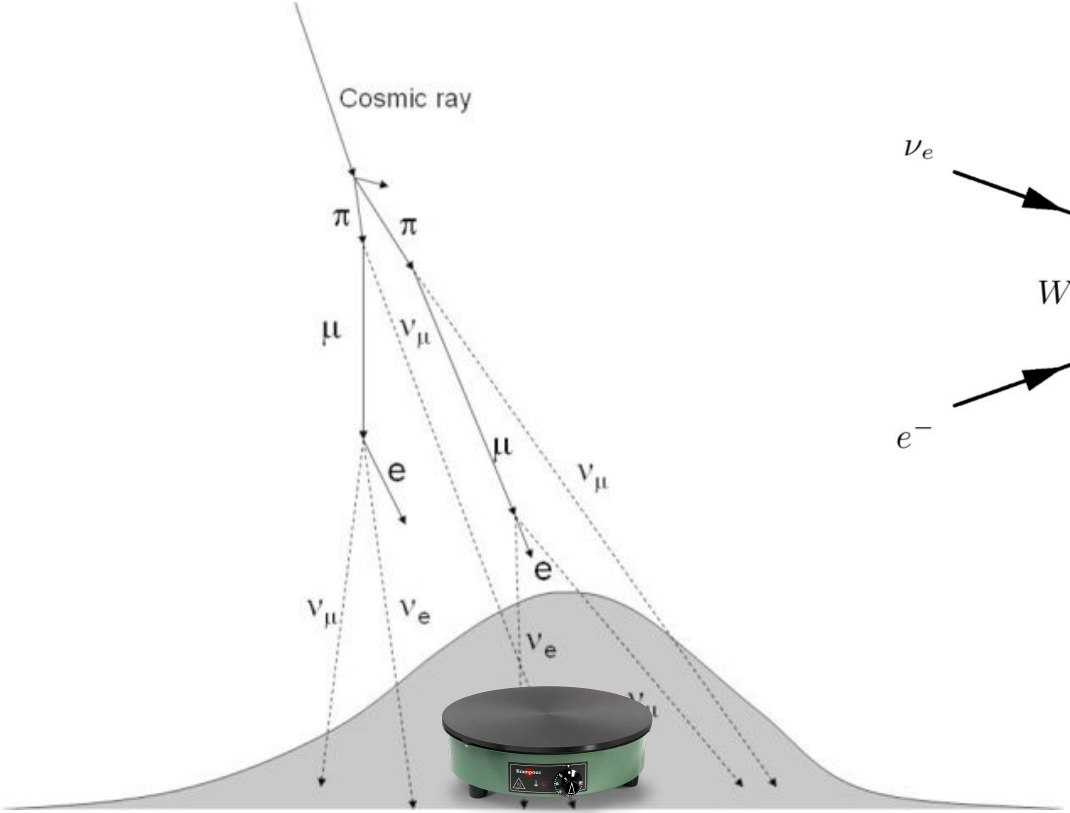
Vertex reconstruction for atmospheric neutrino

I Cheong Hong

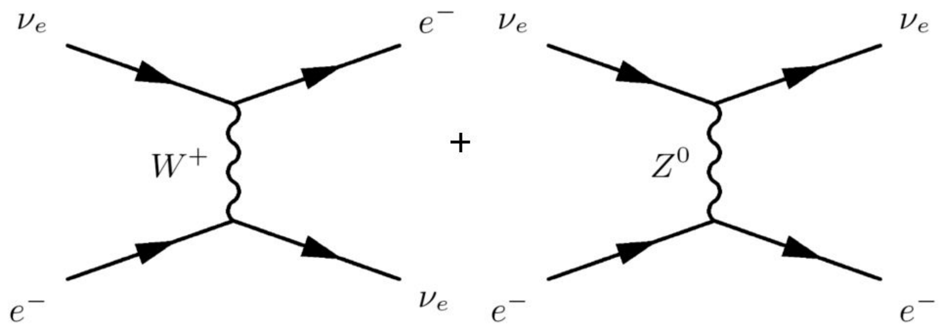
With help from P Granger & H. Souza

More details about the ML approach to vtx reco for atm nu in Pandora from Andy: https://indico.jlab.org/event/459/contributions/11757/attachments/9214/13375/ACha_CHEP_May23.pdf

