Tracking with Hashing Overview

All work

6th November 2024

Jeremy Couthures

Outline

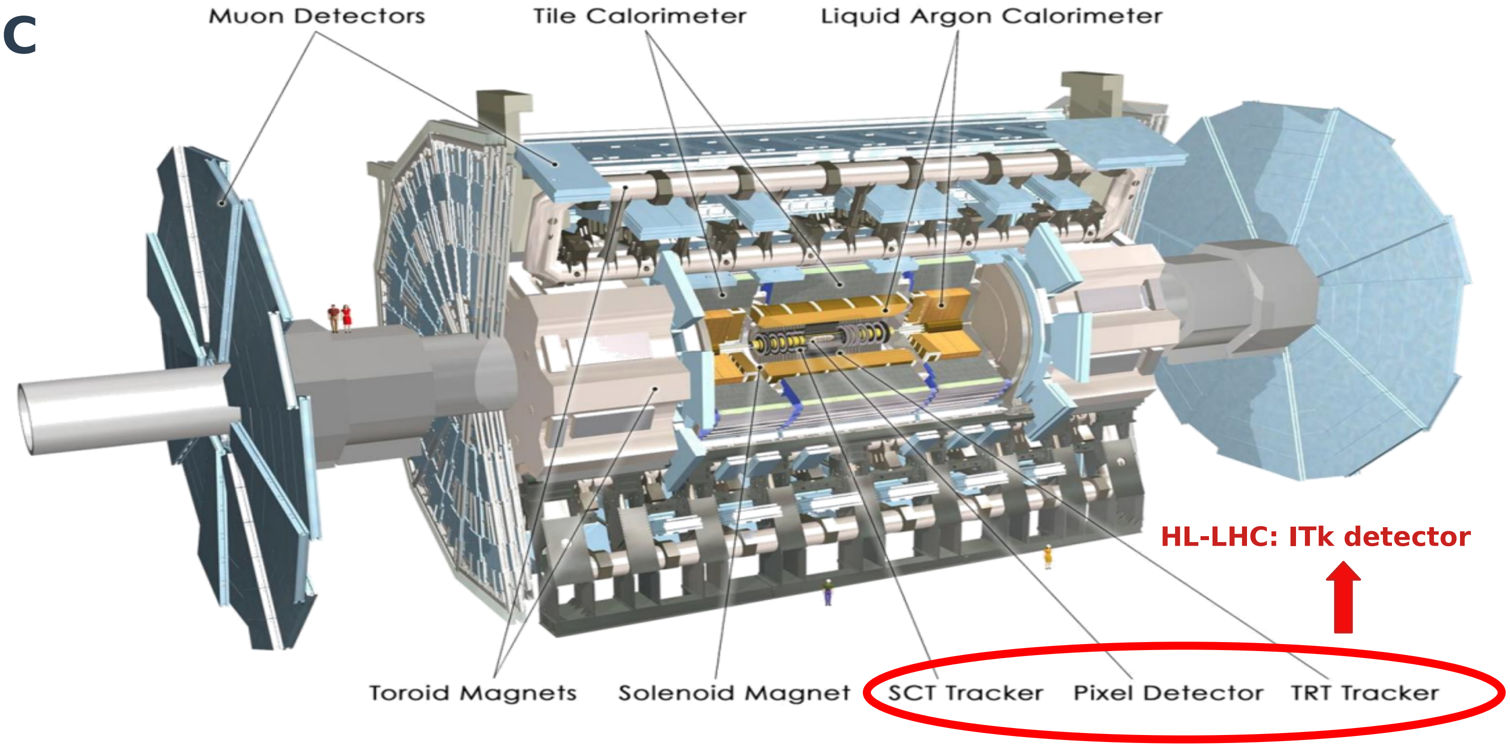
1. Inner Tracker building
2. ATLAS Qualification Task
3. ATLAS Tracking
4. Hashing in ACTS
5. Hashing in Athena
6. Metric Learning
7. Interpretability
8. Formations

INNER TRACKER BUILDING

ATLAS detector for HL-LHC

High Luminosity-LHC (HL-LHC):

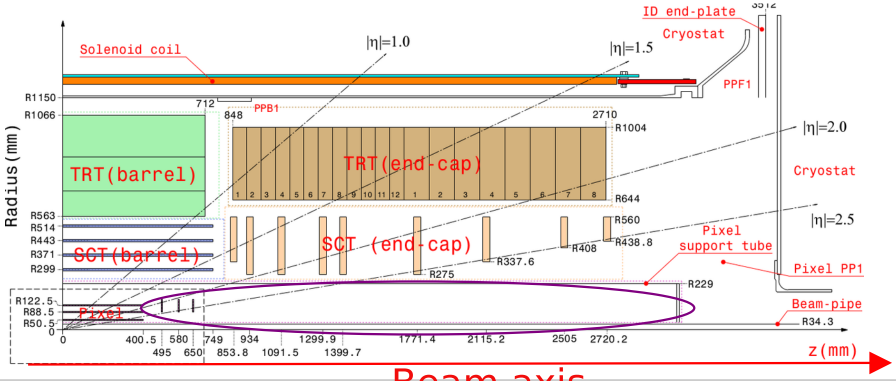
- **Expected in 2029**
- **Increase of luminosity**
  - Luminosity: ~ number of collisions per seconds

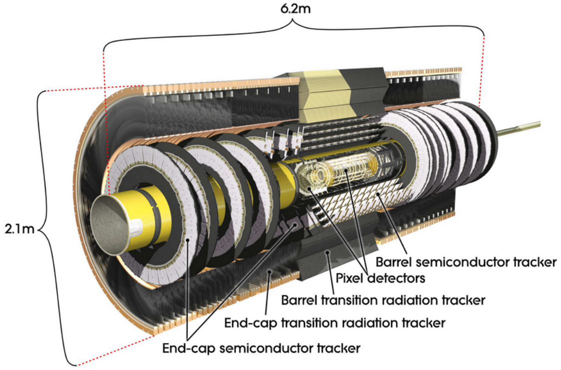


Inner Detector Upgrade

Current Inner detector

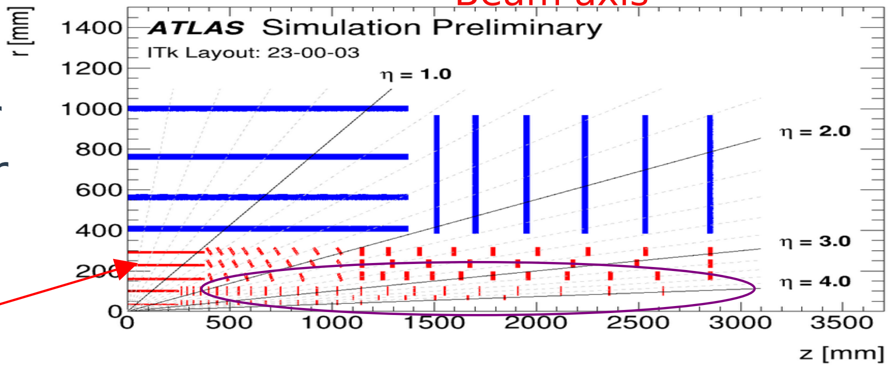
Smallest Pixel size:
50 x 250 µm²

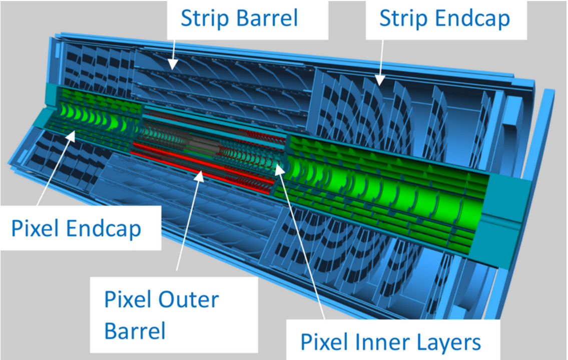




Inner Tracker (ITk) detector

Pixel size:
50 x 50 µm²





ITk: Wider coverage: |η| < 4
Higher granularity

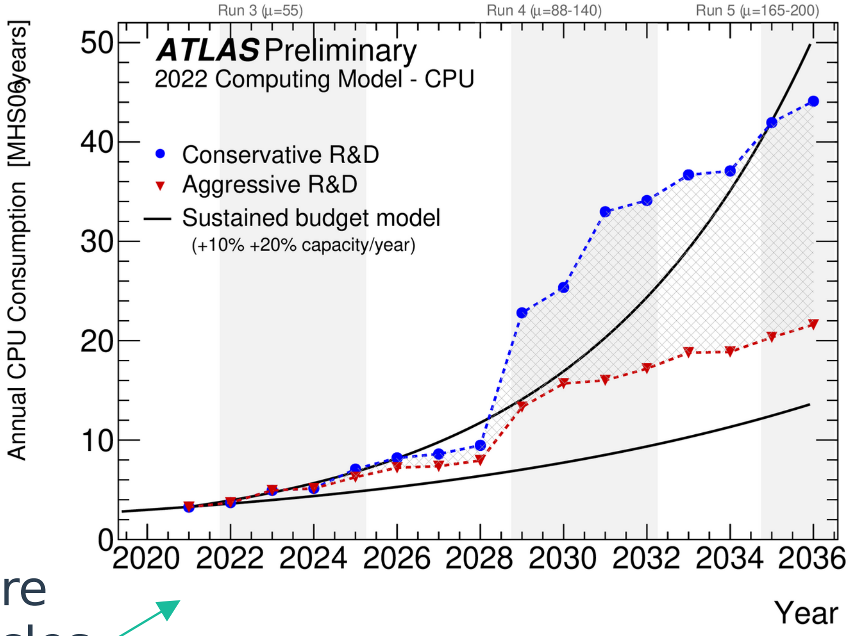
Inner Tracker (ITk) for HL-LHC

- **ITk:**
  - Wider coverage: |η| < 4
  - Higher granularity

More particles detected

- **High Luminosity-LHC (HL-LHC):**
  - Between now and 2029: Peak luminosity x2.5
    - Collisions piles up in an event
    - Pile-up (<μ>): average number of collisions in an event

More particles created

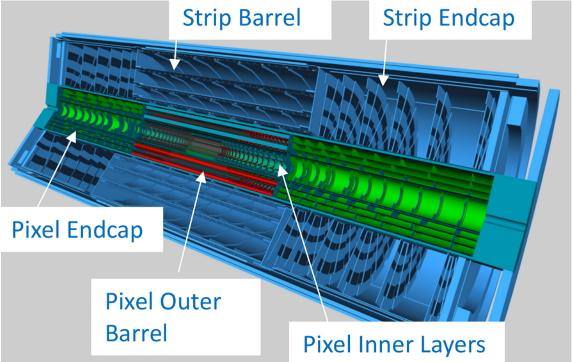


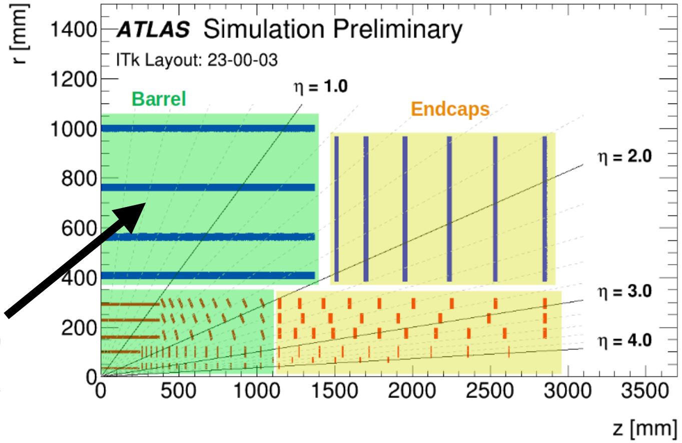
ATLAS CPU previsions: need to improve ***tracking*** performance significantly

ATLAS QUALIFICATION TASK

Inner Tracker building at LAPP

LAPP is producing 75% of the OB Types 0 (5000 pigtails, 400 PP0 boards) and will be integrating 25% of the local supports(*)





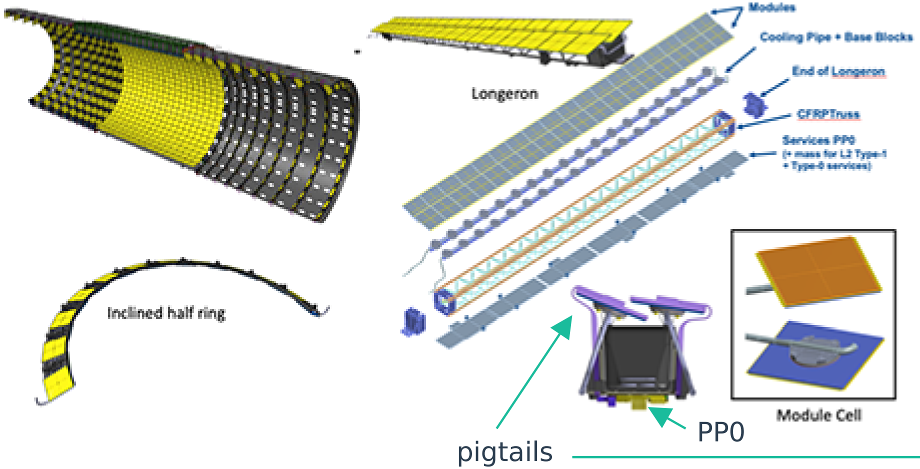
Types 0:
Components directly on the detector

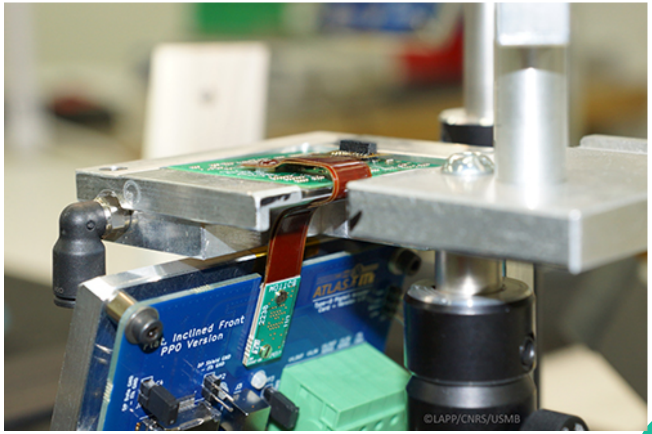
LAPP
Laboratoire d'Annecy de Physique des Particules

(*)With LPSC and CPPM

Inner Tracker Pixel Detector Overview

- **Pigtails: Power supply, monitoring of the cell and transmit data from the module cell**
- **Patch Panel 0 (PP0): Distribute power supply and aggregate data**





https://lapp.in2p3.fr/spip.php?article3307

ATLAS Qualification Task: Production Database

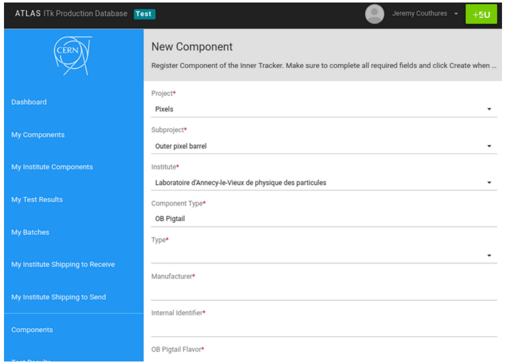
- **ATLAS Production Database**
  - Create components, store quality control data, track shipping, API

- **Qualification Task:**
  - Creation of a dedicated “LAPP Types 0 Web app”
    to improve data registration in the database, robustness and scalability

Registration in the database

Registration speed

1 to 1	1 to many	No operator
Low level UI	Web app	Web app
Fields and buttons	Fields and buttons	JSON files from LabVIEW



Streamlit

Streamlit

LabVIEW

QT Supervision

Repository

GitLab

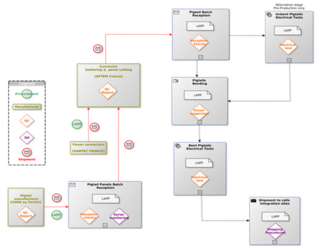
Interface

Streamlit

Hosting

okd

API



Production flows

Database

CERN
ITk Production Database

Monika Wieler

Kenneth Wraight

Jessica Levêque

Renaud Gaglione

Users

LAPP

Andrea Jeremie
Francesco Costanza

Dedicated web app:

Forked from itk-web-apps

Pigtails production flow

Similar production flow for PP0

BAD_HV_TEST
BAD_FROM_ELECTRICAL_TEST_DATA_TRANSMISSION
BAD_FROM_ELECTRICAL_TEST_RESISTANCE

Electrical Tests

Pass/Fail

Unbent Electrical Tests

pre-production

BAD_HV_TEST
BAD_FROM_ELECTRICAL_TEST_DATA_TRANSMISSION
BAD_FROM_ELECTRICAL_TEST_RESISTANCE

Pass/Fail

Reception before cabling

Visual Inspection

Pass/Fail+properties set

BAD_FROM_MANUFACTURER

Reception after cabling

Visual Inspection

Pass/Fail+info

BAD_FROM_CABLING_VI

production

Bending

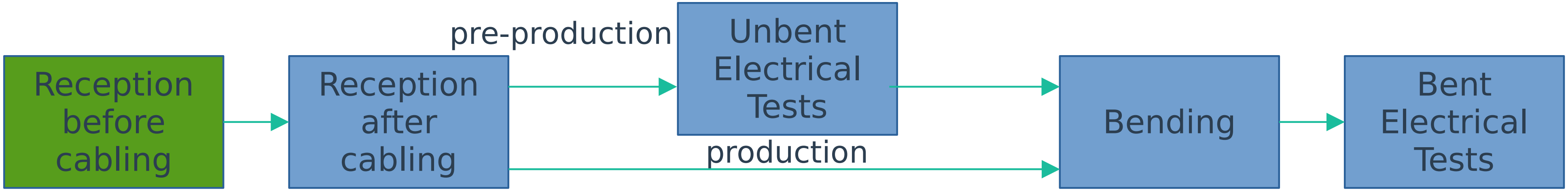
Visual Inspection

Pass/Fail+properties set

BAD_BENDING

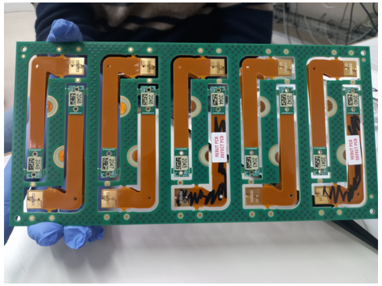
Bent Electrical Tests

Electrical Tests



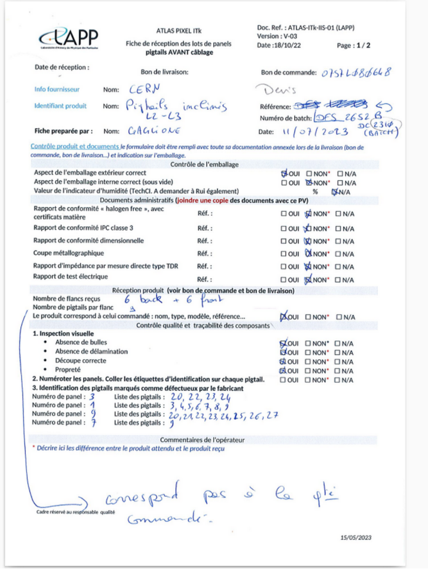
Creation of the pigtails in the database:

- Panel level comment? Pigtail marked as bad? From which panel?



Object

Form



LAPP — ATLAS PIXEL ITk

Fiche de réception des lots de panels **pigtails AVANT câblage**

Doc. Ref. : ATLAS-ITk-IIS-01 (LAPP)
Version : V-03
Date :18/10/22 Page : 1 / 2

Date de réception :

Bon de livraison:

Bon de commande: 0757L080648

Info fournisseur Nom: CERN — Denis

Identifiant produit Nom: Pigtails inclinés L2 - L3

Référence: [illegible]

Numéro de batch: DFS 2652 B — DC 2314 (BATCH)

Fiche preparée par : Nom: GAGLIONE

Date: 11/07/2023

Contrôle produit et documents le formulaire doit être rempli avec toute sa documentation annexée lors de la livraison (bon de commande, bon de livraison…) et indication sur l'emballage.

Contrôle de l'emballage		
Aspect de l'emballage extérieur correct		☒ OUI ☐ NON* ☐ N/A
Aspect de l'emballage interne correct (sous vide)		☐ OUI ☒ NON* ☐ N/A
Valeur de l'indicateur d'humidité (TechCl. A demander à Rui également)		% ☒ N/A
Documents administratifs (joindre une copie des documents avec ce PV)		
Rapport de conformité « halogen free », avec certificats matière	Réf. :	☐ OUI ☒ NON* ☐ N/A
Rapport de conformité IPC classe 3	Réf. :	☐ OUI ☒ NON* ☐ N/A
Rapport de conformité dimensionnelle	Réf. :	☐ OUI ☒ NON* ☐ N/A
Coupe métallographique	Réf. :	☐ OUI ☒ NON* ☐ N/A
Rapport d'impédance par mesure directe type TDR	Réf. :	☐ OUI ☒ NON* ☐ N/A
Rapport de test électrique	Réf. :	☐ OUI ☒ NON* ☐ N/A
Réception produit (voir bon de commande et bon de livraison)		
Nombre de flancs reçus	6 back + 6 front	
Nombre de pigtails par flanc	[illegible]	
Le produit correspond à celui commandé : nom, type, modèle, référence…		☒ OUI ☐ NON* ☐ N/A
Contrôle qualité et traçabilité des composants		
1. Inspection visuelle		
• Absence de bulles		☒ OUI ☐ NON* ☐ N/A
• Absence de délamination		☒ OUI ☐ NON* ☐ N/A
• Découpe correcte		☒ OUI ☐ NON* ☐ N/A
• Propreté		☒ OUI ☐ NON* ☐ N/A
2. Numéroter les panels. Coller les étiquettes d'identification sur chaque pigtail.		☐ OUI ☐ NON* ☐ N/A

3. Identification des pigtails marqués comme défectueux par le fabricant

Numéro de panel : 3 Liste des pigtails : 20, 22, 23, 24
Numéro de panel : 1 Liste des pigtails : 3, 4, 5, 6, 7, 8, 9
Numéro de panel : 9 Liste des pigtails : 20, 21, 22, 23, 24, 25, 26, 27
Numéro de panel : 7 Liste des pigtails : 9

Commentaires de l'opérateur

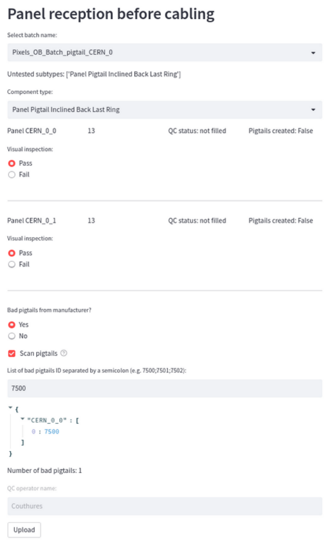
* *Décrire ici les différence entre le produit attendu et le produit reçu*

Correspond pas à la qté commandée.

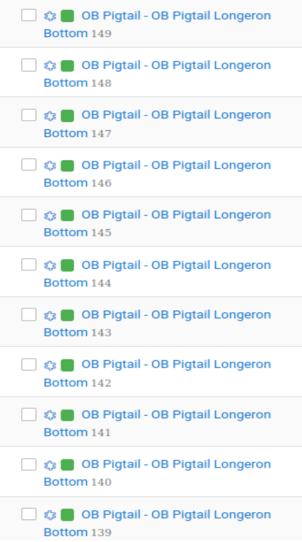
Cadre réservé au responsable qualité

15/05/2023

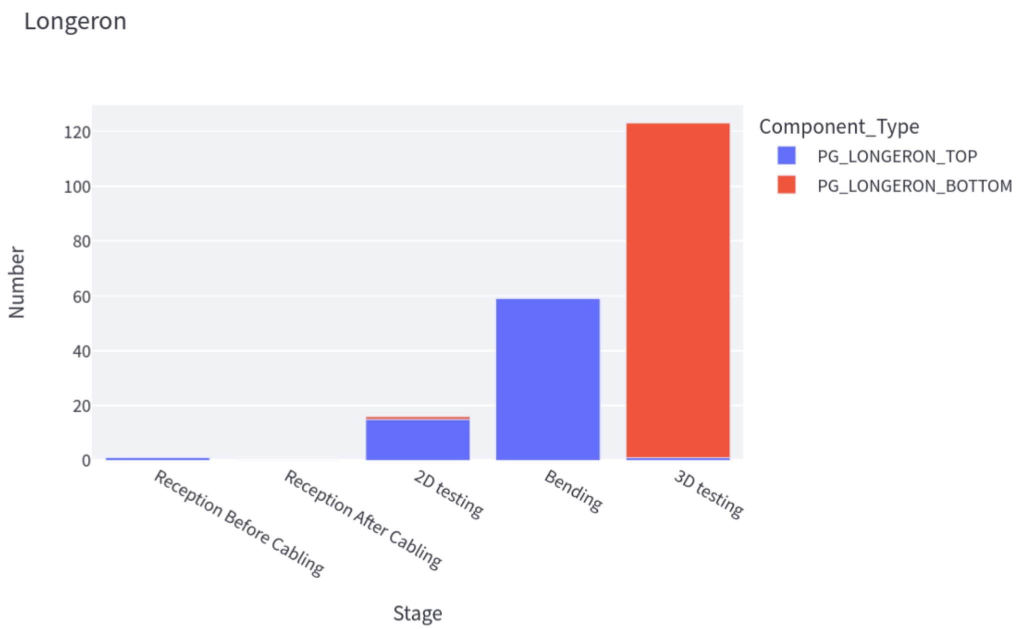
Web app

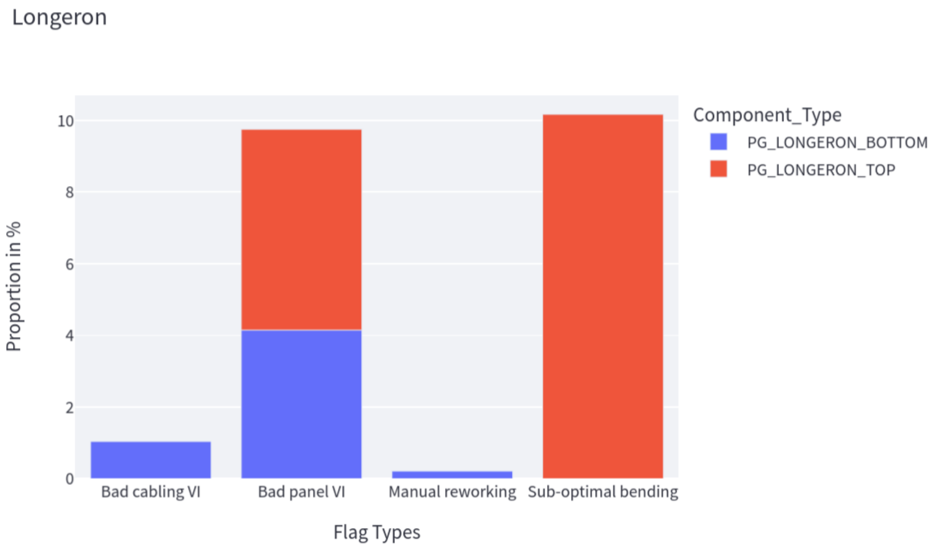


Database



Reporting





Plots not possible without the web app

ATLAS Qualification Task: Types 0 web app

Qualified since January 2024

Code on gitlab:
gitlab repository

Web app link:
https://itk-web-apps-pigtails.app.cern.ch/

QT presentation link:
indico link

Select component type:

OB_PIGTAIL

Select stage:

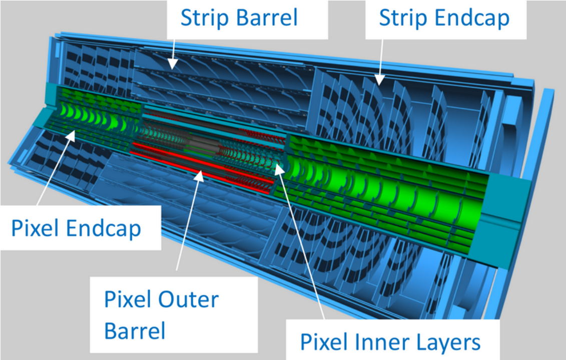
Reception Before Cabling

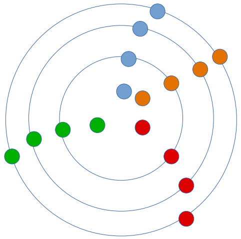
☐ Remove flagged components

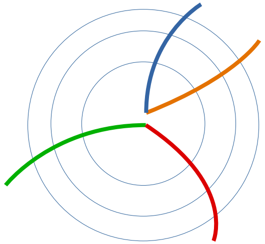
Internal ID	Type	Stage	Link to PDB
lapp_2152	OB Pigtail Longeron Top	Reception Before Cabling	e90a6f1259e399ca44a7f078b5732d76
lapp_2148	OB Pigtail Longeron Top	Reception Before Cabling	86945b16501f77d00332c2244c852355
lapp_2142	OB Pigtail Longeron Top	Reception Before Cabling	f4c2488adcb76d3ad05b05b7fea5f632
lapp_2135	OB Pigtail Longeron Top	Reception Before Cabling	0d56d4136c8b65f739d2d0ad5fcd983b
lapp_2133	OB Pigtail Longeron Top	Reception Before Cabling	2cad78db9727bc97d0f19d78ec8a6f43
lapp_2123	OB Pigtail Longeron Top	Reception Before Cabling	bb4494d8125de87f1ba424f25f0073e8
lapp_2121	OB Pigtail Longeron Top	Reception Before Cabling	6b426487f703254c0a2ad6e9a00e7b8d

ATLAS TRACKING

ATLAS Tracking simplified







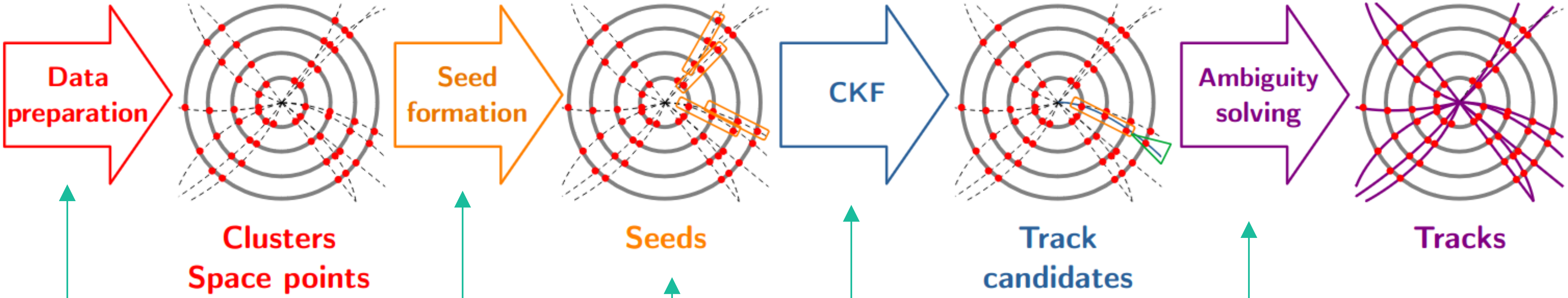
InnerTracker (ITk)

Hits / Space points

Tracks

ATLAS Detector at
High Luminosity LHC

ATLAS Tracking less simplified



Transform detector hits into clusters and space points

Group compatible triplets of space points

Seed:
3 space points

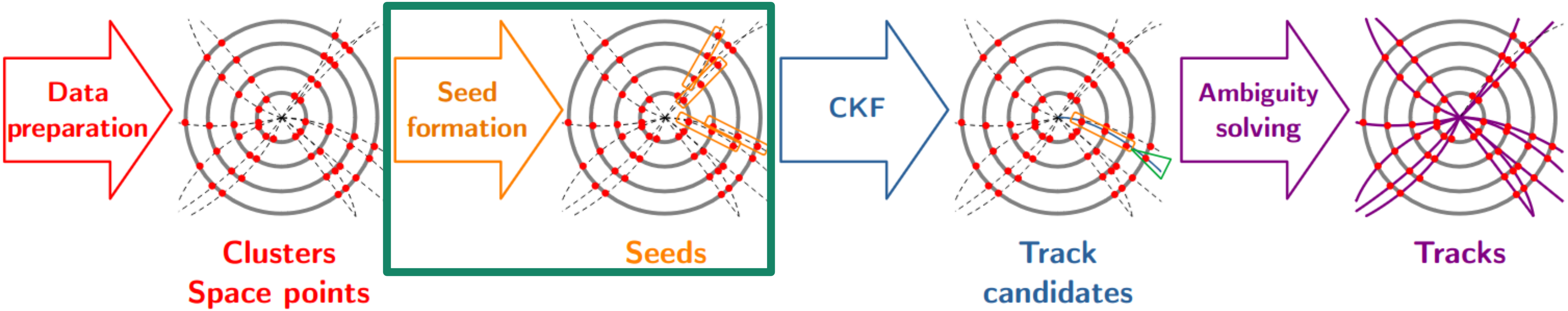
Expand the seeds with remaining space points

Track:
arbitrary number of space points

Remove duplicated tracks and bad ones

Image made by Noemi Calace

Focus on Seeding



- **What do we hope to improve?**
  - Seeds' efficiency: reconstruct at least one seed per track
  - Seeds' purity (fake rate): reconstruct only tracks' seeds
  - Seeds' redundancy (duplication rate): reconstruct just enough seeds per track

Seeding Algorithm steps

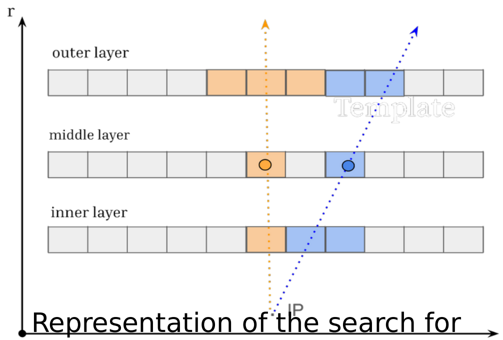
1. Seed Finder

- Check if the triplet forms a nearly straight line in the (r,z) plane

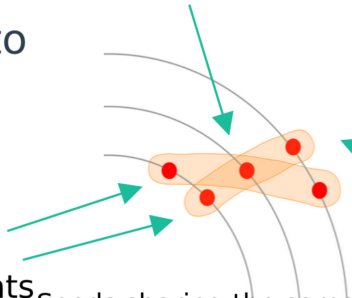
2. Seed Filter

- maxSeedPerSpM cut limits the number of seeds to speed up the tracking

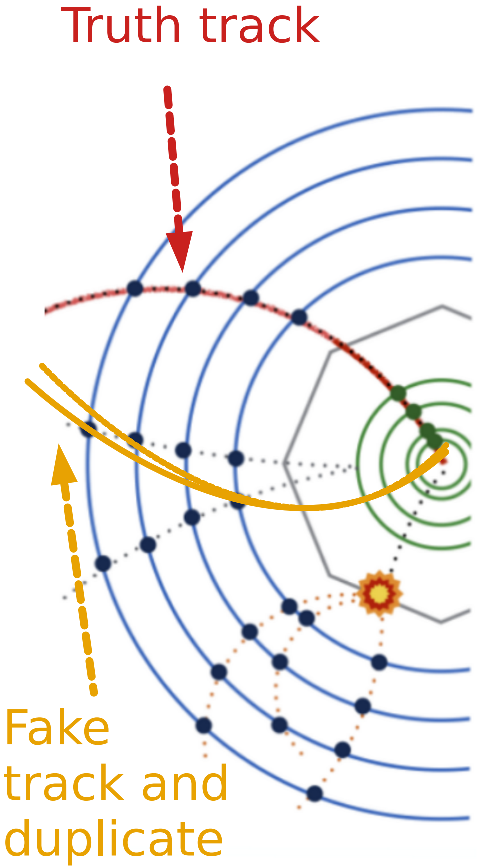
- **Possible improvement:**
  - maxSeedPerSpM: Non physical cut
    → can remove good seeds
- **Can we remove it?**



Representation of the search for triplet combinations in the (r, z) plane. The bins used in the search are represented in different colors.



