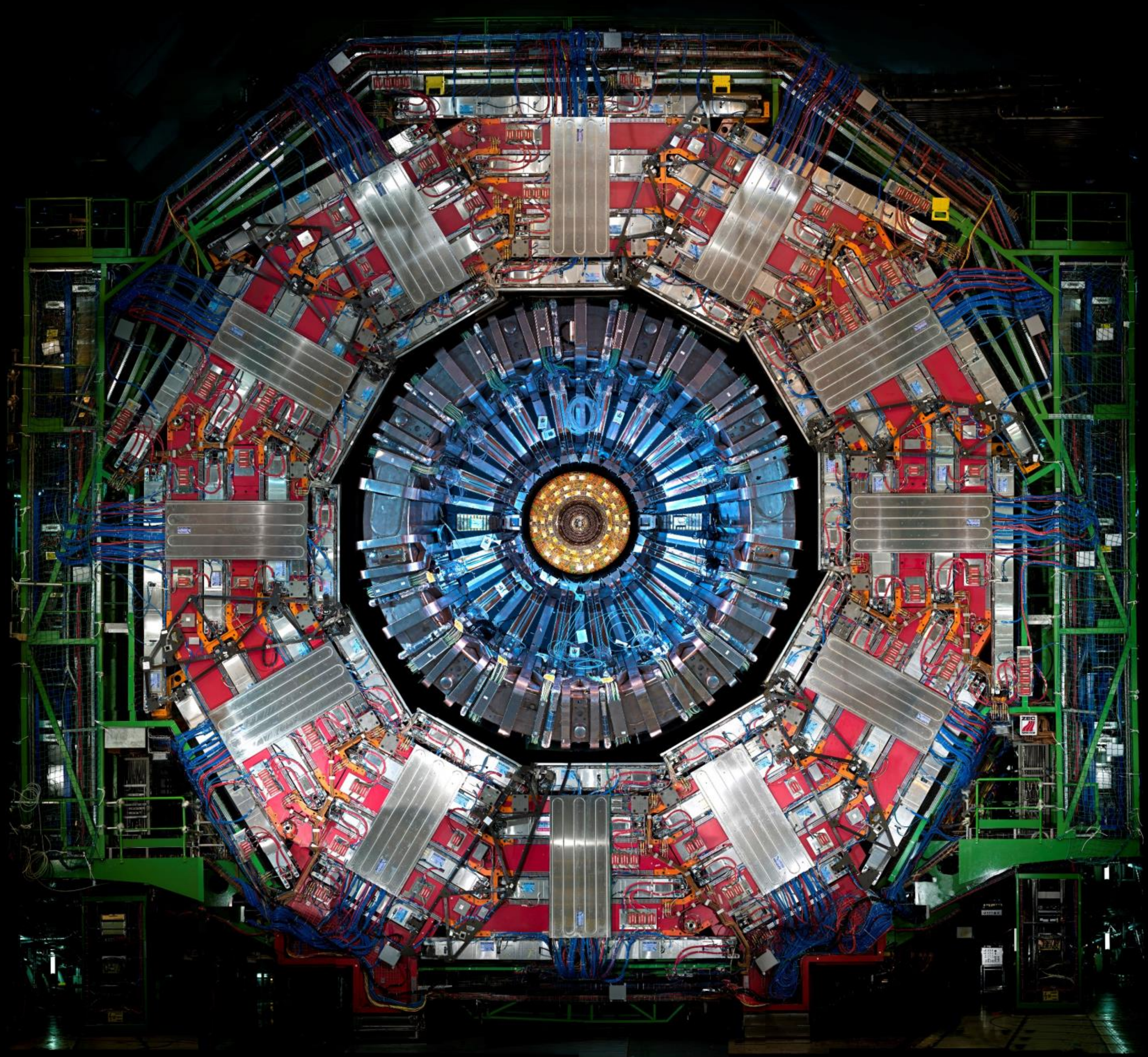
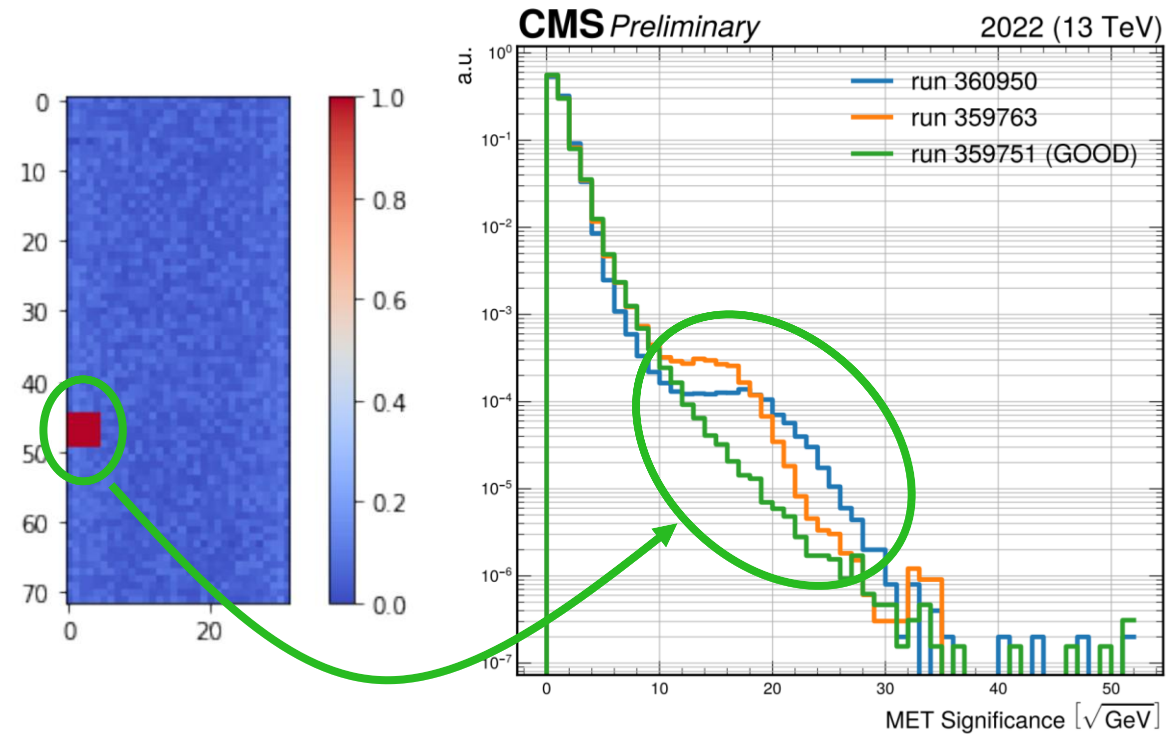


