THE UNIVERSITY OF CHICAGO

DATA SCIENCE INSTITUTE

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

Kaushal Gumpula (University of Chicago)

Giuseppe Cerati (Fermilab)

Daniel Grzenda (University of Chicago)

V Hewes (University of Cincinnati)

on behalf of Exa.TrkX

Background

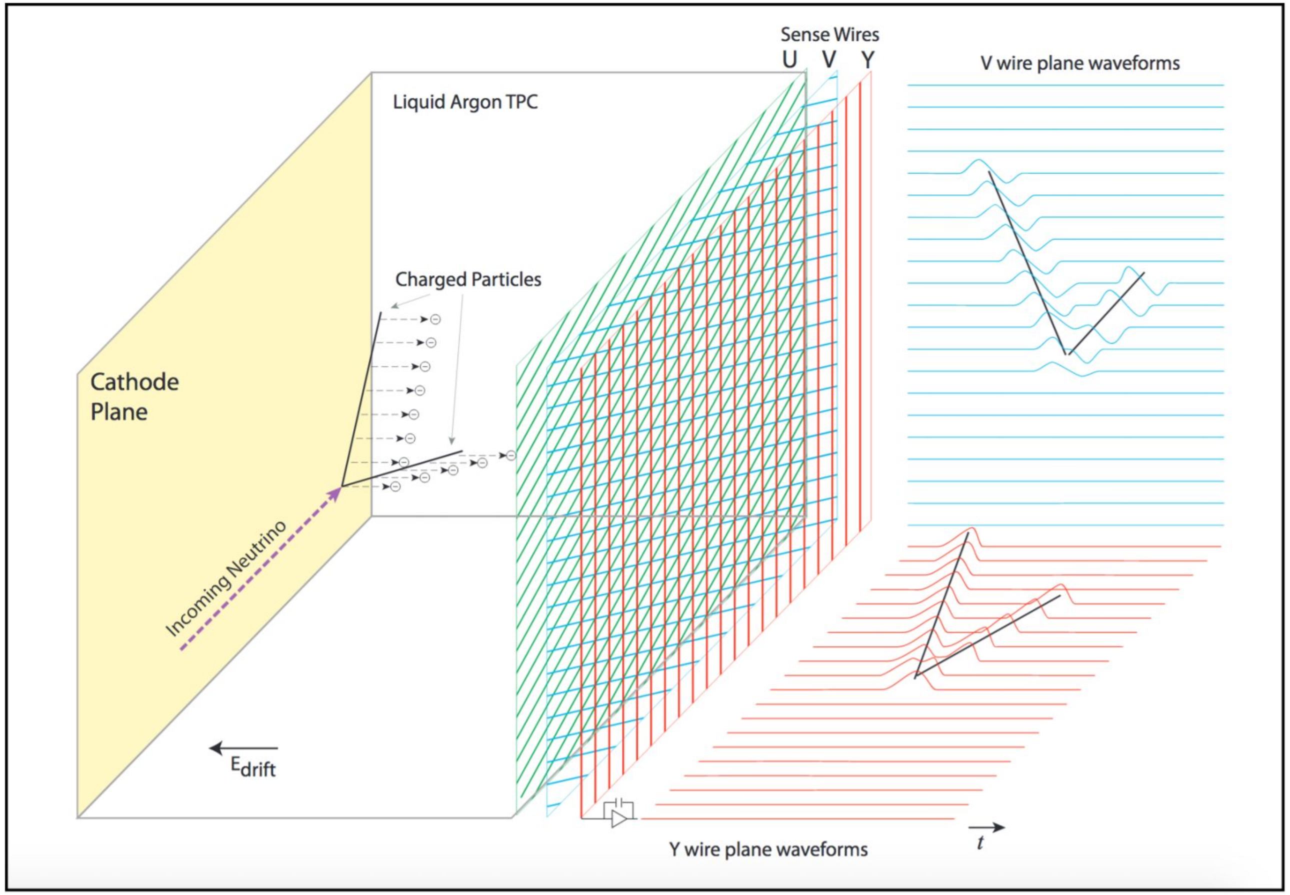
Motivation

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

- Liquid argon time projection chambers (LArTPCs) are the detector technology for modern high-precision neutrino experiments, such as DUNE and MicroBooNE
- Machine learning approaches have been proven successful for reconstruction in LArTPCs, especially for particle identification
    - These approaches are typically based on convolutional neural networks, which treat detector data as images and typically require downsampling of information to fit into memory requirements
- LArTPC detector data is also naturally sparse
    - By translating data into graphs, we can have full usage of the information without downsampling
- We present a graph neural network (GNN) for particle identification and 3D reconstruction for LArTPCs

LArTPC Overview

- Each interaction on each plane is transformed into a graph where each hit is a node and edges are formed between nodes in a certain region or “window” based upon wires and time ticks
- We want to find ways to design a Neural Network, specifically a Graph Neural Network (GNN), and its inputs in order to recognize different particle topologies in these interactions

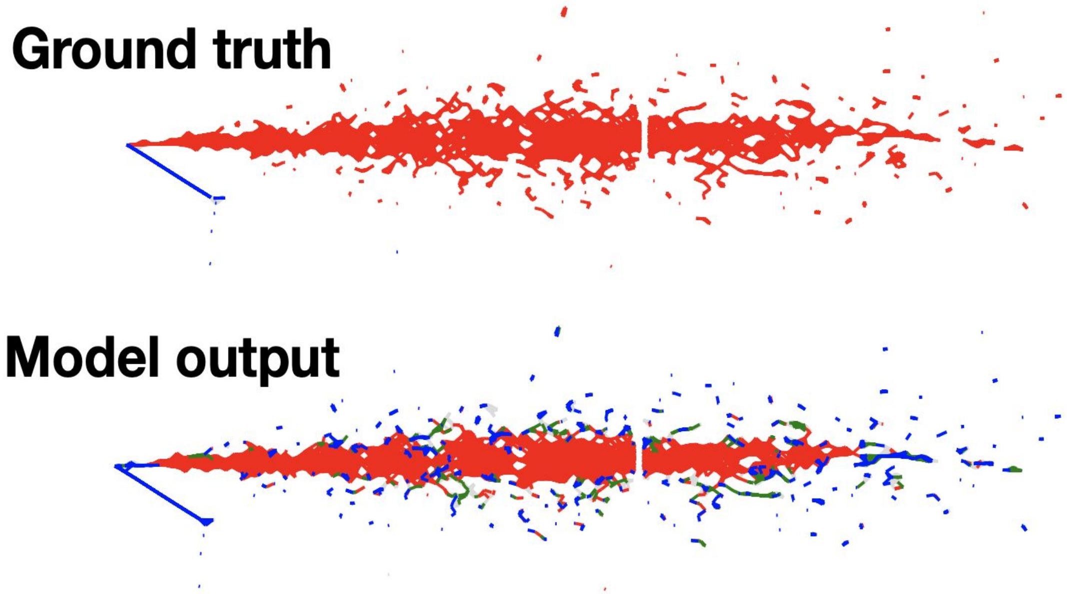


(See arxiv:2103.06233.)

