# High-Throughput GNN-Based Track Reconstruction on GPUs at LHCb

arXiv.2407.12119

International Conference on High Energy Physics 2024
Prague, Friday, July 19, 2024

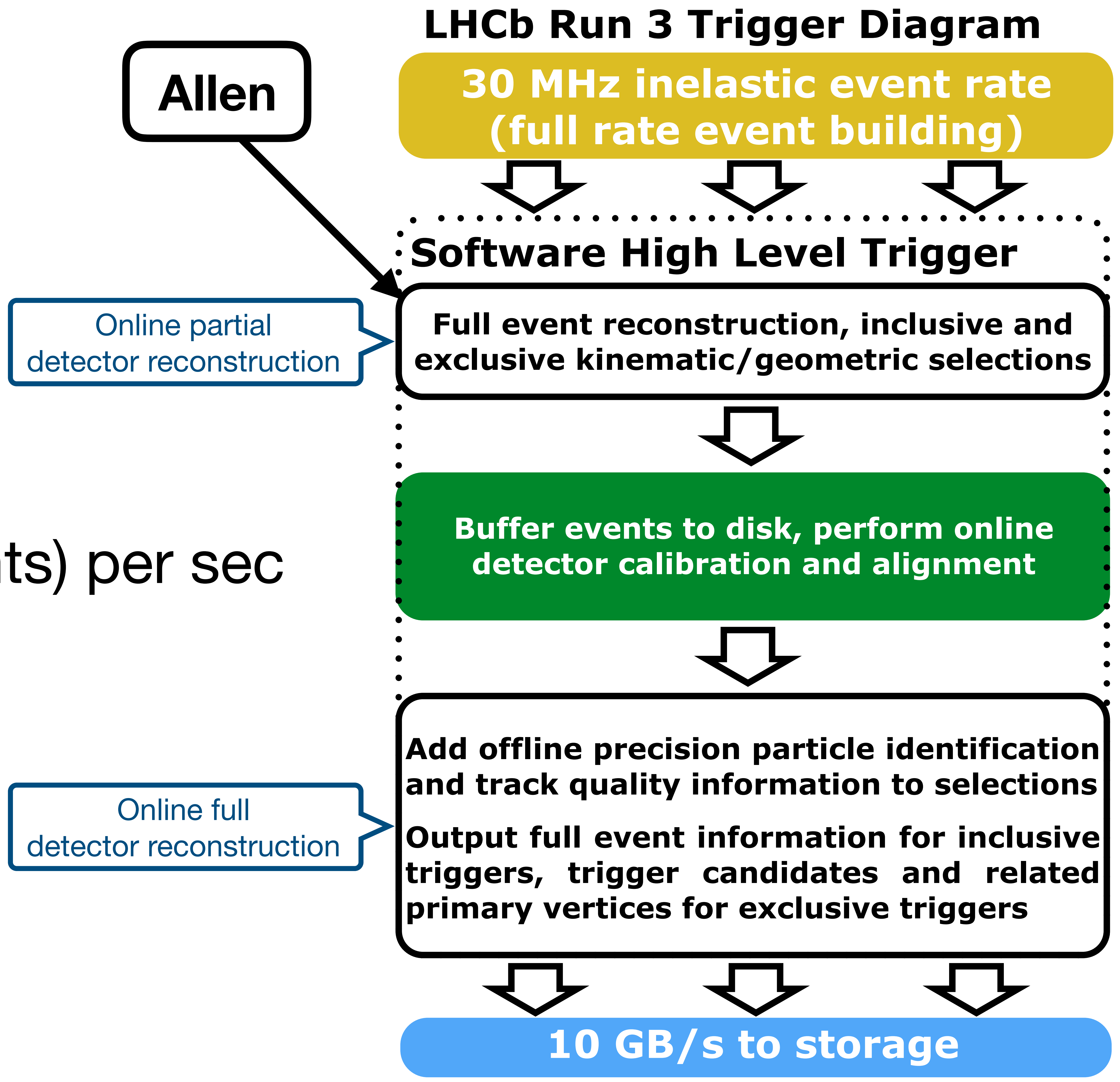
**Fotis I. Giasemis**, Anthony Correia, Nabil Garroum, Vava Gligorov, Bertrand Granado

*On behalf of the LHCb Real-Time Analysis Project*

# LHCb Trigger

## Software trigger of LHCb

- Software high level trigger: 2 levels
- Allen: level 1 of the LHCb trigger
- Filters **30 million** bunch crossings (events) per sec
- Entirely on **GPUs**
- **Track reconstruction**
- Topological triggering on events

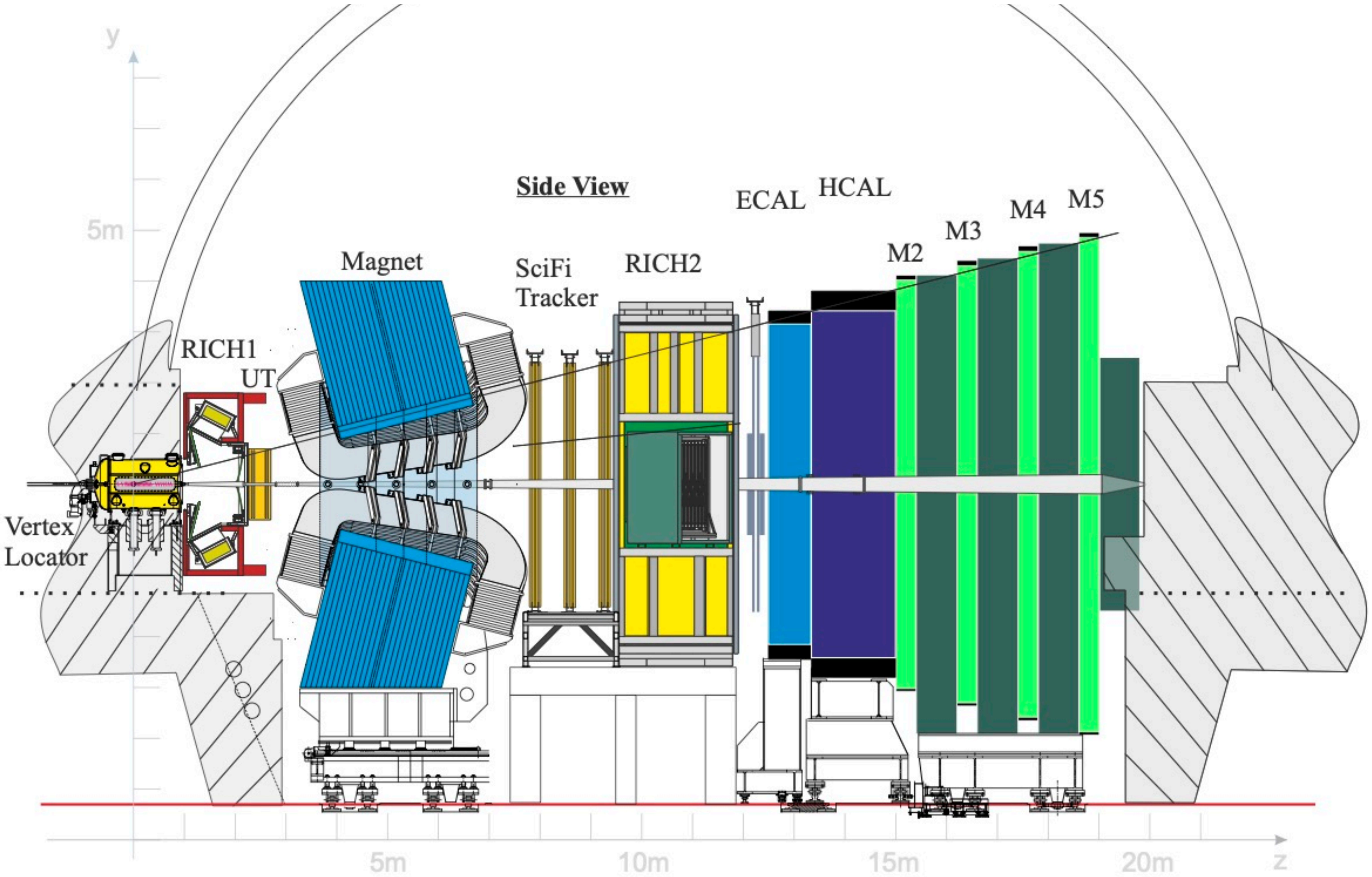



[LHCB-FIGURE-2020-016]

# LHCb Detector

## Focus: Vertex Locator (VELO)

- Momentum resolution: Δp/p ~ 0.5-1%
- B-meson decay time resolution: ~45 fs
- Impact parameter resolution: (15 +29/pT[GeV] ) μm

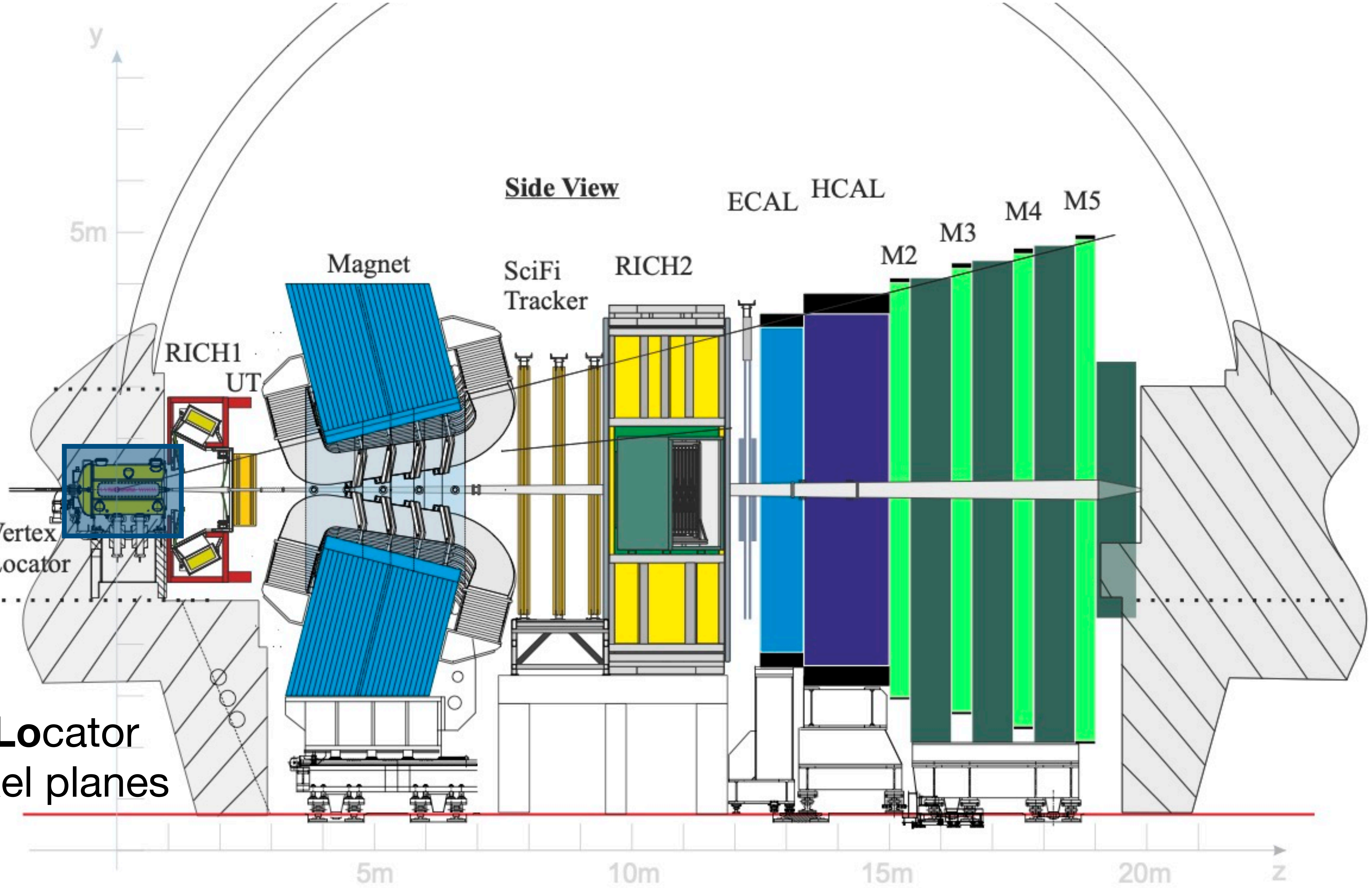



[LHCB-DP-2022-002]

# LHCb Detector

## Focus: Vertex Locator (VELO)

- Momentum resolution: Δp/p ~ 0.5-1%
- B-meson decay time resolution: ~45 fs
- Impact parameter resolution: (15 +29/pT[GeV] ) μm