ML for data quality monitoring

Roberto Seidita – IPA ML workshop 2023, ETH Zürich

Data quality monitornig in CMS

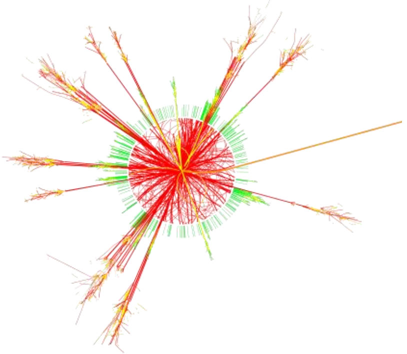
- CMS is a complex machine made of hundreds of systems working in unison to record collision events → things can (and do) go wrong
- Data quality monitoring (DQM) is the set of procedures by which we try to ensure our datasets are as free of issues as possible and thus can be used to make measurements
- This happens at many levels:
  - During data taking
  - After events are reconstructed
  - When calibrations are updated

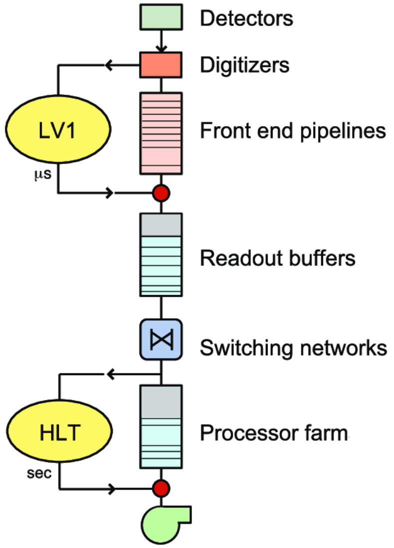


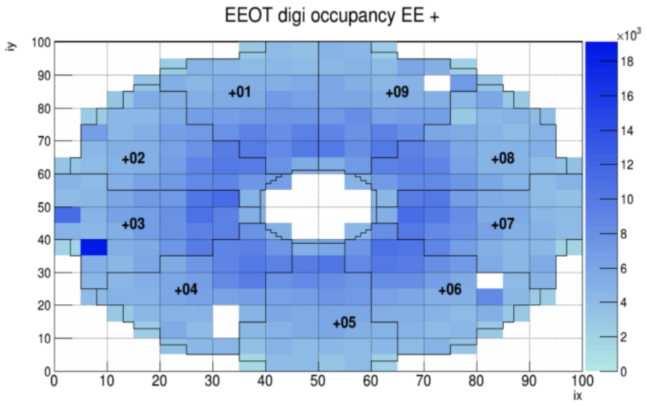
As an example: a noise-dominated channel in the hadronic calorimeter leads to anomalies in the distribution of missing transverse energy (MET) related quantities

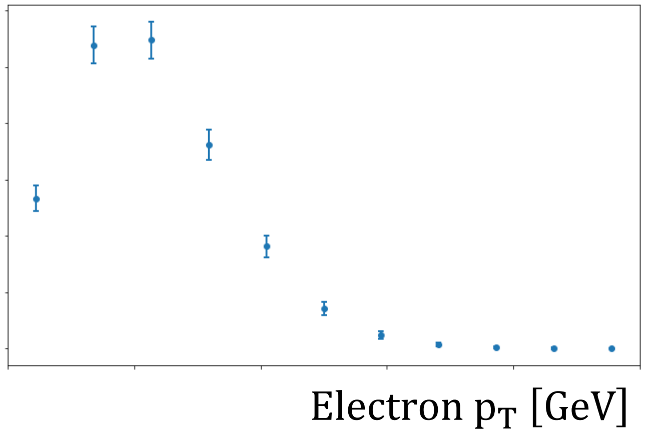
LHC data taking in a nutshell

- The LHC produces collisions with a rate of 40 MHz
- A two-stage trigger filters out uninteresting* events, reducing the rate to ~1 kHz
- Data is monitored "online" in chunks of ~23 s (LSs) and "offline" after reconstruction







