Latest Machine Learning developments for the LHCb experiment

Julián García Pardiñas[1]
on behalf of the LHCb Collaboration

1 CERN (Switzerland)

CERN EP/IT Data Science Seminar **10th July 2024**

Opportunities & challenges in Run 3 and beyond

Latest ML developments

Take-home messages

Opportunities & challenges in Run 3 and beyond

Latest ML developments

Disclaimer: a lot of developments for data analysis, not covered in this seminar.

Take-home messages

Opportunities & challenges in Run 3 and beyond

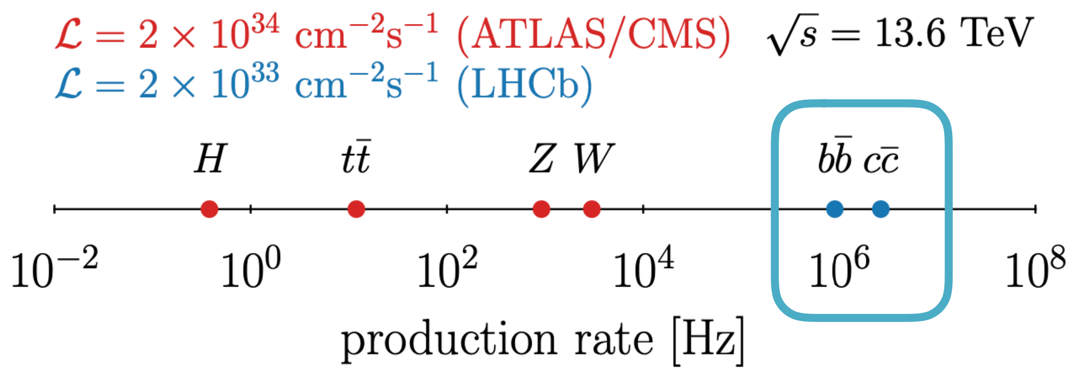
Latest ML developments

Take-home messages

The LHCb detector in Run 3

[The LHCb upgrade I]

- **5× increase** in instantaneous luminosity.
- **Huge signal-production rates**, at the MHz scale!



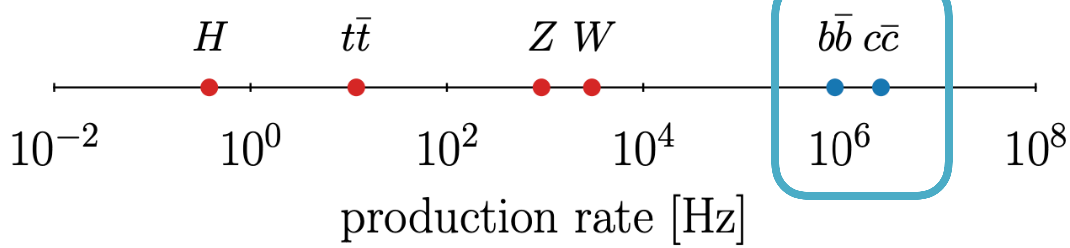
The LHCb detector in Run 3

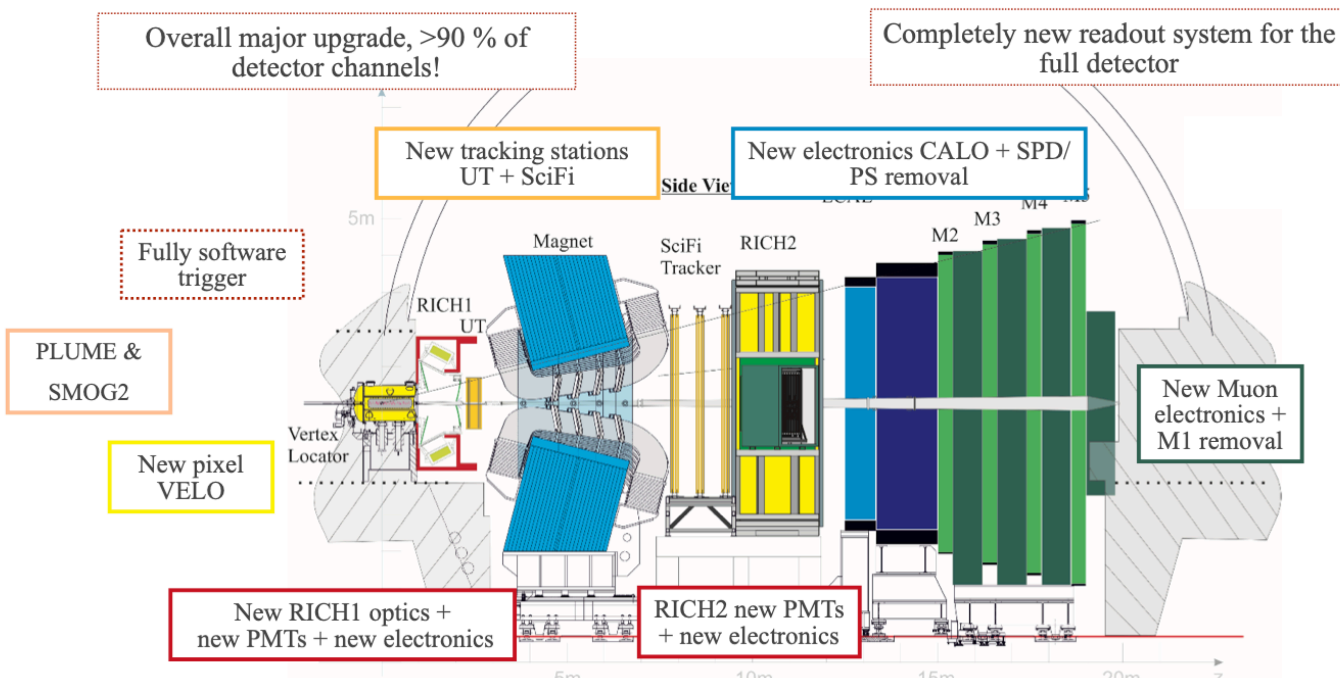
[The LHCb upgrade I]

- **5× increase** in instantaneous luminosity.
- **Huge signal-production rates**, at the MHz scale!

$\mathcal{L} = 2 \times 10^{34}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$ (ATLAS/CMS) $\sqrt{s} = 13.6$ TeV

$\mathcal{L} = 2 \times 10^{33}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$ (LHCb)

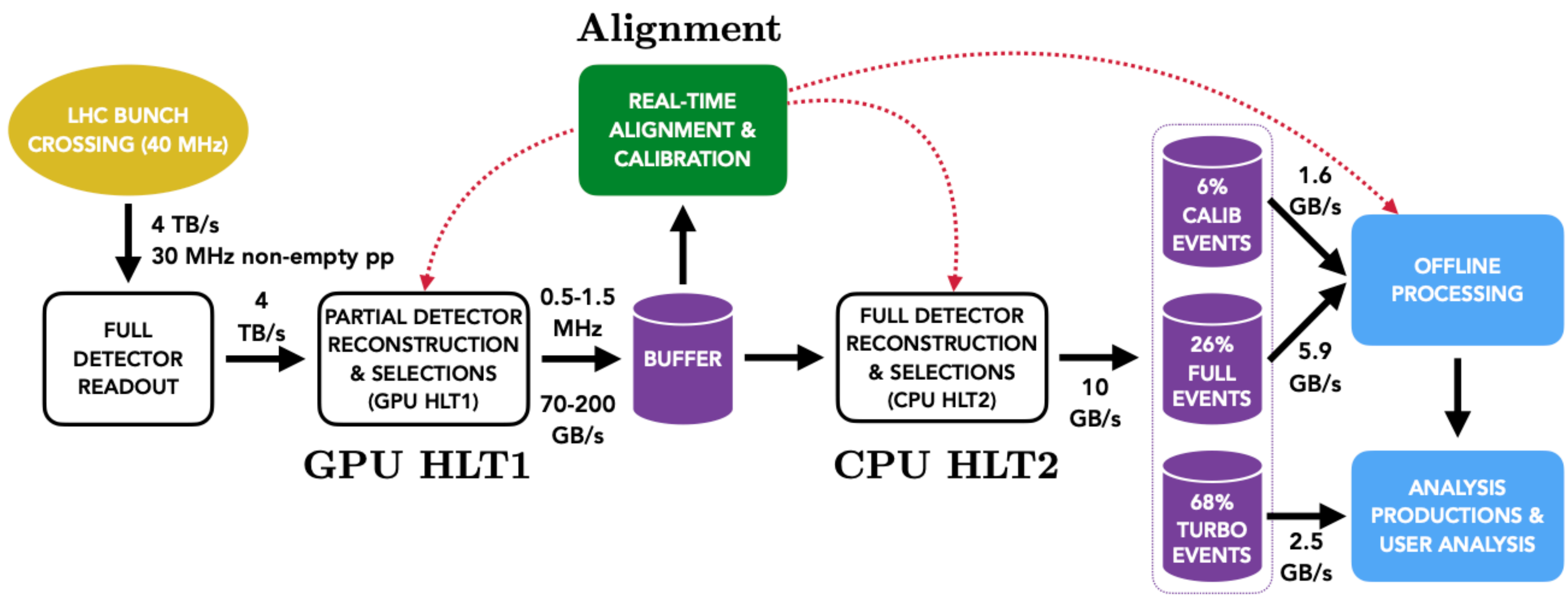




The Run 3 trigger framework

[LHCb-TDR-016]
[LHCb-TDR-018]

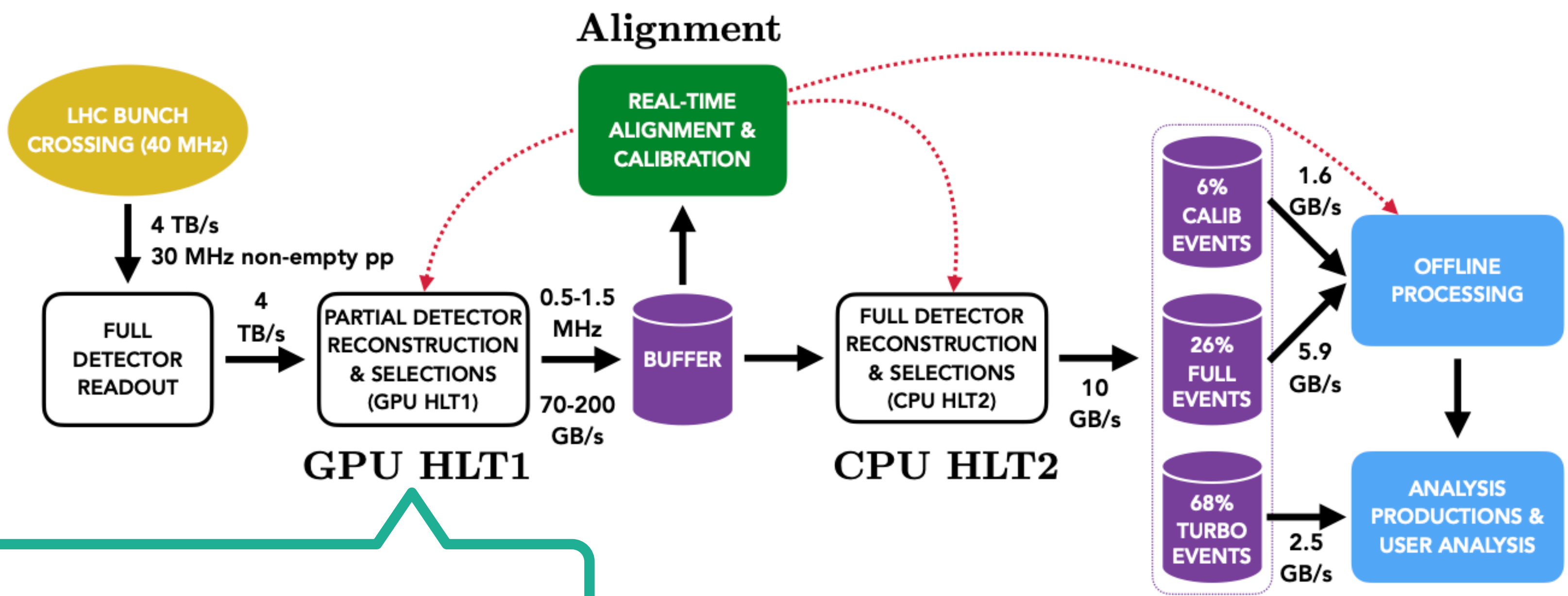
New **fully-software-based trigger**, based on GPU + CPU.



The Run 3 trigger framework

[LHCb-TDR-016]
[LHCb-TDR-018]

New **fully-software-based trigger**, based on GPU + CPU.



Partial event reconstruction + selections.

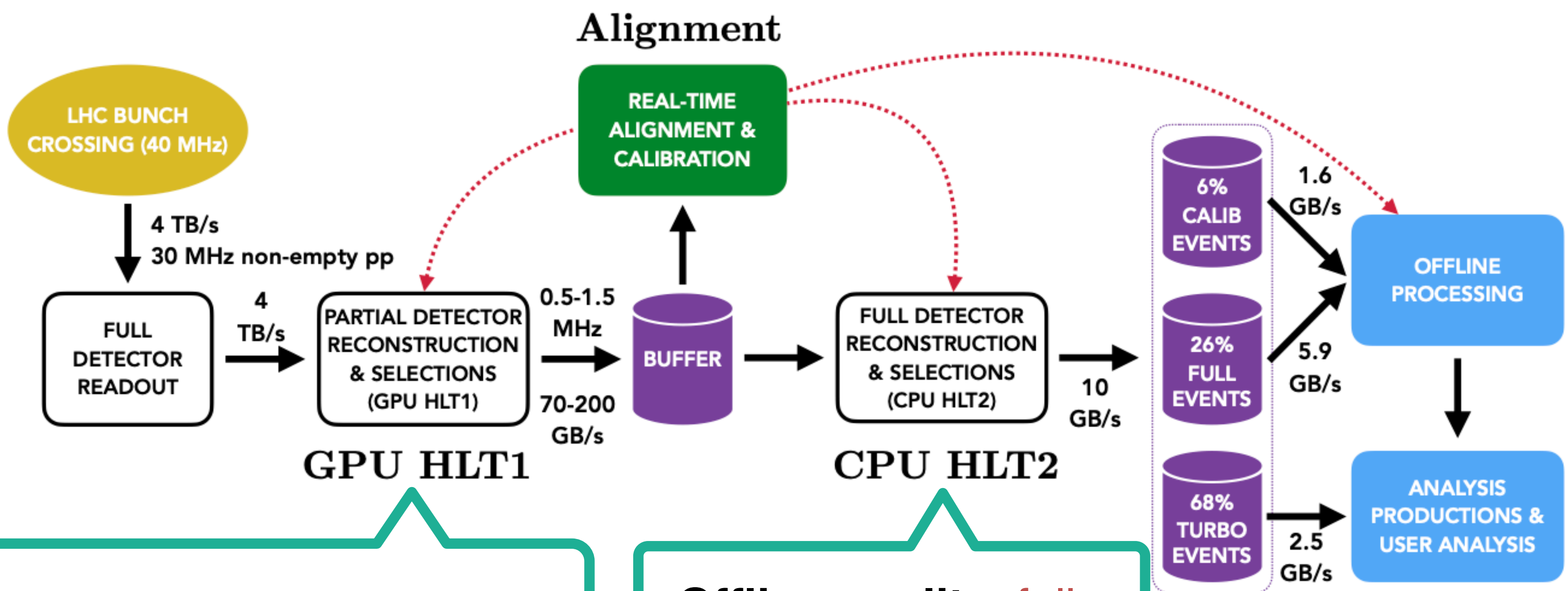
Based on ***Allen***: **GPU-based** trigger software framework.

[CSBS 4, 7 (2020)]
[LHCb-TDR-021]

The Run 3 trigger framework

[LHCb-TDR-016]
[LHCb-TDR-018]

New **fully-software-based trigger**, based on GPU + CPU.



Partial event reconstruction + selections.

Based on ***Allen*: GPU-based** trigger software framework.

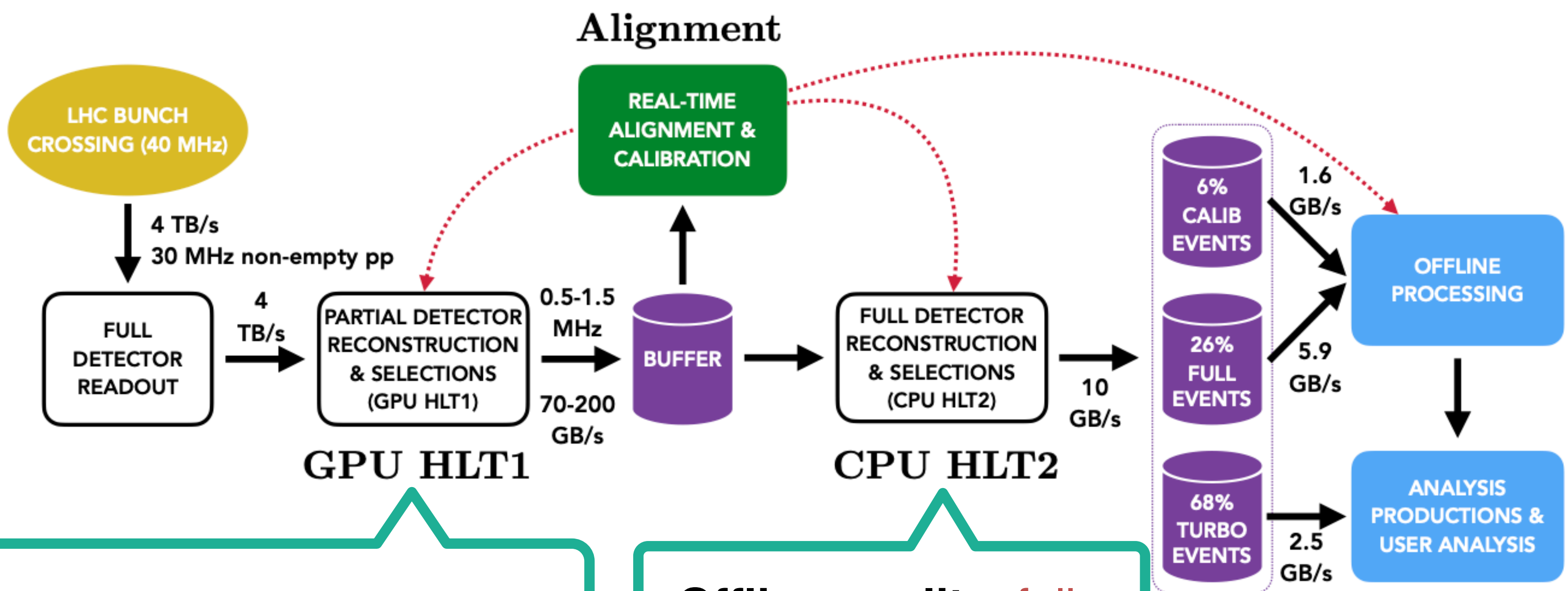
[CSBS 4, 7 (2020)]
[LHCb-TDR-021]

Offline-quality, full event reconstruction + selections.

The Run 3 trigger framework

[LHCb-TDR-016]
[LHCb-TDR-018]

New **fully-software-based trigger**, based on GPU + CPU.



Partial event reconstruction + selections.

Based on ***Allen***: **GPU-based** trigger software framework.

