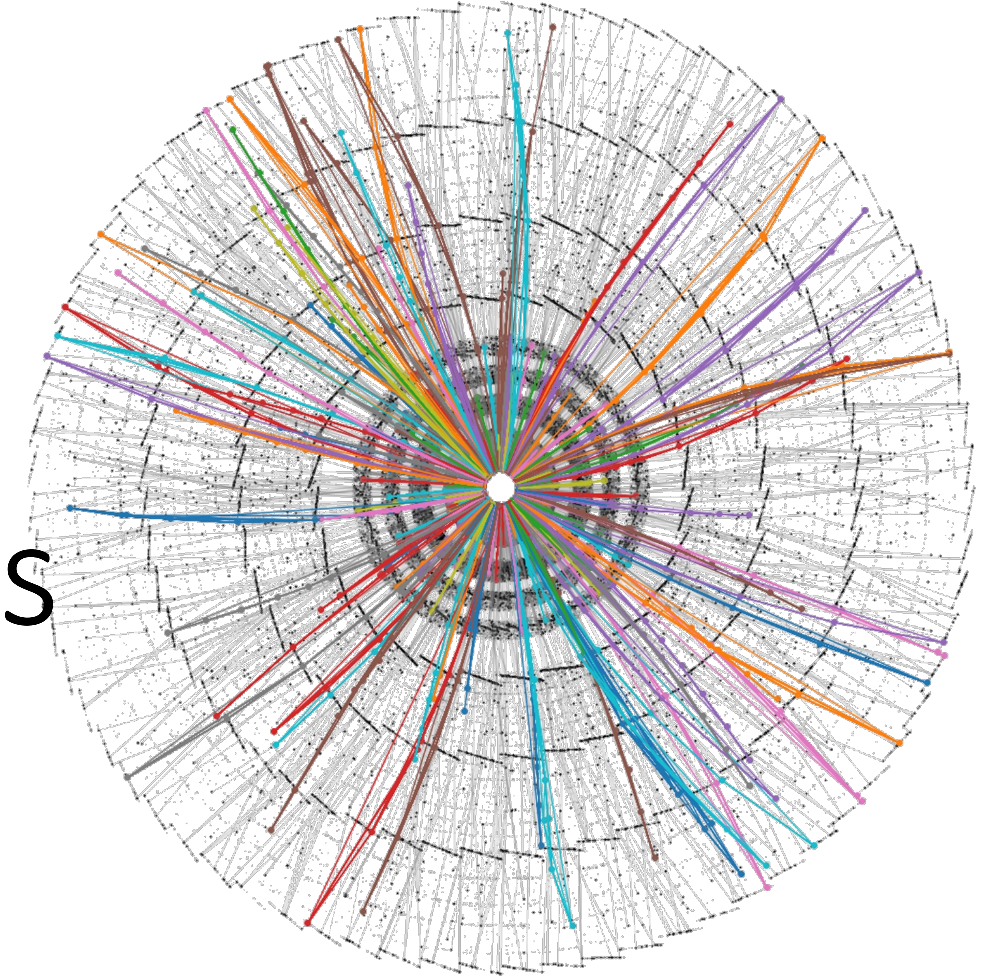
Tracking with a Graph Neural Networks

RHIC/AGS Annual Users' Meeting

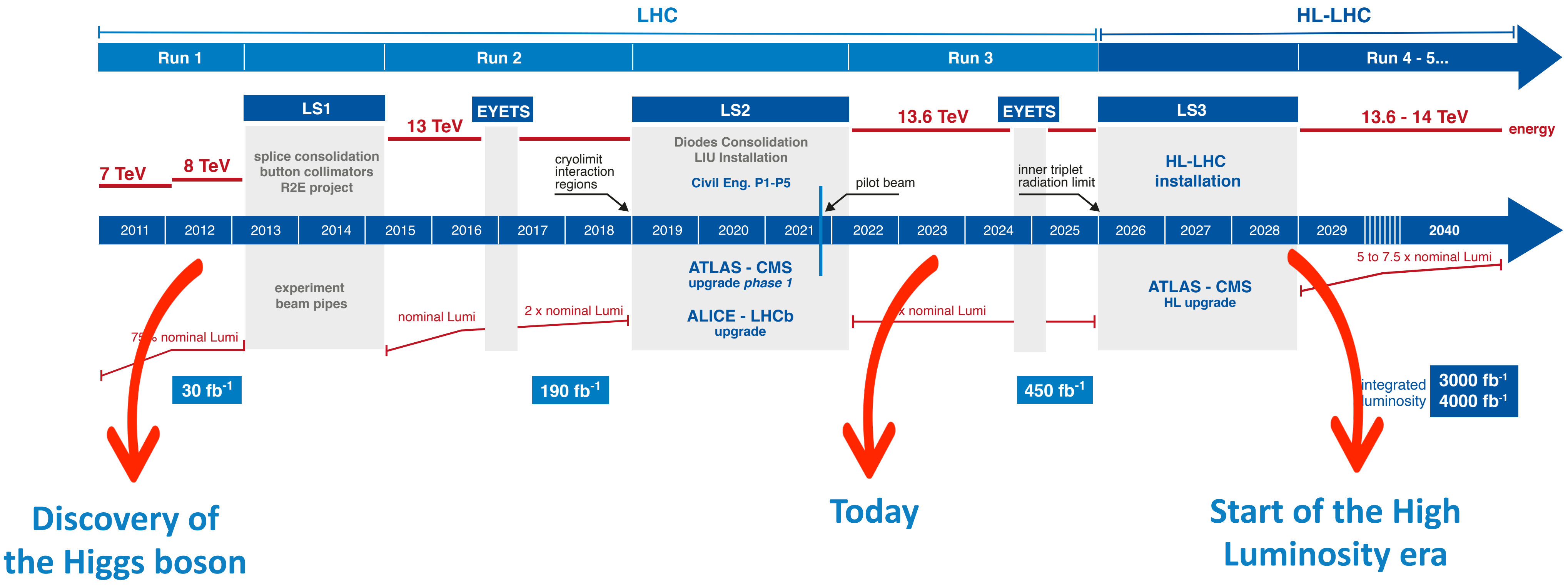
1^{st} August 2023



Charline Rougier

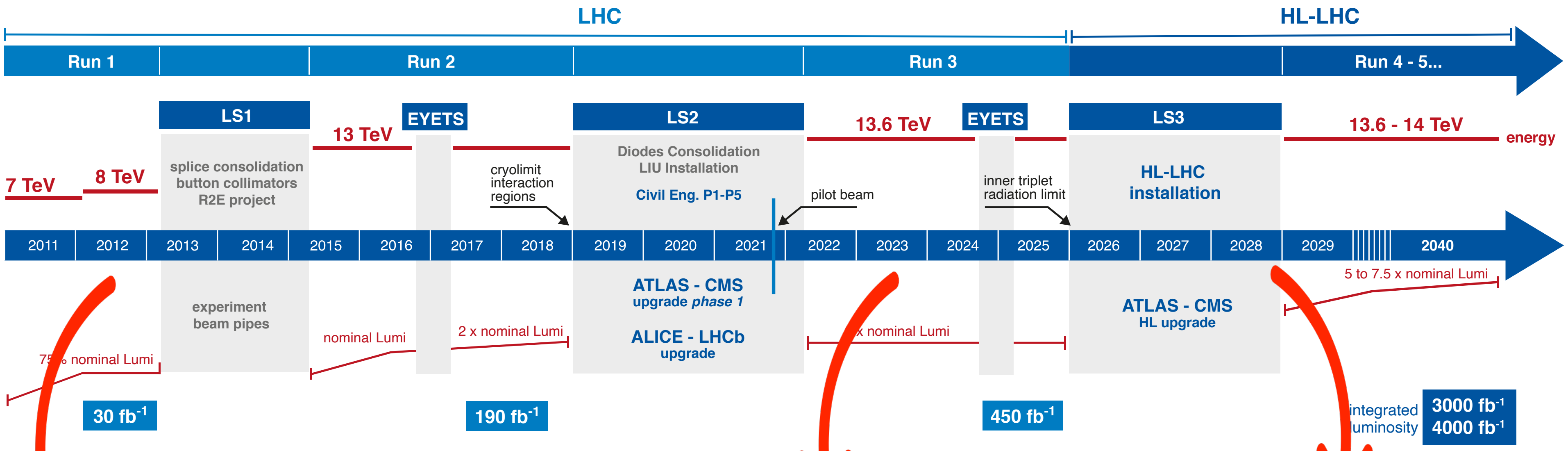
Laboratoire des 2 Infinis - Toulouse

The LHC upgrade: HL-LHC era



- Physics run to start in 2029
- Increase by one order of magnitude the integrated luminosity collected by ATLAS and CMS

The LHC upgrade: HL-LHC era



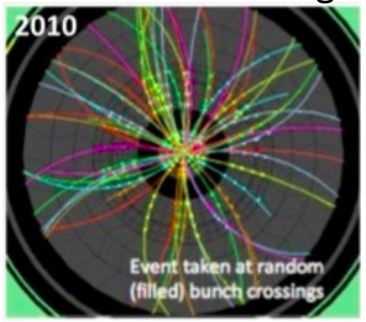
Discovery of the Higgs boson

Today

Start of the High Luminosity era

200 pile-up

Start of data taking



12 pile-up

40 pile-up

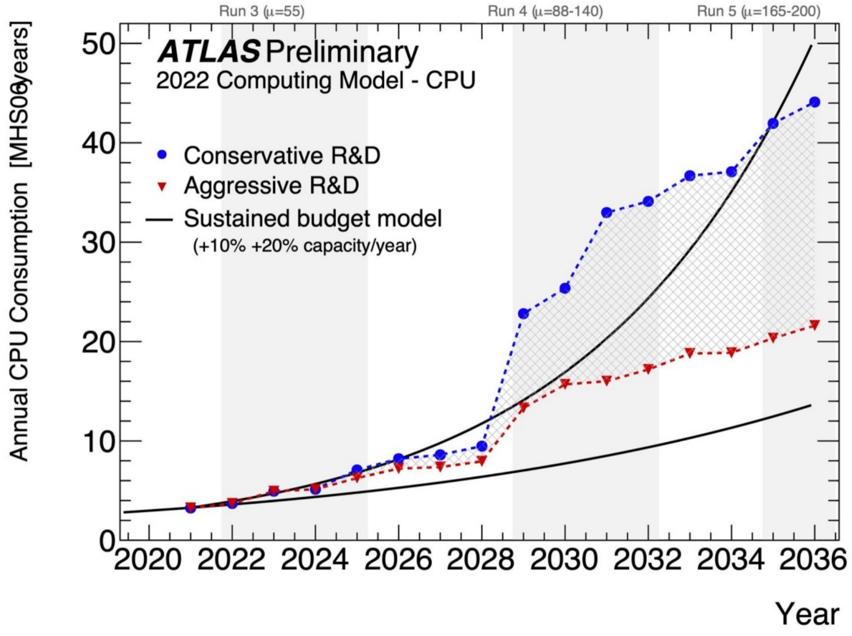
