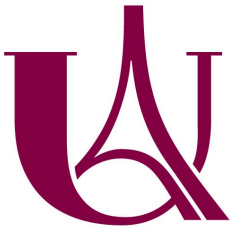
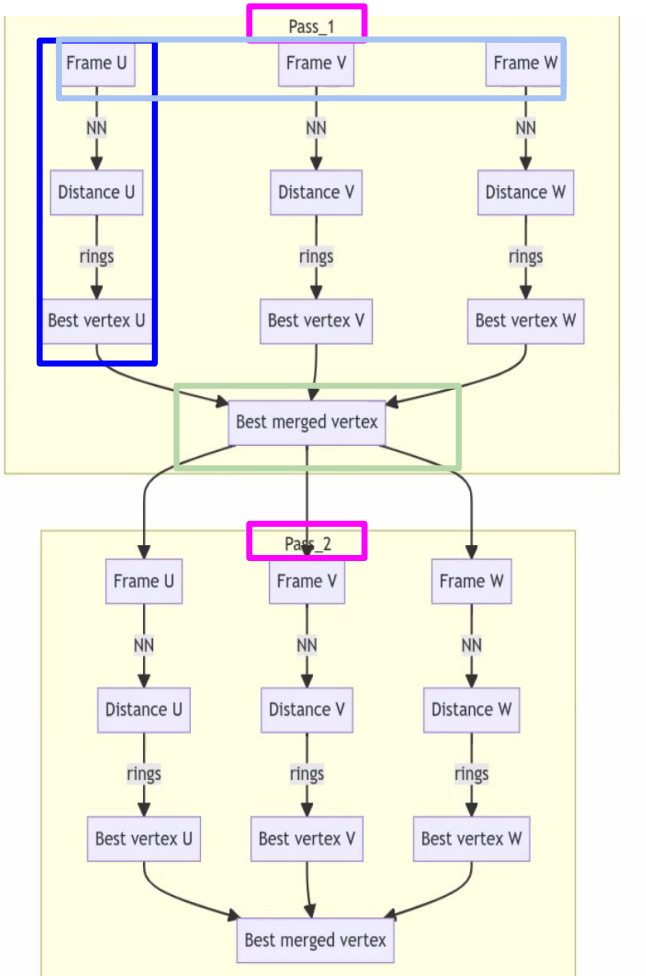
Update on Pandora Vertex finding

I Cheong Hong



Many thanks to P. Granger & H. Souza for help!

Pandora DL vertexing algorithm at a glance



- Two Passes
  - Pass_2 ‘zooms-in’ into the vertices found by Pass_1
  - but are otherwise identical

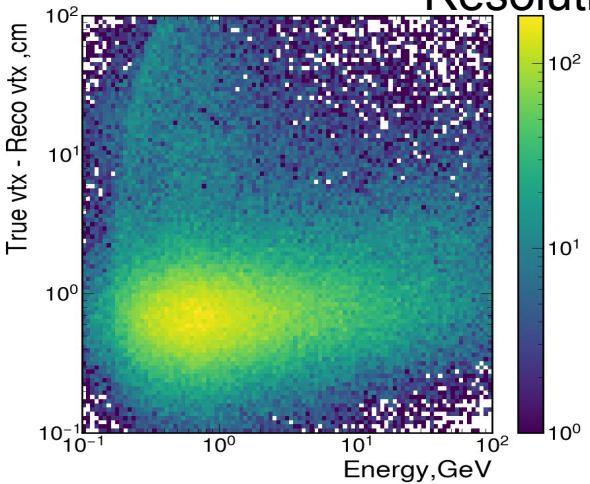
- Each pass made of several steps
  - same steps for each frame and Pass_1 & Pass_2
  - final 3D vertex found by merging info from all 3 frames

