WHEN LESS IS MORE:

GarNet with Attention

Towards a Lightweight Graph Neural Network for Reconstruction

Uzziel Perez on behalf of
Miriam Calvo Gomez, Xavier Vilasis Cardona (La Salle), et al.

CPAN, COMCHA, Valencia, Spain
November 20, 2025

GarNet Collaborators

Growing List of Collaborators

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

laSalle UNIVERSITAT RAMON LLULL

LHCb

PicoCal

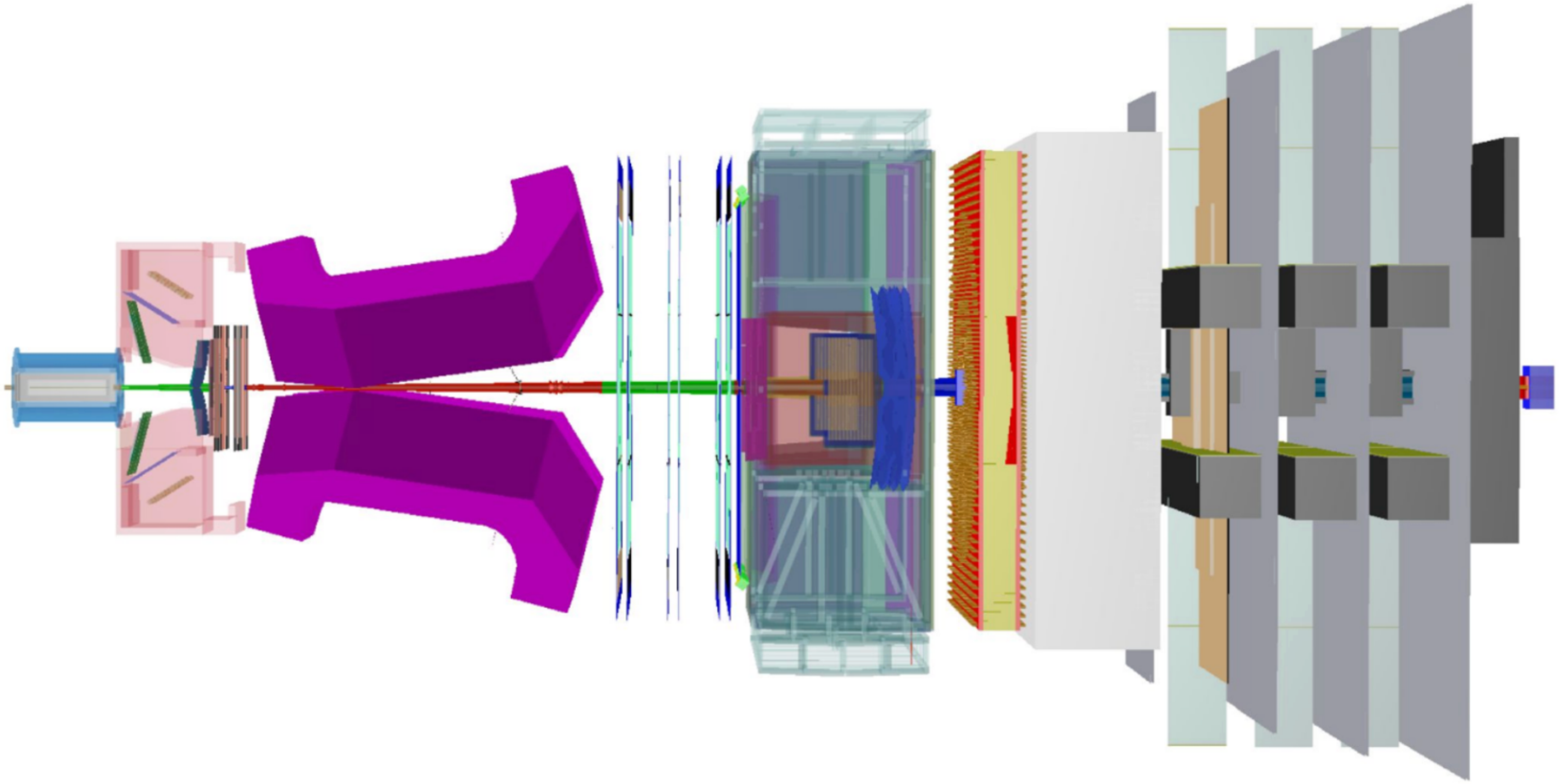
UNIVERSITAT DE BARCELONA

LHCb RTA

Uzziel Perez, Miriam Calvo Gomez, Xavier Vilasis Cardona (La Salle URL, Spain),
Felipe Luan Souza de Almeida (University of Barcelona, Spain),
Justin Bartz, Matthew Rudolph, Wren Vetens, (Syracuse University, USA),
Rafael Silva Coutinho (Centro Brasileiro de Pesquisas Físicas, Brazil)
Katya Govorkova (MIT), Ke Wei (Wuhan University, China)

Introduction

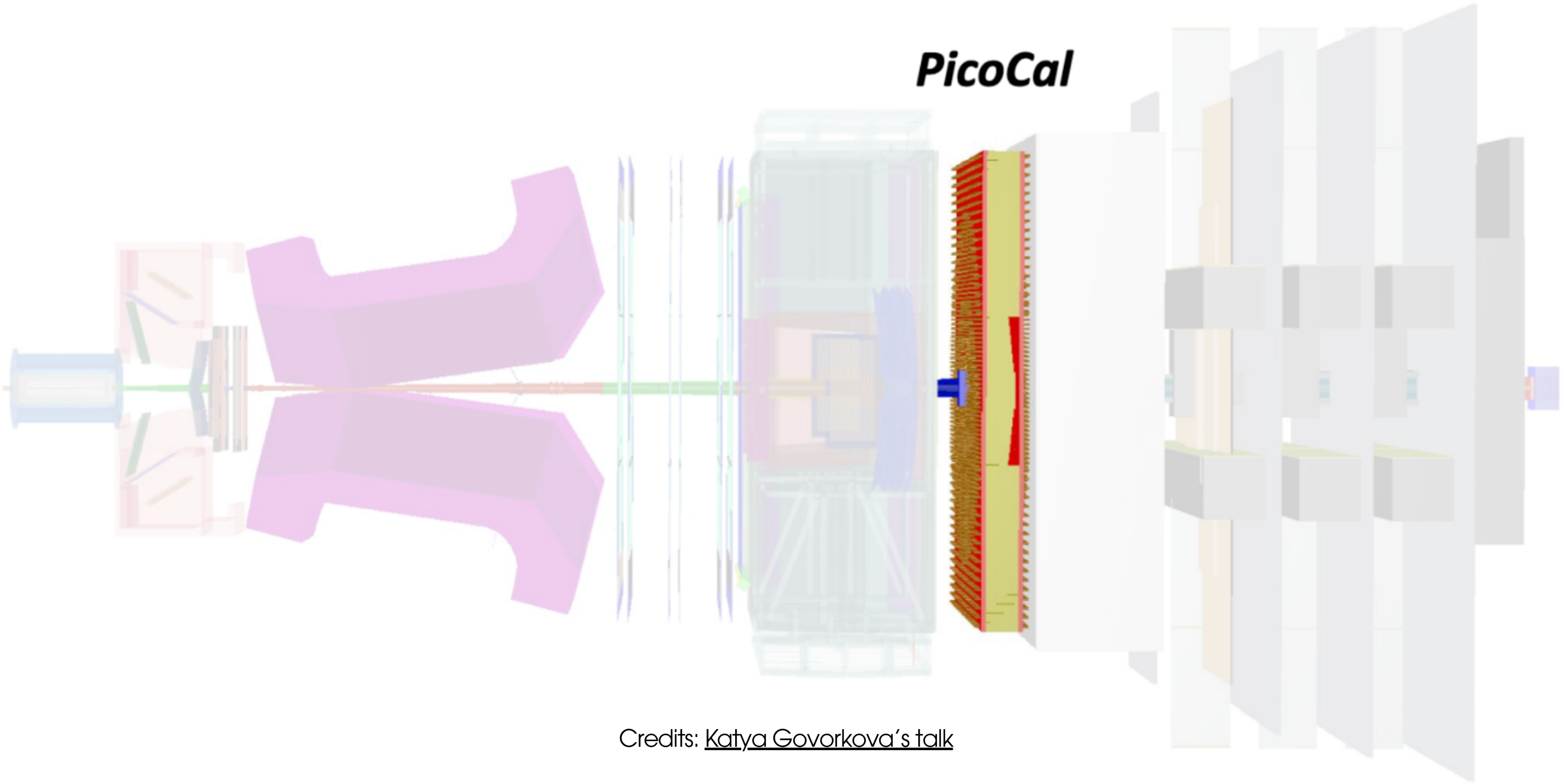
The **LHCb Upgrade II** redesigns LHCb to operate at 5x the instantaneous luminosity with a data rate of 200 Tb/s



Credits: Katya Govorkova's talk

Next Gen: PicoCal Detector

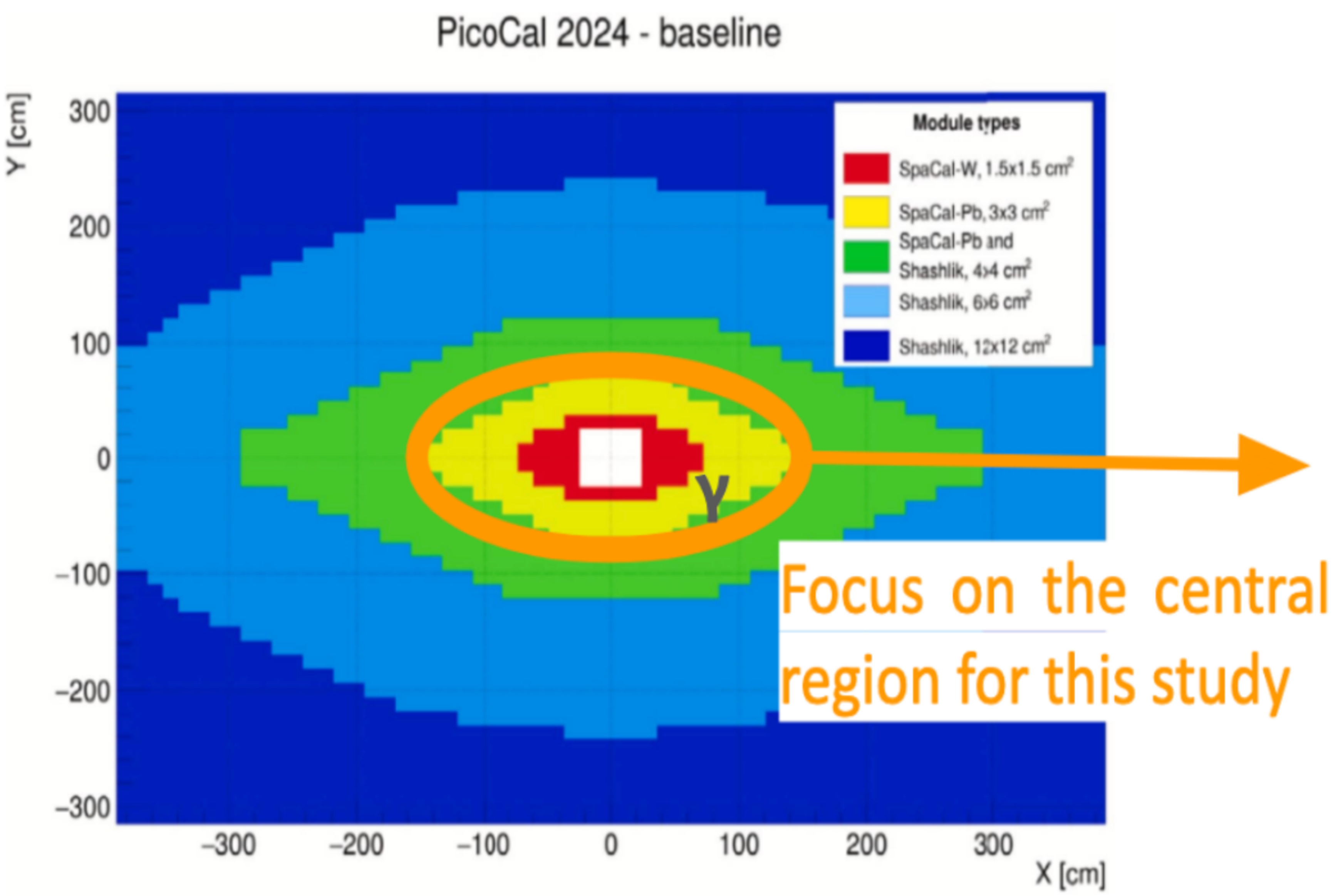
The **PicoCal** is the next-generation of the electromagnetic calorimeter for γ, e^-, pion (neutral) reconstruction, which includes timing information of **O(10) ps** precision

