

# Testing new Pandora developments in analysis

Ryan Cross, Isobel Mawby, Maria Brigida Brunetti, Andy Chappell

19/08/2024

The University of Warwick

The logo for The University of Warwick, featuring a stylized white mountain peak or 'W' shape above the word 'WARWICK' in a dark blue, sans-serif font.

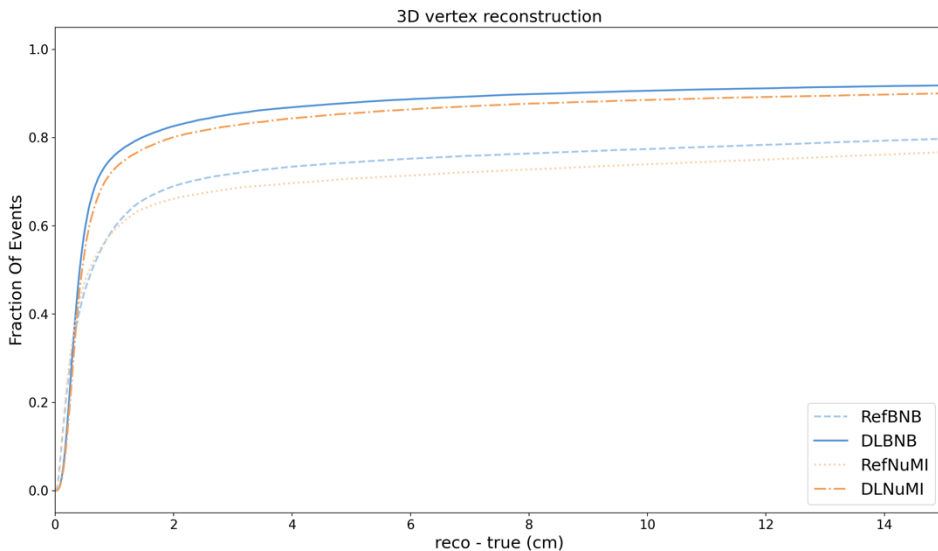
WARWICK

## Goals

- The primary goal is to test Pandora reconstruction improvements (in MCC9) in the context of your analysis
- The new improvements are DL vertexing, and shower growing updates
- The idea is to run tests on modest stats, to compare default MCC9 to the updates
- Aim at hand scanning and understanding how the updates might affect analyses

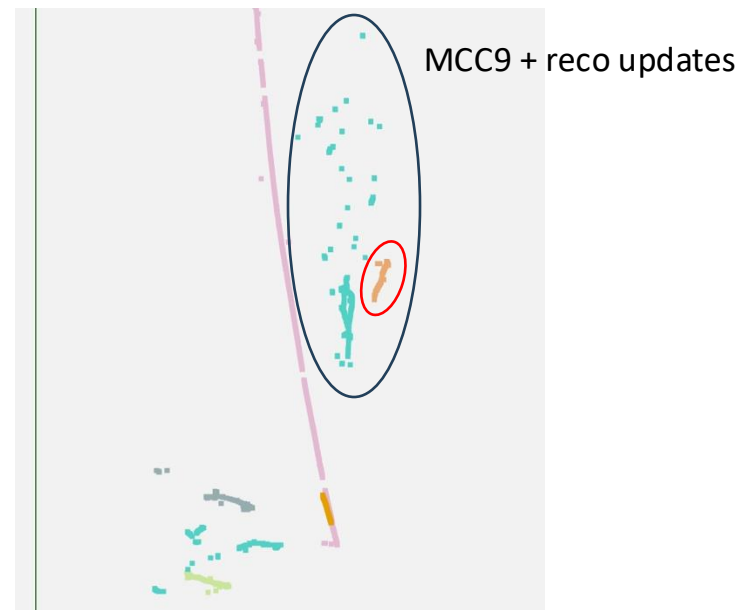
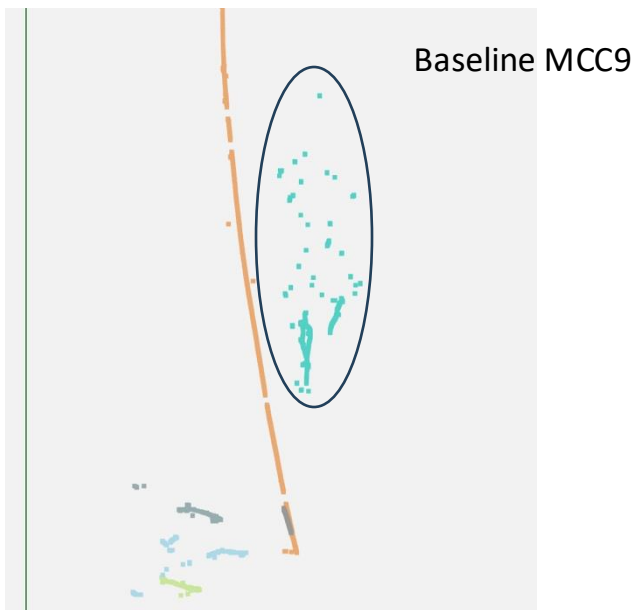
# Vertexing Updates

