

# Pandora event display

## Part 2: Visualising the algorithms

Dom Brailsford for the Pandora team

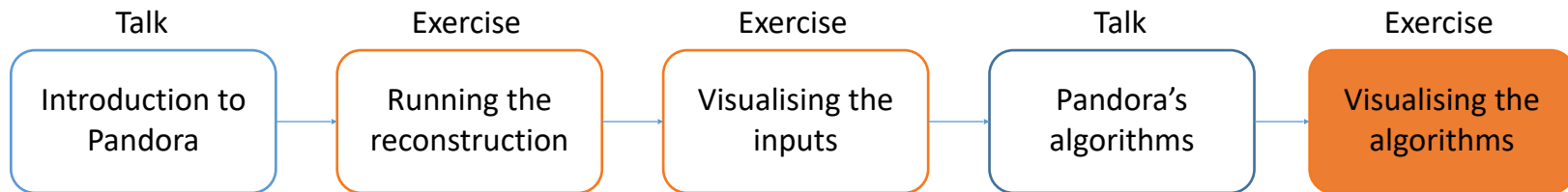
---

06/06/2024

DUNE-France Workshop



# Reconstruction session



Credit: These slides are based on previous LArSoft workshop slides by Lorena Escudero and Andrew Smith

Key references:

[Pandora ProtoDUNE paper](#)  
[Pandora MicroBooNE paper](#)

# Goals

- Main goal - Visualise the status of the pattern-recognition after each main stage
  - Add the visual monitoring algorithm to the Pandora configuration XML file after running the:
    - 2D reconstruction
    - 3D vertex reconstruction
    - Track & Shower reconstruction & particle refinement
    - 3D hit reconstruction
    - Neutrino hierarchy reconstruction
- Please don't worry if you don't get through all of the steps
  - This session is just for you to get some intuition for what Pandora's algorithms do

## Main Goal

Visualise the algorithms

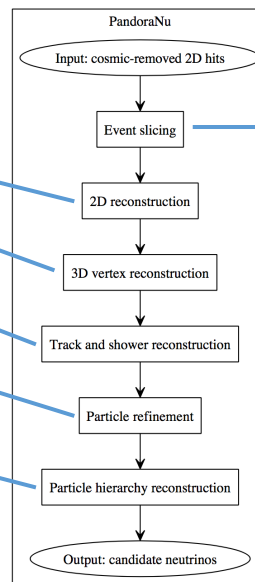
# The neutrino algorithm chain

- Go to our config directory and make a copy of the Pandora `neutrino` XML settings file

```
$ cd /exp/dune/data/users/$USER/reco/config
$ cp $LARPANDORA_DIR/scripts/PandoraSettings_Neutrino_Standard.xml
MyPandoraSettings_Neutrino_Standard.xml
$ vim MyPandoraSettings_Neutrino_Standard.xml
```

- Look through the file for the sections listed below:

```
<!-- TwoDReconstruction -->
<!-- VertexAlgorithms -->
<!-- ThreeDTrackAlgorithms -->
<!-- ThreeDShowerAlgorithms -->
<!-- Repeat ThreeDTrackAlgorithms -->
<!-- ThreeDRecoveryAlgorithms -->
<!-- TwoDMopUpAlgorithms -->
<!-- ThreeDHitAlgorithms -->
<!-- ThreeDMopUpAlgorithms -->
<!-- NeutrinoAlgorithms -->
<!-- Track and shower building -->
```



We're not running the event slicing because we don't have cosmics to deal with

## Point to our neutrino settings file

- Modify `MyPandoraSettings_Master_Standard.xml` and point it to our new neutrino settings file
- Remove the visual monitoring algorithms that we've been using so far

+ MyPandoraSettings\_Master\_Standard.xml

```
<pandora>
  <!-- GLOBAL SETTINGS -->
  <IsMonitoringEnabled>true</IsMonitoringEnabled>
  <ShouldDisplayAlgorithmInfo>true</ShouldDisplayAlgorithmInfo>
  <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

  <!-- ALGORITHM SETTINGS -->
  <algorithm type = "LArPreProcessing">
    <OutputCaloHitListNameU>CaloHitListU</OutputCaloHitListNameU>
    <OutputCaloHitListNameV>CaloHitListV</OutputCaloHitListNameV>
    <OutputCaloHitListNameW>CaloHitListW</OutputCaloHitListNameW>
    <FilteredCaloHitListName>CaloHitList2D</FilteredCaloHitListName>
    <CurrentCaloHitListReplacement>CaloHitList2D</CurrentCaloHitListReplacement>
  </algorithm>

  <algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</CaloHitListNames>
    <ShowDetector>true</ShowDetector>
  </algorithm>

  <algorithm type = "LArMaster">
    <CRSettingsFile>PandoraSettings_Cosmic_Standard.xml</CRSettingsFile>
    <NuSettingsFile>MyPandoraSettings_Neutrino_Standard.xml</NuSettingsFile>
    <SlicingSettingsFile>PandoraSettings_Slicing_Standard.xml</SlicingSettingsFile>

  ... more settings ...

  <algorithm type = "LArVisualMonitoring">
    <ShowCurrentPfos>true</ShowCurrentPfos>
    <ShowDetector>true</ShowDetector>
  </algorithm>
</pandora>
```

