Machine learning and advances in data quality monitoring

Suzanne Klaver, on behalf of the LHCb collaboration;
including material from the ALICE, ATLAS and CMS collaborations

9th Edition of the Large Hadron Collider Physics Conference

Virtual Paris, 9 June 2021

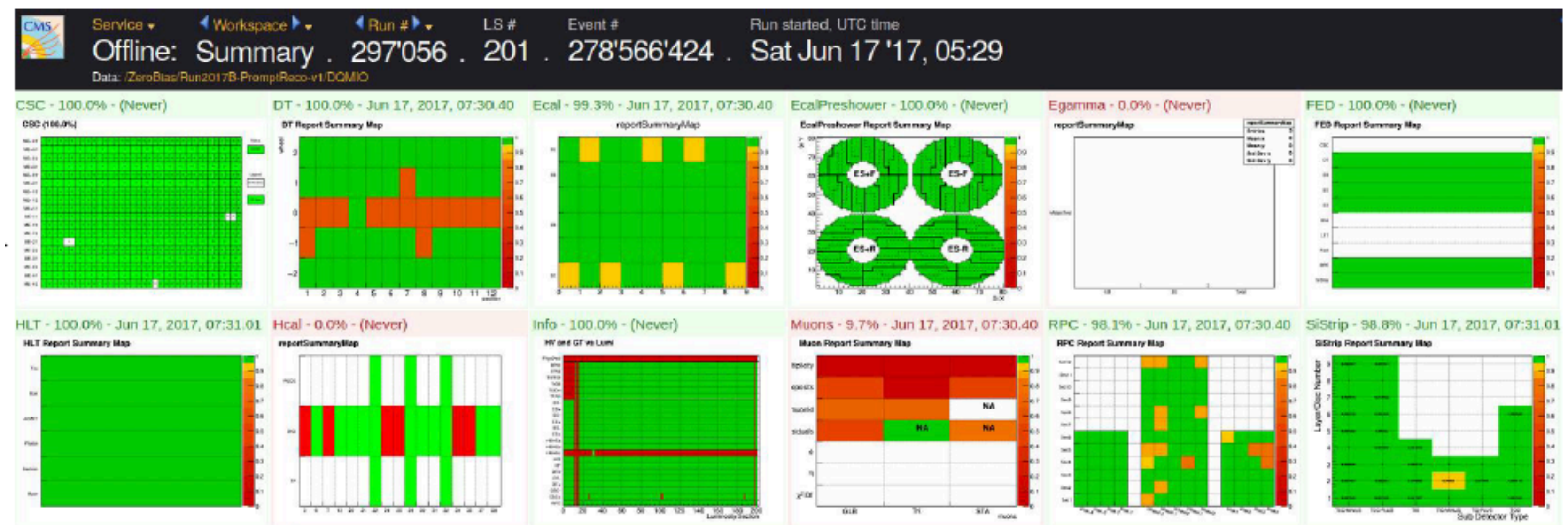
Introduction

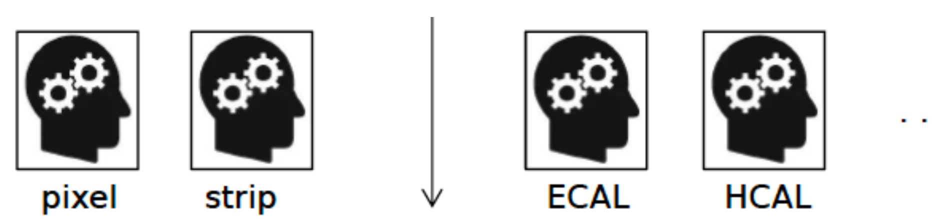
- All LHC experiments are preparing for Run 3
- Data quality monitoring crucial for ensuring correct data taking
  - typically, data flagged by person as GOOD/BAD
- Can we make improvements in data quality monitoring?
  - use machine learning (ML)?
  - many ongoing studies
  - preliminary results
- Advantages ML:
  - less people needed,
  - so less prone typical human errors (e.g. fatigue, lack of focus, dependence on clear visualisation, ...).

General strategy DQ monitoring

- Many subsystems with their own type of histograms and issues
- Afterwards combined to a single flag from all subsystems as GOOD/BAD

example from CMS:

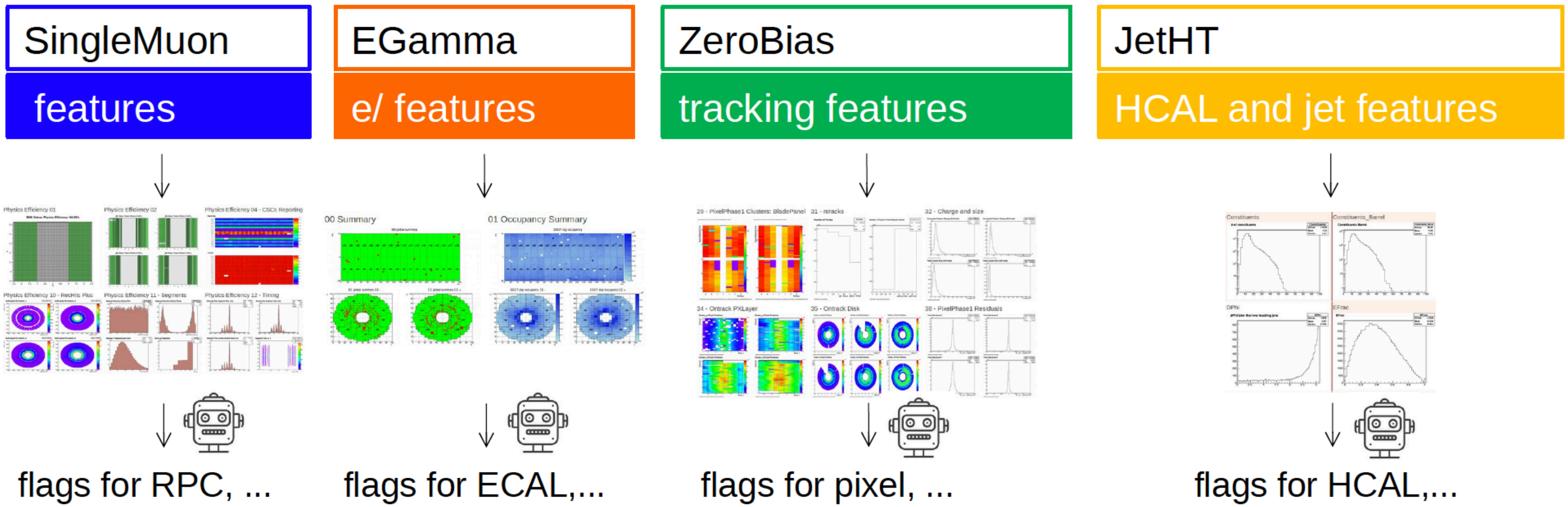




Run Number	btag	castor	cms	csc	ctpps	lowlumi	dt	ecal	es	egamma	hcal	hlt
297046	GOOD	EXCLUDED	GOOD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
297047	GOOD	EXCLUDED	GOOD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
297048	GOOD	EXCLUDED	BAD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
297049	GOOD	EXCLUDED	BAD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
297050	GOOD	EXCLUDED	GOOD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	B G
297056	GOOD	EXCLUDED	GOOD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
297057	GOOD	EXCLUDED	GOOD	GOOD	BAD	BAD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD

