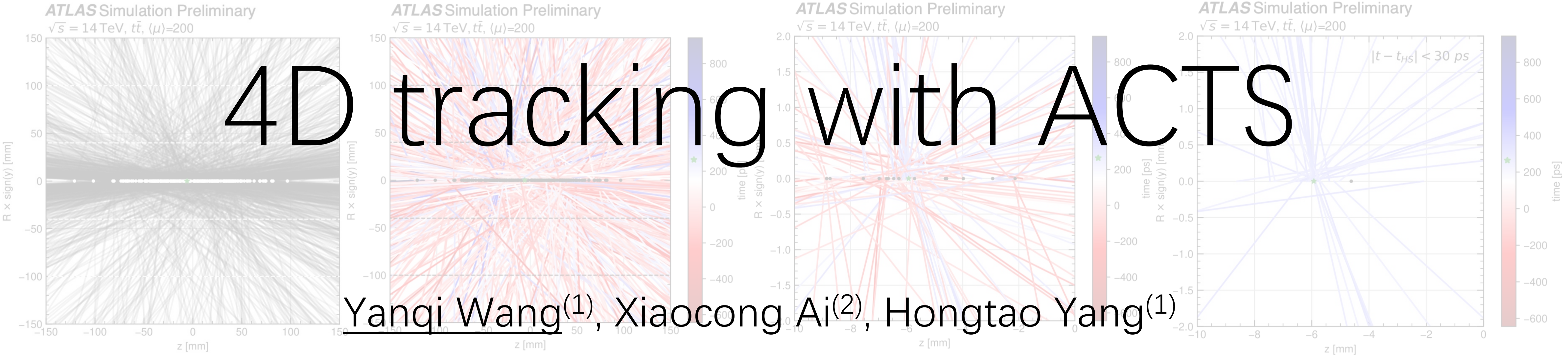
4D tracking with ACTS



Yanqi Wang[(1)], Xiaocong Ai[(2)], Hongtao Yang[(1)]

(1) University of Science and Technology of China, (2) Zhengzhou University

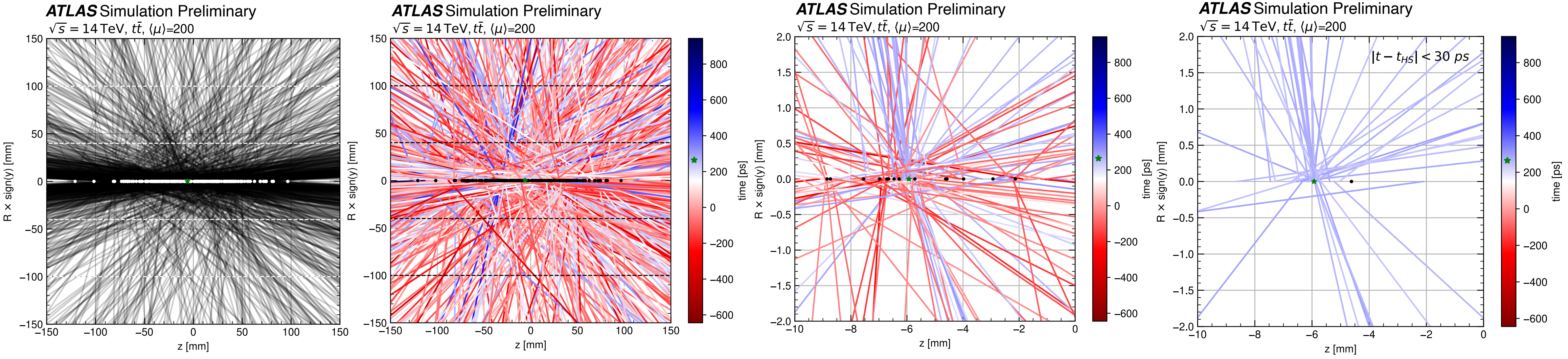
tracking workshop 2025 @ 惠州 on July 22th, 2025

Pictures from: ATL-PHYS-PUB-2023-023: Investigating the impact of 4D Tracking in ATLAS Beyond Run 4

Introduction
Chance of future 4D tracking for ITk

- Motivation – LHC upgraded to HL-LHC

  Luminosity up to $\mathbf{7.5\times10^{34}\ cm^{-2}s^{-1}}$, pile-ups up to **200**

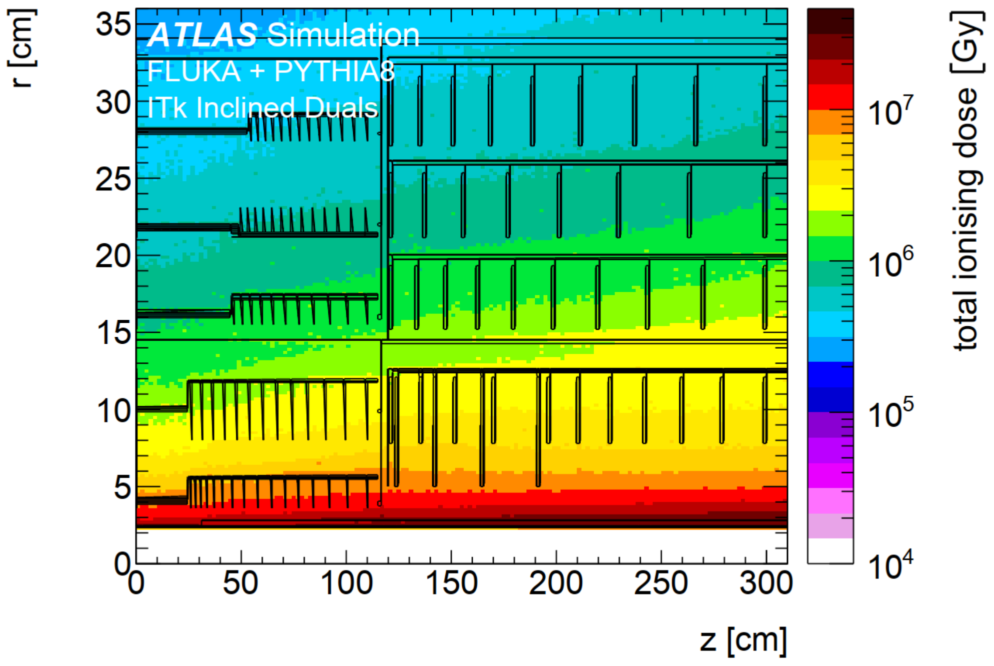


Longitudinal view of a simulated $t\bar{t}$ event with <μ>= 200 in ITk

No time information | Add time information | Zoom in space | Separation by time

Introduction
Chance of future 4D tracking for ITk

- The two innermost layers of the ATLAS ITk Pixel is designed for a maximum luminosity of **2000 fb^{-1}**.
- Chance to implement 4D tracking for the two innermost layers in the middle of HL-LHC operation:
  - ➢ Need to understand the physics potential
  - ➢ This study is focusing on the low-level 4D tracking performance, more about 4D vertexing and b-tagging in [Valentina's talk]



The fluence and dose distributions for the ITk Pixel Detector.

Total ionizing dose

[ATLAS-TDR-030 Inner Tracker Pixel Detector]

ACTS GitHub: https://github.com/acts-project/acts
ACTS docs : https://acts.readthedocs.io/en/latest/index.html

Introduction
Add time measurements for ITk in ACTS

- ACTS is an experiment-independent toolkit for (charged) particle track reconstruction in (high energy) physics experiments

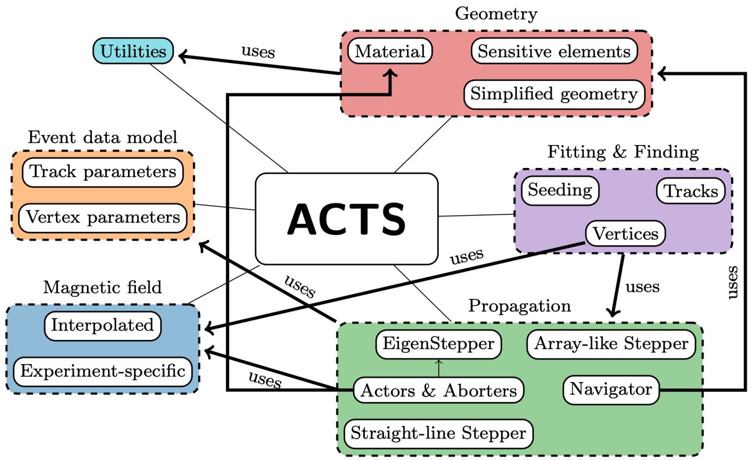
➢ track parameters in ACTS

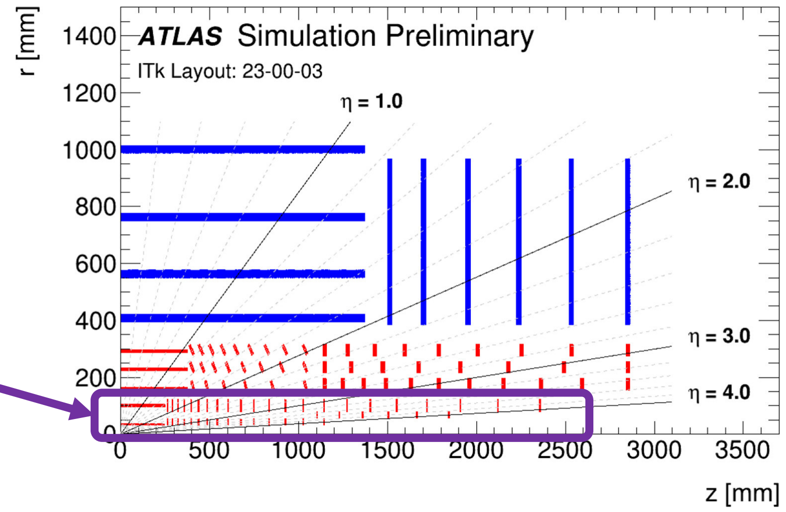
$$\vec{x} = \left(l_0, l_1, \phi, \theta, \frac{q}{p}, t\right)$$

- Smear truth hit information in the ITk innermost 2-layer pixel region with gauss distribution

$$\sigma(t) = 50\,ps$$

to provide time measurements





Outline

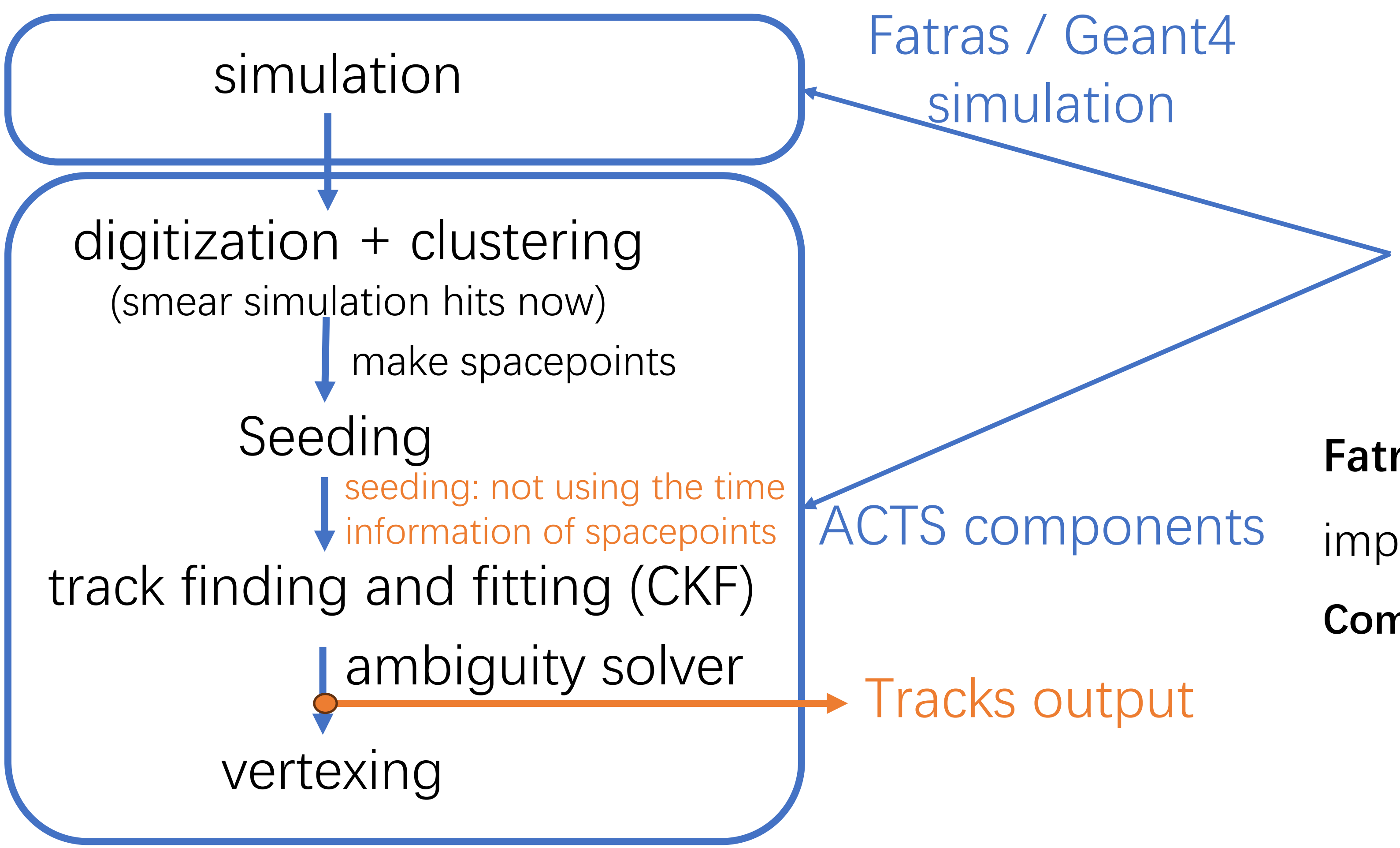
- ACTS-standalone ITk 4D tracking
- ITk 4D seeding with ACTS
- Timing pixel digitization in Athena

ACTS-standalone ITk 4D tracking

Fatras and Truth-seeded 4D track following performance

Geant4 and Reco-seeds seeded 4D track following performance

Workflow of ITk_full_chain in ACTS



Geometry and magnetic field

Fatras (**fa**st **tra**ck **s**imulation) is an ACTS implementation for fast track simulation.

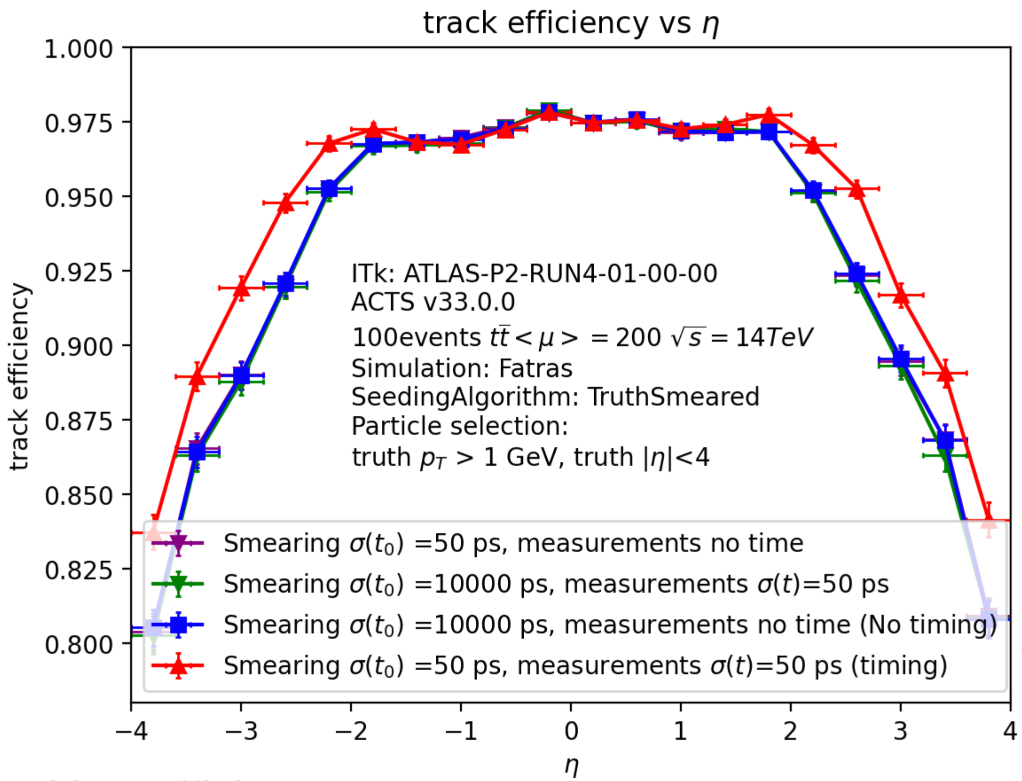
Combinatorial Kalman Filter

ACTS-standalone ITk 4D tracking

Fatras and Truth-seeded 4D track following performance

Geant4 and Reco-seeds seeded 4D track following performance

Tracking efficiency



- Add time for both **initial track parameters** and **measurements** of innermost 2-layer pixel
- Event generation: $\boldsymbol{t\bar{t}}$ **events** $\langle\mu\rangle$=200 with Pythia 8
- **Fatras** simulation & **Truth-Smeared** seeding
- No improvements on tracking efficiency if no timing info is provided for initial track parameters

RED (4D tracking): CKF initial track parameters and measurements have time information

BLUE (3D tracking): no time information for CKF initial track parameters and measurements

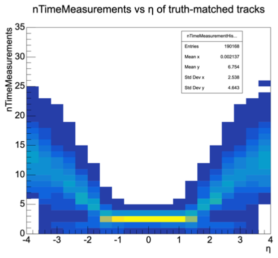
Tracking efficiency

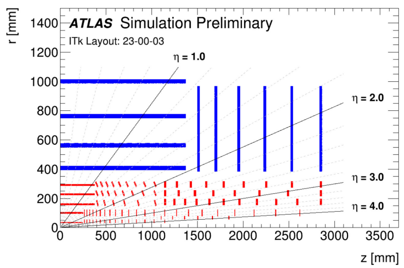
93.72% (no timing) to 94.87%↑ (50 ps timing)

93.71% (only "measured" time for initial track parameters)

