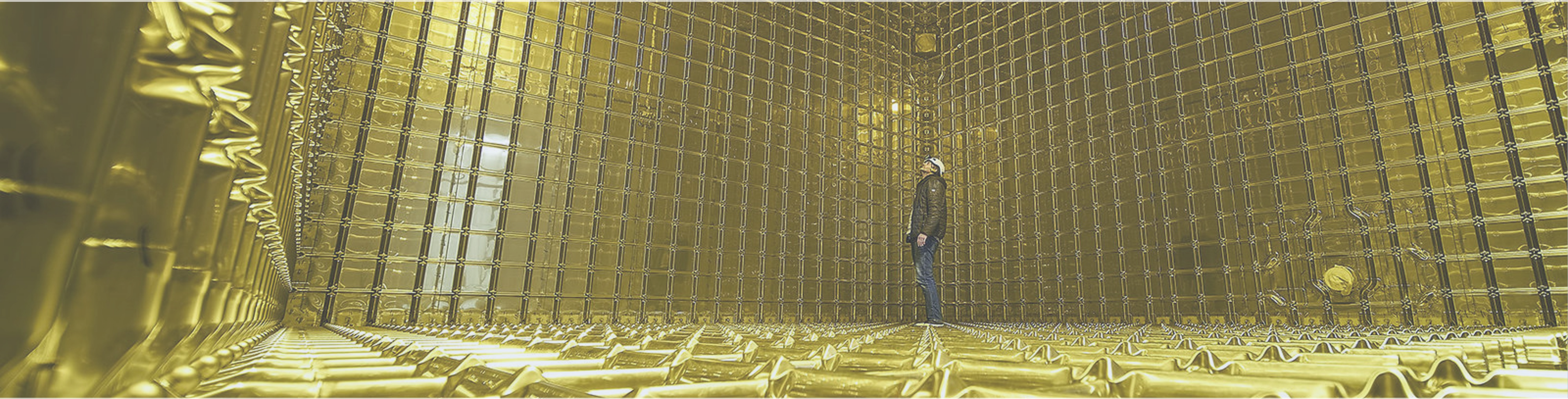
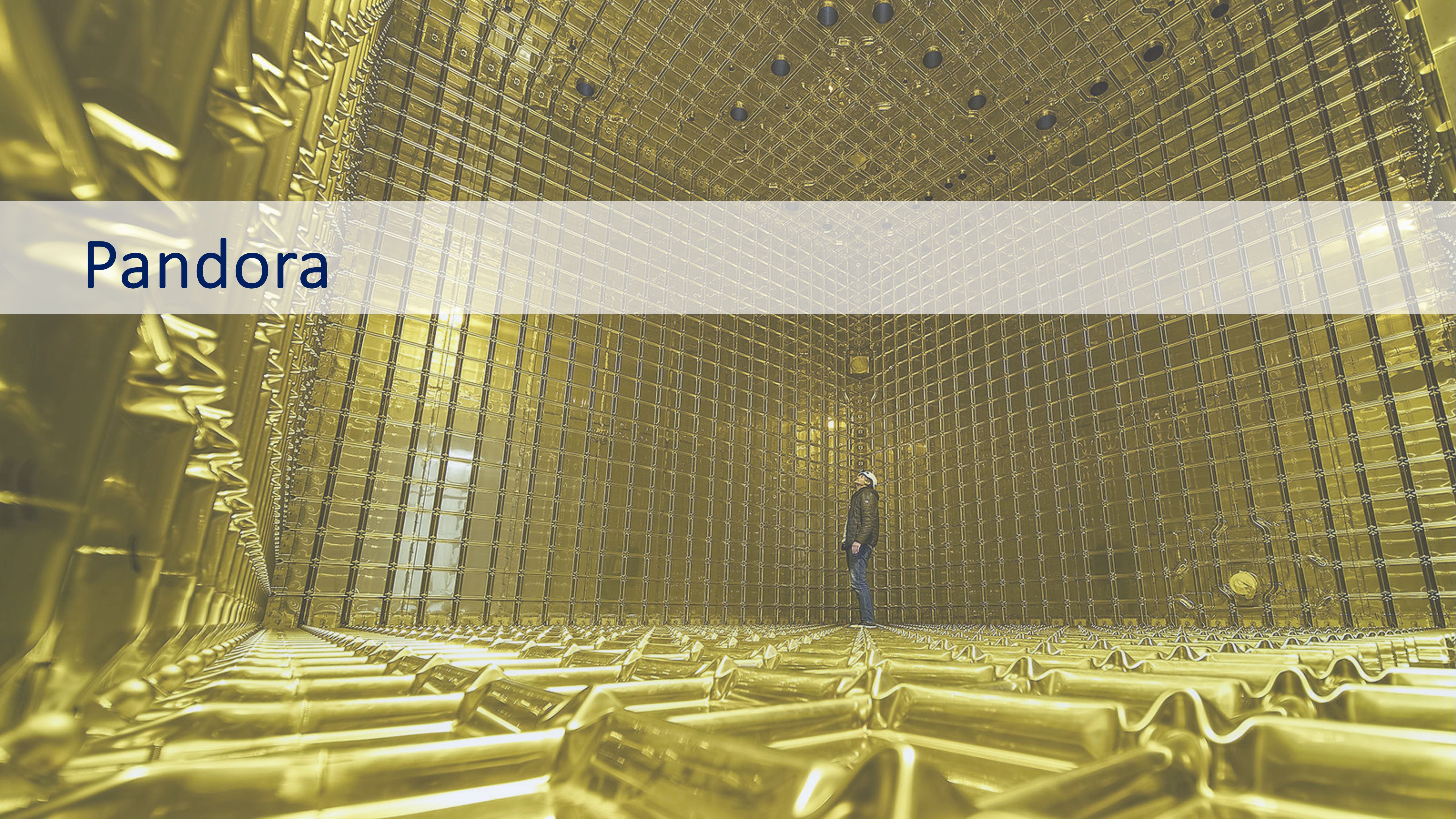
GNNs for Neutrino Hierarchy Building within Pandora



Isobel Mawby (they/them)

