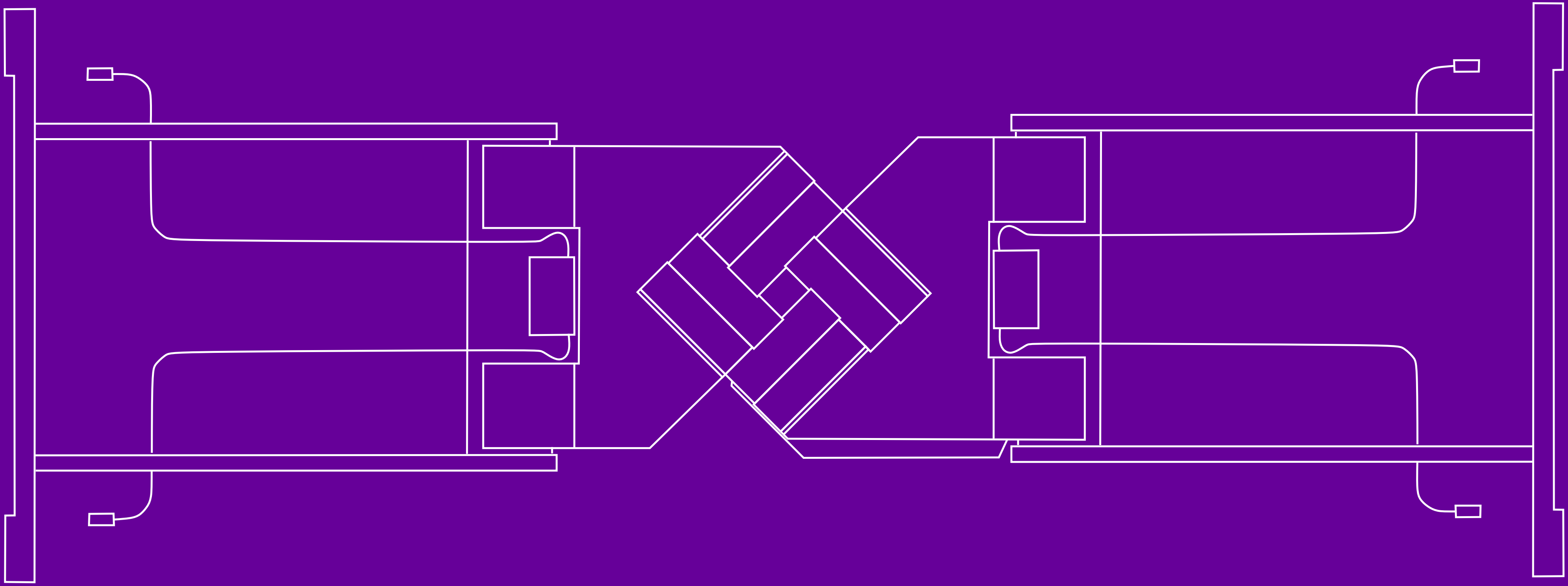
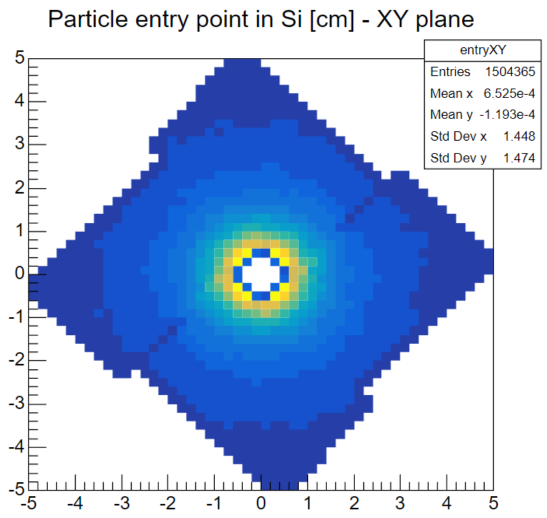


MANCHESTER 1824

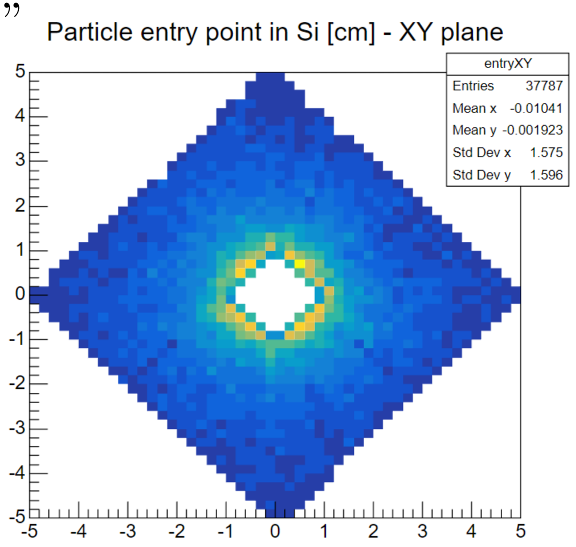
The University of Manchester

- Scenario “X”
  - 7.2 mm beam-to-sensor edge
  - (For now) keep ∼ module layout of Upgrade-I VELO (side length 14.4 mm)
  - $256 \times 256 < 55\,\mu m$ pixels, 50 ps hit resolution → 20 ps / track
  - Cylindrical foil @ 3.5 mm

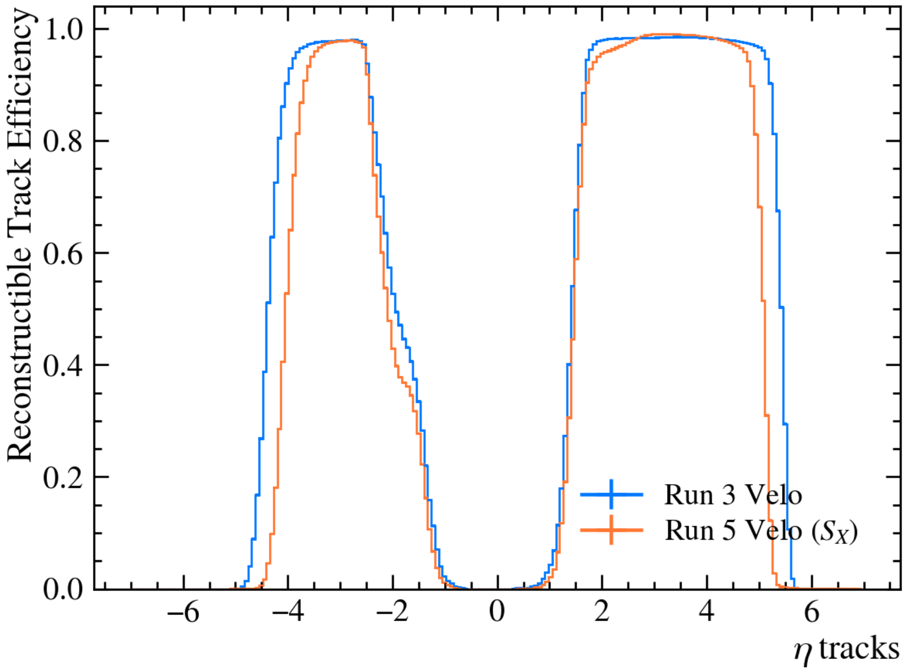
“VP”