charline.rougier@l2it.in2p3.fr

LHC High Luminosity upgrades

- **The LHC upgrade: HL-LHC era**

➡ Increase in event complexity: $\langle \mu \rangle \approx 200$
➡ Increase in data taking rate
➡ ATLAS detector upgrades: new **I**nner **T**rac**k**ing detector **ITk** included

Brings unprecedented **challenges** for software and computing.



From [ATLAS HL-LHC Computing Conceptual Design Report]

The offline reconstruction of ITk data represents about **20%** of computing resource needs.

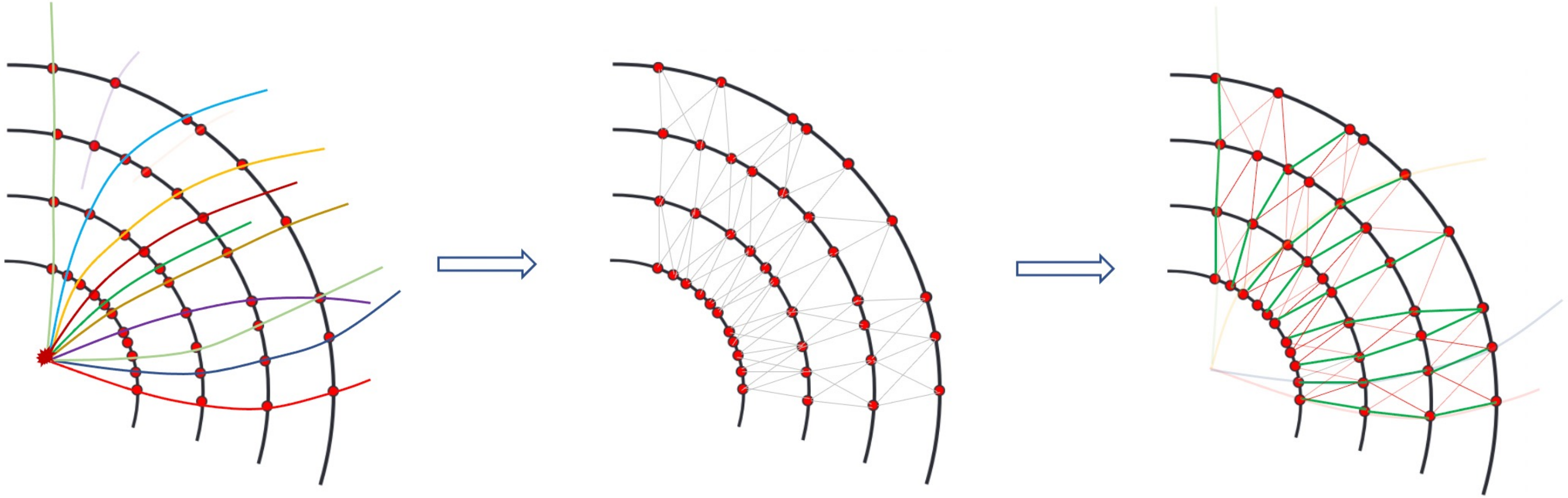
Potential improvements:

- **Hardware** updates, better CPUs, new chip architectures...
- **Software** updates, including machine learning.

