DEEP CONTINUAL LEARNING FOR ANOMALY DETECTION IN PARTICLE PHYSICS EXPERIMENTS

Eirik Drage Steen
eirik.drage.steen@cern.ch
Supervised by Luis Granado Cardoso, Julián García Pardiñas, Arsenii Gavrikov

ABSTRACT

Data quality monitoring (DQM) is the process of identifying usable data in particle physics experiments. Automation of the DQM process can lead to substantially improved data quality with considerably reduced human effort. This work presents a deep continual learning approach for automatic DQM. We introduce the **C**ontinual **S**hifting **T**ransformer (CST), a Transformer-based model designed for anomaly detection in shifting data distributions. The continual learning method is interpretable, adaptable, and significantly surpasses competing algorithms on key metrics. The code is available at: https://github.com/eiriksteen/deep-DQM.

1 INTRODUCTION

Quality data is a prerequisite for physics analysis. Data Quality Monitoring (DQM) ensures that data produced by particle colliders is of sufficient quality to be analyzed scientifically. Manual DQM is done by *shifters*. DQM is subdivided into the *online* and *offline* phase. The online phase consists of real time monitoring to detect data anomalies as quickly as possible. Every 10 minutes, the online shifter inspects the collected data to determine whether anomalies are present. If they are, the run is restarted, and only the data collected before the anomaly occurred is stored. The batches of data aggregated every 10 minutes are eventually stored, and the offline phase begins. In this phase the shifters evaluate the quality of the combined data to make a final decision on what to discard or use for physics analysis.

DQM has traditionally been conducted manually, and the application of machine learning methods can lead to reduced human effort, improved data quality, and superior scientific discoveries. LHCb researchers have recently made efforts to deliver proof-of-concept solutions to automate DQM. Two main approaches have been considered. Parra et al. (2024) introduced a method based on Reinforcement Learning from Human Feedback (RLHF), which yielded high performance on a synthetic dataset. Additionally, there is an LHCb internal prototype of a statistical model based on computing chi-square statistics between a reference histogram and the current histogram.

This work presents a novel continual deep learning approach to automatic DQM. Continual learning is based on training deep models on continuously arriving streams of data. The key is to remember what has been learned (avoiding the catastrophic forgetting problem), while adapting to new conditions. This is the first time that continual deep learning has been tested for DQM on real data from the Large Hadron Collider. The proposed method outperforms the baselines, and is ready for implementation into CERN's monitoring system.

2 PROBLEM DESCRIPTION

The physics variables are measured periodically in histograms of d bins, and each histogram represents the distribution of a physics variable within a fixed time interval. We have n runs, p physics variables, and d bins aggregated in a data matrix of shape $n \times p \times d$. The complete data matrix is not available until all the runs have finished. Thus we model the problem incrementally, where data from $b \ll n$ runs arrive sequentially and chronologically. Each run has a corresponding binary label. The label 0 constitutes a normal run, which means a run of acceptable data quality. The label 1 means the run is anomalous, and should be discarded. The main characteristics of the data stream are:

Non-stationary histograms At any time, the parameters of the run may change. This leads to a distributional shift in the physics variables. What constitutes anomalies and normal conditions change continuously. However, we assume that runs labeled as anomalies in the past can not reappear in the data stream as normal runs.

Large feature space The number p of physics variables is large, and the distributions for the variables differ significantly.

Human feedback Human shifters manually review and label the individual histograms in real time. A method that humans can interact with is desirable.

Imbalanced data The quantity of runs labeled as normal is substantially higher than the number of anomalous runs.

Clustered labels The labels are clustered in sequences. Specifically, $P(y_t = y|y_{t-1} = y) \gg P(y_t = y|y_{t-1} = 1 - y)$, where $y \in \{0, 1\}$ (with the implementation of an automatic DQM method, this may cease to be the case. However, for simplicity, we disregard this in our analysis).

Uninformative labels When a run is labeled as anomalous, this only means that there are anomalies in a subset of the histograms. Consequently, the labels do not tell which histograms are anomalous.

3 METHODS

3.1 CONTINUAL SHIFTING TRANSFORMER

The approach consists of a Transformer architecture specialized to handle shifting distributions, leveraged in conjunction with a replay buffer. The training algorithm is summarized in algorithm 2. We introduce the **C**ontinual **S**hifting **T**ransformer (**CST**), a Transformer model based on Vaswani et al. (2017), that is adapted to continually streaming data. The model is defined by the following components:

Embedding We use a two-layer perceptron to transform the histograms to the embedding space. To account for differing distributions between physics variables, we leverage a learnable absolute positional encoding scheme. A position embedding matrix of shape $p \times d$ (p physics variables and d bins per histogram) is summed with the raw input. The reason for not summing to the embedded input is that we want the embeddings to take context into account. As opposed to language, where the base form of a word can be represented regardless of position, or vision, where there is no inductive bias that certain patterns only occur in a fixed pixel space, the distributions in the physics histograms represent completely different quantities.