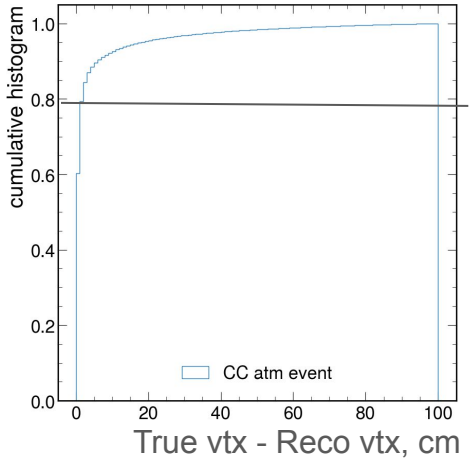
A recap on what was done…

- **Investigation of the current vtx reco algo**
  - General performance: resolution
  - Failure modes: topologies of events where vtx failure happens, type of failures
  - Step by step to identify points of failures

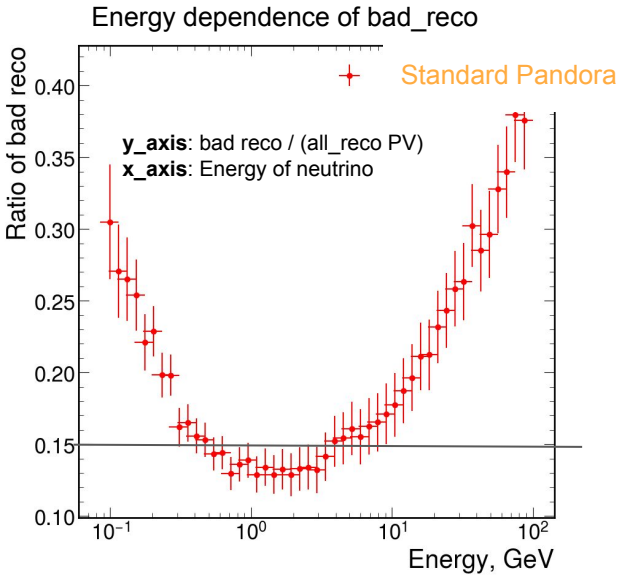
A recap on what was done: general performance

Resolution





17% vtx reco > 3cm for CC



bad_reco: reco vtx > 3cm distance to TruthVtx

Vtx reco worse below ~500MeV and above ~5-10 GeV

*Dataset: ~160k atm CC sample

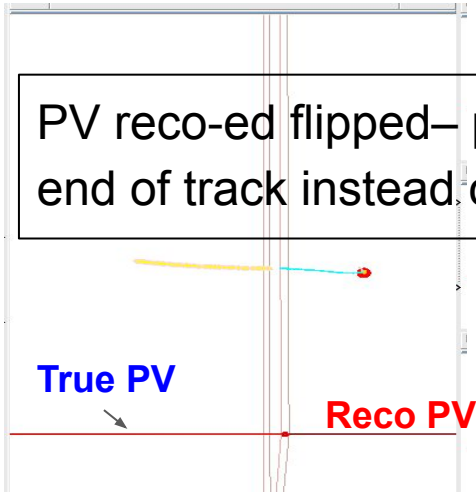
Failure modes:

48%

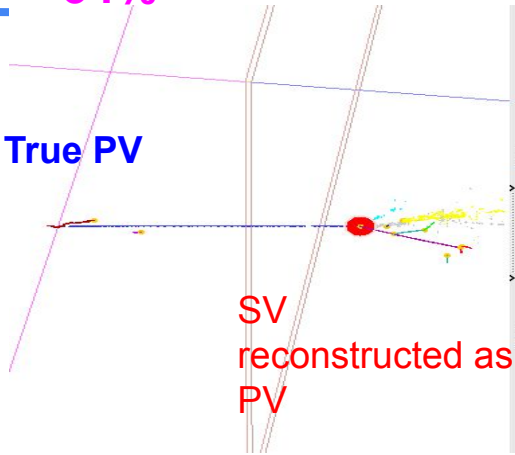
>3cm away from truth, but in the right direction

