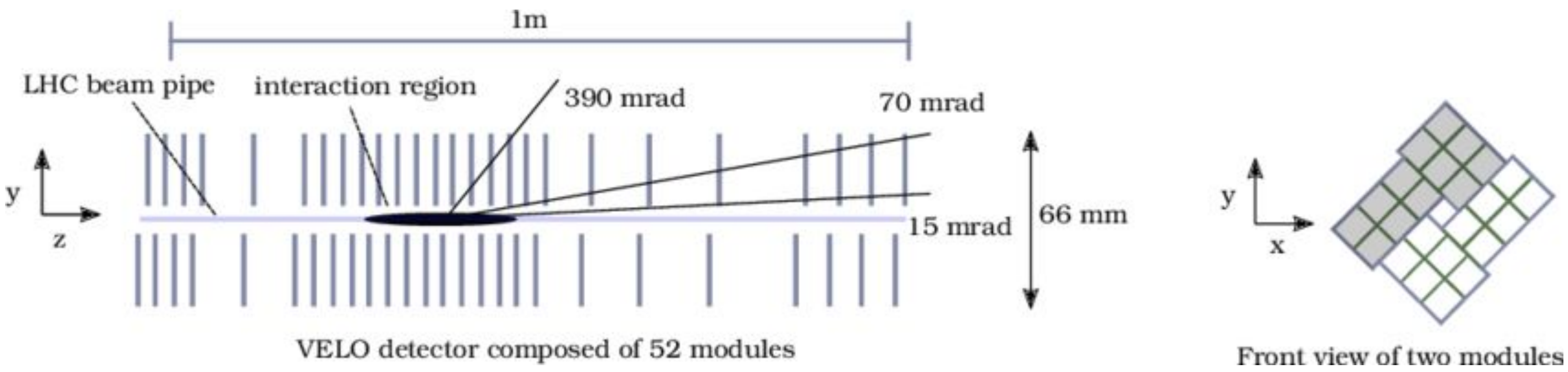
# Timescans for run5: Preliminary results with 4D tracking optimisation in LHCb

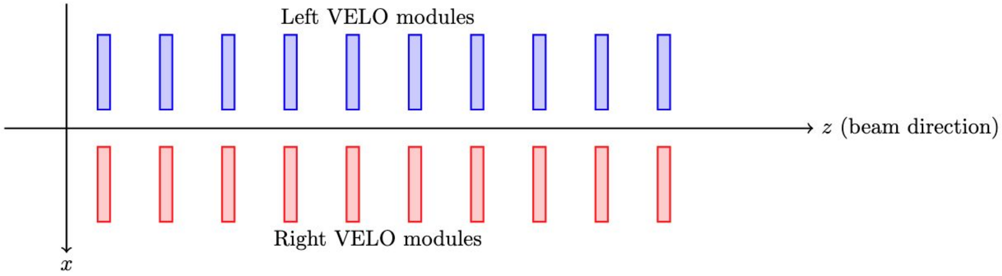
# The VELO detector – schematic

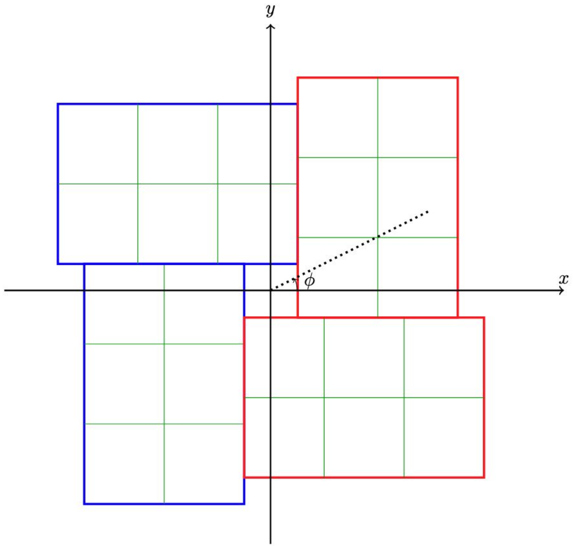



Pérez, Daniel & Neufeld, Niko & Núñez, Agustín. (2022). Search by triplet: An efficient local track reconstruction algorithm for parallel architectures.

Preliminary change for Upgrade 2 (U2): 26 → 32 stations

# The VELO detector – schematic(er)



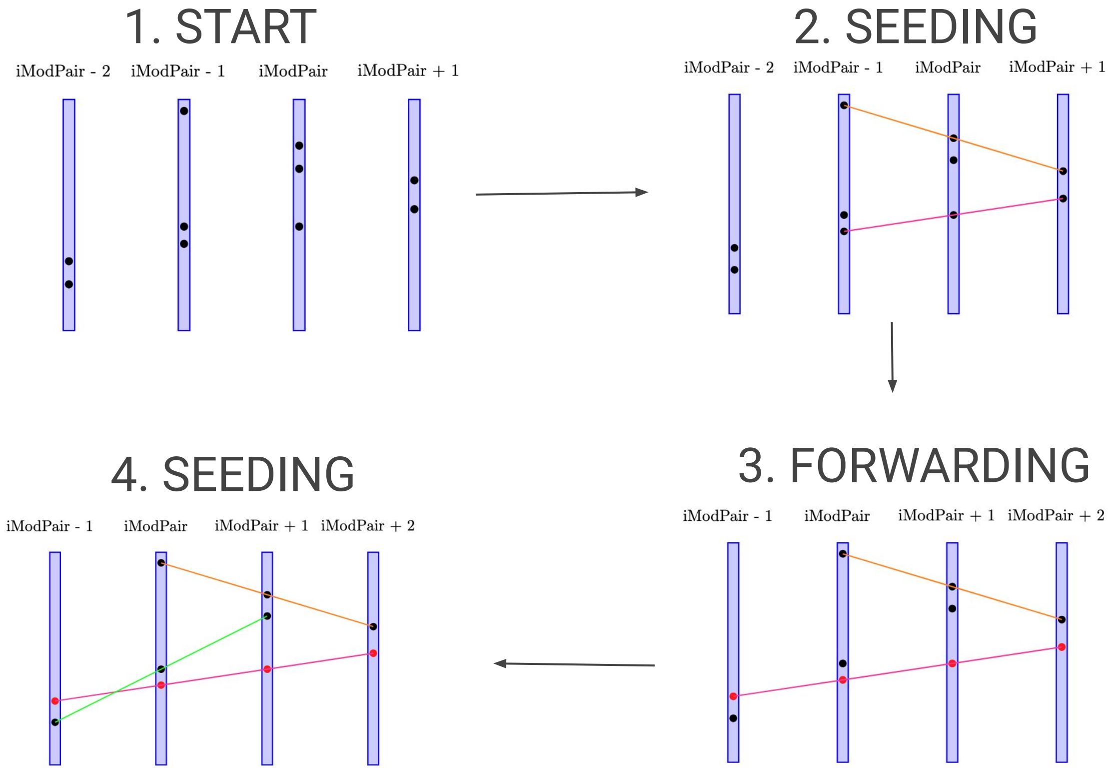

Module Pair
(station)

- NOTE: Modules are NOT perfectly aligned in reality (purposefully)
- HOWEVER, the above is "good enough" for this presentation

# Metadata of runs

- 20000 events with ~50M MCPs produced
- No encoding or decoding: every fired pixel counted as hit
  - **Clustering not yet available** (coming very soon)
  - 4D Spacepoints directly used

# One slide overview of LHCb tracking

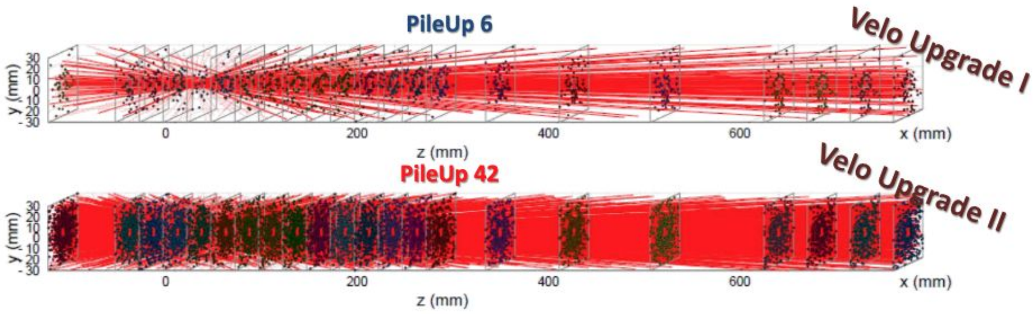


# Efficiencies, ghosts and clones

3D = Out Of The Box

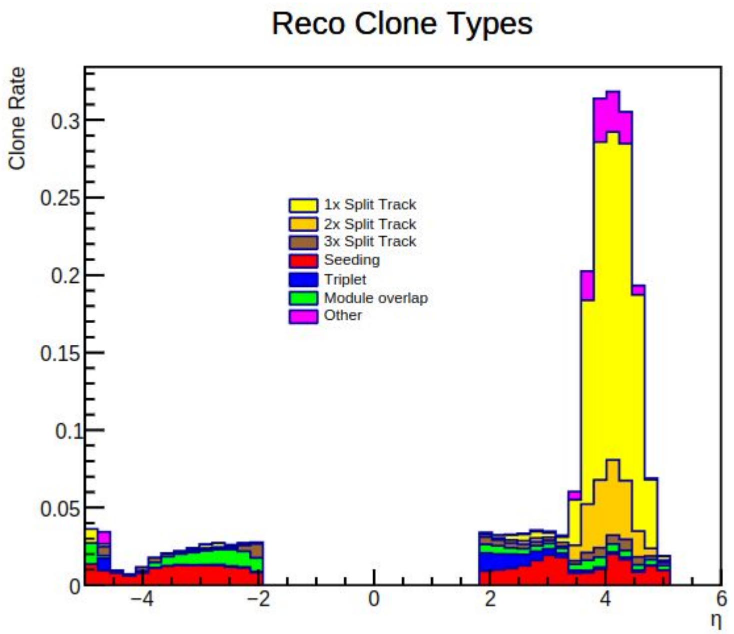
| | Run 3 Baseline (Forward) | Run 5 3D (Forward) | Run 5 3D (Backward) |
|---|---|---|---|
| Efficiency | 98.32% | 98.69% | 97.31% |
| Ghost Rate | 2.12% | 3.24% | 2.50% |
| Clone Rate | 2.87% | 10.31% | 2.30% |

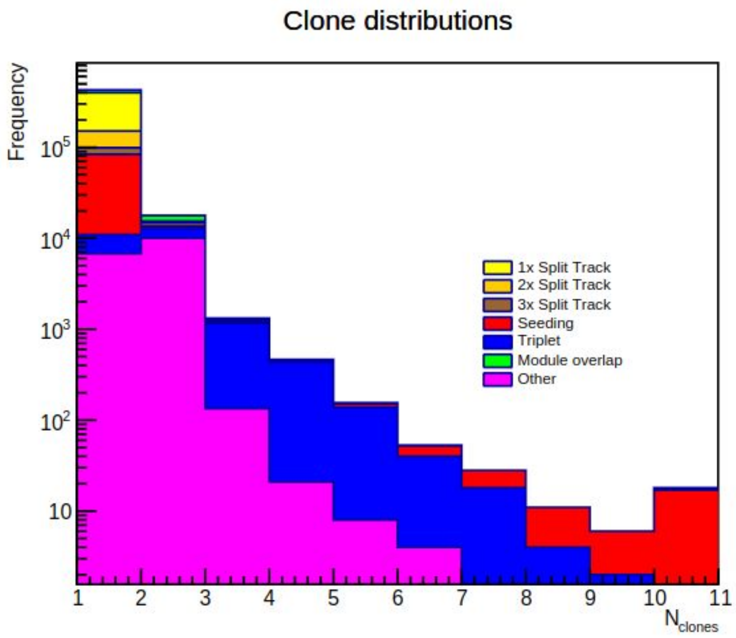
**Remember ×10 occupancy!**




Very high clone rate…
also asymmetrical?

