Downstream tracking and vertexing at the first stage of the LHCb trigger

Arantza de Oyanguren Campos, Brij Kishor Jashal, Volodymyr Svintozelskyi, Valerii Kholoimov, Jiahui Zhuo on behalf of LHCb collaboration

IFIC, U.V. - CSIC

19–25 Oct 2024, Conference on Computing in High Energy and Nuclear Physics 2024, Kraków, Poland

Financiado por la Unión Europea NextGenerationEU

Gobierno de España Ministerio de Ciencia, Innovación y Universidades

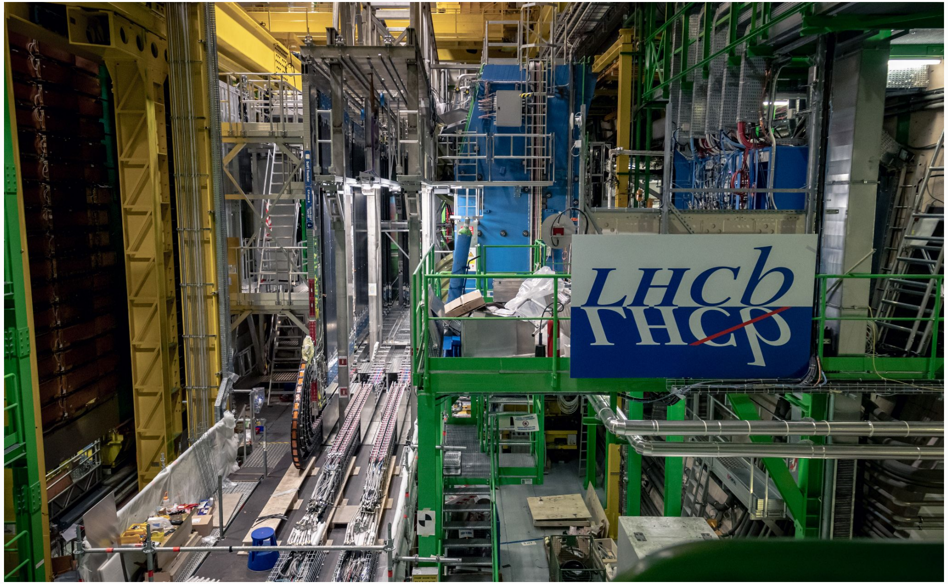
Plan de Recuperación, Transformación y Resiliencia

Generalitat Valenciana Conselleria de Educación, Universidades y Empleo

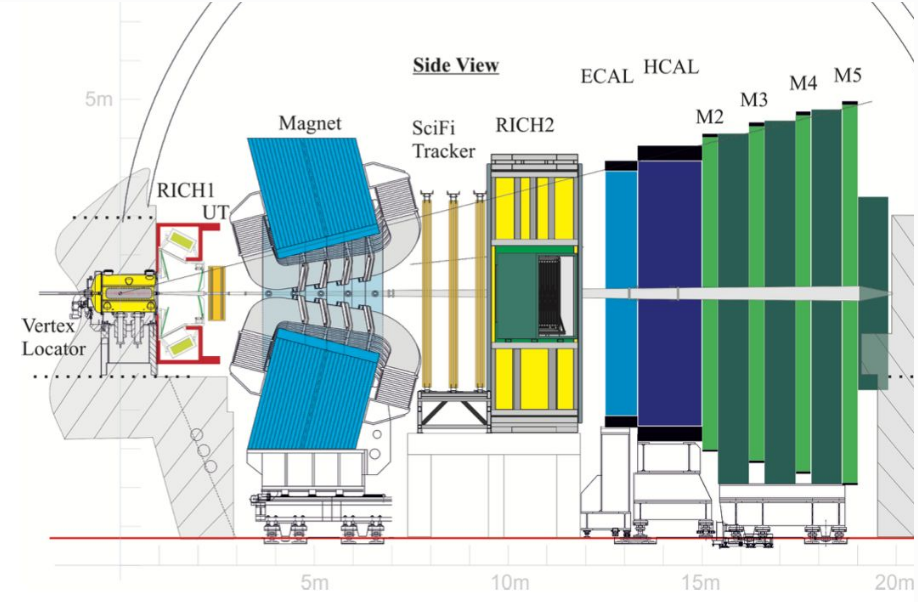
GVANEXT Fondos Next Generation en la Comunitat Valenciana

Outline

- LHCb experiment
  - The LHCb trigger system
  - The LHCb tracking system
- HLT1 Downstream tracking
  - Algorithm design
  - Performance
- Summary



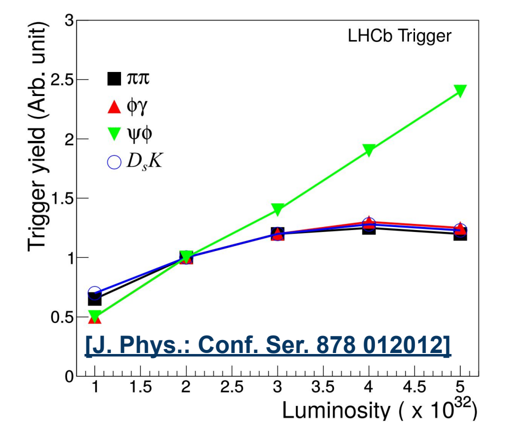
LHCb experiment

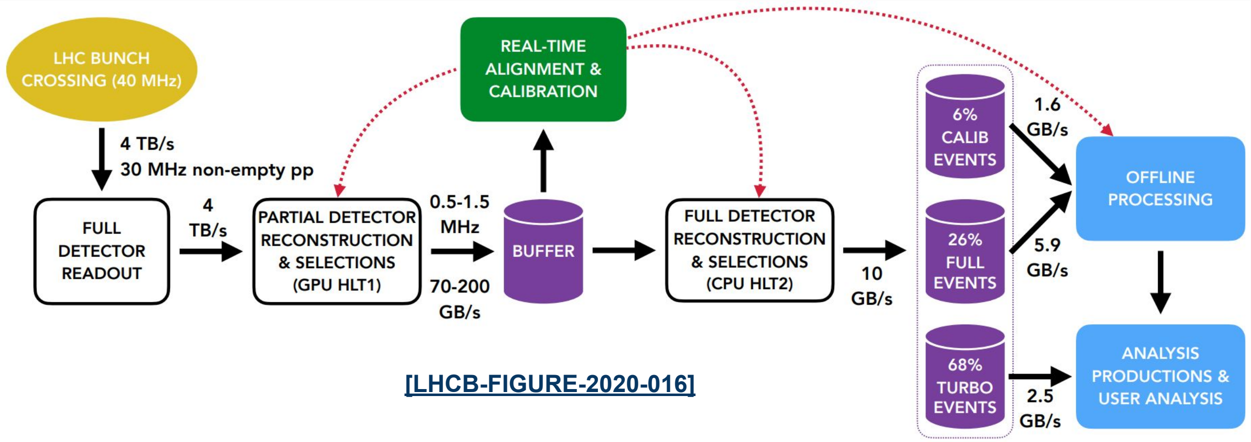


[CERN-LHCC-2014-001; LHCB-TDR-015]

