Timing and 4D Reconstruction in HEP

ECHEP Workshop

University of Edinburgh

17 Feb 2020

Mark Williams

University of Manchester

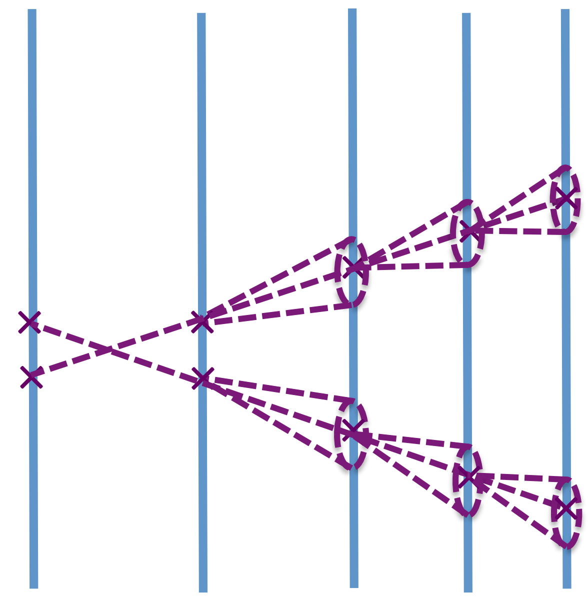
THE ROYAL SOCIETY

MANCHESTER 1824

The University of Manchester

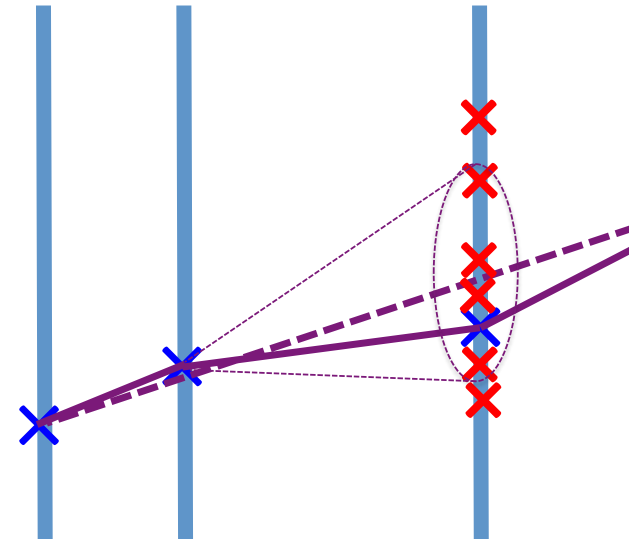
Outline

- Tracking challenges
- Timescales in HEP reconstruction
- State-of-the-art: NA62 GigaTracker
- Ecosystem of fast timing in HL-LHC era
- Timing in HL-LHC Reco – some case studies
- Discussion points



Preamble / disclaimer...

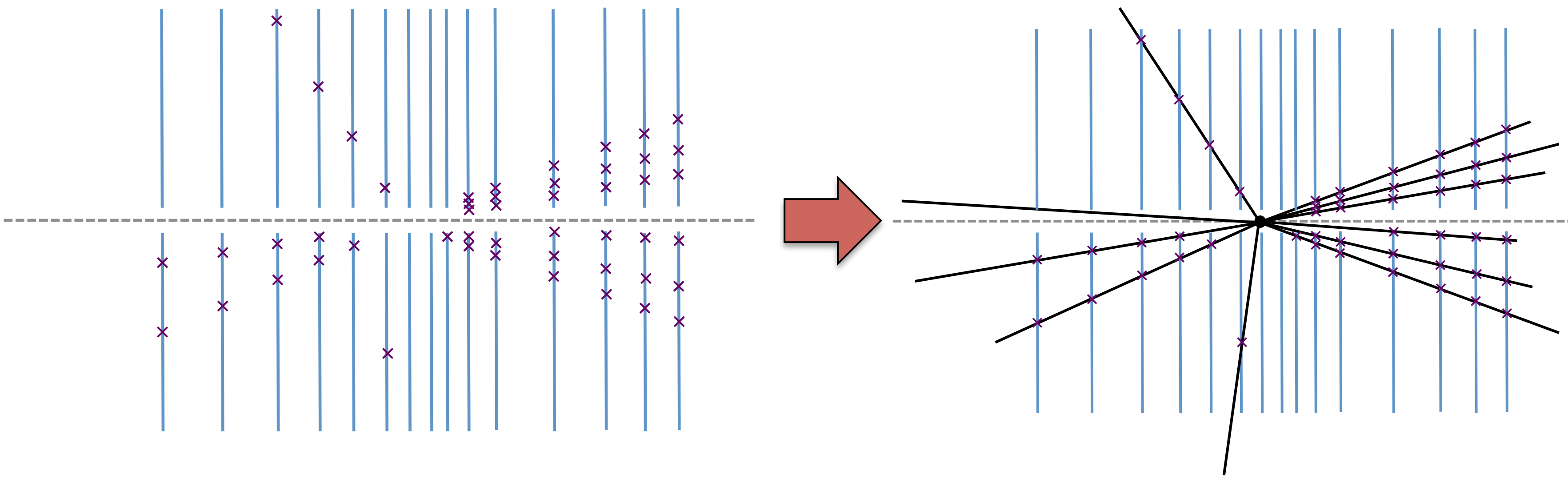
- **Questions, not answers**
- **Aim to provoke discussion**
- **Subjective and selective examples**
- **Some studies in early stages – subject to change**
- **For longer timescale projects (2030 and beyond), I will be quite speculative**



Challenge: particle reconstruction at the HL-LHC

Particle tracking

⇒ **Recover trajectory of particles from set of individual measurements ('hits')**



Set of measurements:
$(x,\sigma_x,y,\sigma_y,z,\sigma_z)$

Set of particle trajectories (=tracks)

Typical hit resolution O(20µm)

- **Straight lines (no B field)**
- **Helices (with B field) − extra free parameter**

Challenge: particle reconstruction at the HL-LHC

Particle tracking

⇒ **Recover trajectory of particles from set of individual measurements ('hits')**

Two main phases:

(1) **Pattern recognition** ⇒ identify + group hits from individual particles

Typically O(10) hits per track; O(1000) tracks per event

Challenge: particle reconstruction at the HL-LHC

Particle tracking

⇒ Recover trajectory of particles from set of individual measurements ('hits')

Two main phases:

(1) Pattern recognition ⇒ identify + group hits from individual particles

(2) **Track fitting** ⇒ model trajectory of particle through detector, accounting for scattering and finite hit resolution (and detector imperfections, e.g. misalignment)

Typical to use Kalman Filter to perform this step
− recursive estimator including noise and correlations
− computationally expensive. e.g. requires inverting covariance matrices

Challenge: particle reconstruction at the HL-LHC

Particle tracking

⇒ **Recover trajectory of particles from set of individual measurements ('hits')**

Two main phases:

(1) Pattern recognition ⇒ identify + group hits from individual particles

(2) **Track fitting** ⇒ model trajectory of particle through detector, accounting for scattering and finite hit resolution (and detector imperfections, e.g. misalignment)

Typical to use Kalman Filter to perform this step
− recursive estimator including noise and correlations
− computationally expensive. e.g. requires inverting covariance matrices

Finally, apply **track quality filter** (e.g. remove tracks with many shared hits)
⇒ Can be resource-intensive to resolve ambiguities

Challenge: particle reconstruction at the HL-LHC

Particle tracking

⇒ Recover trajectory of particles from set of individual measurements (‘hits’)

Two main phases:

(1) Pattern recognition ⇒ identify + group hits from individual particles

(2) **Track fitting** ⇒ model trajectory of particle through detector, accounting for scattering and finite hit resolution (and detector imperfections, e.g. misalignment)

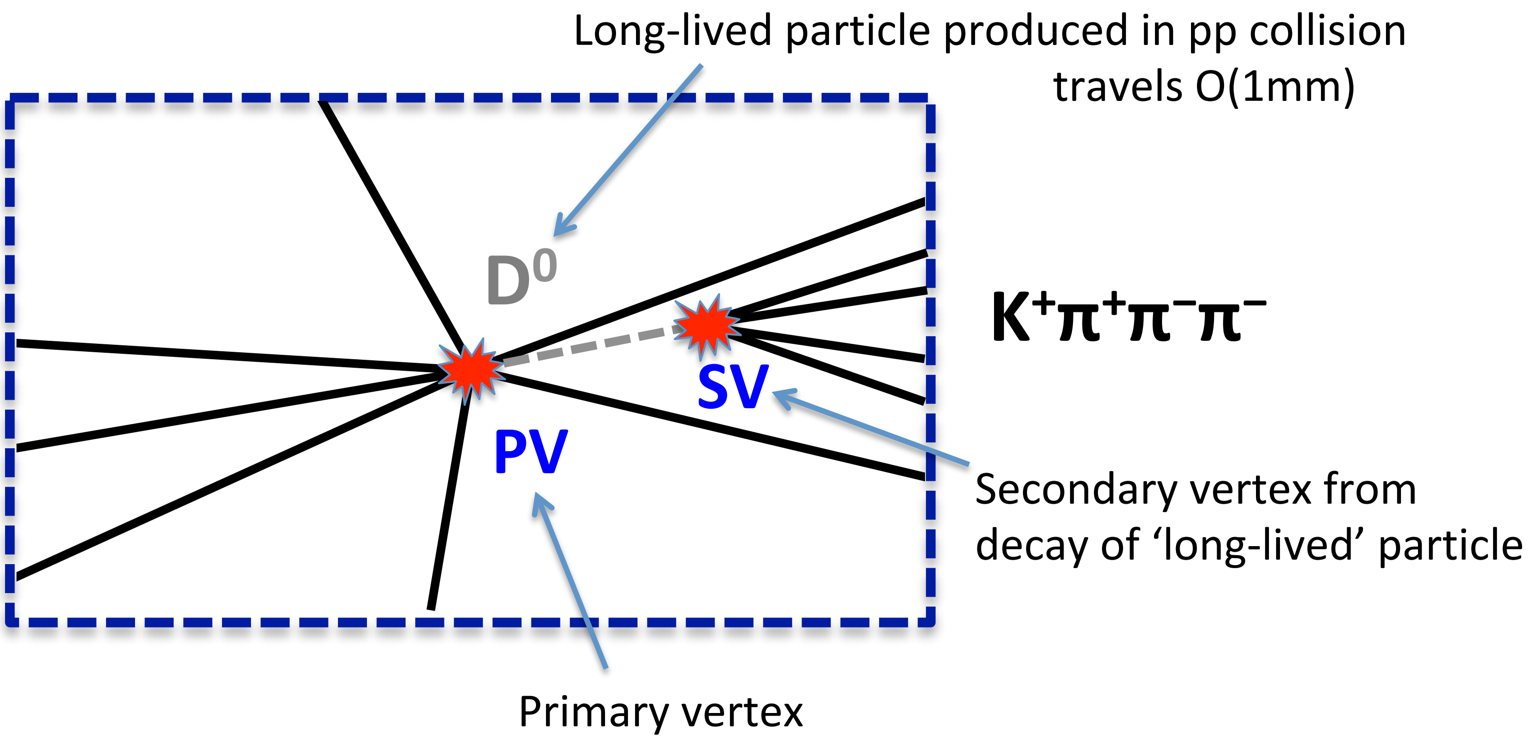
Typical to use Kalman Filter to perform this step
− recursive estimator including noise and correlations
− computationally expensive. e.g. requires inverting covariance matrices

Higher intensity ⇒ Many more possible combinations to check ⇒ Computationally expensive

Challenge: particle reconstruction at the HL-LHC

After tracking, need to **reconstruct primary and secondary vertices**

i.e. points of origin of particles



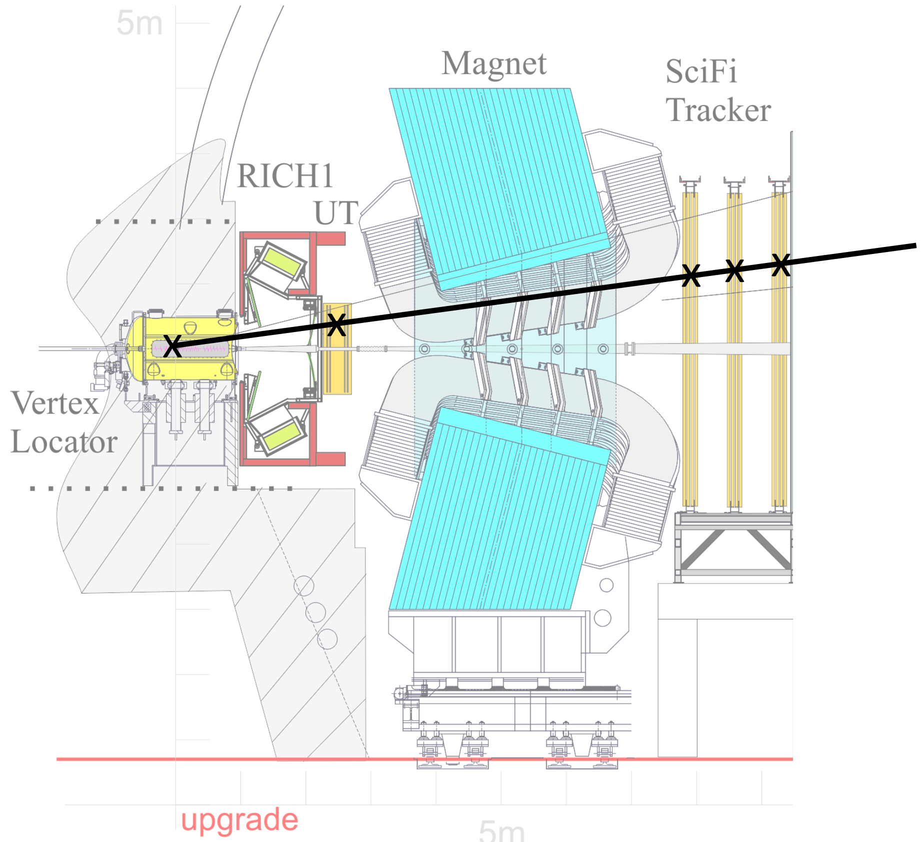
In HL-LHC, expect ~200 individual PVs in a typical event, with spread $\sigma_z \approx 50$mm

Challenge: particle reconstruction at the HL-LHC

Crucial to match tracks / signals between sub-detectors

- Long paths between measurements
- Material scattering
- Magnetic deflection

⇒ Very challenging at high detector occupancy



How fast is fast timing?

