Faster, more efficient, more robust

Machine learning in LHCb's real-time processing

Anton Poluektov
on behalf of LHCb collaboration

Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France

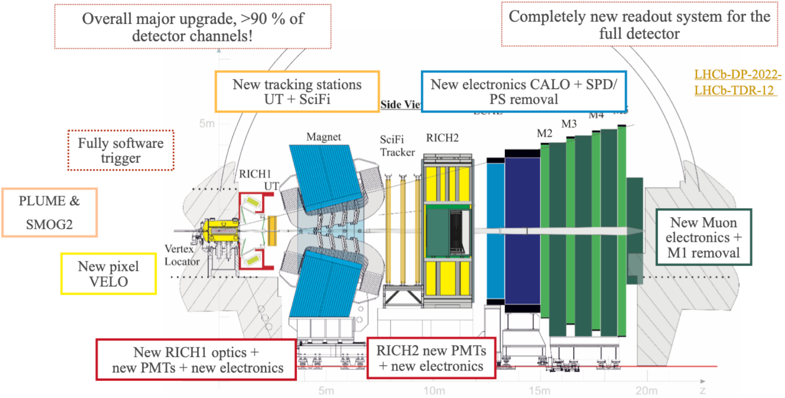
9 July 2025

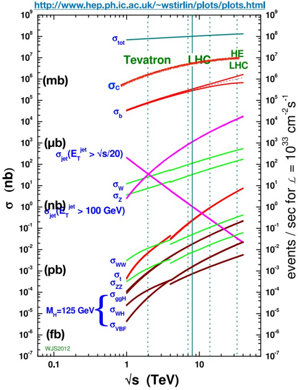
LHCb experiment

LHCb was originally designed to study B-hadron decays in pp environment at the LHC

Extended to study charm and even strange decays from the start of operation

LHCb in Run 3 (since 2022)

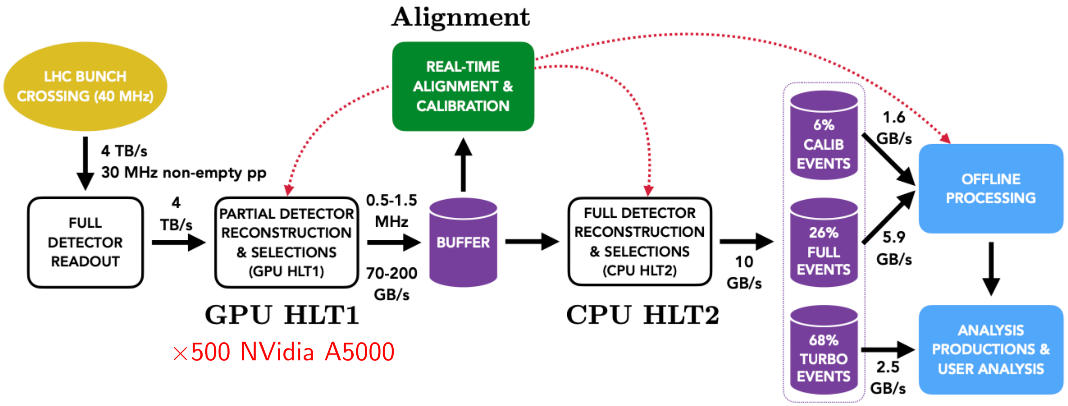




With current $\mathcal{L} \sim 2 \times 10^{33}$ cm^{-2}s^{-1}, signal rates $(b + c)$ are $\mathcal{O}$(MHz)

Output rate evolution: 200 Hz [Trigger TDR] → 2 kHz (Run 1) → 12 kHz (Run 2)

Radical change in Run 3 to maximally utilise MHz-level signal rate:



×500 NVidia A5000

∼ 10 GB/s output bandwidth (can store partial events in Turbo stream) [Talk by Dorothea vom Bruch]

ML in LHCb data processing

ML is used at all stages:

- Subdetector reconstruction
- Reconstruction of physics objects (tracks, neutrals, PVs)
- Exclusive and inclusive selections

real-time

- Flavour tagging and full event interpretation [Talk by John Wendel]
- Calibration, DQ monitoring
- Simulation
- Offline analysis

Requirements for real-time processing:

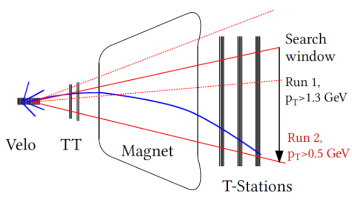
- **Fast**
  - High throughput (30 MHz HLT1, 1 MHz HLT2)
- **Efficient**
  - Applied early in processing chain
- **Robust**
  - Stable against changing running conditions, imperfect MC *etc.*

Due to high-throughput requirement, limited to "simple" architectures (fully-connected ANNs, BDTs) in real-time processing

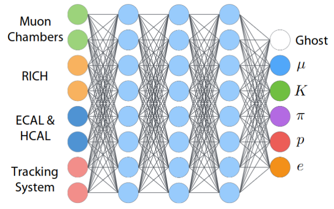
- R&D ongoing with more advanced approaches (GNNs, autoencoders):

Past and present

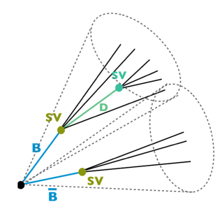
Tracking



Particle ID

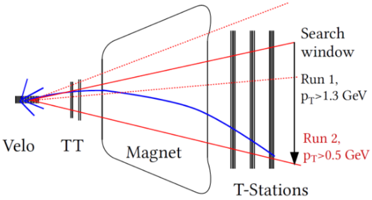


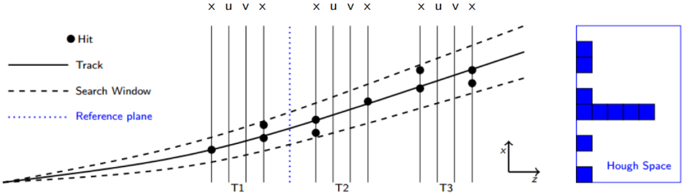
Inclusive B trigger



ANNs for forward tracking

[LHCb-PROC-2017-013]



