Karlsruhe Institute of Technology

Institut für Experimentelle Teilchenphysik

# Graph Neural Networks for reconstruction

**JENNIFER 3 Kickoff-Meeting, 28.01.2025**

Torben FERBER (torben.ferber@kit.edu), with input from Isabel HAIDE, Marc NEU, and Lea REUTER (alphabetical order)
Institute of Experimental Particle Physics (ETP)

# Why Graph Neural Networks?

- Challenges in Particle Physics Detectors:
  - **Sparse Data**: Most detector sensors are not read out in an event (because they are below threshold), leading to high sparsity in the data.
  - **Irregular Sensor Layout**: Sensors are often arranged non-uniformly, complicating efficient data aggregation for conventional neural network by requiring embedding or approximation

- Key GNN Concepts:
  - **Message Passing**: Nodes aggregate information from their neighbors through functions (min, max, average, learned custom, …), iterating across layers of the network.
  - Message passing restricts the **local context** that each node considers per layer
  - After several layers of message passing, nodes (or entire graphs) are represented by dense vector embeddings.

# Graph Neural Networks for Belle II reconstruction tasks

| | Drift chamber (+SVD) | Calorimeter |
|---|---|---|
| Offline (CPU, GPU) including HLT | • Unseeded track finding of an unknown number of tracks of unknown origin followed by conventional track fitting<br>• ML task: **Graph segmentation** and **node assignment** | • Seeded, fuzzy clustering (assign energy fraction of each crystal to one or two objects, and background)<br>• ML task: **Node regression** |
| Online (FPGA, CGRA) | • Hit cleanup followed by unseeded track finding and parameter estimation of an unknown number of tracks of unknown origin<br>• ML task: **Node classification**, and **graph segmentation** and **node regression** | • Unseeded cluster finding and parameter estimation (incl. background classifier of an unknown number of clusters)<br>• ML task: **Graph segmentation** and **node regression** |

CGRA: coarse-grained reconfigurable architecture, e.g. AMD Versal AI cores or SiMa.ai MODALIX (configurable on word level)
FPGA: field programmable gate arrays, e.g. AMD UltraScale or UltraScale+ (configurable on bit level)

# Graph Neural Networks for Belle II reconstruction tasks

| | Drift chamber (+SVD) | Calorimeter |
|---|---|---|
| Offline (CPU, GPU) including HLT | • Unseeded track finding of an unknown number of tracks of unknown origin followed by conventional track fitting<br>• ML task: **Graph segmentation** and **node assignment** | • Seeded, fuzzy clustering (assign energy fraction of each crystal to one or two objects, and background)<br>• ML task: **Node regression** |
| Online (FPGA, CGRA) | • Hit cleanup followed by unseeded track finding and parameter estimation of an unknown number of tracks of unknown origin<br>• ML task: **Node classification**, and **graph segmentation** and **node regression** | • Unseeded cluster finding and parameter estimation (incl. background classifier of an unknown number of clusters)<br>• ML task: **Graph segmentation** and **node regression** |

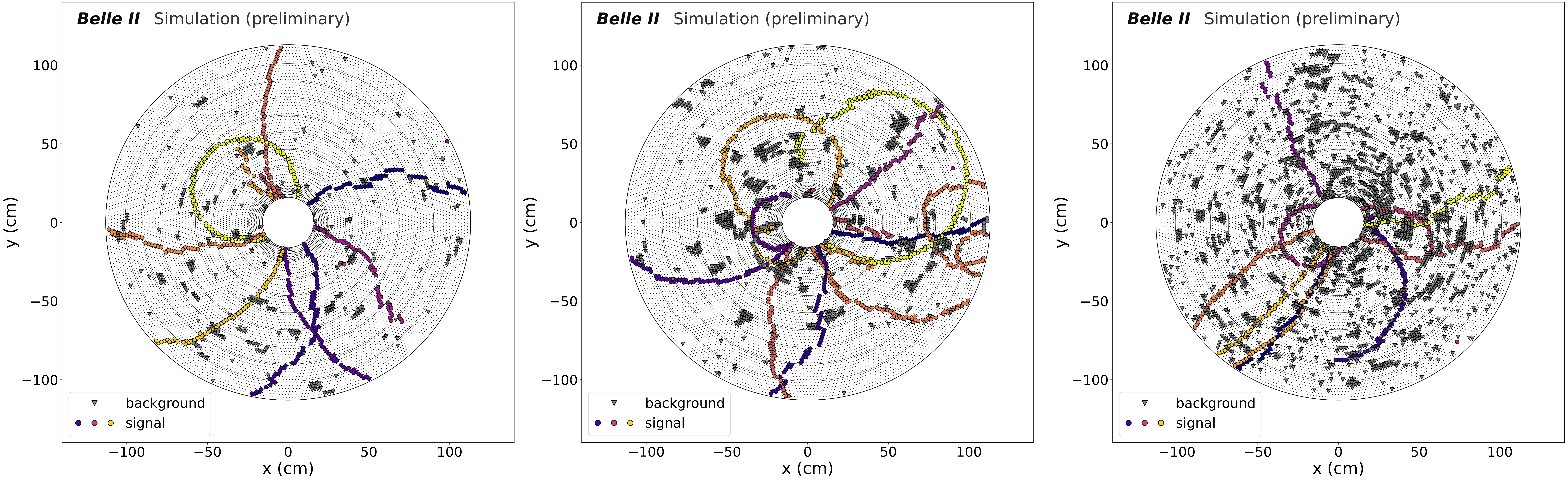
CGRA: coarse-grained reconfigurable architecture, e.g. AMD Versal AI cores or SiMa.ai MODALIX (configurable on word level)
FPGA: field programmable gate arrays, e.g. AMD UltraScale or UltraScale+ (configurable on bit level)

# Offline track finding GNN: Beam backgrounds

Beginning of 2021
360 background CDC hits on average

End of 2022
1280 background CDC hits on average

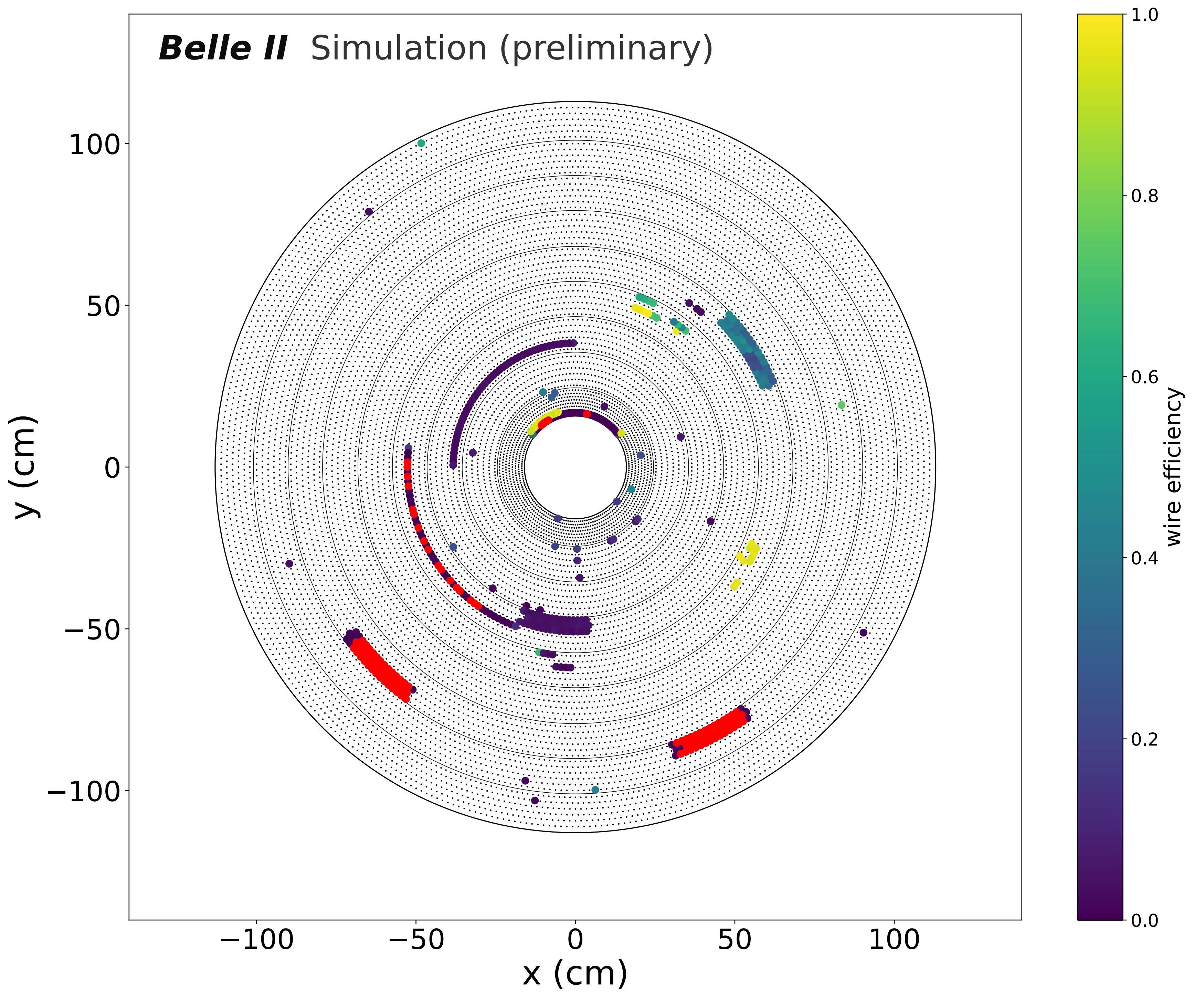
Expected 2030
3000 background CDC hits on average