- **LHCb** is one of the four main experiments at the **LHC**, focused on precise measurements in the **beauty** and **charm sectors**.
- For **Run 3** data-taking, LHCb must handle an instantaneous luminosity **x 5** larger than Run 2 ($\mathcal{L}_{\text{inst.}} = 2 \cdot 10^{33} cm^{-2} s^{-1}$), with a average **pile-up** of 5.2 ($< \mu >= 5.2$).
- A new set of **tracking detectors** have been designed to handle higher radiation damage and increased track multiplicity, and an **upgraded trigger system** has been developed to manage it.

LHCb trigger system



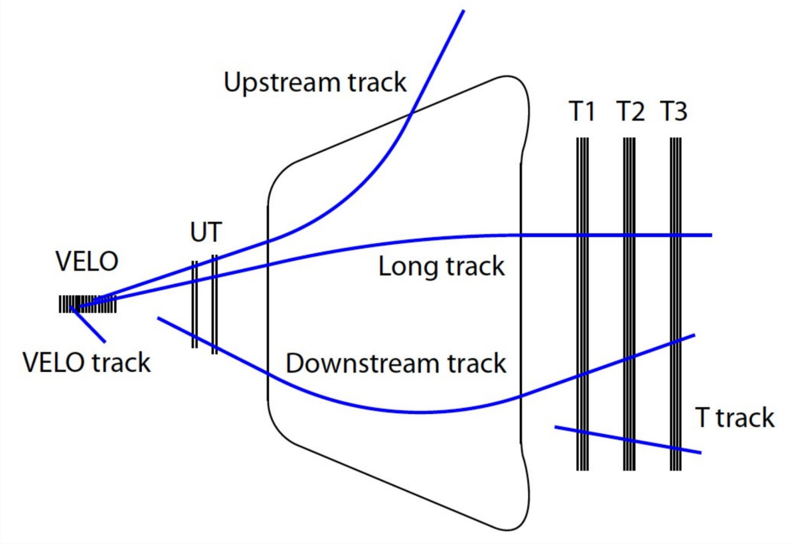


- The **hardware trigger (L0)** reached saturation at high luminosity ⟶ Removal of **L0** in **Run 3** and **HLT1** operating at **30 MHz**.
- **HLT1** and **HLT2** perform Real-Time Analysis (**RTA**) to reconstruct the event and make trigger decisions based on reconstructed objects.
- To handle the high throughput requirements, **HLT1** now runs as a **GPU-based** application called **Allen** during data-taking.

LHCb tracking system

- LHCb has three different **tracking detectors** to reconstruct the trajectories of charged particles:
  - Vertex Locator (**VELO**);
  - Upstream Tracker (**UT**);
  - Scintillating Fibres (**SciFi**) / **T** stations.

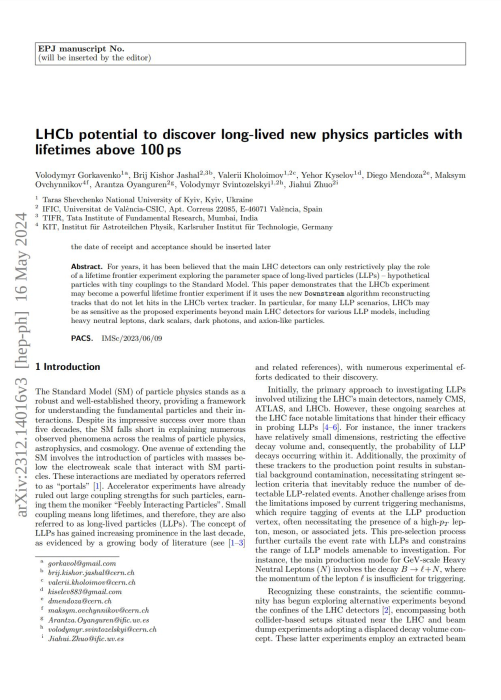
- In the **LHCb tracking system**, reconstructed tracks are classified based on their hits in tracking detectors:
  - **Long track**: tracks with hits in VELO and SciFi, optionally UT;
  - **Downstream track**: tracks with hits in the UT and SciFi only.



[CERN-LHCb-PUB-2021-005]

- **Long tracks** and **downstream tracks** are used in most physics analyses, while the other types either serve as a component of another track type or are mainly used for detector studies.

HLT1 Downstream tracking



EPJ manuscript No.
(will be inserted by the editor)

arXiv:2312.14016v3 [hep-ph] 16 May 2024

LHCb potential to discover long-lived new physics particles with lifetimes above 100 ps

Volodymyr Gorkavenko[1a], Brij Kishor Jashal[2,3b], Valerii Kholoimov[1,2c], Yehor Kyselov[1d], Diego Mendoza[2e], Maksym Ovchynnikov[4f], Arantza Oyanguren[2g], Volodymyr Svintozelskyi[1,2h], Jiahui Zhuo[2i]

[1] Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
[2] IFIC, Universitat de València-CSIC, Apt. Correus 22085, E-46071 València, Spain
[3] TIFR, Tata Institute of Fundamental Research, Mumbai, India
[4] KIT, Institut für Astroteilchen Physik, Karlsruher Institut für Technologie, Germany

the date of receipt and acceptance should be inserted later

Abstract. For years, it has been believed that the main LHC detectors can only restrictively play the role of a lifetime frontier experiment exploring the parameter space of long-lived particles (LLPs) – hypothetical particles with tiny couplings to the Standard Model. This paper demonstrates that the LHCb experiment may become a powerful lifetime frontier experiment if it uses the new `Downstream` algorithm reconstructing tracks that do not let hits in the LHCb vertex tracker. In particular, for many LLP scenarios, LHCb may be as sensitive as the proposed experiments beyond main LHC detectors for various LLP models, including heavy neutral leptons, dark scalars, dark photons, and axion-like particles.

