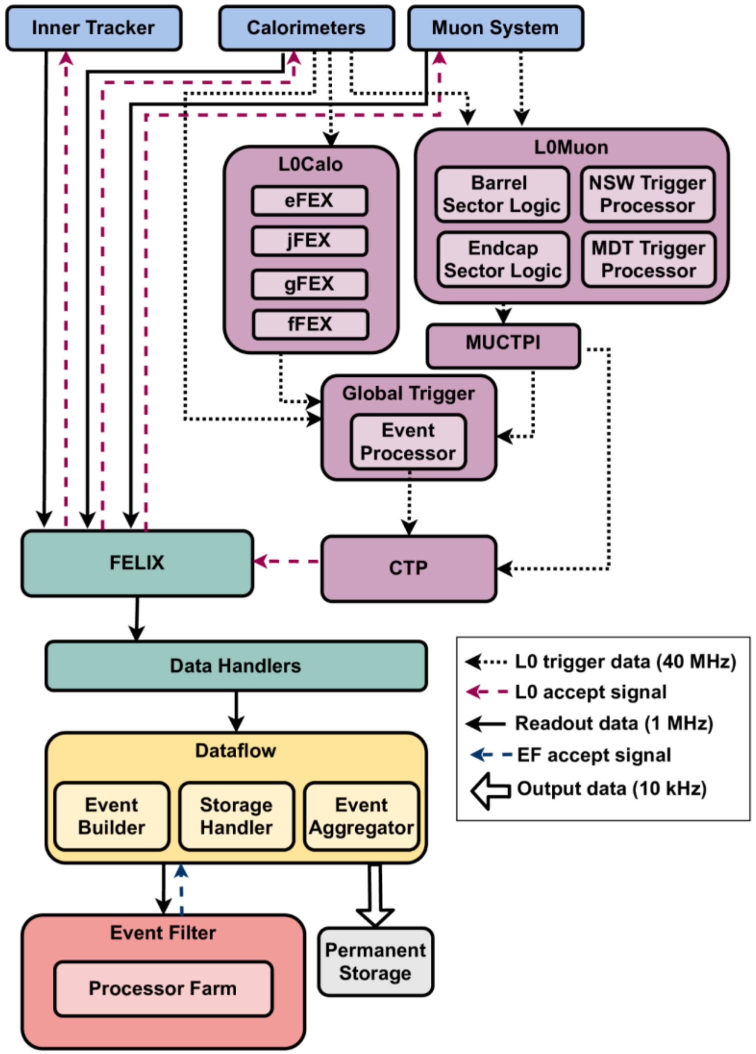
ReTrac(k)ing our Steps: Track Reconstruction with Graph Neural Networks

19.07.2024

Poppy Hicks

Heidelberg @ Trifels

ATLAS Phase-II Upgrade: **TDAQ**

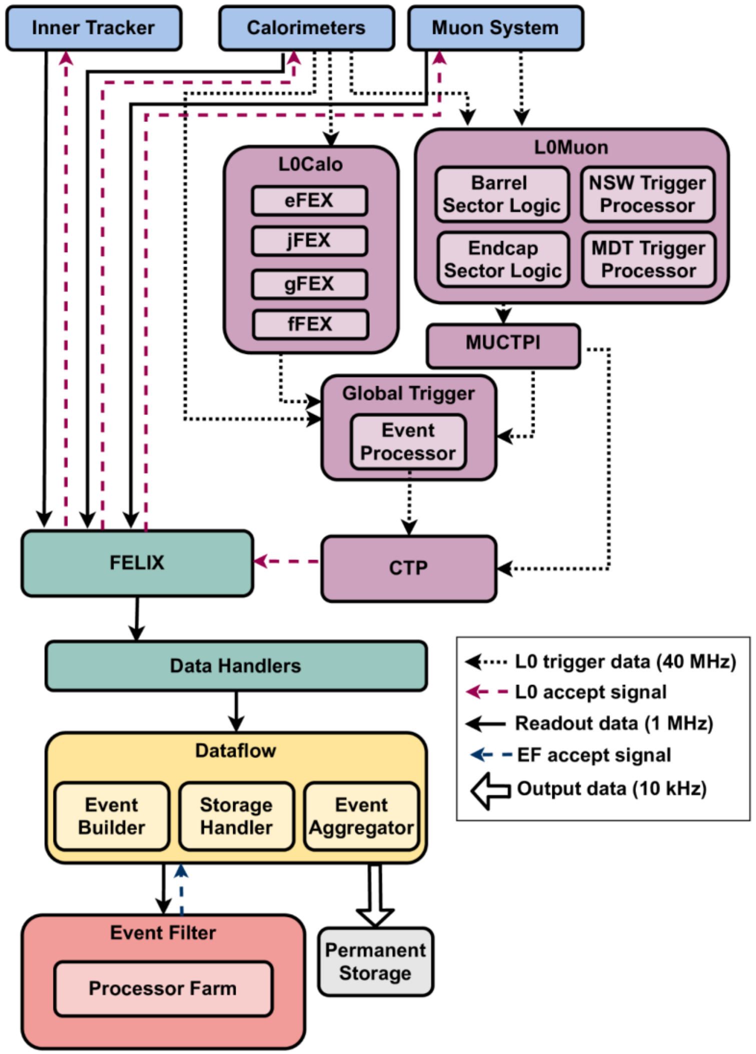


AIM OF TDAQ: to ensure optimal data-taking conditions and select most interesting collision events for study

L0 Trigger: Processes muon and calorimeter data at 40 MHz. Accepts at 1 MHz.

Event Filter: Particle tracks reconstructed with ITk data; full event reconstruction. Accepts at 10 KHz.

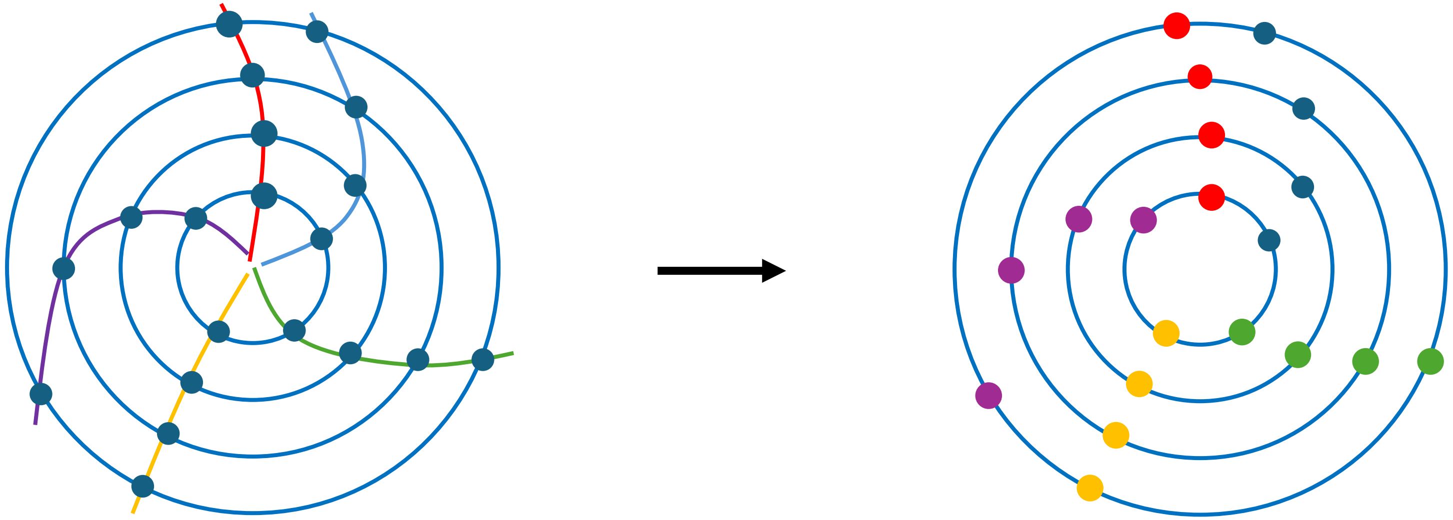
ATLAS Phase-II Upgrade: **TDAQ**



AIM OF TDAQ: to ensure optimal data-taking conditions and select most interesting collision events for study

Provide tracks that allow for filtering down to 10 kHz of "interesting events" by the Event Filter

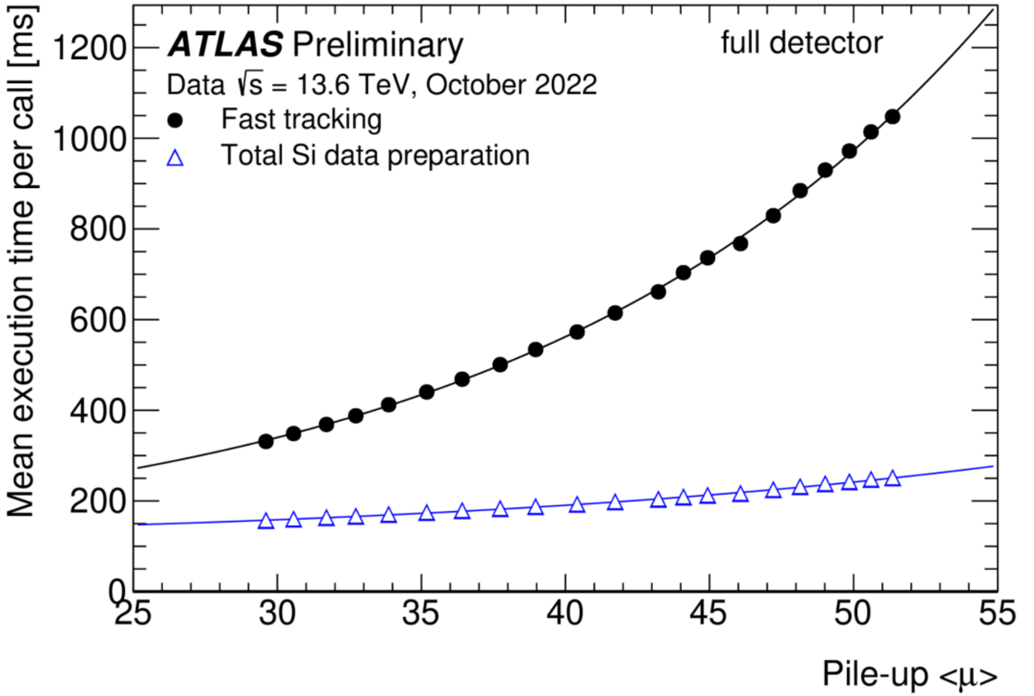
Track Finding



TRACK RECONSTRUCTION:
Label successive hits from the same particle as a ‘track’

Track Finding

- **Combinatorial Kalman Filter (CKF)**
  - Traditional method
  - High efficiency, low fake rate
  - Scales worse than linear in event size
  - 'CPU intensive'
    - [ATLAS TDAQ Tracking Amendment]()



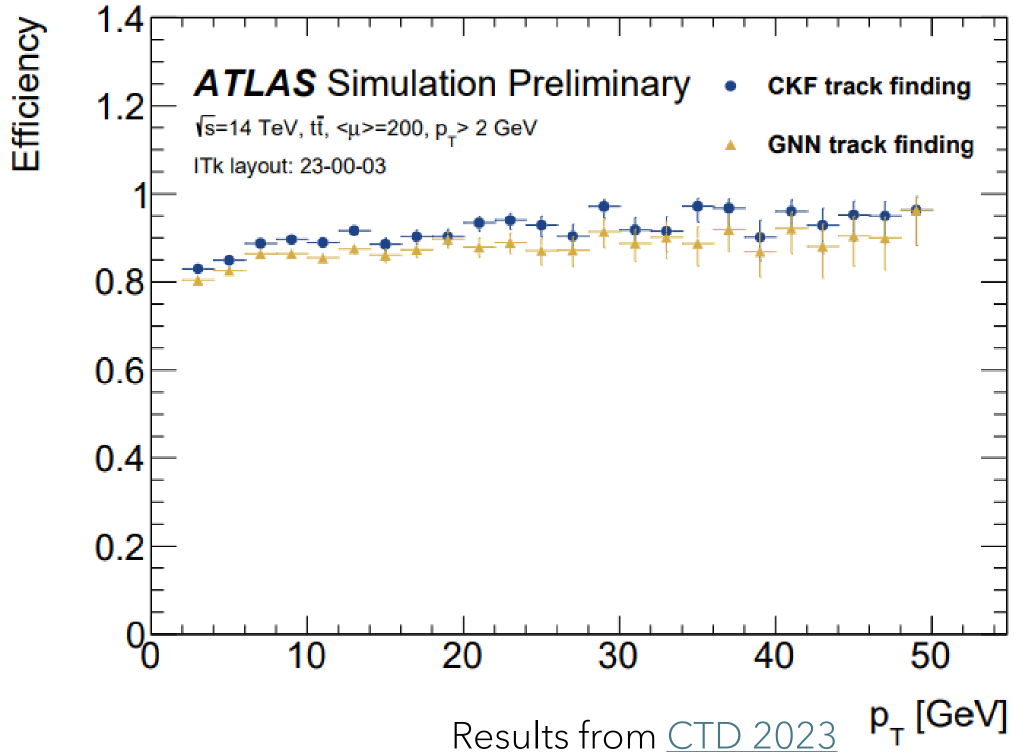
https://cds.cern.ch/record/2875779

→ R&D on track reconstruction acceleration

Track reconstruction with graph neural networks

Track Finding

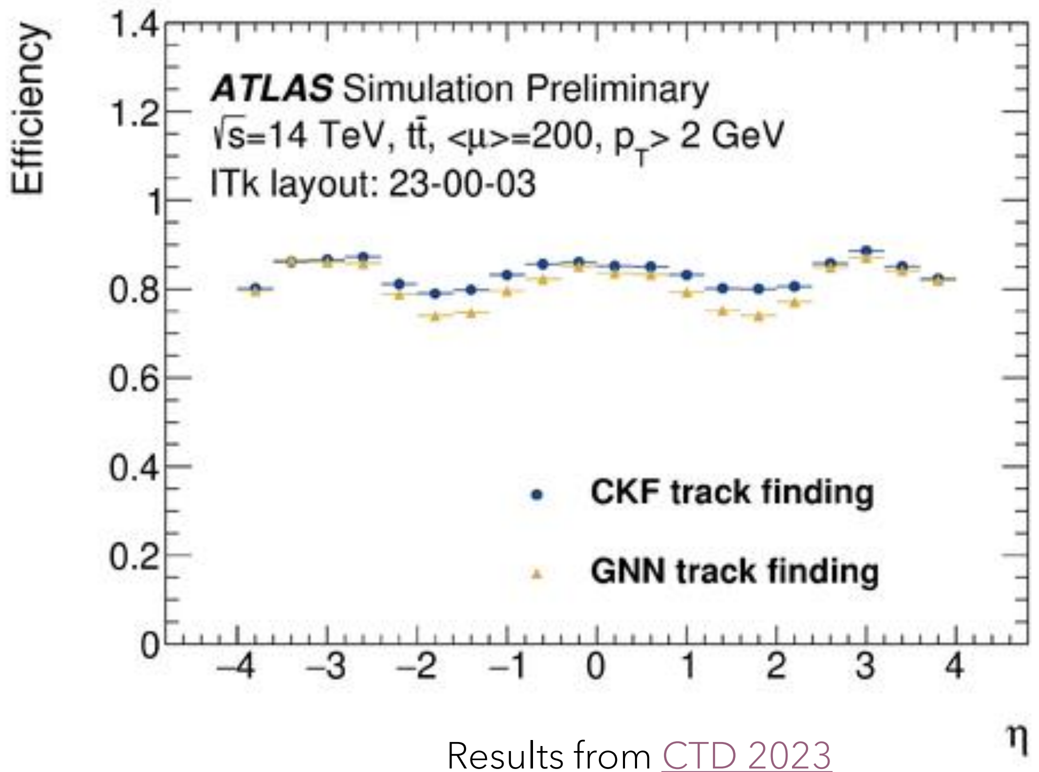
- **Combinatorial Kalman Filter (CKF)**
  - Traditional method
  - High efficiency, low fake rate
  - Scales worse than linear in event size
  - 'CPU intensive'
    - ATLAS TDAQ Tracking Amendment
- **Graph-based machine learning**
  - Comparable efficiency and fake rates to CKF
  - Scaling that is close to linear in event size
    - As shown by Exa.TrkX arXiv:2103.06995
  - Benefits from GPU acceleration



Results from CTD 2023

Track Finding

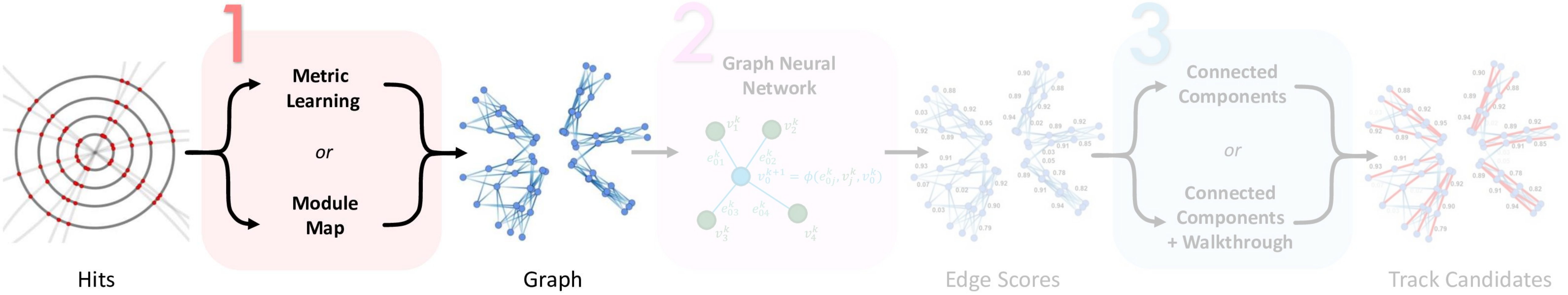
- **Combinatorial Kalman Filter (CKF)**
  - Traditional method
  - High efficiency, low fake rate
  - Scales worse than linear in event size
  - ‘CPU intensive’
    - ATLAS TDAQ Tracking Amendment
- **Graph-based machine learning**
  - Comparable efficiency and fake rates to CKF
  - Scaling that is close to linear in event size
    - As shown by Exa.TrkX arXiv:2103.06995
  - Benefits from GPU acceleration



Results from CTD 2023

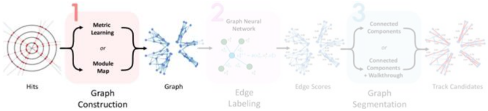
https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/IDTR-2022-01/

How?



