Herrmann, Julian
IKTP Dresden

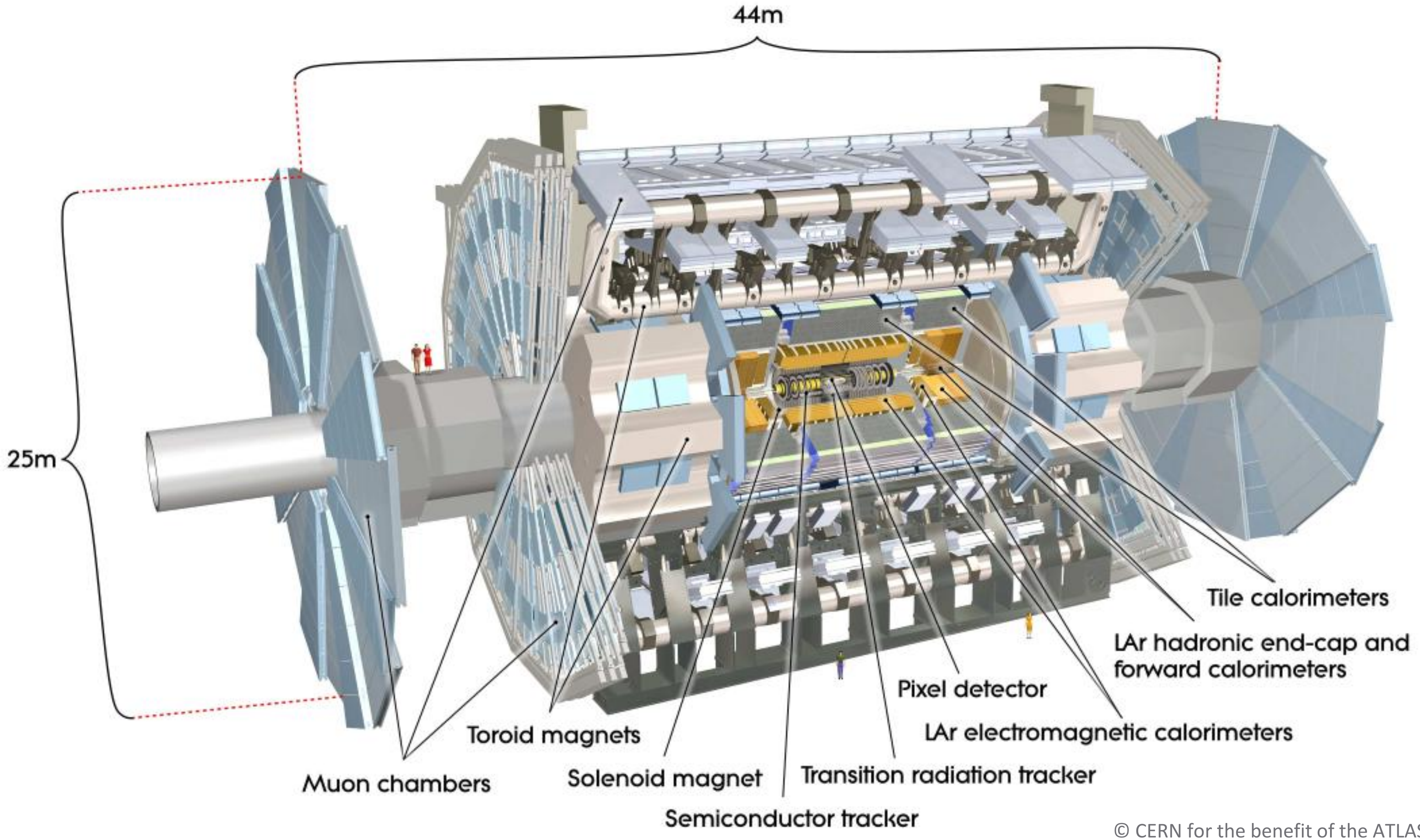
Unsupervised Detection of Noise Bursts in Simulated LAr Signals with Autoencoders

TA5-WP2/WP5: Working meeting - neural networks on FPGAs

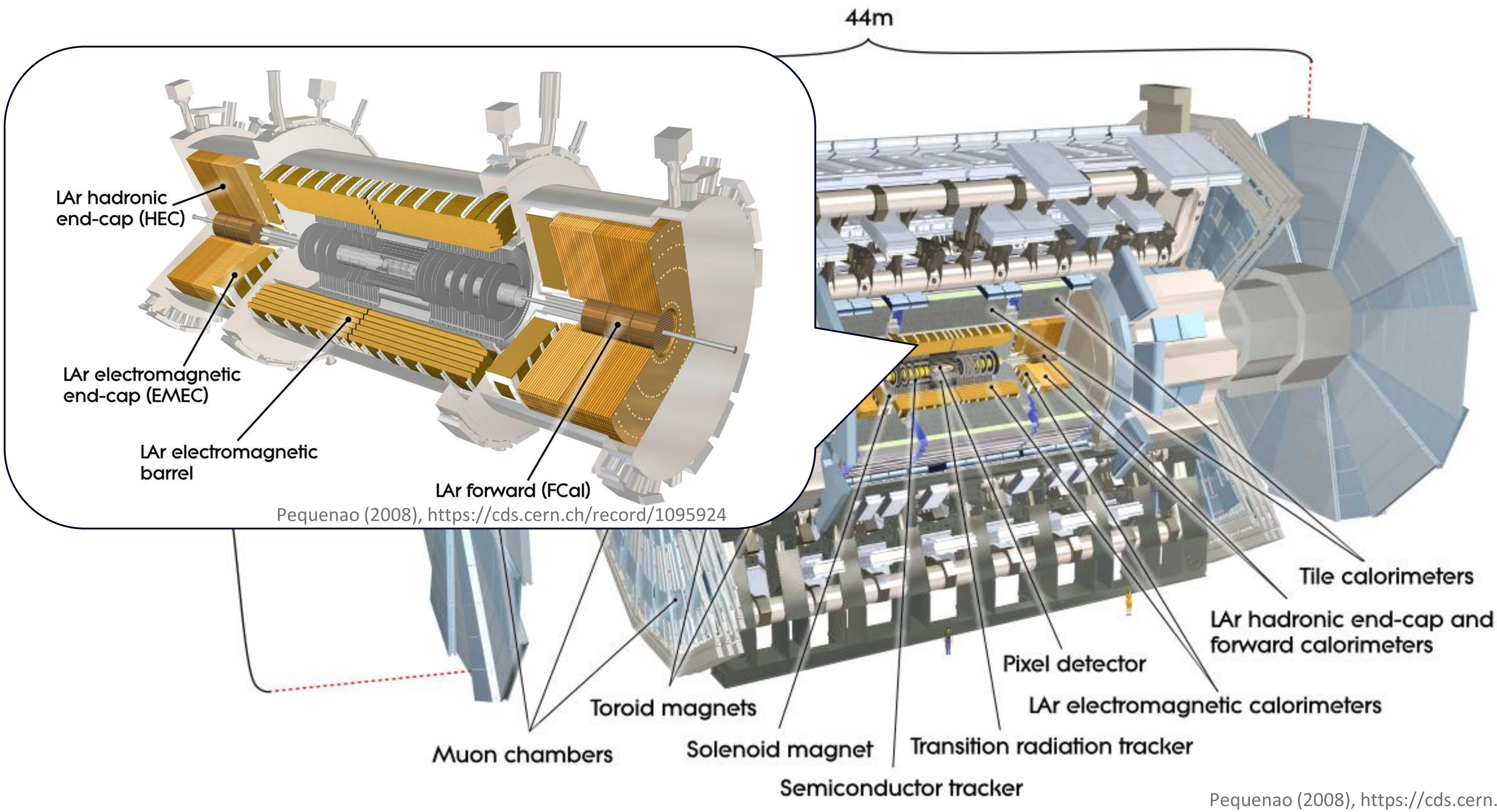
19th May 2025

https://cds.cern.ch/images/CERN-EX-0511013-01/file

ATLAS Detector



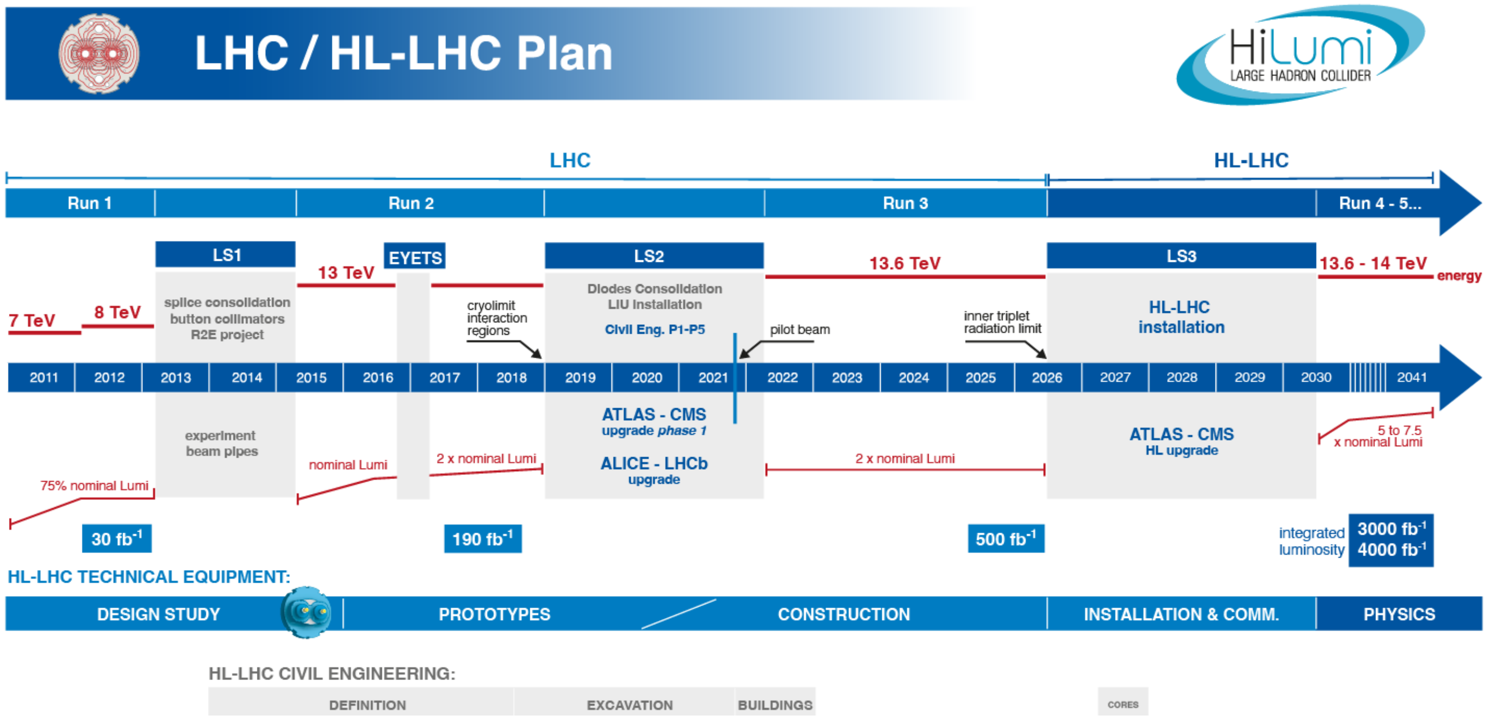
ATLAS Detector



Pequenao (2008), https://cds.cern.ch/record/1095924

HL-LHC Upgrade

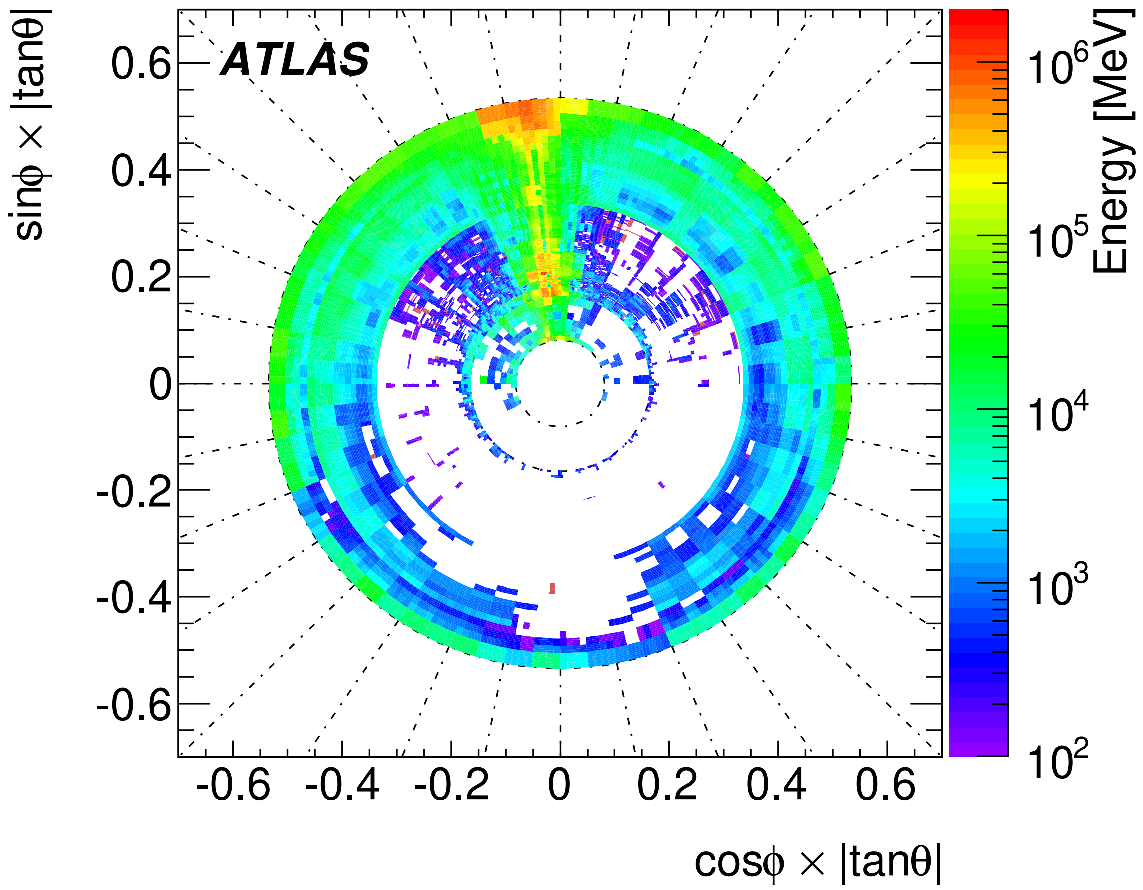
- **Higher data rate**
- **New readout electronics**
- **More computing power**



https://project-hl-lhc-industry.web.cern.ch/sites/default/files/inline-images/HL-LHC_Plan_January2025.png

Noise Bursts (NB)

