An Autoencoder-based Online Data Quality Monitoring for CMS ECAL

Abhirami Harilal, Kyungmin Park,
Michael Andrews, Manfred Paulini

On behalf of the CMS Collaboration

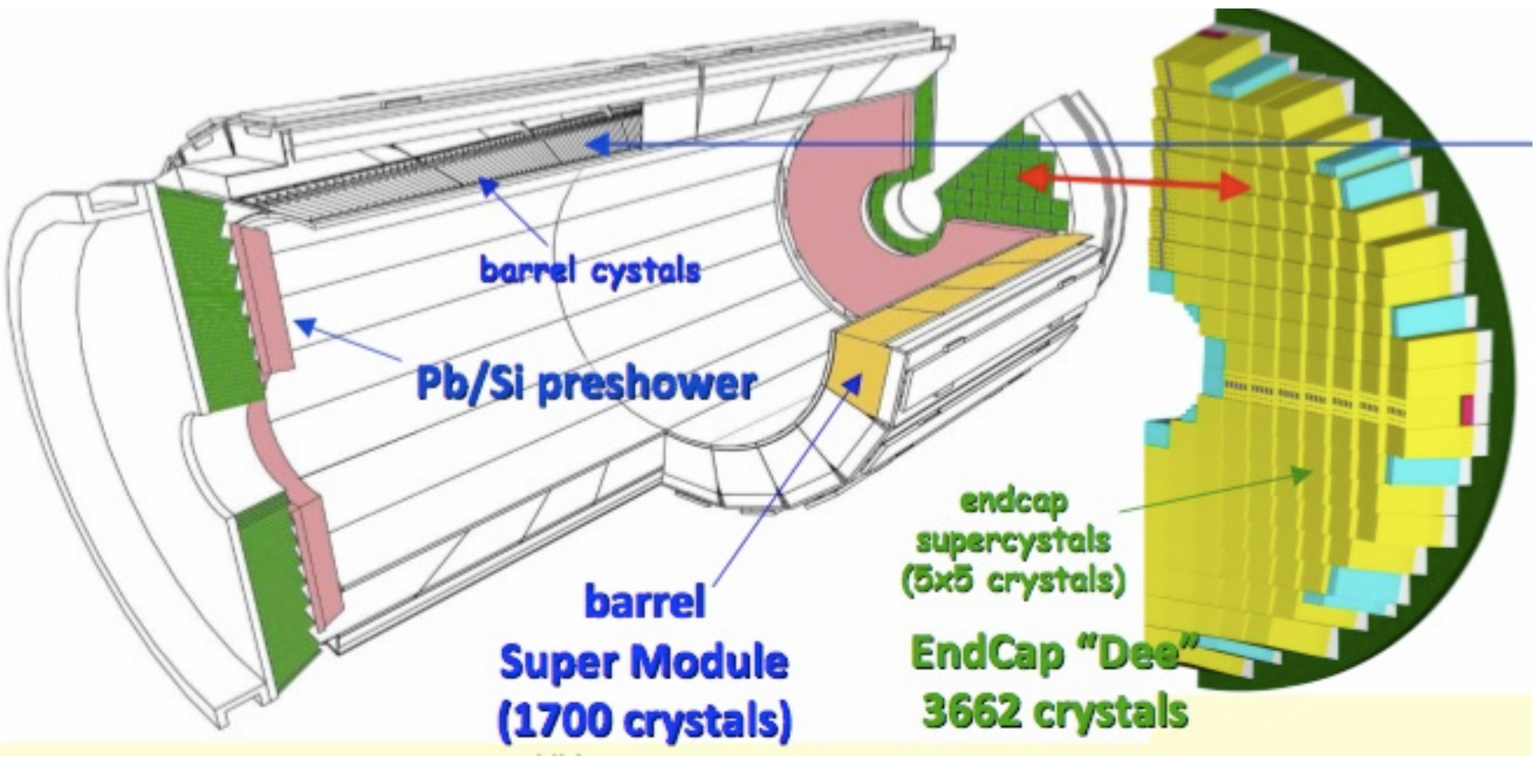
October 27, 2022

@ACAT 2022

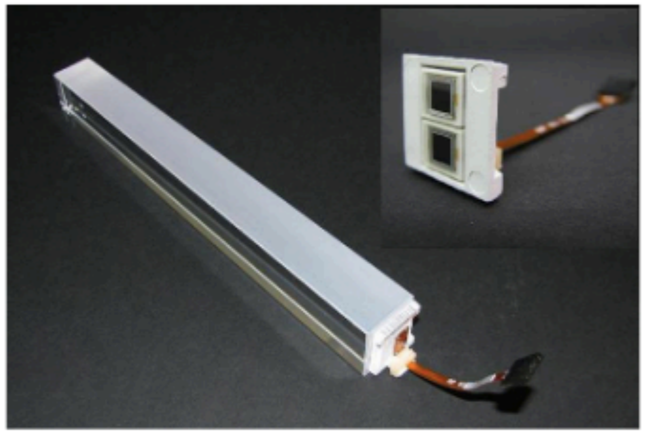
CMS Electromagnetic Calorimeter (ECAL)

- Measures the energy, time and position of electrons and photons as well as EM jet fraction.
  —> crucial to the *discovery and property measurements of the Higgs Boson*

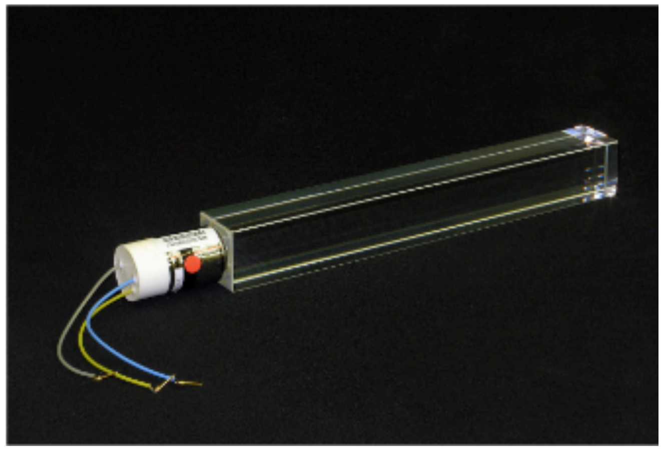
- 75,848 scintillating $PbWO_4$ crystals arranged in a central barrel (EB) section closed by two endcaps (EE+ and EE-)



$PbWO_4$ crystals with photodetectors



Barrel crystal
2.2cm x 2.2cm x 23cm



Endcap crystal
2.47cm x 2.47cm x 22cm

Data Quality Monitoring in ECAL

- Data Quality Monitoring (DQM) is a tool **to identify and localize** detector anomalies that can affect detector performance and quality of data taken.

- Online DQM in CMS ECAL consists of a set of histograms that are populated based on a quick first-pass analysis of a selection of events seen by the detector.

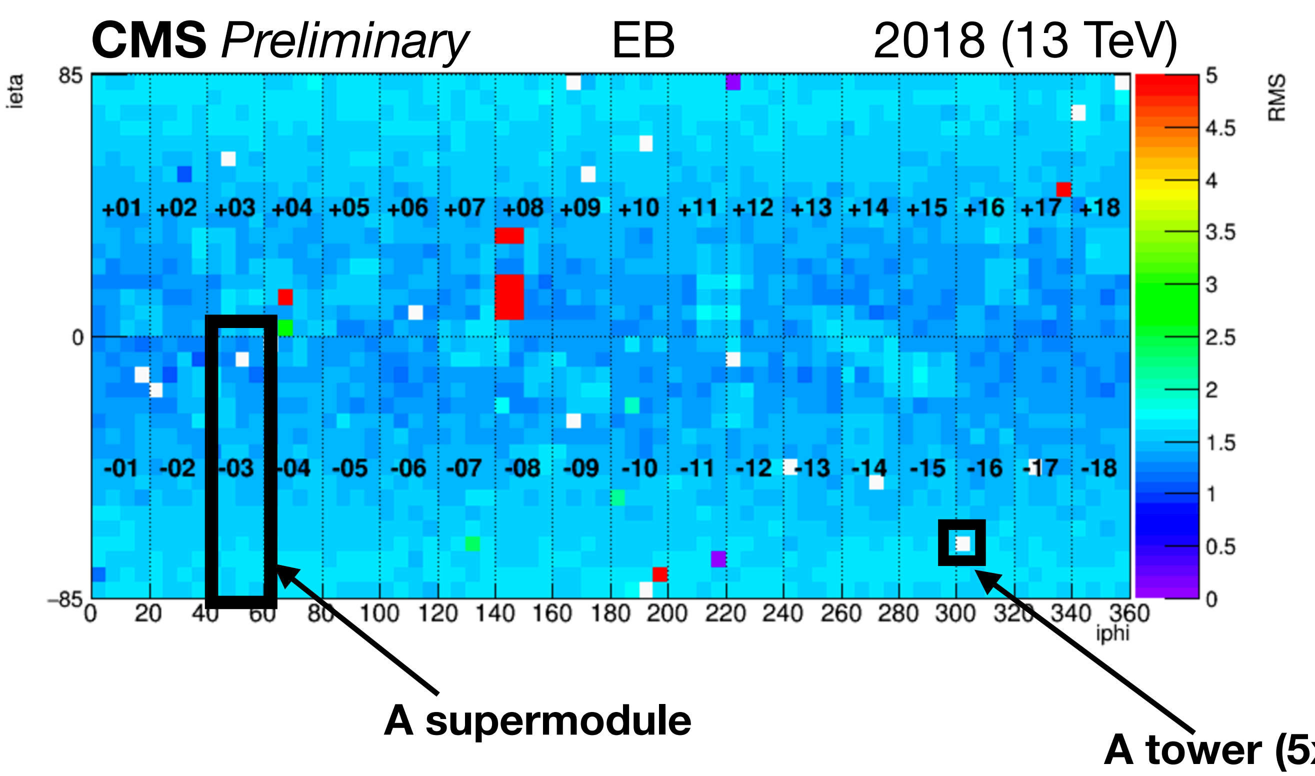
  - **Highly time-sensitive:** must provide feedback to detector experts in real-time to intervene on detector problems ASAP.

  - Time granularity of one luminosity section (LS) ~ 23 seconds.