Time between LHC bunch crossings: **25ns** ⇒ Dictates precision of current generation of ‘fast’ detectors

Reconstruction timeframes within single event:

σ_t(PV) ≈ 200ps → Need **σ(t) < 200ps** to separate tracks from different PVs

How fast is fast timing?

Time between LHC bunch crossings: **25ns** ⇒ Dictates precision of current generation of 'fast' detectors

Reconstruction timeframes within single event:

$\sigma_t(PV) \approx 200ps$ ⟶ Need **σ(t) < 200ps** to separate tracks from different PVs

@v=c 100mm → **330ps**
10m → **33ns** ⟶ Typical times to traverse (sub)detectors

How fast is fast timing?

Time between LHC bunch crossings: **25ns** ⇒ Dictates precision of current generation of 'fast' detectors

Reconstruction timeframes within single event:

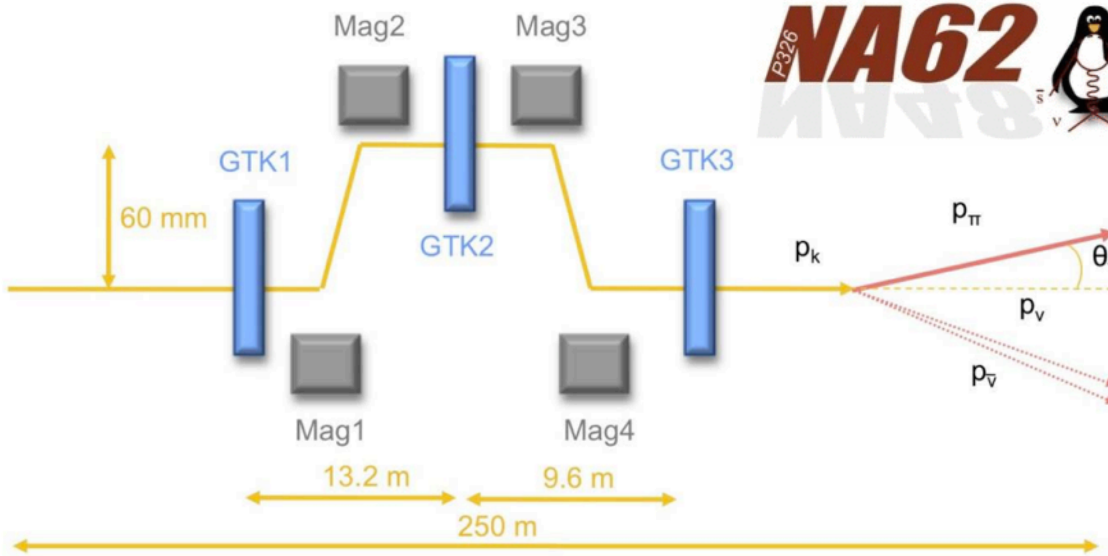
$\sigma_t(PV) \approx$ 200ps → Need **σ(t) < 200ps** to separate tracks from different PVs

@v=c 100mm → **330ps**
10m → **33ns**
→ Typical times to traverse (sub)detectors

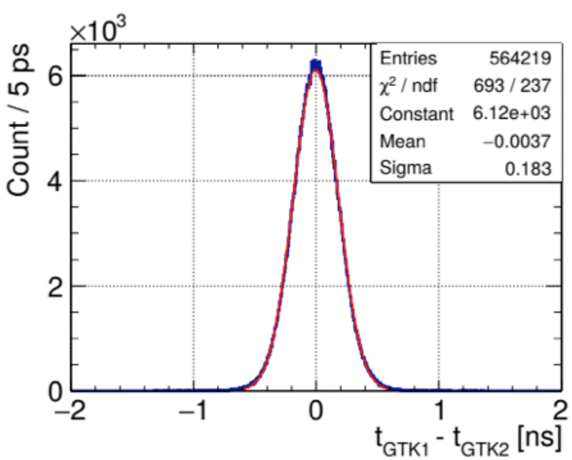
@p=10GeV, 10m flight:
$\Delta t(\pi\text{-}K)$ = **150ps**
$\Delta t(\pi\text{-}p)$ = **580ps**
$\Delta t(K\text{-}p)$ = **430ps**
→ For PID from time-of-flight, need similar precision, **O(50ps)** per object

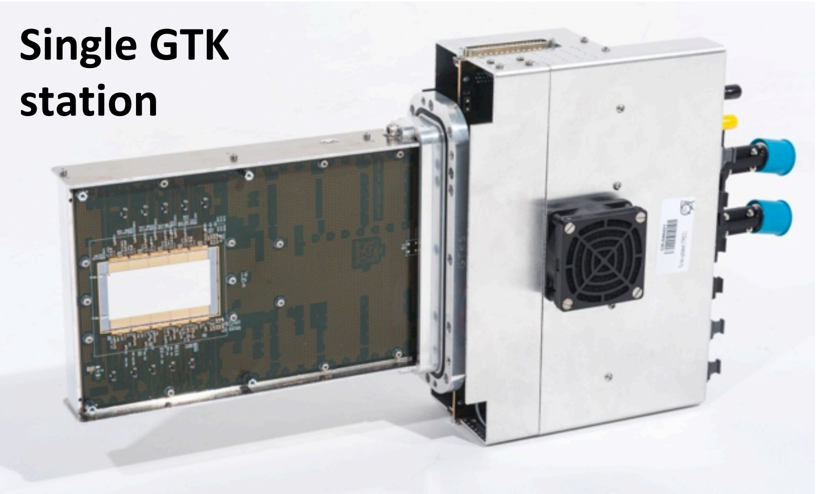
State-of-the-art: NA62 GigaTracker

- Small area (0.002 m^2)
- (300 μm)^2 pixels
- 0.75 GTracks/s
  (0.75 particles/ns)
- 75ns signal window



σ_t(hit) ≈ 130ps per station (@100V bias)





https://arxiv.org/abs/1904.12837

State-of-the-art: NA62 GigaTracker

Precise timing information essential:

(1) Track pattern recognition:

- In signal time window expect ~50 particles through detector.
- Only 3 stations: little geometrical redundancy
  ⇒ PR relies almost entirely on time information
- With σ_t(hit) = 200ps, can separate consecutive particles by >5σ

State-of-the-art: NA62 GigaTracker

Precise timing information essential:

(1) Track pattern recognition:

- In signal time window expect ~50 particles through detector.
- Only 3 stations: little geometrical redundancy
  ⇒ PR relies almost entirely on time information
- With σ_t(hit) = 200ps, can separate consecutive particles by >5σ

(2) Matching particles to downstream detectors

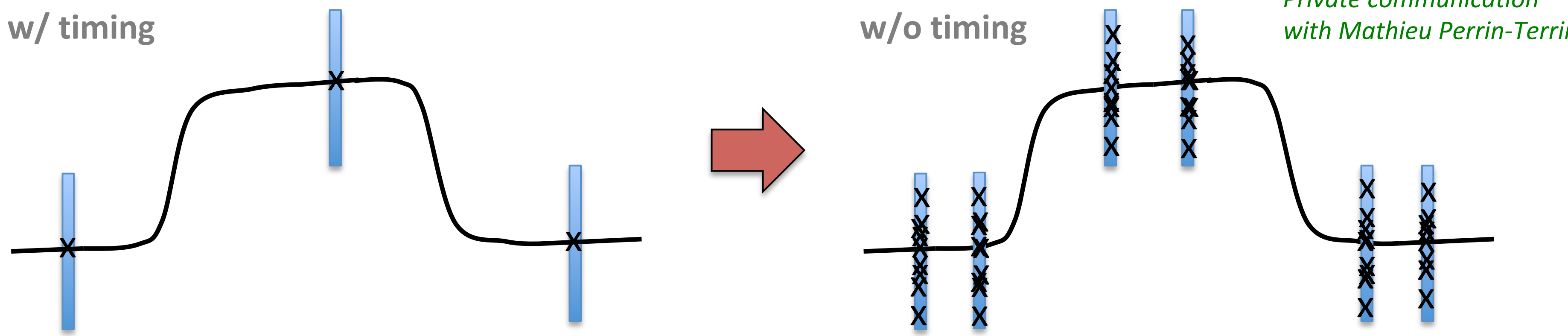
- ≥2 hits: σ_t(track) <100ps
- Unambiguous association of GTK track with RICH detector, 150m downstream (90ps time resolution)

⇒ Essentially no combinatorics in reconstruction or matching. Fast, efficient, and clean.

A GigaTracker without timing?

Thought experiment: *What if no timing information within 75ns window?*

Private communication with Mathieu Perrin-Terrin



No ambiguity ⇒ need just 3 stations to measure momentum and direction

~50 particles through GTK ⇒ need additional stations to determine track properties

More stations ⇒ more scattering ⇒ more background

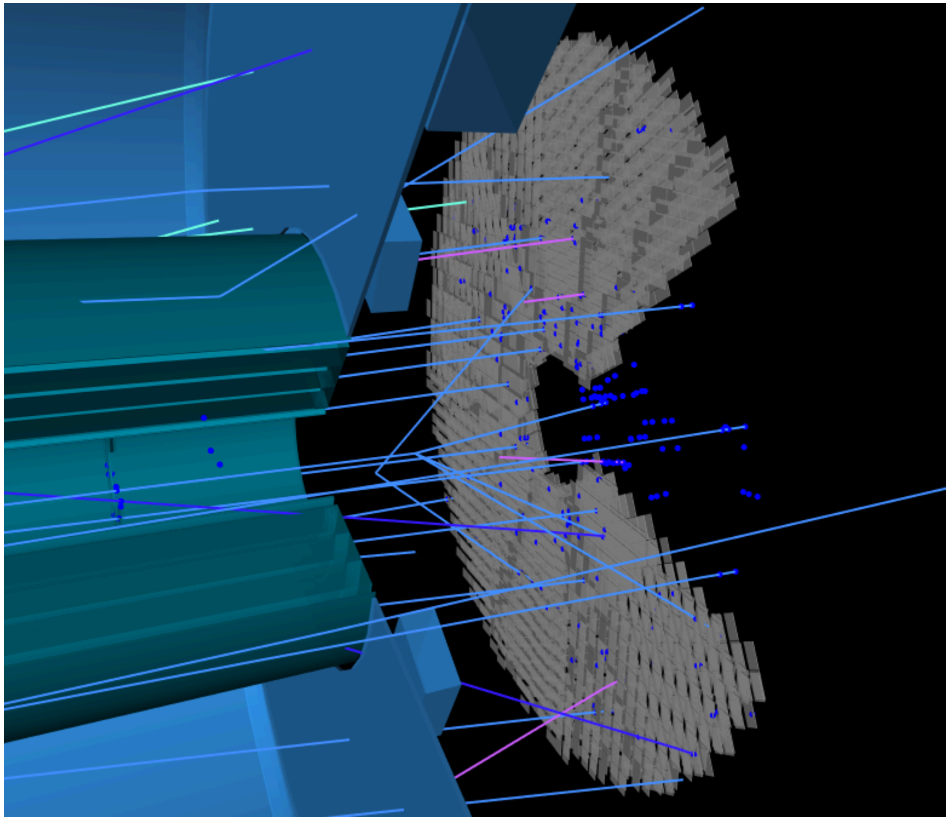
+ Mis-association with downstream detectors would explode

⇒ More complex detectors, more complex analysis, more resource-hungry experiment

Timing at the HL-LHC: Atlas HGTD

The Atlas High Granularity Timing Detector

When?
To be installed in LS3 (~2025)
Why?
Provide 30ps timestamp on tracks in forward region, to mitigate effects of high pileup



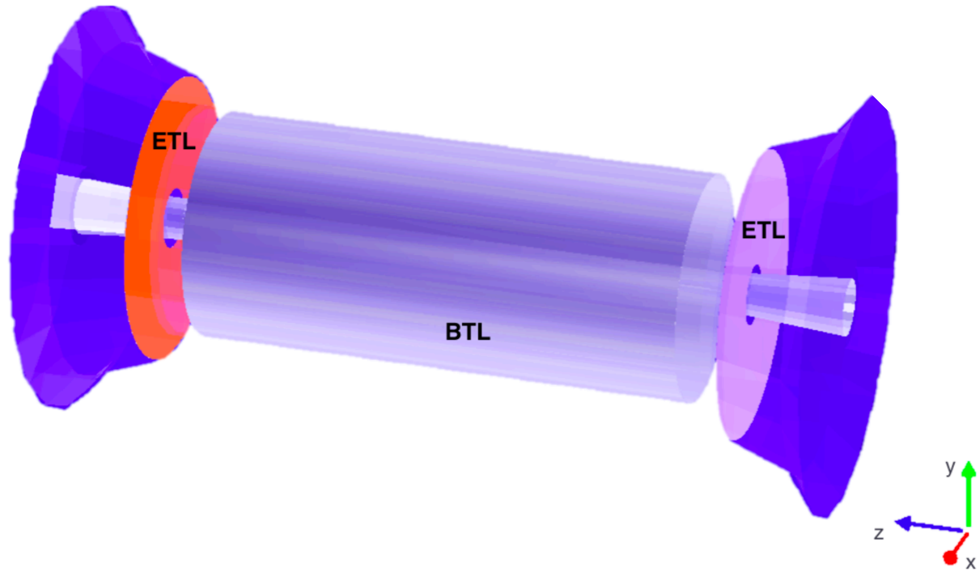
In central (barrel) region, spatial information sufficient to resolve PVs

- $2.4 < |\eta| < 4.0$
- $(1.3\text{mm})^2$ pad LGAD sensors
  ⇒ occupancies < 10%
- 2-3 hits per track

