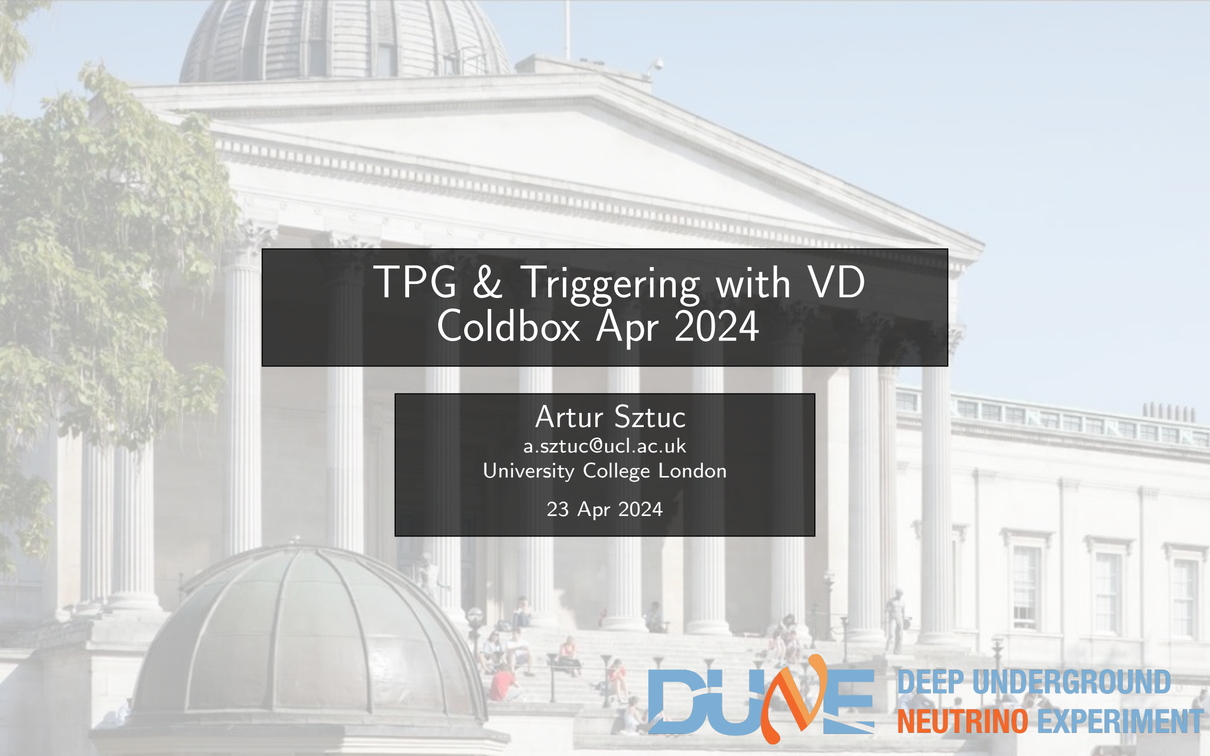


# TPG & Triggering with VD Coldbox Apr 2024

Artur Sztuc
a.sztuc@ucl.ac.uk
University College London

23 Apr 2024

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

# Introduction

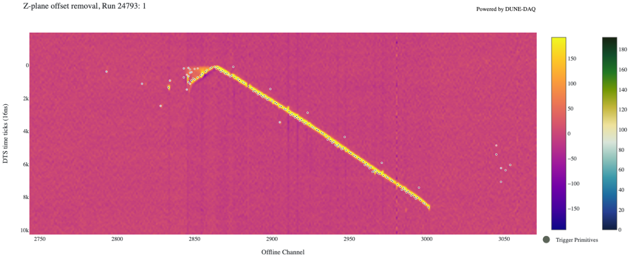
Goals:

