

# DUNE FD Production - Timing Offset Issue

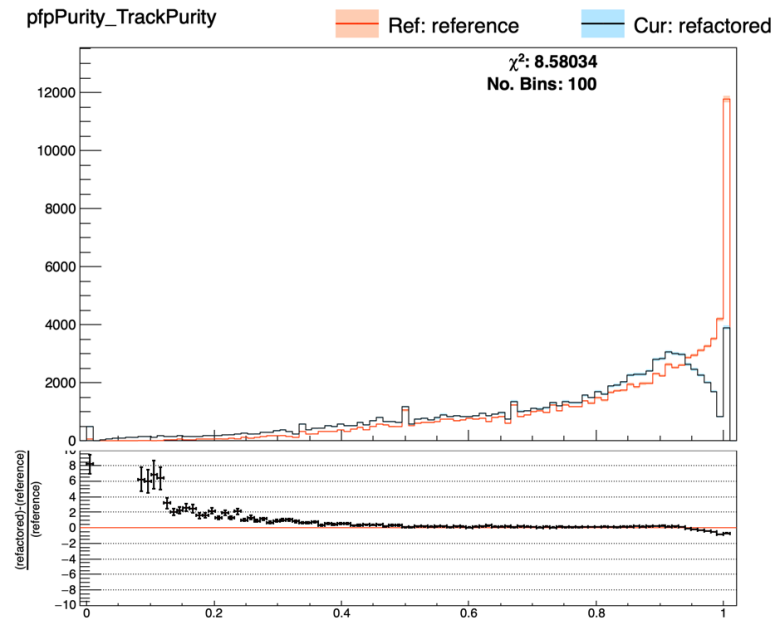
Ryan Cross



# Where we are...

I gave a previous talk that showed the current look of some reconstruction metrics for Pandora with the new production validation files, comparing them to a non-refactored version of `dunetpc`.

There was an unexplained difference with the tracks, which I needed to look into. Showers were being affected by a different bug, so the issues there were less easy to spot.