13%

PV reco-ed flipped– placed at end of track instead of start



34%



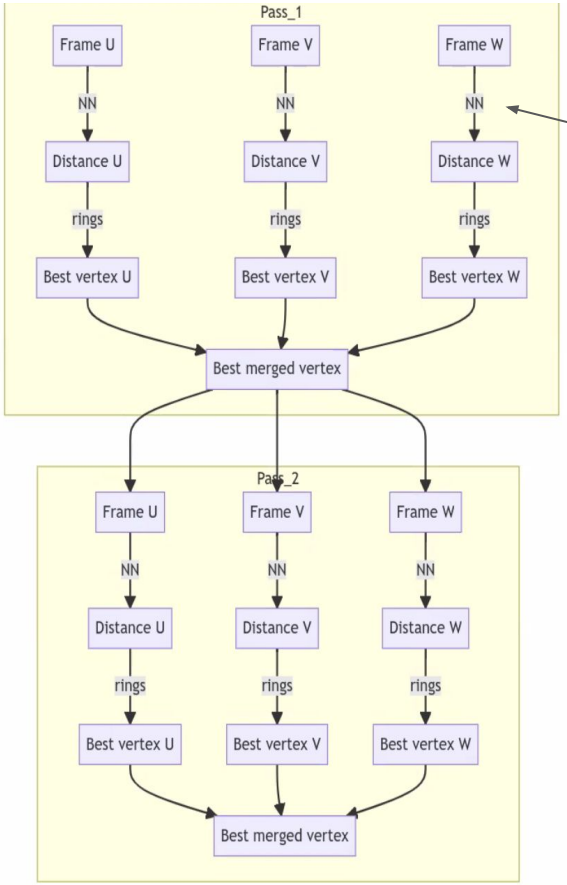
SV wrongly identified as PV

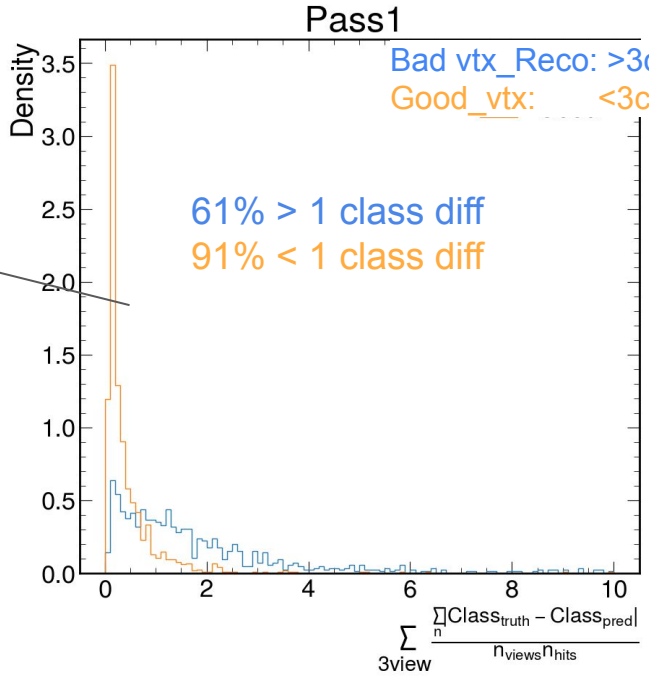
Bad reco_vtx events are also events that

- contain more 1-track events
- contain more short tracks (<40cm)

*Dataset: atm CC sample

Main point of failure





Bad reco sample: large failure (61%) from the first NN

Since the last sim reco

- **Consider more than 1 vtx candidate (look to more than the 1st hot spot)**
  - **in addition: also checked a modified version of Andy's secondary-vertex finder**
- Think if training more than one network, one for a different energy range, makes sense to push
- Design a chain of networks, one that would identify failure modes, and one that would fix them

