Towards Machine-Learning Particle Flow with the ATLAS Detector at the LHC

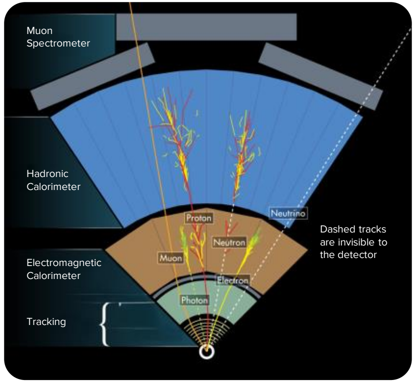
Luca Clissa, Maximilian Swiatlowski, Jessica Bohm, Joshua Himmens, Marko Jovanovic, Iacopo Vivarelli
on behalf of the ATLAS Computing Activity

luca.clissa2@unibo.it

October 19 - 25, 2024
CHEP 2024

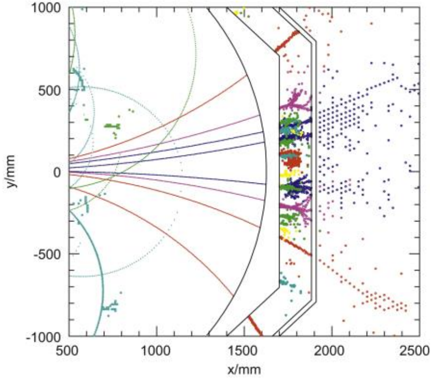
Particle flow in ATLAS

- **Problem:** particle identification & energy calibration
- Particularly challenging when we have jets/showers
- Key: exploit complementary components info:
    - tracker
    - calorimeters (calo)
- Particle flow (p-flow) algorithms reconstruct particle's trajectory and its energy deposit in detector components
- Inputs are tracks in the inner detector and topo-clusters in calorimeter
    - topo-clusters are groups of neighbouring cells ➜ useful to reconstruct showers in the calorimeter
- **Goal:** try to associate topo-clusters to tracks

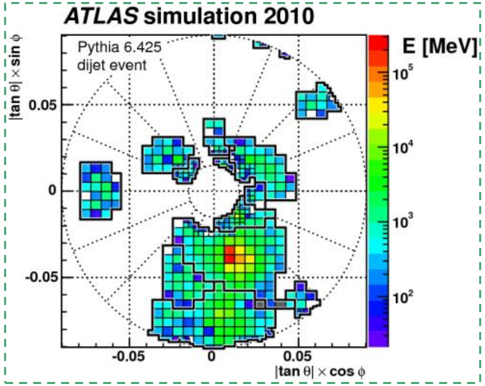


[ATLAS-OUTREACH-2021-052]

ILD experiment



[Nucl.Instrum.Meth.A611:25-40,2009]

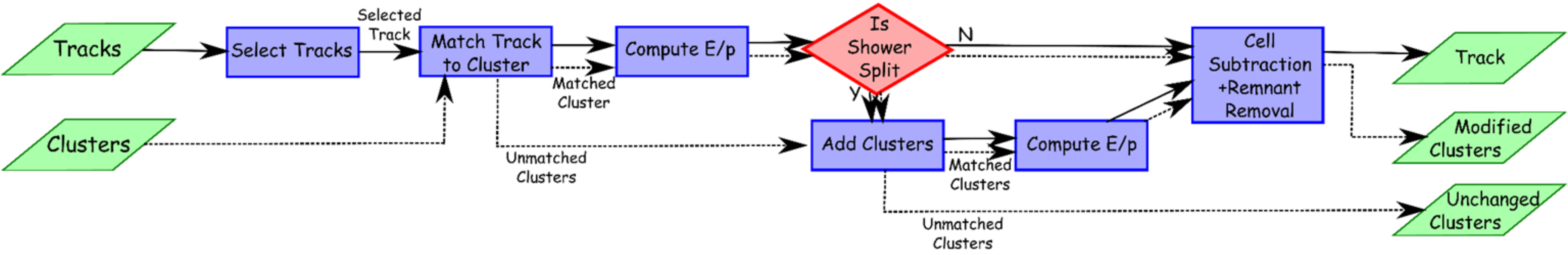


[Eur. Phys. J. C 77 (2017) 490]

ATLAS p-flow algorithm [Eur. Phys. J. C 77 (2017) 466]

For each track in descending pT:

1. associate closest topo-cluster based on angular distance ΔR'
2. compute expected energy deposit based on the topo-cluster position and track momentum
3. if expected and measured energies differ significantly, associate more topo-clusters
4. subtract the expected energy by calo cells
5. if remaining energy lies within expected fluctuations, remove the remnants
6. otherwise, consider leftovers for the next track



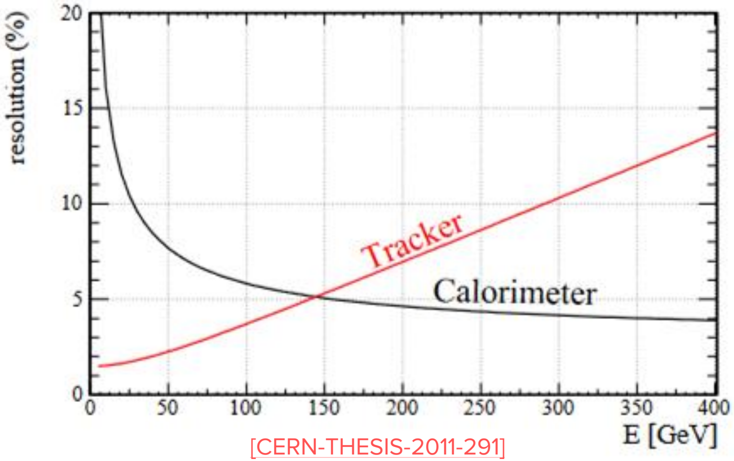
ATLAS p-flow algorithm: pros and cons

Existing ATLAS p-flow algorithm **strengths**:

- Calo + track information:
  ➜ improve energy resolution at low energy
- Good energy and angular resolution
- Pileup mitigation through “charged hadron subtraction”

Main **limitations**:

- Associate track to topo-clusters, not cells directly
  ➜ energy subtraction not flexible
- No calibration, only use detector measurements
- Tracker usage off above 100 GeV to avoid false matches