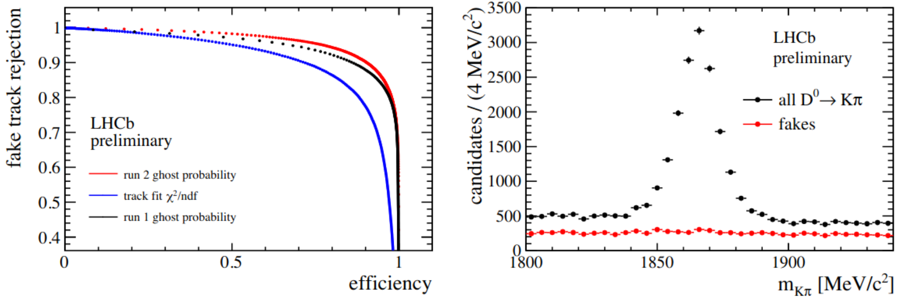


- Forward tracking:
  - Seed tracks: VELO or VELO+TT
  - Clustering in x plane
  - Adding stereo hits $\rightarrow$ Kalman fit
- **Two ANNs** in tracking to reduce combinatorics introduced in Run 2
  - Reject bad x clusters in T stations (9 input, 16+10 nodes HL)
  - Track candidate selection before Kalman fit (16 input, 17+9+5 nodes HL)

ANNs for fake track rejection

Rejection of fake tracks ("ghosts") based on TMVA [LHCb-PUB-2017-011]

- Inputs: 22 variables (χ^2 of track segments, numbers of hits, track kinematics, occupances, *etc.*)
- Trained with MC in different running conditions (pileup, bunch spacing)
- ANN implemented in TMVA, optimised efficiency/fake rate, CPU consumption



Offline (Run 1) → HLT2 (2015) → HLT1 (2016)

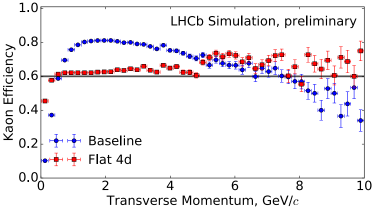
- **Reduces** CPU consumption in HLT2 by 58% (less combinations)

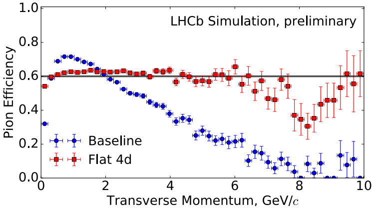
Global particle ID combining information of different subdetectors [LHCb-DP-2018-001]

- Combination of ~ 20 inputs from tracking, RICH, ECAL, HCAL and MUON detectors
- Output: probability estimate for each of the charged PID hypotheses (π, K, p, μ, e)
- Trained on full MC, MLP implemented in TMVA
- Alternative classifiers with specific features (*e.g.* boosted to uniform efficiency, [A. Rogozhnikov, *et al.*, 2015 JINST 10 T03002])

$$L = L_{\mathrm{exp}} + L_{\mathrm{fl}},$$

$$L_{\mathrm{fl}} = \sum_b \int |F_b(s) - F(s)|^2 ds$$



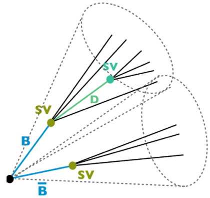


Topological trigger using Bonsai BDT

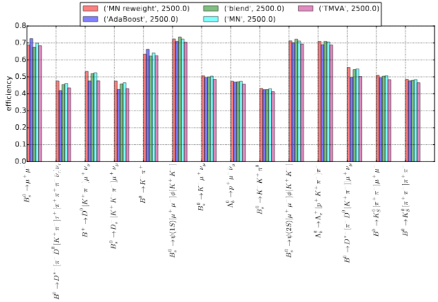
Most interesing signatures in LHCb are: high-p_{T} and displaced vertices/tracks

- Inclusive selections for most of B-hadron decays
- Topological trigger: displaced combinations of ≥ 2 tracks

[V. Gligorov, M. Williams, 2013 JINST 8 P02013]



"Bonsai BDT" with discretised inputs: fast (look-up table) and controlled overfitting



- Different classifiers for 2, 3, 4-body decay vertices
- Inputs: kinematics ($\sum p_{\mathrm{T}}$, $\min p_{\mathrm{T}}$), displacement (IPχ^2, FDχ^2), vertex quality, multiplicity, *etc.*
- Different BDT training algorithms compared (MatrixNet, AdaBoost variations)
- Optimised to different output rates (2.5 kHz, 4 kHz)

[T. Likhomanenko, *et al.*, J. Phys.: Conf. Ser. 664 082025]

Run 3: Topological trigger with monotonic Lipschitz NNs

Want our classifiers to be

- **Robust**: stable against detector instabilities and inaccuracy of simulation
- **Interpretable**: incorporate expected features, *e.g.* that interesing candidates have high p_{T} and high displacement

Both characteristics enforced by construction in **monotonic Lipschitz networks**

[O. Kitouni, N. Nolte, M. Williams, 2023 Mach. Learn.: Sci. Technol. 4 035020]

Lipschitz condition:

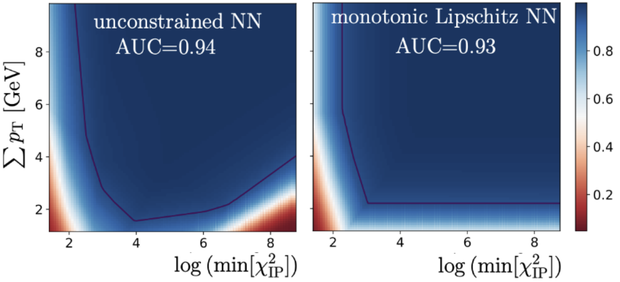
$$|g(x) - g(y)| < \lambda ||x - y||_1$$

by weight normalisation during training

Monotonicity:

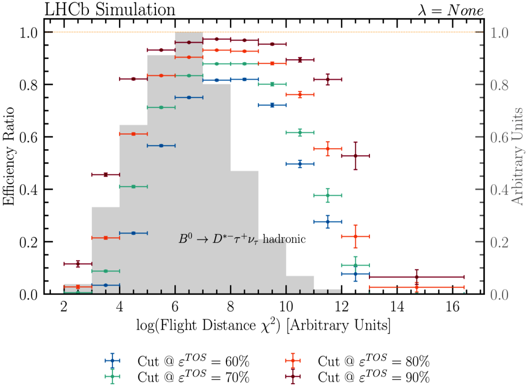
$$f(x) = g(x) + \lambda \sum_i x_i$$

"Tilt" the response with the same λ

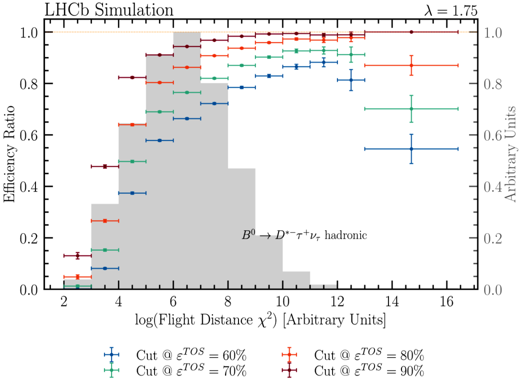


Monotonic Lipschitz NNs: Topological selections in HLT2

[N. Schulte, *et al.*, arXiv:2306.09873]



Unconstrained NN



Lipschitz monotonic NN

Applied to topo selections: ensure monotonicity as a function of p_{T} and flight distance significance

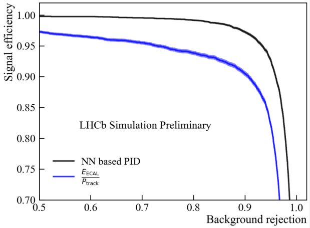
Lipschitz NNs: Lepton ID in HLT1

Same architecture can be applied to other use cases

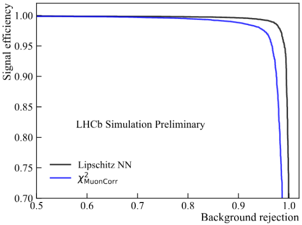
- Lepton (μ, e) identification in HLT1 [LHCB-FIGURE-2024-003] [LHCB-FIGURE-2024-029]

Significant improvement can be obtained wrt. “traditional” methods

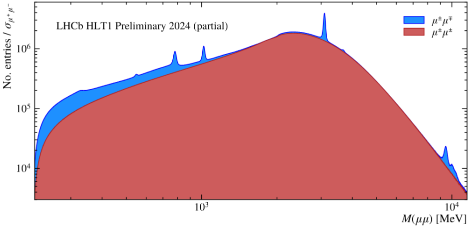
- E/p for electrons
- “Correlated χ^2” for muons [JINST 15 (2020) T12005]



Electron ID performance



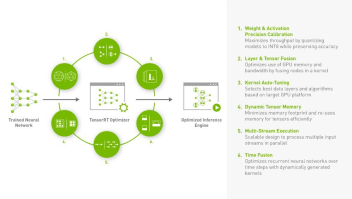
Muon ID performance



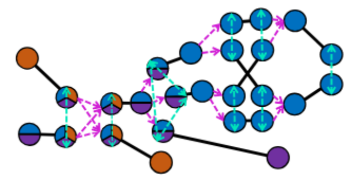
$m(\mu\mu)$ spectrum from HLT1 (data)

R&D

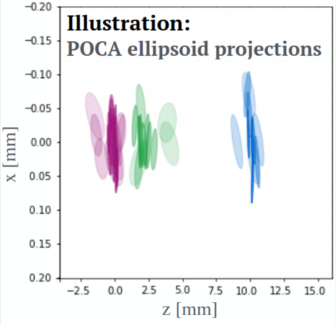
Inference frameworks



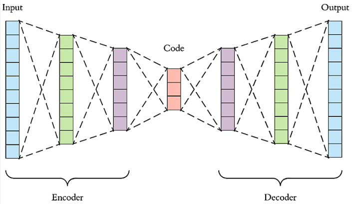
GNN in VELO tracking



GNN and hybrid ANNs for PV finding

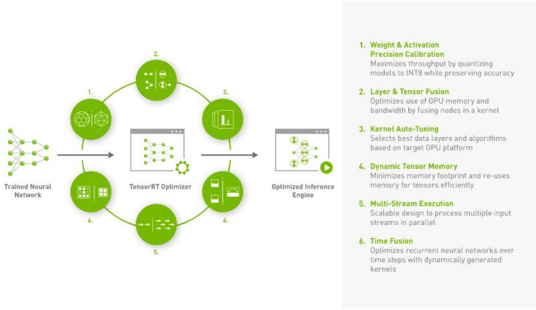


Anomaly detection in muon system

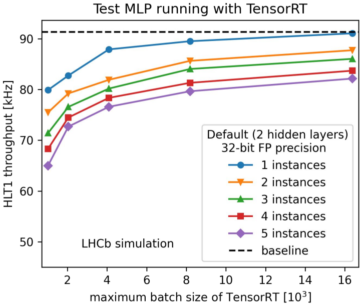


ML inference frameworks: ONNXRuntime and TensorRT

- Custom implementations of ANNs are not flexible and hard to maintain
- Considering dedicated ML inference frameworks:
  - CPU (HLT2): **ONNXRuntime**. Supported by most training software
  - NVidia GPU (HLT1): **TensorRT**
    - Can read ONNXRuntime files
    - Fast inference platform, SDK, optimisation