Remove this algorithm block

Change this line to point to  
`MyPandoraSettings_Neutrino_Standard.xml`

Remove this algorithm block

## 2D reconstruction

# Add in some visualizations

- Add to `MyPandoraSettings_Neutrino_Standard.xml` at the end of the `TwoDReconstruction` section

```

../c/MyPandoraSettings_Neutrino_Standard.xml
<pandora>
  <!-- Output list management -->
  <!-- GLOBAL SETTINGS -->
  <IsMonitoringEnabled>true</IsMonitoringEnabled>
  <ShouldDisplayAlgorithmInfo>true</ShouldDisplayAlgorithmInfo>
  <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

  ... more settings ...

  <algorithm type = "LARKinkSplitting"/>
  <algorithm type = "LARTrackConsolidation">
    <algorithm type = "LARSimpleClusterCreation" description = "ClusterRebuilding"/>
  </algorithm>

  <algorithm type = "LARVisualMonitoring">
    <CaloHitListNames>CaloHitListU</CaloHitListNames>
    <ClusterListNames>ClustersU</ClusterListNames>
    <ShowDetector>true</ShowDetector>
  </algorithm>

  <algorithm type = "LARVisualMonitoring">
    <CaloHitListNames>CaloHitListV</CaloHitListNames>
    <ClusterListNames>ClustersV</ClusterListNames>
    <ShowDetector>true</ShowDetector>
  </algorithm>

  <algorithm type = "LARVisualMonitoring">
    <CaloHitListNames>CaloHitListW</CaloHitListNames>
    <ClusterListNames>ClustersW</ClusterListNames>
    <ShowDetector>true</ShowDetector>
  </algorithm>

  <!-- VertexAlgorithms -->
  <algorithm type = "LARCandidateVertexCreation">
    <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
    <OutputVertexListName>CandidateVertices3D</OutputVertexListName>
    <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
  </algorithm>

```

Set this to **true** - this will print to the terminal all of the algorithms we are running

Modify the **Neutrino** file not the **Master** settings file

Add these visual monitoring blocks. When we run, this will make 3 event displays - each showing the **hits** and **clusters** in the U, V and W views respectively

Add the above lines just before the **VertexAlgorithms** section



# Visualizing the initial 2D reconstruction

```
$ cd /exp/dune/data/users/$USER/reco/work
$ lar -c event_display_driver.fcl -s reco2_1mulp.root -n 1
```

Let's just look at 1 event for now!

Can also run on pre-made reco2 file in

```
$ /exp/dune/data/users/dbrailsf/workshops/annecy2024/complete/reco2_1mulp.root
```

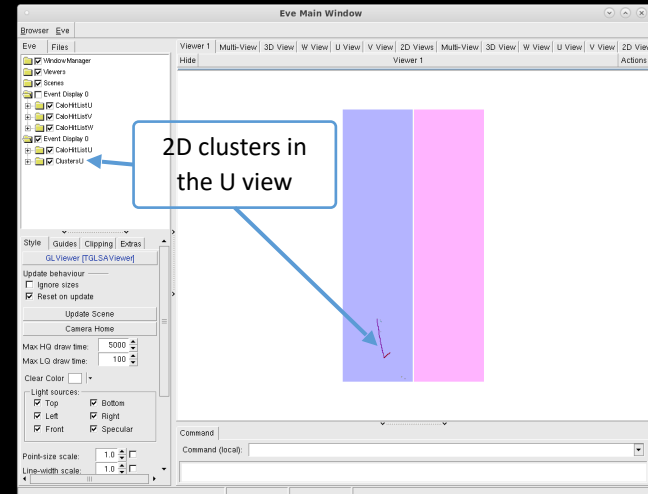
2D clustering  
algorithms in  
the U-view

