

Metric Learning for Graph Neural Network-Based Track Reconstruction at the ATLAS Event Filter

Giulia Fazzino¹ on behalf of the ATLAS Collaboration

1) Physikalisches Institut Heidelberg

EuCAIFCon2026 – 26.08.2026

giulia.fazzino@cern.ch



FSP
Erforschung
von Universum
und Materie



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

IMPRS
for Precision Tests of
Fundamental Symmetries
INTERNATIONAL MAX PLANCK
RESEARCH SCHOOL



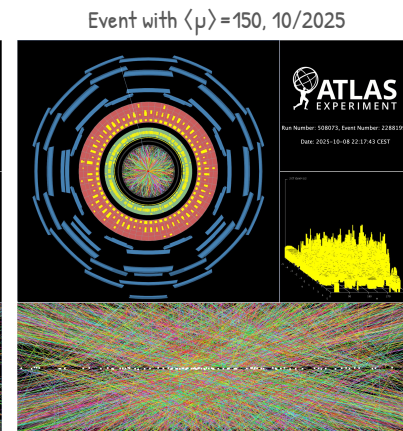
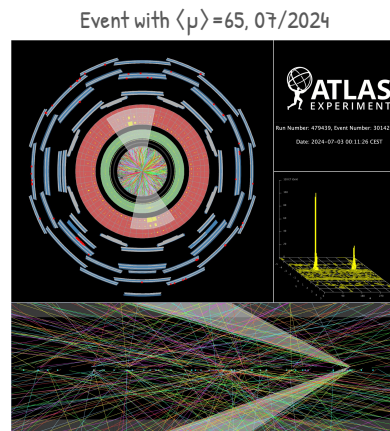
Gefördert durch:



Bundesministerium
für Forschung, Technologie
und Raumfahrt

Motivation

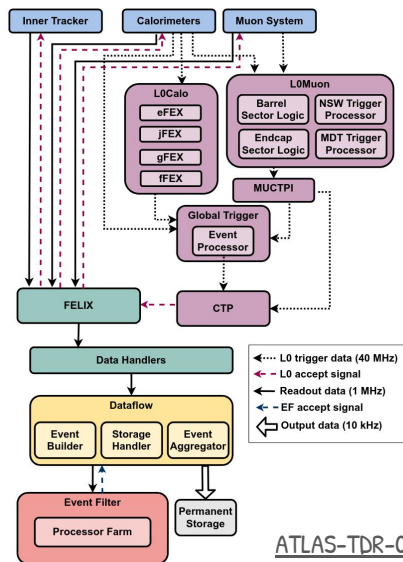
- 2030: HL-LHC, significant increase in collider luminosity
- Pile up in ATLAS expected to increase, reaching up to $\langle\mu\rangle=200$



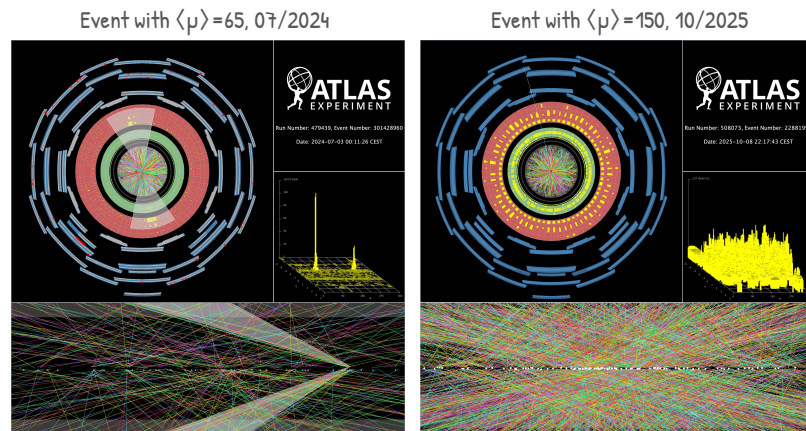
ATLAS Public Event Displays

Motivation

- 2030: HL-LHC, significant increase in collider luminosity
- Pile up in ATLAS expected to increase, reaching up to $\langle\mu\rangle=200$



