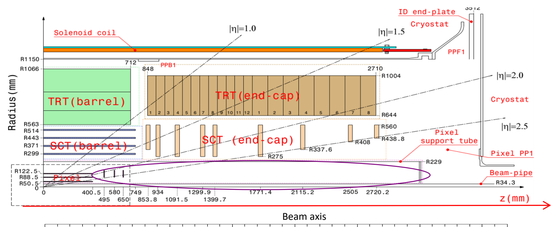
CTD 2023 Poster Session
Seeding with ML in ACTS

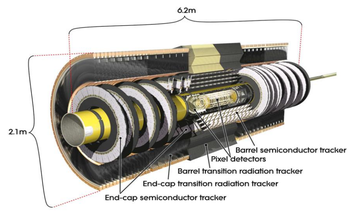
Laboratoire d'Annecy Le Vieux de Physique des Particules (LAPP)
University Savoie Mont Blanc, France

Jeremy Couthures
jeremy.couthures@lapp.in2p3.fr

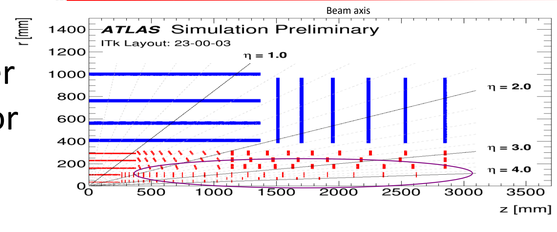
1. ITk and HL-LHC

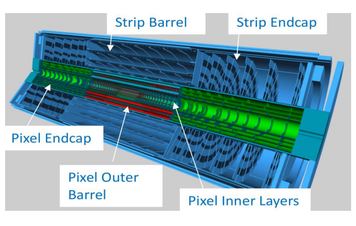
Current Inner detector





Inner Tracker (ITk) detector



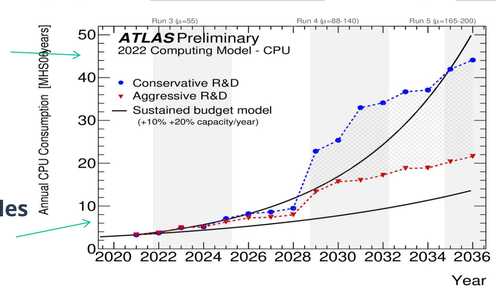


- ITk:
  - Wider coverage: $|\eta| < 4$
  - Higher granularity

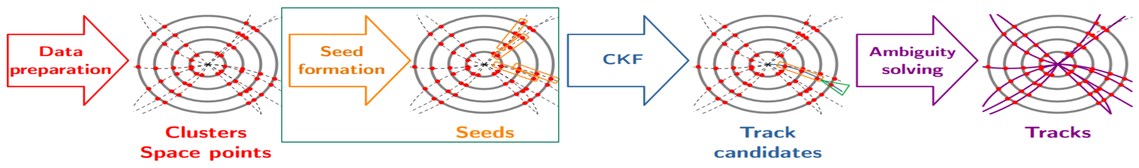
→ **More particles detected**

- High Luminosity-LHC (HL-LHC):
  - Between now and 2029: Luminosity x2.5

→ **More particles created**



2. Goal

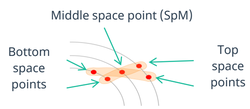


Idea: introduce a novel approach for seed construction (**Hashing**)
What do we hope to improve? **Seeds' efficiency, purity (fake rate), redundancy (duplication rate)**

3. Evaluation

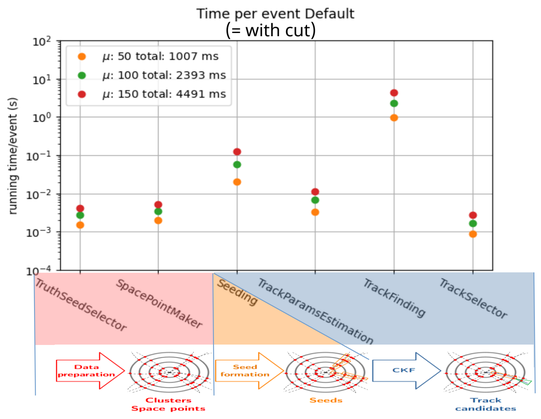
ACTS: A Common Tracking Software (generic tool) (experiment independent)
Samples: 100 $t\bar{t}$ events; $|\eta| \leq 4$; pT > 1GeV; μ = 50, 100, 150
Generic detector (default detector, ideal detector without material)
Softwares: Pythia8 (simulation), FATRAS (propagation)
But not using Geant4: No secondary particles (particles created from the interaction with the detector)

