Machine learning-based identification and mitigation of pile-up contributions in topoclusters in ATLAS

Bastian Schuchardt

September 29, 2025

NWG Delitzsch
Physics Department TU Dortmund

Large Hadron Collider & ATLAS

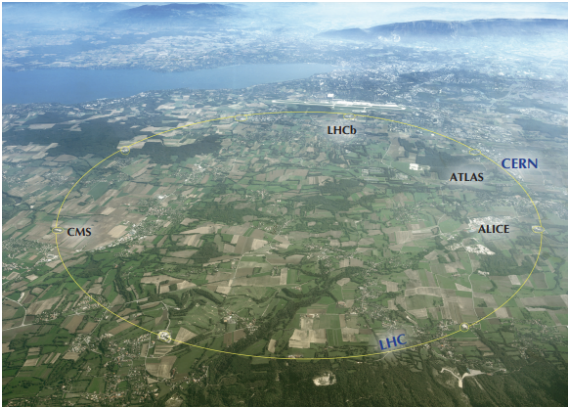


Figure: Aerial view of the LHC.

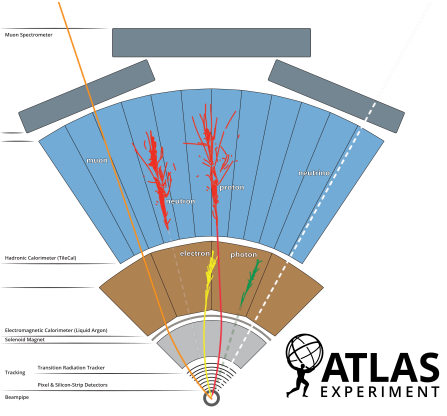


Figure: ATLAS Experiment.

Calorimeter

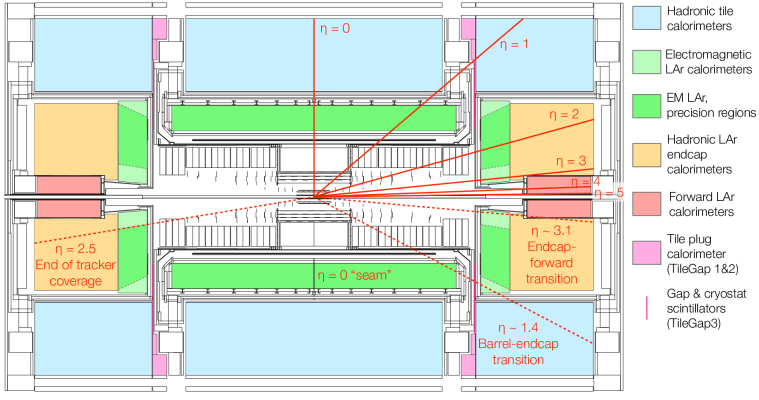


Figure: DOI: 10.1140/epjc/s10052-021-09402-3

What is a Jet?

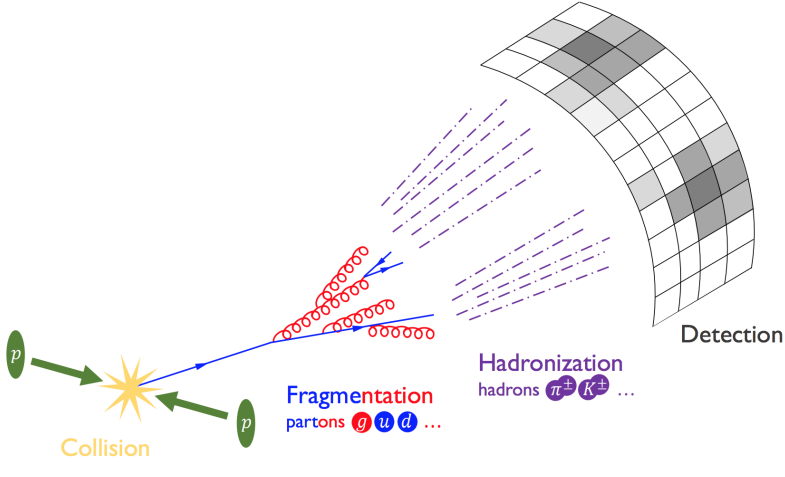


Figure: The Fractal Lives of Jets, Eric M. Metodiev

Pile-Up

- Hard-scatter vertex: interaction vertex with highest $\sum p_T^2$ of tracks (defines the collision of interest)
- Multiple *pp* interactions in the same or neighbouring bunch crossings add energy deposits not from the hard scatter or primary vertex
- In/out-of-time pile-up from the same/neighbouring bunch crossing
- Degrades signal quality → Must be mitigated

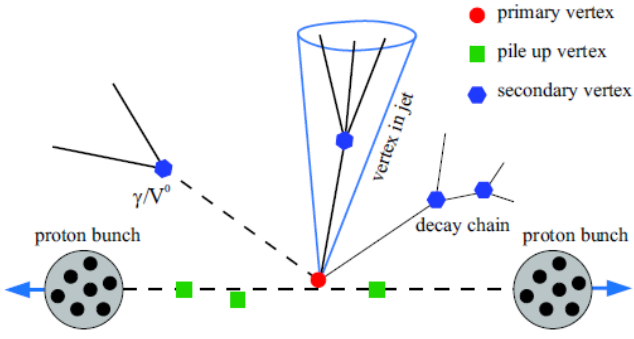
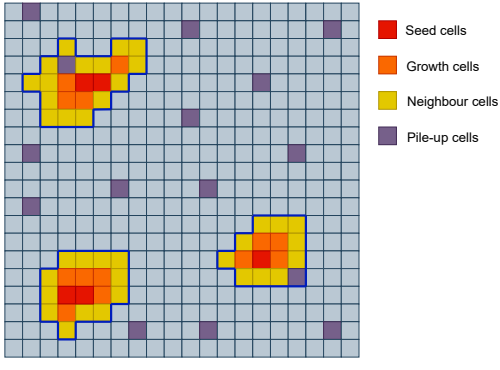


