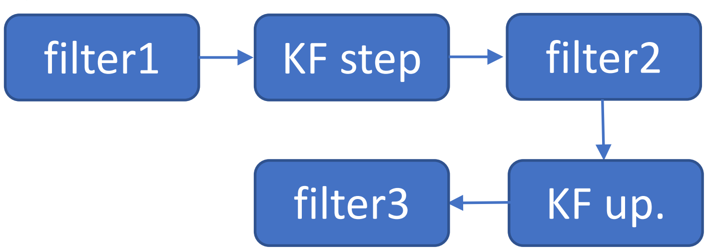
Status of SVD to CDC CKF

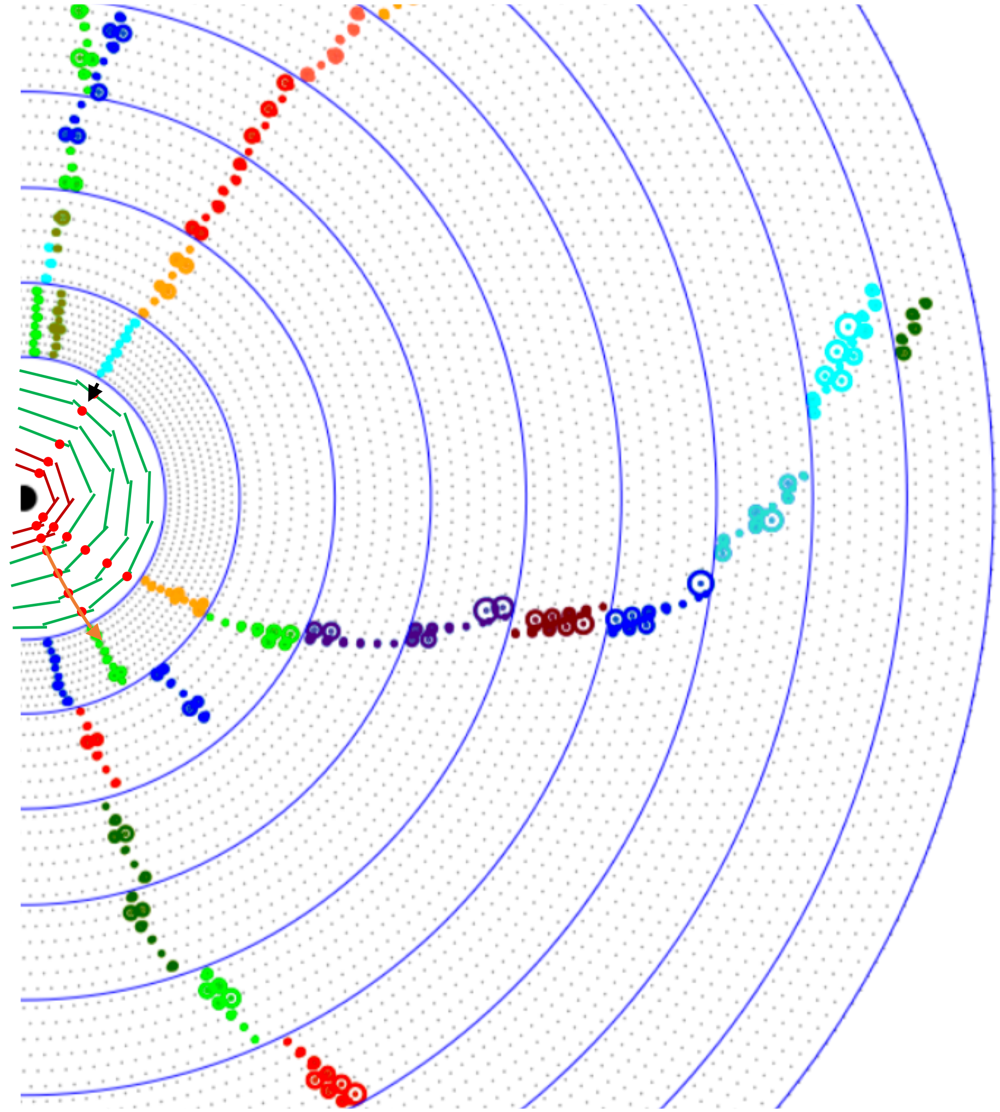
Nils Braun, Sasha Glazov

Miraim Kuenzel, Aiqiang Guo

SVD to CDC CKF

1. Stand-alone CDC tracking
2. Stand-alone SVD tracking
3. Merge CDC →SVD (CKF)
4. For unmerged CDC track, add extra SVD hits with CKF
5. For unmerged SVD track, add extra CDC hits with CKF
6. Extrapolate to PXD with CKF