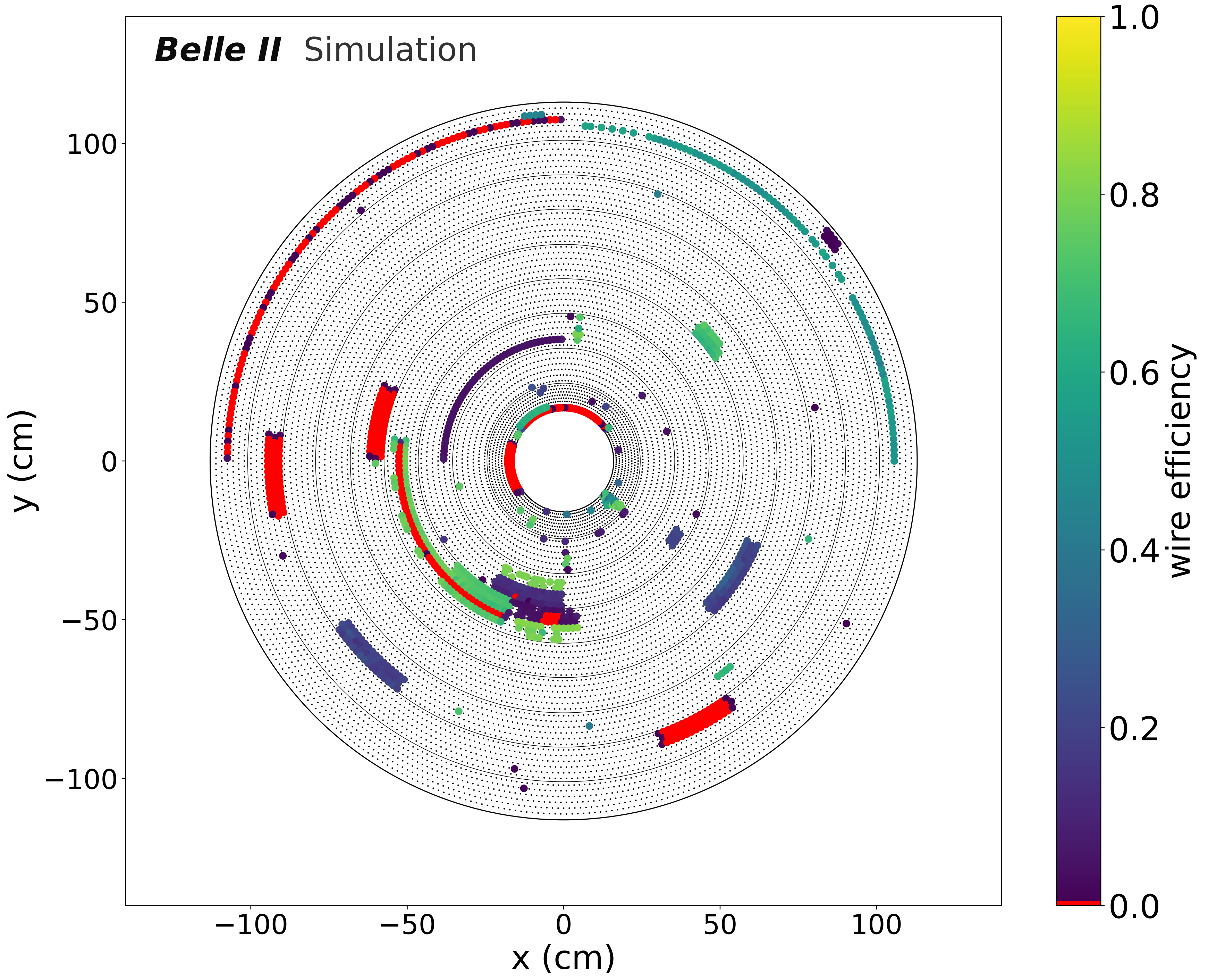

Backgrounds are getting higher, $n^2$-hard problem for tracking-finding

# Offline track finding GNN: CDC performance loss

Beginning of 2021
50 wires off, 368 decreased efficiency
Total of 3% of the CDC wires




End of 2022
168 wires off, 809 decreased efficiency
Total of 7% of the CDC wires



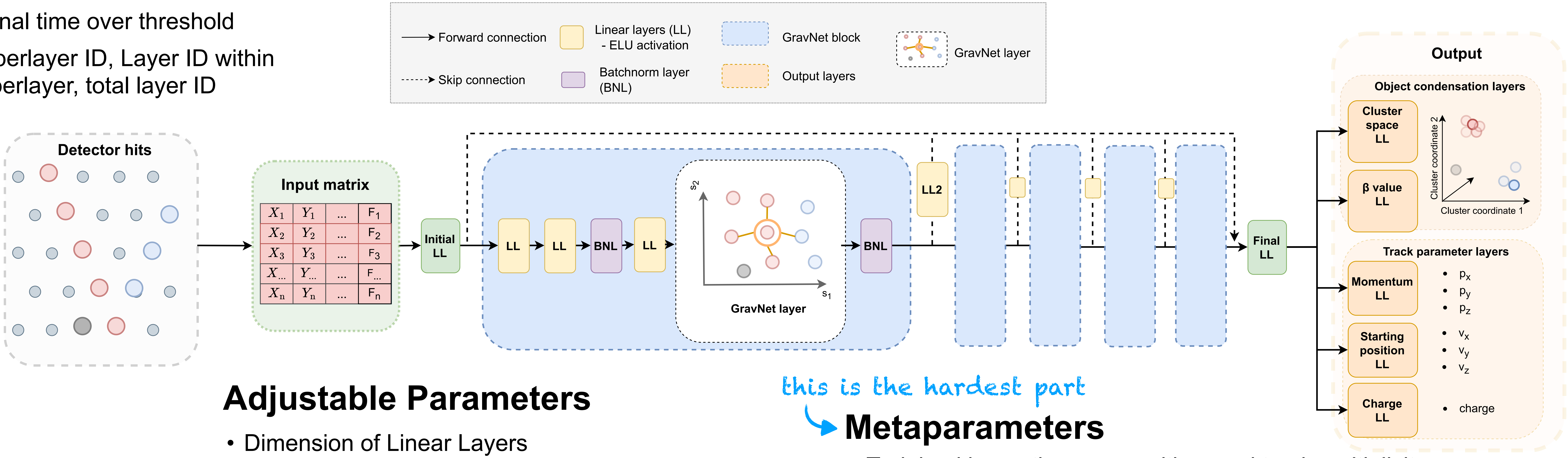
# Offline track finding GNN: GNN architecture