Machine learning applied to tracking

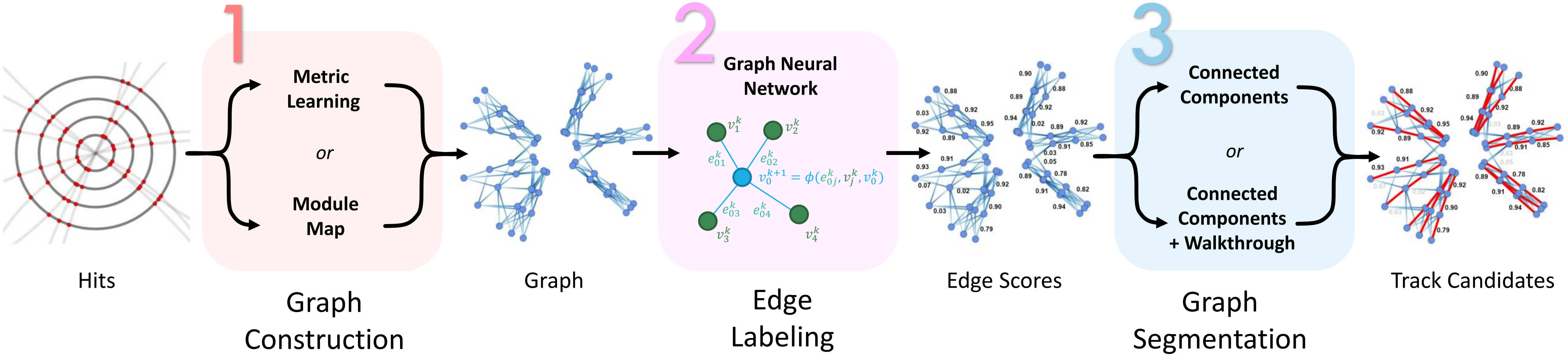
- **Track reconstruction = CPU-intensive stage**
  - ML techniques running on GPUs ? Raw data from tracking detectors are **sparse** data

- **Graph Neural Networks (GNNs):** proof of principle by Exa.TrkX project



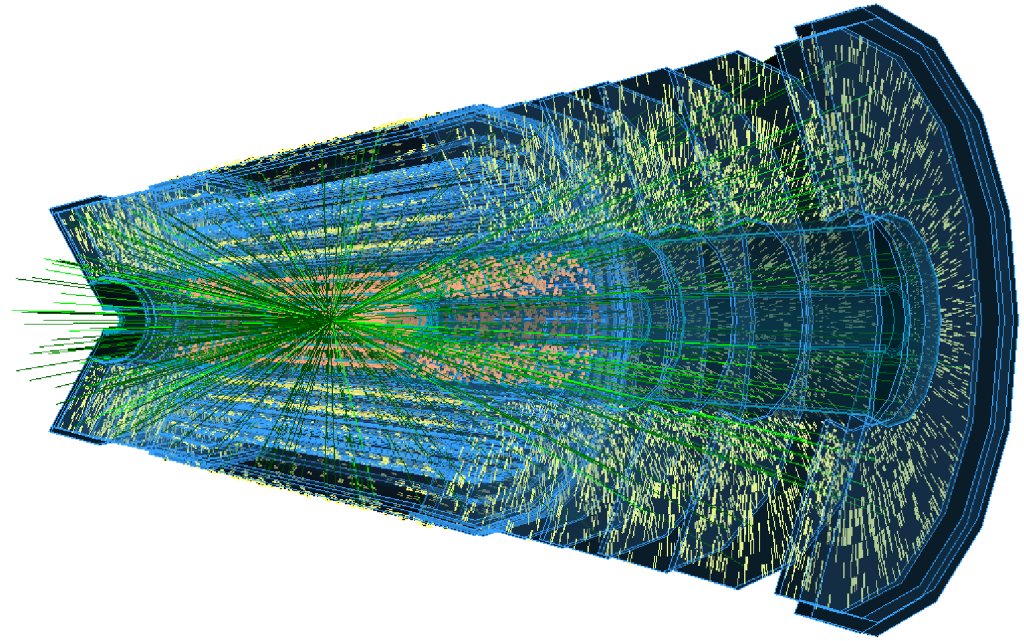
Machine learning applied to tracking

- **Development of a pipeline based on a Graph Neural Network**



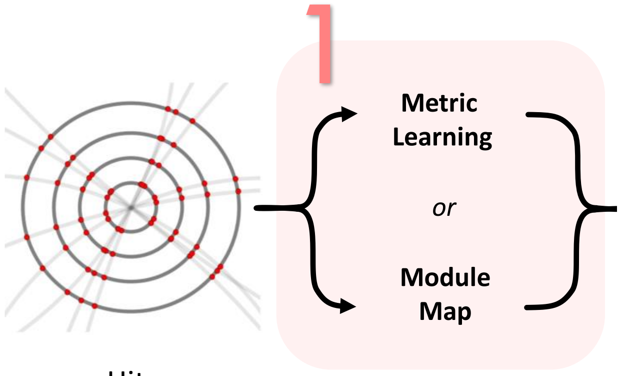
Simulated sample

- **ATLAS simulated sample: $\mathbf{t\bar{t}}$ with $\langle \mu \rangle = 200$ at $\sqrt{s} = 14\ TeV$**
  - About 100k events available
  - About 10k charged particles per event
  - About 300k space-points per event



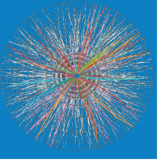
- **Define target particles**
  - $p_T > 1$ GeV
  - No secondaries
  - No electrons
  - At least 3 space-points in the detector

Dominated by soft interactions

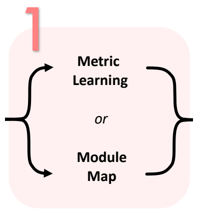


Hits

Graph
Construction



Graph representation of tracking data

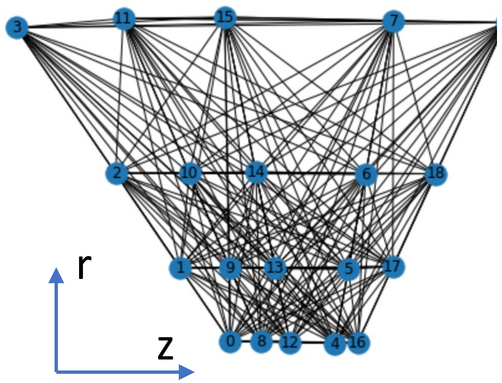


Graph Construction

Node = 1 space-point

Edge = connection between two nodes.

- Existence of edge = the 2 nodes could potentially represent 2 **successive** space-points on the same track.



Example with 19 hits in the (z,r) plane

$\mathrm{O}(300\mathrm{k})$ space-points in an event => fully connected graph $O(10^{10})$ edges

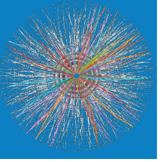
- Comprises unphysical connections

Key question of graph construction:

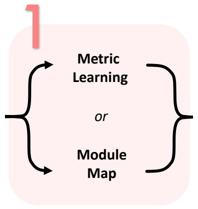
- **How de we choose the connections between nodes ?**

Graph construction is one of the most important parts of a GNN tracking pipeline:

- High efficiency is mandatory: so far lost edges means incorrect track reconstruction
- Graph topology has a huge influence on the performance of a GNN model



Graph creation: the module map

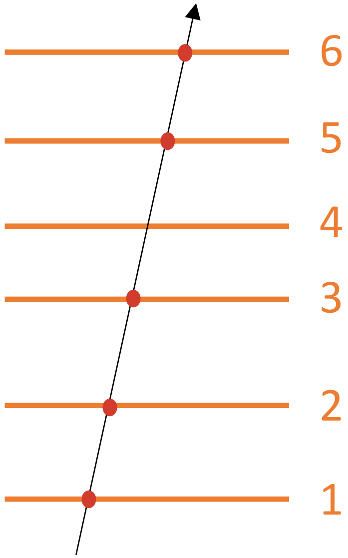


Graph Construction

Module Map

Training stage

The path of a target particle is followed inside ITk to record all possible **connections** between triplet of silicon **modules**.