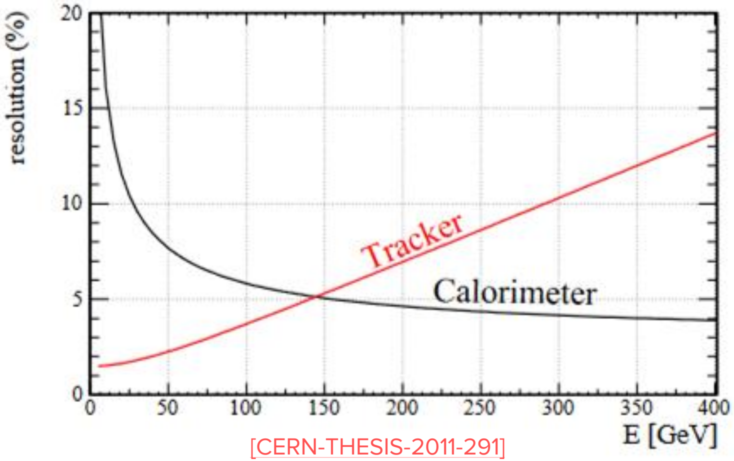


[CERN-THESIS-2011-291]

ATLAS p-flow algorithm: pros and cons

Existing ATLAS p-flow algorithm **strengths**:

- Calo + track information:
  ➔ improve energy resolution at low energy
- Good energy and angular resolution
- Pileup mitigation through “charged hadron subtraction”



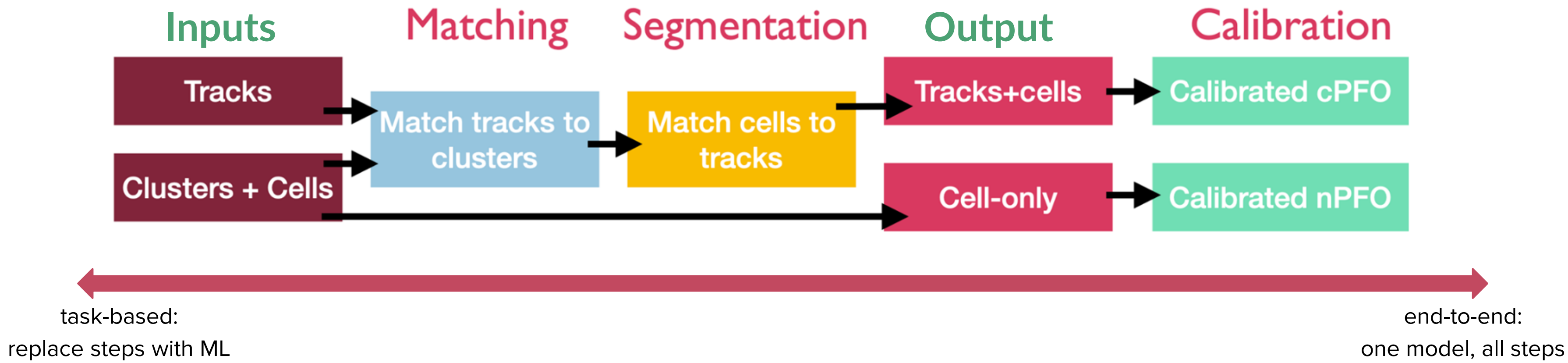
[CERN-THESIS-2011-291]

Main **limitations**:

- Associate track to topo-clusters, not cells directly
  ➔ energy subtraction not flexible
- No calibration, only use detector measurements
- Tracker usage off above 100 GeV to avoid false matches

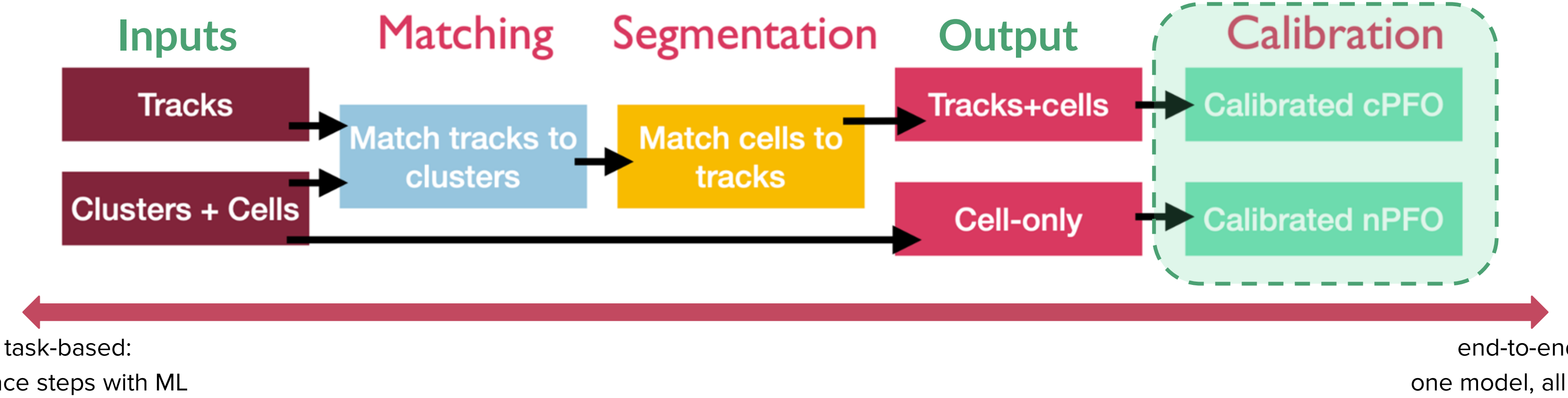
Can we do better? Maybe Machine Learning (ML) can help?

Machine Learning alternatives



- Machine Learning models have already shown promising results under various settings
  - HyperGraphs for end-to-end pflow [Eur. Phys. J. C 83 (2023) 596]
  - ongoing work on task-based solutions (matching, segmentation and calibration)
  - **image-based methods for calibration** [ATL-PHYS-PUB-2020-018] (central barrel reconstruction, $|\eta|<0.7$)

