The upgraded LHCb trigger system

Dorothea vom Bruch
CPPM, Aix-Marseille Université, Marseille, CNRS/IN2P3

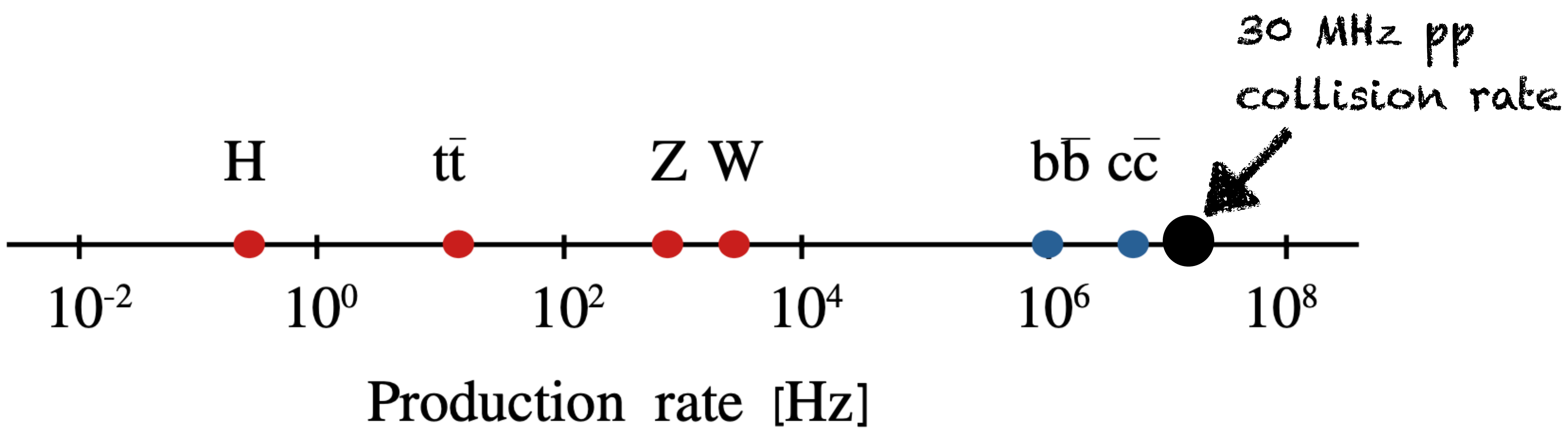
On behalf of the LHCb collaboration

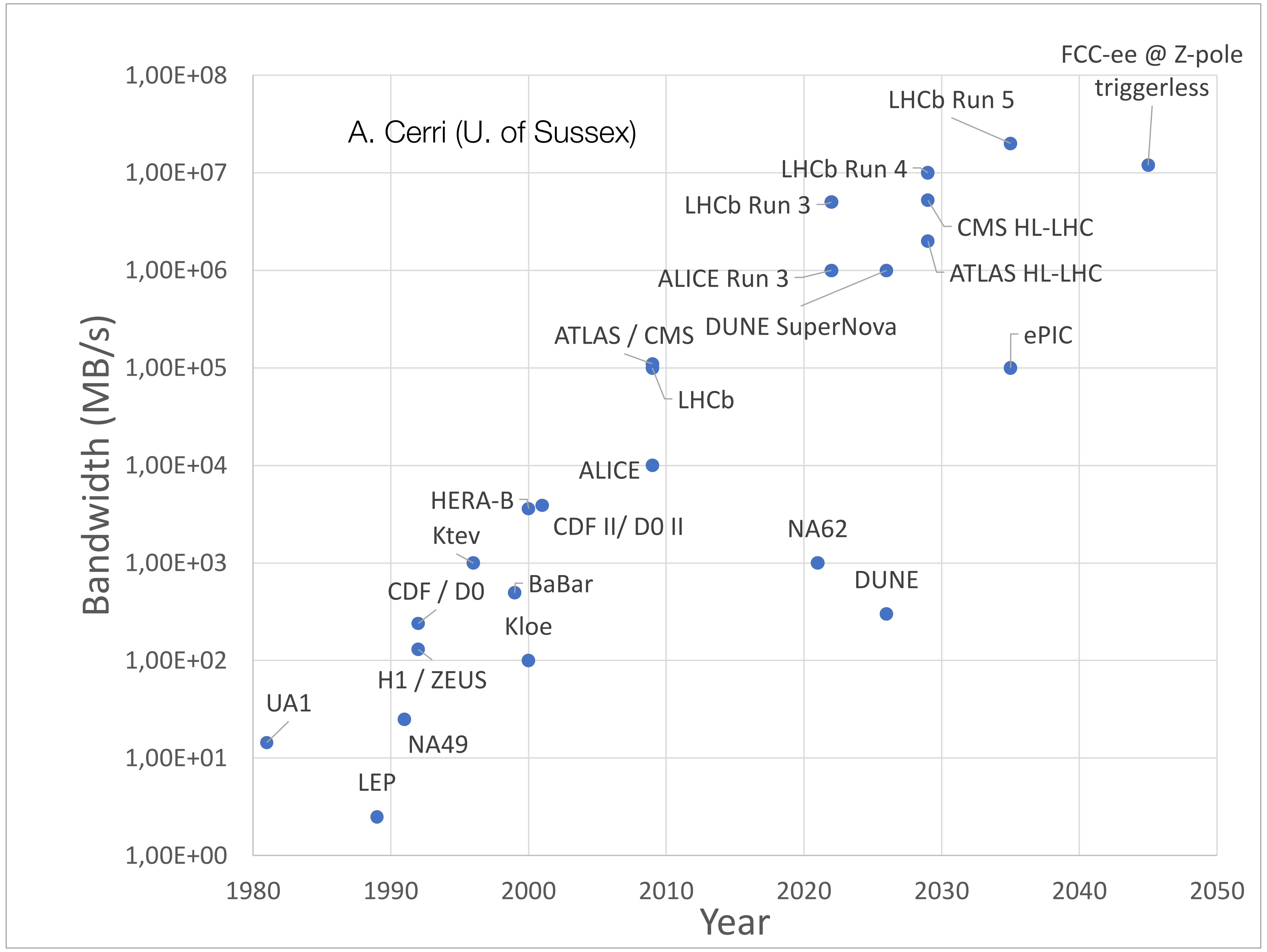
EPS-HEP 2025
July 7th 2025

The LHCb trigger challenge

$\mathcal{L}$ = 2x10^{34} cm^{-2}s^{-1} (ATLAS/CMS)   sqrt(s) = 13.6 TeV

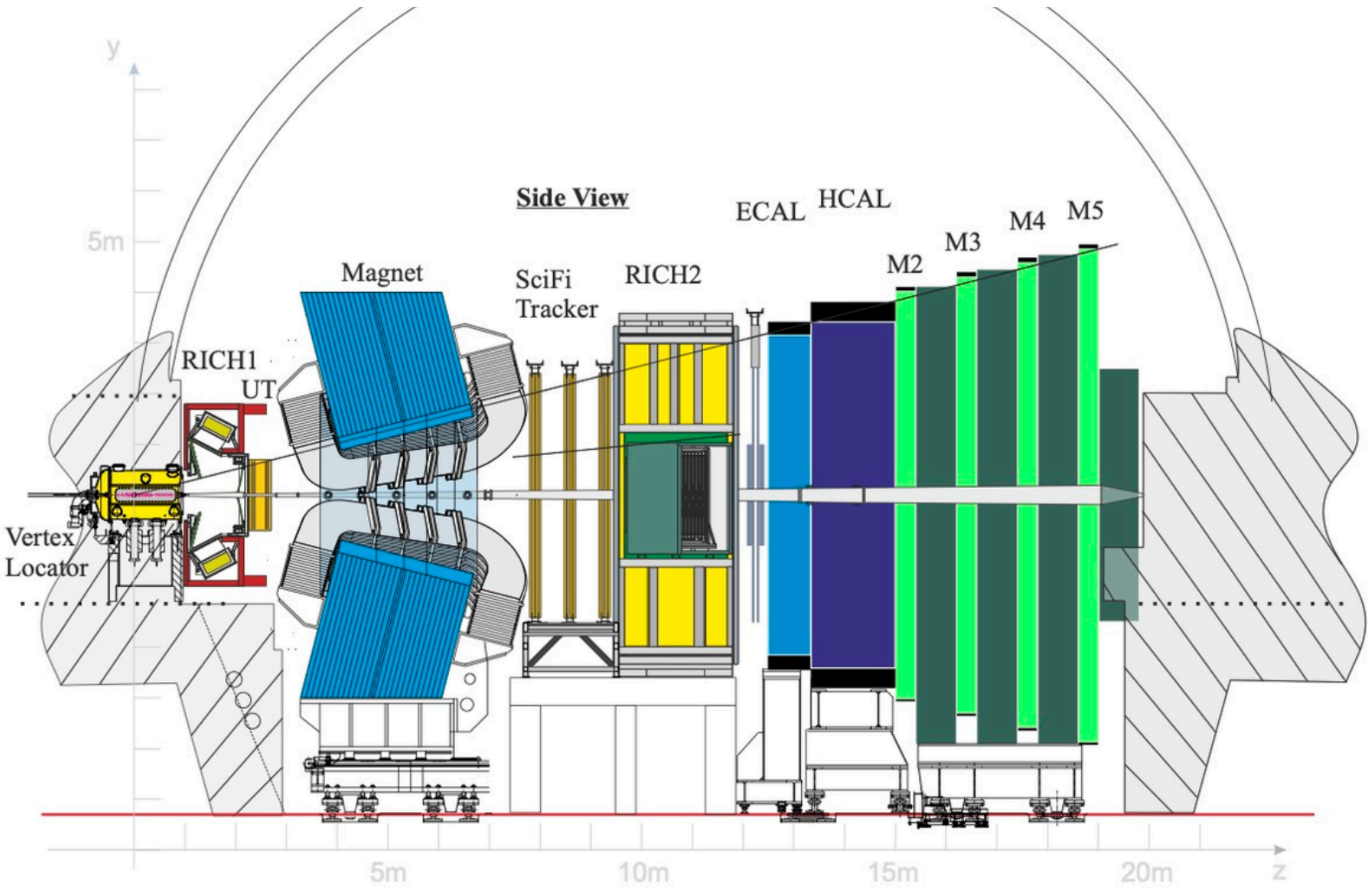
$\mathcal{L}$ = 2x10^{33} cm^{-2}s^{-1} (LHCb)



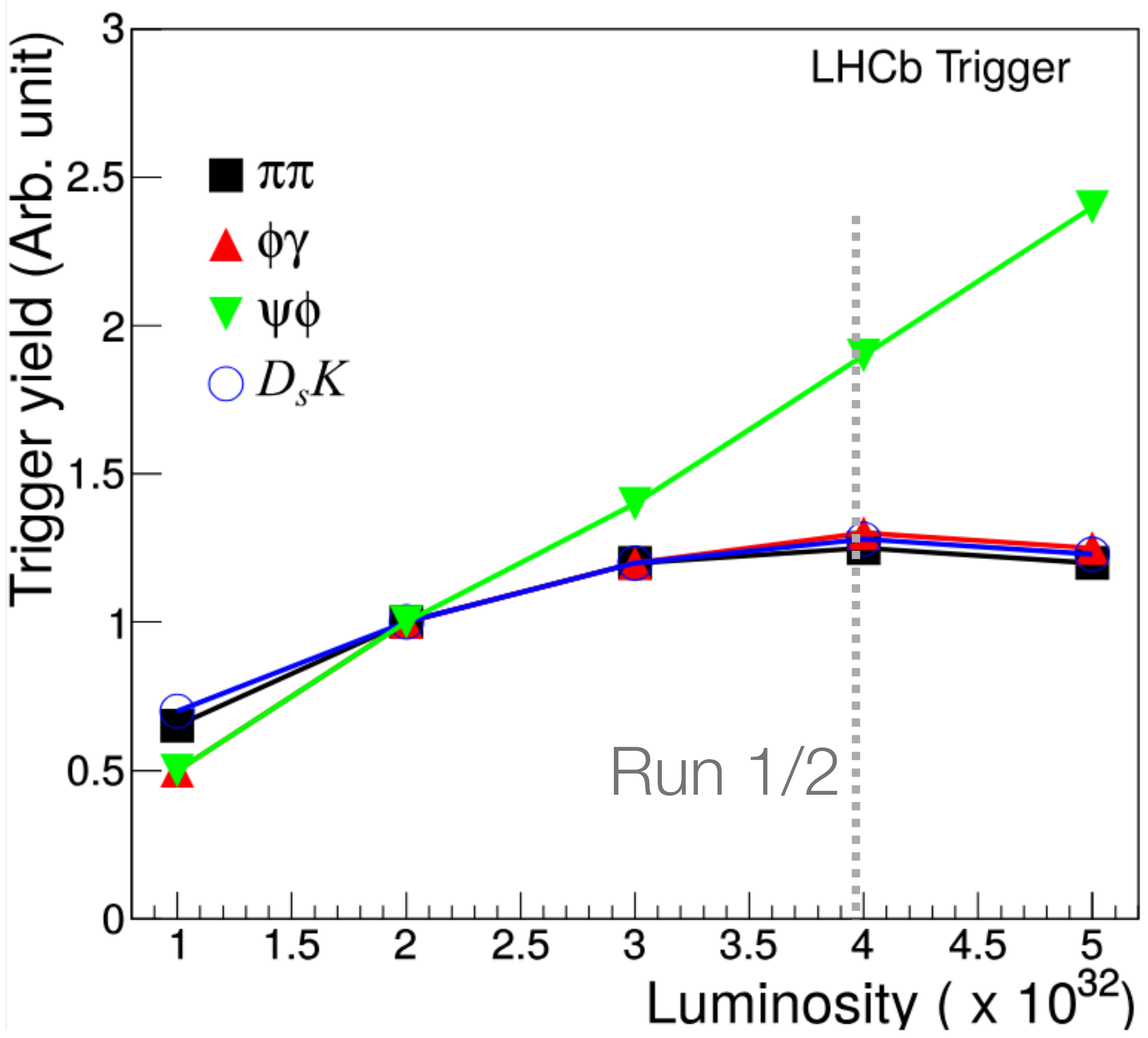


LHCb Run 3 Upgrade

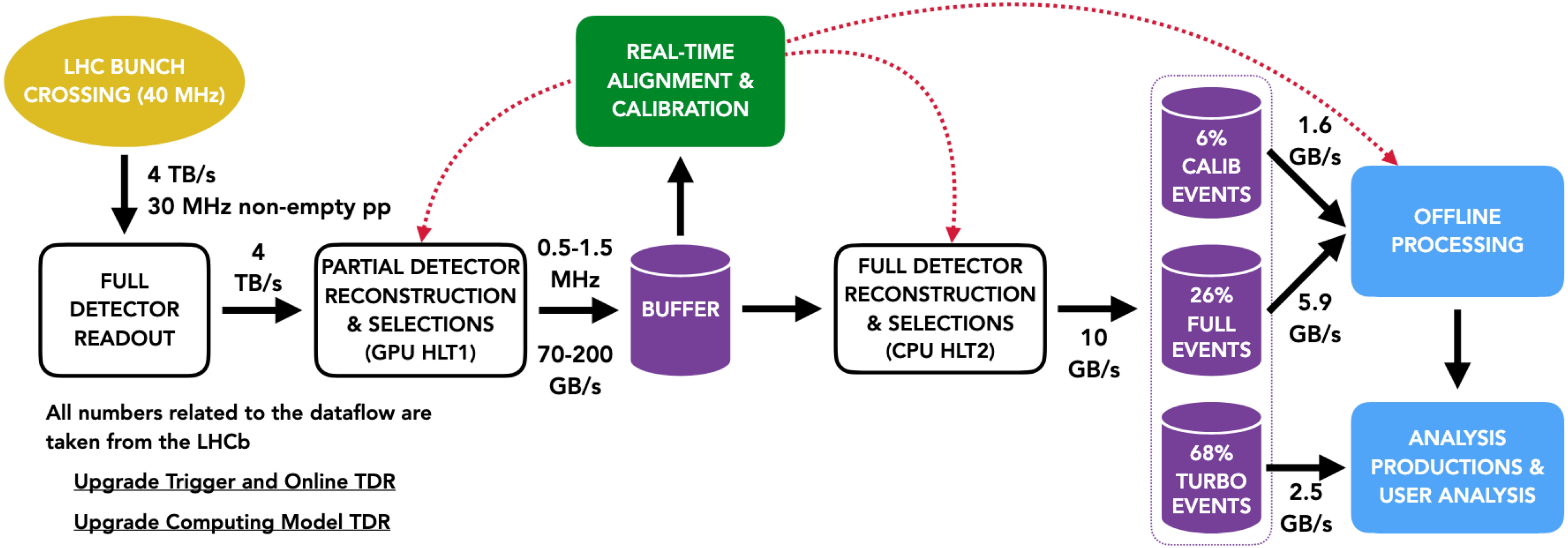
- Five times higher luminosity: 2×10^{33} cm^{-2}s^{-1}
- Increased pileup to <μ> ≈ 5



- Replacement of all tracking detectors
- Removal of hardware trigger
- Fully new readout electronics to read out 40 Tbit/s
- Charged particle reconstruction at 30 MHz in the full detector is necessary