- **Highly energetic**
- Stretch over **large detector regions**
- **Coherent Gaussian noise**
- **Short events** in the order of μs
- Likely due to **High Voltage cables** in the LAr purity system



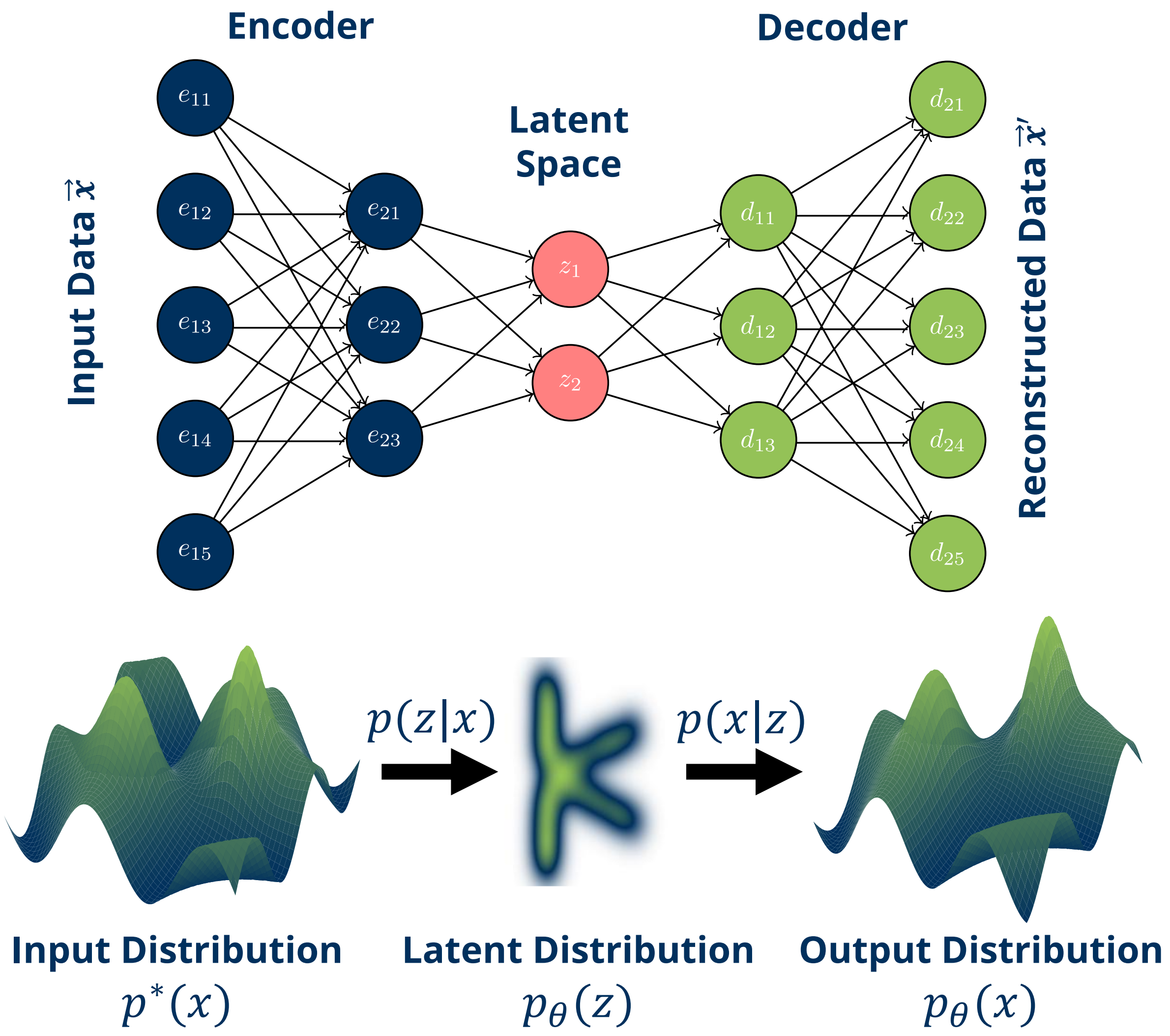
ATLAS Collaboration, CERN-PH-EP-2014-045

Autoencoders

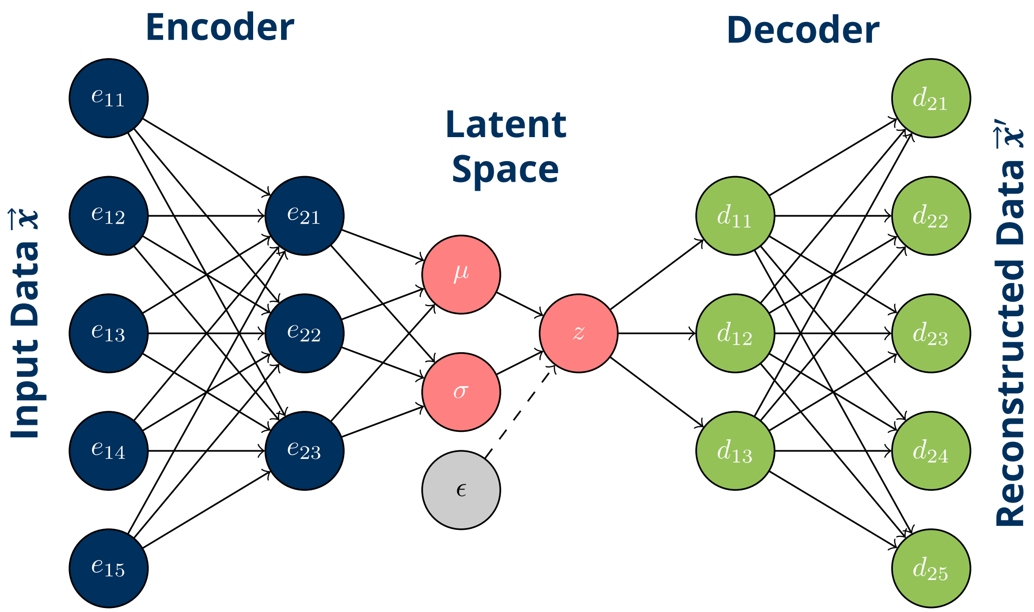
- **Goal:** Reconstruct input data in training
- "Bottleneck" ensures loss of information during encoding process

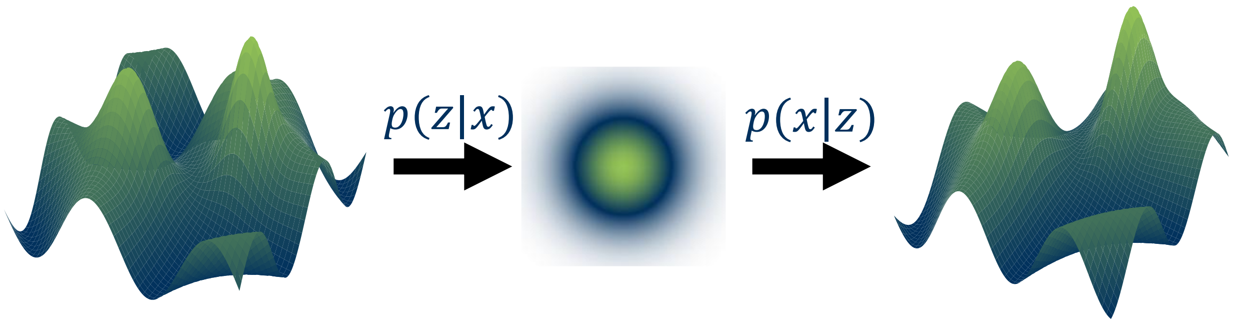
  → e.g. $\mathrm{MSE}(\vec{x}, \vec{x}')$
- **Anomaly detection:** AE fails to find a good reconstruction of anomalous data

  → Threshold for loss term



Variational Autoencoders





Input Distribution $p^*(x)$

Latent Distribution $p_\theta(z) \sim \mathcal{N}(0, \mathbf{I})$

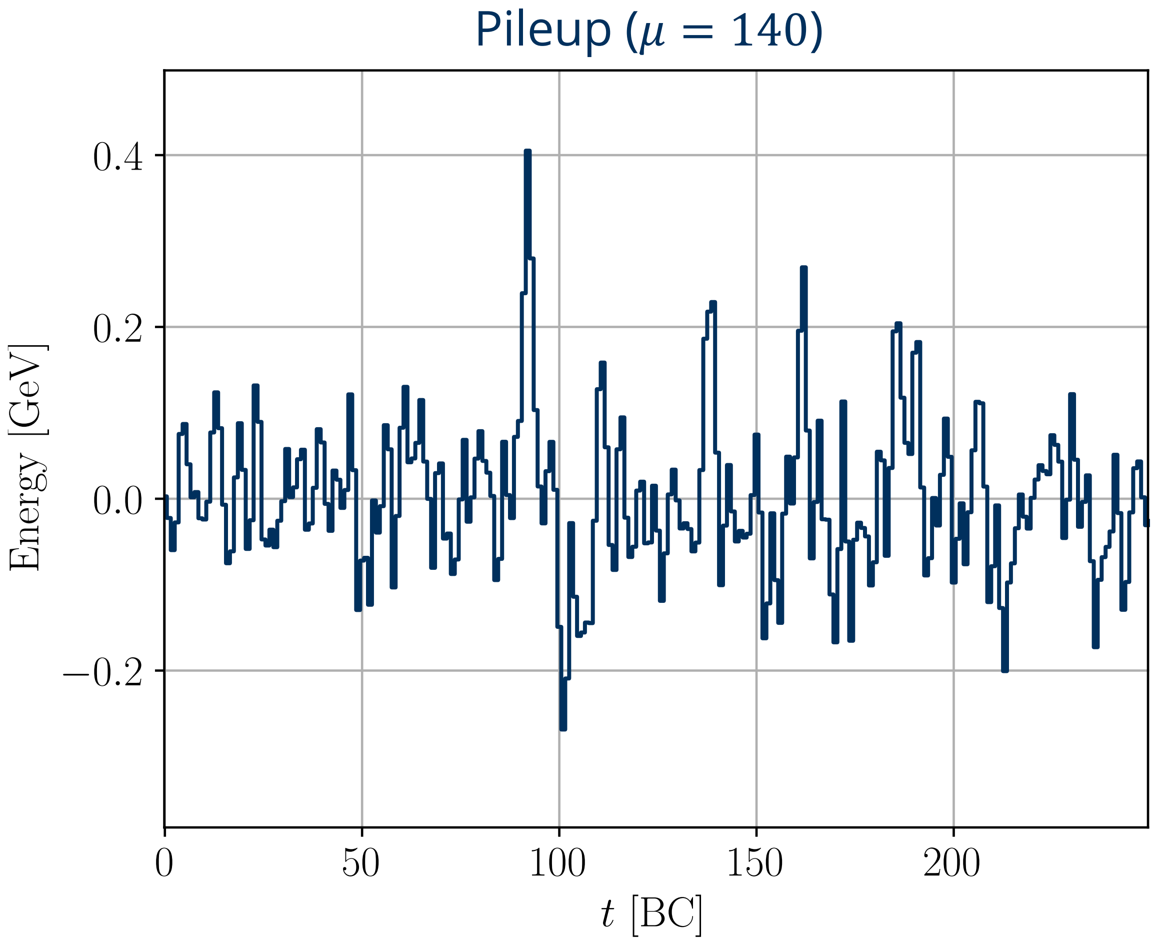
Output Distribution $p_\theta(x)$

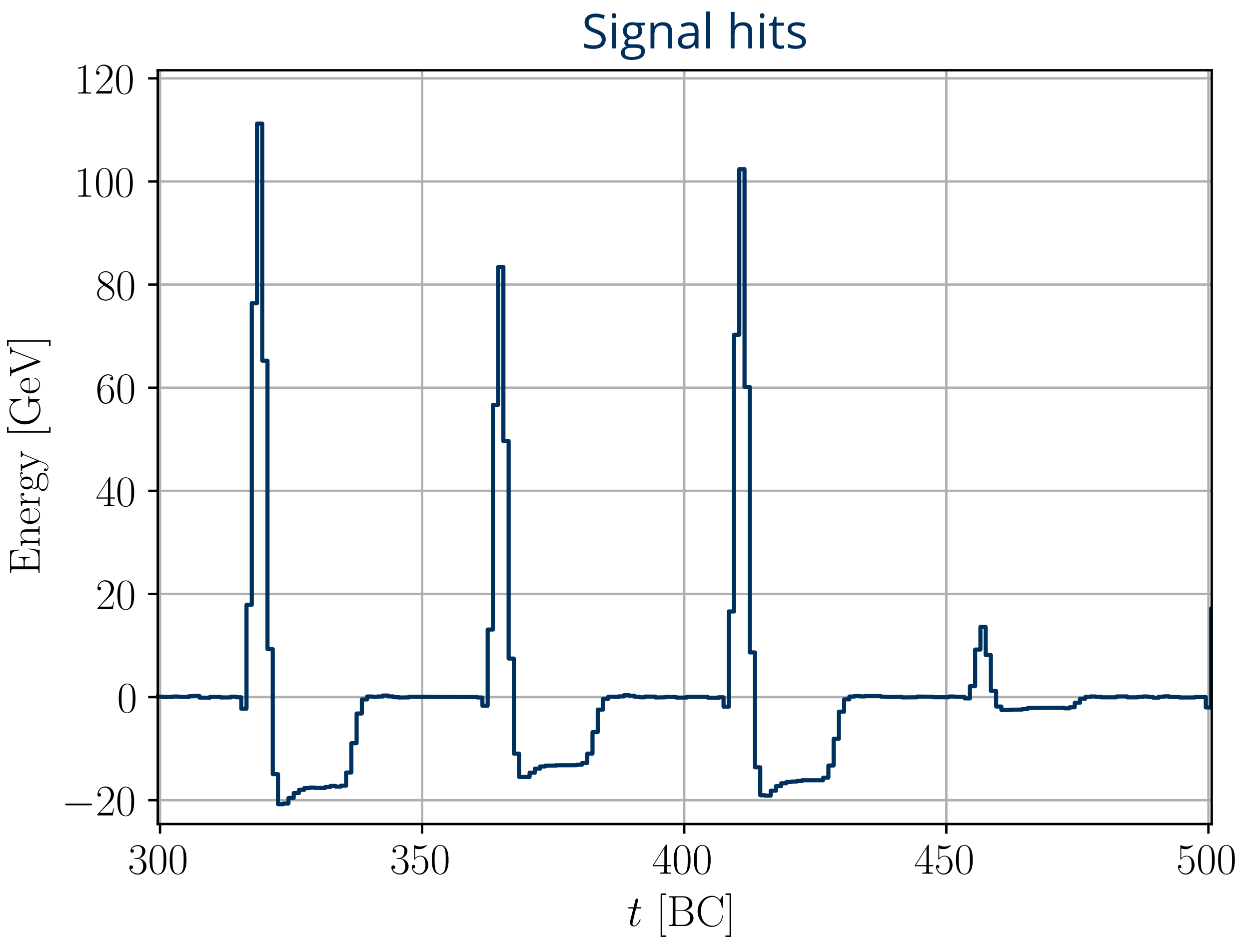
Loss function:

$$\mathcal{L} = (1-\beta)\mathcal{L}_{\text{Rec}}(\vec{x}', \vec{x}) + \beta \cdot D_{\text{KL}}(\mathcal{N}(\mu, \sigma) | \mathcal{N}(0,1))$$

$$\mathcal{L} = (1-\beta)\mathcal{L}_{\text{Rec}}(\vec{x}', \vec{x}) - \frac{\beta}{2}(\log(\sigma^2) + 1 - \mu^2 - \sigma^2)$$