Machine Learning alternatives



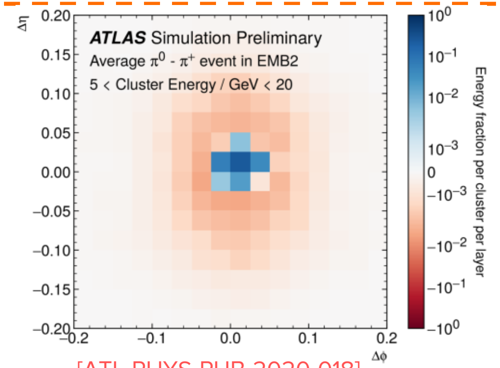
- Machine Learning models have already shown promising results under various settings
  - HyperGraphs for end-to-end pflow [Eur. Phys. J. C 83 (2023) 596]
  - ongoing work on task-based solutions (matching, segmentation and calibration)
  - **image-based methods for calibration** [ATL-PHYS-PUB-2020-018] (central barrel reconstruction, |η|<0.7)
    - ➔ **Outperform Local Hadronic Cell Weighting** (LCW) calibration
    - ➔ Work well for both identification and energy calibration
    - ➔ However, **inefficient representation** and **do not include tracking data**

Point cloud methods for p-flow [ATL-PHYS-PUB-2022-040]

- Focus on **pion identification and energy calibration**,
  - first step towards *hadronic shower reconstruction*
- Leverage **point cloud data**
  - only use actual hits, i.e. natural zero suppression
  - naturally handle varying granularity
  - naturally allow including tracking data
  - easily extend to including more information (momentum, hit confidence, …)
- Test 4 Deep Learning methods for point cloud data:
  - Graph Neural Network (GNN)
  - Deep Sets, Transformers, Merged Deep Fully Connected Network (DNN)
- Outline of extension to segmentation task

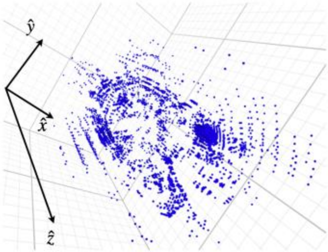
Why point cloud data?

- Image-based approaches are sub-optimal
  - different spatial granularity is difficult to render
  - only encode calorimeter information (**no tracker**)
  - irregular deposition geometries cause sparse images
    ➜ **inefficient representation**



[ATL-PHYS-PUB-2020-018]

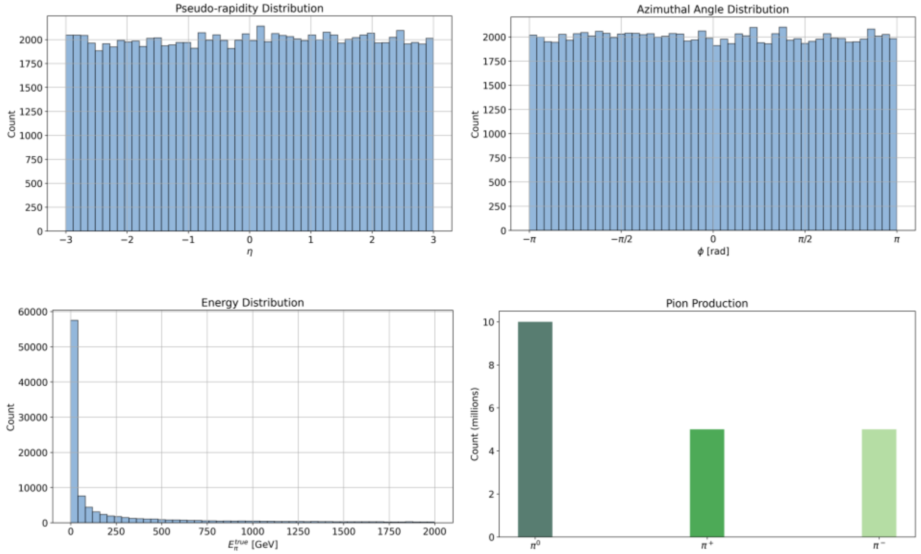
- Point cloud representation has several advantages
  - represent hits as 3D points with properties
    ➜ complex 3D shapes instead of series of images
    ➜ features like energy, hit confidence
  - including tracker is straightforward
  - only uses actual hits
    ➜ efficient representation



ATL-PHYS-PUB-2022-040

Dataset

- **Hadronic showers** originate primarily from pions
  - π^0: decay promptly to photons ➜ EM calo
  - π$^{+/-}$: more fluctuation in energy deposit patterns ➜ hadronic calorimeter
- Full ATLAS simulation using Geant4
- Uniform pion distributions in
  - azimuthal angle
  - pseudo-rapidity
  - log true energy
- 10M π^0, 5M π$^+$, 5M π$^-$
  - 3.5M training, 500k validation, 1M test after quality cuts
  - events with exactly 1 track



