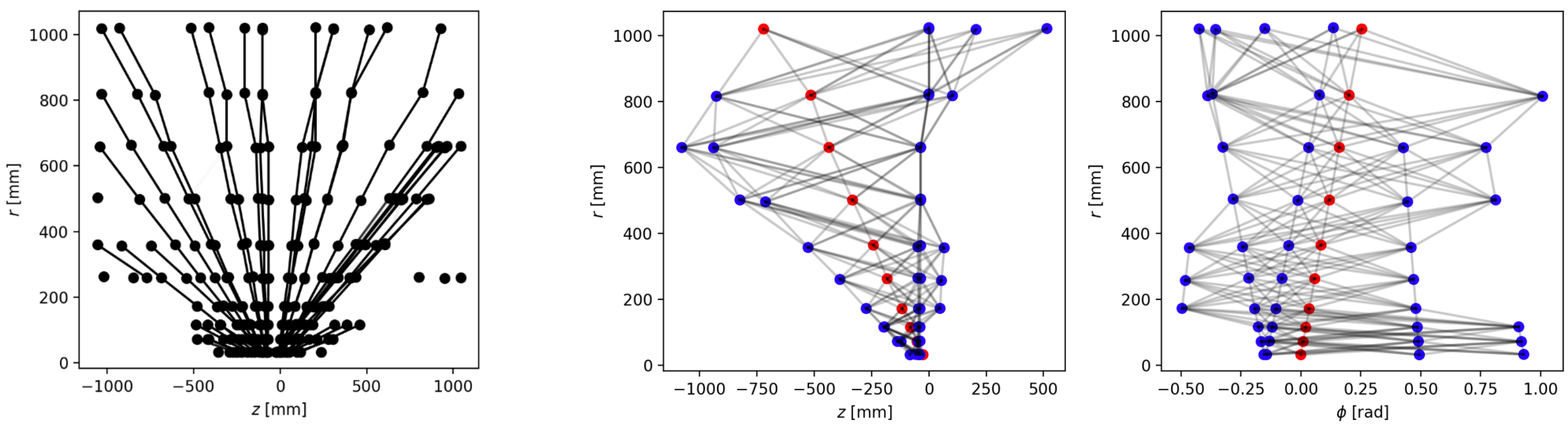
Graph Neural Network for Object Reconstruction in Liquid Argon Time Projection Chambers

Jeremy Hewes, for the Exa.TrkX collaboration
vCHEP 2021
19th May 2021

Graph neural networks

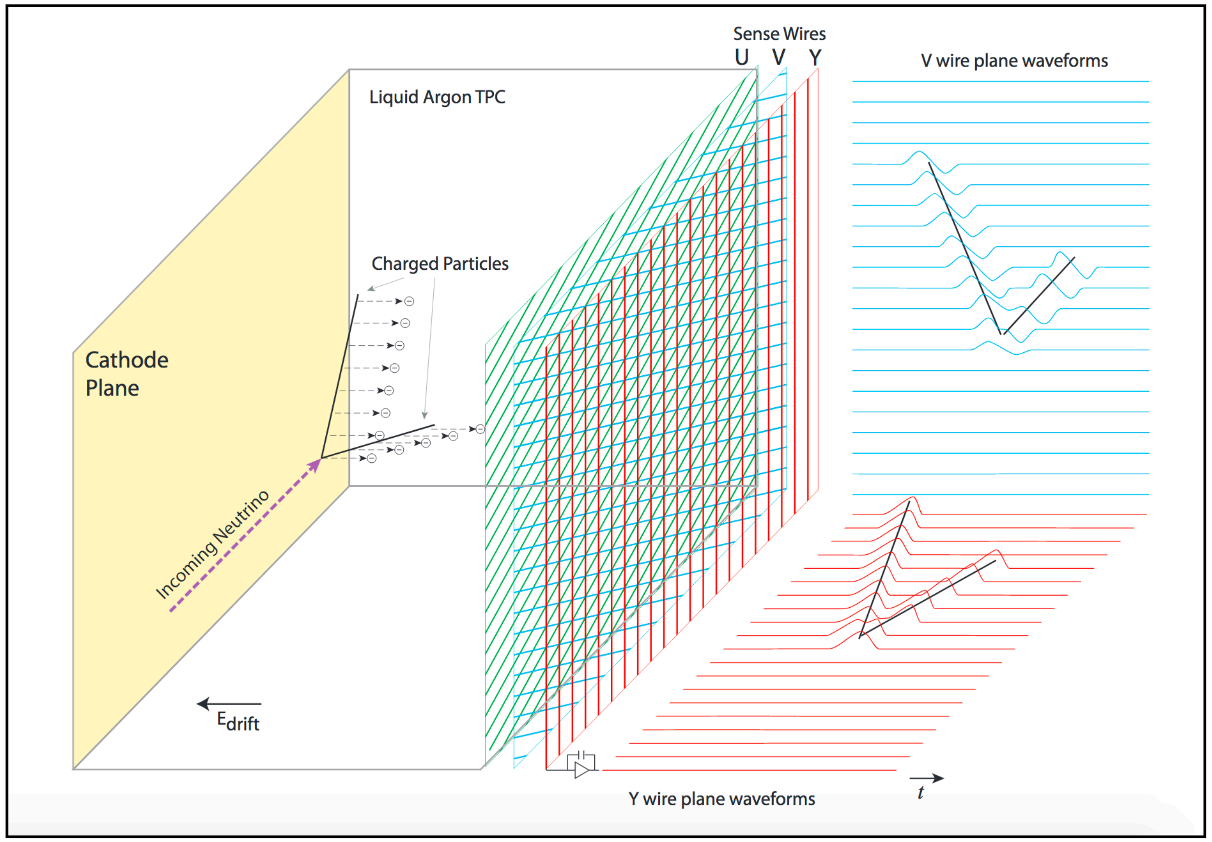
- Investigating the use of **Graph Neural Networks (GNNs)** as an alternative to Convolutional Neural Networks (CNNs).
- Describe information structure as a **graph** represented by **nodes** and **edges**.
- Building on promising results from the **HEP.TrkX** collaboration using such methods for track reconstruction in the LHC world (arxiv:2103.06995).



Liquid Argon TPCs

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

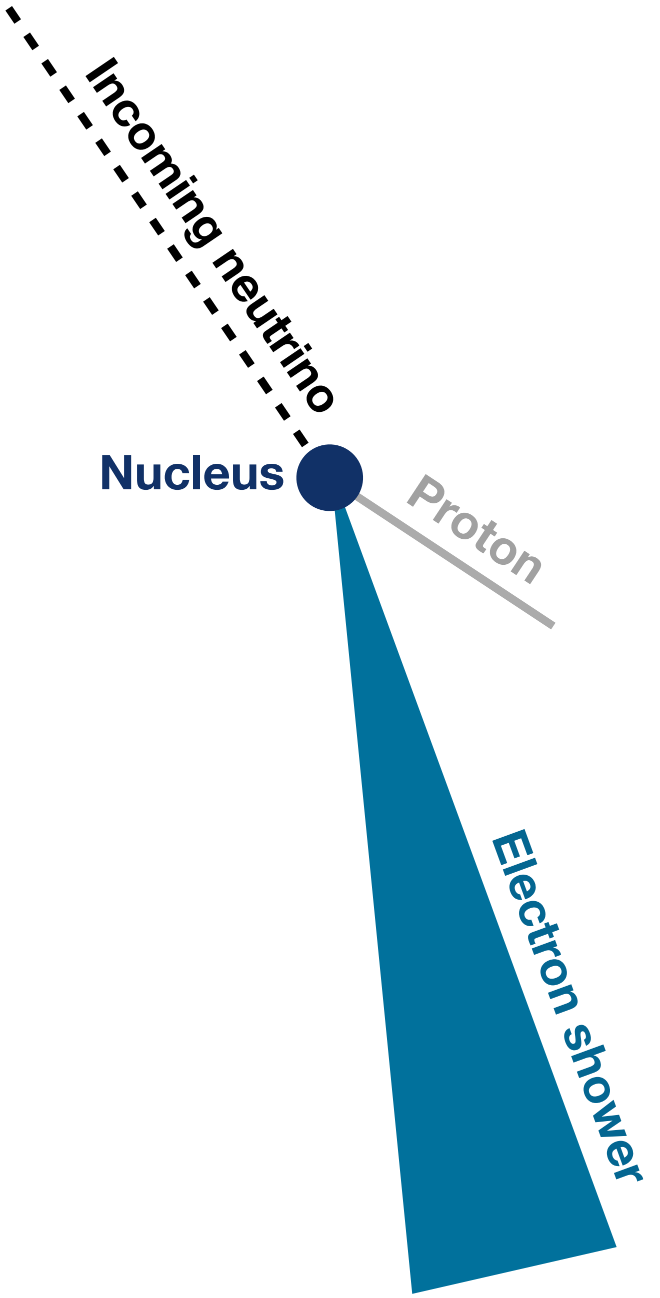
- Charged particles ionize liquid argon as they travel.
- Ionisation electrons drift due to HV electrode field, and are collected by anode wires.
- Wire spacing ~3mm – produce **high-resolution images**.