# Clone rates 3D Algorithm in run5

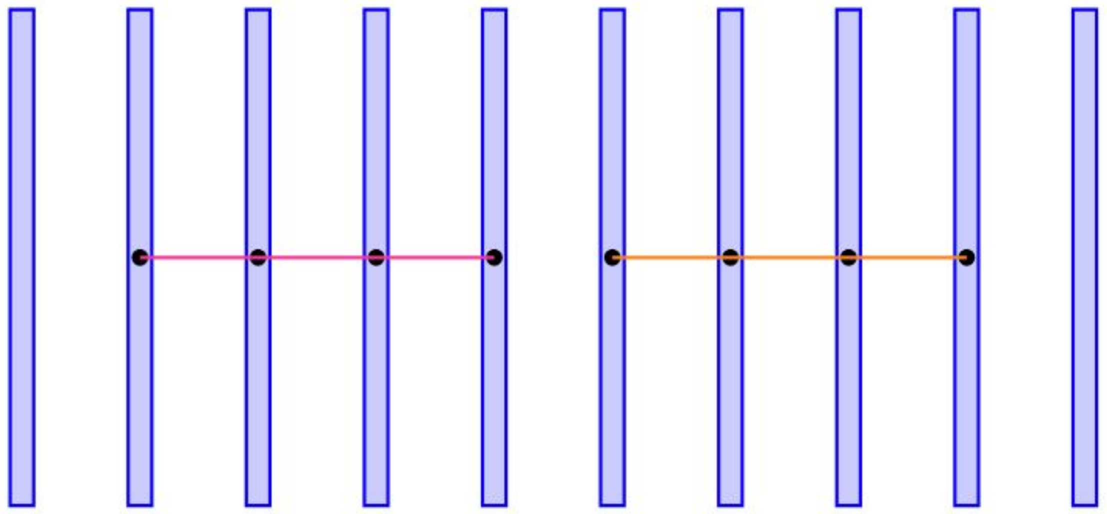



→ There are MANY once-split Split Track Clones!

# Different types of clones

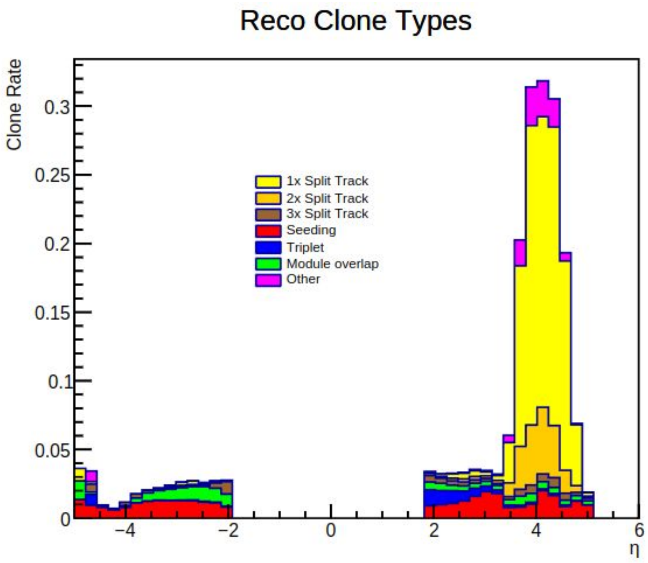
1. Split Track Clones: Leading Order (LO)



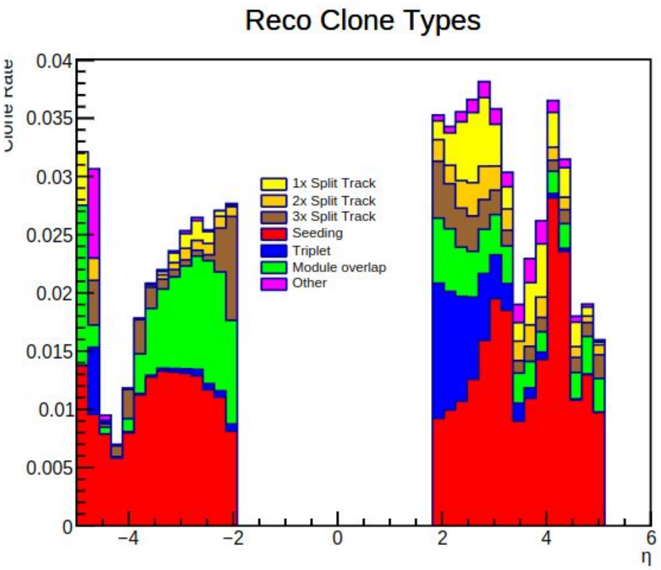
8 hit (MC) track

# Clone rates with deflection z-dependence (still 3D)

Out of the box




Max Deflection = **k × Δz²**

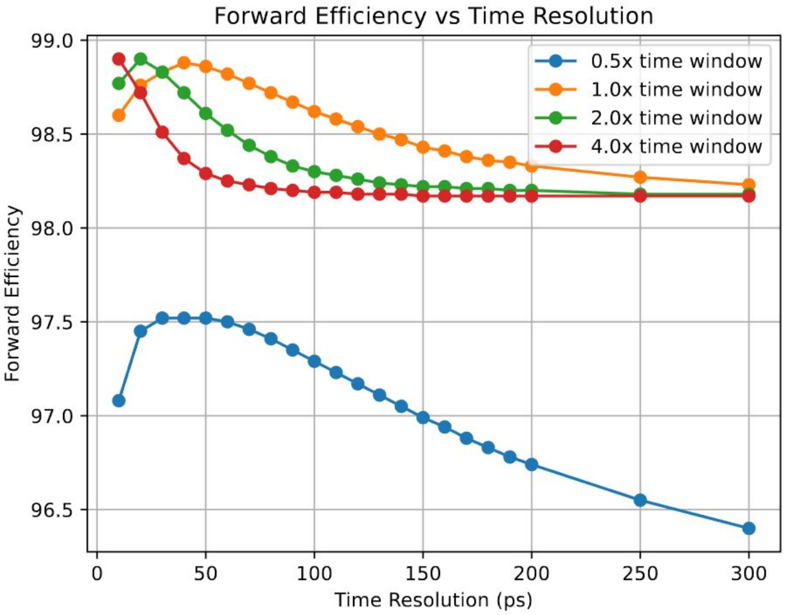


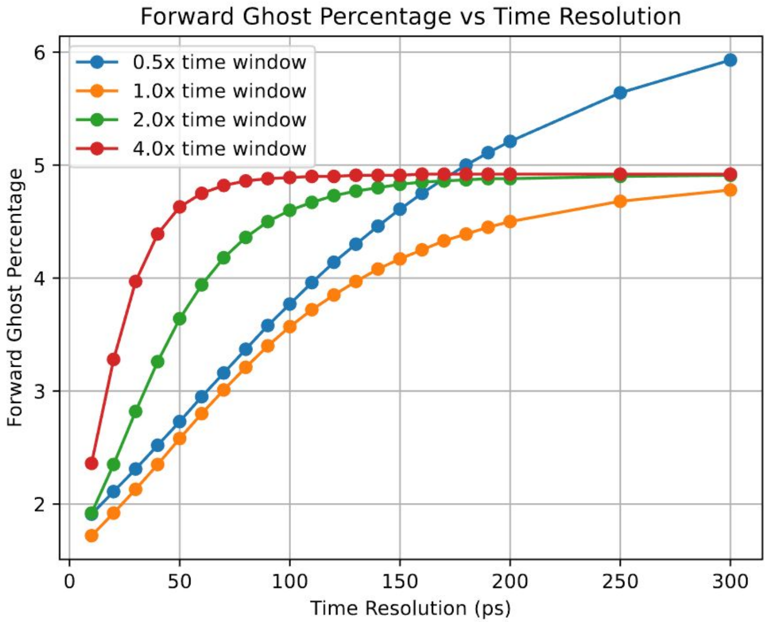

**$k = 2\times10^{-5}$**

# Time based cuts on seeding and forwarding

- Kalman filter: Estimate + Measurement weighted together
- Measurement must lie within specified range of readout resolution
  - How much?
  - Starting point: $3\sqrt{2}\sigma$ in seeding, in forwarding $3\left(1+1/N_{hits}\right)^{\frac{1}{2}}\sigma$
  - Based on simple error propagation (backup slides). Best choice?

# Forward efficiency and fake rate wrt pixel time resolution




1x describes the standard windows from before

# Algorithm tweaks compared with time cuts (50ps, 1x)

| **FORWARD** | Run 3 Baseline | Run 5 3D | Run 5 3D ($z^2$) | Run 5 4D ($z^2$) |
|---|---|---|---|---|
| Efficiency | 98.32% | 98.69% | 98.27% | 98.86% |
| Ghost Rate | 2.12% | 3.24% | 4.40% | 2.58% |
| Clone Rate | 2.87% | 10.31% | 2.90% | 3.54% |

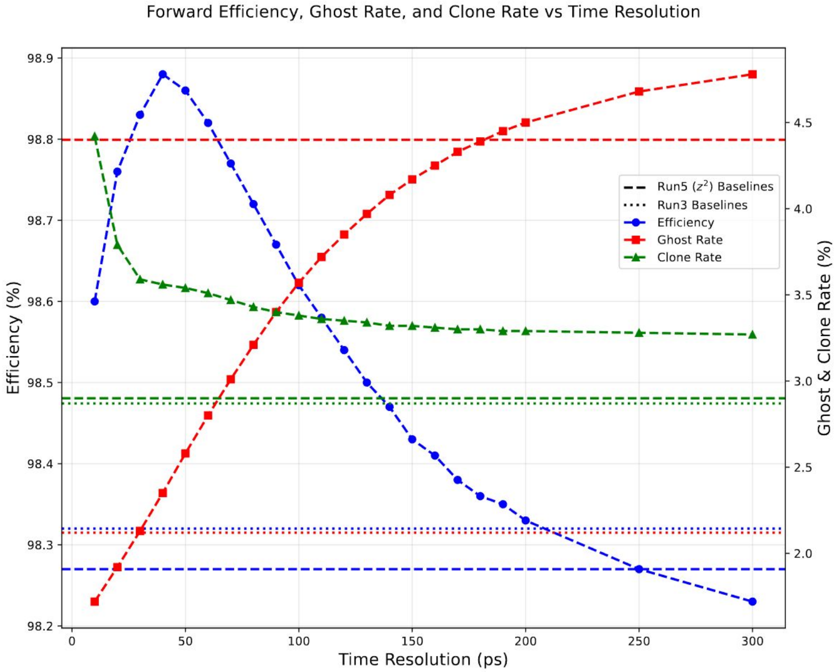
| **BACKWARD** | Run 3 Baseline | Run 5 3D | Run 5 3D ($z^2$) | Run5 4D ($z^2$) |
|---|---|---|---|---|
| Efficiency | 98.32% | 97.31% | 96.87% | 98.11% |
| Ghost Rate | 2.12% | 2.50% | 2.86% | 1.62% |
| Clone Rate | 2.87% | 2.30% | 2.27% | 2.28% |

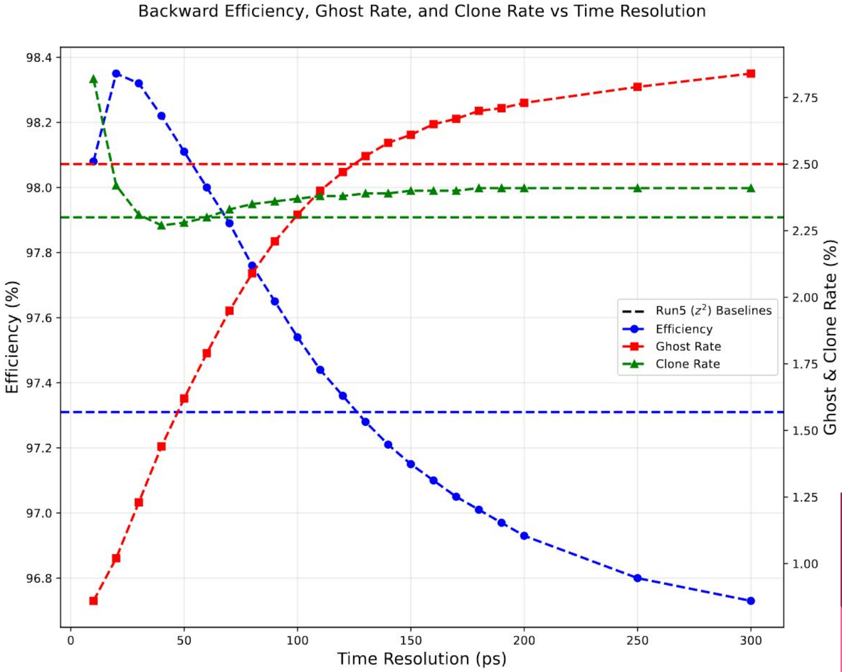
No forward/backward distinction in Run 3 benchmarks!

