Mx2 + 2x2 Combined ML

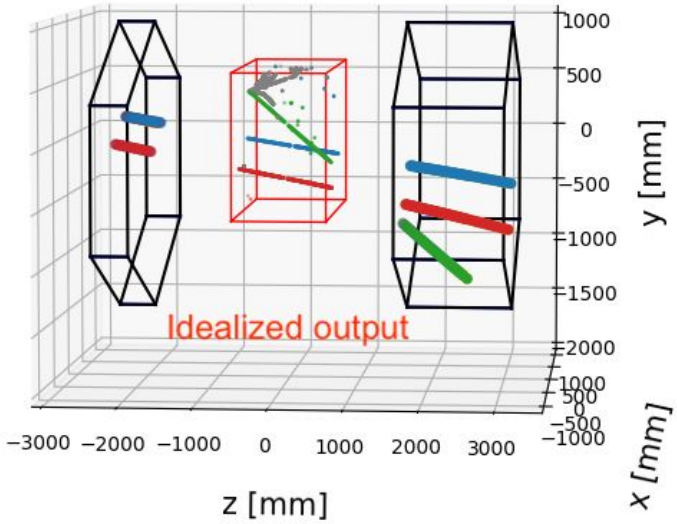
Nate Santiago (MIT Summer Student) and Jessie Micallef

Diving into Mx2 + 2x2 Simulation

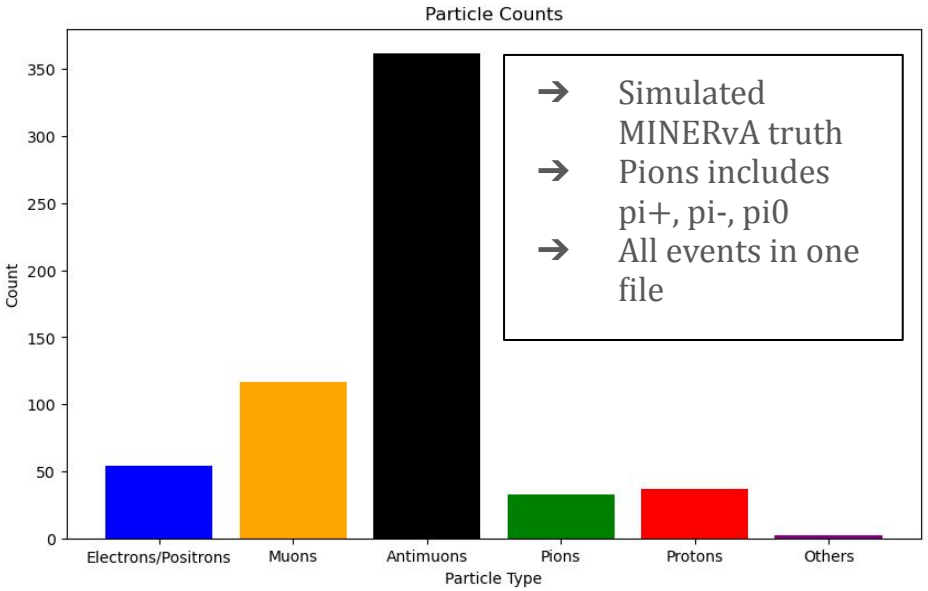
- One simulation output file for Mx2 dst & 2x2 Flow
- Goal: identify common track IDs & parent PIDs from shared simulation

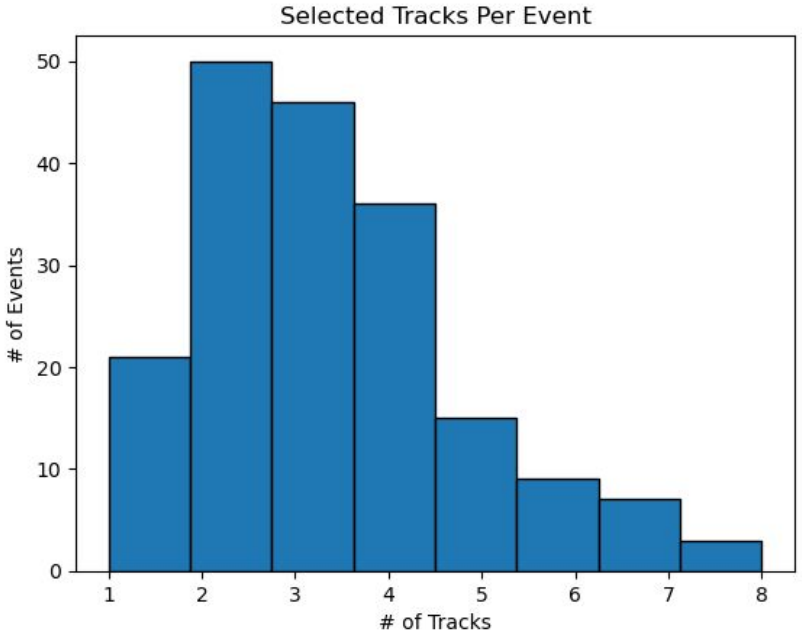
- Explore:
  - Potential for MINERvA Track reconstruction to be used as input for GrapPA
  - Feasibility of track matching between Mx2 & 2x2 using GNN