[CSBS 4, 7 (2020)]
[LHCb-TDR-021]

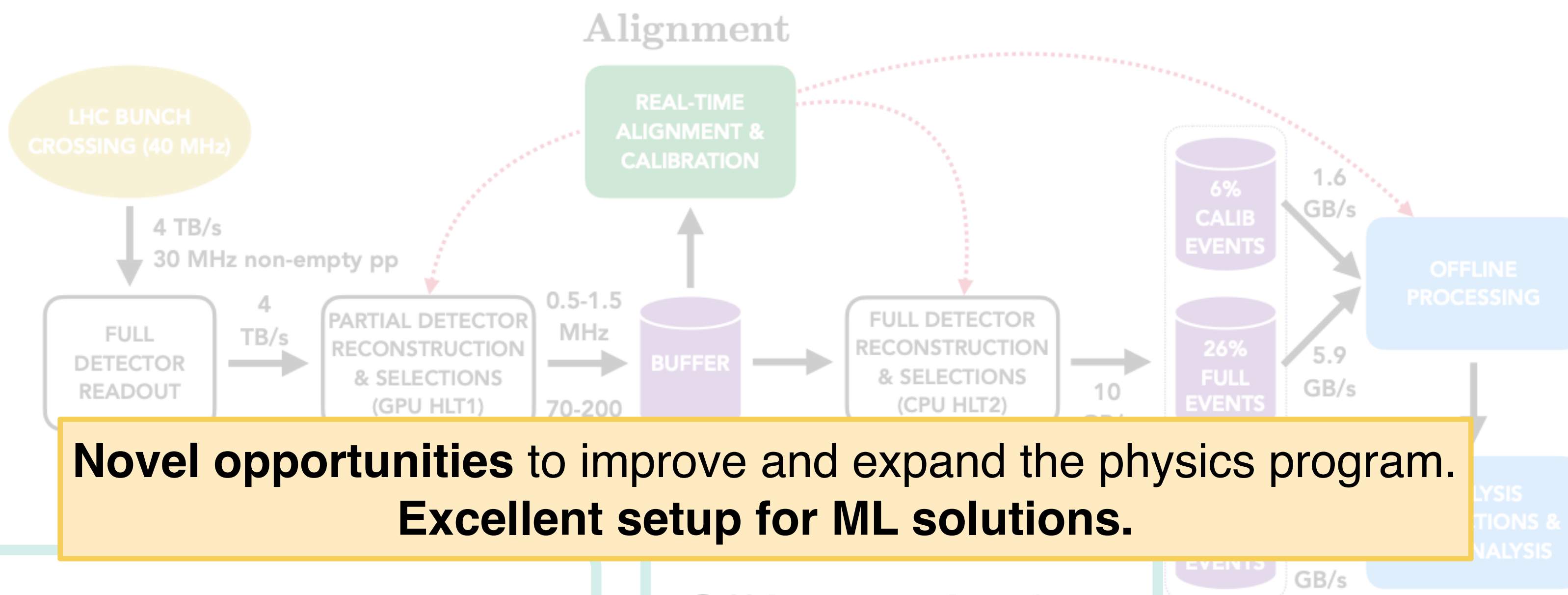
Offline-quality, full event reconstruction + selections.

Sprucing: offline processing+filtering, **can be re-done** (non-destructive).

The Run 3 trigger framework

[LHCb-TDR-016]
[LHCb-TDR-018]

New **fully-software-based trigger**, based on GPU + CPU.



Novel opportunities to improve and expand the physics program.
Excellent setup for ML solutions.

Partial event reconstruction + selections.

Based on ***Allen***: **GPU-based** trigger software framework.

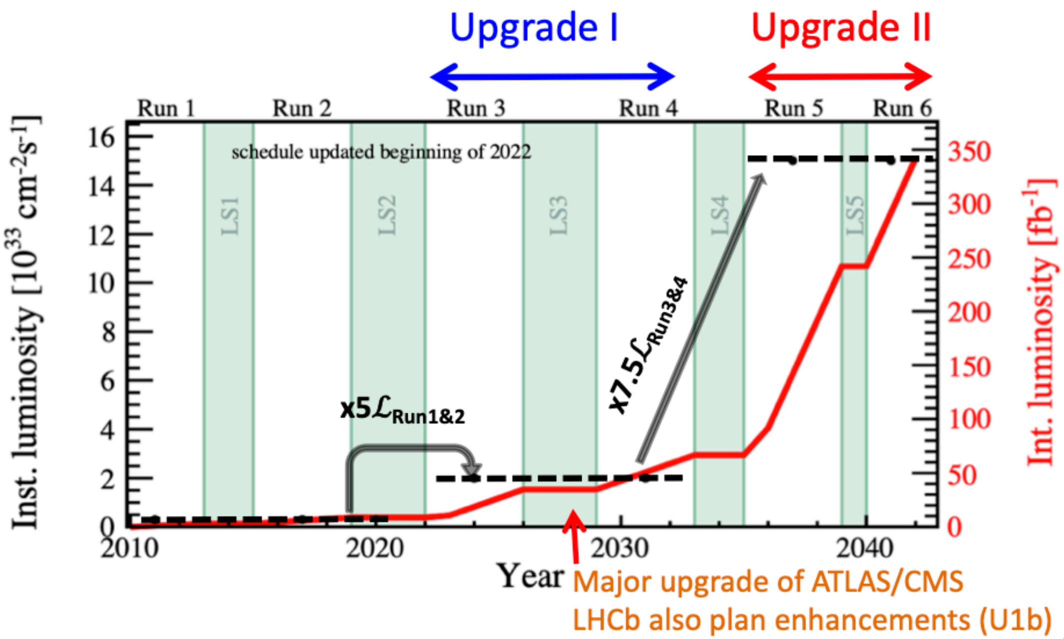
[CSBS 4, 7 (2020)]
[LHCb-TDR-021]

Offline-quality, full event reconstruction + selections.

Sprucing: offline processing+filtering, **can be re-done** (non-destructive).

What next?

[LHCC-2018-027]
[LHCB-TDR-023]

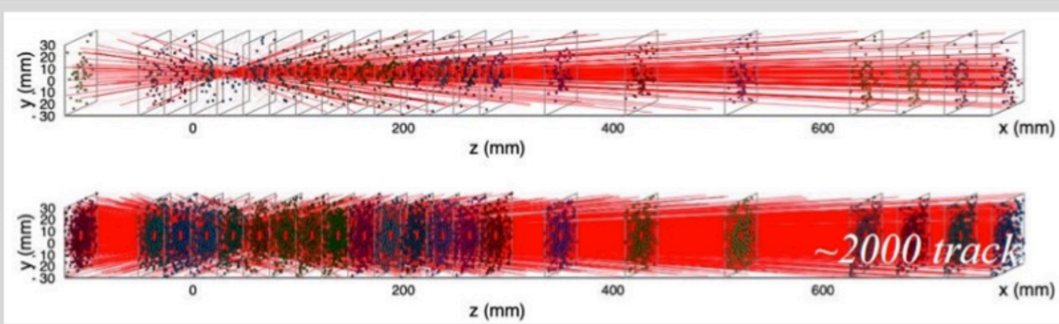


Further increase in instantaneous luminosity for **Upgrade II**.

Bigger challenges!
Potential next step:
GPUs also in HLT2.

Run 3: pile-up ~5

Upgrade II: pile-up ~40



In VELO

~6 cm

⚡ Big challenges ⚡

Trigger

Simulation

Data Quality Monitoring

The constraints of the trigger impose **high throughput** for HLT1 and HLT2 and **low bandwidth** to disk storage.

Bandwidth ∝ Trigger output rate × Event size

Complementary lines of action:

- **Speed up reconstruction algorithms**.
- Improve the data-volume reduction.
  - ➥ **Optimise event selection** (trigger lines).
  - ➥ **Optimise event filtering** (selective persistency)

  [JINST 14 (2019) 04, P04006].

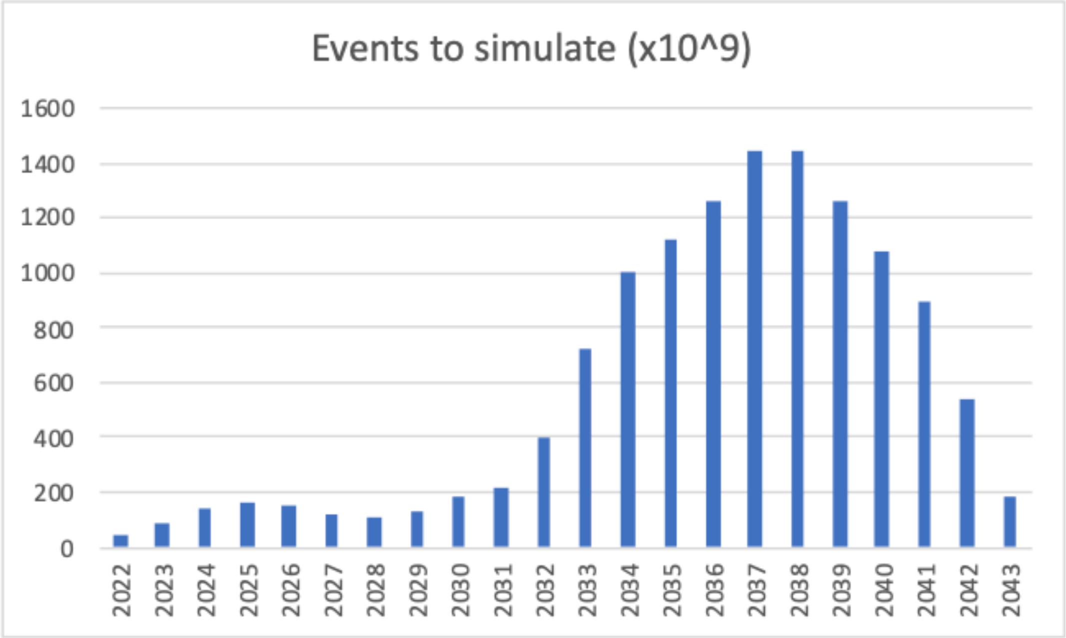
Big challenges

Trigger

Simulation

Data Quality Monitoring

More beam data requires more simulated data.
- Simulation took ~90% of the CPU resources in Run 2.
- Very strong **need for (ultra) fast simulations**.



(See details in this talk.)

Big challenges

Trigger

Simulation

Data Quality Monitoring

Identifying detector anomalies promptly and ensuring the data is safe for physics analysis is always important.

Data Quality Monitoring (DQM) in **detector commissioning times:**

- Challenging: frequent changes in the setup.
- Crucial: effective identification and communication of problems impacts the commissioning schedule.

Currently, the task is done by rotating shifters, hence **very demanding in terms of person power**.
Huge gains could come from automation.

Opportunities & challenges in Run 3 and beyond

Latest ML developments

Take-home messages

Latest ML developments

In/close to production stage

- Trigger
- Simulation

R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

Latest ML developments

In/close to production stage

- Trigger
- Simulation

R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

Lipschitz networks

The algorithms to be used for filtering in the trigger need to be very fast and **avoid introducing complicated effects** in the signal selection efficiencies.

LHCb has being using ML algorithms based on decision trees in trigger selections for many years [arXiv:1510.00572].

Monotonic Lipschitz neural networks
[arXiv:2112.00038]

Impose desired constraints in the behaviour of the network **by construction**:

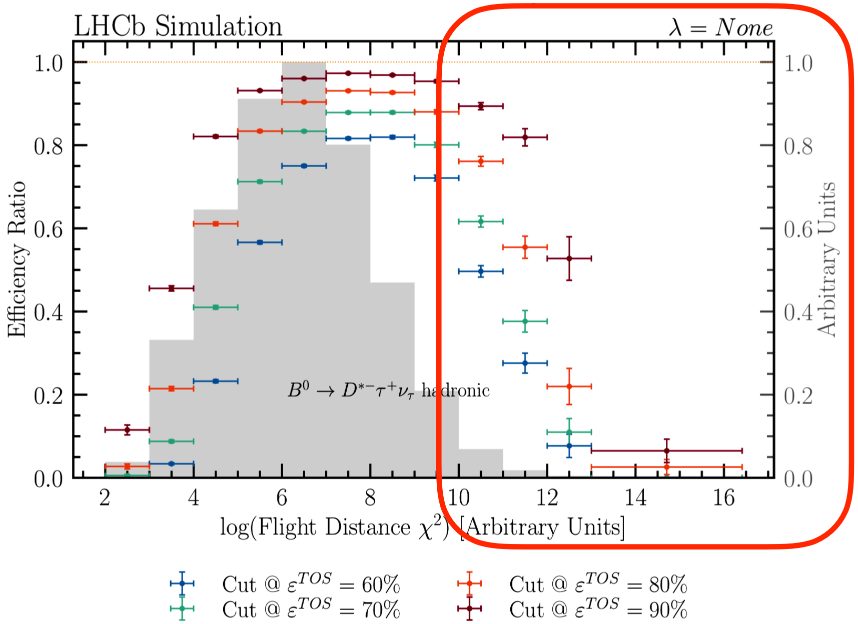
Robustness against detector instabilities and simulation inaccuracies.
➥ Technically done via weight-normalisation scheme during training.

Monotonicity in certain features for out-of-distribution guarantees.
➥ Technically done by adding a residual connection to the network.

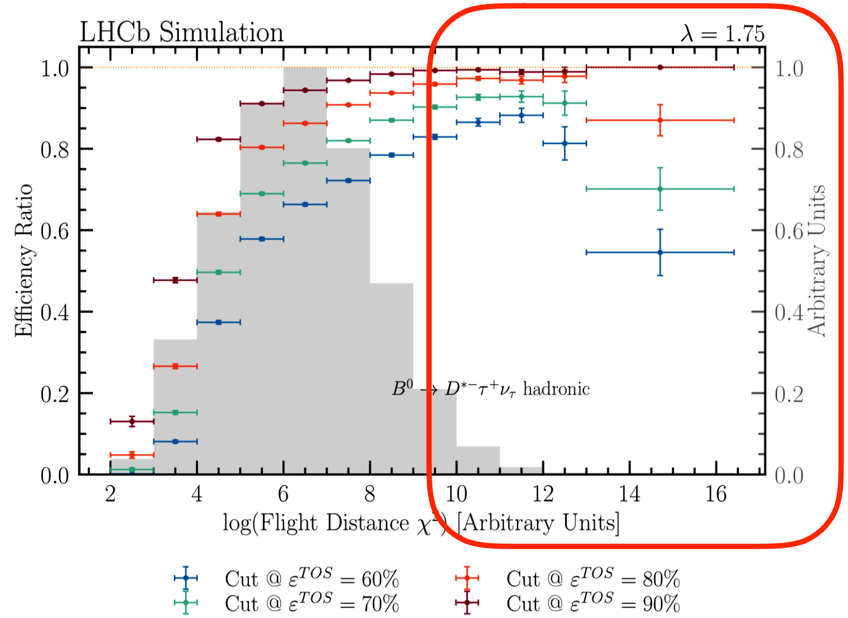
Lipschitz networks: inclusive trigger selections

[arXiv:2312.14265]

Two- and three-body topological triggers in HLT2, aimed at identifying beauty secondary vertices. → Monotonicity imposed in the IP χ^2 and the p_T.



Unconstrained NN



Lipschitz monotonic NN

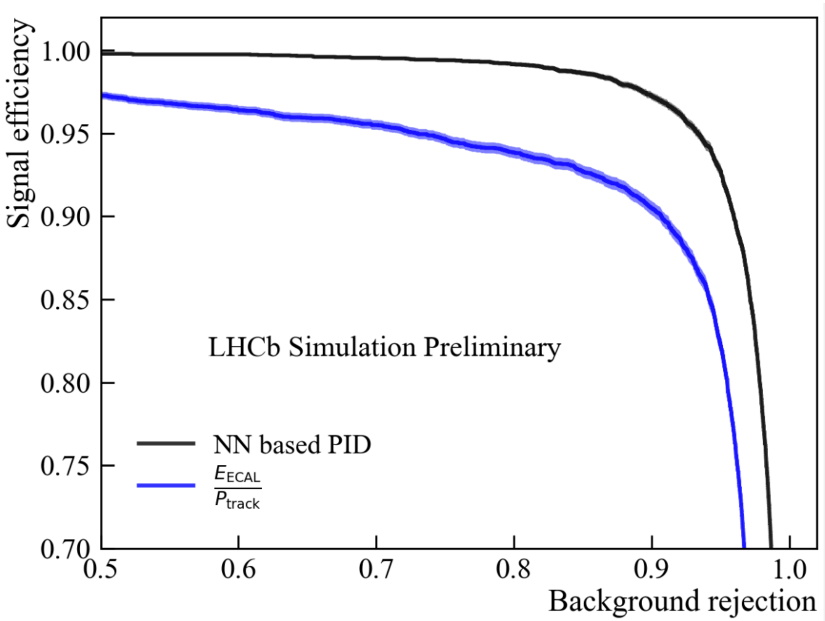
Enhanced sensitivity to long-lived candidates, particularly useful for searches of feebly-interacting particles.

Lipschitz networks: other applications

[LHCB-FIGURE-2024-003]

This type of network is now also used for **electron ID at the HLT1 level**, implemented in Allen.

Large improvement with respect to the conventional (not ML based) algorithm.



Next: the Lipschitz networks are also being investigated in **tracking** and **ghost (fake-track) rejection** algorithms.

ML model serving in the trigger

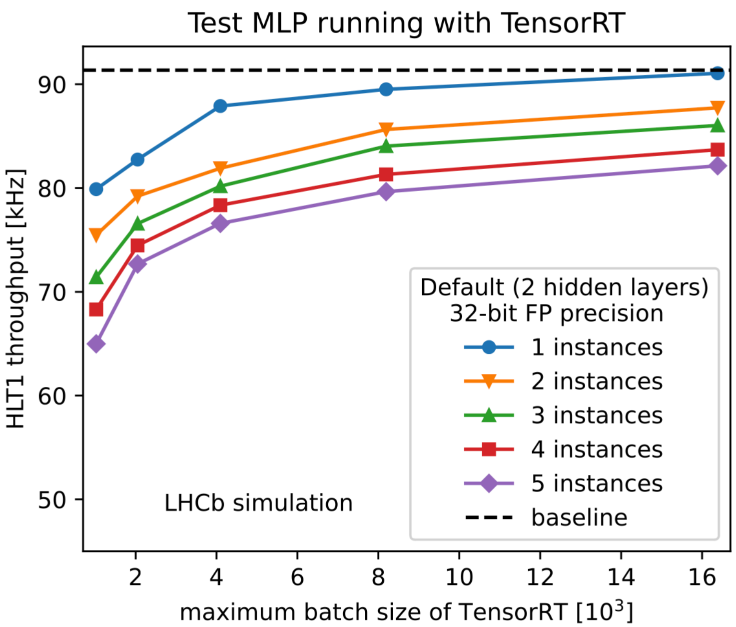
As more and more neural networks permeate HLT1 and HLT2, **maintaining hard/hand-coded implementations becomes increasingly challenging**.

ML model serving in the trigger

As more and more neural networks permeate HLT1 and HLT2, **maintaining hard/hand-coded implementations becomes increasingly challenging**.

Study of **flexible ML model serving backends**, such as TensorRT for HLT1 and ONNXRuntime for HLT2 (see details in this talk).

[LHCB-FIGURE-2023-006]

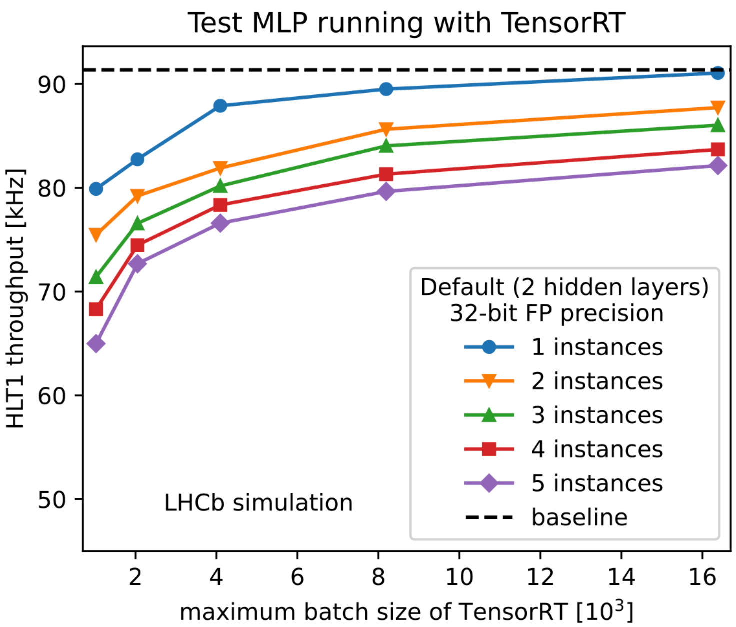


ML model serving in the trigger

As more and more neural networks permeate HLT1 and HLT2, **maintaining hard/hand-coded implementations becomes increasingly challenging**.