ML strategy DQ monitoring

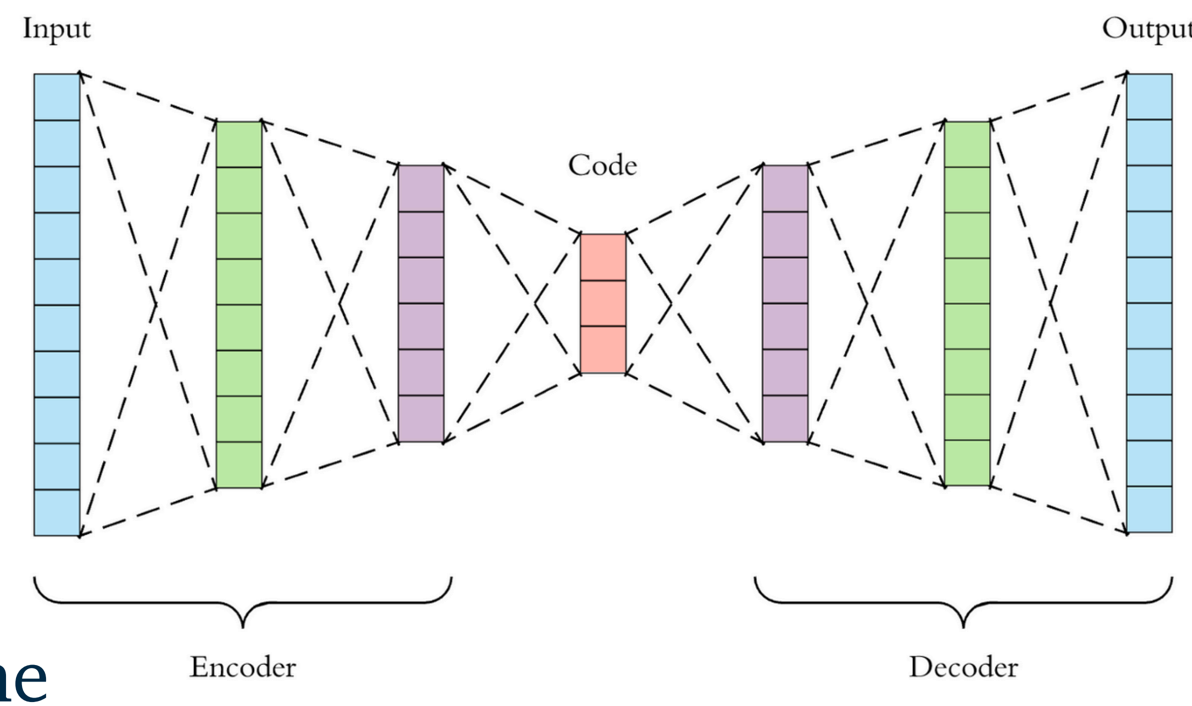
- Many subsystems with their own type of histograms and issues
- Each subsystem finds their own optimal ML method to use



- Afterwards combined to a single flag from all subsystems as GOOD/BAD (similar to human-based approach)
- Many successful approaches, focus on auto encoders

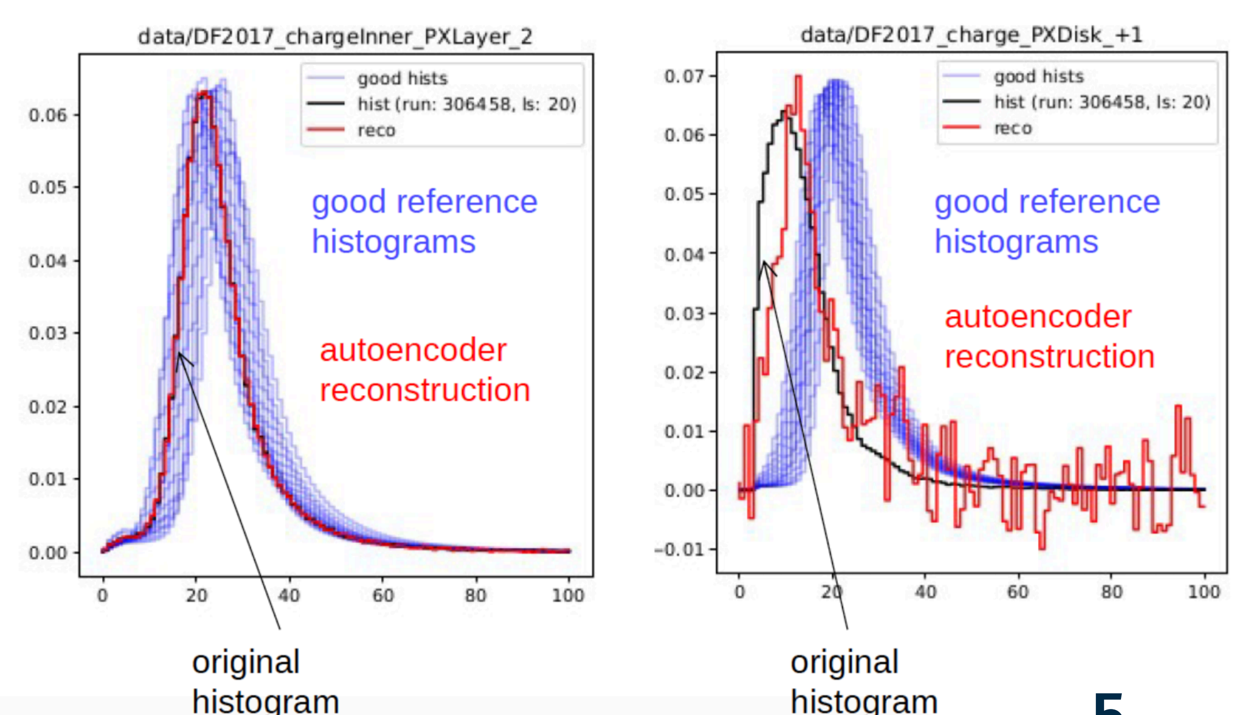
Auto encoders for 1D histograms

- Type of neural network, same number input and output nodes, but smaller in between
- Assume complex data has simpler underlying structure



