Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC

IN2P3/IRFU ML Workshop March 2021

Sylvain Caillou (sylvain.caillou@l2it.in2p3.fr)

On behalf of

L2IT ATLAS and CAD Teams

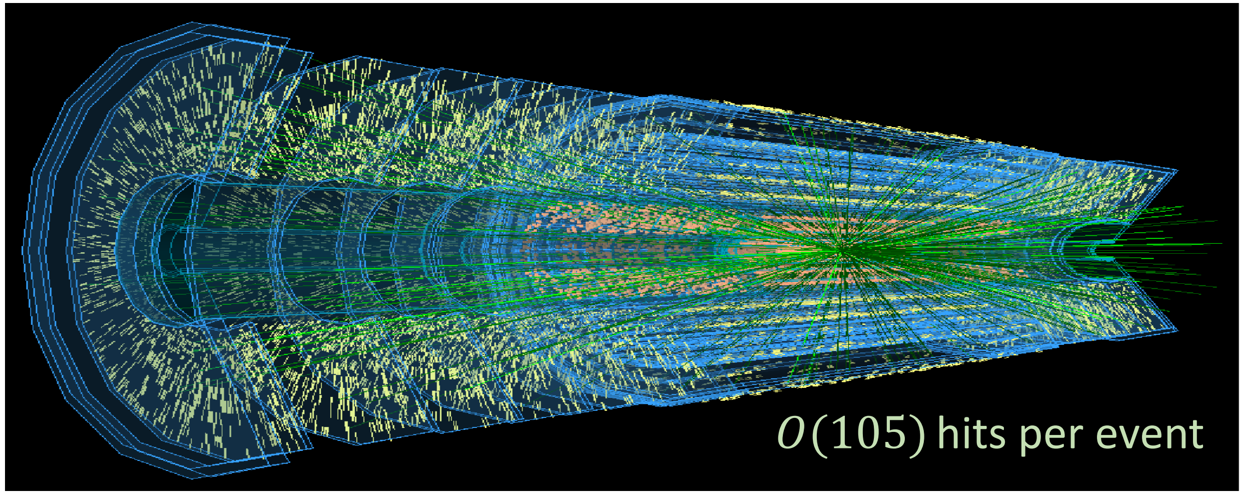
Catherine Biscarat, *Sylvain* Caillou, *Charline* Rougier, *Jan* Stark and *Jad* Zahreddine
Laboratoire des 2 Infinis - Toulouse (L2IT-IN2P3), Université de Toulouse, CNRS, UPS

A few words about the speaker

- CNRS Computer Science Engineer
- 2008 - 2020 Aug : At LIMSI (CNRS INS2I)
  - Artificial Vision, robotics : Image processing, CNN
  - Natural Language Processing: Language model, RNN & LSTM, Attention model
  - Control in Mechanics Energetics : DRL
- Since 2020 Sept : at L2IT (CNRS IN2P3)
  - « Computing, Algorithms and Data» team in close collaboration with the Particle Physics team (ATLAS)
  - Expert en calcul scientifique - Machine Learning and Deep Learning

The HL-LHC computing challenge

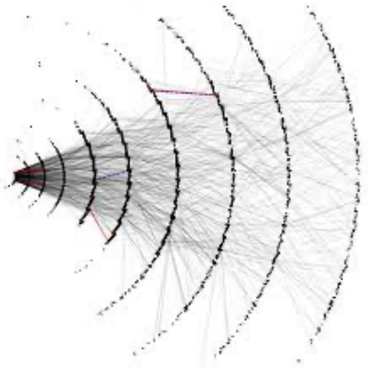
- High-Luminosity LHC (HL-LHC): 3000 fb^{-1} of data (x10 vs LHC); start in 2027
- Larger dataset & more complex events => steep increase in computing resources needed



- Assuming a flat budget => computing hardware will not be able to provide this increase
- Physics reach during HL-LHC will be limited by affordable software and computing and by how efficiently these resources can be used => need to improve the algorithms