The LHCb trigger in LHC Runs 3 & 4



All numbers related to the dataflow are taken from the LHCb

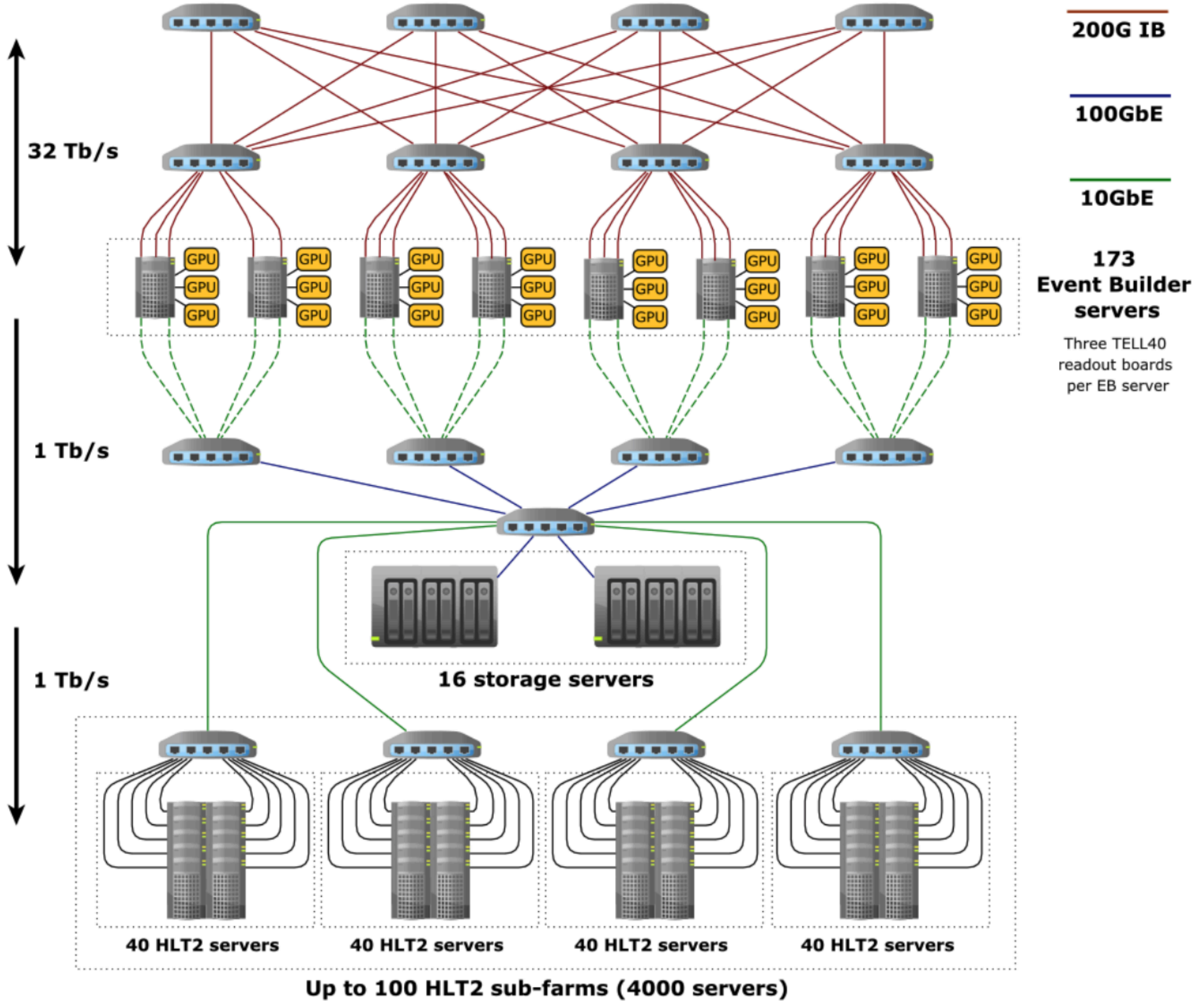
Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

LHCb-FIGURE-2020-016

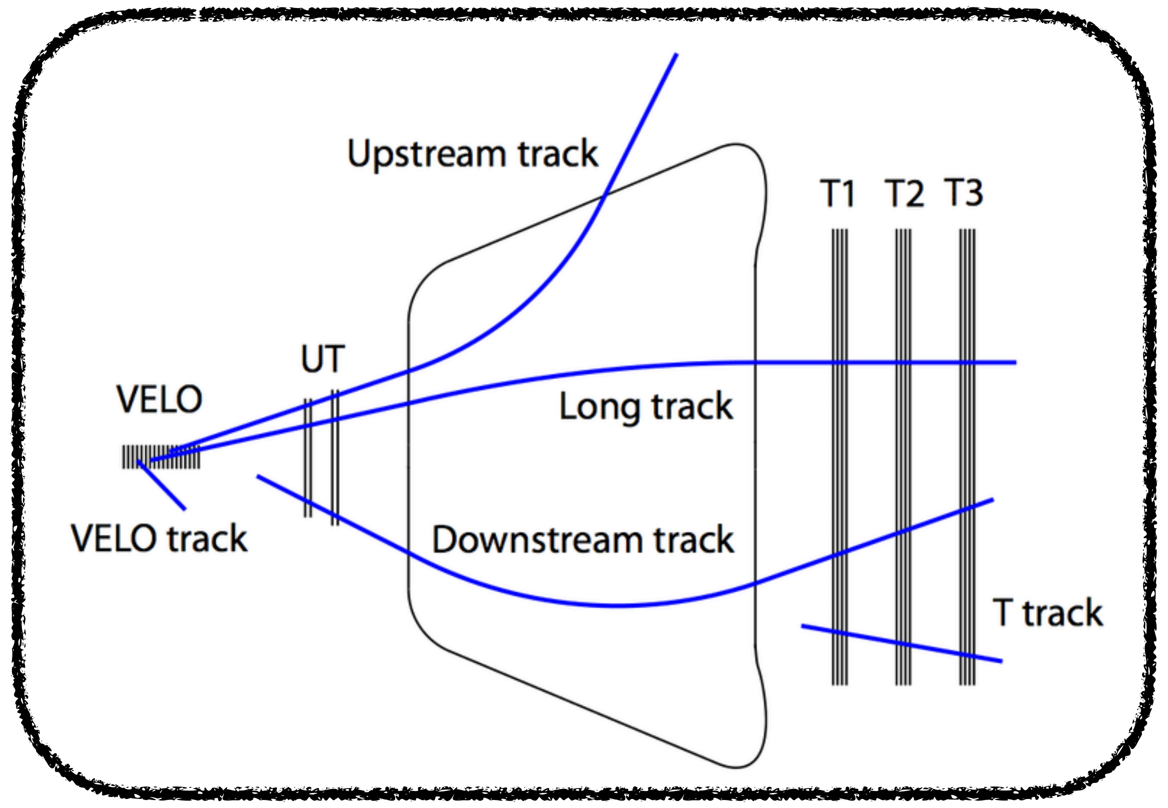
Heterogeneous computing system

- Raw detector data is received by FPGA cards
- 163 event builder servers aggregate data from sub-detectors
- Each server hosts 3 GPU cards on which HLT1 is run
    - Around 500 Nvidia A5000 in total
- Output of HLT1 is stored on storage servers while alignment & calibration is processed
- HLT2 is processed on computing farm with 250k CPU cores

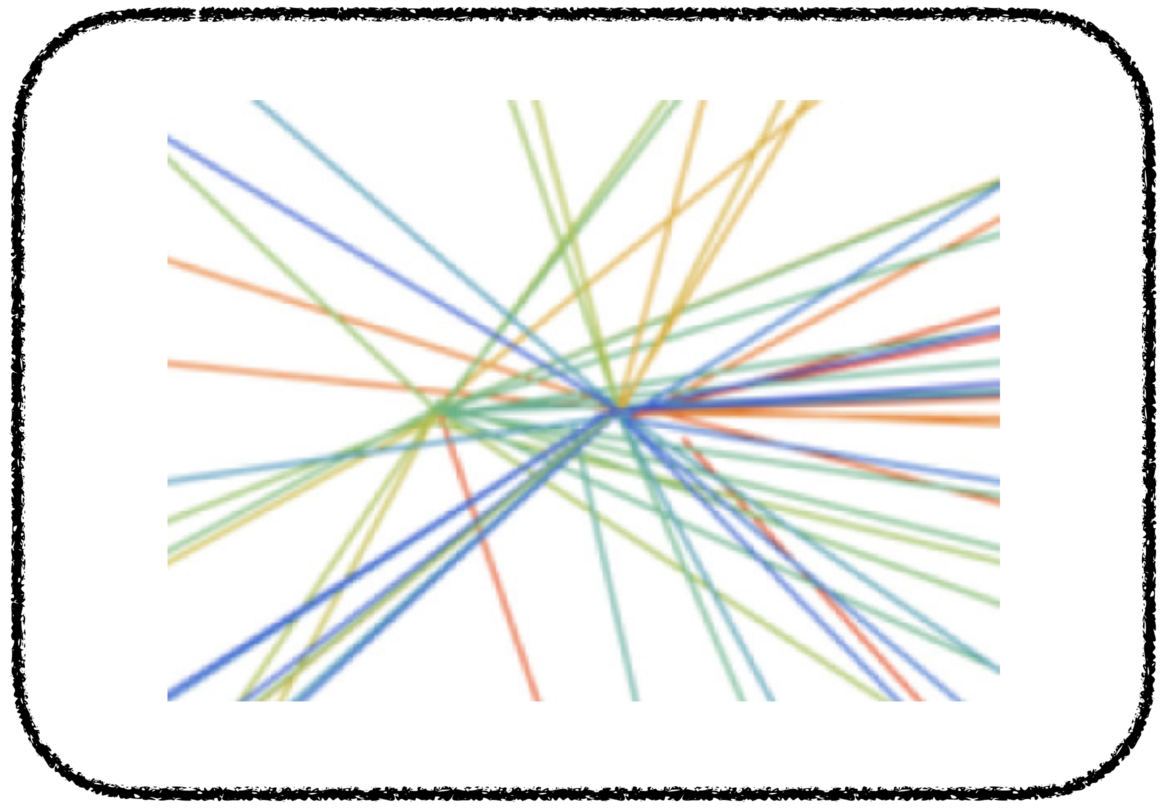


What do we reconstruct in the High Level Trigger (HLT)?

