Calibrating ATLAS calorimeter signals using an uncertainty-aware precision network

Lorenz Vogel

November 6, 2024

Institute for Theoretical Physics
Heidelberg University, Germany

ML4Jets Workshop 2024 — LPNHE, Paris

Application of Bayesian neural networks (BNNs) for the calibration of topological cell clusters in the ATLAS calorimeters

— in collaboration with T. Heimel, P. Loch, T. Plehn, J. M. Sardain and P. Velie

Today's outline

1. Motivation
2. BNN-calibration performance
3. BNN-learned uncertainties
4. Repulsive ensembles
5. Summary and outlook

Motivation

Topo-cluster calibration

Why topo-cluster calibration?

[arXiv:1603.02934, ATL-PHYS-PUB-2023-019]

- clusters of topologically connected cell signals principal calorimeter signals
- calibrated to correctly measure energy deposited by EM showers
- do not compensate for invisible energy losses in complex hadronic showers

multi-dimensional correlated calibration

$$\underbrace{E^{\text{EM}}_{\text{clus}}}_{\text{measured}} \xrightarrow{\text{hadronic calibration}} \underbrace{E^{\text{had}}_{\text{clus}} = E^{\text{dep}}_{\text{clus}}}_{\text{expected/goal}}$$

Standard ATLAS approach: local cluster weighting (LCW)

- four-step sequence with multi-dimensional, binned look-up tables
- non-smooth, step-like transitions between scale factors, no feature correlations, no pile-up measures

Topo-cluster calibration

Why **topo-cluster calibration**?

[arXiv:1603.02934, ATL-PHYS-PUB-2023-019]

- clusters of topologically connected cell signals principal calorimeter signals
- calibrated to correctly measure energy deposited by EM showers
- do not compensate for invisible energy losses in complex hadronic showers

regression network:
response over phase space

$$\mathcal{R}^{\text{BNN}}_{\text{clus}}(\mathcal{X}_{\text{clus}}) \overset{\text{train}}{\approx} \mathcal{R}^{\text{EM}}_{\text{clus}} = \frac{E^{\text{EM}}_{\text{clus}}}{E^{\text{dep}}_{\text{clus}}}$$

15 topo-cluster features → dataset D_{train} given by $(\mathcal{X}_{\text{clus}}, \mathcal{R}^{\text{EM}}_{\text{clus}})$

$$\mathcal{X}_{\text{clus}} = \Big\{ \underbrace{E^{\text{EM}}_{\text{clus}}, y^{\text{EM}}_{\text{clus}}}_{\text{kinematics}}, \overbrace{\zeta^{\text{EM}}_{\text{clus}}}^{\text{signal relevance}}, \underbrace{\text{Var}_{\text{clus}}(t_{\text{cell}}), \lambda_{\text{clus}}, |\vec{c}_{\text{clus}}|, \langle\rho_{\text{cell}}\rangle, \langle\mathfrak{m}^2_{\text{long}}\rangle, \langle\mathfrak{m}^2_{\text{lat}}\rangle, p_{\text{T}}D, f_{\text{emc}}}_{\text{shower nature (position, compactness, signal density, internal time structure)}}, \overbrace{f_{\text{iso}}}^{\text{topology}}, \underbrace{t_{\text{clus}}, N_{\text{PV}}, \mu}_{\text{pile-up}} \Big\}$$

Topo-cluster calibration

Why **topo-cluster calibration**?

[arXiv:1603.02934, ATL-PHYS-PUB-2023-019]

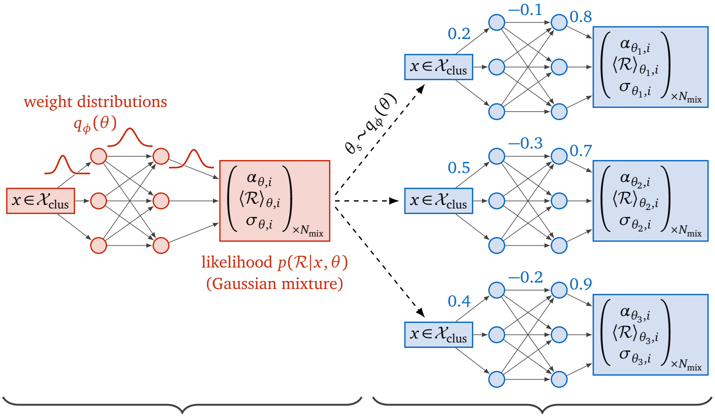
- clusters of topologically connected cell signals principal calorimeter signals
- calibrated to correctly measure energy deposited by EM showers
- do not compensate for invisible energy losses in complex hadronic showers

