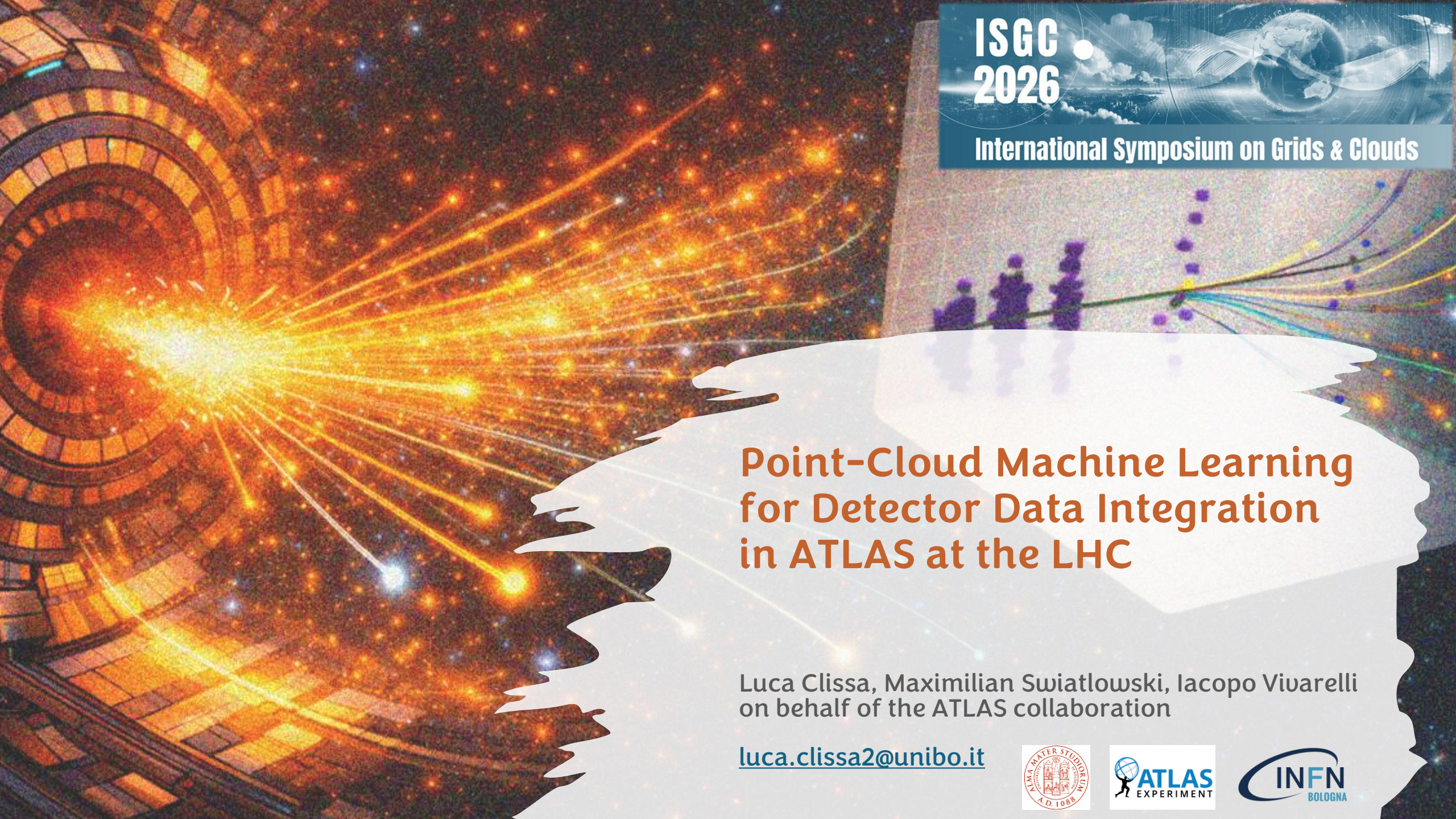


Point-Cloud Machine Learning for Detector Data Integration in ATLAS at the LHC

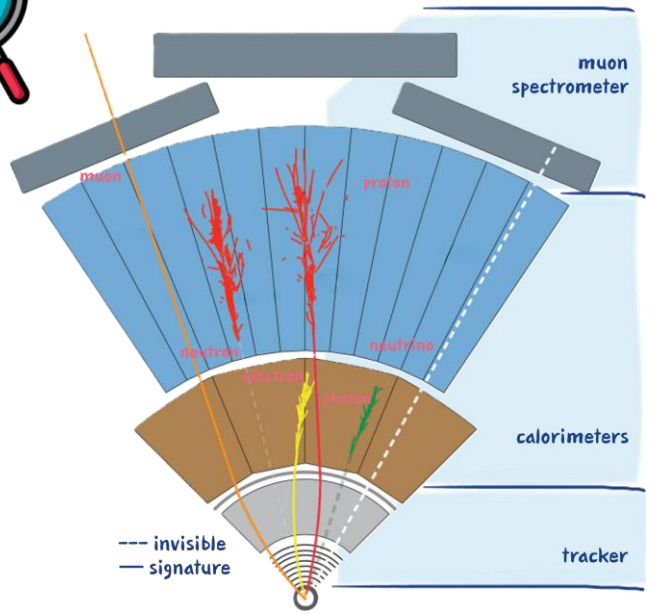
Luca Clissa, Maximilian Swiatlowski, Iacopo Vivarelli
on behalf of the ATLAS collaboration

luca.clissa2@unibo.it

Outline

- Particle flow in ATLAS
- Current approach VS our proposal
- Overall strategy:
  - model, data and loss/metrics we are using
- Lessons learned
- Highlights of preliminary results
- Conclusions and future plans

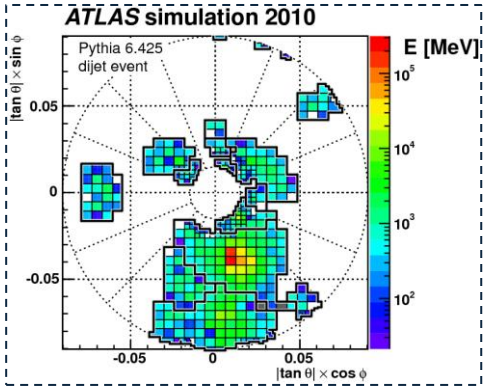
Background & motivation



- **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

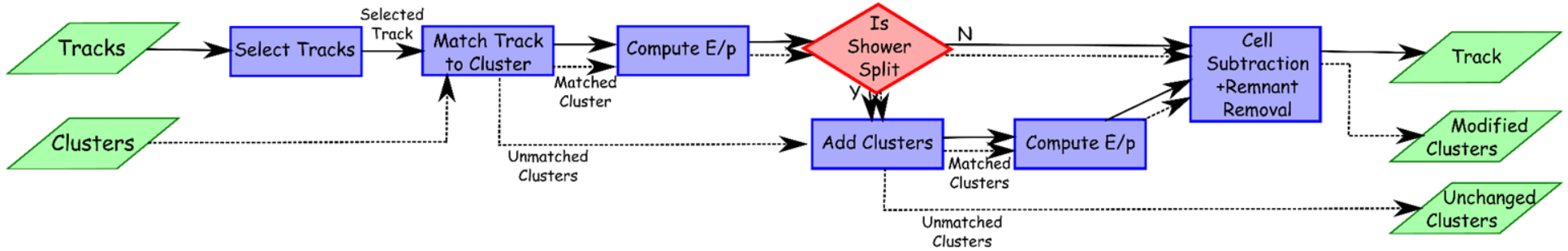
- Inputs are tracks in the inner detector and topo-clusters in calorimeter
  - topo-clusters are groups of neighbouring cells
    → useful to reconstruct showers in the calorimeter
- **Goal:** try to associate topo-clusters to tracks



