Neutrino interaction vertex reconstruction at DUNE

Andy Chappell

08/12/2022

Warwick EPP Seminar

DEEP UNDERGROUND NEUTRINO EXPERIMENT

WARWICK

DUNE and LArTPC reconstruction

DUNE’s science program

WARWICK

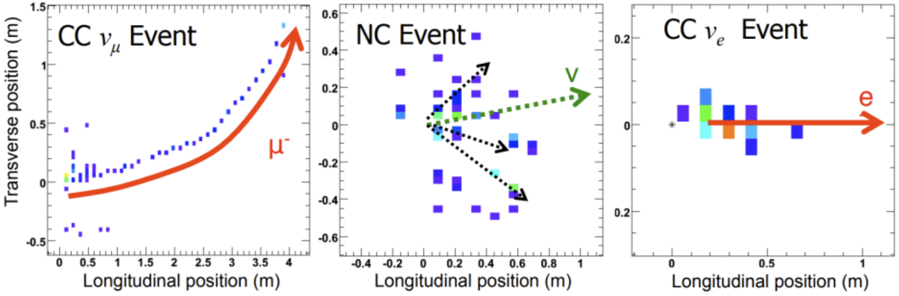
- DUNE aims to address a number of key questions in particle physics and astrophysics
  - Why do we live in a matter dominated Universe?
  - What are the dynamics of supernova neutrino bursts?
  - Do protons decay?
- A single experiment to explore the three-flavour model of neutrino physics
  - CP phase
  - Neutrino mass ordering
  - Precision mixing parameter measurement

The promise of LArTPCs

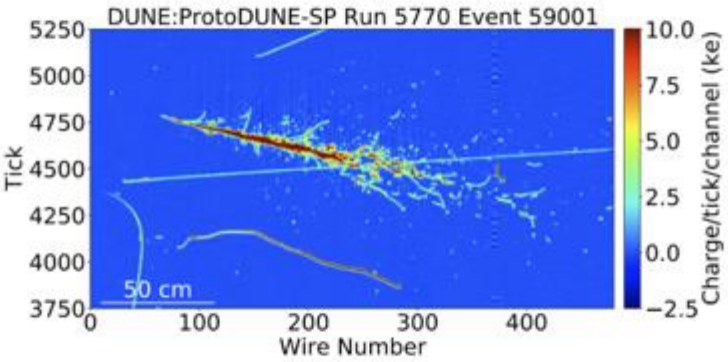
WARWICK

- Imaging detectors for neutrino experiments have undergone substantial evolution since MINOS
- Change in scale
  - 5.4 kton at MINOS
  - 17.5 kton (10 kton fiducial) per module at DUNE
- From shower activity as a monolithic blob to mm precision energy deposition
  - Separate electron and photon showers

MINOS



[J. Evans 2013 arXiv:1307.0721](https://arxiv.org/abs/1307.0721)

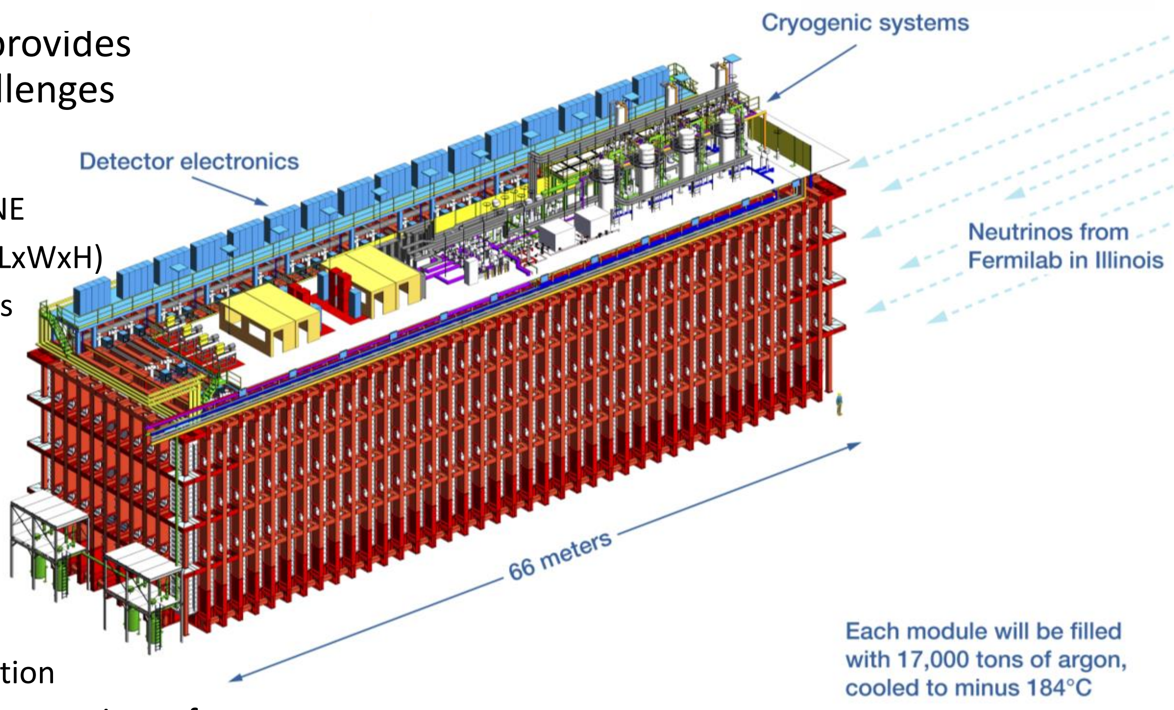


B. Abi et al 2020 JINST 15 P12004

The challenge of LArTPCs

WARWICK

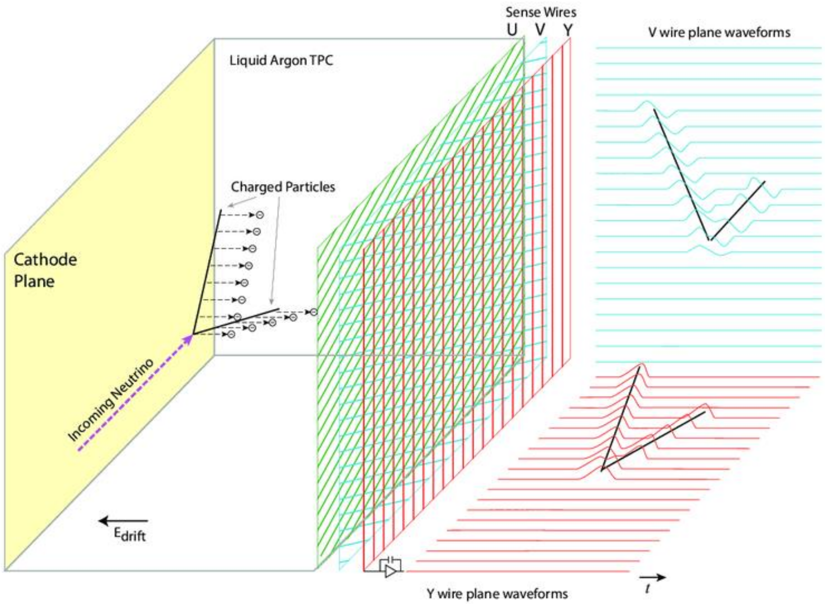
- Increased scale and detail provides opportunities, but also challenges
- Change in scale
  - 17.5kton per module at DUNE
  - Cryostat 66m x 19m x 18m (LxWxH)
  - No 3D readout, 3x 2D images



- Change in resolution
  - mm precision energy deposition
  - Far more demanding of reconstruction software

LArTPC operation

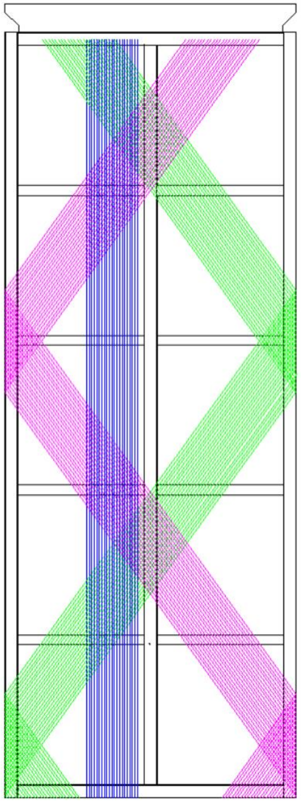
- Fully active interaction medium
- Charged particles ionize argon atoms to produce drift electrons (and scintillation light) along the particle trajectory
- Electrons drift in the electric field (500 V/cm at DUNE)
- Three anode wire planes record the deposited charge using wires of different orientations
- Result is three different 2D projections of the charged particles in the interaction

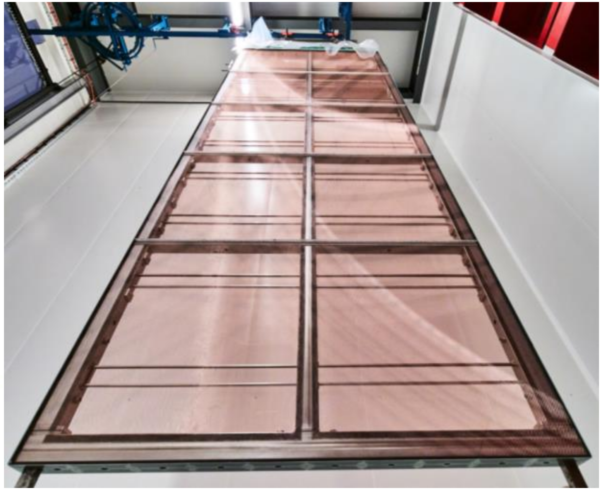


Horizontal Drift Anode Plane Assembly (APA)

WARWICK

- A single frame holds three parallel planes of wires with different orientations
- Two induction planes (U and V) at 35.7^{o} and a collection plane (W) with vertical wires
- Wire pitch is 4.67mm for induction planes and 4.79mm for collection plane
- 2560 readout channels per APA, 150 APAs per far detector module, with production lead by the UK