Modern **(B)NNs** for local topo-cluster calibration correcting for this non-compensation

- single-step training
- exploiting correlations
- smooth and multi-dimensional
- **control and uncertainties key** (access to bottom-up systematics)

[Ph.D. Thesis of Y. Gal, arXiv:2211.01421]

BNNs — Bayesian neural networks



training: parameters linking nodes described by weight distributions $q_\phi(\theta)$

inference: sample from learned weight distributions $\theta_s \sim q_\phi(\theta)$

BNNs **learn distributions of network parameters**, defining output distribution

[arXiv:2003.11099, arXiv:2206.14831, arXiv:2211.01421]

- training: parameters θ described by weight distributions $q_\phi(\theta) \approx p(\theta|D_{\text{train}})$
- inference: sample from weight distributions to get ensemble of networks

BNNs — learning weight distributions

BNNs learn distributions of network parameters, defining output distribution

$\mathcal{R}(x)$ given by probability $p(\mathcal{R})$ encoded in weight configurations:

$$p(\mathcal{R}) = \int \mathrm{d}\theta \; p(\mathcal{R}|\theta)\, p(\theta|D_{\text{train}})$$

BNNs — variational inference

BNNs learn distributions of network parameters, defining output distribution

$\mathcal{R}(x)$ given by probability $p(\mathcal{R})$ encoded in weight configurations:

$$p(\mathcal{R}) = \int \mathrm{d}\theta \; p(\mathcal{R}|\theta) \; p(\theta|D_{\text{train}}) \approx \int \mathrm{d}\theta \; p(\mathcal{R}|\theta) \; q_\phi(\theta)$$

training by **variational approximation** of $p(\theta|D_{\text{train}})$ with simplified and tractable $q_\phi(\theta)$

BNNs — loss function

BNNs learn distributions of network parameters, defining output distribution

$\mathcal{R}(x)$ given by probability $p(\mathcal{R})$ encoded in weight configurations:

$$p(\mathcal{R}) = \int d\theta \, p(\mathcal{R}|\theta) \, p(\theta|D_\text{train}) \approx \int d\theta \, p(\mathcal{R}|\theta) \, q_\phi(\theta)$$

Similarity by minimizing KL-divergence:

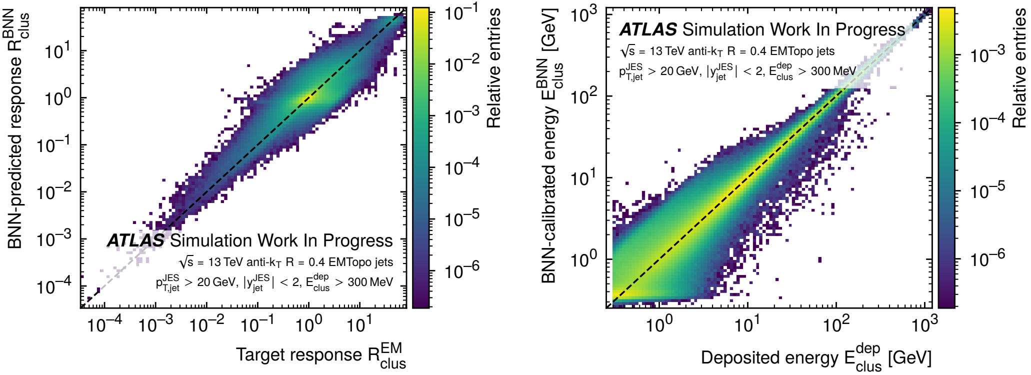
$$\min_\phi D_\text{KL}[q_\phi(\theta), p(\theta|D_\text{train})] \xrightarrow{\text{Bayes}} \mathcal{L}_\text{BNN} = \underbrace{D_\text{KL}[q_\phi(\theta), p_\text{prior}(\theta)]}_{\text{regularization}} - \underbrace{\left\langle \log p(D_\text{train}|\theta) \right\rangle_{\theta \sim q_\phi(\theta)}}_{\text{log-likelihood}}$$