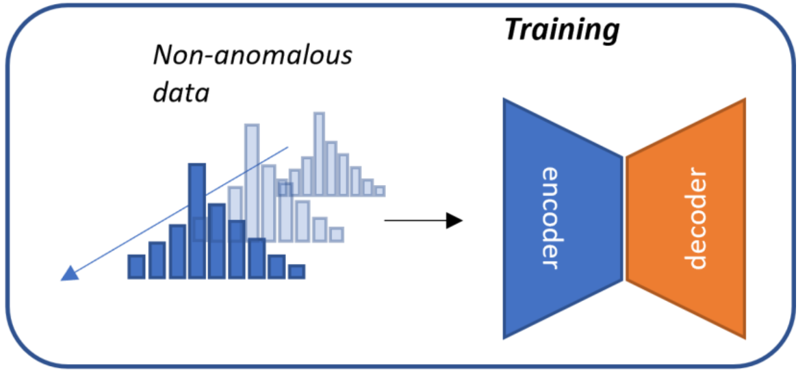


40 MHz collisions → Trigger → Online DQM → Reconstruction and offline DQM

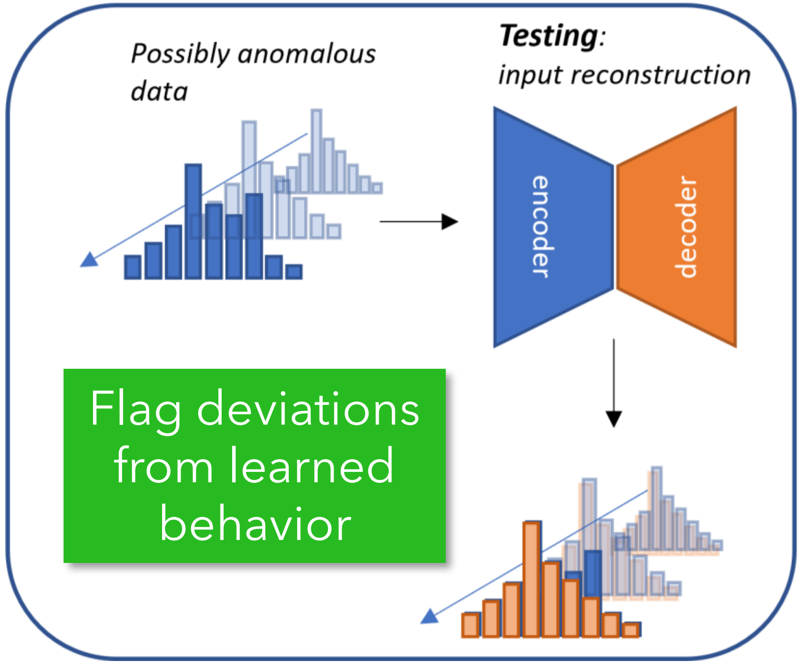
*hopefully, see Thea's talk

The technology

- Anomalies in the operation of the detector are unpredictable by definition
- Unsupervised/semi-supervised methods are the name of the game:
  - Learn “normal” behavior of the detector
  - Flag deviations from known behavior to alert experts



Train on “normal” data

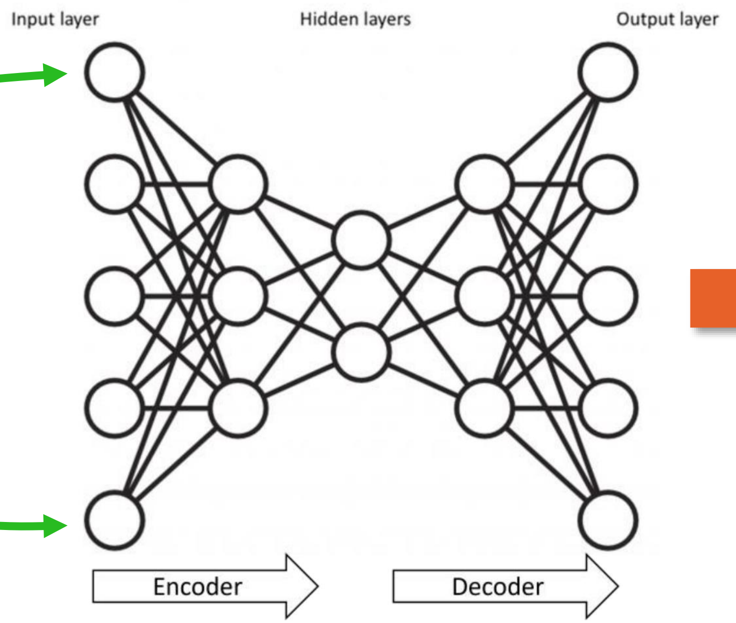


Flag deviations from learned behavior

Aside: how do AEs achieve this?

