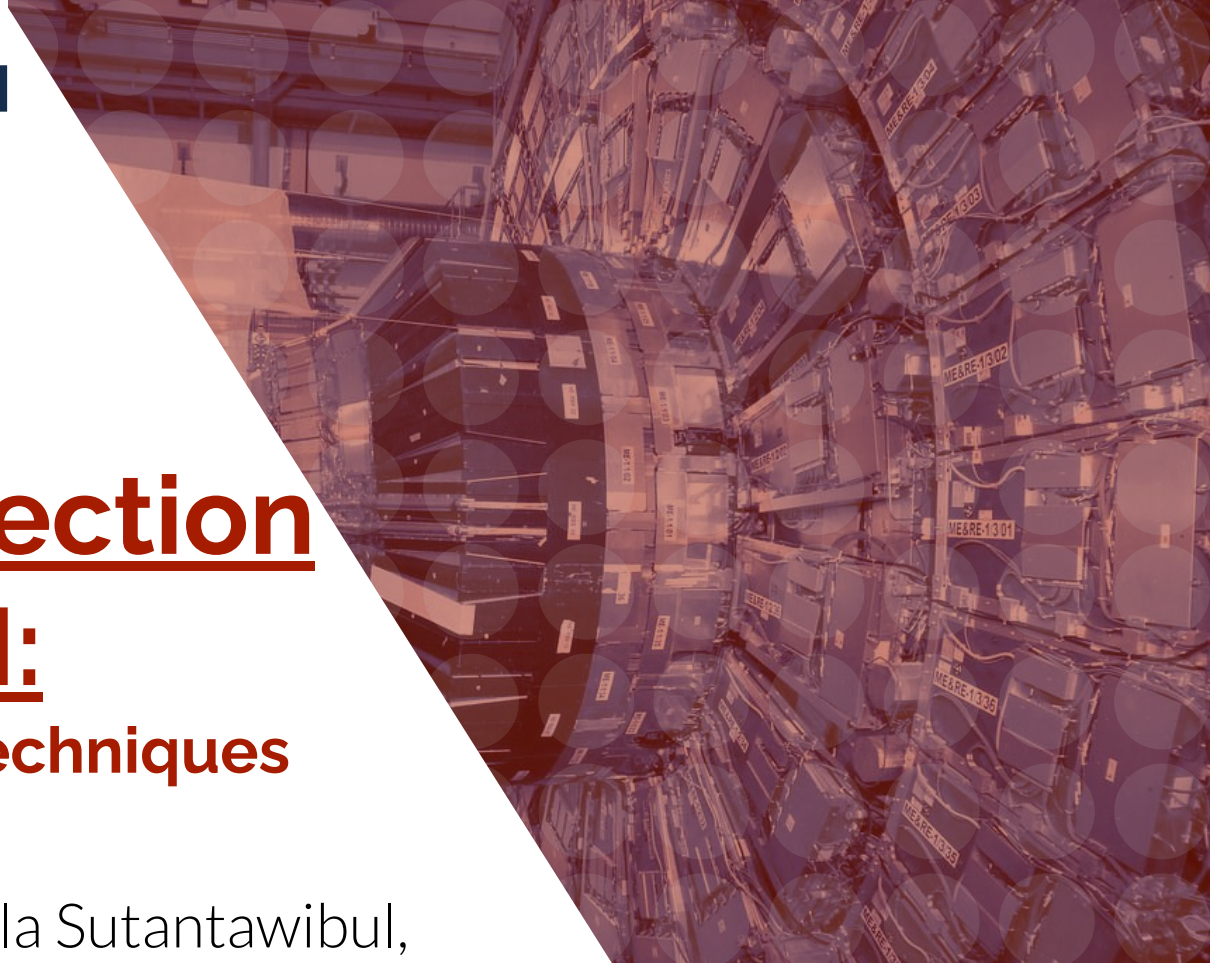
University of BRISTOL

INFN TORINO

AUTO DQM
A STATISTICAL TOOL



Anomaly Detection for DQM:

automation and ML techniques

Andrew Brinkerhoff, Chosila Sutantawibul, Indara Suarez, **Robert White***

27th May 2026

Authors and collaborators

Andrew Brinkerhoff[1,11], Chosila Sutantawibul[1], Indara Suarez[2], **Robert White[3,9]**, Caio Daumann[8], Jonathan Guiang[10], Chad Freer[4,5], Samuel May[2], Bennett Marsh[10], Darin Acosta[7,11], Alex Aubuchon[5], Emanuela Barberis[5], **Aaron Bundock[9]**, Claudio Campagnari[10], Evan Collins[1], Preston Epps[5], Johannes Erdmann[8], **Henning Flaecher[9]**, Junshen Huang[1], Vivan Nguyen[5], Ryan Nie[2], **Sudarshan Paramesvaran[9]**, John Rotter[7,11], Kaitlin Salyer[2], Siddhesh Sawant[1], Tanvi Sheokand[6], Darien Wood[5], and the CMS Muon Detector Collaboration

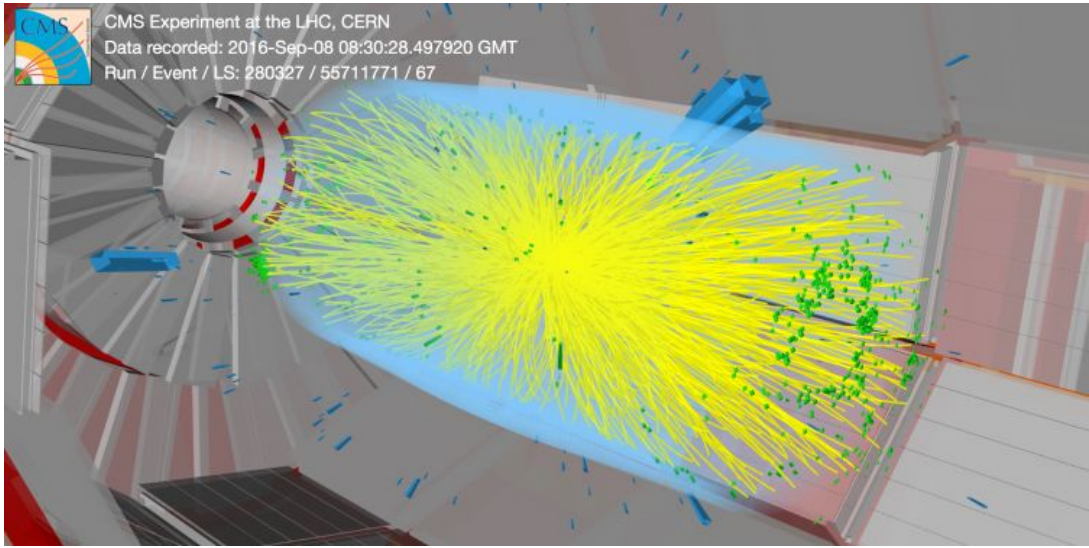
[1] Baylor University, Waco, USA
[2] Boston University, Boston, USA
[3] INFN Sezione di Torino, Turin, Italy
[4] Massachusetts Institute of Technology, Cambridge, USA
[5] Northeastern University, Boston, USA
[6] Panjab University, Chandigarh, India
[7] Rice University, Houston, USA
[8] RWTH Aachen University III, Physikalisches Institut A, Aachen, Germany
[9] University of Bristol, Bristol, UK
[10] University of California Santa Barbara, Santa Barbara, USA
[11] University of Florida, Gainesville, USA

Contents

Proton Collisions

Easy enough to assemble the detector planes…

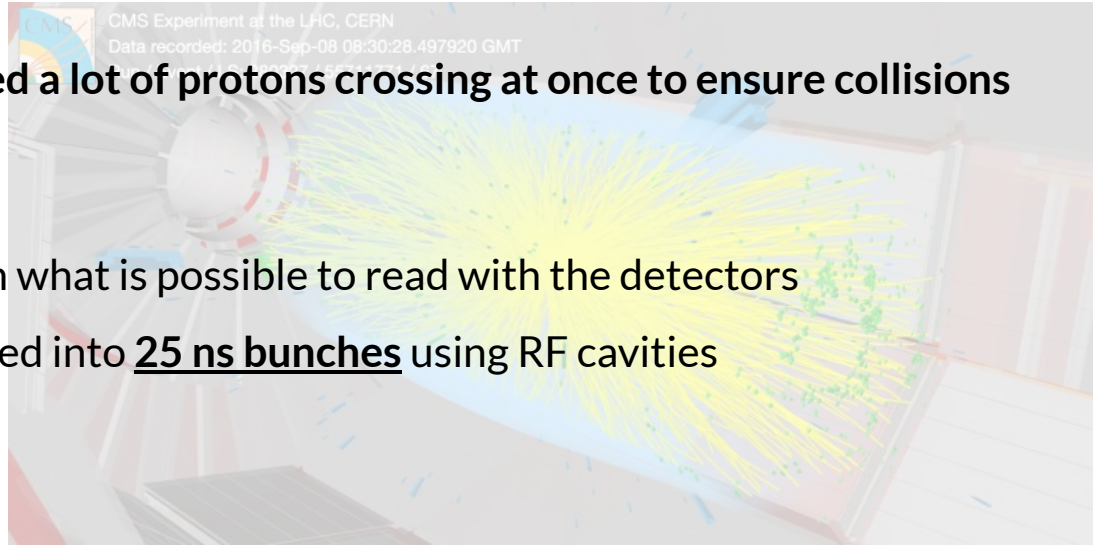
- But will only see flashes as particles interact with these materials
- We don't see a single *pp* collision: in Run 2 (2015-2018) average of 32 collisions per crossing; in Run 3 (2022-2026) average between 50 and 60; HL-LHC expected 120 (called **pileup**)



Proton Collisions

Easy enough to assemble the detector planes…

- But will only see flashes as particles interact with these materials
- We don't see a single *pp* collision: in Run 2 (2015-2018) average of 32 collisions per crossing; in Run 3 (2022-2026) average between 50 and 60; HL-LHC expected 120 (called **pileup**)



As *pp* collision probability is very small, need a lot of protons crossing at once to ensure collisions are recorded

- Balance good number of collisions with what is possible to read with the detectors
- Protons reach LHC and are synchronised into **25 ns bunches** using RF cavities

Triggering and Data Acquisition

Including pileup events:
> 1 GHz

If 25 ns intervals, information read out at a rate of 40 MHz!

1 event of interest:
40 MHz

- The CMS Level 1 Trigger (**L1T**) uses calorimeter and muon detector information for preliminary event reconstruction in ~4 μs

L1T *accepted*:
100 kHz

- The High-Level Trigger (**HLT**) reconstructs events in more detail with a software/firmware farm of CPU and GPU for disk storage

Full event reconstruction:
~ 1 kHz

Triggering and Data Acquisition

Including pileup events:
> 1 GHz

If 25 ns intervals, information read out at a rate of 40 MHz!

1 event of interest:
40 MHz

- The CMS Level 1 Trigger (**L1T**) uses calorimeter and muon detector information for preliminary event reconstruction in ~4 μs

L1T *accepted*:
100 kHz

- The High-Level Trigger (HLT) reconstructs events in more detail with a software/firmware farm of CPU and GPU for disk storage

Full event reconstruction:
~ 1 kHz

L1T programmed into custom FPGA firmware chips to reduce event rate to 100 kHz

- Trigger primitives (**TPs**) are crude info handled within ECAL, HCAL, muon detectors
  - Subsystem TPs filtered then combined in global trigger and further evaluated
    - L1T decides within 4 μs whether to accept or reject the event

L1T TP Handling

L1T programmed into custom FPGA firmware chips to reduce event rate to 100 kHz

- Trigger primitives (**TPs**) are crude info handled within ECAL, HCAL, muon detectors
  - Subsystem TPs filtered then combined in global trigger and further evaluated
    - L1T decides within 4 μs whether to accept or reject the event

POV Calos

L1 Calorimeter Trigger is two-tiered

- TPs containing p_T and quality flags from each of the ECAL and HCAL transmitted via gigabit **fibre optic links** to the first tier, Calo Layer 1 (CL1 - *L1 not meaning Level 1 of the trigger!*)
- Processes information in parallel, summing TP energy deposits into calo **towers**
- Local corrections to p_T and other kinematic quantities with η-, energy- and cluster-shape-dependent calibration

L1T TP Handling

L1T programmed into custom FPGA firmware chips to reduce event rate to 100 kHz

- Trigger primitives (**TPs**) are crude info handled within ECAL, HCAL, muon detectors
  - Subsystem TPs filtered then combined in global trigger and further evaluated
    - L1T decides within 4 μs whether to accept or reject the event

POV Calos

L1 Calorimeter Trigger is two-tiered

- CL1 tower info fed via high-bandwidth links to second tier, Calo Layer 2 (CL2)
- CL2 clusters these towers together to form calo **objects**
- Particle identification and reconstruction algorithms applied (**PFA, CHS, PUPPI - see backup**)
- Global energy sums defined at this stage e.g. missing transverse energy, p_T^{miss} with Si tracker info

L1T TP Handling

L1T programmed into custom FPGA firmware chips to reduce event rate to 100 kHz

- Trigger primitives (**TPs**) are crude info handled within ECAL, HCAL, muon detectors
  - Subsystem TPs filtered then combined in global trigger and further evaluated
    - L1T decides within 4 μs whether to accept or reject the event

POV Calos

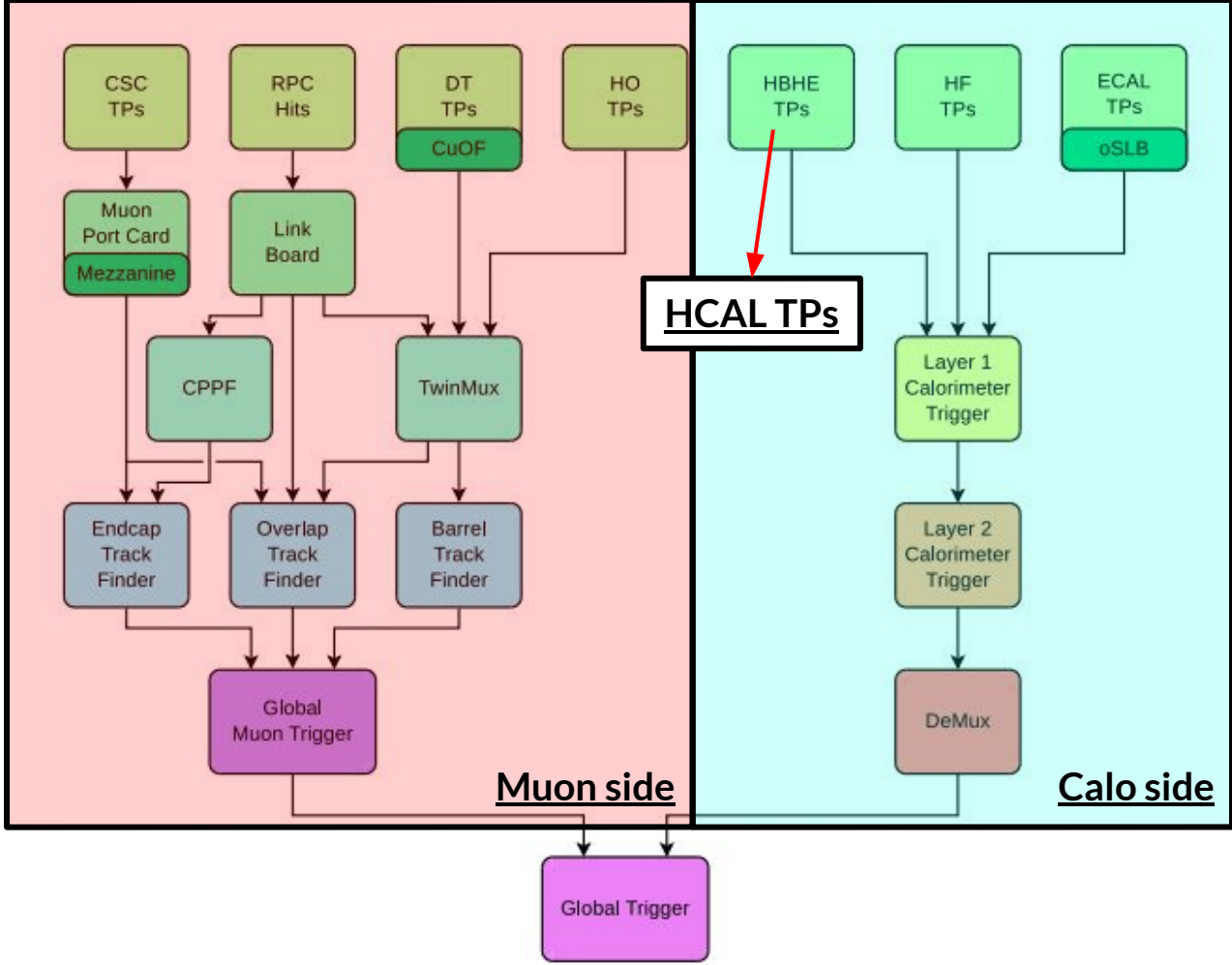
CL2 sends info to global trigger

- Final stage of L1T looks at event objects from both CL2 trigger and **global muon trigger**
- Selection using between 350 and 400 trigger criteria before L1T accepted
- Accepted events sent to HLT at 100 kHz rate

The L1T Flow

The flow that takes ~ 4 µs

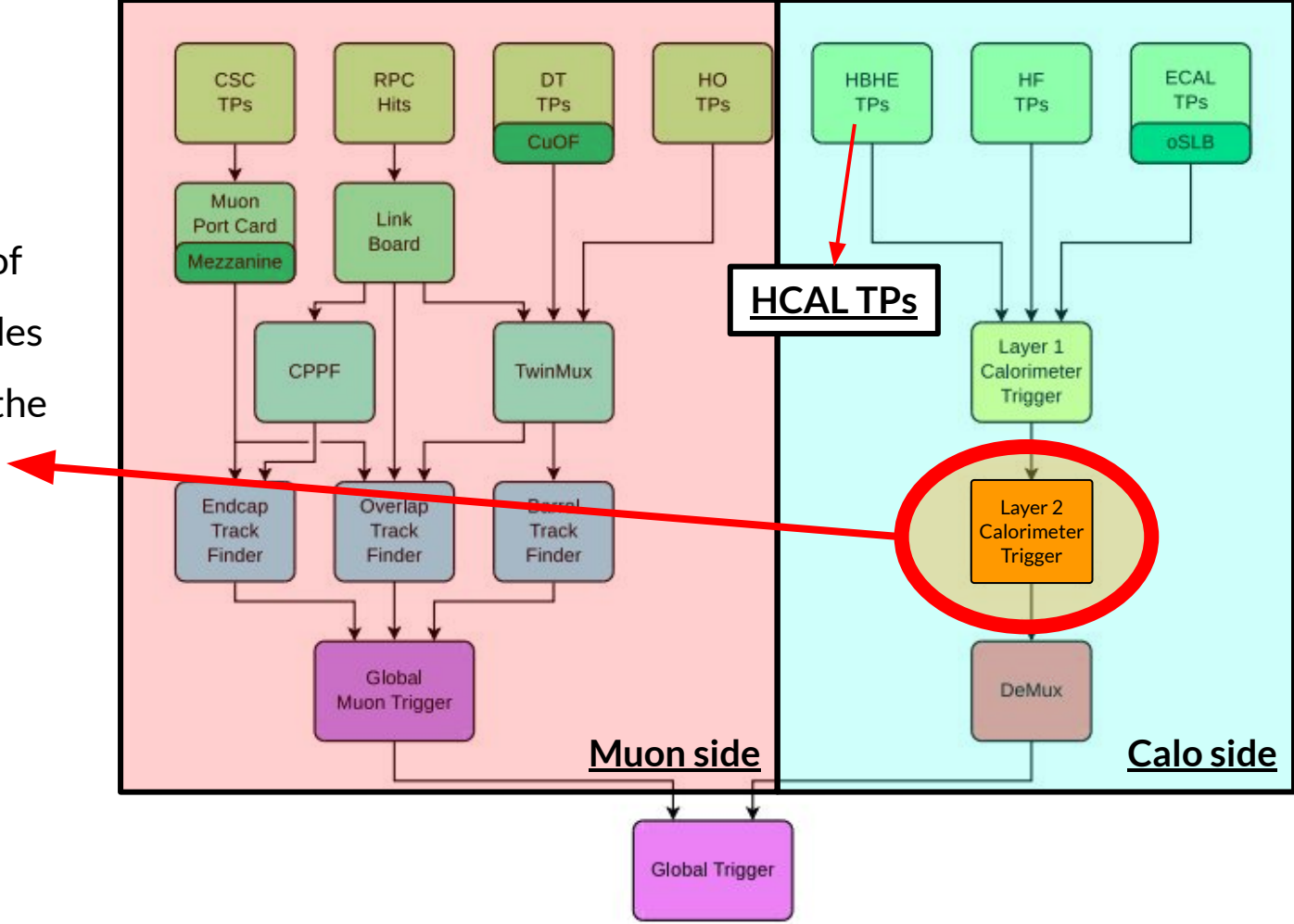
- Physically, a network of processors on both sides