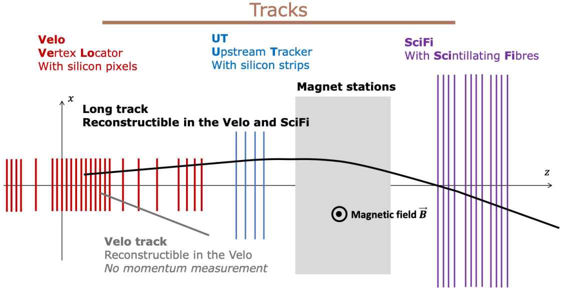
Effect on HLT1 throughput [LHCb-FIGURE-2023-006]



HLT1 throughput requirement: $\sim$ 60 kHz per GPU

- Main bottleneck is kernel overhead
- Several copies of typical MLP are feasible to run

GNN track finding in VELO



LHCb VELO:

- Pixel detector near *pp* interaction region
- 26 planes, 55×55 µm pixels
- No magnetic field: straight tracks
- $\sim$ 2000 hits/event, large combinatorics

Conventional algorithms: **quadratic** scaling with N_{hits}

GNN approach (Exa.TrkX): **near-linear** scaling [Eur. Phys. J. C 81 (2021) 876]

- Developed for Atlas, CMS (4π, magnetic field)

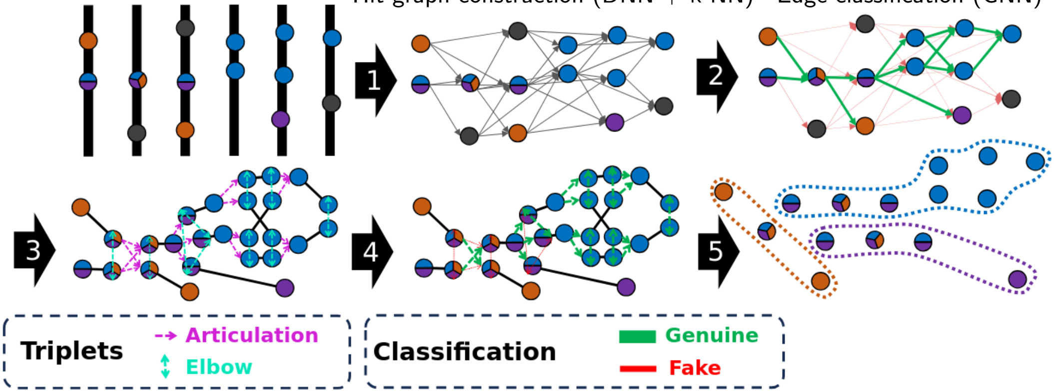
ETX4VELO: based on Exa.TrkX, but adapted to LHCb [A. Correa, *et al.*, PROC-CTD2023-34]

- No magnetic field, detection planes transverse to beam pipe
- Should handle: noise hits, inefficiency, shared hits, multiple hits per plane, material interactions (e^+e^- pair production)

GNN track finding in VELO

[A. Correa, *et al.*, PROC-CTD2023-34]

Hit graph construction (DNN + k-NN) Edge classification (GNN)

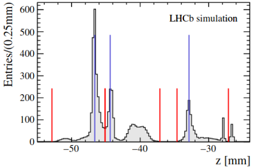


Edge graph construction, triplets Triplet classification, remove fakes Track construction (WCC)

New steps wrt. Exa.TrkX to handle shared hits

Lower ghost rate for same efficiency, improved electron reconstruction

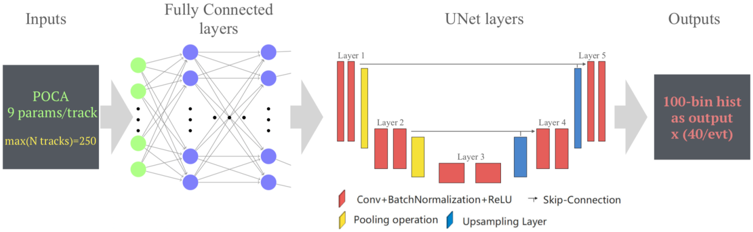
Hybrid **KDE-to-hist** approach [Rui Fang, *et al.* 2020 J. Phys.: Conf. Ser. 1525 012079]

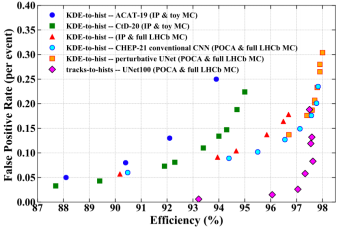


- KDE to produce 1D histogram of track z parameters
- CNN to find peaks and associate them with PVs

Hybrid **Track-to-hist** approach (collaboration betwen Atlas and LHCb)[S. Akar *et al.*, arXiv:2309.12417]