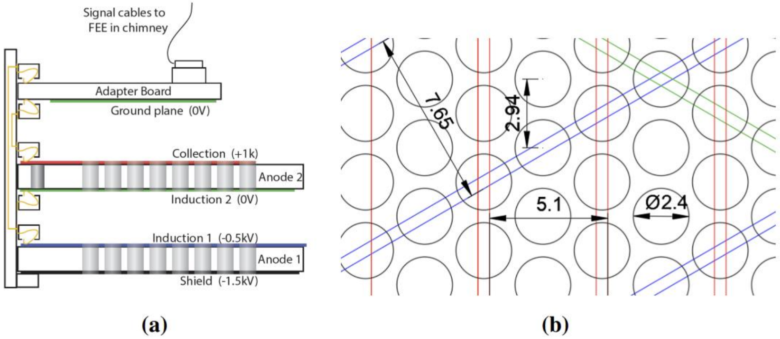




ProtoDUNE-SP APA. Credit: J.M. Ordan

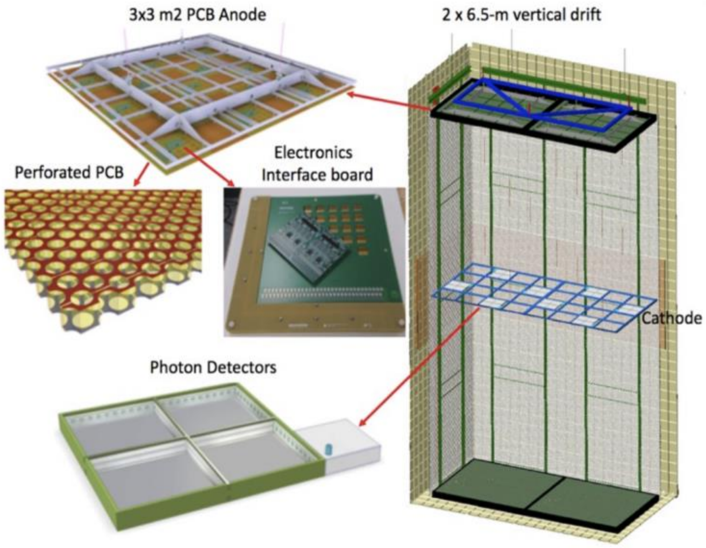
Vertical Drift APA

- A single frame holds upto 6 charge readout planes (CRP), each comprising three parallel planes of perforated printed circuit boards (PCB)
- 80 CRPs at top and bottom of detector, each measuring 3.4m x 3m
- Baseline strip orientation of induction planes at ±30^{o}, collection plane at 90^{o} to beam direction



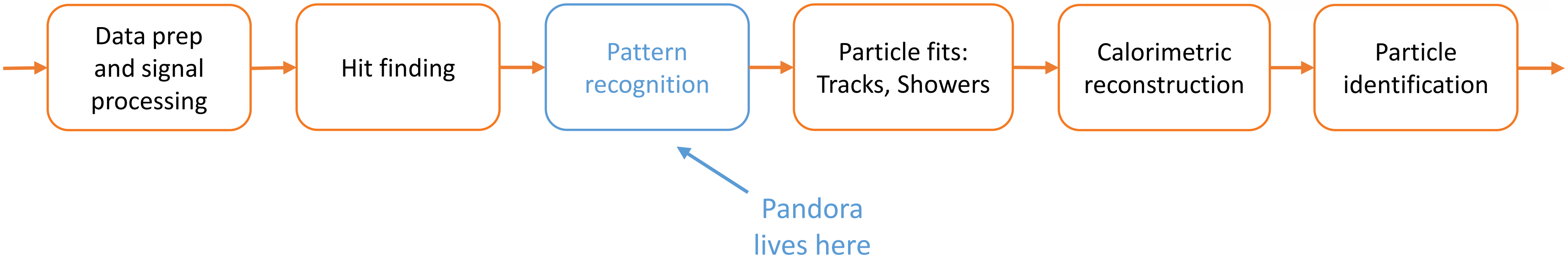
Side view of PCB stack

Anode strip pattern



O. Lantwin 2022 arXiv:2211.11339

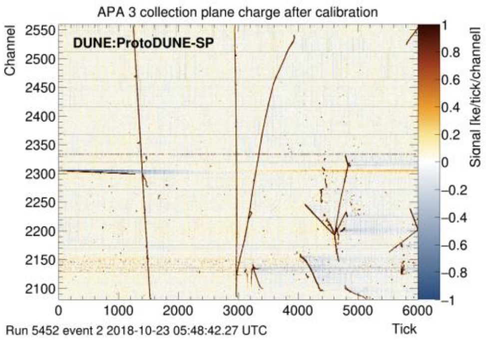
Reconstruction chain



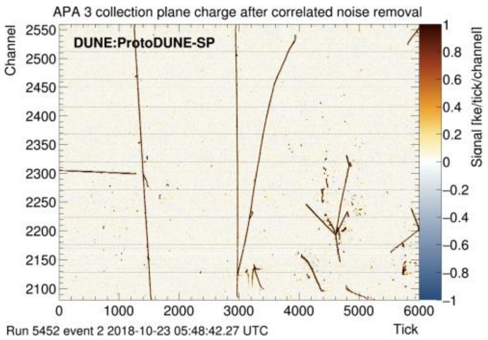
- Pandora pattern recognition sits in a chain of reconstruction steps
  - The raw detector signals are processed, and discrete hits extracted
  - The discrete hits are Pandora’s inputs
  - Pandora then clusters these hits, constructing particle hierarchies
  - These particle hierarchies form the inputs to subsequent high-level reconstruction tasks

Reconstruction chain – data preparation

- Data preparation converts the Analog-to-Digital Converter (ADC) waveform (ADC count per tick - 6000 ticks per 3 ms readout in ProtoDUNE-SP) to a charge waveform
  - Evaluate pedestals – per channel voltages must be kept in device range, so average ADC count absent a signal varies per channel. Fit a Gaussian to the ADC count across 6000 ticks, mean is pedestal, RMS is initial noise estimate.
  - Charge calibration – multiply the pedestal subtracted ADC count by gain determined from charge calibration (injection of a known charge in short pulses)
  - Noise suppression from, for example, low-voltage power supply



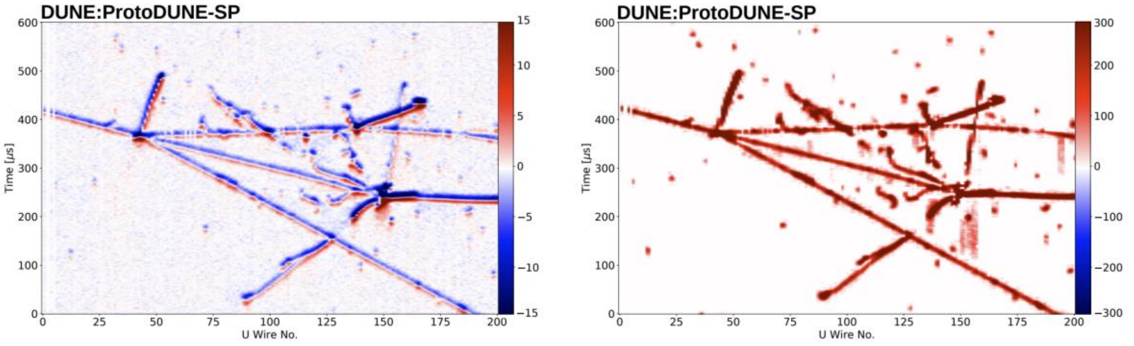
Pedestal subtracted and calibrated
https://arxiv.org/abs/2007.06722v3



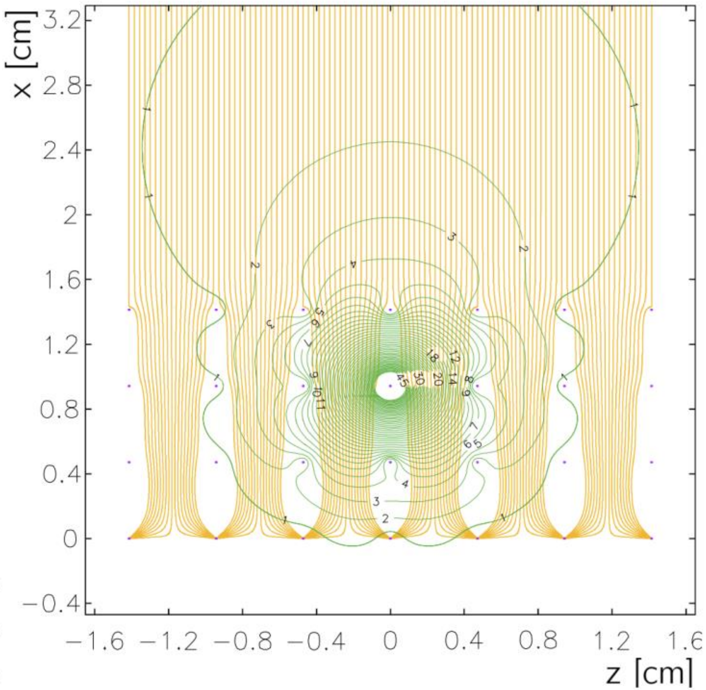
Noise removed

Reconstruction chain – signal processing

- Induced current on a wire contains non-negligible contributions from moving charges upto 10cm in front of the wire plane and 10 wires away
- 2D deconvolution removes both spatial and time components of the field response to extract the ionization electron distribution



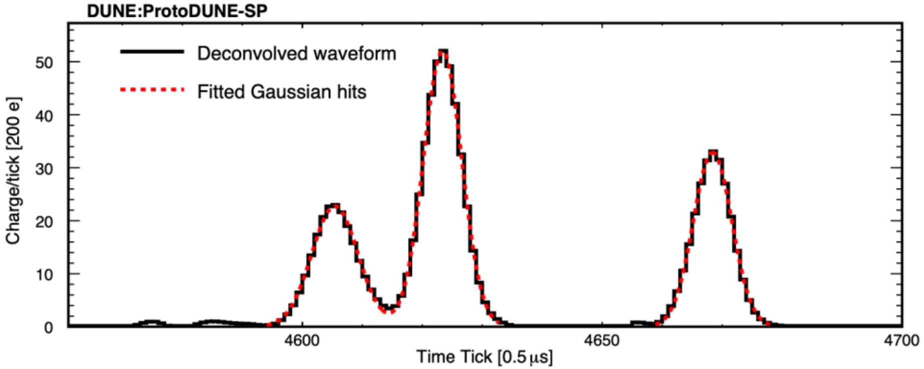
(a) After Noise Filtering (b) After Deconvolution



Drift electron paths to collection plane (yellow) and equi-potentials about wire of interest (green)
https://arxiv.org/abs/2007.06722v3