- The task is very different from classification: **no truth labels during training**
- Simplest (far only) AE architecture: fully connected MLP with bottleneck

$x = (x_0, \ldots, x_n)$



$x' = (x'_0, \ldots, x'_n) \longrightarrow \mathcal{L}(x, x') = \|x - x'\|$

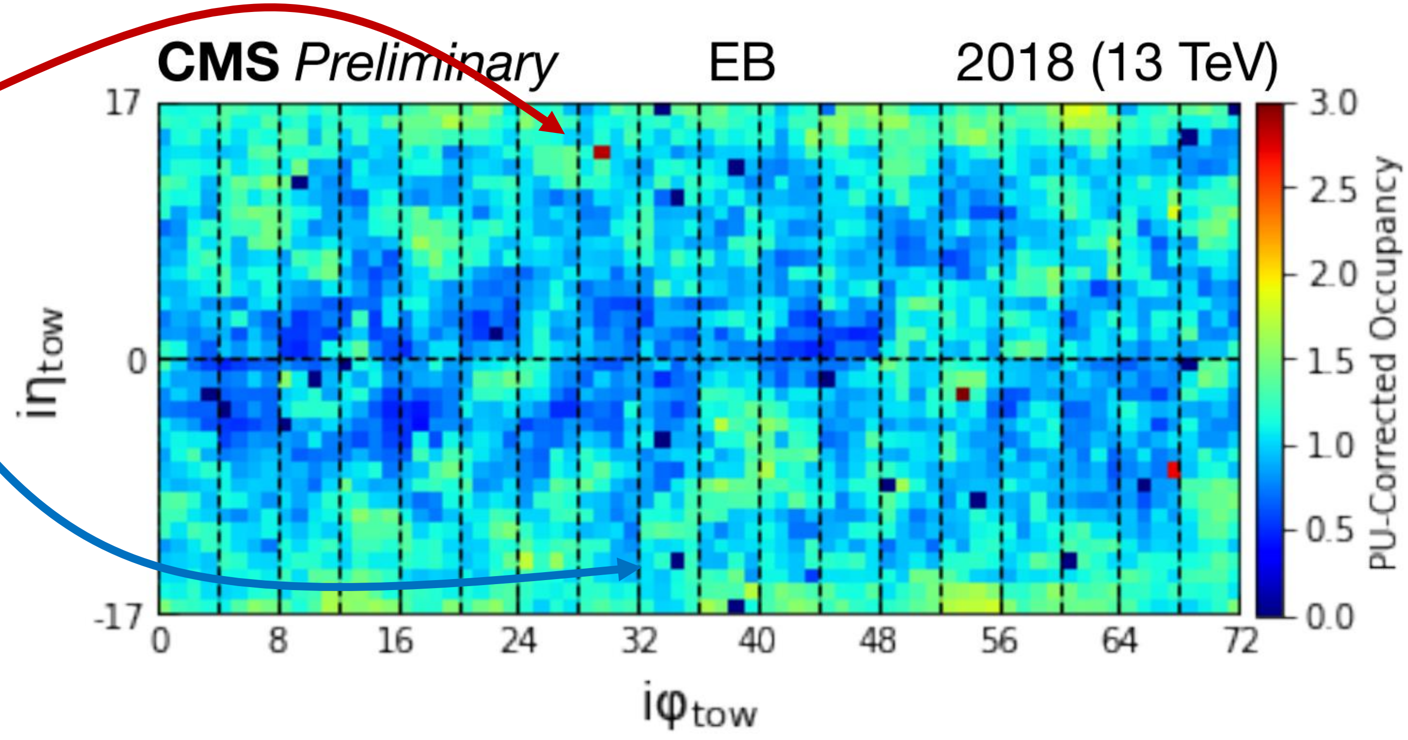
- Train by compressing and reconstructing input; at inference time, monitor the **reconstruction error**

The EM calorimeter

ONLINE DQM

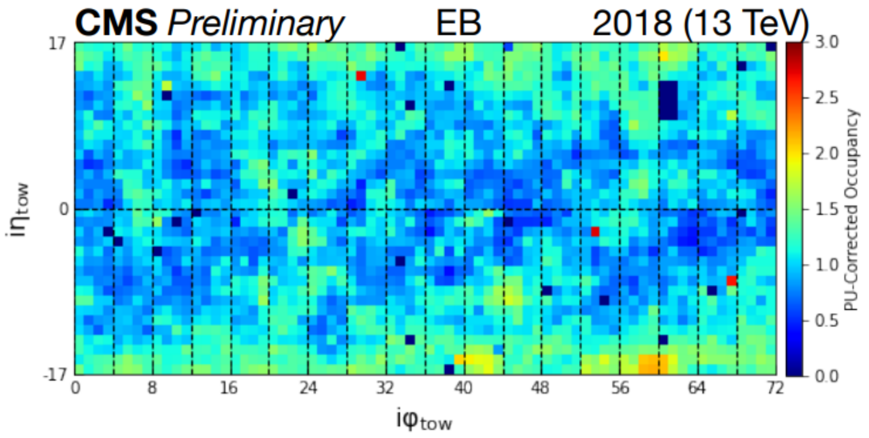
ECAL online DQM

- The aim is to monitor the behavior of the EM calorimeter (ECAL) during operation of CMS, spot issues as promptly as possible to intervene with fixes
- The main quantity being monitored is the occupancy map of calorimeter cells
- Two typical problems:
  - Noisy cells manifest as higher than average occupancy
  - Low-response cells manifest as lower than average (or zero) occupancy
- This data is gathered with LS granularity i.e., every 23 seconds
- **Anomalies can be missed by human operator over hours of shift**