Timing at the HL-LHC: CMS MTD

The CMS MIP Timing Detector

When?
To be installed in LS3 (~2025)
Why?
Provide 35ps timestamp on tracks to mitigate effects of high pileup



Both forward (ETL)
and central (BTL) detectors

+ Proposal to include in **L1 Trigger & HLT**
(waiting for CMS Trigger TDR in Q1 2020…)

Barrel Timing Layer:
- $|\eta| < 1.5$
- LYSO Scintillators
- Cells 6 cm in ϕ; 3mm in z

Endcap Timing Layer:
- $1.6 < |\eta| < 3.0$
- 2 layers of LGAD pads $(1.3\text{mm})^2$

Timing at the HL-LHC: CMS Calorimeters

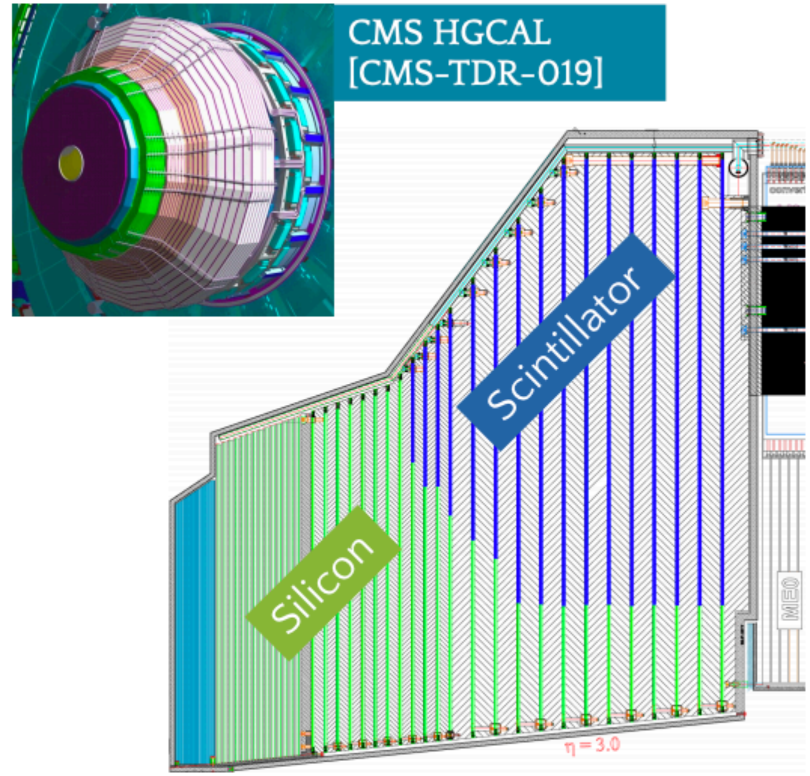
HGCAL (forward) + ECAL Barrel

When?

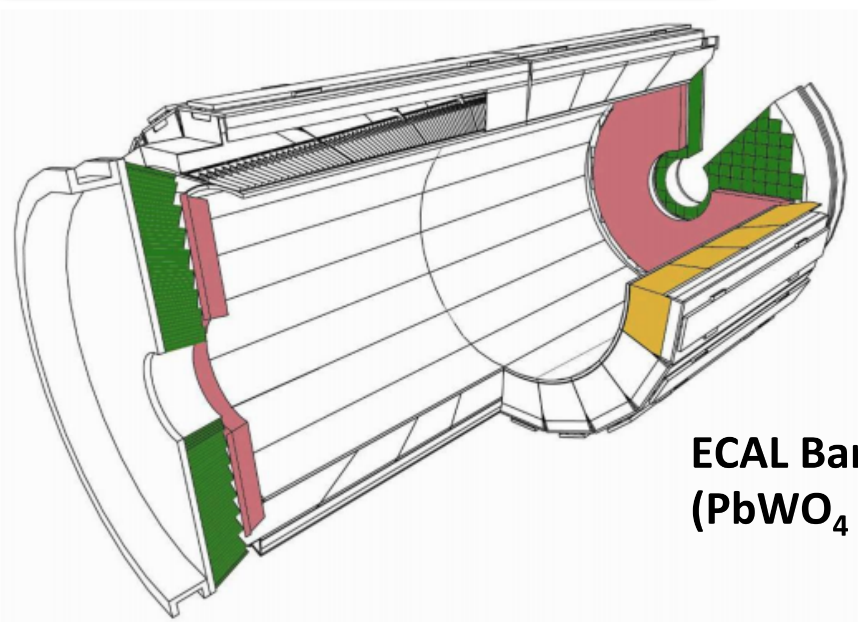
To be installed in LS3 (~2025)

Why?

Provide single-cell resolution of 20-150ps for EM and jet reconstruction and PV matching



HGCAL (silicon + scintillator)



ECAL Barrel ($PbWO_4$ crystals)

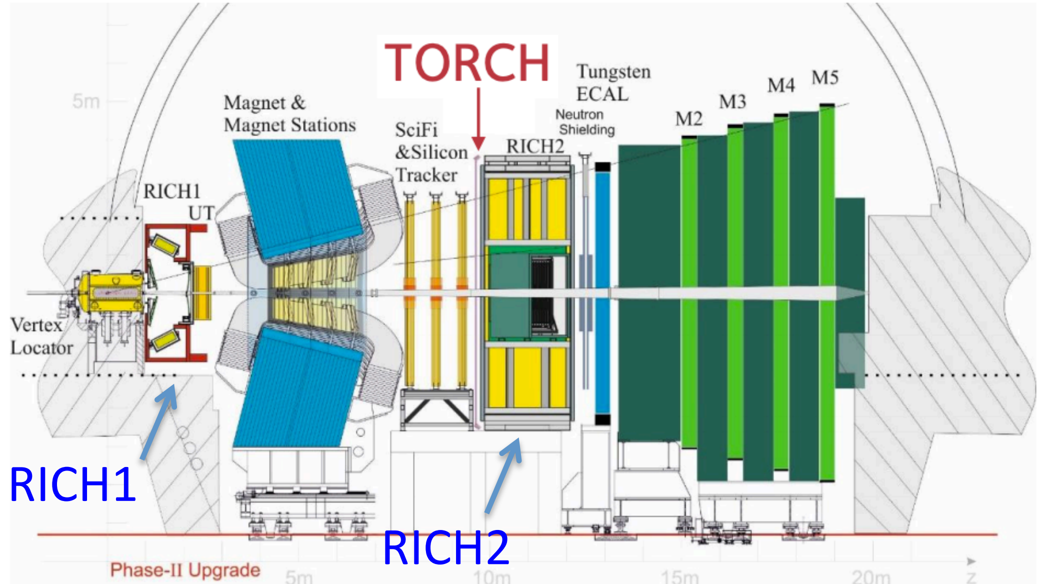
Timing at the HL-LHC: LHCb TORCH + RICH

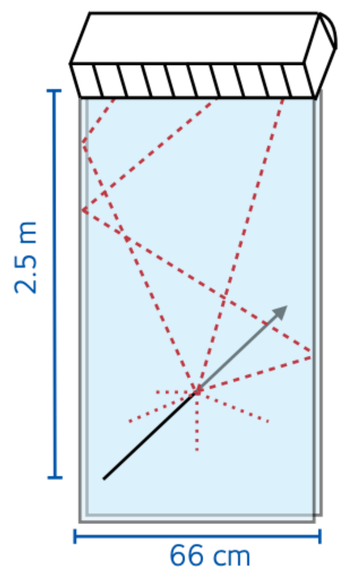
Cherenkov radiation detectors for PID

TORCH (quartz + MCPs)

When?
To be installed in LS3 (~2025)

Why?
ToF for particle ID (π/K/p)
⇒ 15ps per track

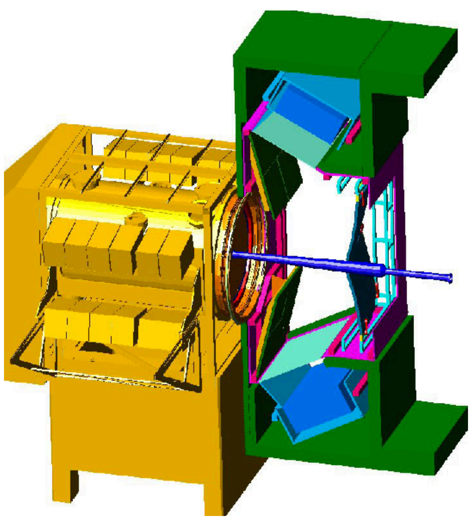




Timing RICH (gas + timing PD)

When?
TBA – LS2/3/4

Why?
Associate RICH objects to tracks
⇒ O(50ps) per track



Timing at the HL-LHC: LHCb Timing VELO

Silicon pixel detector with per-hit timing

When?

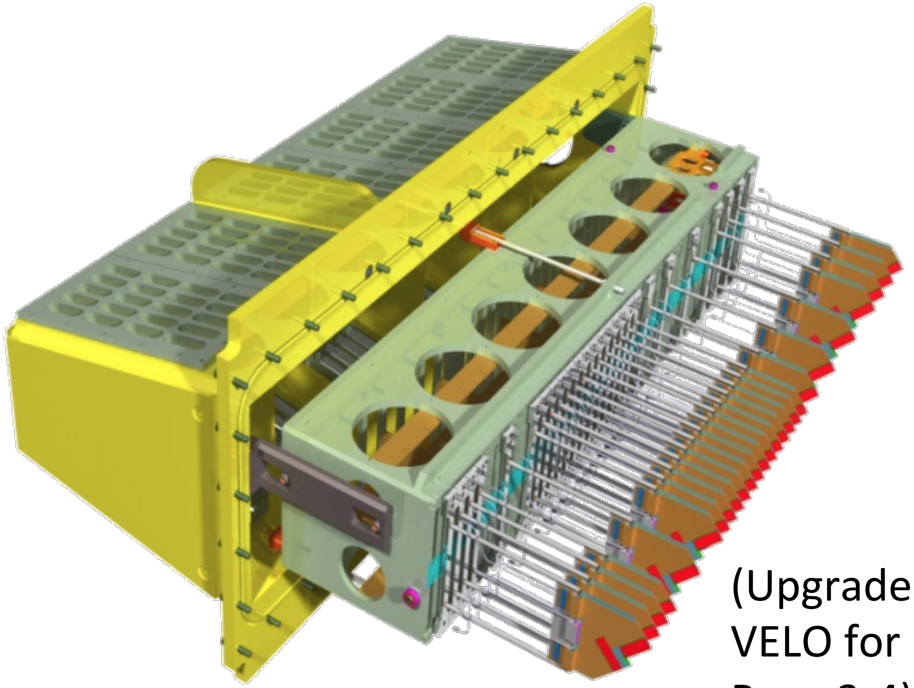
To be installed in LS4 (~2030)

Why?

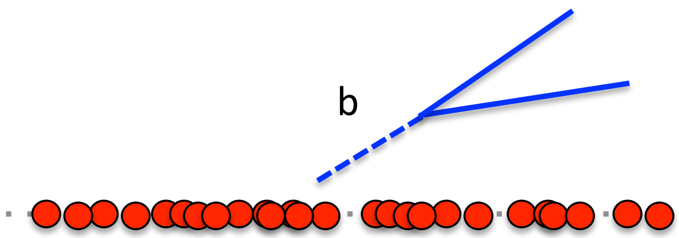
~50ps, 50µm vertex detector for full 4D reconstruction in trigger (=offline)

Early stages: design, precision, technologies not yet clear

All information used in trigger decision

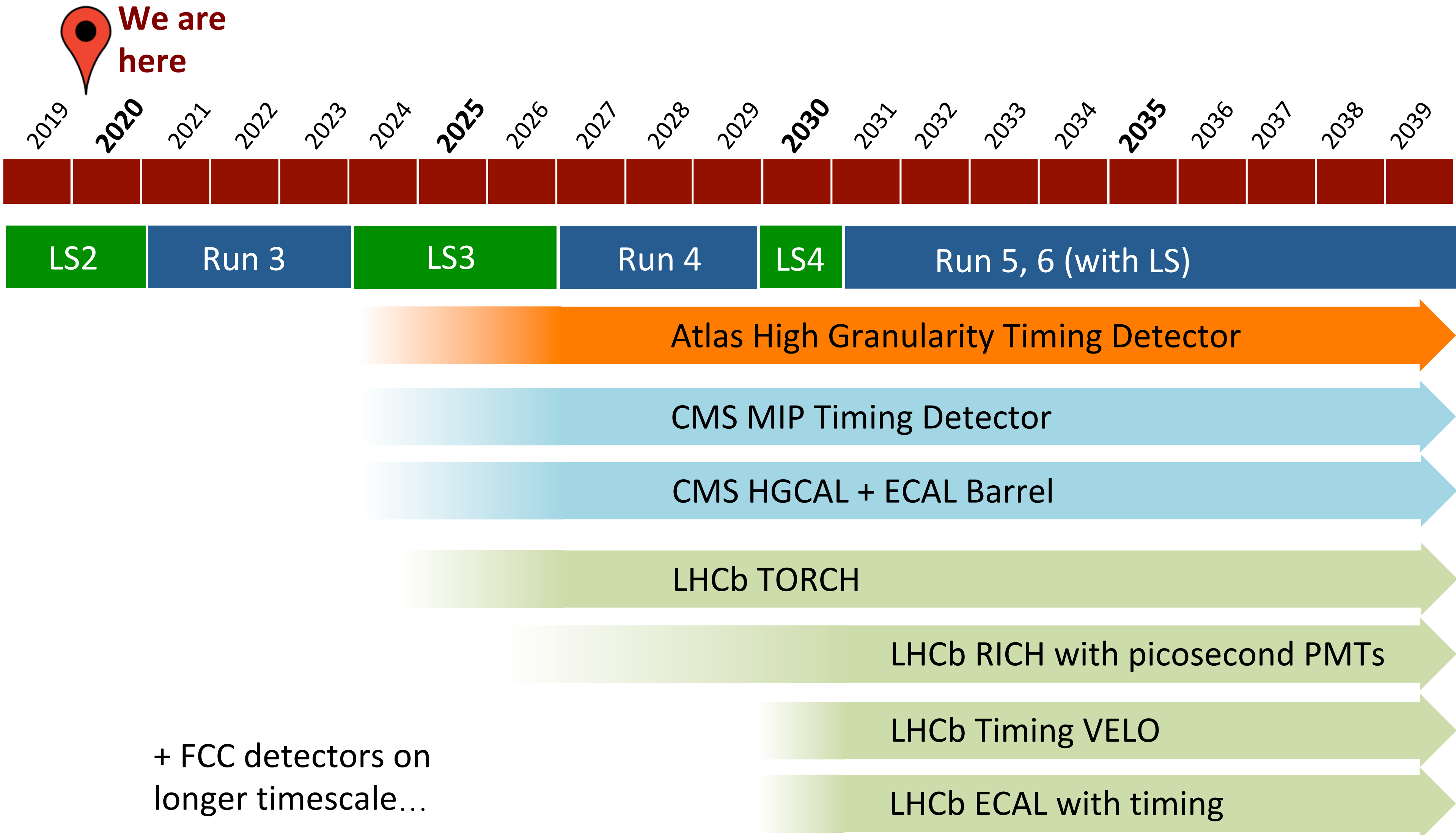


(Upgraded VELO for Runs 3-4)



From which pp collision?