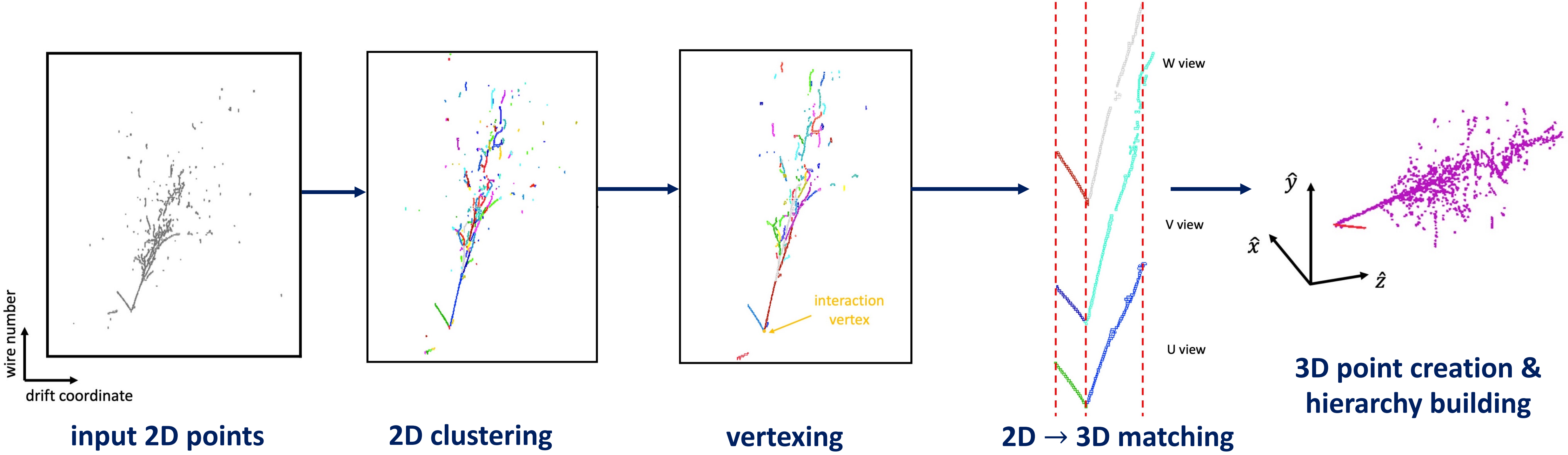
DUNE UK, University of Sheffield, 9-11th July 2024

Pandora



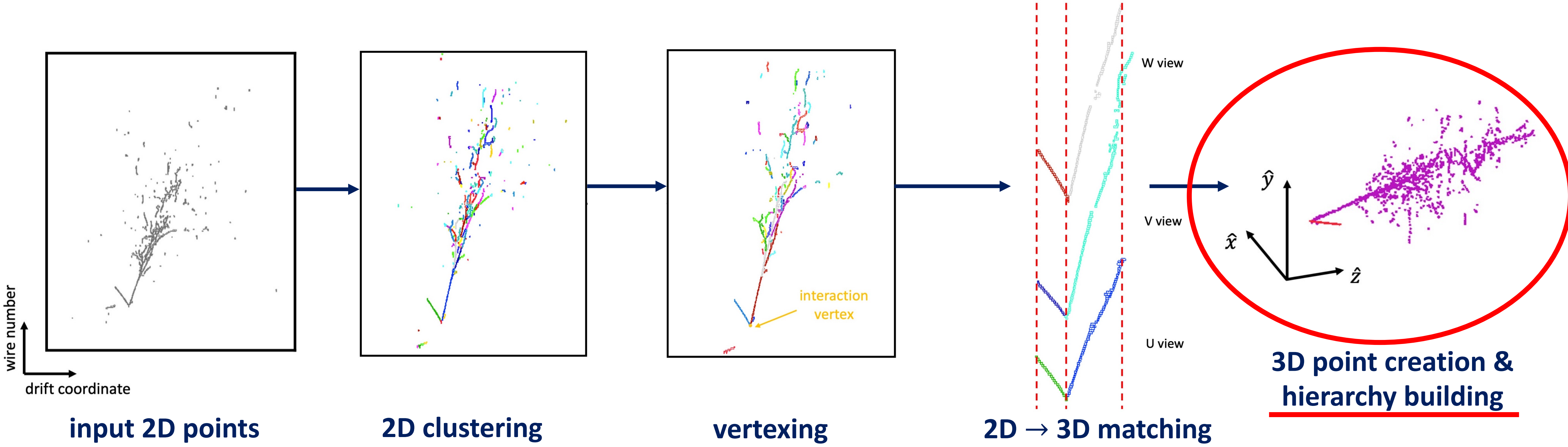
Pandora – the hope left in the jar

- Pandora is a **pattern recognition software**, used to reconstruct neutrino interactions
- The detail of the **fine-grain images** we obtain from LArTPCs presents a **huge** reconstruction challenge
- Pandora overcomes this with a **'multi-algorithm approach',** where the reconstruction is split into stages composed of many **'hand engineered'** and **machine learning-based algorithms**



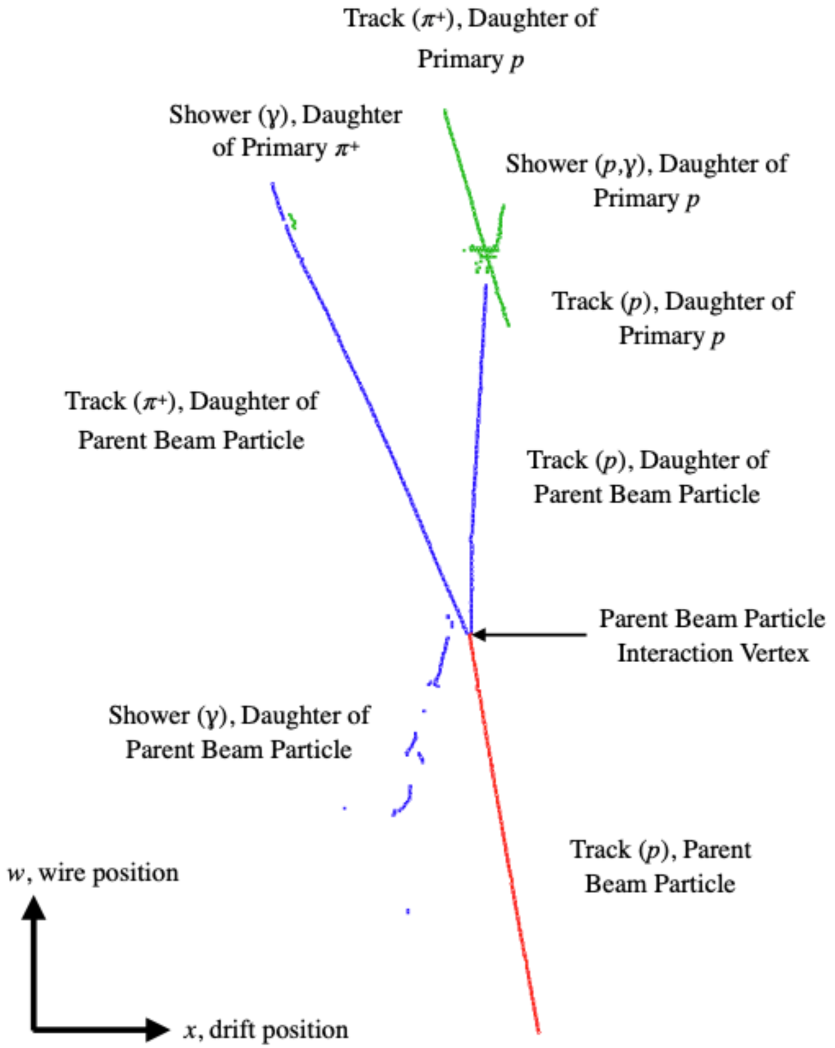
Pandora – the hope left in the jar

- Pandora is a **pattern recognition software**, used to reconstruct neutrino interactions
- The detail of the **fine-grain images** we obtain from LArTPCs presents a **huge** reconstruction challenge
- Pandora overcomes this with a **'multi-algorithm approach',** where the reconstruction is split into stages composed of many **'hand engineered'** and **machine learning-based algorithms**