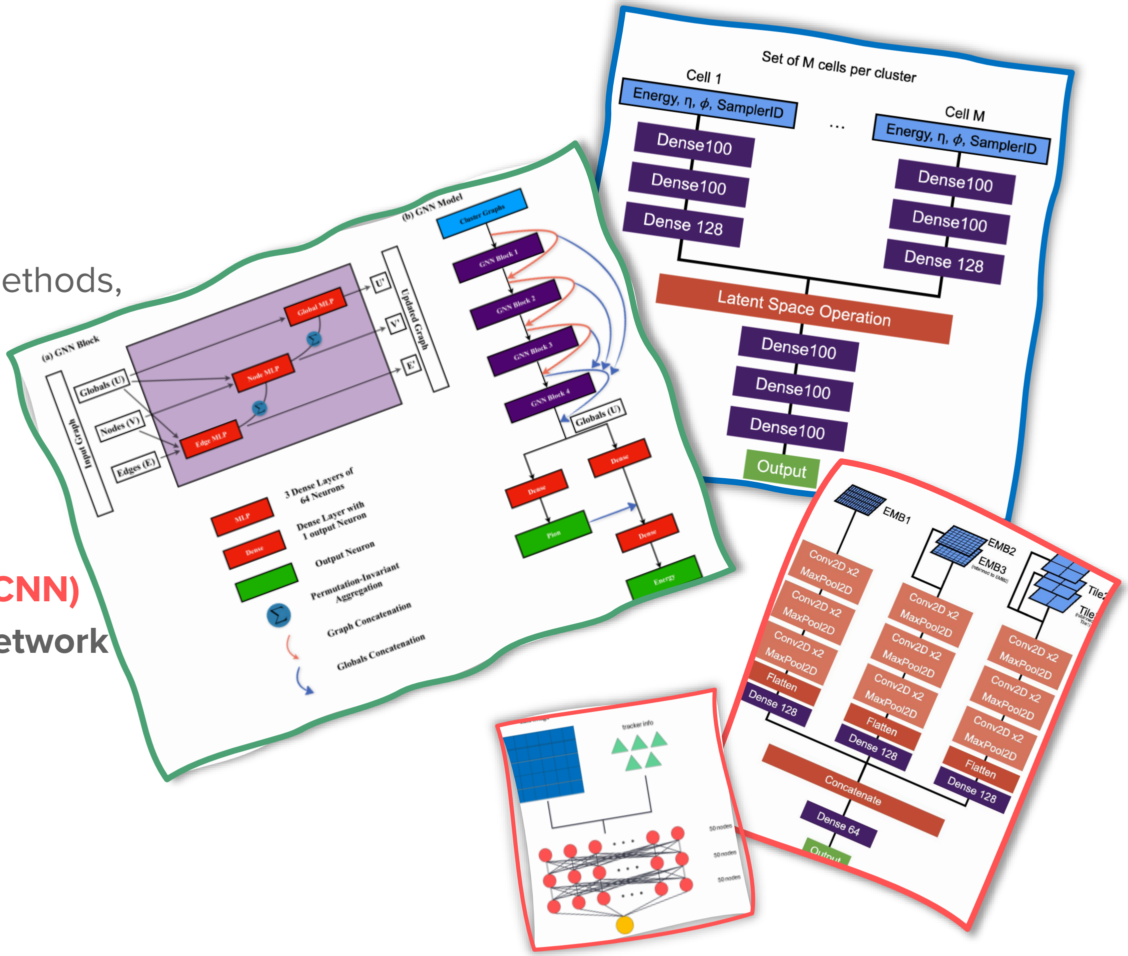
For illustration

Deep Learning methods

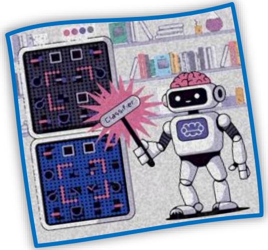
We explored several Deep Learning methods, only some of them shown here:

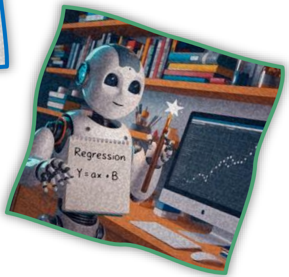
- **Graph Neural Networks (GNN)**
- **Deep Sets**
- Transformers
- **Convolutional Neural Networks (CNN)**
- **Merged Deep Fully Connected Network (DNN)**

→ image-based approaches



Learning tasks





Particle identification ➔ classification: π^0 VS π^+/π^-

- only calorimeter information
  ➔ adding tracks makes classification obvious
- input: one topo-cluster at a time

Energy calibration ➔ regression: calibrated energy

- **only calorimeter** information
- input: one topo-cluster at a time

- **calorimeter + tracker**
- input: one track + topo-clusters in $\Delta R<1.2$

Results

We compare ML approaches against two baselines depending on the learning task:

- classification
  ➜ Electromagnetic (**EM**) scale + initial hadronic calibration step corrections: $\mathcal{P}^{EM}_{cluster}$
- regression
  ➜ full Local Cell Weighting (**LCW**) calibration, i.e. $\mathcal{P}^{EM}_{cluster}$ + additional corrections: $\mathbf{E}^{LCW}_{cluster}$

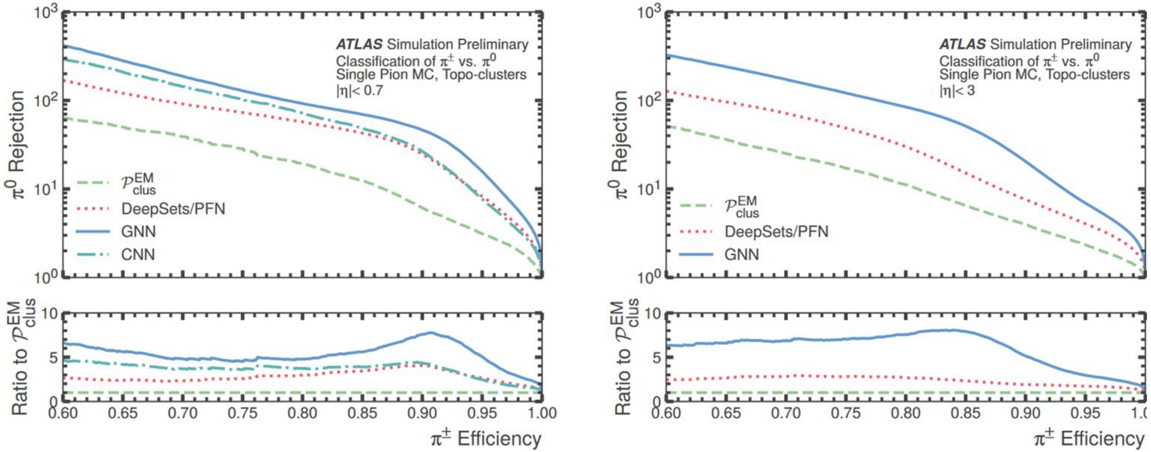
π^0 VS π^+/π^- classification: calo only

[ATL-PHYS-PUB-2022-040]

Metrics: $\pi^{\pm}$ efficiency VS π^0 rejection

- $\pi^{\pm}_{eff}$ = TP/(TP+FN)
- π^0_{rej} = 1 - FPR = TN/(TN+FP)

- ML methods outperform baseline $\mathcal{P}_{clus}^{EM}$
  - 4x to 8x background rejection in |η|<0.7
  - 2x to 6x in full η range
- **GNN performs best**
  ➜ 5x background rejection
- performance increases with higher topo-cluster energy



(a) $|\eta| < 0.7$ (b) $|\eta| < 3$

Model	Rej. @ 90% Eff. for $\|\eta\| < 0.7$	Rej. @ 90% Eff. for $\|\eta\| < 3$
CNN	26.584	-
GNN	46.419	20.500
Deep Sets	24.814	7.608
$\mathcal{P}^{EM}_{clus}$	6.123	3.977

