A Neural-Network Clusterisation Algorithm for the ATLAS Silicon Pixel Detector

K.J.C. Leney on behalf of the ATLAS Collaboration
University of the Witwatersrand, Johannesburg, South Africa
E-mail: katharine.leney@cern.ch

Abstract. A novel technique using a set of artificial neural networks to identify and split merged measurements created by multiple charged particles in the ATLAS pixel detector is presented. Such merged measurements are a common feature of boosted physics objects such as tau leptons or strongly energetic jets where particles are highly collimated. The neural networks are trained using Monte Carlo samples produced with a detailed detector simulation. The performance of the splitting technique is quantified using LHC data collected by the ATLAS detector and Monte Carlo simulation. The number of shared hits per track is significantly reduced, particularly in boosted systems, which increases the reconstruction efficiency and quality. The improved position and error estimates of the measurements lead to a sizable improvement of the track and vertex resolution.

1. The ATLAS Detector

The ATLAS detector [1] is a multi-purpose detector with a forward-backward symmetric cylindrical geometry and nearly 4π coverage in solid angle. ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) and the z-axis along the beam pipe. The x-axis points from the IP to the centre of the LHC ring, and the y-axis points upward. Cylindrical coordinates (r, ϕ) are used in the transverse (x, y) plane, with ϕ being the azimuthal angle around the beam pipe and r being the distance to the z-axis. The three major sub-components of ATLAS are the tracking detector, the calorimeter and the muon spectrometer.

2. The ATLAS Pixel Detector

The silicon pixel detector [2] is the innermost layer of the tracking detector and provides precision measurements of the positions of charged particles. It is crucial for the reconstruction of primary vertices which give an estimate of the number of proton-proton interactions per bunch crossing, and for the identification of long-lived particles via the reconstruction of secondary vertices. The detector consists of 1744 identical modules, each containing 46080 silicon pixel sensors. Pixels are 250 μm thick, and nominally measure 50 μm $(r-\phi) \times 400$ μm (z), although there is a small fraction of longer pixels. A total of 80.8 million read-out channels are provided.

3. Pixel Clustering

Particles traversing the detector typically deposit charge in more than one pixel. The amount of charge deposited is estimated using the time-over-threshold method. Clusters are formed by

grouping together pixels with deposited charge that have a common edge or a common corner, as illustrated in figure 1 (a). Starting from the geometrical centre of the cluster, defined as (x_{centre}, y_{centre}) in the local reference frame $x - y$ of the sensor surface, the position of the particle at the point where it crosses the detector layer is computed from the signal heights inside the cluster of pixels:

$$x_{cs} = x_{\text{centre}} + \Delta_x \cdot \left(\Omega_x - \frac{1}{2}\right)$$

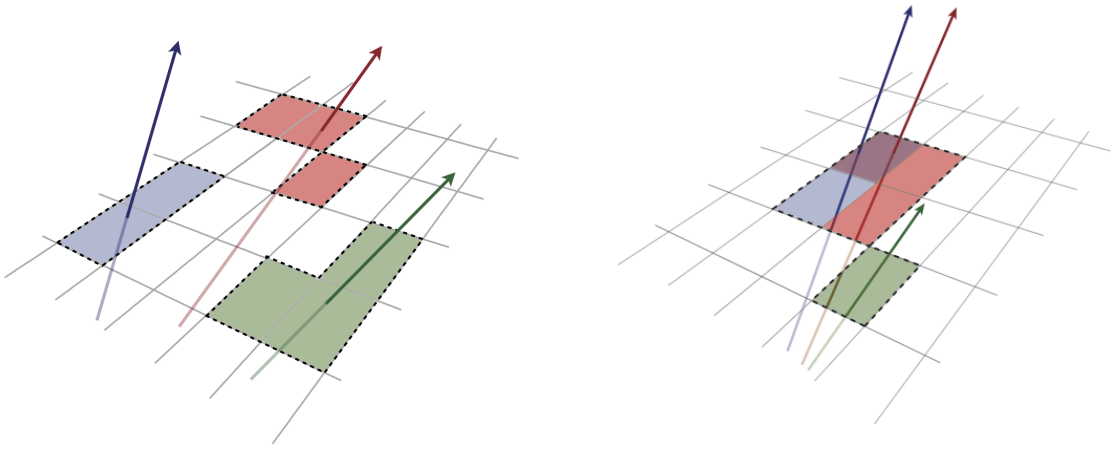
$$y_{cs} = y_{\text{centre}} + \Delta_y \cdot \left(\Omega_y - \frac{1}{2}\right),$$