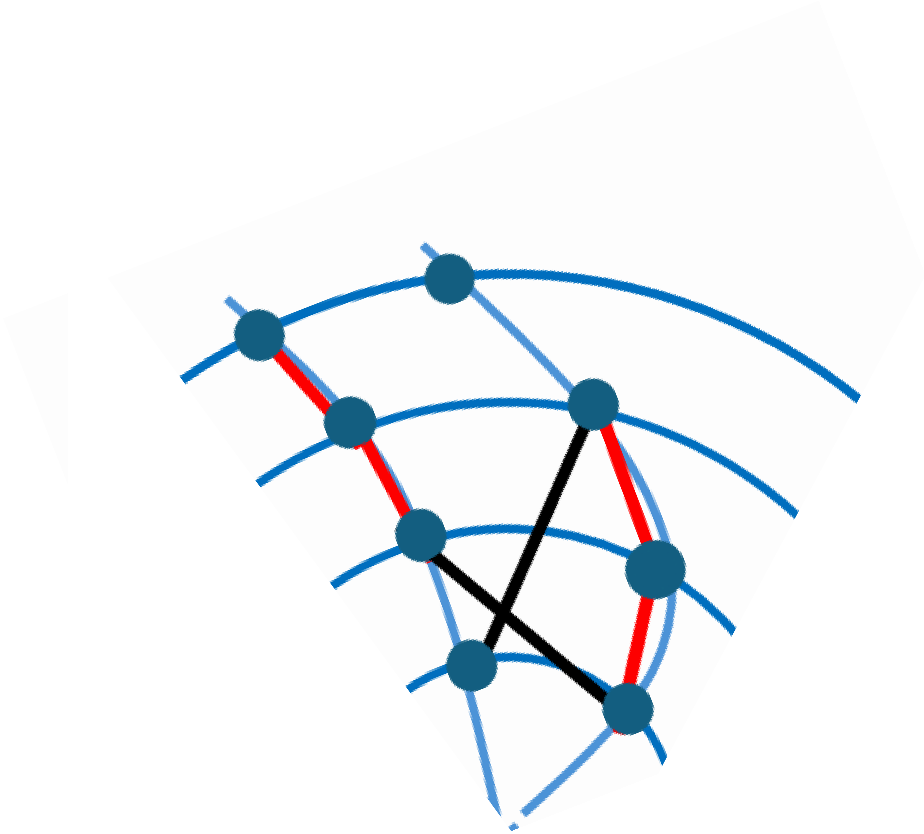
1. GRAPH CONSTRUCTION

Assign hits as nodes; connect these nodes with edges to allow connecting nodes possibility of belonging to the same particle



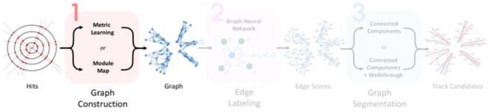
Metric Learning

- Embed hits into a latent space using an MLP
- Connect hits within some radius in latent space



GOAL:

- **Build true edges (efficiency)**
- **Limit false edges (purity)**

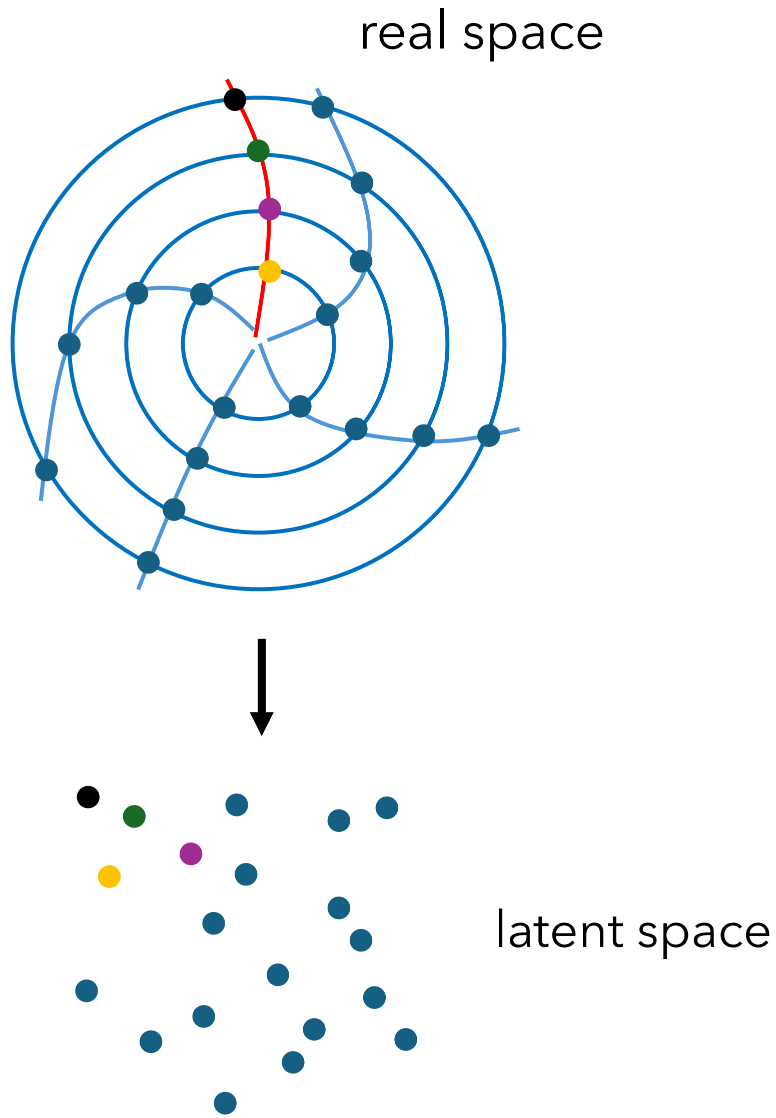


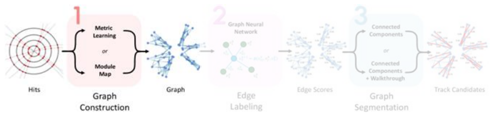
Metric Learning

- Embed hits into a latent space using an MLP
- Connect hits within some radius in latent space

GOAL:

- **Build true edges (efficiency)**
- **Limit false edges (purity)**

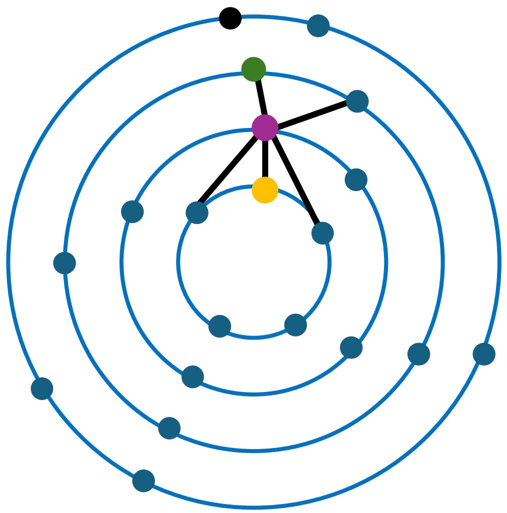




Metric Learning

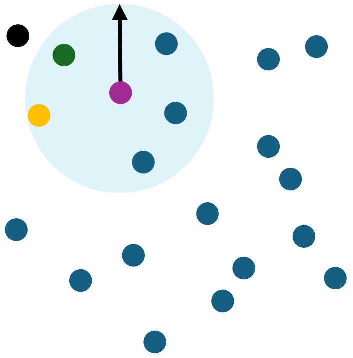
- Embed hits into a latent space using an MLP
- Connect hits within some radius in latent space

real space

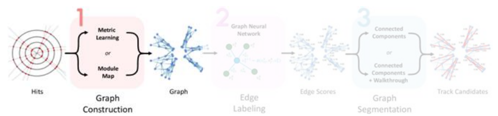


GOAL:

- **Build true edges (efficiency)**
- **Limit false edges (purity)**



latent space



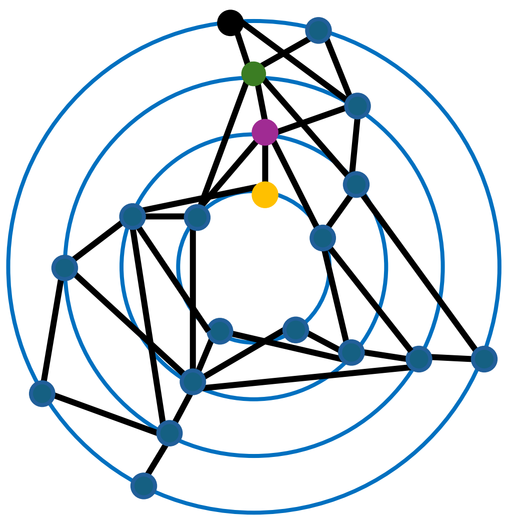
Metric Learning

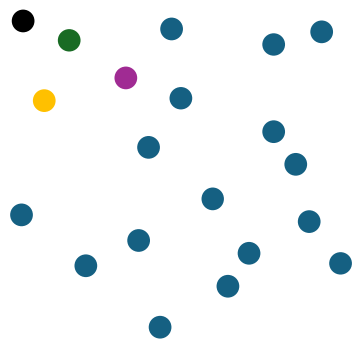
- Embed hits into a latent space using an MLP
- Connect hits within some radius in latent space

GOAL:

- **Build true edges (efficiency)**
- **Limit false edges (purity)**

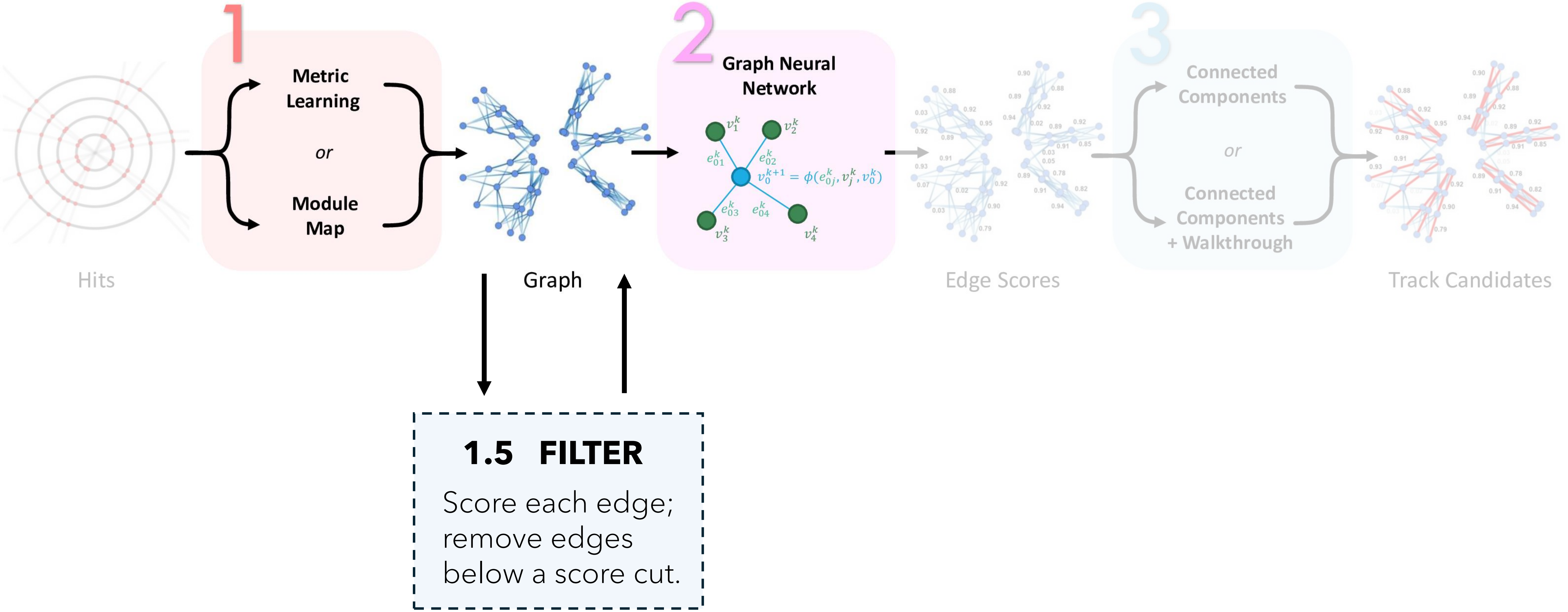
real space





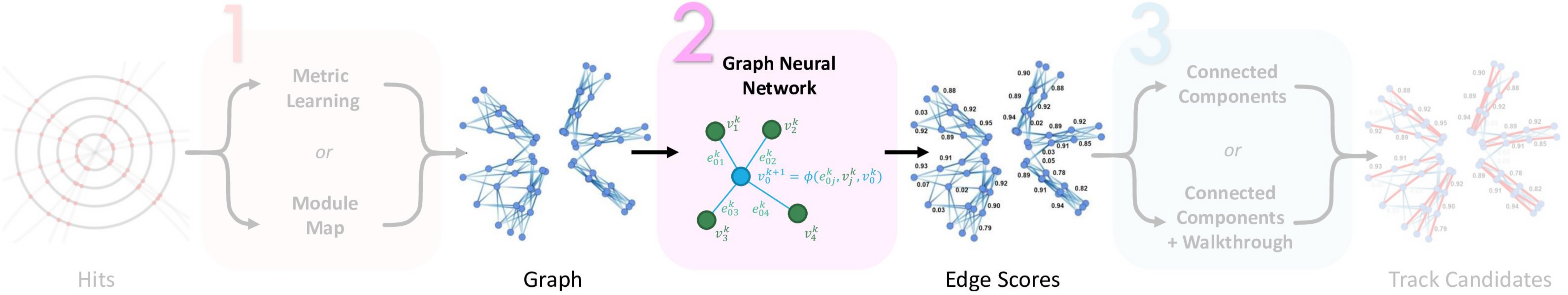
latent space

How?