Distribution shift detection To model distributional shifts, a tensor $\mathbf{P}$ of shape $k \times p \times d$ with the past k normal runs is included as input to the model (k is treated as a hyperparameter). Both the current run and the $\mathbf{P}$ tensor are transformed to the embedding space. Thereafter, the mean of the embeddings corresponding to the past k runs is computed. The absolute difference between the current run embedding and the average of the past embeddings is passed into a *change detection* network. This network is a two-layer perceptron. The output of the network is summed with the input embedding to produce a tensor for further processing.

Transformer The sum of the input embeddings and change logits are passed to a standard Transformer, which consists of one multi-head attention block and one MLP block. Finally, the Transformer output is processed by a linear head to produce the output logits. The Transformer computes attention over the embeddings of the different physics variables. The model is expected to maximally attend to the histogram that is anomalous. Consequently, the attention weights can be used to predict the most likely anomaly source. The anomaly source predictions are computed by summing the head and row dimensions of the attention weight tensor.

Replay buffer To deal with the imbalanced data, catastrophic forgetting, as well as the issue of clustered labels, we leverage a replay buffer. The replay buffer augments the current batch with past runs resampled from the buffer. We save all anomalous runs observed so far. If there are n_1 positively labeled runs in the current batch, and n_0 negatively labeled, n_0 positive and n_1 negative runs of the past are drawn from the replay buffer. This way, the replay buffer functions as a *batch balancer*, that ensures the model is always trained on a batch with equally many positive and negative

runs. The past runs are drawn according to a multinomial distribution where the weight of run t is:

$$w_t = \frac{t}{\sum_{t'=1}^{T} t'} \tag{1}$$

We do it this way since we want to weight more recent runs higher due to changing conditions, and to avoid the bias towards older runs that would occur with uniform sampling over a growing dataset. Additionally, there is a hyperparameter n_{steps} for the number of replay steps. The core of the training algorithm consists of creating n_{steps} augmented batches, and taking a gradient descent step once for each augmented batch. For each step, the newest data stays in the batch, while the rest is redrawn.

The CST forward pass and training algorithms are detailed in the figures 1 and 2. Note that we allow the deep models some warmup runs before evaluation.

Algorithm 1 CST Forward Pass

Require: $\boldsymbol{X} \in \mathbb{R}^{b\times n\times d}$, $\mathbf{P} \in \mathbb{R}^{b\times k\times n\times d}$ {The current and past k histograms}
$\boldsymbol{E}_x \leftarrow \text{embed}(\boldsymbol{x} + \boldsymbol{E}_{\text{pos}}) \in \mathbb{R}^{b\times n\times h}$ {Add positional embedding and embed with MLP}
$\mathbf{E}_p \leftarrow \text{embed}(\boldsymbol{P} + \boldsymbol{E}_{\text{pos}}) \in \mathbb{R}^{b\times k\times n\times h}$
$\bar{\boldsymbol{E}}_p \leftarrow \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{E}_{p_i} \in \mathbb{R}^{b\times n\times h}$
$\boldsymbol{C} \leftarrow f_{\text{change}}(\|\bar{\boldsymbol{E}}_p - \boldsymbol{E}_x\|) \in \mathbb{R}^{b\times n\times h}$ {Compute the change matrix}
$\tilde{\boldsymbol{E}}_x \leftarrow \boldsymbol{E}_x + \boldsymbol{C}$
$\boldsymbol{T}, \mathbf{A} \leftarrow \text{transformer}(\tilde{\boldsymbol{E}}_x)$
return $W_{\text{head}}(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{T}_i) \in \mathbb{R}^b, (\sum_{i=1}^{n}\sum_{h=1}^{n_h} \mathbf{A}_{ih}) \in \mathbb{R}^{b\times n}$

Algorithm 2 Continual Training of ACT

Require: Classifier f_θ, Dataset $\mathcal{D}$, Hyperparameters
  Initialize f_θ, replay buffer $\mathcal{R}$, optimizer, loss function $\mathcal{L}_{\text{BCE}}$
  $\mathcal{S} \leftarrow \{\}$ {Anomaly scores set}
  for each batch number $b \leq n$, batch $B \in \mathcal{D}$ **do**
    $\mathbf{P}_i \leftarrow \text{get\_past\_k\_runs}(i)$ for $i \in [b, b + |B|)$
    $\hat{y}_i \leftarrow f_\theta(\boldsymbol{X}_i, \mathbf{P}_i), \forall i$
    if warmup period is over **then**
      $s_i \leftarrow \sigma(\hat{y}_i), \forall i$ {Sigmoid activation}
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_i\}_{i=1}^{b}$
      if $\exists i : (s_i > 0.5) \neq y_i$ **then**
        $n_{\text{steps}} \leftarrow 2 \cdot n_{\text{default}}$
      else
        $n_{\text{steps}} \leftarrow n_{\text{default}}$
      end if
    end if
    for $k = 1$ to n_{steps} **do**
      $\tilde{\boldsymbol{X}}, \tilde{y}, \tilde{\mathbf{P}} \sim \mathcal{R}(B)$ {Sample from replay buffer}
      $\hat{y} \leftarrow f_\theta(\tilde{\boldsymbol{X}}, \tilde{\mathbf{P}})$
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{BCE}}(\hat{y}, \tilde{y})$
      Update θ using $\nabla_\theta \mathcal{L}$
    end for
    $\mathcal{R} \leftarrow \mathcal{R} \cup B$ {Update replay buffer}
  end for
  return $f_\theta, \mathcal{P}$

3.2 Reconstruction Approach

The reconstruction approach is to train a reconstruction model only on the histograms from normal runs, and then use the mean-squared-error between the original and reconstructed histograms as the anomaly score. The method is based on Japkowicz et al. (1995). The assumption is that reconstruction errors should be higher for anomalous histograms whose distribution the model does not know.