The L1T Flow

The flow that takes ~ 4 μs

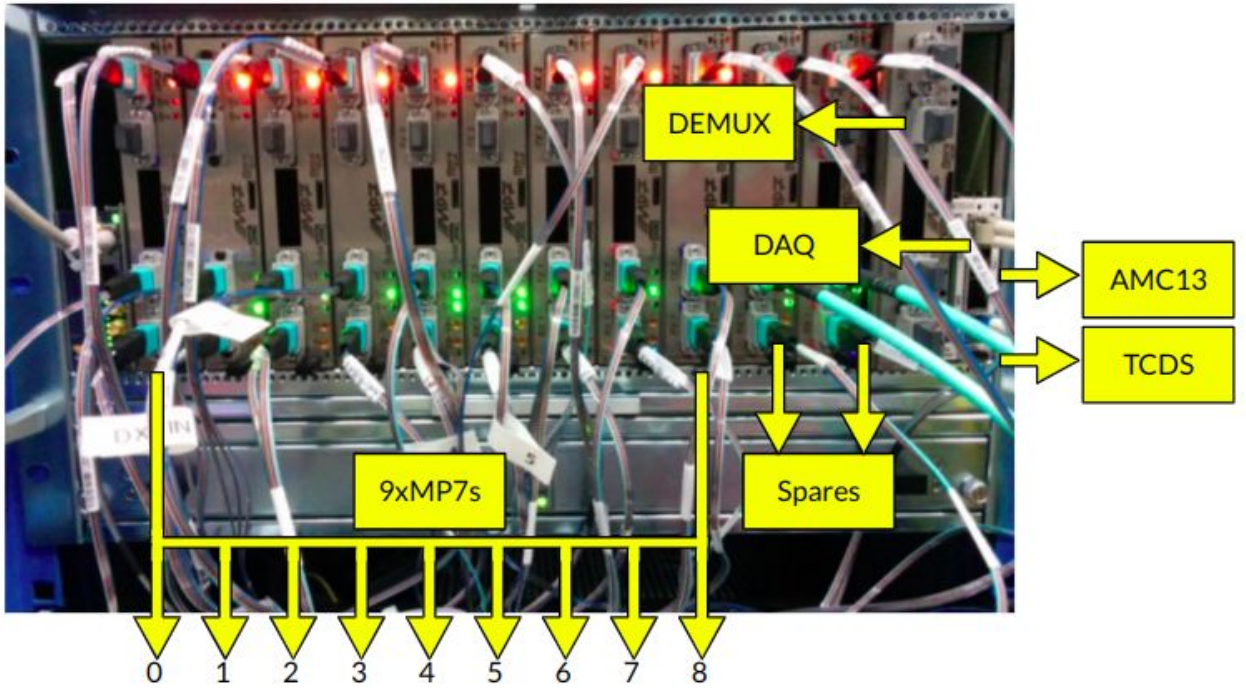
- Physically, a network of processors on both sides
- Let's focus on CL2 on the calo side



CL2 crate

Micro Telecoms Computing Architecture (MicroTCA) crate

- 9 master processors (MP7)
- Each reads a 25 ns bunch crossing at a time
- Info is *time-multiplexed*: calo towers across all calo subsystems from CL1 timestamped and made available



Towers clustered into calo objects

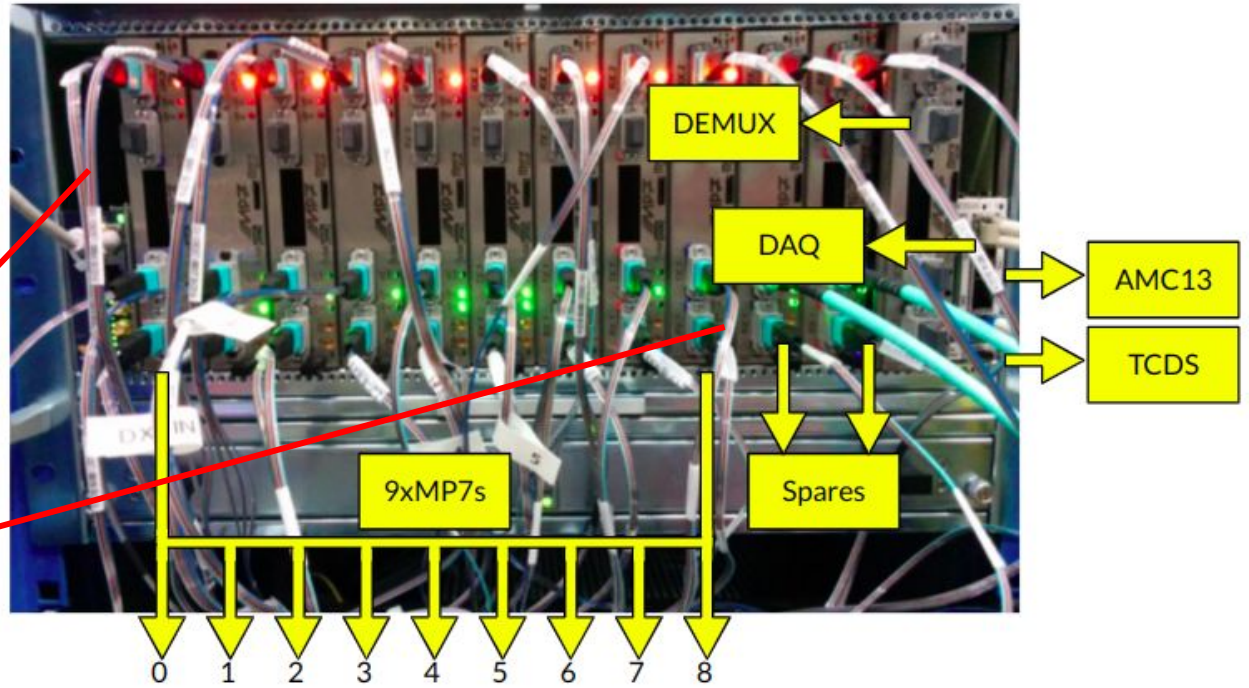
- Calo objects held on single FPGAs within MP7

CL2 crate

Micro Telecoms Computing Architecture (MicroTCA) crate

- 9 master processors (MP7)
- Each reads a 25 ns bunch

Fibre optic cables
"links" in/out of CL2

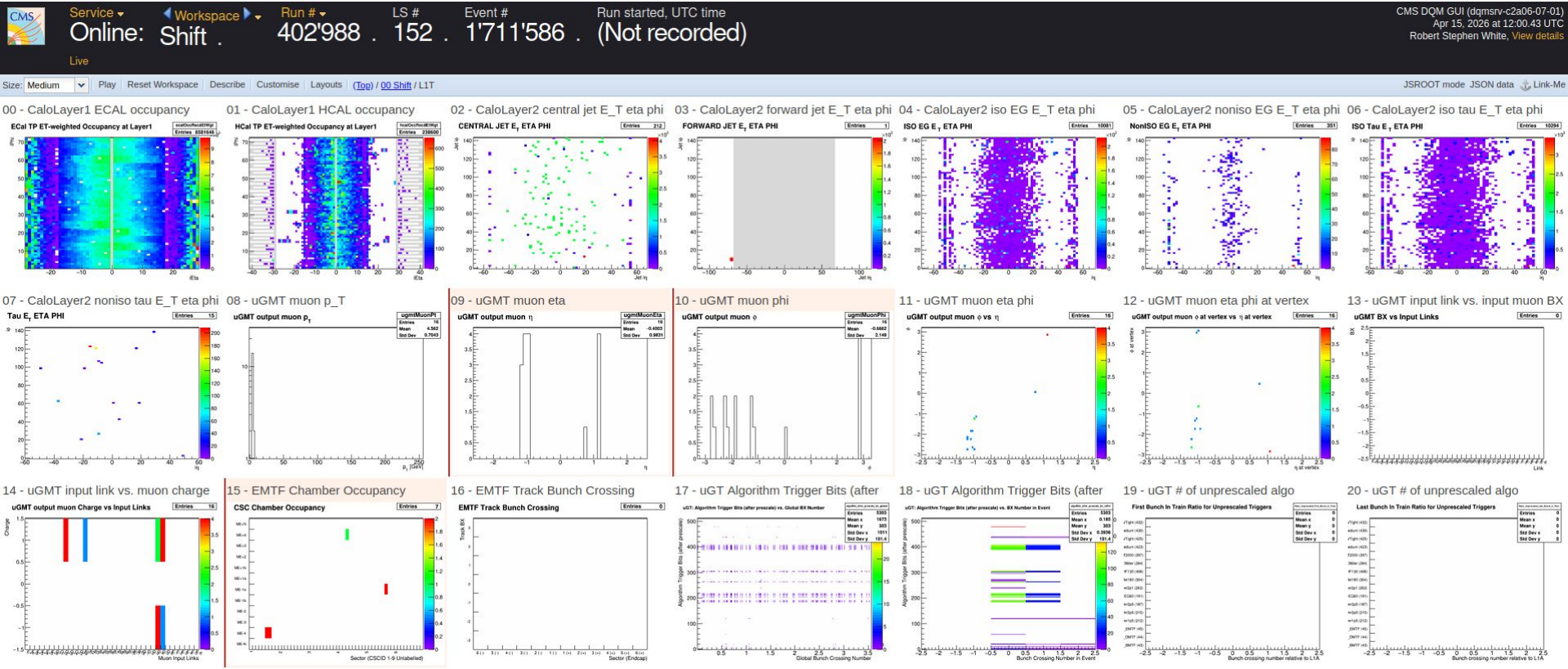


Towers clustered into calo objects

- Calo objects held on single FPGAs within MP7

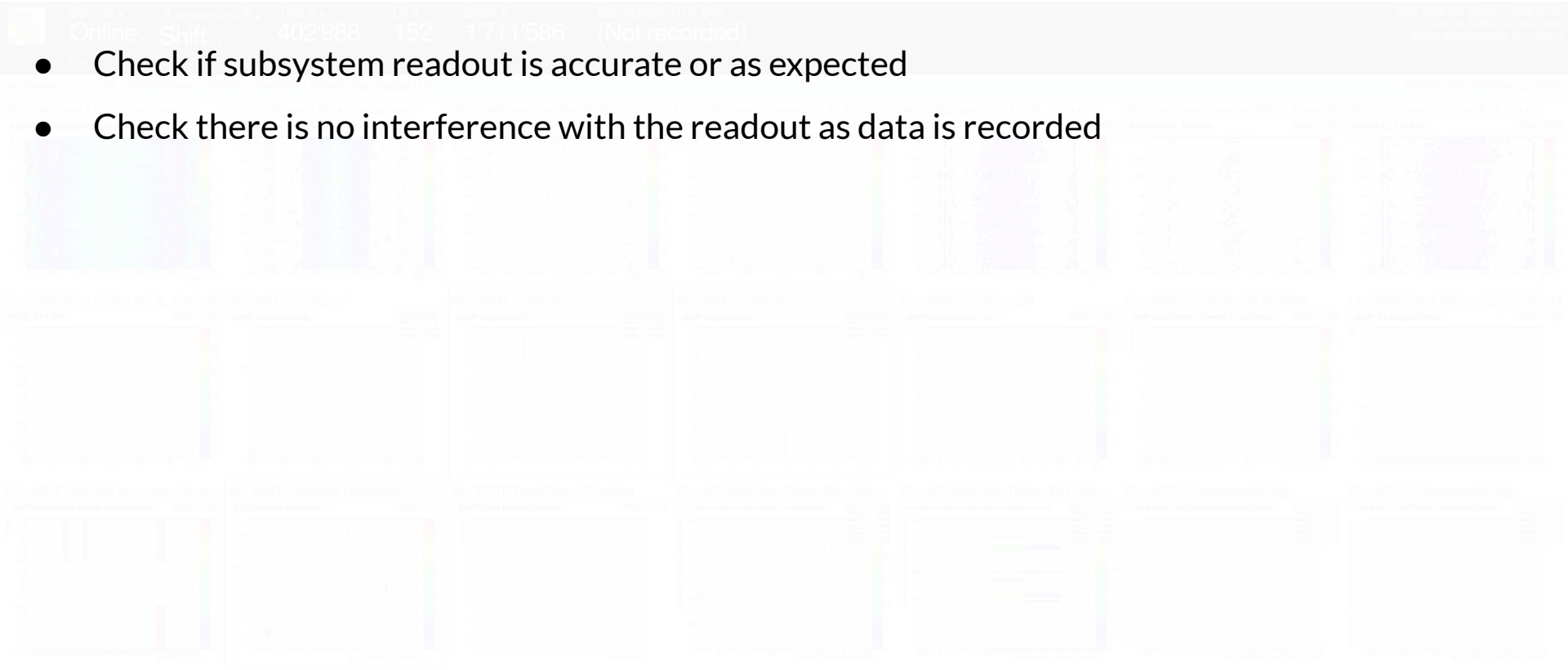
Contents

Data Quality Monitoring (DQM)



Data Quality Monitoring (DQM)

Data readout needs to be monitored continuously, both during acquisition and after



- Check if subsystem readout is accurate or as expected
- Check there is no interference with the readout as data is recorded

Data Quality Monitoring (DQM)

Data readout needs to be monitored continuously, both during acquisition and after

- Check if subsystem readout is accurate or as expected
- Check there is no interference with the readout as data is recorded

During: Online DQM involving real-time diagnostics by *shifters*

- Shifters monitor output histograms/flags/rates from each subsystem and trigger
- Detector experts on-call (DOCs) make final decisions for DQM and detector operations

Data Quality Monitoring (DQM)

Data readout needs to be monitored continuously, both during acquisition and after

- Check if subsystem readout is accurate or as expected
- Check there is no interference with the readout as data is recorded

During: Online DQM involving real-time diagnostics by *shifters*

- Shifters monitor output histograms/flags/rates from each subsystem and trigger
- Detector experts on-call (DOCs) make final decisions for DQM and detector operations

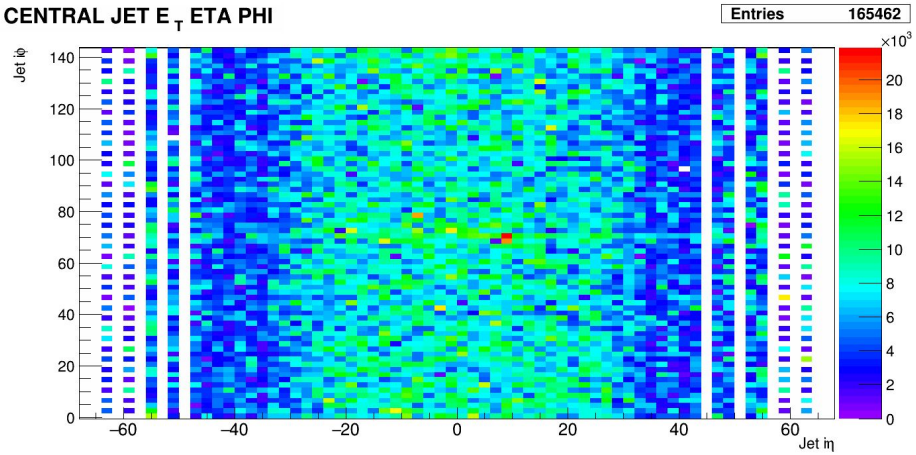
After: Offline DQM when full data sets are available

- Typically efficiency or module performance studies
- In CL2, comparison of calo object kinematics between data and emulator, or diagnostics

CL2 DQM

UK specialises in CL2 trigger

- Calo objects
- Energy sums
- Both 1D and 2D histograms

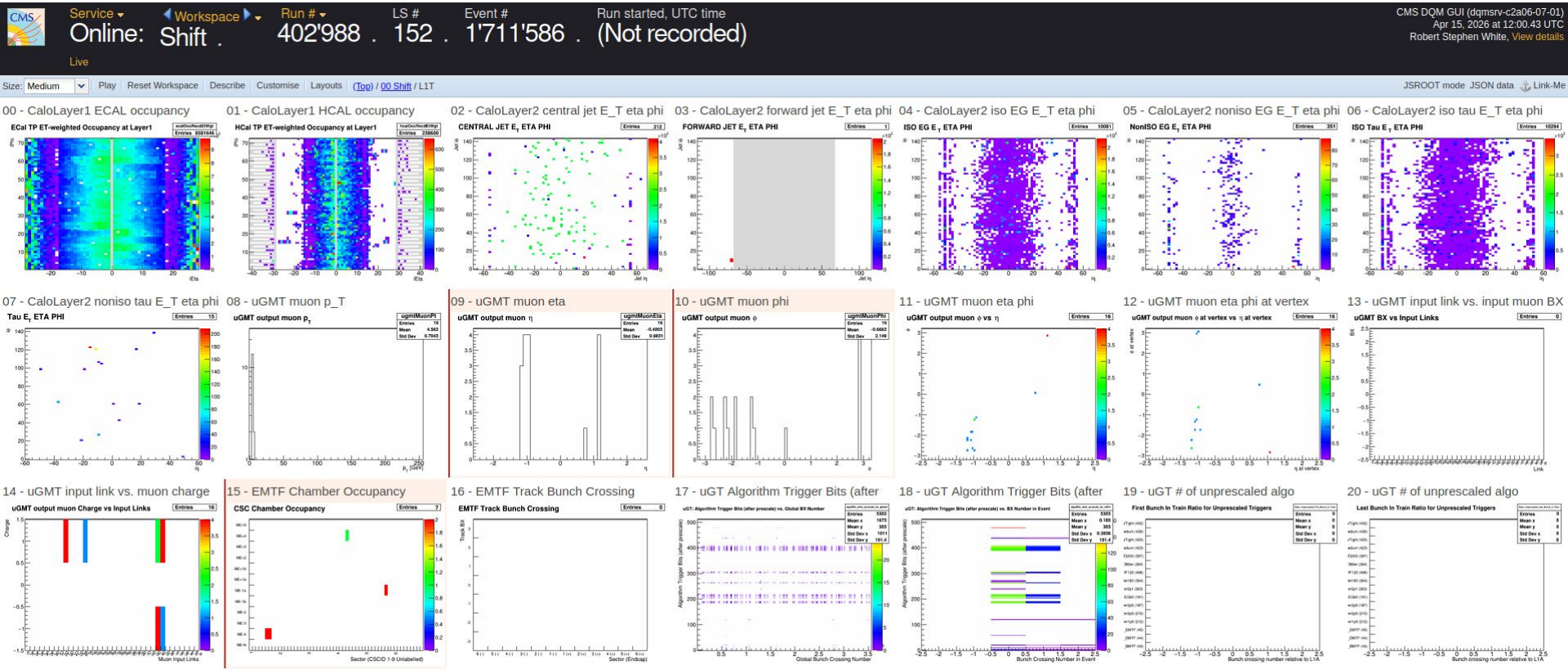


Jets	Central	Occupancy, E_T, (E_T, η, ϕ)
	Forward	
e/γ	Isolated	
	Non-isolated	
τ	Isolated	
	Non-isolated	