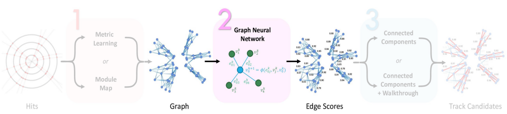
1
Metric Learning
or
Module Map
Hits
Graph
2
Graph Neural Network
$v_0^{k+1} = \phi(e_{0j}^k, v_j^k, v_0^k)$
Edge Scores
3
Connected Components
or
Connected Components + Walkthrough
Track Candidates
1.5 FILTER
Score each edge; remove edges below a score cut.

How?



2. EDGE CLASSIFICATION

Feed graph into a trained interaction network that will assign a score to each edge, quantifying the probability connected nodes belong to same particle.



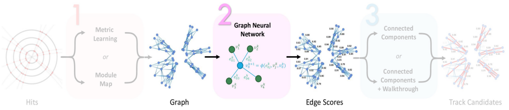
The Interaction Network

Input graph → **Encoders** → **Interaction Network** → **Decoders** → **Edge scores**

Encoders: Embeds graph input features into latent space

Interaction Network: Learns geometric patterns of beyond-nearest nodes using message passing steps

Decoders: Transforms latent features into edge score



The Interaction Network:
Message Passing

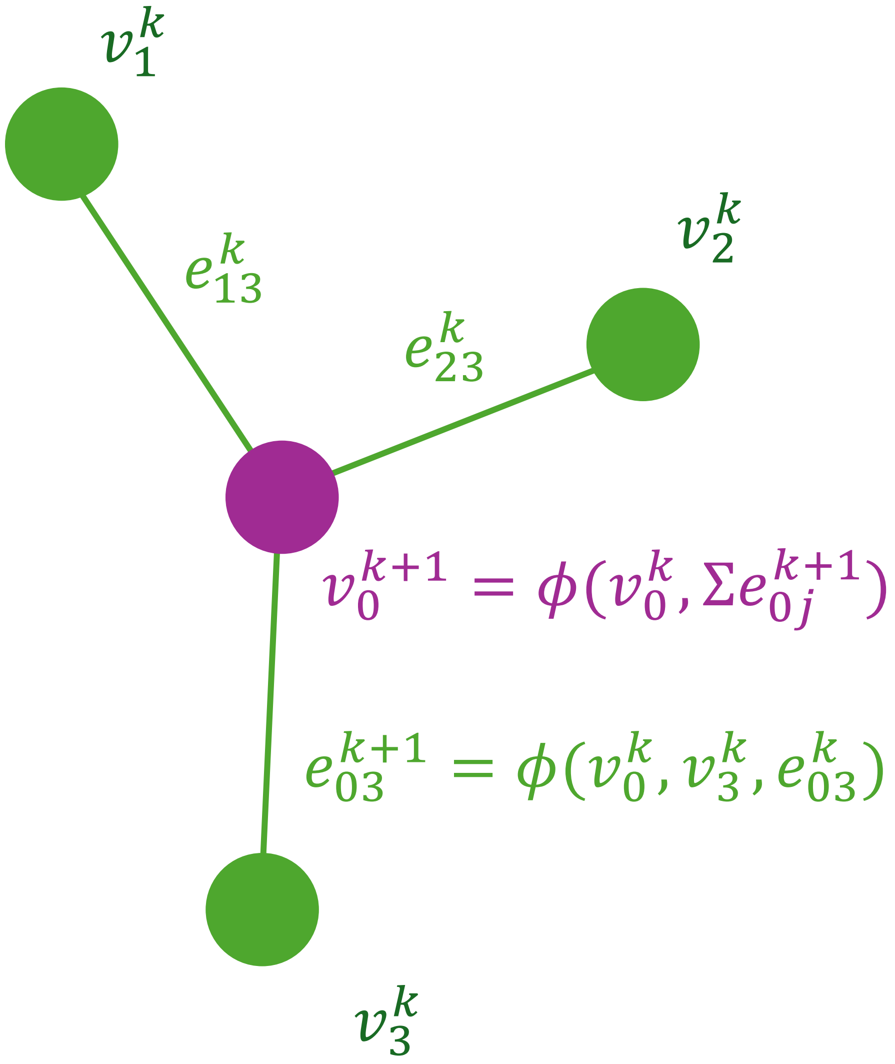
On the k^{th} iteration

Input graph → **Encoders** → **Interaction Network** → **Decoders** → **Edge scores**

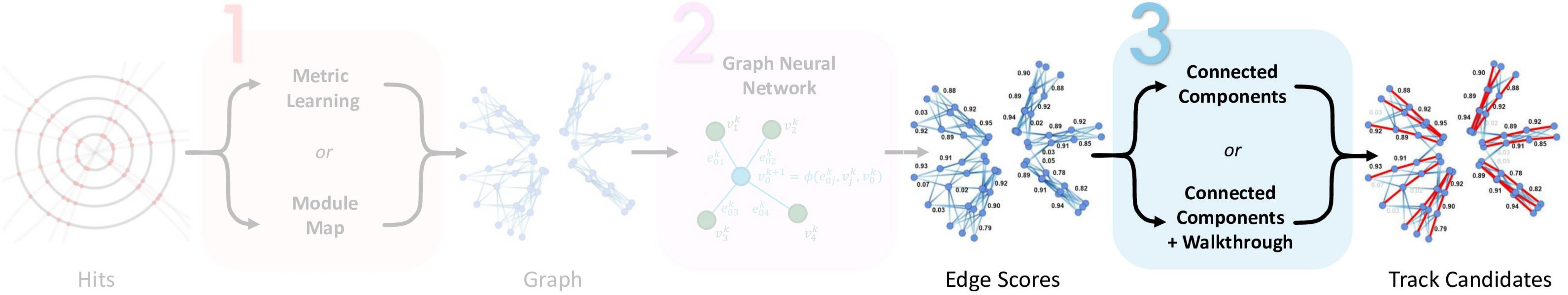
- **Encoders**: Embeds graph input features into latent space
- **Interaction Network**: Learns geometric patterns of beyond-nearest nodes using message passing steps
- **Decoders**: Transforms latent features into edge score

One iteration:

1. Node features (spatial position) are encoded
2. Encoded features concatenated and encoded to create edge features
3. Edge features are aggregated to create next round of encoded node features



How?



3. GRAPH SEGMENTATION

Use an algorithm to connect scored edges into track candidates.

Baseline

Dataset: MC event simulation samples
- pp collisions at $\sqrt{s} = 14\ \text{TeV}$ with a $t\bar{t}$ pair in the final state
- Average 200 pp pileup per bunch crossing
- Full ATLAS detector simulation based on GEANT 4

Target particles:
$p_T > 1\ GeV;\ |\eta| < 4;\ n_{hits} > 3$

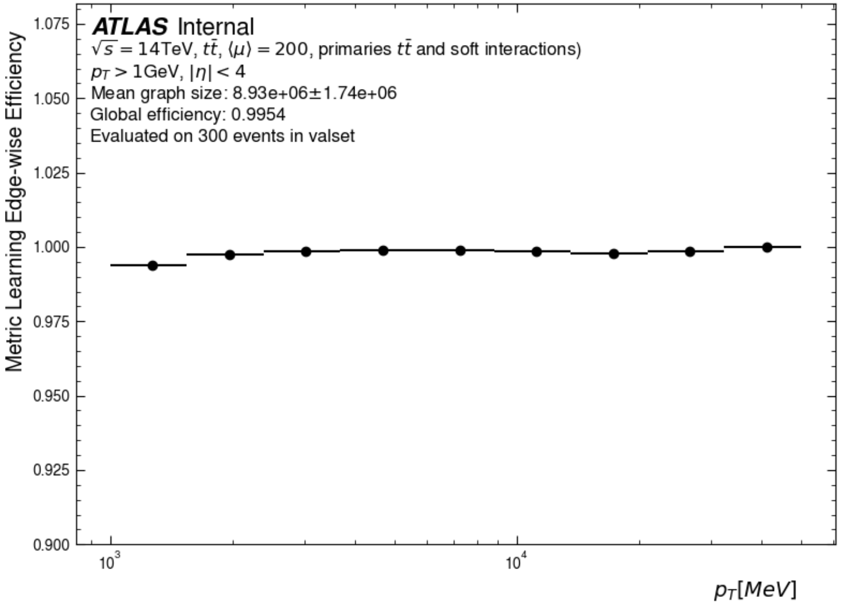
Track efficiency:

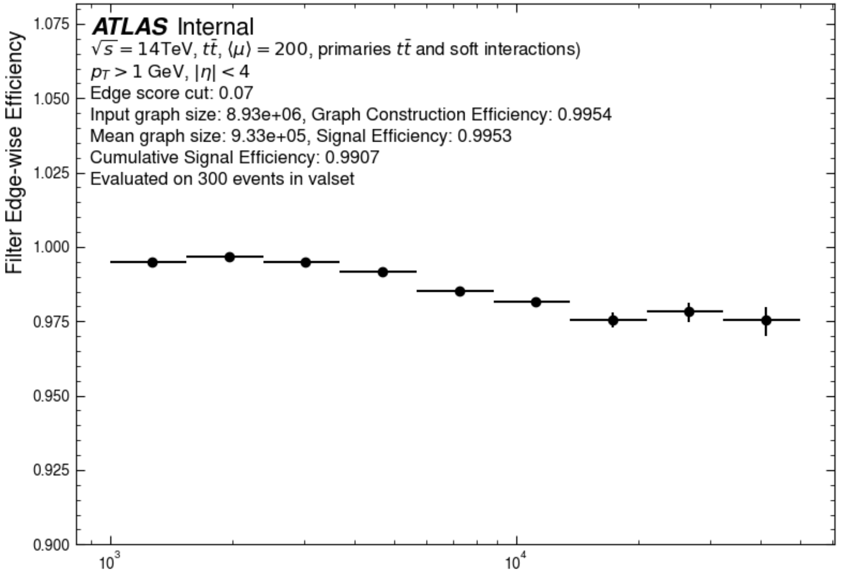
0.985

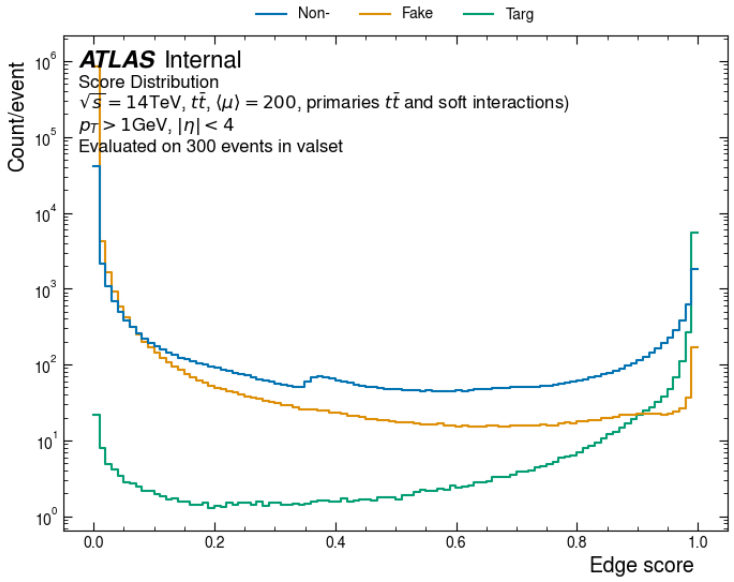
Input graph size:

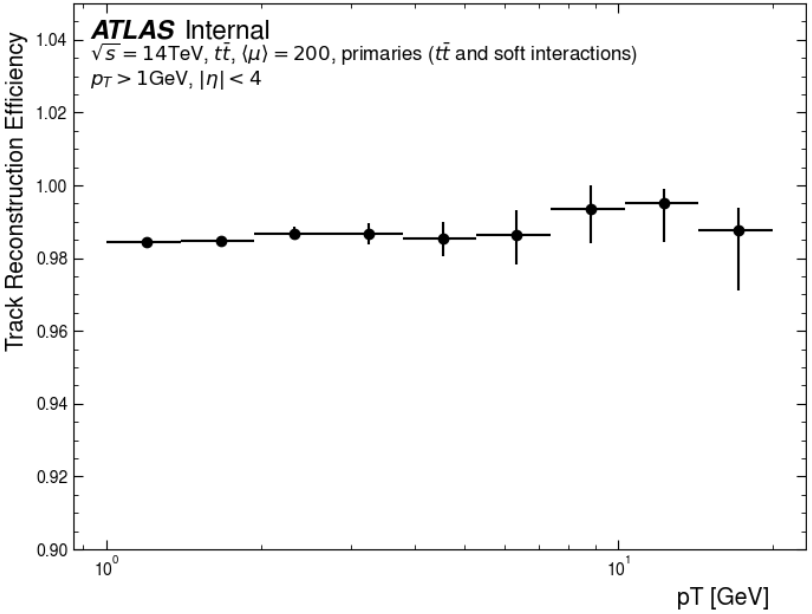
~ 9×10^6 edges

=> 7500 edges / target track



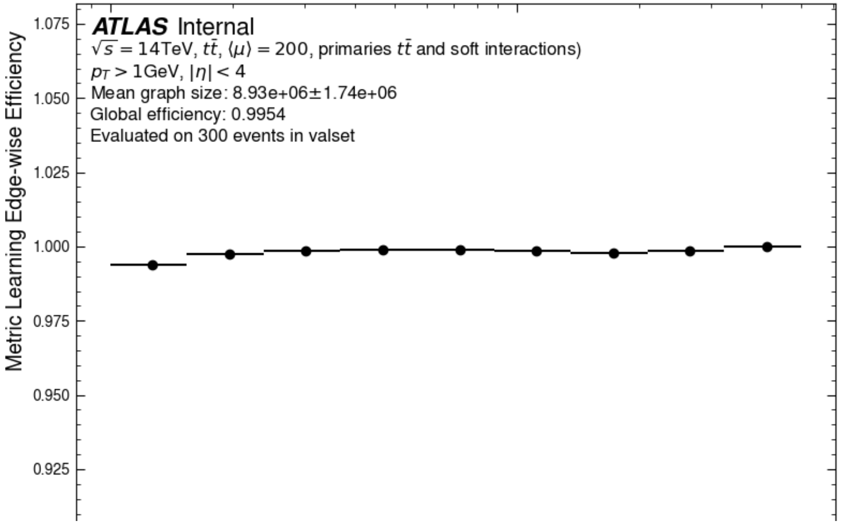


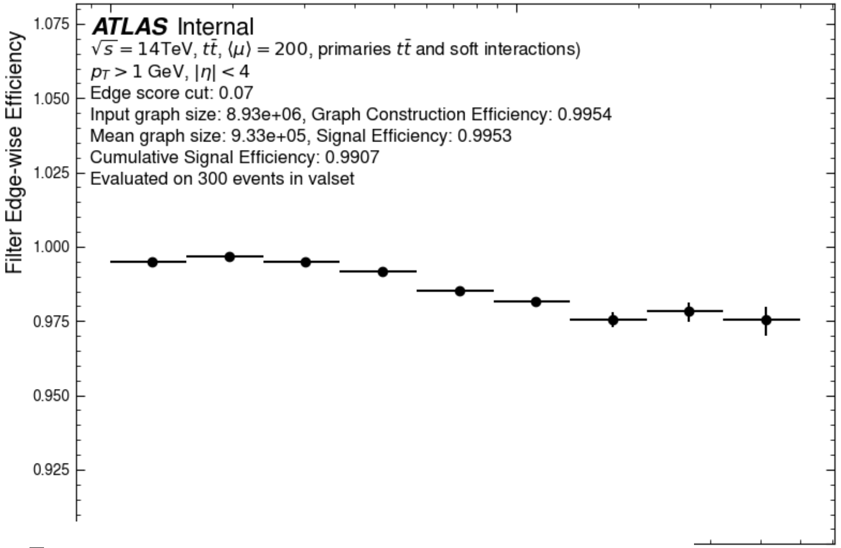




Baseline

Dataset: MC event simulation samples
- pp collisions at $\sqrt{s} = 14$ TeV with a $t\bar{t}$ pair in the final state
- Average 200 pp pileup per bunch (
- Full ATL
  based (





Can we build smaller graphs whilst maintaining track efficiency?