HL-LHC Timing Timeline



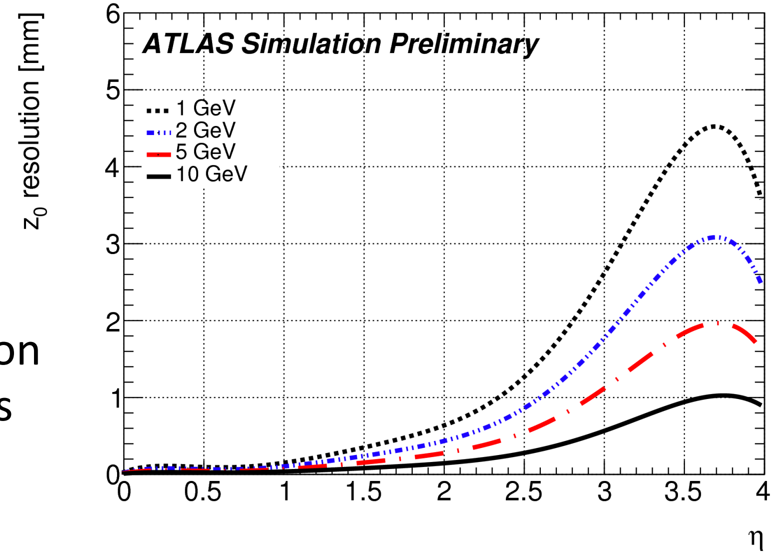
Why timing?

(1) Reduce effective event size for reconstruction

$\langle\mu\rangle$ = 200 PVs per event for HL-LHC Atlas & CMS (~**1.5 PV/mm)**

Spatial resolution of tracks can be >1mm especially in forward region

⇒ A track can be spatially consistent with multiple PVs



With 200ps vertex spread, 30ps track time resolution effectively allows event to be split into 6 time-slices

⇒ Recover $\langle\mu_{eff}\rangle = 30$

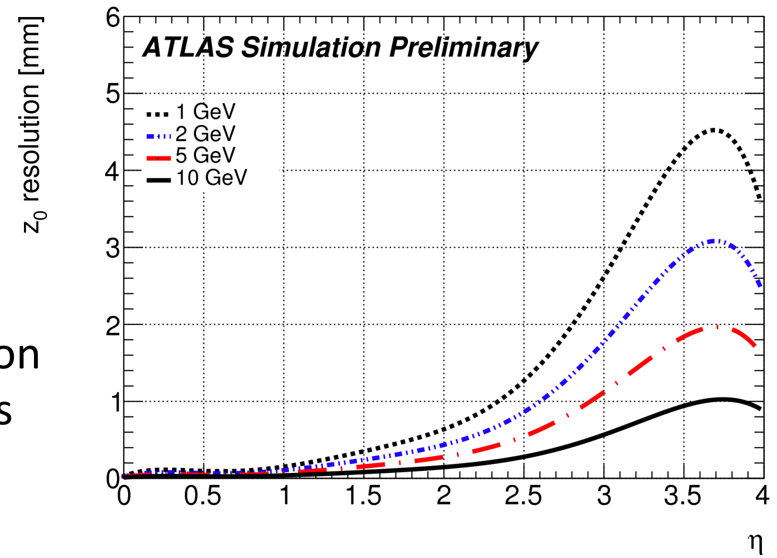
Significantly reduced computation in event reconstruction

Why timing?

(1) Reduce effective event size for reconstruction

<μ> = 200 PVs per event for HL-LHC Atlas & CMS (~**1.5 PV/mm)**

Spatial resolution of tracks can be >1mm especially in forward region
⇒ A track can be spatially consistent with multiple PVs



With 200ps vertex spread, 30ps track time resolution effectively allows event to be split into 6 time-slices
⇒ Recover $\langle\mu_{eff}\rangle = 30$

Significantly reduced computation in event reconstruction

But… timing comes at a cost: more information = larger events.
⇒ Where is break-even point?

Why timing?

(2) Suppress out-of-time backgrounds

Triggering, reconstructing, and analysing backgrounds is wasteful

Timing can suppress backgrounds at all levels

- ⇒ Higher signal efficiency within allowed resources
- ⇒ OR, Maintain efficiency with reduced resource use

Especially powerful for partially-reconstructed decays − missing spatial information

Many studies performed − generally 'physics driven' (i.e. efficiency vs purity)
…but: higher purity = more efficient computing

Can have major impact on computing at both trigger and offline level

Why timing?

(3) Unlock new capabilities

Precision timing often leads to previously impossible analyses

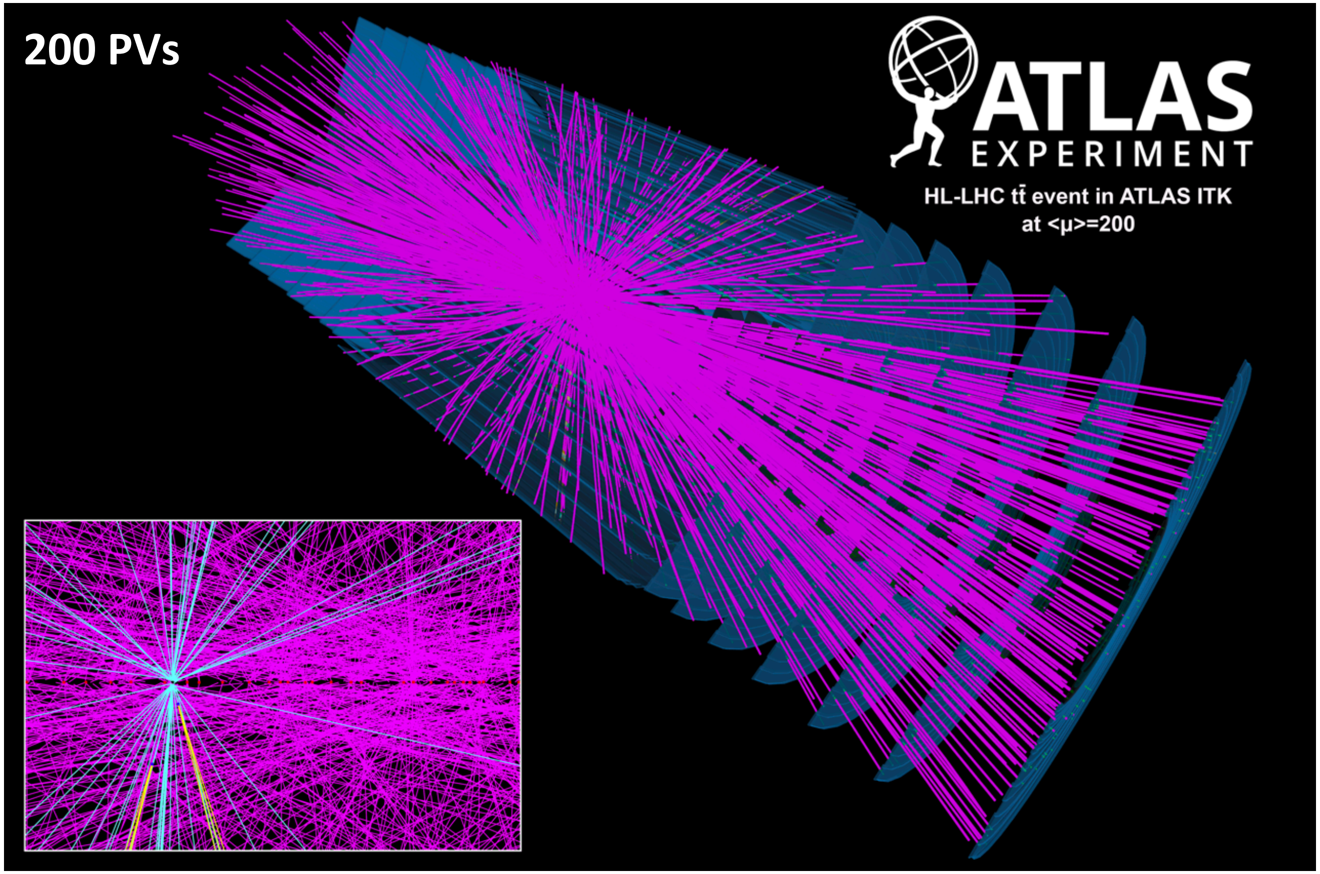
e.g. PID in new kinematic realm

e.g. Searches for exotic long-lived particles

Get more from existing resources − improved value-for-money

Case Study: Atlas HGTD

Track-to-vertex association

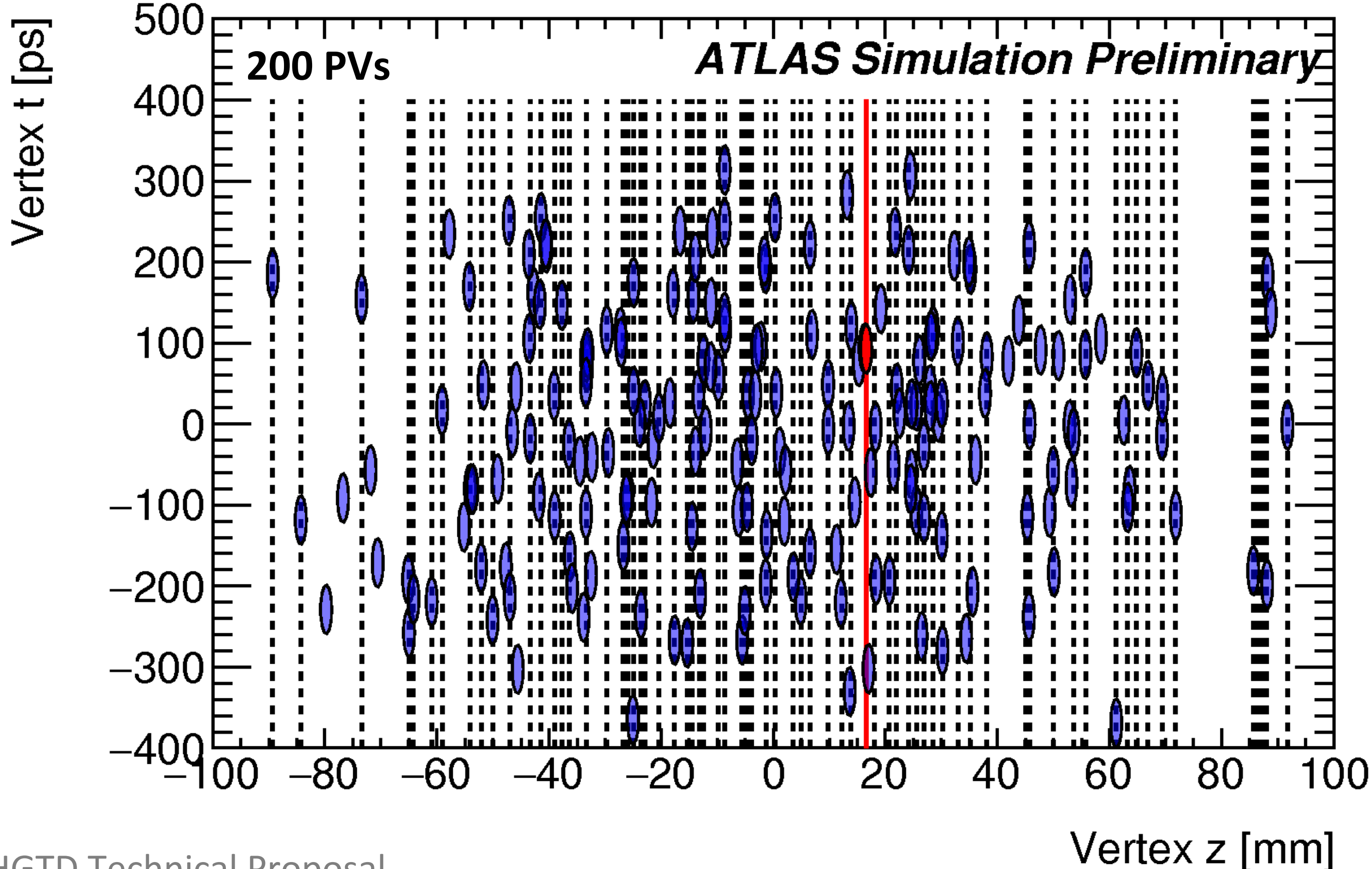


Using spatial information alone, many ambiguities, and incorrect assignments of tracks to vertices