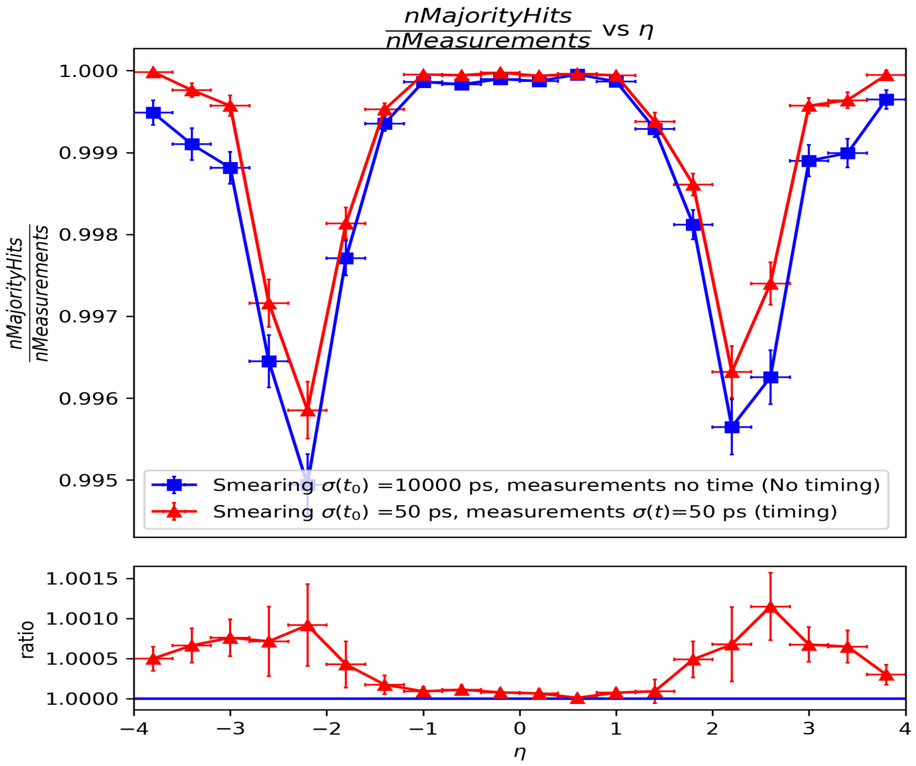
93.63%↓ (only time for measurements)

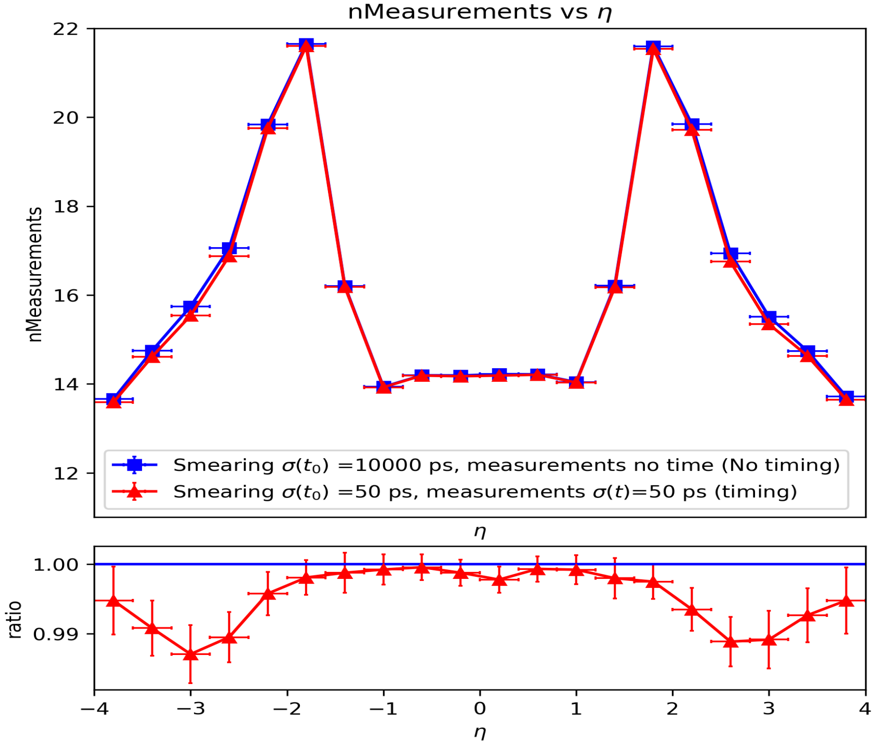
Match probability (a.k.a. purity)





More timing info in forward region

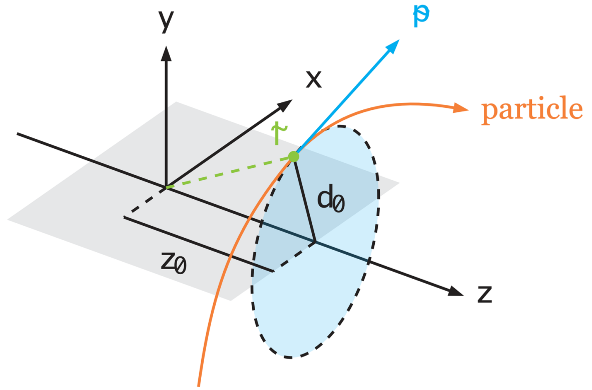




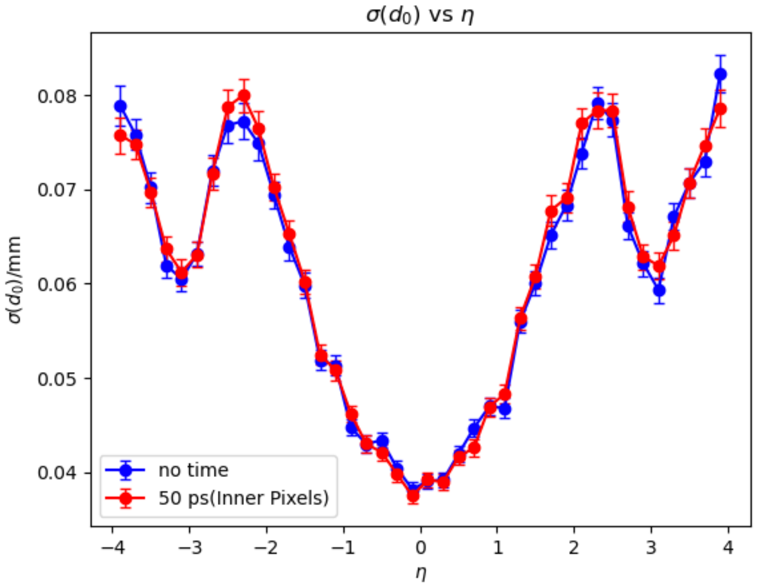
$$\frac{nMajorityHits}{nMeasurements} = \frac{\#\ hits\ that\ belongs\ to\ the\ particle\ with\ the\ highest\ fraction\ of\ hits\ on\ the\ track}{\#\ total\ measuremts\ on\ a\ track}$$

Increase on match probability (LEFT) and decrease on number of total measurements (RIGHT)

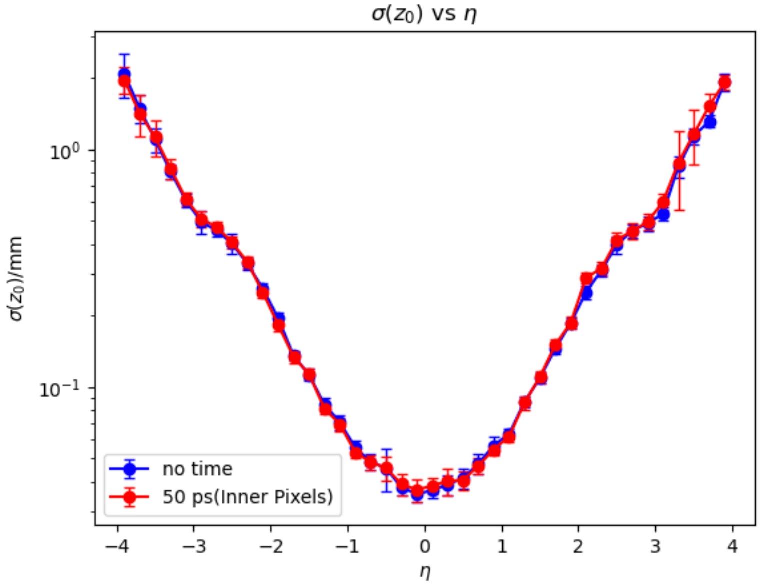
d_0 / z_0 resolution



d_0 resolution



z_0 resolution



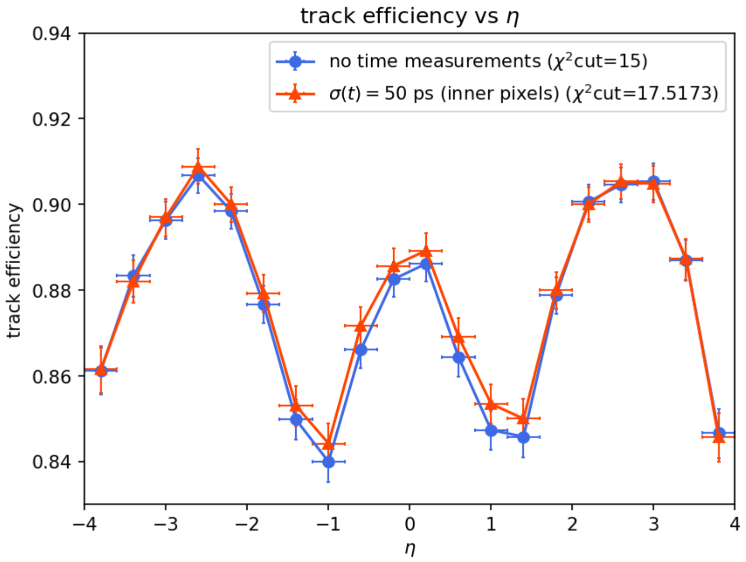
No significant change on the resolution of impact parameters d_0/z_0

ACTS-standalone ITk 4D tracking

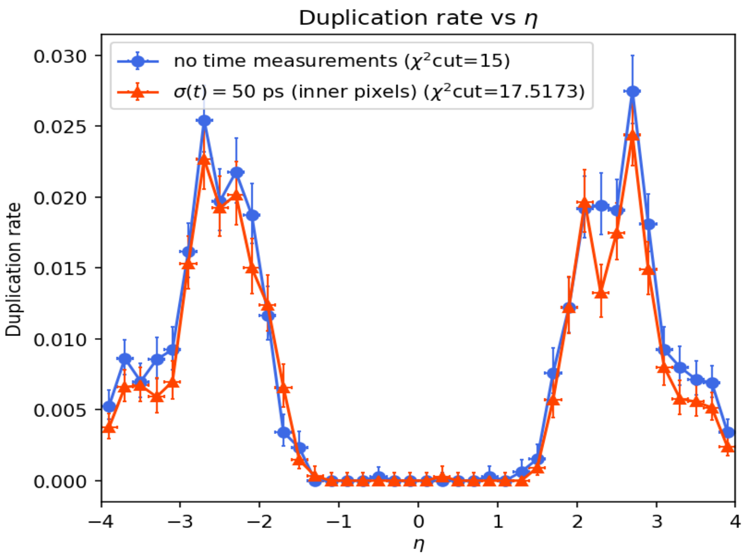
Fatras and Truth-seeded 4D track following performance

Geant4 and Reco-seeds seeded 4D track following performance

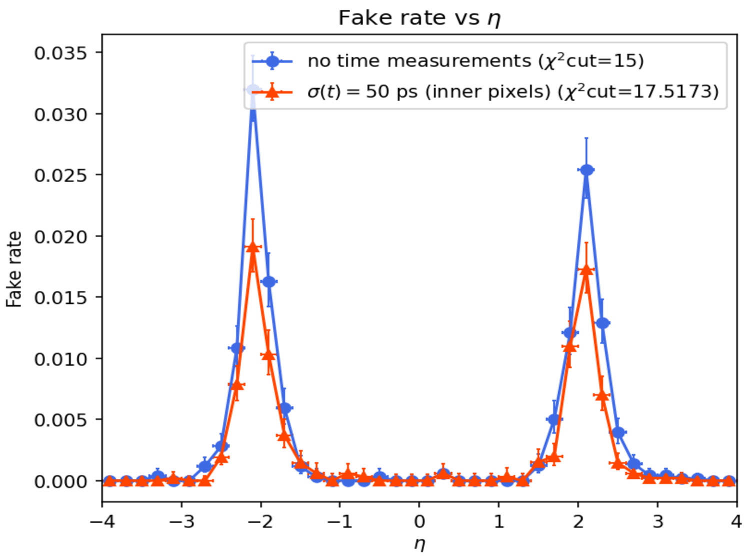
Tracking efficiency and quality



track efficiency



duplication rate



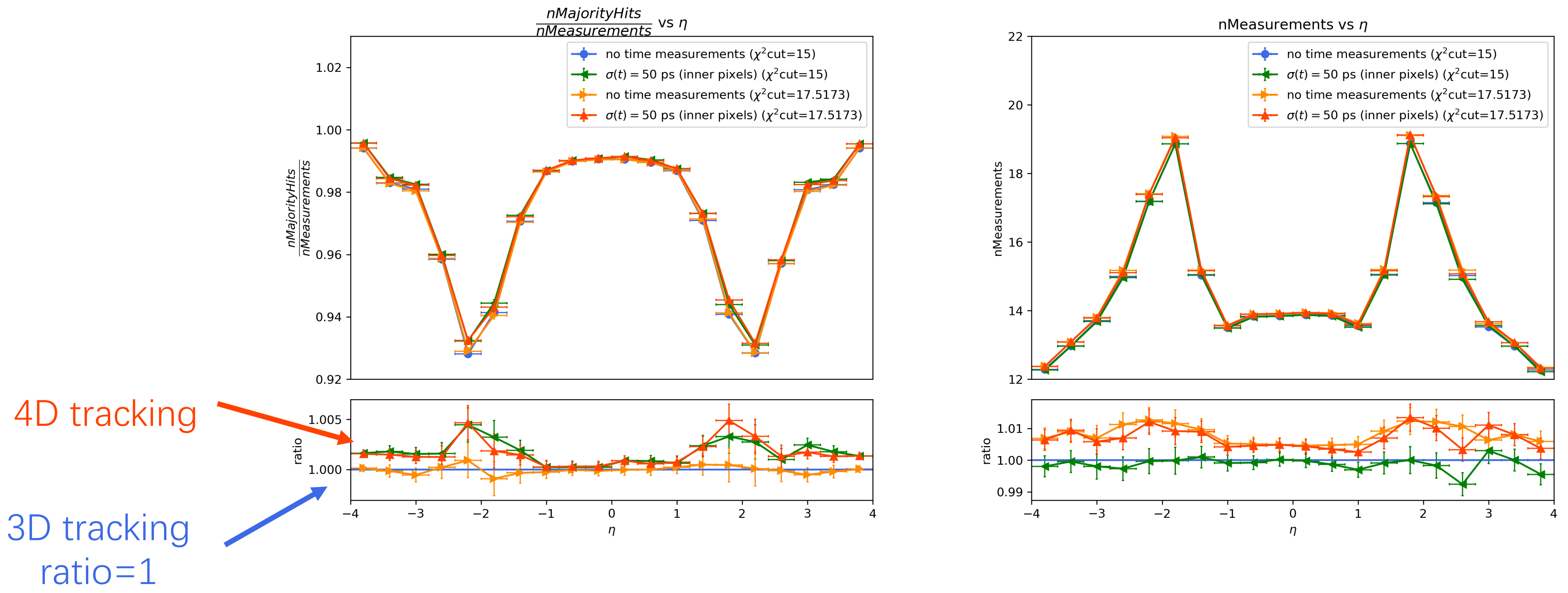
fake rate

track efficiency 87.58%->87.80% - ↑

duplication rate 0.89%->0.77% - ↓

fake rate 0.35%->0.22% ↓

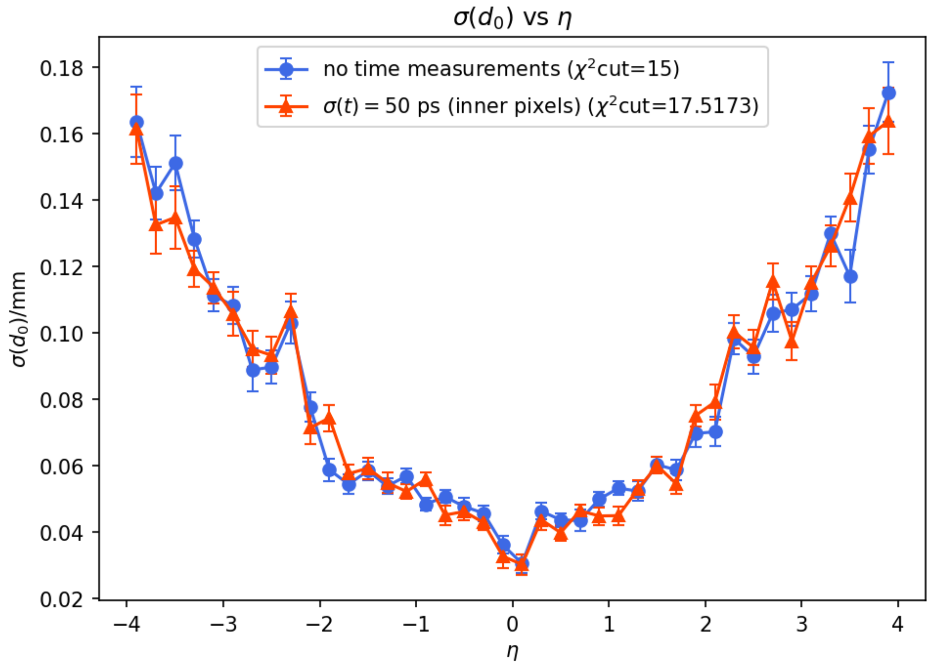
Match probability



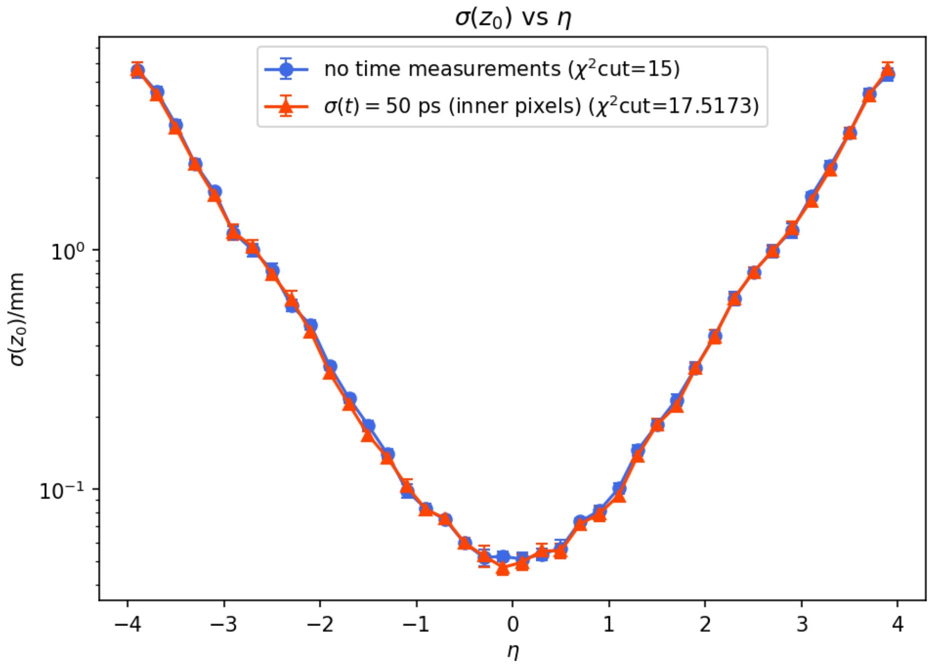
better matching probability and more measurements compared with 3D tracking