## Input per hit

- X - position
- Y - position
- Signal height
- Signal timing
- Signal time over threshold
- Superlayer ID, Layer ID within superlayer, total layer ID

## Output

- Number of track objects, for each object:
  - Momentum (starting direction and curvature)
  - Charge
  - All hits belonging to the track
    → passed to conventional track fitting algorithm




## Adjustable Parameters

- Dimension of Linear Layers
- Number of GravNet Blocks
- Number of k-nearest neighbours in GravNet
- GravNet space dimensions
- Dimension of Cluster Coordinates
- Number of output layers

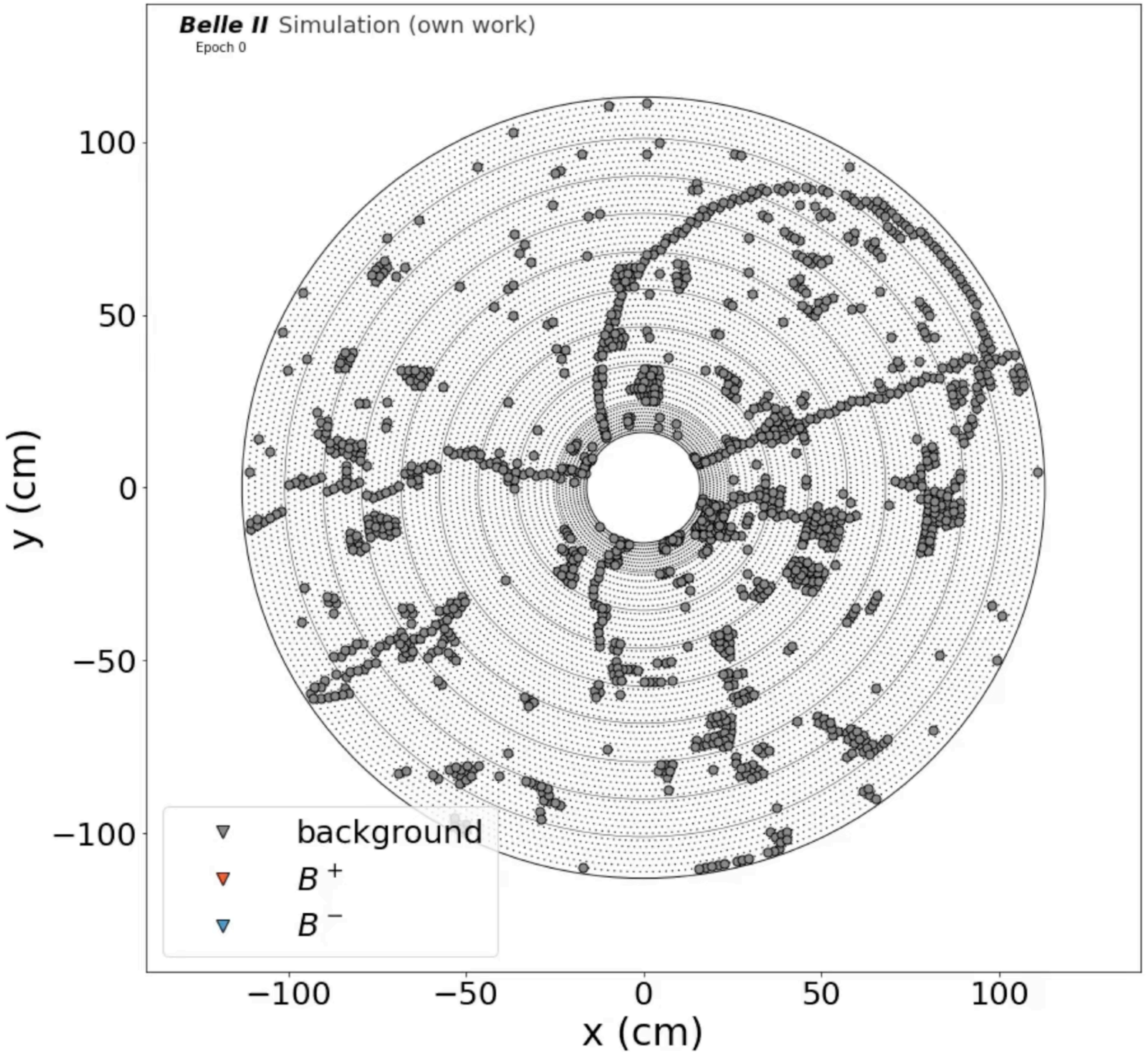
this is the hardest part

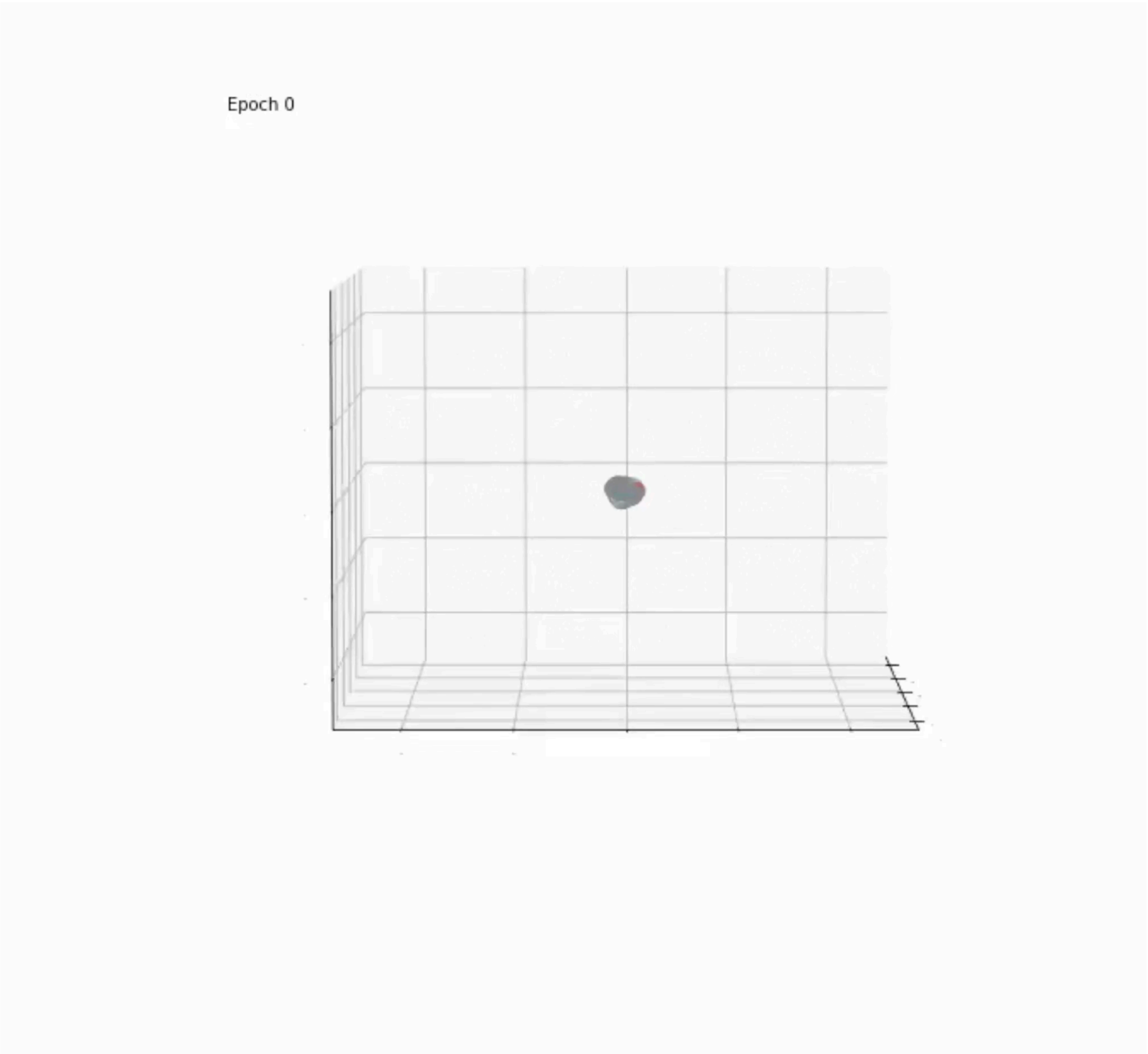