Why does clone rate get worse only for forward?

# Timescans: What about other resolutions?

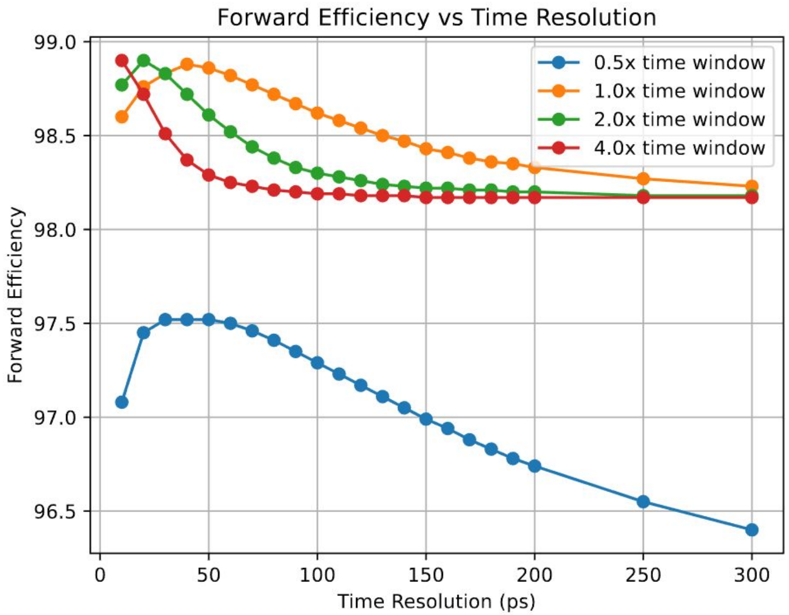
Important note: This is for regular 1x time window!

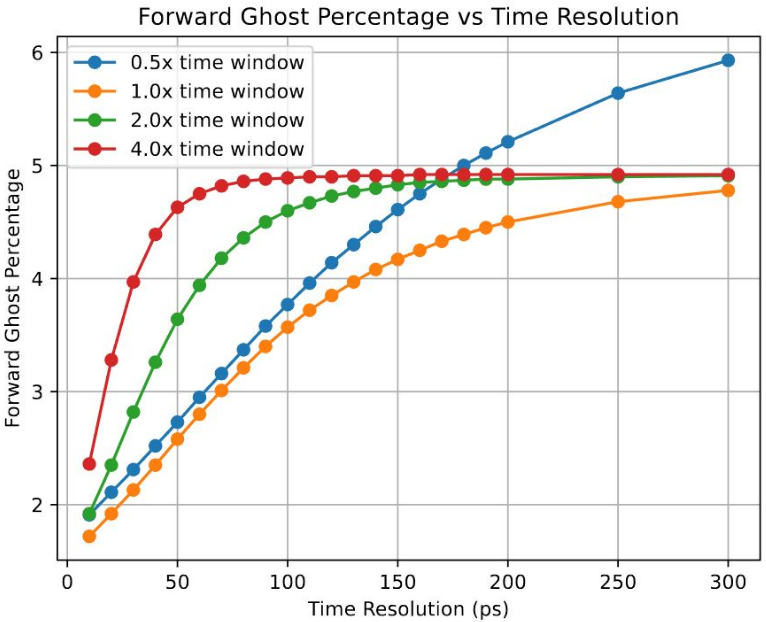


# Timescans: What about other resolutions?

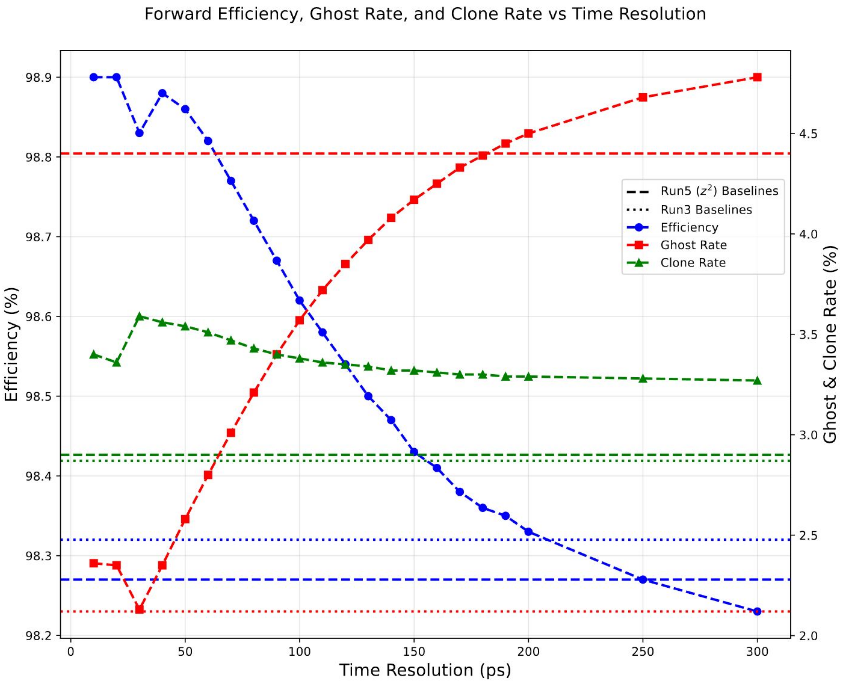
Recall:

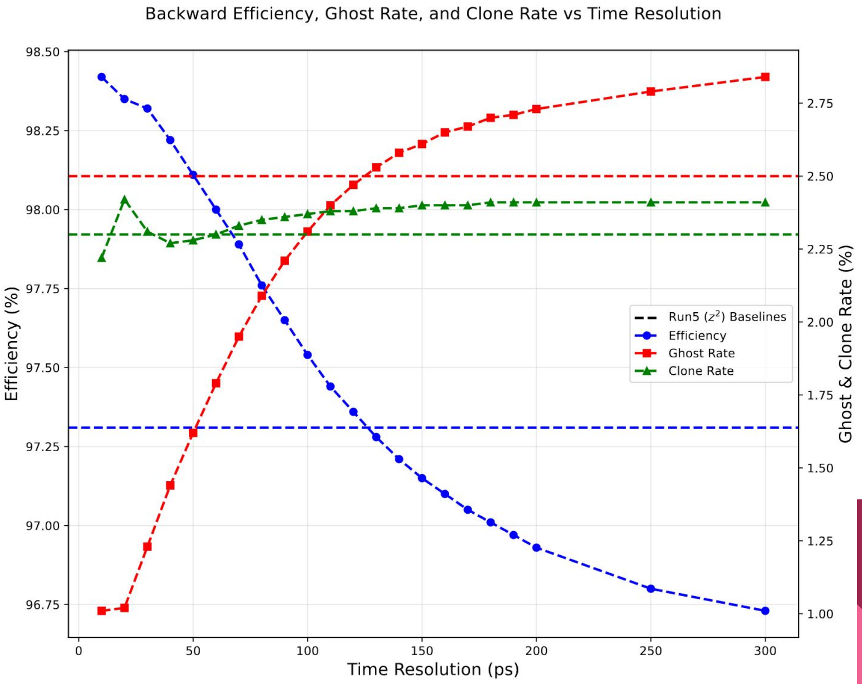


# Timescans: What about other resolutions?

Now for the “best” efficiency of the four windows



# Outlier timestamps

Problem: Clone rates too high

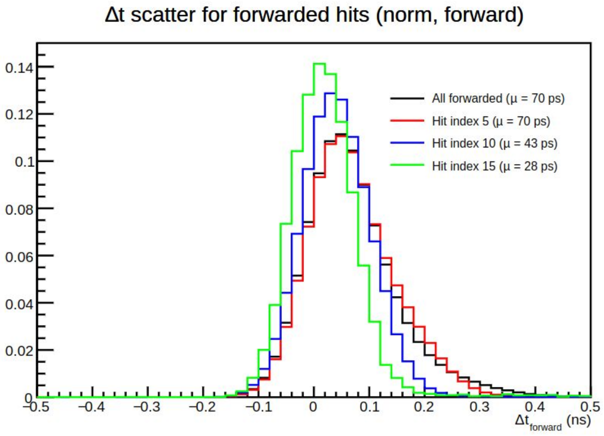
→ Potential solution:  Allow up to 1 (or more) hits in a track that are outside of nominal cut

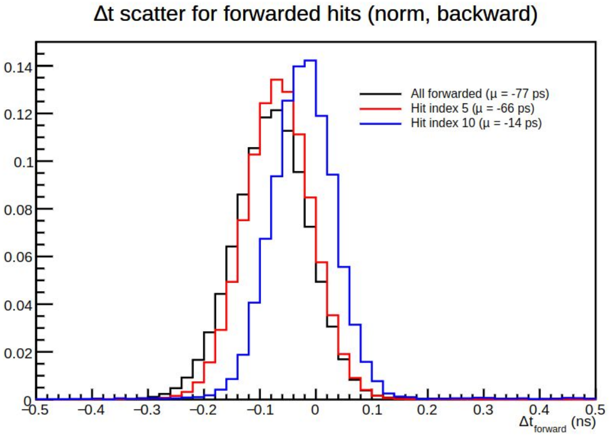
But before we move on, let's have brief look what the time scatter plots tell us…

Do we expect many hits to fall outside of our cut?

Skews – why?

# Outlier Timestamps – Forwarding



# Outlier Timestamps – Forwarding

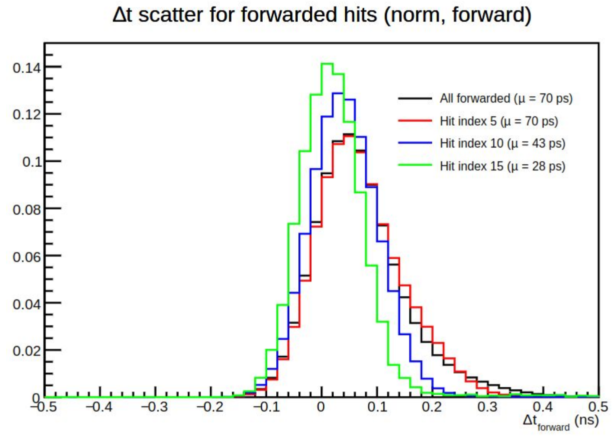
Skews – why?

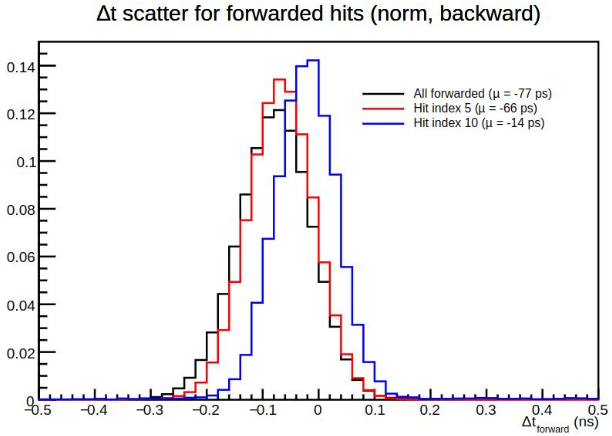
Speed assumed to be exactly c!

$$t_{next}^{pred} = t_{curr} + \Delta r / c < t_{next}^{meas}$$

Options:

- Skip outliers
- Widen forwarding window
- Scale prediction by skew



# Last but not least: Throughput

Ensure around 1min of runtime at least. Repeat 5x for each algorithm configuration to ensure no outlier.