d_0 / z_0 resolution

d_0 resolution



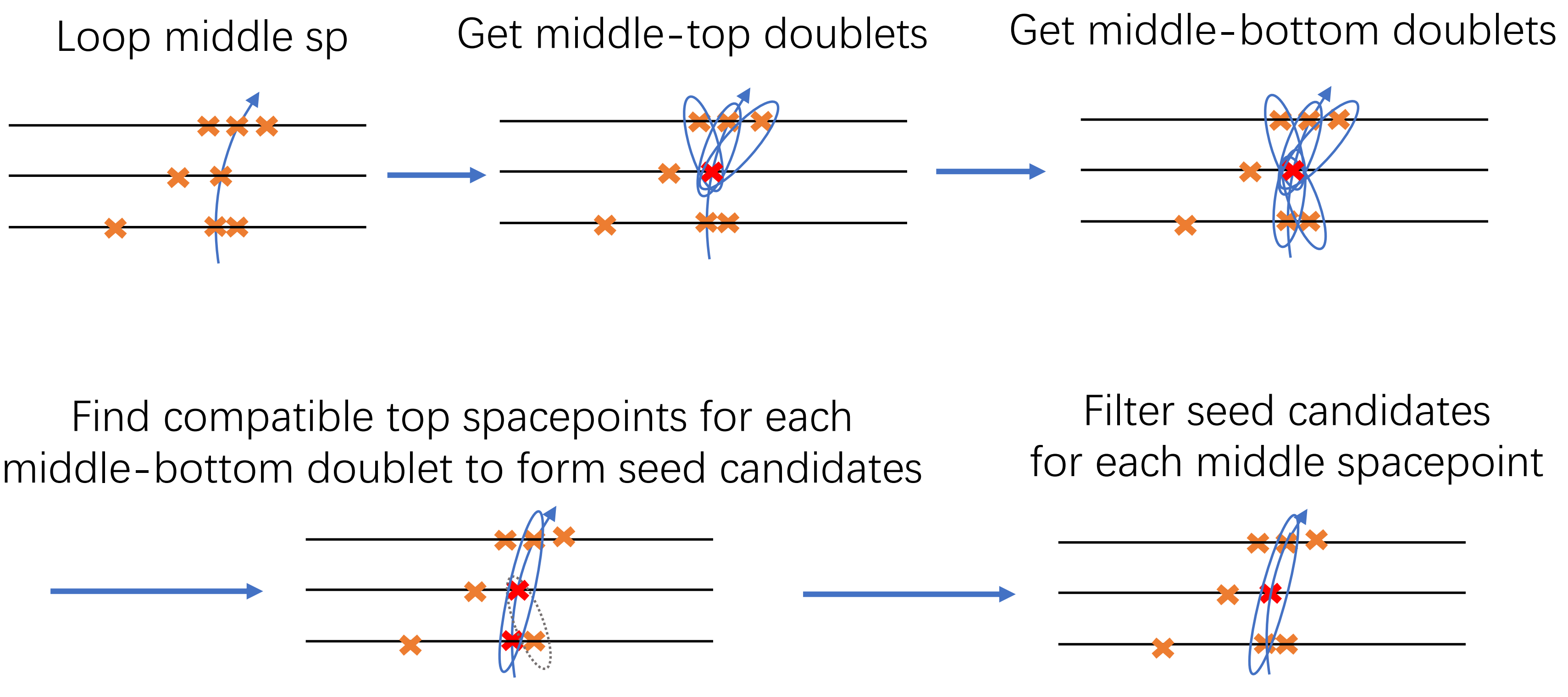
z_0 resolution



No significant change on the resolution of impact parameters d_0/z_0

ITk 4D seeding with ACTS

Default seeding algorithm in ACTS



Seed Finding:

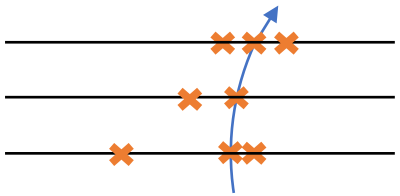
- Doublet check: radius, Δz, z_0, compatibility with IP (conformal transformation x-$y \rightarrow u$-v)
- Triplet check: θ (consider multiple scattering effect with minimum $\mathrm{p_T}$ & estimated $\mathrm{p_T}$) , d_0
- Assign a weight $w = (c_1 \cdot N_t - c_2 \cdot d_0 - c_3 |z_0|)$ $+detector\ specific\ cuts$

Find more details about seeding at
https://acts.readthedocs.io/en/latest/core/reconstruction/pattern_recognition/seeding.html#seeding-core

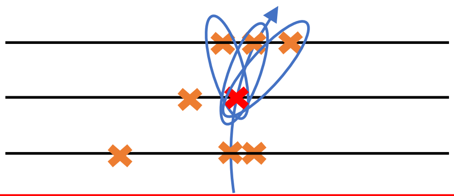
A toy 4D Seeding algorithm

Have added time compatibility check for doublets

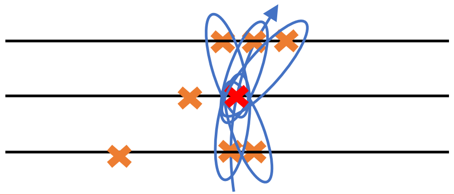
Loop middle sp



Get middle-top doublets

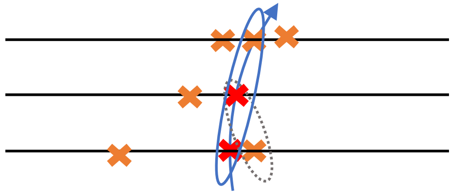


Get middle-bottom doublets

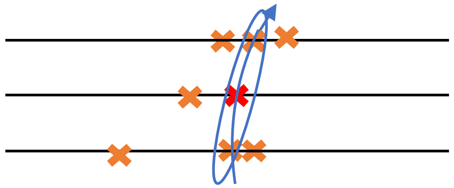


Adding time compatibility check for triplets is in progress

Find compatible top spacepoints for each middle-bottom doublet to form seed candidates



Filter seed candidates for each middle spacepoint



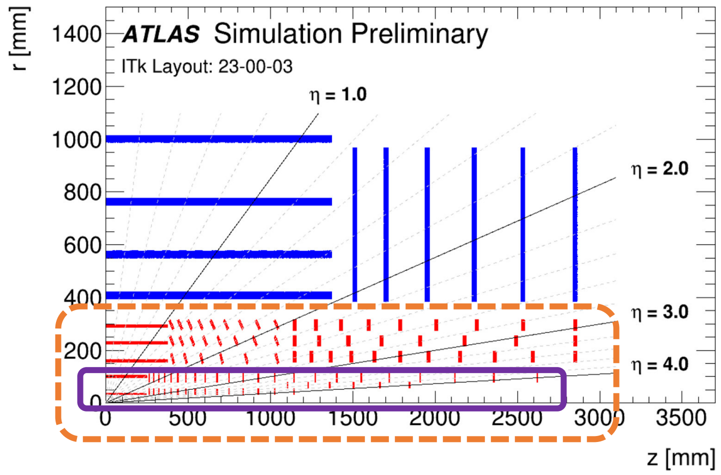
Seed Finding:

- Doublet check: radius, Δz, z_0, compatibility with IP (conformal transformation x-$y \rightarrow u$-v)
- Triplet check: θ (consider multiple scattering effect with minimum $\mathrm{p_T}$ & estimated $\mathrm{p_T}$) , d_0
- Assign a weight

$$w = (c_1 \cdot N_t - c_2 \cdot d_0 - c_3|z_0|)$$
$$+detector\ specific\ cuts$$

Find more details about seeding at
https://acts.readthedocs.io/en/latest/core/reconstruction/pattern_recognition/seeding.html#seeding-core

A toy 4D Seeding algorithm



- How many spacepoints have time?

	bottom sp	middle sp	top sp
muon particle gun	91.03%	73.94%	39.17%
$t\bar{t} < \mu >= 200$	91.51%	75.23%	46.03%

- Using **pixel (5 layers)** spacepoint seeding
- Adding time for innermost **2** layers

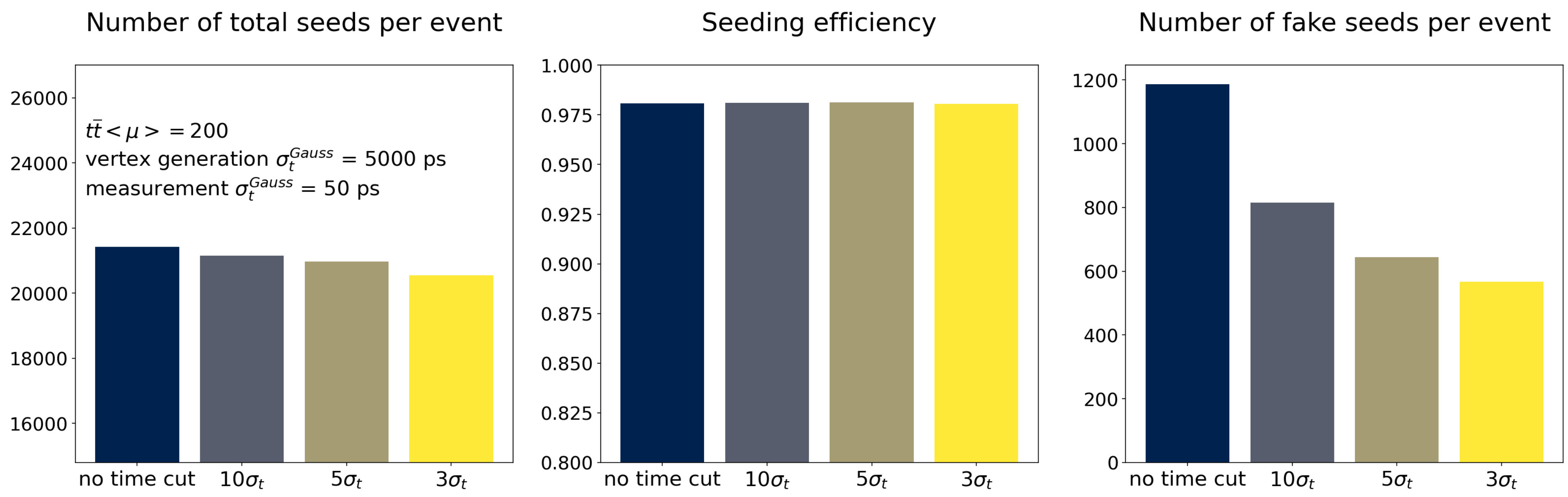
- Variable t_0 for spacepoint

$$t_0 = t_{arrival} - \frac{1}{c} \cdot \sqrt{r^2 + z^2} = t_{arrival} - t_{fly}$$

  - The time when the particle is generated
  - Add filter for doublets if **both** spacepoints have time info

$$|\Delta t_0| = \left| t_0^{middle} - t_0^{top|bottom} \right| > A \cdot \sigma_t$$

ITk $t\bar{t}$ with 200 pile-ups

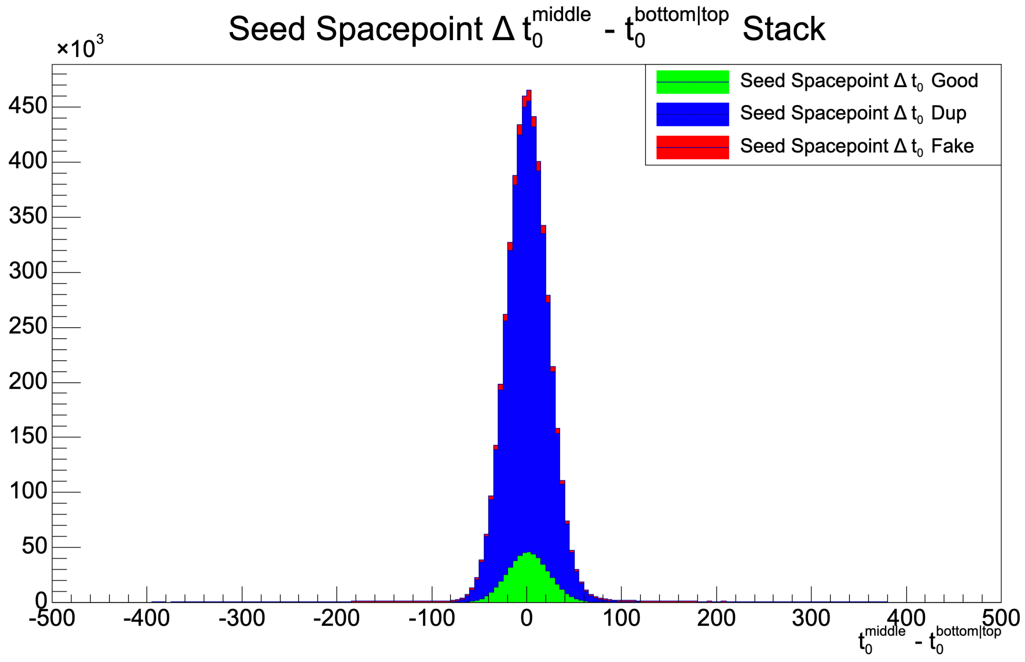


200 events $t\bar{t} <\mu>= 200$, $\sigma_t = 50\ ps$, Fatras simulation

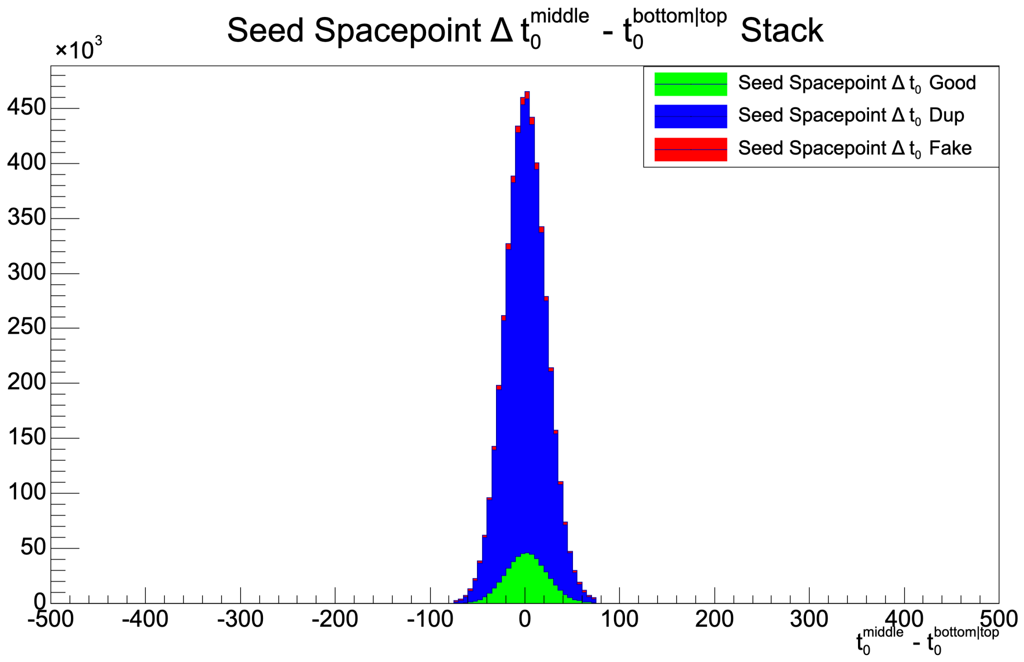
$t\bar{t} <\mu>= 200$	No time cut	$10\sigma_t$	$5\sigma_t$	$3\sigma_t$
Seeding efficiency	98.09%	98.11%	98.12%	98.06%
Number of seeds	21426 / event	21153 / event	20975 / event	20548 / event
Number of fake	1187 / event	815 / event	645 / event	567 / event

How does $\Delta t_0 = t_0^{middle} - t_0^{top|bottom}$ work?

ITk $t\bar{t}$ with 200 pile-ups | stack Δt_0 histogram



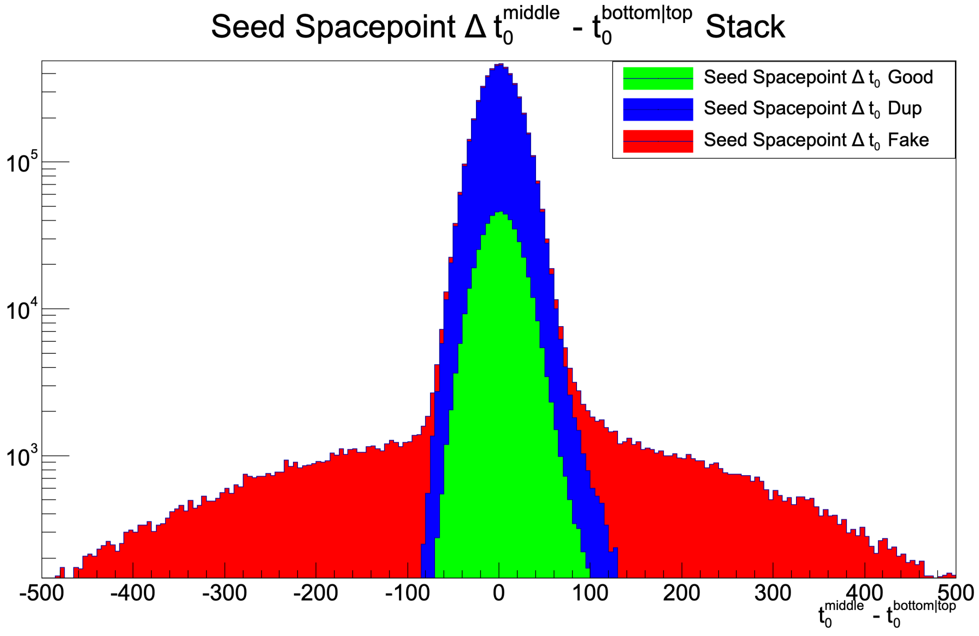
No time cut



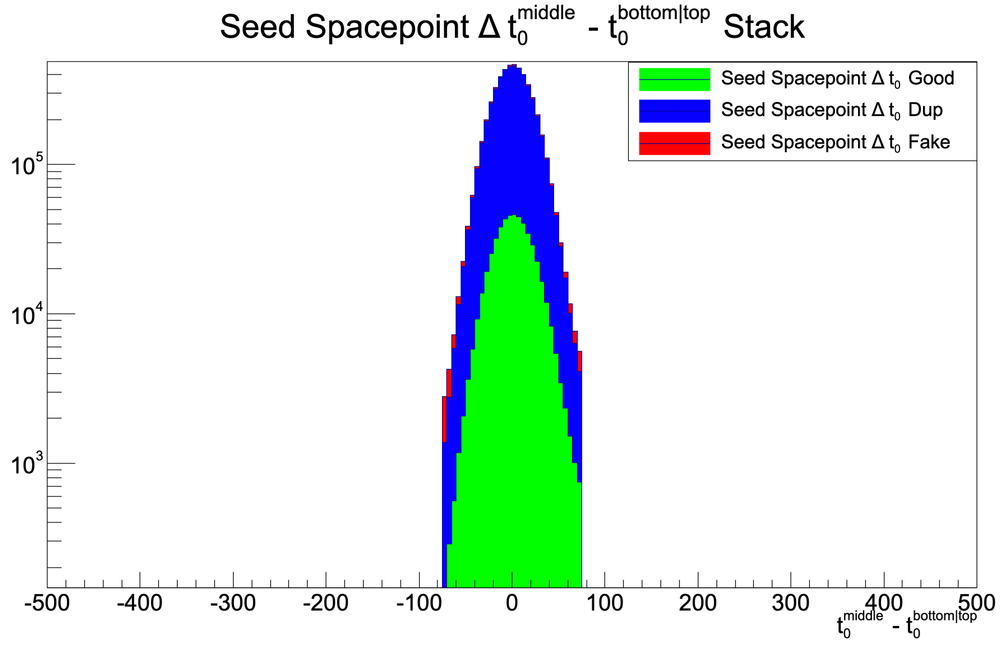
$5\sigma_t$ cut ($\sigma_t = 50\ ps$)

How does $\Delta t_0 = t_0^{middle} - t_0^{top|bottom}$ work?