A little introduction…



Neutrino source : atmospheric neutrinos



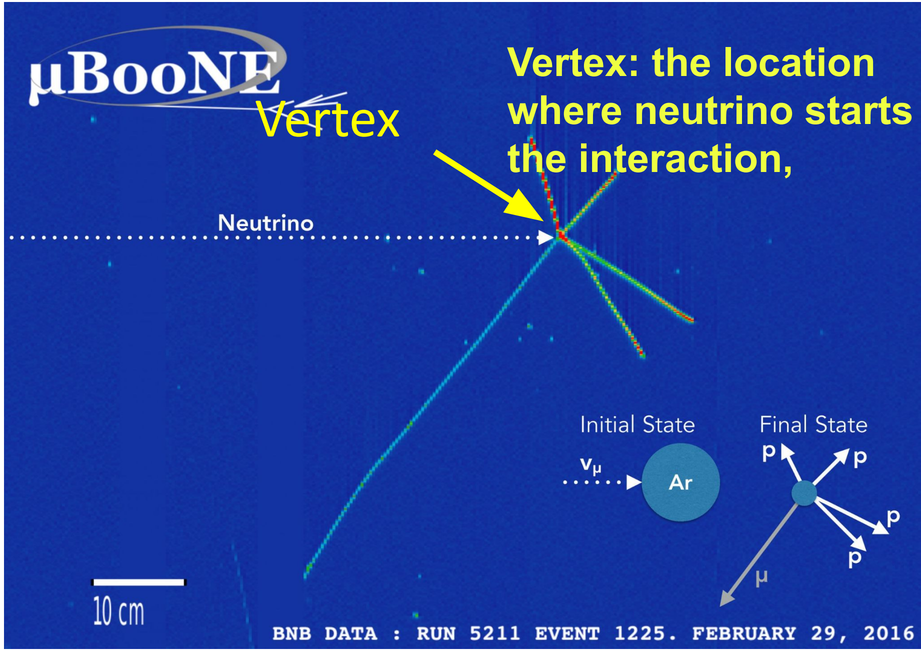
Left: Charged Current (CC)
Right: Neutral Current (NC)

Liquid Argon krampouz!

Project

- Goal: Improve the neutrino reconstruction using machine learning
  We focused on *vertex*, because it is the starting point of the reconstruction

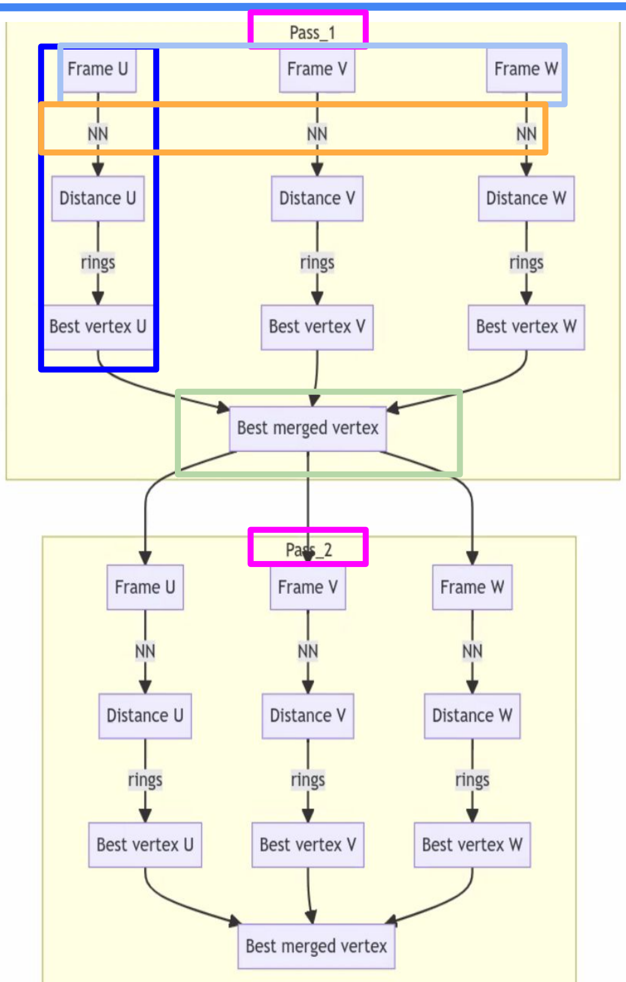
With a precise vertex reconstruction, we can improve the quality of the whole reconstruction



- Approach

1. Assess performance of the current Pandora vertexing algorithm and find its failure modes
2. Identify the causes of the failures
3. Find and implement solutions to address the failures

