Journal of
Instrumentation

PAPER • OPEN ACCESS

Continual learning via ensemble-based depth-wise masked autoencoders for data quality monitoring in high-energy physics

To cite this article: A. Gevorgyan *et al* 2026 *JINST* **21** P10004

View the article online for updates and enhancements.

You may also like

- Fast *b*-tagging at the high-level trigger of the ATLAS experiment in LHC Run 3
G. Aad, B. Abbott, K. Abeling et al.
- Expected tracking performance of the ATLAS Inner Tracker at the High-Luminosity LHC
G. Aad, E. Aakvaag, B. Abbott et al.
- Performance of the ATLAS forward proton Time-of-Flight detector in Run 2
G. Aad, E. Aakvaag, B. Abbott et al.

This content was downloaded from IP address 193.140.1.200 on 07/11/2026 at 19:01

Published by IOP Publishing for Sissa Medialab

Received: *February 24, 2026*
Revised: *July 2, 2026*
Accepted: *August 4, 2026*
Published: *October 2, 2026*

Continual learning via ensemble-based depth-wise masked autoencoders for data quality monitoring in high-energy physics

Dale Julson, Eric Reinhardt, Andrii Krutsylo, Resham Sohal, Guillermo Fidalgo, Sergei Gleyzer, Emanuele Usai and the CMS HCAL collaboration

Full author list at the end of the paper

E-mail: Dale.Julson@CeriumLabs.com

Abstract: Machine learning (ML) techniques have been demonstrated to improve the accuracy and efficiency of anomaly detection (AD) when compared to conventional methods. This has led to the adoption of ML for data quality monitoring (DQM) use cases in order to monitor the operation of certain systems to ensure that they are free of undesirable or potentially deleterious anomalies. For applications in the field of High-Energy physics (HEP), where detectors must operate in long-running, harsh environments, ML models used in DQM that have been trained on static datasets are bound to experience degraded performance due to distributional shifts that naturally occur in the incoming data streams, unless directly mitigated via the inclusion of continual ML techniques. This work introduces DepthViT, a lightweight masked autoencoder architecture that employs unique depth-wise embeddings and depth-wise attention, to perform computationally efficient AD tasks. A continual learning framework is developed in which DepthViT models trained on the most recent data streams are ensembled with older models to create a robust overall system which is more resilient to shifts in incoming data streams. When evaluated on occupancy maps from the Compact Muon Solenoid (CMS) hadron calorimeter across multiple data-taking campaigns, the proposed method maintains precision above 98.8% and a stable ratio of correct anomaly predictions to number of anomalies both under small and large distributional shifts. Beyond HEP, the same ensembling-based continual adaptation strategy can be directly applied to industrial monitoring environments where data also naturally evolve over time. This work therefore presents a path toward adaptive anomaly detection systems capable of sustained operation in dynamic data environments.

Keywords: Detector control systems (detector and experiment monitoring and slow-control systems, architecture, hardware, algorithms, databases); Control and monitor systems online; Software architectures (event data models, frameworks and databases); Analysis and statistical methods

© 2026 CERN. Published by IOP Publishing Ltd on behalf of Sissa Medialab. Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

https://doi.org/10.1088/1748-0221/21/10/P10004

2026 JINST 21 P10004

Contents

1 Introduction **1**

2 Background **2**
- 2.1 HCAL detector 2
- 2.2 Data quality monitoring 2
- 2.3 Continual machine learning 3

3 Dataset description **5**

4 Methodology **6**
- 4.1 DepthViT architecture overview 6
- 4.2 DepthViT based anomaly detection 10
- 4.3 Ensembling technique 11

5 Results and discussion **13**
- 5.1 Baseline model performance 13
- 5.2 Baseline model performance degradation over time 14
- 5.3 Model ensembling & mitigation techniques 15
- 5.4 Model ensembling performance compared to latest model performance 17

6 Conclusion **18**

The CMS HCAL collaboration **21**

2026 JINST 21 P10004

1 Introduction

In recent decades, machine learning (ML) based approaches have rapidly become the preeminent methods used in a wide range of scientific and engineering fields [1]. One such domain that has been revolutionized by the inclusion of ML is anomaly detection (AD). AD is broadly defined as the identification of data points that deviate significantly from the expected distribution of a dataset, and which can often be indicative of errors, failures, or other atypical system level conditions. AD has long been an important effort in many endeavors ranging from manufacturing quality control and scientific discovery, to cybersecurity and even spam filtering in emails. The recent inclusion of ML has allowed for the automation of many AD tasks that previously relied heavily on human operators, resulting in higher throughput and improved accuracy [2]. One such instances is in the field of high energy physics (HEP), where experiments such as those conducted at CERN's Large Hadron Collider (LHC) have traditionally relied on human operators to monitor incoming particle interaction data to ensure the detectors are producing high-quality data that is suitable for future physics analyses [3]. This effort is referred to as data quality monitoring (DQM). The introduction of ML-based DQM methods in HEP has led to significant improvements in accuracy, often surpassing human-level performance [4–8]. A critical challenge that has received comparatively less attention, however, is the susceptibility of ML-based DQM methods to shifts in the underlying data distribution. HEP detectors