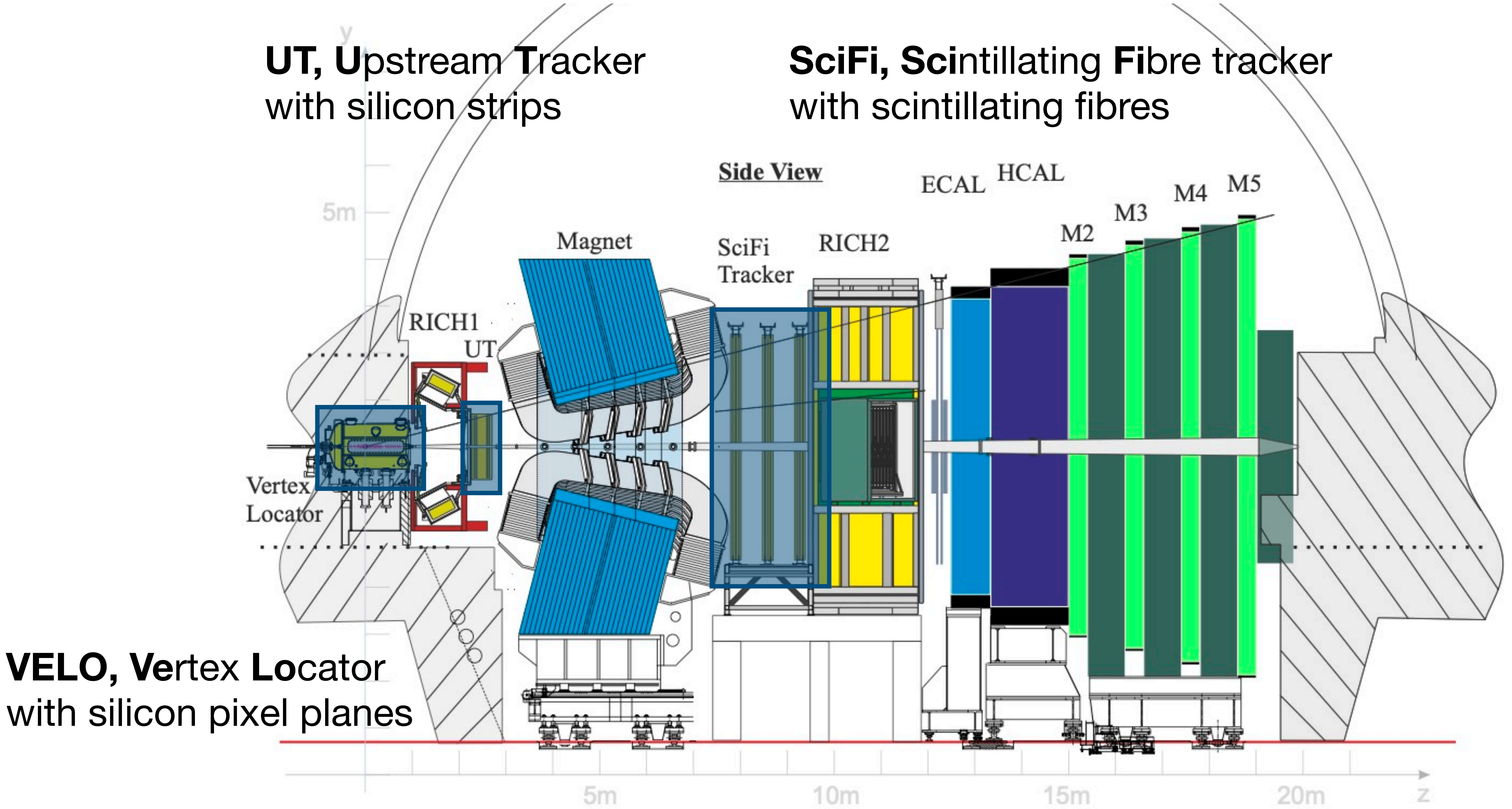

**VELO, Ve**rtex **Lo**cator
with silicon pixel planes

[LHCB-DP-2022-002]

# LHCb Detector

## Focus: Vertex Locator (VELO)

- Momentum resolution: $\Delta p/p \sim 0.5\text{-}1\%$
- B-meson decay time resolution: ~45 fs
- Impact parameter resolution: (15 +29/pT[GeV] ) μm

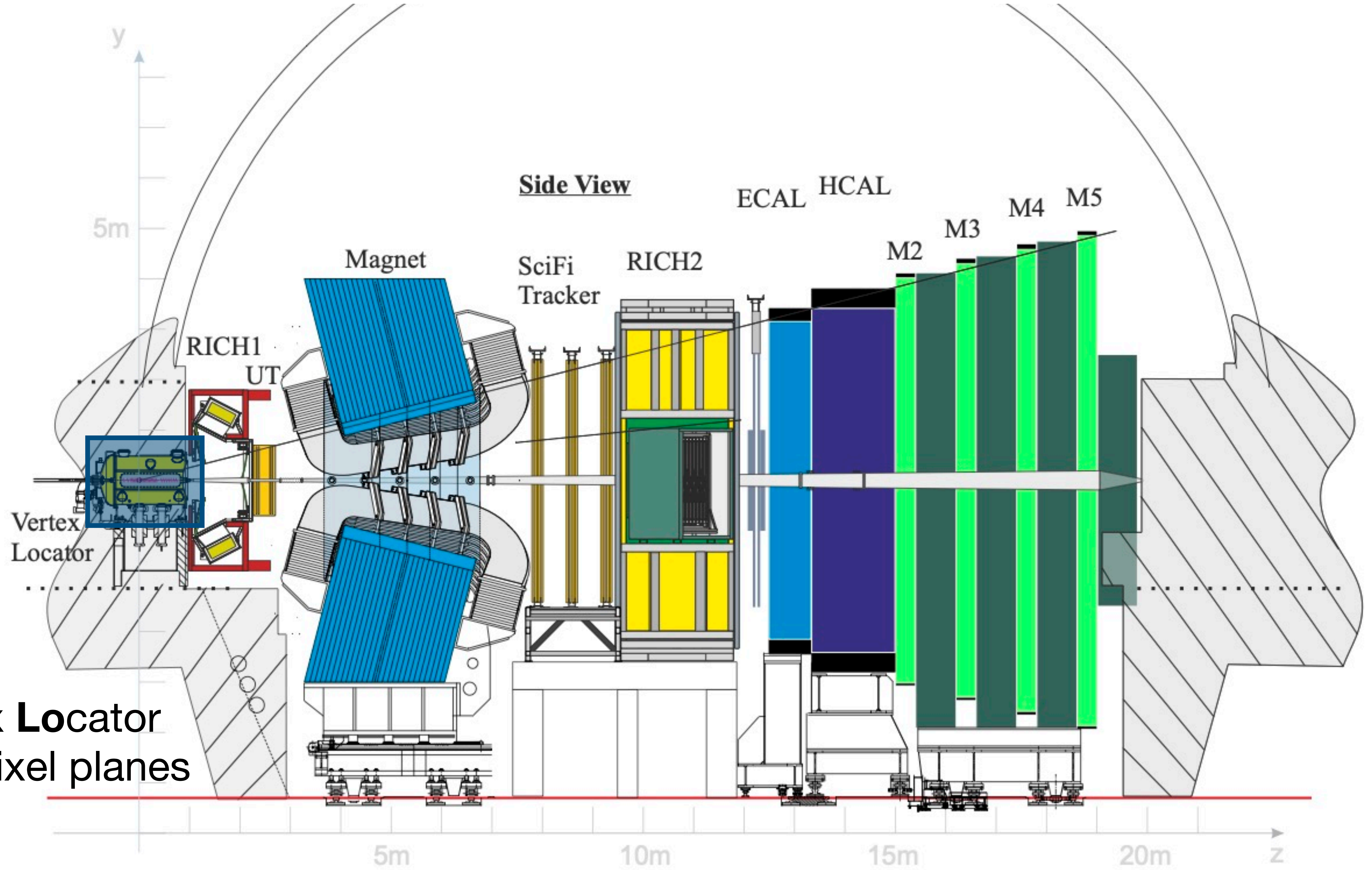



[LHCB-DP-2022-002]

# LHCb Detector

## Focus: Vertex Locator (VELO)

- Momentum resolution: $\Delta p/p \sim 0.5\text{-}1\%$
- B-meson decay time resolution: ~45 fs
- Impact parameter resolution: (15 +29/pT[GeV] ) μm



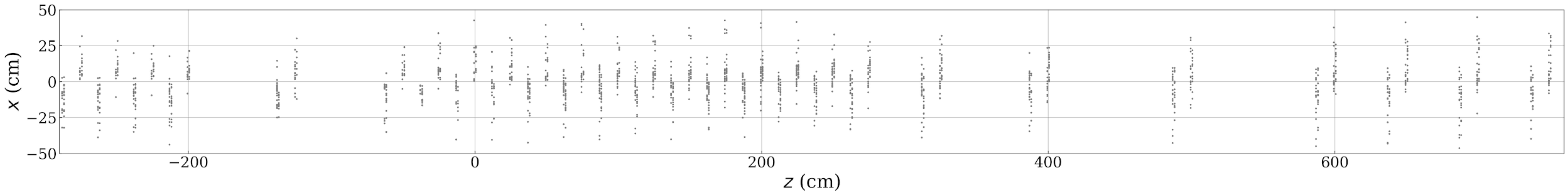

**VELO, Ver**tex **Lo**cator
with silicon pixel planes

[LHCB-DP-2022-002]

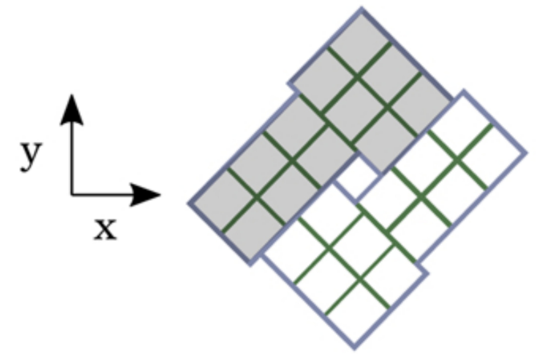
# Track Finding

## Finding tracks from the hits in the detector

“track reconstruction”, “tracking”

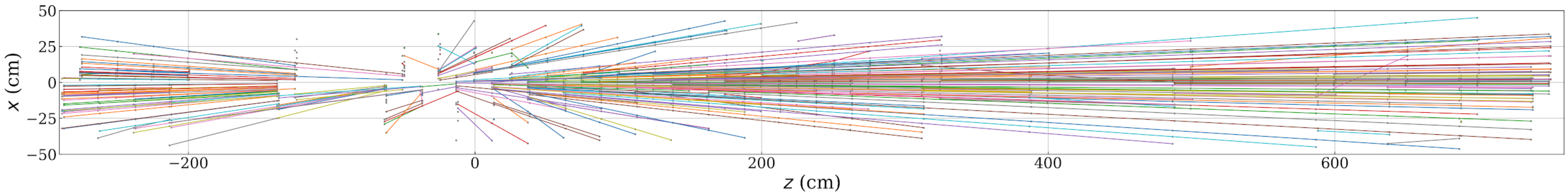



26 pairs of sensors




**Track finding**

**No magnetic field**



# ETX4VELO

## Tracking in the VELO with ETX4VELO

- Question: Will ML allow a more **efficient** use of computing resources?
- Expected increase in luminosity, next generation of detectors
- Inference time **close to linear** on # hits [DOI:10.1140/epjc/s10052-021-09675-8]
- vs classical **worse than quadratic** [DOI:10.48550/arXiv.2012.01563]

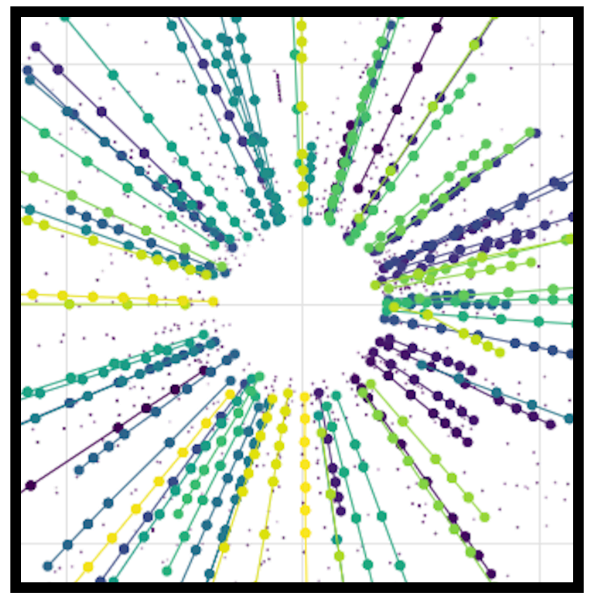
- Starting point: Exa.TrkX collaboration, talk@CHEP2021, PyTorch

# ETX4VELO

## Tracking in the VELO with ETX4VELO

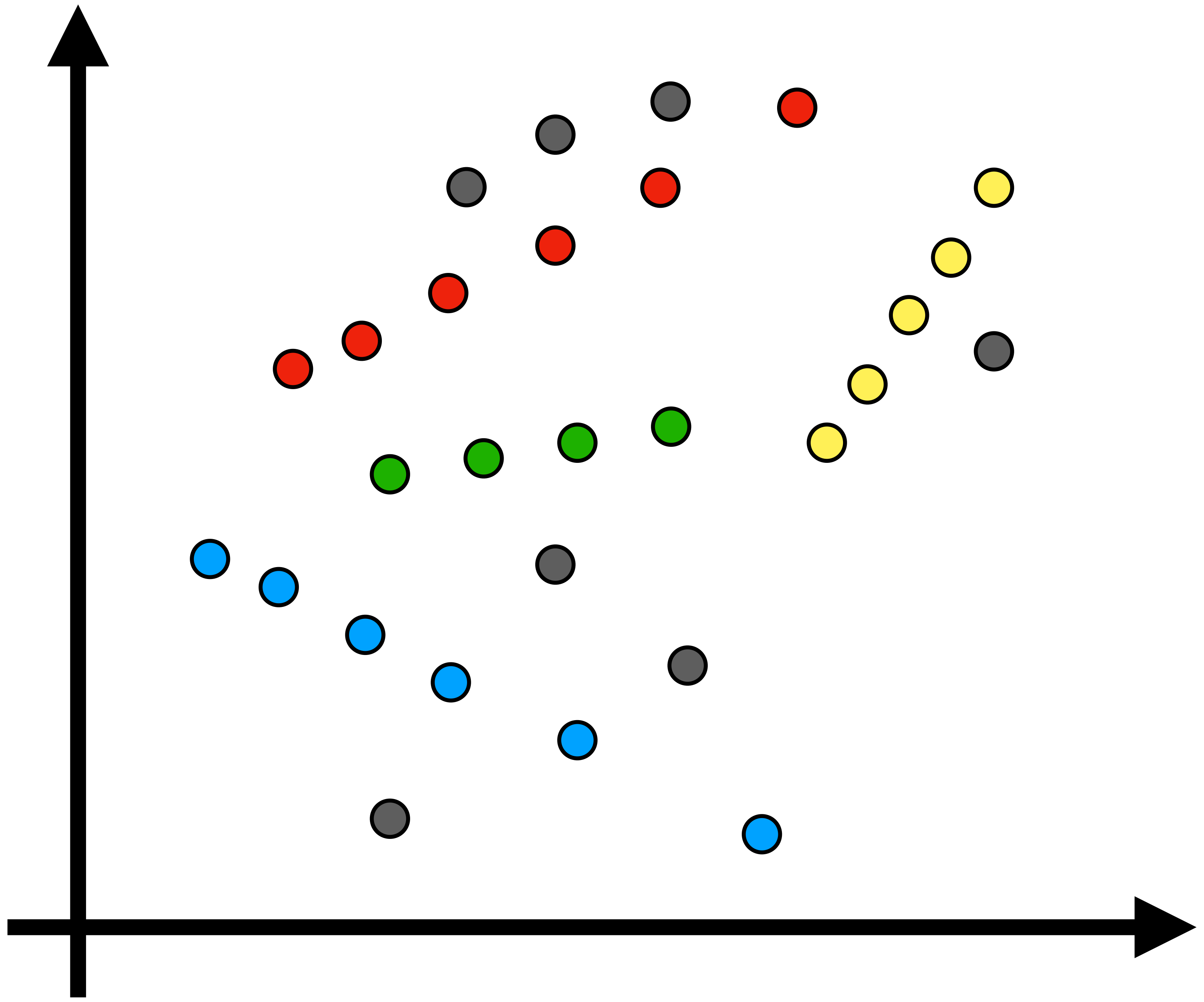
- Graph Neural Network-based pipeline for **track finding** in the VELO
- ETX4VELO, arXiv.2406.12869