- For run3:
  - 500 events
  - 10k repetitions on L40s, 2k on V100
- For run5:
  - 100 events
  - 2k repetitions on L40s, 400 on V100

# Last but not least: (preliminary) Throughput

All numbers given are in kHz of Events

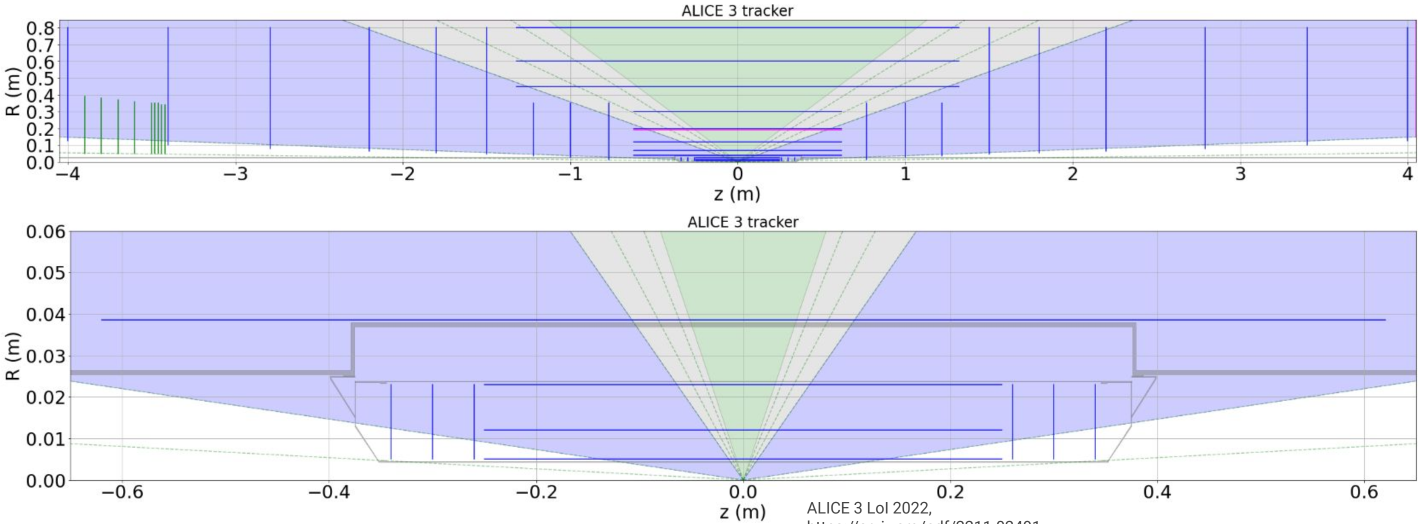
| | Run3 | Run5 3D | Run5 3D ($z^2$) | Run5 4D ($z^2$) |
|---|---|---|---|---|
| L40s | 1560 | 98 | 77 | 70 |
| V100 | 600 | 39 | 25 | 21 |

Unreasonable drop with $z^2$ dependence!

# Going forward from forward: Central rapidities in ACTS

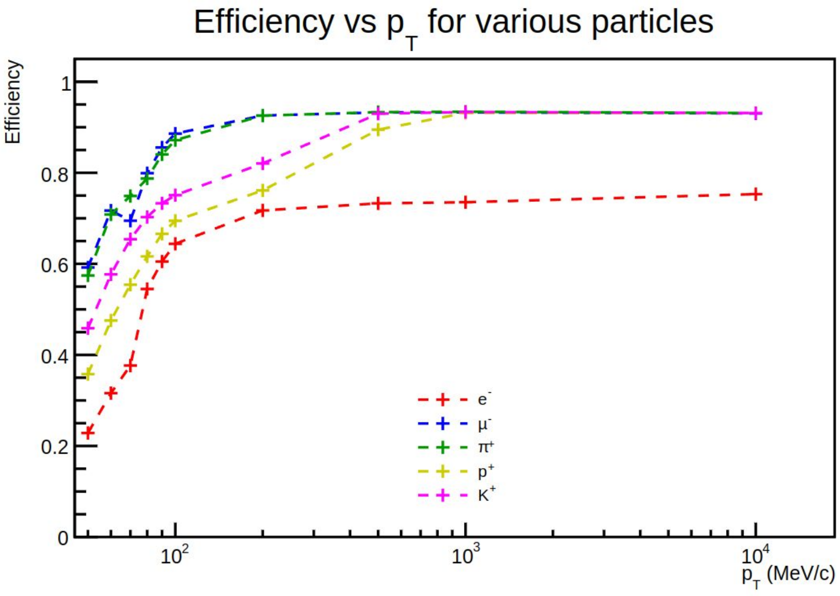
- LHCb: Forward detector
- ALICE 3: Barrel & endcap detector
- ACTS: General tracking framework
  - Shift focus to more central region

ALICE 3 geometry implementation in ACTS ongoing!

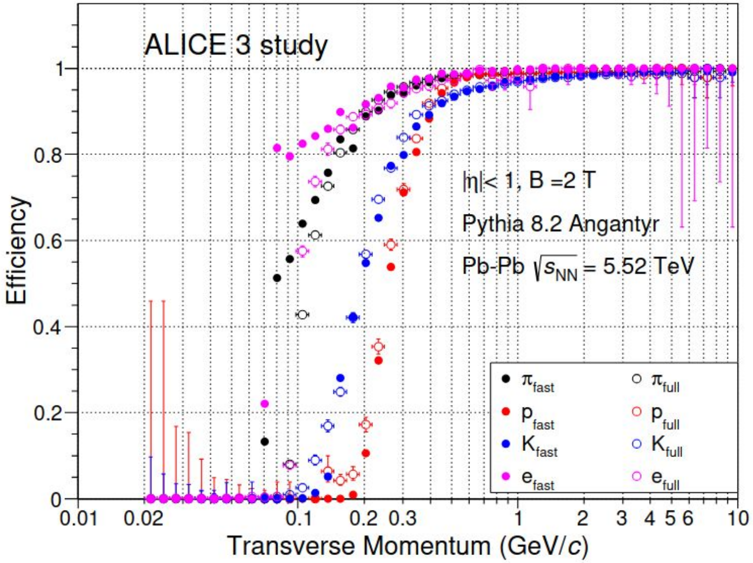



ALICE 3 LoI 2022, https://arxiv.org/pdf/2211.02491

# ACTS ALICE 3 Efficiency status (spoiler: terrible)




LoI reference




Material interactions?
Seeding? Geometry?

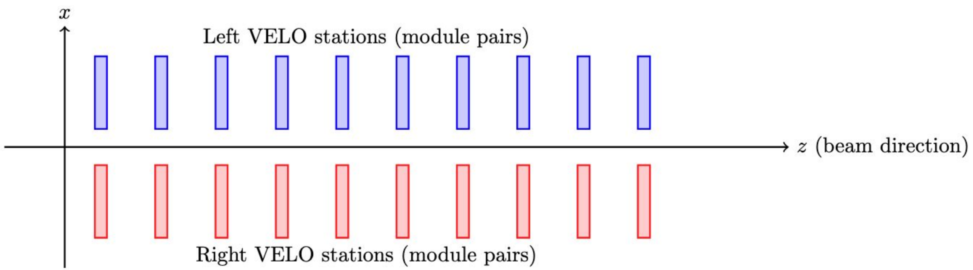
# Conclusions & Going forward

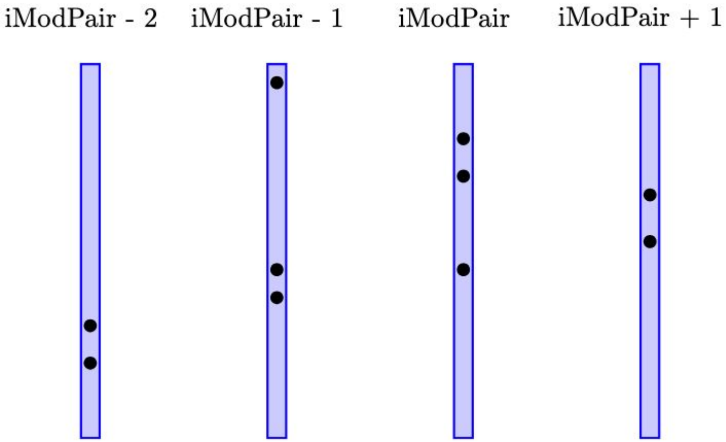
- LHCb results:
  - Introducing simple 1D filtering and cutting on time
    - Improves Efficiency slightly
    - Reduces ghost rate by 40%
    - Increases clone rate by 20%
  - Throughput remains reasonable with increased occupancy
  - Clone rate expected to drop with introduced clustering
  - Minor tweaks possible (outliers, skew correction)
  - Establish systematic error bands
- Moving forward: ACTS
  - Establish baselines and recreate LoI results
  - Move to very low $p_T$ spectrum

# BACKUP SLIDES

# Tracking in Allen: Search By Triplet

- Example case:



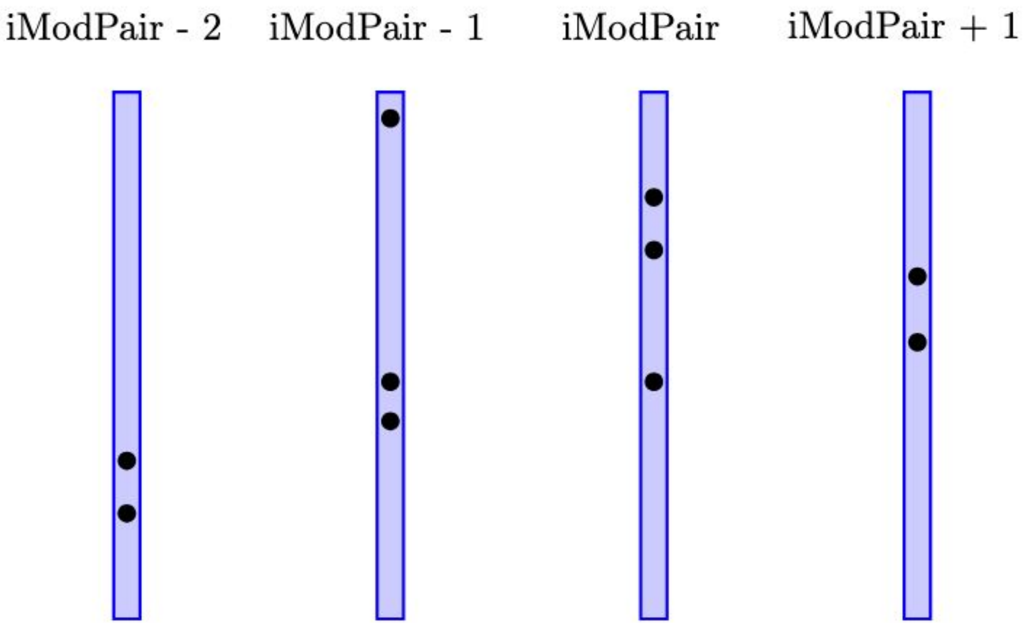
# Error propagation...

$$\sigma_\mu = \frac{\sigma}{\sqrt{N}}$$

Forwarding: Standard error on the mean:

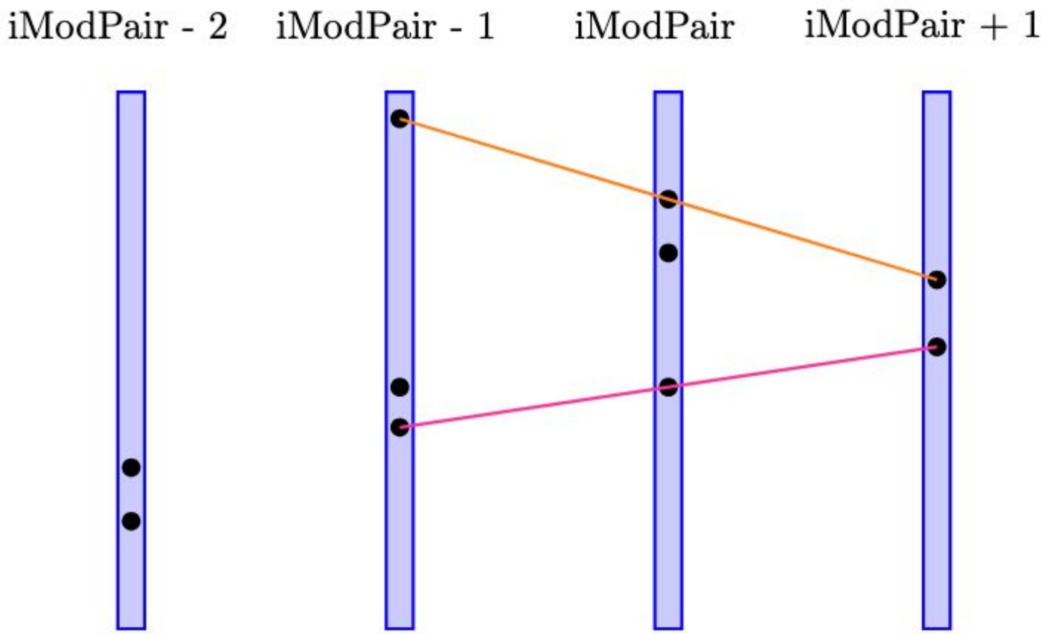
$$\sigma_{\mu,next} = \sqrt{\sigma^2_{\mu,curr} + \sigma^2_{meas}} = \sqrt{\sigma^2/N_{Hits} + \sigma^2} = \sigma\sqrt{1/N_{Hits} + 1}$$

