Calibrating calorimeter signals using an uncertainty-aware neural network

Isabel Sainz

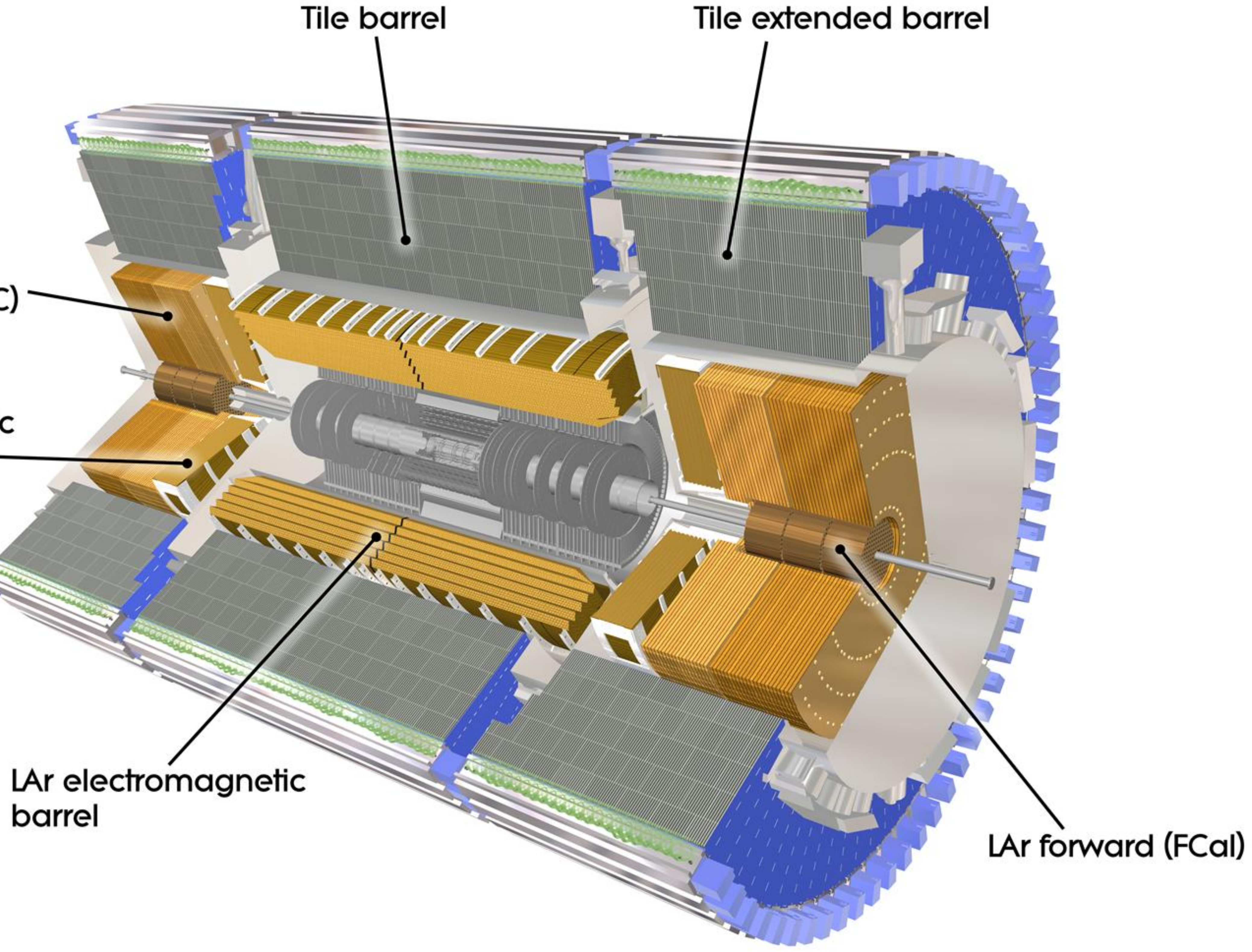
Herbstschule HEP 25 - Bad Honnef

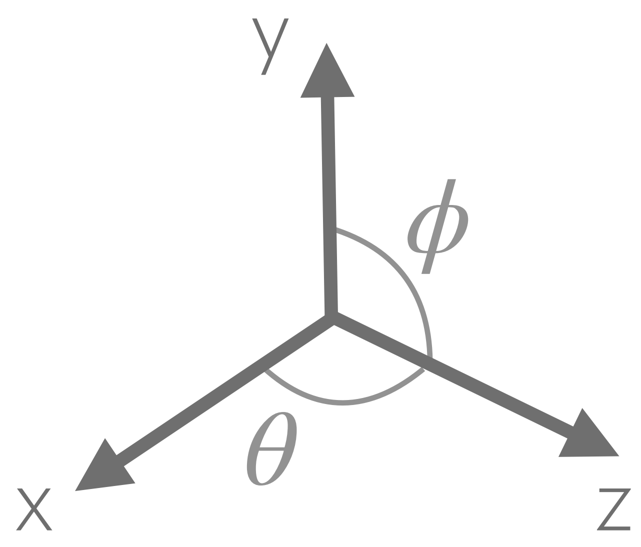
02.09.25

ATLAS EXPERIMENT
FSP ATLAS Erforschung von Universum und Materie
IMPRS for Precision Tests of Fundamental Symmetries
INTERNATIONAL MAX PLANCK RESEARCH SCHOOL
UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386
KIRCHHOFF-INSTITUT FÜR PHYSIK

Calorimeter signals in ATLAS

Reconstruction of energy deposited by the particles in the calorimeter cells

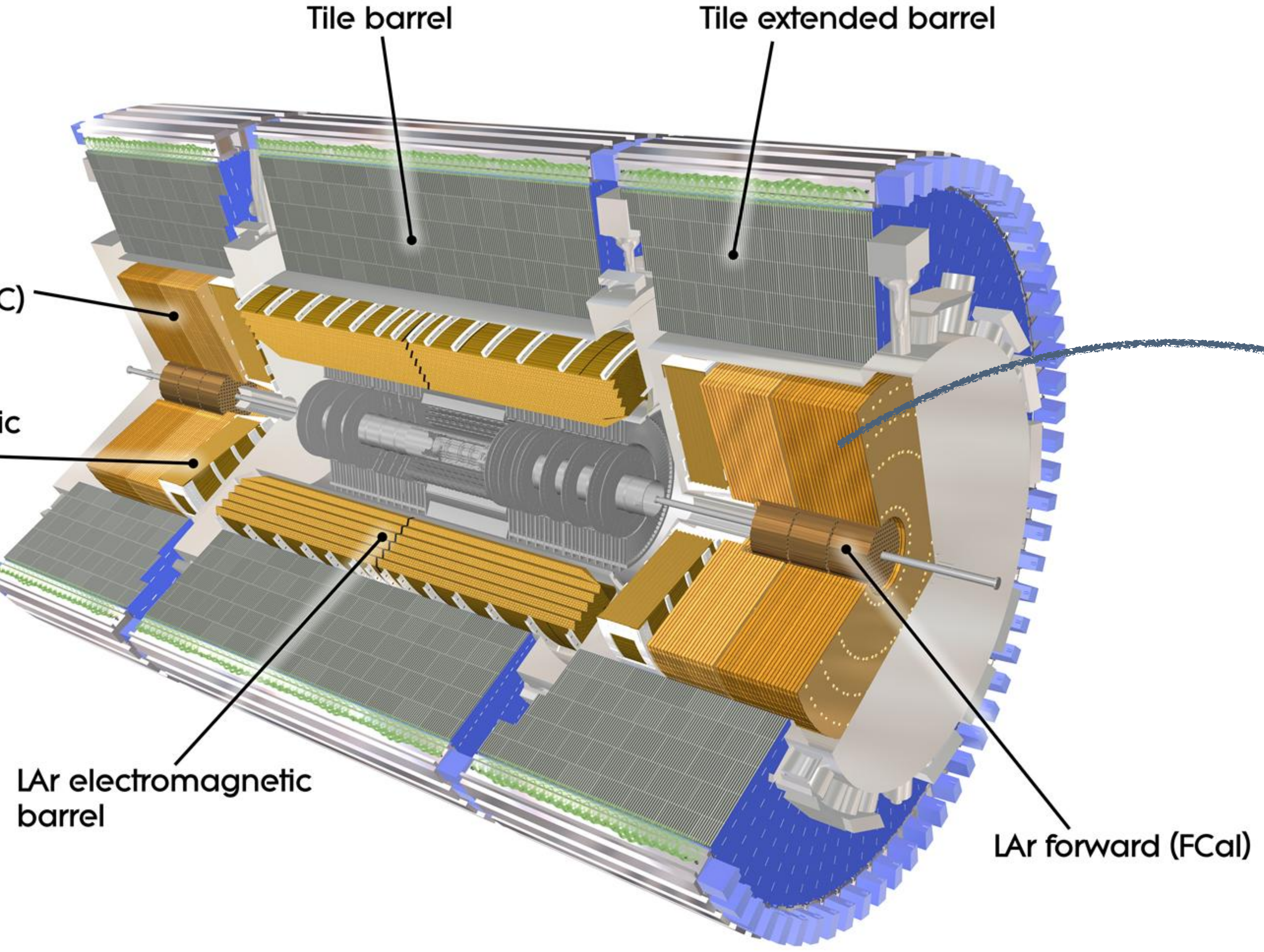


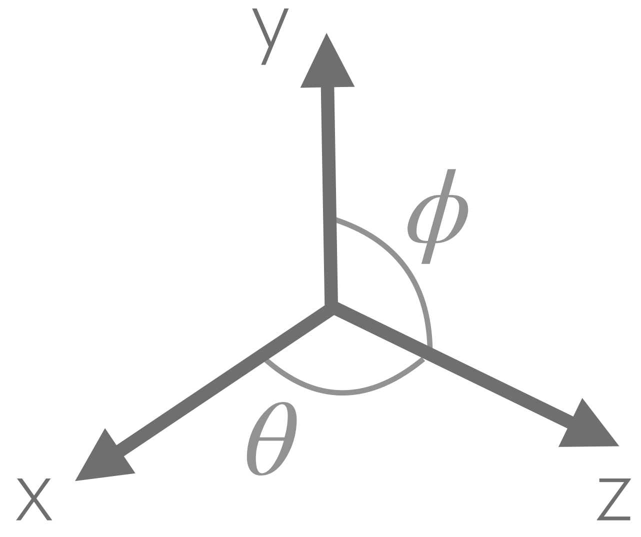


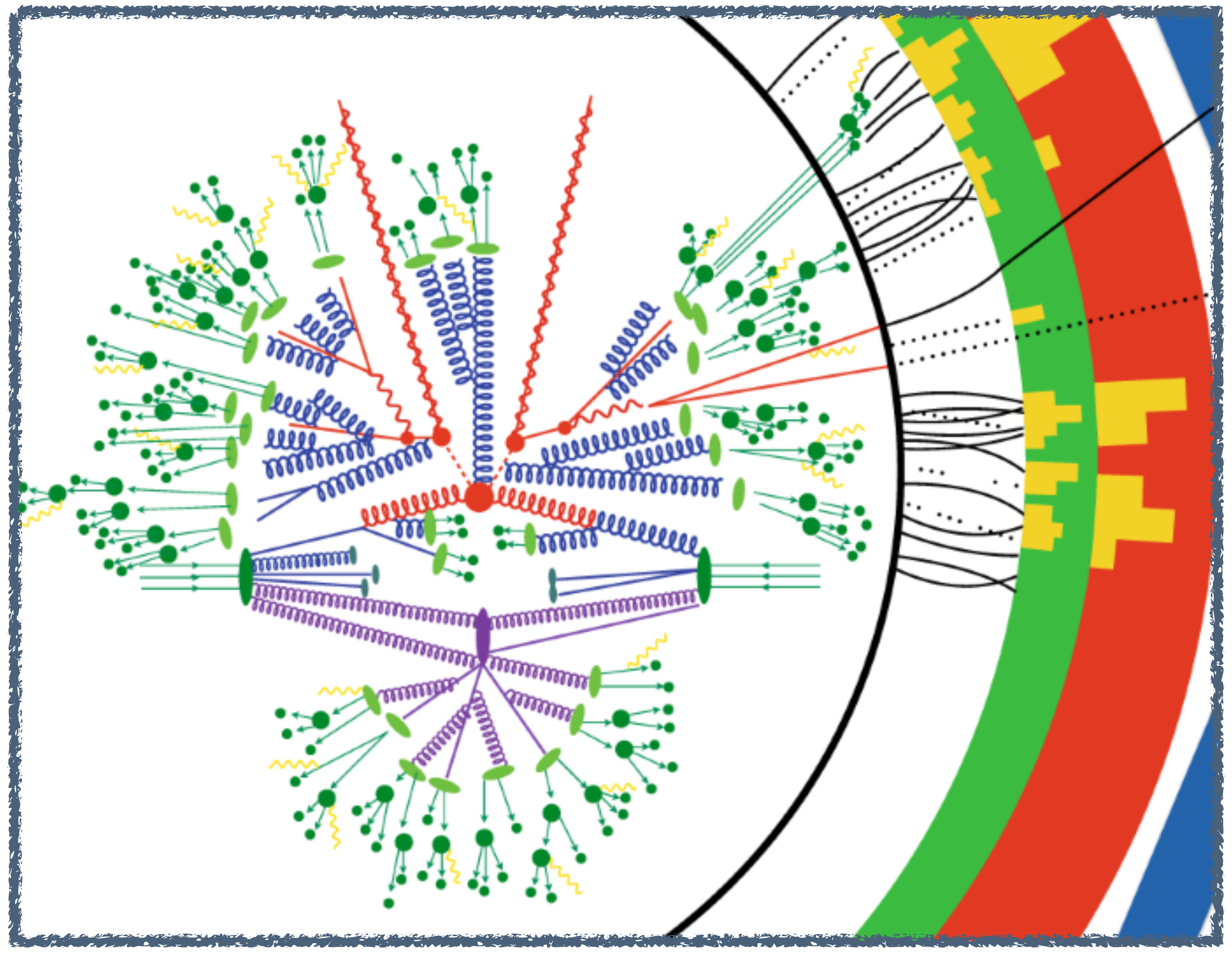
CERN-GE-0803015-01

Calorimeter signals in ATLAS

Reconstruction of energy deposited by the particles in the calorimeter cells







CERN-GE-0803015-01

Topo-cluster Calibration

Topo-clusters: reconstructed clusters of topologically connected cell signals

- Correctly measure EM energy: E_{clus}^{EM} → But no compensation for invisible losses
- **Calibration** to account for hadronic component: E_{clus}^{had}