- Comparable or superior physics performance to Allen
- Excellent **electron reconstruction** achieved using **triplets**
- Significantly reduced **fake rate**



# ETX4VELO

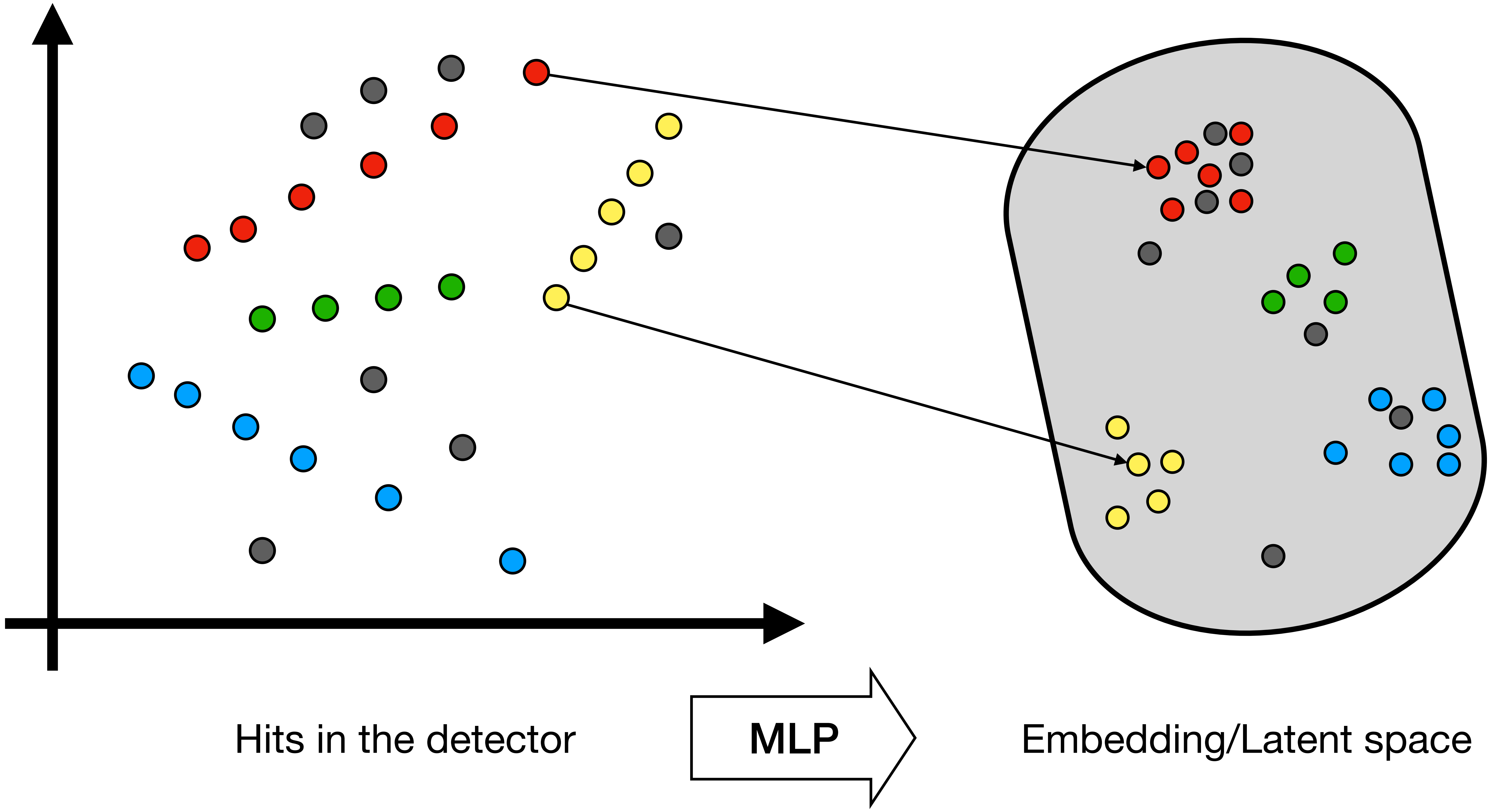
## How do we get a graph from the hits?



