Loop amplitudes from Precision Networks

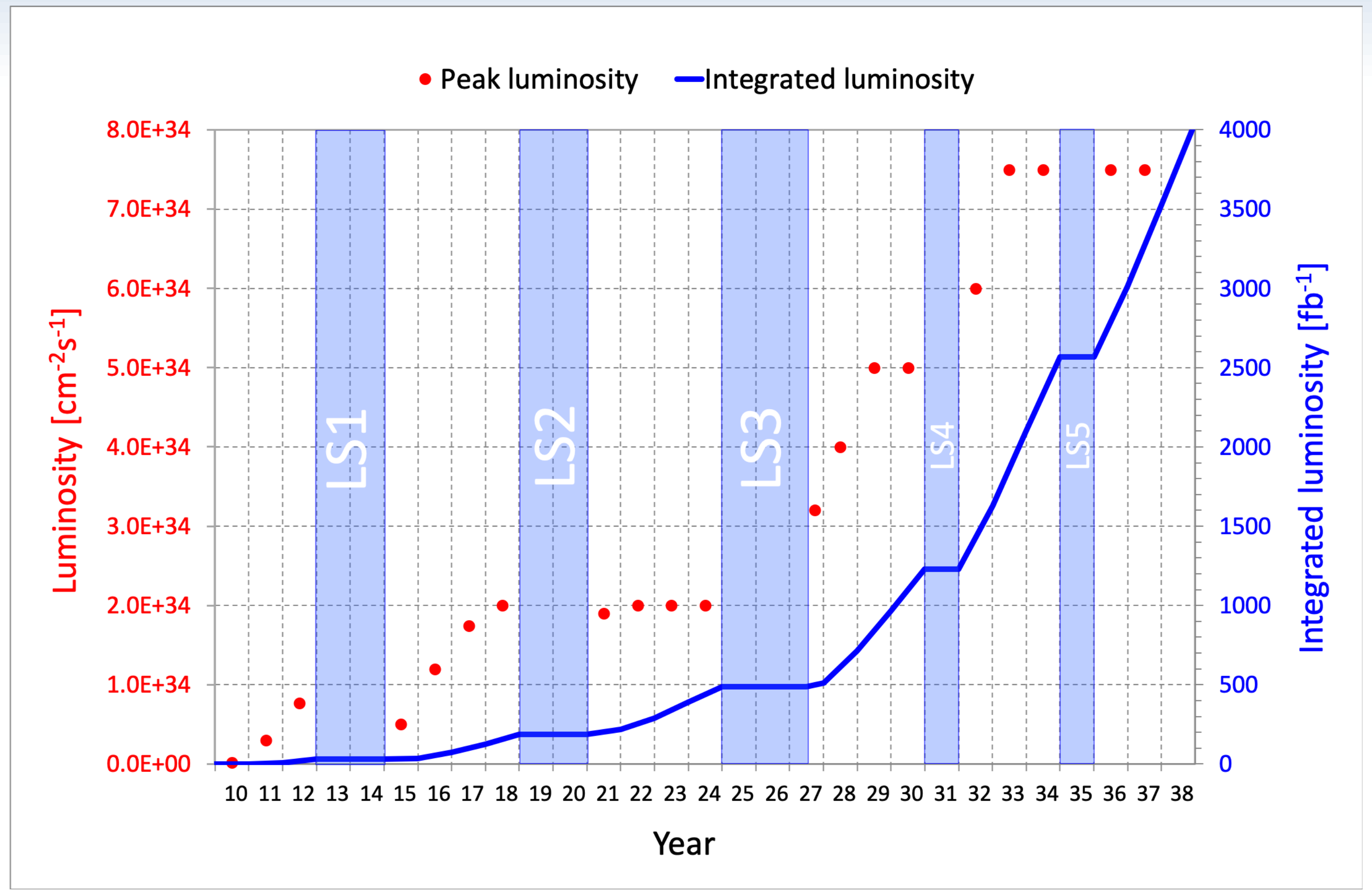
[2206.14831]

ACAT 2022

Anja Butter, LPNHE/ITP

Open questions towards HL-LHC

- 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**
  - Likelihood free inference
    - Optimal Observables, Unfolding
  - 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

$$w_{\text{event}} = \underbrace{f(x_1, Q^2) f(x_1, Q^2)}_{\text{PDF}} \times \underbrace{\mathscr{M}(x_1, x_2, p_1, \ldots, p_n)}_{\text{Matrix element}} \times \underbrace{J(p_i(r))}_{\text{Phase space mapping}}$$

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
     ➡ $J(p_i(r)) = (f \times \mathscr{M})^{-1}$