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

ATLAS p-flow algorithm

- ATLAS uses rule-based selection algorithm



[Eur. Phys. J. C 77 (2017) 466]

Pros:

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

Cons:

- Track-to-topo-clusters match, not cells directly
  → energy subtraction limited to fixed cluster boundaries
- No calibration currently available → only detector measurements
- Tracker usage off above 100 GeV to avoid false matches

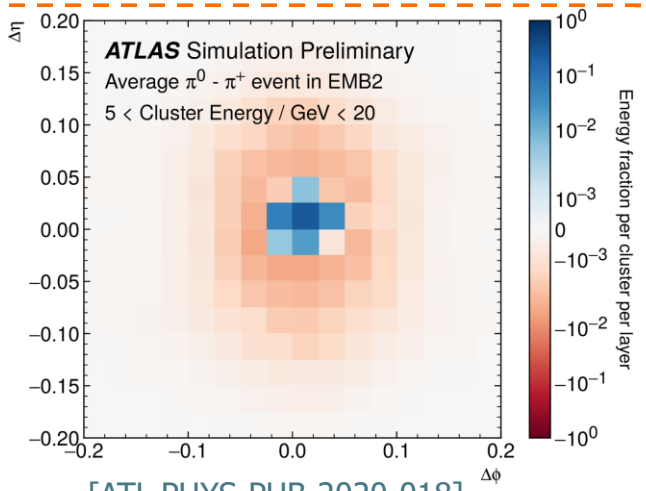
Proposal: point cloud ML for pflow

- 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, ...)

- Perform **cell-to-track association**

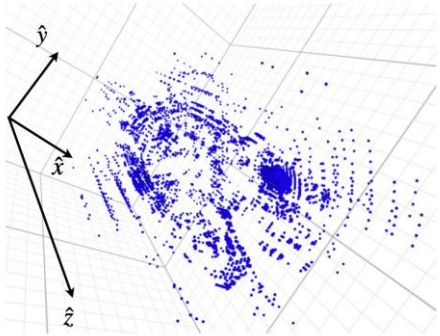
Why point cloud data?

- 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**



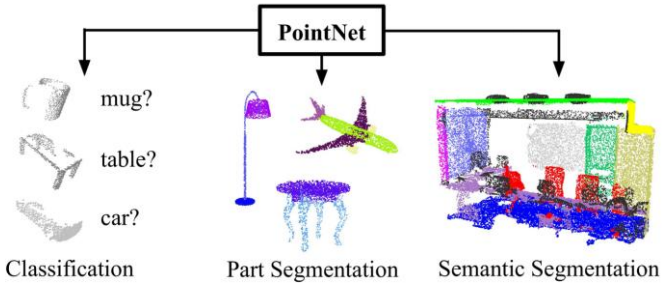
[ATL-PHYS-PUB-2020-018]

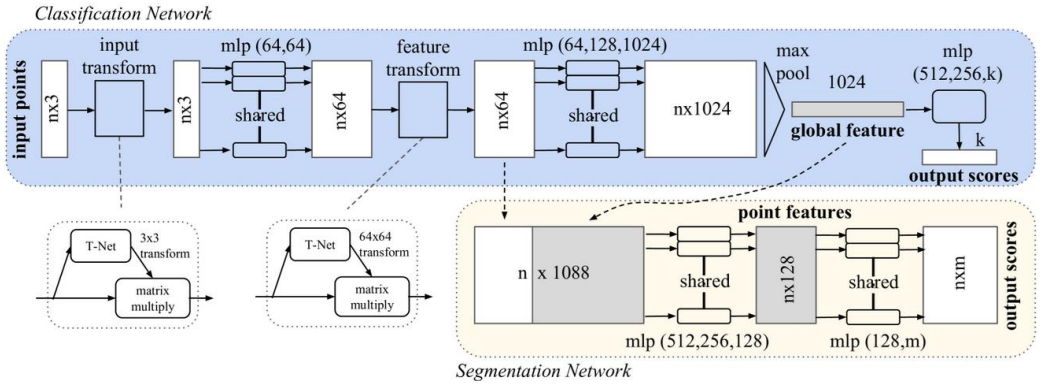
- Point cloud representation has several pros
  - represent hits as 3D points with properties
    → complex 3D shapes instead of series of images
    → features like energy, hit confidence
  - including tracker is straightforward
  - only uses actual hits
    → efficient representation



ATL-PHYS-PUB-2022-040

Methods



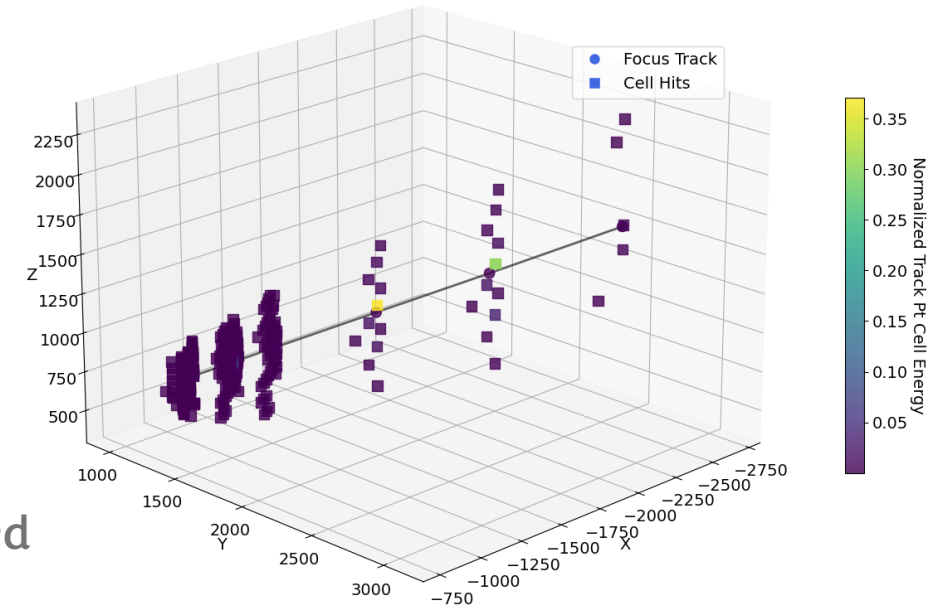


- Several learning tasks: classification, part segmentation, semantic segmentation
- permutation invariant
- transformation equivariance
- both shape classification & segmentation
- robust to data corruption → critical points