PACS. IMSc/2023/06/09

1 Introduction

The Standard Model (SM) of particle physics stands as a robust and well-established theory, providing a framework for understanding the fundamental particles and their interactions. Despite its impressive success over more than five decades, the SM falls short in explaining numerous observed phenomena across the realms of particle physics, astrophysics, and cosmology. One avenue of extending the SM involves the introduction of particles with masses below the electroweak scale that interact with SM particles. These interactions are mediated by operators referred to as "portals" [1]. Accelerator experiments have already ruled out large coupling strengths for such particles, earning them the moniker "Feebly Interacting Particles". Small coupling means long lifetimes, and therefore, they are also referred to as long-lived particles (LLPs). The concept of LLPs has gained increasing prominence in the last decade, as evidenced by a growing body of literature (see [1–3] and related references), with numerous experimental efforts dedicated to their discovery.

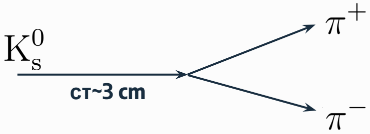
Initially, the primary approach to investigating LLPs involved utilizing the LHC's main detectors, namely CMS, ATLAS, and LHCb. However, these ongoing searches at the LHC face notable limitations that hinder their efficacy in probing LLPs [4–6]. For instance, the inner trackers have relatively small dimensions, restricting the effective decay volume and, consequently, the probability of LLP decays occurring within it. Additionally, the proximity of these trackers to the production point results in substantial background contamination, necessitating stringent selection criteria that inevitably reduce the number of detectable LLP-related events. Another challenge arises from the limitations imposed by current triggering mechanisms, which require tagging of events at the LLP production vertex, often necessitating the presence of a high-p_T lepton, meson, or associated jets. This pre-selection process further curtails the event rate with LLPs and constrains the range of LLP models amenable to investigation. For instance, the main production mode for GeV-scale Heavy Neutral Leptons (N) involves the decay $B \to \ell + N$, where the momentum of the lepton ℓ is insufficient for triggering.

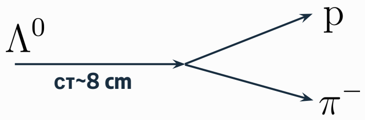
Recognizing these constraints, the scientific community has begun exploring alternative experiments beyond the confines of the LHC detectors [2], encompassing both collider-based setups situated near the LHC and beam dump experiments adopting a displaced decay volume concept. These latter experiments employ an extracted beam

[a] gorkavol@gmail.com
[b] brij.kishor.jashal@cern.ch
[c] valerii.kholoimov@cern.ch
[d] kiselev883@gmail.com
[e] dmendoza@cern.ch
[f] maksym.ovchynnikov@cern.ch
[g] Arantza.Oyanguren@ific.uv.es
[h] volodymyr.svintozelskyi@cern.ch
[i] Jiahui.Zhuo@ific.uv.es

[Eur. Phys. J. C 84, 608 (2024)]

- The main purposes of **Downstream tracking** is to improve the reconstruction efficiencies of decays occurring outside the **VELO** detector:



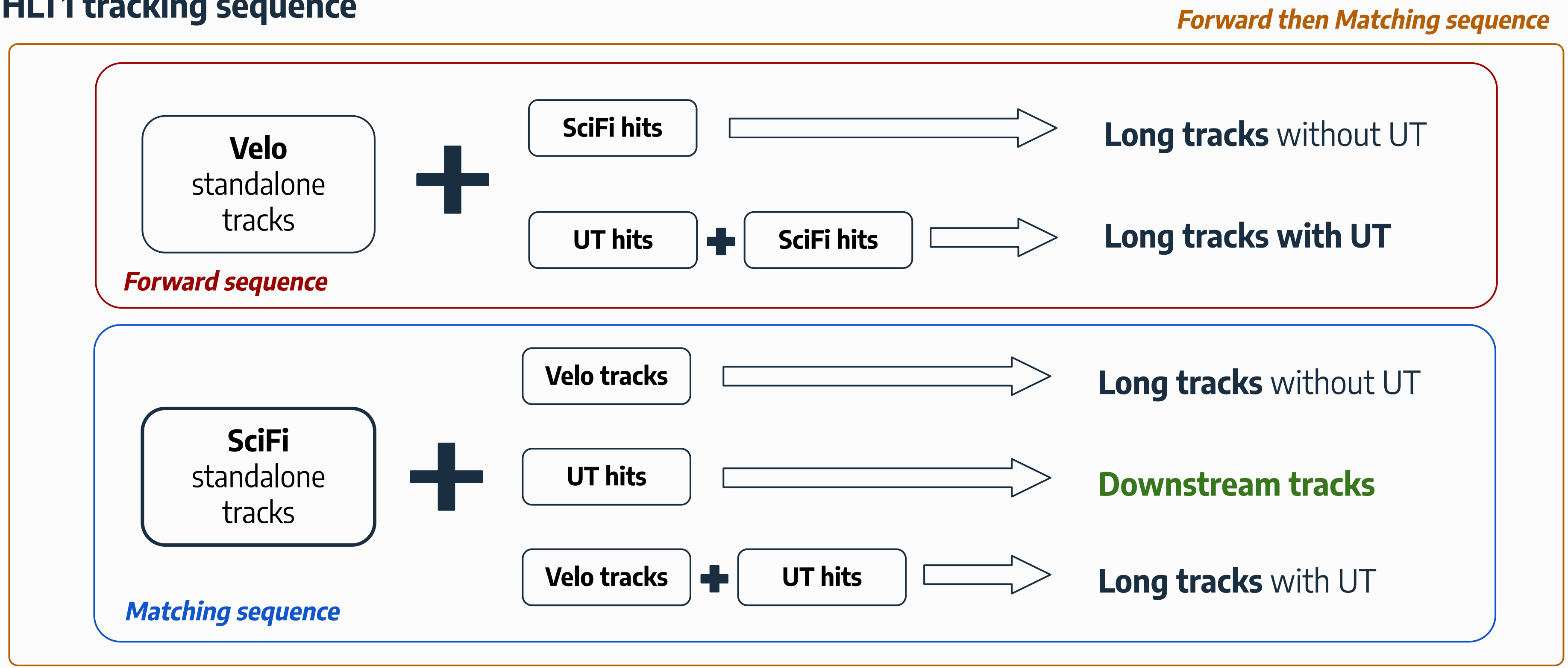


- **Downstream** tracking increase the effective decay volume up to **2.5 m** → More sensitivity for Beyond Standard Model (**BSM**) Long-Lived Particles (**LLPs**) searches.