Continual Reconstruction

For details of the algorithm, see algorithm 3. We process b runs at once, and compute the anomaly score using the reconstruction error (without gradient computation). Each output reconstruction has the shape $p \times d$ (p physics variables and d bins). The mean reconstruction error is calculated per histogram, and the final output is the maximum error. The maximum is used since an anomaly in at least one histogram is enough to label the whole run as anomalous.

After computing the anomaly score for the current run, if the run contains any normal runs, the model trains. For n_{steps} train steps, the current batch is augmented using the same replay buffer as with the CST, and the model is updated to optimize the objective 5.

Variational Autoencoder

As the reconstruction model, we use a variational autoencoder similar to the standard design proposed by Kingma & Welling (2013). The model consists of two encoders and one decoder. The encoders parameterize the means and the log-variances of the conditional normal distribution:

$$q_\phi(\boldsymbol{Z}_t|\boldsymbol{X}_t) = \mathcal{N}(\boldsymbol{Z}_t; \mu(\boldsymbol{X}_t), \sigma^2(\boldsymbol{X}_t)) \tag{2}$$

From this distribution, $\boldsymbol{Z}_t$ is sampled whilst preserving the computational graph using the reparameterization trick:

$$\boldsymbol{Z}_t = \mu(\boldsymbol{X}_t) + \sigma(\boldsymbol{X}_t) \odot \boldsymbol{N}, \quad \boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{3}$$

Thereafter, the reconstruction is computed by the decoder:

$$\hat{\boldsymbol{X}}_t = p_\theta(\boldsymbol{X}_t|\boldsymbol{Z}_t) \tag{4}$$

Finally, the loss function is composed of two terms:

$$\mathcal{L}(\theta, \phi; \boldsymbol{X}_t) = \mathbb{E}_{q_\phi(\boldsymbol{Z}_t|\boldsymbol{X}_t)}[\log p_\theta(\boldsymbol{X}_t|\boldsymbol{Z}_t)] - D_{KL}(q_\phi(\boldsymbol{Z}_t|\boldsymbol{X}_t)||p(\boldsymbol{Z}_t)) \tag{5}$$

Where we have one term for the reconstruction error, and one for latent space regularization.

Algorithm 3 VAE Continual Anomaly Detection

Require: VAE encoder q_ϕ, decoder p_θ, Dataset $\mathcal{D}$, Hyperparameters
  Initialize q_ϕ, p_θ, replay buffer $\mathcal{R}$, optimizer
  $\mathcal{S} \leftarrow \{\}$ {Score set}
  for each batch $B = \{\boldsymbol{X}_i\}_{i=1}^b \in \mathcal{D}$ **do**
    $\mu, \log \sigma^2 \leftarrow q_\phi(\boldsymbol{X}_i), \forall i$
    $\boldsymbol{Z}_i \sim \mathcal{N}(\mu, \sigma^2)$
    $\hat{\boldsymbol{X}}_i \leftarrow p_\theta(\boldsymbol{Z}_i), \forall i$
    if $t > t_{\text{warmup}}$ **then**
      $e_i \leftarrow \|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\|_2, \forall i$ {Reconstruction error (L2 norm)}
      $s_i \leftarrow \max(e_i), \forall i$ {Anomaly scores}
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_i\}_{i=1}^b$
    end if
    $B_{\text{normal}} \leftarrow \{\boldsymbol{X}_i \in B : \text{is\_normal}(\boldsymbol{X}_i)\}$ {Non-anomalous runs}
    if $|B_{\text{normal}}| > 0$ **then**
      for $k = 1$ to n_{steps} **do**
        $\tilde{B} \sim \mathcal{R}(B_{\text{normal}})$ {Resample from replay buffer}
        $\mu, \log \sigma^2 \leftarrow q_\phi(\boldsymbol{X}_i), \forall \boldsymbol{X}_i \in \tilde{B}$
        $\boldsymbol{Z}_i \sim \mathcal{N}(\mu, \sigma^2)$
        $\hat{\boldsymbol{X}}_i \leftarrow p_\theta(\boldsymbol{Z}_i), \forall \boldsymbol{X}_i \in \tilde{B}$
        $\mathcal{L}_{\text{VAE}} \leftarrow \frac{1}{|\tilde{B}|} \sum_{\boldsymbol{X}_i \in \tilde{B}} [\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\|_2^2 + \text{KL}(q_\phi(\boldsymbol{Z}_i|\boldsymbol{X}_i)\|p(\boldsymbol{Z}))]$
        Update ϕ, θ using $\nabla_{\phi,\theta} \mathcal{L}_{\text{VAE}}$
      end for
    end if
    $\mathcal{R} \leftarrow \mathcal{R} \cup B_{\text{normal}}$ {Update replay buffer}
  end for
  return $q_\phi, p_\theta, \mathcal{S}$

