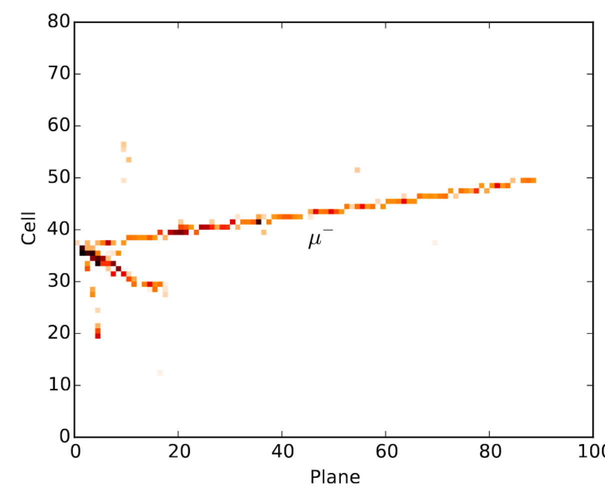
Funded by DoE through the Exa.TrkX project

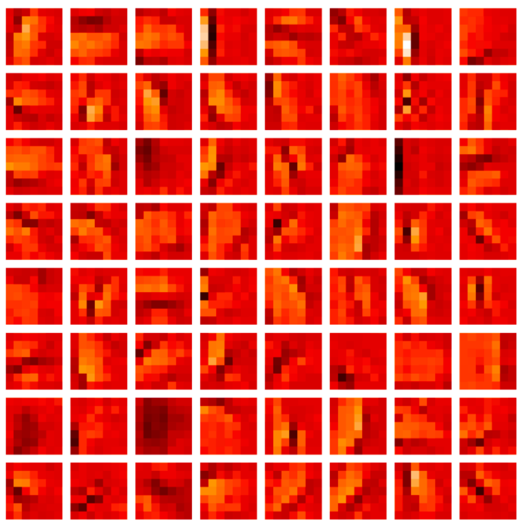
# Graph Neural Networks for Reconstruction in Liquid Argon Time Projection Chambers

Jeremy Hewes
Connecting the Dots workshop
April 22nd 2020

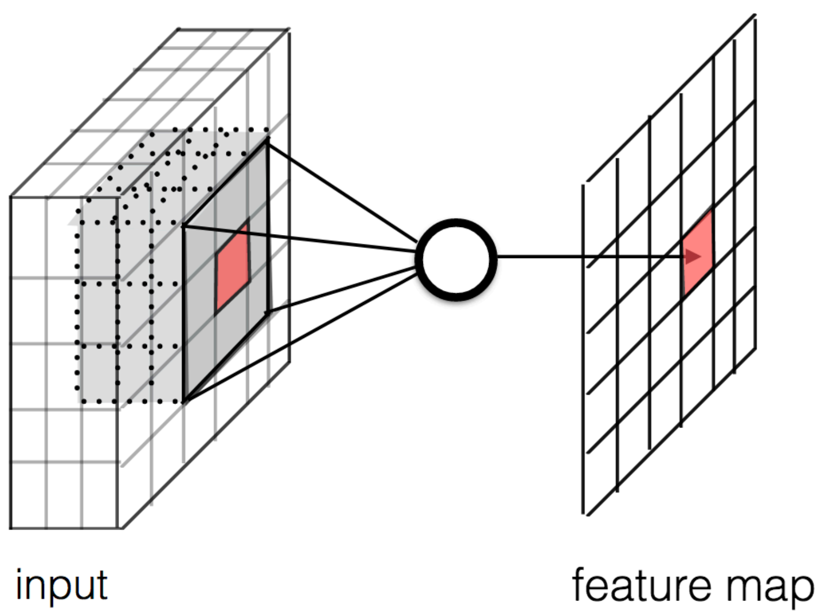
# Neutrino physics

- Convolutional neural networks show great promise in image classification over the past decade.
- Most neutrino detector technologies naturally provide pixel maps which can be classified using CNNs.
- Examples: NOvA, MicroBooNE, DUNE.



**arXiv:1604.01444**

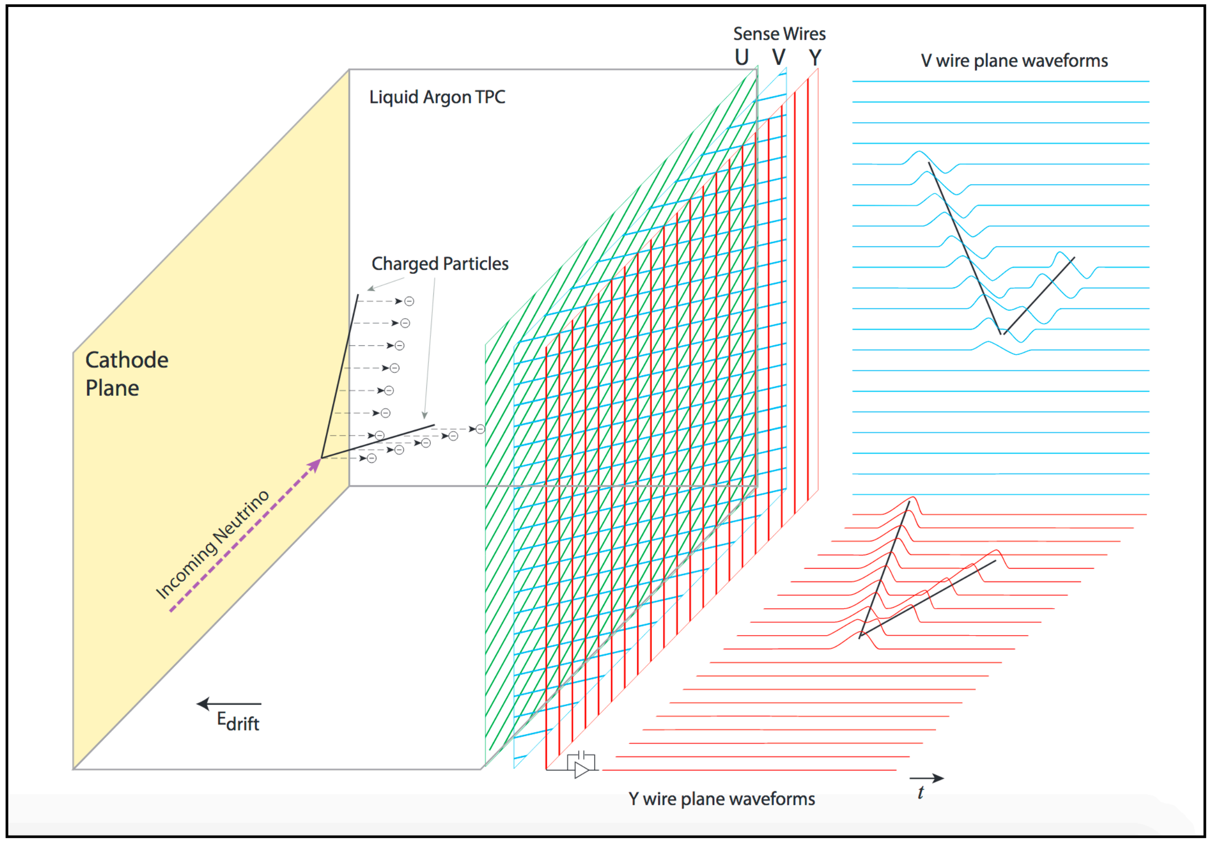



- Issues with this approach:
  - **Dense** representation of **sparse** data.
  - Operate over mostly empty space!
  - Need to transform 3D representation into voxels.
- GNNs can work with reconstructed spacepoints natively.