Topo-cluster Calibration

Topo-clusters: reconstructed clusters of topologically connected cell signals

- Correctly measure EM energy: E_{clus}^{EM} → But no compensation for invisible losses
- **Calibration** to account for hadronic component: E_{clus}^{had}

→ Since Run 1: Local Cell Weighting (LCW)

- Returns scale factors in multi-dimensional tables
- Does not correlate the topo-cluster features

Topo-cluster Calibration

Topo-clusters: reconstructed clusters of topologically connected cell signals

- Correctly measure EM energy: E_{clus}^{EM} → But no compensation for invisible losses
- **Calibration** to account for hadronic component: E_{clus}^{had}
  - → Since Run 1: Local Cell Weighting (LCW)
  - → Alternative: Machine Learning-based Calibration
    - Returns multi-dimensional calibration function
    - Exploits correlations between features

NN-based topo-cluster calibration

- Goal: $E_{clus}^{EM} \xrightarrow{\text{Calibration}} E_{clus}^{had} = E_{clus}^{dep}$

- Regression Network: $R_{clus}^{model}(\chi_{clus}) \approx R_{clus}^{EM} = \frac{E_{clus}^{EM}}{E_{clus}^{dep}}$
  - → Well reconstructed
  - → Truth information

- (Training) Data: $\{R_{clus}^{EM}, \chi_{clus}\}_j$ with $j = 1,\ldots,N_{train}$ → Topo-clusters (di-jet samples)

$$\chi_{clus} = \{\underbrace{E_{clus}^{EM}, y_{clus}^{EM}, \zeta_{clus}^{EM}, t_{clus}, \mathrm{Var}(t_{cell}), \lambda_{clus}, |\vec{c}_{clus}|, f_{emc}, |, \langle\rho_{cell}\rangle, \langle m_{long}^2\rangle, \langle m_{lat}^2\rangle, p_T D}_{\text{Topo-cluster properties}}, \overbrace{f_{iso}, N_{PV}, \mu}^{\text{Event properties}}\}$$

Event properties

Topo-cluster properties

DNN-HGM

- Deterministic network:
  weights are fixed trainable values

- Assume **Gaussian Mixture** (GM) likelihood

- Minimize **heteroscedastic** loss (H):
  introduces uncertainty σ_{clus}^{model} ! (see backup slide)

DNN-HGM

- Deterministic network:
  weights are fixed trainable values

- Assume **Gaussian Mixture** (GM) likelihood

- Minimize **heteroscedastic** loss (H):
  introduces uncertainty σ_{clus}^{model} ! (see backup slide)

- Idea from T. Plehn and L. Vogel (ITP Heidelberg) and based on BNN model [arXiv:2412.04370]

- DNN more straightforward inference

DNN-HGM

- Deterministic network:
  weights are fixed trainable values

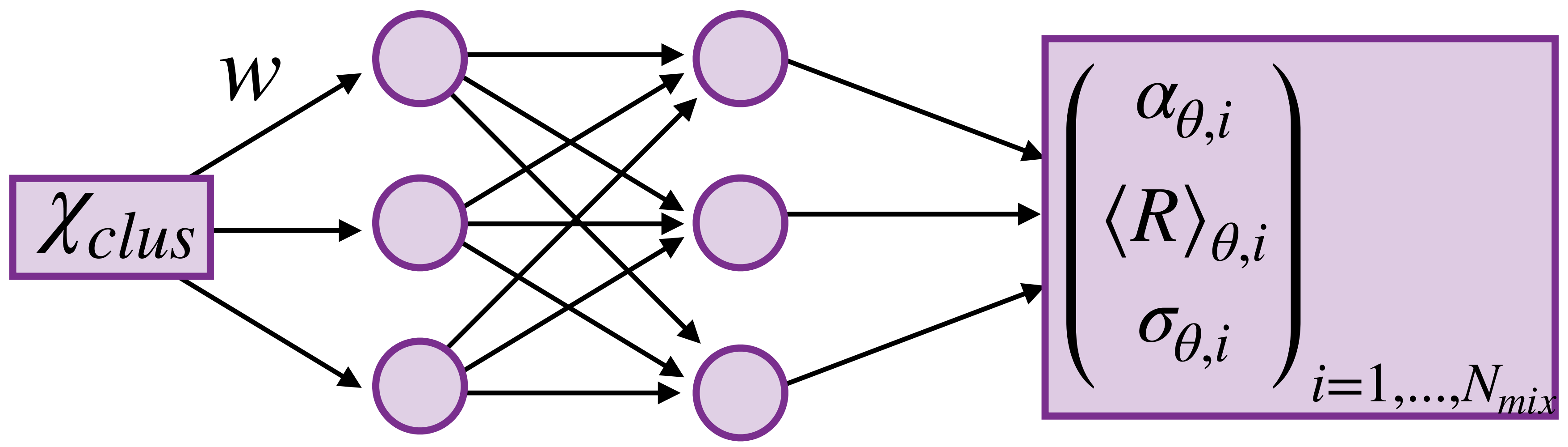
- Assume **Gaussian Mixture** (GM) likelihood

- Minimize **heteroscedastic** loss (H):
  introduces uncertainty σ_{clus}^{model} ! (see backup slide)

- Idea from T. Plehn and L. Vogel (ITP Heidelberg) and based on BNN model [arXiv:2412.04370]

- DNN more straightforward inference

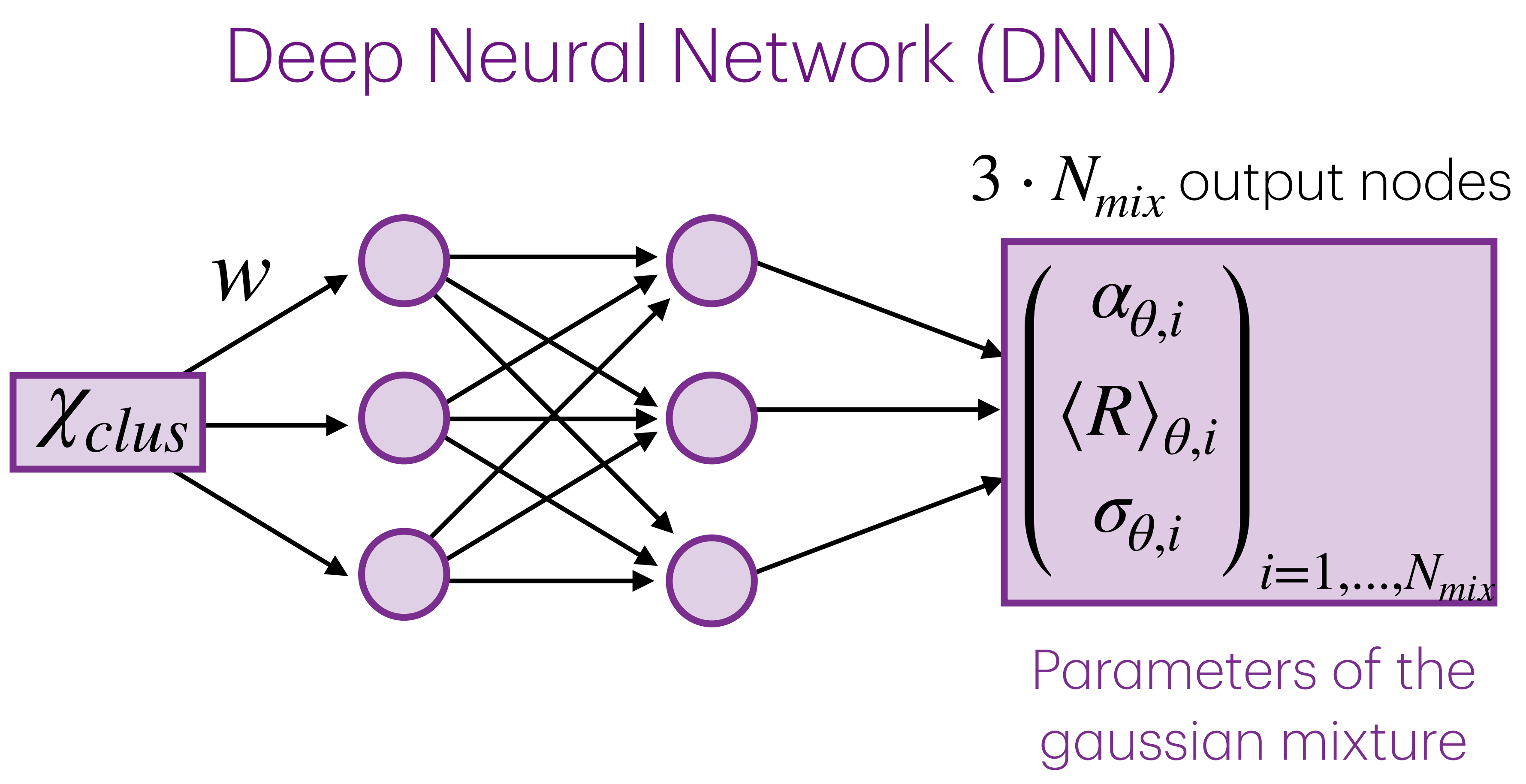
Deep Neural Network (DNN)



Parameters of the gaussian mixture

DNN-HGM

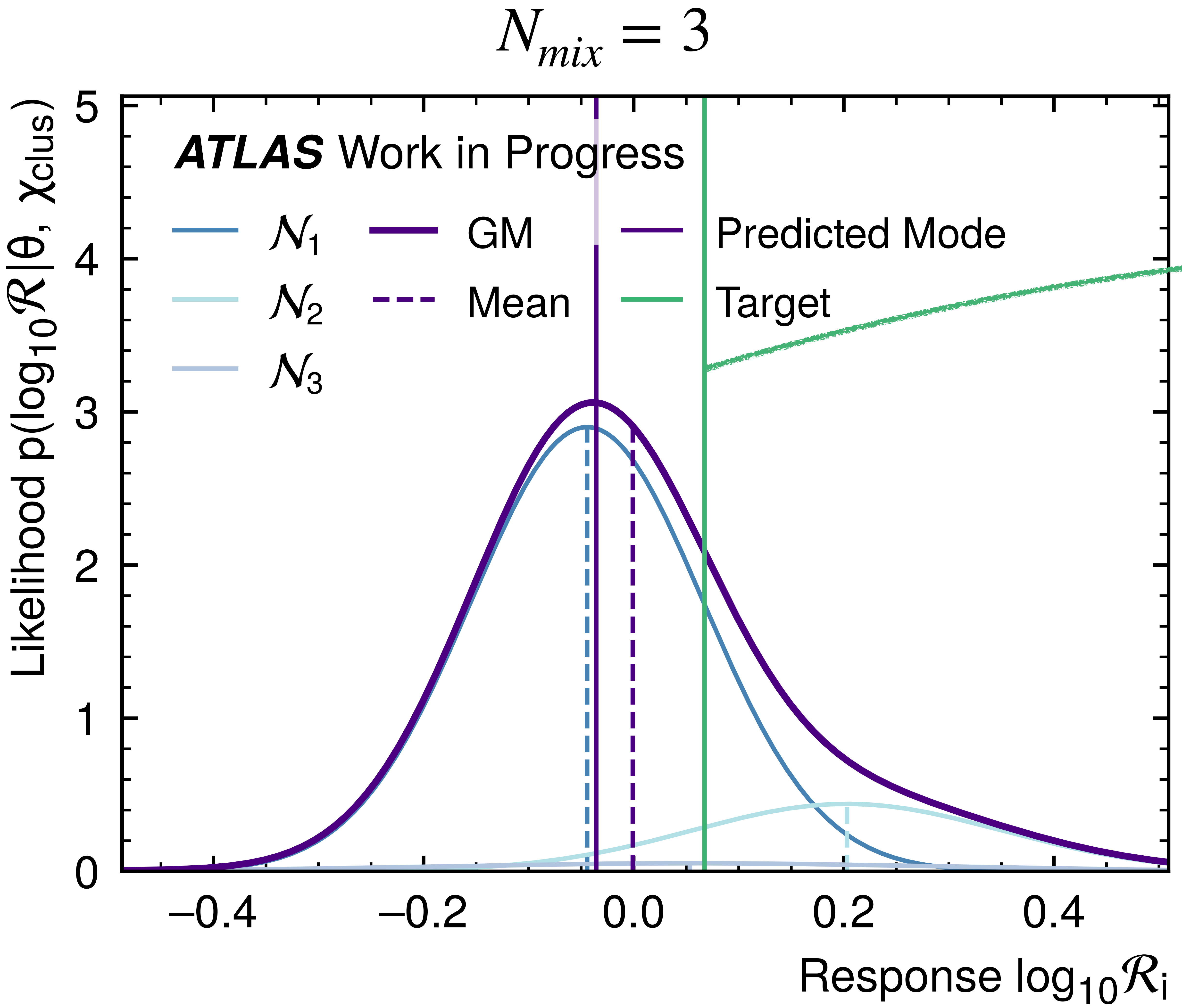
- Deterministic network:
  weights are fixed trainable values
- Assume **Gaussian Mixture** (GM) likelihood
- Minimize **heteroscedastic** loss (H):
  introduces uncertainty σ_{clus}^{model} ! (see backup slide)
- Idea from T. Plehn and L. Vogel (ITP Heidelberg) and based on BNN model [arXiv:2412.04370]
- DNN more straightforward inference



$$p\left(\{R_{clus}^{EM}\}\,|\,\theta, \{\chi_{clus}\}\right) = \sum_{i=1}^{N_{mix}} \alpha_{\theta,i}\,\mathcal{N}\left(R_{clus}^{EM}\,|\,R_{\theta_i}, \sigma_{\theta_i}\right)$$

Prediction of a PDF per topo-cluster!