Pandora DL vertexing algorithm

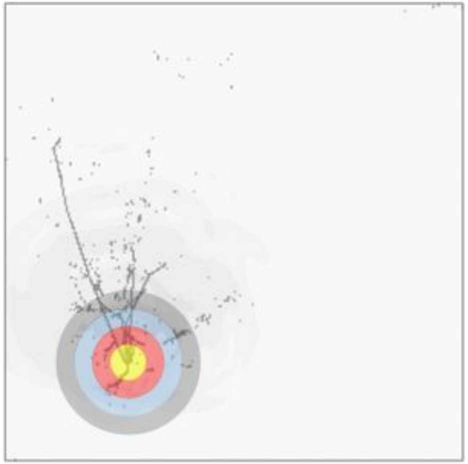


- **Two Passes**
  - Pass_2 ‘zooms-in’ into the vertices found by Pass_1
  - but are otherwise identical

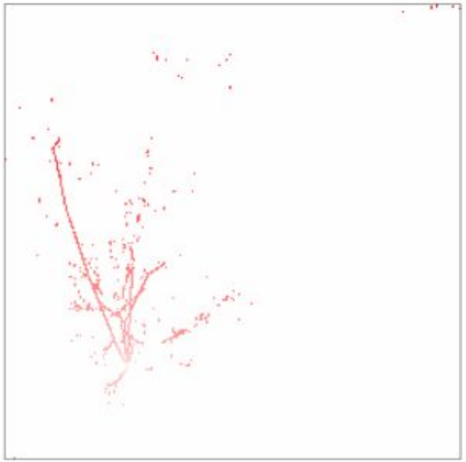
- **Each pass made of several steps**
  - same steps for each frame and Pass_1 & Pass_2
  - first step: a deep learning algo
  - final 3D vertex found by merging info from all 3 frames

Pandora DL vertexing algorithm

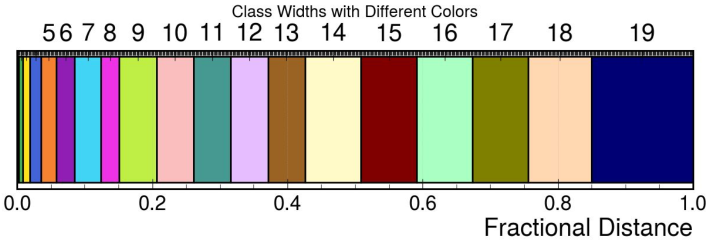
In training hits are assigned a class according to distance from true vertex



Network trained to learn those distances from input images



Network infers hit distances and resultant heat map isolates candidate vertex



"Draw rings"
or…..
making crepes

Our approach

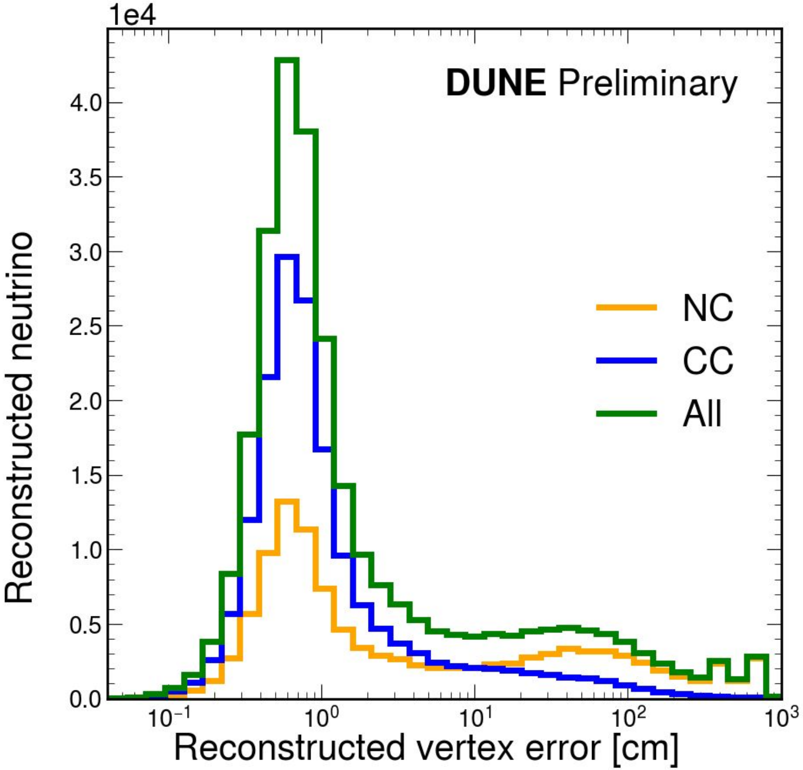
1. **Assess the general performance of the present algo**
    a. failure rate
    b. type of failures