ITk $t\bar{t}$ with 200 pile-ups | stack Δt_0 histogram (log scale)



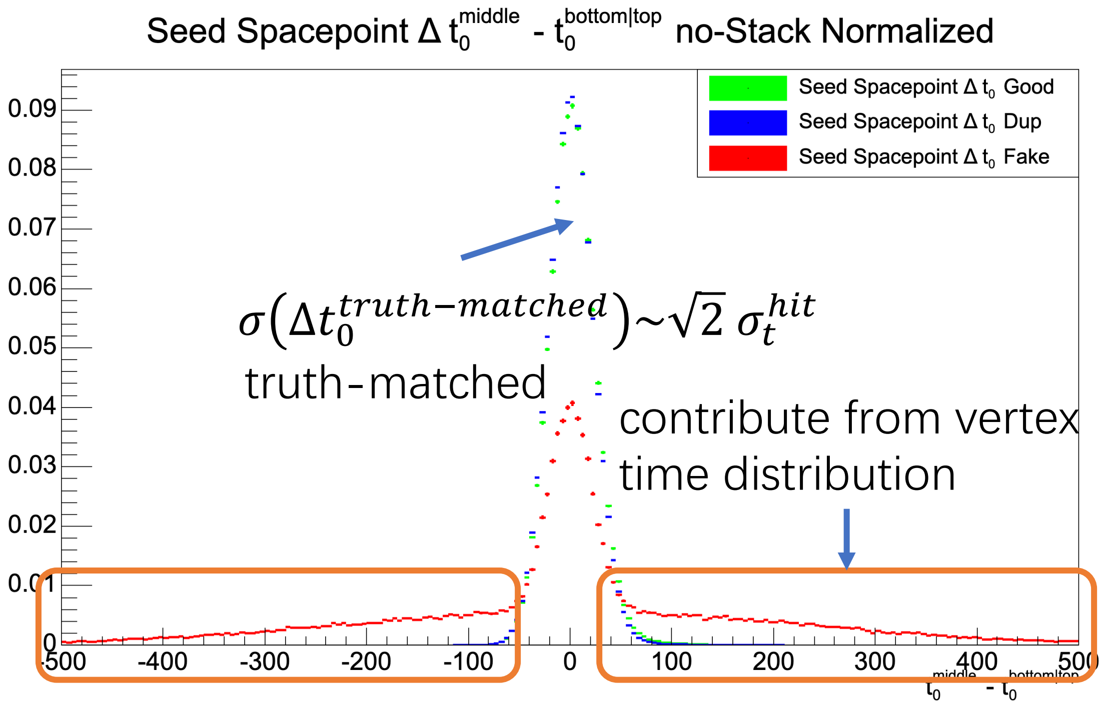
No time cut



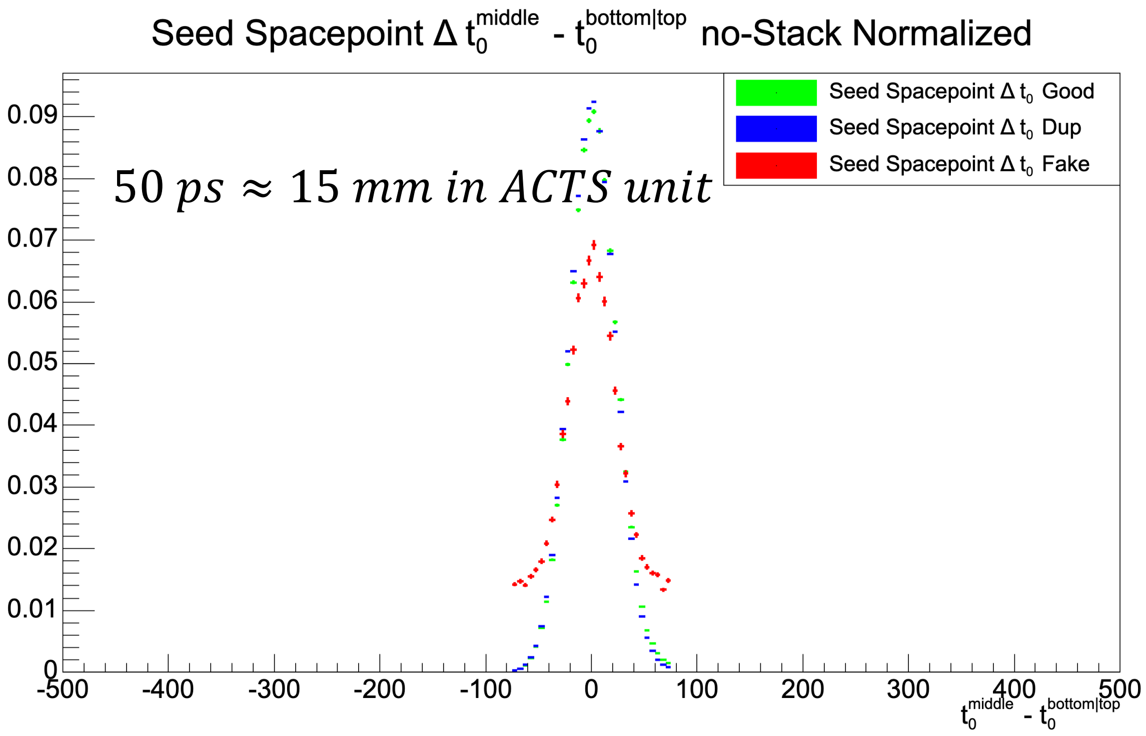
$5\sigma_t$ cut ($\sigma_t = 50\ ps$)

How does $\Delta t_0 = t_0^{middle} - t_0^{top|bottom}$ work?

ITk $t\bar{t}$ with 200 pile-ups | normalized Δt_0 distribution

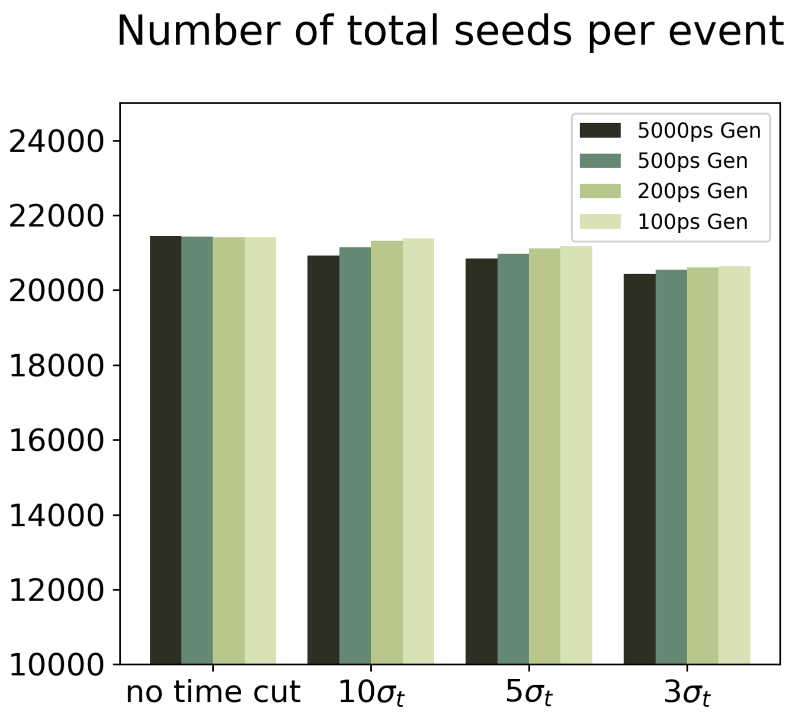


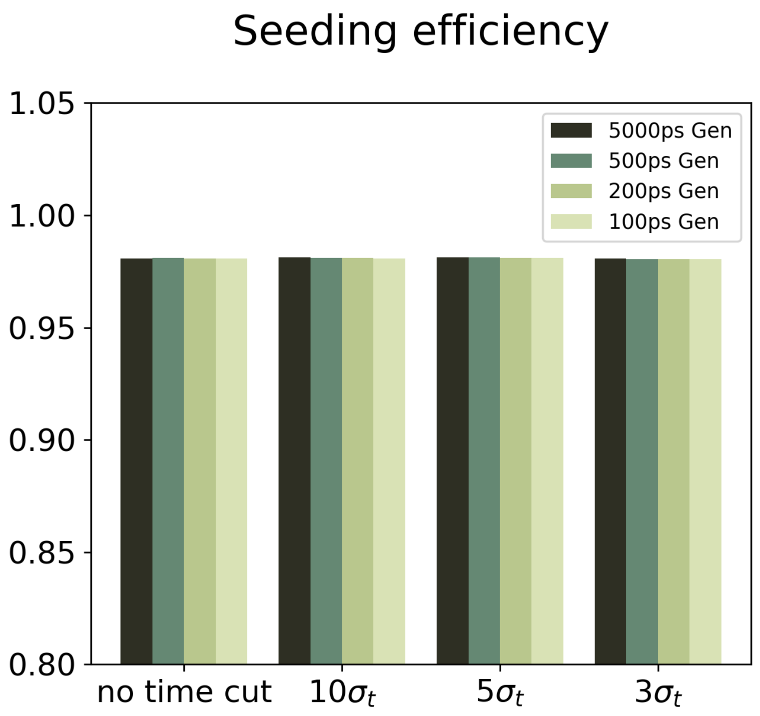
No time cut

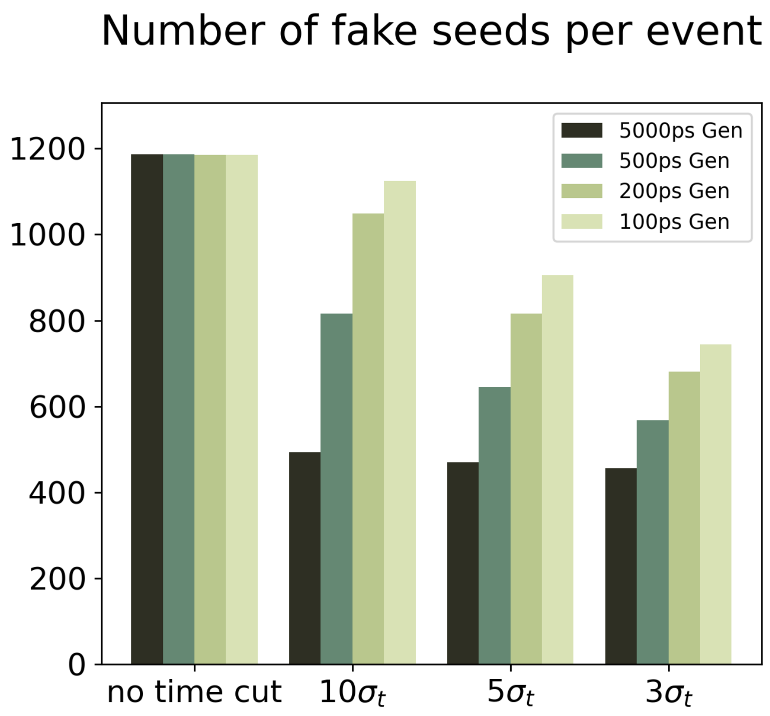


$5\sigma_t$ cut ($\sigma_t = 50\ ps$)

Different simulated vertex time distribution





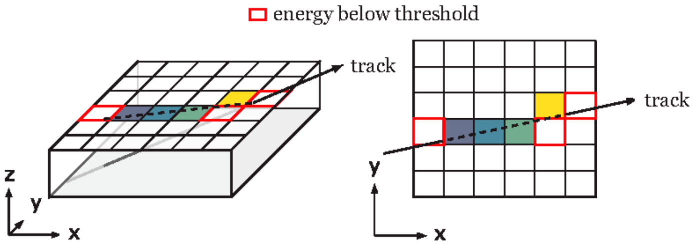


The power of identifying fake doublets will drop, as the vertex time distribution decreases

Timing pixel digitization in Athena

Workflow of time digitization in Athena

- Digitization
  - simulate detector response from hit simulation information
  - HIT (Geant4 simulation) -> RDO (Raw data output)
  - (position, charge…) -> (identifier/position of cells, ToT, ToA)



Graphs from [ACTS docs]

Geant4 hit time

smeared time
(add random gaussian distribution)

simulate pulse and calculate ToA
based on parameterized pulse waveform

fast time digitization

digitized time
stored in RDO

more realistic digitization
(the same method with HGTD)

example 1.732 ns

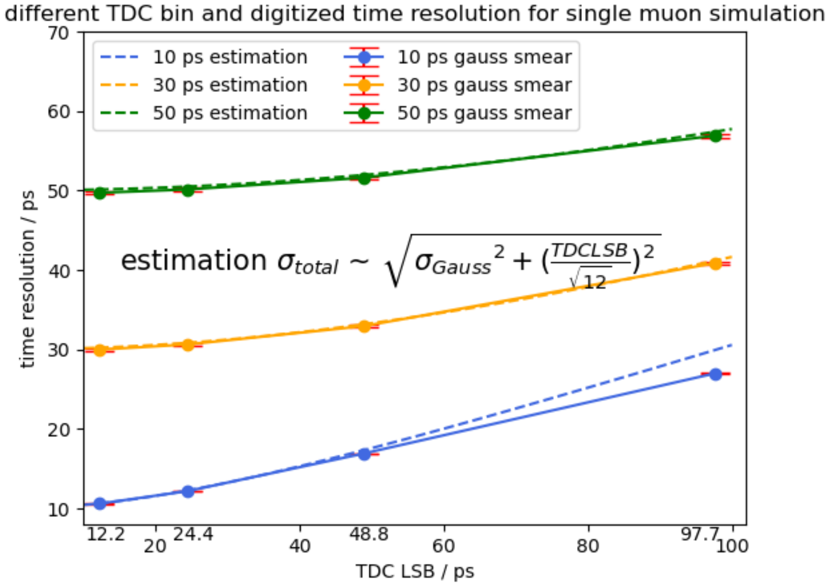
1.851 ns

00011010…

Fast time digitization

- Add gaussian distribution to Geant4 hit time , and store it in RDO.
- Can configure **time resolution** (σ_{Gauss}) and **TDC LSB**, to reach different total time resolution (σ_{total}).
- For fast time digitization $\sigma_{total} \sim \sqrt{\sigma_{Gauss}^2 + (\frac{TDC\ LSB}{\sqrt{12}})^2}$
  - Different components are represented by σ_{Gauss}
- Realistic condition

$$\sigma_{total} \sim \sqrt{\sigma_{Landau}^2 + \sigma_{time-walk}^2 + \sigma_{jitter}^2 + \sigma_{clock}^2 + (\frac{TDC\ LSB}{\sqrt{12}})^2}$$

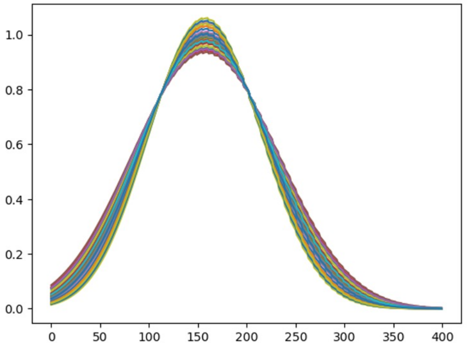


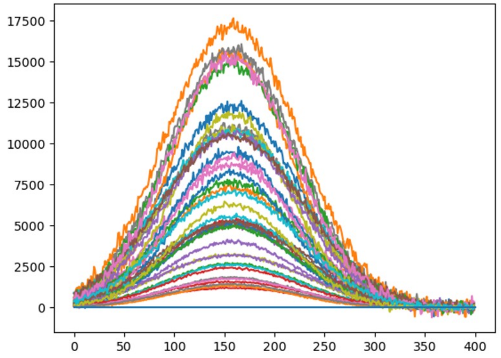
Gaussian fit results of histograms of $t_{reconstructed} - t_{hit}$, with different σ_{Gauss} *and TDC LSB* scenarios.

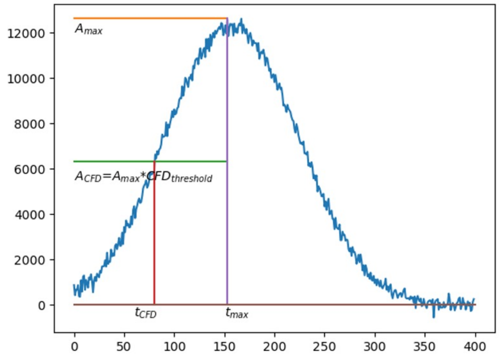
$$\sigma_{total} \sim \sqrt{\sigma^2_{Landau} + \sigma^2_{time-walk} + \sigma^2_{jitter} + \sigma^2_{clock} + \left(\frac{TDC\ LSB}{\sqrt{12}}\right)^2}$$

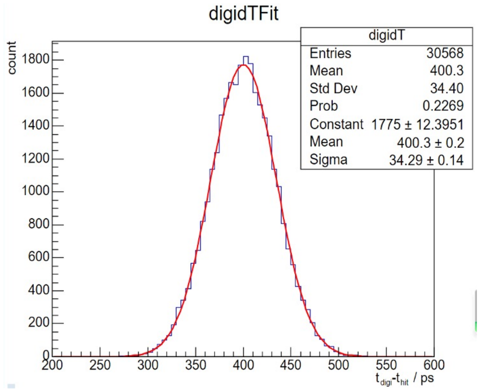
More realistic time digitization (HGTD method)

- Implementation
  - Simulate pulse → calculate pulse → timing pulse → store in RDO









bit-word

  - Simplified digitization using parameterized pulse waveform.
  - Consider Landau (gauss) , time-walk, jitter, clock (gauss), TDC LSB
- ITk Pixel waveform is not applied. HGTD pulse waveform is still kept as a place holder now.

Summary

Thanks!

- Validate full chain of simulation & tracking workflow in standalone ACTS for 4D tracking study

- Investigating possible 4D tracking seeding algorithm in ACTS
  - No seeding efficiency loss, while suppressing rate of fake seeds
- Working on implementing 4D tracking workflow in Athena
  - 4D digitization algorithm is implemented for ITk
  - 4D seeding will be implemented in Athena using the outputs of 4D digitization and 4D clustering (almost ready by Spyros)
- Outlook: physics case using the 4D tracking + 4D vertexing + 4D flavor tagging

Back-ups

Definition of duplication rate and fake rate

- Track candidates classification

Track candidates (after ambiguity solver)

Track selection

- **Reconstructed**
  - Match
    - **Truth-matched**
      - **Real** (only one for the same generated particle)
      - **Duplicated**
    - **Fake** (not matched with any simulated particle)
- **Unconstructed**

$$Duplication\ rate = \frac{Duplicated}{Truth\ Matched}$$

$$Fake\ rate = \frac{Fake}{Reconstructed}$$

Track selection:

- number of measurements (pixel + strip) ≥ 7,
- $p_T^{reco} > 900$ MeV (400 MeV) in $|\eta|<2.0$ ($2.0<|\eta|<4.0$)
- $|d_0^{reco}| \leq 2$ mm (10 mm) in $|\eta|<2.6$ ($2.6<|\eta|<4.0$)
- $|z_0^{reco}| \leq 20$ cm

Matched to a reconstructed track if the matching probability is larger than 50%.

