Hashing for Track Reconstruction

CHEP 2019

Sabrina Amrouche, T.Golling, M.Kiehn (Université de Genève)
A.Salzburger,N.Calace (CERN)

c.amrouche@cern.ch

Motivation

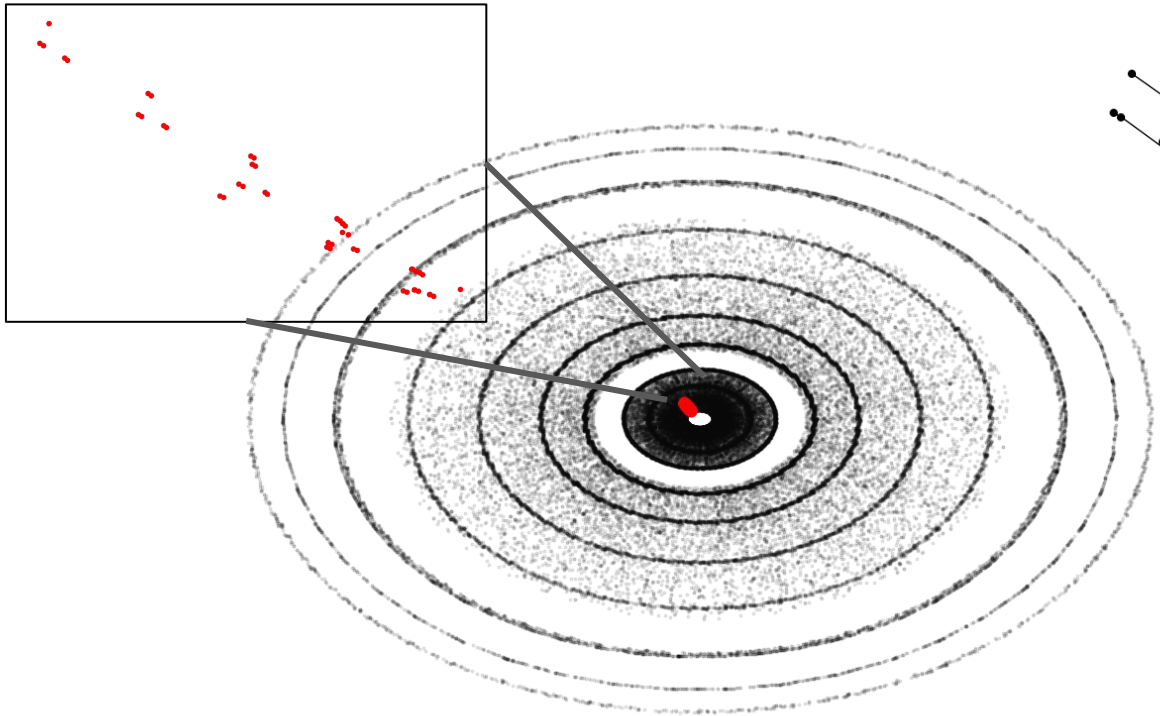
- High-luminosity LHC (and future FCC) will bring very high pile-up scenarios
- Average 200 proton-proton collisions for HL-LHC (up to 1000 proton-proton collisions for FFC)
- Combinatorics for track reconstruction becomes very challenging

Outline

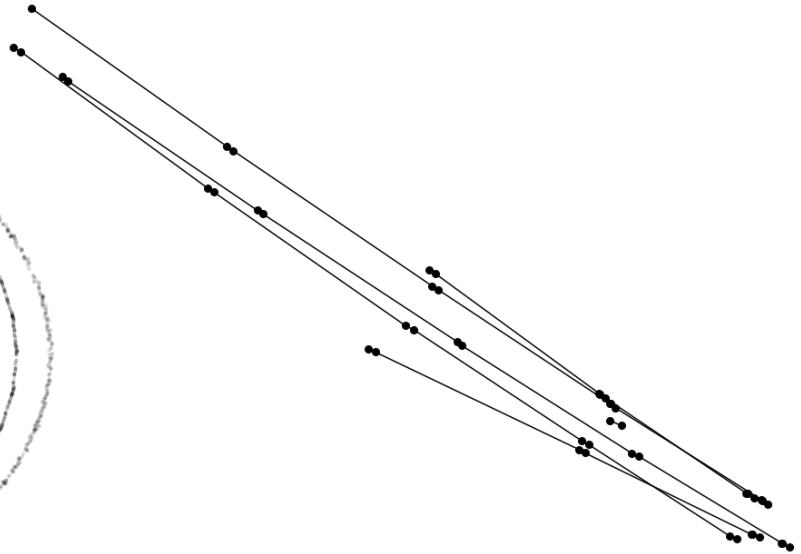
```
#parallel, fast and generic
def similarity_hashing(hits):
    ...
    return tracks
```

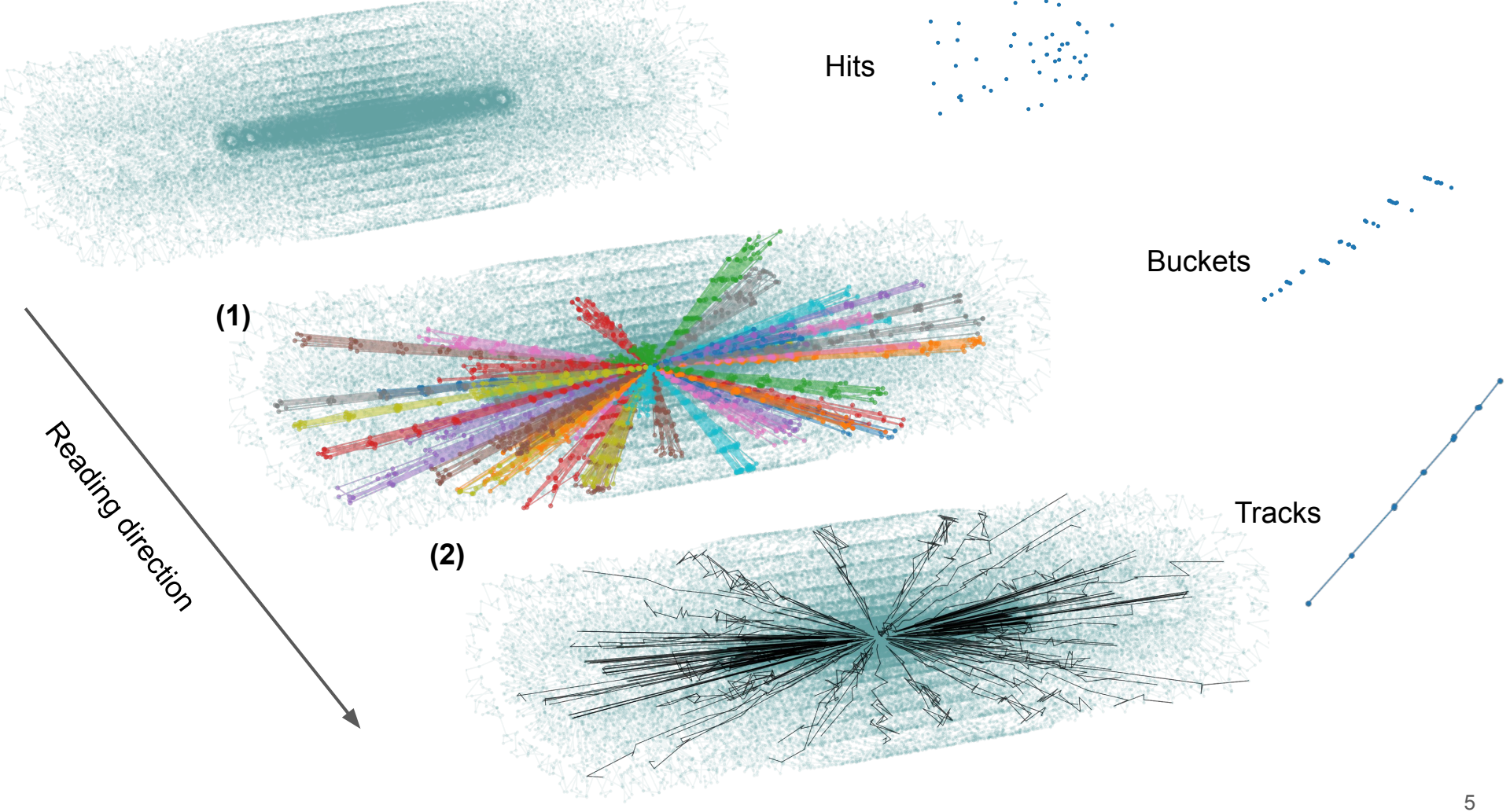
This approach is based on:

(1) Translating the dataset to small sets

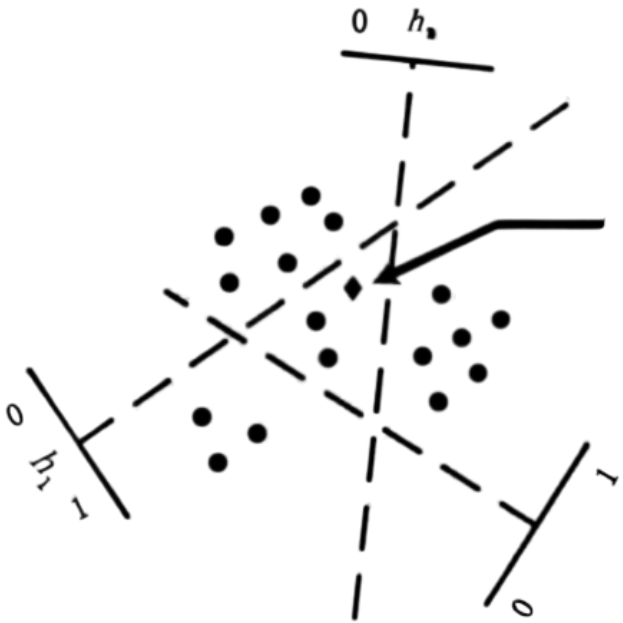


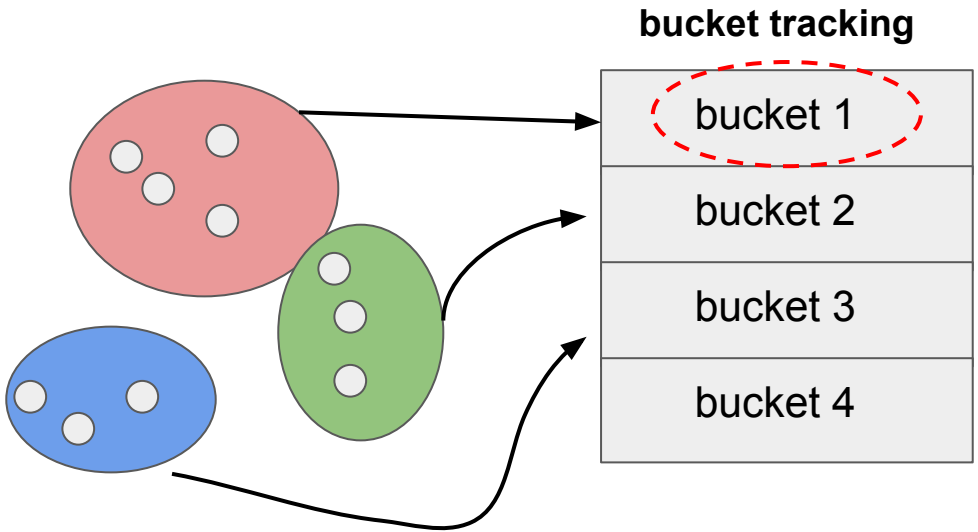
(2) Tracking in way less complex environment




Hits
Buckets
(1)
Tracks
(2)
Reading direction

Partition strategy : Hashing





Partition strategy : Hashing

Tree-based

Hashing-based

Quantization

Relative Neighborhood Graphs

Navigable Small-World Networks

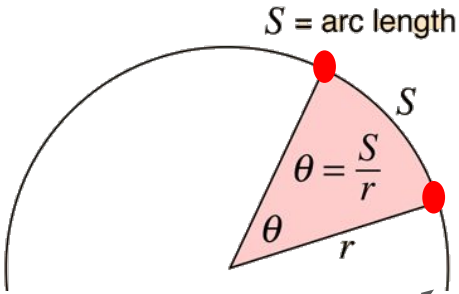
KD-tree **R-tree** **VA-file** **Locality Sensitive Hashing** **Product Quantization**

Neighbors definition : distance

For a given query point, ANN returns the nearest neighbors ,i.e whose **distances are the smallest**.

Metric used : <u>angular distance</u>

$\theta = S / R$



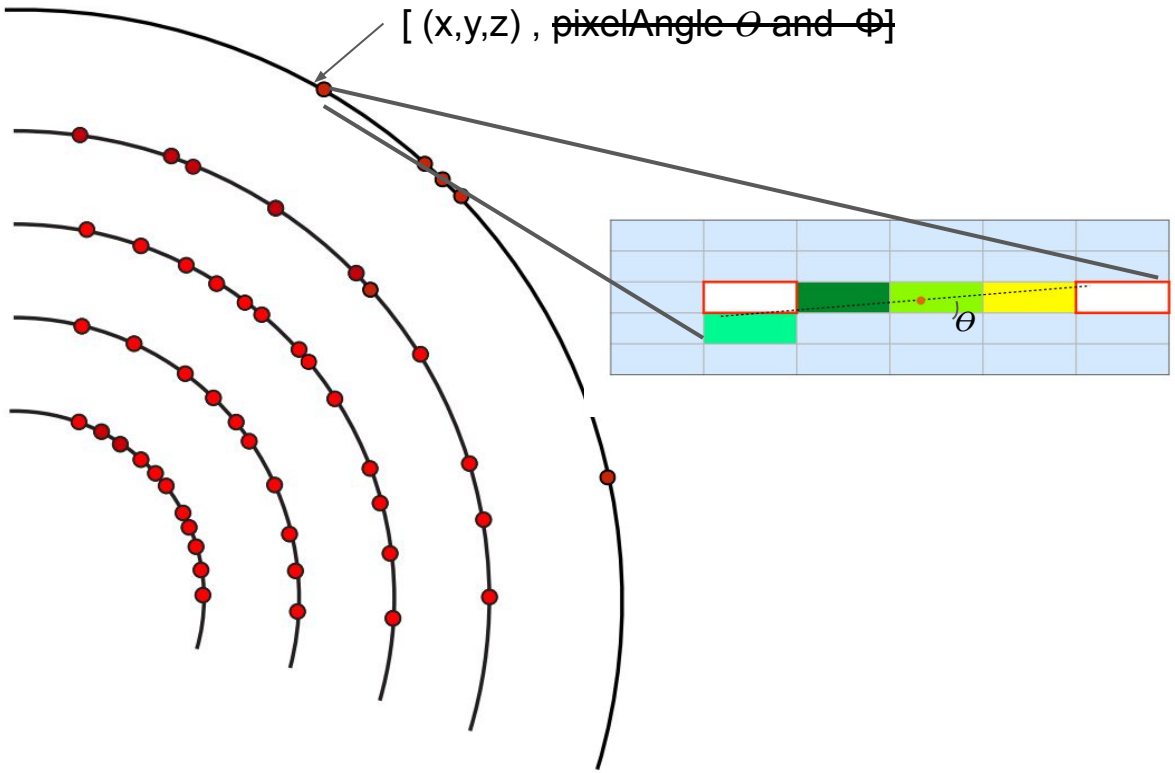
where S = distance travelled and R = radius of the circle

other metrics are possible : metric learning, DNN...