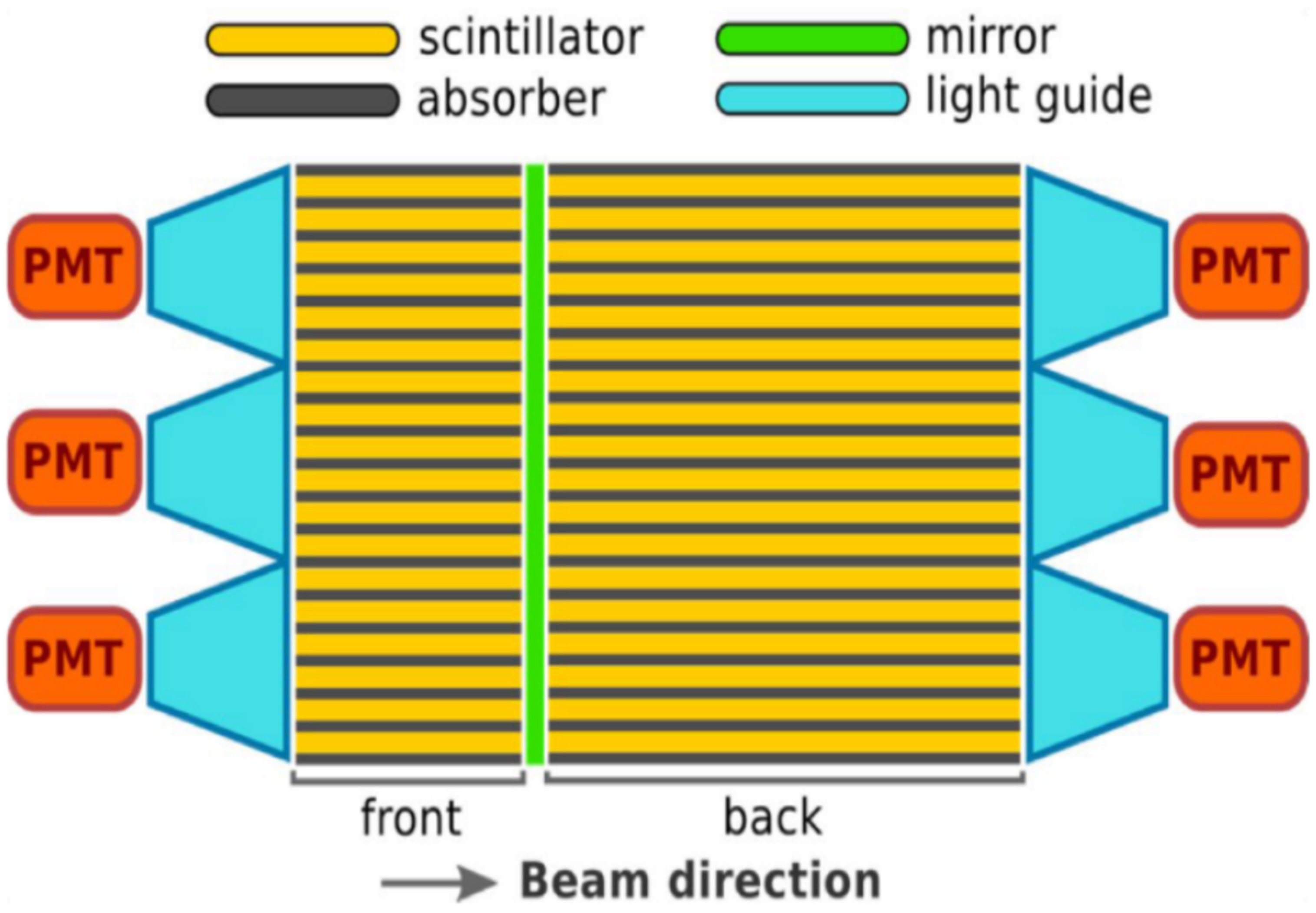


Credits: Katya Govorkova's talk

Next Gen: PicoCal Detector

The central region will be replaced with radiation-tolerant **SpaCal Modules** which have W/Pb absorbers and crystal/plastic scintillating fibers (LHCb-TDR-026)



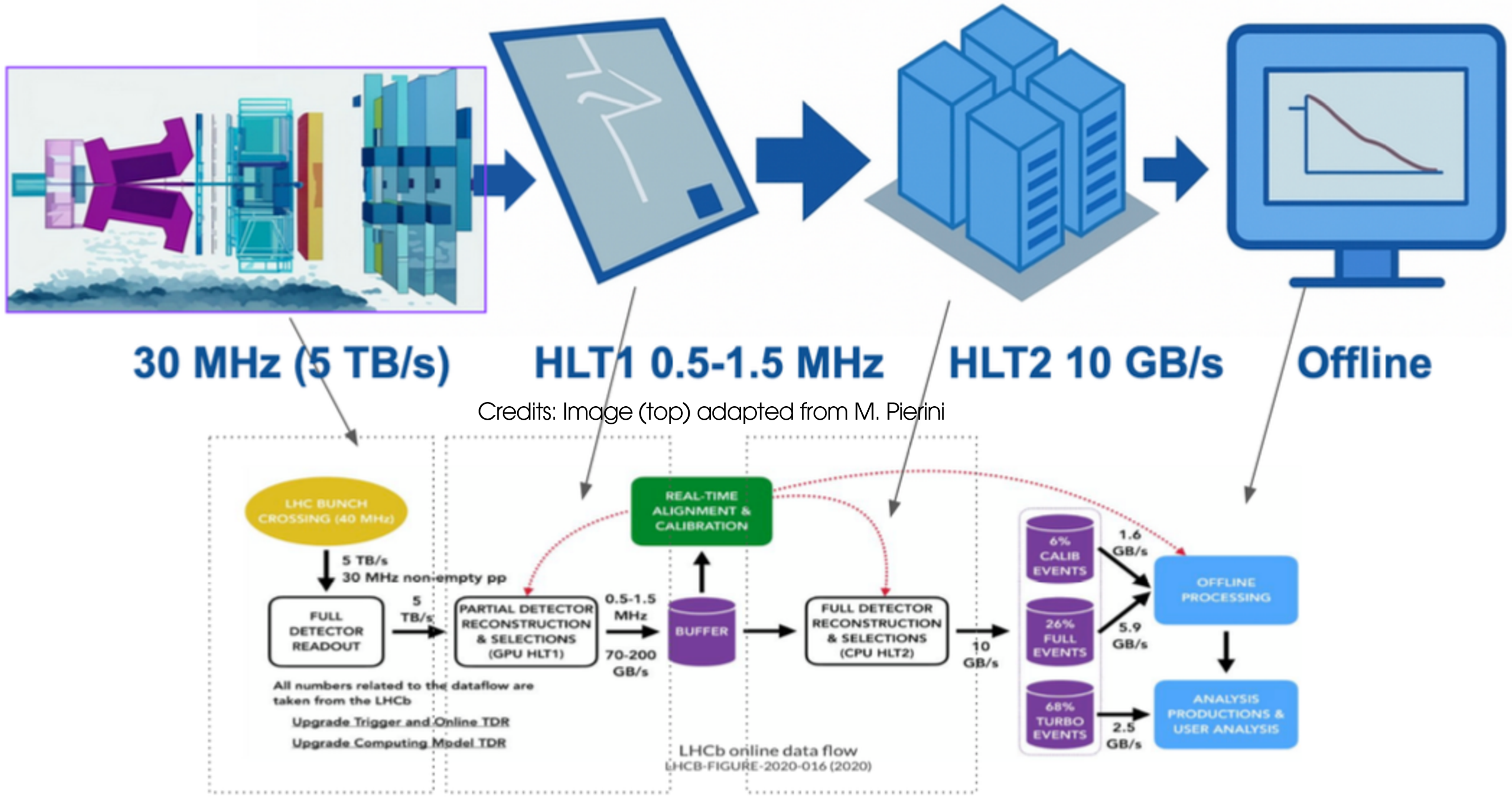


Credits: Katya Govorkova's talk

What are the challenges for Real-Time Reconstruction?

Latency and Throughput

What are the requirements and Bottlenecks?



Credits: Image (top) adapted from M. Pierini

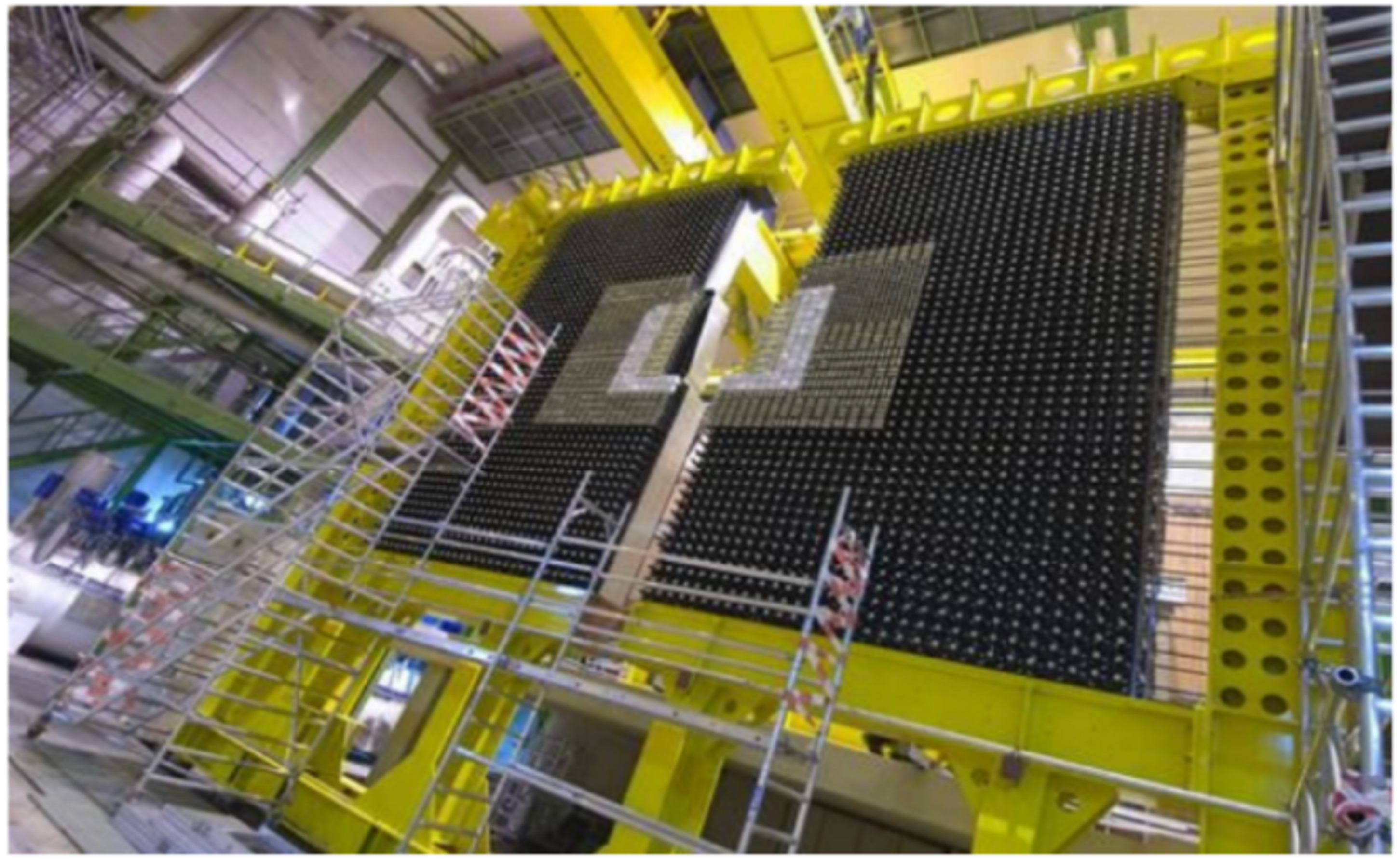
→ LHC bunch crossings occur every **25 ns** → Latency requirement: **~10 µs** w/ buffers absorbing **~0.1-1 ms**

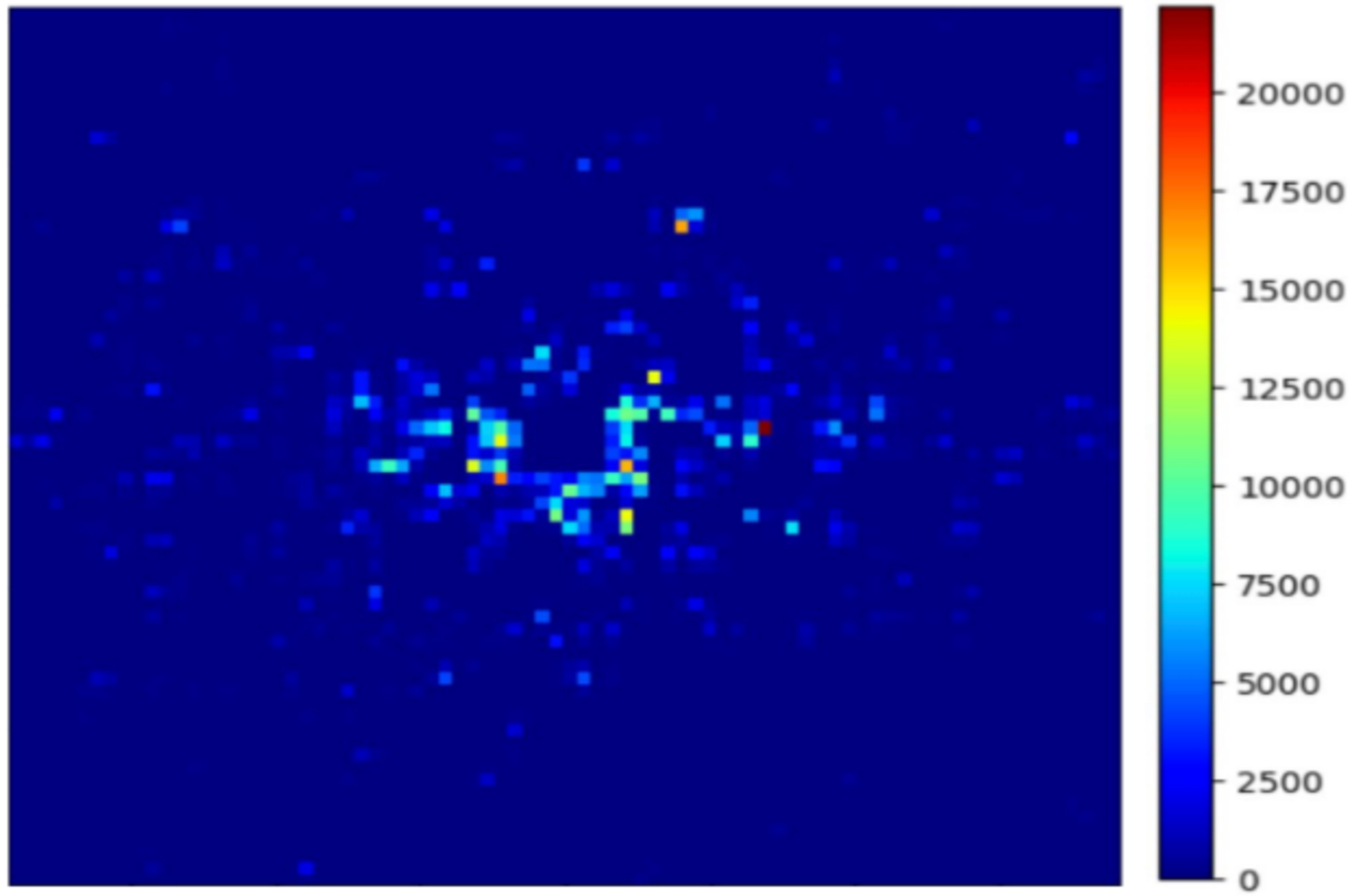
→ HLT2 has a 10GB/s throughput bottleneck. To avoid backpressure → have low latency reconstruction

Evolving Reconstruction Algorithm

Evolution of Reconstruction Algorithms

→ The current Graph Clustering Algorithm is less than 60% faster than the legacy Cellular Automata
→ GNNs seem like natural successors for future reconstruction