University of CINCINNATI

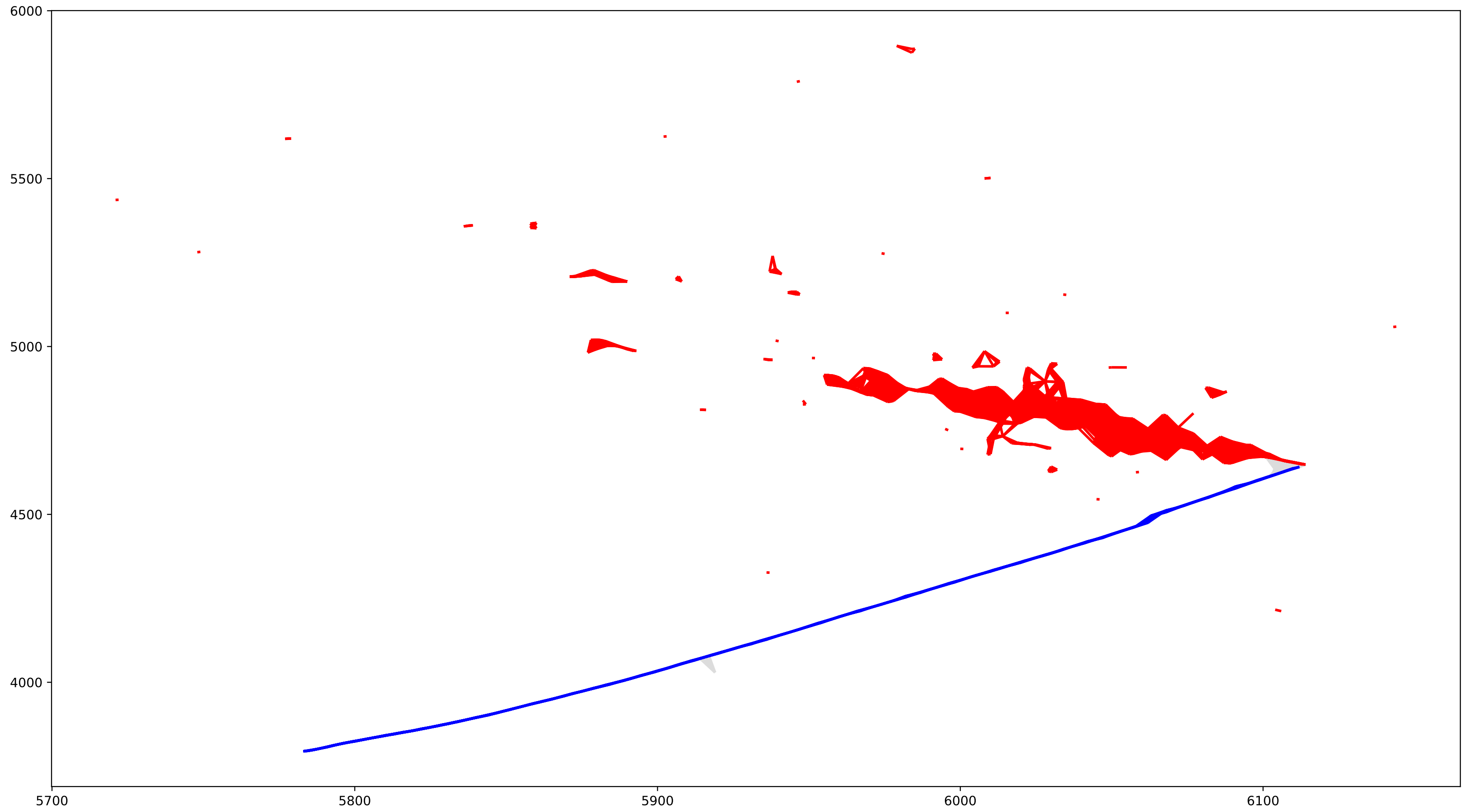
Simulation

- Physics problem: **reconstruct neutrino interactions in a LArTPC by classifying detector hits and grouping them into objects**.
- Use **CCQE beam neutrino interactions**
  - Few-GeV energy neutrinos.
  - Neutrinos travel along beam direction.
  - Typically “clean” interactions – primary lepton (e,μ) and minimal hadronic activity.
- Minimal sim/reco chain:
  - GENIE/Geant4 simulation.
  - Detector simulation.
  - Wire deconvolution & hit finding.



Graph construction

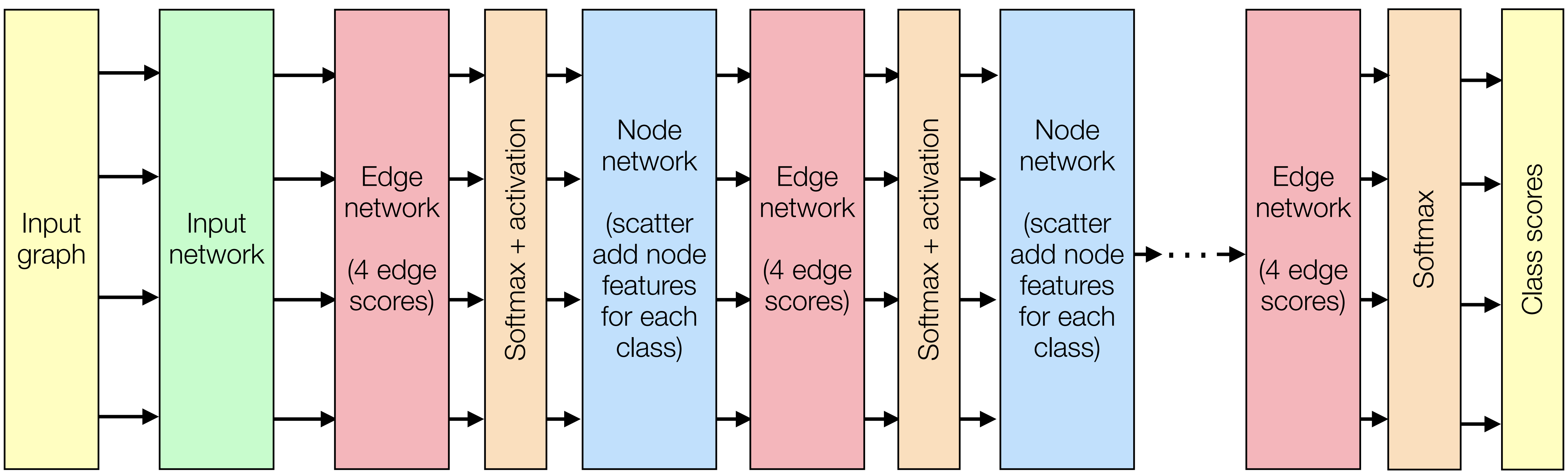
- Potential graph edges formed for **hits in close proximity** (5 wires & 50 time ticks).
- Potential edges then classified as **hadronic**, **muon**, **shower** or **false** as an objective for learning.



- Edges are classified as **false** if the two hits were not produced by the same particle in the underlying simulation.
- **Muon** edges are hits produced by the primary muon, **shower** edges by the primary electron, and **hadronic** edges are the remainder.