Dataset : TrackML

- 10k particles, 100k points
- **3** features : global x, y, z
- Hashing library used :

  Annoy (spotify)
- Buckets parameters
  - bucket size
  - nb trees (graphs)



Simulated with ACTS, Ttbar event, mu 200

[See talk on Tuesday by Jean-Roch Vlimant]

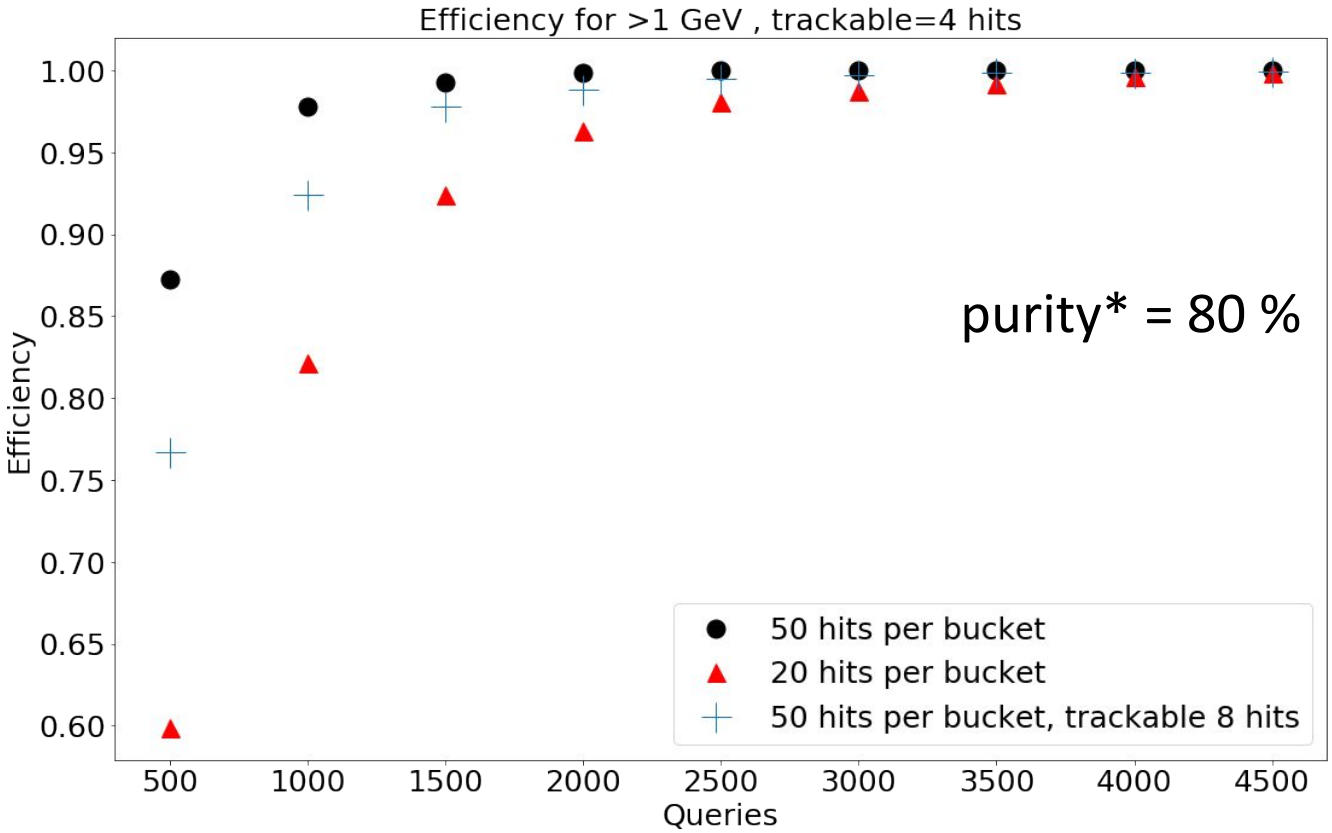
Bucket quality : Efficiency

- Using truth information :
  - Count number of trackable particles found in a bucket
  - “Trackable particles” = 8 points (avg 11)
  - Two bucket sizes : 20 and 50
  - Efficiency as :

    nb_trackable_particles_found/ total_trackable_particles (~9K)

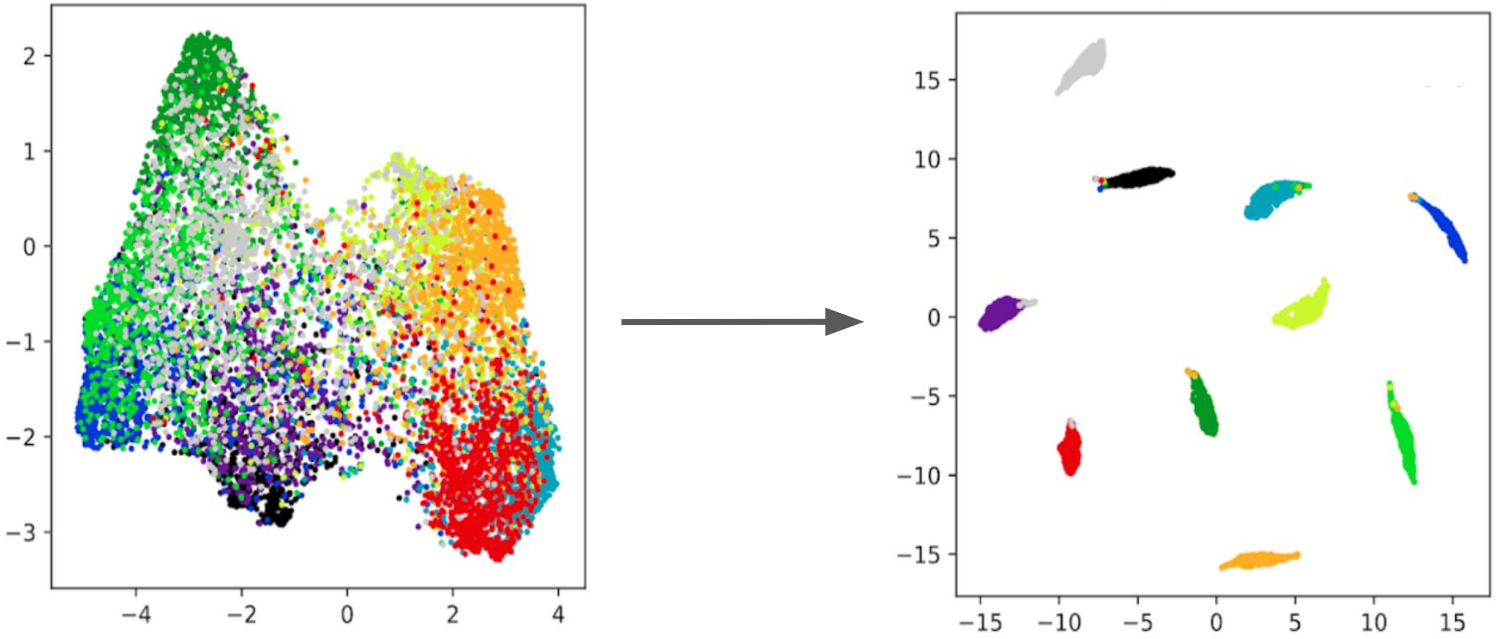
Bucket quality : Efficiency

Rebuilding the data structure, after removal of found particles,improves the efficiency (in backup)



* A track will be marked as reconstructed if 80 % of its hits are found inside the same bucket

Metric (similarity) Learning



Metric Learning

- Learn a projection that improves tracks separability
  - Local Fischer Discriminant Analysis (LFDA[1])
  - Solved as a generalized eigenvalue problem

Fig 1 Transverse view (x/y)