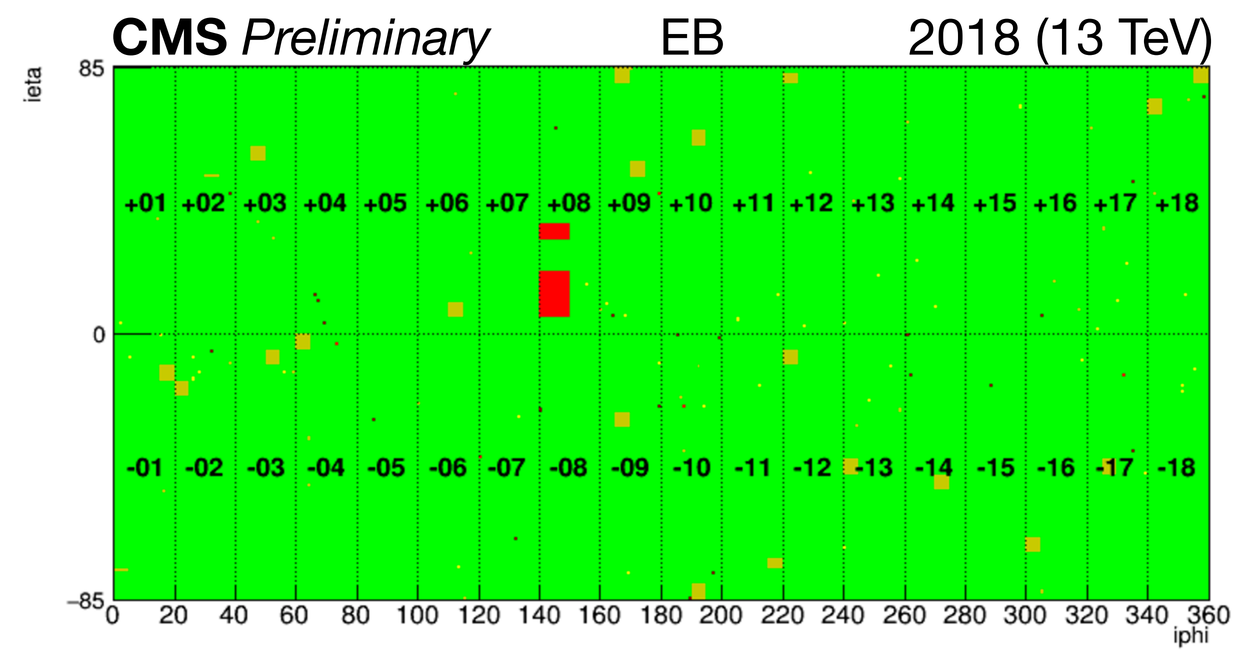
Ecal experts monitoring DQM
at CMS Control Room during Run3 start up

DQM plots in ECAL

Occupancy histogram: collect statistics



Quality histogram: Assess quality



In the quality histogram:

GREEN = good

RED = bad

BROWN = known problem

YELLOW = no data (which may or may not be problematic – depends on context).

The Need for Machine Learning

- **Problem:** anomalies come in all shapes and sizes —> impossible to anticipate all possible failure modes.

- **Semi/Unsupervised ML as potential solution:** robust anomaly detection & localization.

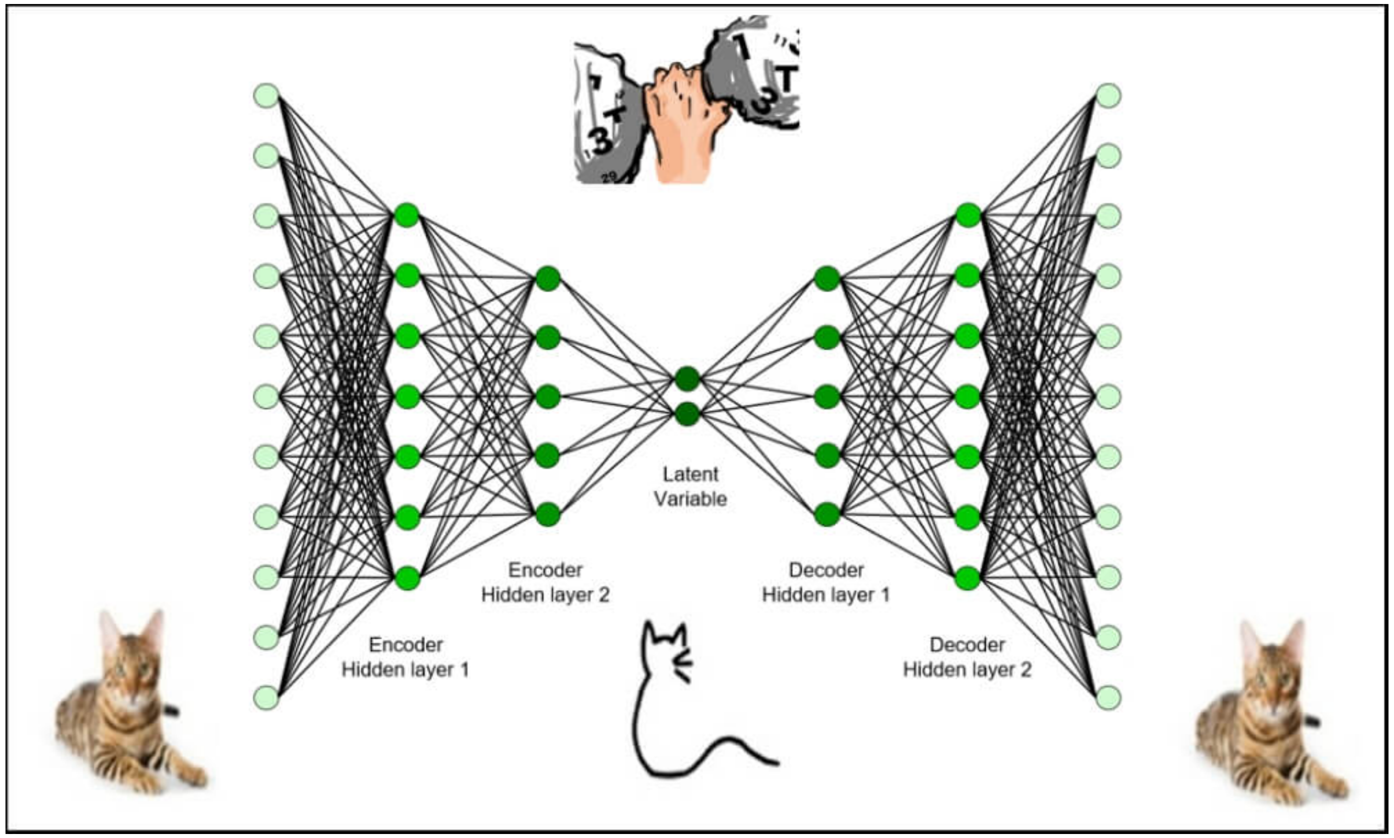
- **Autoencoders**: Learn patterns of good data and make its own quality assessments, eliminating the need for hand coded rules for every possible component geometry in the ECAL.

- More adaptable to changing running conditions and experimental setup.
  —>**This method can be possibly extended to other subsystems and experiments outside of CMS.**

Other efforts in Machine Learning for DQM in CMS

- ML based DQM already in production for CMS DT (arXiv:1808.00911)
- Previous attempt in ECAL using Auto encoder (EPJ Web of Conferences 214, 01007 (2019))
- Various other subsystems in CMS like RPC [link], Tracker [link], HCAL are also exploring ML based anomaly detection.
- The work presented here for ECAL DQM is available in CMS-DPS-2022/043

Auto Encoder (AE) for Anomaly Localization



Input image

Output
AE reconstructed image

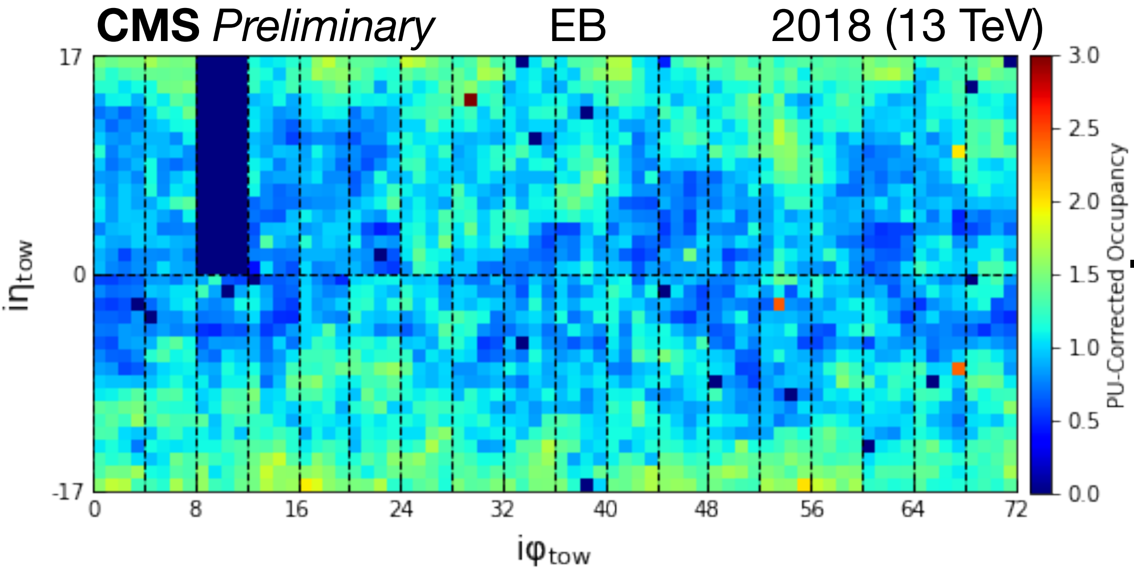
The reconstruction loss (squared error) is computed which measures how well the output matches the input.