Connections record :

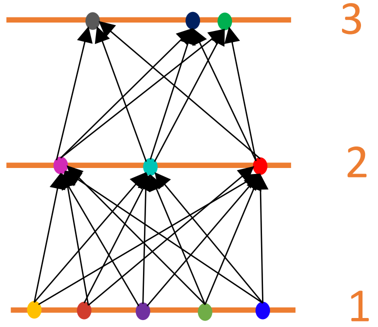
1 → 2 → 3
2 → 3 → 5
3 → 5 → 6

The Module Map is built using 90 000 events. It comprises **1 242 665** connections.

Edge creation

Edges are created following the connections of the Module Map.

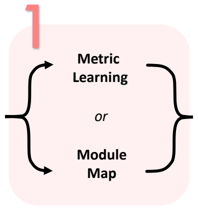
1 → 2 → 3
2 → 3 → 5
3 → 5 → 6



Direction "***inside-out***" are given to edges.

O(billions) edges → additional pruning applied to reduce to O(1.3 million) edges

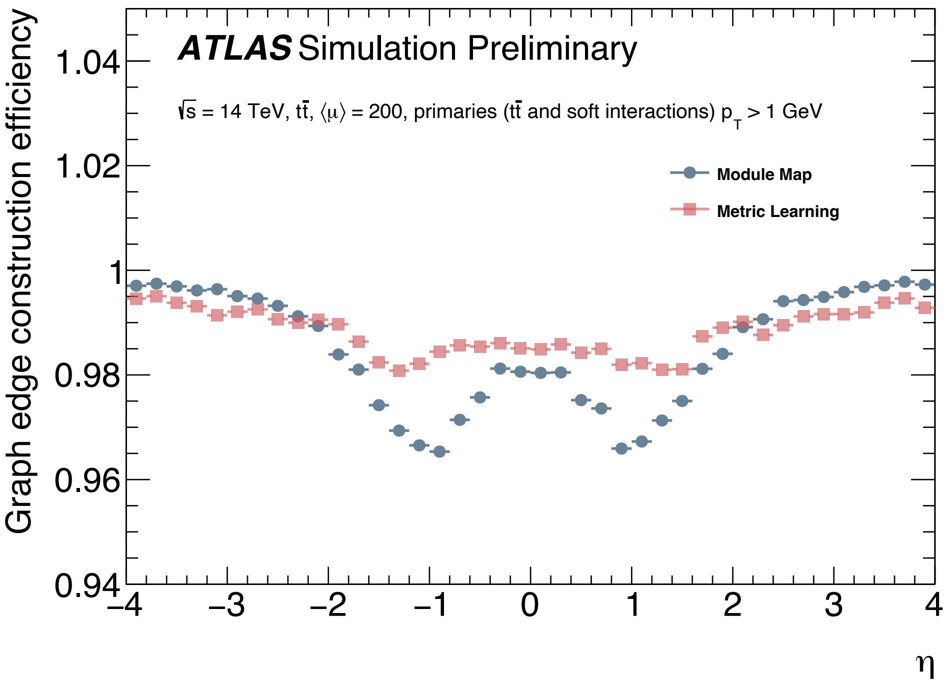
Graph edge construction efficiency

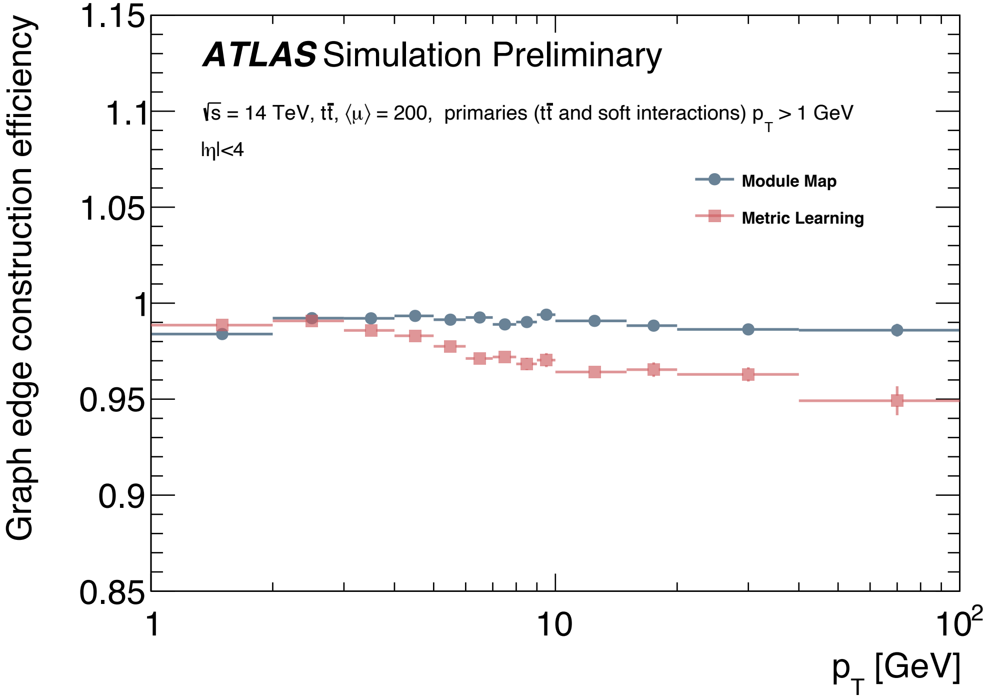


Graph Construction

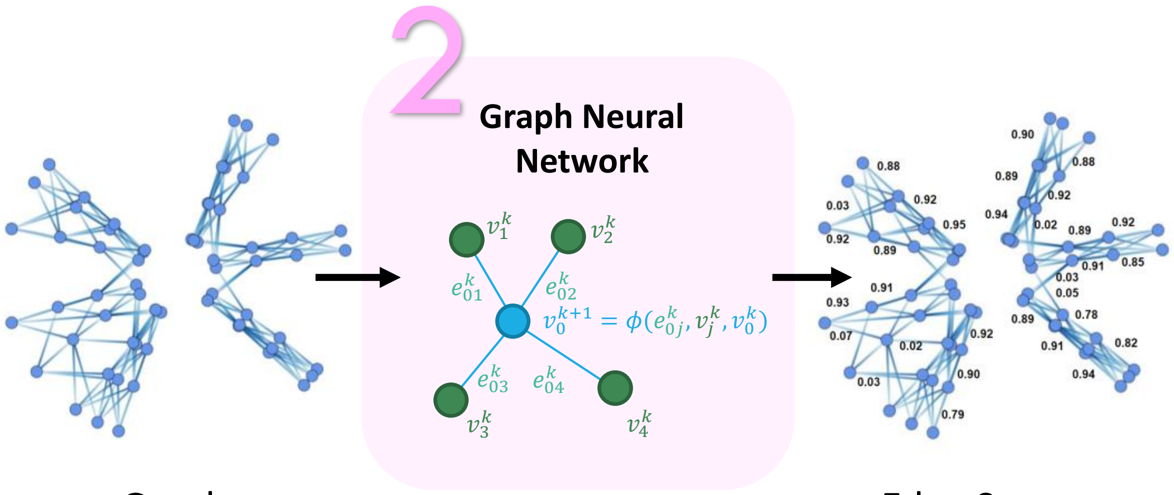
- **Graph edge construction efficiency**

High efficiency is a necessity: an edge lost during the graph construction can't be recovered later.