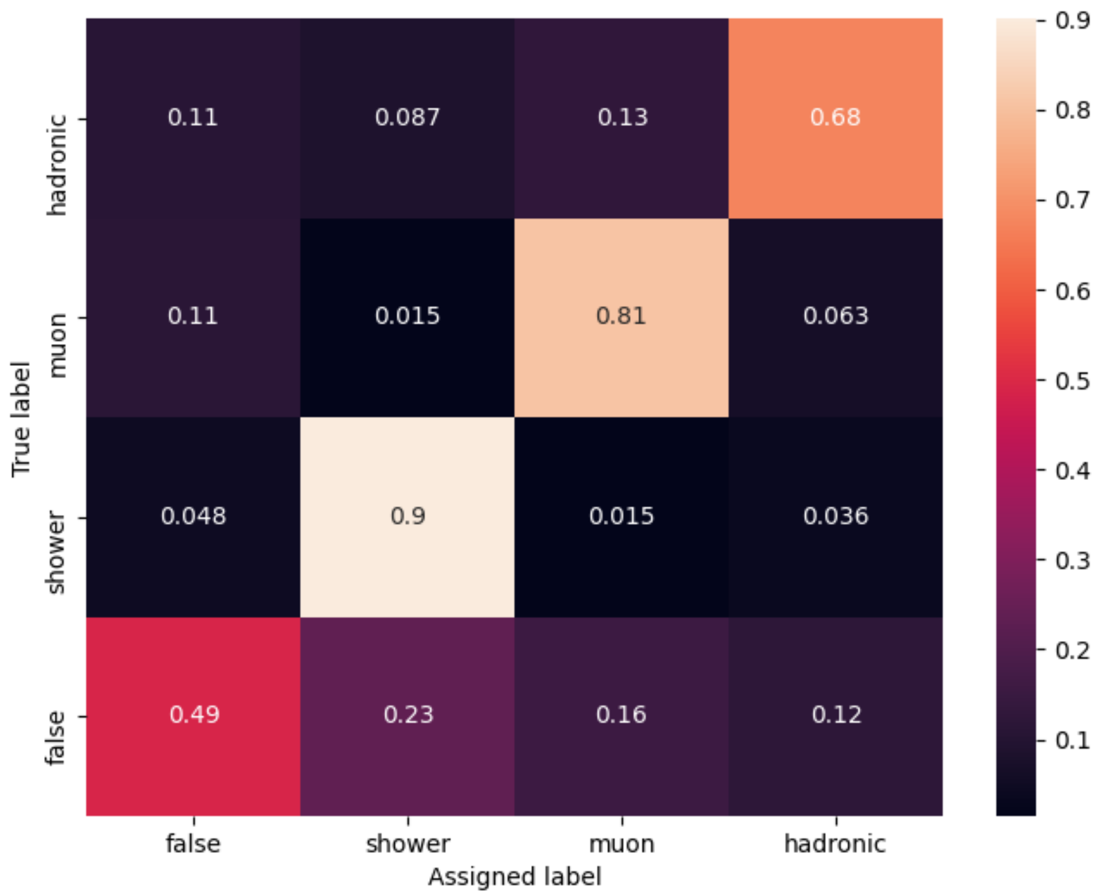
Multihead attention message-passing network

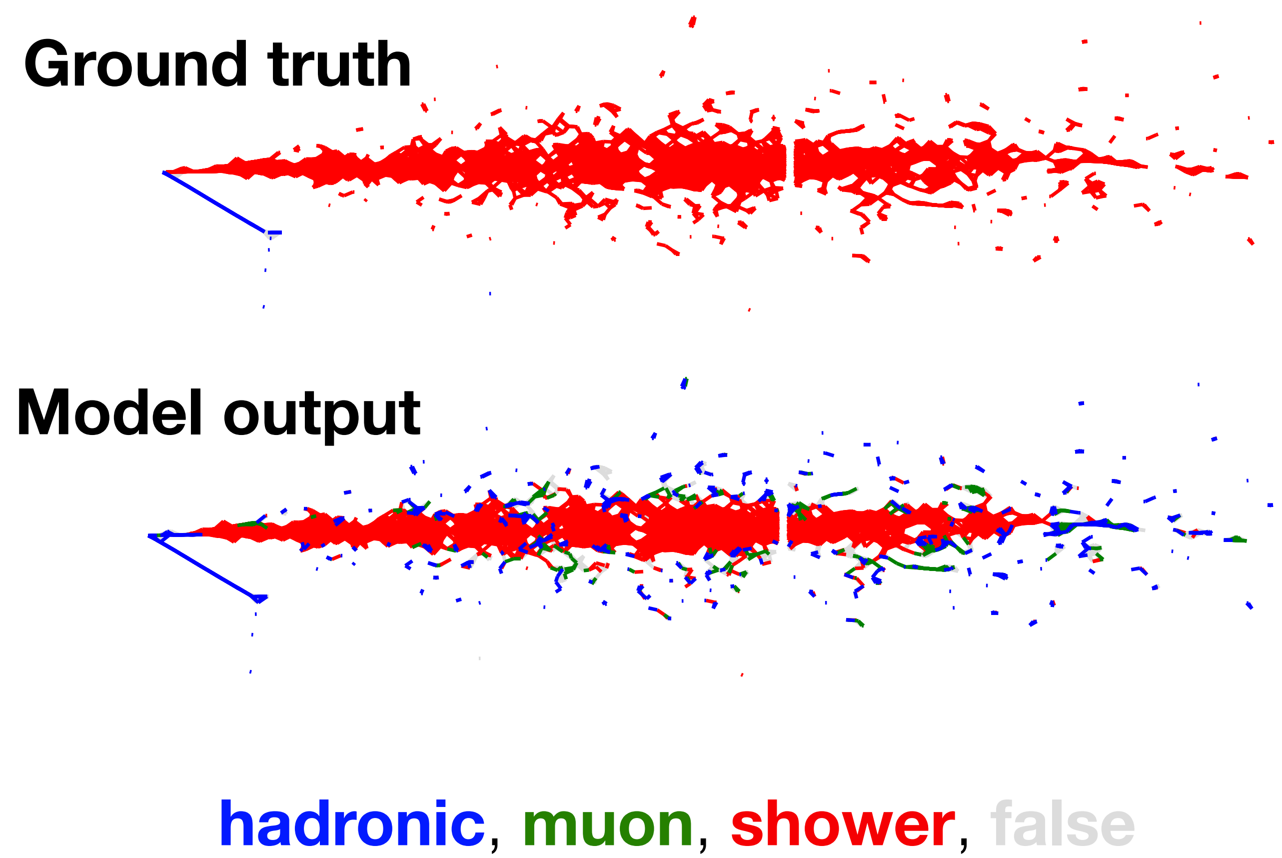
- Build on Exa.TrkX binary edge classifier, to **classify graph edges to determine the relationships between detector hits**.
  - Pass messages + form node features independently for each class.
  - Produce 4 edge attention scores on each edge.
  - Take the softmax of those edges **with each iteration**.
  - If an edge is strongly shower-like, the track-like classes will be weighted down accordingly.