## Metaparameters

- Training kinematics, composition, and track multiplicity coverage
- Background level
- CDC wire inefficiencies
- Working point tuning (efficiency, resolution, fake rates)
- Track fitting algorithm optimization

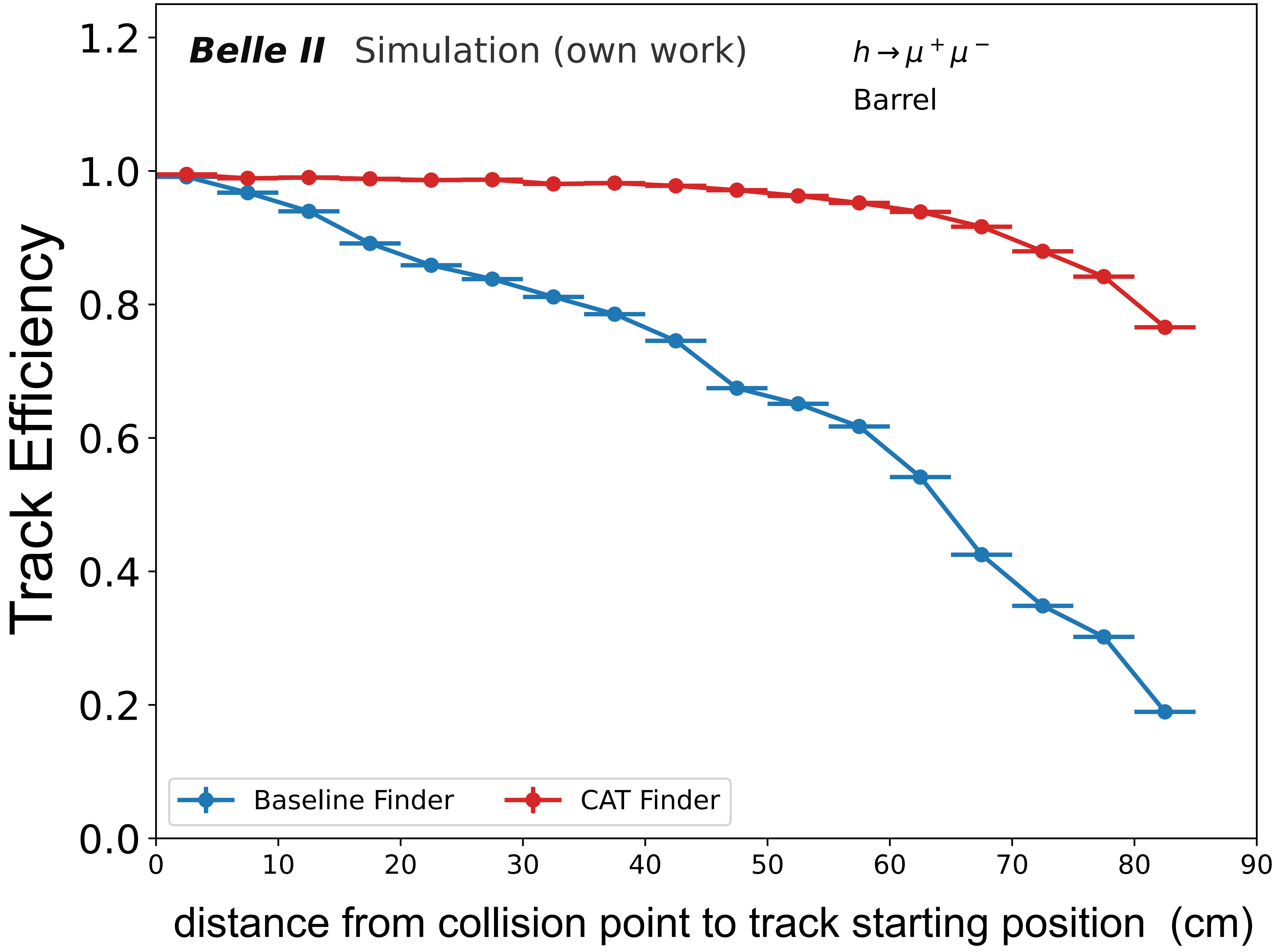

GNN details (unrelated to tracking): arXiv:2002.03605, arXiv:1902.07987

# Offline track finding GNN: GNN inner workings (animation)



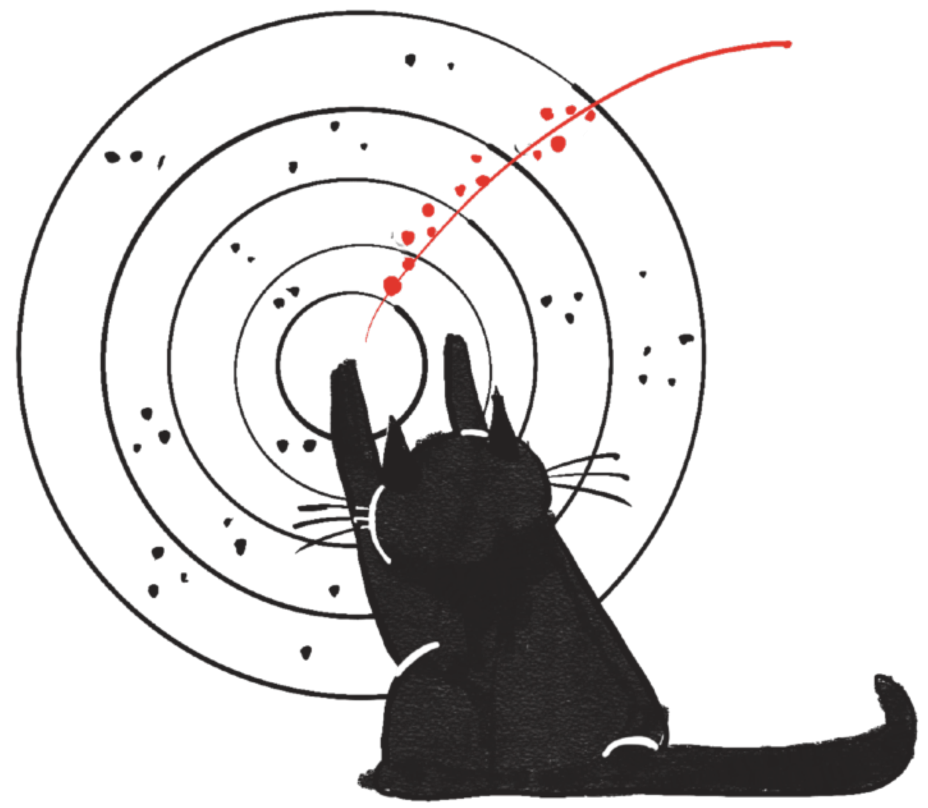



# Offline track finding GNN performance for displaced tracks




| averaged | Efficiency | Purity |
|---|---|---|
| **Baseline** | **0.574$^{+/-0.001}$** | **0.964$^{+/-0.001}$** |
| **CAT Finder** | **0.892$^{+/-0.001}$** | **0.978$^{+/-0.001}$** |

