Vertex finding with Pandora Deep Learning

Andy Chappell

11/04/2022

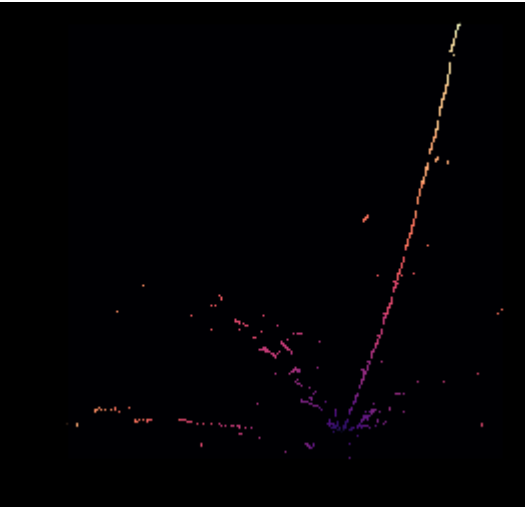
FD Sim/Reco Meeting

DEEP UNDERGROUND NEUTRINO EXPERIMENT

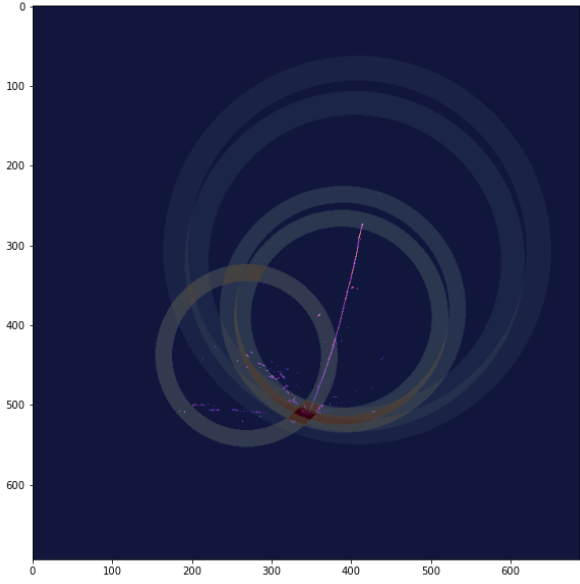
WARWICK

Reminder

- Identifying the neutrino interaction vertex seems like something a CNN should be good at
- But defining a single point/region as the signal means you need a huge number of events to train the network



- Instead encode the truth in all hits in each event, by describing the distance each hit is from the interaction vertex
- Gives the network a direct handle on how all of the information in a single event relates back to the vertex location
- Post-process the distance classification to resolve the interaction vertex



Evolving the network

WARWICK

- Preliminary results were quite reasonable (dr68 ~2.6 cm), particularly with the lack of tuning, slightly under-performing the existing vertex finding (dr68 ~2.1 cm)
- Problem 1: Resolution
  - Events are sampled to a 256 x 256 pixel image per view
  - If the event spans more than a couple of metres the pixels begin to represent large regions, limiting the network resolution
- Adding a second pass
  - To address this problem, take the result of the first pass and zoom in to this region to identify the vertex at higher precision