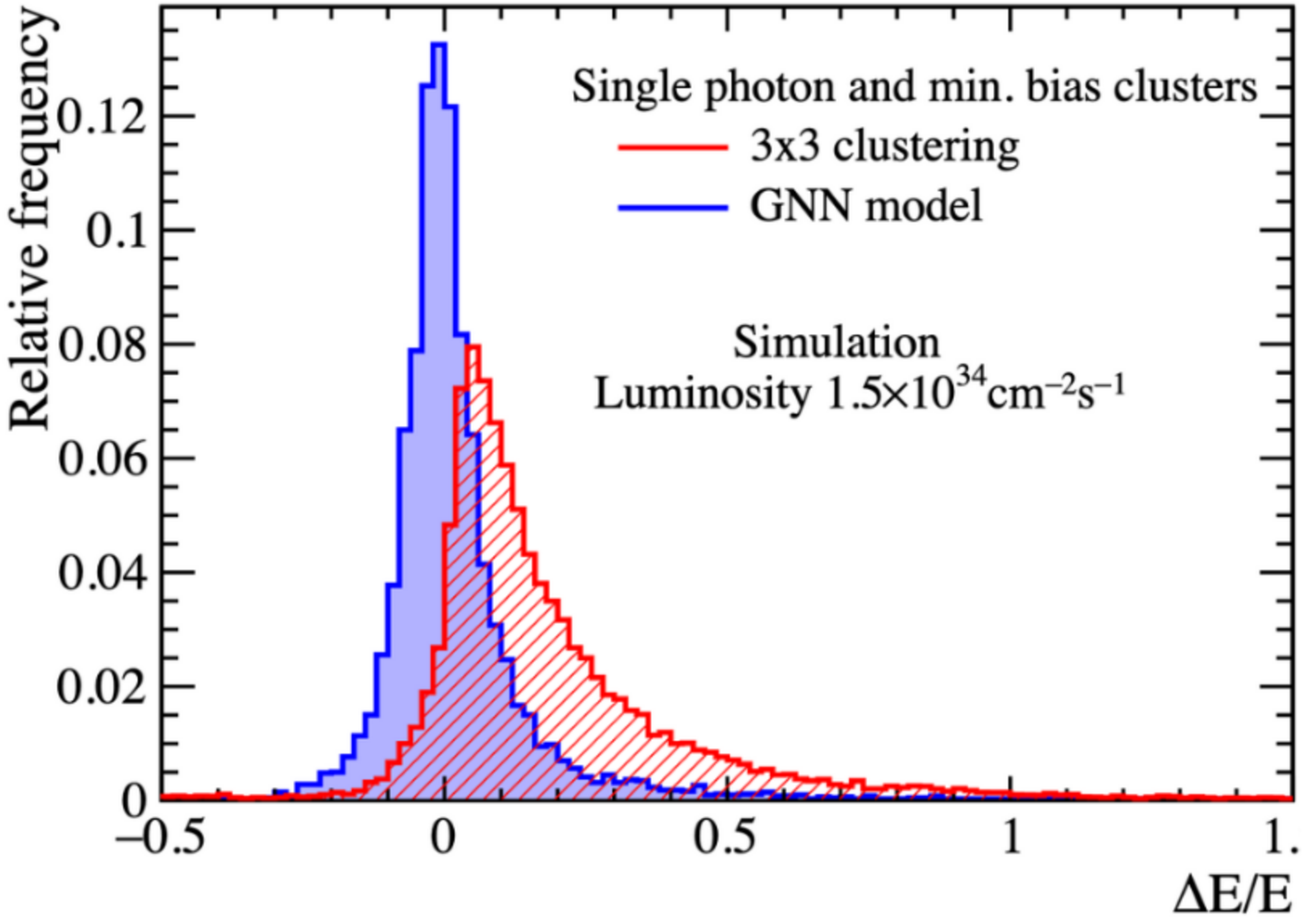


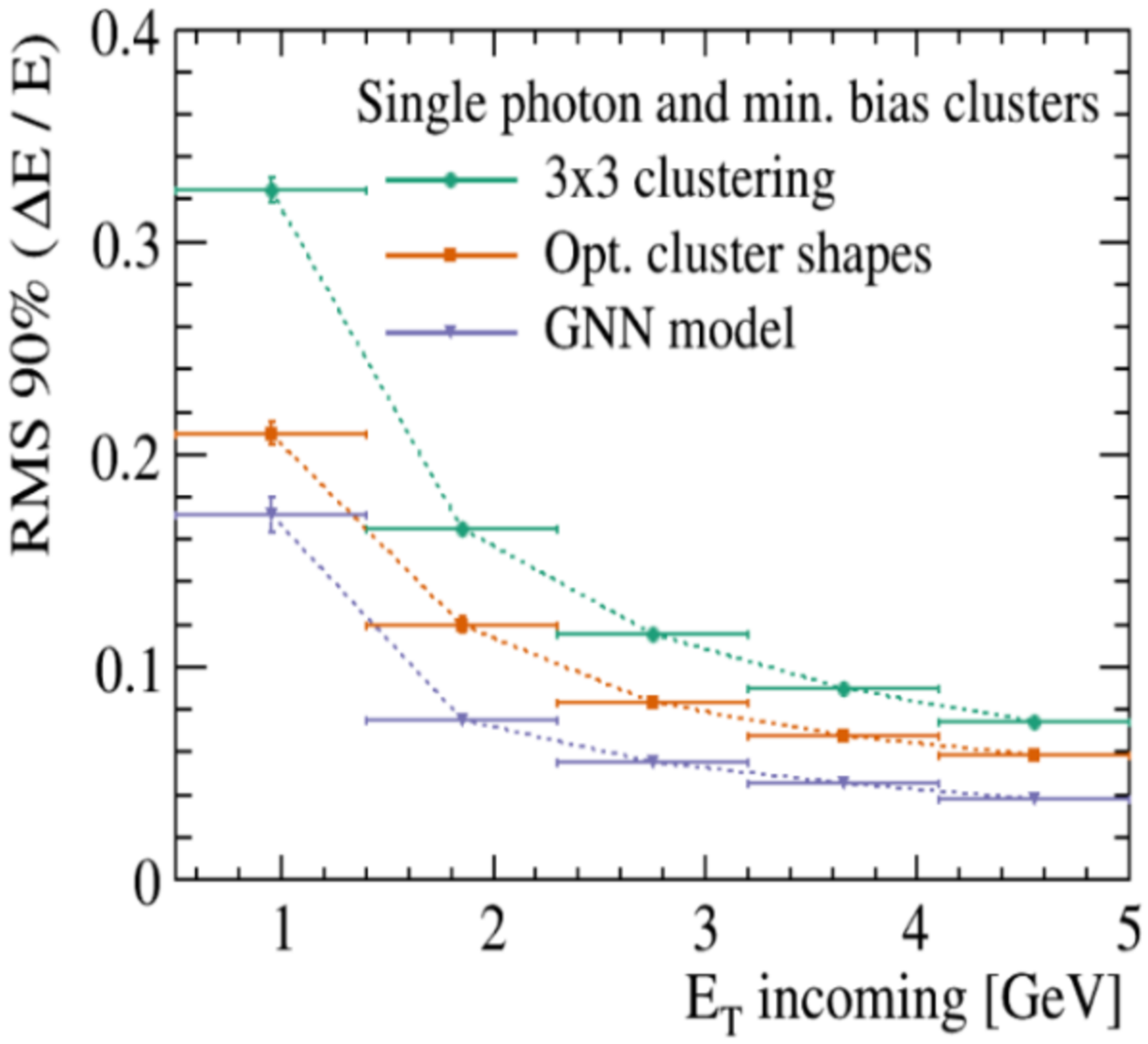
Cellular Automaton → Graph Clustering → Graph Neural Networks

PAST PRESENT FUTURE

Why Graph Neural Networks?

→ Keeps the graph structure of clustering but make the aggregation rules learnable
→ More adaptable for handling irregular detector geometry



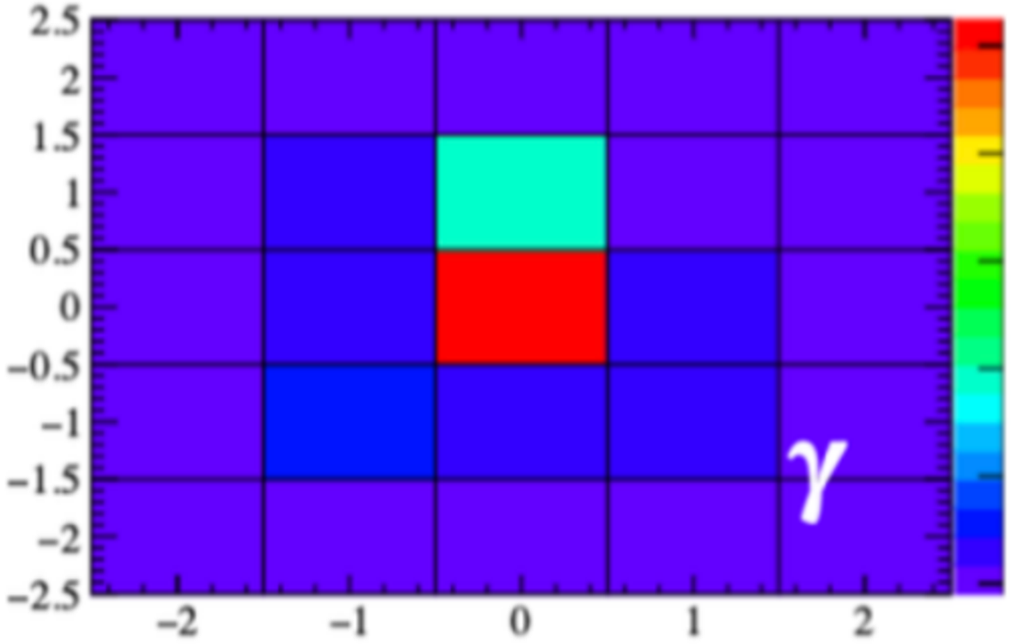


The lower the better!

Data Preprocessing

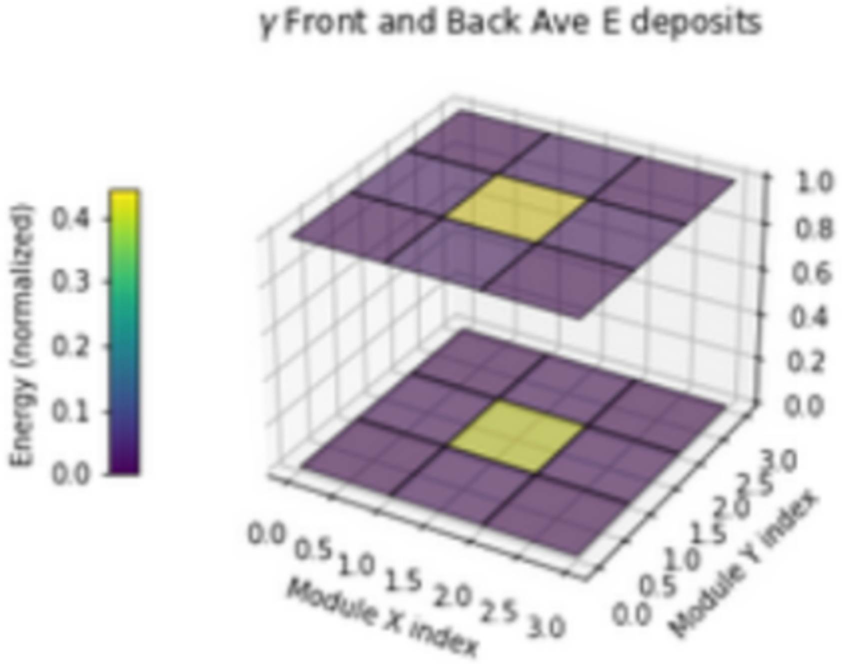
→ Spacal Simulation with Single Photons (particle gun) and minbias clusters
→ Raw PicoCal Data converted to KNN-based graph → node (E, position), edge (spatial links), and global (seed position) features

$\mathcal{L}_{\text{beam}} = 1.5 \times 10^{-34} cm^{-2} s^{-1}$

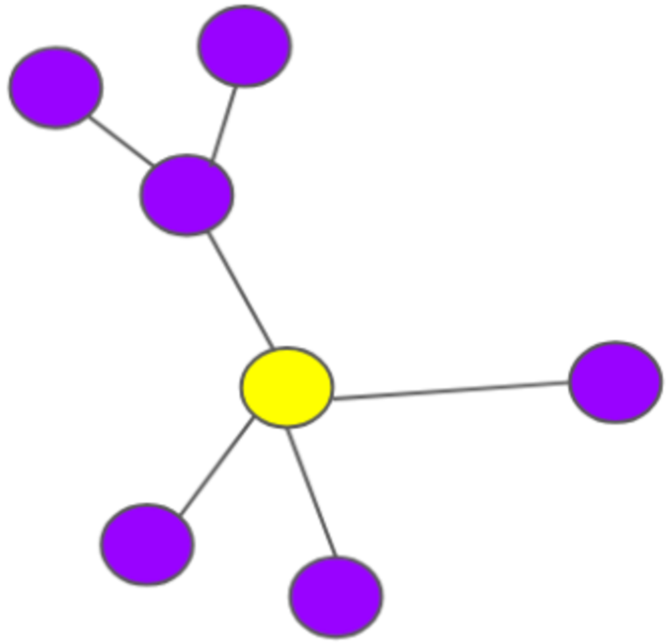


$0.5 \leq E_T \leq 5\,\text{GeV}$

SpaCal Simulation



Choose 3 x 3 clusters
w/ Front and Back Energies

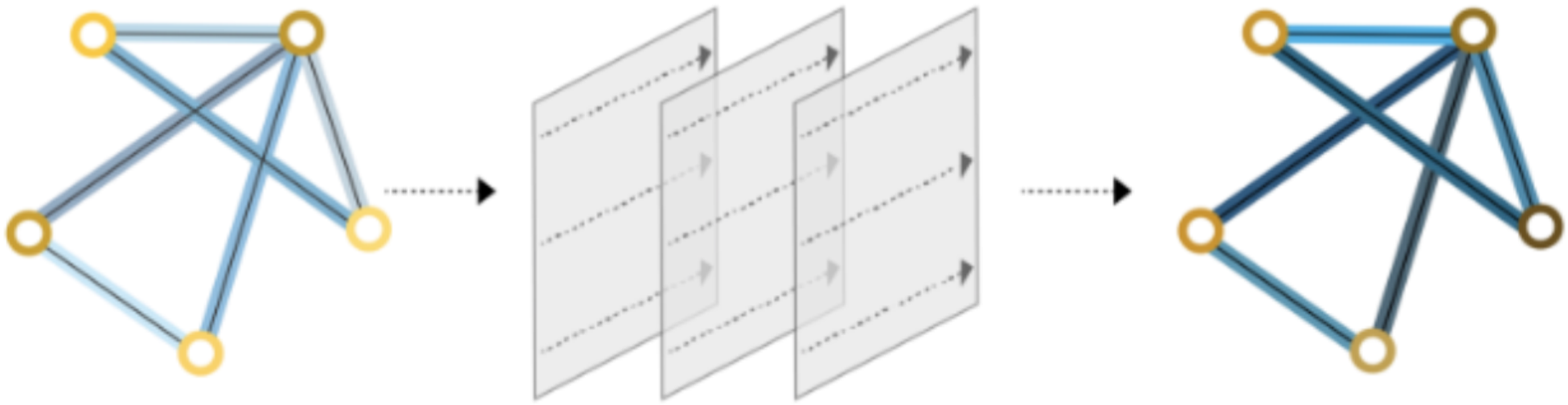


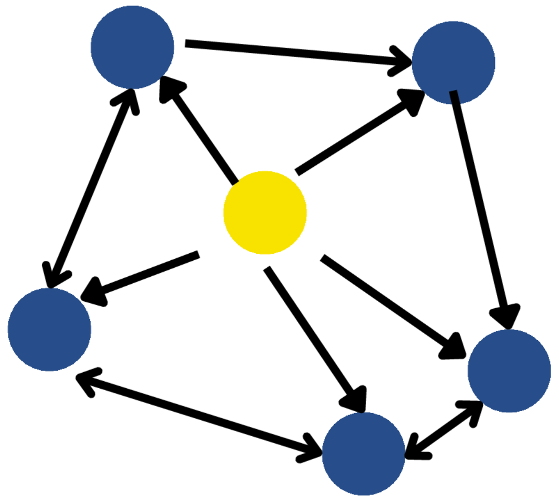
Graph Inputs:
Nodes: Etot/cell, Efront, Eback, x, y, relative position to seed
Edges: Δx, Δy, ΔE, dij
Global: Etot/particle