Hits in the detector

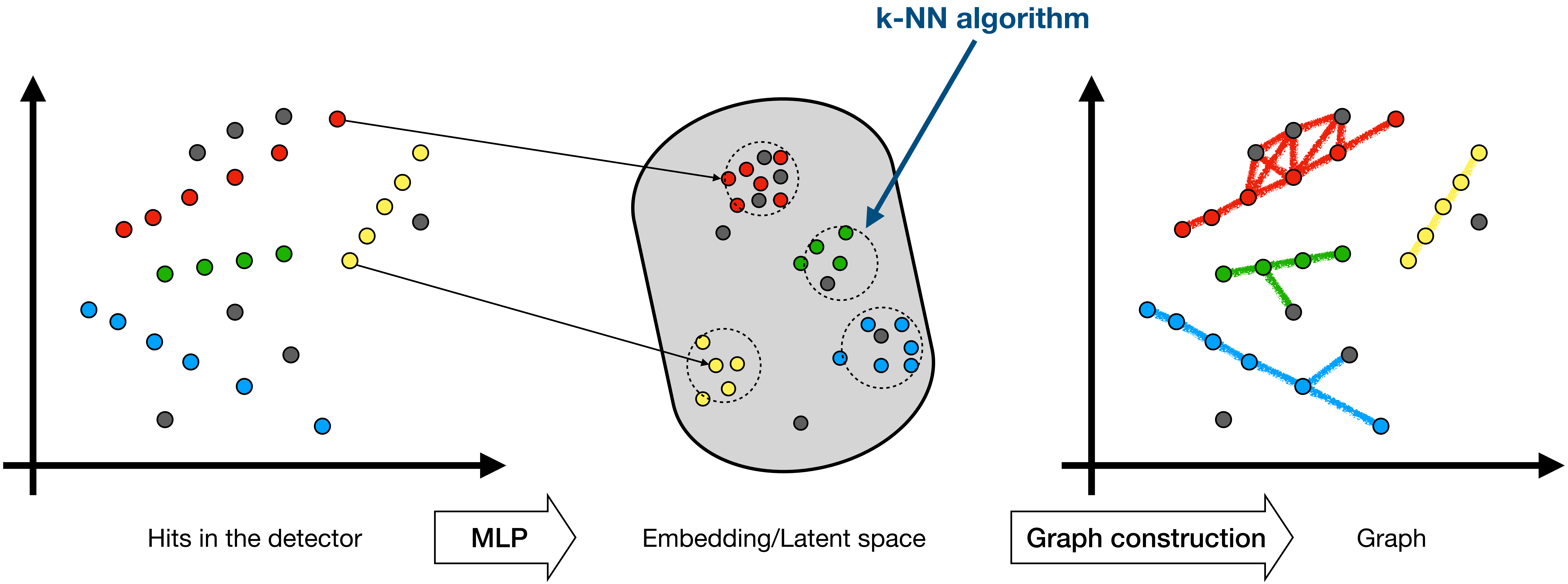
# ETX4VELO

## How do we get a graph from the hits?



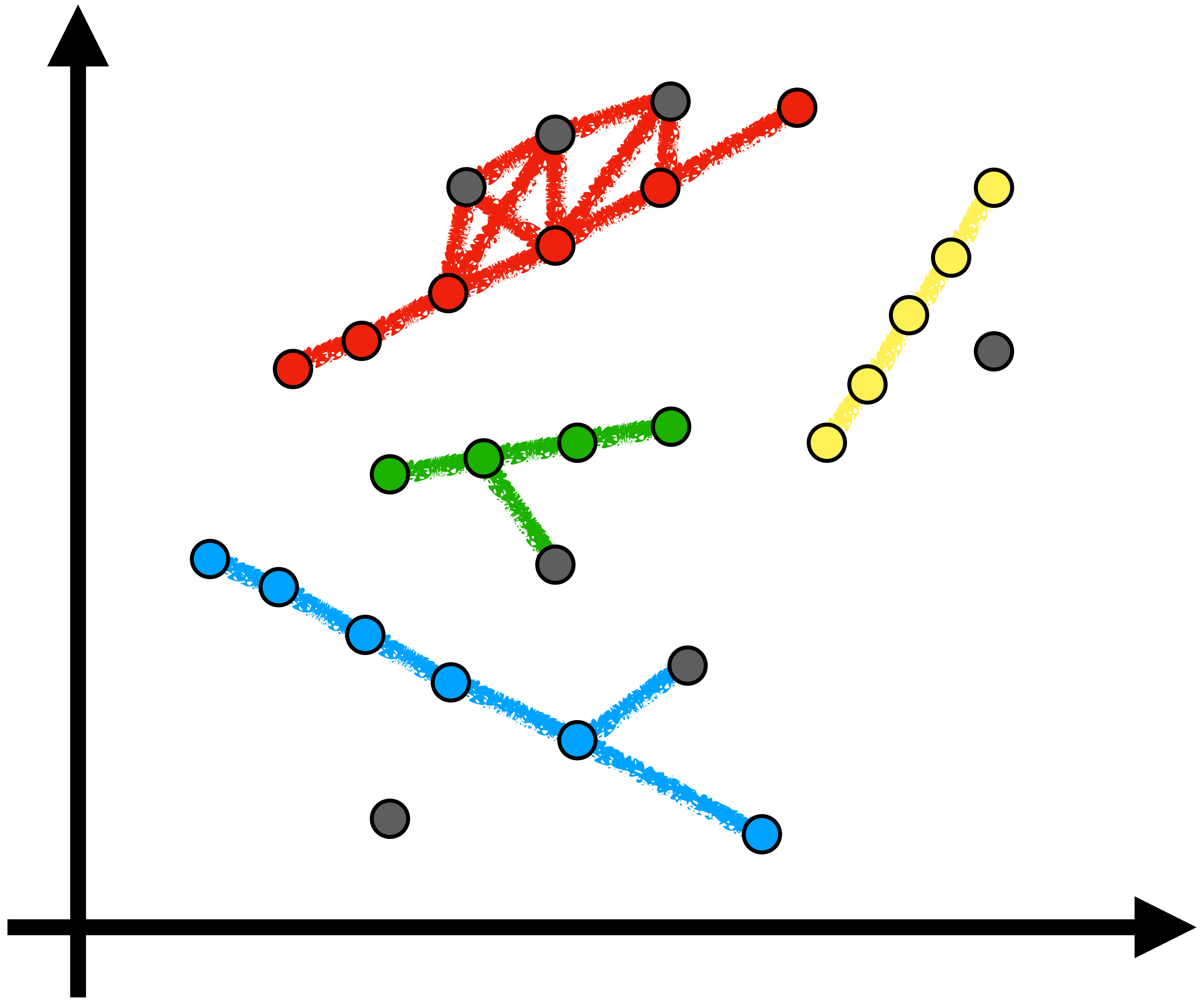
# ETX4VELO

## How do we get a graph from the hits?



# ETX4VELO

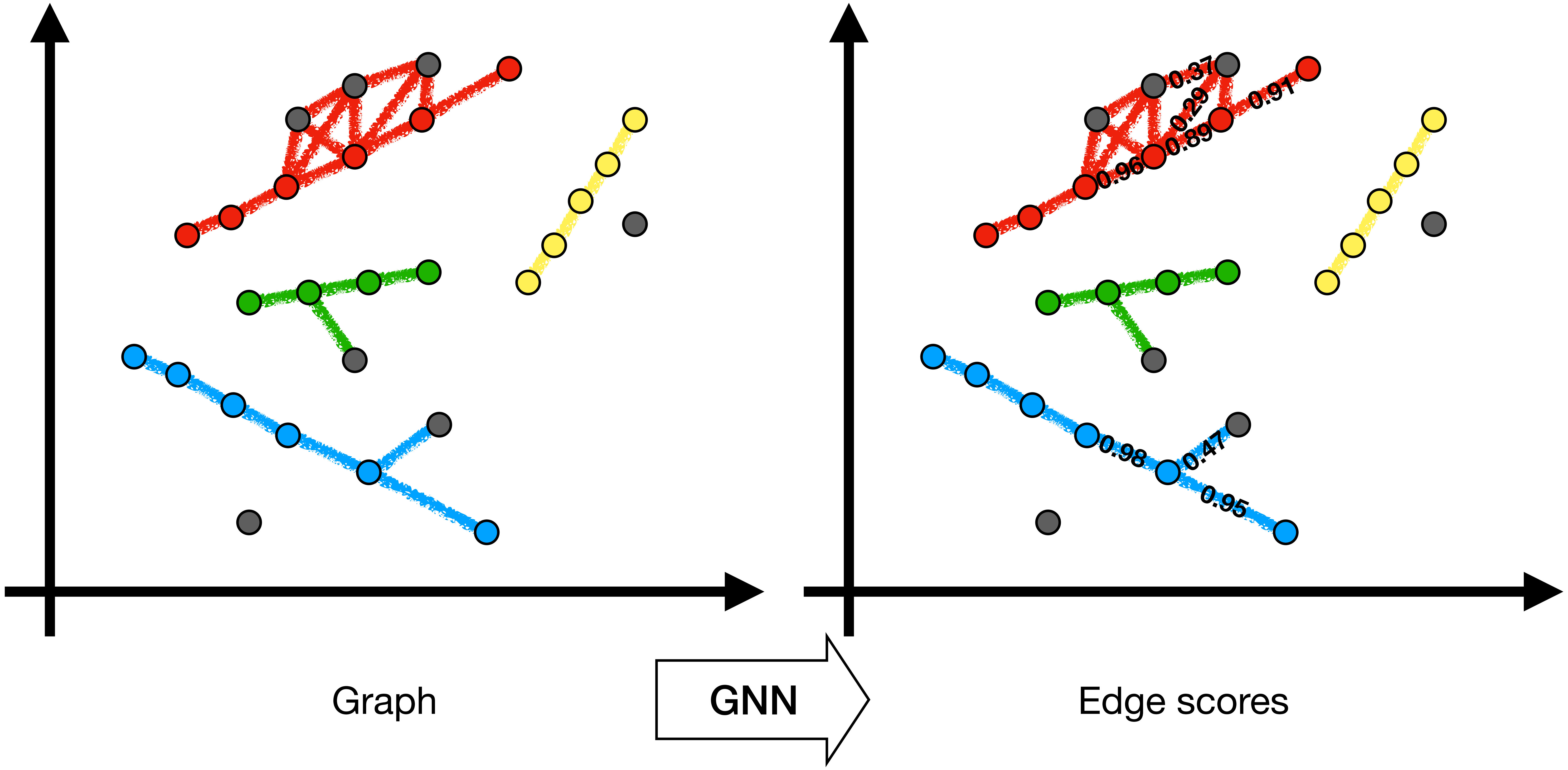
## How do we get tracks?