Study of **flexible ML model serving backends**, such as TensorRT for HLT1 and ONNXRuntime for HLT2 (see details in this talk).

[LHCB-FIGURE-2023-006]



Efforts ongoing towards developing **flexible and standardised pipelines for ML model serving, as well as for ML model training**, that facilitate the long-term maintenance.

Latest ML developments

In/close to production stage

- Trigger
- Simulation

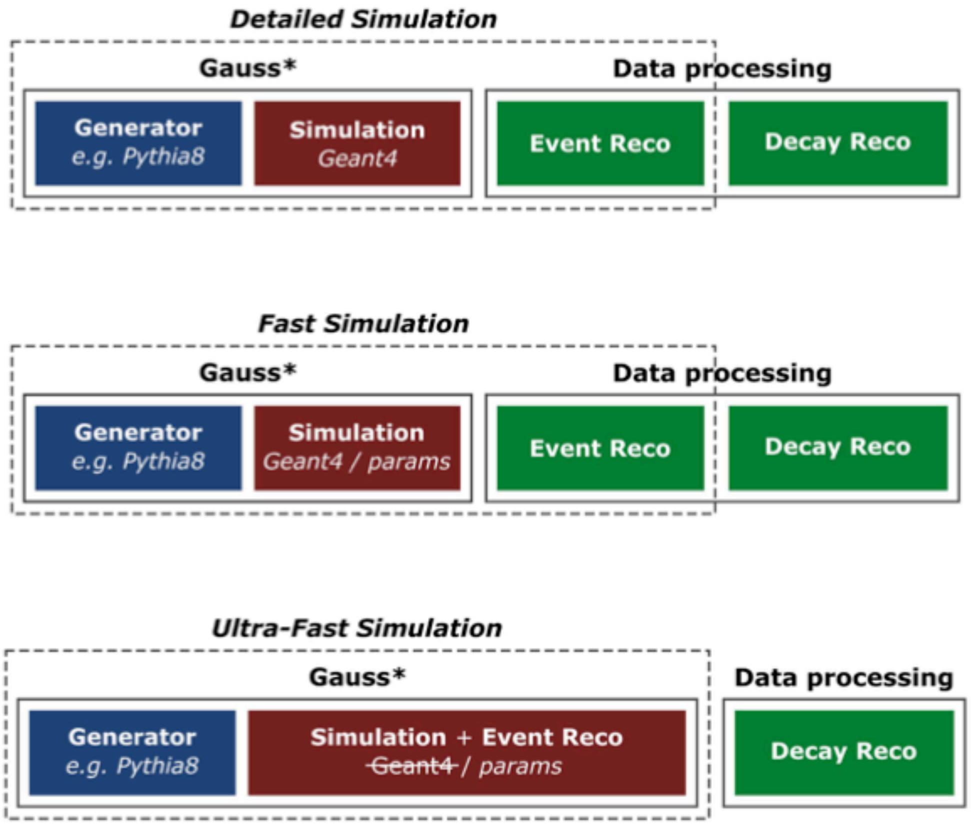
R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

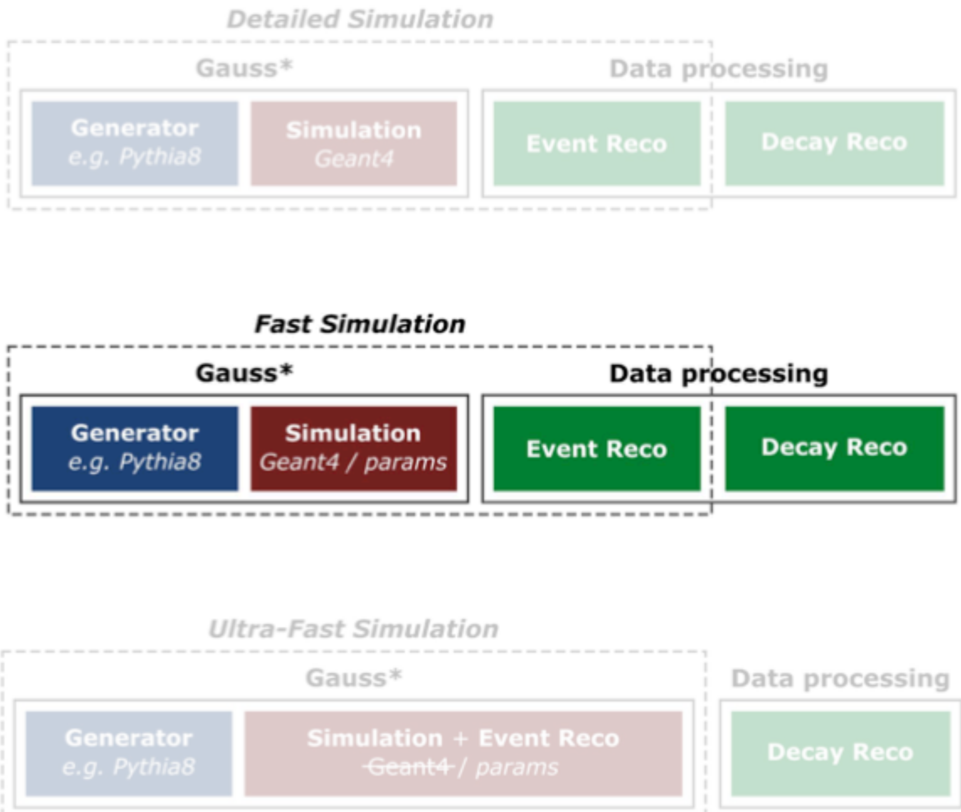
Types of simulation at LHCb

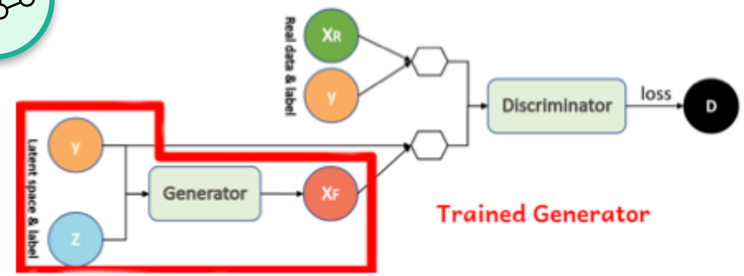
Multiple complementary techniques to speed up the simulation process.



Types of simulation at LHCb

Multiple complementary techniques to speed up the simulation process.

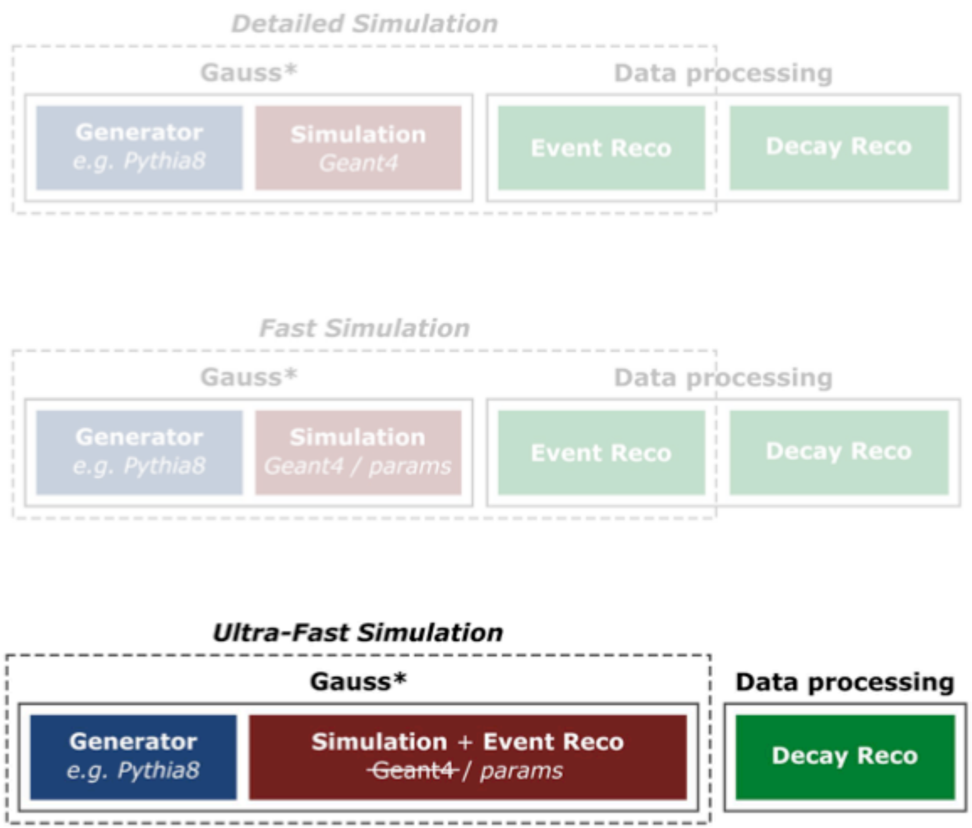




E.g. shower simulation with GANs (see details in this poster).
➥ Detector simulation **speed up factor 20x**.

Types of simulation at LHCb

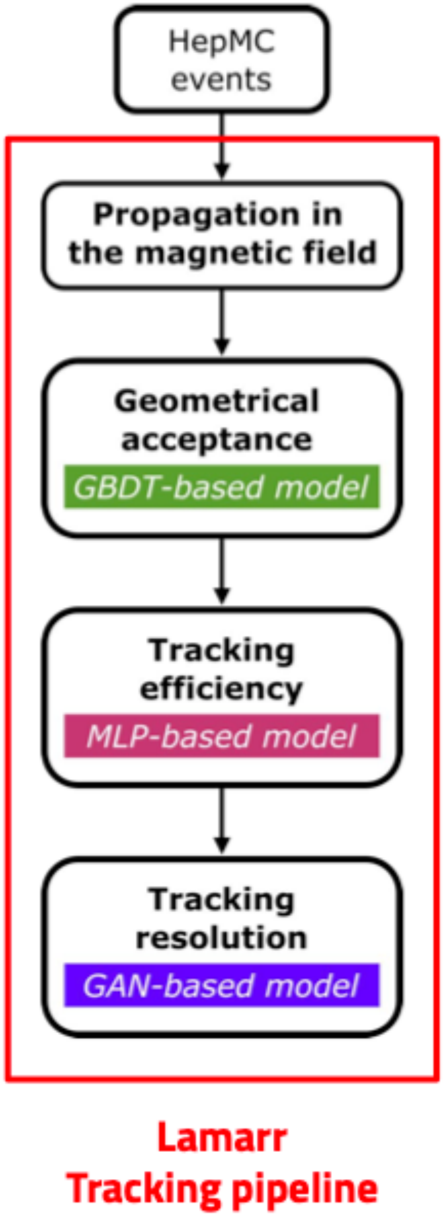
Multiple complementary techniques to speed up the simulation process.



LAMARR, ultra-fast simulation using ML-based parametrizations [arXiv:2309.13213].
➥ Detector simulation **speed up factor of 2 orders of magnitude**.

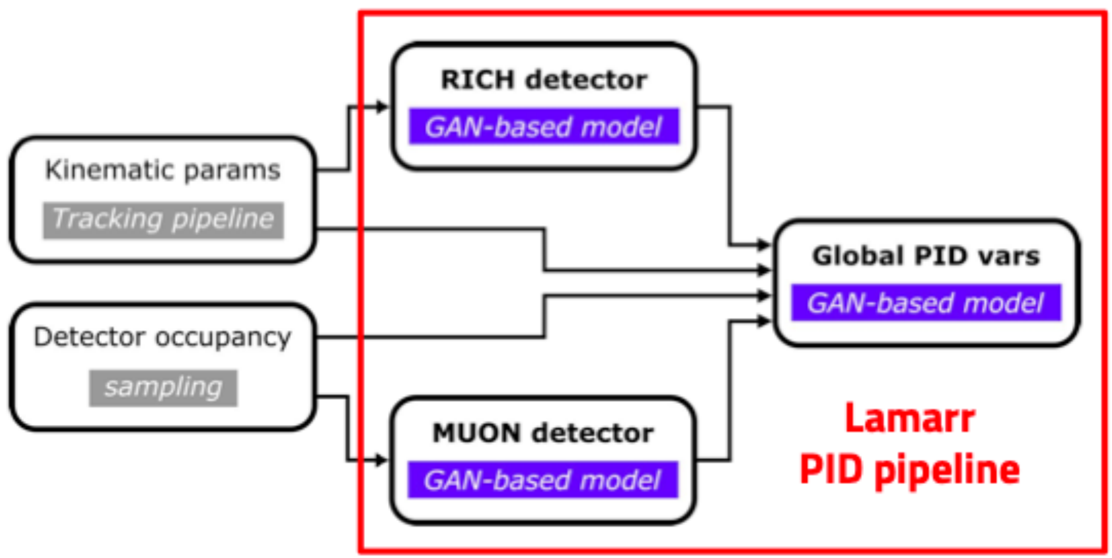
LAMARR

[arXiv:2309.13213]



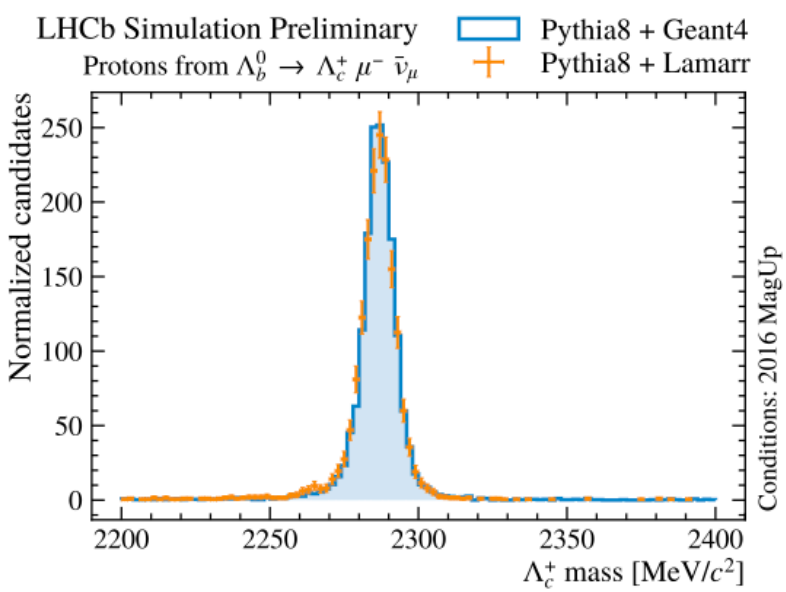
Pipeline of modules parameterising both the detector response and the reconstruction algorithms of the LHCb experiment.

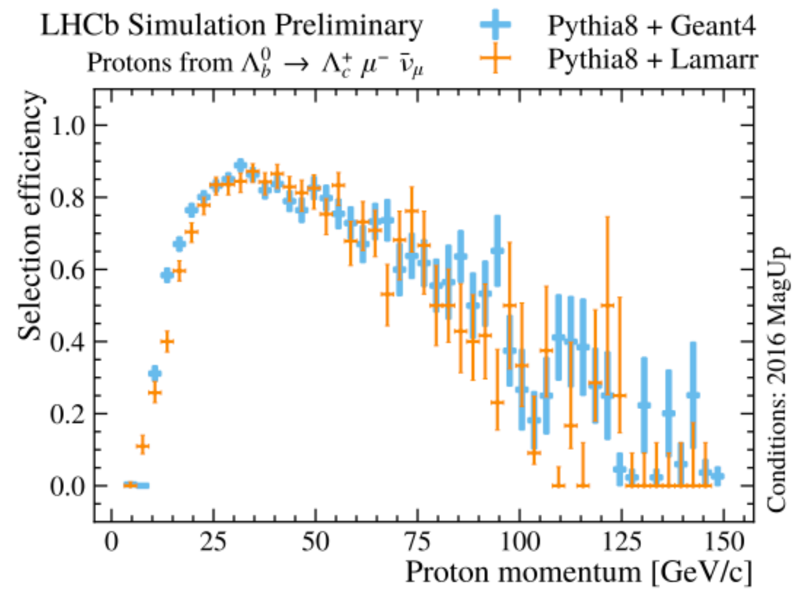
➥ **Output high-level quantities directly**, including uncertainties on reconstructed quantities.



LAMARR: status and next steps

[arXiv:2309.13213]





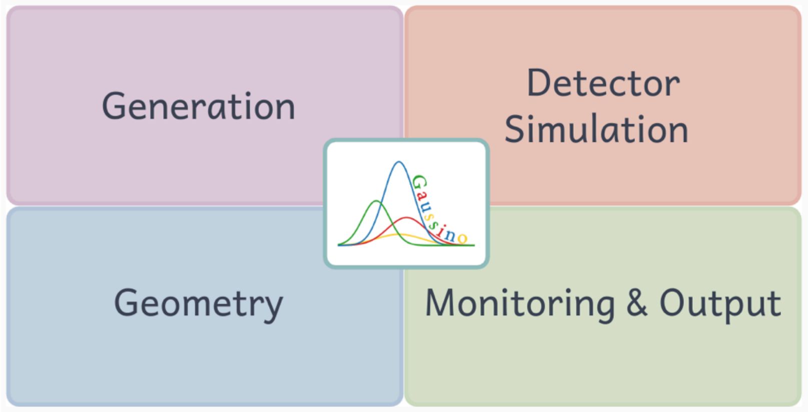
First validation studies show **excellent performance**.

LAMARR is built within the LHCb simulation framework.

➥ Next: integration in the **production system**.

From LHCb-specific to experiment-independent

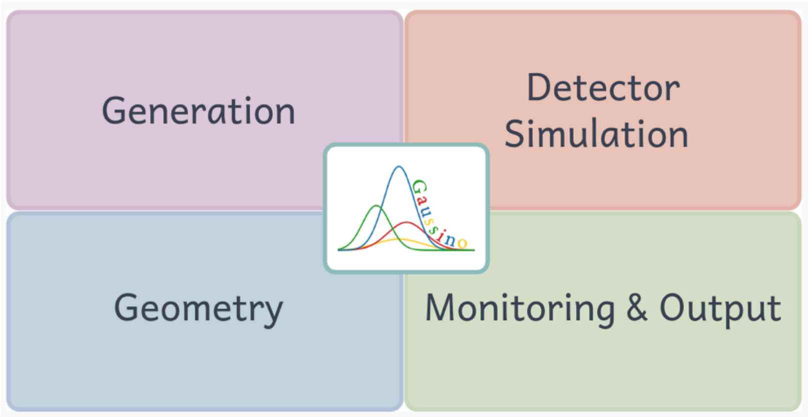
Gaussino is the new core simulation framework extracted from the LHCb simulation framework [https://gaussino.docs.cern.ch/].



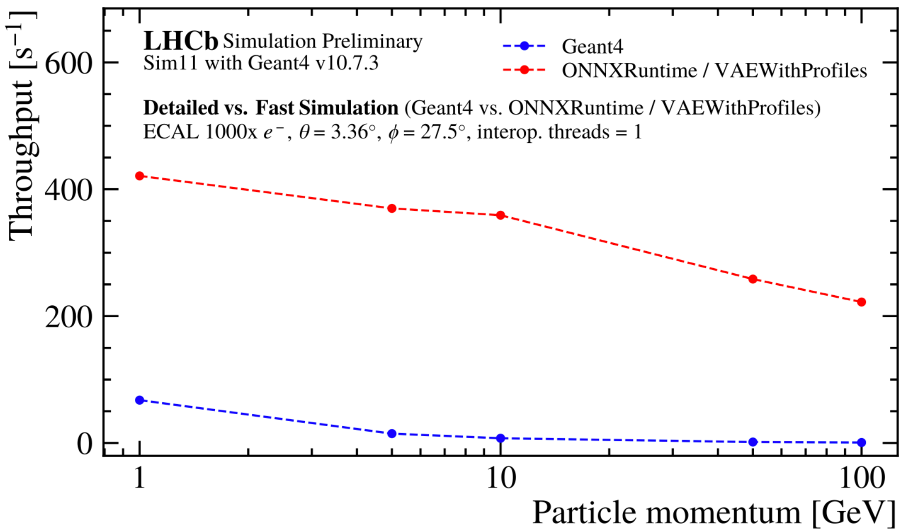
From LHCb-specific to experiment-independent

Gaussino is the new core simulation framework extracted from the LHCb simulation framework [https://gaussino.docs.cern.ch/].