where the parameters $\Omega_{x(y)}$ are defined as:

$$\Omega_{x(y)} = \frac{q_{\text{last row (column)}}}{q_{\text{first row (column)}} + q_{\text{last row (column)}}},$$

and q is the deposited charge. The variables Δx and Δy are calibrated from data, using the linear dependence of the distance between the extrapolated track and the cluster centre on the $\Omega_{x(y)}$ parameters. They are parametrised as a function of the incidence angle in ϕ (θ) of the track with respect to the pixel module and the number of pixels within the clusters in the x (y) axis direction.

Although the standard clustering provides excellent resolution for most clusters, it is inadequate for dense environments with multiple charged particles where charge is deposited in neighbouring pixels and clusters are shared between particles, as is illustrated in figure 1 (b). Objects such as highly energetic jets and multi-prong hadronic decays of tau leptons are most affected. Jets with transverse momentum > 1 TeV will typically produce merged clusters. Unless the shared clusters can be resolved between particles then the track parameters will be mis-estimated. In such cases it is therefore desirable to split the clusters between tracks.



(a) (b)

Figure 1. Pixels with deposited charge are grouped into clusters if they have a common edge or a common corner. In most cases the clusters are well separated and the resolution is excellent (a), but in dense environments with multiple charged particles, charge is deposited in neighbours and clusters are shared (b).

4. Properties of Tracks in Dense Environments

The characteristics of tracks in dense environments, such as in the core of jets, are studied as a function of jet transverse momentum ($p_{\mathrm{T}}^{\mathrm{jet}}$) and of the track distance (ΔR) from the core of the jet for $40\,\mathrm{GeV} < p_{\mathrm{T}}^{\mathrm{jet}} < 1200\,\mathrm{GeV}$ using proton-proton collisions with centre-of-mass energy $\sqrt{s} = 8$ TeV recorded with the ATLAS detector during 2012.

The sample corresponds to integrated luminosity of $20.7\,\mathrm{fb}^{-1}$. Monte Carlo samples are modelled using PYTHIA8 [3]. Jets are reconstructed from topological clusters using the anti-k_t algorithm [4] with a distance parameter $R = 0.4$ and required to have $0 < |\eta^{\,\mathrm{jet}}| < 1.2$. Events are selected using a trigger-dependent minimum $p_{\mathrm{T}}^{\mathrm{jet}}$ threshold which was imposed on all jets in the event. This threshold is chosen to ensure that the efficiency of a jet passing the trigger is $> 99\%$.

Events are required to have at least one primary vertex reconstructed using Inner Detector tracks. The vertex with the largest $\Sigma \left(p_{\mathrm{T}}^{\mathrm{track}}\right)^2$ is identified as the hard-scattering vertex. Tracks are required to pass the following selection:

- Transverse momentum (p_{T}) $> 0.5\,\mathrm{GeV}$
- One or more hits in the pixel detector.
- Six or more hits in the semi-conductor tracker.
- Transverse impact parameter (d_0) $< 1.5\,\mathrm{mm}$
- $z_0 \sin\theta < 1.5\,\mathrm{mm}$

where d_0 and z_0 are measured with respect to the hard-scattering vertex. No cut on the track χ^2 is applied. Tracks are associated with jets using a simple geometric algorithm. If the distance in $\eta - \phi$ space, $\Delta \mathrm{R}_{\mathrm{track-jet}}$ between the track and the jet is less than the distance parameter used in the jet reconstruction (R = 0.4), the track is assigned to the jet. Track parameters are evaluated at the vertex perigee and are not extrapolated to the calorimeter.