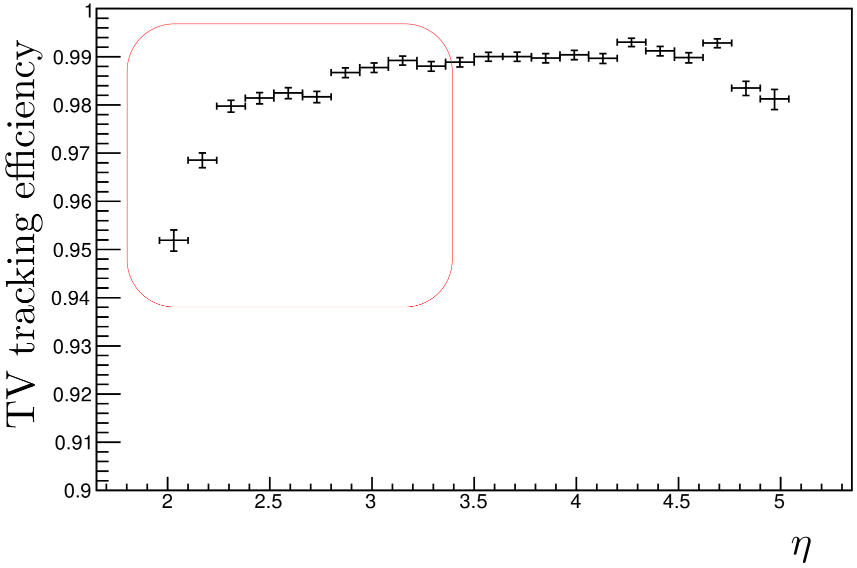
“TV”



Reconstructible checker:



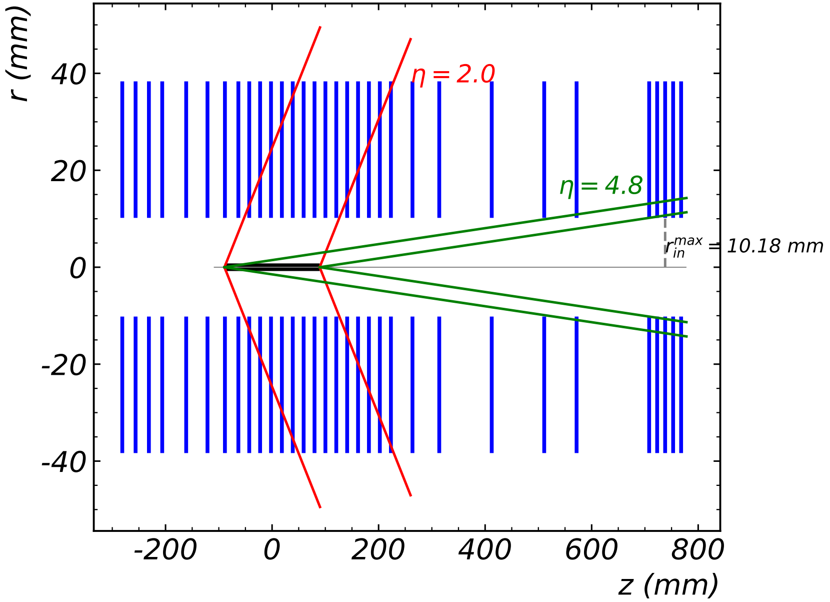
Reconstruction efficiency



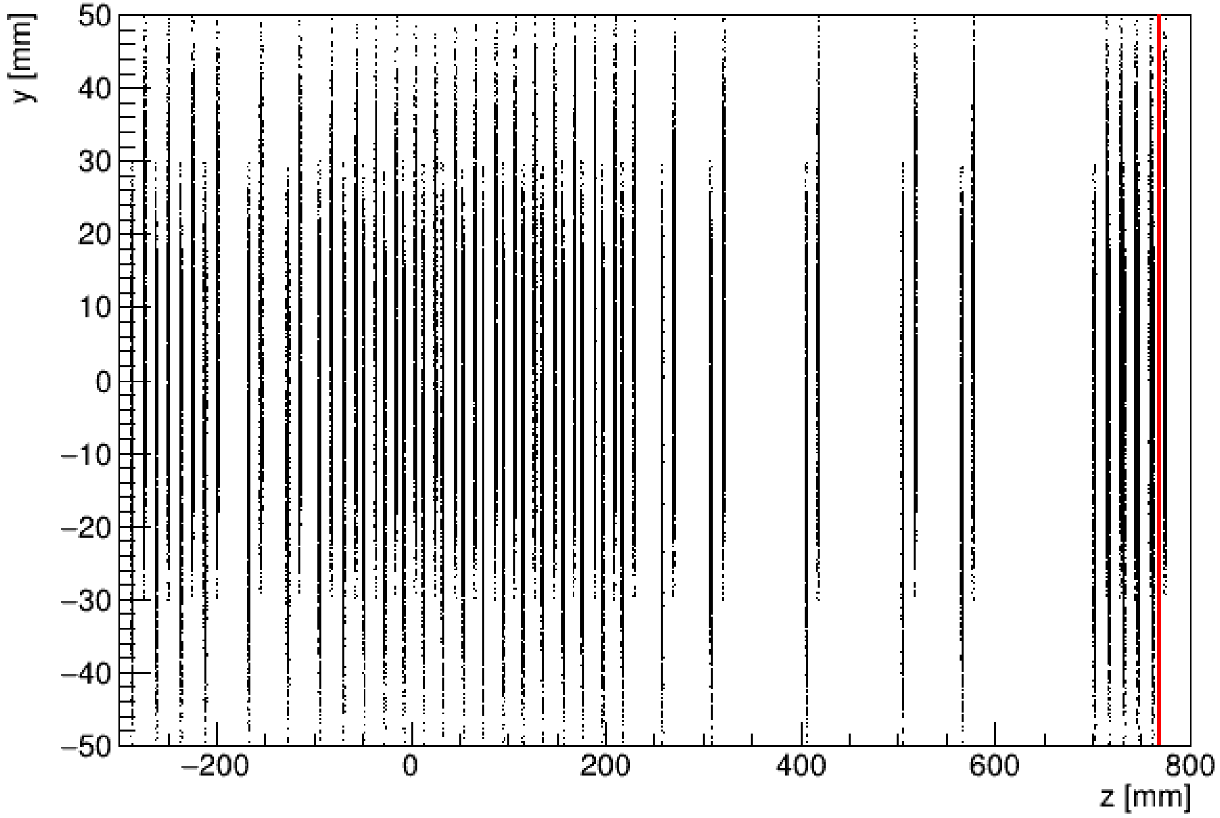
- Moving modules changes η coverage
- Can reoptimising the z-layout recover this?

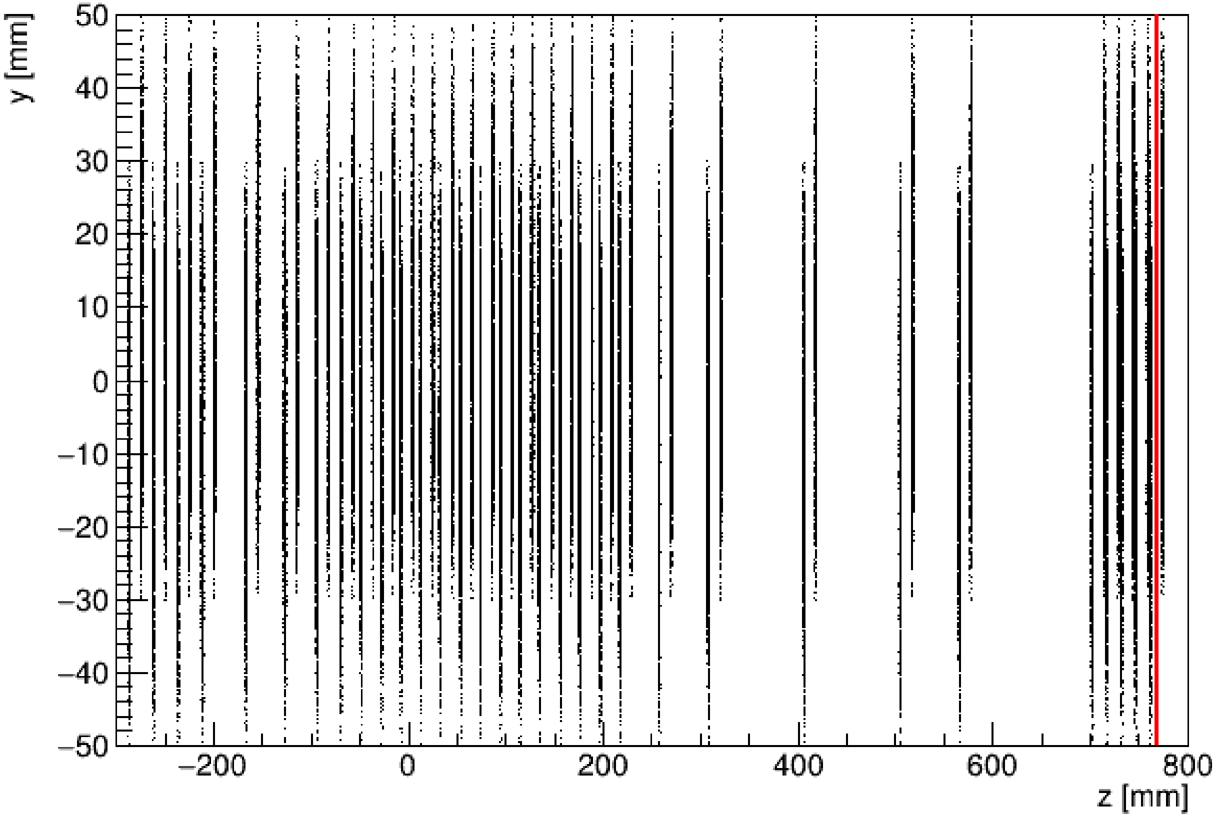
1. Change spacing / add modules to get to the same acceptance
2. Change layout of each module (number of ASICs / pixels per ASIC)
    - Will need to do this anyway once pitch is fixed