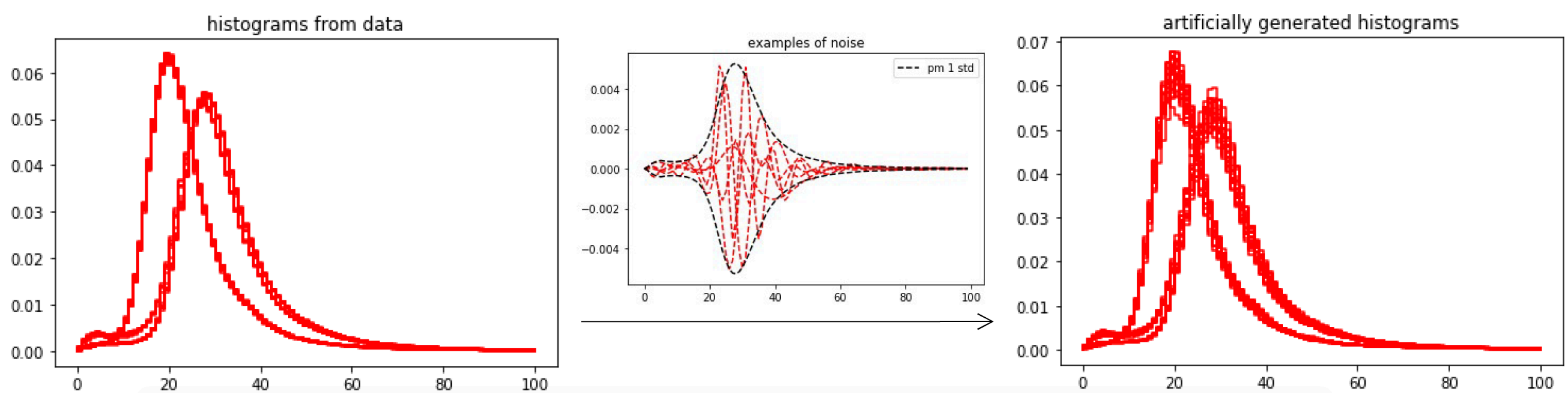
- Most histograms are GOOD data; the BAD ones are anomalous
- Train auto encoder on large set of histograms to learn generic features
  - also those dependent on pile-up etc!
- GOOD histograms similar between original and reconstructed
- BAD ones have a large difference

example:



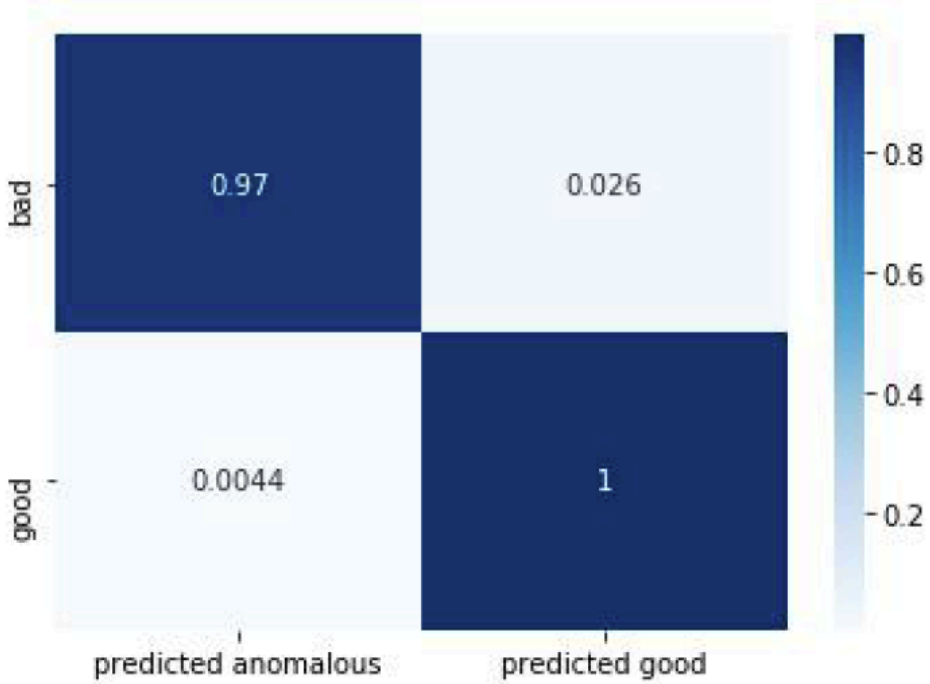
Resampling

- Auto encoders are trained on 2017 data sample
  - little bad data + lack of reliable labeled training/testing data
- Label small set of data by hand
  - resample to make similar (non-identical) histograms (seeds) and add some noise
    - advantage: not limited by statistics
    - disadvantage: choice of seeds may be biased



Preliminary results

- Autoencoder approach flags most of anomalous data correctly, with low fake rate!
  - even pointed to anomalous-looking histograms flagged as GOOD by people
- Aim to run first tests at start of Run 3, in parallel with human-based DQM and data certification