3.3 CHI-SQUARE MODEL

The Chi2 model updates a reference histogram over time, purely computed from past encountered non-anomalous runs. The reference is computed through an exponential weighted moving average (EWMA) procedure. Let $\boldsymbol{X}_t$ denote the non-anomalous run t, and α the hyperparameter that controls how quickly the reference is updated. The run is weighted by the inverse of its poisson uncertainty:

$$\omega_t = \frac{1}{\sigma^2_{\boldsymbol{X}_t}} \tag{6}$$

Using this, we compute the EWMA weights, sum, mean, and uncertainty:

$$\boldsymbol{W}_{t+1} = \alpha \boldsymbol{W}_t + (1-\alpha)\omega_t \tag{7}$$

$$\boldsymbol{S}_{t+1} = \alpha \boldsymbol{S}_t + (1-\alpha)\omega_t \boldsymbol{X}_t \tag{8}$$

$$\mu_{t+1} = \frac{\boldsymbol{S}_{t+1}}{\boldsymbol{W}_{t+1}} \tag{9}$$

$$\boldsymbol{S}_{\sigma,t+1} = \alpha \boldsymbol{S}_{\sigma,t} + (1-\alpha)\omega_t (\boldsymbol{X}_t - \mu_t)^2 \tag{10}$$

Once we have the mean and uncertainty, we conduct a Chi2 test between the next run and the current histogram to arrive at an anomaly score. To deal with the run-length bias (as described in subsection 4.2), the reference is resampled to match the statistics of the incoming run. Finally, the Log-Chi2 statistic is saved as the anomaly score.

4 DATA

4.1 SYNTHETIC

The synthetic data is designed to closely mirror the real world particle collision data. The characteristics of clustered and imbalanced labels, differing distributions per physics variable, and changing conditions are all taken into account. The statistics are summarized in table 1.

We start by setting the number of runs n, physics variables p, and bins d. We define sets of anomalous and normal parameters. Each physics variable is randomly assigned a set of normal and anomalous parameters, and the parameters change every 100 steps. To generate the histograms, samples are drawn for each histogram from a normal distribution with the assigned parameters. Additionally, each physics variable is assigned a nonlinear transformation, and this transformation is applied to the normal sample to generate the measurements. The possible transformations are:

- $t_1(x) = x$
- $t_2(x) = \sin(x)\log^2(x)$
- $t_3(x) = \log(x)e^x$
- $t_4(x) = \sqrt{x}\log(x) + \log(x)$
- $t_5(x) = |x|\log(x)$

The transformations reflect the fact that different physics variables have differently shaped histograms, and were arbitrarily chosen. Finally, the histograms are computed from the empirical distribution of the samples. At each timestep, the label of the run is determined by:

$$y_i \sim \begin{cases} \text{Bernoulli}(0.8), & \text{if } y_{i-1} = 1 \\ \text{Bernoulli}(0.025), & \text{if } y_{i-1} = 0 \end{cases}$$

If the previous run was anomalous, there is a high probability that the current one also is, and vice versa.

4.2 LHCB 2018

The LHCb 2018 dataset contains particle collision measurements for 1183 runs and 109 physics variables from Run 2 of the Large Hadron Collider beauty experiment. The variables represent quantities such as mass and momentum of particles. Each histogram is rebinned to 100 bins. Label imbalance is a significant challenge, since roughly 10% of the runs are anomalous.

As with the synthetic data, the labels are clustered and conditions change frequently. However, there are two notable distinctions. Firstly, the anomalies are considerably more subtle; they usually consist of small changes to the input distribution rather than different underlying parameters, as in the synthetic case. Secondly, the labels are noisy. There are confirmed occurrences of both false positives and false negatives in the labels.

Additionally, there is a bias induced by differing runlengths between anomalous and normal runs. Since the runs labeled as anomalous are usually cancelled before completion, the histograms will consist of fewer observations, and the total sum of the bin counts will be lower. Some of the effect can be removed through normalization. Regardless, the histogram shapes will remain affected as the runs with more observations will have smoother histograms closer to the true underlying distributions.

Table 1: Dataset Statistics

Statistic	**LHCb 2018 Data**	**Synthetic Data**
Number of variables	109	100
Number of runs	1183	2000
Number of positive labels	113	210
Number of negative labels	1070	1790

5 EXPERIMENTS

5.1 SETUP

We follow standard continual learning practice and evaluate the models on the immediate next run in the data stream. Thus, all metrics are computed on the basis of data the model has not been trained on. We train each model for 5 runs, and report the mean and standard deviation of the resulting metrics.

We compare the CST model to the Chi2 method and the VAE reconstruction algorithm. We note that the deep learning models need a warmup period before producing meaningful predictions. For the deep models, the metrics are computed on the predictions after the burn-in ends, which is around run 100.