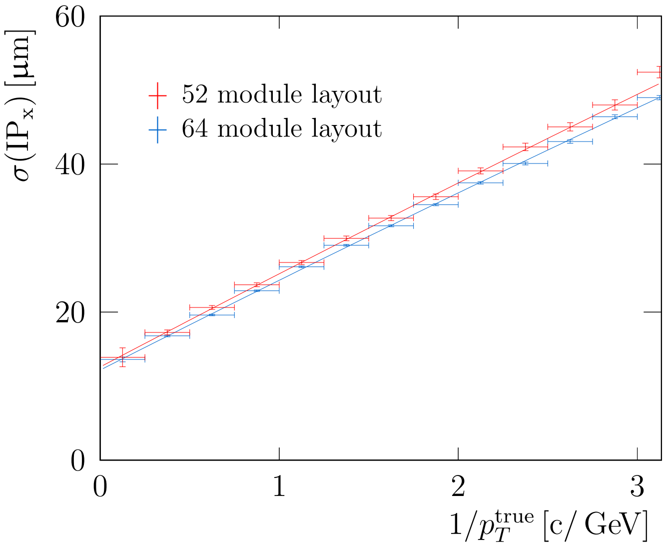
1. Change spacing / add modules to get to the same acceptance
2. Change layout of each module (number of ASICs / pixels per ASIC)
    - Will need to do this anyway once pitch is fixed



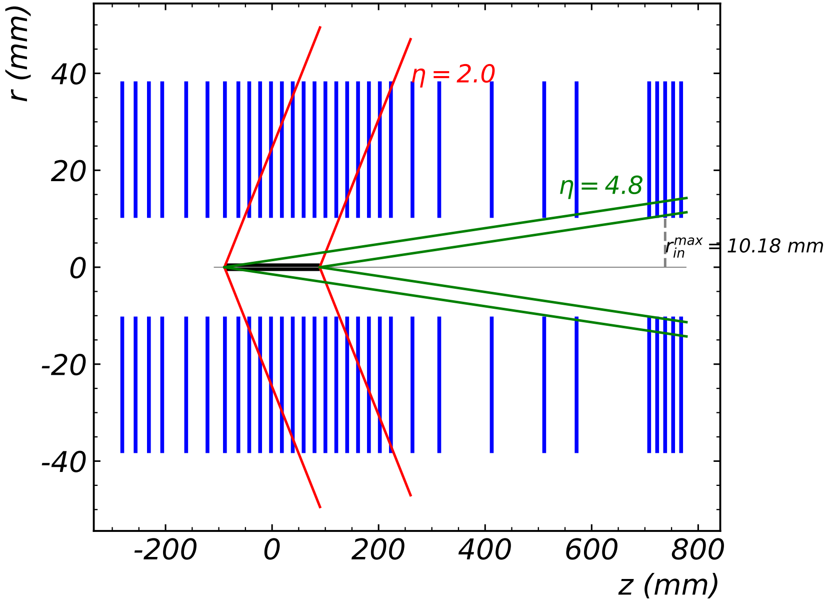
- Baseline layout for scoping
- 32 stations / 64 modules
- Spacing quite tight in some regions
- Optimisation requires >=6 hits / track


y [mm]
50
40
30
20
10
0
−10
−20
−30
−40
−50
−200
0
200
400
600
800
z [mm]


y [mm]
50
40
30
20
10
0
−10
−20
−30
−40
−50
−200
0
200
400
600
800
z [mm]


$\sigma(\mathrm{IP_x})$ [μm]
60
40
20
0
0
1
2
3
52 module layout
64 module layout
$1/p_T^{\mathrm{true}}$ [c/ GeV]

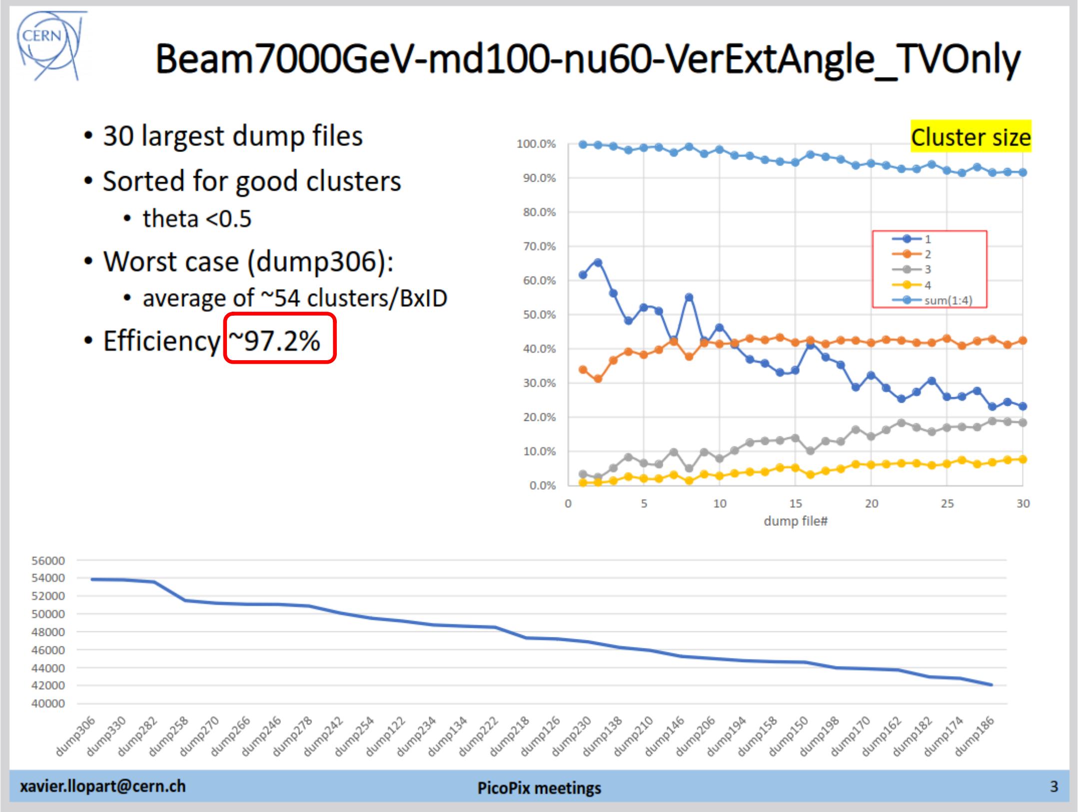
1. Change spacing / add modules to get to the same acceptance
2. Change layout of each module (number of ASICs / pixels per ASIC)
   - Will need to do this anyway once pitch is fixed