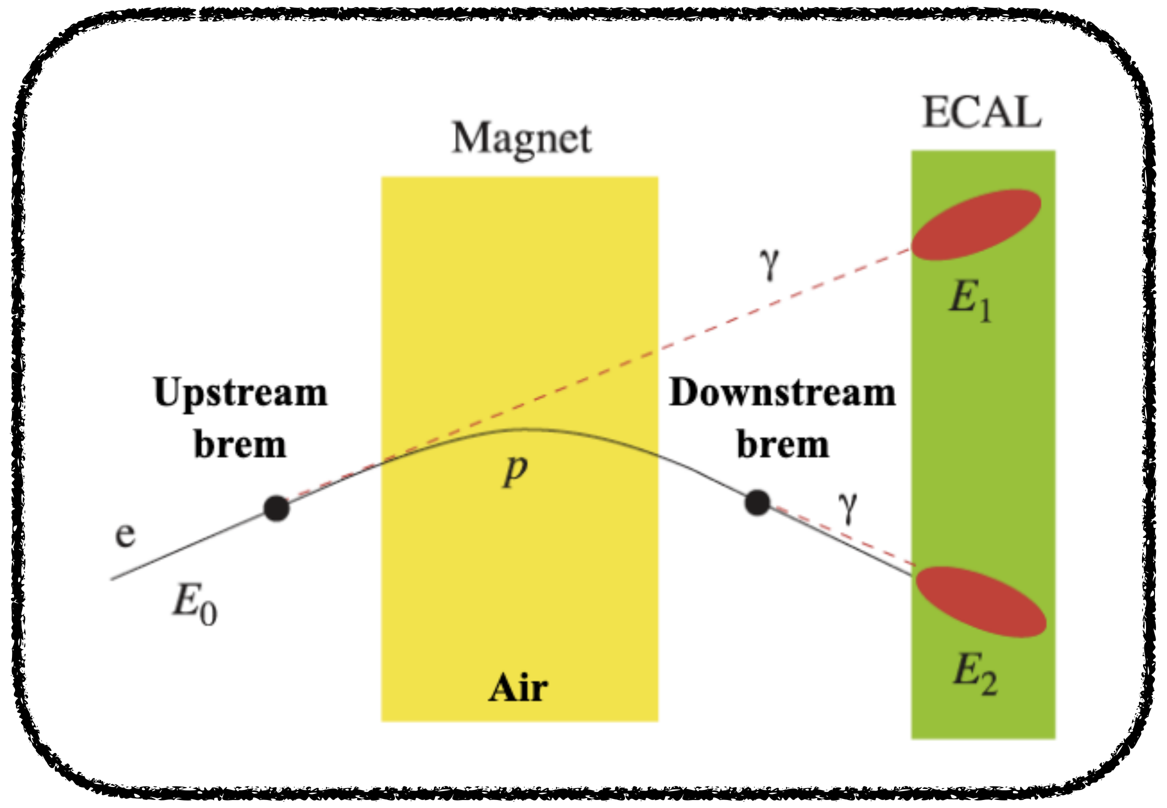
Tracks



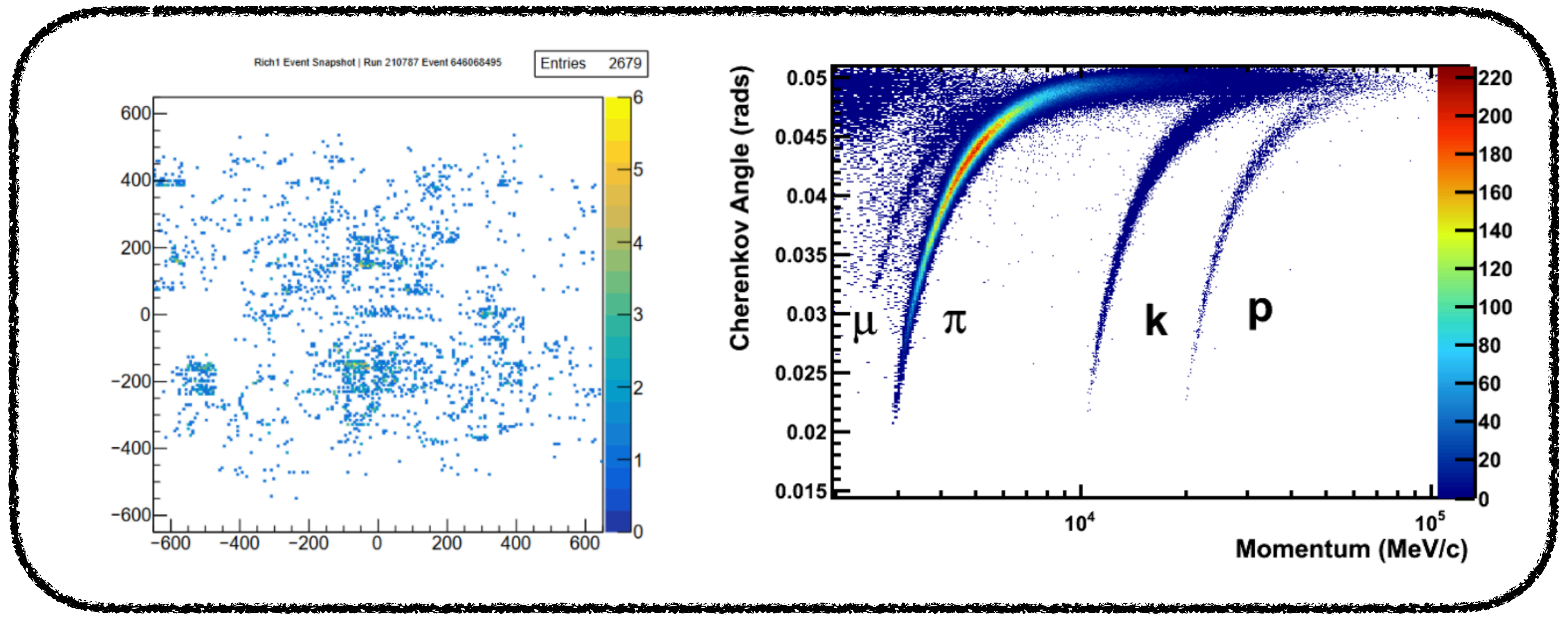
Vertices



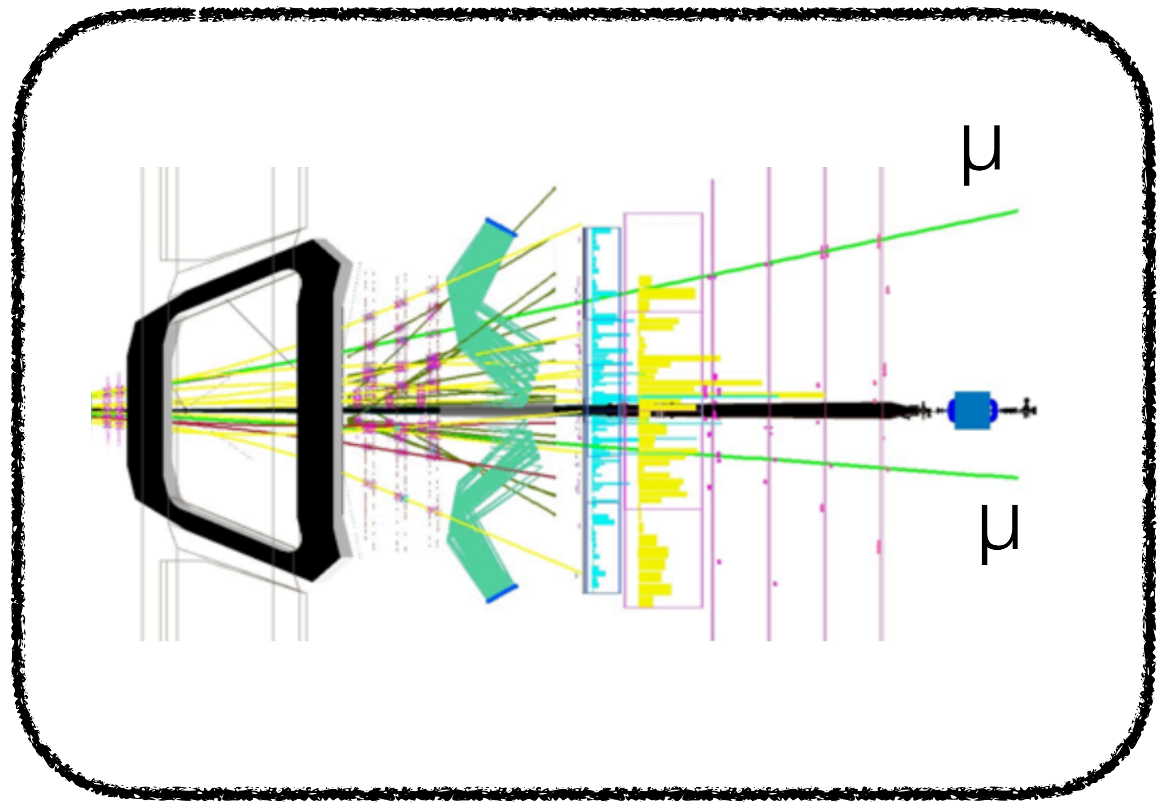
Calorimeter objects



Cherenkov rings



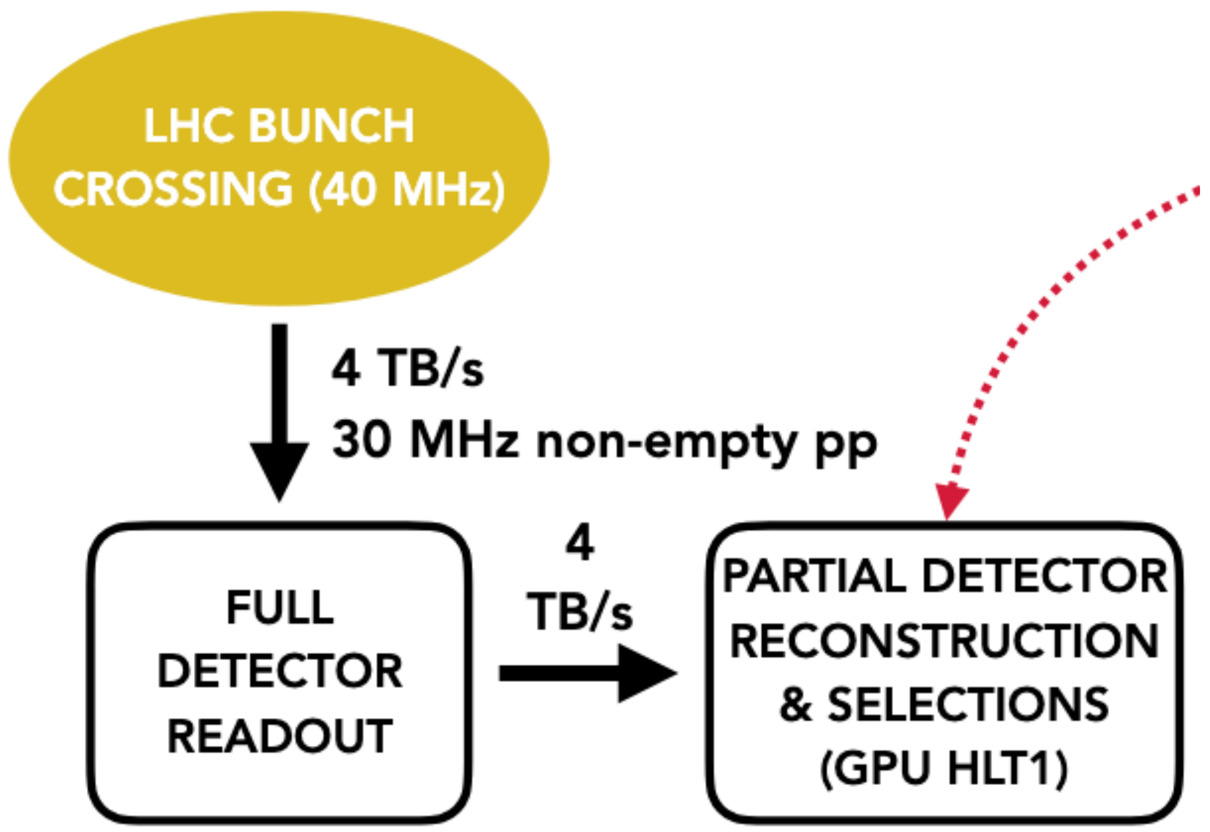
Muon hits



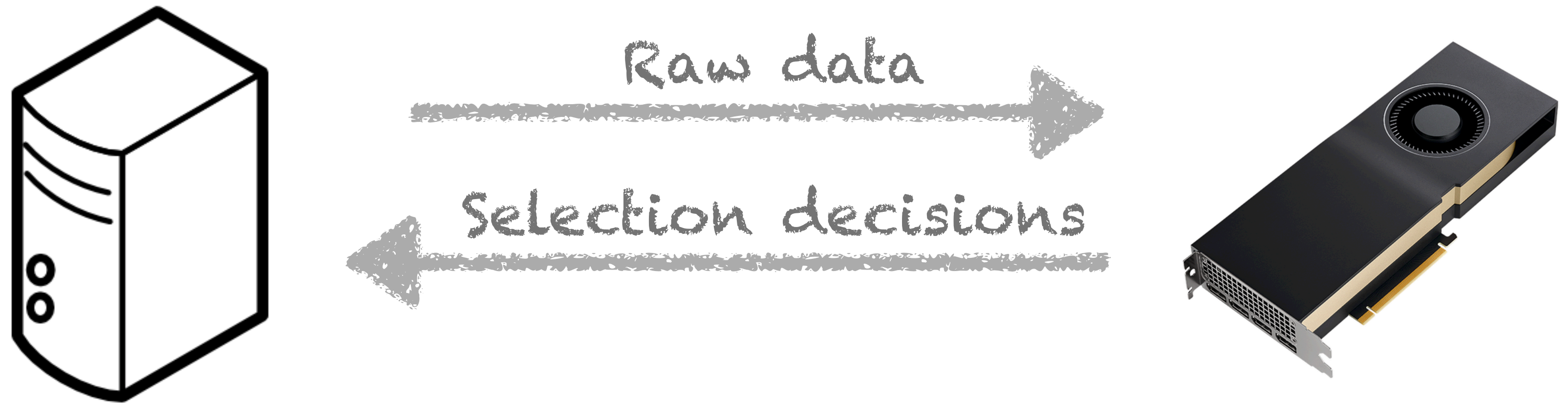
How does the HLT map to GPUs?

Characteristics of LHCb's HLT	Characteristics of GPUs
Intrinsically parallel problem: Process collisions and objects in parallel	Suited for: • Data-intensive parallelizable applications • High throughput applications
Huge compute load	Many TFLOPs available
Full data stream is read out →No stringent latency requirements	Higher latency than CPUs Not as predictable as FPGAs
Small raw event data (~100 kB)	• Connection via PCIe → limited I/O bandwidth • Thousands of events fit into O(10) GB of memory

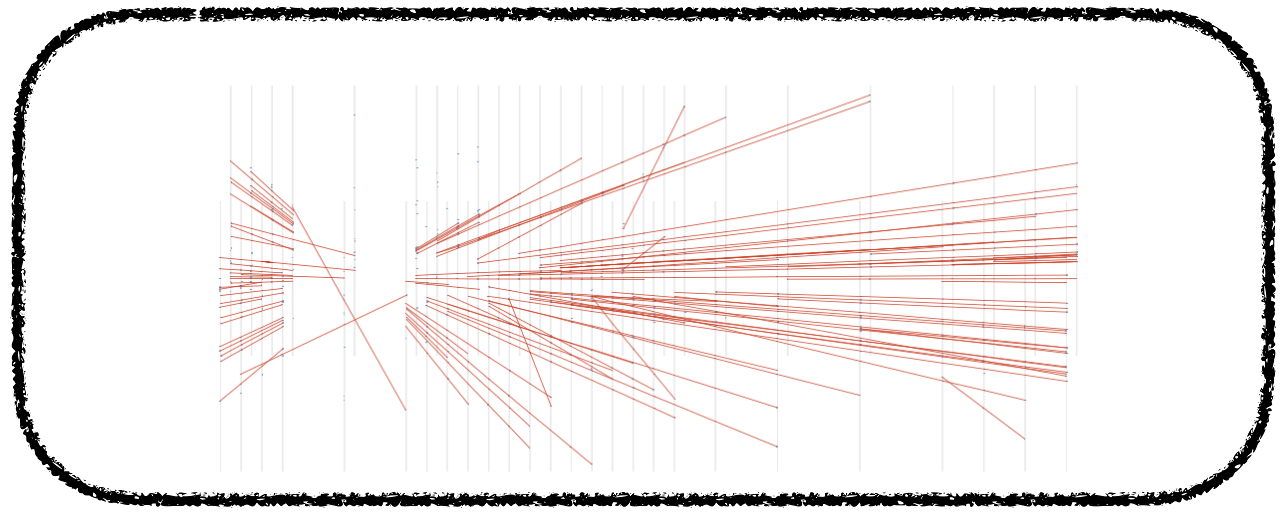
Design of the Allen GPU HLT1



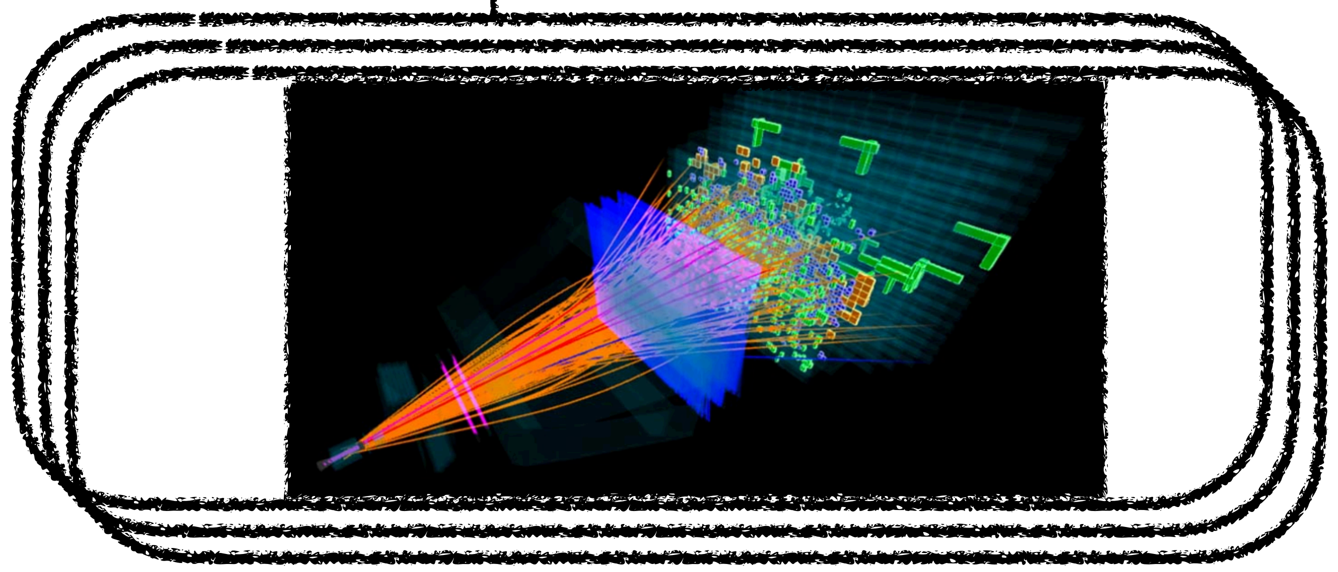
- Do all work on the GPU
  - Minimise copies to/from the GPU
- Parallelise on multiple levels
- Maximise GPU algorithm performance
  - Design software framework to optimise algorithm throughput performance
  - Interleave ML/AI and classical algorithms
  - Single precision only
- Execute on multiple compute architectures
- Simple event model



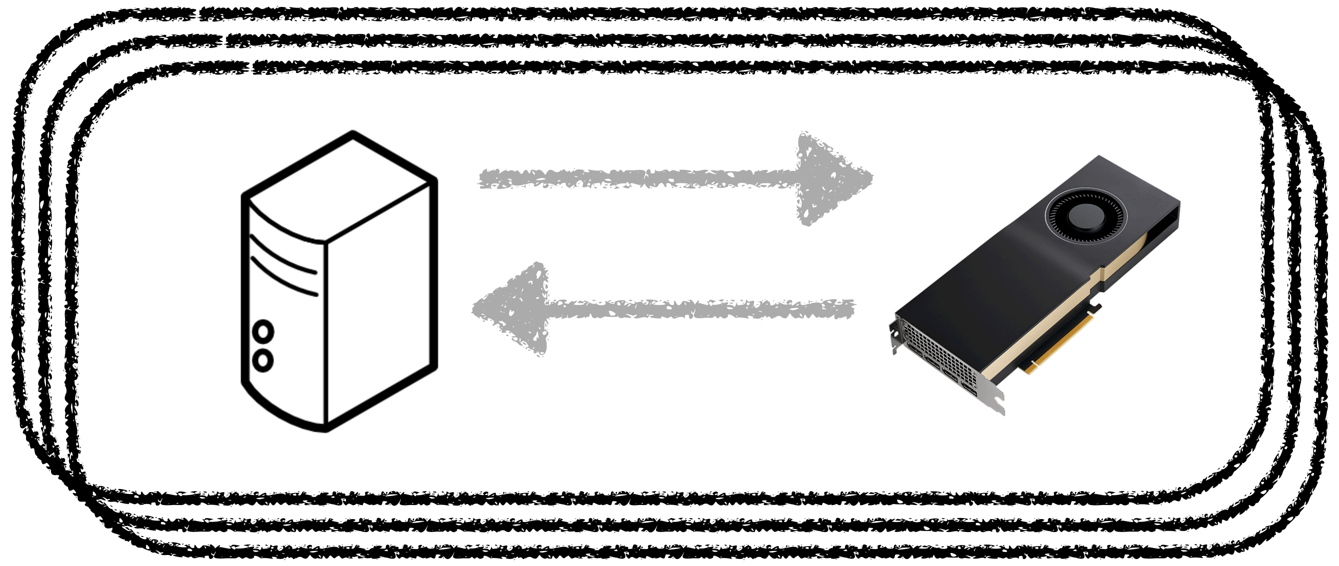
Intra collision: tracks, vertices, ...