Gaussian Mixture (GM) Model



Prediction of a PDF per topo-cluster!

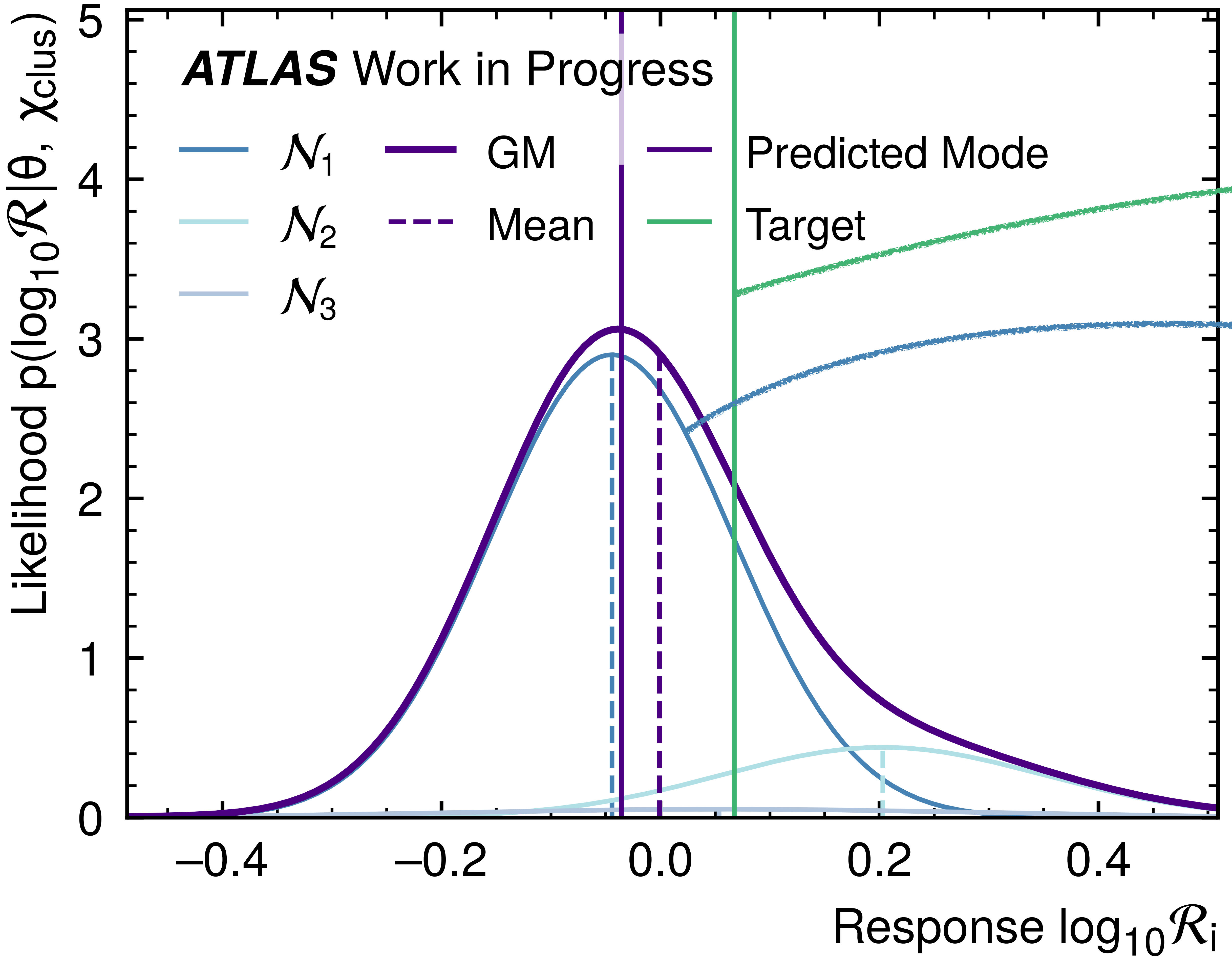
Target: $\log_{10} R^{EM}_{clus}$

Note: We actually train on $\log_{10} R$!!

Gaussian Mixture (GM) Model

$N_{mix} = 3$

Prediction of a PDF per topo-cluster!



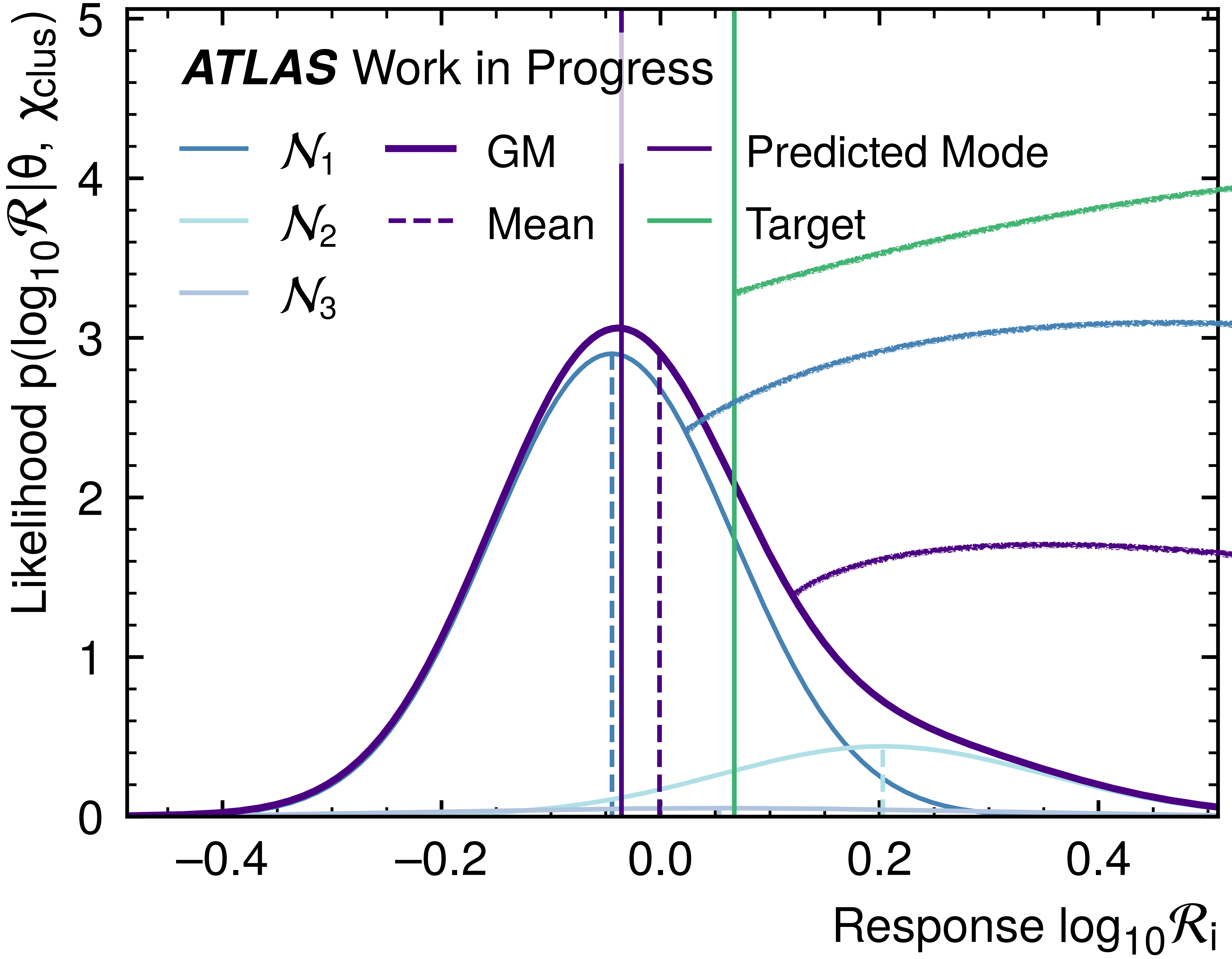
Target: $\log_{10} R^{EM}_{clus}$

$\mathcal{N}\left(\log_{10} R^{EM}_{clus} \,|\, \log_{10} R_1, \sigma_1\right)$

Note: We actually train on $\log_{10} R$!!

Gaussian Mixture (GM) Model

$N_{mix} = 3$



Prediction of a PDF per topo-cluster!

Target: $\log_{10} R^{EM}_{clus}$

$\mathcal{N}\left(\log_{10} R^{EM}_{clus} \,|\, \log_{10} R_1, \sigma_1\right)$

Sum of 3 Gaussians as likelihood
Prediction = Mode

Note: We actually train on $\log_{10} R$!!

Signal linearity

Reminder:

$$E_{clus}^{EM} \xrightarrow{\text{Calibration}} E_{clus}^{had} = E_{clus}^{dep}$$

$$\rightarrow E_{clus}^{model}$$

Signal linearity

Reminder:

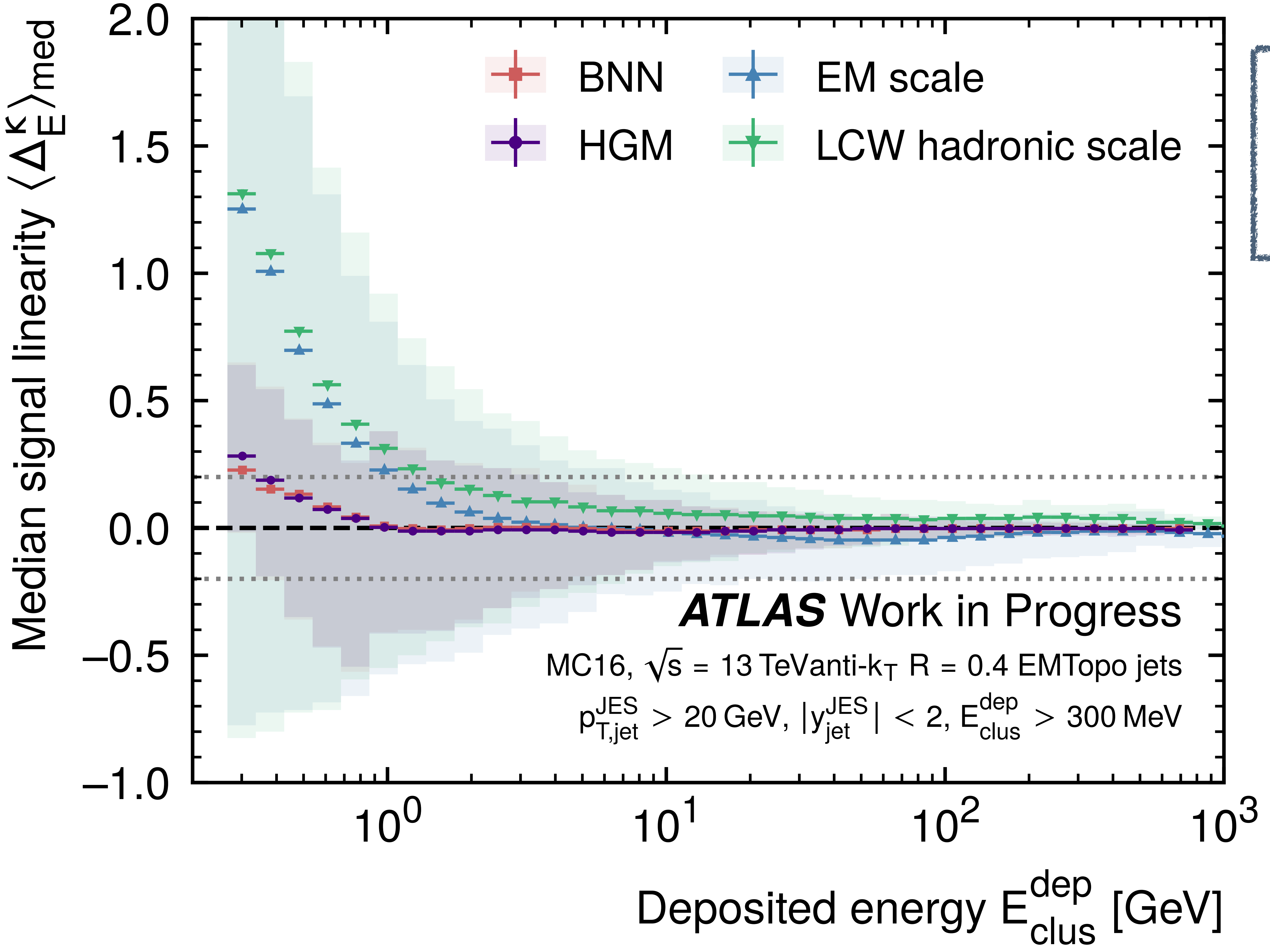
$$E_{clus}^{EM} \xrightarrow{\text{Calibration}} E_{clus}^{had} = E_{clus}^{dep}$$

$$\rightarrow E_{clus}^{model}$$

$$\Delta_E^{model} = \frac{E_{clus}^{model}}{E_{clus}^{dep}} - 1 \quad \text{with} \quad E_{clus}^{model} = \frac{E_{clus}^{EM}}{R_{clus}^{model}}$$