# Liquid Argon Time Projection Chambers

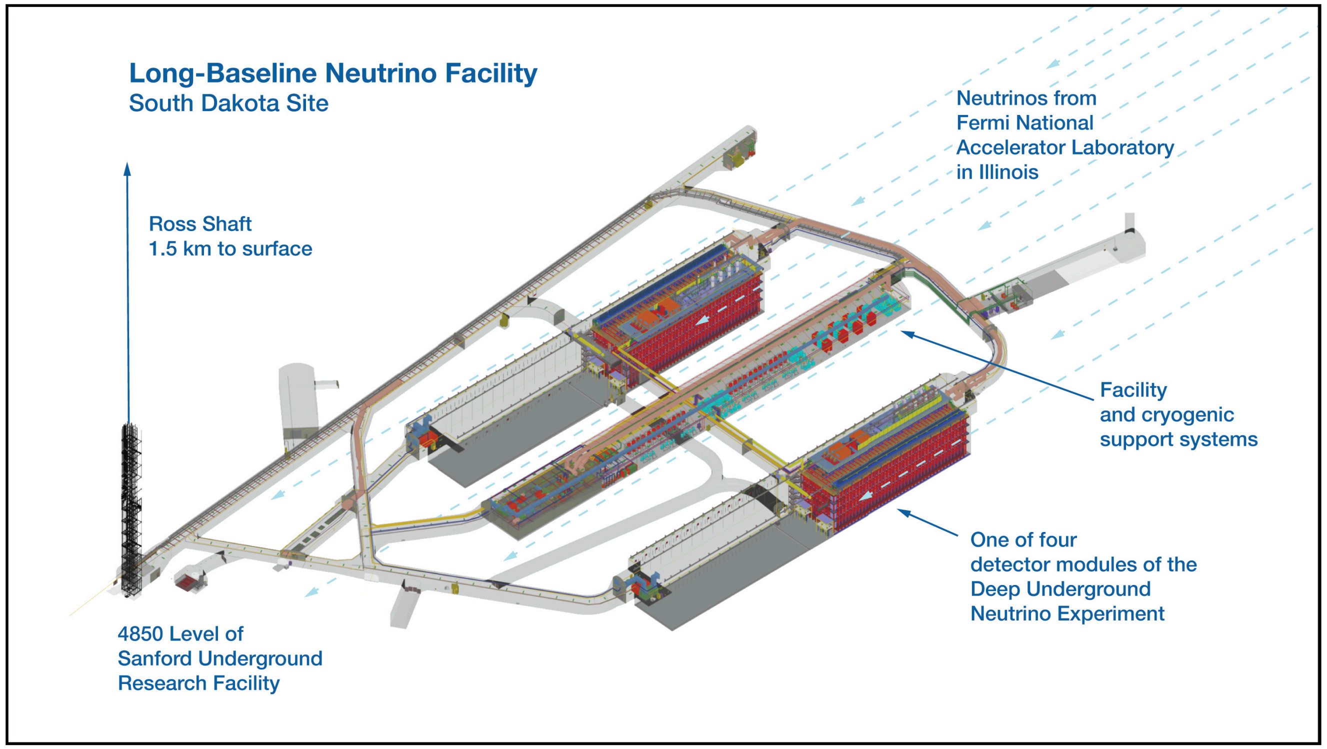
- Liquid Argon Time Projection Chambers (LArTPCs) are currently a very important detector technology for neutrino physics.
  - At FNAL: MicroBooNE, Icarus, SBND.
  - Future: DUNE (70kT LArTPC deep underground, plus near detector).

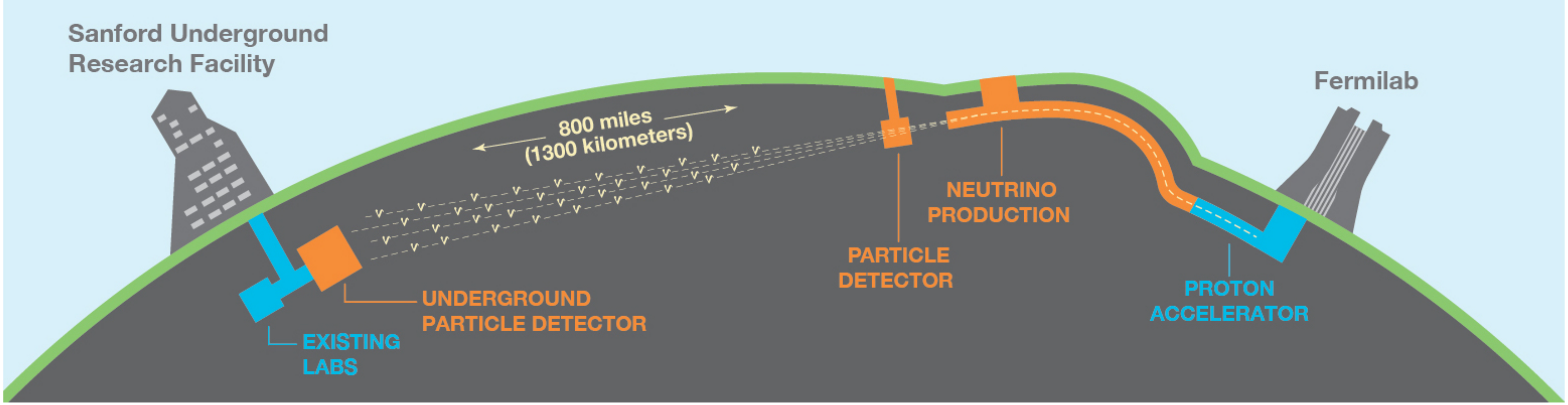
- Charged particles ionize liquid argon as they travel.
- Ionisation electrons drift due to HV electrode field, and are collected by anode wires.
- Wire spacing ~3mm – produce **high-resolution images**.



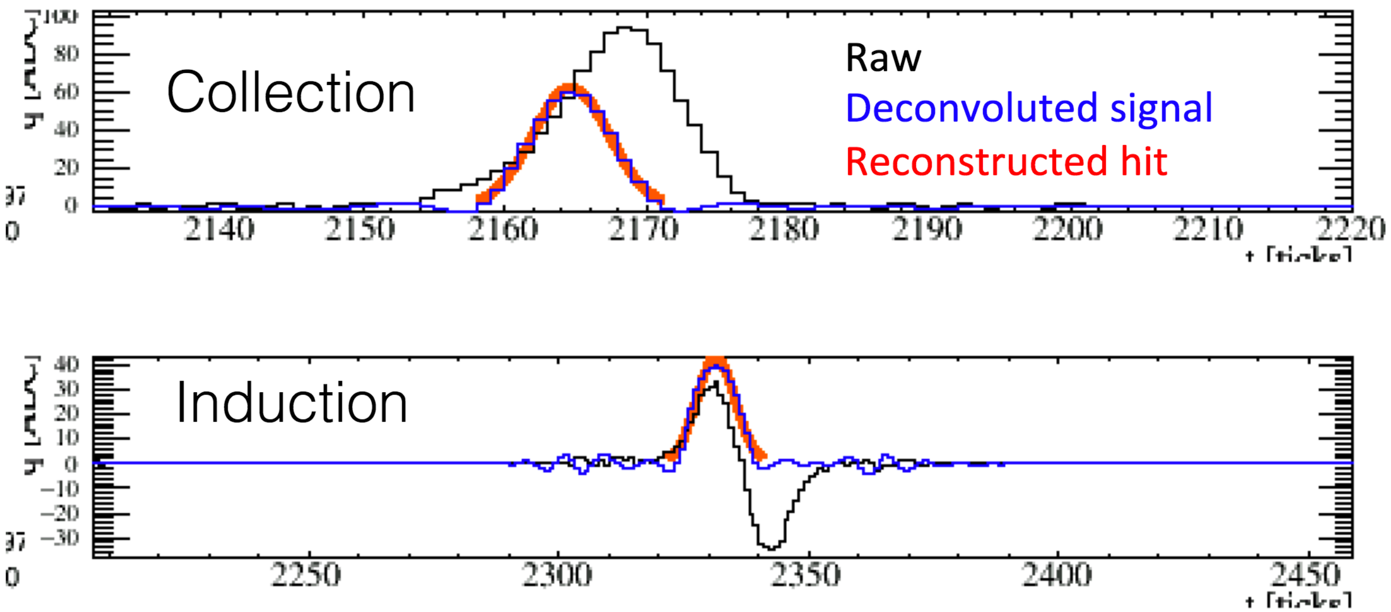
# DUNE far detector

- **70 kt** LArTPC, **1.5km** underground.
- High exposure in low-background environment.
- **Modular design:**
    - Four large detector modules.
    - Each consists of 200 individual TPCs.
    - Transformations necessary to combine data across multiple modules in 2D.