Bertacchi et al., Track Finding at Belle II (arXiv:2003.12466)



"End-to-End Multi-Track Reconstruction using Graph Neural Networks at Belle II" https://arxiv.org/abs/2411.13596

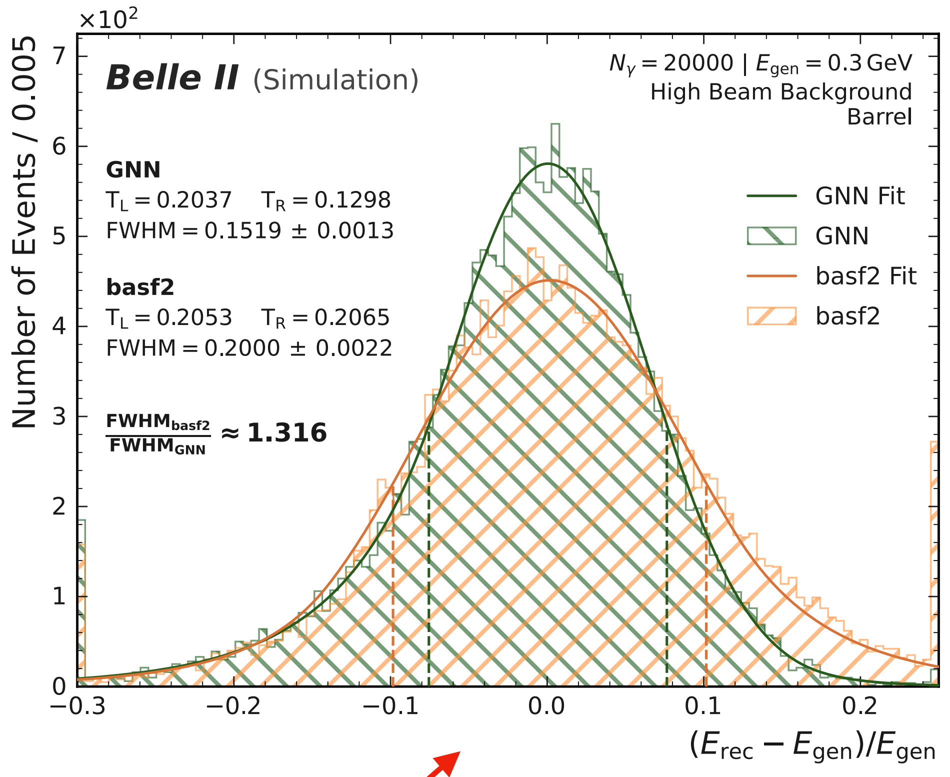
# Graph Neural Networks for Belle II reconstruction tasks

| | Drift chamber (+SVD) | Calorimeter |
|---|---|---|
| Offline (CPU, GPU) including HLT | • Unseeded track finding of an unknown number of tracks of unknown origin followed by conventional track fitting<br>• ML task: **Graph segmentation** and **node assignment** | • Seeded, fuzzy clustering (assign energy fraction of each crystal to one or two objects, and background)<br>• ML task: **Node regression** |
| Online (FPGA, CGRA) | • Hit cleanup followed by unseeded track finding and parameter estimation of an unknown number of tracks of unknown origin<br>• ML task: **Node classification**, and **graph segmentation** and **node regression** | • Unseeded cluster finding and parameter estimation (incl. background classifier of an unknown number of clusters)<br>• ML task: **Graph segmentation** and **node regression** |

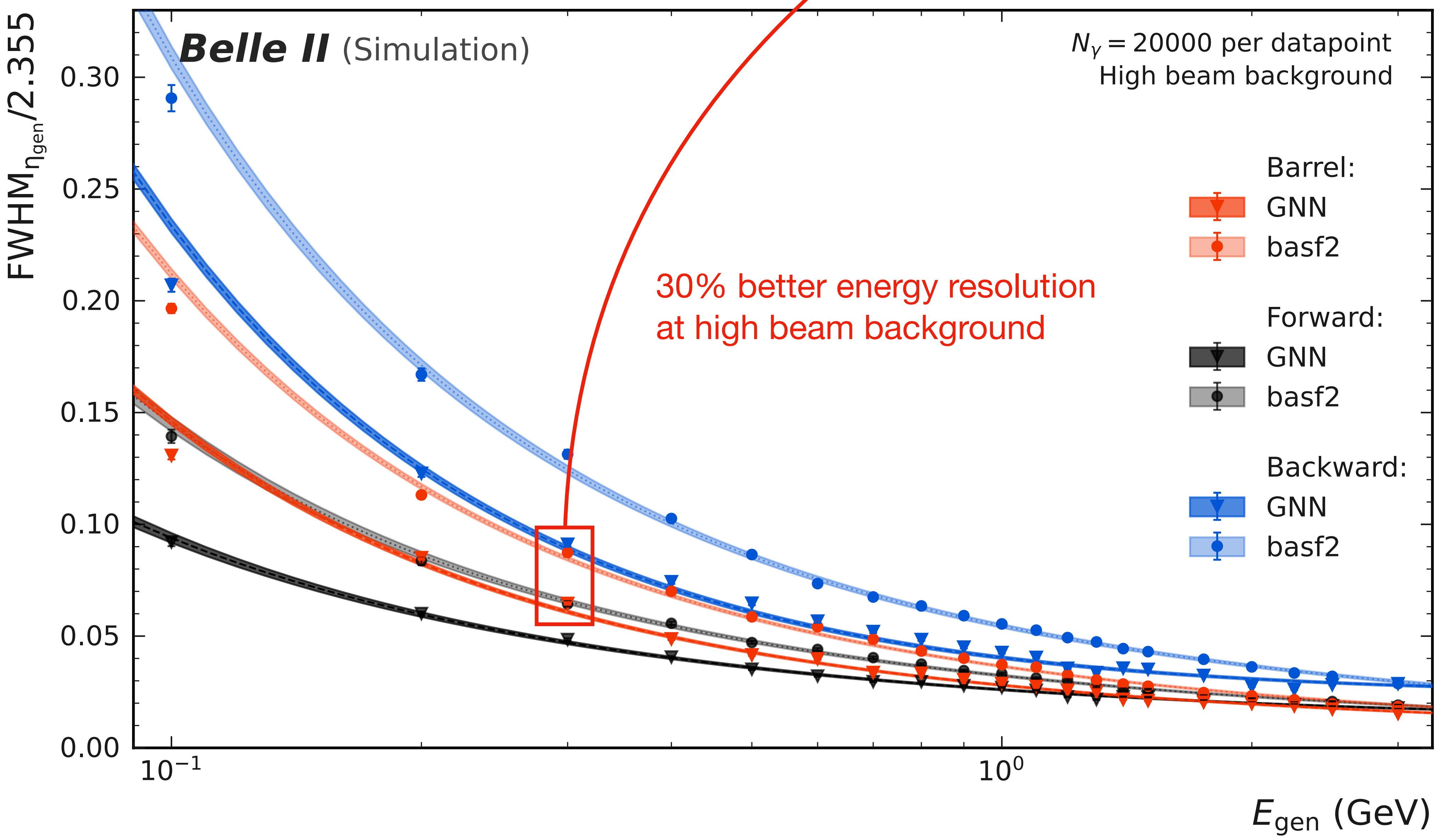
CGRA: coarse-grained reconfigurable architecture, e.g. AMD Versal AI cores or SiMa.ai MODALIX (configurable on word level)
FPGA: field programmable gate arrays, e.g. AMD UltraScale or UltraScale+ (configurable on bit level)