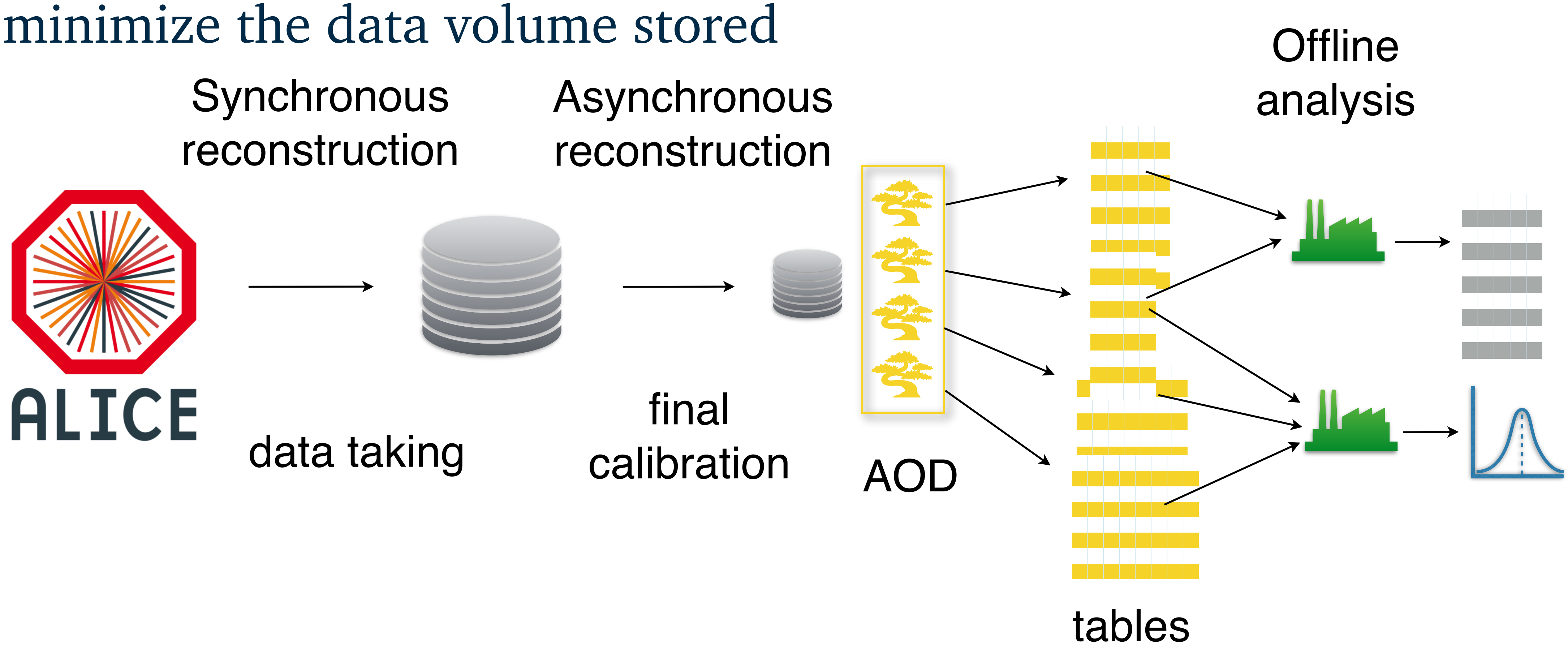


DQ monitoring in ATLAS

- ATLAS is also envisaging using ML for data monitoring
  - no public results yet
- working on:
  - looking for anomalies in blocks of data to alert shifters
  - use machine learning to predict reference plots that can vary on data-taking conditions, like pile up

ALICE reconstruction strategy in Run 3

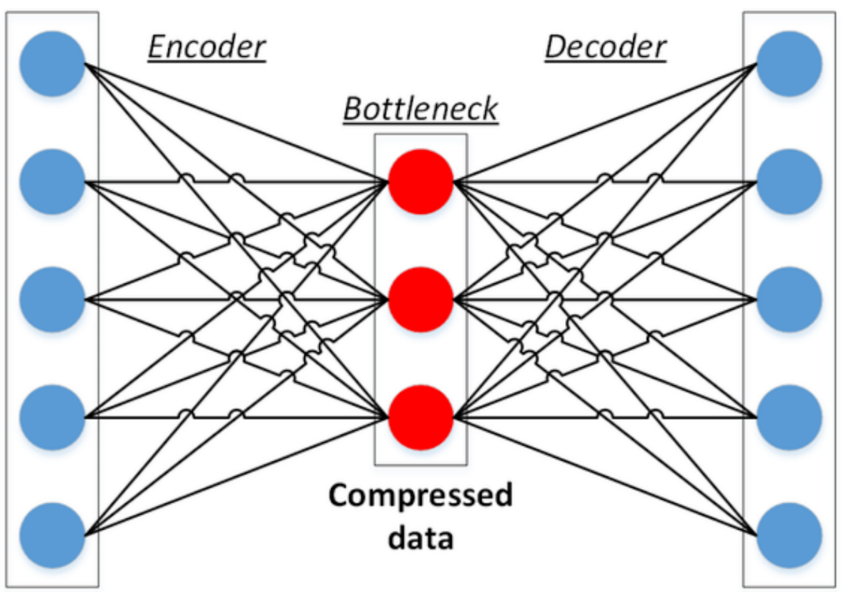
- ALICE will run in Run 3 in continuous data taking mode, without triggers
- Event reconstruction while collecting data:
  - minimize the data volume stored



- Unique framework for both DQM and Quality Assurance
- No usual distinction between online/offline reconstruction
  - **not possible to redo event track/reconstruction!**

Autoencoders for Quality Control

- **Goal**: Identify reconstruction failures on chunk of data of a given length



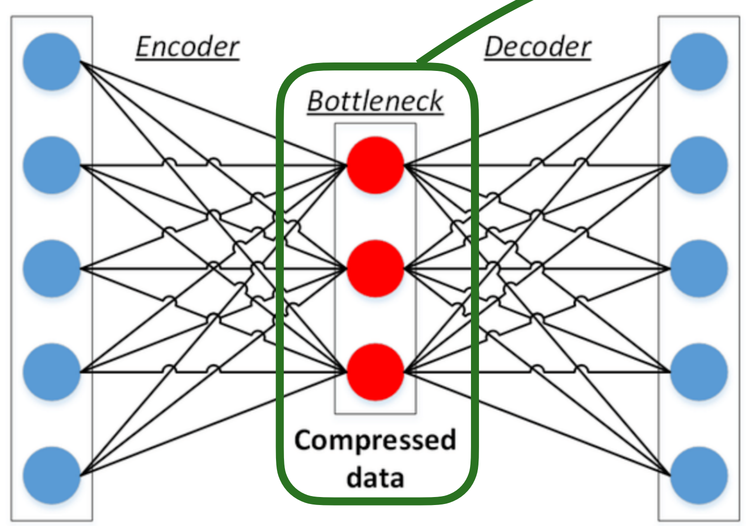
- **Algorithm**: autoencoders (AE) and variational AE (VAE) in semi-supervised approach
- **Input data**: ~200 detector-level and reconstruction-level quantities:
  - number of TPC clusters/track for low and high p_{T} tracks
  - vertex position
  - gas parameters
  - drift velocity
  - …