Evaluation

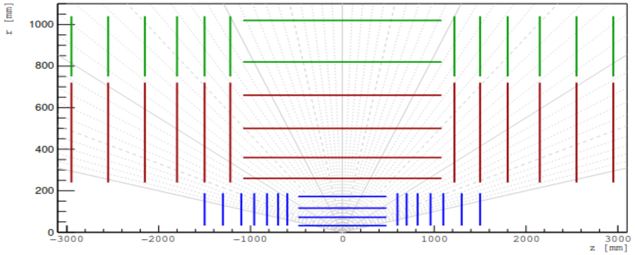


Evaluate on tracks:

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

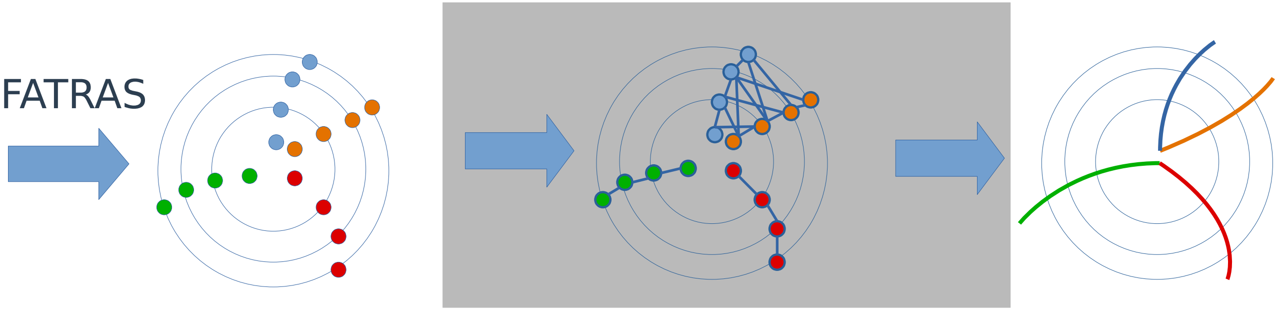
Initial study

Generic detector
(virtual toy detector)



https://arxiv.org/pdf/2105.01160.pdf

acts github



Combinatorics

→ maxSeedsPerSpM=1

Run 4:

<μ> = 140

Pythia8: 100 $t\bar{t}$ events

μ = 50, 100, 150

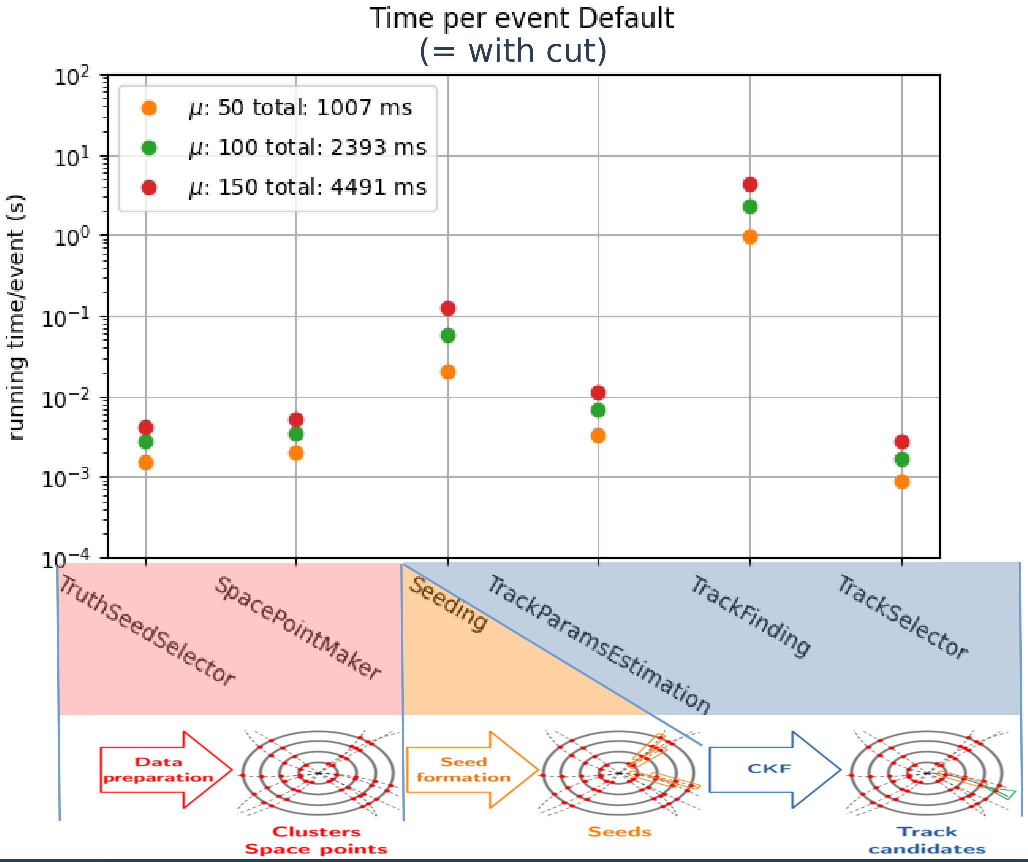
Not using Geant4:

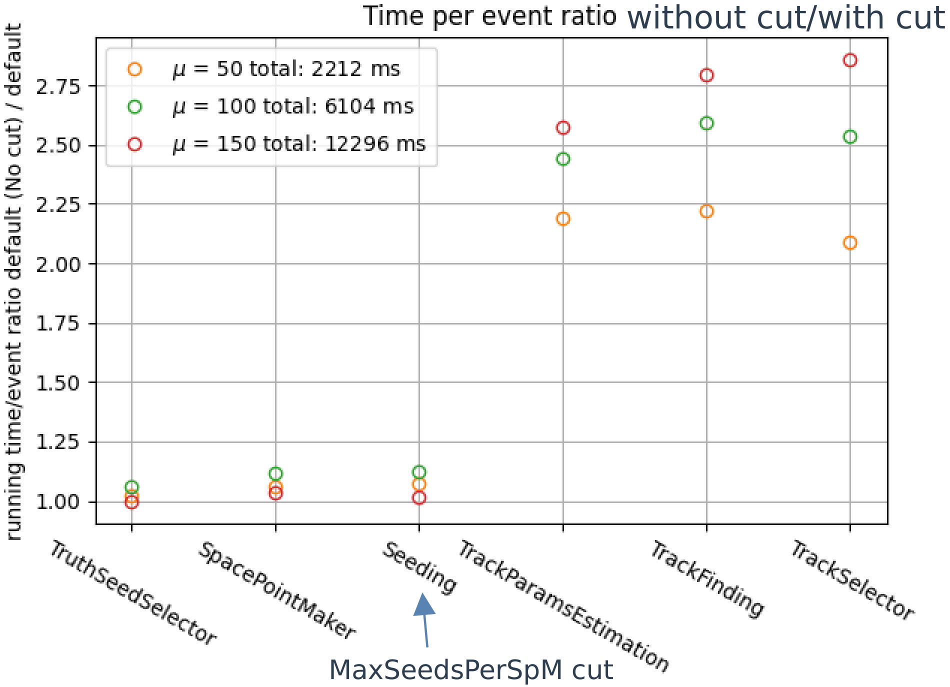
→ no secondaries

$|\eta| \leq 4$

pT > 1GeV

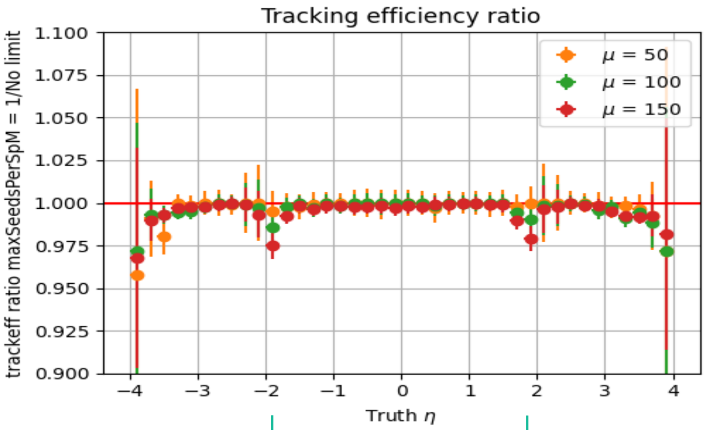
ACTS performance: Timing/event

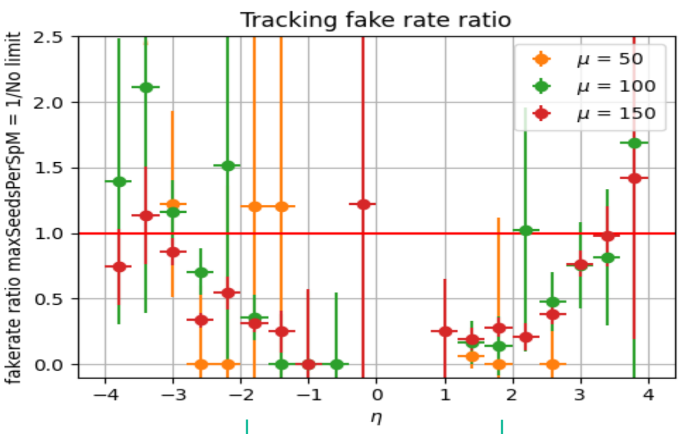


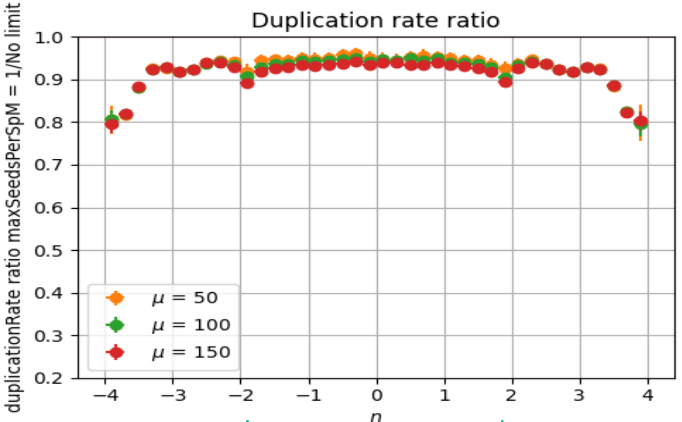


Timing multiplied by ~2-3 without the cut

ACTS performance: Physics







Forward region | Central region | Forward region

Forward region | Central region | Forward region

Forward region | Central region | Forward region

Same in central region,
Lower efficiency in forward region

Less fake tracks in central region,
Same in forward region

Less duplicated tracks,
Even less in forward region

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

Target Goal

- **Without the cut: improve performance <u>but</u> timing is crucial**

- **Goal: Improve performance with same timing**
  - Keep the cut but try to bypass it

HASHING IN ACTS

A new method:
Machine Learning/Hashing in the Seeding

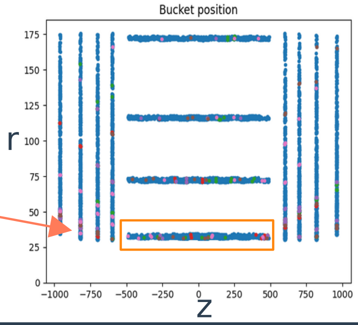
Hashing:
1. Group similar space points into buckets
2. Do the seeding on each bucket

Algorithm used:
Approximate Nearest Neighbors Oh Yeah (**Annoy**)
→ Used by Spotify

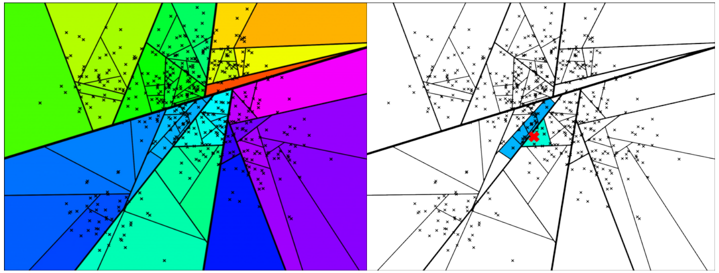
- Machine Learning algorithm type:
  - k Nearest Neighbors (unsupervised)
  - Random based

- Find Neighbors of the points in layer 0

1 space point in layer 0 → 1 bucket



Annoy:



Space separation

Look for neighbors in the closest regions

Parameter: Number of Neighbors (bucket size)

Use the distance between the points
→ **need to define a (relevant) metric**

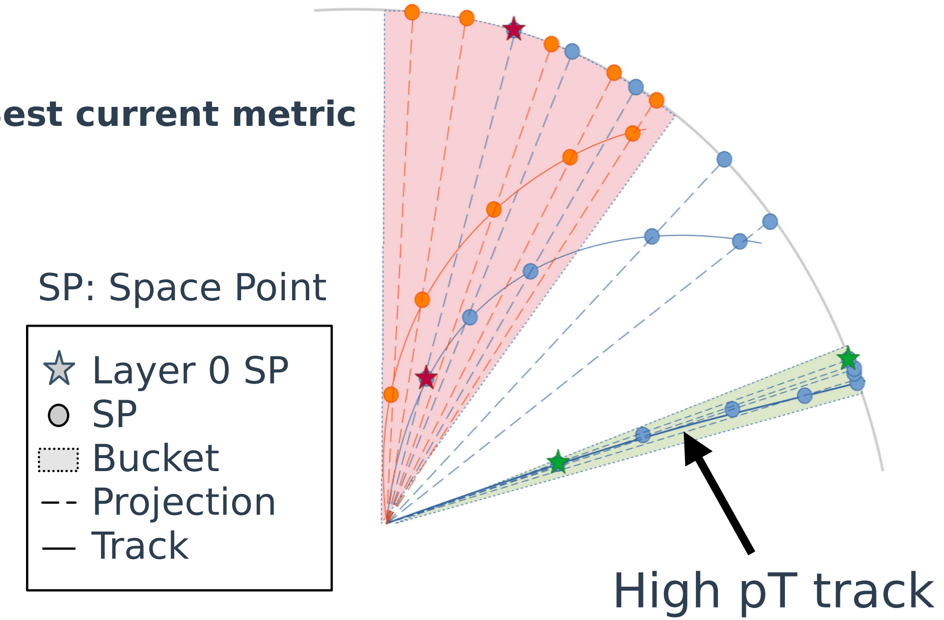
Metric and bucket size

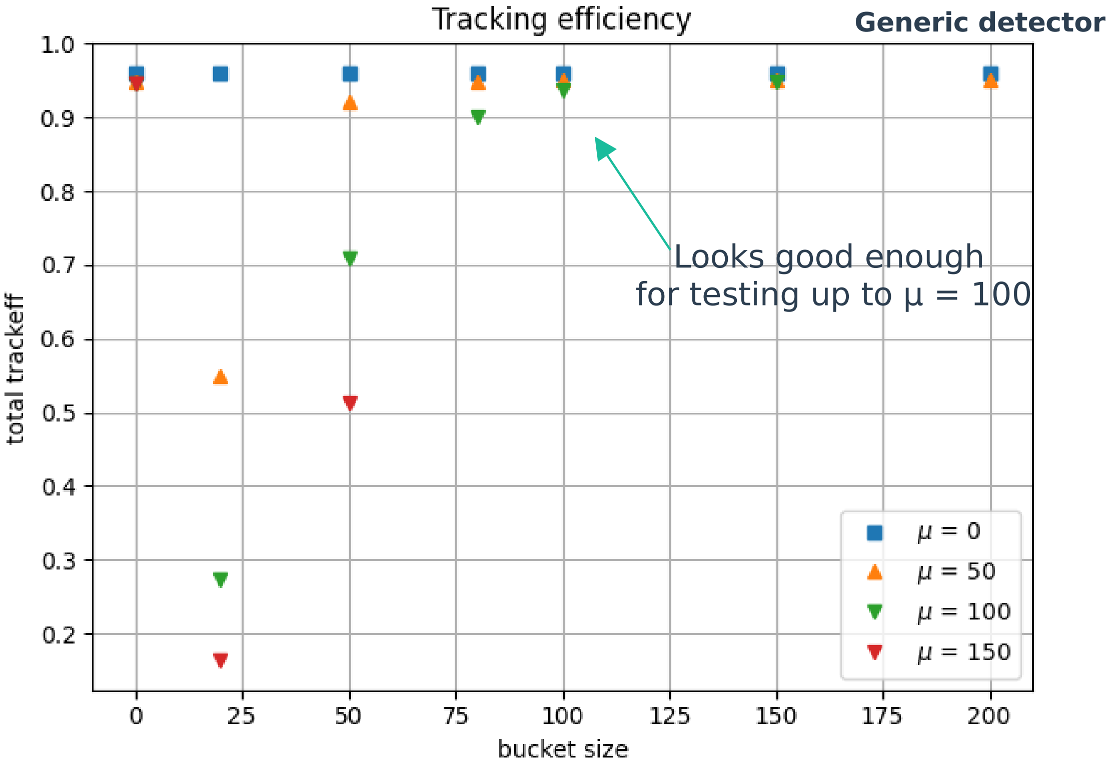
Metric: Δφ

Suitable for high pT tracks

Best current metric

SP: Space Point

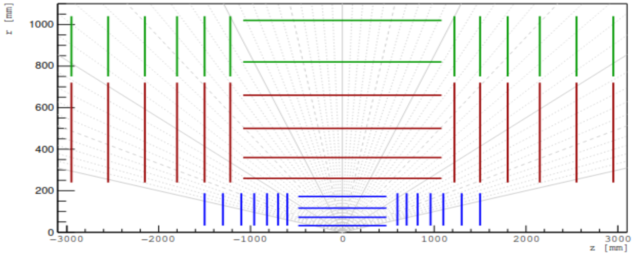




Fix the bucket size to 100

Overview

Generic detector
(virtual toy detector)

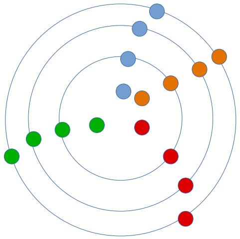


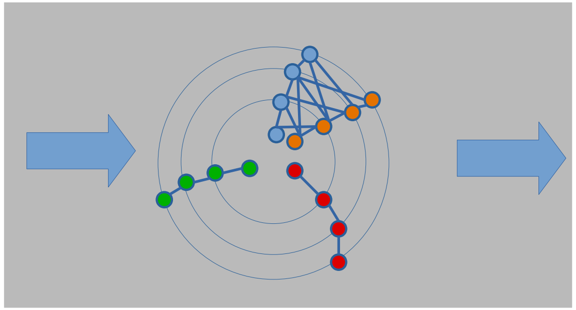
https://arxiv.org/pdf/2105.01160.pdf

Pythia8: 100 $t\bar{t}$ events
μ = 50, 100, 150

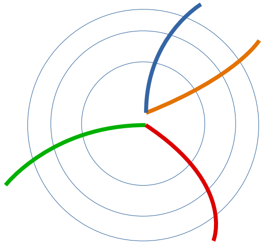
|η| ≤ 4
pT > 1GeV

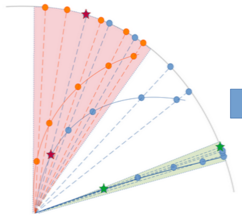
FATRAS



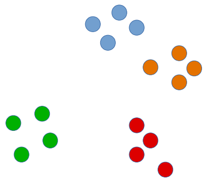


Combinatorics
→ maxSeedsPerSpM=1

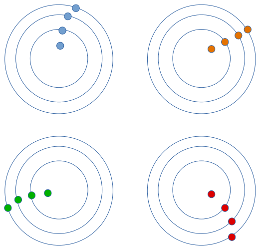




Metric: Δφ

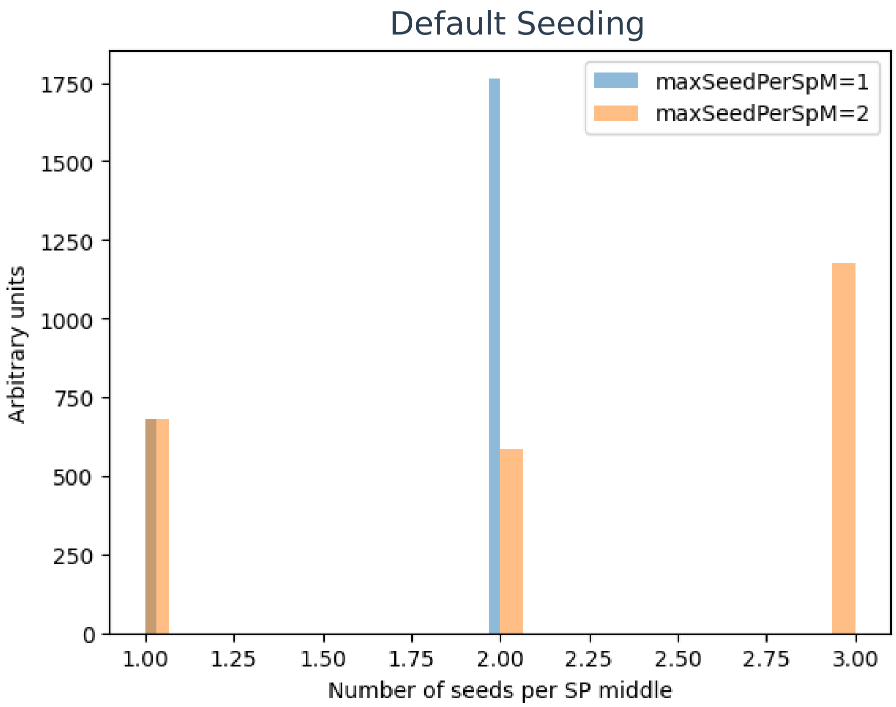


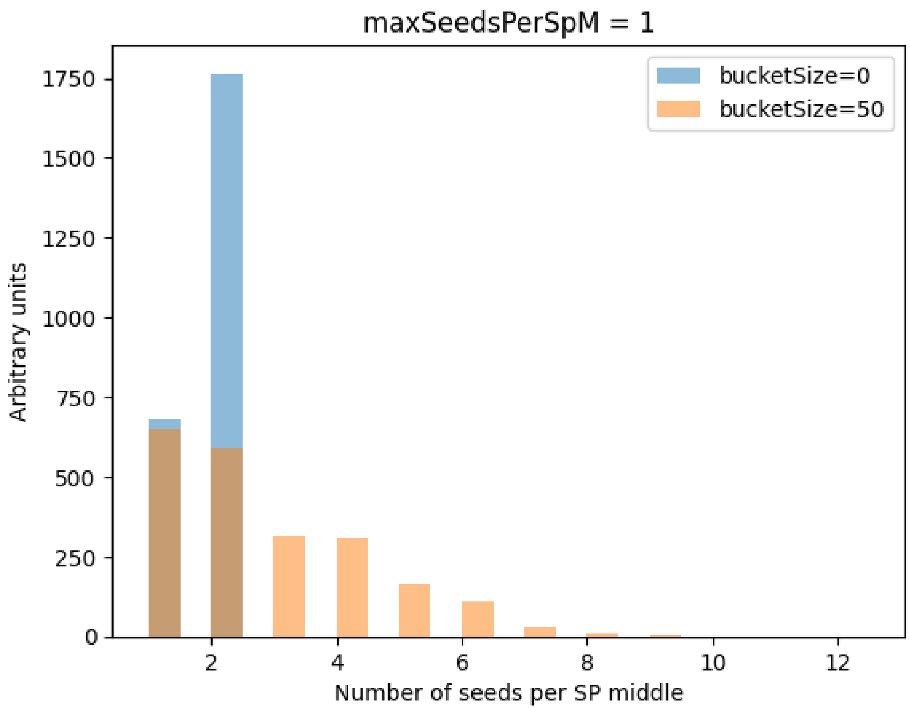
Clustering:
Annoy



Parallelization

MaxSeedsPerSpM cut vs Hashing





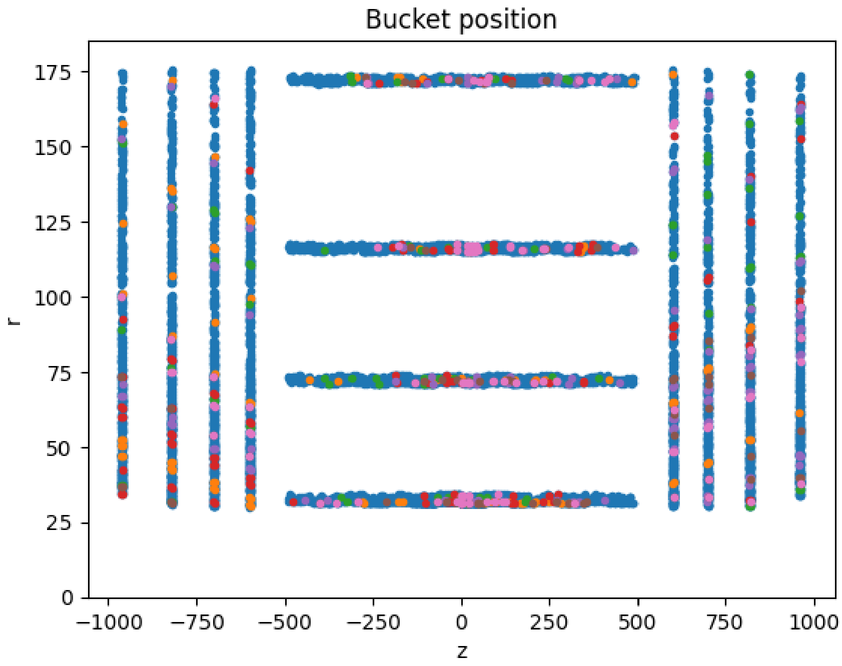
→ Hashing get through the cut

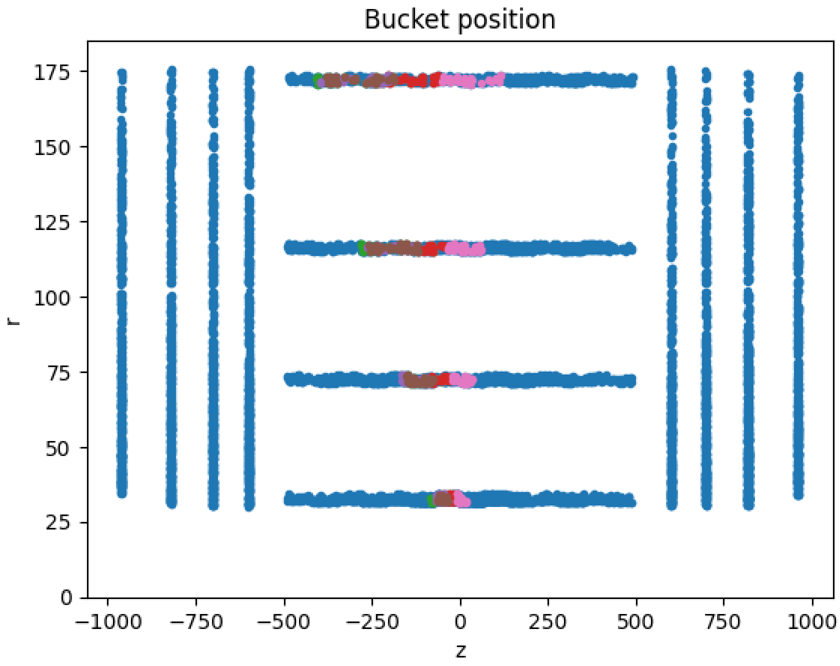
Other metric: ΔR

Angular: Δφ

$\Delta R = \sqrt{(\Delta\varphi^2 + \Delta\eta^2)}$

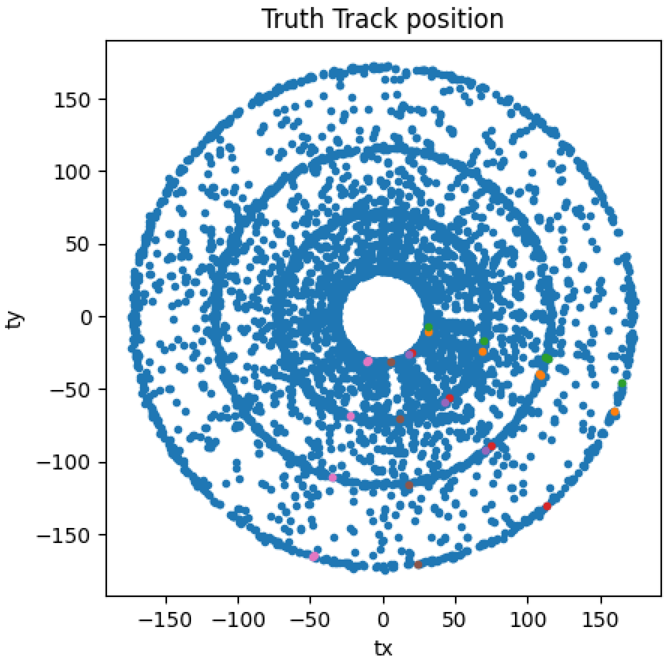
If $\Delta\varphi > \pi$:
$\Delta\varphi = 2*\pi - \Delta\varphi$



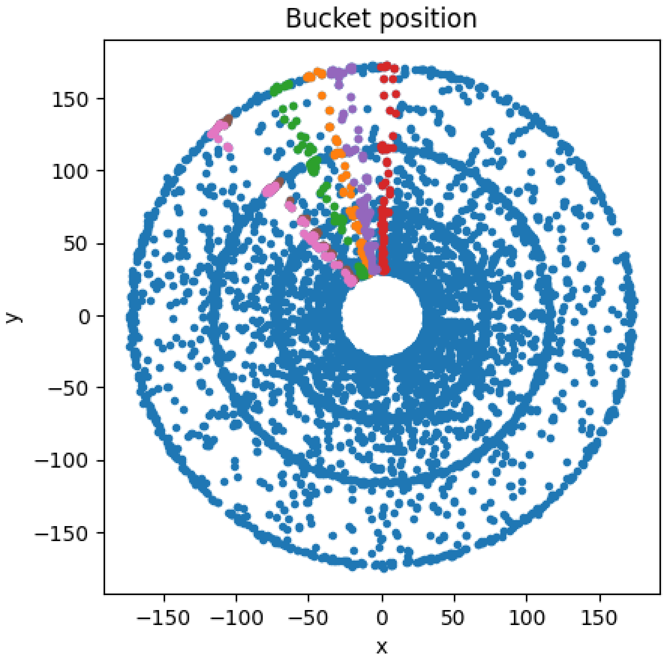


Comparison (x,y) plan

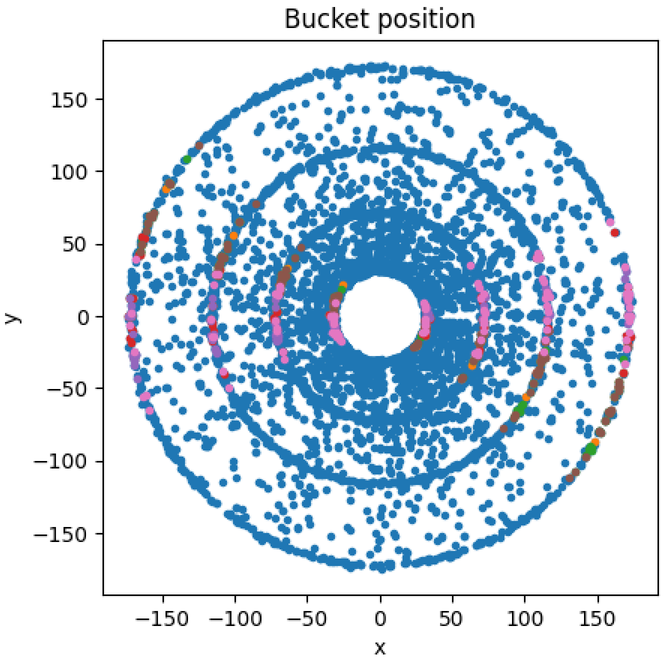
Truth Tracks (hits)