Reconstruction chain – hit finding

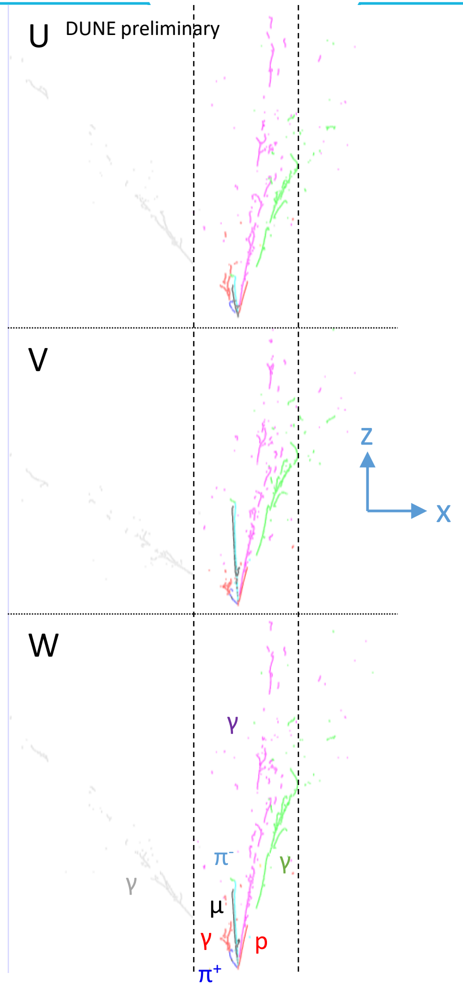
- Given the deconvolved waveform on a single wire look to find the peaks in that waveform
- Fitted Gaussians define the arrival time and width of discrete hits
- Peak separation is not always so clean
  - Particle trajectories parallel to the wire plane can leave very wide charge deposits
- These hits form the inputs to Pandora's pattern recognition



https://arxiv.org/abs/2007.06722v3

Pattern recognition - inputs

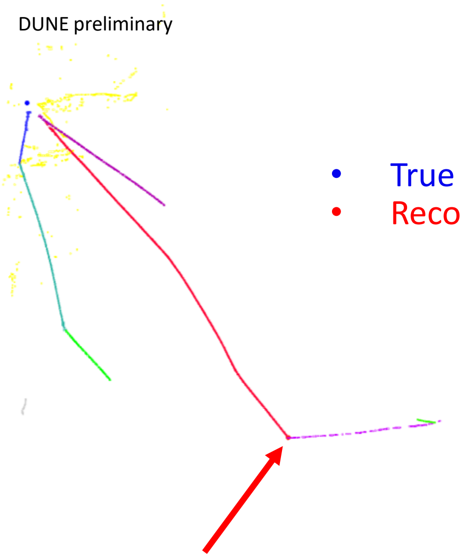
- 2D hits from each wire/strip plane
  - Drift coordinate, x, is common to all planes
  - PID indicated here for reference only
  - Vertex is quite clear here, but downstream reconstruction remains challenging
  - Colinear muon and pi minus
  - Little separation between pi plus and one downstream photon
  - Small opening angle between two high energy photons from pi zero decay
- For a broader discussion of Pandora's pattern recognition see Maria Brigida's EPP seminar



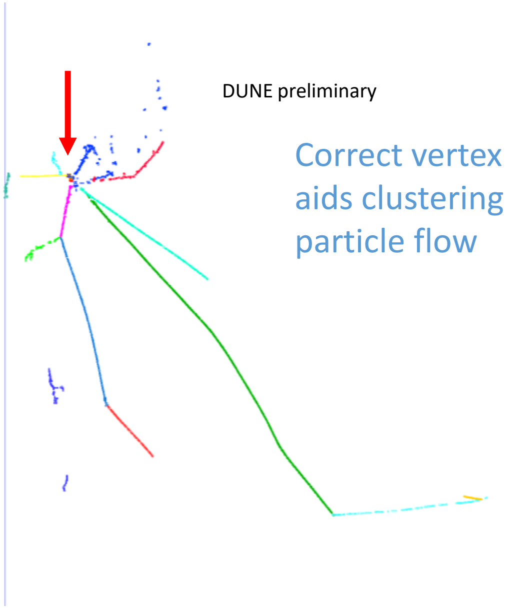
Vertex finding

Vertex finding

- A critical component of LArTPC reconstruction is identification of the neutrino interaction vertex
- Downstream reconstruction algorithms rely on vertex location to make clustering decisions and determine the particle flow



Incorrect vertex placement leads to mis-clustering and incorrect particle flow



Correct vertex placement aids clustering and improves particle flow

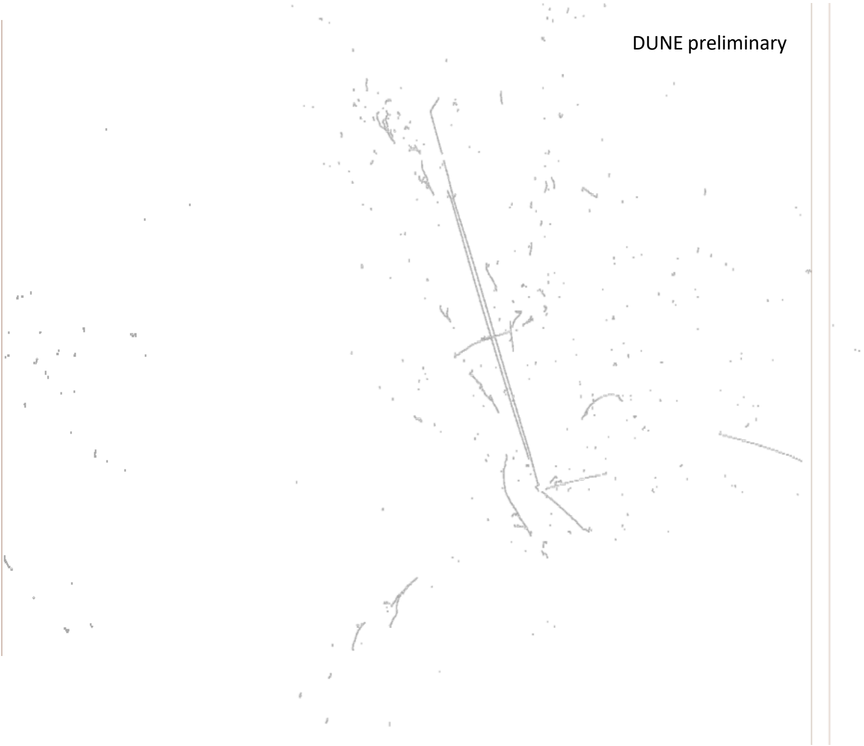
Machine learning

- LArTPCs are imaging detectors and vertex identification is a pattern recognition task at which humans excel
- Advances in convolutional neural networks in the past 7 years have seen machines begin to outperform humans in many pattern recognition tasks and so CNNs look like a natural choice to tackle the problem
- There are complications, however:
  - DUNE’s neutrino interaction topologies are varied and complex
  - Readout is 3x 2D images, so overlapping projected trajectories are common
  - At “full resolution” DUNE images are often far larger than typical input images sizes for CNNs

Humans versus machines

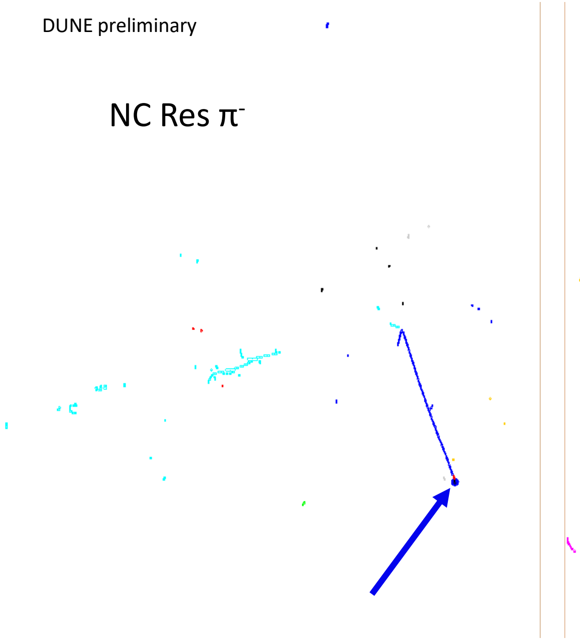
WARWICK

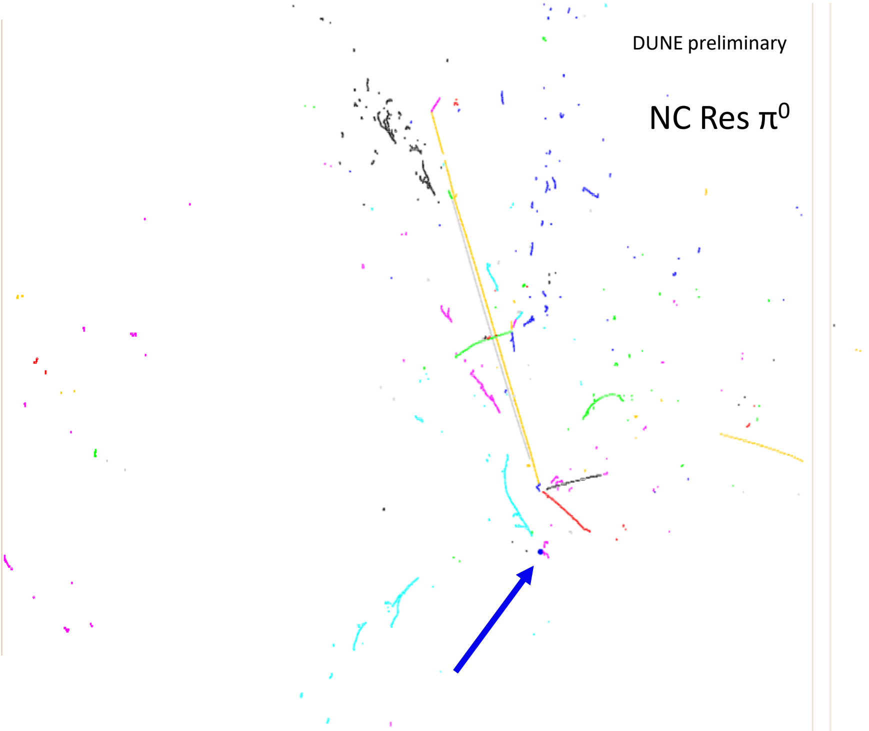
- A couple of events where precision vertexing is not so simple, even for humans