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

Problem setting

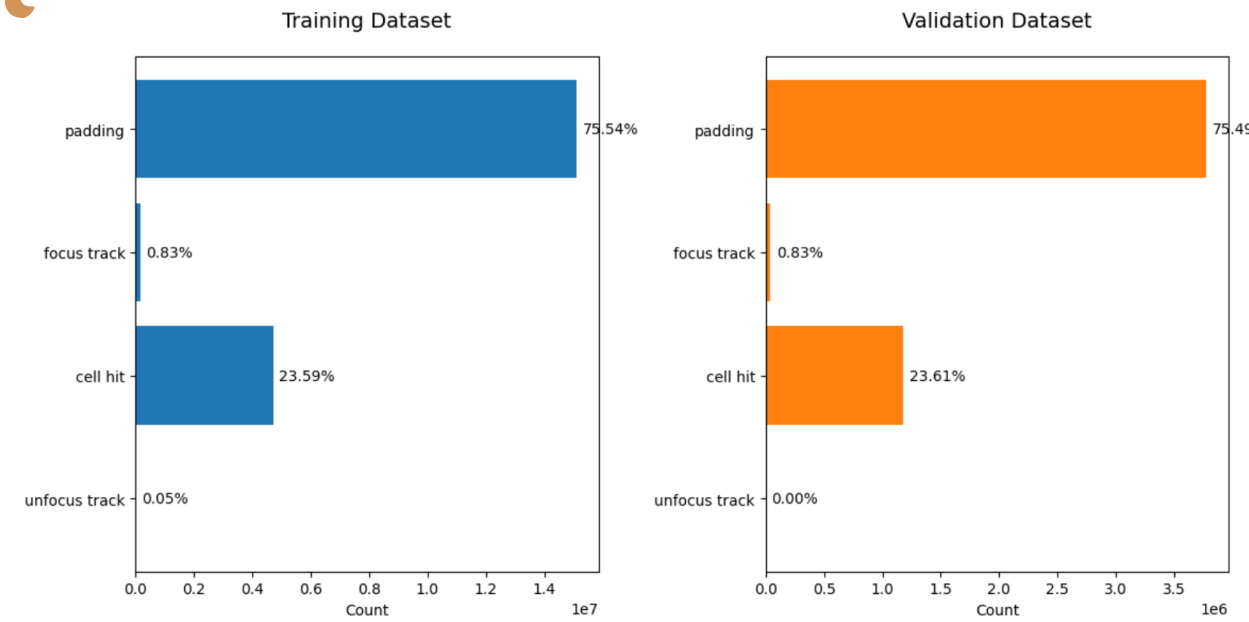
- Similarly to what is currently done in ATLAS:
  - Consider **one track at a time** (focus track)
  - Create point clouds including **all hits within ΔR=0.2** from track (tracker + calo)
  - This creates a sample → point cloud fed into model*
    *Note: **padding** points to ensure all point clouds have same dimension
  - Then change focus track and repeat until all tracks are used



- For each hit (point) we include:
  - **x, y, z** cartesian coordinates
  - **category**: point is focus track (0), cell hit (1), unfocus track (2) or padding (-1)
  - **energy**: either cell energy or track momentum (properly rescaled)
  - **distance**: distance from focus track projection

Monte Carlo dataset

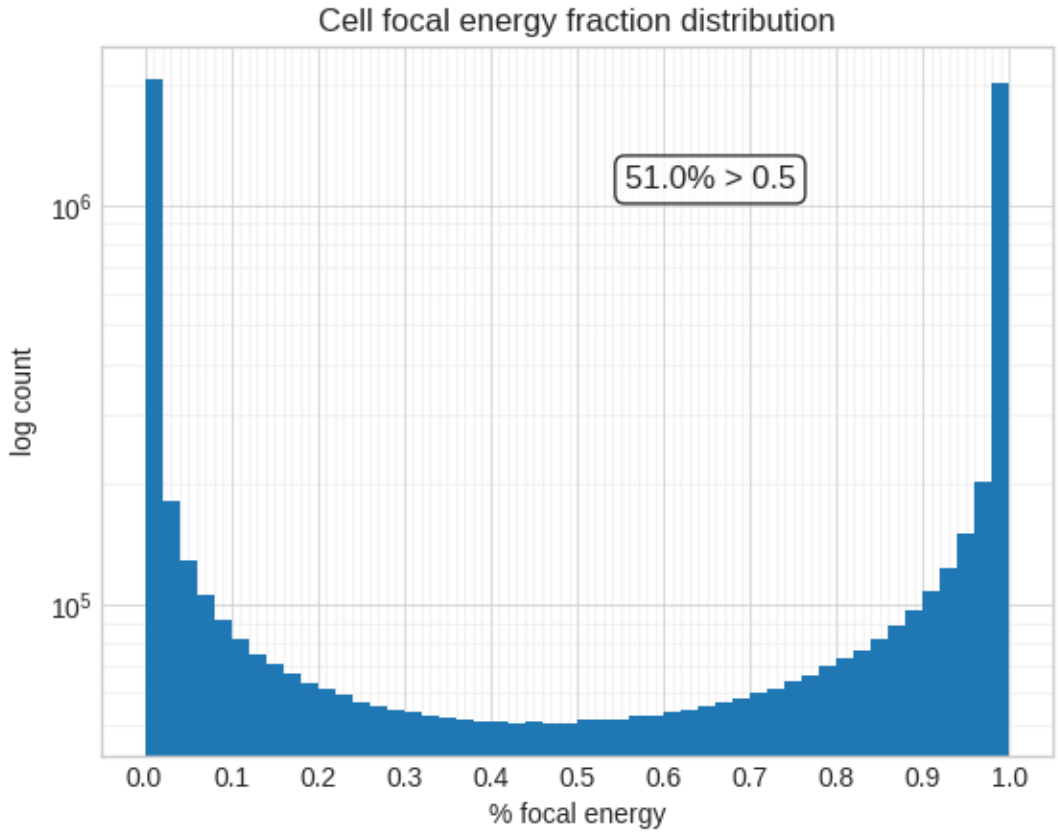
- simple benchmark decay: $\rho^{+/-} \rightarrow \pi^{+/-}\ \pi^{0}$
- Uniform pion distributions
  - azimuthal angle
  - pseudo-rapidity
  - log true energy
- Statistics:
  - **39k events**: 31k for training, 8k for validation
  - **40k tracks** (samples), mostly events with **one track**
  - **800 points per sample** (point cloud): roughly 95th percentile
  - Cell hits are the majority of relevant points
  - **Padding amount is significant**



Monte Carlo dataset (cont'd)

- For each point, we assing a label depending on:
  - Class 1 → Truth_focal_energy_pct > 0.5
  - Class 0 → otherwise

- **Notes:**
  - ***Truth_E*** > 0.5 is an imperfect proxy, but effect should be reasonably limited in this case
  - Tracker and calorimeter **energy scales** are substantially different
    → careful considerations during feature scaling



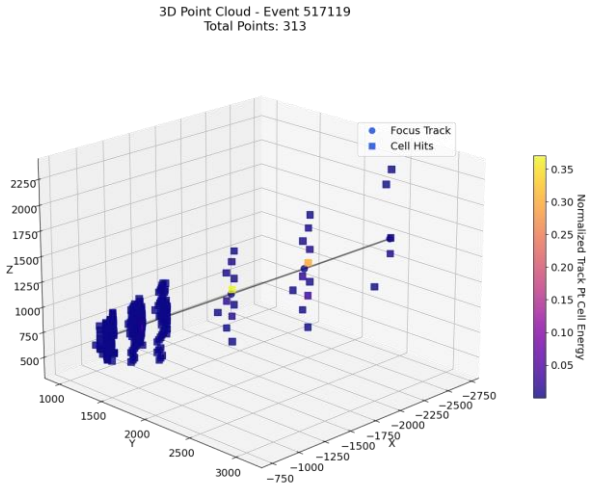
Experiments

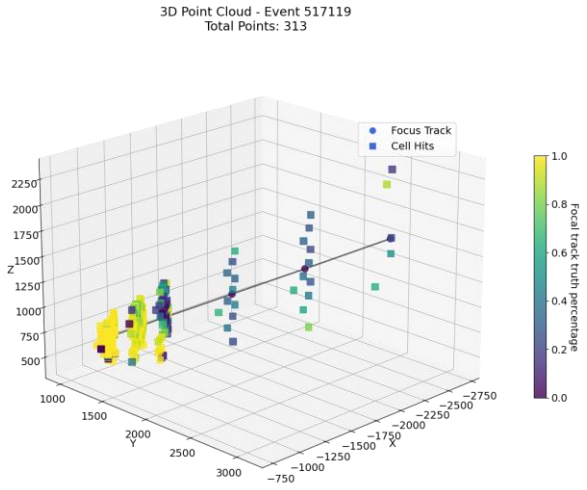
- Learning task is **semantic segmentation**:

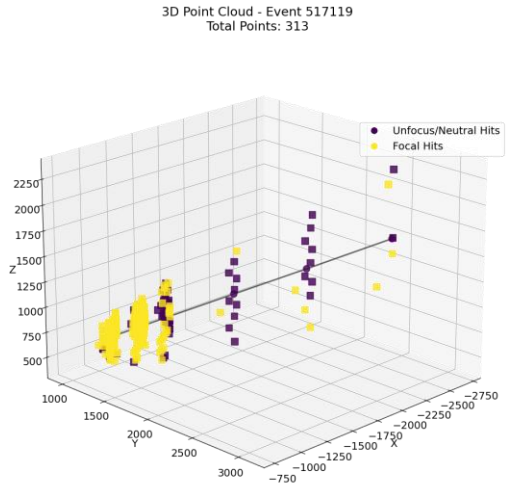
  → **binary classification at point level**

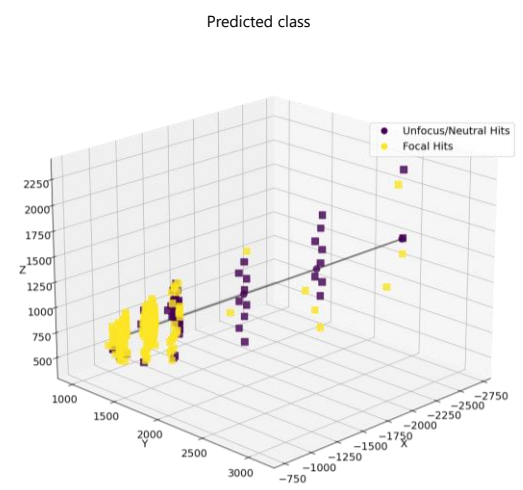
- In practice, all points fed into the network, but we mainly care about cell points

  → **cell-to-track association**





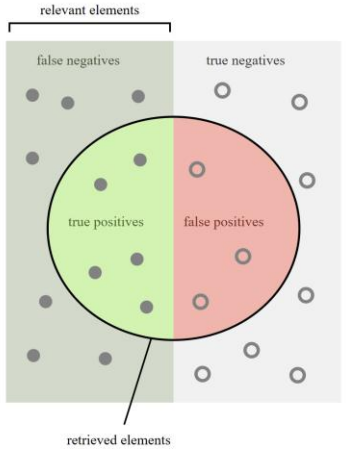




Metrics

How do we measure performance?

- Use masking for selecting only **cell points**
- Loss and metrics are weighted by **energy**



How many retrieved items are relevant?

Precision = $\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$

How many relevant items are retrieved?

Recall = $\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$

From Wikipedia
By Walber - Own work, CC BY-SA 4.0

Definitions:

- TP: true positives
- FP: false positives
- FN: false negatives

Loss function (several attempts):

- Weighted Binary Cross-Entropy (wBCE)
- Weighted Focal loss
- Weighted Dice loss

Metrics

- Accuracy: (TP+TN) / (ALL)
- Precision (purity): P = TP / (TP + FP)
- Recall (signal efficiency): R = TP / (TP + FN)
- F1 score: 2 * P * R / (P + R)

Baseline performance

How good is good?

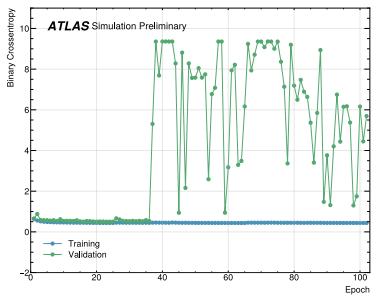
- Single metrics can be misleading out of context
- Two baselines:
  - Always 1
  - Always 0
- Naïve but nice performance with imbalanced dataset

```
Validation: 8819 samples, 69 batches
Computing data checksums for tracking...
Validation files: 50 files tracked
Data checksums saved locally:
- Validation: /home/private/jetpointnet/mlops/results/pointn
_checksums.json
8819it [10:36, 15.15it/s]2025-05-27 09:13:41.790229: W tenso
ous.cc:404] Local rendezvous is aborting with status: OUT_OF
8819it [10:36, 13.86it/s]
Dumb baseline: PREDICT ALWAYS 1
Val Loss: 0.2930
Val Acc: 0.1194
Val Acc Masked: 0.5056
Val Acc Masked + Weighted: 0.5708
        Weighted Metrics:
Val Precision: 0.5708
Val Recall: 1.0000
Val F1: 0.7268
        Unweighted Metrics:
Val Precision: 0.1194
Val Recall: 1.0000
Val F1: 0.2133
```

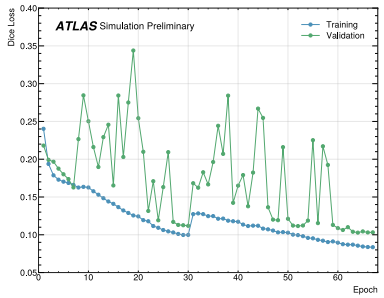
[ATL-COM-PHYS-2025-488]

Preliminary results

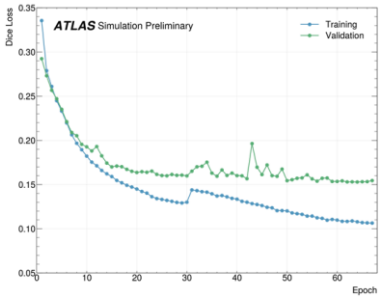
- Complex task, many challenges:
  - Padding strategy affects results
  - Unstable training
  - Class imbalance