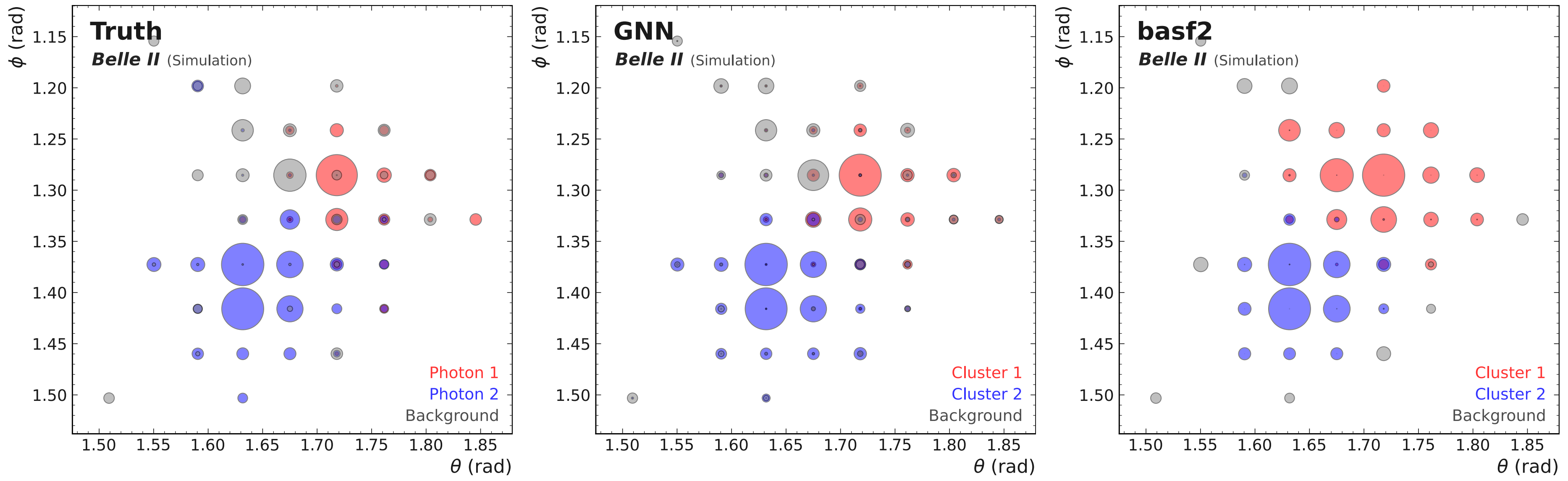
# Offline clustering in the calorimeter

- Non-ML seed finder (local maximum and energy threshold)
- 9×9 crystals as nodes to a GNN with regression targets to predict the energy faction belonging to two (seed or background) or three (seed 1, seed 2, or background) classes
  - GNN with dynamic graph-building to account for geometry and crystal shapes especially in the endcaps
- Significant improvement for low energy photon energy resolution with (very) high beam backgrounds




New GNN algorithm
current algorithm





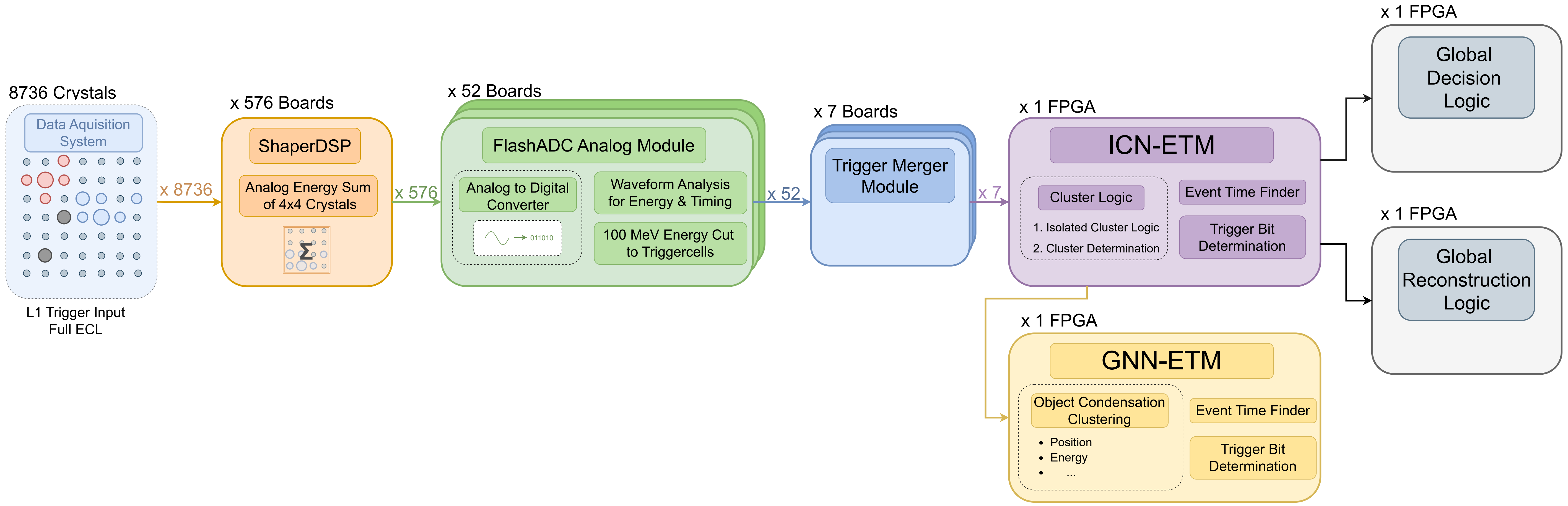

"Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks" https://arxiv.org/abs/2411.13596

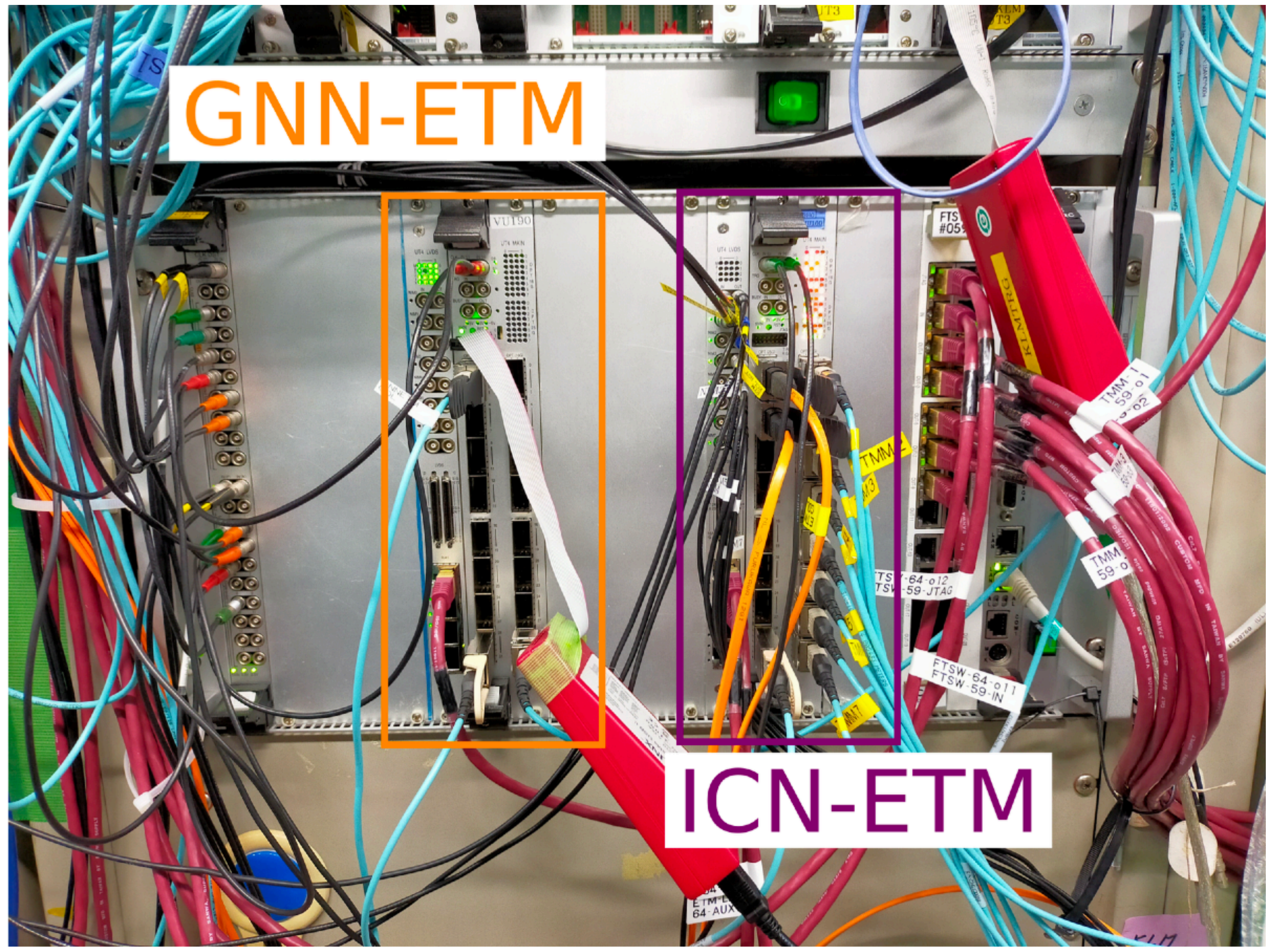
# Graph Neural Networks for Belle II reconstruction tasks