Humans versus machines

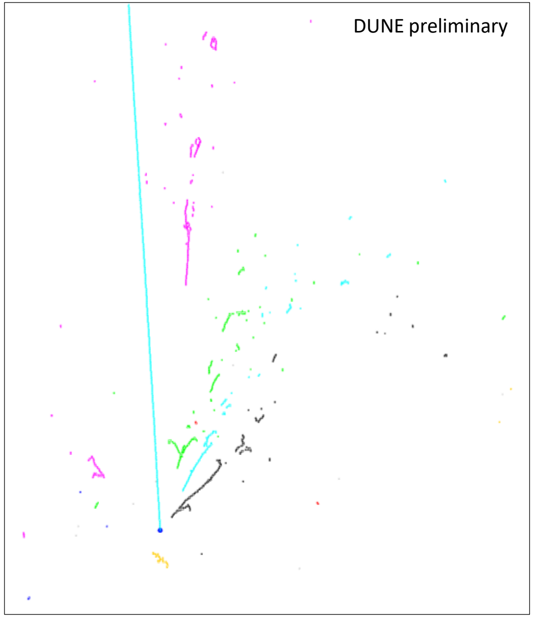
- A couple of events where precision vertexing is not so simple, even for humans





Truth definition

- How should we define the truth?
- CNNs are excellent for classification, but tend not to be great for regression (e.g. Fischer et al 2015)
  - Getting a CNN to learn two continuous variables from highly varied input hits is unlikely to succeed
- Can reframe the problem in the form of semantic segmentation

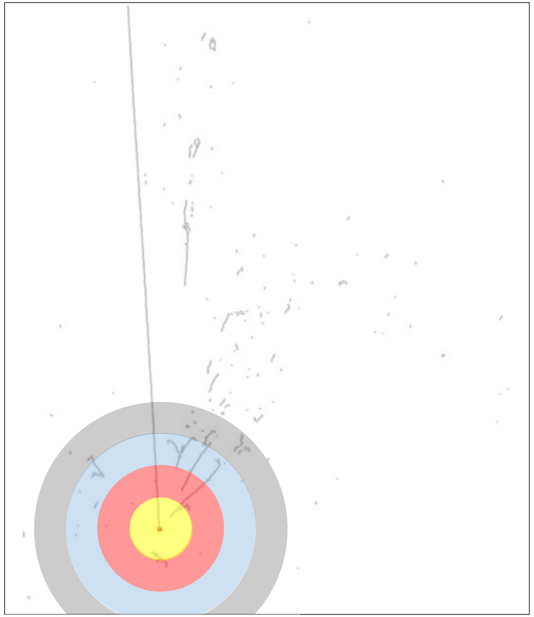


$\bar{\nu}_{\mu} + \mathrm{Ar} \rightarrow \mu^{+} + 3\pi^{0}$

Truth definition

WARWICK

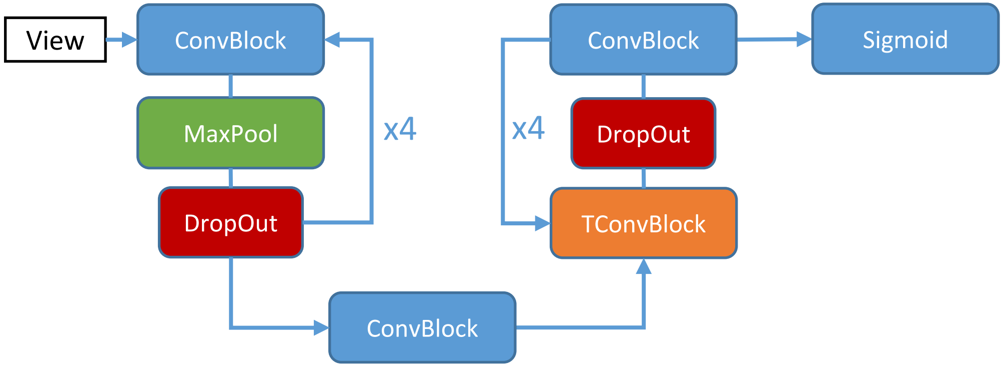
- Each hit in an image can be related to the interaction vertex in terms of its distance to it
  - And direction, of course, but we'll ignore that for now
- Encode this distance information in the form of 19 classes, each spanning a particular range of distances
- This turns the vertex finding procedure into a semantic segmentation problem, suitable for a U-Net
  - Every hit contributes directly to the loss function
  - Hits are spatially correlated
- At inference time the network attempts to determine the appropriate distance class for every hit
- But note this doesn't give us a vertex location…

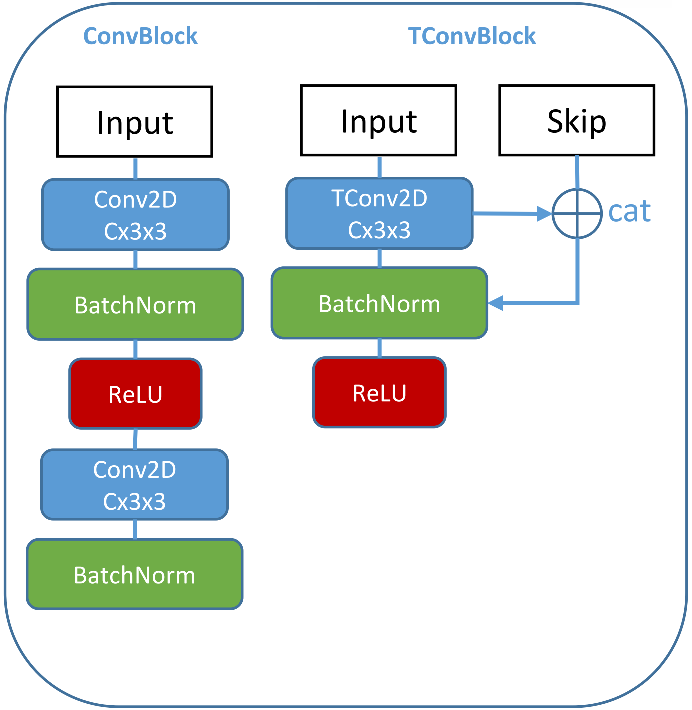


$\bar{\nu}_\mu + Ar \rightarrow \mu^+ + 3\pi^0$

Network architecture

- U-Net concept introduced in 2015 for biomedical image segmentation (arXiv:1505.04597)
- The name comes from the conceptual structure of the network
- Attempt to classify every pixel in an image





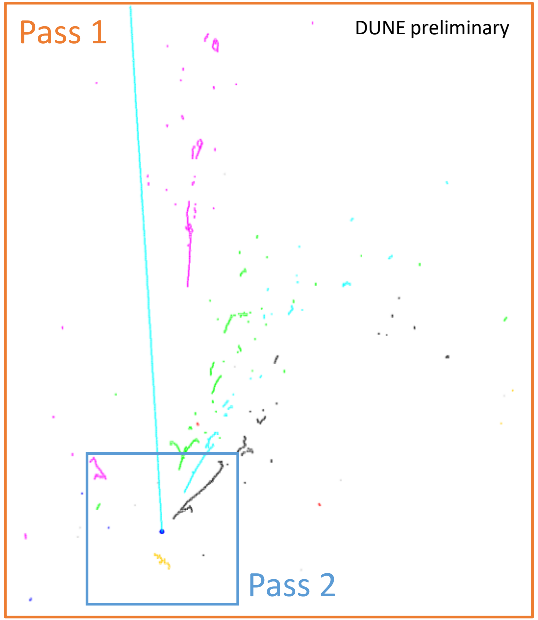
Performance considerations

WARWICK

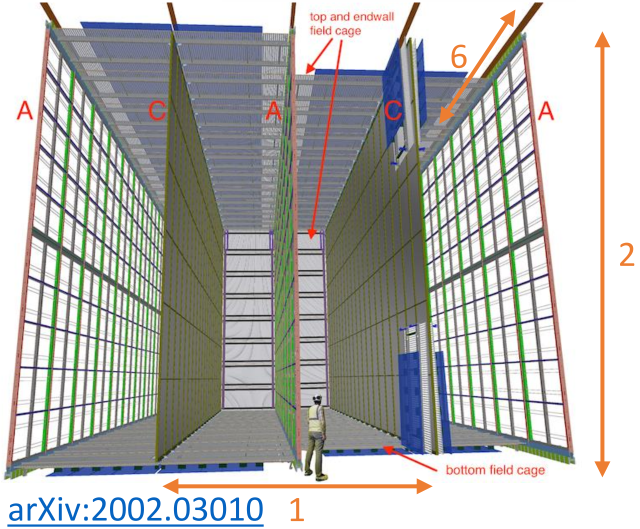
- Semantic segmentation is memory intensive
- Instead of a single classification output for an image, there is a classification output for each pixel
- Strong constraint on input image size and batch size
- 256 x 256 is about the limit with a training batch size of 32
- Single pixel errors represent several centimetres for events spanning workspace geometry

Network training

- ~93,000 event training sample using horizontal drift far detector 1x2x6 APA geometry
  - 75:25 split between training and validation



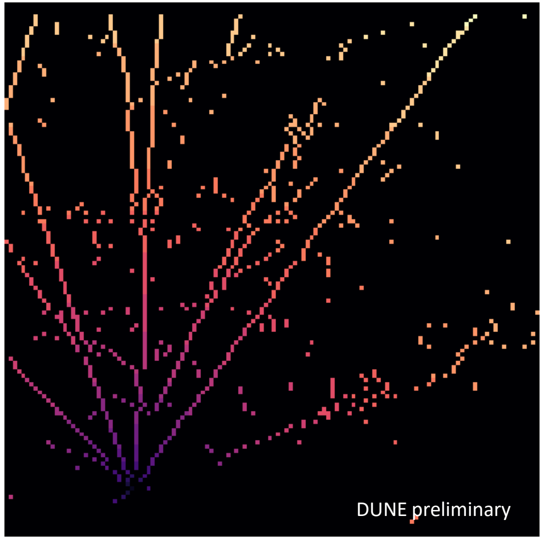
- Two pass approach
  - Pixels often exceed ~0.5 cm wire pitch
  - First pass region finding with 256x256 pixel input containing whole event
  - Second pass comprises 128x128 pixel input containing 64 cm x 64 cm region



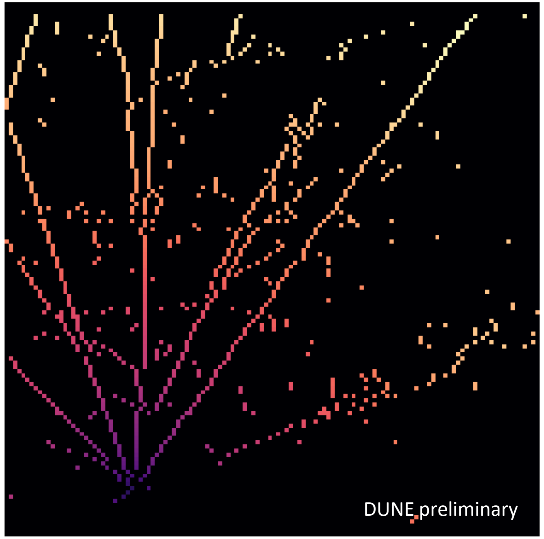
arXiv:2002.03010

Example event

- As noted previously, the network does not return the vertex location
- Instead, the spatial relationship between hits and the vertex is learned
- An example pass 2 image is presented here
- The network is clearly able to learn spatial relationships between the hits and the vertex