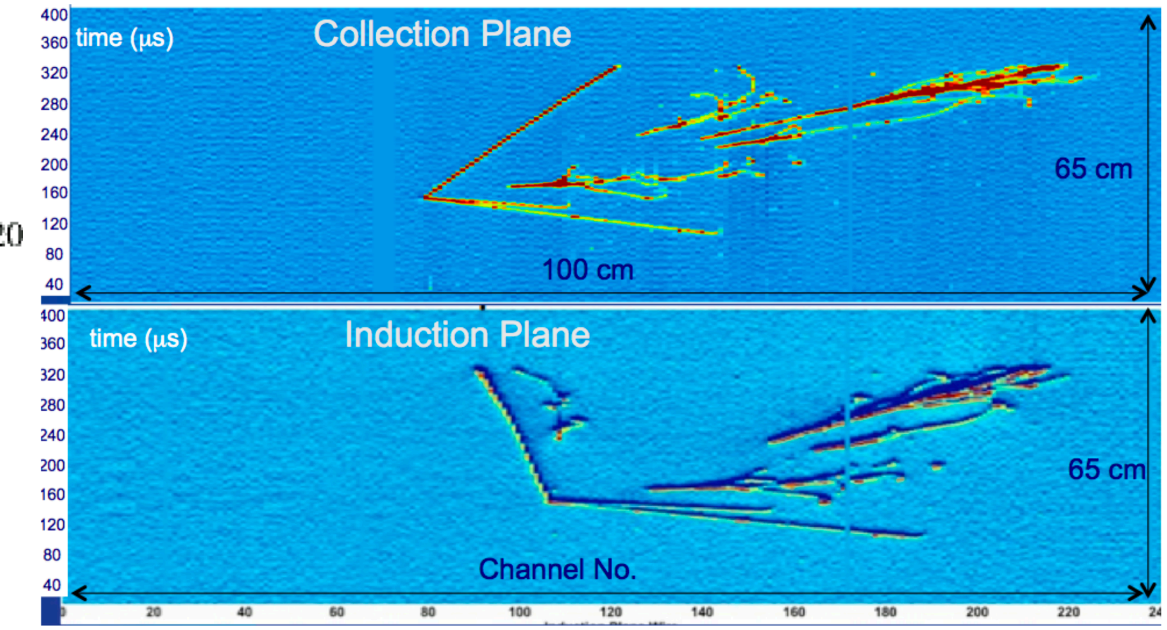


# Standard reconstruction chain




T. Yang (ICHEP 2016)

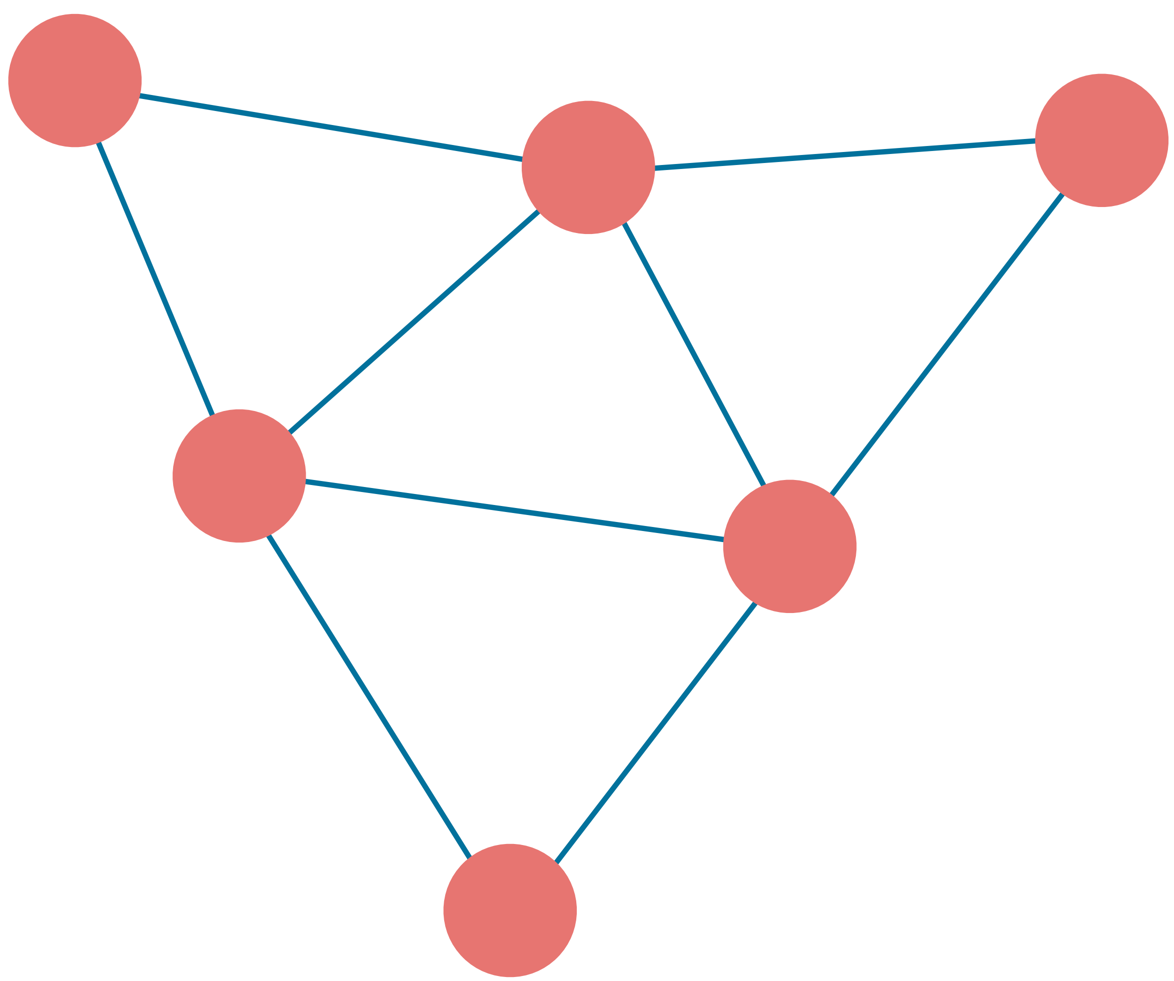



ArgoNeuT data event

- Raw TPC output is wire waveforms.
- Waveforms are then deconvolved and hit-finding is applied to produce Gaussian hits.
- Each wire plane forms a 2D image in the space of wire vs readout time.
- Three wire planes angled at -36°, 0°, 36° provide three 2D representations of the event.
- These 2D representations can be used to construct a 3D representation of the event.

# Graph neural networks

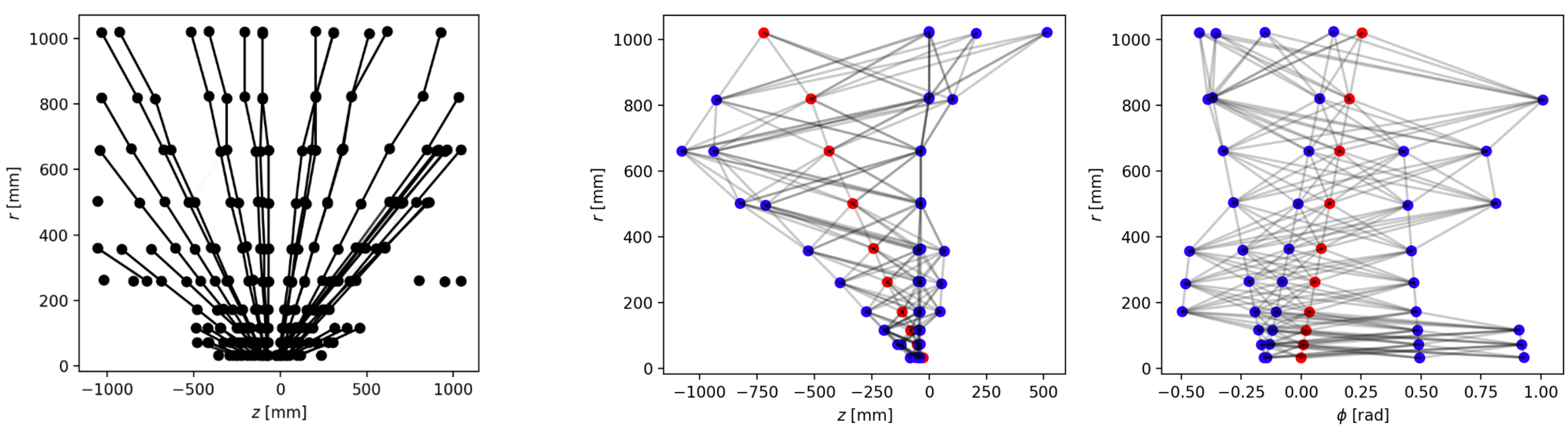
- Describe information structure as a **graph** represented by **nodes** and **edges**.



- **Nodes** are generalised as quantised objects with some arbitrary set of **features**.
- **Edges** describe the **relationships** between nodes.
- Perform convolutions on nodes and edges to learn relationships within the graph.
- Output is user-defined:
  - Classify nodes or edges.
  - Classify full graph.
  - Regression outputs.