- $Now \quad P_{match} = \frac{N_{Pixel}^{common} + N_{Strip}^{common}}{N_{Pixel}^{track} + N_{Strip}^{track}} = \frac{nMajorityHits}{nMeasurements}$

Definition of track efficiency

- Track efficiency definition

  Matched particle: if the selected particle have at least one **truth-matched** track associated with the particle

Particle selection

Generated particles → Selected particle → Matched particle

Generated particles → Unselected particle

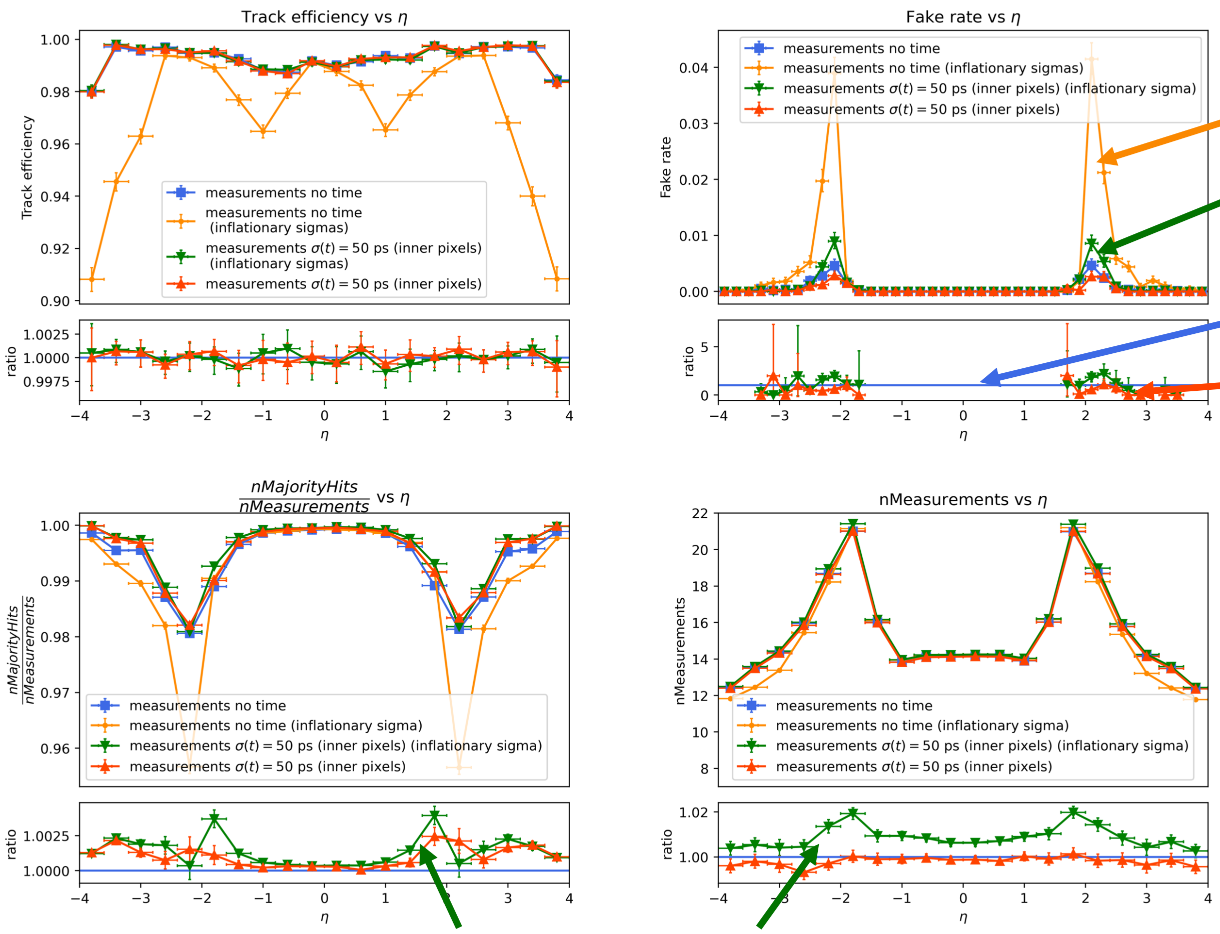
Selected particle → Unmatched particle

Particle selection:
- p_T>1 GeV and |η|<4.0,
- being produced by the primary interactions

$$Track\ efficiency = \frac{Matched\ particle}{Selected\ particle}$$

* Matched particles should satisfy Particle selection criteria, or called “matched” and “selected”.

Fatras and Default seeding



some interesting things

增大CKF初始状态协方差矩阵元

三维重建（效率97.31%，假径迹率0.436%）

四维重建（效率99.27%，假径迹率0.090%）

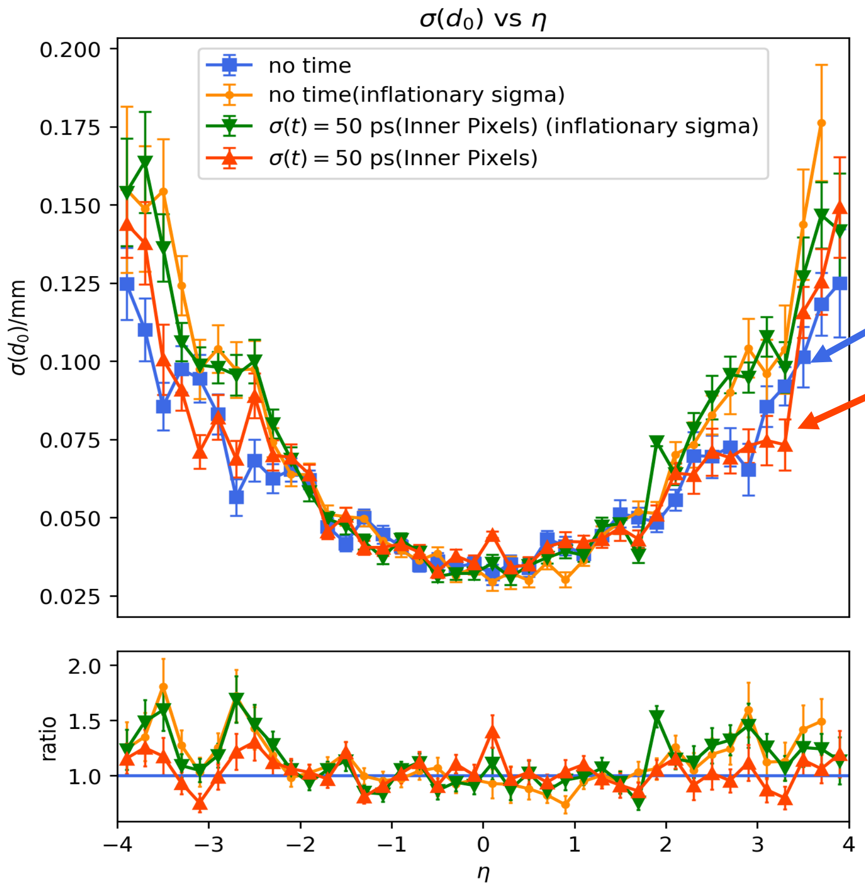
三维径迹重建基准

四维径迹重建

径迹重建效率	99.28%->99.29% -
假径迹率	0.061%->0.036% ↓

时间信息帮助**降低假径迹率**（右上），**提高匹配度**（左下），**总测量数略有下降**（右下）

Fatras and Default seeding



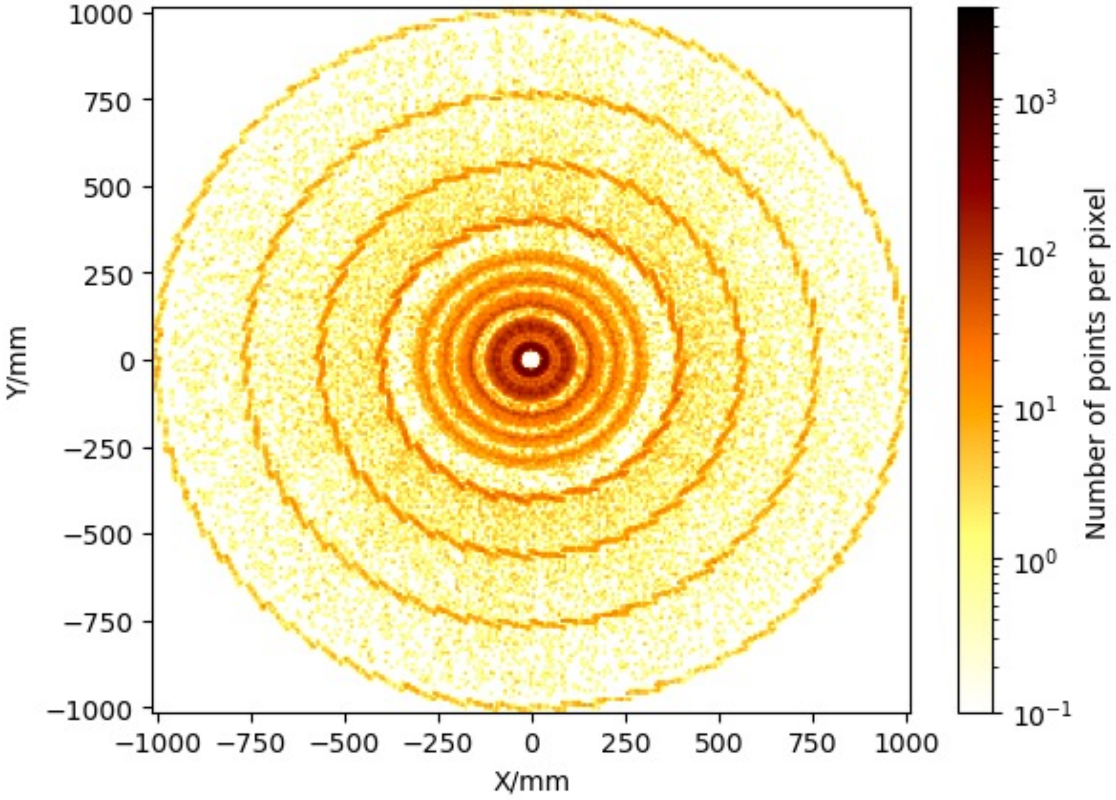
三维径迹重建基准

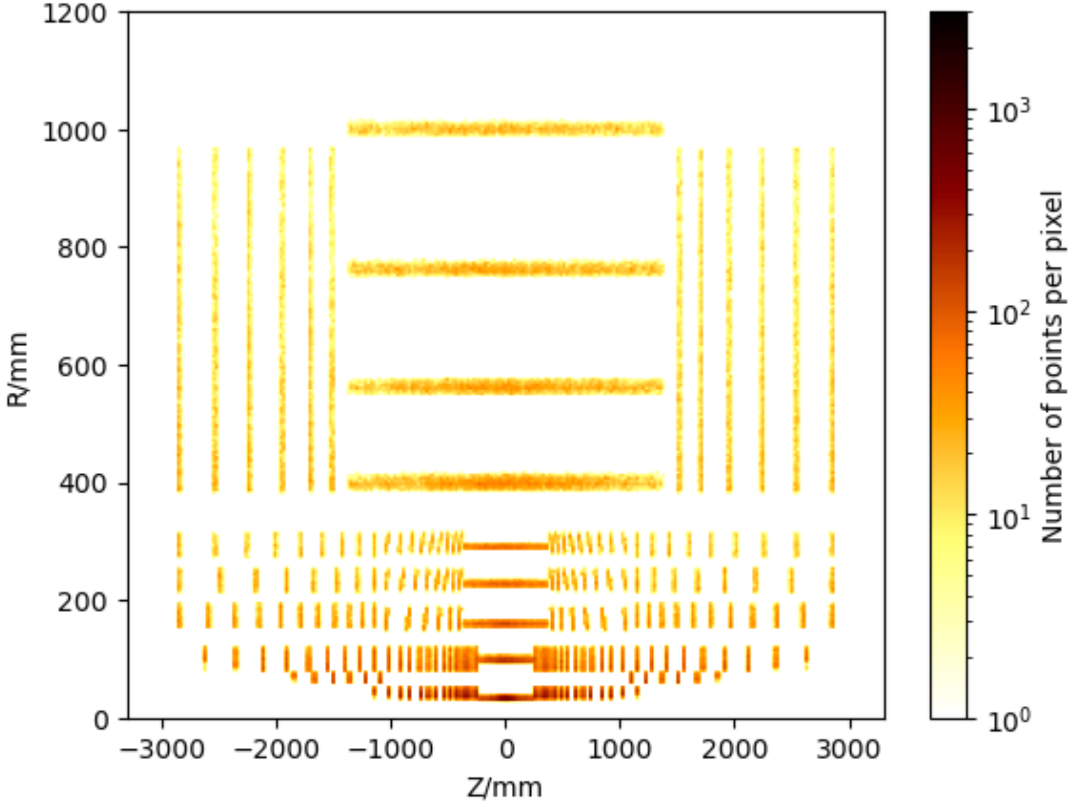
四维径迹重建

三维和四维径迹重建
碰撞参数d_0分辨没有明显变化

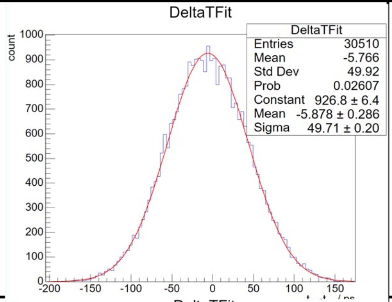
全模拟工作链几何验证

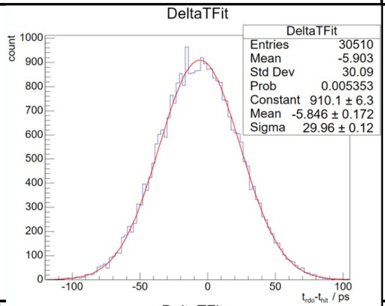
- Geant4中得到的击中信息

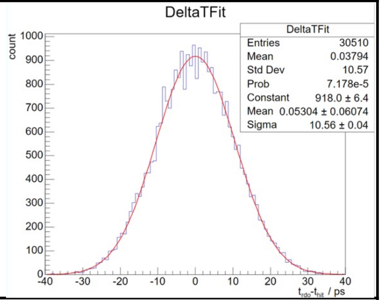


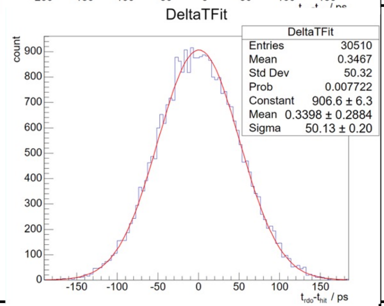


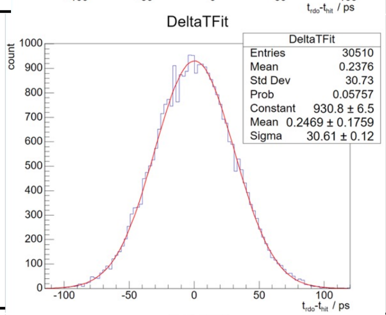
	50 ps smear	30 ps smear	10 ps smear
11-bit ToA (**12.2 ps**)	49.1	29.96	10.56
10-bit ToA (**24.4 ps**)	50.13	30.61	12.14
9-bit ToA (**48.8 ps**)	51.60	32.91	16.86
8-bit ToA (**97.7 ps**)	56.90	40.86	>26.98 be

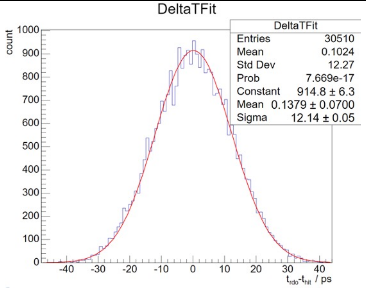


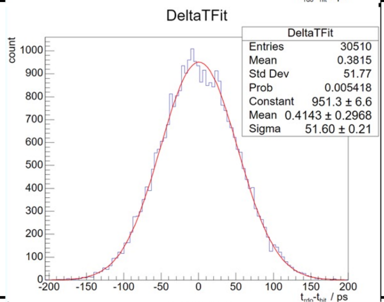


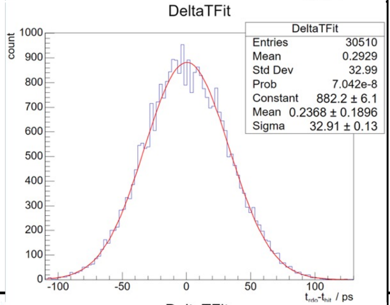


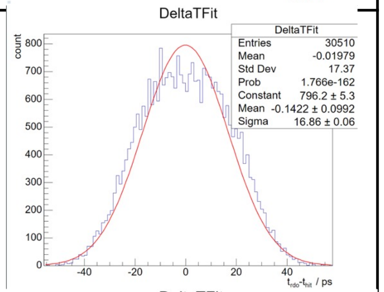


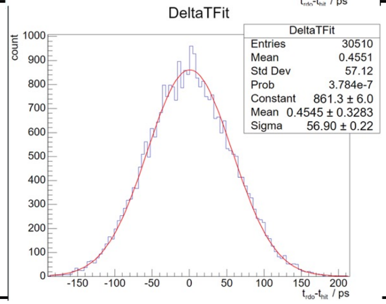


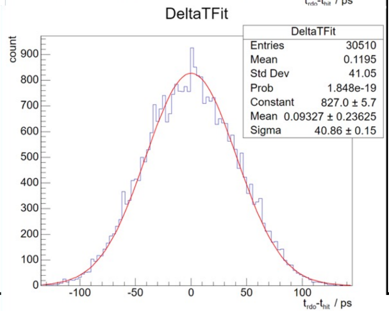


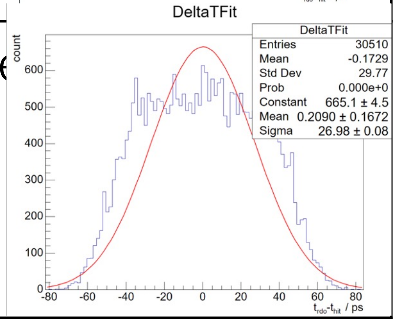










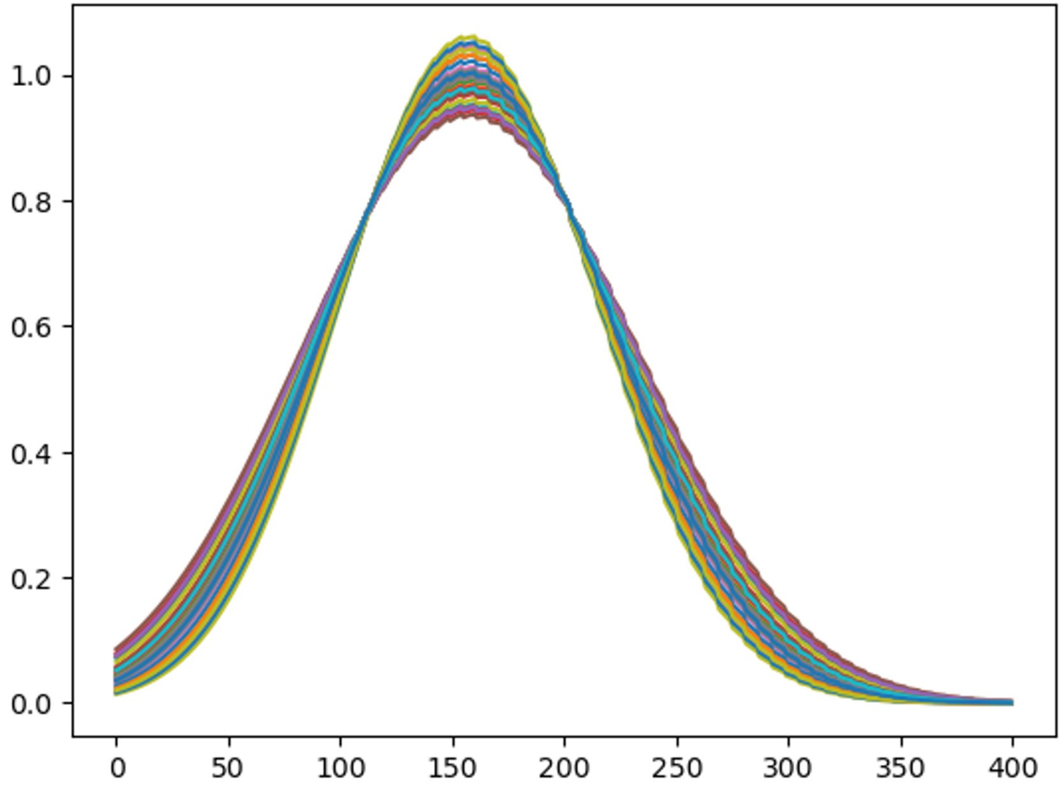


3. More realistic time digitization (HGTD method)

Simulate pulse

- Convolution of gauss and landau to parameterize the waveform[a]test beam
- 400 points – 2 ns, so = 5 ps/point[a2]
- Using one random parameter in landau function to change **the**