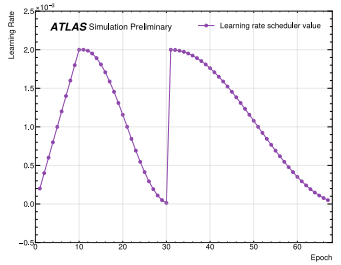
WBCE + Adam



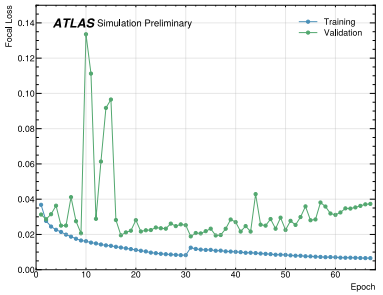
Dice + Adam-W



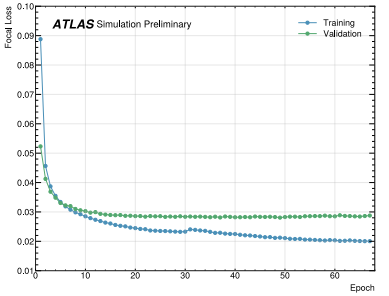
Dice + SGD



Cyclical LR w\ith warmup



Focal + Adam-W



Focal + SGD

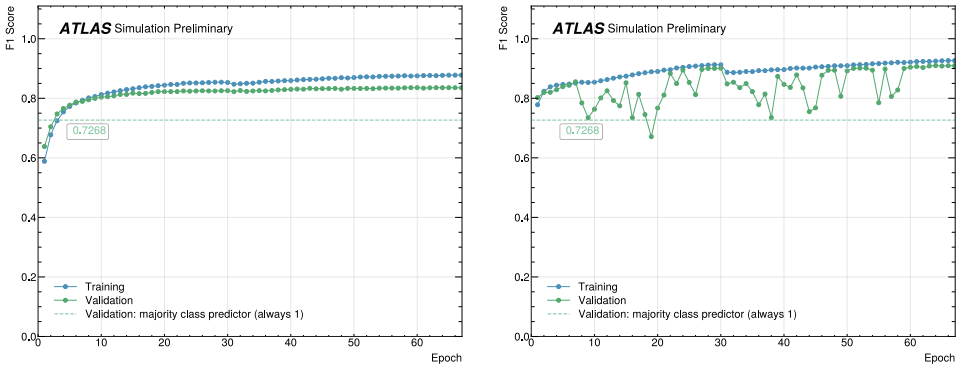
- Promising configs (more in backup):
  - Dice [10] and Focal [11] losses better than weighted BCE
  - Adam-W [12] produces better performance, also reducing instability
  - SGD [13] further stabilize training, but slower to converge
  - Cyclic learning rate [14] (with warm-up [15]) is key for convergence

Preliminary results (cont'd)

- Select loss/metrics wisely:
- Masking is crucial
- Accuracy typically misleading due class imbalance
  → F1 score more robust
- Set meaningful baselines (e.g. trivial models for majority class)

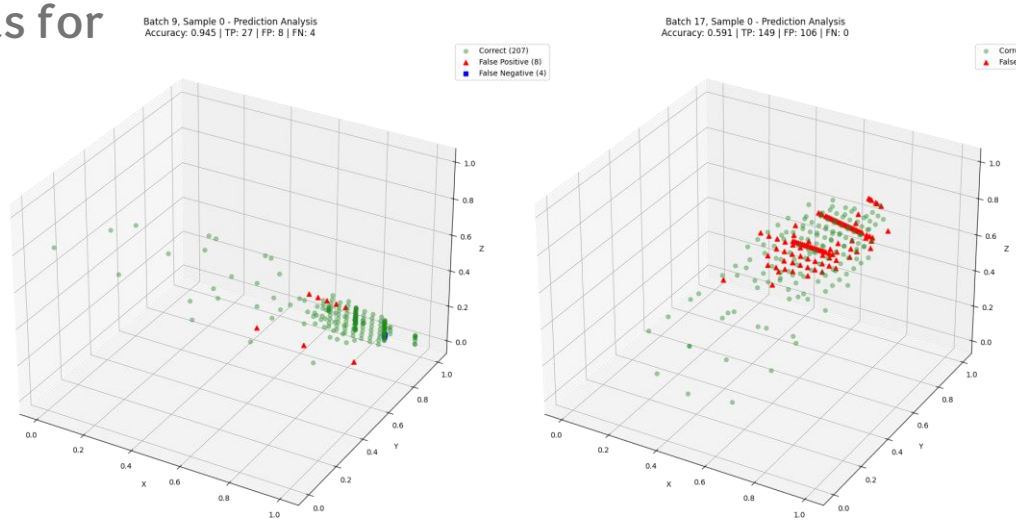
Visual inspection corroborates global metrics

Still room for improvement though



Focal + SGD

Dice + Adam-W



[ATL-COM-PHYS-2025-488]

Conclusion and next steps

more efficient and natural modelling
of pflow via **point cloud** methods

promising results on simple testbed with 1 track per event

- **BCE noisier and less effective** than Focal and Dice
- **Dice and Focal perform best** so far
- **AdamW increase performance** over SGD, but more unstable

highly non-linear development! **Key focal points:**

- Ensure **metrics** are relevant
- Double check implementations
- **Overfitting and unstable training**

Next steps:

- Extend to more complex data (jets)

  → partially explored, need more work
- Explore more models:

  → PointNet++, GTNs, MoE, PINNs

Questions?

Contacts: luca.clissa2@unibo.it

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).
[10] F. Milletari, N. Navab and S. -A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 2016, pp. 565-571, doi: 10.1109/3DV.2016.79.
[11] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.
[12] Clissa Luca, "Towards Machine-Learning Particle Flow with the ATLAS Detector at the LHC," Proceedings of 27th International Conference on Computing in High Energy & Nuclear Physics, Kraków, Poland, 19 - 25 Oct 2024.

Backup

Focal loss

- Slight variation of BCE:

where:

$$LOSS_{FOCAL}(\hat{p_t}) = \alpha_t(1-p_t)^{\gamma}\ln(p_t)$$

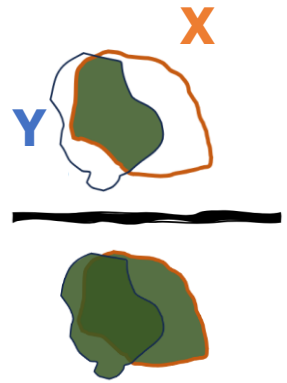
$$p_t = \begin{cases} \hat{p} & se\ y = 1 \\ 1-\hat{p} & altrimenti \end{cases} \quad \text{and} \quad \begin{cases} \alpha_t & peso\ classe\ t \\ \gamma & penalty \end{cases}$$

- Less weight to «easy data», more focus on difficult examples
- This mechanism helps mitigating issues with imbalanced datasets

Dice loss

- dice coefficient is a measure of "similarity"

$$DICE = \frac{2\,|X \cap Y|}{|X| + |Y|}$$



- dice loss $LOSS_{DICE}(y, \hat{p}) = 1 - \frac{2\,y\hat{p} + \epsilon}{y + \hat{p} + \epsilon}$