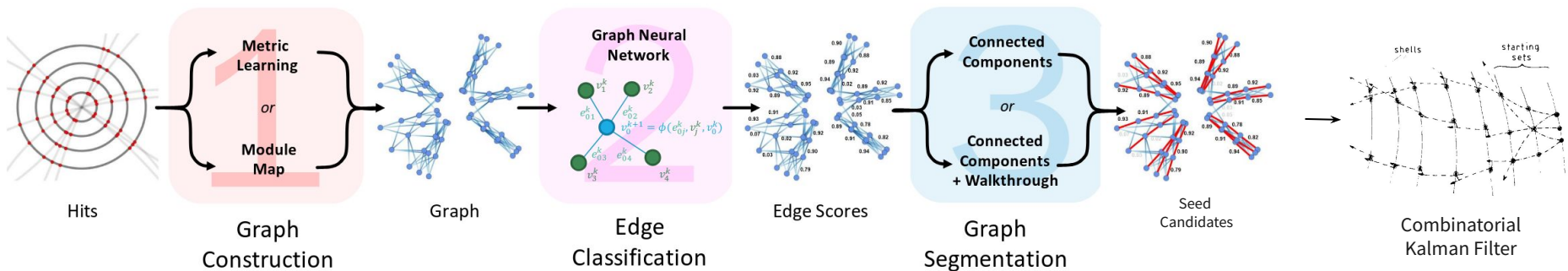
ATLAS-TDR-029-ADD-1



ATLAS Public Event Displays

- ATLAS upgrades relevant to this talk
 - New all-silicon Inner Tracker (ITk)
 - New Trigger and Data Acquisition System architecture
- Software trigger (**Event Filter**)
 - Track reconstruction at 1 MHz
 - R&D process concluded last year to investigate **new tracking algorithms** and use of **hardware accelerators**
 - Heterogeneous farm based on CPUs and GPUs, balancing technology risks and opportunities

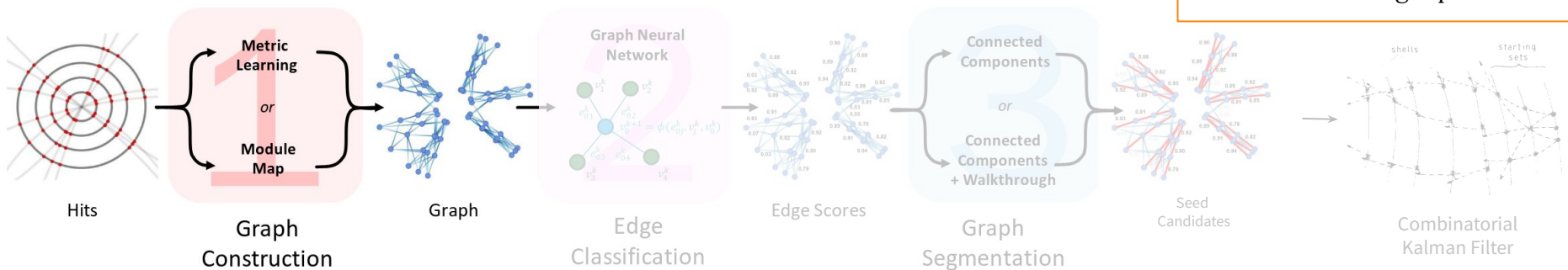
Building Seeds with Graph Neural Networks



Adapted from [ATL-ITK-PROC-2022-006](#)

Progressive track recognition with a Kalman-like fitting procedure

Graph Construction



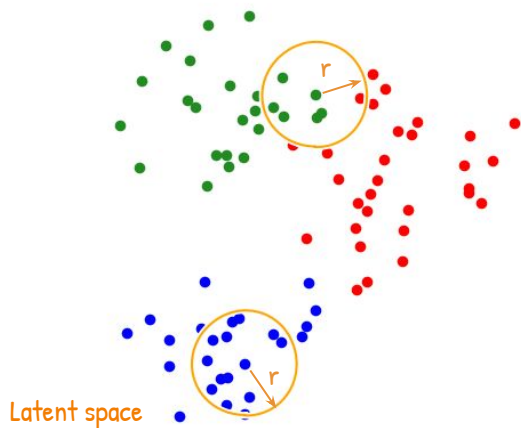
Adapted from [ATL-ITK-PROC-2022-006](#)