Before processing by the deep models, each histogram is mean-centered and scaled to unit variance. The hyperparameters of each model are shown in table 2. For the deep models, they were manually chosen, while Optuna was used to find the optimal Chi2 α's.

Table 2: Hyperparameters

CST Hyperparameters	
n_{steps}	2
k_{past}	5
Learning Rate	1e-4
Batch Size	1
VAE Hyperparameters	
n_{steps}	4
Learning Rate	1e-4
Batch Size	2
Chi2 Hyperparameters	
$\alpha_{\text{synthetic}}$	1e-4
α_{lhcb}	0.529

5.2 METRICS

We report the balanced accuracy, average precision, F1-score, roc-auc, and the accuracy for all models. The former 3 metrics are known for being well suited to imbalanced data, and we also include the latter 2 for benchmarking overall performance.

Furthermore, we note that the clustered labels make the standard metrics unable to verify whether the models truly learn the patterns in the data. Due to the clustered labels, a model that simply outputs the label of the previous run will seemingly perform well. The problem with the standard metrics is that they do not capture the ability to detect an anomalous run *before it has happened*. To address this limitation, we introduce *flip metrics*. These metrics are computed exclusively on cases where the current run's label differs from the previous run's label. For a sequence of labels $y_1, y_2, ..., y_T$, flip metrics are calculated on the subset $\{(x_t, y_t)|y_t \neq y_{t-1}, t > 1\}$. The flip cases occur relatively infrequently, so the metrics will be biased, and we therefore report flip metrics in addition to the standard ones.

We also want to evaluate the model's capability to determine the physics variables that originated the anomalies. We frame this as a multilabel classification problem. Specifically, for all the runs with positively predicted labels, the sums over the heads and rows of the attention weight tensors are stored as the anomaly source predictions. We compute the accuracy (exact match ratio), F1, roc-auc, and the average precision for the multilabel predictions. In addition, we quantify the models ability to capture at least one anomaly by computing the metrics in a single-label fashion exclusively on the predictions corresponding to the highest predicted score. We compute the metrics solely on the runs that the model labels as anomalous. An imagined use-case would be that the model predicts the anomaly, and subsequently outputs the histogram with the highest probability of being anomalous. A human would then inspect the histogram, rather than all the histograms, and be able to conclude whether the run is anomalous.

5.3 RESULTS

The anomaly detection metrics are summarized in tables 3, 4, 5, 6. The multilabel metrics are in table 7.

Anomaly Detection

Table 3: Overall Performance Comparison for the Synthetic Dataset

Metrics	CHI2	CST	VAE
Accuracy	0.989 ± 0.000	$\mathbf{0.992 \pm 0.004}$	0.960 ± 0.018
Balanced Acc	0.985 ± 0.000	$\mathbf{0.988 \pm 0.002}$	0.954 ± 0.009
AUPRC	0.792 ± 0.000	$\mathbf{0.984 \pm 0.003}$	0.944 ± 0.005
F1-score	0.947 ± 0.000	$\mathbf{0.962 \pm 0.019}$	0.829 ± 0.063
ROC-AUC	0.982 ± 0.000	$\mathbf{0.990 \pm 0.004}$	0.975 ± 0.003

Table 4: Overall Performance Comparison for the LHCb 2018 Dataset

Metrics	CHI2	CST	VAE
Accuracy	0.754 ± 0.000	0.872 ± 0.036	$\mathbf{0.912 \pm 0.015}$
Balanced Acc	0.810 ± 0.000	$\mathbf{0.868 \pm 0.015}$	0.757 ± 0.007
AUPRC	0.462 ± 0.000	$\mathbf{0.532 \pm 0.044}$	0.350 ± 0.008
F1-score	0.307 ± 0.000	$\mathbf{0.470 \pm 0.069}$	0.458 ± 0.038
ROC-AUC	0.881 ± 0.000	$\mathbf{0.921 \pm 0.013}$	0.763 ± 0.006