Figure: Pile-up: Taking a closer look at LHC

Topological Clusters

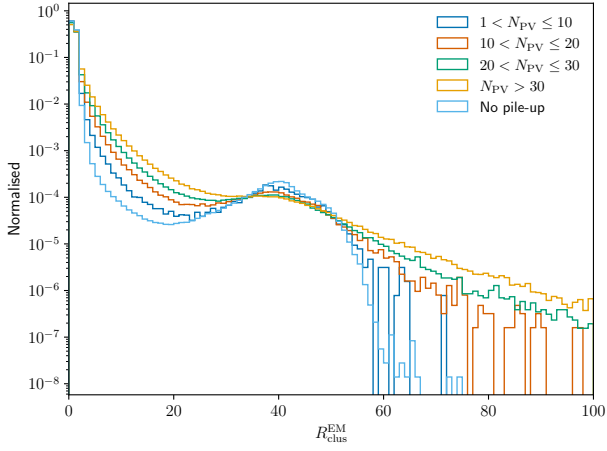
- Cells **cannot** directly be used as input objects for jet reconstruction algorithms
- Cells can be grouped in topoclusters
- Constructed from seed cells with large signal-to-noise ratios
  - $\zeta_{\text{cell}}^{\text{EM}} = \frac{E_{\text{cell}}^{\text{EM}}}{\sigma_{\text{noise, cell}}^{\text{EM}}}$

- **Seed threshold:** $|\zeta_{\text{cell}}| > 4$
- **Growth threshold:** $|\zeta_{\text{cell}}| > 2$
- **Neighbour threshold:** $|\zeta_{\text{cell}}| > 0$
- **Pile-up cells**



Cluster Response

- Cluster response $R^{\mathrm{EM}}_{\mathrm{clus}} = \frac{E^{\mathrm{EM}}_{\mathrm{clus}}}{R^{\mathrm{dep}}_{\mathrm{clus}}}$
- Number of primary vertices N_{PV}
- Higher N_{PV} → more pile-up

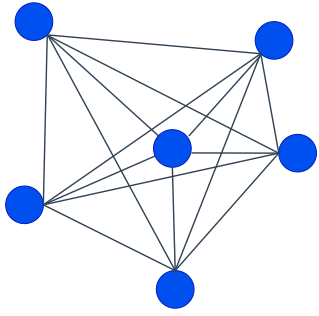


Motivation

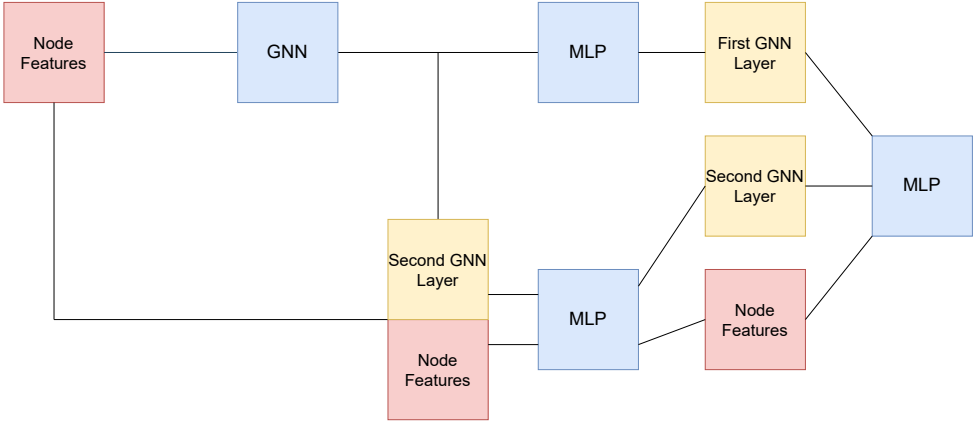
- Classify topoclusters as hard-scatter or mixed (hard-scatter + pile-up)
- Recalibrate the topoclusters of mixed origin
- Reduce the long tails in the cluster response distribution
- Deep Neural Networks (DNNs) were already developed for this task
- Improve with Graph Neural Networks (GNNs)
  - Leverage the geometry of topoclusters in a jet to extract more information
- Train and test models on Monte Carlo generated samples for Run 2 and Run 3 of ATLAS

Machine Learning Models

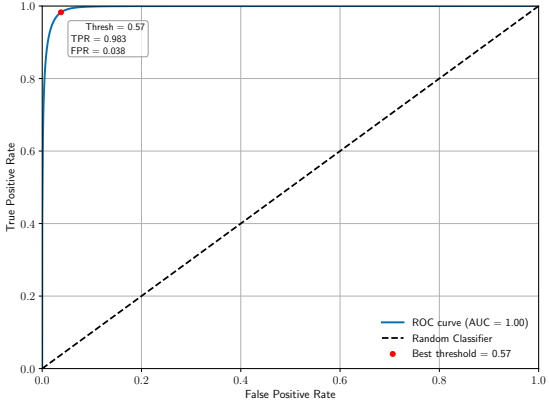
- DNN
  - Adopted from Giulia Fazzino's master's thesis
  - Baseline to improve on
- DNN trained with jet features (JetDNN)
  - For a fair performance comparison
- GNN
  - Utilises the geometry of topoclusters in a jet
  - Graph Convolutional Network (GCN) → Did not work as intended
  - Graph Attention Network (GAT)
  - Adapted and from hard-scatter vs. pile-up study