Signal linearity

Run 2

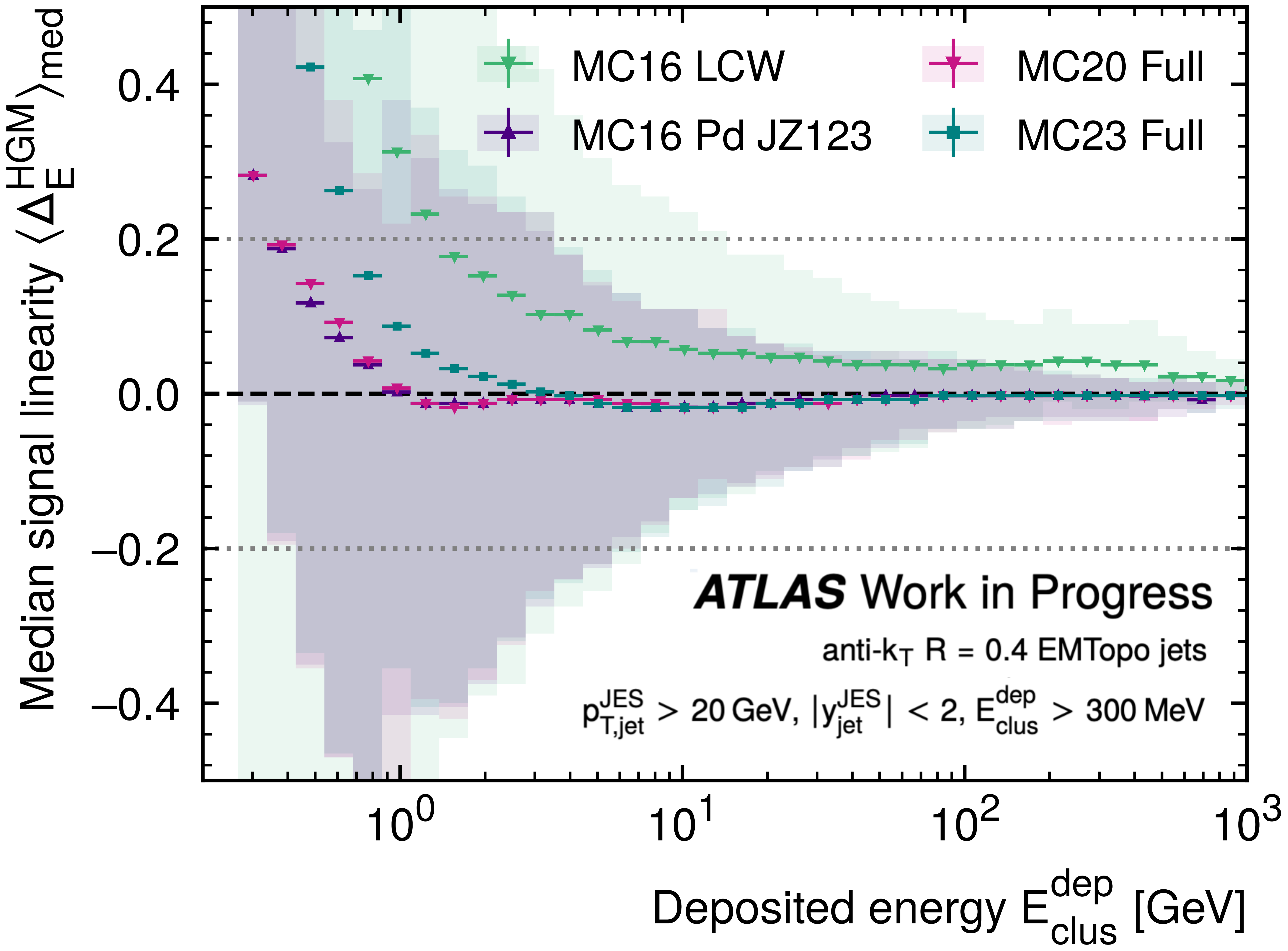


$$\Delta_E^{model} = \frac{E_{clus}^{model}}{E_{clus}^{dep}} - 1 \text{ with } E_{clus}^{model} = \frac{E_{clus}^{EM}}{R_{clus}^{model}}$$

- Better than EM scale and current hadronic calibration
- Performance similar to BNN in all features

Signal linearity

Run 2 & Run 3



$$\Delta_E^{model} = \frac{E_{clus}^{model}}{E_{clus}^{dep}} - 1 \quad \text{with} \quad E_{clus}^{model} = \frac{E_{clus}^{EM}}{R_{clus}^{model}}$$

- Better than current hadronic calibration (LCW)
- Similar performance between MC16&20 - Run 2
- Room for optimizing MC23 - Run 3

Pull distributions

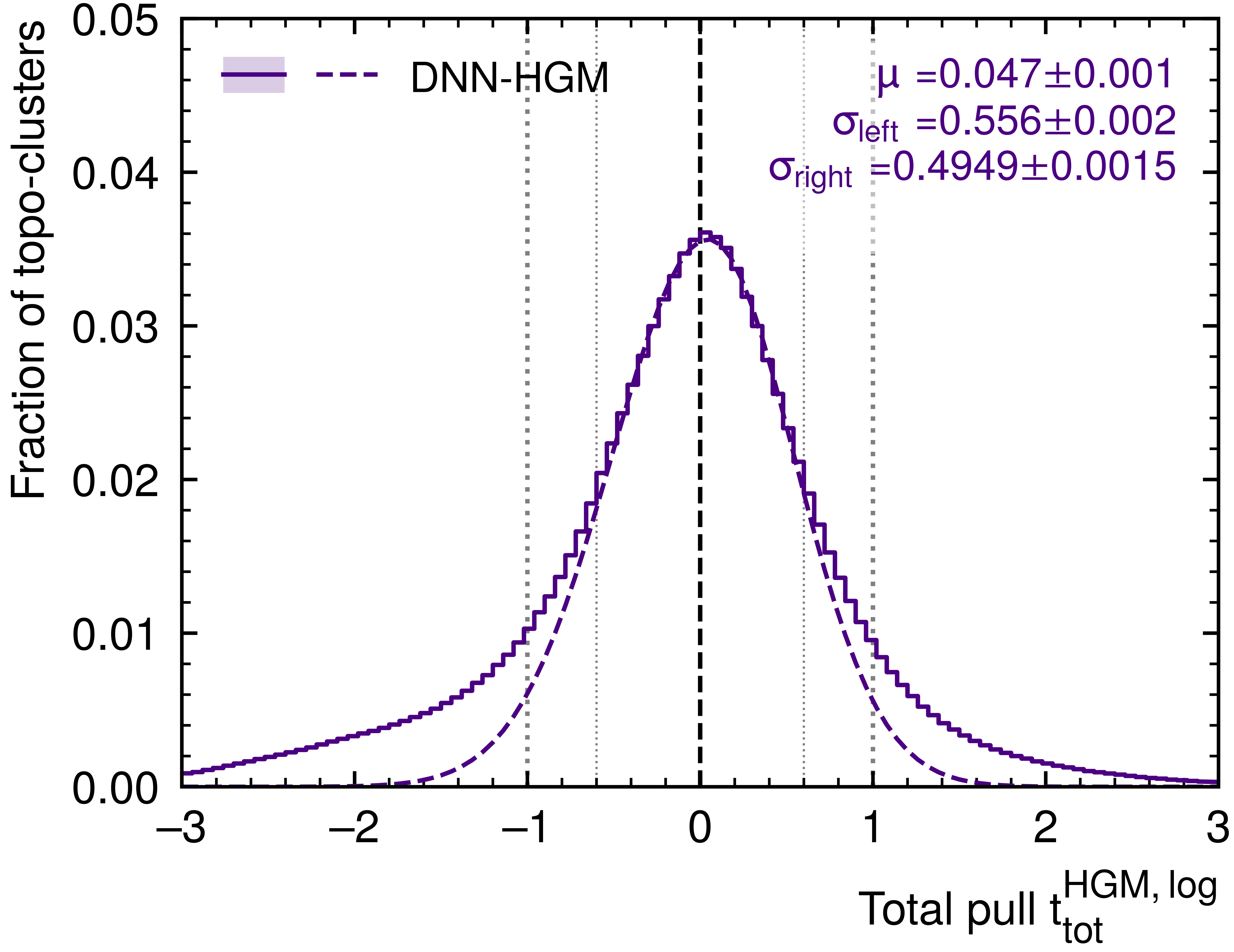
Run 2

$$t(x) = \frac{R^{model}_{clus}(x) - R^{EM}_{clus}}{\sigma^{model}_{tot}(x)}$$

- Uncertainty includes stochastic effects

  $\Rightarrow$ Expect $\mathcal{N}(0,1)$

- Predictions:
  - Unc. overestimation ($\sigma_{pull} < 1$)



Summary & outlook

- NN-based alternative to hadronic calibration in ATLAS
- Better performance than the standard calibration
- Still room for improvement in Run 3
- Conservative uncertainty prediction
- Next steps: jet level studies!

Summary & outlook

- NN-based alternative to hadronic calibration in ATLAS
- Better performance than the standard calibration
- Still room for improvement in Run 3
- Conservative uncertainty prediction
- Next steps: jet level studies!

Thanks for listening! Any questions?

Backup

Previous models

- **BNN Model**

ATL-COM-PHYS-2024-747

  - Probabilistic network:

    weights are trainable Gaussian distributions, $q(\theta)$

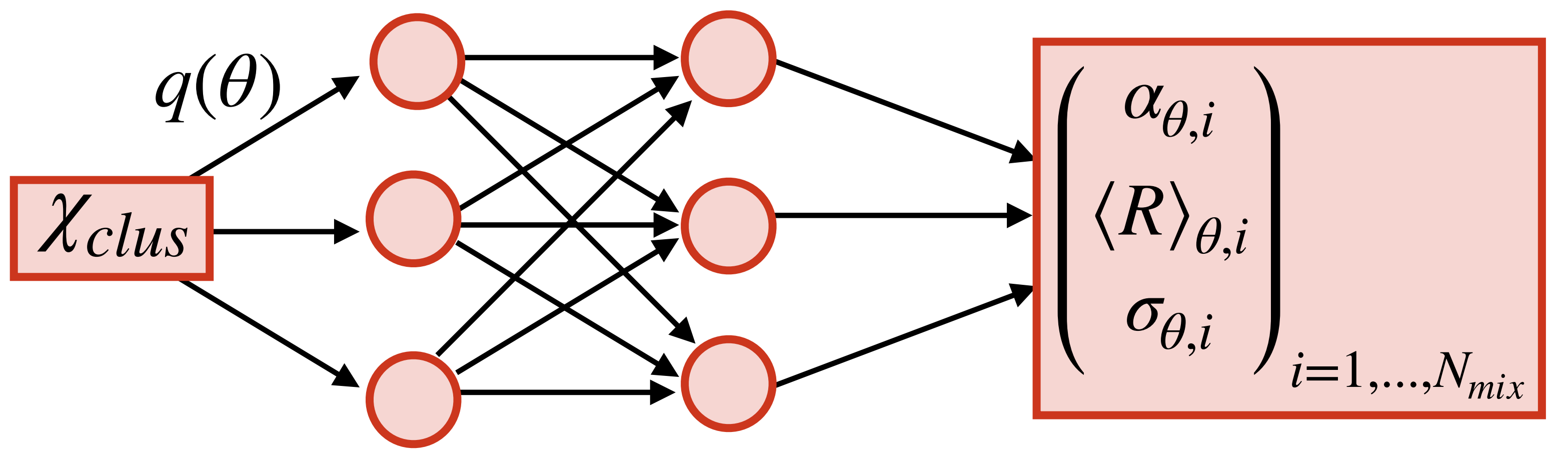
  - Assumes Gaussian Mixture likelihood

    → includes σ_{syst}

  - Inference by sampling $q(\theta)$

    → Includes σ_{stat}

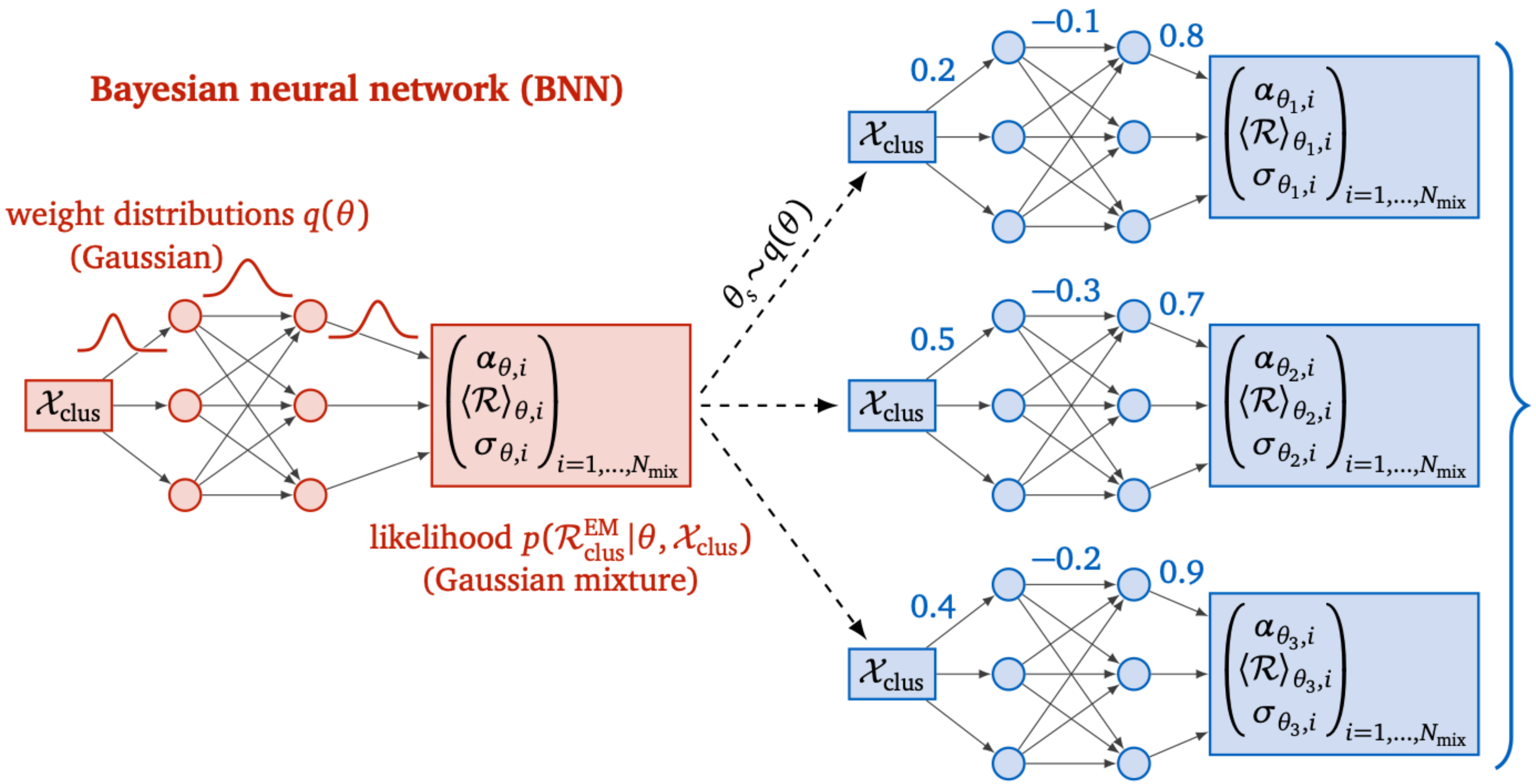
Bayesian Neural Network (BNN)