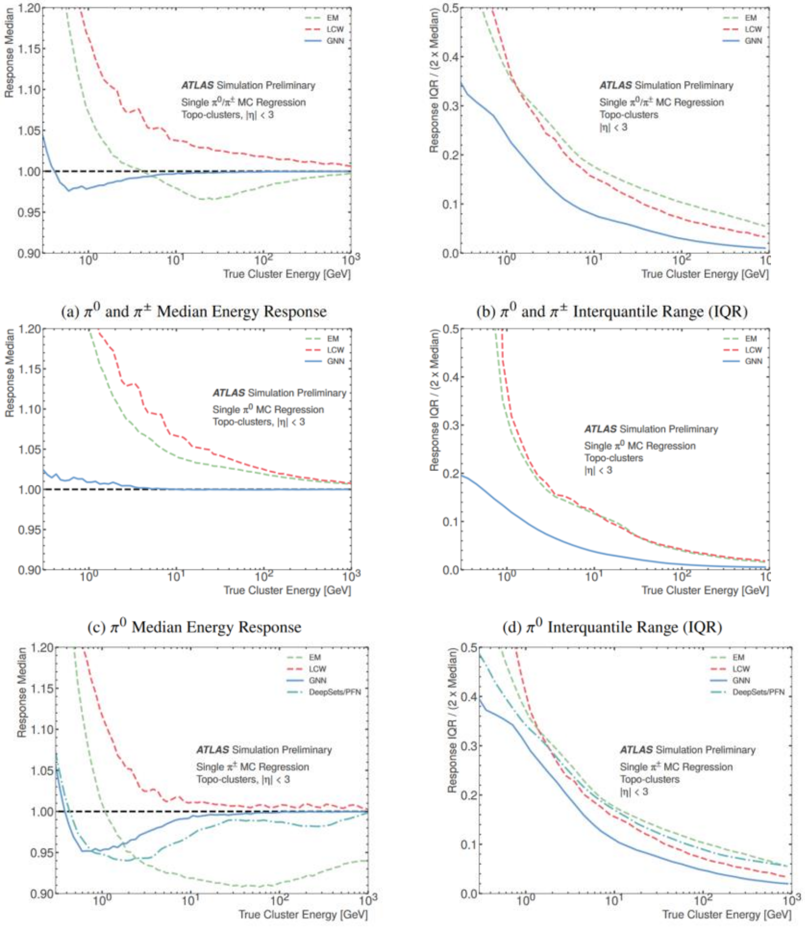
[ATL-PHYS-PUB-2022-040]

Energy regression: calo only

Metrics: median energy response and resolution
- energy response, $R = E_{pred}/E_{true}$
- resolution, IQR = median R ± 1σ (16-84%)

- ML significantly better than traditional calibrations across entire energy spectrum
  ➜ R closer to 1; lower IQR

- **GNN is best overall**
- **Deep Sets better than baseline for charged pions, especially at low-energy** (< 1 GeV)
  ➜ known weakness in conventional techniques
- ML mitigates long-standing calibration issues
  - high-energy $\pi^{\pm}$ underestimation
  - low-energy π^{0} overestimation



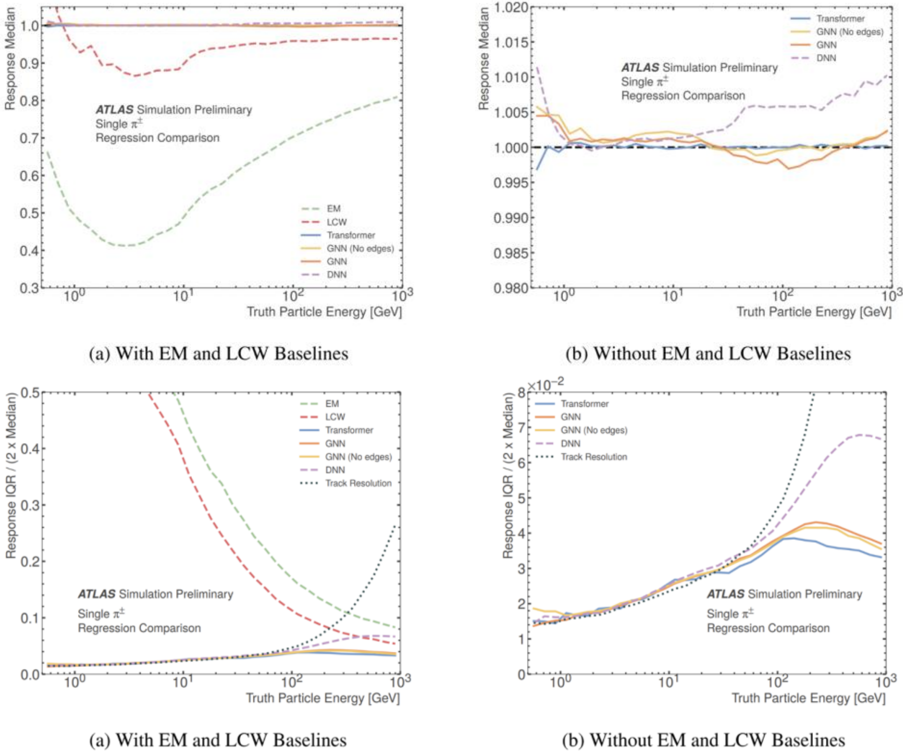
(a) π^0 and $\pi^\pm$ Median Energy Response
(b) π^0 and $\pi^\pm$ Interquantile Range (IQR)
(c) π^0 Median Energy Response
(d) π^0 Interquantile Range (IQR)
(e) $\pi^\pm$ Median Energy Response
(f) $\pi^\pm$ Interquantile Range (IQR)

Energy regression: calo + tracker

Metrics: median energy response and resolution
- energy response, $R = E_{pred}/E_{true}$
- resolution, IQR = median R ± 1σ (16-84%)

- Point cloud models VS baseline: significantly outperform EM and LCW calibration
  - better R and IQR across the full energy spectrum
- Point cloud VS image-based (DNN):
  - comparable median accuracy for E < 30 GeV
  - superior performance for E > 30 GeV
- Track information dramatically improves prediction
  ➜ IQR consistently below 0.1
  (VS 0.4 for cluster-only)
- Adding cell-level info further improves resolution, particularly at high energy (more in backup slides)

[ATL-PHYS-PUB-2022-040]