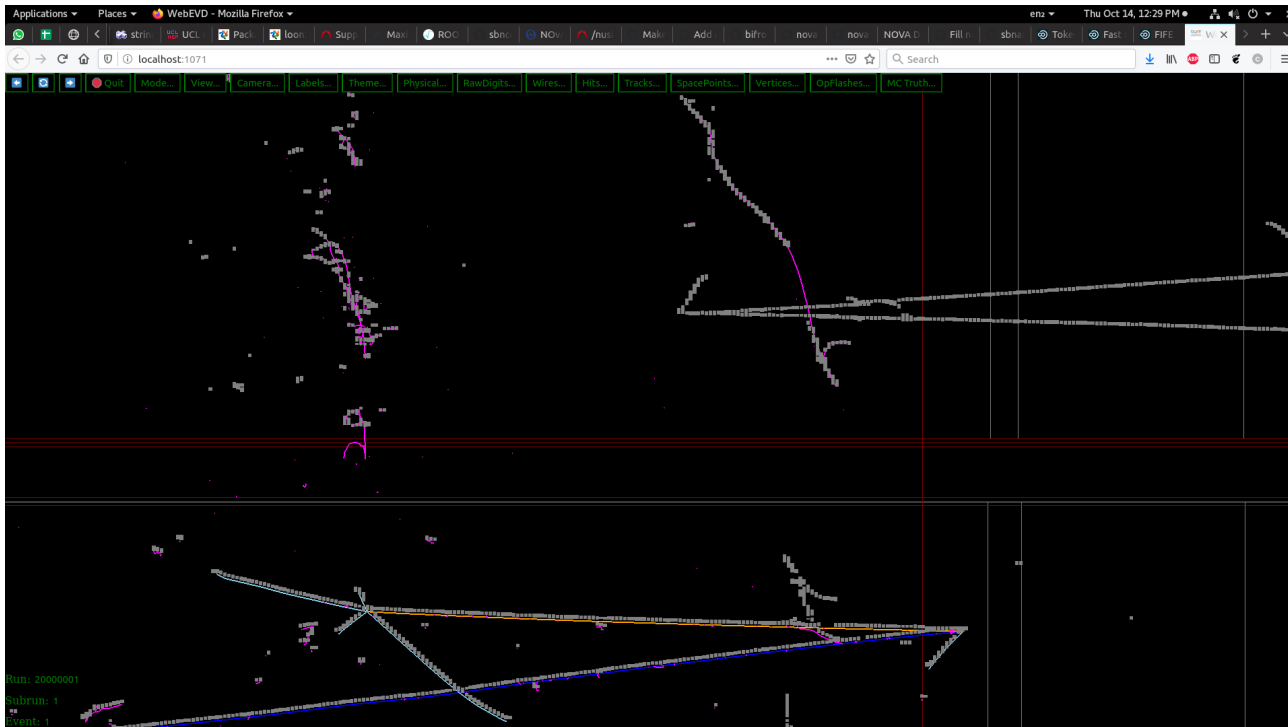


Previous previous slides are [here](#).

# Where we are...

Additionally, gave a second update showing that there was seemingly an odd timing offset seen when using the refactored workflow.

This offset seemingly wasn't uniform (which was throwing 3D hit generation off), but there wasn't an obvious explanation for why there was a non-uniform offset once using the outputs from `wirecell`.



Previous slides are [here](#).

# Software Assumptions

After some further investigation and checking the raw digits and wires, James Shen found a previous discussion in the context of ProtoDUNE-SP where they had and fixed the same issue!

Tingjun was able to confirm the issue and the fix: **Normally, the detector property service corrects the offset between planes. However, wirecell also does this and saves it, so if the detector property service is left as normal, it can be applied twice.**

The fix then, which is in place in the `DetectorPropertiesServiceProtoDUNESp`, is to not apply the offset.

More details are in the `#lartpc_view_timing_offset` slack channel.

# FCL Config Used

This can fairly easily be tested, though a longer term solution, and perhaps a larger conversation around this will be needed, as the PDSP service, is unsurprisingly, specific to PDSP and does stuff we don't need or use.

Minimal changes are needed in the FCL, mostly to disable various loading things since the FD doesn't need to look anything up, and remove stuff it doesn't need.

```
services.DetectorPropertiesService.service_provider:  
    "DetectorPropertiesServiceProtoDUNEsp"  
  
services.DetectorPropertiesService.GetHVDriftfromSamweb: false  
services.DetectorPropertiesService.GetReadOutWindowSizefromSamweb: false  
services.DetectorPropertiesService.UseRunDependentTemperature: false  
  
services.DetectorPropertiesService.DriftVelFudgeFactor: @erase  
services.DetectorPropertiesService.UseIcarusMicrobooneDriftModel: @erase
```

# Code Changes

The only other required (well...required is a strong word, it was the easiest change) was to disable part of the IFDH bit of the PDSP detector service, since again we can't / don't want to load any information from DBs etc that don't exist.

This is not a "fix" of any sort, just an easy hack to test with.