ML for Amplitudes
-
Approximation with Regression

Limitations of a standard network

Example

$gg \rightarrow \gamma\gamma g(g)$ @LO

90k training amplitudes
870k test amplitudes

Standard approach

Training data

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

Loss

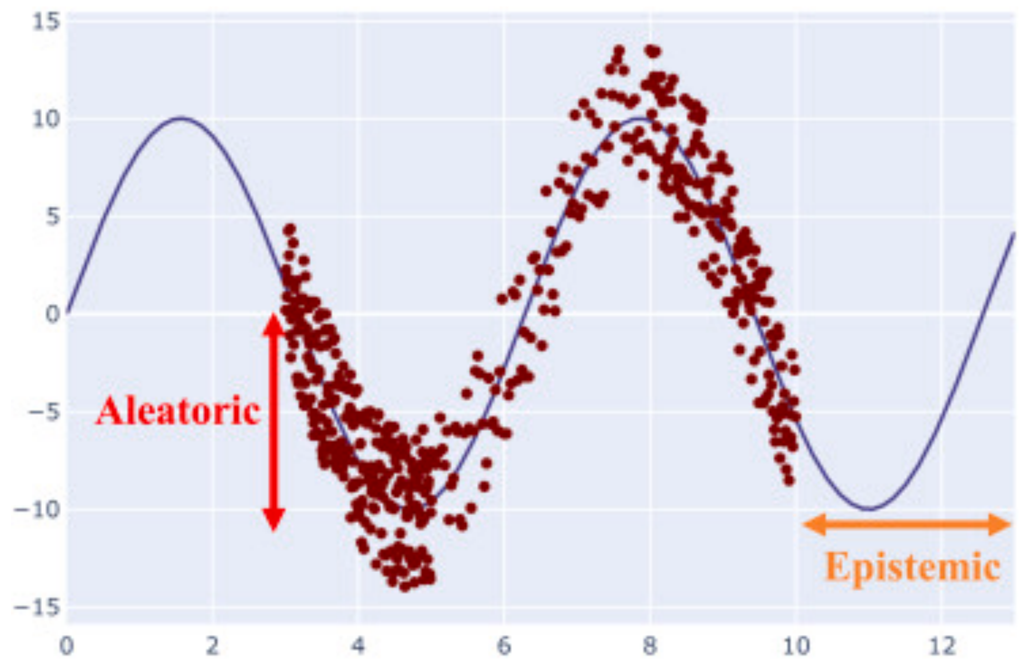
$\mathscr{L} = (A'(x) - NN(x))^2$

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

→ A better formulation of the problem

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

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

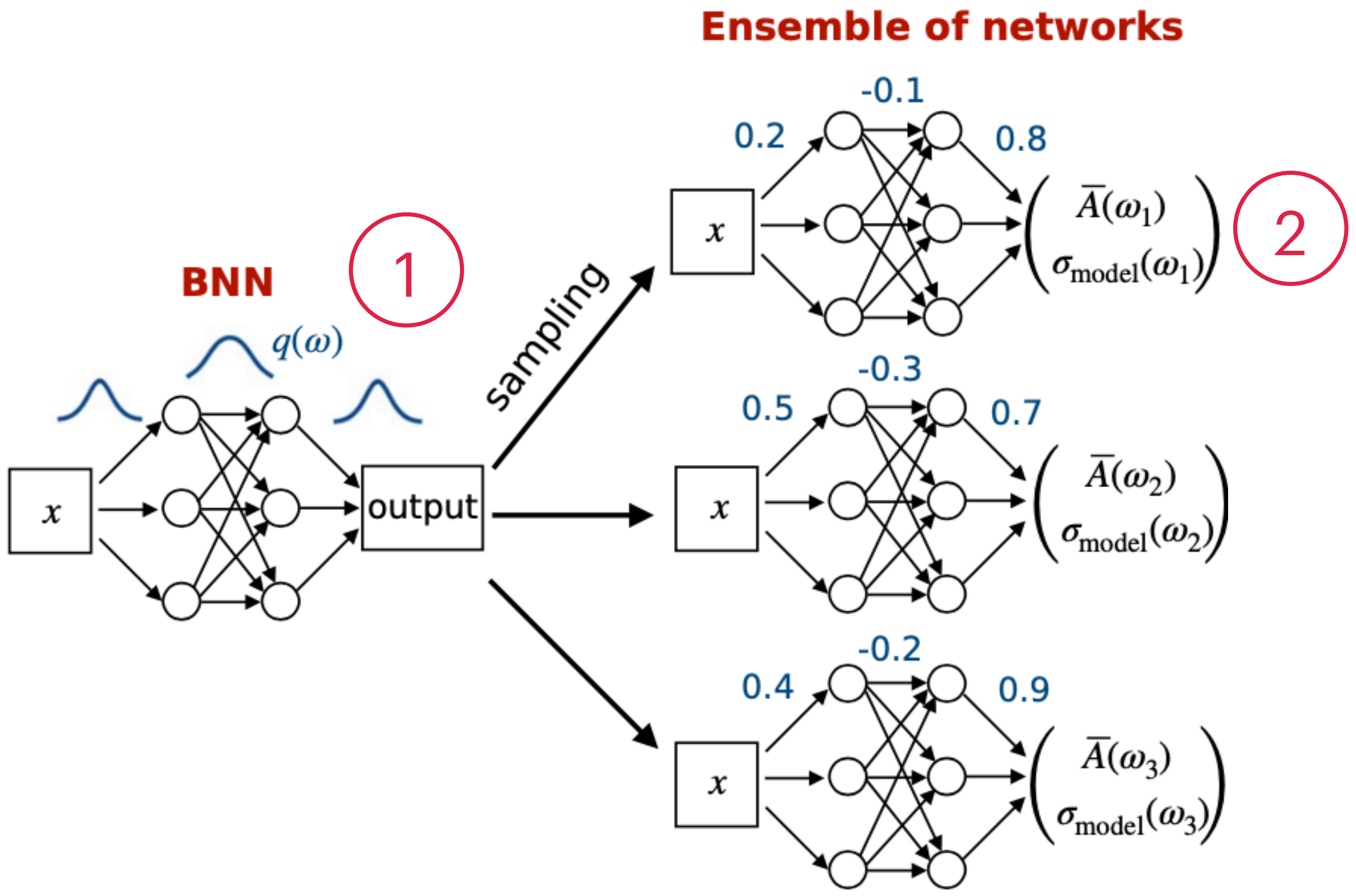


Capturing probabilities with Bayesian networks

$$p(A) = \int dw\ p(A\,|\,w)p(w\,|\,T) \approx \int dw\ p(A\,|\,w)q(w)$$

2 1

Bayesian network



Building the loss function

Approximate $q(w)$ by minimizing KL divergence

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