2D clustering  
algorithms in  
the V-view

2D clustering  
algorithms in  
the W-view

First visualization

```
> Running Algorithm: Alg0001, LARPreProcessing
> Running Algorithm: Alg0002, LARClusteringParent
---> Running Algorithm: Alg0003, LARTrackClusterCreation
> Running Algorithm: Alg0004, LARLayerSplitting
> Running Algorithm: Alg0005, LARLongitudinalAssociation
> Running Algorithm: Alg0006, LARTransverseAssociation
> Running Algorithm: Alg0007, LARLongitudinalExtension
> Running Algorithm: Alg0008, LARTransverseExtension
> Running Algorithm: Alg0009, LARCrossGapsAssociation
> Running Algorithm: Alg0010, LARCrossGapsExtension
> Running Algorithm: Alg0011, LAROvershootSplitting
> Running Algorithm: Alg0012, LARBranchSplitting
> Running Algorithm: Alg0013, LARKinkSplitting
> Running Algorithm: Alg0014, LARTrackConsolidation
> Running Algorithm: Alg0016, LARClusteringParent
---> Running Algorithm: Alg0017, LARTrackClusterCreation
> Running Algorithm: Alg0018, LARLayerSplitting
> Running Algorithm: Alg0019, LARLongitudinalAssociation
> Running Algorithm: Alg0020, LARTransverseAssociation
> Running Algorithm: Alg0021, LARLongitudinalExtension
> Running Algorithm: Alg0022, LARTransverseExtension
> Running Algorithm: Alg0023, LARCrossGapsAssociation
> Running Algorithm: Alg0024, LARCrossGapsExtension
> Running Algorithm: Alg0025, LAROvershootSplitting
> Running Algorithm: Alg0026, LARBranchSplitting
> Running Algorithm: Alg0027, LARKinkSplitting
> Running Algorithm: Alg0028, LARTrackConsolidation
> Running Algorithm: Alg0030, LARClusteringParent
---> Running Algorithm: Alg0031, LARTrackClusterCreation
> Running Algorithm: Alg0032, LARLayerSplitting
> Running Algorithm: Alg0033, LARLongitudinalAssociation
> Running Algorithm: Alg0034, LARTransverseAssociation
> Running Algorithm: Alg0035, LARLongitudinalExtension
> Running Algorithm: Alg0036, LARTransverseExtension
> Running Algorithm: Alg0037, LARCrossGapsAssociation
> Running Algorithm: Alg0038, LARCrossGapsExtension
> Running Algorithm: Alg0039, LAROvershootSplitting
> Running Algorithm: Alg0040, LARBranchSplitting
> Running Algorithm: Alg0041, LARKinkSplitting
> Running Algorithm: Alg0042, LARTrackConsolidation
> Running Algorithm: Alg0044, LARVisualMonitoring
PandoraMonitoring:InitializeEve(): DISPLAY environment set to :1001.0
Press return to continue ...
```



# Initial 2D reconstruction – U View

Turn off the hits, we've included them so you can always refer back to the inputs if you like

Expand the list of clusters

Try turning on and off some of the clusters so you can see what they correspond to in the viewer

Clusters are ordered by the total energy deposited

Viewer 1

Multi-View | 3D View | W View | U View | V View | 2D Views | Multi-View | 3D View | W View | U View | V View | 2D Views

Viewer 1

Hide

Use Viewer 1 so we can check and uncheck boxes

After the initial 2D reconstruction you will probably find 2 main clusters (for the proton and muon) but possibly many smaller clusters at kinks and bifurcations

Each colour corresponds to a different cluster

Command

Command (local):