Graph

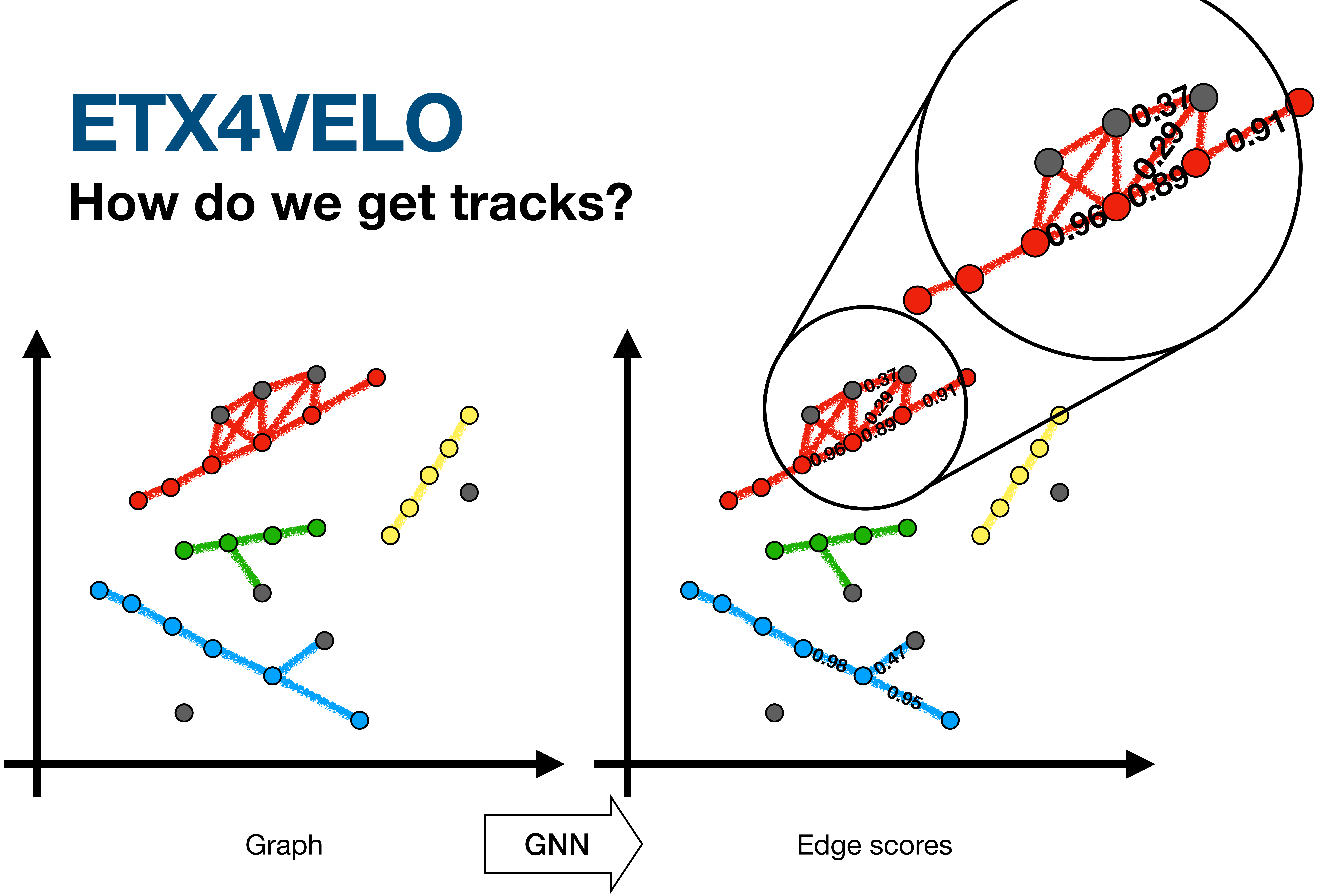
# ETX4VELO

## How do we get tracks?



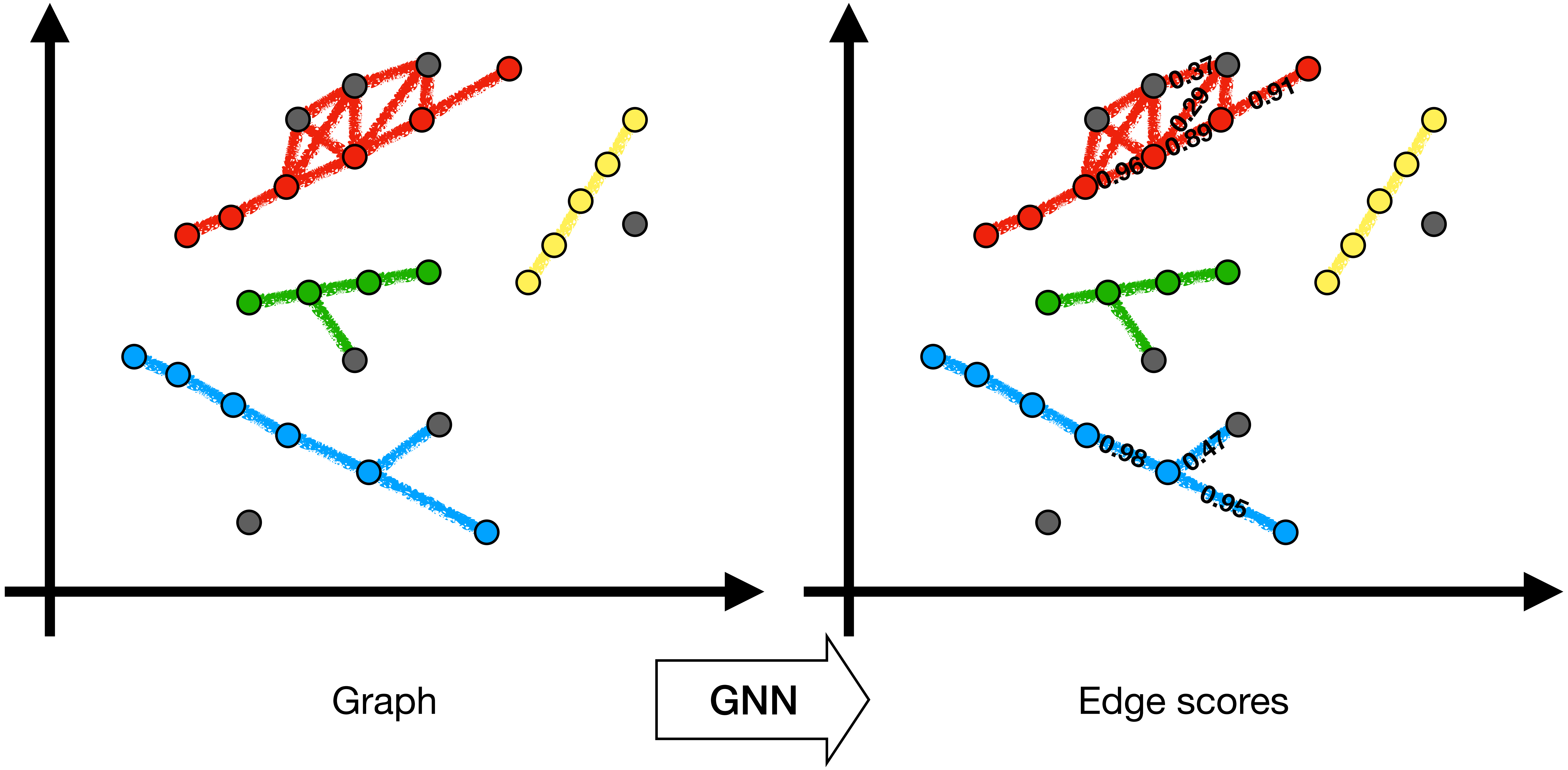
# ETX4VELO

## How do we get tracks?



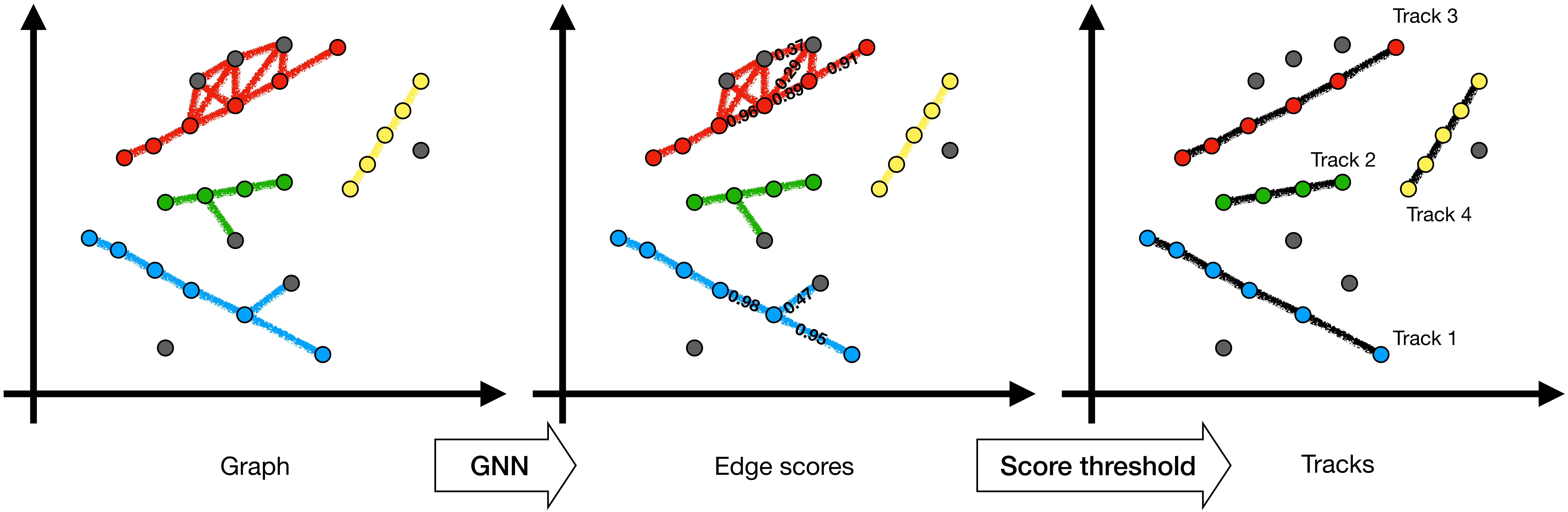
# ETX4VELO

## How do we get tracks?



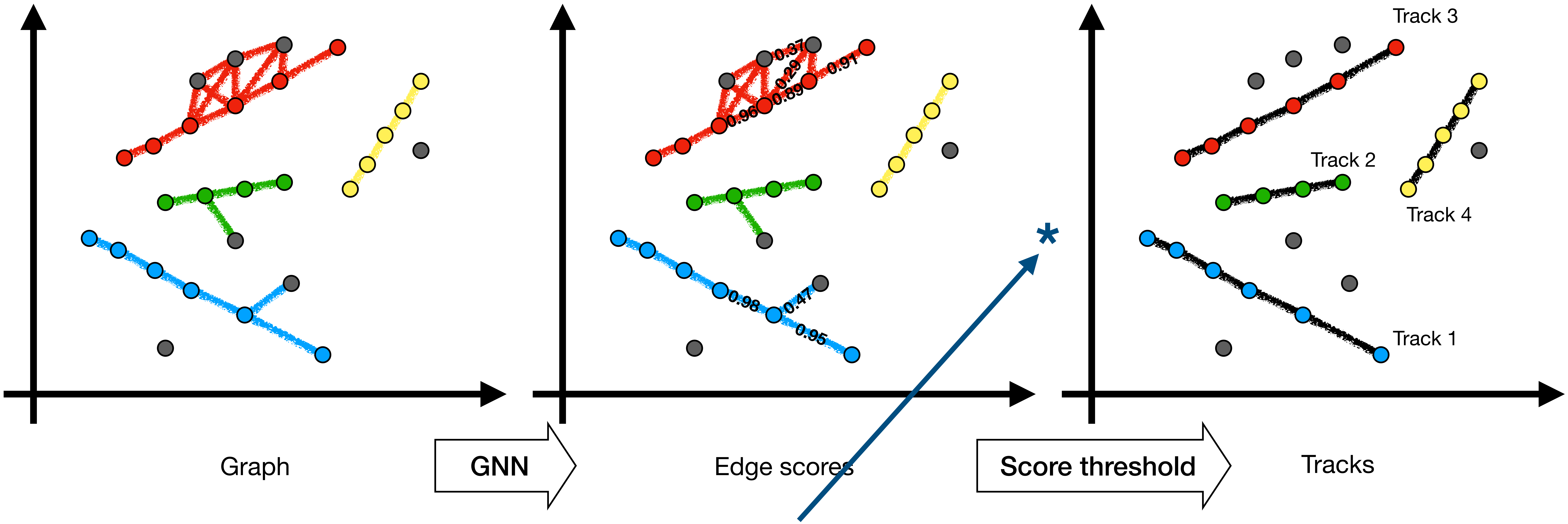
# ETX4VELO

## How do we get tracks?



# ETX4VELO

## How do we get tracks?




*** Triplet-based methodology to improve electron reconstruction, see backup**

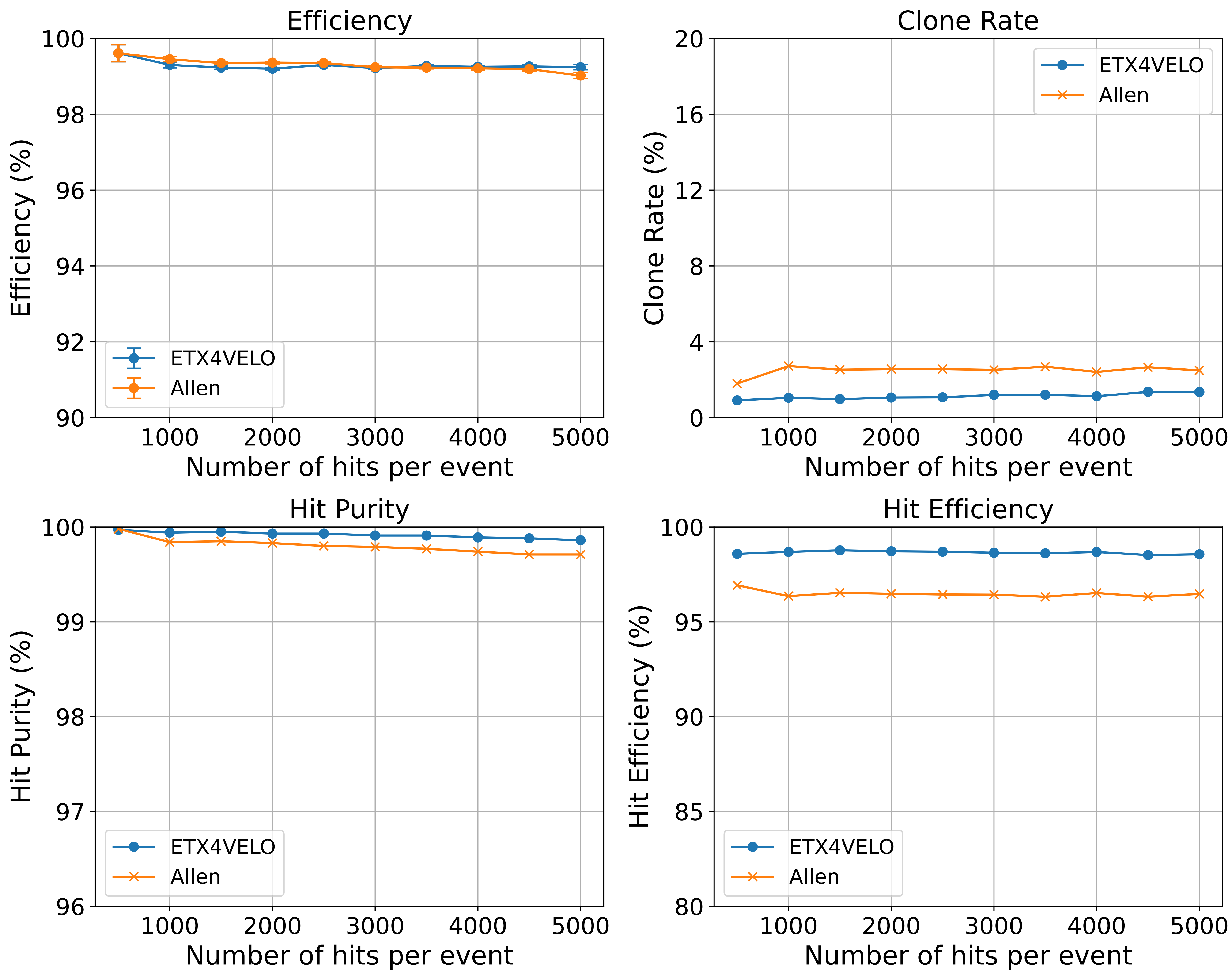
# ETX4VELO

## Physics performance

| | ALLEN | ETX4VELO |
|---|---|---|
| Fake rate | 2.17 % | 1.04 % |

*For particles leaving long tracks*

*For particles leaving long tracks, no electrons*



# ETX4VELO Main Objectives

- **Neural network for tracking with state-of-the-art physics performance** ✓
- High computational performance (throughput)

# ETX4VELO Main Objectives

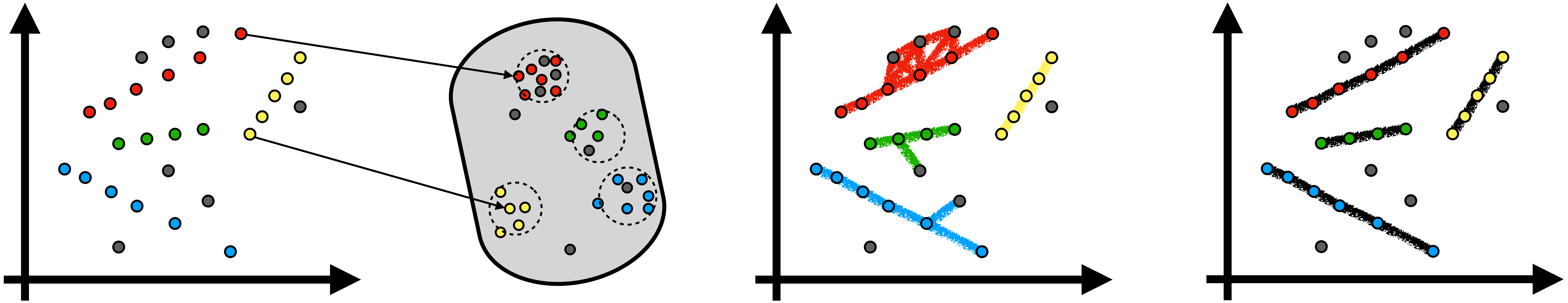
- Neural network for tracking with state-of-the-art physics performance ✓

- **High computational performance (throughput)**

**ETX4VELO inside LHCb framework (Allen)**

# ETX4VELO GPU Version

## Inference steps



| | | | | |
|---|---|---|---|---|
| Hits in the detector | **MLP** | Embedding/Latent space | **Graph construction** | Graph |
| Graph | **GNN** | Edge scores | **Score threshold** | Tracks |

# ETX4VELO GPU Version

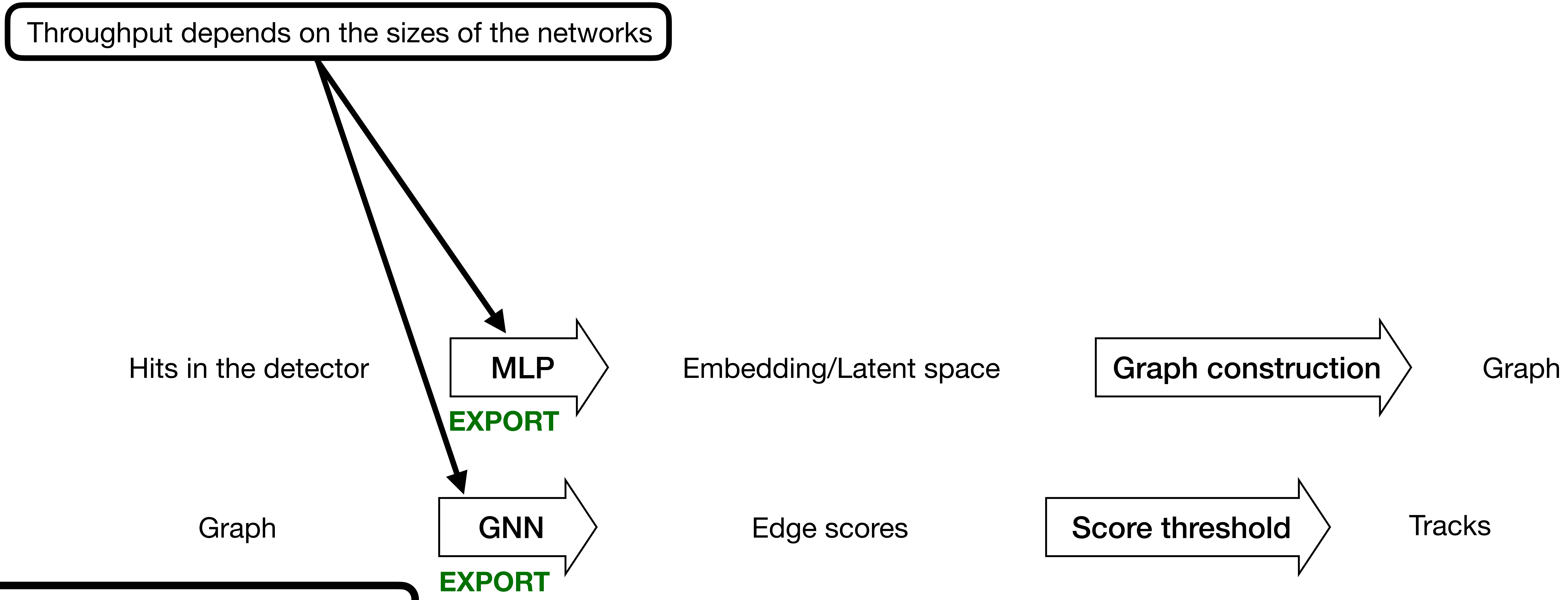
## Inference steps

Hits in the detector → **MLP** → Embedding/Latent space → **Graph construction** → Graph