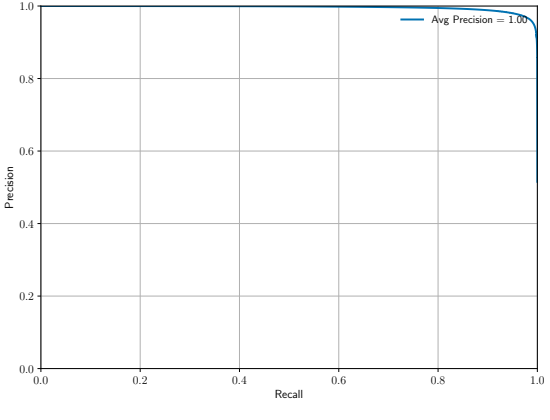


Graph Neural Networks

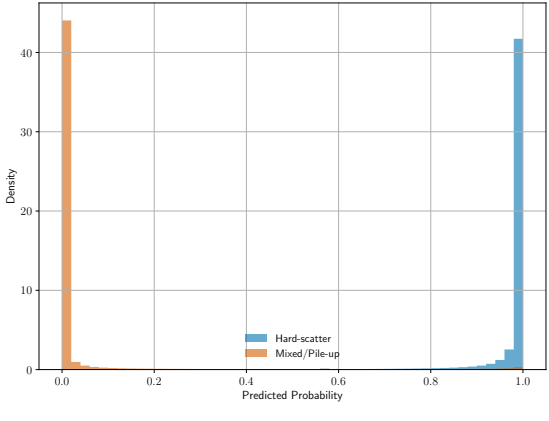


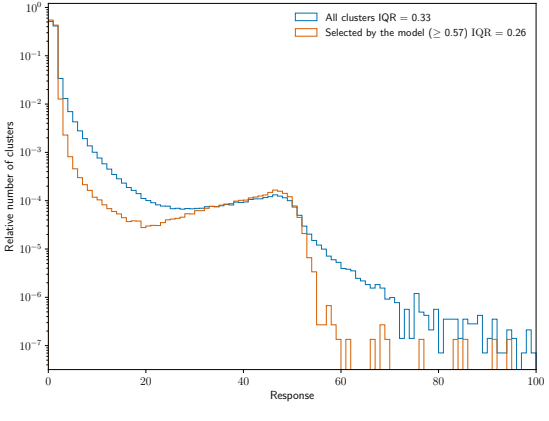
GAT Results on Run 2



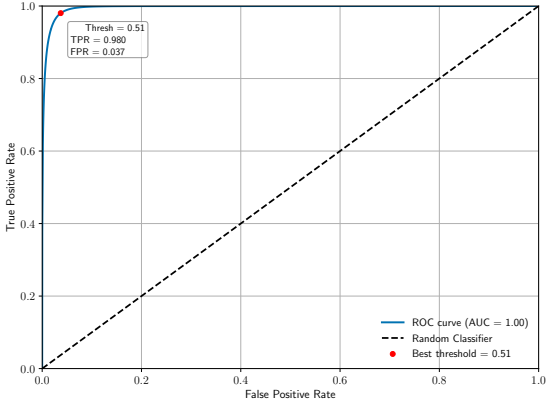


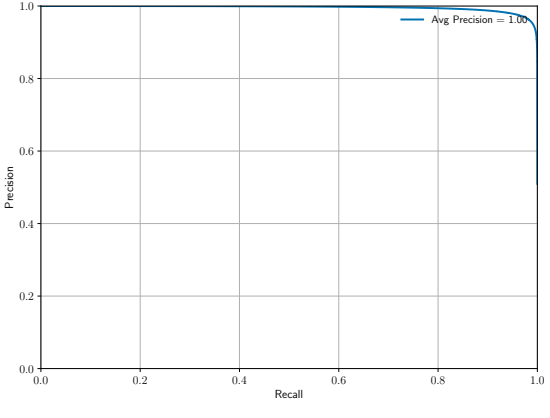
GAT Results on Run 2



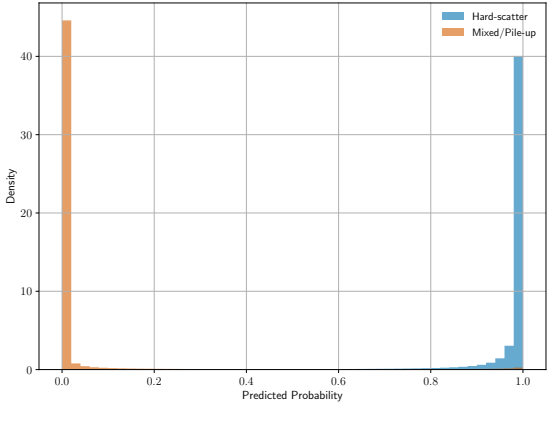


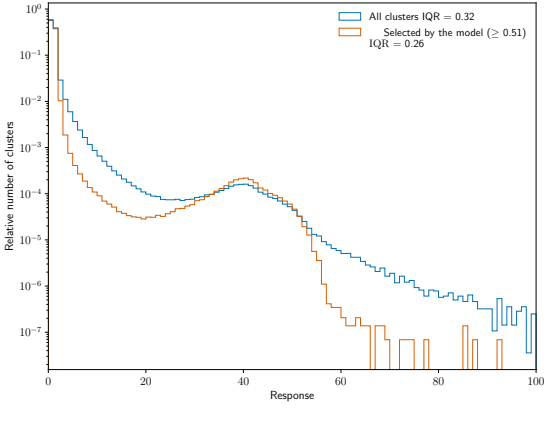
GAT Results on Run 3





GAT Results on Run 3





Summary & Outlook

- GAT performed close to perfection and thereby much better than the DNN
  - AUC of 1.00 instead of 0.96 on Run 2
  - AUC of 1.00 instead of 0.86 on Run 3
- JetDNN showed similar performance to DNN → Performance increase of GNNs is truly due to utilising the geometry
- Labelling needs improvement
  - New MC method allows us to know how much pile-up contribution is in a mixed cluster
- Check performance in full physics analysis

Backup Slides

Results on Run 2

Table: Results of DNN, JetDNN, and GAT. The models were trained on the dataset corresponding to 2018. Below the dashed line are the results for the models trained on the datasets corresponding to the years 2015 and 2016. The best performance metrics in each column are highlighted in bold.