Squared Error Reconstruction Loss:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2$$

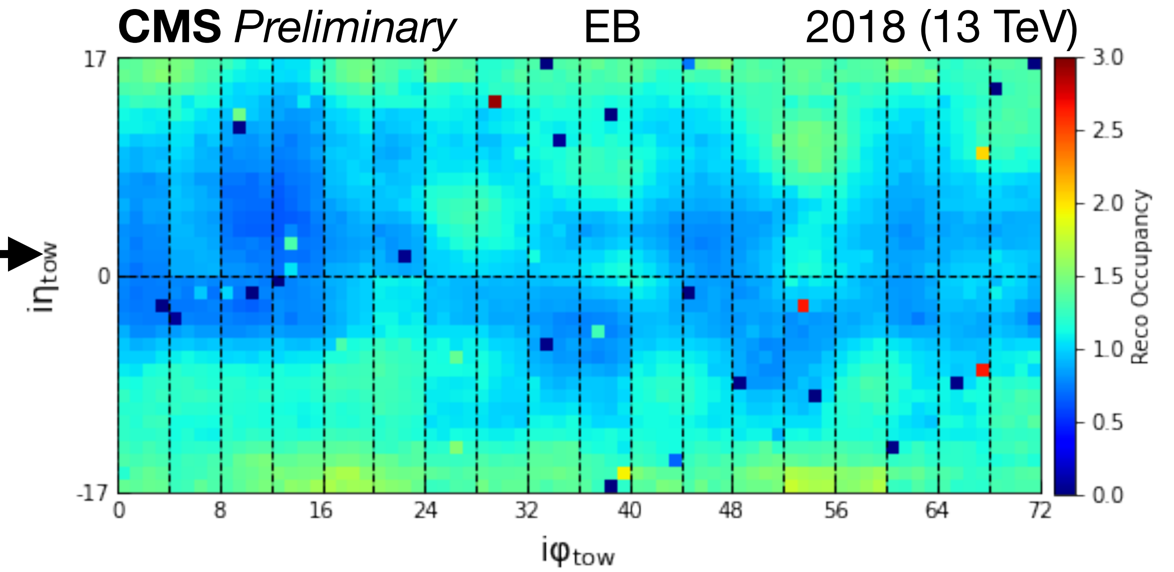
x is input data,
x′ is reconstructed data

Anomaly Localization Strategy

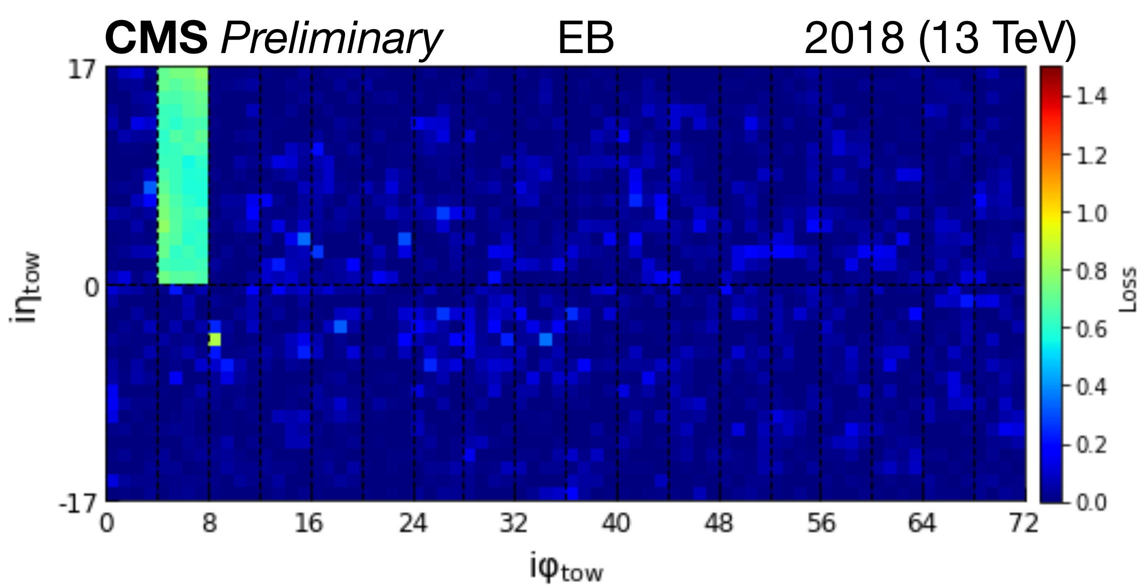


Input occupancy map with anomaly
- Missing supermodule

AE
(Barrel)

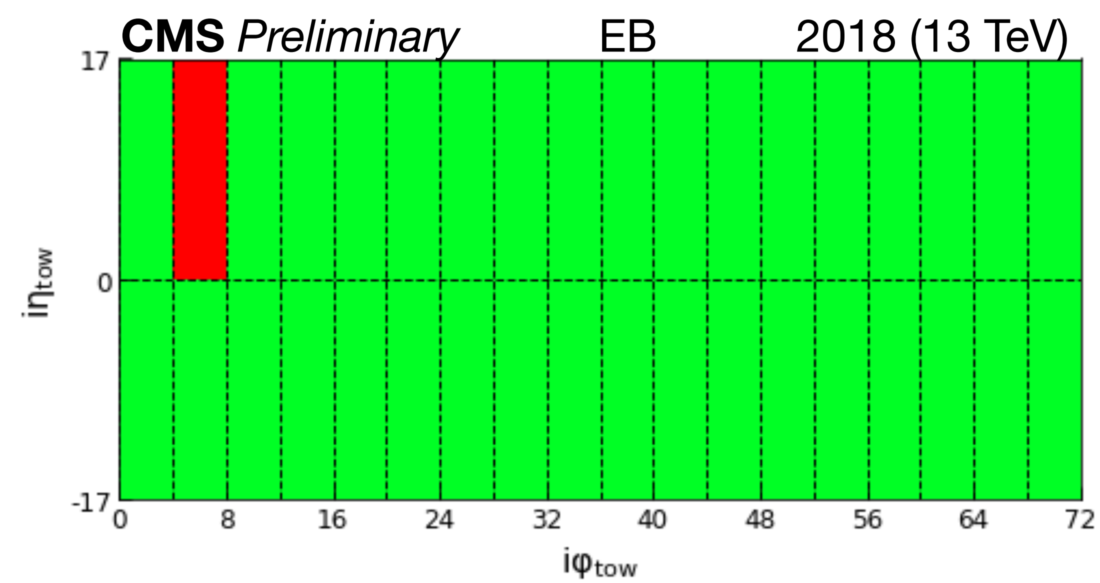


AE reconstructed image
(Does not reconstruct the anomaly)



Loss map: anomalous region shows high loss

Use threshold derived from training and validation for flagging anomaly



Final quality output: bad towers: red, good towers: green

Training Strategy

- Use occupancy maps of ECAL barrel (EB) and endcaps (EE) digitized hits at tower-level (5x5 crystals)
  ➡ 1 image = 1 LS with **constant no.of events**:
  actual events per LS may vary during data taking. This is accounted for during deployment.
  ➡ **Normalize input images w.r.t PileUp** (No.of interactions per bunch crossing)**:**
  consistent quality interpretation vs. LHC luminosity.

- **Dataset:** Certified GOOD runs from 2018.

- **Network: ResNet** for both encoding and decoding.

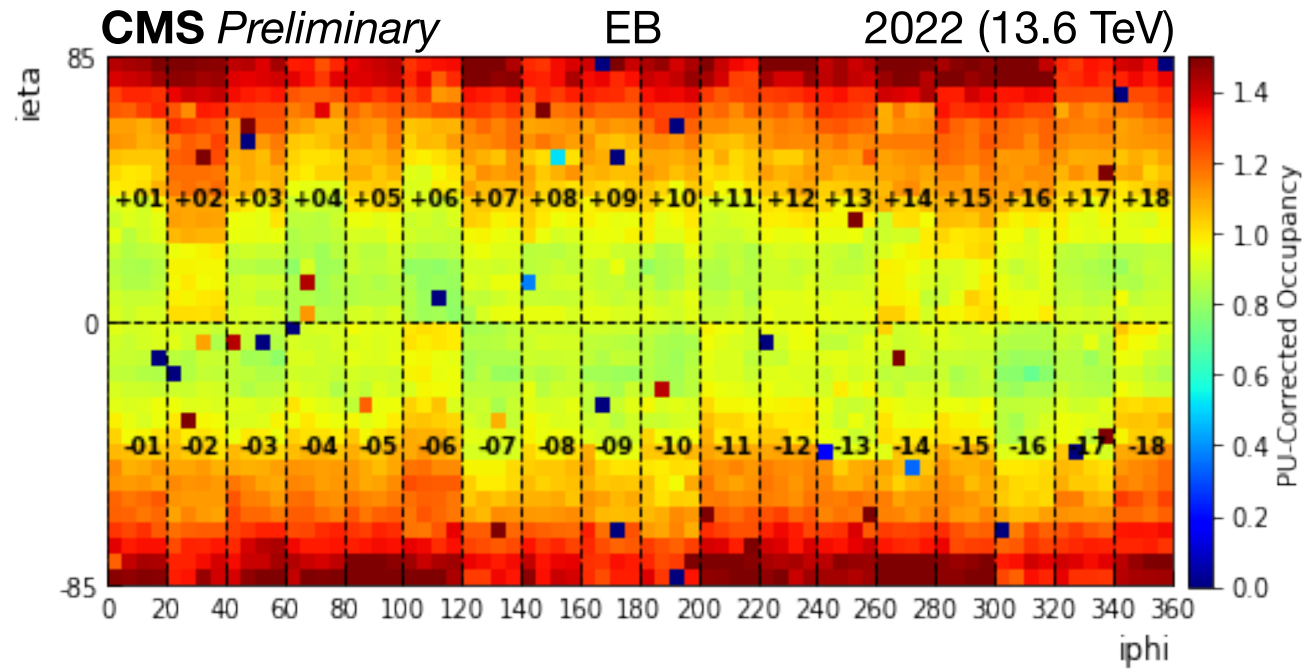
- **Model**: Separate AE models trained for EB, EE+ and EE-.