Neutrino Hierarchy Construction

The Neutrino ‘Hierarchy’



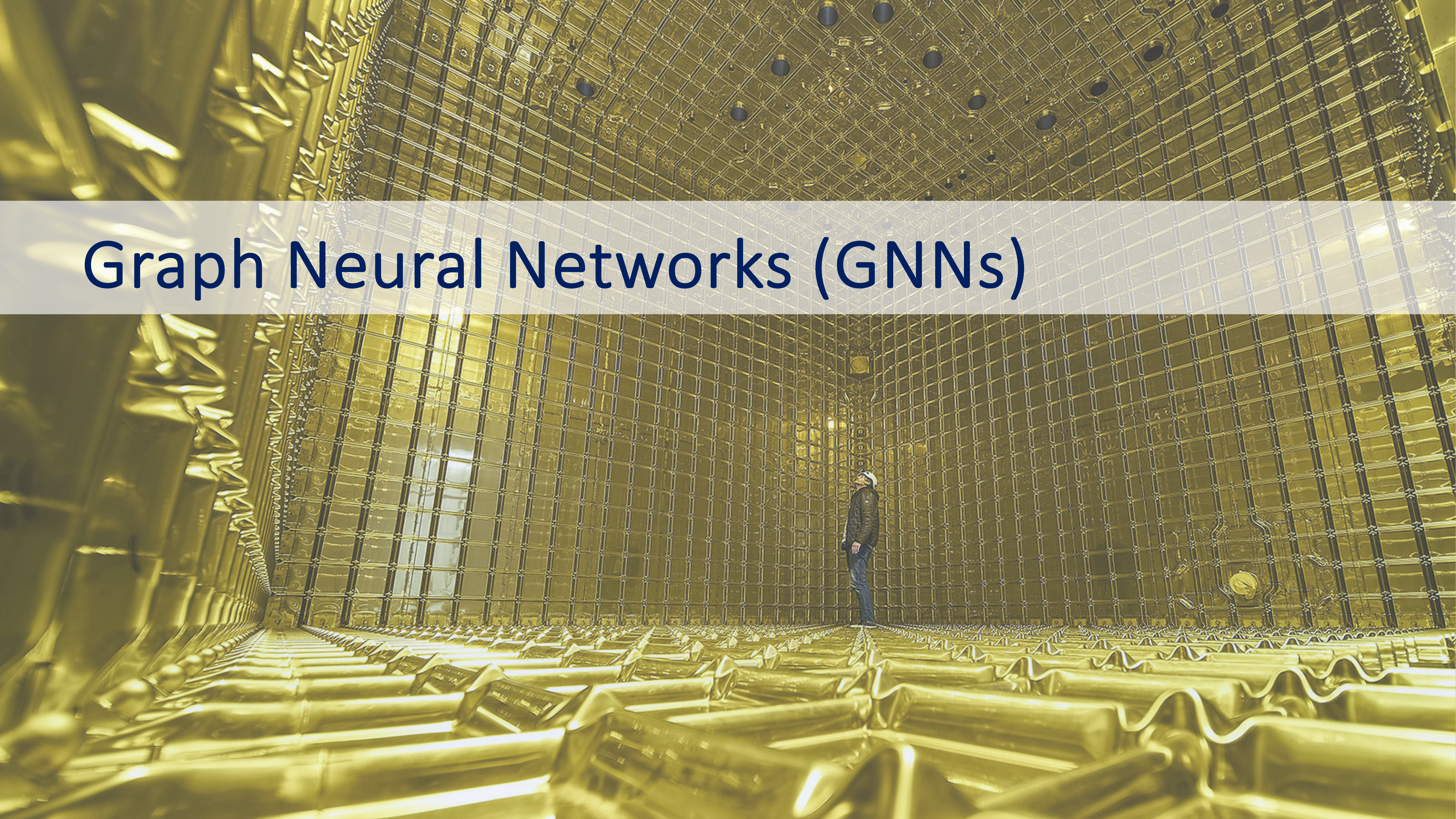
- Important for analyses?
  - **Cross-section analyses,** where one searches for specific topologies
  - **CP-violation analyses,** where μ/π separation is crucial
  - **Energy estimation,** where a particles evolution shapes assumptions made/energy estimators used

- Currently Pandora uses **a ‘hand engineered’ algorithm**, which identifies most particles as neutrino children unless absolutely sure otherwise

$\Longrightarrow$ this needs to be improved!