Proton-proton collisions



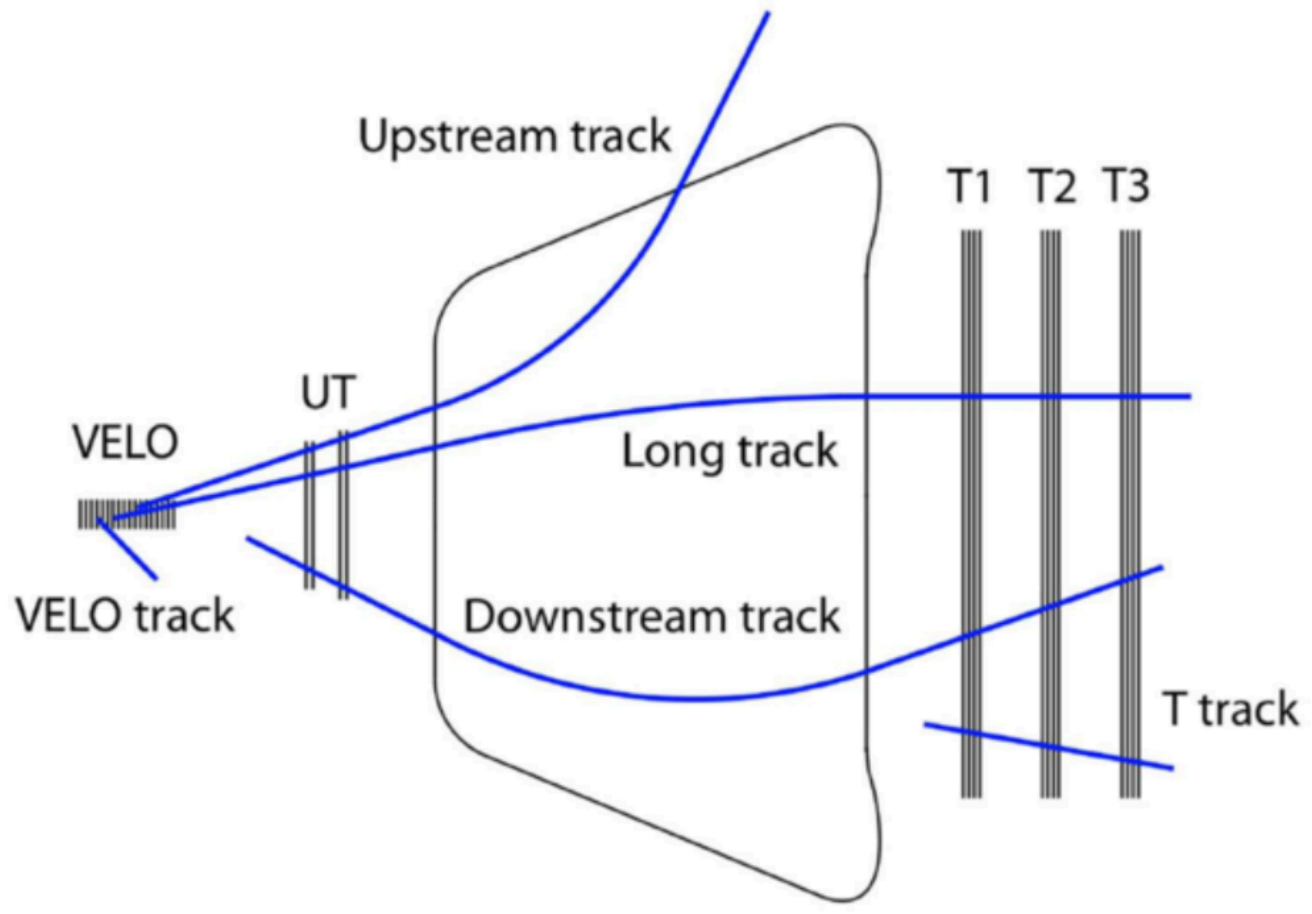
Collision batches

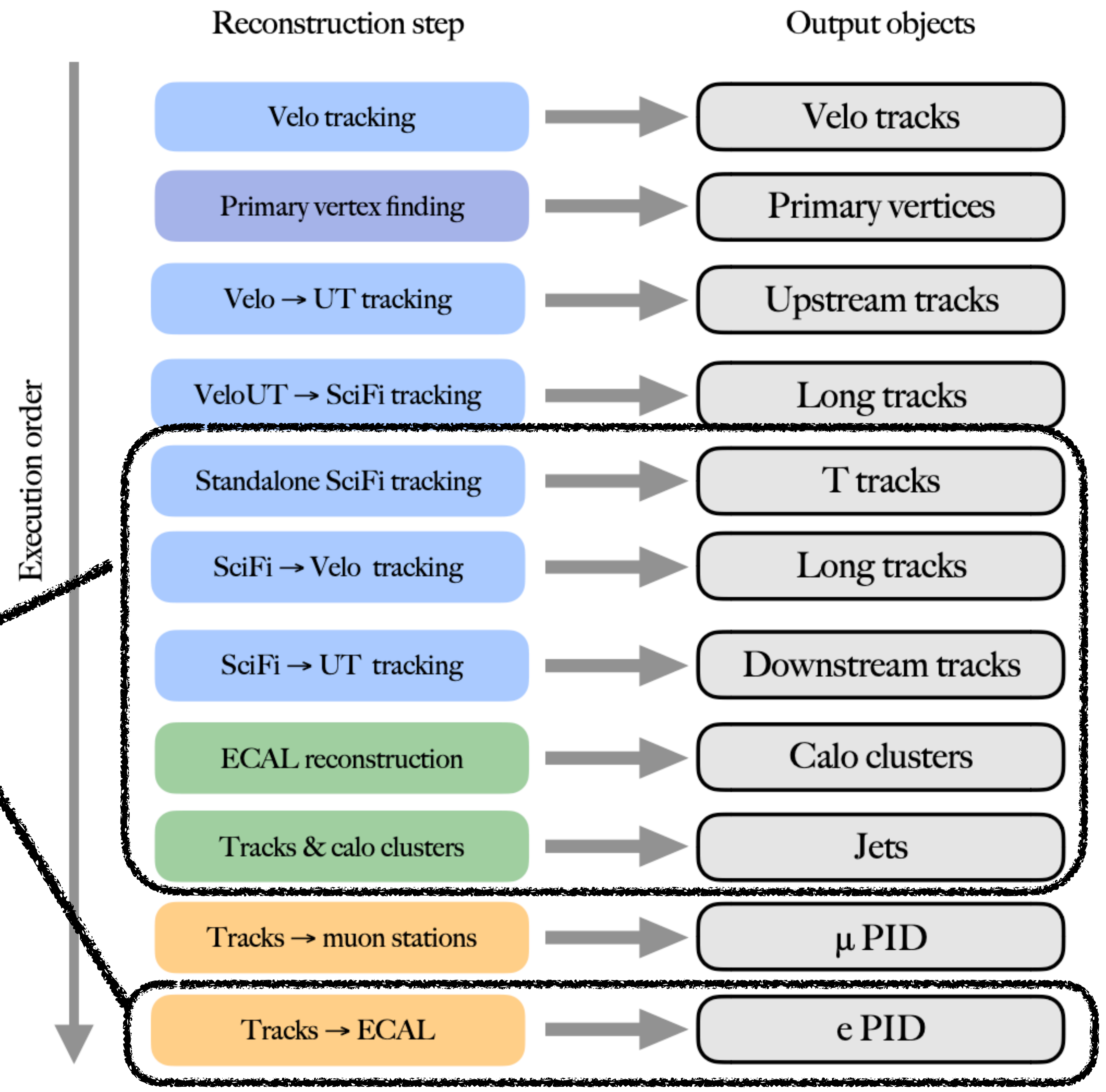


HLT1 reconstruction

Many algorithms beyond TDR design:

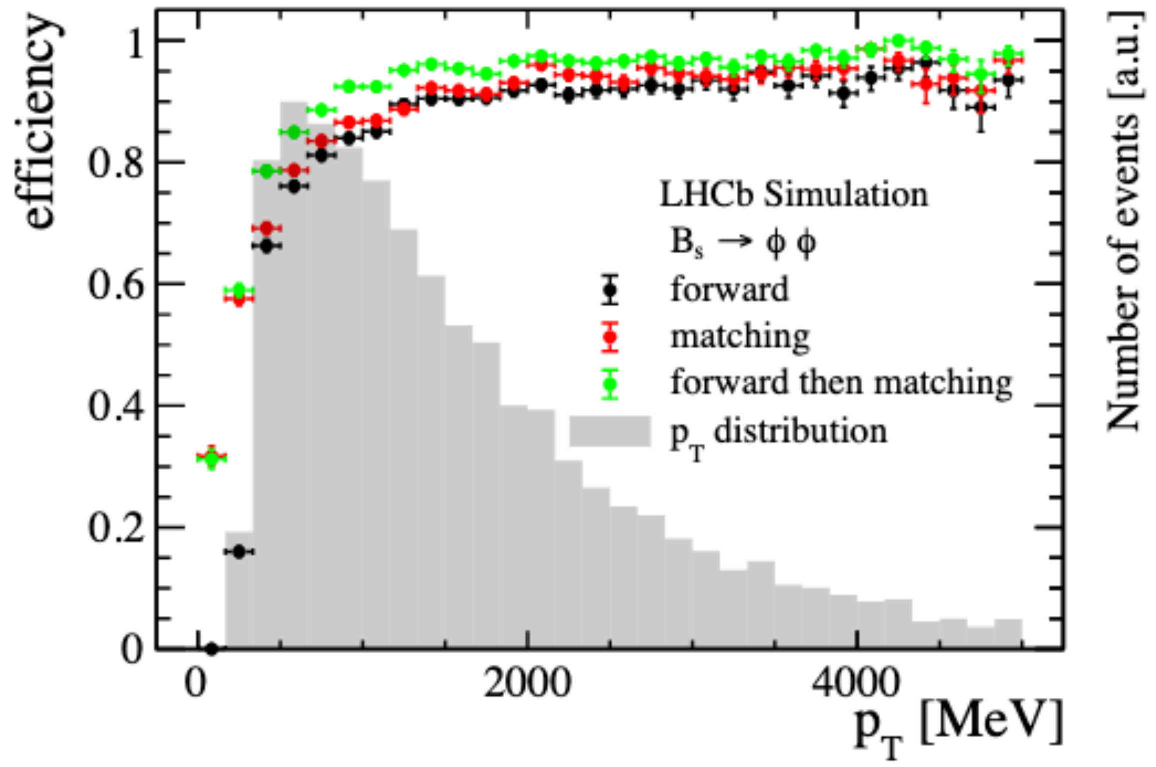
- Additional long track reconstruction method
  - Gained 20% efficiency at pT = 500 MeV
  - Robust with respect to detector occupancy
- Downstream track reconstruction
- Ecal & Jet reconstruction
- Kalman filter with parameterised scattering errors and transport through the magnetic field (since 2025)



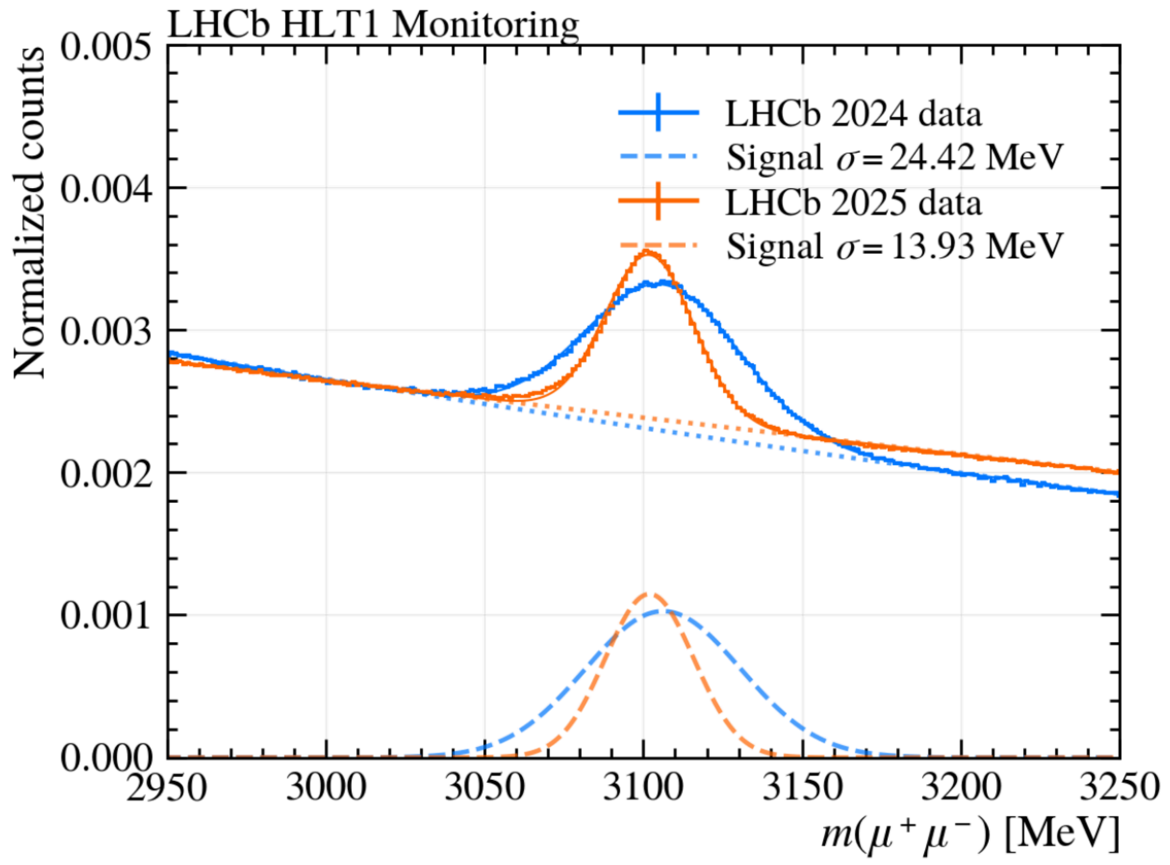


Beyond TDR reconstruction

Long tracks from b-hadrons in HLT1



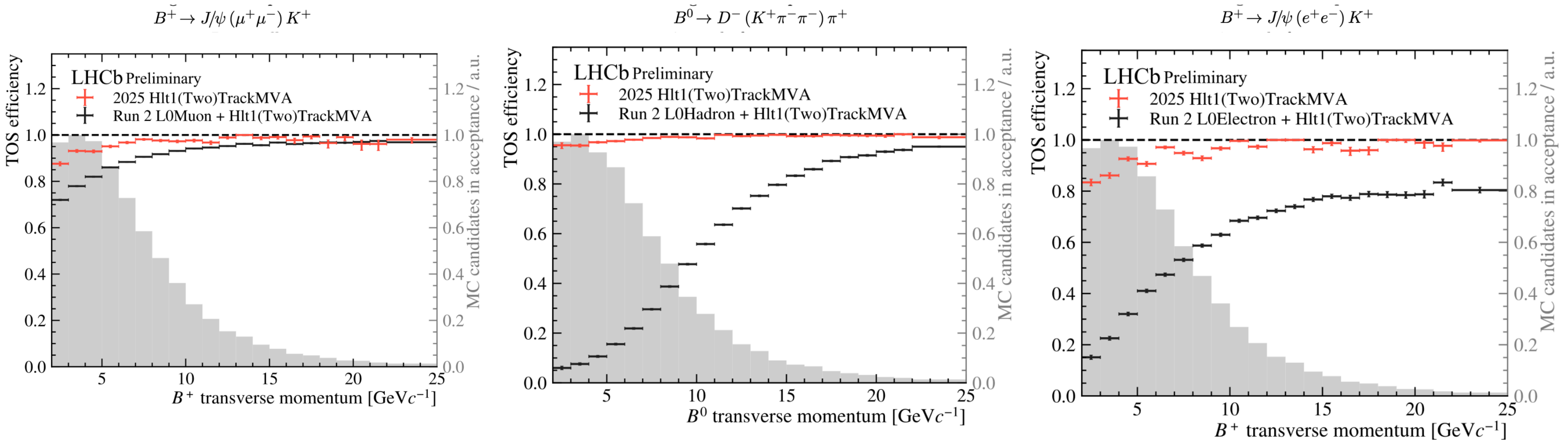
Impact of Kalman filter on J/Psi mass resolution



