Downstream:

A new algorithm at LHCb to reconstruct Long-Lived particles in the first level of the trigger.

Arantza De Oyanguren Campos, Brij Kishor Jashal, Jiahui Zhuo

Instituto de Física Corpuscular (IFIC), CSIC - Univ. of Valencia

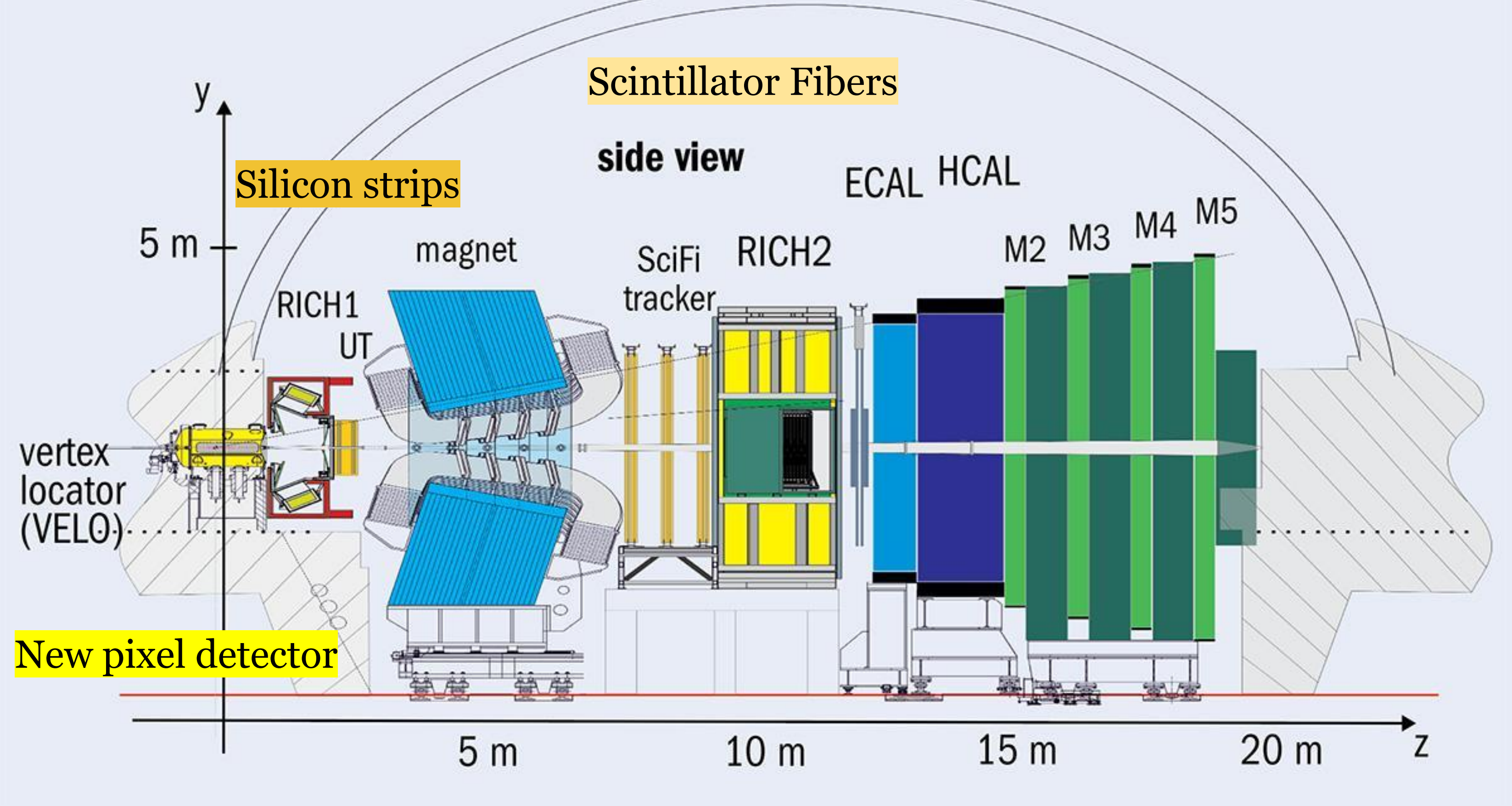
On behalf of LHCb-RTA collaboration

Outline:

- Introduction
- *Downstream* at HLT1 level
- Algorithm design
- Performance
- Summary

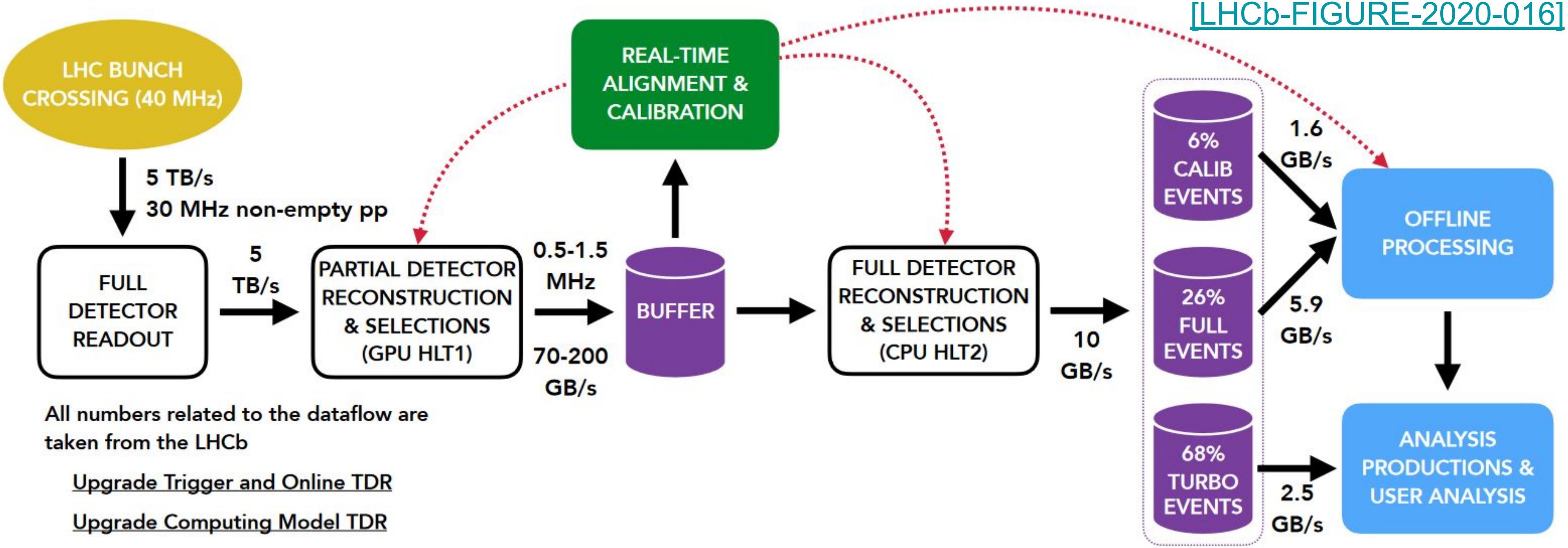
Introduction: the LHCb detector

[CERN Courier: LHCb's momentous metamorphosis]