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

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

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

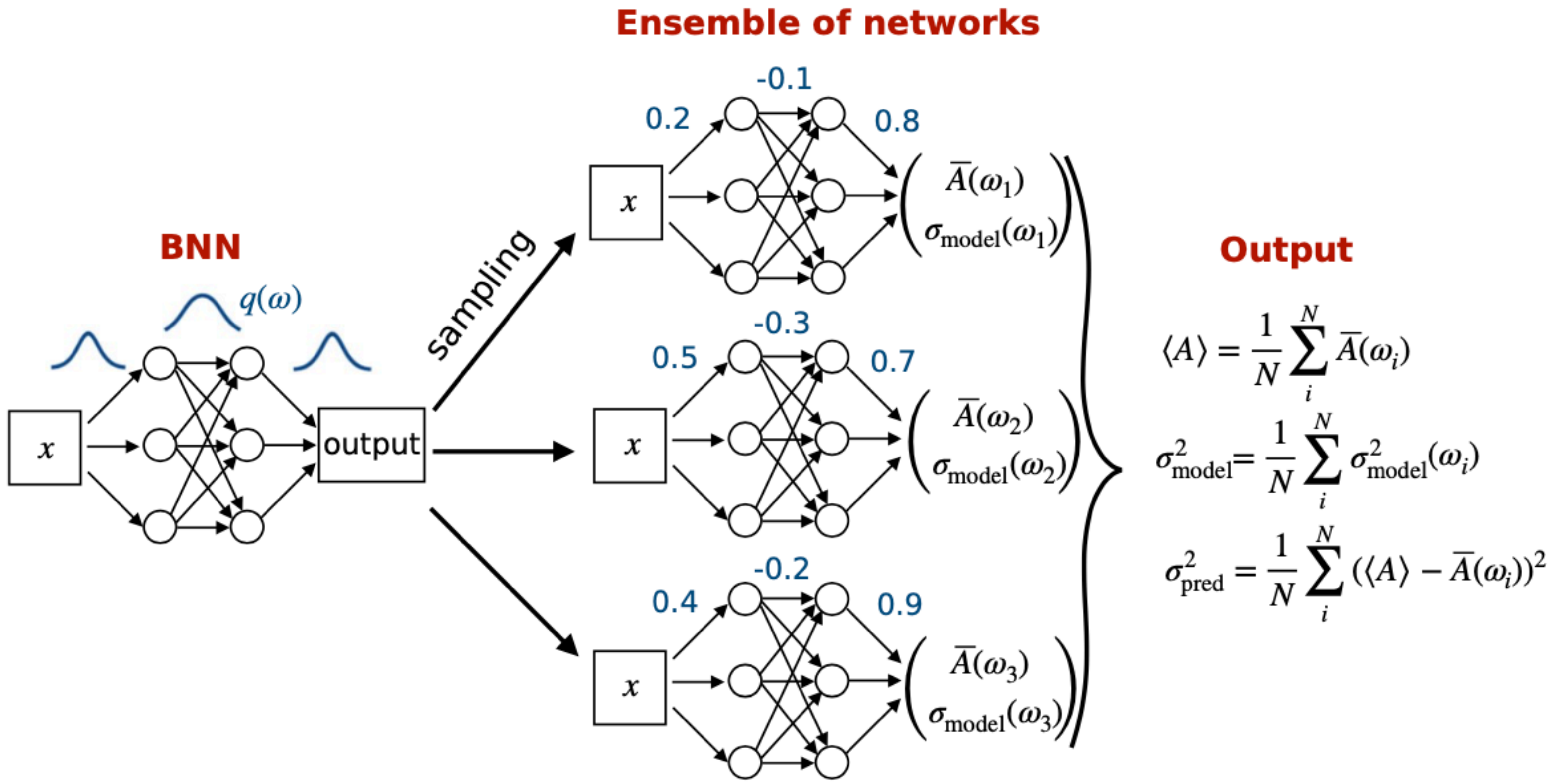
1 Gaussian prior

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

2 Gaussian uncertainty

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

How to obtain A and σ



$$\langle A\rangle = \int dA\, d\omega\, A\, p(A\,|\,\omega, T)\, q(\omega)$$

$$\equiv \int d\omega\, q(\omega)\, \bar{A}(\omega) \qquad \text{with} \qquad \bar{A}(\omega) = \int dA\; A\, p(A\,|\,\omega)$$

Uncertainty splits into σ_{model} and σ_{pred}

$$\sigma^2_{\text{tot}} = \int dA\, d\omega\, \big(A - \langle A\rangle\big)^2\; p(A\,|\,\omega)\, q(\omega)$$

$$= \int d\omega\, q(\omega)\Big[\overline{A^2}(\omega) - \bar{A}(\omega)^2 + \big(\bar{A}(\omega) - \langle A\rangle\big)^2\Big]$$