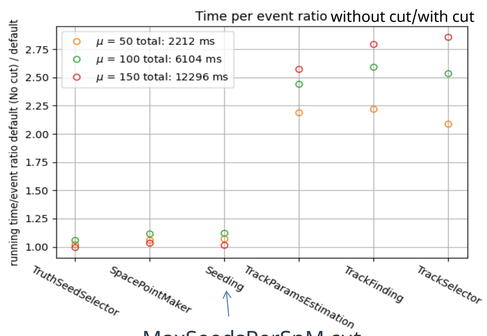
Evaluate on tracks:
1. **Efficiency**: Reconstruct as much "truth" tracks as possible
2. **Fake rate**: Reconstruct as low "fake" tracks as possible
3. **Duplication rate**: Avoid to duplicate tracks
4. **Running time**: Going as fast as possible



✓ Application: remove ACTS **MaxSeedPerSpM** cut, limiting the number of seeds to speed up the tracking → Non physical cut

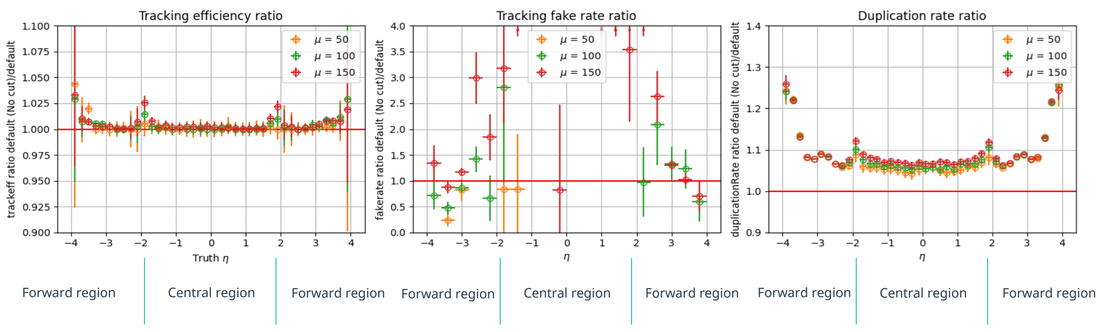
Seeds sharing the same middle space point





MaxSeedsPerSpM cut

Timing multiplied by ~2-3 without the cut



Same in central region, Better in forward region

Worst in central region, Same in forward region

Worst everywhere, Even worse in forward region

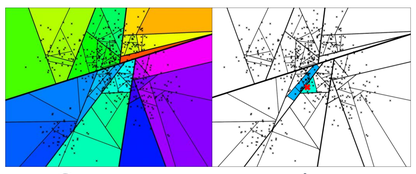
MaxSeedPerSpM cut decreases the performance in forward region
But improves in central region

4. Methodology

Hashing: Group similar space points into buckets and do the seeding on each bucket

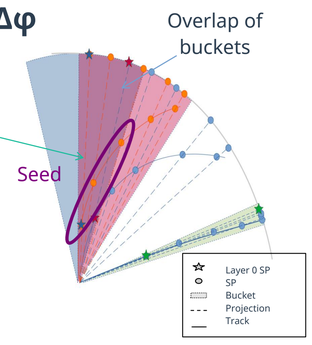
Annoy:
- k Nearest Neighbors (unsupervised)
- Random based
- Parameter: Number of Neighbors (bucket size)
- Use the distance between the points

→ **need a metric**



Space separation

Metric: Δφ

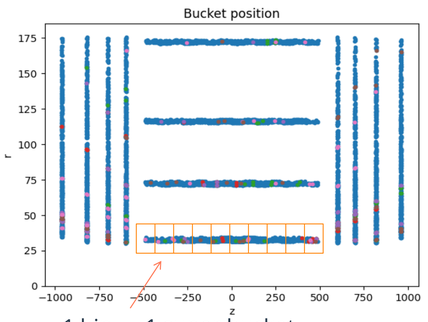


✘ Hashing introduces **overlaps**:
The same seed can be reconstructed in several buckets (14 times in average)

