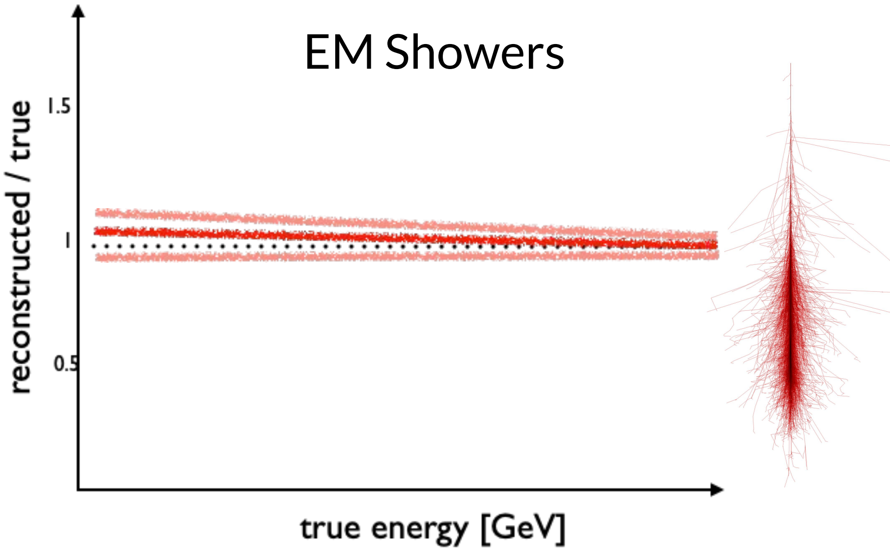
Application of DeepSets Machine Learning in FPGA to Improve ATLAS L0 Global Trigger for HL-LHC

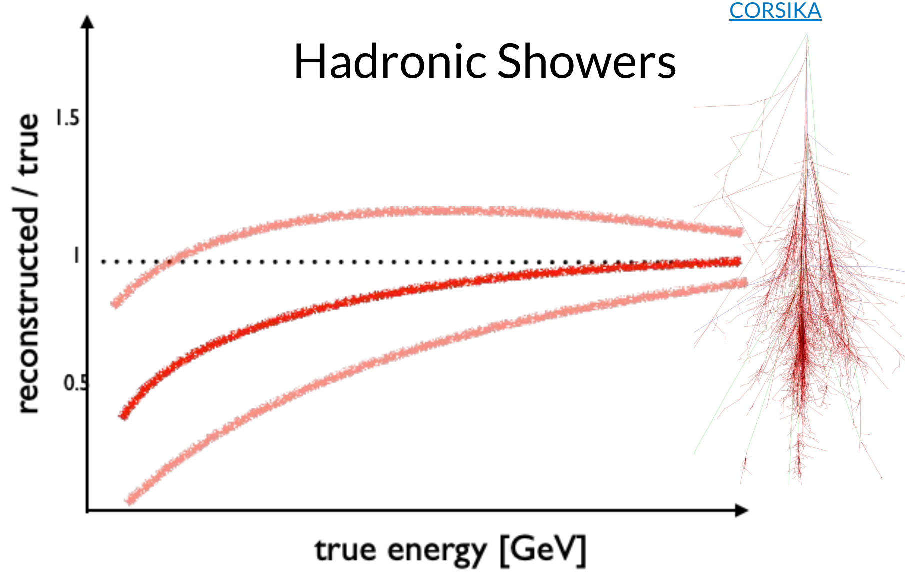
WNPPC-2025
2025-02-15

Kelvin Leong, Maximilian Swiatlowski, Colin Gay, Wotjtek Fedorko

Calorimeters & Showers

Simulations from [CORSIKA]





EM showers (e, γ, $\pi^0 \rightarrow 2\gamma$) are well-calibrated, measured "correctly".

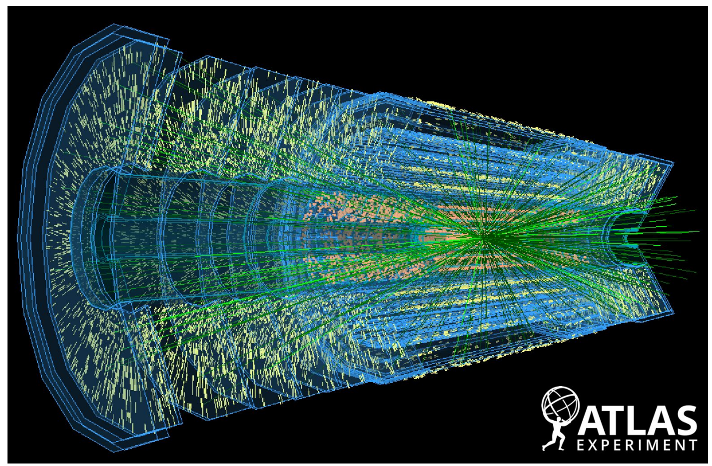
All showers are similar, resolution is good.

Hadronic showers (p, n, $\pi^{\pm}$) are more difficult to calibrate

Each shower is unique, huge resolution penalty from variations.