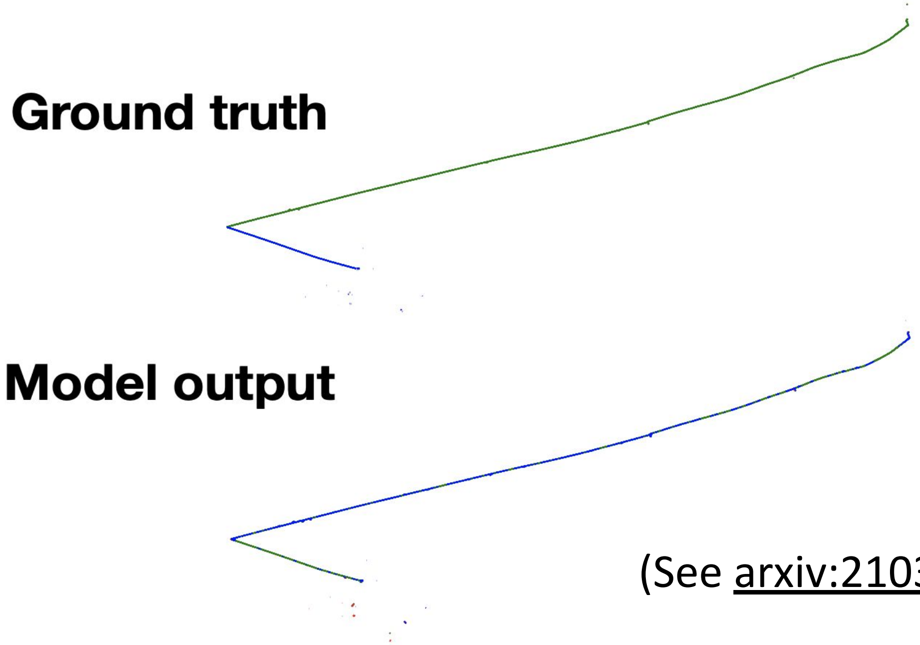
Original GNN Classification

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Electron Neutrino Event



Muon Neutrino Event



(See arxiv:2103.06233.)

hadronic, muon, shower, false

Updated Classification

Previous: 4 categories (edges)

- Hadron, muon, shower, false
- Hadronic class was very general (umbrella class), difficult for the model to learn
- False class was nonphysical, also difficult for model to learn

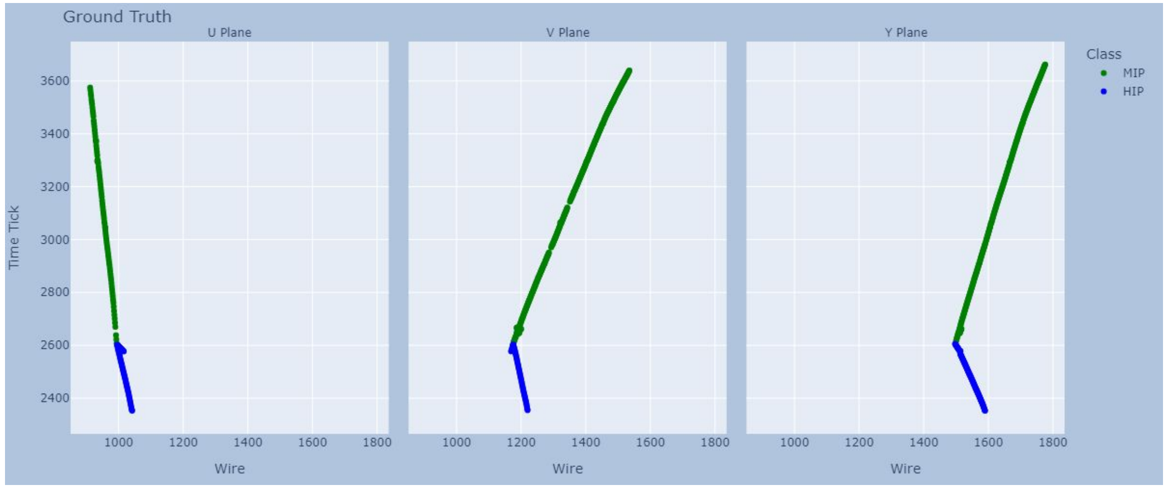
Updated: 8 categories (hits/nodes)

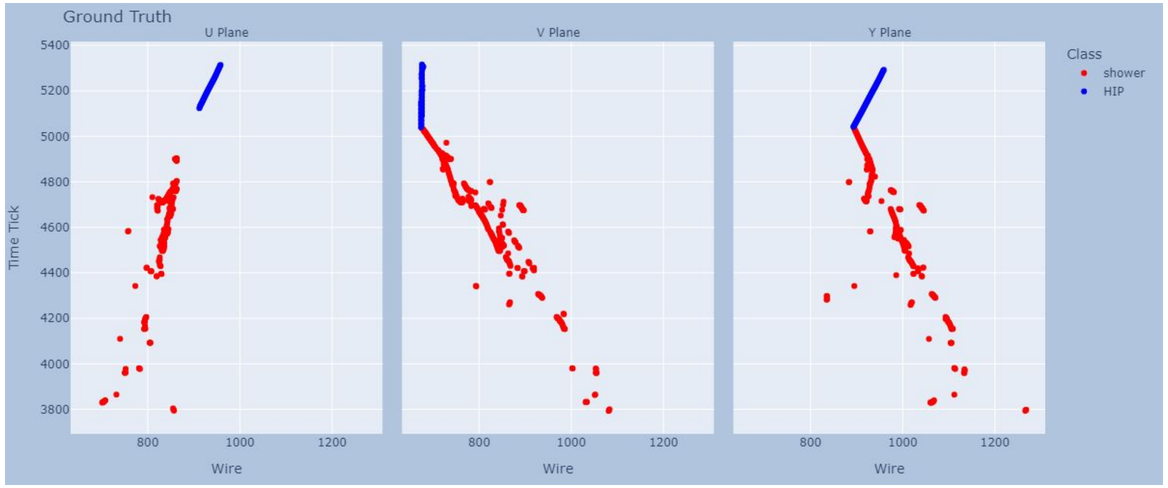
- Pion, muon, kaon, hadron, shower, michel, delta, diffuse

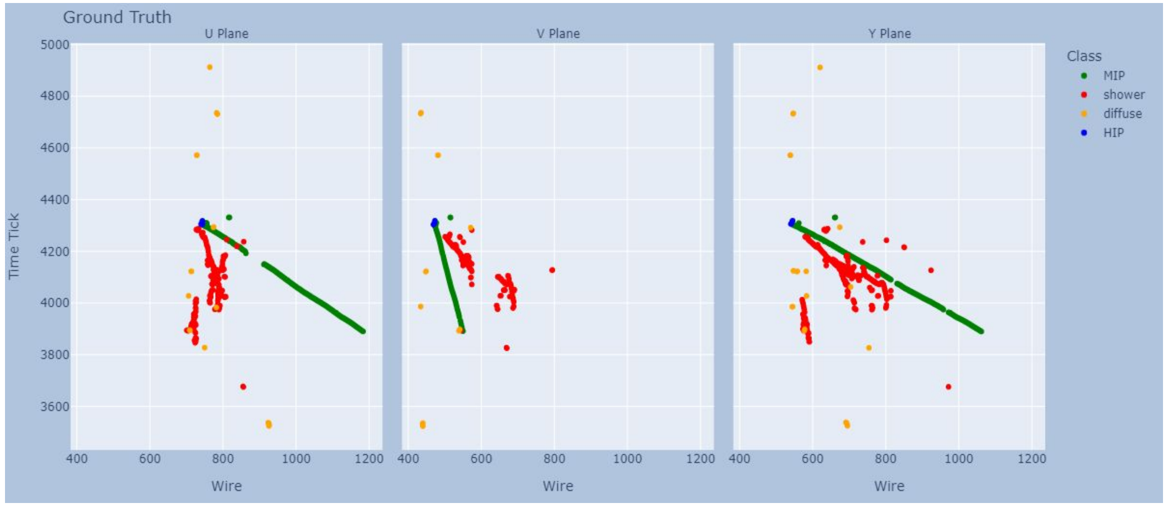
Two New Labeling Schemes were introduced

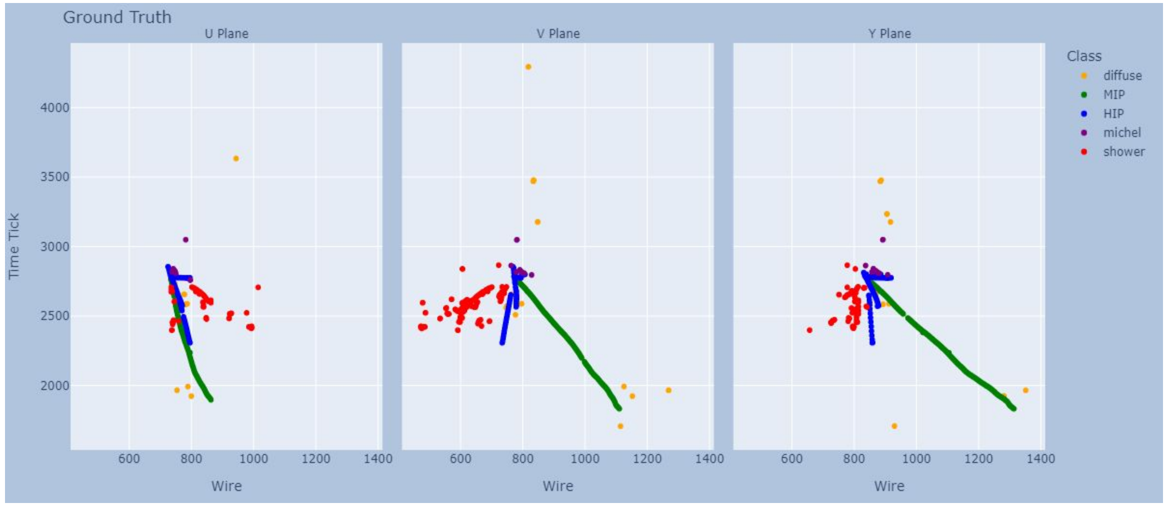
- Simple Scheme
  - MIP(muon, delta, pion), HIP(kaon, hadron), shower, michel, diffuse
- Full Scheme
  - Muon, delta, HIP(kaon, hadron), pion, shower, michel, diffuse