In the following the module map is the method used to build graphs.
The graphs built have 100% efficiency (events have been used during the module map creation).

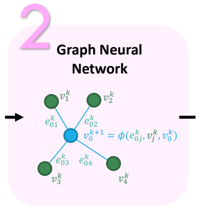


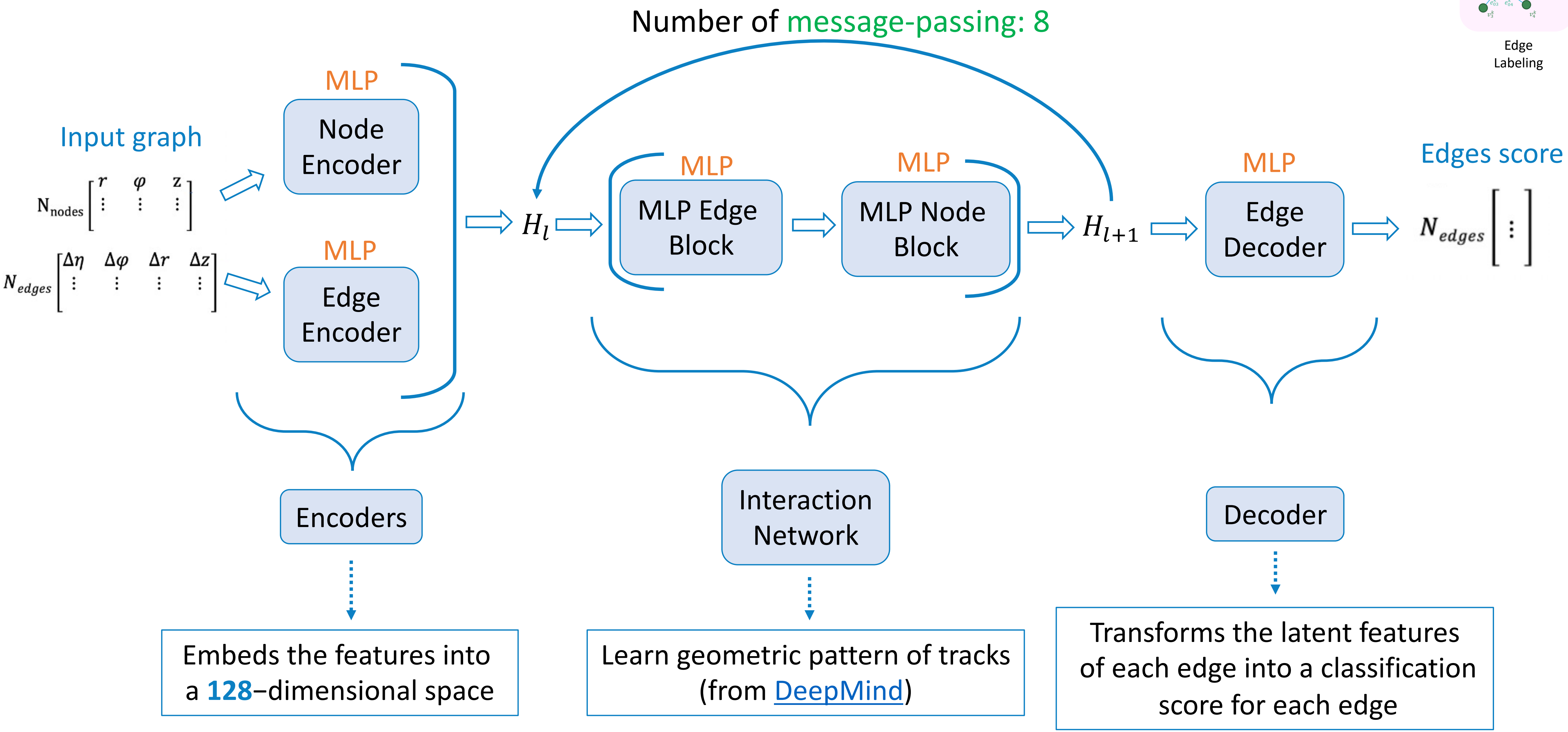
Graph

Edge
Labeling

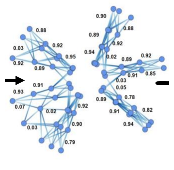
Edge Scores

Graph Neural Network model

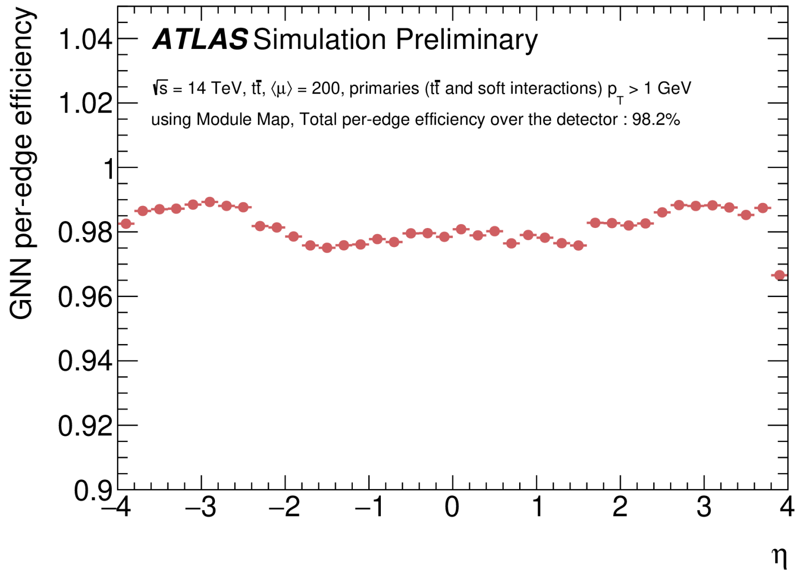




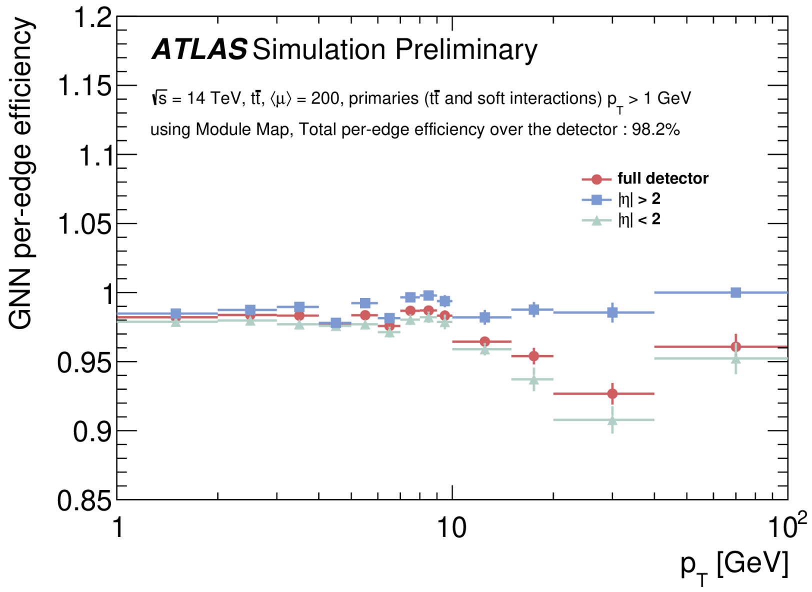
GNN edge-level performance



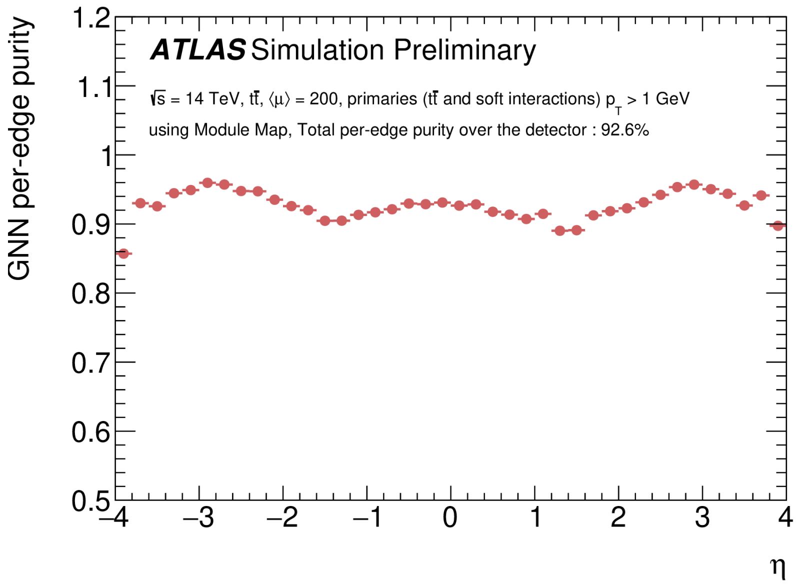
Efficiency vs. η

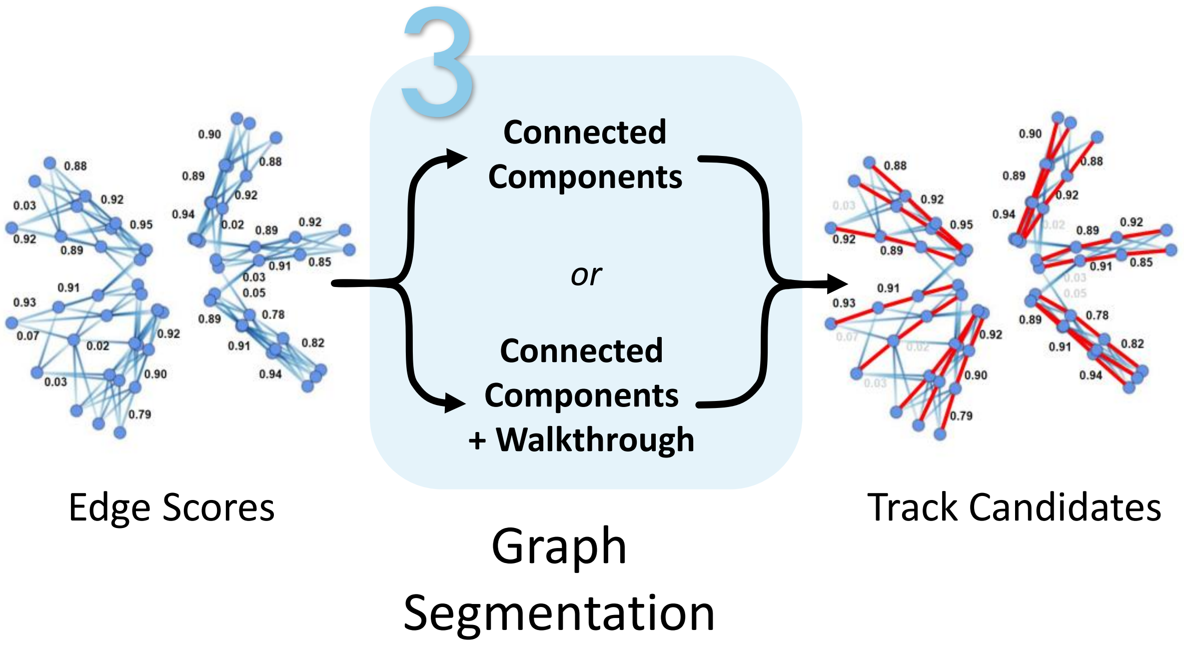


Efficiency vs. p_T

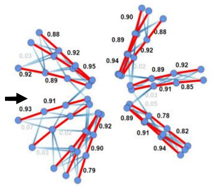


Purity vs. η




3
Connected Components
or
Connected Components + Walkthrough
Edge Scores