Graph Neural Networks (GNNs): A promising ML solution

- ExaTkrx Project: breakthrough in GNN-based algorithm for track pattern recognition
  - S. Farrell et al. (2018), arXiv:1810.06111 [hep-ex] (**NeurIPS 2019**)
  - Great promising performance on the TrackML dataset
  - Proof of principle: Not a real-world experiment
    - Only central part of the detector, split into 16 sections
    - Graph construction algorithm only works for trivial detector geometries



- Since then, significant effort in the development of algorithms based on GNNs
  - X. Ju, S. Farrell, P. Calafiura, D. Murnane, Prabhat, L. Gray, T. Klijnsma, K. Pedro, G. Cerati, J. Kowalkowski et al. (2020), arXiv:2003.11603 [physics.ins-det]
  - N. Choma, D. Murnane, X. Ju, P. Calafiura, S. Conlon, S. Farrell, Prabhat, G. Cerati, L. Gray, T. Klijnsma et al. (2020), arXiv:2007.00149 [physics.ins-det]
  - X. Ju, D. Murnane, P. Calafiura, N. Choma, S. Conlon, S. Farrell, Y. Xu, M. Spiropulu et al. (March 2021), arXiv:2103.06995 [hep-ex]

GNN for track pattern recognition

- Why GNN?
  - Graph can represent *sparse* data
  - Message passing convolution generalises CNN from flat to arbitrary geometry
  - Can learn node (i.e. hit / spacepoint) features and embeddings, as well as edge (i.e. relational) features and embeddings

=> Very suitable for track pattern recognition

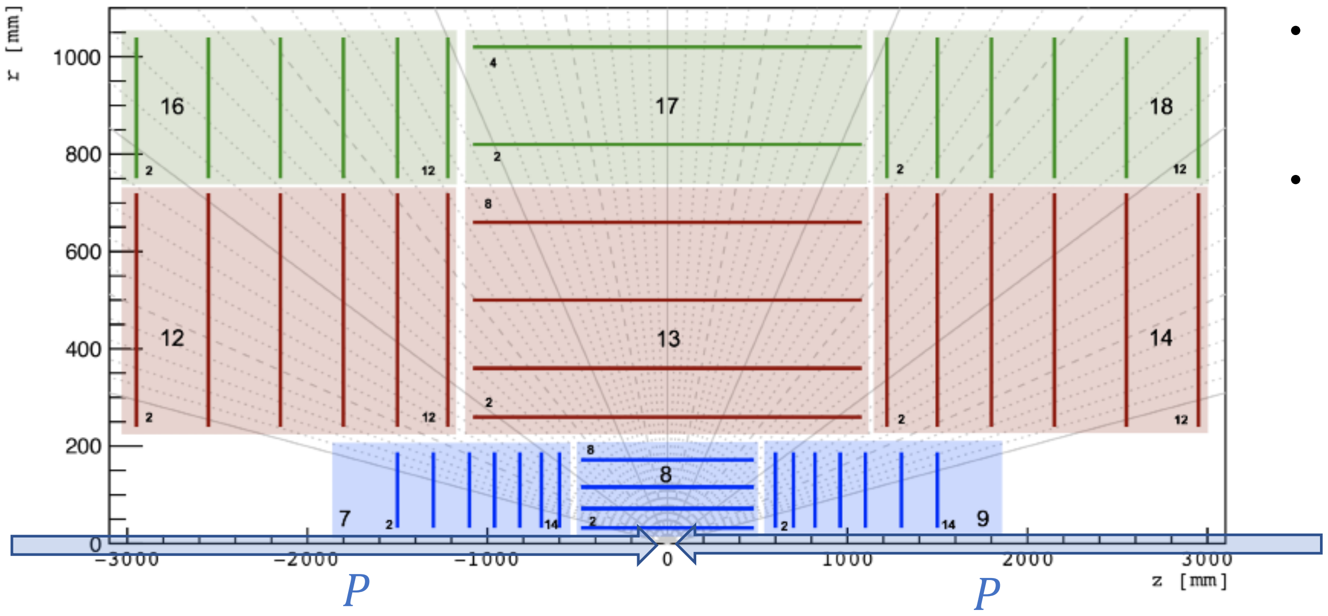
Towards a realistic track reconstruction algorithm