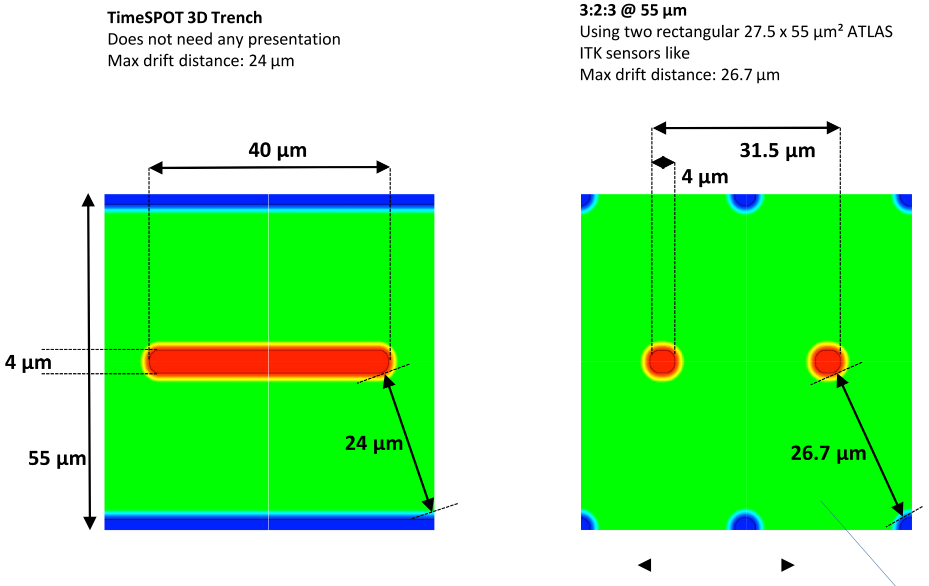


- Baseline layout for scoping
- 32 stations / 64 modules
- Spacing quite tight in some regions
- Optimisation requires >=6 hits / track

Why we design for >= 6 hits

- Significant losses from pileup
  - Improved w.r.t Scenario A (94%)
  - This is with “deadtime” $< 50\,\mathrm{ns}$
  - Curiously, decreasing peak luminosity doesn’t improve this much (97.6% @ 1e34)
- Could be a small interplay with pixel size





- If we want to use 3D sensors, will likely have to learn to live with less than 100% fill factors
  - Tilting is not a preferred solution
  - 0-order can add more layers
  - 1-order consider impact on pattern recognition
  - 2-order impact on resolution

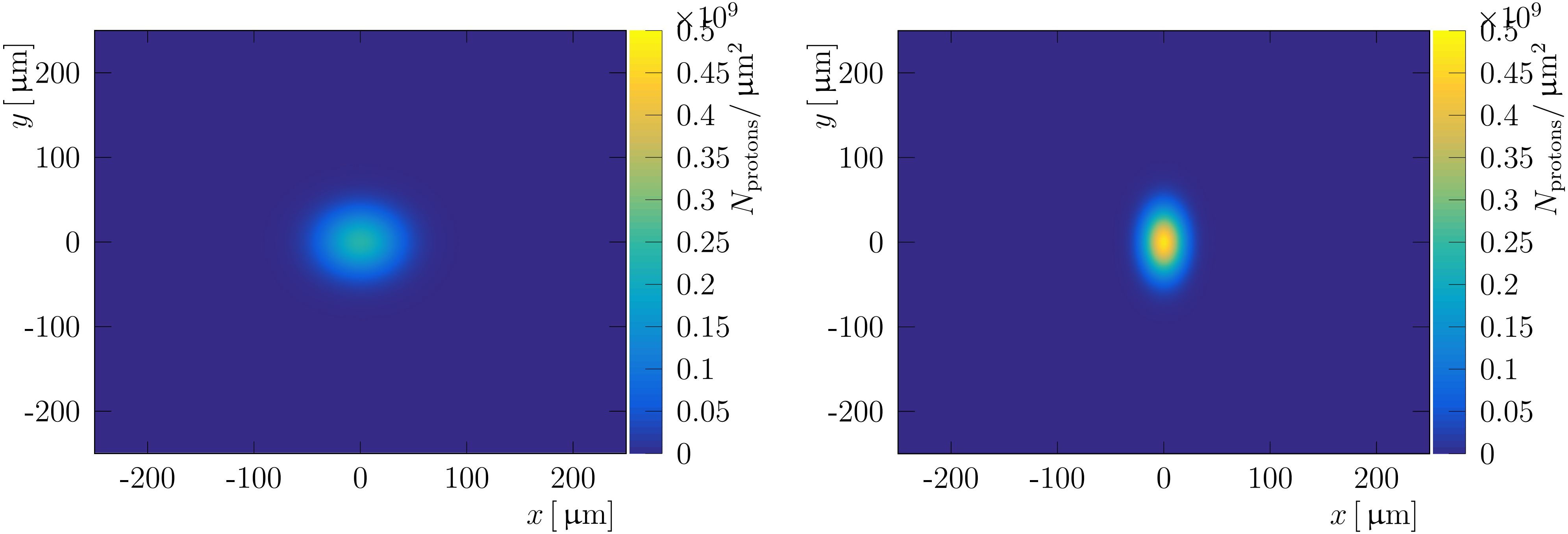
Variability of time resolution with (intra)pixel intercept

Borrowed from a recent meeting

https://indico.cern.ch/event/1382569/contributions/5812580/attachments/2800171/4884944/Tag3.pdf

A quick aside on flat optics *

* by popular demand



$$\sigma_{x,y} = \sqrt{\beta^*_{x,y}\varepsilon/\gamma}$$

Round $\rightarrow$ flat

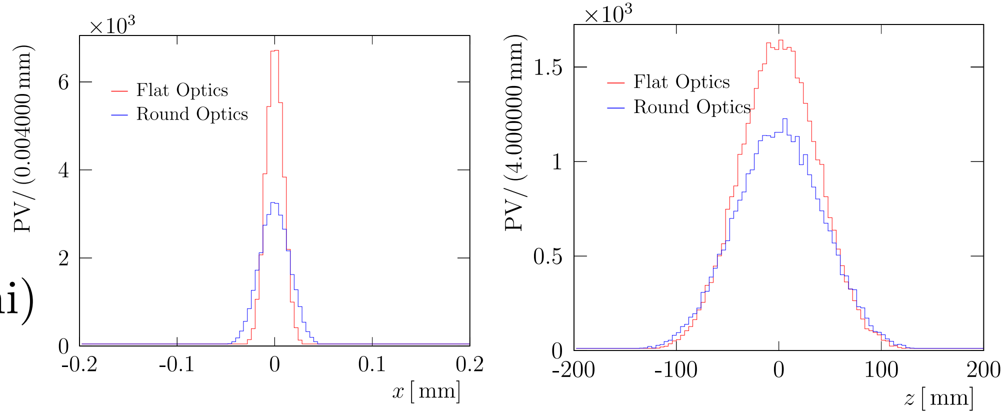
$$\varepsilon \sim 2.5\,\mu\text{m}, \mathcal{O}(\beta^*) \sim 1m, \gamma = E_p/m = 7000$$

Virtual * peak luminosity is proportional to $1/\sigma_{x,y,z}$ ∼ flat optics factor brings a factor of 1.5 in $\mathcal{L}_{virt}$