BNN-calibration performance

BNN — response prediction and energy calibration

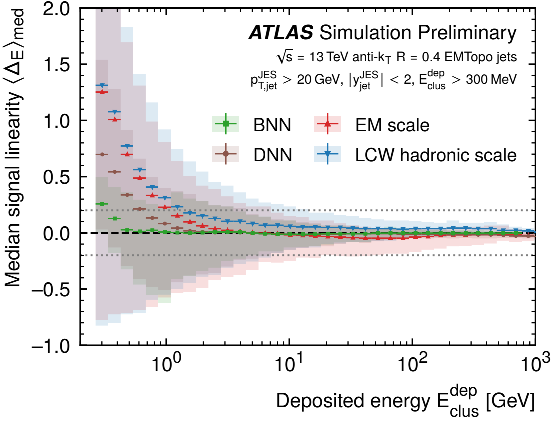
$$\mathcal{R}^{\text{BNN}}_{\text{clus}} \xrightarrow{\text{train}} \mathcal{R}^{\text{EM}}_{\text{clus}} = \frac{E^{\text{EM}}_{\text{clus}}}{E^{\text{dep}}_{\text{clus}}}$$

$$E^{\text{BNN}}_{\text{clus}} = \frac{E^{\text{EM}}_{\text{clus}}}{\mathcal{R}^{\text{BNN}}_{\text{clus}}} \longrightarrow E^{\text{dep}}_{\text{clus}}$$



agreement of BNN prediction and regression target:
correlation curves for predicted response and calibrated energy look promising

BNN — median signal linearity

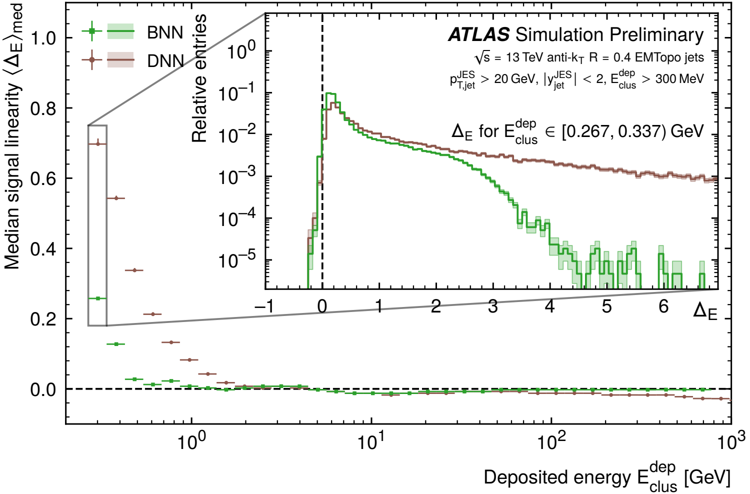


Signal linearity: ratio of calibrated over deposited energy

$$\Delta_E^\kappa = \frac{E^\kappa_{\text{clus}}}{E^{\text{dep}}_{\text{clus}}} - 1 \quad \text{with} \quad E^\kappa_{\text{clus}} = \frac{E^{\text{EM}}_{\text{clus}}}{\mathcal{R}^\kappa_{\text{clus}}}$$

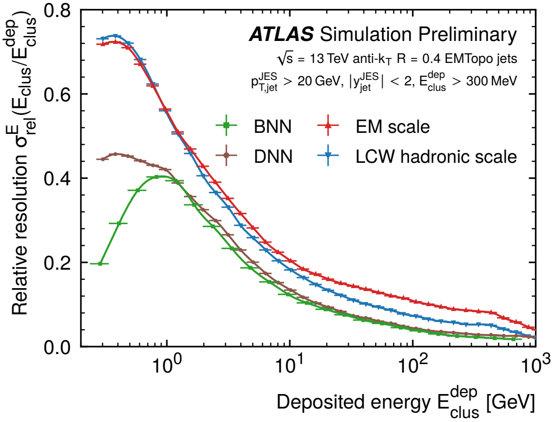
- scales $\kappa \in \{\text{EM}, \text{LCW}, \text{DNN}, \text{BNN}\}$
- should peak at zero after successful calibration
- evaluated as function of features and deposited energy
- BNN better over whole energy range, most significant at low energies

BNN — bin-wise signal linearity



BNN-derived calibration shows significantly suppressed tails compared to DNN

BNN — relative local energy resolution

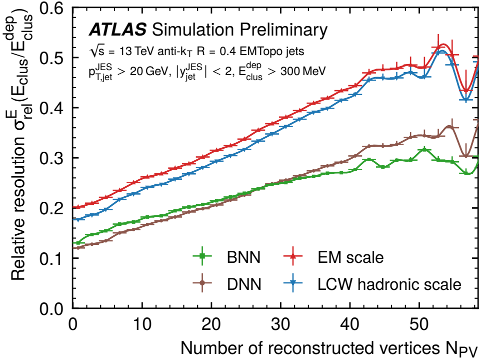


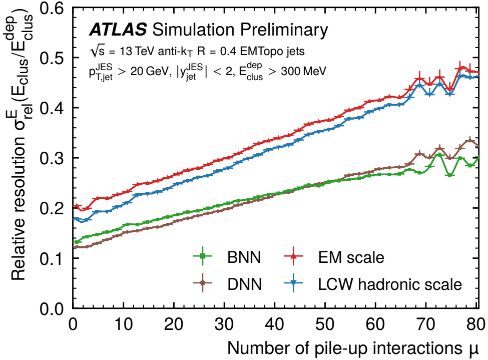
Relative local energy resolution:

$$\sigma_{rel}^{E} = \frac{Q_{f=68\,\%}^{w}}{2\langle E_{clus}^{\kappa}/E_{clus}^{dep}\rangle_{med}}$$

- $Q_{f=68\,\%}^{w} \equiv 68\,\%$ inter-quantile range
- BNN better over whole energy range, spectactular at low energies
- BNN learns signal-source transition from inelastic hadronic interactions to ionisation-dominated signals

BNN — relative local energy resolution

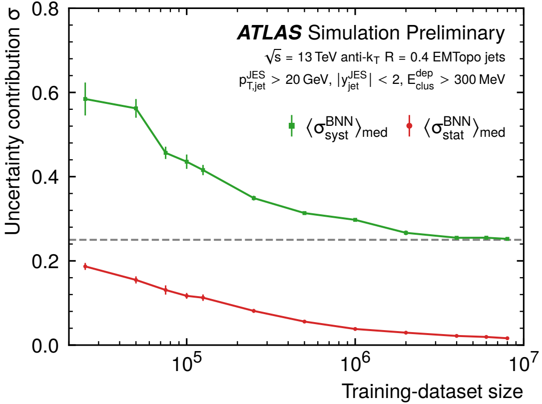




relative local energy resolution vs in-time and out-of-time pile-up activity
→ BNN shows cluster-by-cluster pile-up mitigation

BNN-learned uncertainties

BNN — systematic and statistical uncertainties



Learned σ_{tot} with two origins:

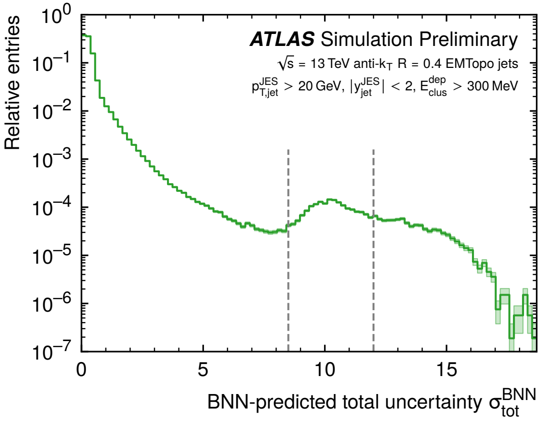
[arXiv:1904.10004, arXiv:2003.11099, arXiv:2206.14831]

- **statistics σ_{stat}**
  training statistics,
  vanishing for good training statistics
- **systematics σ_{syst}**
  stochastic training data,
  limited network expressivity,
  bad hyper-parameters,
  plateau for good training statistics

For well-trained LHC models:

$$\sigma_{\text{tot}} \equiv \sqrt{\sigma_{\text{syst}}^2 + \sigma_{\text{stat}}^2} \approx \sigma_{\text{syst}} \gg \sigma_{\text{stat}}$$

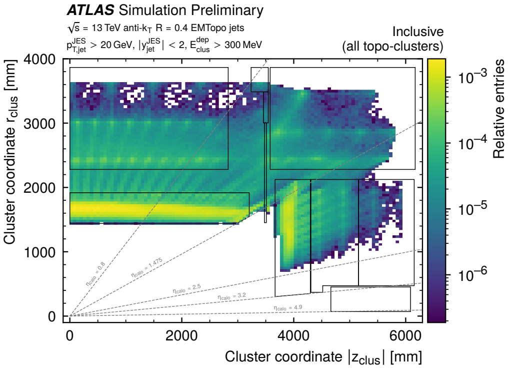
BNN — uncertainties as explainable AI

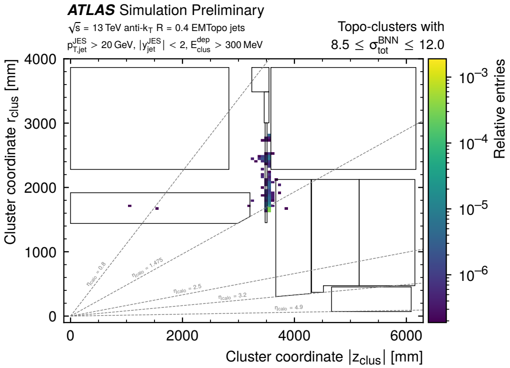


Use BNN uncertainty to understand data

- uncertainty spectrum shows distinctive secondary maximum
- what feature leads the BNN to flag these topo-clusters with large learned uncertainties?
- analyze anomalous clusters in terms of detector geometry