(a) With EM and LCW Baselines

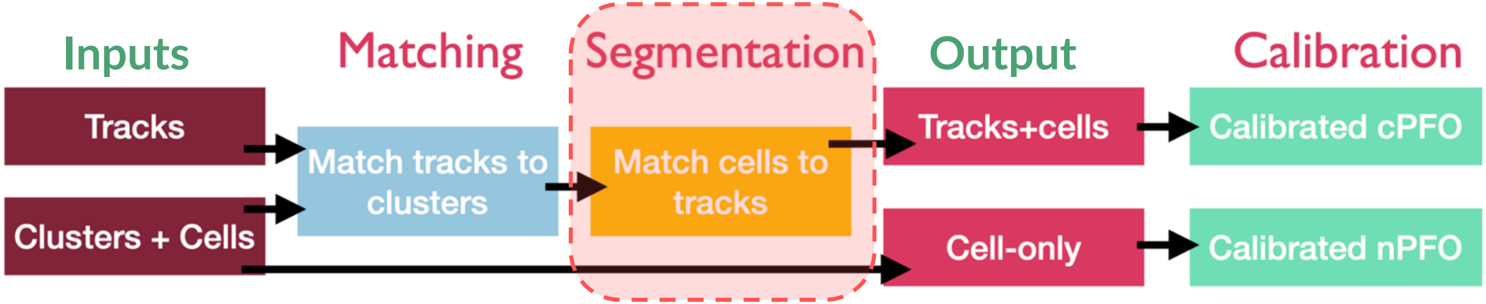
(b) Without EM and LCW Baselines

(a) With EM and LCW Baselines

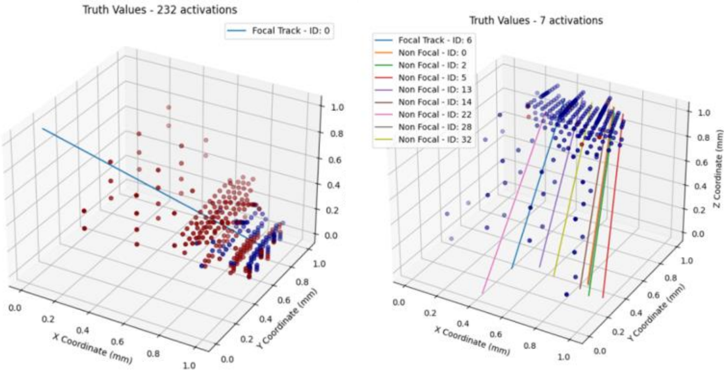
(b) Without EM and LCW Baselines

Next steps

Cells-to-track matching



- Extend point cloud methods to tackle cells-to-track matching
  - one focus track at a time
  - all hits within ΔR=0.2 (tracker + calo)
  - associate hits with track contributing the most energy (>50%)
  - PointCloud architecture [6], attempt with MaskFormers [7]
- Promising results for simple ρ, Δ decays (~1 track per event)
- Trying to generalize to more challenging dijets scenarios



For illustration

Conclusion

- Significant improvement in $\pi^0/\pi^\pm$ classification and energy regression
- Key findings from calorimeter-only regression:
  - GNN and Deep Sets outperform traditional calibrations across all energies
  - They mitigate long-standing calibration issues at the boundaries of energy values
  - **point cloud methods outperform image-based approaches**
    ➙ and **more efficient!**
- Combined calorimeter and tracker regression:
  - ML models surpass EM/LCW scales
  - Dramatic improvement in energy resolution (IQR/median < 0.1)
  - Pointcloud advantage increases at high energies (> 30 GeV)
  - Granular cell-level data further enhances results
- Outlook: Promising step towards ML-optimized Particle Flow in ATLAS

References

[1] Aaboud, M., Aad, G., Abbott, B. et al. Jet reconstruction and performance using particle flow with the ATLAS Detector. Eur. Phys. J. C 77, 466 (2017). https://doi.org/10.1140/epjc/s10052-017-5031-2
[2] Di Bello, Francesco Armando, et al. "Reconstructing particles in jets using set transformer and hypergraph prediction networks." The European Physical Journal C 83.7 (2023): 596.
[3] Angerami, Aaron, and Piyush Karande. Deep Learning for Pion Identification and Energy Calibration with the ATLAS Detector. No. LLNL-JRNL-813169; ATL-PHYS-PUB-2020-018. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.
[4] ATLAS collaboration. Point Cloud Deep Learning Methods for Pion Reconstruction in the ATLAS Experiment. ATL-PHYS-PUB-2022-040, CERN, Geneva, 2022.
[5] Thomson, M. A. "Particle flow calorimetry and the Pandora PFA algorithm." Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 611.1 (2009): 25-40.
[6] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[7] Van Stroud, Samuel, et al. "Vertex Reconstruction with MaskFormers." arXiv preprint arXiv:2312.12272 (2023).
[8] Aad, Georges, et al. "Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1." The European Physical Journal C 77.7 (2017): 1-73.
[9] Fleischmann, Sebastian. "Tau lepton reconstruction with energy flow and the search for R-parity violating supersymmetry at the ATLAS experiment." (2012).

Any questions?