- Future goal:
  - Integrate shower matching between Mx2 & 2x2



Diving into Mx2 Truth

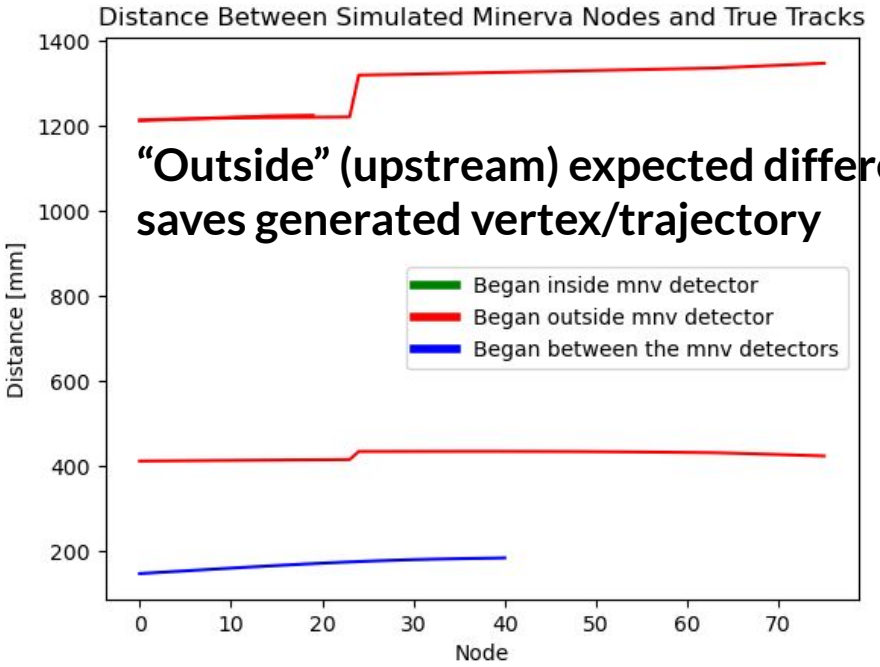




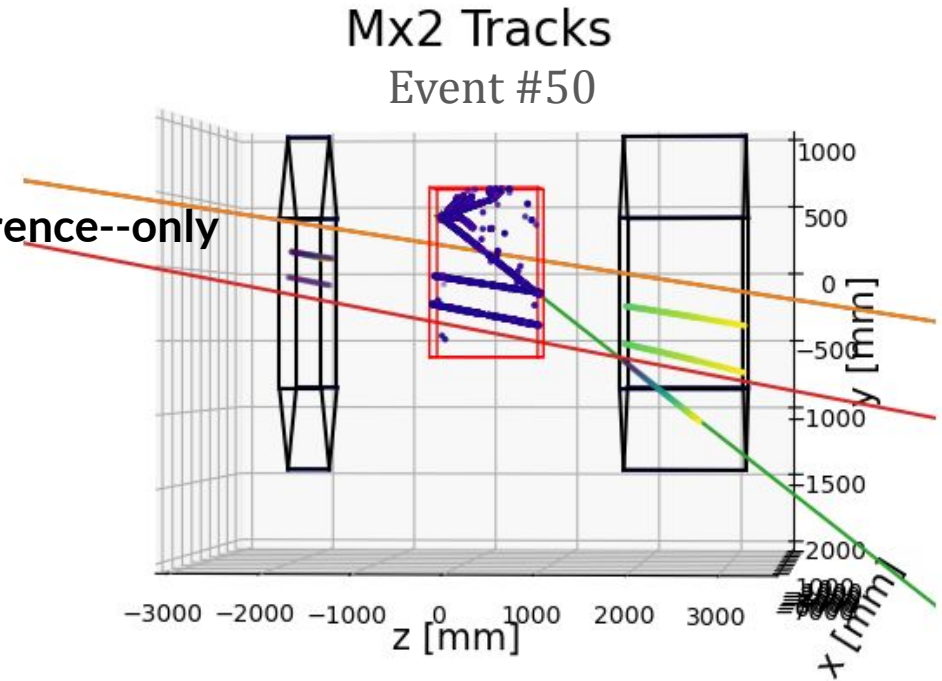
In ONE File: 192 Entries → 605 Trajectories

- 381 Begin Outside MINERvA Detectors
- 141 Begin Inside MINERvA Detectors
- 83 Begin Between MINERvA Detectors

Mx2 Truth vs. Reco



“Outside” (upstream) expected difference--only saves generated vertex/trajectory



In Total: 192 Entries → 605 Trajectories

- 381 Begin Outside MINERvA Detectors
- 141 Begin Inside MINERvA Detectors
- 83 Begin Between MINERvA Detectors

Mx2 Truth vs. Reco Inside/Between

Tracks: The first algorithm examines all possible combinations of three candidates in which no two candidates share the same plane orientation. Such a combination of candidates are formed into a 3D-track if they overlap. All 3D-tracks that are found are fit by a Kalman filter fit routine.

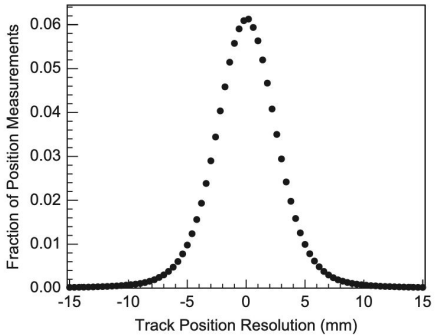
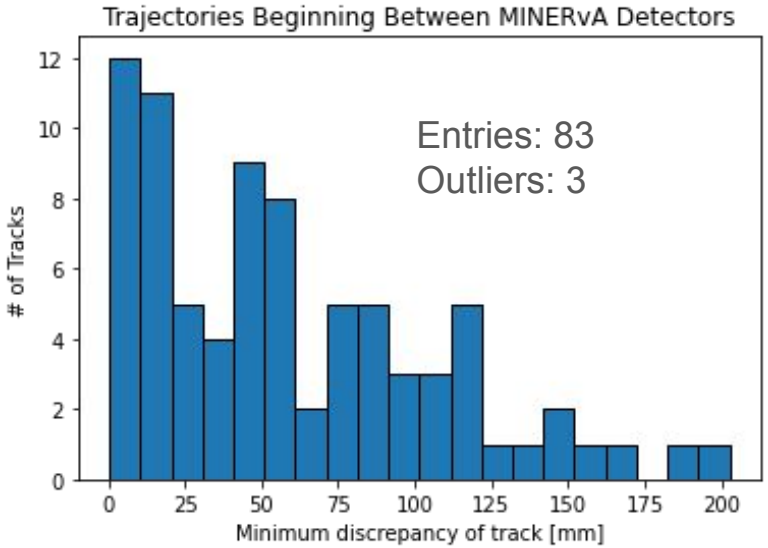
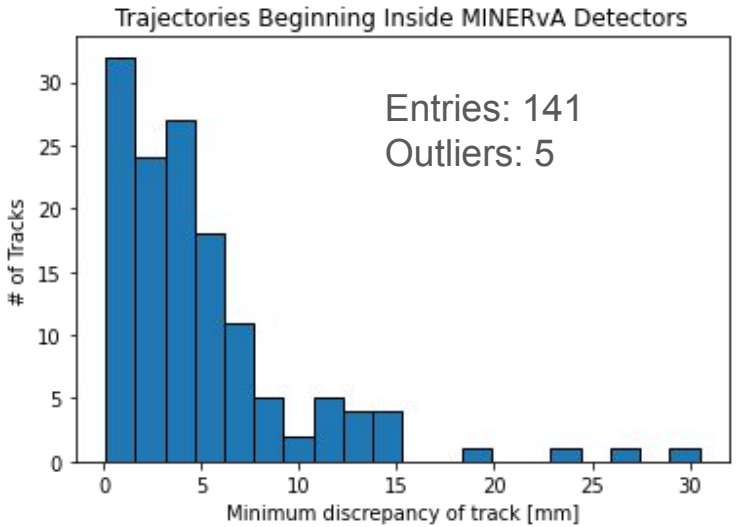


Fig. 34. Resolution of the fitted positions along a track relative to the measured cluster positions for a sample of data rock muons. The RMS of the distribution is 3.1 mm.