- Specific for segmentation tasks

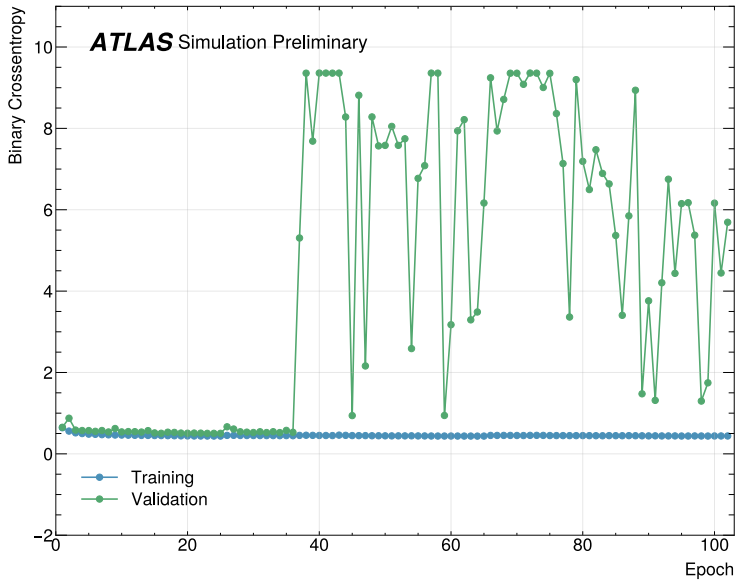
0: perfect overlap

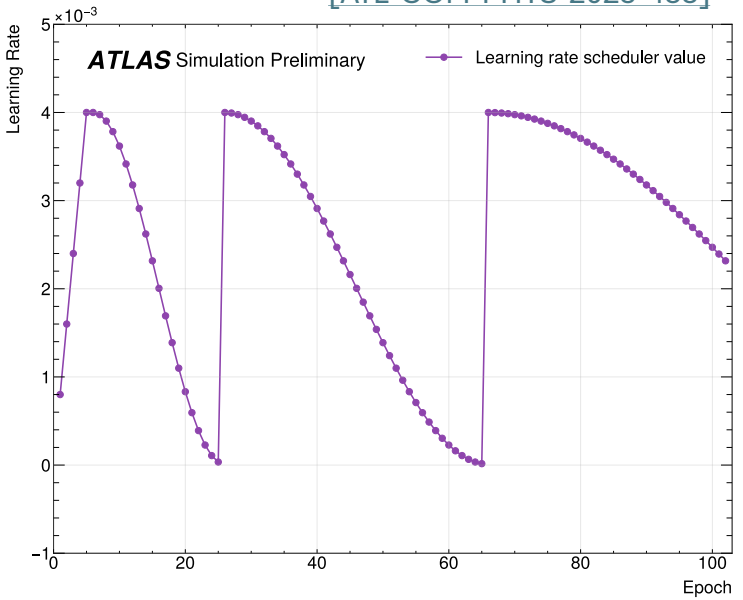
1: no overlap at all

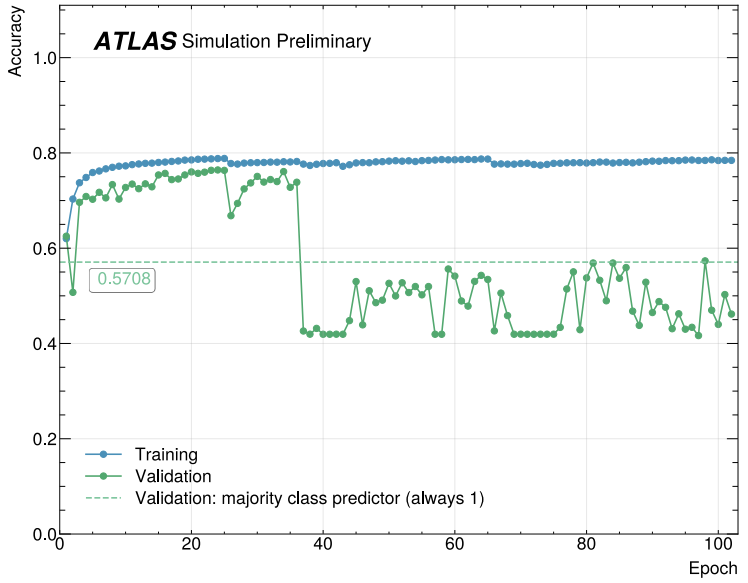
[ATL-COM-PHYS-2025-488]

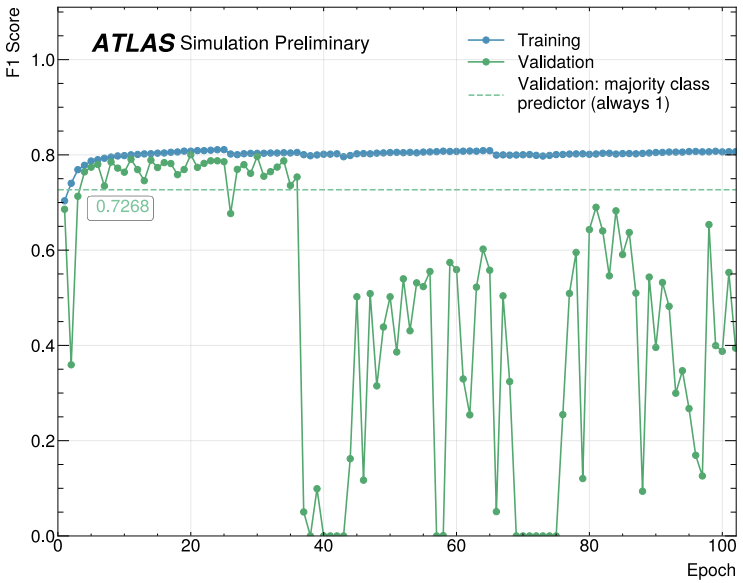
WBCE + adam + cosine annealing

- Very instable
- Training diverge
- Although initial metrics are satisfying, comparison with trivial baseline suggest model is just learning to predict majority class