- Pending merge into uboonecode of the new updates
  - Notable increase in vertex reconstruction efficiency in both BNB and NuMI
  - Network has so far only been trained with BNB events
  - We anticipate NuMI performance could be improved with additional training



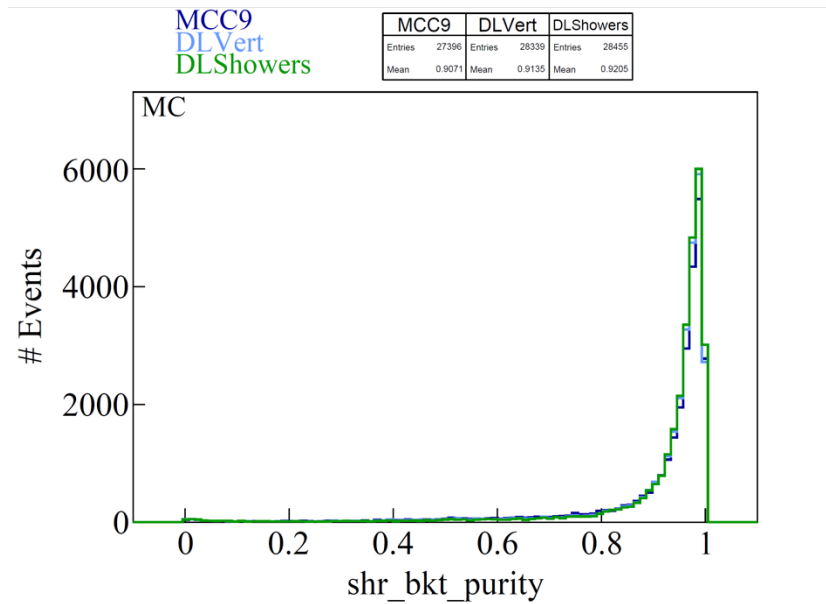
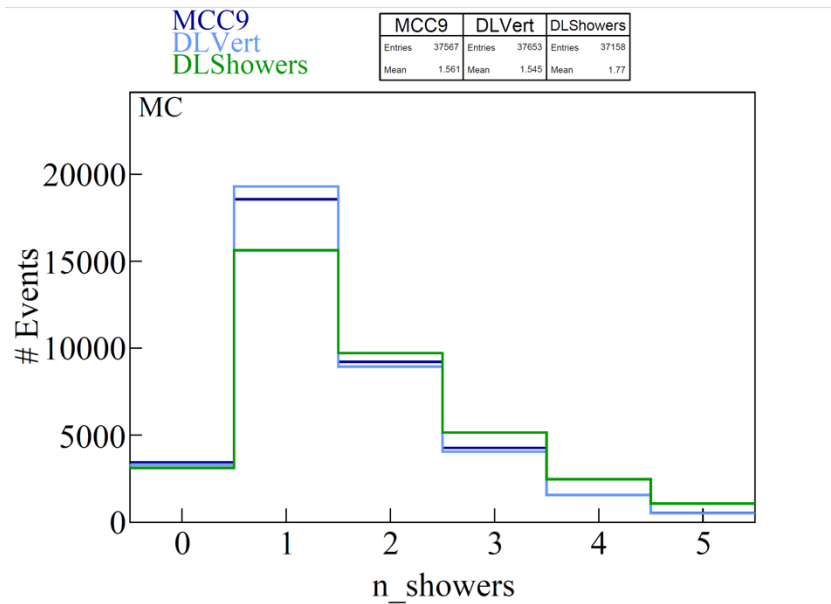
# Shower Updates