- **Ongoing integration of LAMARR in Gaussino**, via SQLamarr (repo, docs) and PyLamarr (repo).



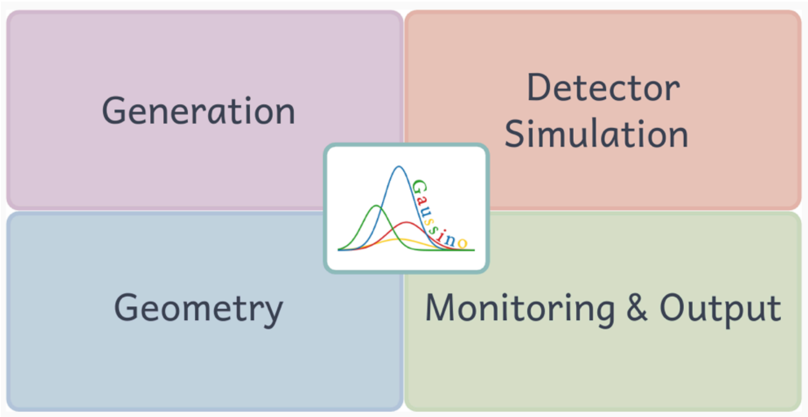
In addition, **general ML model serving interface implemented in Gaussino**, including pyTorch C++ API and ONNXRuntime (see details in this poster and this talk).



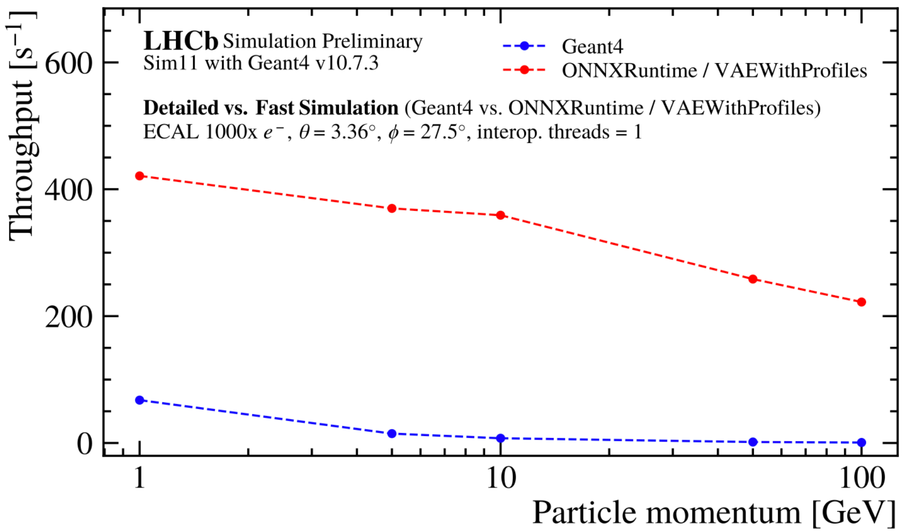
From LHCb-specific to experiment-independent

Gaussino is the new core simulation framework extracted from the LHCb simulation framework [https://gaussino.docs.cern.ch/].

- **Ongoing integration of LAMARR in Gaussino**, via SQLamarr (repo, docs) and PyLamarr (repo).



In addition, **general ML model serving interface implemented in Gaussino**, including pyTorch C++ API and ONNXRuntime (see details in this poster and this talk).



This could be the basis for a more general **ML serving interface for Gaudi.**

Latest ML developments

In/close to production stage

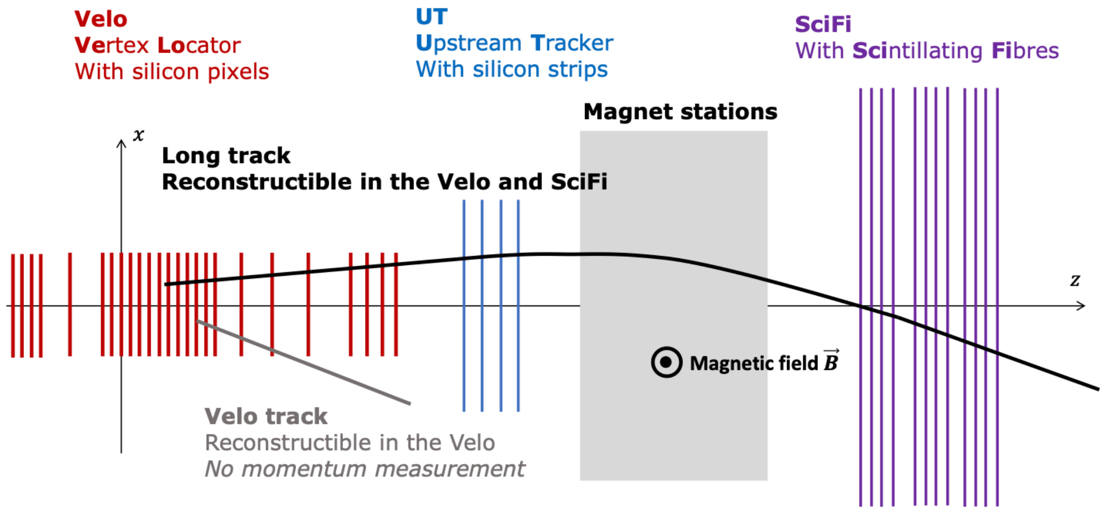
- Trigger
- Simulation

R&D stage

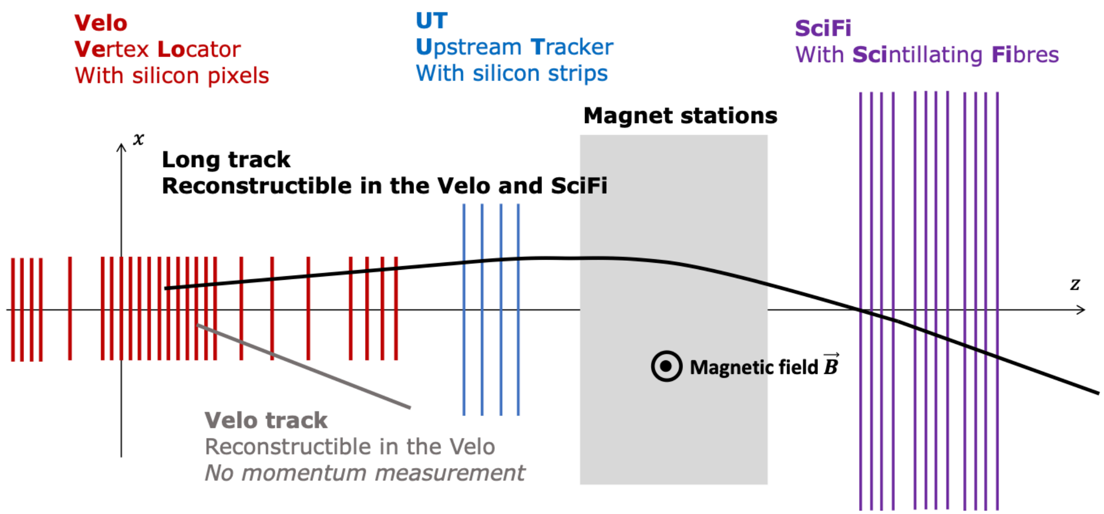
- **VELO tracking**
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

Track finding in LHCb



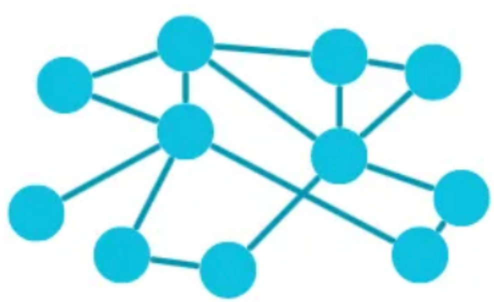
Track finding in LHCb



Conventional algorithms for track finding often **scale quadratically (or worse)** with the number of hits.

The usage of **Graph Neural Networks (GNNs) can offer near-linear inference** with # hits [Eur. Phys. J. C 81, 876 (2021)].

➥ High parallelisation potential thanks to the GPU-based trigger.

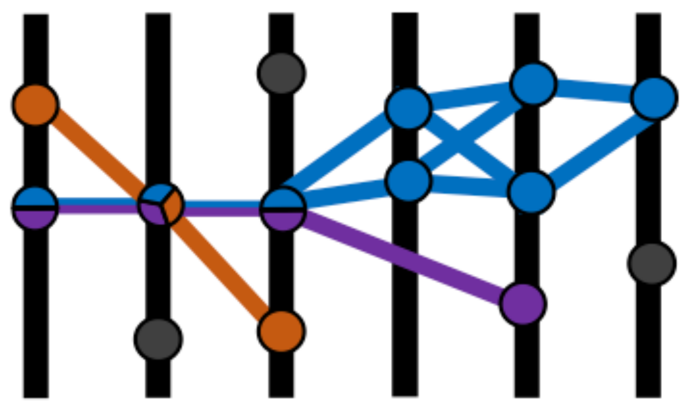


The ETX4VELO project

[arXiv:2406.12869]

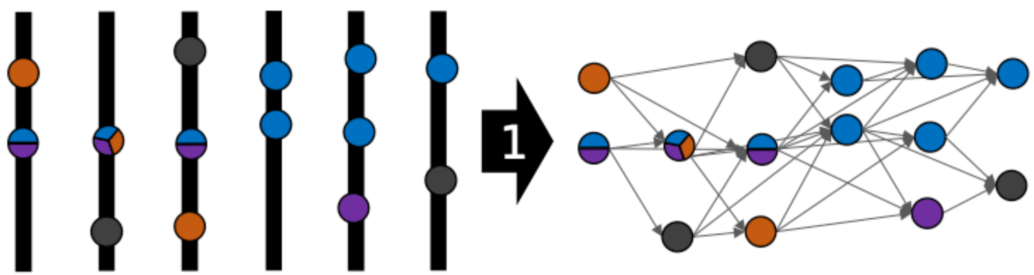
Based on the Exa.TrkX approach [Eur. Phys. J. C 81, 876 (2021)], originally tailored for 4π tracking detectors in a magnetic field, akin to ATLAS and CMS.

Goal: reconstruct forward tracks without a magnetic field, accounting for hit overlaps and inefficiencies.
➥ ETX4VELO introduces **new triplet-related stages** compared to the Exa.TrkX approach, **to handle tracks with shared hits**.



The model

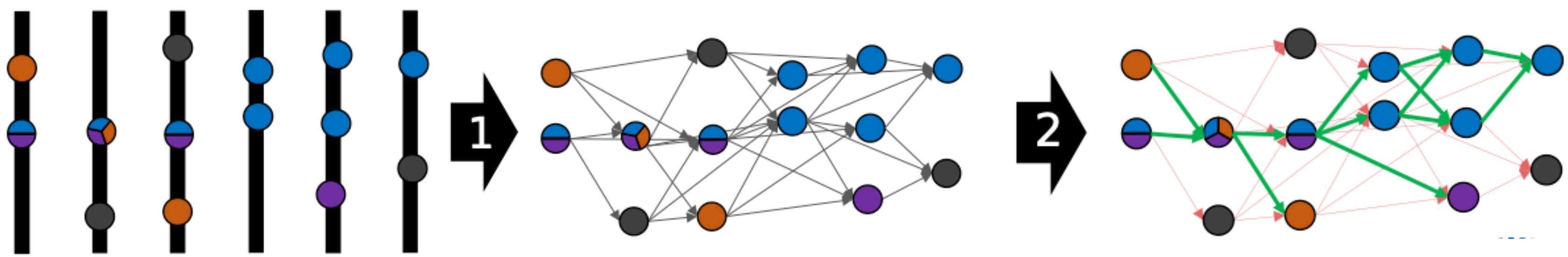
[arXiv:2406.12869]



Hit graph construction: DNN that embeds each hit into a latent space + k-Nearest Neighbours.

The model

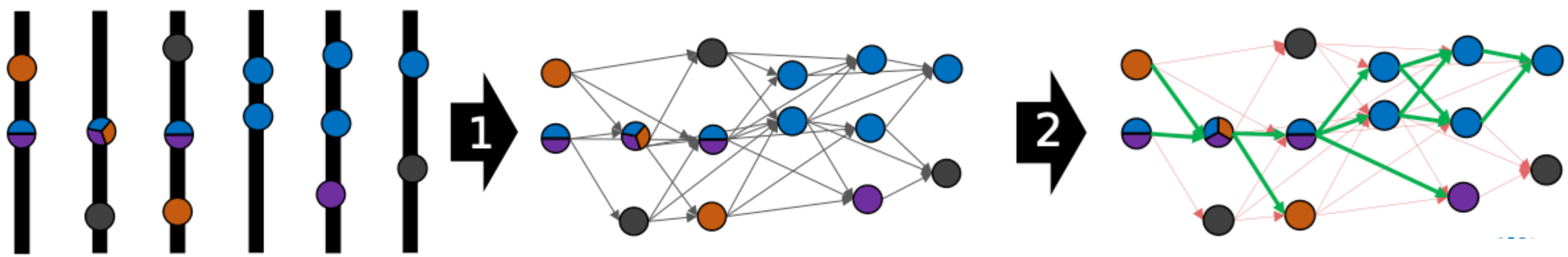
[arXiv:2406.12869]

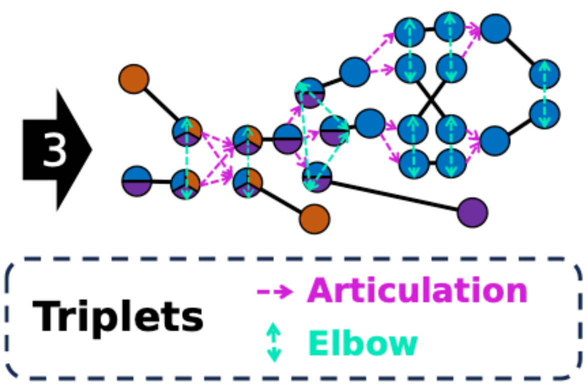


Edge classification: GNN.

The model

[arXiv:2406.12869]

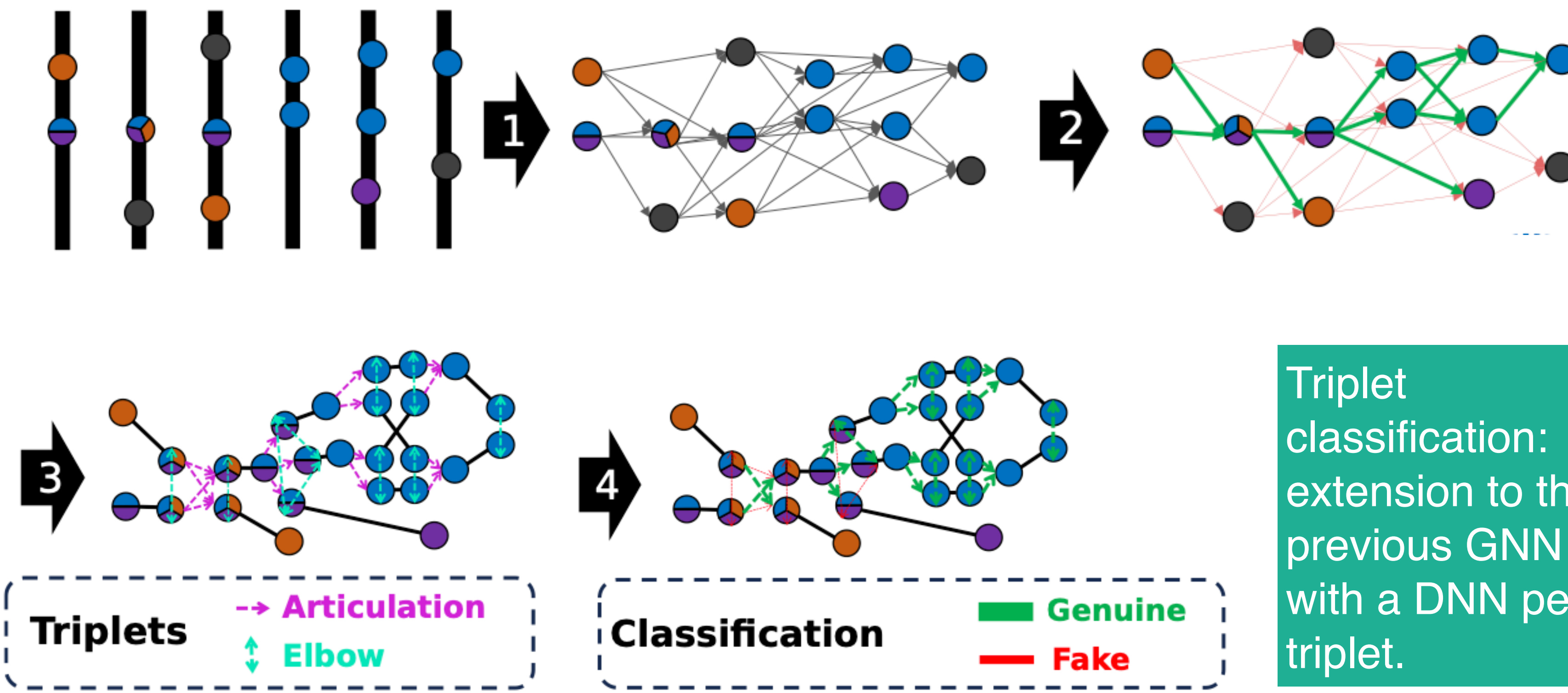




Formation of triplets (edge-edge connections).