# Tracking in Allen: Search By Triplet



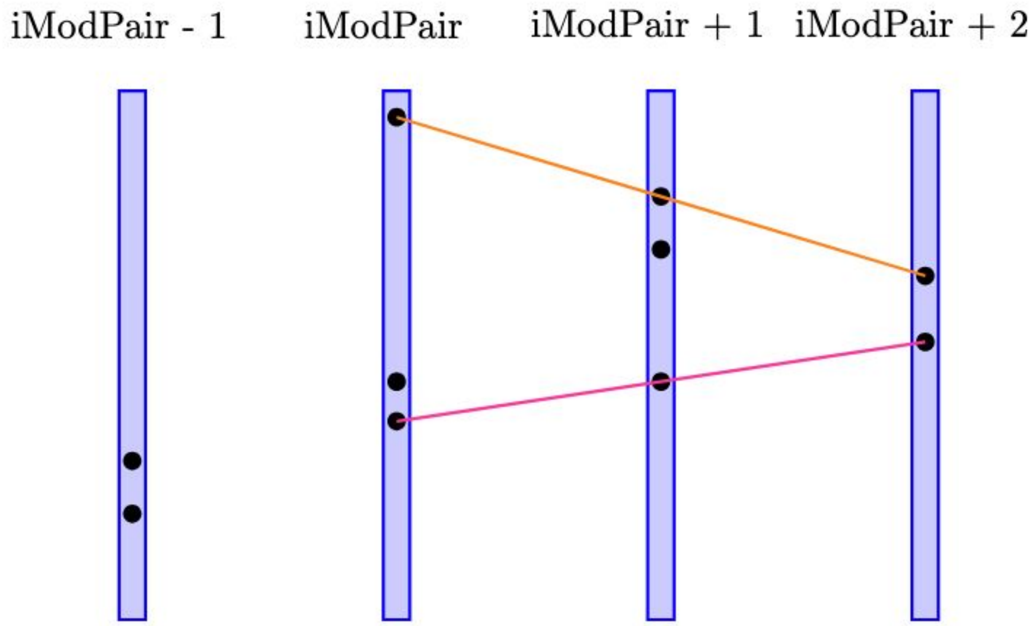

- First: Create set of triplets starting in last layer.

# Tracking in Allen: Search By Triplet



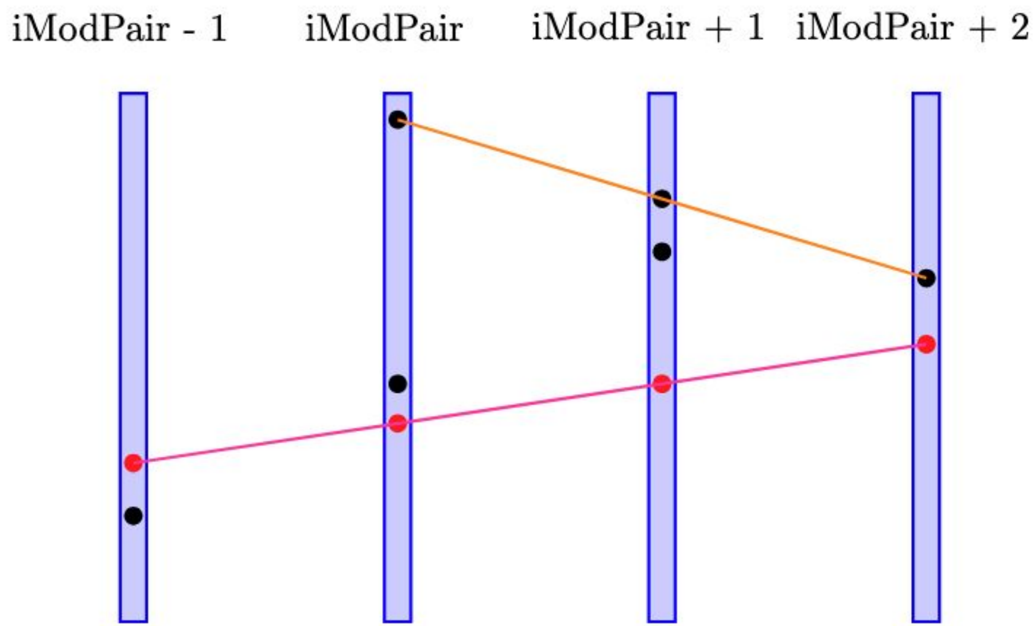

- First: Create set of triplets starting in last layer.Then repeat following:
    a. Move to next module pair

# Tracking in Allen: Search By Triplet



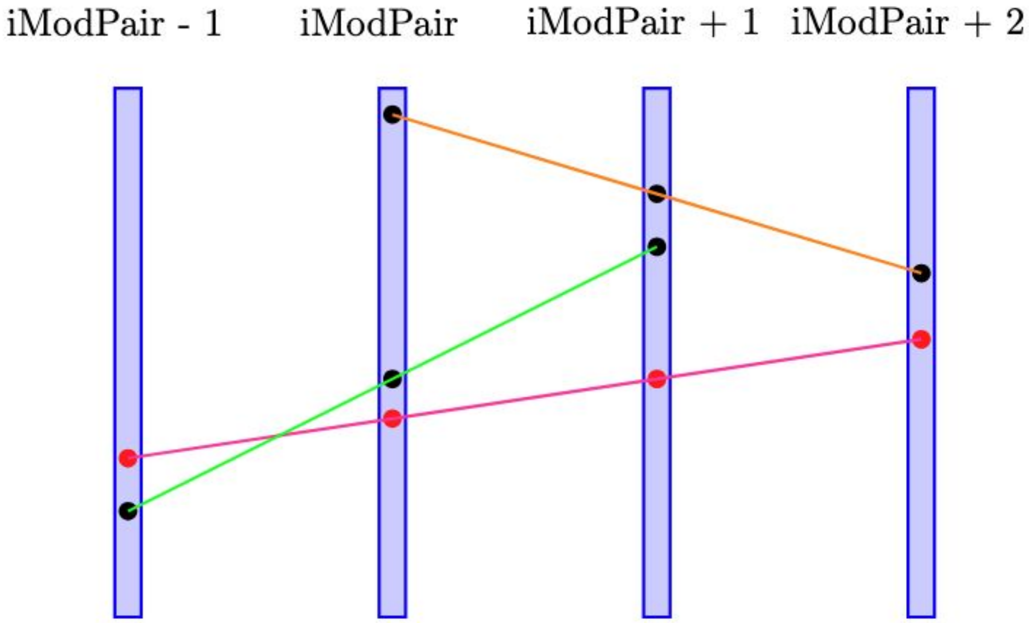

- First: Create set of triplets starting in last layer. Then repeat following:
  a. Move to next module pair
  b. Forward existing tracks and tag hits as used

# Tracking in Allen: Search By Triplet




- First: Create set of triplets starting in last layer. Then repeat following:
  a. Move to next module pair
  b. Forward existing tracks and tag hits as used
  c. Create seeds with remaining hits in layer

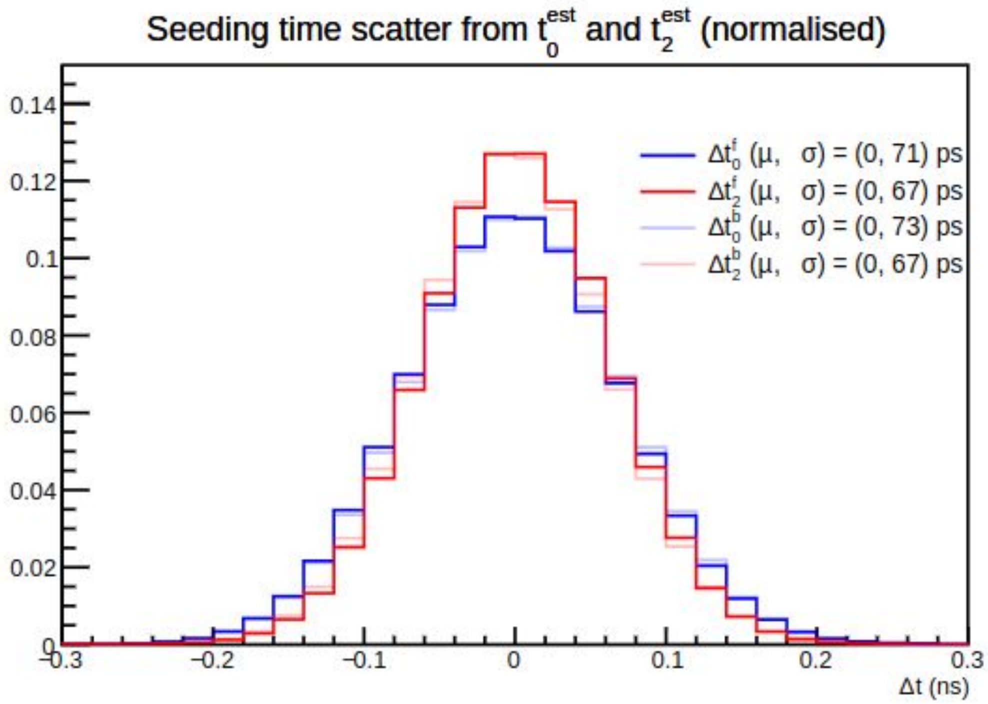
# Tracking in Allen: Search By Triplet




- First: Create set of triplets starting in last layer. Then repeat following:
  a. Move to next module pair
  b. Forward existing tracks and tag hits as used
  c. Create seeds with remaining hits in layer
- And so on…

# Outlier timestamps – Seeding

Simple idea: Allow for up to 1 (or more) hits in a track that are outside of nominal cut