Wheel up - zoom out  
Wheel down - zoom in

Wheel press + drag - pan viewport



W - wireframe mode

R - return from wireframe mode

# Looking at the other views

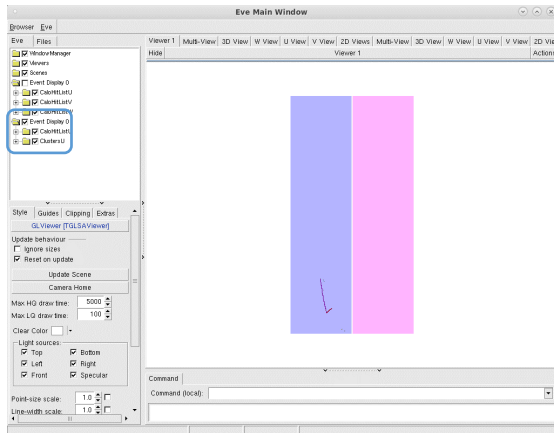
Click in the terminal window  
and press Return ↵ to  
visualize the other views

```

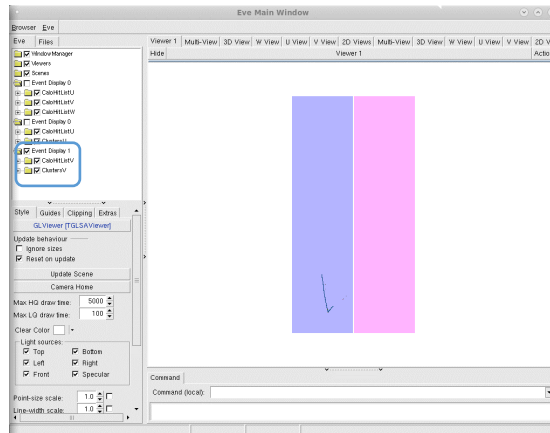
<algorithm type = "LArVisualMonitoring">
  <CaloHitListNames>CaloHitListU</CaloHitListNames>
  <ClusterListNames>ClustersU</ClusterListNames>
  <ShowDetector>>true</ShowDetector>
</algorithm> Return

<algorithm type = "LArVisualMonitoring">
  <CaloHitListNames>CaloHitListV</CaloHitListNames>
  <ClusterListNames>ClustersV</ClusterListNames>
  <ShowDetector>>true</ShowDetector>
</algorithm> Return

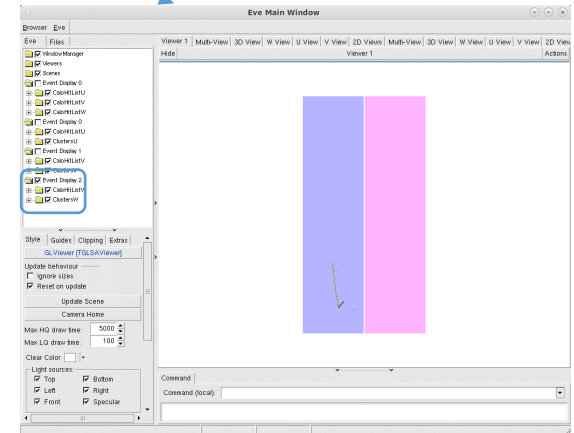
<algorithm type = "LArVisualMonitoring">
  <CaloHitListNames>CaloHitListW</CaloHitListNames>
  <ClusterListNames>ClustersW</ClusterListNames>
  <ShowDetector>>true</ShowDetector>
</algorithm> Return
  
```



U view clusters



V view clusters



W view clusters

## 3D vertex reconstruction

# Add in some more visualisations

This tutorial's reconstruction uses the older vertex reconstruction. DUNE-FD now uses deep-learning assisted vertex reconstruction



- Add to `MyPandoraSettings_Neutrino_Standard.xml` at the end of the `VertexAlgorithms` section

```

<!-- VertexAlgorithms -->
<algorithm type = "LArCandidateVertexCreation">
  <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
  <OutputVertexListName>CandidateVertices3D</OutputVertexListName>
  <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
  <EnableCrossingCandidates>>false</EnableCrossingCandidates>
</algorithm>
<algorithm type = "LArEnergyKickVertexSelection">
  <InputCaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</InputCaloHitListNames>
  <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
  <OutputVertexListName>NeutrinoVertices3D</OutputVertexListName>
  <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
  <FeatureTools>
    <tool type = "LArEnergyKickFeature"/>
    <tool type = "LArLocalAsymmetryFeature"/>
  </FeatureTools>
</algorithm>
<algorithm type = "LArVertexSplitting">
  <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
</algorithm>

<algorithm type = "LArVisualMonitoring">
  <ClusterListNames>ClustersW</ClusterListNames>
  <VertexListNames>CandidateVertices3D</VertexListNames>
  <ShowDetector>true</ShowDetector>
</algorithm>

<algorithm type = "LArVisualMonitoring">
  <ClusterListNames>ClustersW</ClusterListNames>
  <VertexListNames>NeutrinoVertices3D</VertexListNames>
  <ShowDetector>true</ShowDetector>
</algorithm>

<!-- ThreeDTrackAlgorithms -->
<algorithm type = "LARThreeDTransverseTracks">
  <InputClusterListNameU>ClustersU</InputClusterListNameU>
  <InputClusterListNameV>ClustersV</InputClusterListNameV>
  <InputClusterListNameW>ClustersW</InputClusterListNameW>

```

The `LArCandidateVertexCreation` algorithm creates a list of 3D candidate vertices at positions that project onto the ends of the existing 2D clusters

The `LArEnergyKickVertexSelection` algorithm selects the neutrino vertex from the candidates

Visualise the 3D candidate vertices along with the W-view clusters for comparison

Visualise the selected neutrino vertex along with the W-view clusters for comparison

Add the above lines just before the `ThreeDTrackAlgorithms` section



# Candidate 3D vertices vs W-view clusters

Expand the list of candidate vertices - there will be many!

