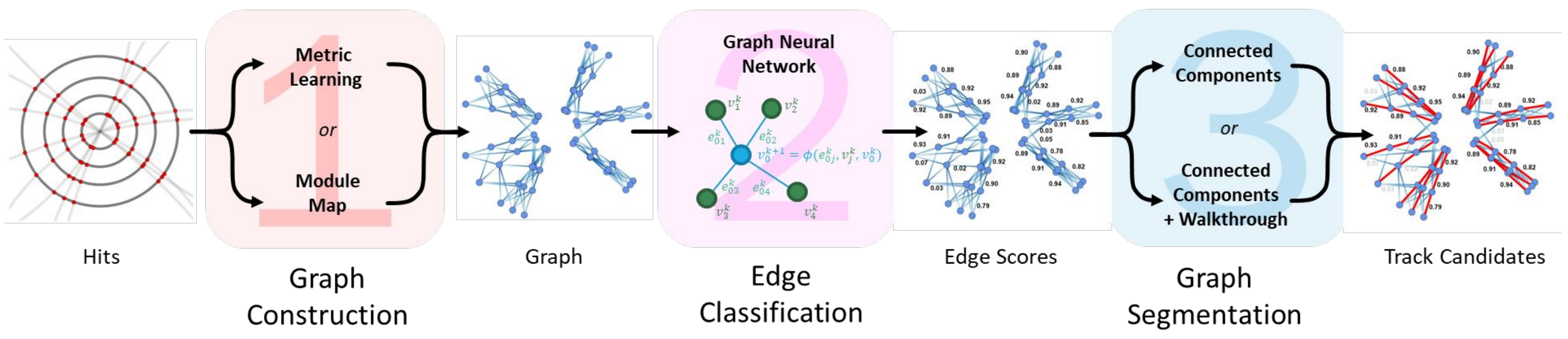
Performance of the ATLAS GNN4ITk Particle Track Reconstruction GPU pipeline

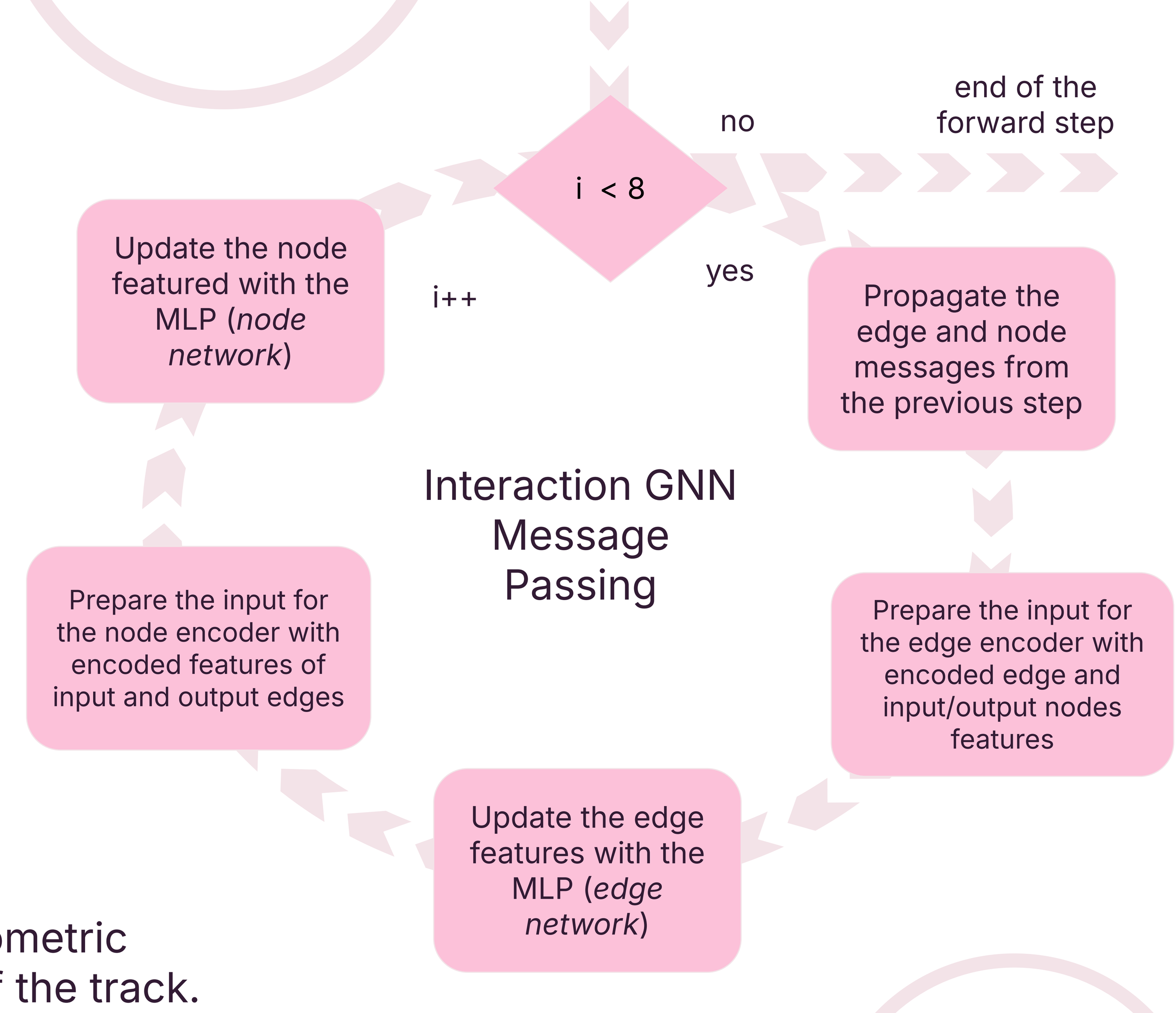
Aleksandra Poreba (CERN/Heidelberg University) on behalf of the ATLAS Collaboration
CHEP 2024, aleksandra.poreba@cern.ch