Likelihood $p(R^{EM}_{clus} | \theta, \chi_{clus})$
Gaussian Mixture

No straightforward implementation in the ATLAS Framework

BNN Architecture



Central-value prediction (maximum likelihood) and uncertainties for a Gaussian mixture model:

$$\mathcal{R}^{\text{BNN}}_{\text{clus}}(\mathcal{X}_{\text{clus}}) = \arg\max_{\mathcal{R}} \frac{1}{N}\sum_{s=1}^{N} p(\mathcal{R}|\theta_s,\mathcal{X}_{\text{clus}})$$

$$\sigma^2_{\text{syst}}(\mathcal{X}_{\text{clus}}) = \frac{1}{N}\sum_{s=1}^{N}\left[\sum_{i=1}^{N_{\text{mix}}}\alpha_{\theta_s,i}\left(\sigma^2_{\theta_s,i}+\langle\mathcal{R}\rangle^2_{\theta_s,i}\right)-\langle\mathcal{R}\rangle^2_{\theta_s}\right]$$

$$\sigma^2_{\text{stat}}(\mathcal{X}_{\text{clus}}) = \text{Var}\left(\langle\mathcal{R}\rangle_{\theta_s}\right) = \frac{1}{N}\sum_{s=1}^{N}\left[\langle\mathcal{R}\rangle-\langle\mathcal{R}\rangle_{\theta_s}\right]^2$$

with

$$\langle\mathcal{R}\rangle_{\theta_s} = \sum_{i=1}^{N_{\text{mix}}}\alpha_{\theta_s,i}\langle\mathcal{R}\rangle_{\theta_s,i} \quad \text{and} \quad \langle\mathcal{R}\rangle = \frac{1}{N}\sum_{s=1}^{N}\langle\mathcal{R}\rangle_{\theta_s}$$

Training: Weights linking the nodes of adjacent layers are described by weight distributions $q(\theta)$

Inference: Learned weight distributions $q(\theta)$ are sampled N times to generate a set of network parameters θ_s and thus an ensemble of networks

Taken from ATL-COM-PHYS-2024-747

DNN Architecture

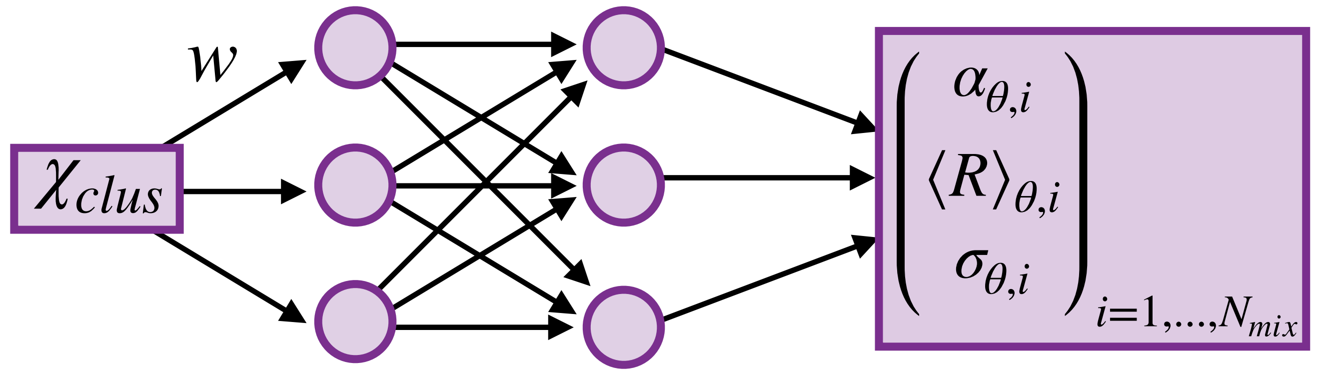
Central-value prediction (maximum likelihood) and uncertainties for a GMM:

$$R_{clus}^{DNN}(\chi_{clus}) = \underset{R}{\text{arg max}}\, p(R_{clus}^{EM} | \theta, \chi_{clus})$$

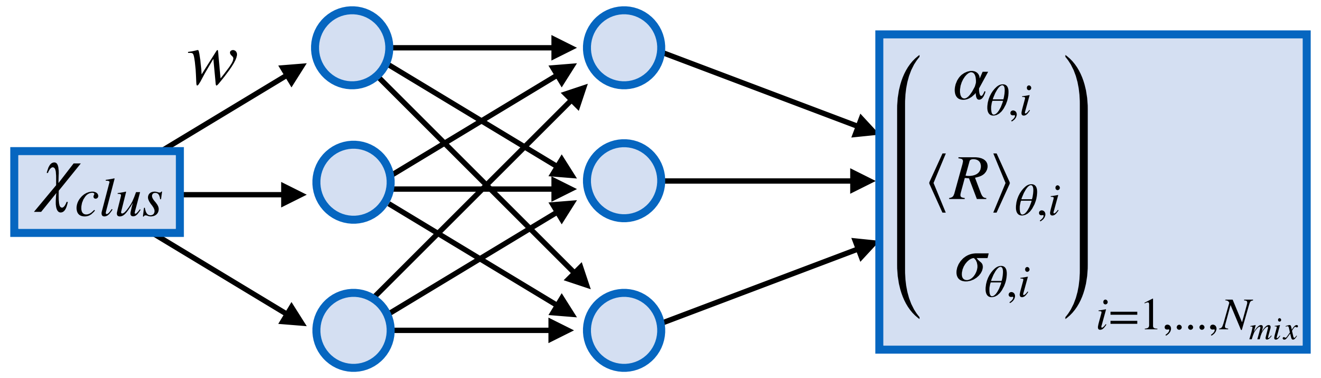
$$\sigma_{syst}^2(\chi_{clus}) = \sum_{i=1}^{N_{mix}} \alpha_{\theta,i} \left(\sigma_{\theta,i}^2 + \langle R \rangle_{\theta,i}^2\right) - \left(\sum_{i=1}^{N_{mix}} \alpha_{\theta,i} \langle R \rangle_{\theta}\right)^2$$

$$\sigma_{stat} = 0$$

Deep Neural Network



Likelihood $p(R_{clus}^{EM} | \theta, \chi_{clus})$
Gaussian Mixture



Training: weights linking the nodes of adjacent layers are fixed values w

Inference: Straightforward, no weight sampling

Network training

- Maximise probability that the set of θ describes training data D_{train}: $p(\theta | D_{train})$

↕

Bayes' Theorem

- Minimise loss function: $L = -\log p(\theta | D_{train}) = -\log p(D_{train} | \theta) - \log p(\theta)$

Prior (regularisation)

Likelihood of choice

Gaussian Likelihood

$(R_\theta = R_{clus}^{model})$

- Assume single Gaussian:

$$p\left(\{R_{clus}^{EM}\}\,|\,\theta, \{\chi_{clus}\}\right) = \prod_{j=1}^{N_{train}} \frac{1}{\sqrt{2\pi}\sigma_{\theta,j}} \exp\left(-\frac{|R_{clus}^{EM} - R_\theta(x_j)|^2}{2\sigma_j^2}\right)$$

→ 2 output nodes

  - If uncertainties are invariant wrt inputs:

$$L = \frac{1}{2\sigma^2}\sum_{j=1}^{N_{train}} |R_{clus}^{EM} - R_\theta(x_j)|^2 + \ldots \equiv \frac{1}{2\sigma^2} MSE + \ldots$$

Homoscedastic Loss

  - If they depend on inputs:

$$L = \sum_{j=1}^{N_{train}} \left(\frac{|R_{clus}^{EM} - R_\theta(x_j)|^2}{2\sigma_\theta(x_j)^2} + \log\sigma_\theta(x_j)\right) + \ldots$$

Heteroscedastic Loss

DNN-HGM

- Assume **gaussian mixture**

$$p\left(\{R^{EM}_{clus}\}\,|\,\theta,\{\chi_{clus}\}\right) = \sum_{i=1}^{N_{mix}} \alpha_{\theta,i}\,\mathcal{N}\left(R^{EM}_{clus}\,|\,R_{\theta_i},\sigma_{\theta_i}\right)$$

- **Heteroscedastic loss** function: $\sigma_{\theta,i}$ depend on inputs

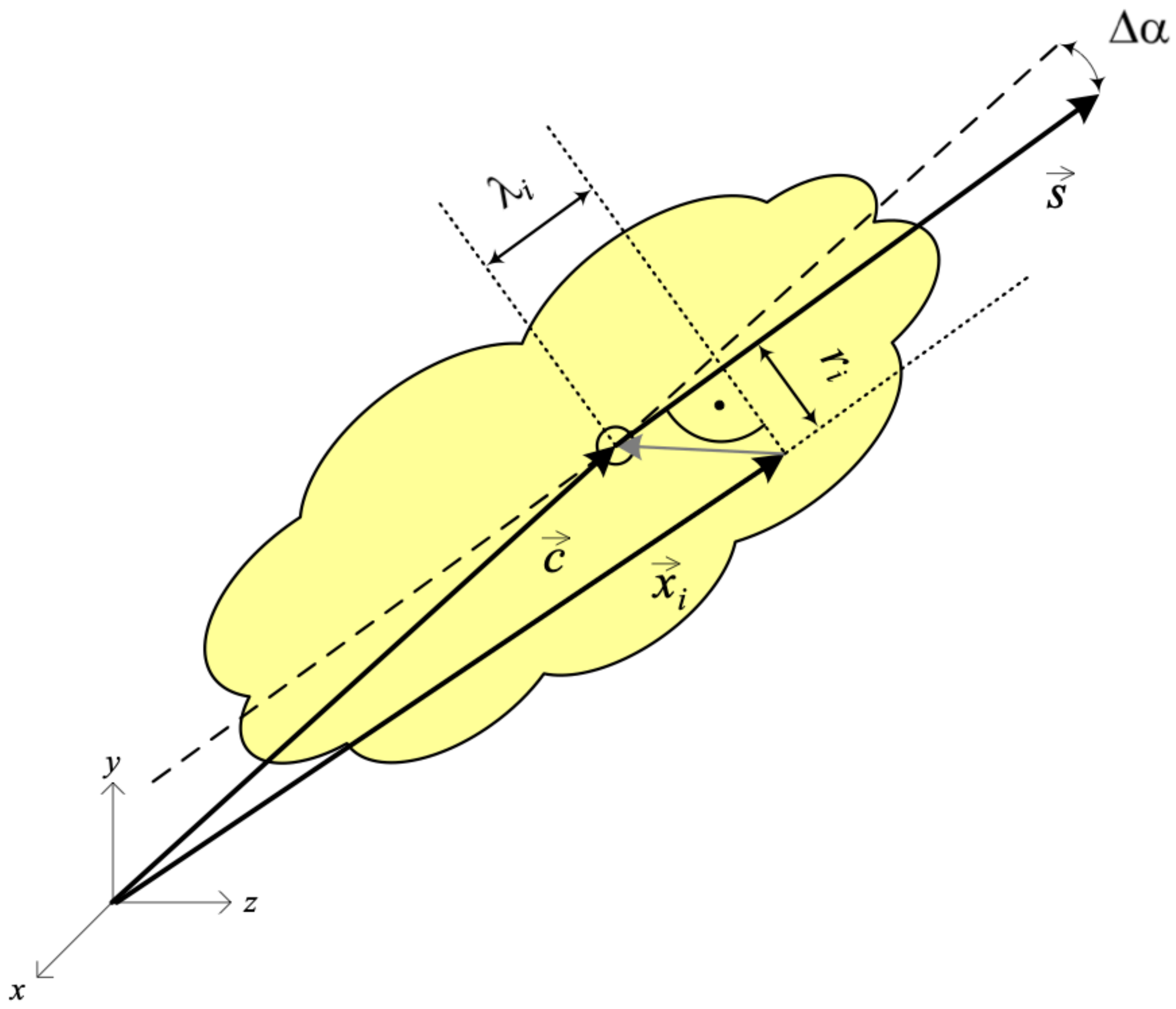
$$L = -\log\left[\sum_{i=1}^{N_{mix}} \alpha_{\theta,i}\frac{1}{\sqrt{2\pi}\sigma_{\theta,i}(x_j)}\exp\left(\frac{|R^{EM}_{clus} - R_{\theta,i}(x_j)|^2}{2\sigma_{\theta,i}(x_j)^2}\right)\right] + \ldots \quad \text{with } j = 1,\ldots,N_{train}$$

→ Introduces uncertainty σ^{model} !