The mean number of tracks and the mean value of the Σp_{T} per $\Delta \mathrm{R}_{\mathrm{track-jet}}$ bin are shown in figure 2. Agreement between data and simulation is good for both distributions for the full range of $p_{\mathrm{T}}^{\mathrm{jet}}$ although for the higher bins of $p_{\mathrm{T}}^{\mathrm{jet}}$, jets in data are slightly broader than in simulation. Tracks become more collimated for the more energetic jets, which are observed to have more tracks in the core of the jet.

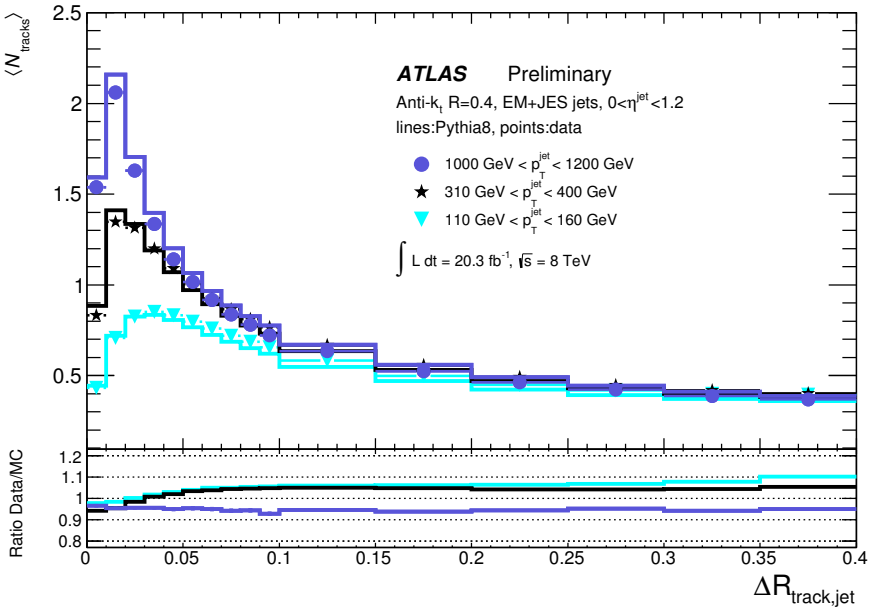
Distributions of the average number of hits in the pixel detector associated with a track and the average number of shared pixel hits are shown in figure 3. In cases where a track is extrapolated to pass through a silicon module which was not functioning, the 'dead' module is included in the hit count. Tracks in the core of the jet are observed to have fewer hits associated to them, and to share more hits with their neighbouring tracks.

5. A Neural Network Pixel Clustering Algorithm

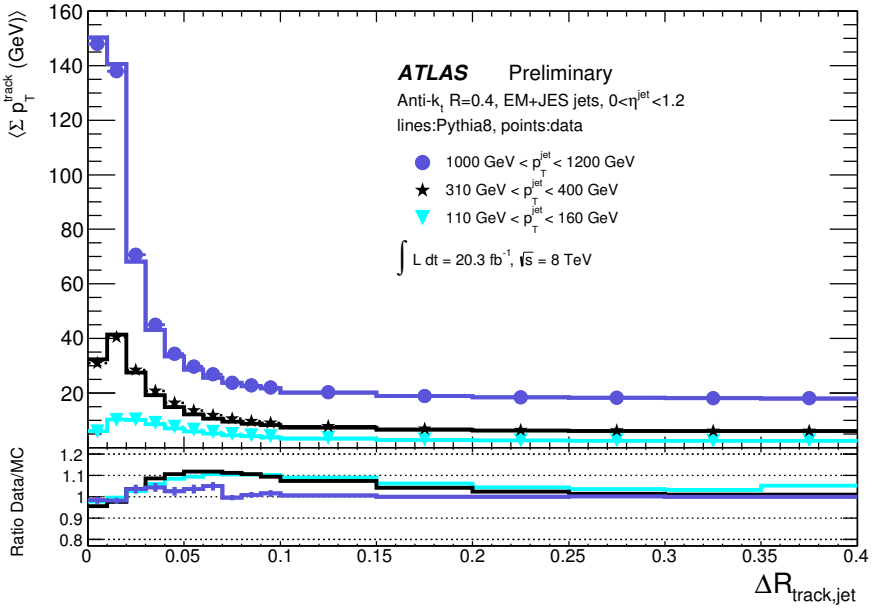
Neural networks are powerful tools for pattern recognition problems and can handle non-linear correlations between input variables. They are attractive solutions for problems with many degrees-of-freedom, and inputs can be given different weights in the hidden layers of the neural network to finally determine the output. Neural networks are a good choice for a pixel clustering algorithm since many cluster properties are almost meaningless when considered alone (e.g. the charge deposited in a single pixel), but when cluster properties are combined and put into context (e.g. once the charge deposits on adjacent pixels are known) then the information can be combined and used for successful pattern recognition.