Scalar	H_T	E_T, ϕ
	E_T	
Vector	E_T^{miss}	
	H_T^{miss}	
	E_T	

Shifters' Remit

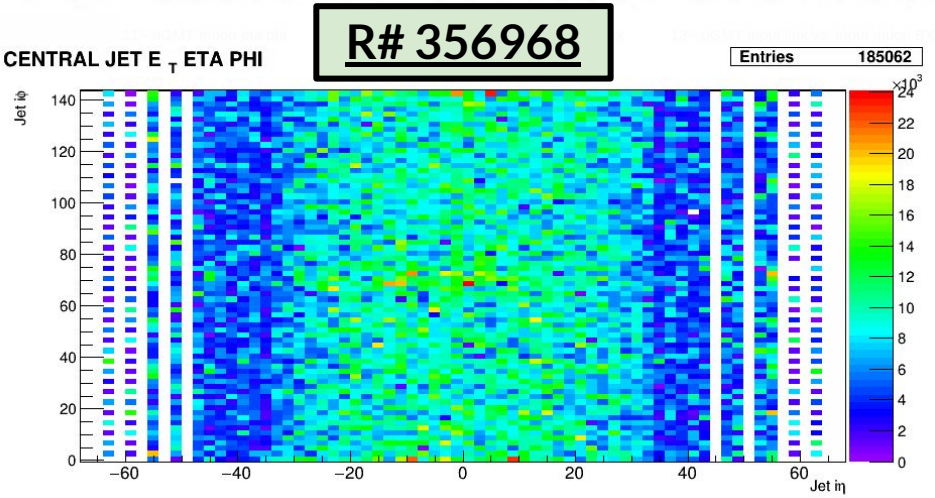
Go to CMS DQM Online or CMS OMS and navigate to Shift->L1T tab *(former needs certificates)*



Shifters' Remit

Shifters look for anomalies in these histograms

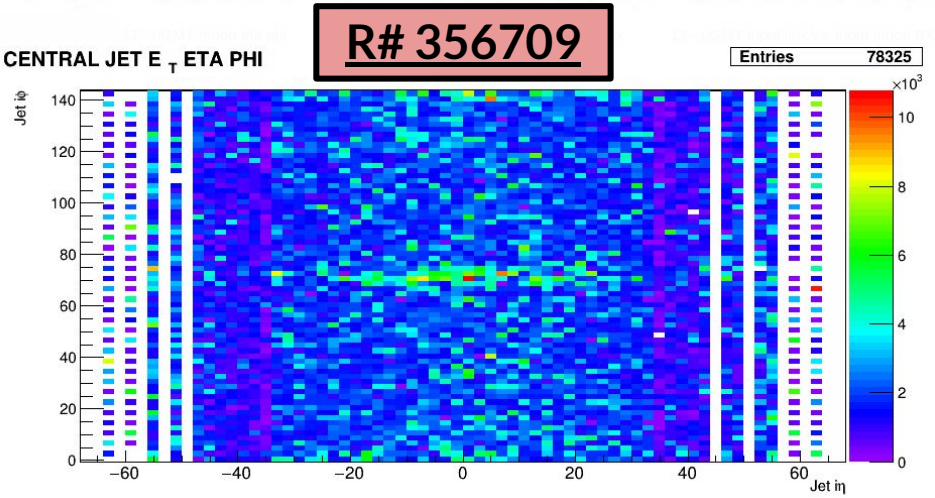
- Check for **spikes** (high occupancy) due to hot towers (*ECAL/HCAL*), calo tower geometry (*CL1*) or faulty trigger link (*CL2 affected only*)
- Check for **holes** (masked or dead towers)
- Masks applied by shifters on-the-fly by ECAL/HCAL DOCs
- **Asymmetries** in η halves
- Large **dead zones** (subsystem fault/inactive)
- Can refer masks to Layer 1 histograms



Shifters' Remit

Shifters look for anomalies in these histograms

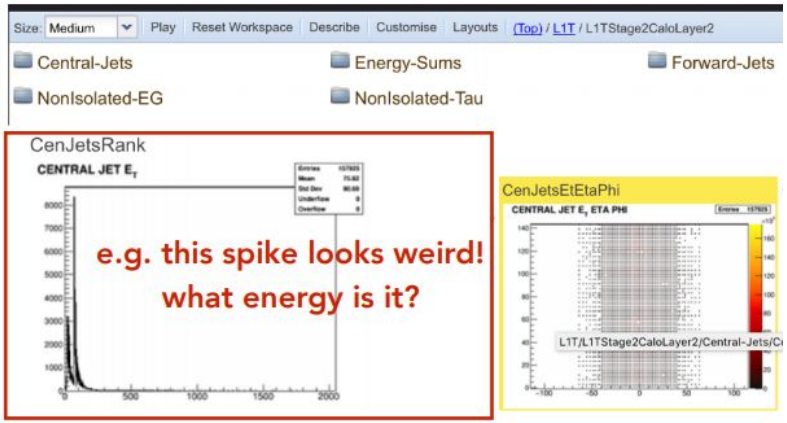
- Check for **spikes** (high occupancy) due to hot towers (*ECAL/HCAL*), calo tower geometry (*CL1*) or faulty trigger link (*CL2 affected only*)
- Check for **holes** (masked or dead towers)
- Masks applied by shifters on-the-fly by ECAL/HCAL DOCs
- **Asymmetries** in η halves
- Large **dead zones** (subsystem fault/inactive)
- Can refer masks to Layer 1 histograms

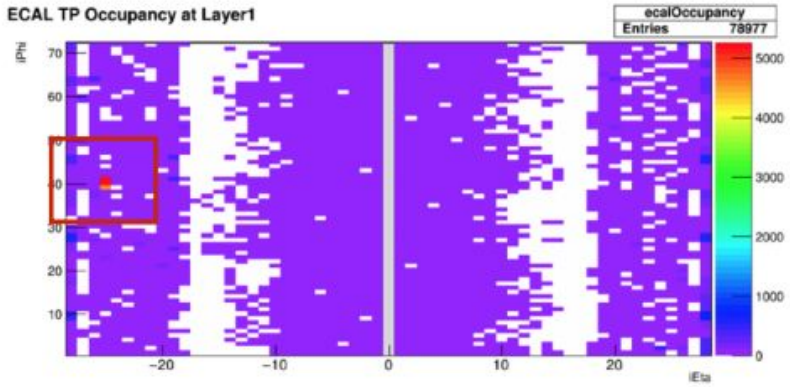


Shifters' Remit

Textbook example for debugging between calo trigger tiers

- Problem appears in CL2 Central Jets E_T histogram
- Trace the spike backwards to CL1 and ECAL subsystems
- Found in CL1: TP occupancy anomalously high in one bin
- DOC will recommend to mask this bin if problem persists





Shifters' Remit

Involves a lot of histogram checking for the shifters

- Could end up sifting through hundreds of histograms to trace and identify an issue
- Shifters not always experts in the detector subsystems (hello 👋)
- Human error is always a factor
- Not all issues are so obvious: what if there are obscure anomalies invisible to our eyes?

This can easily be an automated job performed by a stats validation tool

Contents

1. The CMS Detector
2. Data quality monitoring (DQM)
3. Automated DQM using statistical analysis
4. Automated DQM using machine learning tools
5. Automated DQM today

AutoDQM

Automated DQM software to help shifters detect anomalies

- Designed to find hard-to-spot issues faster than shifters (online and offline)
- **Pull value** calculated for each bin in 2D histograms (Poissonian errors)

$$\text{pull value} = \frac{(x_1 - x_2)^2}{\epsilon_1^2 + \epsilon_2^2}$$

- Beta-binomial (**βB**) tests in 1D and 2D

AutoDQM

Automated DQM software to help shifters detect anomalies

- Designed to find hard-to-spot issues faster than shifters (online and offline)
- **Pull value** calculated for each bin in 2D histograms (Poissonian errors)

$$\text{pull value} = \frac{(x_1 - x_2)^2}{\epsilon_1^2 + \epsilon_2^2}$$

- Beta-binomial (**βB**) tests in 1D and 2D

Histograms are "anomalous" if above max pull/Chi2 threshold, informed by DPGs/detector experts

Set up for all subsystems: https://github.com/AutoDQM/AutoDQM

Test bed: https://cmsweb-testbed.cern.ch/dqm/autodqm (*needs certificates*)

Live site for muon subsystems: https://muon-autodqm.web.cern.ch/

βB tests

Histograms from different runs have vastly different numbers of entries

- Can be evenly distributed or concentrated in a small set of bins
- Test histogram with entries d_i in bin i treated as frequency in D trials, D as total event number
- Reference histogram from a prior run compared by r_i in integral R as frequency
- Compute likelihood $\mathscr{L}_i$ to observe d_i in each bin using βB function from SciPy

βB tests

Histograms from different runs have vastly different numbers of entries

- Can be evenly distributed or concentrated in a small set of bins
- Test histogram with entries d_i in bin i treated as frequency in D trials, D as total event number
- Reference histogram from a prior run compared by r_i in integral R as frequency
- Compute likelihood $\mathcal{L}_i$ to observe d_i in each bin using βB function from SciPy

$$\mathcal{L}_i = f(d_i|D,\alpha,\beta) = \binom{D}{d_i} \frac{B(d_i+\alpha, D-d_i+\beta)}{B(\alpha,\beta)}$$

$$\alpha = \alpha_0 + r_i$$
$$\beta = \beta_0 + R - r_i$$
$$\alpha_0 = \beta_0 = 1 \quad \text{-> uniform priors}$$

$$\mathcal{L}_i^{\max} = D\frac{r_i}{R}$$

$$\mathcal{L}_i^{\text{rel}} = \frac{\mathcal{L}_i}{\mathcal{L}_i^{\max}} \mapsto Z_i^2 = -2\ln(\mathcal{L}_i^{rel})$$

$$\tau = 1/\sqrt{1+(10^{-4}r_i)^2}$$

- Compute $\mathcal{L}_i$ for all bins, compare to $\mathcal{L}_i^{\max}$, and compute pull value Z_i (standard deviation units)
- Scale r_i and R by τ when calculating $\mathcal{L}_i^{\text{rel}}$ for tolerance of 1% in high-occupancy bins (τr_i -> 10^4)
- For more than one reference run, use average $\mathcal{L}_i^{\text{rel}}$ values

βB tests

If observed data matches the expectation for a histogram from one of many reference runs, pull values small

- For instance, if seven reference runs yield $\mathscr{L}_i^{rel} \approx 0$ ($Z_i \approx \infty$) for a given bin, but one reference run gives $\mathscr{L}_i^{rel} \approx 0.33$ ($Z_i \approx 1.5$), $Z_i = \sqrt{(-2 \ln(0.33/8))} \approx$ **2.5**

So βB accounts for systematic variations between running conditions that would otherwise flag as an anomaly

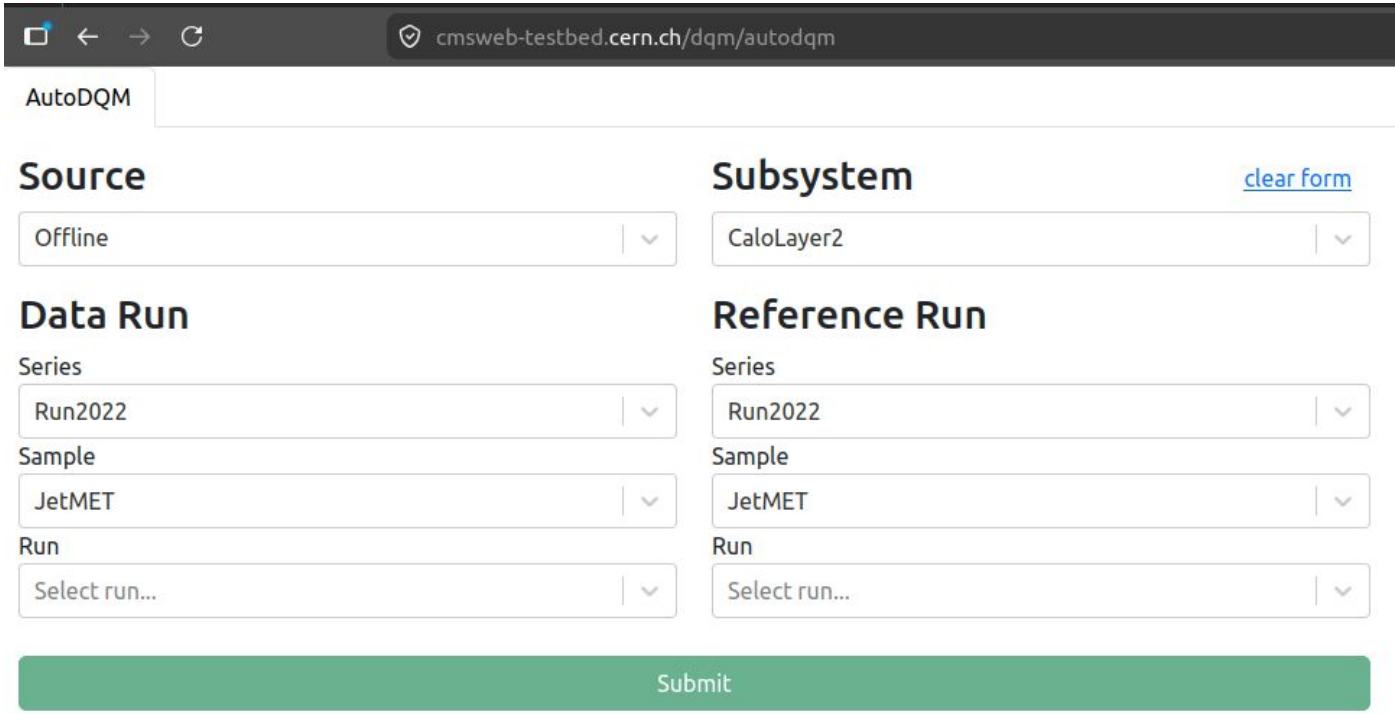
AutoDQM test bed GUI

Earlier we saw two runs from 2022

- 356968 (good)
- 356709 (bad)

We saw that for the "bad" run the Central Jet E_TEtaPhi was strange

- Let's compare them on the test bed



AutoDQM test bed GUI

Two of each histogram:
- 1D: βB
- 2D: Pull value + βB

Only shows "anomalous" plots by default

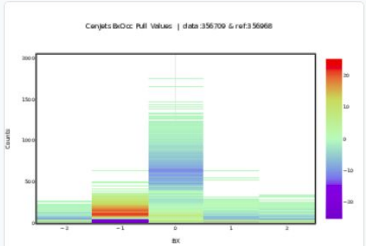
Known *"bad"* run

Example *"good"* run

DQM Report
source: Offline
ystem: CaloLayer2
n 2023 10:35:59 GMT

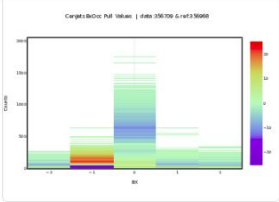
Details	Data Run	Ref Run
Series	Run2022	Run2022
Sample	JetMET	JetMET
Run	356709	356968

Show hidden plots

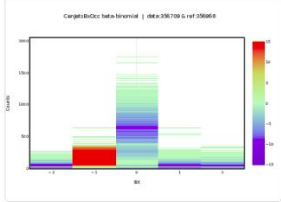


Name	CenJetsBxOcc
Comparator	pull_values
Anomalous	true
Chi_Squared	13.35
Max_Pull_Val	30.48
Data_Entries	184524.0

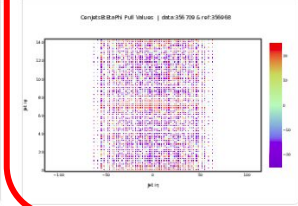
CenJetsBxOcc



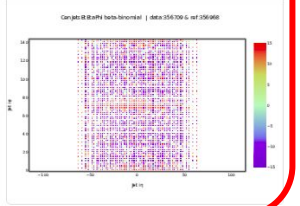
CenJetsBxOcc



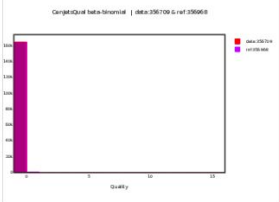
CenJetsEtEtaPhi



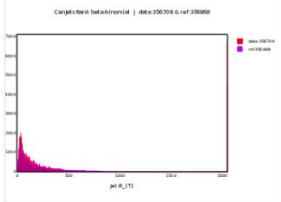
CenJetsEtEtaPhi



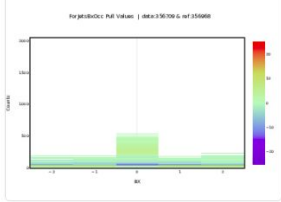
CenJetsQual



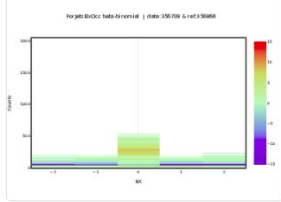
CenJetsRank



ForJetsBxOcc



ForJetsBxOcc

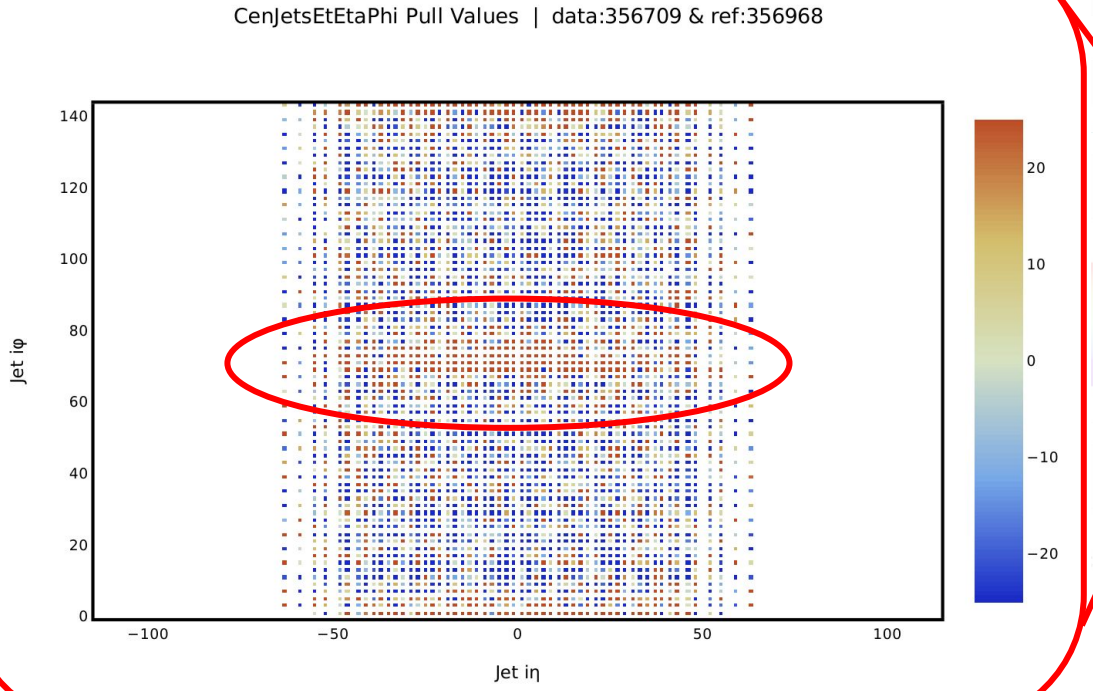


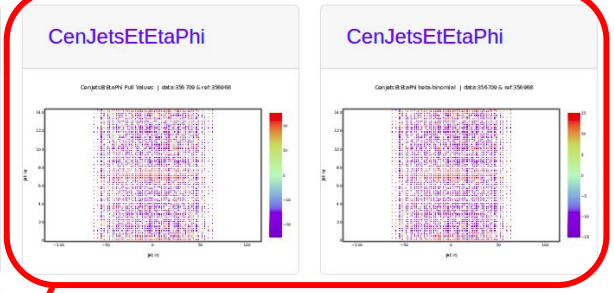
AutoDQM test bed GUI

Known *"bad"* run

Example *"good"* run

Data Run	Ref Run
Run2022	Run2022
JetMET	JetMET
356709	356968

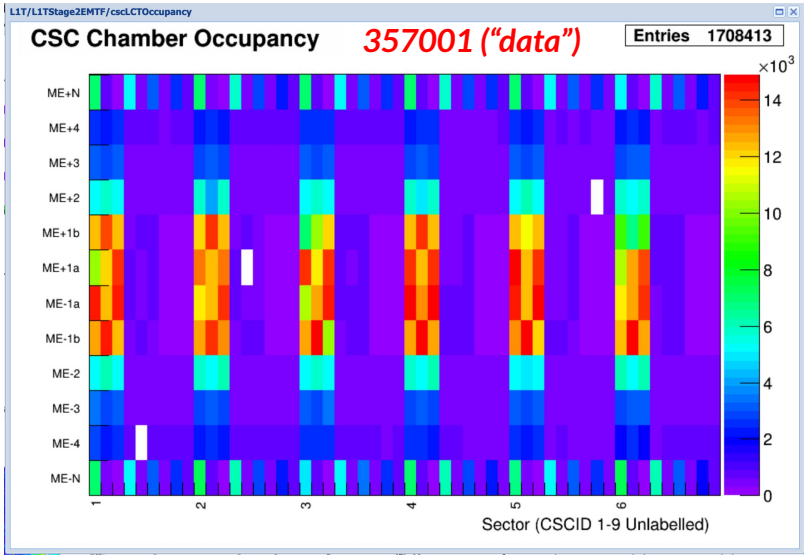


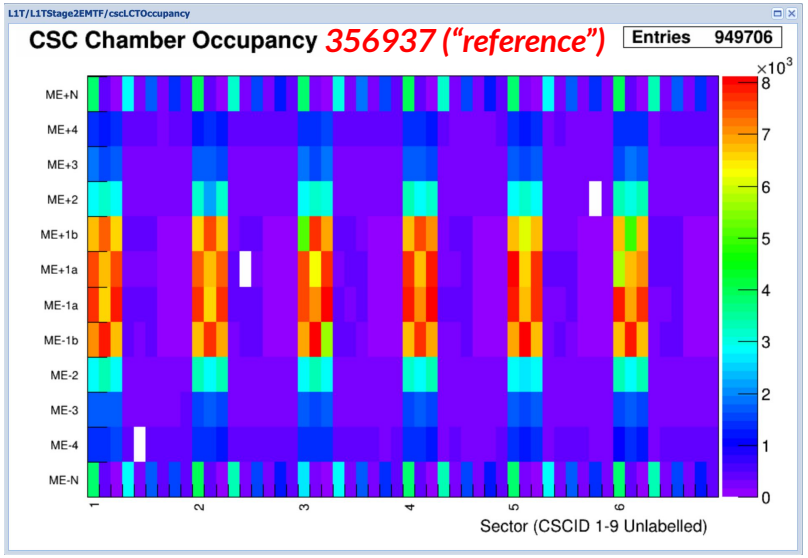


Both pull value and βB raise this as anomalous

Towards multiple run comparisons

…because in reality shifters look at multiple runs to understand an under-study histogram