BNN — uncertainties as explainable AI



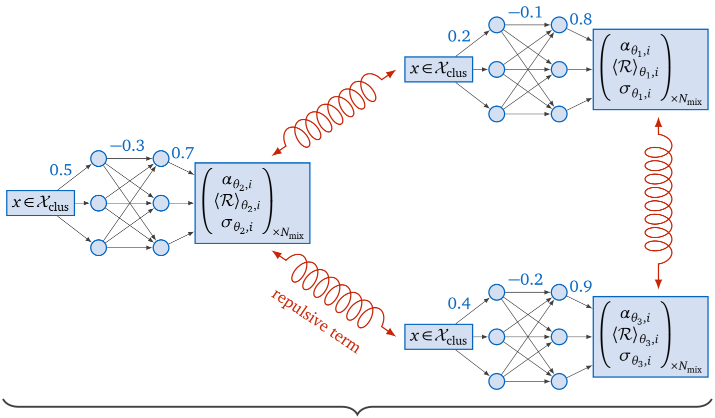


large uncertainties from tile-gap scintillator region:

not a regular calorimeter → feature quality in this region is insufficient

Repulsive ensembles

REs — repulsive ensembles

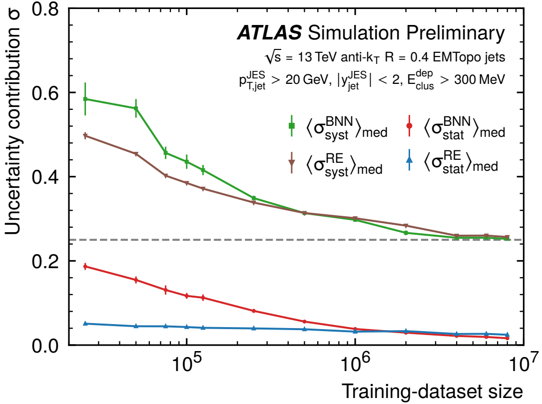


training: repulsive term connecting function space of all simultaneously trained networks forces ensemble to spread out and cover loss around actual minimum

Alternative way
for uncertainty estimation
[arXiv:2106.11642, arXiv:2211.01421, arXiv:240313899]

- regular ensembles do not sample from weight posterior
- introduce repulsive force between ensemble members during optimization such that $\theta \sim p(\theta|D_{\text{train}})$
- **repulsive term** ensures that uncertainty covers probability distribution over space of network functions

REs — systematic and statistical uncertainties

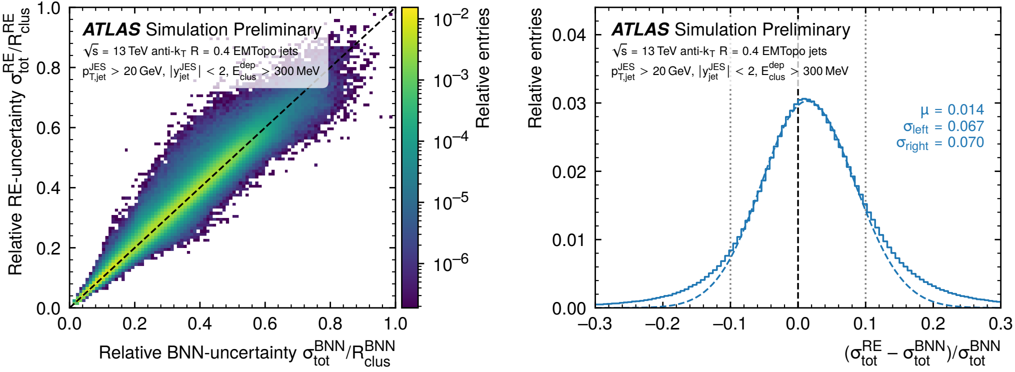


Repulsive force ensures that uncertainty covers probability distribution over space of network functions

[arXiv:2106.11642, arXiv:2211.01421, arXiv:240313899]