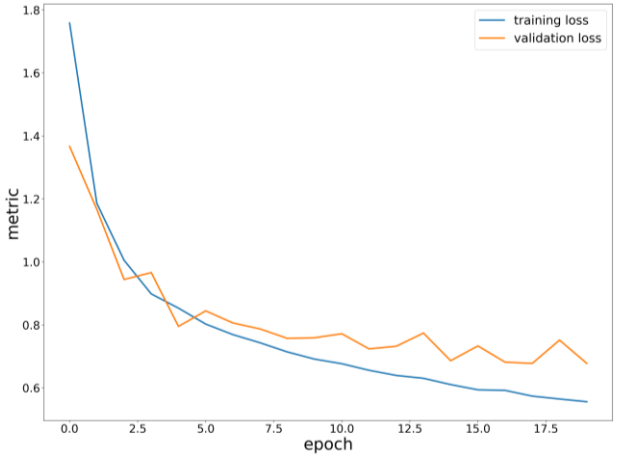
Training the second pass

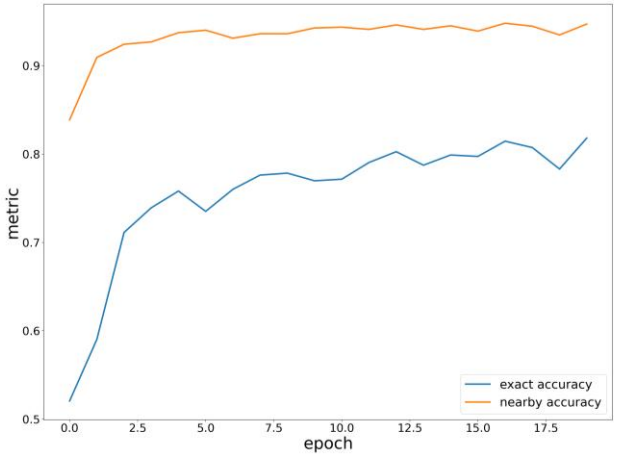
WARWICK

- We want this to be quick, so we compromise
  - Smaller images at 128 x 128 pixels
  - 64 cm x 64 cm event region, allowing for 0.5 cm resolution
- If the first pass vertex is off by much more than 32 cm, we're probably out of luck
  - Will comment on mitigation later
- To define the training dataset I take a perturbed version of the true vertex
  - Gaussian (0 cm, 15 cm) perturbation in X and Z
  - Treat this as the centre of the image
  - Our first pass reconstruction will be imperfect and we want to ensure the network doesn't simply learn to pick the centre of the image in the second pass
  - A check is applied to ensure that the region contains hits
  - No check on true vertex containment because this technique can, in principle, find uncontained vertices, so we should let it try

Training the second pass

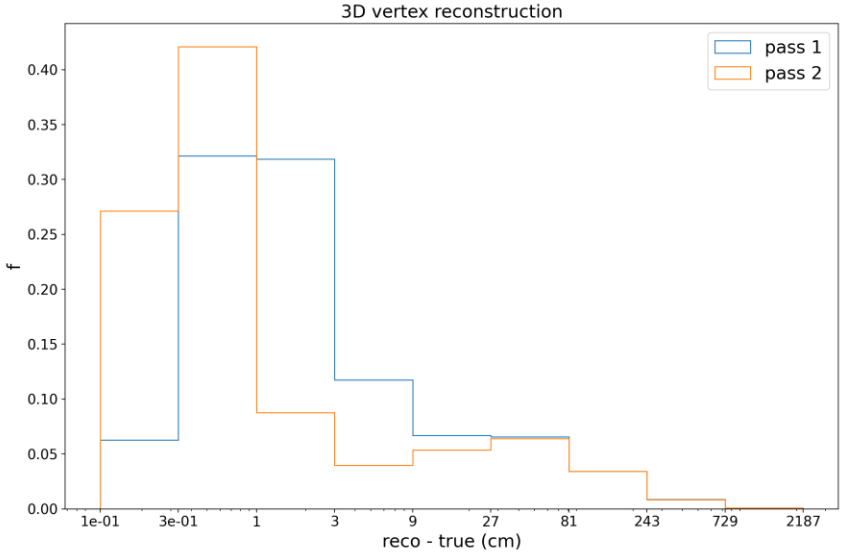
- Second pass network appears to train well
- W view indicates there may be scope for further improvement with tweaks to the training procedure
- Given 19 distance classes accuracy is quite good
  - ~80% for exact class matches
  - ~94% for exact or adjacent class match