100k in FULL ECAL, 12k in the Spacal Region

GNN Fundamentals

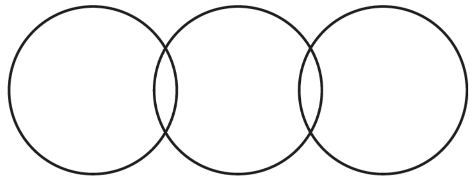
→ Input data is also a graph
→ Initial node features updated by message passing layers
→ Nodes are updated by applying a FF-NN on a previous state and received messages
→ Encoder-Decoder is a typical GNN architecture

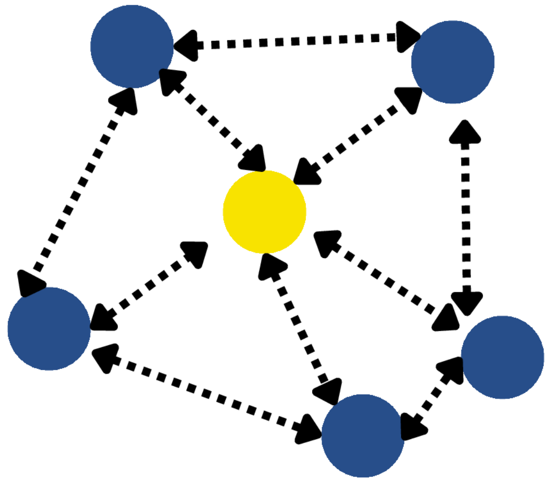




Full Message Passing

GNN Flavours



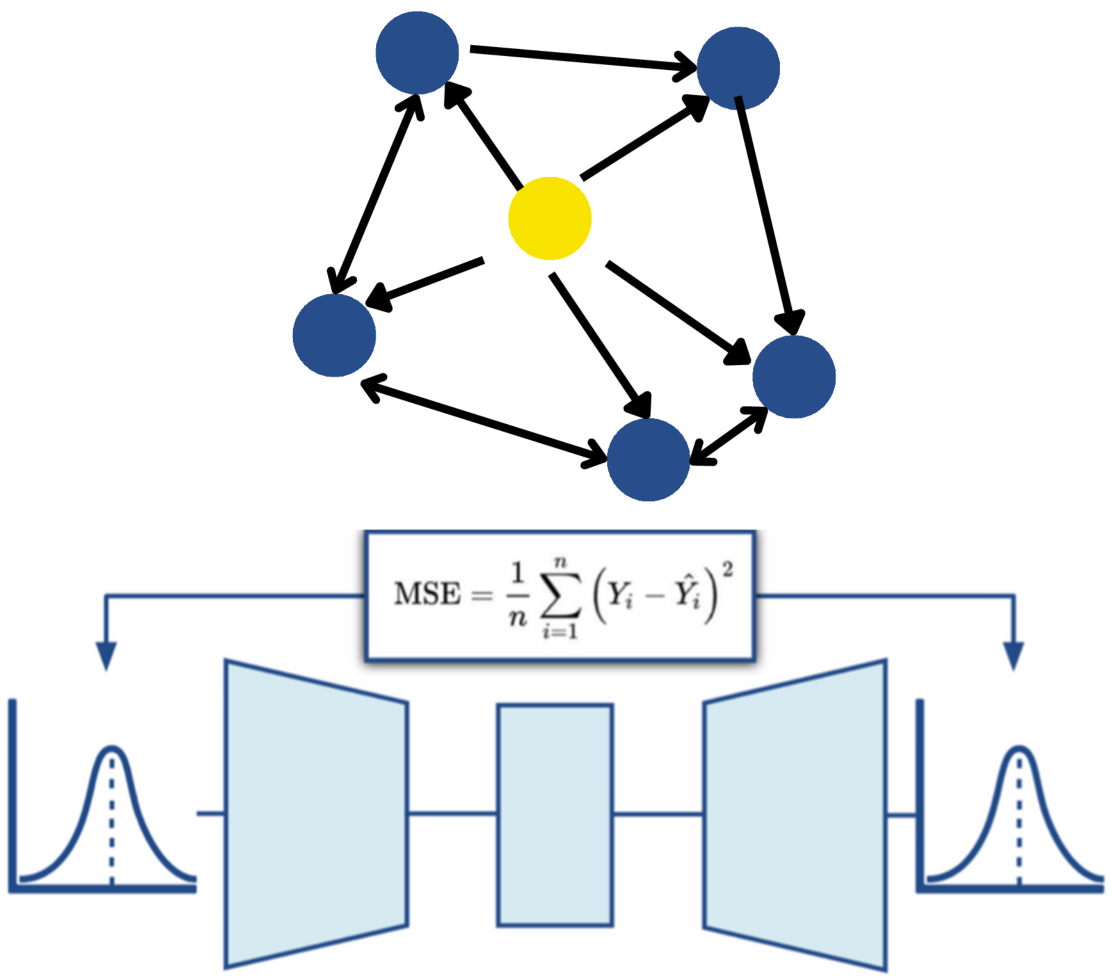


GarNet

GNN Flavours

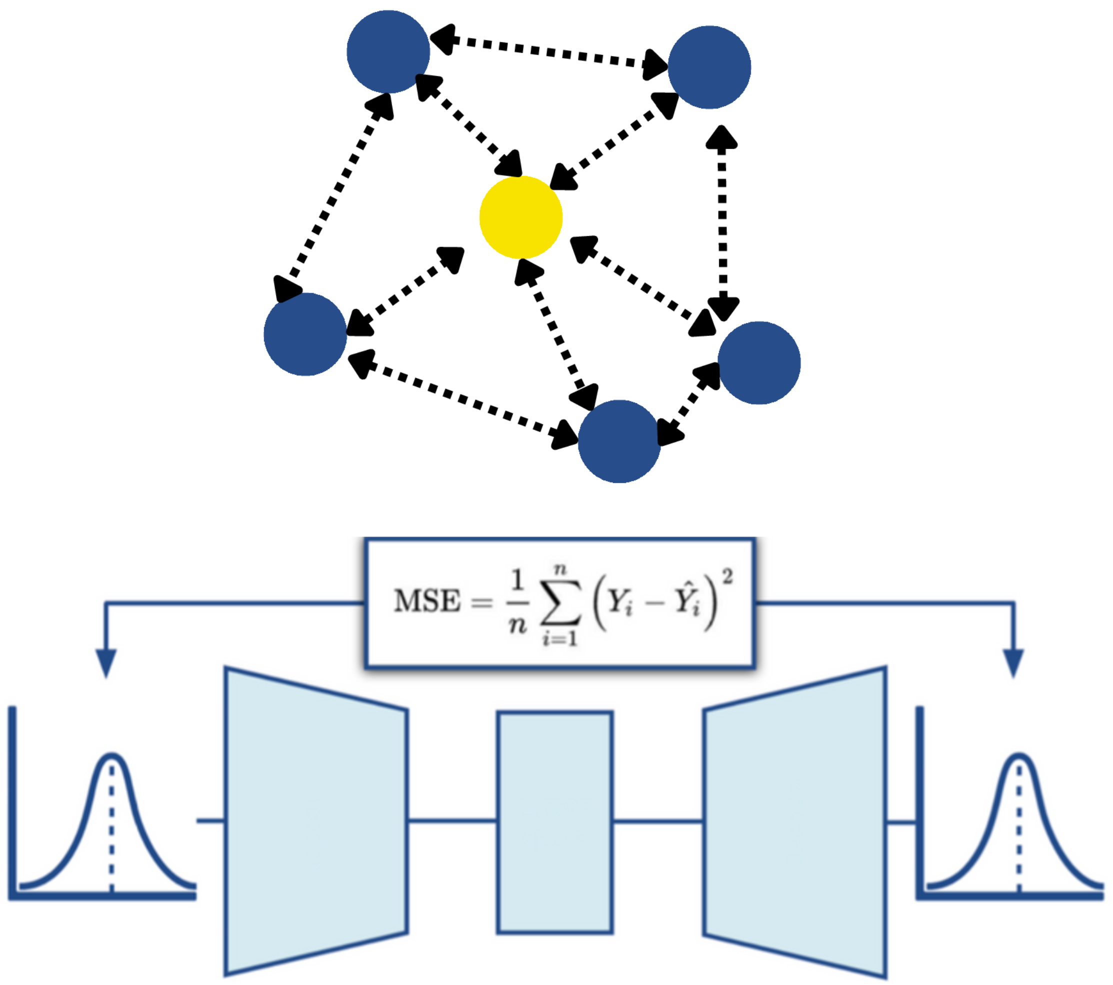
In a Nutshell

With Edges (Δx, Δy, ΔE)



Encoder-Processor-Decoder

No Edges

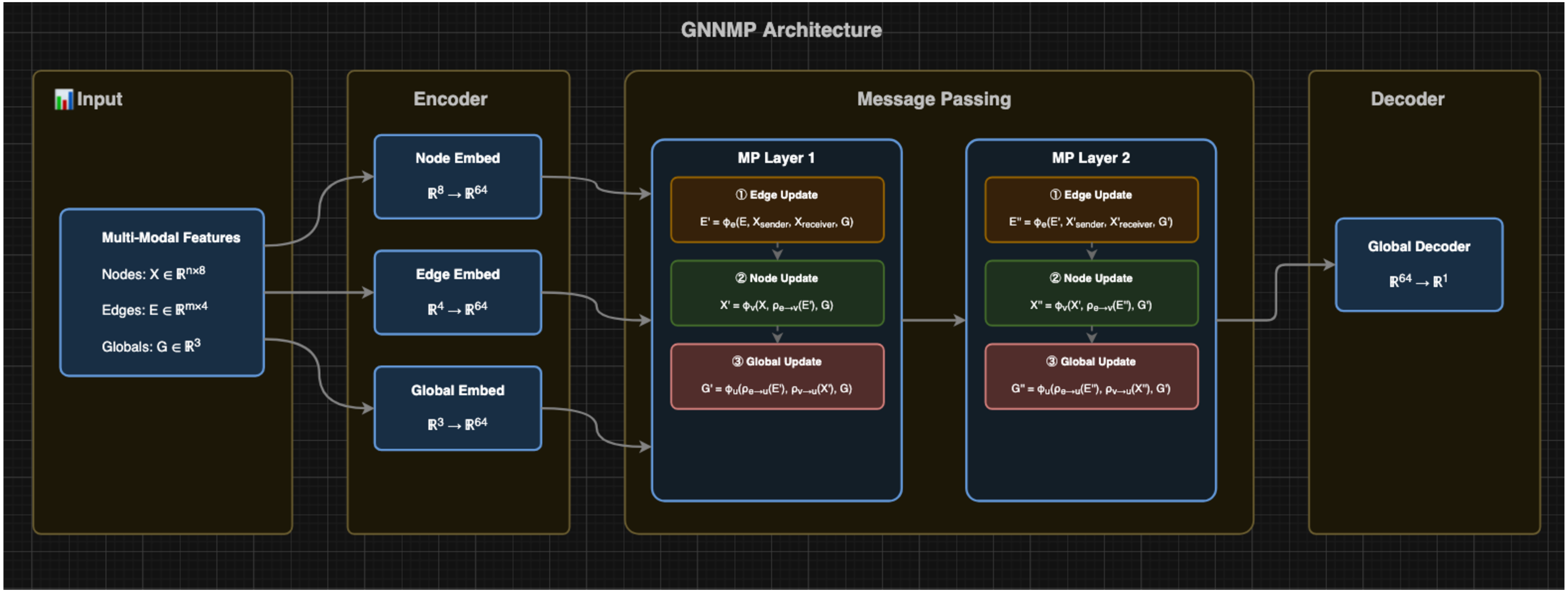


GarNet Layers 1 + 2 + Global Mean Pool

GNN w/ Full Message Passing

Full Message Passing → GarNet (see Felipe's Talk!)