2D edge classification network

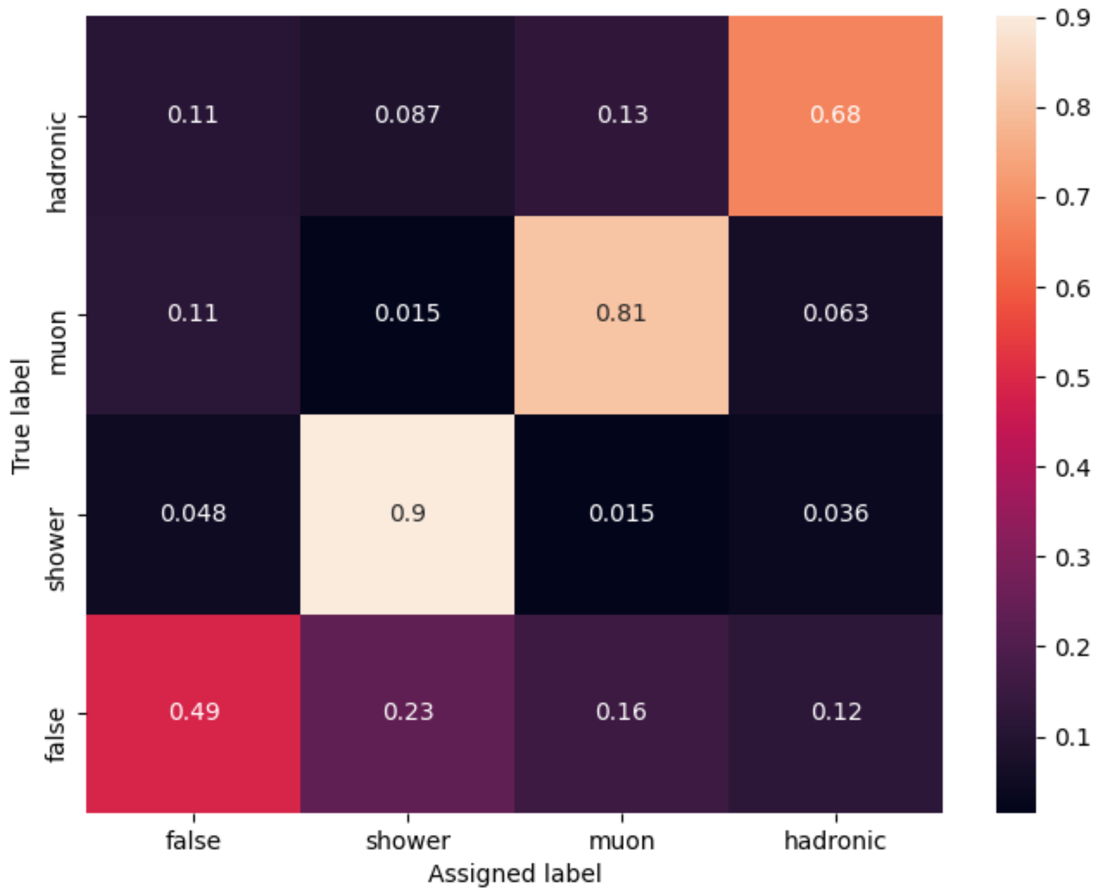
- Current iteration achieves 84% accuracy in classifying graph edges.
  - Performs well on showers, but still room for improvement in tracks.
  - See arxiv:2103.06233.

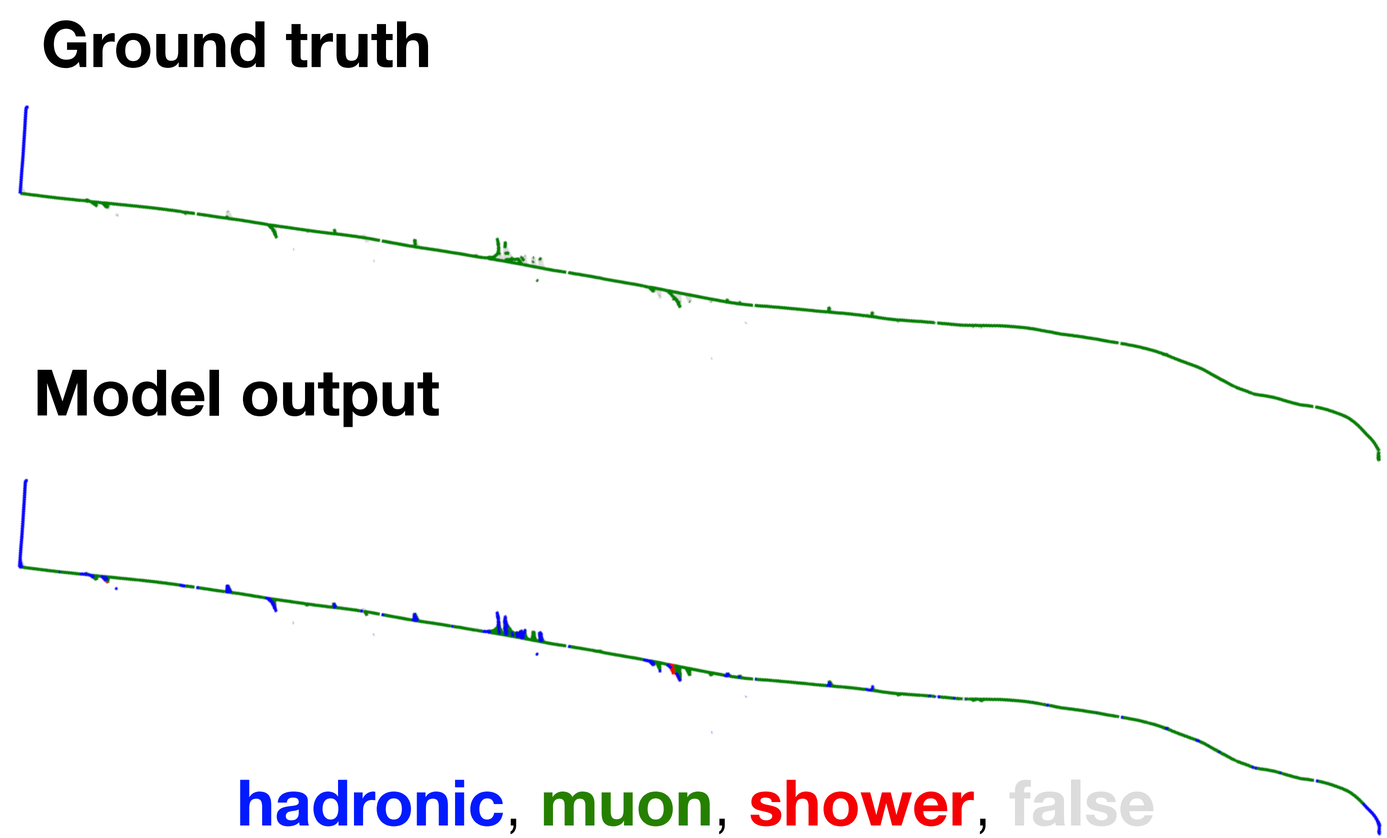




2D edge classification network

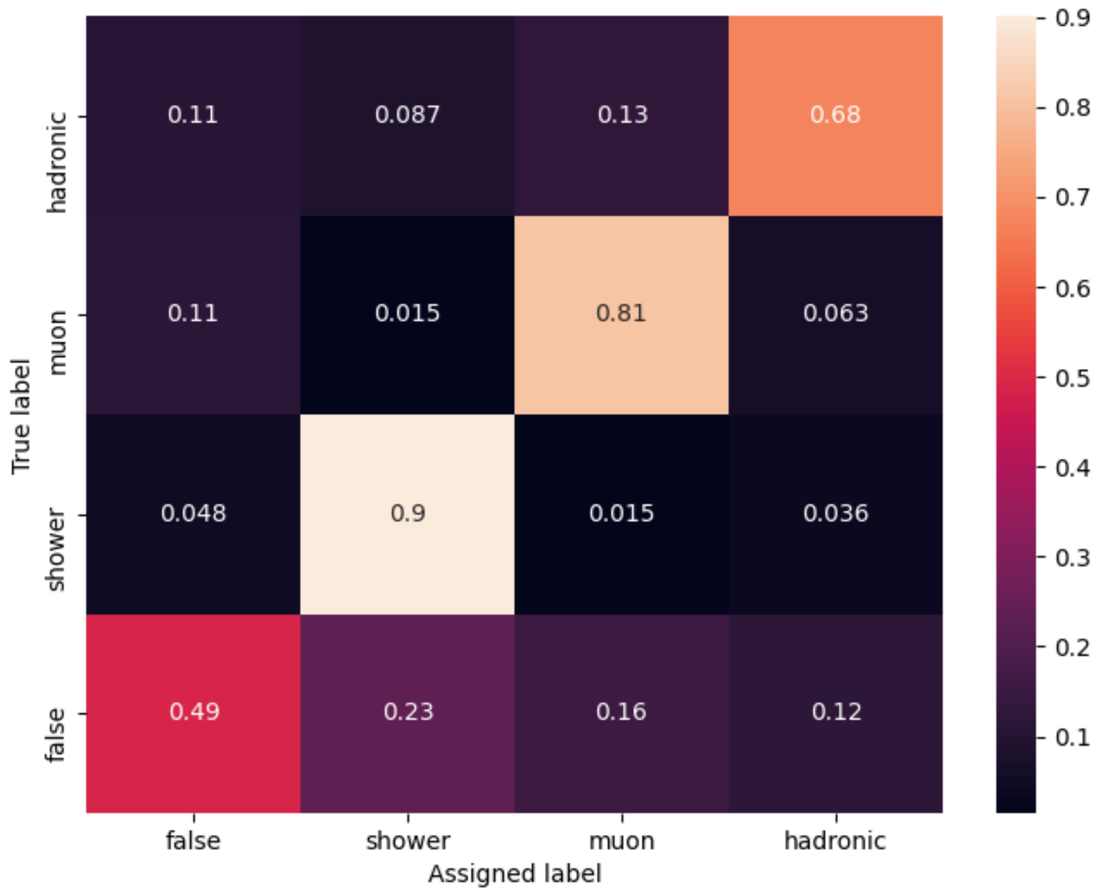
- Current iteration achieves 84% accuracy in classifying graph edges.
  - Performs well on showers, but still room for improvement in tracks.
  - See arxiv:2103.06233.