Angular: Δφ

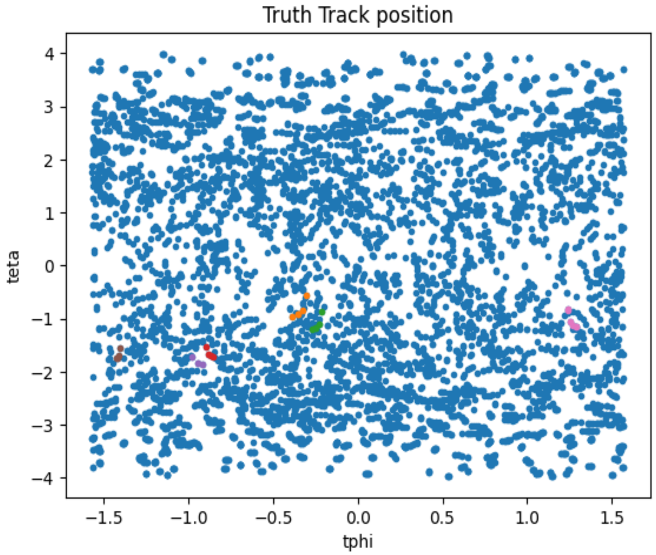


ΔR

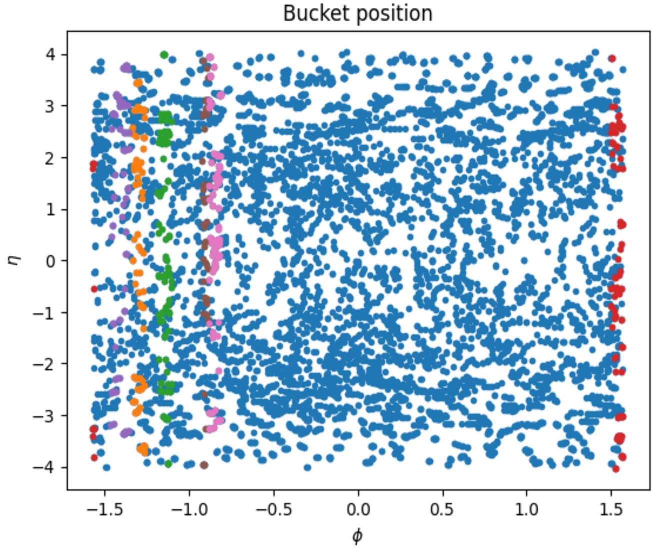


Comparison (φ,η) plan

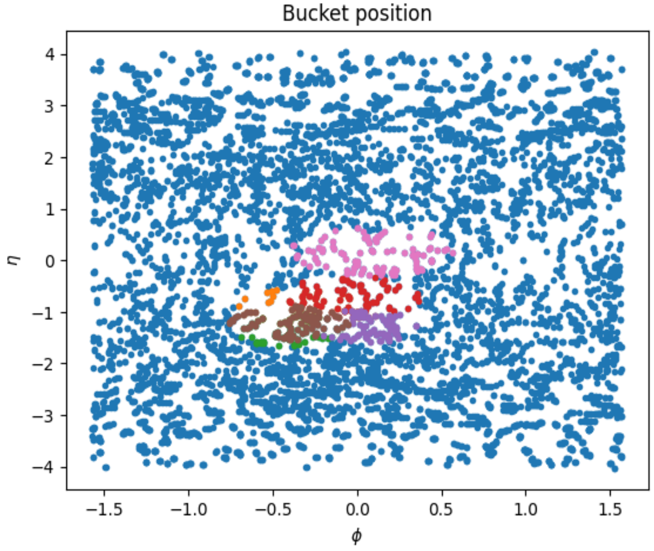
Truth Tracks (hits)



Angular: Δφ

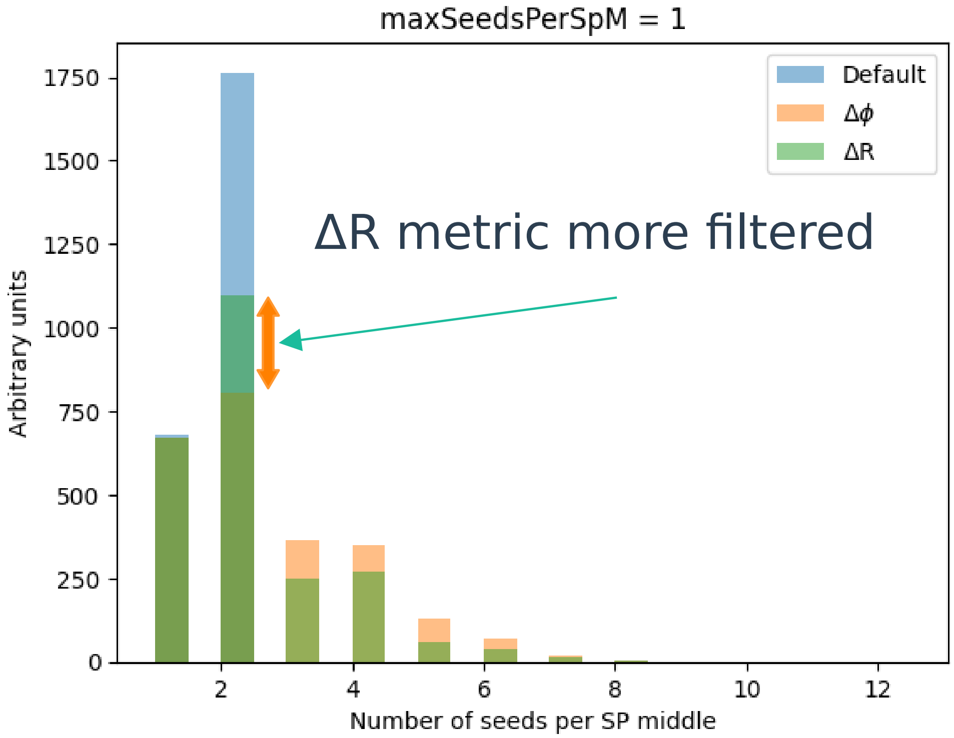


ΔR



MaxSeedsPerSpM and ΔR metric

On 1 event:



Filtered Middle Space points are on the maxSeedsPerSpM bin

Some of the “Buckets shared Middle Space points” are on the bins after the maxSeedsPerSpM bin

Differences in the bins before maxSeedsPerSpM correspond to lost seeds

Default nSeeds: 4208
Δφ nSeeds: 6053
ΔR nSeeds: 5300

Hashing and overlap

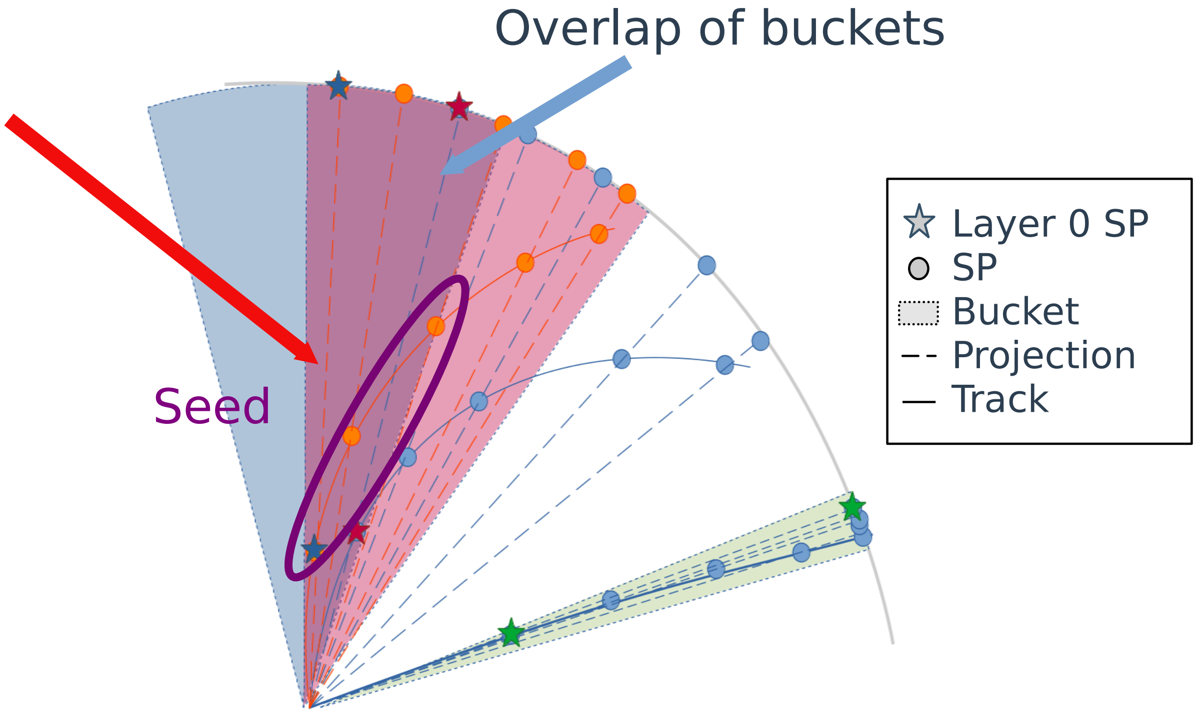
Hashing introduces overlaps:

- The same seed can be reconstructed in several buckets (14 times in average)

Generic detector

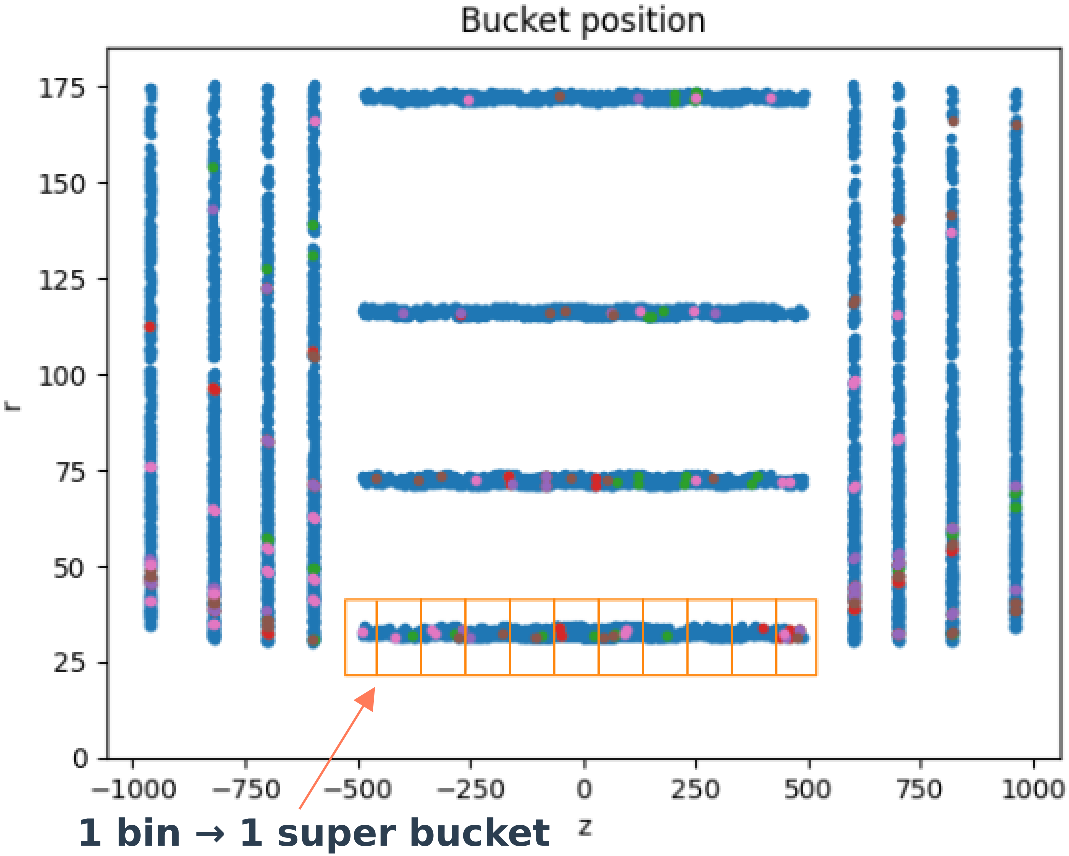
μ = 150	Timing/ event (ms)
Without Hashing	4491
With Hashing	7909

Hashing made timing x2



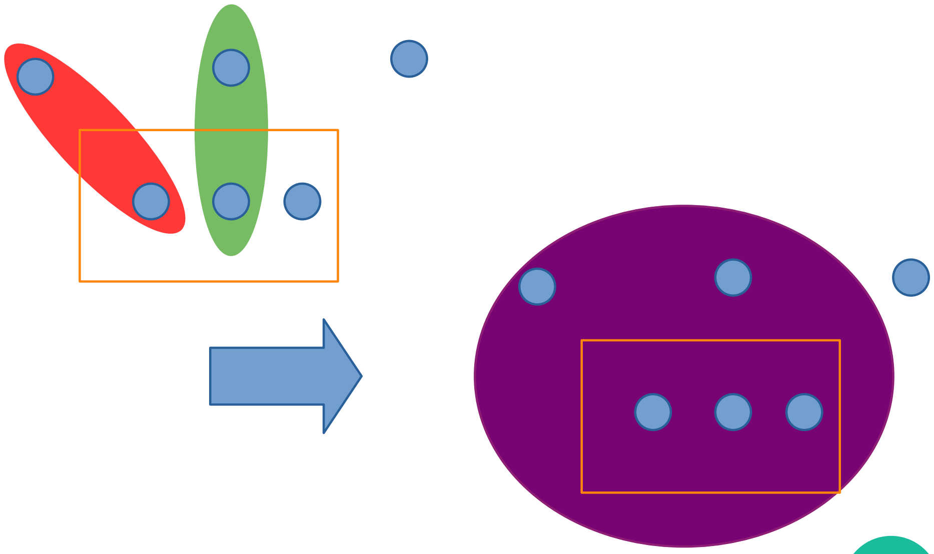
New idea: Group buckets
→ less overlap

Super buckets and binning



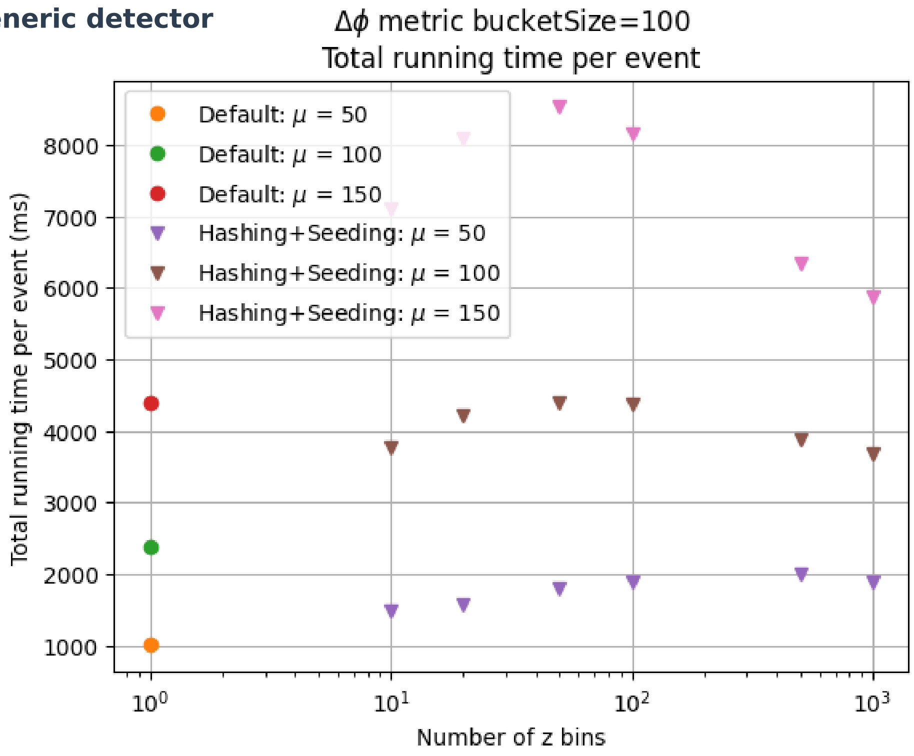
1 bin → 1 super bucket

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

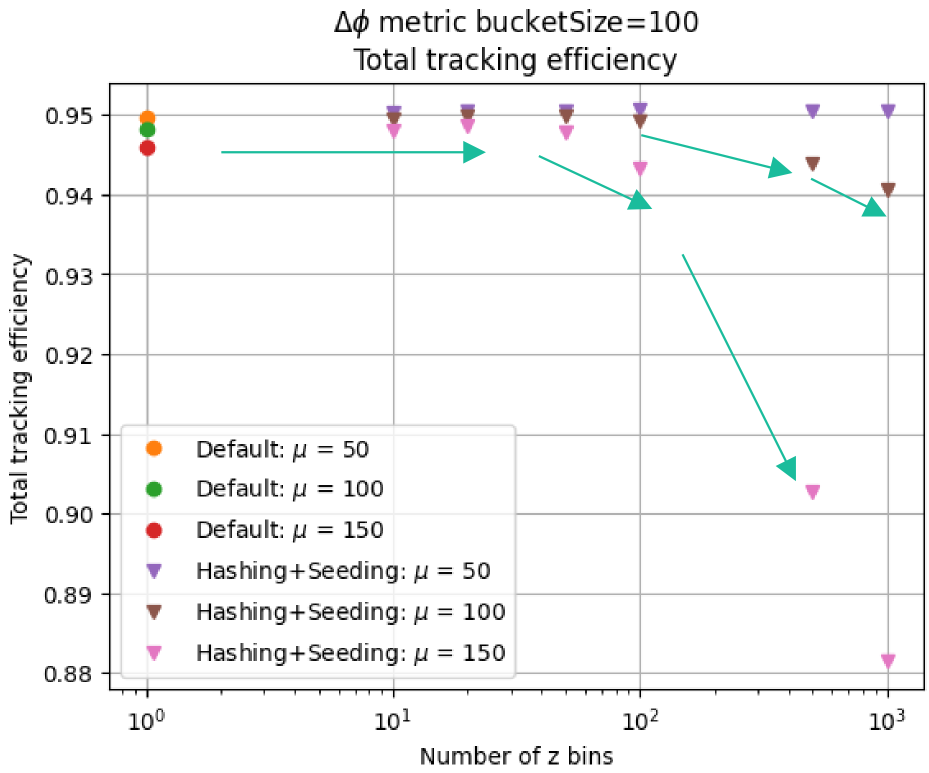


Hashing performance: Timing and efficiency

Generic detector



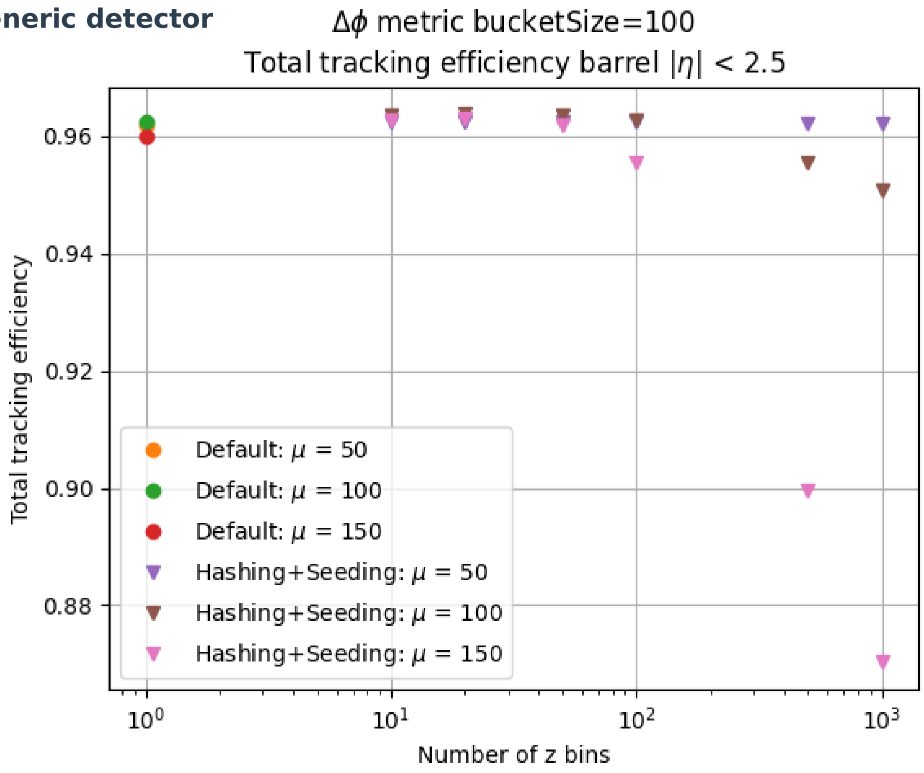
Running time ~ x2



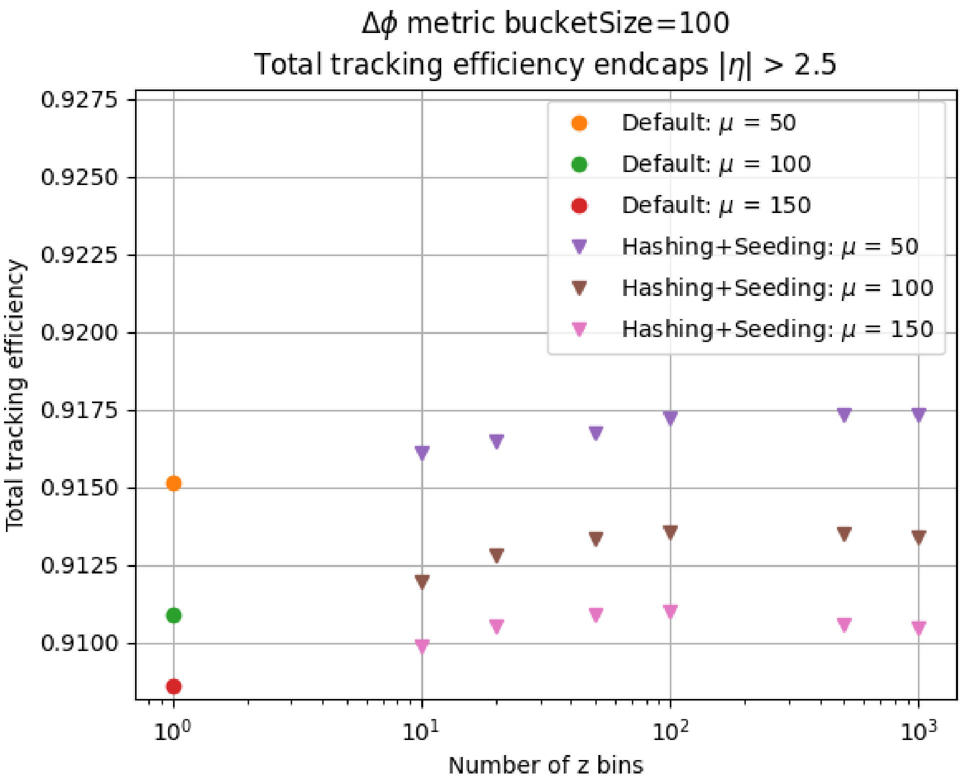
Improvement for small number of bins

Hashing performance: Efficiency (detailed)

Generic detector



Improves then drops



Always improve

Initial study summary

- **Current state:**
  - Slight physics improvement in the forward region with Generic detector
  - Hashing code as an ACTS example

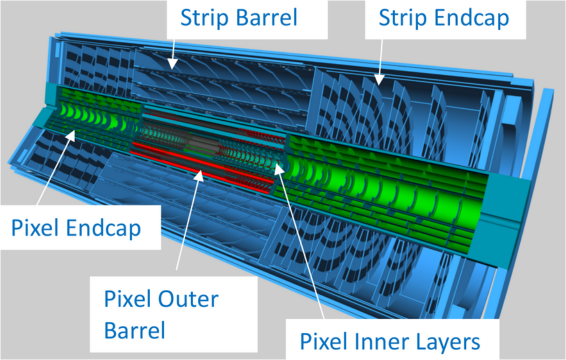
Next:

  - More realistic case

HASHING IN ATHENA

Realistic case

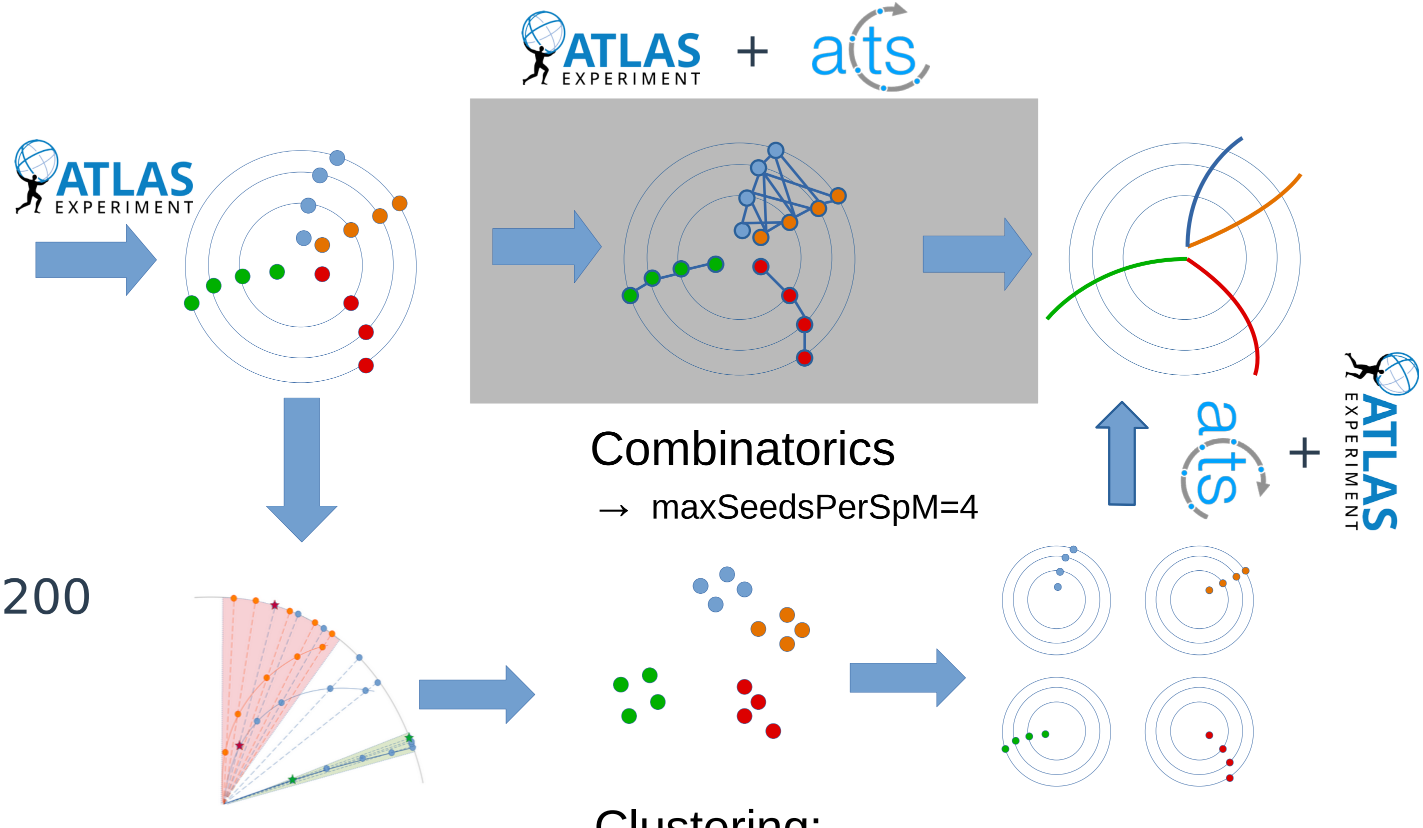
Inner Tracker (ITk)
(beeing built)



HL-LHC:

$\langle\mu\rangle = 200$

Official simulations
1000 $t\bar{t}$ events
$\mu = 0, 60, 140, 200$



Metric: Δφ

Dataset

Remade the official datasets to have also pile-up truth information:

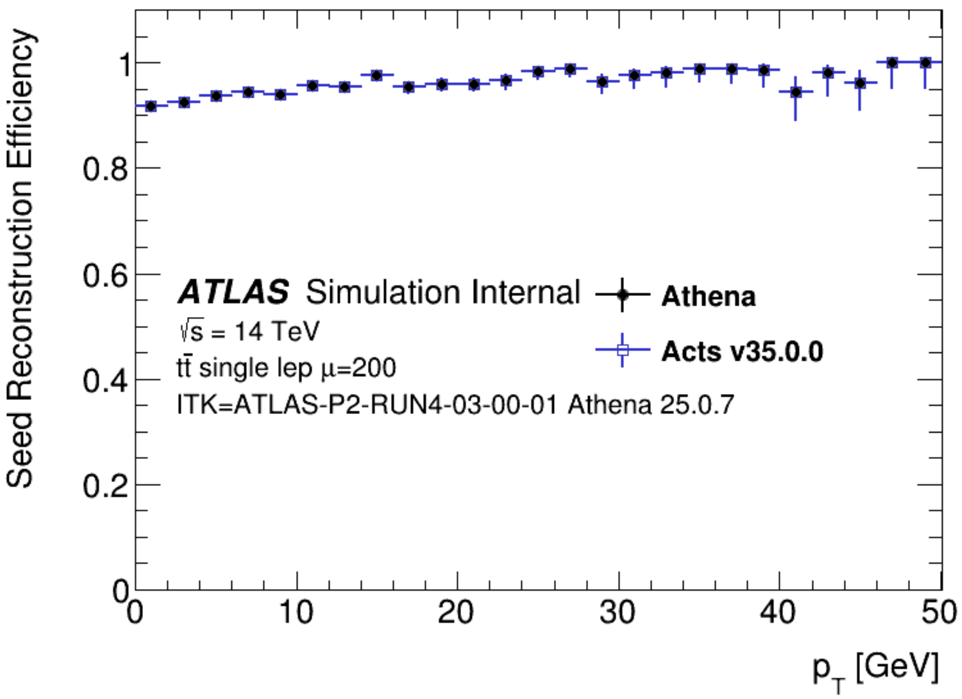
- **μ=0,60,140,200**
- **AMI tags: e8481_s4272_s4275**
- **Athena release to make the dataset: 25.0.0**
- **Digitization: *StandardInTimeOnlyTruth***

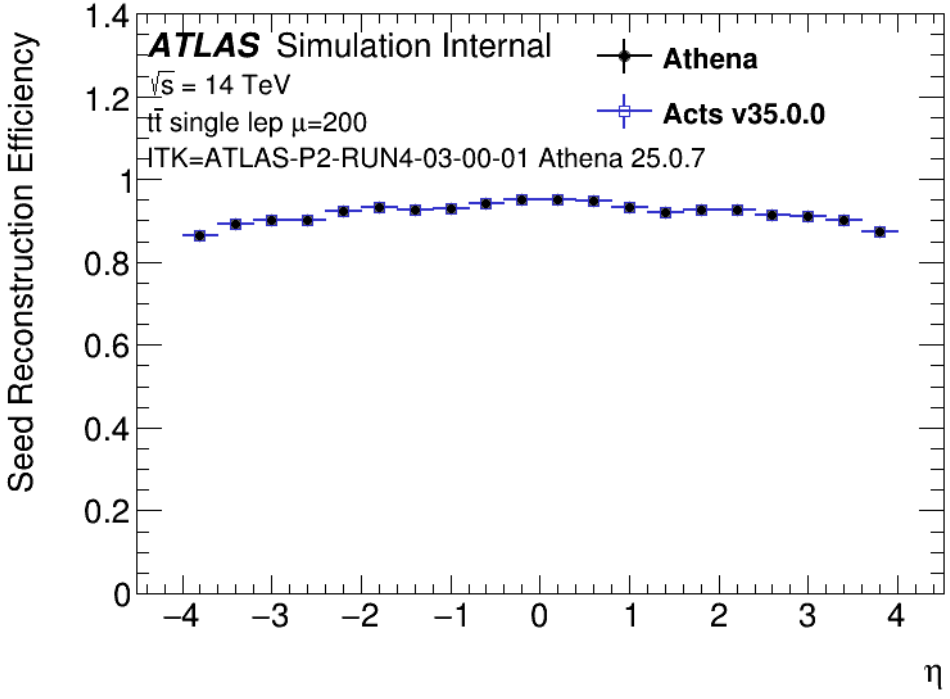
Moving to Athena

1) **Move the Hashing code to core:**
   1) Move the code to core (local) √
   2) Make the code compile √
   3) Ensure same results than before √
   4) Pull requests: Hashing √, Container Policy, Event Timing, Root seed writer √, Root comparison √
2) **Link ACTS+Hashing version in Athena (TWiki) √**
3) **Reproduce official plots (slide 17):**
   1) Athena + Default: μ = 0 √; μ = 200 √
   2) Athena + ACTS μ = 0 √; μ = 200 √
   3) Athena + ACTS (custom) μ = 0 √; μ = 200 √
4) **Edit seeding tool (in Athena) with Hashing √**
5) **Reproduce the plots with Hashing (Eff √ + CPU~)**