2. **Look individually at each step of the algo**
3. **Decide where/how to start fixing things**

Sample used:

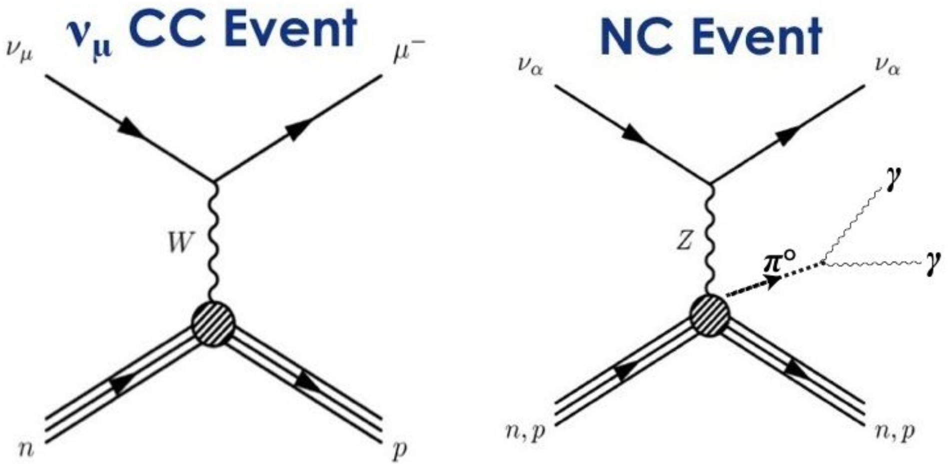
- 10k atmospheric w/ 0.1->100GeV. Flux exp(-2.5*E)
- unless specified, CC+NN events are shown
- network training was done on an atm sample (60K events)

General performance: **resolution**



*Dataset: ~160k atm sample

- Many atmospheric neutrino events, at present, have their vertex reconstructed more than 3cm away from the true position
  - 16% of CC events
  - 42% of NC events



Caption: Distribution of the reconstructed vertex position error, for neutral current (NC) and charged current (CC) events, as well as their sum. All neutrino and antineutrino flavors are included in the samples.

Difference in performance: can be explained by the nature of the interaction

Our approach

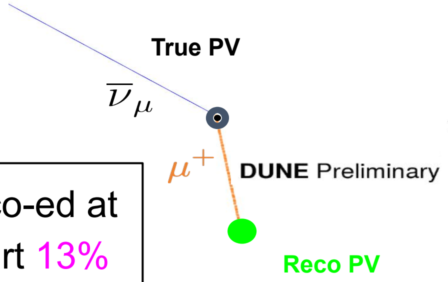
1. **Assess the general performance of the present algo**
    a. failure rate
    b. type of failures

2. **Look individually at each step of the algo**

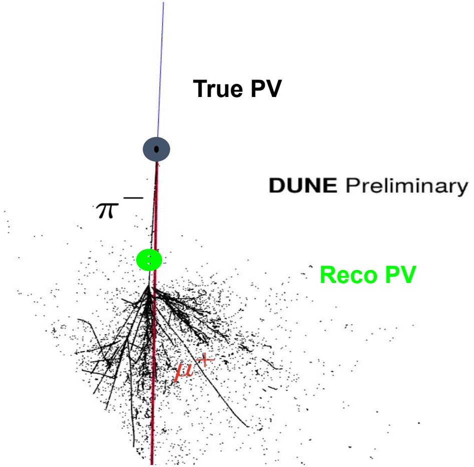
3. **Decide where/how to start fixing things**