The ATLAS pixel clustering uses feed-forward multi-layer perceptron neural networks to compute:

- The number of particles per cluster.



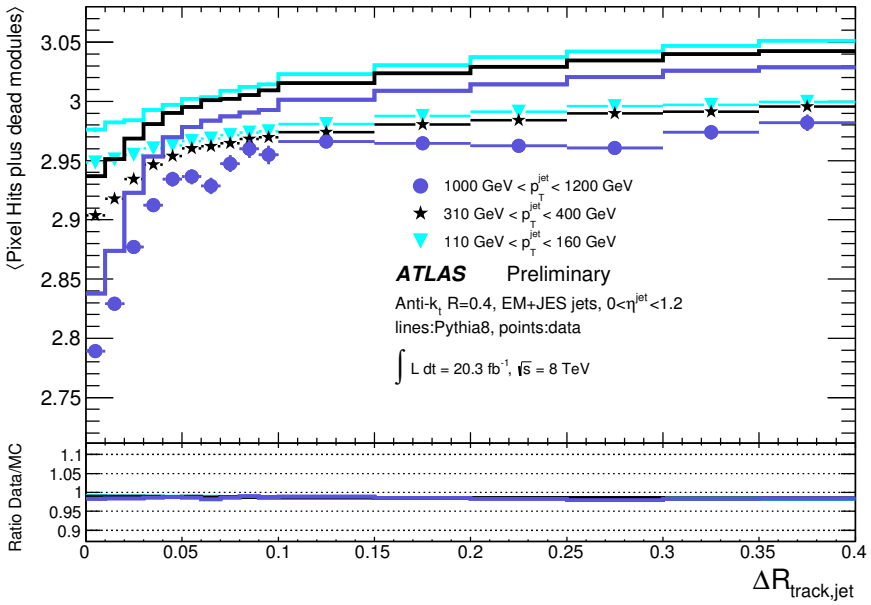
(a)



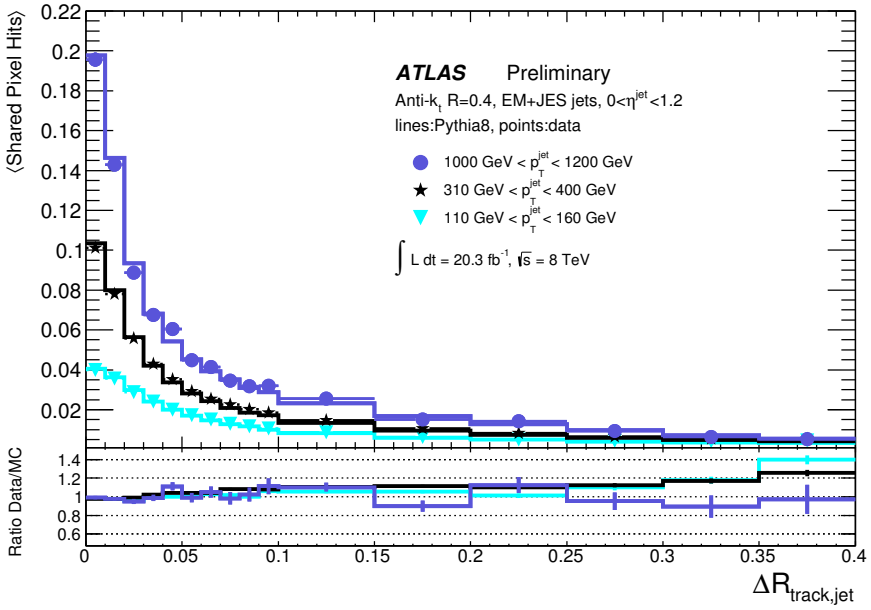
(b)