Particle track reconstruction is one of the most crucial and the time consuming steps of the event reconstruction in particle detectors.

For High Luminosity LHC, the pileup will increase and the track reconstruction in the ATLAS experiment will be more computationally expensive. At the same time, it needs to fit in the online trigger restrictions of 1 s for processing an event.

One of the considered approaches to accelerate particle track reconstruction in ATLAS is the usage of machine learning, i. e. the GNN4ITk project [1] proposing **Interaction Graph Neural Network**.





After the graph construction and initial filtering, the IGNN is used to score the edges based on their geometric properties (features) populated via **message passing**. Labelled edges are used to classify nodes as part of the track.

To even further improve the performance of the framework for a future use at the software trigger (Event Filter), different accelerators are proposed: **GPU** (focus of this poster) and FPGA.

The work presented in this poster focuses on the optimizations done to the GPU accelerated IGNN considering the **memory consumption** and the **inference time.**

FPGA Talk S. Dittmeier [Thu, 13:30, Track 2]

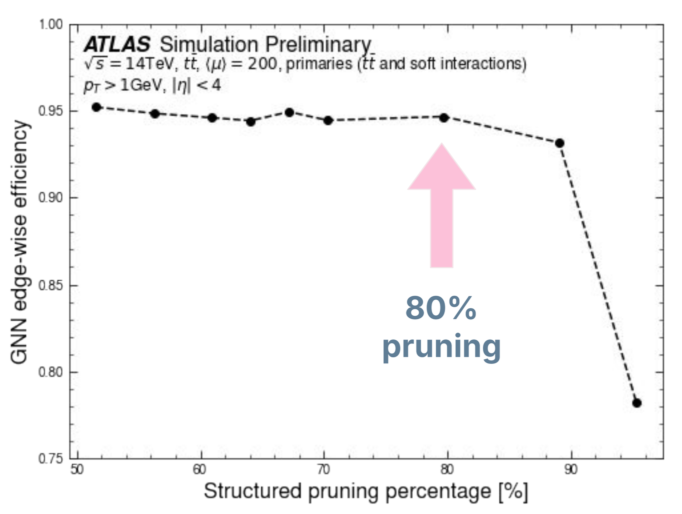
INFERENCE TIME

One of the ways to reduce the IGNN inference time would be to reduce the network size by applying **structured pruning** [3]. It effectively removes blocks (here rows) of the MLP weight matrix, therefore reduces the time of the operations.

