Pre-clustering & charge ID

2024-08-23

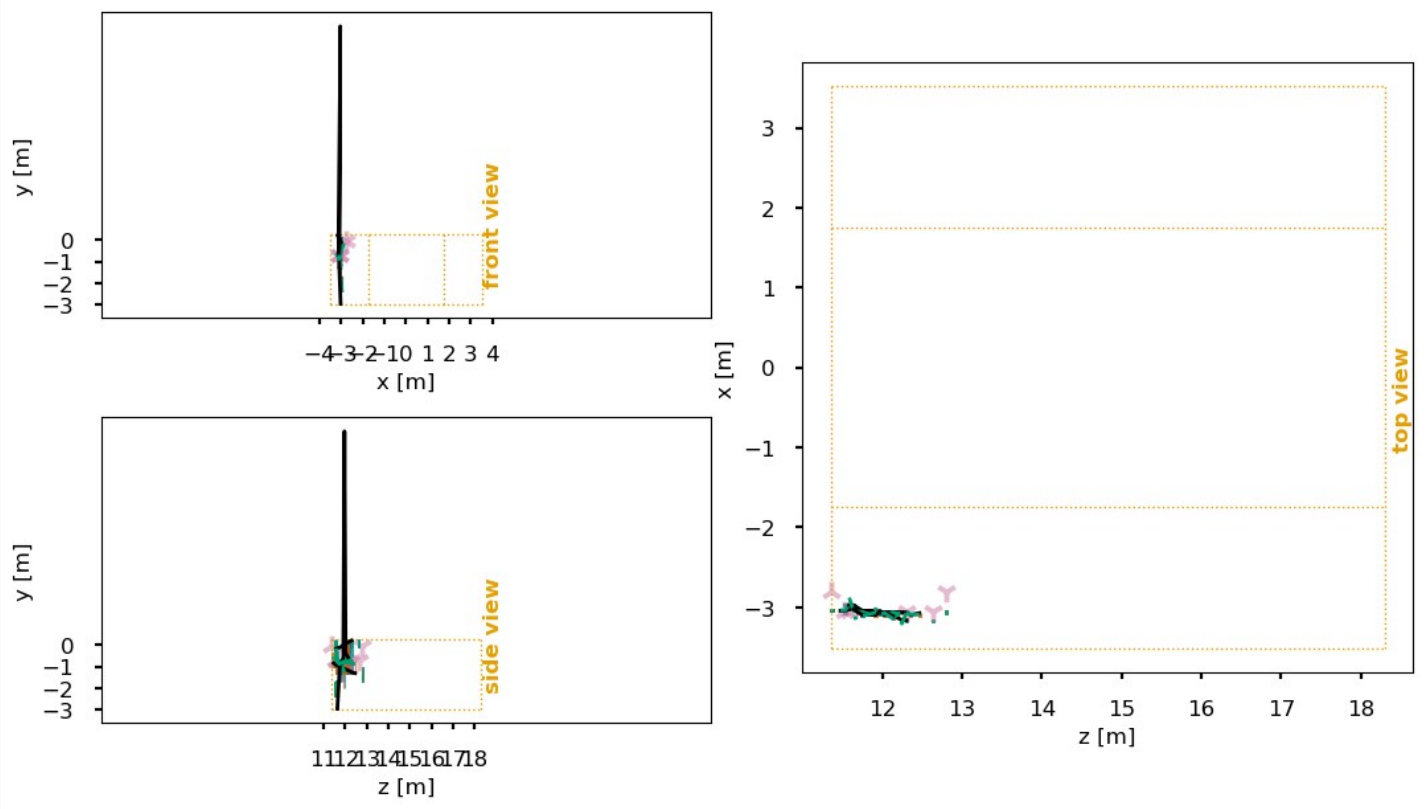
Asa Nehm

DEEP UNDERGROUND NEUTRINO EXPERIMENT

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

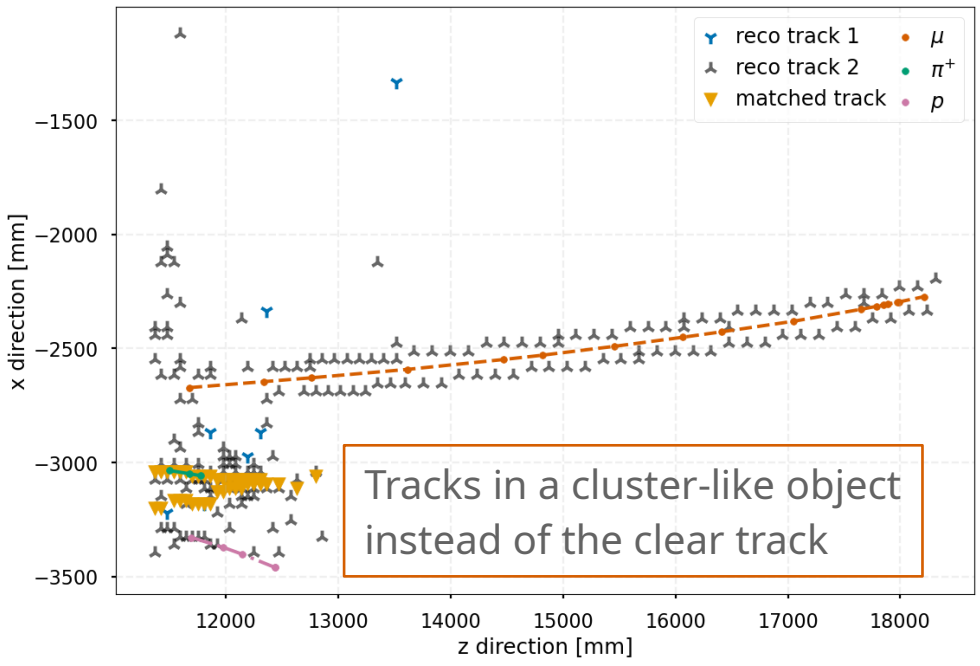
Problem event for Kalman filter

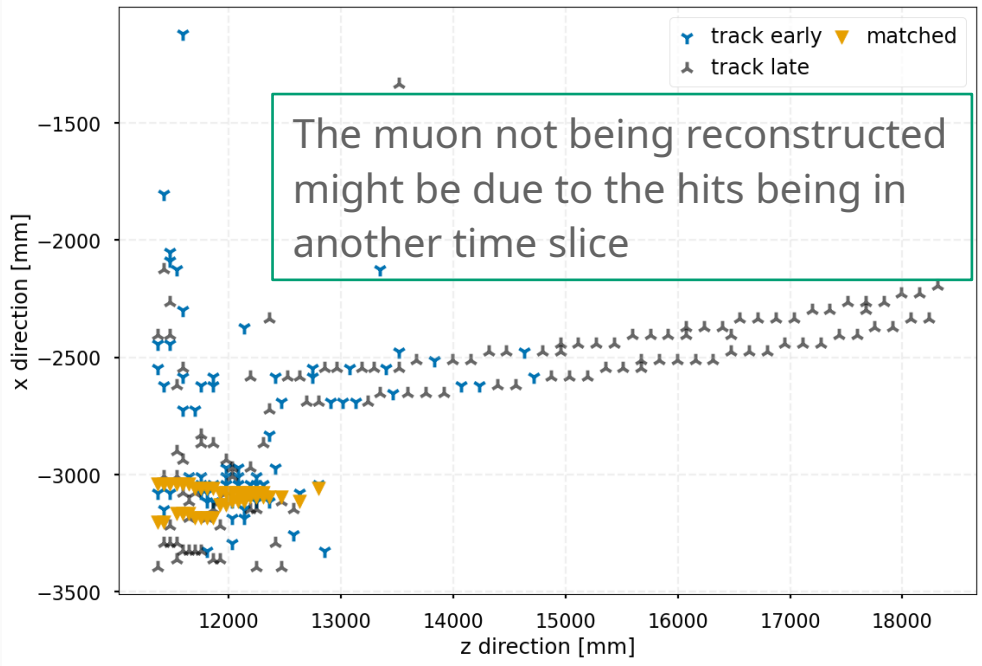