operate under extreme conditions, including high radiation environments, cryogenic temperatures, and strong magnetic fields. These conditions can induce gradual or abrupt degradation of detector components, leading to shifts in the incoming data streams, even for data that is non-anomalous [9]. For ML-based DQM systems trained on static datasets, such distributional shifts can cause severe performance degradation, a phenomenon we refer to as *model degradation*. As a result of model degradation, anomaly-free data may be incorrectly classified as anomalous. The need to address such challenges motivates the field of continual machine learning (CML), which seeks to develop architectures that can continuously adapt to new data streams while preserving capabilities learned from previous data streams. In this work, we present a novel ML architecture for use in DQM and AD more generally that achieves competitive performance relative to current state-of-the-art approaches in the HEP domain. Our proposed architecture is significantly more lightweight, requiring approximately $1/100^{th}$ of the total parameters compared to existing architectures. Furthermore, we introduce a straightforward ensemble-based CML strategy that effectively mitigates model degradation, leading to improved performance on evolving data streams compared to single-model baselines.

2 Background

This section provides background information relevant to the hadronic calorimeter (HCAL) sub-detector, existing anomaly detection and DQM schemes, and the need for CML strategies to mitigate the effects of distributional shifts in the incoming data streams.

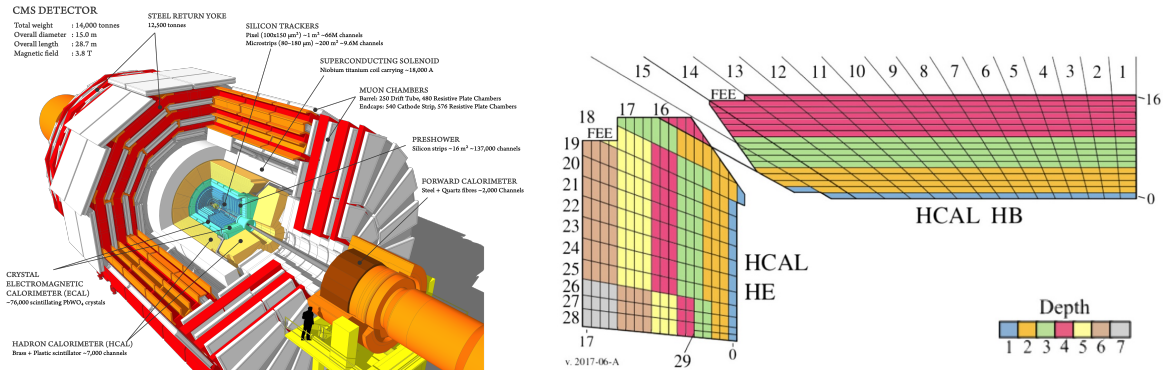
2.1 HCAL detector

The field of HEP seeks to study the fundamental forces and interactions which underlie the observable universe, often through the use of particle collider experiments. The LHC at CERN is currently the world's premier facility for collider-based HEP research, hosting several major experiments [10]. Among these is the Compact Muon Solenoid (CMS) detector (figure 1(a)), a general-purpose experiment designed to record the products of proton-proton and heavy-ion collisions for subsequent physics analyses [11]. The CMS detector is composed of multiple sub-detectors, including the HCAL (shown in yellow), which measures the energy of charged and neutral hadrons produced in collisions [12]. The HCAL consists of alternating layers of brass absorber and plastic scintillator, segmented longitudinally into *depths* (figure 1(b)). The HCAL is further organized into sections such as the HCAL Endcap (HE) and HCAL Barrel (HB), which are physically distinct portions of the HCAL that can be delineated by their depth, azimuthal angle (ϕ) and pseudorapidity (η), which is related to the polar angle θ by $\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$. During the standard operation of the CMS experiment, where the beam conditions are stable, particle interactions are recorded in periods of approximately 23 seconds, known as *lumisections* (LS). The total number of particle interaction events, which comprise an individual LS, varies between LS due to additional event selection criteria, referred to as *triggers*, which help select for unique and interesting particle interaction events [13]. These LS are then bundled together into larger datasets, referred to as *runs*, which are denoted by a RunID value.

