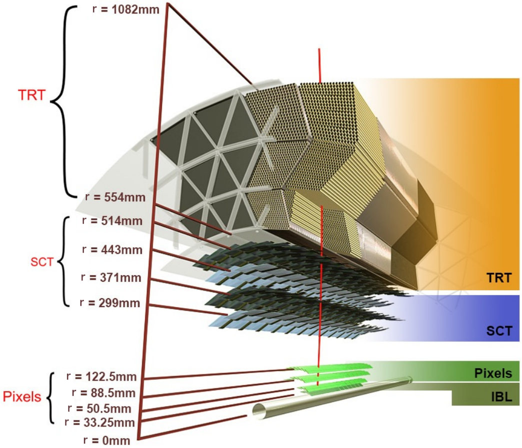
Implementation and performance of the ATLAS pixel clustering neural networks

Louis-Guillaume Gagnon, PhD student, Université de Montréal
On behalf of the ATLAS collaboration

Connecting The Dots 2018
University of Washington, Seattle

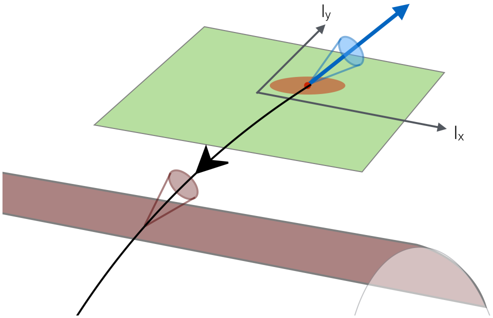
The ATLAS inner detector



- High performance tracker with 3 subdetectors:
  - TRT: straw-tube gaseous detector
  - SCT: silicon strips detector
  - Pixel+IBL: silicon pixel detector
- Pixel+IBL:
  - Up to 4 very precise measurements per track
  - Crucial for optimal tracking & vertexing performance

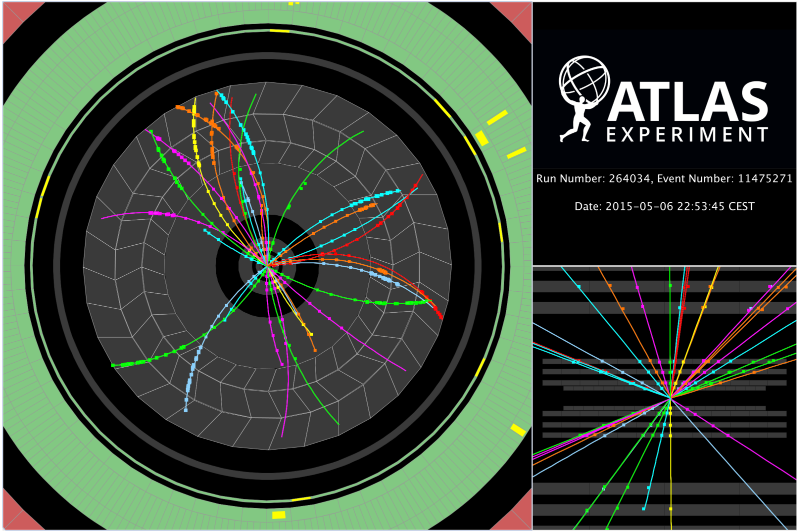
Track finding in ATLAS

- Detector measures space-points (3D coordinates of hits)
- Track seeds: sets of 3 space-points passing p_T, impact parameter cuts
- Use Kalman filter to produce track candidates
  1. Get track parameters from current candidate
  2. Compute probability distribution over position of hit on next layer
  3. Use compatible space-points to update/create track candidate
- Possibly many tracks competing for same space-points → ambiguity solving required



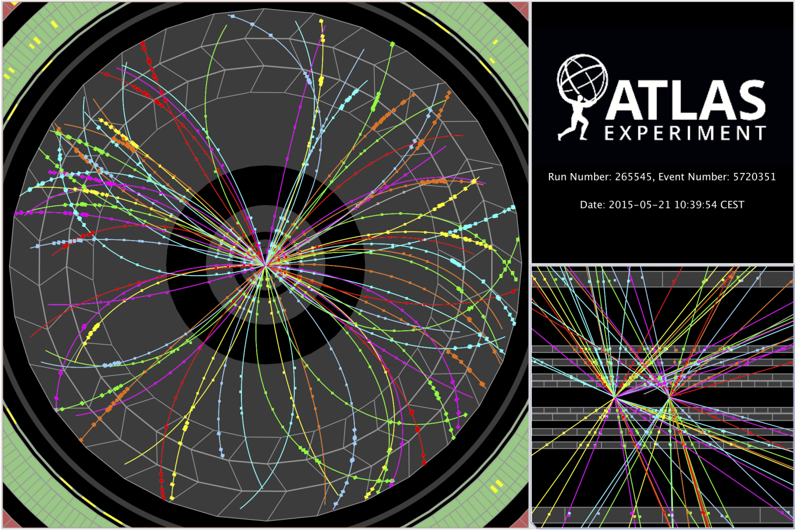
Track finding: example

- Sometimes easy . . . ($\sqrt{s} = 900$ GeV, $\langle \mu \rangle \approx 0.05$)