ATLAS Upgrade

- LHC will be upgraded to High-Luminosity LHC in ~2026-2030
  - Subsequent run periods will expect an environment to 200 events per pp-collision bunch crossing (pileup), ~4x more than current run

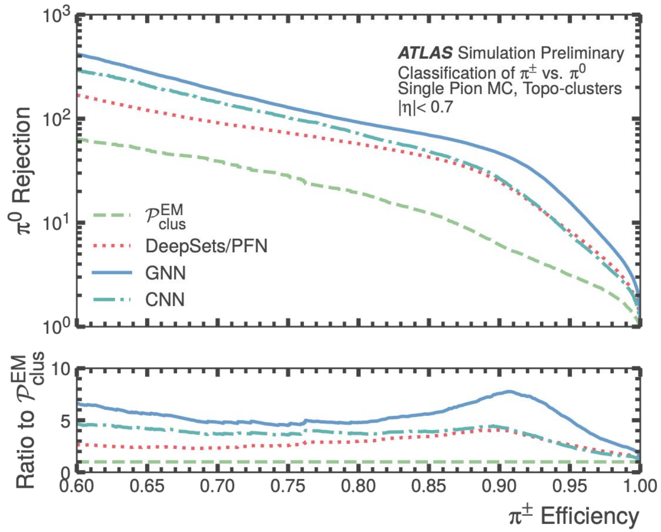


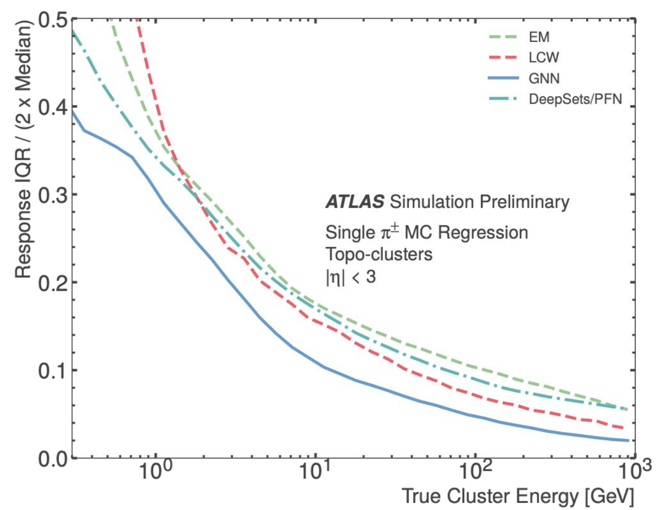
HL-LHC simulated event in tracker
ATLAS-PHOTO-2023-042-1

- Need a more efficient trigger system in ATLAS in the high-pileup environment
  - Interesting events could be inaccurately recorded, or missed by trigger and lost forever

Machine Learning for Offline Shower Types Identification & Calorimeter Calibration

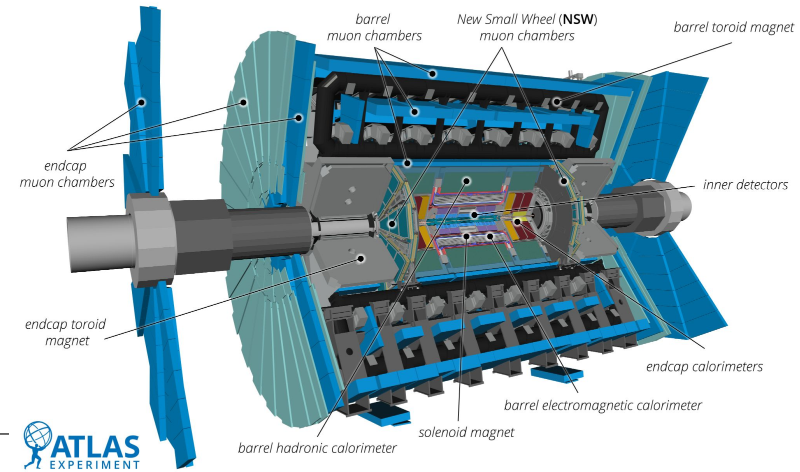
- Pions ($\pi^0, \pi^{\pm}$) are most abundant particle produced in collisions in LHC
- Previous studies showed ML methods (e.g. GNN, DeepSets, …) outperforms the current algorithms for identifying shower types and calibrating energy in the calorimeter in offline.
    - ATL-PHYS-PUB-2020-018
    - ATL-PHYS-PUB-2022-040

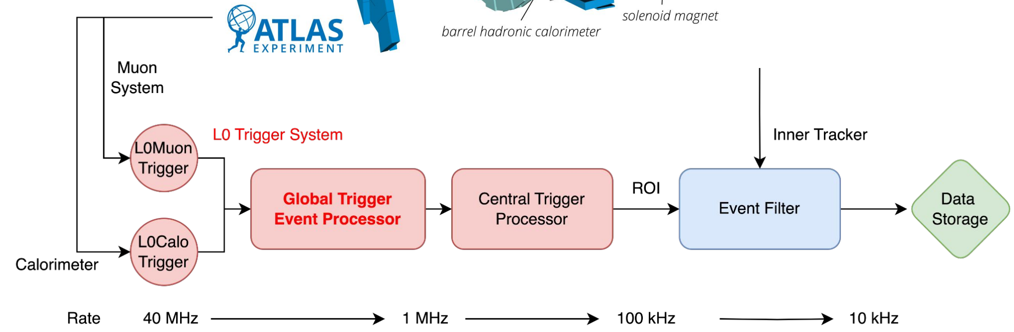




Can we apply ML to online trigger?

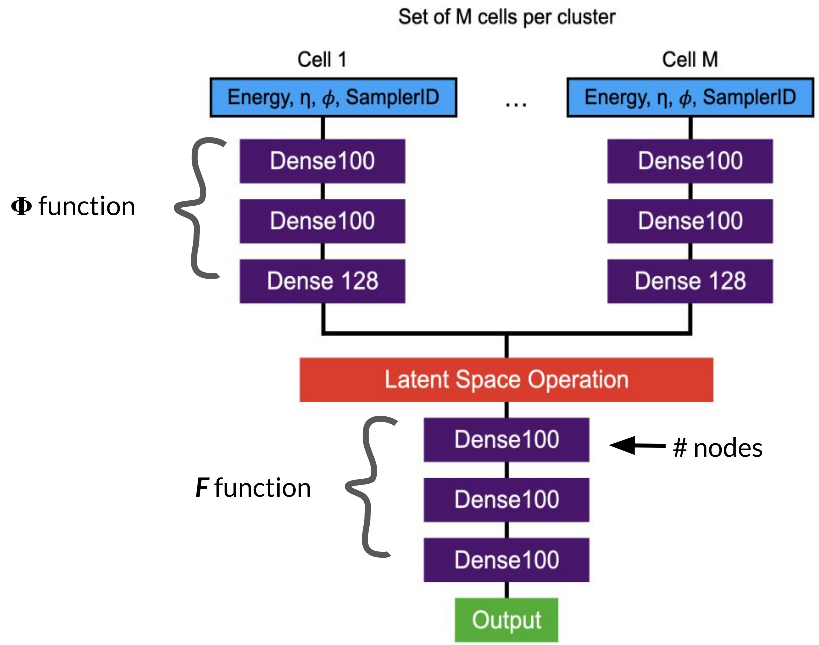
- Can ML be applied in the L0 global trigger event processor for pion calibration?
  - L0 Event Selection:
    - Topo-clustering, calibration possible
    - Identify target items (leptons, jets)
    - Reduces data rate 40MHz → 1MHz, in total latency of 10 μs
  - FPGA in the event processor
    - Each algorithm in global trigger aims for ~5% resources in FPGA, algorithms to be improved / added over time
    - Requires conversion of NN to hardware code