Truth



Classification

Vertex extraction

- We have a set of distance classes for each occupied pixel
- Draw a ring, centred on the hit with radii corresponding to the distance bounds
- Weight the pixels in the ring inversely proportional to its area
- Vertex could be anywhere within the shaded region of one ring

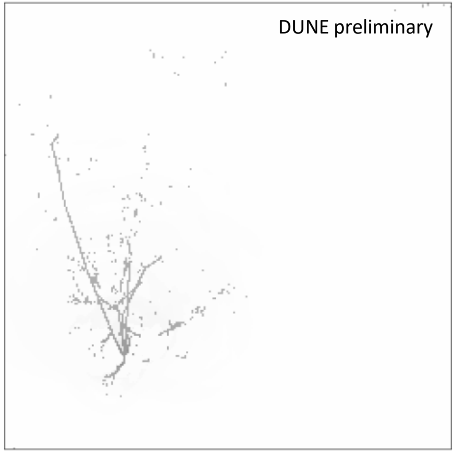
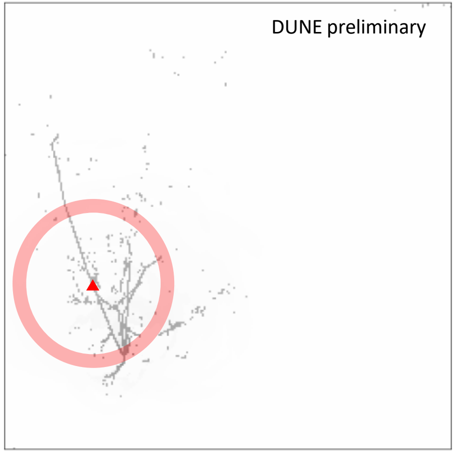


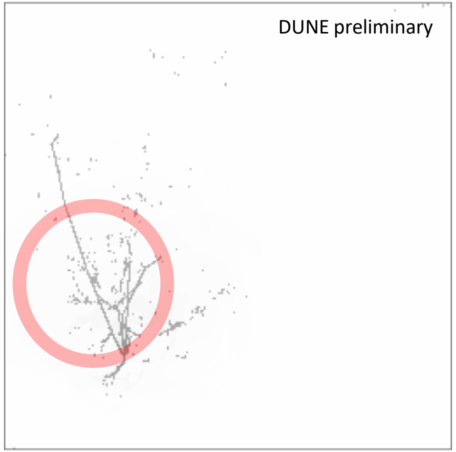
Image from a single wire plane



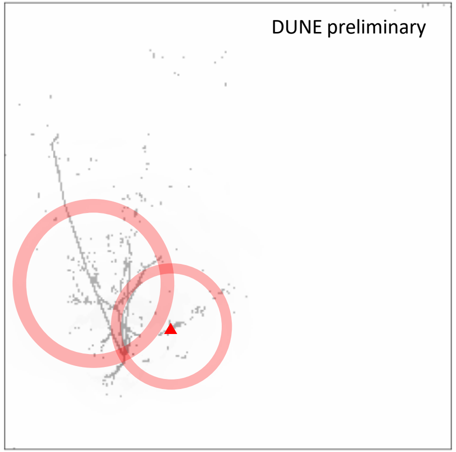
Heat map from one classified pixel

Vertex extraction

- We have a set of distance classes for each occupied pixel
- Draw a ring, centred on the hit with radii corresponding to the distance bounds
- Weight the pixels in the ring inversely proportional to its area
- Many rings for a heat map, where high weight indicates likely location



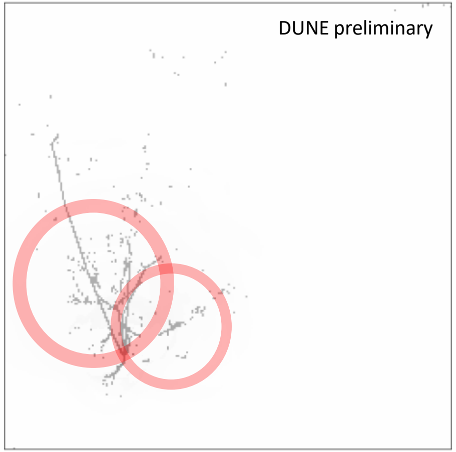
Heat map from one classified pixel



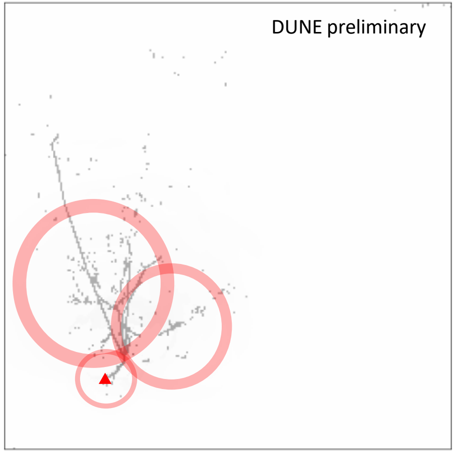
Heat map from two classified pixels

Vertex extraction

- We have a set of distance classes for each occupied pixel
- Draw a ring, centred on the hit with radii corresponding to the distance bounds
- Weight the pixels in the ring inversely proportional to its area
- Many rings for a heat map, where high weight indicates likely location



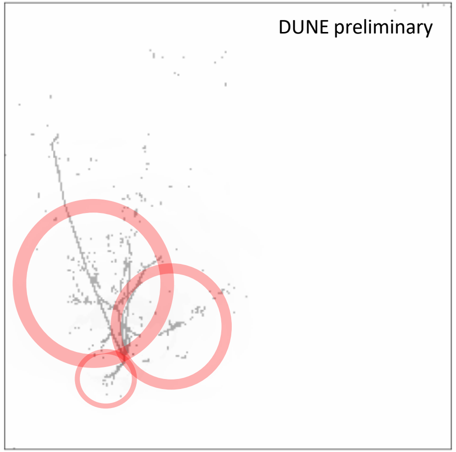
Heat map from two classified pixels



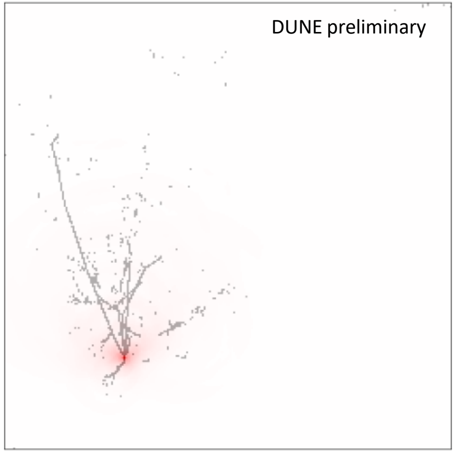
Heat map from three classified pixels

Vertex extraction

- We have a set of distance classes for each occupied pixel
- Draw a ring, centred on the hit with radii corresponding to the distance bounds
- Weight the pixels in the ring inversely proportional to its area
- Many rings for a heat map, where high weight indicates likely location

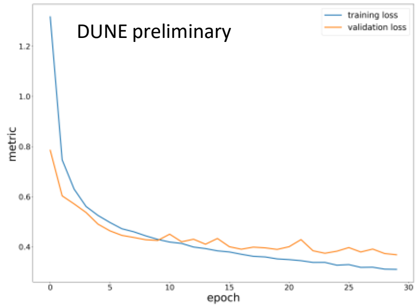


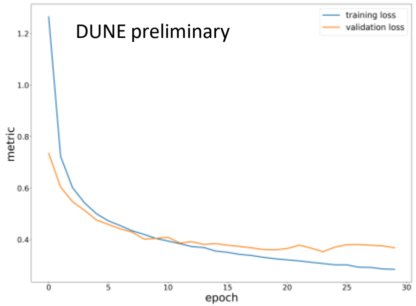
Heat map from three classified pixels

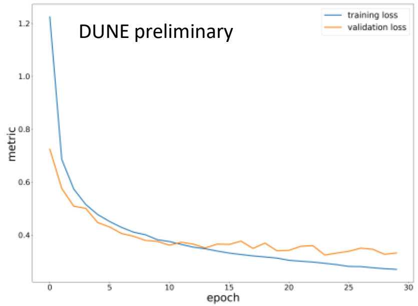


Heat map from all classified pixels

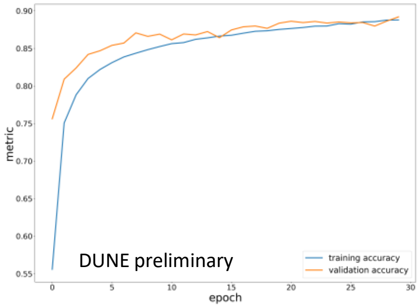
Training the first pass

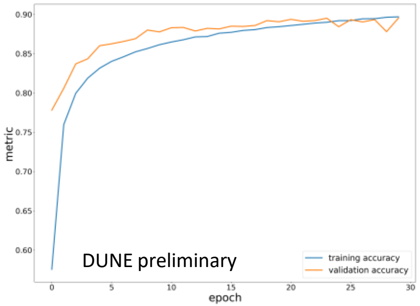


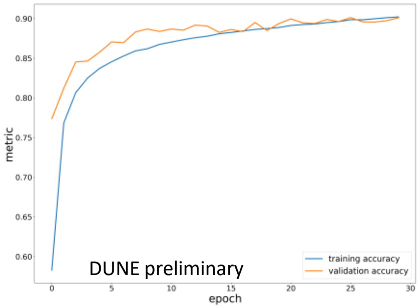




- First pass network appears to train well
  - Smooth loss metric and accuracy evolution, no evidence of over-fitting
  - ~90% classification accuracy across all hits