Setup validation: Efficiency μ=200

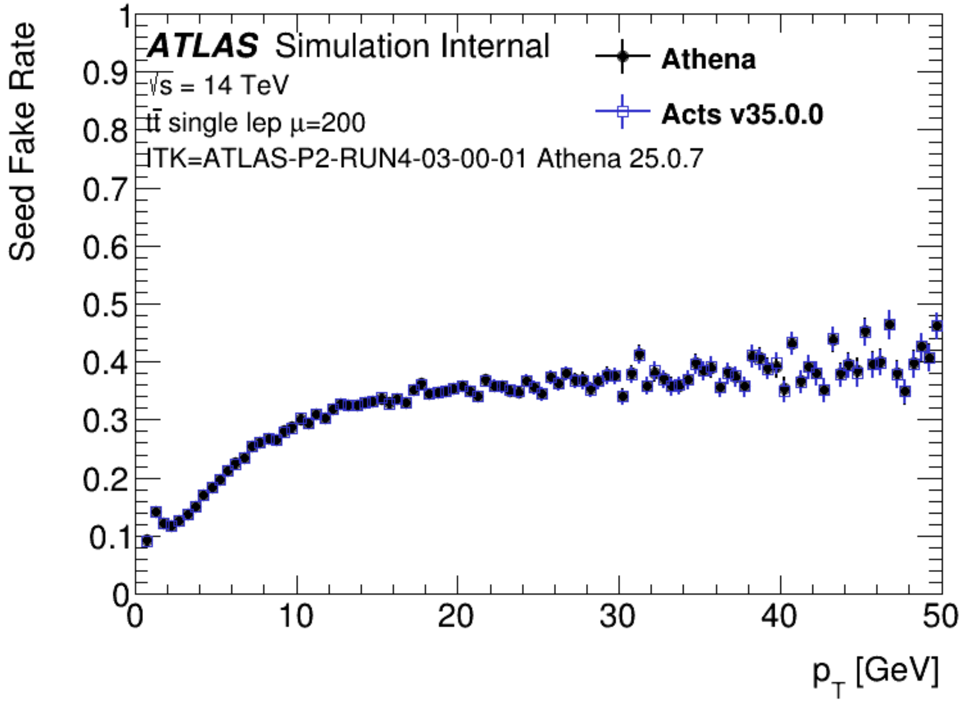
InnerTracker

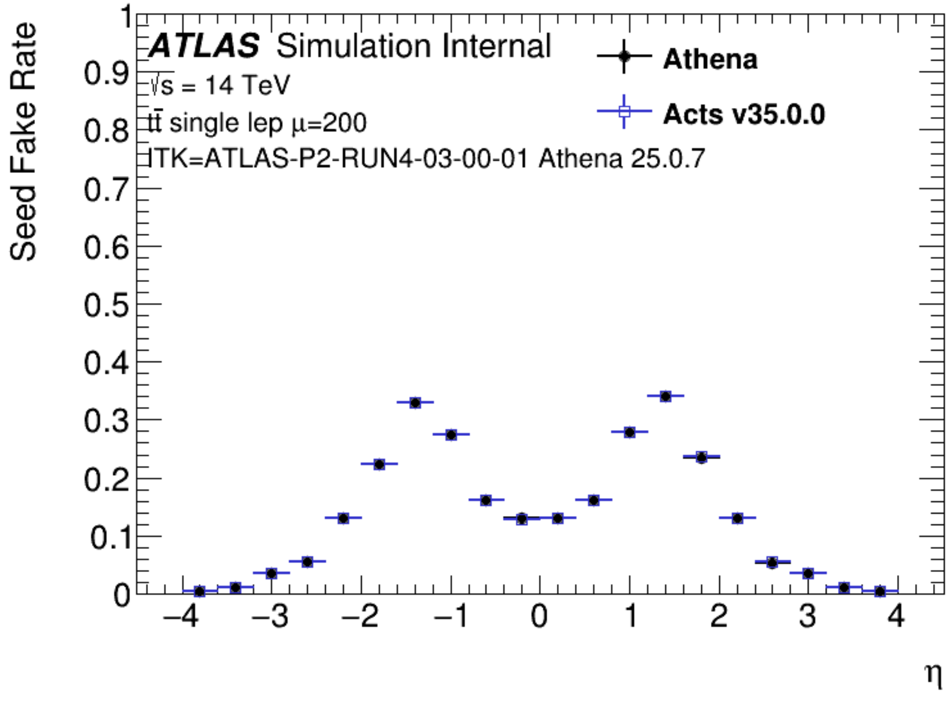




Setup validation: Fake Rate μ=200

InnerTracker





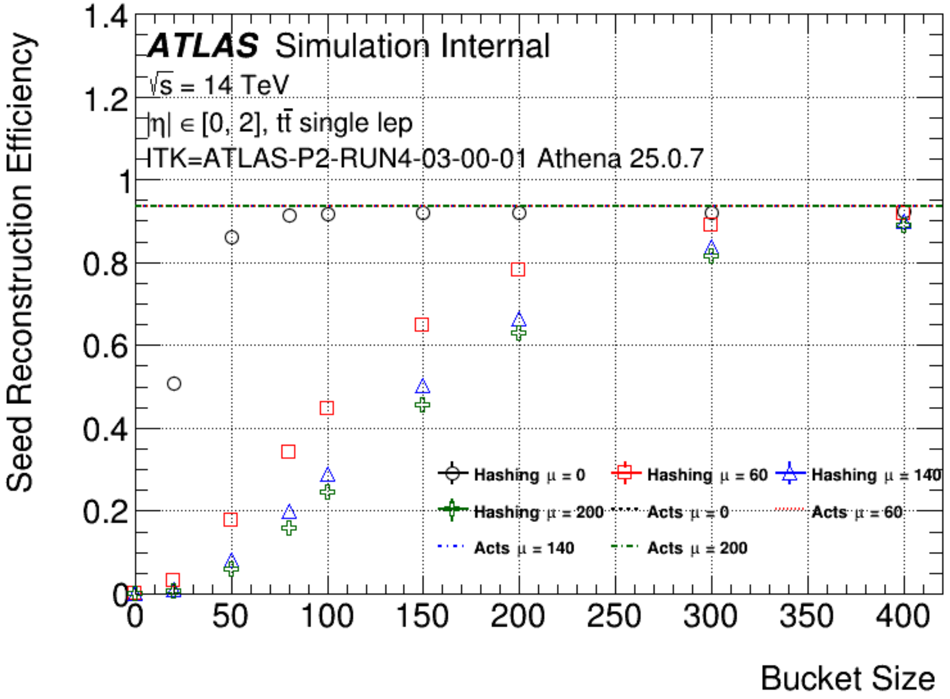
1000 events

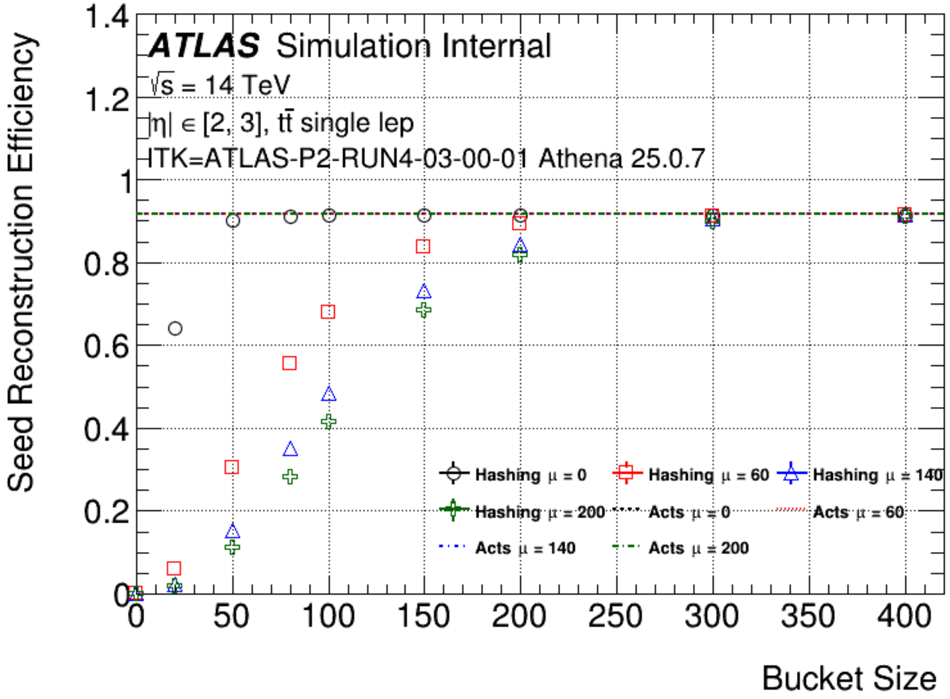
RESULTS

Bucket Size Δφ: η

InnerTracker

WARNING: not only first layer selected



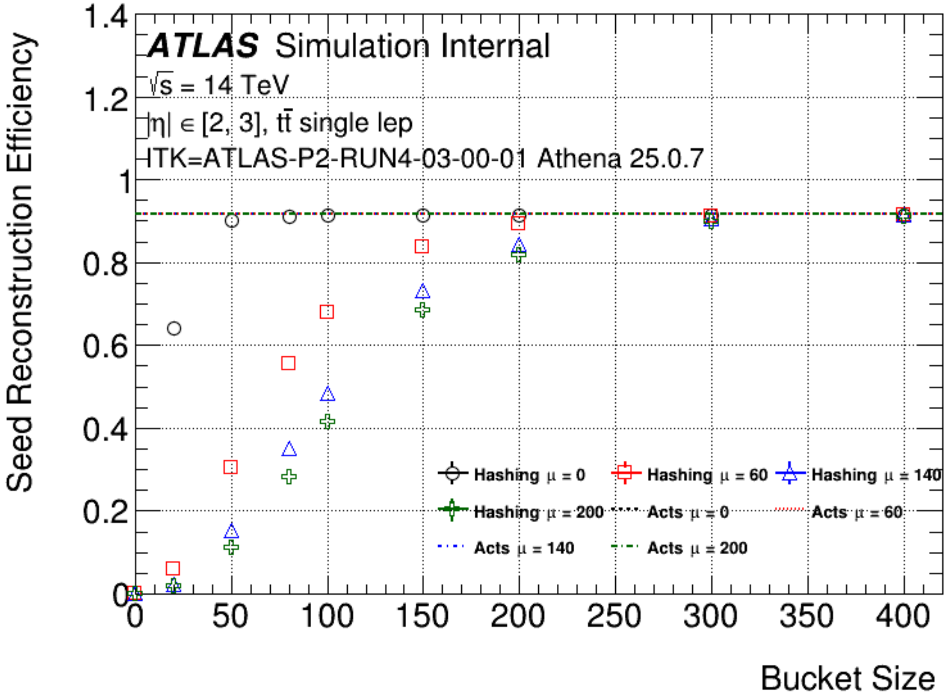


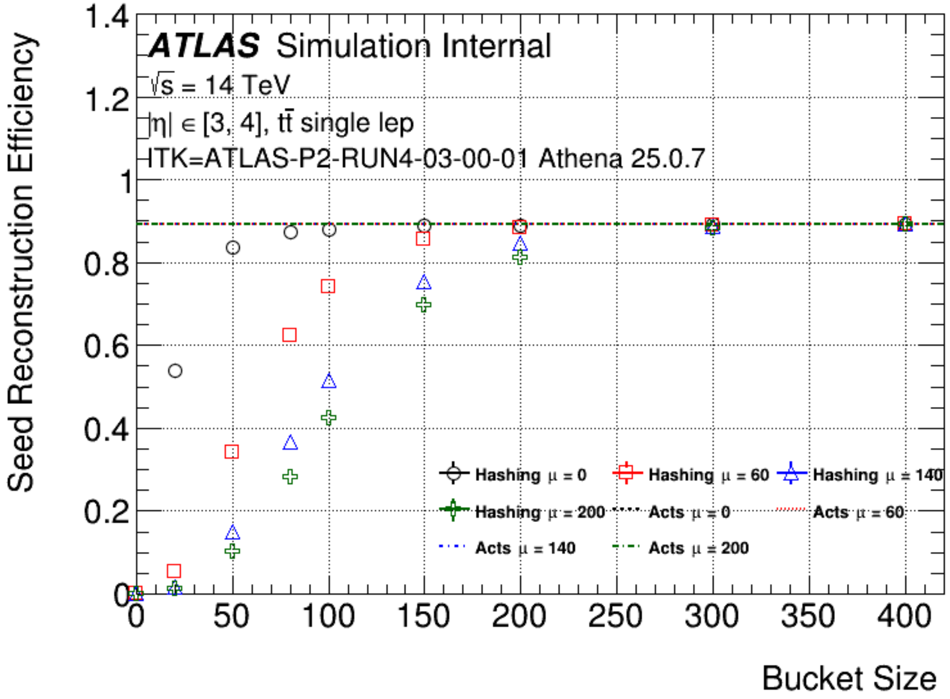
1000 events

Bucket Size Δφ: η

InnerTracker

WARNING: not only first layer selected



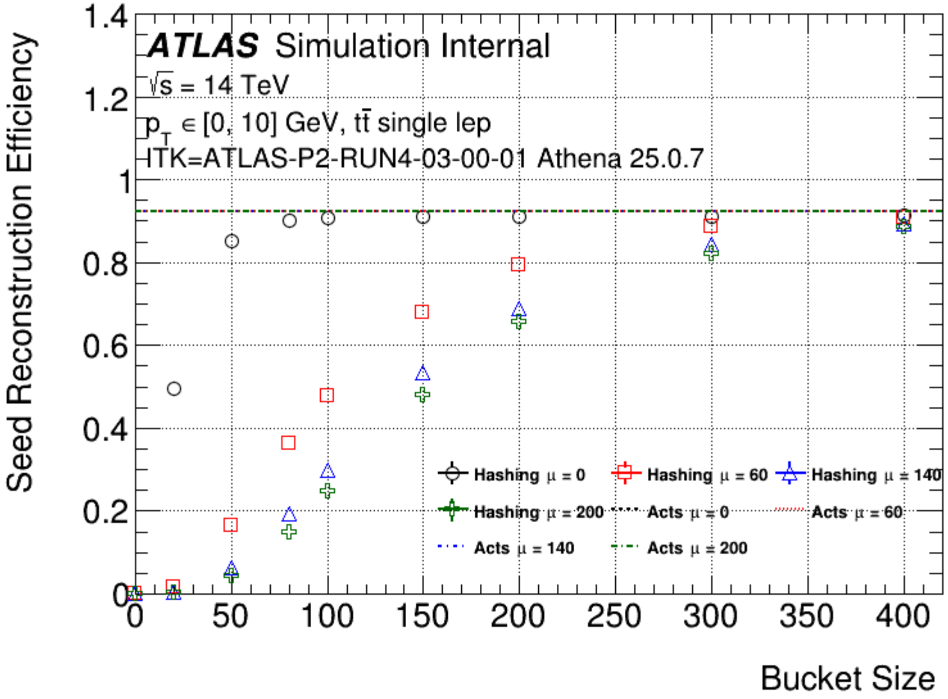


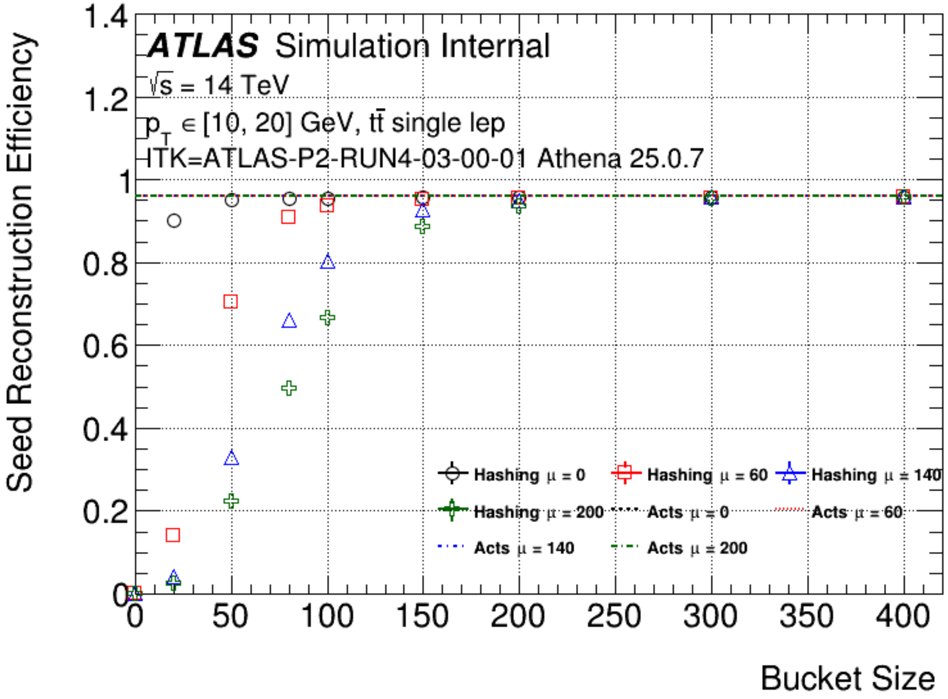
1000 events

Bucket Size Δφ: pT

InnerTracker

WARNING: not only first layer selected



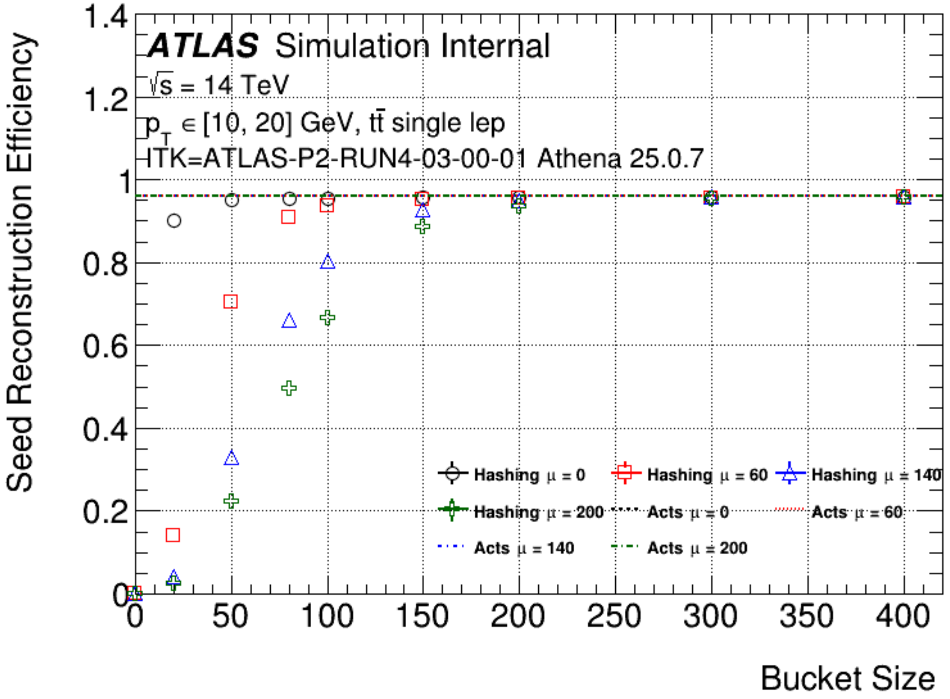


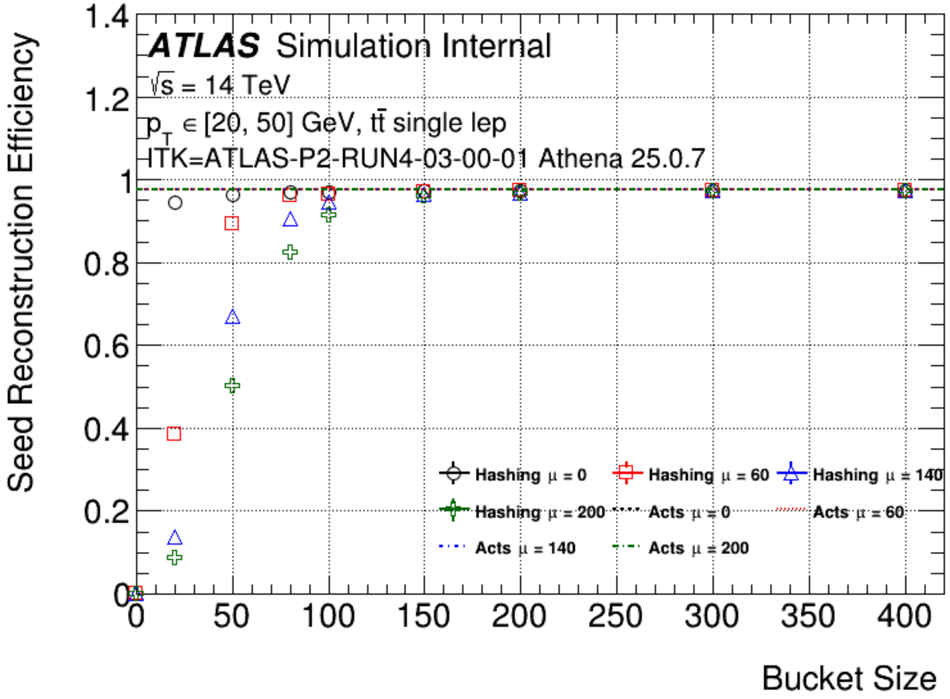
1000 events

Bucket Size Δφ: pT

InnerTracker

WARNING: not only first layer selected

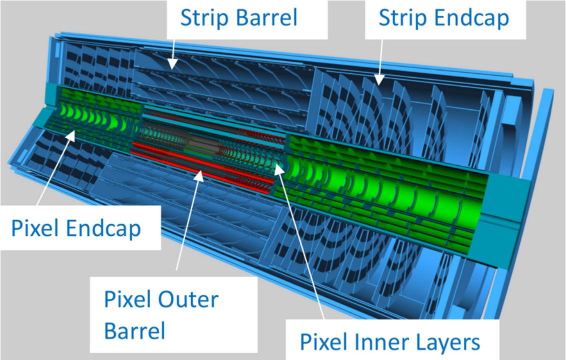




1000 events

Realistic case

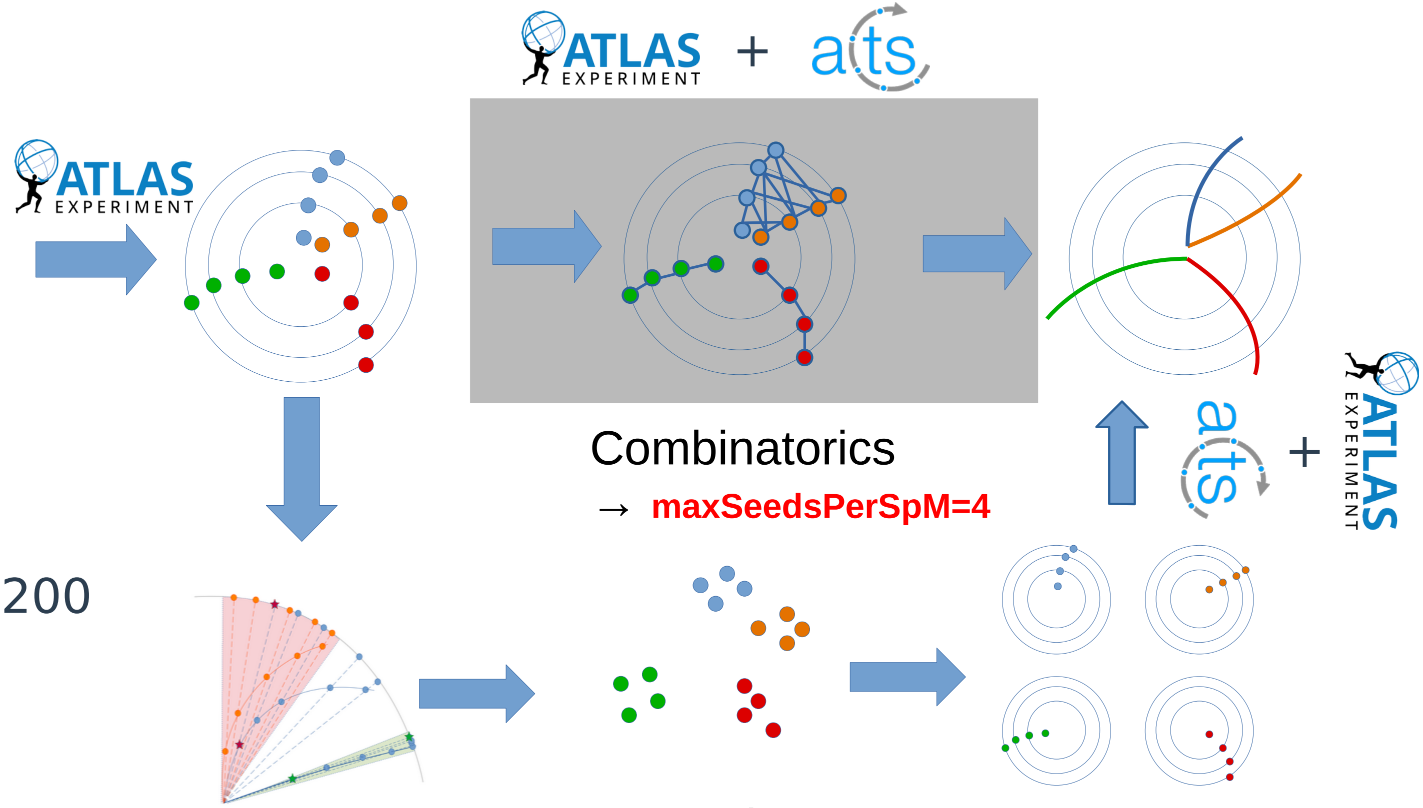
Inner Tracker (ITk)
(beeing built)



HL-LHC:

<μ> = 200

Official simulations
1000 $t\bar{t}$ events
μ = 60, 140, 200



Combinatorics

→ **maxSeedsPerSpM=4**

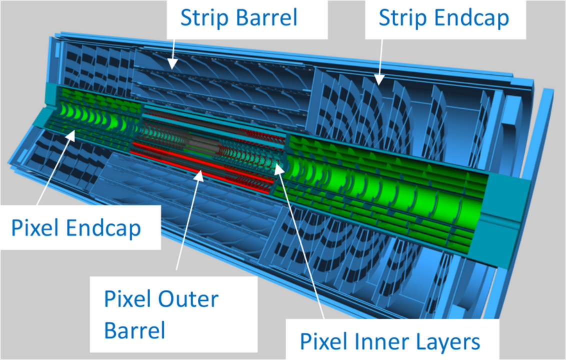
Metric: Δφ

Clustering:
Annoy

Parallelization

Realistic case

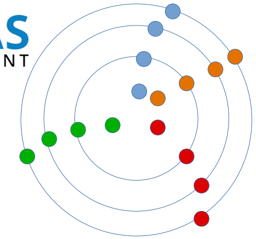
Inner Tracker (ITk)
(beeing built)

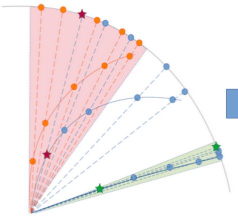


HL-LHC:

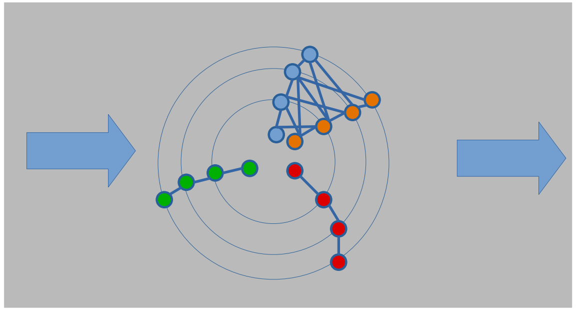
$\langle\mu\rangle = 200$

Official simulations
1000 $t\bar{t}$ events
μ = 60, 140, 200



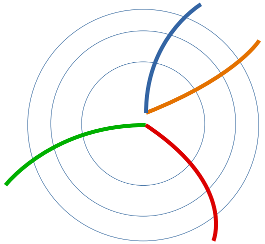


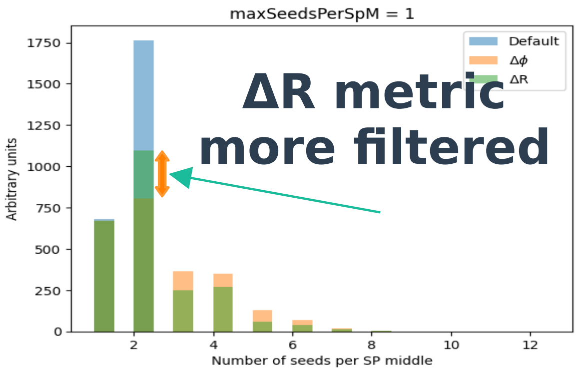
Metric: Δφ



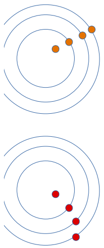
Combinatorics

→ **maxSeedsPerSpM=4**





ΔR metric more filtered

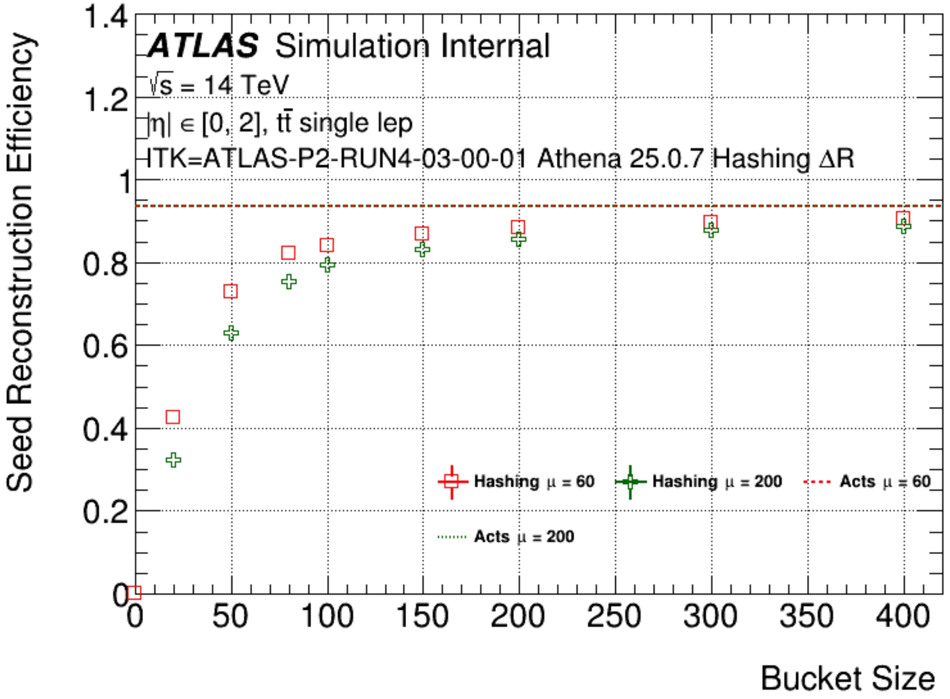


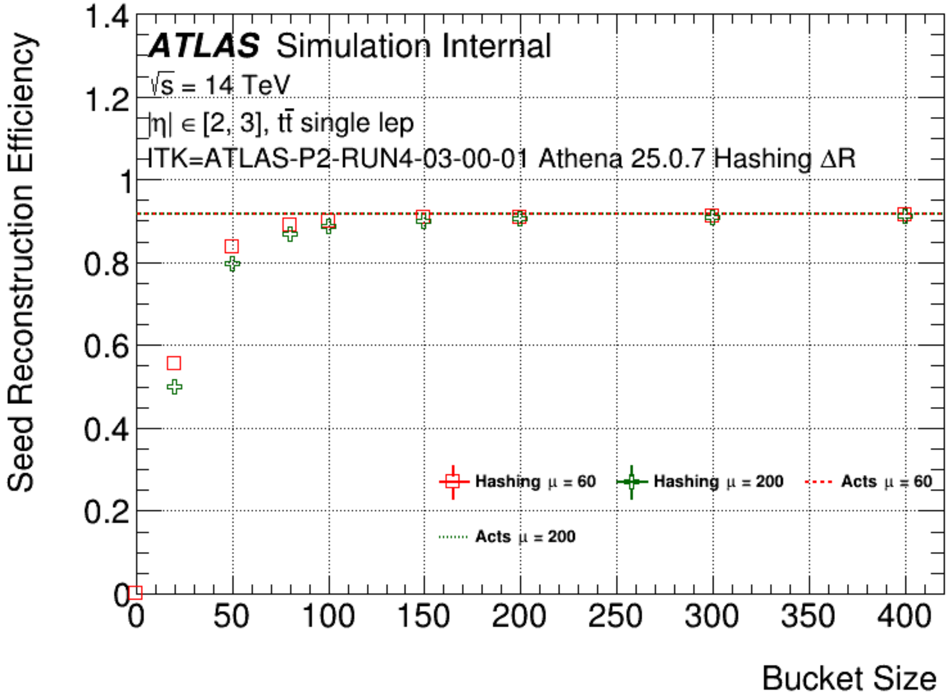
zation

Bucket Size ΔR: η

InnerTracker

WARNING: not only first layer selected



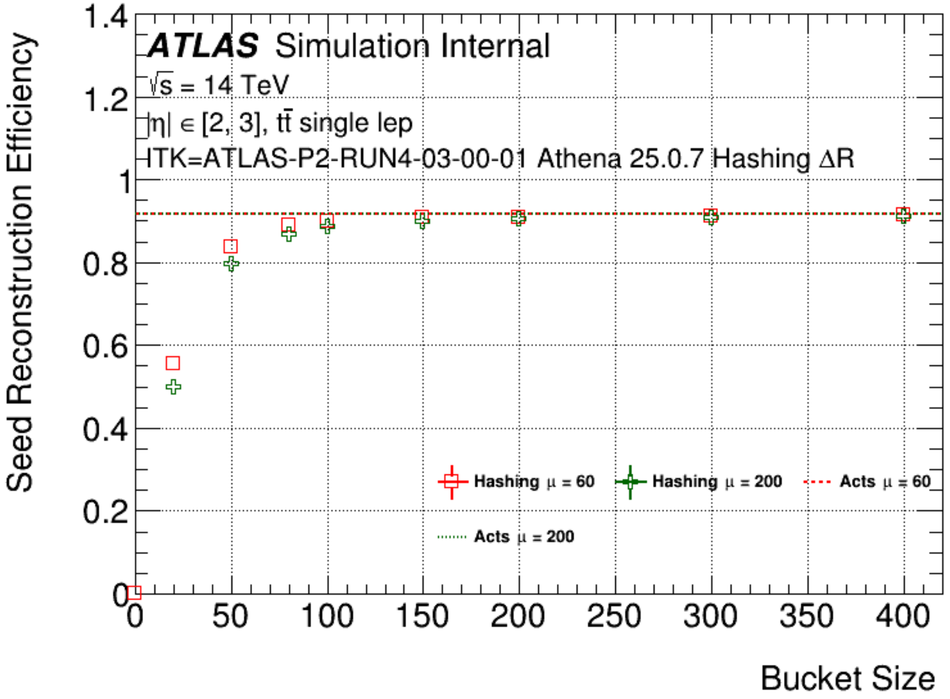


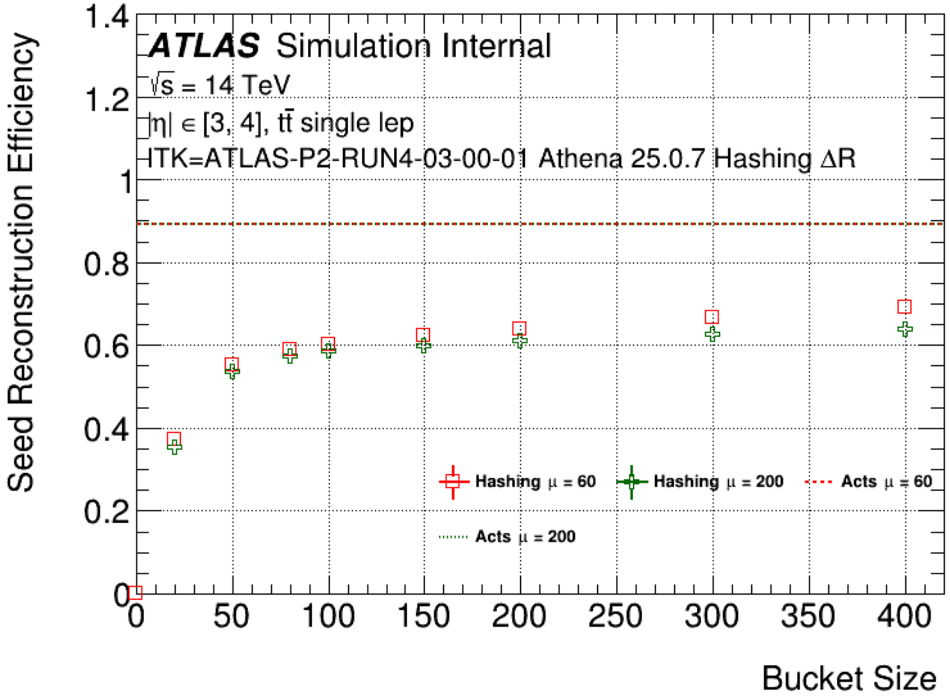
1000 events

Bucket Size ΔR: η

InnerTracker

WARNING: not only first layer selected



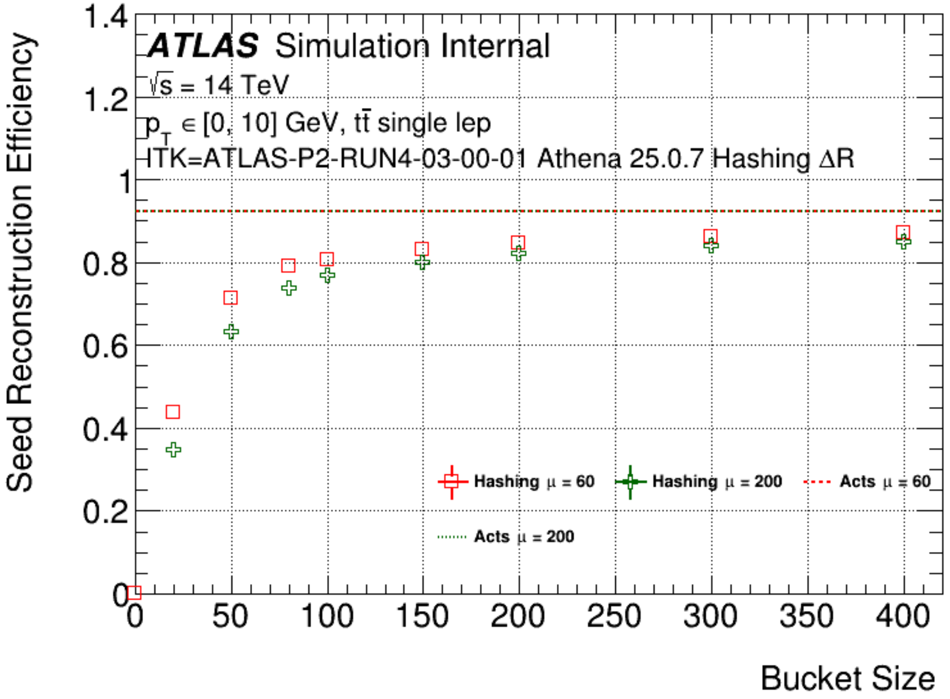


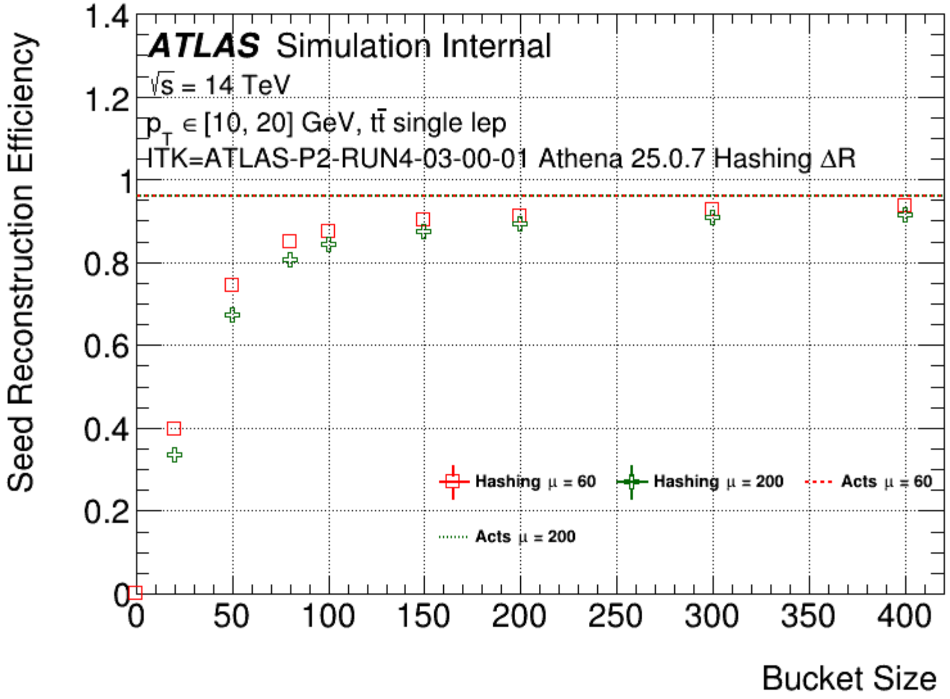
1000 events

Bucket Size ΔR: pT

InnerTracker

WARNING: not only first layer selected



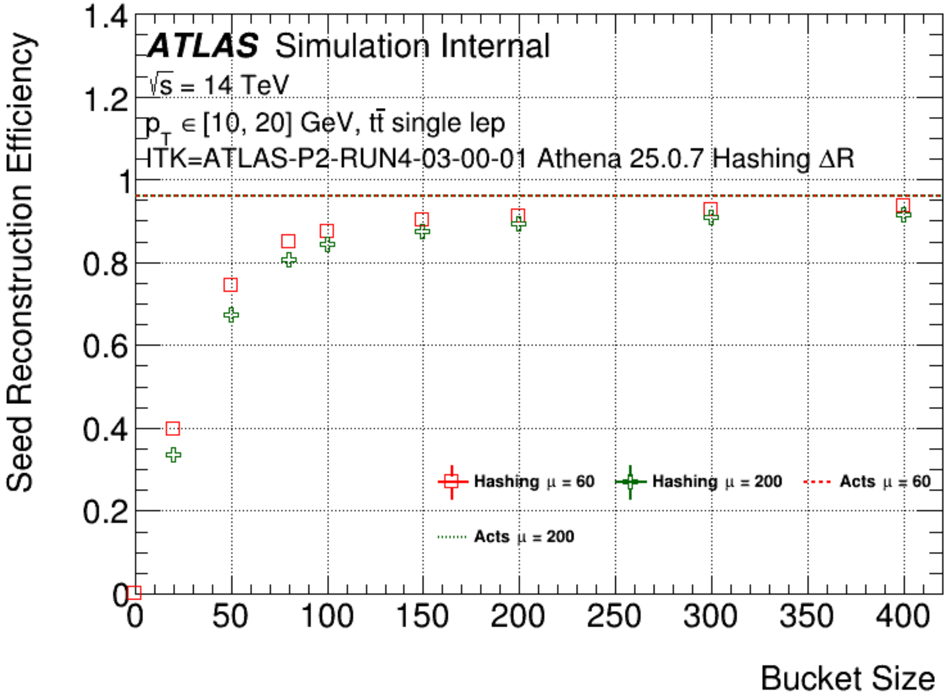


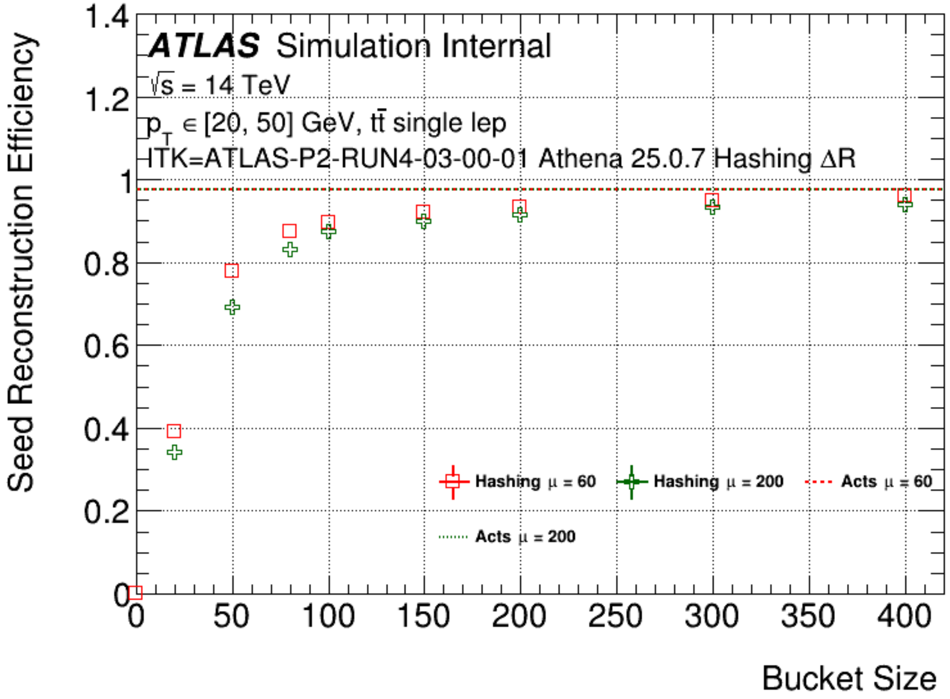
1000 events

Bucket Size ΔR: pT

InnerTracker

WARNING: not only first layer selected



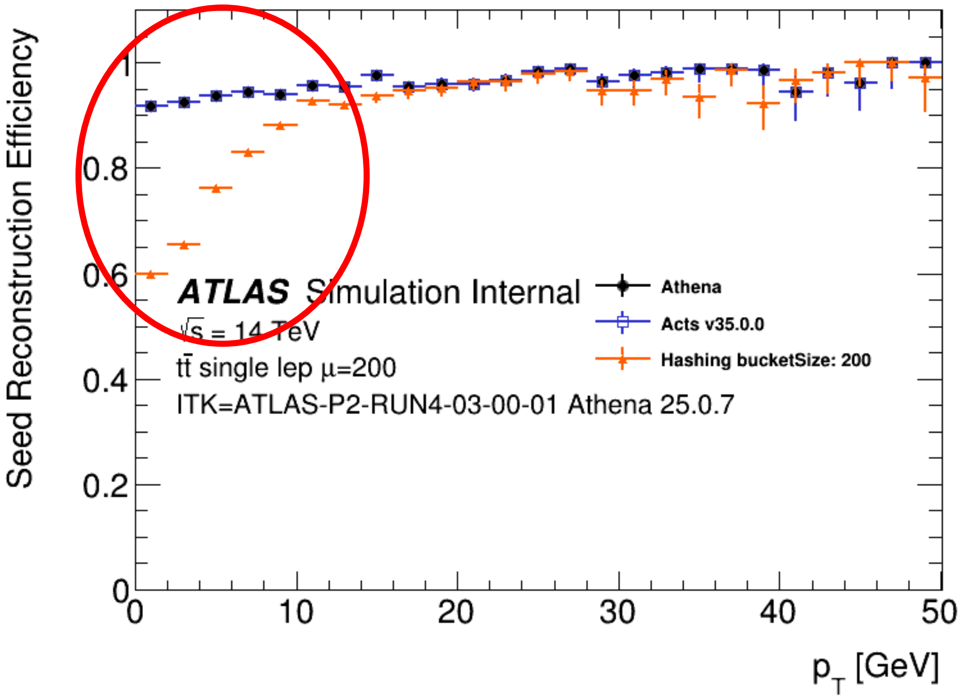


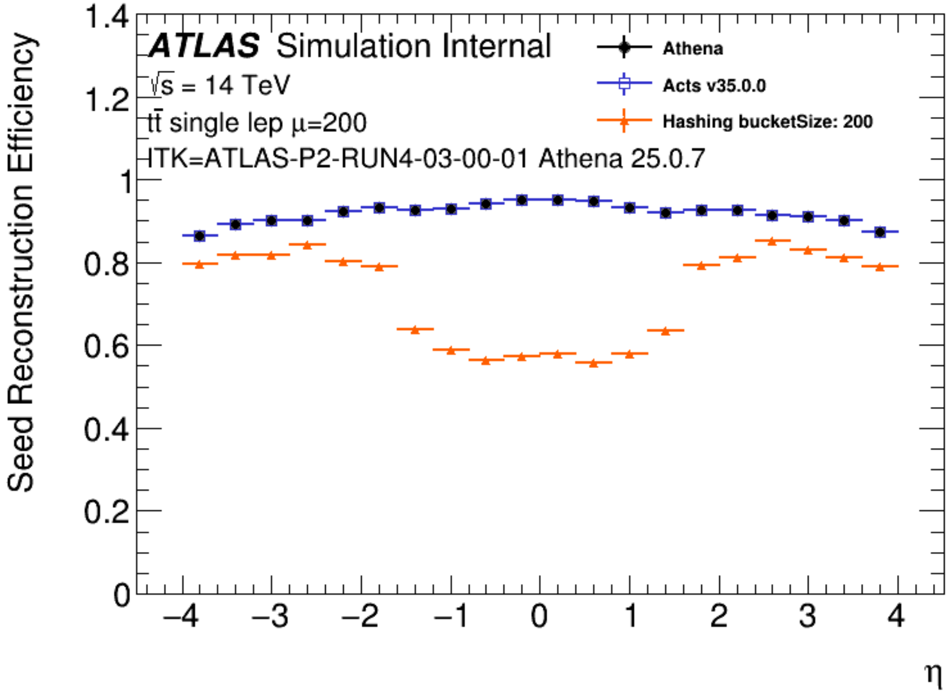
1000 events

Δφ: Seed Efficiency μ=200

InnerTracker

WARNING: not only first layer selected



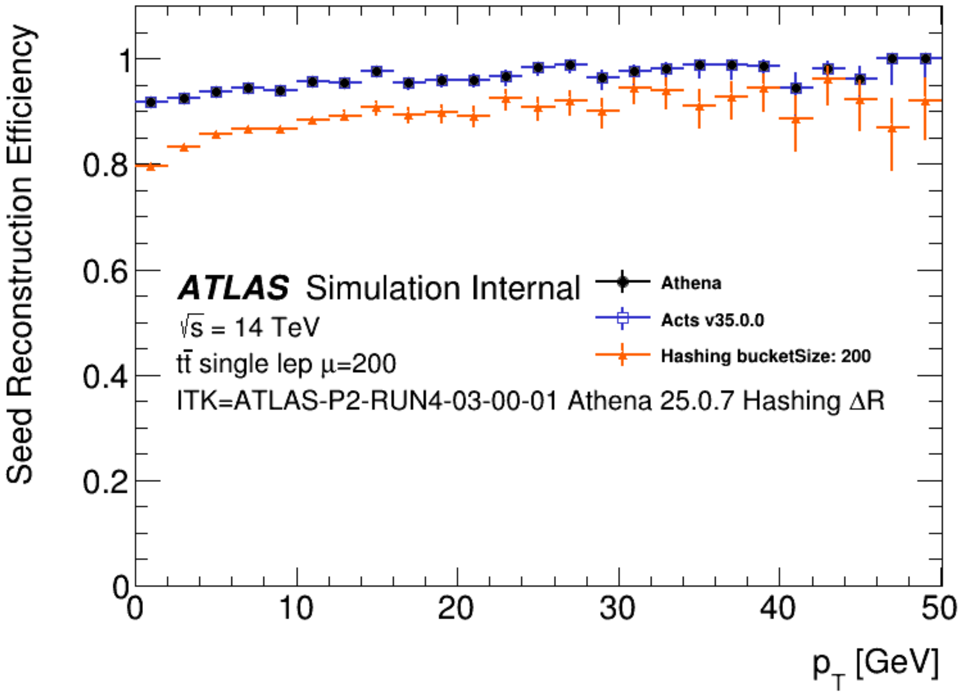


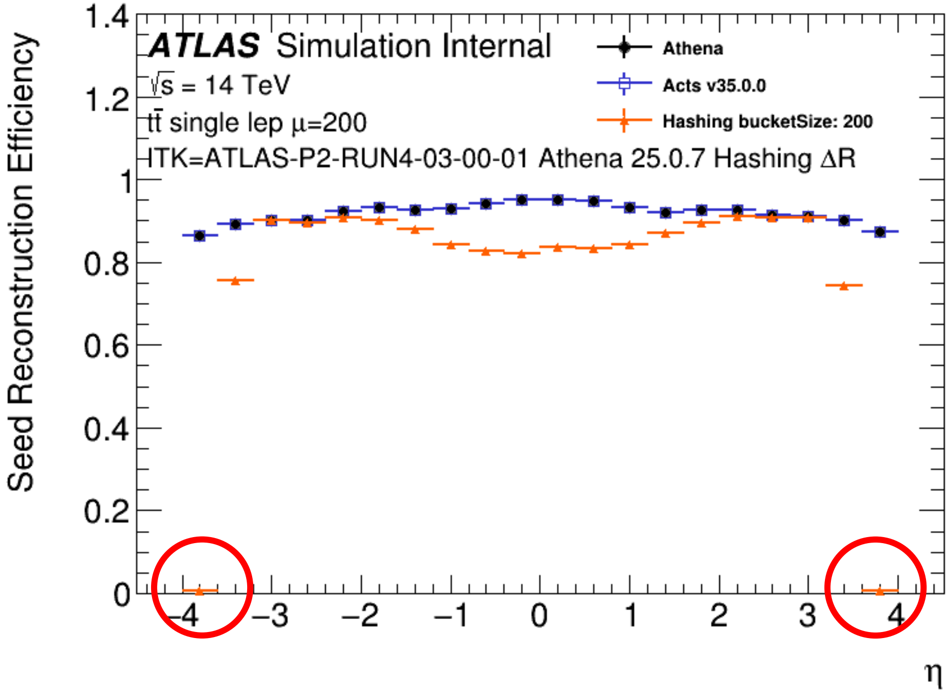
1000 events

ΔR: Seed Efficiency μ=200

InnerTracker

WARNING: not only first layer selected

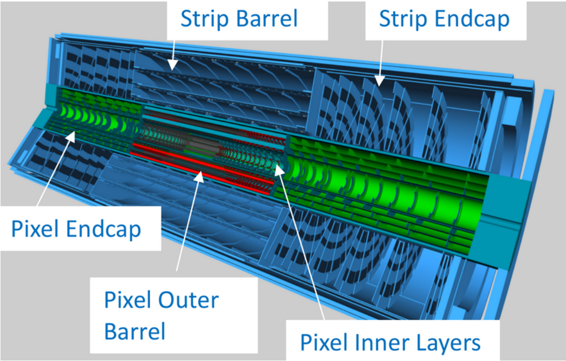




1000 events

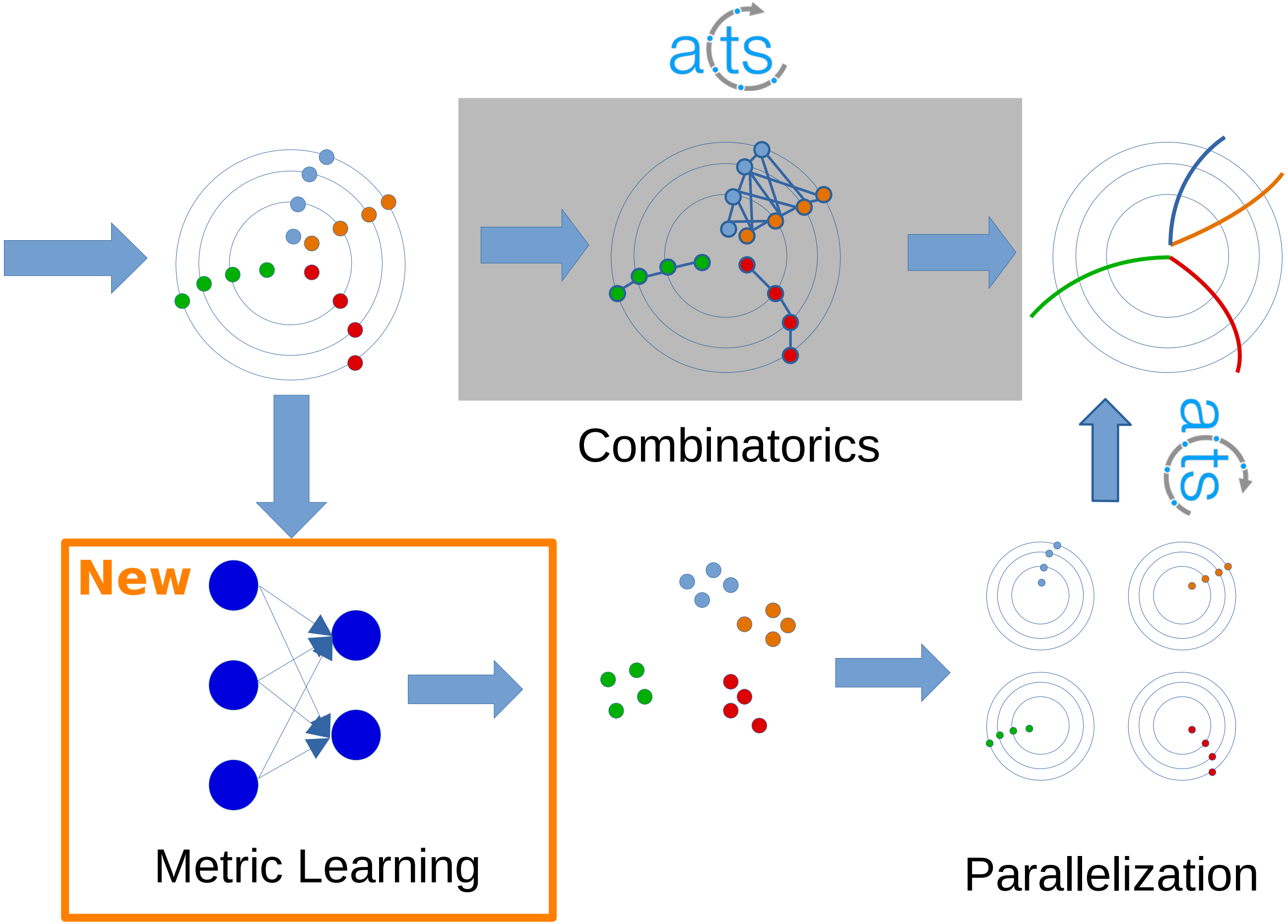
METRIC LEARNING

Tracking



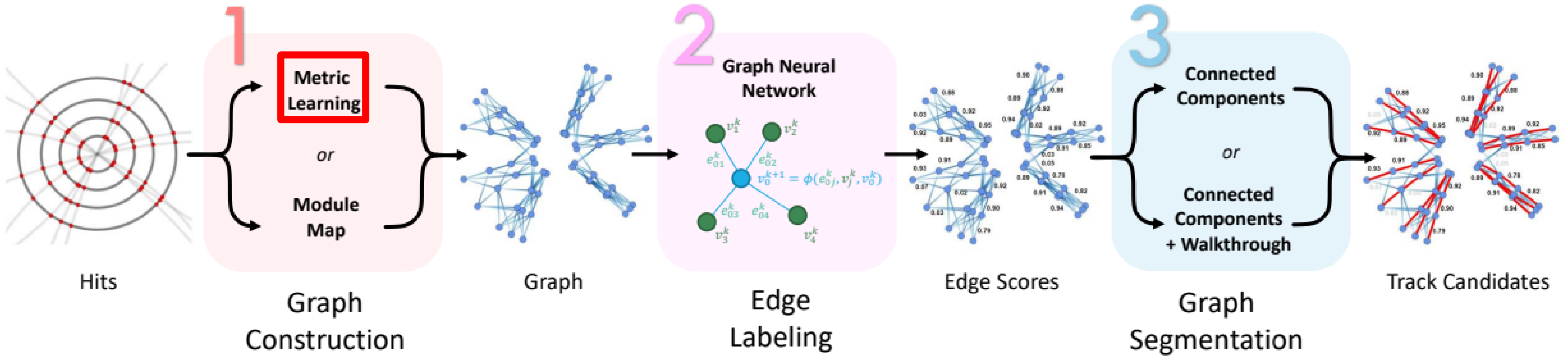
InnerTracker (ITk)

ATLAS Detector at High Luminosity LHC



Acorn

gitlab



Schematic overview of the GNN-based track finding pipeline

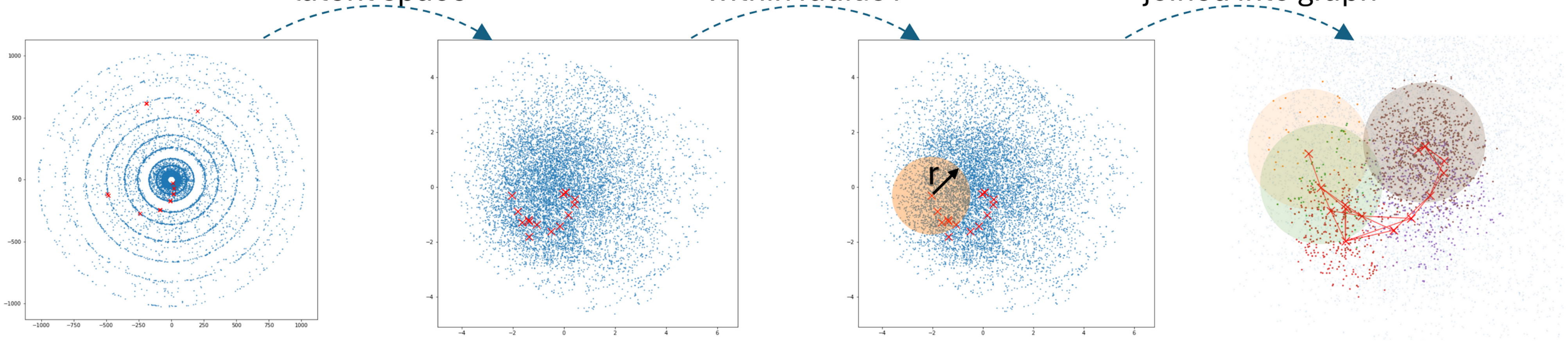
https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/IDTR-2022-01/

GNN Metric Learning

Embed into learned latent space

Connect all space points within radius r

All space point pairs joined into graph



Model

- **Example 2**

1. First, we build our input data from the raw Athena events:

```
acorn infer data_reader.yaml
```

2. We start the graph construction by training the Metric Learning stage:

```
acorn train metric_learning_train.yaml
```

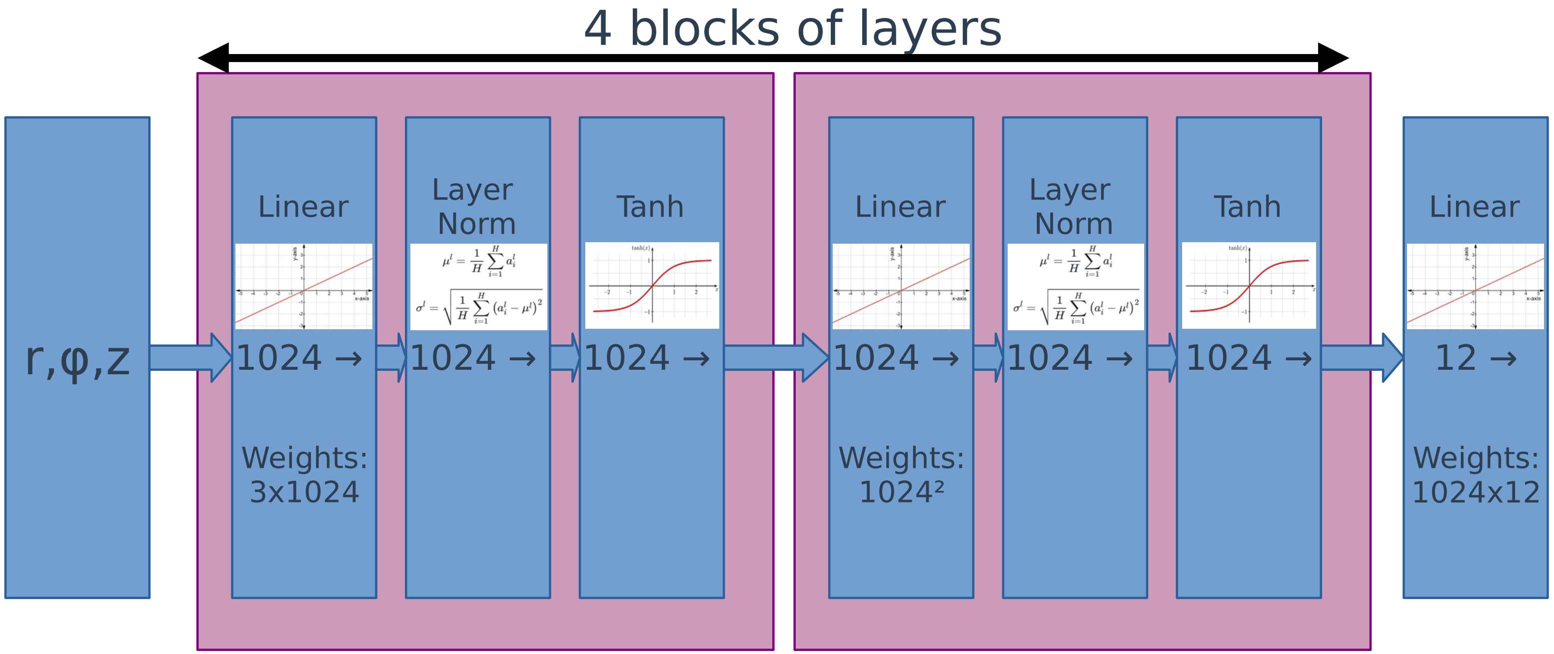
3. Then, we build graphs using the Metric Learning in inference:

```
acorn infer metric_learning_infer.yaml
```

```
# Model inference parameters
r_infer: 0.1
knn_infer: 1000
```

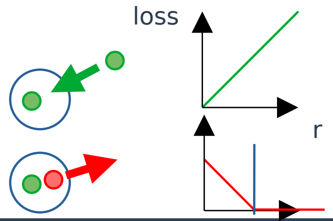
```
hard_cuts:
  pt: [1000, .inf]

# Model parameters
undirected: True
node_features: [r,    phi,   z]
node_scales:   [1000, 3.14, 1000]
emb_hidden: 1024
nb_layer: 4
emb_dim: 12
activation: Tanh
randomisation: 1
points_per_batch: 50000
r_train: 0.1
knn: 50
knn_val: 1000

# Training parameters
warmup: 5
margin: 0.1
lr: 0.01
factor: 0.7
patience: 10
max_epochs: 100
metric_to_monitor: f1
metric_mode: max
```