[Progressive track recognition with a Kalman-like fitting procedure](#)

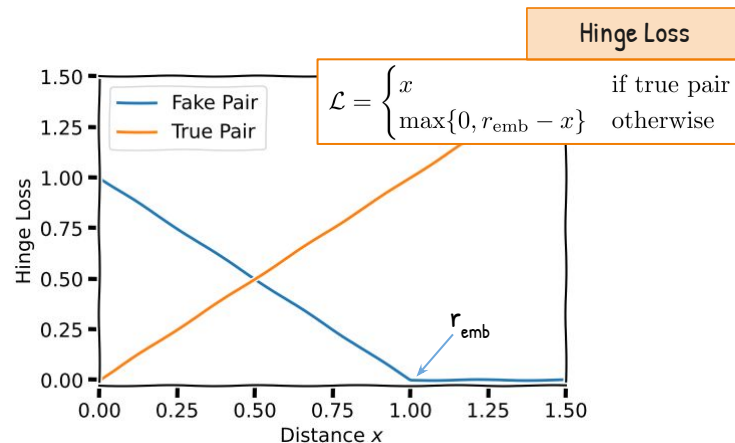
- Hits in the **ITk pixel detector** are connected to form a graph
- High pile up → can't build every possible hit pair connection
- **Goal**: keep as many true edges as possible, minimize fake edges

Metric Learning Outline

1. A Multi Layer Perceptron **embeds hits** in a 4-dimensional latent feature space
 - Embed successive hits of the same particle close to each other

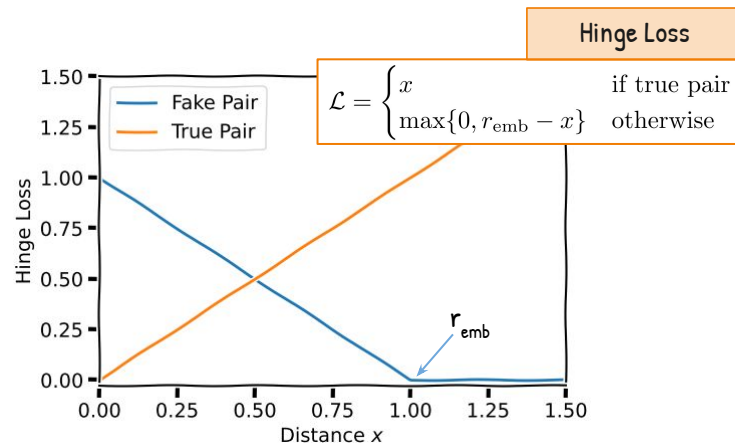
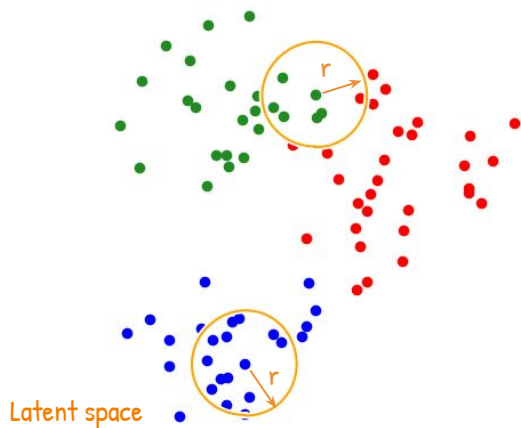


2. Edges are built connecting hits within a distance r (clustering radius) in the latent space



Metric Learning Outline

1. A Multi Layer Perceptron **embeds hits** in a 4-dimensional latent feature space
 - Embed successive hits of the same particle close to each other