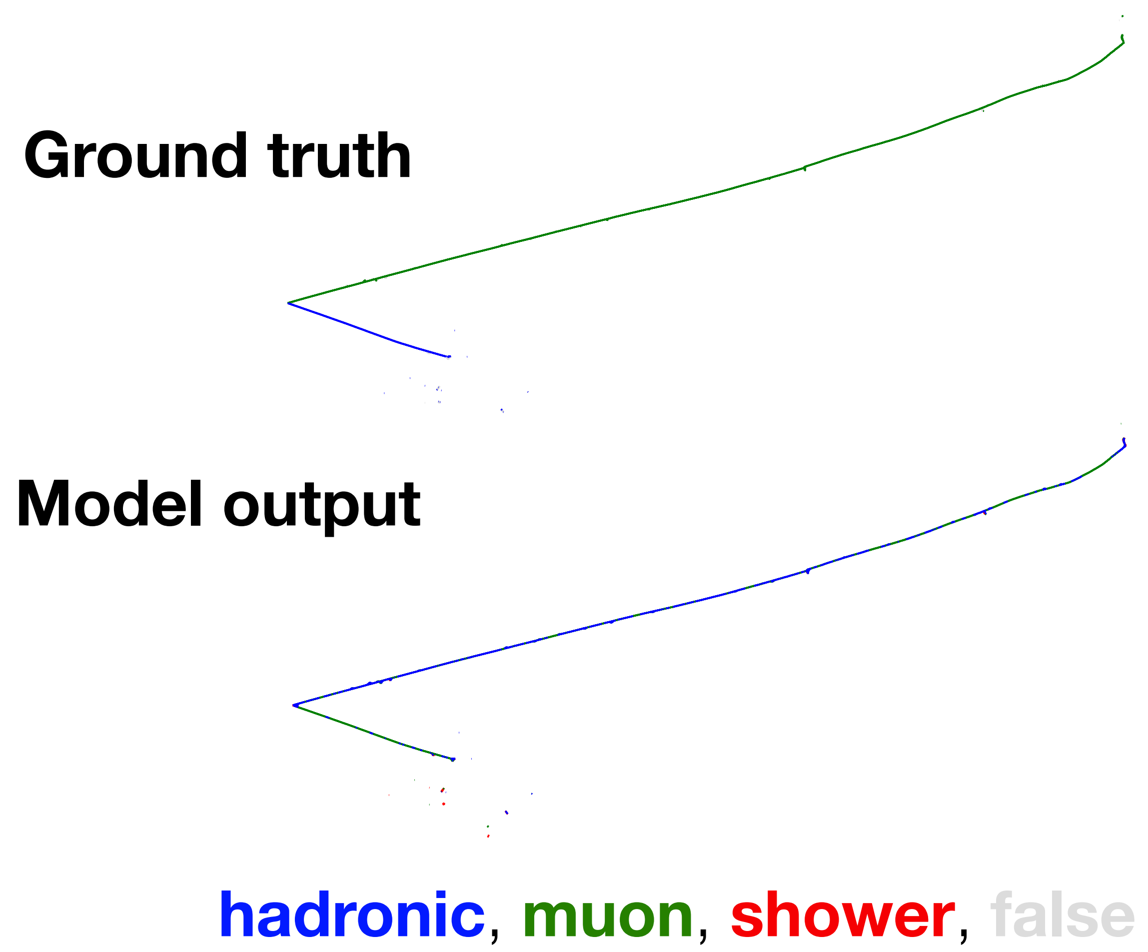




2D edge classification network

- Current iteration achieves 84% accuracy in classifying graph edges.
  - Performs well on showers, but still room for improvement in tracks.
  - See arxiv:2103.06233.



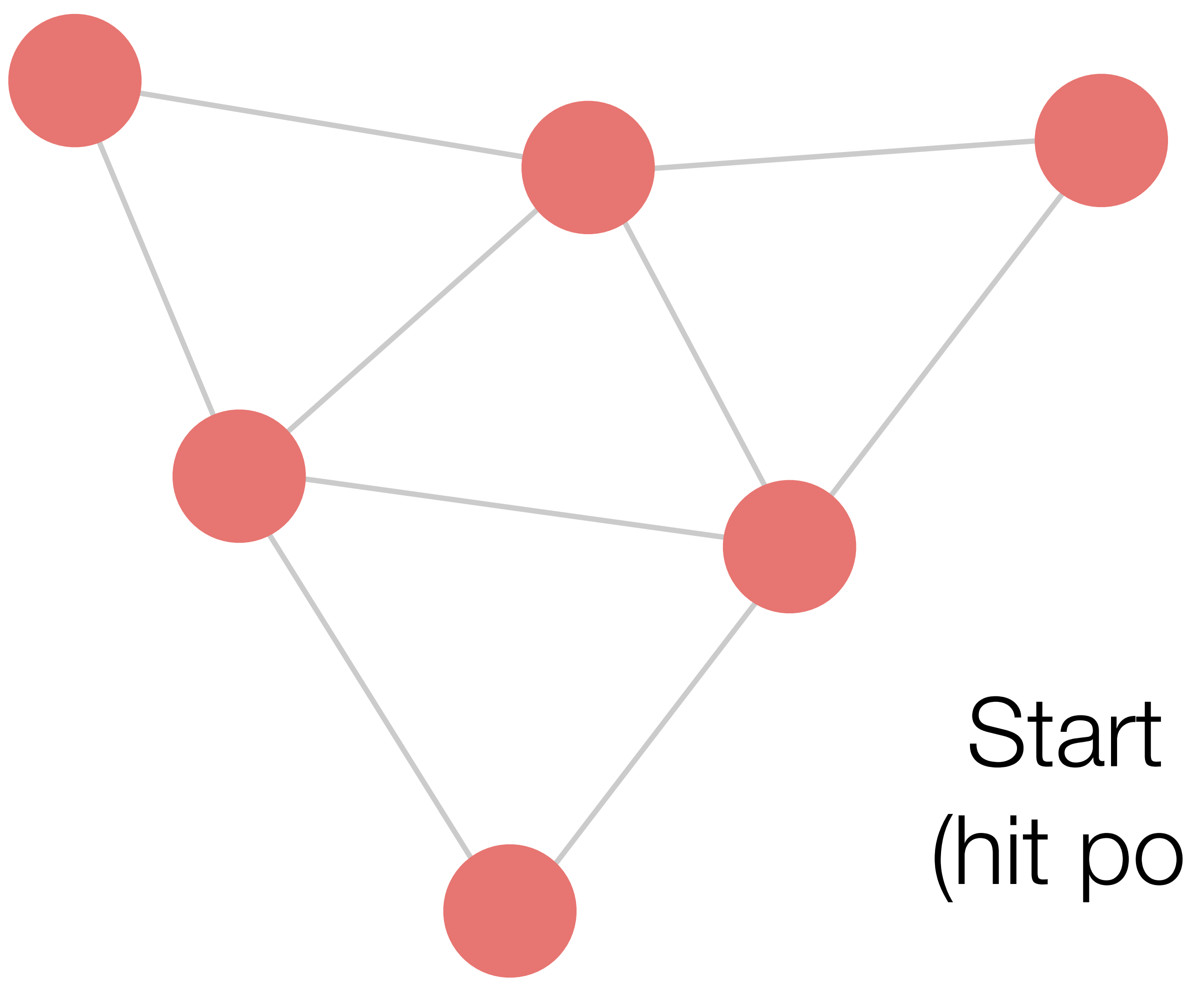


Future plans

- This first version of our model performs well at reconstructing particle types in simulated LArTPC neutrino interactions.
- Edge classification works well for layered LHC detectors, but less well-suited to the problem of clustering hits into dense objects.
  - Need a scheme to collapse disparate classified edges into objects.
  - Incorporate concepts from instance segmentation for object-finding.
- Move beyond simple CCQE interactions to more complex event topologies
  - More sophisticated definitions of ground truth.
    - More granular taxonomy: Michel electrons, e/γ showers, δ rays, π, κ.
- 3D-aware model which passes information between planes to ensure 2D representations are consistent with each other.