Architecture

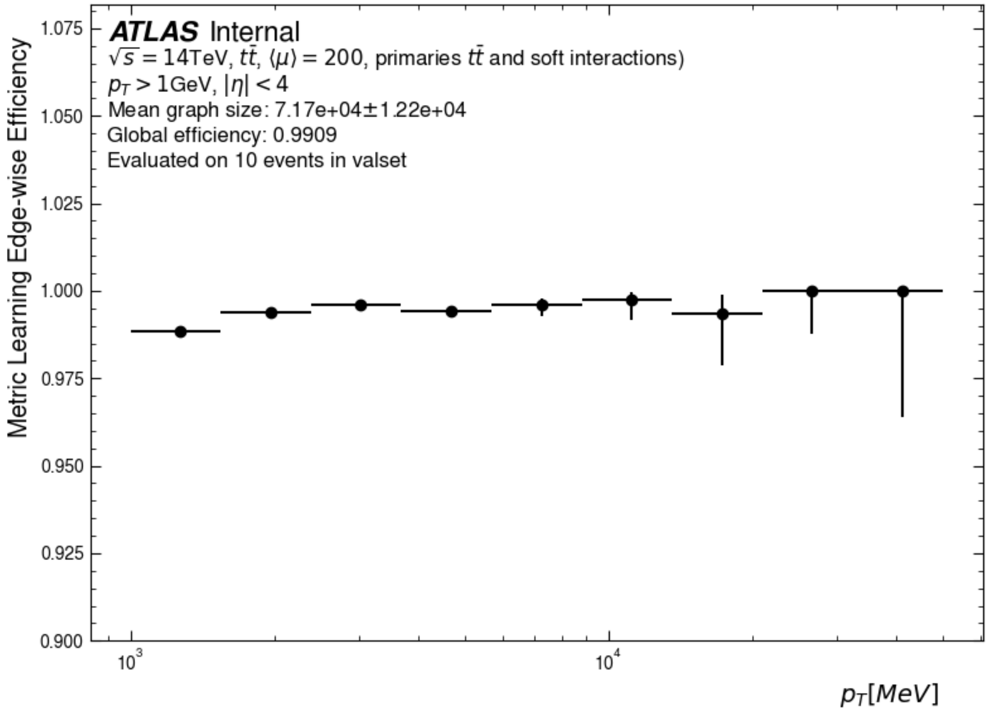


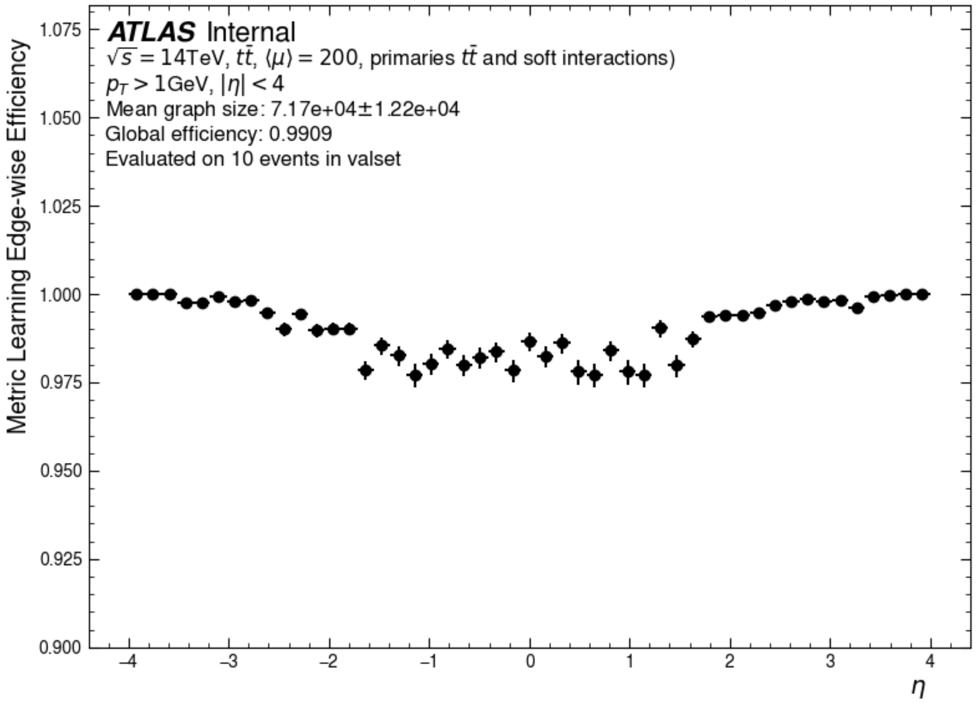
Hinge Loss

$$l_n = \begin{cases} x_n, & \text{if } y_n = 1, \\ \max\{0, margin - x_n\}, & \text{if } y_n = -1, \end{cases}$$



Performance

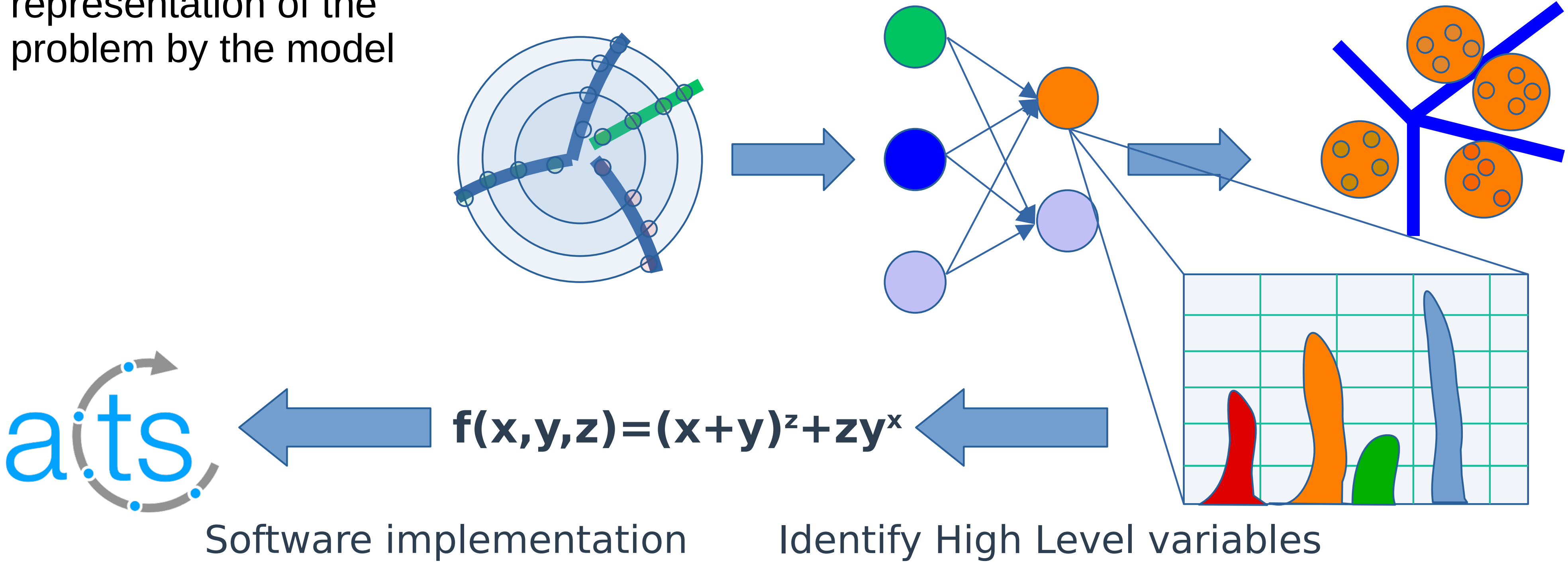




INTERPRETABILITY

Interpretability

- study the internal representation of the problem by the model

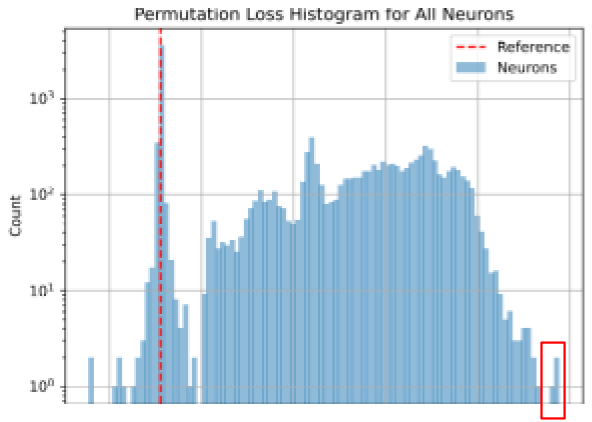


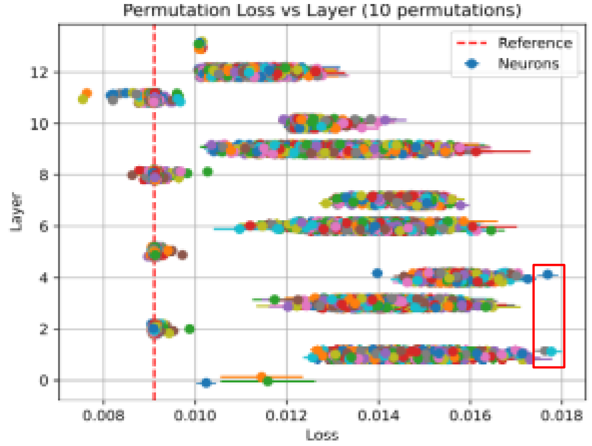
Interpretability plan

- **Assume the model is building an algorithm internally: *mechanistic interpretability***
- **Goal: identify parts of this algorithm (relevant pieces)**
- **Steps:**
  1) Identify relevant neurons
  2) Symbolic regression to obtain a formula of the quantity approximated
  3) Identify relevant parts of the equation
  4) Compare with known physics high-level variables

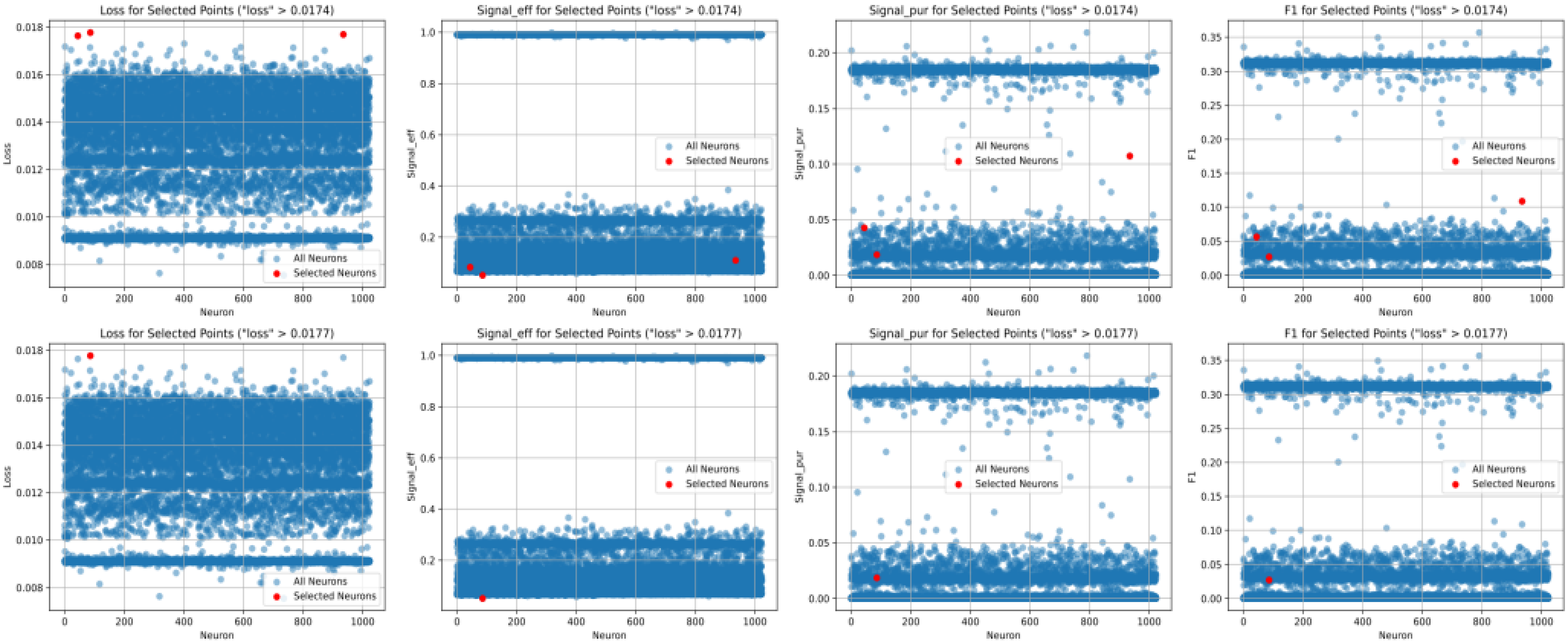
Neuron identification: Permutation loss

- **3 promising neurons:**
  - 2 on layer 1 (***Linear*** with input layer)
  - 1 on layer 4 (More complex)
- **Normalization Layers (3n-1) not perturbed by permutation → Information is shared among neurons**

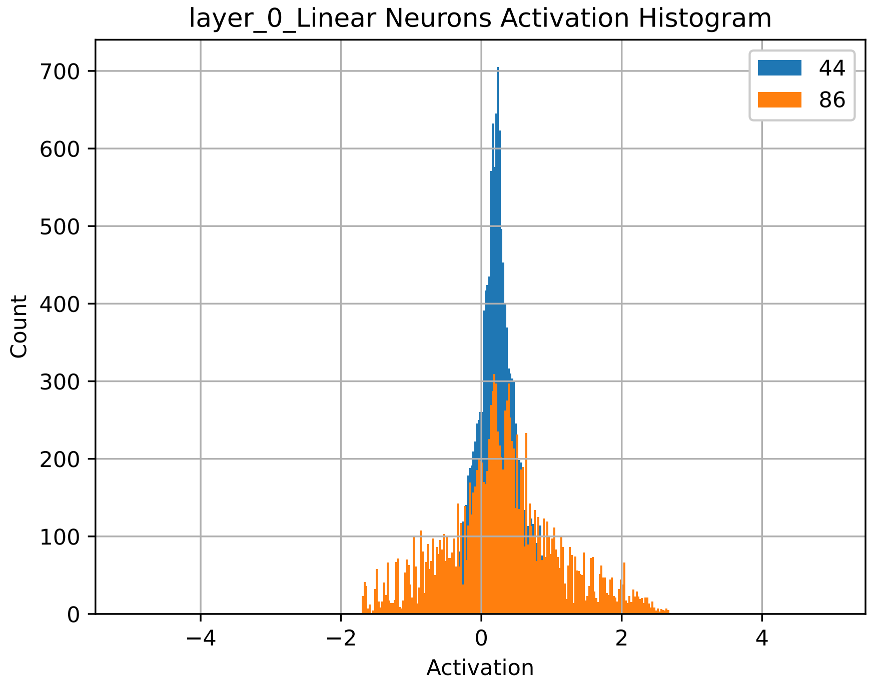


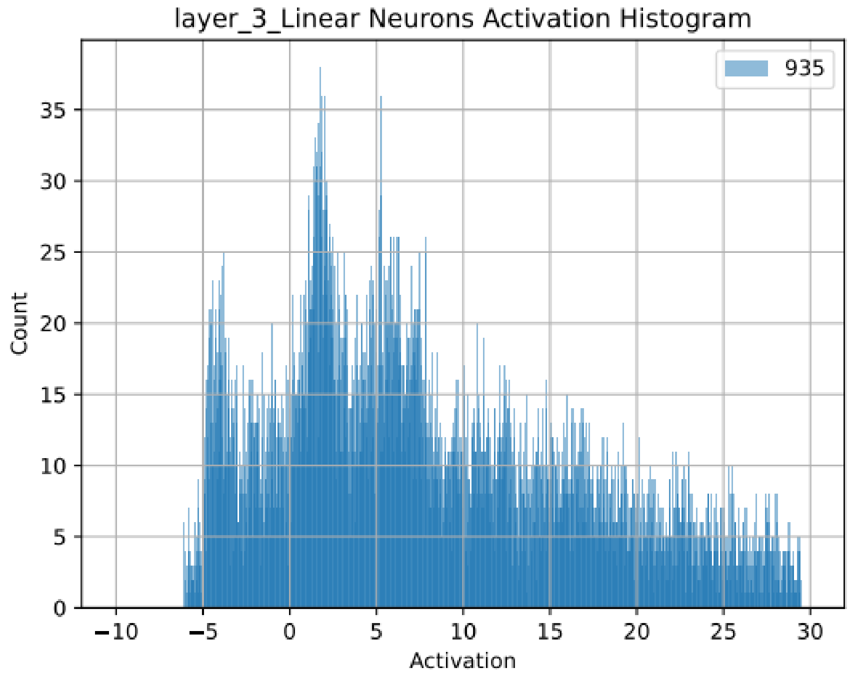


Neuron specificities



Activations

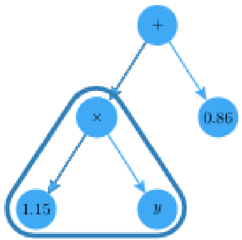


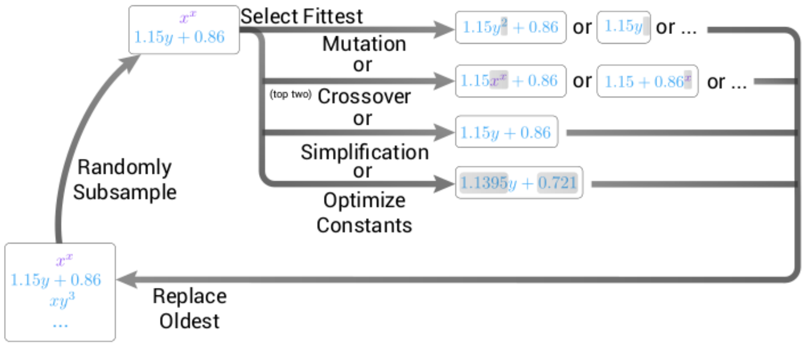


Symbolic regression

PySR

$1.15y + 0.86$





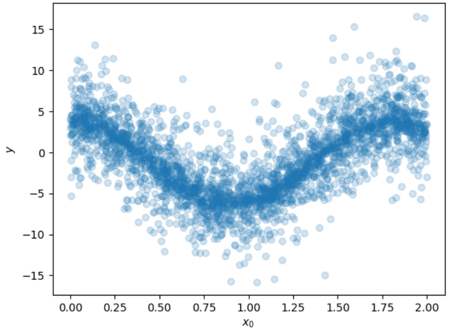
Genetic Algorithm

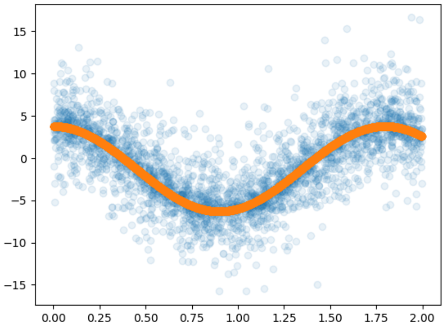
$$\sigma \sim U(0.1, 5.0)$$
$$\epsilon \sim N(0, \sigma^2)$$
$$y = 5\ \cos(3.5x_0) - 1.3 + \epsilon.$$

Truth

$$5.0337477 \cos\left(3.496164 x_0\right) - 1.29099218487498$$

Learned





Neural Nets + Symbolic Regression

https://github.com/MilesCranmer/PySR

FORMATIONS

Formations

- **Ecole Doctorale (UGA):**
  - Requires 120 hours: 1/3 Scientific, 1/3 Professional, 1/3 Transversal
  - Current: 113/120
- **Professional:**
  - “S'ADAPTER A SON ENVIRONNEMENT DE TRAVAIL” (10 hours)
  - “Formation Entreprenariat PhDiscovery 2024” (30 hours)
- **Scientific:**
  - Workshops: ATLAS ML, ITk Tracking, ATLAS Induction Day and Software Tutorial (44 hours)
- **Transversal:**
  - Opened Science and HAL (4 hours)
  - “JOURNEE DE RENTREE DES DOCTORANTS 2022” (10 hours)
  - Mooc on ethics (15 hours) (not finished yet)
  - MOOC “Intégrité scientifique dans les métiers de la recherche” (15 hours)

Poster and publications

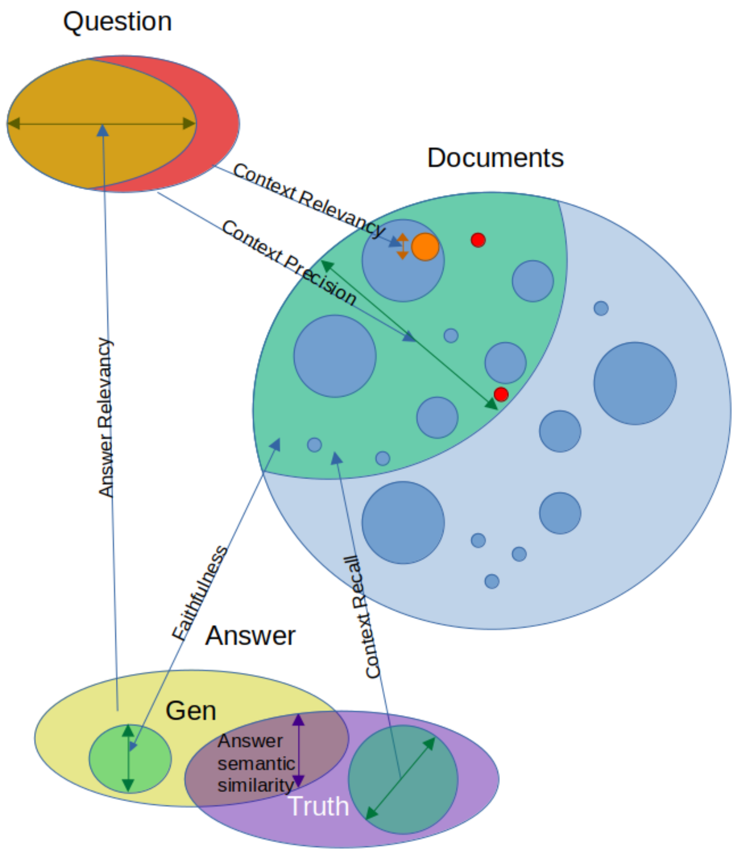
- Poster Connecting The Dots 2023
- Proceeding Poster Connecting The Dots 2023
- Proceeding Journée Rencontre Jeunes Chercheurs 2023
- Tutoriel ATLAS Machine Learning Workshop - chATLAS

chATLAS

- **ATLAS chatbot with ATLAS protected documents**

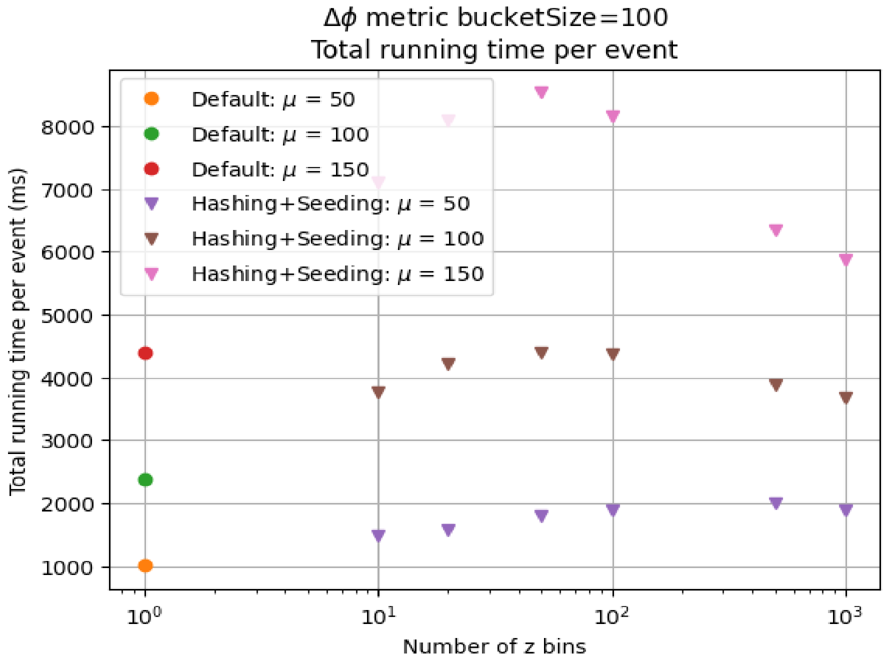
  (Retrieval Augmented Generation)

- **Worked on the evaluation**
- **Quitted team in september 2024**

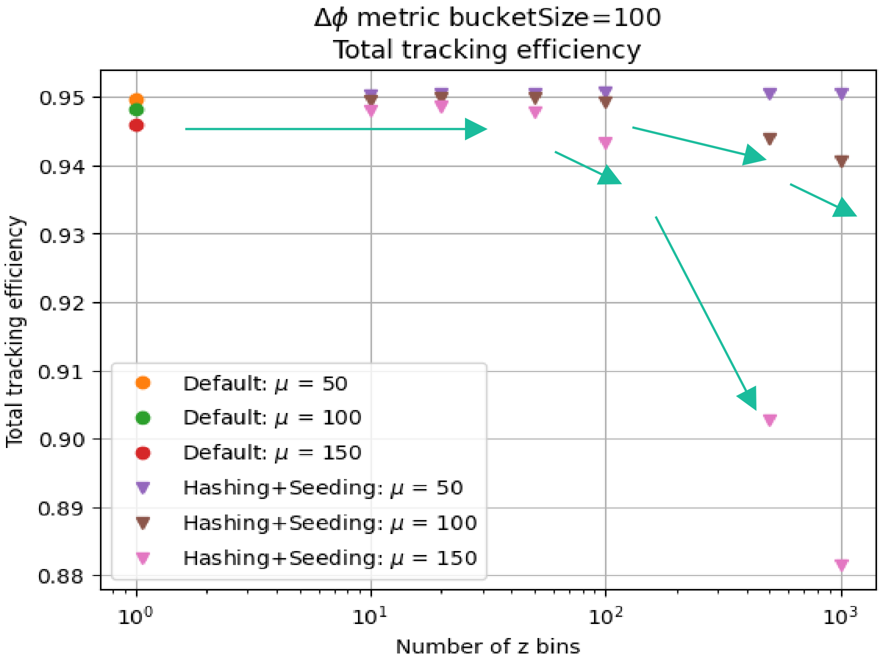


BACKUP

Hashing performance: Timing and efficiency

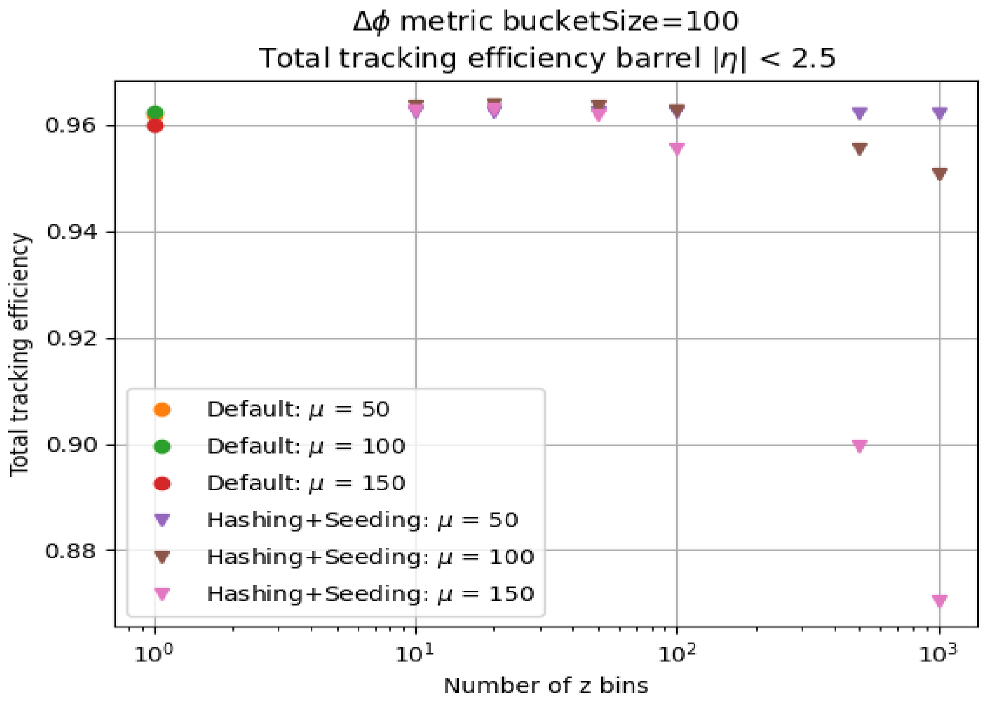


Running time ~ x2

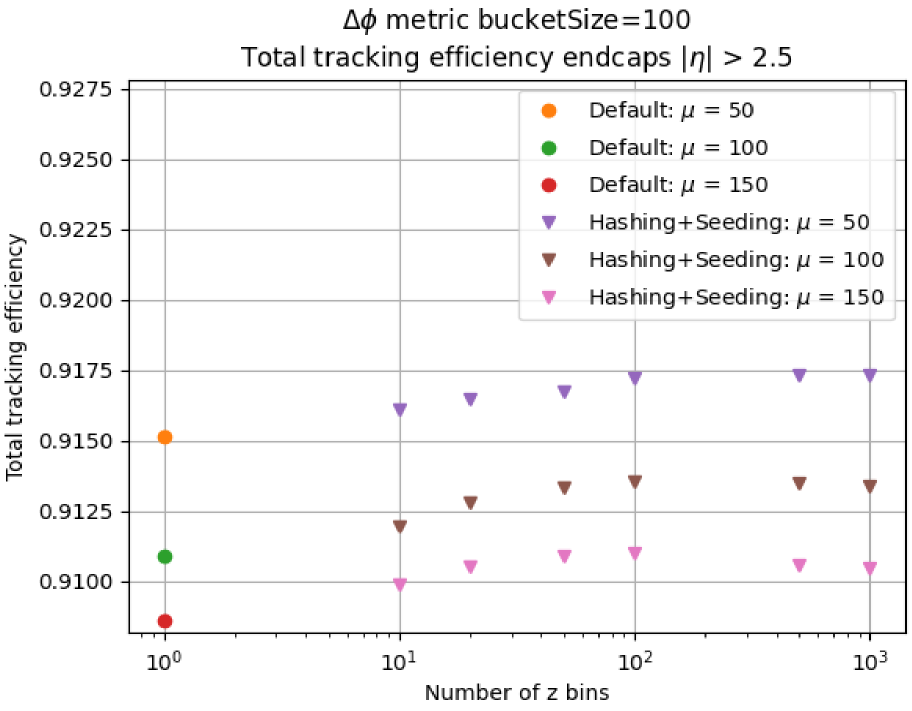


Improvement for small number of bins

Hashing performance: Efficiency (detailed)

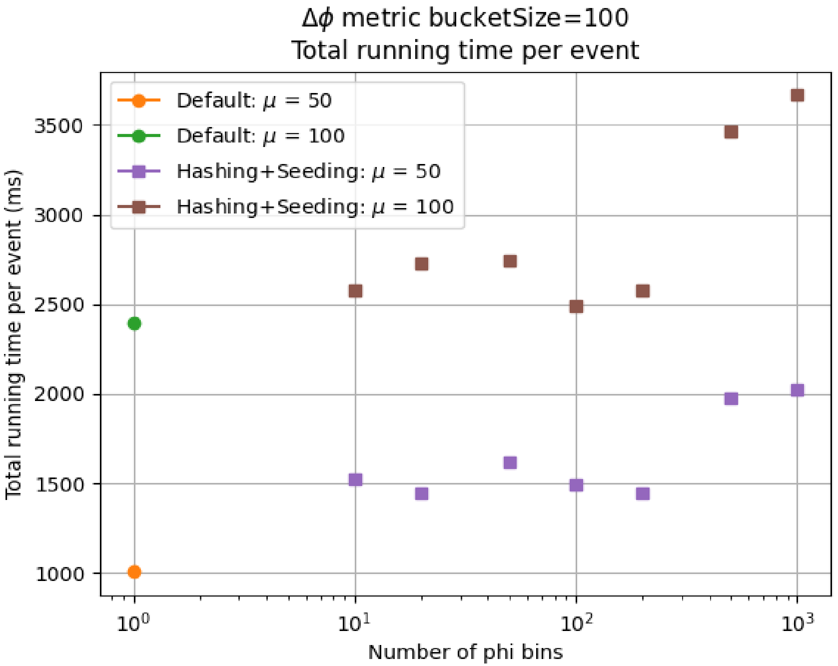


Improves then drops

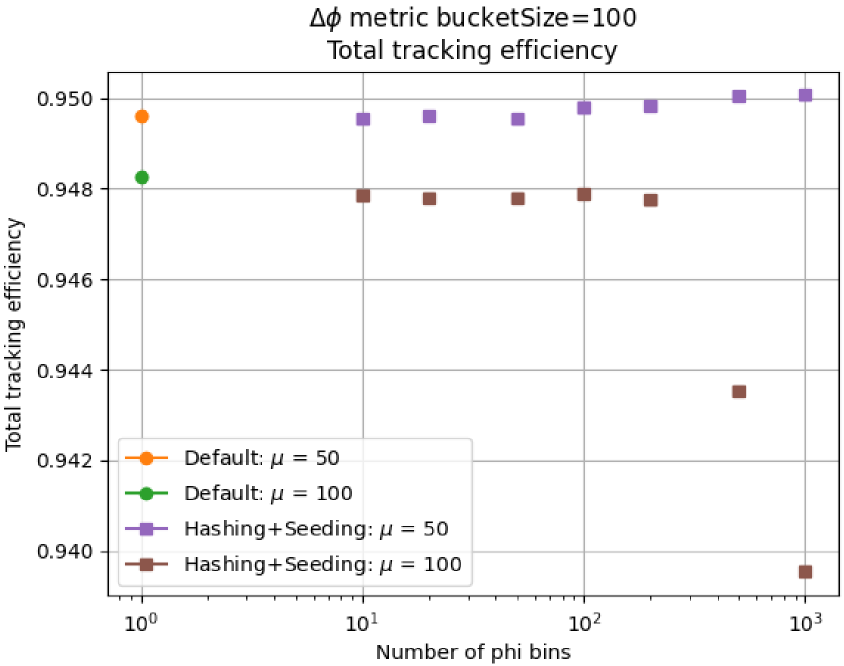


Always improve

Hashing φ bins: Timing and efficiency

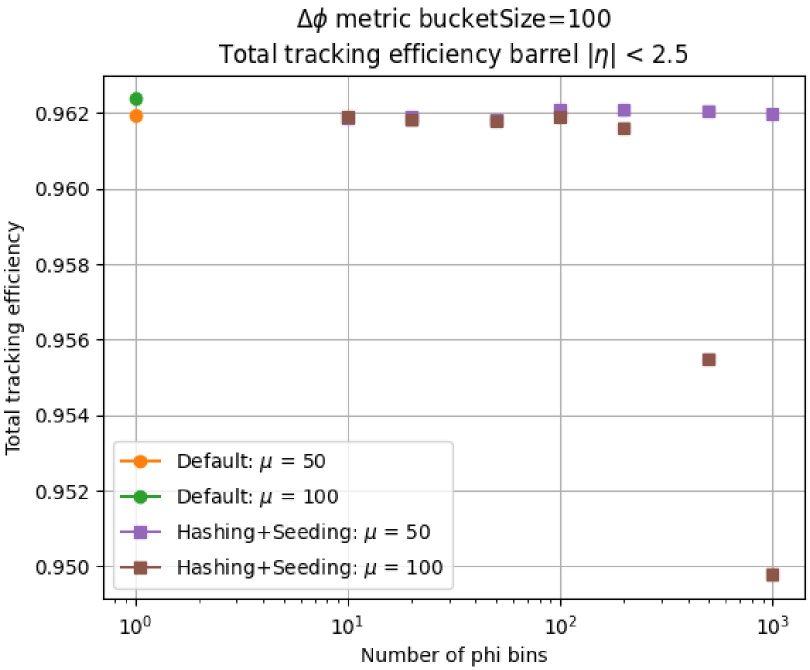


Similar running times

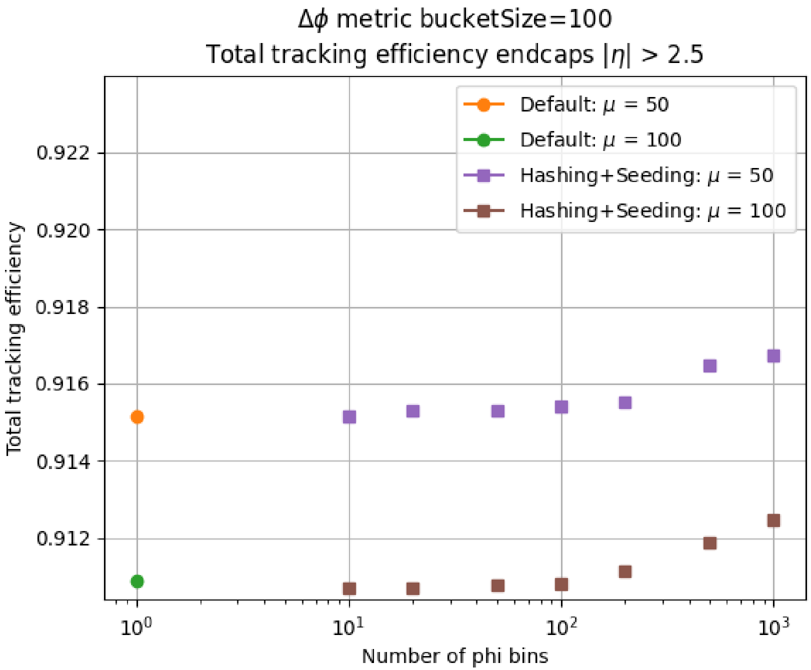


Small loss of efficiency

Hashing φ bins: Efficiency (detailed)