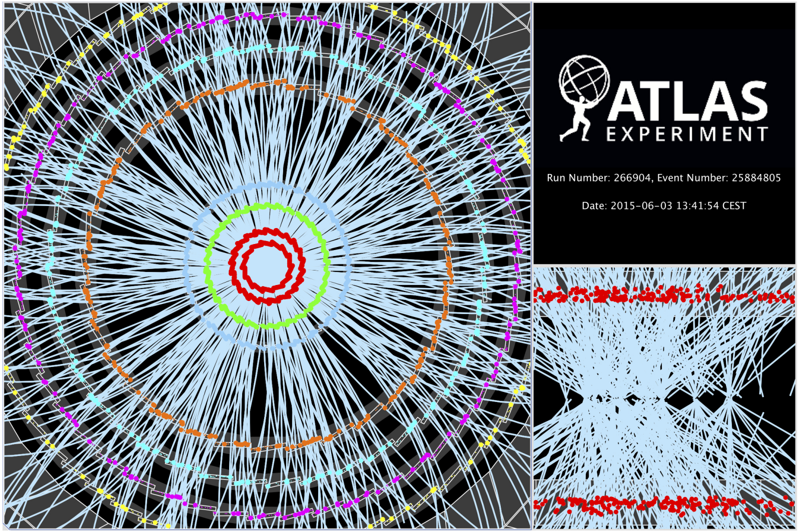
Track finding: example

- ... sometimes difficult ... ($\sqrt{s} = 13$ TeV, $\langle\mu\rangle \approx 1$)



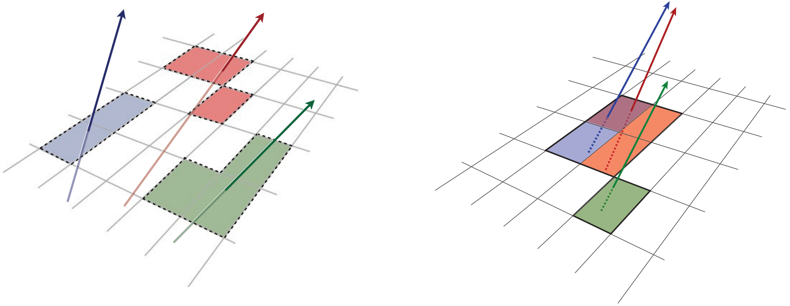
Track finding: example

- ... sometimes crazy! ($\sqrt{s} = 13$ TeV, $\langle\mu\rangle \approx 10$)



Charge clusters

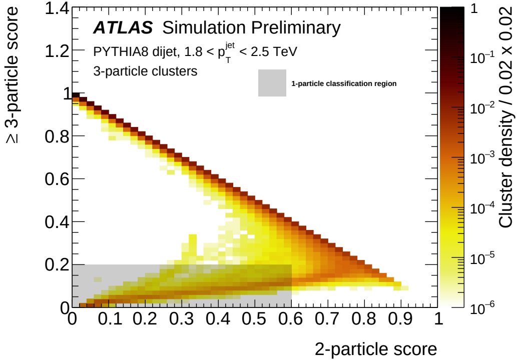
- This picture of 3D space-points processed by a Kalman filter is simplified
- In reality, energy deposits in the tracker are not point-like!
  - Sensor elements have non-infinitesimal sizes
  - Charge can be deposited in many pixels (Charge diffusion, drift due to B-field, δ-rays, ...)
- In dense environments (e.g. core of high-p_T jets), charge clusters can merge



- In ATLAS, use 3 sets of neural networks to identify clusters originating from > 1 particle and measure the hit positions and associated uncertainties

Particle multiplicity estimation

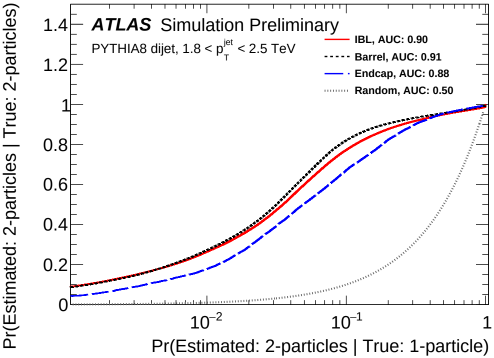
- Inputs:
  - 7x7 discretized charge matrix
  - Pixel pitches in longitudinal direction
  - Angles of incidence of track candidate
  - Detector region
- Outputs length-3 probability vector:
  - Pr(1-particle)
  - Pr(2-particle)
  - Pr($\geq$ 3-particle)
- Class assigned based on cuts on Pr(2-particle) and Pr($\geq$ 3-particle)
- When classified as $>$ 1-particle, remove ambiguity solving penalty for shared hits
- Trained on 12 million clusters from high-p_T dijet sample



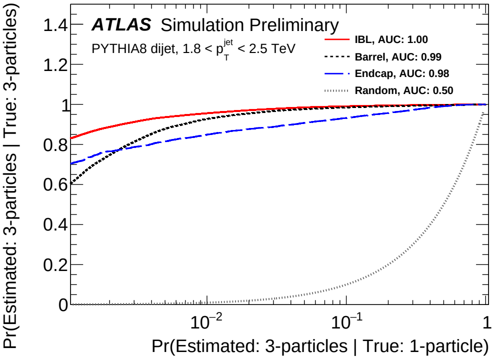
- Shaded: 1-particle classification region

Particle multiplicity estimation

- 2-particle vs 1-particle



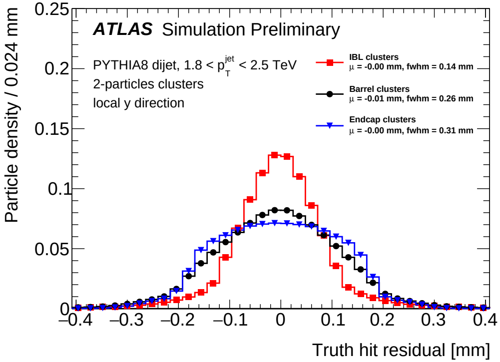
- 3-particle vs 1-particle



- WP defined by cutting on 2 probs and doesn't correspond to point on pairwise ROC curves
- Overall Pr(2 or 3 | 1-particle): IBL/Barrel $\approx 4\%$, Endcap $\approx 7\%$