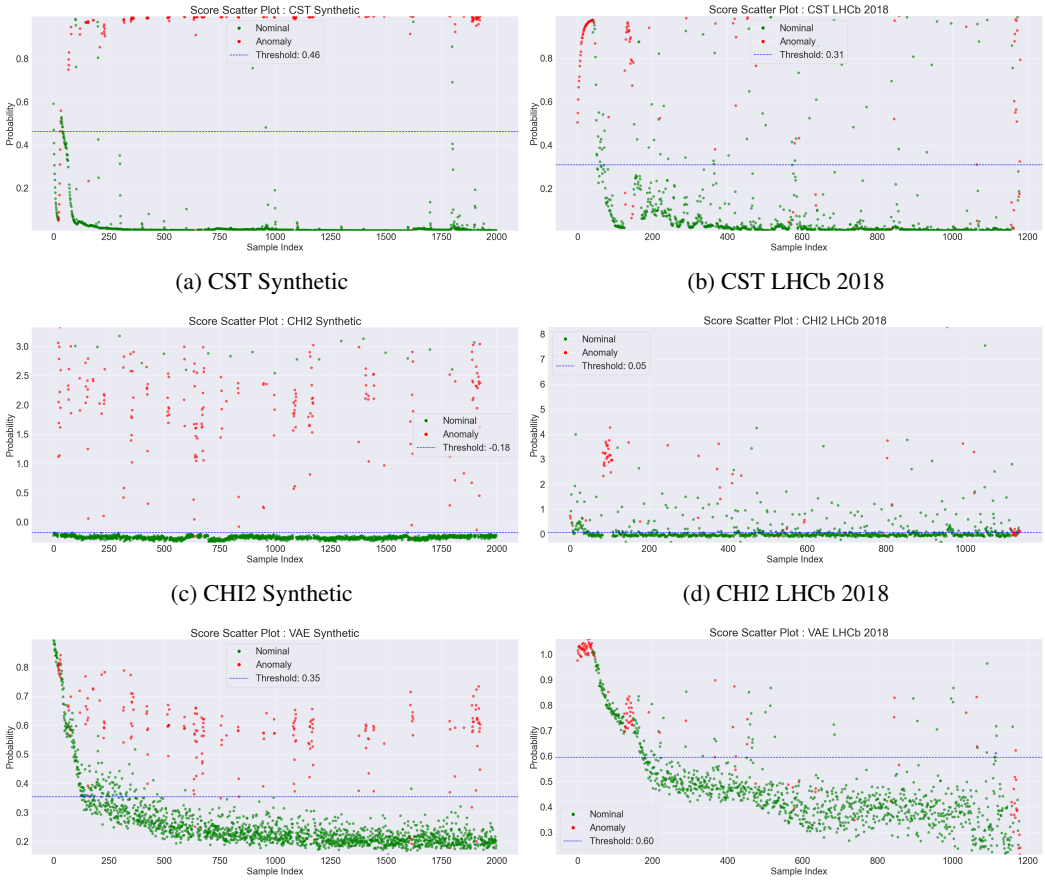
Table 5: Flip Metrics Performance Comparison for the Synthetic Dataset

Metrics	CHI2	CST	VAE
Accuracy	0.977 ± 0.000	$\mathbf{0.983 \pm 0.010}$	0.908 ± 0.021
Balanced Acc	0.977 ± 0.000	$\mathbf{0.983 \pm 0.010}$	0.908 ± 0.021
AUPRC	0.955 ± 0.000	$\mathbf{0.992 \pm 0.007}$	0.965 ± 0.003
F1-score	0.977 ± 0.000	$\mathbf{0.983 \pm 0.010}$	0.906 ± 0.021
ROC-AUC	0.965 ± 0.000	$\mathbf{0.988 \pm 0.013}$	0.948 ± 0.007

Table 6: Flip Metrics Performance Comparison for the LHCb 2018 Dataset

Metrics	CHI2	CST	VAE
Accuracy	0.685 ± 0.000	$\mathbf{0.718 \pm 0.065}$	0.657 ± 0.015
Balanced Acc	0.685 ± 0.000	$\mathbf{0.717 \pm 0.066}$	0.662 ± 0.015
AUPRC	$\mathbf{0.801 \pm 0.000}$	0.784 ± 0.070	0.741 ± 0.030
F1-score	0.739 ± 0.000	$\mathbf{0.744 \pm 0.048}$	0.567 ± 0.021
ROC-AUC	$\mathbf{0.760 \pm 0.000}$	0.754 ± 0.052	0.645 ± 0.017

We observe that the CST compares favorably to the other models. It outcompetes them on almost all metrics (although the difference is within one standard deviation for most metrics on the synthetic data), and performs similarly to the Chi2 on the flips. For the standard metrics, the CST average precision improvement is particularly notable. The VAE is the worst performer. It achieves solid results on the synthetic data, but is substantially inferior at detecting real anomalies. This may be due to the subtle nature of the anomalies, which a simple reconstruction error does not capture. However, it performs respectably on the synthetic dataset. This indicates that it may benefit from more data, and that the performance gap might close with more runs. We note that the Chi2 performs well overall, and is a strong model considering its relative simplicity.



(a) CST Synthetic

(b) CST LHCb 2018

(c) CHI2 Synthetic

(d) CHI2 LHCb 2018

(e) VAE Synthetic

(f) VAE LHCb 2018

Figure 1: Scatterplots of predicted anomaly scores over time

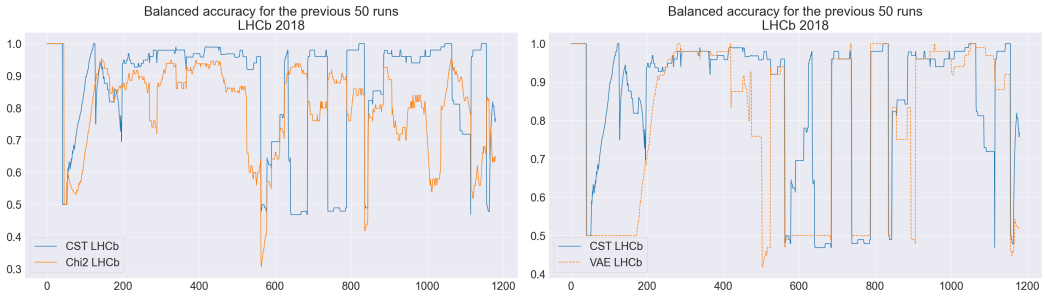
The plots in figure 1 show the model predictions over runs through all the data. The red points are true anomalies, the green points are true normal runs, the horizontal blue dotted line shows the decision boundary, and the vertical orange line shows the time of the burn-in end. The x-axis represents the current time, and the horizontal axis gives the score predicted by the model. The plots show that its easier to detect anomalies in the synthetic data, but CST and Chi2 separate the classes well for both datasets. We see from the burn-in that the deep models require some time to learn the basics of the task. The VAE separates the data well in the synthetic case, but not for the LHCb data.