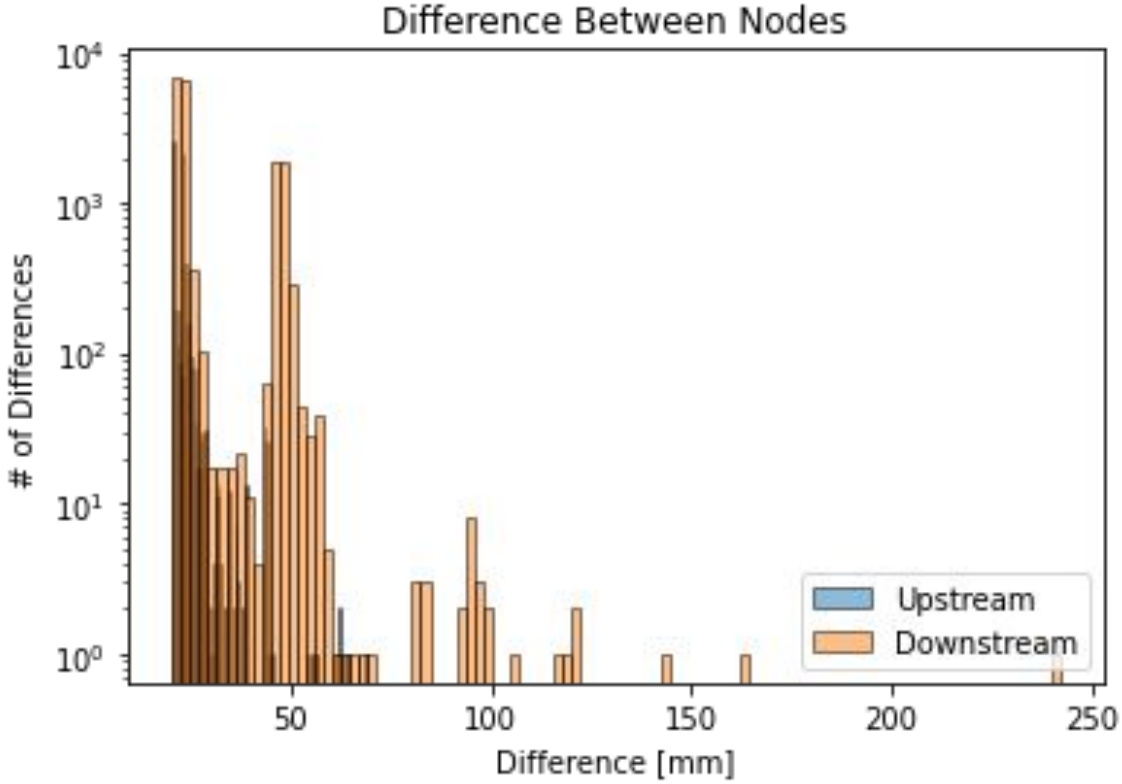




Compare: MINERvA True Trajectory vs. track nodes. Find closest node to track

Backup has 381/605 tracks beginning upstream MINERvA detectors

How detailed are the Mx2 Reconstructed Tracks?



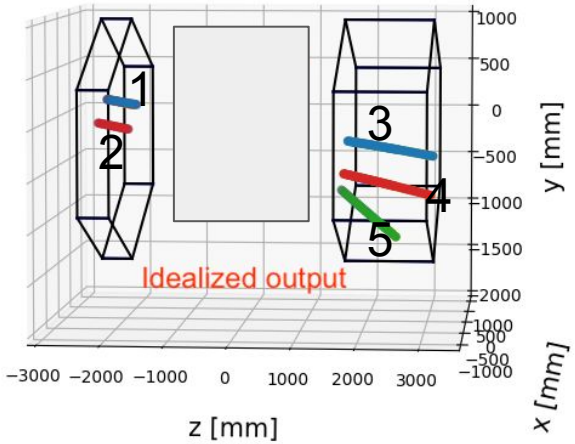
Resolution between reconstructed track nodes.

Upstream & downstream separated

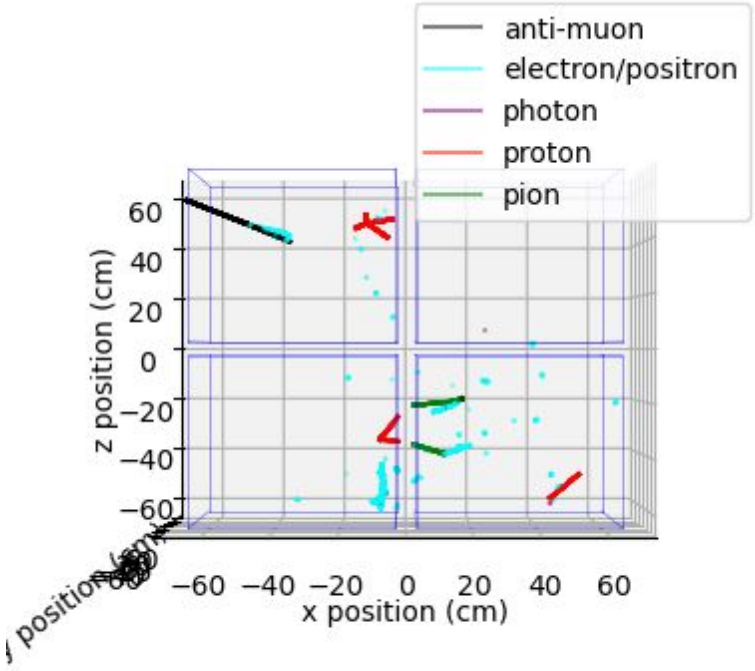
What causes 2 peak structure?