- Liam pointed out this event being problematic for the Kalman filter



Problem event for Kalman filter

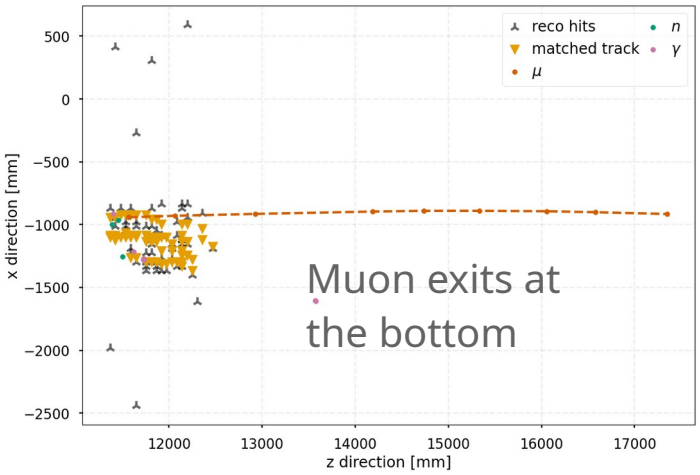
- Liam pointed out this event being problematic for the Kalman filter
- Looking into the actual hits for the reconstruction reveals this

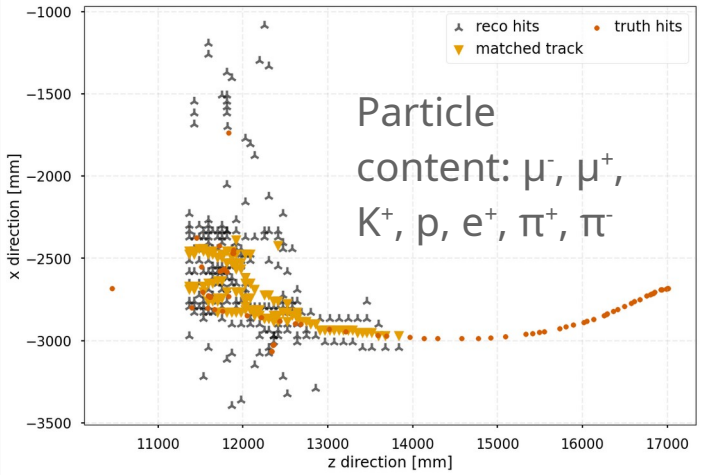




Is this a single occurrence?