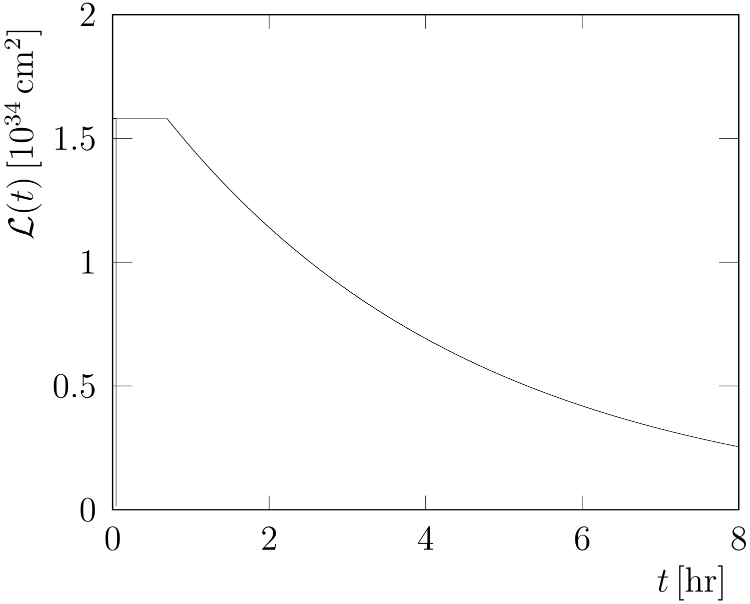
- Biggest effect is on transverse beam distribution but this is not so important
- But it has "second order" effects on longitudinal distribution

$$\sigma_z, \sigma_t \sim \left(\frac{\sigma_{\text{RMS}}}{\sqrt{2}} \left(1 + \sigma^2_{\text{RMS}} \frac{\varepsilon}{\gamma} \left(\theta_v^2 \beta_y + \theta_h^2 \beta_x \right) \right)^{-1/2}, \frac{\sigma_{\text{RMS}}}{\sqrt{2}c} \right)$$

- Our best knowledge is a 15 → 20% compression of luminous region
- Impact on PV reconstruction seems tolerable (particularly with lower peak lumi)

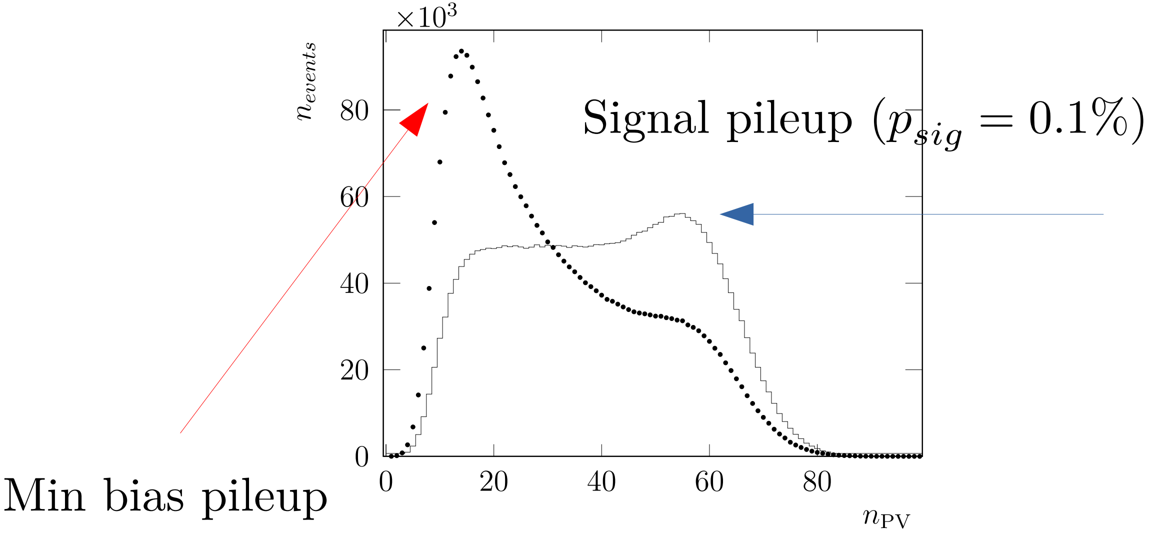


* Virtual luminosity = head-on luminosity at the start of fill



$$\tau_{level} \sim \tau_{beam} \times \log\left(\mathcal{L}_{virt}/\mathcal{L}_{target}\right)$$

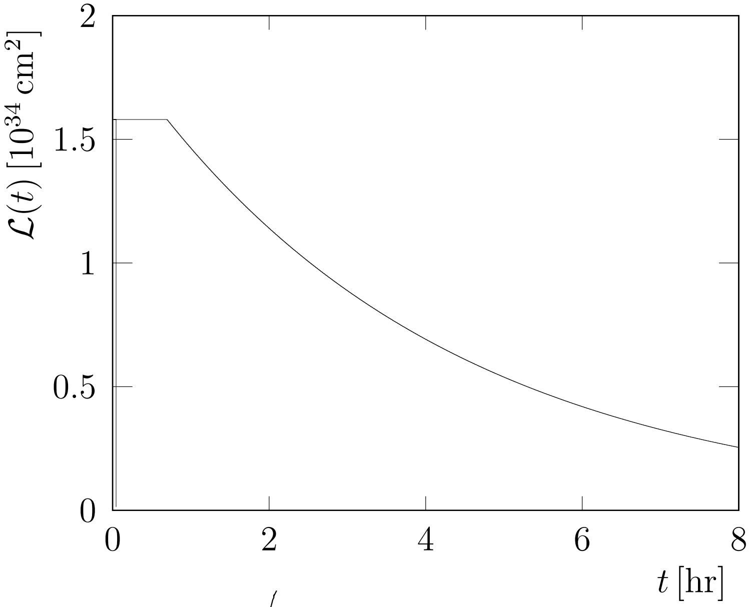
$$\tau_{beam} \sim [4 \to 6] \text{ hours}$$

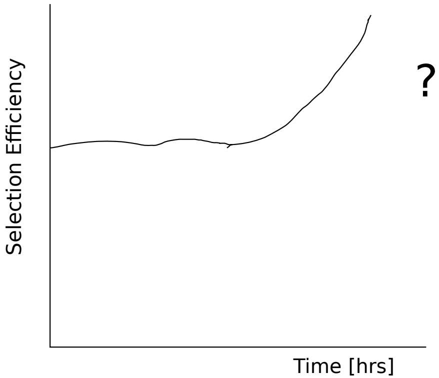


Hang on - are you not a levelled experiment?

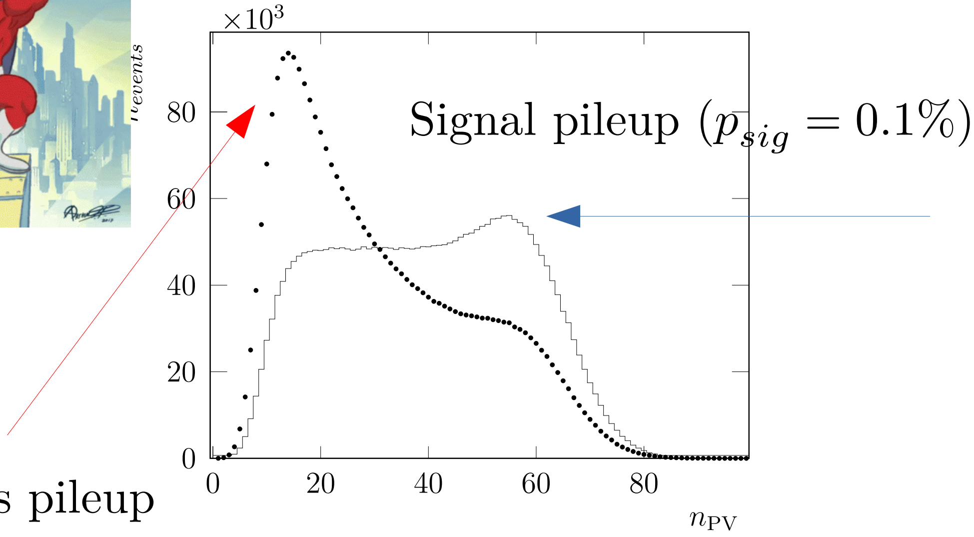
$$\tau_{level} \sim \tau_{beam} \times \log\left(\mathcal{L}_{virt}/\mathcal{L}_{target}\right)$$

$$\tau_{beam} \sim [4 \rightarrow 6] \text{ hours}$$



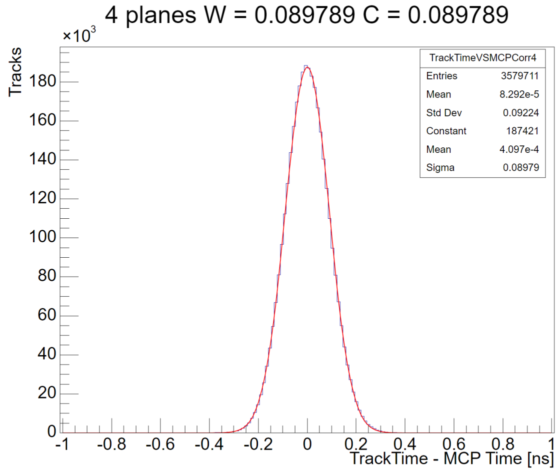


?