2. Edges are built connecting hits within a distance r (clustering radius) in the latent space

Resource intensive on hardware

- Floating point arithmetic (FPGAs)
- Computing 4D distance of each pair of hits

Hardware Targeted Optimizations

Resource intensive on hardware

- Floating point arithmetic (FPGAs) —→ **Quantization Aware Training** with fixed bit width (6 bit)
- Computing 4D distance of each pair of hits

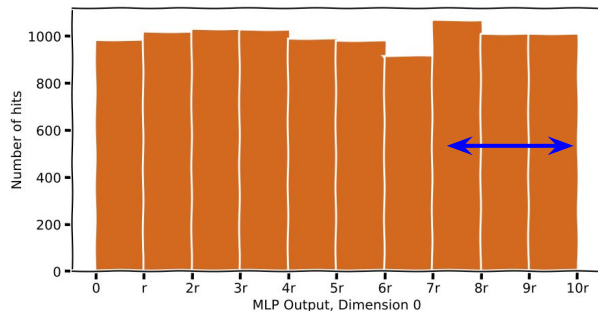
Hardware Targeted Optimizations

Resource intensive on hardware

- Floating point arithmetic (FPGAs) —————→ **Quantization Aware Training** with fixed bit width (6 bit)
- Computing 4D distance of each pair of hits

Train the MLP to produce a **uniform distribution** in one output dimension

- Compute distance only for hits in neighbouring bins



Already too far: no need to compute 4D distance

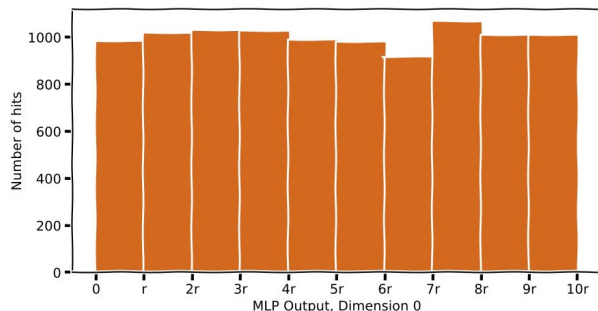
Hardware Targeted Optimizations

Resource intensive on hardware

- Floating point arithmetic (FPGAs) —→ **Quantization Aware Training** with fixed bit width (6 bit)
- Computing 4D distance of each pair of hits

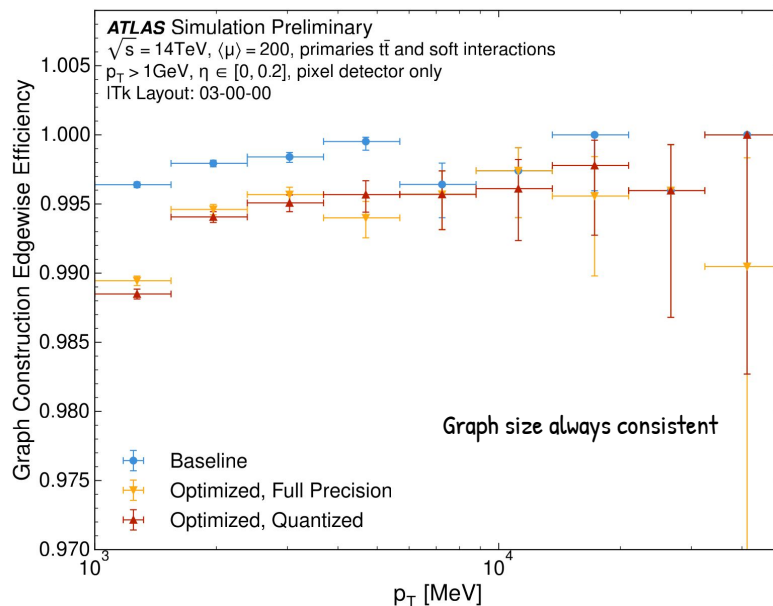
Train the MLP to produce a **uniform distribution** in one output dimension