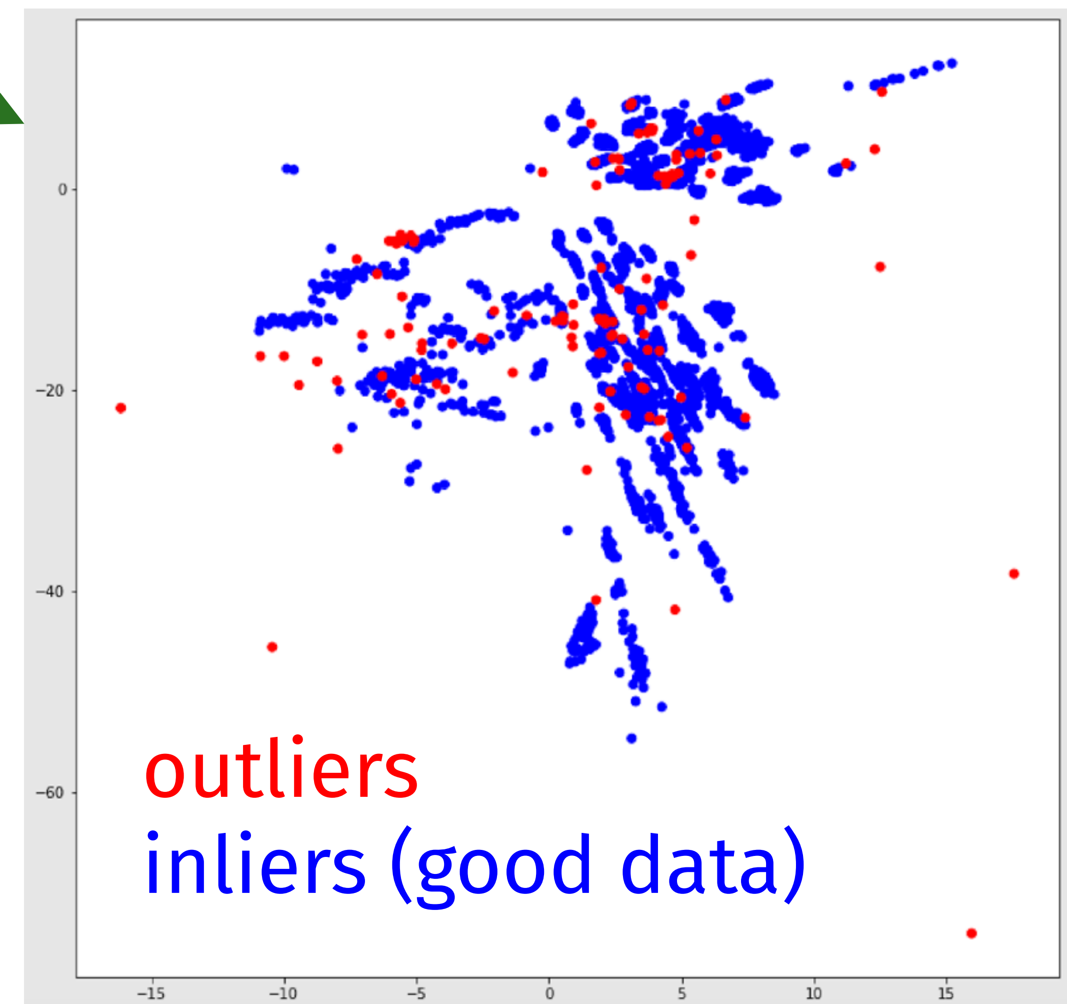
- AE and VAE trained on dataset previously tagged as GOOD or BAD by comparing to references

Some preliminary results

- Autoencoder with only 2 hidden layers

AE hidden layer



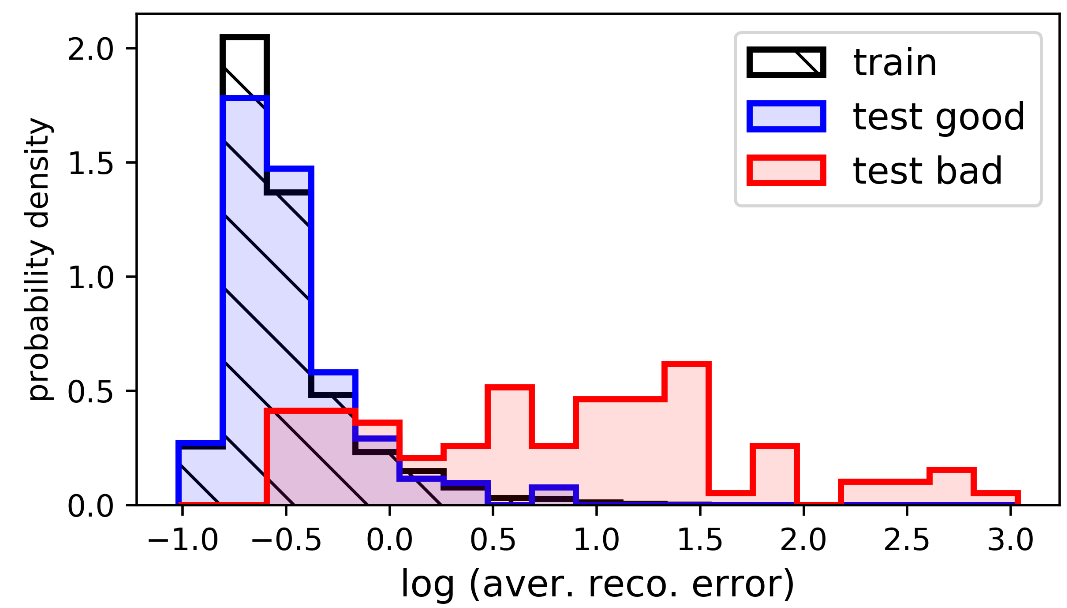


- The autoencoder maps 200 input paramaters into the plane in a way that is *meaningful enough* to reproduce the input in the decoder part