Target particles:
$p_T > 1\ GeV;\ |\eta| < 4; n_{hits} > 3$

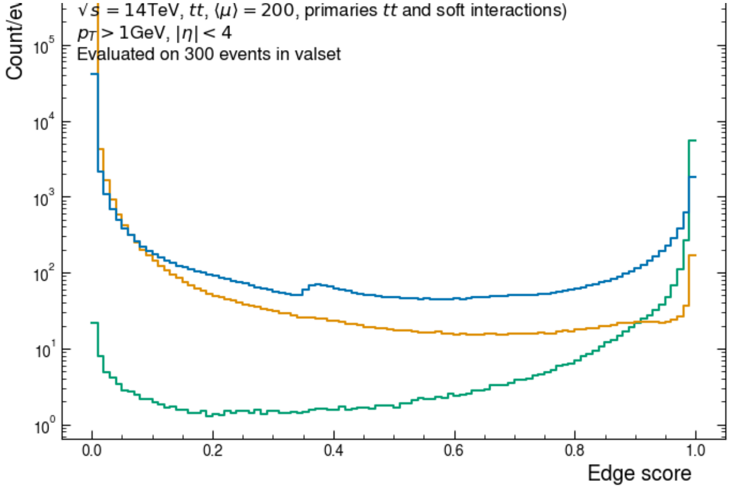
Track efficiency:

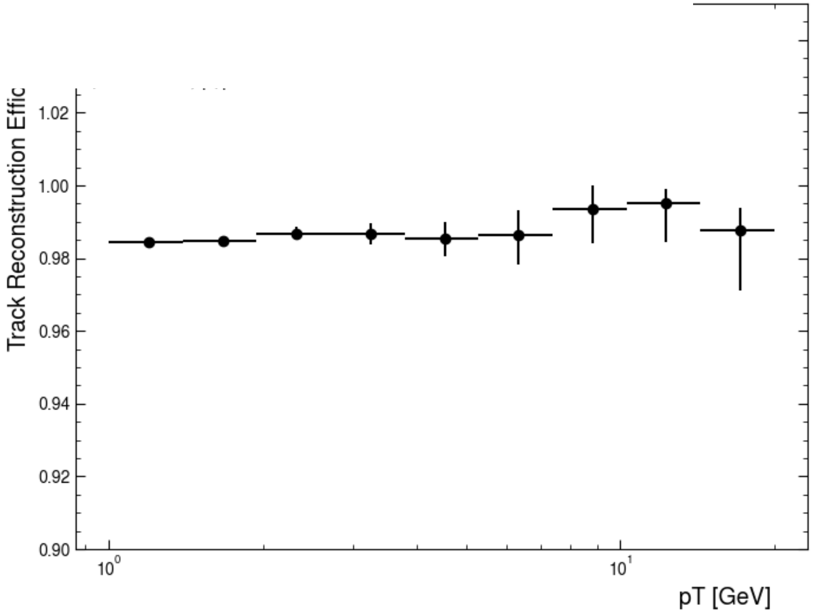
0.985

Input graph size:

~ 9×10^6 edges

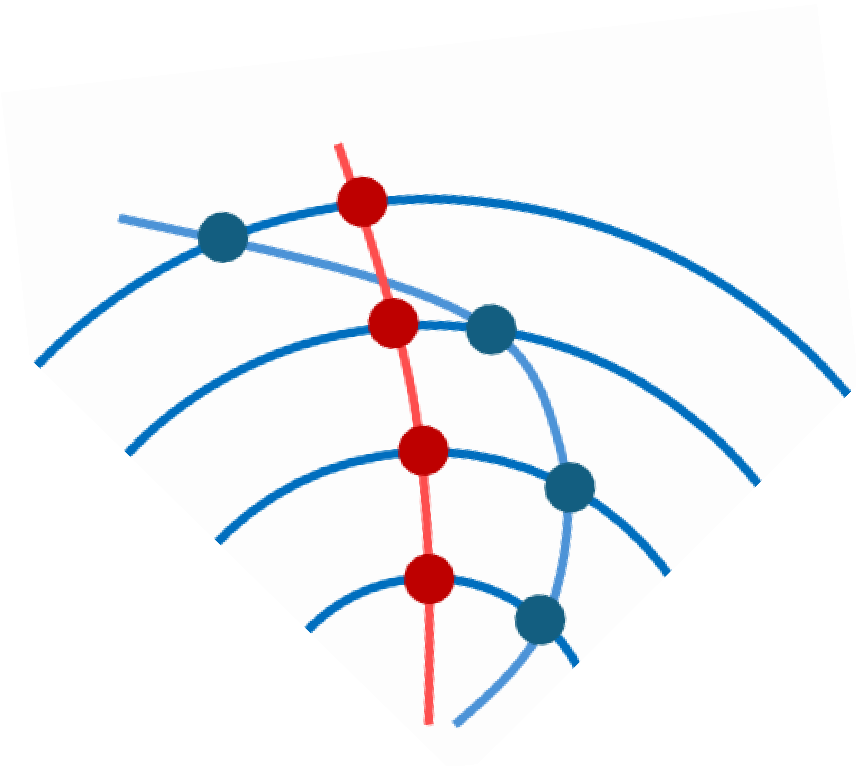
=> 7500 edges / target track





Physics-Informed Graph Optimization

- High momentum and low momentum tracks are easy to differentiate
- Is it possible to train networks to construct only high (or low) momentum tracks?
- Would these specialised networks build smaller graphs?



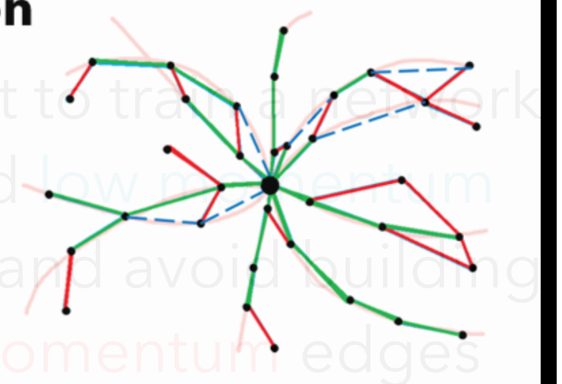
Building only low p_T tracks - **NO**

Evaluation terms

true

false

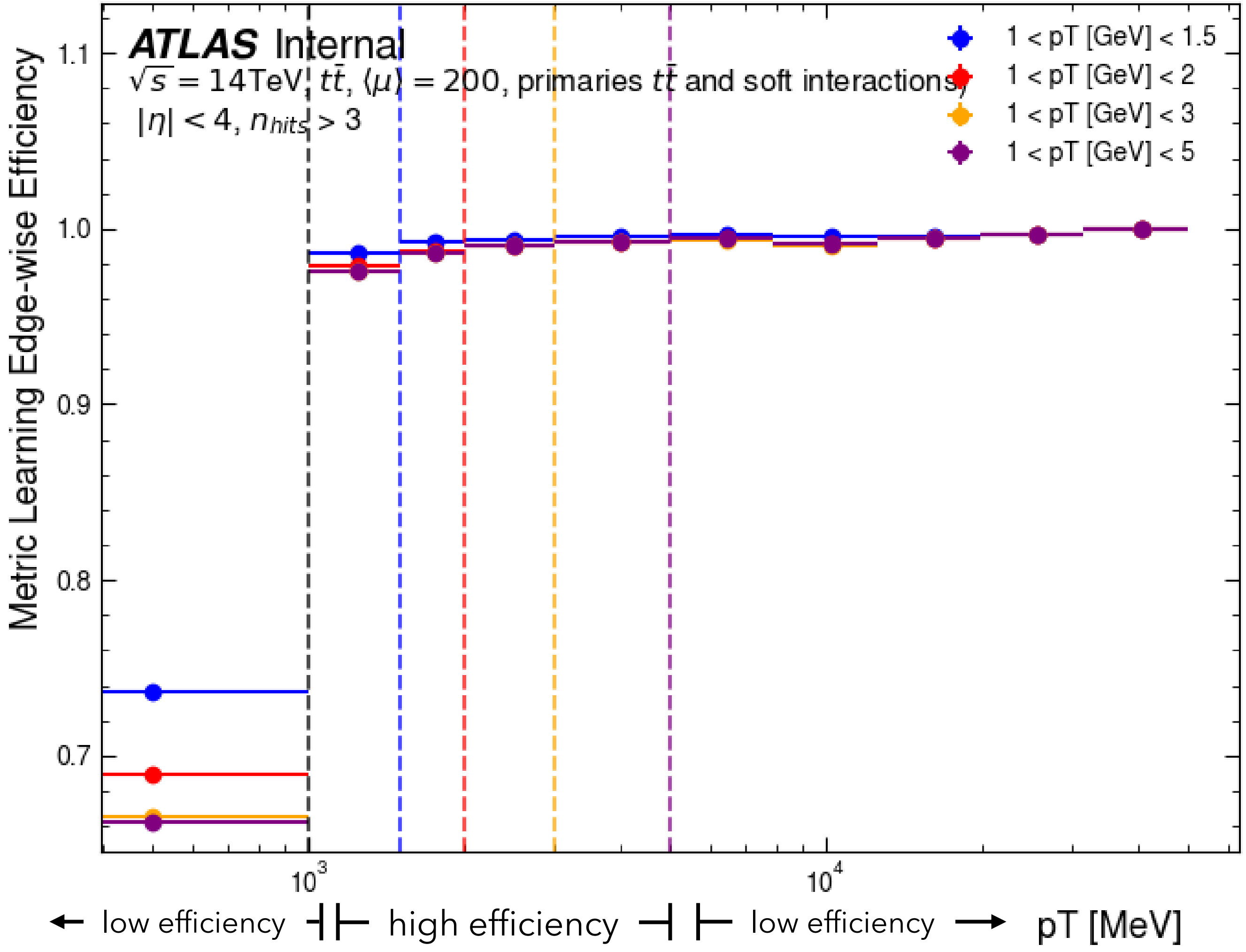
missed



$$\text{efficiency} = \frac{\text{true}}{\text{true} + \text{missed}}$$

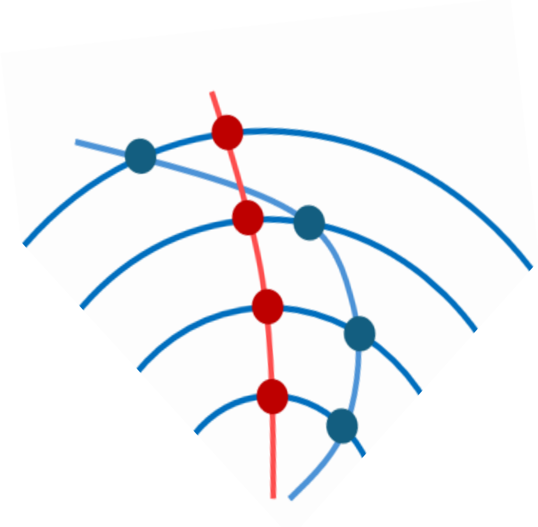
$$\text{purity} = \frac{\text{true}}{\text{true} + \text{false}}$$

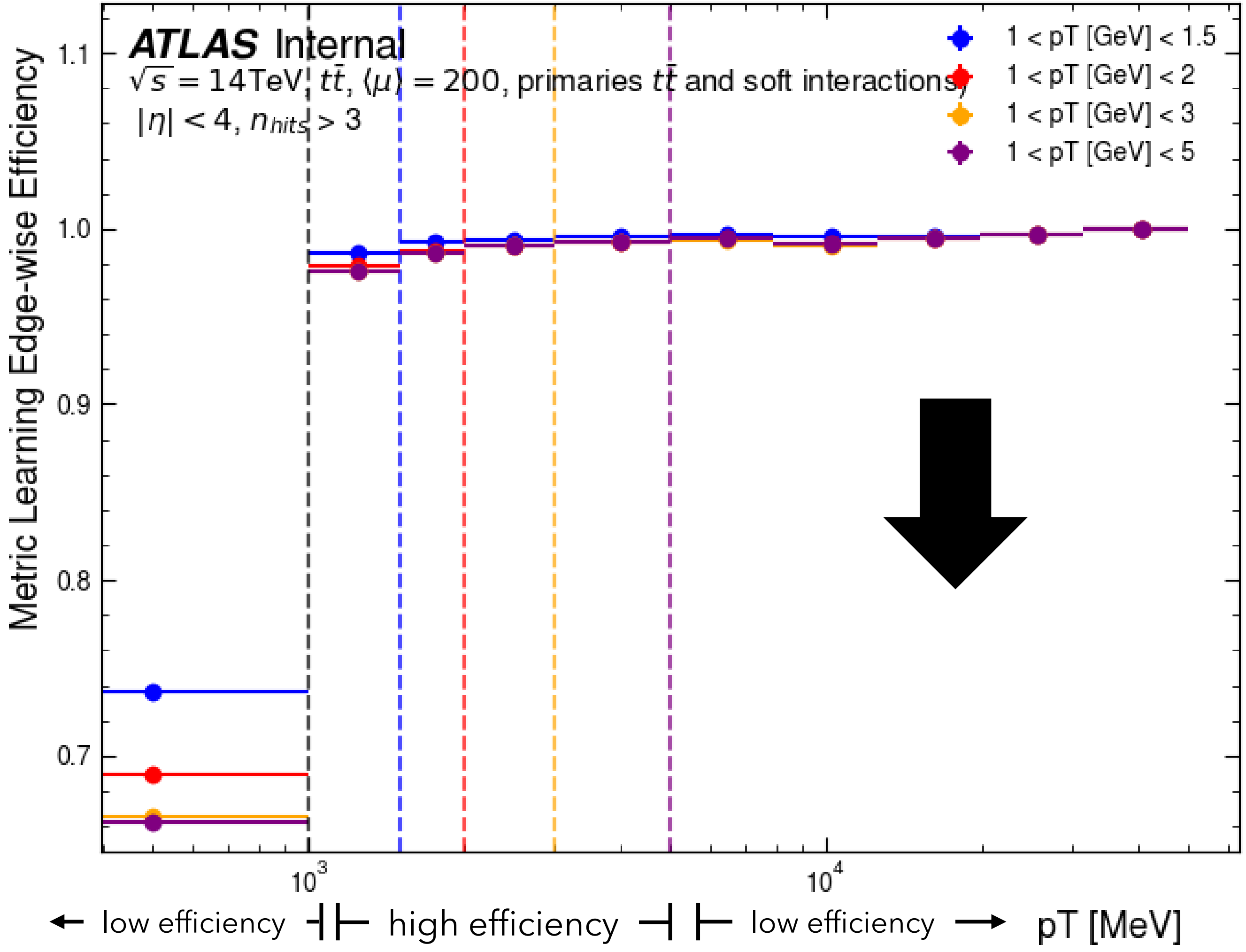
$$\text{graph size} = \text{true} + \text{false}$$



Building only low p_T tracks - **NO**

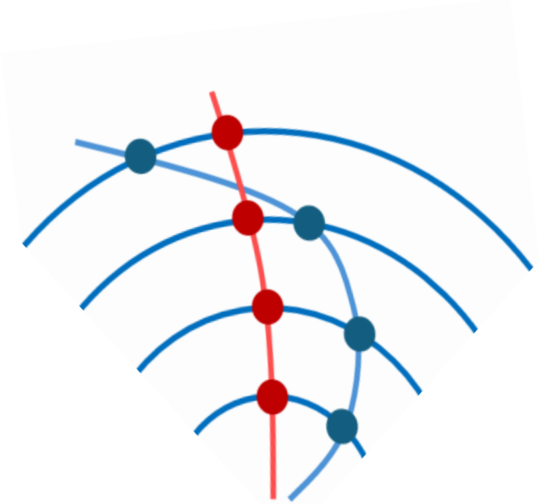
- Difficult to train a network to build low momentum edges and avoid building high momentum edges