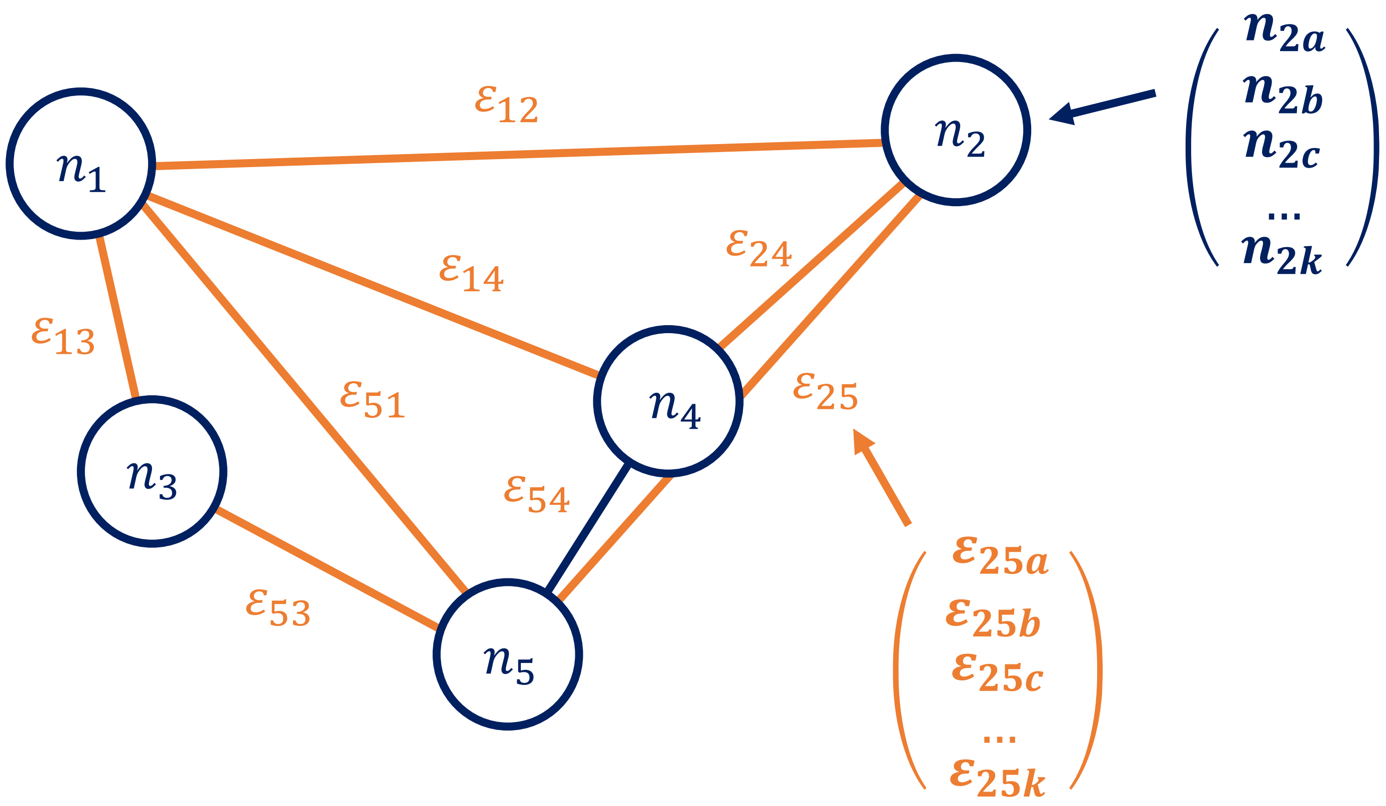
Particle hierarchies look like graphs! Maybe we could use a GNN for this task…

Graph Neural Networks (GNNs)



Graph Neural Networks (GNNs)

- Inputs are graphs composed of **nodes (n)** and **edges (ε)**
- Nodes **MUST** have node features, whilst edges **CAN** have edge attributes and weights



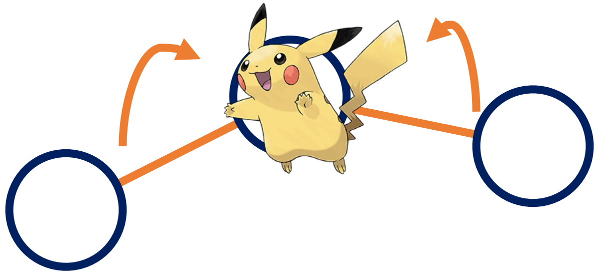
GNNs are used for:

- node classification
- edge classification
- link prediction
- graph classification

Graph Neural Networks (GNNs)

- GNNs are composed of **layers**
- In training, the aim is for the GNN to learn the best **node representation** for a given classification task
- I've been constructing a message-passing neural network (MPNN), which considers both node features and edge attributes

A message between each pair of connected nodes is calculated



The message is a function of the source and target node features and the connecting edge attributes

Each node aggregates its received messages (sum, max, etc…)

Each node updates its features

This transformation is a function of its current features and aggregated messages

- The nodes of a deeper network therefore see **information from further afield**

GNNs for Hierarchy Building

Defining our task

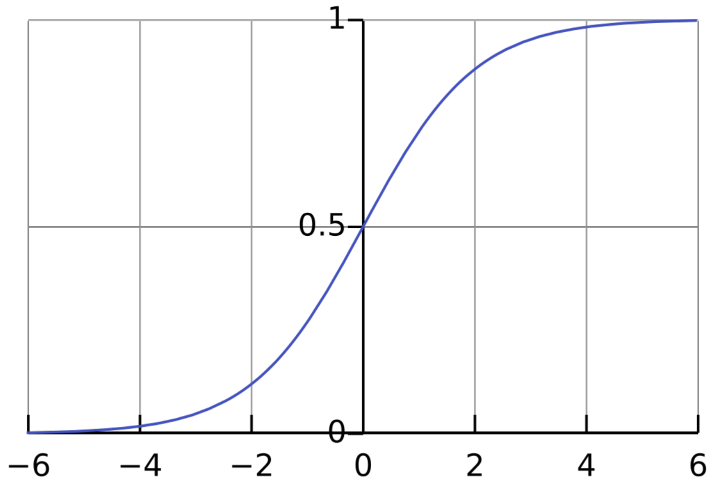
- We want to determine **parent-child links** $\Longrightarrow$ **link prediction**
- But how do we use our node representations to determine whether an edge is true/false?

We take the dot product!

$$\begin{pmatrix} n'_{1a} \\ n'_{1b} \\ \dots \\ n'_{1k} \end{pmatrix} \bullet \begin{pmatrix} n'_{2a} \\ n'_{2b} \\ \dots \\ n'_{2k} \end{pmatrix}$$