2.2 Data quality monitoring

The quality of incoming data collected by HEP experiments must be routinely monitored in order to ensure the produced datasets are free of any abnormalities or detector anomalies which may confound

2026 JINST 21 P10004



(a) (b)

Figure 1. (a) The CMS detector is comprised of four sub-detectors in addition to a 3.8T superconducting solenoid magnet. Each sub-detector specializes in the detection of different particle species originating from collision interactions. (b) The HCAL sub-detector consists of multiple regions, including the HE and HB regions. The depth and $i\eta$ and $i\phi$ (discretely index pseudorapidity and azimuthal angle) values of these regions are shown here [14]. The HCAL is symmetric in ϕ.

future physics analyses [15]. In the case of HCAL, DQM is often performed through the use of DigiOccupancy maps. These maps are three-dimensional histograms that represent the total digitized hits recorded by each individual cell within the HCAL detector during the duration of the LS. The significance of these values is that if an individual cell is not operating properly (i.e. a detector anomaly is present), the DigiOccupancy value will reflect this. Common anomalies that can occur are the presence of *dead channels*, in which the cell records no particle interactions at all, *degraded channels*, where the cell under-records particle interactions, and *hot channels*, in which the cell either over-records particle interactions, or incorrectly records noise as signal. These anomalies can be transient in nature, occurring during a single LS, or persistent in which they occur across multiple LS. Similarly, they can affect individual cells, or entire groupings of cells, depending on the origin of the anomaly. If these anomalies are allowed to persist uninterrupted or otherwise occur undetected, they may either obscure or mimic the very rare signals which are often of interest to particle physicists, or interfere with event triggers which can lead to the collection of uninteresting data.

2.3 Continual machine learning

CML studies the sequential optimization of a model under nonstationary data distributions. Formally, a learner observes a stream of datasets, denoted as $\{\mathcal{D}_t\}_{t=1}^{T}$. For data collected by the CMS detector, t corresponds to one LS and the input distribution, denoted $p_t(x)$, corresponds to collected data where x is a η-ϕ-depth coordinate and the overall distribution may drift with t. The objective is to maintain high performance over time while preventing *catastrophic forgetting*, which refers to a measurable *performance degradation* on earlier data when the model adapts to new distributions [16].

Classical solutions involve training a single model using regularization, experience replay, or architectural isolation to mitigate forgetting. However, these solutions are often ineffective when distribution shifts are large, interfering, or unpredictable. Regularization methods such as Elastic Weight Consolidation (EWC) [17] and Synaptic Intelligence (SI) [18] assume that task optima lie in a locally smooth region of the loss landscape and degrade when gradients from new data are strongly misaligned

2026 JINST 21 P10004

with past importance directions. Replay methods like Gradient Episodic Memory (GEM) [19] and the exemplar-driven incremental classifier iCaRL [20] require stored exemplars or synthetic samples whose representativeness deteriorates under heavy distribution drift. Architectural isolation approaches such as Progressive Networks [21] and PackNet [22] mitigate interference but increase memory or structural cost linearly with the number of tasks. As a result, none of these single-model strategies scale well when shifts are continuous, highly non-stationary, or lack explicit task boundaries.

In contrast, ensemble-based continual learning distributes model capacity across multiple experts. Allowing only a subset to adapt decouples plasticity (rapid adaptation) from stability (knowledge retention), enabling models to specialize in different regions of the data distribution. Below, we review the most relevant ensemble-based CML methods and their trade-offs.

CoMA and its Fisher-weighted variant CoFiMA [23] address forgetting through parameter averaging. After each task, the newly fine-tuned model is averaged with the previously consolidated model. For instance, if the weights of the newly fine-tuned model are denoted ω_t and the weights of the previous model are denoted ω_{t-1}, then the weights of the parameter averaged model denoted by ω_t^* will be given by $\omega_t^* = \lambda\omega_t + (1-\lambda)\omega_{t-1}$, where λ is a tunable hyperparameter. In CoFiMA, the averaging is weighted by the Fisher information so that parameters important for earlier tasks change less. The resulting averaged model combines plasticity from the current task with proximity to previous optima, retaining prior knowledge without runtime ensemble cost. These methods rely on the assumption of connected loss basins, which refers to joined regions of the parameter loss landscape that are connected by paths of low loss values. If successive optima are separated by high-loss regions, averaging can instead degrade performance.