Event Visualizations (Simple)





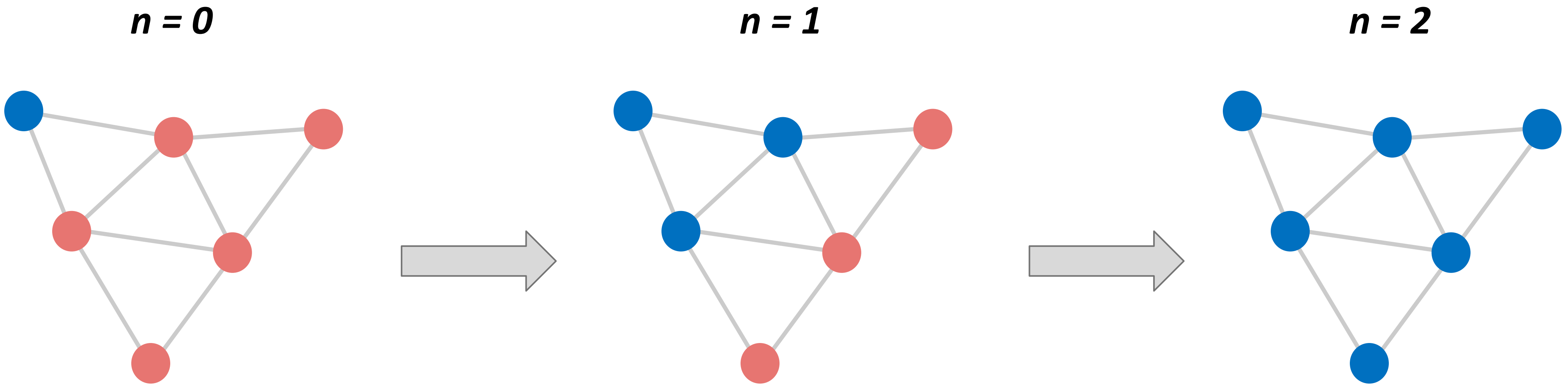




Message Passing and GNN Structure

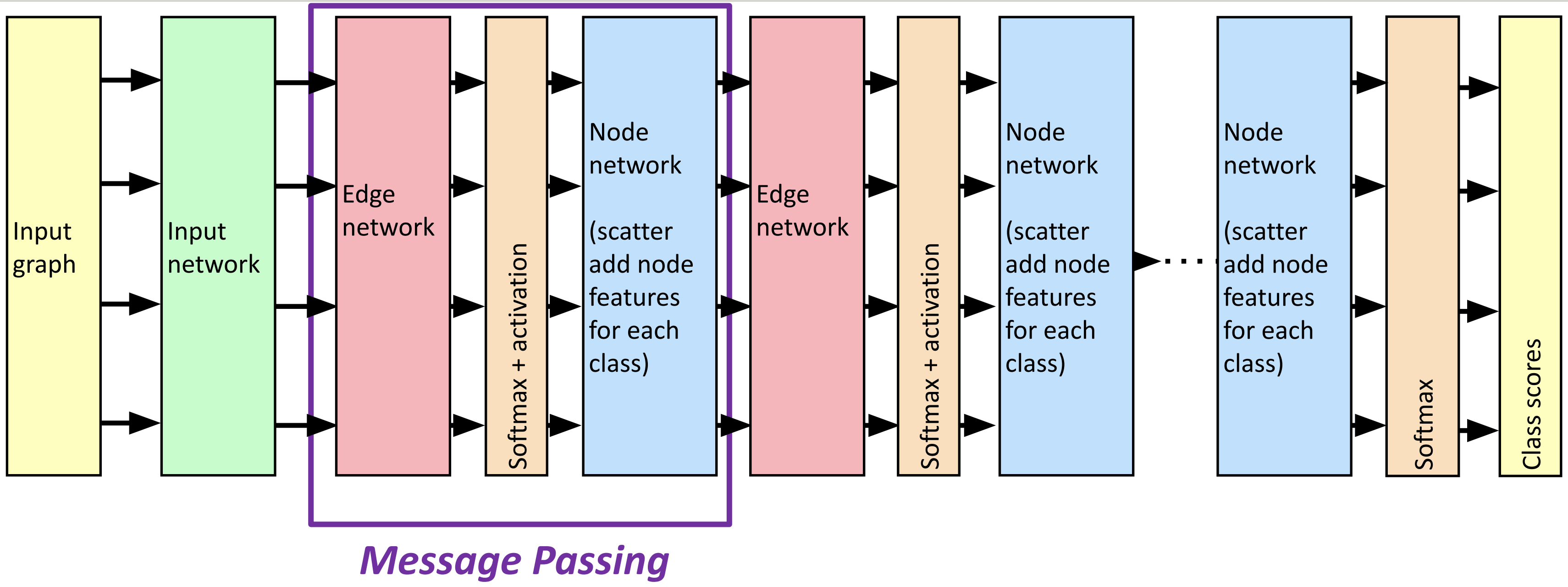
GNN Message Passing

Each Node in the Graph has associated node features such as hit position and amplitude (deposited charge)

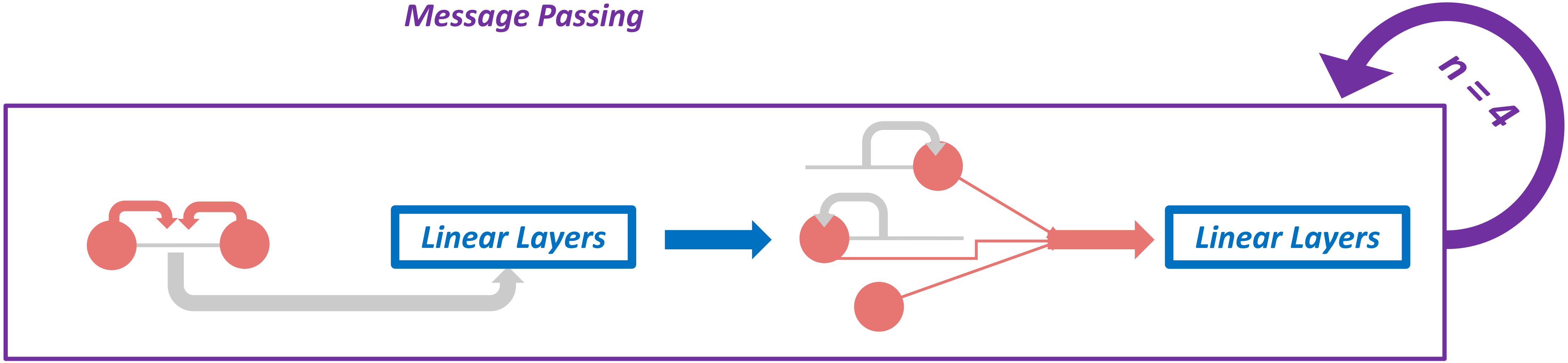


We can explore what happens to information (node features) in our initial node after message passing iteration *n.*