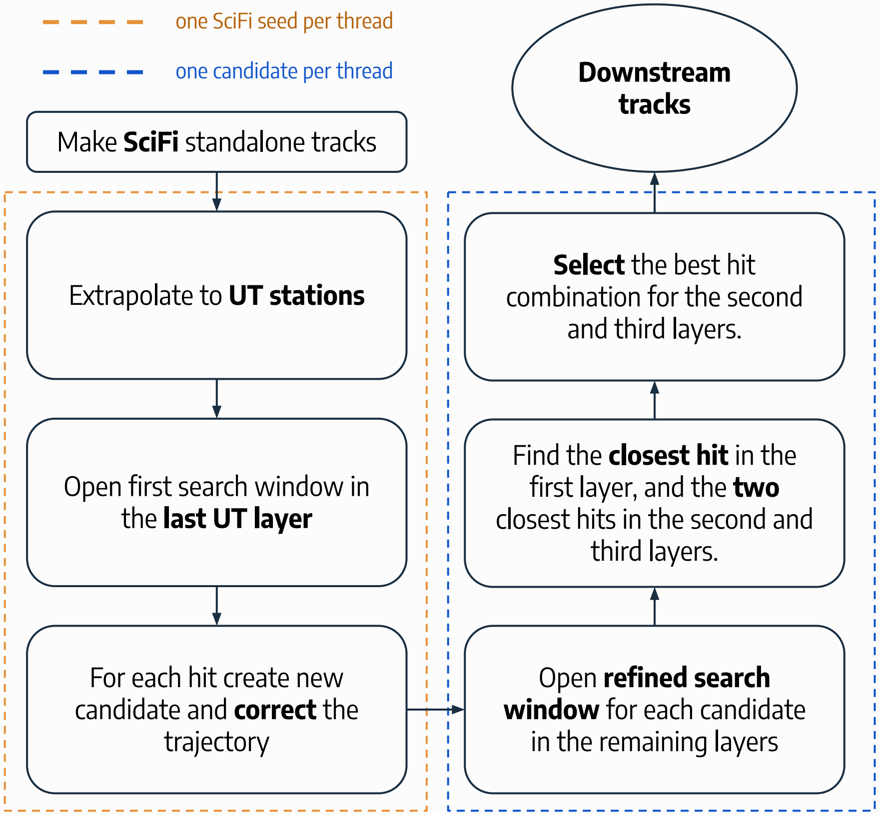
- Downstream tracking **couldn't** be implemented in **HLT1** during **Run 2** due to limit timing budgets, but thanks to the **trigger system upgrade** in **Run 3**, we successfully developed **GPU** version of **Downstream** tracking:
    - This has been running in data-taking since **October 2024**.

HLT1 Downstream tracking

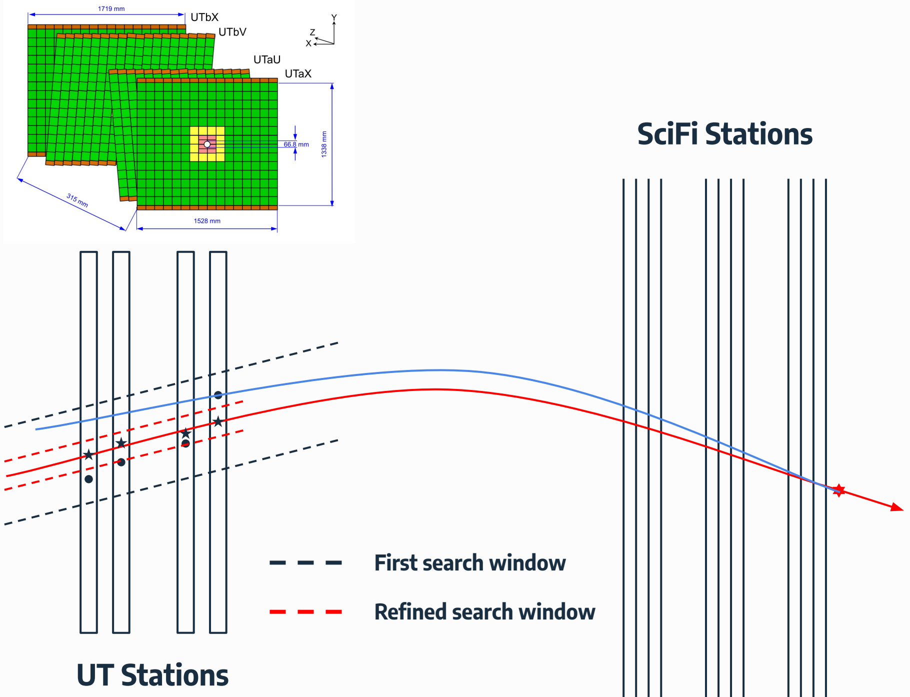
HLT1 tracking sequence



HLT1 Downstream tracking



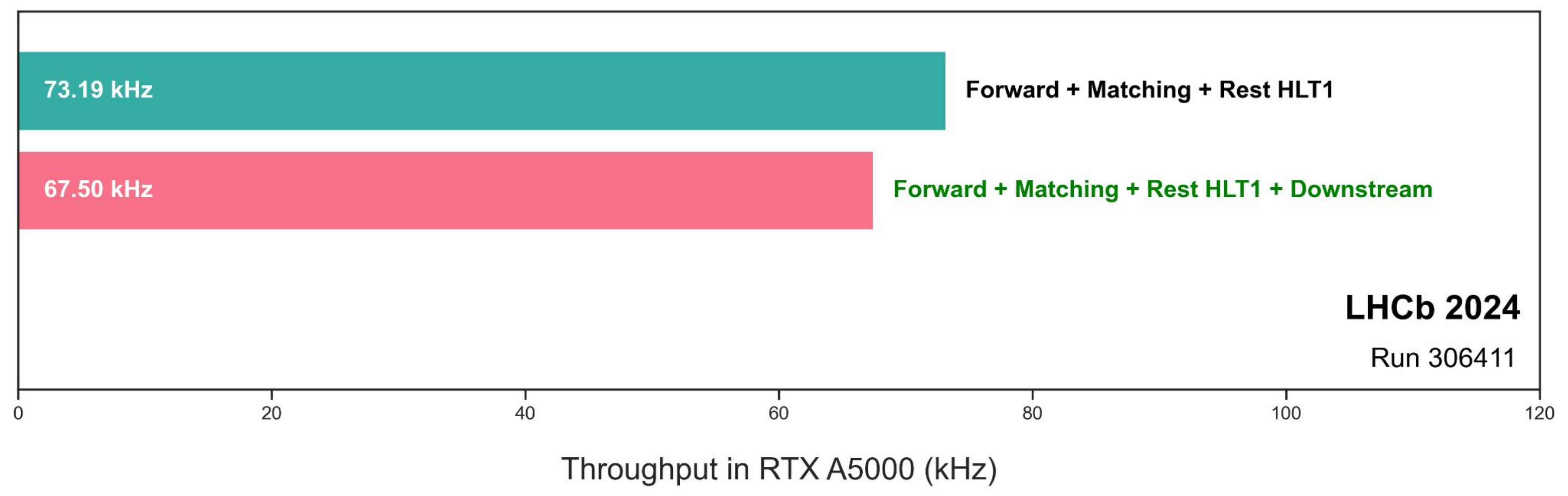
[LHCB-TDR-015]