While CoMA and CoFiMA preserve information through weight averaging, Subspace Ensemble (SE) [24] achieves similar stability geometrically. SE constrains model parameters to a low-dimensional affine subspace parameterized by convex combinations (all non-zero and summing to one) of a small set of anchor weights. Sampling a new convex coefficient for each mini-batch yields an implicit ensemble with the runtime cost of a single model. That is to say, if there exists a collection of trained models whose weights are denoted by ω_i, where i is an index that references an individual model, then the SE approach creates a new model that contains weights which are a linear combination of the previous *anchor models*. The new model's weights, $\hat{\omega}$, can be expressed as $\hat{\omega} = \sum_{i=1}^{n} \alpha_i\omega_i$, where $\sum_{i=1}^{n} \alpha_i = 1$. A subsequent variant introduces connectivity regularization to maintain compatibility between subspaces across tasks. Thus, SE relaxes CoMA's connectivity assumption by explicitly learning compatible regions in parameter space.

Building on the subspace idea, Expandable Subspace Ensemble (EASE) [25] targets class-incremental learning on pretrained backbones. EASE freezes the core network and attaches a lightweight adapter for each new task. Each adapter defines a task-specific feature subspace, and inference combines all adapters through subspace reweighting. A semantic prototype complementation module synthesizes old-class prototypes in each new subspace, updating predictions for old classes without replaying old data. Compared with SE, EASE scales to hundreds of classes and maintains low memory overhead by freezing the shared backbone.

Where EASE adds new experts deterministically per task, Differentiable Encoders & Ensembles (DE&E) [26] route updates dynamically. DE&E employs a frozen feature extractor and multiple classifiers. A differentiable soft-kNN gate selects the most relevant classifiers based on feature-space similarity, and only those receive gradient updates. This concentrates plasticity where it is most

2026 JINST 21 P10004

needed, without assuming known task boundaries. Predictions from all classifiers are combined through a similarity-weighted vote.

In contrast, Selective Expert Ensemble Diversification (SEED) [27] assumes explicit task boundaries and assigns exactly one expert to each task. Expert selection is based on similarity to Gaussian class prototypes, and only the chosen expert is fine-tuned. SEED enhances specialization and diversity across experts but depends on fixed prototypes and cannot easily adapt to continuous domain shifts.

When task boundaries are unknown, the Evolved Mixture Model (EEM) [28] detects novel distributions autonomously. It monitors the Hilbert-Schmidt Independence Criterion between each expert's latent space and current data, spawning a new VAE — classifier pair when dependence falls below a threshold. This dynamic expansion enables unsupervised adaptation to drifting data streams, at the cost of growing model complexity.

Overall, ensemble-based CML methods mitigate performance degradation by either (i) averaging parameters (CoMA, CoFiMA), (ii) constraining or expanding representational subspaces (SE, EASE), or (iii) routing or selecting experts (DE&E, SEED, EEM). CoMA and CoFiMA are lightweight but rely on smooth loss connectivity; SE and EASE improve geometric stability with additional parameters; DE&E and SEED enhance specialization through selective updating yet require gating or task information; and EEM adapts autonomously but expands the ensemble over time. These trade-offs — between runtime cost, memory footprint, reliance on task boundaries, and resilience to unlabelled distribution drift — define the design space for ensemble continual learning and motivate the approach developed in this work.

3 Dataset description

The datasets used in this study consist of DigiOccupancy maps recorded during the 2018 (Run2) and 2022 (Run3) proton-proton collision data-taking campaigns at the CMS experiment. These datasets have been certified as suitable for physics analyses by CMS DQM and are registered as "*Golden JSON*" [3]. The 2018 dataset represents a subset of all data recorded by the detector that year, comprising 6 runs collected over a 21-day period (an average of 3.5 days between runs). Similarly, the 2022 dataset is a limited selection of 20 runs spanning a 139-day period (an average of approximately one week between runs). Each DigiOccupancy map is partitioned into 3D spatial coordinates denoted by $i\eta \in [-32, 32]$, $i\phi \in [1, 72]$, and $depth \in [1, 7]$. Here, $i\eta$ is proportional to the pseudorapidity (η) which represents the angle relative to the beamline, $i\phi$ is proportional to the azimuthal angle (ϕ), and $depth$ is approximately the segmented radial component within the HCAL. A mask was applied over the datasets so that only the HE portion of the HCAL sub-detector was considered. The HE was chosen given that the endcap regions exhibit a more complex detector geometry than the HB, including more detector depths, equivalent depths located at physically different distances from the interaction point, etc. (see figure 1(b)). Therefore, the HE provides both a challenging and representative test case for the CML-based DQM methods proposed here. Including data from the HB would increase the dataset size but is not expected to alter the conclusions of this study.