→

Pass through a sigmoid function



→

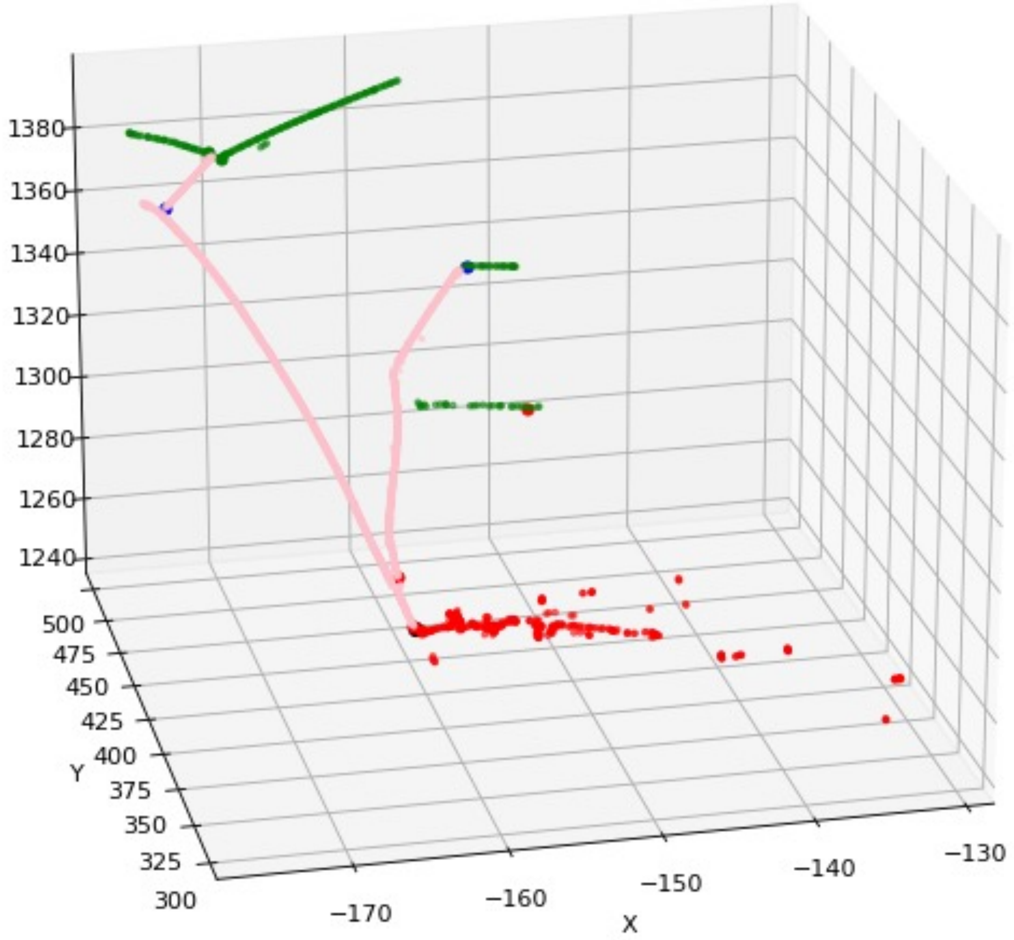
Compare to optimised threshold

prediction > threshold → **True edge**

prediction < threshold → **False edge**

Network truth/target

We want to determine parent-child links ⟹ link prediction



- A **positive edge** corresponds to a **true parent-child link**

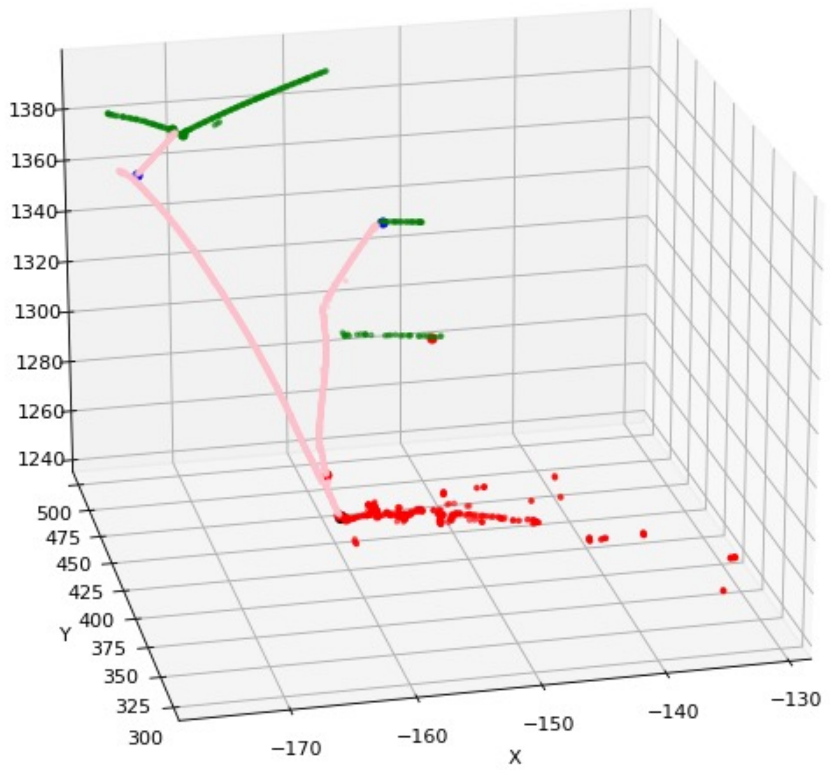
- But who is the true reconstructed parent of:
  - a π^0 decay photon?
  - a proton from neutron interaction?
  - a particle whose parent hasn't been reconstructed?
  - the second half of a split muon?
  - etc…

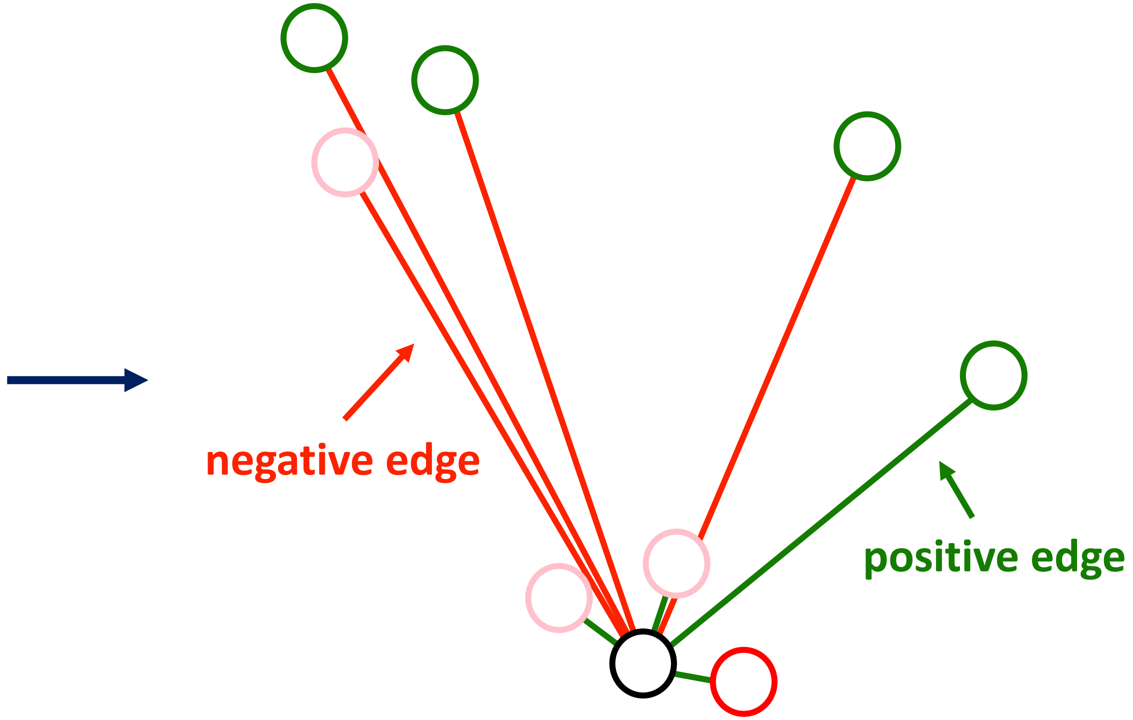
- My truth definition is that a particle's parent is the **latest ancestor that has been reconstructed**