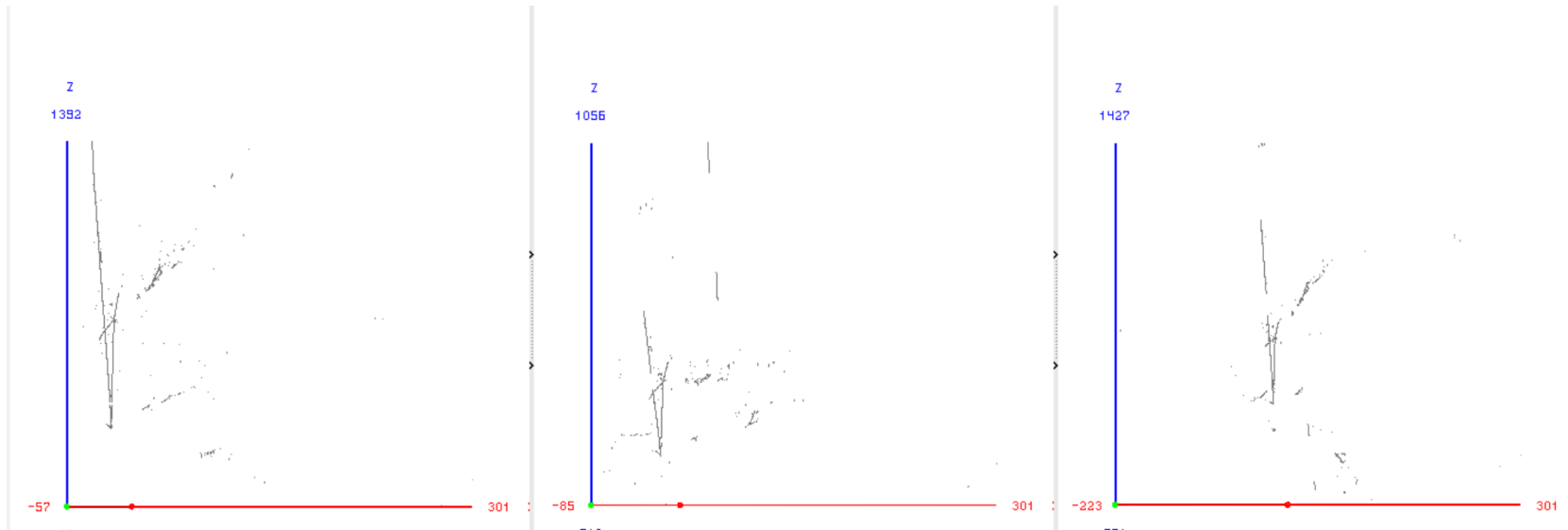
```
diff --git a/dune/Protodune/singlephase/DetectorServices/Services/DetectorPropertiesServicePr
index 0be82d8..ff8f893 100644
--- a/dune/Protodune/singlephase/DetectorServices/Services/DetectorPropertiesServicePr
+++ b/dune/Protodune/singlephase/DetectorServices/Services/DetectorPropertiesServicePr
@@ -104,7 +104,7 @@ namespace spd{

    // std::cout<<"new run?"<<isNewRun<<std::endl;
-   if(isNewRun){
+   if(false){

    auto start = filename.rfind("/"); //finds the final "/"
    if (start == std::string::npos)
```