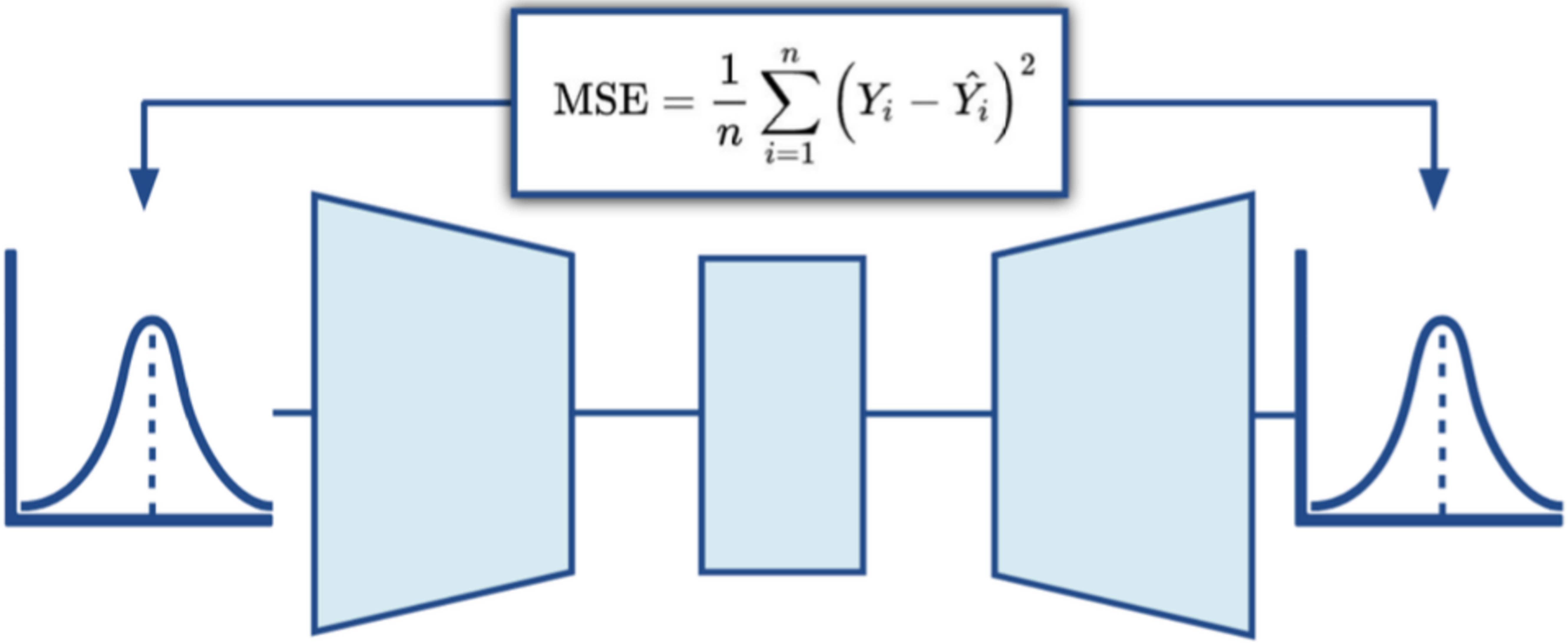
→ Encoder: Projects all heterogeneous features into a common embedding space
→ Message Passing: Iterative information propagation between neighbor elements
→ Decoder: Aggregates learned representation to predict shower energy



GNNMP Training

Full Message Passing

→ **MSE Loss:** Mean squared error between predicted and true energy
→ **Backpropagation:** Adam Optimizer
→ **Early Stopping:** Stops training if validation loss does not improve for a certain number of epochs

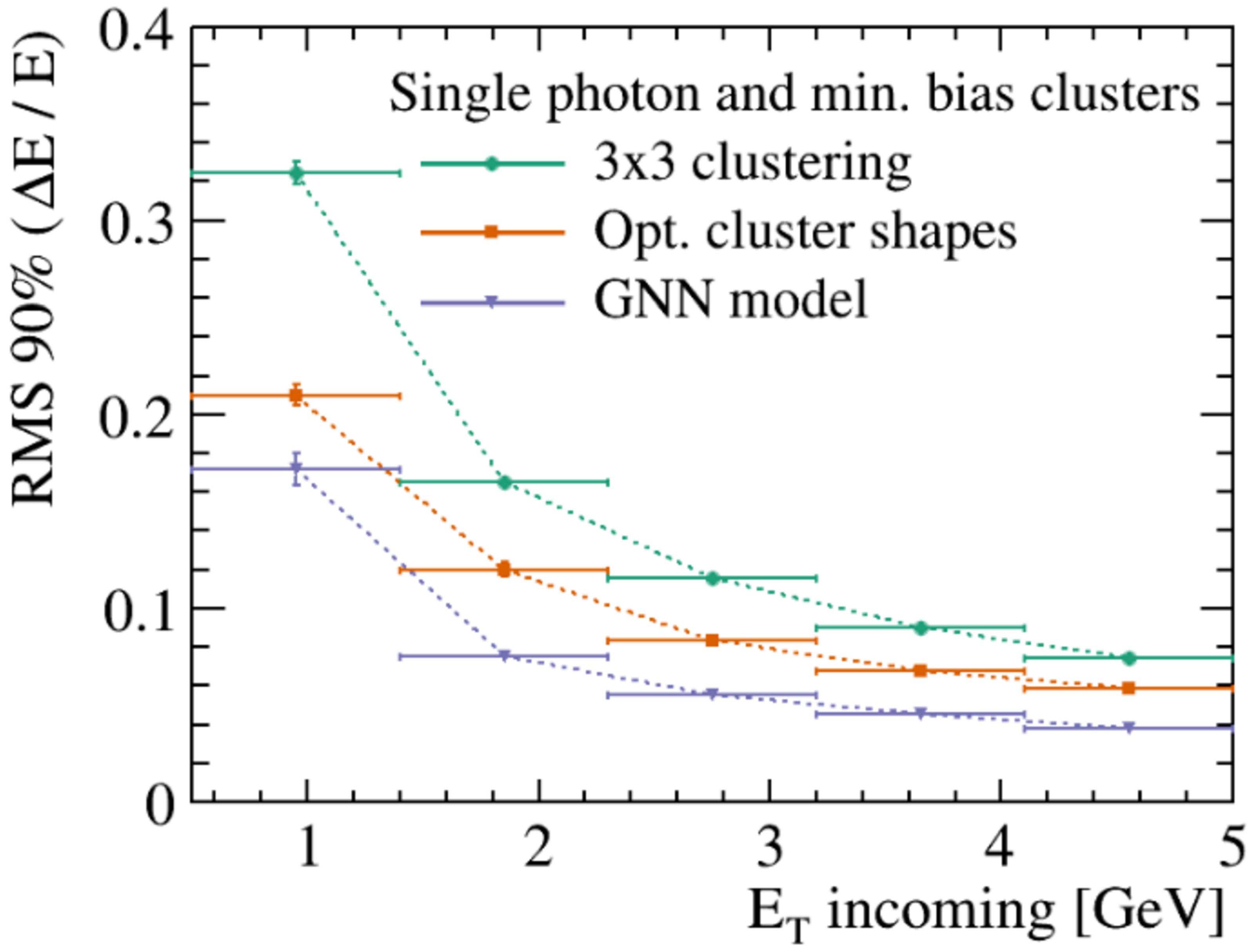


GNNMP Energy Resolution

Superior energy resolution

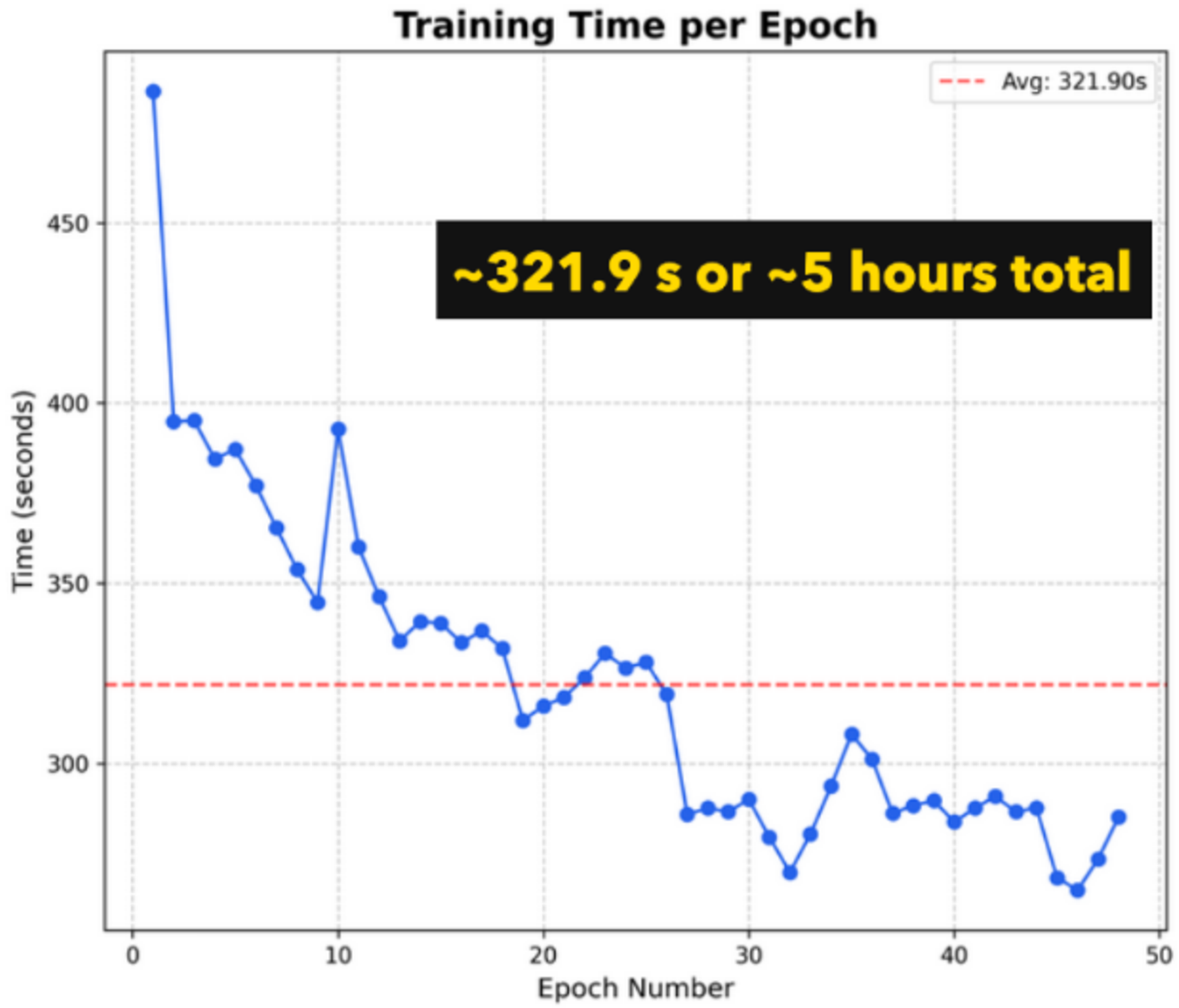
→ Note: 3x3 Clustering resolution of Cellular Automata and Graph Clustering are equivalent

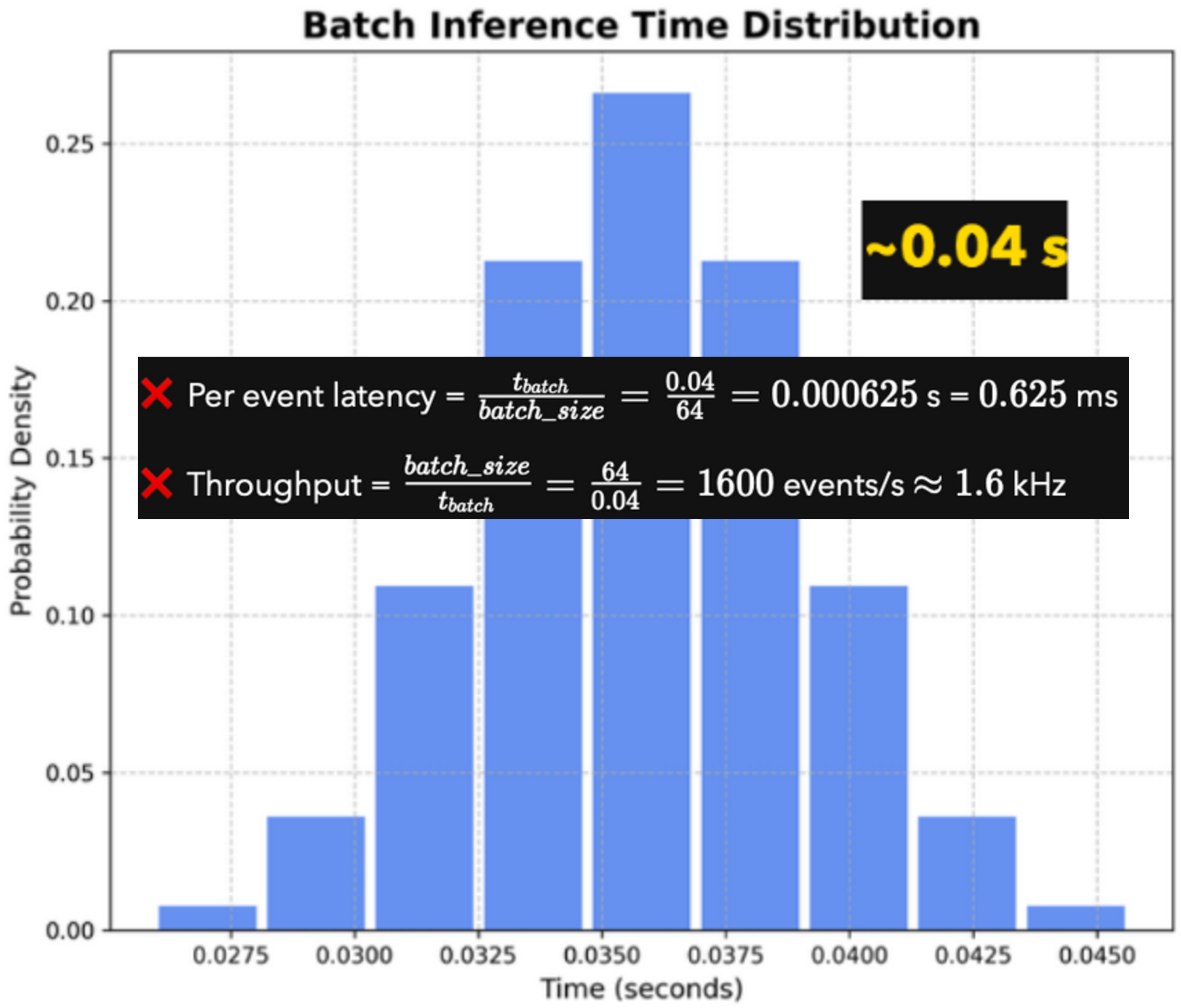
→ GNNMP Model better



Time-expensive

Too close to the ~1 ms latency benchmark requirement for a single-photon cluster!