LHCb-FIGURE-2025-xxx

See also: « BuSca: New strategies for LLP Searches at 30 MHz at LHCb », J. Zhuo,

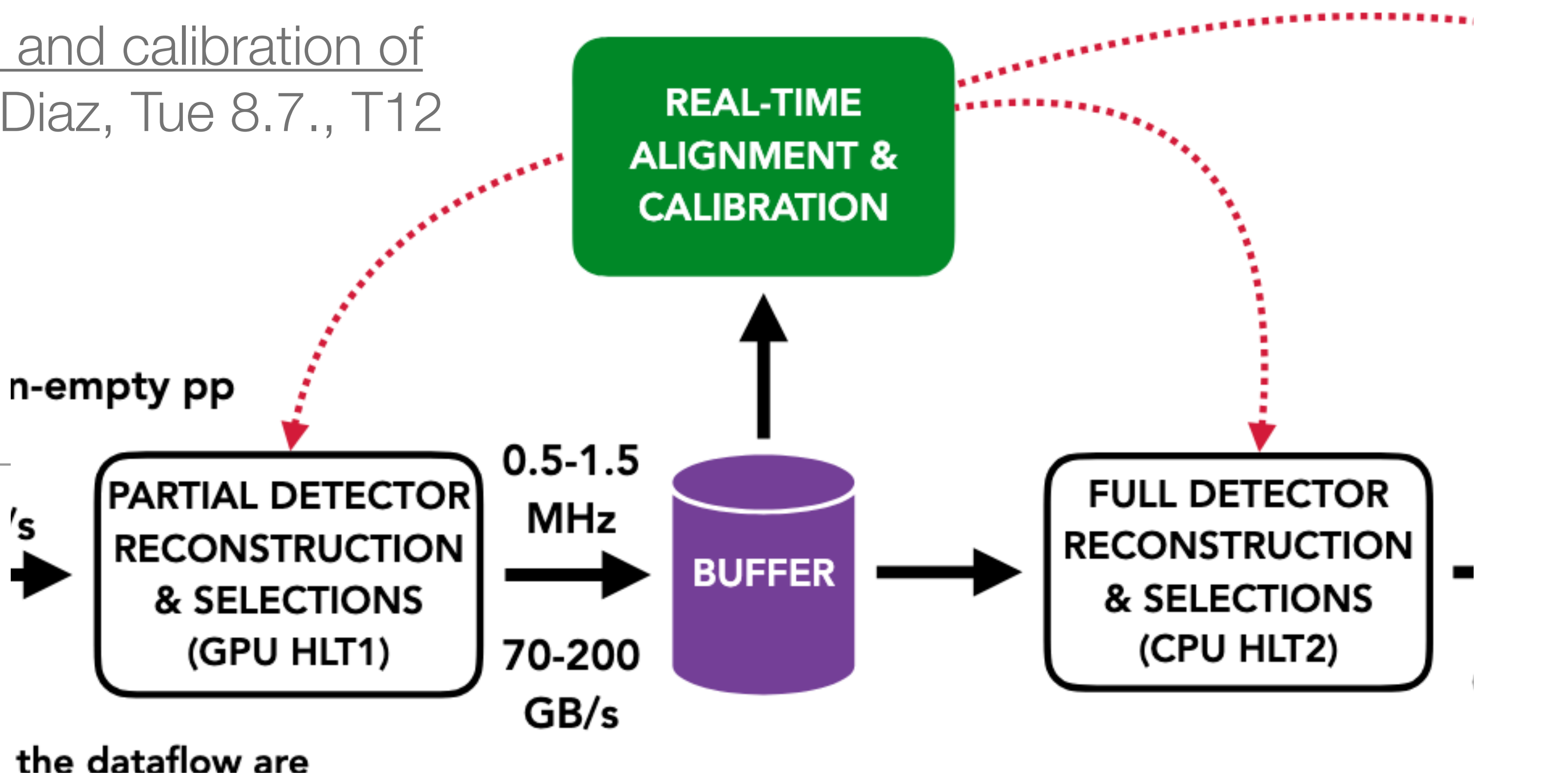
HLT1 performance on 2025 data



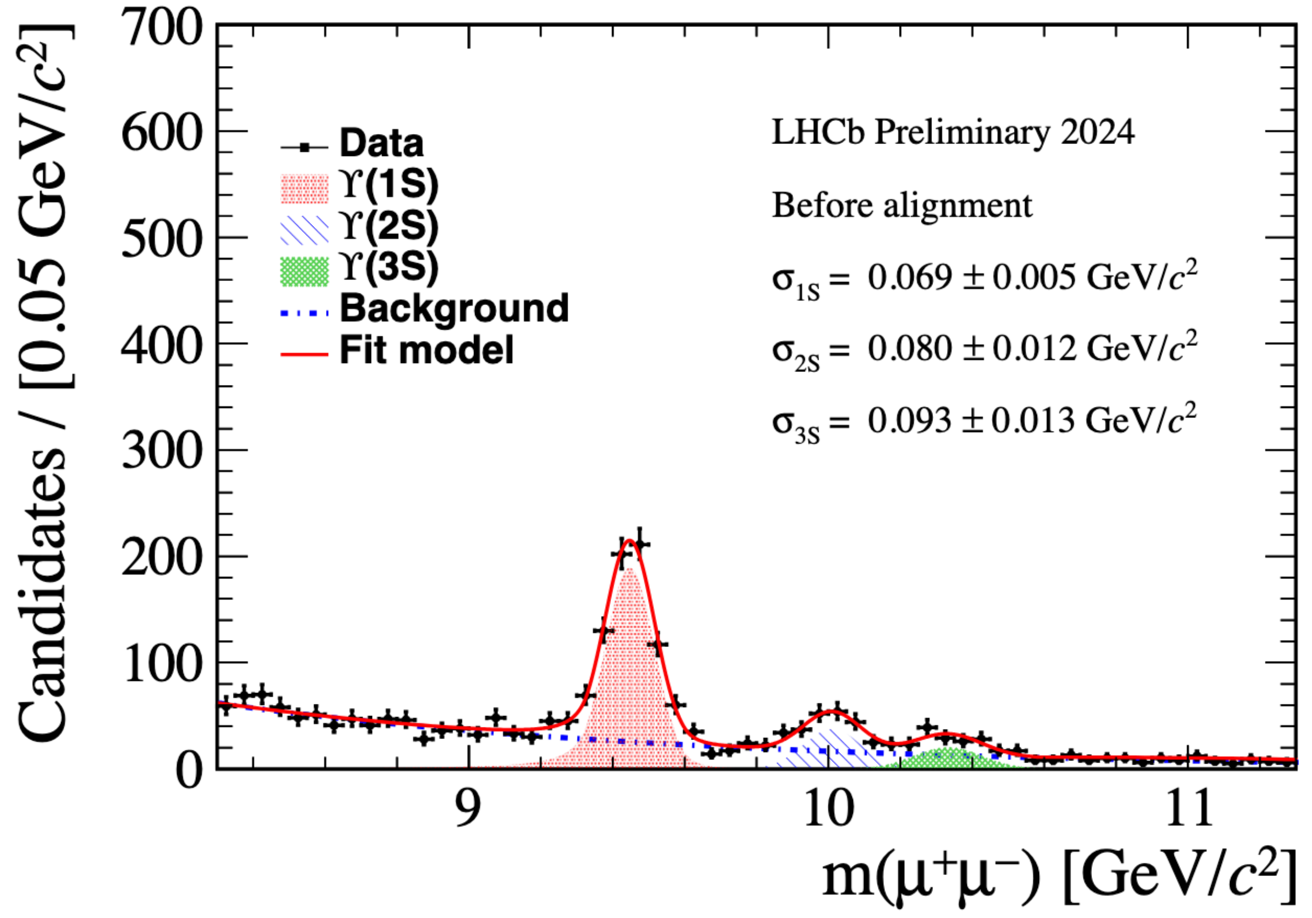
LHCb-FIGURE-2025-xxx

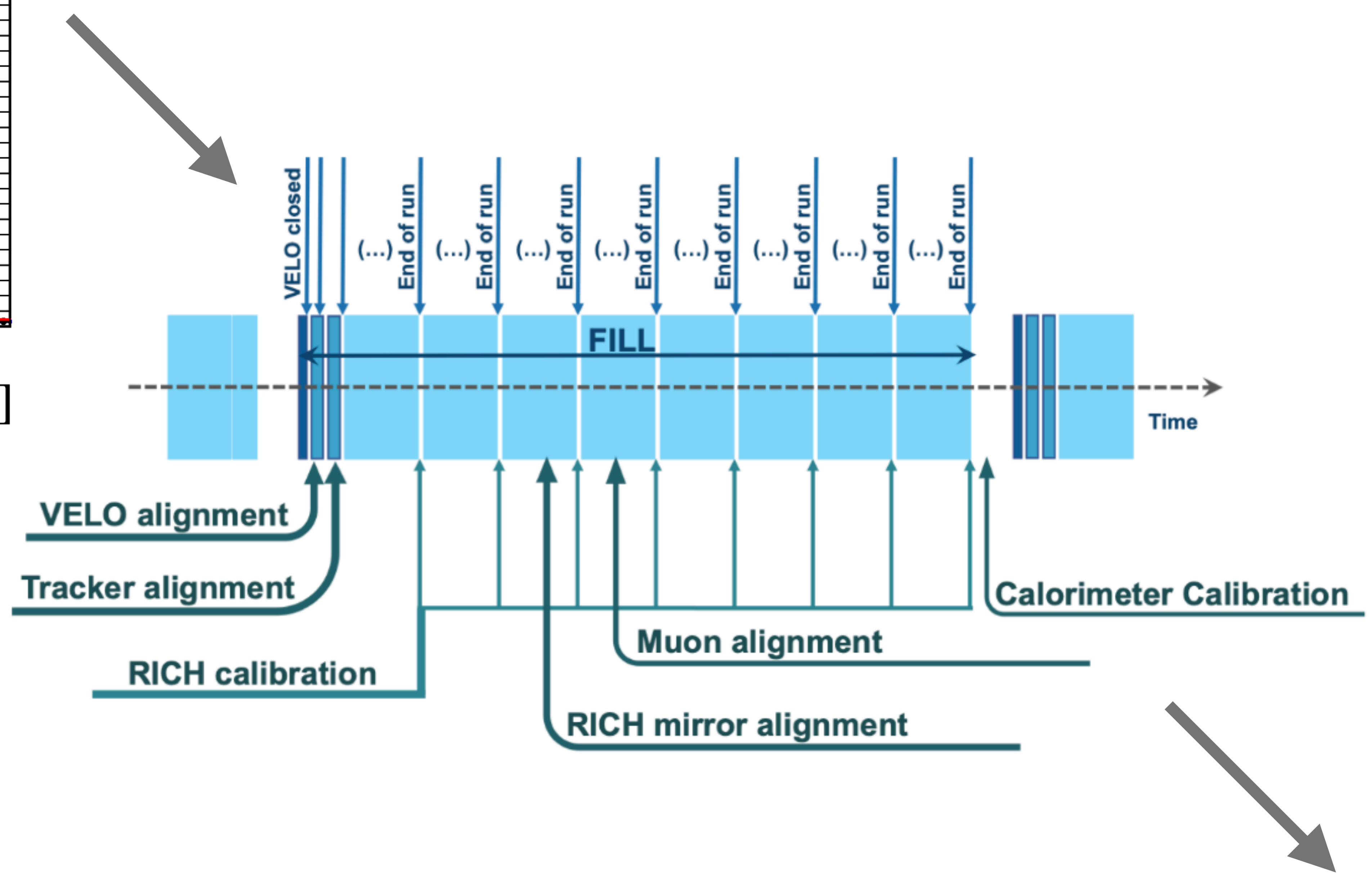
- Heterogeneous software trigger has been a full success
- HLT1 trigger performance greatly surpasses TDR goals set in 2014 and 2020

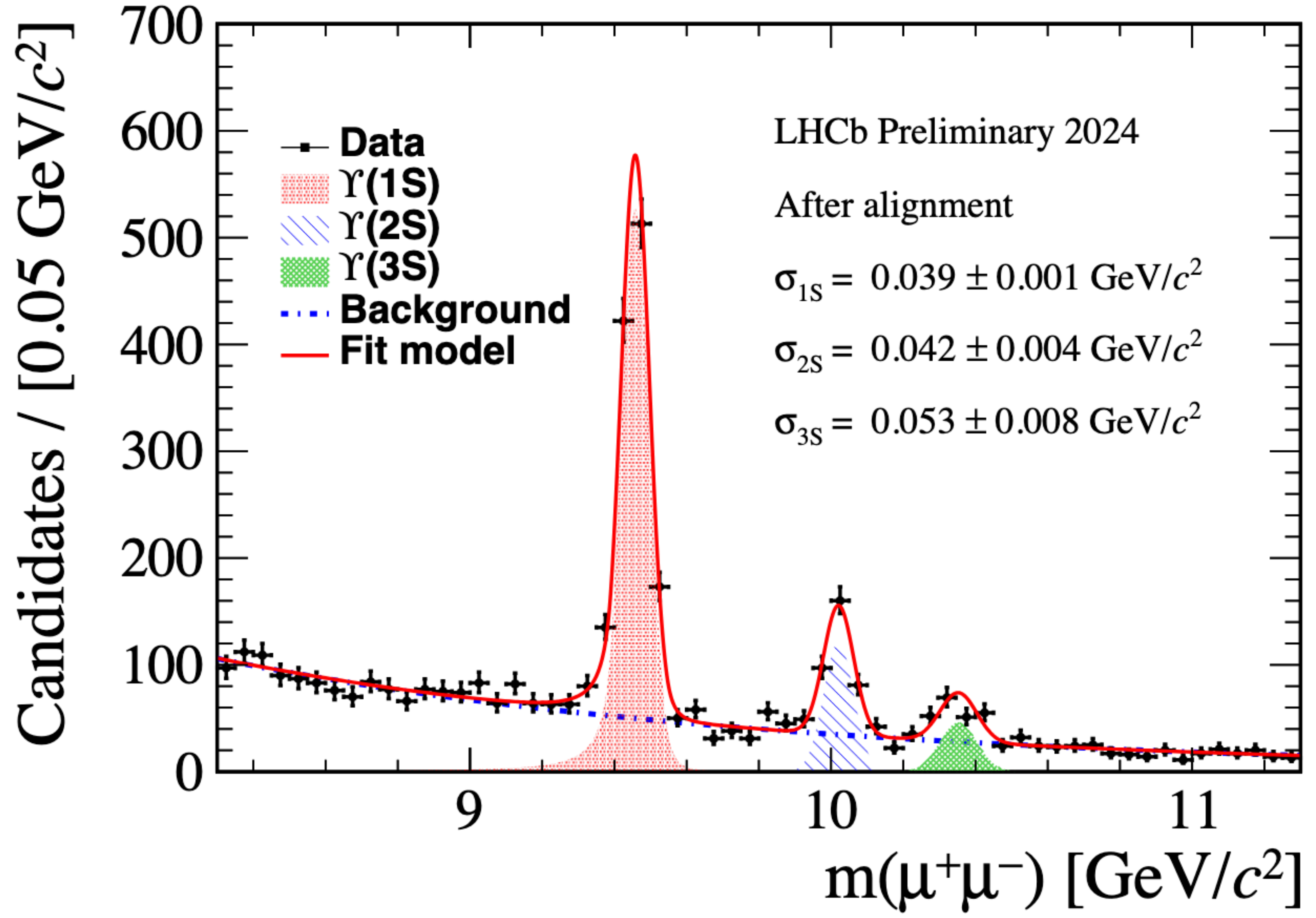
See also: Performance of the real-time alignment and calibration of the LHCb detector in Run 3 of the LHC, M. Ruiz Diaz, Tue 8.7., T12



Real-time alignment and calibration





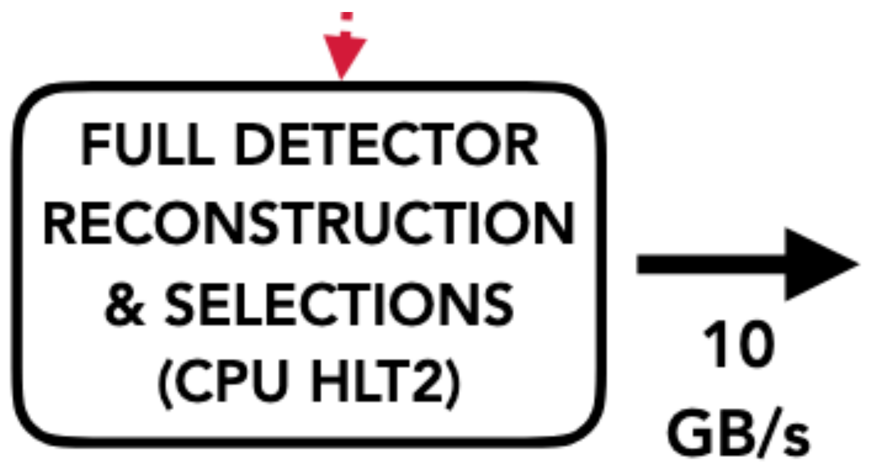


- Crucial for offline-quality HLT2 reconstruction
- Updated constants are also propagated to HLT1

LHCb-FIGURE-2024-037

HLT2 reconstruction

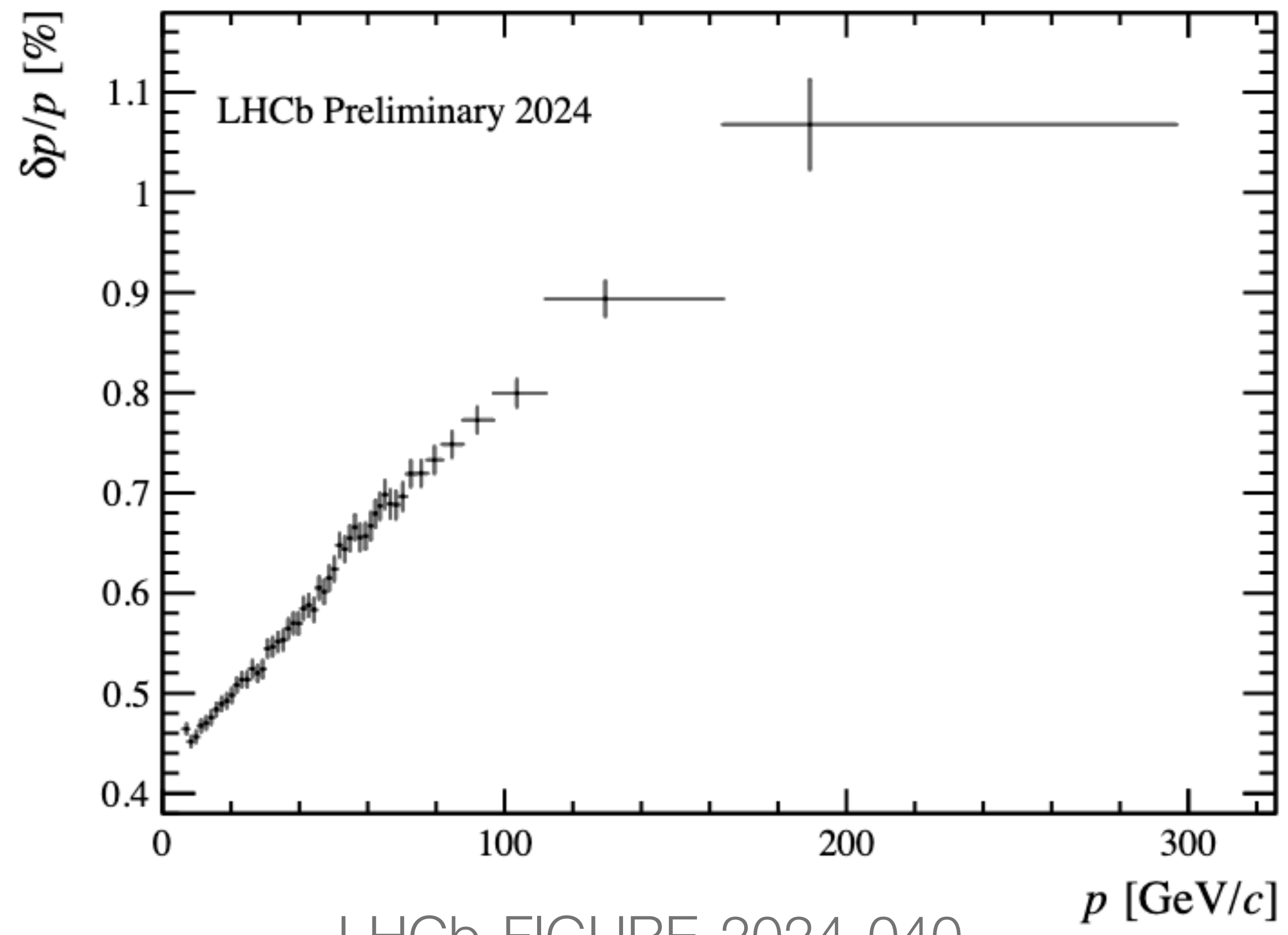
See also: Tracking and PID performance with the upgraded LHCb detector, G. Cavallero, Tue 8.7., T12



Offline quality reconstruction

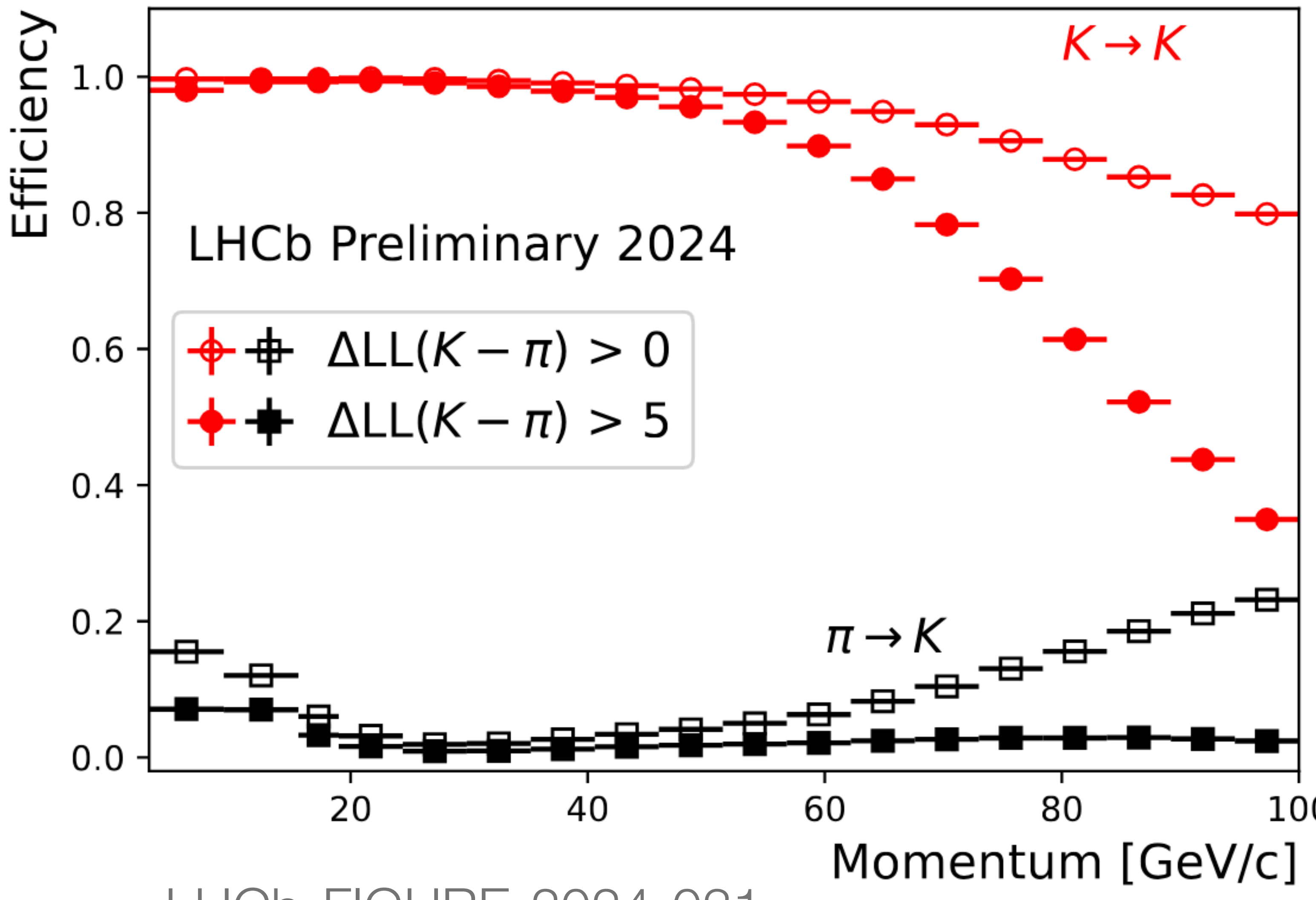
- Track reconstruction algorithms tuned for efficiency
- Kalman filter with parameterised scattering errors
- Hadron and global PID