The *default pytorch structural pruning* doesn't remove the rows, just sets the values to 0. The experimental *Saliency Pruner* [4] in pytorch does densify the weight matrix, but does not support complicated networks like this IGNN. Therefore, a multistep approach has been applied:

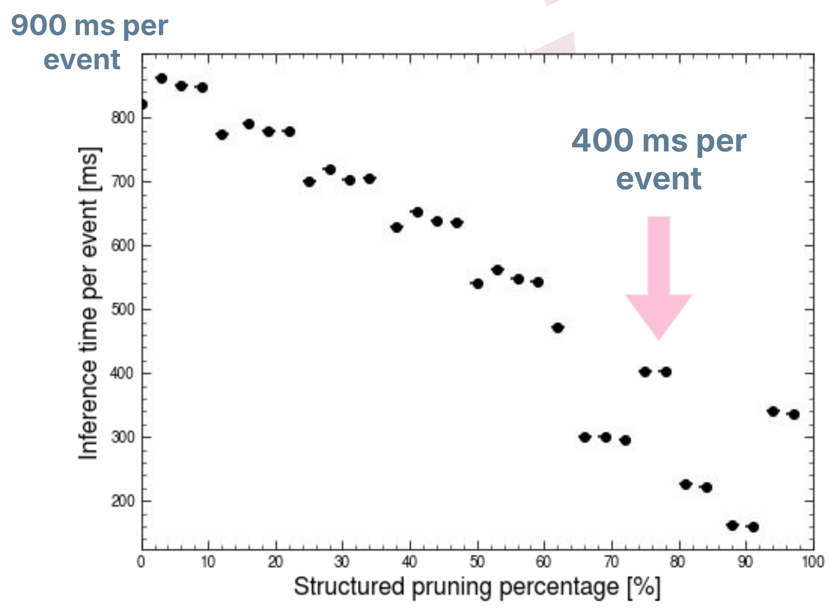
Train and prune using pytorch structural pruning with l2 norm

Transform each MLP separately into dense structure using Saliency Pruner



Measured GNN edge-wise efficiency on a IGNN trained and pruned on a simulated ATLAS events dataset restricted to ⅛ of the detector [5]

$$\text{GNN edge-wise efficiency} = \frac{\text{true-positive edges}}{\text{total true edges}}$$



Expected performance improvement measured on a standalone model on Nvidia RTX A5000. The shape of the plot depends on the batch normalization function performance

The inference time can be improved by more than two times without compromising the efficiency.

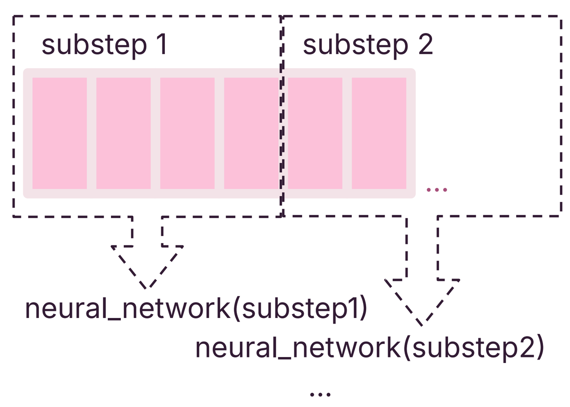
With the structured pruning we can maintain the efficiency while significant gains in the inference time are expected. Full implementation of the structured pruning within the GNN4ITk framework is in progress.