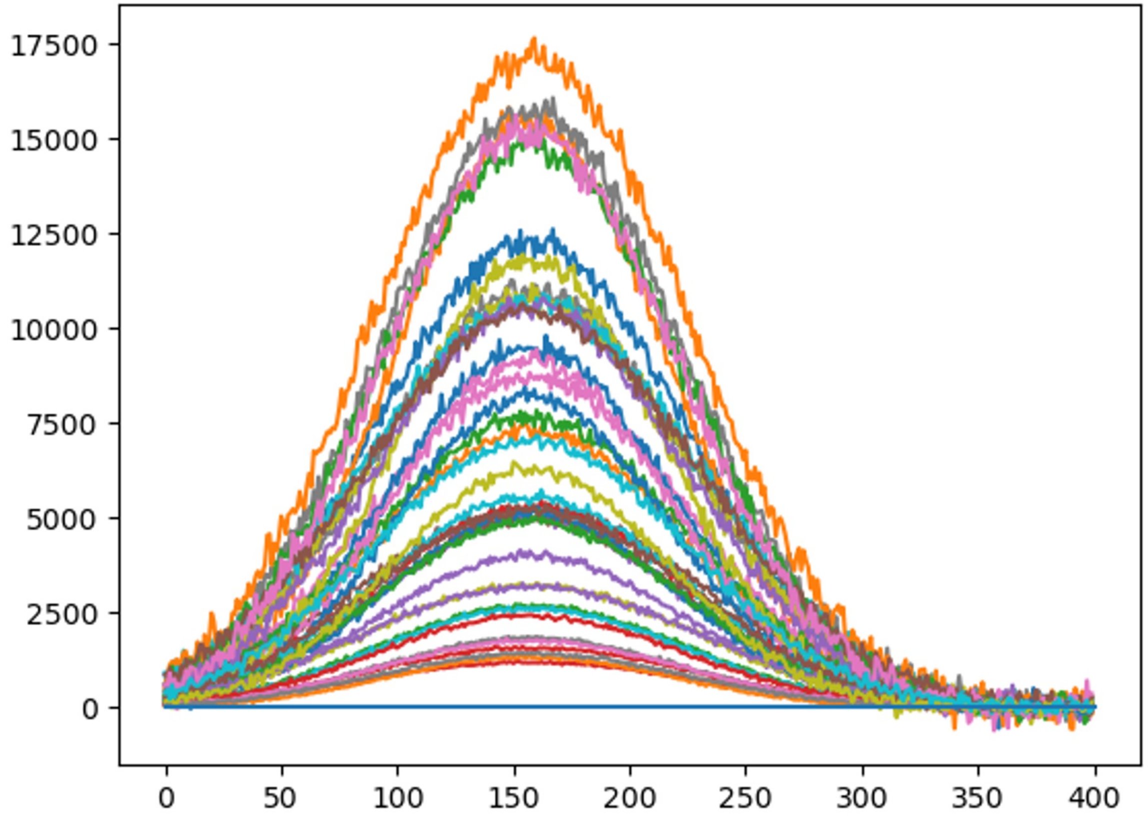
Time-walk **pe of the waveform**[1] **test beam**



3. More realistic time digitization (HGTD method)

Calculate pulse

- Scale the waveform by energy
- Add **random noise**[2] to waveform
  - This could be confused if the random noise is totally proportional to the magnitude[b] of the waveform (or the charge) (may need more reference later, what about low energy deposit???)



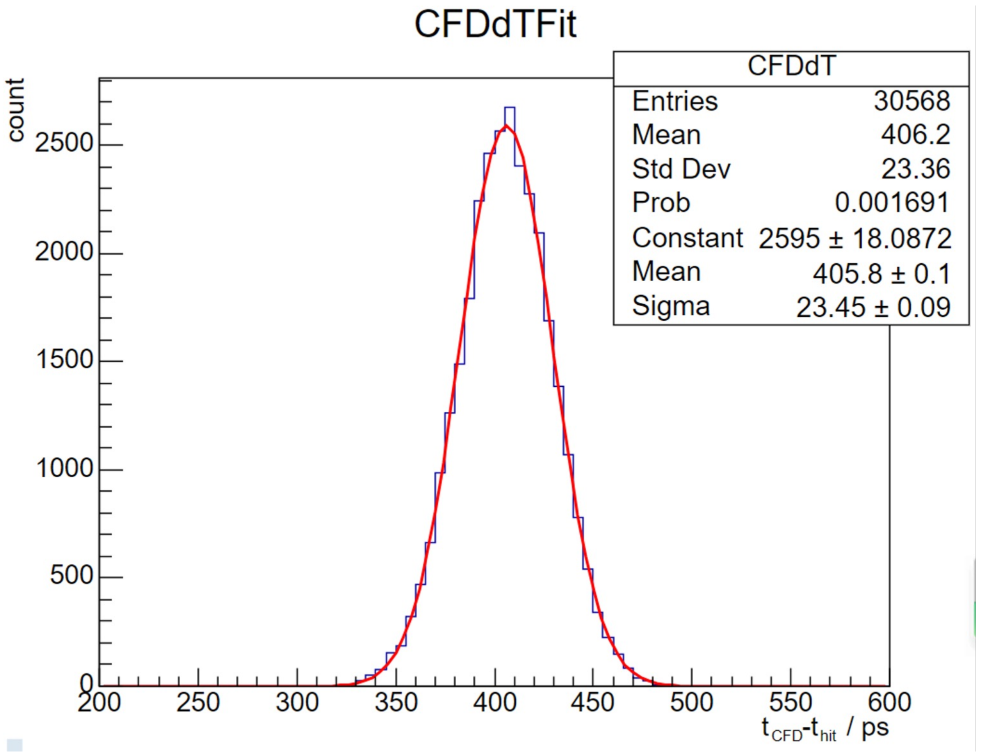
jitter

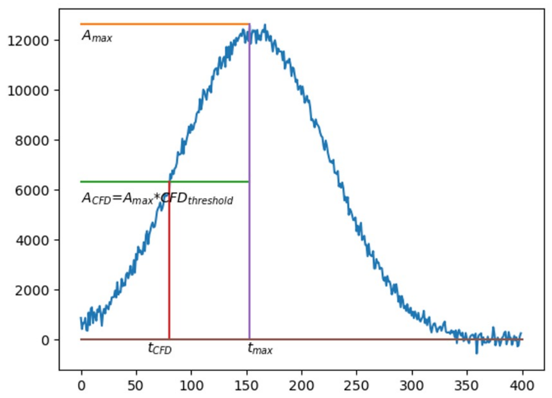
```
energy = pulseWaveform[point] * E +
         CLHEP::RandGaussZiggurat::shoot(rndm_engine, 0, 0.015) * E;
```

3. More realistic time digitization (HGTD method)

Timing pulse

- Use constant fraction discriminator[b] to determine the raise CFD time (float)
  - Notice that we have granularity of 5ps/point when simulating pulse waveform. (linear interpolation?)





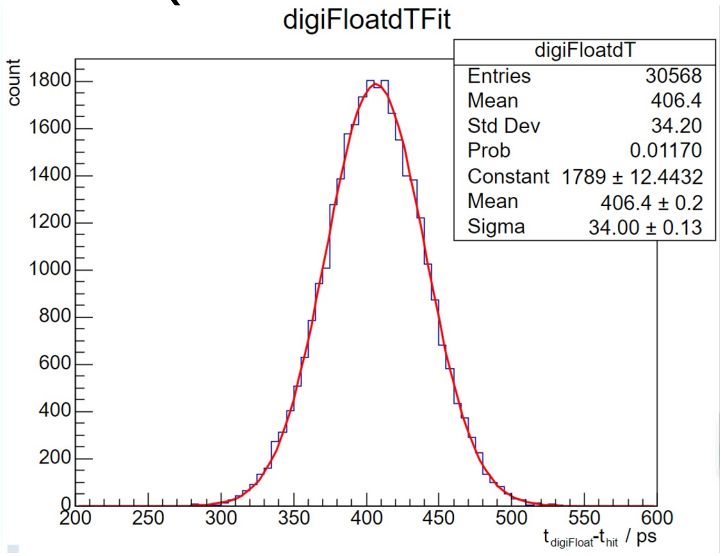
3. More realistic time digitization (HGTD method)

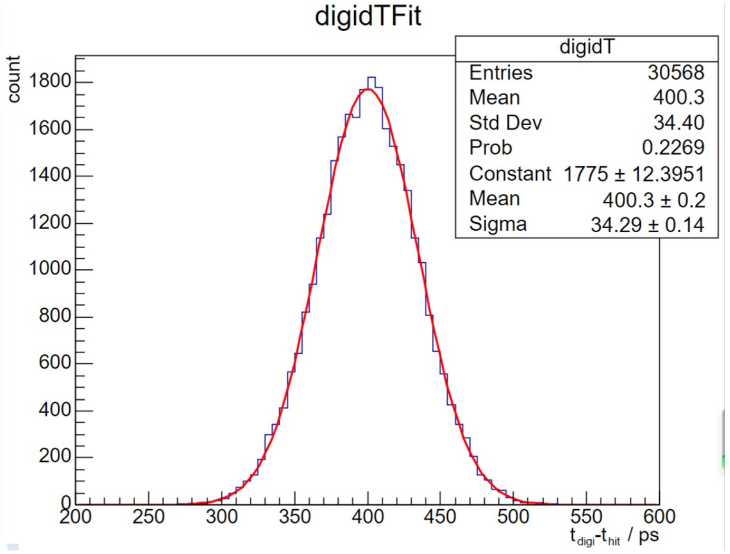
Give ToA Landau+clock

- **Smear**[3] time given by last step (CFD) with gauss distribution.

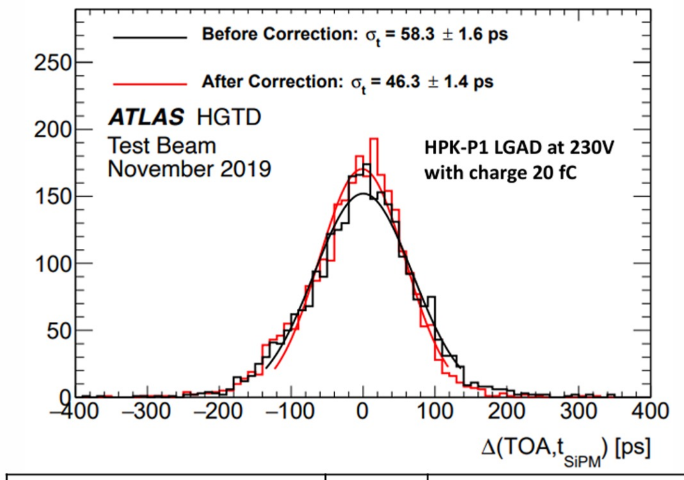
Store time in TDC bin

- **Encode**[4] TDC bin

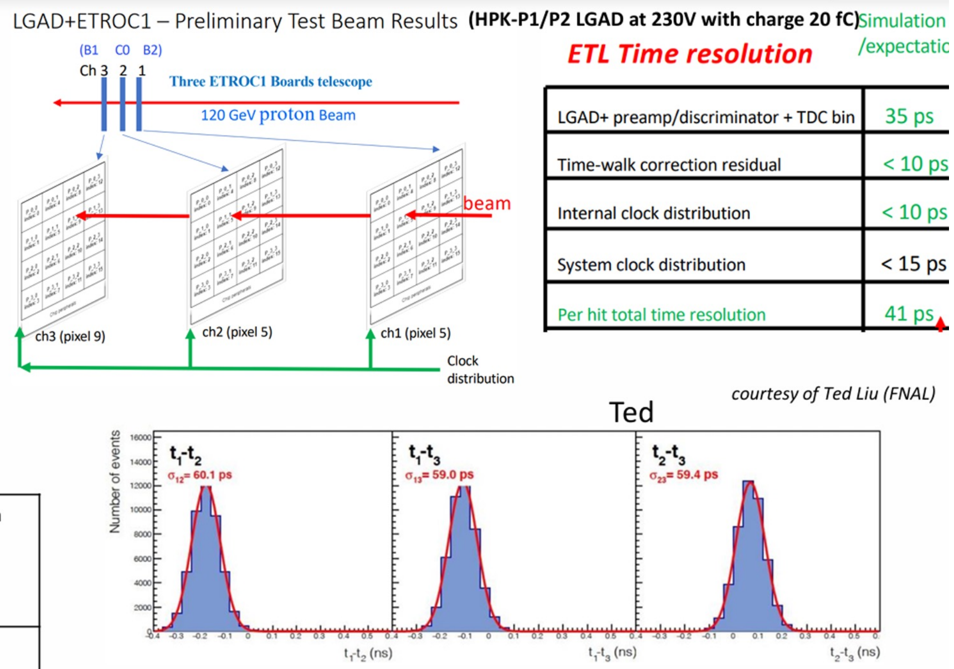


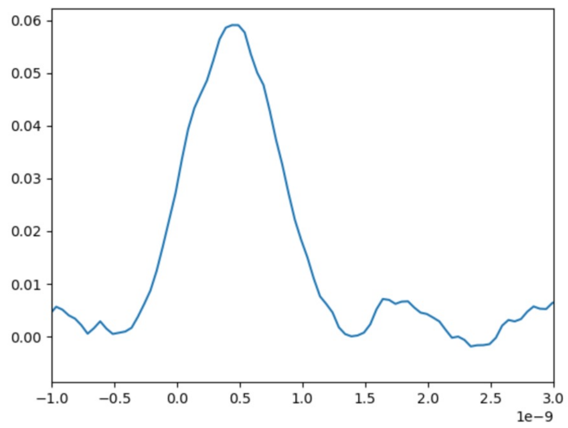


Time resolution (HGTD LGAD)



LGAD Landau	25 ps	the reason for part of the high jitter found **26 (known reasons)**
Jitter+system/internal clock+time walk residual	37 ps	
TDC clock	7 ps	Total resolution: 36 ps (likely achievable for 20 fC)/ 70 ps for (4 fC)
Per hit total	46 ps	