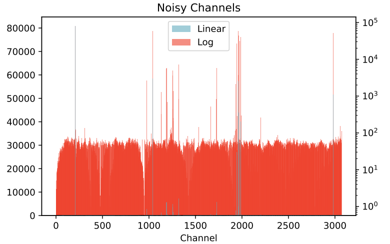
1. Obtain stable TPG via parameter scans for all three TPG algorithms
   - SimpleThreshold, StandardRS, AbsRS
2. Select TPG algorithm most suitable for software triggering
3. Test pre-prepared software triggering algorithm configurations
   - HorizontalMuon, DBSCAN, ChannelDistance, ChannelAdjacency
4. Find suitable candidates for PD2.
5. Collect raw waveforms with long tracks for TPG studies.

We occupied coldbox for five days, and in the end, reached all the goals!

# TPG studies

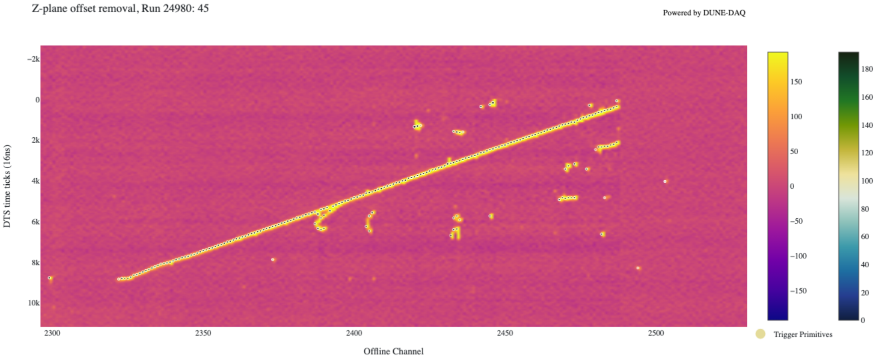
- We started by running all three TPG algorithms with default configs.
  - Ivana was using that data to calculate RMS of noise to set the thresholds.
  - We also used this data to find hot channels – there were a few.
- We then started running threshold scans for each of the three algorithms.
  - SimpleThreshold: thresholds as low as 45 were giving good TPs & low noise.
  - StdRS: Initially best for induction, very low TP rates, noisy on Collection.
  - AbsRS: Found it difficult to tune to give low noise with tracks without gaps.
- We also found ghost TPs.





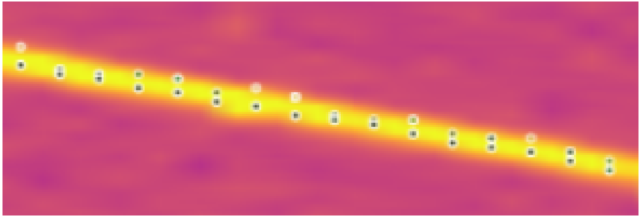
# TPG fix

- We had to re-make the hot channel mask every morning.
- Adam spent ∼2 days with us fixing the TPG ghosts – and succeeded.
- After this, we managed to get good TPG configs with all three algorithms.
- StandardRS was giving the lowest TP rates, with nice, unbroken tracks.
    - On all three planes!
    - SmpTh on collection & StdRS on induction could give us better results.
    - Require different thresholds – not supported on different planes (yet).



# TPG studies summary

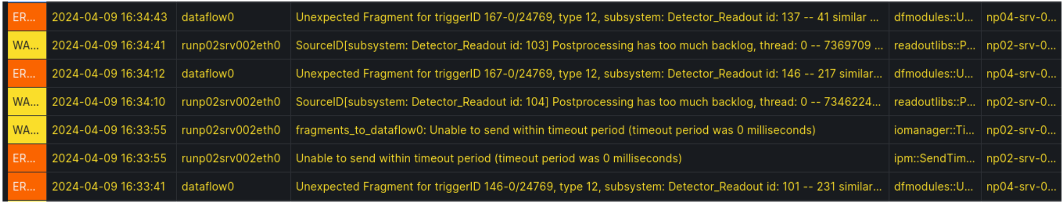
- We found stable configurations with all three algorithms.
- SimpleThreshold gave us cleanest tracks on collection, not induction.
- AbsRS was difficult to tune, margins for stable config seemed narrow.
  - It would often give us double TPs: somewhat unavoidable.