Training Dataset





Different signal scenarios alternate every 10000 BC

Data Preprocessing

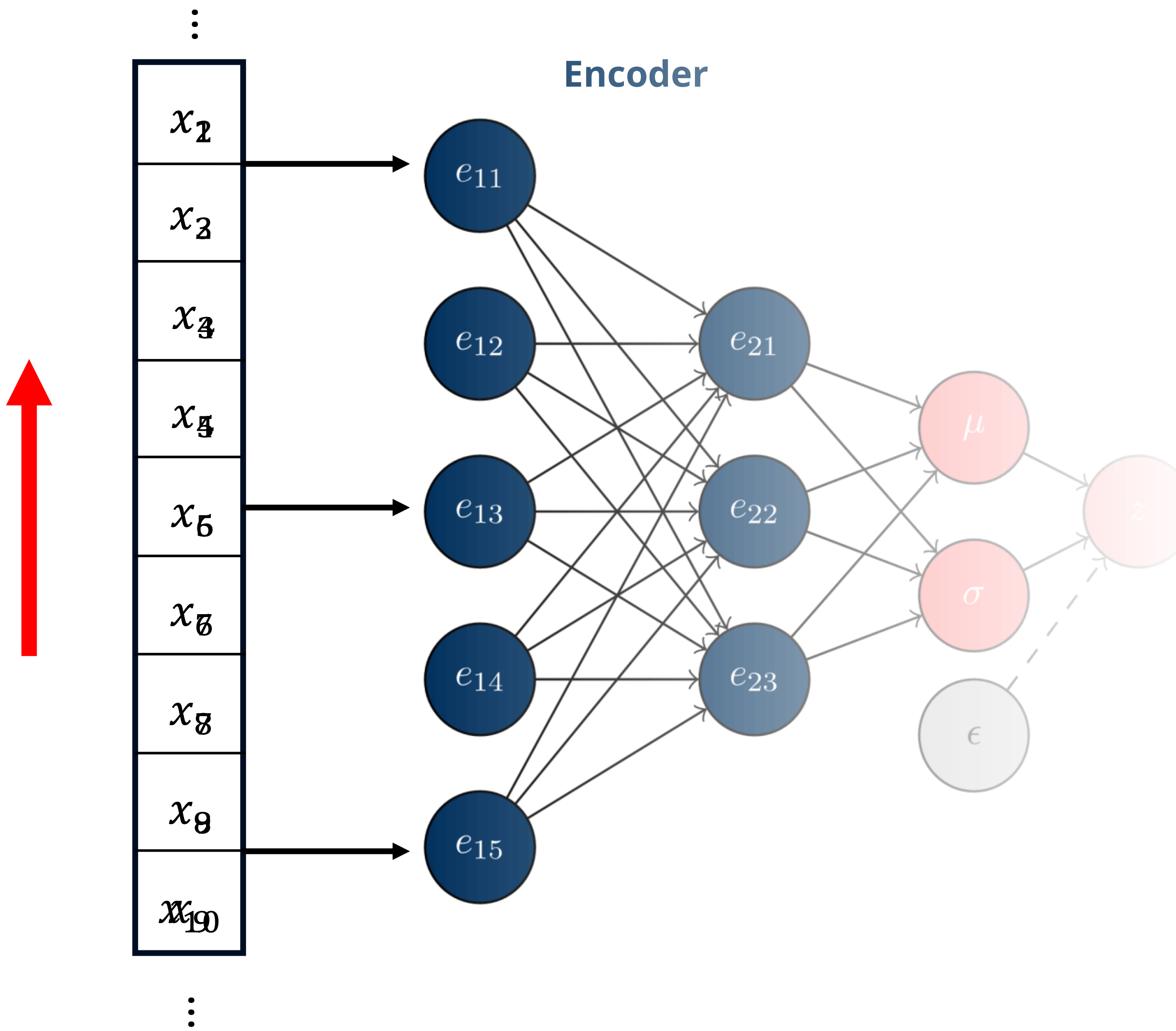
Normalization:

$$n(x_i) = \mathrm{sign}(x_i) \cdot \ln(1 + |x_i|)$$

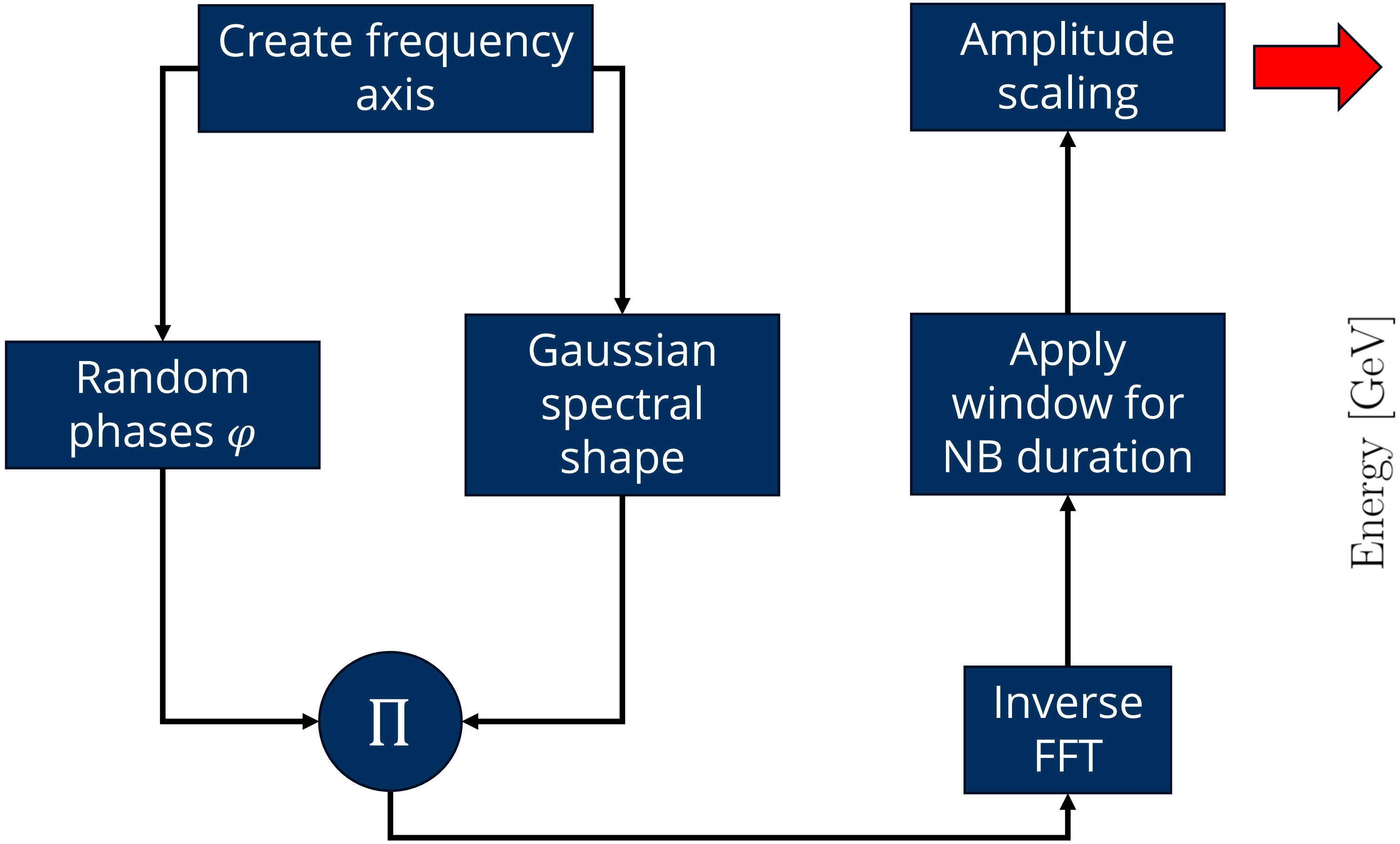
Arranging input data into sliding window

- Window size = 10

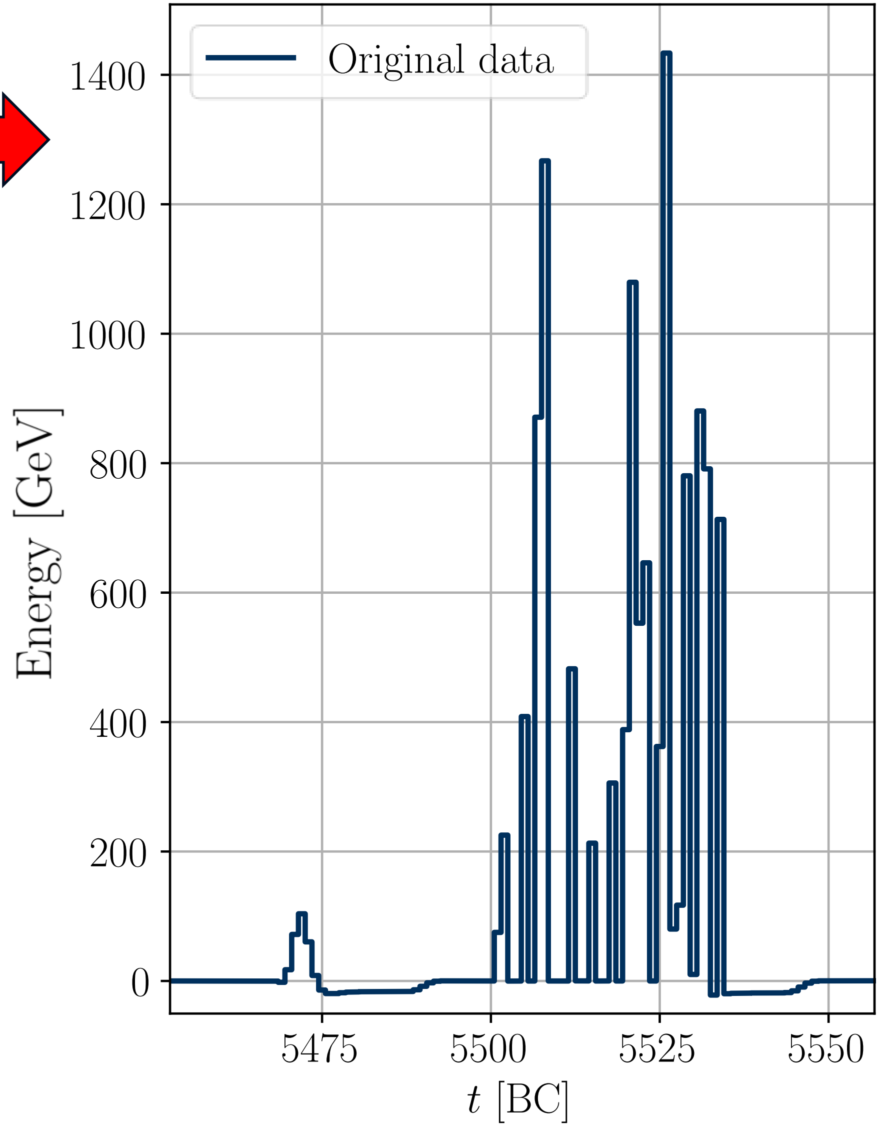
Time t →				
x_1	x_2	x_3	x_4	x_5
x_2	x_3	x_4	x_5	x_6
x_3	x_4	x_5	x_6	x_7
x_4	x_5	x_6	x_7	x_8
x_5	x_6	x_7	x_8	x_9



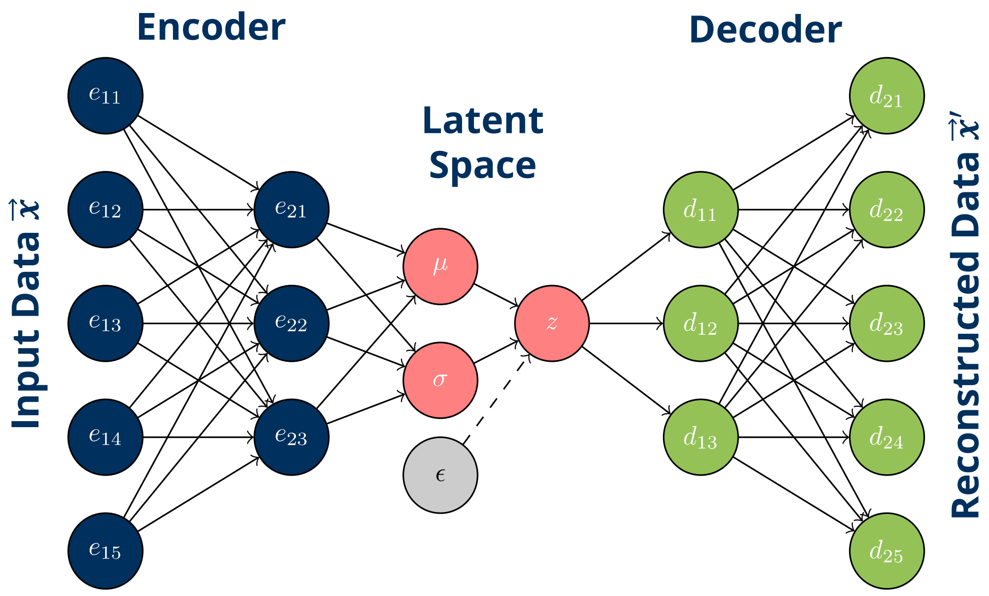
Simulation of Noise Bursts



$$c(f) = e^{-\frac{1}{2}\left(\frac{f}{f_{\text{peak}}}\right)^2} \cdot e^{i\varphi}$$



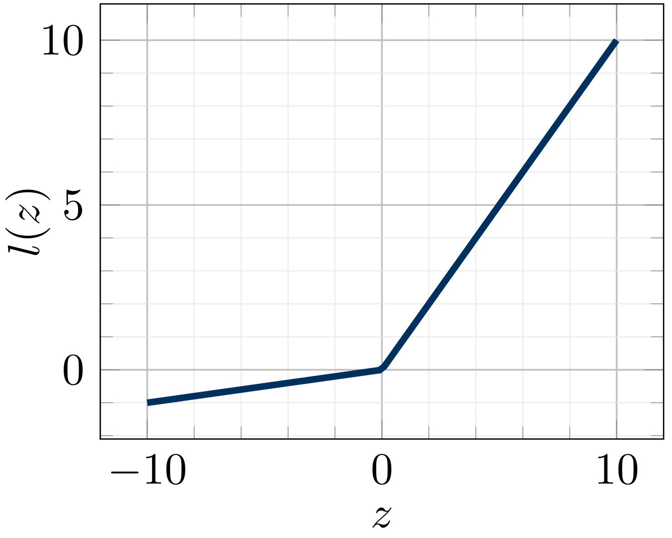
Finding the best network



Constrains for hyperparameter search:

1. Latent dim. < Input dim.
2. Number of nodes in the encoder must be decreasing from one layer to the next
3. Max. 500 parameters for en- and decoder

$$\mathcal{L}_{\text{Rec}} = \begin{cases} \frac{1}{N} \sum_i \frac{(\hat{o}_i - o_i)^2}{2}, & |\hat{o} - o| < 1 \\ \frac{1}{N} \sum_i |\hat{o}_i - o_i| - \frac{1}{2}, & |\hat{o} - o| > 1 \end{cases}$$

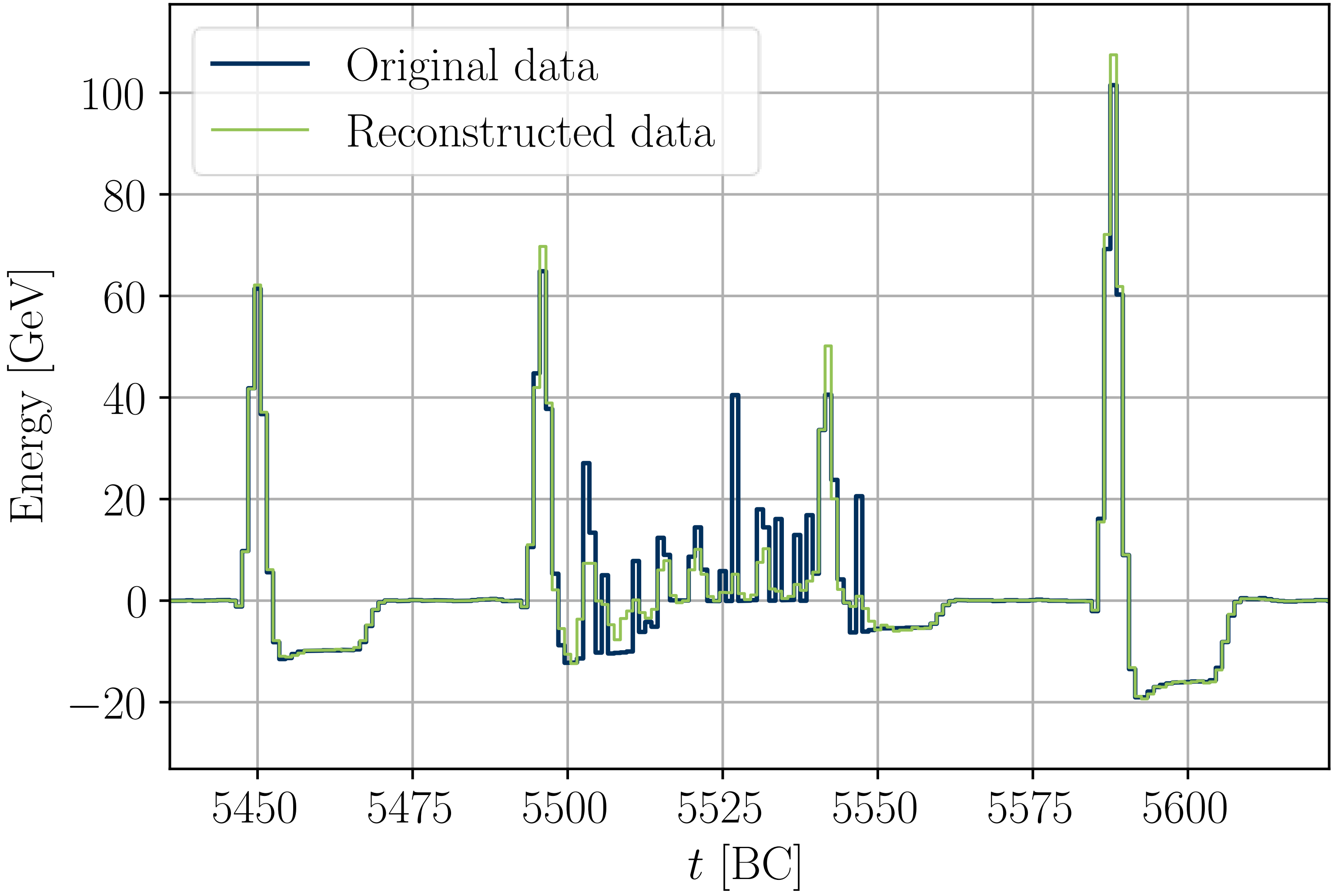


β	Batch Size	Optimizer	Initial LR	Activation
0.01	1000	Adam	$1 \cdot 10^{-3}$	Leaky ReLU

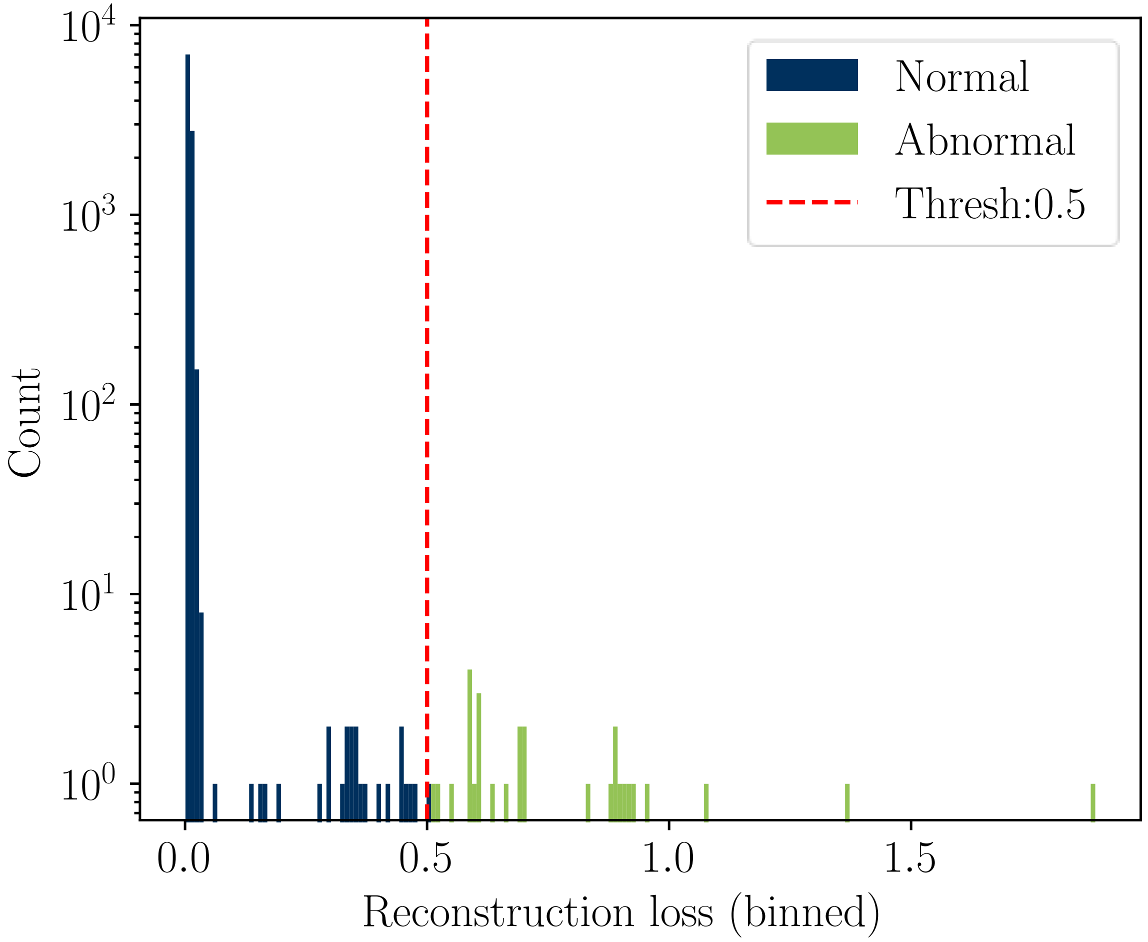
Hyperparameter optimization with asynchronous **Hyperband** search

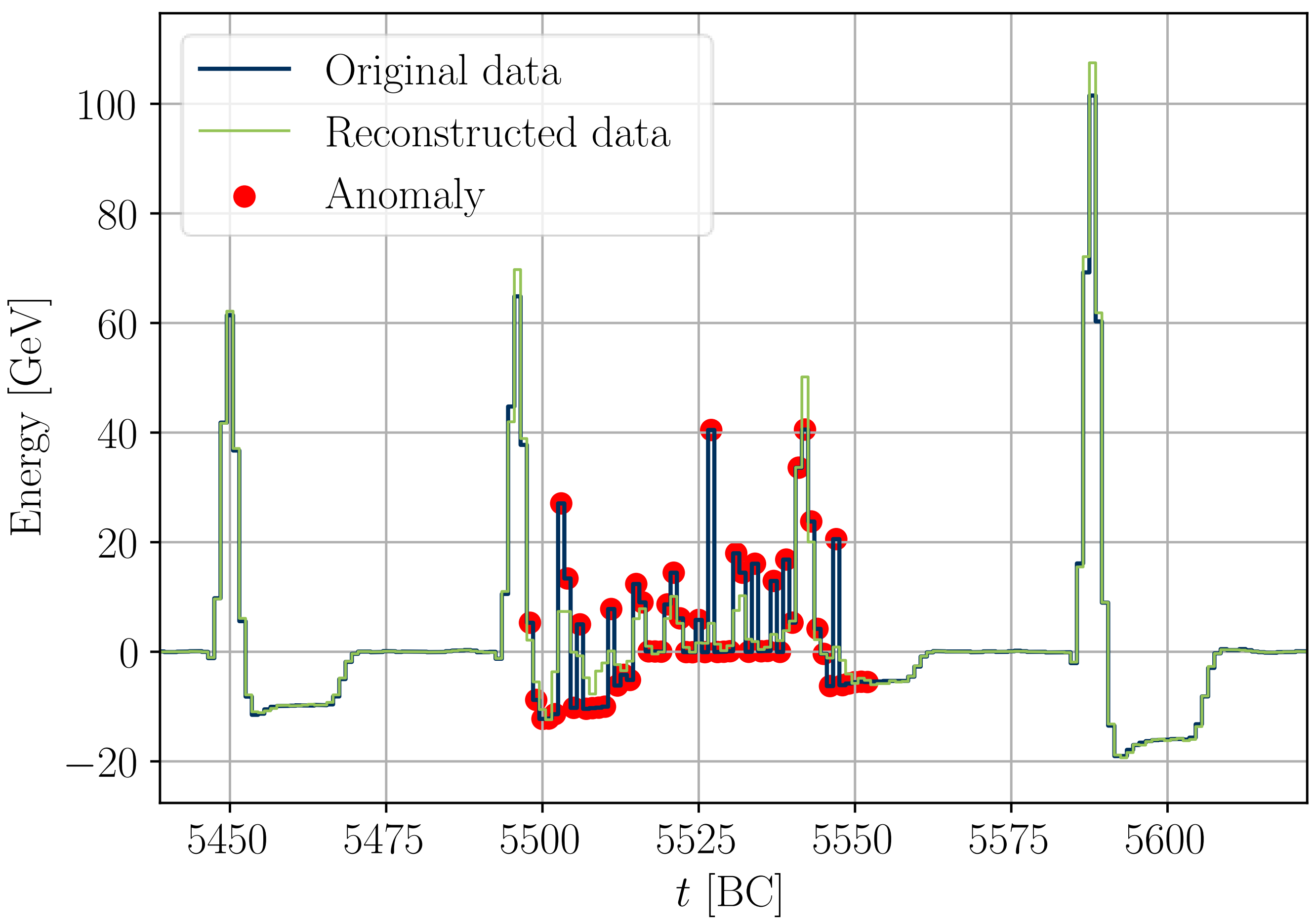
Selected Network

Parameter	Value
Input dim.	10
Dim. Layer 1	20
Dim. Layer 2	8
Latent dim.	6
Epochs	200