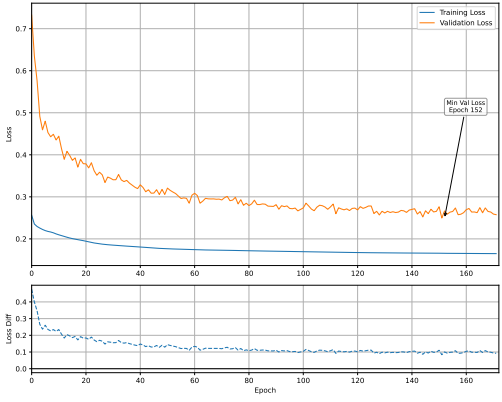
Model	AUC	Avg. Precision	Threshold	TPR	FPR	IQR Drop
DNN	0.96	0.96	0.50	0.907	0.136	**0.07**
JetDNN	0.96	0.97	0.49	0.907	0.131	**0.07**
GAT	**1.00**	**1.00**	0.57	**0.983**	**0.038**	**0.07**
DNN	0.82	0.84	0.51	0.804	0.335	0.04
JetDNN	0.83	0.86	0.51	0.802	0.316	0.03
GAT	**0.99**	**0.99**	0.44	**0.973**	**0.056**	**0.05**

Results on Run 3

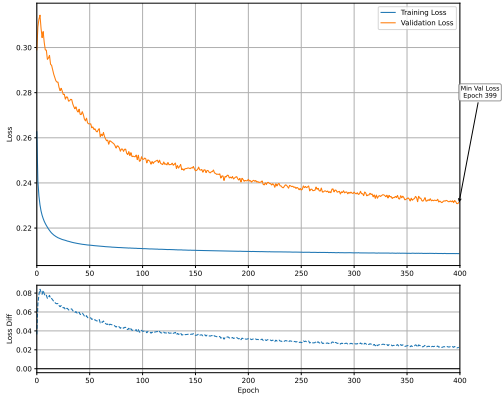
Table: Results of DNN, JetDNN, and GAT. The models were trained on the dataset corresponding to 2024. Below the dashed line are the results for the models trained on the dataset corresponding to 2022. The best performance metrics in each column are highlighted in bold.

Model	AUC	Avg. Precision	Threshold	TPR	FPR	IQR Drop
DNN	0.86	0.85	0.50	0.823	0.281	0.05
JetDNN	0.88	0.87	0.49	0.831	0.257	0.05
GAT	**1.00**	**1.00**	0.51	**0.980**	**0.037**	**0.06**
DNN	0.84	0.82	0.49	0.811	0.300	0.05
JetDNN	0.85	0.83	0.51	0.821	0.298	0.05
GAT	**0.99**	**0.99**	0.62	**0.976**	**0.044**	**0.06**

GCN Training History



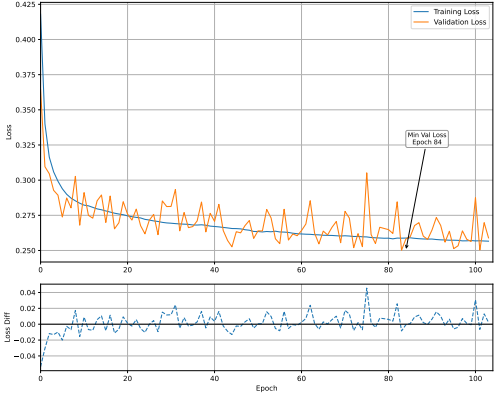
(a)



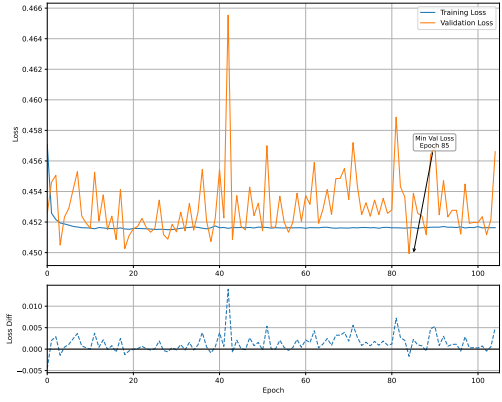
(b)

Figure: GCN training history for 2018 (left) and 2024 (right).

DNN Training History



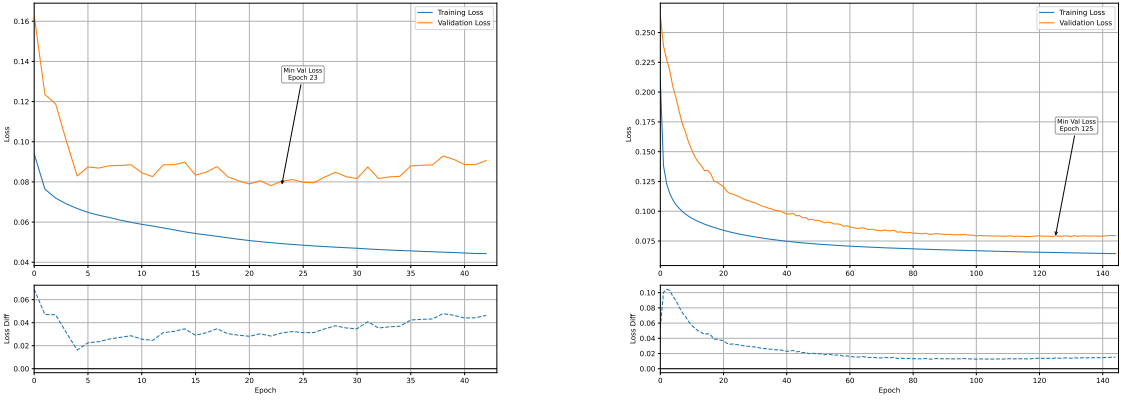
(a)



(b)

Figure: DNN training history for 2018 (left) and 2024 (right).

GAT Training History



(a) (b)

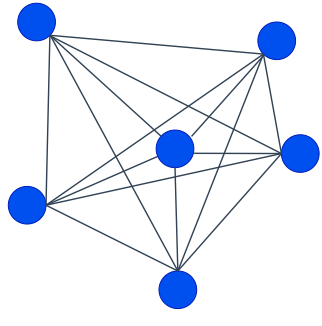
Figure: GAT training history for 2018 (left) and 2024 (right).