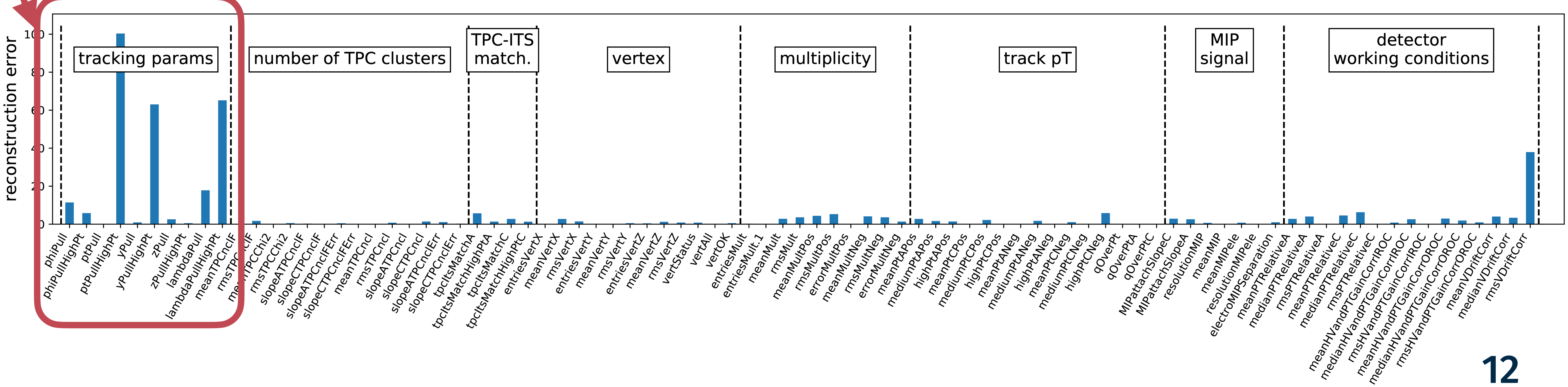
→ proximity between clusters denotes similarity and potentially similar quality issues

Some preliminary results II

- Reconstruction error = difference between output and input of the AE
- Large reconstruction error averaged over parameters suggest anomalies

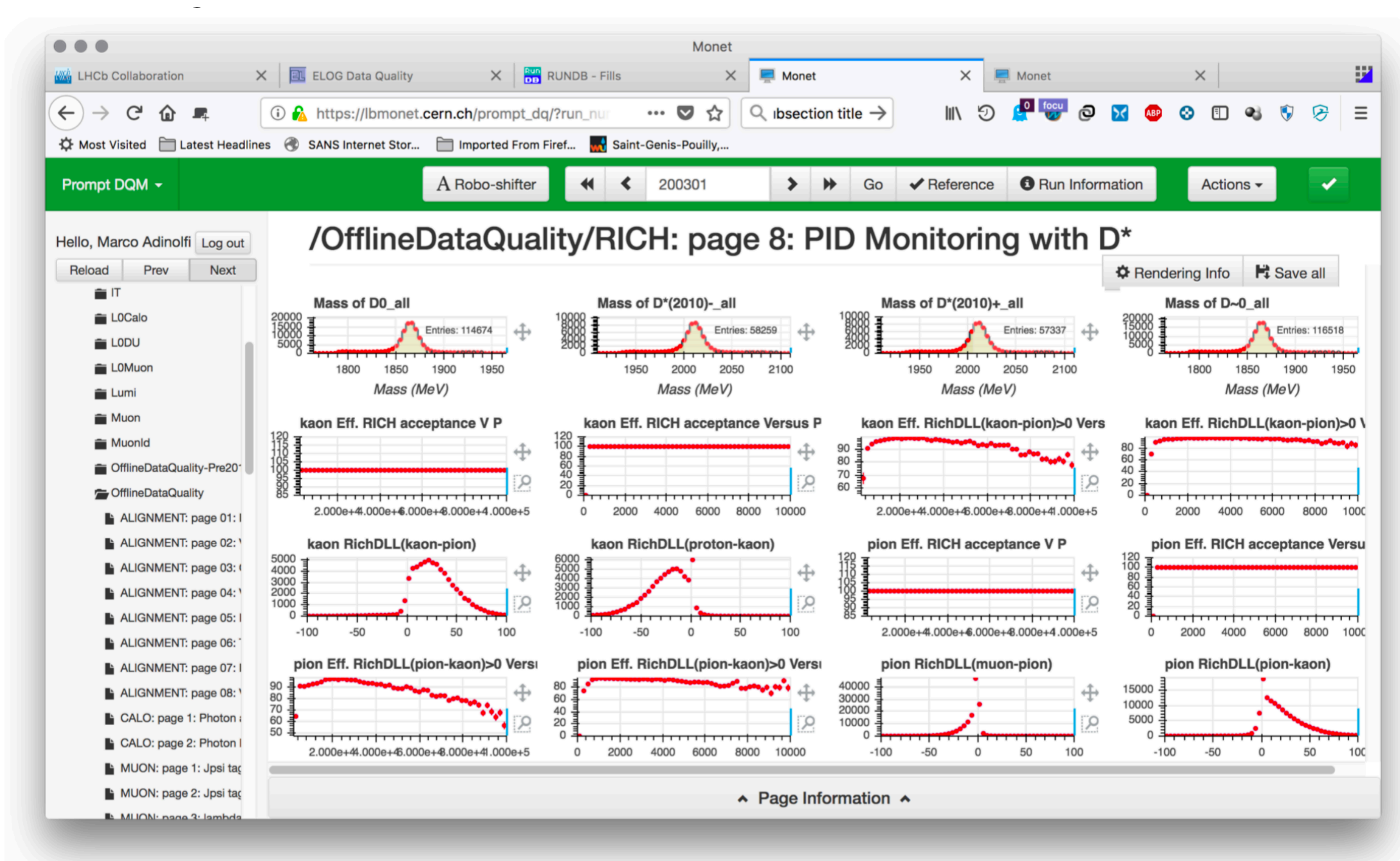


- Which parameters have higher reconstruction error:
  - provide immediate hint of anomaly for the experts



LHCb DQ monitoring in Run 2

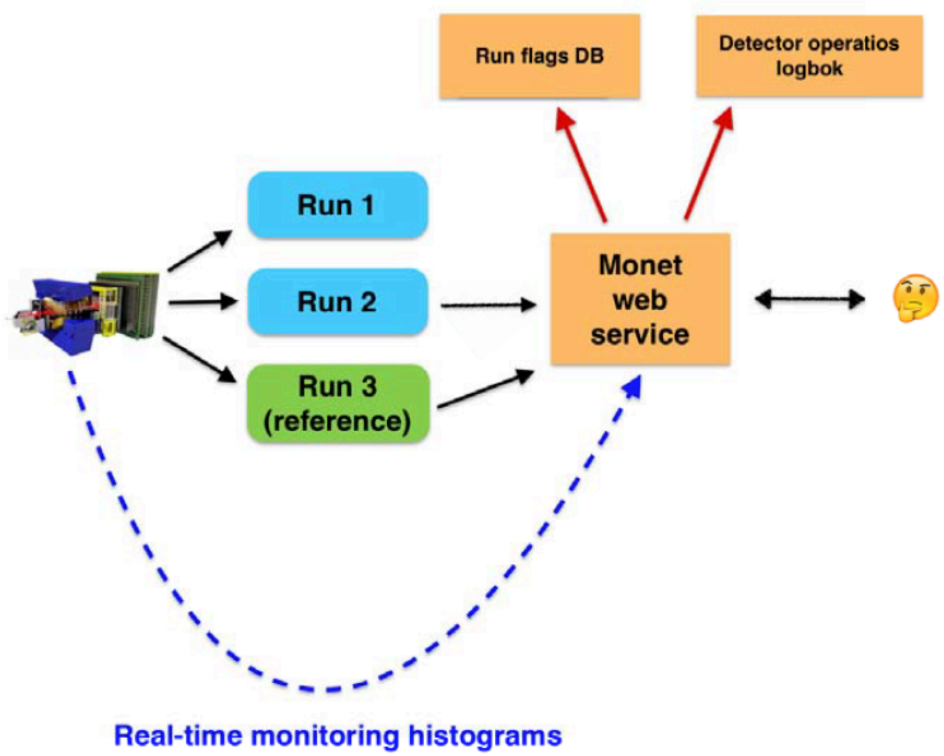
- Data quality shifter studies subset of data selected by trigger
  - flag as GOOD/BAD by comparing to reference run
  - moved to web-based application ***Monet*** during Run 2
  - 48 pages, with multiple histograms