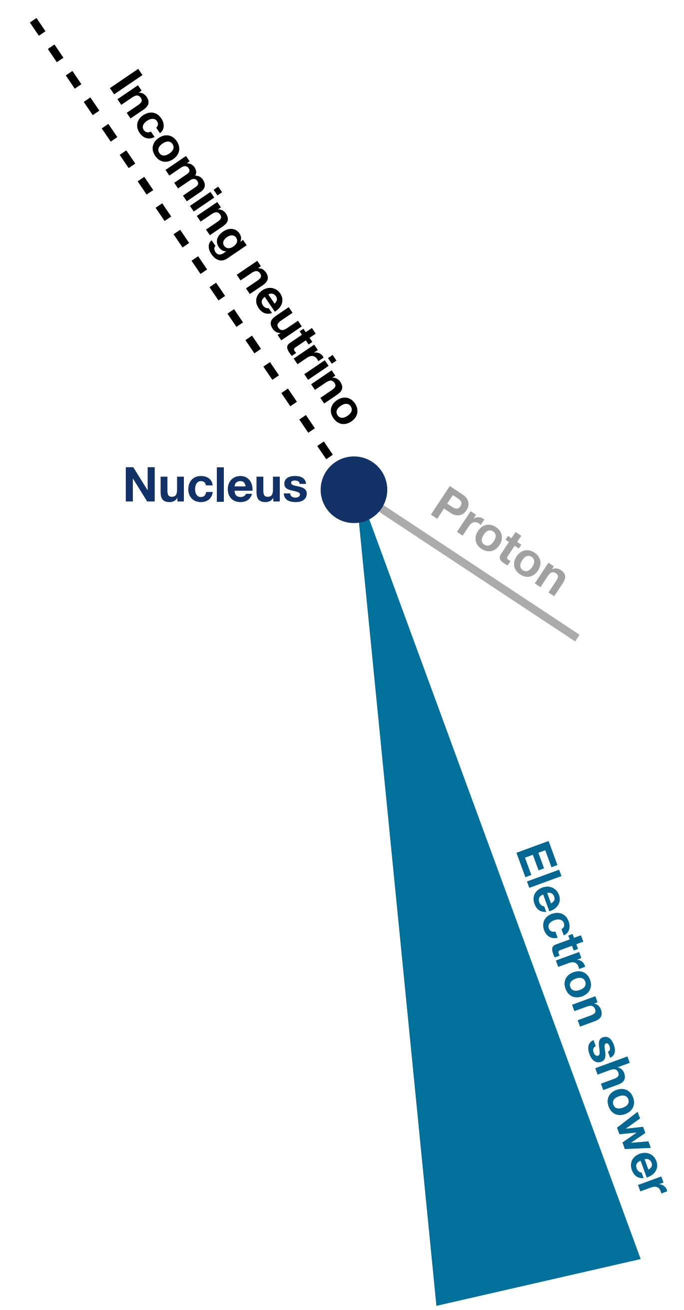
# Graph networks in HEP

- Investigating the use of **Graph Neural Networks (GNNs)** as an alternative to Convolutional Neural Networks (CNNs).
- Building on promising results from the **HEP.TrkX** collaboration using such methods for track reconstruction in the LHC world.
  - See talks by Daniel Murnane & Nicholas Choma in yesterdays session.
- Promising results from using GNNs in late-stage processing of segmentation CNNs.
  - See poster talk by Francois Drielsma later in this session.

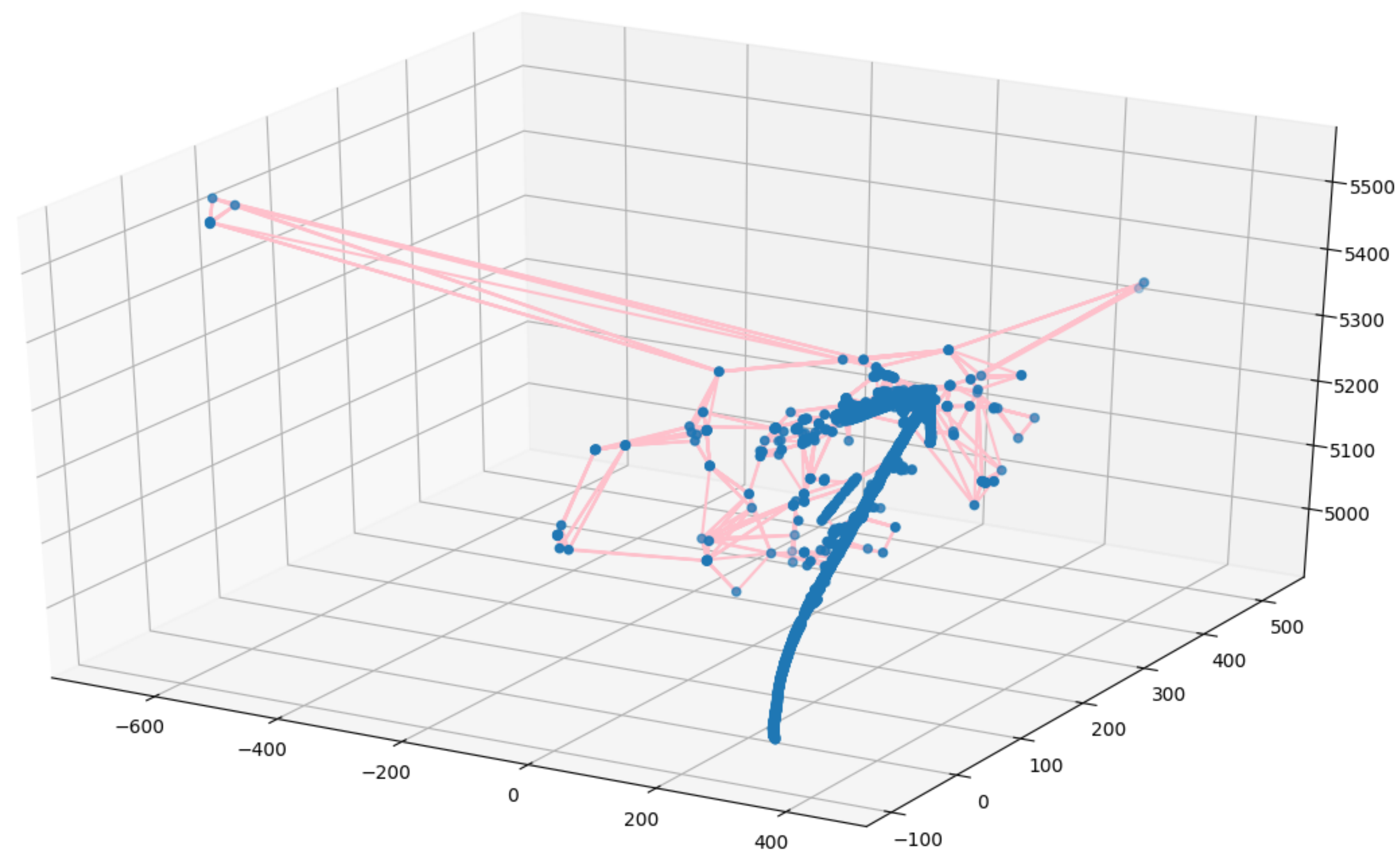


# Simulation

- Utilising two sets of simulation for these studies:
  - **Atmospheric neutrino interactions**
    - Higher in primary neutrino energy (typically ~tens of GeV).
    - Broad angular distribution.
    - Higher occupancy events.
  - **CCQE beam neutrino interactions**
    - Few-GeV energy.
    - Neutrinos travel along beam direction.
    - Typically “clean” interactions – primary lepton (e,μ) and minimal hadronic activity.

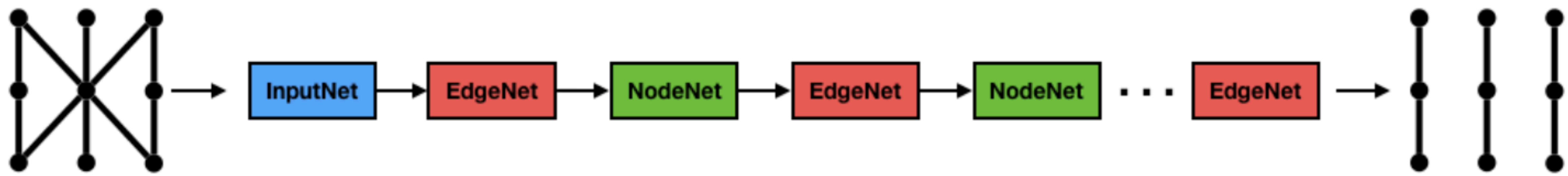


# Clustering




- First approach: cluster reconstructed spacepoints in 3D.
- Draw potential connections between 3D spacepoints.
- Classify edges as true or false based on whether the same underlying simulated particle was responsible for producing them.