Track Candidates
Graph Segmentation

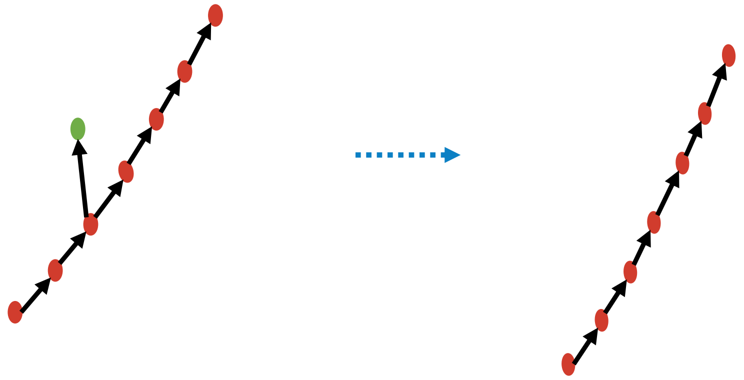
Building track candidates



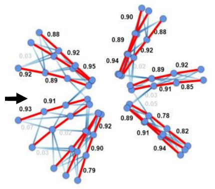
Track Candidates

- **Step 1: connected component**
  - ➡ This stage is intended to prune the graph from fake edges.
  - ➡ Apply a very low edge score cut: 0.01
  - ➡ ~ 1.3M edges -> 30k edges

- **Step 2: additional filtering, if needed**
  - ➡ If further selections are needed, applied an iterative algorithm which used higher cut on the edge classification score.



Track reconstruction matching criteria



Track Candidates

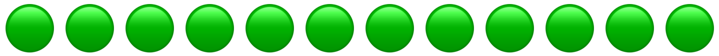
- **Evaluation of the track candidates**

No track fit is applied.

Evaluation done on $t\bar{t}$ + PU.