DNN-GMHet hyperparameters

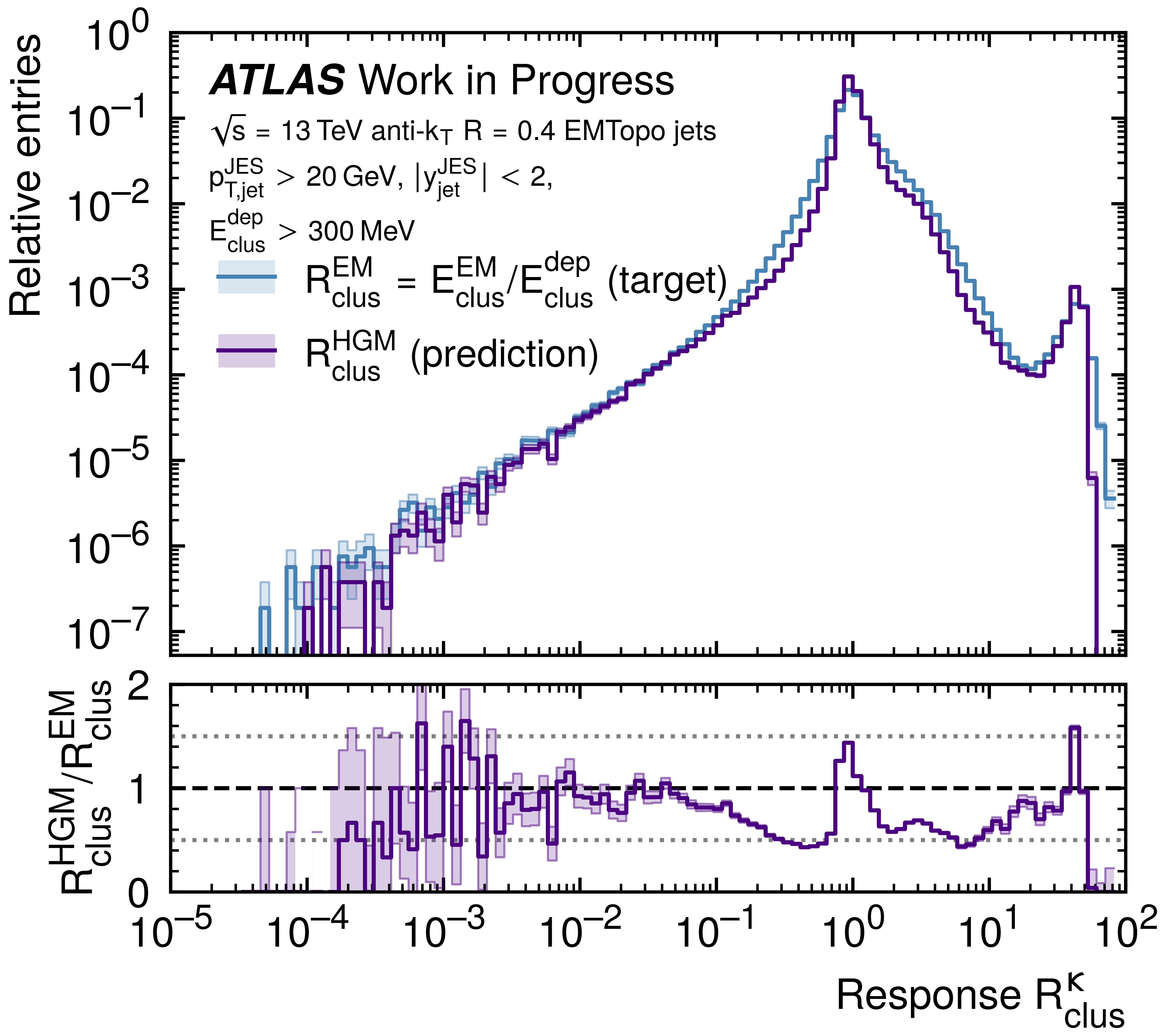
Hyperparameter	DNN-GMHet
Likelihood Loss	Gaussian Mixture (GM)
Number of modes (N_{mix})	3
Number of layers and nodes per layer	{64, 64, 64, 64}
Activation functions	ReLU (inner layers) and none (last later)
Prediction	Maximum-likelihood value
Optimiser	ADAM
Learning rate (LR)	10^{-4}
Learning-rate scheduler	STEPLR, epochs {25, 100}, γ=0.1
Number of training epochs	150
Batch size for training, testing	4096, 512
Dataset sizes for training/validation/testing	8.7M, 500k, 5.3M

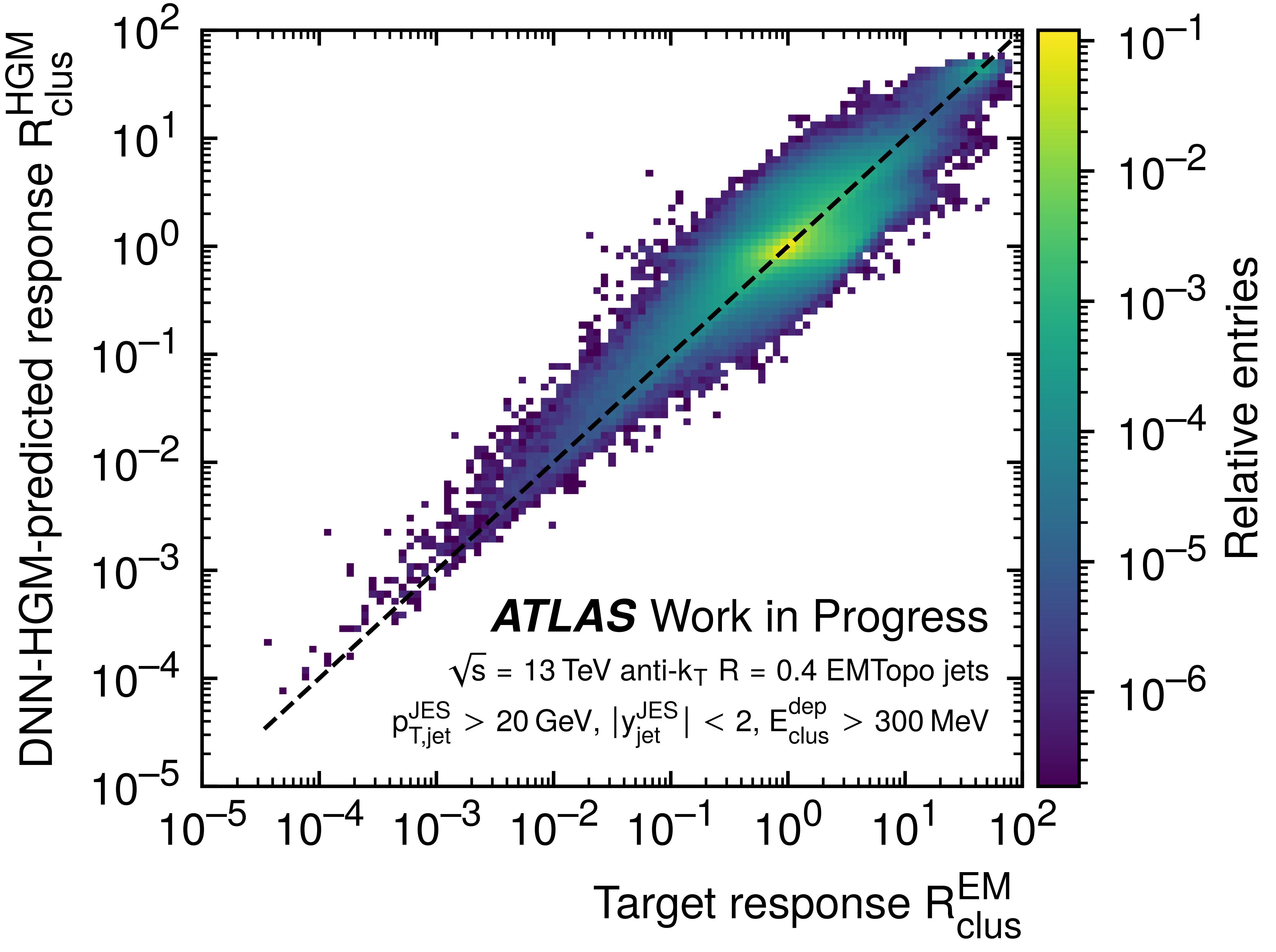
Topo-cluster moments



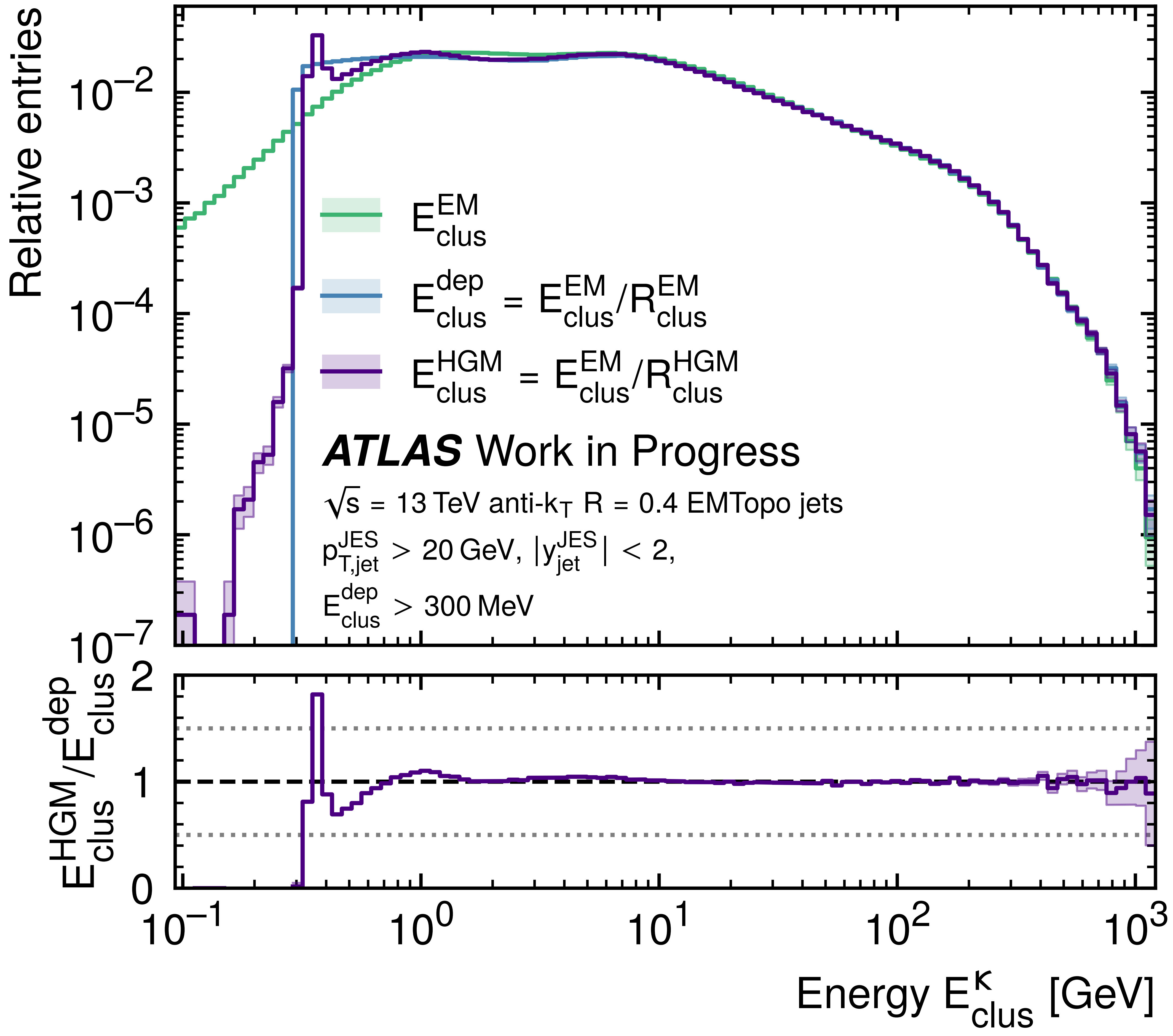
$\vec{c}$	*centre of gravity of cluster, measured from the nominal vertex* $(x = 0, y = 0, z = 0)$ *in ATLAS*
$\vec{x}_i$	*geometrical centre of a calorimeter cell in the cluster, measured from the nominal detector centre of ATLAS*
$\vec{s}$	*particle direction of flight (shower axis)*
$\Delta\alpha$	*angular distance* $\Delta\alpha = \angle(\vec{c}, \vec{s})$ *between cluster centre of gravity and shower axis* $\vec{s}$
λ_i	*distance of cell at* $\vec{x}_i$ *from the cluster centre of gravity measured along shower axis* $\vec{s}$ $(\lambda_i < 0$ *is possible)*
r_i	*radial (shortest) distance of cell at* $\vec{x}_i$ *from shower axis* $\vec{s}$ $(r_i \geq 0)$