Long track momentum resolution

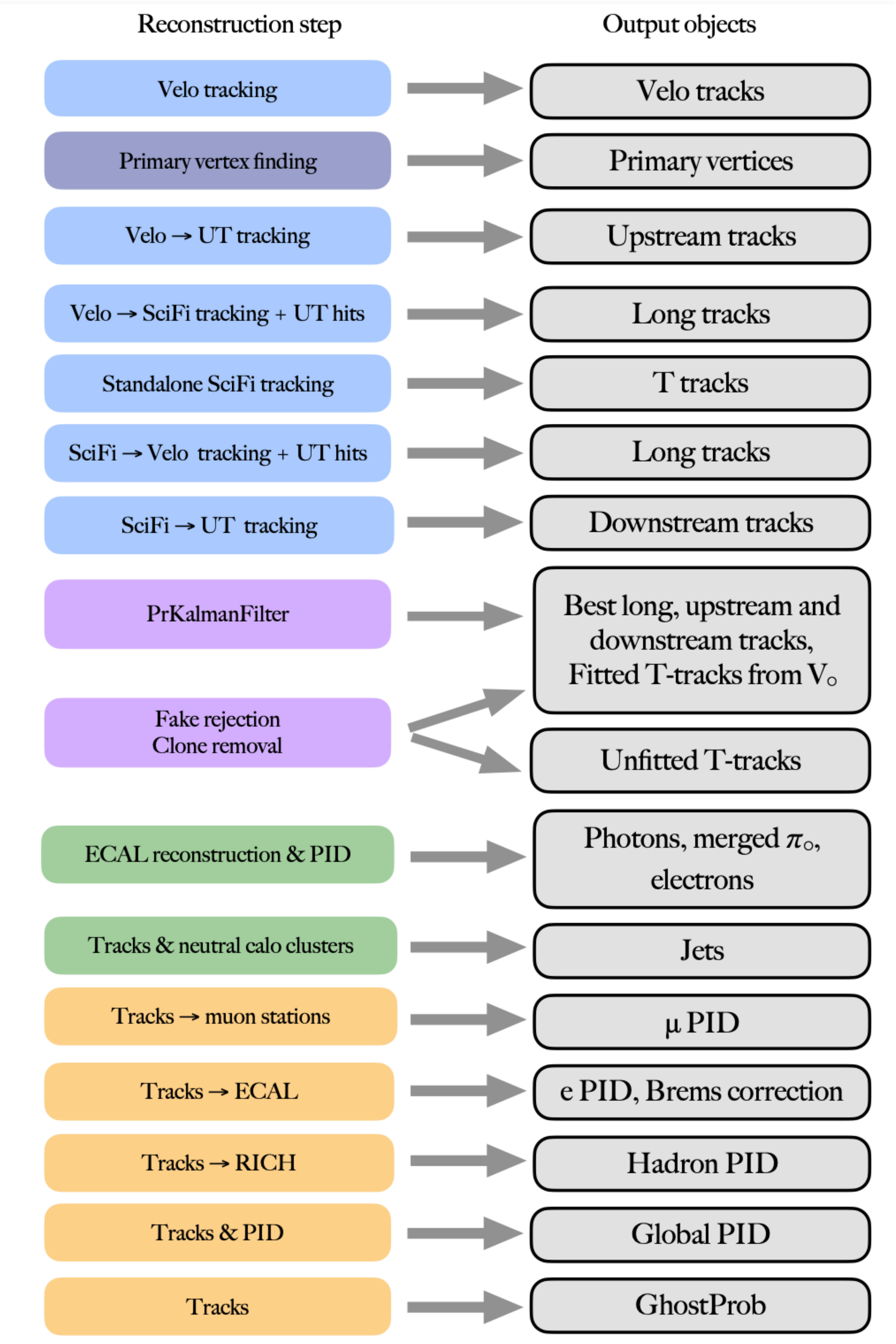


LHCb-FIGURE-2024-040

Hadron PID performance

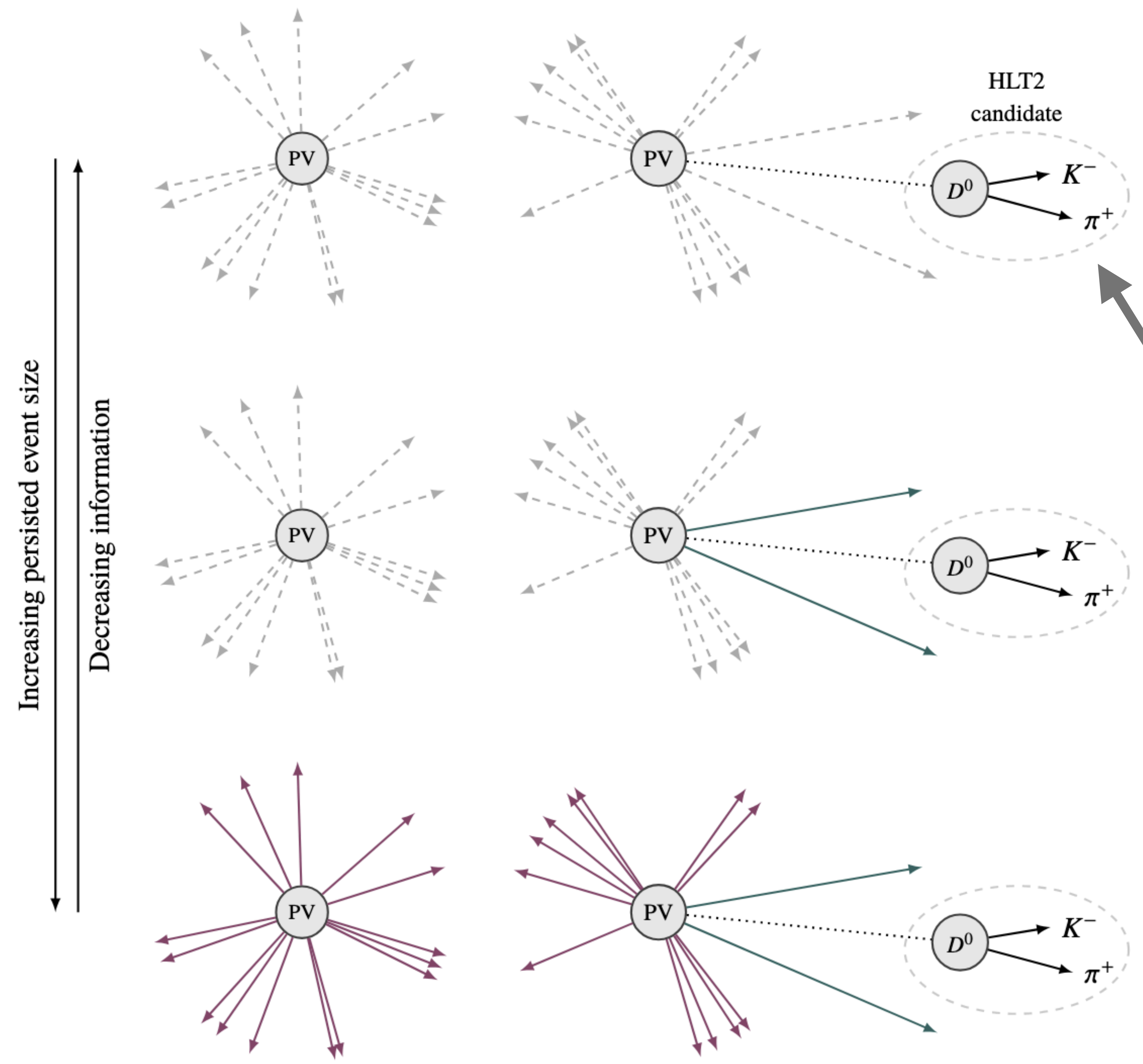


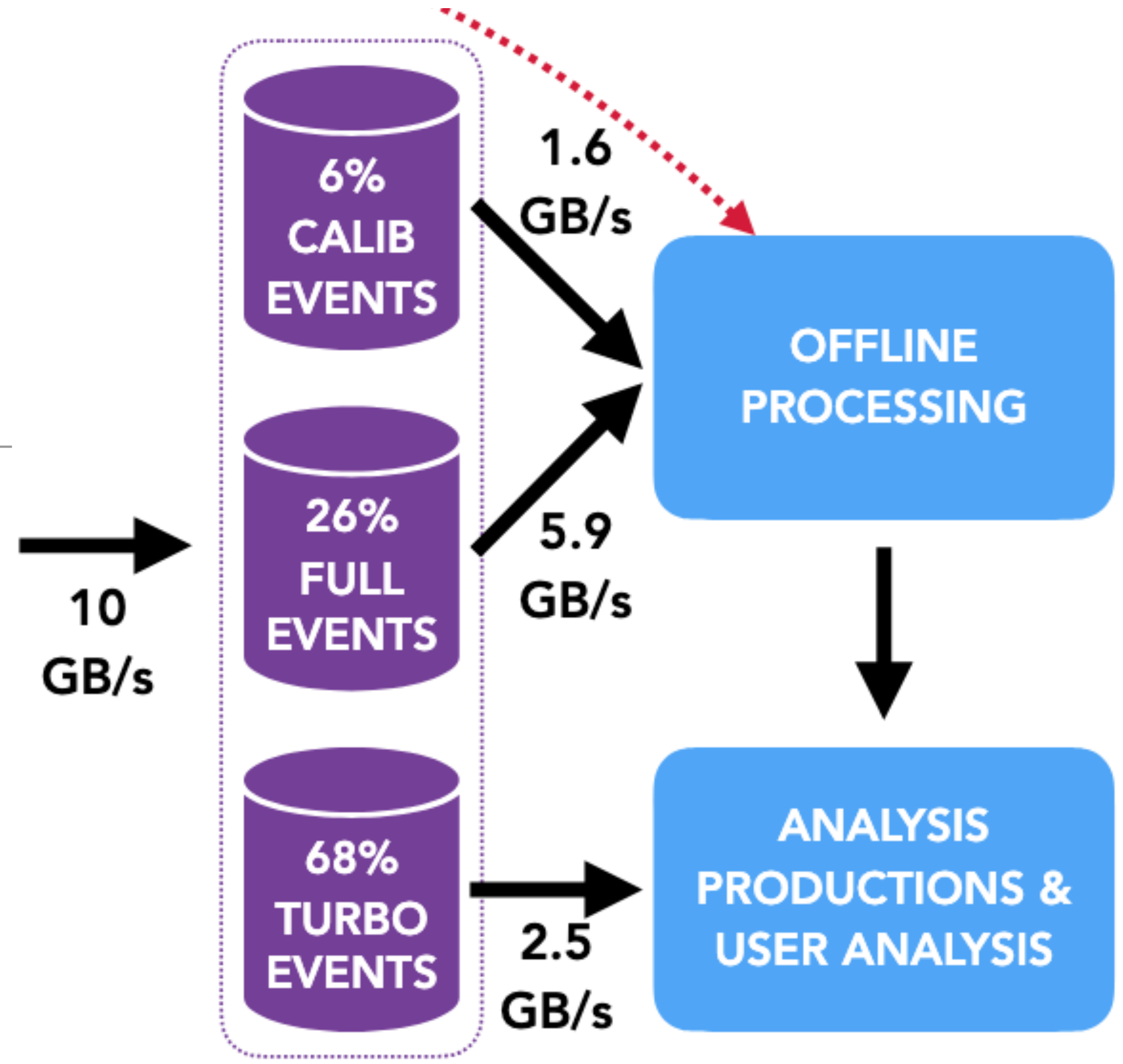
LHCb-FIGURE-2024-031



HLT2 persisted output

JINST 14 (2019) P04006



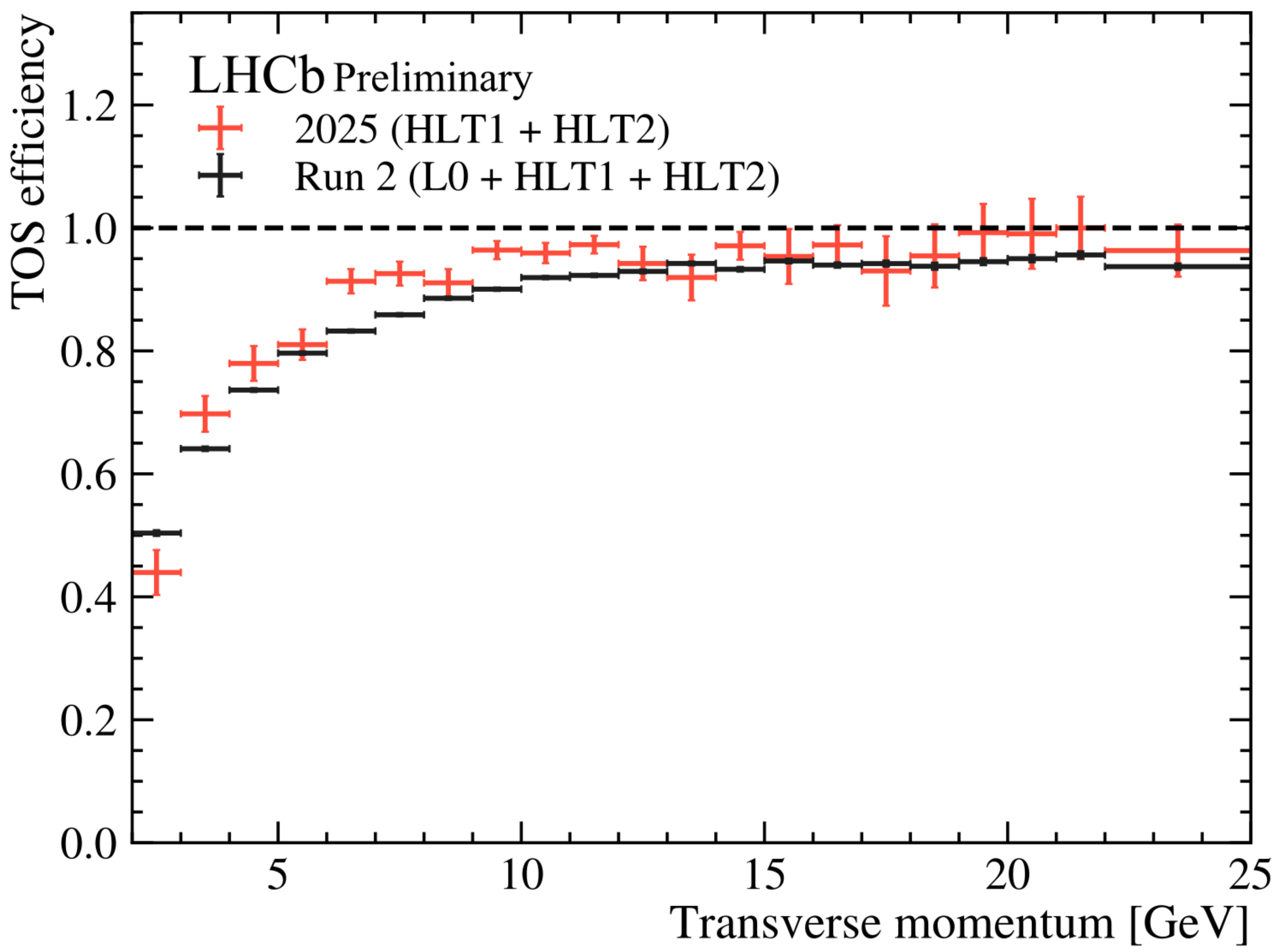


Three levels of persistency

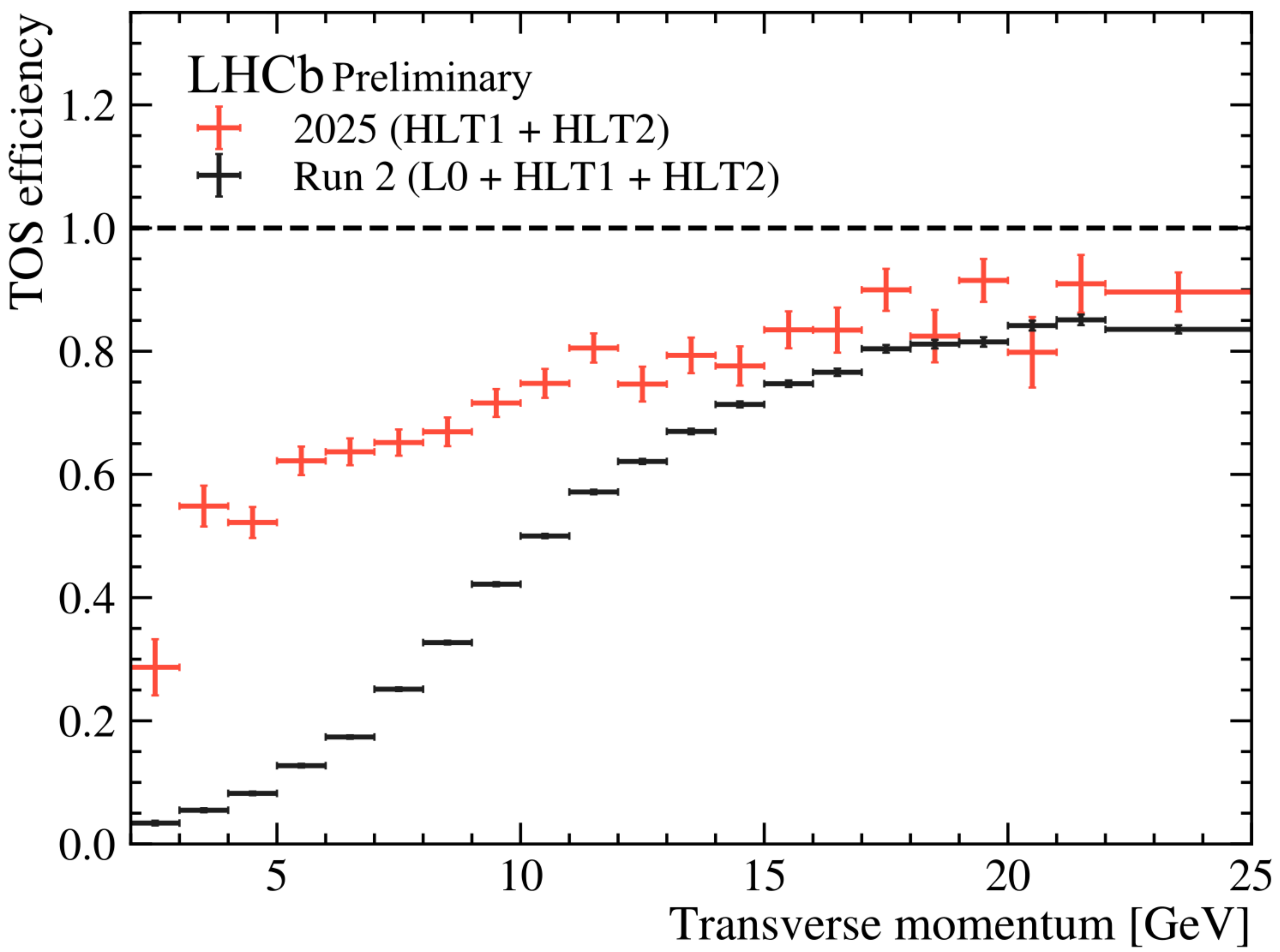
- Turbo: Save only information related to signal candidate reducing event size by a factor 10
- Selective persistency: Save additional objects
- Full persistency