Each vertex here is displayed as a yellow circle in the viewer

**Eve Main Window**

Viewer 1 | Multi-View | 3D View | W View | U View | V View | 2D Views | Multi-View | 3D View | W View | U View | V View | 2D View

Hide | Viewer 1 | Actions

Event Display 3  
ClustersW  
Candidate/Vertices3D

- Vertex[] x: -185.167 y: 23.2
- Vertex[] x: -149.962 y: -0.1
- Vertex[] x: -185.081 y: 22.6
- Vertex[] x: -150.127 y: -0.1
- Vertex[] x: -149.962 y: 0.06
- Vertex[] x: -149.456 y: 2.81
- Vertex[] x: -149.962 y: 1.67
- Vertex[] x: -152.995 y: 5.57
- Vertex[] x: -152.363 y: 4.89
- Vertex[] x: -149.393 y: -3.0
- Vertex[] x: -150.127 y: 0.55
- Vertex[] x: -120.51 y: 8.045
- Vertex[] x: -120.051 y: 8.21
- Vertex[] x: -149.456 y: 0.16
- Vertex[] x: -149.393 y: -1.7
- Vertex[] x: -152.995 y: 19.5
- Vertex[] x: -152.363 y: 15.6
- Vertex[] x: -185.167 y: 22.1
- Vertex[] x: -149.962 y: -0.4
- Vertex[] x: -185.145 y: 22.1
- Vertex[] x: -150.171 y: -0.3
- Vertex[] x: -149.383 y: 5.76
- Vertex[] x: -149.962 y: 3.44
- Vertex[] x: -152.915 y: 7.61
- Vertex[] x: -152.336 y: 6.67
- Vertex[] x: -149.393 y: -6.1
- Vertex[] x: -150.171 y: 1.45
- Vertex[] x: -120.51 y: 7.946
- Vertex[] x: -149.393 y: 0.16

In this viewer the 3D vertices are projected into the X-Z plane - this matches up with the W-view clusters

The yellow circles are the candidate vertices

Style | Guides | Clipping | Extras  
GLViewer [TGLSAViewer]  
Update behaviour

Command  
Command (local):



Wheel up - zoom out  
Wheel down - zoom in

Wheel press + drag - pan viewport



W - wireframe mode  
R - return from wireframe mode

# Candidate 3D vertices vs W-view clusters

Expand the list of candidate vertices - there will be many!

Each vertex here is displayed as a yellow circle in the viewer

When you are finished, press Return ↵ to move to the next display

The screenshot shows the 'Eve Main Window' with the '3D View' selected. The left sidebar shows a tree view with 'CandidateVertices3D' expanded, listing numerous vertices with their coordinates (x, y, z). The main 3D view shows a cluster of yellow circles representing these vertices. A coordinate system is visible with axes labeled X, Y, and Z. The Y-axis is green, the X-axis is red, and the Z-axis is blue. The origin is marked with 0, and the Y-axis has a value of 40, the X-axis has -185, and the Z-axis has 409. A blue arrow points from the text 'Each vertex here is displayed as a yellow circle in the viewer' to the cluster of yellow circles. Another blue arrow points from the text 'We're mimicking neutrinos coming in this way' to the Y-axis. The bottom of the window shows a 'Command' field and a 'Command (local)' dropdown.