- We are members of the ATLAS collaboration and we are comitted to ITk sotfware
- New effort at L2IT to implement a realistic GNN-based algorithm that can be deployed in an HL-LHC experiment:
  - C. Biscarat, S. Caillou, C. Rougier, J. Stark, and J. Zahreddine (March 2021), arXiv:2103.00916 [physics.ins-det]
  - Novel algorithm for graph construction that can handle any complex geometry of a realistic detector
  - Methods for memory management that allow GNN training on the full detector
  - Integration into the ACTS framework (A Common Tracking Software)
    - Facilitate their use with detailed simulation studies of HL-LHC detector
    - Direct comparison with other tracking algorithms (e.g. Kalman Filter)

Simulated data

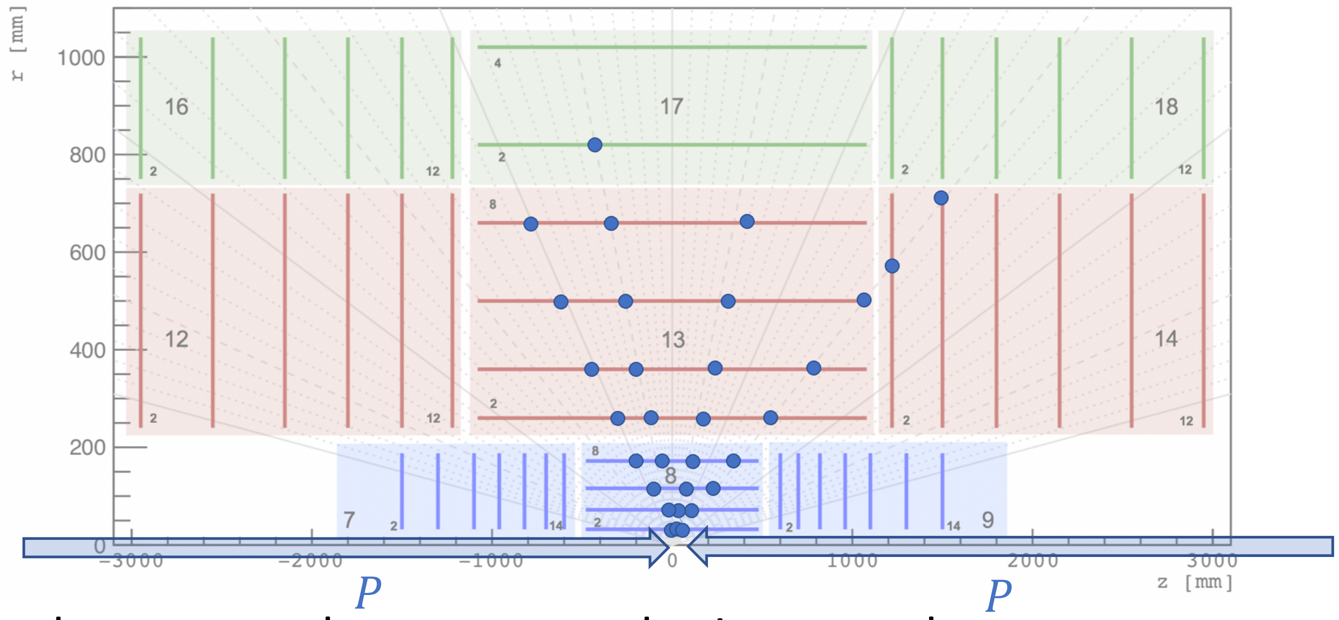
1. Generation of 1000 $t\bar{t}$ events in proton-proton collisions
   - $\sqrt{s} = 14\, TeV, \mu = 200$ (HL-LHC conditions), events generated using PYTHIA8
2. Fast track simulation inside the ACTS framework
   - Response of the *Generic Detector* defined in ACTS

   Inspired by initial layouts for the ITk detector (future ATLAS tracker)