Figure 2. (a) Mean number of tracks per bin and (b) mean value of $\Sigma p_T^{\text{track}}$ per bin as a function of ΔR, the distance between the track and the jet, for bins in $p_{\text{T}}^{\text{jet}}$ for data and Monte Carlo.

- The position (and uncertainty) in two dimensions of the impact point of the particle(s) on the pixel sensor.



(a)



(b)

Figure 3. (a) Mean number of pixel hits and (b) mean number of shared pixel hits as a function of $\Delta R_{\mathrm{track-jet}}$, the distance between the track and the jet, for bins in $p_{\mathrm{T}}^{\mathrm{jet}}$ for data and Monte Carlo.

The neural network for computing the number of particles per cluster has 60 input nodes, consisting of:

- 7x7 pixel matrix of the collected charge on each pixel.
- Vector of the longitudinal sizes of the pixels in the matrix.
- Direction of the candidate charged particles traversing the cluster.

Particles are initially assumed to have been produced at the estimated centre of the luminous region of the beam-beam interaction. This is estimated using the primary vertices of collisions produced prior to the current interaction. The incidence angle of the particles with respect to the module is calculated by a straight line connecting the beam spot to the cluster position. Once track candidates have been identified, the approach is repeated using this more refined measurement of the particle direction. This is possible because the ATLAS track reconstruction strategy allows measurements to be re-calibrated after they have been assigned to a track candidate. Three output nodes are used, and can be interpreted as the posterior Bayes probabilities for a cluster to correspond to one, two or three charged particles, thus providing an event classification.

An additional set of neural networks, configured for interpolation, are used to estimate the cluster position and the probability density function for the residual of the impact point. Different neural networks are used depending on the estimated number of charged particles traversing the cluster, and trained on the true number of particles in simulation. Additional neural networks are used to calculate the residual in the transverse and longitudinal directions. All networks use the same input variables as the classification network. A total of ten neural networks are needed to assess up to three sub-clusters.

The networks are trained on simulations of pair-produced top-quarks and highly energetic di-jet events, which are divided into test samples and training samples. The number of training patterns is ensured to exceed the number of network parameters by at least a factor of 1000.

6. Performance of the Neural Network Clustering Algorithm

The performance of the splitting technique is quantified using LHC data with $\sqrt{s}$ =7 TeV collected by the ATLAS detector in 2011, and Monte Carlo simulation. Figure 4 illustrates the improved classification and position resolution achieved when using the neural network cluster splitter. The single cluster is resolved into two sub-clusters and the positions of the impact points are accurately resolved.

Figure 5 shows the mean number of pixel clusters split by the pixel cluster splitting neural network as a function of $\Delta R_{\mathrm{track-jet}}$ for different $p_{\mathrm{T}}^{\mathrm{jet}}$ bins. Clusters are split at a higher rate in the core of the jet, where the track density is higher (see figure 2 (a)). Slightly better performance is observed in simulation than data. This is likely to be due to the fact that the neural network was trained on simulation.