These datasets exhibit both *small shifts* and *large shifts* in the data distributions. Small shifts can be observed in figure 2, which shows the *mean DigiOccupancy* value across the entire HE on a per-LS basis. The mean DigiOccupancy decreases during the duration of individual runs due to a decrease in the instantaneous luminosity resulting from the many particle interactions that occur [29]. Further shifts are observed to occur between runs, as well as between years, caused by changing machine

2026 JINST 21 P10004

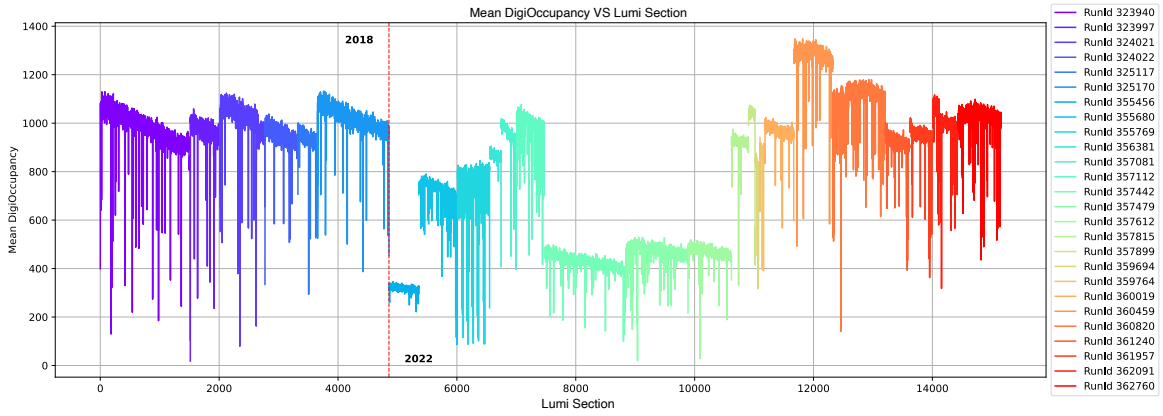


Figure 2. Mean DigiOccupancy versus LS. The red vertical line delineates data originating from 2018 (left) and data originating 2022 (right). The data is temporally ordered both within and between runs, however it should not be assumed that run data is recorded in equally spaced intervals of time.

parameters. For instance, there is an approximately 60% reduction in the mean DigiOccupancy value between the final run considered in the 2018 dataset (RunID 325170), and the first run considered in the 2022 dataset (RunID 355456), and an approximately 188% increase between the first run considered in the 2022 dataset (RunID 355456) and the final run considered in the 2022 dataset (RunID 362760). As will be further discussed and demonstrated in section 5, these small shifts accrue over time and become sufficient to degrade the performance of ML-based DQM models that have been trained on static datasets and which do not employ CML techniques.

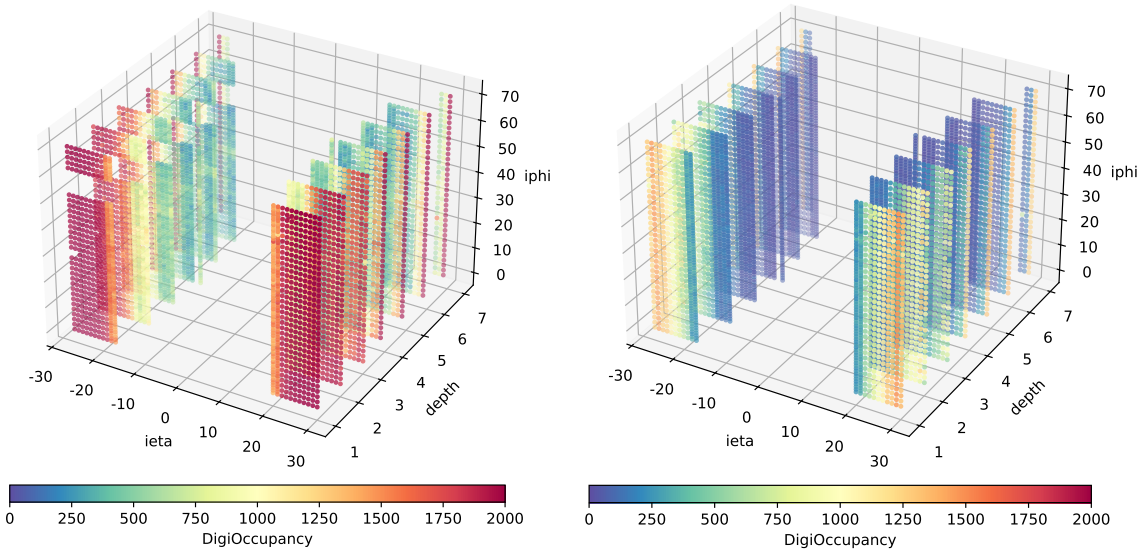
In contrast, figure 3 demonstrates an example of a *large shift* that occurred inbetween the 2018 and 2022 data taking campaigns. In particular, during 2018 an approximately 40° section in the azimuthal angle (ϕ) of the HE section of the HCAL lost power, resulting in a large portion of the sub-detector producing no DigiOccupancy values [30]. This is reflected in the upper left portion of the 2018 DigiOccupancy plot shown (left, $Depth \in [1, 7]$, $ieta \in [-30, -16]$, $iphi \in [55, 62]$). As a result, it is expected that a DQM ML model trained on data originating from the 2018 dataset will most likely perform very poorly when evaluated on data originating from the 2022 dataset. Although only a limited number of runs from 2018 and 2022 are used, the presence of these two types of distributional shits in the recorded data presents an excellent basis with which to test and evaluate methods of CML applicable to ML-based DQM. Adding additional runs or additional years of data would primarily increase the amount of nominal data, but is not expected to change the central conclusions regarding model performance and adaptation.