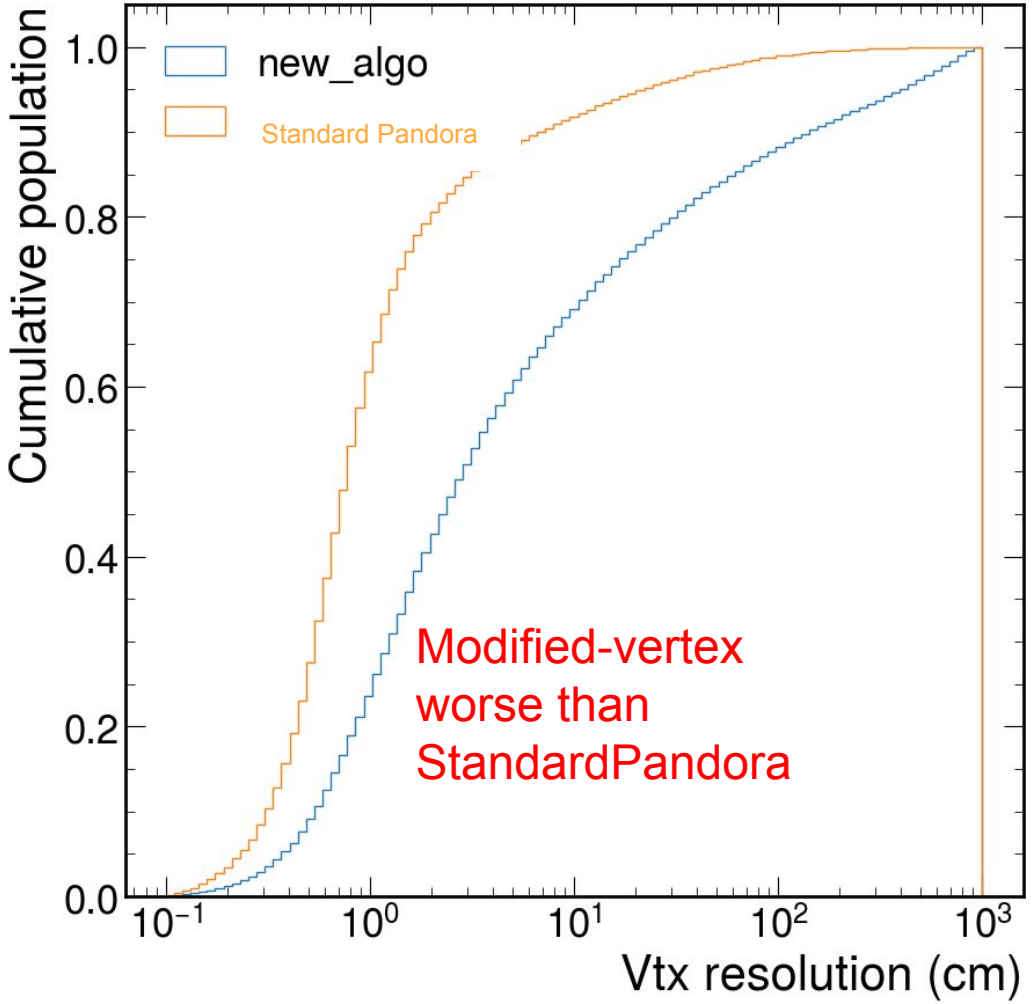
- **Modified 2ndary vtx finder**

1.Start from Pandora 2nd vtx-finder algorithrim, to get the vertex candidates for each view