- *Layers* (18728 silicon *modules)* parallel or perpendicular to the beam line
- When a charged particle traverses a *module*, it leaves a space-point *hit*

Graph representation of data

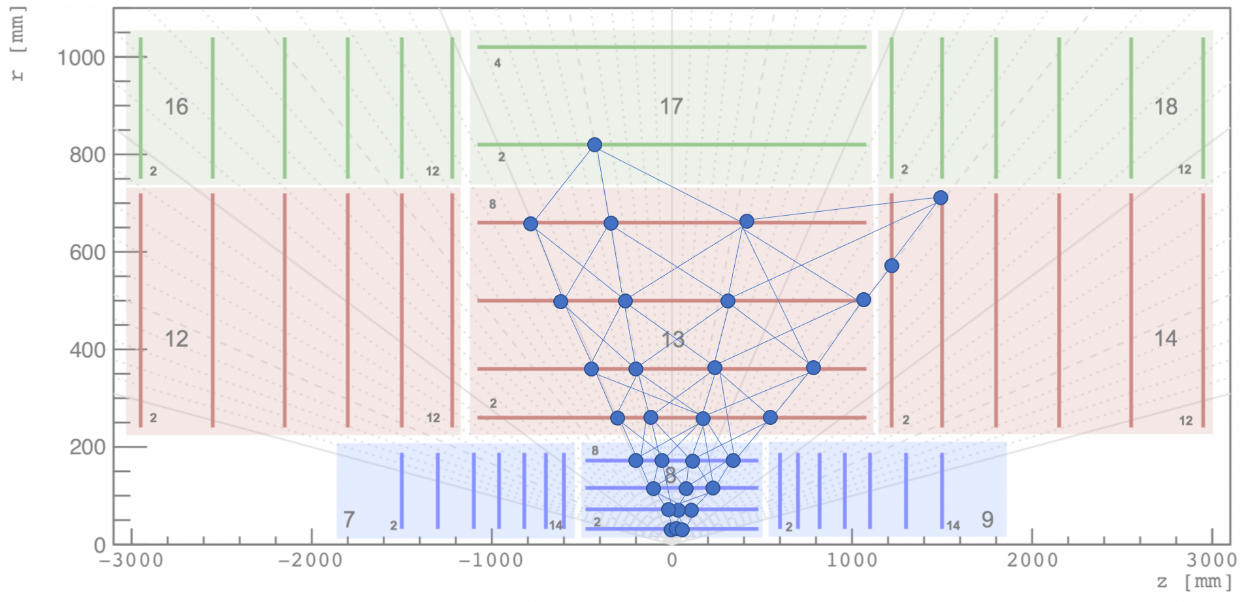


Data from a tracking detector can be represented using a graph

- A node represents a hit
- An edge between two nodes = nodes could *potentially* represent two successive hits on a track
- An edge can be *true* (segment of a track) or *fake* (not a segment of a track)
- Our goal: identify the ***true*** edges among all the edges
- Fully connected graph => $O(10^{10})$ edges

$\Rightarrow$ Key feature of graph construction: initial *choice* of the edges (increasing the purity of true edges)

Graph representation of data



Data from a tracking detector can be represented using a graph

- A node represents a hit
- An edge between two nodes = nodes could *potentially* represent two successive hits on a track
- An edge can be *true* (segment of a track) or *fake* (not a segment of a track)
- Our goal: identify the ***true*** edges among all the edges
- Fully connected graph => $O(10^{10})$ edges

$\Rightarrow$ Key feature of graph construction: initial *choice* of the edges (increasing the purity of *true* edges)