4 Methodology

4.1 DepthViT architecture overview

Here, we introduce a novel masked autoencoder architecture with depth-wise embeddings and depth-wise attention, referred to as *DepthViT*. A graphical representation of the DepthViT architecture is shown in figure 4. To better understand the motivation behind this architecture, there is need to further discuss the data distributions produced by CMS DQM systems. Between the individual depths of the HCAL detector, the DigiOccupancy data distributions are not necessarily spatially related. This

2026 JINST 21 P10004



(a) (b)

Figure 3. (a) DigiOccupancy map corresponding to *Run325117*, LS 0 (2018). Note the large absence of DigiOccupancy values in the upper left hand corner which resulted from a noted detector failure. (b) DigiOccupancy map corresponding to *Run355456*, LS 0 (2022).

is because a jet (a collimated spray of hadronic particles) moving through the HCAL detector will initiate further hadronic showers, leading to different shower profiles for different detector depths. The features extracted at one detector depth might not therefore have spatial information that is significant with respect to other detector depths. This is in contrast to a traditional photo image, in which the red, green, and blue channels are expected to all represent the same physical image, leading to contextually rich feature extraction between channels. In this section, the terms *channel* and *depth* are used interchangeably, depending on if the input data is image data or DigiOccupancy maps.

In a typical vision transformer architecture [31], and in vision models more generally [32], the first step is ordinarily to "patchify" the image. This step often involves strided convolutions of the form:

$$Y_f = \sum_{c=1}^{C} R_{f,c} * X_c, \tag{4.1}$$

where $X \in \mathbb{R}^{H\times W\times C}$ is the input image, $R \in \mathbb{R}^{H''\times W''\times C\times F}$ is the convolution kernel (or *filter*), and $Y \in \mathbb{R}^{H'\times W'\times F}$ is the outputted patch. H, W, C, and F refer to height, width, channel number, and filter number, respectively. '$*$' is the convolution operation which is a strided dot product between the overlapping values of R and X, which is then summed over the channel dimension. This summation operation immediately mixes information across the input channel dimension. This is appropriate when the images have a channel-wise pixel-level symmetry. This can safely be assumed for RGB color channels in images because the three colors correspond to the same physical point in space, but is not necessarily true in CMS detector data due to different tracks traveling different distances within the detector, and the angled and curved motion of charged particle tracks bending due to the magnetic field. To preserve channel-specific information during patching, the DepthViT architecture uses *depth-wise* convolutional patching, shown in eq. (4.2). Here the key difference is that the channel

2026 JINST 21 P10004

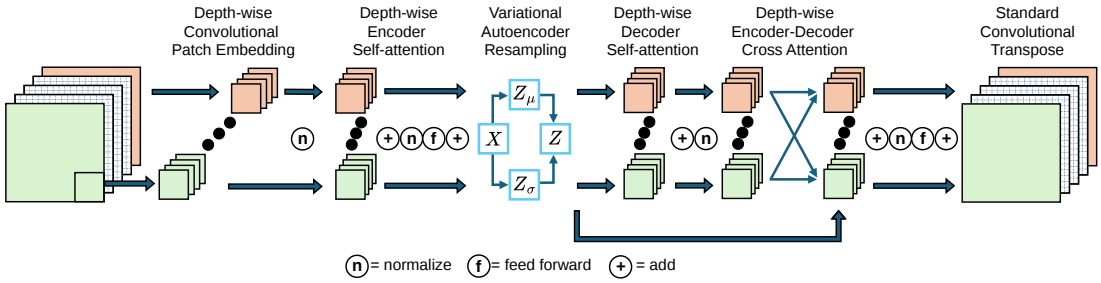


Figure 4. DepthViT variational autoencoder architecture which uses depth-wise convolutional embeddings and depth-wise attention, latent space resampling of the encoder outputs.