- Replace KDE with DNN acting on track parameters



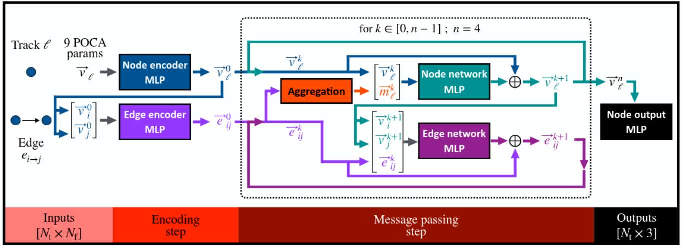


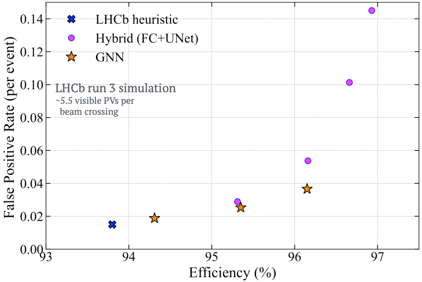
Similar efficiency and much lower flase positive rate compared to best KDE-to-hist model

ML in PV finding

Alternative approach: GNN for PV finding [S. Akar, poster at EuCAIFCon 2024]

- GNN using the same inputs as the VELO track finding model
- Output is true PV coordinates x_i, y_i, z_i



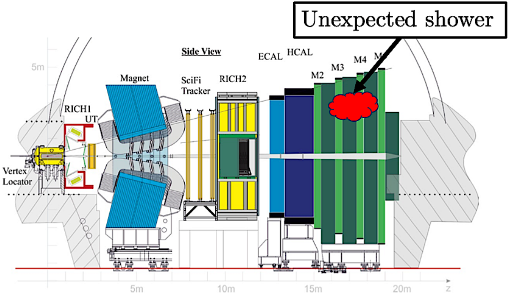


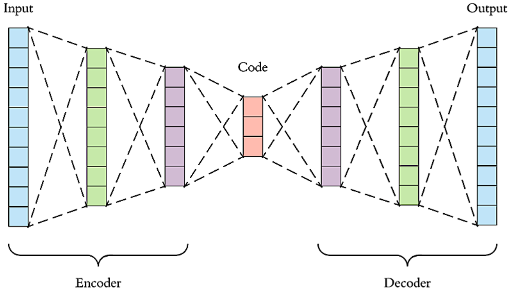
- Overall, slightly better physics performance wrt. hybrid model
- Track-PV association by construction

Autoencoders for anomaly detection in HLT1

Anomaly detection for showers in the muon detectors with HLT1 [LHCB-FIGURE-2024-015]

Search for signatures of **long-lived particles** (*e.g.* axions, ALPs, HNLs)



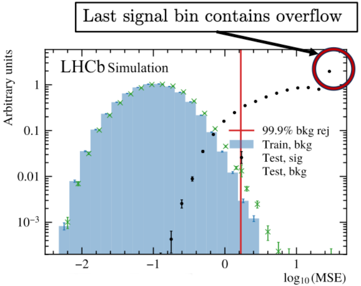


Autoencoders (AE) trained to generalise via a bottleneck layer in the architecture

- Minimise the difference between input and output
- Train on "normal" data (no anomalies, minimum bias (MB))

Autoencoders for anomaly detection in HLT1

[LHCB-FIGURE-2024-015]



Model	Parameters	Axion	$N \to eX$, 1.6 GeV	$N \to eX$, 4 GeV
BDT	< 3760	$(48.4 \pm 0.4)\%$	$(6.1 \pm 0.2)\%$	$(8.3 \pm 0.2)\%$
NN	1.4×10^6	$(51.4 \pm 0.3)\%$	$(5.1 \pm 0.2)\%$	$(7.9 \pm 0.2)\%$
Siamese	4.2×10^6	$(27.8 \pm 0.4)\%$	$(3.9 \pm 0.2)\%$	$(4.6 \pm 0.2)\%$
AE	4.3×10^6	$(38.9 \pm 0.2)\%$	$(3.3 \pm 0.2)\%$	$(5.3 \pm 0.2)\%$
VAE	1.7×10^6	$(20.8 \pm 0.2)\%$	$(0.4 \pm 0.1)\%$	$(0.6 \pm 0.1)\%$
GANVAE	2×10^5	$(20.1 \pm 0.2)\%$	$(0.3 \pm 0.1)\%$	$(0.5 \pm 0.1)\%$
NAE	2.5×10^6	$(80 \pm 0.5)\%$	$(10.3 \pm 0.3)\%$	$(15.7 \pm 0.3)\%$

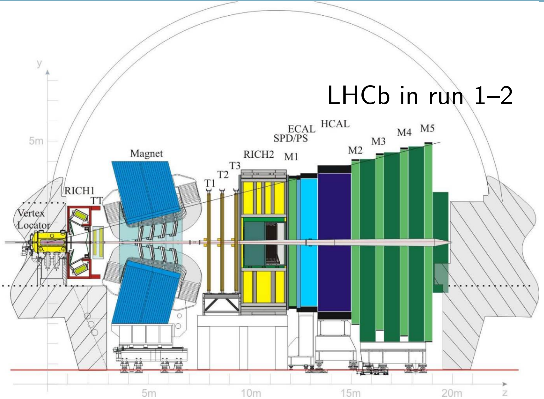
- Several variations of multivariate classifiers and, in partcular, AE are compared
  - Axions: $H \to AA$, $A \to \tau\tau$, $\tau \to 3\pi\nu$
  - Heavy neutral leptons: $N \to eX$
- The best performance offered by Normalised Autoencoder (NAE)[S. Yoon, *et al.*, arXiv:2105.05735]
  - Sampling reconstructible space outside MB domain and penalise AE giving small error on it