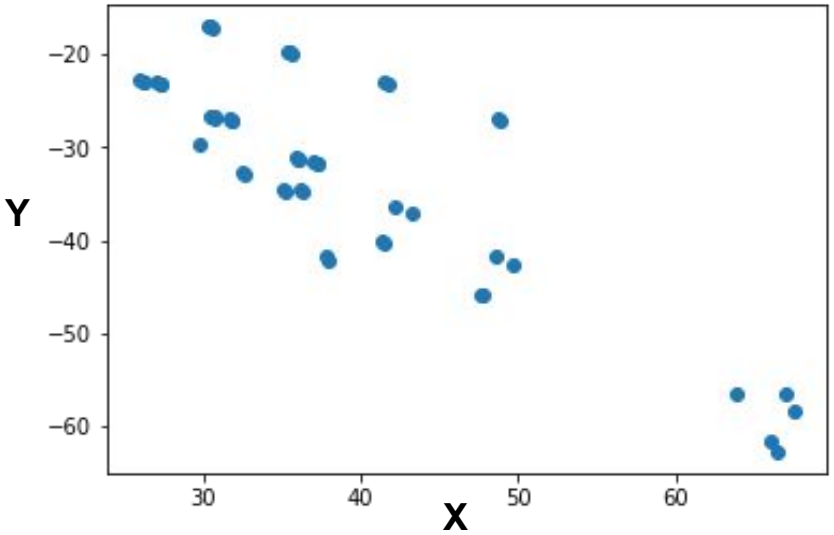
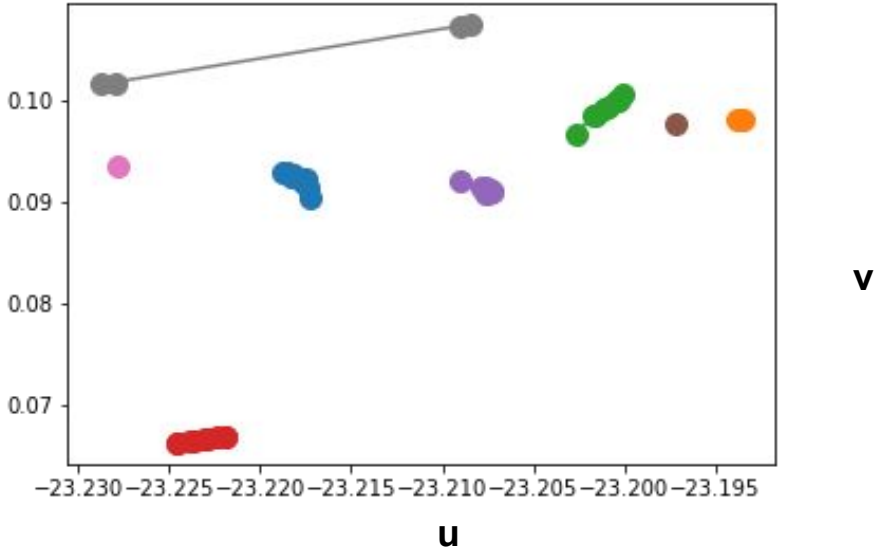


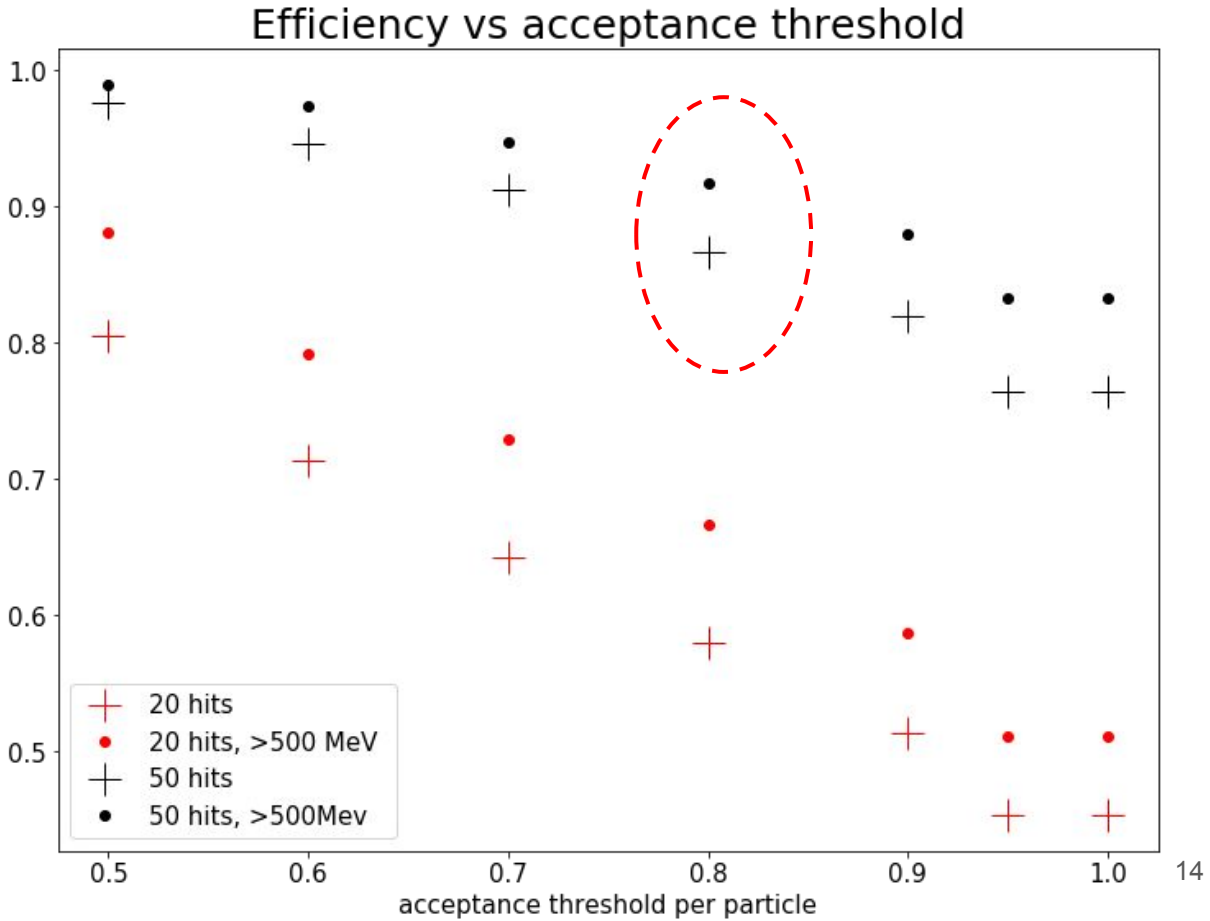
Fig 2 Learned (u/v) space



[1] Sugiyama, Masashi. "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis." *Journal of machine learning research* 8.May (2007): 1027-1061.

Metric Learning

- 20 and 50 hits per bucket
- Min particle size = 5 hits
- **100K** buckets
- Min **pT=500 MeV**



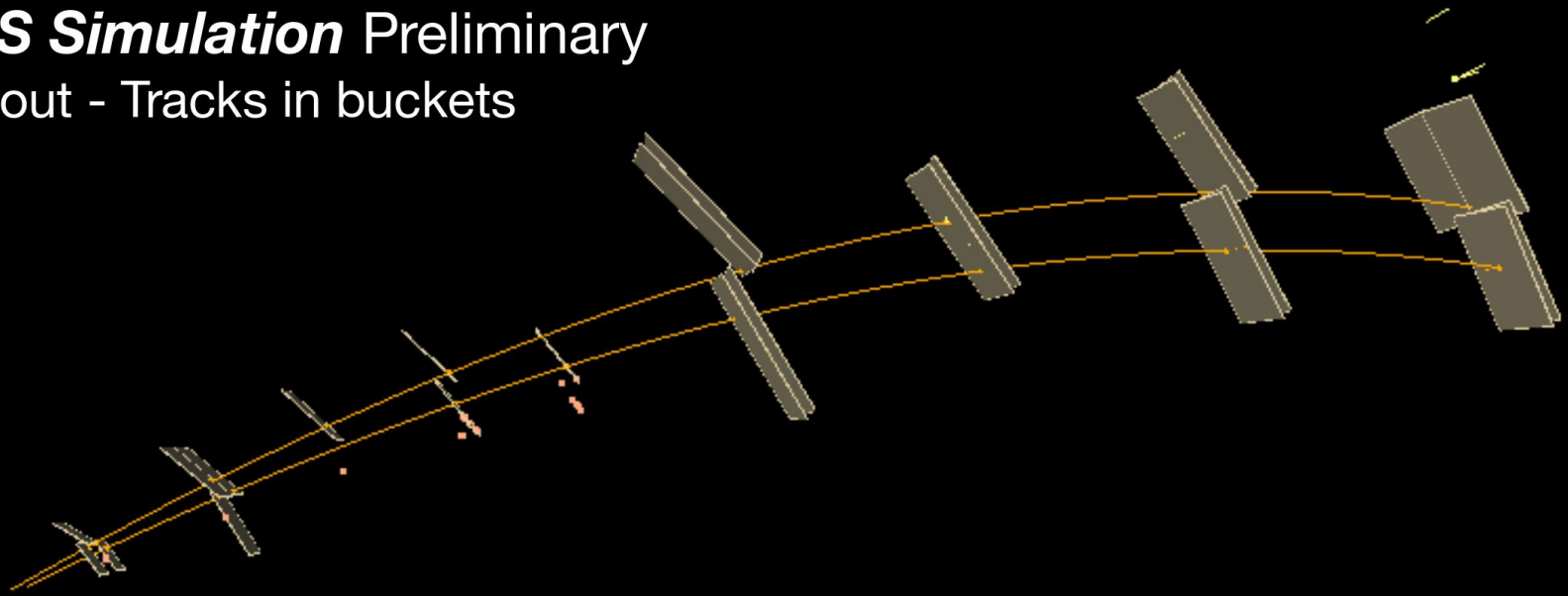
Testing on ATLAS Phase-2 tracker (ITK)