# Message-passing networks

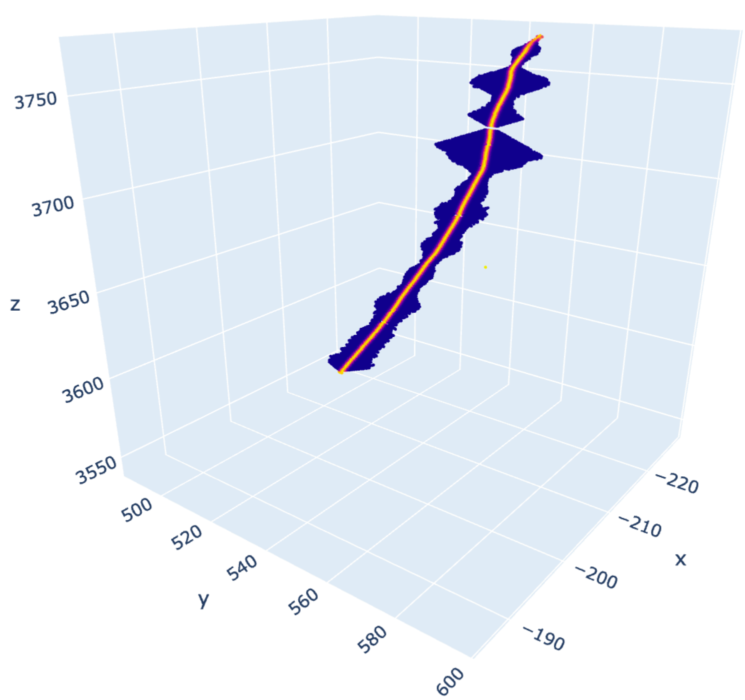


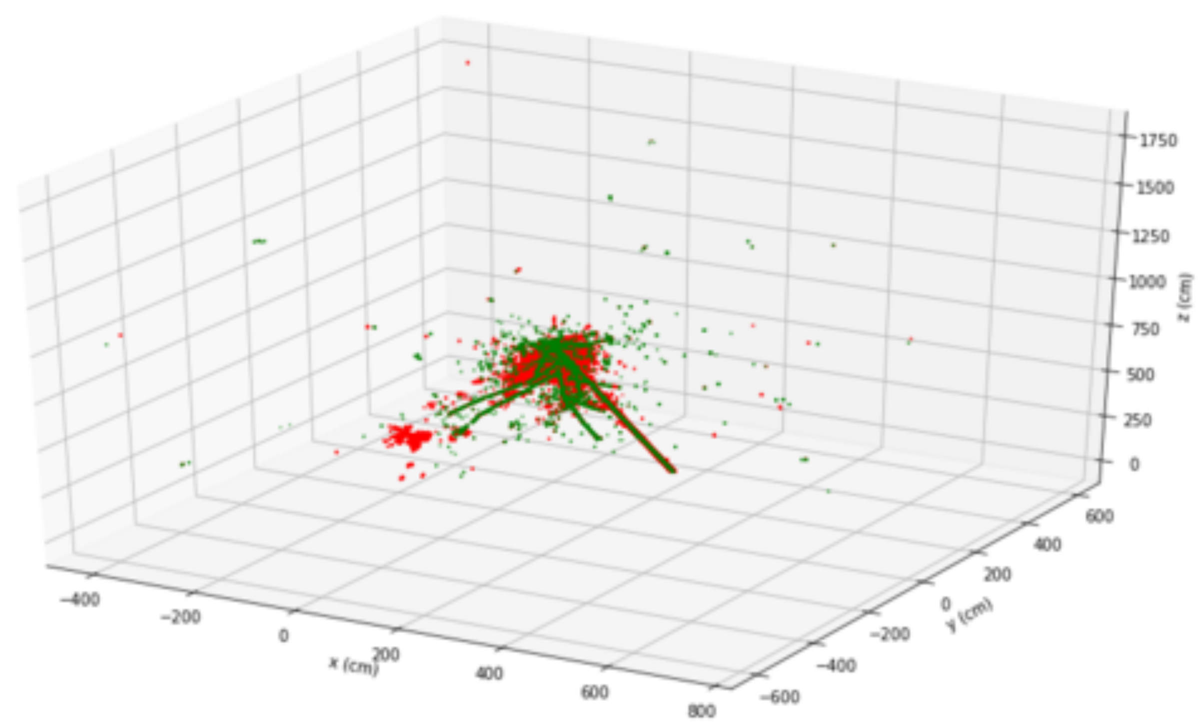

**arxiv:1810.06111**

- Message-passing network aggregates information from neighbouring nodes across edges to form new features on each node, utilising an attention mechanism to weight up useful edges.
- Repeat the same network multiple times in order for information to travel further across the graph over multiple iterations (the "message passing").
- **Edge classifier:**
  - Input for each node is the features of incoming and outgoing nodes.
  - Two multi-layer perceptrons, using Tanh and sigmoid activations.
  - Outputs sigmoid score on each edge.
- **Node classifier:**
  - Uses edge score to aggregate each node's features with incoming & outgoing edges as input.
  - Two multi-layer perceptrons with Tanh activation.
  - Produces new features for each node.

# Spacepoint reconstruction

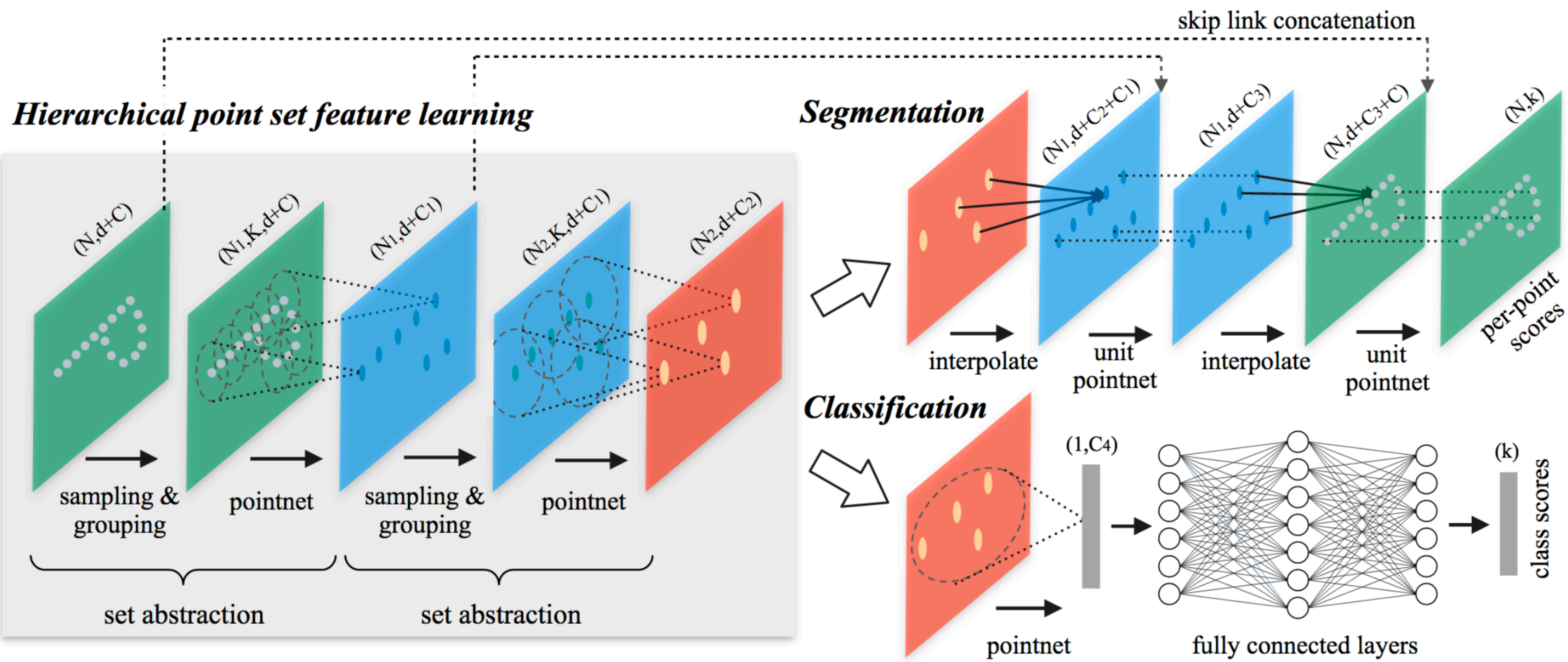
- Moving from three 2D representations of an energy deposition to one 3D representation is a noisy procedure.
- Early attempt: utilise **graph node** classification to retain good 3D representations and remove spurious ones.
- Construct graph edges using k-nearest-neighbour (kNN) technique.



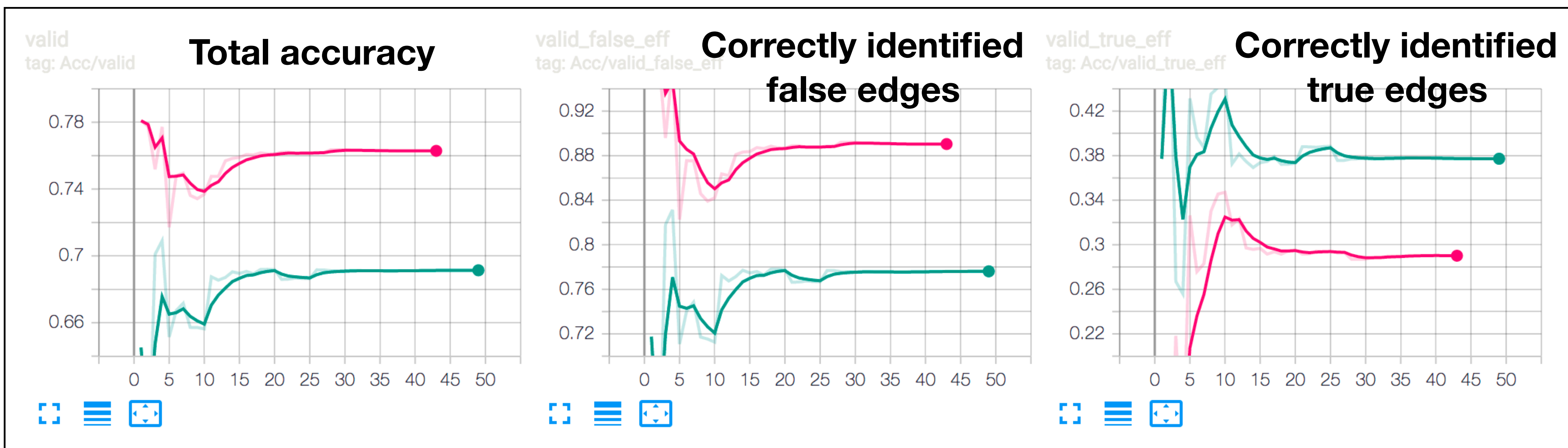
# Spacepoint clustering

- Investigated use of **PointNet++** spacepoint graph network (arxiv:1706.02413).
  - This network is specifically designed to operate on point clouds.
  - Utilises **set abstraction** to aggregate local features, similar to a U-net for CNNs.
- PyTorch implementation of up & down-sampling too slow for large point clouds.