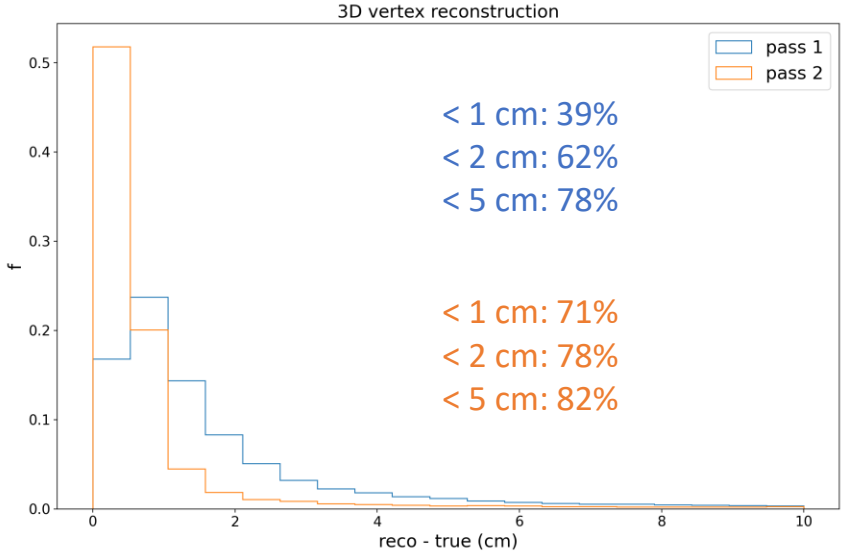




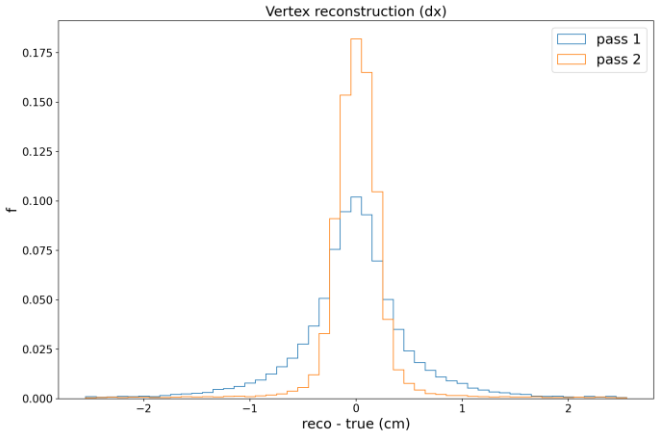
Reco – True Vertex Deltas

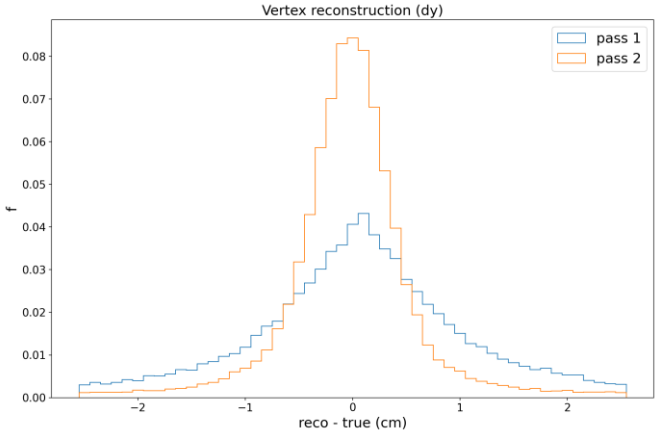
- Compared first and second pass performance on a 50,000 event sample
  - Even split ν_μ and ν_e MCC 11 1x2x6
- Pass 1 dr68: 2.6 cm
- Pass 2 dr68: 0.9 cm
- Unsurprisingly, performance is similar beyond about 10 cm, as pass 1 sets the scale