Failure mode

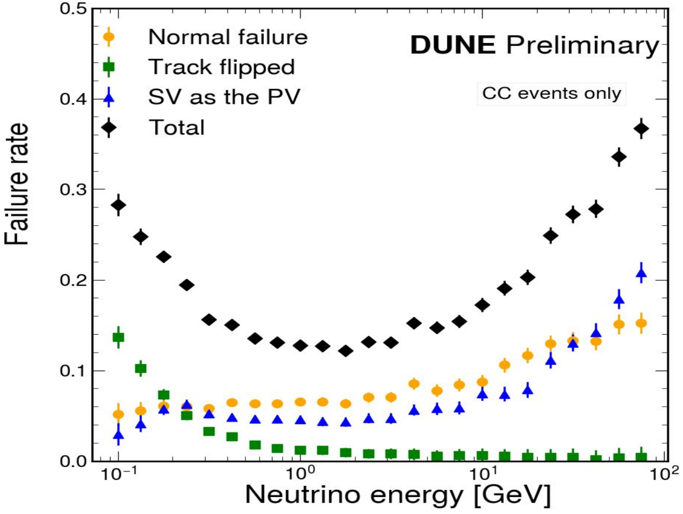
A) ‘**Normal failure**’: vtx reco-ed >3cm away from truth, but in the right direction 48%



B) ‘**Track flipped**’: PV reco-ed at end of track instead of start 13%



C) ‘**SV as PV**’: a secondary vtx is wrongly identified as PV 34%



$$y = \frac{\text{number of events with that failure type}}{\text{number of events}}$$

Caption: Energy dependence of different types of failures of the vertex reconstruction. Only CC events are used for this plot, but all neutrino and antineutrino flavors are included.

Our approach

1. **Assess the general performance of the present algo**
    a. failure rate
    b. type of failures

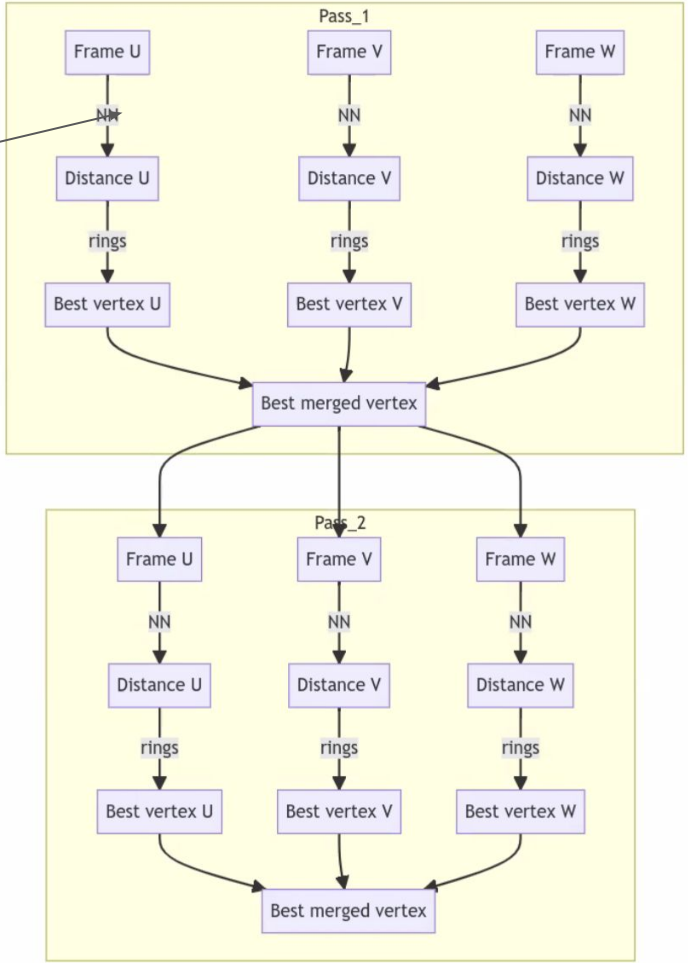
2. **Look at each step of the algo**

3. **Decide where to start fixing things**

Algorithm step-by-step

- Main/first source of failure: the NN algo in Pass 1

→ ~70% of the failures are seen already in Pass 1 NN ← the vertex is put in the wrong position by the network from the very beginning