- Compute distance only for hits in neighbouring bins



Edgewise Efficiency

Fraction of true edges present in the graph over all true edges



Metric Learning run in the ACORN framework

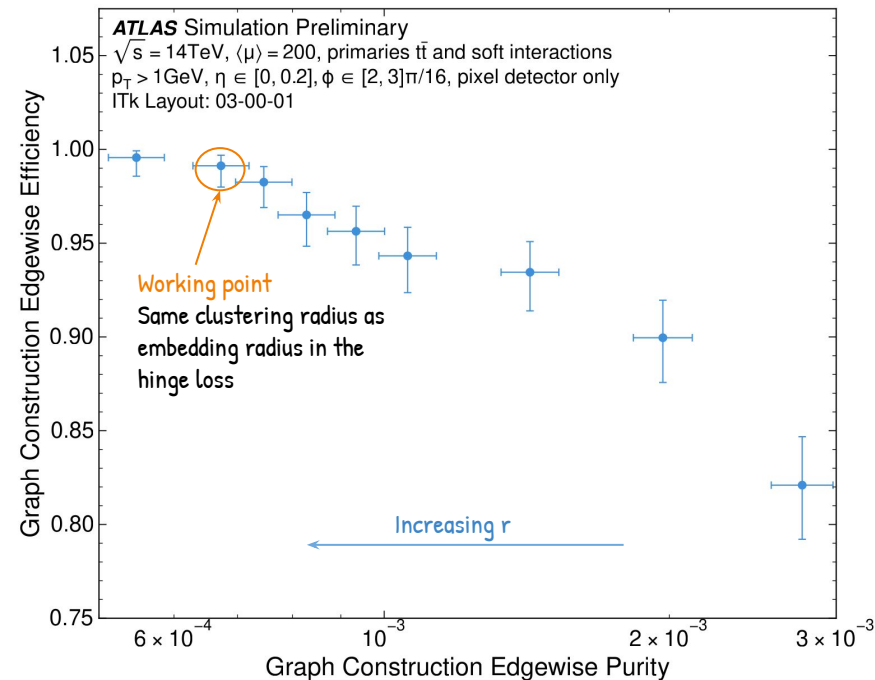
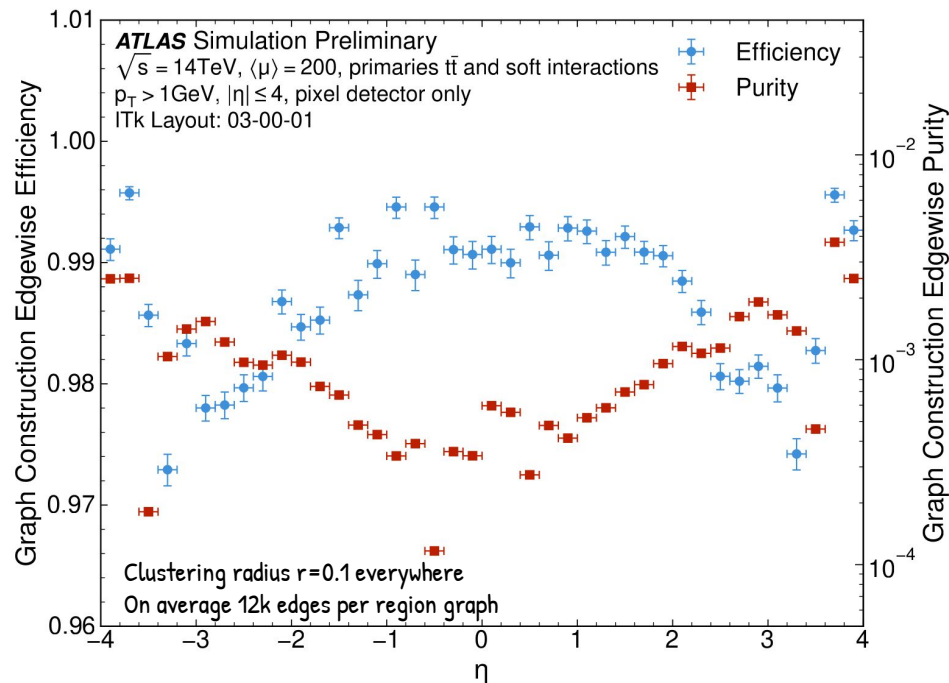
Model training: PyTorch