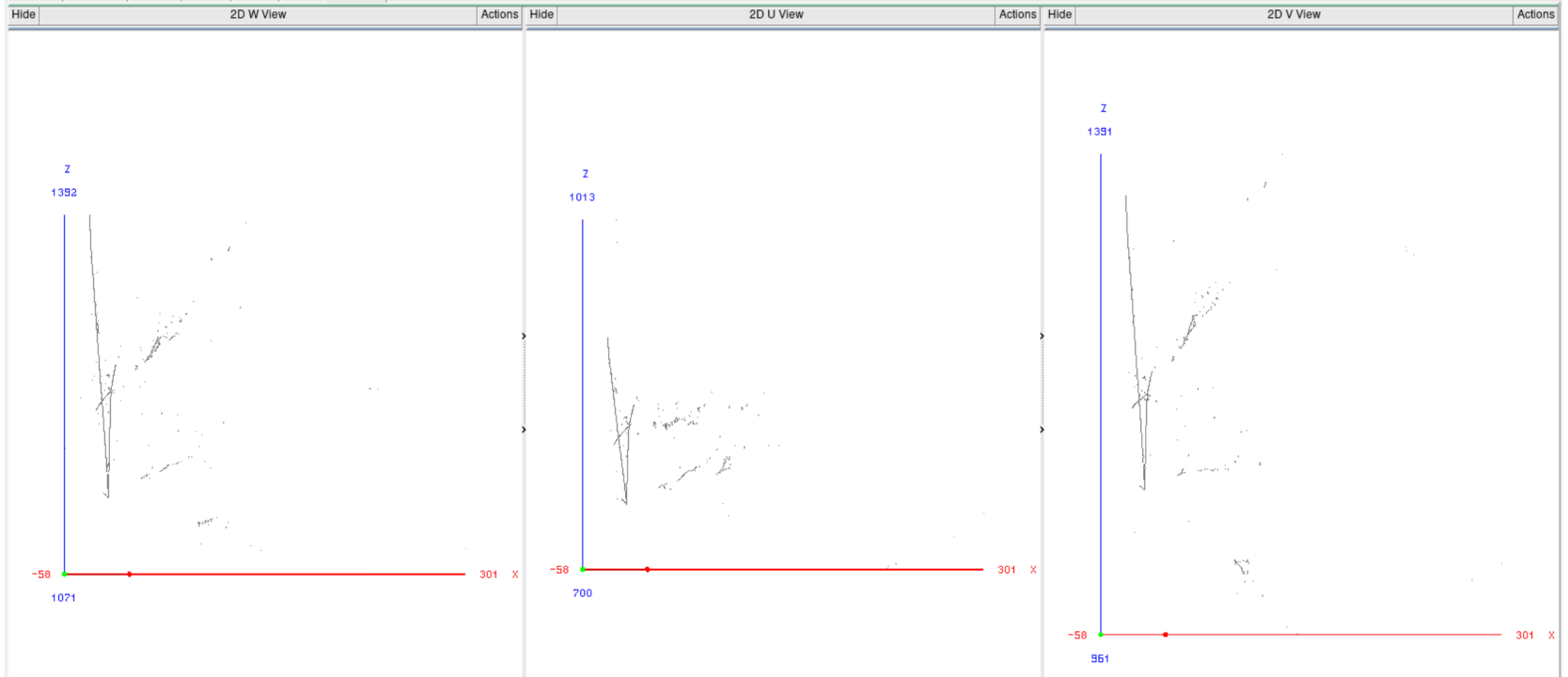
# Impact

So, with that change we end up with the following differences in Pandora and in the spacepoints.