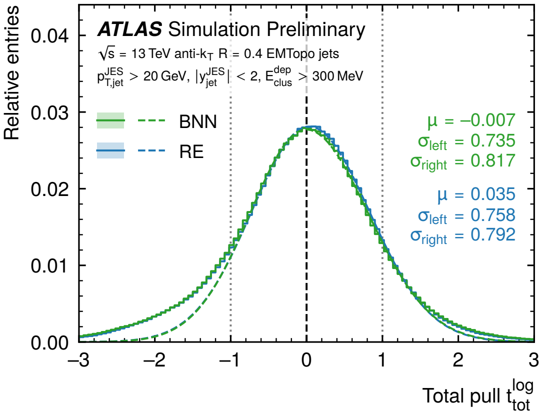
- gives two uncertainties
- **systematics σ_{syst}**
  plateau for good training statistics, part of likelihood (same as for BNN)
- **statistics σ_{stat}**
  vanishing for good training statistics (but with flatter slope)

BNN and RE — consistency check



10 % agreement between uncertainty estimates
and both uncertainty predictions track each other well

BNN and RE — uncertainties vs data spread



Pull: central prediction and uncertainty
Does uncertainty cover data spread?

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

- evaluated in $\log_{10} \mathcal{R}_{\text{clus}}$ space
- stochastic data defining shape
- Gaussian with order-one width
- BNN and RE errors consistent
- per-cluster error conservative

Summary and outlook

Summary and outlook

ATLAS Paper

JETM-2024-01

3rd November 2024

Journal: SciPost Phys.

Draft version 1.0

Calibrating calorimeter signals in the ATLAS experiment using an uncertainty-aware precision neural network

The ATLAS Collaboration[1]

The ATLAS experiment at the Large Hadron Collider (LHC) explores the use of modern neural networks for a multi-dimensional calibration of its calorimeter signal defined by clusters of topologically connected cells (topo-clusters). The Bayesian neural network (BNN) approach not only yields a continuous and smooth calibration function that improves performance relative to the standard calibration but also provides uncertainties on the calibrated energies for each topo-cluster. The results obtained by using a trained BNN are compared to the standard local hadronic calibration and to a calibration provided by training a deep neural network (DNN). The uncertainties predicted by the BNN are interpreted in the context of a fractional contribution to the systematic uncertainties of the trained calibration. They are also compared to uncertainty predictions obtained from an alternative estimator featuring repulsive ensembles.

© 2024 CERN for the benefit of the ATLAS Collaboration.
Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.

[1] The full author list can be found at: https://atlas.web.cern.ch/Atlas/PUBNOTES/ATL-PHYS-PUB-2024-XXX/authorlist.pdf

Modern uncertainty-aware BNNs for multi-dimensional calorimeter-signal calibration

- continuous and smooth calibration of topo-clusters
- improved performance relative to LCW and DNN
- meaningful per-cluster systematics
- BNNs and REs: learn reliable uncertainties

Next steps:

- further tune BNN performance
- full performance study (apply trained calibration to data)
- **ATLAS paper in preparation...** coming soon!

  [ATLAS JETM-2024-04 preliminary public plots]

References and further reading

ML-based topo-cluster calibration

- ATLAS Collaboration
  The application of neural networks for the calibration of topological cell clusters in the ATLAS calorimeters
  ATLAS PUB Note (2023)

ML with uncertainties

- Y. Gal
  Uncertainty in Deep Learning
  Ph.D. Thesis, University of Cambridge (2016)
- T. Plehn, A. Butter, B. Dillon, T. Heimel, C. Krause and R. Winterhalder
  Modern Machine Learning for LHC Physicists
  arXiv:2211.01421 [hep-ph] (continuously updated on website)

References and further reading

Bayesian neural networks (BNNs) and repulsive ensembles (REs)

- G. Kasieczka, M. Luchmann, F. Otterpohl and T. Plehn
  Per-object systematics using deep-learned calibration
  SciPost Phys. 9, 089 (2020), arXiv:2003.11099 [hep-ph]
- S. Badger, A. Butter, M. Luchmann, S. Pitz and T. Plehn
  Loop amplitudes from precision networks
  SciPost Phys. Core 6, 034 (2023), arXiv:2206.14831 [hep-ph]
- F. D'Angelo and V. Fortuin
  Repulsive Deep Ensembles are Bayesian
  arXiv:2106.11642 [cs.LG]
- L. Röver, B. M. Schäfer and T. Plehn
  PINNferring the Hubble Function with Uncertainties
  arXiv:2403.13899 [astro-ph.CO]

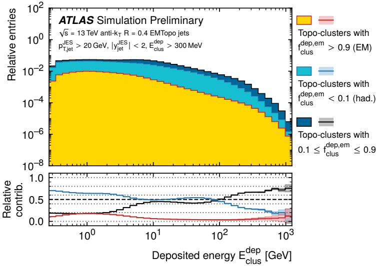
Backup slides...