Original GNN Design

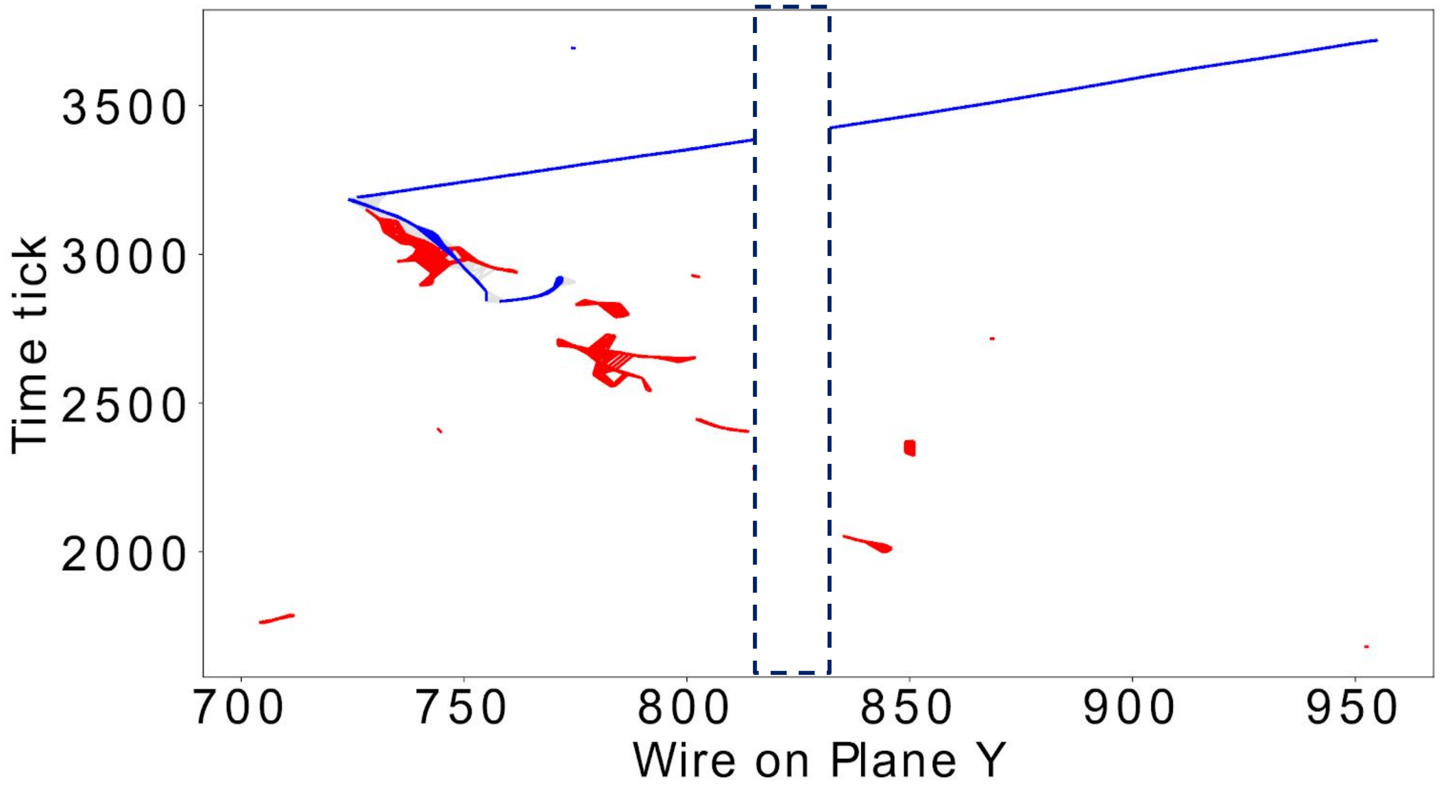


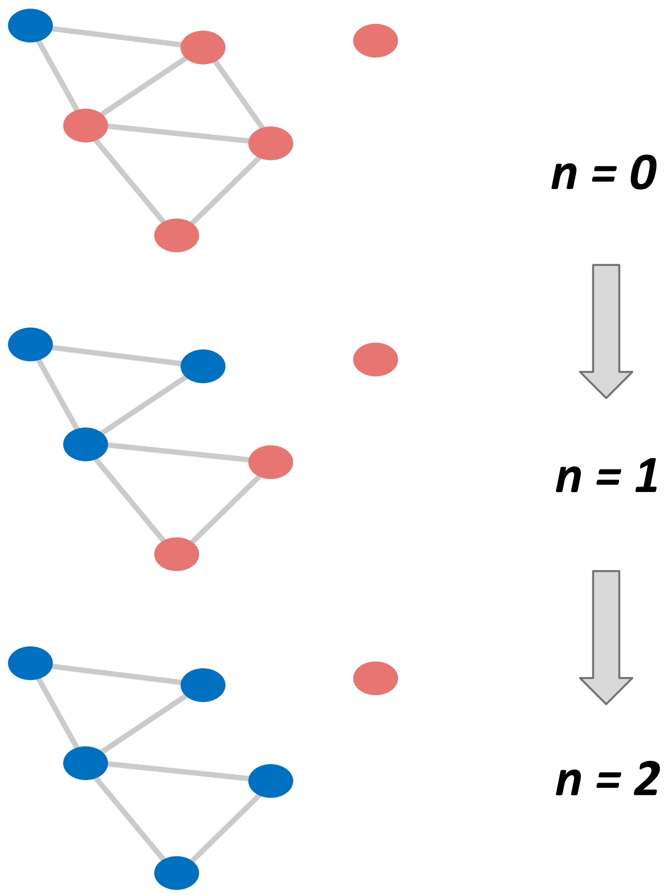
**Thanks to V for this visual!*



Graph Creation

Working on Realistic Simulation





Connectivity in graphs becomes a notable issue for the message passing mechanism

***The interrupted connectivity shown in this example is due to unresponsive wire regions (Not limited to just this case as is evident with the electron shower)**

Edge-Forming Techniques

- In response to this problem, we implemented 4 different edge-forming schemes
  - Window (Original)
    - Connects nodes within a certain (Time Tick x Wire) Window
  - Delaunay
    - Computes Delaunay Triangulation of all nodes in a graph
  - kNN
    - Connects node with its k nearest neighbours
  - Radius
    - Connects node with all neighbours in a certain radius

Dataset Statistics

- Training Set has 651,135 Graphs with an average of 352.84 Nodes per Graph
- 1 Full Interaction translates to 3 Graphs with 1 per plane (U,V,Y)

Class Breakdown across Labeling Schemes

Dataset	Diffuse	Michel	Shower	MIP	HIP	Muon	Delta	Pion
Simple	3.77%	1.15%	24.65%	54.02%	16.41%	- -	- -	- -
Full	3.77%	1.15%	24.75%	- -	11.70%	52.91%	0.95%	4.77%

Average Edges across Edge-Forming Schemes

Edge-Forming Technique	Window	Delaunay	kNN	Radius
Average Edges Per Graph	966.82	1046.48	1400.52	1291.39

2D Results

2D Model Configuration Summary

THE UNIVERSITY OF CHICAGO DATA SCIENCE INSTITUTE

- 8 Total Model Configurations
    - 4 Different Edge-Forming Schemes
    - 2 Different Labeling Schemes (Simple and Full)