- obtain Nu, Nv, Nw numbers of 2D vertices for each frame

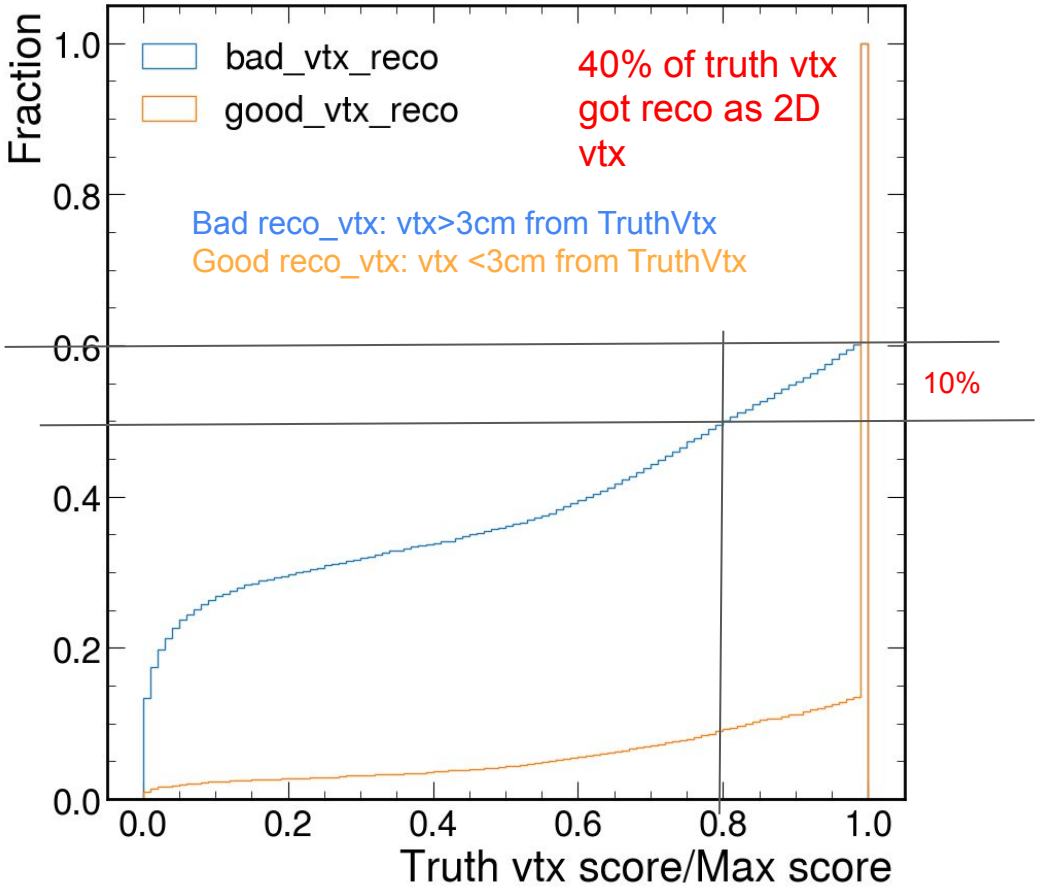
2.To get one 3D primary vertex:

- make all possible combinations with the 2D vertices (Nu-s, Nv-s, Nw-s)
- keep the 3D vtx with highest average score + low chi^2



- **Looking at more than 1 hotspot**

- **Truth vtx score:**
  - the score asssgiend by NN to the TruthVtx position
- **Max score:**
  - the highest HS-score assigned to the view, where the PV is picked by the algo
-
- y-axis: cumulative population
  - nominator: events w/ score < x-axis
  - denominator: all events