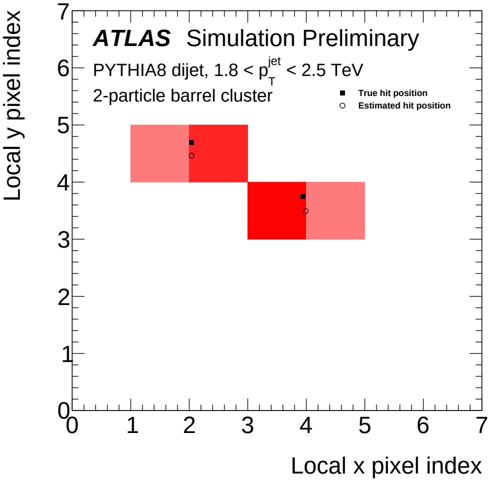
Hit position estimation

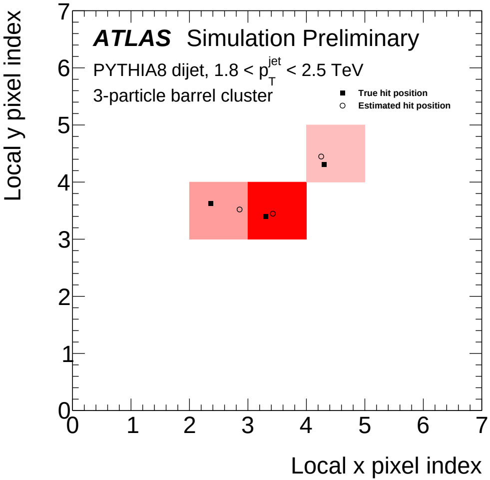
- Three networks according to multiplicity (1, 2, or 3 particles)
- Inputs: Same as multiplicity network
- Outputs: (1/2/3) 2D position vectors
  - Unit: Number of pixel lengths from charge centroid of cluster
  - Integer part: identify containing pixel
  - Fractional part: offset within the pixel
- Trained on 12 million clusters from high-p_T dijet sample



- Colors: IBL, Barrel, Endcap

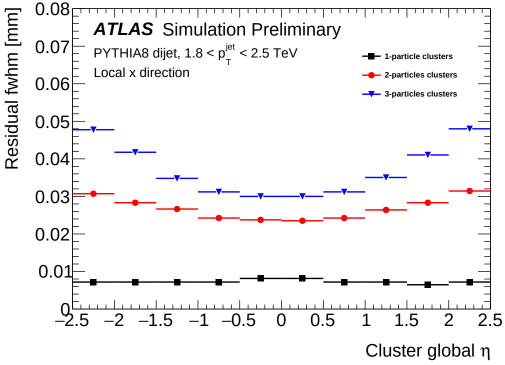
Hit position estimation



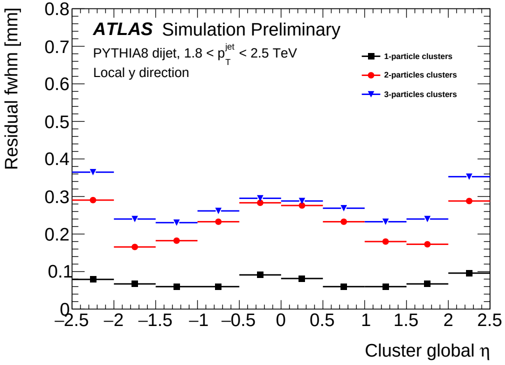


Hit position estimation

- Transverse direction

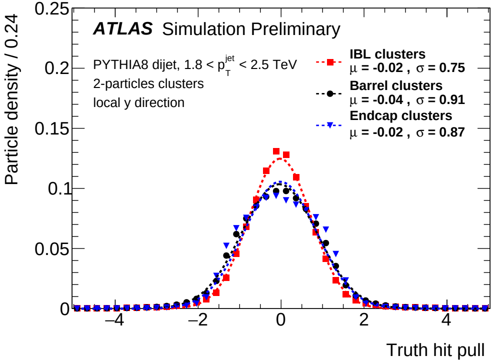


- Longitudinal direction



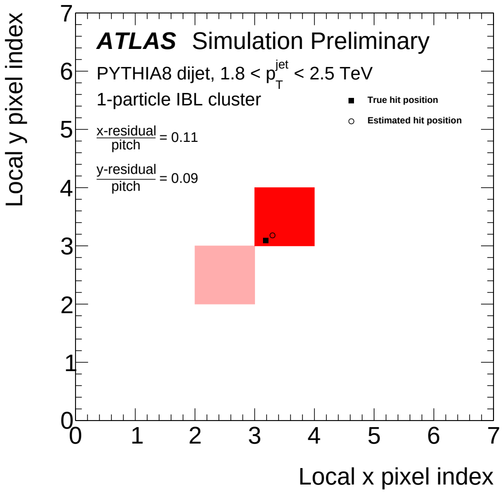
Uncertainty estimation

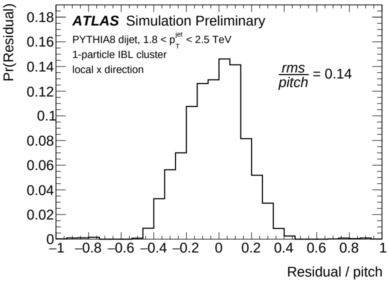
- Six networks according to multiplicity (1, 2, or 3 particles) and direction (transverse/longitudinal)
- Inputs: Same as multiplicity network + hit position estimation
- Uncertainty not known beforehand: unsupervised learning problem
- Outputs: (1/2/3) binned probability distributions over residual
- Point estimate of uncertainty: rms of distribution
- Trained on 12 million clusters from high-p_T dijet sample