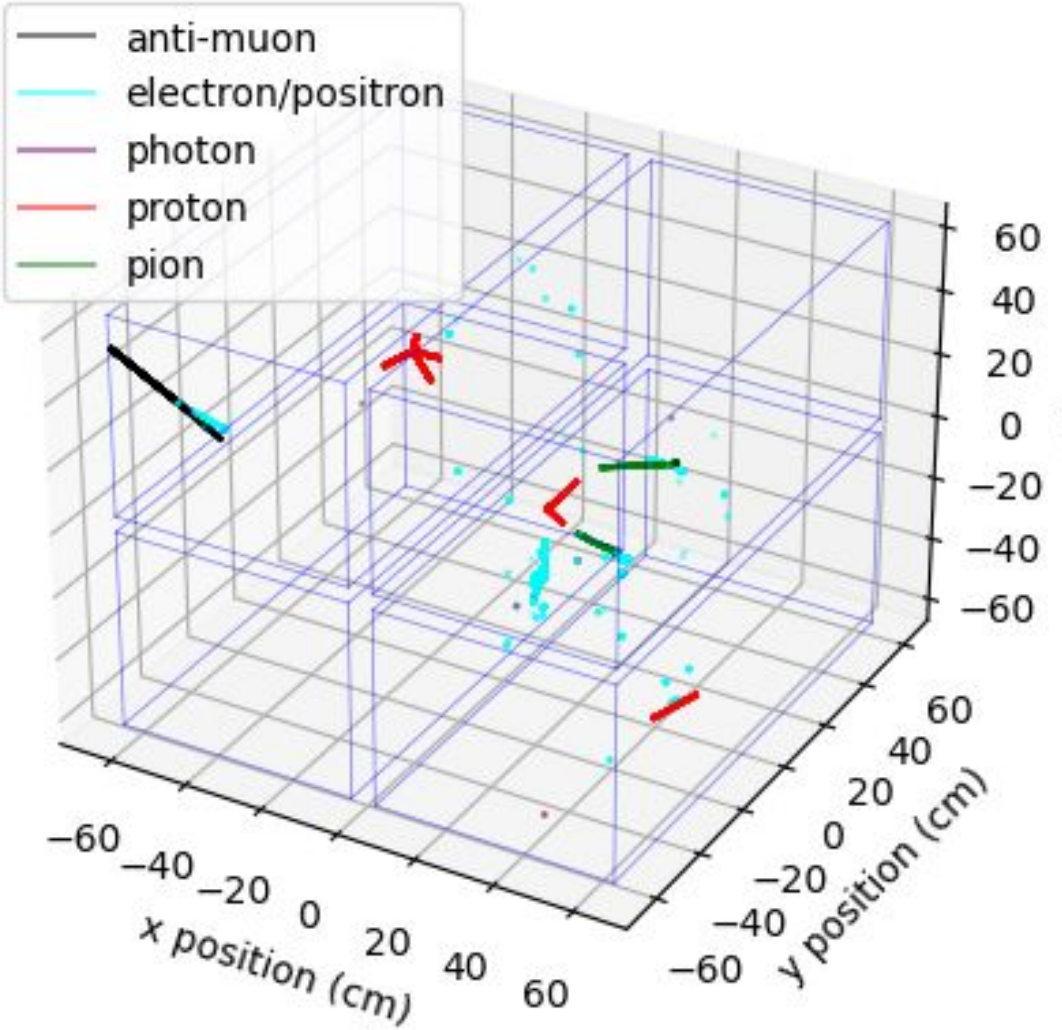
Open Questions Mx2

- MINERvA “charge” information per track node is stored as q/P
  - What is P?
  - q/P is quite small/repetitive for each track node
- Does MINERvA store unique identification for reconstructed track PER entry
  - Are the upstream & downstream separately reconstructed?
  - Would like to call each reco-ed track as one “cluster”



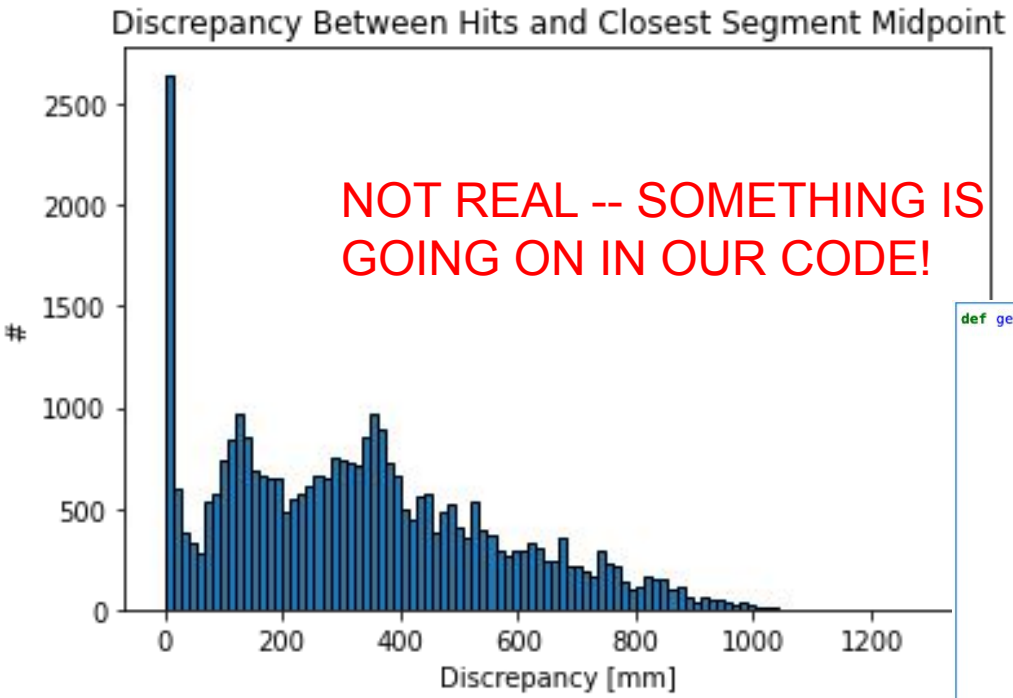
Diving into 2x2 Truth





Event #34

How detailed are the Mx2 Reconstructed Tracks?



Difficulty lining up hits and segments (see next slide)

? - 104 entries missing that have empty backtracked hits?

```
def get_truth_2x2(NdFlow,entry):

        NdFlow.event_hit_indices[entry]

        #We then get the start and stop indices for the hits corresponding to the event
        hit_start_index = NdFlow.event_hit_indices.fields("start")[entry]
        hit_stop_index = NdFlow.event_hit_indices.fields("stop")[entry]

        #Get the backtracked hits using the indices
        backtracked_hits = NdFlow.backtracked_hits.fields("segment_id")[hit_start_index:hit_sto
        backtracked_hits_frac = NdFlow.backtracked_hits.fields("fraction")[hit_start_index:hit_

        # Get the segment ids with fraction != 0, because only
        #non-0 fractions contribute to the event
        segment_ids = np.unique(backtracked_hits[backtracked_hits_frac != 0])

        #Get the segments by putting a condition on the segment ids we
        #retrieved above and you can see that all of them have the same 'event_id'
        #(This is the mc truth event id, not the id in charge events)
        segments = NdFlow.segments[segment_ids]

        #Now get the trajectories using the event id above
        trajectories = NdFlow.true_traj[NdFlow.true_traj.fields("event_id") == segments['event_
        print("TRUE TRAJ LEN", len(trajectories['vertex_id']))
        return trajectories['vertex_id'], segment_ids
```

Open Questions 2x2

- Hits['charge/calib_final_hits/data'] vs hits_backtrack['mc_truth/calib_final_hit_backtrack/data']
  - Backtrack gives associated segment & true trajectory information
  - Hits is what we will have

[40]:
```
class NdData:
    def __init__(self, filename):
        self.flow_file = h5.File(filename, 'r')
        self.data = self.flow_file['charge']['calib_final_hits']['data'] #prompt or final
        self.match_region = self.flow_file['charge']['events']['ref']['charge']['calib_final_hits']['ref_region']
        self.backtracked_hits = self.flow_file['mc_truth']['calib_final_hit_backtrack']['data'] #prompt
```

- hits_backtrack.fraction is an array with 200 entries and mostly 0s
- hits.Q has single entry

→ are these not the same?