GAT Architecture

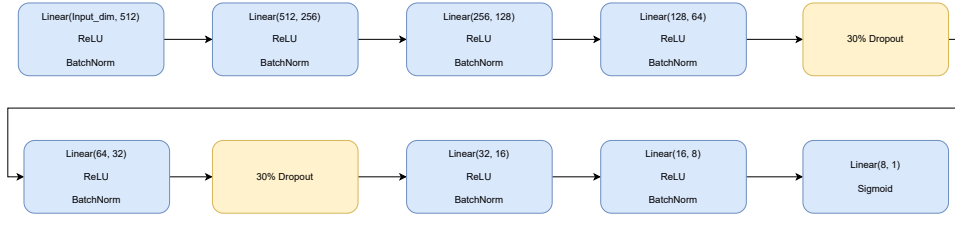
- Two GAT layers
  - Learn attention coefficients for the neighbouring nodes
  - Multi-head attention (8 heads)
  - LayerNorm → ReLU after each attention layer
  - Output dims: *1st* GAT: $16 \times 8 = 128$, *2nd* GAT: $32 \times 8 = 256$
- Three parallel branches
  - Raw node features: MLP $32 \rightarrow 32$
  - After 1st GAT: MLP $128 \rightarrow 64 \rightarrow 64$
  - After 2nd GAT: MLP $256 \rightarrow 128 \rightarrow 64$
- Fusion Head
  - Concatenate branch outputs $\Rightarrow 32 + 64 + 64 = \mathbf{160}$
  - Classifier MLP: $160 \rightarrow 128 \rightarrow 64 \rightarrow 1$
  - Sigmoid output (per-cluster probability)

Graph Building

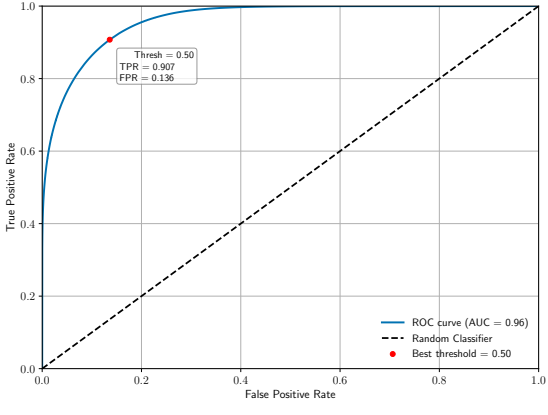
- Nodes:
  - Features used for DNN
  - Jet features to capture geometry
- Edges:
  - Fully connected graphs have $n(n-1)/2$ edges for the number of nodes n
  - They need to be uni-directional $\Rightarrow n(n-1)$ edges
  - Memory and computing-intensive compared to the previous dataset