# Impact

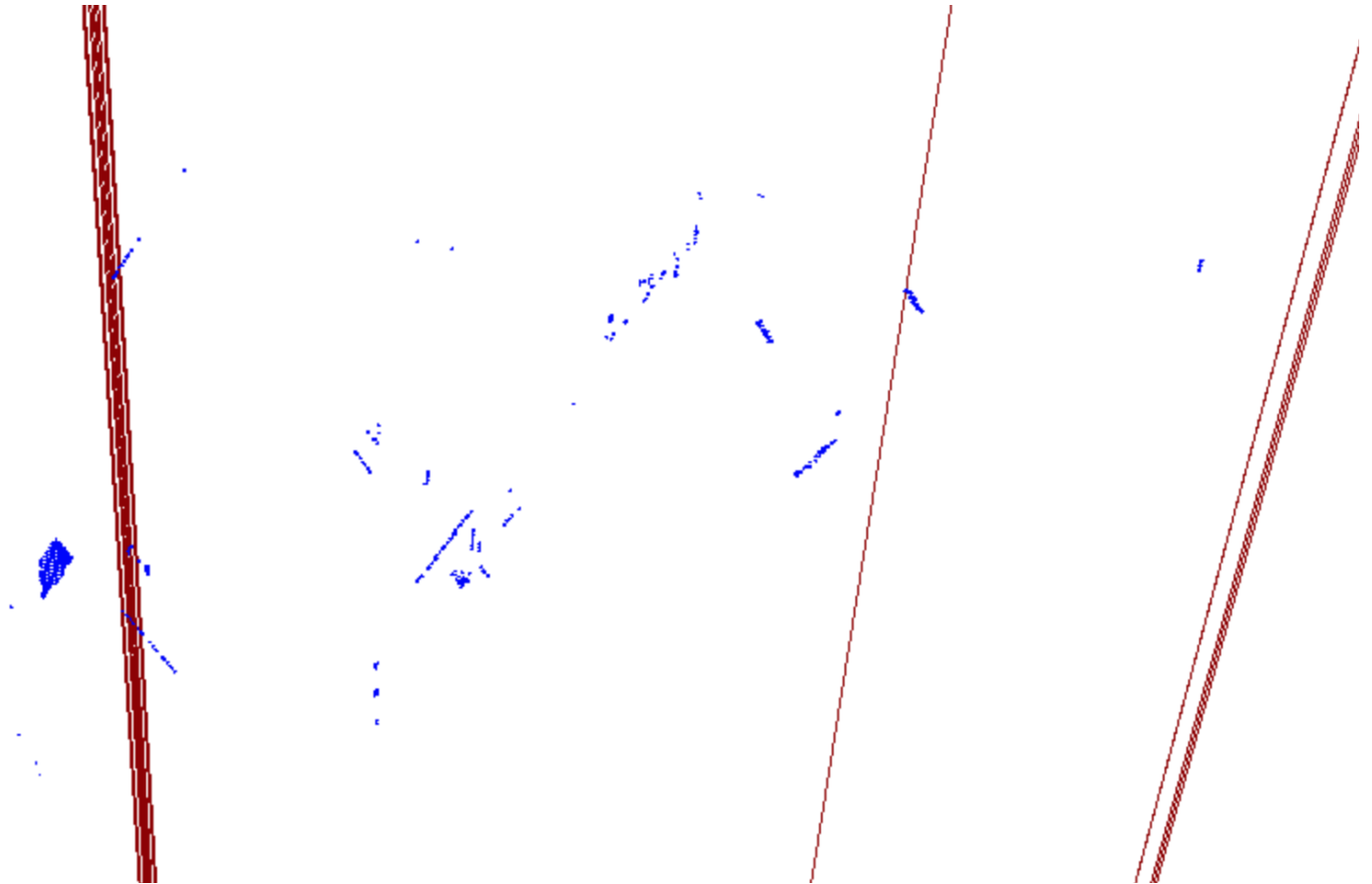
So, with that change we end up with the following differences in Pandora and in the spacepoints.