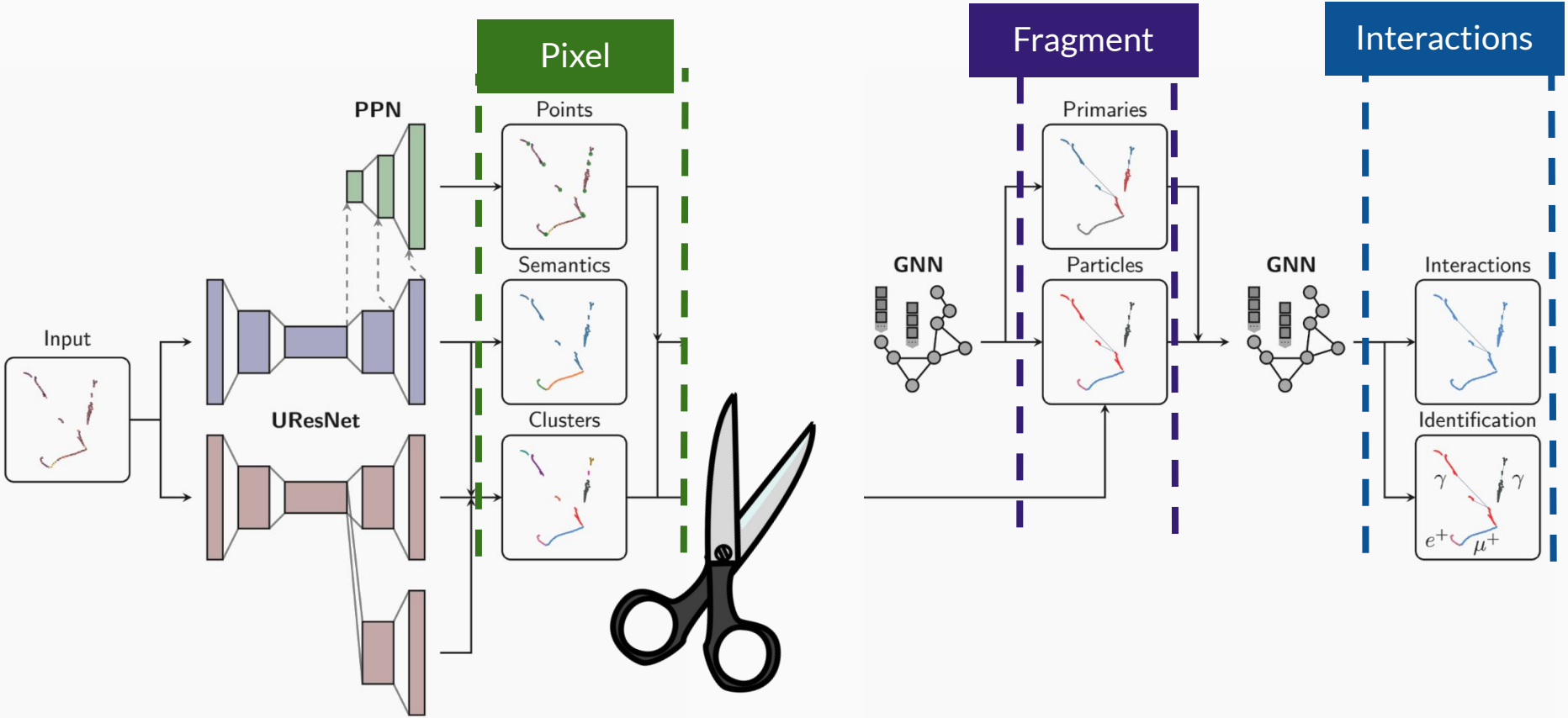
*[41]:
```
def hit_trackdown(NdFlow,event_number):
    hit_ref_slice = NdFlow.match_region[event_number]
    hits = NdFlow.data[hit_ref_slice[0]:hit_ref_slice[1]]
    hits_bt = NdFlow.backtracked_hits[hit_ref_slice[0]:hit_ref_slice[1]]
    segments = NdFlow.segments

    for hit in hits_bt:
        for cont in range(len(hit['fraction'])):
            if abs(hit['fraction'][cont]) > 0.0001:
                seg_id = hit['segment_id'][cont]
                seg = segments[seg_id]
                traj_evtid = segments["traj_id"]
                print("segment: ", seg['x_start'], seg['x'], seg['x_end'])
                print("hit: ", hits[cont]['x'])

hit_trackdown(NdFlow,i)
```

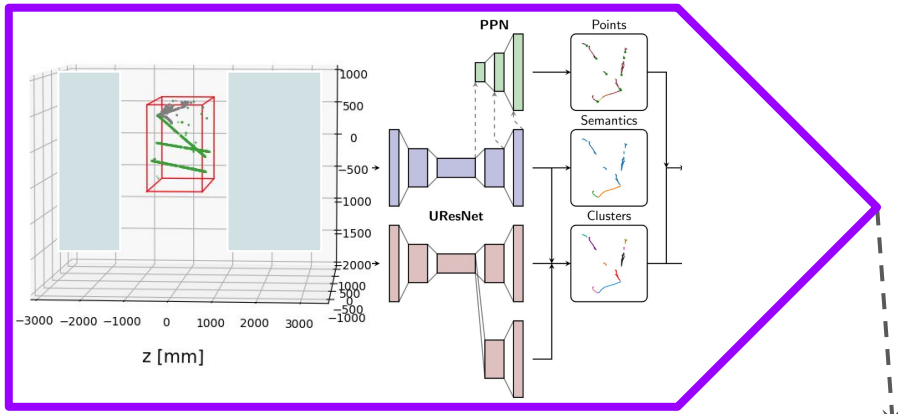
```
segment:  19.865473 19.880016 19.894558
hit:  20.32665133780989
segment:  19.894558 19.868067 19.841576
hit:  20.27875776334526
segment:  19.865473 19.880016 19.894558
hit:  20.32665133780989
segment:  19.894558 19.868067 19.841576
```