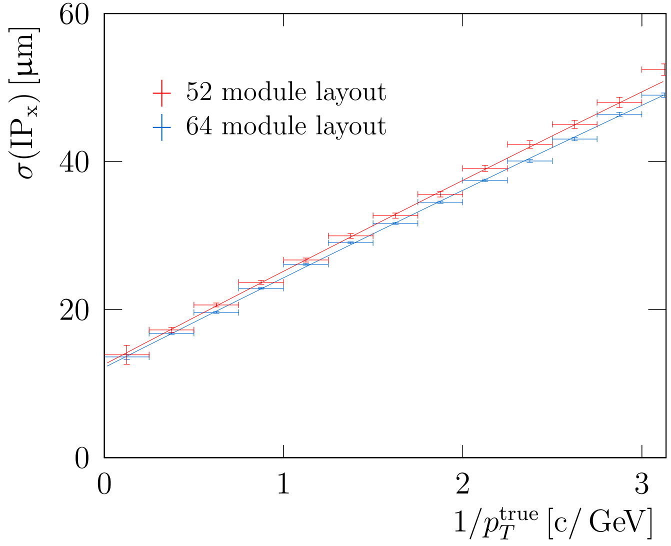
Back to the VELO

- We've replaced two "straw man" scenarios with one that is practical
- There are open questions of implementation for many details
  - Cylindrical foil is a big benefit in most η range but have to make it open and close somehow?
  - Modules need to be very close together - maybe can design the mechanics to attach modules together to make easier to build?
  - Can we achieve space/time resolution requirements?
- BUT we do not find a fundamental showstopper so far..

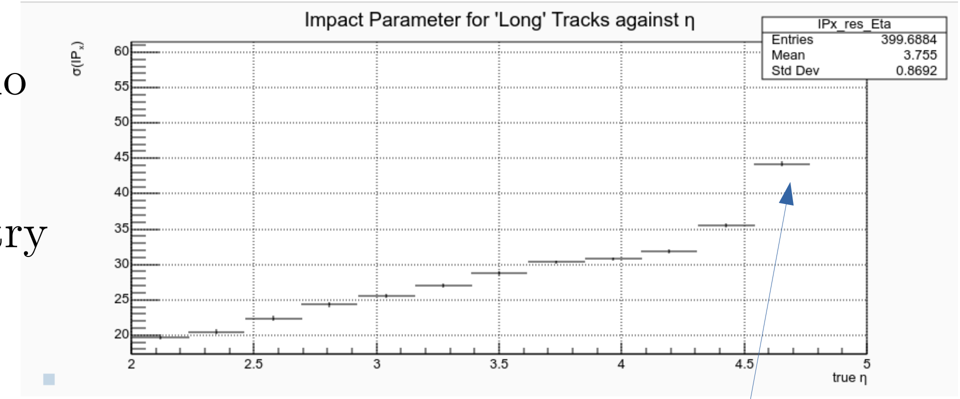


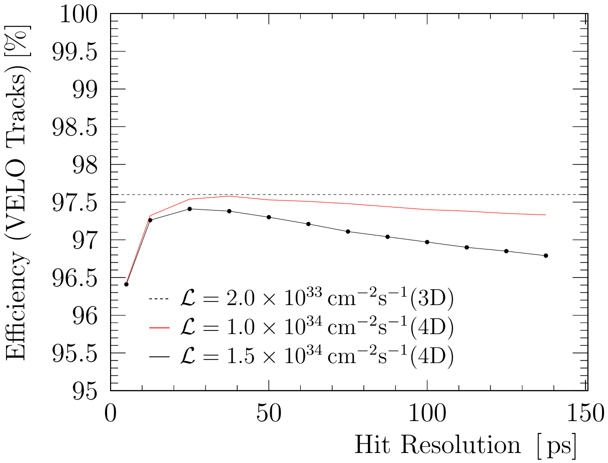
Performance and Reconstruction

- By design, performance is ~ invariant under scenario $\implies$ results from the FTDR era are still valid
- Studies should be repeated now we have the geometry , but first indications are positive

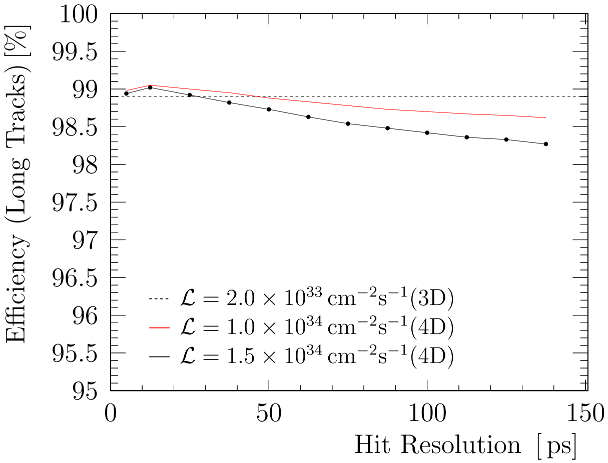


- By design, performance is $\sim$ invariant under scenario $\implies$ results from the FTDR era are still valid
- Studies should be repeated now we have the geometry , but first indications are positive

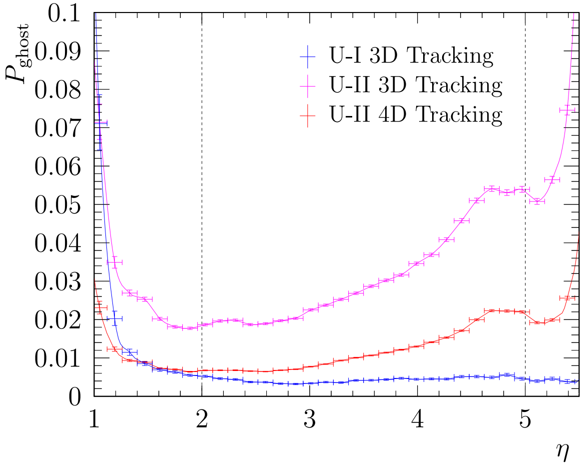




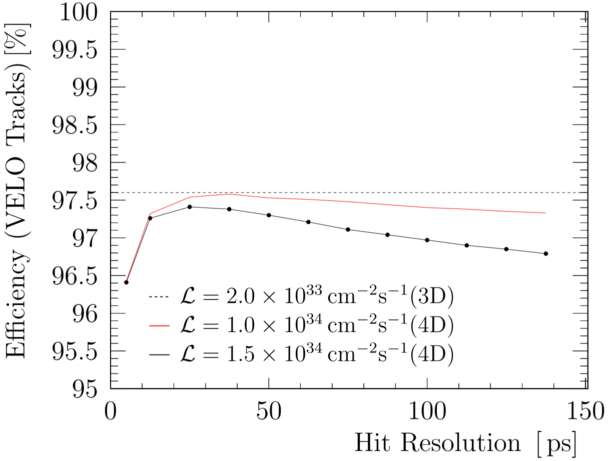
Efficiency (VELO Tracks) [%]
Hit Resolution [ps]
$\mathcal{L} = 2.0 \times 10^{33}\,\mathrm{cm^{-2}s^{-1}}$(3D)
$\mathcal{L} = 1.0 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$(4D)
$\mathcal{L} = 1.5 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$(4D)