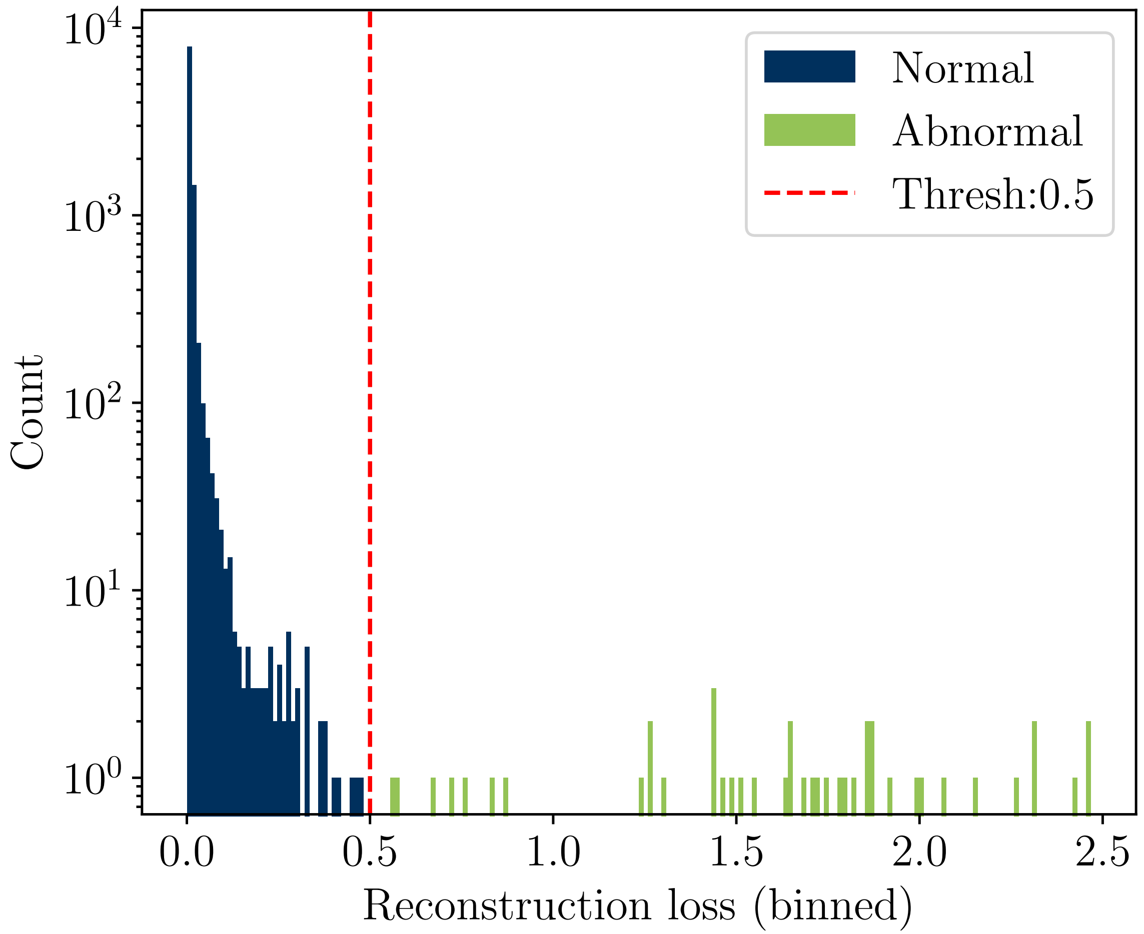


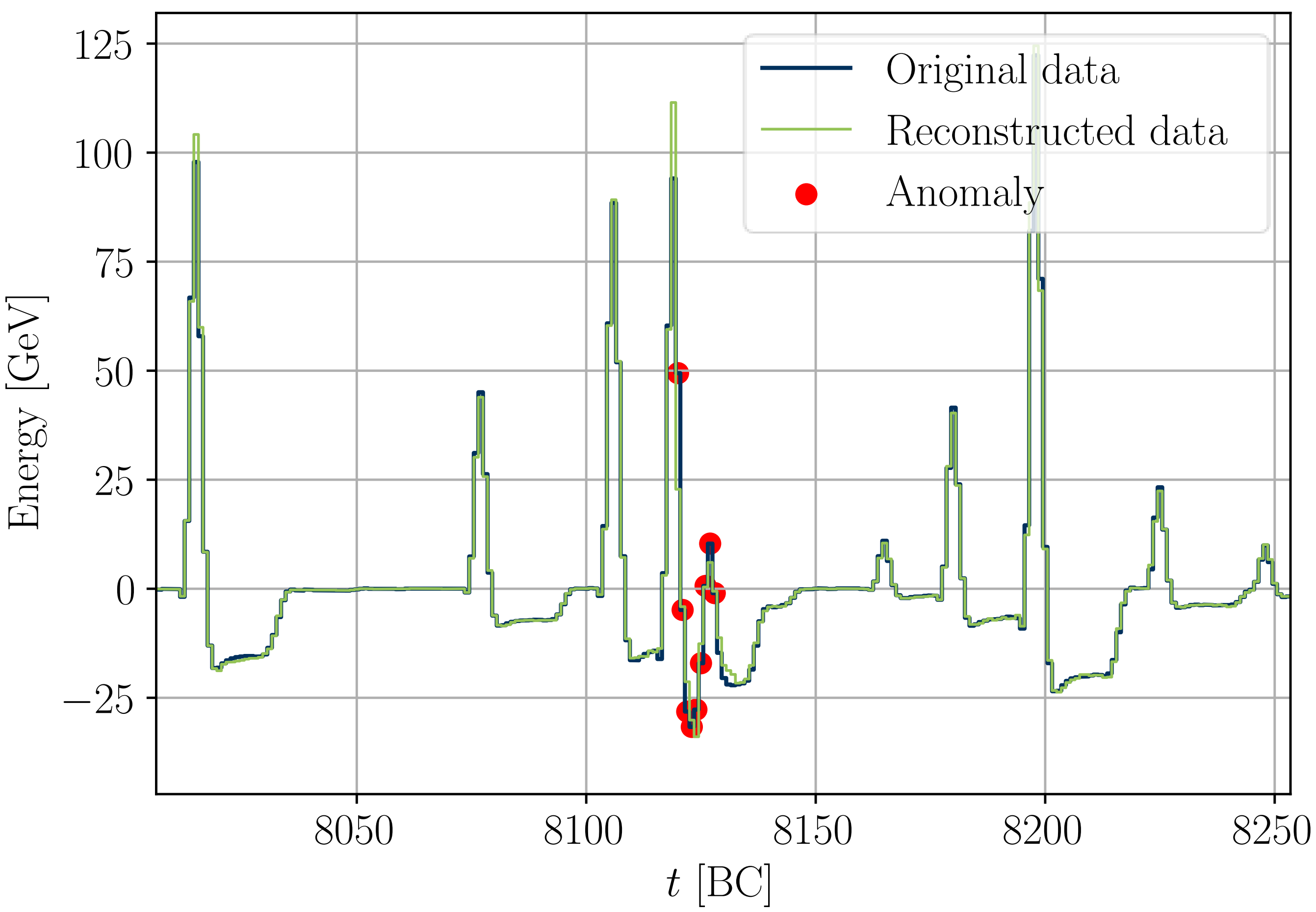
Selected Network



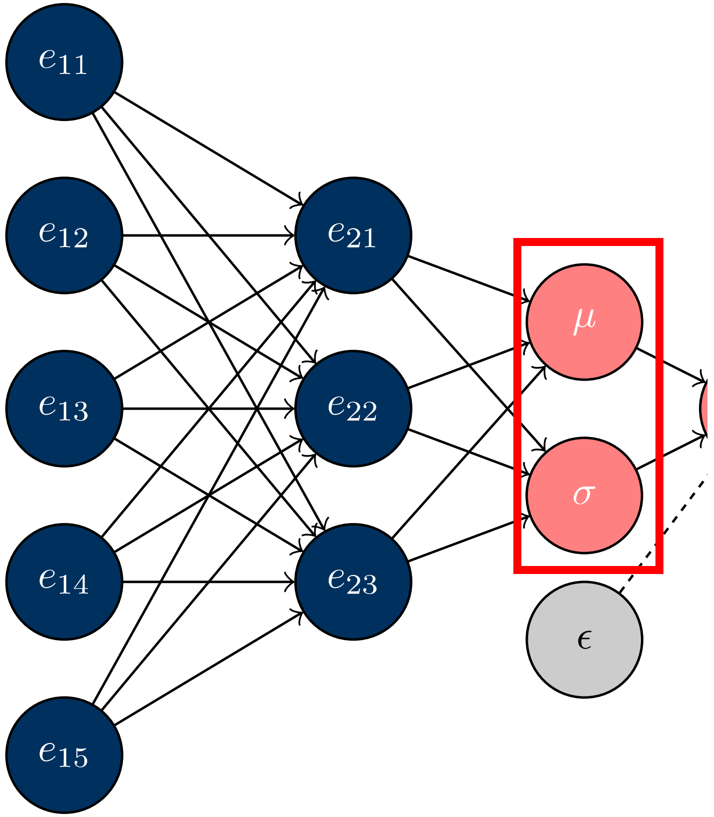


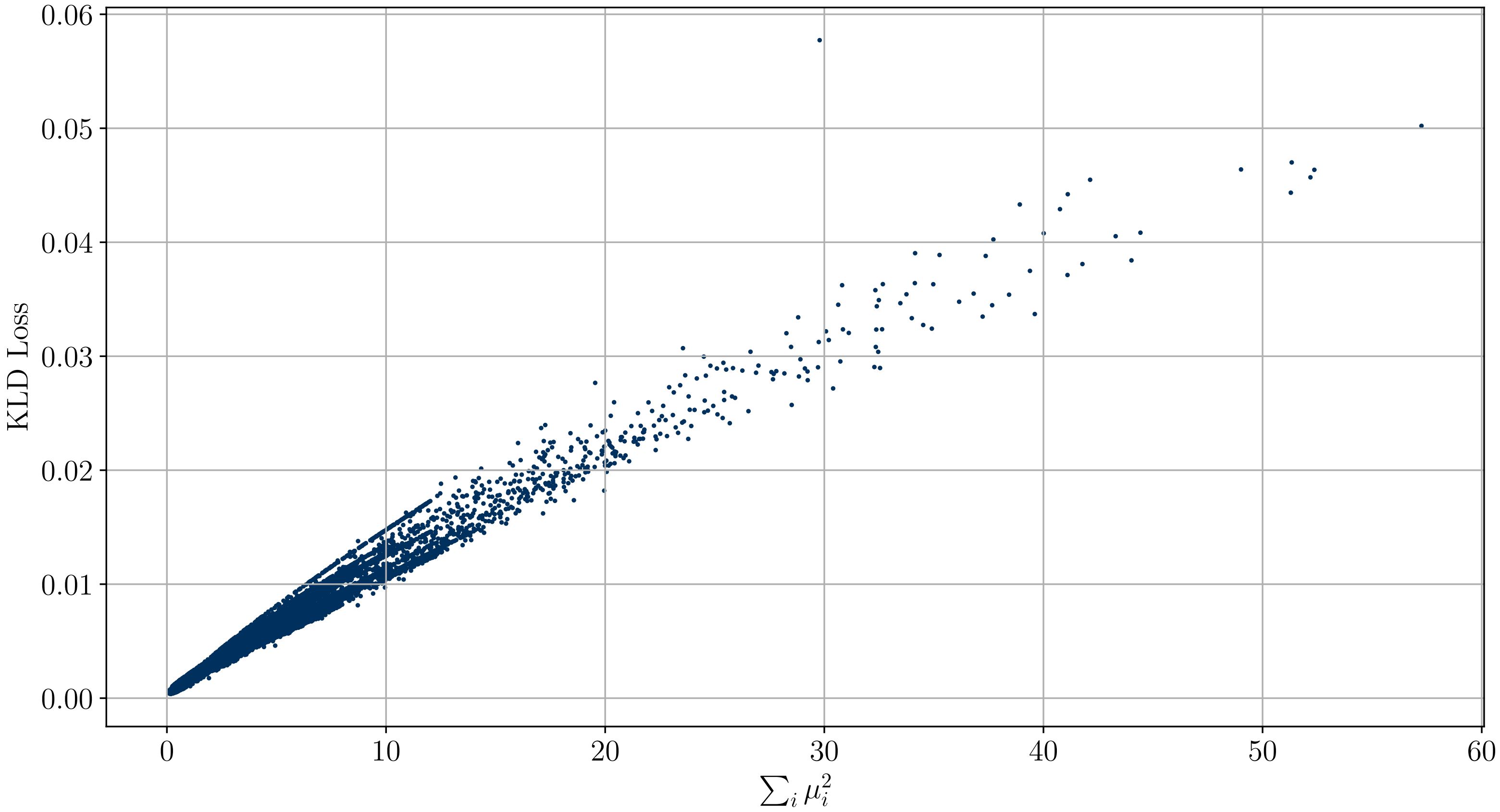
Selected Network





Splitting the VAE and tuning of β

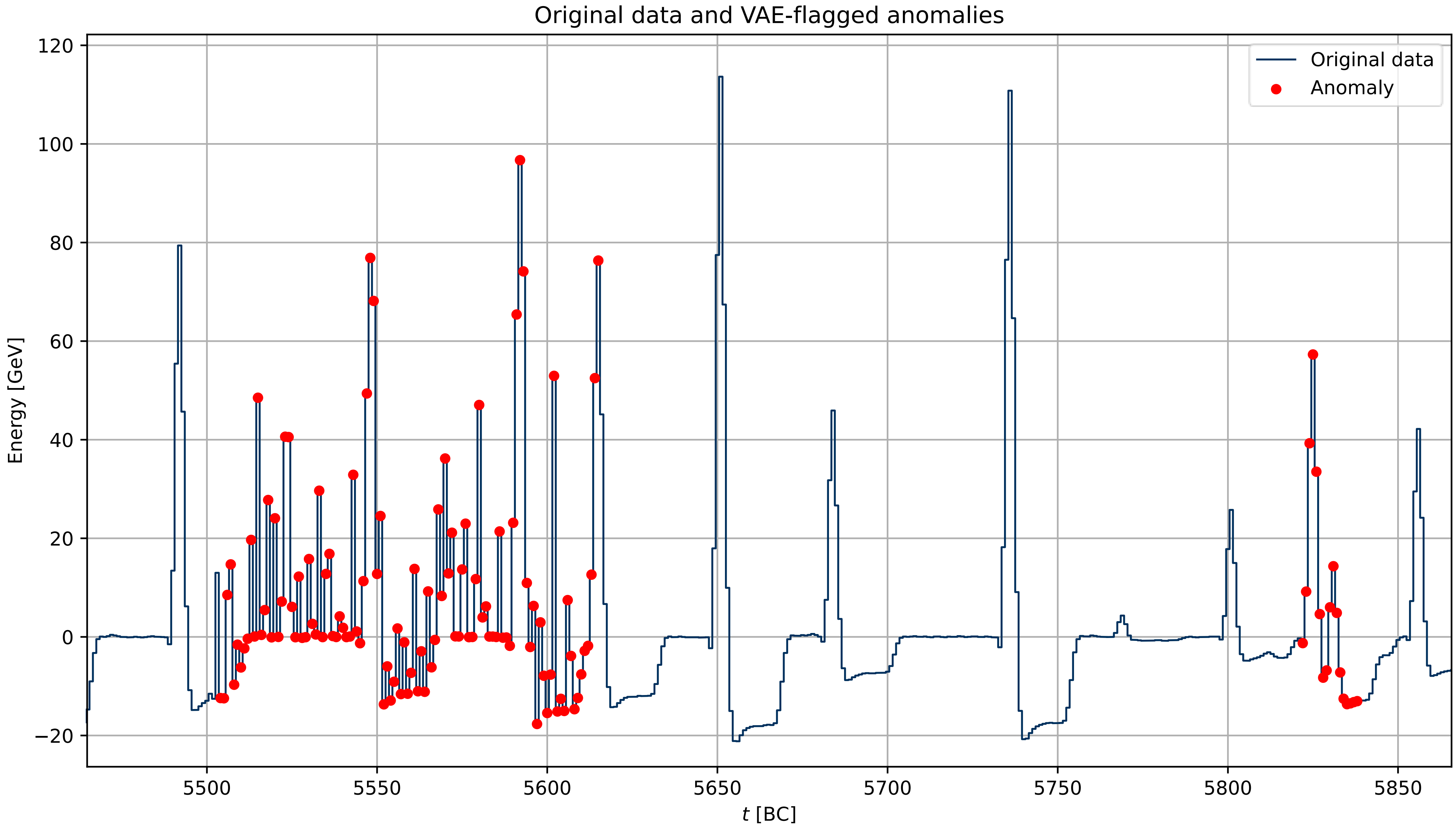




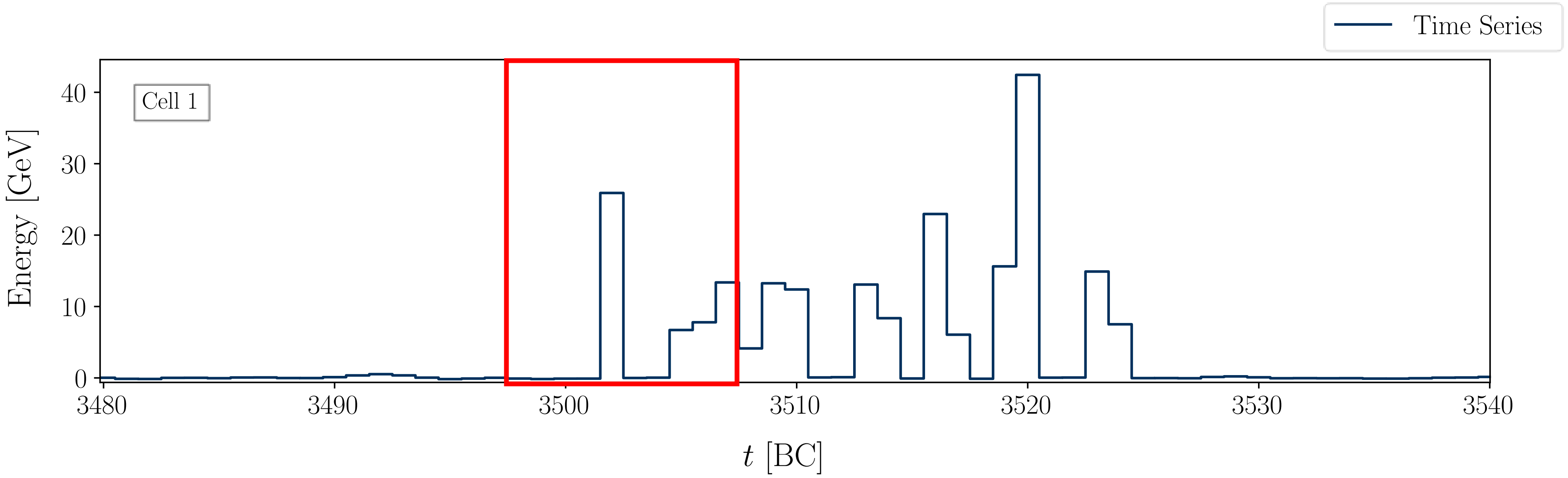
KLD Loss term:

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{2}(\mu^2 + \sigma^2 - 1 - \log(\sigma)) \rightarrow \mathrm{L} = \sum_i \mu_i^2$$

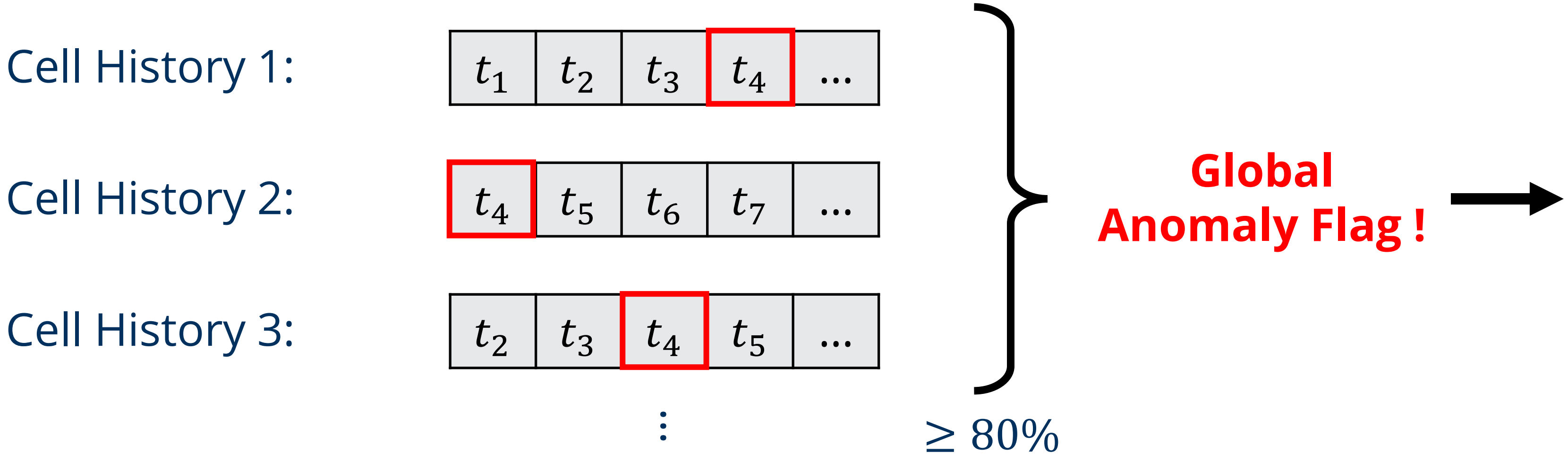
New score shows same behaviour as full VAE



Multiple cell Anomaly Detection



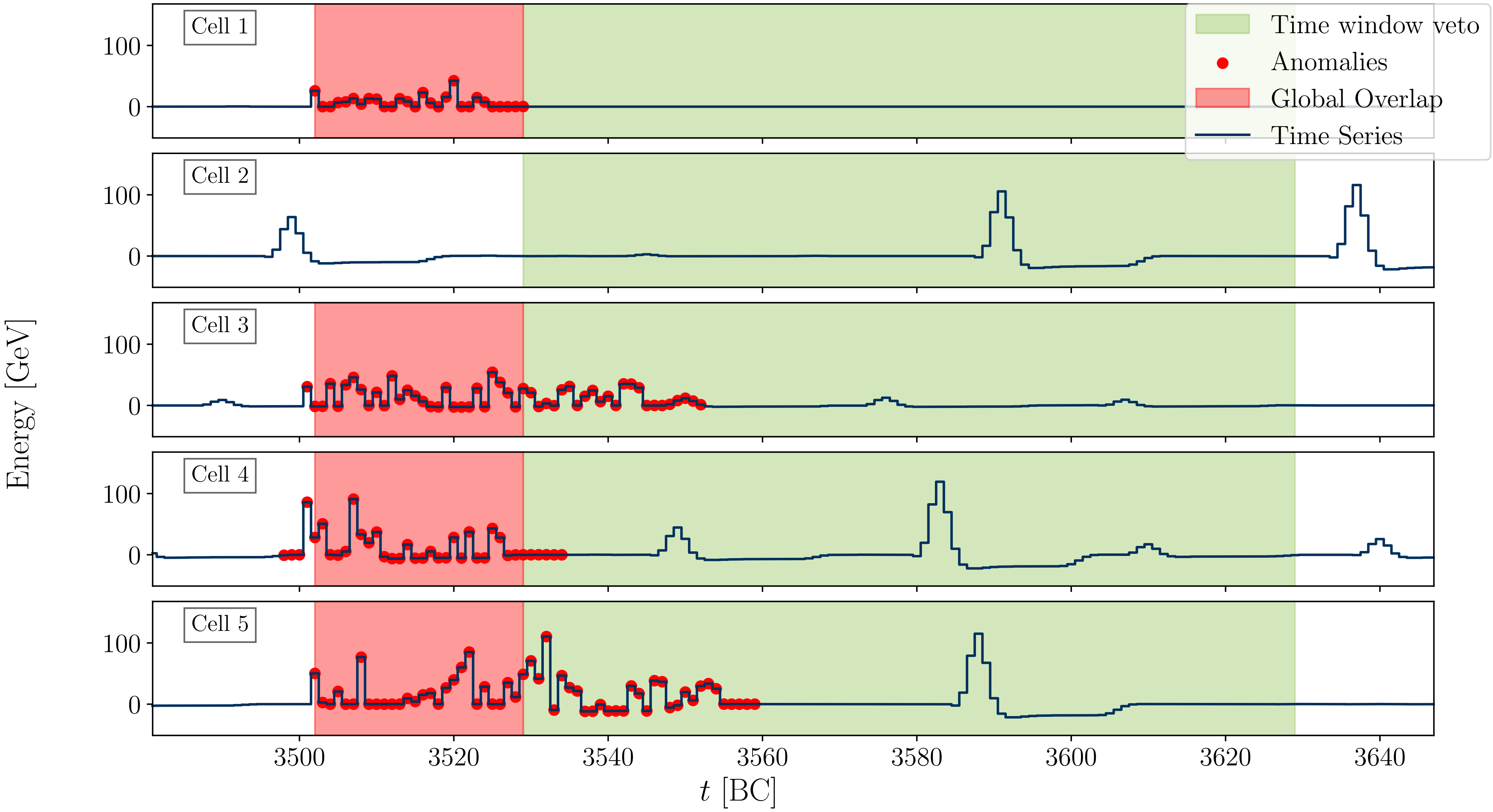
No Local Anomaly



Time Window

Every anomaly flagged regardless of overlap

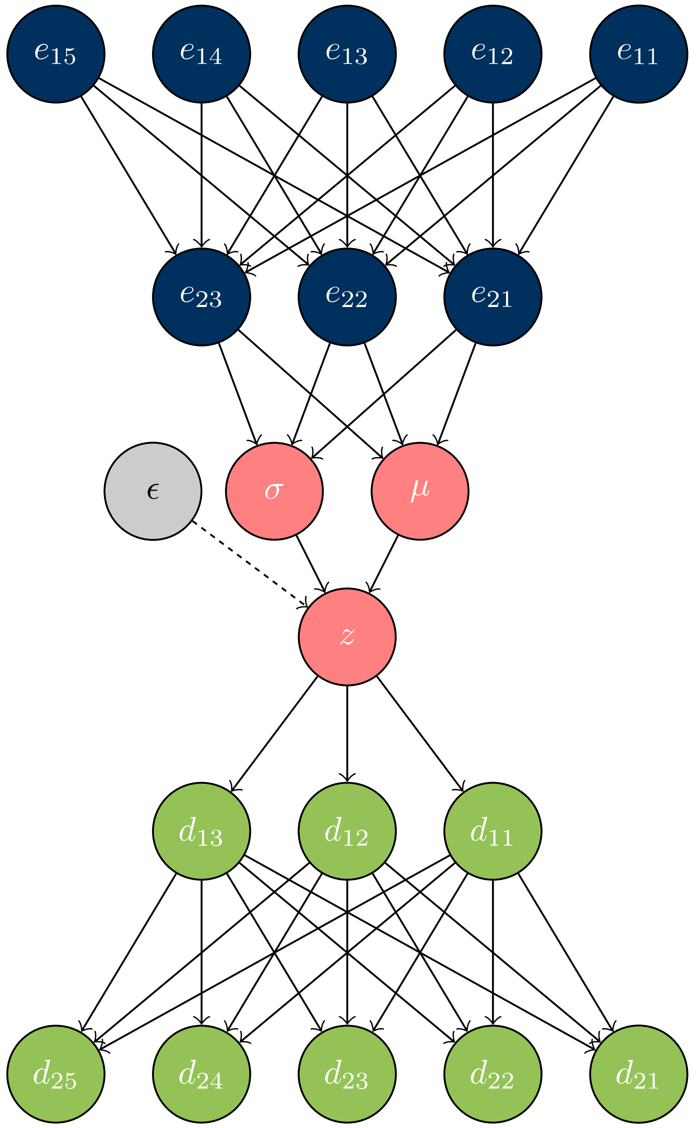
Multiple cell Anomaly Detection: Results



Summary and Outlook

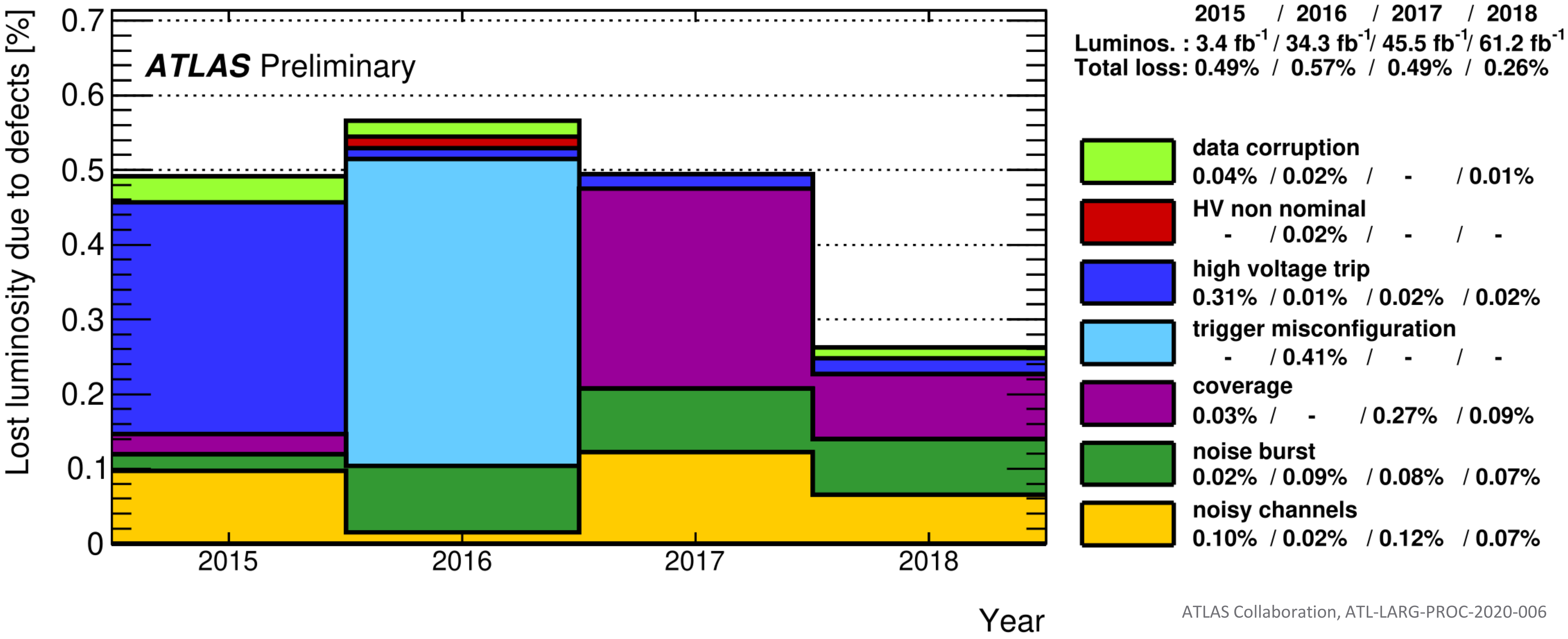
- Simulation of Noise Burst signals using FFT
- Training of Variational Autoencoder for anomaly detection
- Splitting of VAE to reduce computing resources
- Algorithm for multiple cell anomaly detection

- Testing simulated NB and VAE on real data
- Comparison with other classical anomaly detection methods
- Optimization for FPGA (further minimization of network size)
- Training on additional signal properties (Q-Factor)