Case Study: Atlas HGTD

ATLAS EXPERIMENT

Track-to-vertex association



Realistic time (30ps) and spatial (1mm) distributions of PV-track matching

On average a forward track will have ~**13 PVs** in search window

Time information breaks ambiguity

HGTD Technical Proposal
CERN-LHCC-2018-023

Case Study: Atlas HGTD

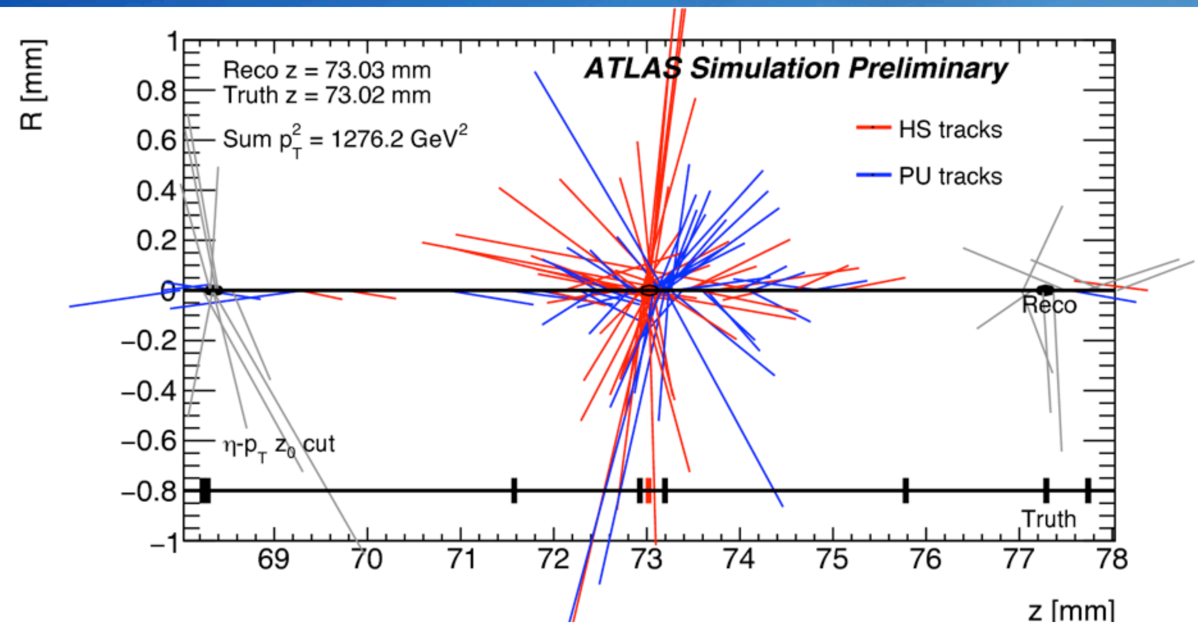
Vertex reconstruction

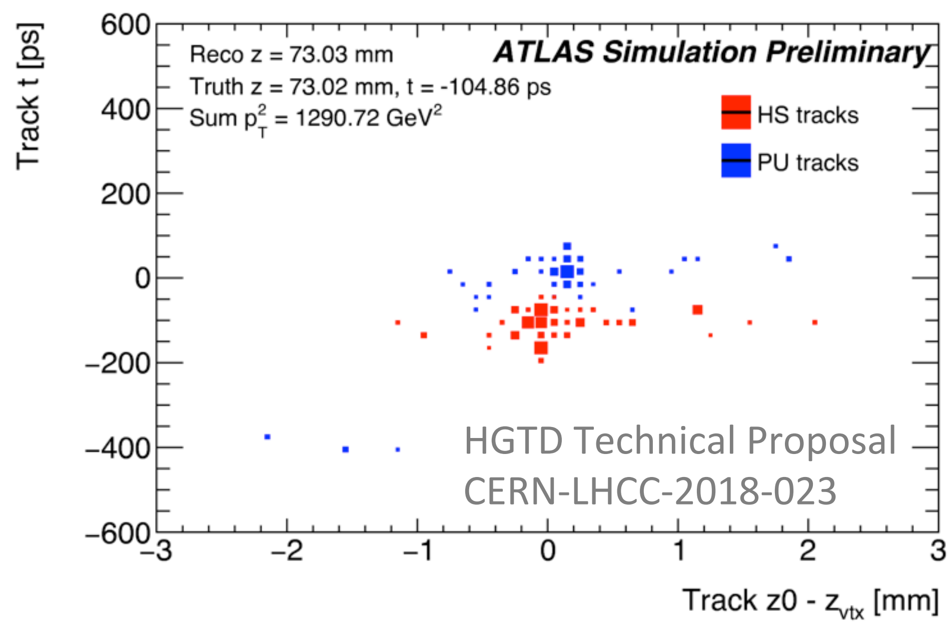
Timing can reject spurious tracks from PV of interest

Simplify the event ⇒ simplify the trigger and offline analysis

For Atlas & CMS, biggest gains in forward region where spatial information is degraded

For LHCb, everything is forward!





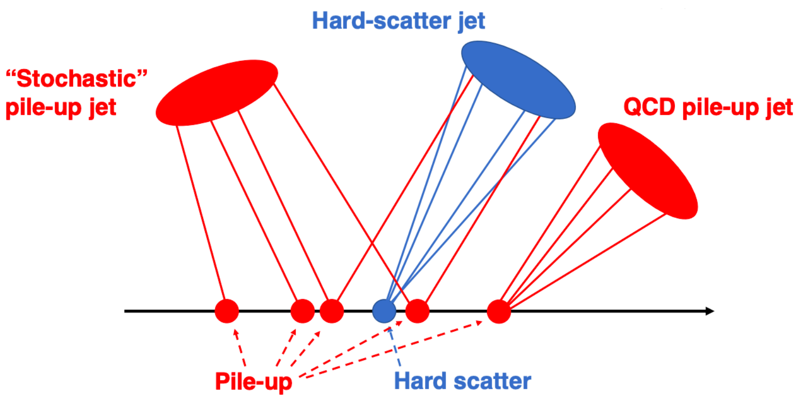
Case Study: Atlas HGTD

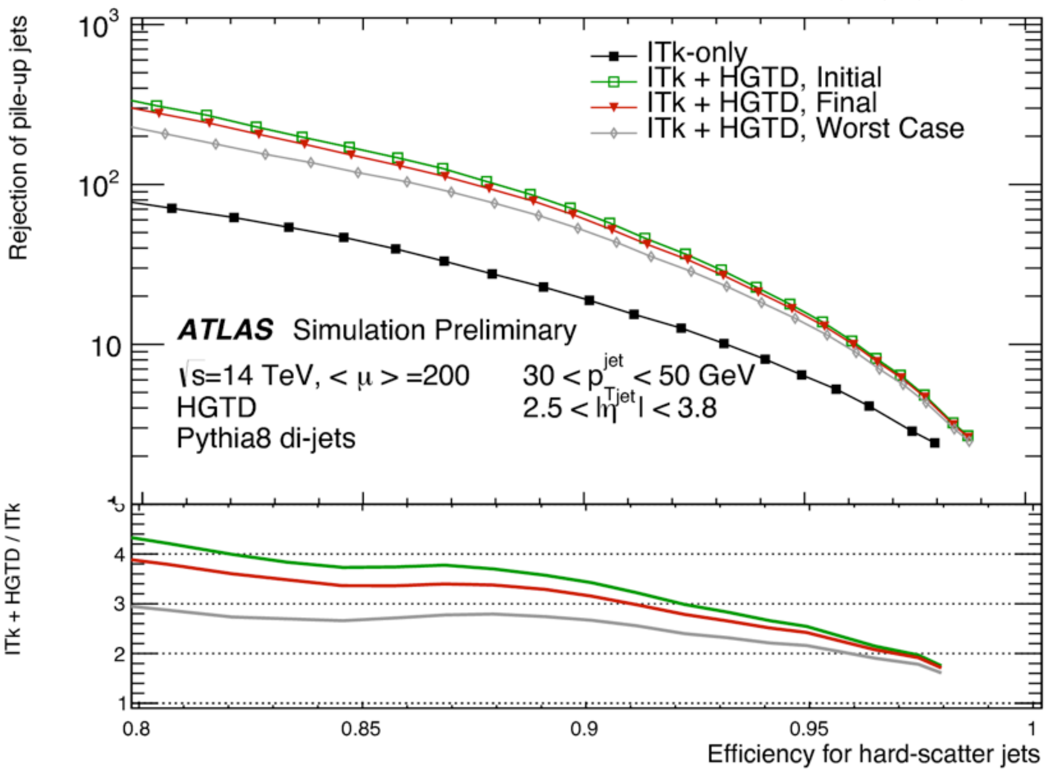
HGTD Technical Proposal
CERN-LHCC-2018-023

Consequences:

(1) Suppression of pileup jets

- Reduce number of particles considered in jet reconstruction
- Improve trigger & offline efficiency and purity





Timing ⇒ Factor 2-4 improvement in rejection of pile-up jets

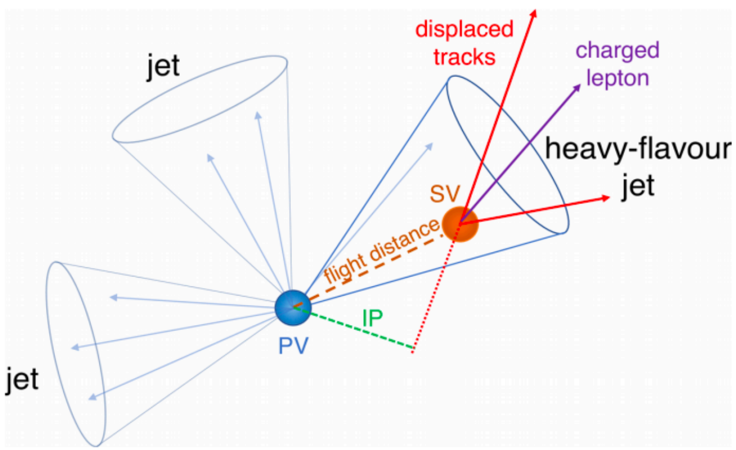
Case Study: Atlas HGTD

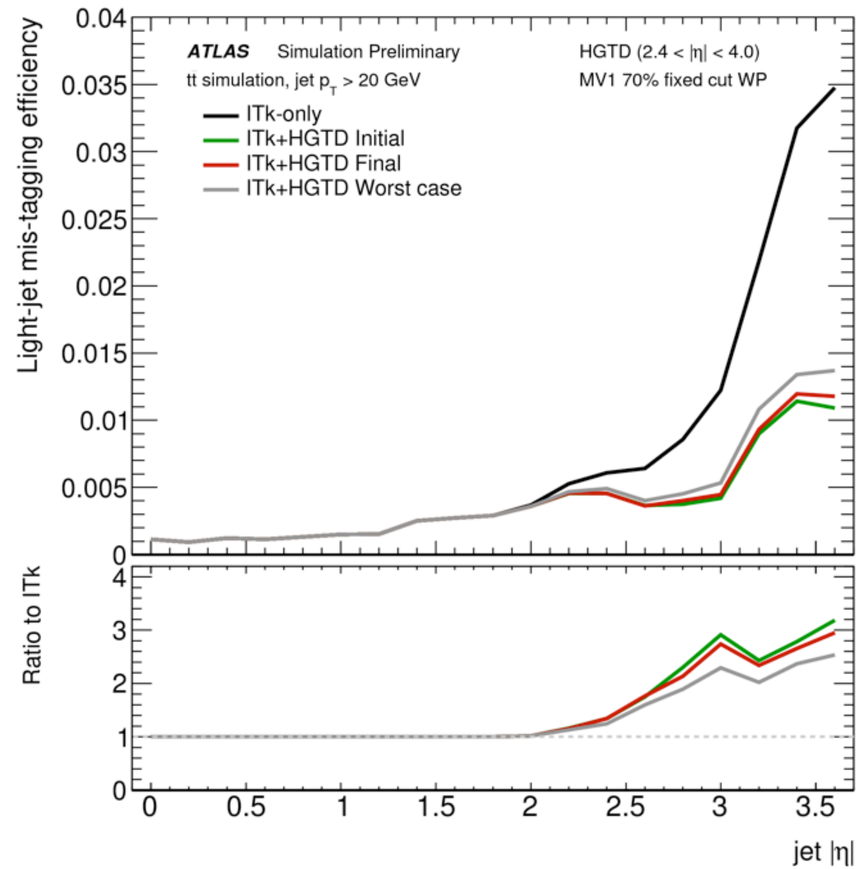
HGTD Technical Proposal
CERN-LHCC-2018-023

Consequences:

(2) Improved jet flavour-tagging

- Suppress number of displaced tracks considered by algorithm
- Improve trigger & offline efficiency and purity