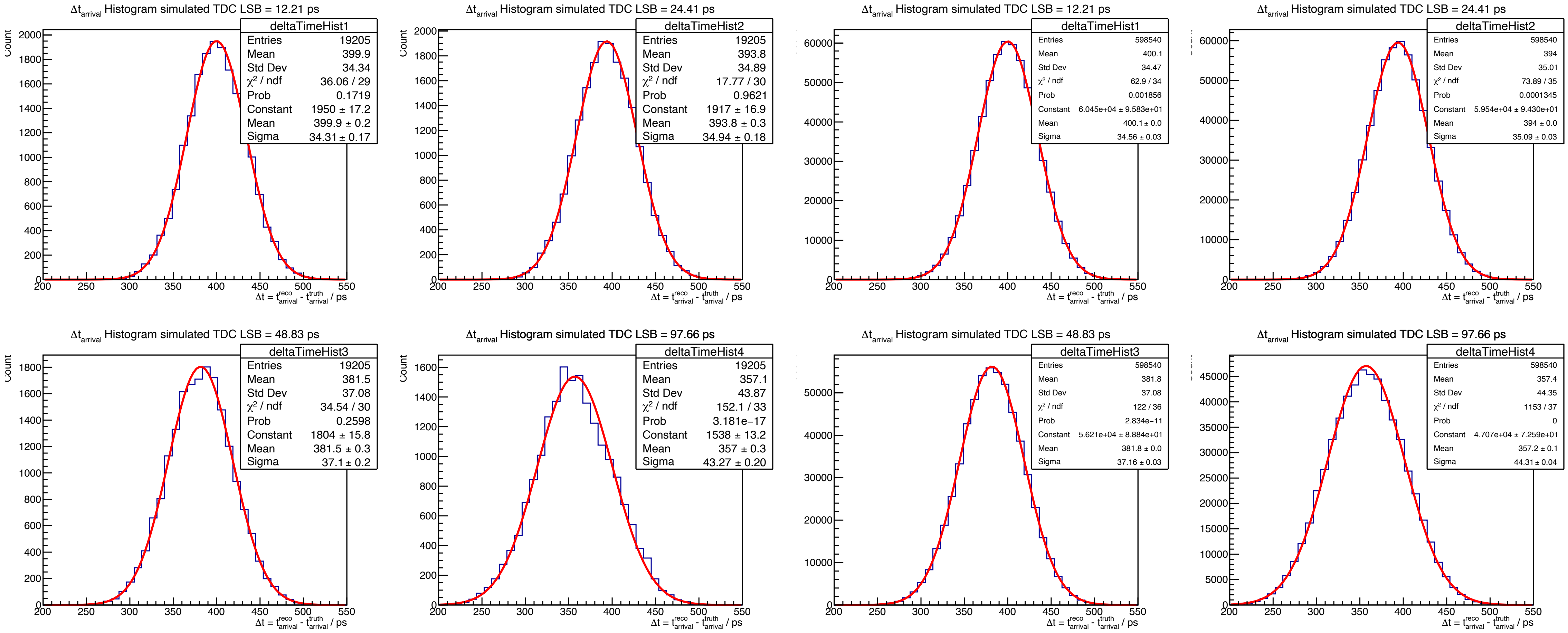




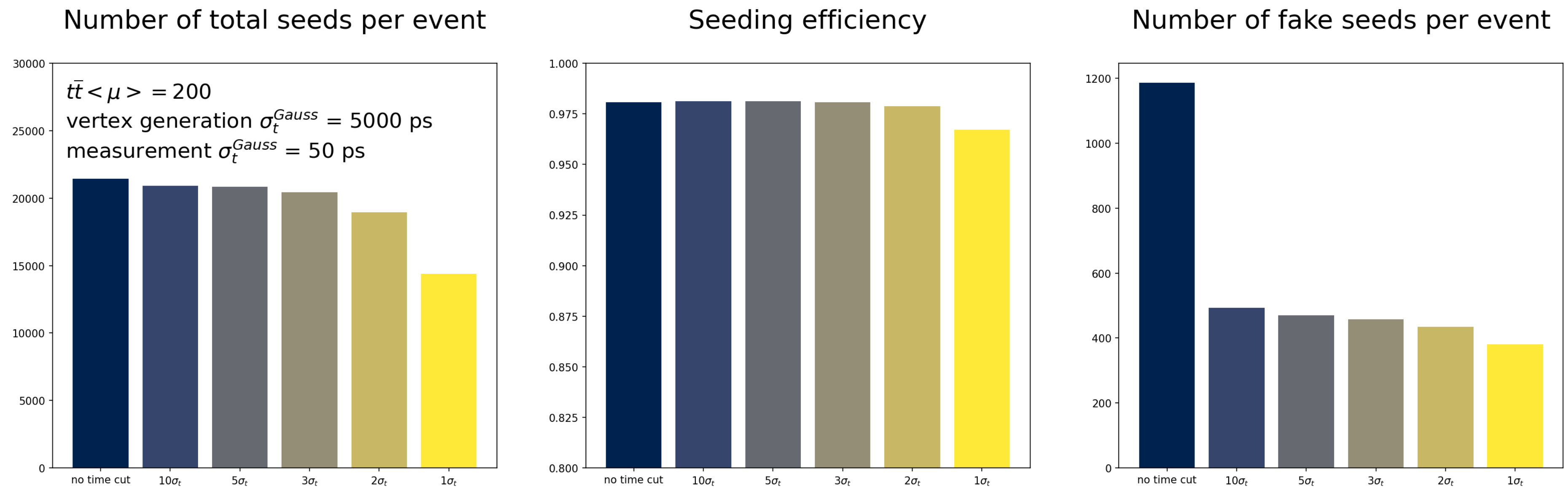
Pulse waveform of HGTD

Single muon 1000 events

$t\bar{t} <\mu> = 200$



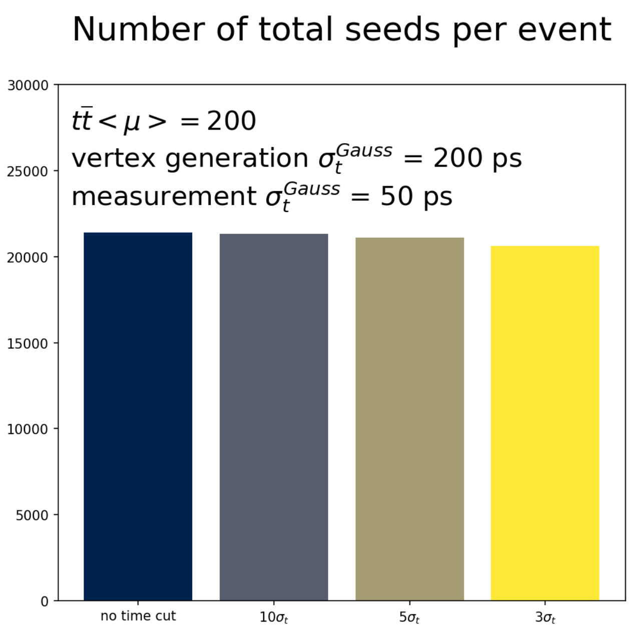
ITk $t\bar{t}$ with 200 pile-ups (Gen 5000ps)

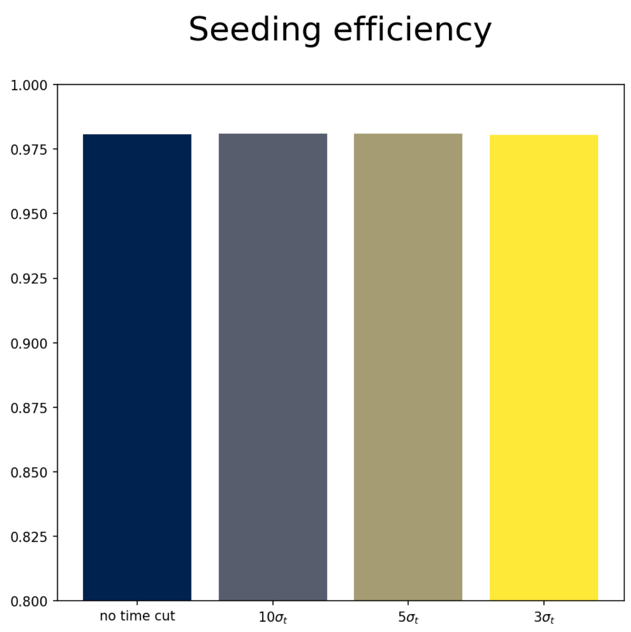


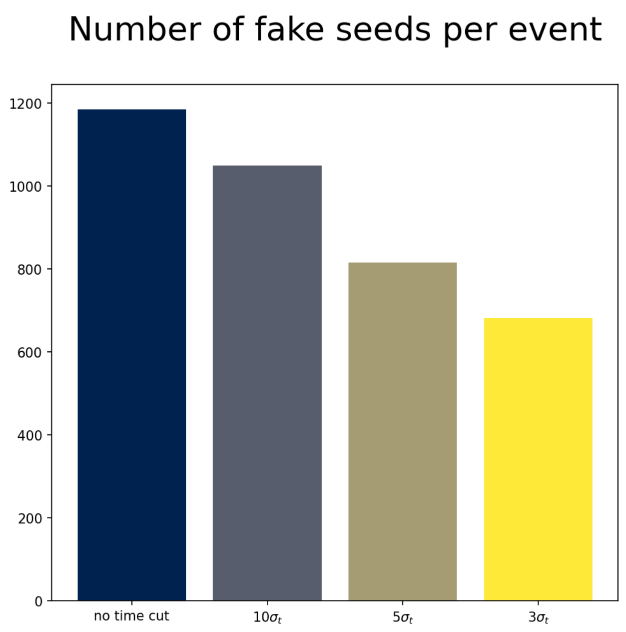
200 events $t\bar{t} < \mu >= 200$, $\sigma_t = 50\ ps$, Fatras simulation

$t\bar{t} < \mu >= 200$	No time cut	$10\sigma_t$	$5\sigma_t$	$3\sigma_t$	$2\sigma_t$	$1\sigma_t$
Seeding efficiency	98.08%	98.13%	98.12%	98.07%	97.87%	96.71%
Number of seeds	21451 / event	20918 / event	20847 / event	20430 / event	18959 / event	14379 / event
Number of good	2077 / event	2081 / event	2077 / event	2066 / event	2027 / event	1881 / event
Number of fake	1187 / event	493 / event	470 / event	457 / event	435 / event	380 / event

ITk $t\bar{t}$ with 200 pile-ups (Gen 200ps)





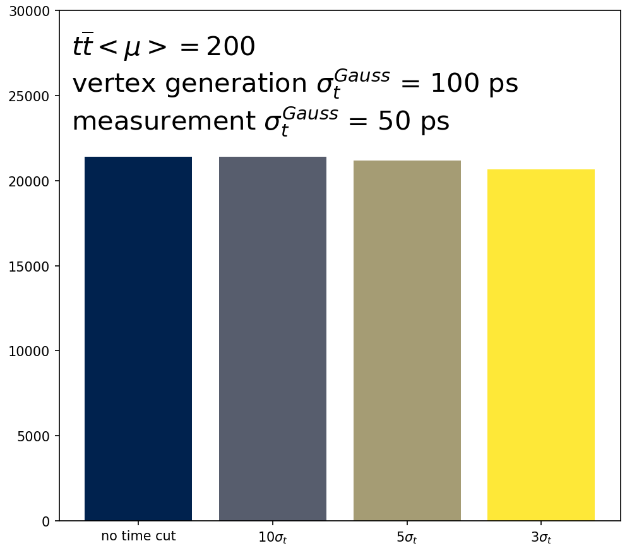


200 events $t\bar{t} <\mu>= 200$, $\sigma_t = 50\ ps$, Fatras simulation

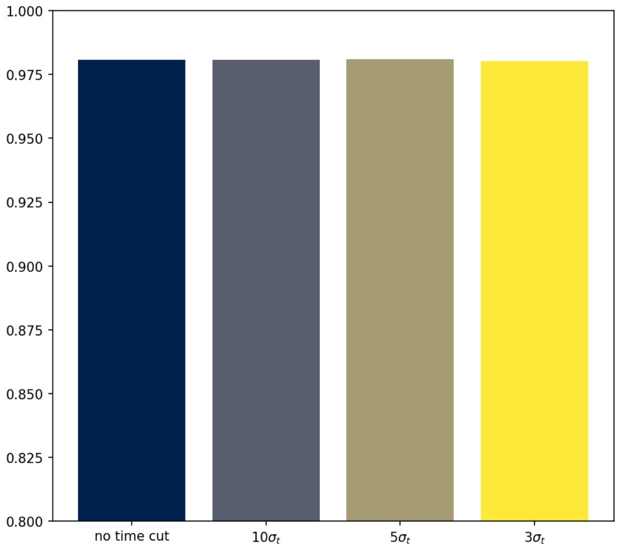
$t\bar{t} <\mu>= 200$	No time cut	$10\sigma_t$	$5\sigma_t$	$3\sigma_t$
Seeding efficiency	98.09%	98.10%	98.10%	98.05%
Number of seeds	21426 / event	21314 / event	21118 / event	20613 / event
Number of fake	1185 / event	1049 / event	816 / event	681 / event

ITk $t\bar{t}$ with 200 pile-ups (Gen 100ps)

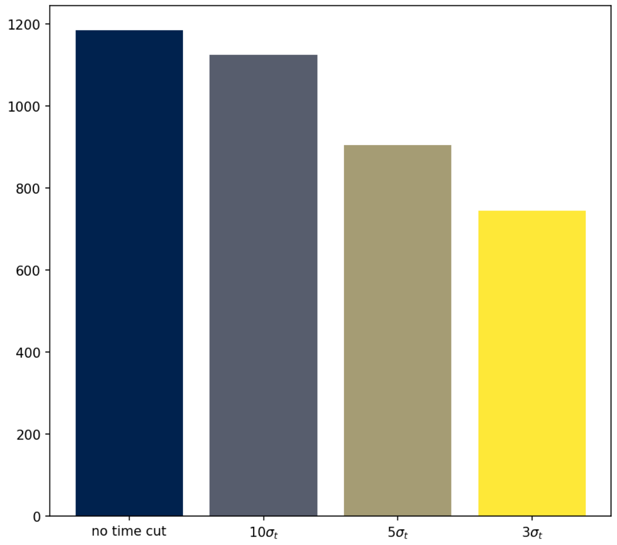
Number of total seeds per event



Seeding efficiency



Number of fake seeds per event

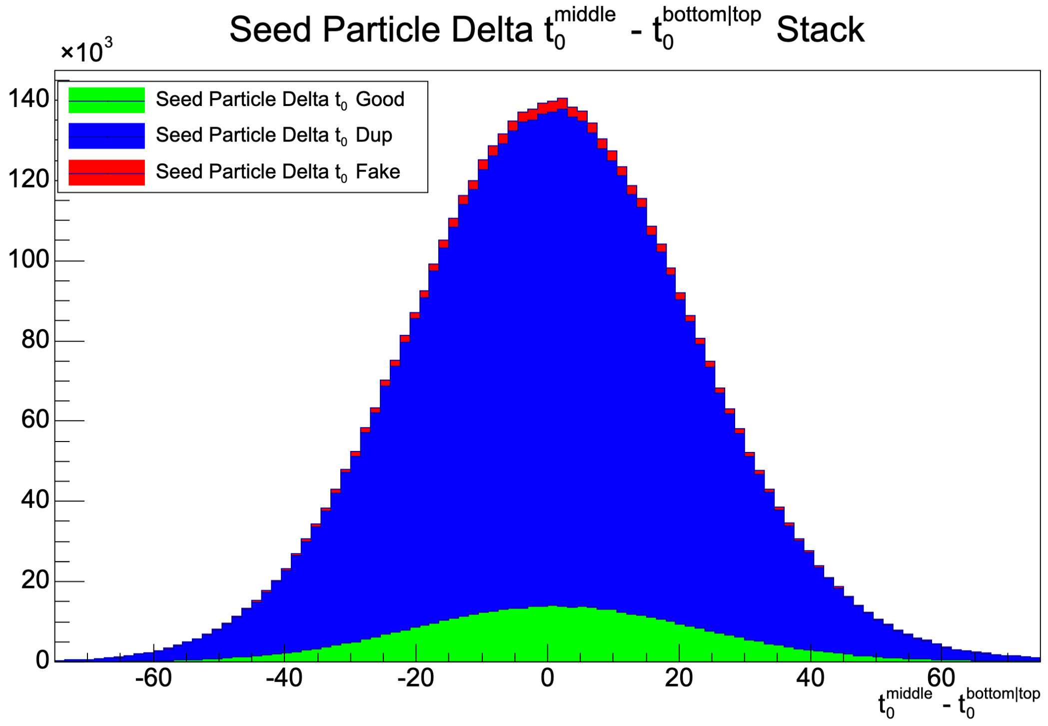


200 events $t\bar{t} < \mu >= 200$, $\sigma_t = 50\ ps$, Fatras simulation

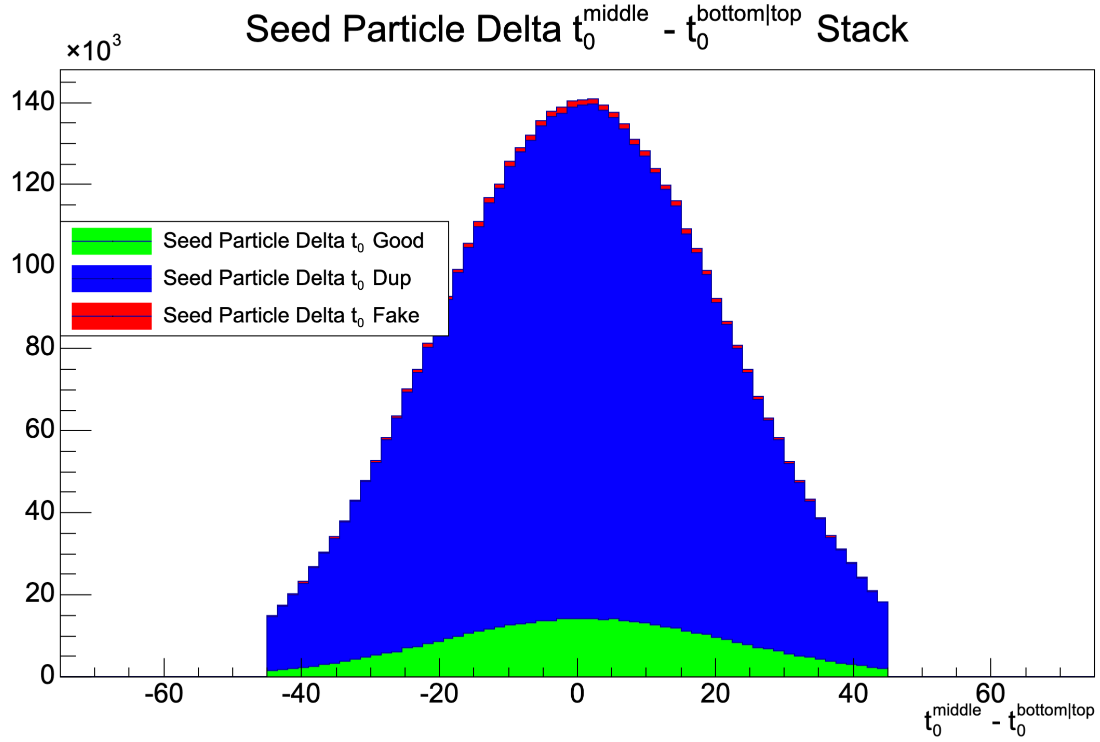
$t\bar{t} < \mu >= 200$	No time cut	$10\sigma_t$	$5\sigma_t$	$3\sigma_t$
Seeding efficiency	98.09%	98.08%	98.10%	98.03%
Number of seeds	21421 / event	21388 / event	21182 / event	20643 / event
Number of fake	1185 / event	1125 / event	905 / event	744 / event

How does $\Delta t_0 = t_0^{middle} - t_0^{top|bottom}$ work?