Backup Slides

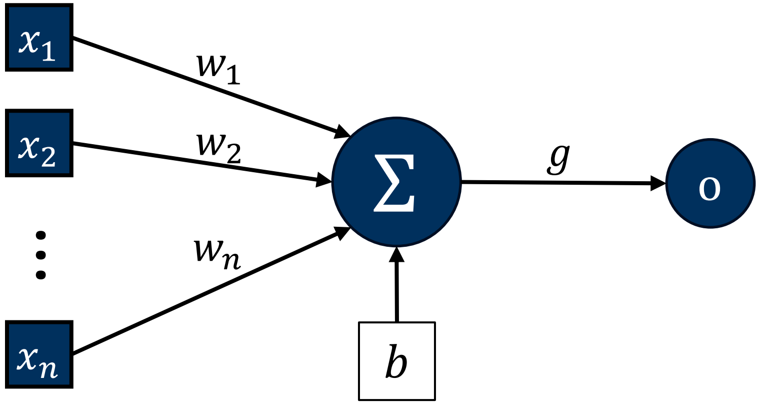
Lost Luminosity in Run 2



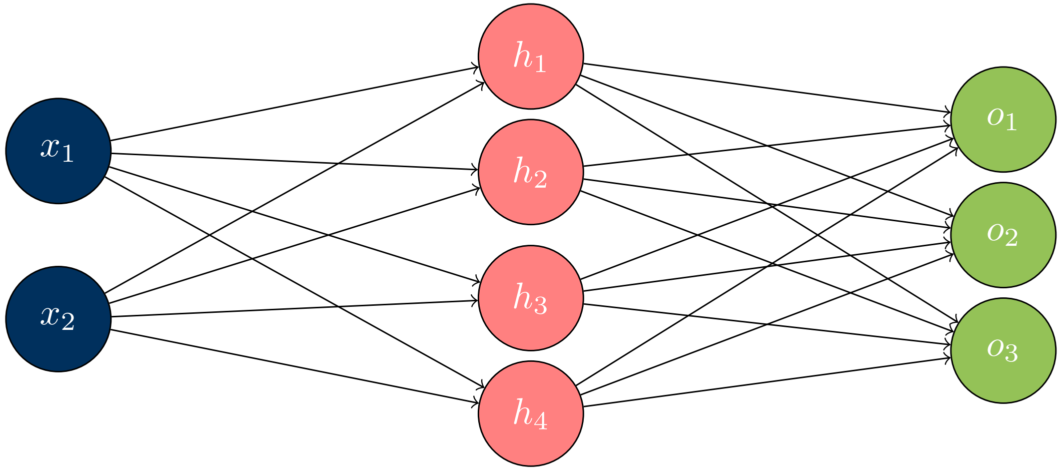
ATLAS Collaboration, ATL-LARG-PROC-2020-006

Machine Learning

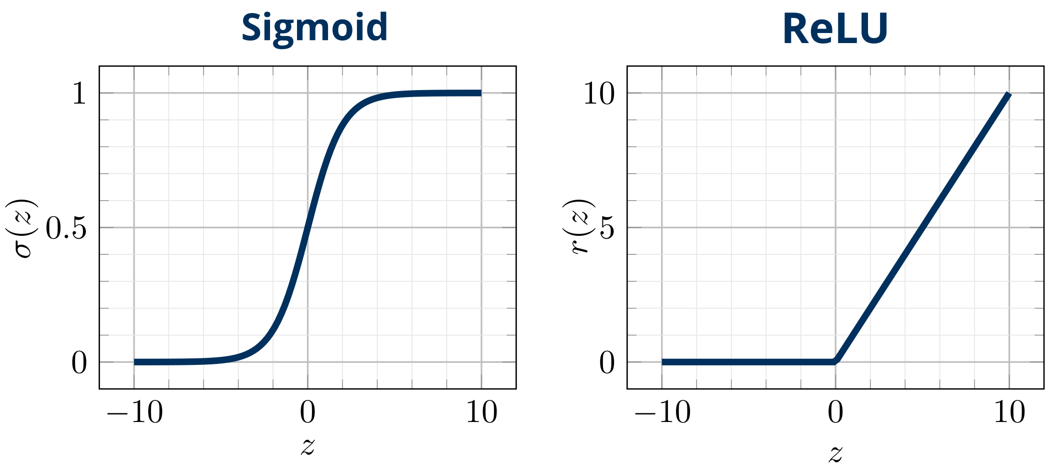
Single Node: $o = g(\sum_i w_i \cdot x_i + b)$



Input Layer **Hidden Layer** **Output Layer**



$H(\vec{X}) = g_1\left(W_1\vec{X} + \vec{b}_1\right)$ $\quad O(\vec{H}) = g_2\left(W_2\vec{H} + \vec{b}_2\right)$



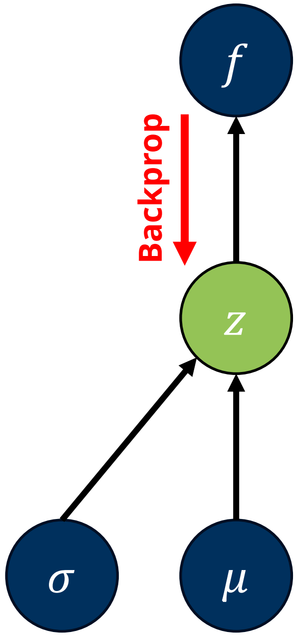
Loss Functions: desired output $\hat{O}$

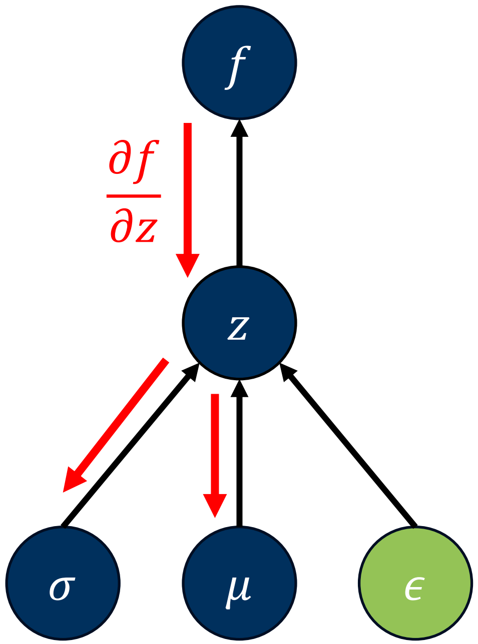
$$\mathcal{L}_{\text{MAE}}\,(\hat{O}, O) = \frac{1}{N} \sum_i \left|\hat{O}_i - O_i\right|$$

$$\mathcal{L}_{\text{MSE}}\,(\hat{O}, O) = \frac{1}{N} \sum_i \left(\hat{O}_i - O_i\right)^2$$

Reparameterization trick

How can we perform backpropagation through a sampling operation?





Deterministic Node

Stochastic Node

$$z = \mu + \sigma \odot \epsilon$$

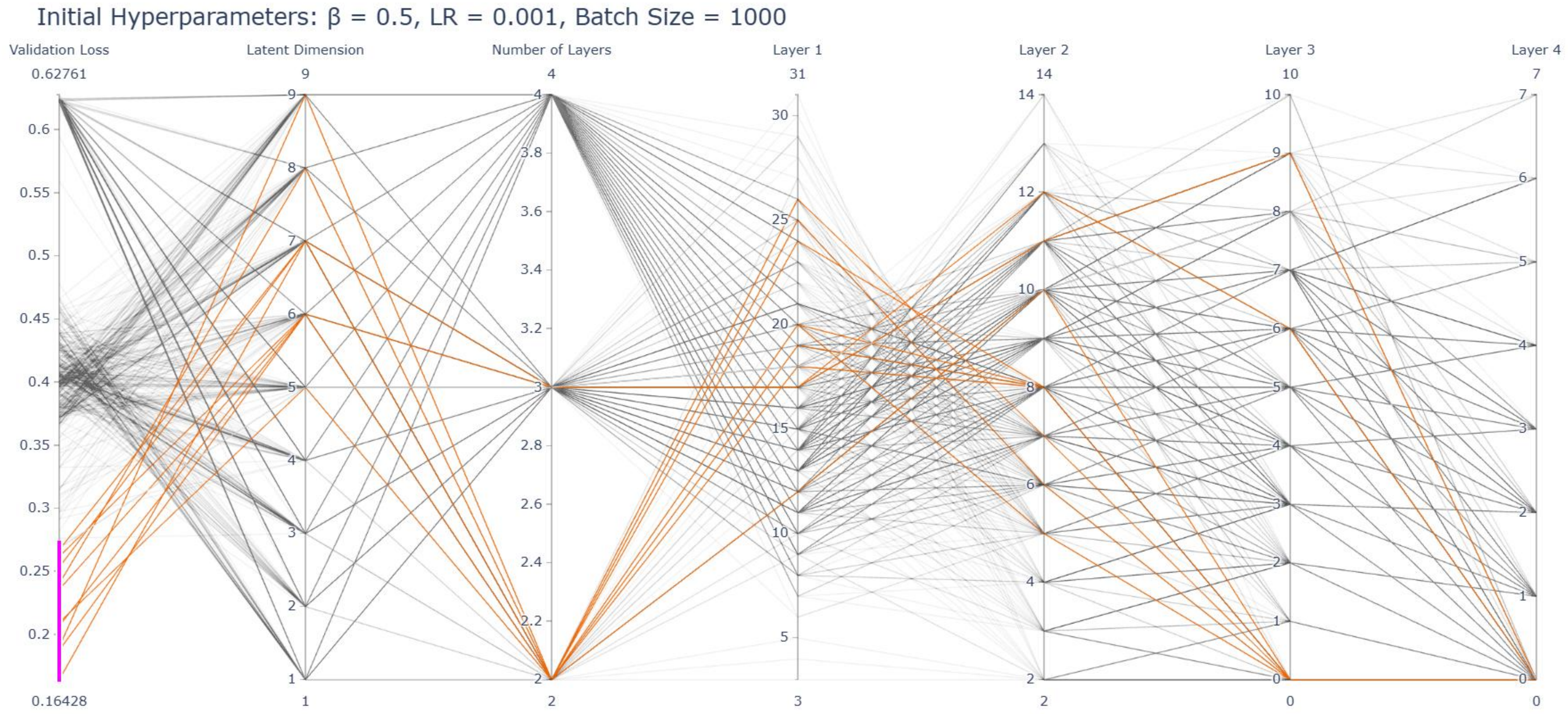
Training Dataset

Both Pileup ($\mu = 140$) and signal hits

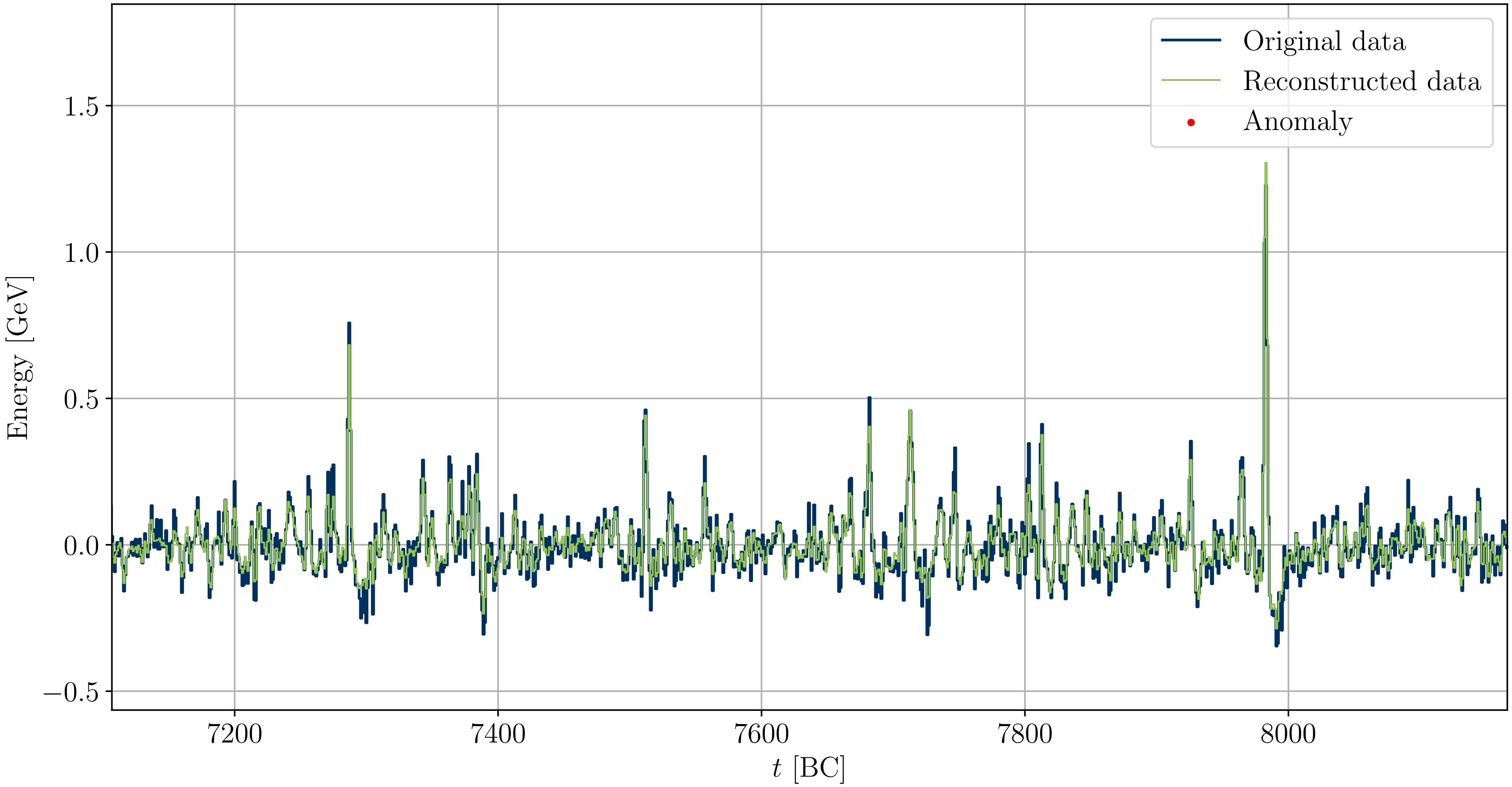
Signal type	Energy range [GeV]	Number of input units
Constant gap ($\Delta t_{\text{gap}} = 45$ BC)	≈[0, 150]	100
Gaussian gap ($\mu_{\text{gap}} = 30$ BC, $\sigma_{\text{gap}} = 10$ BC)	≈[0, 20]	100
Uniform gap (t_{gap} between 0 and 70 BC)	≈[0, 150]	100
Only-pileup	≈[0, 5.5]	100

- Bunch length = 5 cm, BTS = 1 (all bunches filled)
- Scenarios alternate every 10000 BC
- Smaller Validation Dataset with different signal arrangement