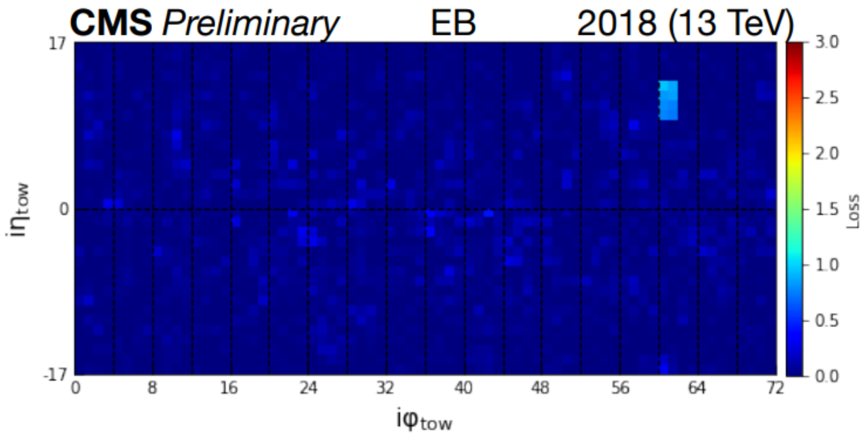


Automatizing ECAL DQM

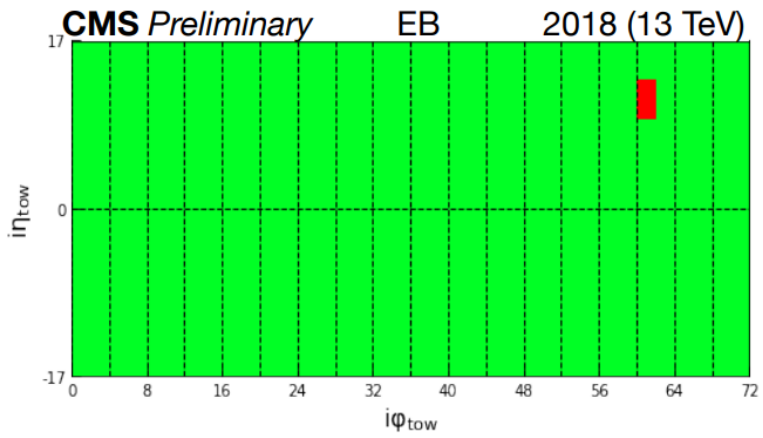
- Train an AE to reconstruct the occupancy map as an image
- ResNET-based architecture with conv layers for both the encoder and decoder
- The pixel-by-pixel reconstruction error (loss map) makes anomalies evident
- Lower false positive rate w.r.t. simple threshold on cell occupancy