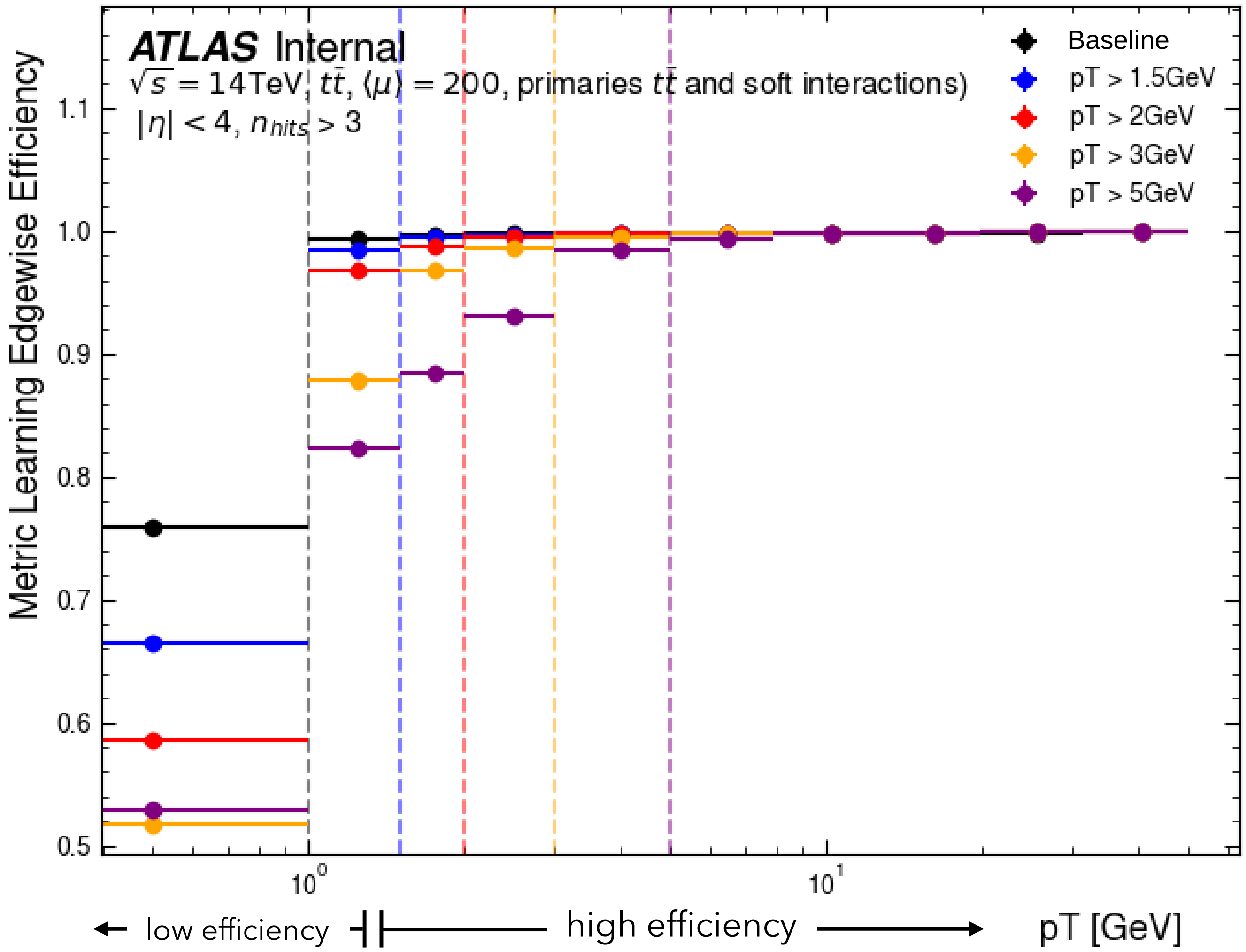


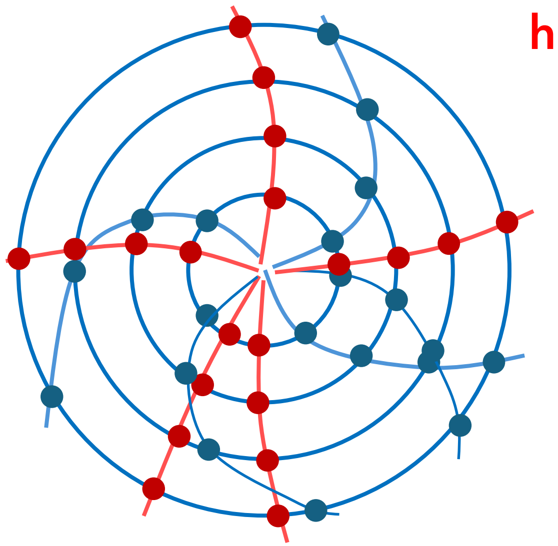


Building only high p_T tracks - **YES**

- It is possible to train a network to build high momentum edges whilst avoid building low momentum edges!





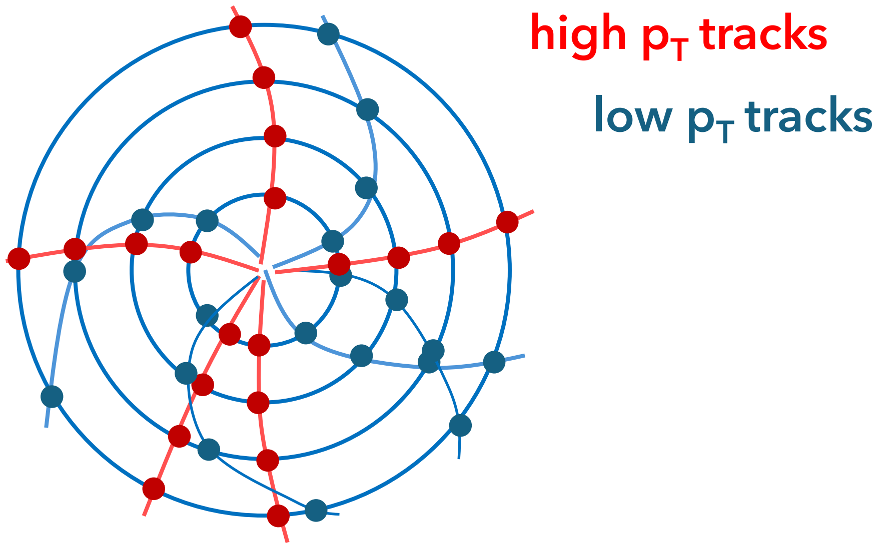


STAGE ONE

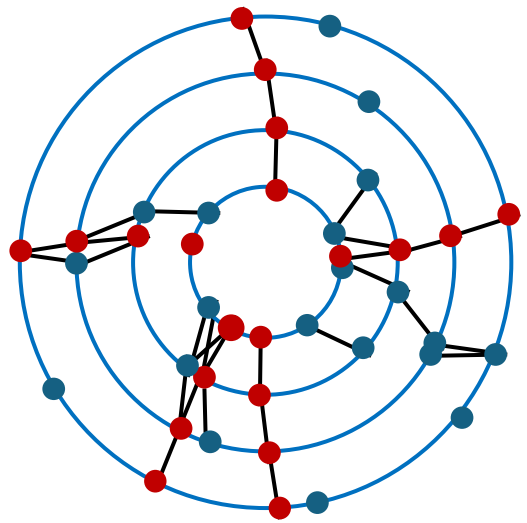
- Build graphs aimed at $p_T > [1.5, 2, 3, 5]$ GeV
- Construct tracks
- Remove hits associated with constructed tracks

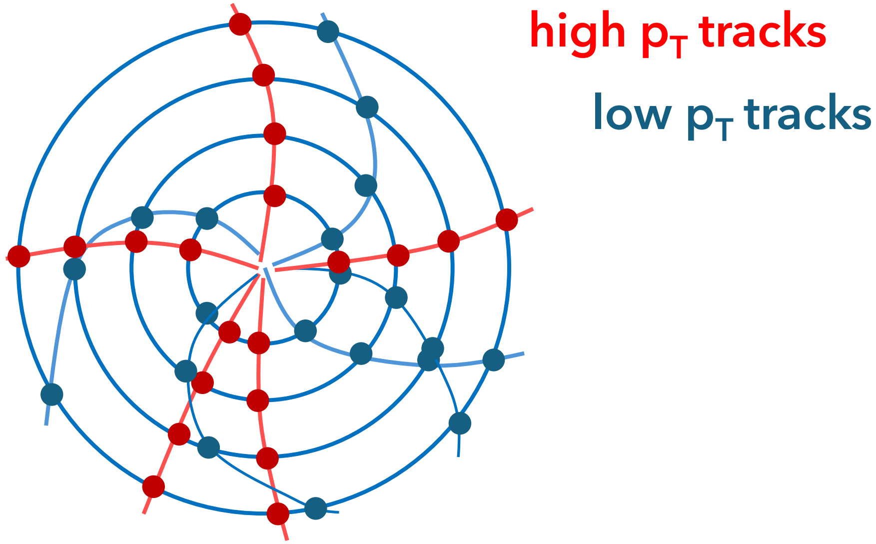
STAGE TWO

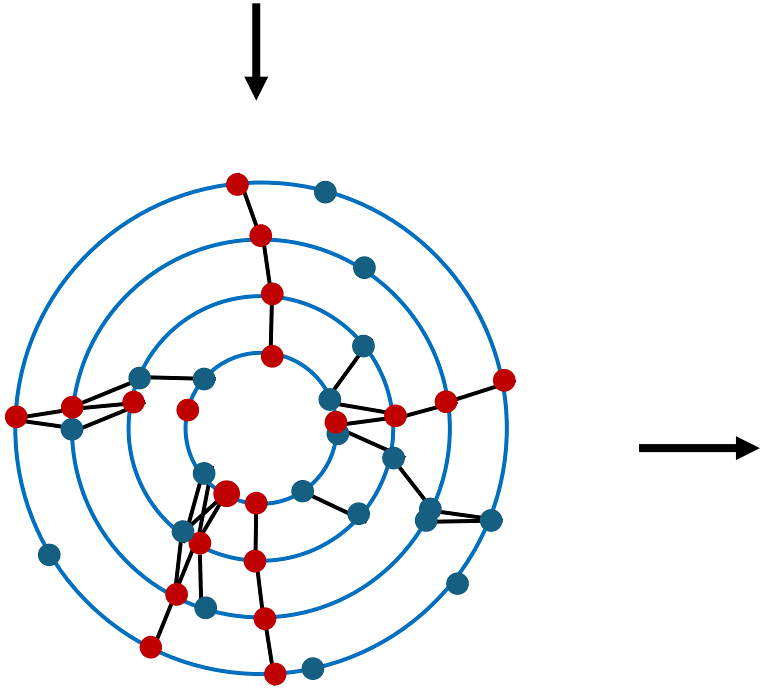
- Build graphs aimed at $1 < p_T\,[\text{GeV}] < [1.5, 2, 3, 5]$ using reduced dataset
- Construct tracks