Main point of failure of the algorithm

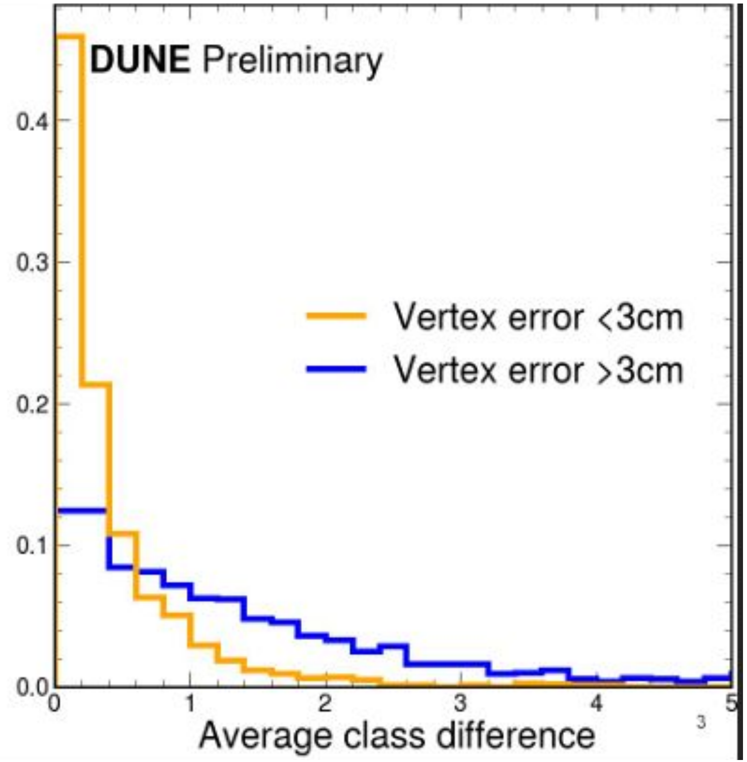
Bad reco: 45%
> 1 class diff
Good reco: 9%
> 1 class diff

The main point of failure happens at the very beginning of the vertex algorithm, at the NN stage: the class assigned to hits is wrong.

Caption: Average over hits, and over the three views (U, V, W), of the difference in assigned class considering the truth vertex position ($class^{truth}$) and the predicted class for each hit ($class^{pred}$).
It tries to quantify how wrong the raw output from the network (ie, the distance class assignments) is from the very beginning of the vertex reconstruction algorithm.

$$\text{Average class difference} = \frac{1}{n_{views}} \sum_{j=1}^{n_{views}} \left(\frac{1}{H_j} \sum_{i=1}^{H_j} |class_{i,j}^{truth} - class_{i,j}^{prod}| \right)$$

- n_{views} =3 is the total number of views (U, V, W)
- H is the total number of hits in each of the 3 views



x-axis: No units (average of an average of a difference in classes that represent fractions of an image size expressed in pixels)

Our approach

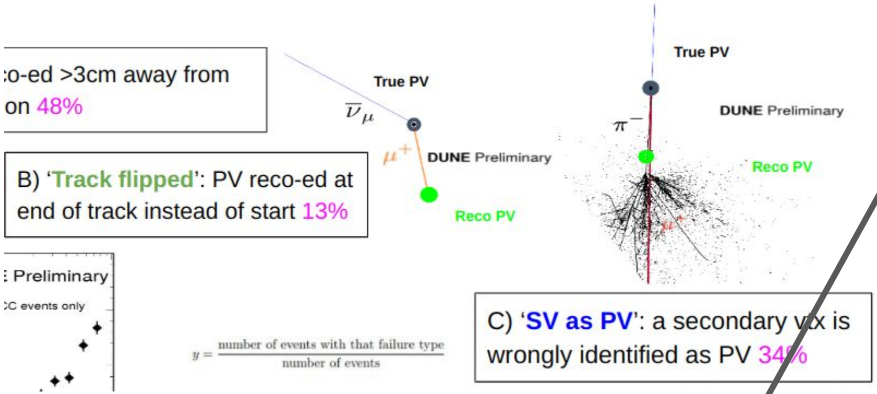
1. **Assess the general performance of the present algo**
    a. failure rate
    b. type of failures
    c. topology of the events in which vtx reco fails

2. **Look individually at each step of the algo**

3. **Decide where how to start fixing things**

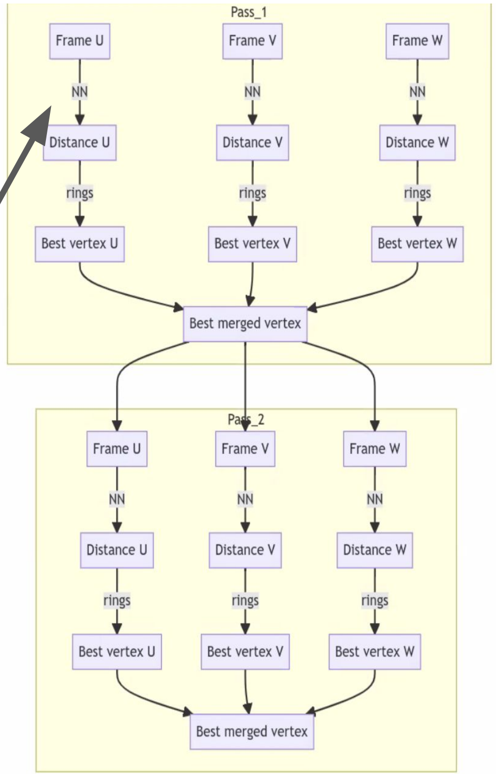
Solutions to address the failures

A bit remind of failure modes:



1. Modify current algo:
   a. increase the trainig sample
   b. replace Pandora's U-net with a Graph neural network (GNN) (to help w/ all failure modes)

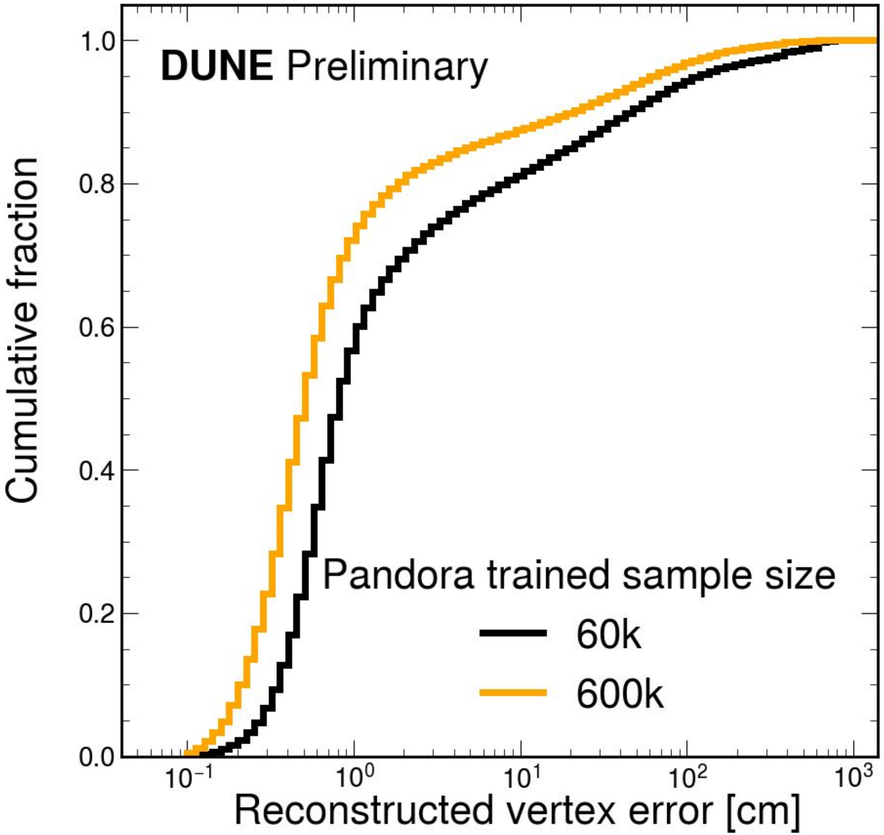
2. Add a filter at the end of Step 2, to select failed-vtx candidates, and combine with an additional algorithm that fix it (to help w/ the flipped track and SV type of failures)