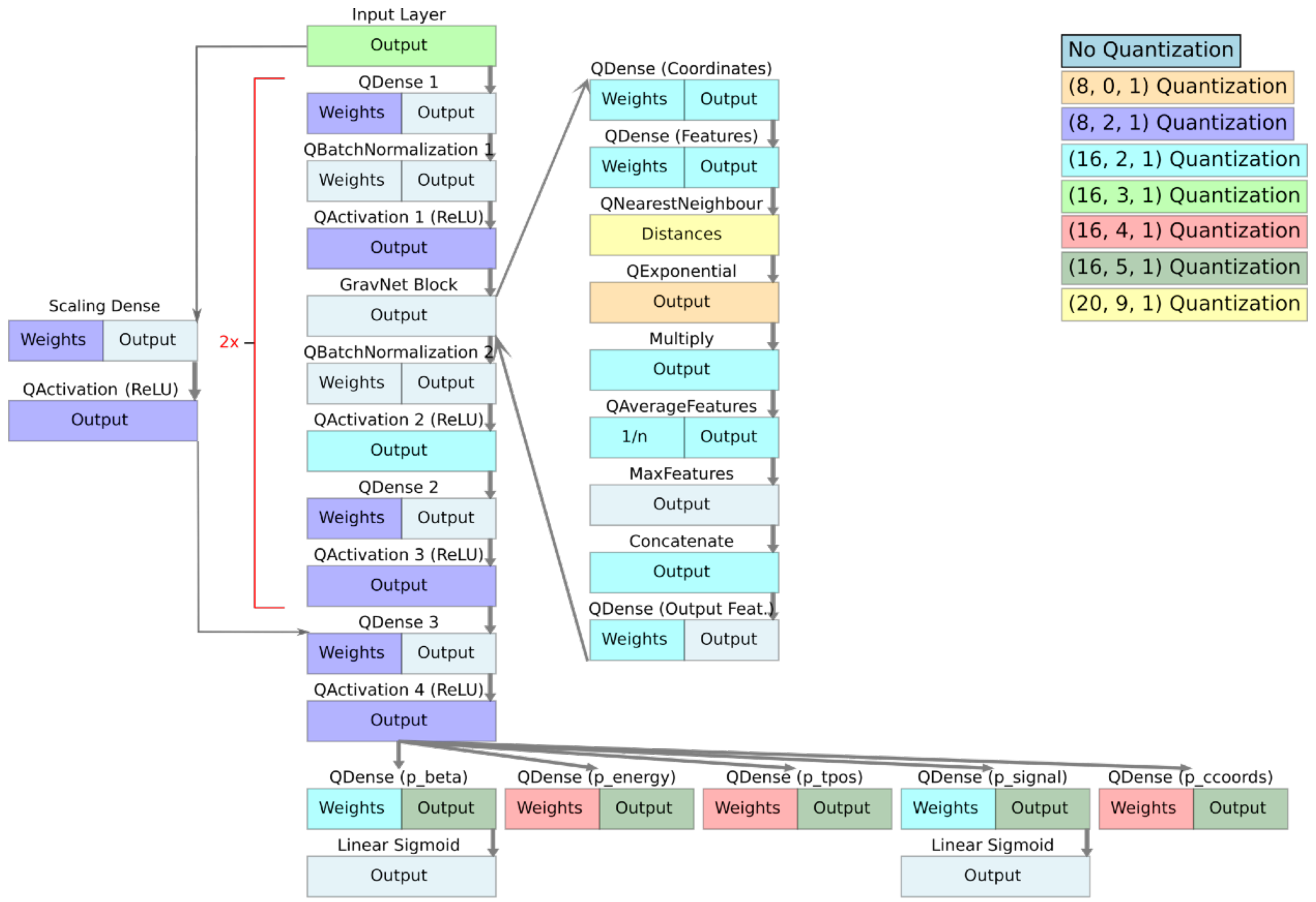
| | Drift chamber (+SVD) | Calorimeter |
|---|---|---|
| Offline (CPU, GPU) including HLT | • Unseeded track finding of an unknown number of tracks of unknown origin followed by conventional track fitting<br>• ML task: **Graph segmentation** and **node assignment** | • Seeded, fuzzy clustering (assign energy fraction of each crystal to one or two objects, and background)<br>• ML task: **Node regression** |
| Online (FPGA, CGRA) | • Hit cleanup followed by unseeded track finding and parameter estimation of an unknown number of tracks of unknown origin<br>• ML task: **Node classification**, and **graph segmentation** and **node regression** | • Unseeded cluster finding and parameter estimation (incl. background classifier of an unknown number of clusters)<br>• ML task: **Graph segmentation** and **node regression** |

CGRA: coarse-grained reconfigurable architecture, e.g. AMD Versal AI cores or SiMa.ai MODALIX (configurable on word level)
FPGA: field programmable gate arrays, e.g. AMD UltraScale or UltraScale+ (configurable on bit level)