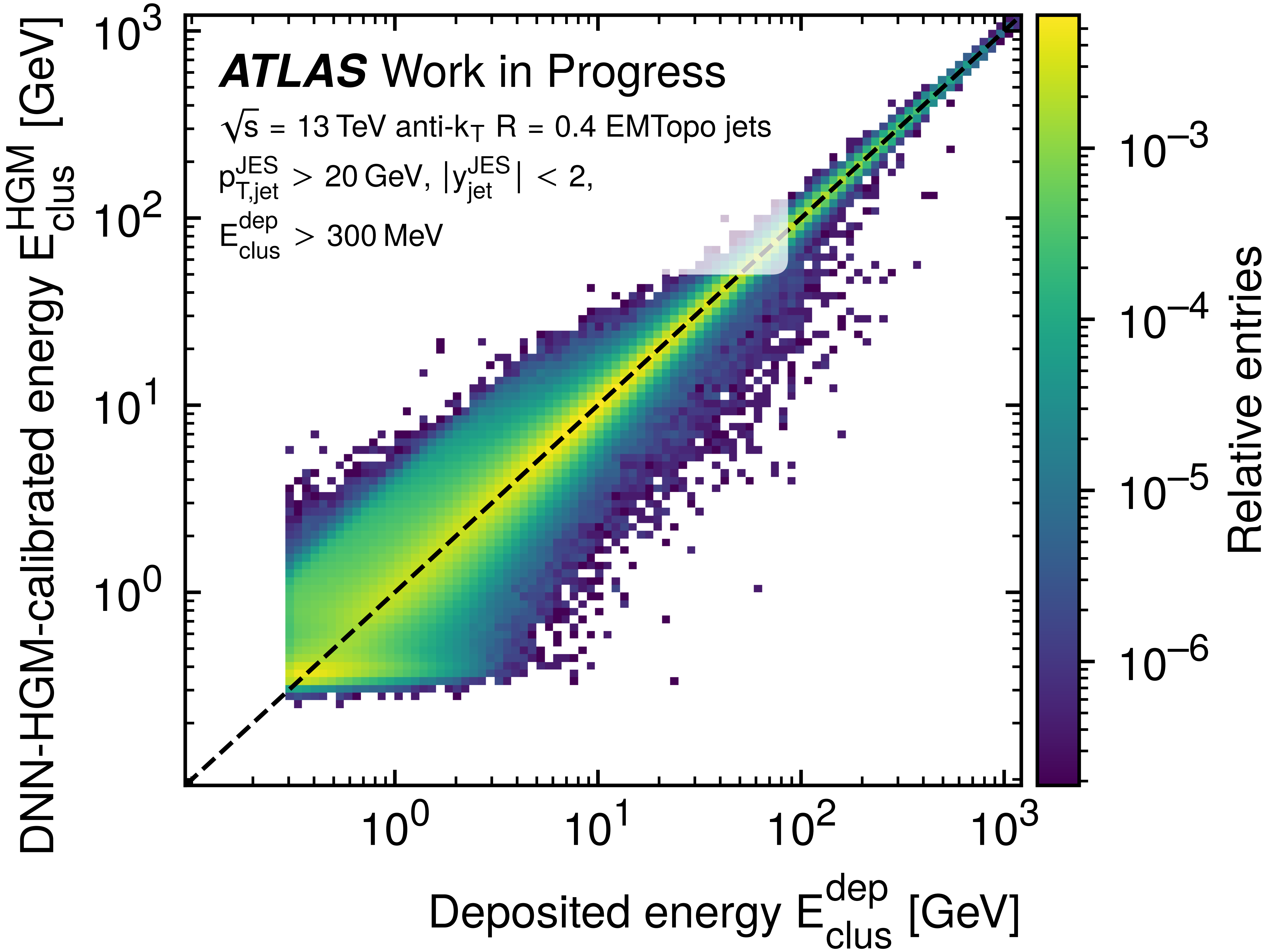
Response results





Energy results





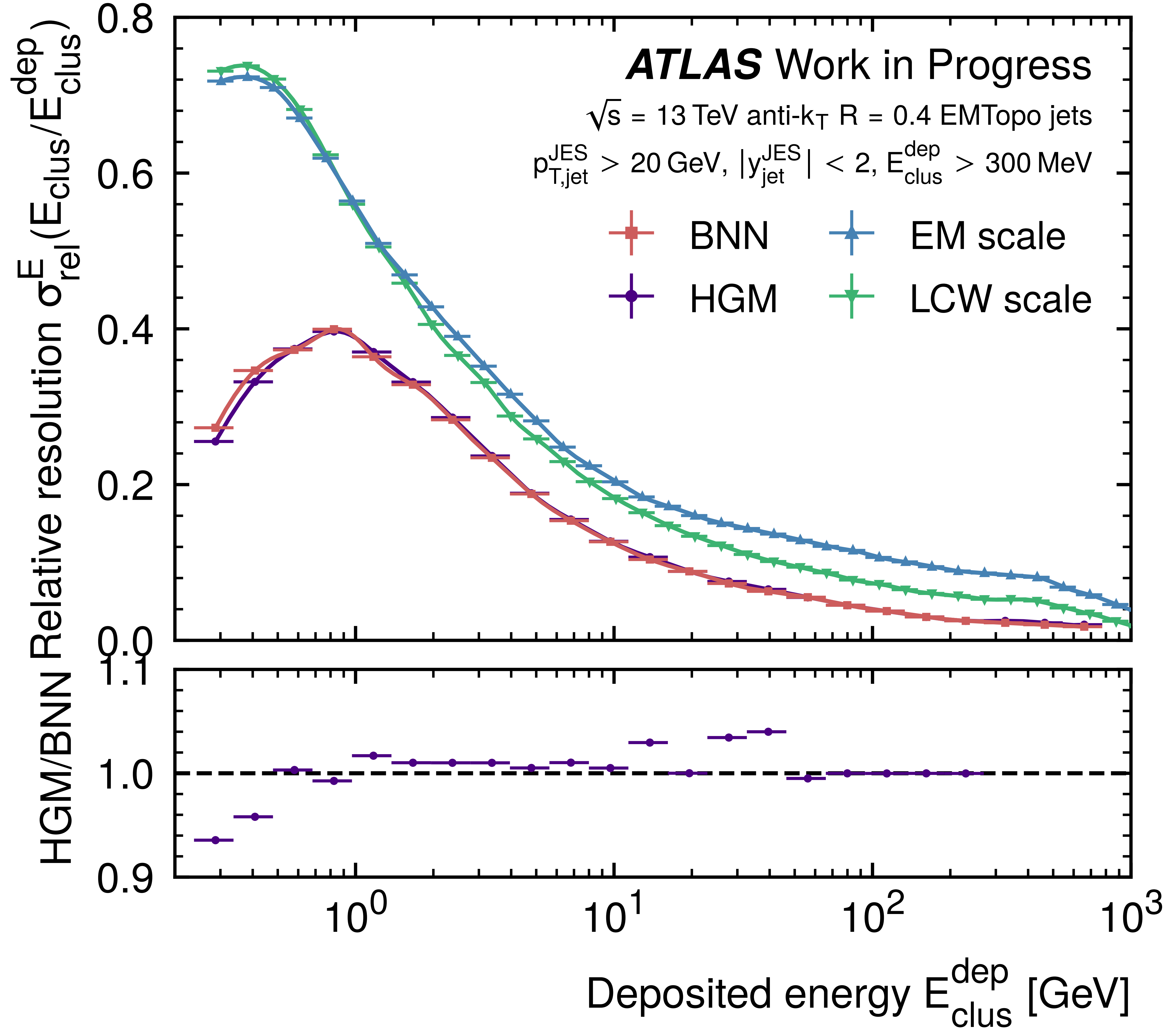
Relative local energy resolution

mc16

$$\sigma^E_{rel} = \frac{Q^w_{f=68\%}(E^\kappa_{clus}/E^{dep}_{clus})}{2\,\langle E^\kappa_{clus}/E^{dep}_{clus}\rangle_{med}}$$

$\kappa \in$ {EM, LCW, BNN, HGM}

- Better than EM scale and current hadronic calibration
- Performance similar to BNN in all features



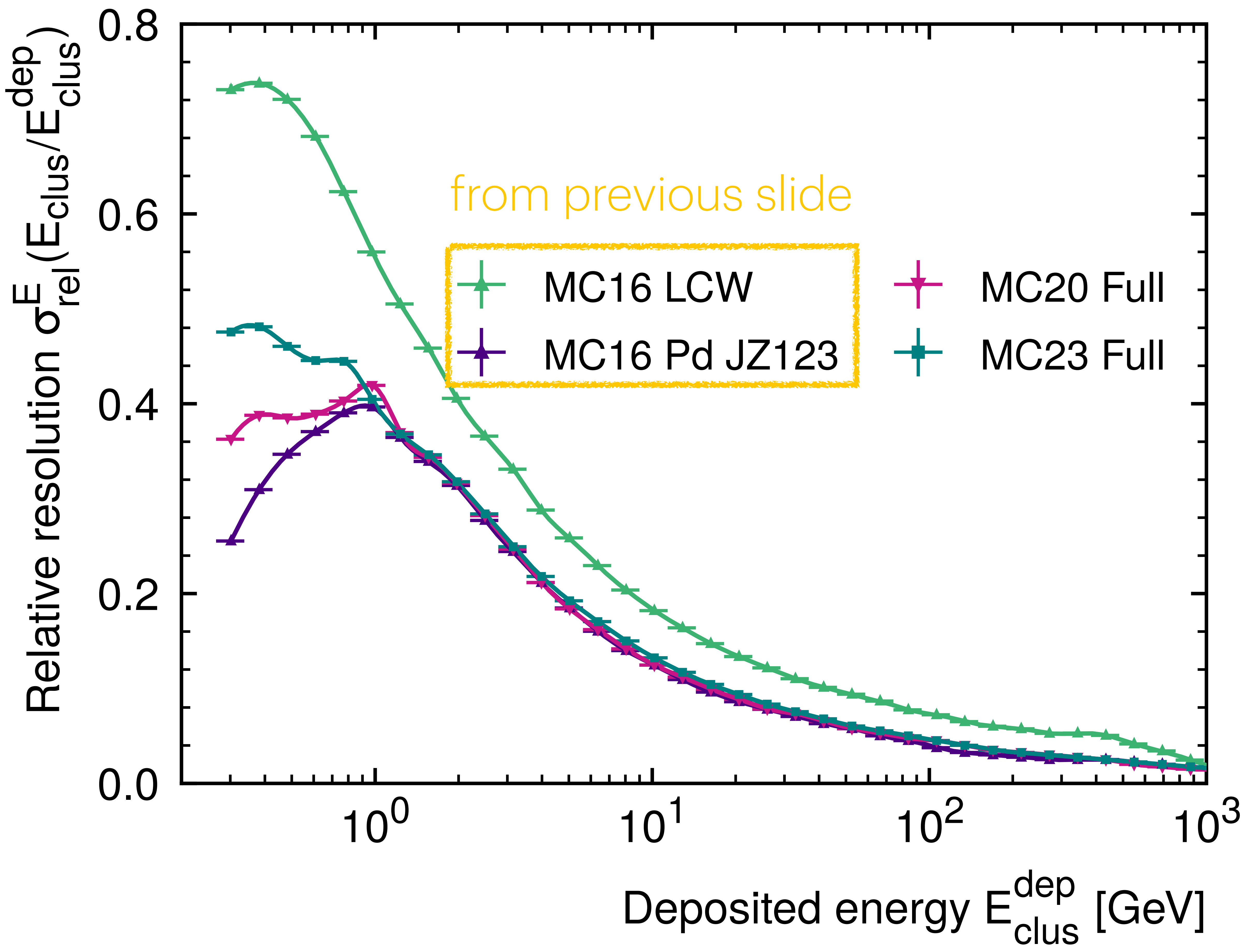
Relative local energy resolution

$$\sigma_{rel}^{E} = \frac{Q^{w}_{f=68\%}(E^{\kappa}_{clus}/E^{dep}_{clus})}{2\,\langle E^{\kappa}_{clus}/E^{dep}_{clus}\rangle_{med}}$$

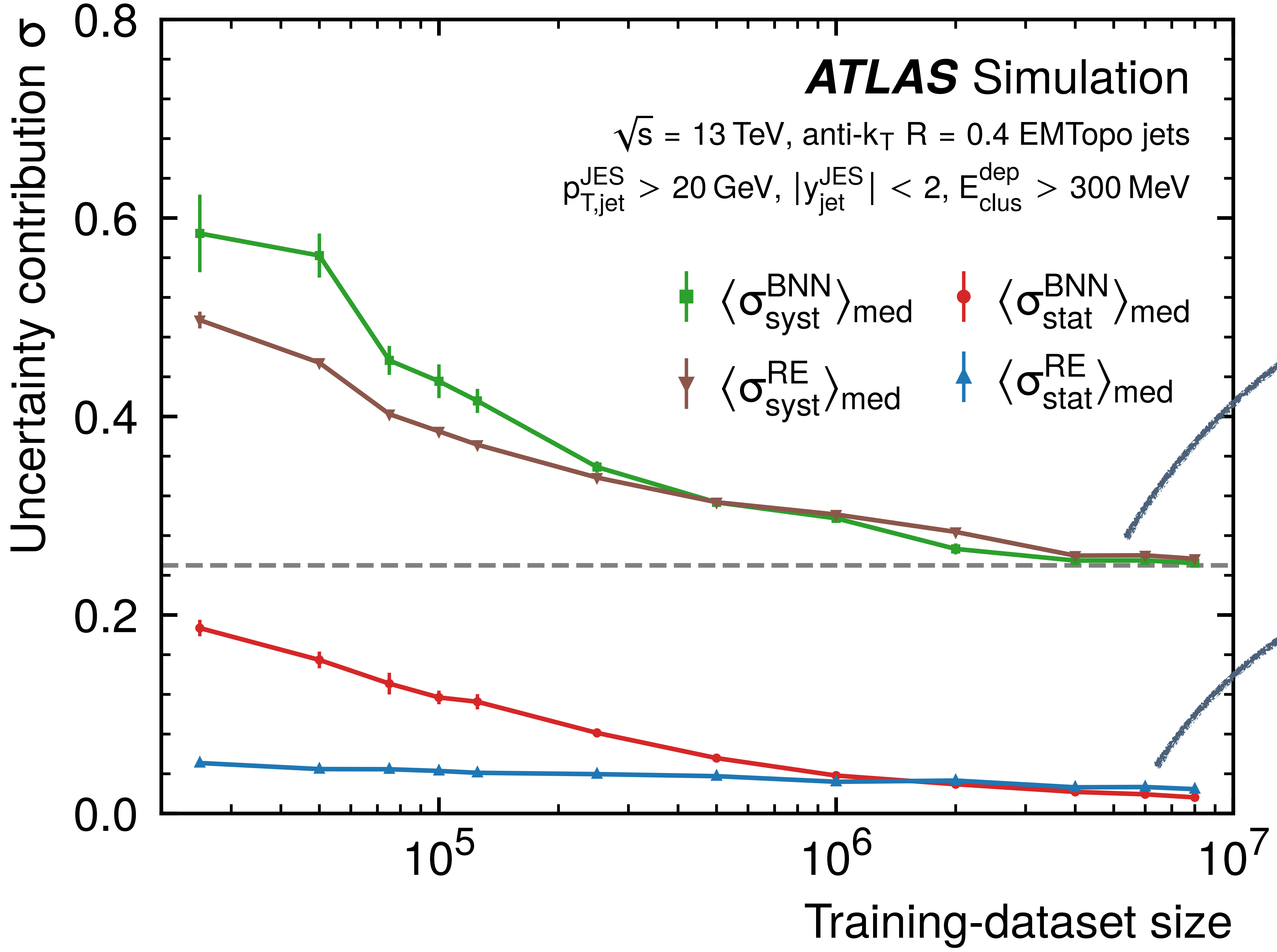
$\kappa \in$ {LCW MC16, HGM MC16, HGM MC20, HGM MC23}