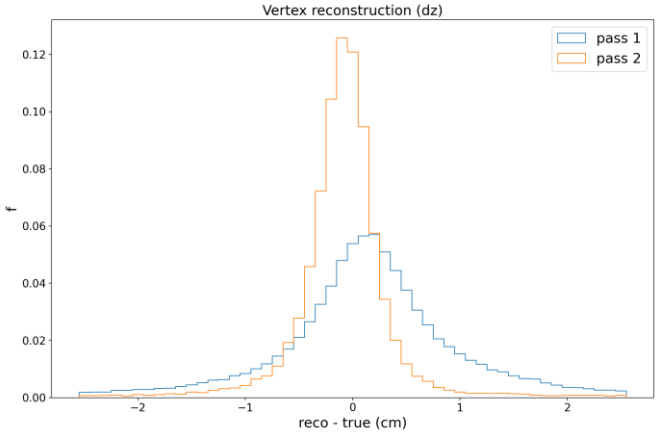




Reco – True Vertex Deltas





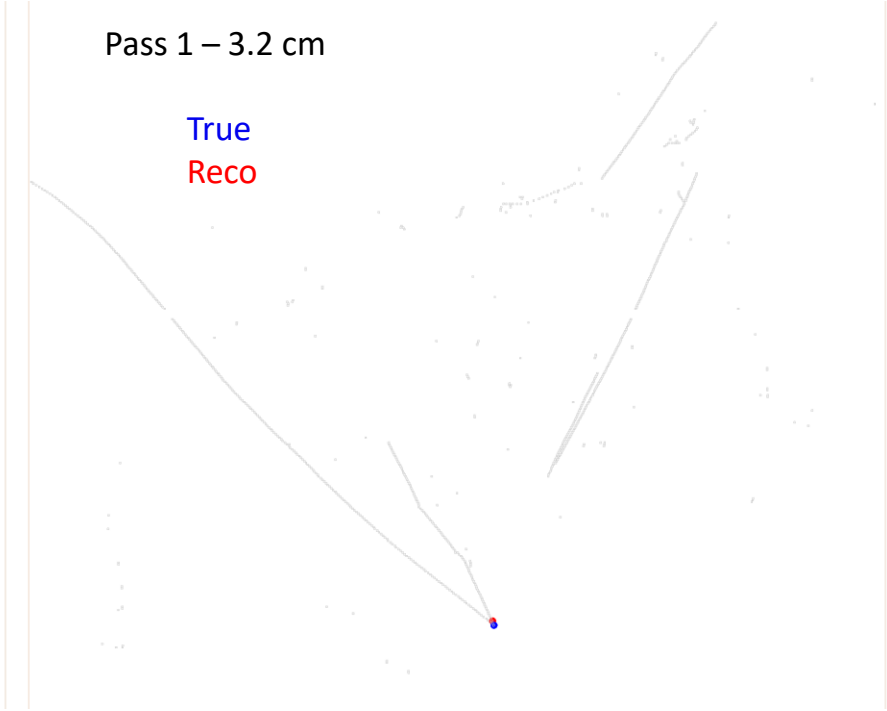


- Reco – True plots centred on zero in x and y
- Pass 2 z shows bias to low reconstructed z
  - Peak at ~-0.1 cm
- Interestingly pass 1 shows a slightly larger bias to high reconstructed z
  - Peak at ~0.2 cm

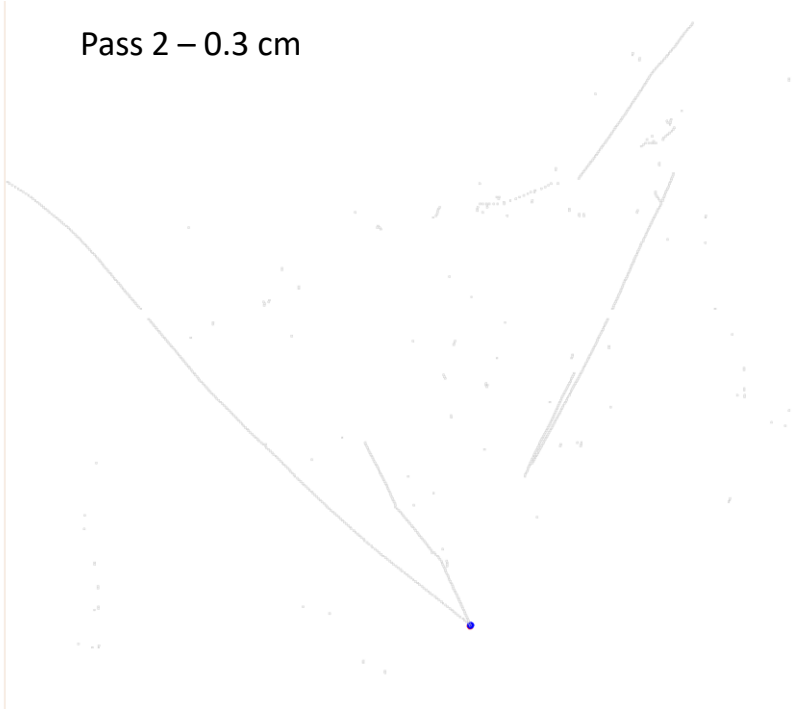
Example

Pass 1 – 3.2 cm

True
Reco



Pass 2 – 0.3 cm



Next steps

WARWICK

- Dealing with large pass 1 errors
  - Picking entirely the wrong region in pass 1 means pass 2 isn't too helpful
  - Identify a few candidates in pass 1 and zoom in on each
- Consistency checks between passes
  - Sometimes the zoom region can be a bit sparse/messy
  - This can make the second pass more difficult to assess than the first
  - Allow 2D->3D matching to consider both pass 1 and pass 2 results
- BDT integration
  - Might the vertex position information prove a complementary variable to other BDT variables?
  - This is the approach Jhanzeb was looking into when working on vertexing
- Atmospheric neutrino vertexing and vertexing in the Vertical Drift geometry
- Longer term
  - Secondary vertexing