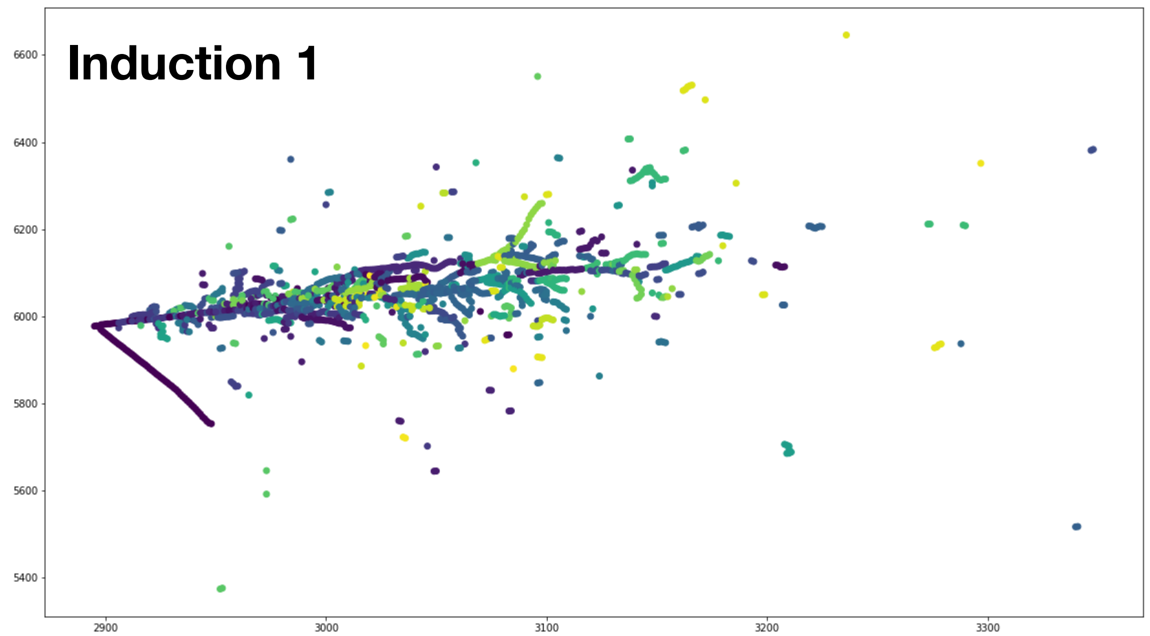


# 2D approaches

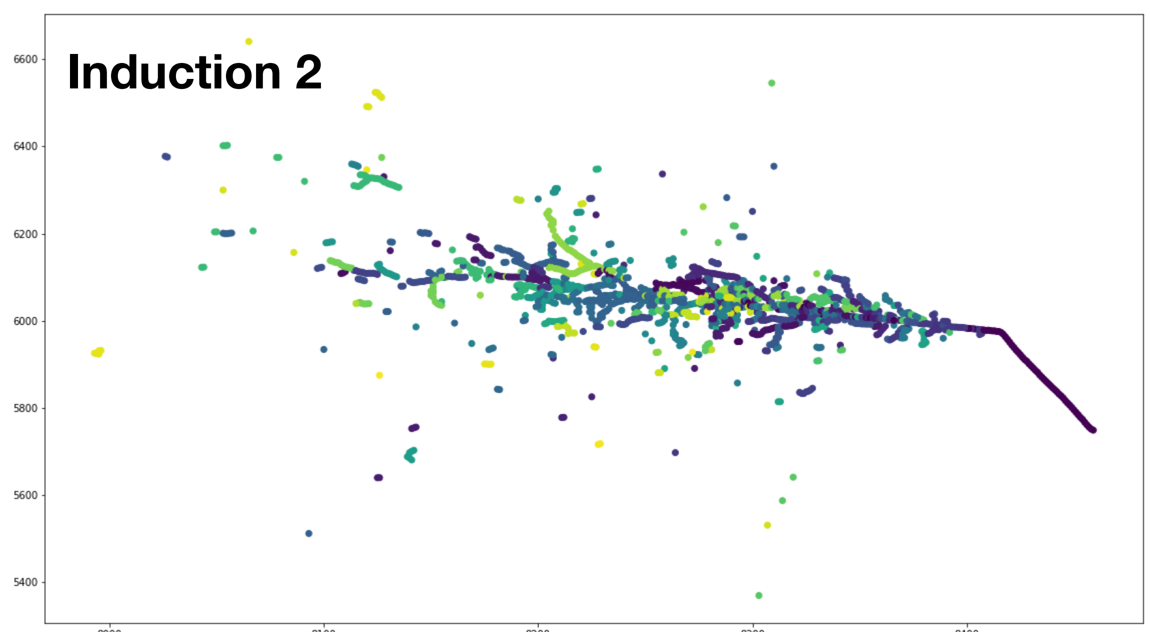
- The 3D approaches explored were not found to be effective.
  - Only learn marginally above noise level.
- Next step: investigate reconstruction of interactions in 2D representations.
  - Conceptually closer to LHC approach.
  - Can leverage structure of detector to sparsify number of edges and reduce graph size.