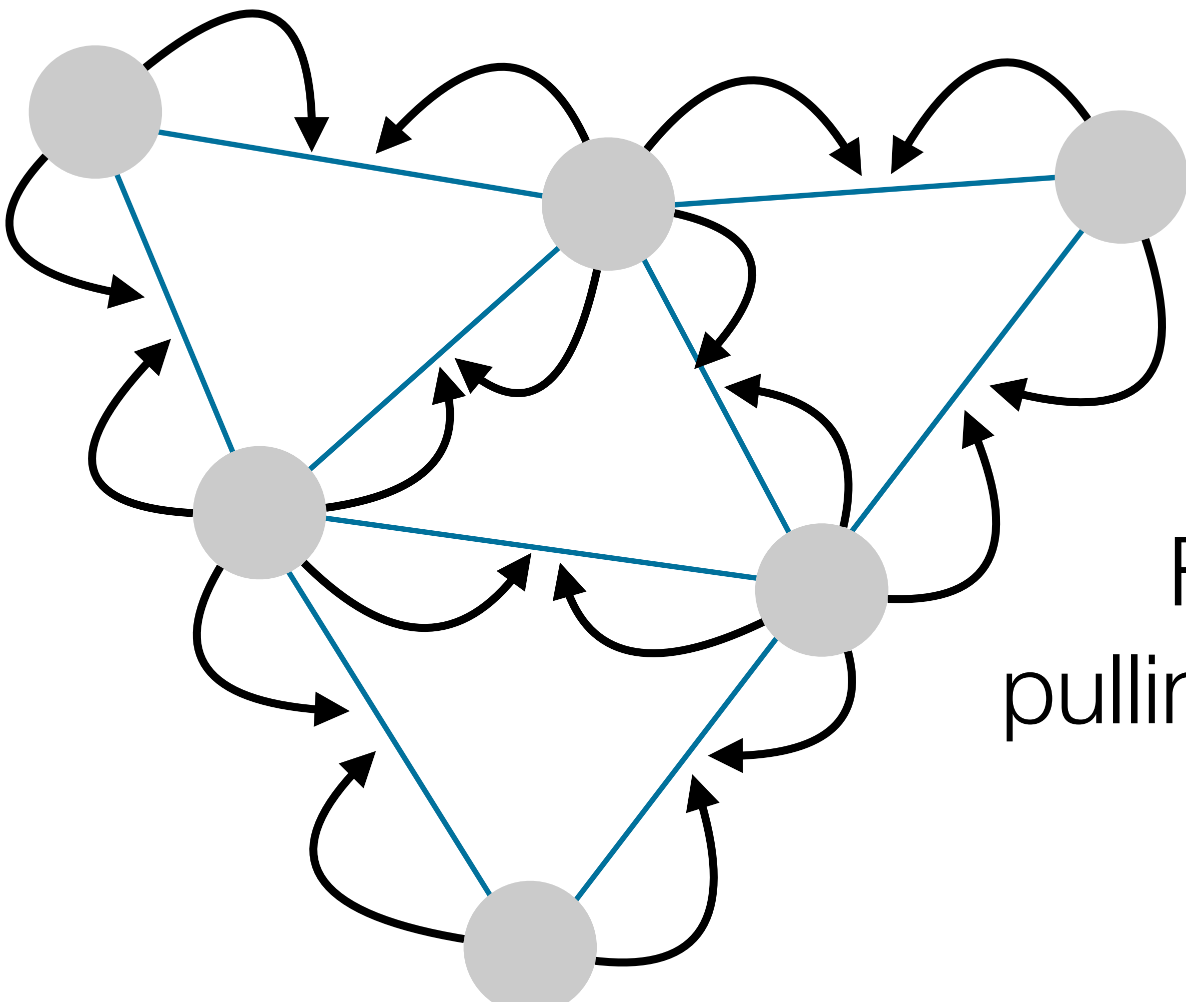
Backup

Message-passing network



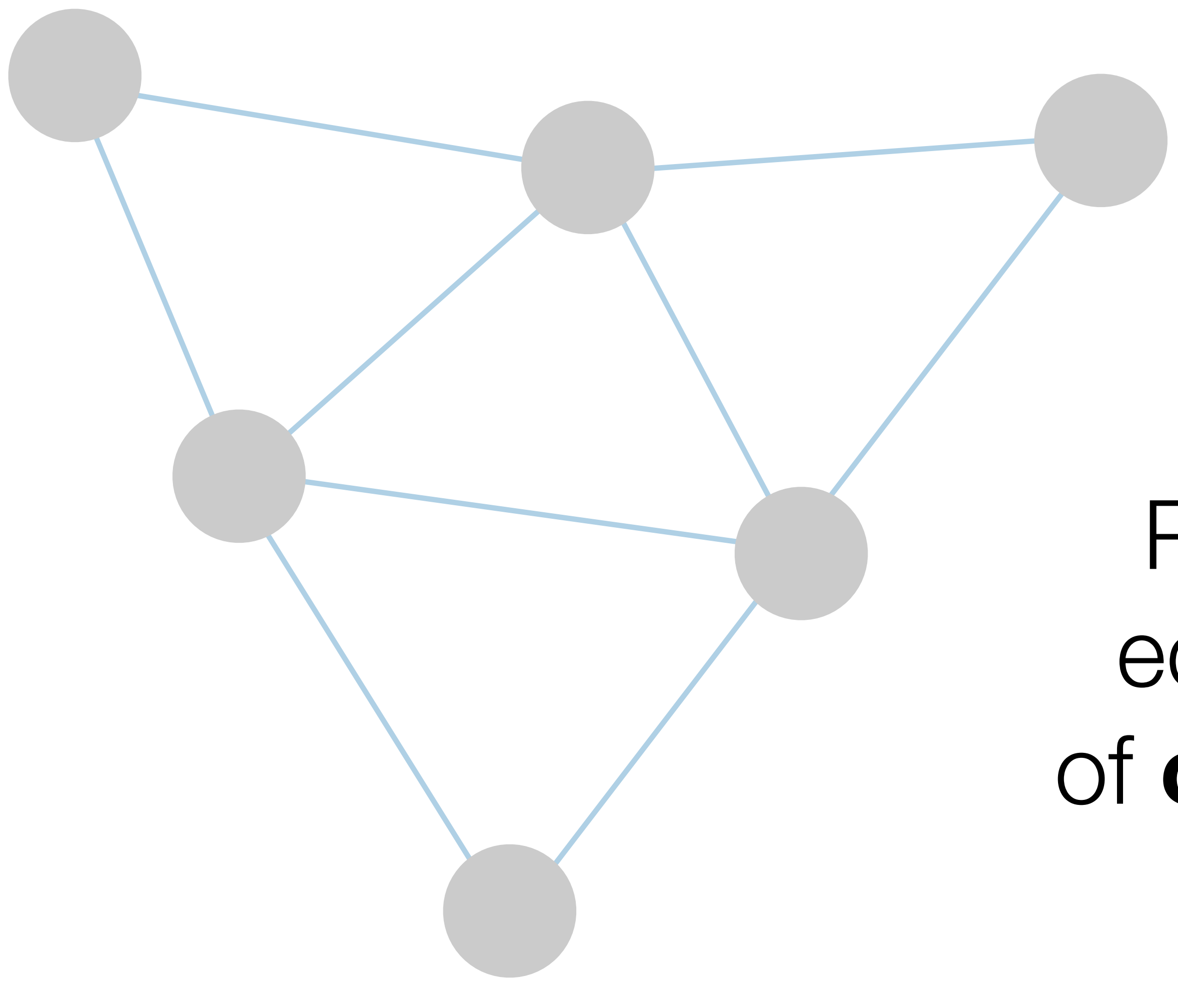
Start with graph **node features**
(hit position, amplitude, RMS, etc)