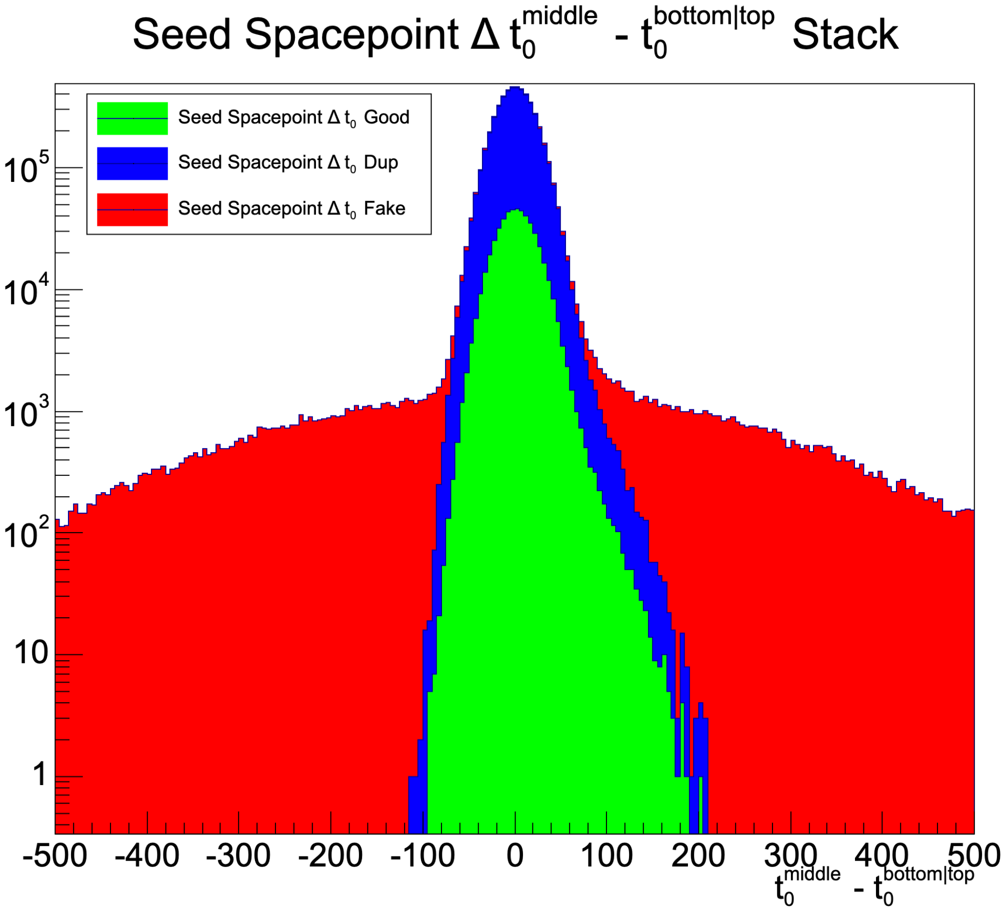
ITk $t\bar{t}$ with 200 pile-ups | stack Δt_0 histogram (5 ns Gen)



No time cut



$3\sigma_t$ cut ($\sigma_t = 50\ ps$)



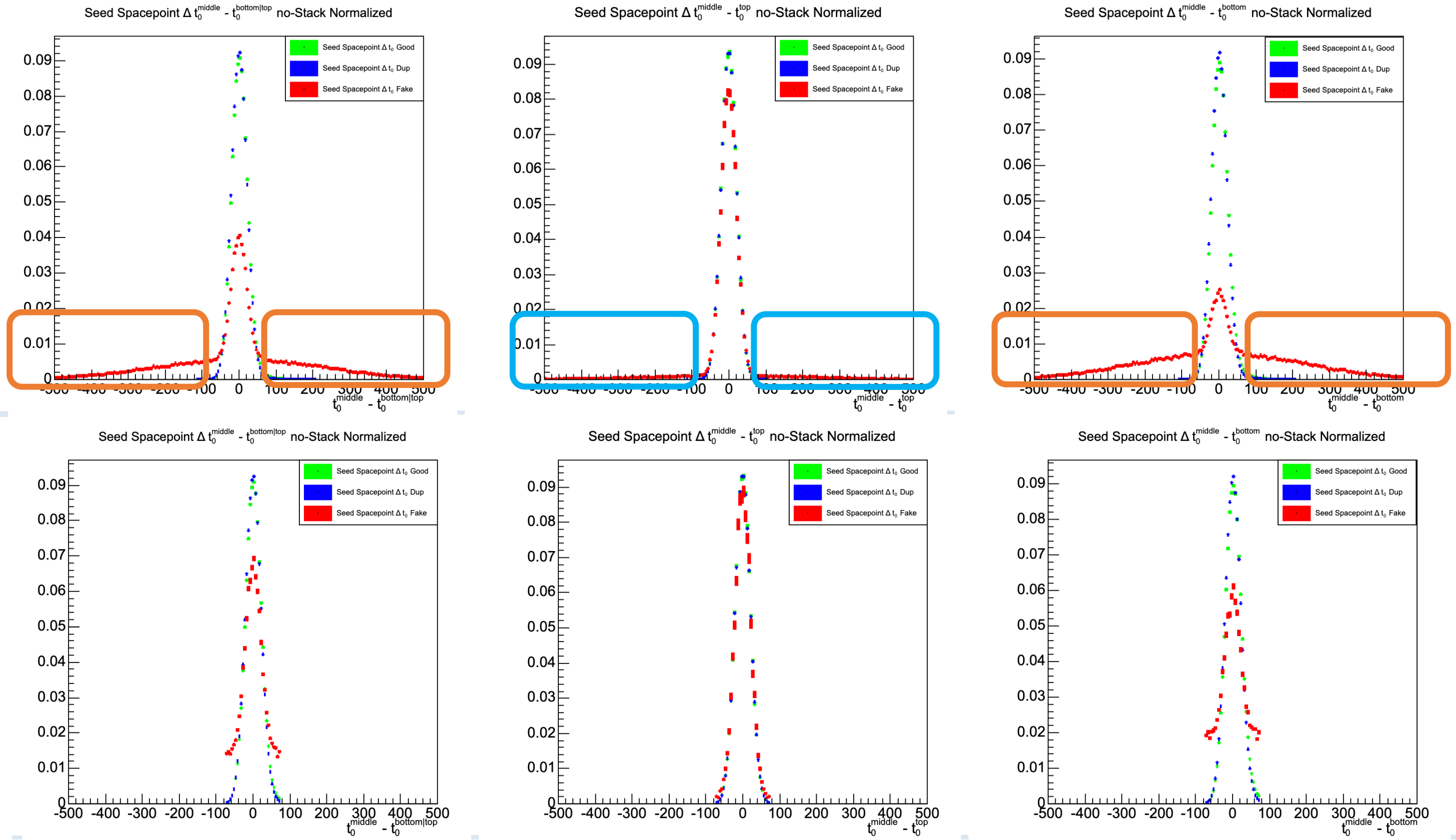
Seed Spacepoint $\Delta\, t_0^{middle} - t_0^{bottom|top}$ Stack
Seed Spacepoint $\Delta\, t_0$ Good
Seed Spacepoint $\Delta\, t_0$ Dup
Seed Spacepoint $\Delta\, t_0$ Fake
10^5
10^4
10^3
10^2
10
1
-500 -400 -300 -200 -100 0 100 200 300 400 500
$t_0^{middle} - t_0^{bottom|top}$

How does $\Delta t_0 = t_0^{middle} - t_0^{top|bottom}$ work?

ITk $t\bar{t}$ with 200 pile-ups | normalized $\Delta t_0^{middle-top}$ and $\Delta t_0^{middle-bottom}$ distribution

No time cut

$5\sigma_t$ cut ($\sigma_t = 50\ ps$)



Middle-bottom spacepoint doublets contribute

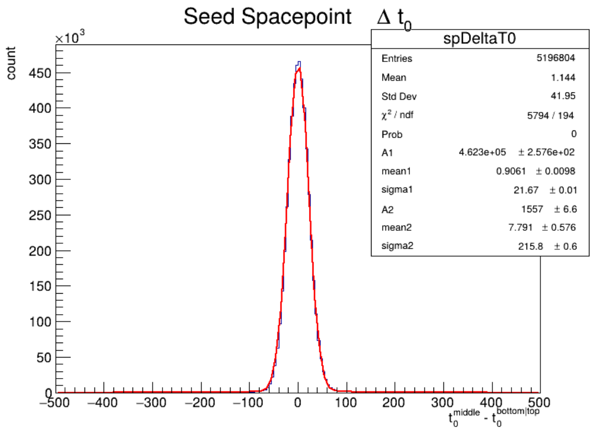
Number of timing sp in fake seeds per event

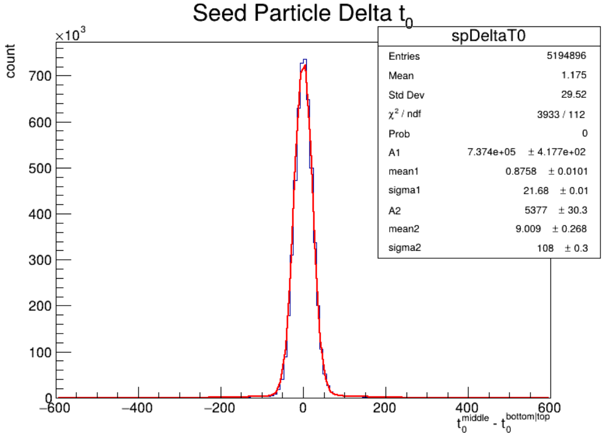
1253->498 (-60%)

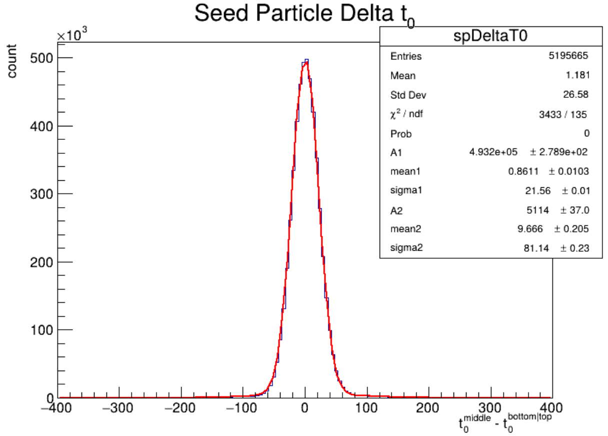
337->146(-57%)

916->352(-62%)

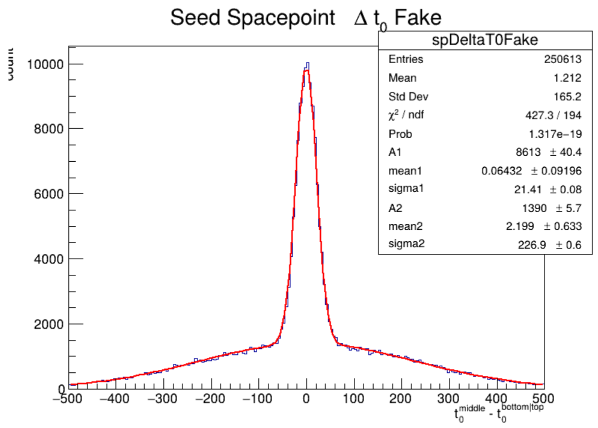
Vertex distribution / ps	5000	500	200	100
σ_{time1} / mm		21.67	21.68	21.56
σ_{time2} / mm		215.8	108	81.14
fake σ_{time1} / mm		21.41	21.62	21.57
fake σ_{time2} / mm		226.9	114.2	87.49
fake σ_{time2} / ps		757	381	292
$\sqrt{2\sigma_{t\|hit}^2 + 2\sigma_{t\|vertex}^2}$ /ps		711	292	158

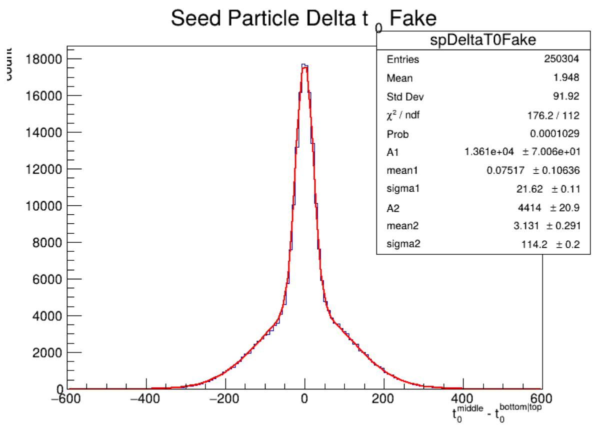


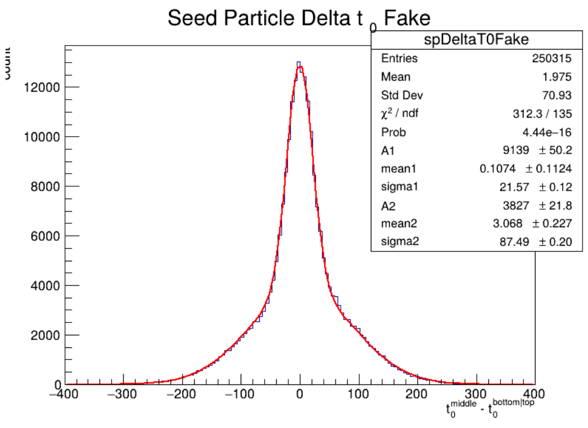


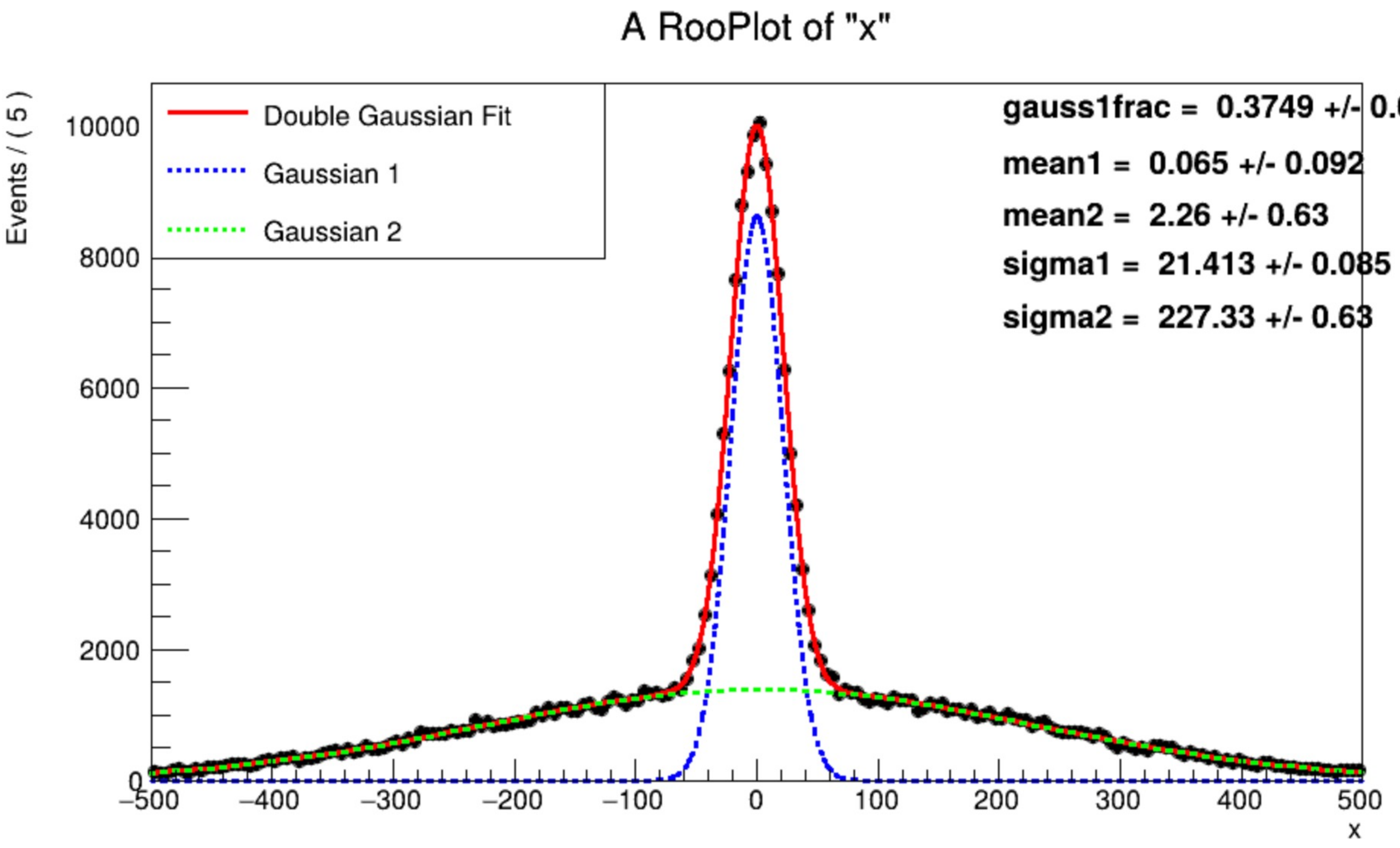


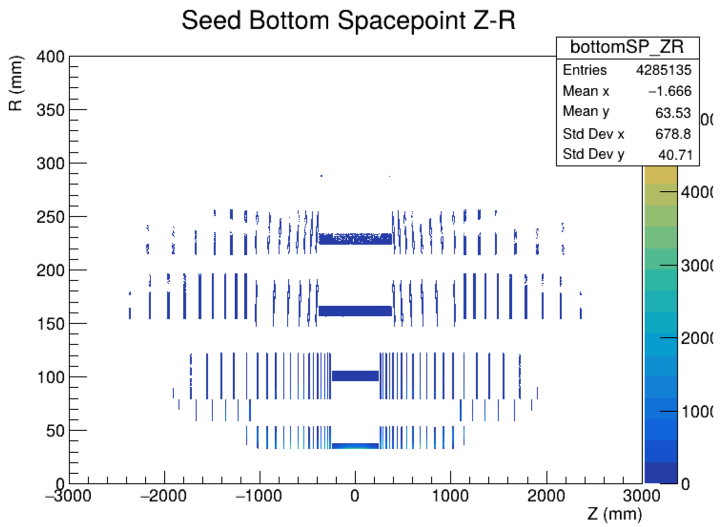
$$\sqrt{2\sigma_{t|hit}^2} = 21.199 \text{ mm}$$

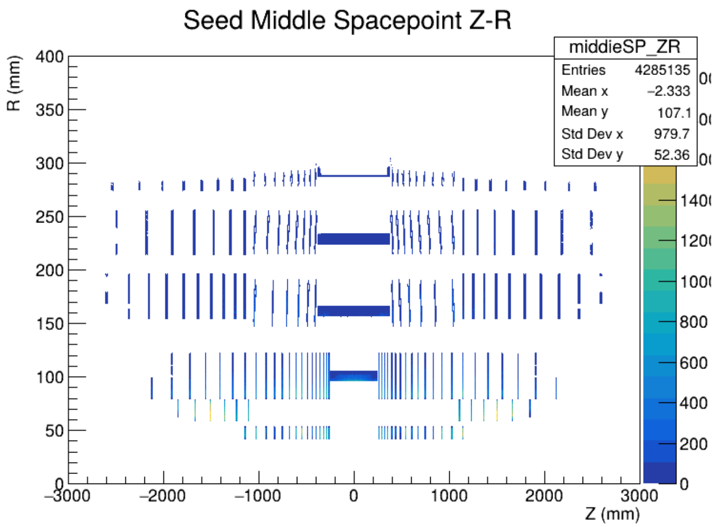


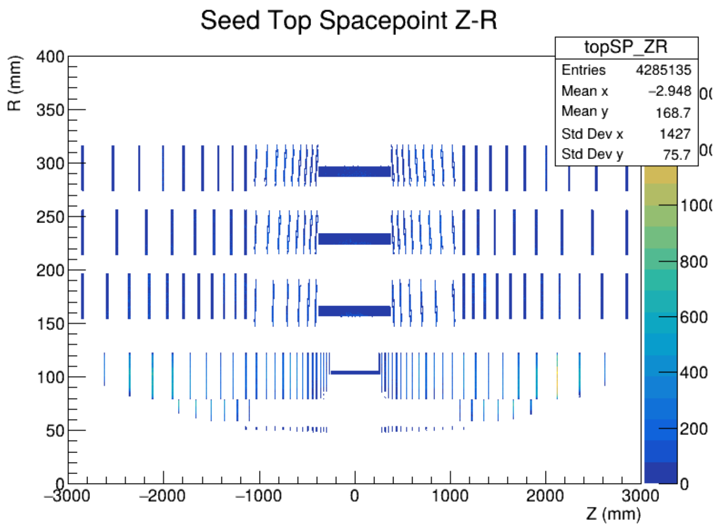





A RooPlot of "x"
Double Gaussian Fit
Gaussian 1
Gaussian 2
gauss1frac = 0.3749 +/- 0.0013
mean1 = 0.065 +/- 0.092
mean2 = 2.26 +/- 0.63
sigma1 = 21.413 +/- 0.085
sigma2 = 227.33 +/- 0.63
Events / (5)
x
10000
8000
6000
4000
2000
0
−500
−400
−300
−200
−100
0
100
200
300
400
500


Seed Bottom Spacepoint Z-R
bottomSP_ZR
Entries 4285135
Mean x −1.666
Mean y 63.53
Std Dev x 678.8
Std Dev y 40.71
R (mm)
Z (mm)


Seed Middle Spacepoint Z-R
middleSP_ZR
Entries 4285135
Mean x −2.333
Mean y 107.1
Std Dev x 979.7
Std Dev y 52.36
R (mm)
Z (mm)


Seed Top Spacepoint Z-R
topSP_ZR
Entries 4285135
Mean x −2.948
Mean y 168.7
Std Dev x 1427
Std Dev y 75.7
R (mm)
Z (mm)