Backup

Graph Neural Network

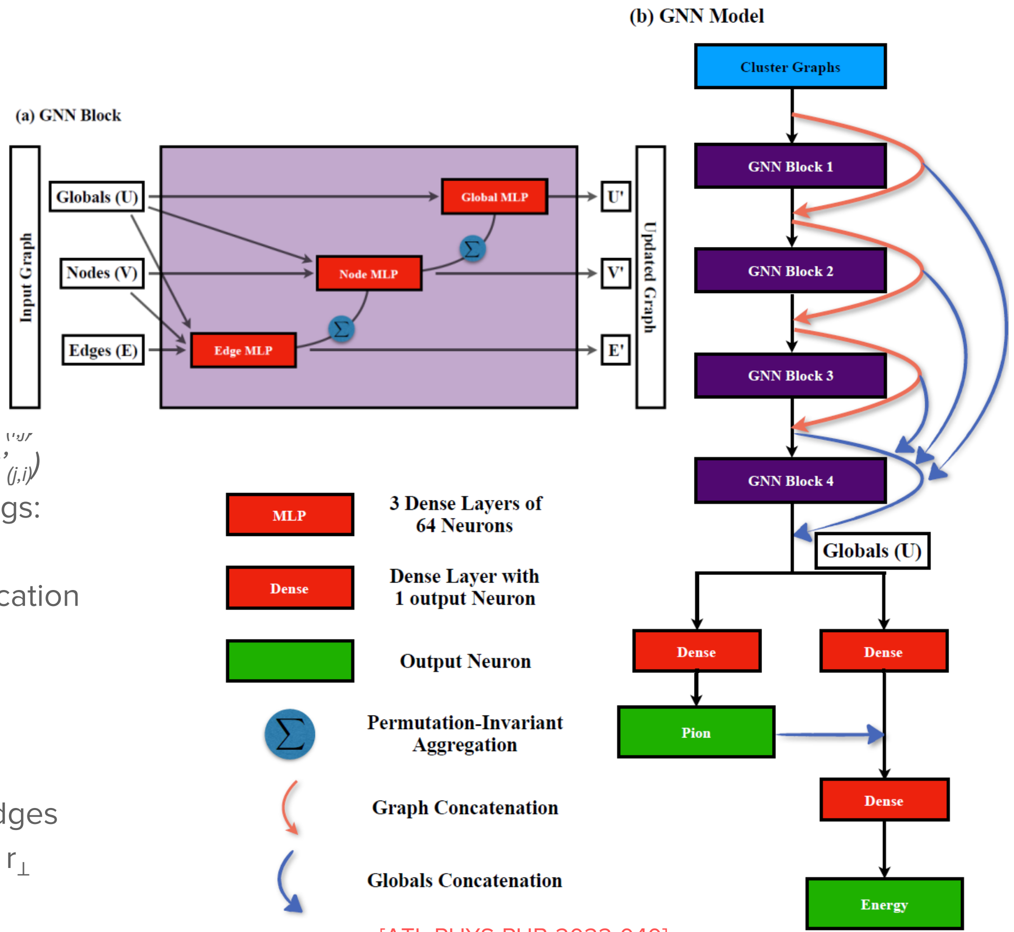
Architecture

- 4 GNN blocks with Multi-Layer Perceptrons (MLP)
- Message passing to learn hidden representation
  - update edges: $x'_{(i,j)} = f_{edge}(x_i, x_j,$
  - update nodes: $x'_i = f_{node}(x_i, \Sigma_{j \in N_i} x'_{(j,i)})$
- Graph-level features as function of node embeddings:

  $g'_i = f_{global}(g, \Sigma_{i \in N} x'_j)$
- Global features concatenated with input for classification
- Simultaneous classification and regression tasks

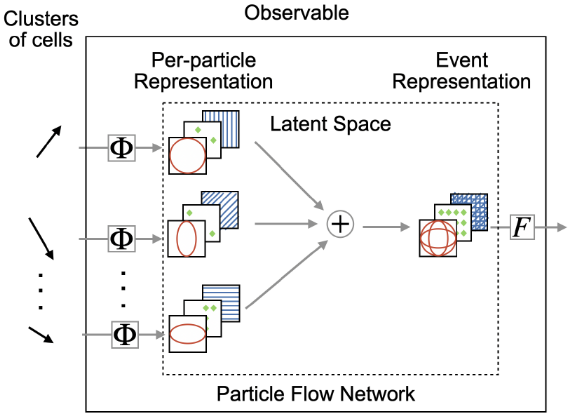
Components

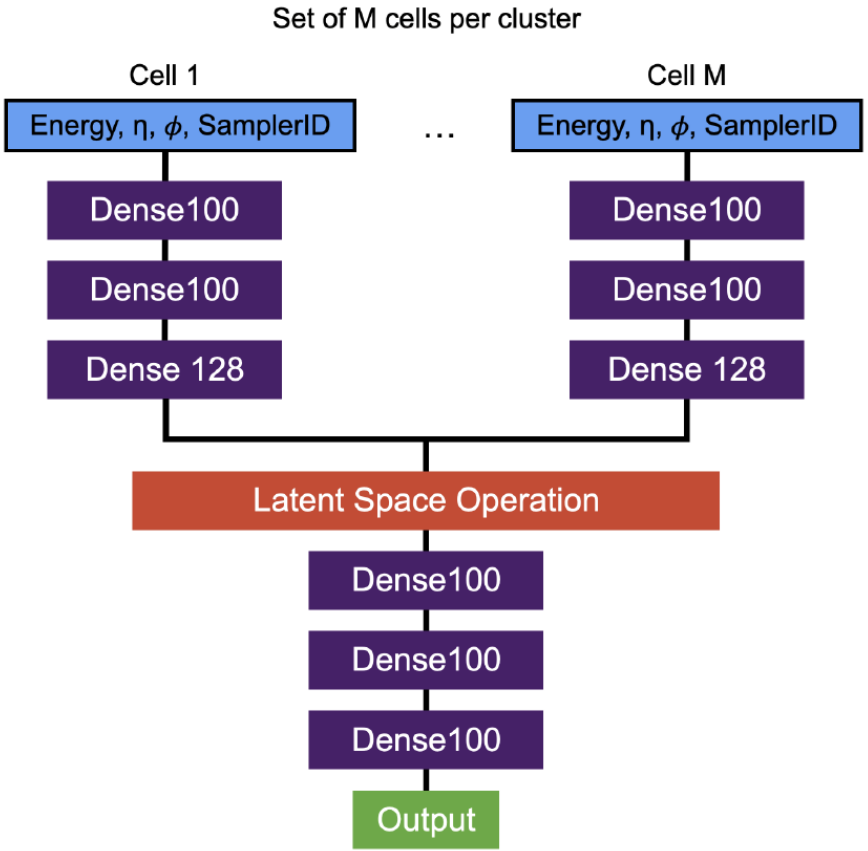
- Cells are nodes, neighboring cells connected by edges
- Node features: energy sampling layer η, $\Delta\eta$, ϕ, $\Delta\phi$, $r_\perp$
- Edge features: type of connection



[ATL-PHYS-PUB-2022-040]

Deep Sets

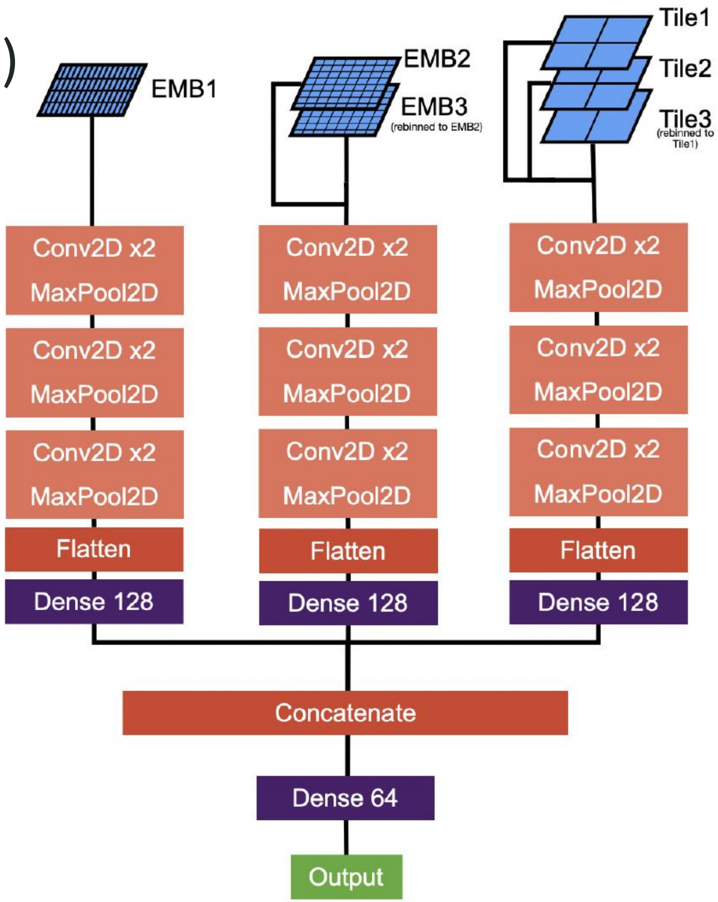

Clusters of cells
Observable
Per-particle Representation
Event Representation
Latent Space
Particle Flow Network
Φ
F