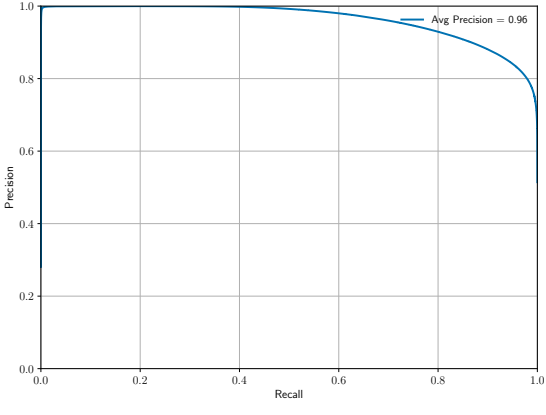


DNN Architecture

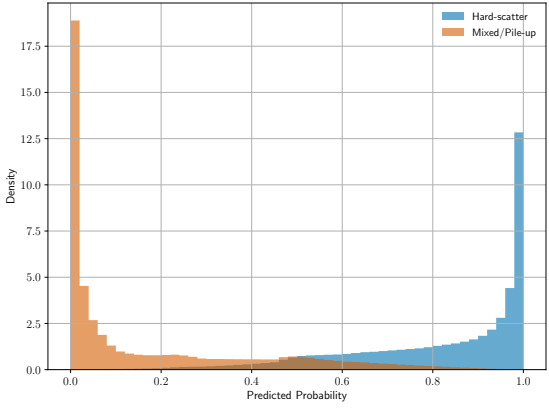


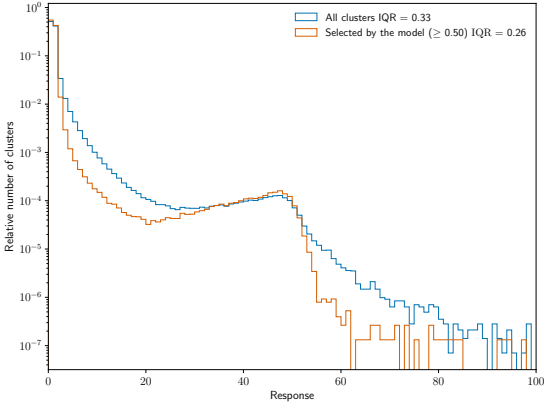
DNN Results on Run 2



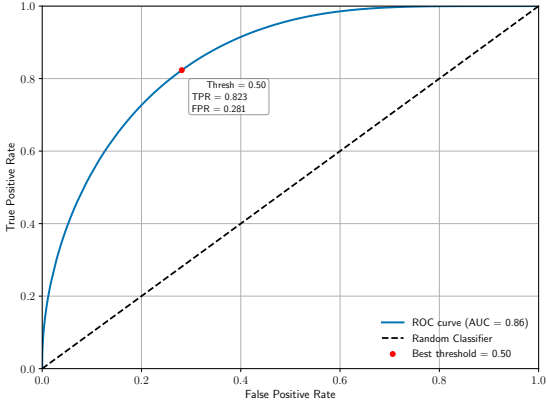


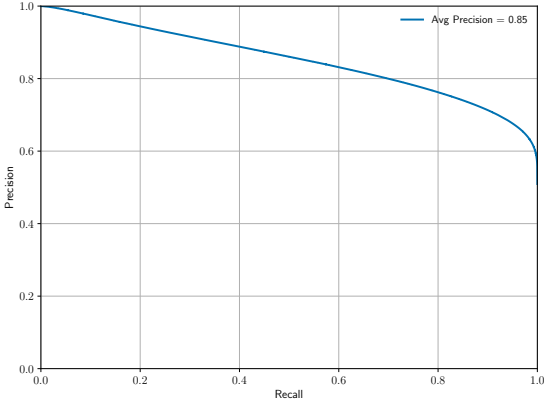
DNN Results on Run 2



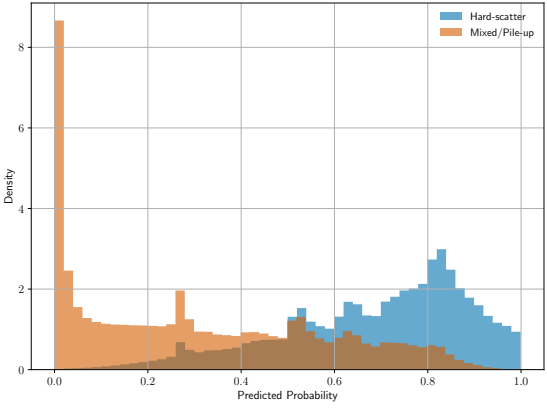


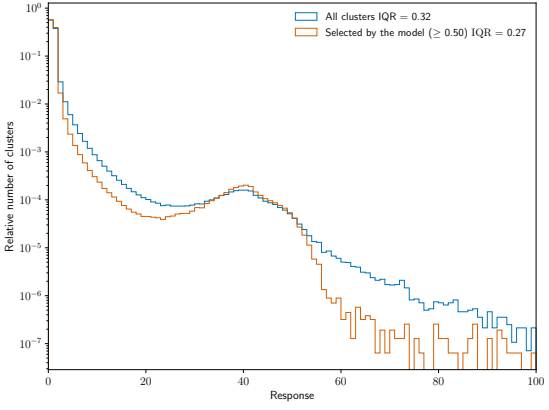
DNN Results on Run 3



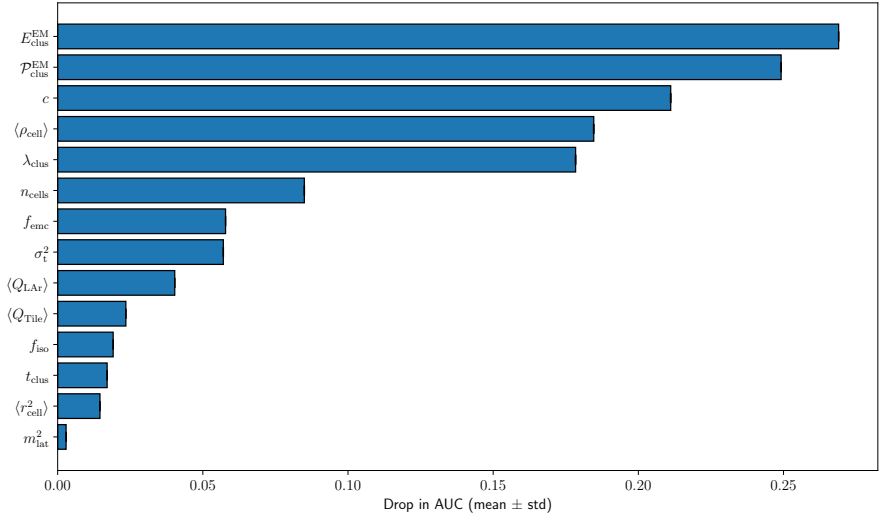


DNN Results on Run 3

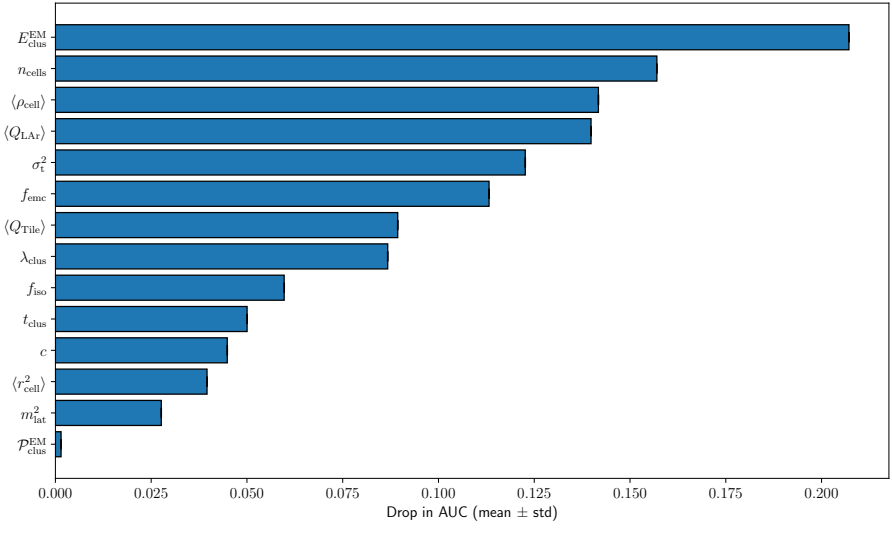




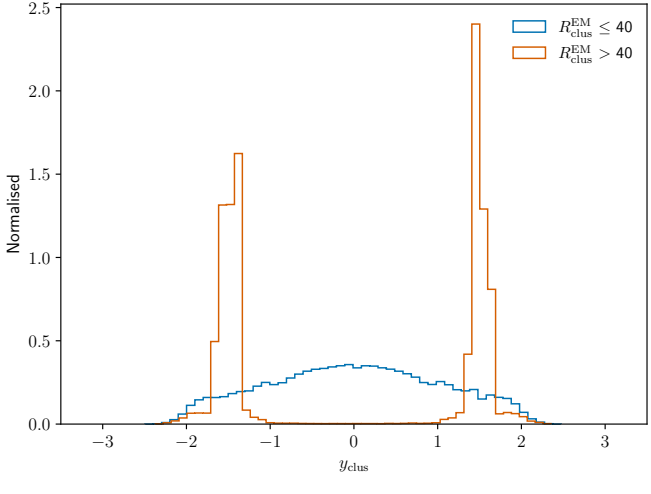
DNN Feature Importance on Run 2



DNN Feature Importance on Run 3



Location of High Responses in the Detector



Cluster Response in Detail

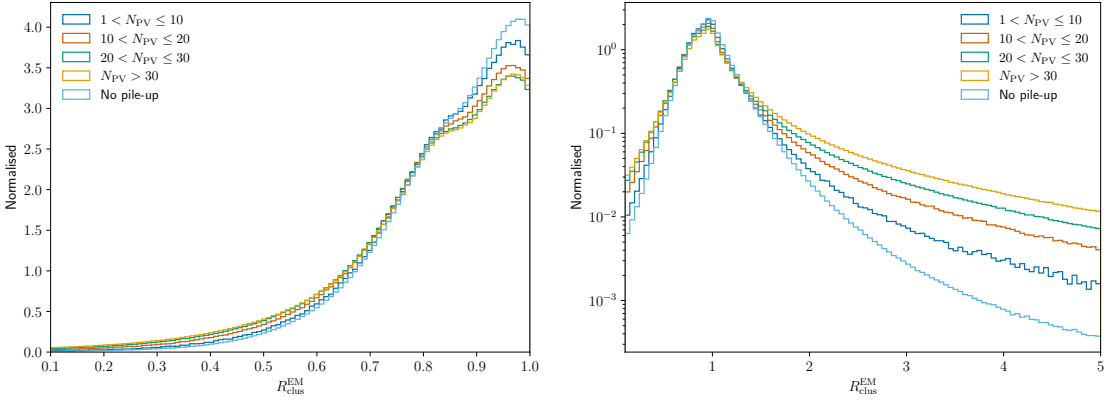
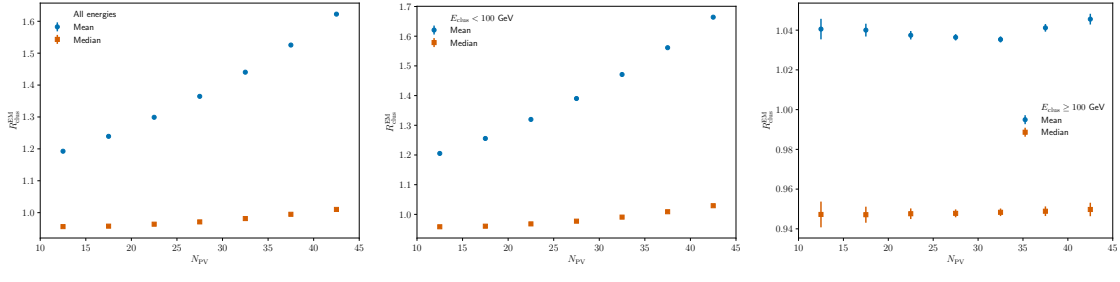


Figure: Cluster response R^{clus}_{EM} distribution between **0.1** and **1.0** (left) and between **0.1** and **5.0** (right).

Cluster Response in N_{PV} Bins



(a) (b) (c)

Figure: Mean and median cluster response for different N_{PV} bins for a) all clusters, b) clusters with $E_{clus} < 100$ GeV, and c) clusters with $E_{clus} \geq 100$ GeV.

