IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 May 2024, accepted 7 August 2024, date of publication 12 August 2024, date of current version 27 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3442573

RESEARCH ARTICLE

Looking Forward: A High-Throughput Track Following Algorithm for Parallel Architectures

AURELIEN BAILLY-REYRE[1], LINGZHU BIAN[2], PIERRE BILLOIR[1], DANIEL HUGO CÁMPORA PÉREZ[3], VLADIMIR VAVA GLIGOROV[1,4], FLAVIO PISANI[4], RENATO QUAGLIANI[1,4,5], ALESSANDRO SCARABOTTO[1,6], AND DOROTHEA VOM BRUCH[7]

[1]LPNHE, Paris Diderot Sorbonne Paris Cité, CNRS/IN2P3, Sorbonne Université, 75005 Paris, France
[2]School of Physics and Technology, Wuhan University, Wuhan 430079, China
[3]Universiteit Maastricht, 6211 LK Maastricht, The Netherlands
[4]European Organization for Nuclear Research (CERN), 1211 Geneva, Switzerland
[5]Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
[6]Fakultät Physik, Technische Universität (TU) Dortmund, 44221 Dortmund, Germany
[7]CNRS/IN2P3, CPPM, Aix Marseille University, 13007 Marseille, France

Corresponding authors: Daniel Hugo Cámpora Pérez (dcampora@cern.ch), Vladimir Vava Gligorov (vladimir.gligorov@cern.ch), Renato Quagliani (renato.quagliani@cern.ch), and Alessandro Scarabotto (alessandro.scarabotto@cern.ch)

The work of Vladimir Vava Gligorov and Alessandro Scarabotto was supported by European Research Council (RECEPT) under Grant 724777. The work of Dorothea Vom Bruch was supported by European Research Council Starting under Grant ALPACA 101040710.

ABSTRACT Real-time data processing is a central aspect of particle physics experiments with high requirements on computing resources. The LHCb (Large Hadron Collider beauty) experiment must cope with the 30 million proton-proton bunches collision per second rate of the Large Hadron Collider (LHC), producing 10^9 particles/s. The large input data rate of 32 Tb/s needs to be processed in real time by the LHCb trigger system, which includes both reconstruction and selection algorithms to reduce the number of saved events. The trigger system is implemented in two stages and deployed in a custom data centre.

We present Looking Forward, a high-throughput track following algorithm designed for the first stage of the LHCb trigger and optimised for GPUs. The algorithm focuses on the reconstruction of particles traversing the whole LHCb detector and is developed to obtain the best physics performance while respecting the throughput limitations of the trigger. The physics and computing performances are discussed and validated with simulated samples.

INDEX TERMS CUDA, GPU, track reconstruction, particle tracking, parallel programming.

I. INTRODUCTION

The real-time or near-real-time reconstruction of charged particle trajectories (tracking) has been a central element of detectors at hadron colliders since the UA1 experiment at CERN (1981-1990) [1]. The rate and complexity of particle collisions have increased over the past decades and so have the computational demands placed on real-time tracking algorithms. This motivates continued research into high-throughput tracking algorithms which can efficiently exploit modern parallel computing architectures. Tracking algorithms consist in associating together hits left by charged particles in the detector to form tracks. These hits are then fitted to a track model in order to extract kinematic and geometric properties. Reconstruction algorithms are generally one of the most time-consuming components of data processing pipelines, which reconstruct physics quantities of interest with the highest possible precision and fidelity compared to the already recorded data to permanent storage. The need to reduce the computational cost and

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

energy consumption of offline processing therefore motivates research into efficient tracking algorithms. In the past, experiments often deployed special real-time algorithms tuned to specific real-time processing architectures [2], [3], [4] which were decoupled from the offline algorithms and architectures. Increasingly, however, both real-time and offline processing is carried out in data centres populated by heterogeneous computing architectures. It is therefore important in both cases to carefully benchmark, in as general a manner as possible, the tradeoffs between physics performance and computational scalability for different architectures.

In this paper we present "Looking Forward", a high-throughput algorithm for reconstructing charged tracks in the LHCb [5] detector during Run 3 of the Large Hadron Collider (LHC) which is taking place from 2022 to 2025 inclusive, achieving an instantaneous luminosity of $\mathcal{L} = 2 \times 10^{33} cm^{-2} s^{-1}$, five times larger compared to Run 2. The computing resources of LHCb require that the input data rate of 32 Tb/s is reduced to 10 GB/s using selections which are primarily based on the properties of reconstructed tracks. In order to achieve this goal, LHCb uses a two-stage real-time processing pipeline deployed in a custom data centre [6]. The first stage is executed entirely on GPU processors, after which the data is buffered and the best fidelity detector alignment and calibration are deployed. Subsequently, a second stage is executed entirely on CPU processors. This division of labour between CPU and GPU processors has been optimised [5] for Run 3 conditions, but may evolve in the future. The Looking Forward algorithm is therefore designed for deployment on both GPU and CPU architectures. The scope of the algorithm is to reconstruct tracks traversing the whole LHCb detector which are fundamental when triggering on interesting events at LHCb. These tracks must be reconstructed by the algorithm at a speed-level of few microseconds per event on a GPU architecture, which is orders of magnitude faster compared to the current state-of-the-art techniques which aim to a reconstruction at the millisecond-level.

II. CONTEXT

The Run 3 LHCb detector [5] is a single-arm spectrometer optimised for the study of heavy flavour hadrons, primarily instrumented in the pseudorapidity range $2 < \eta < 5$. The layout of LHCb's tracking detectors and the corresponding dipole magnetic field intensity profile are shown in Figure 1. The tracking detectors consist of a silicon pixel vertex locator (VELO), located around the original proton-proton (pp) collisions region, the silicon-strip Upstream Tracker (UT), located upstream of the magnet, and the Scintillating Fibre tracker (FT or SciFi), located downstream of the dipole magnet. The tracks in the VELO are used to reconstruct the position of the pp collisions as well as to discriminate between tracks originating in those collisions from tracks originating in the decays of other long-lived particles products. The UT and FT subdetectors are built from quadruplets of stations, in which each quadruplet is arranged in an **x-u-v-x** layout. The two **x** layers are oriented in the **x-y** plane with vertical silicon strips and fibres in UT and FT, respectively. The **u** and **v** layers are oriented in the **x-y** plane with strips (fibres) placed at ±5 degrees from vertical, which allows the three-dimensional position of the track to be reconstructed. The UT consists of one such quadruplet while the FT consists of three, referred as T1-T3 or collectively as the T-stations. In addition to tracking the LHCb detector is instrumented with particle identification subdetectors, which are not central to the algorithm presented in this paper.

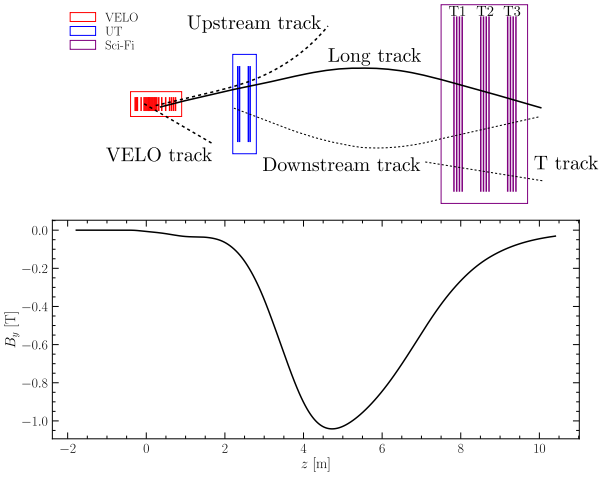

FIGURE 1. The geometric layout of LHCb's tracking system in the bending plane (x-z), shown in a right-handed coordinate system with the z axis running from left to right. The top panel shows the different tracking detectors, described in the text, and the different track types reconstructed by LHCb's pattern recognition algorithms. The bottom panel shows the strength of the main component of LHCb's dipole magnetic field, orthogonal to the bending plane, as a function of the z position. The dipole magnet is located between the UT and T1-T3 stations in this picture.

During Run 3 the LHC will provide up to 30 million colliding bunch crossings ("events") per second, each of which will contain an average of five individual pp collisions in nominal LHCb pp data-taking. This leads to an average of around 60 charged particles which traverse the VELO and the FT in each event. The LHCb trigger system reduces the input data rate by a combination of selections, isolating the events containing processes of interest by computing high-level quantities of interest to physics analysis in real time. This allows up to 90% of data to be discarded even for selected bunch crossings [5]. The majority of LHCb's physics analyses concern the decays of hadrons containing beauty or charm quarks. Because these hadrons have significant lifetimes, they typically travel around 1 cm in the laboratory frame before decaying. In addition, these hadrons produce decay products which have significant momentum transverse to the LHC beamline direction (p_T). It is this combination of displacement from the primary pp collision **and** significant p_T which allows the bulk of LHCb's physics to be separated from generic Quantum Chromodynamic (QCD) processes occurring in LHC pp collisions. Of the track types shown in Figure 1 mainly long tracks allow both displacement and p_T to

be precisely reconstructed, therefore, they form the backbone of both physics analyses and real-time selections at LHCb.

There are two strategies for reconstructing long tracks within the LHCb geometry. The first, known as "forward" tracking [7], begins by reconstructing tracks in the VELO and subsequently extrapolates them through the UT and magnet region to the T-stations. The second, known as "matching" [8], independently reconstructs tracks in the VELO and FT and subsequently matches them to each other, while the UT hits are used to improve track fit and help discriminating between correct and fake matches. The Looking Forward algorithm described in this paper follows the first strategy, but contains significant conceptual and practical modifications compared to previous LHCb forward algorithms [7], [9]. These techniques heavily rely on sequentially handling the tracks and extrapolate them inside the LHCb magnetic field. The superior method presented in this paper reduces the algorithmic complexity by handling in parallel the tracks and performing a simultaneous extrapolation.