The model

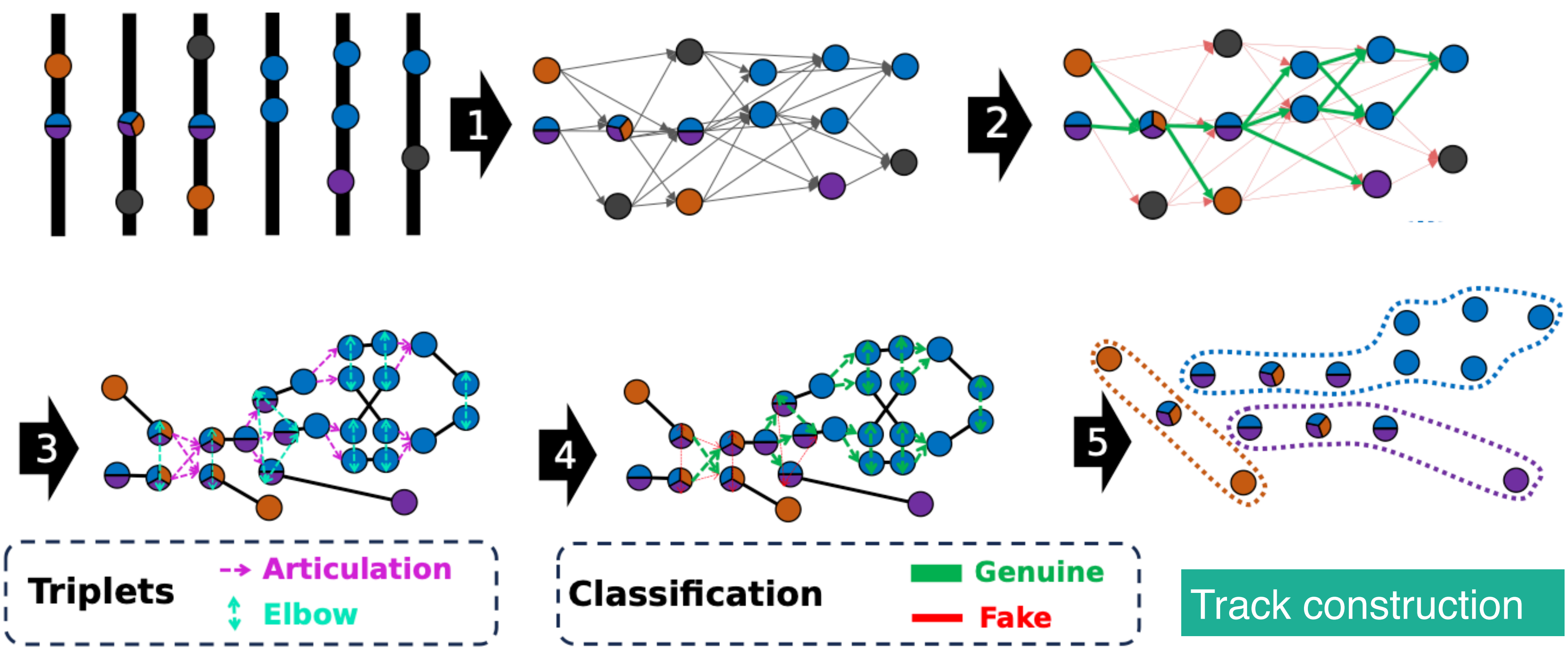
[arXiv:2406.12869]



Triplet classification: extension to the previous GNN with a DNN per triplet.

The model

[arXiv:2406.12869]



Results

[arXiv:2406.12869]

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

Compared to the default algorithm in LHCb:

- Similar efficiency.
- **Improved reconstruction for electrons.**
- **Lower ghost (fake-track) rate.**

Next: optimise the throughput for usage in HLT1.

➥ **Batching over events in the GPU recently achieved.**

Latest ML developments

In/close to production stage

Trigger

Simulation

R&D stage

VELO tracking

PV finding

DFEI

Anomaly detection

Neutral particle simulation

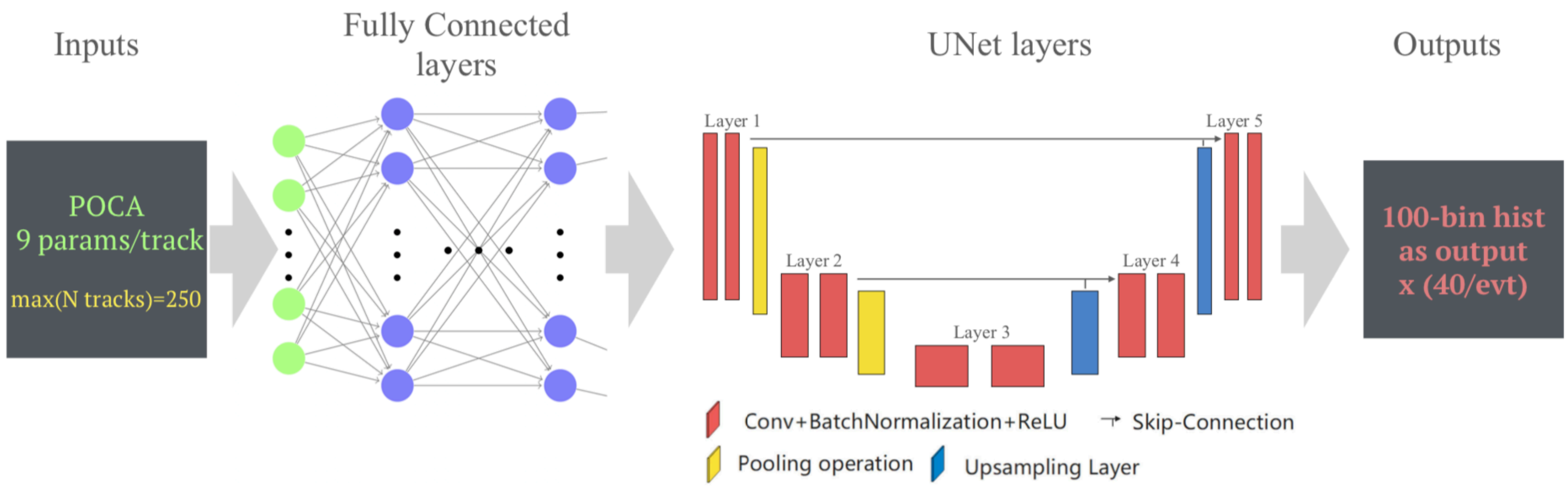
Others

Data Quality Monitoring

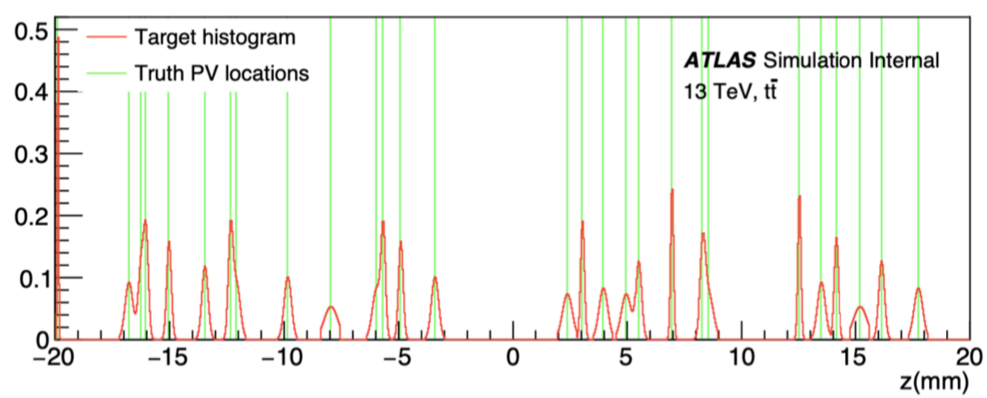
PV finding with a hybrid model

[arXiv:2309.12417]

Collaborative effort between people in **LHCb and ATLAS**.
LHCb uses a hybrid model, composed of DNN + Convolutional Neural Network (CNN).

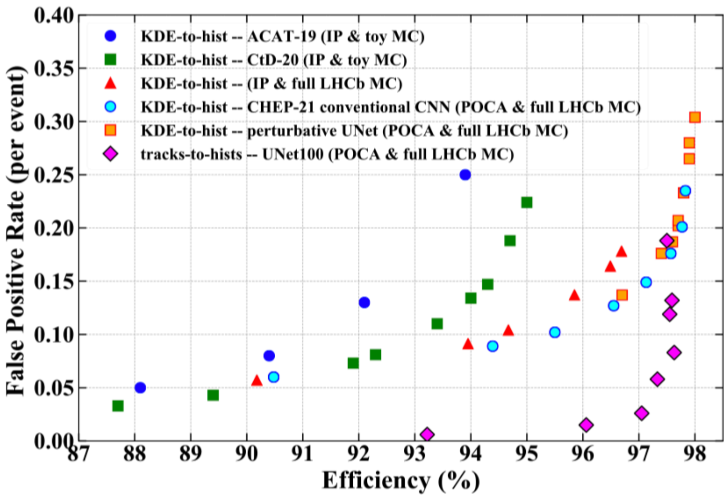


- Input: tracks in the event.
- Target: Gaussian distributions whose heights and widths reflect the expected **PV resolutions**.

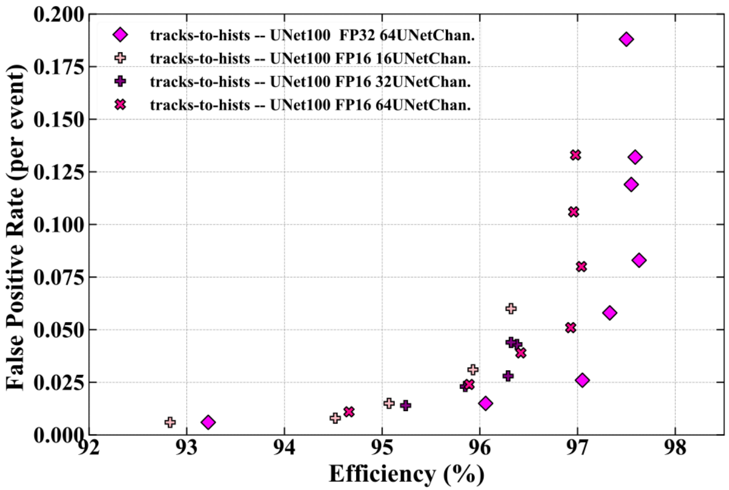


PV finding with a hybrid model [arXiv:2309.12417]

Many iterations of **design improvement** aiming at increased performance.



Studies towards **speeding up the inference**, for application in HLT1.



Next:
implementation in the Allen framework.

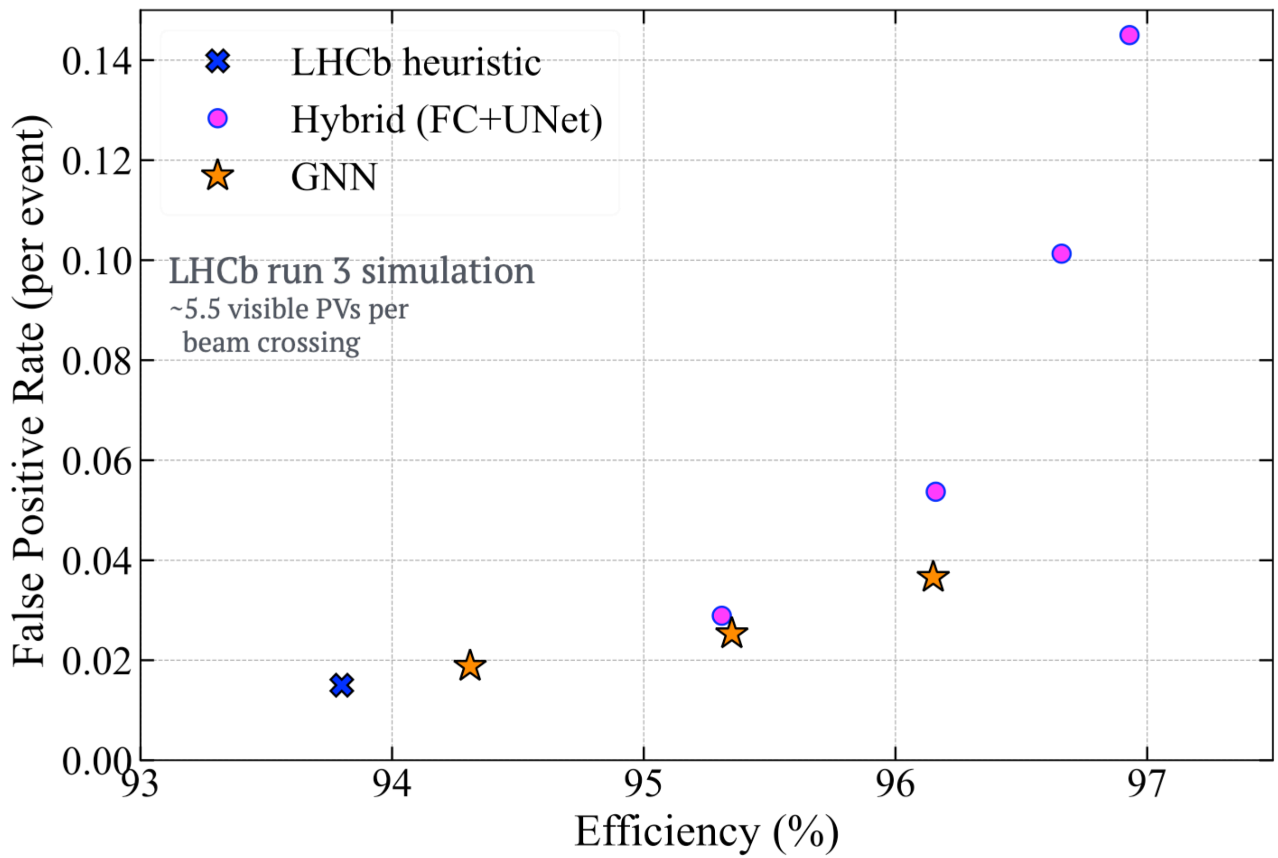
In ATLAS [ATL-PHYS-PUB-2023-011]

- Modified algorithm, based on CNN.
- Comparison to the default AMVF algorithm:
  - ➡ **2x better vertex resolution**.
  - ➡ Similar efficiency and false positive rates.

PV finding with GNN

Alternative approach: GNN model based on the ETX4VELO one, tracks represented as nodes, same input features as the hybrid model, coordinates of associated PV as target for the nodes, custom loss (see details in this poster).

The GNN model achieves **slightly better physics performance** and offers a new complementary functionality: **track-to-PV association**.



Latest ML developments

In/close to production stage

- Trigger
- Simulation

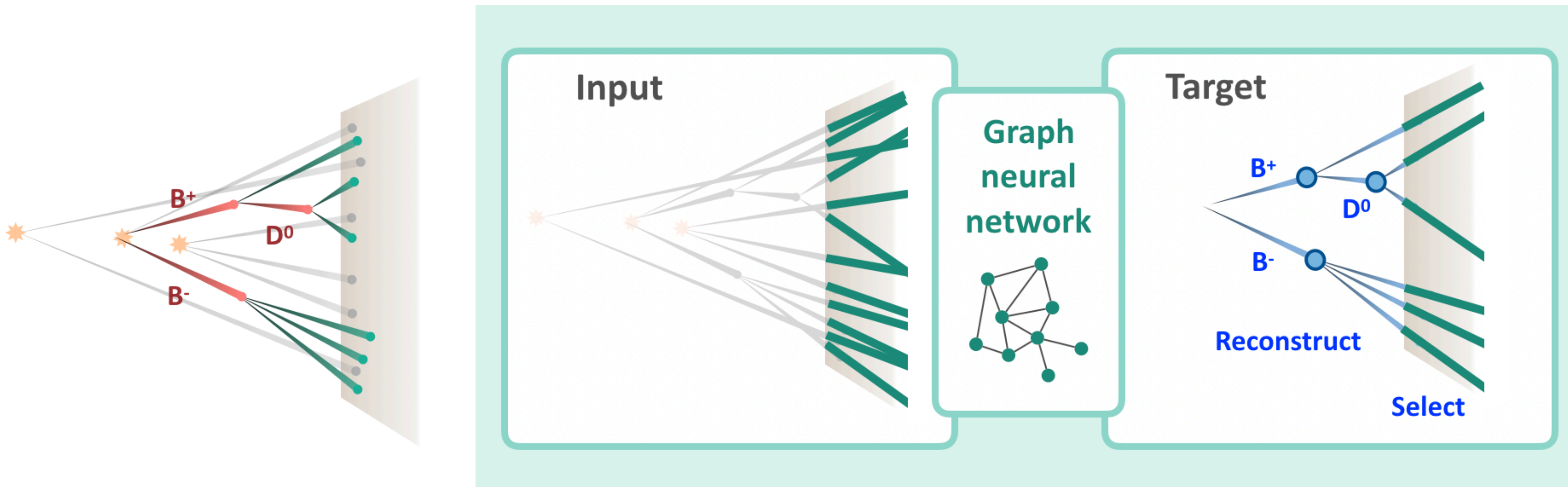
R&D stage

- VELO tracking
- PV finding
- **DFEI**
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

Deep-learning based Full Event Interpretation (DFEI)

[Comput Softw Big Sci 7, 12 (2023)]



One-go inclusive **multi-signal reconstruction + pileup suppression**, **targeting optimal event filtering**.

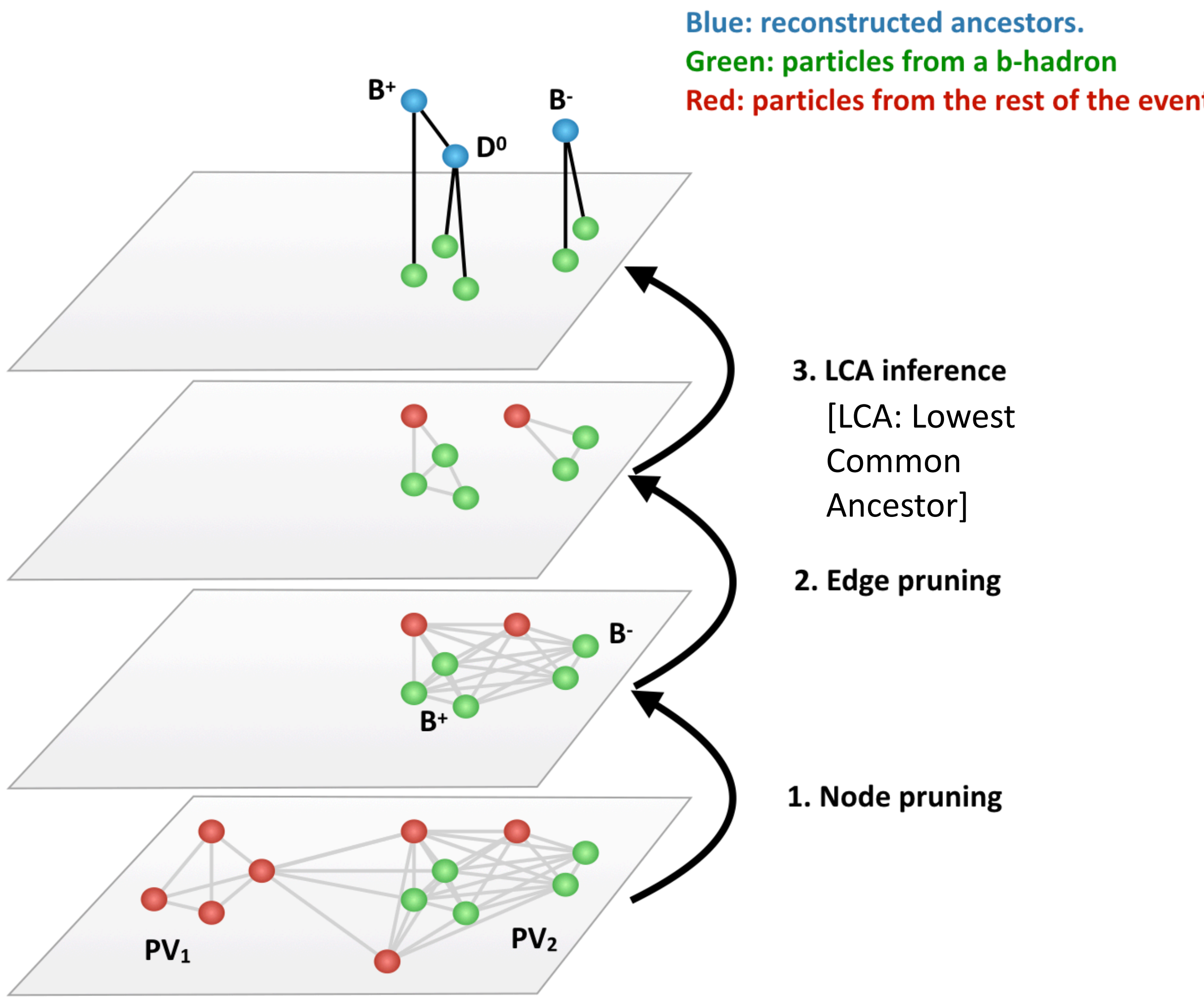
➥Alternative to current approach: OR between HLT2/Sprucing lines + selective persistency of other associated objects in the event.

Type of decay-chain reconstruction similar to that of the FEI algorithm at Belle II [Comput.Softw.Big Sci. 3 (2019) 1 6], but targeting the harsher LHC environment.