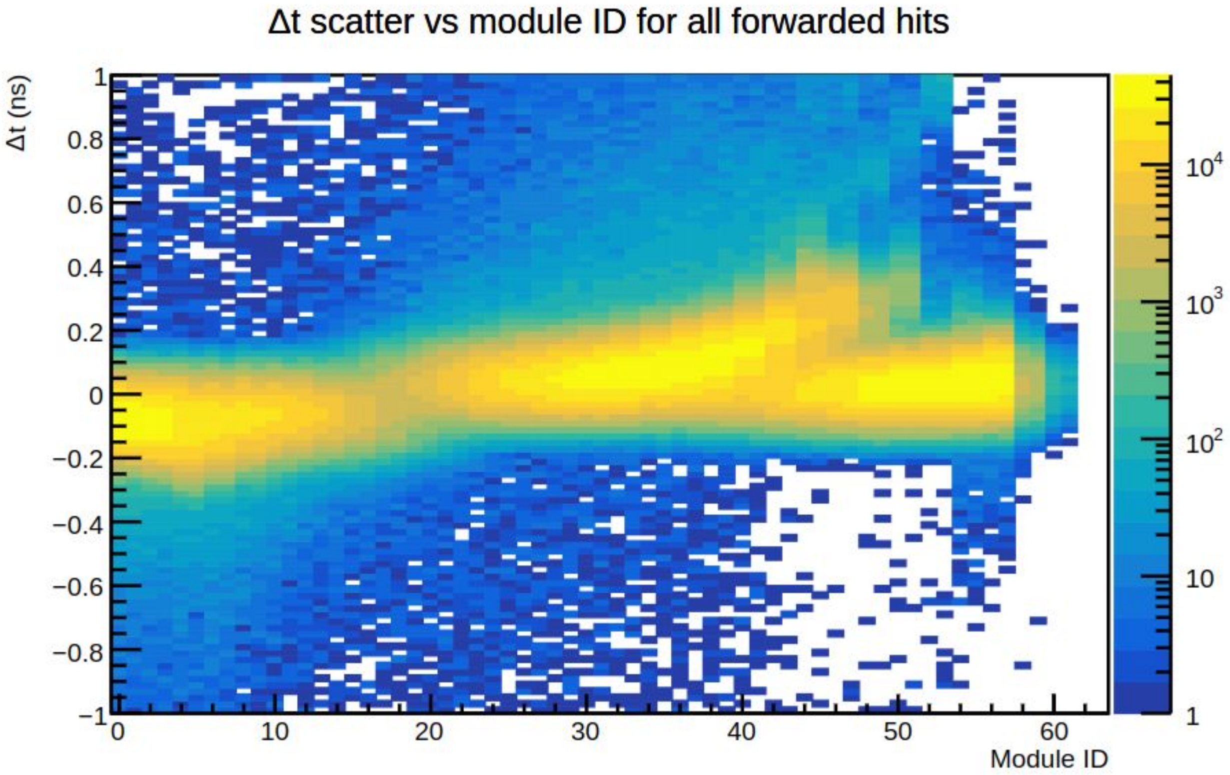



5000 events

No noticeable deviation in seeding

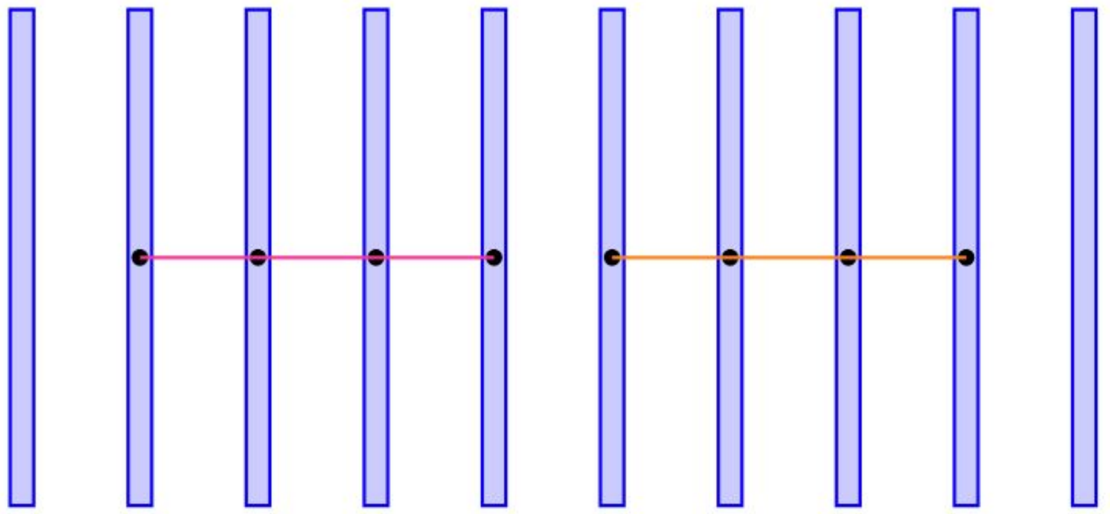
$3\sqrt{2}\sigma$ is plenty

# Outlier timestamps: Forwarding 2D



# Clone rates – Why the increase at all? Speculation…

1. Split Track Clones: Leading Order (LO)

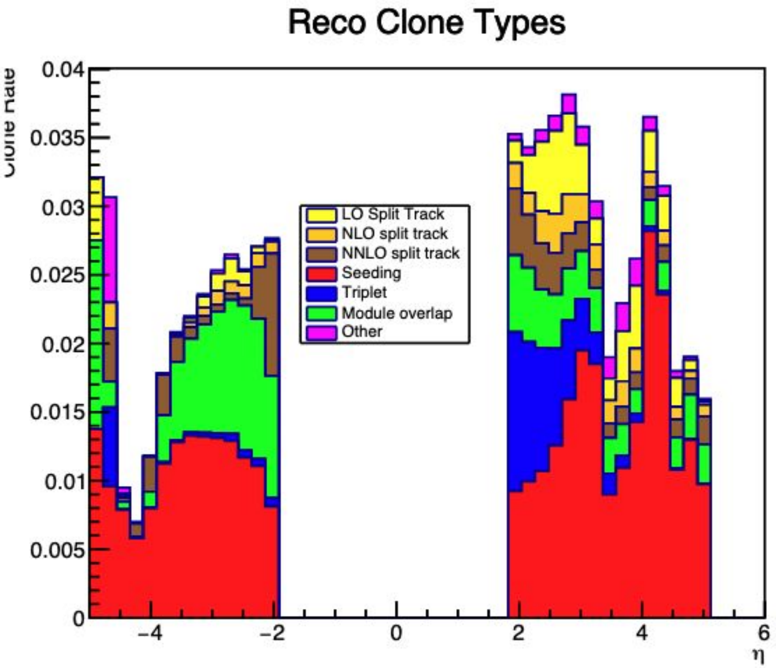


8 hit (MC) track

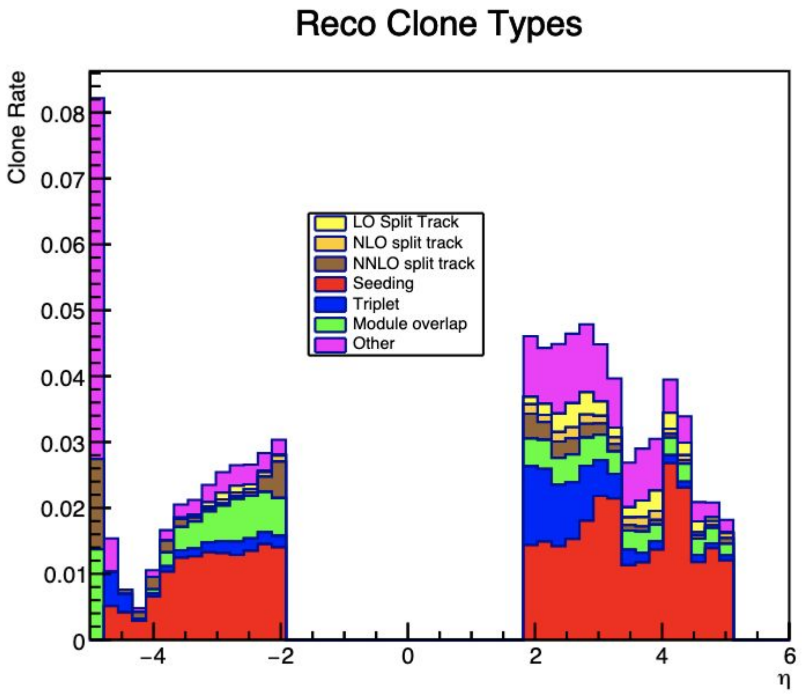
# Clone rates – Why the increase at all? Analysis

# Clone Diagnosis – Rates wrt η

Max Deflection = $\mathbf{k \times \Delta z^2}$, $\mathbf{k = 2\times10^{-5}}$

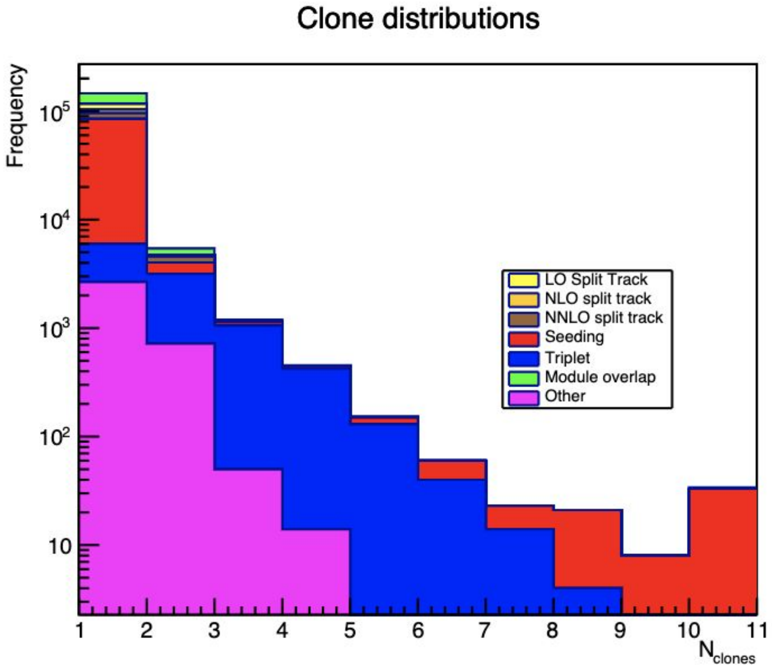



With time cuts included

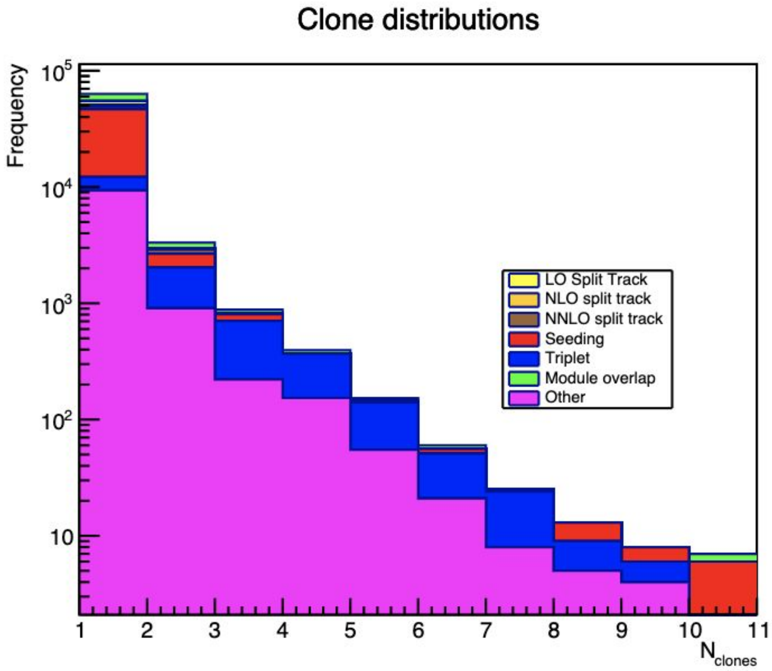


# Clone Diagnosis – Number of clones per MC particle

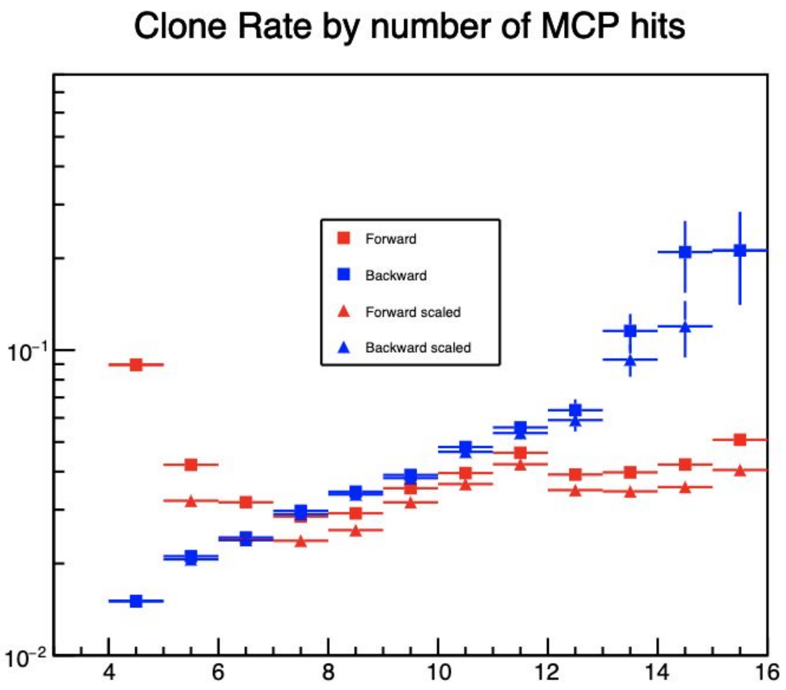
Max Deflection = $\mathbf{k \times \Delta z^2}$, $\mathbf{k = 2\times10^{-5}}$