The LHCb real-time tracking challenge in Run 3 can be put in a wider context by comparing it to the real-time tracking of the general-purpose LHC detectors: ATLAS and CMS. These aim to take data with around 60 *pp* collisions per bunch crossing in Run 3 [10], [11], each of which produces an average of around 15 charged particles in the detector acceptance. However, their trigger systems are only allowing to reconstruct tracks in real-time at around 100 kHz, once interesting *pp* collisions have been selected using information from calorimeters and muon detectors. These interesting collisions contain a greater than average number of tracks per event compared to LHCb [12], [13], nevertheless the overall number of tracks which the Run 3 ATLAS and CMS real-time systems have to reconstruct per second is around one order of magnitude smaller than it is for LHCb. The ALICE detector will reconstruct thousands [14] of charged particles in lead-lead collisions at 50 kHz in Run 3, again several factors smaller in terms of tracks per second than LHCb. The comparison of the number of reconstructed tracks per second for the four major experiments at the LHC is shown in Table 1. The real-time reconstruction of tracks in the LHCb detector is one of the biggest tracking challenges ever attempted in high-energy physics.

The future high-luminosity upgrades of ATLAS [15] and CMS [16] will operate real-time tracking at far higher rates and multiplicities, as will a planned future upgrade [17] of the LHCb detector. The study provided in this document will provide an insight for the future tracking challenges.

III. REQUIREMENTS

The first stage of LHCb's real-time processing is implemented [18] using A5000 NVIDIA GPUs hosted in up to 190 dual-socket servers with 32 physical cores and 512 GB of RAM per socket based on the AMD EPYC architecture. This processing pipeline is referred to as "HLT1" following standard LHCb nomenclature. One GPU was deployed per server during 2022 data-taking, increasing to two GPUs per server for 2023 data-taking. The reconstruction algorithms are implemented in the Allen [19] framework and run almost entirely on the GPU, with the host CPU(s) responsible for copying data to and from the GPU and a certain amount of auxillary monitoring tasks. During 2022, the LHC provided non-empty pp collisions at around 20 MHz which means each GPU must be able to process around 105 kHz of data input rate. In 2023, with a doubled number of GPUs and a LHC input rate of 30 MHz, each GPU can handle up to 80 kHz of data rate.

TABLE 1. Comparison of the four major experiments at the LHC in terms of instantaneous luminosity $\mathcal{L}$, number of pp collisions per bunch crossing or pile-up, the rate at which the track reconstruction is performed and the number of reconstructed tracks per second. pp collisions per bunch crossing.

	LHCb	ATLAS	CMS	ALICE
$\mathcal{L}\ [cm^{-2}s^{-1}]$	2×10^{33}	2×10^{34}	2×10^{34}	6×10^{27}
pile-up	5	60	60	1
reconstruction rate	30 MHz	100 kHz	100 kHz	50 kHz
reconstructed tracks/s	1800 M	90 M	90 M	10 M

The track finding forms only one part of HLT1 employing around 50 % of all available HLT1 resources. As we will show later, the Looking Forward algorithm can typically use around a 20 % of the total resources for an HLT1 sequence which fits into the overall budget.

The second stage of LHCb's real-time processing, "HLT2", is implemented using around 3500 dual-socket CPU servers [5] of varying generations and core counts, primarily using Intel architectures. It is required to run at around 500 Hz on an "average" server representative of the overall data centre performance. LHCb's physics analyses use simulated events, which have been processed in the same way as data, to correct for detector inefficiencies. Simulation is processed exclusively using CPU servers. Therefore the Looking Forward algorithm is optimised for execution on GPUs but is required to compile for, and run on, CPU architectures. It must be sufficiently fast when executed on a CPU such that it can be deployed to simulate LHCb events without an increase of the overall cost of simulation production.

The algorithm is required (and tested) to be deterministic, however strict bitwise reproducibility when running in a different environment or on a different architecture is not a design requirement. Such reproducibility is impossible without emulators on parallel architectures in general, including between different CPU generations or instruction sets. However an emulator would clash with the earlier requirement that the Looking Forward algorithm remains computationally efficient when executed on a CPU. Strict reproducibility is not necessary because LHCb reconstructs its data only once, in real time. The selected data is annotated with provenance information that describes the objects which caused the real-time processing to keep it. Since LHCb simulation does not perfectly agree with data, the collaboration has a number of strategies [20], [21] for calibrating data-simulation differences. These methods can

be applied to correct for small differences in the Looking Forward algorithm when executed on a CPU or on a GPU. The differences in the results of the Looking Forward algorithm when executed on a CPU or on a GPU must be at the permille level or smaller. The difference is considered negligible compared to other, typically percent level, data-simulation differences [20], [21] in LHCb's tracking.

In order to satisfy LHCb's physics requirements the Looking Forward algorithm is required to have an efficiency which is as close as possible to the forward tracking which runs, under much more relaxed computational constraints, in HLT2. In particular, performance for tracks with $p_T > 500$ MeV and tracks produced in the decays of beauty hadrons are required to be within a few percent different from what is achieved in HLT2. The algorithm is required to be configurable such that physics performance can be smoothly traded off against computational complexity, and to be robust against inefficiencies in the detector itself. As the UT subdetector was not installed in time for the 2022 data-taking, the changes to the Looking Forward algorithm to deal with its absence, which are documented in this paper, represent a stress test of the robustness requirement.

IV. BENCHMARKING SETUP

High energy physics experiments have traditionally benchmarked algorithmic performance in terms of wall clock time, i.e. how many seconds a given algorithm takes to process an average bunch crossing on a reference computing node. This metric can be suitable for serial algorithms, but is inherently flawed when benchmarking highly parallel architectures. In such a regime the correlation between a given algorithm's reported resource usage and the overall sequence throughput is weak at best. For these reasons our primary computational benchmarking metric is the overall throughput of the nominal LHCb HLT1 sequence. We additionally cite the GPU resource usage of Looking Forward, as reported by the NVIDIA Nsight Compute profiling tool, and compare it to other parts of the HLT1 sequence.

Computational benchmarking is carried out using a dedicated testbench server hosting a range of NVIDIA GPU cards, as well as a dual-socket server equipped with AMD EPYC 7502 CPUs. The specifics of the hardware are detailed in Table 2.

Throughput is measured on samples of simulated "minimum bias" events produced with the full LHCb detector simulation under nominal Run 3 conditions. Throughput is benchmarked as a function of the number of pp collisions in the event to study the scalability of the Looking Forward algorithm in different running conditions.

Physics benchmarking is carried out using full LHCb detector simulation under Run 3 conditions. The basic performance metrics are described in [22] and recapitulated here for convenience. Reconstructed tracks are matched to ground truth information in the simulated samples to determine whether they represent a genuine charged particle trajectory. It is required that more than 70% of detector hits on a reconstructed track and a ground truth particle match in order to consider that particle correctly reconstructed. The algorithm's efficiency to correctly reconstruct particles is measured with respect to "reconstructible" charged particles which leave a minimal number of hits in the tracking detectors (VELO, UT, FT). The fake rate is defined as the fraction of reconstructed tracks which are not matched to a ground truth particle. The efficiency and fake rate are benchmarked as a function of kinematic and geometric properties of the particles. The resolution on the reconstructed track momenta and the resolution on the track slopes are also reported. The performance of the algorithm for CPU and GPU architectures is compared by processing the same set of simulated events on each architecture and comparing the reconstructed quantities of interest on a track-by-track basis between the two.

TABLE 2. GPU and CPU hardware used to benchmark the HLT1 throughput. A comparison is presented for three NVIDIA graphic cards (GeForce RTX 3090, RTX A5000 and GeForce RTX 2080 Ti) and an AMD EPYC 7502 CPU. The number of cores, the maximum frequency, the cache, the dynamic random-access memory (DRAM) and the thermal design power (TDP) are shown.

Unit	# cores	Max freq. [GHz]	Cache [MiB - L2]	DRAM [GiB]	TDP [W]
GeForce RTX 3090	10496	1.69	6	24 GDDR6X	350
RTX A5000	8192	1.69	6	24 GDDR6	230
GeForce RTX 2080 Ti	4352	1.54	6	11 GDDR6	250
EPYC 7502 32-Core	32	3.35	128 (L3)	64 DDR4	180

V. PROPAGATION OF TRACKS IN LHCB'S MAGNETIC FIELD

In order to accurately reconstruct charged particle trajectories it is essential to have an accurate model of their bending in the experiment's magnetic field. The general equation of motion of a charged particle with momentum $\vec{p}$, charge q and velocity $\vec{v}$ in a magnetic field $\vec{B}$ is

$$\frac{\mathrm{d}\vec{p}}{\mathrm{d}t} = q\vec{v} \times \vec{B},$$

which leads to the following equations in the three dimensions considering the momentum components p_x, p_y, and p_z:

$$\frac{\mathrm{d}p_x}{\mathrm{d}z} = q(t_y B_z - B_y),$$
$$\frac{\mathrm{d}p_y}{\mathrm{d}z} = q(B_x - t_x B_z),$$
$$\frac{\mathrm{d}p_z}{\mathrm{d}z} = q(t_x B_y - t_y B_x).$$

Here $t_x = p_x/p_z = \frac{\mathrm{d}x}{\mathrm{d}z}$ and $t_y = p_y/p_z = \frac{\mathrm{d}y}{\mathrm{d}z}$ are the track slopes.