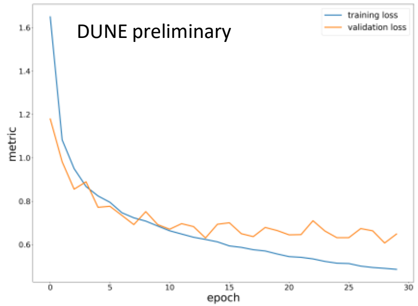


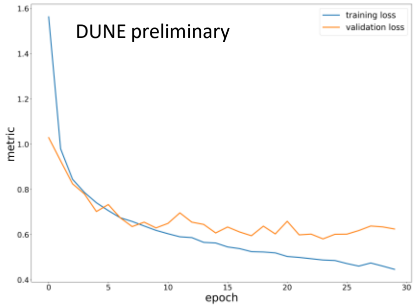


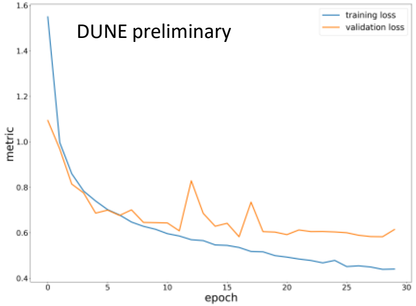
Training the second pass

- 64 cm x 64 cm event region, allowing for 0.5 cm resolution
- Smaller images at 128 x 128 pixels
- If the first pass vertex is off by much more than 32 cm, pass two won't help
- 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 (check hit containment and try again if no hits)
  - First pass will be imperfect - we don't want the network to learn to pick the centre all the time
- I don't like this approach, alternatives:
  - Smaller, uniform perturbation of truth, using hit distribution to frame event similarly to pass one
  - Use unperturbed pass one vertex result, using hit distribution to frame event similarly to pass one

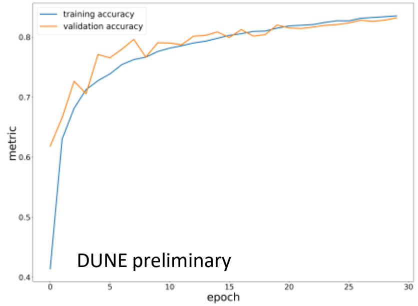
Training the second pass

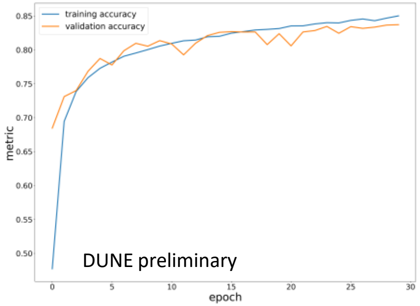


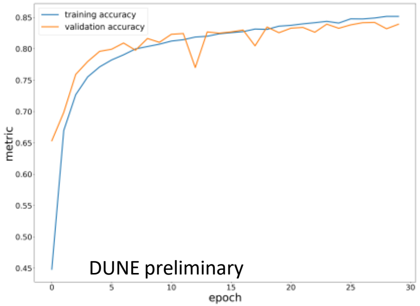




- Second pass network appears to train well
  - Smooth loss metric and accuracy evolution (a couple of sharp changes in W), no evidence of over-fitting
  - ~80% classification accuracy across all hits

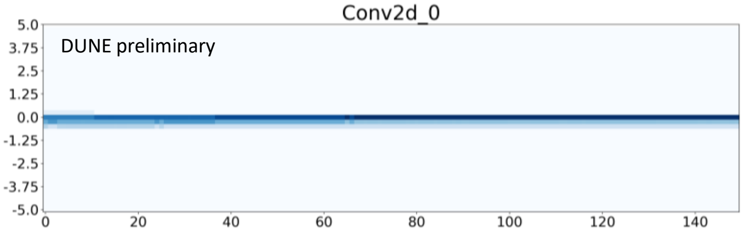


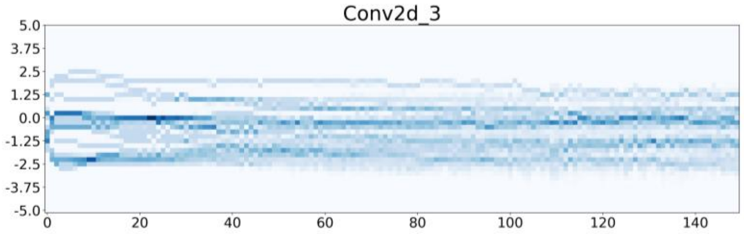


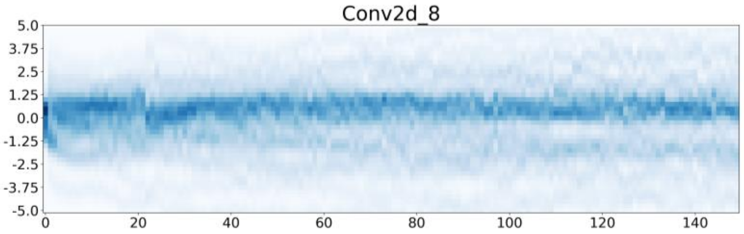


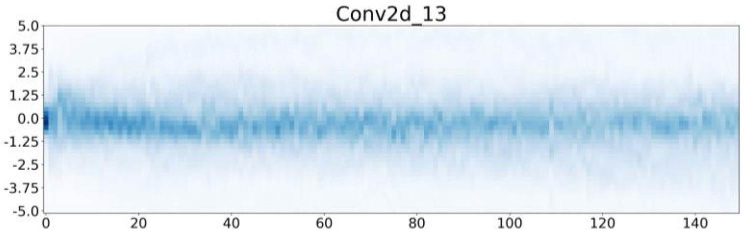
Investigating network activations

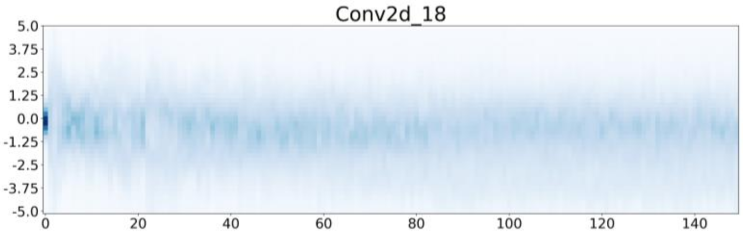
WARWICK











- Distribution of activations per down-sampling layer mostly reasonable
  - Reasonably broad spread of activations
  - No sharp discontinuities as training proceeds
- First layer has too many activations near zero

Exploring the loss landscape

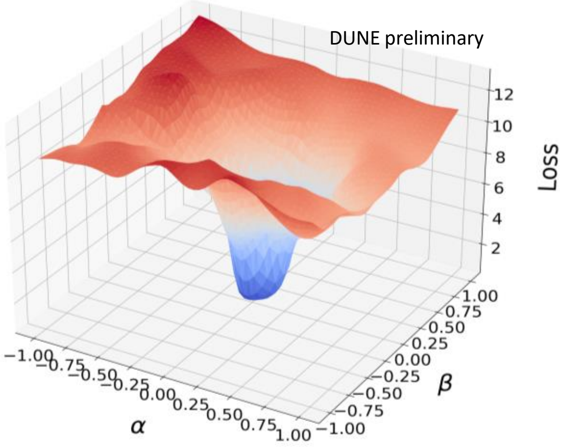
- Loss landscape looks to visualize the mean loss over many inputs with respect to different parameter values
  - Map the ~2.2M-dimensional space onto two parameters to visualize
  - Use the method developed by Li et al (arXiv:1712.09913)
  - Generate two random Gaussian direction vectors (N = 2.2M), $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$
  - Pick parameters α and β on a grid [-1, 1] and take a step $\alpha\boldsymbol{\delta} + \beta\boldsymbol{\eta}$ away from the training minimum
  - Compute the mean loss over 1024 events in validation set for the model with these parameters
  - Aim for smooth loss landscape for efficient, effective learning
- Also consider distribution of trained weights
  - Aim for many small weights
  - Better generalisation than relying on fewer substantial weights

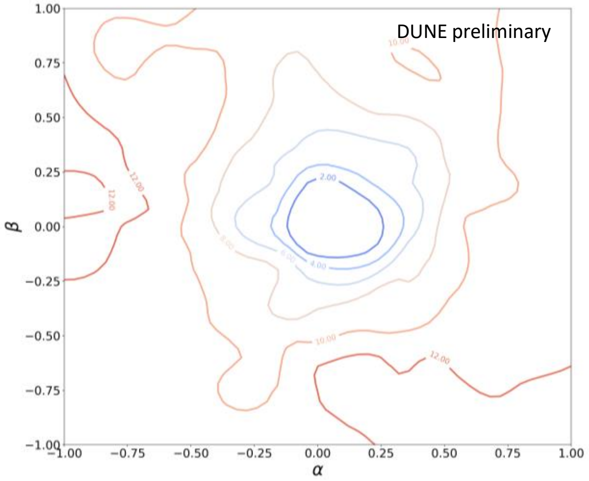
Exploring the loss landscape

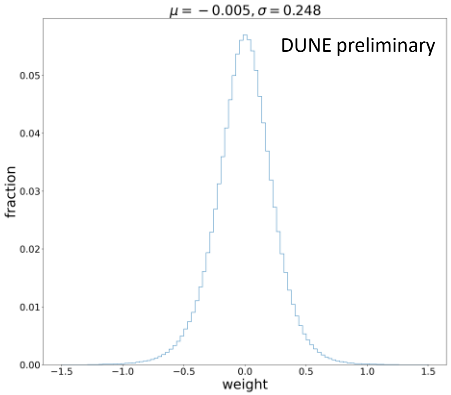
- The loss landscape can help identify problems with architecture or hyperparameters
- Multiple local minima indicate either poorly chosen hyperparameters (e.g. learning rate) or an architecture not conducive to smooth weight evolution
- Here the loss landscape is quite smooth
- Few large weights

Loss landscape at the end of training

Visualisation method from [arXiv:1712.09913]