Message-passing network



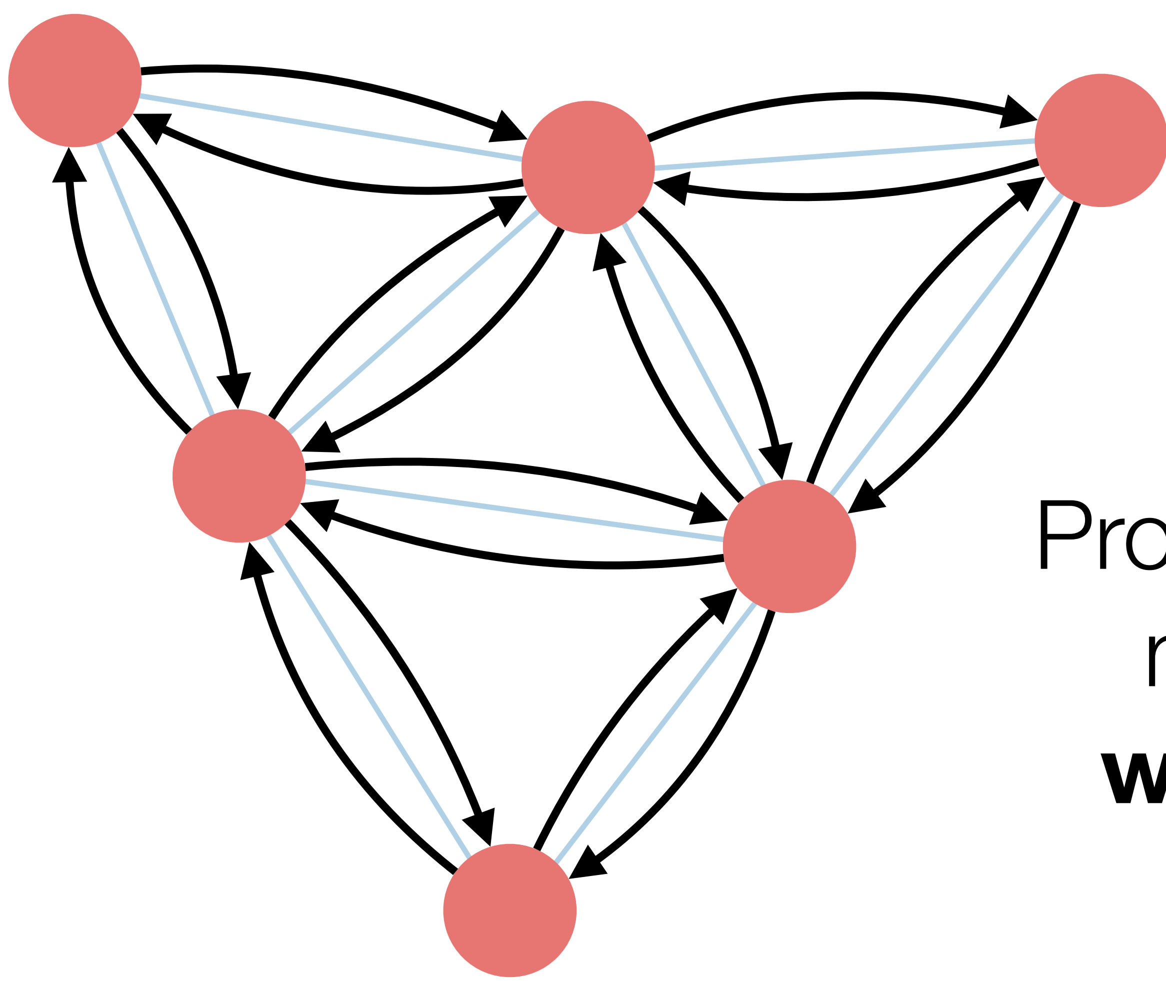
Form **edge features** by pulling in features from incoming and outgoing node

Message-passing network



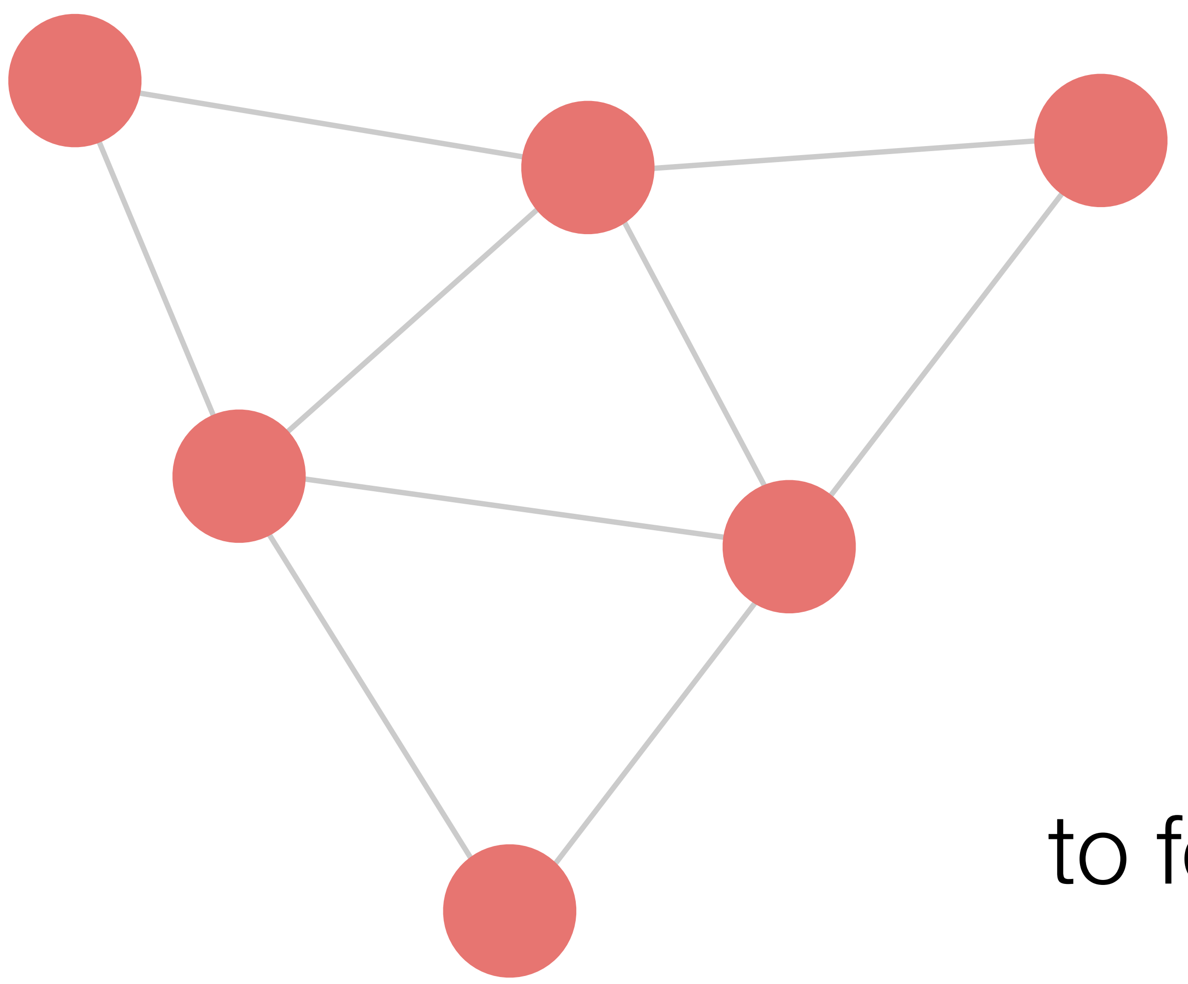
Perform convolutions on edge scores to form a set of **class-wise probabilities**