- Changes make shower growing less greedy
  - Reduces over-merging of showers



## Shower Updates (2)

- Changes make shower growing less greedy
  - Reduces over-merging of showers
  - Total shower count increases
  - Purity of each shower also increases



## Testing new developments - recipe

1. Setting up MCC9
2. Creating a local build
3. Checking out and building with Ryan's branch
4. Running reco2
5. Running comparison studies between standard workflow and new development workflow
  1. Use Ryan's comparison scripts (based on Afro's scripts)
  2. Look at variables that matter to your analysis
  3. Hand-scan individual events, perform visual debugging (Isobel kindly provided a friendly event display)

Ryan has put together instructions at: [/exp/uboone/app/users/rcross/public/microboone\\_reco](/exp/uboone/app/users/rcross/public/microboone_reco)

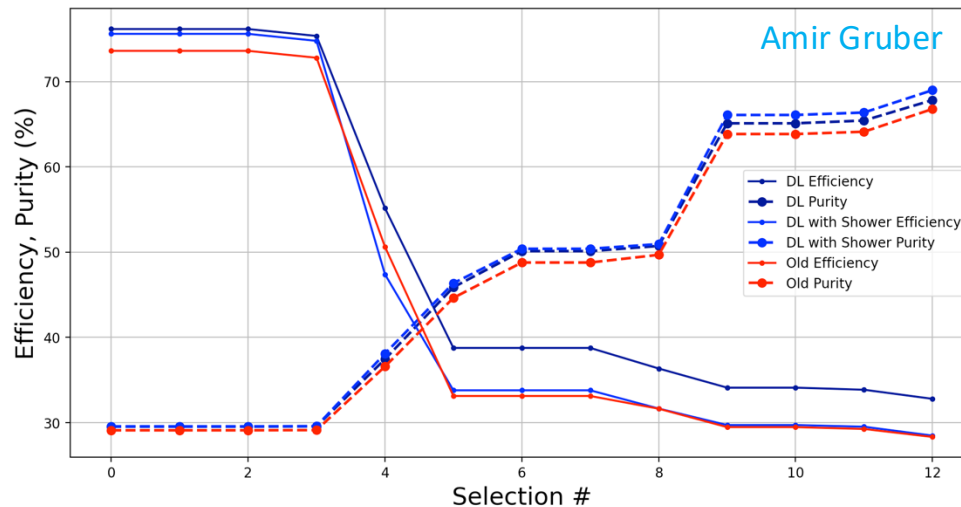
Backup slides contain these instructions, and those to run Isobel's event display

## A note on analysis-specific tests

- You are the expert here!
- Look at metrics that matter to your analysis
  - Larger reconstructed samples (e.g. due to recovering events where no vertex was previously found, affecting downstream reconstruction)
  - Better handles on PID or other important selection variables
  - Improved precision on final measurements
  - Etc.
- When there are improvements, or where improvements are expected but not found, it would be interesting to debug, including hand-scanning individual events

# An example: 1e1p0pi cross-section analysis

- Amir is testing the new DL vertexing on Maitreyee's 1e1p0pi cross-section analysis to test impact on selection efficiency
- Initial positive results
- More work is ongoing





## Conclusions

- Finally, thank you very much being here and for your efforts on this.
- We hope that this will be a productive and useful week!