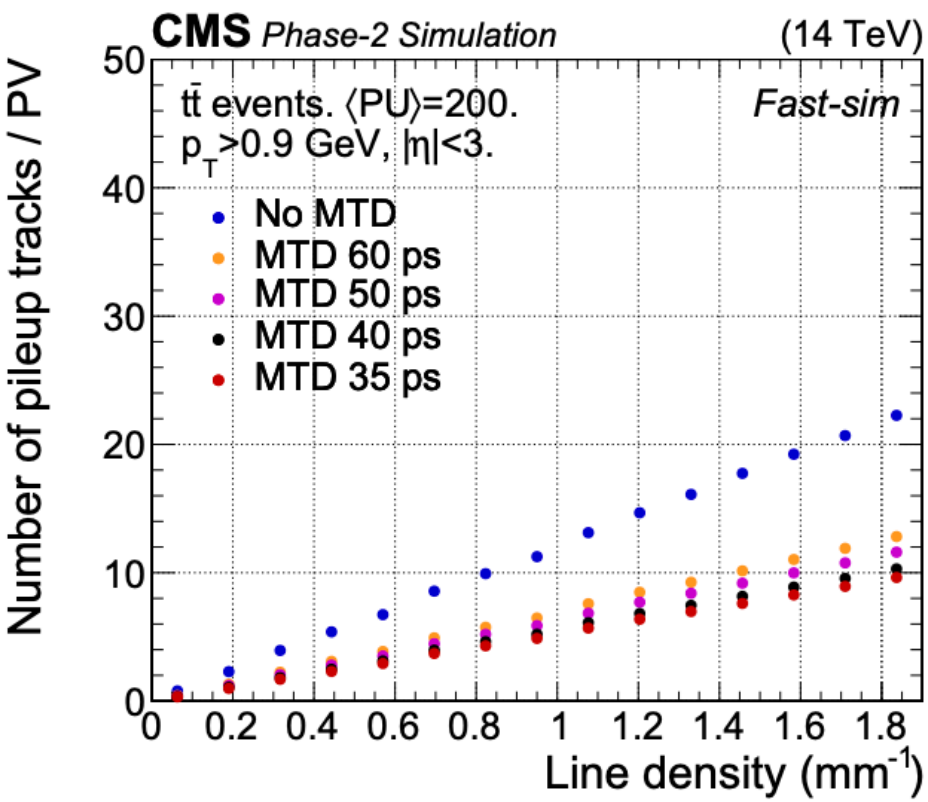


Timing ⇒ Factor 2-3 improvement in rejection of light jets

Case Study: CMS Vertexing

MTD TDR
CERN-LHCC-2019-003

15% PV merging → Add σ=30ps track timestamp → 1% PV merging

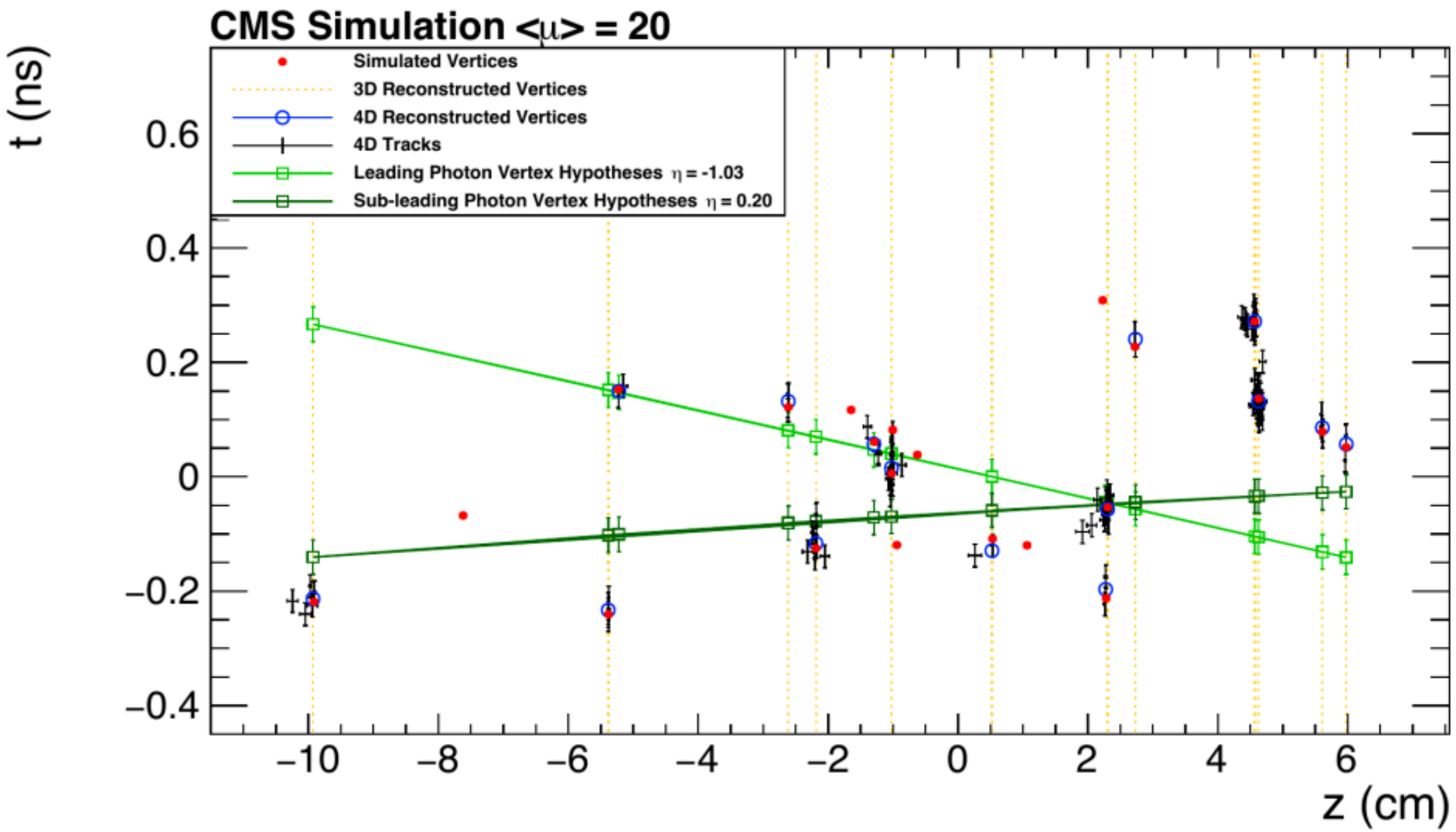


Timing ⇒ Factor >2 reduction in pile-up tracks in signal PV

Case Study: CMS γ⟷PV association

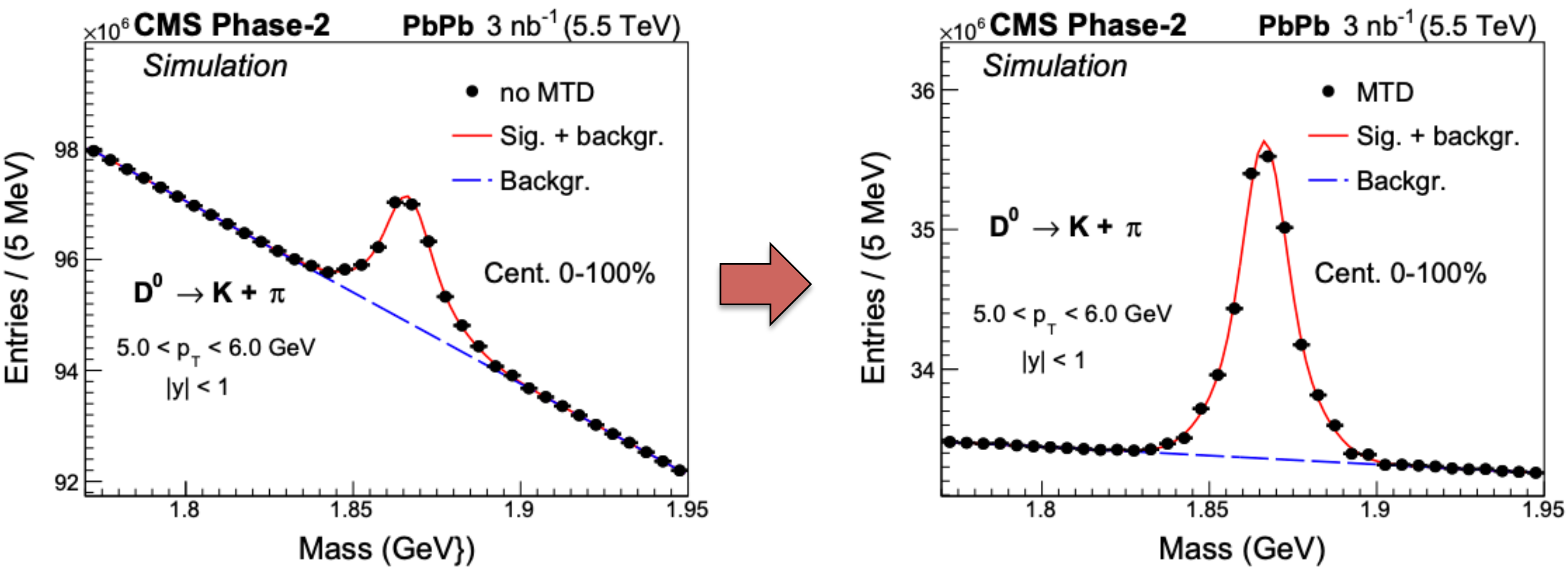
MTD TDR
CERN-LHCC-2019-003

Timing ⇒ Can match photons (using ECAL timing) to PV even without any associated track (e.g. H→γγ)



Case Study: CMS PID by Time-of-Flight

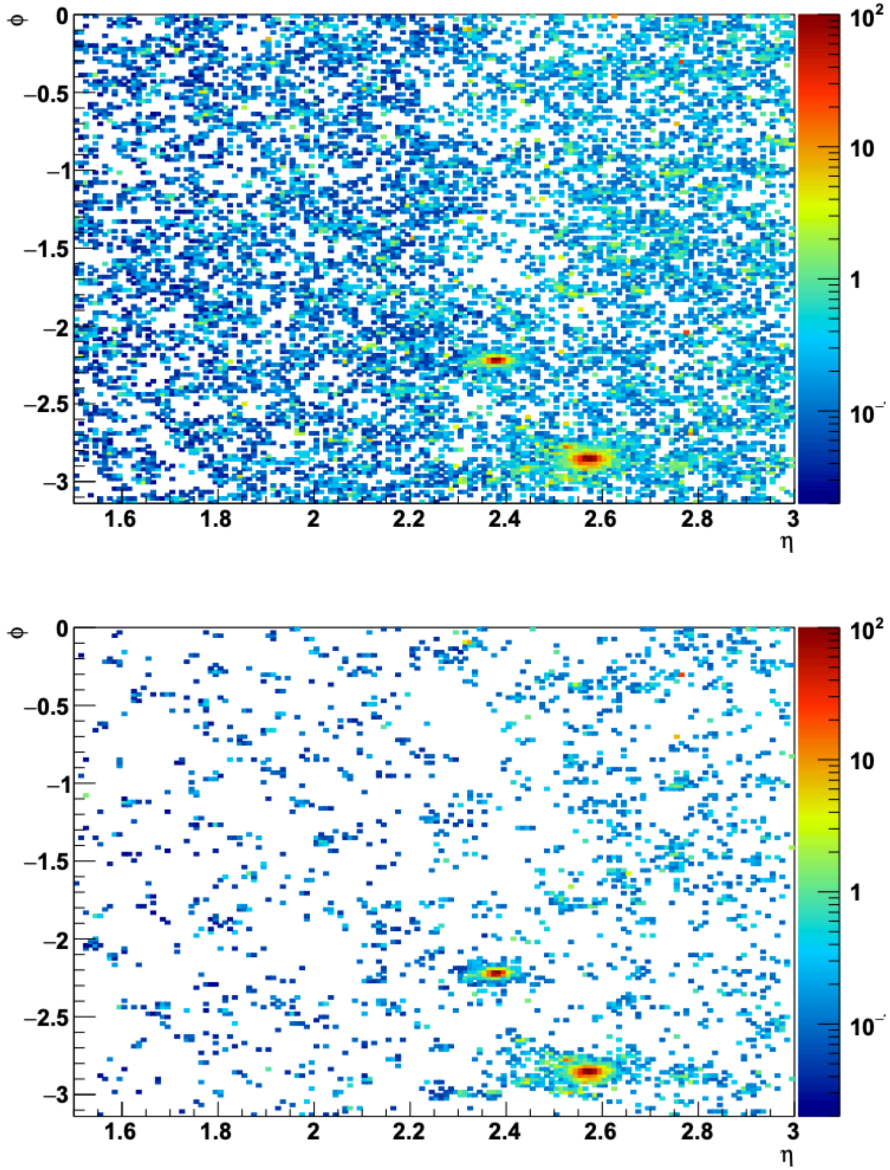
Timing ⇒ For low p_T particles, PID from time-of-flight can suppress backgrounds (e.g. for heavy ion or flavour programme)



MTD TDR
CERN-LHCC-2019-003

Case Study: CMS Calorimetry

Timing ⇒ Suppress out-of-time calorimeter deposits to simplify object reconstruction



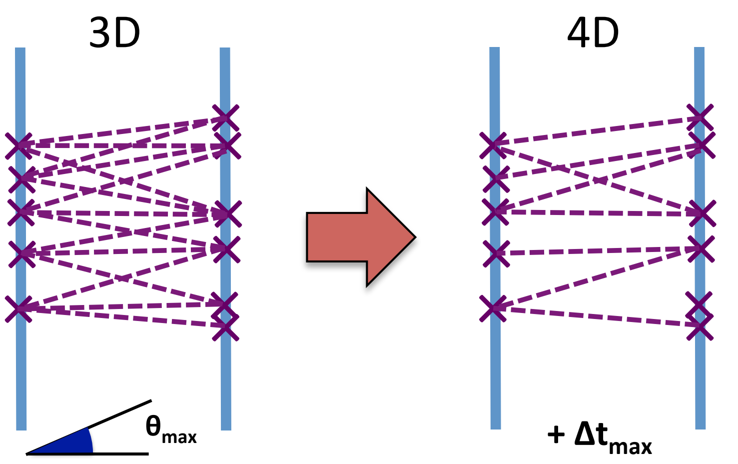
Example H→γγ event without (top) and with (bottom) time information (|Δt| < 90ps)

HGCAL TDR
CERN-LHCC-2017-023

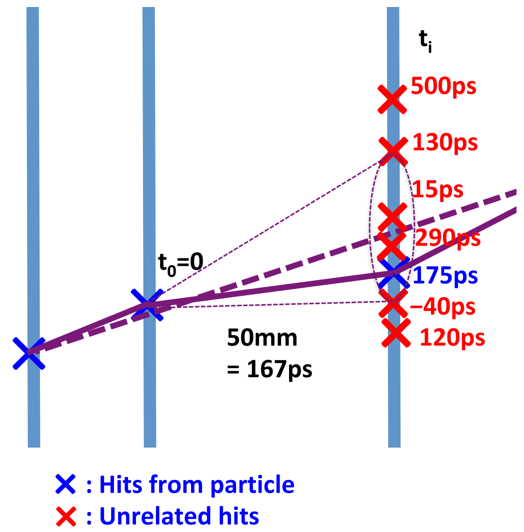
Case Study: LHCb Timing VELO

50ps timestamps on all pixel hits ⇒ additional dimension in pattern recognition

Track seeding



Track propagation



Immediate reduction in combinations:

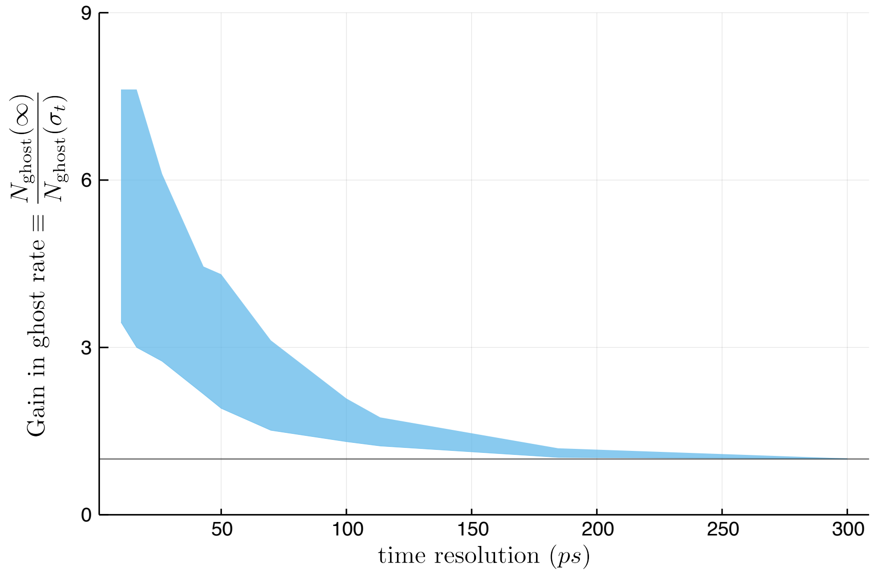
- Faster pattern recognition
- Reduced rate of fake tracks ⇒ reduced data volume

All happening at the trigger level

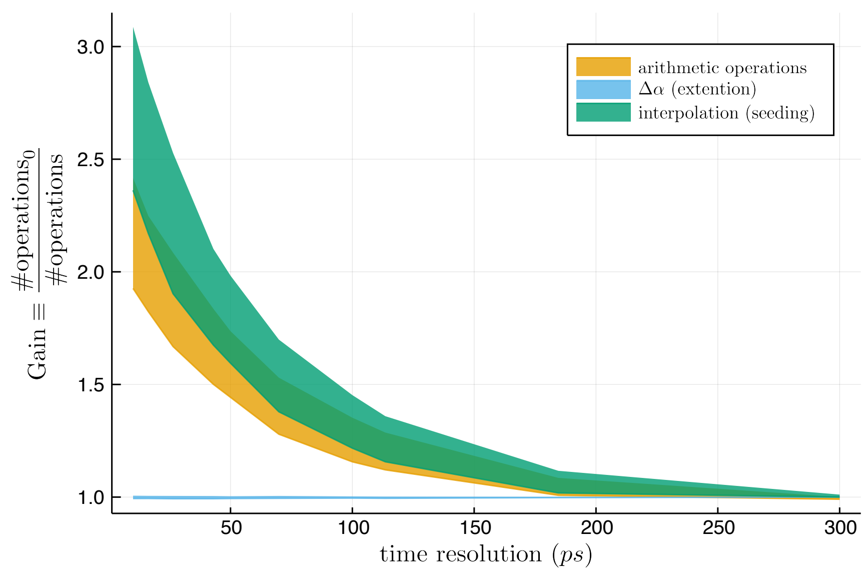
Case Study: LHCb Timing VELO

Early studies with parameteric simulations show large potential benefits

Large reduction in fake tracks…



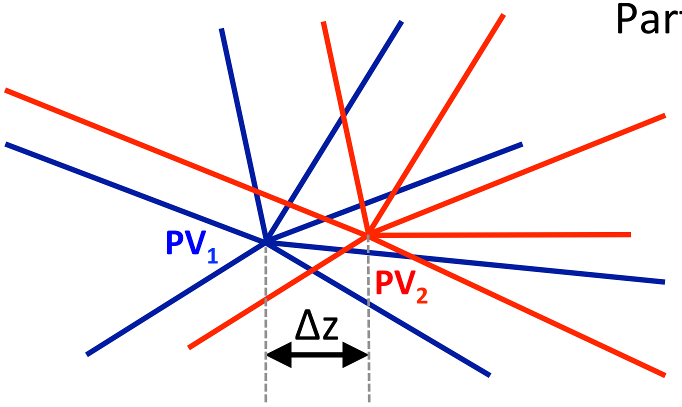
… and in number of operations



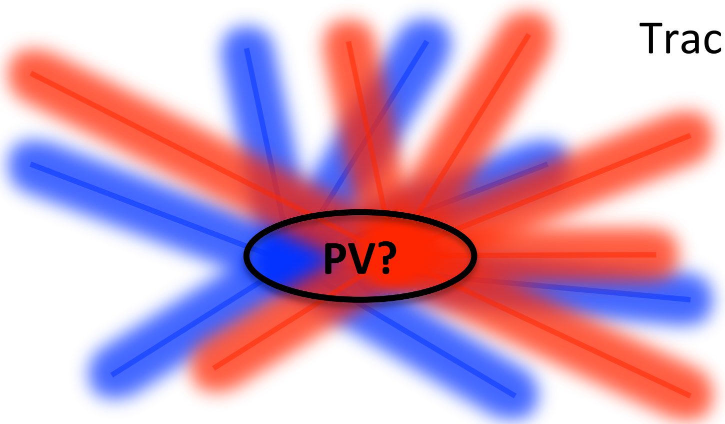
Case Study: LHCb Timing VELO

+ Studies on full simulation with naïve use of timing

Particles



Tracks…



Improved PV reconstruction
− fewer split or merged vertices

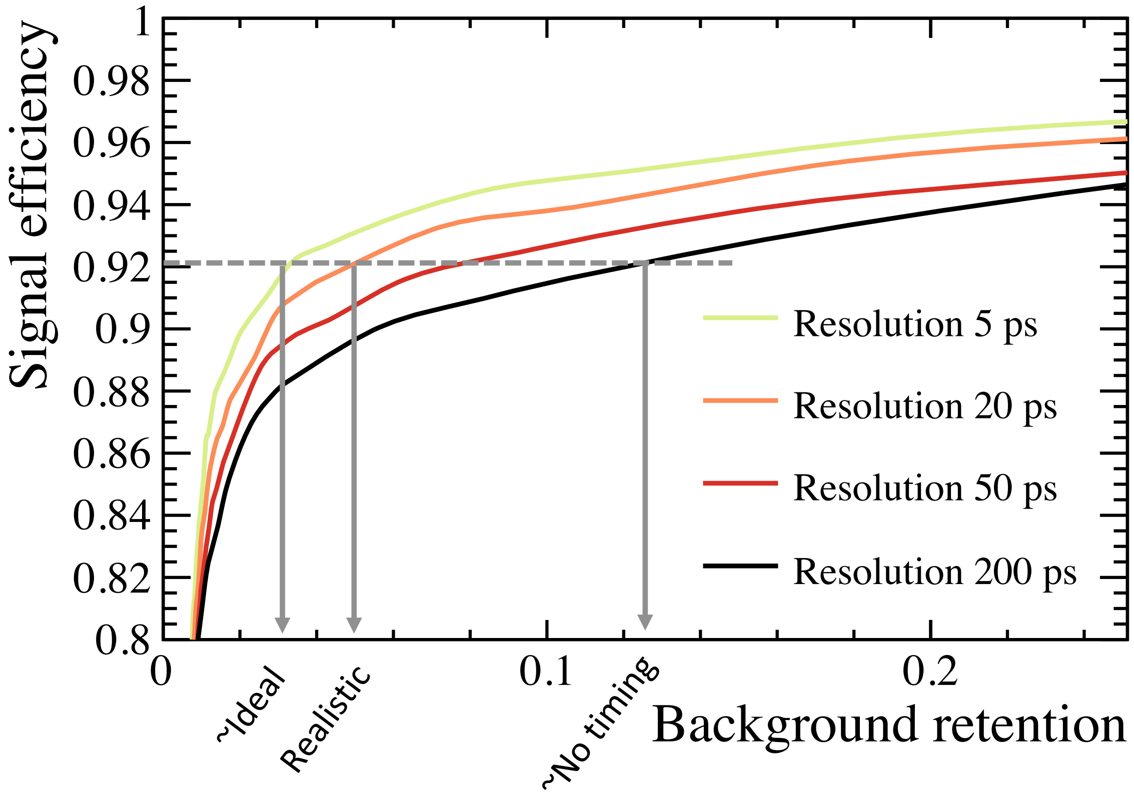
With 50ps per-hit time precision, can recover PV performance of Run 3, under HL-LHC conditions (7.5x higher lumi)

…plus improved event reconstruction time… (studies in progress)

Case Study: LHCb Timing VELO

+ Studies on full simulation with naïve use of timing

Timing will be crucial for triggering on typical LHCb signals (tracks from long-lived b and c hadrons)



Displaced track trigger performance with simple use of timing

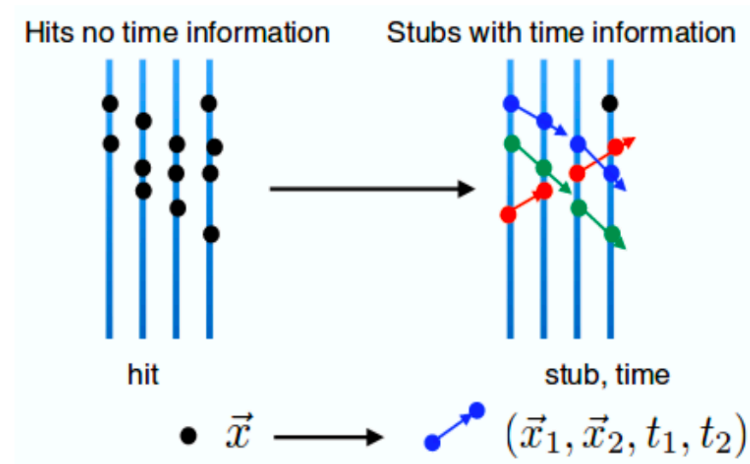
e.g. 20ps/track gives factor >2 reduction in BG

In reality, much more to be gained, e.g. with (2,3,…)-track triggers

Generic 4D Reconstruction R&D

Uses “Stubs” (hit doublets) in algorithm to allow highly-parallelised pattern recognition

LHCb Timing VELO as an example application

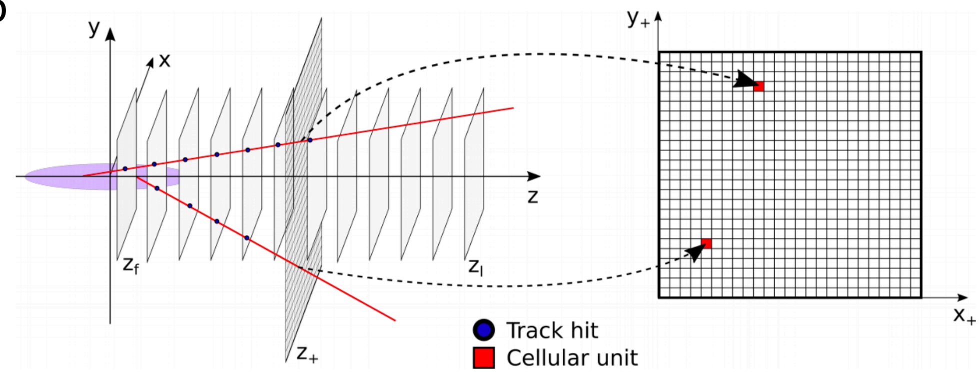


Selected stubs are projected onto 2D reference plane

Clustering engine then combines stubs into track candidates

Uses modular, scalable FPGA architecture:

- One FPGA for each detector plane doublet – to select stubs
- One FPGA for each ‘track region’ for clustering engine

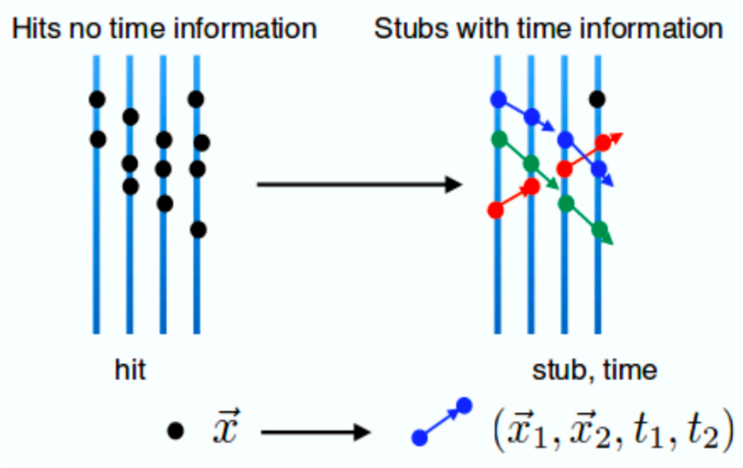


JINST 11 (2016), C11040
+ Talk by M. Petruzzo at ‘Connecting the Dots’ workshop (Apr 2018)

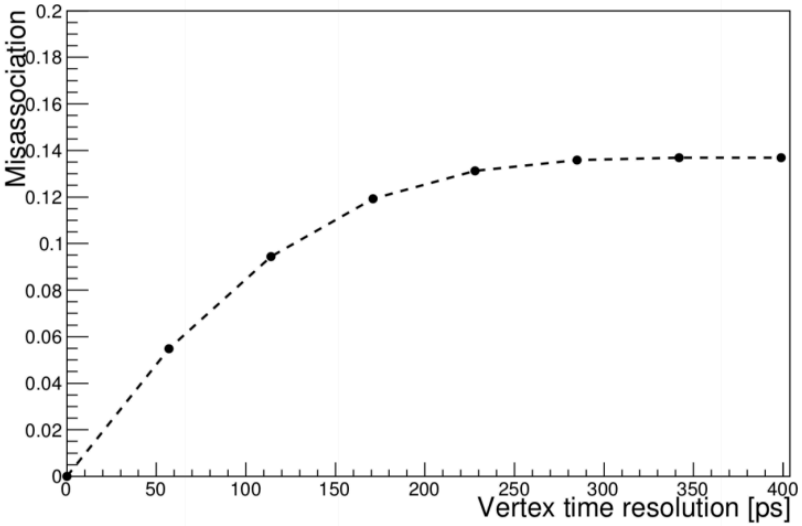
Generic 4D Reconstruction R&D

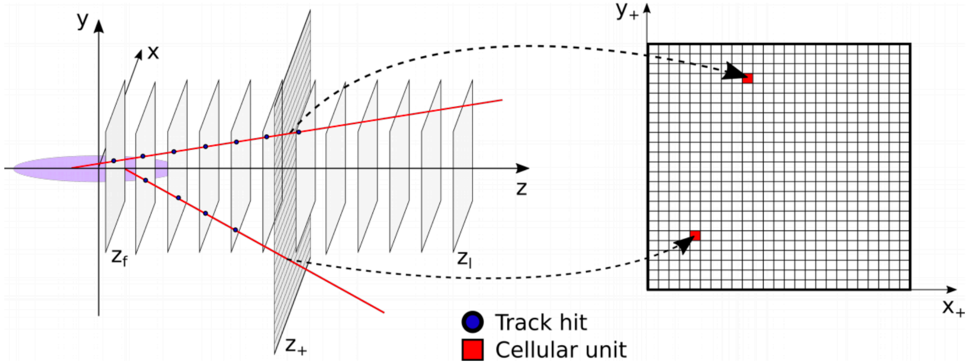
Uses “Stubs” (hit doublets) in algorithm to allow highly-parallelised pattern recognition

LHCb Timing VELO as an example application



- **Improved track efficiency and purity**
- **Reduces track-vertex mis-association from >10% to <1%**





JINST 11 (2016), C11040
+ Talk by M. Petruzzo at ‘Connecting the Dots’ workshop (Apr 2018)

Open Questions: Avenues for R&D

How to benchmark computing performance & value-for-money?