The two differential equation for the tracks slopes in the **x-z** and **y-z** planes respectively are

$$\frac{\mathrm{d}t_x}{\mathrm{d}z} = \frac{q}{p}\sqrt{1 + t_x^2 + t_y^2}(t_x t_y B_x - (1 + t_x^2)B_y + t_y B_z),$$

$$\frac{\mathrm{d}t_y}{\mathrm{d}z} = \frac{q}{p}\sqrt{1+t_x^2+t_y^2}((1+t_y^2)B_x - t_x t_y B_y - t_x B_z).$$

The LHCb magnetic field has been mapped [23], [24] in a series of dedicated measurement campaigns. Its dominant component is B_y, so that tracks traversing the magnet are deviated almost entirely in **x**, with deviations in **y** being smaller than the detector resolution in most cases. Assuming therefore that $B_x \sim 0$, $B_z \sim 0$ then for small $|t_x|$ and $|t_y|$ the earlier equations can be approximated keeping only the first order terms, leading to

$$\frac{\mathrm{d}^2x}{\mathrm{d}z^2} = \frac{t_x}{z} \sim -\frac{q}{p}B_y,$$
$$\frac{\mathrm{d}^2y}{\mathrm{d}z^2} = \frac{t_y}{z} \sim 0.$$

The y-z equation results in a simple linear model,

$$y(z) = y_0 + t_y(z - z_0),$$

where y_0 is the y coordinate at a given reference position z_0.

For the **x-z** track projection, the B_y dependence on z can be parameterised at first order as

$$B_y(z) \sim B_0 + B_1(z - z_0)$$

in the magnetic field tails within the FT acceptance, assuming a linear decrease of the absolute value of the B_y component along the z direction. This in turn leads to the following dependence of the track's **x** position as a function of **z**, within the FT acceptance

$$x(z) = x_0 + t_x(z - z_0) + \frac{q}{2p}B_0(z - z_0)^2(1 + \mathrm{dRatio}(z - z_0)), \quad (1)$$

where x_0 is the coordinate at a reference position z_0 and the quantity $\mathrm{dRatio} = \frac{B_1}{3B_0}$ is roughly constant in the region where the tracks are extrapolated. This parameterization avoids the slowdowns due to real-time usage of the LHCb magnetic field maps, such as memory access, copying to GPU, GPU memory size consumption, ...by keeping the precision required by the HLT1 reconstruction.

The track fit model within the FT acceptance region depends linearly on five adjustable parameters: two related to the **y-z** projection, y_0 and t_y, and three related to the **x-z** projection, x_0, t_x and $B_0\frac{q}{p}$, similarly as what is in [25].

VI. ALGORITHM LOGIC

The Looking Forward algorithm begins with tracks which have been reconstructed by the search by triplet [26] algorithm in the VELO and the CompassUT [27] algorithm in the UT. The track state used to define the parameters in Equation (1) is calculated at the downstream end of the UT detector.

The Looking Forward algorithm is composed of four main steps:

1) **Defining the search windows opening**: search windows tolerances are evaluated in each SciFi layer extrapolating the input VELO-UT tracks to reduce the number of hit combinations;
2) **Triplet seeding**: triplet of hits are combined using the x-layers information;
3) **Extending triplets to other layers**: triplets are extended to the other layers and a track fit is performed;
4) **Quality filter**: candidates are filtered and selected thanks to a track quality factor based on the fit χ^2.

After these four steps, the calculation of the momentum of the particle and evaluation of the track states are performed. The overall methodology is summarised in the diagram shown in Figure 2.

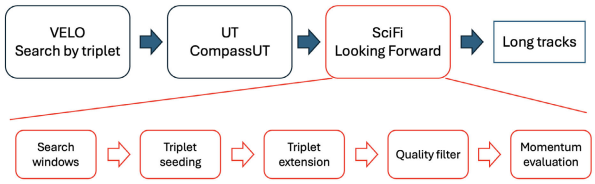

FIGURE 2. Diagram of the long track reconstruction at the first high-level trigger at LHCb. Tracks are reconstructed before the magnet by the search by triplet [26] and CompassUT [27] algorithms, respectively in the VELO and UT detectors. They are then extrapolated to SciFi detector with the Looking Forward algorithm, which is divided into four reconstruction steps culminated by a momentum evaluation step.

A. DEFINING THE SEARCH WINDOWS

Defining the search windows in the FT is a necessary step to reduce the number of hit combinations. Each SciFi layer contains an average number of hits of $n_{hits} \sim 400$ which are combined into triplets, reaching a maximum number of combinatorics of $n_{hits} \times n_{hits} \times n_{hits}$ multiplied by the number of input tracks. In order to reduce the algorithm complexity, search windows are defined in each SciFi layer using the input track information.

The track's slope in the non-bending plane is assumed constant, $t_y^{\mathrm{Velo}} = t_y^{\mathrm{FT}}$. The t_y information defines whether the particle is traveling in upper or lower half of the FT, since tracks emerging from the VELO are unlikely to cross both halves of the FT. This allows one half of the FT hits to be removed from consideration, immediately reducing the computational burden.

A fast momentum estimation is the so-called p_{T}-kick method [28], [29]. The effect of the magnetic field between two detectors is parametrised as an instantaneous kick to the momentum vector at the center of the magnet, $z_{bending}$, as illustrated in Figure 3. The momentum kick, $\Delta\vec{p}$ is defined as

$$\Delta\vec{p} = q \cdot \int d\vec{l} \times \vec{B},$$

with an integral of $\vec{B}$ the magnetic field along the path followed by the track.

Since, as discussed earlier, the bending can be assumed to occur only in the **x** plane with $B_x \sim 0$, $B_z \sim 0$ and for small

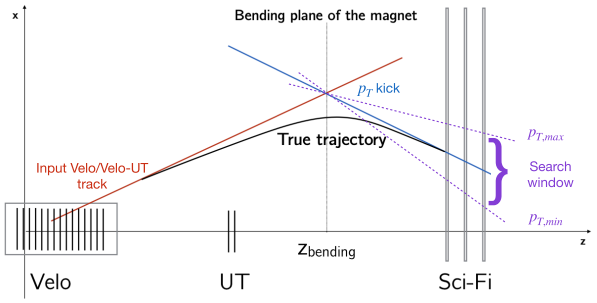

FIGURE 3. An illustration of the p_T-kick method [29].

$|t_x|$ and $|t_y|$, this simplifies to

$$\frac{q}{p} = \frac{1}{q \cdot \int |d\vec{l} \times \vec{B}|_x} \times \left(\frac{t_{x,f}}{\sqrt{1 + t_{x,f}^2 + t_{y,f}^2}} - \frac{t_{x,i}}{\sqrt{1 + t_{x,i}^2 + t_{y,i}^2}} \right), \quad (2)$$

with $t_{x,i}$ ($t_{x,f}$) and $t_{y,i}$ ($t_{y,f}$) respectively the x and y initial (final) slopes of the track's segments. In this way, it is sufficient to know the initial and final track slopes to have an estimate of the particle's momentum. Alternatively, given a VELO-UT track and an assumed momentum, the p_T-kick method can be used to define search windows in the FT and further reduce the number of hits considered for a specific input track. The values of $\frac{1}{q \cdot \int |d\vec{l} \times \vec{B}|_x}$ are stored in a lookup table and used by the Looking Forward algorithm for fast processing.

The extrapolated center of the search windows for each VELO-UT track is defined in every FT layer i as x^i_{extrap}, y^i_{extrap}, and is calculated by extrapolating the track from the **z**-position of the center of the magnet with the p_T-kick method. The size of the search windows in the x layers is defined as:

$$x_{\text{min}} = x_{\text{extrap}} - x_{\text{tol}} - x^{\text{left}}_{\text{asym}}$$
$$x_{\text{max}} = x_{\text{extrap}} + x_{\text{tol}} + x^{\text{right}}_{\text{asym}}$$

with $[x_{\text{min}}, x_{\text{max}}]$ the search window region in x, x_{tol} the window size or tolerance factor, and $x^{\text{left/right}}_{\text{asym}}$ a magnetic field asymmetry factor. The x_{tol} factor defines the search window size as:

$$x_{\text{tol}} = 150 \text{ mm} + \frac{2000 \text{ GeV/c}}{p}$$

as a function of the particle's momentum p. The last term, $x^{\text{left/right}}_{\text{asym}}$, takes into account the fact that the magnetic field is not a pure dipole, but is affected by fringe effects mainly away from its center. It is defined as:

$$x^{\text{left}}_{\text{asym}} = \begin{cases} 100 \text{ mm}, & \text{if } q = +1 \text{ \& MU OR } q = -1 \text{ \& MD} \\ 0 \text{ mm}, & \text{otherwise} \end{cases}$$
$$x^{\text{right}}_{\text{asym}} = \begin{cases} 100 \text{ mm}, & \text{if } q = -1 \text{ \& MU OR } q = +1 \text{ \& MD} \\ 0 \text{ mm}, & \text{otherwise} \end{cases} \quad (3)$$

depending on the charge of the particle q and the polarity of the magnet (MU or MD).

The center of the search window in the **u** and **v** layers is defined by extrapolating the track as a straight line from the neighbouring **x** layer. This extrapolation is corrected by a $\pm \sin 5°$ factor of the **u** and **v** layer, while a tolerance of ± 800 mm around the center defines the search window for the hits.

The maximum number of hits allowed in the search window for any single layer is defined to be $n^{\text{window}}_{\text{hits}}$, whose values depending on the sequence is reported in Table 3. This limit constraints the number of combinations passed to the next stage and consequently the memory consumption of the algorithm. If there are more than $n^{\text{window}}_{\text{hits}}$ hits in a given search window, the hits kept are chosen symmetrically around the center of the window.

B. TRIPLET SEEDING

Once the search windows have been defined, the pattern recognition begins by forming triplets of hits (h_0, h_1, h_2) from the **x** layers. Two configurations of layers are used for this search: either the first layer of each T-station or the last layer of each T-station. First, all possible hit doublets are formed for each VELO-UT track from the first and last station (h_0, h_2). Two conditions are used to filter doublet candidates:

1) If the VELO-UT input track has momentum < 5 GeV/c, the bending of track from the VELO to the UT is required to be in the same direction as the bending of the track from the VELO to the FT doublet;
2) A maximal tolerance in the **x**-opening at the **z** position of the magnet z_{bending} is defined as a function of the momentum p^{VeloUT} of the VELO-UT track and on the doublet's slope t_x^{doublet}. It varies from 8 to 40 mm. The difference in the straight line extrapolation of the VELO-UT track and the FT doublet to the center of the magnet is required to be smaller than this tolerance.