- Looking through more events from the Kalman filter
- Categorizing in problems from the Kalman filter and from the reconstruction
  - Many of the reconstruction side have this (tracks in cluster-like structure) problem when Kalman filter result looks weird





What causes this?

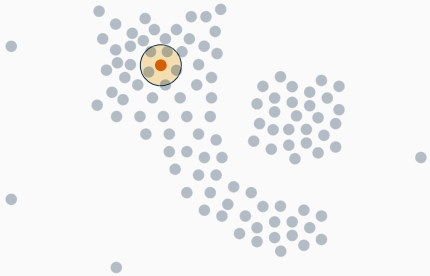
- Current way of reconstruction: track finding → cluster finding on rest
  - Finds (many) tracks in big clusters which ignores the cluster structure

- Alternative idea: pre-clustering → track finding → final clustering on rest
  - Track finding would extend to the hits in the pre-cluster as well if necessary
  - This would ensure that cluster-like objects (blobs) are not reconstructed as tracks, but keeps the existing reconstruction functioning

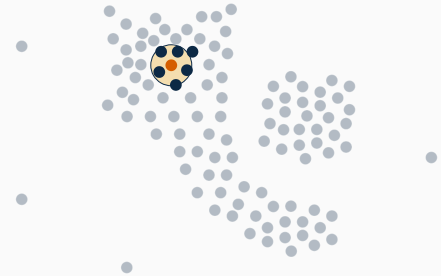
DB Scan clustering algorithm

Unclustered points

Choose random point as start



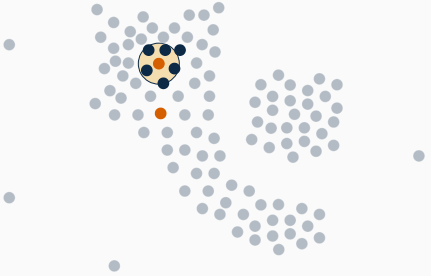
Look for other points within **certain distance**



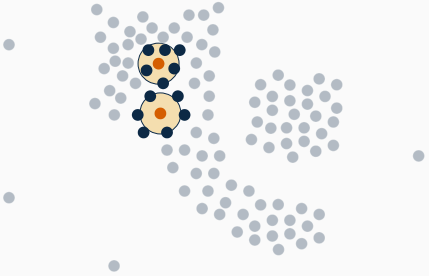
These close points are neighbours

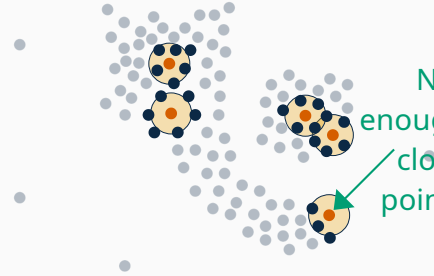
source

Repeat neighbour count for all points

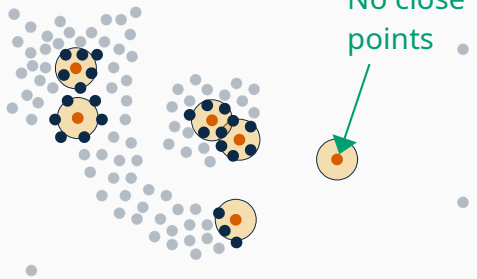


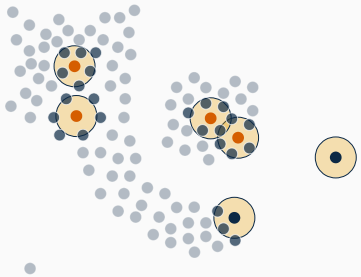
Define core points that have at least **certain amount of neighbours**

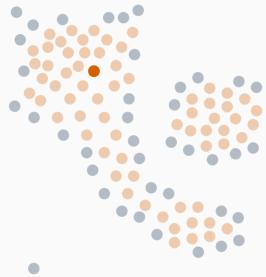




DB Scan clustering algorithm







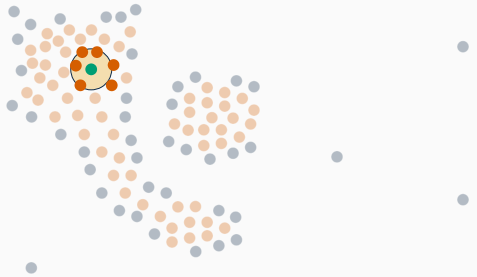
Only points with enough neighbours can become core points

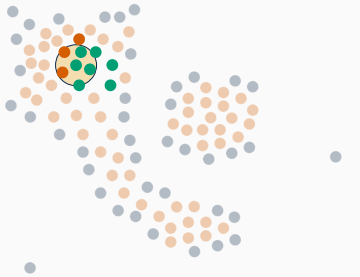
Identified all core points

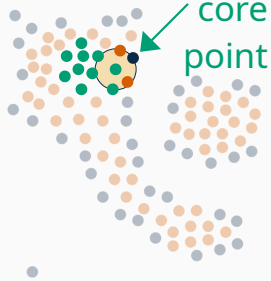
Pick random core point as start of cluster

source

Add all neighbouring core points and their neighbours to cluster until no more can be added