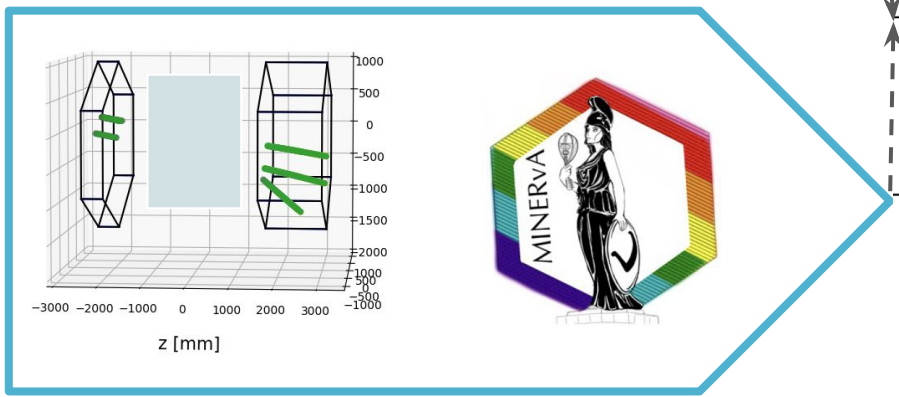
SPINE + Muon Detector

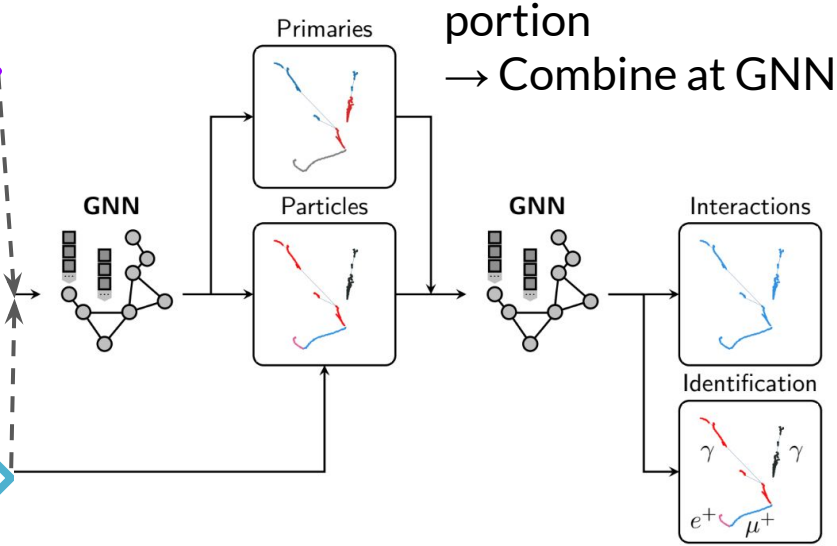


SPINE: Adding MINERvA

Idea for direction:
→ Keep SPINE structure the same
→ Add a new pixel-level "branch" for MINERvA portion
→ Combine at GNN

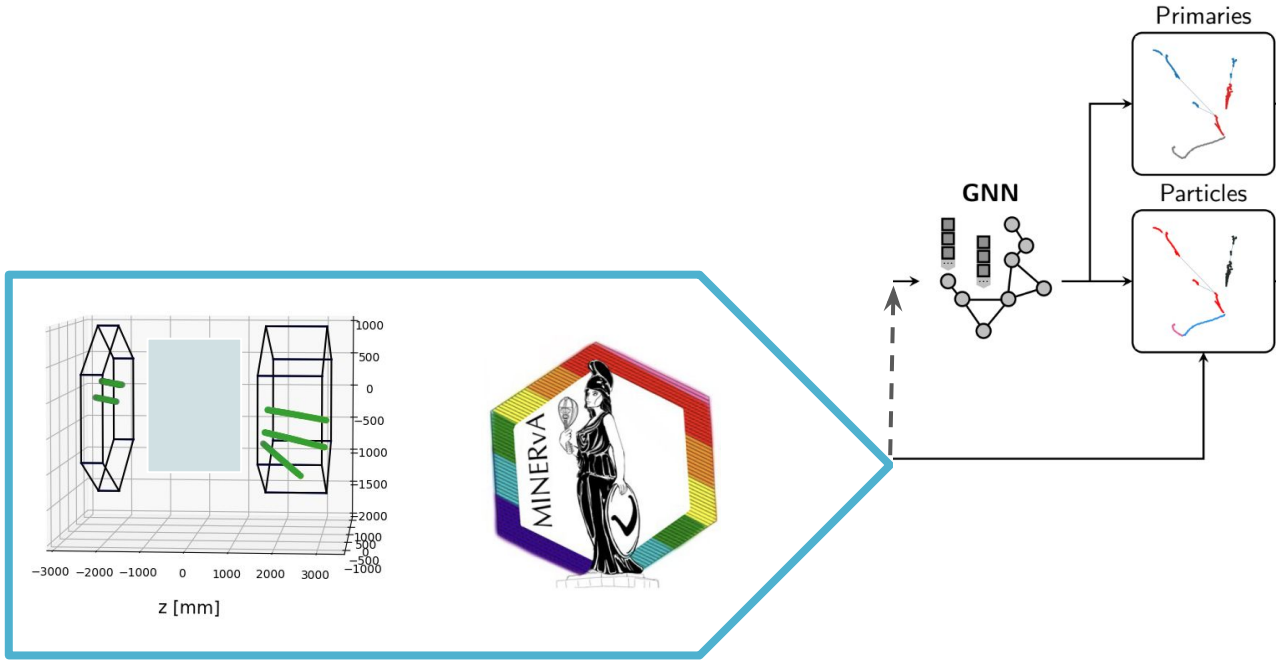






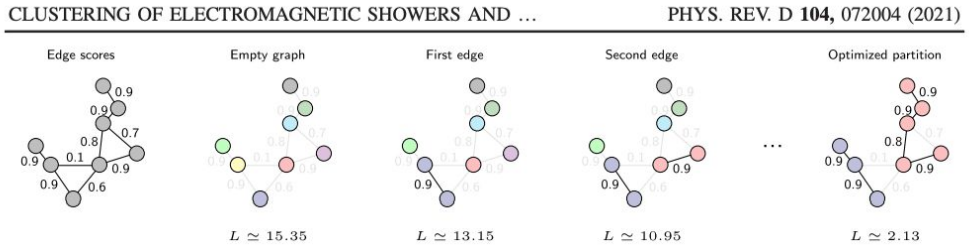
SPINE: Adding MINERvA

Idea for direction:
→ Keep SPINE structure the same
→ Add a new pixel-level "branch" for MINERvA portion
→ Combine at GNN



Next Steps

- Insert just MINERvA Up+Downstream into GrapPA



- Do we need to calculate these geometric features ourselves or is it in SPINE repo?