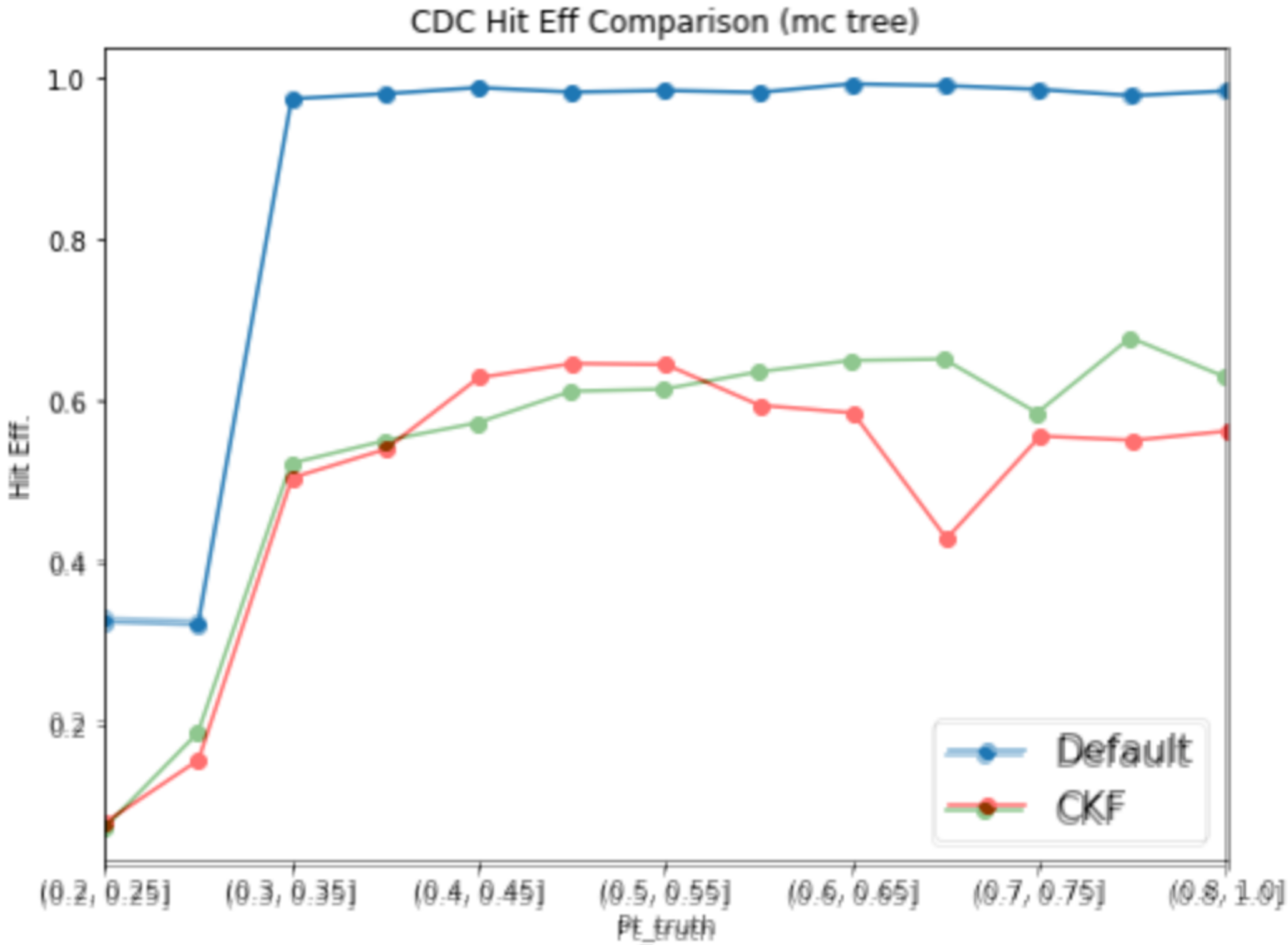




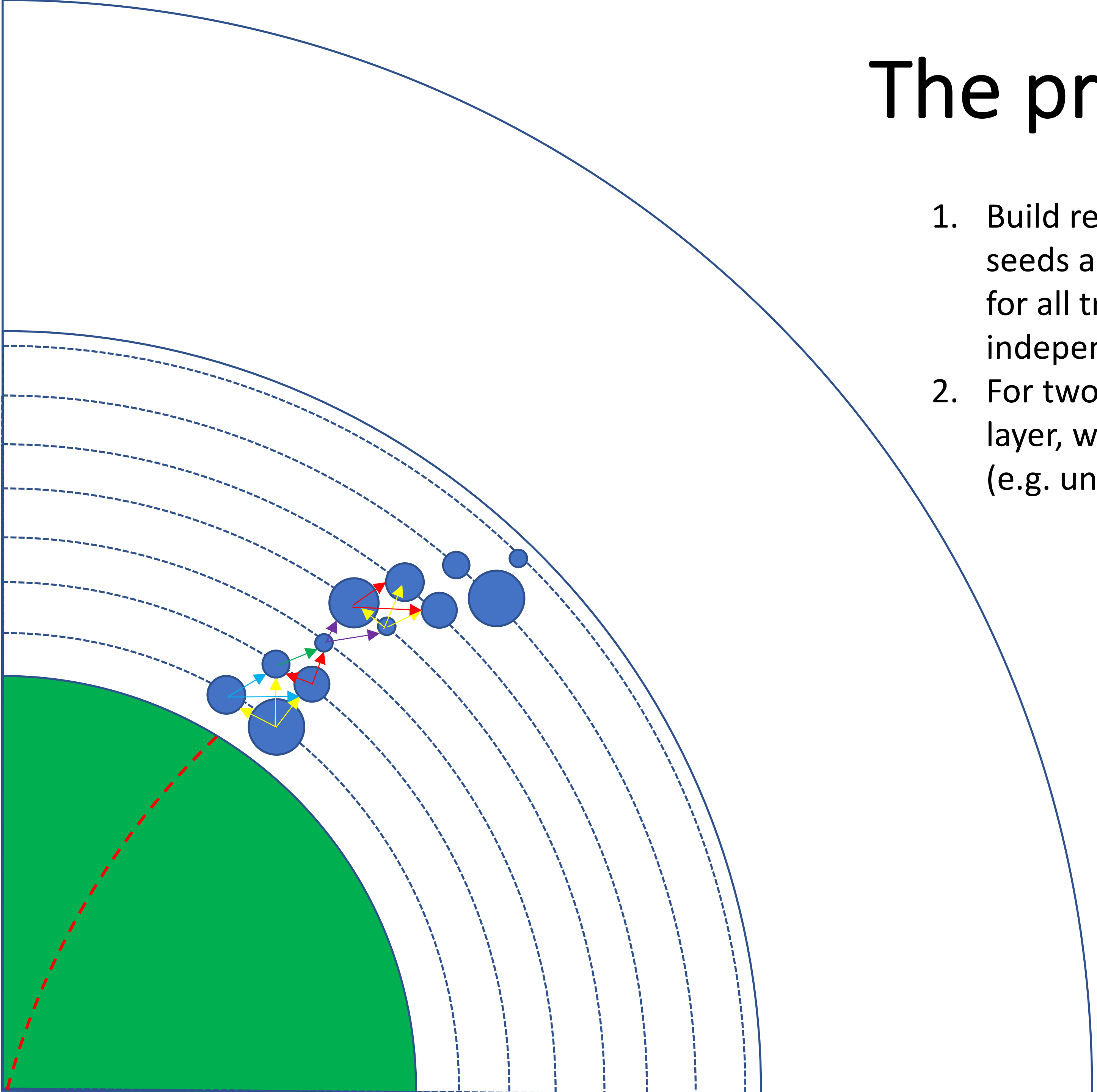
The filter variables

- Filter 1 and 2 (already fully implemented) will be distance-based filters, filter 3 will probably be a χ^2 filter
  - Filter 1: distance xy + sameHemisphere
  - Filter 2: distance mSoP xy + sameHemisphere
  - Filter 1 and 2 are momentum dependent
- For test purposes, I simulated one- and two-muon events with a **particle gun**, assuming a **uniform distribution** in the momentum spectrum between **0.2 and 3.5 GeV**