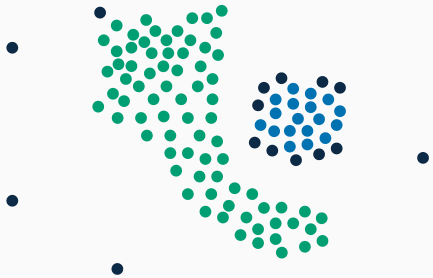




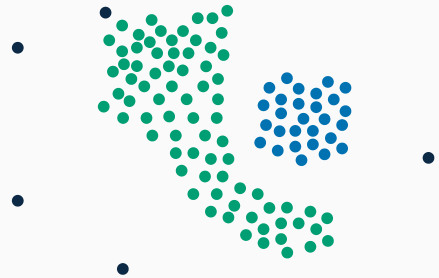
DB Scan clustering algorithm

Now add all neighbouring non-core points to cluster, but not their neighbours

Nothing close can be added anymore



Repeat for all remaining core points



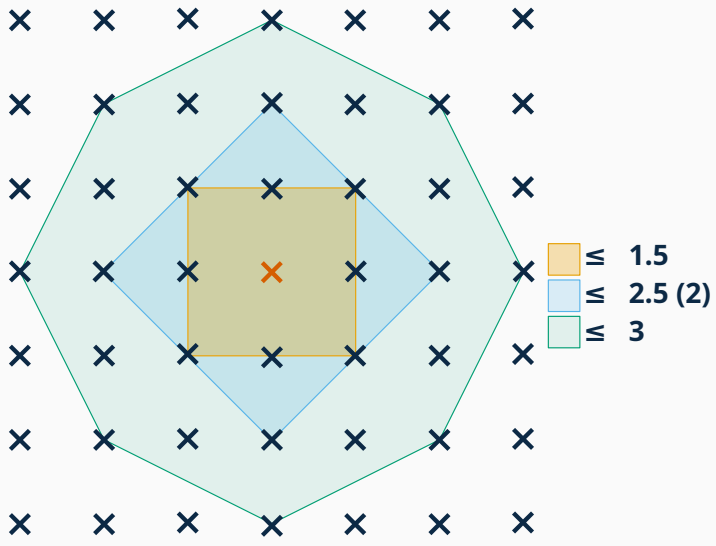
The fully identified clusters

source

Implementation

- Reconstruction has option to run DB scan before track finding already, just wasn't really working well → hijack this code
- Implement separate parameters for the pre-clustering to avoid clash with final clustering
- Original parameter values
  - Distance to neighbours: 2.5 (bar and plane number)
  - Neighbours for core points: 2
- Try out different parameter values for these

Neighbours	2	3	4	5	6	7	8	9	10
Coverage [%]	16.7	25	33.3	41.7	50	58.3	66.7	75	83.3