- Hashing performed as described
- ATLAS reconstruction applied on buckets
- HL-LHC scenario

Atlas-IDTR-2019-008

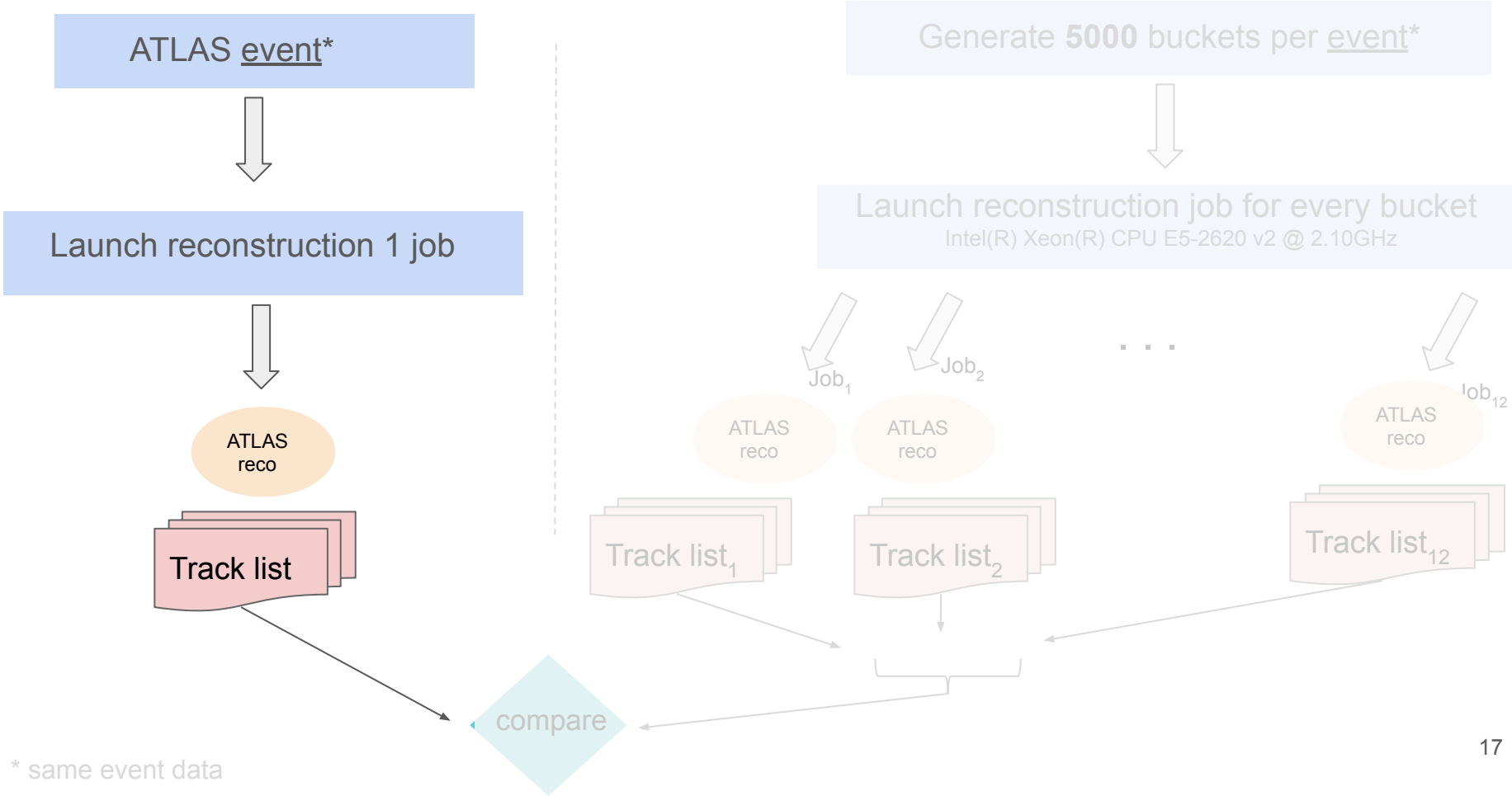
ATLAS Simulation Preliminary

ITk layout - Tracks in buckets



Tracking in buckets

(1) Standard ATLAS Phase-2 reconstruction



* same event data

Tracking in buckets

(1)