Hit efficiency μ^- and μ^+

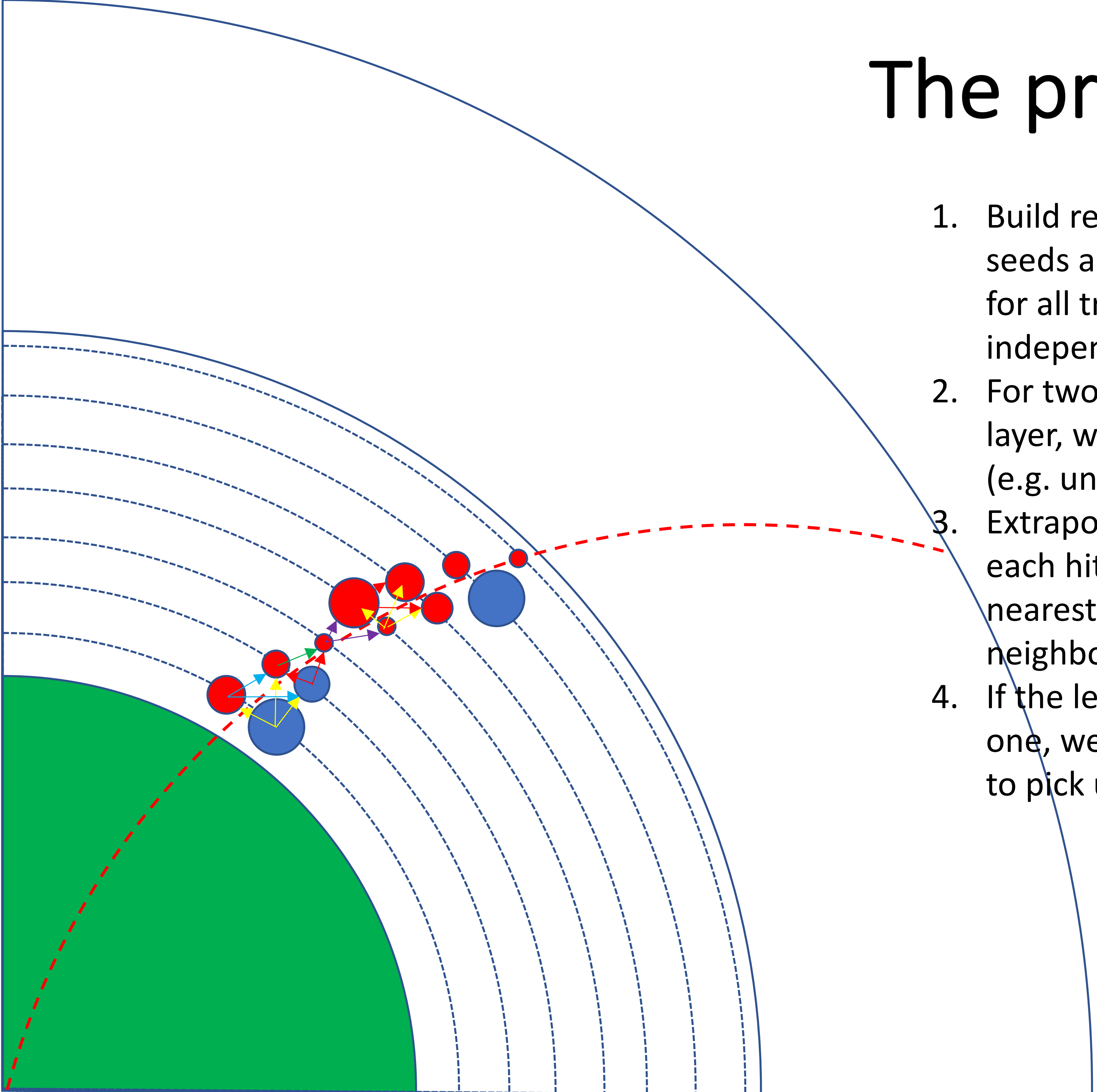


The problem



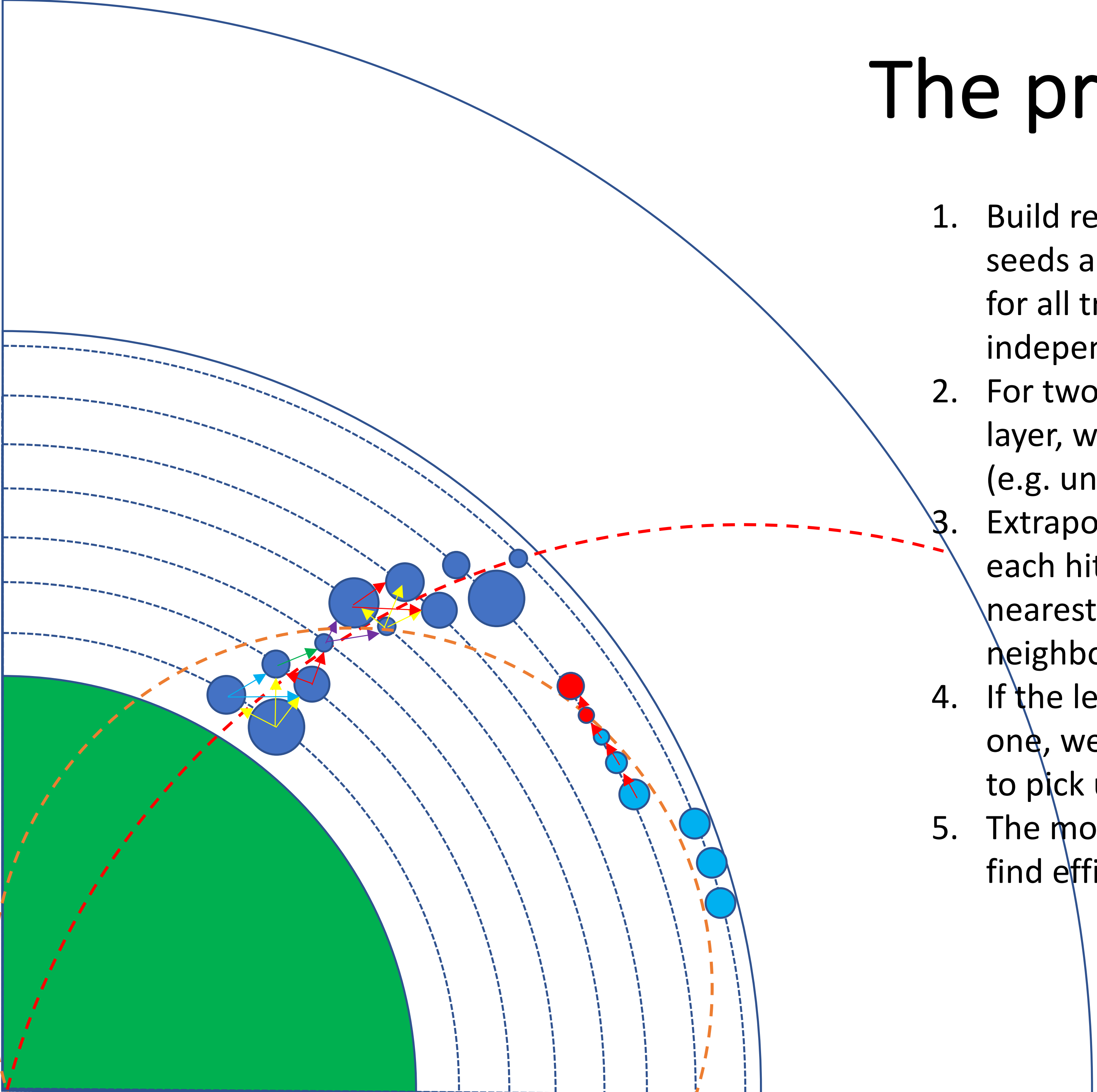
1. Build relations between seeds and hits (hits to hits) for all tracks (charge independent)
2. For two hits on the same layer, we have order them (e.g. unti-clockwise)

The problem



1. Build relations between seeds and hits (hits to hits) for all tracks (charge independent)
2. For two hits on the same layer, we have order them (e.g. unti-clockwise)
3. Extrapolate the seed track to each hit (state),pick the nearest one firstly from its all neighbors.
4. If the left hit is the nearest one, we don't have chance to pick up the right one.