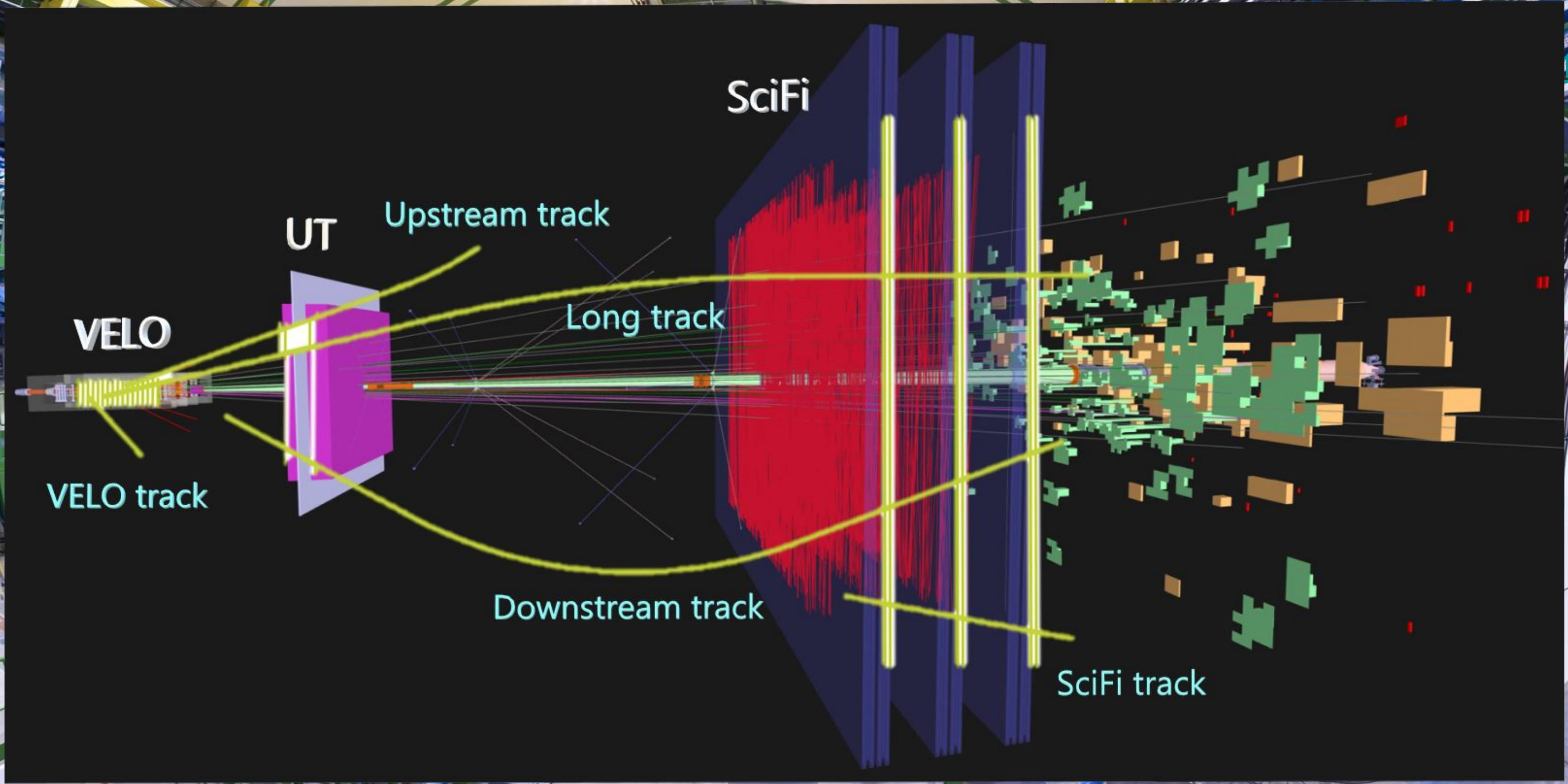
Introduction: the LHCb RTA data flow

[LHCb-FIGURE-2020-016]



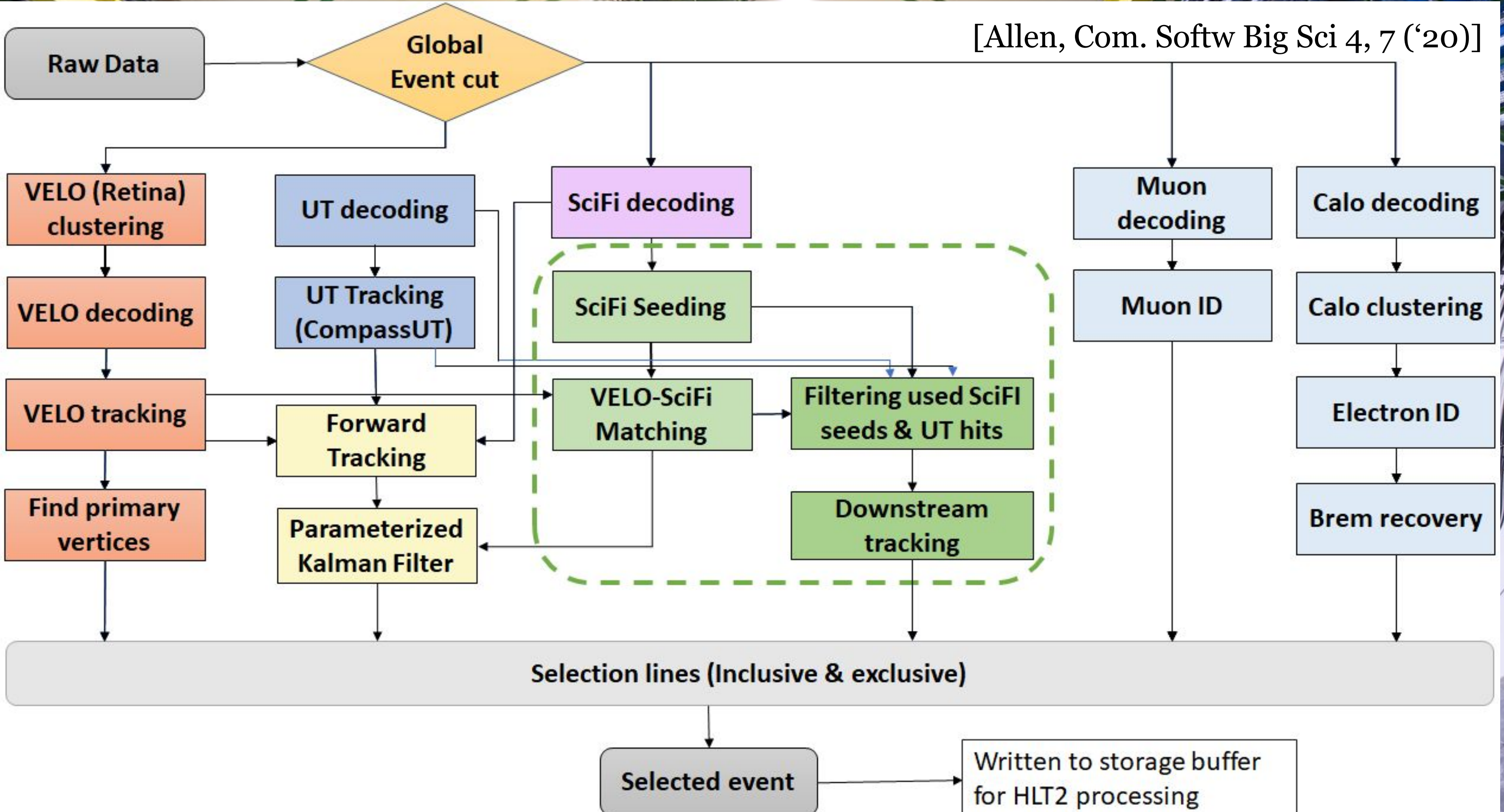
[LHCB-TDR-021]