The algorithm

First prototype: [Comput Softw Big Sci 7, 12 (2023)]

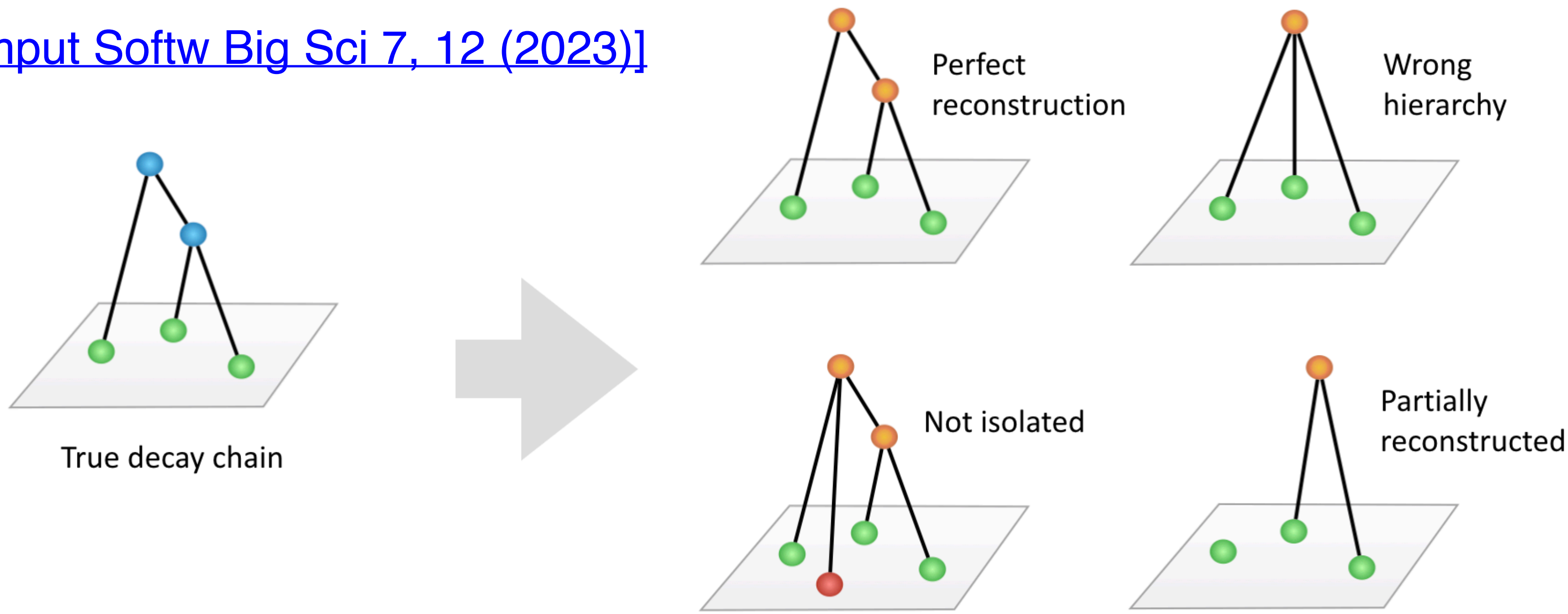
- Based on **three sequential GNN modules**.
- Restricted to b-hadron decays and charged stable particles.
- Only considers target ancestors which are “topologically” reconstructible(*).
- Trained on **custom simplified simulation** in Run3-like conditions.



(*) Target ancestors discarded if they are very short lived or don’t have enough charged descendants to form a vertex.

Performance: single-decay reconstruction

[Comput Softw Big Sci 7, 12 (2023)]

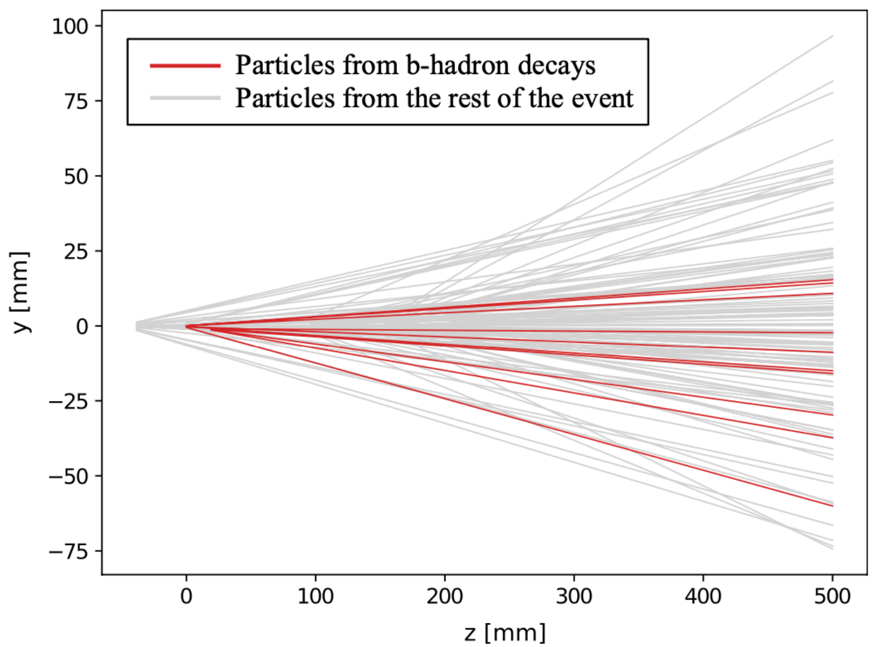


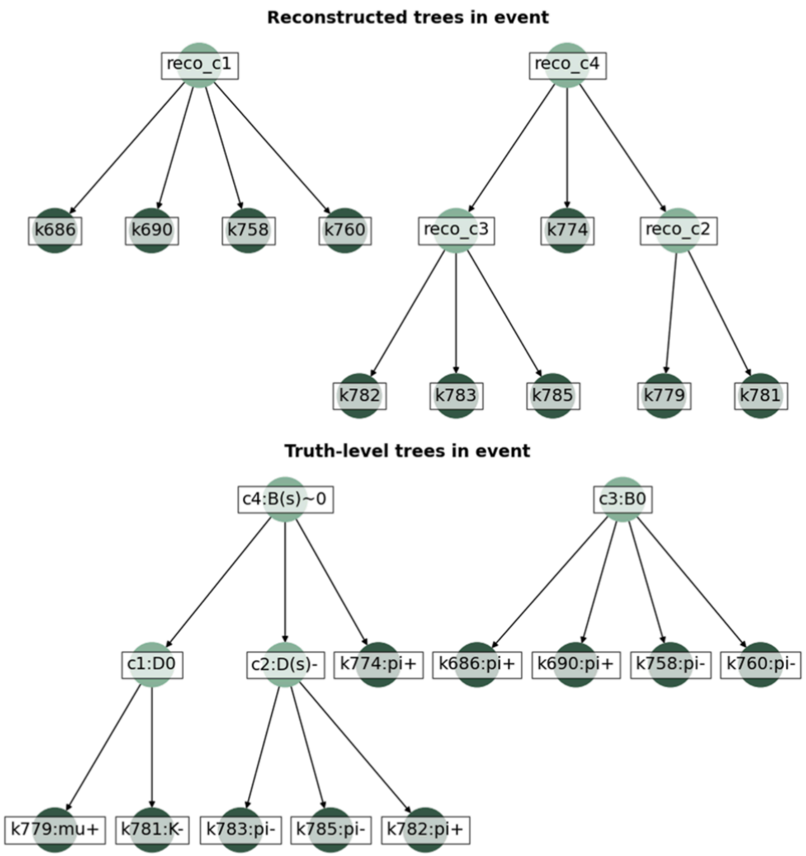
Decay mode	Perfect (%)	Wrong hierarchy (%)	Not iso. (%)	Part. reco. (%)
Inclusive H_b decay	4.6 ± 0.1	5.9 ± 0.1	76.0 ± 0.2	13.4 ± 0.1
$B^0 \to K_0^*[K\pi]\mu^+\mu^-$	35.8 ± 0.7	19.2 ± 0.6	44.9 ± 0.7	<0.02
$B^0 \to K^+\pi^-$	38.0 ± 0.7	–	54.7 ± 0.7	7.2 ± 0.4
$B_s^0 \to D_s^-[K^-K^+\pi^-]\,\pi^+$	32.8 ± 0.7	7.1 ± 0.4	53.7 ± 0.8	6.4 ± 0.4
$B^0 \to D^-[K^+\pi^-\pi^-]D^+[K^-\pi^+\pi^+]$	22.7 ± 0.6	22.4 ± 0.6	54.9 ± 0.8	<0.02
$B^+ \to K^+K^-\pi^+$	35.7 ± 0.7	10.2 ± 0.4	46.4 ± 0.7	7.7 ± 0.4
$\Lambda_b^0 \to \Lambda_c^+[pK^-\pi^+]\,\pi^-$	21.7 ± 1.0	8.9 ± 0.7	36.8 ± 1.2	32.6 ± 1.1
$B_s^0 \to J/\psi[\mu^+\mu^-]\,\phi[K^+K^-]$	26.9 ± 0.6	20.5 ± 0.5	52.5 ± 0.6	<0.02

Performance: multi-decay reconstruction

[Comput Softw Big Sci 7, 12 (2023)]

Fraction of **perfectly-reconstructed events** in inclusive b-hadron simulation (example below) in the ballpark of the tag-side efficiency for Belle (II) [Comput.Softw.Big Sci. 3 (2019) 1 6].

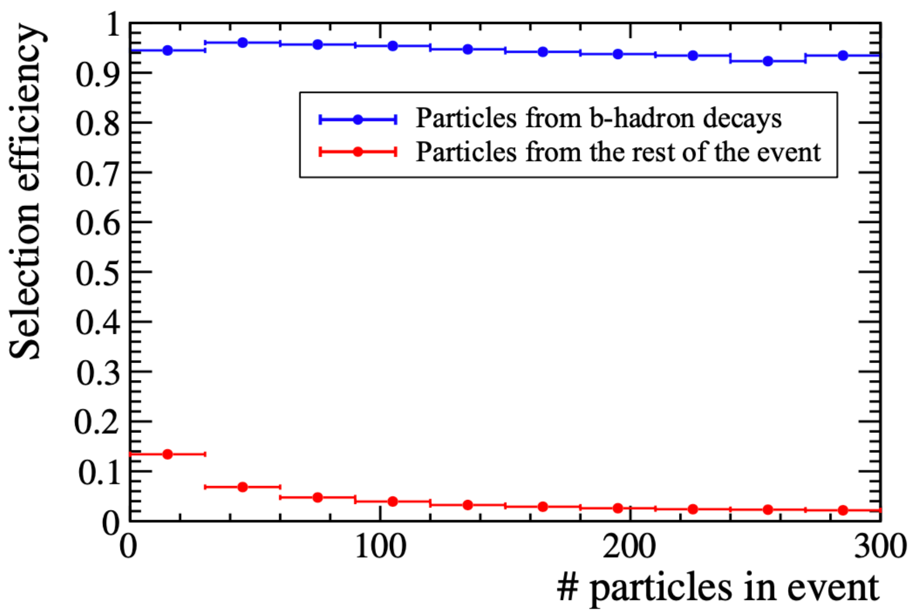


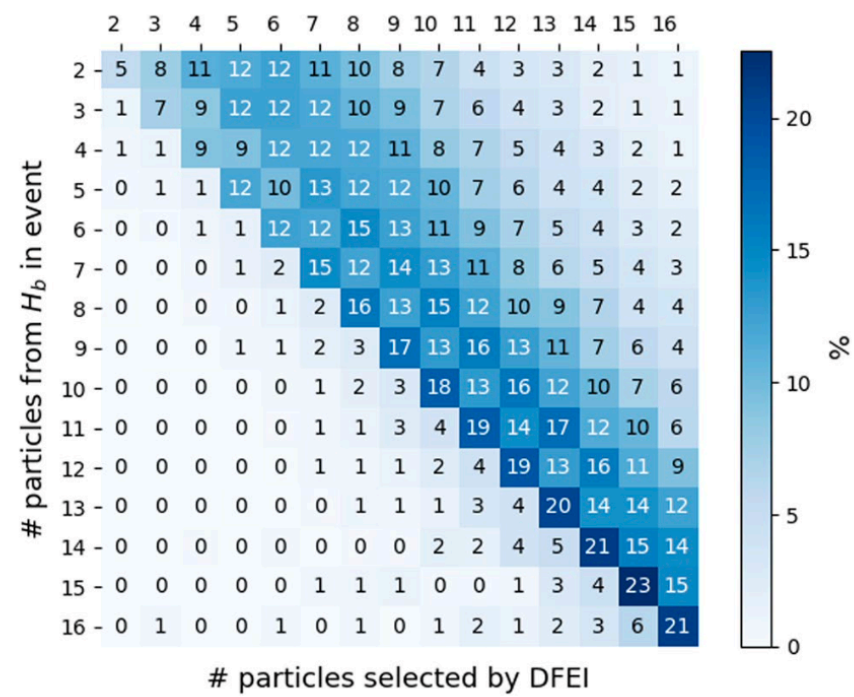


Performance: event filtering (pileup suppression)

[Comput Softw Big Sci 7, 12 (2023)]

Powerful event-filtering irrespectively of the particle multiplicity, as found in inclusive b-hadron simulation.





Recent improvements to model inference

First prototype:

- **Quadratic scaling of the inference time with the particle multiplicity**, dominated by the node-pruning GNN module.
- Overall evaluation time on the order of few seconds per event on CPU.
- **Inference pipeline on python with TensorFlow**.

Recent improvements to model inference

First prototype:

- **Quadratic scaling of the inference time with the particle multiplicity**, dominated by the node-pruning GNN module.
- Overall evaluation time on the order of few seconds per event on CPU.
- **Inference pipeline on python with TensorFlow**.

Modifications to the model (see details in this talk):

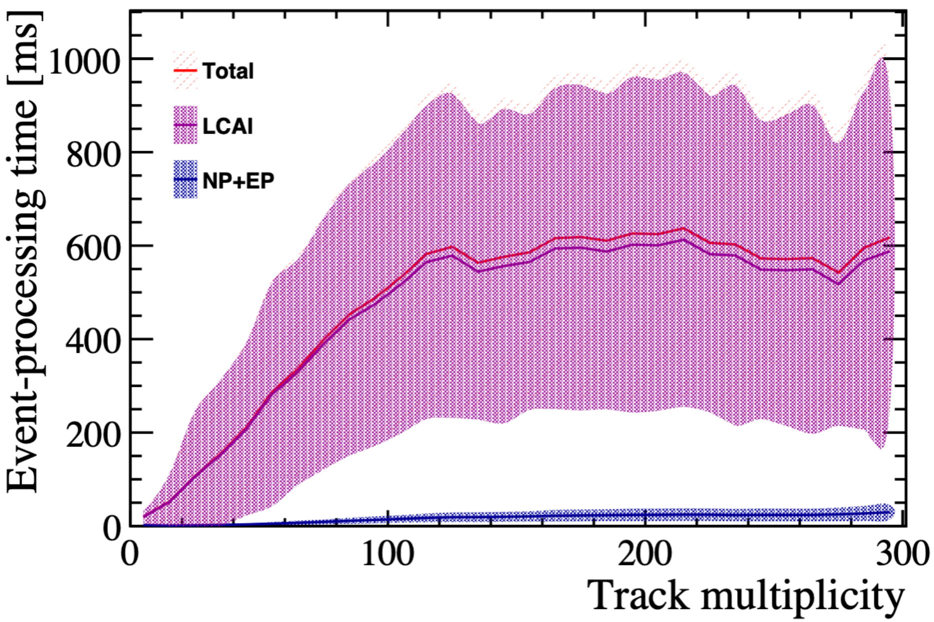
- **Model simplification**: substitution of the GNNs used for node- and edge-pruning by simpler classifiers (BDT).

- Implementation of the **full inference pipeline in C++**, with the LCAI GNN module converted **thanks to the recent additions in TMVA::SOFIE**.

Recent improvements to model inference

First prototype:

(See details in [this talk](#).)



CentOS Linux 7 (Core) x86
2.8 GHz Intel Core Processor

n the particle multiplicity,

con

lk):

ls used for node- and edge-

Sub-linear scaling achieved. Time now dominated by the LCAI algorithm. Significant **overall speed up** (final number pending on a hyper-parameter tuning of the LCAI).

- Implementation of the **full inference pipeline in C++**, with the LCAI GNN module converted thanks to the recent additions in [TMVA::SOFIE](#).

Present and future of the DFEI project

Ongoing developments:

- Studies for **applications in data analysis**.
- Expansions (include neutral stable particles, charm hadrons, …).
- Design improvements to the GNN model.

Next:

- **Implementation in the LHCb software stack**: targeting Sprucing for the near future and HLT2 in the long term.
- Detailed performance studies with simulation and with data.

Latest ML developments

In/close to production stage

- Trigger
- Simulation

R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

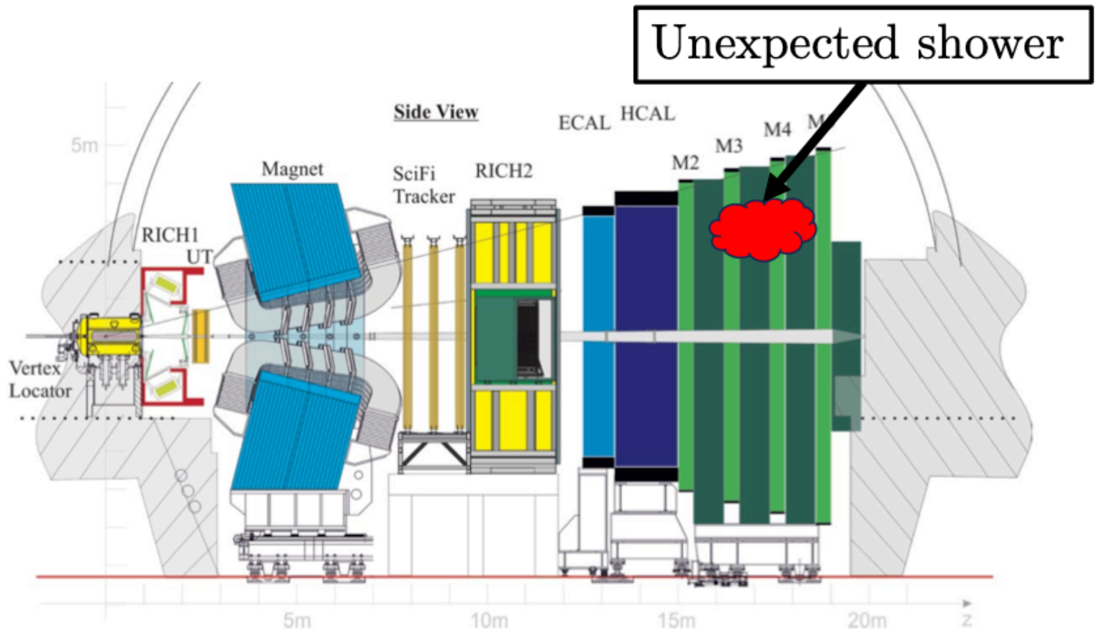
Anomaly detection in the muon system

[LHCb-FIGURE-2024-015]

Goal: **inclusive trigger line** to search for signatures of **Long Lived Particles** (LLPs).

Idea: use the **muon detector as a sampling calorimeter.**

- Very clean environment.
- Information of hit coordinates and multiplicities available.
  ➥ No energy-deposit measurements.