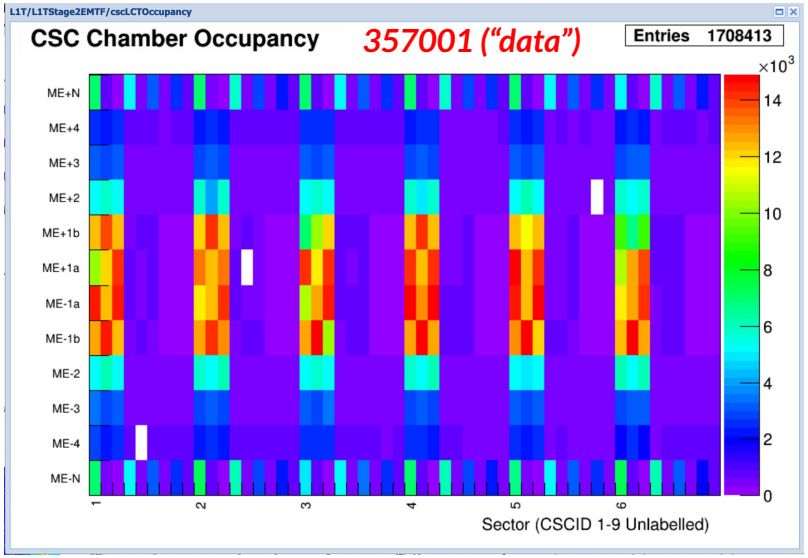


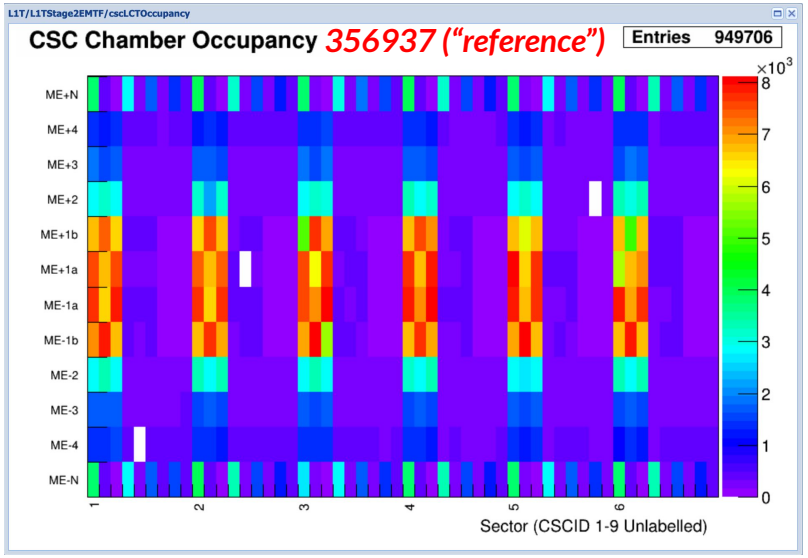


- Looking at CSC (L1T muon): 2D muon track "stub" occupancy
- Appear almost identical to shifters in the DQM GUI

Towards multiple run comparisons with βB

Use a set of 8 “good” reference runs

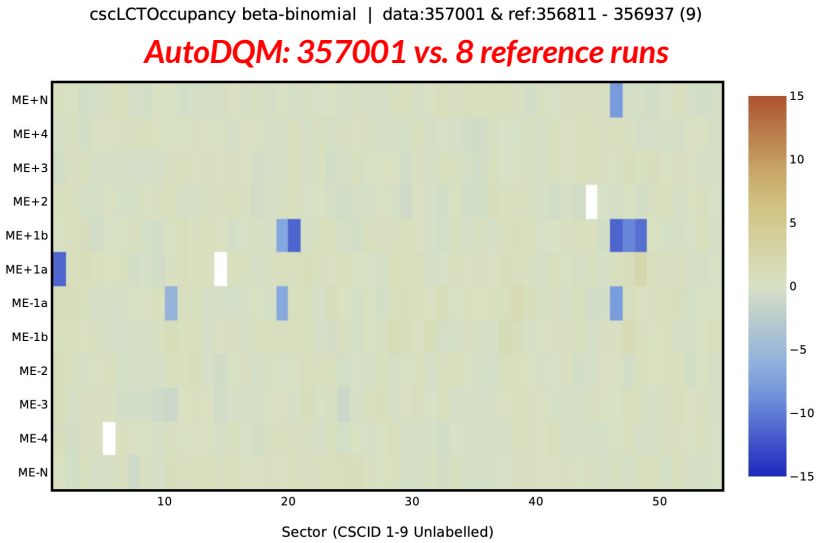




- Implementation of multiple references only in muon AutoDQM currently
- For each bin ***i***, βB computes probability p_i to observe d_i entries out of ***D*** total in the data histogram, given r_i out of ***R*** entries in the same bin of reference hist

Towards multiple run comparisons with βB

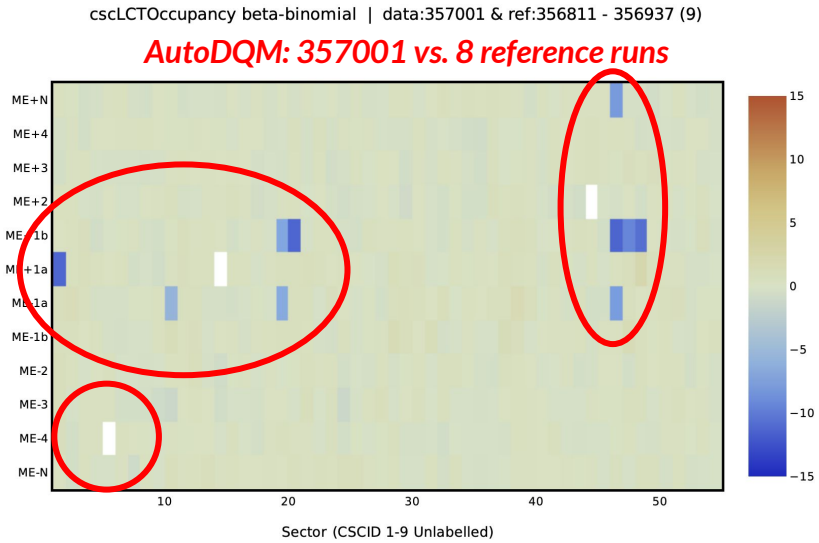
Take average p_i from all ref. comparisons



- Convert per-bin likelihood ratio p_i / $p(\max)_i$ to pull value σ_i in units of standard deviations
- "Anomalousness" measured with χ^2 (sum of σ_i^2 / # of bins) and maximum pull value $\max[\sigma_i]$ (corrected for look-elsewhere effect based on # of bins)

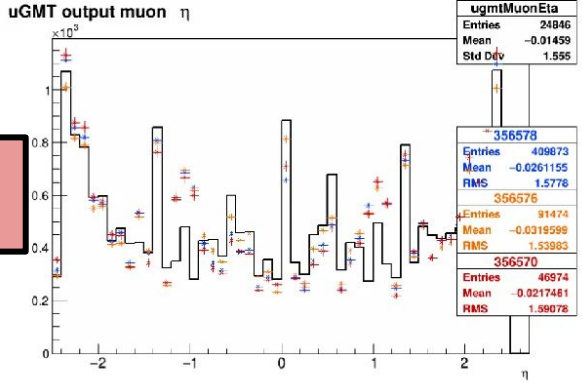
Towards multiple run comparisons with βB

There are real issues here

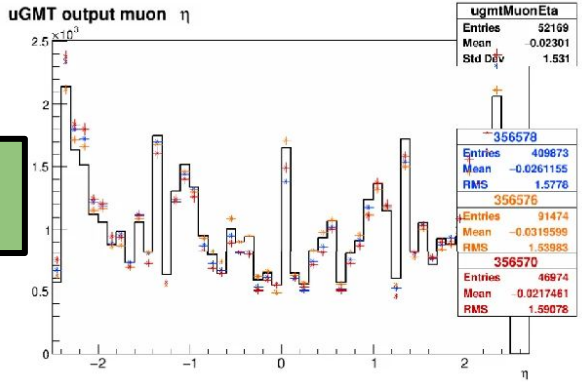


- Already AutoDQM has been used for L1T muon diagnostics

R# 356580

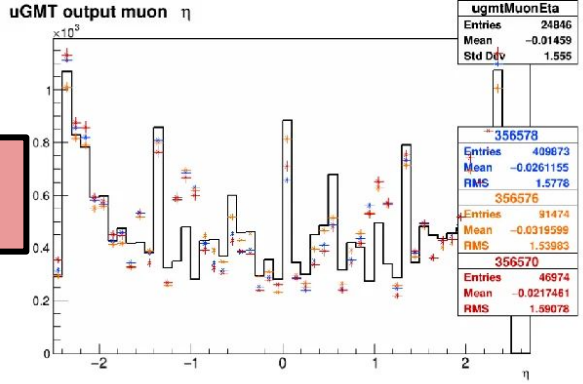


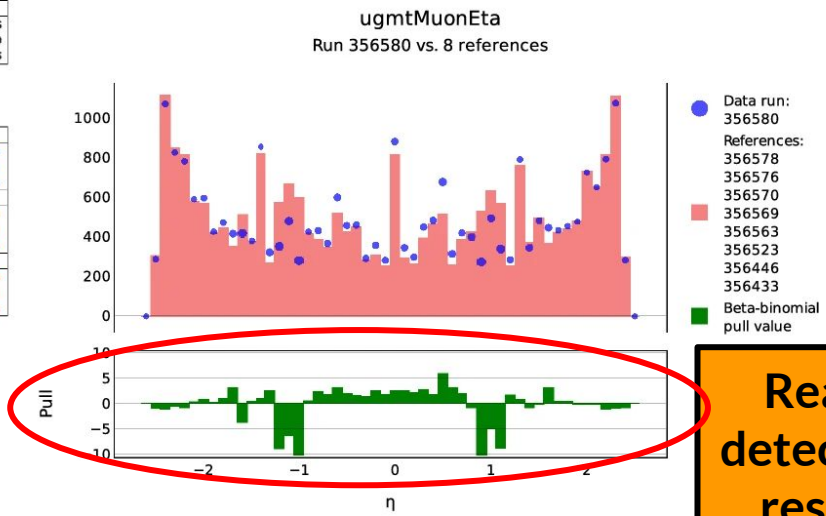
R# 356582



Reco muon track η in L1T
3 “good” ref. runs overlaid

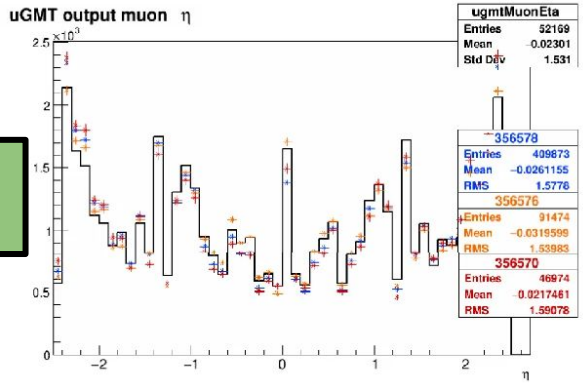
R# 356580

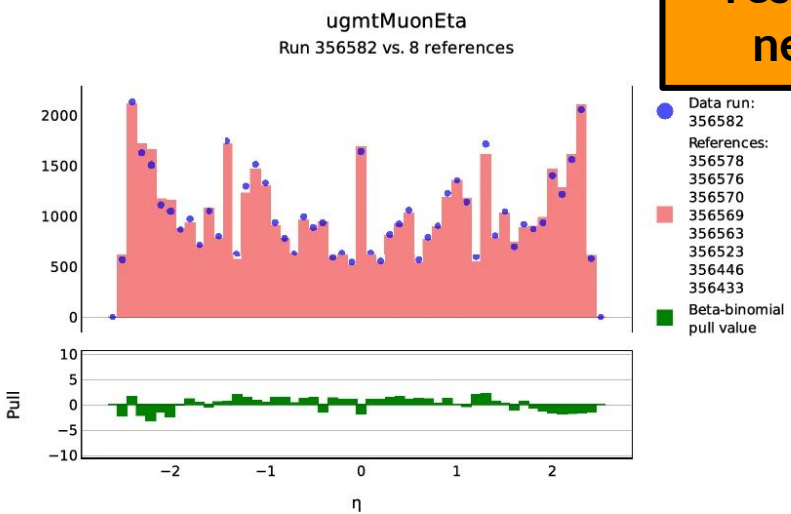


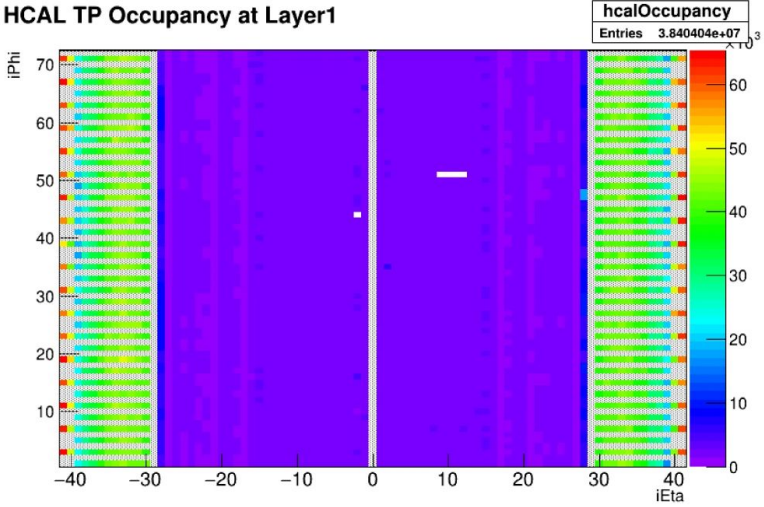


Real muon detector issue, resolved in next run

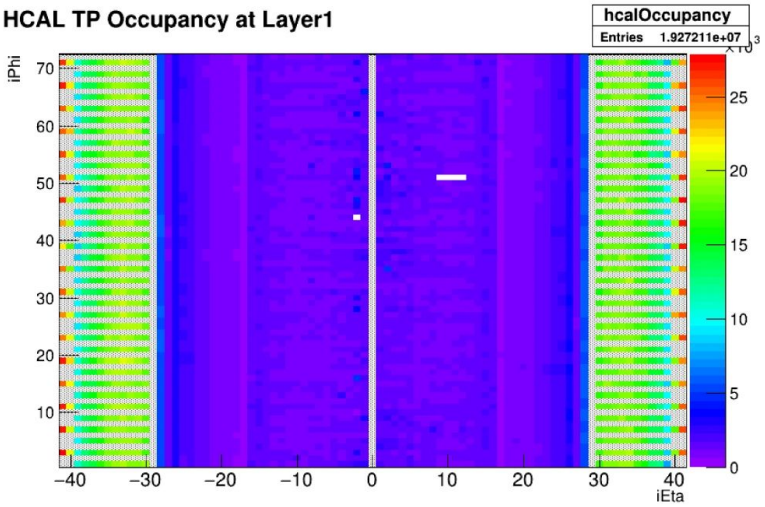
R# 356582





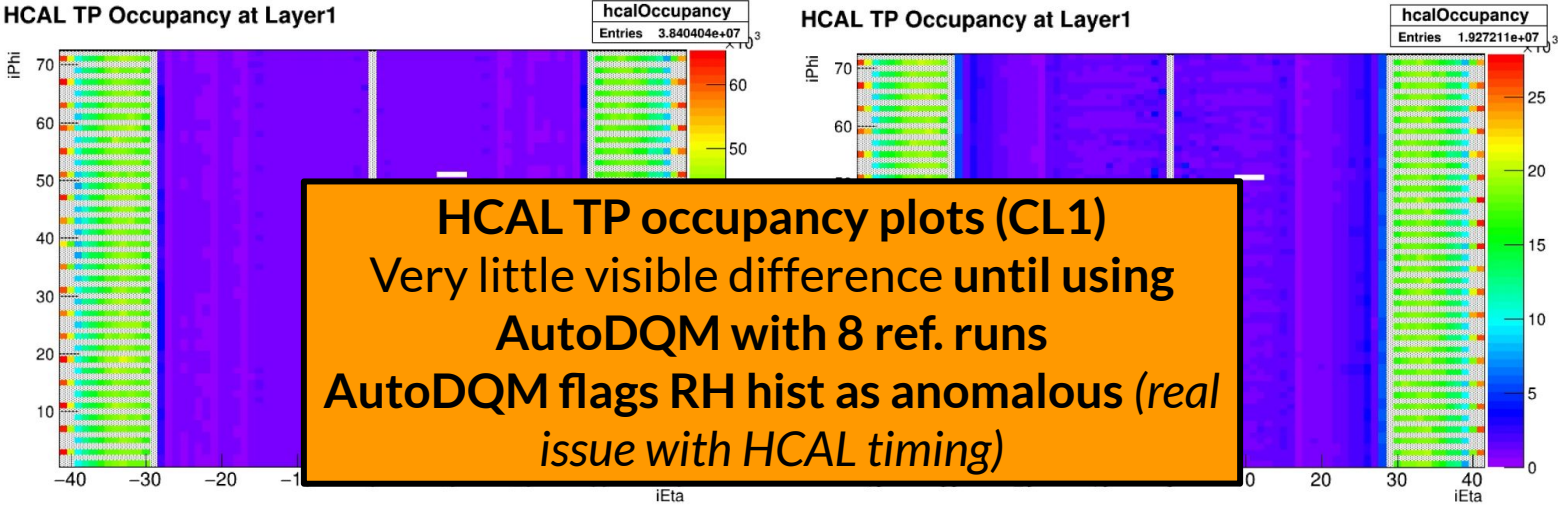


R# 357814

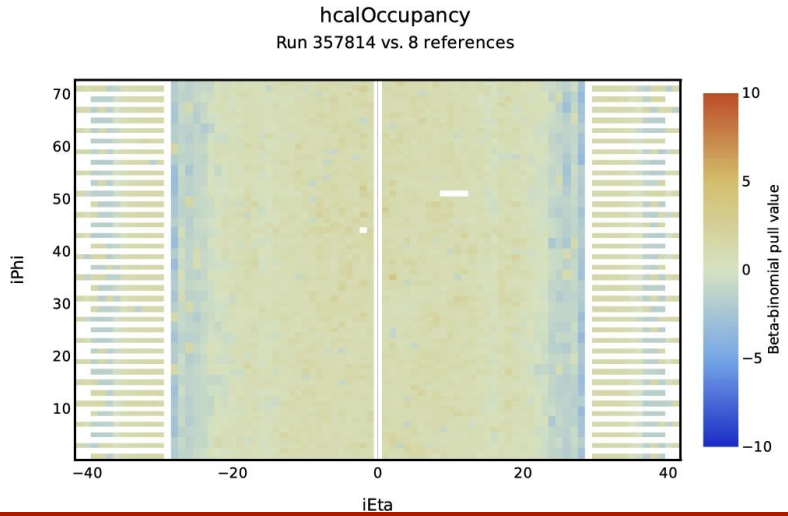


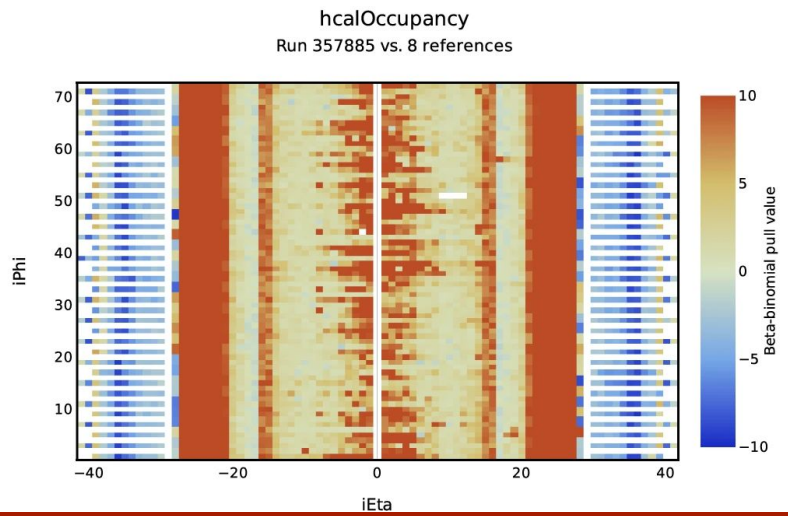
R# 357885

HCAL TP occupancy plots (CL1)
Very little visible difference…



HCAL TP occupancy plots (CL1)
Very little visible difference **until using AutoDQM with 8 ref. runs**
AutoDQM flags RH hist as anomalous *(real issue with HCAL timing)*





