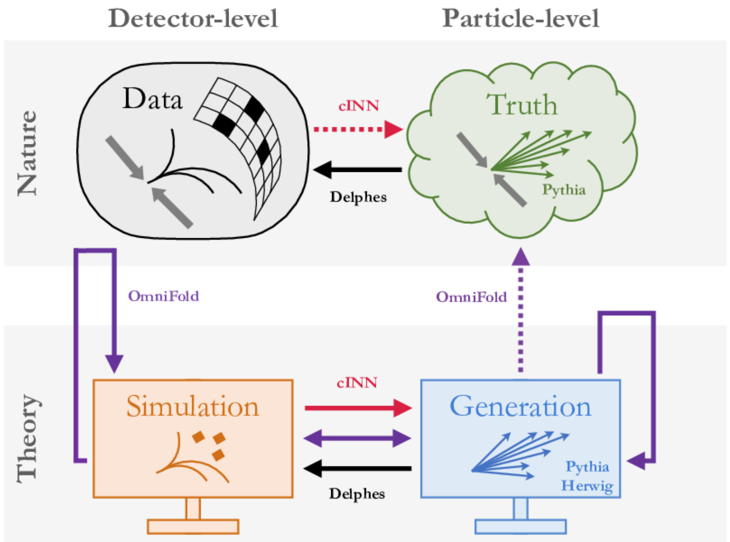












New data are currently on their way...