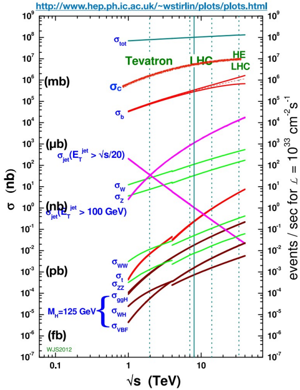
Summary

- ML permits maximally efficient utilisation of LHCb data and contributes to excellent physics performance
- ML is used practically at all stages of data processing, from subdetector reconstruction to offline analysis
- In real-time processing, due to high data rate requirement, limited to simple and robust architectures (fully-connected ANN, BDT)
  - Since the very start of LHCb (Run1): PID classifiers, ghost track rejection, topological trigger
  - In Run 3: PID, topo trigger with Lipschitz ANNs
- Work ongoing to integrate ML inference frameworks (ONNXRuntime, TensorRT)
- R&D in many areas:
  - Track finding
  - PV finding
  - Anomaly detection

Backup

LHCb experiment in 2010-2018

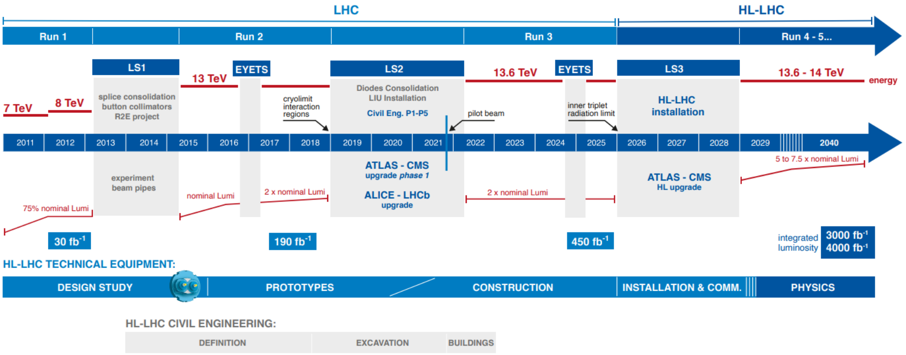




Forward spectrometer, optimised for b and c decays. $2 < \eta < 5$

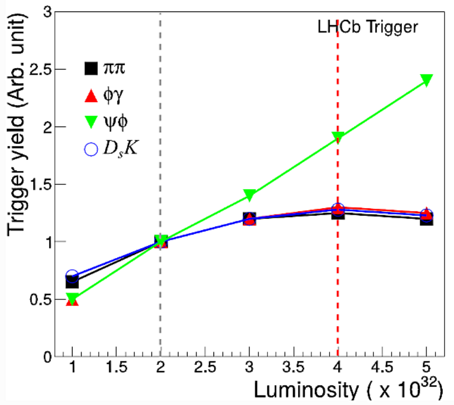
- Excellent vertex resolution (weak decays)
- High-precision tracking before and after the magnet
- PID in broad range of momenta $3 < p < 150\,\mathrm{GeV}$
- Efficient trigger, including fully-hadronic final states, $\sim$12 kHz output rate

LHC timeline



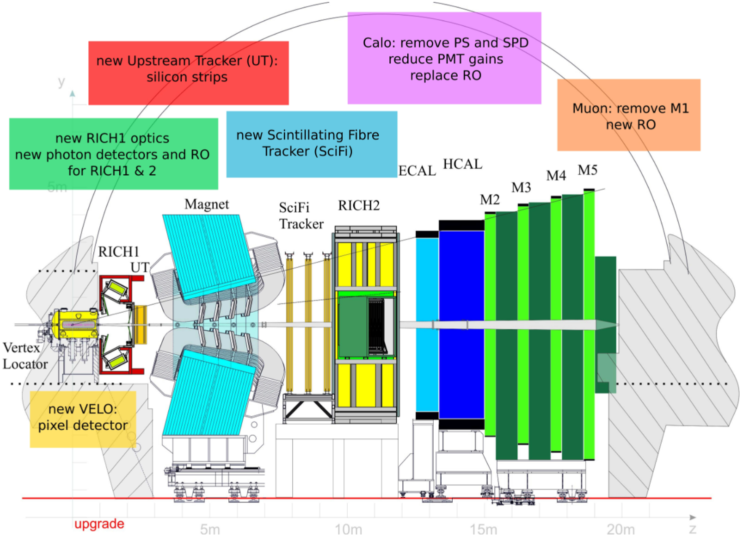
- LHC Run 2 finished in 2018
  - LHCb: $\int \mathcal{L}dt = 9\,\mathrm{fb}^{-1}$ collected in 2010-2018
- Long shutdown until 2022: upgrade of the machine and detectors
  - LHCb Upgrade I: major upgrade/replacement of the subsystems and readout
- Run 3 until 2026 $\rightarrow$ HL-LHC upgrade $\rightarrow$ Run 4 ...
  - LHCb goal: $50\,\mathrm{fb}^{-1}$ by the end of Run 4 $\rightarrow$ Upgrade II

LHCb upgrade case

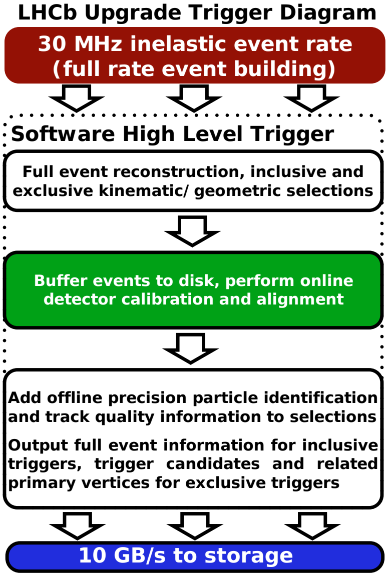


- Instantaneous luminosity: 4×10^{32} (Run 2) $\to 2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$
- Run 1–2 trigger:
  - First stage: hadrware L0 (40→1 MHz) using high p_T/E_T signatures
  - 1 MHz limit saturates hadronic modes already in Run 2 (higher rate ⇒ higher thresholds)
- The only solution: read full event at bunch-crossing rate and apply
  track reconstruction/IP selections.
- Upgrade/replace subsystems:
  - Cope with higher occupancy.
  - Faster/higher precision tracking
- Fully replace DAQ and trigger.