$$\equiv \sigma^2_{\text{model}} + \sigma^2_{\text{pred}}$$

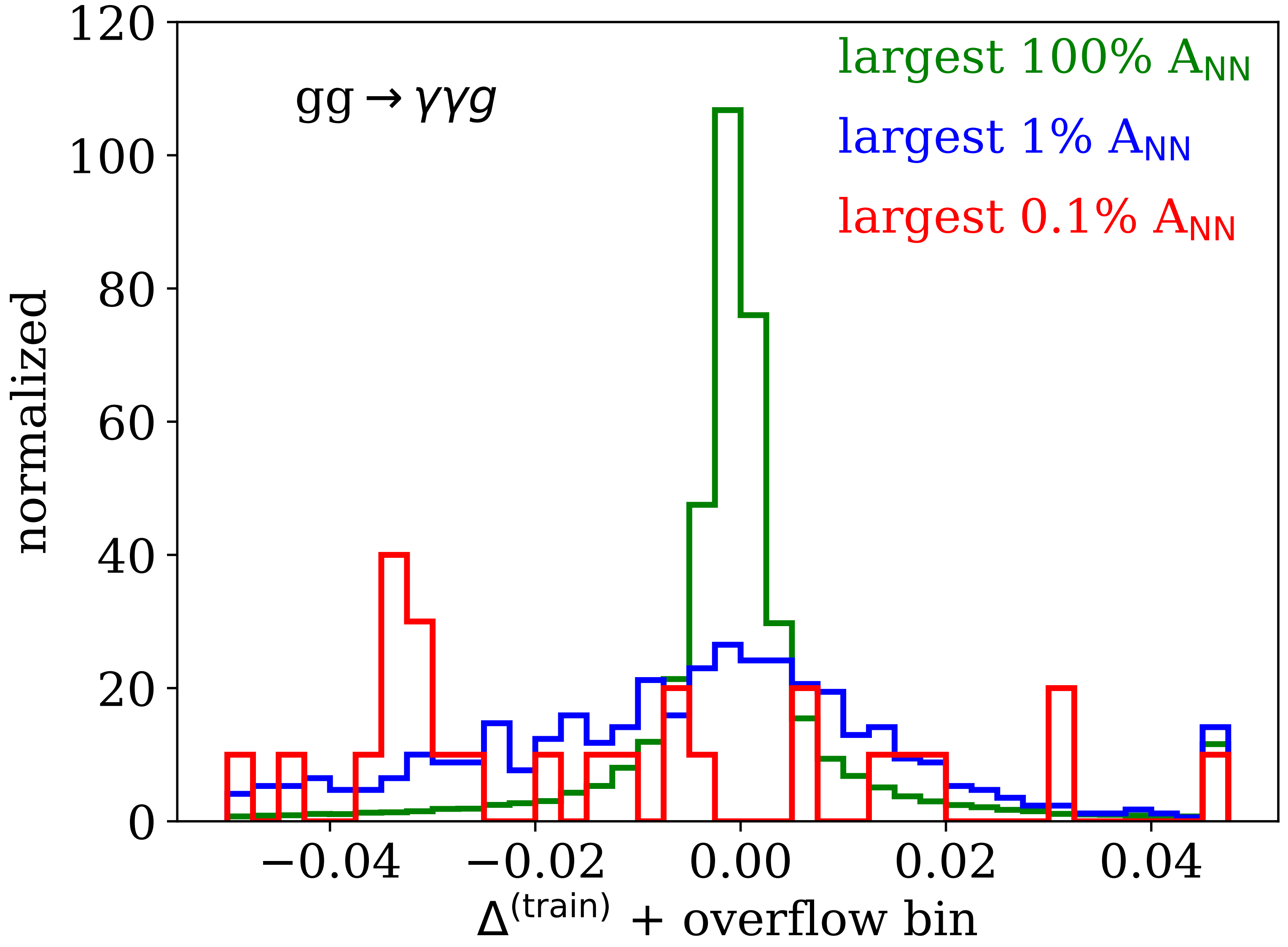
Data intrinsic (noise)* Convergence*

* not easily separable in the limit of exact training data

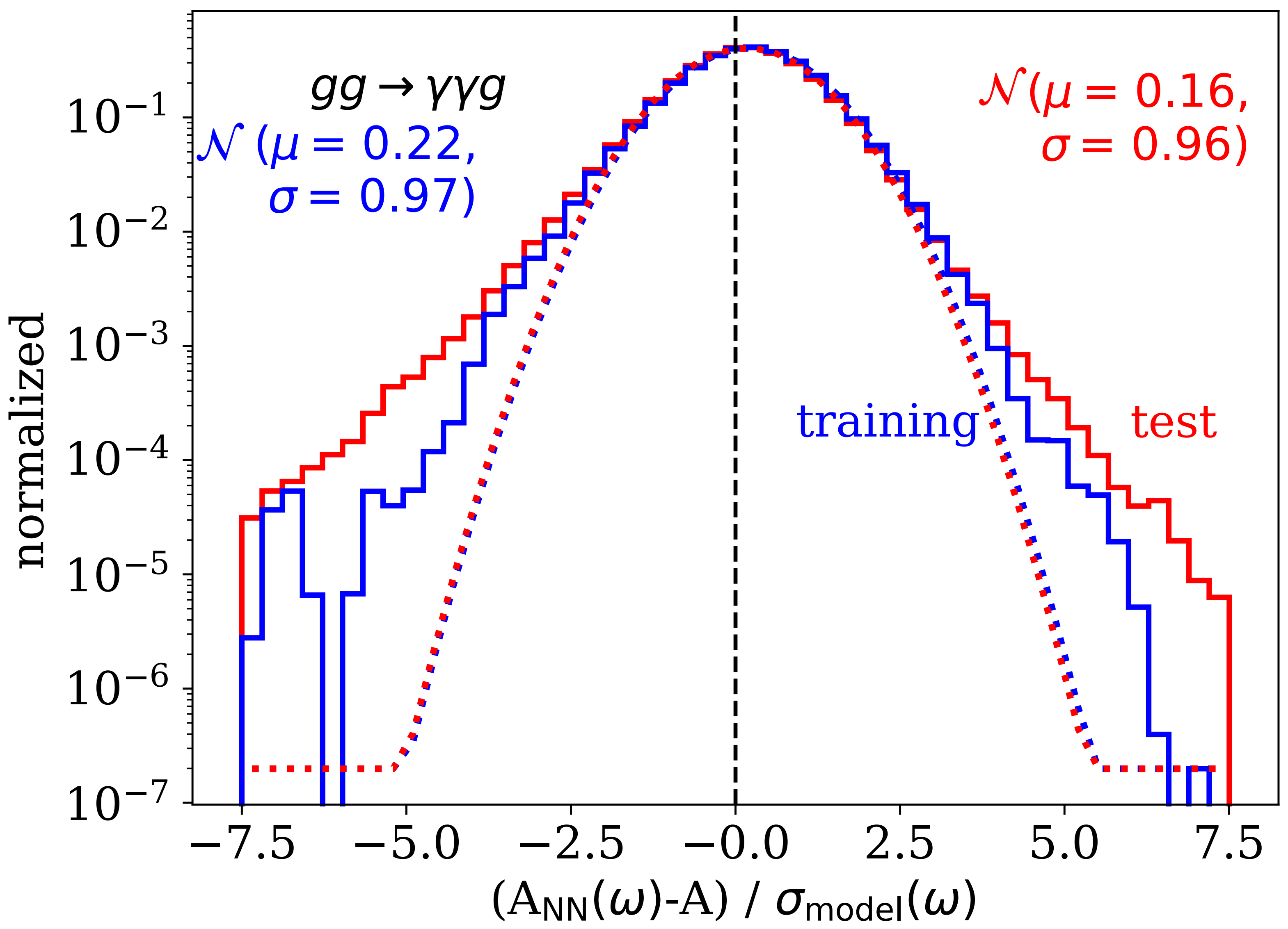
Results - out of the box

+ Deviations at 1 percent level

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



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



Performance worse for rare points with large amplitudes (collinear)