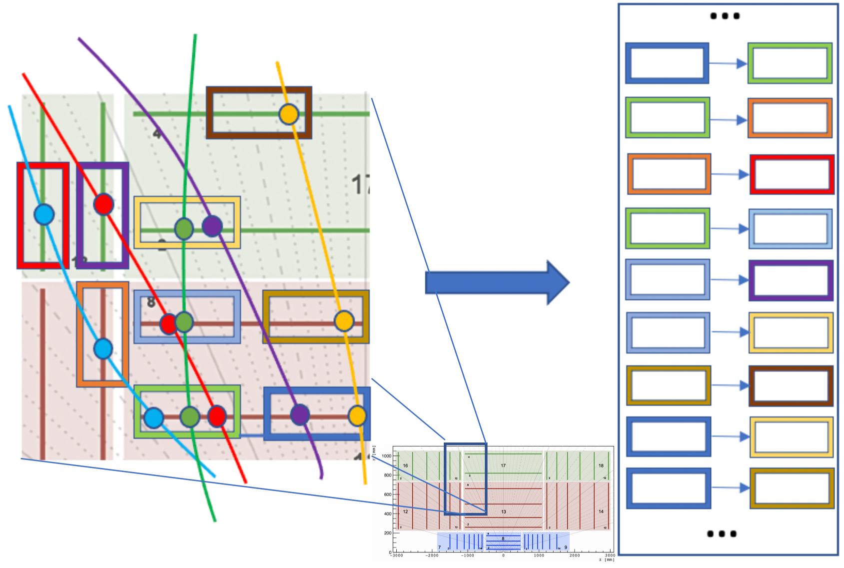
Graph representation of data

⇒ New data-driven graph construction algorithm

⇒ Basic idea: build graphs starting from a list of possible connections

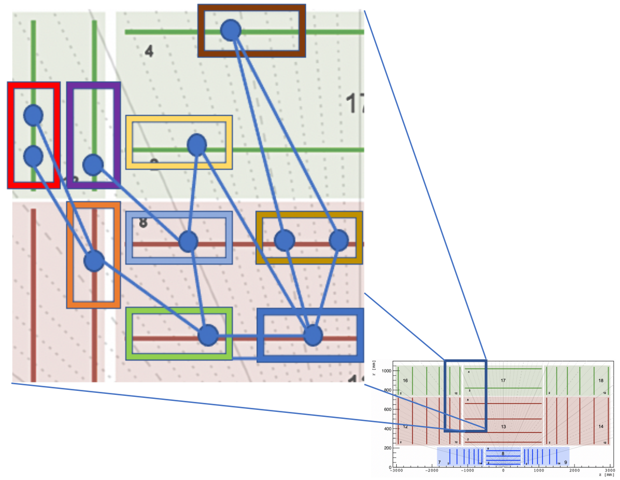
Done once:

- Observation of 1k simulated events
- Module map: list of possible connections from one silicon module to another