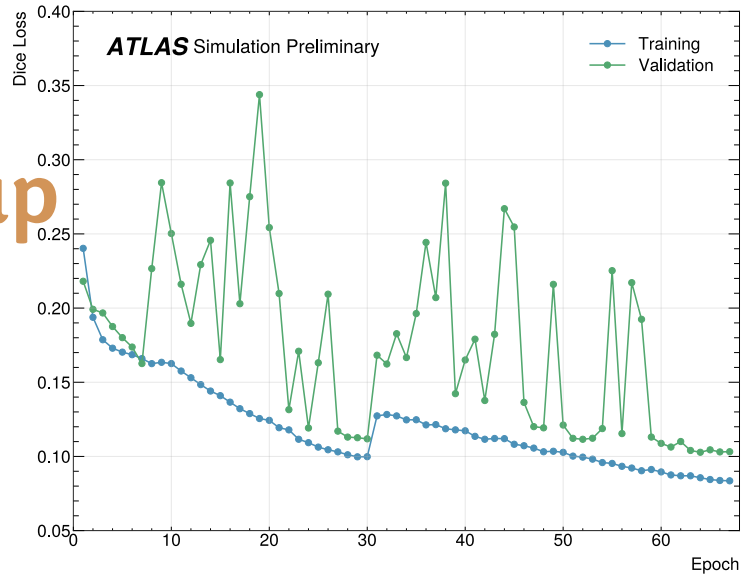


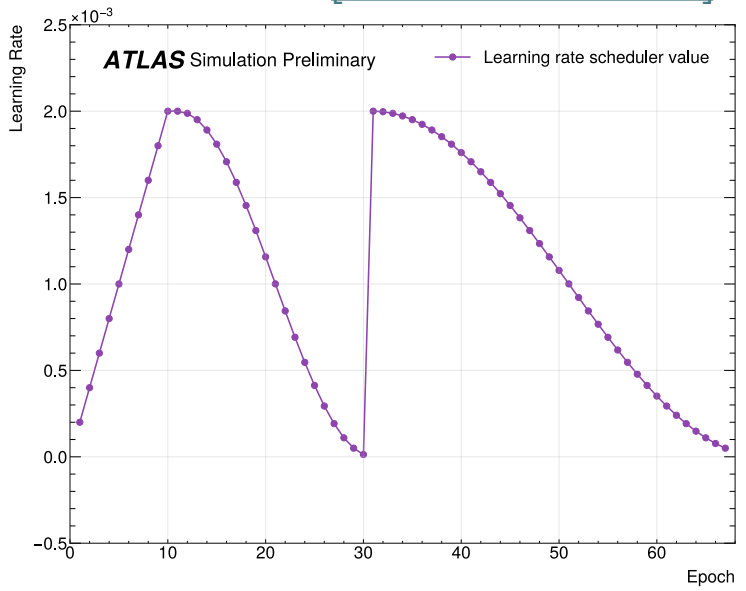


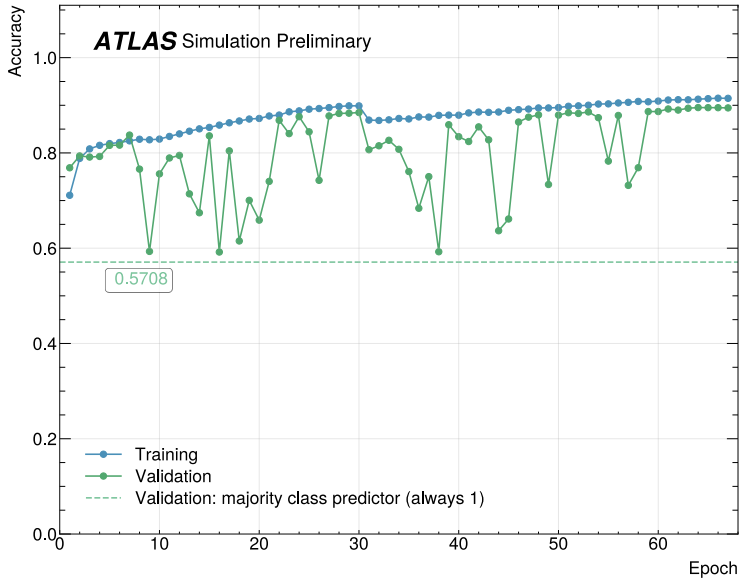
[ATL-COM-PHYS-2025-488]

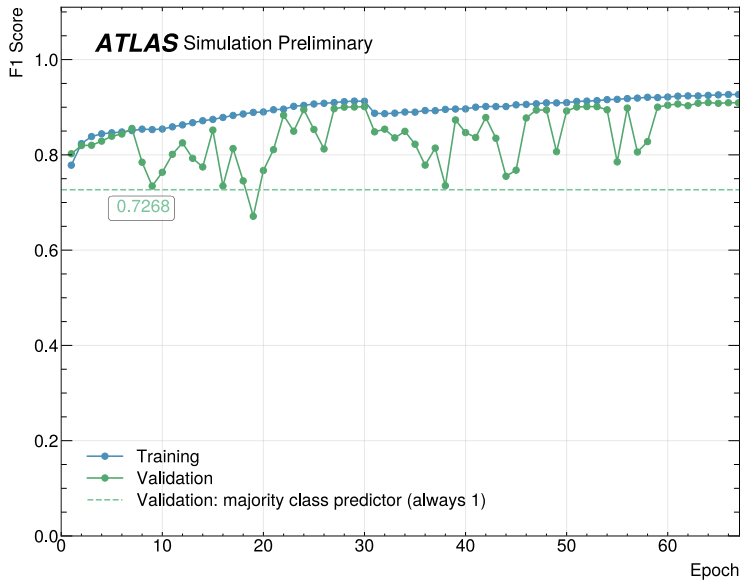
Dice + adamW + cyclical LR with warmup

- More stable, although high variability in validation curves
- Sound training curves suggest little overfitting and potential to still improve
- F1 score close to 90%, much better than trivial baseline





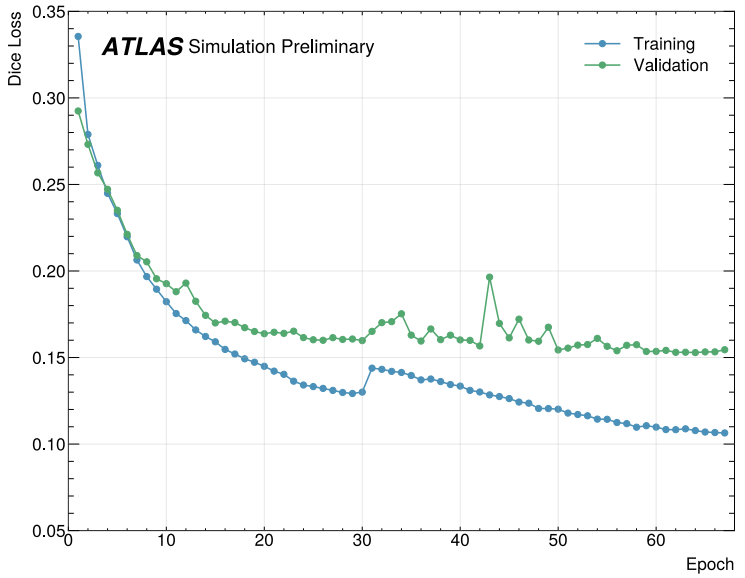


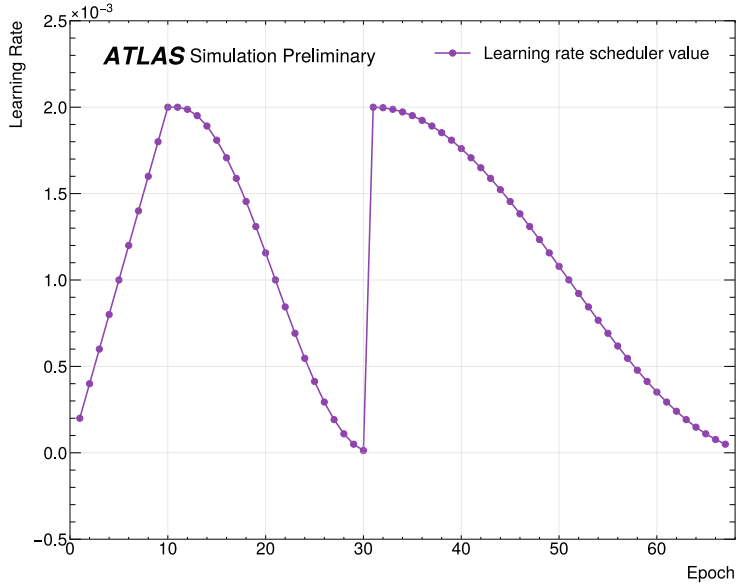


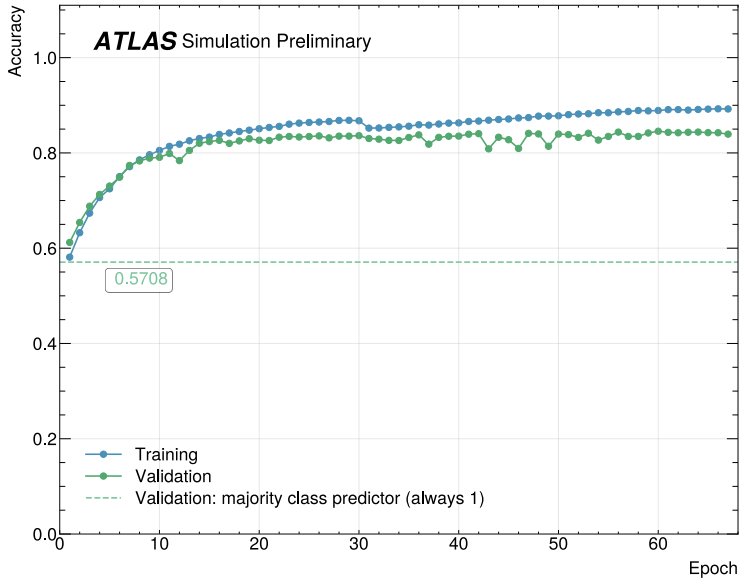
[ATL-COM-PHYS-2025-488]