this is definitely an area for improvement…

Two step approach

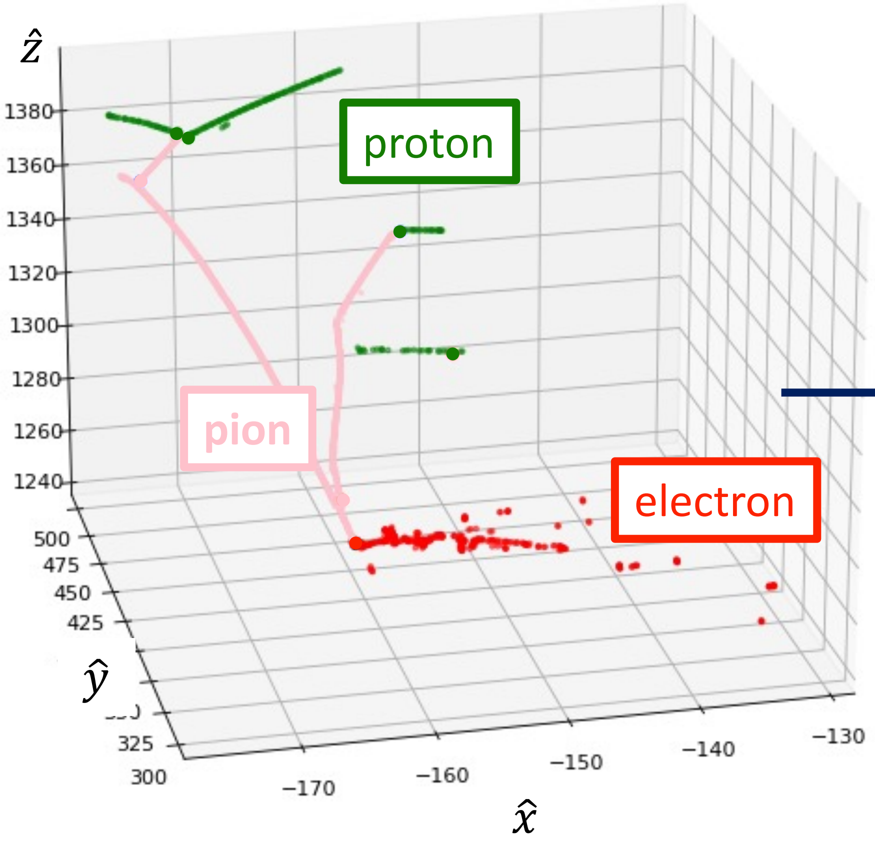
⟹ **use a two-step approach:** first work out **nu-particle edges** (THIS TALK!), then particle-particle edges





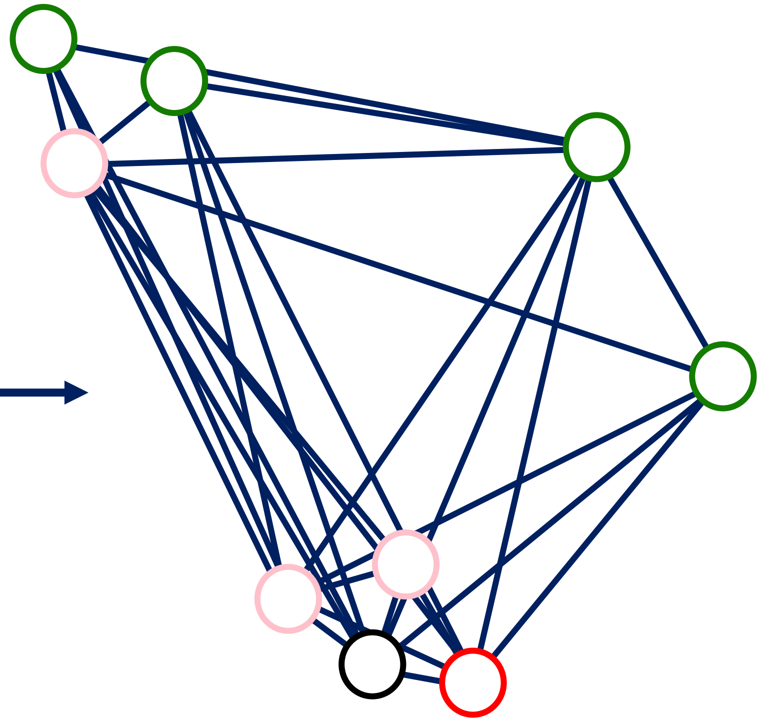
Creating the Graph

Pandora 3D reco. output



1) Turn each (> 100 hits) particle into a node
2) Turn each nu-particle link into an edge
3) Rank each particle-particle link based on their:
   a) separation
   b) opening angle
4) Create a combined rank
5) Turn 80% of the best ranked links into edges

message passing graph



* colours only to aid visualisation

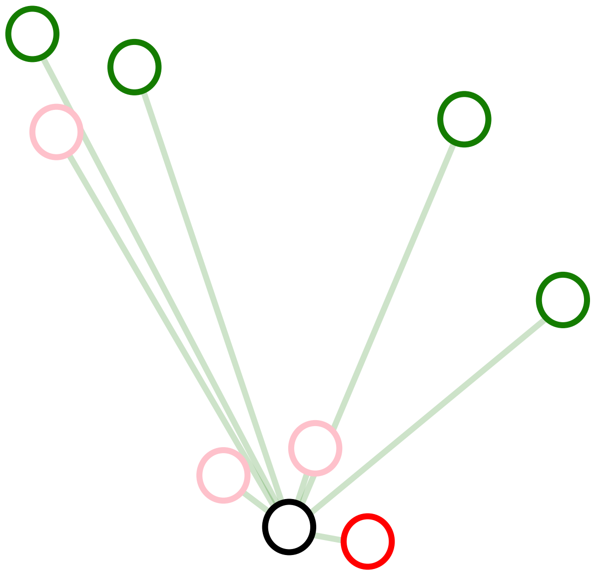
Input Graph: Information

Node features:

- # 2D hits
- Is neutrino?
- length
- PID scores
- track-shower score

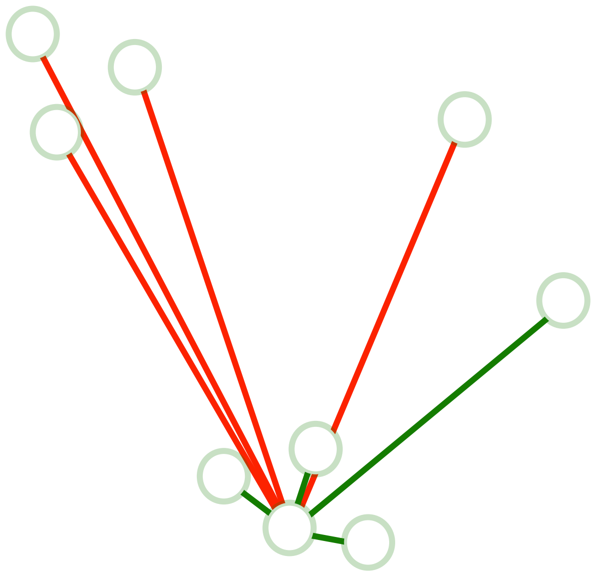
- charge
- charge distribution

- vertex x/y/z position
- end x/y/z position
- initial x/y/z direction
- displacement
- distance of closest approach



Edge attributes:

- distance of closest approach



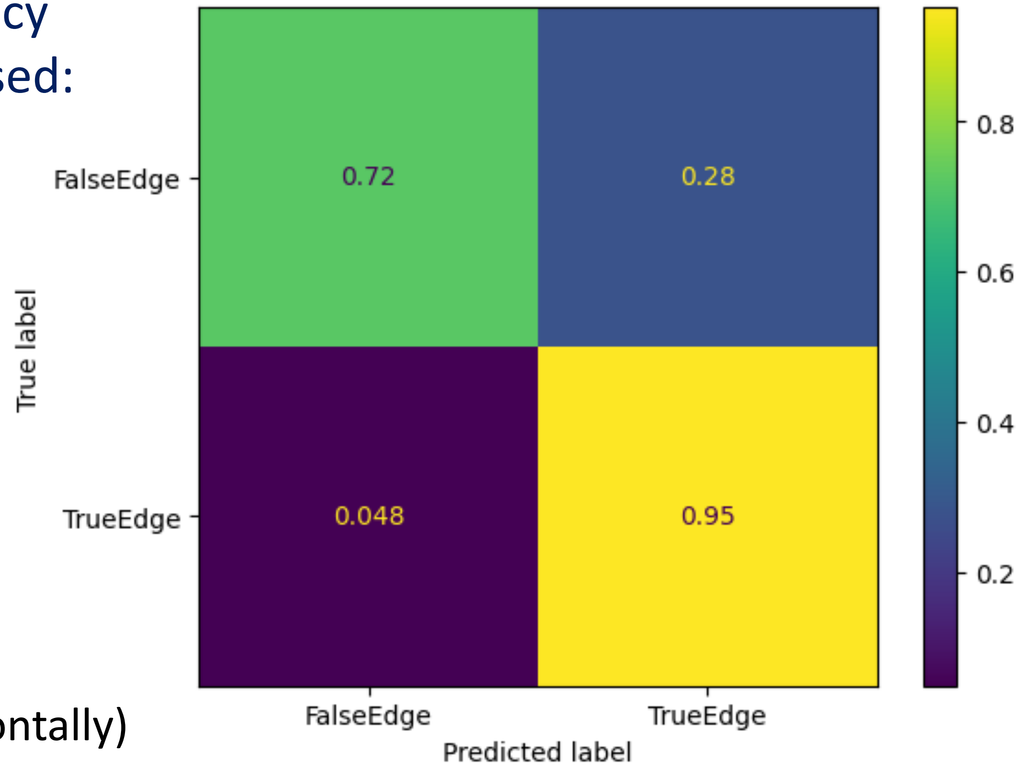
- First iteration, definitely an area for development
- Some variables are cheated to (somewhat) detach reco. and network performance ⟹ **can focus on network architecture**

Performance

Performance

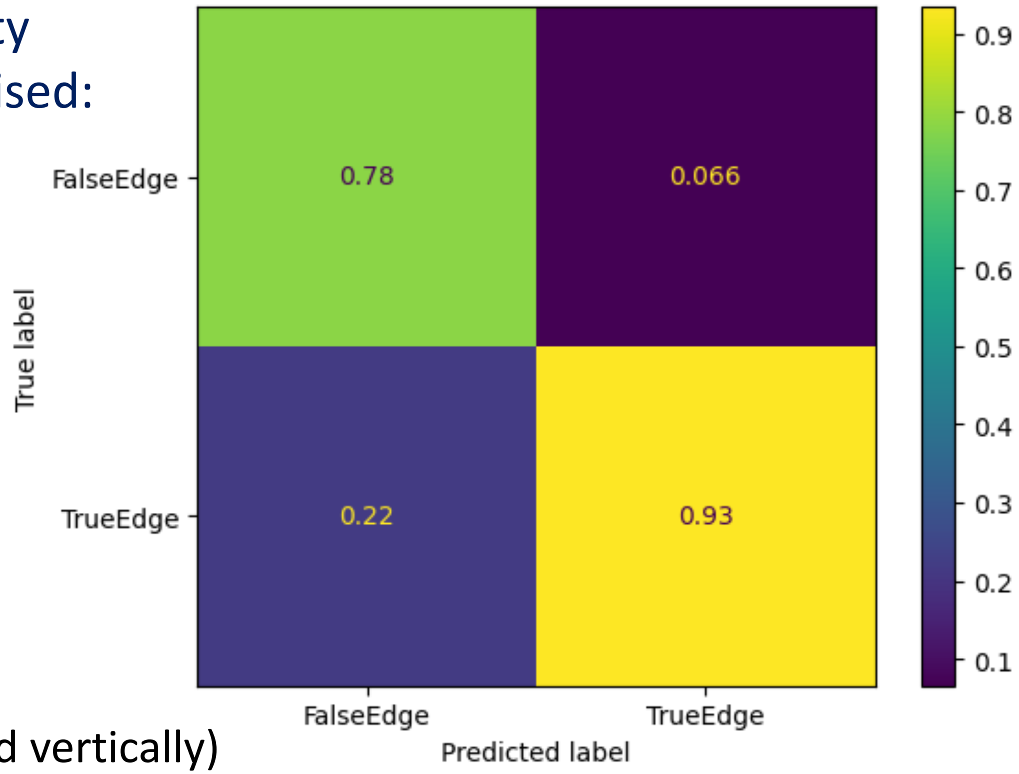
- Network is trained on ~200,000 graphs and validated on ~51,000 graphs
- 85%(15%) of edges are positive(negative), network loss function is weighted accordingly
- Trained with 40 epochs

Efficiency normalised:



(read horizontally)