# Online clustering in the calorimeter



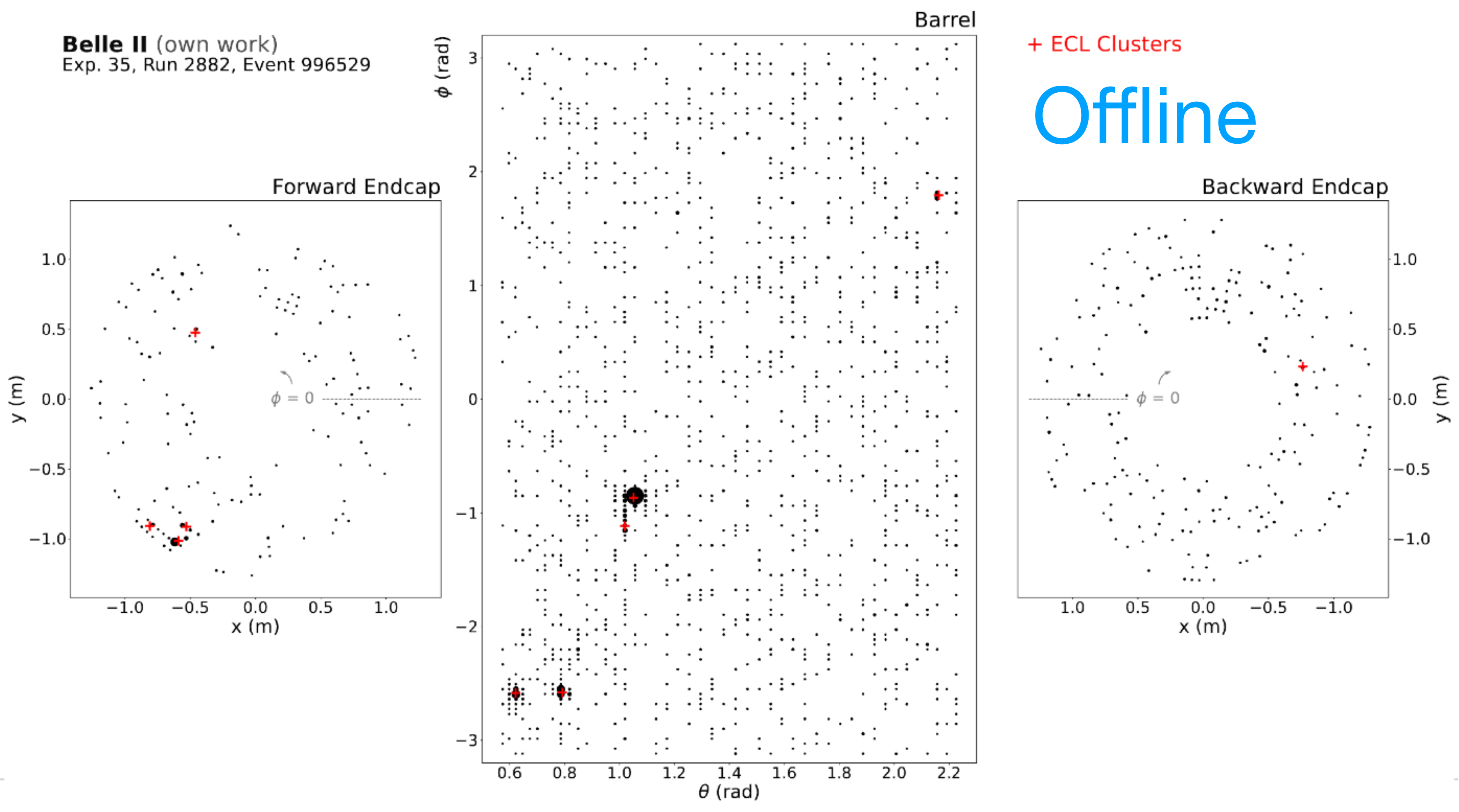



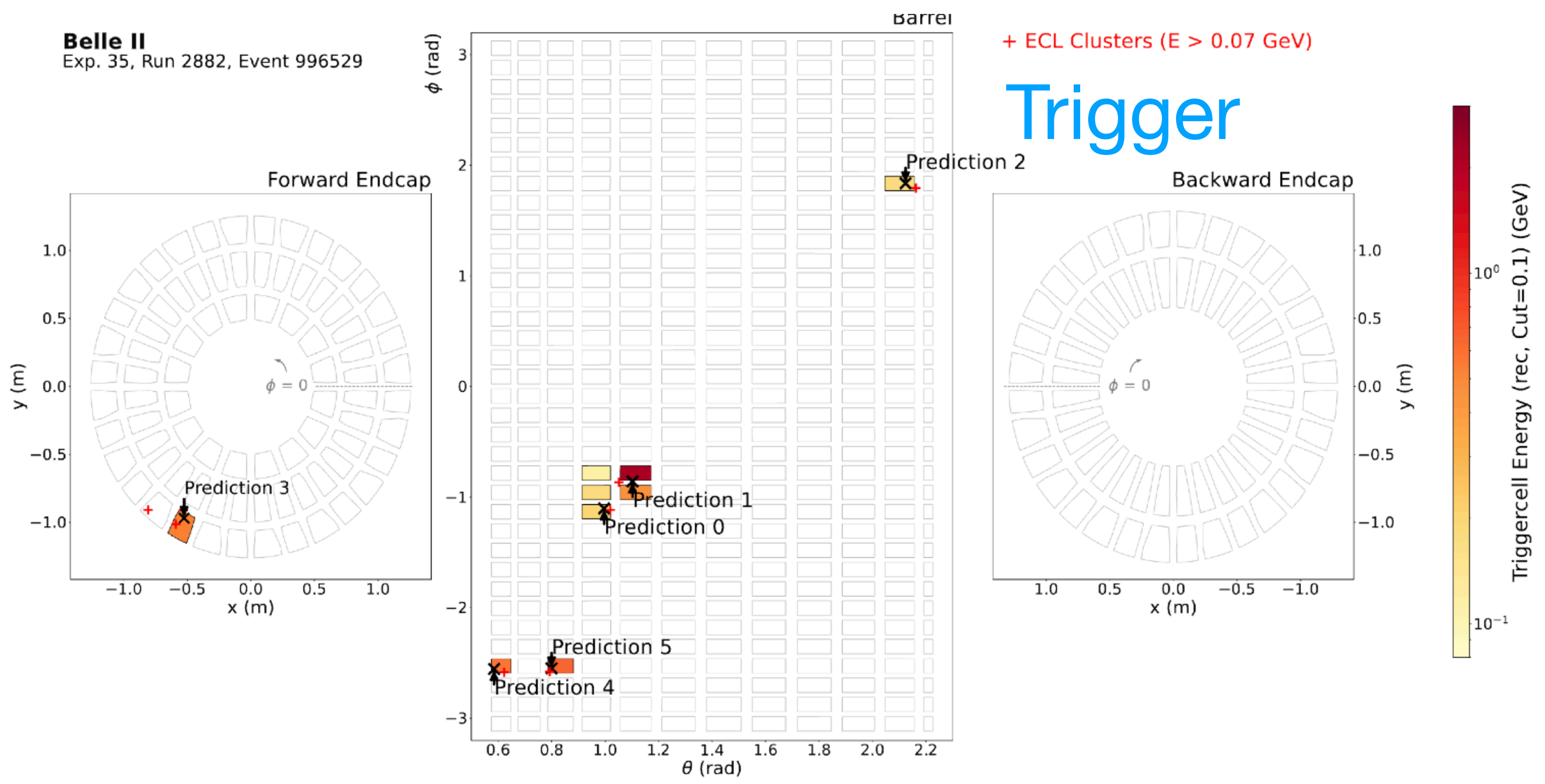




- Similar algorithm as for GNN tracking (slide 7) but with much smaller networks and without node assignment to graphs
- Implemented on AMD UltraScale with 3µs latency (requirement for trigger is ~1.5µs latency)
- Input: up to 32 4x4 cell inputs (energy, time, position)
- Dynamic graph-building with GravNet layers and Object Condensation loss function
- Output: energy, position, background classifier

# Online clustering in the calorimeter

- Included in Belle II physics data taking on 25./26.12.2024
- Fully implemented on FPGA, but operated with dummy weight-files
- Post-processing with identical input TCs using functionally identical Python and C-simulation (right) show expected performance with improved spatial separation and better energy resolution at very high energies.



# Summary and plans

- GNN based trackfinding (CATFinder) is maturing quickly and we aim to make it available in the official Belle II software in the coming months
  - CATFinder is very robust against wire efficiency losses and masked readout boards
  - Next: SVD standalone (much easier than CDC) and SVD+CDC combined tracking
- GNN based clustering in the calorimeter outperforms conventional clustering for photons.
  - Next: Hadron clustering and seed classifiers to optimise missing energy observables
- First real-time GNN trigger operated for the first time five weeks ago.
  - Next: Improve latency (quantization and pruning optimization) and re-optimize for low energies
  - Next: Implement real-time CDC hit cleanup (not shown today)

"Real-time Graph Building on FPGAs for Machine Learning Trigger Applications in Particle Physics" https://arxiv.org/abs/2307.07289