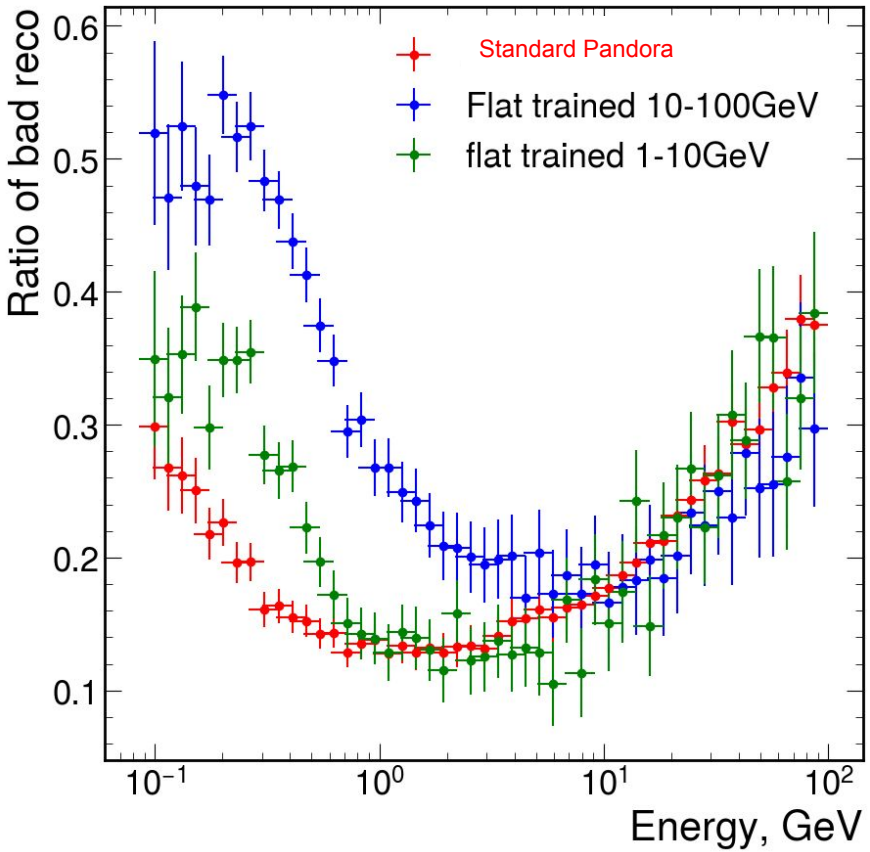


Bad vtx sample : only 10% of TruthVtx contain score between 0.8->1

Since the last sim reco

- ~~Consider more than 1 vtx candidate~~
- **Think if training more than one network, one for a different energy range, makes sense to push**
- Design a chain of networks, one that would identify failure modes, and one that would fix them

- **Training more than one network for a different energy range**



- **Standard Pandora**: trained on a 60k atm sample; falling-E spectra
- **Flat trained:** trained on 60k atm sample, flat-E spectra within specified range

The hyperparamater / training details same as in StandardPandora training

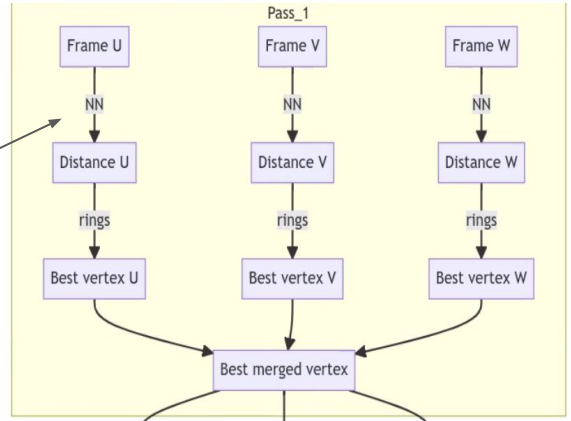
Not leading significant improvement

Since the last sim reco

- ~~Consider more than 1 vtx candidate~~
- ~~**Think if training more than one network, one for a different energy range, makes sense to push**~~
- **Design a chain of networks, one that would identify failure modes, and one that would fix them**

Ideas right now:

1. A GNN (or any alternative network) put in the Pass 1 to improve performance since **70%** of the failure sample already failed at Pass 1

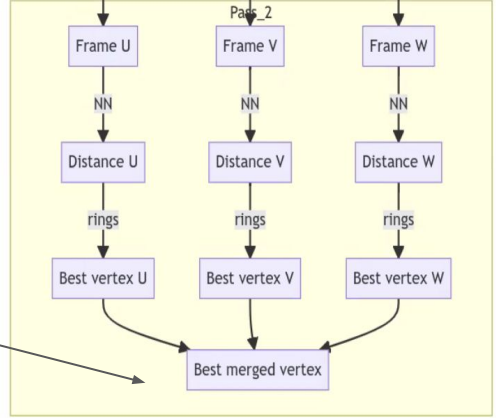


OR

2. Filter&Fix : after vtx+track reco

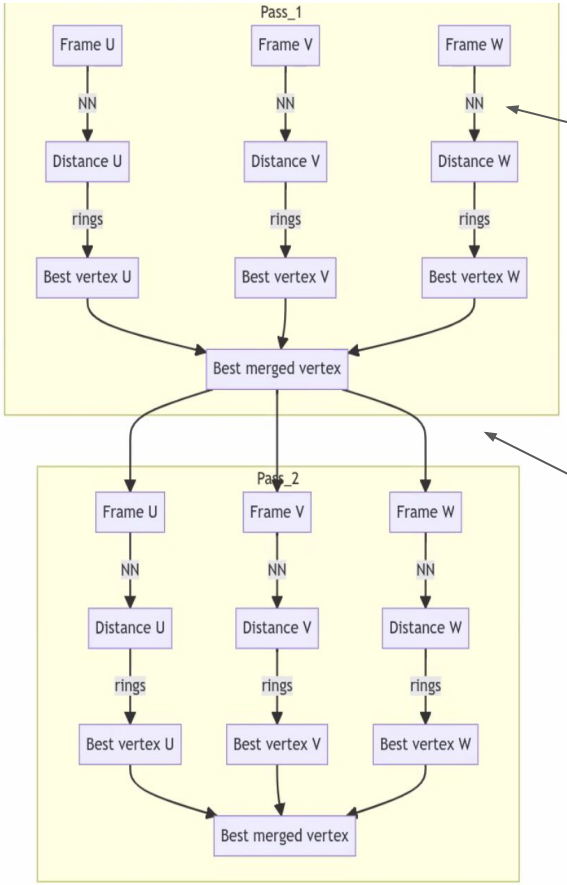
A) Use a filter to find bad vertices.
   - ex: check if any track is flipped using en deposition pattern;