⇒ CPU / GPU / FPGA

⇒ Flexible vs application-limited resources

Can we emulate in simulation?

Can we pass information (e.g. detector alignment) to the algorithms?

Need set of time-aware reco algorithms (e.g. for LHCb timing VELO) before we can benchmark performance.

Open Questions: Avenues for R&D

How to benchmark computing performance & value-for-money?

⇒ CPU / GPU / FPGA

Benefits of **per-hit** versus **per-track** timestamps

⇒ How essential is 4D tracking for HL-LHC and FCC applications?
(both for physics performance and resource use)

⇒ More detailed and realistic studies will be needed, with different use-cases

There is a cost to adding timing (data and algorithmic) – when does it become beneficial?

Open Questions: Avenues for R&D

How to benchmark computing performance & value-for-money?

⇒ CPU / GPU / FPGA

Benefits of **per-hit** versus **per-track** timestamps

⇒ How essential is 4D tracking for HL-LHC and FCC applications?
(both for physics performance and resource use)

Can we gain even more by considering timing globally – time-aware Kalman filter?

⇒ CMS now working to incorporate timing into particle flow, but no timing in tracker / vertex detector

Open Questions: Avenues for R&D

How to benchmark computing performance & value-for-money?

⇒ CPU / GPU / FPGA

Benefits of **per-hit** versus **per-track** timestamps

⇒ How essential is 4D tracking for HL-LHC and FCC applications?
(both for physics performance and resource use)

Can we gain even more by considering timing globally – time-aware Kalman filter?

⇒ CMS now working to incorporate timing into particle flow, but no timing in tracker / vertex detector

Matching objects between detectors relies on knowledge of particle type (=speed)

⇒ i.e. we really need 5D reconstruction (space + time + particle ID)

⇒ Inverting logic, could we derive PID info from global '5D' Kalman fit?

Open Questions: Avenues for R&D

How to benchmark computing performance & value-for-money?

⇒ CPU / GPU / FPGA

Benefits of **per-hit** versus **per-track** timestamps

⇒ How essential is 4D tracking for HL-LHC and FCC applications? (both for physics performance and resource use)

Can we gain even more by considering timing globally – time-aware Kalman filter?

⇒ CMS now working to incorporate timing into particle flow, but no timing in tracker / vertex detector

Matching objects between detectors relies on knowledge of particle type (=speed)

⇒ i.e. we really need 5D reconstruction (space + time + particle ID)

Can we reduce **simulation** resources? Generate only in-time part of events?

Summary and Outlook

In HL-LHC era ⇒ running out of slack in the system for 'easy' gains
(faster algorithms, parallelisation, new architectures)

Computing needs will increasingly inform (dictate?) detector design and technology
⇒ Precision timing will be a big theme here

Summary and Outlook

In HL-LHC era ⇒ running out of slack in the system for 'easy' gains
(faster algorithms, parallelisation, new architectures)

Computing needs will increasingly inform (dictate?) detector design and technology
⇒ Precision timing will be a big theme here

So far, timing studies mainly focused on gains in physics performance
⇒ Reconstruction + trigger benefits come as a side effect
⇒ But gains here often larger

Tendency to assume everything is 'physics driven' – i.e. build the best detector and then deal with the reco/computing later
⇒ Not sustainable in HL-LHC era
⇒ LHCb Run 3 experience will hopefully help to change minds

Summary and Outlook

In HL-LHC era ⇒ running out of slack in the system for 'easy' gains
(faster algorithms, parallelisation, new architectures)

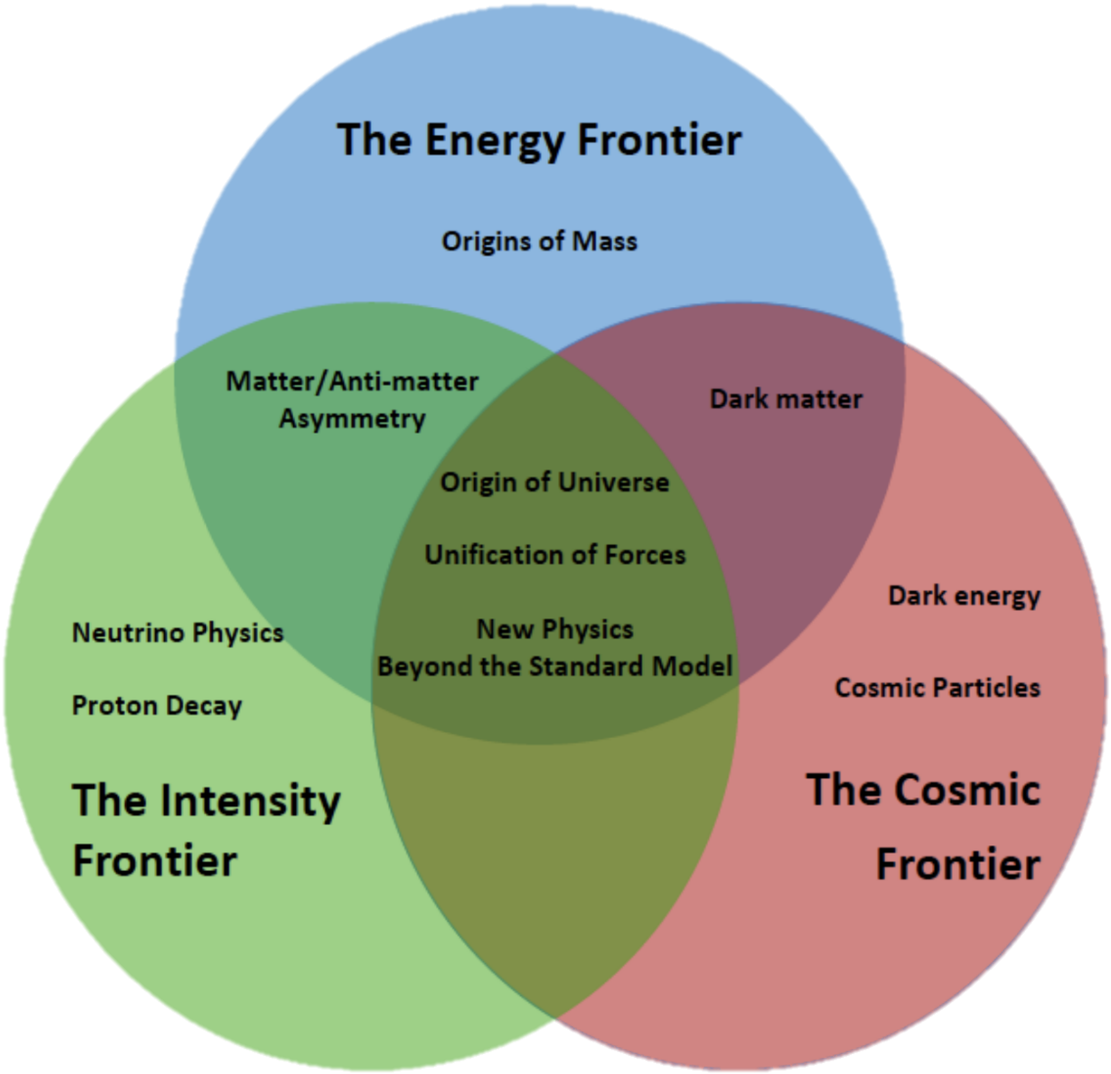
Computing needs will increasingly inform (dictate?) detector design and technology
⇒ Precision timing will be a big theme here

So far, timing studies mainly focused on gains in physics performance
⇒ Reconstruction + trigger benefits come as a side effect
⇒ But gains here often larger

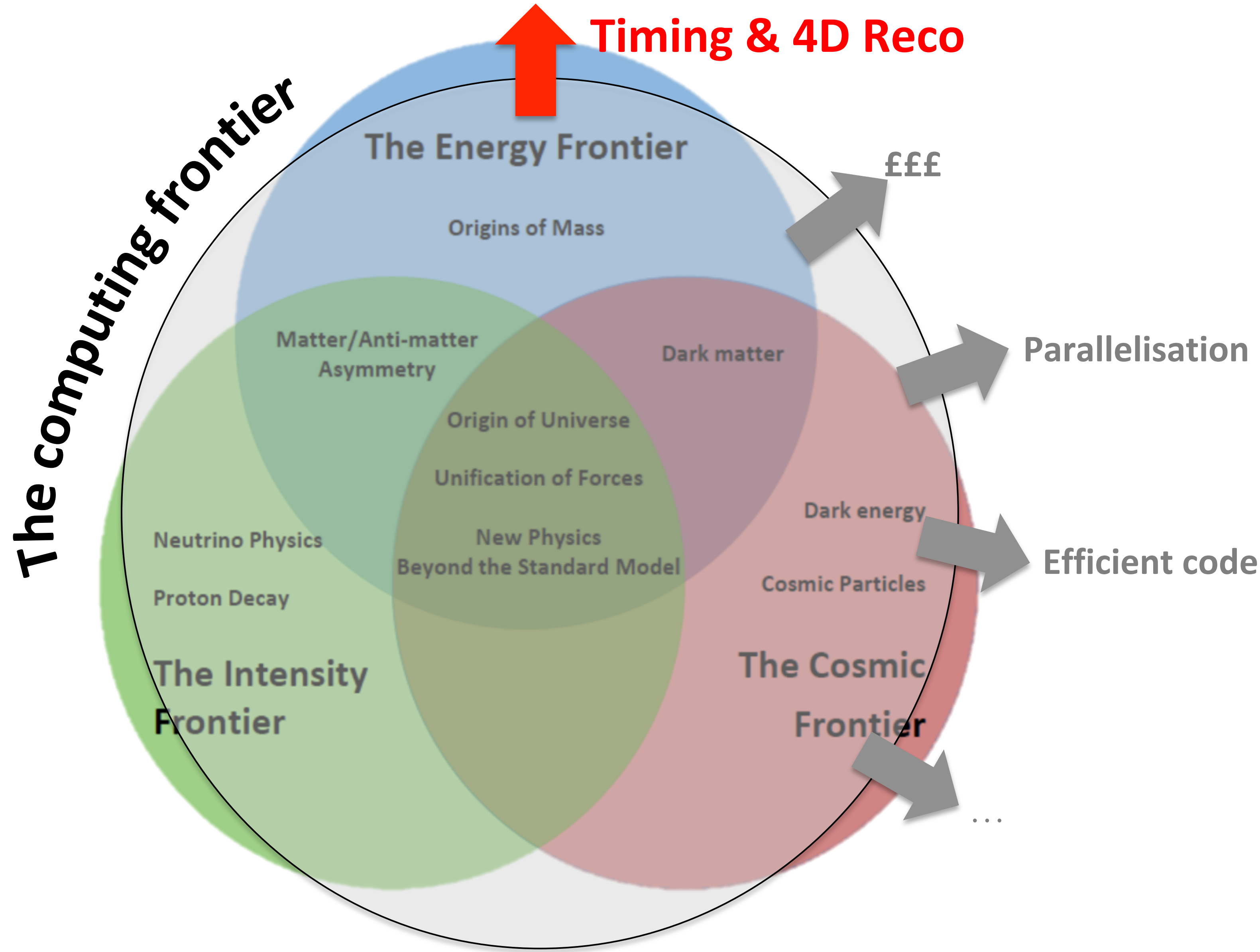
Tendency to assume everything is 'physics driven' – i.e. build the best detector and then deal with the reco/computing later
⇒ Not sustainable in HL-LHC era
⇒ LHCb Run 3 experience will hopefully help to change minds

Need dedicated studies to quantify these improvements further
⇒ Crucial to motivate future detector designs
⇒ Crucial to plan for computing needs, and avoid unexpected surprises

A new frontier…



A new frontier…



Future Plans: Detectors

Upgraded and new detectors for LS3 and LS4 moving through formal review process

- Atlas HGTD Technical Design Report being finalised
- CMS TDRs (timing plane + barrel + calorimeter upgrades) already public
- LHCb Future Upgrades approved to go to FTDR (and SoI approved by STFC)

Expect many more detailed studies in coming ~18 months

⇒ Likely 'physics driven'

⇒ Important to also motivate from computing side