Quantization: Brevitas

Metric Learning Performance

Edgewise Purity

Fraction of true edges in the graph over all edges in the graph



Metric learning run in the ATLAS Athena software framework
 Full precision, optimized regional models exported to onnx



Summary and Outlook

- Graph Neural Network-based pattern reconstruction can be used to build track seeds in the high pile-up environment of ATLAS during HL-LHC
- Metric Learning produces highly efficient graphs while maintaining a reasonable graph size
- Hardware-driven optimizations (targeting FPGAs, beneficial for GPUs too) can be employed without hindering method performance
- Further improvements by fine-tuning working points for each regional network are possible
- Work is ongoing towards publishing advanced seeding on FPGA studies

Thank you for your attention!

Backup

$$\mathcal{L} = \mathcal{L}_{\text{emb}} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{\text{uni}} \cdot \mathcal{L}_{\text{uni}}$$

$\mathcal{L}_{\text{emb}} = \begin{cases} x & \text{if true pair} \\ \max\{0, r_{\text{emb}} - x\} & \text{otherwise} \end{cases}$

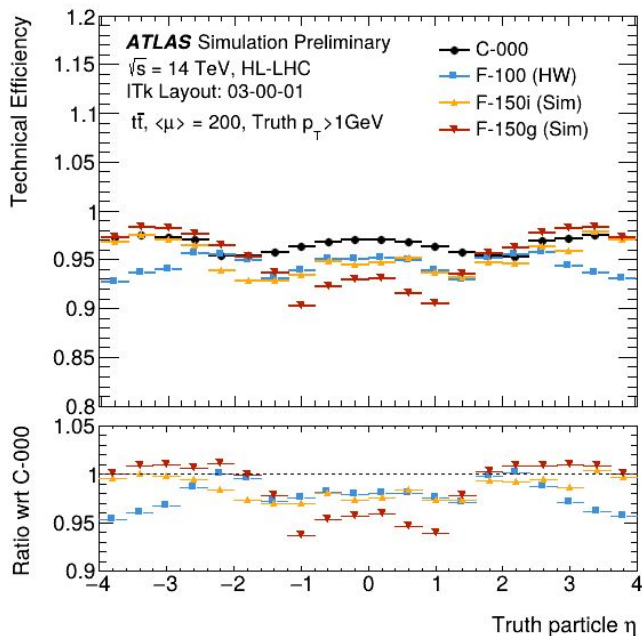
$\mathcal{L}_{L1} = \sum_{p=1}^P |\theta_p|$

$\mathcal{L}_{\text{uni}} = \max_x |F_{\text{emp}}(x) - F_{\text{uni}}(x)| + \lambda_{\text{spread}} \frac{1}{x_n - x_1}$

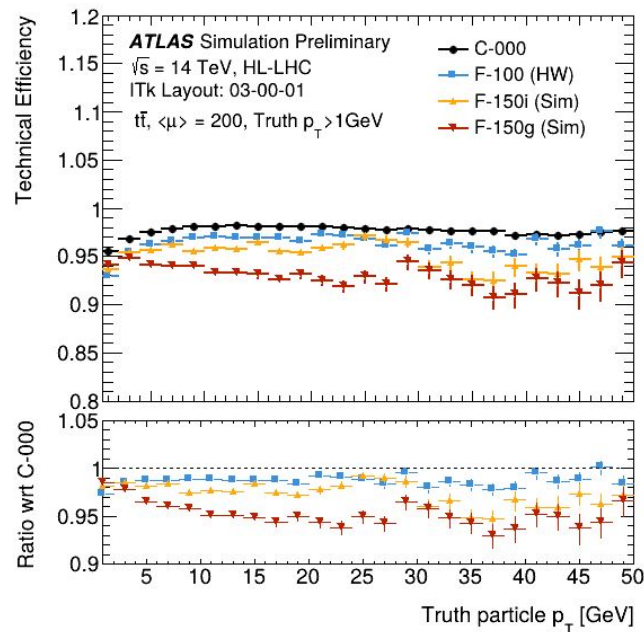
λ_{L1}	10^{-6}
λ_{uni}	0.01
λ_{spread}	0.3