The filtered doublets are upgraded to triplets by adding the hit in the corresponding second T-station using a straight line extrapolation corrected by

$$x_1^{\text{expected}} = z_1 \cdot t_x^{(h_0\text{-}h_2)} + (x_0 - t_x^{(h_0\text{-}h_2)} \cdot z_0) \cdot K_{x_1},$$

where x_1^{expected} is the extrapolated **x**-position of the h_1 hit, z_i and x_i are the **z**- and **x**-positions of the respective hits, $t_x^{(h_0\text{-}h_2)}$ is the slope in the bending plane between the h_0 and h_2 hits and K_{x_1} is a sagitta-like correction factor.

Because of the residual magnetic field in the T-stations the value of K_{x_1} is different depending on whether the first or last T-station layers are being used: $K^{\text{first}}_{x_1} = 1.00177513$ and $K^{\text{last}}_{x_1} = 1.00142634$.

The hit with **x**-value closest to x_1^{expected} is added to form the triplet candidate if the $\chi^2_{\text{triplet}} = (x_1^{\text{hit}} - x_1^{\text{expected}})^2 < T_{\text{triplet}}$, where T_{triplet} is a tolerance reported in Table 3. The maximum number of selected triplets is a configurable parameter. A parameter scan shows that the optimal number in current LHCb data-taking conditions is $n^{\text{track}}_{\text{triplets}} = 12$ triplets per

VeloUT track. If the maximum number is exceeded, the 12 candidates with the smallest χ^2_{triplet} are selected.

C. EXTENDING TRIPLETS TO OTHER LAYERS

Triplets are extended to the remaining FT layers in order to form full track candidates. A track candidate must contain at least 9 hits, with at least one hit in the **u** or **v** layers per T-station. This threshold is chosen to keep the fake rate manageable.

The extrapolation is performed using Equation 1. Tracklets are first extended to the three missing **x** layers, computing the expected **x**-position of the hit in layer i:

$$x_i^{\text{expected}}(z) = a_x + t_x \cdot dz_i + c_x \cdot dz_i^2 \cdot (1 + \text{dRatio} \cdot dz_i) \tag{4}$$

where the values of a_x,t_x,c_x and dRatio are evaluated using the position information from the triplet hits. The effect of the magnetic field is parametarized [29] as a function of $dz_i = z_i - z^{\text{ref}}$, the difference between the **z**-position of the i layer and a reference plane at $z^{\text{ref}} = 8520$ mm. For each **x**-layer i, the hit with **x**-value closest to x^{expected} is added to the triplet if $\chi^2_{x_i} = (x_i^{\text{hit}} - x_i^{\text{expected}})^2 < T_x$, where the T_x tolerance factors are reported in Table 3.

Each tracklet candidate is then extended looking for hits in the **u** and **v** layers, where Equation 4 is corrected for the $\pm \sin 5°$ angle of those layers. Hits are again added according to a χ^2 tolerance listed in Table 3. However for the **u** and **v** layers the tolerance is a function of the track's slopes in order to allow more generous windows for more peripheral tracks in both **x** and **y**.

TABLE 3. Parameters and tolerances used in the forward reconstruction. The tolerance relative to the y-position T_y has a much larger value compared to the x-position due to the fact that the information only comes from the u and v layers and therefore has a poorer resolution. T_{t_y} represents the y-z slope difference tolerance when performing the mean squared fit in the y direction. The maximum number of hits per search window and triplets considered for the same input tracks are also reported.

parameters	Forward	Forward no-UT
T_{triplet}	8.0	2.0
T_x	2.0	0.5
T_{uv}	$50 \cdot (t_x + t_y)$	$15 \cdot (t_x + t_y)$
T_{t_y}	0.02	0.003
T_y	800	800
T_Q	0.5	0.5
$n_{\text{hits}}^{\text{window}}$	32	64
$n_{triplets}^{track}$	12	20

D. QUALITY FILTER

The reconstructed tracks are filtered in order to reduce the fake rate. First, a linear least square fit in the y-direction is performed on the candidates evaluating the t_y^{expected} slope and χ^2 value $\chi^2_{y_i} = (y_i^{\text{hit}} - y_i^{\text{expected}})^2$ based on the expected and measured **y**-positions of the **u** and **v** hits. The difference in the measured and expected slopes in the non-bending plane is required to be less than a tolerance T_{t_y} whose value is given in Table 3.

The total **x** and **y** quality factors are determined as:

$$Q_x = \sum_{i=0}^{N_{\text{x hits}}} \frac{\chi^2_{x_i}}{T_x} + \sum_{j=0}^{N_{\text{uv hits}}} \frac{\chi^2_{uv_j}}{T_{uv}},$$
$$Q_y = \sum_{i=0}^{N_{\text{uv hits}}} \frac{\chi^2_{y_i}}{T_y}. \tag{5}$$

Here $\chi^2_{x_i}$ and $\chi^2_{uv_j}$ are the χ^2 values evaluated previously for each respective hit normalized by their tolerances T_x and T_{uv}, while $\chi^2_{\text{y-fit}}$ is the **y**-fit χ^2 normalized to its tolerance T_y.

Each track is assigned an overall quality factor

$$Q = \left(\frac{Q_x}{nDoF_x} + \frac{Q_y}{nDoF_y}\right) \cdot C(n_{\text{hits}}), \tag{6}$$

where $nDoF_{x\text{-}y}$ is the number of fit degrees of freedom in the **x** and **y** directions and $C(n_{\text{hits}})$ is a multiplicative parameter dependent on the number of hits on the track candidate. The values of $C(n_{\text{hits}})$ are reported in Table 4 and favor tracks made out of a greater number of hits. The number of degrees of freedom $nDoF_x$ when performing the fit in the **x**-**z** plane is $nDoF_x = n_{\text{hits}}^{\text{total}} - 3$ as only information from the three **x**-layers is used. In the **y**-**z** plane, $nDoF_y = n_{\text{hits}}^{uv} - 2$ as only two uv-hits are used for the linear fit.

Track candidates are accepted as reconstructed if their quality factor is lower than the tolerance T_Q given in Table 3. If more than one candidate is found for a given VELO-UT track, the one with lowest value of T_Q is kept.

TABLE 4. Values of the multiplicative parameter $C(n_{\text{hits}})$ as a function of the number of hits on the track candidate.

n_{hits}	$C(n_{\text{hits}})$ value
9	5.0
10	1.0
11	0.8
12	0.5

E. NO-UT TUNING

The fact that the UT tracker was not installed in time for the 2022 data-taking required the forward tracking to be retuned to extrapolate tracks directly from the VELO to the FT. Since VELO tracks do not have a charge or momentum estimate, their search windows must be double-sided and significantly wider than those of the VELO-UT tracks.

The VELO input tracks are extrapolated as a straight line directly to the various FT layers. As the tracks momenta are unknown, they are set to a minimum value (maximum allowed curvature) which is a tunable parameter of the algorithm. The baseline no-UT thresholds are either a momentum of 5 GeV or a transverse momentum of 1 GeV. This transverse momentum threshold is converted into a momentum threshold using the VELO track slopes, and the tighter of the two momentum thresholds is used to determine the search window. The VELO track t_y slope is still used to determine which half of the FT, lower or upper, is used when extending the track to the various layers. Figure 4 shows a scheme of the search window strategy.

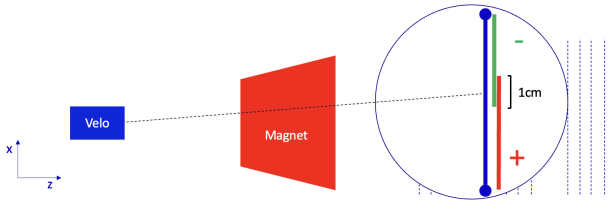

FIGURE 4. Scheme of the search window strategy for the forward tracking in the absence of the UT. A small overlap of 1 cm is allowed between the two windows to account for very high momentum tracks, which are expected to travel in a straight line.

The tolerance of the search window is evaluated with the following polynomial approximation [30]

$$x_{tol} = \frac{1}{p^{\text{Velo}}}\Big[c_0 + t_x(c_1 + c_2 t_x) + t_y^2(c_3 + t_x(c_4 + c_5 t_x)) + \frac{1}{p^{\text{Velo}}}(c_6 t_x + c_7 \frac{1}{p^{\text{Velo}}})\Big] \quad (7)$$

where p^{Velo} is the assumed VELO track momentum, c_i are the coefficients defined in Table 5, and t_x and t_y are the slopes in the **x**-**z** and **y**-**z** plane of the VELO input track. The polynomial approximation is used to determine a curvature correction assuming a minimum momentum. It is applied instead of the more common Runge-Kutta method [31], which provides a higher precision but is too computationally demanding for our requirements.

TABLE 5. Polynomial coefficients tuned on simulation [30] approximating the curvature of a VELO track with an assumed minimum momentum.

c_0	c_1	c_2	c_3
4824.3	426.3	7071.1	12080.4
c_4	c_5	c_6	c_7
14077.8	13909.3	9315.3	3209.5

The double-sided **x**-layer search windows are defined as

$$\begin{aligned} x_{\text{min}}^{LEFT} &= x_{\text{extrap}} - x_{\text{tol}}; \\ x_{\text{max}}^{LEFT} &= x_{\text{extrap}} + x_{\text{overlap}}; \\ x_{\text{min}}^{RIGHT} &= x_{\text{extrap}} - x_{\text{overlap}}; \\ x_{\text{max}}^{RIGHT} &= x_{\text{extrap}} + x_{\text{tol}}. \end{aligned} \quad (8)$$

Here x_{extrap} is the VELO track's straight line extrapolation **x**-position in the layer, x_{tol} the window tolerance defined in Equation 7, and $x_{\text{overlap}} = 5$ mm is an overlap factor between the two windows. The overlap is necessary in order to ensure that hits on very high momentum tracks are considered for both charge assumptions, minimizing inefficiencies in these cases.