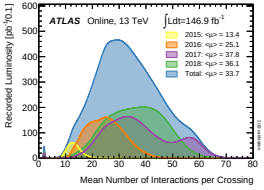
Performance Sensitivity of the Cluster Energy and Transition Region

Table: Performance of the DNN trained on the 2024 Run 3 dataset, evaluated in different cluster energy bins and with the transition region excluded. The best performance metrics in each column are highlighted in bold.

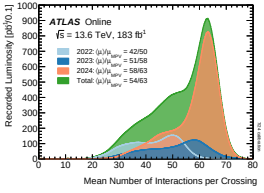
Bin	AUC	Avg. Precision	Thr.	TPR	FPR	IQR Drop
Inclusive	0.84	0.82	0.49	0.811	0.300	**0.05**
$E \in [0, 40]$ GeV	**0.86**	**0.86**	0.50	**0.831**	**0.276**	**0.05**
$E \in (40, 100]$ GeV	0.81	0.81	0.49	0.757	0.313	0.00
$E \in (100, 300]$ GeV	0.77	0.77	0.50	0.729	0.337	0.01
Excl. $1.37 \leq \lvert\eta\rvert \leq 1.52$	**0.86**	0.85	0.50	0.819	0.277	0.04

Data

- Monte Carlo campaigns for Run 2 and Run 3 of ATLAS
- Dijet events produced with Pythia 8
- NNPDF2.3 LO PDF
- Lund hadronisation model
- A14 tune



(a) Run 2: 2015–2018



(b) Run 3: 2022–2024

Figure: ATLAS Public Luminosity Results for Run 2 and Run 3.