Public AutoDQM: current status

Successfully applied in muon detector monitoring

- There are 540 CSCs in the CMS endcaps (Run 3)
- Very rarely, ~10s of chambers temporarily malfunction
- AutoDQM shows geometric regions with low occupancies
- Across multiple reference runs, can identify these anomalies as new or long-running
- CSC experts can assess these issues and promptly intervene if necessary

Public AutoDQM: current status

Successfully applied in muon detector monitoring

- There are 540 CSCs in the CMS endcaps (Run 3)
- Very rarely, ~10s of chambers temporarily malfunction
- AutoDQM shows geometric regions with low occupancies
- Across multiple reference runs, can identify these anomalies as new or long-running
- CSC experts can assess these issues and promptly intervene if necessary

There are some dedicated automated DQM systems

- ECAL: *Autoencoder-based Anomaly Detection System for Online Data Quality Monitoring of the CMS Electromagnetic Calorimeter*
- HCAL: *Spatio-Temporal Anomaly Detection with Graph Networks for Data Quality Monitoring of the Hadron Calorimeter*

But AutoDQM is general purpose: it's very easy to add a module/subsystem

Deep Anomaly Detection

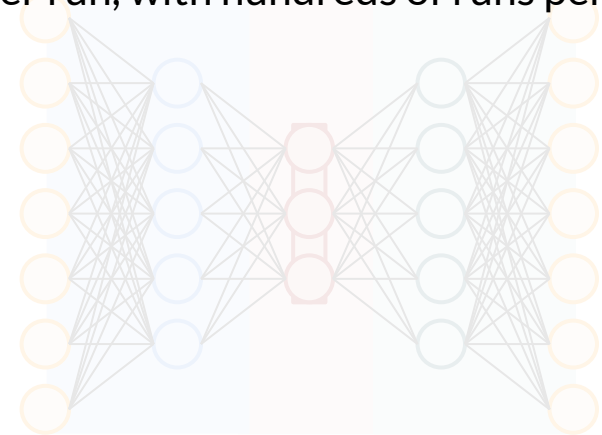
Shifters only have *another* tool to play with

AUTO DQM
A STATISTICAL TOOL

- Not all issues are so obvious: what if there are obscure anomalies invisible to our eyes?

Obvious choice is to use *machine learning*

- We have years and years of data, and hundreds of histograms per run, with hundreds of runs per year
- Plenty to train with!



Contents

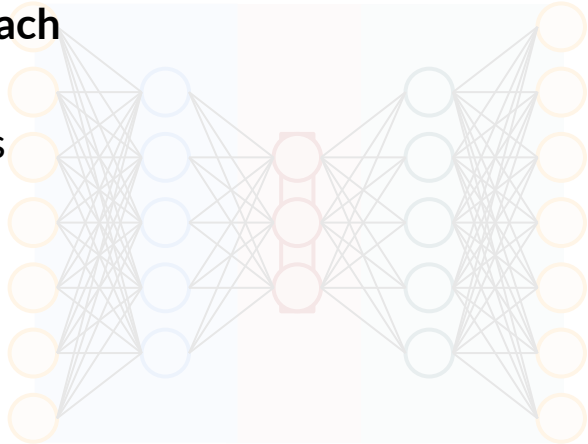
AutoDQM ML

Starting with the most simple anomaly detection algorithms

- Principal Component Analysis (PCA) and Autoencoder (AE) algorithms are readily available
  - Scikit-learn/TensorFlow implementation
- Can now look at sets of runs *en masse* (e.g. all of 2022) and large numbers of histograms

No longer need specific reference data with an unsupervised approach

- Bad data are rare and often uncorrelated across different issues
- Sufficient to train on a subset of good data



Formal construction

Collection of histograms from *good* runs converted into lower-dimensional *latent space*

- Learn a transformation $\mathcal{T}$ from the input space (i.e. entries per bin) to compress into latent space
- Latent space contains just enough features to reconstruct the original histogram
- Once $\mathcal{T}$ is learnt, the anomaly score for an input test histogram can be evaluated

All principles are the same for both the PCA and AE

- We performed meta studies using L1T branches of Run 2022 data
- Need to determine which runs to study, and which L1T histograms are most interesting

L1T Meta Study

Use 62 plots from L1T DQM, e.g. Calo Layer-1/-2, uGMT, L1T Objects

- Mostly L1T shifter histograms, with added subsystem-specific histograms
- Histograms from HLTPhysics data set in PromptReco files containing L1T object branches

Full run list for 2022 extracted from DCSOnly and Golden JSON files using BrilCalc

- All runs required to have **>5 mins** (**>13 LS**) and **3 pb^{-1}** of recorded data with **>500** colliding bunches (**>0.05 pb^{-1}/LS**) for sufficient statistics and detector performance

Label	GOOD (TRAIN)	BAD	GOOD (TEST)
LS criteria	>30 min (>77 LS)	> 40 pb^{-1} *bad* data	>5 mins (>13 LS)
PPD ***good*** data criteria	> 90% *good* data	or < 50% *bad* data	
# runs	216	43	265 [216+49]

L1T Meta Study

Use 62 plots from L1T DQM, e.g. Calo Layer-1/-2, uGMT, L1T Objects

- Mostly L1T shifter histograms with added subsystem-specific histograms
- Histograms from HLTPhysics dataset in Prompt

Full run list for 2022 extracted from DCSOnly and

- All runs required to have **>5 mins** (**>13 LS**) and
  bunches (**>0.05 pb^{-1}/LS**) for sufficient statistics and detector performance

Exclude runs labelled *bad* due to tracker issues, as this is invisible to L1T

Label	GOOD (TRAIN)	BAD	GOOD (TEST)
LS criteria	>30 min (>77 LS)	> 40 pb^{-1} *bad* data	>5 mins (>13 LS)
PPD ***good*** data criteria	> 90% *good* data	or < 50% *bad* data	
# runs	265	43	216

Unsupervised learning in PCAs

More info on AutoDQM's PCA implementation

PCA implemented using scikit-learn

- Unsupervised and specialised in dimensionality reduction (**compression**)
- Transforms 1D/2D histograms from 216 training runs into **two key components**

Unsupervised learning in PCAs

PCA implemented using scikit-learn

- Unsupervised and specialised in dimensionality reduction (**compression**)
- Transforms 1D/2D histograms from 216 training runs into **two key components**

Anomalies are rare: they will not be identified as key components

Tests show for our purposes only 2 components needed: no marked improvement with 3 or 4

More info on AutoDQM's PCA implementation

Unsupervised learning in PCAs

PCA implemented using scikit-learn

- Unsupervised and specialised in dimensionality reduction (**compression**)
- Transforms 1D/2D histograms from 216 training runs into **two key components**

Requires pre-processing of input data and processing of reconstruction for each histogram family

- 2D histograms are flattened into 1D for both training and evaluation

More info on AutoDQM's PCA implementation

Unsupervised learning in PCAs

PCA implemented using scikit-learn

- Unsupervised and specialised in dimensionality reduction (**compression**)
- Transforms 1D/2D histograms from 216 training runs into **two key components**

Requires pre-processing of input data and processing of reconstruction for each histogram family

- 2D histograms are flattened into 1D for both training and evaluation
- Low-occupancy bins are merged iteratively to reduce impact of statistical fluctuations
  - Can lead to a series of empty bins in the reconstruction
  - Require bins to have at least 0.33% of entries averaged over the full training data set for *that* family
  - Remove zero-occupancy bins at start and end of histogram data

More info on AutoDQM's PCA implementation

Unsupervised learning in PCAs

PCA implemented using scikit-learn

- Unsupervised and specialised in dimensionality reduction (**compression**)
- Transforms 1D/2D histograms from 216 training runs into **two key components**

Requires pre-processing of input data and processing of reconstruction for each histogram family

- 2D histograms are flattened into 1D for both training and evaluation
- Low-occupancy bins are merged iteratively to reduce impact of statistical fluctuations
  - Can lead to a series of empty bins in the reconstruction
  - Require bins to have at least 0.33% of entries averaged over the full training data set for *that* family
  - Remove zero-occupancy bins at start and end of histogram data
- Transformation from latent space to reconstruction uses RELU at final layer to remove -ve bins
  - Avoids non-physical reconstructions and biased weights in reconstruction stage

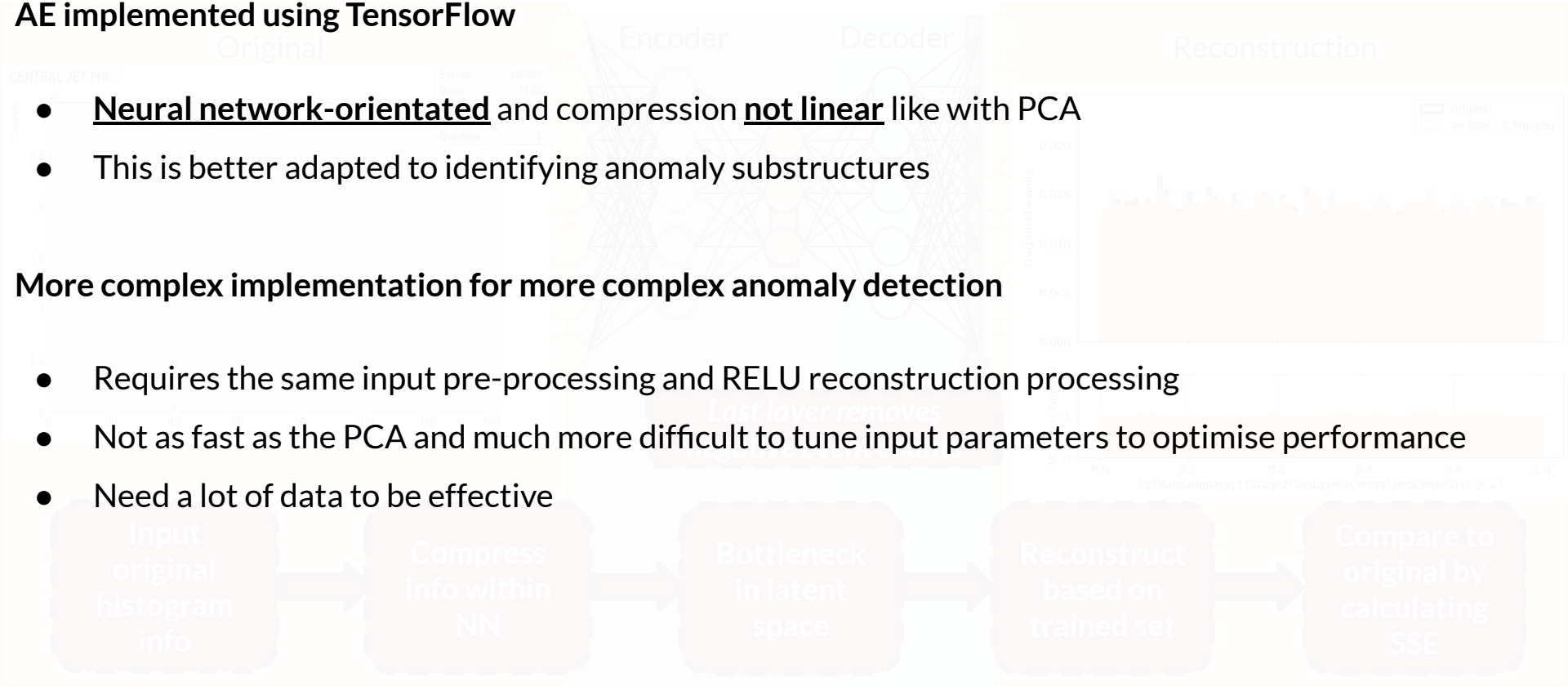
Unsupervised learning in AEs

AE implemented using TensorFlow

- **Neural network-orientated** and compression **not linear** like with PCA
- This is better adapted to identifying anomaly substructures

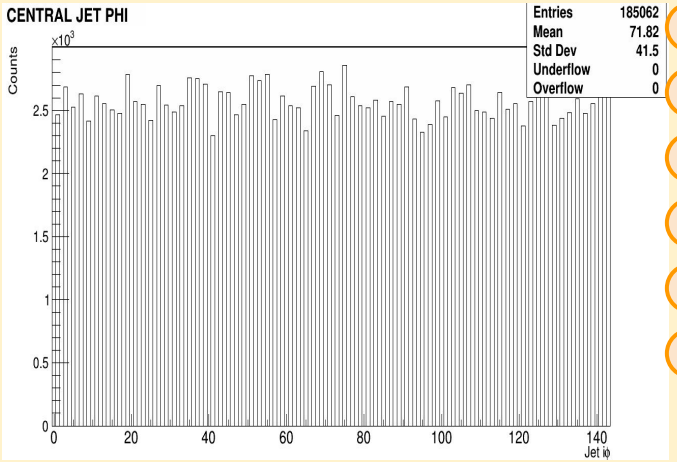
More complex implementation for more complex anomaly detection

- Requires the same input pre-processing and RELU reconstruction processing
- Not as fast as the PCA and much more difficult to tune input parameters to optimise performance
- Need a lot of data to be effective



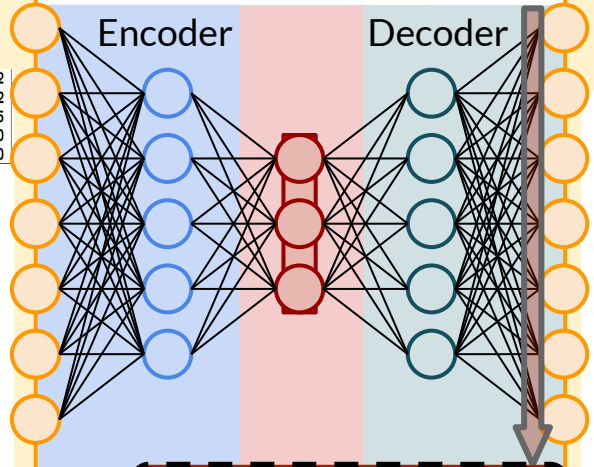
Unsupervised learning in AEs

Original



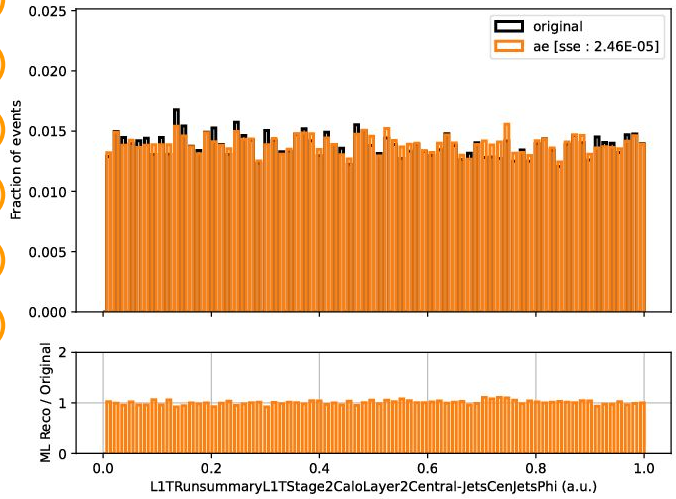
Encoder

Decoder



Last layer removes negative event counts

Reconstruction



Input original histogram info → **Compress info within NN** → **Bottleneck in latent space** → **Reconstruct based on trained set** → **Compare to original with an anomaly score**

Anomaly evaluation

Evaluate and flag if a histogram is anomalous if the anomaly score is above a certain threshold

- Anomaly score higher when the reconstruction is a poor match for the input histogram

Measure the Square Sum of Errors (SSE) score between original and reco

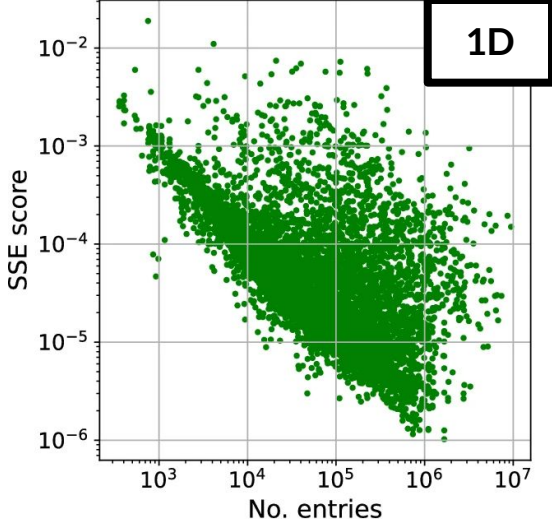
$$\mathrm{SSE} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (d_i' - d_i)^2$$

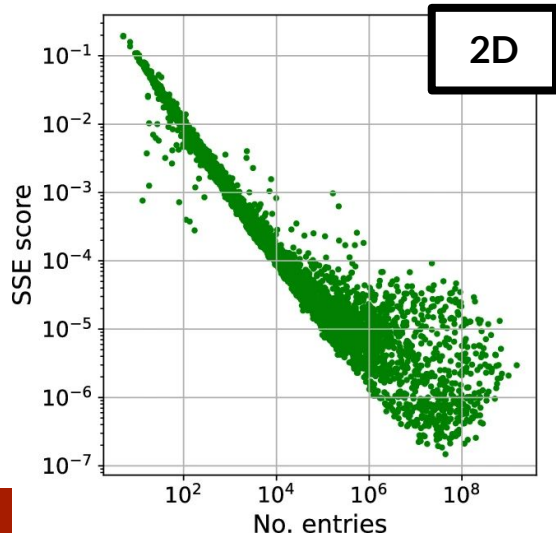
- d_i contents of bin i of normalised input histogram; d_i' reconstruction equal to $\mathcal{T}^{-1}(\mathcal{T}d)$
- Normalised SSE metric is anti-correlated with the number of entries in a histogram
- So shorter runs consistently result in higher anomaly scores