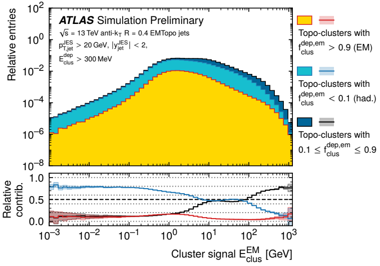
Dataset — topo-cluster features

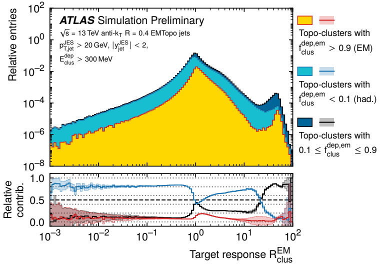
Table 1: The dataset consists of topo-clusters reconstructed in MC simulations of full proton-proton collision events at $\sqrt{s} = 13\,\text{TeV}$ (LHC Run 2) with multi-jet final states

category	symbol	description / comment
kinematics	$E^{\text{EM}}_{\text{clus}}$, $y^{\text{EM}}_{\text{clus}}$	cluster signal and rapidity at the EM energy scale
signal strength	$\zeta^{\text{EM}}_{\text{clus}}$	signal significance
timing time structure	t_{clus} $\text{Var}_{\text{clus}}(t_{\text{cell}})$	signal timing variance of the cell-time distribution in the cluster
shower depth	λ_{clus} $\lvert\vec{c}_{\text{clus}}\rvert$	distance of the CoG from the calorimeter front face distance of the CoG from the nominal vertex
shower shape, compactness	f_{emc} $\langle\rho_{\text{cell}}\rangle$, $p_{\text{T}}D$ $\langle\mathfrak{m}^2_{\text{long}}\rangle$, $\langle\mathfrak{m}^2_{\text{lat}}\rangle$	energy fraction in the EM calorimeter (EMC) cluster signal density and signal compactness energy dispersion along/perpendicular to main cluster axis
topology	f_{iso}	cluster isolation measure
pile-up	N_{PV} μ	number of reconstructed primary vertices number of pile-up interactions per bunch crossing

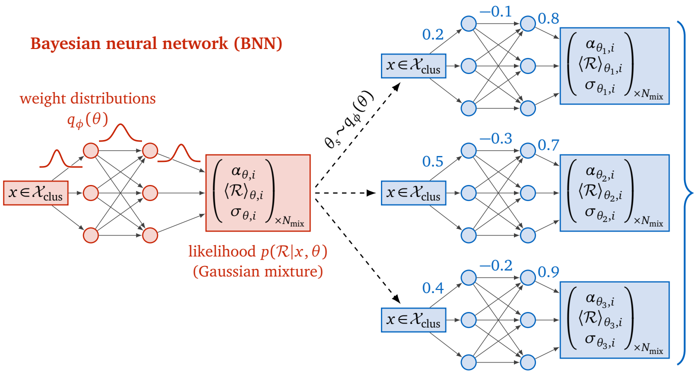
Dataset — energy and response distributions







BNN — network architecture



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

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

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

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

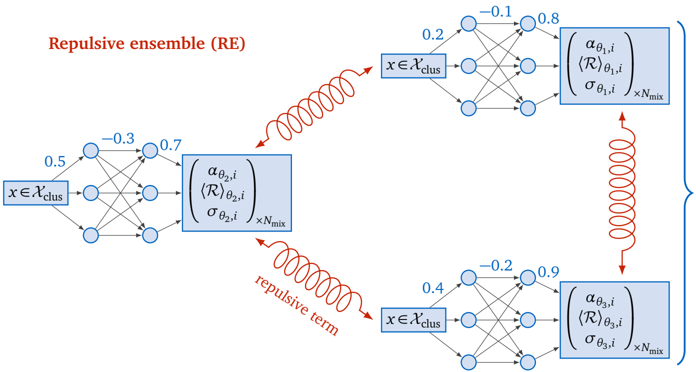
with

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

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

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

RE — network architecture



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

$$\mathcal{R}^{\mathrm{RE}}_{\mathrm{clus}}(x) = \arg\max_{\mathcal{R}} \frac{1}{N}\sum_{s=1}^{N} p(\mathcal{R}|x,\theta_s)$$

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

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

with

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

Training: Repulsive term connecting the function space of all N simultaneously trained networks forces the ensemble to spread out and cover the loss around the actual minimum