## MCC9 setup instructions

- Load into an SL7 container, since none of the MicroBooNE Code works on AL9.

```
apptainer exec \
```

```
-B /cvmfs,/exp/uboone,/pnfs/uboone,/run/user,/etc/hosts,/etc/localtime \
```

```
--env UPS_OVERRIDE='-H Linux64bit+3.10-2.17' \
```

```
"/cvmfs/uboone.opensciencegrid.org/containers/uboone-devel-sl7" \
```

```
/bin/bash
```

- Source the MCC9 MicroBooNE setup script

```
source /cvmfs/uboone.opensciencegrid.org/products/setup_uboone_mcc9.sh
```

- Load the MicroBooNE software stack and some associated bits...

```
setup uboonecode v08_00_00_82 -q "e17:prof"
```

```
setup ninja v1_11_1
```

## Make a local MCC9 build

- After running the base MCC9 setup, if you want to start making changes, say to reconstruction or analysis code, you will need to make your own local build.

- Switch to `uboonebuild03.fnal.gov`, then move to your user app directory

```
cd /exp/uboone/app/users/$USER/
```

- Make a sensible folder to store your new build in

```
mkdir mcc9_local_valid
```

```
cd mcc9_local_valid
```

- Run `mrB` in that folder...

```
mrB newDev
```

```
source localProducts_*/setup
```

## Checking out the new Pandora developments

- The DL Vertexing requires to update different repositories:
  - **larpandoracontent** - This is the main Pandora algorithm project, so the DL vertexing itself lives here.
  - **larpandora** - The interface between LArSoft + Pandora.
  - **ubreco** - This wraps the Pandora code, so we need to add a new "Deep Learning" aware wrapper.
- It is best to run a compile between each of these steps, as there are some complicated dependencies between each of the steps: ensure that your environment contains a built version of DL Pandora library, before trying to build ubreco.

## Checking out the new Pandora developments (2)

- First, cd to the srcs folder

```
cd $MRB_SOURCE
```

- Run the setup commands from slide 7-8, including source localProducts
- Grab the larpandoracontent git repo...

```
mr b g larpandoracontent
```

- Checkout Ryan's branch with new developments

```
cd larpandoracontent
```

```
git remote add crossr https://github.com/CrossR/LArContent.git
```

```
git fetch --all
```

- Swap to the DL Vertexing branch

```
git checkout CrossR/mcc9_stable/DLVertexing
```

## Checking out the new Pandora developments (3)

- Now we can repeat that process for LArPandora

```
cd $MRB_SOURCE
mrb g -t v08_03_00_23 larpandora
cd larpandora
git remote add crossr https://github.com/CrossR/LArPandora.git
git fetch --all
git checkout CrossR/uboone_mcc9_dl_vertexing
```

- At this point, you now have a copy of all the DL Vertexing Pandora code, so let's build it (next slide)
- Afterwards, you will also checkout and build ubreco

# Building with the new Pandora developments

- Swap to the build folder

```
cd $MRB_BUILDDIR
```

- Setup the environment. You should only need to do this once, if you are doing repeated builds.

```
mrbsenv
```

- Kick off a build

```
mrbs --generator ninja -i
```

- If you are doing a follow-up build, the above might fail. Instead do the following to clean up and start over

```
mrbs z
```

```
mrbsenv
```

```
mrbs --generator ninja -i
```



## Checkout and build ubreco

```
cd $MRB_SOURCE
```

```
mrblib g ubreco
```

```
cd ubreco
```

```
git fetch --all
```

```
git checkout feature/rcross_pandoraDL
```

- Copy over the DL Vertexing files before building

```
cp /exp/uboone/app/users/rcross/public/microboone_reco/dl_vertexing/*.pt  
ubreco/MicroBooNEPandora/scripts/
```