SSE score shortcomings

Looking at all histogram SSE scores using the PCA

- Anomaly score clearly anti-correlated with number of entries
- χ^2 intrinsically less susceptible to statistical fluctuations SSE
- Define a χ^2 metric driven by the βB probability function





SSE score shortcomings

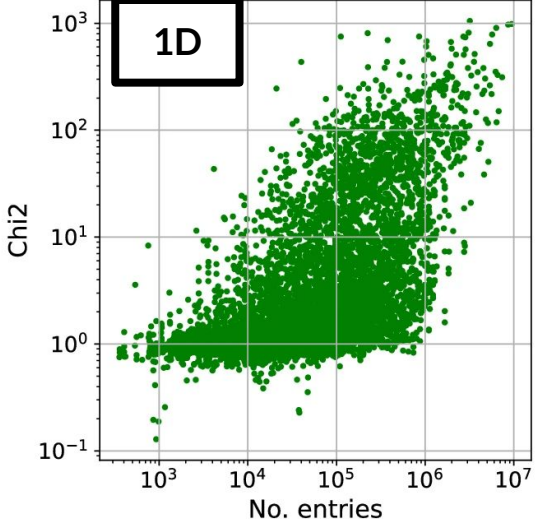
Looking at all histogram SSE scores using the PCA

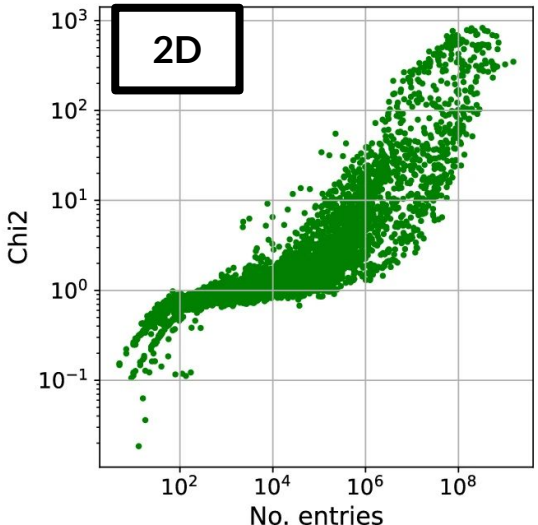
- Anomaly score clearly anti-correlated with number of entries
- χ^2 intrinsically less susceptible to statistical fluctuations SSE
- Define a χ^2 metric driven by the βB probability function

Take input histogram of bin occupancy d_i and integral D (unnormalised).
Take reconstructed histogram d_i' and D' scaled by 100 as reference histogram.

$$\alpha = \alpha_0 + d_i' \times 100$$
$$\beta = \beta_0 + d_i' \times 100$$
$$\alpha_0 = \beta_0 = 1$$

Factor of 100 suppresses statistical uncertainty in reconstruction.
Result is a χ^2 metric that is consistently low for histograms with few entries





SSE score shortcomings

Take input histogram of bin occupancy d_i and integral D (unnormalised).
Take reconstructed histogram d_i' and D' scaled by 100 as reference histogram.

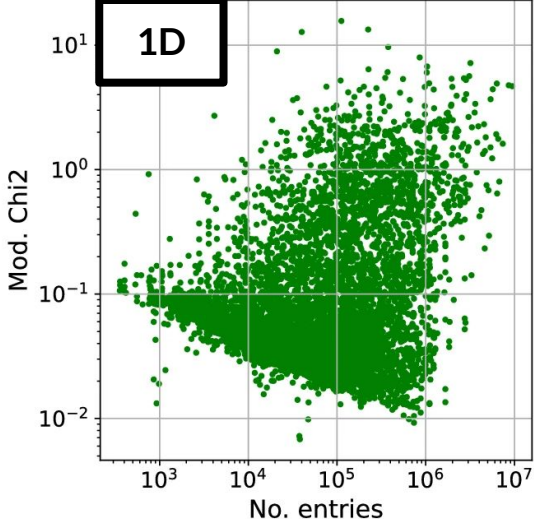
$$\alpha = \alpha_0 + d_i' \times 100$$
$$\beta = \beta_0 + d_i' \times 100$$
$$\alpha_0 = \beta_0 = 1$$

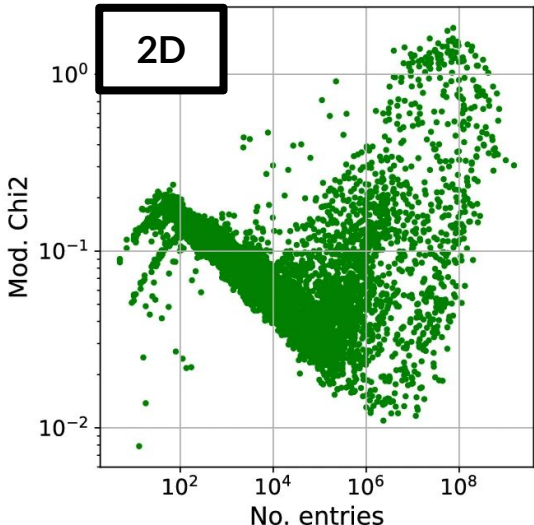
Factor of 100 suppresses statistical uncertainty in reconstruction.
Result is a χ^2 metric that is consistently low for histograms with few entries.

Mitigate occupancy bias by scaling by the integral D ~ gradient of RH plots.

$$\chi^{2'} = \chi^2 / D^{1/3}$$

This is the **modified χ^2** score.



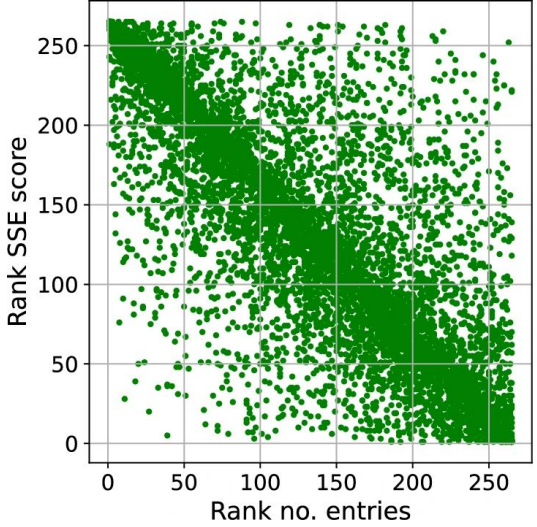


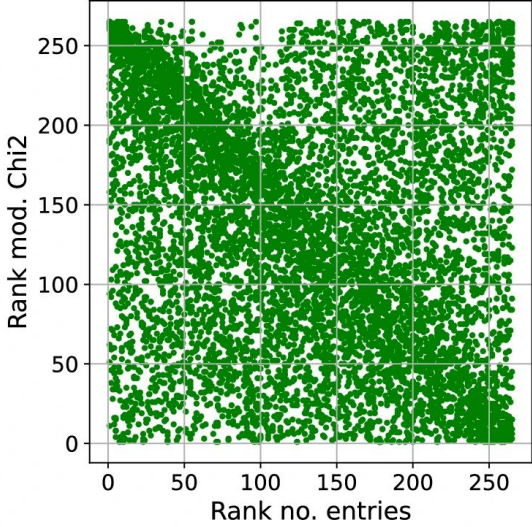
Further correlations

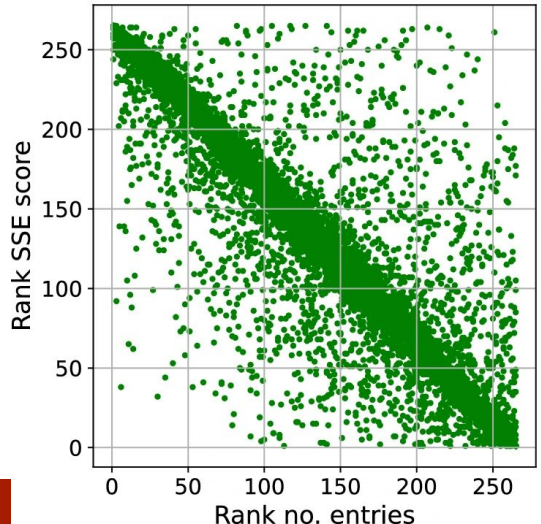
Ranks of scores against number of entries

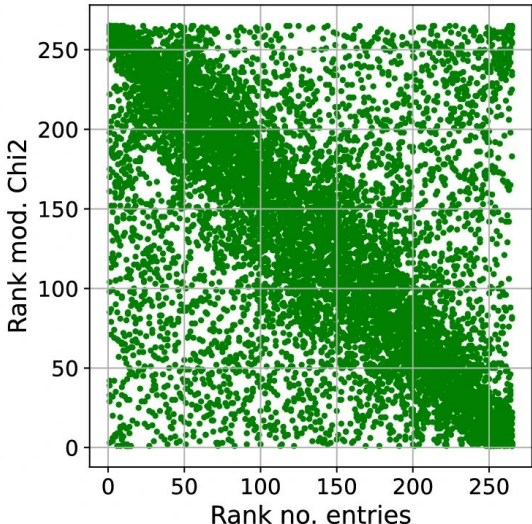
- (Top) Correlation between PCA-derived SSE and modified χ^2 score ranks and number of entries in 1D L1T DQM histograms in the 265 good runs
- (Bottom) Correlation in 2D L1T DQM histograms in the 265 good runs

Modified χ^2 correlation much lower than for SSE score, and give large values even in most-occupied plots

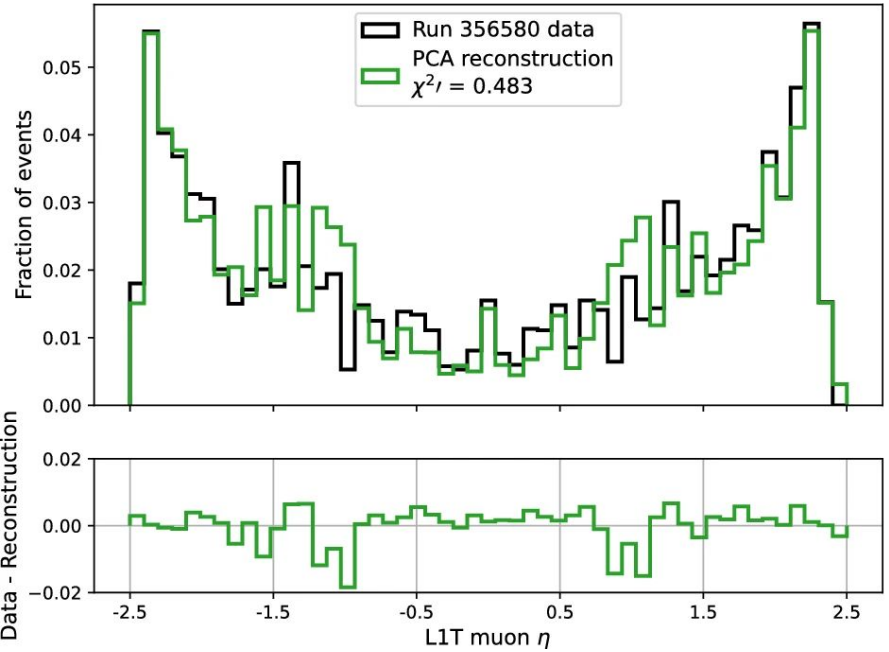


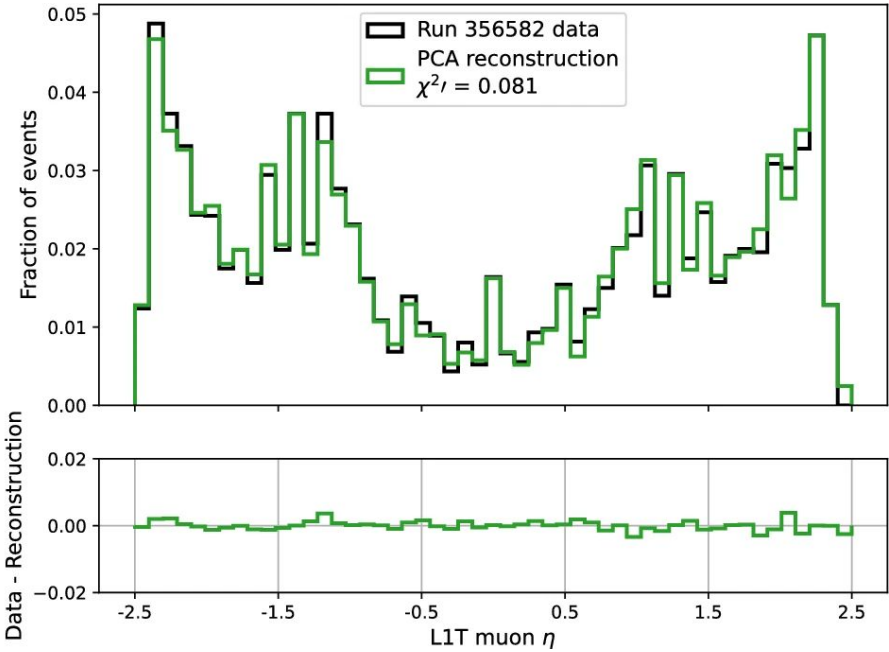






PCA performance

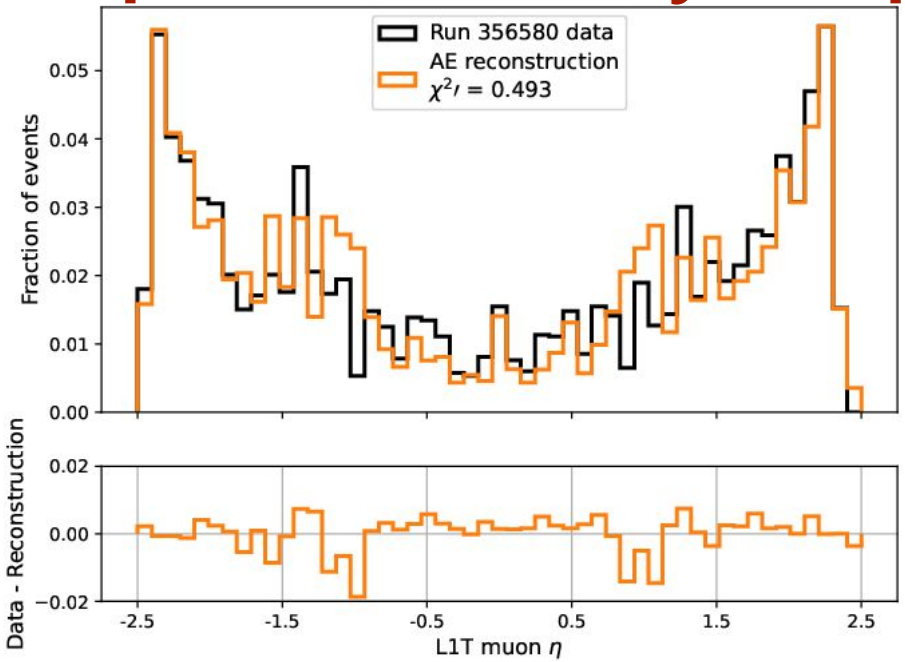


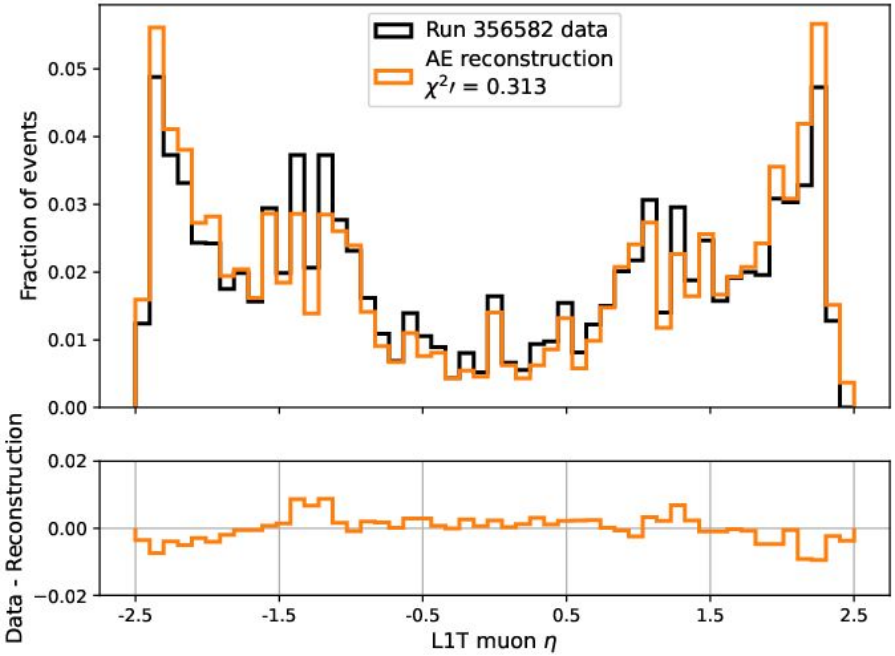


L1T muon track distributions in η (1D) should result in an anomaly flag on the left

- A deficit in tracks between $0.9 < |\eta| < 1.2$ is noted in Run 356580
- Difference in modified χ^2 and subtraction plot reflects this clearly

AE performance by comparison





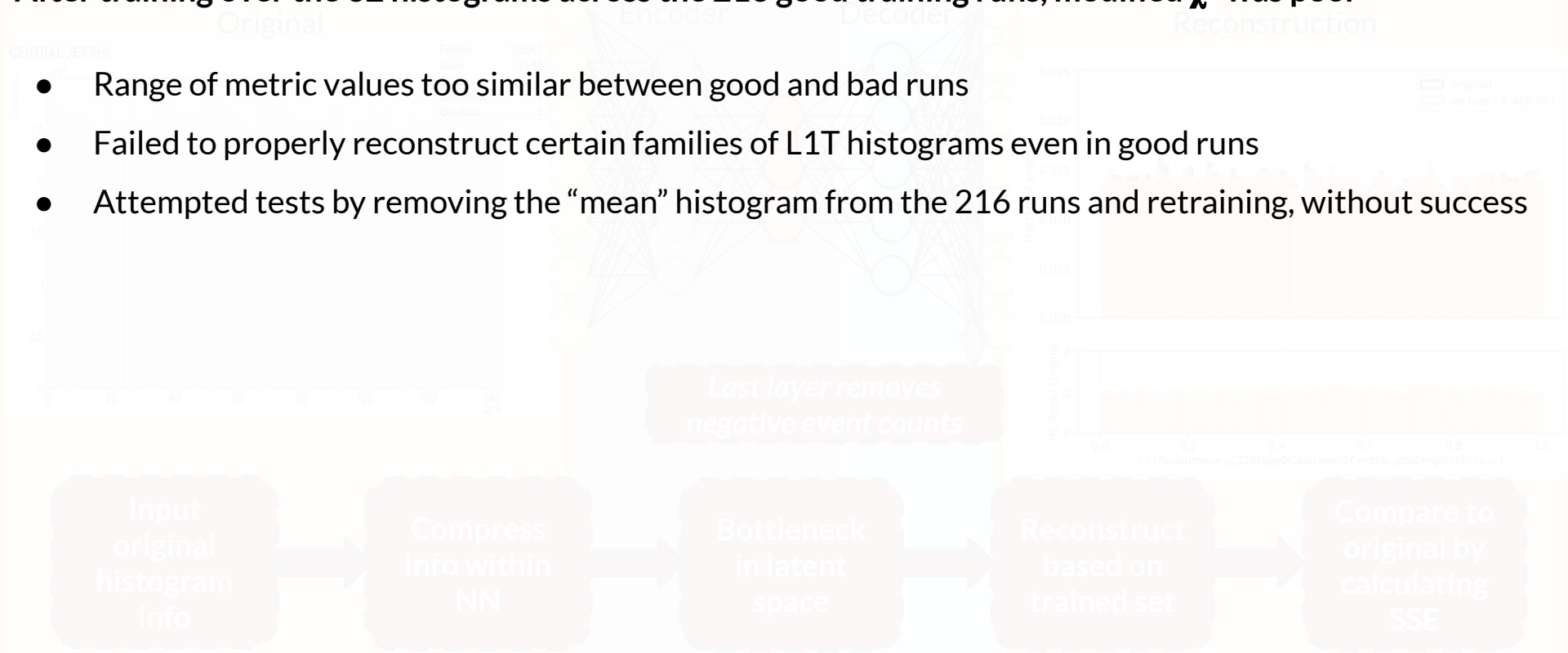
AE does correctly produce a higher anomaly score in Run 356580