Input Features

Feature name	Description
n_{cells}	Number of cells in cluster
$E_{\text{clus}}^{\text{EM}}$	Cluster signal at the electromagnetic scale
t_{clus}	Cluster timing
σ_t^2	Second moment of the cell timing distribution
$P_{\text{clus}}^{\text{EM}}$	Probability for the cluster to be from an electromagnetic shower
f_{emc}	Fraction of energy in the electromagnetic calorimeter
$\langle \rho_{\text{cell}} \rangle$	Energy-weighted first moment of cell signal density
c	Distance of center-of-gravity from nominal vertex
λ_{clus}	Distance from calorimeter front face
m_{lat}^2	Energy dispersion perpendicular to cluster axis
m_{long}^2	Energy dispersion parallel to the cluster axis
f_{iso}	Measure of cluster isolation
$\langle Q_{\text{LAr}} \rangle$	Average signal quality in the LAr calorimeter
$\langle Q_{\text{Tile}} \rangle$	Average signal quality in the Tile calorimeter

Input Features

Feature name	Description
$\max\{\zeta_{\text{clus}}^{\text{EM}}\}$	Highest cell signal significance in cluster
y_{clus}	Cluster rapidity
ϕ_{clus}	Cluster azimuth
$\Delta\alpha$	Angular distance between the principal shower axis w.r.t. to its direction from the nominal vertex
$\Delta\theta$	Polar distance between the principal shower axis w.r.t. to its direction from the nominal vertex
$\Delta\phi$	Azimuthal distance between the principal shower axis w.r.t. to its direction from the nominal vertex
$\langle r_{\text{cell}}^2 \rangle$	Second moment of the radial distances of the cells to the principal cluster axis
$\langle \lambda_{\text{cell}}^2 \rangle$	Second moment of the distances of the cells from the cluster centre along its main axis

Jet Features

- Transverse momentum fraction z_T
  - Ratio of transverse momentum w.r.t. the beamline
- Longitudinal momentum projection z_L
  - $z_L \approx 1$ →cluster is strongly aligned with jet
  - $z_L \approx 0$ →cluster is mostly orthogonal to jet
  - $z_L < 0$ →cluster is pointing away from the jet
- Relative transverse momentum z_{Rel}
  - Measures how off-axis the clusters are with w.r.t. the jet cone

$$z_T = \frac{p_T^{\text{cluster}}}{p_T^{\text{jet}}}$$

$$z_L = \frac{\vec{p}_{\text{cluster}} \cdot \vec{p}_{\text{jet}}}{\left|\vec{p}_{\text{jet}}\right|^2}$$

$$z_{\text{Rel}} = \frac{\left|\vec{p}_{\text{cluster}} \times \vec{p}_{\text{jet}}\right|}{\left|\vec{p}_{\text{jet}}\right|^2}$$

$$\Delta\eta = \eta_{\text{cluster}} - \eta_{\text{jet}}$$

$$E_{\text{jet}}^{\text{const}}$$

Jet Input Features

Feature name	Description
$E_{\text{jet}}^{\text{const}}$	Jet energy at constituent scale
z_{T}	Ratio of the transverse momentum of the cluster and jet w.r.t. the beamline
z_{L}	Longitudinal momentum projection of the cluster on the jet
z_{Rel}	Relative transverse momentum of the cluster w.r.t to the jet
$\Delta\eta$	Difference of η_{cluster} and η_{jet}

Cuts

Feature	Cut
$E_{\text{clus}}^{\text{dep}}$	> 300 MeV
$E_{\text{clus}}^{\text{EM}}$	> 0
λ_{clus}	> 0
$\langle \rho_{\text{cell}} \rangle$	> 0
σ_{t}^2	> 0
$\zeta_{\text{clus}}^{\text{EM}}$	> 0
$\langle \mu \rangle$	> 20
$E_{\text{jet}}^{\text{const}}$	> 0
$p_{\text{T}}^{\text{jet}}$	> 0

Preprocessing

- Some features were removed because of missing pile-up sensitivity
- $v(y) = \frac{\langle y \rangle - y}{std(y)}$ standard scaling with $y = x$ or $y = log_{10}(x)$
- Cube root for t_{clus}

$$y(t_{clus}) = \begin{cases} -\sqrt[3]{|t_{clus}|} & \text{if } t_{clus} < 0 \\ \sqrt[3]{t_{clus}} & \text{otherwise} \end{cases}$$

Feature name	Normalisation method
n_{cells}	Standard
E^{EM}_{clus}	Logarithmic
t_{clus}	Cube root, then standardised
σ^2_t	Logarithmic
P^{EM}_{clus}	Logarithmic
f_{emc}	Standard
$\langle \rho_{cell} \rangle$	Logarithmic
c	Logarithmic
λ_{clus}	Logarithmic
m^2_{lat}	Standard
m^2_{long}	Standard
f_{iso}	Standard
$\langle Q_{LAr} \rangle$	Logarithmic

Preprocessing

- Stratified **60 %** train, **20 %** validation, and **20 %** test data split
- Graph labels by majority vote
- Mean and standard deviations are computed on the train dataset and applied to the validation and test datasets

Feature name	Normalisation method
$\langle Q_{\text{Tile}} \rangle$	Logarithmic
$E_{\text{jet}}^{\text{const}}$	Logarithmic
z_{T}, $z_{\text{L}'}$, z_{Rel}	$p_{\text{T}}^{\text{clus}}$ and $p_{\text{T}}^{\text{jet}}$ standardised
$\Delta\eta$	η_{clus} and η_{jet} standardised

Cluster Energy Calibration

1. **L**ocal Hadronic **C**ell **W**eighting (**LCW**) Calibration
   - $w_{\text{cell}}^{\text{cal}} = P_{\text{cluster}}^{EM} \cdot w_{\text{cell}}^{\text{em-cal}} + (1 - P_{clus}^{EM}) \cdot w_{\text{cell}}^{\text{had-cal}}$
2. Machine Learning Based Calibration
   - Calibration networks target cluster response
   - $R_{\text{clus}}^{\text{EM}} = \frac{E_{\text{clus}}^{\text{EM}}}{E^{\text{dep}}}$

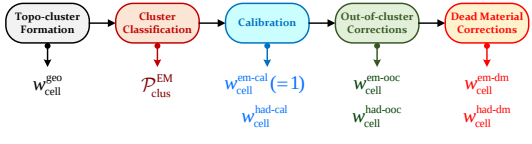


Figure: DOI: 10.1140/epjc/s10052-017-5004-5