- Differences in performance for MC16 and MC20/23 at low energies

- Same trend in all features



BNN Uncertainties



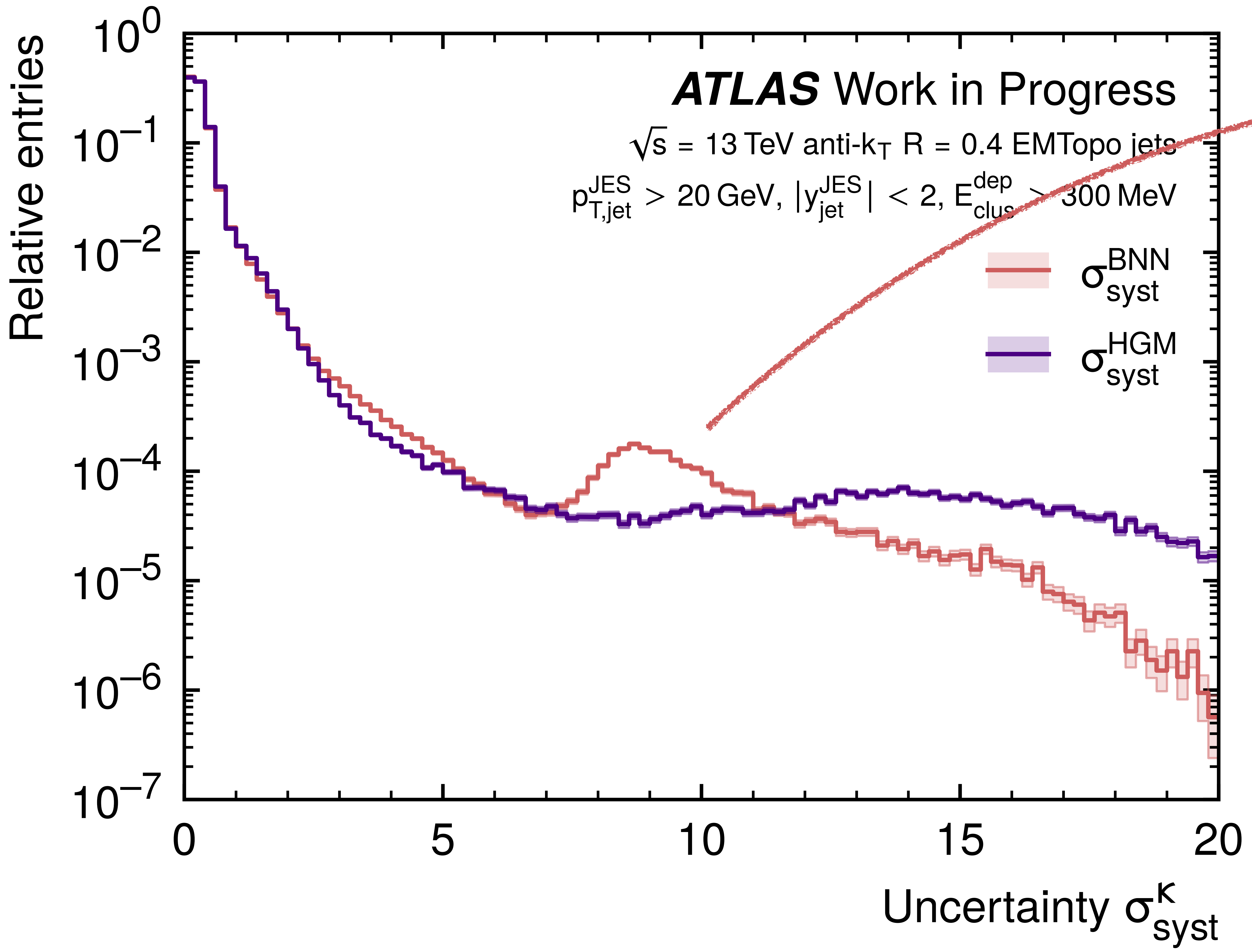
ATL-COM-PHYS-2024-747

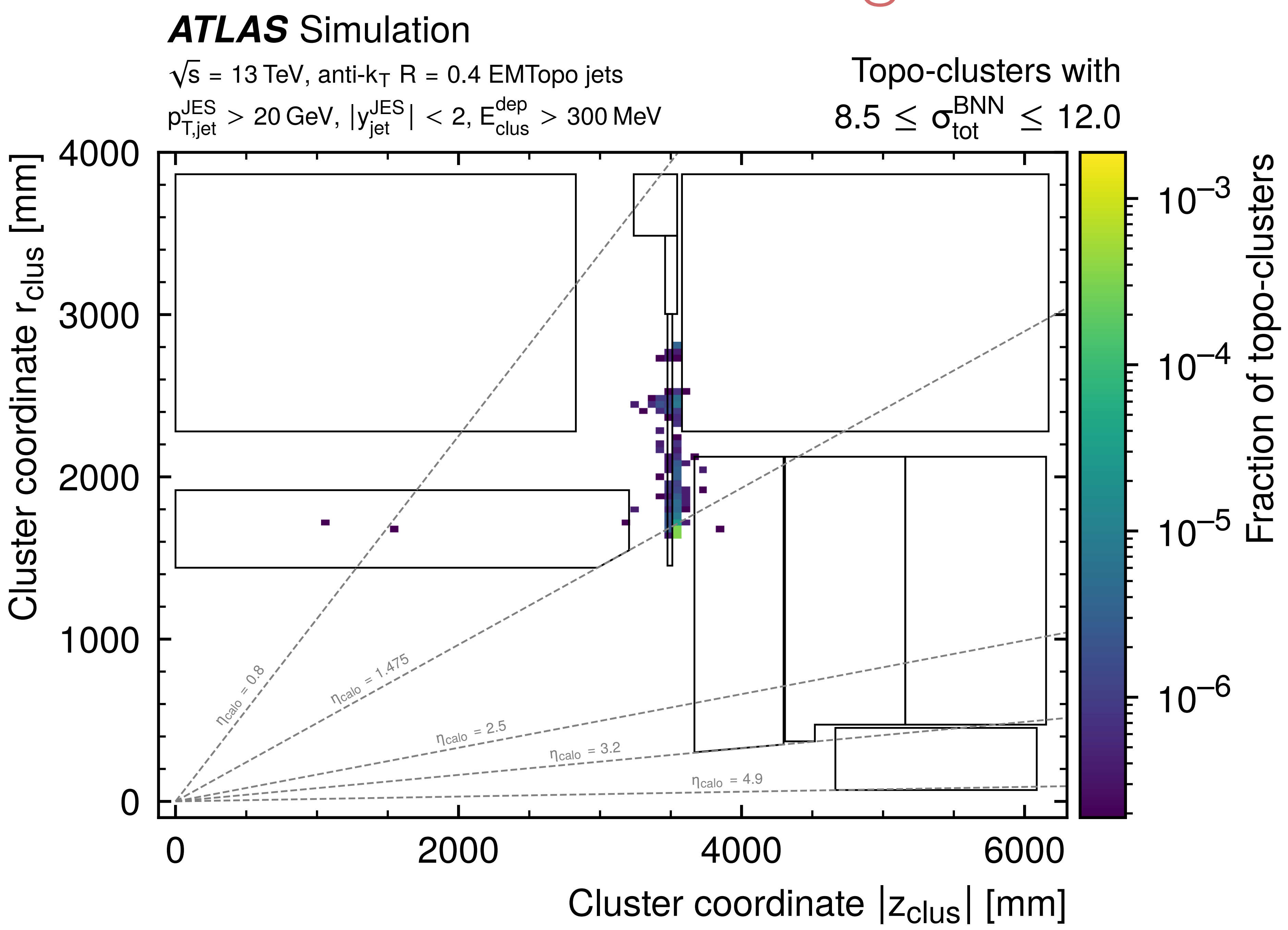
σ_{syst} is the relevant uncertainty

$\sigma_{stat} \to 0$ for our training data size

Predicted uncertainty

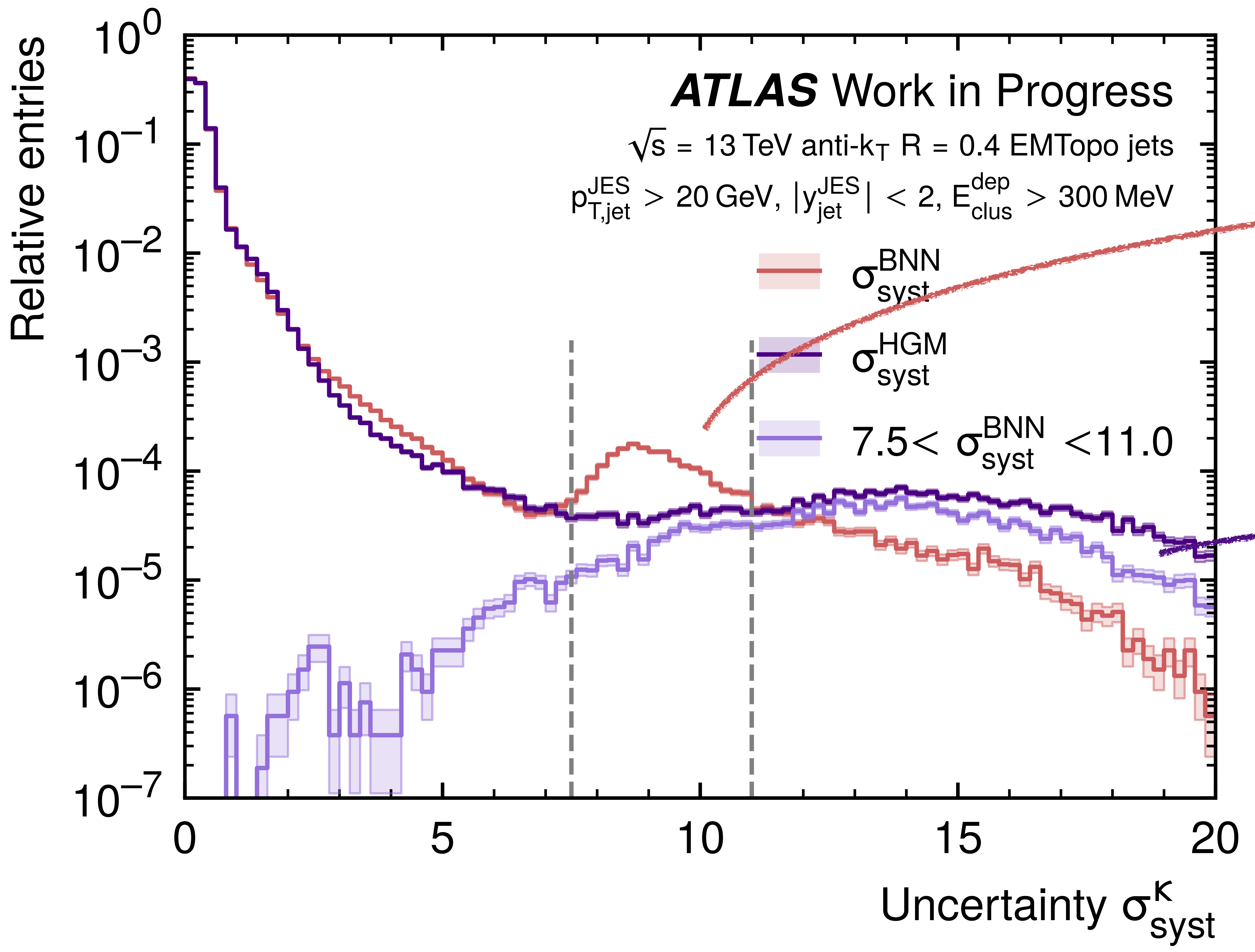
Weak detector region!





ATL-COM-PHYS-2024-747

Predicted uncertainty



Tracing the location of topo-clusters

⇒ Weak detector region!

(Tile Gap)

Uncertainties shifted to higher values

Uncertainties - Pull

Analysis of the uncertainty prediction per topo-cluster

$$t(x) = \frac{R_{clus}^{model}(x) - R_{clus}^{EM}}{\sigma_{tot}^{model}(x)}$$

→ Prediction for $\log_{10} R_{clus}^{model}$

$$t^{log}(x) = R_{clus}^{model}(x) \cdot \ln 10 \cdot \frac{\log_{10} R_{clus}^{model} - \log_{10} R_{clus}^{EM}}{\sigma_{tot}^{model}(x)}$$

Pull distributions

mc16

$$t(x) = \frac{R^{model}_{clus}(x) - R^{EM}_{clus}}{\sigma^{model}_{tot}(x)}$$

- Uncertainty includes stochastic effects

  $\Rightarrow$ Expect $\mathcal{N}(0,1)$

- Conservative uncertainty estimate

- Both methods provide equivalent predictions