With time cuts included

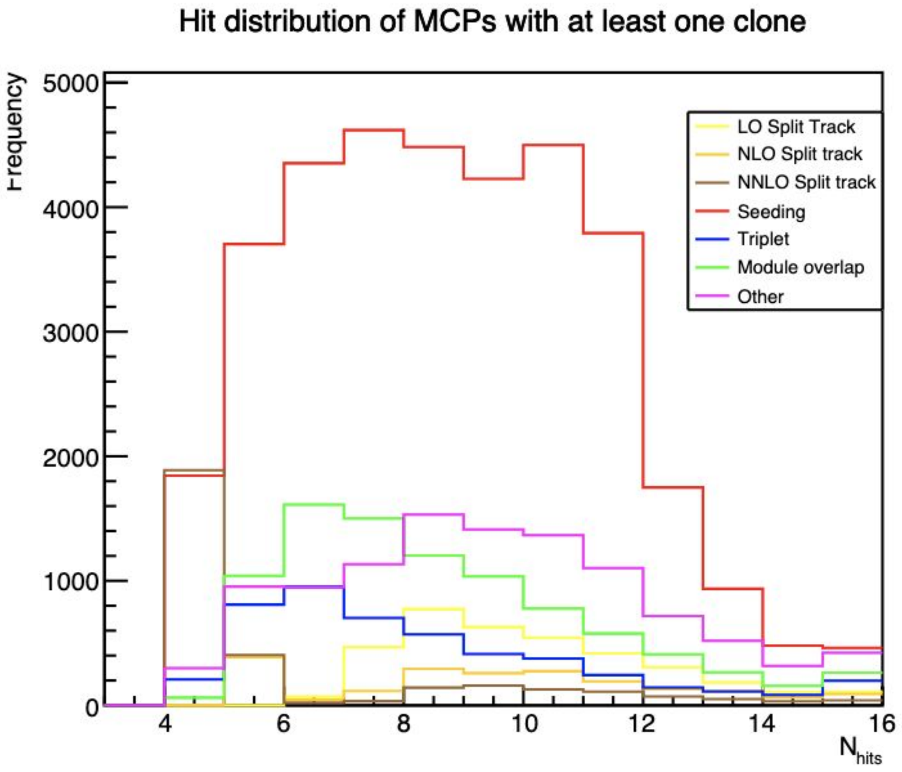


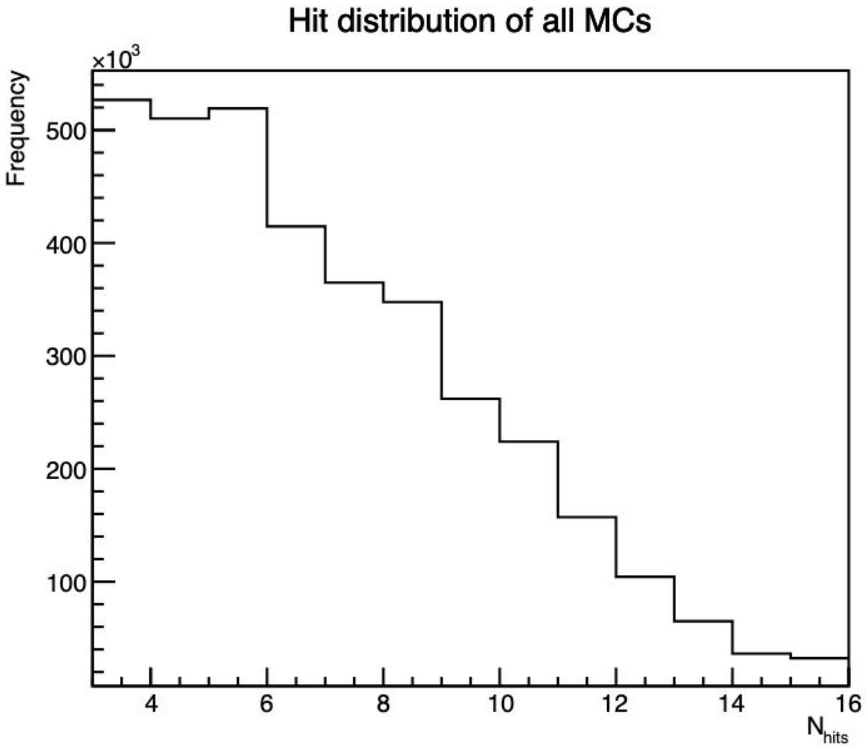
# Clone Diagnosis – What are we dealing with here?




- Increase dominated by "Other" clone type
- By OOM, dominated by single clones
- No clear MC track length correlation
  - Slight overrepresentation in 8-10 hit region

# Clone Diagnosis – Number of hits of MCP



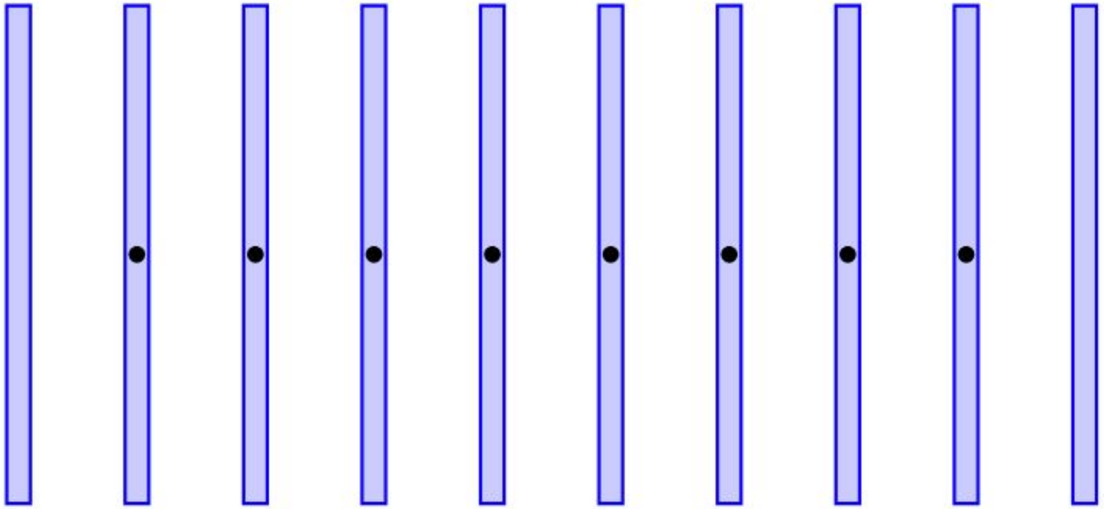



# Different types of clones

1. Split Track Clones
2. Seeding Clones
3. Module Overlap Clones
4. Triplet Clones
5. Other Clones

# Different types of clones

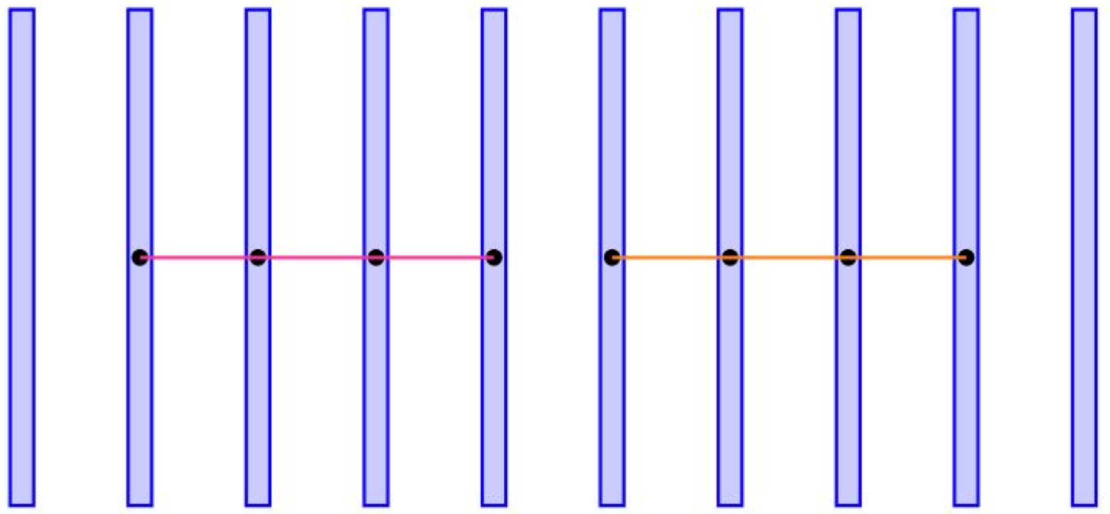
1. Split Track Clones



8 hit (MC) track

# Different types of clones

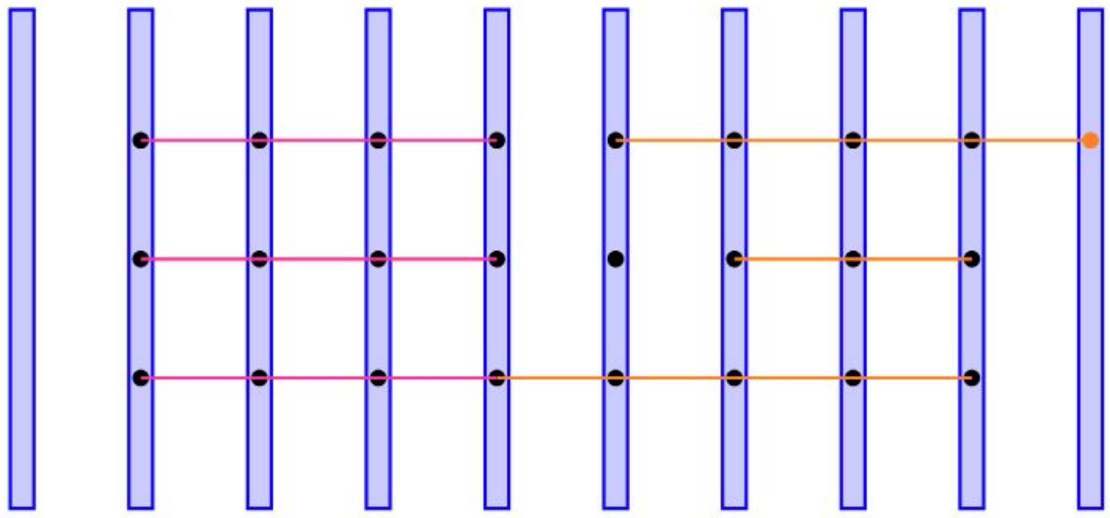
1. Split Track Clones: Leading Order (LO)



8 hit (MC) track

# Different types of clones

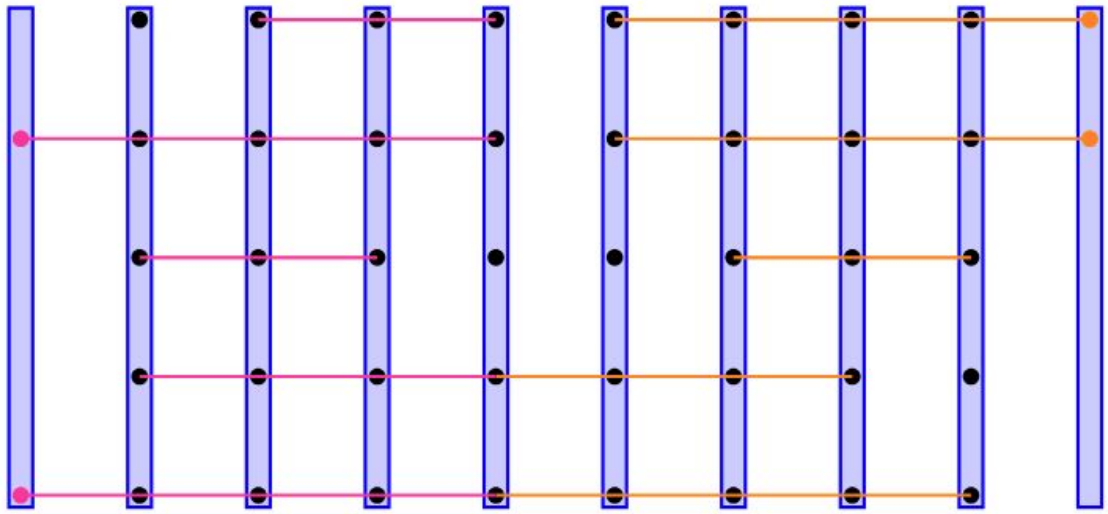
1. Split Track Clones: Next to Leading Order (NLO)
   a. One extra, missing, or overlap



8 hit (MC) track

# Different types of clones

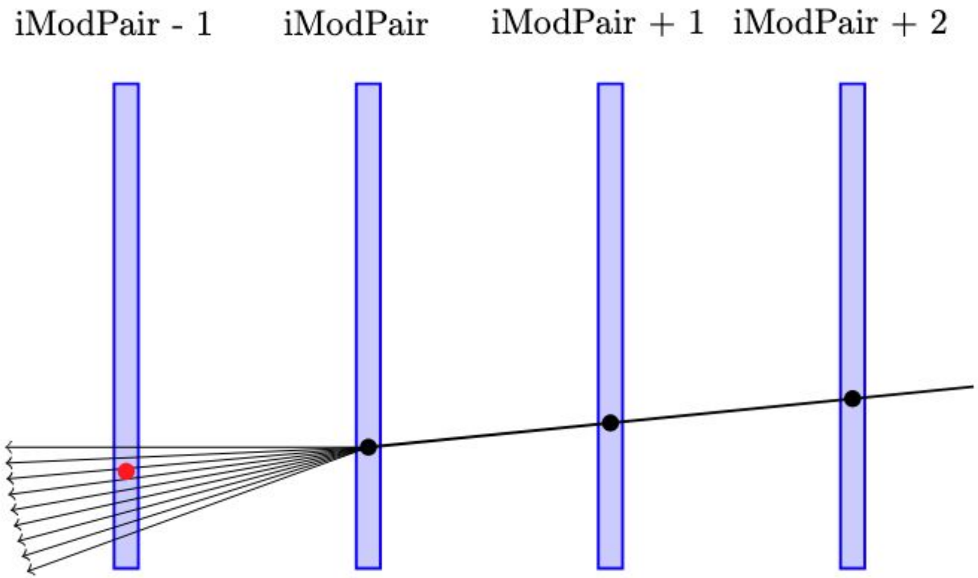
1. Split Track Clones: Next to Next to Leading Order (NNLO)
    a. Two extra, missing, or overlap (any combination possible)