The problem

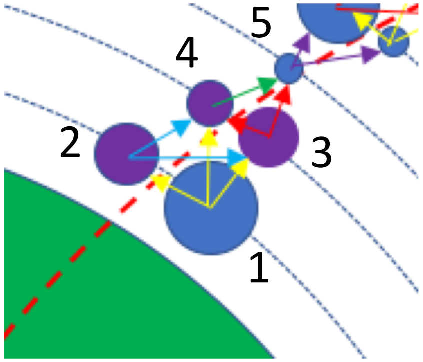


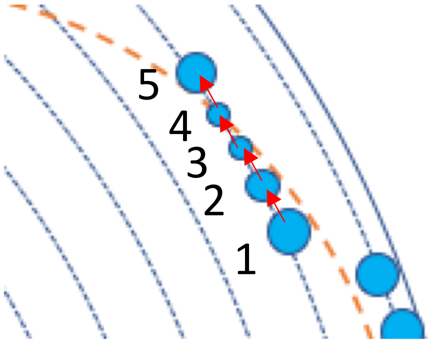
1. Build relations between seeds and hits (hits to hits) for all tracks (charge independent)
2. For two hits on the same layer, we have order them (e.g. unti-clockwise)
3. Extrapolate the seed track to each hit (state),pick the nearest one firstly from its all neighbors.
4. If the left hit is the nearest one, we don’t have chance to pick up the right one.
5. The momentum lower, the find efficiency worse