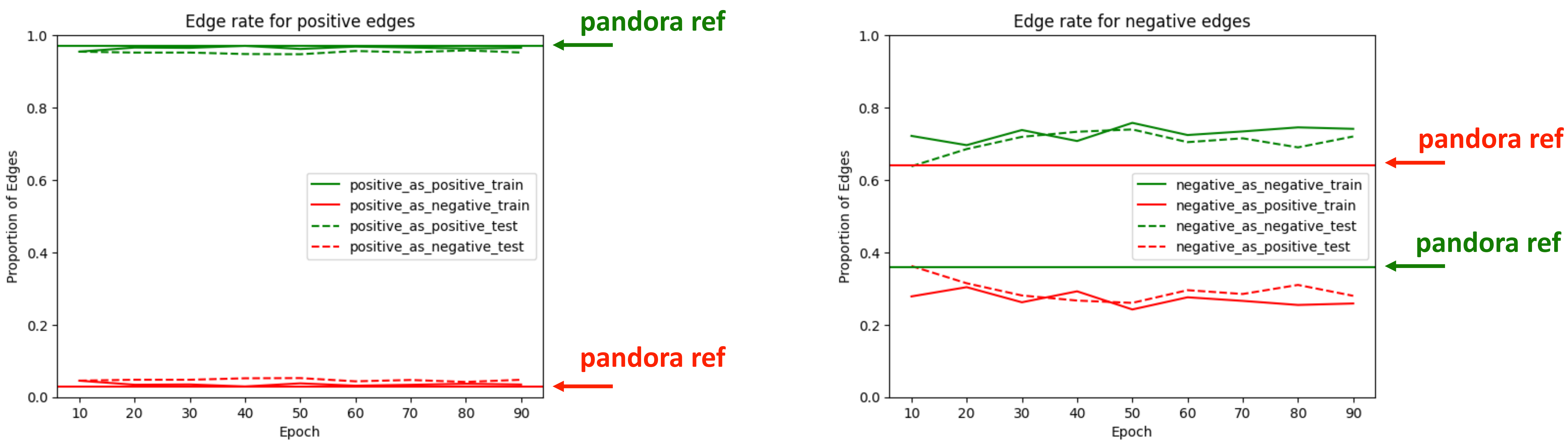
Purity normalised:



(read vertically)

Comparison

- Comparing with the current Pandora algorithm, we see that we already have an improvement

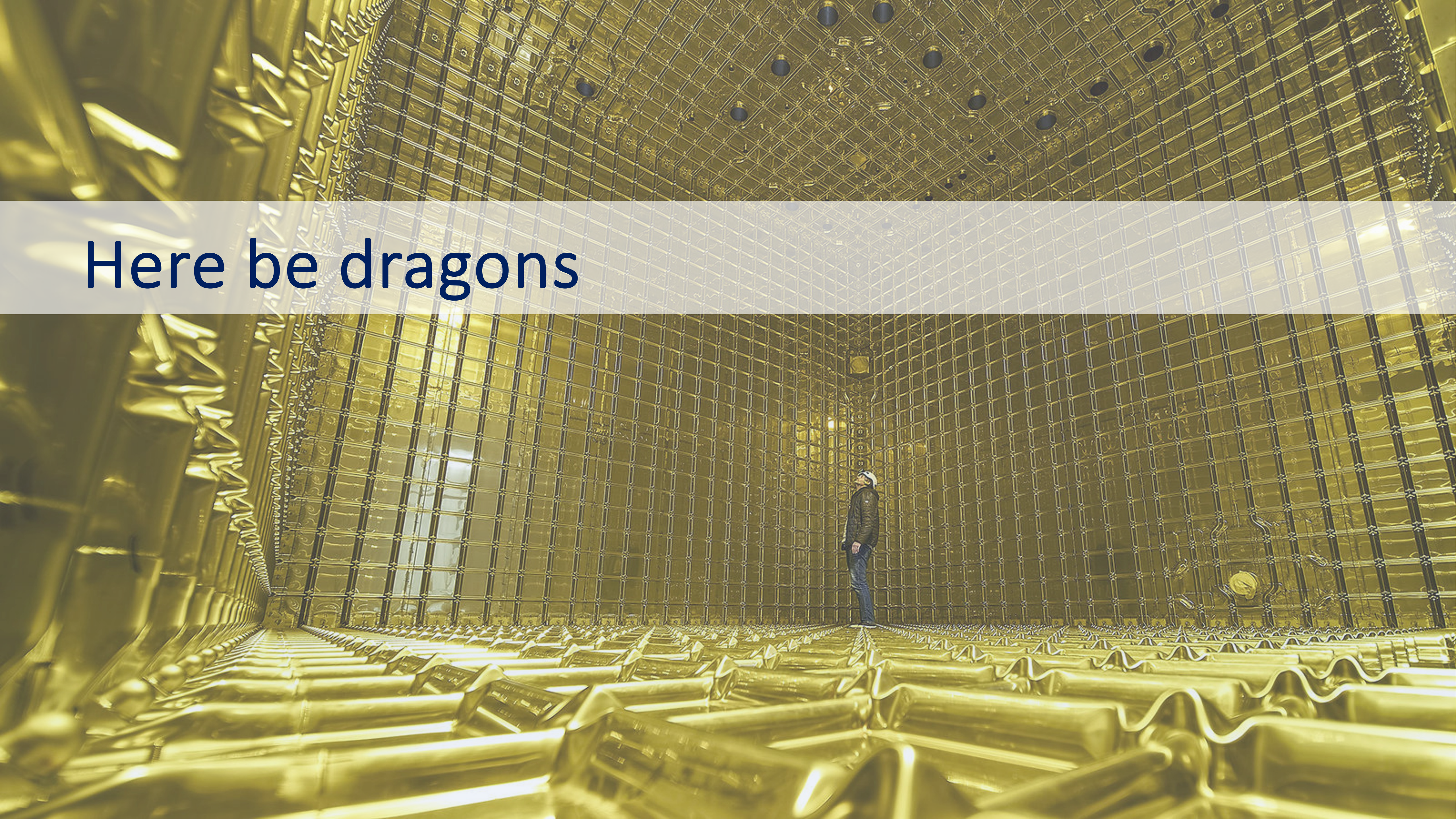


Conclusions

- Pandora is a **pattern recognition software,** used to reconstruct neutrino interactions
- Our reconstruction performance is best understood in terms of **physics analysis,** and these results are used to **drive developments**
- The success of the hierarchy building is **crucial for many physics analyses** across all LArTPC experiments that utilise Pandora (not just DUNE!)
- I've introduced a new approach to hierarchy building, which uses a GNN
- First iteration shows promise! But lots of work to be done:
    - Updates to the truth definition
    - Expansion of node features and edge attributes
    - Implementation of the second pass
    - Exploration of heterogenous graphs?
    - etc…

Thank you for listening!

Here be dragons



Here be dragons