8 hit (MC) track

# Different types of clones

1. Split Track Clones
2. Seeding Clones

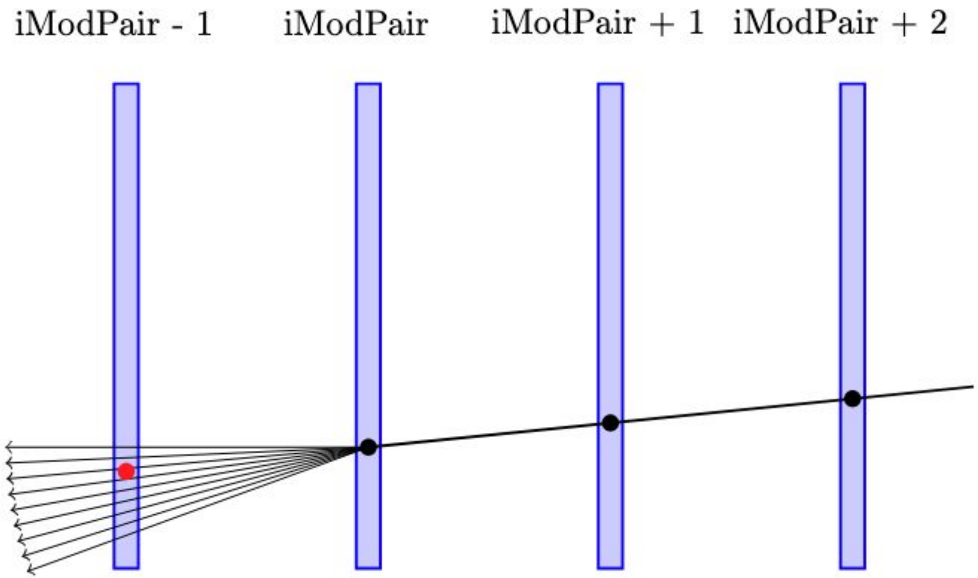


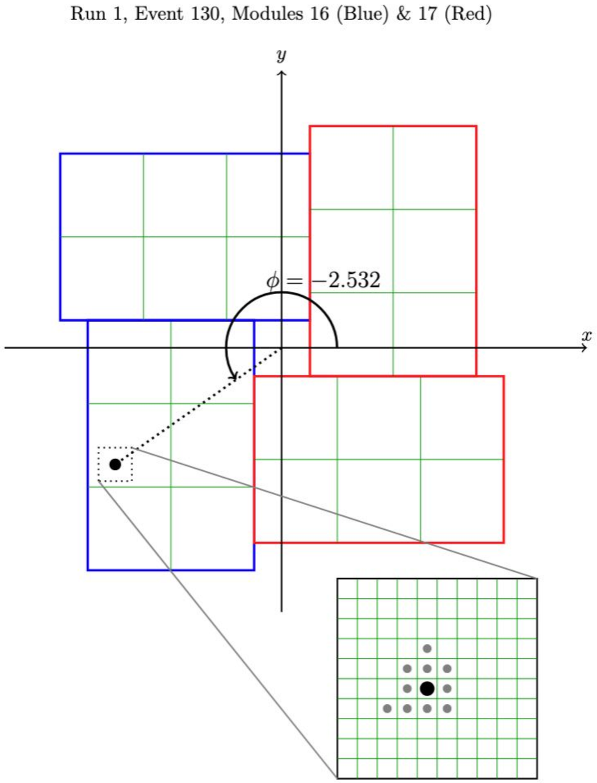

Hits at certain $\phi$:

- iModPair - 1: **1**
- iModPair: **MANY**
- iModPair + 1: **1**

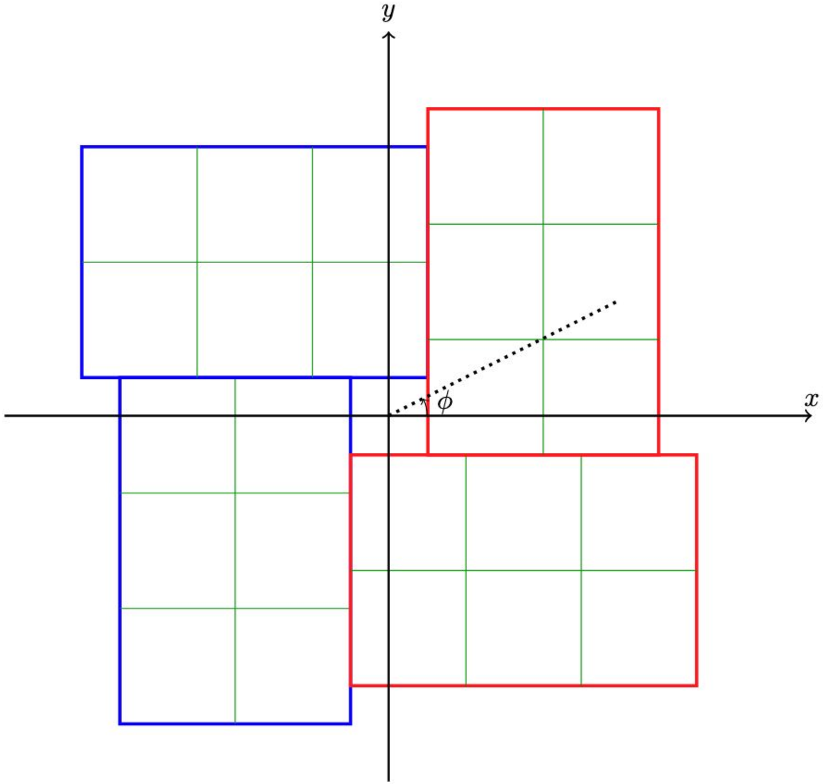
# Different types of clones

1. Split Track Clones
2. Seeding Clones



# Tracking in Allen: Search By Triplet (SBT)

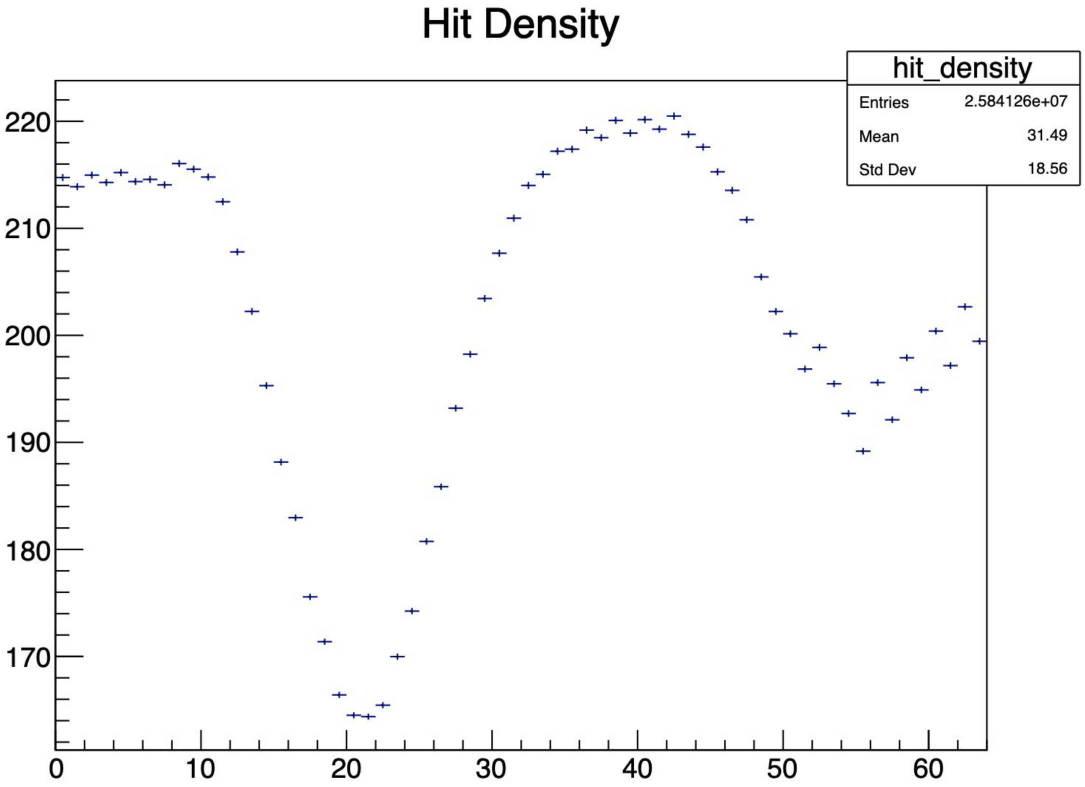



- Input: Clusters
  - x, y, z, t, ID
- Clusters sorted by $\phi$ per module

# Tracking in Allen: Search By Triplet

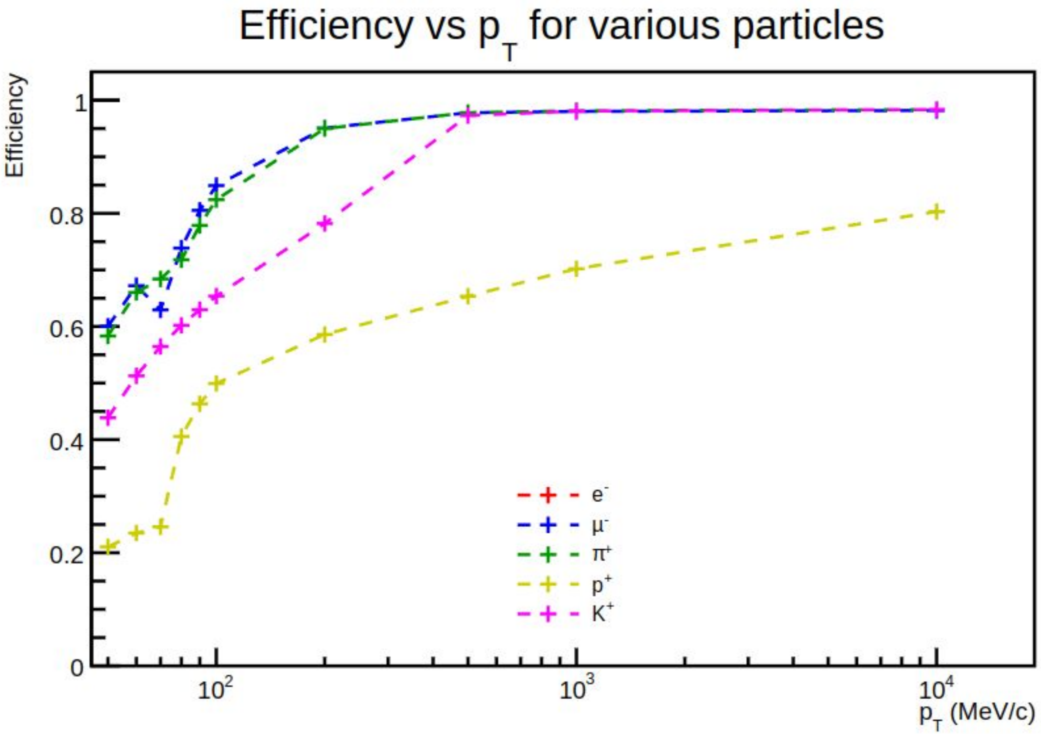
- Sweep through detector from back
- First: Create set of triplets starting in last layer. Then repeat following:
    a. Move to next module pair
    b. Forward existing tracks and tag hits as used
    c. Create seeds with remaining hits in layer

# Hit density wrt modules




No clear forward/backward preference

# ACTS Truth Seeding efficiency



# ACTS D0 resolution