HLT1 Downstream tracking

Throughput

[LHCB-FIGURE-2024-035]

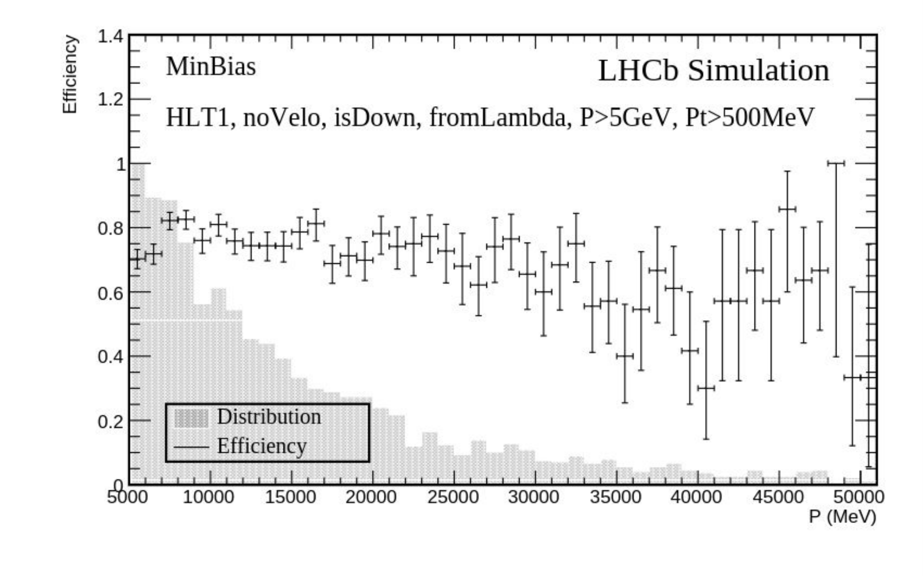


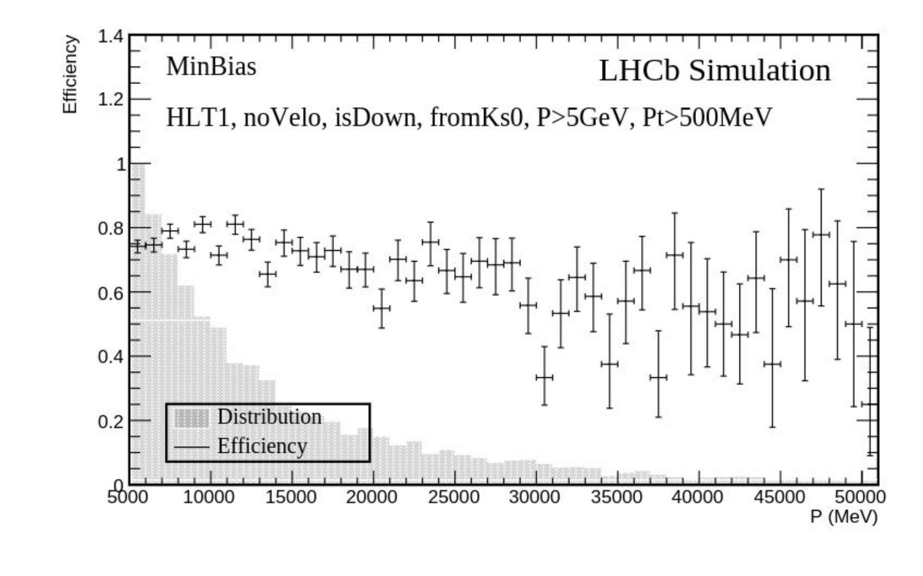
- Adding **Downstream tracking** to HLT1 reduces throughput by approximately **9%** to **67.50 kHz** per GPU, meeting the HLT1 requirement of over **60 kHz** per GPU.

HLT1 Downstream tracking

Tracking efficiency

[LHCB-FIGURE-2023-028]



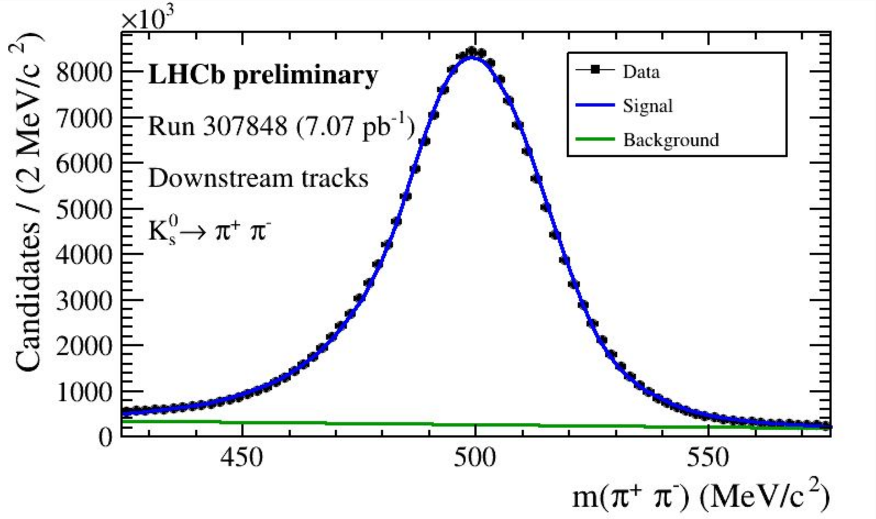


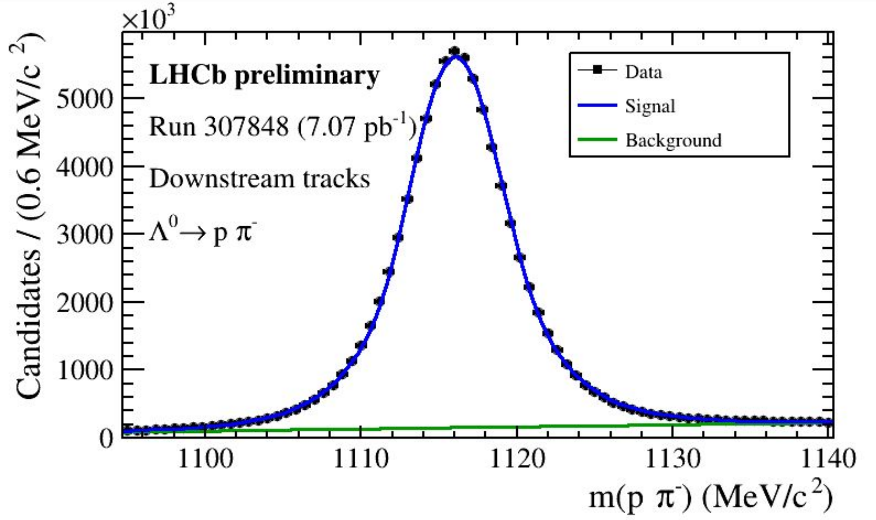
- The **HLT1** Downstream tracking shows an average efficiency of about **75%** for Λ^0 and K^0_s.