Each vertex here is displayed as a yellow circle in the viewer

We're mimicking neutrinos coming in this way



Wheel up - zoom out  
Wheel down - zoom in  
Wheel press + drag - pan viewport



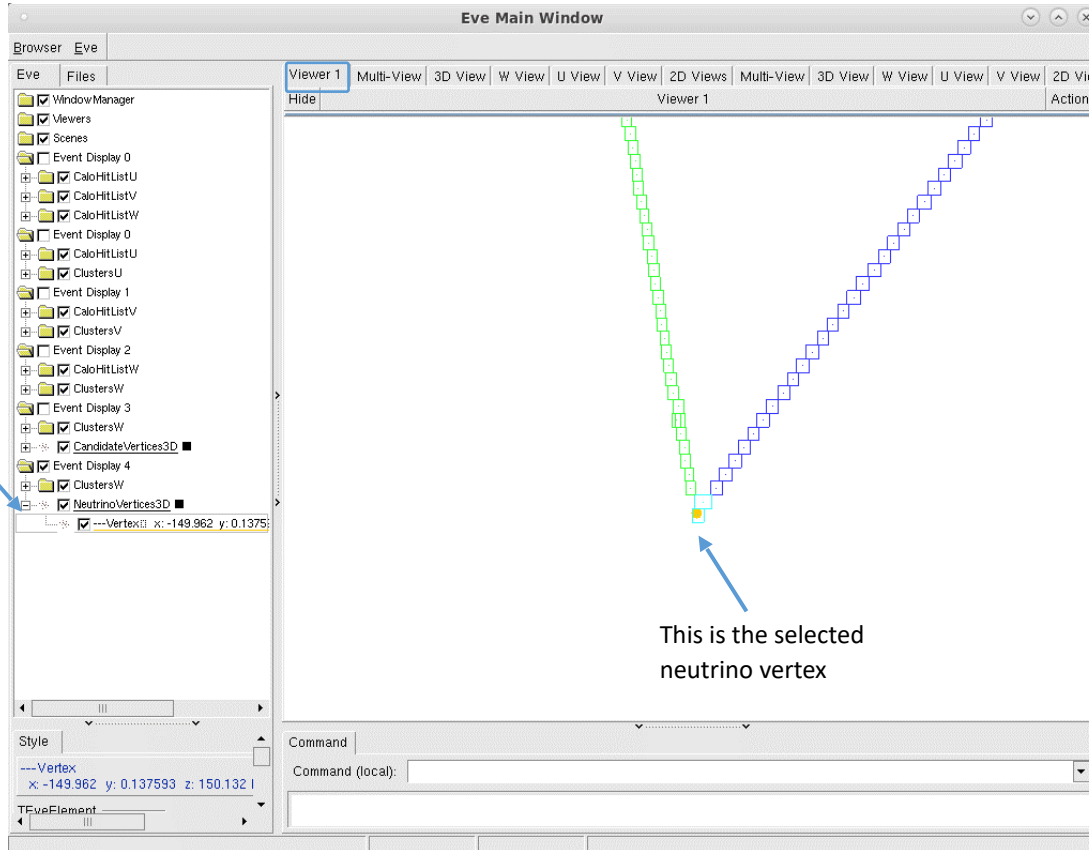
Left press + drag - rotate 3D view



W - wireframe mode  
R - return from wireframe mode



# Selected neutrino vertex vs W-view clusters

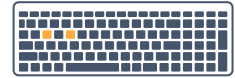


Expand the list of selected neutrino vertices - there should only be one

When you are finished, press Return ↵ to move to the next display



Wheel up - zoom out  
Wheel down - zoom in  
Wheel press + drag - pan viewport



W - wireframe mode  
R - return from wireframe mode

## 3D track & shower reconstruction

## Add in some more visualizations

- Add to `MyPandoraSettings_Neutrino_Standard.xml` at the end of the `TwoDMopUpAlgorithms` section

```

<!-- TwoDMopUpAlgorithms -->
<algorithm type = "LARBoundedClusterMopUp">
  <PfoListNames>ShowerParticles3D</PfoListNames>
  <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>
<algorithm type = "LARConeClusterMopUp">
  <PfoListNames>ShowerParticles3D</PfoListNames>
  <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>
<algorithm type = "LARNearbyClusterMopUp">
  <PfoListNames>ShowerParticles3D</PfoListNames>
  <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>

<algorithm type = "LARVisualMonitoring">
  <PfoListNames>TrackParticles3D ShowerParticles3D</PfoListNames>
  <ShowDetector>true</ShowDetector>
</algorithm>

<!-- ThreeDHitAlgorithms -->
<algorithm type = "LARCUTPFOCharacterisation">
  <TrackPfoListName>TrackParticles3D</TrackPfoListName>
  <ShowerPfoListName>ShowerParticles3D</ShowerPfoListName>
  <PostBranchAddition>true</PostBranchAddition>
  <UseThreeDInformation>>false</UseThreeDInformation>
</algorithm>

```

Visualize the track-like and shower-like reconstructed particles

Add the above lines just before the `ThreeDHitAlgorithms` section

Run Pandora once again!

```

$ /exp/dune/data/users/$USER/reco/work
$ lar -c event_display_driver.fcl -s reco2_1mulp.root -n 1

```

- After the event display has loaded press Return ↵ five times, to skip through our visualizations from parts 1-2

# Reconstructed track & shower-like particles

Expand all of the menus to see the clusters at this point and how they have been matched together into reconstructed particles (PFOs)

Here there are 2 track-like PFOs reconstructed

Hover over a cluster to see which view it belongs to - in this case it's the W view

In this event there are no shower-like particles to see

Viewer 1 Multi-View 3D View W View U View V View 2D Views Multi-View 3D View W View U View V View 2D V

Hide Viewer 1 Action

Each cluster is given a different colour

Remember, in Viewer 1 we display all views on top of each other

Recall previously there were a number of tiny clusters! Now pandora has merged and split them to have zero or one cluster per view per PFO

Clusters are matched between views!