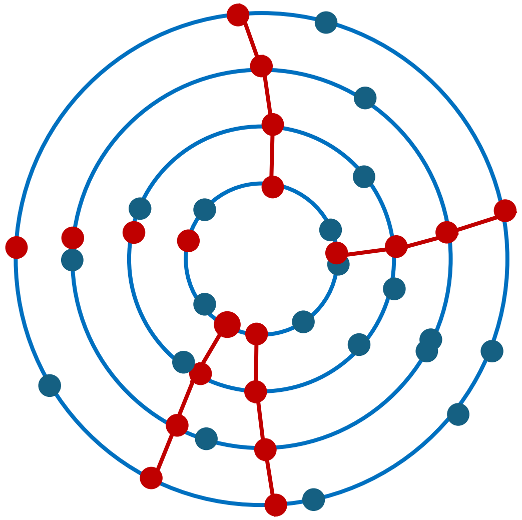
1. Build high p_T edges



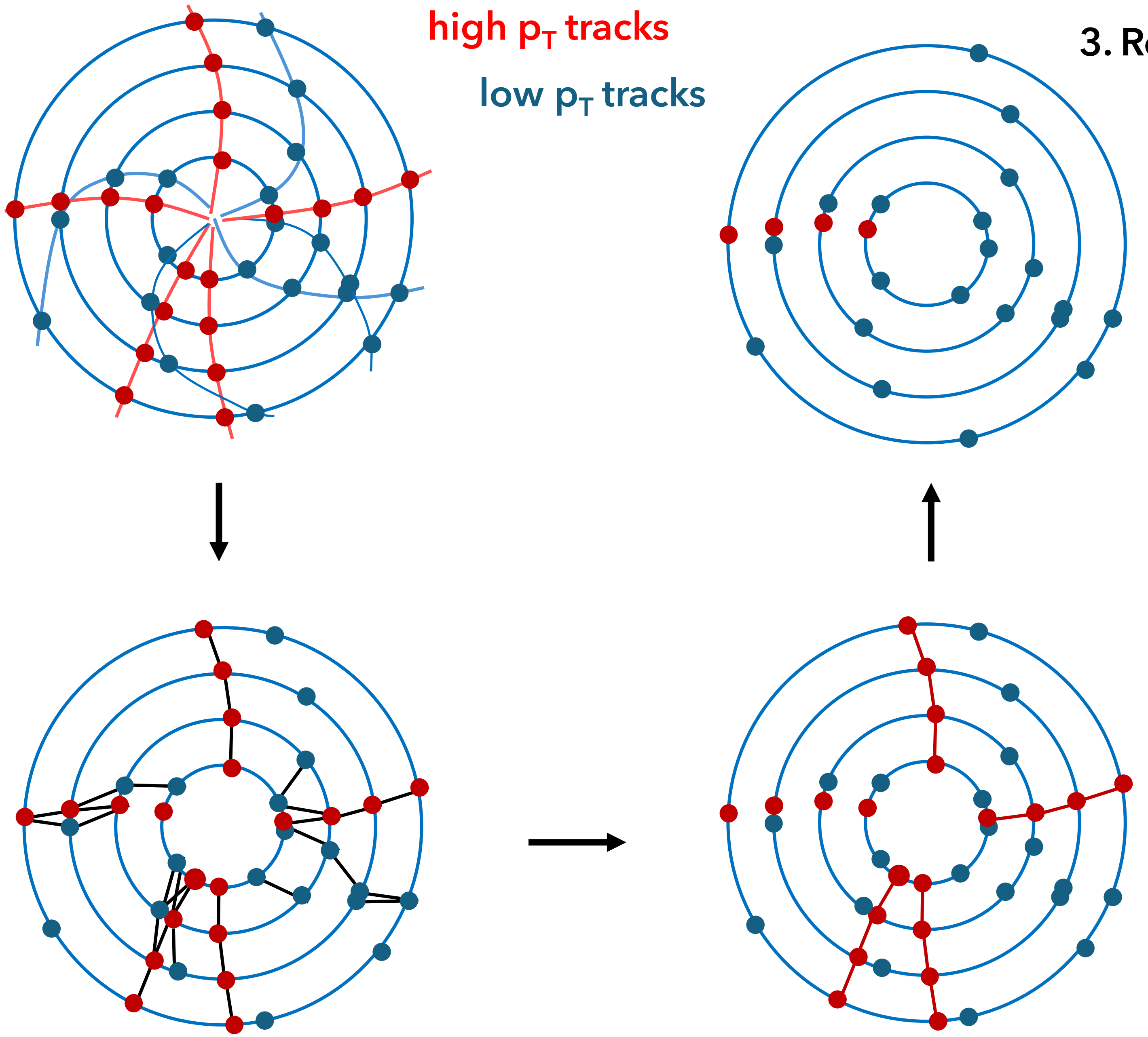


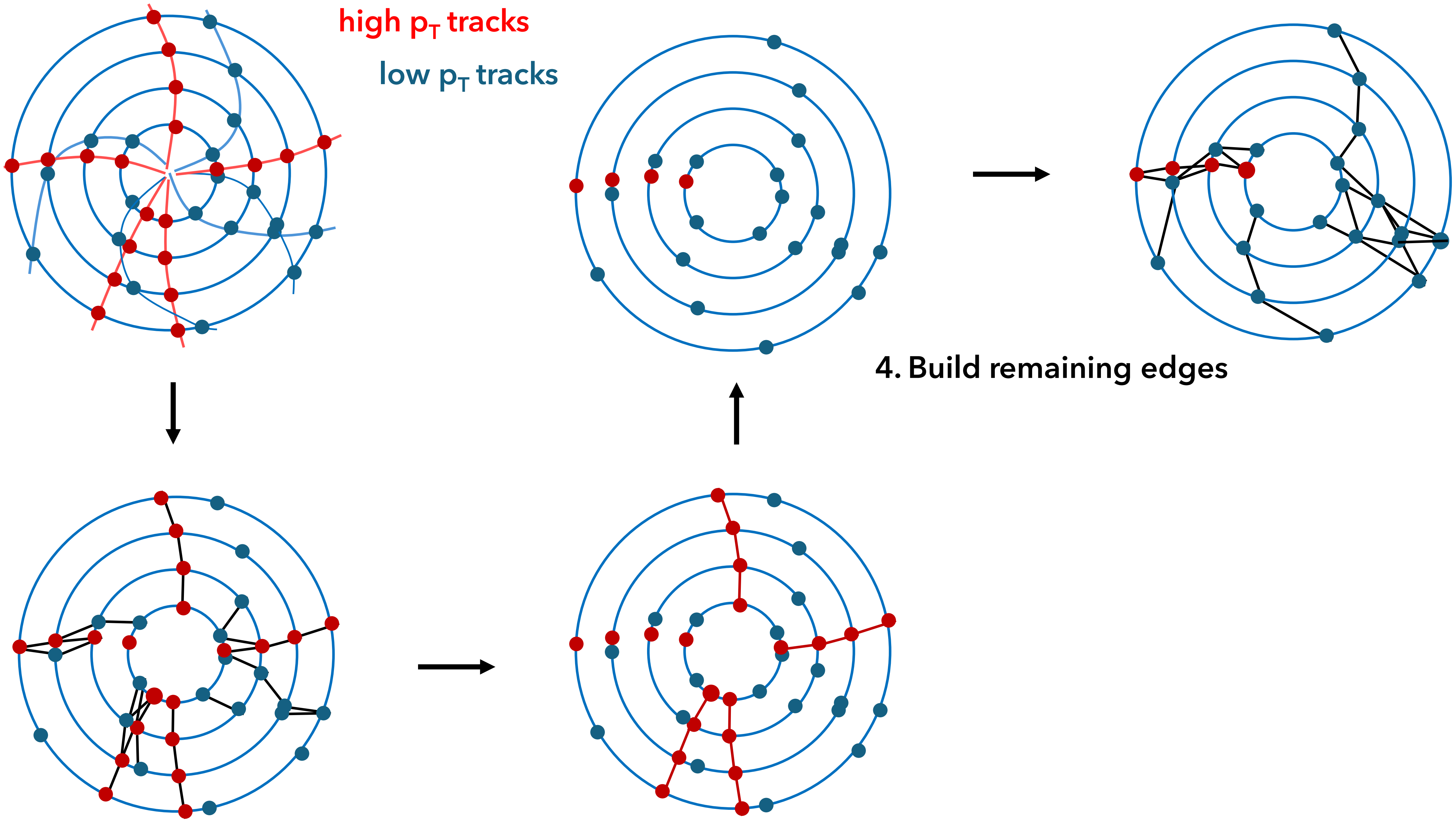


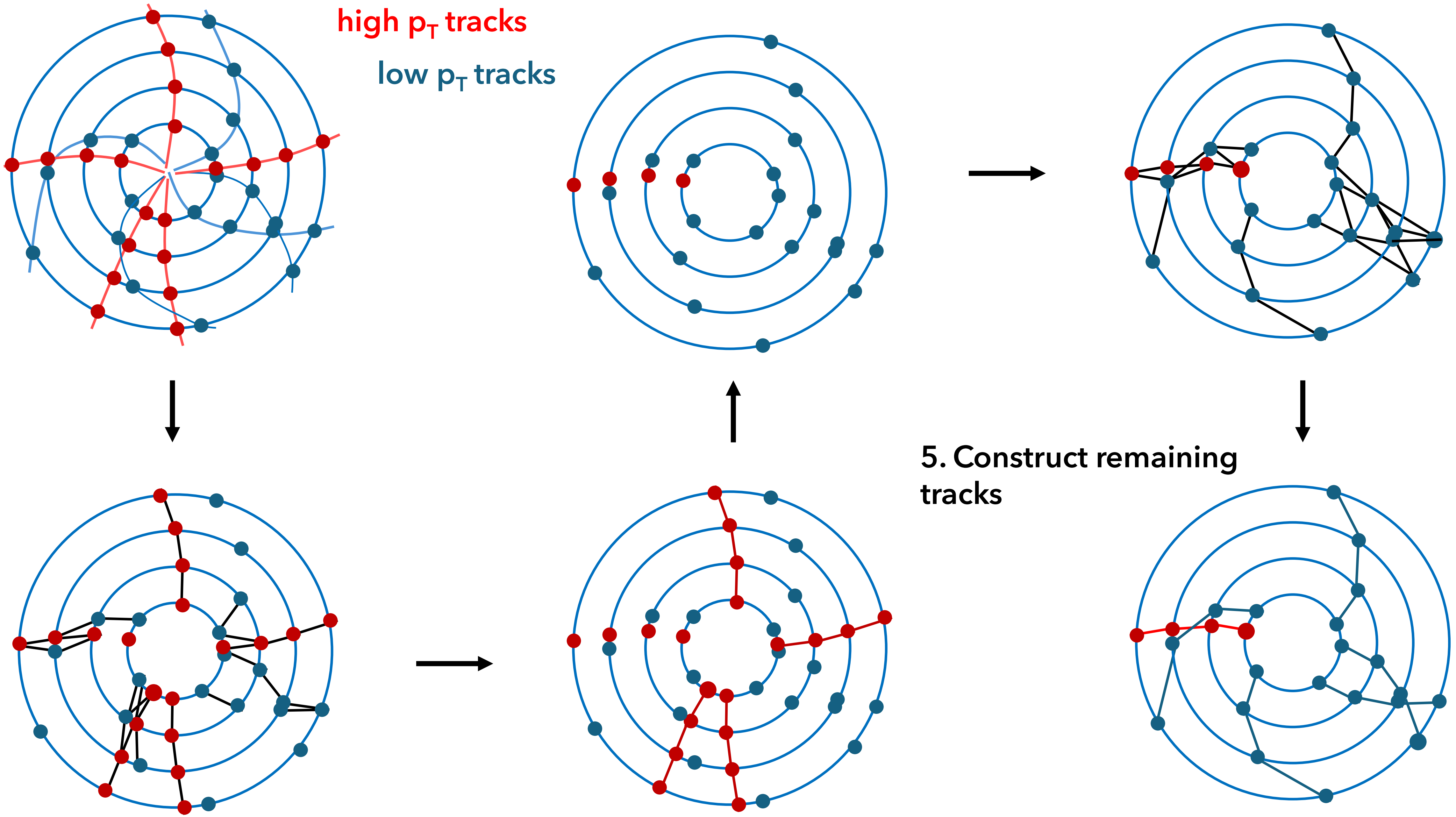
2. Construct tracks



3. Remove associated hits




high p_T tracks
low p_T tracks
4. Build remaining edges

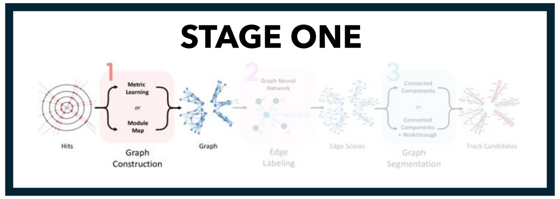

high p_T tracks
low p_T tracks
5. Construct remaining tracks

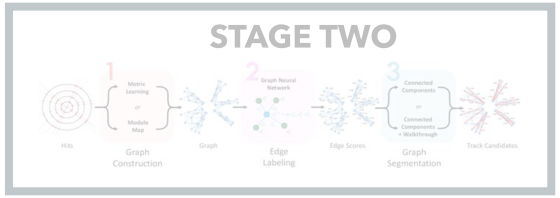
Stage One:

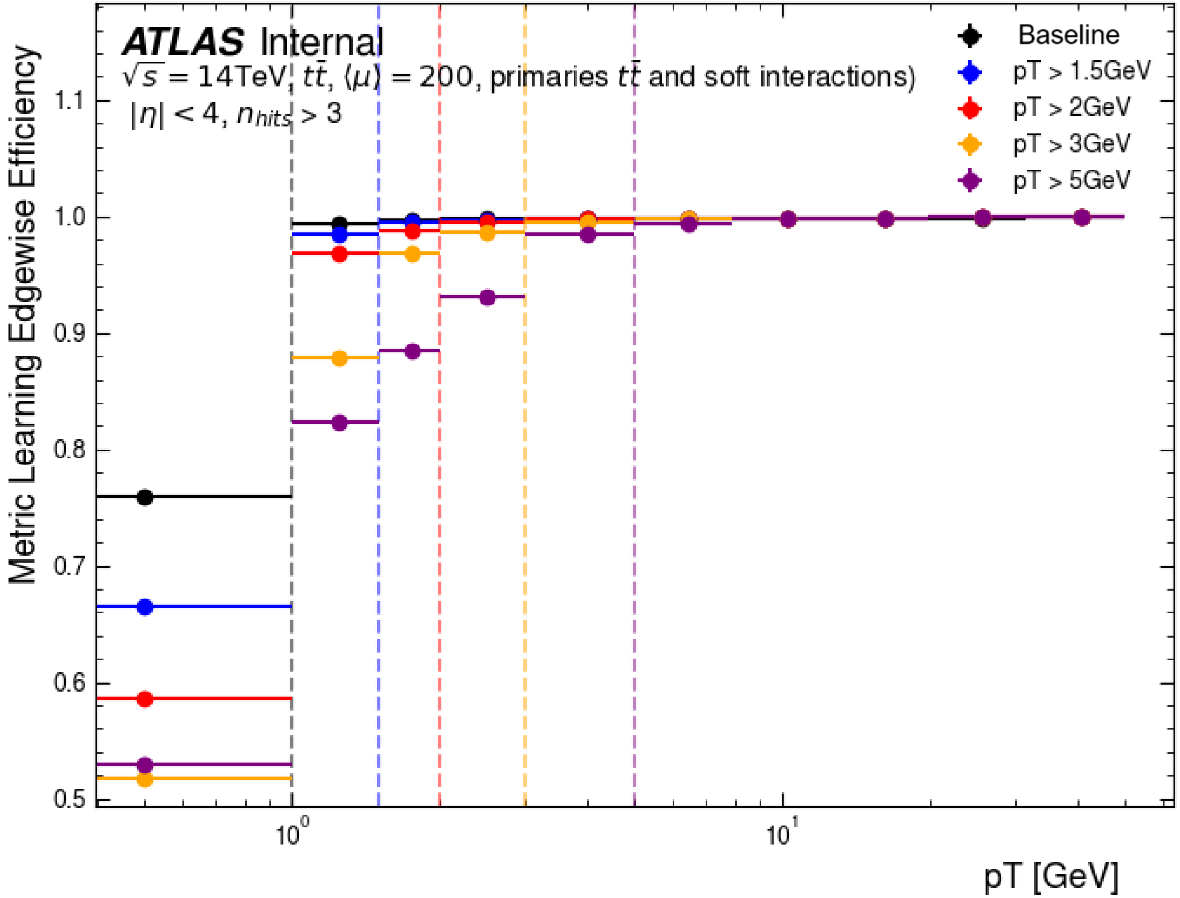
Graph Construction

Build graphs aimed at $p_T > [1, 1.5, 2, 3, 5]$ GeV

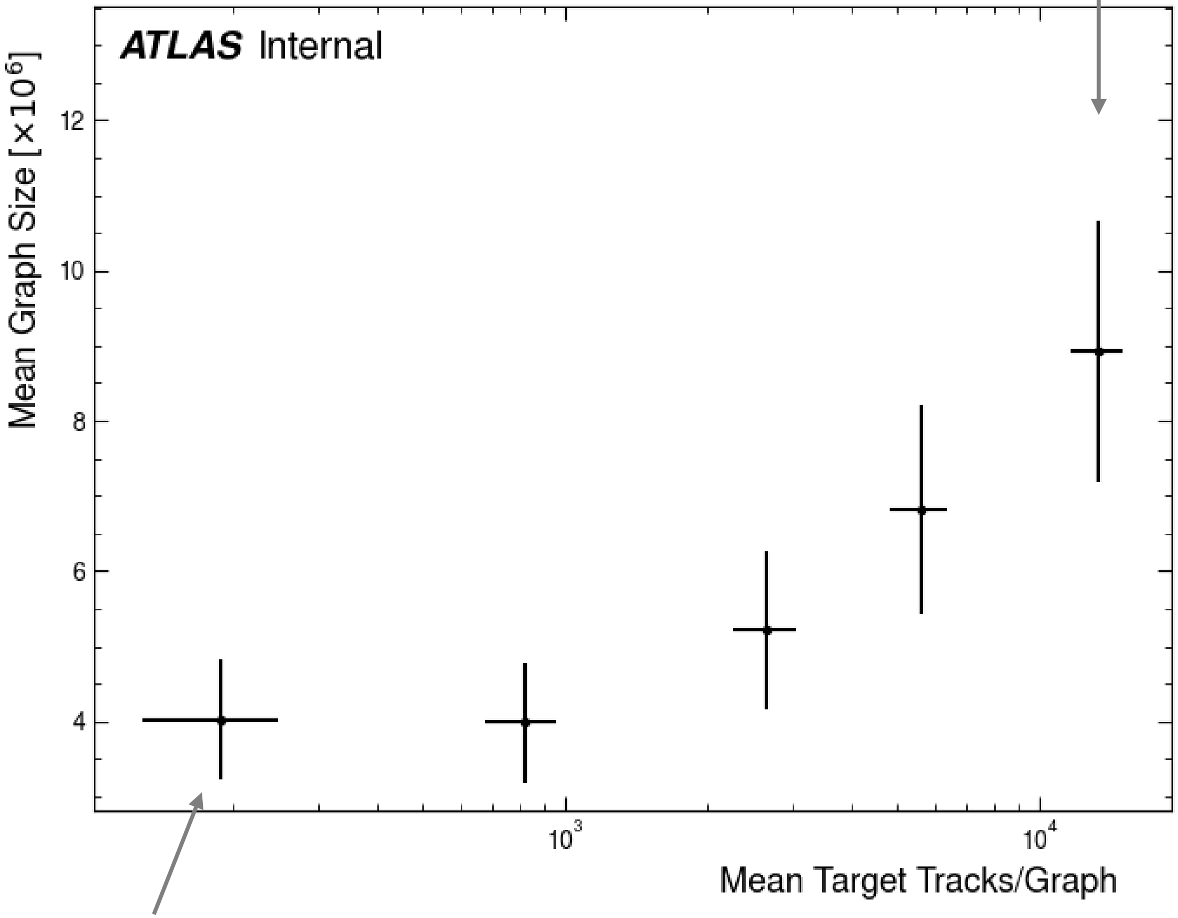
AIM: Maximising target purity whilst requiring 99.5% target efficiency