HLT1 Downstream tracking

Mass resolution

[LHCB-FIGURE-2024-035]

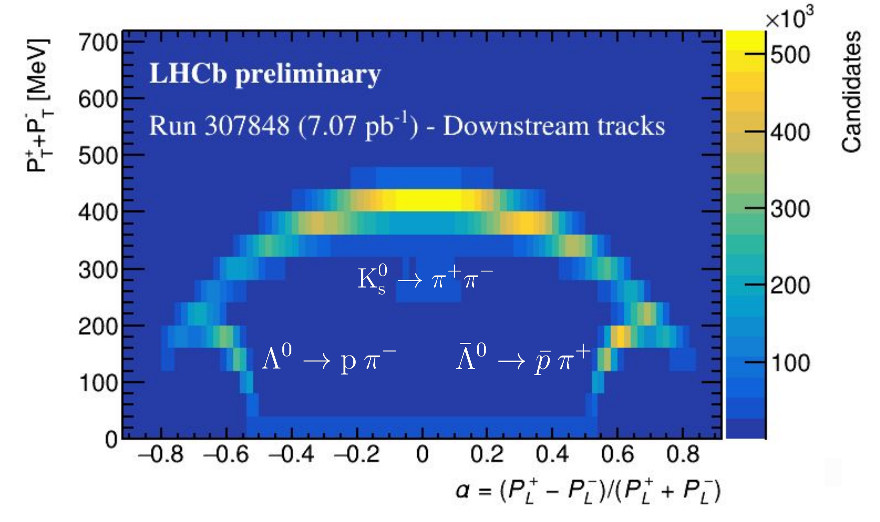




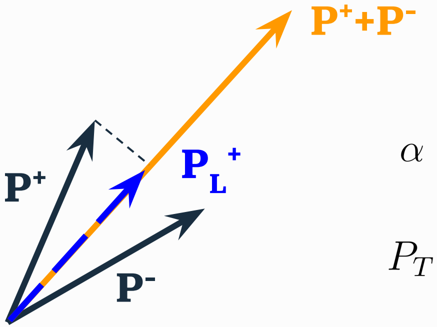
- The mass resolution of HLT1 Downstream tracking is approximately **15.3 MeV/c²** for K^0_s and **3.2 MeV/c²** for Λ^0.

HLT1 Downstream tracking

Armenteros-Podolanski Plot



[LHCB-FIGURE-2024-035]



$$\alpha = \frac{P_L^+ - P_L^-}{P_L^+ + P_L^-}$$

$$P_T = P_T^+ = P_T^-$$

In the (α, PT)-plane, particles from a two body decay define an ellipse:

$$\frac{(\alpha - \alpha_0)^2}{r_\alpha^2} + \frac{P_T^2}{P_{cm}^2} = 1$$

Where:

$$(\alpha_0, 0) = (\frac{m_1^2 - m_2^2}{M^2}, 0) \qquad (r_\alpha, r_{P_T}) = (\frac{2P_{cm}}{M}, P_{cm})$$

center of the ellipse — radii of the ellipse

Summary

- LHCb has **upgraded** its **detector** and **trigger system** for **Run 3**.
- **Downstream tracking** is crucial for reconstructing decays outside the VELO.
- **The HLT1 Downstream tracking** has been taking data since **October 2024.**
- The downstream tracking achieves **~75%** efficiency for Λ^0 and K_s^0, with mass resolutions of ~**3.2 MeV/c²** for Λ^0 and ~**15.3 MeV/c²** for K_s^0.

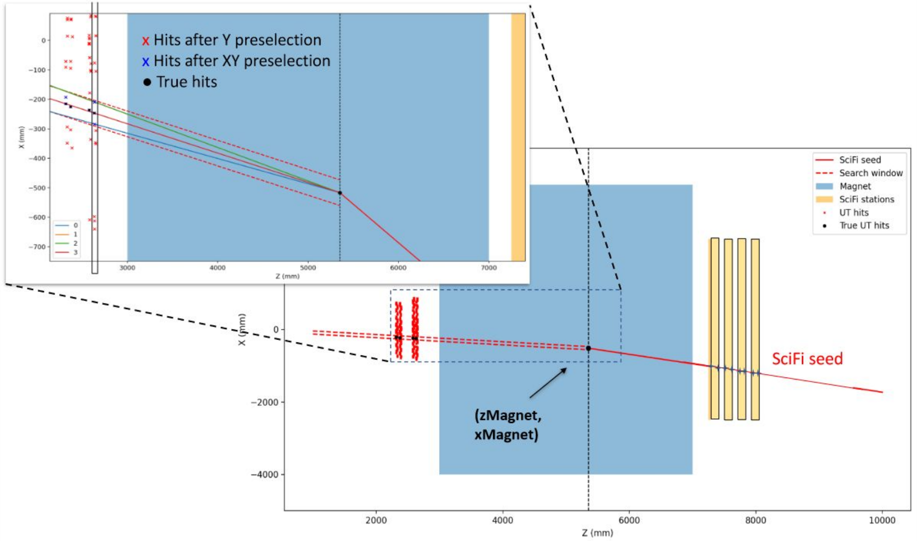
Thanks for listening!

Any questions?