Disclaimer: lxplus CPU times (Xeon Silver 4216), DDP-gloo with 4 processes

CMS Reconstruction

Taking inspiration from two papers from CMS on *distance-weighted graph neural networks* and their *FPGA implementations,* we experimented with a similar variant dubbed as **GarNet with Attention**

Learning representations of irregular particle-detector geometry with distance-weighted graph networks

Shah Rukh Qasim[1,2] (14beesqasim@seecs.edu.pk), Jan Kieseler[1] (jan.kieseler@cern.ch), Yutaro Iiyama[1] (yutaro.iiyama@cern.ch), and Maurizio Pierini[1] (maurizio.pierini@cern.ch)

[1] CERN, Geneva, Switzerland
[2] National University of Sciences and Technology, Islamabad, Pakistan

July 25, 2019

Abstract. We explore the use of graph networks to deal with irregular-geometry detectors in the context of particle reconstruction. Thanks to their representation-learning capabilities, graph networks can exploit the full detector granularity, while natively managing the event sparsity and arbitrarily complex detector geometries. We introduce two distance-weighted graph network architectures, dubbed GarNet and GravNet layers, and apply them to a typical particle reconstruction task. The performance of the new architectures is evaluated on a data set of simulated particle interactions on a toy model of a highly granular calorimeter, loosely inspired by the endcap calorimeter to be installed in the CMS detector for the High-Luminosity LHC phase. We study the clustering of energy depositions, which is the basis for calorimetric particle reconstruction, and provide a quantitative comparison to alternative approaches. The proposed algorithms provide an interesting alternative to existing methods, offering equally performing or less resource-demanding solutions with less underlying assumptions on the detector geometry and, consequently, the possibility to generalize to other detectors.

1 Introduction

Traditionally, Machine Learning (ML) techniques are a key ingredient to event processing at particle colliders, employed in tasks such as particle reconstruction (clustering), identification (classification), and energy or direction measurement (regression) in calorimeters and tracking devices. The first applications of Neural Networks to High Energy Physics (HEP) date back to the '80s [1, 2, 3, 4]. Starting with the MiniBooNE experiment [5], Boosted Decision Trees became the state of the art, and played a crucial role in the discovery of the Higgs boson by the ATLAS and CMS experiments [6]. Recently, a series of studies on different aspects of LHC data taking and data processing workflows have demonstrated the potential of Deep Learning (DL) in collider applications, both as a way to speed up current algorithms and to improve their performance. Nevertheless, the list of DL models actually deployed in the centralized workflows of the LHC experiments remains quite short.[1] Many of these studies, which are typically proof-of-concept demonstrations, are based on convolutional neural networks (CNN) [10], which perform computing vision tasks by applying translation-invariant kernels to *raw* digital images. CNN architectures applied on HEP data thus imposes a requirement for the particle detectors to be represented as regular arrays of sensors. This requirement, common to many of the approaches described in Section 2, creates problems for realistic applications of CNNs in collider experiments.[2]

In this work, we propose novel Deep Learning architectures based on graph networks to improve the performance and reduce the execution time of typical particle-reconstruction tasks, such as cluster reconstruction and particle identification. In contrast to CNNs, graph networks can learn optimal detector-hits representations without making specific assumptions on the detector geometry. In particular, no data preprocessing is required, even for detectors with irregular geometries. We consider the specific case of particle reconstruction in calorimeters, for which this characteristic of graph networks may become especially relevant in the near future. In view of the High-Luminosity LHC phase, the endcap calorimeter of the CMS detector will be replaced by a novel-design digital calorimeter, the High Granularity Calorimeter (HGCAL),

[1] As an example, at the moment such a list for the CMS experiment consists of a set of b-tagging algorithms [7, 8] and a data quality monitoring algorithm for the muon drift tube chambers [9]. Other applications exist at the analysis level, downstream from the centralized event processing. In data analyses, one typically considers abstract four-momenta and not the low-level quantities such as detector hits, making the use of DL techniques easier.

[2] The picture is completely different in other HEP domains. For instance, CNNs have been successfully deployed in neutrino experiments, where the regular-array assumption meets the geometry of a typical detector.

FERMILAB-PUB-20-405-E-SCD

Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics

Yutaro Iiyama
University of Tokyo
Tokyo 113-0033, Japan

Gianluca Cerminara, Abhijay Gupta, Jan Kieseler, Vladimir Loncar,[*] Maurizio Pierini, Shah Rukh Qasim,[†] Marcel Rieger, Sioni Summers, Gerrit Van Onsem, Kinga Anna Wozniak[‡]
European Organization for Nuclear Research (CERN)
CH-1211 Geneva 23, Switzerland

Jennifer Ngadiuba
California Institute of Technology
Pasadena, CA 91125, USA

Giuseppe Di Guglielmo
Columbia University
New York, NY 10027, USA

Javier Duarte
University of California San Diego
La Jolla, CA 92093, USA

Philip Harris, Dylan Rankin
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Sergo Jindariani, Mia Liu, Kevin Pedro, Nhan Tran[§]
Fermi National Accelerator Laboratory
Batavia, IL 60510, USA

Edward Kreinar
HawkEye360
Herndon, VA 20170, USA

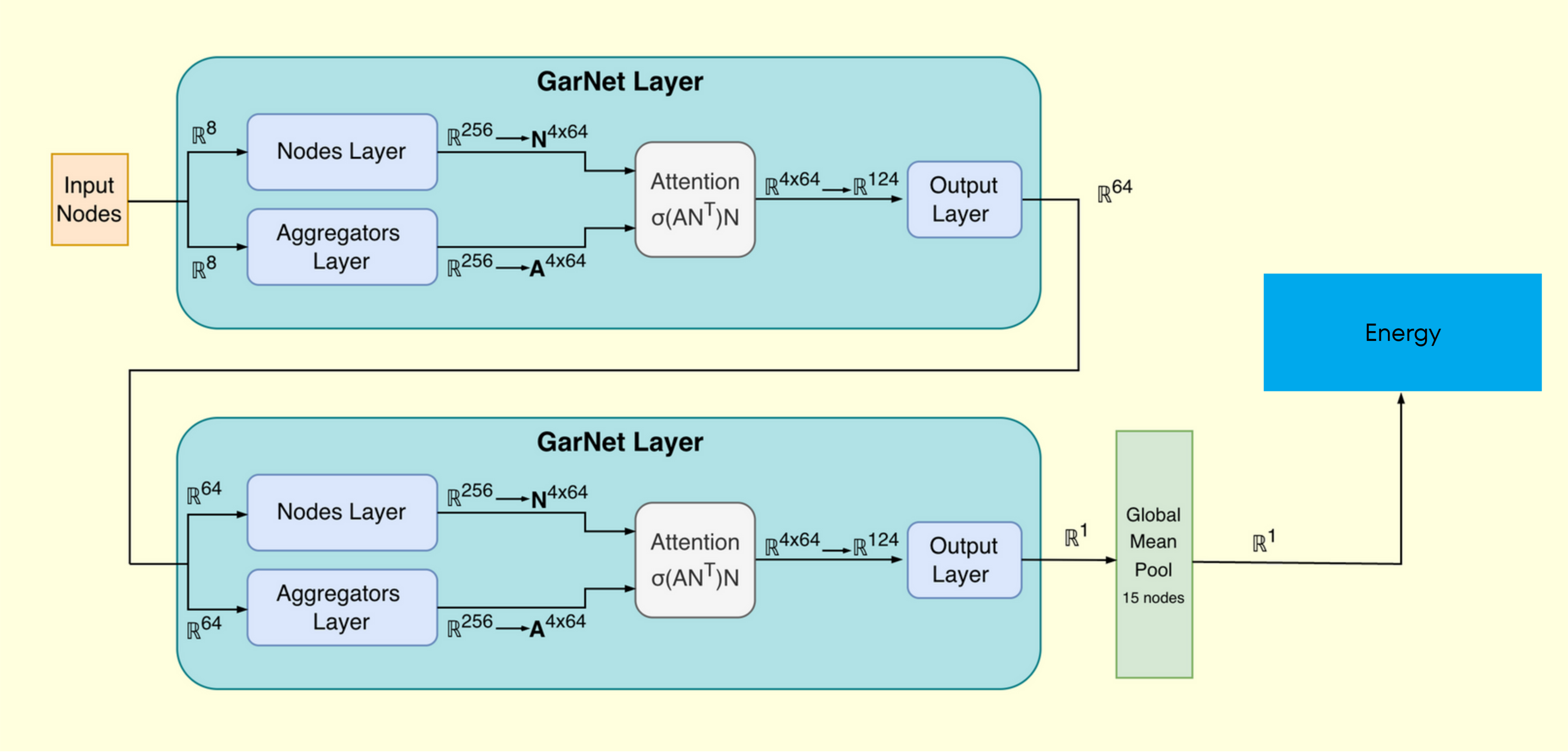
Zhenbin Wu
University of Illinois at Chicago
Chicago, IL 60607, USA

February 8, 2021

[*] Also at Institute of Physics Belgrade, Pregrevica 118, Belgrade, Serbia
[†] Also at Manchester Metropolitan University, Manchester M15 6BH, UK
[‡] Also at University of Vienna, 1010 Vienna, Austria
[§] Also at Northwestern University, Evanston, IL 60208, USA

GarNet with Attention

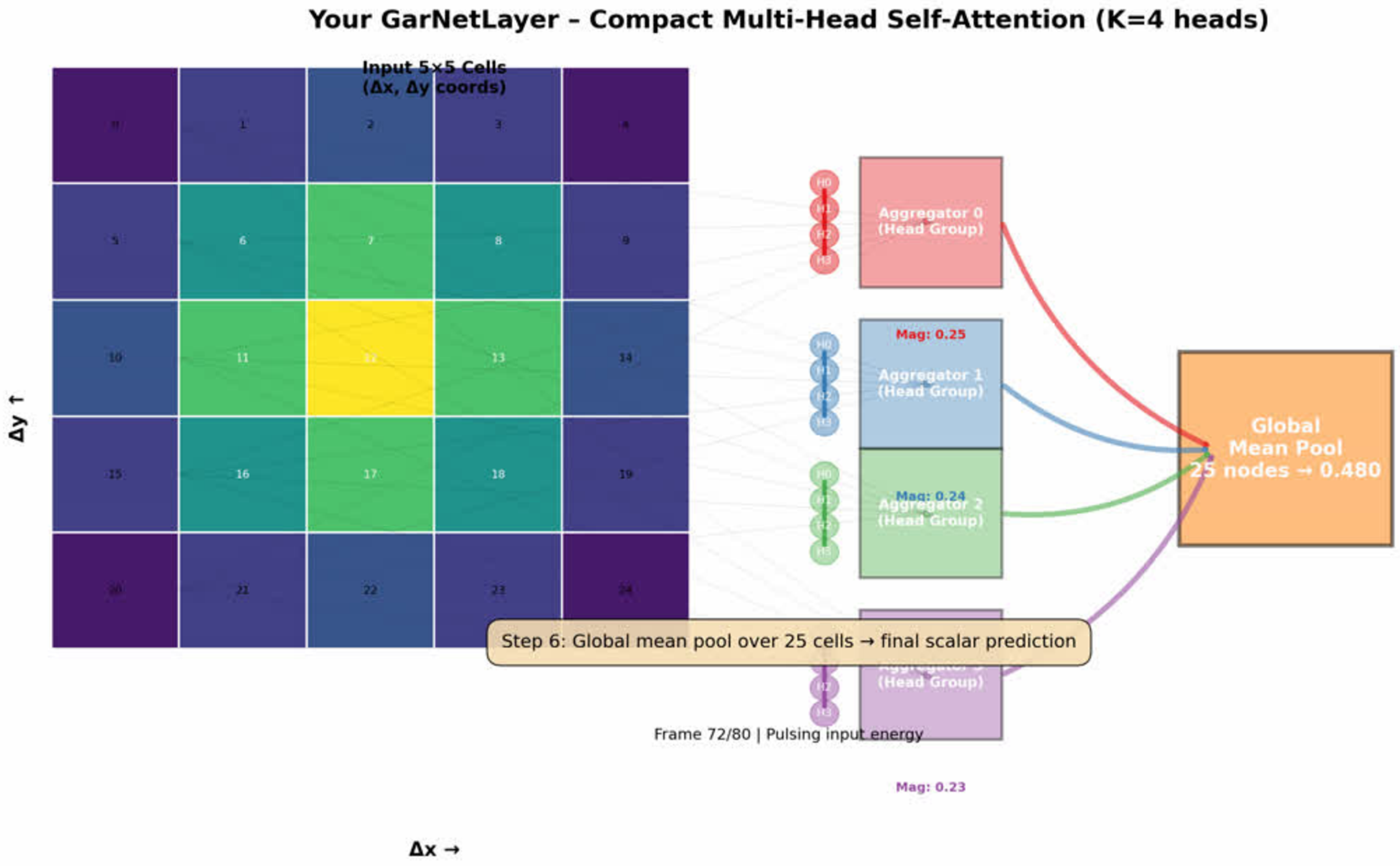
→ Explicit edge features, i.e. relative distances and energies removed
→ **Encoder-Processor-Decoder** replaced with simpler **2 GarNet layers** and a **Global Mean Pool**
→ GarNet Layer: Learned aggregators + distance attention to predict incident particle energy from 3×3 front/back cell energies



How does the attention mechanism work in GarNet?