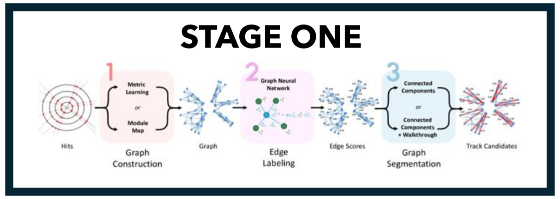
Baseline, pT > 1 GeV

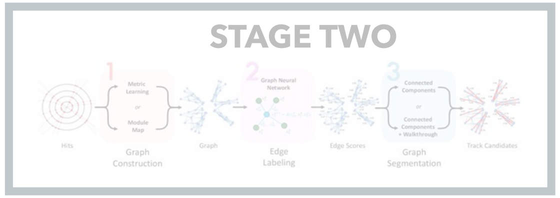


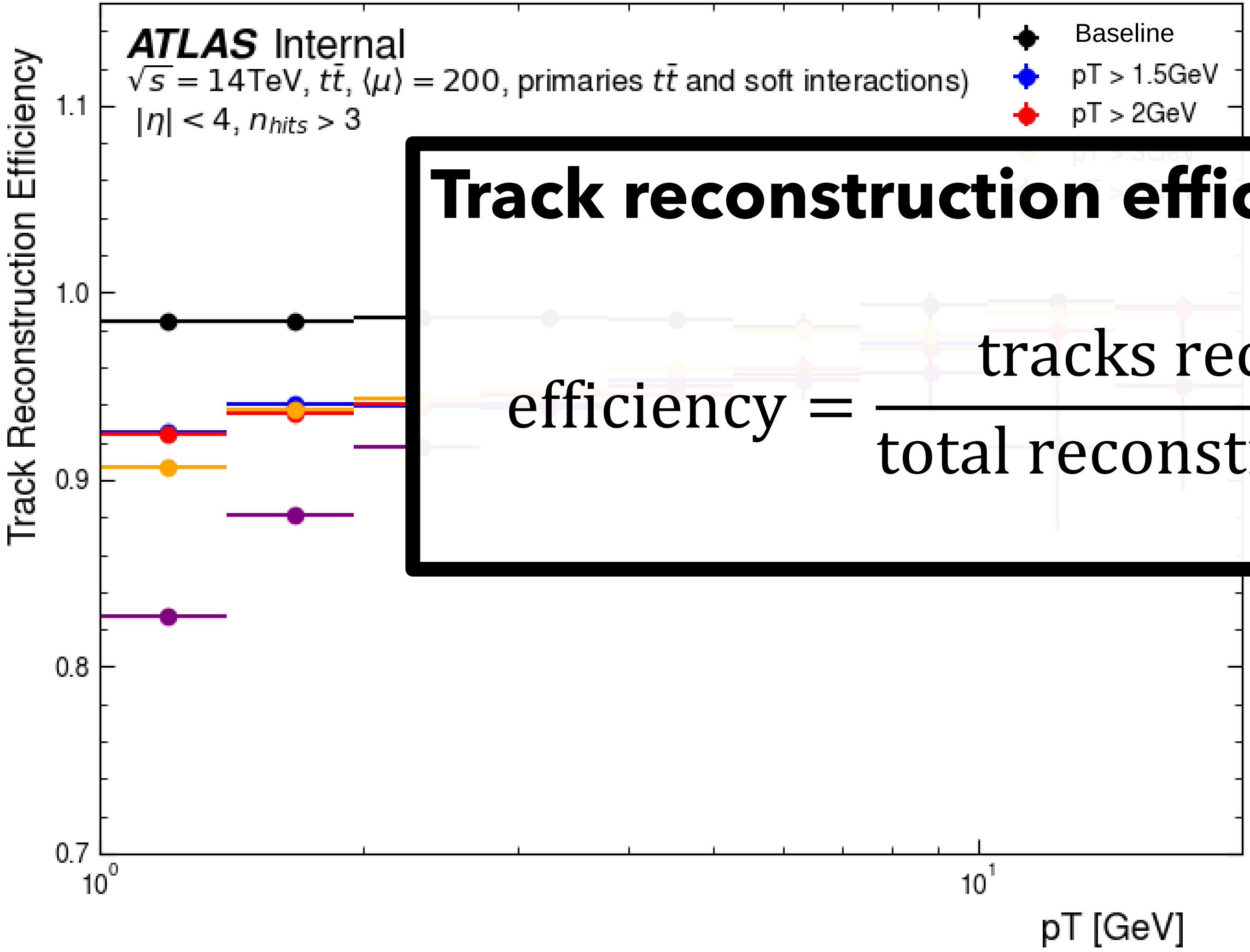
pT > 5 GeV

Stage One:

Track Reconstruction



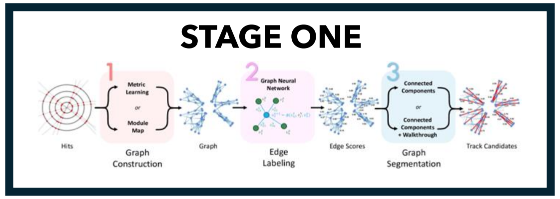


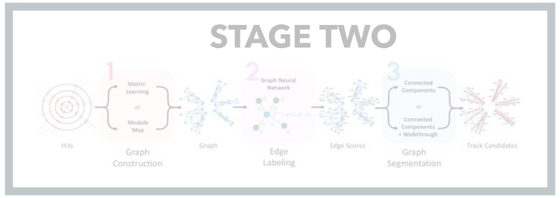


Track reconstruction efficiency:

$$\text{efficiency} = \frac{\text{tracks reconstructed}}{\text{total reconstructable tracks}}$$

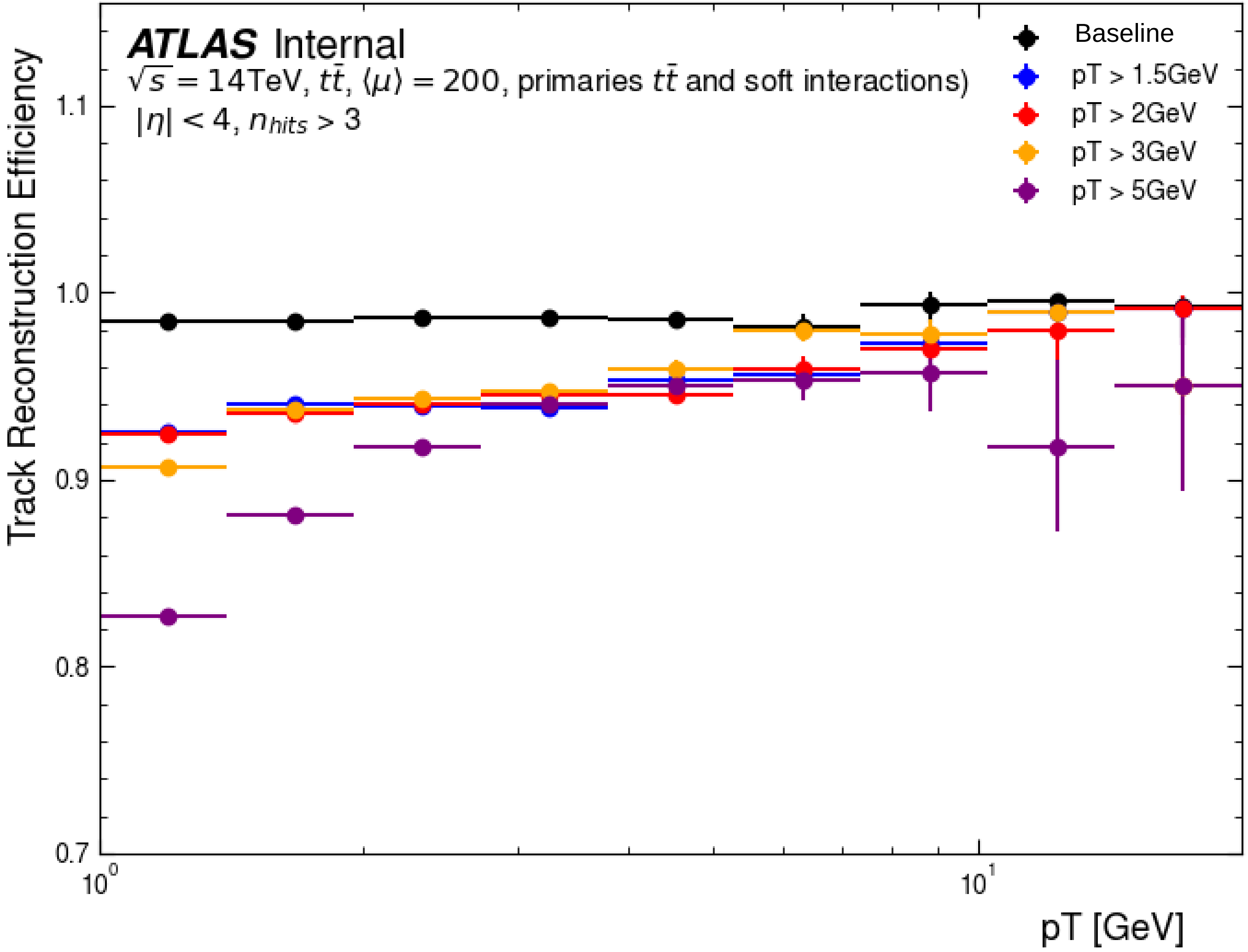
pT > X [GeV]	Reconstruction Efficiency
Baseline	0.985
1.5	0.933
2	0.931
3	0.924
5	0.864





Stage One:

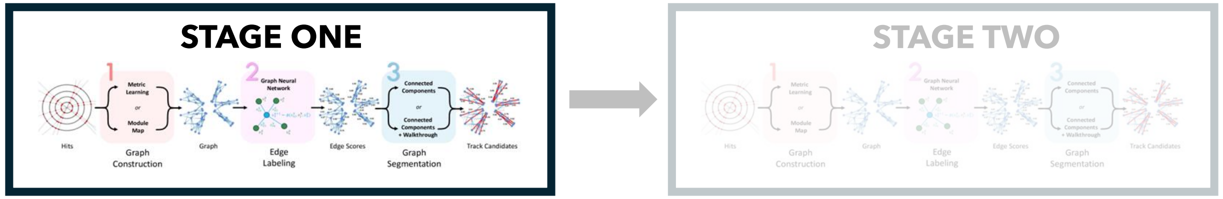
Track Reconstruction

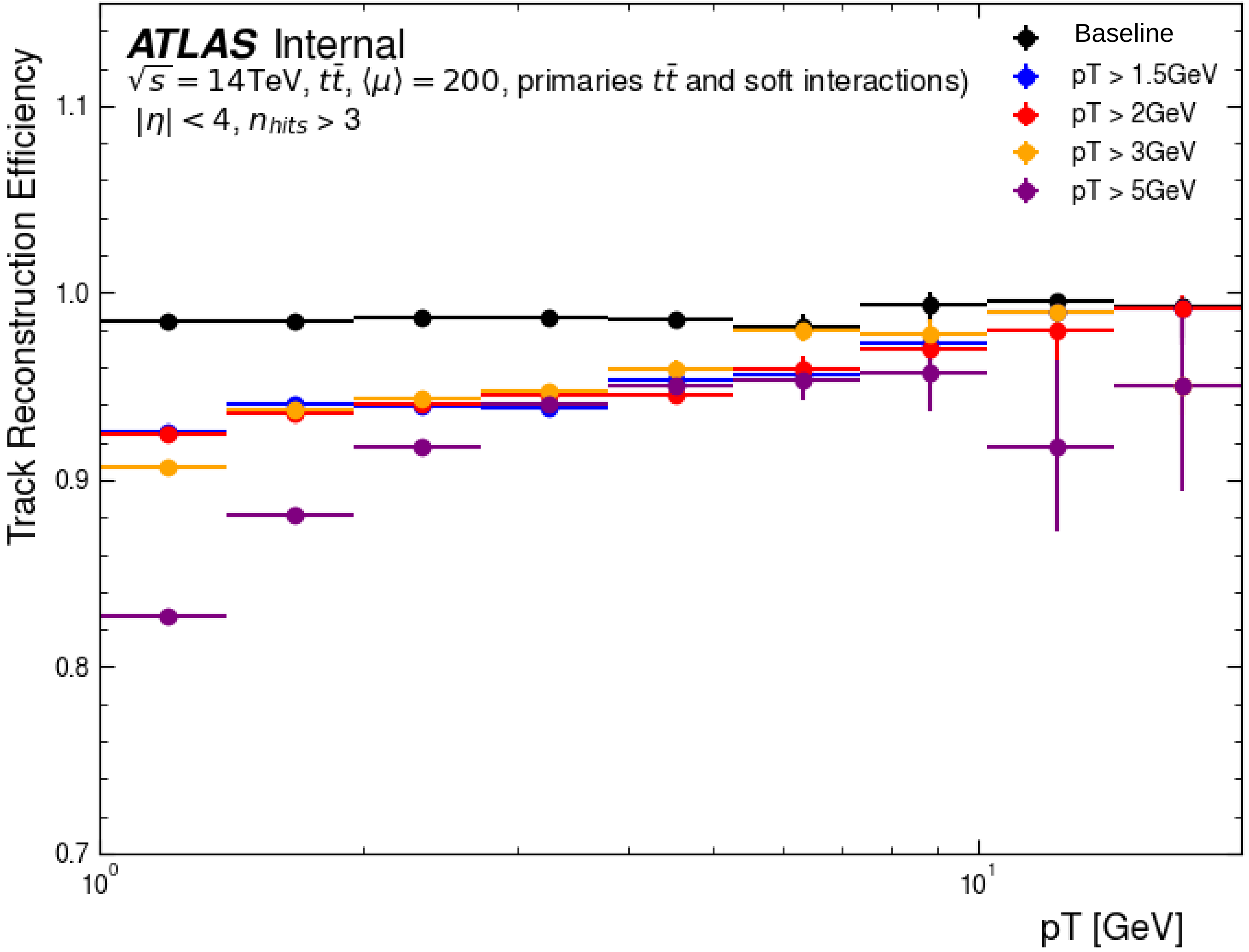


pT > X [GeV]	Reconstruction Efficiency
Baseline	0.985
1.5	0.933
2	0.931
3	0.924
5	0.864

Stage One:

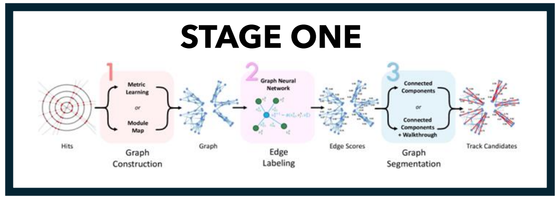
Track Reconstruction

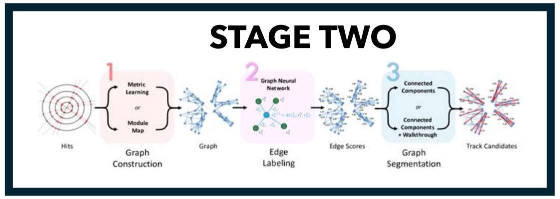




NEXT: STAGE TWO

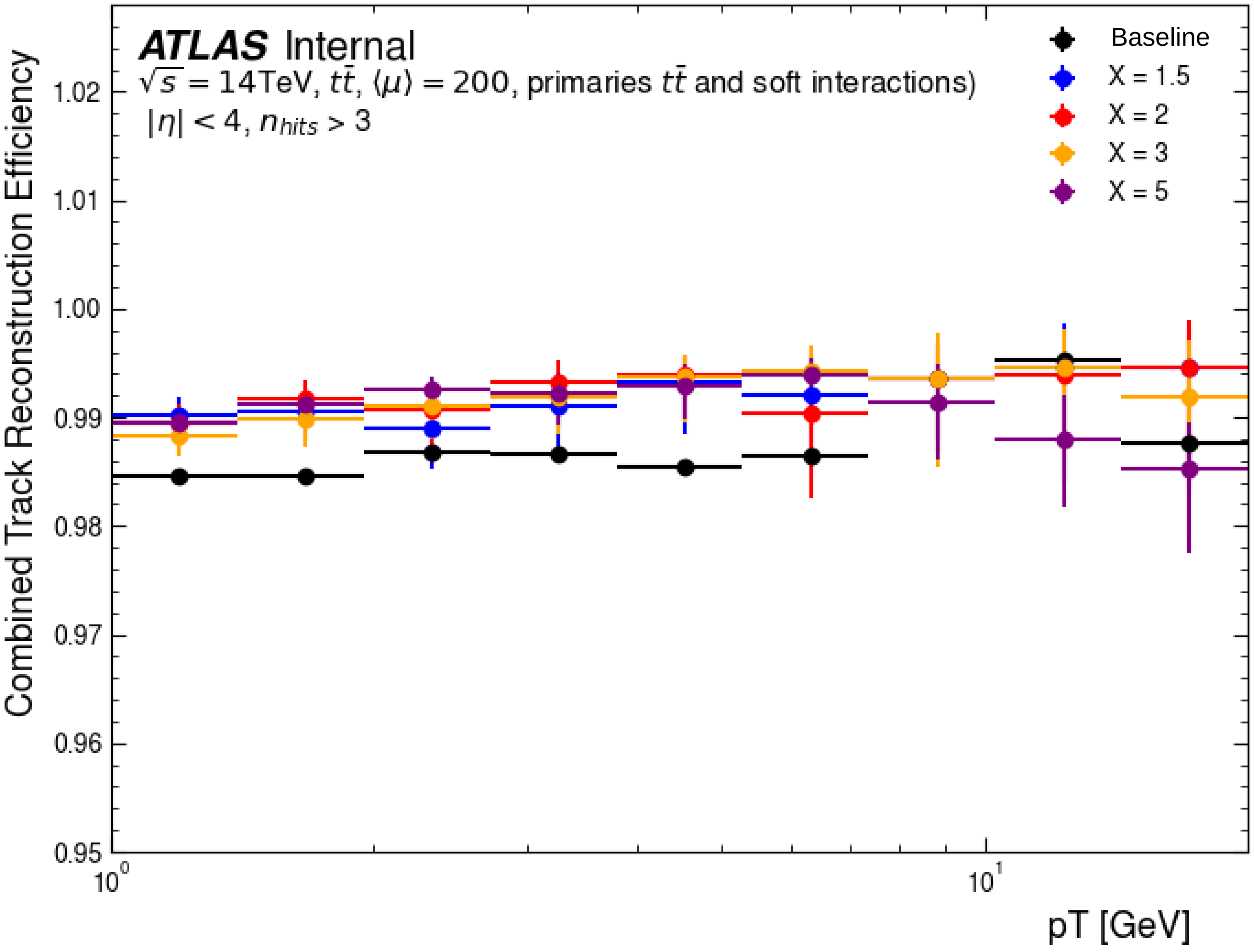
1. **Remove hits associated with reconstructed tracks from dataset**
2. **Graph Construction**
   - Build graphs with reduced dataset aimed at $1 < p_T$ [GeV] $< [1.5, 2, 3, 5]$
3. **Send through pipeline**