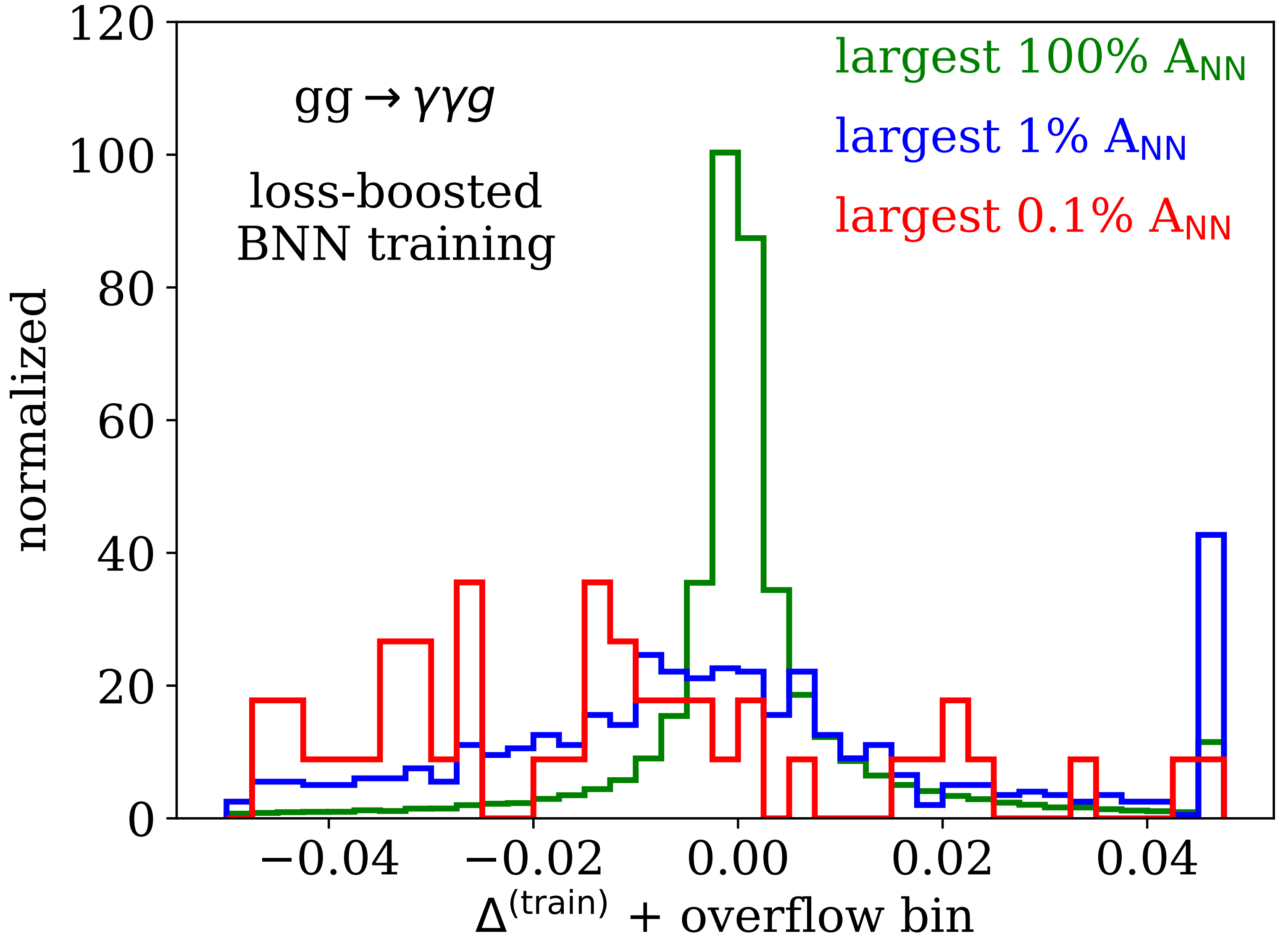
Roughly Gaussian but enhanced tails

Loss boosting

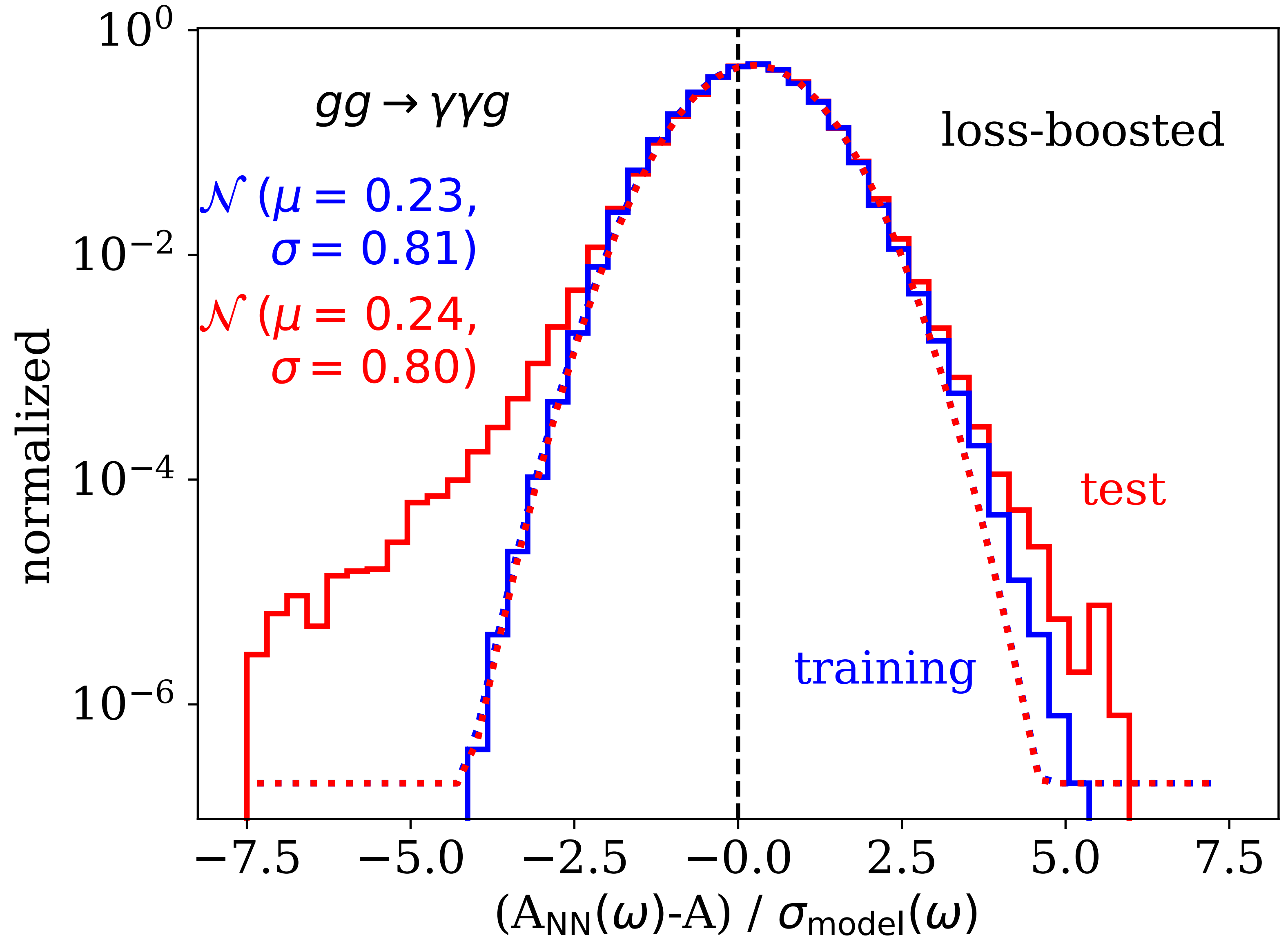
Enforce training on samples with $\Delta A > 2\sigma$