2D Performance Summary

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

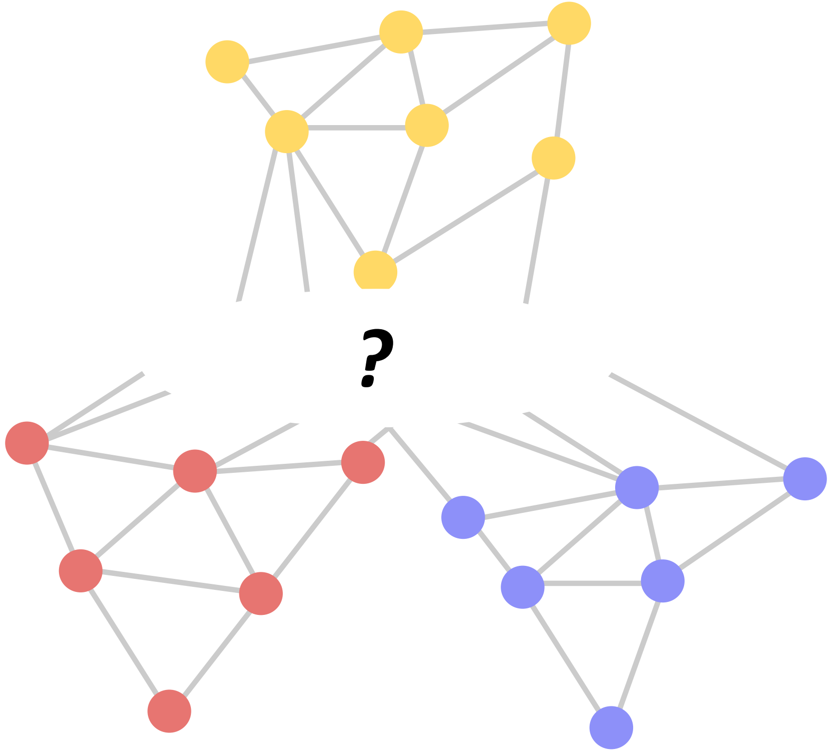
Edge Type	Data Type	Labeling Scheme	Accuracy	Consistency
Delaunay	**2D**	**Simple**	**86.24%**	**74.45%**
Window	2D	Simple	76.9%	60.27%
kNN	2D	Simple	81.14%	64.62%
Radius	2D	Simple	78.58%	61.26%
Delaunay	**2D**	**Full**	**81.97%**	**67.38%**
Window	2D	Full	74.64%	56.89%
kNN	2D	Full	78.52%	60.54%
Radius	2D	Full	76.02%	57.60%

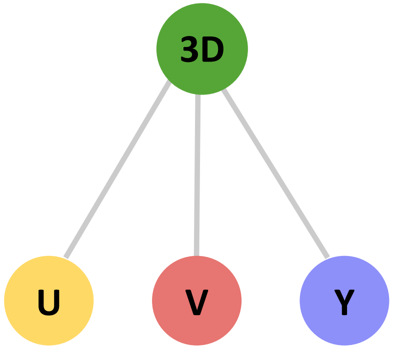
- Delaunay Edge-Forming Scheme performed the best for both Simple and Full Labeling Schemes
- Introduction of Consistency Metric

3D SpacePoints

How do we leverage all three views?

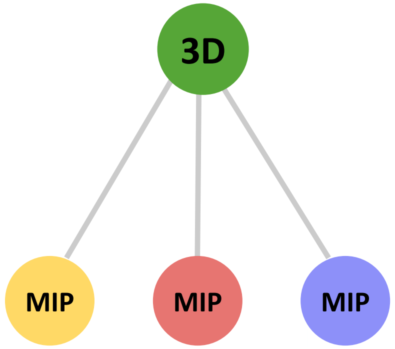
While having access to three different “views” of an event, the model does not leverage this information



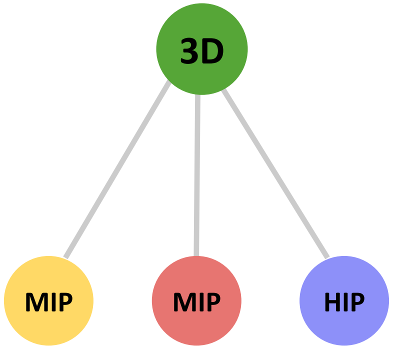


3D Representation (SpacePoint)

(See arxiv:2002.03005.)