Introduction: LHCb track types



[6]

Introduction: HLT1 sequence



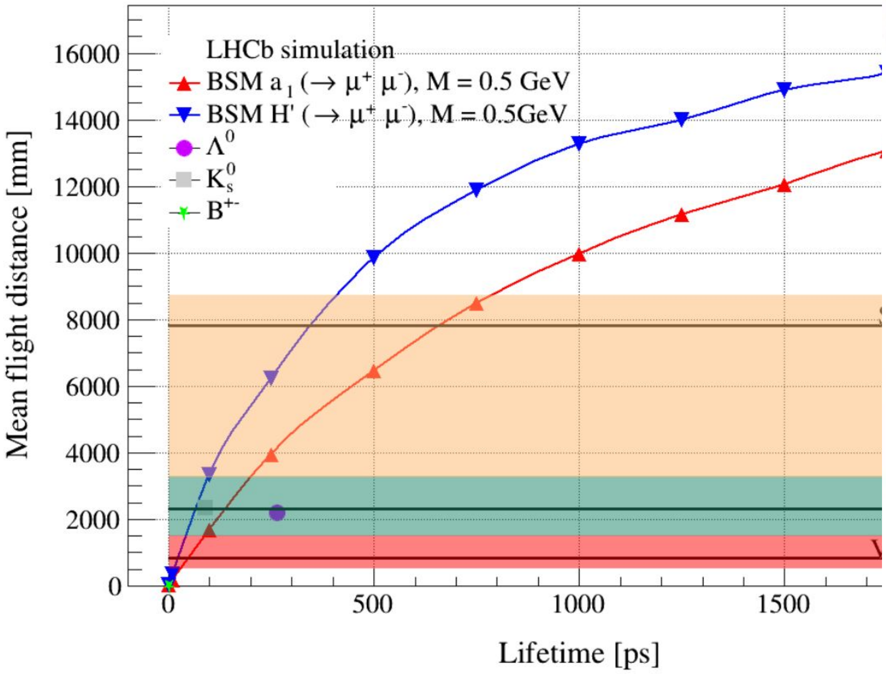
Downstream at HLT1 level (why ?)

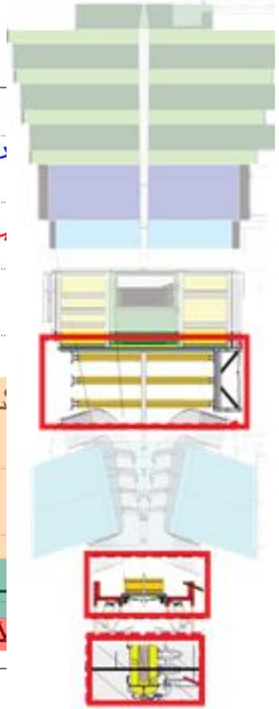
Downstream at HLT1 level (why ?)

- Great LHCb performance for *b*- and *c*-meson decays (*long tracks*)

- But for particles with **τ > 100ps** many decays happen out of the VELO detector:
  produce *downstream* and *T*-tracks

  → **not selected by HLT1 !**





L. Calefice *et al*. (2022). Frontiers in Big Data, 2022.1008737

Downstream at HLT1 level (why ?)

2022.1008737

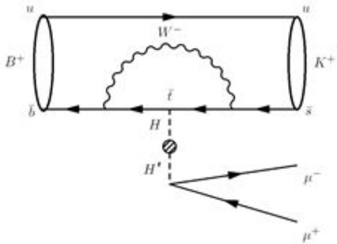
Sensitivity to Λ^0 and K_s^0

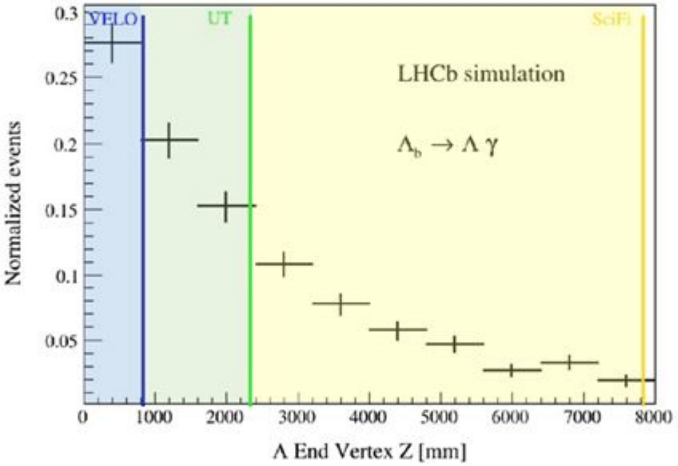
HLT1 reconstruction efficiency (trigger on the signal, TOS)

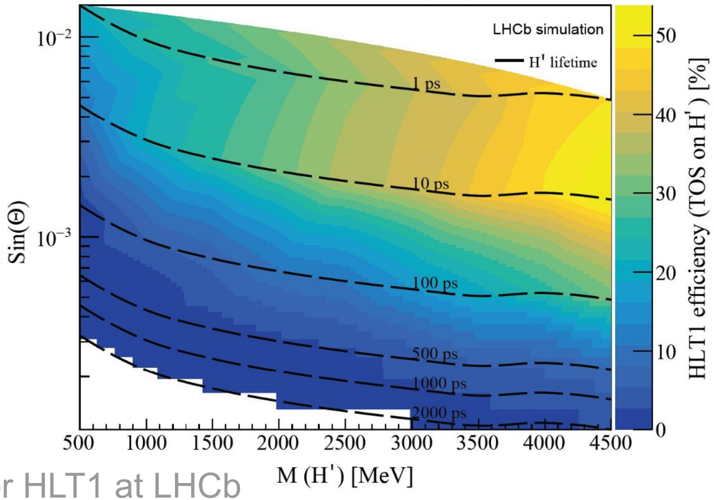
	LL	DD	TT	HLT1 eff (TOS)
Λ^0	12%	51%	37%	< 10%
K_s^0	46 %	38 %	16 %	< 25%