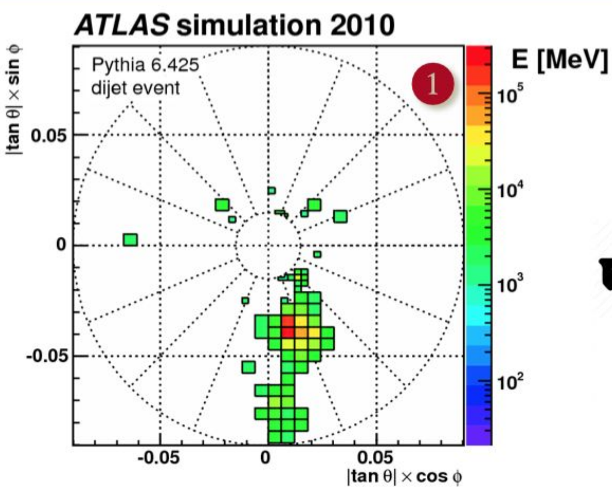


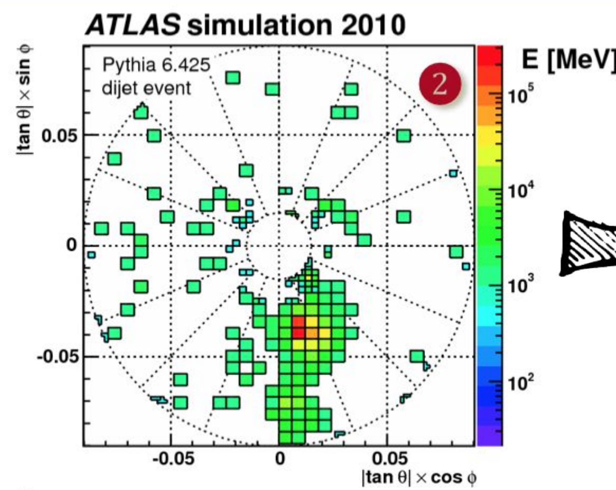
DeepSets Neural Network

- DeepSets: (3 stage model)
  - Run a Φ function (standard NN) on every cell
  - Sum all the results to “latent space”
  - Run a final ***F*** network, get outputs (classification & regression)
  - Relationships of cells are encoded in latent space



Topo-Clustering for HL-LHC





arXiv:1603.02934

❸ **Apply neighbouring clusters splitting**

❶ **Initial seed collection**

All cells above seed threshold

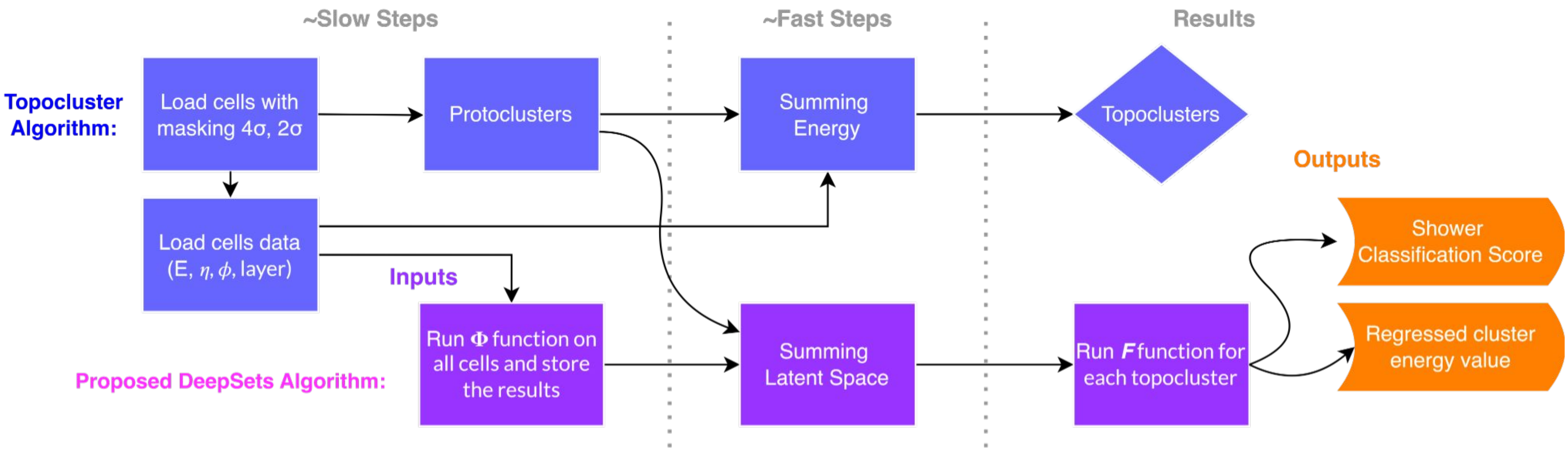
$$|E_{\text{cell}}^{\text{EM}}|/\sigma_{\text{noise,cell}}^{\text{EM}} > 4$$