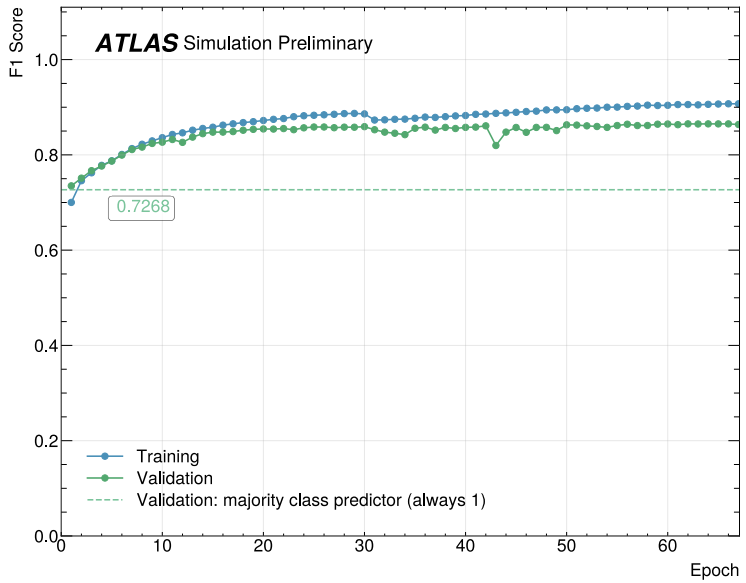
Dice + SGD + cyclical LR with warmup

- SGD stabilizes training, even validation curves are smoother
- Wider training/validation gap
- Flatten validation improvement at the end of training
- F1 score close to 85%, much better than trivial baseline but slightly worse than adamW results





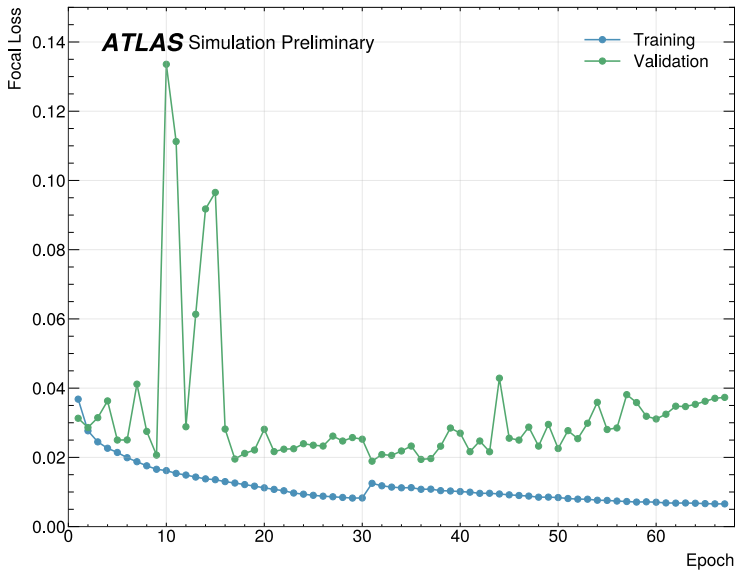


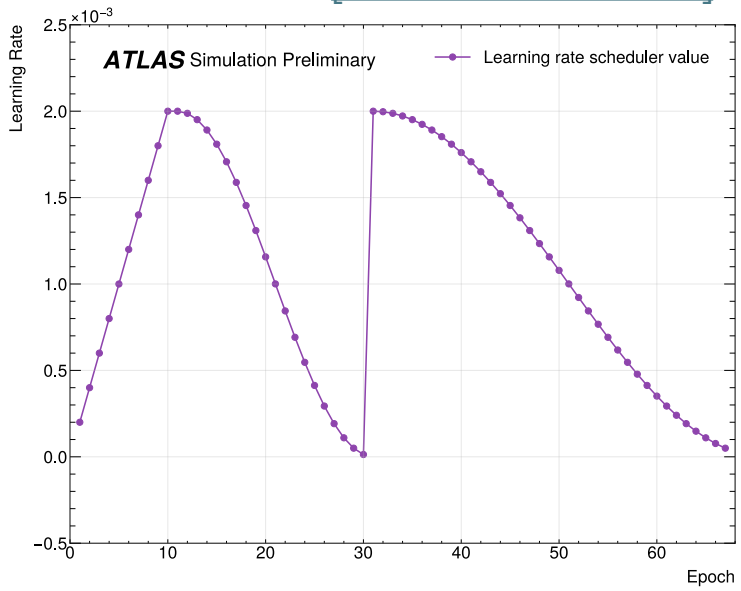


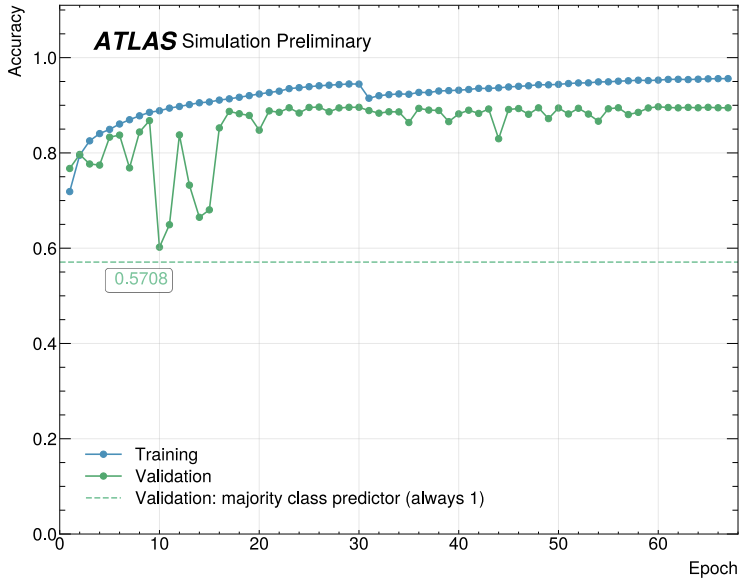
[ATL-COM-PHYS-2025-488]

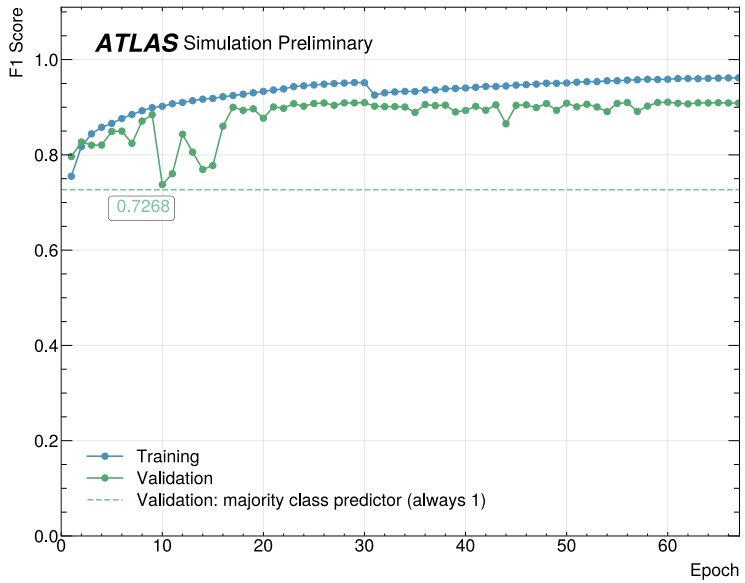
Focal + adamW + cyclical LR with warmup

- More stable, although high variability in validation curves
- Increasing overfitting in final epochs
- F1 score close to 90%, much better than trivial baseline
- Comparable to Dice alternative (just slightly better)









[ATL-COM-PHYS-2025-488]

Focal + SGD + cyclical LR with warmup

- SGD stabilizes training, even validation curves are smoother
- Widening training/validation gap at end of training
- Flatten validation improvement in the end
- F1 score close to 90%, much better than trivial baseline and close to best performance obtained