Tracks are allowed to share a pixel cluster with another track, only if the cluster has not already been split and the neural network output is compatible with a possible merged cluster. The most noticeable improvements are in the innermost layer of the pixel detector (the b-layer) where the particle density is highest. Figure 6 shows the number of shared b-layer hits when comparing the neural network clustering approach to the standard clustering. Ambiguities are shown to be reduced by an order of magnitude when using the neural network.

A dramatic improvement in the cluster resolution (track-to-measurement residual) in the $r\phi$-direction is observed when using the neural network clustering, as shown for 4-pixel clusters in figure 7. The non-linear treatment of charge resolution allows a single peak to be recovered in track-to-hit residuals. Improved cluster resolution leads to improved track parameters, and a 15% improvement in the longitudinal impact parameter is observed when using the neural

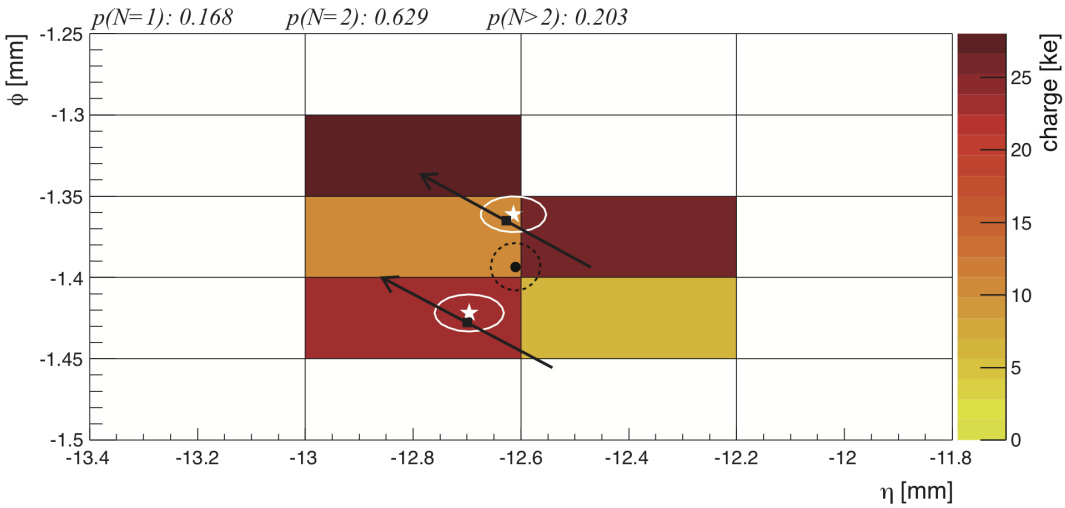


Figure 4. Example of merged clusters created by two particles in Monte Carlo simulation. The arrows show the true directions of the particles, and the square black dots their intersection with the detector layer. The black dot and the dotted black circle show the non-split cluster position and residual respectively. The white stars and circles show the estimated cluster positions and residuals after splitting. The values of $p\,(N = i)$ denote the probabilities for the cluster to be created by i particles, as estimated by the neural network.