Each PFO has up to one cluster per view

Command  
Command (local):



Wheel up - zoom out  
Wheel down - zoom in  
Wheel press + drag - pan viewport



W - wireframe mode  
R - return from wireframe mode

When you are finished,  
press Return ↵ to  
move to the next  
display

# 3D hit reconstruction

## Add in some more visualizations

- Add to `MyPandoraSettings_Neutrino_Standard.xml` at the end of the `ThreeDHitAlgorithms` section

```

<tool type = "LARMultiValuedTransverseTrackHits"><MinViews>3</MinViews></tool>
<tool type = "LARCLEARTransverseTrackHits"><MinViews>2</MinViews></tool>
<tool type = "LARCLEARLongitudinalTrackHits"><MinViews>2</MinViews></tool>
<tool type = "LARMultiValuedLongitudinalTrackHits"><MinViews>2</MinViews></tool>
</HitCreationTools>
</algorithm>
<algorithm type = "LARThreeDHitCreation">
  <InputPfoListName>ShowerParticles3D</InputPfoListName>
  <OutputCaloHitListName>ShowerCaloHits3D</OutputCaloHitListName>
  <OutputClusterListName>ShowerClusters3D</OutputClusterListName>
  <HitCreationTools>
    <tool type = "LARThreeViewShowerHits"/>
    <tool type = "LARTwoViewShowerHits"/>
    <tool type = "LARDeltaRayShowerHits"/>
  </HitCreationTools>
</algorithm>

<algorithm type = "LARVisualMonitoring">
  <PfoListNames>TrackParticles3D ShowerParticles3D</PfoListNames>
  <ShowDetector>true</ShowDetector>
</algorithm>

<!-- ThreeDMopUpAlgorithms -->
<algorithm type = "LARSlidingConePfoMopUp">
  <InputPfoListNames>TrackParticles3D ShowerParticles3D</InputPfoListNames>
  <DaughterListNames>ClustersU ClustersV ClustersW TrackClusters3D ShowerClusters3D</Daugh
</algorithm>

```

Visualise the reconstructed particles again

Add the above lines just before the  
`ThreeDMopUpAlgorithms` section

Run Pandora once again!

```
$ cd /exp/dune/data/users/$USER/reco/work
```

```
$ lar -c event_display_driver.fcl -s reco2_1mulp.root -n 1
```

- After the event display has loaded press Return ↵ six times, to skip through our visualisations from parts 1-3