Consistent 3D Representation

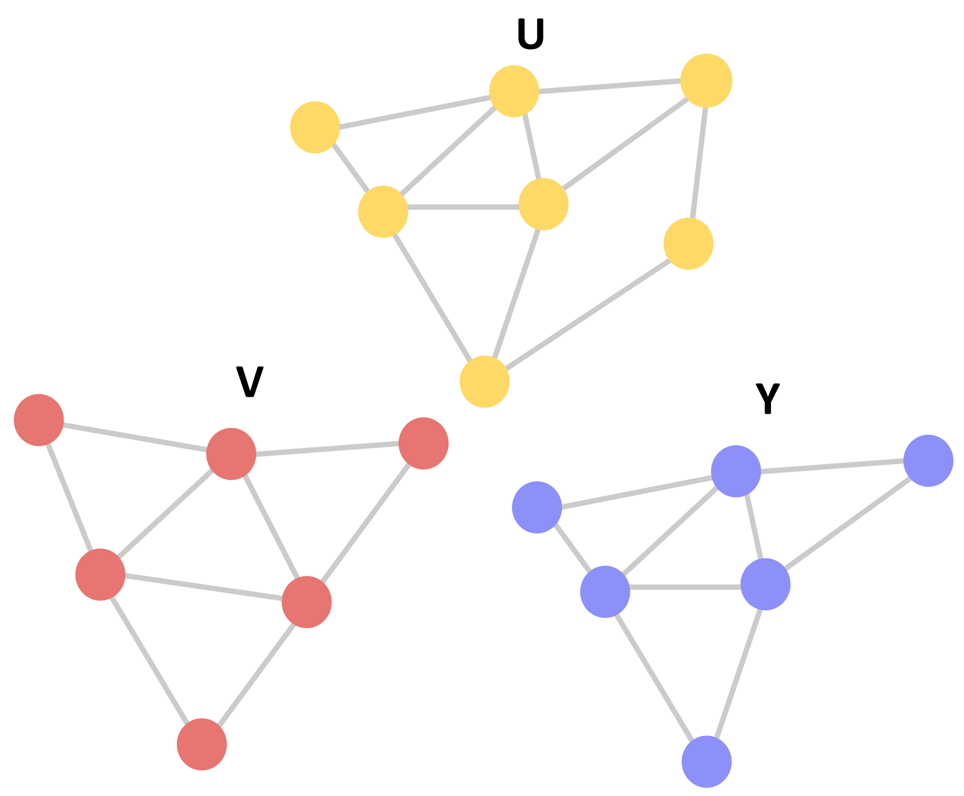


Inconsistent 3D Representation

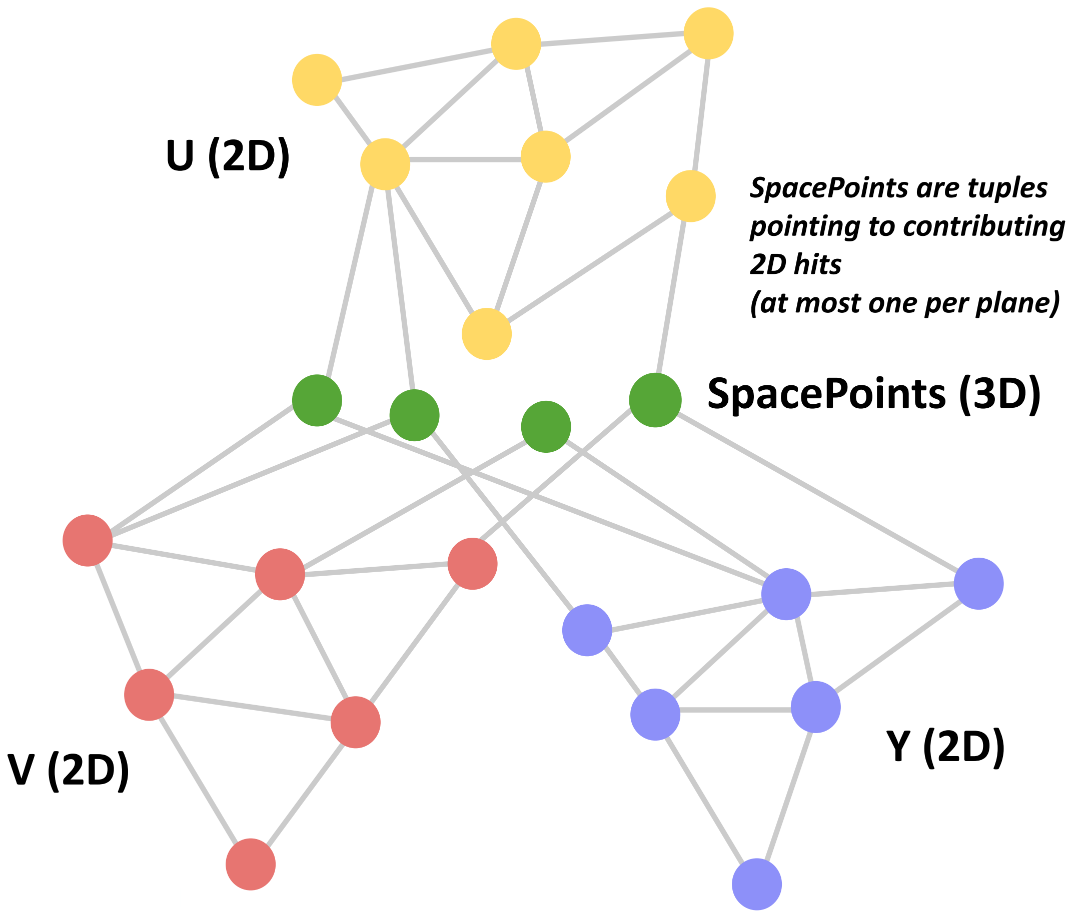
Implementation of 3D SpacePoints

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

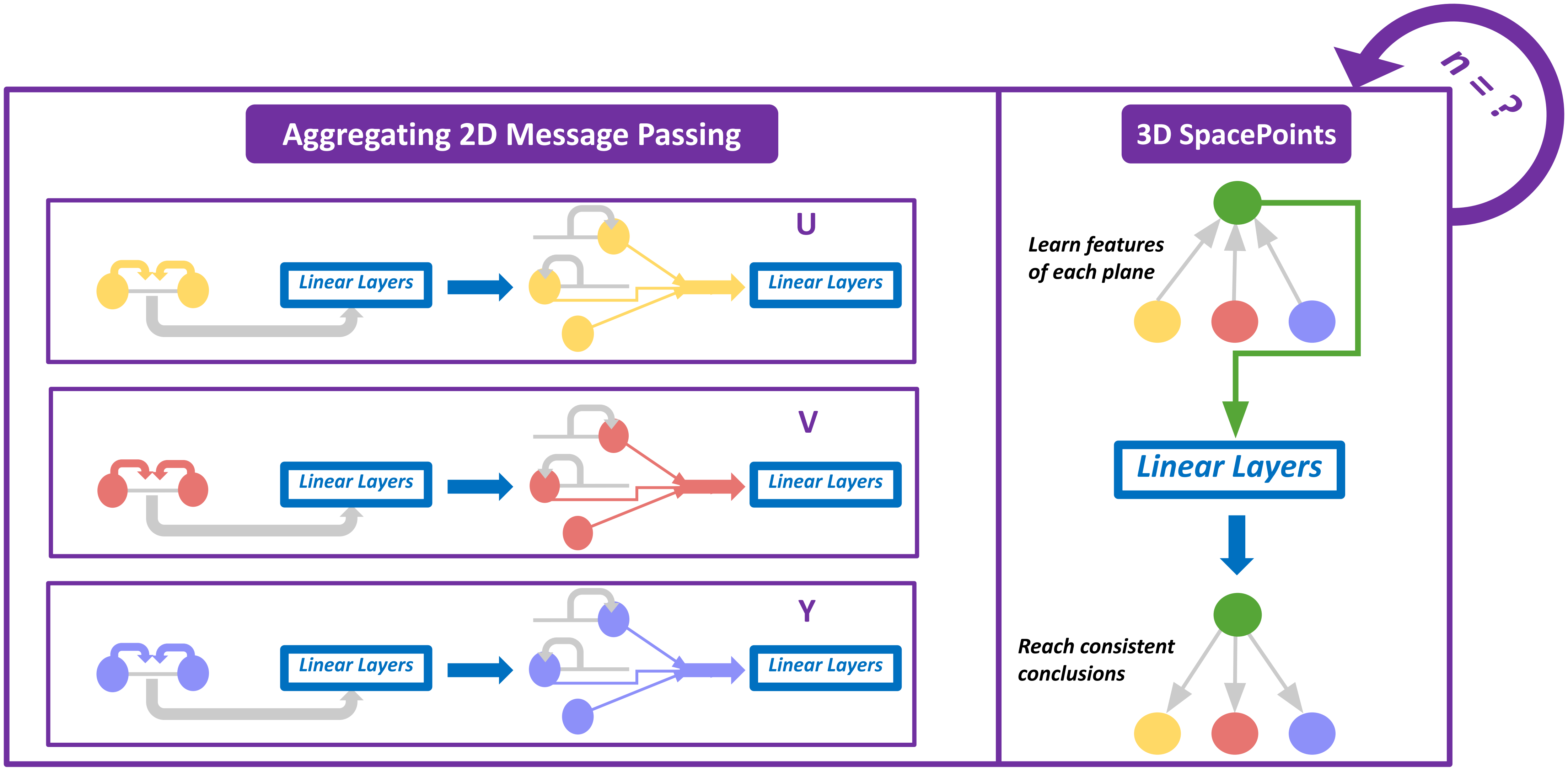
Package all three planes into one Object



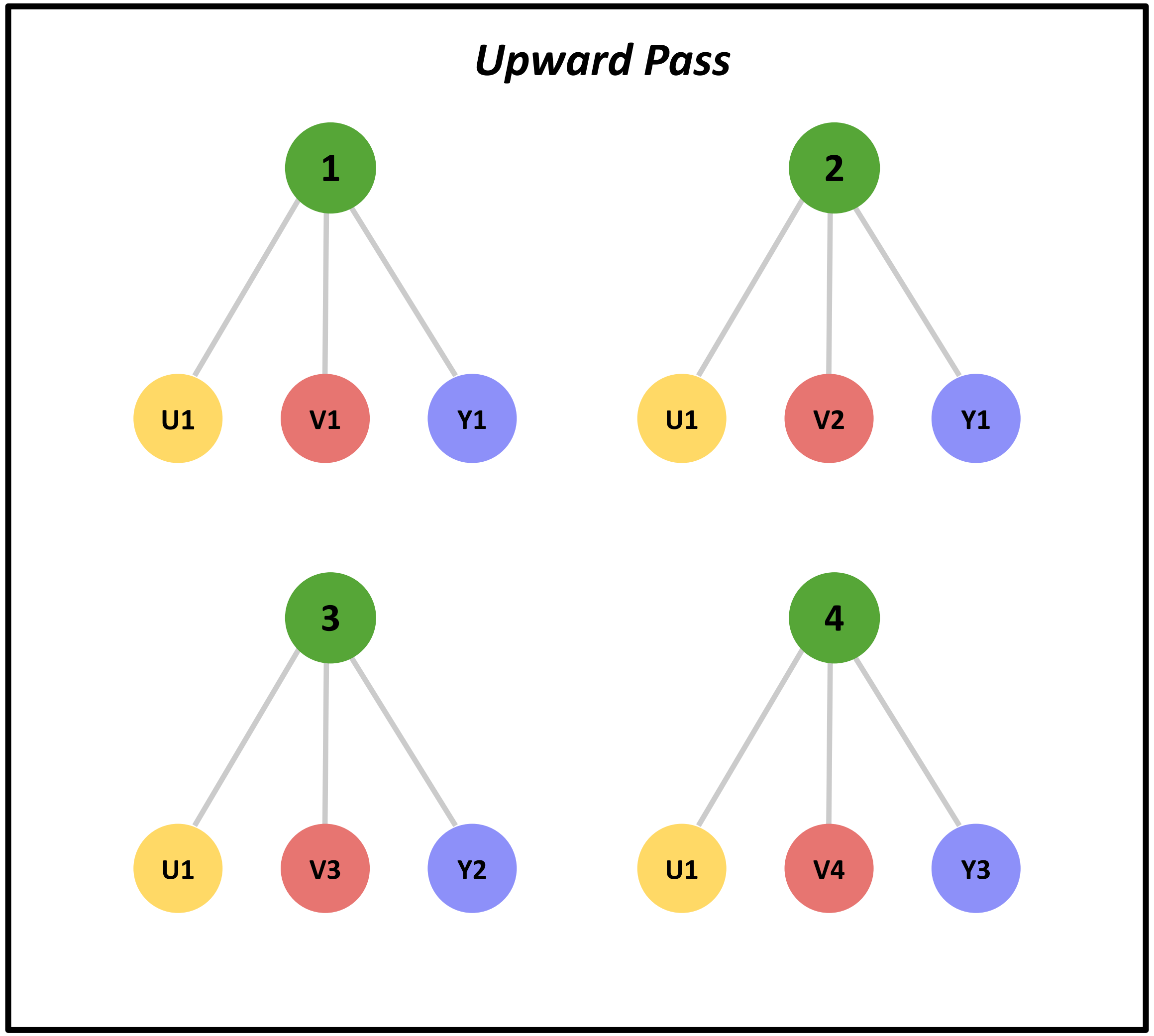
Connect three views through 3D SpacePoints