BSM: sensitivity to $B^+ \to K^+ H' [\to \mu^+\mu^-]$

- Decent efficiency (30-50 %) for low lifetime
- Poor efficiency (< 10 %) for τ > 100 ps
- Loss in sensitivity for small H' mass

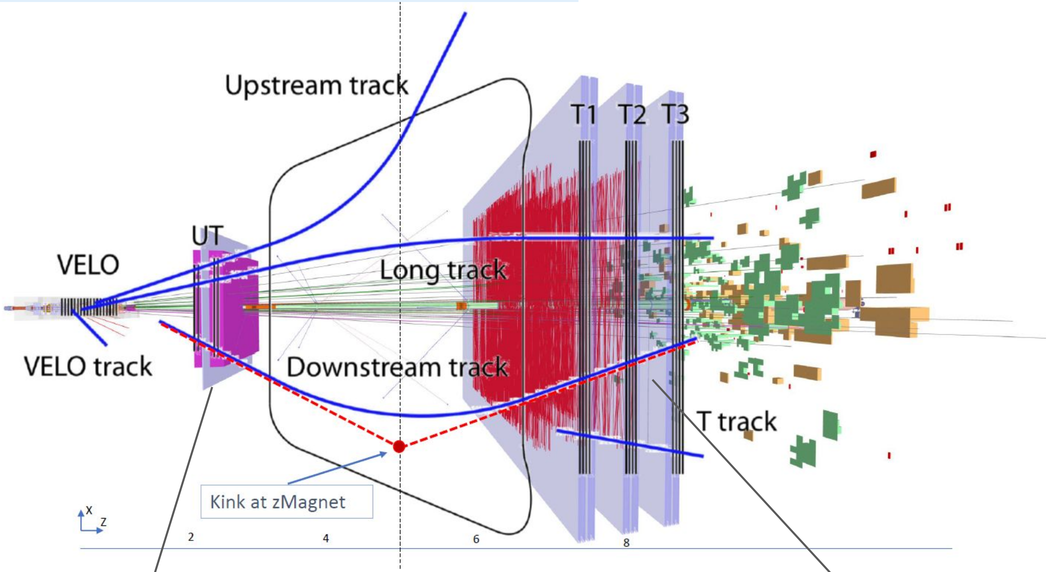


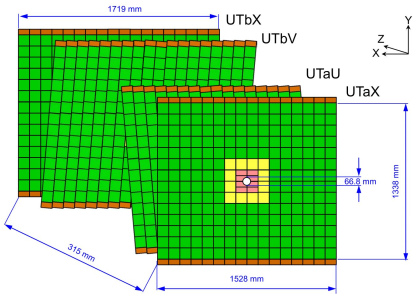


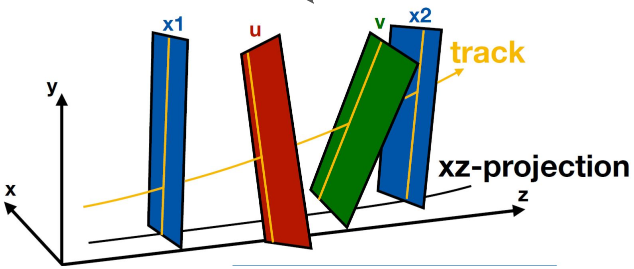


Algorithm design

Algorithm design: strategy





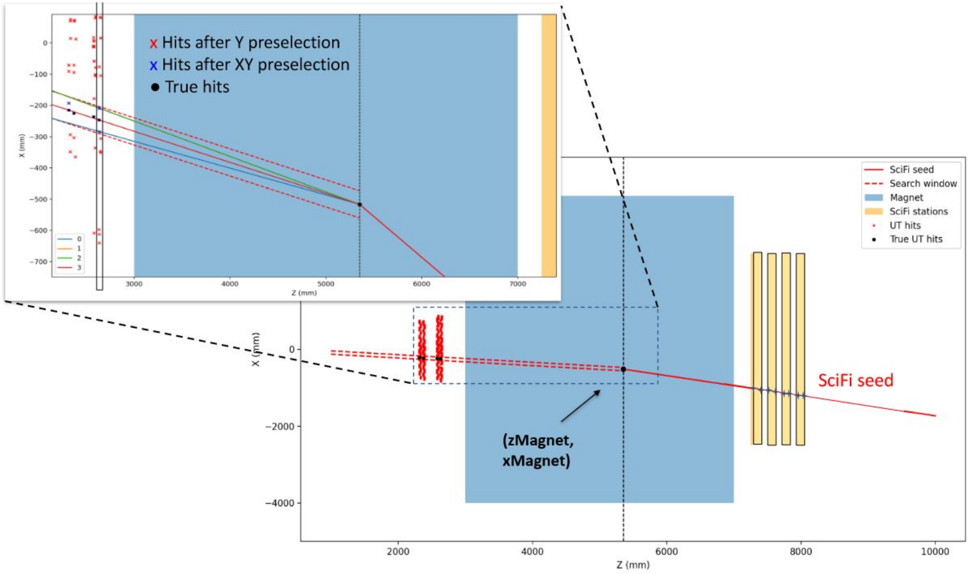


Extrapolate SciFi seeds to UT

- Take the output of SciFi seeding
- Filter out the used seeds
- Extrapolate to UT stations (through *magnet point*)

Algorithm design: track model

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



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

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

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

Algorithm design: track model

First slope estimation

$$\text{First\_}t_{xUT} = \frac{x_{\text{Magnet}}}{z_{\text{Magnet}}} + dt_x,$$

Correction to the first slope

$$dt_x = \alpha_0 + \alpha_1 \cdot t_{y_{\text{SciFi}}} + \alpha_2 \cdot q/p.$$

Expected position at layer_i

$$y_{\text{layer}_i} = y_{\text{Magnet}} + t_y \times (z_{\text{layer}_i} - z_{\text{Magnet}}),$$
$$x_{\text{layer}_i} = x_{\text{Magnet}} + t_x \times (z_{\text{layer}_i} - z_{\text{Magnet}}).$$

(Find UT hits and build UT track)

Tolerances:

For X layers i.e. UTbX and UTaX

$$T(layer_i) = \alpha_0 + \alpha_1 \cdot |q/p|$$

For UV layers i.e. UTbV and UTaU

$$T(layer_i) = \alpha_0 + \alpha_1 \cdot |q/p| + \alpha_2 \cdot |(q/p)^2|$$

Momentum estimation

$$q/p = \frac{\Delta_{\text{slope}}}{\gamma_0 + \gamma_1 \cdot t_x^2 + \gamma_2 \cdot t_y^2} \cdot \text{magnet\_polarity}$$

Algorithm design: flow

Algorithm is divided into 3 main kernel functions:

Kernel function 1:
128 SciFi seeds per thread block

- Filtering used SciFI seeds
- For each input SciFi seed, extrapolate to last x layer (UTbX)
- Store up to 10 best candidates
- Update slope of each candidate using magnet point and hit position.