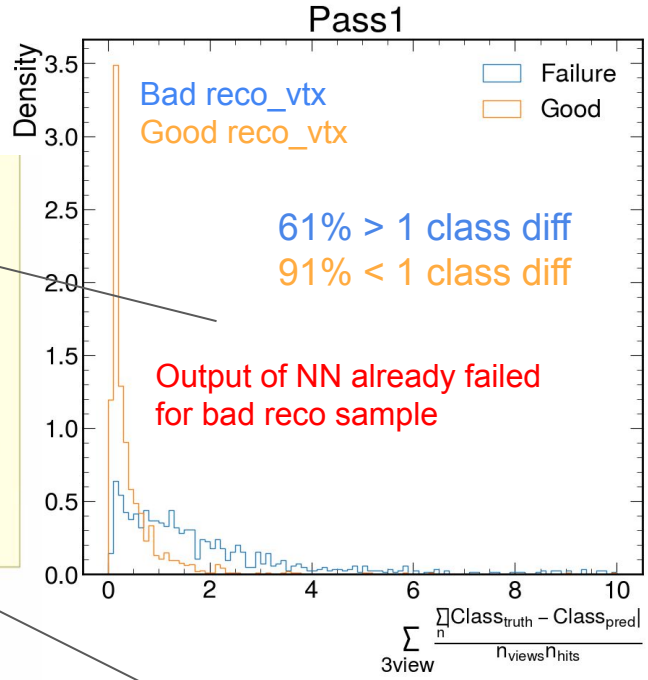
B) Run an algo (standard or ML) to find the right vtx

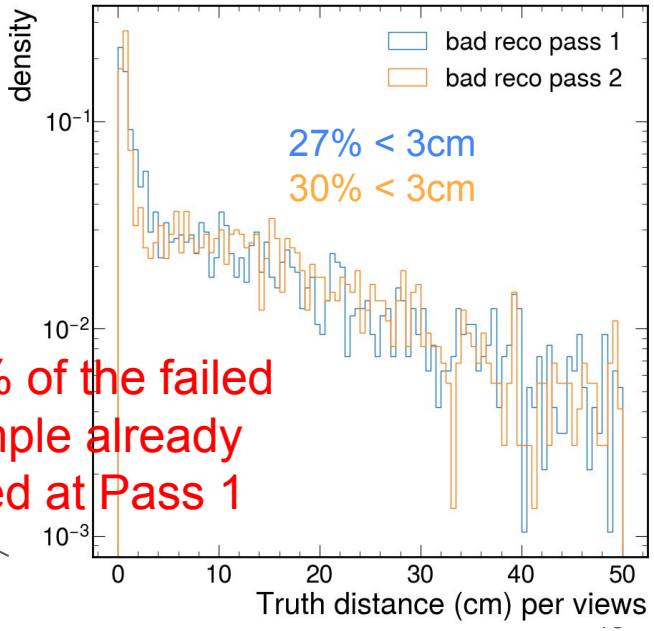


Input welcome!!!!!