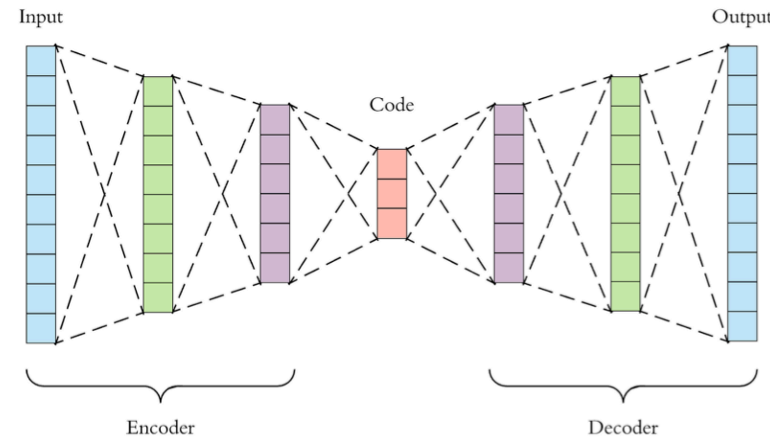
[LHCb Upgrade I]

Similar searches done by ATLAS [PRD 106 (2022) 3, 032005] and CMS [PRL 127 (2021) 26, 261804, arXiv:2402.18658].

Normalised autoencoders (NAE)

[arXiv:2105.05735]

Autoencoders (AEs) are trained to minimise the difference between the input and the reconstructed output.

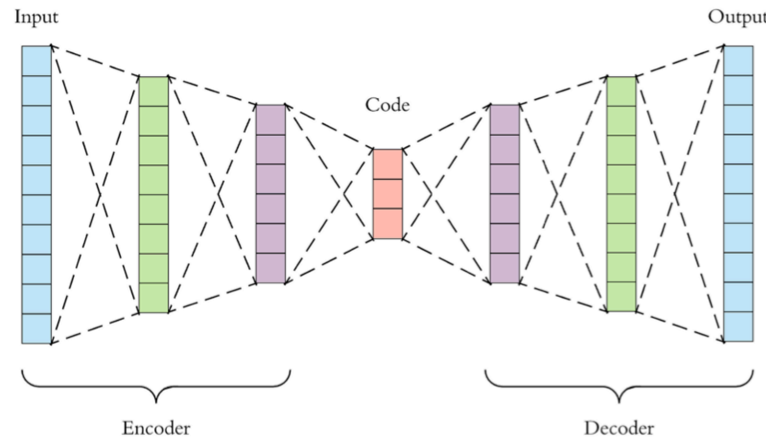


Application in anomaly detection:

- Train on normal (non-anomalous) data only.
- Evaluate on all data: expect low reconstruction error for normal data and large error for anomalous data.

Normalised autoencoders (NAE)

[arXiv:2105.05735]



Autoencoders (AEs) are trained to minimise the difference between the input and the reconstructed output.

Application in anomaly detection:

- Train on normal (non-anomalous) data only.
- Evaluate on all data: expect low reconstruction error for normal data and large error for anomalous data.

Possible drawback of **standard AE: the model can generalise "too well"** and also reconstruct anomalous data, preventing the discrimination.

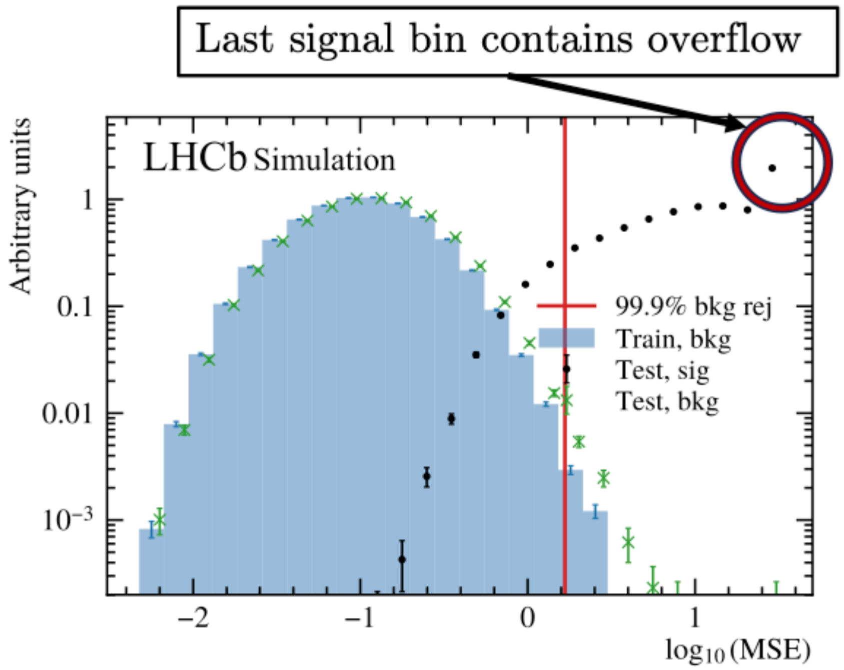
NAE: add a normalisation term to the loss function, estimated via MC sampling.

➥ **Good reconstruction if, and only if, data is normal.**

NAE for anomaly detection in the muon system

[LHCb-FIGURE-2024-015]

Studies: train the model on simulated minimum-bias events; evaluate the reconstruction error for those types of events and for specific simulated signals.



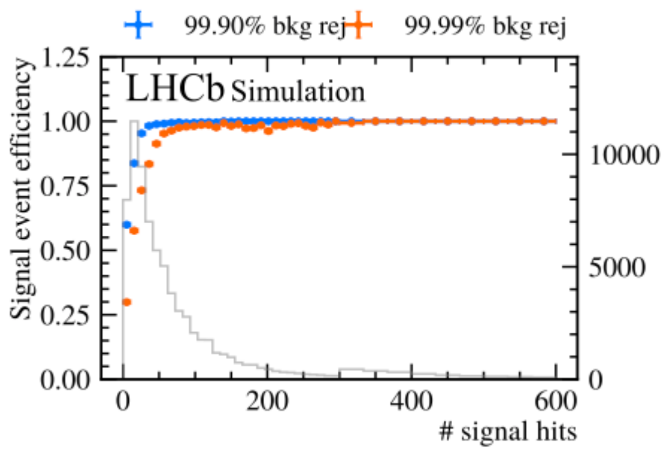
Axion sample as signal in the plot:

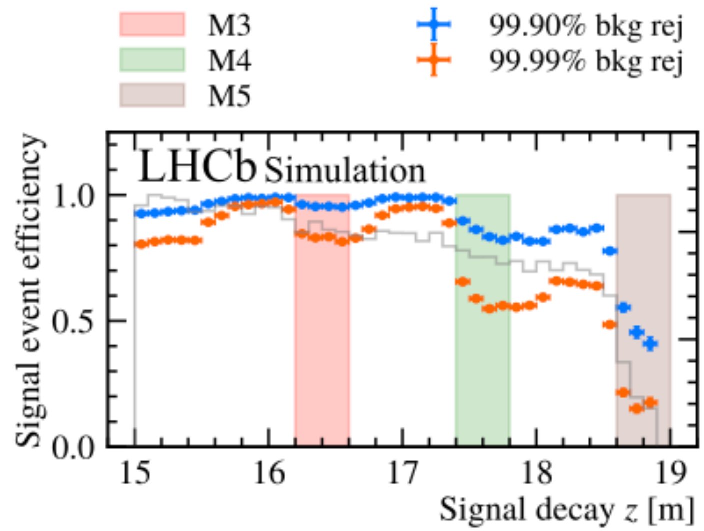
$H \to AA,\ A \to \tau^+\tau^-,\ \tau^\pm \to \pi^\pm\pi^\pm\pi^\mp\nu,$
$m_A = 10\ \mathrm{GeV},\ \tau_{\mathrm{axion}} = 1\ \mathrm{ns}$

Good separation found.

Important: for the eventual application, **the model can be trained on real data**.

Performance study [LHCb-FIGURE-2024-015]



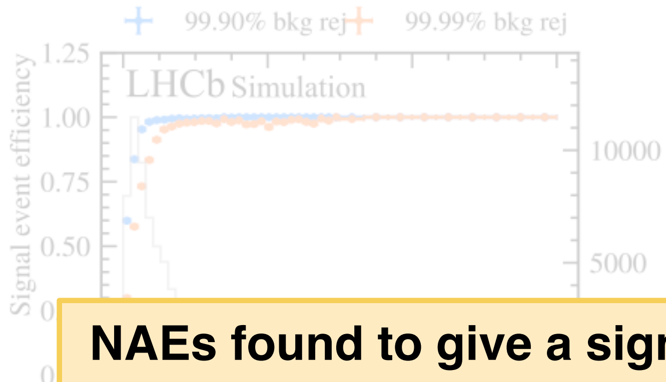


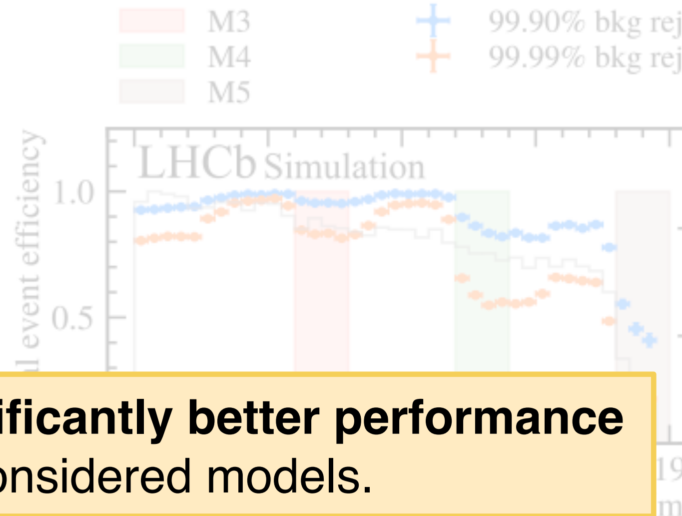
Comparison with other models in terms of number of parameters and signal efficiencies for different LLPs, for a 99.99% background rejection power:

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	$(80 \pm 0.5)\%$	$(10.3 \pm 0.3)\%$	$(15.7 \pm 0.3)\%$

Performance study

[LHCb-FIGURE-2024-015]





NAEs found to give a significantly better performance than the other considered models.

Comparison with other models in terms of number of parameters and signal efficiencies for diffe

Next: implement the algorithm in Allen.

ction power:

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	$(80 \pm 0.5)\%$	$(10.3 \pm 0.3)\%$	$(15.7 \pm 0.3)\%$

Latest ML developments

In/close to production stage

- Trigger
- Simulation

R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- **Neutral particle simulation**

Others

- Data Quality Monitoring

Neutral particles in LAMARR

(Studies covered in this talk.)

To extend the LAMARR simulation to photons and electrons, an **accurate simulation of the high-level ECAL response** is required.

Technical challenge:

number of generated particles ≠ number of reconstructed objects

(due to bremsstrahlung radiation, converted photons, and merged π^0)

Neutral particles in LAMARR

(Studies covered in this talk.)

To extend the LAMARR simulation to photons and electrons, an **accurate simulation of the high-level ECAL response** is required.

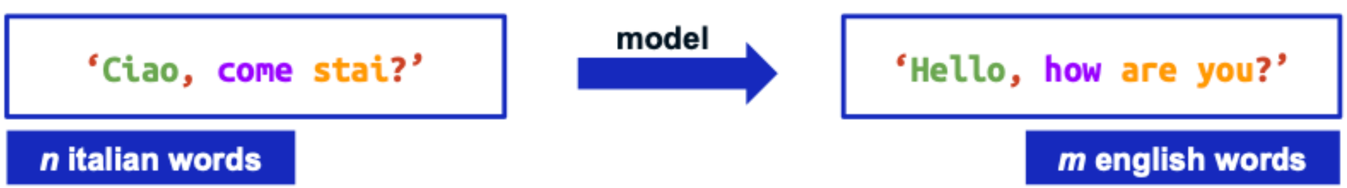
Technical challenge:

number of generated particles ≠ number of reconstructed objects

(due to bremsstrahlung radiation, converted photons, and merged π^0)

Two complementary approaches:

- **Signal photons** (produced in decay modes under study): one-to-one relation possible. → Similar treatment as for charged particles.
- **Secondary photons**: event-level description inspired by translation problems.

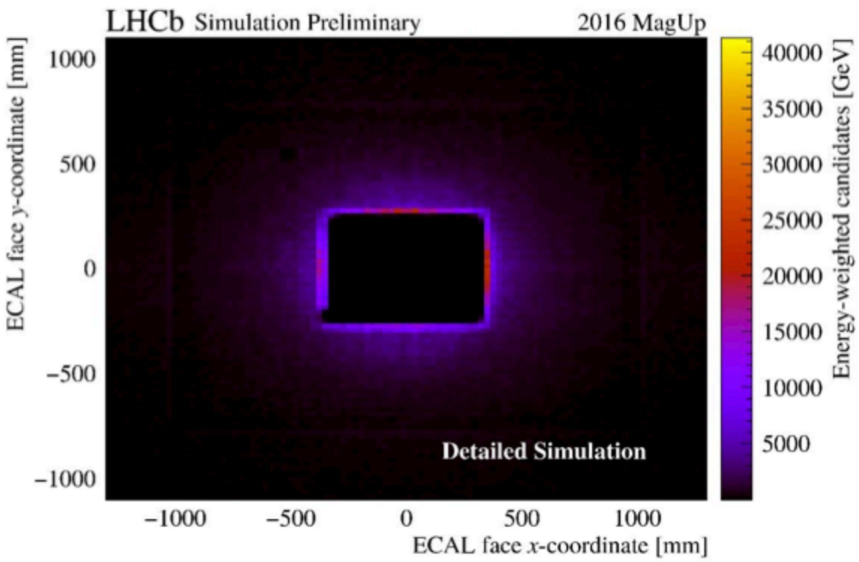


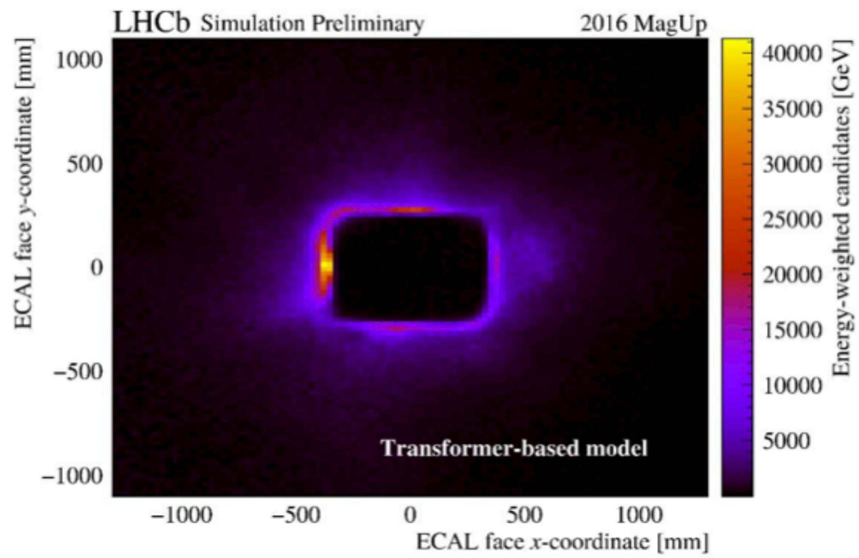
Treatment of secondary photons

(Studies covered in this talk.)

Two types of algorithms under study: **Transformers** (see below a generation example) and **GNNs**.

➥ Models **trained in an adversarial way, with DeepSets** as discriminators.





Next: further work to improve and compare the performance of both algorithms.

Latest ML developments

In/close to production stage

- Trigger
- Simulation

R&D stage

- VELO tracking
- PV finding
- DFEI
- Anomaly detection
- Neutral particle simulation
- Data Quality Monitoring

Others

DQM system

DQM goal: disentangle pp collision datasets in good conditions (OK) from those presenting **detector-induced anomalies** (BAD).

Two regimes:

- **Online:** datasets collected at fixed time intervals (10' in LHCb). Shifters inspect the data continuously, aiming to **identify anomalies as soon as possible** to get them fixed.

- **Offline:** datasets correspond to full runs, that have been previously collected. Shifters inspect the data with much looser time limitations, aiming at a **very accurate classification**.

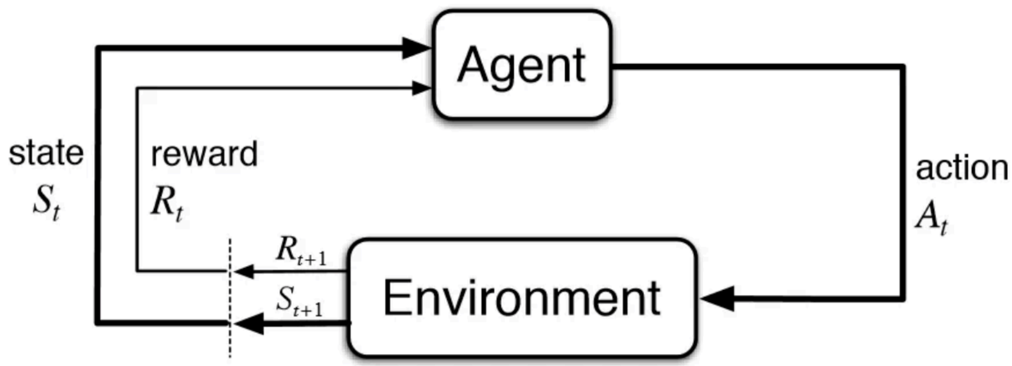
DQM system

Currently, the task is performed by rotating shifters:

- Limited classification **accuracy**.
- High cost in terms of **person power**.
- Challenging adaptation to **changes in operational conditions**, which requires frequent update of histogram references by detector experts.

Reinforcement Learning from Human Feedback (RLHF)

RL: a ML "agent" interacts with the environment, performing actions and receiving rewards for them. The agent is trained to behave in a way that maximises the reward expected to be received in the long term.



RLHF: the rewards are derived from human decisions.

RL techniques are used at CERN for example for **accelerator control** (see e.g. [this recent talk]).

RLHF for DQM

Proof-of-concept studies in [arXiv:2405.15508], presented in the following.

➥ **First application of RLHF for DQM at HEP experiments.**

➥ The approach is **experiment independent**.

RLHF for DQM

Proof-of-concept studies in [arXiv:2405.15508], presented in the following.

➥ **First application of RLHF for DQM at HEP experiments.**

➥ The approach is **experiment independent**.

Why RL?

- **Capture trends** by **training continuously** during data taking.
- Allow the possibility to **globally optimise multiple correlated tasks**, partially involving human actors.
  - ➥ **Balance data collection efficiency vs operational costs**.

RLHF for DQM

Proof-of-concept studies in [arXiv:2405.15508], presented in the following.

➥ **First application of RLHF for DQM at HEP experiments.**

➥ The approach is **experiment independent**.

Why RL?

- **Capture trends** by **training continuously** during data taking.
- Allow the possibility to **globally optimise multiple correlated tasks**, partially involving human actors.
  - ➥ **Balance data collection efficiency vs operational costs**.

Foreseeable challenges?

- **Training/evaluation time?** Not a technical limitation at the current level of knowledge.
- **Data scarcity?** Approach: **data-augmentation techniques.**
- **Degradation of algorithm's response** by absorbing human mistakes? Approach: produce evolving reference templates that experts can check.

Proof-of-concept (POC) studies

[arXiv:2405.15508]

Synthetic data: 1D histograms generated in Nominal or Anomalous conditions, ordered sequentially. Distributions can change at a certain point in time.

RL algorithm: PPO actor-critic [arXiv:1707.06347].

Proof-of-concept (POC) studies

[arXiv:2405.15508]

Synthetic data: 1D histograms generated in Nominal or Anomalous conditions, ordered sequentially. Distributions can change at a certain point in time.

RL algorithm: PPO actor-critic [arXiv:1707.06347].

RL environment for Offline regime:

- Goal: maximise accuracy.
- **One RL agent,** that classifies a histogram as good or bad.
- Trained with labels from the shifter, available for every histogram.

Proof-of-concept (POC) studies

[arXiv:2405.15508]

Synthetic data: 1D histograms generated in Nominal or Anomalous conditions, ordered sequentially. Distributions can change at a certain point in time.

RL algorithm: PPO actor-critic [arXiv:1707.06347].

RL environment for Offline regime:

- Goal: maximise accuracy.
- **One RL agent,** that classifies a histogram as good or bad.
- Trained with labels from the shifter, available for every histogram.