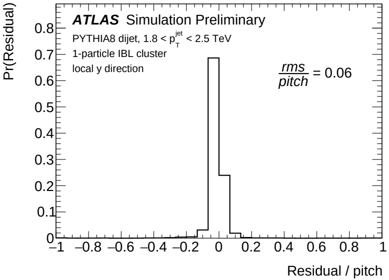


pull ≡ residual / uncertainty

Uncertainty estimation: example

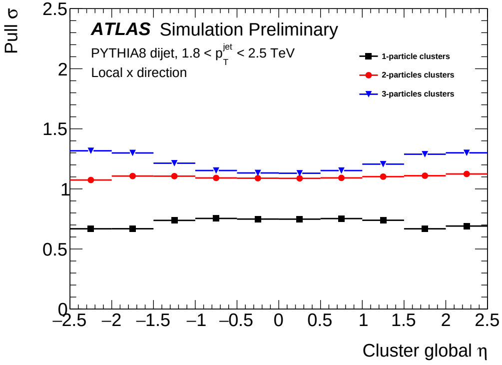




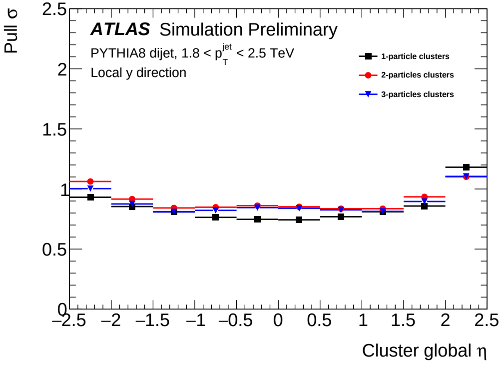


Uncertainty estimation

- Transverse direction



- Longitudinal direction



Hyperparameters

Hyperparameter	Number network	Position networks	Error networks
Structure	(60)-25-20-(3)	(60)-40-20-(2/4/6)	(62/64/66)-15-10-(30/50/60)
Hidden layers activation	Sigmoid	Sigmoid	Sigmoid
Output activation	Sigmoid	Linear	Sigmoid
Learning rate	0.08	0.04	0.3
L2 regularized	1e-7	1e-7	1e-6
Momentum	0.4	0.3	0.7
Batch size	60	30	50
Loss function	categorical crossentropy	mean squared error	categorical crossentropy

- Small networks: good tradeoff between performance and runtime

Future directions

- Inner detector will be replaced during ATLAS phase-II upgrade
  - Different geometry and pixel depth/pitches
  - Opportunity to optimize and re-think the method!
  - See Felix Cormier's talk earlier today: "*Tracking in Dense Environments for the HL-LHC ATLAS Detector*"
- Consider different network architectures
  - Idea: Mixture density networks
  - Output priors C_i, means μ_i and variances σ_i^2 of component of gaussian mixture.
  - Likelihood of a cluster:

$$P(cluster) = \sum_i C_i \mathcal{N}(true\ position; \mu_i, \sigma_i^2)$$

  - After training, use μ_i and σ_i^2 with highest C_i
  - Obtain estimation of position with uncertainty with single network instead of 3

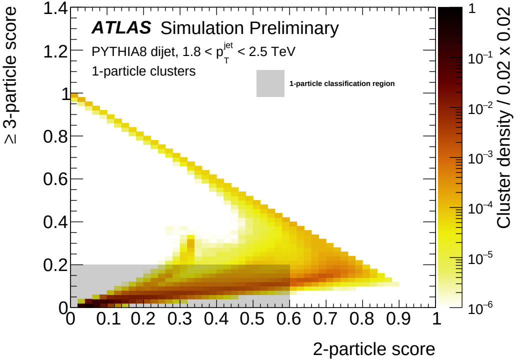
Merci!