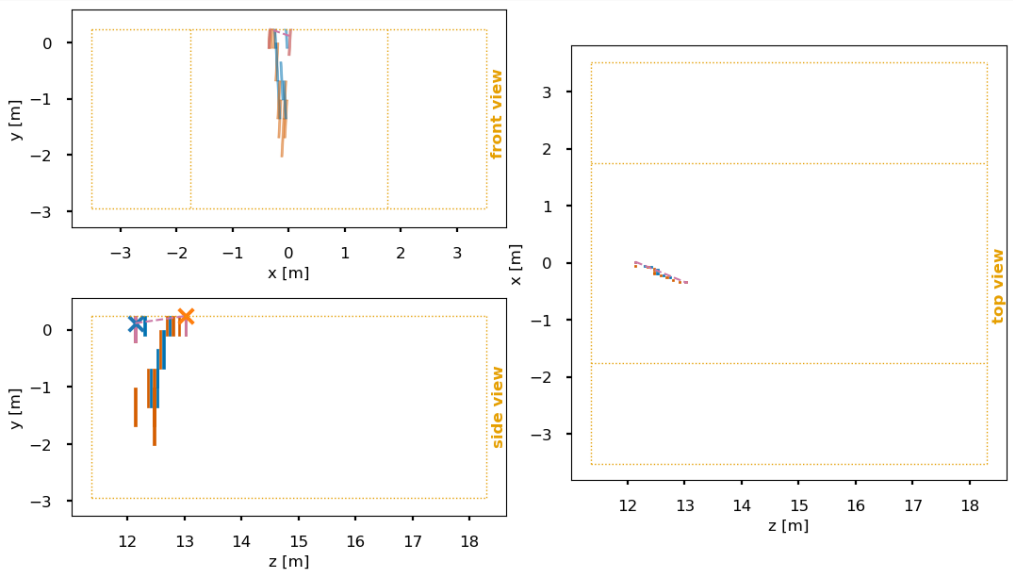


Implementation

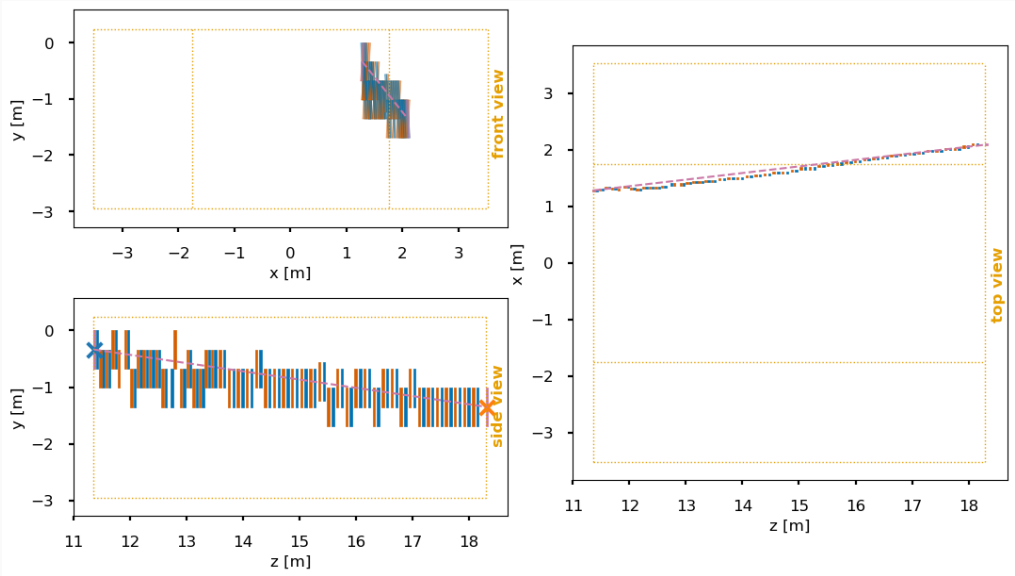
- Back-extension from tracks after pre-clustering into pre-clusters
  - Additional call to Extrapolation()
- Looking through results made some existing problems in pattern recognition obvious
  - Requirements for track matching were too tight (plane number) and too loose (time difference)
    - Tighten time difference to 16 ns
    - New mechanism: 3 out of 4 conditions need to be met for simple track matching
      - front and back, time and plane number
  - Extrapolation doesn't work for all tracks, some get 'randomly' reverted → add sorting call