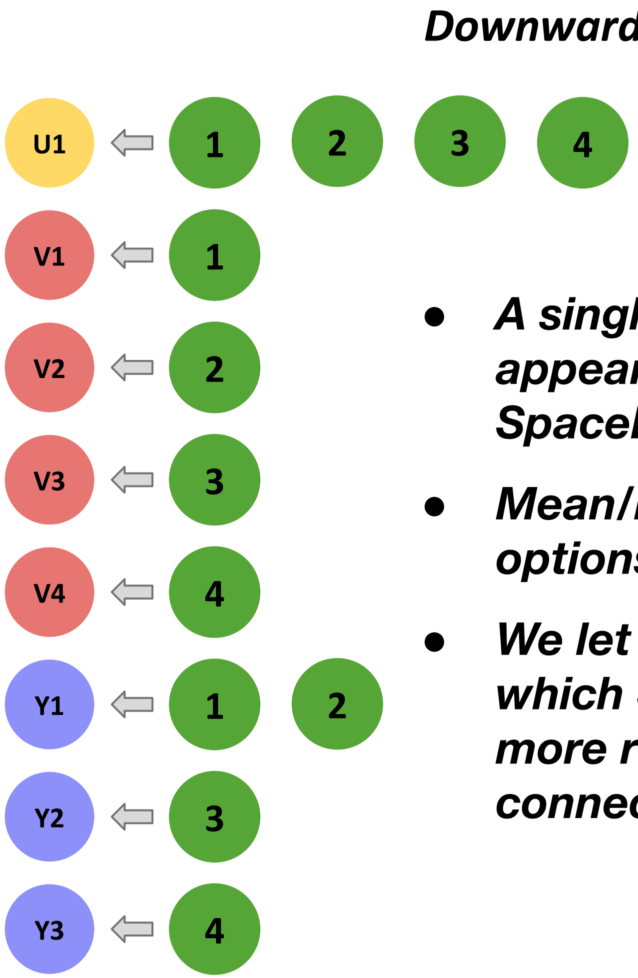


Message Passing with 3D SpacePoints



SpacePoint Pooling and Attention





- ***A single 2D hit can appear in multiple SpacePoints***
- ***Mean/Max Pooling were options we tested***
- ***We let the model learn which SpacePoints are more reliable using their connections***

2D/3D Result Summary

Full Model Configuration Summary

- 16 Total Model Configurations
  - 4 Different Edge-Forming Schemes
  - 2 Different Labeling Schemes (Simple and Full)
  - 2D vs 3D

Full Performance Summary

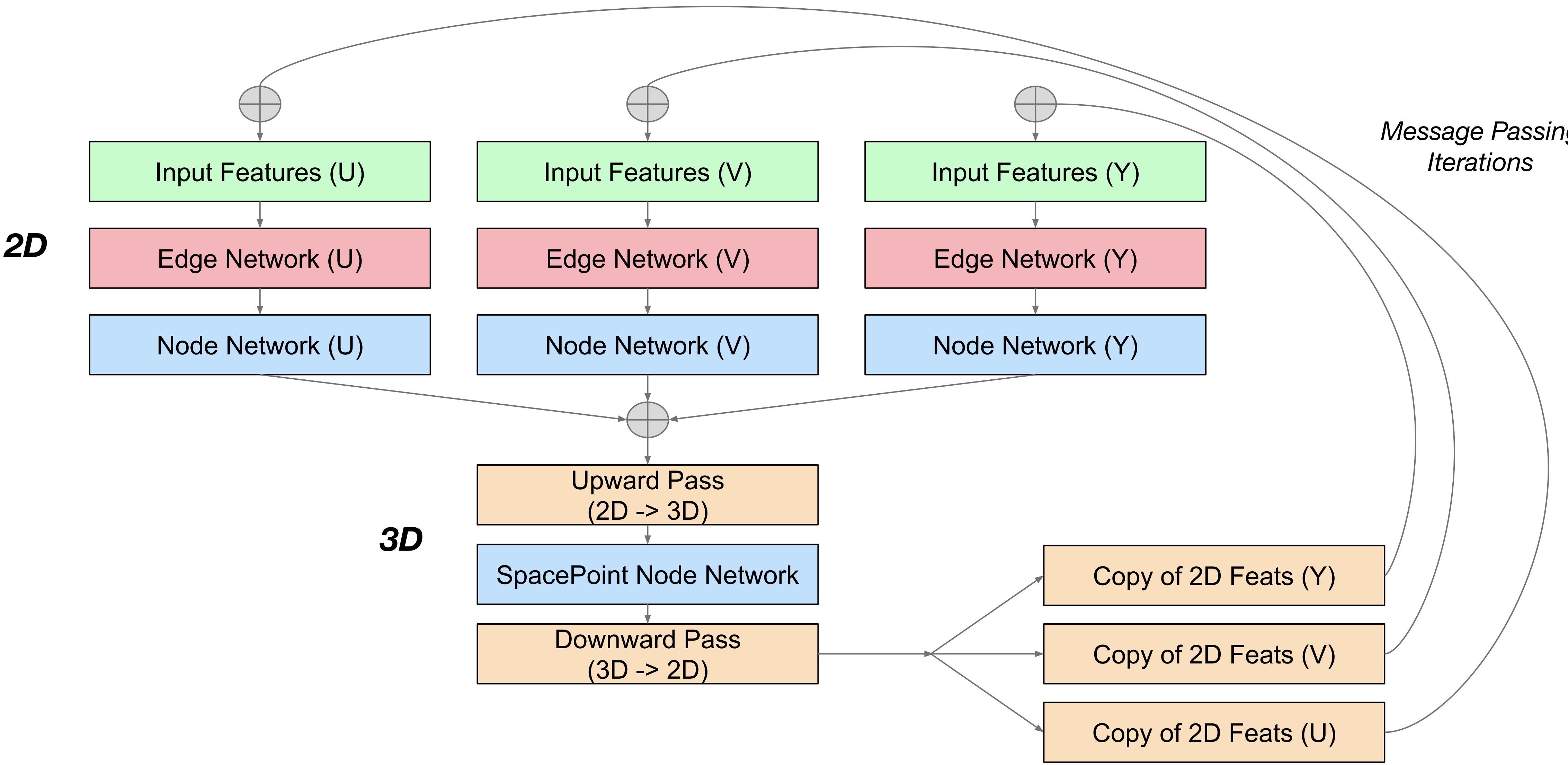
Edge Type	Data Type	Labeling Scheme	Accuracy	Consistency
Delaunay	**2D**	**Simple**	**86.24%**	**74.45%**
Window	2D	Simple	76.9%	60.27%
kNN	2D	Simple	81.14%	64.62%
Radius	2D	Simple	78.58%	61.26%
Delaunay	3D	Simple	82.41%	96.34%
Window	**3D**	**Simple**	**78.18%**	**97.35%**
kNN	3D	Simple	80.89%	96.61%
Radius	3D	Simple	79.57%	96.48%
Delaunay	**2D**	**Full**	**81.97%**	**67.38%**
Window	2D	Full	74.64%	56.89%
kNN	2D	Full	78.52%	60.54%
Radius	2D	Full	76.02%	57.60%
Delaunay	**3D**	**Full**	**79.02%**	**94.38%**
Window	3D	Full	75.13%	93.60%
kNN	3D	Full	76.77%	93.67%
Radius	3D	Full	75.92%	94.14%

- 3D Results are very consistent, but lack in accuracy when compared to their 2D counterparts
- However, the 2D models have comparatively higher accuracy, but terrible consistency
- Consistency in truth across 3D SpacePoints is around 95-97% which suggests that there is learnable information
- How do we bridge this accuracy-consistency gap?