For more details: ATL-PHYS-PUB-2018-002

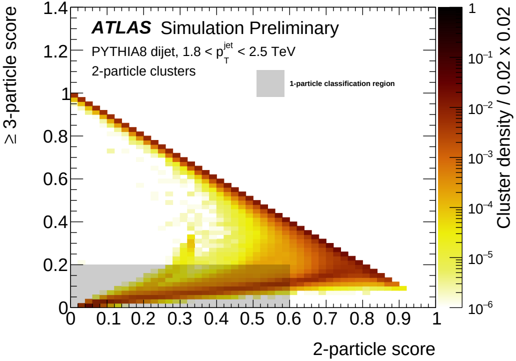
BACKUP

Particle multiplicity scores

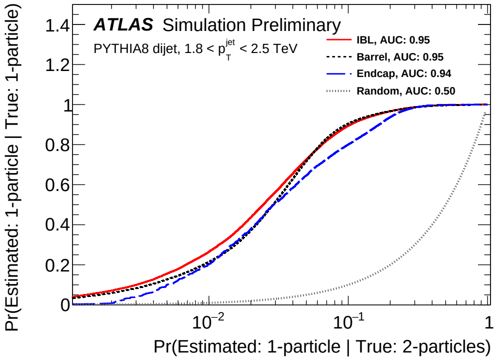
- 1-particle clusters

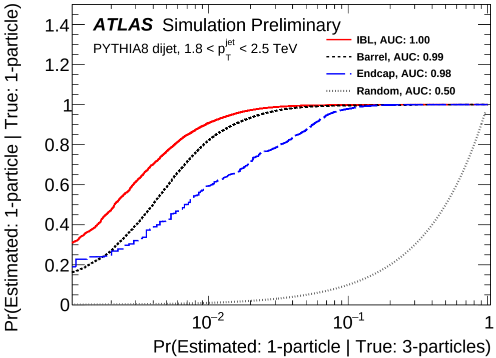


- 2-particle clusters

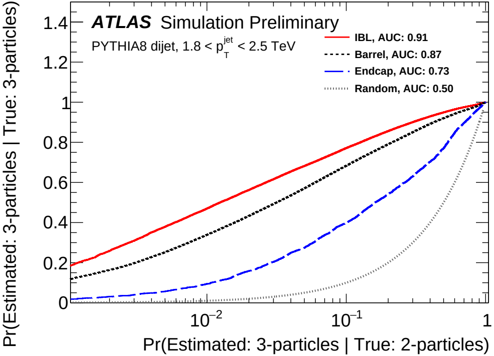


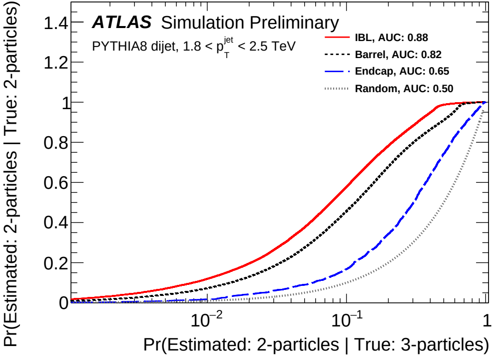
Particle multiplicity ROCs



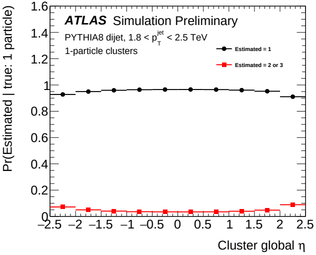


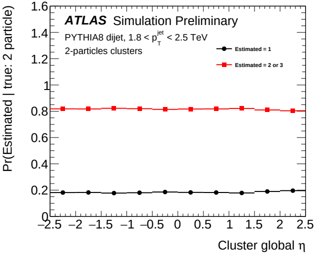
Particle multiplicity ROCs

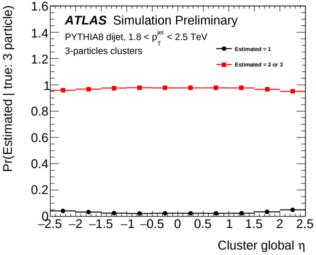




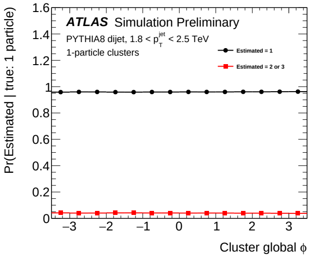
Particle multiplicity true/false positive rates

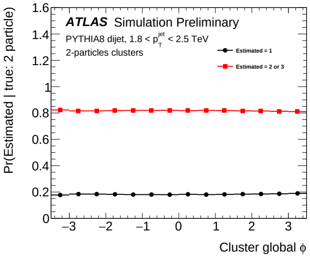


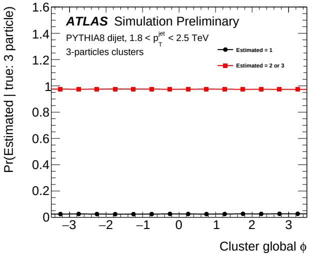




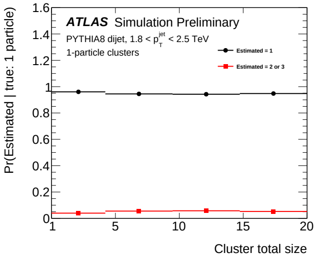
Particle multiplicity true/false positive rates

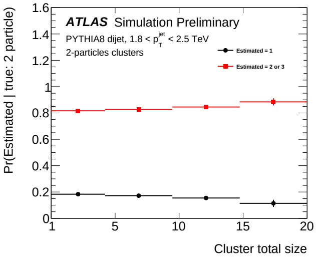


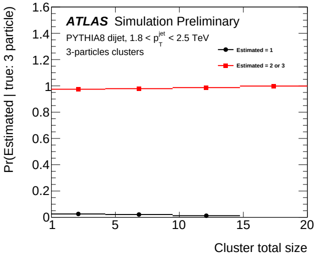




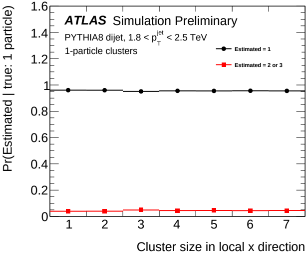
Particle multiplicity true/false positive rates

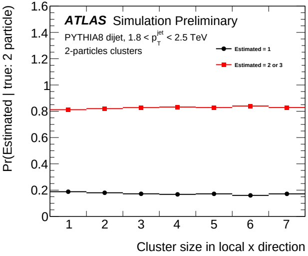


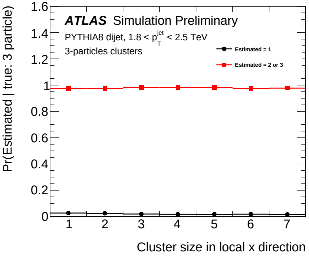




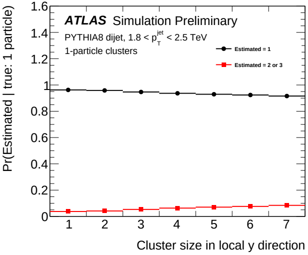
Particle multiplicity true/false positive rates