The **u** and **v** layer search windows are computed analogously to Equation 8, with a tolerance of $x_{\text{tol}} = 1200$ mm. The extrapolation to these layers is similarly analogous to the case where UT information is used.

The tunable parameters reported in Table 3 are optimised for no-UT configuration. The limit of hits for each search window is set to $n_{\text{hits}}^{\text{window}} = 64$ hits, symmetrically around the center of each window. It is double that of the forward tracking with the UT because of the much larger search windows. In order to compensate the higher number of combinations, tighter χ^2 thresholds are defined in the next section to handle the fake rate.

The triplet seeding is analogous to the algorithm which uses UT information, with the combinations performed separately for each charge assumption. The **x** tolerance at the **z** position of the magnet is fixed to 10 mm in the baseline no-UT tuning since the algorithm is primarily searching for higher momentum tracks which bend less. The maximum number of selected triplets is increased to $n_{\text{triplets}}^{\text{track}} = 20$ because of the wider search windows. From this point on the algorithm follows the same steps as for the case with UT information, with generally tighter χ^2 tolerance requirements which improve the computational performance and reduce the fake rate. The charge and the momentum evaluations are performed with the p_{T}-kick method, as explained in the following.

F. CALCULATION OF MOMENTUM AND TRACK STATES

The evaluation of the track momentum is obtained using the Equation (2). Here the term associated to the integrated magnetic field along the track trajectory is parameterised according to a fourth order polynomial expansion as a function of dSlope $= t_x^{\text{FT}} - t_x^{\text{Velo}}$, the measured variation of the track slope in the bending plane. The value of $\int \vec{B} \times d\vec{L}$ is evaluated with a dedicated parameterisation of fourth order polynomial expansion $\int \vec{B} \times d\vec{L} = F(t_x^{\text{Velo}}, t_y^{\text{Velo}}, \text{dSlope})$, where the coefficients of the polynomial function F depend on a given track's entry slope direction in the field:

$$F(t_x^{\text{Velo}}, t_y^{\text{Velo}}, \text{dSlope}) = \sum_{i=0}^{4} c_i \text{dSlope}^i \quad (9)$$

In order to determine $c_{i=0,1,2,3,4}$ as a function of $t_{x,y}^{\text{Velo}}$, a set of toy tracks are generated. These toy tracks equi-populate the acceptance of $t_{x,y}^{\text{Velo}}$, and have a flat q/p spectrum in each region of $t_{x,y}^{\text{Velo}}$. Each parameter c_i is fitted with dedicated two dimensional polynomials $c_i = \sum_{k,m} c_i^{km} \left(t_x^{\text{Velo}}\right)^k \left(t_y^{\text{Velo}}\right)^m$. The expansion is done up to a seventh degree ($k + m \leq 6$) to ensure a full LHCb acceptance coverage for the parameterisation. The composition of polynomials are done to ensure the B field symmetries are respected selecting only even/odd combinations of m, k. The fits to the parameters c_i are shown in Figure 5.

VII. PERFORMANCE

The algorithm is developed and optimised to achieve a tracking efficiency for long tracks with momentum above 5 GeV and p_{T} above 1 GeV around 90 % by keeping the overall HLT1 throughput budget per GPU card higher than 105 kHz. The throughput of the HLT1 sequence is evaluated using a sample of simulated minimum bias pp collisions

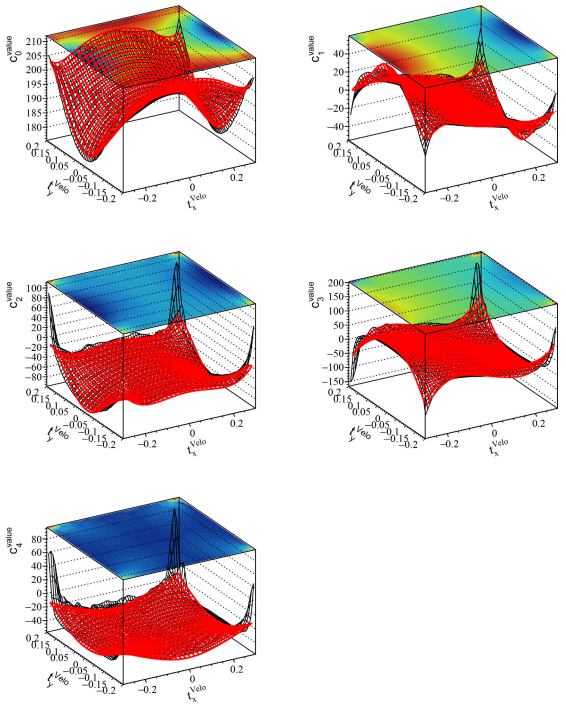

FIGURE 5. Fits determining the fourth order expansion polynomial terms for c_i. The c_i values from the polynomial expansion allow to parameterise the q/p of tracks given their entry slope in the dipole magnet and the observed change in slope after traversing the dipole magnet region.

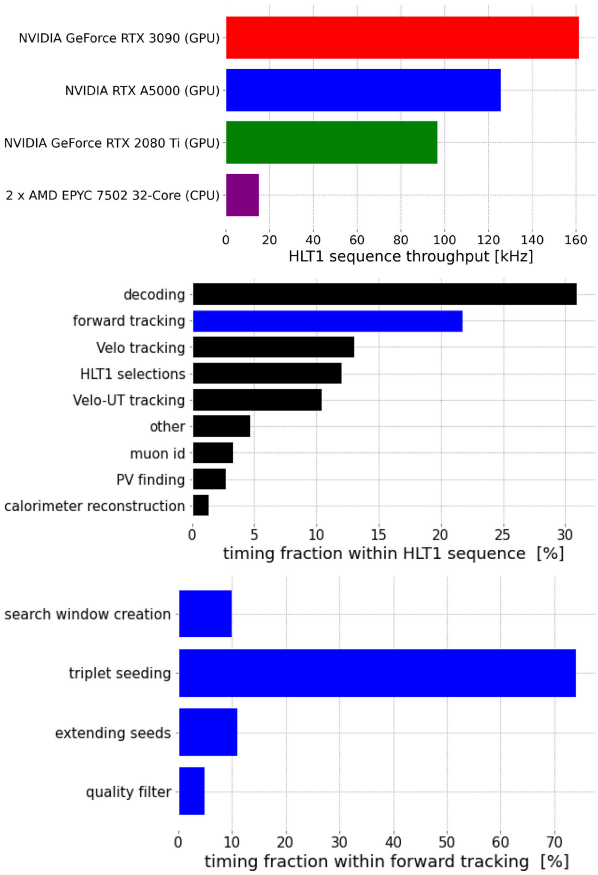

FIGURE 6. The throughput of the reference HLT1 sequence for three different NVIDIA GPU cards and a dual-socket AMD EPYC 7502 CPU server. The breakdown of the HLT1 throughput is shown among the algorithms in the sequence including reconstruction, selection and decoding of the detectors information algorithms. The timing fraction of the substeps of the forward tracking are shown in blue ordered by their execution time.

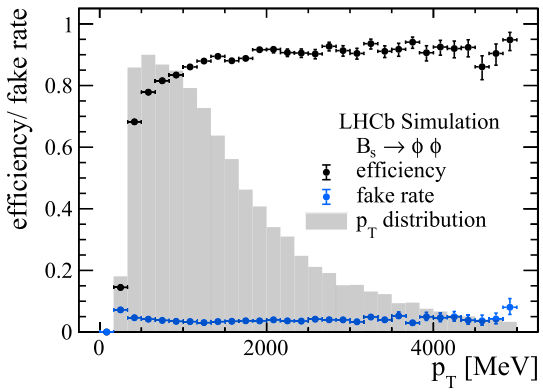

FIGURE 7. The efficiency and fake rate of the Looking Forward algorithm as tuned for the reference HLT1 sequence, plotted as a function of particle p_T. The different components are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

which represent the collisions and detector response in real data. It is shown in Figure 6 for both GPU and CPU architectures, as well as the breakdown of this throughput among algorithms. The measured throughput of the HLT1 sequence on a NVIDIA A5000 GPU card is 130 kHz and around six times higher than on AMD EPYC CPU servers. The Looking Forward algorithm employs around 20 % of the total HLT1 sequence, running at a throughput of 650 kHz. Within the forward tracking sequence, the largest time fraction is occupied by the triplet seeding step employing a 70 % fraction of the whole algorithm.

The forward tracking algorithm optimised on a multi-core CPU and employed by LHCb at the HLT2 stage achieves a higher efficiency of around 95 %, however, running with a throughput of 10 kHz per CPU node [32]. The superior parallelization of the Looking Forward algorithm allows the reconstruction of tracks with a speed almost 60 times larger compared to the alternative version optimised for CPUs, accepting a minor loss in physics efficiency [33].

Physics performance is evaluated on a standard sample of simulated $B_s^0 \to \phi(K^+K^-)\phi(K^+K^-)$ decays, which has historically been the decay mode of choice for benchmarking the performance of LHCb tracking algorithms. The efficiency for tracks produced in the decays of beauty hadrons, as well as the fake rate, is shown in Figure 7 as a function of particle transverse momentum, Figure 8 as a function of particle momentum and in Figure 9 as a function of particle pseudorapidity. The efficiency plateaus is above 90% at high transverse momenta or momenta when integrated in the pseudorapidity range $2 < \eta < 5$. Except at the edges of the algoright acceptance, the fake rate is generally flat as a function of both pseudorapidity, transverse momentum and momentum.