- But the values are quite similar, and the reconstruction in Run 356582 is not ideal
- Actually signals the AE is not good with small variations between runs

Systemic problems with AE

After training over the 62 histograms across the 216 good training runs, modified χ^2 was poor

- Range of metric values too similar between good and bad runs
- Failed to properly reconstruct certain families of L1T histograms even in good runs
- Attempted tests by removing the “mean” histogram from the 216 runs and retraining, without success



Systemic problems with AE

After training over the 62 histograms across the 216 good training runs, modified χ^2 was poor

- Range of metric values too similar between good and bad runs
- Failed to properly reconstruct certain families of L1T histograms even in good runs
- Attempted tests by removing the “mean” histogram from the 216 runs and retraining, without success

Possible explanation is not that the AE was a bad algorithm, but its choice in this context was

- 62 histograms per run across 216 training runs
- Each histogram can have anywhere between ~100s of bins and ~10000s of bins (especially in 2D)
- Cannot reduce number of bins further: rebinning for occupancy at higher thresholds than 0.033% eventually leads to loss of histogram substructures
- AE cannot learn ~100s or ~1000s of features from a data set of 216 alone

Systemic problems with AE

AE performance only in arXiv up to simple original vs reconstruction comparison (2501.13789)

Journal version does not mention the autoencoder

(EPJ Research Infrastructures)

Results of full analysis available on request

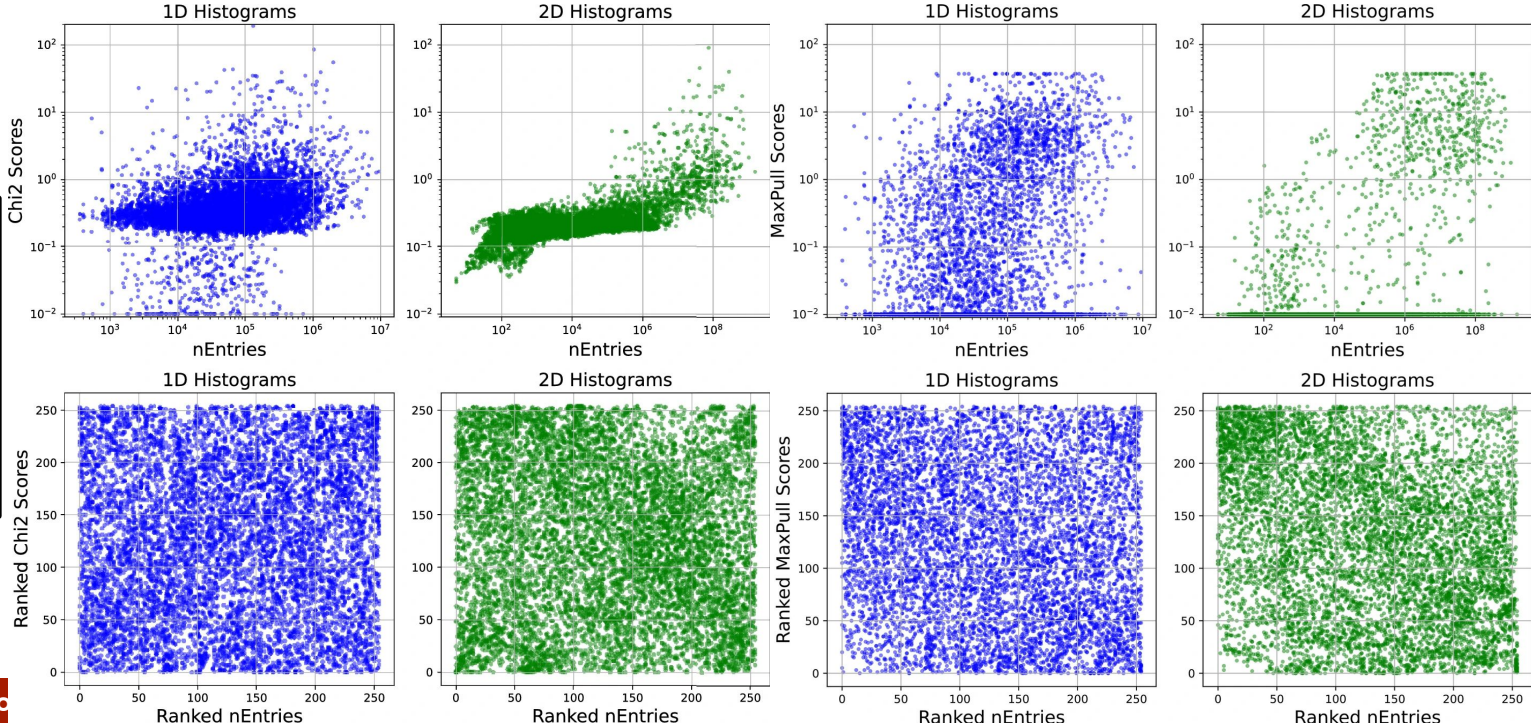
(I only continue with PCA and βB in these slides)

βB using 8 reference runs in same data set

Use βB to perform the same meta study but with appropriate anomaly scores

- Simple χ^2 score should inherently be unbiased by number of entries in each histogram
- Consider also the maximum single-bin pull value, Z'_{max}

No signs of correlation between histogram occupancy and anomaly scores, and good spread of values



L1T Meta Study

Used ~300 runs from 2022, selected 62 L1T histograms of interest per run

- 265 (~85%) good runs for general CMS use
- Anomaly scores for different algorithms:
  - PCA: modified χ^2
  - βB: χ^2 and maximum pull value, Z'_{max} (generated from 8 reference runs)

L1T Meta Study

Used ~300 runs from 2022, selected 62 L1T histograms of interest per run

- 265 (~85%) good runs for general CMS use
- Anomaly scores for different algorithms:
  - PCA: modified χ^2
  - βB: χ^2 and maximum pull value, Z'_{max} (generated from 8 reference runs)

Generate a set of flagging thresholds unique to each algorithm, score, and histogram

- Rank scores from good runs for each algorithm and histogram
- For tighter acceptance thresholds, a higher fraction of histograms will be flagged as anomalous

0^{th} threshold	**1^{st} highest score + ½ x (1^{st} - 2^{nd} highest score)**
1^{st} threshold	**½ x (1^{st} + 2^{nd} highest score)**
n^{th} threshold	**½ x (n^{th} + $(n+1)^{st}$ highest score)**

L1T Meta Study

Two new metrics to assess the AutoDQM anomaly-flagging performance

- Divided between good and bad runs (a priori labels)

1) **Mean # histogram flags (HF) per run**
   - Want to see more histogram flags in bad runs
   - Possible for histogram to be flagged by both algorithms
2) **Fraction of runs (RF) with at least *N* histogram flags**
   - Intended to reflect the shifter experience: *how often do we see a significant number of flags*?
   - *N* must be fairly small (1, 3, 5) to avoid alert fatigue (although performance for *N*>5 doesn't change)

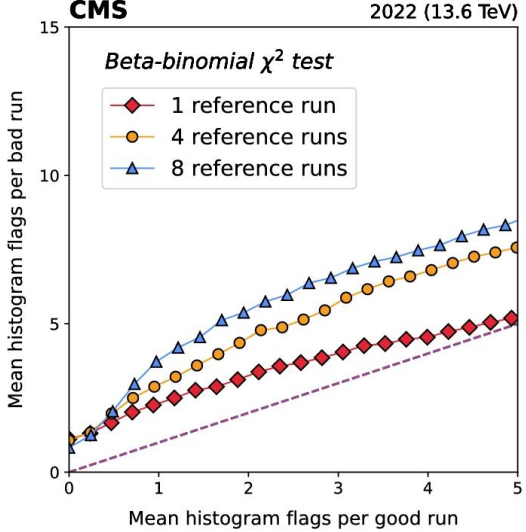
Construct ROC curves based on HF-ROC and RF-ROC metrics

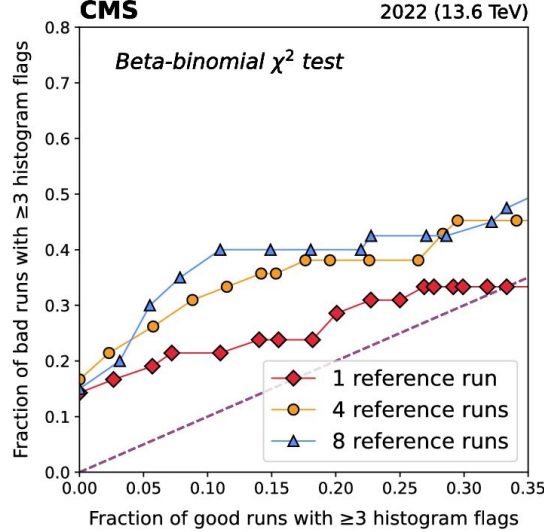
- Number of flags equal to number of anomaly scores exceeding the n^{th} threshold

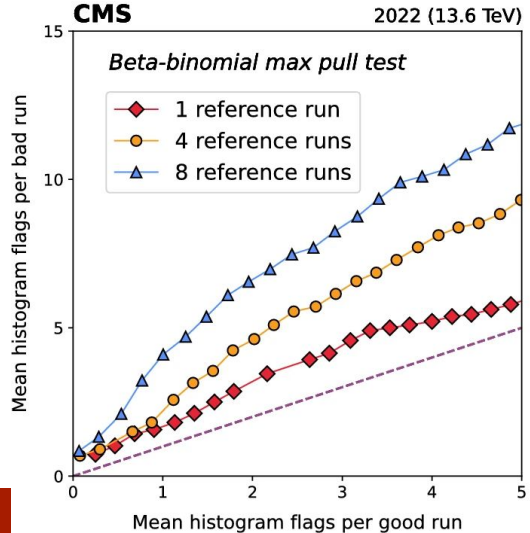
Performance of βB

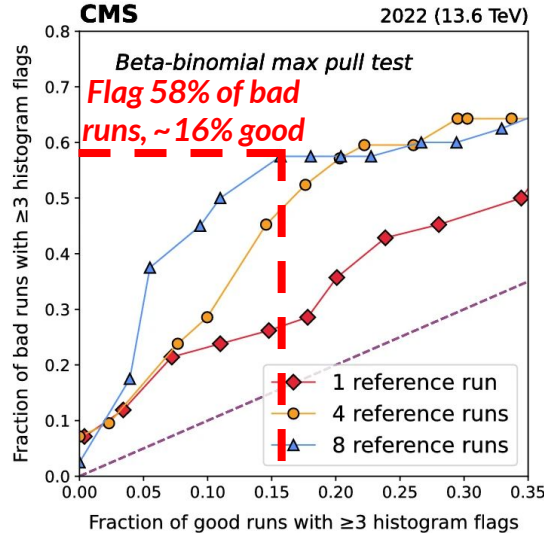
βB results compared to *M* reference runs

- Performances improve considerably for more reference runs
- HF-ROC and RF-ROC perform similarly
- Max pull test (Z'_{max}) slightly more sensitive to flagging histograms in good runs





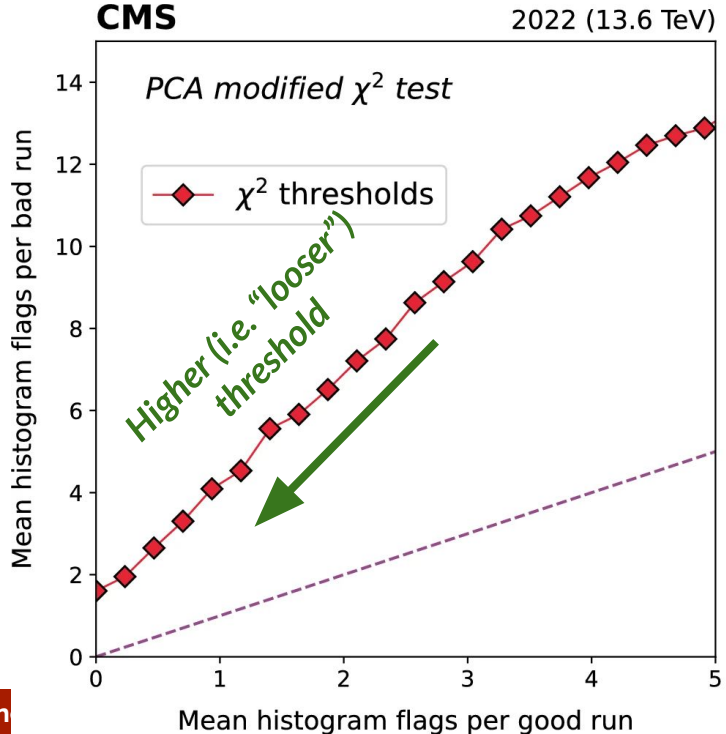


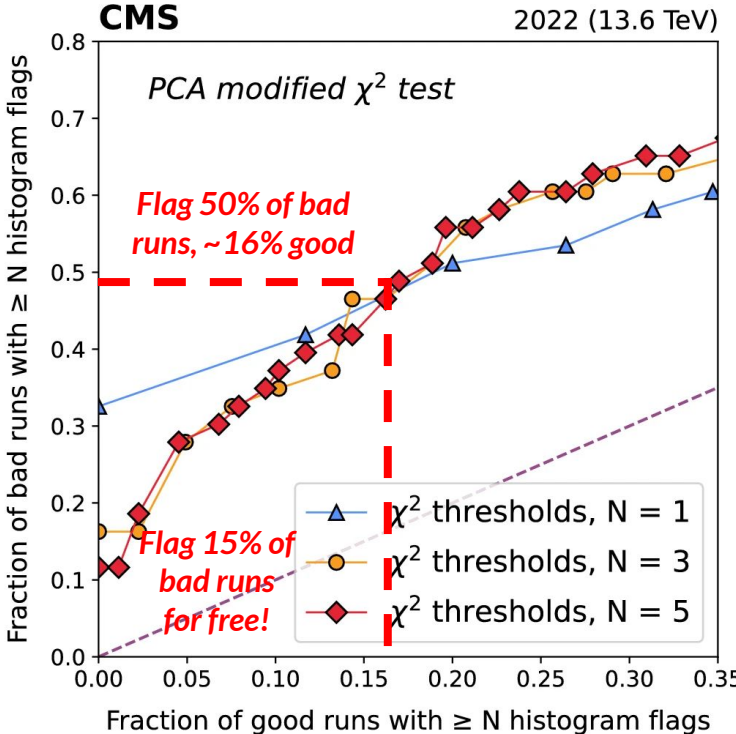


Performance of PCA

PCA results trained on all good *train* runs

- Seemingly very good at flagging anomalies in bad runs before being sensitive to good run flags
- Insensitive to variations in pileup conditions as PCA trained across the full 2022 set of runs





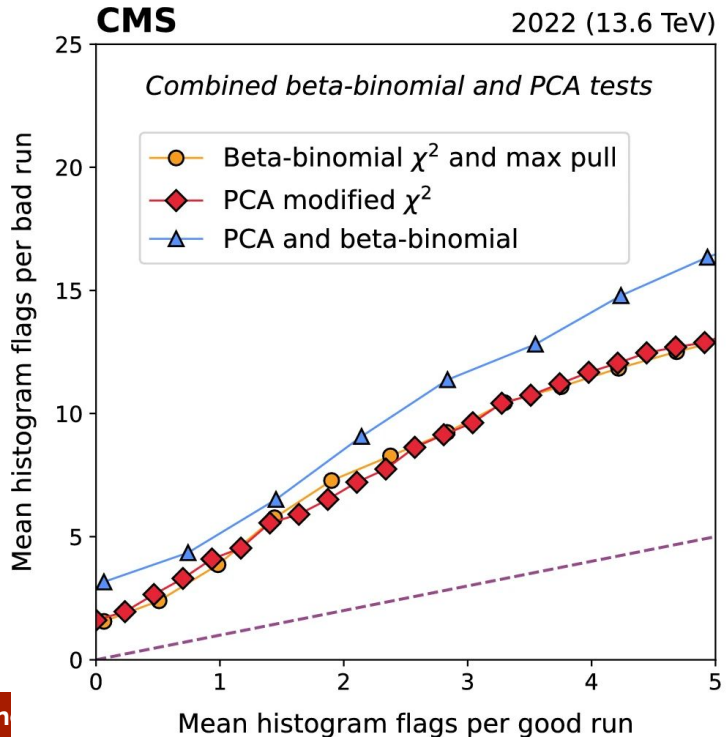
Performance of combined algorithms

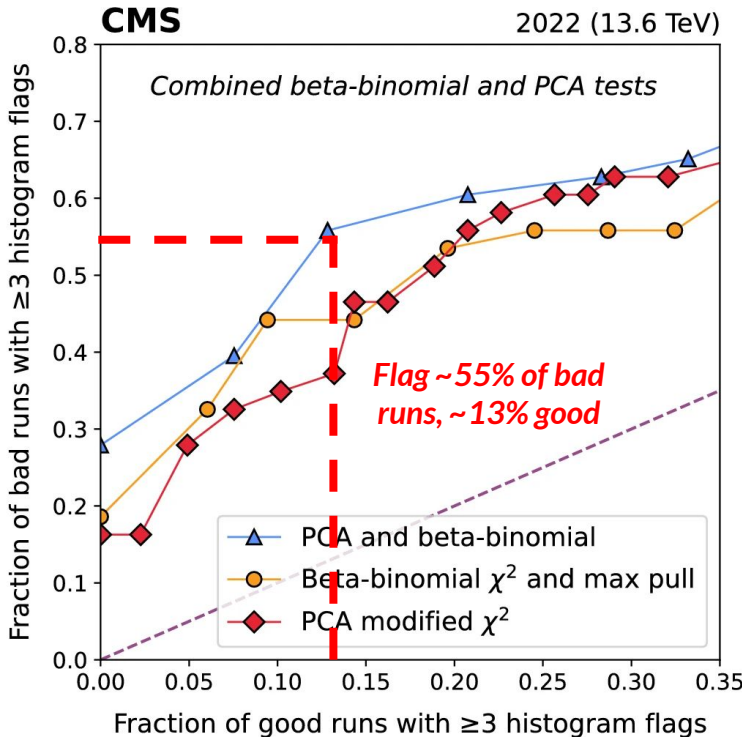
Best performance obtained from a combination of quality tests

- HF-ROC flags histograms 4-6 times more from bad runs than good runs
- **55% bad runs have at least 3 flags vs 13% good runs with at least 3 flags at same threshold**

No score or algorithm is definitively superior to the other

For the combined RF-ROC, if the same histogram is flagged by 2 tests, it is counted as 2 flags





Remarks on the Meta Study

AutoDQM is not expected to identify 100% of bad runs in the L1T data set

- Reason for a bad run may not have affected the L1T
- Issue not visible in the online DQM histograms

Nor is AutoDQM expected to never flag good runs as anomalous

- Many good runs contain true anomalies

Nevertheless, AutoDQM in L1T managed to detect half of the serious issues affecting CMS data in 2022, where less than 12% of good runs are flagged as anomalous

Contents

The Power of AutoDQM

From the POV of shifters and experts, the mean # HF and RF with at least 3 of 62 histograms flags will help us define thresholds for identifying anomalous runs in AutoDQM

For each run, the default AutoDQM display shows only the N flagged histograms

- The shifter can then study just those histograms
- Reduces the workload of the shifter up to ~4-6 times
- AutoDQM plots also highlight the source of the anomaly physically in the detector

The shifter can then contact experts to understand the anomaly

- For “online” shifters, allows us to catch and fix real problems / mis-configurations quickly
- For certification shifters, enable more accurate flagging of “good” and “bad” runs