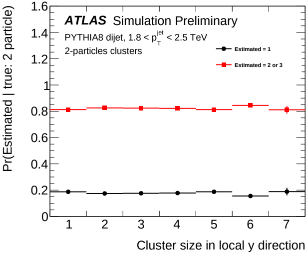


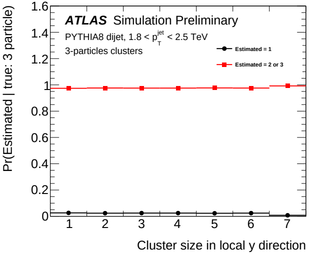




Particle multiplicity true/false positive rates

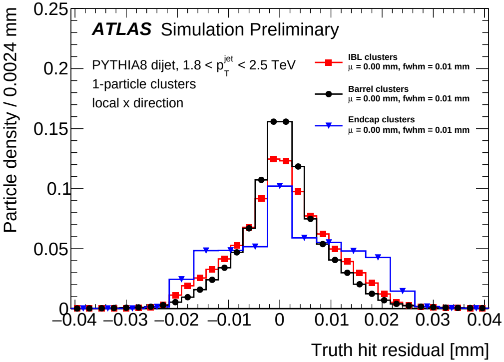




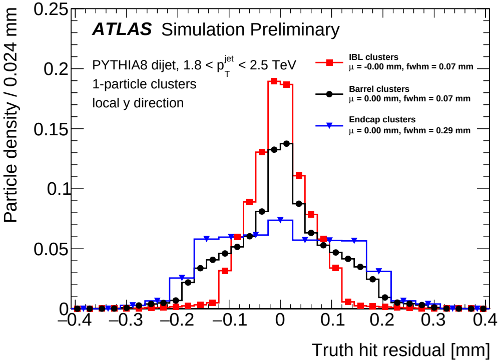


Hit position estimation

- Transverse direction

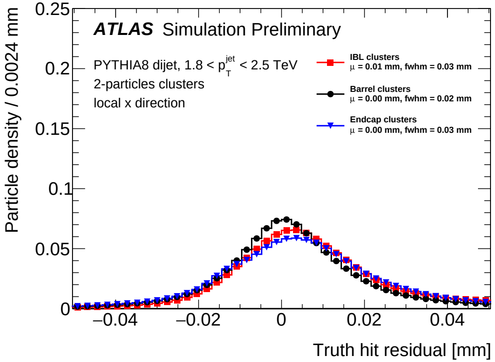


- Longitudinal direction

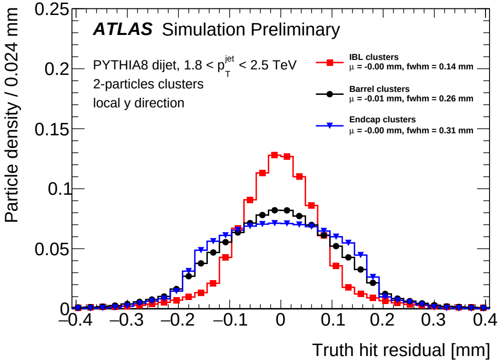


Hit position estimation

- Transverse direction

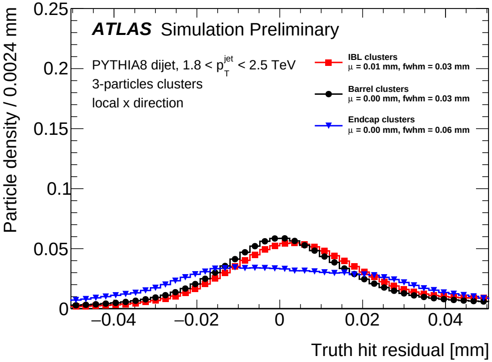


- Longitudinal direction

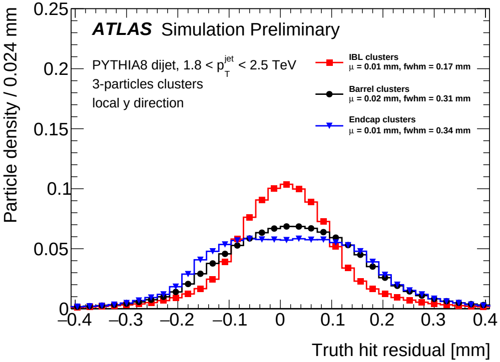


Hit position estimation

- Transverse direction

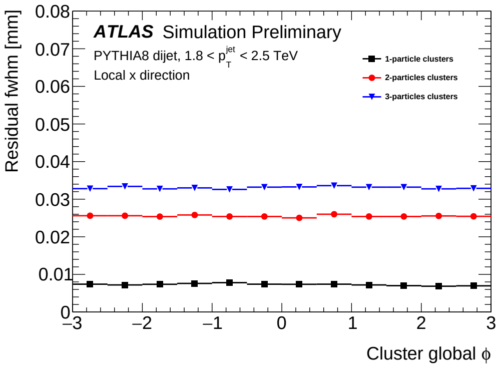


- Longitudinal direction

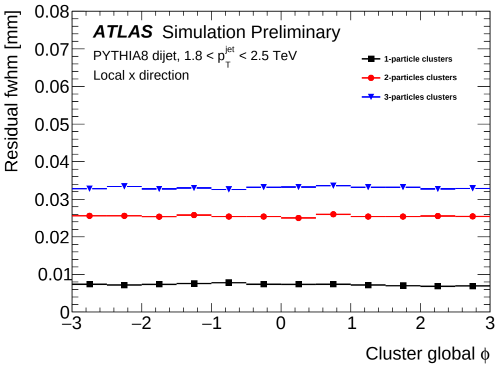


Hit position estimation

- Transverse direction

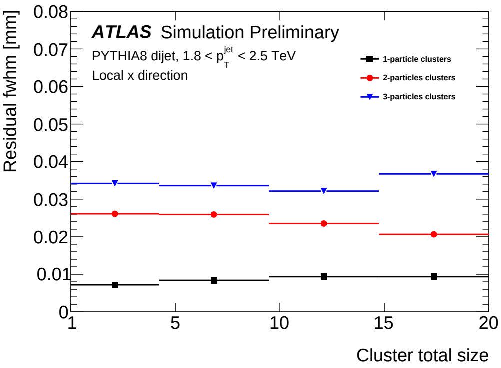