ATLAS event*

Launch reconstruction 1 job

ATLAS reco

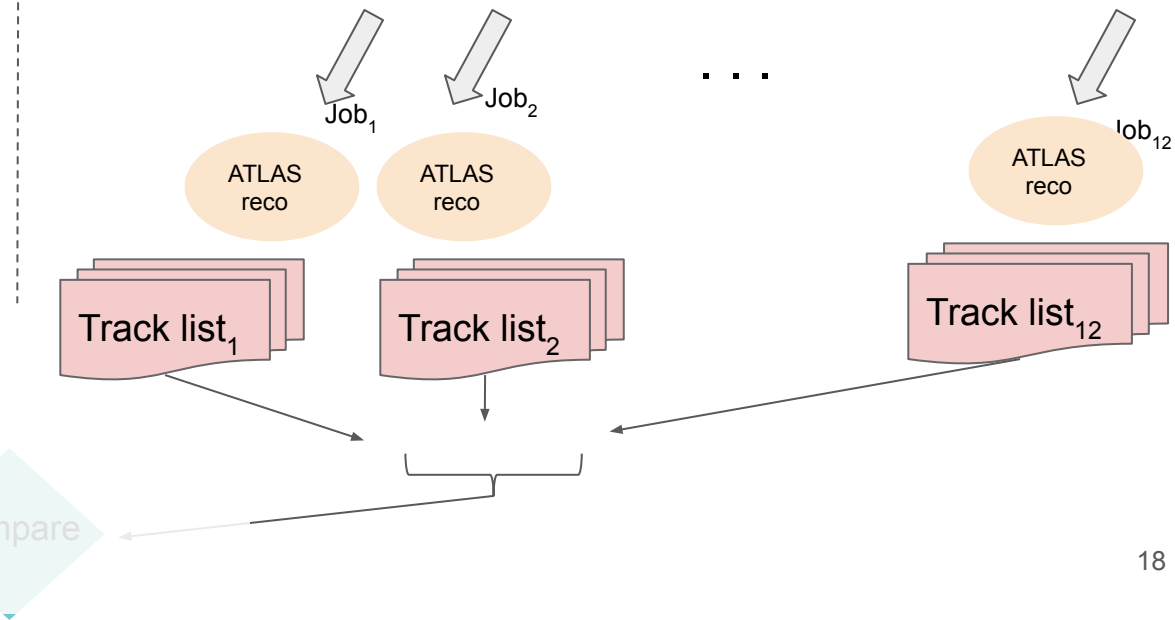
Track list

(2) Adapted reconstruction with bucket input

Generate **5000** buckets per event*

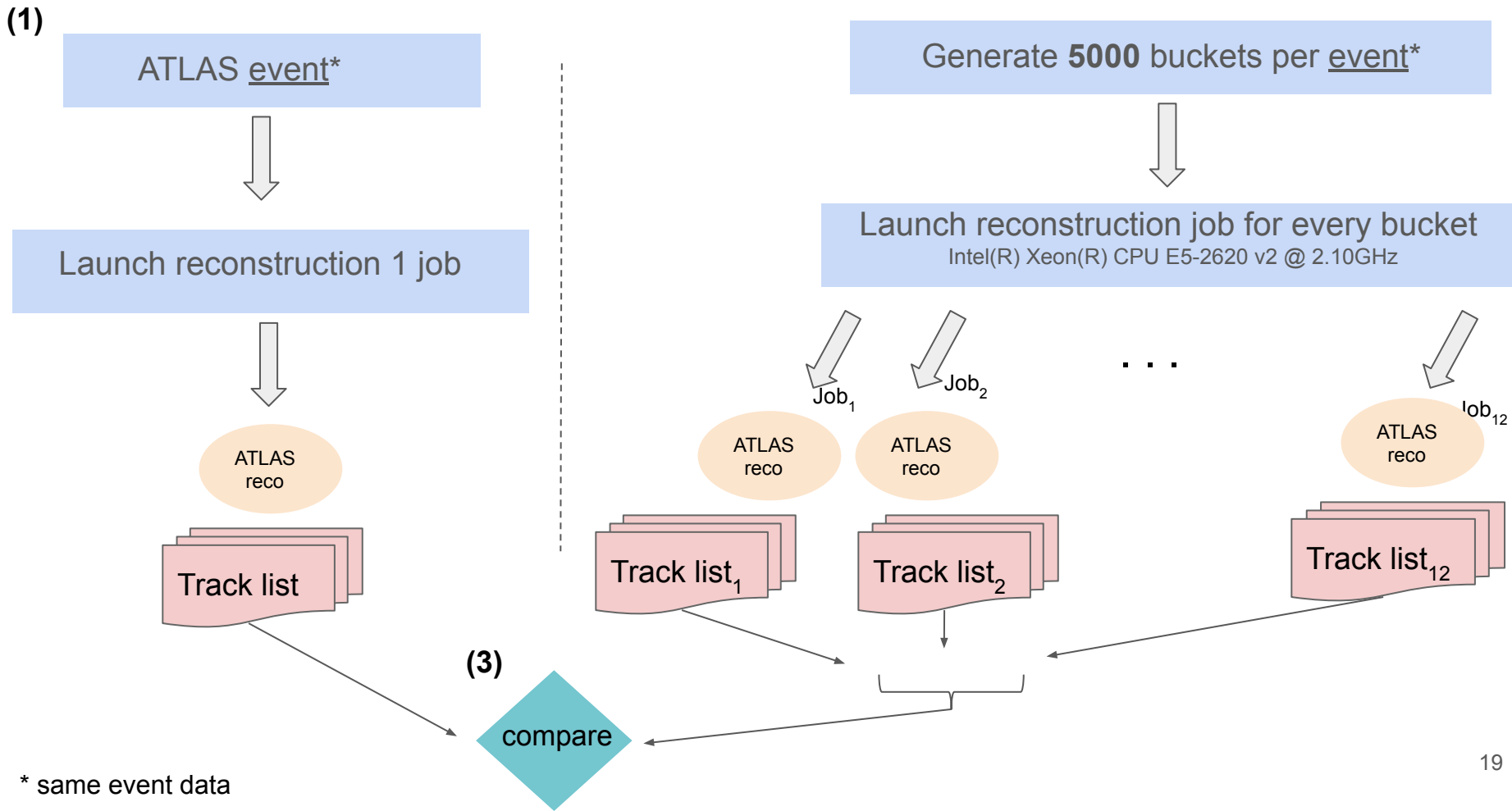
Launch reconstruction job for every bucket

Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz



* same event data

Tracking in buckets



* same event data

Comparison synthesis : **12%** of total ATLAS tracks found

- **Full match**

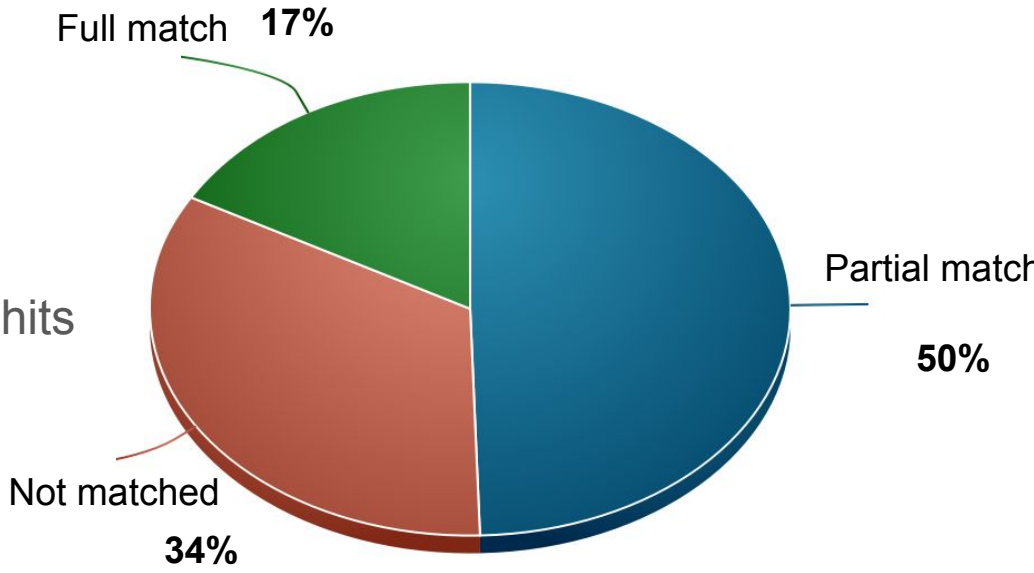
  {ATLAS tracks} = {Bucket tracks}

- **Partial match**

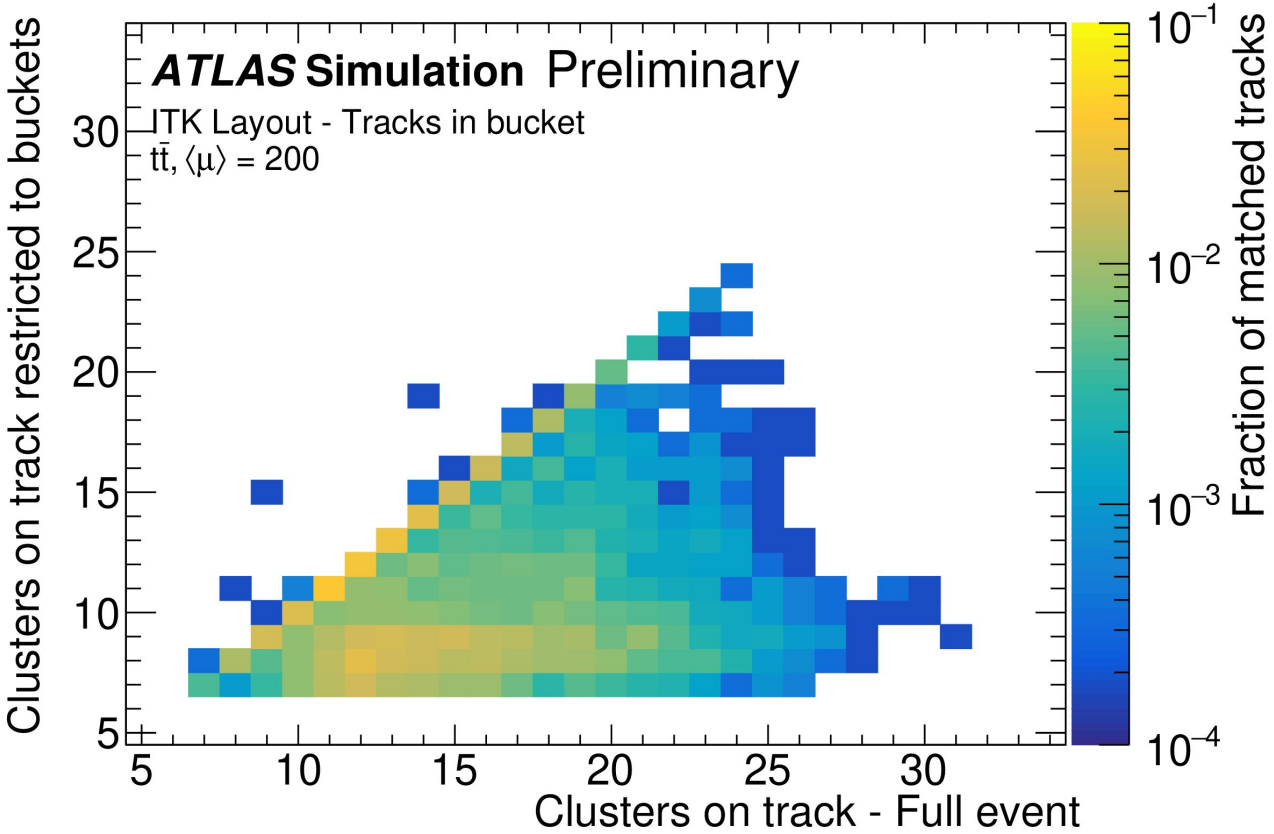
  {ATLAS tracks} ∩ {Bucket tracks}≥ 7 hits