❷ **Cells above growth control threshold collection**

Cells potentially contributing to cluster growth

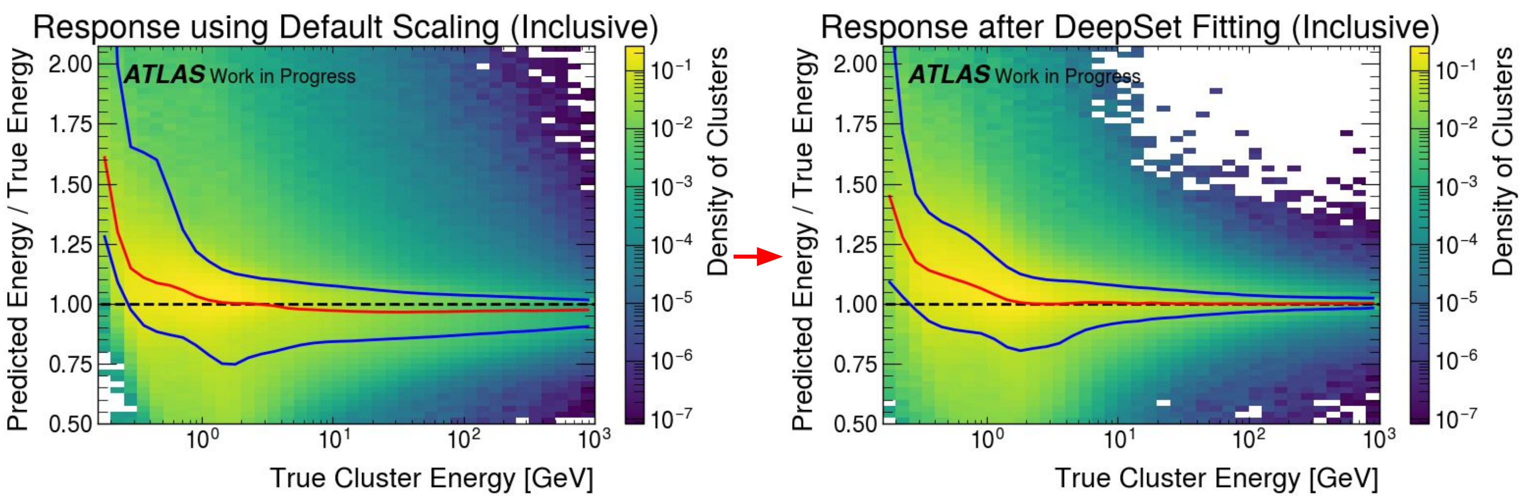
$$|E_{\text{cell}}^{\text{EM}}|/\sigma_{\text{noise,cell}}^{\text{EM}} > 2$$

Proposal:
DeepSets under the shadow of Topo-Clustering



Neural network Performance

- Trade-off studies with MC simulation
  - To work within the shadow of topo-clustering algorithm, we want small # of nodes (especially in $\mathbf{\Phi}$ function), while balancing # of hidden layers
  - Best candidate so far:

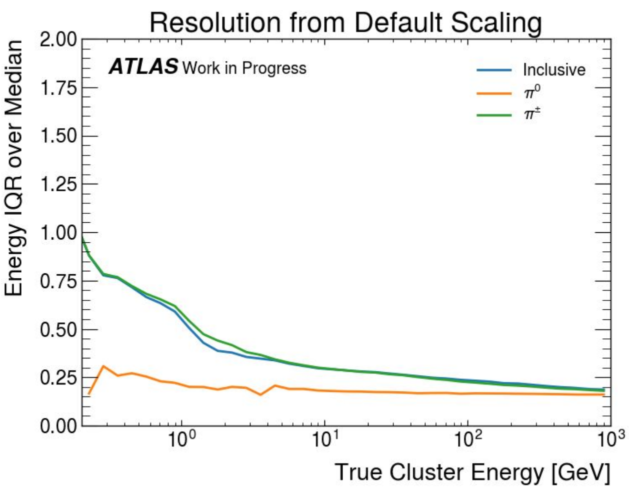


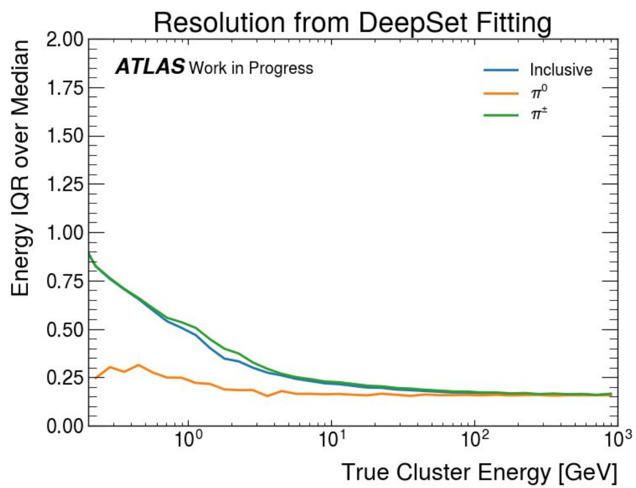
- Training details:
  - ~14M clusters with π^0 & $\pi^{\pm}$ labels (mc21)
  - Epochs: 100
  - Learning rate: 0.001
  - Loss weights:
    - Regression: 99%
    - Classification: 1%
- Model details:
  - $\mathbf{\Phi}$ nodes: 96
  - $\mathbf{\Phi}$ layers: 5
  - $\boldsymbol{F}$ nodes: 192
  - $\boldsymbol{F}$ layers: 5

- Red line=Median response of test samples; blue line=1σ response

Neural network Performance

- Trade-off studies with MC simulation
  - To work within the shadow of topo-clustering algorithm, we want small # of nodes (especially in **Φ** function), while balancing # of hidden layers
  - Best candidate so far:



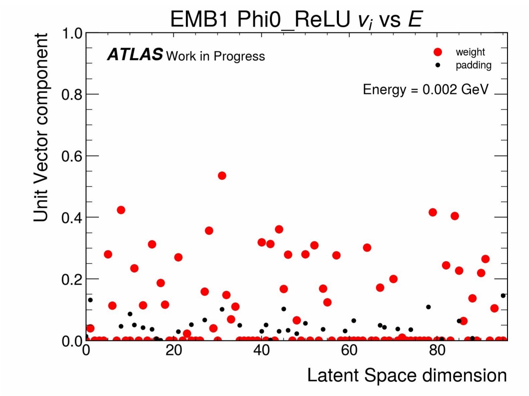


- Training details:
  - ~14M clusters with π^0 & $\pi^\pm$ labels (mc21)
  - Epochs: 100
  - Learning rate: 0.001
  - Loss weights:
    - Regression: 99%
    - Classification: 1%
- Model details:
  - **Φ** nodes: 96
  - **Φ** layers: 5
  - ***F*** nodes: 192
  - ***F*** layers: 5

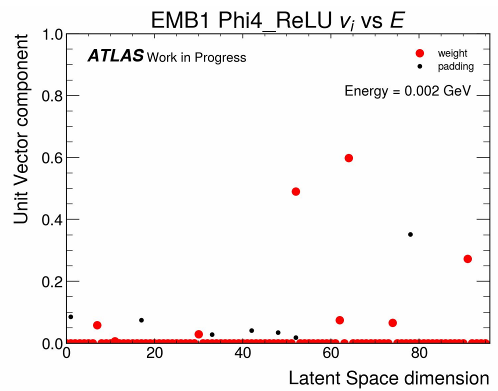
Possible Improvement for NN

- Probe trained model with 1-cell-cluster, and varying cell energy

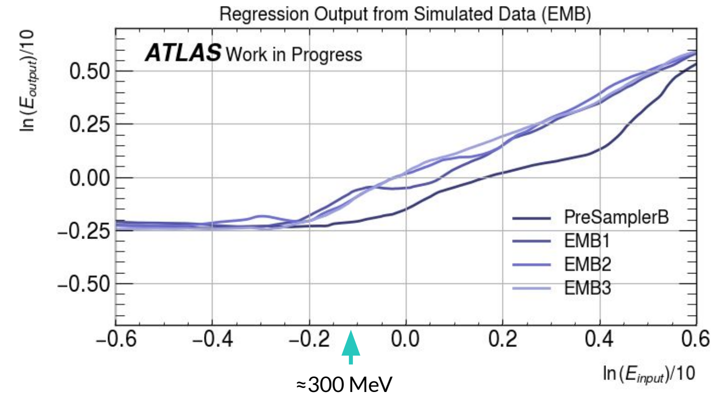
(First hidden layer)



(Last hidden layer)



(Final output)



- Information are encoded in a few vector components in later hidden layers, rather than using all vectors
  - Possible to further reduce resource usage in **Φ-layers**

- Plateau at <300 MeV could be electronic noise
  - Possible explanation for over-prediction for <300 MeV in NN fitting