- Longitudinal direction

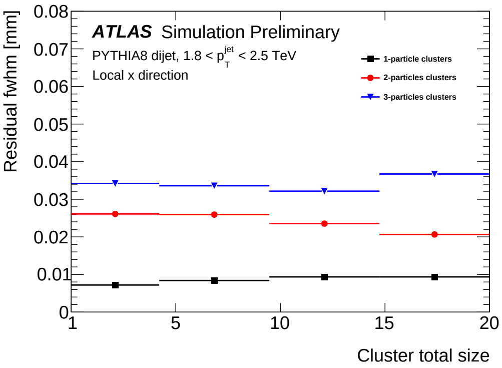


Hit position estimation

- Transverse direction

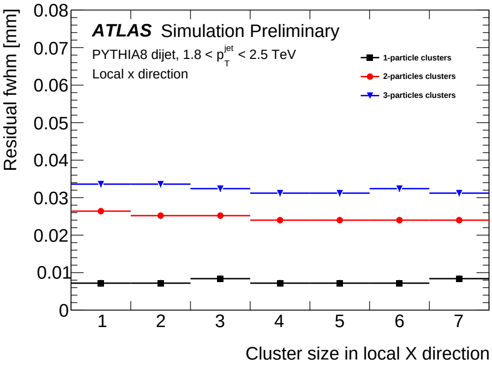


- Longitudinal direction

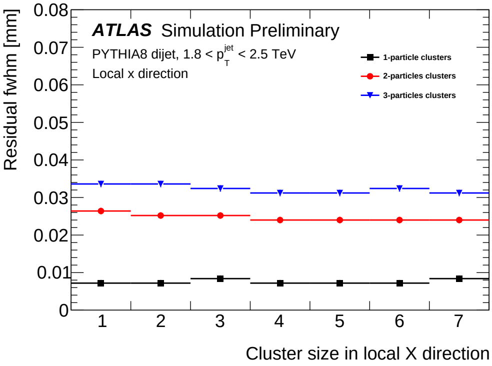


Hit position estimation

- Transverse direction

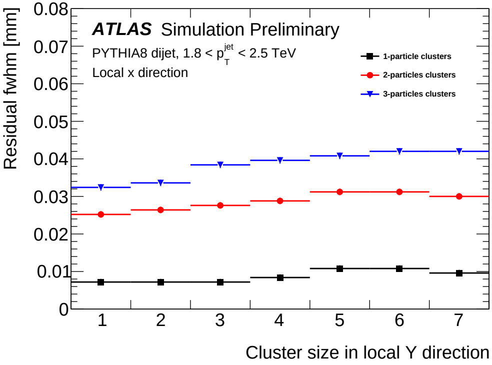


- Longitudinal direction

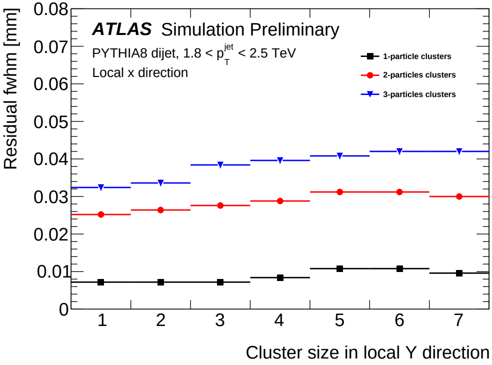


Hit position estimation

- Transverse direction

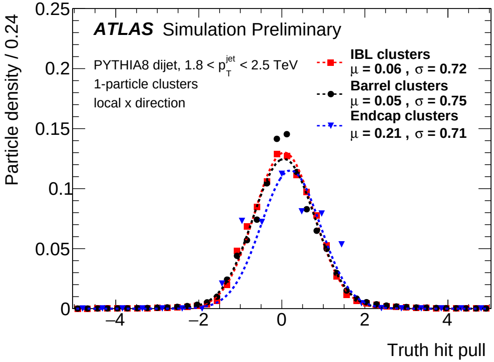


- Longitudinal direction

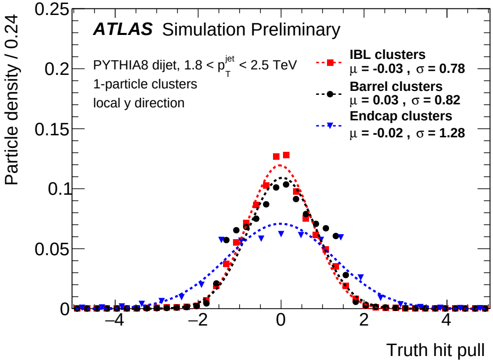


Uncertainty estimation

- Transverse direction

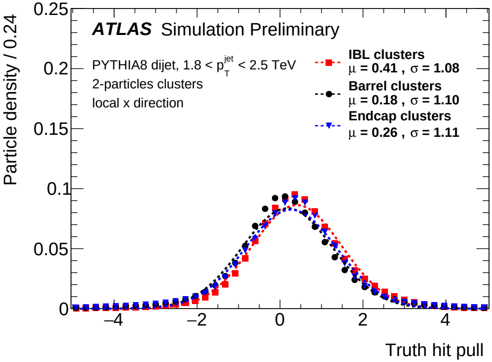


- Longitudinal direction

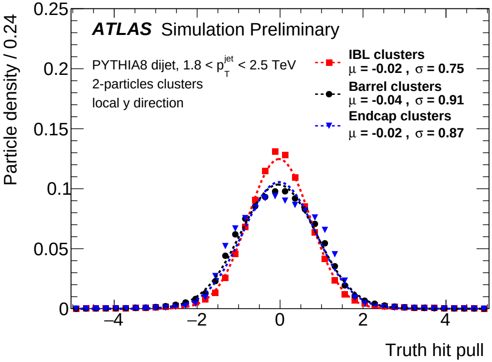


Uncertainty estimation

- Transverse direction