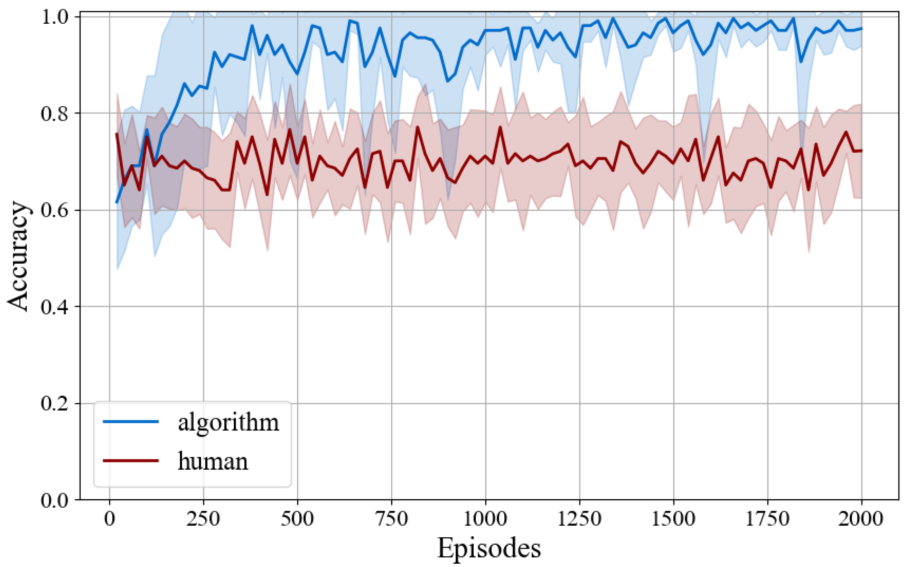
RL environment for Online regime:

- Goal: balance classification accuracy with the need for human intervention.
- **Two RL agents**, one that classifies and one that calls the shifter when needed.
- Additionally, **concept of problem fixing** introduced in the dataset generation.

Studies in the Offline regime: accuracy improvement

[arXiv:2405.15508]

Target label shaped in 30% of the cases during training, to emulate human mistakes.



The algorithm learns how to filter away this noise and achieve a **higher accuracy than the shifter**.

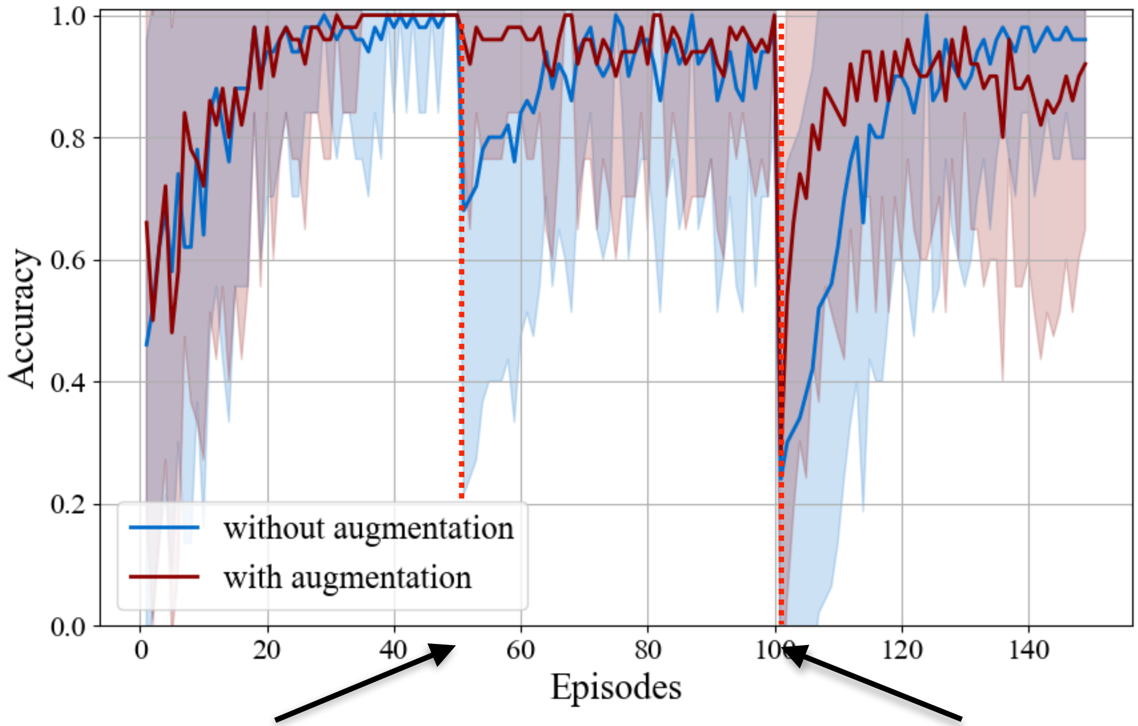
(This is a typical behaviour in neural networks, that happens only if the noise distribution is flat in the phase space.)

The conclusion still holds if shifters can see the outcome of the algorithm before making their decision and get partially influenced by it (see backup).

Studies in the Offline regime: data augmentation (DA)

[arXiv:2405.15508]

Insert artificial data points, generated using evolving references for nominal histograms and predefined (generic) types of anomalies.



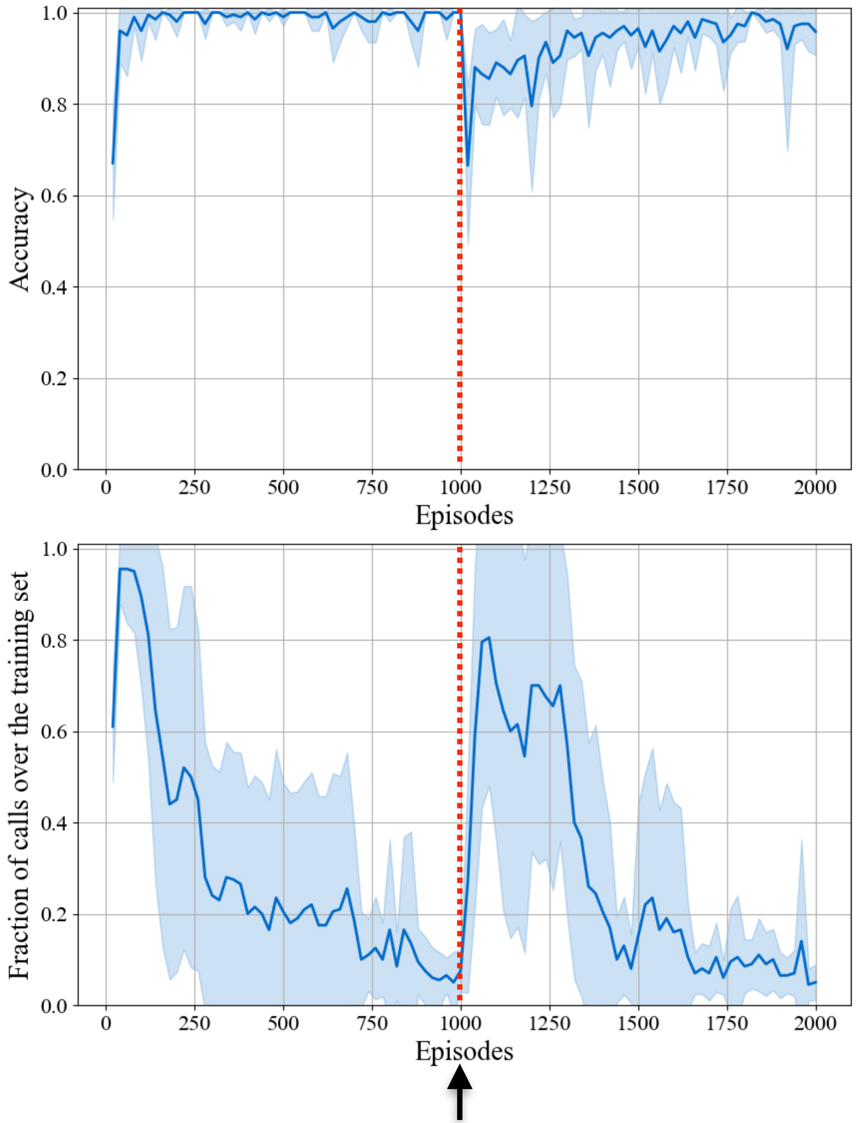
Change in type of anomalies

DA improves generalisation

Change in nominal conditions

DA speeds up adaptation

Studies in the Online regime: accuracy vs workload



Change in nominal conditions

[arXiv:2405.15508]

The algorithm achieves a **high accuracy with a limited number of calls to the shifter**, which are focused on the most relevant moments.

Conclusion of the POC studies

[arXiv:2405.15508]

Promising results from the proof-of-concept studies for both the Online and Offline regimes, **in terms of accuracy and level of automation**.

Next: do studies on LHCb data.

Since the approach is experiment independent, it could be applied to **other experiments** (some people in CMS and ALICE already manifested potential interest).

Latest ML developments

In/close to production stage

Trigger

Simulation

R&D stage

VELO tracking

PV finding

DFEI

Anomaly detection

Neutral particle simulation

Others

Data Quality Monitoring

Other ML developments

Inclusive Flavour Tagging with DeepSets

[arXiv:2404.14145]

Approach: consider all tracks in the event instead of subsets of them.

- **Improved physics performance** over classical taggers.
- **Very fast training and inference**.

Other ML developments

Inclusive Flavour Tagging with DeepSets

[arXiv:2404.14145]

Approach: consider all tracks in the event instead of subsets of them.

- **Improved physics performance** over classical taggers.
- **Very fast training and inference**.

Robust Neural Networks for particle identification

[arXiv:2212.07274]

Goal: reduce biases due to the specific decay samples used for training.
Approach: disentangle common and decay-specific components in the input.

- **Improved physics performance** compared to conventional algorithms.

Opportunities & challenges in Run 3 and beyond

Latest ML developments

Take-home messages

Take-home messages

★ LHCb is currently in an optimal spot for the development, deployment and usage of ML techniques: **new software & hardware opportunities + big-data challenges**.

Take-home messages

★ LHCb is currently in an optimal spot for the development, deployment and usage of ML techniques: **new software & hardware opportunities + big-data challenges**.

★ Increasing focus on **long-term maintainability** of ML solutions and development of **common pipelines**.

Take-home messages

★ LHCb is currently in an optimal spot for the development, deployment and usage of ML techniques: **new software & hardware opportunities + big-data challenges**.

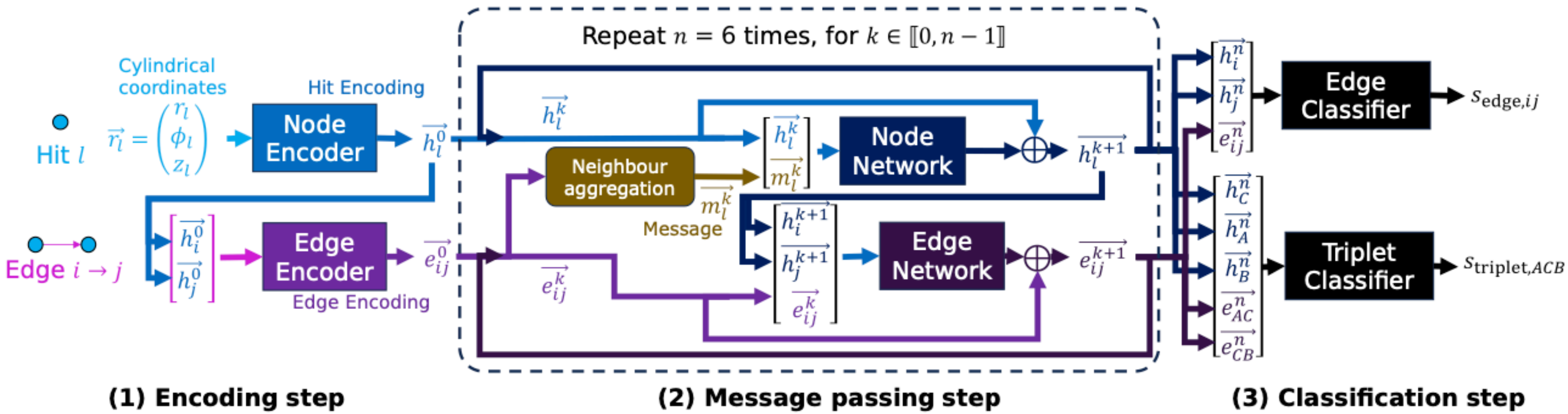
★ Increasing focus on **long-term maintainability** of ML solutions and development of **common pipelines**.

★ At the same time, many ongoing R&D efforts for the present and future of the experiment, that make use of **state-of-the-art algorithms** and in several cases constitute **pioneering applications** at the LHC experiments.

Backup slides

GNN model for ETX4VELO project

[arXiv:2406.12869]



Offline DQM: environment

[arXiv:2405.15508]

States	- The **histograms are fully independent** from each other. - Fixed probability to generate BAD histograms. - **Time dependency only through type of generation distributions**.
Episodes/steps	Episodes made out of a single step.
Agents	One single agent (neural network).
Actions	One decision: **label as OK or BAD**.
Rewards	- The "shifter" provides (target) OK/BAD labels for every histogram. - Reward: **+1(-1) if correctly (incorrectly) classified**.

Online DQM: environment

[arXiv:2405.15508]

States

- The histograms depend on each other (**concept of "problem fixing"**).

➥ If the current histogram is BAD, the next one will also be BAD, unless the algorithm labels it as BAD.
➥ If the current histogram is OK, or if the algorithm has labelled it as BAD, the next histogram will be BAD with a fixed probability (as in the offline case).

Online DQM: environment

[arXiv:2405.15508]

States

- The histograms depend on each other (**concept of "problem fixing"**).
- **Target labels OK/BAD only available when the shifter looks at the data**, that happens in two cases.

➥ If requested by the algorithm.
➥ Randomly, with a fixed probability ("checkpoint").

Online DQM: environment

[arXiv:2405.15508]

States

- The histograms depend on each other (**concept of "problem fixing"**).
- **Target labels OK/BAD only available when the shifter looks at the data**, that happens in two cases.

Episodes/steps

Episodes made out of a variable number of steps, separated by two consecutive "checkpoints".

Agents

- **Two agents**, one to classify (***predictor***) and one to call the shifter (***checker***), acting one after the other.
- The *checker* can see the output of the *predictor*.

Actions

One decision per agent: **label as OK or BAD; call or not the shifter**.

Rewards

Separate reward per agent:

- *Predictor*: same reward as in the Offline case, but only when the shifter labels are available.
- *Checker*: **reward derived from the *predictor*'s "confidence" on its decision, mildly penalising unnecessary calls** (see next slide).

Reward schemes for the Online-DQM agents

[arXiv:2405.15508]

Predictor If the shifter's label is available in the current state, the reward is $+1$ if the predictor's decision matches the human label, and -1 otherwise. If the shifter's label is not available, the predictor's reward is zero.

Checker If the checker did not request a check, the reward is zero. If the checker requested a check, the reward is $\omega - p$, where ω is for the predictor's *mis-tagging "probability"* (defined below) and $p \in (0, 1)$ is a hyperparameter (we set it to 0.1) that regulates the amount of penalisation given to the checker for calling the shifter "unnecessarily", i.e. when the predictor is doing well. The ω variable is a proxy of the probability that the predictor outputs the wrong label, using its own output. It is constructed using the two logits $(lp_{\mathrm{n}}, lp_{\mathrm{a}})$ in the predictor's output, for nominal and anomalous predictions respectively. We compute "probabilities" by passing this vector through a softmax layer, then define ω as the "probability" associated to the outcome that was not chosen by the shifter, i.e. $\omega = p_{\mathrm{a}}$ if the shifter's label is "nominal", and $\omega = p_{\mathrm{n}}$ otherwise.

Offline DQM: human-machine interaction

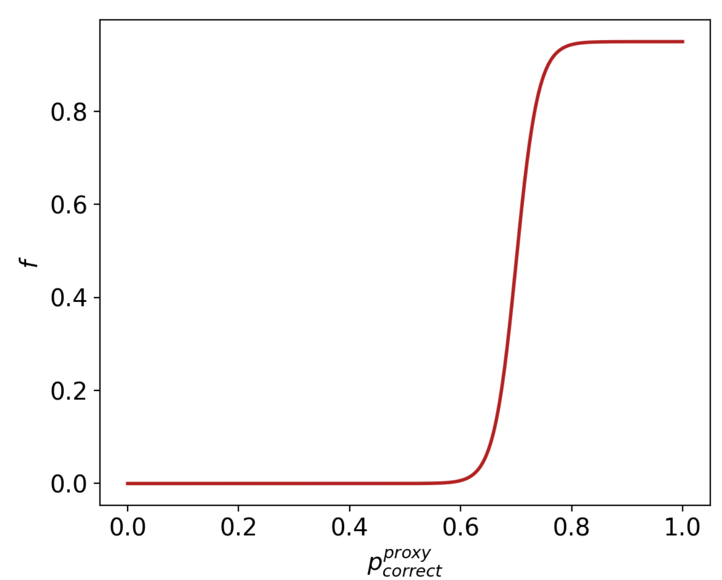
Let's assume the same situation as before, but **now the shifter can see the algorithm's prediction before making the decision** and get (partially) influenced by it.

➥ Does this prevent the algorithm from reaching "superhuman" performance?

[arXiv:2405.15508]

We consider the following setup:

1. **The shifter can see a proxy probability for the algorithm to be correct** (in this case computed from its output logits).
2. **The shifter randomly "trusts" the algorithm in a fraction of cases** that has a dependency on that probability.
3. When the shifter trusts the algorithm, their decision is replaced by the one of the algorithm.
4. **The algorithm is trained using those a-posteriori shifter decisions**.



Offline DQM: human-machine interaction

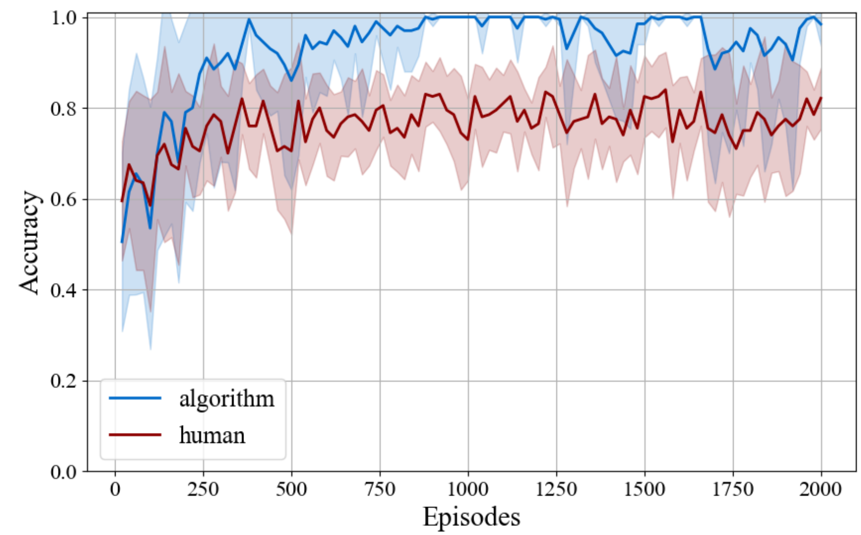
Let’s assume the same situation as before, but **now the shifter can see the algorithm’s prediction before making the decision** and get (partially) influenced by it.

➥ Does this prevent the algorithm from reaching “superhuman” performance?

[arXiv:2405.15508]

We consider the following setup:

1. **The shifter can see a proxy probability for the algorithm to be correct** (in this case computed from its output logits).
2. **The shifter randomly “trusts” the algorithm in a fraction of cases** that has a dependency on that probability.
3. When the shifter trusts the algorithm, their decision is replaced by the one of the algorithm.
4. **The algorithm is trained using those a-posteriori shifter decisions**.



Both the algorithm and the shifter perform better than the baseline case.