Message-passing network



Propagate features from each node to adjacent nodes, **weighted by edge score**

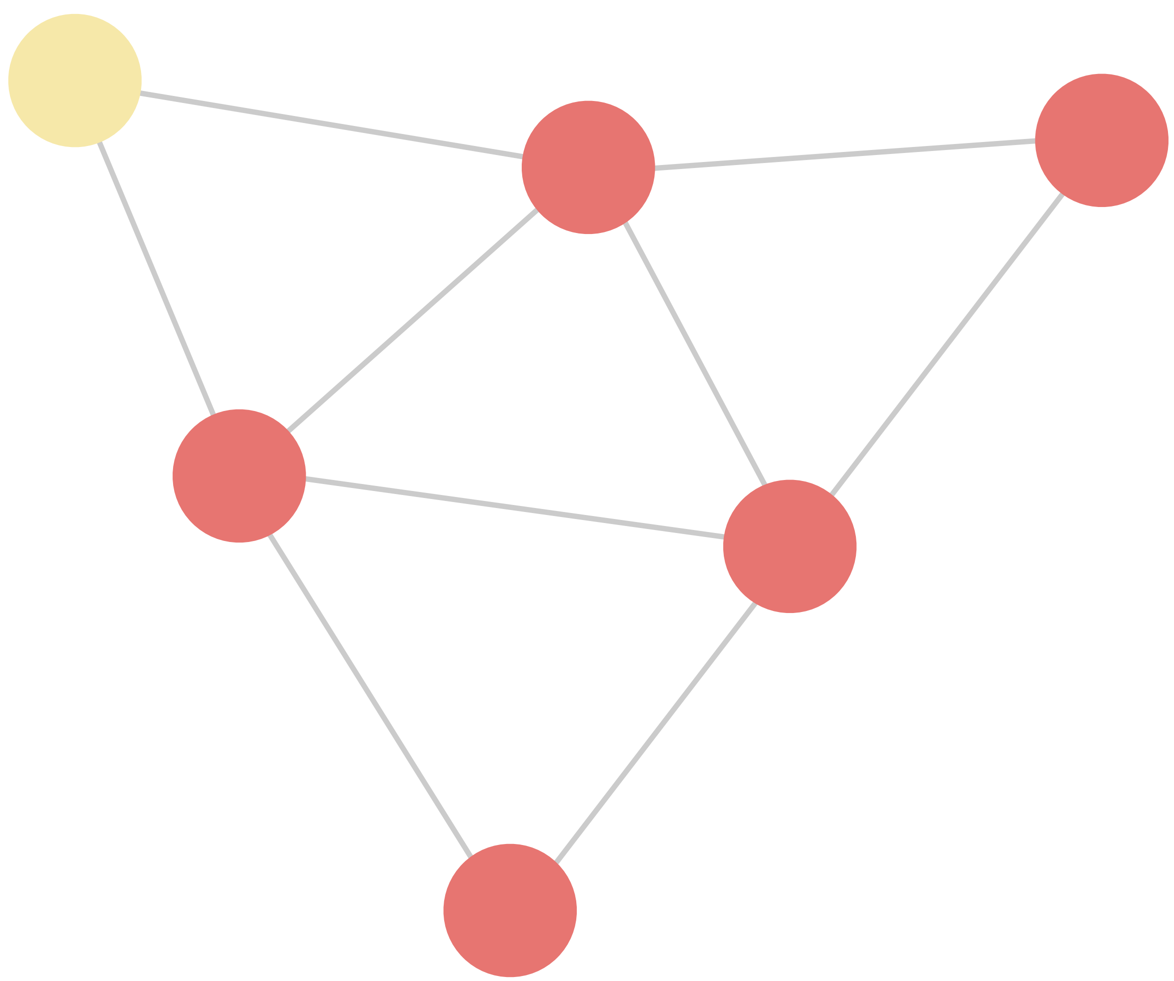
Message-passing network



Perform convolutions
to form new **node features**

Message-passing network

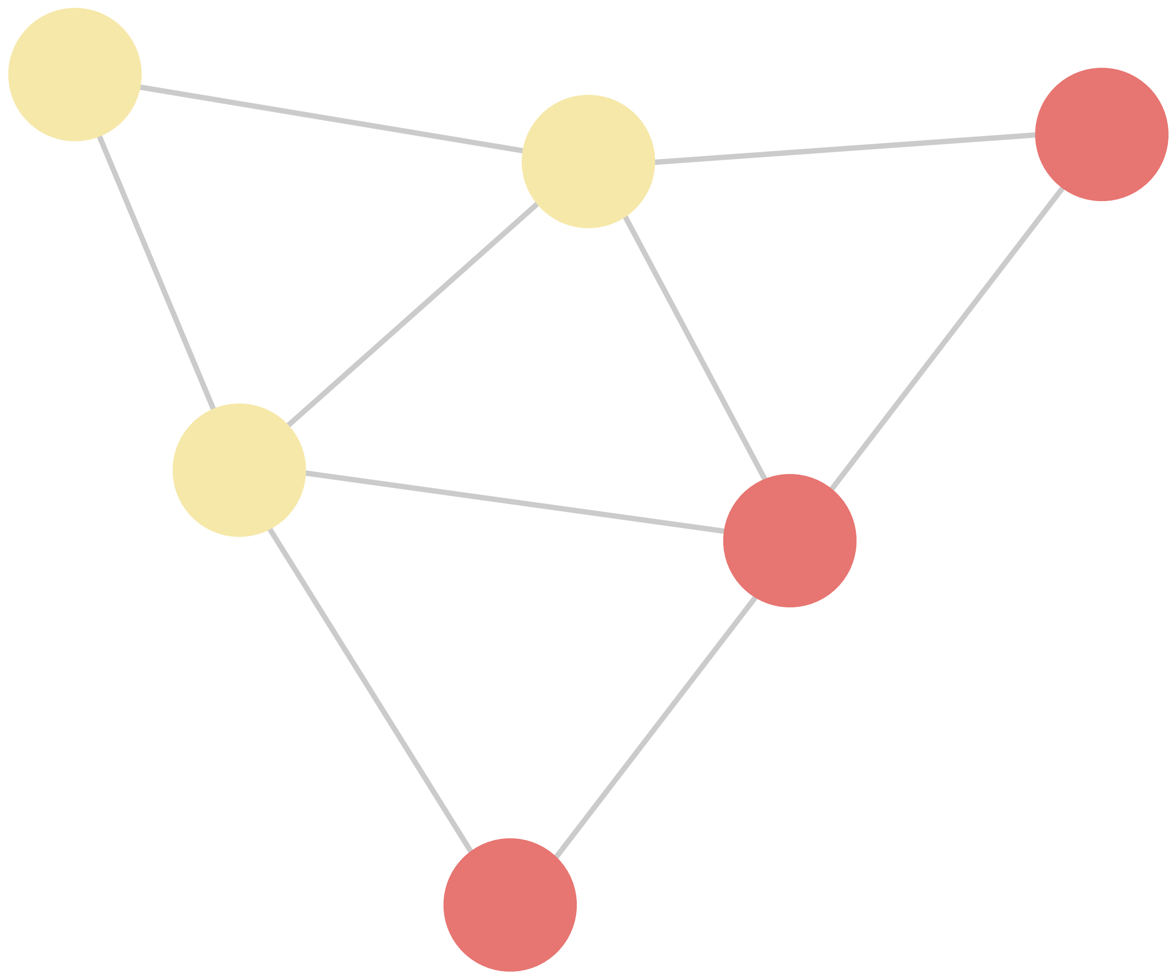
Repeating this process causes information to spread across the graph



Iteration 0

Message-passing network

Repeating this process causes information to spread across the graph



Iteration 1

Message-passing network

Repeating this process causes information to spread across the graph

Iteration 2