Point of failure

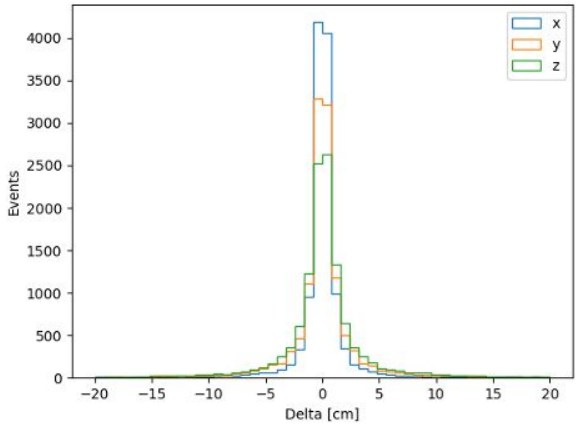


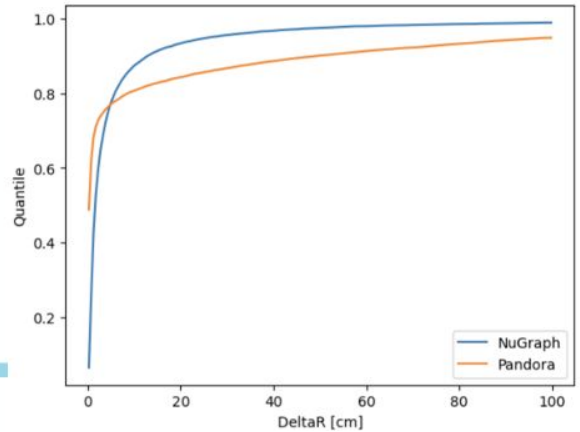




Vertex position classification

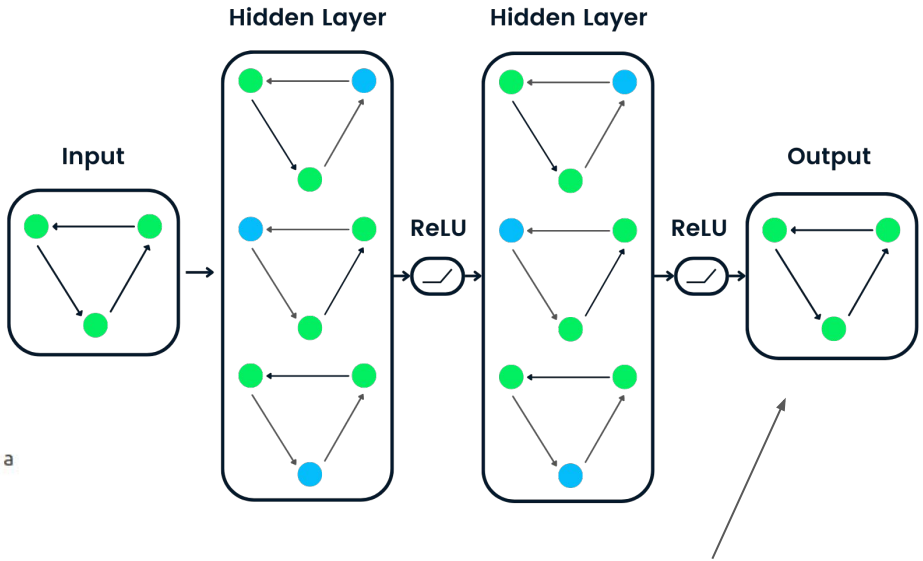
- Vertex position decoder using LSTM aggregator
- Preliminary work demonstrates that our GNN is able to identify the neutrino interaction position in 3D
  - currently O(cm) level resolution in each coordinate
- Compared to current vertex reconstruction this version shows worse percentile at low ΔR, but better at larger ΔR
  - worse at finding exact point, better at avoiding catastrophic errors
- Issues related to ground truth definition identified and being fixed, expect to achieve better results soon





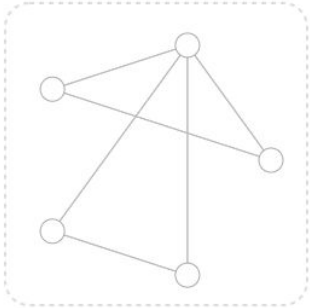
GNN

Main concept: process everything in graph



Graph:

To start, let's establish what a graph is. A graph represents the relations (*edges*) between a collection of entities (*nodes*).



V Vertex (or node) attributes
e.g., node identity, number of neighbors

E Edge (or link) attributes and directions
e.g., edge identity, edge weight

U Global (or master node) attributes
e.g., number of nodes, longest path

Need an extra step to get meaning output from the output graph