Graph → **GNN** → Edge scores → **Score threshold** → Tracks

# ETX4VELO GPU Version

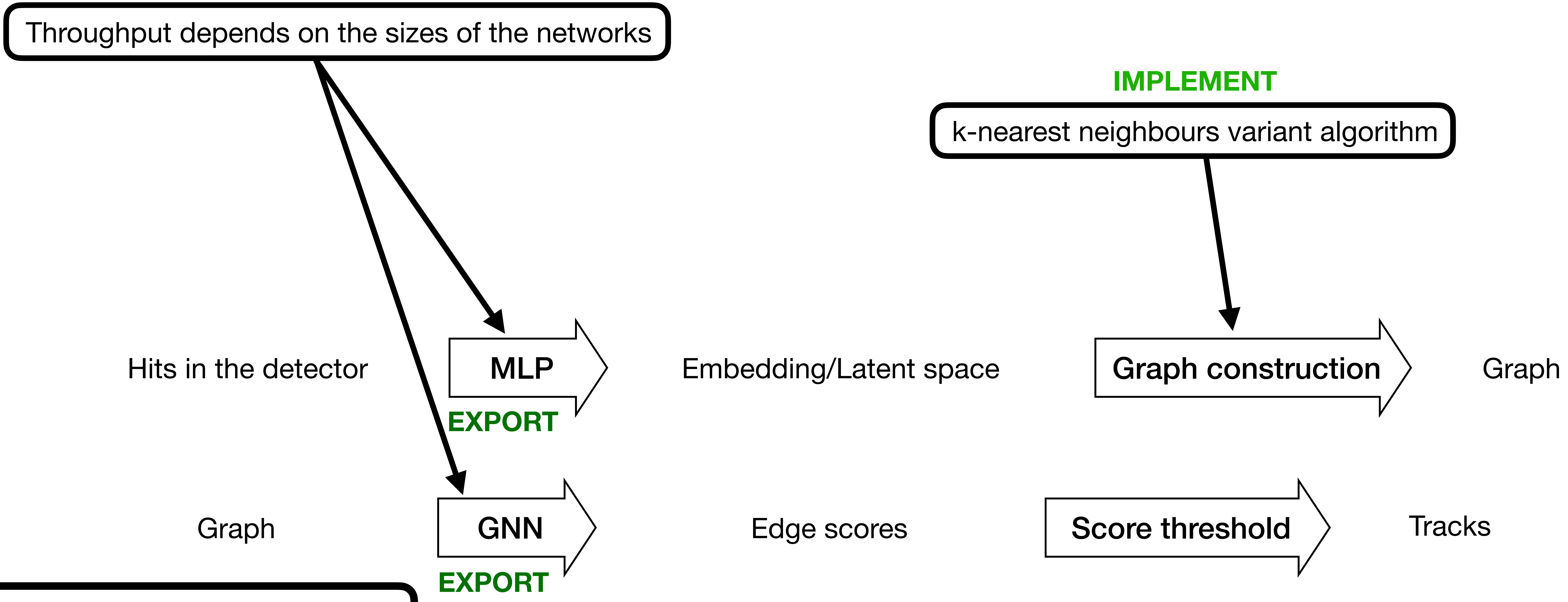
## Inference steps




- **EXPORT: ONNX**
- **IMPLEMENT: C++/CUDA**

# ETX4VELO GPU Version

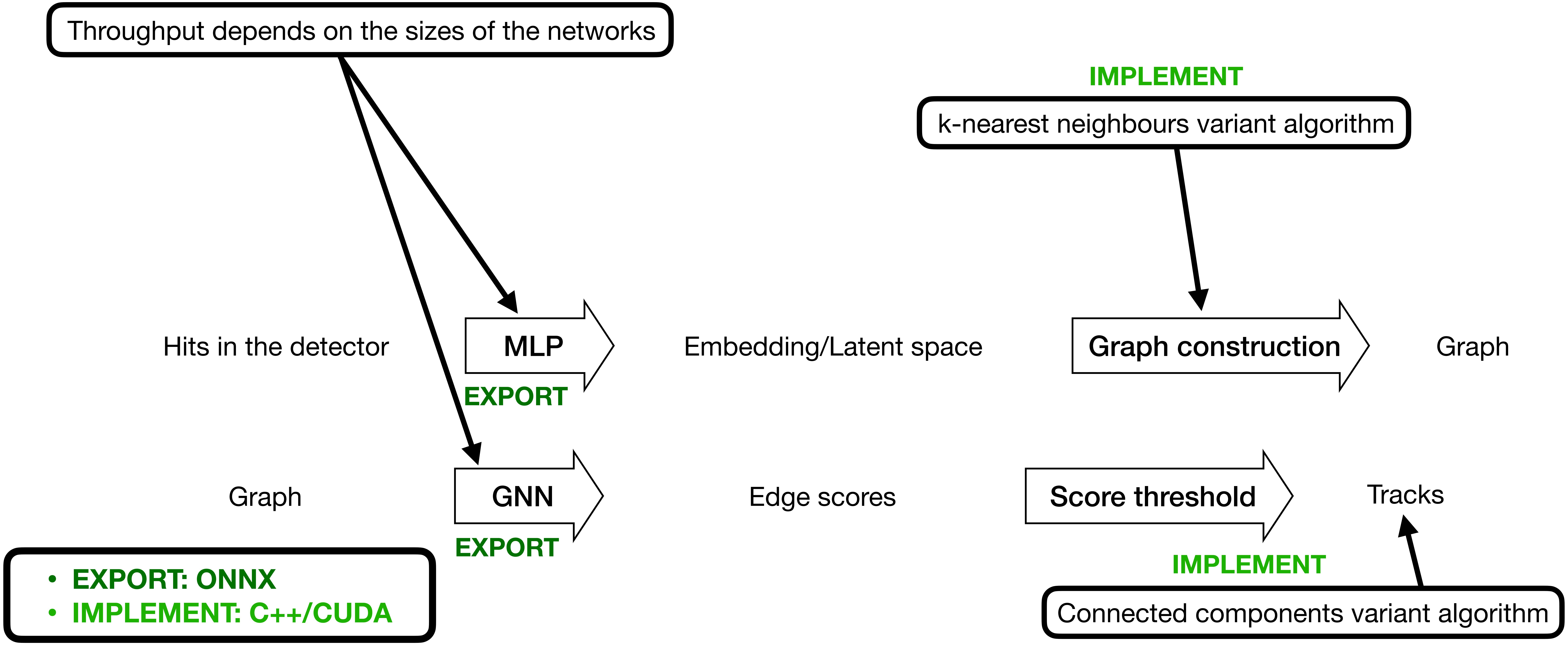
## Inference steps




- **EXPORT: ONNX**
- **IMPLEMENT: C++/CUDA**

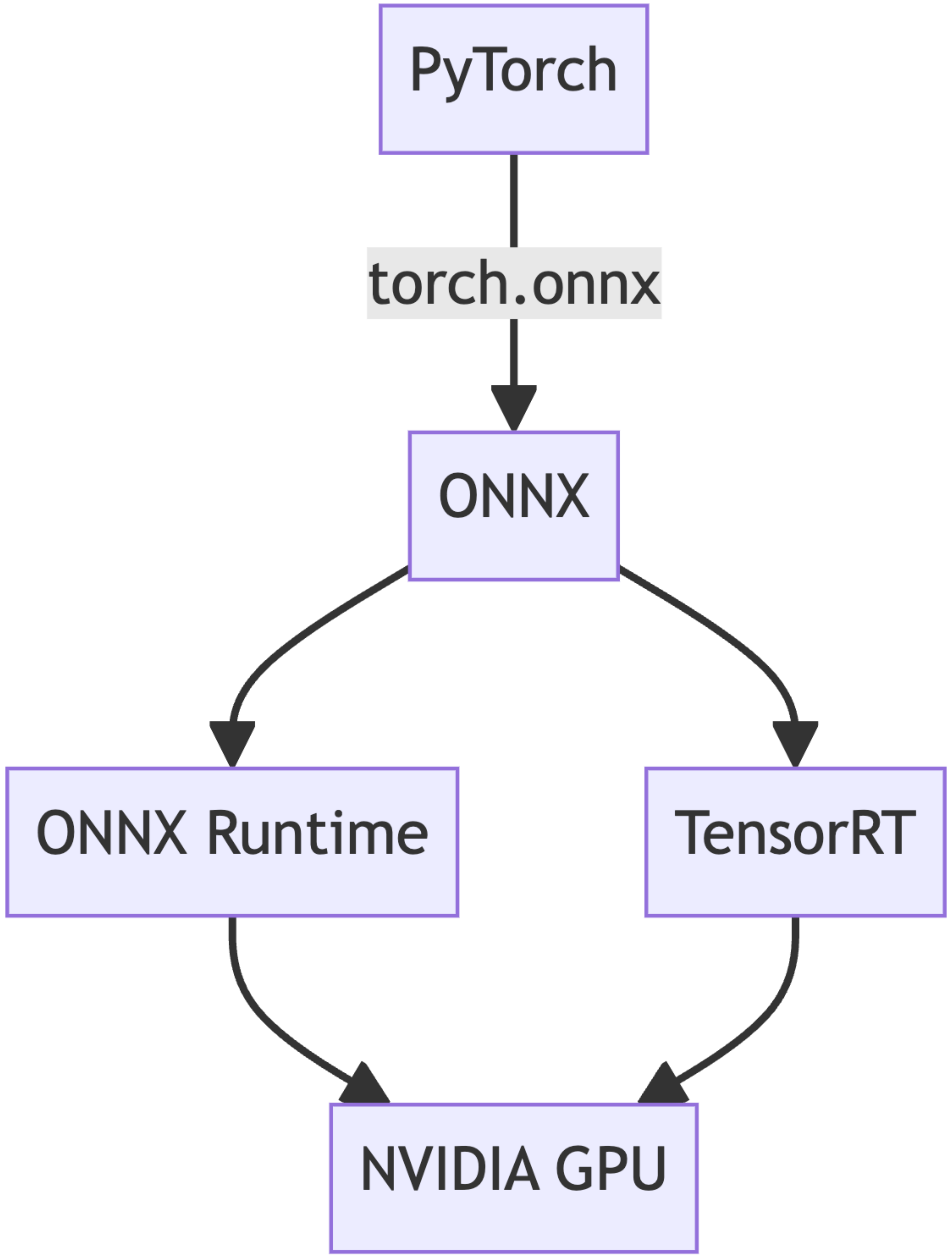
# ETX4VELO GPU Version

## Inference steps



# ML Inference on GPU (Allen)

- Throughput: infer events in **batches**
- Maximum number allowed by GPU memory

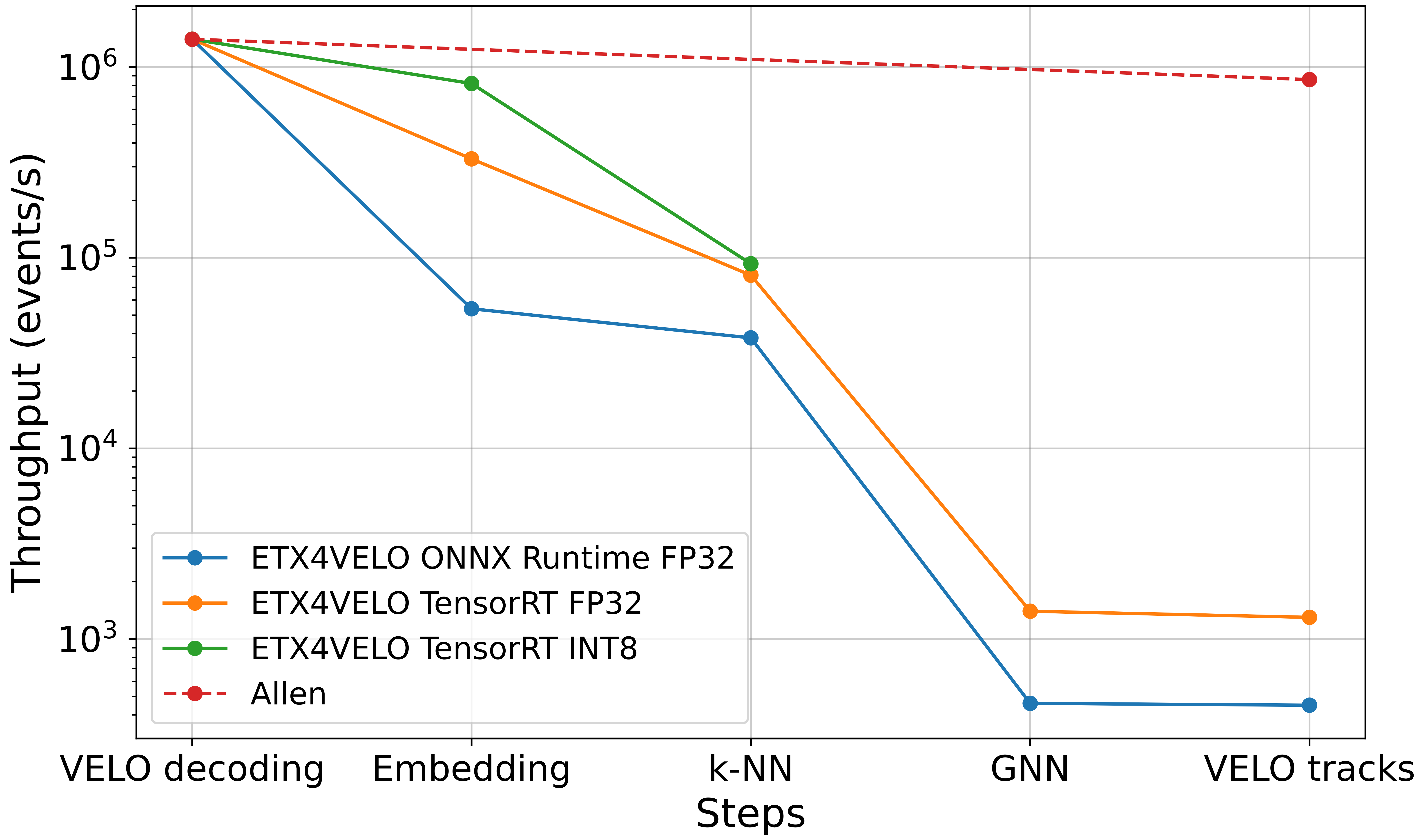



- **ONNX Runtime** + CUDA Execution Provider
- **TensorRT**

# ETX4VELO

## Computational performance

*NVIDIA GeForce RTX 3090*



# Conclusion

arXiv.2407.12119

**Track finding with ETX4VELO**

- **Superior physics performance** to state of the art reachable

**GPU version of ETX4VELO**

- **End-to-end** implementation in **LHCb** (Allen)

**Next steps**

- Quantization of the **GNN**
- Further optimization of the pipeline

## Thank you!

I would also like to thank the LHCb RTA reviewers, Núria Valls Canudas, Simon Akar, and Da Yu Tou, for their constructive comments