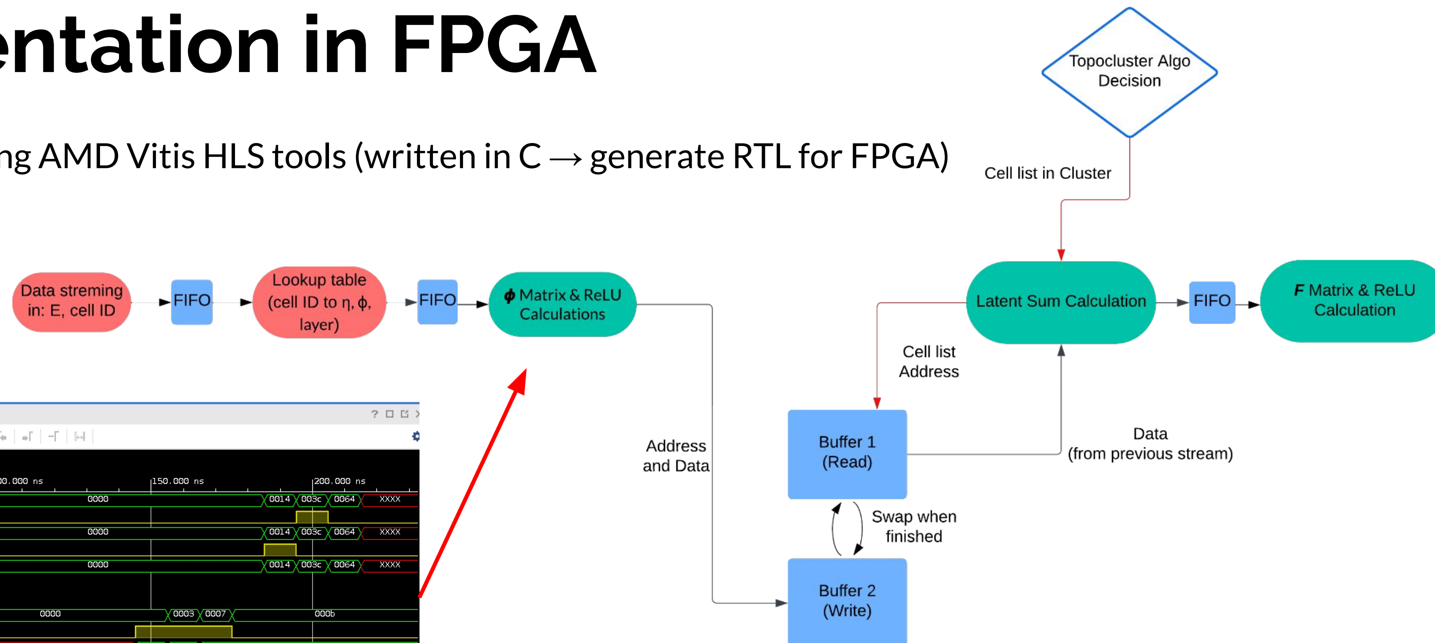
Conclusion

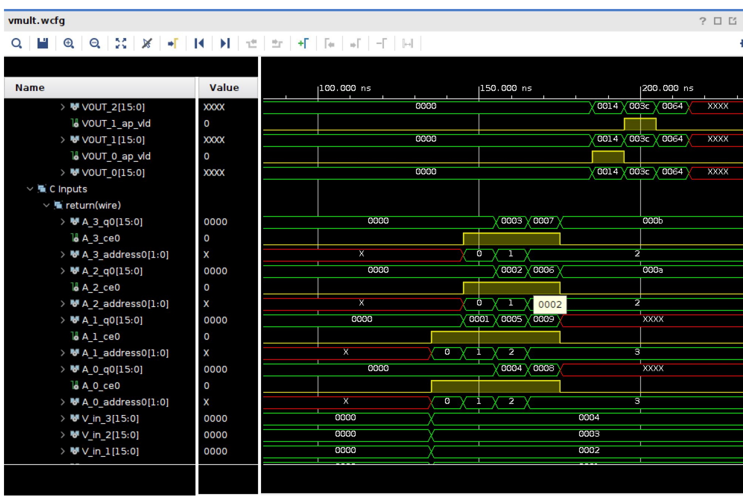
- **Problem trying to solve:**
  - Improve calorimeter calibration for online trigger during the HL-LHC
- **Why ML?**
  - ML performs better than current algorithms in pion classification and energy calibration
- **Challenges ahead:**
  - To be run in FPGA in the L0 global trigger under the shadow of the current topocluster algorithm
  - Strict requirement on latency and limitation of resource to implement the DeepSets neural network
  - Need DeepSets to recognize electronic noise to prevent over-prediction at low energy
  - Integration to future data flow and format with other trigger processes in hardware

Backup

Implementation in FPGA

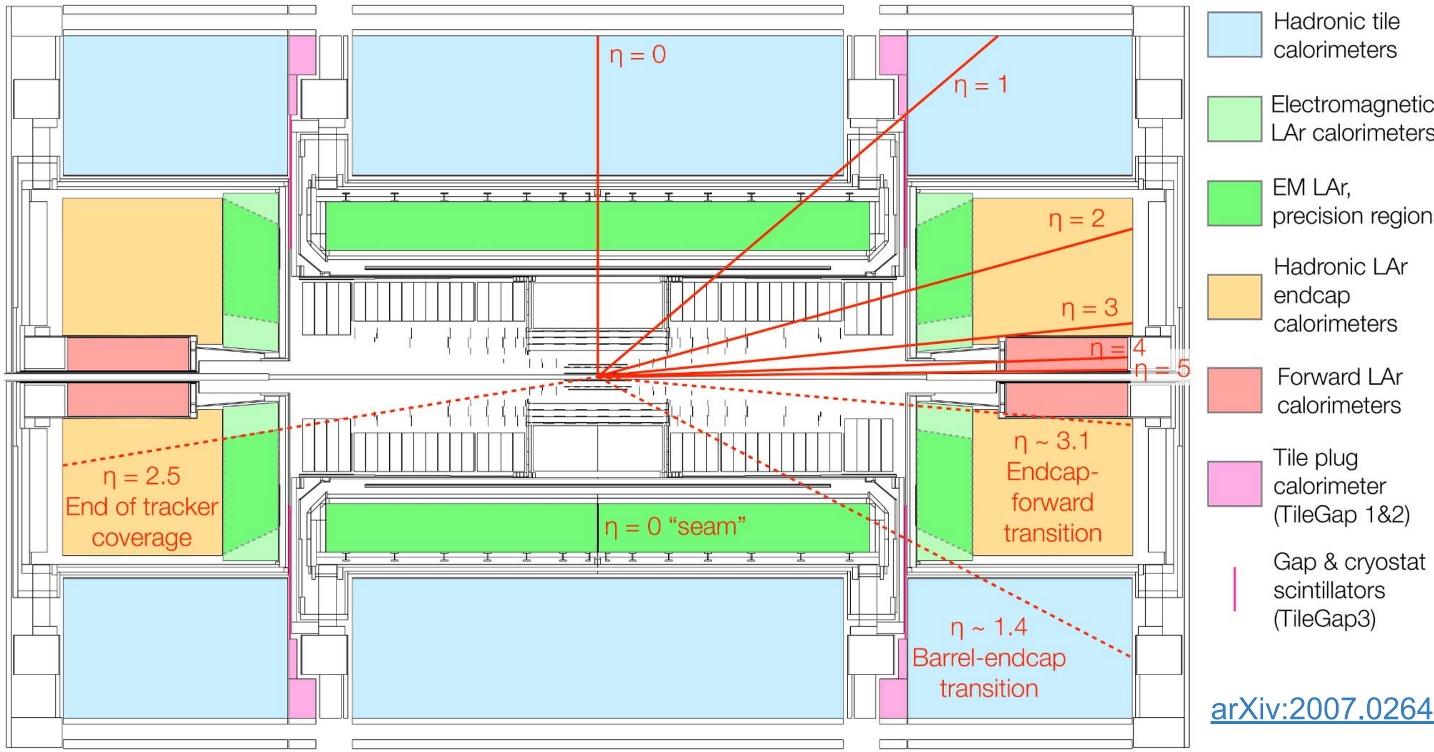
- Prototyping using AMD Vitis HLS tools (written in C → generate RTL for FPGA)