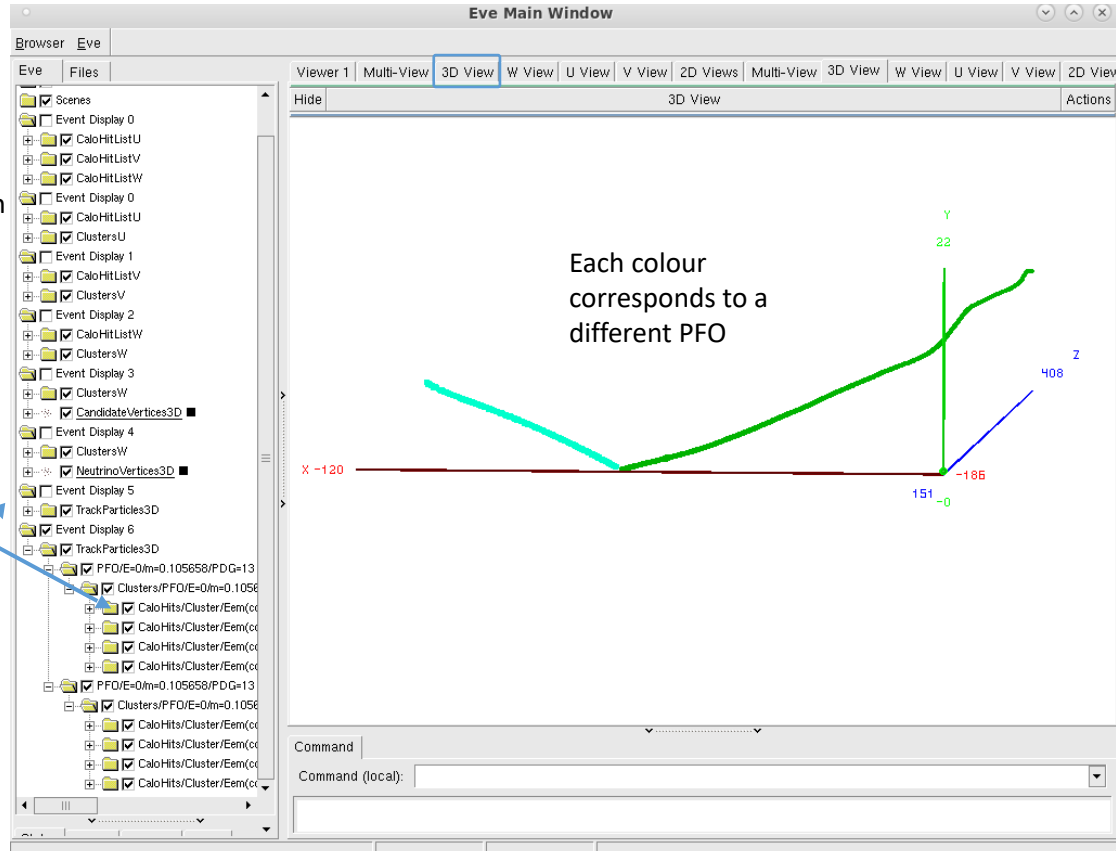
## 3D hits

When you are finished,  
press Return ↵ to  
close the event display

Expand all of the menus again  
to see what we now have

Our PFOs now have a new  
cluster of 3D hits that we  
have just created

Note that we use different  
3D hit creation algorithms  
depending on the PFOs  
track-shower classification



Wheel up - zoom out  
Wheel down - zoom in  
Wheel press + drag - pan viewport



Left press + drag - rotate 3D view



W - wireframe mode  
R - return from wireframe mode

# Neutrino hierarchy reconstruction



## Add in some more visualizations

- Add to `MyPandoraSettings_Neutrino_Standard.xml` at the end of the file

```

<OutputVertexListName>DaughterVertices3D</OutputVertexListName>
</algorithm>
<algorithm type = "LArNeutrinoProperties">
  <NeutrinoPfoListName>NeutrinoParticles3D</NeutrinoPfoListName>
</algorithm>

<!-- Track and shower building -->
<algorithm type = "LArTrackParticleBuilding">
  <PfoListName>TrackParticles3D</PfoListName>
  <VertexListName>DaughterVertices3D</VertexListName>
</algorithm>

<!-- Output list management -->
<algorithm type = "LArPostProcessing">
  <PfoListNames>NeutrinoParticles3D TrackParticles3D ShowerParticles3D</PfoListNames>
  <VertexListNames>NeutrinoVertices3D DaughterVertices3D CandidateVertices3D</VertexListNames>
  <ClusterListNames>ClustersU ClustersV ClustersW TrackClusters3D ShowerClusters3D</ClusterListNames>
  <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW CaloHitList2D</CaloHitListNames>
  <CurrentPfoListReplacement>NeutrinoParticles3D</CurrentPfoListReplacement>
</algorithm>

<algorithm type = "LArVisualMonitoring">
  <ShowCurrentPfos>true</ShowCurrentPfos>
  <ShowDetector>true</ShowDetector>
</algorithm>
</pandora>

```

Visualize the final reconstructed particles

Run Pandora once again!

```

$ cd /exp/dune/data/users/$USER/reco/work
$ lar -c event_display_driver.fcl -s reco2_1mulp.root -n 1

```

- After the event display has loaded press Return ↵ seven times, to skip through parts 1-4

# The final outcome

Expand all of the menus again to see what we now have

The PFOs are now arranged in a hierarchy! The top-level PFO has PDG code = 14  $\Rightarrow \nu_\mu$

The neutrino PFO has 2 daughter PFOs which each have clusters of 2D & 3D hits

The PFOs have been classified once more as track-like (assigned PDG 13) or shower-like (assigned PDG 11 – none here)

Every PFO has a vertex this is the reconstructed start position

The screenshot shows the 'Eve Main Window' interface. On the left, the 'Browser' pane displays a hierarchy of objects under 'Eve'. The top-level object is 'WindowManager', followed by 'Viewers', 'Scenes', and 'Event Display 0'. Under 'Event Display 0', there are several 'CaloHitList' objects (U, V, W) and 'Event Display 1'. The main area is divided into three viewports: '3D View' (top left), '2D U View' (top right), '2D V View' (middle right), and '2D W View' (bottom right). The 3D View shows a yellow track-like object. The 2D views show hit clusters with red and blue lines and numerical labels like '-185' and '-120 X'. A 'Command' field is at the bottom.

When you are finished,  
press Return  $\leftarrow$  to  
close the event display

## Got spare time?

Run your FHiCL file again over multiple events

Do you understand what Pandora is doing in each of the steps?