Tool is most effective across a large set of reference runs, with PCA insensitive to conditions

- *Maybe* this helps create a full Run 3 anomaly detection tool

The Future of AutoDQM

Partly overlapping with the Phase II upgrade

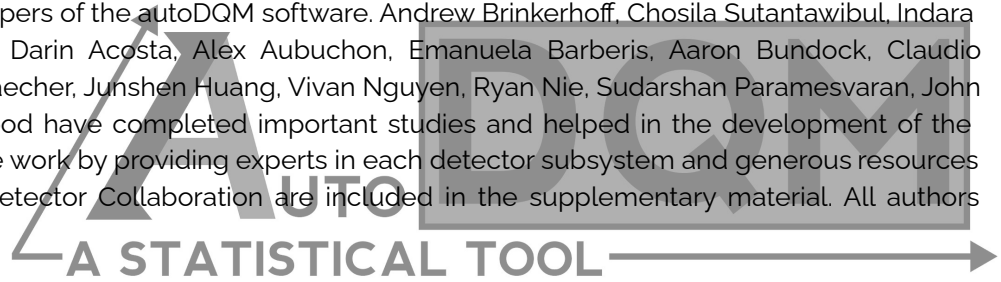
- Tighter integration with other monitoring tools (OMS, hardware status, etc.)
- Incorporation of at least some AutoDQM elements into central DQM
- Improved ML tools
- Develop GUI specifically for pedagogical effectiveness for shifters
  - Hover-over explanations
  - Links to information about the detector component, algorithms, etc.
- Extension to MC release validation (RelVals)
- Bringing in new Phase II detector components

Acknowledgements

We congratulate our colleagues in the CERN accelerator departments for the excellent performance of the LHC and thank the technical and administrative staffs at CERN and at other CMS institutes for their contributions to the success of the CMS effort. In addition, we gratefully acknowledge the computing centers and personnel of the Worldwide LHC Computing Grid and other centers for delivering so effectively the computing infrastructure essential to our analyses. Finally, we acknowledge the enduring support for the construction and operation of the LHC, the CMS detector, and the supporting computing infrastructure provided by the following funding agencies: SC (Armenia), BMBWF and FWF (Austria); FNRS and FWO (Belgium); CNPq, CAPES, FAPERJ, FAPERGS, and FAPESP (Brazil); MES and BNSF (Bulgaria); CERN; CAS, MoST, and NSFC (China); MINCIENCIAS (Colombia); MSES and CSF (Croatia); RIF (Cyprus); SENESCYT (Ecuador); ERC PRG, RVTT3 and MoER TK202 (Estonia); Academy of Finland, MEC, and HIP (Finland); CEA and CNRS/IN2P3 (France); SRNSF (Georgia); BMBF, DFG, and HGF (Germany); GSRI (Greece); NKFIH (Hungary); DAE and DST (India); IPM (Iran); SFI (Ireland); INFN (Italy); MSIP and NRF (Republic of Korea); MES (Latvia); LMTLT (Lithuania); MOE and UM (Malaysia); BUAP, CINVESTAV, CONACYT, LNS, SEP, and UASLP-FAI (Mexico); MOS (Montenegro); MBIE (New Zealand); PAEC (Pakistan); MES and NSC (Poland); FCT (Portugal); MESTD (Serbia); MCIN/AEI and PCTI (Spain); MOSTR (Sri Lanka); Swiss Funding Agencies (Switzerland); MST (Taipei); MHESI and NSTDA (Thailand); TUBITAK and TENMAK (Turkey); NASU (Ukraine); STFC (United Kingdom); DOE and NSF (USA).

Author Contributions

This project is a collaborative project, supported by many members of the CMS Collaboration over the years. This effort was coordinated by Andrew Brinkerhoff, Indara Suarez, Chad Freer, and Samuel May. Andrew Brinkerhoff, Chosila Sutantawibul, Indara Suarez, Robert White, Caio Daumann, Jonathan Guiang, Chad Freer, Samuel May, and Bennett Marsh are the main developers of the autoDQM software. Andrew Brinkerhoff, Chosila Sutantawibul, Indara Suarez, Robert White, and Caio Daumann compiled the manuscript. Darin Acosta, Alex Aubuchon, Emanuela Barberis, Aaron Bundock, Claudio Campagnari, Evan Collins, Preston Epps, Johannes Erdmann, Henning Flaecher, Junshen Huang, Vivan Nguyen, Ryan Nie, Sudarshan Paramesvaran, John Rotter, Kaitlin Salyer, Siddhesh Sawant, Tanvi Sheokand, and Darien Wood have completed important studies and helped in the development of the autoDQM software. The CMS Muon Detector Collaboration supported the work by providing experts in each detector subsystem and generous resources for the development. The names of all members of the CMS Muon Detector Collaboration are included in the supplementary material. All authors reviewed the manuscript.



BACKUP

Backup

2022: Data certification reports 9th, 16th, 23rd Aug 2022 (Set 1, Set 2, Set 3)

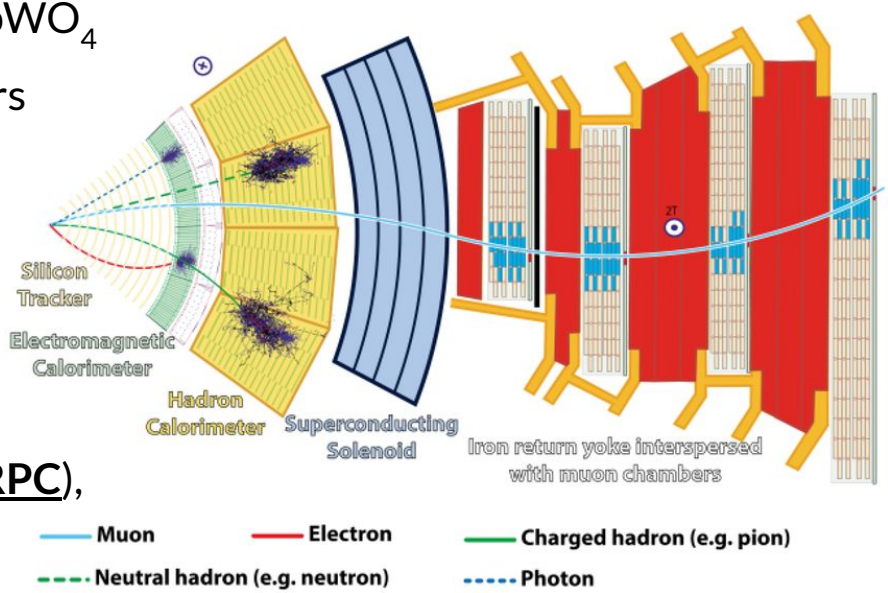
Certification Run3: https://twiki.cern.ch/twiki/bin/viewauth/CMS/CertificationOfCollisions22

Anomaly Detection Tools in CMS for DQM

- AutoDQM: Anomaly Detection for Automated Data Quality Monitoring in the CMS Detector
- ECAL: *Autoencoder-based Anomaly Detection System for Online Data Quality Monitoring of the CMS Electromagnetic Calorimeter*
- HCAL: *Spatio-Temporal Anomaly Detection with Graph Networks for Data Quality Monitoring of the Hadron Calorimeter*

The CMS Detector Layers

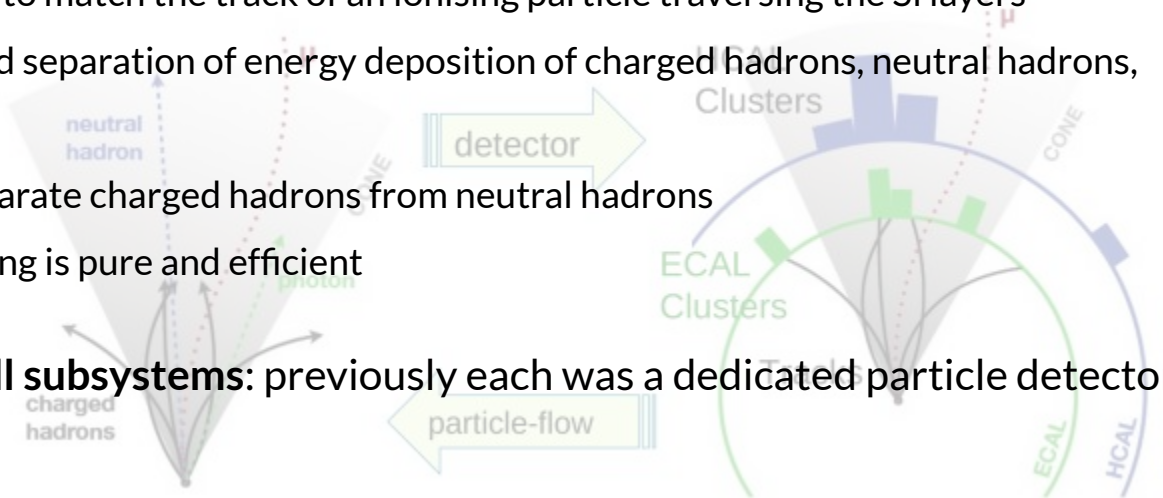
- Pixel and strip **Si tracker** strips for charged-particle detection
- Electromagnetic calorimeter (**ECAL**) formed of $PbWO_4$ crystals for detecting photon and electron showers
- Hadron calorimeter (**HCAL**) formed of brass and plastic scintillators to detect neutral and charged particles
- Surrounded by NbTi superconducting **solenoid**
- **Muon** drift tubes (**DT**), resistive plate chambers (**RPC**), gas electron multipliers (**GEM**) and cathode strip chambers (**CSC**) for tracking



Particle identification and location in L1T

We need to reconstruct jets and match them to particles

- Electrons, muons, tau leptons, photons, and hadrons identified using a global particle-flow algorithm (**PFA**)
- PFA relies on information across all detector subsystems to reconstruct jets
  - Based on a segmented tracker to match the track of an ionising particle traversing the Si layers
  - Fine-granularity ECAL for good separation of energy deposition of charged hadrons, neutral hadrons, and photons in jets
  - Coarse, hermetic HCAL to separate charged hadrons from neutral hadrons
  - Muon identification and tracking is pure and efficient



PFA integrates information across all subsystems: previously each was a dedicated particle detector

Matching particles to one event in L1T

We cannot reconstruct ~50-60 events per crossing: we only need one of interest

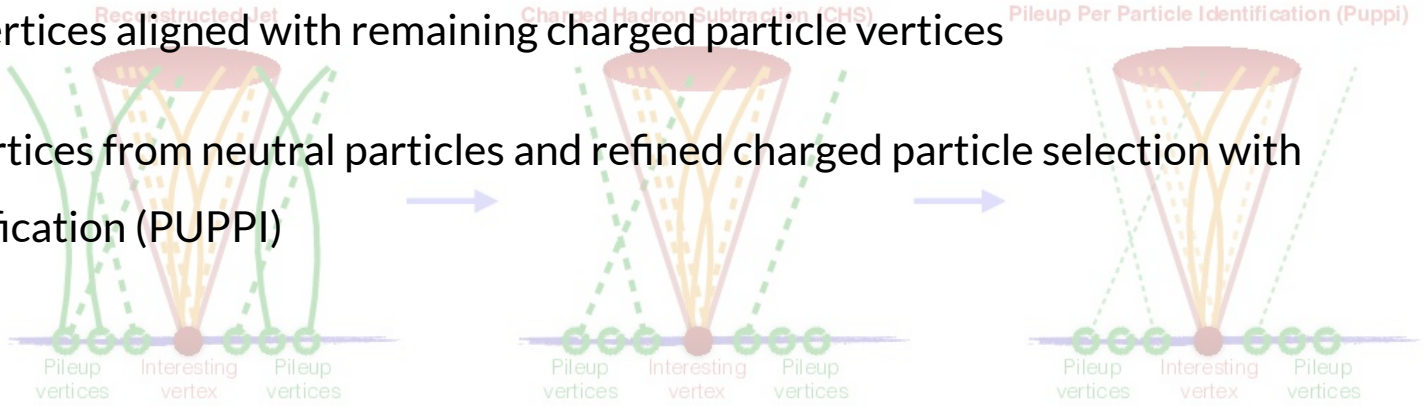
-> Find the primary vertex of the *hardest scattering* process (HSP) based on Si tracker info

-> Find *jets* (gluons, quarks that *hadronise* and possibly decay further) belonging to the HSP

-> Subtract charged hadrons (CHS) from vertices that do not correspond to the HSP

-> Find neutral particle vertices aligned with remaining charged particle vertices

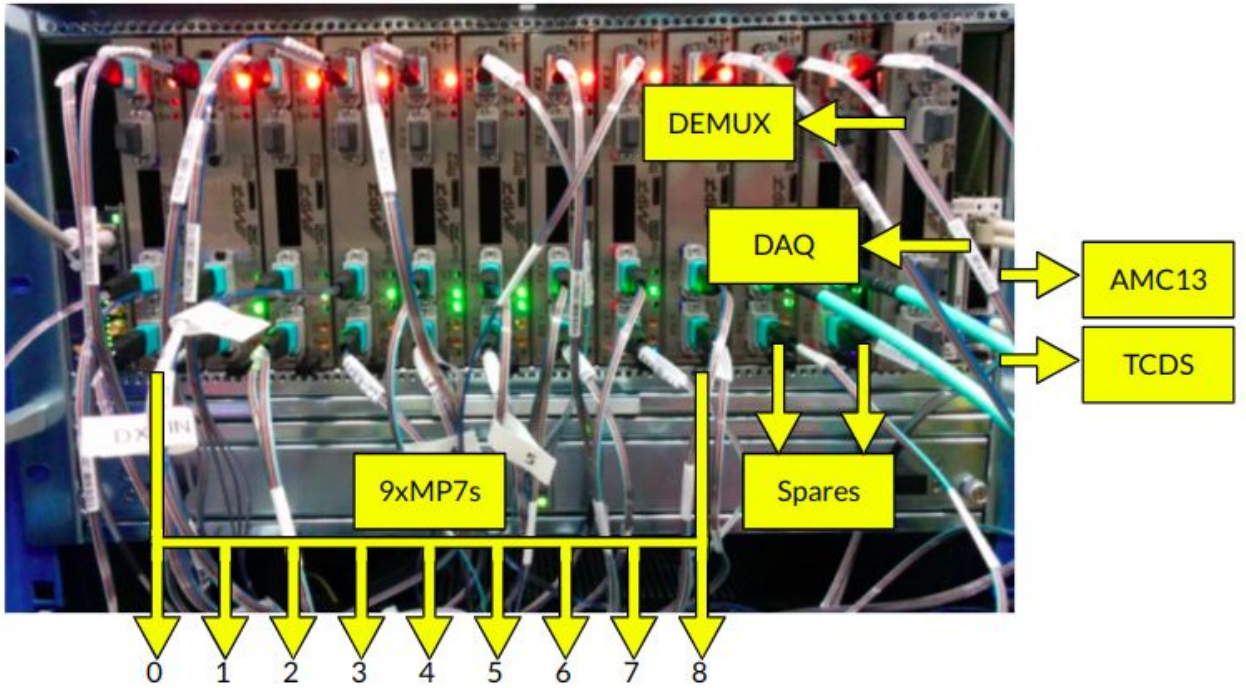
-> Filter out unrelated vertices from neutral particles and refined charged particle selection with pileup per-particle identification (PUPPI)



CL2 crate

Micro Telecoms Computing Architecture (MicroTCA) crate

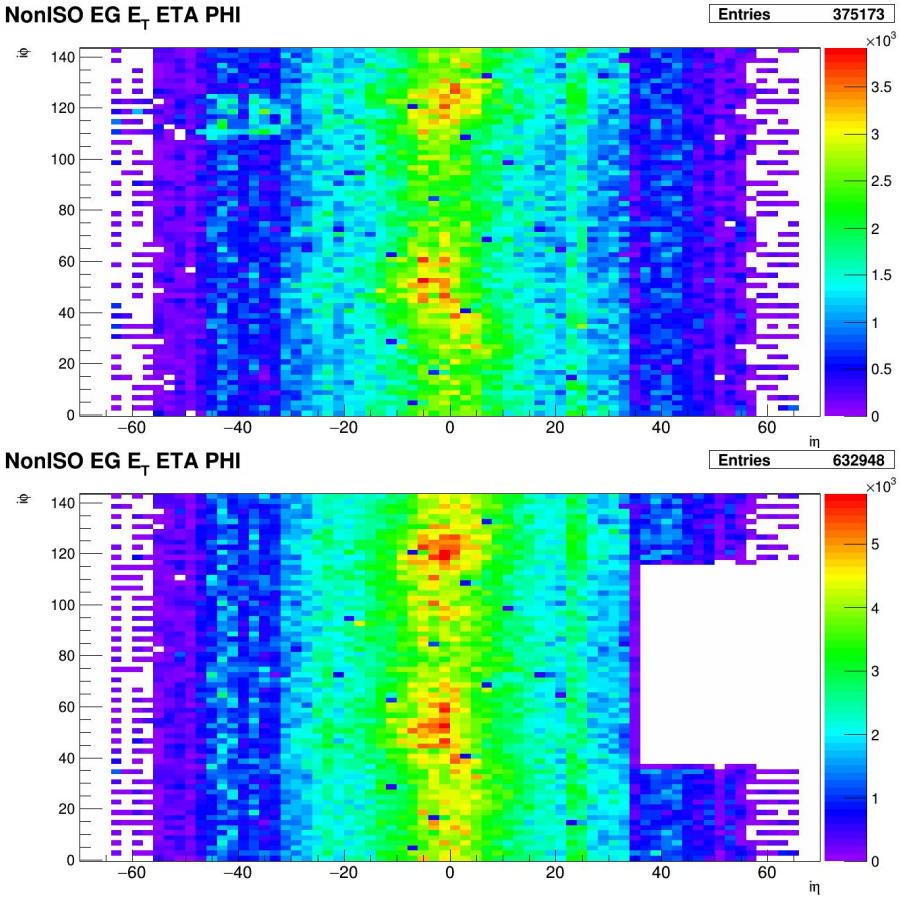
- 9 master processors (MP7)
- Each reads a 25 ns bunch crossing at a time
- Info is *time-multiplexed*: calo towers across all calo subsystems from CL1 timestamped and made available



Towers clustered into calo objects

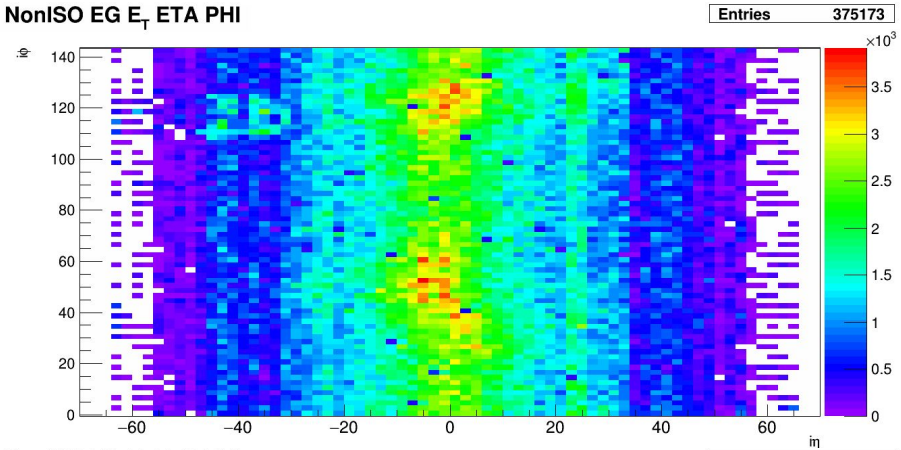
- Calo objects held on single FPGAs within MP7
- DEMUX formats processed data from MP7s for GT
- TCDS, AMC13 apply clock and signal timestamps

A very blatant example



- Run2018 MET comparison between runs 325170 and 316944
  - 325170 as “good” ref run
  - 316944 “bad”: loss of eff due to missing feds in EE+

A very blatant example



You wouldn't actually need AutoDQM to know that modules were disabled

- Run2018 MET comparison between runs 325170 and 316944
  - 325170 as "good" ref run
  - 316944 "bad": loss of eff due to missing feds in EE+