Backup

HLT1 Downstream tracking sequence

SciFi state: $\vec{S}_i = (x, y, t_x, t_y, q/p)^T$



Particle movement through magnet (Kink)

$$z_{\text{Magnet}} = \alpha_0 + \alpha_1 \cdot t_y^2 + \alpha_2 \cdot t_x^2 + \alpha_3 \cdot \frac{q}{p} + \alpha_4 \cdot |x_{\text{SciFi}}| + \alpha_5 \cdot |y_{\text{SciFi}}| + \alpha_6 \cdot |t_y| + \alpha_7 \cdot |t_x|.$$

$$x_{\text{Magnet}} = x_{\text{SciFi}} + t_{x_{\text{SciFi}}} \cdot (z_{\text{Magnet}} - z_{\text{SciFi}}).$$

$$y_{\text{Magnet}} = (y_{\text{SciFi}} + dy) + t_{y_{\text{Magnet}}} \cdot (z_{\text{Magnet}} - z_{\text{SciFi}}).$$

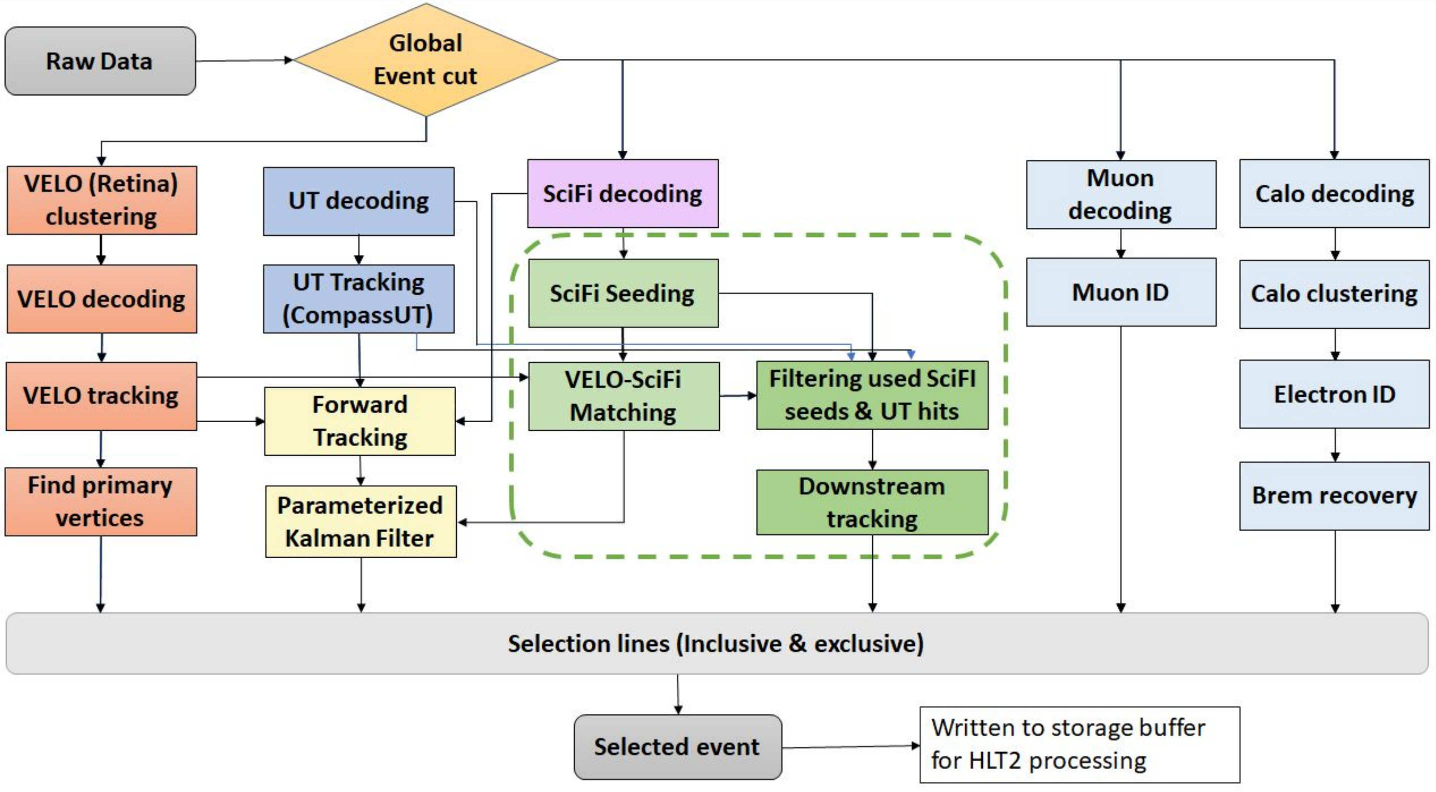
$$t_{y_{\text{Magnet}}} = t_{y_{\text{SciFi}}} + dt_y.$$

dy and dt_y are the special extrapolation corrections In y_{Magnet} since its extracted from stereo tilt

$$dy = \beta_0 + \beta_1 \cdot y_{\text{SciFi}} + \beta_2 \cdot t_{y_{\text{SciFi}}} + \beta_3 \cdot q/p.$$

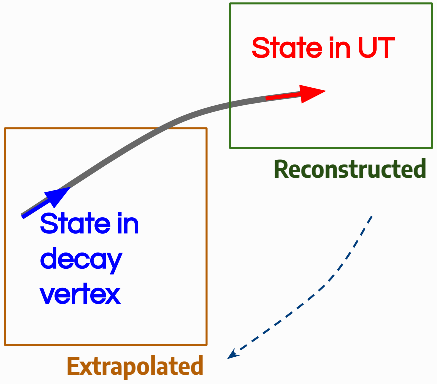
$$dt_y = \gamma_0 + \gamma_1 \cdot y_{\text{SciFi}} + \gamma_2 \cdot t_{y_{\text{SciFi}}} + \gamma_3 \cdot q/p.$$