Xilinx waveform for a matrix calculation

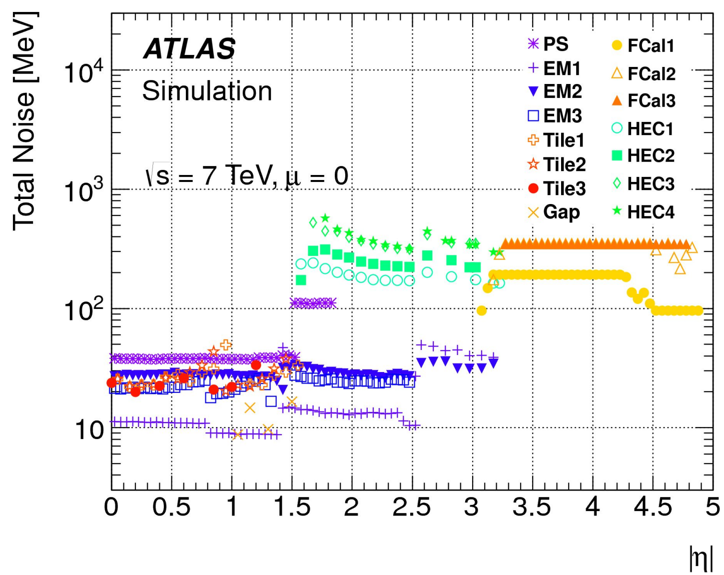
Geometry Reference



arXiv:2007.02645

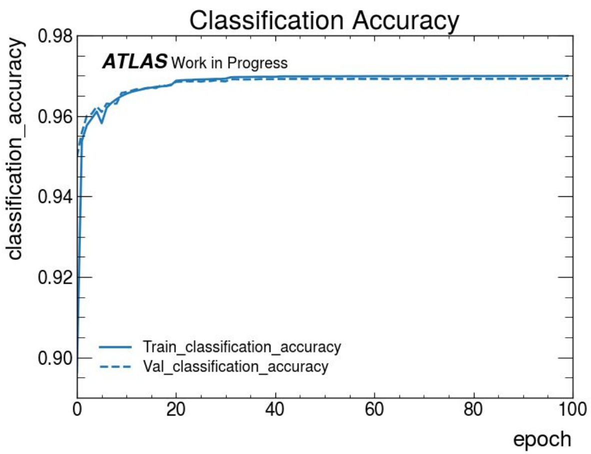
Electronic Noise in Calorimeter

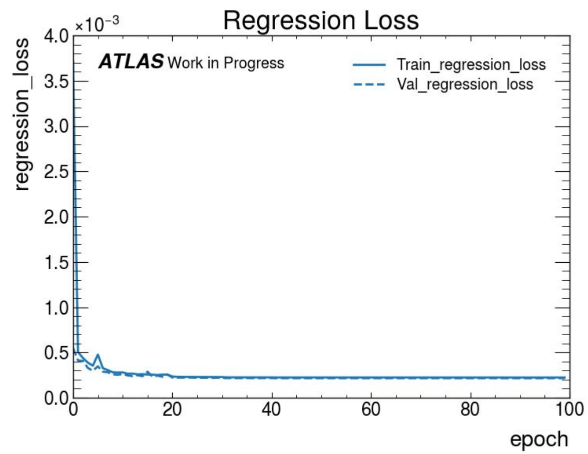
Electronic noises in the detectors are similar across all runs