Kernel function 2:
256 candidates per thread block

- Add hits from rest of the UT layers
- Find best combination of U/V hits
- Compute the scores based on distance b/w extrapolation and real UT hit positions

Kernel function 3:
256 candidates per thread block

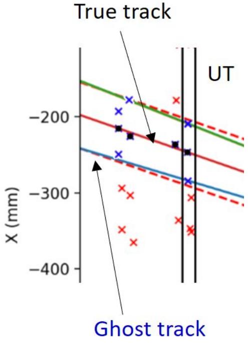
- Find best candidate based on the scores from previous function
- Check for hit duplication
- Perform ghost killing

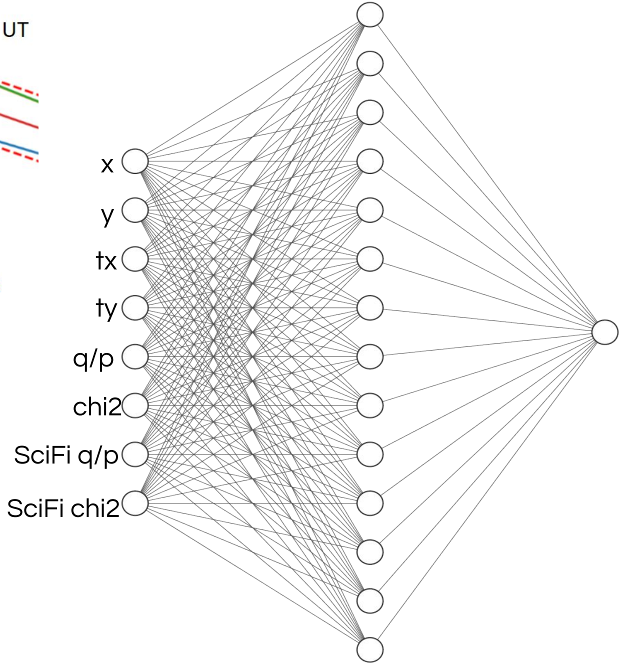
Prepare output:

- Copy hits and tracks to output(compact SOA container)
- Create standard multi-event viewer

Algorithm design: ghost killer NN

- A single hidden (14 nodes) layer fully connected NN
- It utilizes **8 variables** as input:
  - *Downstream* track state ($\mathbf{x,y,t_x,t_y,q/p,}\ \boldsymbol{\chi^2}$)
  - SciFi track properties ($\mathbf{q/p,}\ \boldsymbol{\chi_y^2}$)
- The model was trained using $B_s \to \phi\phi$ events.
- In order to boost speed, certain C++/CUDA tricks are applied, such as using **static structs**, employing **fast math functions,** and **unwinding for-loops**.





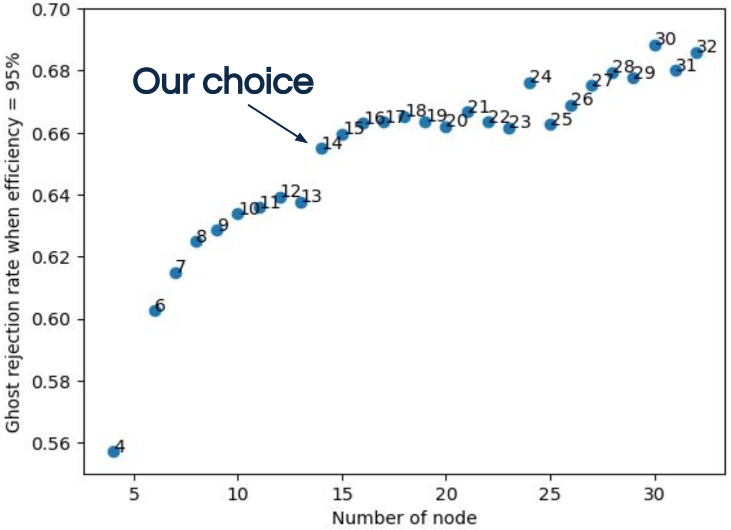

```
namespace DownstreamGhostKiller {

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