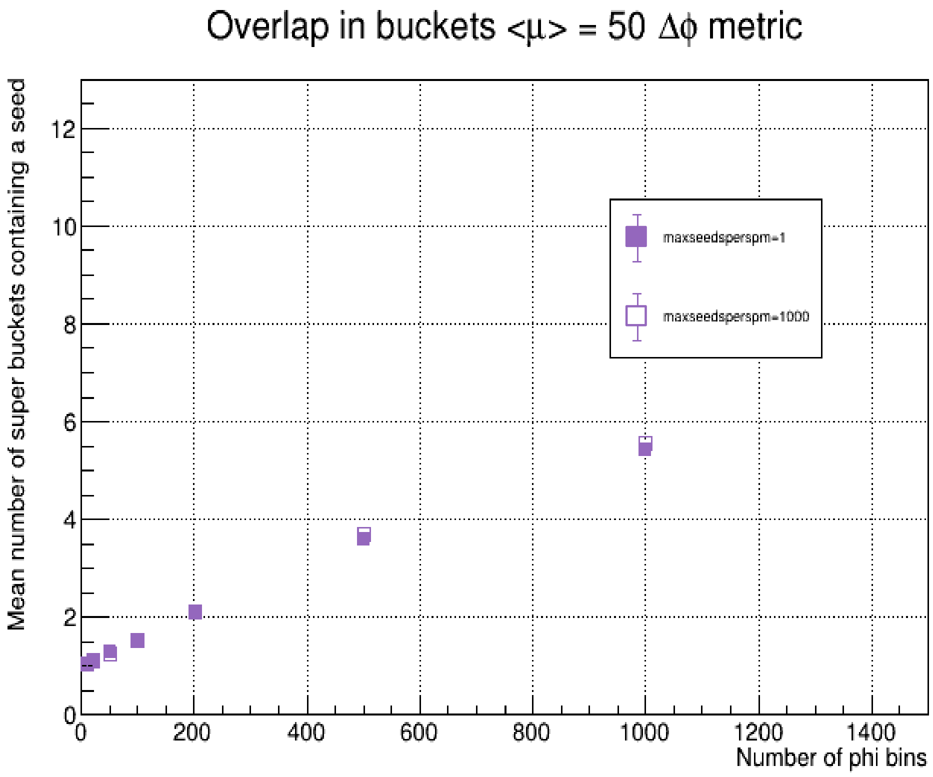


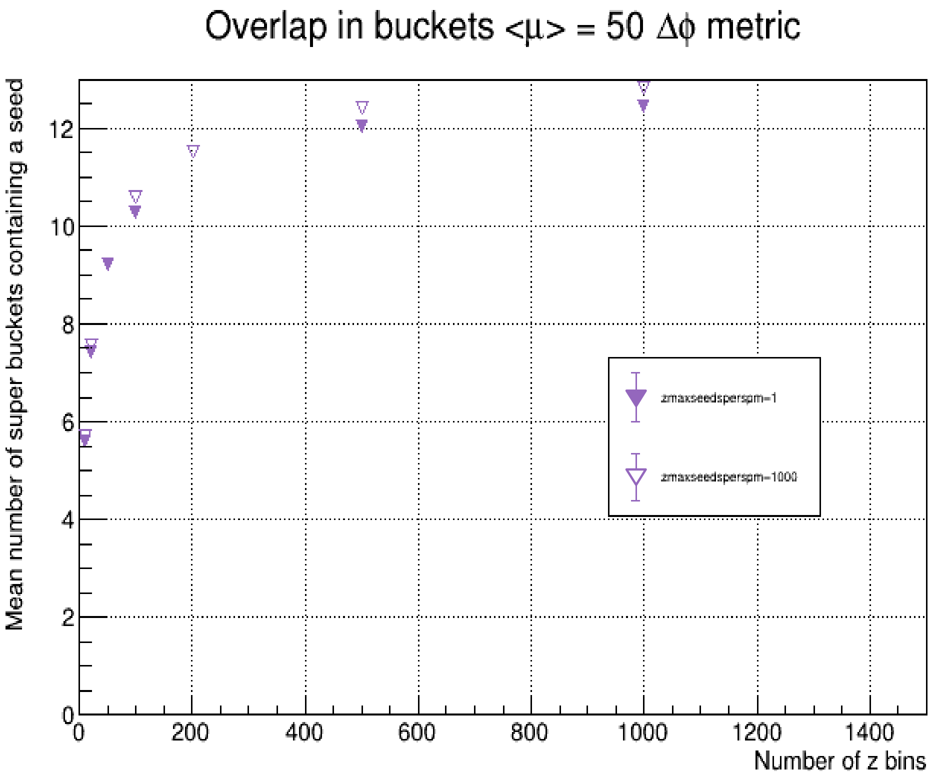
Drop of efficiency in the barrel



Better efficiency in the endcaps

Overlap in buckets





Less overlaps between buckets with φ binning

Some timing plots: Δφ

Some timing plots: ΔR

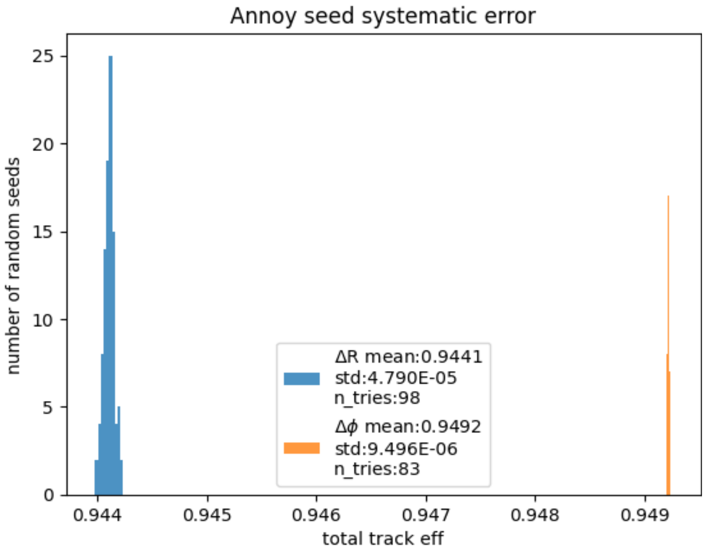
Seed finder configuration

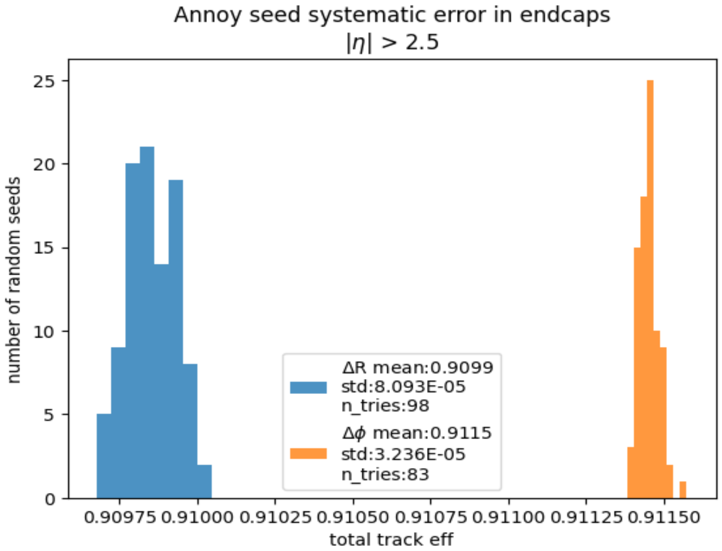
```
SeedfinderConfigArg = SeedfinderConfigArg(
        r=(None, 200 * u.mm),  # rMin=default, 33mm
        deltaR=(1 * u.mm, 60 * u.mm),
        collisionRegion=(-250 * u.mm, 250 * u.mm),
        z=(-2000 * u.mm, 2000 * u.mm),
        maxSeedsPerSpM=1,
        sigmaScattering=5,
        radLengthPerSeed=0.1,
        minPt=500 * u.MeV,
        bFieldInZ=1.99724 * u.T,
        impactMax=3 * u.mm,
        cotThetaMax=cotThetaMax # =1/tan(2×atan(e^(-eta)))
    )
```

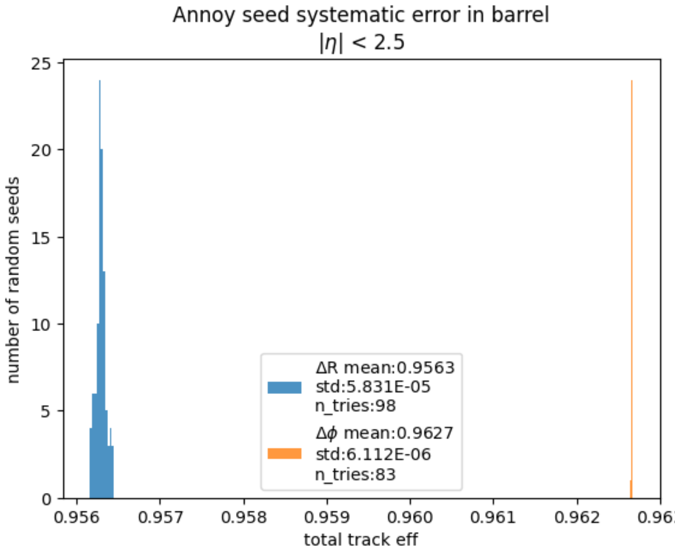
MaxSeedsPerSpM cut

- **Purpose:**
  - Reduce the number of seeds to expand to speedup the track finding
- **Idea:**
  - Only keep at most maxSeedsPerSpM+1 seeds sharing the same middle space point
- **Implementation:**
  - Uses a score to compare the seeds
  - The score is related to how close the impact parameter is to 0
- **Benefit:**
  - speedup and less memory used
- **Consequence:**
  - Loss of efficiency

Annoy random seed systematic error





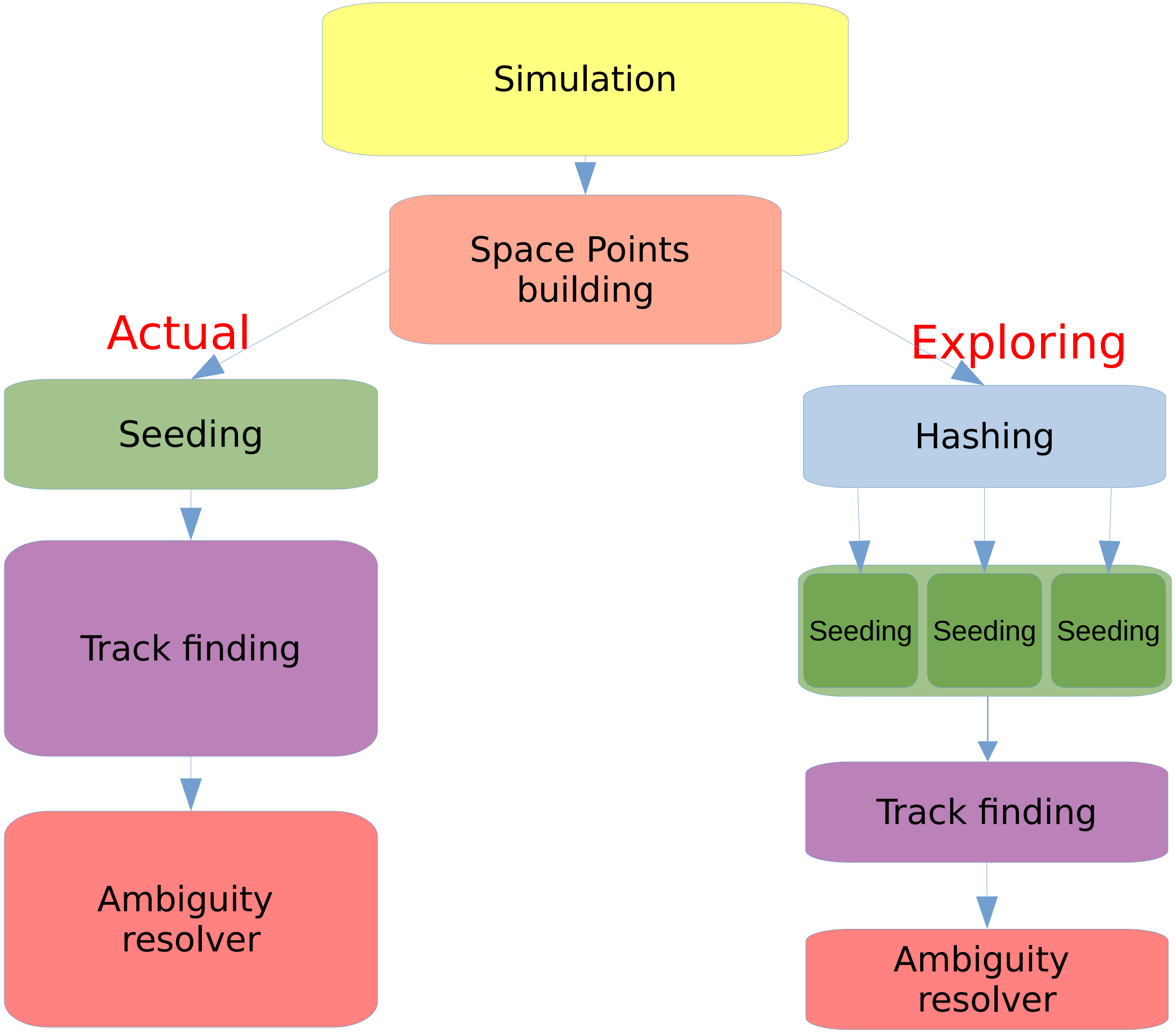


1000 events
in each try

BucketSize: 100
Mu: 50

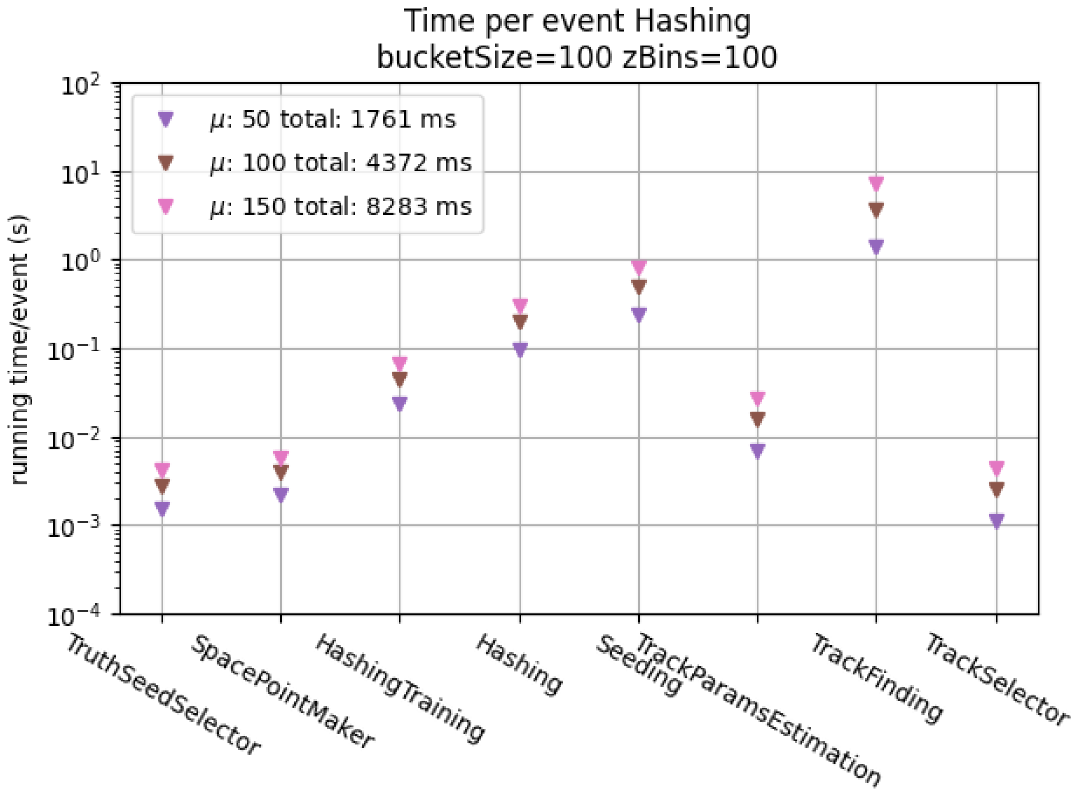
Δφ is better

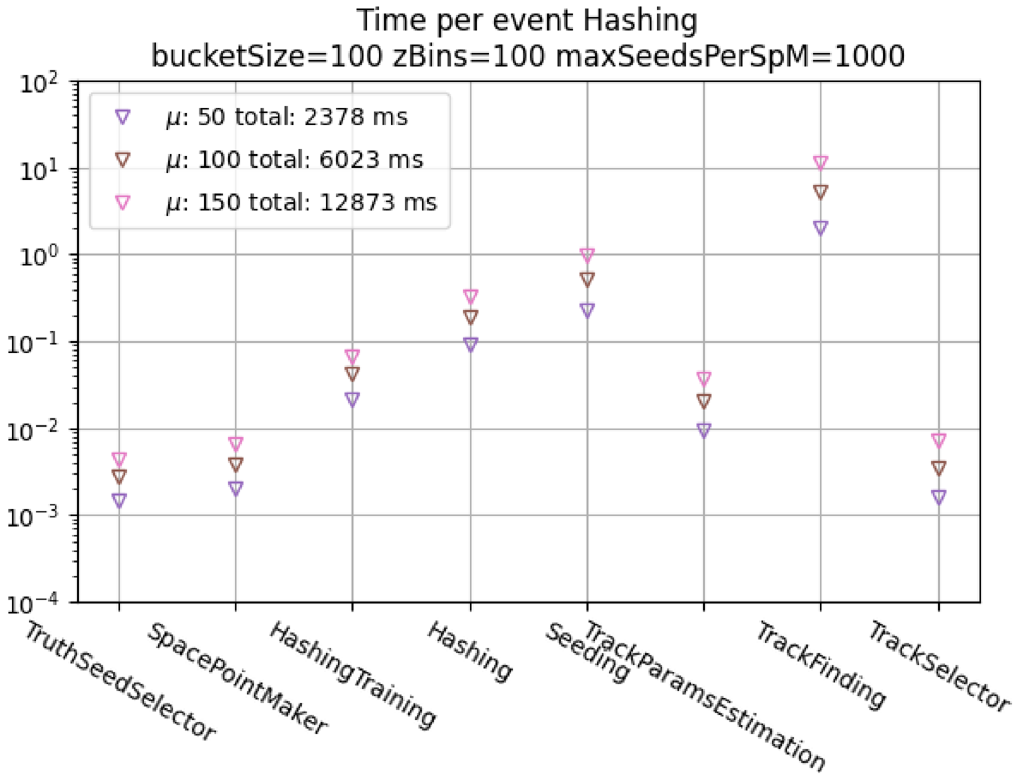
Approaches



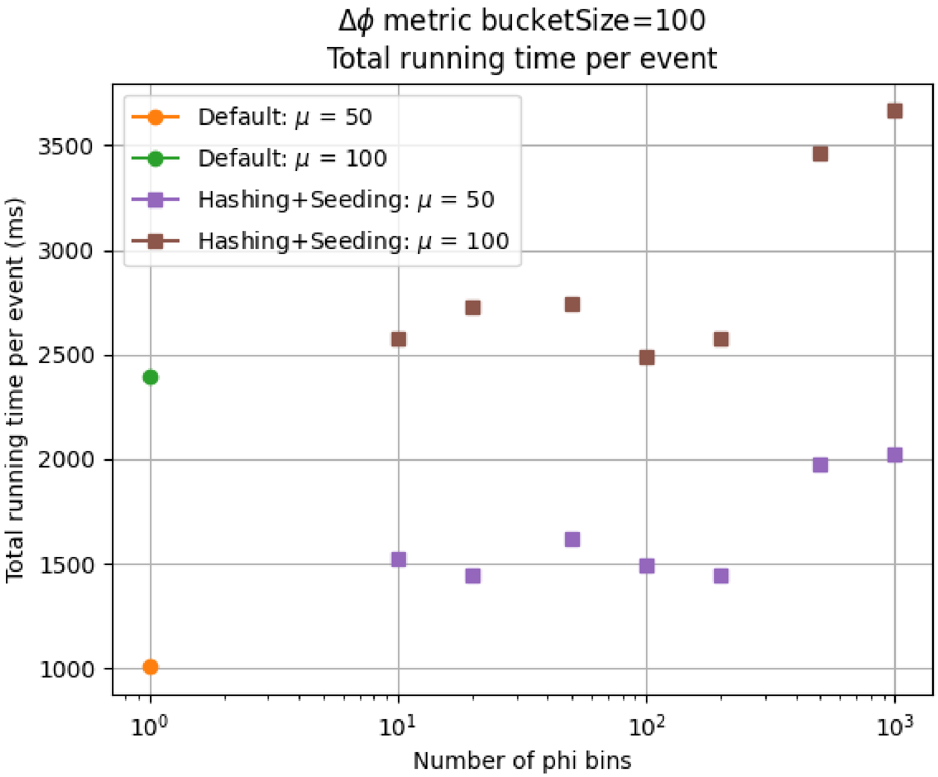
- Seeding parallelization
- Hashing groups space points into buckets
- Hashing reduces the number of space points at a time (focus on relevant space points) → less seeds per bucket