LHCb upgrade



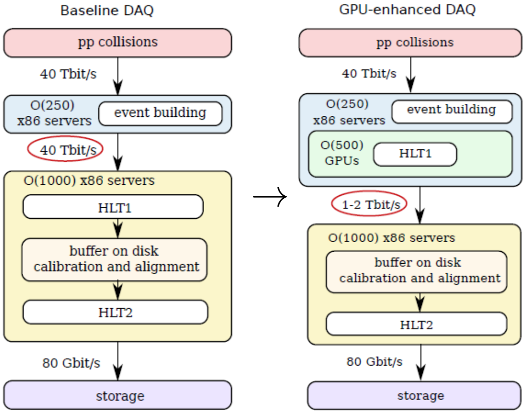
Complete replacement of DAQ, fully software trigger (HLT1 + HLT2)



HLT1: [LHCb upgrade computing TDR]

- Subdetector reconstruction:
  - VELO: clustering, tracking, vertex reconstruction
  - UT, SciFi: tracking
  - Muon: Hit-track matching
- Global event reconstruction:
  - Track fit (Kalman filter)
  - Reconstruction of secondary vertices
- Selections: [LHCb-PUB-2019-013]
  - Single displaced tracks
  - Two-track displaced vertices
  - Single displaced muons
  - Low-mass displaced two-muon vertices
  - High-mass dimuons

Baseline CPU-based design was replaced by GPU-accelerated one



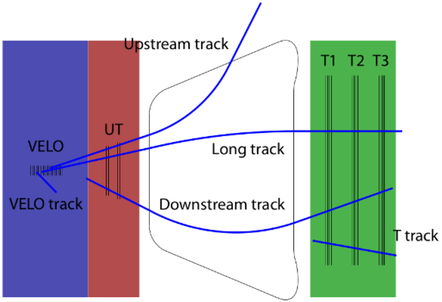
- HLT1 runs on EB nodes
- Reduce network bandwidth between EB and filter farms
- Free up filter farm CPU for HLT2 only

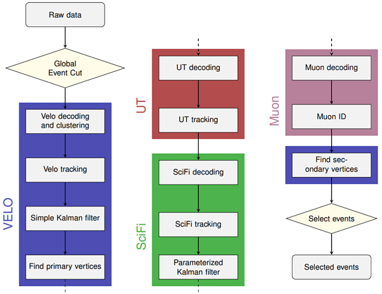
Warning: the exact numbers for BW, N(servers) have evolved since then

Allen project: HLT1 on GPU

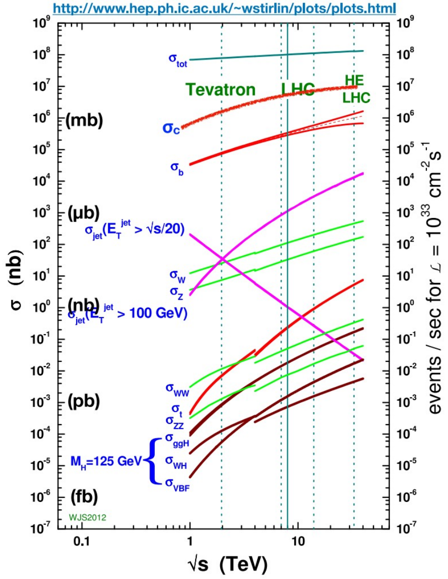
[Comput Softw Big Sci 4, 7 (2020)]

- Framework for GPU-based execution of an algorithm sequence
  [GitLab repo], [Documentation]
- Cross-architecture compatibility:
  Runs on CPU, NVidia GPU (`CUDA`), AMD GPU (`HIP`)
- Algorithm sequences defined in `python`, generated at runtime
- Three levels of parallelism:
  Intra-collision (tracks, clusters), collisions, collision batches





HLT2 signal rates

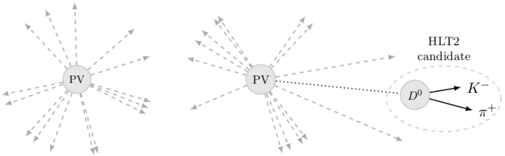


- Signal rates at $\mathcal{L} = 2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$:
  O(10) MHz charm
  O(1) MHz beauty
- Output bandwidth limited to 10 GB$/\,\mathrm{s}$.
  Up to 100 kHz with full event size of 100 kB.
- Need to reduce the event size for higher rate

Persistency model

[LHCb upgrade computing TDR]

Selective persistency: write out only the "interesting" part of the event.



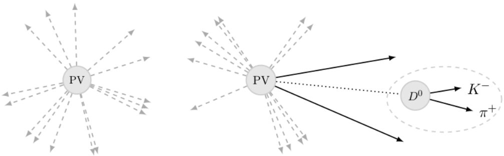
- `Turbo` stream:
  - Minimum output: only HLT2 signal candidates

  Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc. Advantage: Event size $O(10)$ smaller than RAW

Persistency model

[LHCb upgrade computing TDR]

Selective persistency: write out only the "interesting" part of the event.