MEMORY CONSUMPTION

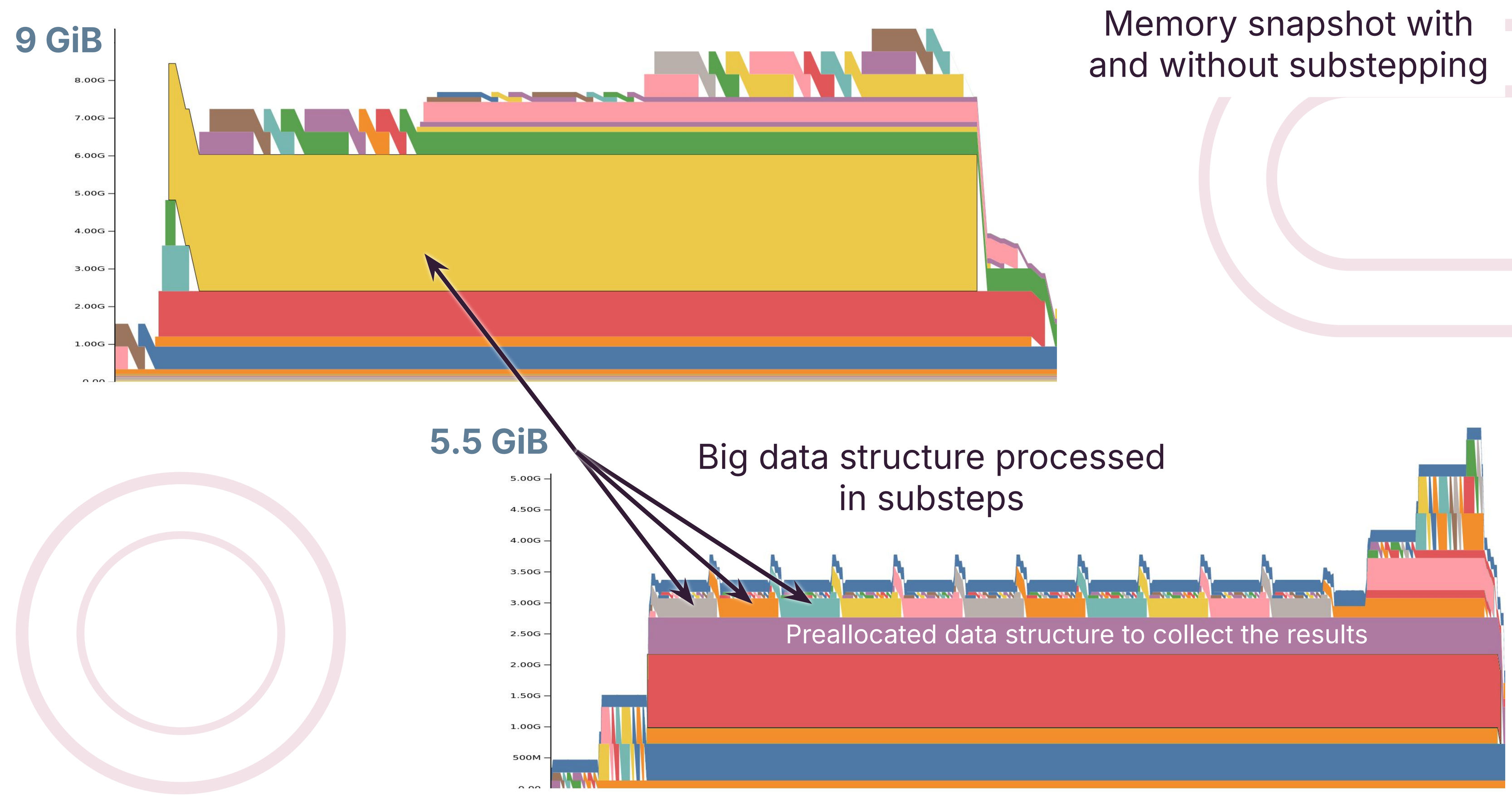
The input to the IGNN is an event represented as a **graph**, with node and edge features describing the geometric properties of hits (nodes), for example (x, y, z) position and track candidates (edges).

The size of the input graph determines the memory consumption of the GNN inference - an average graph has 1.95M edges and 316 k nodes [2]. However, during data-taking we must be able to reconstruct even the busiest events that can reach up to 3.7M edges [2] and requiring over **32 GiB** to process.

The high memory consumption is caused by the inputs to the IGNN steps that contain 12 features of the track edges encoded in the latent space of size *D* (by default D=128).



However, the evaluation of the network scoring the edges doesn't have to be done for all the graph edges at the same time - it is independent for each edge. The operation can be split into **substeps** executed sequentially of defined size, that would fit into a chosen GPU.



Memory snapshot with and without substepping

The substepping also improves the inference time by **20%** due to the performance scaling of pytorch concatenation operator.

Substepping mechanism allows to reduce the memory consumption of the IGNN inference and therefore use GPUs with less memory available. The track reconstruction efficiency is not affected.

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Federal Ministry of Education and Research

References:
[1] ATLAS ITk Track Reconstruction with a GNN-based pipeline *ATLAS Collaboration*
[2] Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain *H. Torres*
[3] Structured Pruning of Deep Convolutional Neural Networks *S. Anwar, K. Hwang , W. Sung*
[4] Saliency Pruner https://github.com/pytorch/pytorch/tree/main/torch/ao/pruning/_experimental/pruner
[5] https://twiki.cern.ch/twiki/bin/view/AtlasPublic/EFTrackingPublicResults

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 13E18CHA)

Poster theme based on a template from Slidesgo

ATLAS EXPERIMENT

CERN