dimension is not summed over, hence the k, c dimensionality in place of f:

$$Y_{k,c} = R_{k,c} * X_c, \qquad Y \in \mathbb{R}^{H' \times W' \times C \times F}. \tag{4.2}$$

This difference is visualized in figure 5(a), which shows the traditional approach, and figure 5(b), which shows the approach taken within DepthViT. Because the convolutional kernel weights are shared across all patches from the image, some level of information about channel-specific relationships between patches is retained, while most channel-wise relationships are abstracted away. To reintroduce channel-wise relationships, we additionally introduce a novel, depth-wise attention mechanism.

A traditional vision transformer attention mechanism involves taking as input a collection of embedded image patches, $Z \in \mathbb{R}^{L \times D}$, where L is the sequence length and D is the embedding dimension. For vision transformers using convolutional patch embeddings, L is the flattened input image sequence length ($L = H' \cdot W'$) and D is equal to the number of kernel filters ($D = F$). Learned linear projection operators W_Q, W_K, and W_V are then applied along the embedding dimension to produce:

$$Q = ZW_Q, \qquad K = ZW_K, \qquad V = ZW_V, \tag{4.3}$$

where $Q, K, V \in \mathbb{R}^{L \times D}$ are the query, key, and value projections of the embedded image patches, respectively. Next, an outer product is taken of Q and K^T along the sequence length dimension to compute the scores, S:

$$S_{ij} = \frac{1}{\sqrt{D}} \sum_{r=1}^{D} Q_{ir} K_{jr}, \qquad S \in \mathbb{R}^{L \times L}. \tag{4.4}$$

The scores are then passed through a softmax function to get the attention weights, A:

$$A_{ij} = \frac{\exp(S_{ij})}{\sum_{j'=1}^{L} \exp(S_{ij'})}, \qquad A \in \mathbb{R}^{L \times L}. \tag{4.5}$$

Finally, the attention weights are taken as a dot product along the key and value sequence length dimension to get the attention output which is added back to the original embedded input:

$$O_{ij} = \sum_{r=1}^{L} A_{ir} V_{rj} \qquad O \in \mathbb{R}^{L \times D}. \tag{4.6}$$

2026 JINST 21 P10004

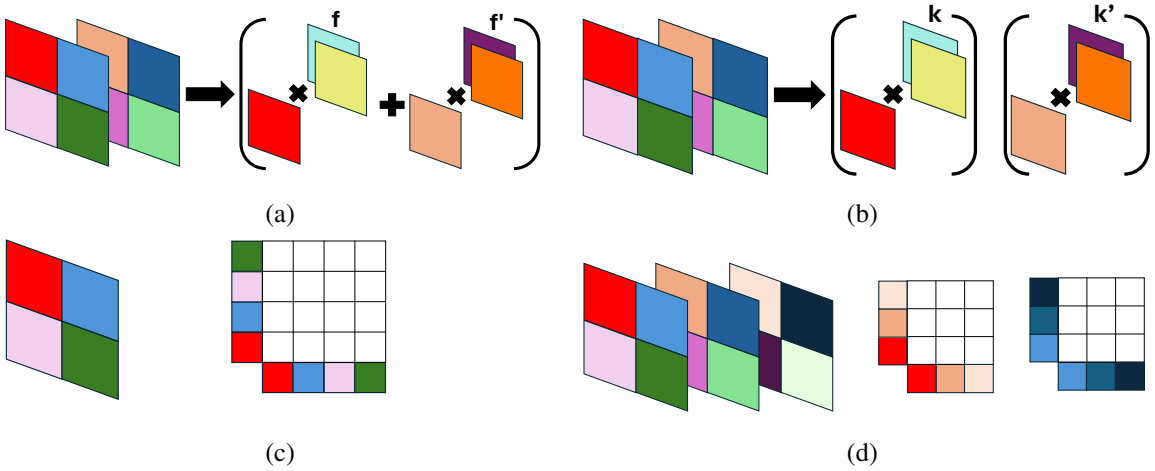


Figure 5. (a) Traditional convolution procedure used within vision transformer architectures, where f represents the shared kernel filters. (b) convolution procedure used within the DepthViT architecture, where k and k' represent the filters derived separately for each channel. (c) Traditional attention mechanism used within vision transformer architectures where the attention mechanism attends along the input sequence. (d) Depthwise attention mechanism used within the DepthViT architecture where the attention mechanism instead attends in a depth-wise manner.