- SimpleRS was giving us very clean, unbroken tracks on induction channels.
  - After TPG ghost fix, even the collection plane TPs looked good!
  - This is what we ended up using, without any plane masking.
- Preferable option: StandardRS on induction, SimpleThreshold on induction.
  - Different TPG thresholds/parameters on different planes not supported.

# Software triggering: DBSCAN

- First tested DBSCAN with config Alex prepared using Nov 2023 VD CB.
- Trigger rates were $\sim$ order magnitude larger, dfo & readout warnings/errors.
- Strangely enough, when replaying with the same data, we get some of these errors, but not with Nov 2023 VD CB!



```
ER...  2024-04-09 16:34:43  dataflow0          Unexpected Fragment for triggerID 167-0/24769, type 12, subsystem: Detector_Readout id: 137 -- 41 similar ...   dfmodules::U...  np04-srv-0...
WA...  2024-04-09 16:34:41  runp02srv002eth0   SourceID[subsystem: Detector_Readout id: 103] Postprocessing has too much backlog, thread: 0 -- 7369709 ...   readoutlibs::P...  np02-srv-0...
ER...  2024-04-09 16:34:12  dataflow0          Unexpected Fragment for triggerID 167-0/24769, type 12, subsystem: Detector_Readout id: 146 -- 217 similar...   dfmodules::U...  np04-srv-0...
WA...  2024-04-09 16:34:10  runp02srv002eth0   SourceID[subsystem: Detector_Readout id: 104] Postprocessing has too much backlog, thread: 0 -- 7346224...    readoutlibs::P...  np02-srv-0...
WA...  2024-04-09 16:33:55  runp02srv002eth0   fragments_to_dataflow0: Unable to send within timeout period (timeout period was 0 milliseconds)               iomanager::Ti...   np02-srv-0...
ER...  2024-04-09 16:33:55  runp02srv002eth0   Unable to send within timeout period (timeout period was 0 milliseconds)                                       ipm::SendTim...    np02-srv-0...
ER...  2024-04-09 16:33:41  dataflow0          Unexpected Fragment for triggerID 146-0/24769, type 12, subsystem: Detector_Readout id: 101 -- 231 similar...   dfmodules::U...  np04-srv-0...
```

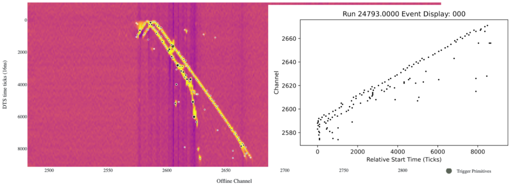
- We suspect three issues:
  - This run was before the TP ghost fix.
  - We had a look at the code – it's a mess, identified many areas for improvement. We suspect it's very slow.
  - The configuration: found potentially more reasonable configs, but with very different parameters.

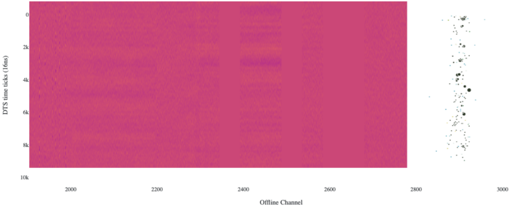
# Software triggering: HorizontalMuon

- Seemed to run without any errors or warnings, with the config I prepared using November data.
    - This is already AFTER the TP ghost fix.
- It did, however, generate some TRs with empty raw data fragments.
- This happened without any verbose warnings/errors...
- Did the TD request readout window that's already outside the wibframe buffer?