```
...
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
```

```
// 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]);
});
```

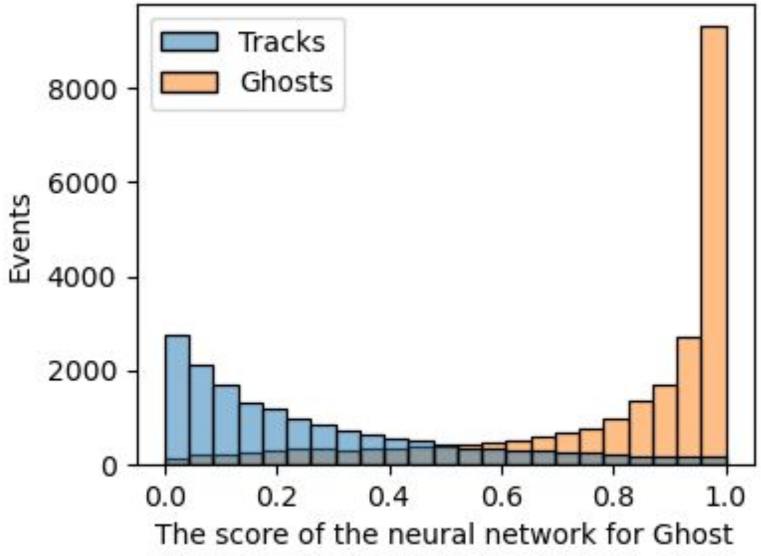
Algorithm design: ghost killer neural network

Ghost rejection vs number of nodes in the hidden layer



Num Operation = Num Input * Num node

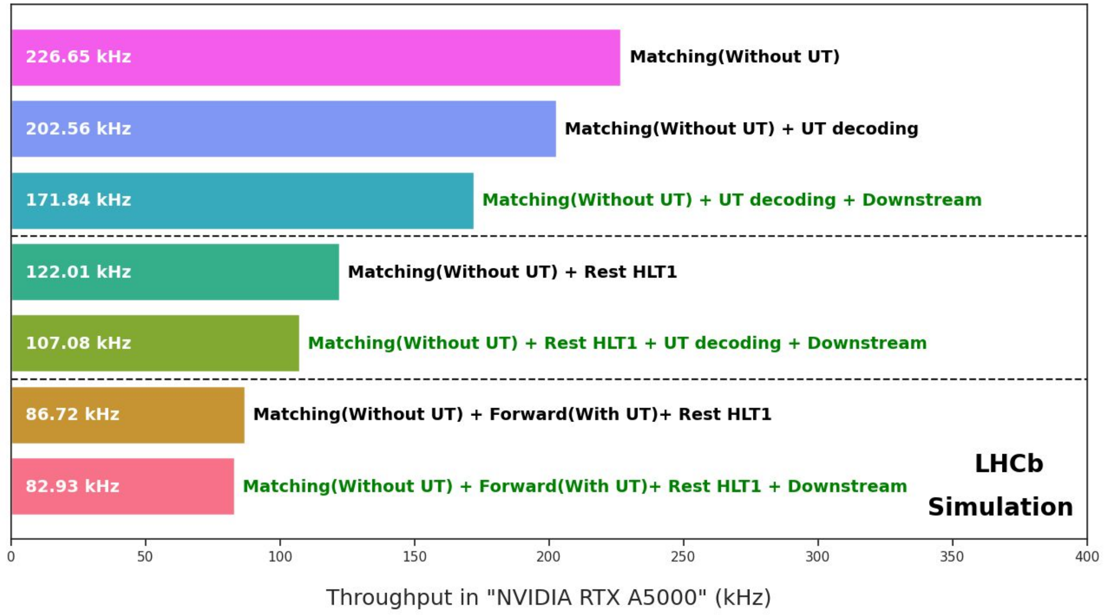
Distribution of the classifier output: default threshold value 0.5



Performance

Throughput: of HLT1 sequences on RTA A5000 [KHz]

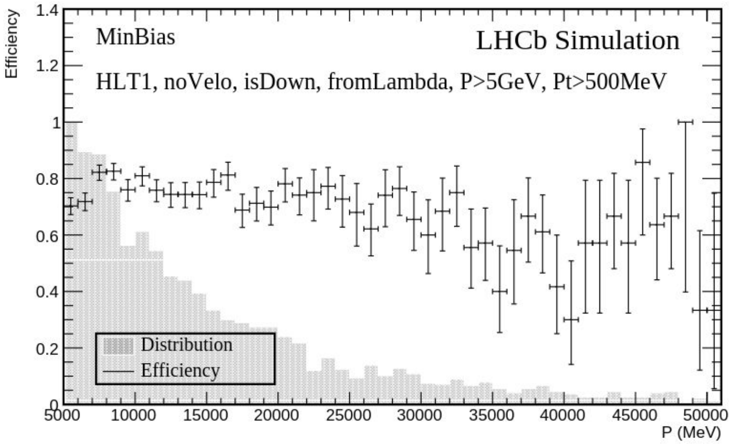
HLT1 sequence for *long* and *downstream* tracking reconstruction:

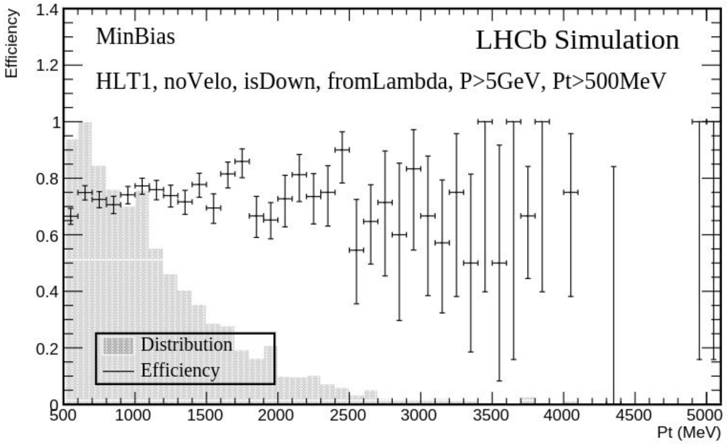


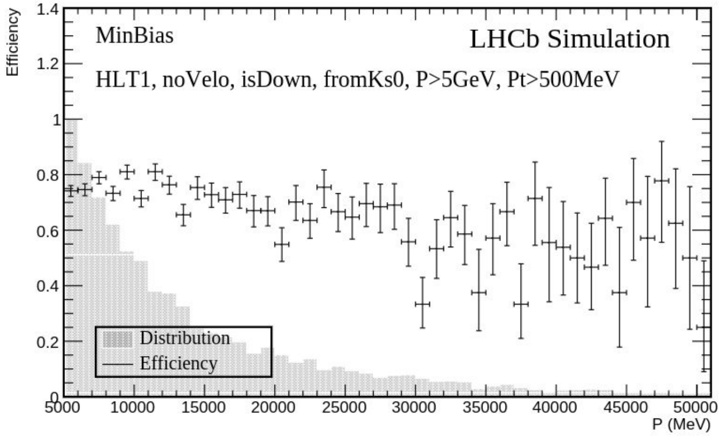
LHCb-FIGURE-2023-028

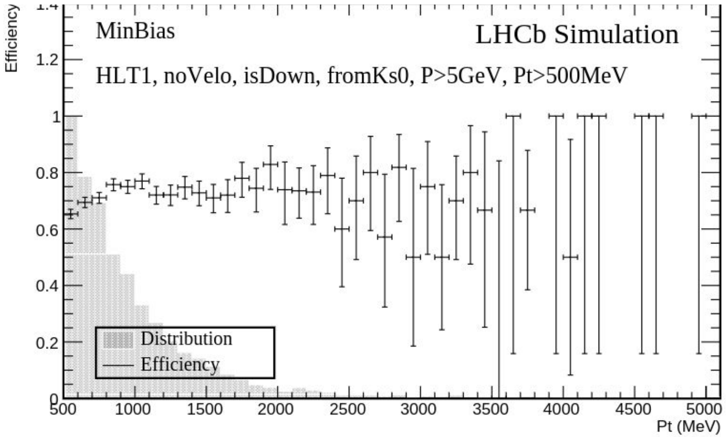
Efficiency

Λ and K_s from from MinBias (p and p_t)





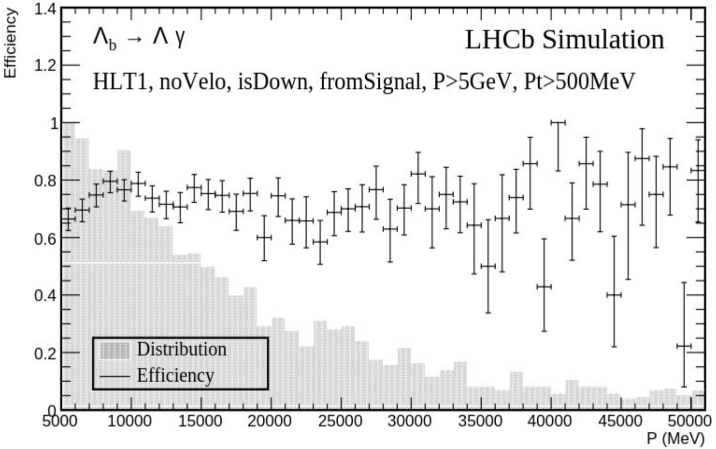


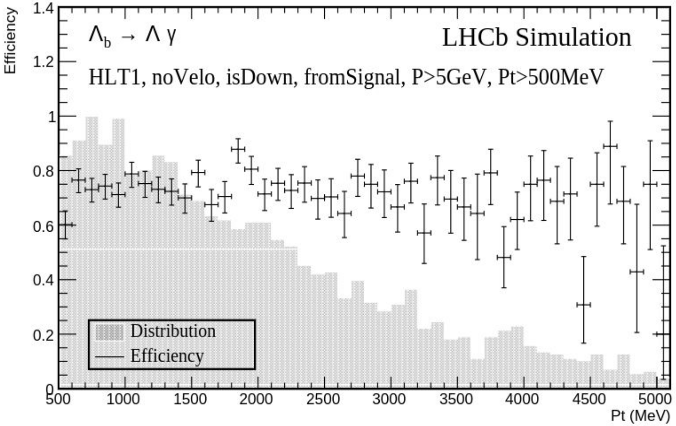