$\rightarrow$ include them 5 times in each epoch

$\rightarrow$ Repeat 4 times



No change in performance



Tails reproduced for training data

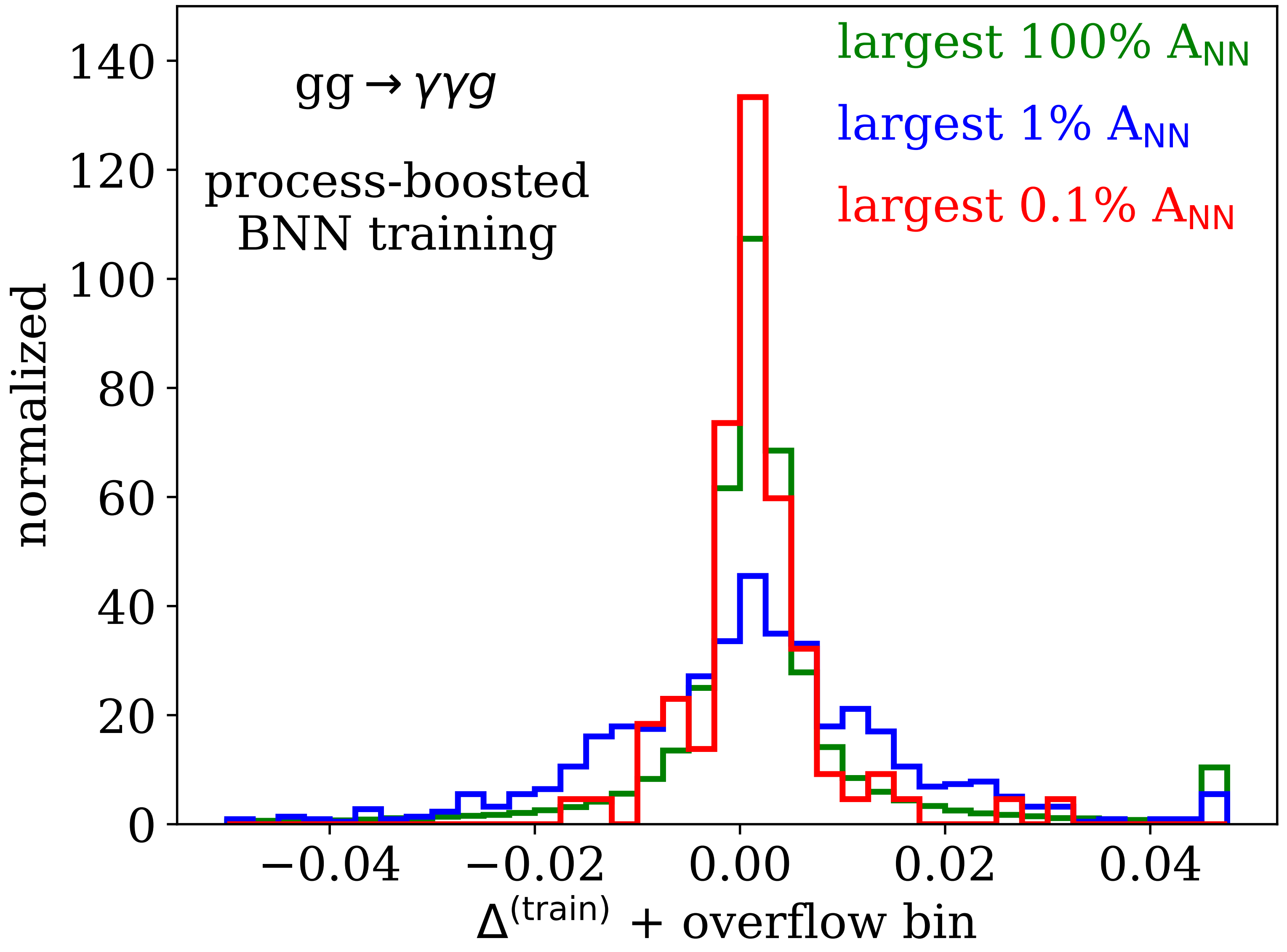
Improvement for test data

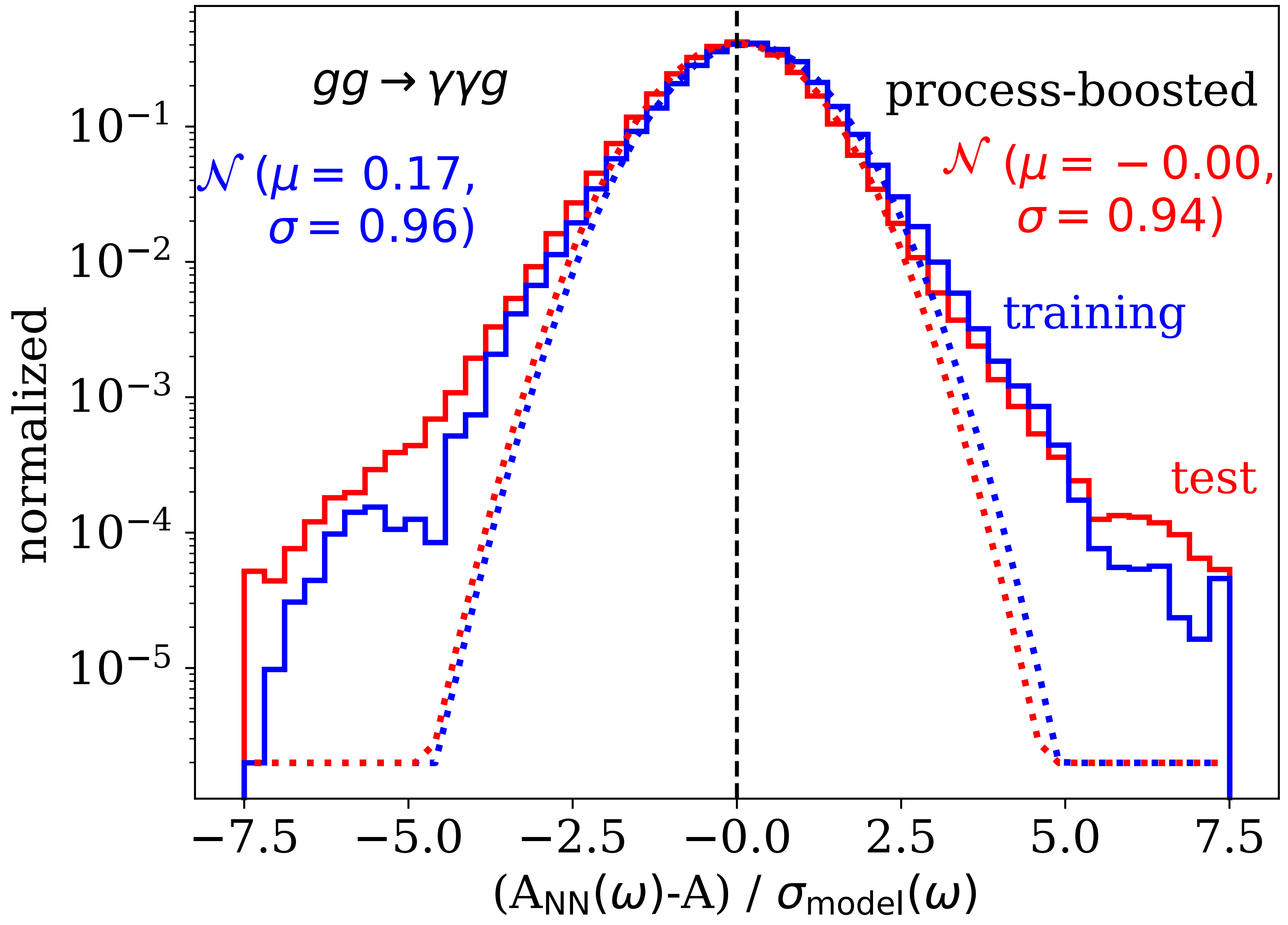
Performance boosting

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