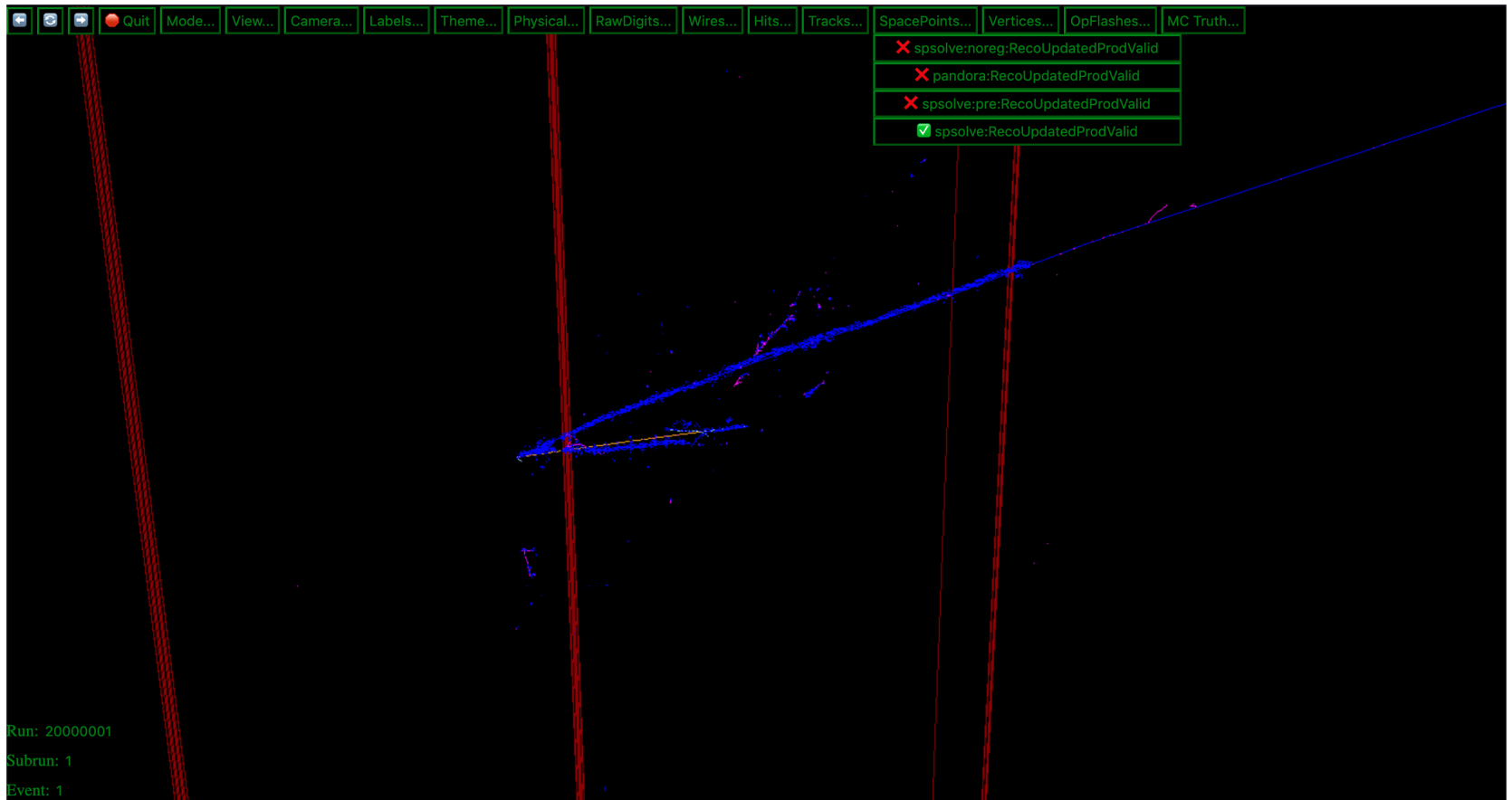
# Impact

So, with that change we end up with the following differences in Pandora and in the spacepoints.



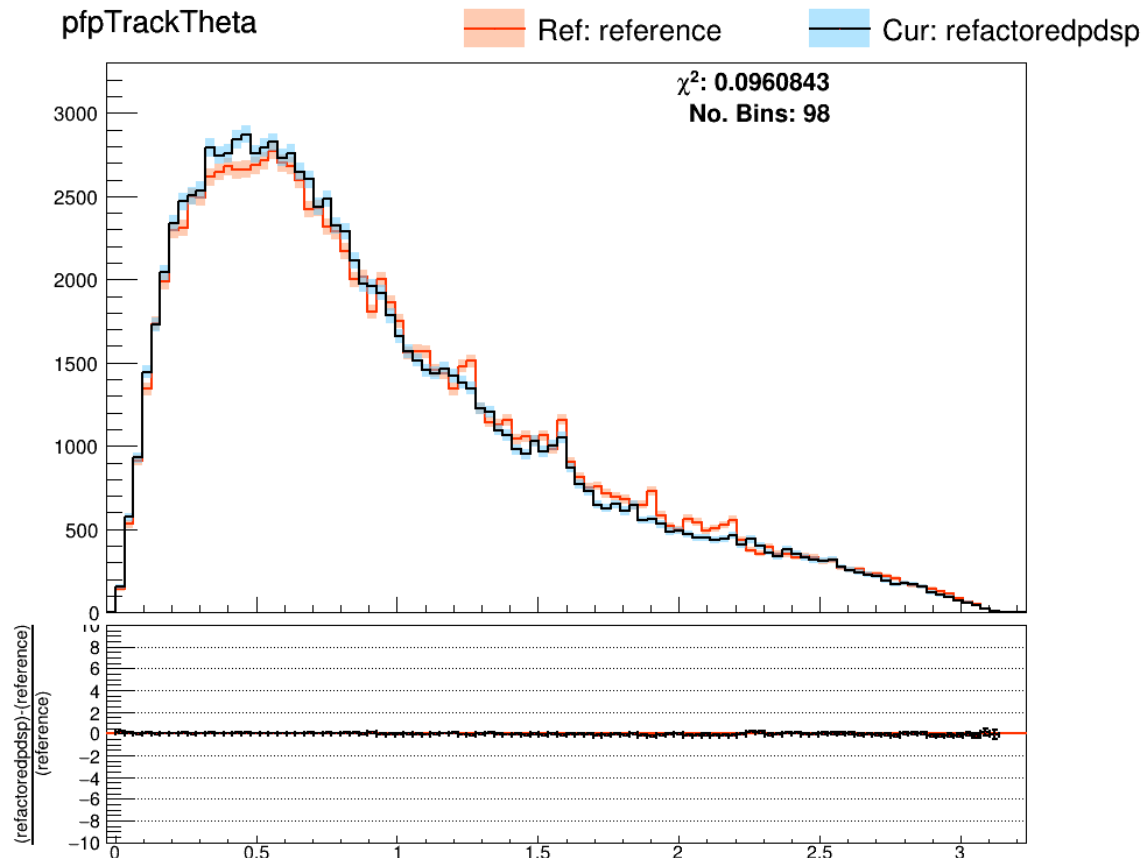
# Impact

So, with that change we end up with the following differences in Pandora and in the spacepoints.