Some improvements

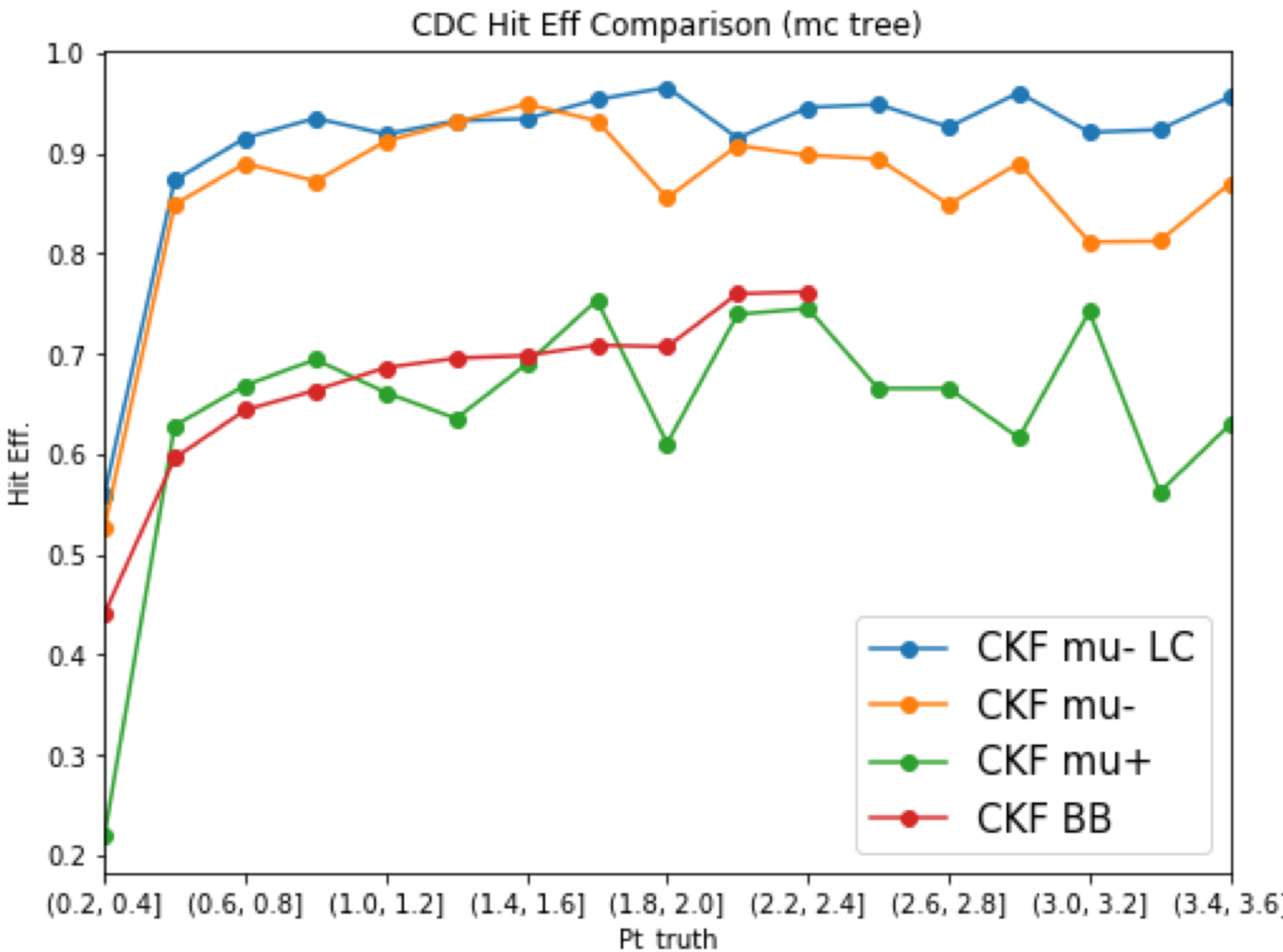
By Sasha

- Solution1:
  - Get charge information according to SVD seed track
  - Build track dependent relation
- Difficulty: the charge of the tracks with high momentum is hard to be determined correctly by SVD
- Solution2:
- Use the state with lowest cell-State index, which is set by cellular automation