- **Matching criteria**

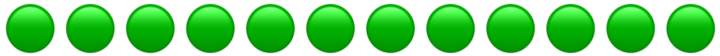
Particle



Track candidate

Standard matching

$$P_{\mathrm{match}} > 0.5$$

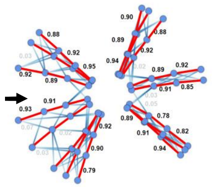


Strict Matching

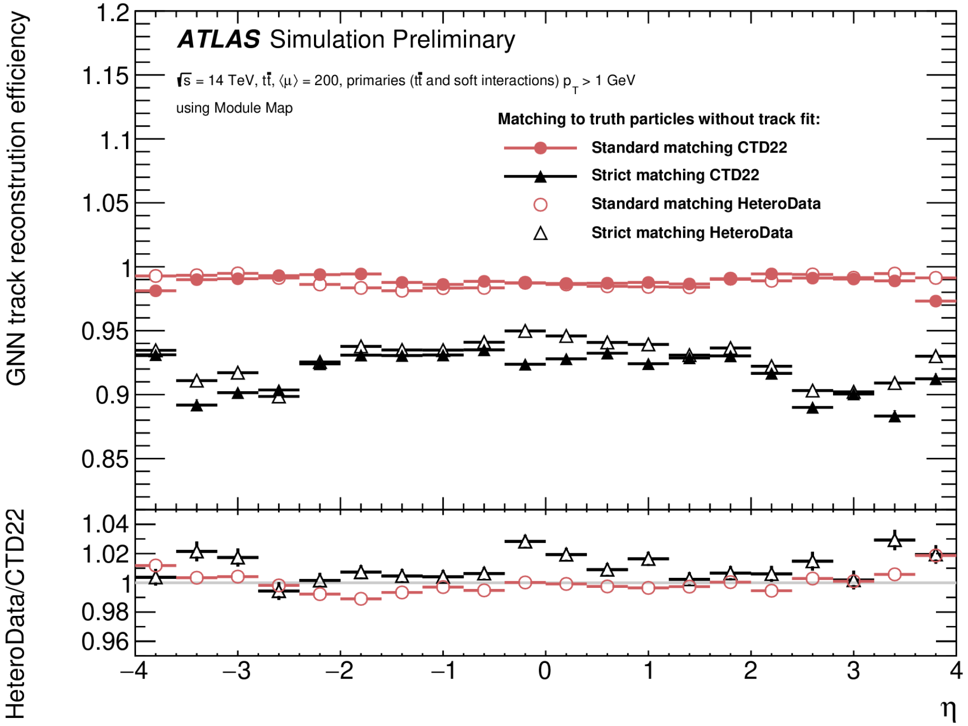
$$P_{\mathrm{match}} = 1$$

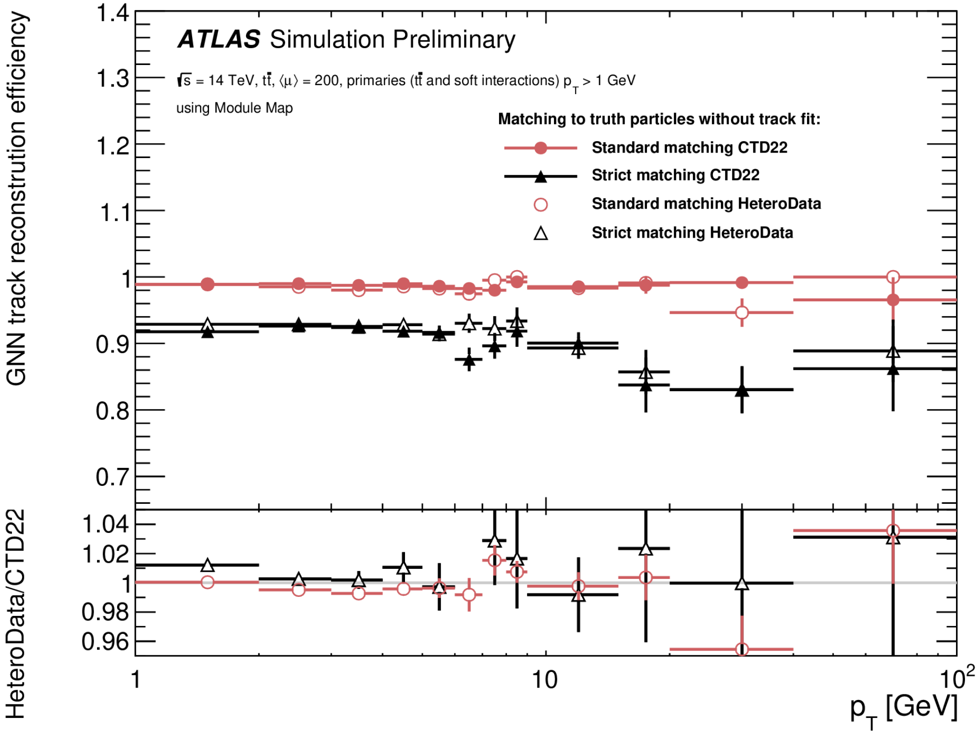
Track purity = 1

GNN track reconstruction efficiency



Track Candidates



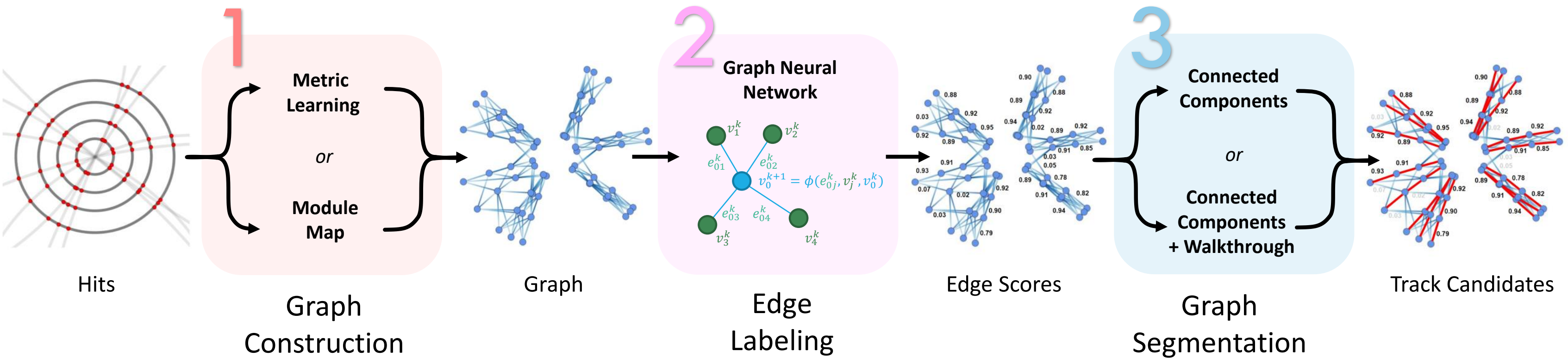


Track candidate not matched to any particle = fake track

➡ found to be $O(10^{-3})$

GNN tracking framework

Public pipeline



The pipeline is **public** here.

To run the pipeline, simulated sample could be obtain from ACTS using the OpenDataDetector, or using official sample from experiment.