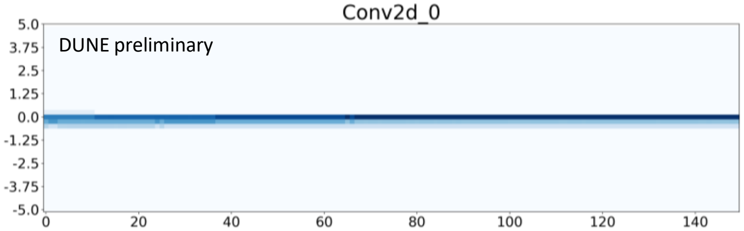






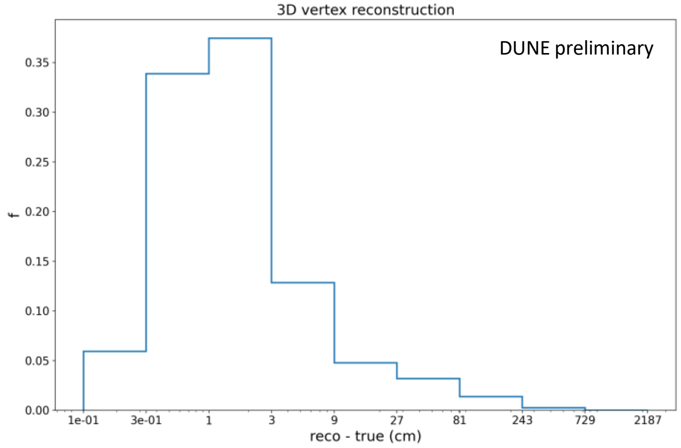
Improving network behaviour

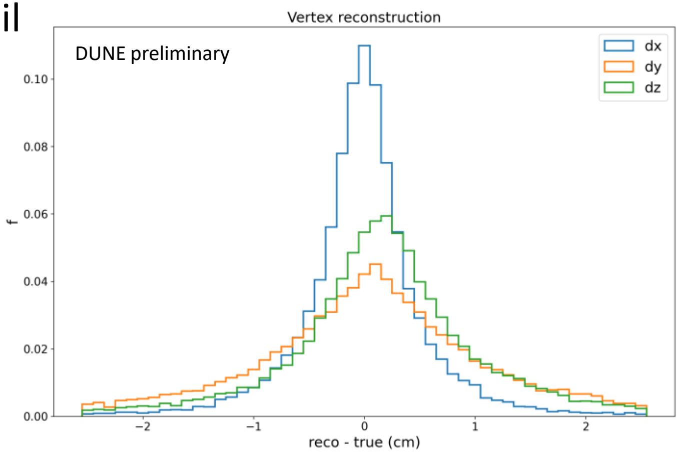
- Most layers appear to have reasonable distributions of activations
- But the network doesn't get off to a good start
- Most activations are near zero
- Potential solutions:
  - Weight initialisation – activations can collapse if initial distribution of weights leads to vanishing gradients
  - Current architecture only uses ReLU activations within convolution blocks
    - Add them after blocks as well
  - Current architecture is a U-Net, not a U-ResNet
    - In principle, easier to learn residuals between input and output rather than full mapping from input to output in convolutional layers

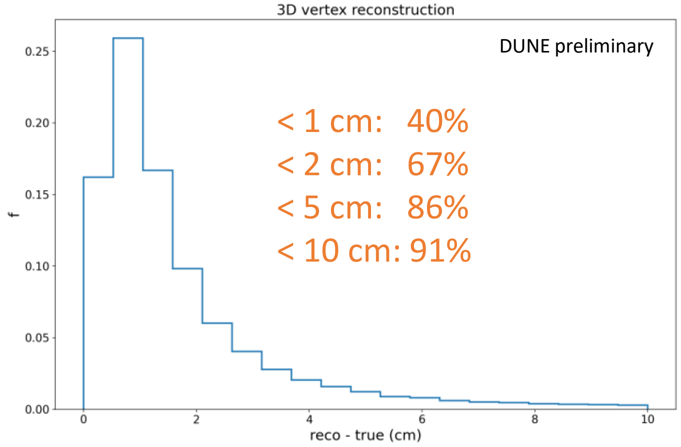


First pass performance

- Tested on an independent test sample
  - 28,611 events passing fiducial volume cut
  - Approximately even split of ν_μ and ν_e events
- Typically isolates region well
- Narrow dx – all three views have direct measure
- Modest downstream bias
- Very long tail

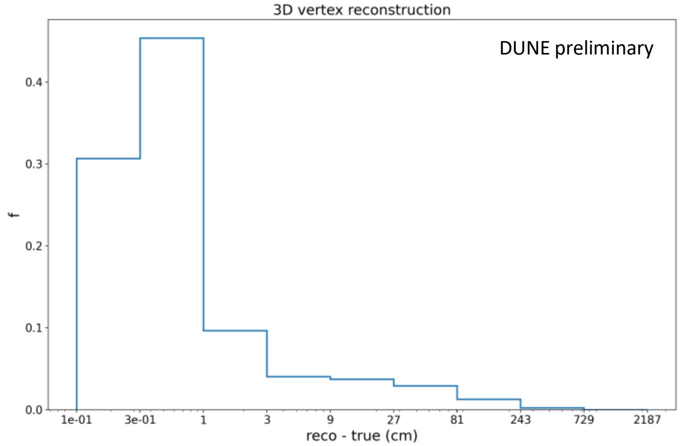


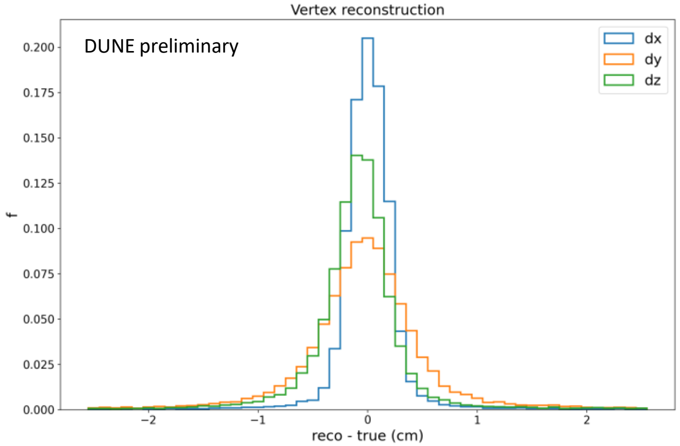


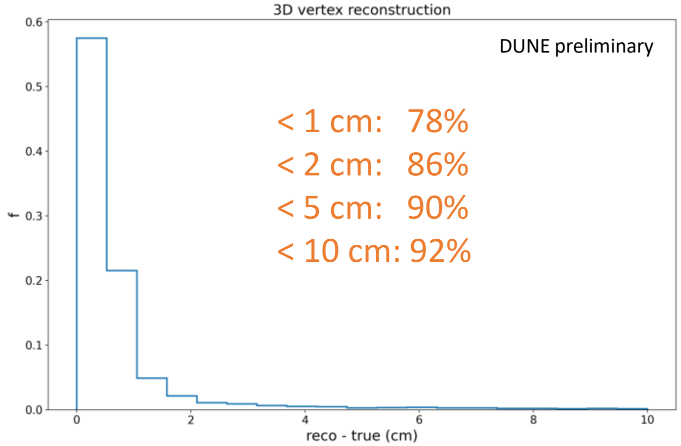


Second pass performance

- Tested on an independent test sample
  - 28,611 events passing fiducial volume cut
  - Approximately even split of ν_μ and ν_e events
- Typically sub 1cm resolution
- Little bias
- Long tail remains as expected



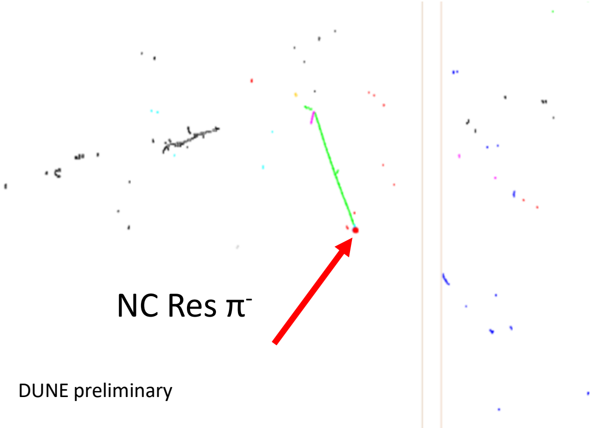




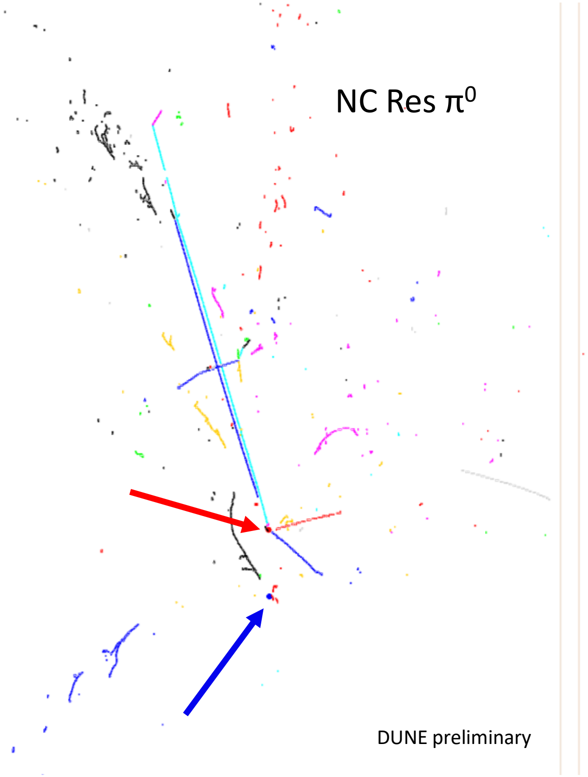
Example events

WARWICK

- Correct NC Res π^- vertex
  - Likely guided by learning beam direction
  - What if this had been an atmospheric neutrino?
- Incorrect NC Res π^0 vertex
  - Understandable choice given particle multiplicity
  - Perhaps a human weights the π^0 decay photons