- 2 hidden layers of 16 nodes, 3 input features (hit r, hit z, hit phi), 4 outputs -> 440 parameters
- For quantized model: all parameters quantized to 6 bits, input 2 integer bits, 9 fractional bits
- ReLU activation function, batch norm

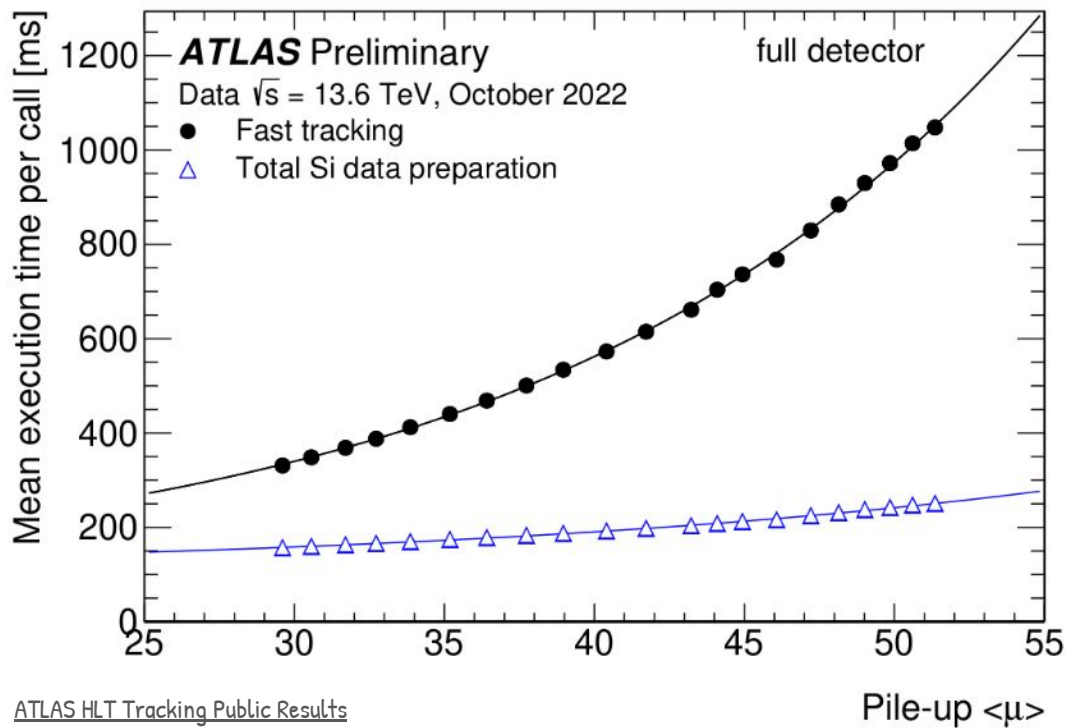
Backup - (GNN Seeding - with Module Map)



ATL-COM-DAQ-2026-047



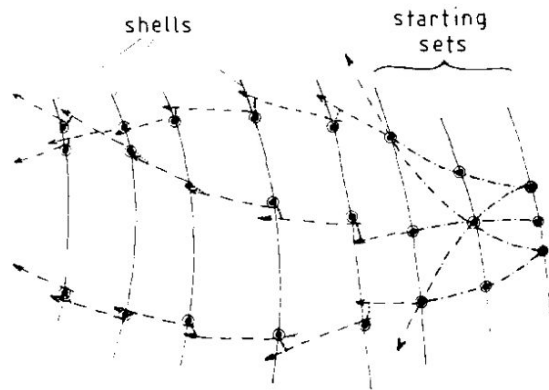
Backup



Backup

Combinatorial Kalman Filter

- Start from a track seed (small set of close measurements)
- Compute a prediction of the track extension in the next layer
- If there is a measurement compatible with the prediction, add it to the set
- Reiterate until no more hits pass this selection



Progressive track recognition with a Kalman-like fitting procedure

Backup