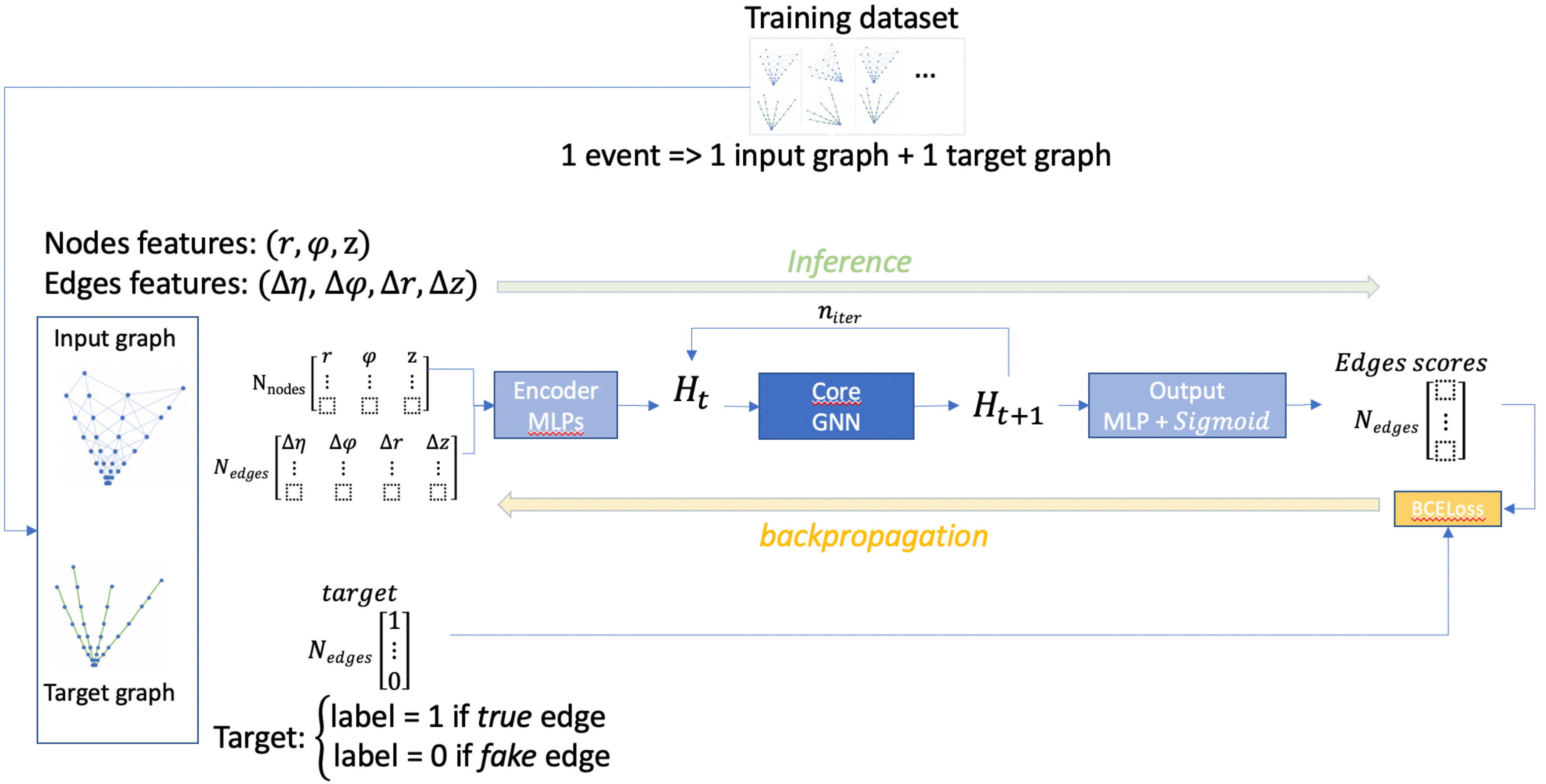


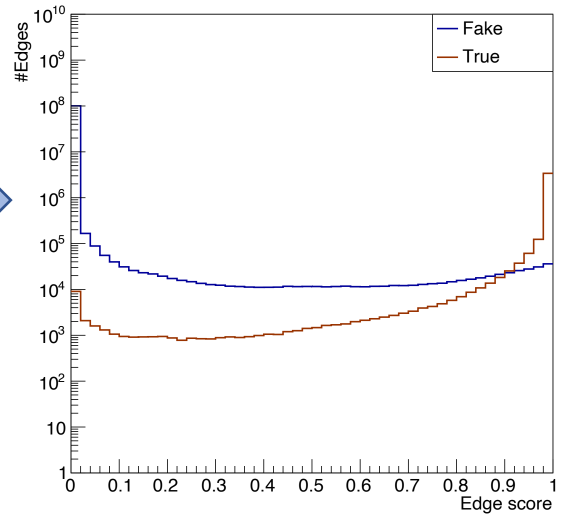
Graph construction:

- Connecting the hits if the modules connection is in the module map ⇒ $O(10^7)$ edges
- Cuts are applied on geometric parameters => $O(10^6)$ edges



GNN training and inference





Edges classification score
(on a *test* dataset)

Memory considerations

- Consider an L-layer GCN with hidden state size of H running on an N-edges graph

  => Automatic differentiation + Storing the intermediate hidden states requires $O(LNH) \Rightarrow O(10^2\,GB)$ >> Any GPU memory on the market

- Use of Large Model Support in Tensor Flow (IBM TFLMS)