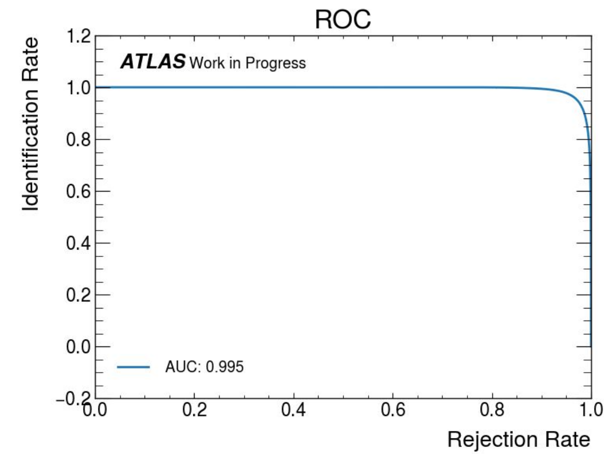


arXiv:1603.02934

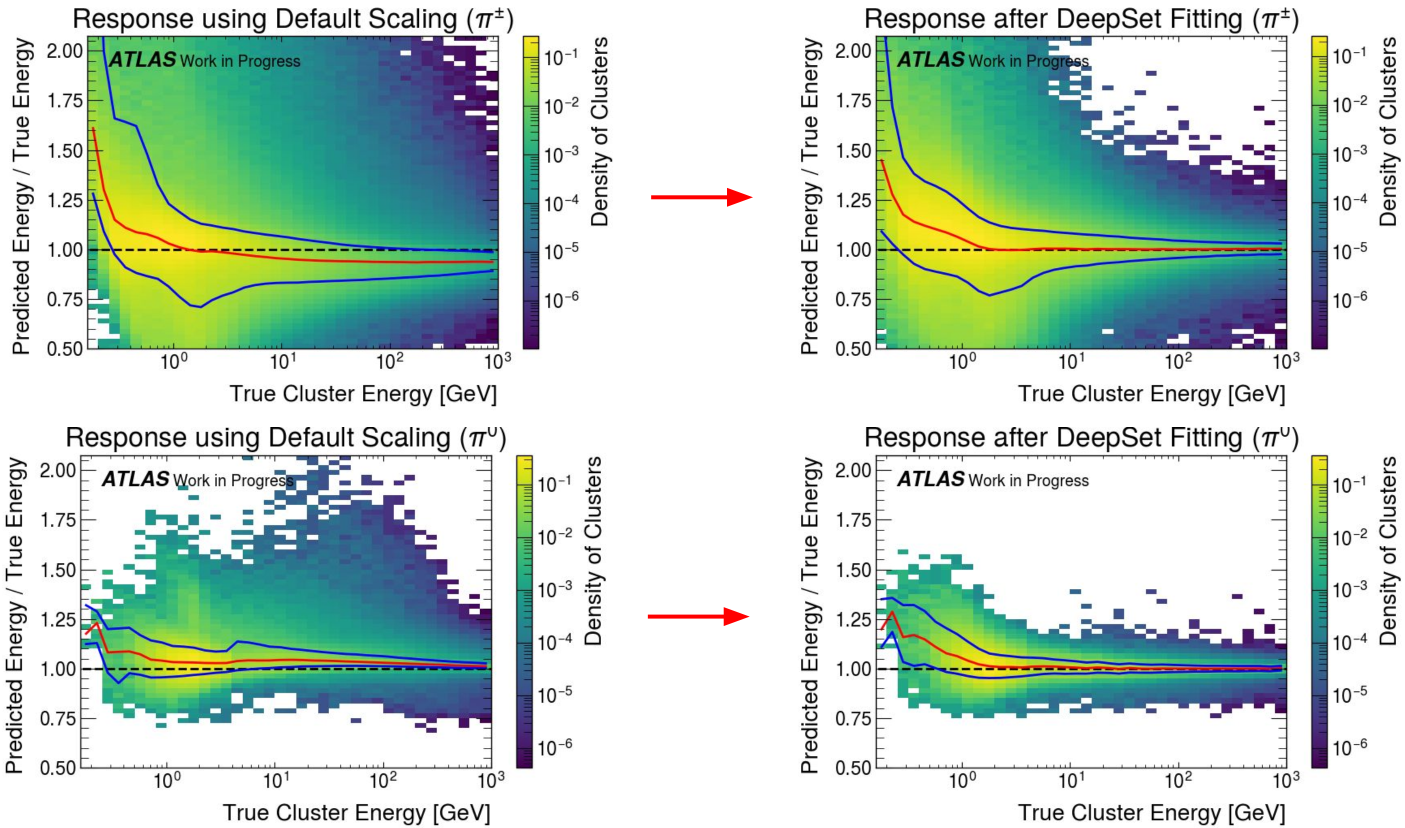
Accuracy, losses, ROC plots



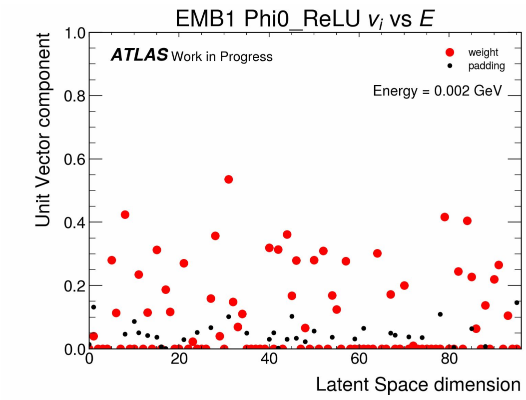


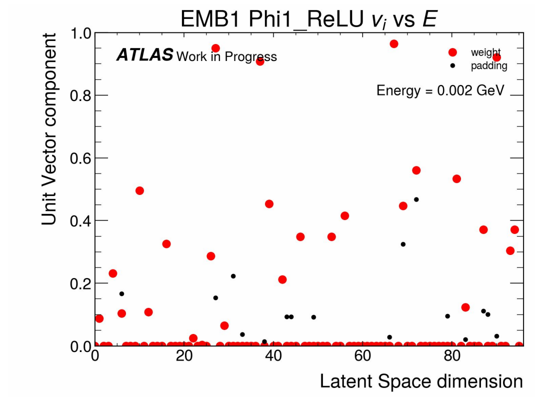


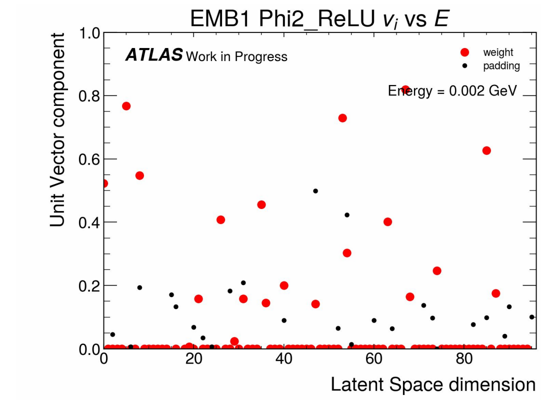
Regional Plots from Fitting

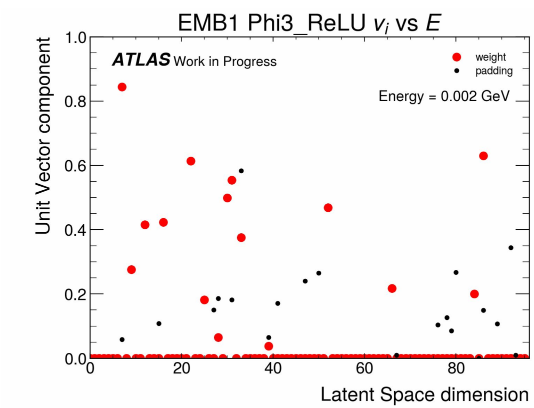


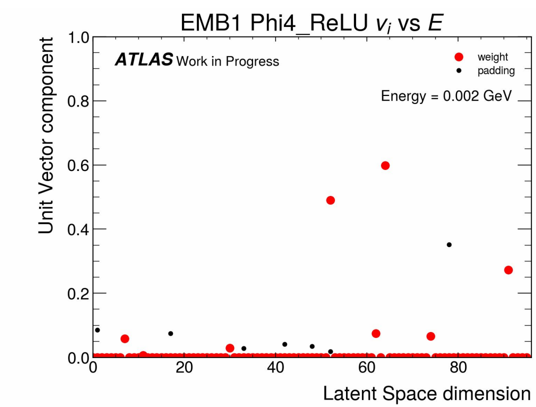
Comparison of Φ-layers unit vectors











Comparison of *F*-layers unit vectors