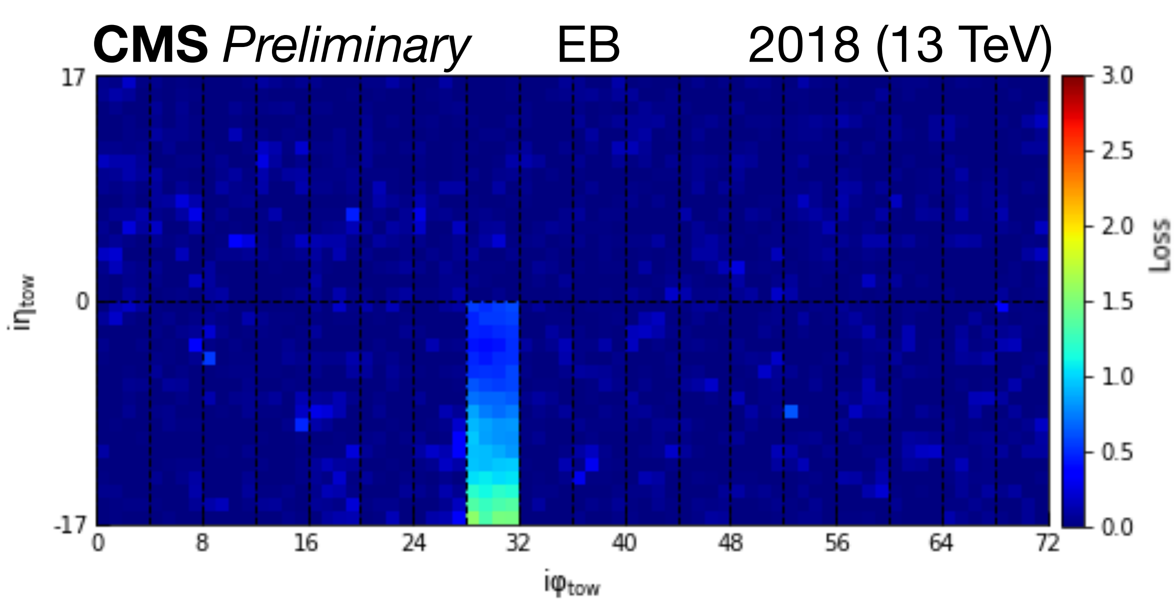
- **Training set:** 90k GOOD images, **Validation set:** 10k GOOD images + same 10k images with "fake" anomalies introduced, **Test set:** real Run2 and Run3 anomalies.

Spatial Response Correction

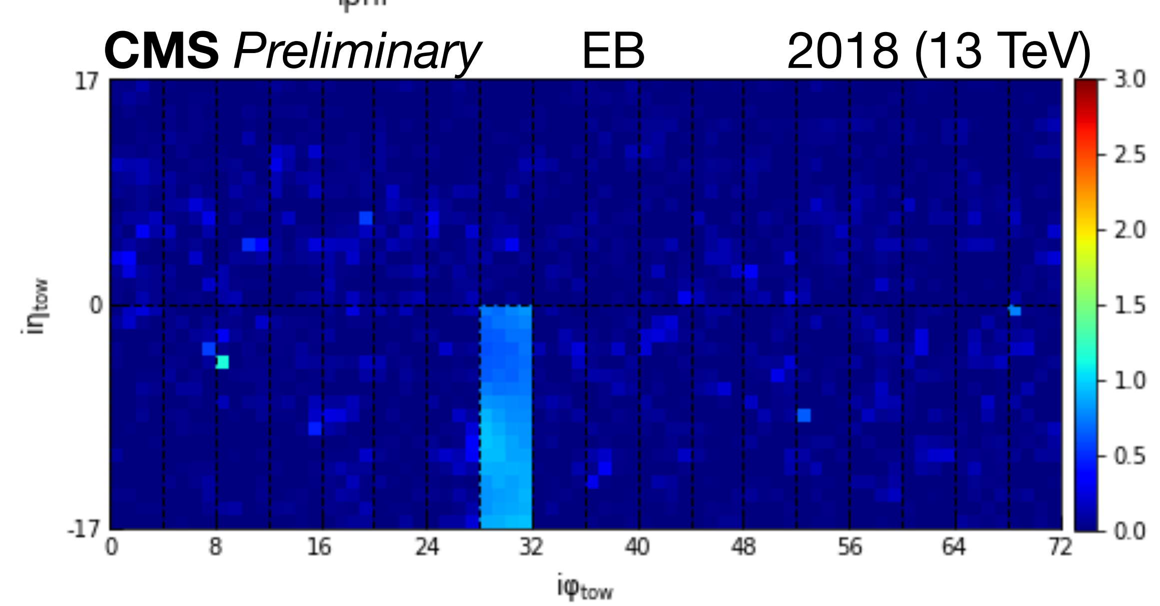
- Normalize **loss map** w.r.t average ECAL occupancy response to "flatten" anomaly response
  - ➡ Dead towers in central η region should be interpreted as equally anomalous as dead tower at high η

Average EB occupancy response





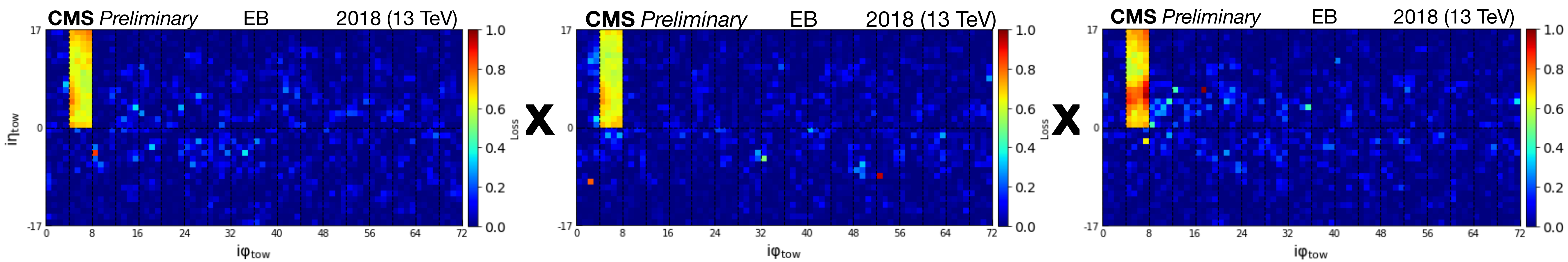
AE Loss map before response correction



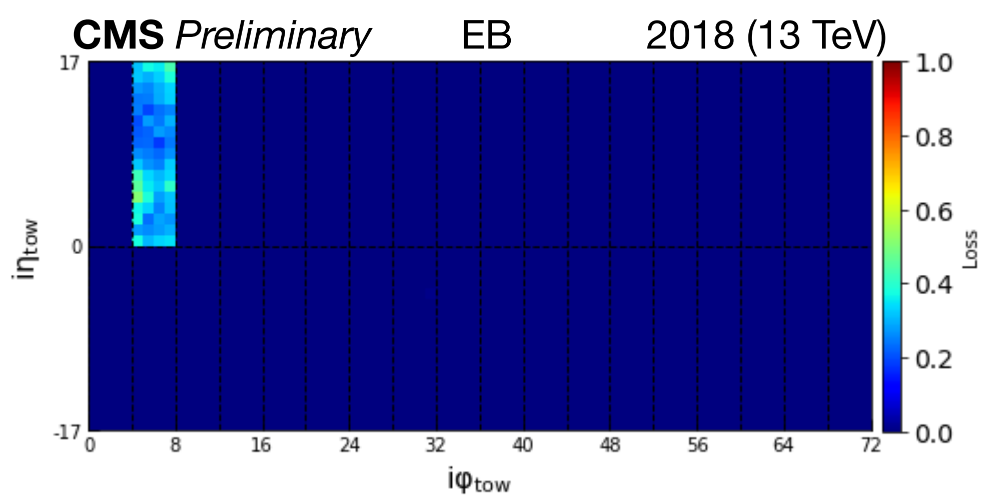
AE Loss map after response correction

Time Correction

- Exploit **time-dependent nature of anomalies**
    - Real anomalies persist with time, fluctuations are transient.
    - Multiply loss map of the last two LS with current one
      —> Lag of about ~1min: very reasonable trade-off, can be tuned.



Time multiplied loss map



Significant reduction in false positives!

Results, Part I

Testing on Fake Anomaly Scenarios

Results on Fake Anomalies

Performance Metric: False Discovery Rate (FDR) at 99% anomaly detection

If we choose a threshold to catch 99% of the anomalies, what fraction of towers labelled BAD will be due to GOOD towers

- What fraction of shifter calls will be **false alarms** ?

FDR at 99% anomaly detection							
Scenario	**Missing Supermodule**	**Zero Occupancy Tower**			**Hot Tower** (10% hot for EB, 20% hot for EE)		
	Barrel	**Barrel**	**EE+**	**EE-**	**Barrel**	**EE+**	**EE-**
No correction	3.6 %	51 %	86 %	87 %	2.8 %	0.01 %	< 1/30k
After *spatial* correction	3.1 %	49 %	13 %	14 %	2.9 %	0.06 %	0.05 %
After *spatial and time* correction	0.13 %	4.1 %	5.6 %	6.3 %	< 1/10k	< 1/30k	< 1/30k

The lower the FDR the better

Mostly due to **actual anomalies** contaminating certified GOOD data. (See Backup).