- **Not matched**

  tracks not found by ATLAS



Reconstruction in bucket vs full event : #clusters

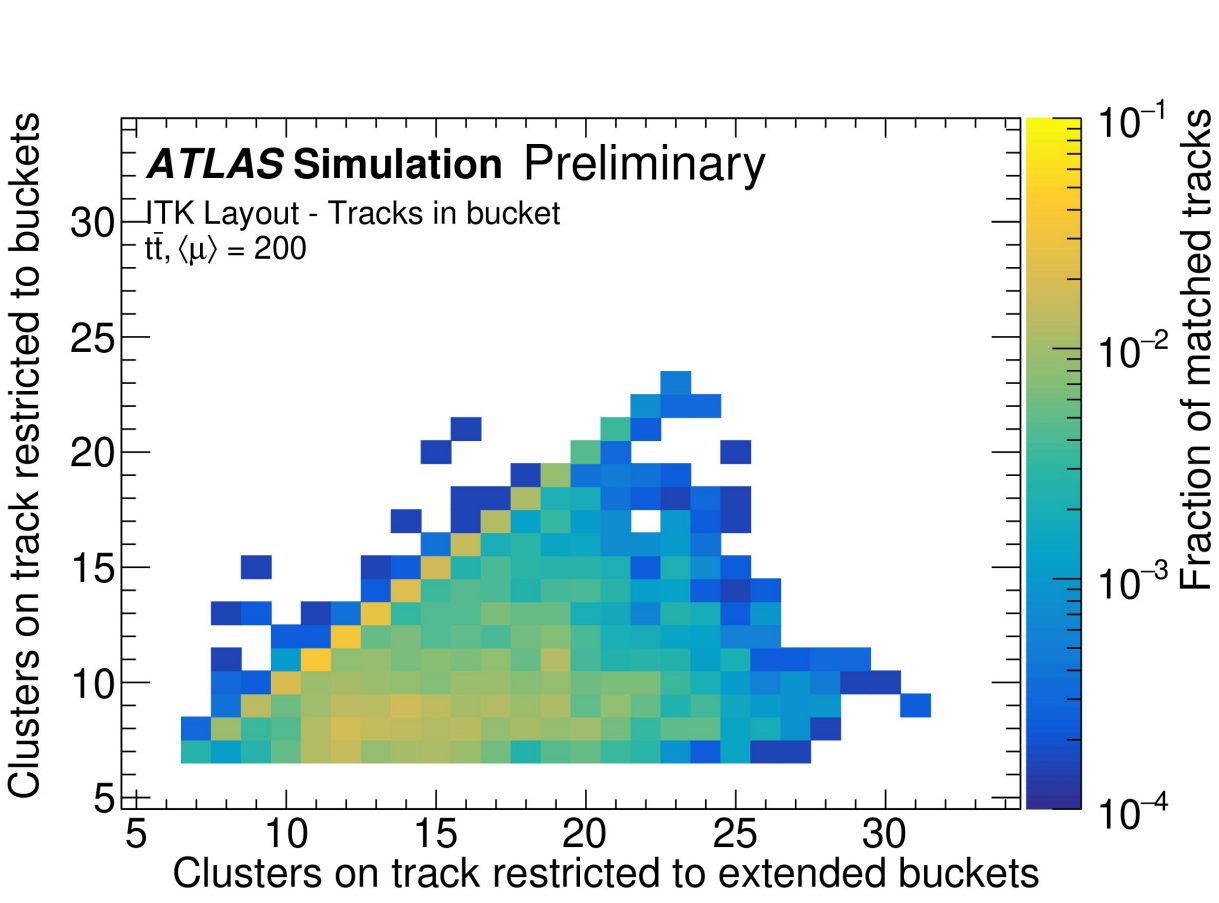


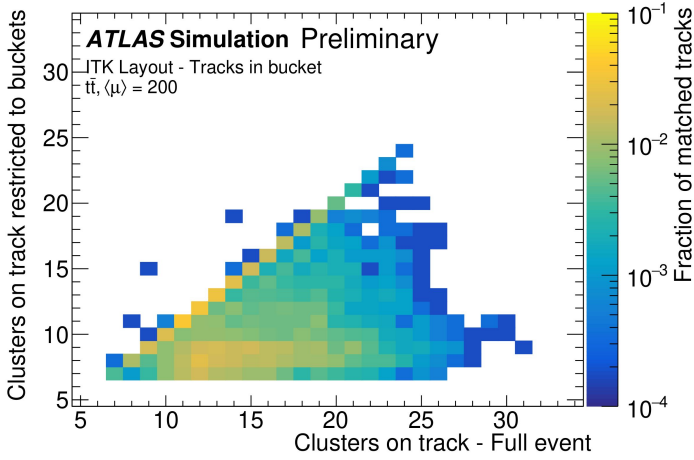
Number of clusters on tracks running the full event reconstruction versus restricting the track reconstruction algorithms to a bucket of 50 hits.

Bucket extension : Motivation

- Bucket is not guaranteed to contain all hits of a track
- Combinatorial track finding of ATLAS is designed to find compatible hits in a road
- Emulate a perfect road extension by augmenting the buckets with truth hits

Reconstruction in bucket vs extended bucket : #clusters





Conclusion and next steps

- Fast and efficient data structures for tracking with TrackML dataset
- First tests on ATLAS Phase-2 Monte Carlo
- Open source hashing /ANN libraries github.com/spotify/annoy

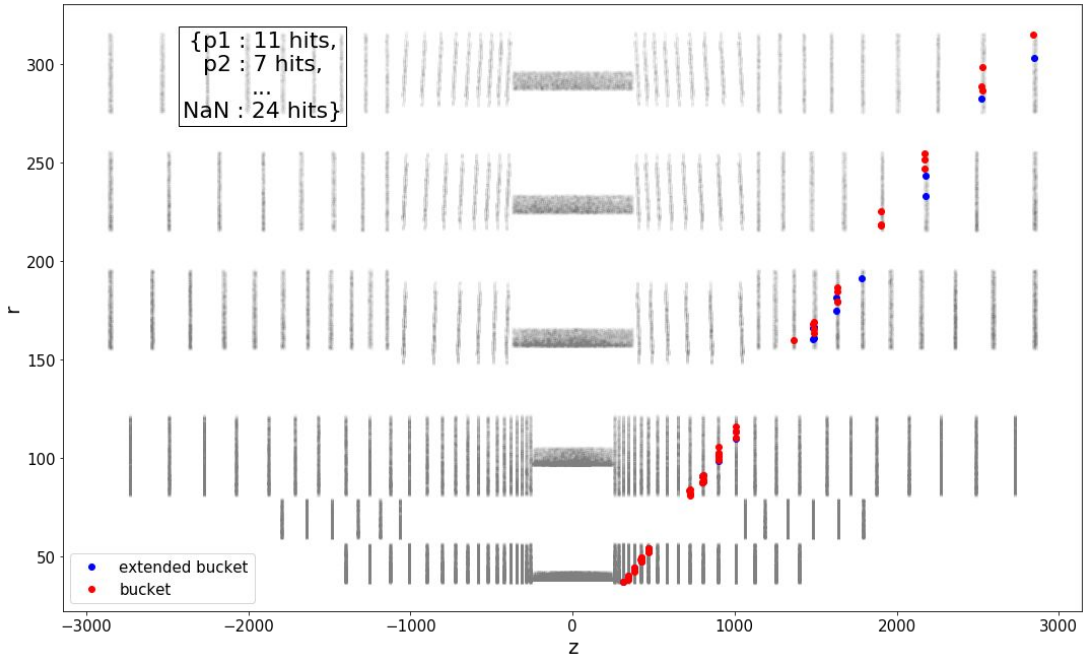
Next

- First implementation of this setup being tested in ACTS
- On-demand bucket extension done by track finder