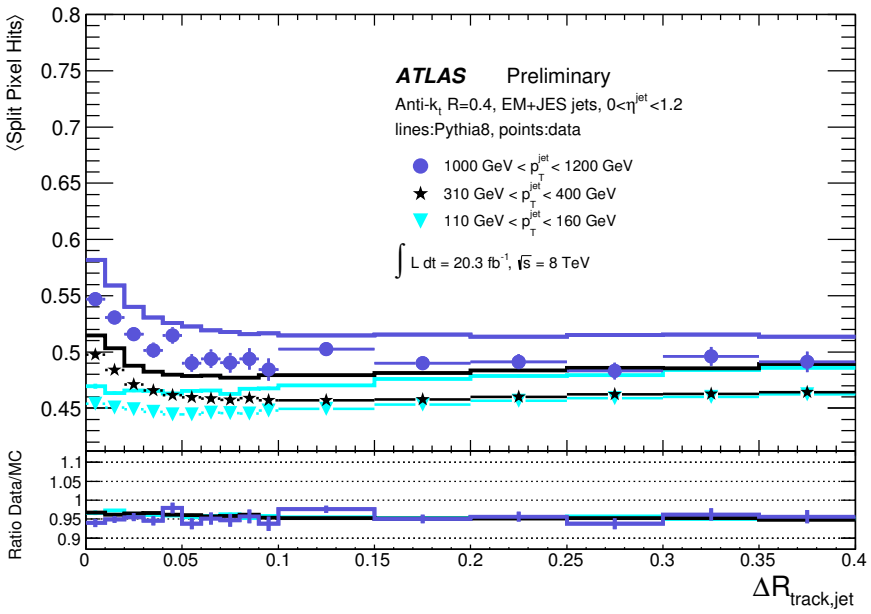


Figure 5. Mean number of pixel clusters split by the neural network pixel cluster splitting algorithm as a function of $\Delta R_{\mathrm{track-jet}}$.

network over the standard clustering. This is illustrated in figure 8. The impact parameter is a

key variable for identifying long-lived particles (e.g. heavy flavour quarks).

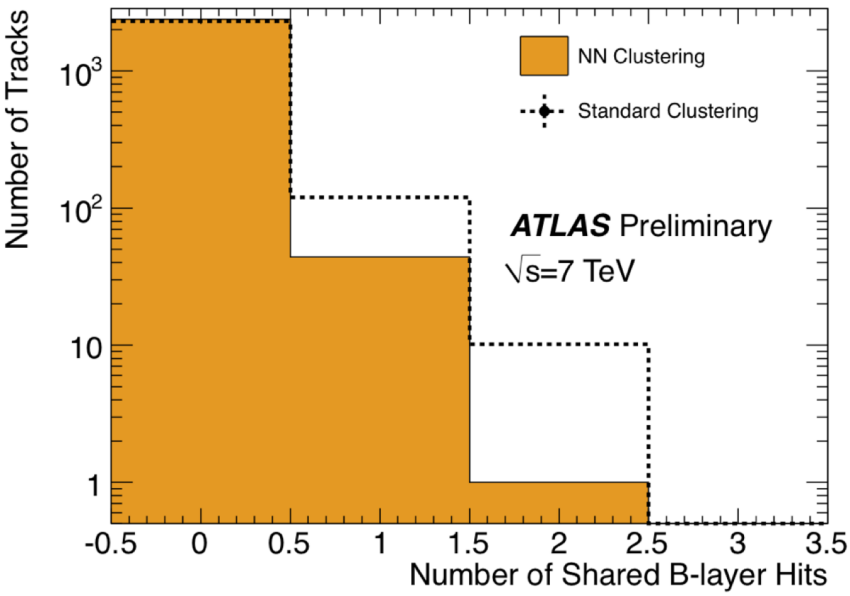


Figure 6. The number of shared hits in the innermost Pixel layer, reconstructed with and without the neural network clustering algorithm [5].

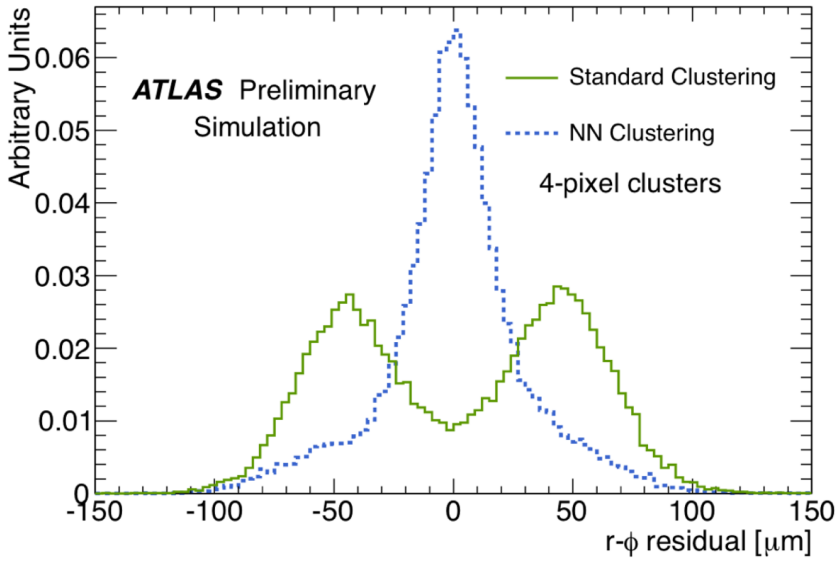


Figure 7. The cluster residual in the $r\phi$-direction for 4-pixel clusters in simulation, reconstructed with and without the neural network clustering algorithm [5].