Set of M cells per cluster
Cell 1
Cell M
Energy, η, ϕ, SamplerID
Energy, η, ϕ, SamplerID
Dense100
Dense100
Dense 128
Latent Space Operation
Dense100
Dense100
Dense100
Output

Convolutional Neural Networks (CNN)

- pixels are bidimensional projections of cell baricenters
- pixel intensity reflects energy deposit
- considers calo layers separately to account for different granularity
    - EMB1 alone
    - EMB2, EMB3 together
    - Tile1, Tile2 and Tile3 together

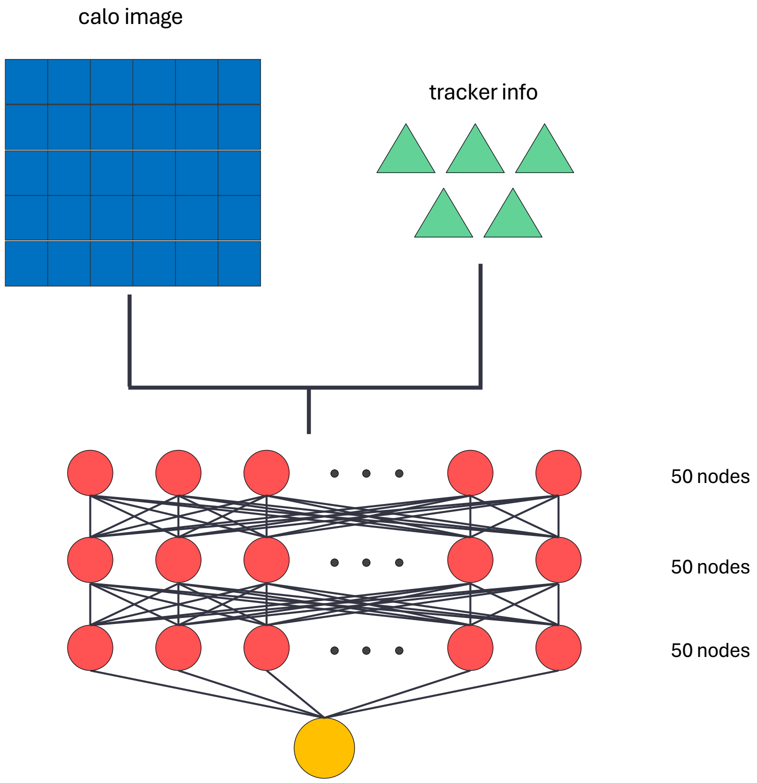
Calorimeter Layer	$(\Delta\eta, \Delta\phi)$ Granularity
EMB1	128×4
EMB2	16×16
EMB3	8×16
Tile1	4×4
Tile2	4×4
Tile3	2×4



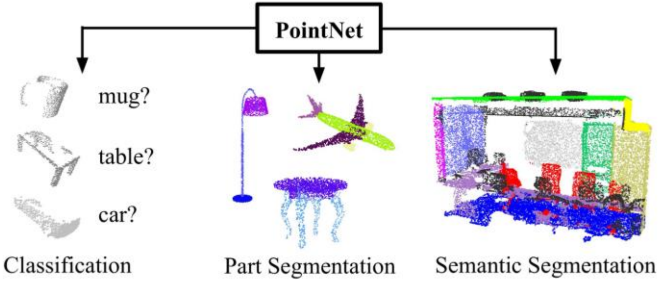
Merged Deep Fully Connected Neural Networks (DNN)

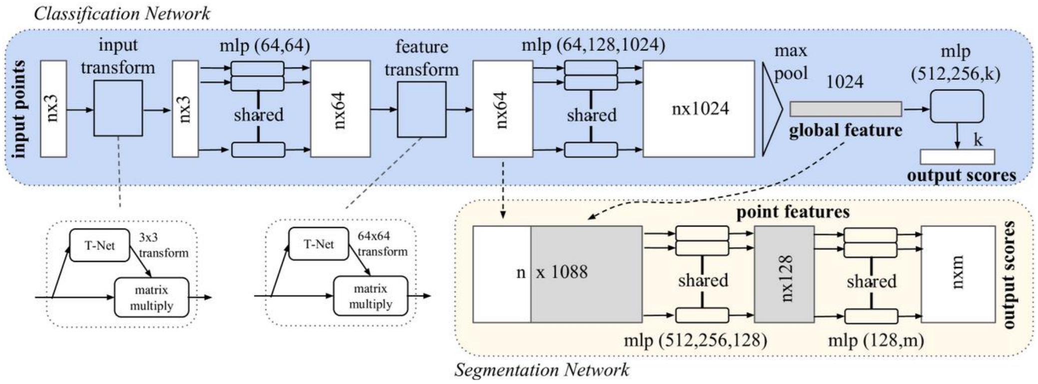
- image-based approach
  - EMB1 alone
  - EMB2, EMB3 together
  - Tile1, Tile2 and Tile3 together

- 3 fully connected hidden layers
- 50 nodes in each hidden layer
- outputs calibrated energy values



PointNet model





- 👍 Several learning tasks: classification, part segmentation, semantic segmentation
- 👍 permutation invariant
- 👍 transformation equivariance
- 👍 both shape classification & segmentation
- 👍 robust to data corruption → critical points

- 👎 no local context → global feature learning
- 👎 generalization to unseen scenes → global features depend on absolute coordinates
- 👎 no rotation/shape equivariance

Calo + track results using cell-level information

- Severall GNN configurations attempted
  - Leadining cluster only VS all clusters
  - With VS w/o edges
  - With VS w/o cell info
- GNN with cell-level data (red, light blue) improves resolution compared to versions trained without this information under several configurations