Tuning of the 2 parameters

- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



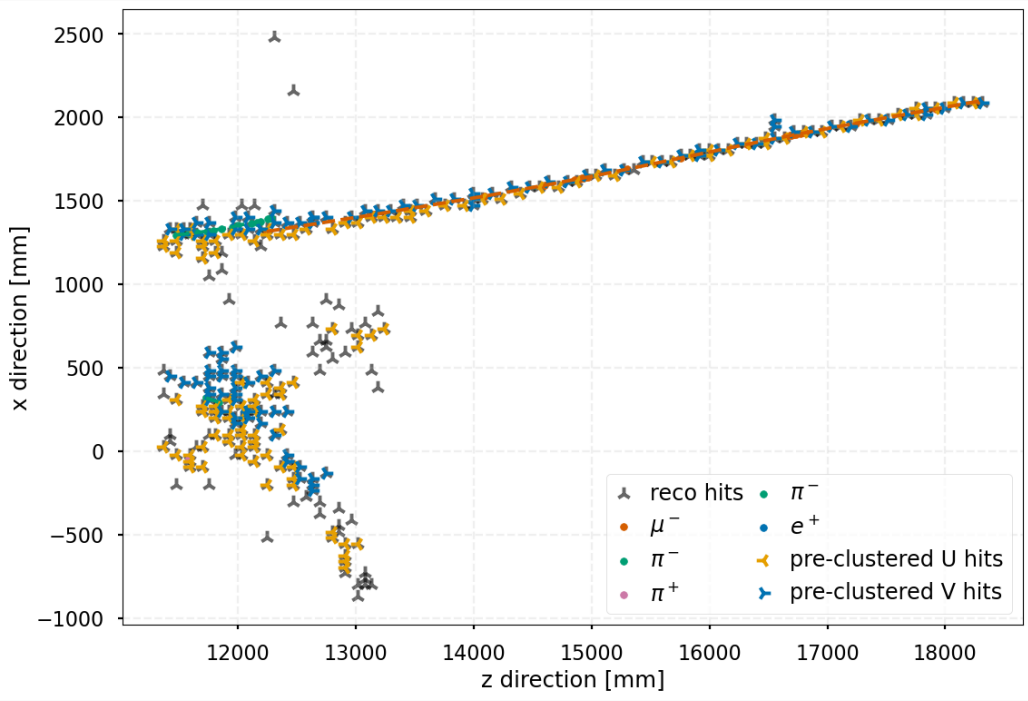
Original / status before any changes



Problems solved

Tuning of the 2 parameters

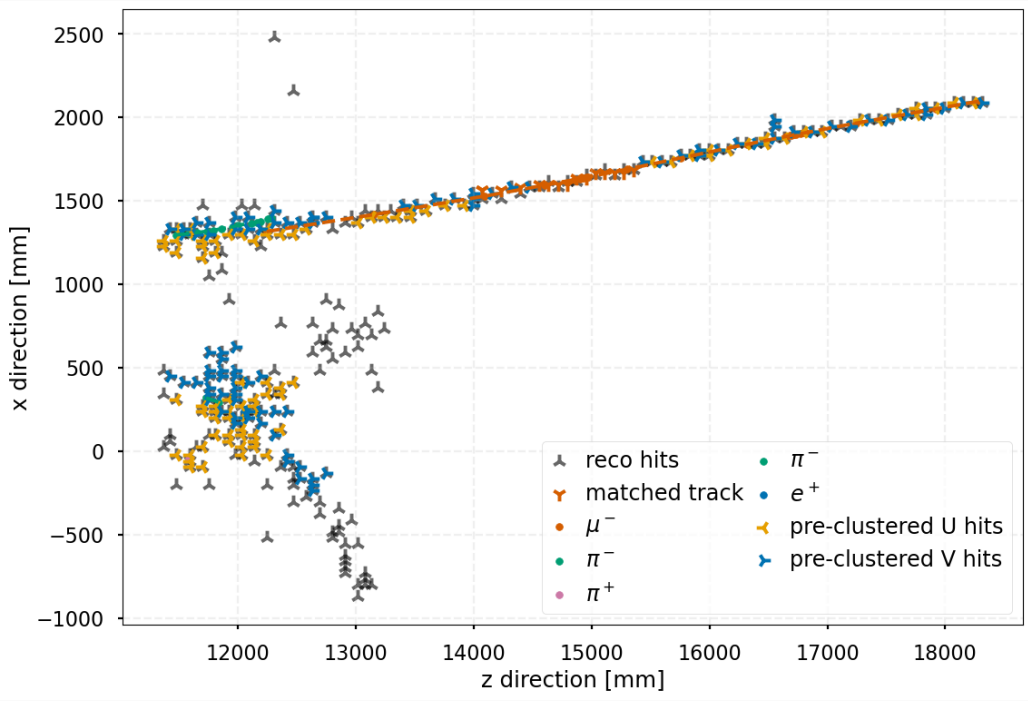
- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



5

Tuning of the 2 parameters

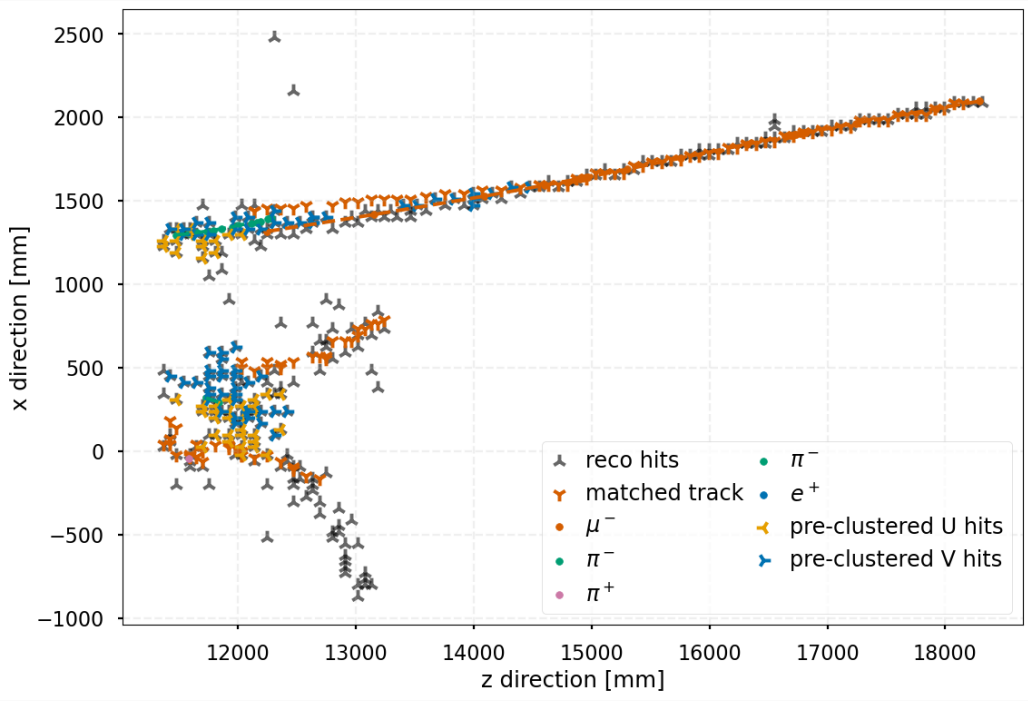
- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