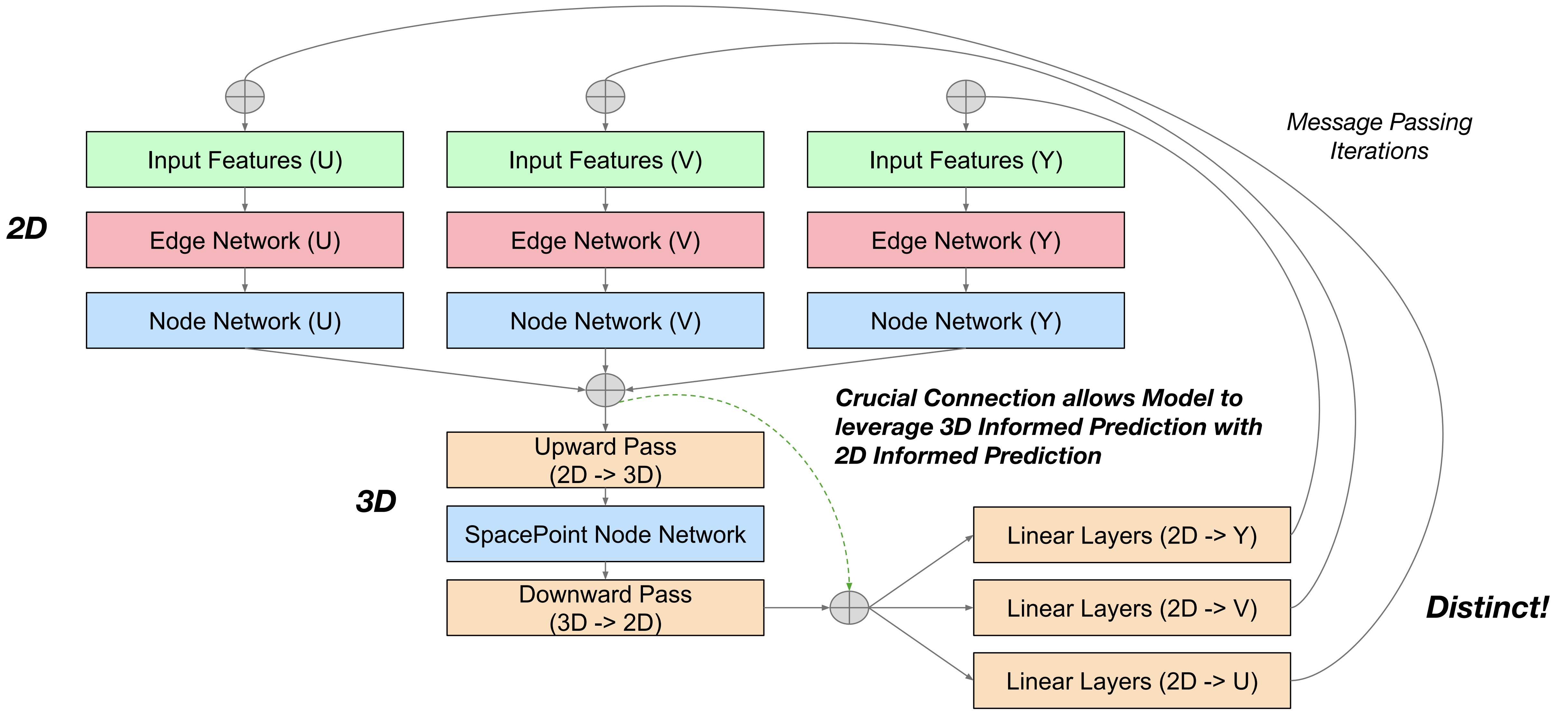
The SpacePoint Problem

Current Network (Simplified)

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE



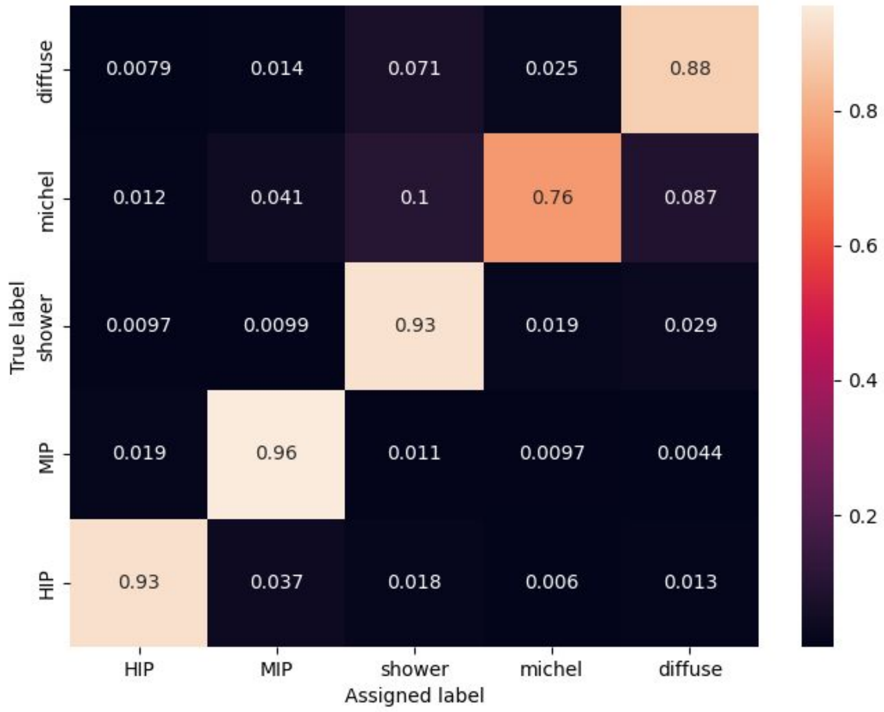
Enforcing vs Encouraging Consistency



Preliminary Performance Breakdown

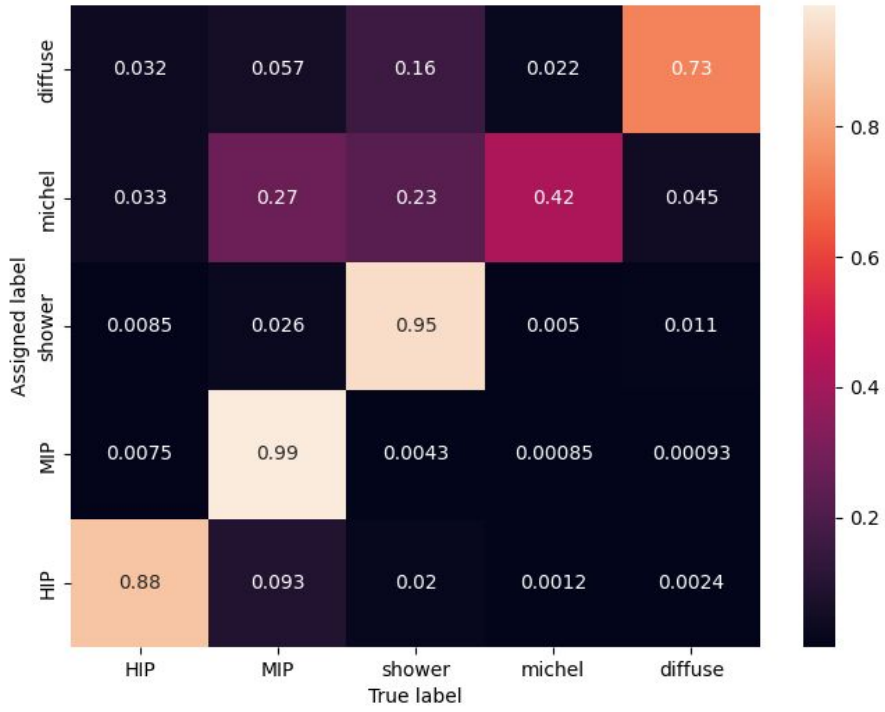
THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Efficiency



*for element **ij** in the matrix, it tells us what fraction of **class i in truth** is classified as **class j by the model***

Purity



*for element **ij** in the matrix, it tells us what fraction of **class i classified by the model** is actually **class j in truth***

Next Steps

Future Improvements

- Continue developing 3D Model
  - After the previously discussed change, the 3D Model was able to reach **94% Accuracy (first-pass results)**
  - As further proof-of-concept, this iteration of the model also had **96% Consistency**
  - Even without "enforcing" consistency across all three planes, the model was able to learn it and bridge the accuracy-consistency gap by itself
- Hyperparameter Tuning
  - Optimal Message-Passing Iterations
  - Convergence given # of Epochs
  - Batch Size and Size of Hidden Input Dimension
- Panoptic Segmentation

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Thank You!