Hyperparameters optimization results

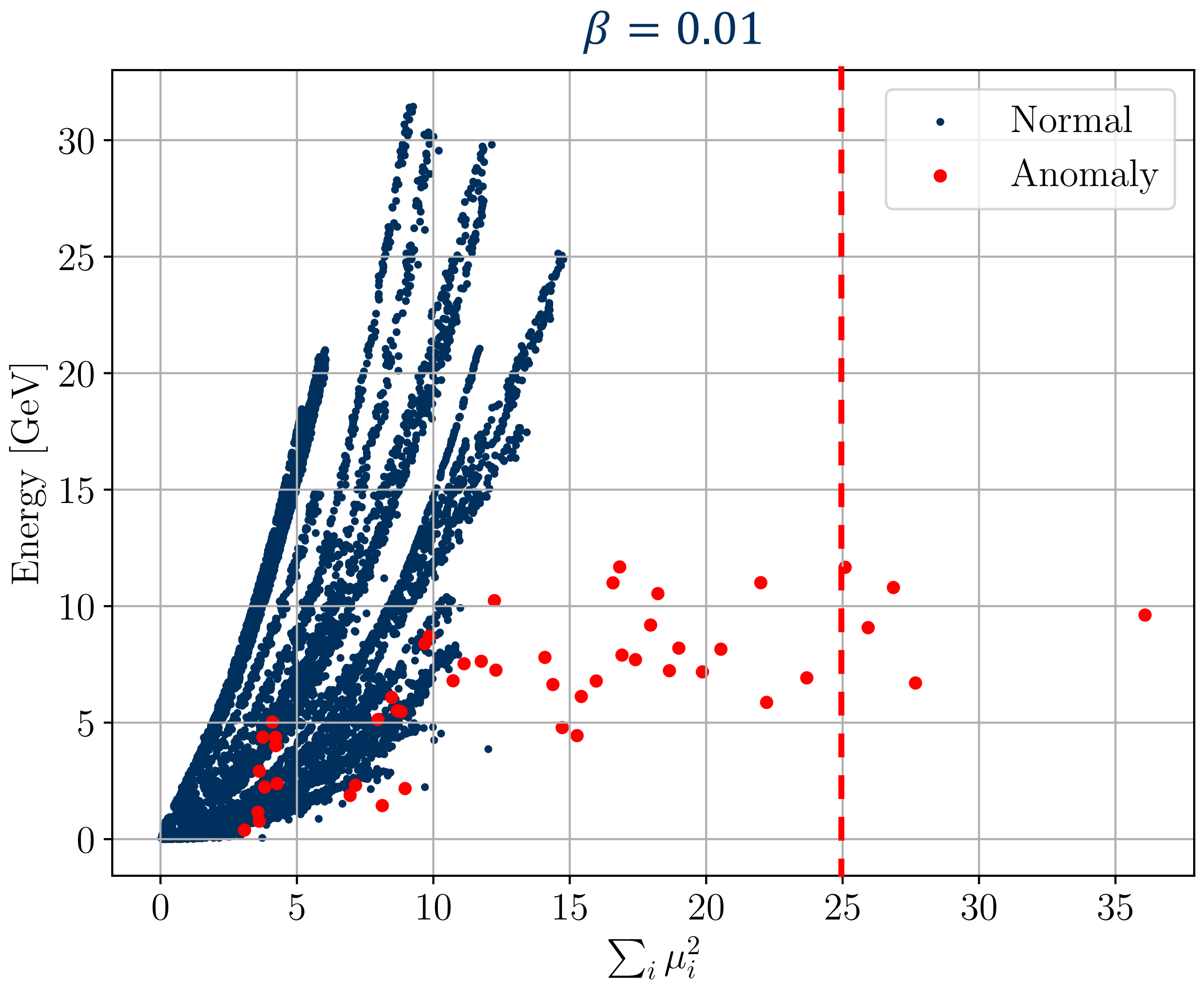


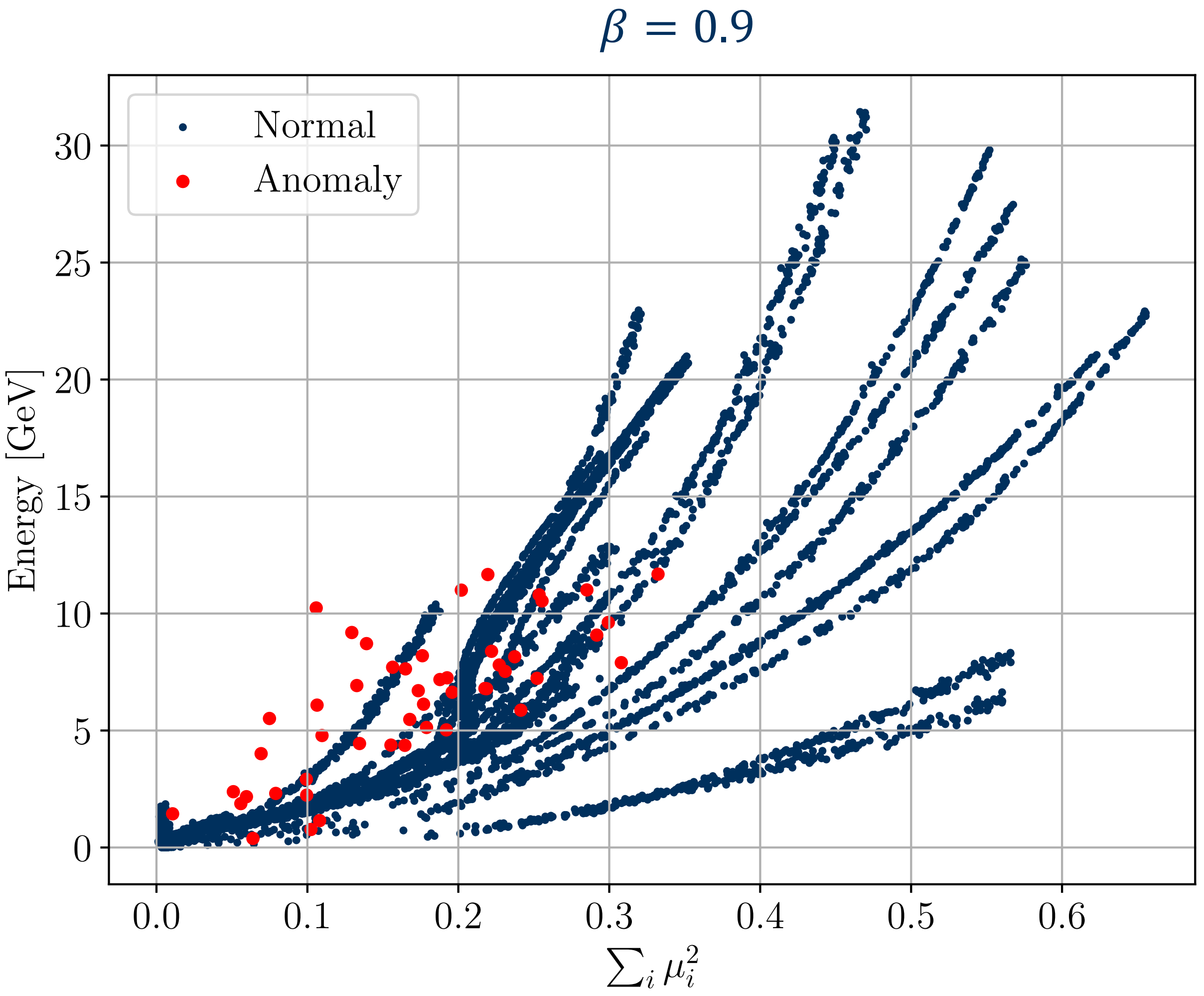
Reconstruction performance on Only Pileup samples (full VAE)



Correlation between new anomaly score and energy

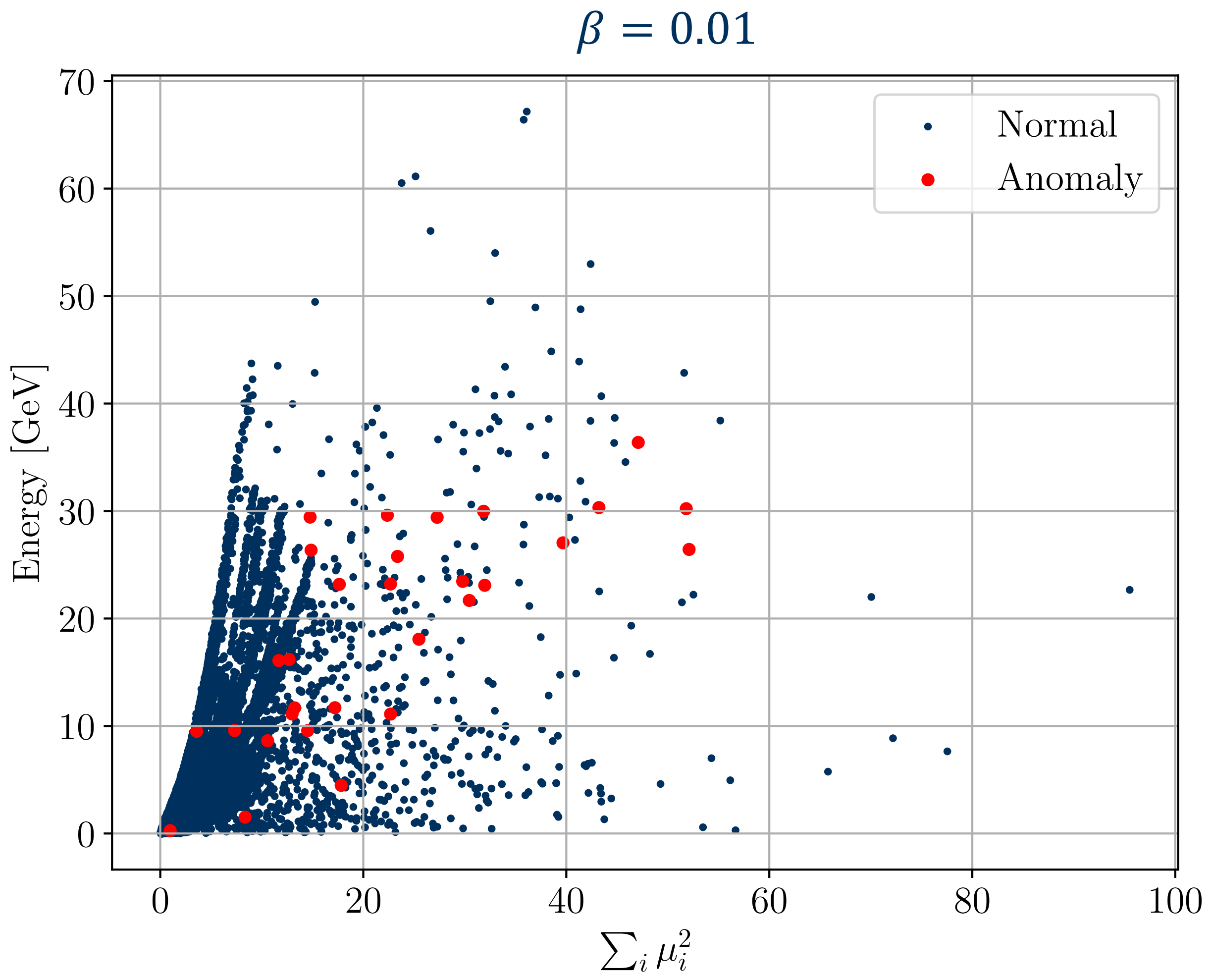
Test data: High energy signal peaks with a constant gap

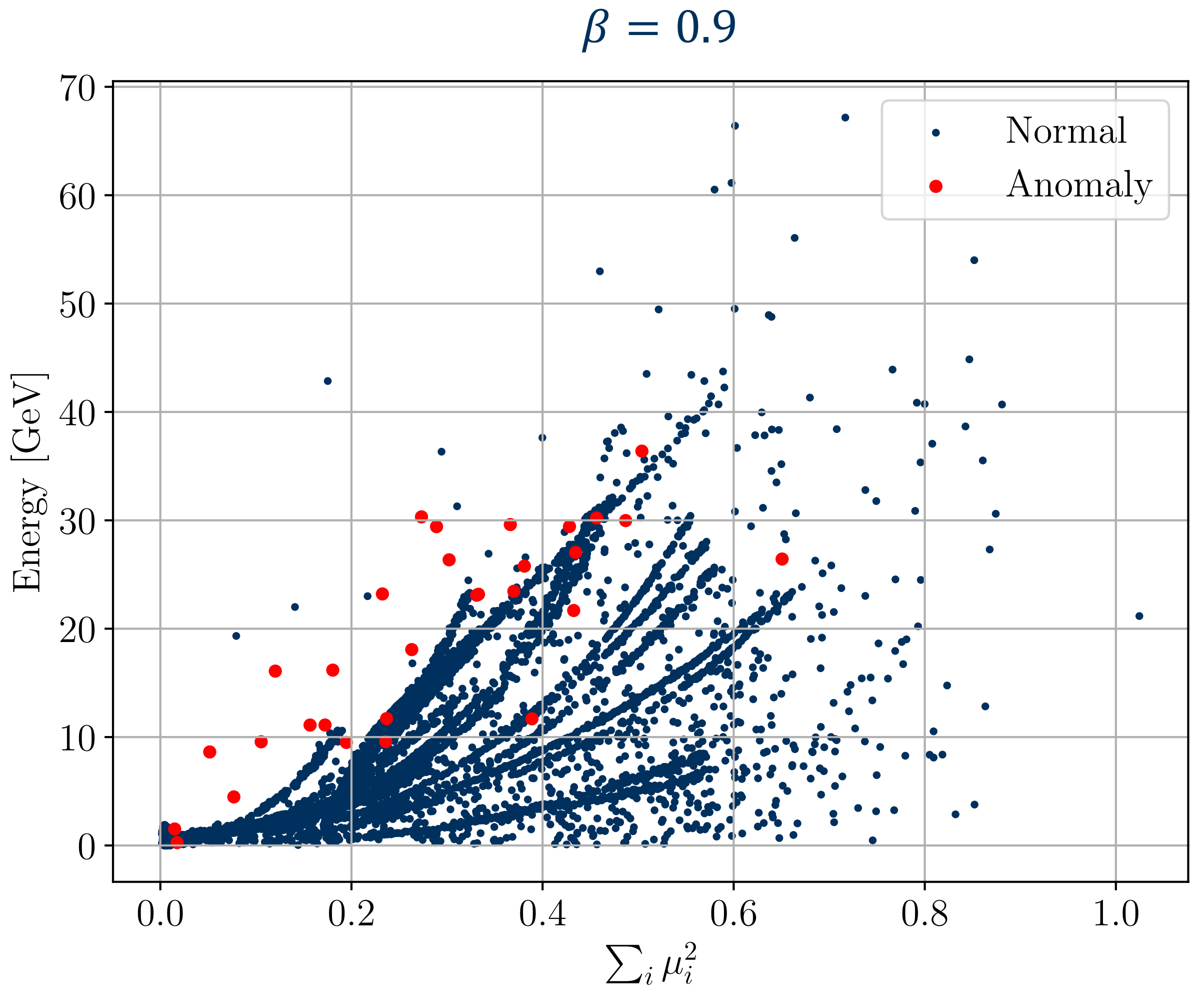




Correlation between new anomaly score and energy

Test data: High energy signal peaks with a uniform gap





Multiple cell Anomaly Detection: Results uniform gap samples