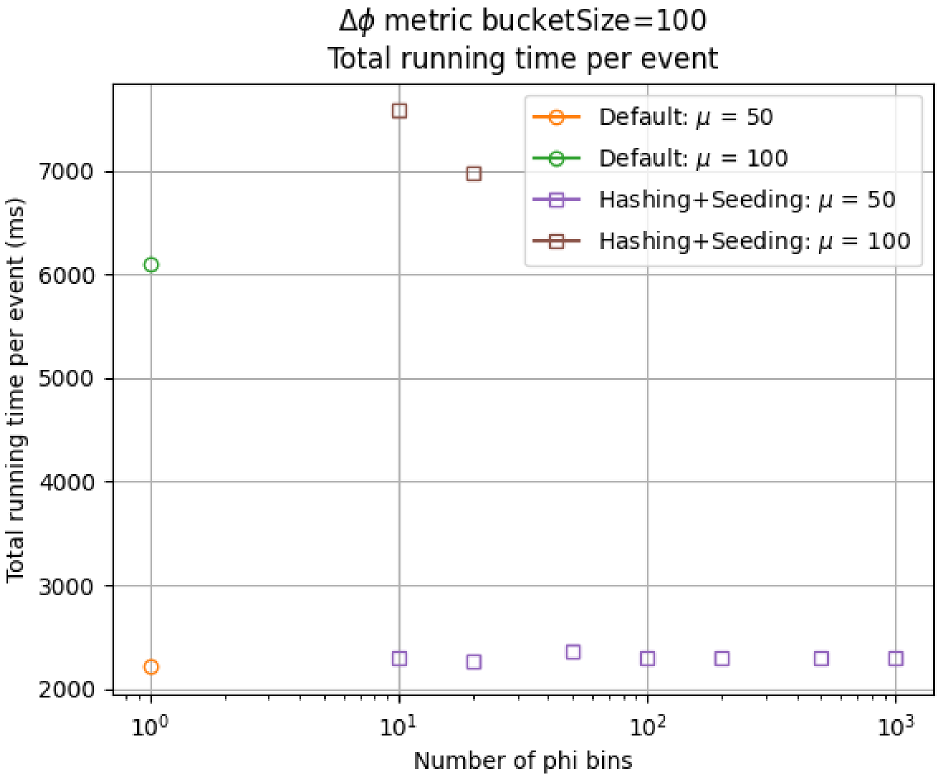
Running time no cut



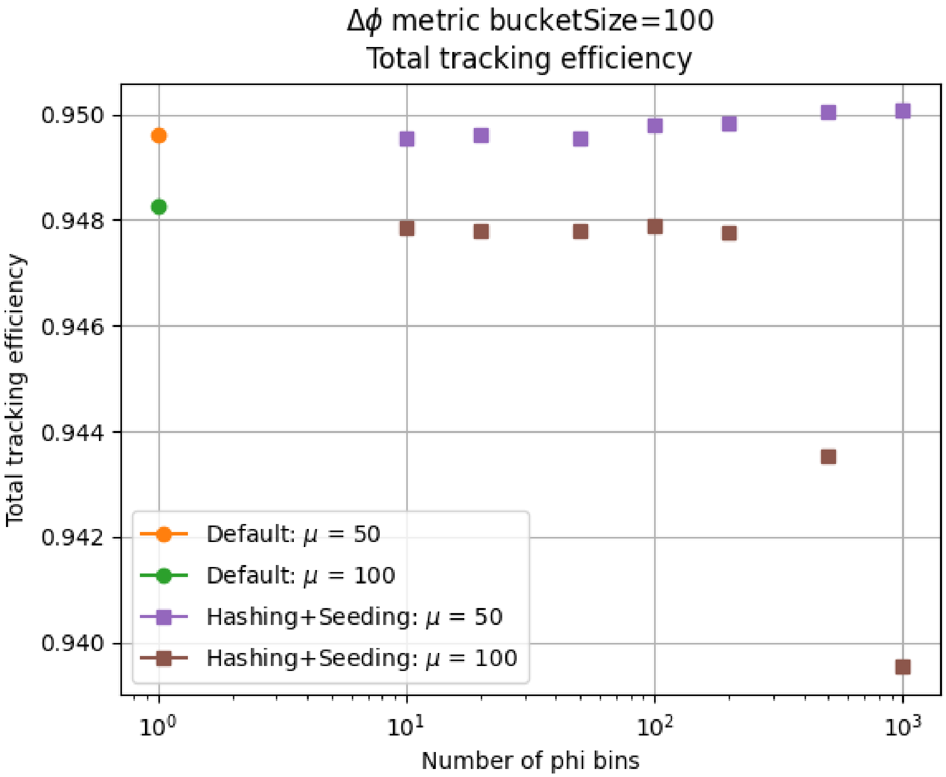


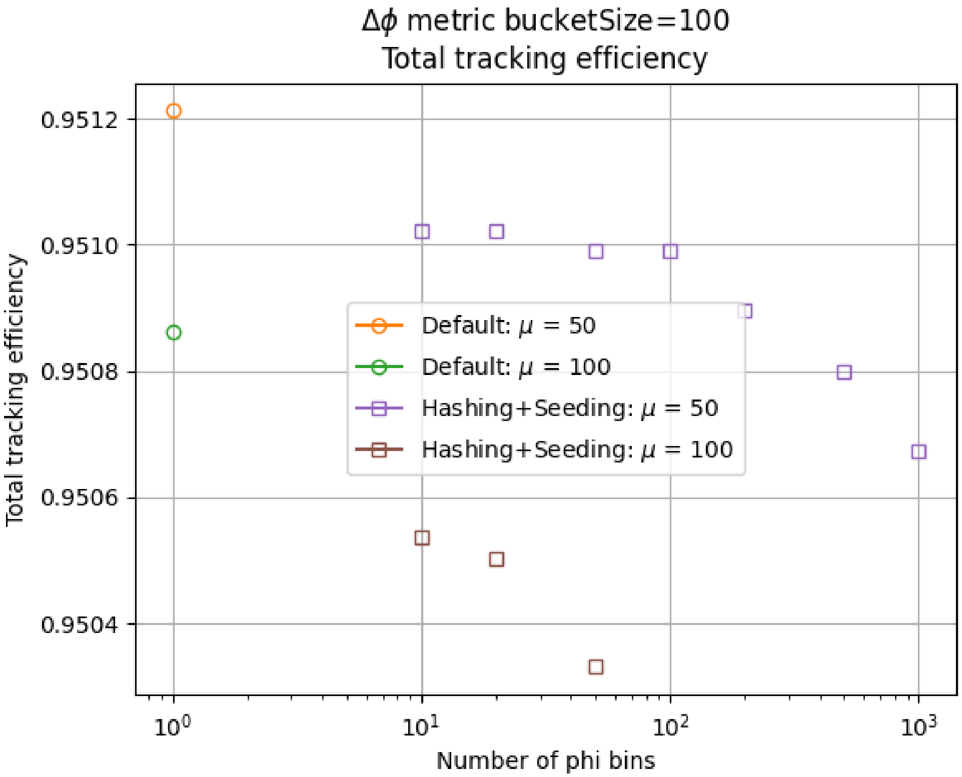
Phi bins: Timing



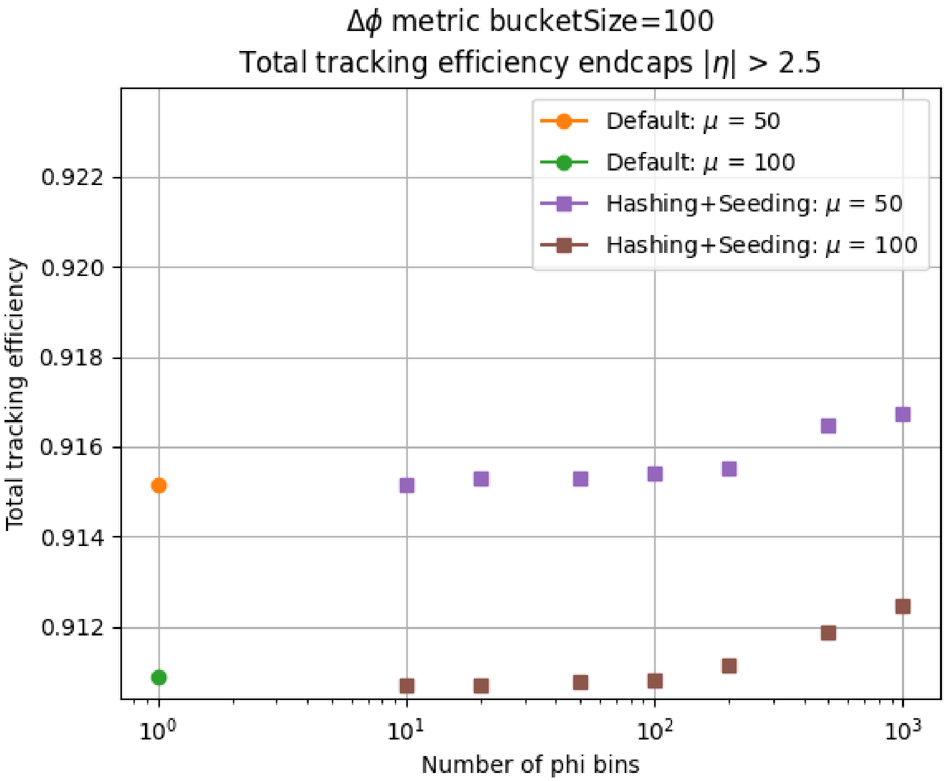


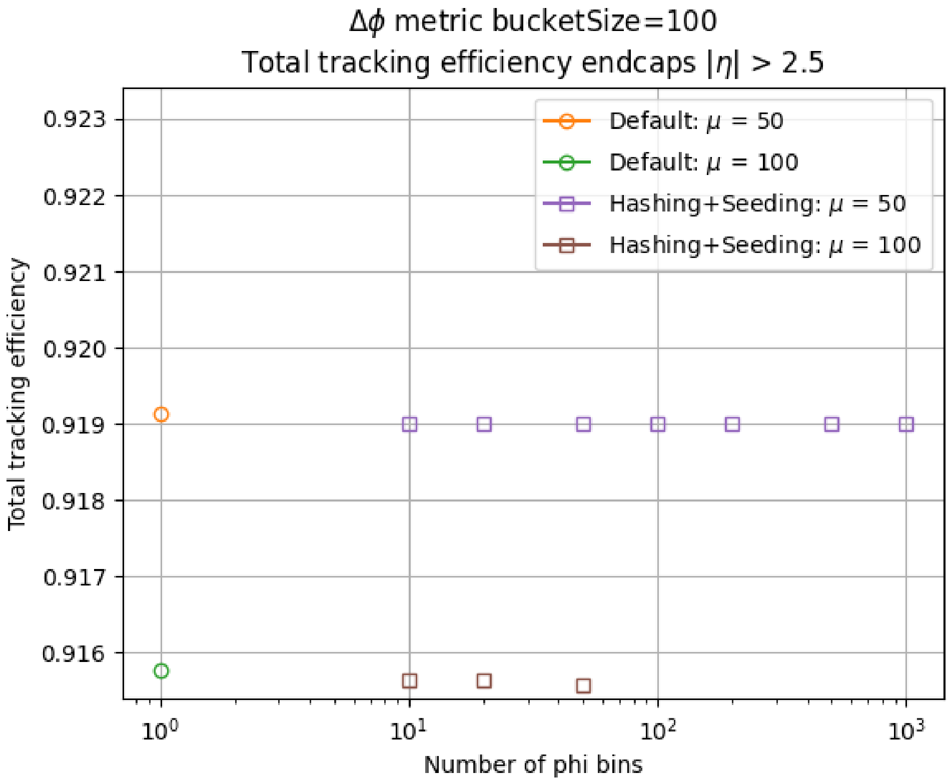
Phi bins: Tracking efficiency



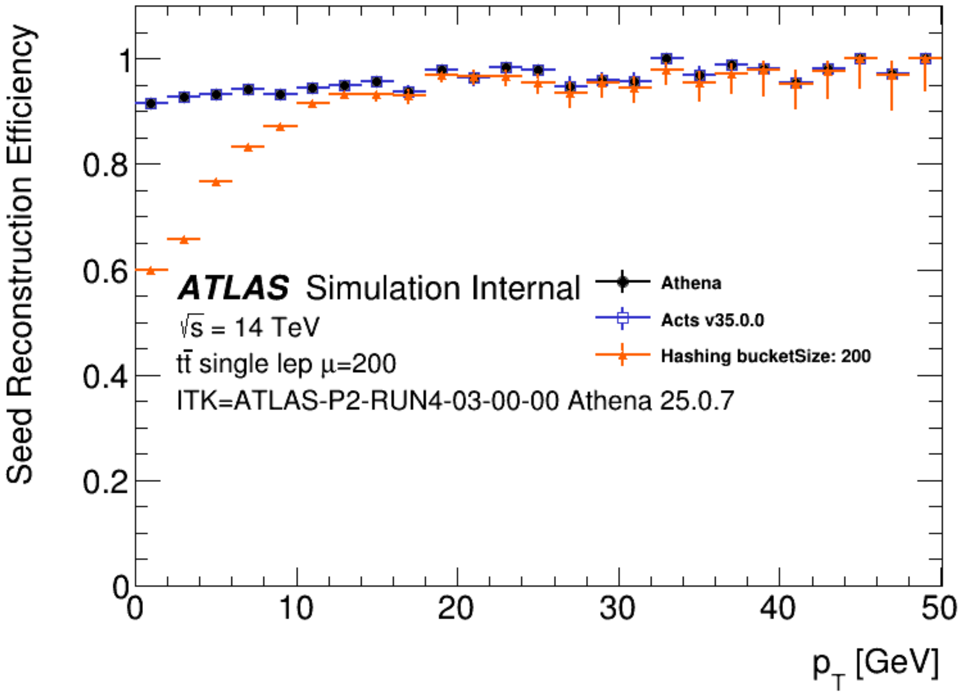


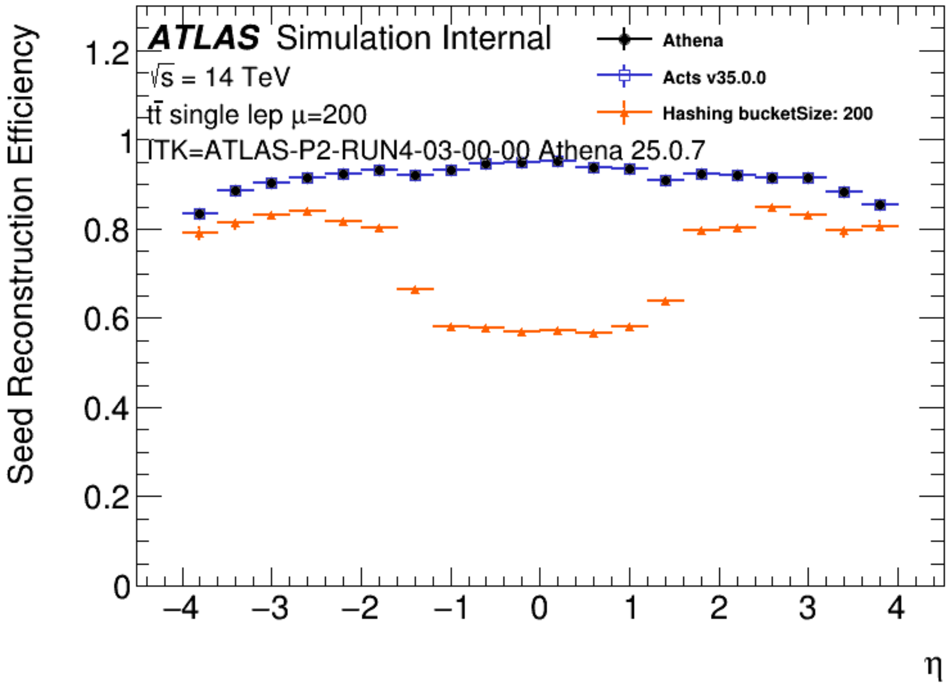
Phi bins: Tracking efficiency





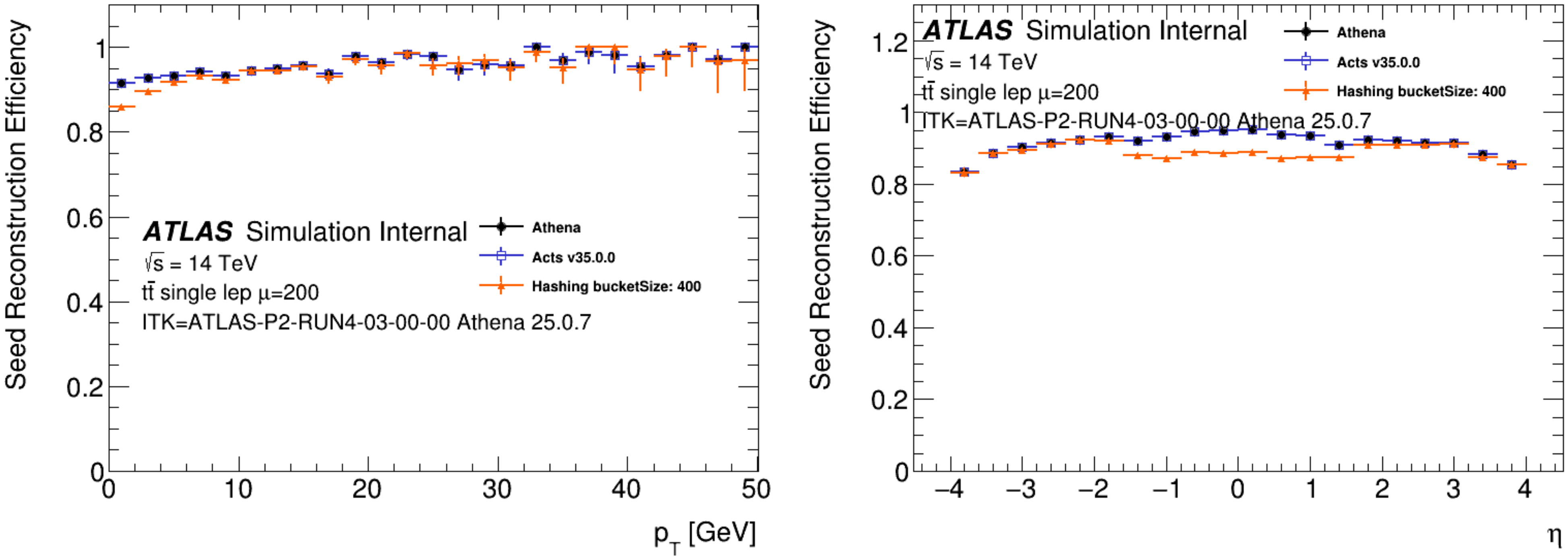
Hashing μ = 200





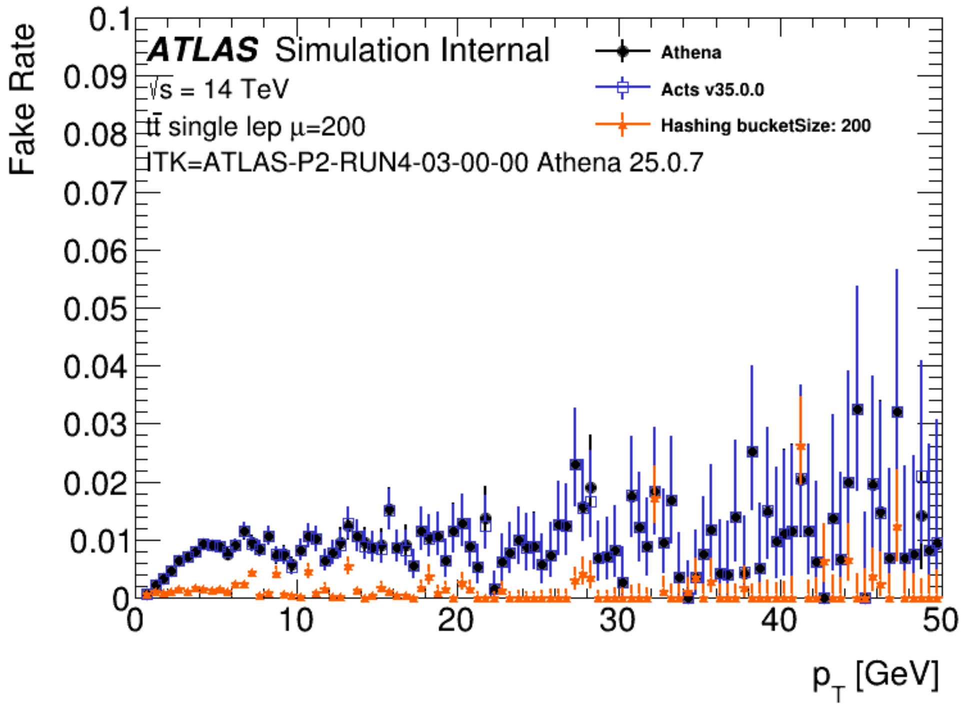
1000 events

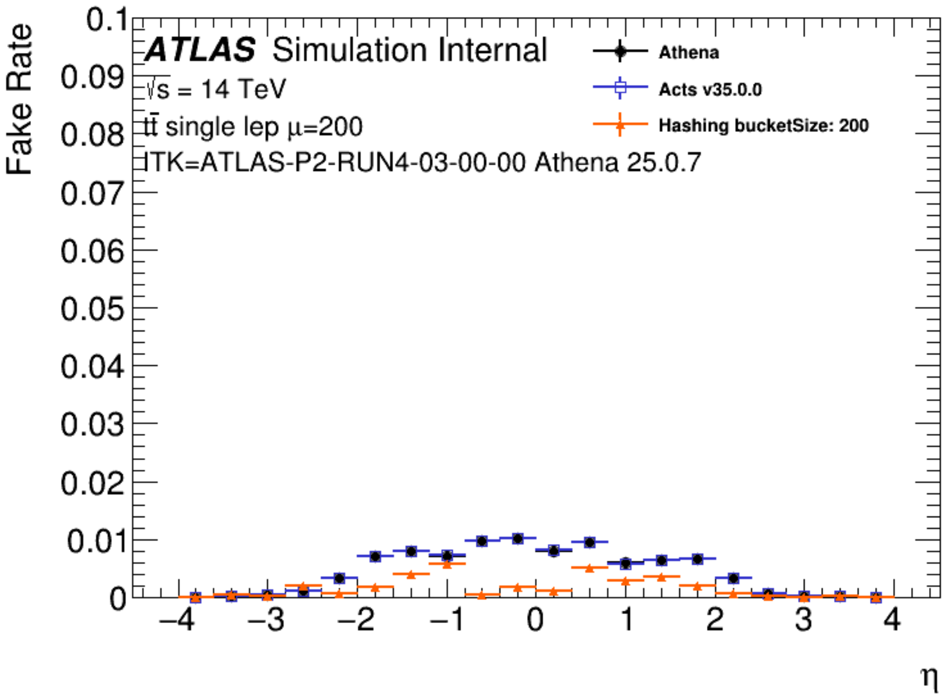
Hashing μ = 200



1000 events

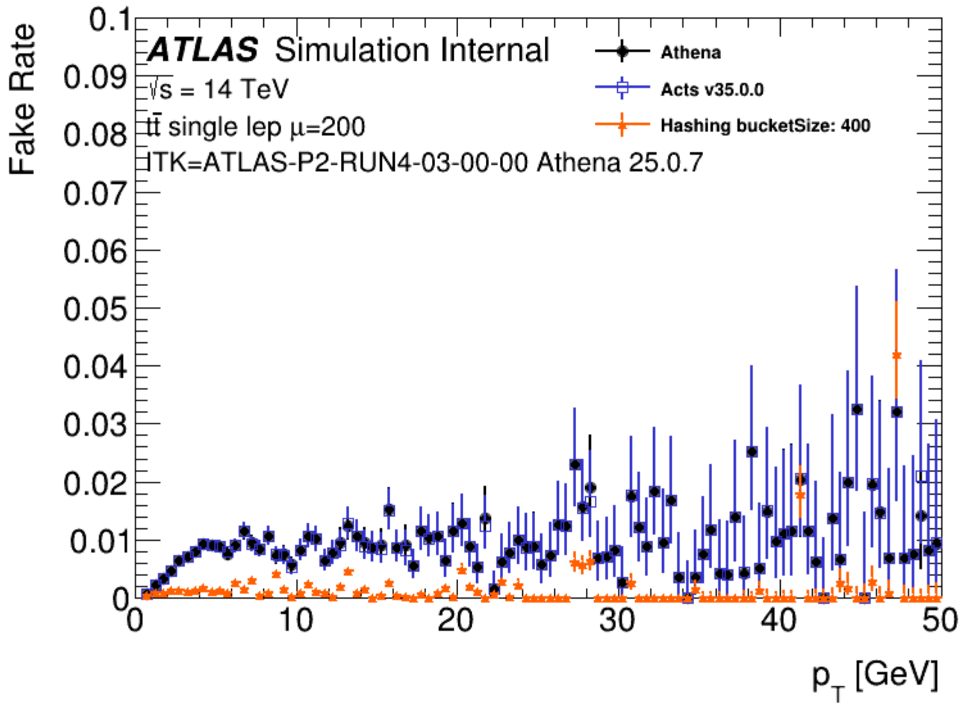
Hashing μ = 200

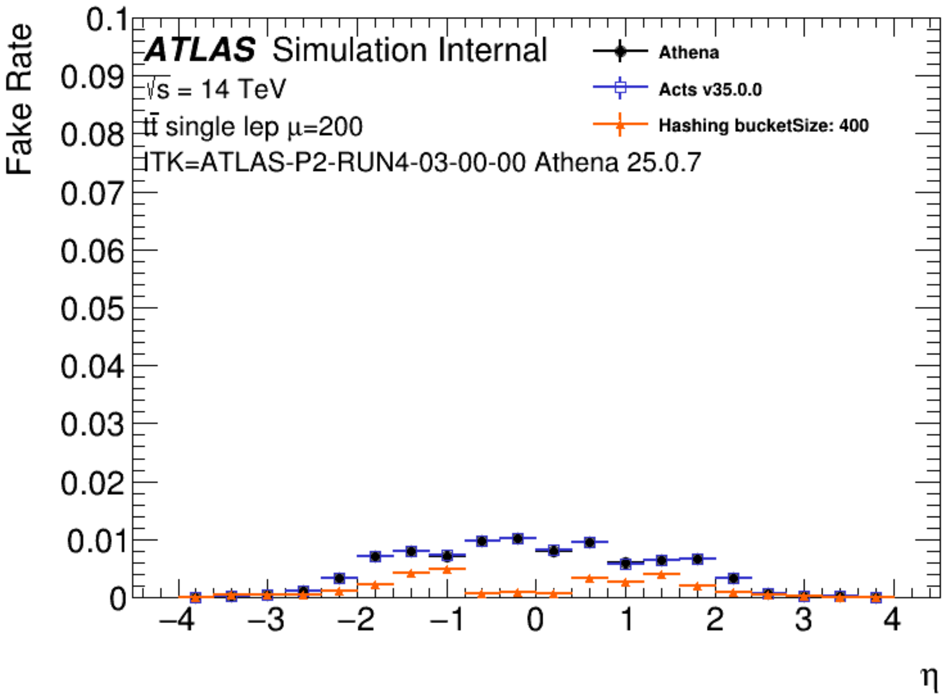




1000 events

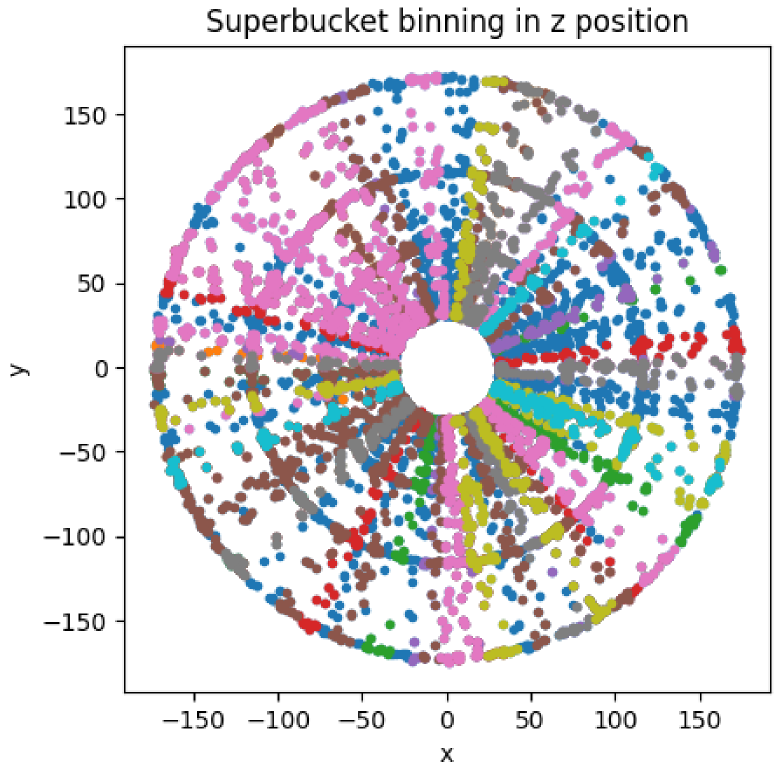
Hashing μ = 200

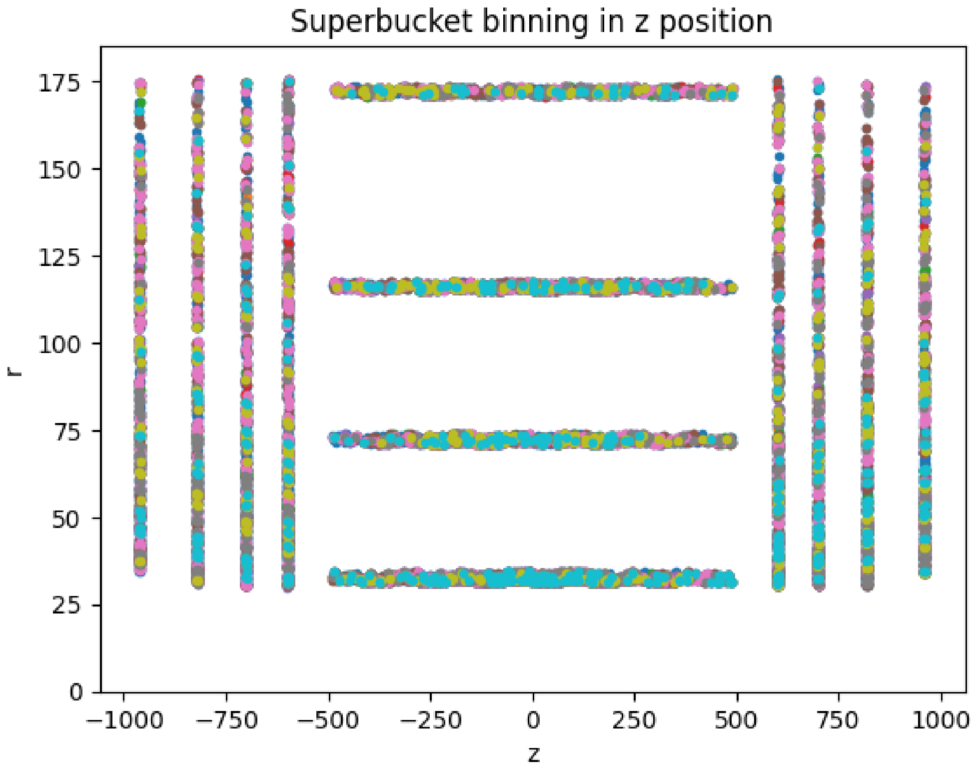




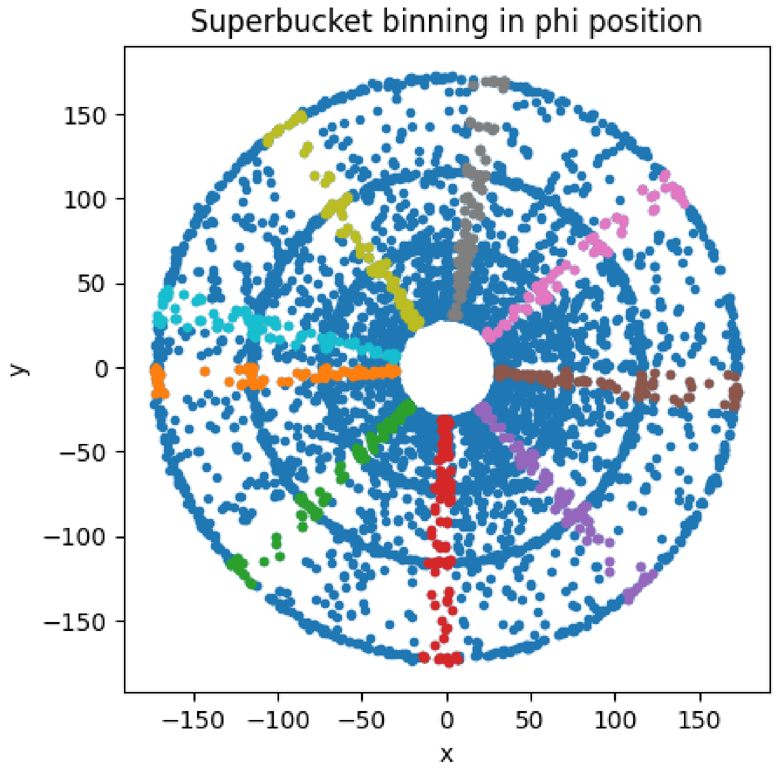
1000 events

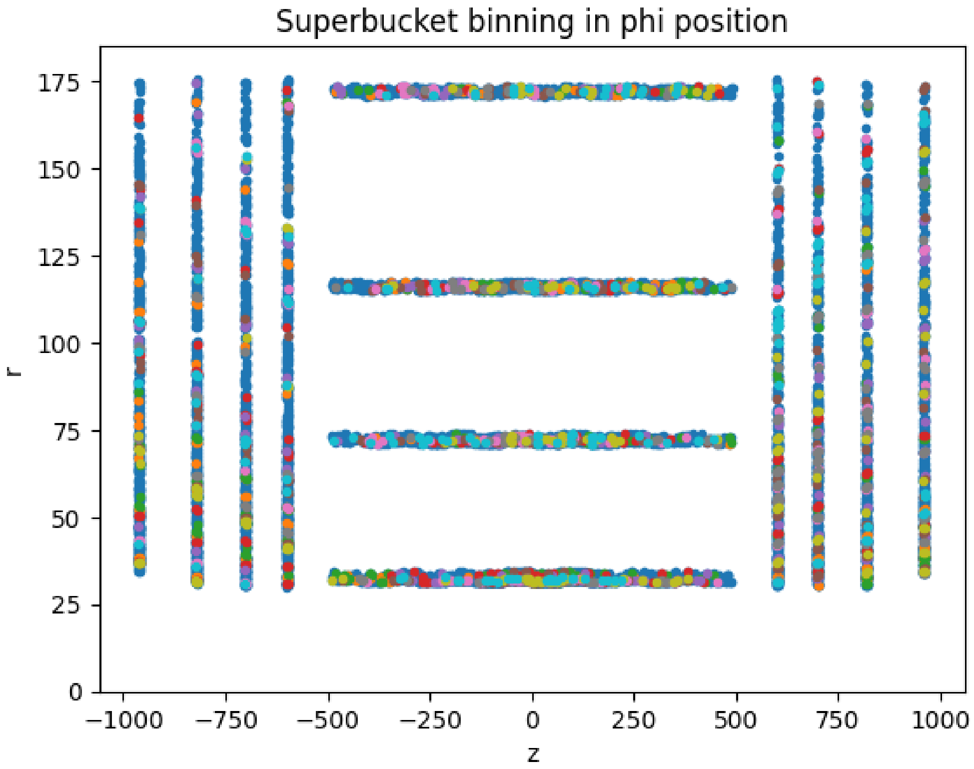
Superbucket binning in Z position





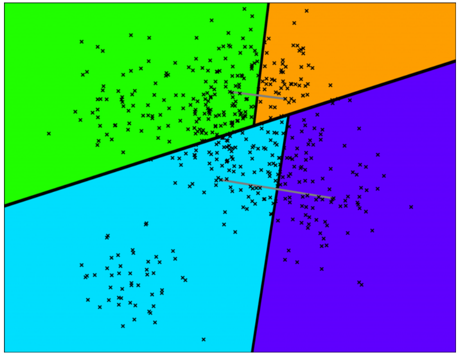
Superbucket binning in Phi position

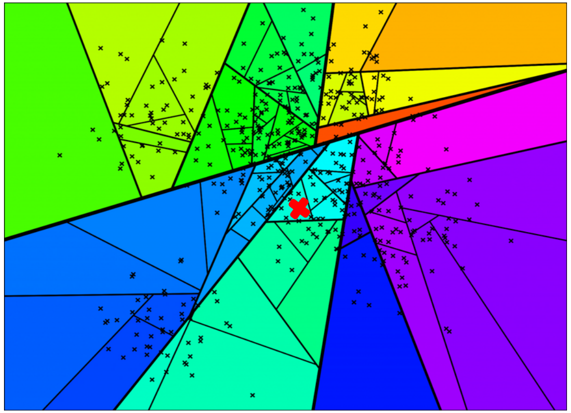




Annoy training

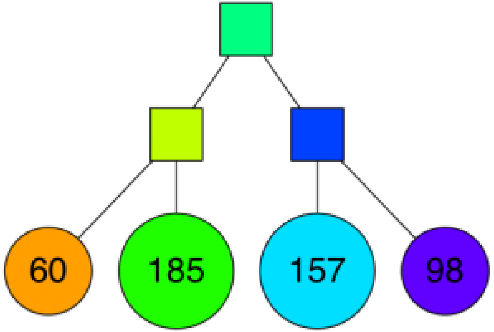
Space separation

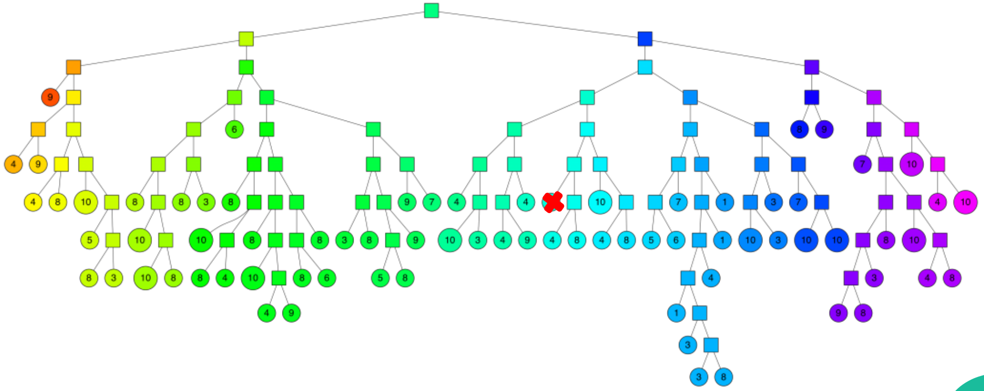




Takes two random points iteratively

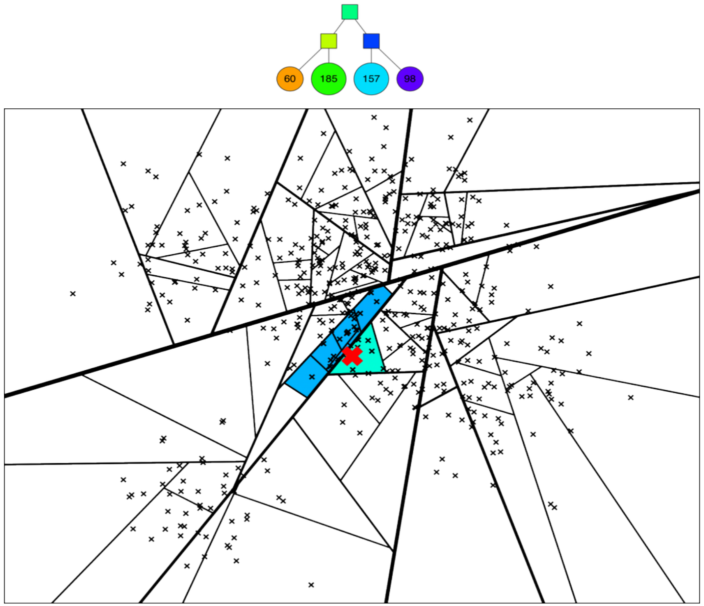
Corresponding binary tree



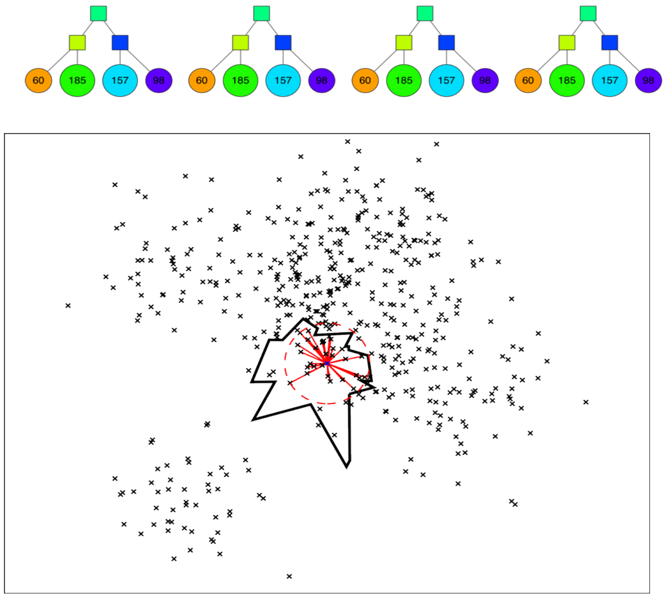


Annoy query

Approximation



Merge neighbor subspaces



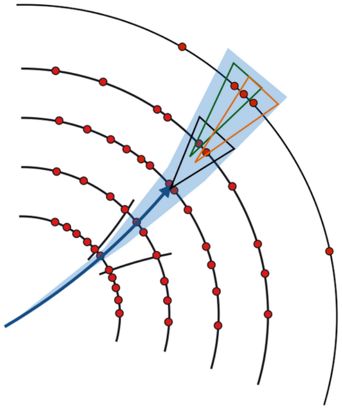
Union of trees' subspace

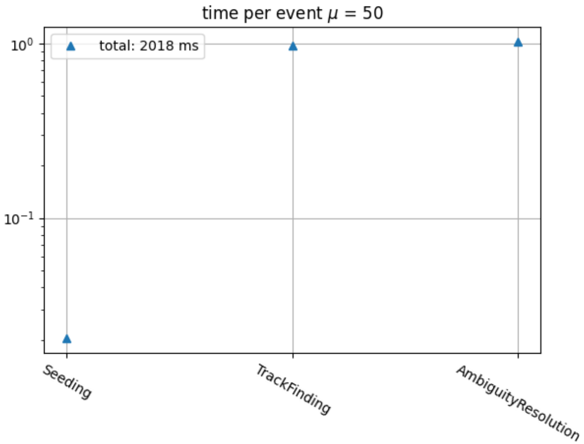
- Annoy tuning parameters: number of neighbors, number of trees, metric used, features used, number of subspace to look at

Combinatorial problem

Combinatorial Kalman Filter:

- Several possibilities of expanding the seeds at each layer → need to test them all
- Number of combinations increases exponentially with the number of layers





- **Every seed is expanded:**
  - Less seeds → less tracks → less bad quality and duplicated tracks

ACTS Poor man's Ambiguity resolver

How to get less seeds?

→ Remove the bad ones!

- How?
  - Current: Filter the seeds + detailed optimisation
  - My work: Build the seeds differently

Seeding: Skipping triplets check with sets

Overlap indicator

- **Event 98: Hashing mu=50 bucketSize=100**
- **9860 Space Points → ~100.000.000 possible doublets**

Set name	Set size	nSkipped	Ratio
Bad bottom	**24.433.199**	322.132.498	13,18
Good bottom	**3.592.664**	63.294.324	17,62
Bad top	**30.363.102**	392.248.454	12,92
Good top	**4.973.975**	91.166.619	18,33
Triplets	**18.204.058**	269.635.750	14,81
Seeds	**5.623**	x	x

Total running time x1.5