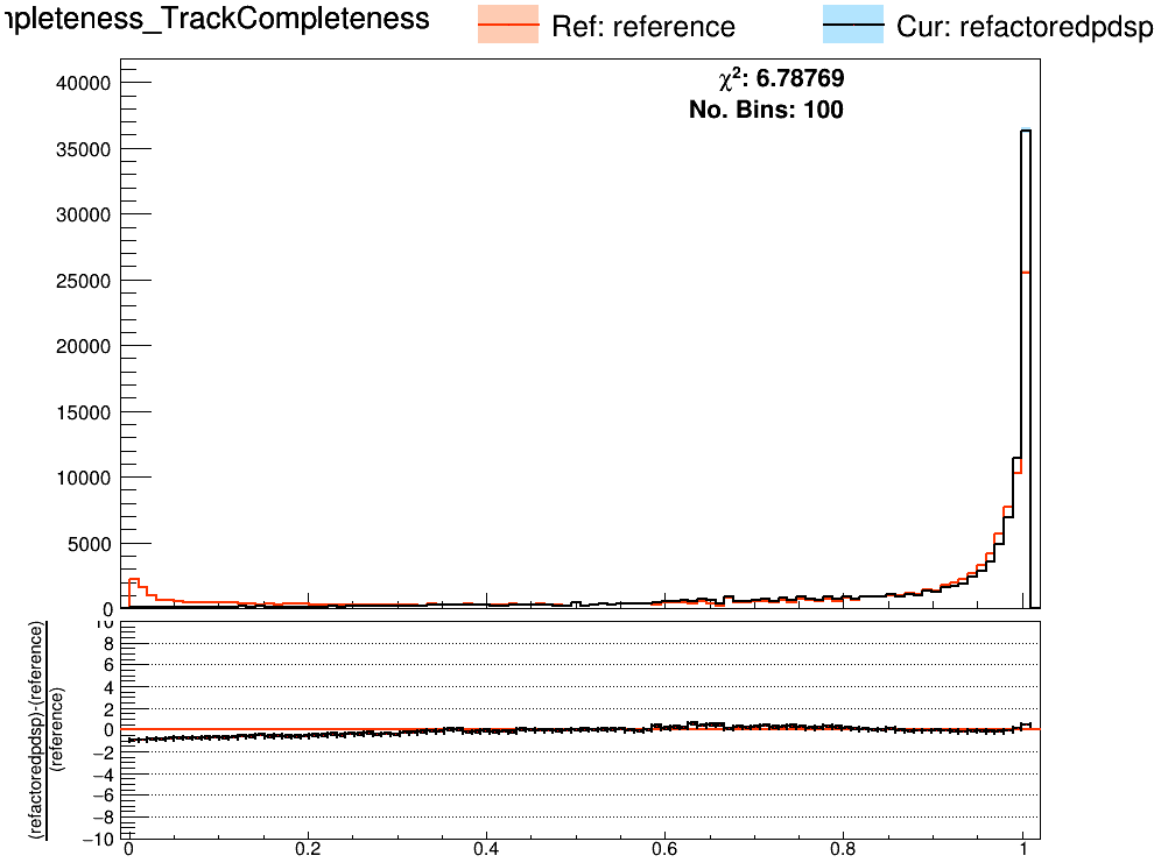
# Pandora Metrics

Finally, with this "fix" in place, I can go back and re-process the files and check the metrics for the tracks again, to see how they look. Broadly, things look a lot better, and the track based features are now very similar (small changes are expected).



# Pandora Metrics

However, there is seemingly still some issues that haven't been accounted for. I'm leaning towards these being similar to / caused by the shower issues as now the event displays look sensible now.



# Final Notes

So overall, it looks like the change to the detector service has fixed the issues with the 2D hits we were seeing.

A few points to think about / work on going forward:

- What change do we want to make to the detector service? Hacking the PDSP service like I have isn't really a real solution, but do we edit the existing service? Add a new one? FHICL option?
- There is potentially a slightly larger scale chat needed as well, as `wirecell` is in use or considered for basically the other selection of LArTPC experiments. I believe the same assumption is made everywhere, and it would be nice if every group didn't need to re-discover this issue and then the fix each time.
- Need a final pass over the metrics to check what is causing the final issues. There is still an issue with the showers MC which muddies the plots somewhat, so could be related to that. Can test and compare a few test files I have.

# Reconstruction Issues Exploration for HD Production

Ryan Cross



# Backup Slides