## Building ubreco

- And if you are doing a full DL Vertexing build it can get a bit messy
- The DL Vertexing changes add a new DL version of Pandora, that by default, ubreco isn't aware of. So, you should follow the normal build instructions, and it will likely fail and throw up an error about not being able to link / find things. You should have already build larpandoracontent and larpandora at this point. You are going to load those 2 built bits, and re-build ubreco using them
- Try a regular build as above first. If that isn't happy, from the build dir, first source your locally built products.

mrbslp

- With your local bits loaded, force CMake to re-run and pick up the new bits

touch \$MRB\_SOURCE/ubreco/CMakeLists.txt

- Finally, re-build and you should see CMake re-run first

## Running reco2 to produce PeLEE ntuples

- A number of hopefully well named FCL files for running the PeLEE ntuple generation, re-running reco2 and running the Pandora DL vertexing can be found in:

`/exp/uboone/app/users/rcross/public/microboone_reco/fcls/`

- `run_reco2_with_dl_vertexing_run3.fcl`
- `run_reco2_run3.fcl`

The fcls you run for your analysis may differ!

- The above fcl files require the following Pandora XML settings files that enable DL vertexing, and can be found in `/exp/uboone/app/users/rcross/public/microboone_reco/fcls/xml`
- `PandoraSettings_Master_MicroBooNE_DLVertexing.xml`
- `PandoraSettings_Neutrino_MicroBooNE_DLVertexing.xml`

## Sample samweb definitions

- A number of hopefully well named FCL files for running the PeLEE ntuple generation, re-running reco2 and running the Pandora DL vertexing can be found in:

`/exp/uboone/app/users/rcross/public/microboone_reco/fcls/`

- `run_reco2_with_dl_vertexing_run3.fcl`
- `run_reco2_run3.fcl`
  
- The above fcl files require the following Pandora XML settings files that enable DL vertexing, and can be found in `/exp/uboone/app/users/rcross/public/microboone_reco/fcls/xml`
  - `PandoraSettings_Master_MicroBooNE_DLVertexing.xml`
  - `PandoraSettings_Neutrino_MicroBooNE_DLVertexing.xml`

## Ryan's plotting scripts

- Plotting Scripts are in `/exp/uboone/app/users/rcross/public/microboone_reco/plotting_scripts/`
- `GetEventsList.C`
  - Loads two (PeLEE) ntuple files and pulls out the same set of events from both.
  - Use it by calling ``GetEventsList.C`` then the old, new and the target tree.
  - If using on a non-PeLEE file, may need the run, subrun and event branch names changing.
  - Usage example:

```
root
GetEventsList.C(\"old/nue/neutrinoselection_filt.root\", \"dl/nue/neutrinoselection_filt.root
\", \"nuselection/NeutrinoSelectionFilter\");
```

## Ryan's plotting scripts (2)

- Plotting Scripts are in `/exp/uboone/app/users/rcross/public/microboone_reco/plotting_scripts/`
- RecoComparison.C
  - This builds up the actual comparison histograms, based on Afro's scripts ([https://github.com/afropapp13/myPreSelection/tree/master/Run\\_4\\_5\\_Validation](https://github.com/afropapp13/myPreSelection/tree/master/Run_4_5_Validation))
  - Use it by updating the two paths at the top of that file to point to your files that were output from the GetEventsList.C script.
  - Usage example:

```
root -q RecoComparison.C
```

## Ryan's plotting scripts (3)

- Plotting Scripts are in `/exp/uboone/app/users/rcross/public/microboone_reco/plotting_scripts/`
- `compare_dr.py`
  - This is just an example of a Python script using `uproot` to plot some values from the filtered root ntuple files.
  - Once you have a python env with `uproot` / `matplotlib` / `numpy` you can then run:

```
python compare_dr.py /path/to/reference/file/filteredReference.root  
/path/to/new/file/filteredCurrent.root
```

# Running Isobel's event display

- Isobel has created a fantastic event display that you can use as an alternative to existing tools, and they have kindly put together instructions to run this:
- <https://indico.fnal.gov/event/65564/contributions/298469/attachments/180898/247980/EventDisplayInstructions.pdf>