~10^4 is the size of the validation set

- **In real anomalies, we demonstrate that the AE catches towers with non zero, low occupancy. (See p18)**

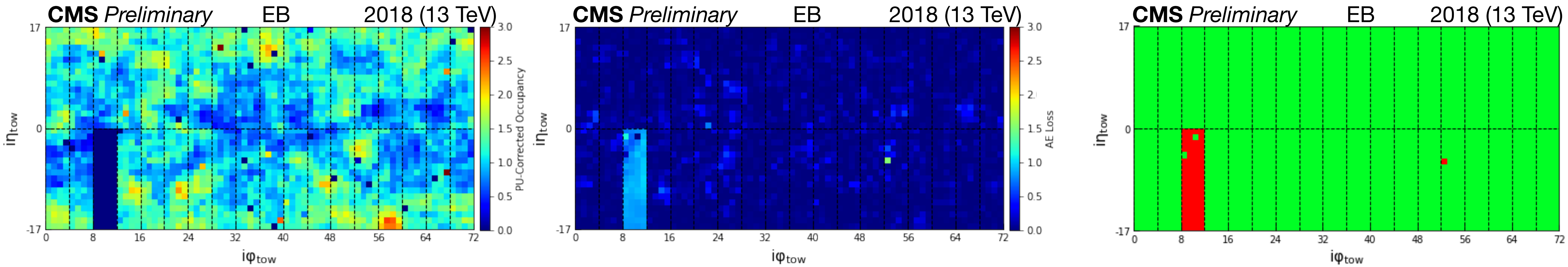
Results, Part II

Testing on Real Anomalies

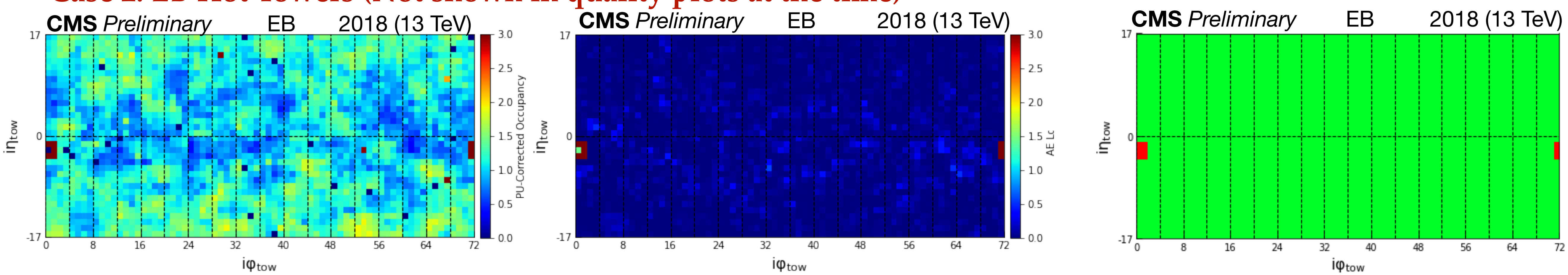
Results on Real Anomalies from Run2 and Run3

Input image | **AE loss map** | **Final Quality plot**

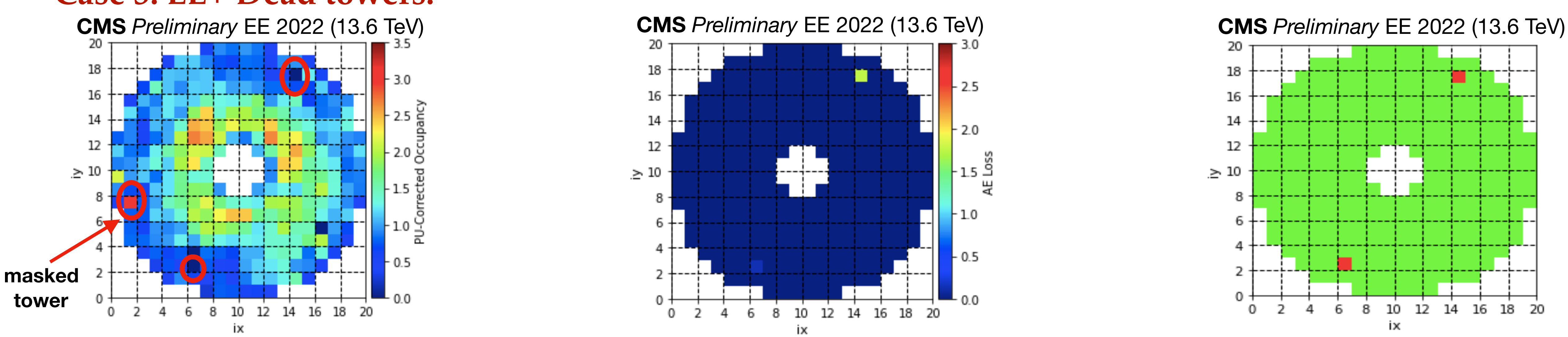
- **Case 1: EB Missing Supermodule**



- **Case 2: EB Hot Towers (Not shown in quality plots at the time)**



- **Case 3: EE+ Dead towers.**



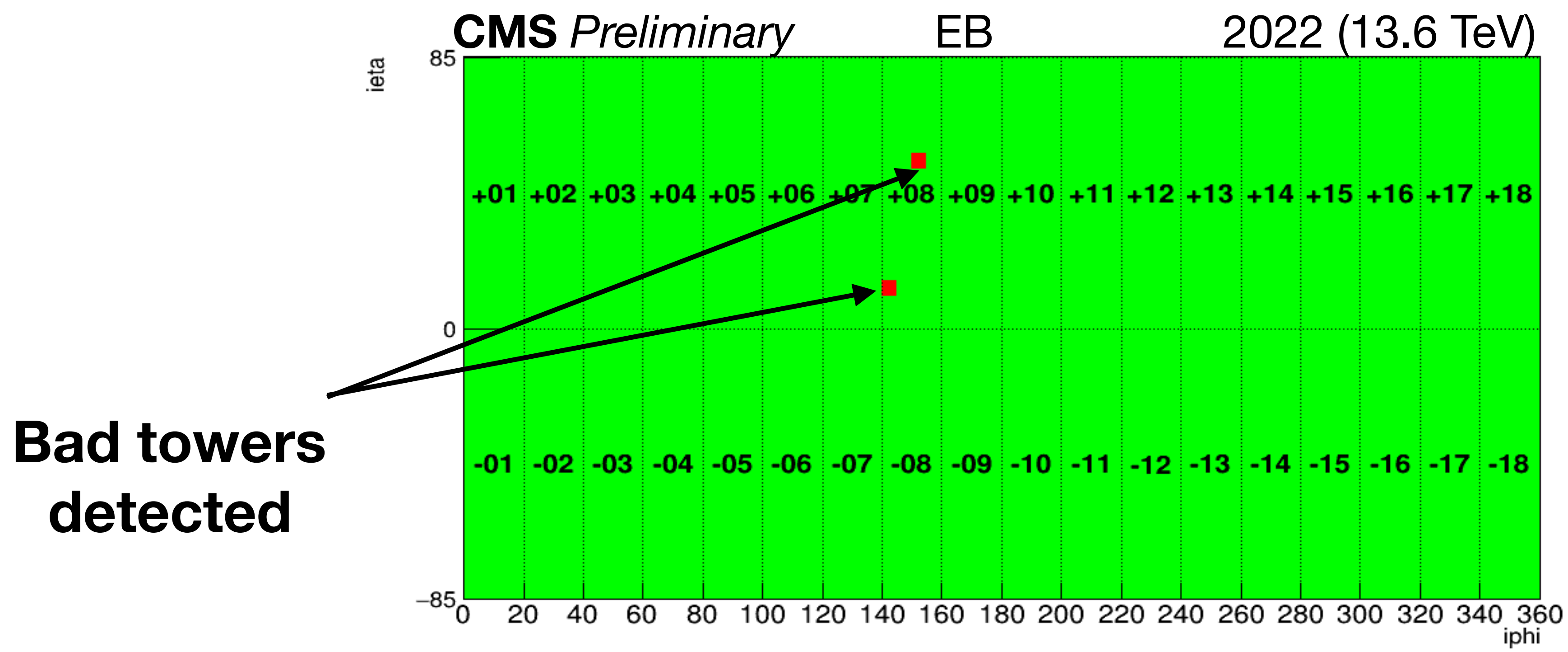
Results, Part III

Run 3 commissioning for Barrel

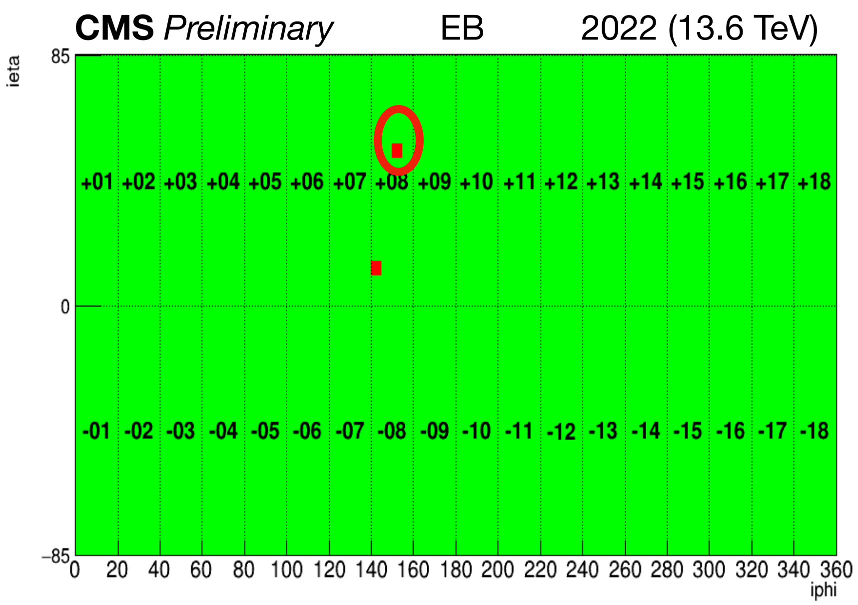
Run3 Commissioning in ECAL Online DQM

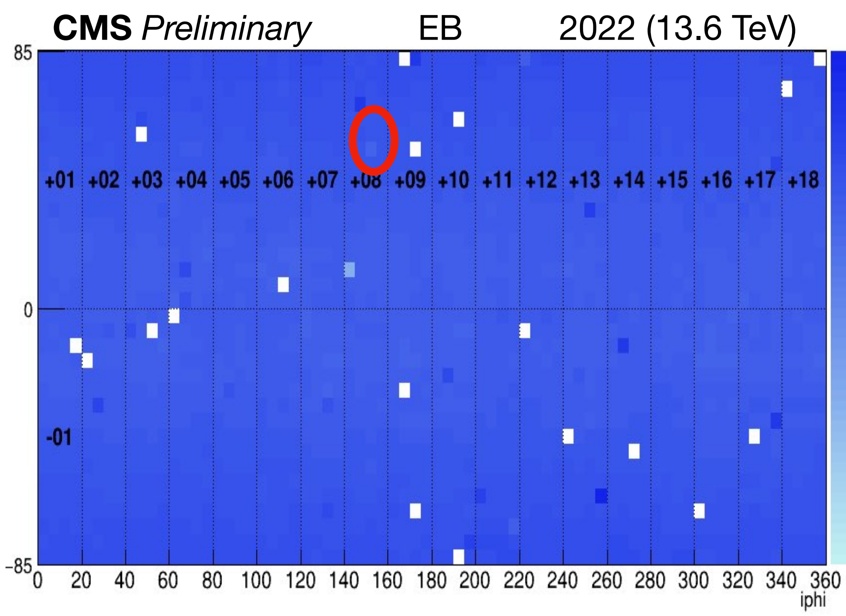
- The MLDQM is now deployed in the ECAL Online DQM CMSSW workflow for **ECAL Barrel**, as a new ML quality plot from the Autoencoder.
    - **Model Inference**: Trained Pytorch model exported to ONNX and run in production using ONNX Runtime.
    - The Endcaps implementation is undergoing further tests and fine tuning before deployment.
- **Doing really well on live data from the detector**.