Result of revised algorithm

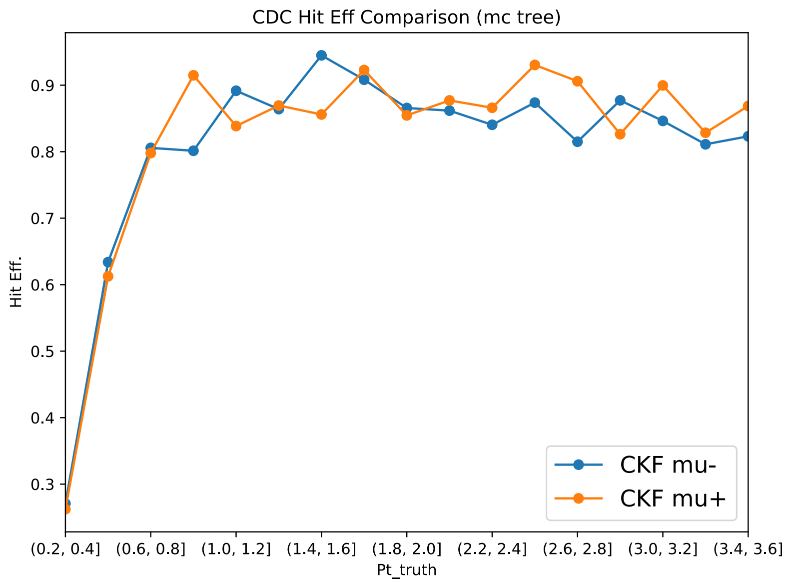


Solution3

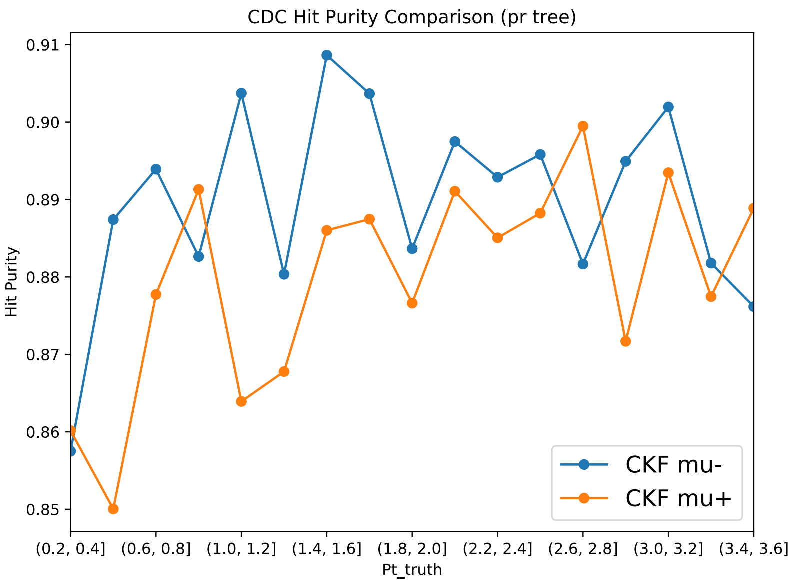
- Pick up the best hit at each layer firstly.
- Add the second best hit in each layer if there exist

By Sasha

Hit efficiency in CDC



Hit purity in CDC



The MVA filters

By Miraim

- Produce MVA weight files for the three filters. FastBDT
- Training set: 5000 one muon events, 0.3 <= p_t <= 3.5 GeV (uniform), uniform phi distribution, varying seed number
- Testing set: 5000 two muon events, 0.3 <= p_t <= 3.5 GeV (uniform), uniform phi distribution

Variables in the 3 filters

In filter1	In filter2	In filter3
inCDC	inCDC	inCDC
superlayer	superlayer	superlayer
phi	phi	phi
axial	axial	axial
tan lambda	tan lambda	tan lambda
pt	pt	pt
layer	layer	layer
arcLengthOfCenterPosition	arcLengthOfCenterPosition	arcLengthOfCenterPosition
z distance	z distance	z distance
xy distance	xy distance	xy distance
Same_hemisphere	Same_hemisphere	Same_hemisphere
arcLengthOfHitPosition	arcLengthOfHitPosition	arcLengthOfHitPosition
	mSoP z distance	mSoP z distance
	mSoP xy distance	mSoP xy distance
		chi2
		residual

Efficiencies result:

By Miraim

UseNStates=1 used for MVA filter…	… 1	… 2	… 3
Track Finding Efficiency	91.45	86.25	90.3
Hit Finding Efficiency	88.89	53.59	58.71
Clone Rate	3.99	1.99	2.32
Fake Rate	6.57	13.34	9.8

- Efficiencies were calculated using the filter chain MVA1 – Kalman step – MVA2 – Kalman update – MVA3
- UseNResults = 1 is either used for MVA1, MVA2 or MVA3
- Same variable set for all three MVA weight files

Efficiencies result:

By Miraim

UseNStates=1 used for MVA filter…	… 1	… 2	… 3
Track Finding Efficiency	89.7	85.1	90.65
Hit Finding Efficiency	85.77	43.9	52.8
Clone Rate	3.34	1.79	2.32
Fake Rate	5.16	13.21	8.30

- Increasing number of variables

Summary

- All the codes are ready
- More tuning and optimization are needed
- The algorithm probably can be implemented in the next release as an option of default TF

Backup

The basic version

- Branch name:

  feature/VXD_CDC_CKF_01

- Filter1: [>0.5], [0.5, 0.3], [0.3, 0]
- **HelixDistance< 2.3, 2.17, 2.19 Loose Cut**
- ***HelixDistance< 1.0, 1.0, 1.0 Tight Cut***
- Filter2:
- Residual < 2, 1.78, 1.75
- Residual2< 1.1, 0.6, 0.7
- Filter3:

The implementation

The current strategy is as follows:

- Load all the needed items: VXD seed tracks and CDC hits
- Turn every loaded object into one CKF state
- Implement the seed-hit and hit-hit relations(those may be based on clusters, segments or layers)
- Hits allowed by the relation maps have to survive the following chain: filter 1 - Kalman step - filter 2 -Kalman update - filter 3(Use NResults where possible)
- The result is written out as one (preferred) or more reco tracks