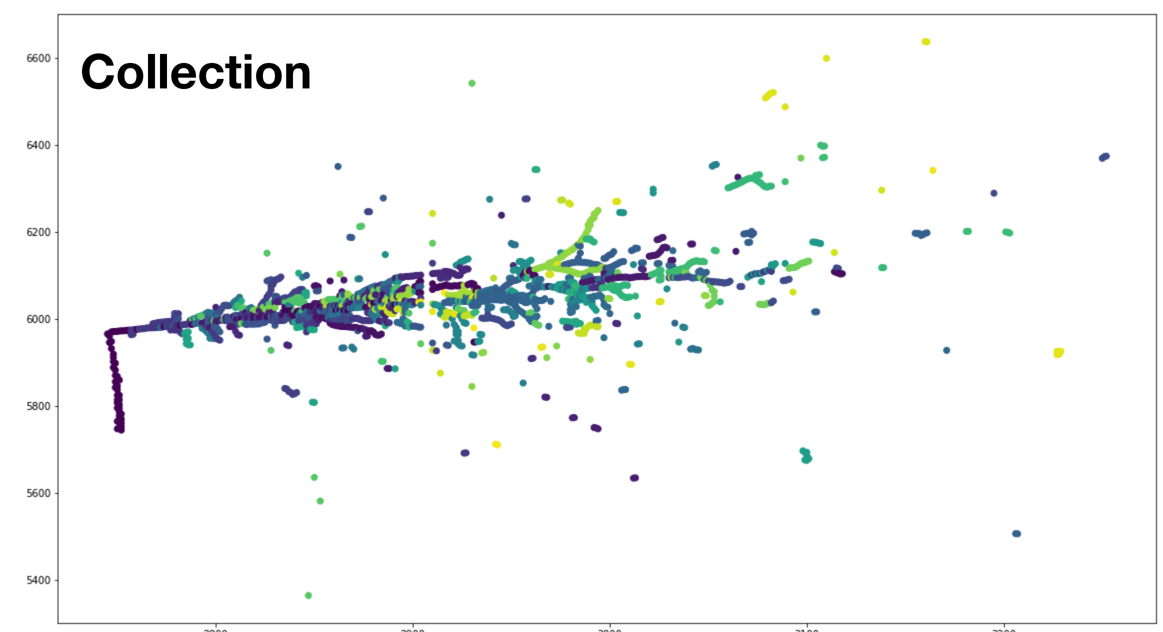


# 2D reconstruction



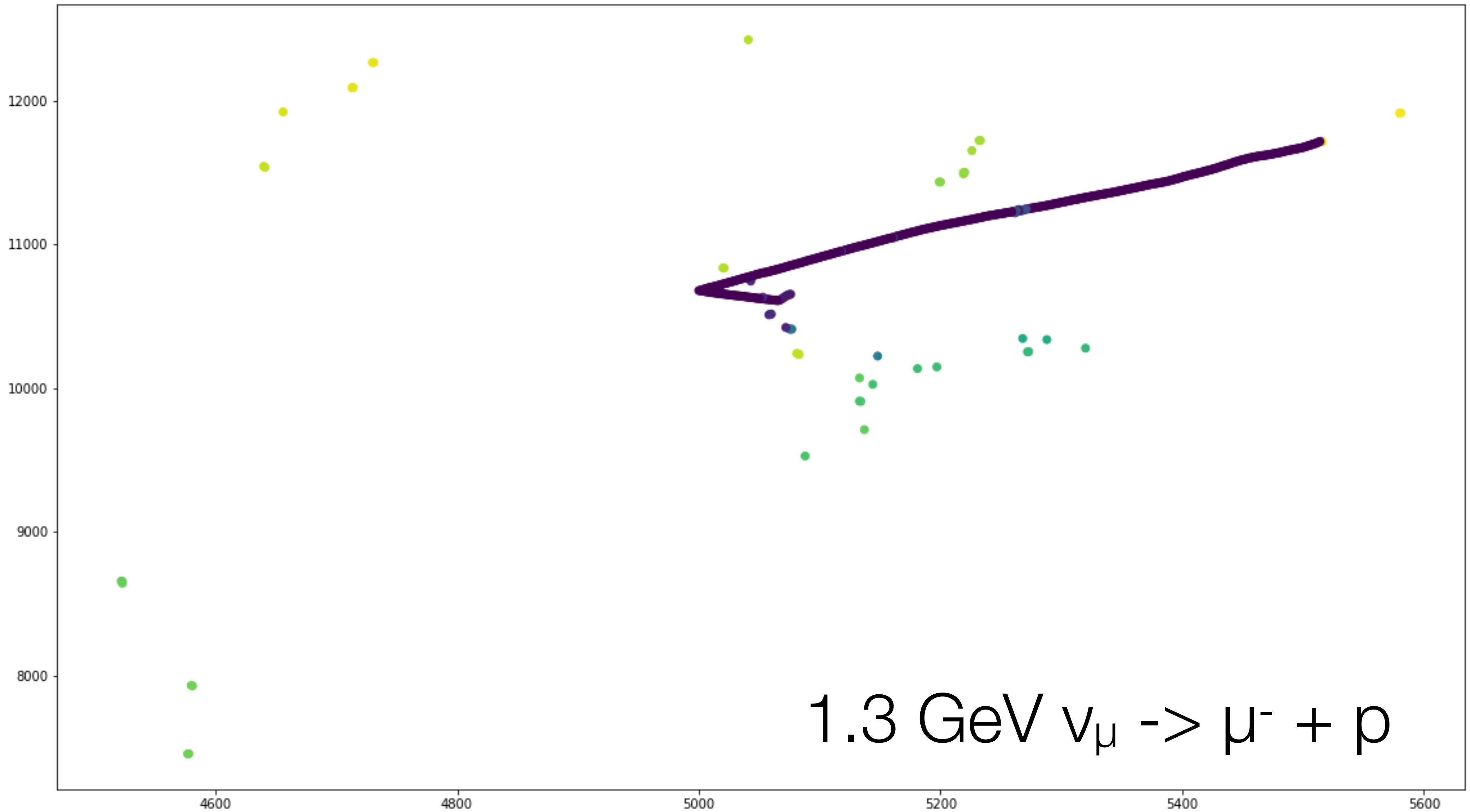

- Alternate approach: start with 2D representation and build up using graph network.
- Colour coded according to **true simulated particle**.
- Three 2D representations of the same 3D interaction.



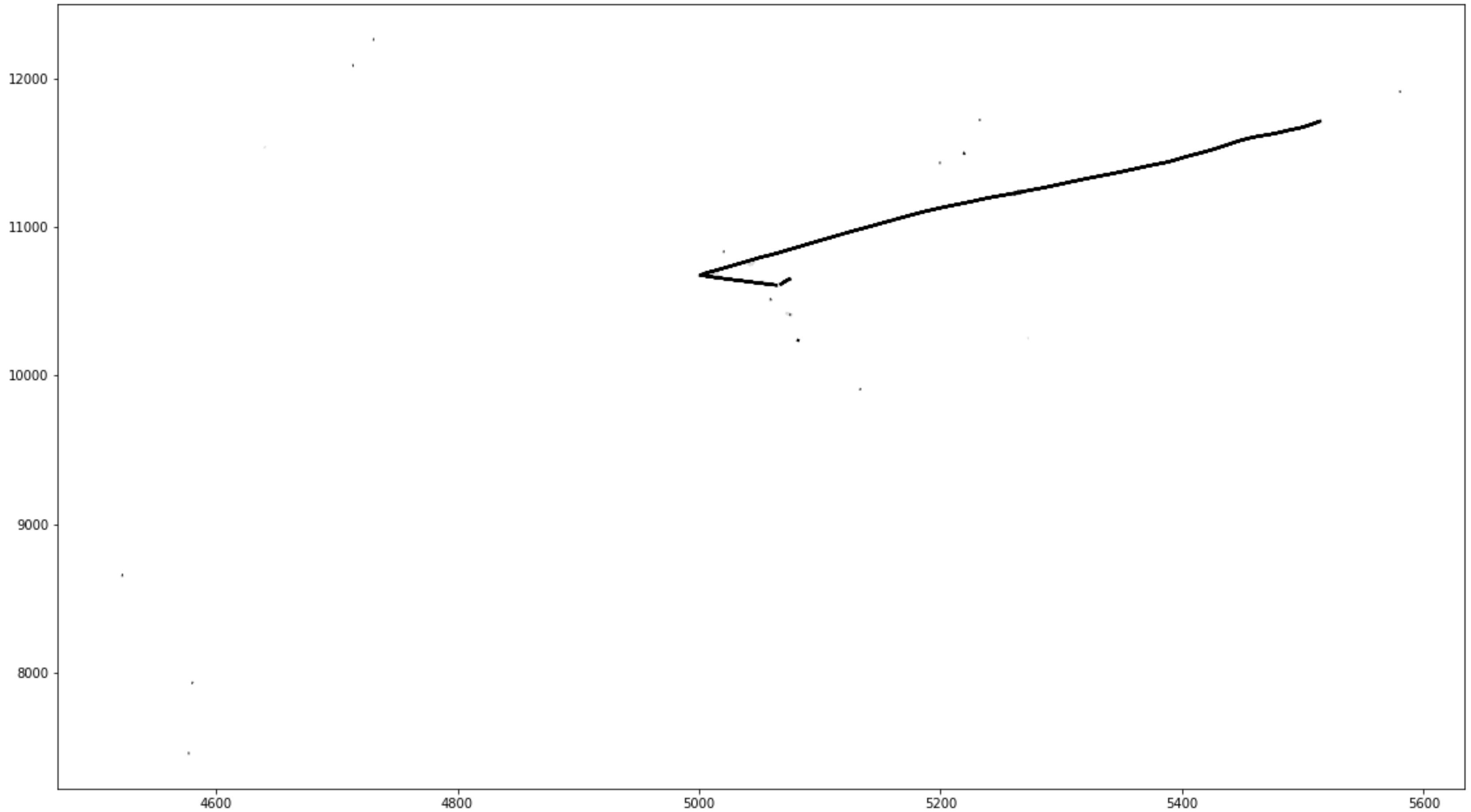
# $\nu_\mu$ graph construction

- Connect hits that are adjacent in wire and time with **potential edges**.
- Potential edges drawn in grey between nodes.



# νμ graph construction

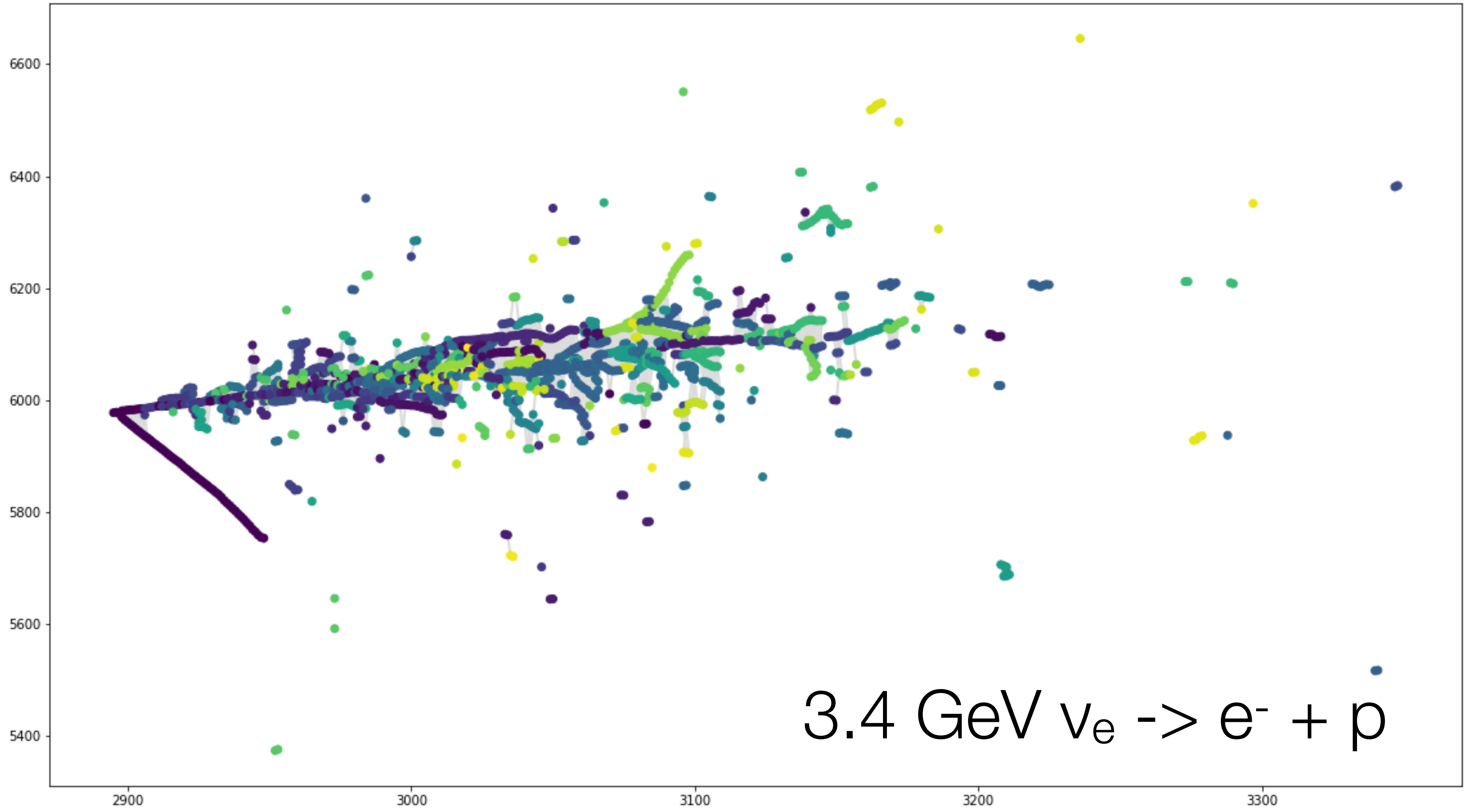
- Potential graph edges formed for **hits in close proximity** (5 wires & 50 time ticks).
- Potential edges then classified as **true** or **false** as an objective for learning.