Original occupancy map



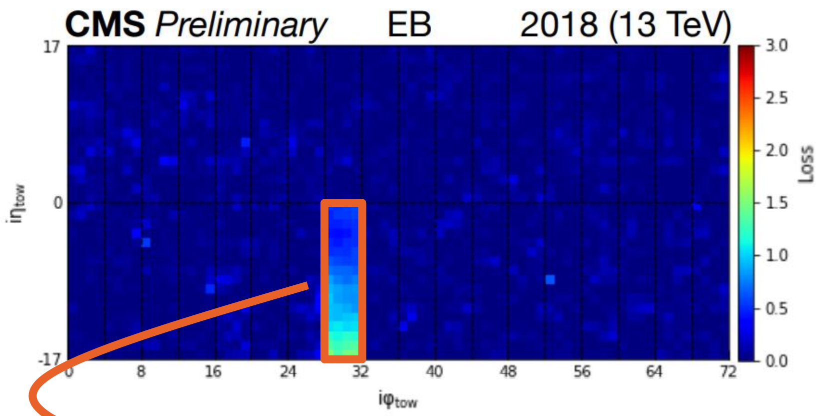
Loss map of the AE



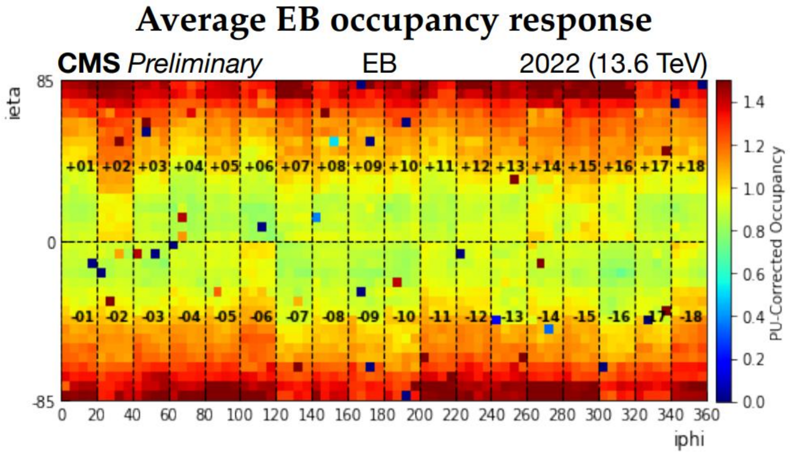
Loss map after threshold

A few more details

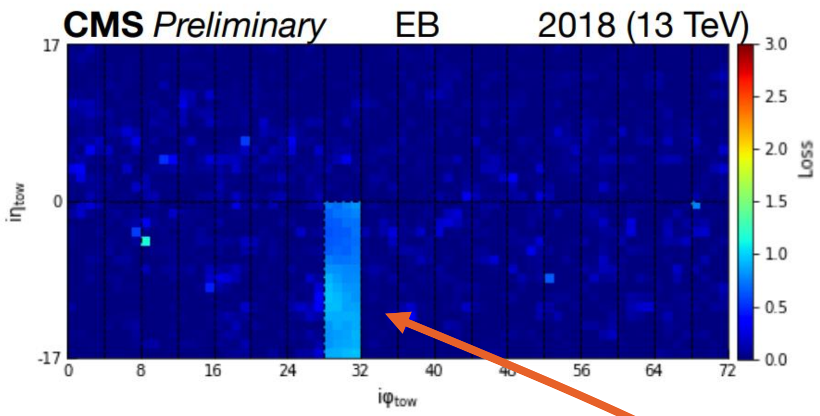
- Correct processing of the data is crucial (*garbage in, garbage out*), for example:
- Because of collider physics, high pseudorapidity (η) regions have higher occupancy
- But we still want the same relative discrepancies to be flagged → normalize by average



- Whole block off, but
- AE learned to expect more hits at high η, so
- Higher loss at higher η



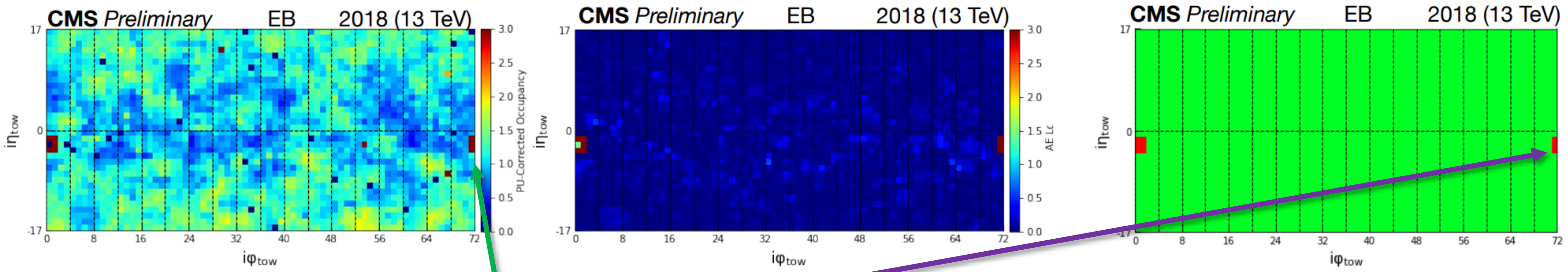
Normalize loss to average occupancy



Loss is flat over η for the same anomaly

Deploying on 2022 data

- This technology was deployed in CMS for 2022 data taking
- Strong performance on new data



Noticing **this** instead of **this** in real time is much, much easier, especially after 8h of shift :)

- Supervised nature means that any* potential anomaly will be flagged as anomalous

*not so simple: see outlier reconstruction problem in Florian's talk

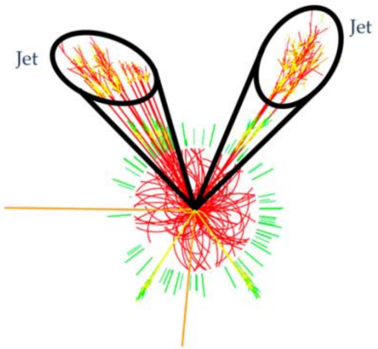
Jets and MET

OFFLINE DQM

JetMET offline DQM

- After data is acquired and reconstructed, a second set of quality checks is performed
- The focus is moved from detector-level quantities, like hits and efficiencies, to physics-level quantities, like muons, electrons, hadronic jets[1], missing transverse energy[2] (MET)
- For the specific case of hadronic jets and MET (JetMET), data is grouped in runs spanning O(hours) → an issue in one run can potentially void hours of data taking
- Per-LS granularity desirable → pinpoint issues to limited set of "bad" data, save the rest
- Every run has O(1000) LSs, prohibitive with "standard" (i.e., human looking at distributions) approach → need an automated effort

[1]: hadronic jets are sprays of particles originating from a high energy quark or gluon