6

Tuning of the 2 parameters

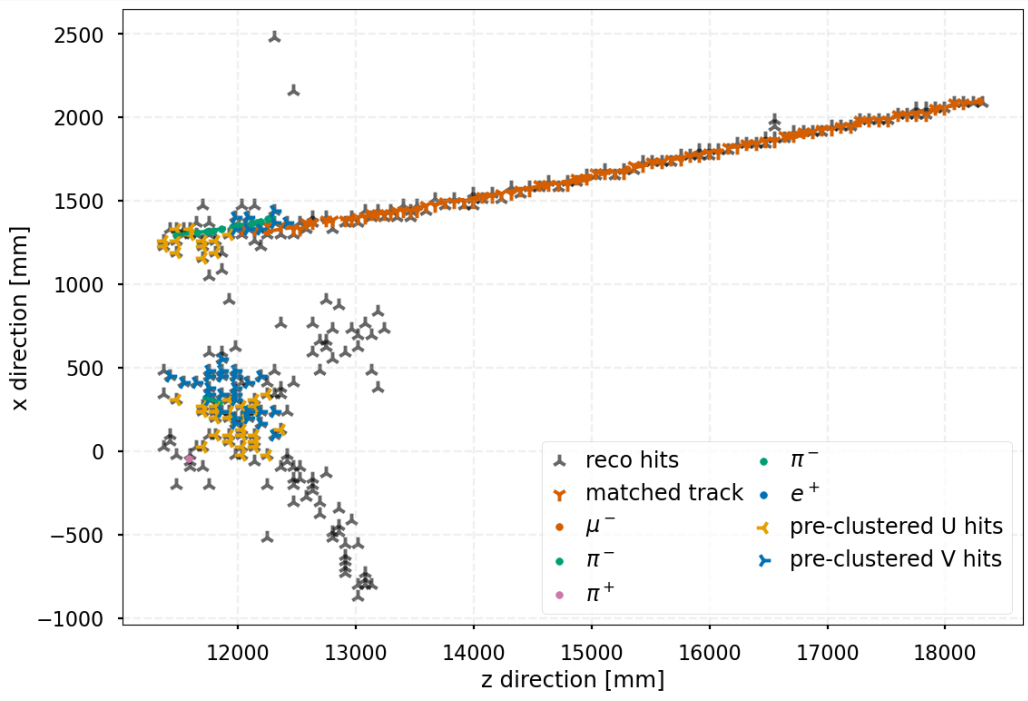
- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



7

Tuning of the 2 parameters

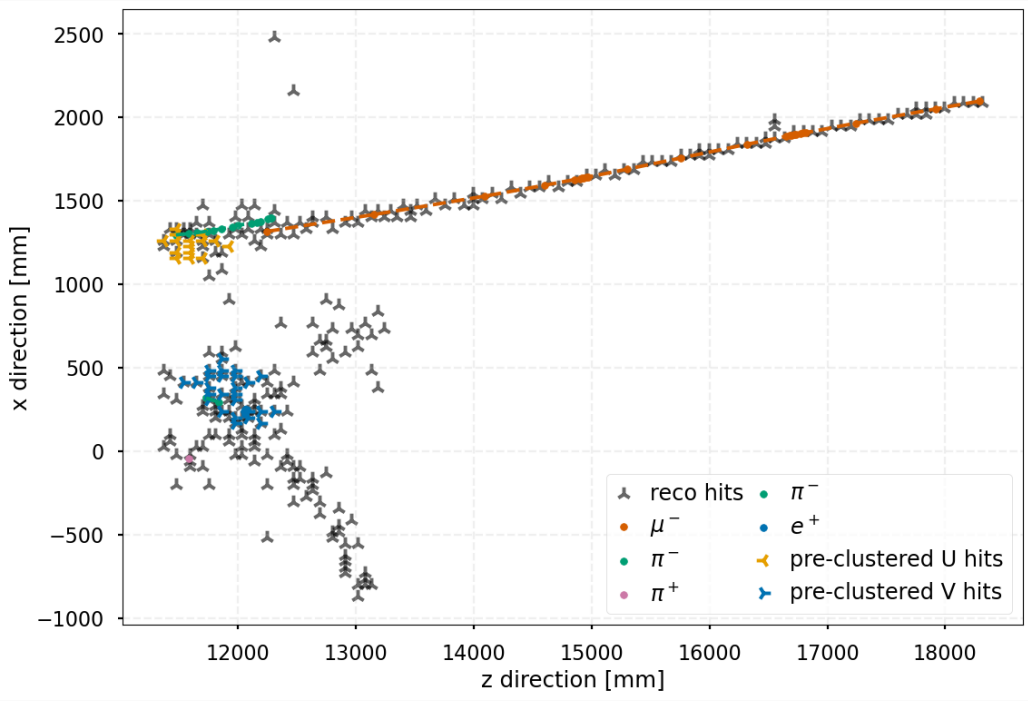
- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



8

Tuning of the 2 parameters

- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)



9

Tuning of the 2 parameters

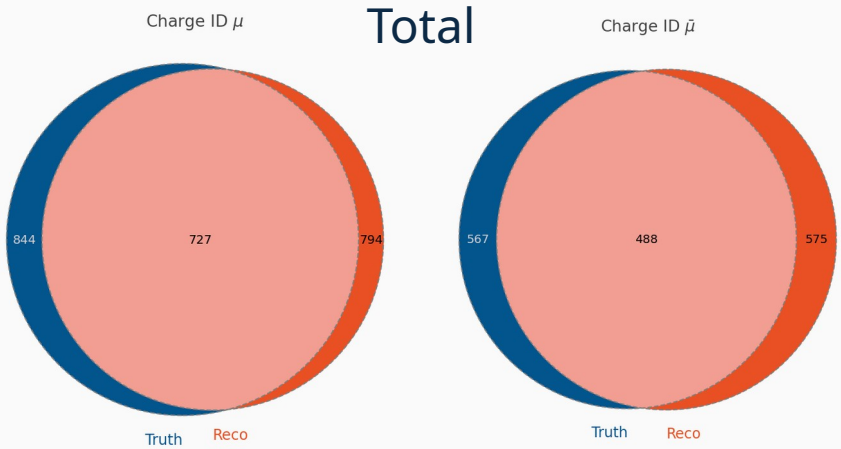
- Focus on number of neighbours for core points and keep distance at 2.5
- Antineutrino.0... file (NERSC)
- (For this event and small sample of events) the best value seems to be 7
- Compare for more events with not pre-clustered reconstruction (135)
  - There 7 doesn't seem good, and 9 much better