- Edges are classified as true if the same particle was responsible for the two hits in the underlying simulation.
- Colour coding:
  - **False edges**
  - **True edges**

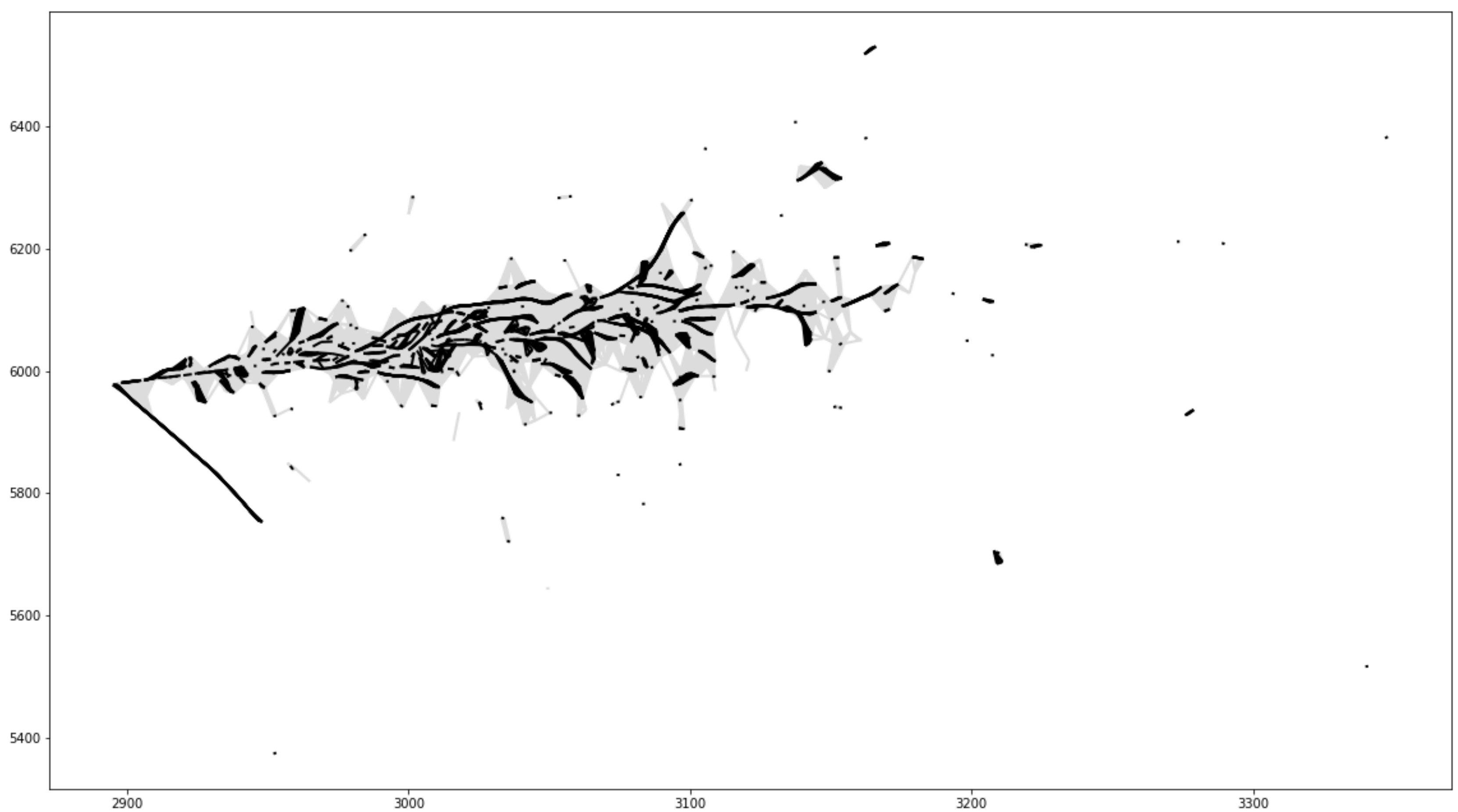
# $\nu_e$ graph construction

- Connect hits that are adjacent in wire and time with **potential edges**.
- Potential edges drawn in grey between nodes.



# $\nu_e$ graph construction

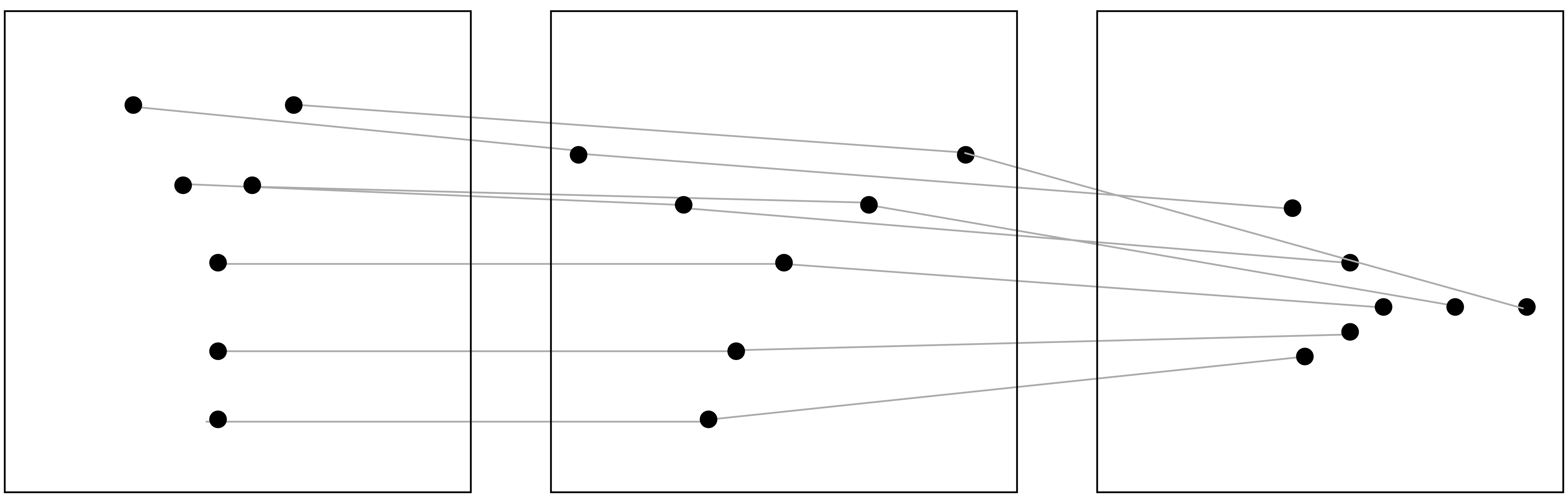
- Potential graph edges formed for **hits in close proximity** (5 wires & 50 time ticks).
- Potential edges then classified as **true** or **false** as an objective for learning.




- Edges are classified as true if the same particle was responsible for the two hits in the underlying simulation.
- Colour coding:
  - **False edges**
  - **True edges**

# 2D hit matching

- In addition to reconstruction within each wire plane, can also consider 3D representations by matching 2D energy depositions between planes.
  - Work backwards from 2D hits to trajectory of underlying 3D simulated particle to draw true associations.
- Benefit of this approach: utilise heterogeneous graph nodes (ie. LArTPC optical detectors) to match between different detector technologies.
- Message-passing *between* planes may aid with clustering *within* each plane.



University of CINCINNATI

# Summary

- Investigating development of GNNs for low-level reconstruction in LArTPCs.
- Graph approaches efficient on sparse data in its native form.
- 3D reconstruction applications (clustering, spacepoint reconstruction) not found to be effective.
  - Message-passing is not optimal for densely connected 3D graph.
  - Point cloud-based techniques are not computationally efficient for large point clouds.
- Developing 2D reconstruction:
  - Particle clustering in 2D views.
  - Clustering between 2D views & construction of higher-level objects.