See tutorial available here

➡ **You are very much welcome to join the effort ☺**

Conclusion and prospects

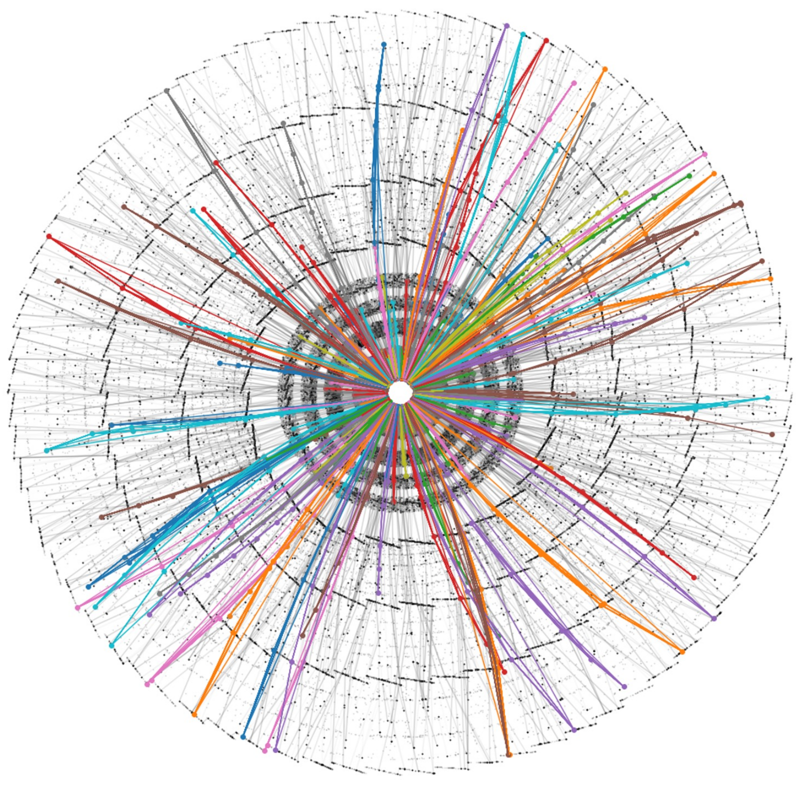
- **Conclusion**

First results using a GNN-based track reconstruction with ITk simulated data are promising and realistic.

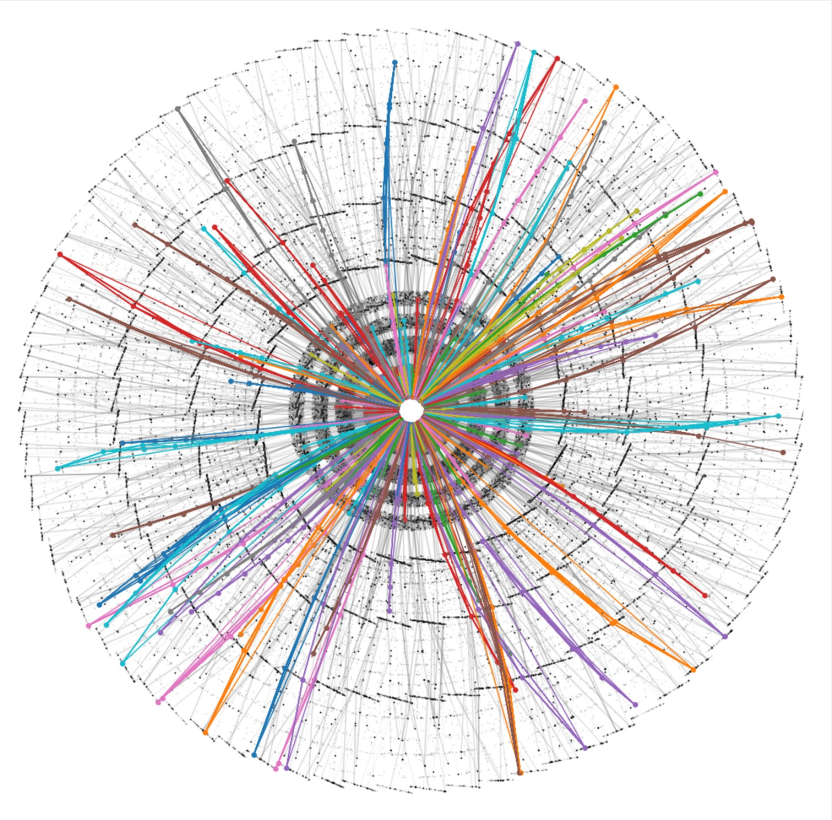
- **Prospects**

Several studies are ongoing:

- Finish integration ACTS and Athena
- Fair comparison (timing and reconstruction performances) with Athena and ACTS Combinatorial Kalman Filter
- New track building stage, able to run on GPU
- Ongoing studies to assess the feasibility of the solution for online tracking



Thanks for your attention ☺



BACK-UP

GNN for tracking and timing

Timing consideration

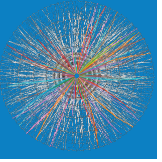
- The target is to run the full pipeline in **< 1 second.**
- Need to be fully run on GPU.

TrackML timing (Similar graph size as for ITk)

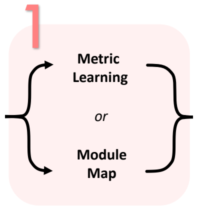
Pipeline step	V100 GPU
Graph construction (metric learning)	~ 500 ms
Graph construction (module map)	In progress target ~ 100 ms
GNN	~170 ms
Connecting component	~100 ms

(See this paper)

How to improve: GPU kernels have dedicated operation for NN. But the GNN model is much complex with its 8 message-passing operations and the way the memory is therefore handle.
Using dedicated GNN kernels could only improve the timing, the memory consumption and the energy cost.



Graph creation: the metric learning



Graph Construction

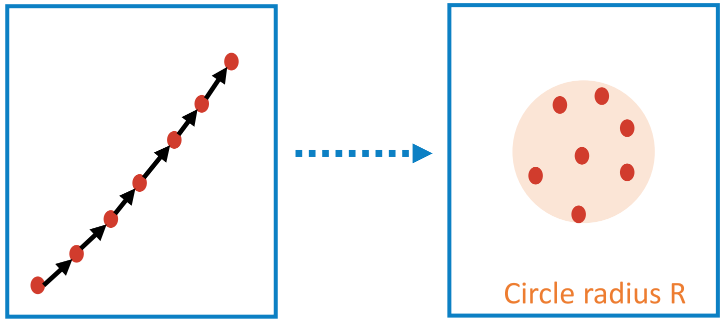
Metric Learning

Training stage

All space-points belonging to the same target particles are **learned** by a Multi-Layer Perceptron (MLP) to be embedded into a **space** where they are close.

Euclidian space

N-dimensional space learned by the MLP



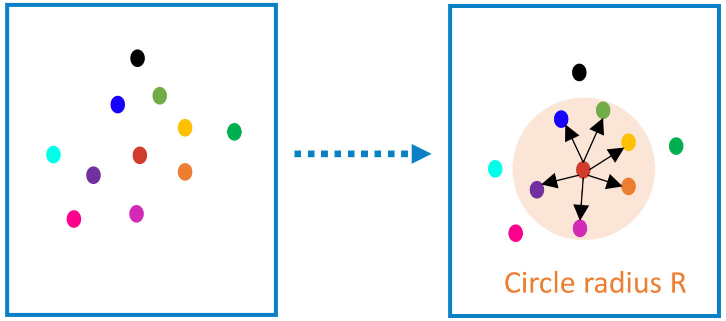
The network is trained with a few thousand of events

Edge creation

Given a source node, edges between this node and all nodes within a radius R from the source are created.

N-dimensional space learned by the MLP

Edges created



No particular meaning of direction.

O(3 million) edges → additional pruning applied to reduce to O(1.3 million) edges