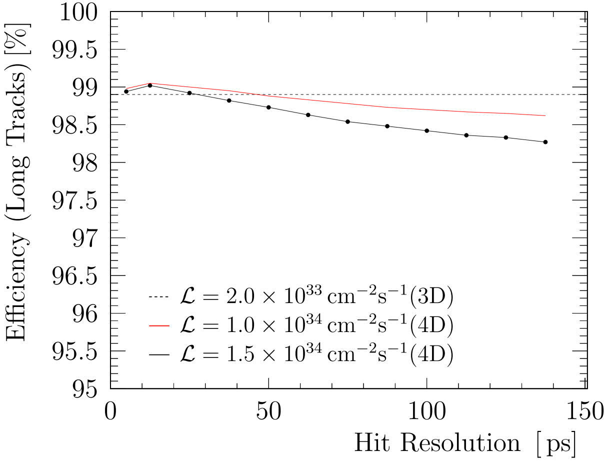


Efficiency (Long Tracks) [%]
Hit Resolution [ps]
$\mathcal{L} = 2.0 \times 10^{33}\,\mathrm{cm^{-2}s^{-1}}$(3D)
$\mathcal{L} = 1.0 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$(4D)
$\mathcal{L} = 1.5 \times 10^{34}\,\mathrm{cm^{-2}s^{-1}}$(4D)

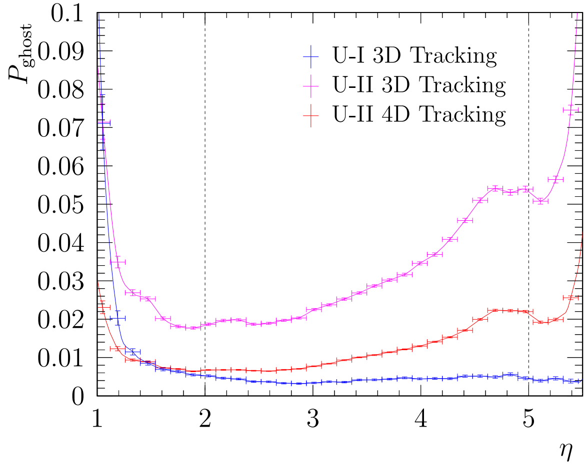


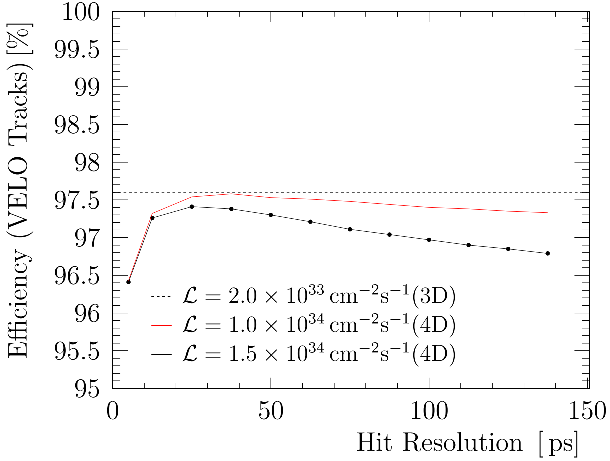
P_{ghost}
η
U-I 3D Tracking
U-II 3D Tracking
U-II 4D Tracking

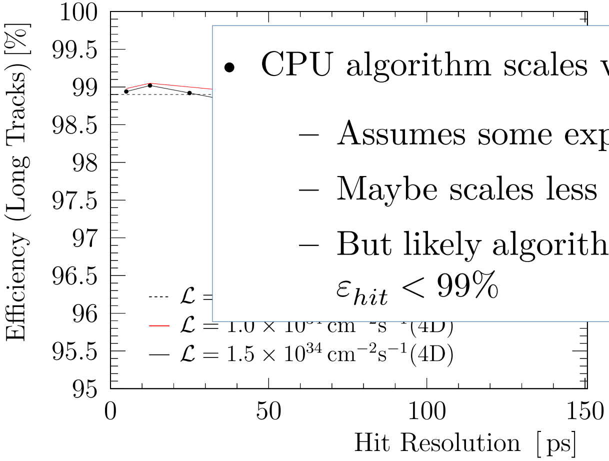




Conditions	n_{tracks}	t/event [μs]	t/track [μs]	$\varepsilon_{\mathrm{velo}}$ [%]	$\varepsilon_{\mathrm{long}}$ [%]	P_{ghost} [%]
algorithm: Upgrade I 3D tracking						
U-I	215	314	1.46	98.1	99.1	0.5
U-II	1690	5780	3.42	95.4	97.3	2.4
U-II (no foil)		5303	3.13	97.1	98.4	2.1
algorithm: Upgrade II 3D re-optimised						
U-I	215	244	1.10	97.6	98.9	0.4
U-II	1690	1792	1.06	95.1	97.0	1.9
U-II (no foil)		1623	0.96	96.7	98.1	1.7

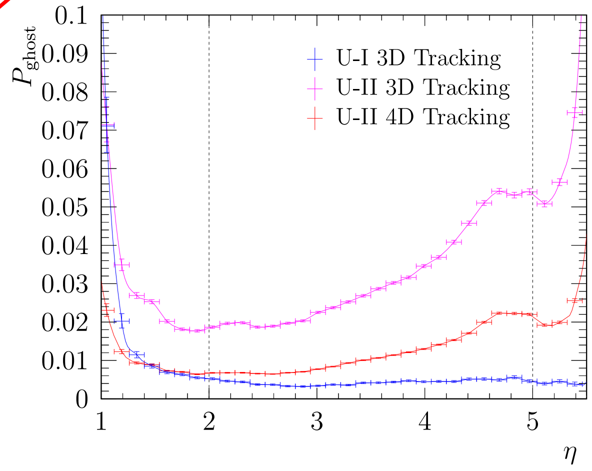


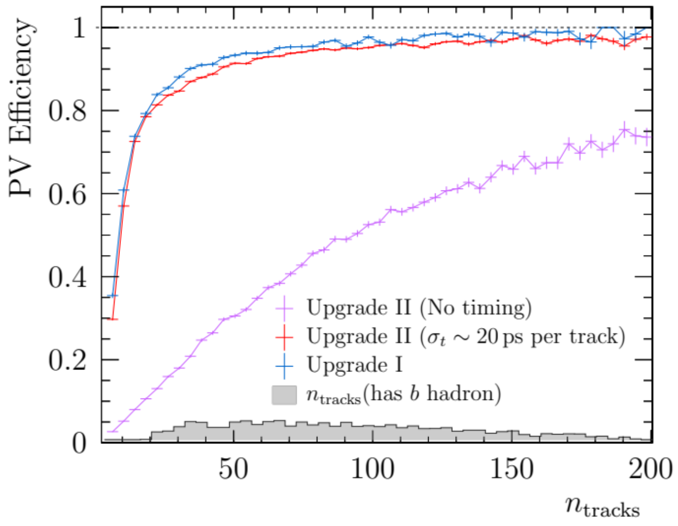


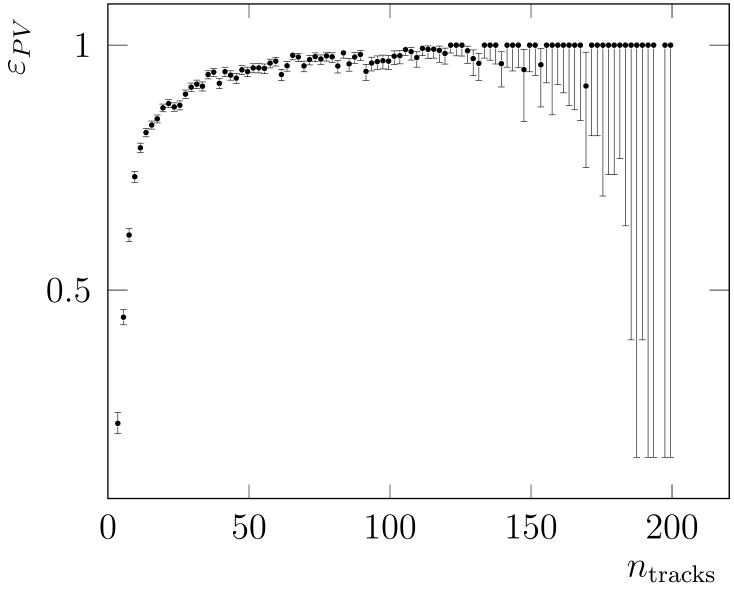


- CPU algorithm scales very well, some assumptions
  - Assumes some expensive preprocessing can be offloaded to FPGA
  - Maybe scales less well on GPUs - but does not seem so critical
  - But likely algorithm will need to be more complicated to compensate $\varepsilon_{hit} < 99\%$