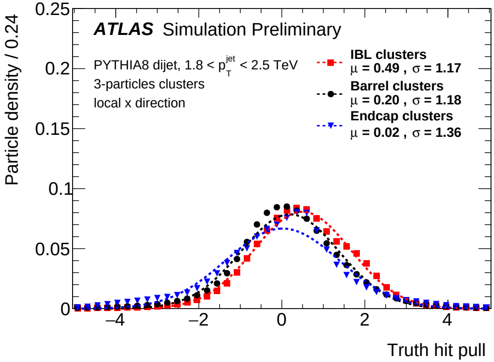


- Longitudinal direction

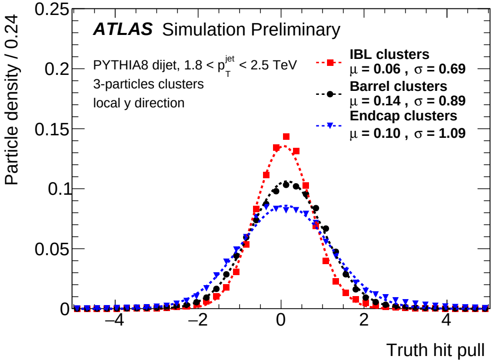


Uncertainty estimation

- Transverse direction

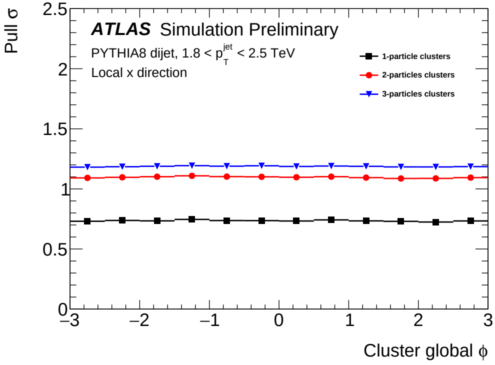


- Longitudinal direction

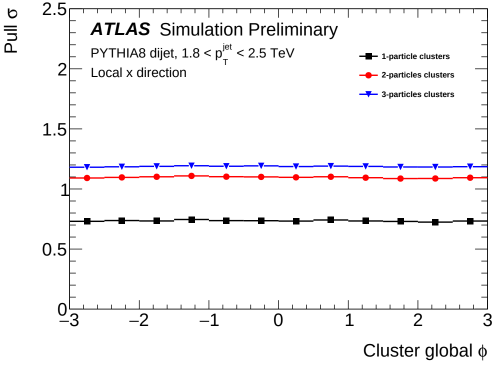


Uncertainty estimation

- Transverse direction

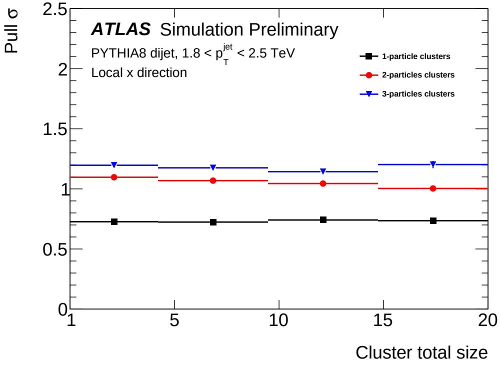


- Longitudinal direction

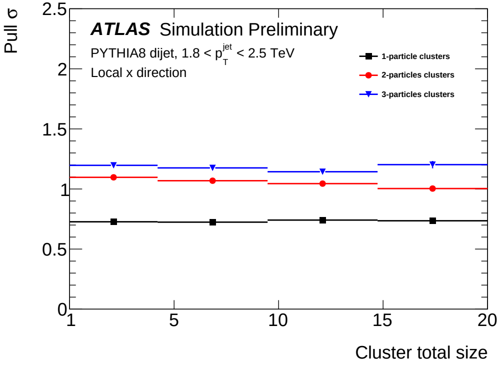


Uncertainty estimation

- Transverse direction

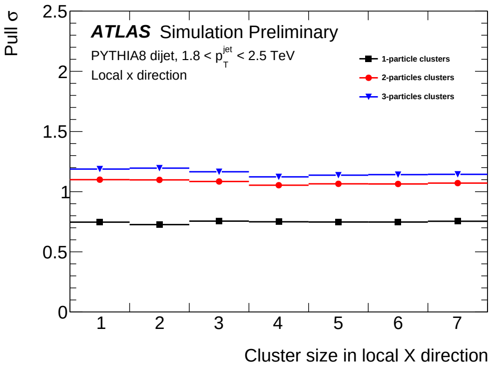


- Longitudinal direction

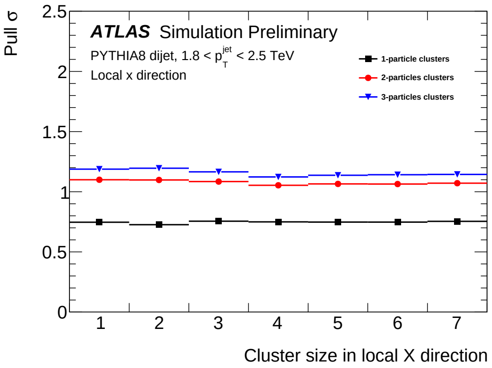


Uncertainty estimation

- Transverse direction

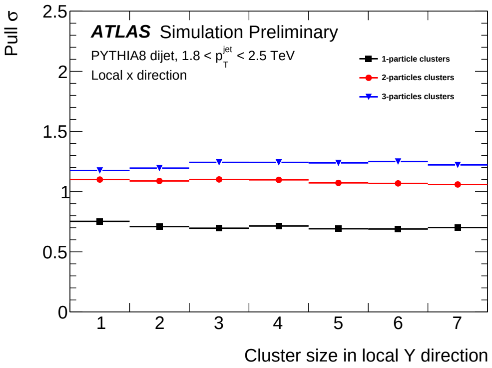


- Longitudinal direction



Uncertainty estimation

- Transverse direction



- Longitudinal direction