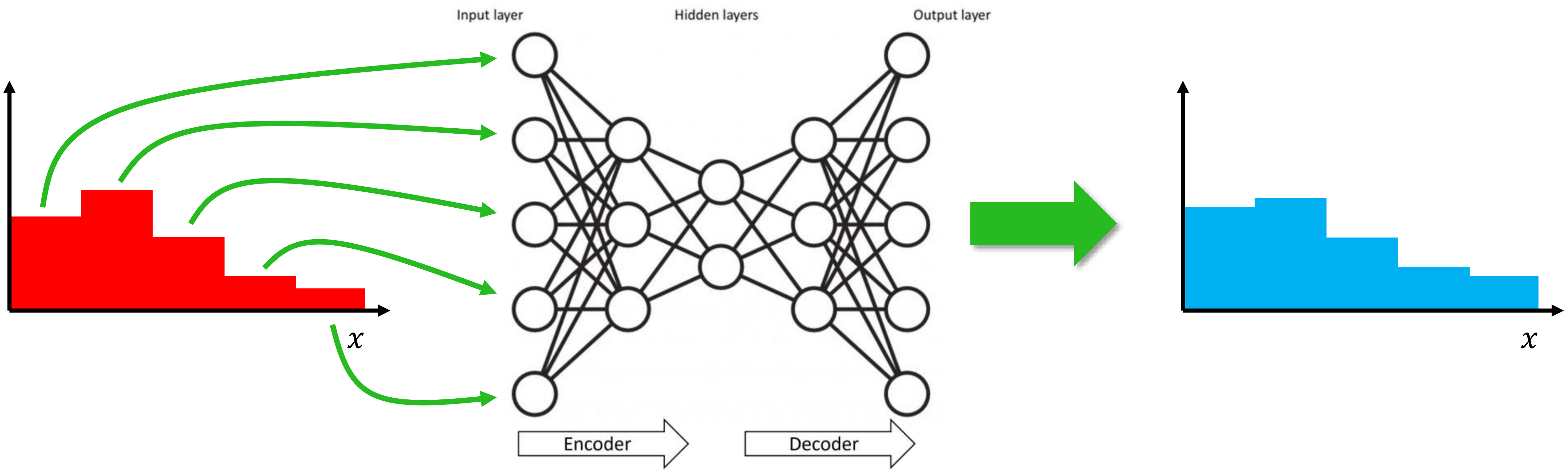


[2]: MET is the momentum imbalance on the plane transverse to the colliding beams

$$MET = \left| \sum_i \vec{p}_{T,i} \right|$$

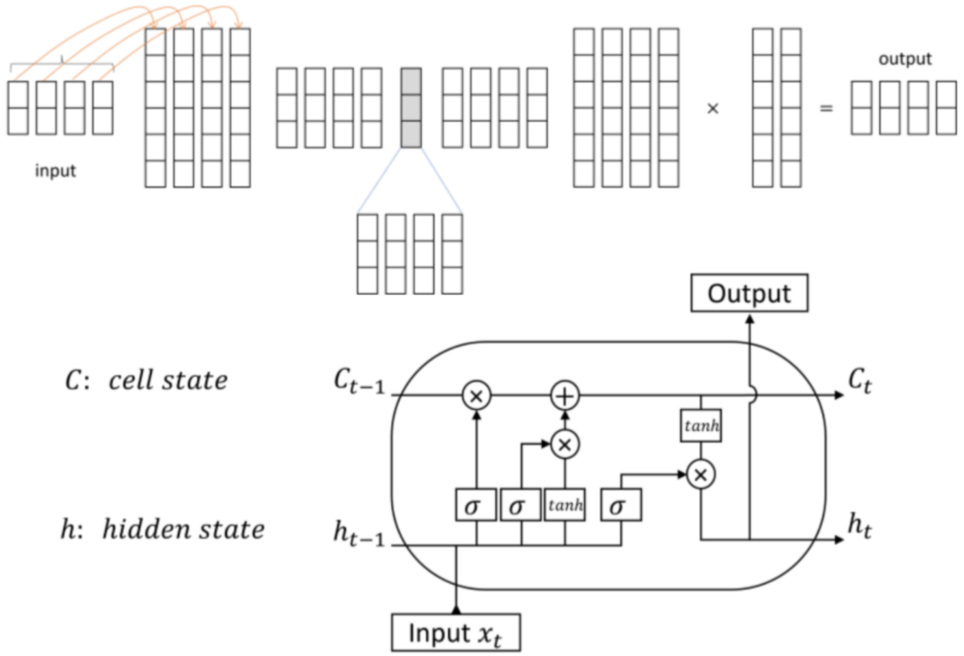
Achieving per-LS granularity

- Slightly different challenge: if a run shows issues, have a tool that goes through each LS and flag the ones that are potentially anomalous
- Expert has to look at O(10) plots instead of O(1000)
- Achieved with simple yet effective AE that encodes/decodes histograms