Adaptability

We now compare the models on how well they adapt to shifting distributions. To capture this, we compare plots of the cumulative balanced accuracies. For the plots in 2 and 3, the value at time t is the cumulative balanced accuracy computed on the previous $t - 50$ to t predictions.

There is a high degree of correlation between the plots. It appears that the models often agree upon which runs are difficult. At the same time, it varies how extreme the negative effect on the performance is. For example, we can see that the CST struggles substantially with an LHCb run around time 650. The Chi2 model is unaffected by it. On the contrary, we see that the CST handles the LHCb runs from time 900 to 1100 better than the Chi2. For these cases, it would be a good idea to combine the models to gain from their respective strengths. The CST's performance generally follows a smoother trajectory, with fewer small ups and downs than the Chi2.

To gauge the speed of recovery upon facing shifting distributions, we can observe how long the p50 balanced accuracy stays low until recovery. As expected, the VAE has the largest downtime, while the CST and Chi2 usually recover quickly. The widest gap apart from the VAE appears in the Chi2 plot for the LHCb data, with a duration of around 100 runs between 500 to 600.



(a) Balanced accuracies CST vs Chi2 on LHCb 2018 (b) Balanced accuracies CST vs VAE on LHCb 2018

Figure 2: Comparison of balanced accuracies on LHCb 2018

Figure 3: Balanced accuracies on the synthetic data

Multilabel Prediction

Table 7 shows how well the attention weights detect the anomaly source on the synthetic dataset. We see that the exact match ratio (overall accuracy) is low, while the accuracy of the highest anomaly score is high.

Table 7: CST Multilabel Metrics for Anomaly Source Prediction on the Synthetic Data

Metrics	Overall	Maximum
Accuracy	0.018 ± 0.006	0.954 ± 0.013
F1-score	0.104 ± 0.009	0.977 ± 0.007
ROC-AUC	0.657 ± 0.057	Undefined
Average Precision	0.426 ± 0.046	0.954 ± 0.013

This is as expected, due to the unimodal nature of the softmax activation in the attention layer. Thus, while the multilabel outputs can be trusted to detect a histogram that is anomalous with a high probability, it is not adept at detecting all histograms that are anomalous. Regardless, we observe qualitatively that the multilabel predictions are able to capture a significant subset of the anomalies. An example is shown in figure 4; this would count as a correct maximum prediction, but incorrect as a multilabel prediction (since it is not an exact match).

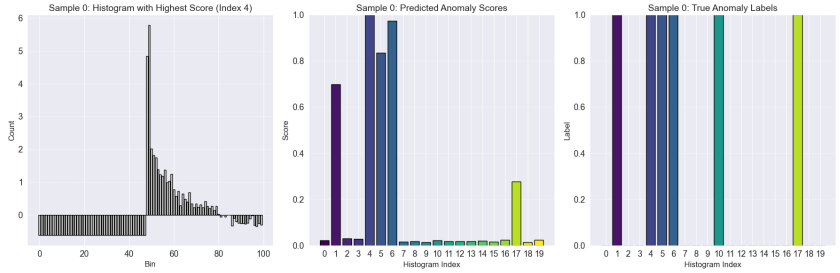


Figure 4: Anomaly source predictions for a synthetic dataset with 20 physics variables

Ablation

We now study the performance effects of the CST components. We train the CST model without the addition of the change matrix, and without the Transformer layer. The results are shown in tables 8, 9, 10, 11:

Table 8: CST Overall Performance Comparison for the Synthetic Dataset

Metrics	Original	Without Change Term	Without Transformer
Accuracy	**0.992 ± 0.004**	0.987 ± 0.005	0.925 ± 0.003
Balanced Acc	**0.988 ± 0.002**	0.969 ± 0.002	0.907 ± 0.005
AUPRC	**0.984 ± 0.003**	0.963 ± 0.002	0.824 ± 0.009
F1-score	**0.962 ± 0.019**	0.936 ± 0.024	0.702 ± 0.008
ROC-AUC	**0.990 ± 0.004**	0.980 ± 0.002	0.948 ± 0.001