- Resolve indexing issues
  - Between 2x2 truth and charge data
  - Between 2x2 and Mx2

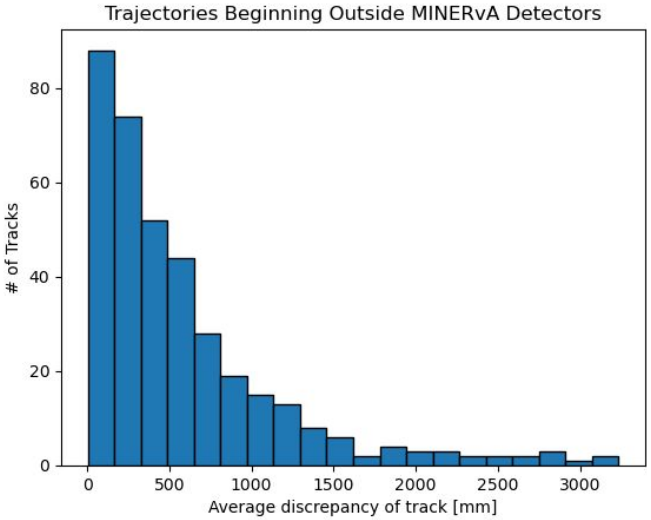
Geometric features are a list of summary statistics of the distribution of fragment voxels in Euclidean space. It includes the following 22 features:

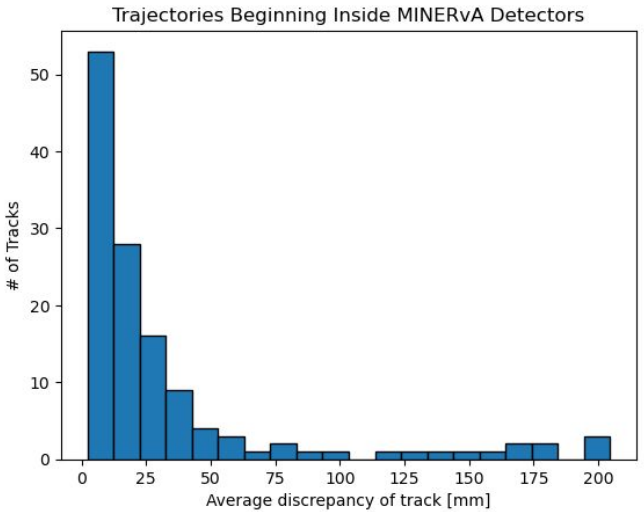
(i) normalized covariance matrix (9 features);
(ii) normalized principal axis (3 features);
(iii) centroid (3 features);
(iv) number of voxels (1 feature);
(v) initial point (3 features);
(vi) normalized initial direction (3 features).

Backup

Outlier track distances [mm] not shown in the histogram (out):

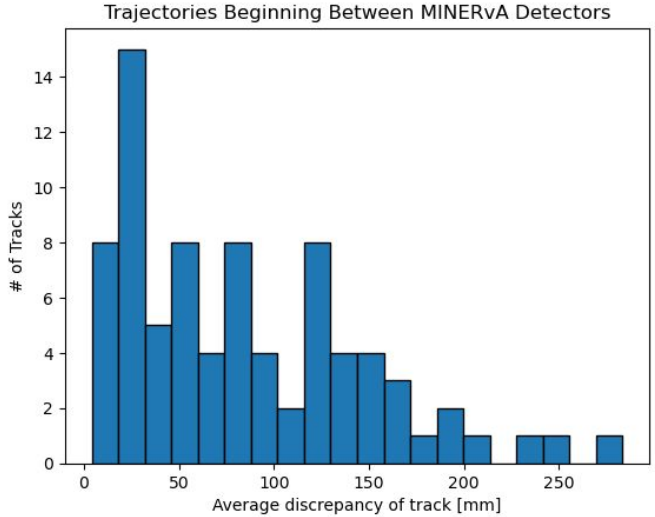
1. 4027.941 94238
2. 13053.00 520443
3. 4063.440 77296
4. 6536.354 77547
5. 4500.534 86683
6. 5197.468 74855
7. 4725.637 48528
8. 4906.706 67921
9. 4335.718 98159
10. 4192.697 55047





Outlier track distances [mm] not shown in the histogram (between):

1. 967.99903437
2. 339.18152325
3. 2463.7028581
4. 277.28518761
5. 906.02641152
6. 1154.3405612
7. 262.7109147
8. 266.50595075
9. 285.9578416
10. 1082.20153785
11. 614.49998759

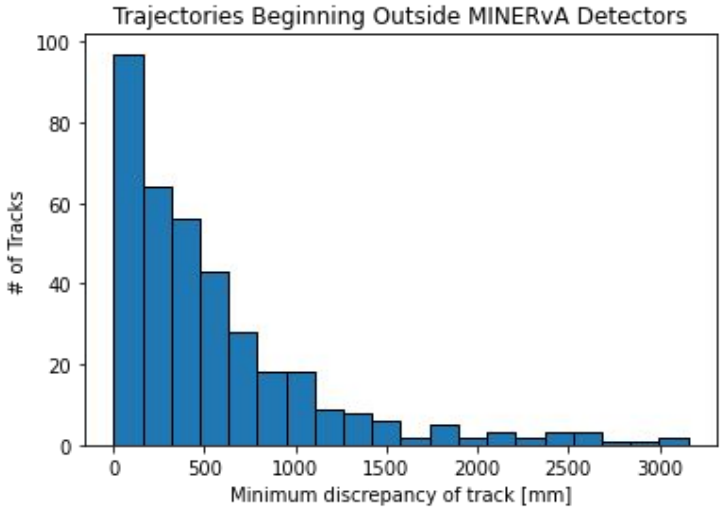


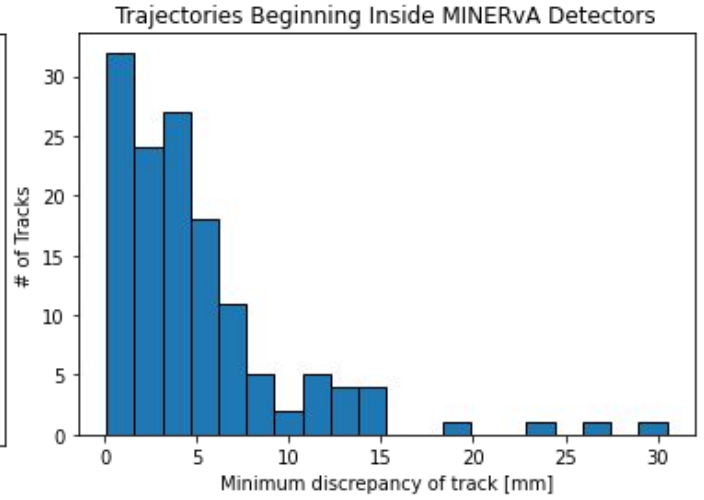
Outlier track distances [mm] not shown in the histogram (between):