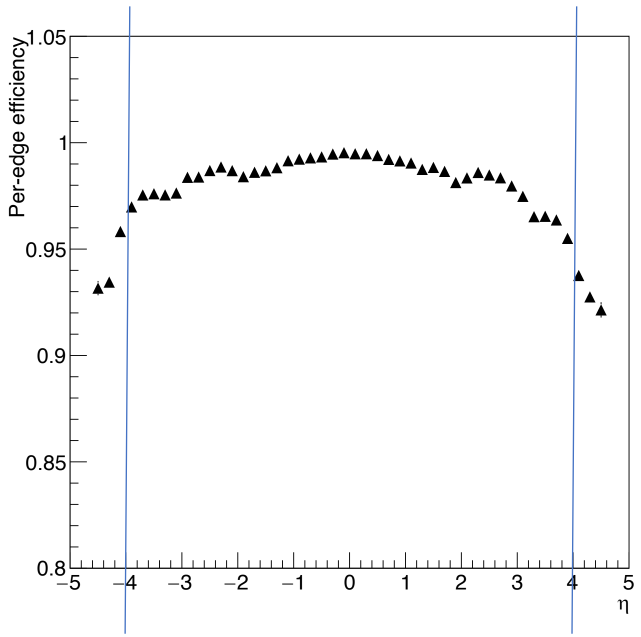
  => Tensors can be temporarily swapped to the host when they are not immediately needed

- Hardware platform (thanks to CC-IN2P3):
  - GPUs: 2 x Nvidia Quadro RTX 8000 GPUs with **48 GB** of memory each
  - Host: two AMD EPYC 7262 processors with 8 cores each and with **1024 GB** of memory