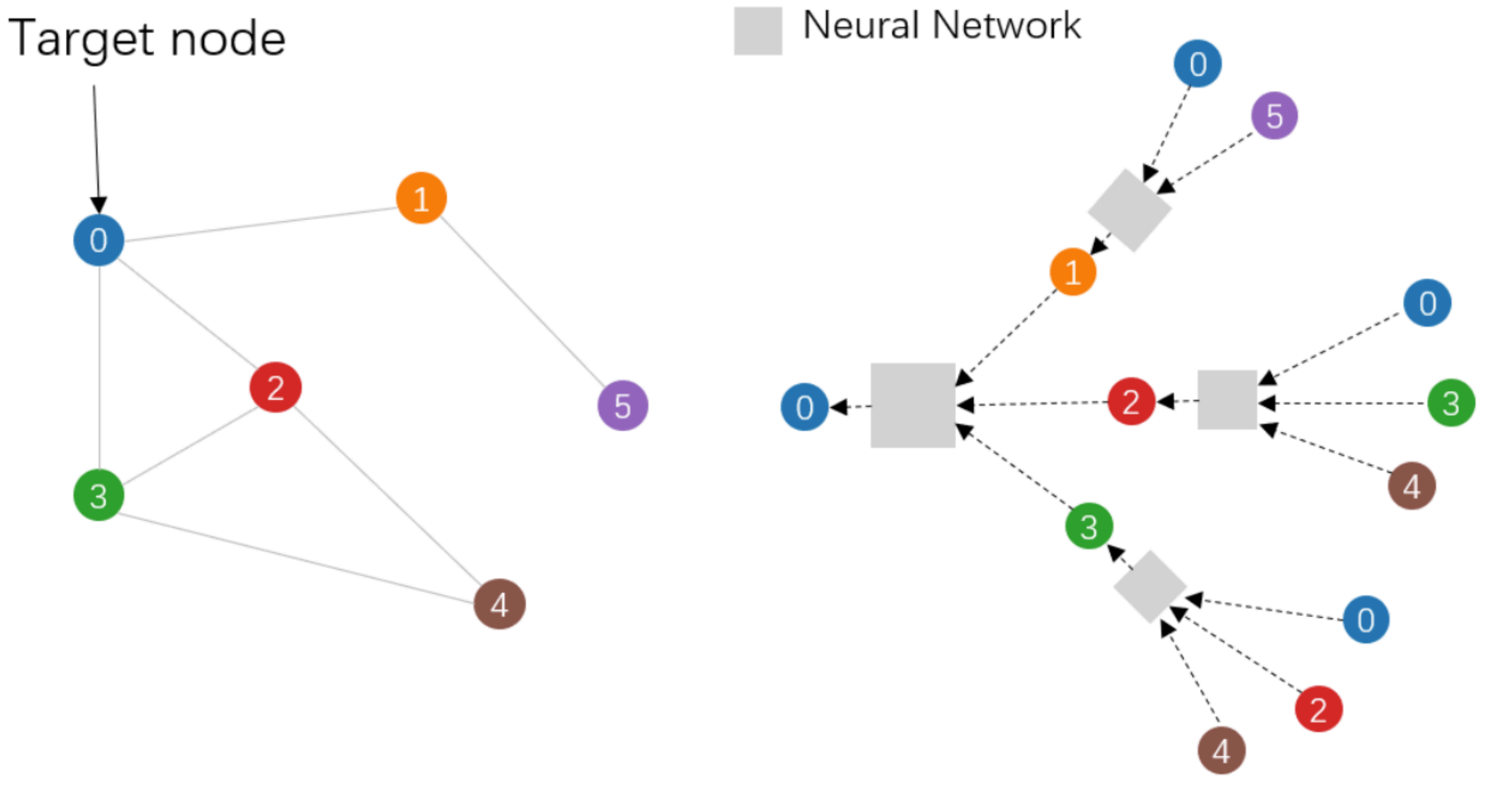
This work is part of the SMARTHEP network and it is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086, and in collaboration with Ivan Kisel and FIAS under the ANN4Europe project.

# Backup

# Graph Neural Networks

## How?

- GNN architecture:
  - Node features
  - Aggregation
  - **Neural message passing**

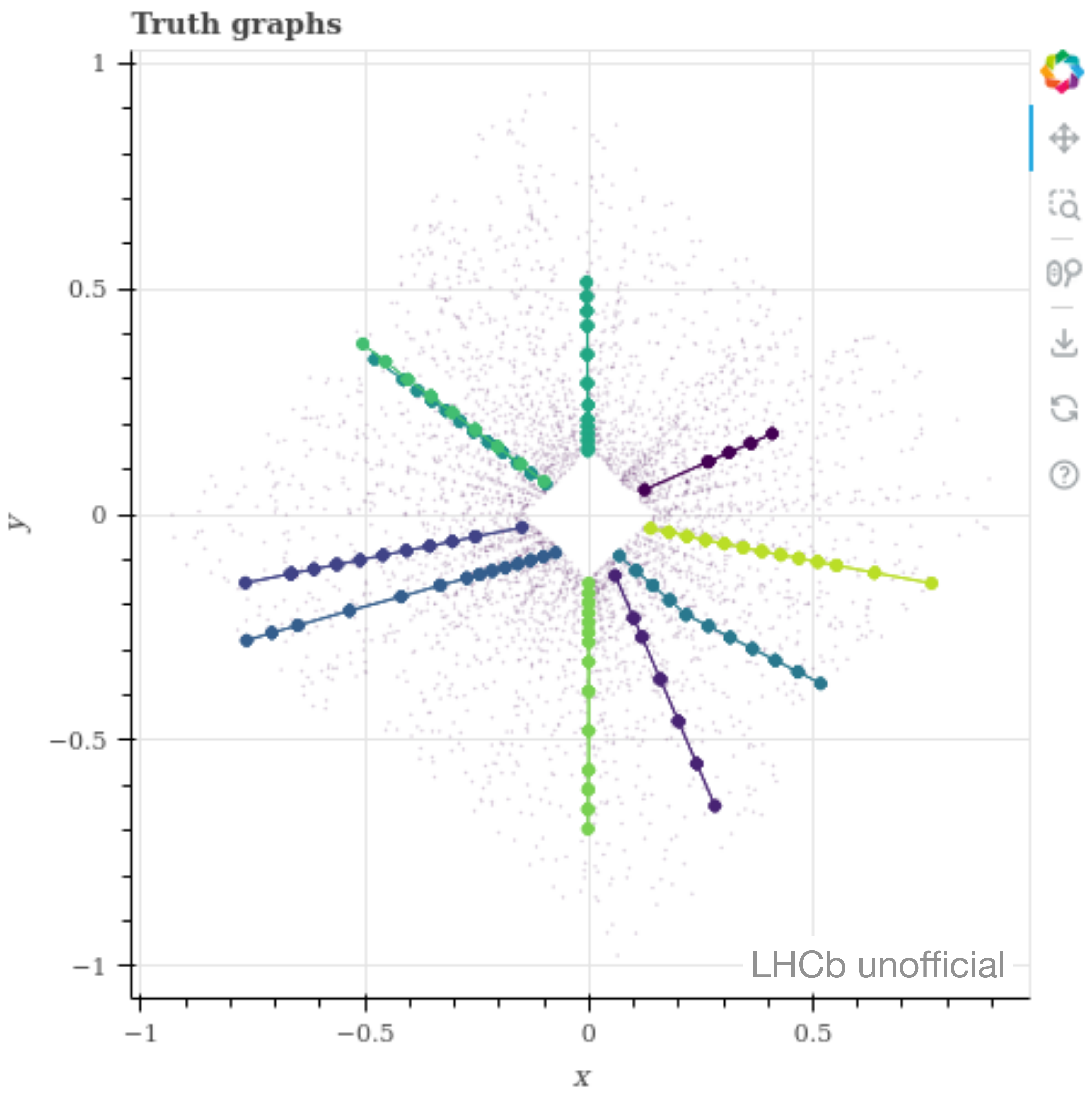


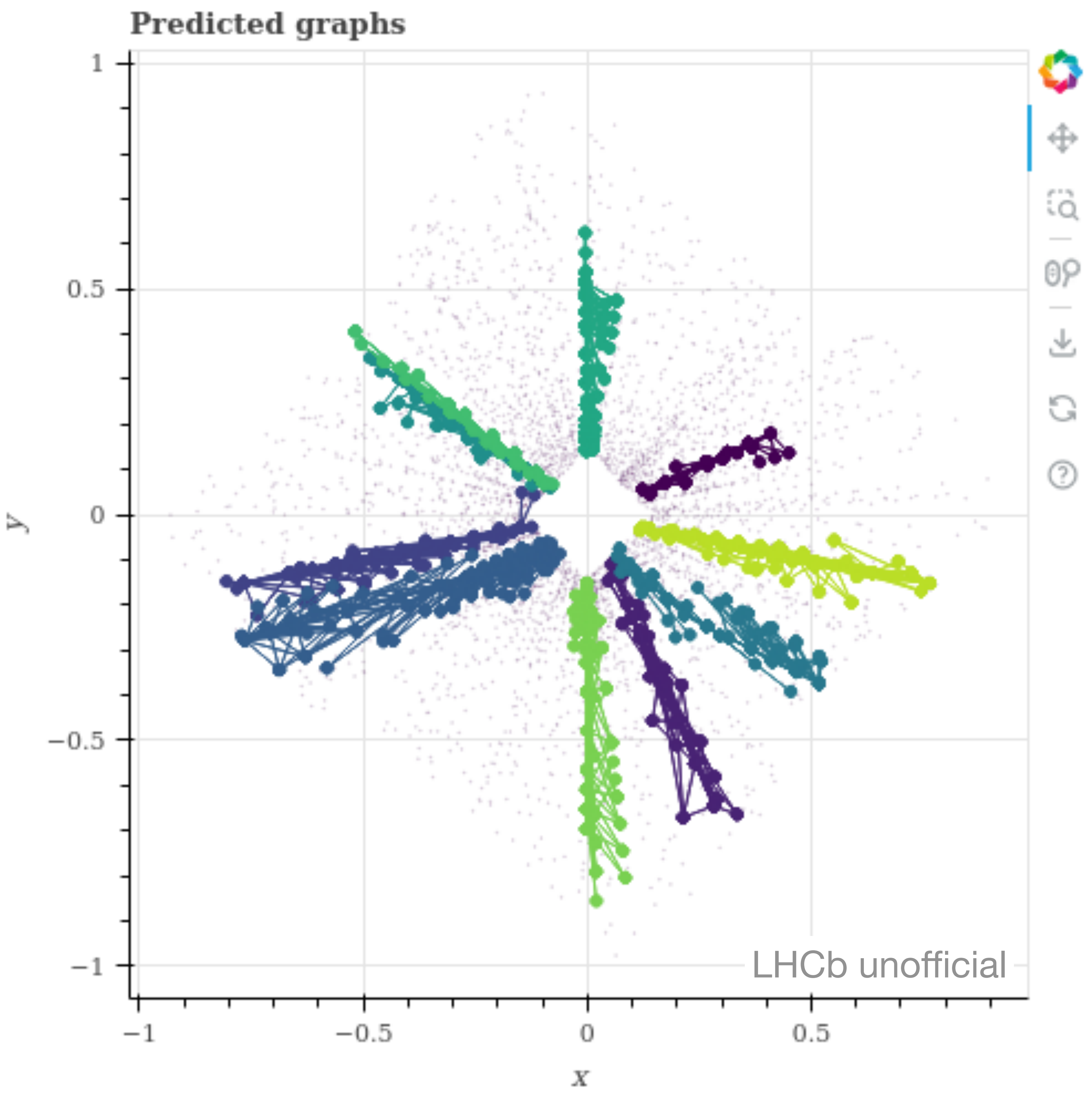
(a) Input graph (b) Neighborhood aggregation

[DOI:10.1109/TVCG.2022.3148107]

# ETX4VELO

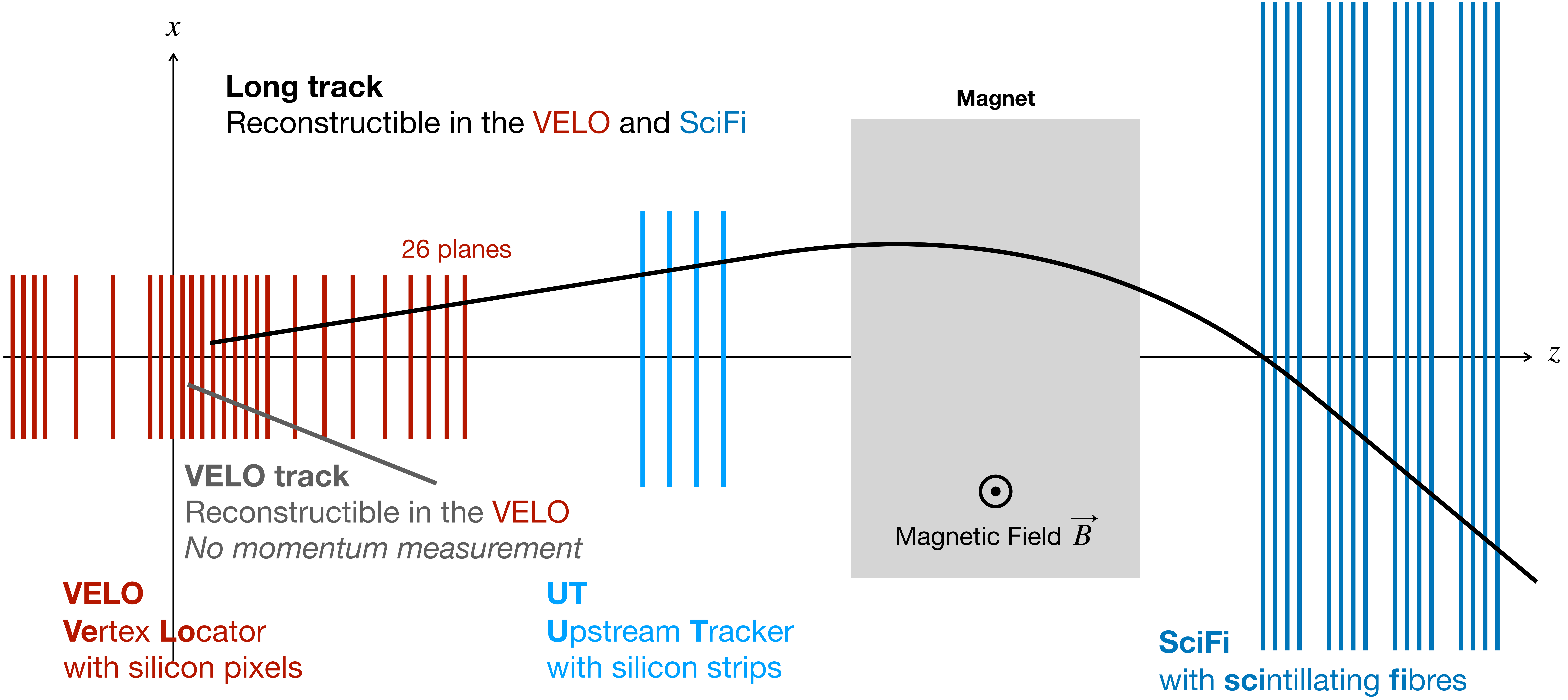
## How do we get a graph from the hits?