μ = 150	Timing/event (ms)
Without Hashing	4491
With Hashing	7909

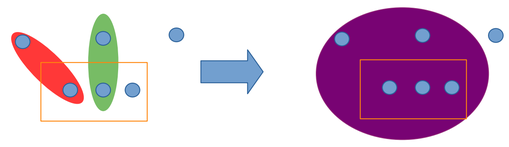
Hashing made timing x2

Group buckets

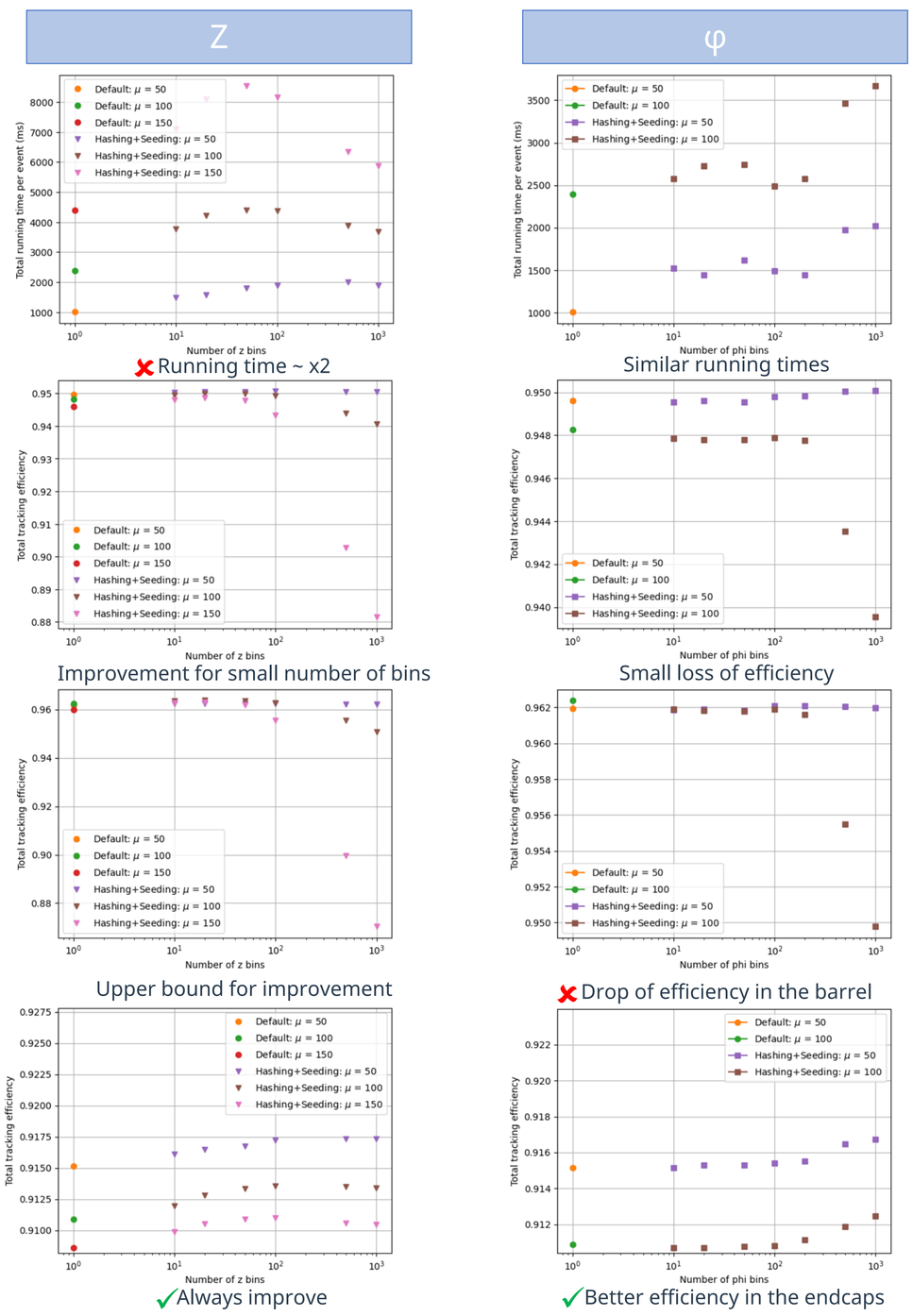


1 bin → 1 super bucket

✓ **Super bucket**:
Merging of the buckets created from the space points inside the bin



5. Results



✘ Running time ~ x2

Similar running times

Improvement for small number of bins

Small loss of efficiency

Upper bound for improvement

✘ Drop of efficiency in the barrel

✓ Always improve

✓ Better efficiency in the endcaps

Slower but better
Can be better in every regions

Not as good but as fast
Better in forward region

Conclusion

✓ **Comparable performance with the baseline (slight improvement in the forward region!)**
✓ **Trade off between timing and performance (Z binning vs φ binning)**
✘ **Not better timing than baseline for now**

Future work

- **Full sim with Geant4 (secondaries)**
- **ITk**
- **metric learning**
- **binning in (z, φ)**
- **changing bucket size with detector region (or different algorithms)**
- **Track finding and seeding on the buckets**

Projet ANR-21-CE31-0022