Attention Mechanism

→ Each node represents a detector cell
→ **Attention weights**: Each node compared to another and finds "similarities", $\alpha_{ij} = \texttt{softmax}(q_i \cdot k_j)$
→ **Output:** weighted sum of all node features $h_i = \Sigma_j \alpha_{ij} \cdot v_j$
→ The model learns to focus on relevant patterns across the detectors

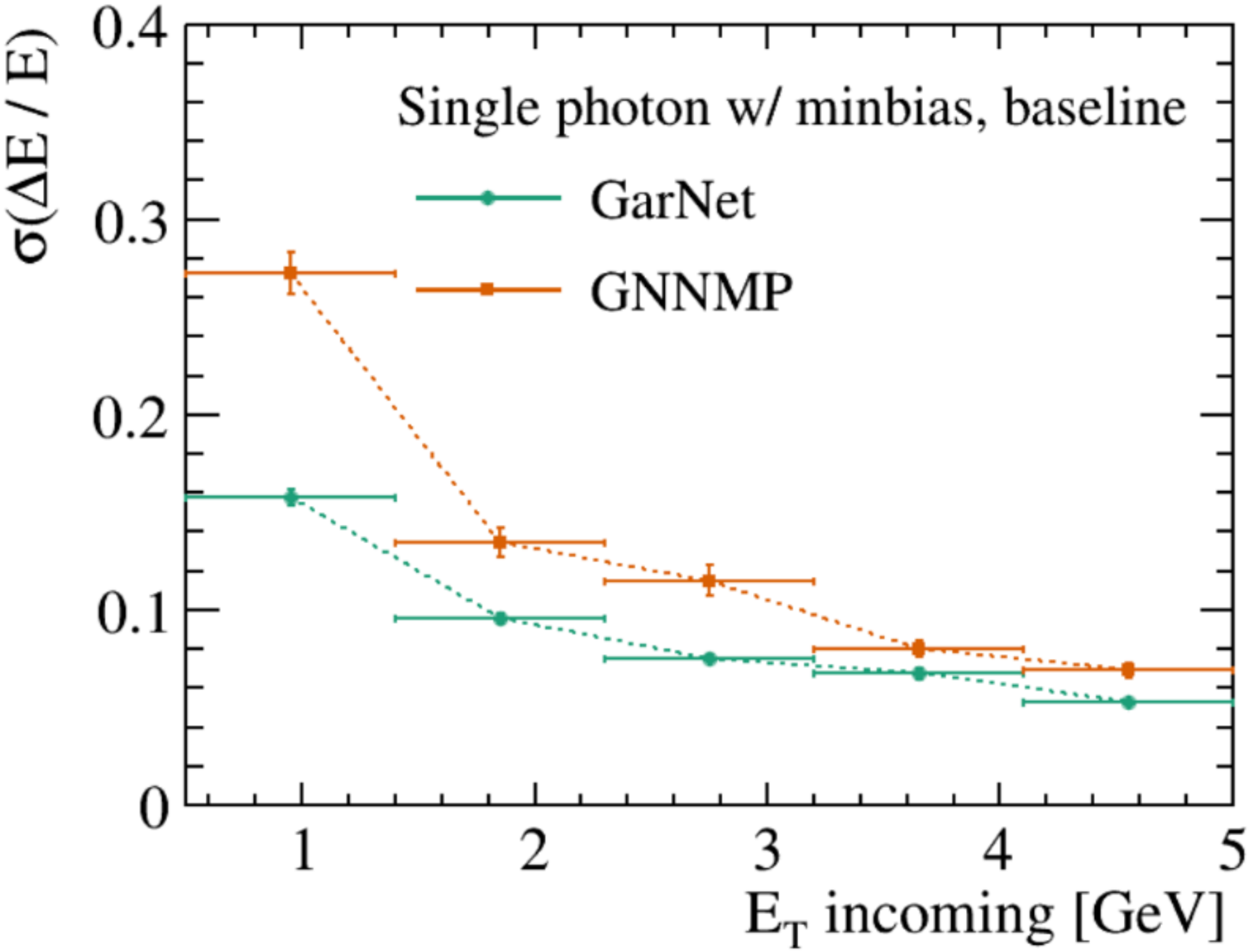


Attention Mechanism

→ **Nodes:** Chaos of fires and spills
→ **Attention weights:** Priority scores
→ **Output**: weighted response strategy $h_i = \Sigma_j \alpha_{ij} \cdot v_j$
→ Prioritizes which chaos to fix first and the **Global Mean Pool** gives a big picture

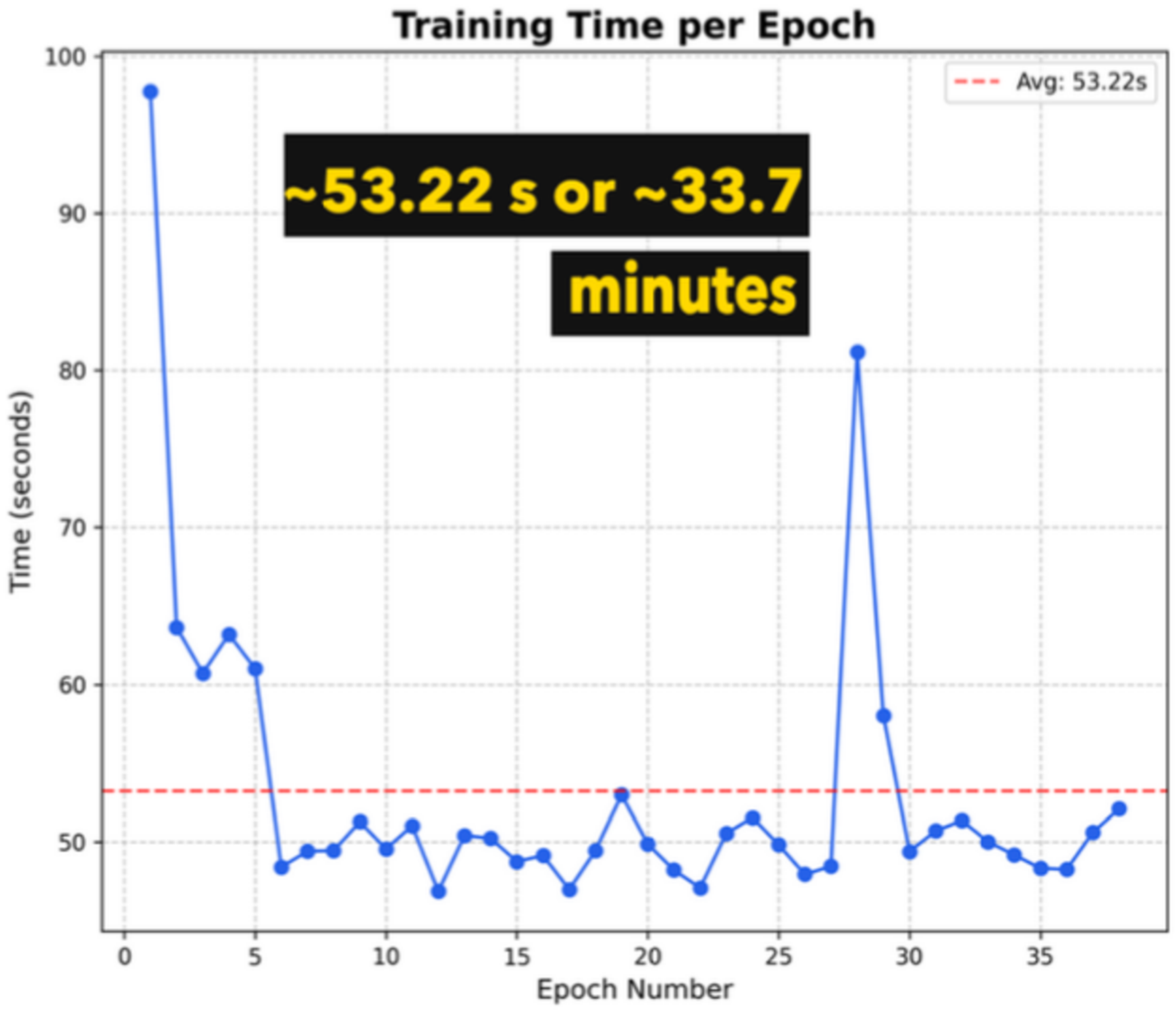
Attention Mechanism

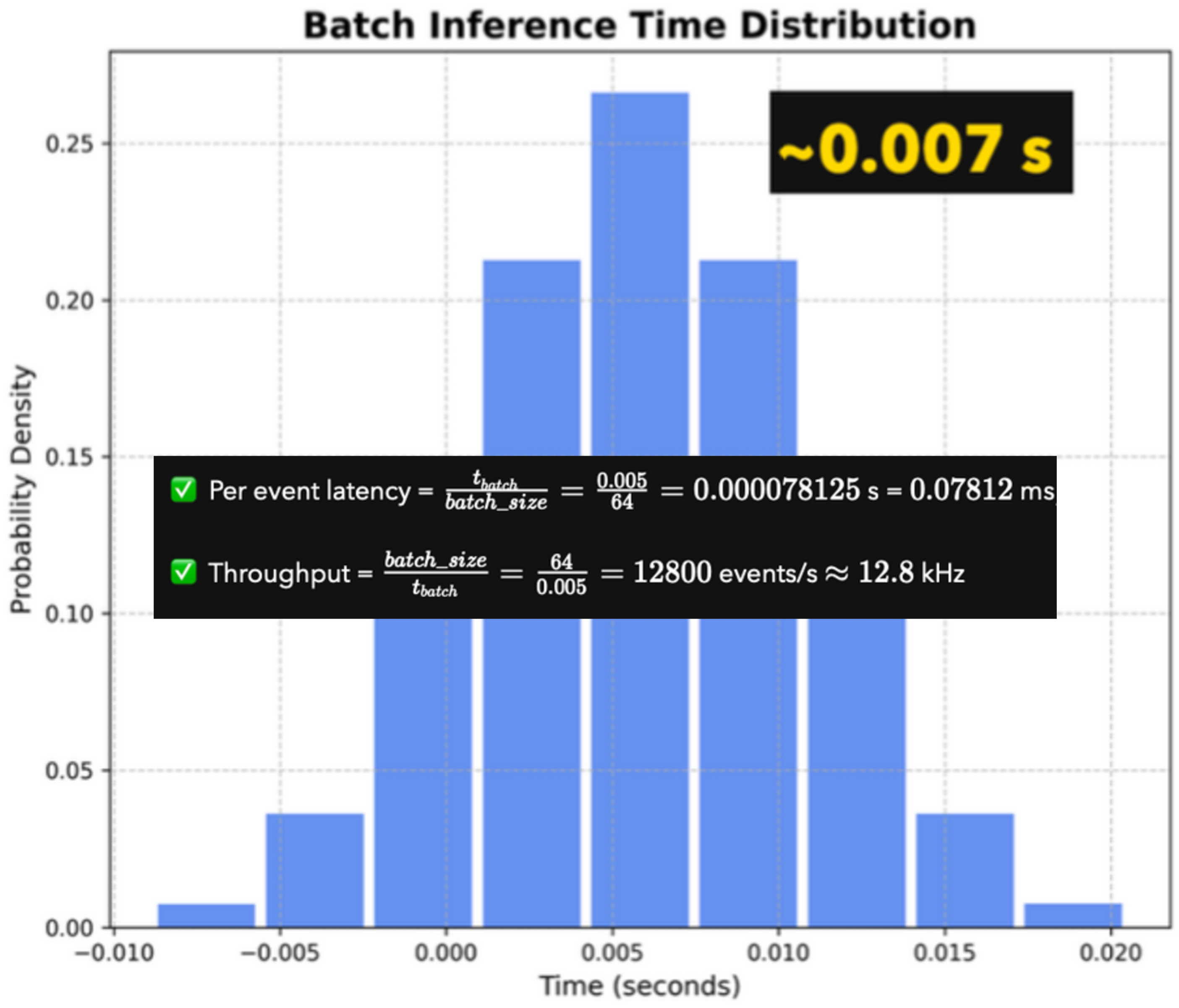
→ ✅ Superior energy resolution despite removing edge features



GarNet Time Complexity

→ ✅ ~8x speedup: Training (from 4 hours down to 30 minutes) and Inference Time (0.05 s to 0.007 s)

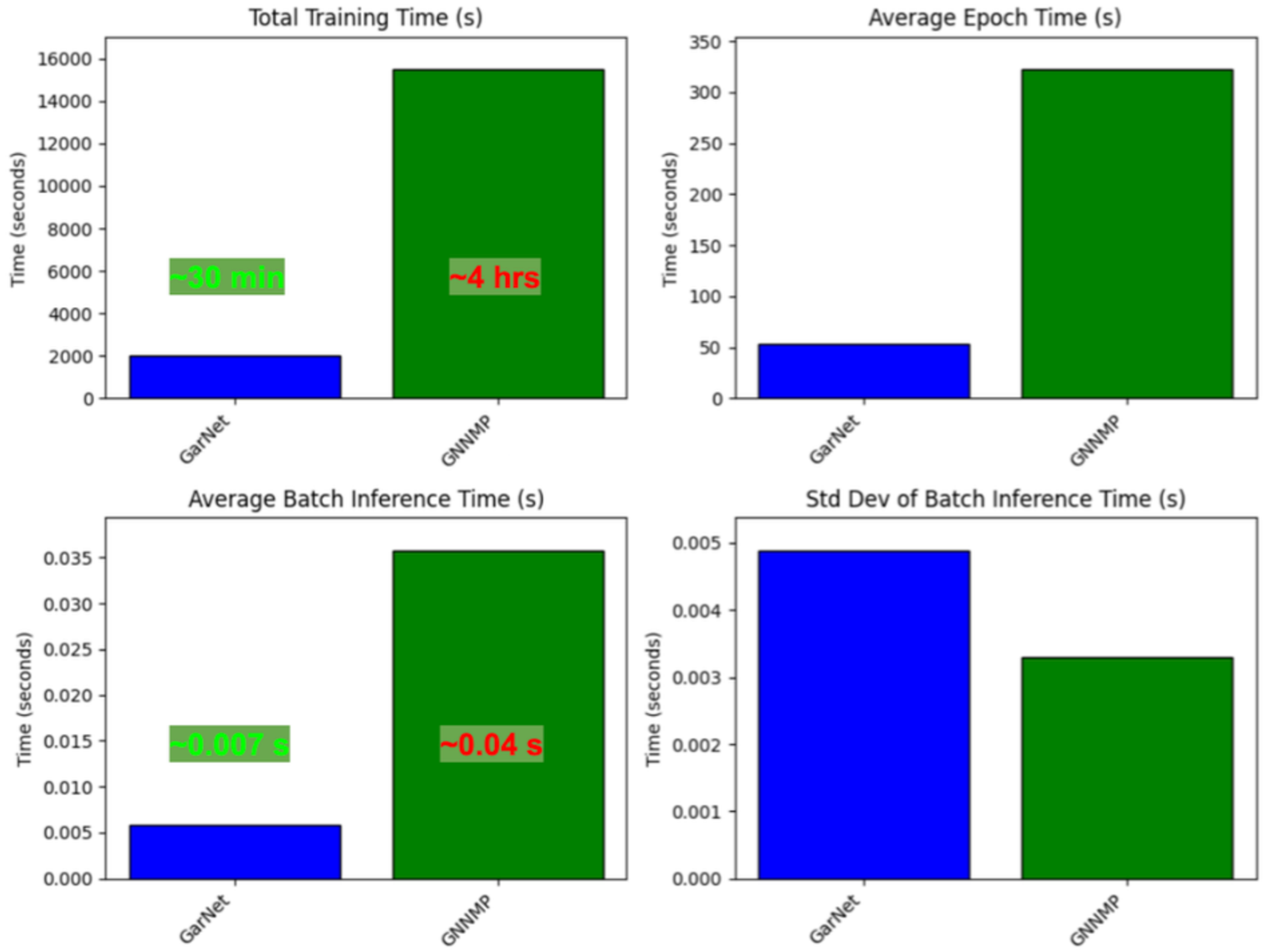




Disclaimer: lxplus CPU times (Xeon Silver 4216), DDP-gloo with 4 processes

Head-to-Head vs GNNMP

→ ✅ Removing explicit edge features → less mathematical operations
→ GNNMP Encoder-processor-Decoder → 2 GarNet Layers and Global Mean Pool