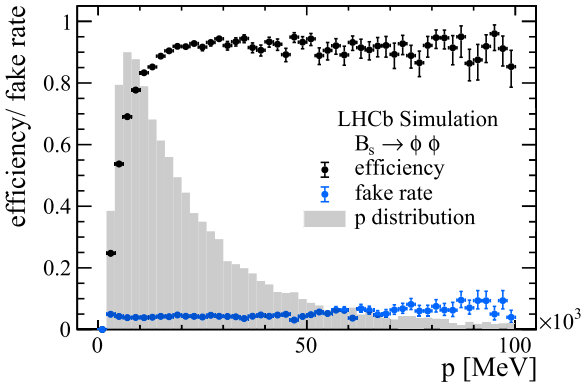

FIGURE 8. The efficiency and fake rate of the Looking Forward algorithm as tuned for the reference HLT1 sequence, plotted as a function of particle momentum. The different components are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

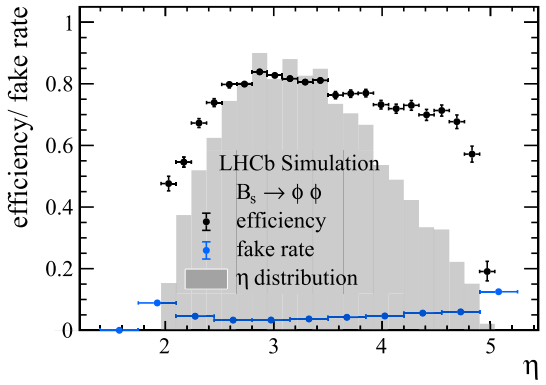

FIGURE 9. The efficiency and fake rate of the Looking Forward algorithm as tuned for the reference HLT1 sequence, plotted as a function of particle pseudorapidity. The different components are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

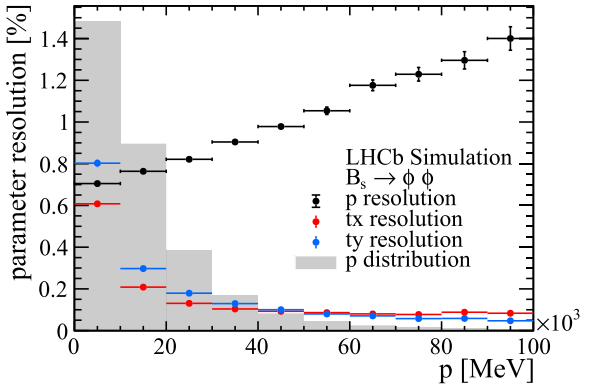

FIGURE 10. The momentum and track slope at the last SciFi detector layer resolutions of the Looking Forward algorithm as tuned for the reference HLT1 sequence, plotted as a function of particle *p*. The different components are described in the figure legend.

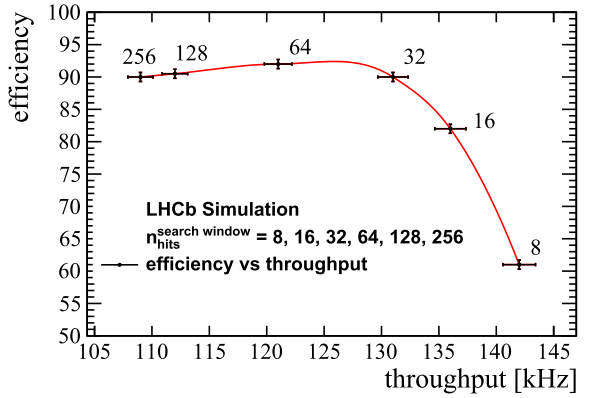

FIGURE 11. The efficiency and throughput results for different $n_{\text{hits}}^{\text{window}}$ number of hits in the search window values. The $n_{\text{hits}}^{\text{window}}$ variable is proportional to the search window size. The efficiency is evaluated for tracks with momentum above 5 GeV and p_T above 1 GeV and the throughput is measured on a RTX A5000 GPU card using the reference HLT1 sequence.

The resolution on track momenta and track slopes is shown in Figure 10 as a function of particle momentum. Resolutions below 1% for both momentum and track slopes are achieved for the great majority of tracks.

The efficiency and throughput results are shown for different $n_{\text{hits}}^{\text{window}}$ values in Figure 11. The throughput increases as $n_{\text{hits}}^{\text{window}}$ and the search window size decreases, as less hit combinations are computed, while the total tracking efficiency decreases. If more hit combinations are allowed, enlarging the search window and $n_{\text{hits}}^{\text{window}}$, the efficiency increases however it reaches a plateau above $n_{\text{hits}}^{\text{window}} = 64$ as the algorithm does not achieve the precision required to select the right track candidate among many combinations. The working point of $n_{\text{hits}}^{\text{window}} = 32$ is chosen to the reference as it optimises both efficiency and throughput.

The scalability of this reference configuration is tested by measuring the efficiency and fake rate as a function of the number of *pp* collisions in a simulated event, as shown in Figure 12. While efficiency degrades, and the fake rate increases, with an increasing number of collisions, this deterioration is gradual and limited in absolute size. In order to benchmark throughput in the same way, dedicated samples of events containing a specific number of *pp* collisions are created from standard LHCb simulation samples. The results are shown in Figure 13. The throughput decreases gradually as function of the number of *pp* collisions, as the number of tracks to be reconstructed in an event increases as a function of it.

We have profiled the forward subsequence of algorithms with the Nsight Compute profiler. The subsequence is dominated by triplet seeding, consisting in 74% of the subsequence. Triplet seeding has an arithmetic intensity of 29.78, making it compute bound. We observe a low GPU compute throughput of 33%. The reason behind this inefficiency are stalls in the gangs of threads (warps) scheduled by the CUDA scheduler and can be solved issuing more warps in the kernel call. We are implicitly solving this by running several streams concurrently, which cannot be detected unfortunately by Nsight Compute, as it runs kernels in isolation.

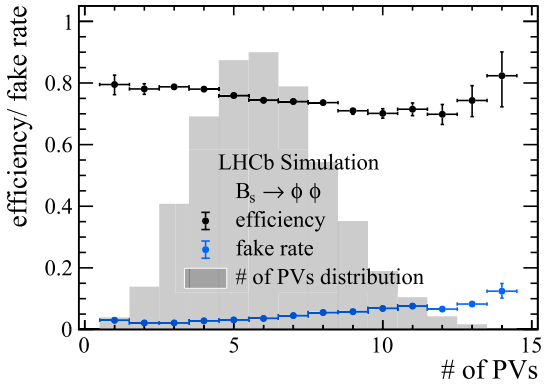

FIGURE 12. The efficiency and fake rate of the Looking Forward algorithm as tuned for the reference HLT1 sequence, plotted as a function of the number of *pp* collisions in the event. The different components are described in the figure legend.

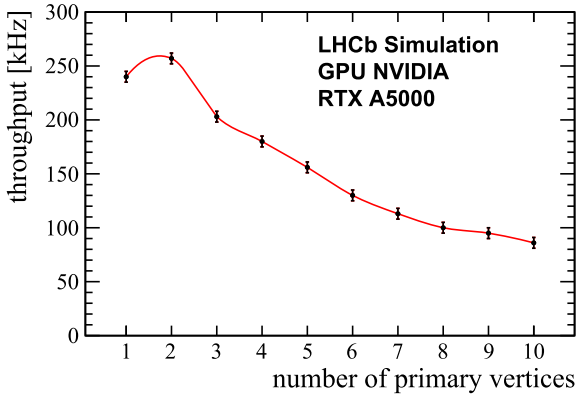

FIGURE 13. The throughput of the reference HLT1 sequence on the RTX A5000 card as a function of the number of *pp* collisions in the event.

Out of the three parameters that impact occupancy, block size and shared memory are reportedly already optimized for this architecture. However, an area of possible improvement is register utilization, which limits the occupancy of warps to 70% of what would be theoretically achievable on the A5000 Streaming Multiprocessors. Our kernel requires 72 registers currently, and we are considering reformulations of its critical sections to improve it.

Memory access patterns do not seem to be an issue. We are loading hits in a coalesced manner onto shared memory, and we save tracks upon request by every thread. Triplet seeding only requires fp16 precision for arithmetic, and thus we use fp16 to reduce shared memory utilization in this kernel. As hit data comes in fp32, this results in higher memory pressure than required. If fp16 were reused across other algorithms, it would be sensible to store hit data in fp16 in addition to fp32, however as it stands only triplet seeding can benefit from fp16 and so it is better to pay the arithmetic price once in a non-memory bound kernel. Using fp16 allows us to use `half-2` vectorized instructions on the GPU, processing two combinations at a time and leading to an efficient formulation that maximizes throughput.

The compatibility of physics results on GPU and CPU architectures is investigated by comparing the momentum and slopes of reconstructed tracks and is shown in Figure 14 for the reference tuning. The momentum relative difference is below 0.3 % for a wide range of momenta while the track slopes differences around 0.01 %, matching the per-mille level agreement required, as mentioned in the introduction. The compatibility is found to be insensitive to the specific algorithm tuning and to whether UT information is used or not.

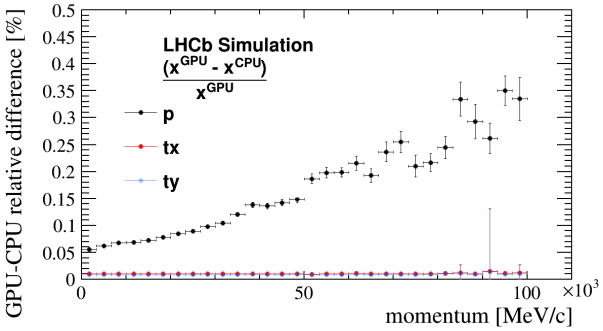

FIGURE 14. The relative difference between track parameters reconstructed on the A5000 GPU and EPYC 7502 CPU architectures, normalised to the results on the GPU architecture. Around 0.3% of tracks are found on only one architecture and are discarded for this comparison. The different parameters which are compared are described in the figure legend.