Efficiency

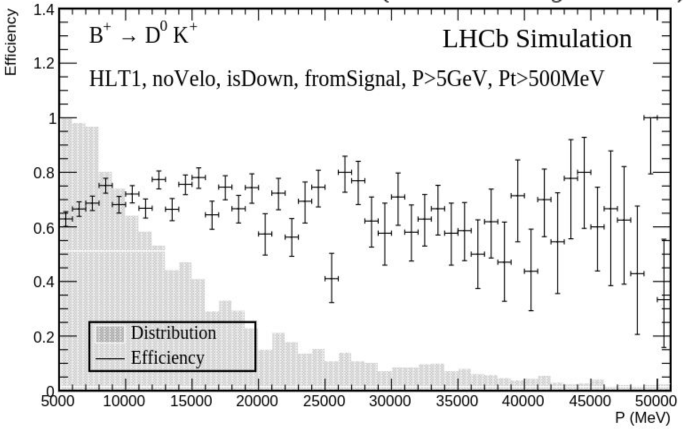
LHCb-FIGURE-2023-028

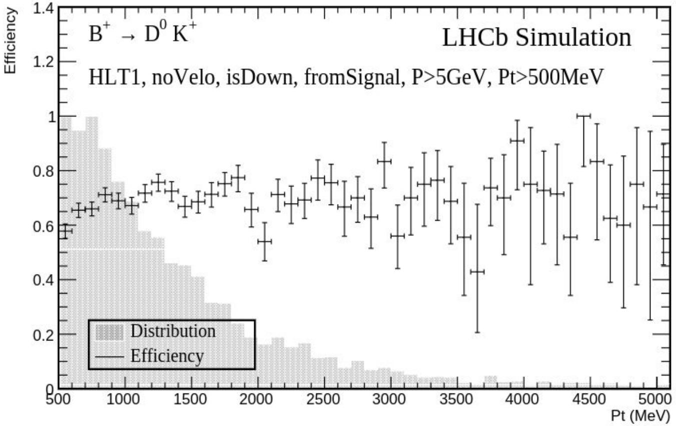
$$\Lambda_b \to \Lambda^0\gamma$$





$$B^+ \to (D^0 \to K_s^0\pi^+\pi^-)K^+$$

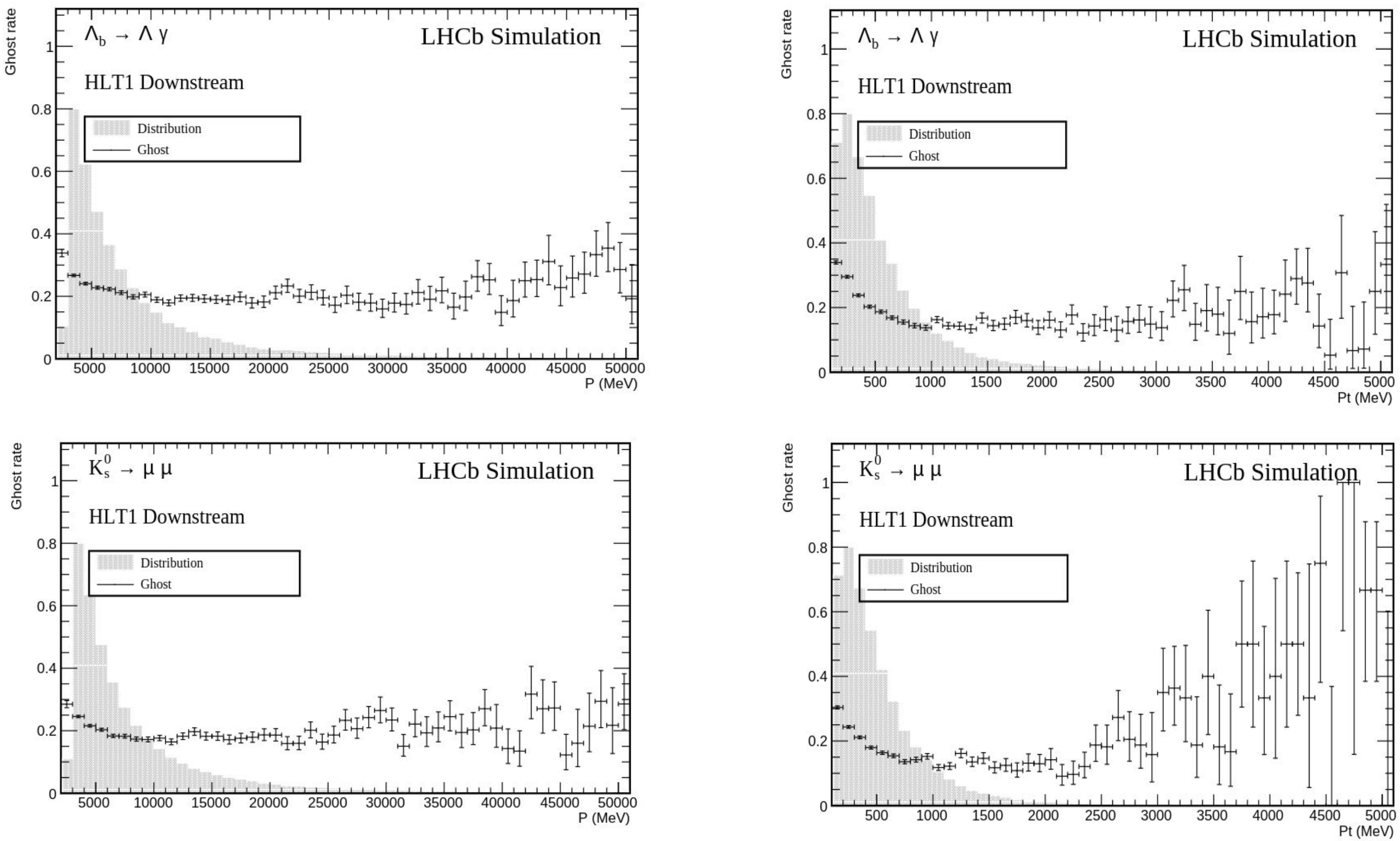




Ghost rates

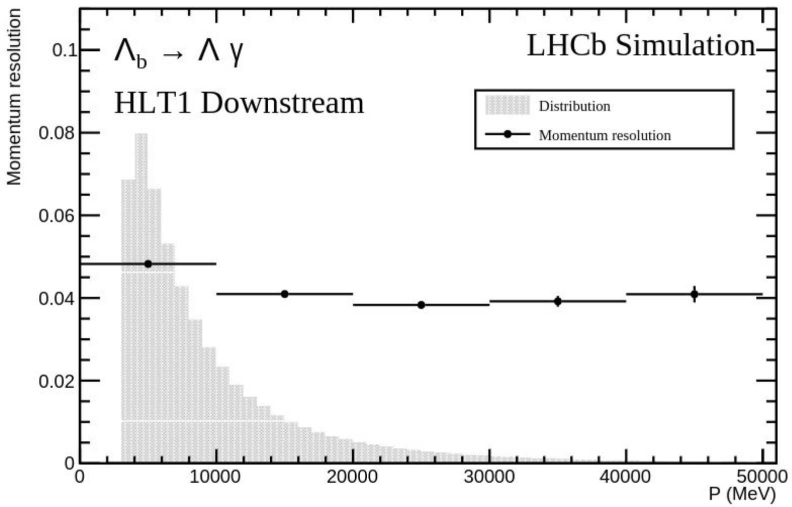
LHCb-FIGURE-2023-028

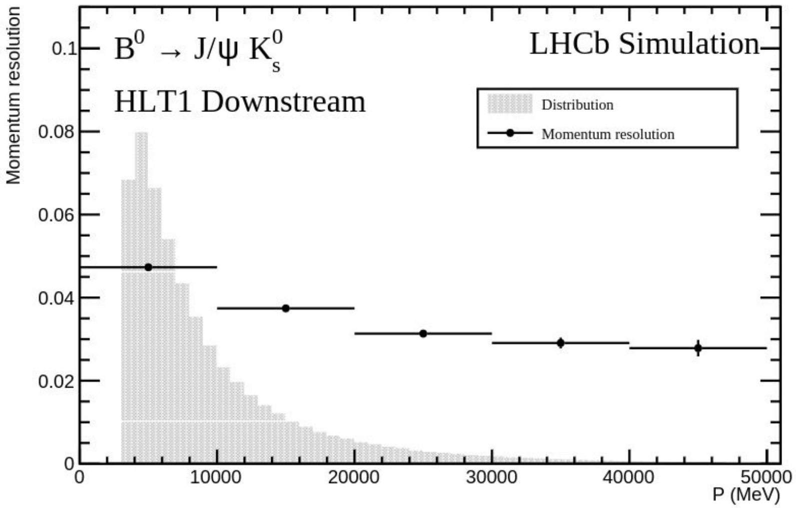
Ghost rates after NN rejection



Momentum resolution

Momentum resolution: **below 6% !**

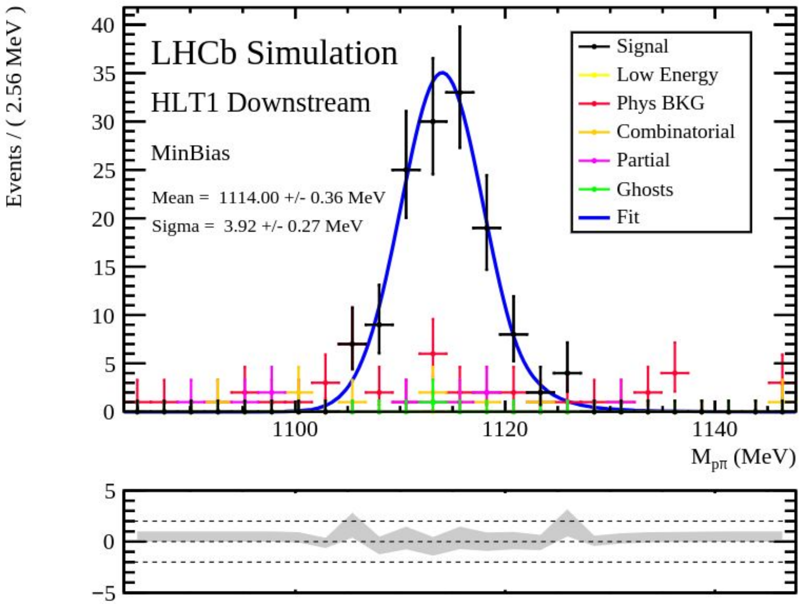




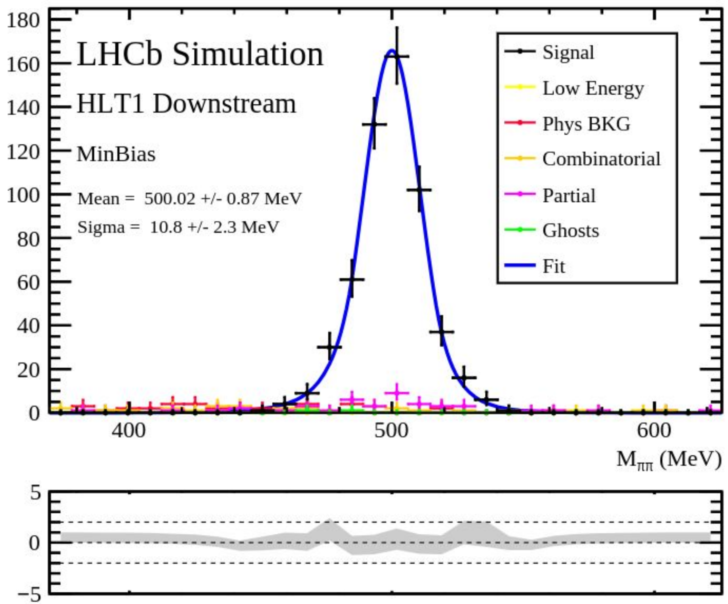
Selection lines using *Downstream* at HLT1 (Run3 conditions)

Selection lines output (after vertexing two *downstream* tracks):

$\Lambda(\rightarrow p\pi^-)$



$K^0_S(\rightarrow \pi^+\pi^-)$



LHCb-FIGURE-2023-028

Summary

- First implementation of ***Downstream*** track reconstruction at HLT1!
- Very good throughput on A5000:
  Sequence with (without) *Downstream* **87** (83) KHz