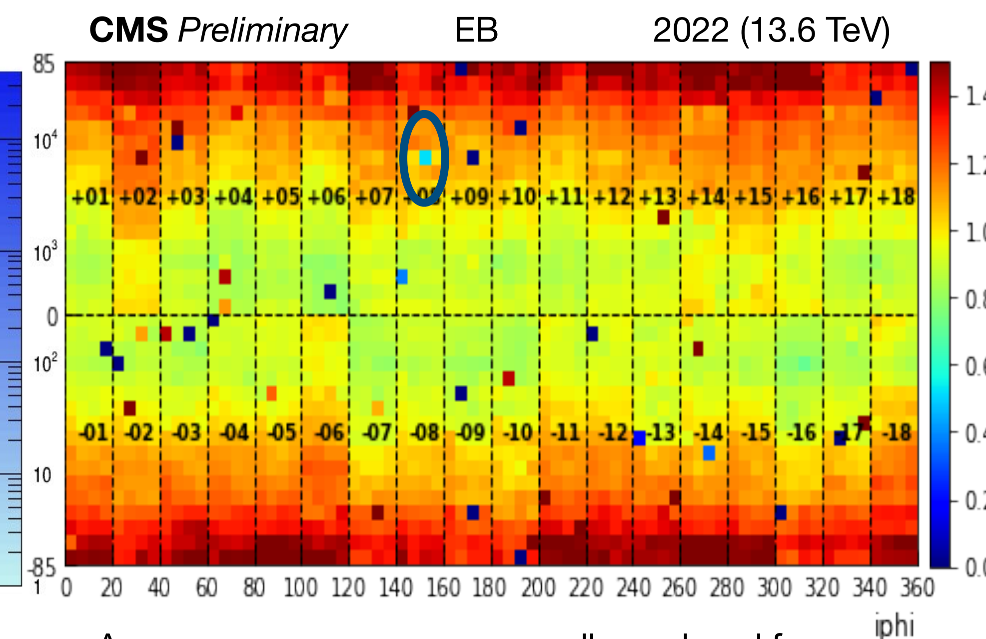
ML Quality plot from ECAL Online DQM during a Run3 run



Detects potential new bad towers







Average occupancy map manually produced from several runs from Run3.

- A tower that had **low occupancy** in several LSs, but not in all – semi-transient anomaly, but still shows up with low occupancy in the average occupancy in Run3.
  - Could be a **degrading channel**.

- Introduce a new plot: occupancy map of bad towers accumulating over a run, to see how frequently they are flagged by the AE.

  —> help experts keep an eye on them.

Summary and Plans

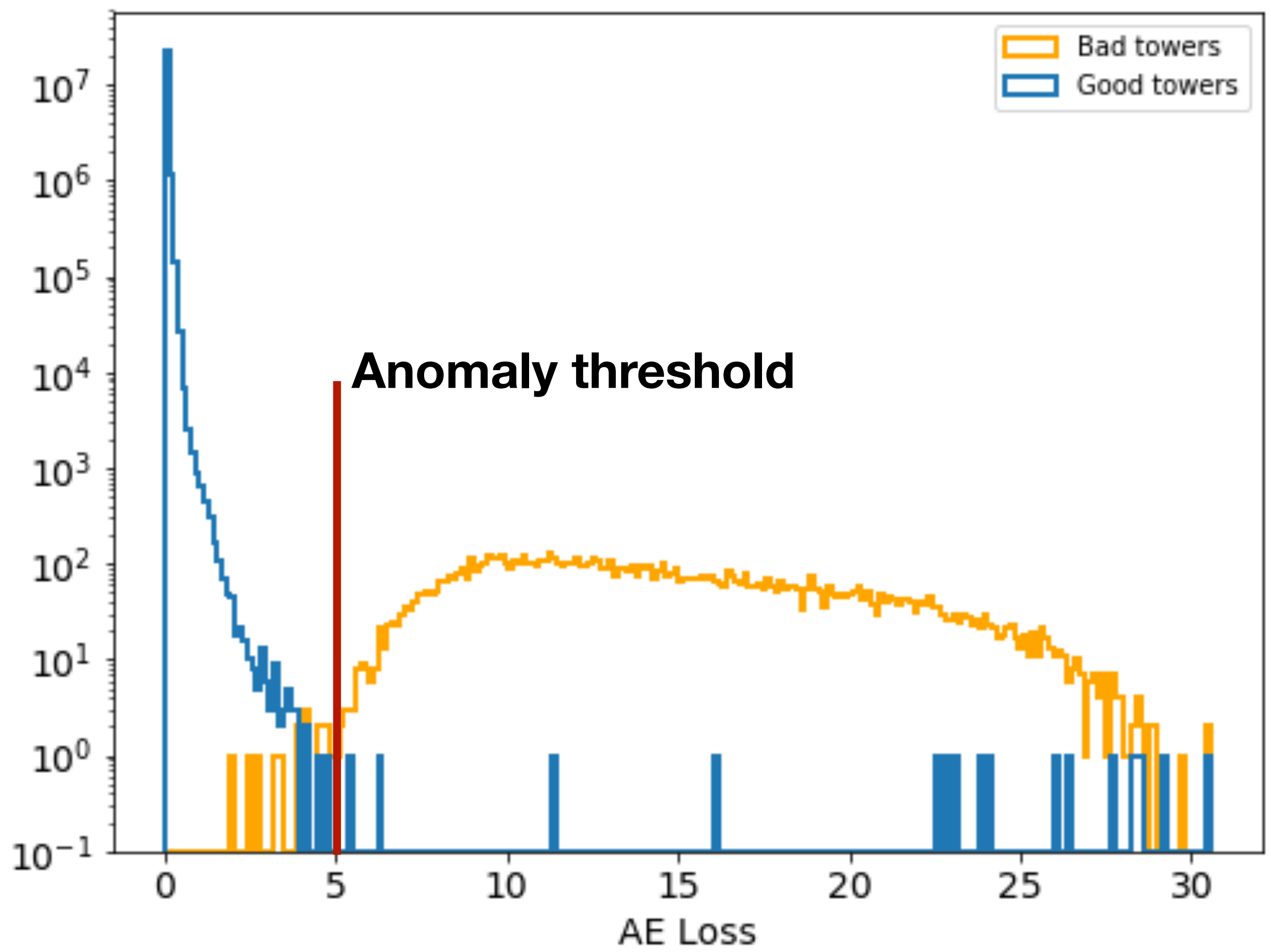
- We have developed a **robust ML based anomaly detection & localization system** for ECAL Barrel and Endcaps.
  - First application to exploit **time-dependent nature of anomalies** for an order of magnitude improvement in performance.
  - Detects anomalies of **varying degrees, shapes and sizes.**
  - Identifies **potential bad towers and degrading channels**
    —> helps to **monitor detector health.**
  - Does not replace existing DQM, but serves as an additional check for improved detection and reducing false alarms.
- **MLDQM for ECAL Barrel** is now **deployed in the ECAL Online DQM CMSSW** workflow, and performing well on live data from Run3.
  - For the Endcaps, further tests are on-going before deployment in DQM CMSSW workflow.
- This AE based method can be generalized and extended to other experiments.

Thank You!

BACKUP

What contributes to the remaining FDR?

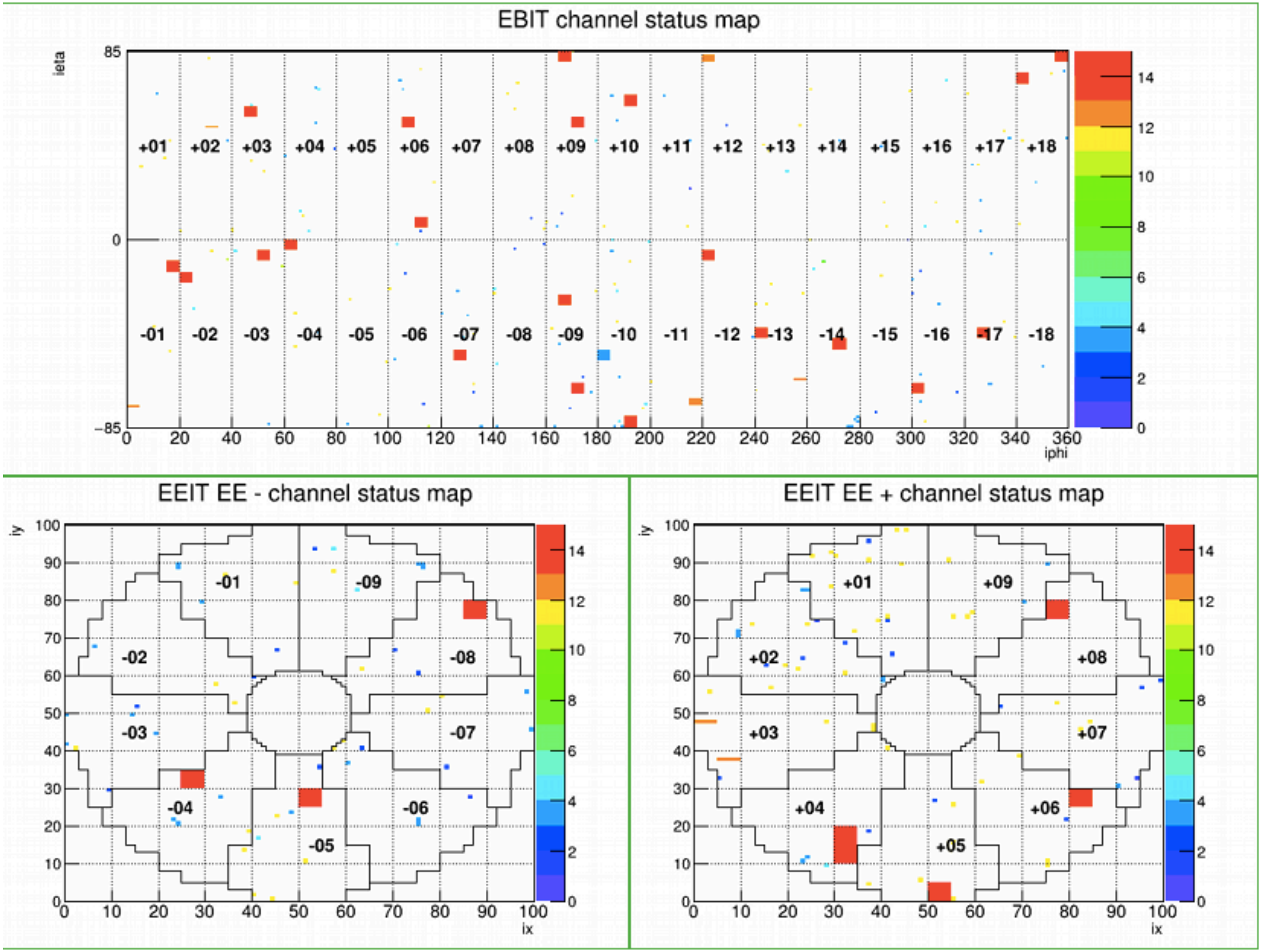
Typical tower-level loss distribution over several LSs:



$$\text{FDR} = \frac{\text{No.of good towers above anomaly threshold}}{\text{No.of good and bad tower above anomaly threshold}}$$

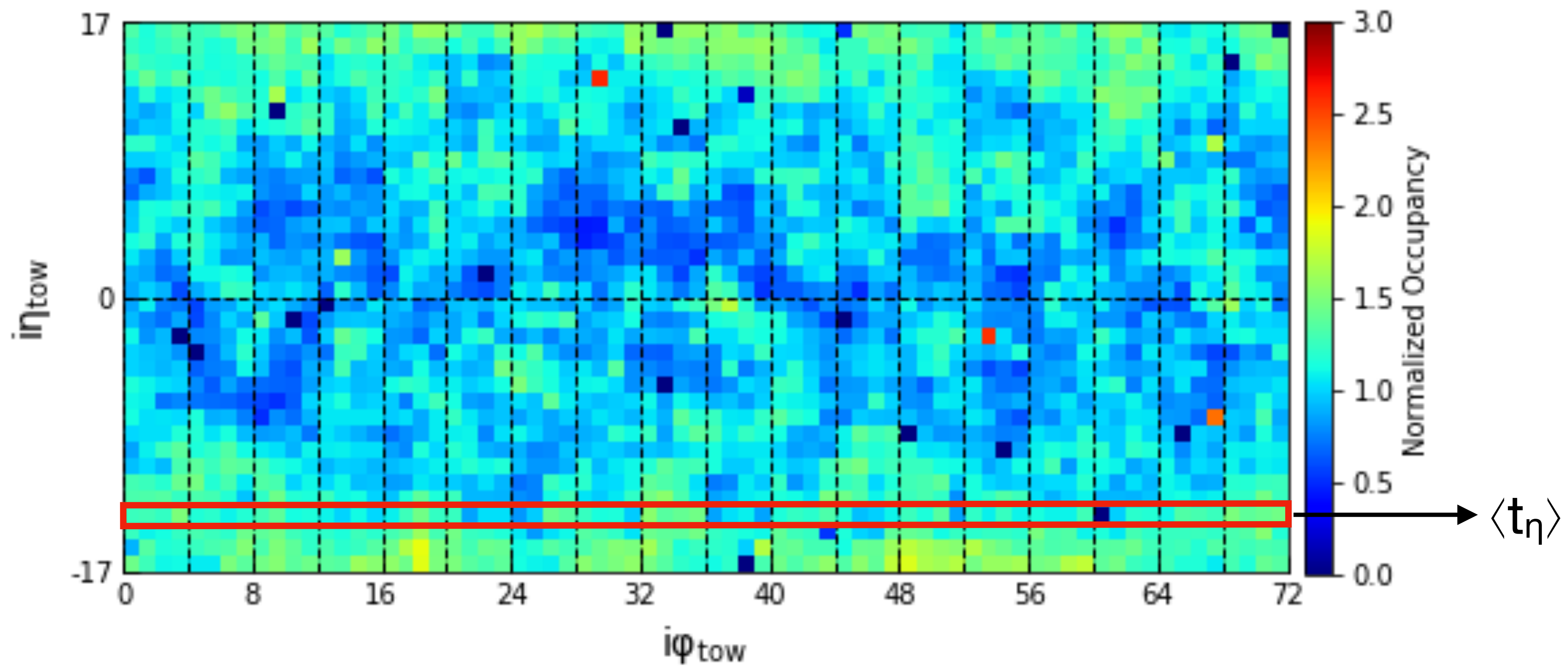
- The "false positives" in the tail of the good towers, with higher loss are actual anomalies in the detector.
- These correspond to bad towers which were not masked in DQM, because it didn't happen often enough —> not considered fatal.

Masked known bad channels



Baseline for comparison

- **Baseline Loss per tower**:
  Compare each tower occupancy $t_{\phi,\eta}$ to η-ring average occupancy $\langle t_\eta \rangle$.
  Define baseline tower $\text{loss}_{\phi,\eta} = | \; t_{\phi,\eta} - \langle t_\eta \rangle \; |$



EB Results with baseline study

Performance Metric: False Discovery Rate (FDR) at 99% anomaly detection

If we choose a threshold to catch 99% of the anomalies, what fraction of towers labelled BAD will be due to GOOD towers (i.e. what fraction of shifter calls will be false alarms)?

Scenario 1: Missing Super modules

	FDR
Baseline	14%
Baseline after time corr	5.9 %
AE before time correction	3.1%
AE after time correction	0.13 %

Scenario 2: Dead tower

	FDR
Baseline	90%
Baseline after time corr	80 %
AE before time	49%
AE after time correction	4.1 %

Mostly due to **actual anomalies** contaminating certified GOOD data.

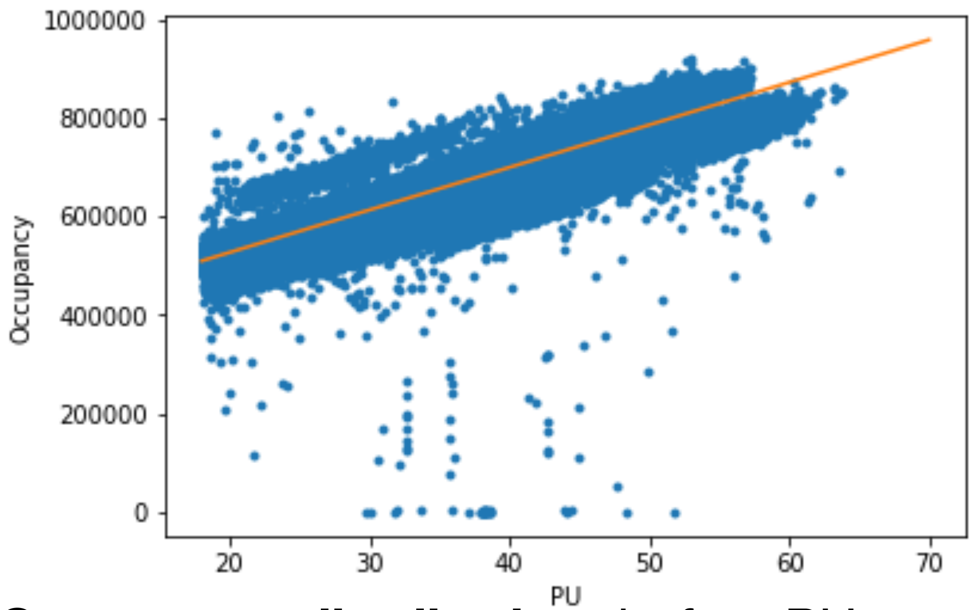
Scenario 3: Hot tower 10% hot

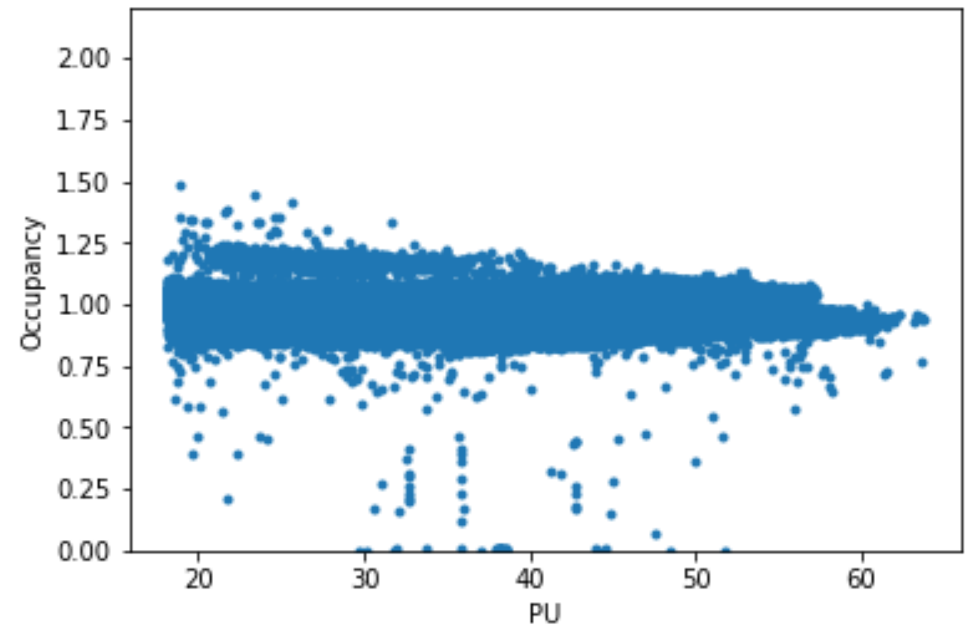
	FDR
Baseline	5.2%
Baseline after time corr	$< 1/10^4$
AE before time correction	2.9%
AE after time correction	$< 1/10^4$