# Software triggering: ChannelDistance

- Algo from Alex, picks TPs based on their channel distance, given a time window. Ran for ∼2h, similar to HMA.

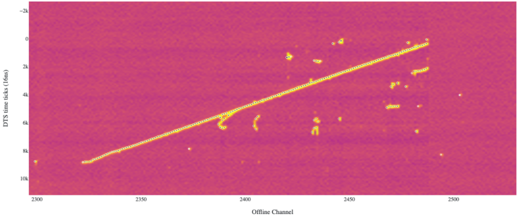


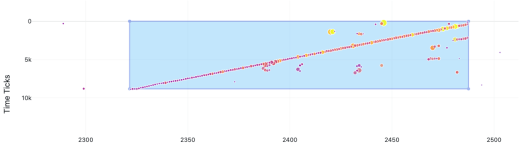


- Not many warnings, if any, but there were missing raw data fragments.
- Got worse when we tried to expand the readout window.
  - One one of the goals for TPG was to collect data with long tracks and readout windows for pedestals.
- Alex already identified a possible area for speed improvements in the code.

# Software triggering: ChannelAdjacency

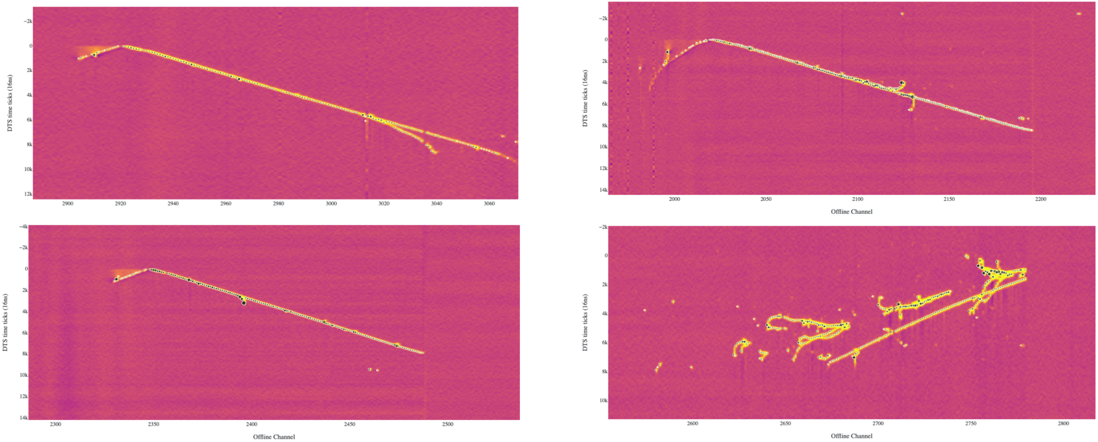
- Algo from Simranjit, picks TPs based on their channel adjacency, sliding window.
- Seemed to run out of the box. No warnings, no missing fragments.





- We collected data with long readout windows for offline TPG studies.
- One run lasted 3h36min without warnings with ∼5600 TRs.
- The only caveat is that we did see some missing fragments when running with Random HSI.

# Software triggering: ChannelAdjacency



# Summary

- We found stable TPG configurations and software triggering config.
    - A great future feature: configurable TPG algorithms on different planes.
- Collected a lot of data for the TPG and trigger group with ChannelAdjacency.
    - We had long runs with ChannelDistance too, but with some missing fragments.
- ChannelAdjacency currently our best bet for stable triggering with NP04.
- Not giving up on the rest:
    - Identified areas for speedup on ChannelAdjacency/Distance, DBSCAN.
    - Planning on more thorough latency measurements this/next month.
- All the runs: https://docs.google.com/spreadsheets/d/1RsVAnVUVL_QiMXzWflRapU55t1N6s-0CCMzpPSlhAv0/edit#gid=0
- Summary: https://docs.google.com/document/d/10472PbuAlTdmqjuUAWXHwkauDcq3tmXjW0luV-AtE1c/edit