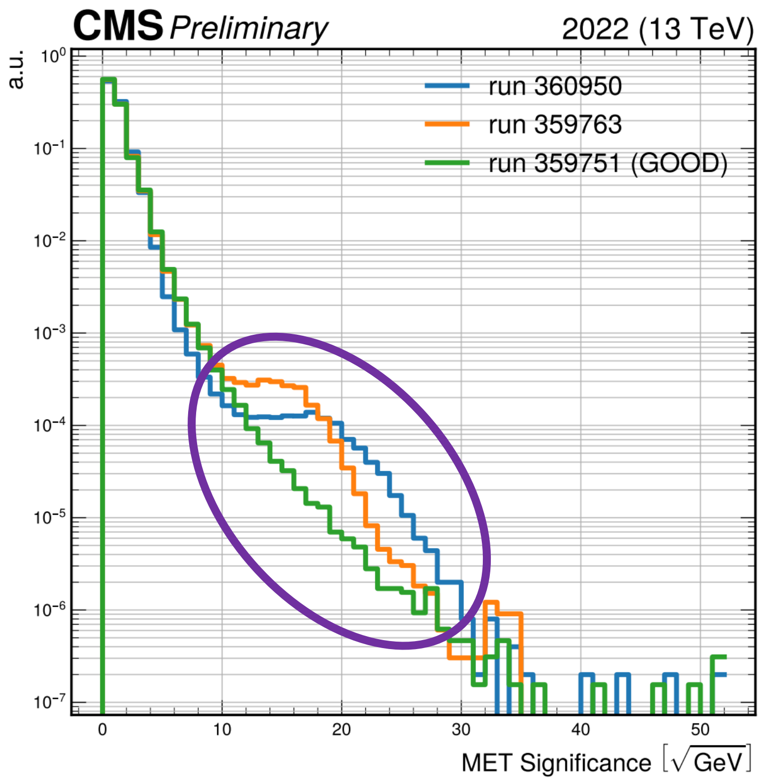
Architecture of the AE

- One consideration is that some distributions are expected to naturally shift during
- E.g., the number of interactions per collision event gradually diminishes during a “fill” because of degradation of the beams in the LHC
- This morphs the distributions of physics observables, like MET, over time; can lead to false positives when looking for anomalies
- Tackled by treating a sequence of histograms from successive LSs as a time series
- Long-Short-Term-Memory (LSTM) nodes read the “sliding window” → account for natural shifts but still sensitive to sudden changes
- Anomalous drifts (on longer time scales) are monitored by integrating over runs



Deployment on 2022 data

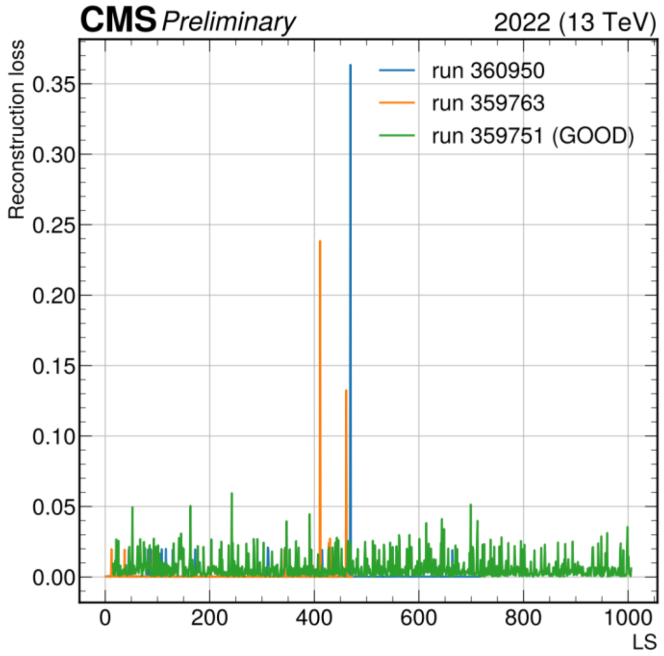
- During 2022 data taking, some runs exhibited clearly anomalous behavior in JetMET related quantities, such as the simplified MET significance[1] (METSig)
- Clearly unphysical bump in the METSig tail
- This is the region of interest for many analyses searching for undetectable particles
- These runs lasted O(1d): removing them from the pool of "good for analysis" runs would have meant a significant loss of data
- Almost $350\ \mathrm{pb}^{-1}$ of data, or 1% of the whole 2016 CMS dataset
- Ideal testbed for deploying the per-LS AE



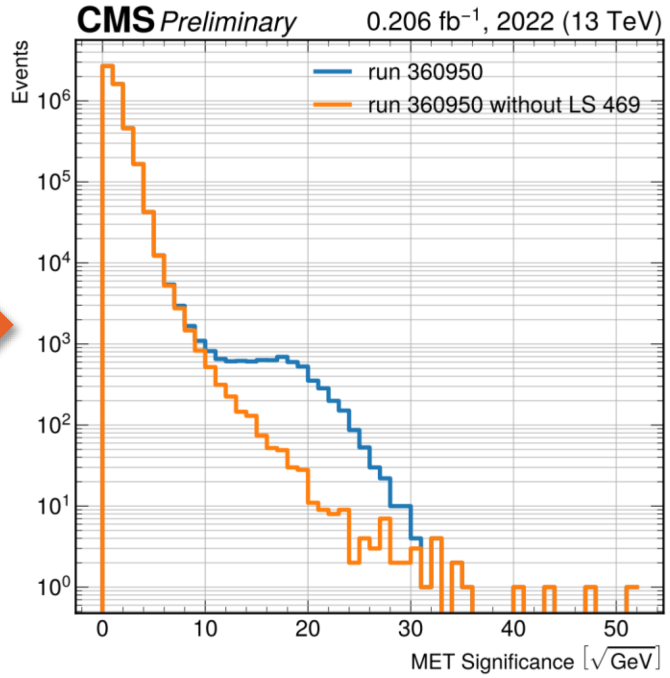
[1]: $METSig \equiv MET/\sqrt{\sum p_T}$ estimates the probability of MET being genuine i.e., not from detector inefficiencies

Results

- After deploying the tool on the two anomalous runs, the issues were restricted to a total of 3 LS (i.e., 3x24 seconds of data taking)



Removing LS 469



- This run alone contained ~1600 LSs → with a simple AE we could pinpoint the issue in a matter of minutes and fully recover data that would otherwise have been lost

In summary

- I hope these two examples from DQM in HEP illustrate a few key points:
- The "ML approach to programming" (see Mauro's introduction talk from yesterday) means that we don't have to invent a new algorithm for every problem, because
- We can abstract the problem from "how do I find LSs in which the METSig shows a bump in any position" to "anomaly detection"
- Use a well-known technology, like AEs, to efficiently deploy reliable and powerful tools
- Exploit the fact that humans have low false positive rates, but can miss many anomalies, while an ML-based anomaly detection tool can be tuned to miss very few of them