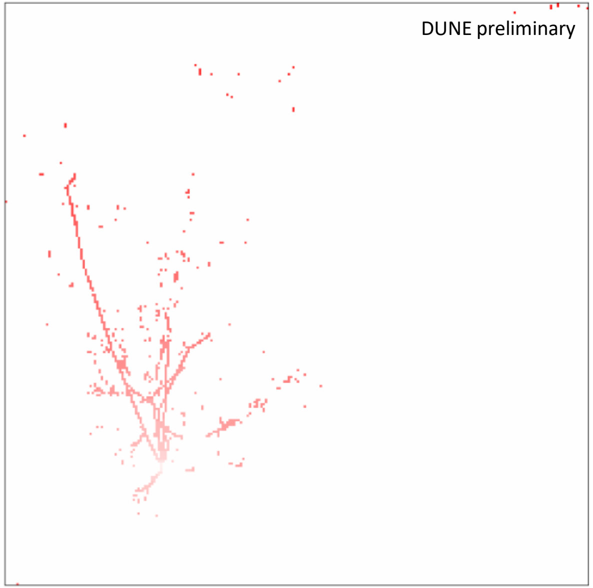
True
Reco



Ongoing work and future plans

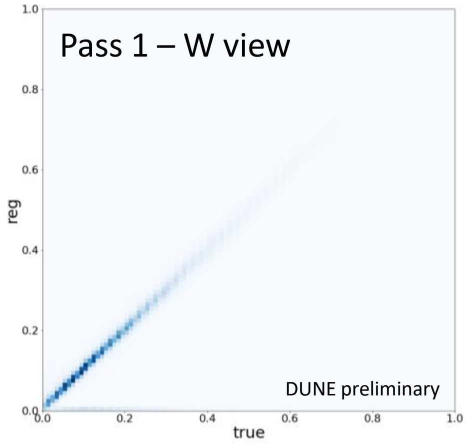
Per pixel regression

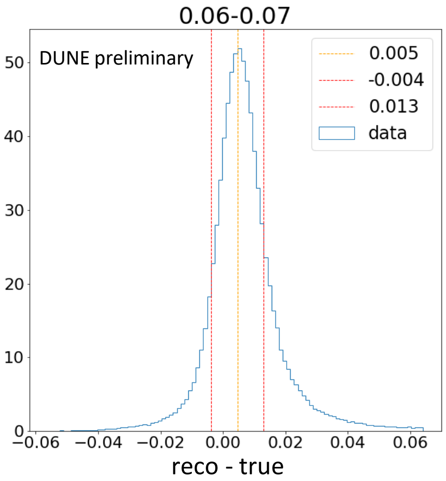
- Use of classes to represent hit distances due to feasibility concerns for learning continuous distances
- Current distance thresholds are somewhat arbitrary
- Worth checking if the network can learn continuous distances rather than discrete classes
- This change does require some care in handling loss functions

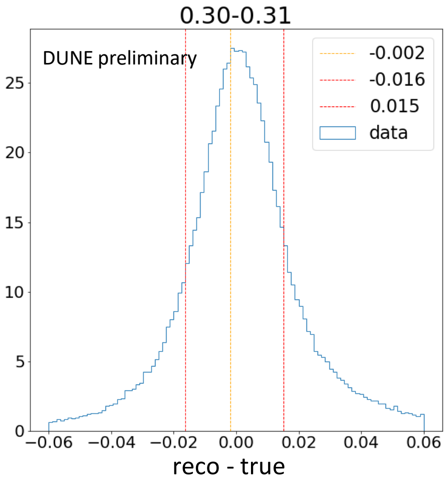


Network inference behaviour

- Key aspect of the per-pixel regression is the distribution of deltas between inferred distances and true distances
- Network frequently finds a close correspondence between inference and truth
- However, when looking at given true distance bins (e.g. 0.06-0.07) there is generally a bias in the inferred distance, the uncertainty can become quite large and some distributions are rather skewed. More work needed.







Alternative measures of distance

- Existing implementation is radial distance
- Could opt for perpendicular distance in drift coordinate and wire coordinates
- Ring drawing algorithm becomes a band drawing algorithm
  - Potential for simultaneous determination of the drift coordinate in all three views
  - Simple, fast implementation including some directional information

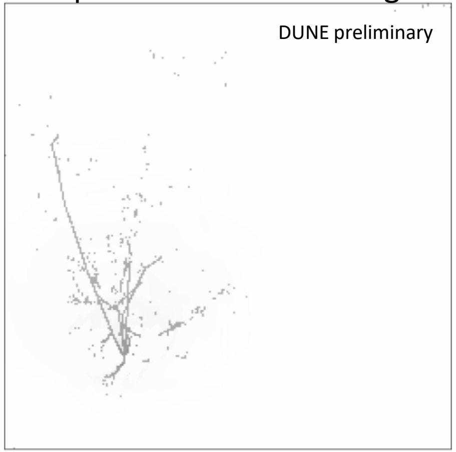
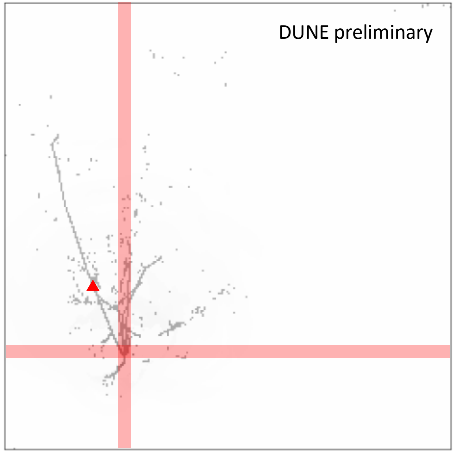


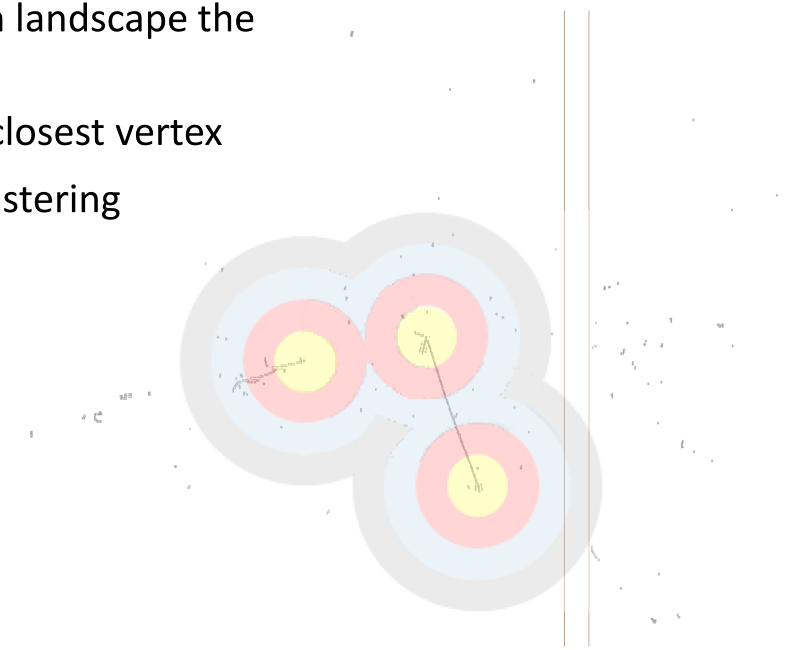
Image from a single wire plane



Heat map from one classified pixel

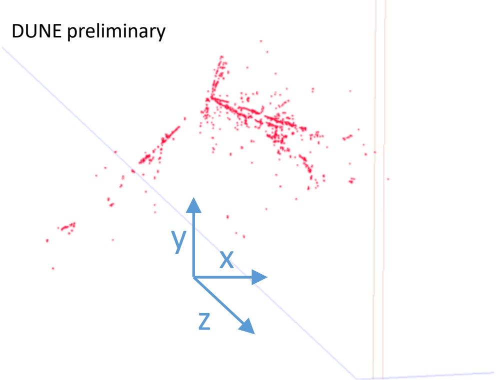
Secondary vertices

- If the network can learn from a more complex truth landscape the idea can be extended to secondary vertices
- Individual hits could now indicate proximity to the closest vertex
- Produce a graph of links between vertices to aid clustering decisions during downstream reconstruction



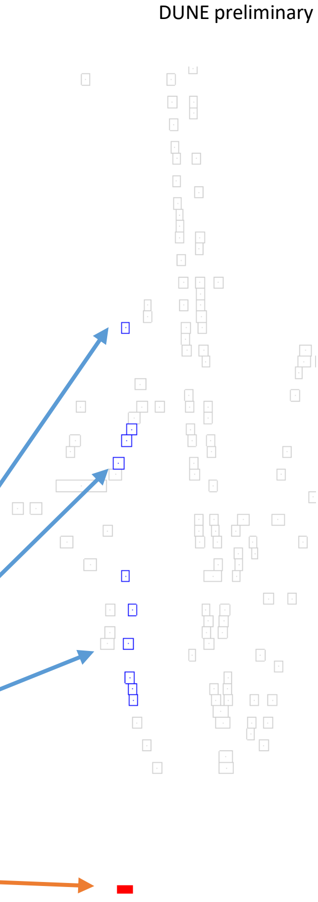
3D hits

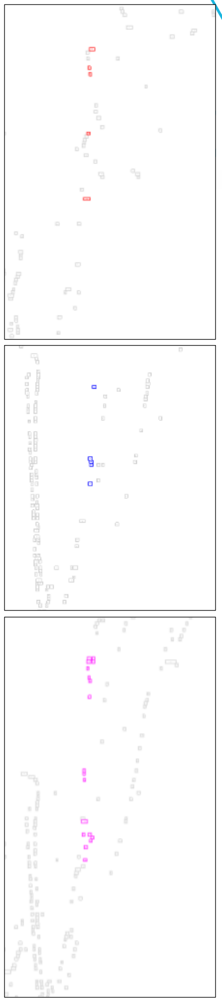
- Can we infer 3D hit locations early in reconstruction?
  - Vertex finding approach could be adapted from pixels to voxels
  - Also opens possibility of graph neural network-based solutions
- Use the common drift-coordinate to determine which hits in each view could plausibly come from the same 3D hit
  - Combinatorics are challenging, but tractable



Candidate matches in V

Hit of interest in U





Expanded set of potential relationships in U, V and W

Conclusion

- Automated vertex finding in multi-kiloton LArTPCs is challenging
- Deep learning can yield accurate and precise reconstructed vertices across many events
- A significant long tail nonetheless remains
- Various avenues to improving performance are being explored

Backup

Ring-drawing efficiency

- Events can contain a relatively large number of hits (constrained by image size)
- Rings have to be drawn for each hit (or at least a sufficiently large subset)
- This process needs to be efficient or post-processing will require more computation than the network inference
- Bresenham mid-point circle algorithm provides efficient circle drawing needing only integer operations to determine which pixel to fill next as you move around an octant and then mirror to the remaining octants
  - Performance benefits vary by CPU architecture, but ~1.5-2x speed up for integer versus float in modern architectures for add, sub, mul
  - Note, double precision typically as fast as single precision for add, sub, mul and div in modern architectures
- Caveat: we need rings, not single pixel wide circles, and need to ensure that each pixel within the ring is filled once and only once, but Bresenham can be extended to achieve this while retaining integer-only arithmetic

Sparse implementation

- Our images are predominantly empty space
- Dense convolutions spend a lot of time multiplying by zero
- Non-hit regions contribute to loss function
- Sparse implementations exist to allow convolutions to be performed on sparse tensors (e..g MinkowskiEngine)
- Principal issue at the moment is the need to run in C++ on a CPU in the DUNE reconstruction workflow
    - Requires conversion of networks to TorchScript
    - Networks with non-standard tensor inputs have problems with this conversion
    - Future workflow likely to offer GPU as a service