# The LHCb Detector

## Tracks in LHCb



# ETX4VELO

## Physics performance

| Proportion of | ALLEN | ETX4VELO |
|---|---|---|
| Reconstructed particles | 99.06 % | 99.23 % |
| Duplicate tracks | 2.63 % | 1.37 % |
| Fake tracks | 2.17 % | 1.04 % |

*For particles leaving long tracks*

# ETX4VELO

## Problem with electrons: shared hits

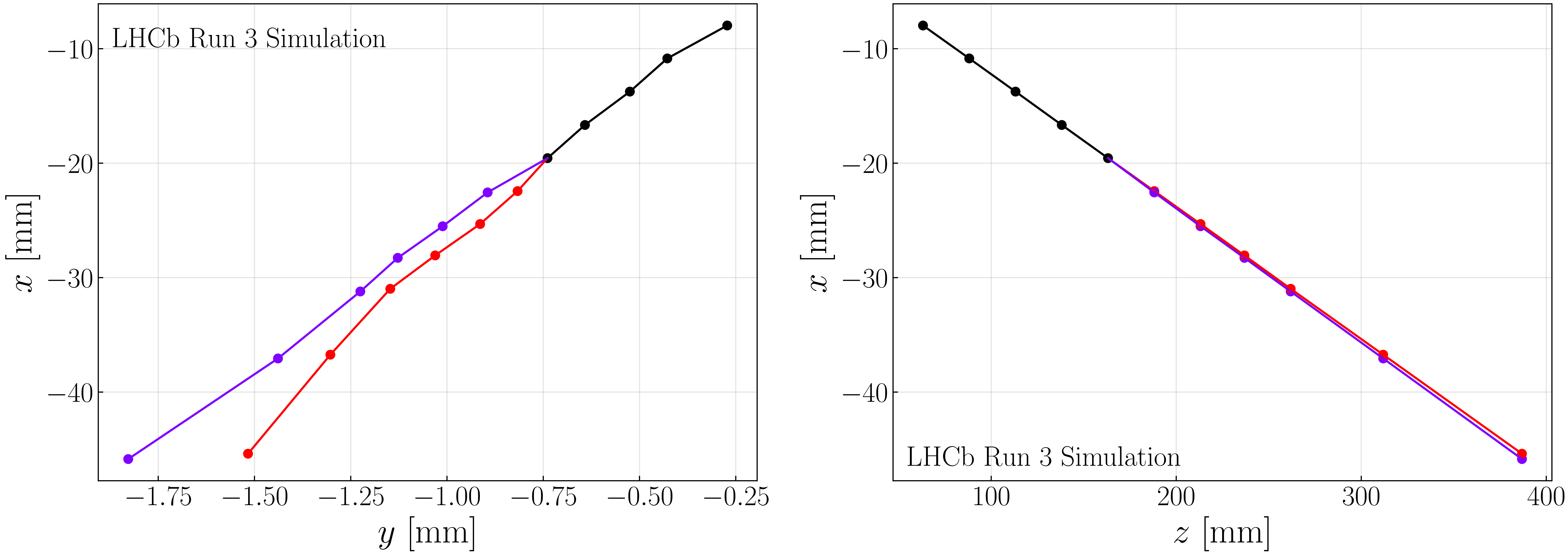
- Problem with electrons:
  - Material interactions —> positron-electron pairs
  - ~ **55%** electrons/positrons share hits with one another
  - Then split up
  - Electrons with “long tracks” = **“long electrons”**
  - Important for the LHCb physics program
- Solution: use edge-edge connections (triplets)

```
TrackChecker output                          :     38049/  1117828   3.40% ghost
01_velo                                      :    491643/   520515  94.45% ( 95.
02_long                                      :    286719/   296345  96.75% ( 97.
03_long_P>5GeV                               :    185866/   189727  97.96% ( 98.
04_long_strange                              :     13654/    15243  89.58% ( 90.
05_long_strange_P>5GeV                       :      6606/     7229  91.38% ( 92.
06_long_fromB                                :       497/      513  96.88% ( 96.
07_long_fromB_P>5GeV                         :       335/      343  97.67% ( 97.
08_long_electrons                            :     16634/    21330  77.98% ( 78.
09_long_fromB_electrons                      :        41/       58  70.69% ( 76.
10_long_fromB_electrons_P>5GeV               :        30/       38  78.95% ( 81.

*** Benchmark score:  94.01

| Categories             | Efficiency    | Average efficiency    | % clones    | Average h
|:-----------------------|:--------------|:----------------------|:------------|:---------
| Velo                   | 90.37%        | 91.08%                | 1.41%       | 99.03%
| Long                   | 95.49%        | 95.97%                | 0.97%       | 99.33%
| Velo, no electrons     | 94.45%        | 95.11%                | 0.89%       | 99.30%
| Velo, only electrons   | 69.30%        | 69.84%                | 4.91%       | 97.15%
| Long, only electrons   | 77.98%        | 78.93%                | 3.54%       | 97.36%

| Categories    | # ghosts    | # tracks    | % ghosts    |
|:--------------|:------------|:------------|:------------|
| Everything    | 38,049      | 1,117,828   | 3.40%       |
```

LHCb unofficial

# ETX4VELO

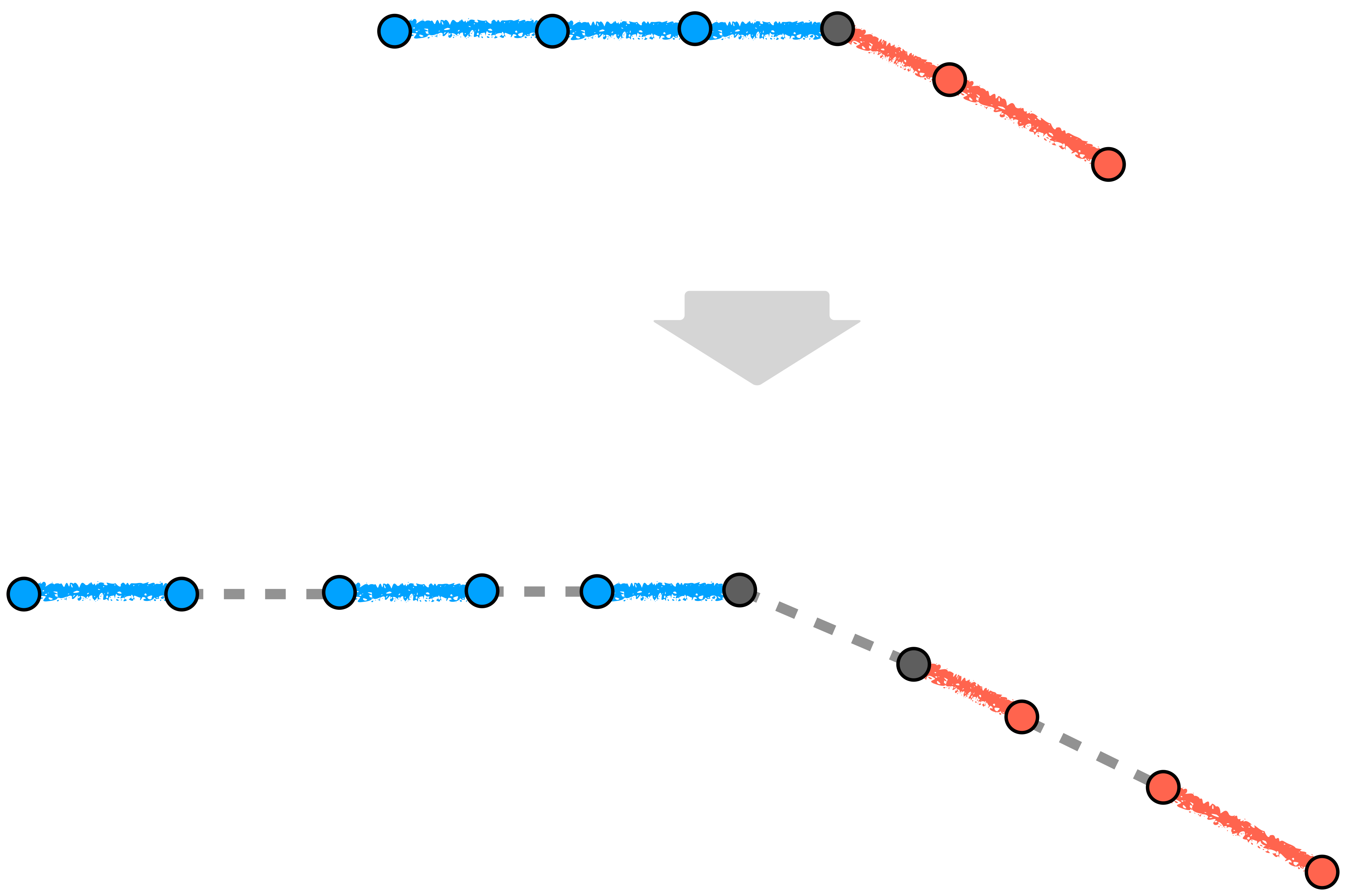
## Problem with electrons: shared hits



# ETX4VELO

## Problem with electrons: the solution

- Problem with electrons:
  - Pipeline cannot separate particle with shared edges
  - **Hit-hit** connections are not enough
  - Solution:
  - Use **edge-edge** connections (**triplets**)
  - Use GNN again on triplets



# ETX4VELO

## Computational performance

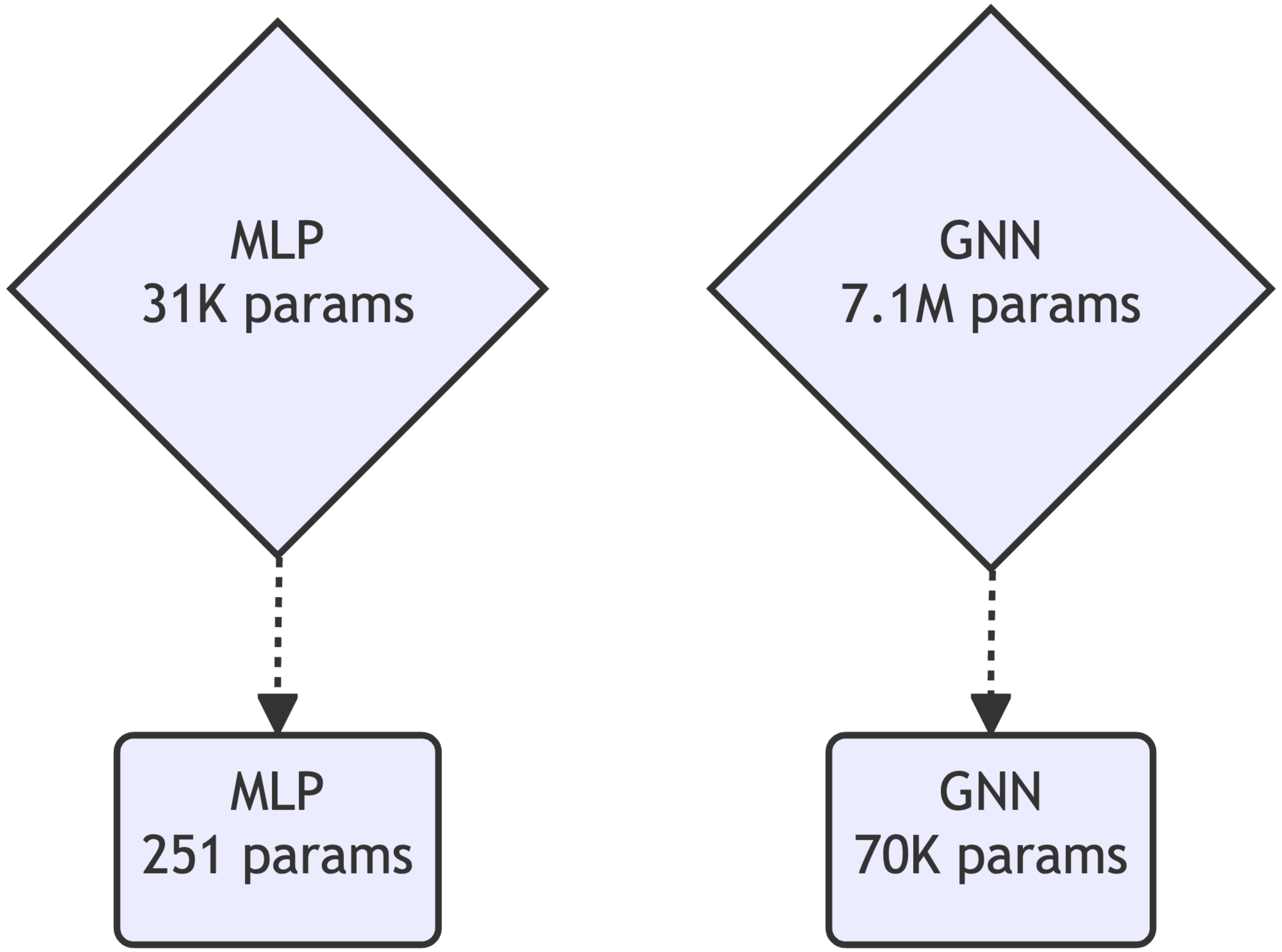
*NVIDIA GeForce RTX 3090, 500 events, 50 repetitions*

| Pipeline | Up to step | Throughput (events/s) | | |
|---|---|---|---|---|
| | | **ONNX Runtime FP32** | **TensorRT FP32** | **TensorRT INT8** |
| **ETX4VELO** | VELO decoding | 1,400k | | |
| | Embedding | 54k | 330k | **820k** |
| | k-NN | 38k | 81k | 93k |
| | GNN | 0.46k | 1.4k | - |
| | VELO tracks | 0.45k | **1.3k** | - |
| **ALLEN** | VELO tracks | **860k** | | |

# ETX4VELO GPU Version

## Implementation details

- Global nearest neighbour search vs **using only adjacent planes**
- Unidirectional vs bidirectional graph
- General connected components algorithm vs **geometry-specific**
- TensorRT plugin for GNN operation
- Custom **memory allocator** for the inference engine



# ETX4VELO GPU Version

## ONNX Runtime vs TensorRT implementation

| ONNX Runtime | TensorRT |
| --- | --- |
| Better out-of-the-box support | Better documentation |
| CPU backend | Lower memory footprint |
| | Higher throughput |
| | Memory managers reconciled more easily |