The 2022-2023 LHCb data-taking provided an inadvertent test of the Looking Forward algorithm's robustness because the UT detector could not be installed in time to participate. It was therefore necessary to reoptimize the algorithm to function without UT information, by extrapolating directly from the VELO to the FT as described in the previous section. This logic is analogous to that of the original LHCb forward tracking, used for data-taking from 2009 to 2012, which also extrapolated directly from the VELO to the stations downstream of the magnet. It was however thought [34] that such an approach would be impossible for HLT1 under Run 3 conditions because of the much greater event rate and number of *pp* collisions per event. Nevertheless it is proved possible [35] to use the algorithm's tunable parameters to achieve an overall throughput of ∼130 kHz for the HLT1 sequence as a whole, thus validating its fundamental robustness. The Looking Forward "no-UT" employs around 40 % of the HLT1 throughput, twice the amount of the nominal algorithm as more hit combinations are computed due to the lack of the UT information. The physics performance of a "no-UT" Looking Forward tuning is compared to that of the reference tuning in Figure 16 as function of momentum and in Figure 15 as a function of p_T. The fake rate roughly triples for the same efficiency, which is expected since the UT hits are essential in discriminating between correct and false matches of VELO and FT track segments. Nevertheless the fake rate remains below 15% for most tracks. The momentum and track slopes resolutions are shown in Figure 17 which can be compared to Figure 10 and shows only a modest degradation.

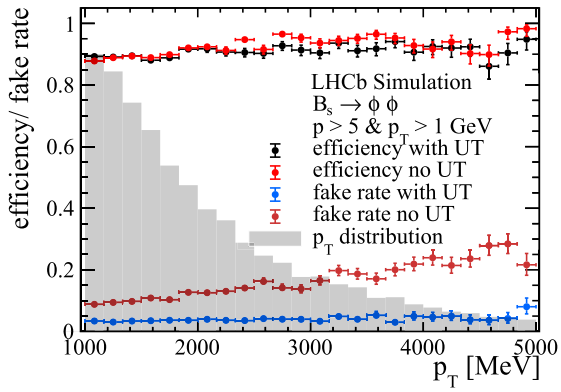

FIGURE 15. The efficiency and fake rate plotted as a function of particle p_T for the with-UT and no-UT Looking Forward configurations. The different algorithms are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

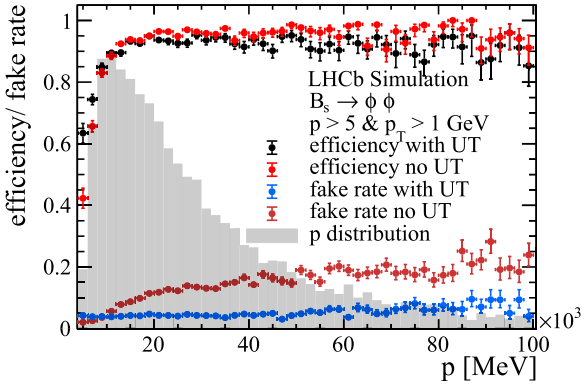

FIGURE 16. The efficiency and fake rate plotted as a function of particle momentum for the with-UT and no-UT Looking Forward configurations. The different algorithms are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

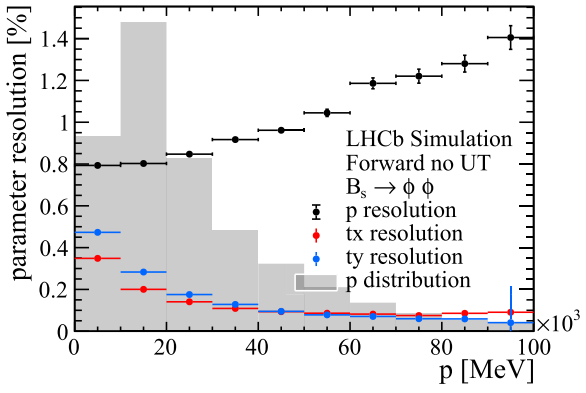

FIGURE 17. The momentum and track slope at the last SciFi detector layer resolutions of the Looking Forward algorithm as tuned for the no-UT HLT1 sequence, plotted as a function of particle p. The different components are described in the figure legend.

To exploit the second GPU card installed by LHCb in 2023, tradeoffs between computational cost and physics performance are studied in HLT1 by implementing a best long track reconstruction similarly on how tracks are reconstructed in HLT2 [8]. This algorithm first reconstruct tracks with the Looking Forward method, then flags all the unused hits in the detector and finally reconstruct with SciFi seeding and matching GPU-optimised strategy [36]. As shown in Figure 18, the seeding and matching method performs better at lower momenta while the forward one at higher momenta. The best long track reconstruction tries to exploit the advantages of both approaches.

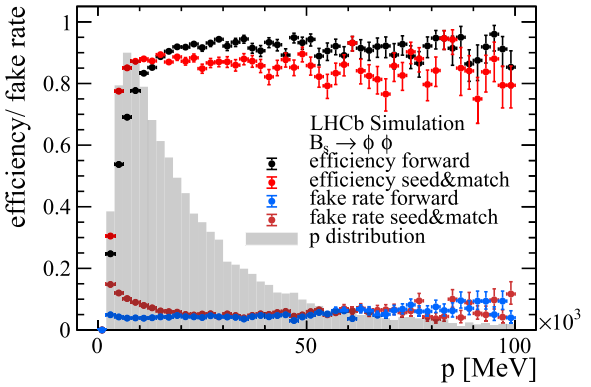

FIGURE 18. The efficiency and fake rate plotted as a function of particle momentum for the forward and seeding & matching configurations. The different algorithms are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

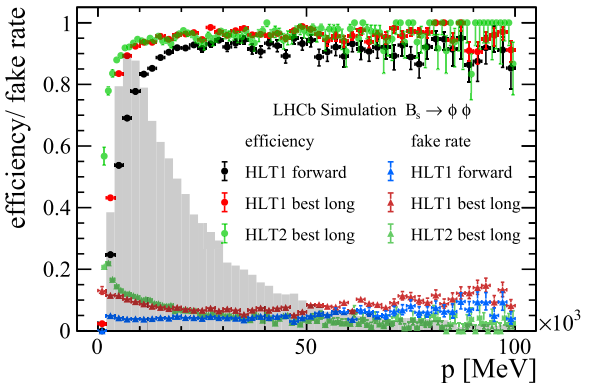

FIGURE 19. The efficiency and fake rate plotted as a function of momentum of particles for the forward and best long track reconstruction in HLT1 and HLT2. The different algorithms are described in the figure legend. The distribution of reconstructible charged particles, normalised to unit area, is shown as a shaded histogram to give an idea of the relative physics importance of different kinematic regions.

The overall HLT1 sequence throughput is reduced by around 30 % reaching 90 kHz, with all the additional resource usage allocated to the long track reconstruction. This is an extreme scenario and not necessarily representative of how the collaboration will use its resources, but rather provides an upper bound on these tradeoffs. Figure 19 shows the efficiency and fake rate as a function of particle momentum, comparing the Looking Forward algorithm with the best long track reconstruction configurations implemented in HLT1 and HLT2. The best long track sequence improves the tracking efficiency of the HLT1 reconstruction at low

momentum. The long tracking efficiency of HLT2 remains higher as lower momentum requirements are applied when performing the reconstruction and a full Kalman filter is exploited to improve the track resolution.

VIII. PROSPECTS AND CONCLUSION

We have presented Looking Forward, a new algorithm exploiting the track following approach and optimised on parallel GPU architectures. We developed the algorithm in order to maximise the throughput while achieving the best physics performance.

The method extrapolates tracks reconstructed before the LHCb dipole magnet to SciFi tracker after the magnet. The algorithm parellalises the search for hits in SciFi detector over the input tracks. The hits found in this way are combined in parallel to form candidate tracklets, which are then combined to the input tracks. The method achieves a 90 % tracking efficiency across a large spectrum of momenta and a momentum resolution below 1 %. The measured throughput on a A5000 card is ∼130 kHz.

The algorithm features tunable parameters which can be adapted on the physics requirements. The absence of the UT subdetector during the 2022 data-taking provided a stress test to the algorithm which was adapted to handle the missing information. The method in such configuration maintains the physics and computational performances in a sub-range of tracks with momentum greater than 5 GeV/c and transverse momentum greater than 1 GeV/c.

The algorithm was included and commissioned during the 2022 LHCb data-taking, and planned to be used in production in the coming years. We will continue exploring techniques to obtain better performances in the current and upcoming hardware generations.

ACKNOWLEDGMENT

The authors would like to thank LHCb's Real-Time Analysis Project for its support, for many useful discussions, and for reviewing an early draft of this manuscript and also would like to thank the LHCb computing and simulation teams for producing the simulated LHCb samples used to benchmark the performance of the algorithm presented in this article. The development and maintenance of LHCb's nightly testing and benchmarking infrastructure which their work relied on is a collaborative effort and also they are grateful to all LHCb colleagues who contribute to it.