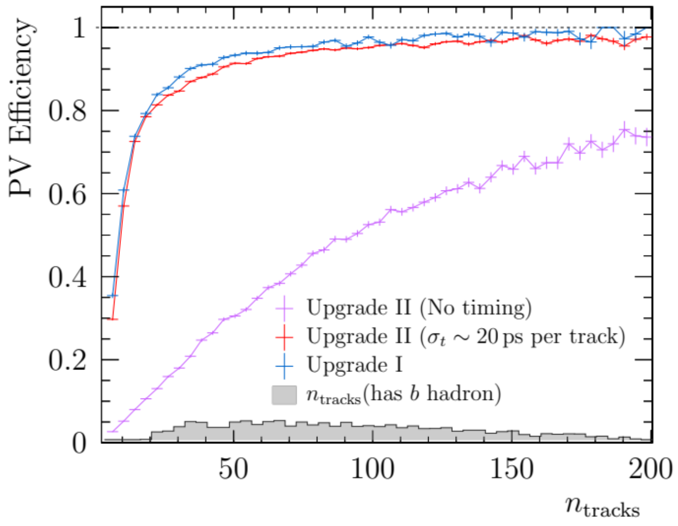
Conditions	n_{tracks}	t/event [µs]	t/track [µs]	$\varepsilon_{\text{velo}}$ [%]	$\varepsilon_{\text{long}}$ [%]	P_{ghost} [%]
algorithm: Upgrade I 3D tracking						
U-I	215	314	1.46	98.1	99.1	0.5
U-II	1690	5780	3.42	95.4	97.3	2.4
U-II (no foil)		5303	3.13	97.1	98.4	2.1
algorithm: Upgrade II 3D re-optimised						
U-I	215	244	1.10	97.6	98.9	0.4
U-II	1690	1792	1.06	95.1	97.0	1.9
U-II (no foil)		1623	0.96	96.7	98.1	1.7

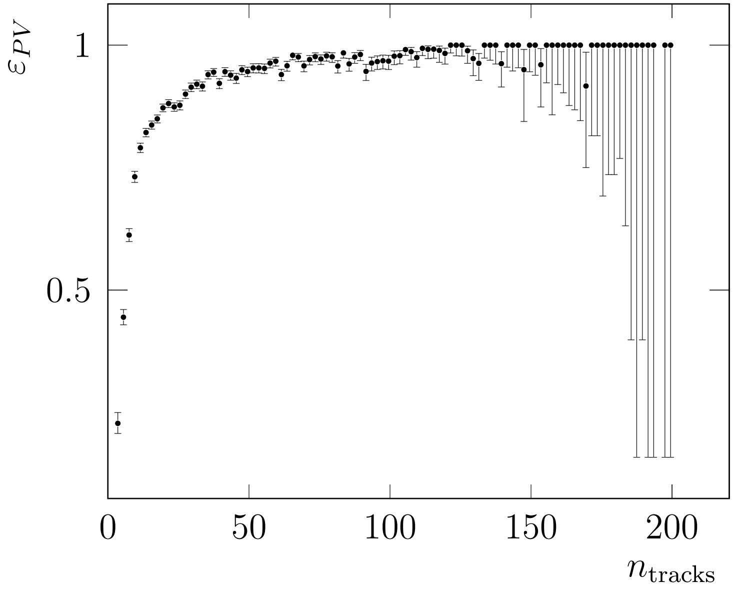






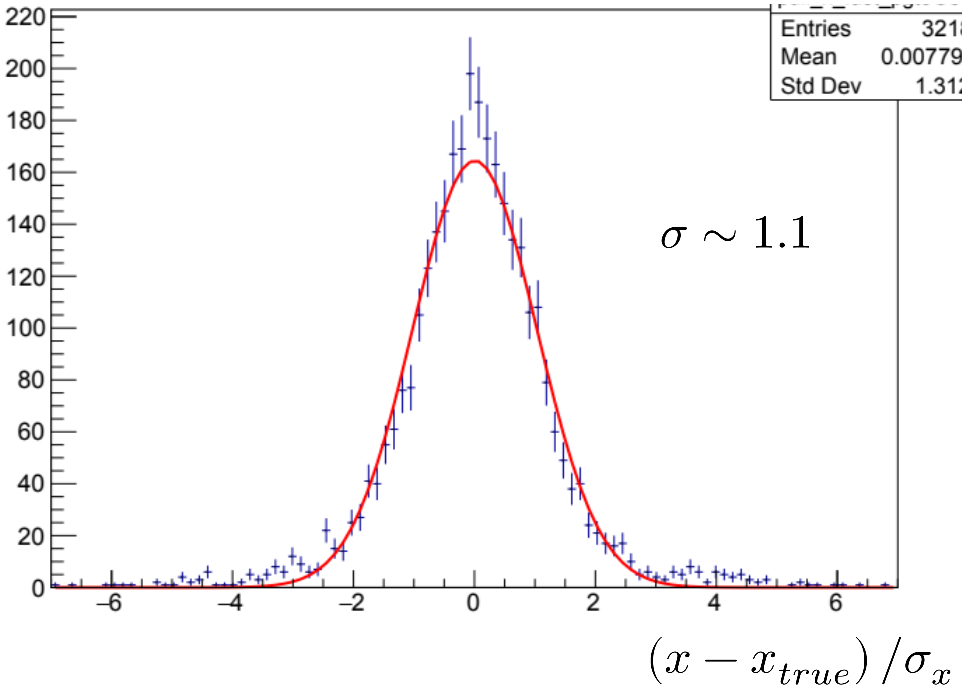
- Scenario X @ 1.5e34, performing slightly better than FTDR
  - $\varepsilon_{PV} \sim 90\%$
  - Aim to squeeze some more performance out

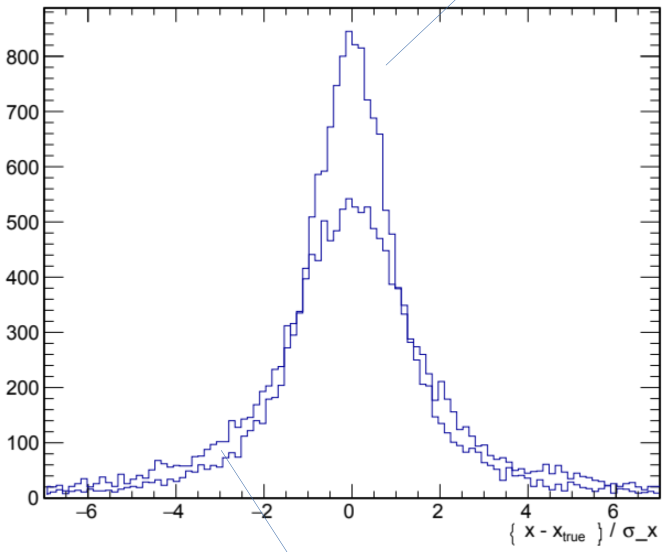




- Scenario X @ 1.5e34, performing slightly better than FTDR
  - $\varepsilon_{PV} \sim 90\%$
  - Aim to squeeze some more performance out

- Huge effort from Andrii & Ben to ensure we can use detailed material description for extrapolation in DD4Hep
- Able to do first tests on this:





- The scenario formerly known as X looks like it will have performance as good as A
  - Still places where algorithms likely need some improvement
  - Can do track fits with detailed geometry $\rightarrow$ very convenient for not misleading ourselves when modifying geometry quickly