HLT1 Downstream tracking sequence

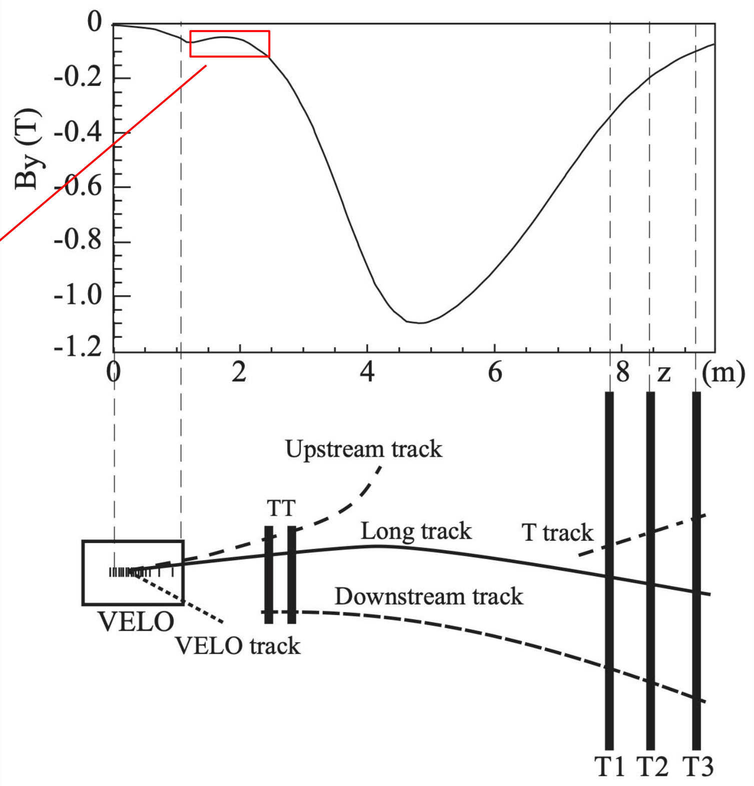


Downstream vertexing

Track extrapolation



Assume constant magnetic field

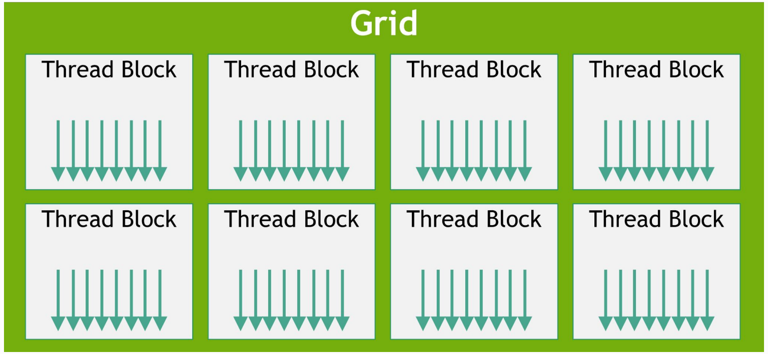


$$x(z) = x_0 + t_x(z - z_{UT}) + \gamma(z - z_{UT})^2$$

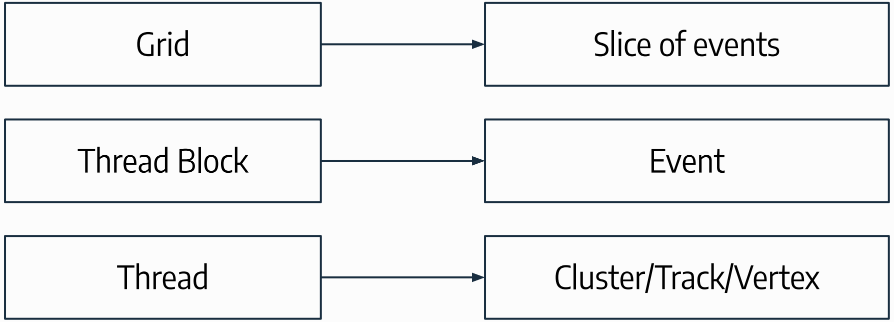
We can parametrize $\gamma = \gamma(\frac{q}{p})$

Allen: A high level trigger on GPUs for LHCb

GPU Programming abstraction



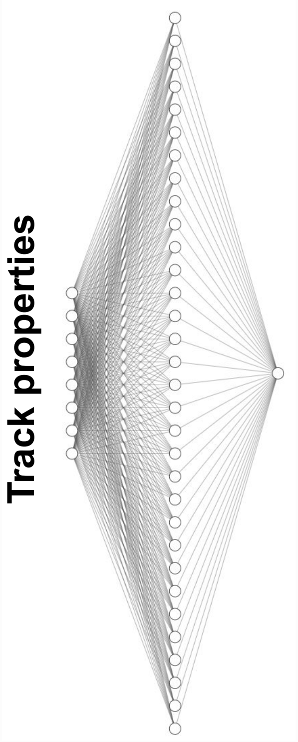
Parallelization mapping

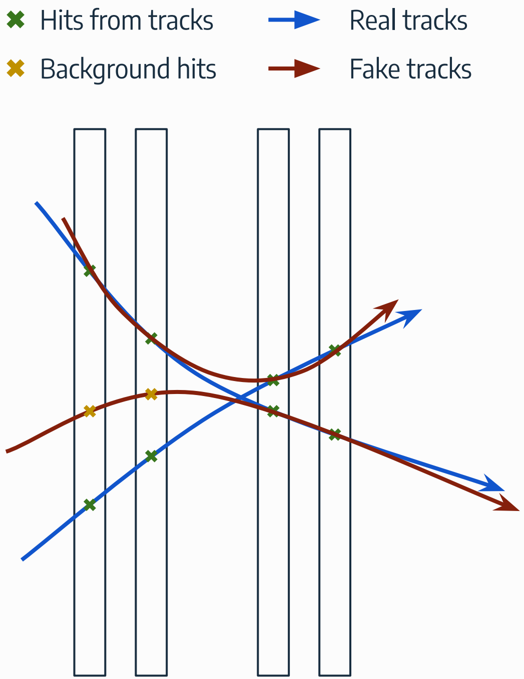


- To take advantage of multi-threading in the GPU, we process data in slices of events (1 slice ~ 1000 events). Each thread block is mapped to a single event in the slice, and each thread within the block works on an independent candidate for reconstruction or selection (e.g., cluster, track, or vertex).
- Threads within the same block can synchronize and share resources through shared memory.
- An RTX A5000 has 64 Streaming Multiprocessors (SMs), each containing 128 CUDA cores, for a total of 8,192 CUDA cores. If each event uses 128 threads, this means we can process 64 events simultaneously, which should significantly improve the throughput of HLT1.

HLT1 Downstream tracking

- A single hidden (**32** nodes) layer Fully Connected Neural Network (**FCNN**) is trained to suppress the fraction of **fake tracks**.
- It utilizes **track properties** as input, and output the probability of a reconstructed track being a fake track.
- The is trained with **minimum bias pp collision simulations**, present great discrimination power in **fake track**





HLT1 Downstream tracking

C++/CUDA TRICKS

STATIC STRUCT

It's not necessary to determine the size of your NN in the runtime.

```cpp
namespace DownstreamGhostKiller {

  namespace Model {
    constexpr unsigned num_node = 14;
    constexpr unsigned num_input = 8;
```

UNWIND FOR-LOOP

Use C++ template to explicitly unroll the for-loop.

```cpp
// First layer
DownstreamHelpers::unwind<0, Model::num_node>([&](int i) {
  DownstreamHelpers::unwind<0, Model::num_input>([&](int j) {
    h1[i] += input[j] * Model::weights1[i][j];
  });
  h1[i] = ActivateFunction::relu(h1[i] + Model::bias1[i]);
});
```

USE FAST MATH FUNCTIONS

Use fast math functions in CUDA.

```cpp
...
namespace ActivateFunction {
  // rectified linear unit
  __device__ inline float relu(const float x) {
    return x > 0 ? x : 0;
  }
  // sigmoid
  __device__ inline float sigmoid(const float x) {
    return __fdividef(1.0f, 1.0f + __expf(-x));
  }
} // namespace ActivateFunction
```

CUDA Architecture