Combined HLT1 & HLT2 performance on 2025 data

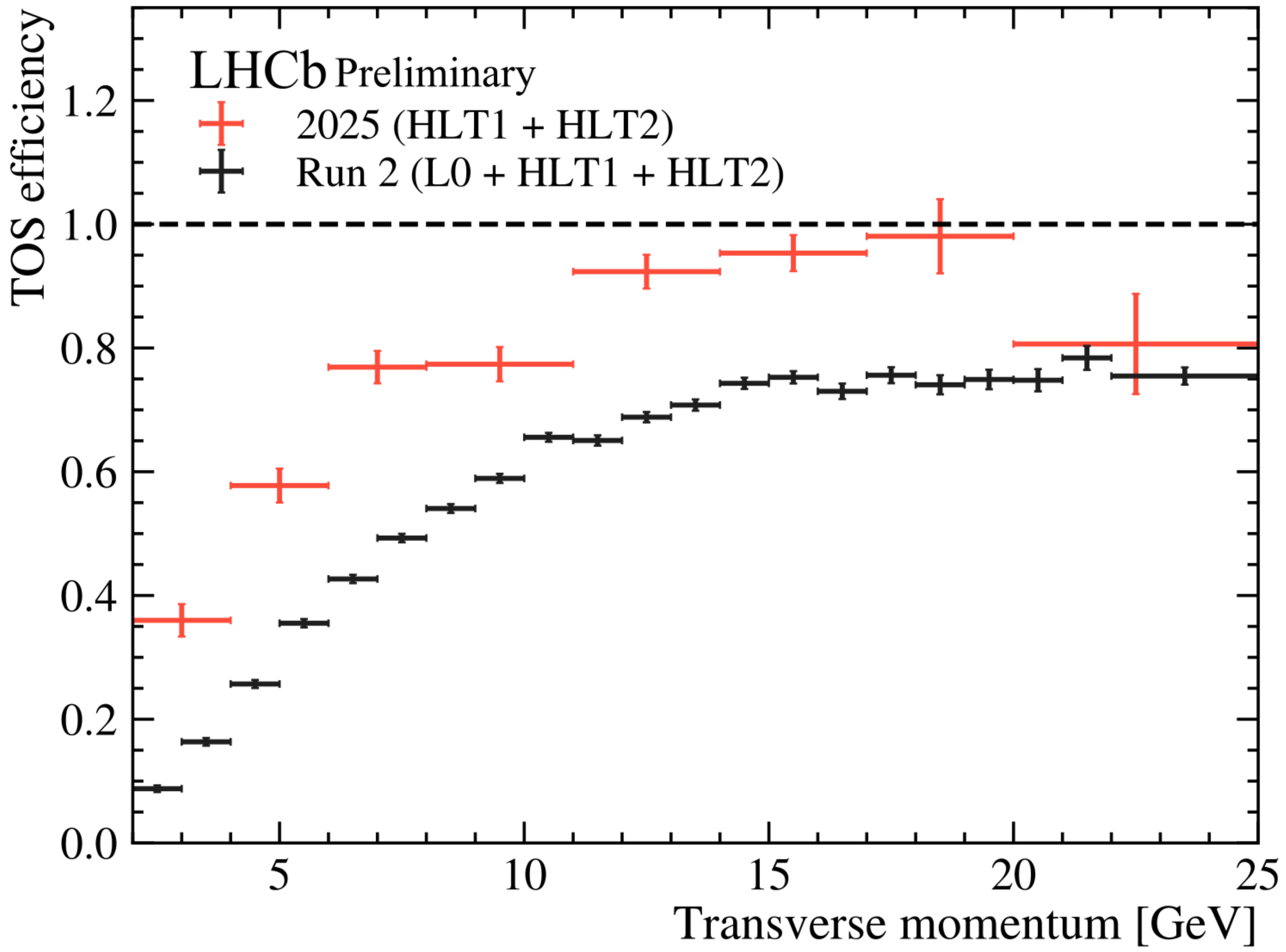
$B^+ \to J/\psi\,(\mu^+\mu^-)\,K^+$



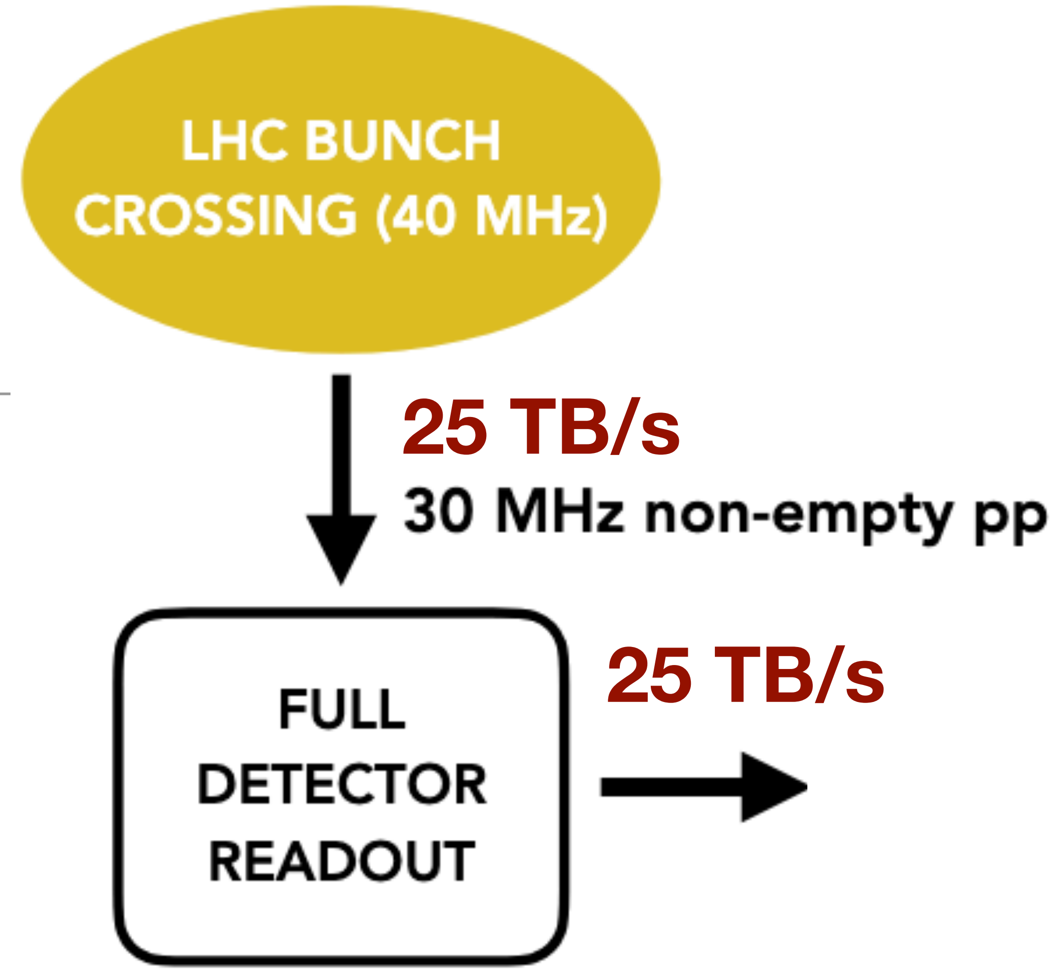
$B^0 \to D^-\,(K^+\pi^-\pi^-)\,\pi^+$



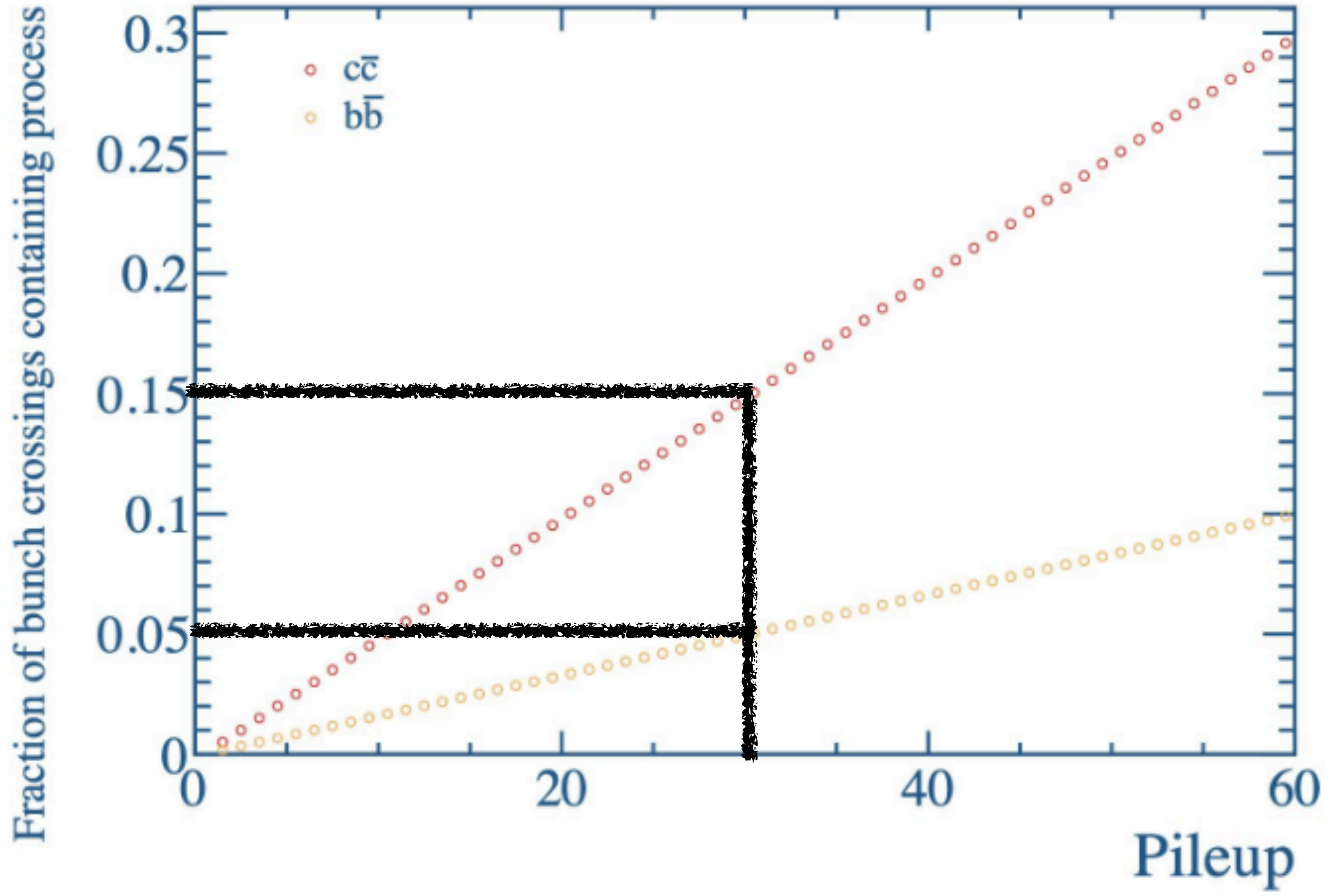
$B^+ \to J/\psi\,(e^+e^-)\,K^+$



The LHCb trigger challenge in Upgrade 2 (LHC Run 5)



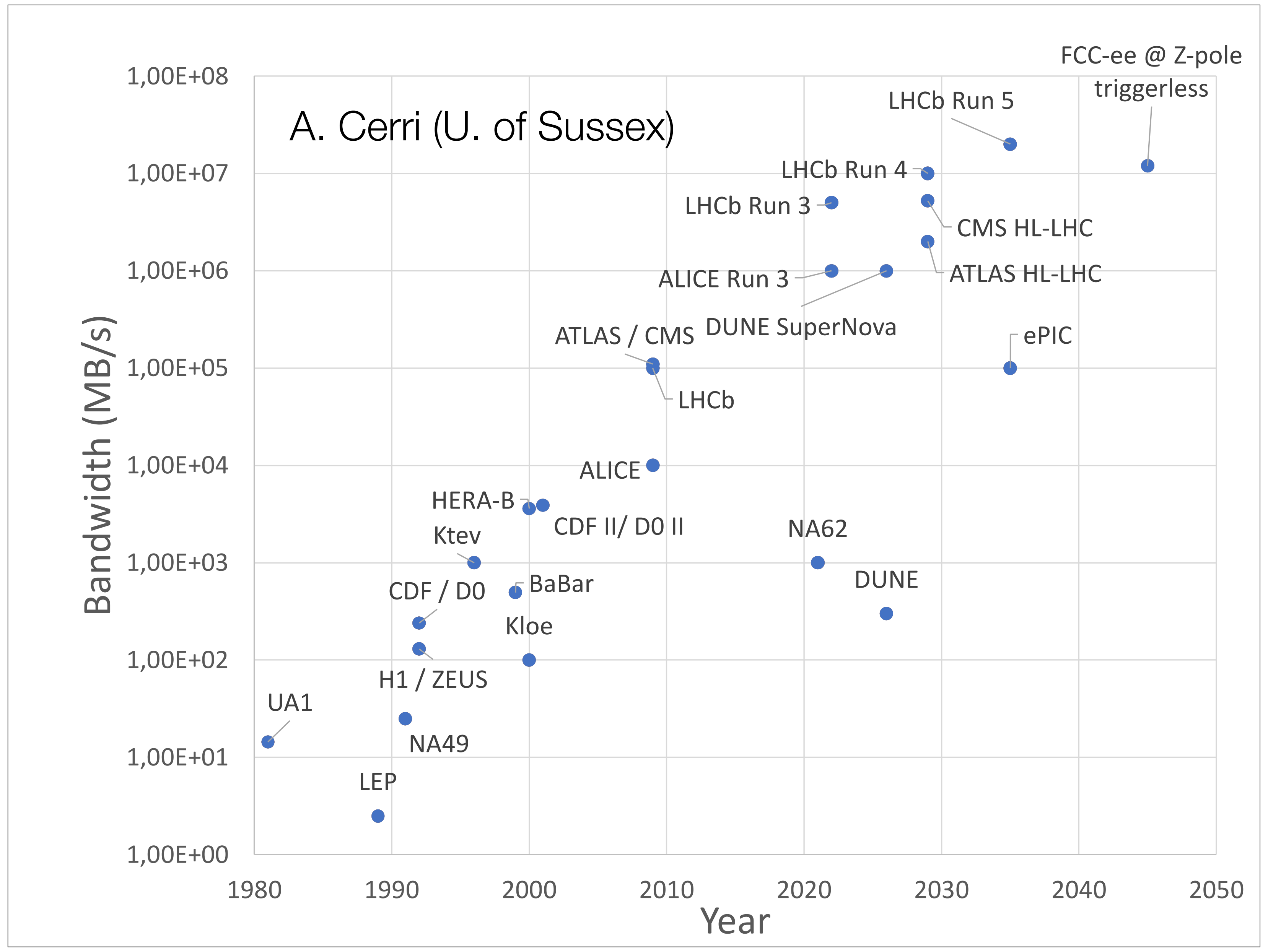
Run 1	Run 2	Runs 3/4	Run 5
x86 only	x86 only	x86 & GPU	x86 & GPU & ??
Offline reconstruction & calibration required	Quasi real-time analysis quality reconstruction & calibration	Quasi real-time analysis quality reconstruction & calibration	Real-time analysis quality reconstruction & calibration
4×10^{32} cm^{-2}s^{-1}		2×10^{33} cm^{-2}s^{-1}	1.5×10^{34} cm^{-2}s^{-1}
Heavy ions	Heavy ions	Heavy ions	Heavy ions
Proof-of-concept	Pilot program at limited centralities	Collision & fixed target program up to 30% centrality	Collision & fixed target program at all centrality values