→ include them +3 times in each epoch

→ Repeat 20 times

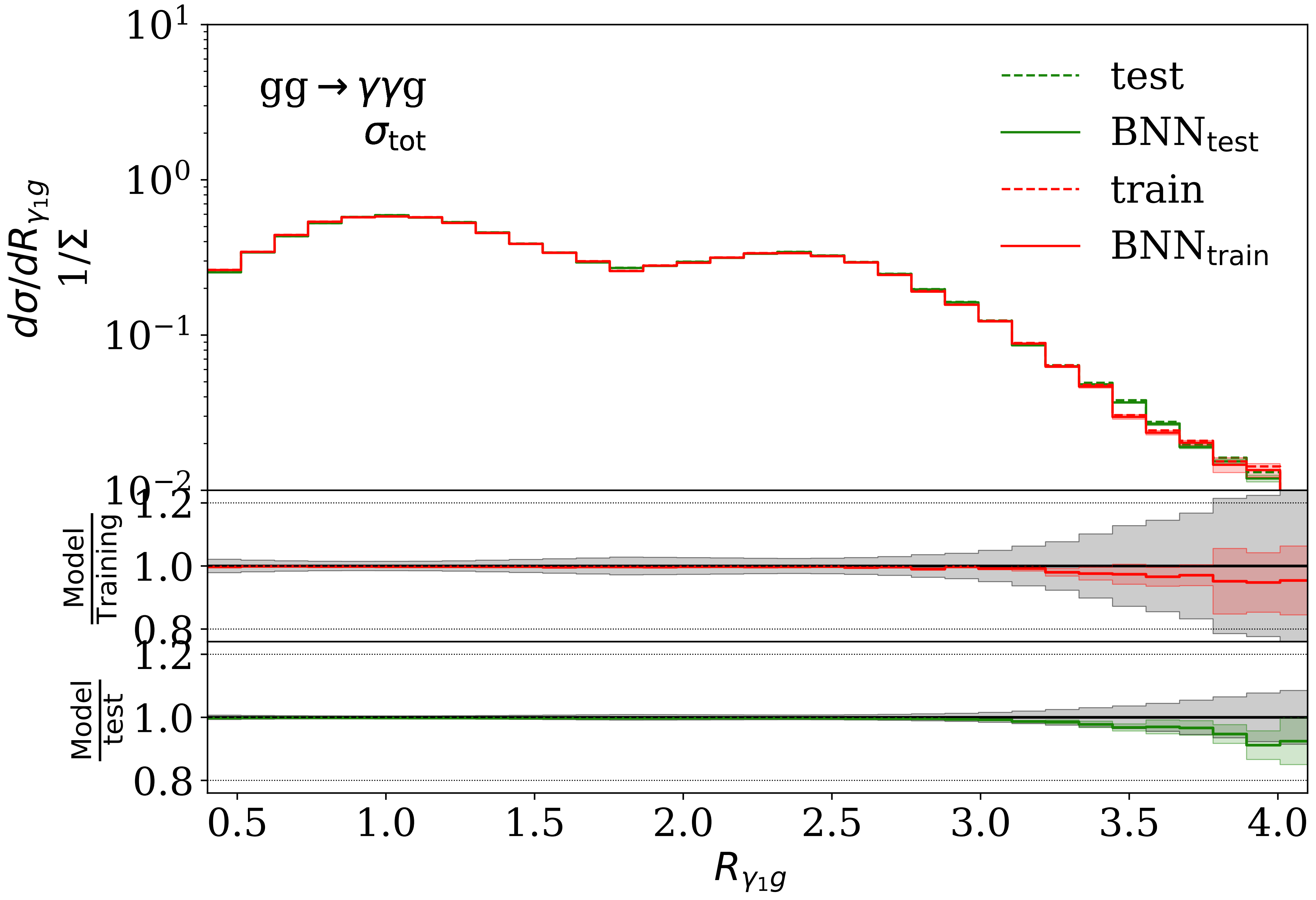




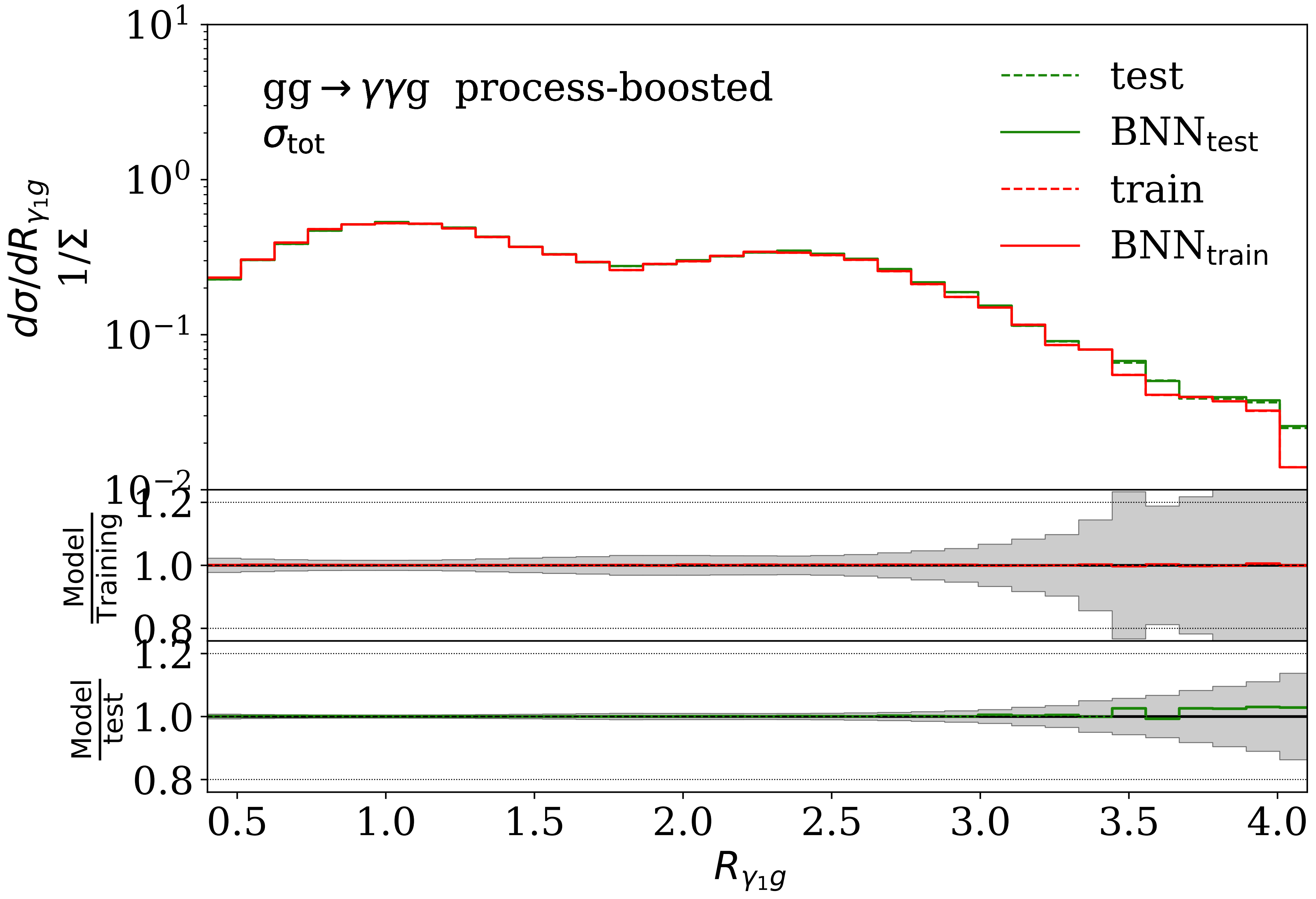
Significant improvement in performance

Kinematic distributions

Standard BNN



Precision boosted BNN



Gray shades indicate statistical limitation of training data…

Precision networks for loop amplitudes

Fast evaluation of NN can **accelerate** event generation

Standard NN give little control over prediction

Bayesian networks provide **uncertainty estimates**

Boost network performance for precision or calibration

Precision better than 1%
Method applicable to all regression problems