REFERENCES

[1] J. Dorenbosch, "Trigger in UA2 and in UA," in *Proc. eConf*, 1985, pp. 134–151.
[2] R. Aaij et al., "Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC," *J. Instrum.*, vol. 14, no. 4, Apr. 2019, Art. no. P04013, doi: 10.1088/1748-0221/14/04/P04013.
[3] The ATLAS Collaboration, "Operation of the ATLAS trigger system in Run 2," *J. Instrum.*, vol. 15, no. 10, Oct. 2020, Art. no. P10004, doi: 10.1088/1748-0221/15/10/P10004.
[4] V. Khachatryan et al., "The CMS trigger system," *J. Instrum.*, vol. 12, no. 1, Jan. 2017, Art. no. P01020, doi: 10.1088/1748-0221/12/01/P01020.
[5] R. Aaij et al., "The LHCb upgrade I," May 2023, *arXiv:2305.10515*.
[6] T. Colombo, P. Durante, D. Galli, U. Marconi, N. Neufeld, F. Pisani, R. Schwemmer, S. Valat, and B. Voneki, "The LHCb DAQ upgrade for LHC Run 3," *IEEE Trans. Nucl. Sci.*, vol. 66, no. 7, pp. 982–985, Jul. 2019, doi: 10.1109/TNS.2019.2920393.
[7] M. Benayoun and O. Callot, "The forward tracking, an optical model method," Tech. Rep. LHCb-2002-008, CERN-LHCb-2002-008, Feb. 2002.
[8] A. Gunther. (2023). *Track Reconstruction Development and Commissioning for LHCb's Run 3 Real-Time Analysis Trigger*. [Online]. Available: https://cds.cern.ch/record/2865000
[9] O. Callot and S. Hansmann-Menzemer, "The forward tracking: Algorithm and performance studies," Tech. Rep. LHCb-2007-015, CERN-LHCb-2007-015, May 2007.
[10] The ATLAS Collaboration, "The ATLAS experiment at the CERN large hadron collider: A description of the detector configuration for Run 3," 2023, *arXiv:2305.16623*.
[11] The CMS Collaboration, "Development of the CMS detector for the CERN LHC Run 3," 2023, *arXiv:2309.05466*.
[12] (2021). *Software Performance of the ATLAS Track Reconstruction for LHC Run 3*. [Online]. Available: https://cds.cern.ch/record/2766886
[13] (2023). *Performance of Track Reconstruction at the CMS High-Level Trigger in 2022 Data*. [Online]. Available: https://cds.cern.ch/record/2860207
[14] P. Antonioli, A. Kluge, and W. Riegler, "Upgrade of the ALICE readout rigger system," ALICE, Tech. Rep. CERN-LHCC-2013-019, ALICE-TDR-015, 2013.
[15] Collaboration ATLAS, "Technical design report for the phase-II upgrade of the ATLAS trigger and data acquisition system—Event filter tracking amendment," CERN, Geneva, Switzerland, Tech. Rep. ATLAS-TDR-026, 2022. [Online]. Available: https://cds.cern.ch/record/2802799
[16] CMS Collaboration, "The phase-2 upgrade of the CMS data acquisition and high level trigger," CERN, Geneva, Switzerland, Tech. Rep. CMS-CR-2021-076, 2021. [Online]. Available: https://cds.cern.ch/record/2759072
[17] CERN (Meyrin) LHCb Collaboration, "Framework TDR for the LHCb upgrade II—Opportunities in flavour physics, and beyond, in the HL-LHC era," CERN, Geneva, Switzerland, Tech. Rep. LHCB-TDR-023, 2021. [Online]. Available: https://cds.cern.ch/record/2776420
[18] CERN (Meyrin) LHCb Collaboration, "LHCb upgrade GPU high level trigger technical design report," CERN, Geneva, Switzerland, Tech. Rep. LHCB-TDR-021, 2020, doi: 10.17181/CERN.QDVA.5PIR. [Online]. Available: https://cds.cern.ch/record/2717938
[19] R. Aaij et al., "Allen: A high-level trigger on GPUs for LHCb," *Comput. Softw. Big Sci.*, vol. 4, no. 1, p. 7, Dec. 2020, doi: 10.1007/s41781-020-00039-7.
[20] R. Aaij et al., "Measurement of the track reconstruction efficiency at LHCb," *J. Instrum.*, vol. 10, no. 2, Feb. 2015, Art. no. P02007, doi: 10.1088/1748-0221/10/02/p02007.
[21] R. Aaij et al., "Measurement of the electron reconstruction efficiency at LHCb," *J. Instrum.*, vol. 14, no. 11, Nov. 2019, Art. no. P11023, doi: 10.1088/1748-0221/14/11/p11023.
[22] P. Li, E. Rodrigues, and S. Stahl, "Tracking definitions and conventions for Run 3 and beyond," CERN, Geneva, Switzerland, Tech. Rep. LHCb-PUB-2021-005, 2021. [Online]. Available: https://cds.cern.ch/record/2752971
[23] J. Andre, P. Charra, W. Flegel, P. Giudici, O. Jamet, P. Lancon, M. Losasso, F. Rohner, and C. Rosset, "Status of the LHCb magnet system," *IEEE Trans. Appl. Supercond.*, vol. 12, no. 1, pp. 366–371, Mar. 2002, doi: 10.1109/TASC.2002.1018421.
[24] J. Andre, W. Flegel, P. Giudici, O. Jamet, and M. Losasso, "Status of the LHCb dipole magnet," *IEEE Trans. Appl. Supercond.*, vol. 14, no. 2, pp. 509–513, Jun. 2004, doi: 10.1109/TASC.2004.
[25] S. Aiola, Y. Amhis, P. Billoir, B. K. Jashal, L. Henry, A. O. Campos, C. M. Benito, F. Polci, R. Quagliani, M. Schiller, and M. Wang, "Hybrid seeding: A standalone track reconstruction algorithm for scintillating fibre tracker at LHCb," *Comput. Phys. Commun.*, vol. 260, Mar. 2021, Art. no. 107713, doi: 10.1016/j.cpc.2020.107713.
[26] D. H. C. Pérez, N. Neufeld, and A. R. Núñez, "Search by triplet: An efficient local track reconstruction algorithm for parallel architectures," *J. Comput. Sci.*, vol. 54, Sep. 2021, Art. no. 101422, doi: 10.1016/j.jocs.2021.101422.
[27] P. F. Declara, D. H. C. Pérez, J. Garcia-Blas, D. V. Bruch, J. D. García, and N. Neufeld, "A parallel-computing algorithm for high-energy physics particle tracking and decoding using GPU architectures," *IEEE Access*, vol. 7, pp. 91612–91626, 2019, doi: 10.1109/ACCESS.2019.2927261.

[28] E. Bowen and B. Storaci, "VeloUT tracking for the LHCb upgrade," CERN, Geneva, Switzerland, Tech. Rep. LHCb-PUB-2013-023, 2014. [Online]. Available: https://cds.cern.ch/record/1635665
[29] R. Quagliani. (2017). *Study of Double Charm B Decays With the LHCb Experiment at CERN and Track Reconstruction for the LHCb Upgrade*. [Online]. Available: https://cds.cern.ch/record/2296404
[30] C. Hasse. (2019). *Alternative Approaches in the Event Reconstruction of LHCb*. [Online]. Available: https://cds.cern.ch/record/2706588
[31] W. Kutta, "Beitrag zur näherungsweisen integration totaler differentialgleichungen," *Zeitschrift Mathematik Physik*, vol. 46, pp. 435–453, 1901.
[32] P. A. Günther, "LHCb's forward tracking algorithm for the Run 3 CPU-based online track-reconstruction sequence," 2022, *arXiv:2207.12965*.
[33] R. Aaij et al., "A comparison of CPU and GPU implementations for the LHCb experiment Run 3 trigger," *Comput. Softw. Big Sci.*, vol. 6, no. 1, p. 1, Dec. 2022, doi: 10.1007/s41781-021-00070-2.
[34] The LHCb Collaboration, *LHCb Trigger and Online Upgrade Technical Design Report*, CERN, Meyrin, Switzerland, May 2014.
[35] A. Scarabotto. (2022). *Tracking on GPU at LHCb's Fully Software Trigger*. [Online]. Available: https://cds.cern.ch/record/2823783
[36] L. Calefice. (2022). *Standalone Track Reconstruction on GPUs in the First Stage of the Upgraded LHCb Trigger System and Preparations for Measurements With Strange Hadrons in Run 3*. [Online]. Available: https://cds.cern.ch/record/2856339

AURELIEN BAILLY-REYRE received the Ph.D. degree in physics from the University of Cergy-Pontoise. He is currently a Computer Science Engineer with the LPNHE Laboratory, Sorbonne University. He is busy as a Cloud and Grid Administrator. He also has experience in high performance computing technologies.

LINGZHU BIAN received the Ph.D. degree in physics from Wuhan University. Her research interests include the study of magnetic field parametrizations for fast reconstruction algorithms and indirect searches for physics beyond the standard model. She is currently an Engineer with IHEP, mainly responsible for the development of the experimental control and data acquisition system for HEPS.

PIERRE BILLOIR is currently an Emeritus Professor in physics with Sorbonne Université. His research interest includes fast track fitting in high-energy physics environments together with the evaluation of magnetic field parametrizations.

DANIEL HUGO CÁMPORA PÉREZ received the Ph.D. degree in computer engineering from the University of Seville. He is currently a Senior AI Devtech Engineer with NVIDIA. His main research interest includes the optimizations of high-throughput real-time processes in physics reconstruction with parallel architectures.

VLADIMIR VAVA GLIGOROV received the Doctorate degree from the University of Oxford and the Habilitation degree from Sorbonne Université. He is currently the CNRS Research Director of the LPNHE Laboratory, Paris. His research interests include real-time data processing and direct and indirect searches for physics beyond the standard model.

FLAVIO PISANI received the Ph.D. degree in physics from the University of Bologna "Alma Mater Studiorum." He is currently a Staff Member with CERN. He is also a Physicist specialized in data acquisition systems and high-throughput interconnection technologies. He is also responsible for the DAQ network and software of the LHCb experiment.

RENATO QUAGLIANI received the Ph.D. degree from the University of Bristol and the Université Paris-Saclay, through the Co-Tutelle Programme. He is currently a Research Staff Member with CERN. He has been a Postdoctoral Researcher with Université Sorbonne and École Polytechnique Fédérale de Lausanne (EPFL). His main research interests include the implementation of efficient and fast track reconstruction algorithms and the study of rare and very rare decays of heavy flavor particles.

ALESSANDRO SCARABOTTO received the Ph.D. degree in physics from Sorbonne Université, Paris, France. He is currently a Postdoctoral Researcher with Dortmund Technische Universität working for the LHCb experiment with CERN. His research interest includes optimization of high-throughput particle reconstruction algorithms developed for GPUs.

DOROTHEA VOM BRUCH received the Doctorate degree from Heidelberg University. She is currently a Research Scientist with CNRS, Particle Physics Center of Marseille (CPPM). Her main research interests include real-time reconstruction and data selection systems optimized for GPUs and on testing the standard model of particle physics in semileptonic heavy flavor decays.