pp analysis-level performance at same level, despite higher pileup

The path towards processing 25 TB/s

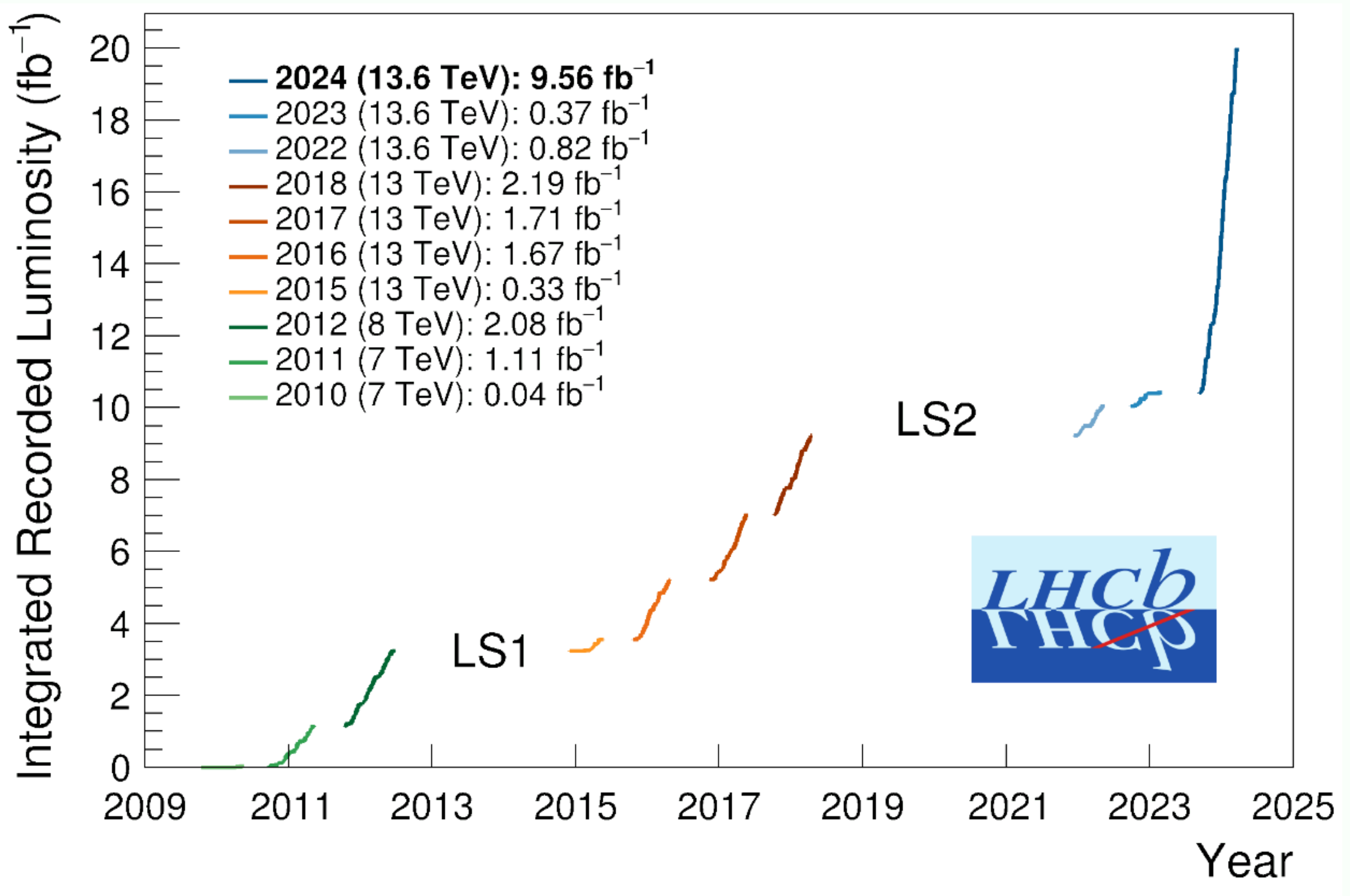
- Only currently viable option is to process the full reconstruction on GPUs
- Best suited architecture to process thousands of selections still to be determined
- Split (or no split) between HLT1 and HLT2 still to be determined
- R&D to be carried out over the next years
- Plan to adapt Allen and Gaudi software frameworks to cope with these challenges
- ML/AI techniques will become increasingly important for reconstruction, classification, selection
- HEP-specific challenge: Fast inference at 30 MHz on GPUs, FPGAs and emerging technologies



LHCb Upgrade 2 trigger can act as technology driver and catalyst for FCC-ee experiments

Summary

- LHCb has revolutionised its trigger system in Run 3 to use a fully software system based on GPUs and CPUs
- Allen framework has enabled HLT1 to go far beyond the original TDR design
  - Allen provides generic tools for high-performant processing on heterogeneous architectures
- HLT1, alignment & calibration, HLT2 achieved expected performance in 2024
  - Surpassed Run 2 performance in many cases
- Next challenge: LHCb Upgrade 2 with 25 TB/s to process
  - Will leverage our experience with heterogeneous software systems to process the full reconstruction on GPUs
  - Also keep flexibility to explore emerging technologies
- LHCb Upgrade 2 trigger can act as technology driver for FCC-ee experiments

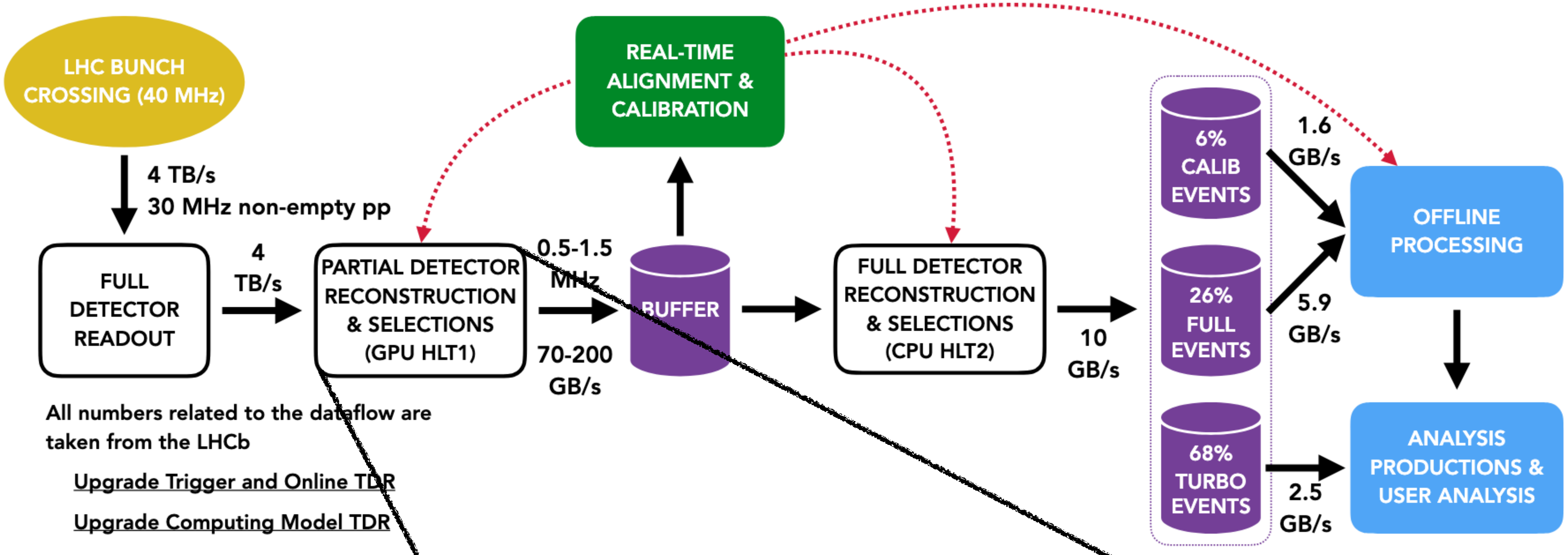


Backup

The Allen software framework

- Named after Frances E. Allen
- Hosted on giblab: https://gitlab.cern.ch/lhcb/Allen
- Documentation pages: https://allen-doc.docs.cern.ch/index.html
- Built with CMake
- Single source code, runs on CPU and GPU (Nvidia and AMD)
    - Portability between architectures provided by macros and few simple coding guide lines
- Standalone build and integrated with LHCb software stack
- Memory manager
- Multi-event scheduler
- Configuration via python
- Monitoring for development and data-taking
- Geometry loaded from DD4Hep, converted to simple structs for easy use on GPU

The LHCb trigger in LHC Runs 3 & 4

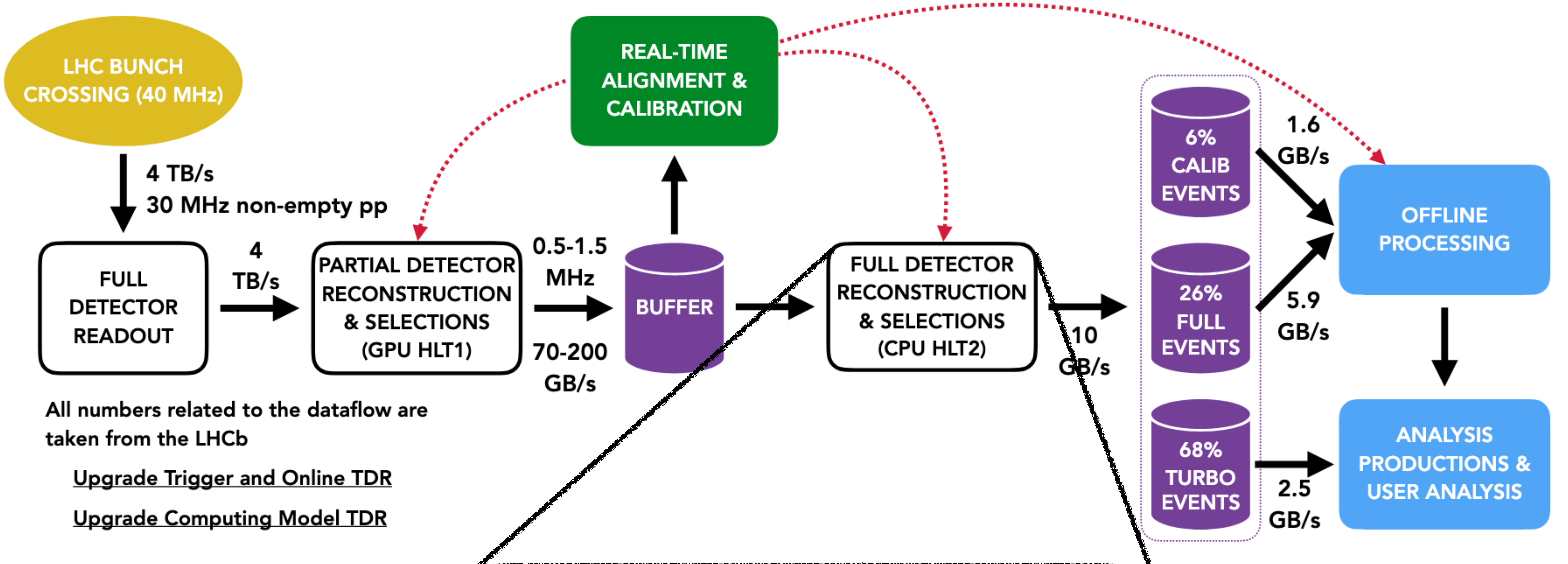


High Level Trigger 1 (HLT1)

- 30 MHz input
- Not latency bound
- O(100) algorithms to maintain
- O(10) developers
- 125k lines of code
- Processed by Allen software on O(500) GPUs

LHCb-FIGURE-2020-016

The LHCb trigger in LHC Runs 3 & 4

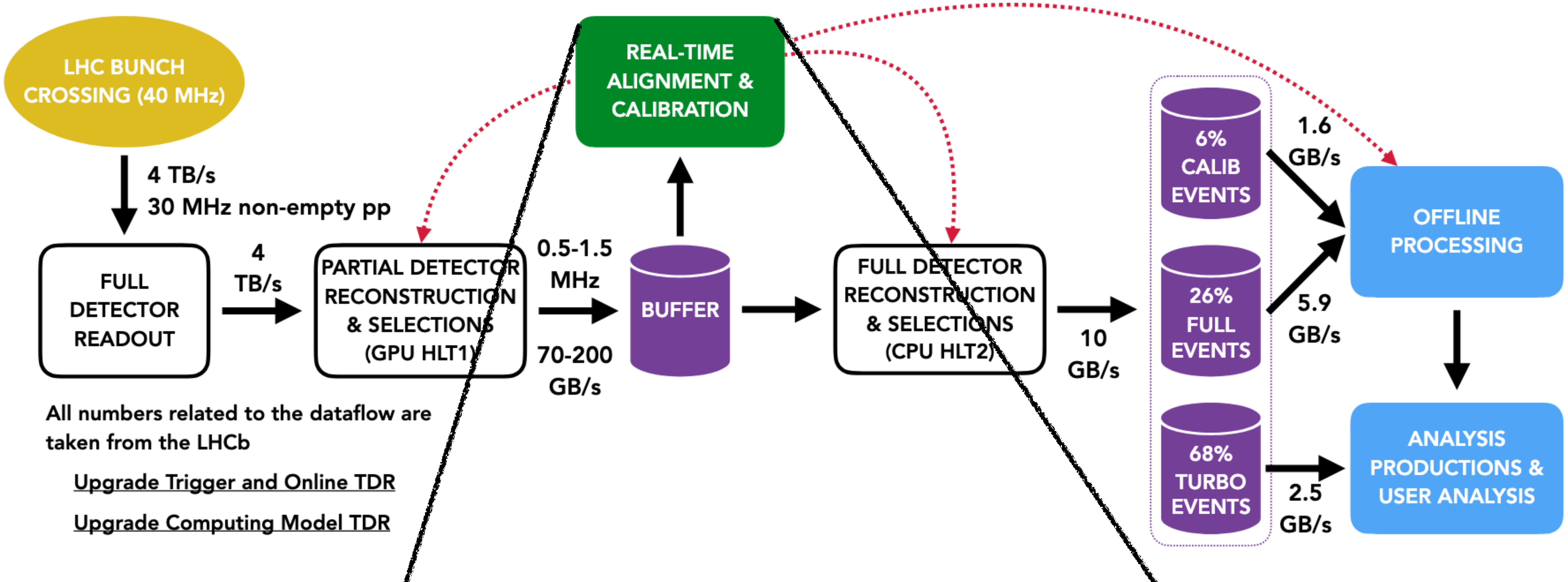


High Level Trigger 2 (HLT2)

- 1 MHz input
- Not latency bound
- O(2000) algorithms to maintain
- O(100) developers
- 2M lines of code
- Processed by Moore software on CPUs

LHCb-FIGURE-2020-016

The LHCb trigger in LHC Runs 3 & 4

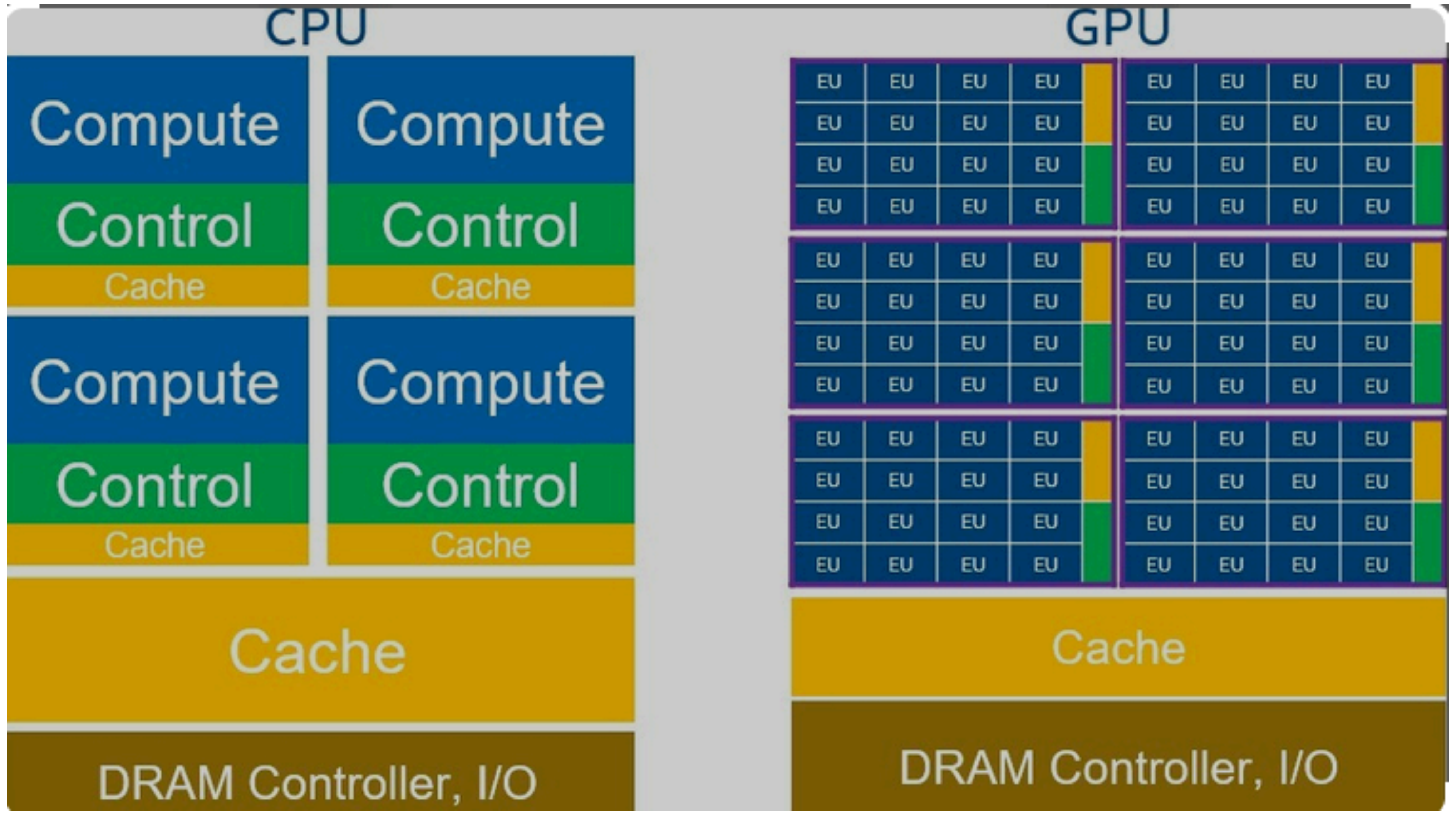


Alignment & Calibration

- Determine detector position based on reconstructed quantities
- RICH mirror alignment & refractive index calibration
- Calibrate calorimeter

LHCb-FIGURE-2020-016

Allen: Memory management and algorithm scheduling



CPU
- Few strong cores
- Suited for serial workloads
- Quick access to large system memory

GPU
- Many weaker cores
- Suited for parallel workloads
- Limited memory on card

GPU challenges

- Fit workload into limited memory resources
    - Allen uses count first - write later model
        - Only allocate as much memory as needed
    - Allen provides memory manager and algorithm scheduler
        - Only keep objects in memory for as long as needed
- Sufficient parallelisable work to utilise compute cores
    - In Allen, every algorithm processes many collision events in parallel
    - « Multi-event scheduler » in Allen generates static sequence of algorithms to be processed using event masks

Allen: Versatile and scalable user interface

Users

Developers

Core developers

Python interface

Versatile configuration of mixed algorithm sequence

Algorithm source code (CUDA)
- Classical reconstruction algorithms
- AI/ML algorithms
- Selection algorithms

Core framework components (C++)
- Memory management
- Scheduling