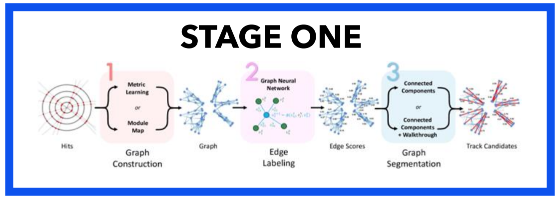


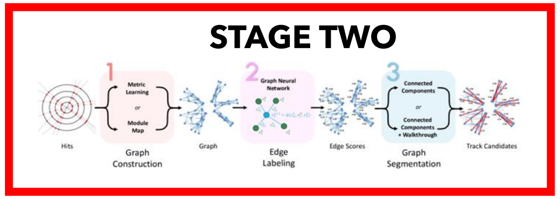
Combined results

of graphs pT > X GeV and 1 < pT [GeV] < X



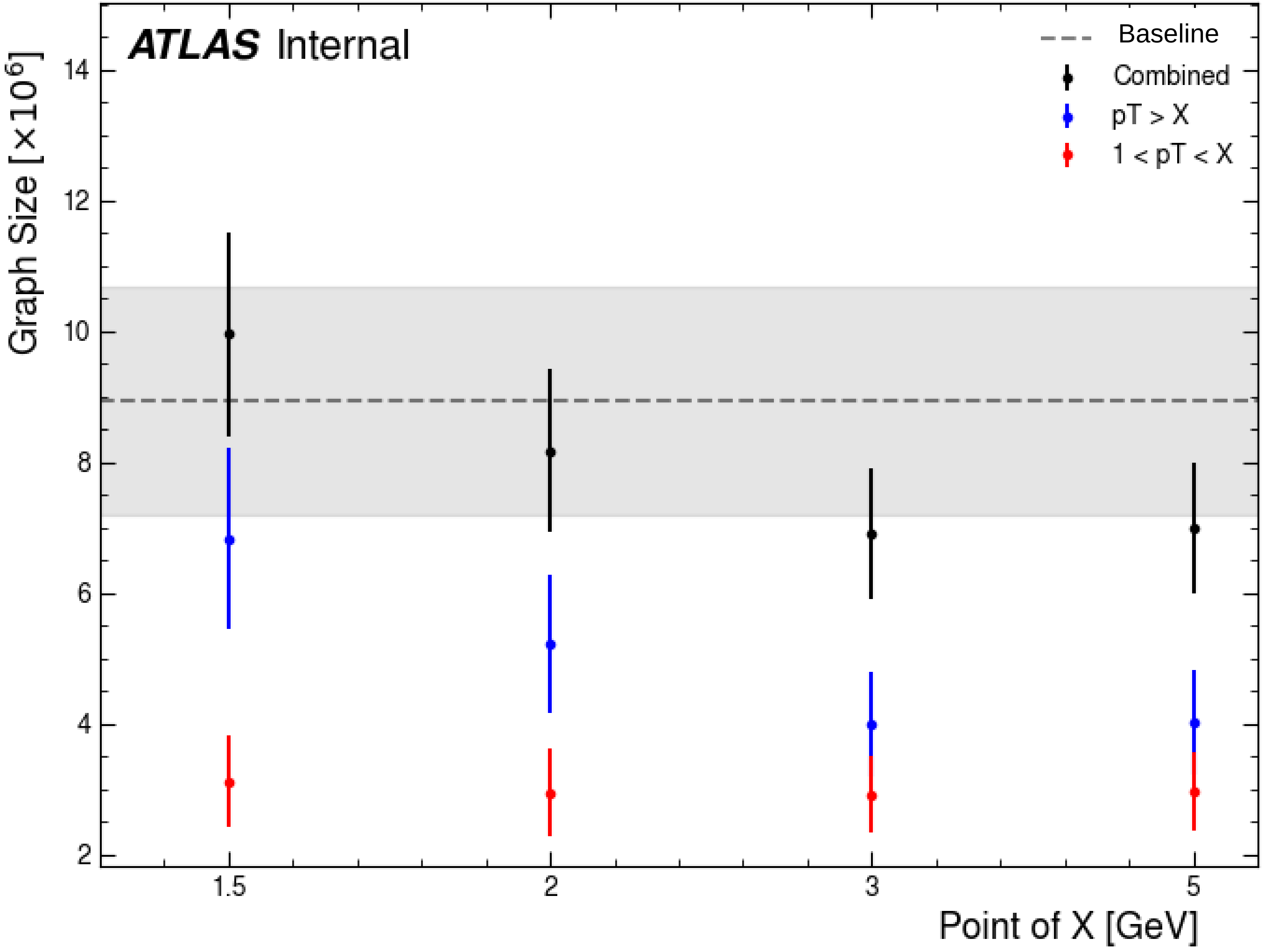
Point of X [GeV]	Combined Reconstruction Efficiency
Baseline	0.985
1.5	0.990
2	0.991
3	0.989
5	0.991





Combined results

of graphs pT > X GeV and 1 < pT [GeV] < X



Point of X [GeV]	Relative Graph Size (to reference)
1.5	1.11
2	0.91
3	0.77
5	0.78

Conclusions

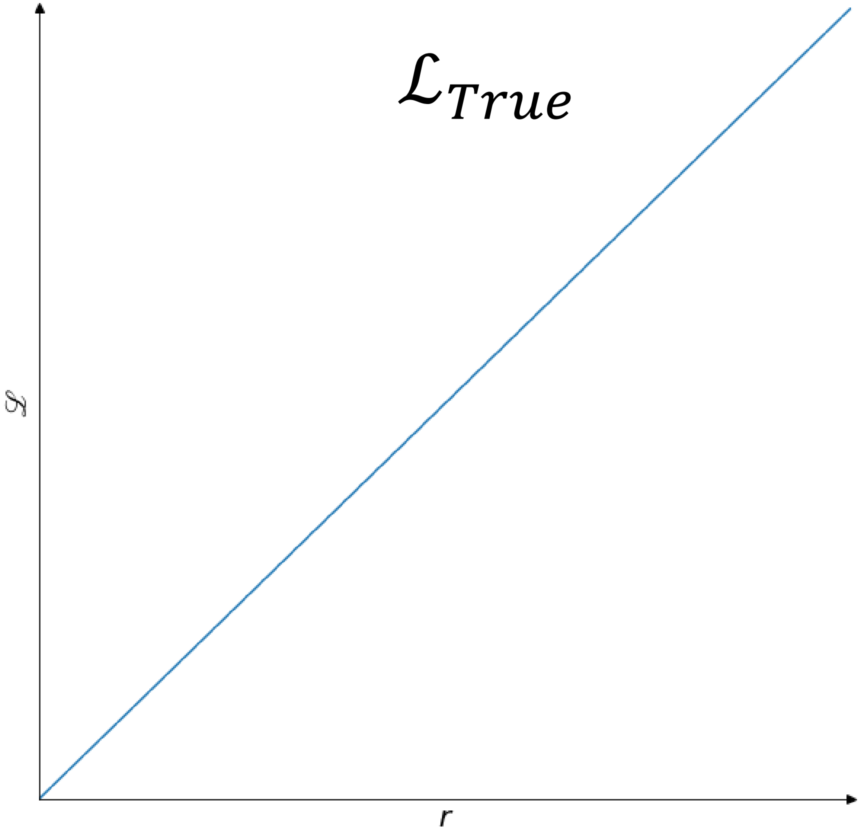
- Graph neural networks a promising avenue for track finding
- Achievement of 98.5% track efficiency as baseline – room for improvement
- Reduction in graph size is possible by building pT - specialised graphs
- What about throughput?
- Other physics-informed optimizations?

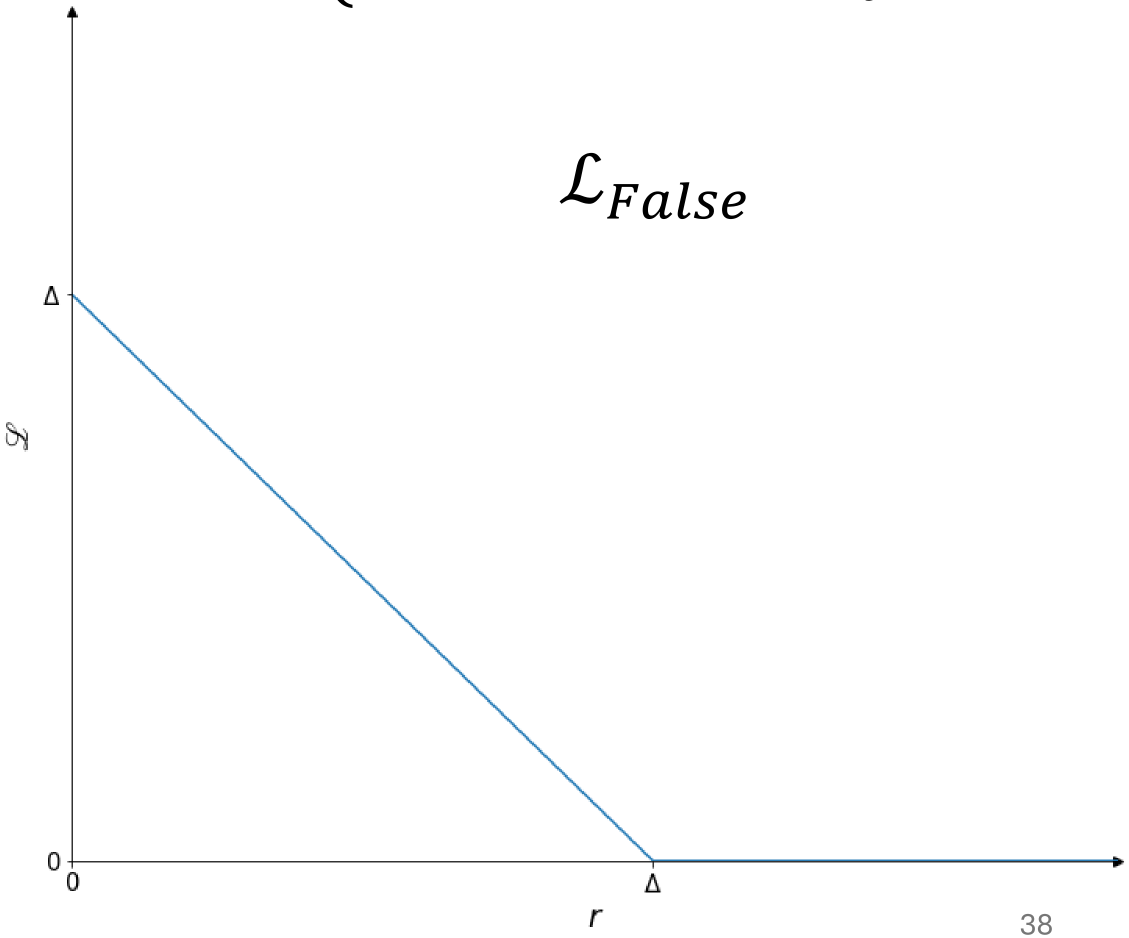
BACKUP

The MLP is trained using a weighted hinge loss.

$$\mathcal{L} = w_{true}\mathcal{L}_{True} + w_{false}\mathcal{L}_{False}$$

$$\mathcal{L}_y = \begin{cases} r & \text{if } y = \text{True} \\ \max(0, \Delta - r) & \text{if } y = \text{False} \end{cases}$$

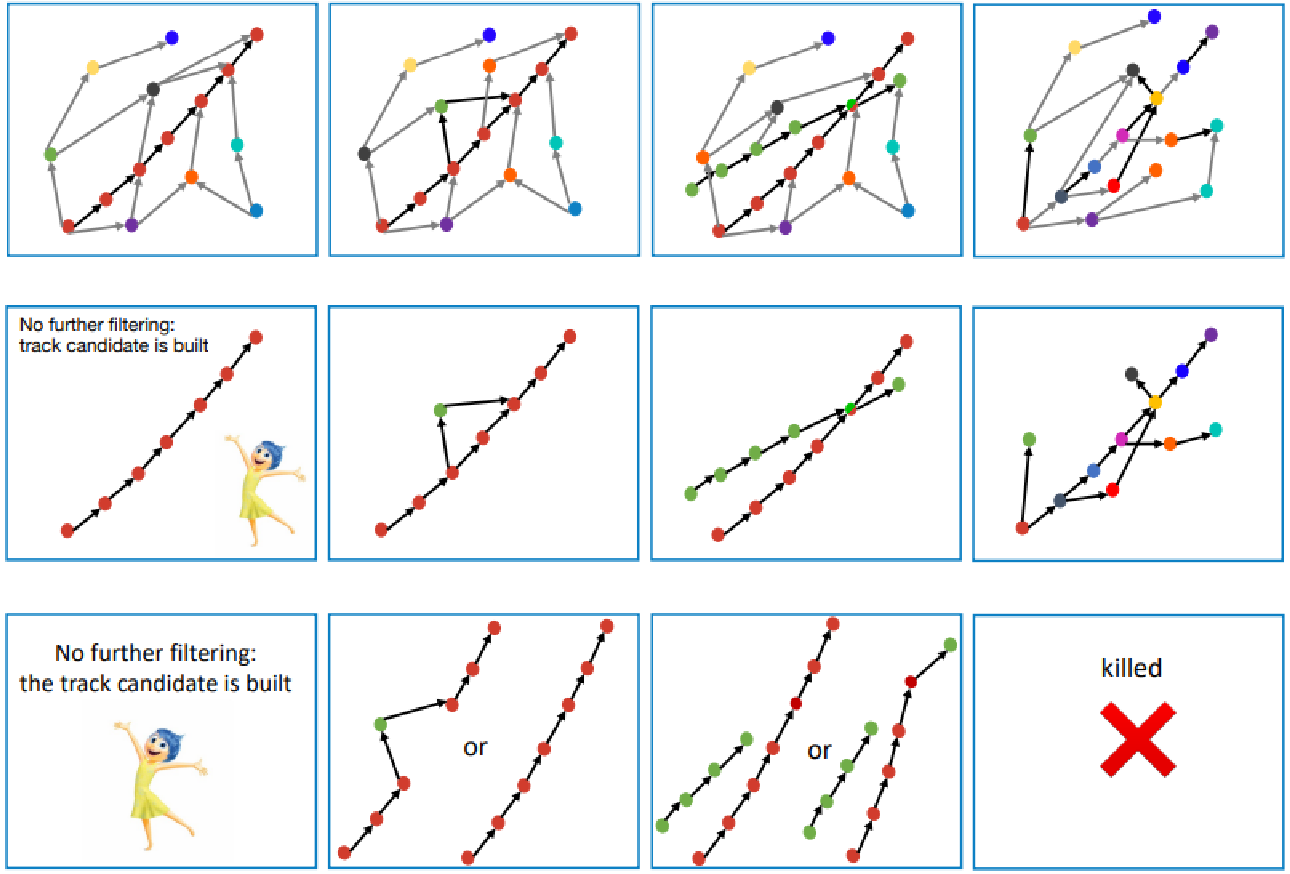




Graph segmentation:
Connected components + Walkthrough

[Charline Rougier at CTD 2022]

Legend:
Edge below threshold
Edge above threshold
Same color nodes = same particle nodes



Connected component algorithm,
with loose edge score cut

The better GNN performance
the more tracks are ready at this stage
and the faster the reconstruction is

Walk-through algorithm,
with tighter edge score cut

(Current graph segmentation mainly developed to complete the chain.
Not yet optimized, e.g. could be combined with a Kalman Filter.)

- FPGA: AMD Alveo U250
- GPU: Nvidia V100