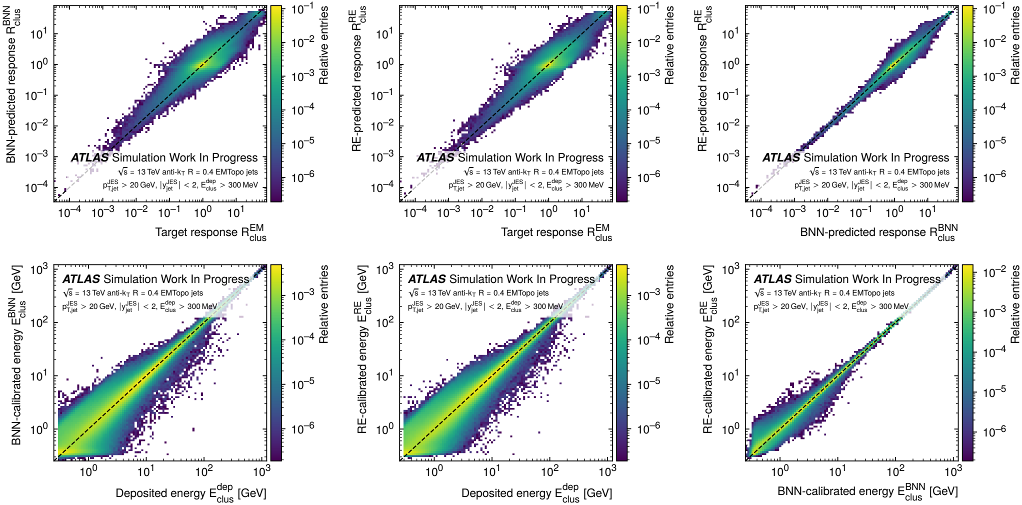
Inference: Same formulas as for the BNN, using the N simultaneously trained ensemble members

BNN — network setup and hyper-parameters

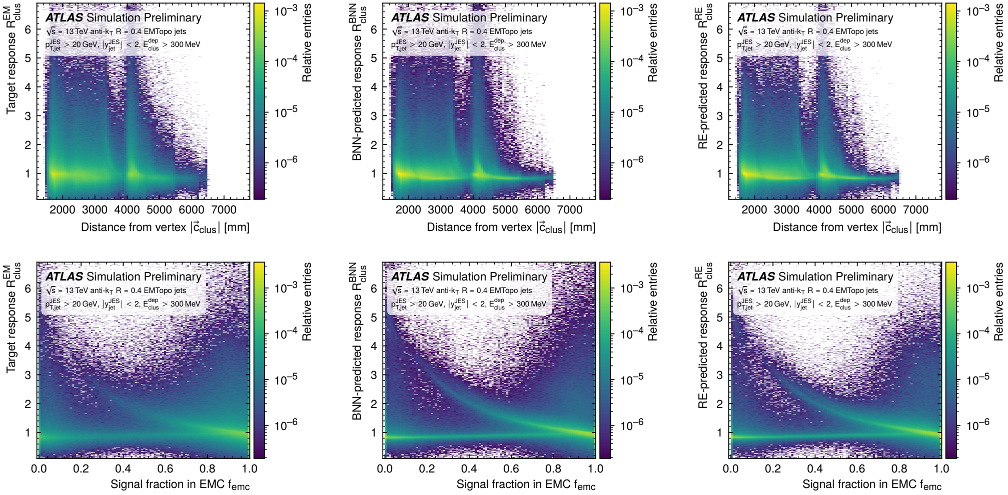
Table 2: BNN setup for the three-mode Gaussian mixture likelihood

hyper-parameter	**BNN architecture and setup**
likelihood loss	Gaussian mixture model (GMM)
number of modes (mixture components N_{mix})	3 (i.e. 9 output nodes)
number of layers and nodes per layer	4 hidden layers with $\{64, 64, 64, 64\}$ nodes
activation functions	ReLU (inner layers) and none (last layer)
prediction	maximum-likelihood value ("mode")
optimizer and learning rate (LR)	ADAM with LR $= 10^{-4}$
learning-rate scheduler	STEPLR, epochs $\{25, 100\}$, $\gamma = 0.1$
number of training epochs	150
batch size for training (testing)	4096 (512)
dataset sizes for training, validation, testing	$\{8.7\text{M}, 500\text{k}, 5.3\text{M}\}$
re-sampling for inference (Monte-Carlo samples S)	50 times

BNN and RE — response and energy correlation curves



EM, BNN and RE — response vs features



DNN, BNN and RE — signal linearity vs features