Table 9: CST Overall Performance Comparison for the LHCb 2018 Dataset

Metrics	Original	Without Change Term	Without Transformer
Accuracy	0.872 ± 0.036	0.893 ± 0.018	**0.896 ± 0.008**
Balanced Acc	0.868 ± 0.015	0.850 ± 0.013	**0.875 ± 0.003**
AUPRC	**0.532 ± 0.044**	0.372 ± 0.015	0.451 ± 0.012
F1-score	0.470 ± 0.069	0.489 ± 0.033	**0.509 ± 0.018**
ROC-AUC	0.921 ± 0.013	0.883 ± 0.011	**0.938 ± 0.002**

Table 10: CST Flip Metrics Performance Comparison for the Synthetic Dataset

Metrics	Original	Without Change Term	Without Transformer
Accuracy	**0.983 ± 0.010**	0.889 ± 0.034	0.727 ± 0.013
Balanced Acc	**0.983 ± 0.010**	0.889 ± 0.034	0.727 ± 0.013
AUPRC	**0.992 ± 0.007**	0.959 ± 0.006	0.861 ± 0.009
F1-score	**0.983 ± 0.010**	0.889 ± 0.030	0.740 ± 0.014
ROC-AUC	**0.988 ± 0.013**	0.938 ± 0.008	0.807 ± 0.008

Table 11: CST Flip Metrics Performance Comparison for the LHCb 2018 Dataset

Metrics	Original	Without Change Term	Without Transformer
Accuracy	0.718 ± 0.065	0.620 ± 0.028	**0.771 ± 0.008**
Balanced Acc	0.717 ± 0.066	0.621 ± 0.027	**0.772 ± 0.008**
AUPRC	0.784 ± 0.070	0.745 ± 0.010	**0.855 ± 0.003**
F1-score	0.744 ± 0.048	0.606 ± 0.042	**0.774 ± 0.008**
ROC-AUC	0.754 ± 0.052	0.659 ± 0.014	**0.826 ± 0.006**

Interestingly, there is a divide between the performance on the synthetic and real data. For the synthetic data, all components are clearly well used. On the contrary, the network that excludes

the Transformer layer achieves top performance on the LHCb data (excepting the average precision score). This may be caused by a bias in the data, potentially related to the run-length. It could also be that the change signal is all the model needs, and that the Transformer only adds noise. Regardless, the substantially higher average precision indicates that it leads to better separation of the positive class. It makes sense that the flip metrics are improved without the Transformer, as the flips are inherently and exclusively about abrupt change. Further tests on more data are required to determine if the Transformer is truly beneficial in a real data scenario.

However, while the Transformer appears to induce noise in the LHCb 2018 case, removing it likely leads to reduced adeptness at detecting the anomaly sources. To compare the models multilabeling abilities, we use the change logits to compute an equivalent scoring for the network without the Transformer layer. We do this by summing the change outputs for each physics variable, and min-max normalizing each sum. The subsequent evaluation procedure is identical to that of the multilabeling Transformer. The results are displayed in table 12:

Table 12: Comparison of Source Prediction Metrics on the Synthetic Data

Metrics	**Overall**		**Maximum**	
	Original	**Without Transformer**	**Original**	**Without Transformer**
Accuracy	$\mathbf{0.0175 \pm 0.0064}$	0.0000 ± 0.0000	$\mathbf{0.9541 \pm 0.0130}$	0.7841 ± 0.0083
F1-score	0.1042 ± 0.0089	$\mathbf{0.3898 \pm 0.0203}$	$\mathbf{0.9765 \pm 0.0068}$	0.8790 ± 0.0052
ROC-AUC	0.6573 ± 0.0570	$\mathbf{0.8929 \pm 0.0010}$	Undefined	Undefined
Average Precision	0.4260 ± 0.0455	$\mathbf{0.6303 \pm 0.0094}$	$\mathbf{0.9541 \pm 0.0130}$	0.7841 ± 0.0083

We see that that both models are highly inaccurate on exact matches, but that including the Transformer leads to significantly improved performance on identifying at least one of the anomalous histograms. The other metrics for multilabeling are higher without the Transformer, but as they do not correspond to the identification of anomalies, they are less important.

6 Conclusion

This work introduced a continual learning method for anomaly detection in particle physics experiments. We conducted the first ever experiments using continual deep learning for DQM on real data from the LHCb experiment, and introduced a novel architecture to solve the task. Experiments show that the model performs well, both under controlled settings, and with real collision data.

Further work should be done to handle the run-length differences to arrive at an unbiased evaluation of the model performance on true collision data. Additionally, the CST backbone can be combined with other training algorithms, such as the RLHF method. Work can also be done to generalize the model to continual anomaly detection in other kinds of data, such as images, text, and time series. Finally, the next step is to implement the continual learning algorithm into the LHCb monitoring system for further testing. The application of the method may lead to reduced human workload, upgraded data quality, and more scientific discoveries.

References

Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pp. 518–523, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603638.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL `https://arxiv.org/abs/1312.6114`.

Olivia Jullian Parra, Julián García Pardiñas, Lorenzo Del Pianta Pérez, Maximilian Janisch, Suzanne Klaver, Thomas Lehéricy, and Nicola Serra. Human-in-the-loop reinforcement learning for data quality monitoring in particle physics experiments, 2024. URL `https://arxiv.org/abs/2405.15508`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.