	7 (110)		8 (125)		9 (130)	
	Good	bad	Good	bad	Good	bad
new	–	–	–	–	1	–
changed	7 / 36	31 / 36	9 / 21	15 / 21	8 / 10	1 / 10
lost	2 / 25	23 / 25	3 / 10	7 / 10	–	5 / 6

Neighbours	2	3	4	5	6	7	8	9	10
Coverage [%]	16.7	25	33.3	41.7	50	58.3	66.7	75	83.3

Charge ID

- Implement charge identification code in the 3D track matching as well
  - This was determined outside of the reconstruction in a python script (→ Xiaoyan's xhuang6)
  - Adapted the script and added it into the general reconstruction code
    - Charge ID as output in TMS_Track
  - Created PR and validation plots to merge this

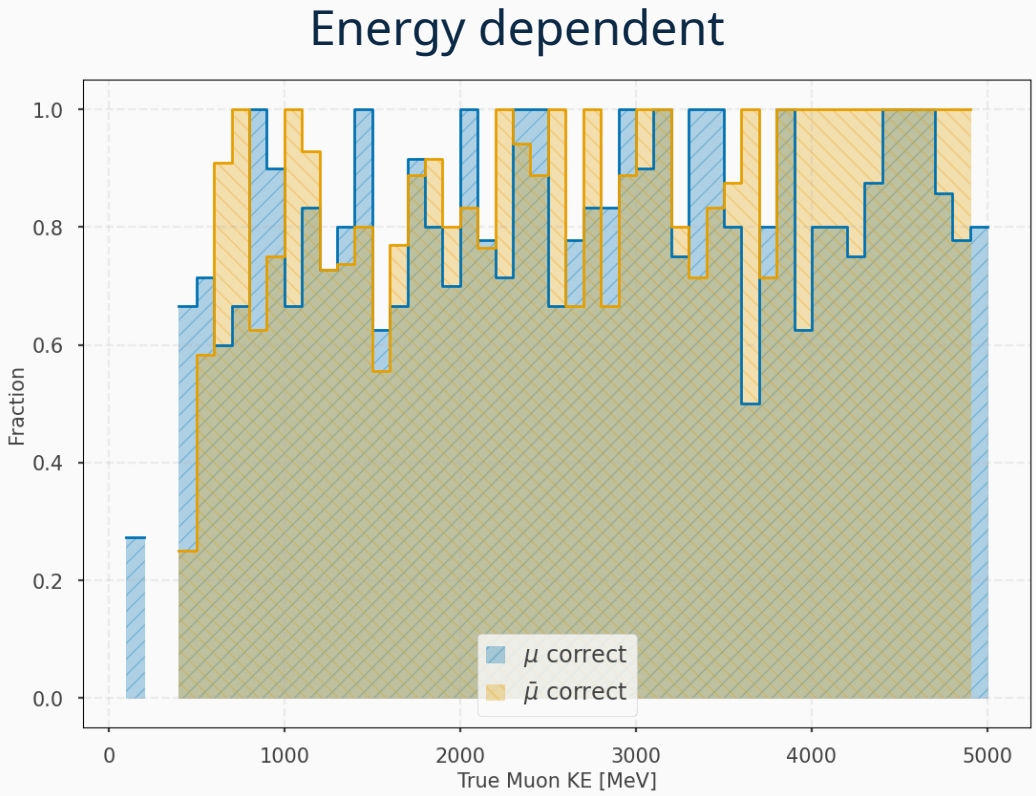


/pnfs/dune/persistent/users/abooth/Production/MiniProdN1p2-v1r1/run-spill-build/output/MiniProdN1p2_NDLAr_1E19_RHC.spill/EDEPSIM_SPILLS/00000/MiniProdN1p2_NDLAr_1E19_RHC.spill.00000.EDEPSIM_SPILLS.root

```
N entries: 128602
Too many tracks in event. Limit to first 5
Too many tracks in event. Limit to first 5
#events reconstruction:  1905 # events truth:  1905
Charge ID numbers
  Not identified:   42
  True muons:       727
  True antimuons:   488
  False muons:      67
  False antimuons:  129  here the not identified go into!
  Muons (efficiency | purity):      0.893  |  0.916
  Antimuons (efficiency | purity):  0.879  |  0.849
  Accuracy (both):  0.888
```

Charge ID

- Implement charge identification code in the 3D track matching as well
  - This was determined outside of the reconstruction in a python script (→ Xiaoyan's xhuang6)
  - Adapted the script and added it into the general reconstruction code
    - Charge ID as output in TMS_Track
  - Created PR and validation plots to merge this
  - No cut on start in ND-LAr and end in TMS!!!



Outlook

- Add pre-clustering as option in main branch of reconstruction (PR)
- Need to evaluate the efficiency of the reconstruction with and without the pre-clustering

Backup