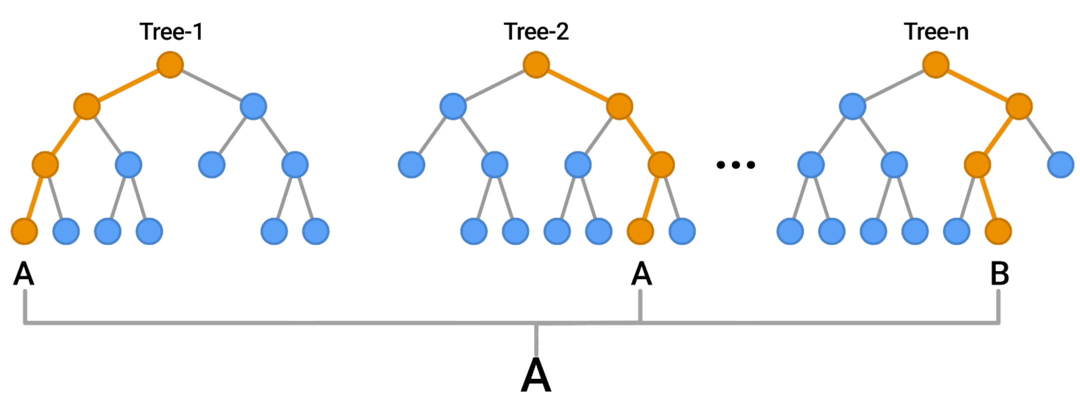


LHCb Roboshifter

J.Phys.Conf.Ser. 898 (2017) 9, 092027

- *Monet* is python-based
  - ML libraries for anomaly detection
  - Kolmogorov-Smirnov distances histograms ↔ references
  - train **shallow BDT**
    one tree per histogram



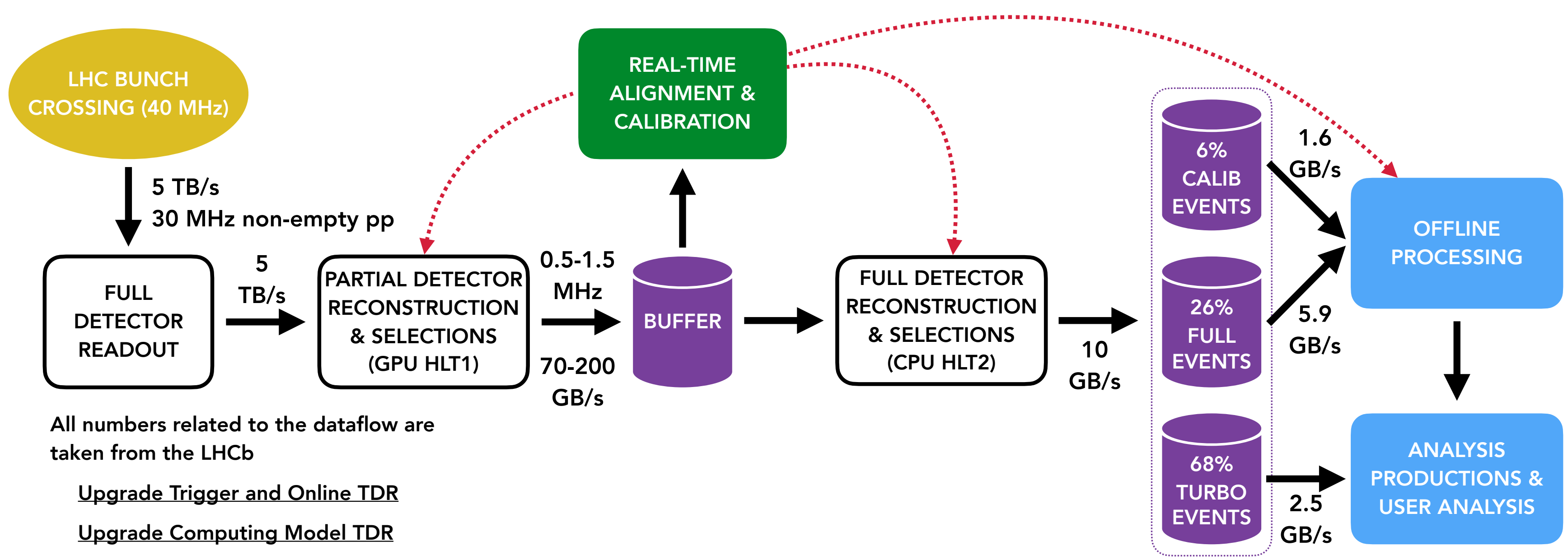


- problematic histograms presented to aid the DQ shifter
- same strategy will be used for Run 3

LHCb in Run 3

LHCB-FIGURE-2020-016

- Currently undergoing a major upgrade for Run 3
  - 5 times higher instantaneous luminosity than Run 2
  - new trigger system developed:
    software only and *real-time alignment + calibration*



  - calibration samples collected in real-time
    → optimise online calibration to minimise differences online/offline
  - monitor not only DQ, but also *software development*