1. 471.5258871
2. 594.4696957
3. 713.83769277

Outlier track distances [mm] not shown in the histogram (out):

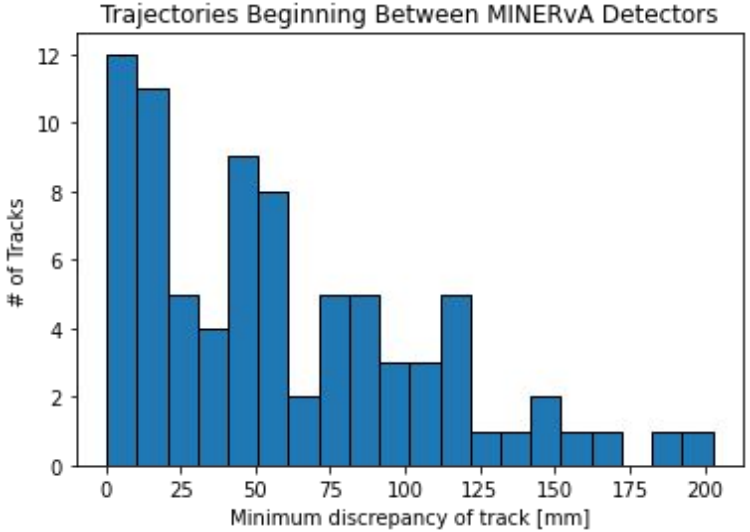
1. 4000.444 6819
2. 12996.53 684748
3. 4061.688 1255
4. 6240.834 16229
5. 4483.841 91624
6. 4974.562 18886
7. 4621.536 28791
8. 4905.332 4456
9. 4273.779 45061
10. 3862.628 517





Outlier track distances [mm] not shown in the histogram (in):

1. 119.1060 0575
2. 53.03724 349
3. 56.06059 828
4. 46.04747 848
5. 41.53936 816

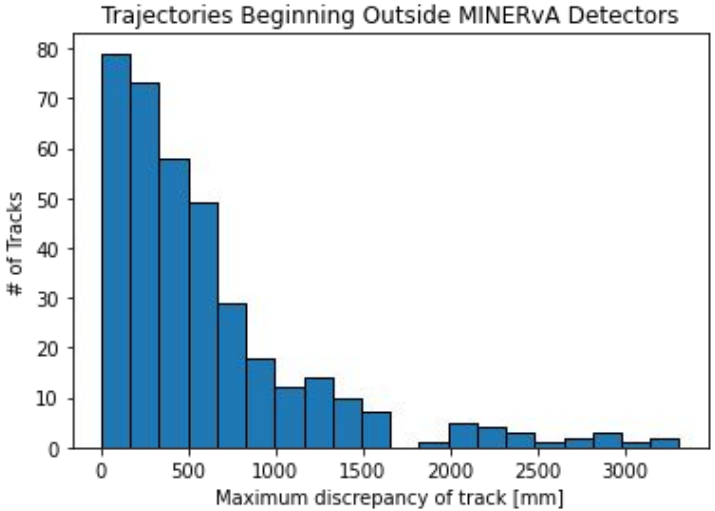


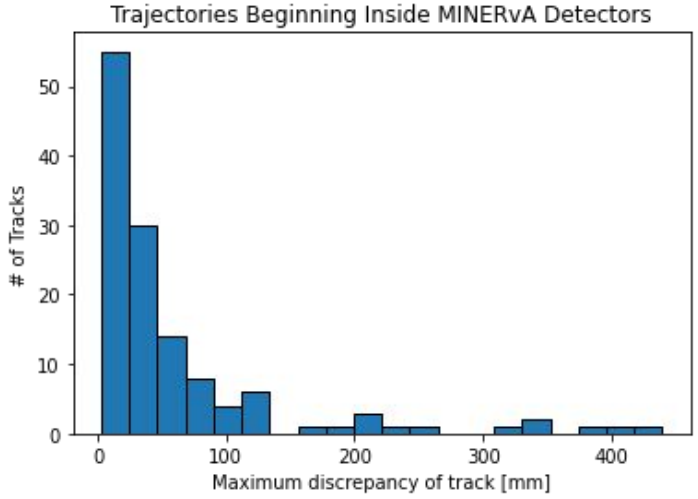
Outlier track distances [mm] not shown in the histogram (in):

1. 345.0270 7264
2. 467.9203 6432
3. 569.8452 2099

Outlier track distances [mm] not shown in the histogram (out):

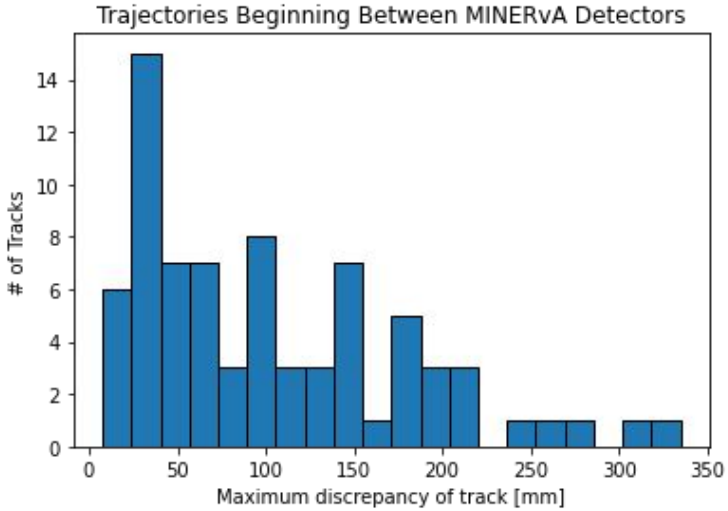
1. 4054.943 49821
2. 13111.18 250128
3. 4066.199 53907
4. 6738.770 41954
5. 4516.998 88031
6. 5337.771 09498
7. 4835.318 30675
8. 4908.049 98466
9. 4425.181 05338
10. 4332.387 83857





Outlier track distances [mm] not shown in the histogram (in):

1. 3905.494 84552
2. 4190.587 41558
3. 817.4711 3494
4. 1666.170 13213
5. 2275.395 16008
6. 3314.329 35965
7. 712.3594 5205
8. 532.1478 7352
9. 752.7516 2336
10. 3065.024 80392
11. 1180.372 75475



Outlier track distances [mm] not shown in the histogram (in):

1. 644.14634367
2. 774.26967903
3. 453.08657559
4. 473.03673724
5. 908.24857615
6. 477.38120921 650.06884511