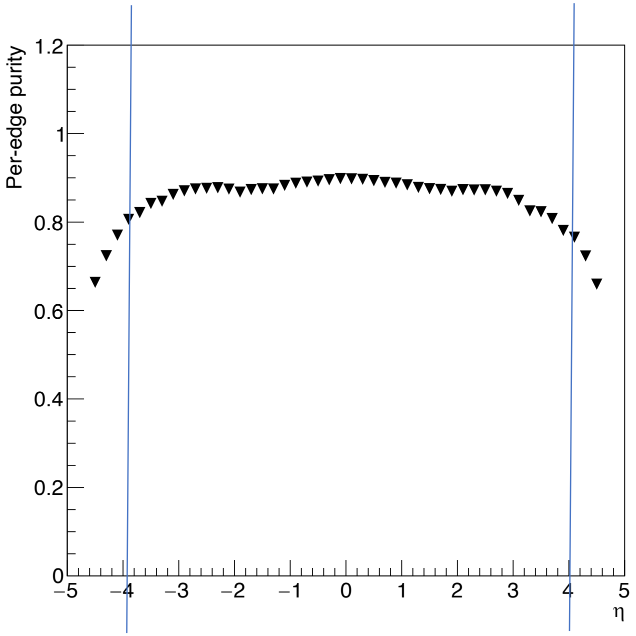
GNN prediction on the full detector

Uniform performance for $|\eta| < 4$

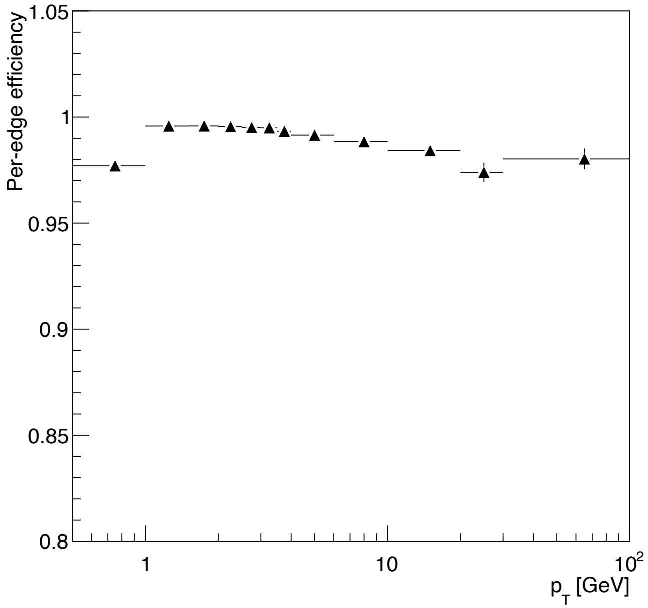
Uniform performance in p_T



Per-edge efficiency vs. η



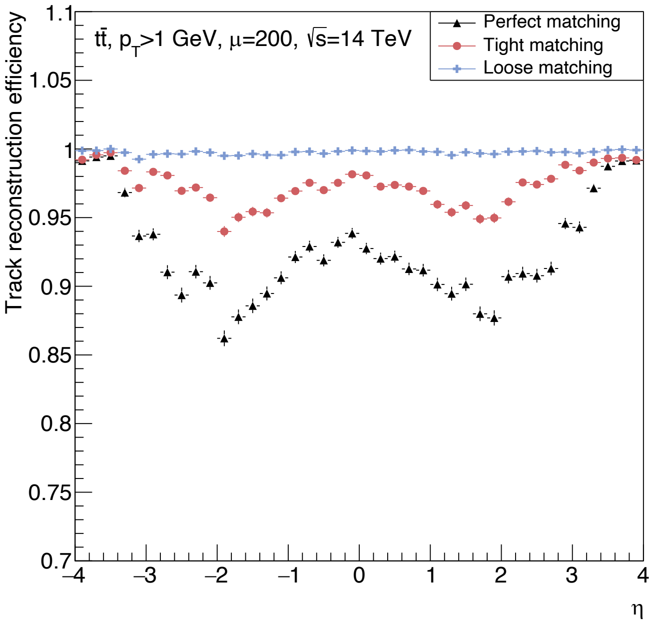
Per-edge purity vs. η



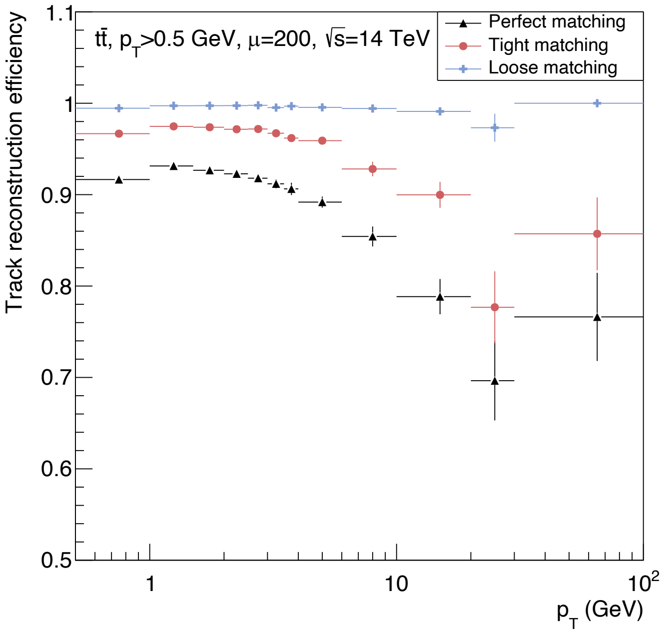
Per-edge efficiency vs. p_T

Track reconstruction results

- Next step: build tracks starting from the graph and the edge scores
- Baseline studies with a very naive graph walk-through algorithm



Track reconstruction efficiency vs. η



Track reconstruction efficiency vs. p_T

Perfect matching: all hits from a given particle are on a reconstructed track, and only these hits

Loose matching: at least 50% of the hits on a track come from the same particle

Conclusion and outlook

- Our work until now:
  - GNN-based algorithm for track pattern recognition at the HL-LHC
    - Novel algorithm for graph construction
    - Use of advanced methods for memory management (for GNN training)
  - Scale to the size and to the complexity of any realistic detector
  - Promising results for track reconstruction on the full detector
- Starting now:
  - Full collaboration with ACTS and ATLAS for integration with the latest version of ITk
  - Complete and optimised solution
    - GNN: optimised architectures and hyperparameters
    - High performance track reconstruction algorithm based on graph theory algorithm
  - Performance comparaison studies with other tracking algorithms