Disclaimer: lxplus CPU times (Xeon Silver 4216), DDP-gloo with 4 processes

Moving Forward

Active Efforts

Promising Initial Results from:

→ Distillation and Quantization providing additional speedup with and superior resolution (c/o Irvin Chacon/ F. Siles of Univ. of Costa Rica)

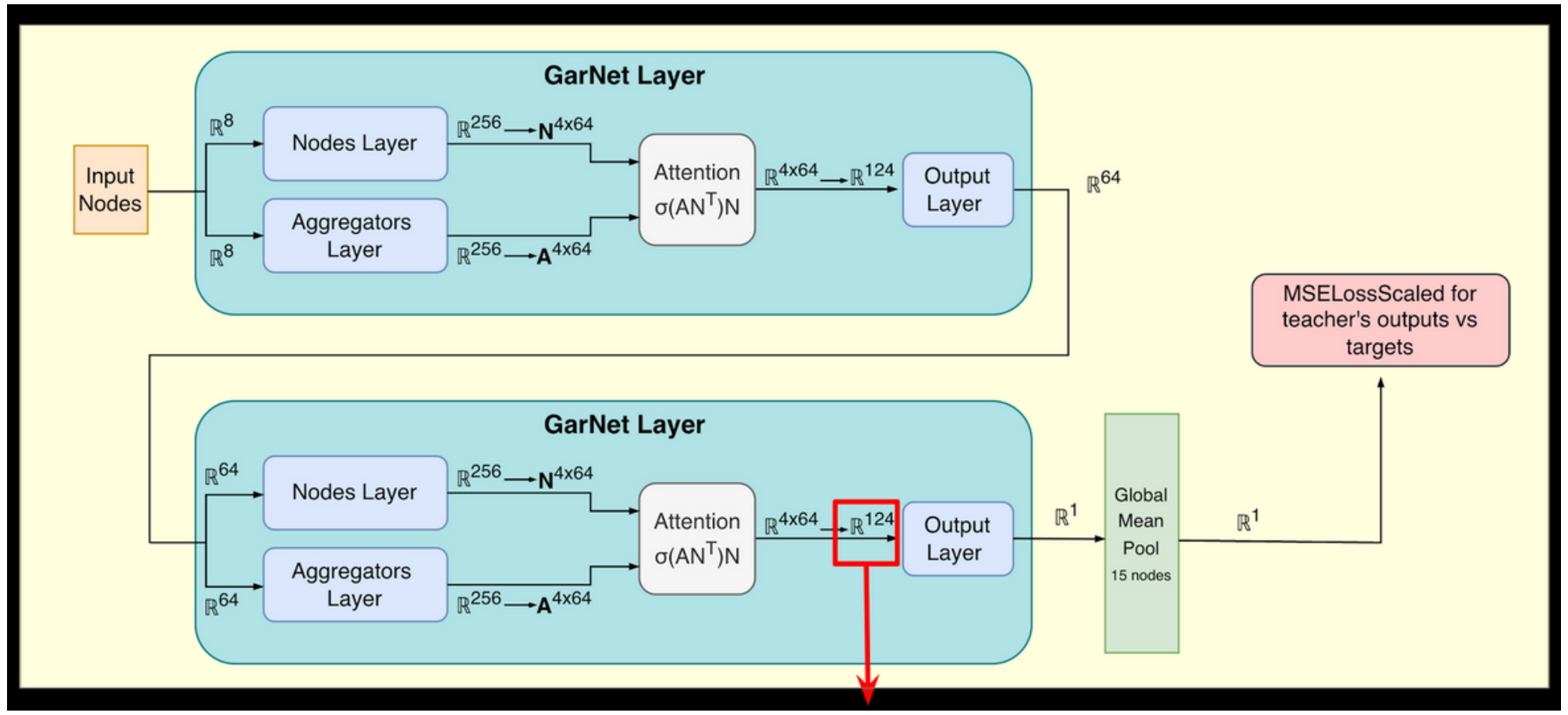
→ Conversion to ONNX Format with even more speedup (c/ Ronald Caravaca/ F. Siles of Univ. of Costa Rica)

→Testing with Allen Framework for Run 5 with the PicoCal (in collaboration with C. Agapopolou, G. Khreich, A. L. Salvia, J.F. Marchand, et al. of the ODISSEE Project)

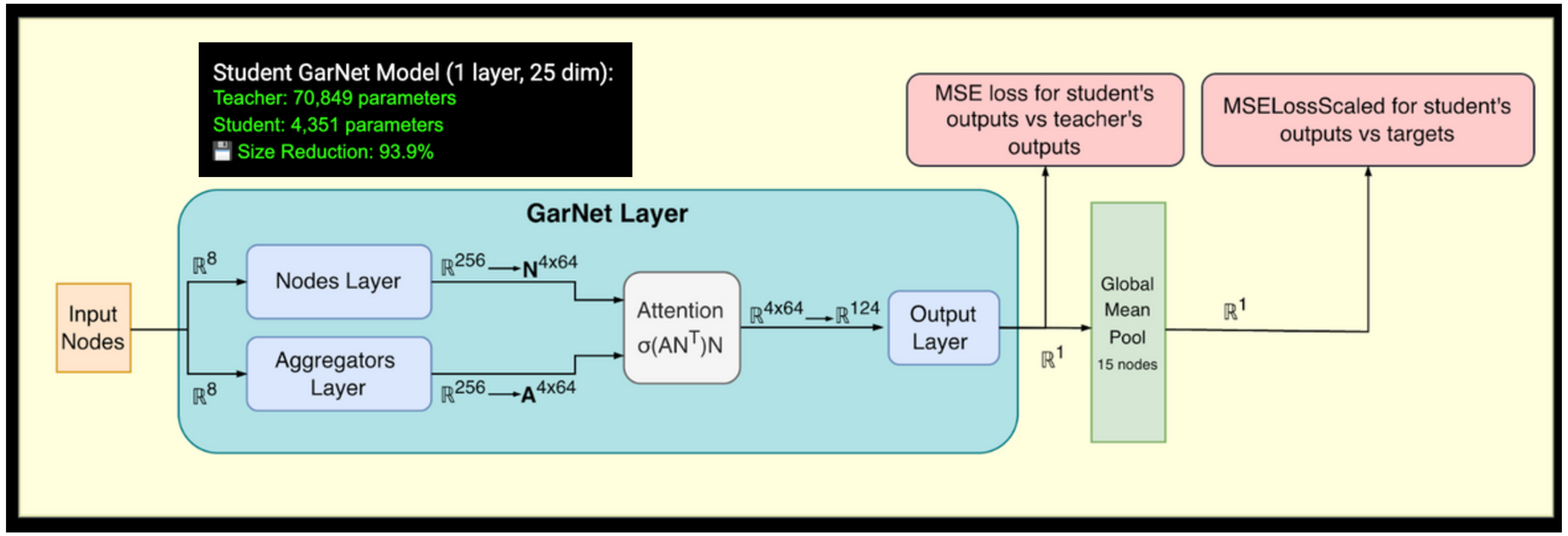
Distillation

Promising Initial Results from:

→ Distillation involves a smaller student network that learns from the output of a teacher network.
→ Loss function is a composite of student learning (λ) on its own and learning from teacher (β)
→ Credits to Irvin Chacon! Stay tuned in another conference for the actual results.
→ We get additional speedup and resolution improvements



Student learns from this output



Total Loss = **β***MSELossScaled + **λ***MSE Loss, **β=0.3, λ = 0.7**

Benchmark: CPU - MacOS M3 chip, using torch.utils.benchmark.Timer with single thread

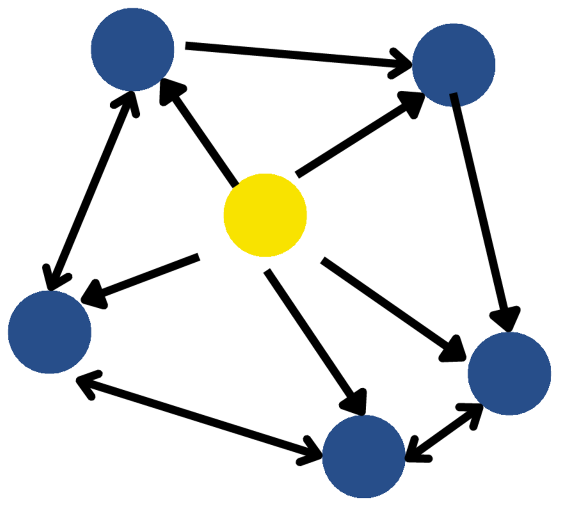
Summary

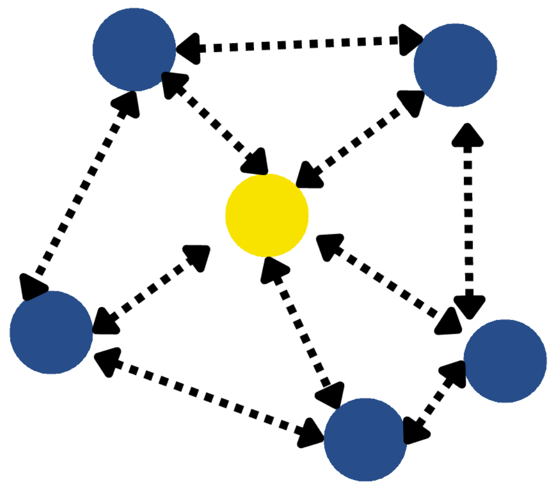
Promising Initial Results from:

→ Clearer path towards HLT1 reconstruction with Graph Neural Networks with Allen Framework with PicoCal

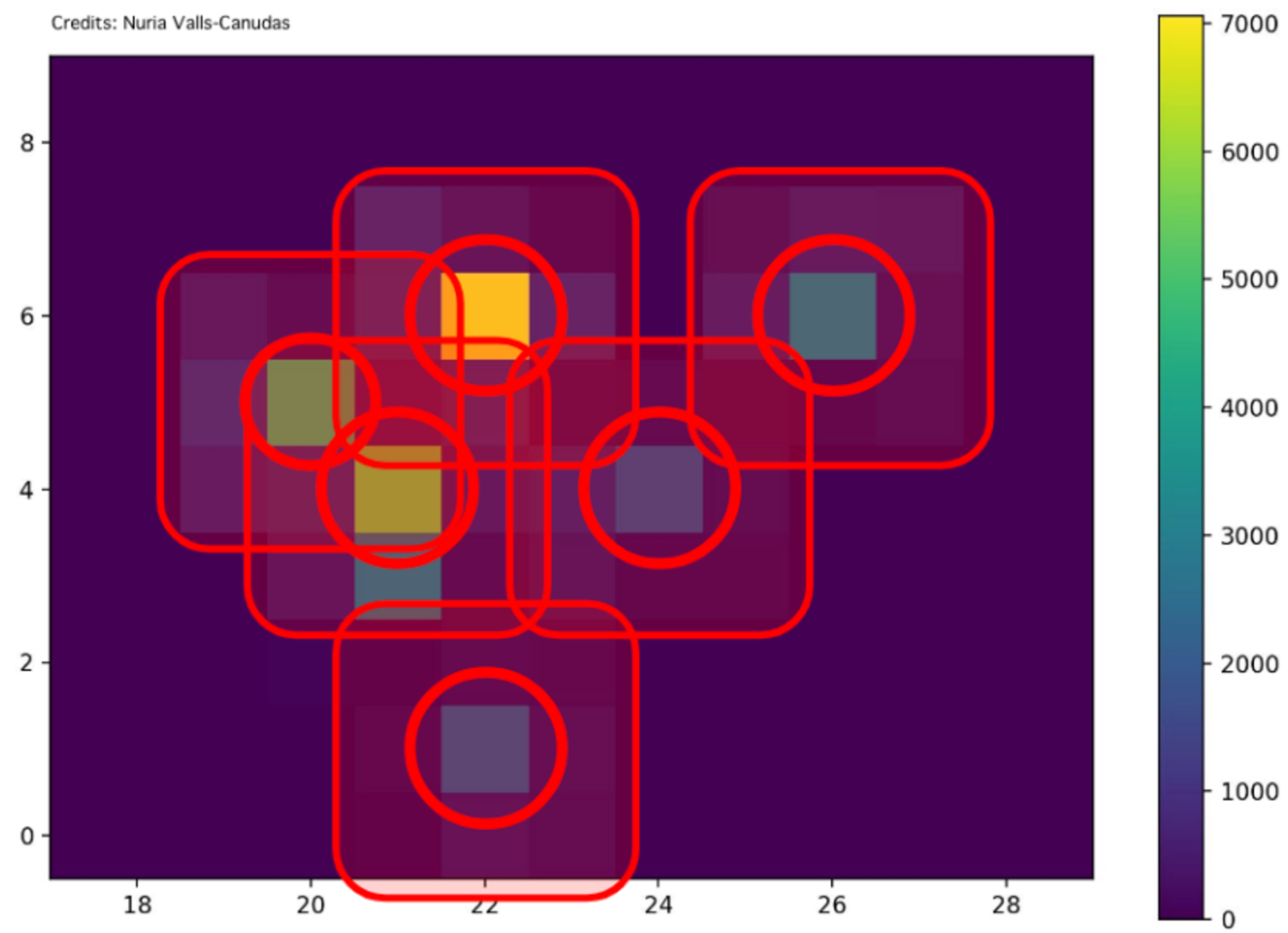
→ We begin with a lightweight version that can be further compressed/ distilled if necessary achieving ~30-100x speedup with our initial results using lxplus CPUs and (stay tuned for details)

→ Update Testing with Allen Framework for Run 5 with the PicoCal and get more accurate throughput/ latency numbers for Runs 4 and 5





THANK YOU!



Credits: Nuria Valls-Canudas
8
6
4
2
0
18
20
22
24
26
28
7000
6000
5000
4000
3000
2000
1000
0

Key Design Principles

Efficient Architecture for PD

```
❯ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  28
  On-line CPU(s) list:   0-27
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
    CPU family:          6
    Model:               85
    Thread(s) per core:  1
    Core(s) per socket:  1
    Socket(s):           28
    Stepping:            7
    BogoMIPS:            4199.76
```

Future Work and Contact Information

Email

uzzie.perez@cern.ch