- Physics efficiency around 75% and independent of the physics channel.
- First implementation of a NN at HLT1 for ghost killer: used for selection line development.
- Expected huge impact on physics!

Channel	DD/LL proportion	Interest
b-hadron decays		
$\Lambda_b^0 \to \Lambda\gamma$	3.4	γ polarization, BR
$\Xi_b^- \to \Xi^-\gamma$	25	γ polarization, BR
$\Omega_b^- \to \Omega^-\gamma$	13	γ polariation, BR
$B^+ \to K_S^0 K_S^0 \pi^+$	2.8	CPV, BR
$B^+ \to K_S^0 K_S^0 K^+$	2.7	CPV, BR
$B_s^0 \to K_S^0 K_S^0$	3.6	CPV, BR
Charm physics		
$\Lambda c^+ \to \Lambda K^+$	4.4	Polarization studies
$\Xi_c^- \to \Xi^- \pi^-$	8.4	Polarization studies
$D^0 \to K_S^0 K_S^0$	1.8	CPV
$J/\psi \to \Lambda\bar{\Lambda}$	4.8	Polarization studies, BR
Strange physics		
$K_S^0 \to \mu^+\mu^-$	0.6	BR
$K_S^0 \to \mu^+\mu^-\mu^+\mu^-$	0.8	BR
$K_S^0 \to \gamma\mu^+\mu^-$	0.8	BR

Thank you!