Solution 2: at the end

Flowchart of current DL vertex reconstruction

Modify algo – 1a: Train the network with more events

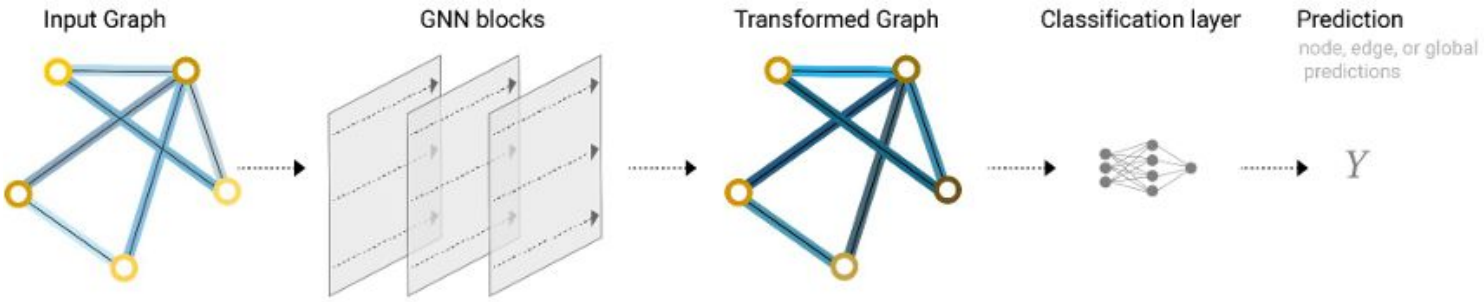


Reco-Truth vtx distance	<1cm	<3cm	<10cm	<100cm
Events used for CNN training: 60k / 600k				
All	56% / 72%	73% / 83%	81% / 87%	94% / 97%

Caption: Cumulative histogram of the error in vertex reconstruction, when training the U-Net with two different event sample sizes.

Modify algo – 1b: replace U-Net w/ a Graph Neural Network

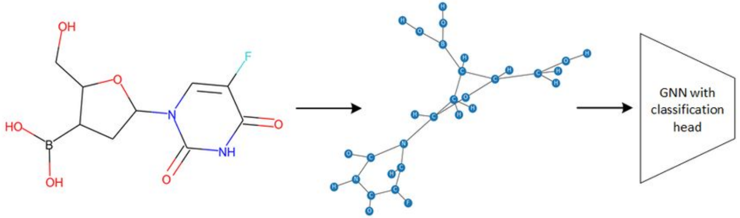
Just a little bit intro for GNN….



In order to use a GNN for vertex reco, we must also have a method to transform hits to graph (call it node-connection method in the slides afterwards)

Still on trial & error stage

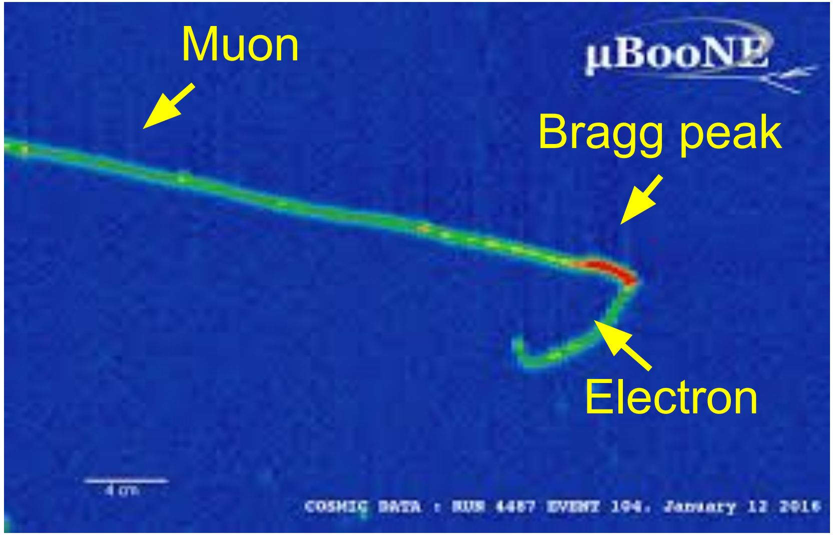
Benefit: GNN suppose to understand the structure of interaction better than CNN + without the limitation of pixelized input



Examples: Moluecular as a graph

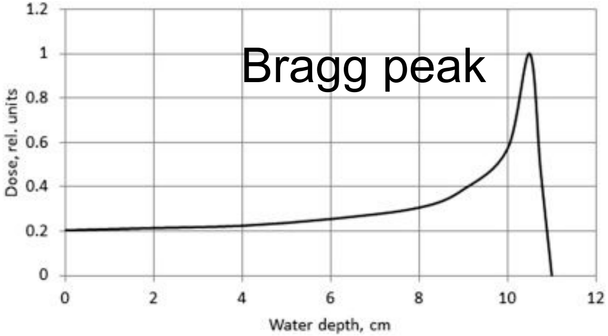
Filter failed events at the end of current algo (option 2)

- Create a filter using track energy information



If we can infer the direction of the track, we are able to fix failure mode b & c by checking if a track is flipped or not

We can use the Bragg peak of the muon track, to determine the direction of the muon track



Summary

1. **Assess the general performance of the present algo**
    a. failure rate
    b. type of failures

2. **Look individually at each step of the algo**
    a. main point of failure: NN part

3. **Decide where how to start fixing things**
    a. Training with more samples
    b. Graph Nueral network
    c. Flipping track filter