~10^4 is the size of the validation set

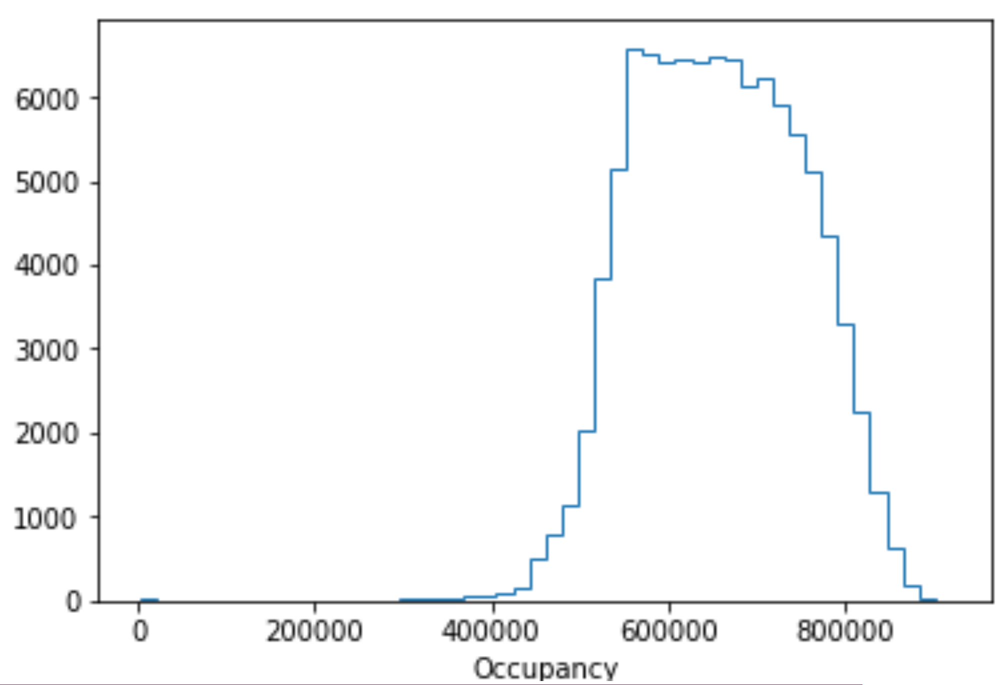
Preparing the data: PU normalization

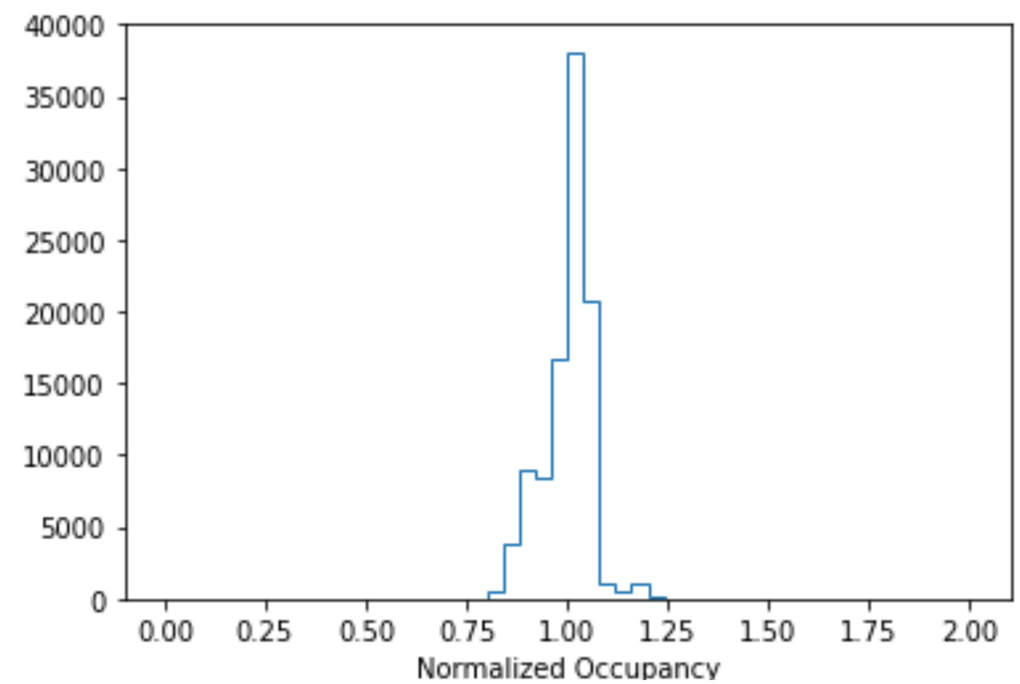
- **Crucial step**: Normalize images wrt PU to improve sensitivity and generalization over different fill conditions.
- Pileup (PU) dependence is removed **—>** as occupancy is determined by the selective readout and PU, while the selective readout is not PU dependent



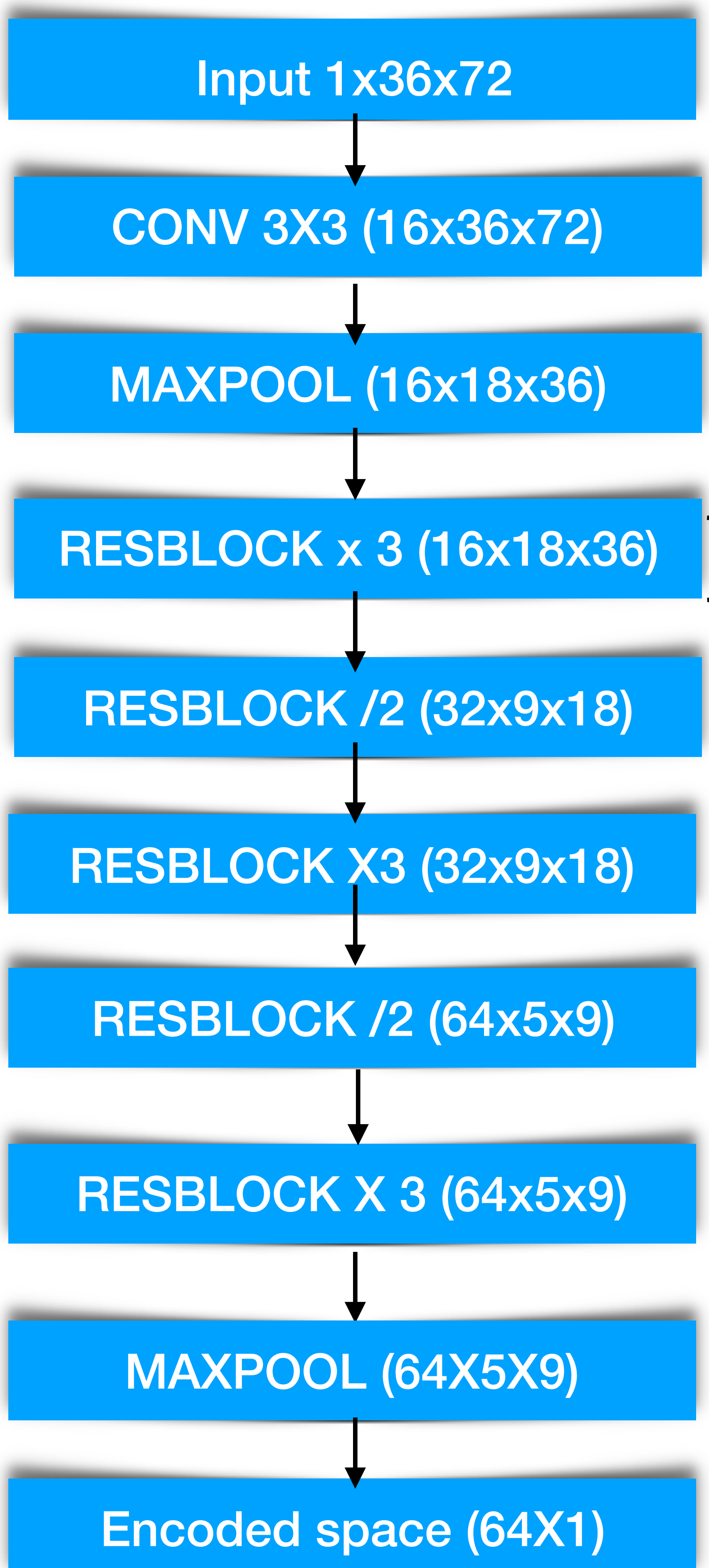


Occupancy distribution, before PU normalization after PU normalization

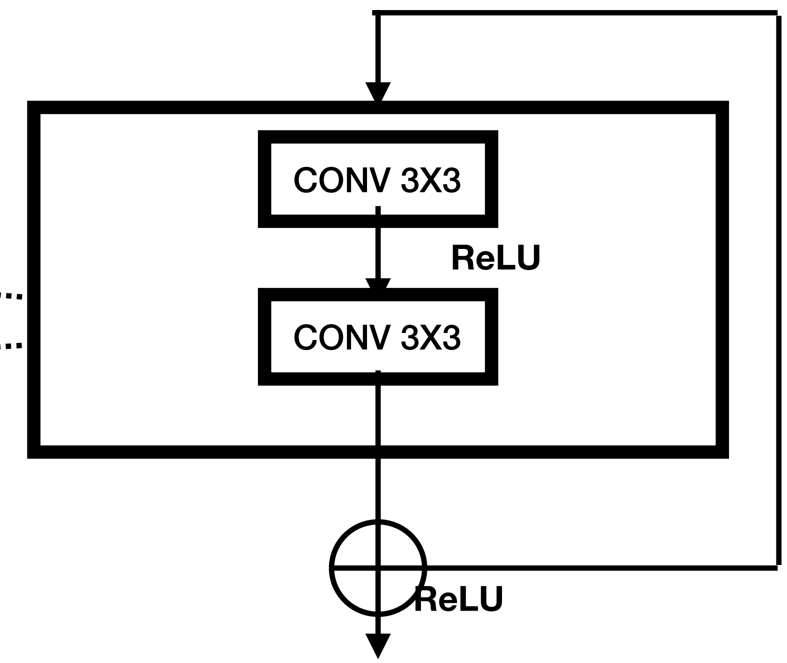




The Encoder Network



ResBlock



- **Decoder network is the reverse of this**