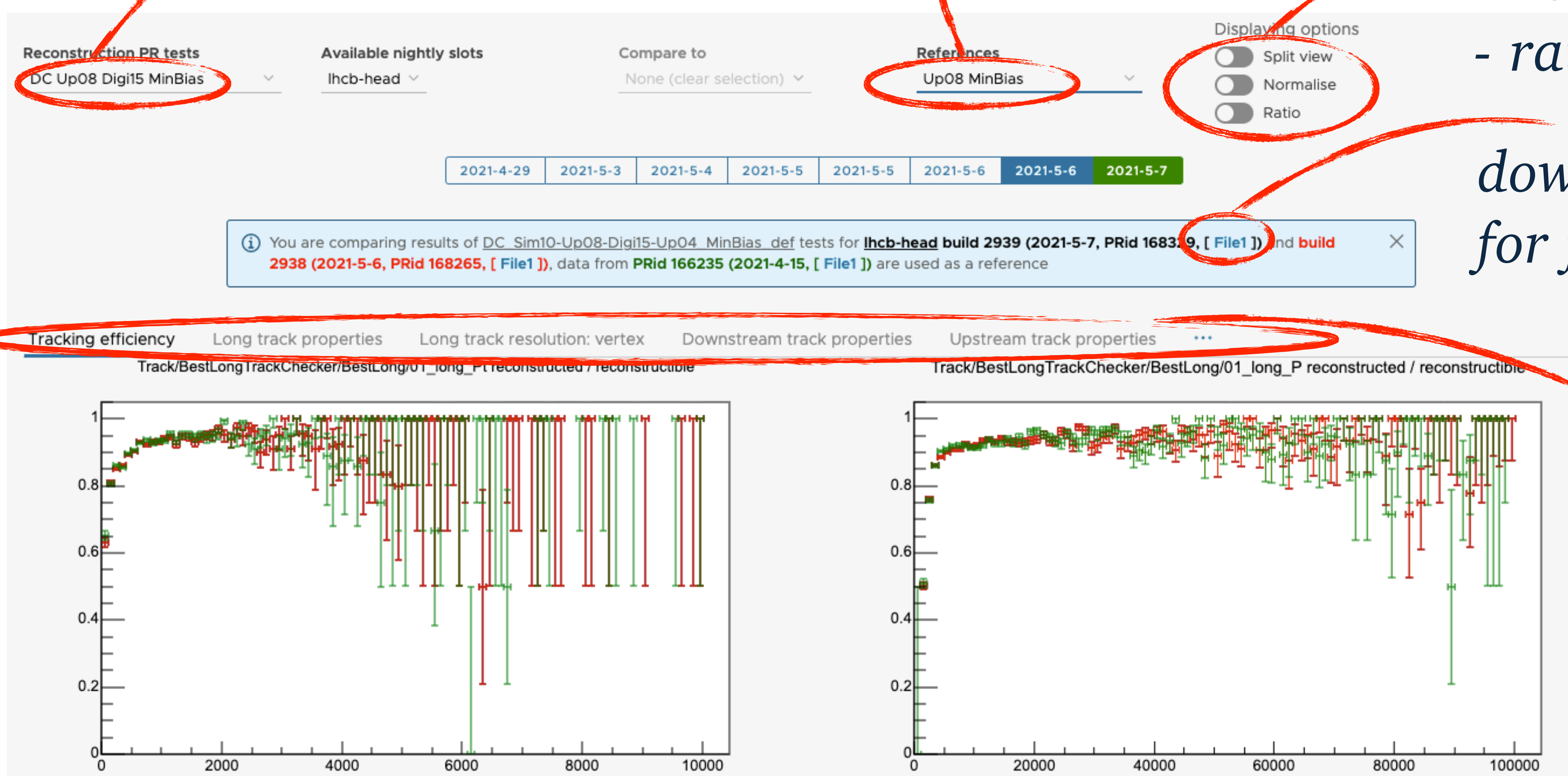
Dashboard to monitor software

- Selected histograms to check performance while changing reconstruction (trigger) software

sample to check
- different decay channels covering majority physics program
- different simulation versions

sample to compare to

- split/overlay samples
- normalise
- ratios

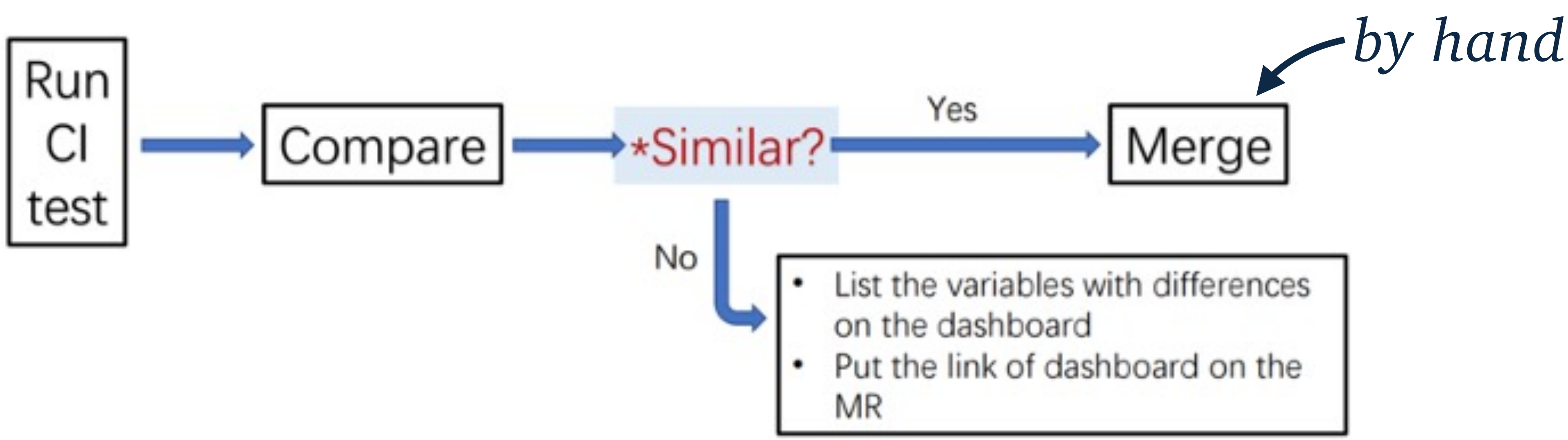


download files locally for further study

many tabs different plots categories

Software monitoring

- **Separate** from DQ monitoring
- Reconstruction tests run nightly over various decay modes
- In addition to existing continuous integration (CI) tests on gitlab, add histogram comparison to each merge request



- Automatic testing allows to test larger amount of possible changes than manual checks, always checked by software shifter

Conclusion

- Progress on data quality monitoring from all experiments, most still in **preliminary stage**
- Experimenting with different ML techniques, most promising seem to be:
  - auto encoders
  - shallow BDTs
- Results look promising and will ease the work of DQ shifters
- Not only DQ monitoring, also software monitoring
- Looking forward to first results of Run 3!