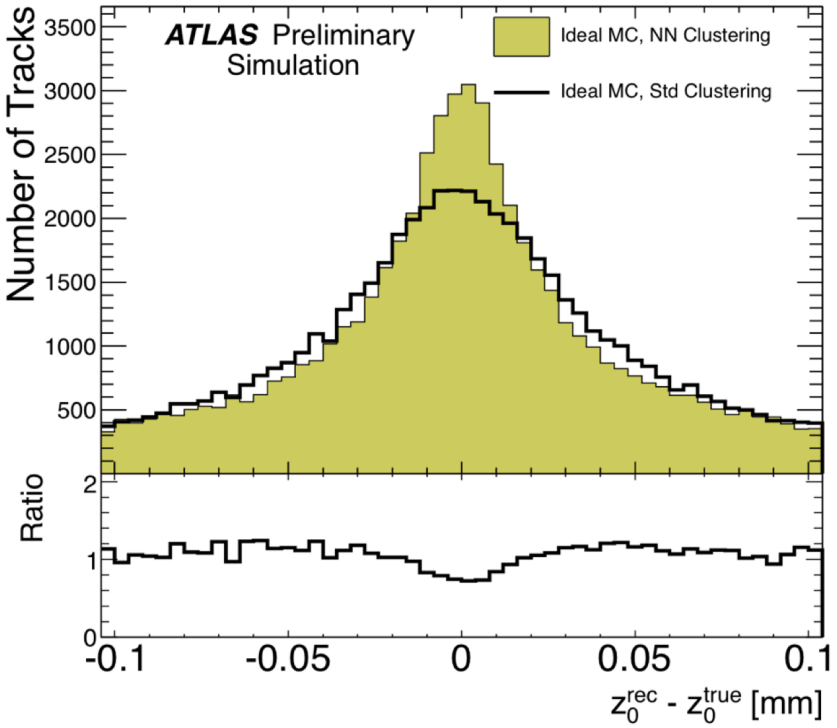


Figure 8. Longitudinal impact parameter resolution in simulation, reconstructed with and without the neural network clustering algorithm [5].

7. Conclusions

A neural network approach to clustering in the ATLAS pixel detector has been used to improve detector performance and make full use of the detector design potential. By taking into account all correlations, clusters created by multiple charged particles can be identified and split. This leads to sizeable improvements in track measurements, particularly in dense environments such as in jet cores and hadronic tau decays. The non-linear treatment of charge collection improves the impact parameter resolution, even for isolated tracks, allowing significantly improvements to the identification of long-lived particles. This improved two-particle separation will become increasingly important during future LHC upgrades, as the number of proton-proton collisions per bunch crossing (and therefore the particle density inside the detector) increases.

[1] ATLAS Collaboration *The ATLAS experiment at the CERN Large Hadron Collider* JINST **3** S08003 (2008)
[2] Aad G et al. *ATLAS pixel detector electronics and sensors* JINST **3** P07007 (2008)
[3] Sjöstrand T, Mrenna S and Skands P *A Brief Introduction to PYTHIA 8.1* CPC **178**:852-867 (2008)
[4] Cacciari M, Salam G P and Soyez G *The anti-k_t jet clustering algorithm* JHEP **4** 063 (2008)
[5] ATLAS Collaboration *Neural Network cluster reconstruction in the pixel detector* ATL-COM-PHYS-2012-476 http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/IDTRACKING/PublicPlots/ATL-COM-PHYS-2012-476.