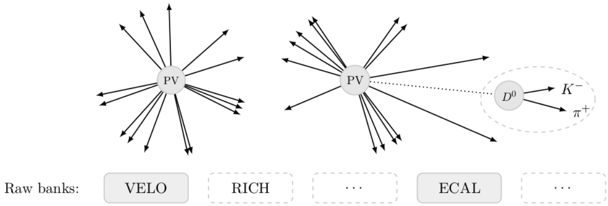


- `Turbo` stream:
  - Minimum output: only HLT2 signal candidates
  - Optionally: (parts of) *pp* vertex (*e.g.* "cone" around candidate for spectroscopy searches)

  Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc. Advantage: Event size $O(10)$ smaller than RAW

 [LHCb upgrade computing TDR]

Selective persistency: write out only the “interesting” part of the event.



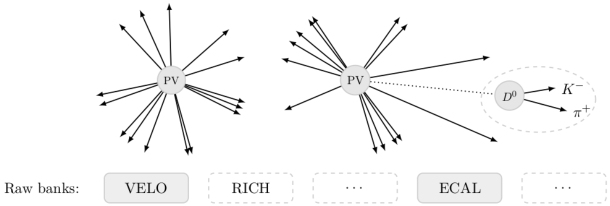
- Turbo stream:
  - Minimum output: only HLT2 signal candidates
  - Optionally: (parts of) *pp* vertex (*e.g.* "cone" around candidate for spectroscopy searches)

  Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc. Advantage: Event size $O(10)$ smaller than RAW
- FULL stream: all reconstructed objects in the event
  - + selected RAW banks

Persistency model

[LHCb upgrade computing TDR]

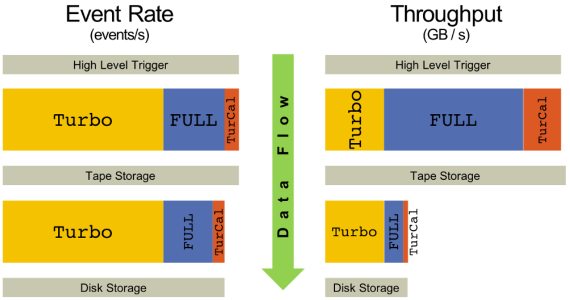
Selective persistency: write out only the "interesting" part of the event.



- `Turbo` stream:
  - Minimum output: only HLT2 signal candidates
  - Optionally: (parts of) pp vertex (*e.g.* "cone" around candidate for spectroscopy searches)

  Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc. Advantage: Event size $O(10)$ smaller than RAW
- `FULL` stream: all reconstructed objects in the event
  - $+$ selected RAW banks
- `TurCal` stream: HLT2 candidates and selected RAW banks

  Used for offline calibration and performance measurement



Rate and bandwidth to tape

stream	rate fraction	throughput (GB/s)	bandwidth fraction
FULL	26%	5.9	59%
Turbo	68%	2.5	25%
TurCal	6%	1.6	16%
total	100%	10.0	100%

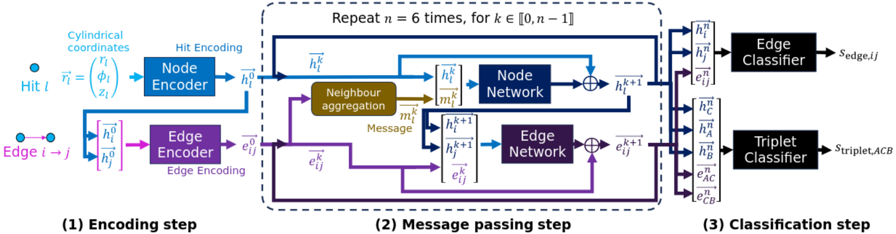
Disk bandwidth

stream	throughput (GB/s)	bandwidth fraction
FULL	0.8	22%
Turbo	2.5	72%
TurCal	0.2	6%
total	3.5	100%

GNN track finding in VELO

[A. Correa, *et al.*, PROC-CTD2023-34]

1. Hit graph construction
   - DNN to convert (r, ϕ, z) hits to 4D embedding space (close for hits from the same tracks)
   - k-NN algorithm in embedding space to connect hits into graph
2. Edge classification with GNN
   - Encode each hit and edge into 256D representation
   - 6-step *message passing* phase, update hit and edge encodings with DNNs
   - DNN edge classifier based on updated encodings



GNN track finding in VELO

3. Edge graph construction [A. Correa, *et al.*, PROC-CTD2023-34]
   - Solves the problem with shared hits
   - *Edge graph* (edges of hit graph are now nodes, edge-edge connections are *triplets* sharing a hit)
4. Triplet classification
   - Reuse hit and edge encodings from GNN step to avoid involving another GNN
   - DNN classifier for triplet score
5. Track construction
   - WCC (Weakly Connected Component) algorithm from Exa.TrkX

Long category	Efficiency	
	Allen	`ETX4VELO`
No electrons	99.26	99.28 (99.51)
Electrons	97.11	98.80 (99.22)
From strange	97.69	97.50 (98.06)

Velo-only category	Efficiency	
	Allen	`ETX4VELO`
No electrons	96.84	97.03 (97.86)
Electrons	67.81	85.10 (86.69)
From strange	93.53	93.07 (96.05)

	Allen	ETX4VELO	
		$d^2_{\max} = 0.010$	$d^2_{\max} = 0.020$
Ghost rate	2.18%	0.76%	0.81%

- Improves reconstruction of electrons wrt. default LHCb algorithm
- Lower ghost (fake track) rate with the similar efficiency