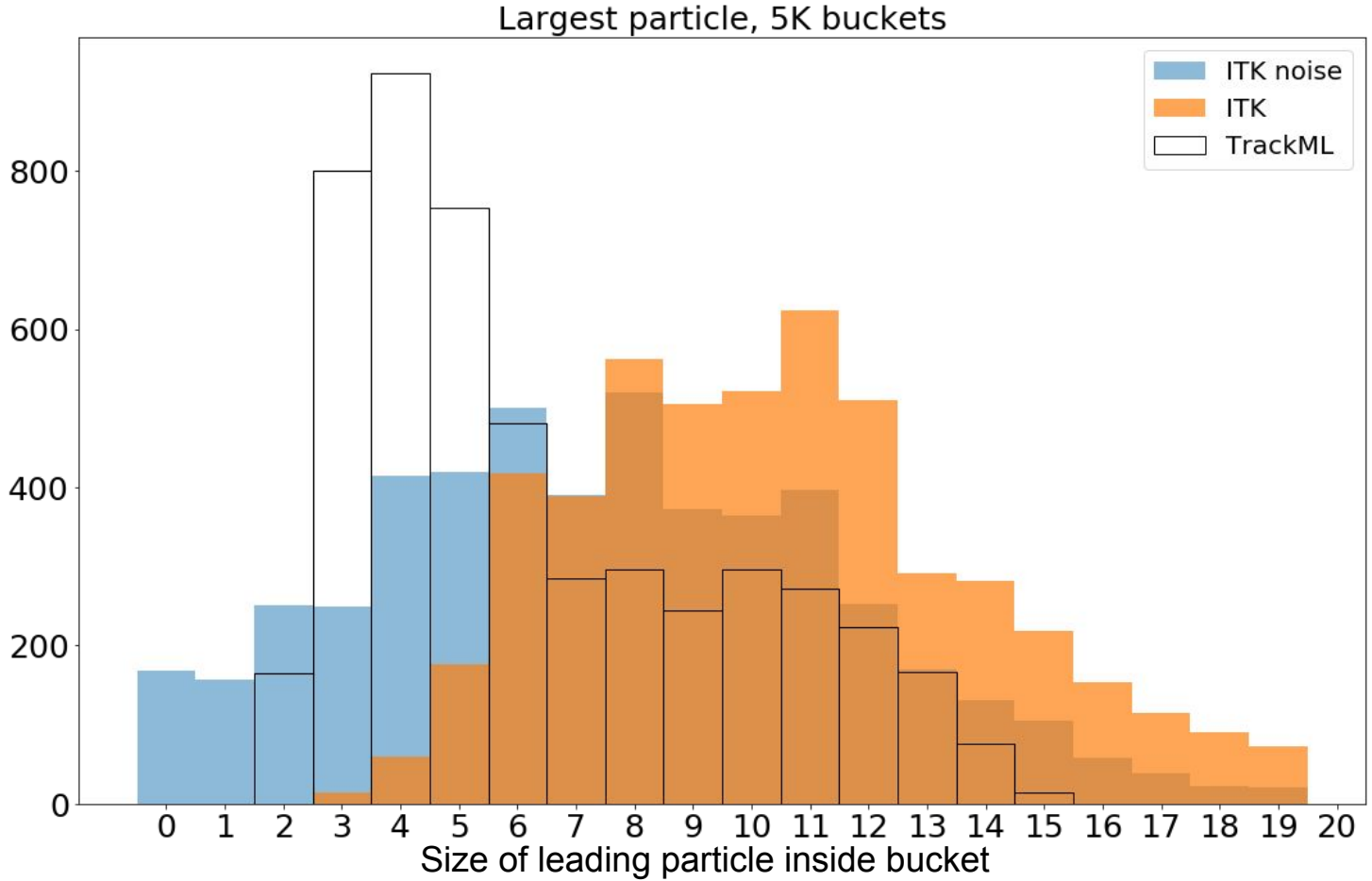
Backup

Bucket extension

1. Seed inside the bucket, attach the “extended” hits and continue the tracking chain
    - Use all the clusters into the extended bucket, while converting to space points only the ones in the original bucket
2. Run one reco job per bucket : restrict reconstruction to the bucket

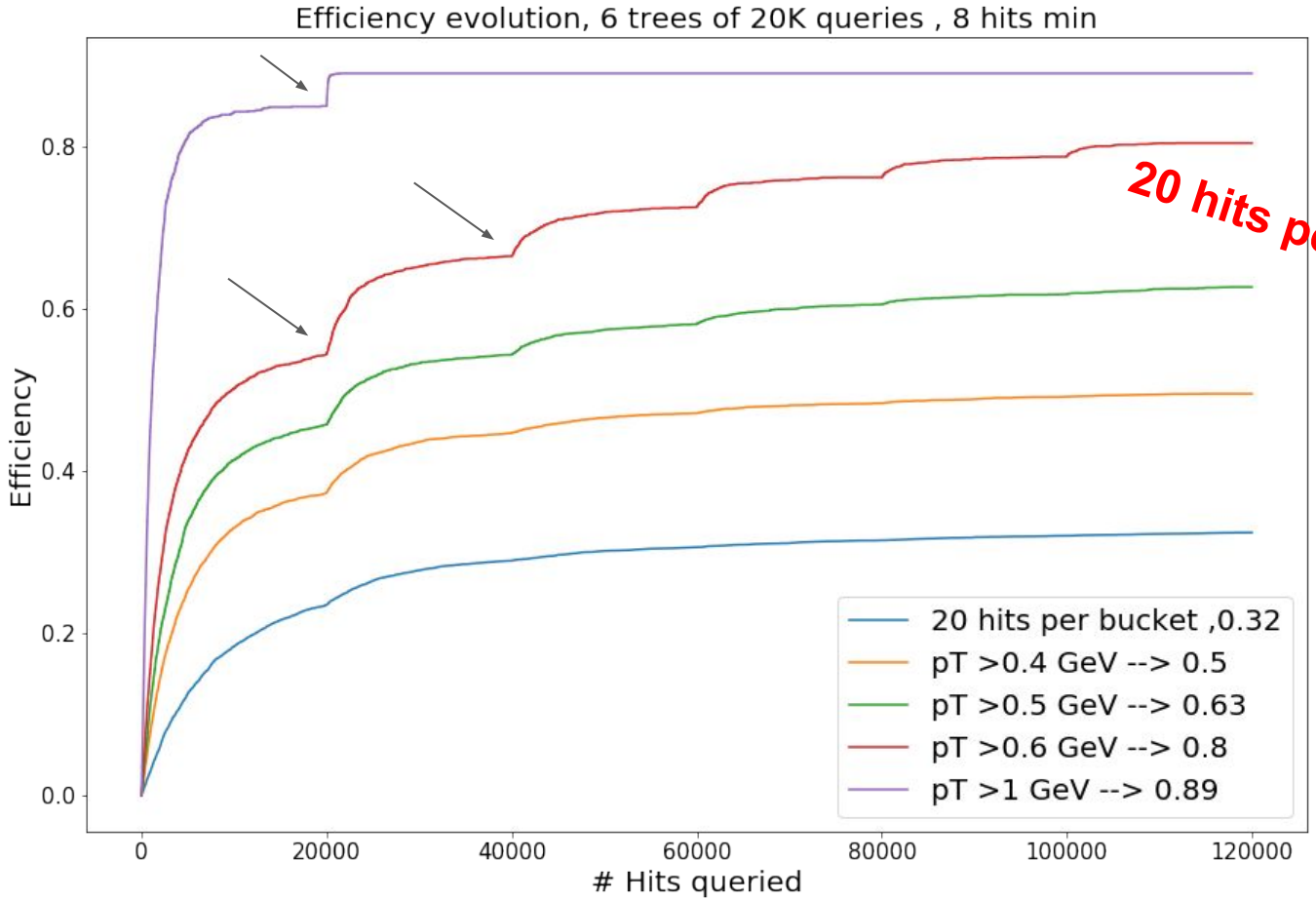


Bucket quality : #same-particle hits, 20 hits per bucket



Rebuilding the data structure

0.07 ms per query



20 hits per bucket