In such an attention framework, the attention weights are considered based on relationships between patches, as visualized in figure 5(c). In the DepthViT architecture, the attention mechanism considers relationships between patches along the channel dimension, rather than along the patch dimension. This difference is visually demonstrated in figure 5(d). Here, because we are using depth-wise convolutional patch embedding, $Z \in \mathbb{R}^{L\times C\times F}$ and $Q, K, V \in \mathbb{R}^{L\times C\times F}$, where L is again the flattened input sequence length ($L = H' \cdot W'$), C is the channel dimension, and F is the number of kernel filters per channel. The scores are then computed by taking an outer product along the channel dimensions:

$$S_{ijkl} = \frac{1}{\sqrt{F}} Q_{ijl} K_{ikl}, \qquad S \in \mathbb{R}^{L\times C\times C\times F}, \tag{4.7}$$

where i indexes over patches, l over filters, and j and k are projections along the channel dimension. The attention maps also have a softmax applied along the channel dimension rather than the sequence dimension.

$$A_{ijkl} = \frac{\exp(S_{ijkl})}{\sum_{k'=1}^{C} \exp(S_{ijk'l})}, \qquad A \in \mathbb{R}^{L\times C\times C\times F}. \tag{4.8}$$

Finally, the output vectors are computed from an inner product along the key and value channel dimensions with C as the number of channels:

$$O_{ijl} = \sum_{k=1}^{C} A_{ijkl} V_{ikl}, \qquad O \in \mathbb{R}^{L\times C\times F}. \tag{4.9}$$

This alternative attention mechanism broadcasts along the sequence dimension computing cross-depth attention. This eliminates information sharing between patches in a layer, but can save substantially on parameters for cases where cross-channel symmetries are more significant. Within DepthViT this depth-wise attention is applied in both a self-attention manner, in which q, k, and v all originate from the same

2026 JINST 21 P10004

sequence, and in a cross attention manner, in which q originates from the decoded sequence and k and v originate from the sequence before decoding. After the attention reweighting is completed, the images are transformed back to the original dimensions using a standard convolutional transpose which is analogous to an inverse convolution operation which mixes channel information during reconstruction.

To improve the performance of the model in AD, some subset of all input patches (set to 50% for this implementation) are randomly masked with a tunable likelihood during training. This masking acts as an information bottleneck to ensure that the model does not learn to simply pass the identity through the transformer encoder and decoder. Similarly, after depth-wise encoding the sequence is passed through an optional light variational bottleneck which encourages smoother, more noise-tolerant latent features. As a result of these novel vision transformer techniques, DepthViT is a lightweight masked autoencoder architecture that relies on approximately 300,000 parameters. In comparison, the smallest default ViT-B/16 [31], commonly used for such tasks, has 86M parameters, and the lightweight Self-Distilled Masked Auto-Encoder [33] has 3M parameters. In addition, the computational complexity changes from $O(L^2D)$, where L is the number of patches or sequence length and D is the number of filters or embed dimension, to $O(LC^2D)$ which means that these architectures also scale well in compute time when $C \ll L$.

4.2 DepthViT based anomaly detection

In this application, AD is performed by computing Z-score values and using them to quantify deviations from nominal detector behavior. This technique is performed in the following manner. A DepthViT model is first trained on pristine detector data which contains no anomalies. The *prediction error* (pred_{err}), defined as the difference between the model's prediction and the target data, is computed on a per-pixel basis by deploying the model on its associated test data. The mean (μ_{err}) and standard deviation (σ_{err}) of these errors are calculated for each pixel, and used as a baseline performance metric for the model. During inference, a time-window technique is employed, in which T LS are processed together. For each time-window T, the Z-score is calculated in the following manner:

$$Z = \frac{\text{abs}(\sum_{t=1}^{t=T} \text{pred}_{\text{err}} - T * \mu_{\text{err}})}{\sqrt{T} * \sigma_{\text{err}}}, \tag{4.10}$$

where abs is the absolute value function, and t refers to the individual LS present within time-window T that are summed over. The intuition behind this procedure is that for time-window LS values which do not contain anomalies, it is expected that pred_{err} values will represent a sampling from a distribution with mean of μ_{err} and therefore the quantity in the numerator of eq. (4.10) will equate to zero. For time-windowed LS which do contain anomalies, pred_{err} is no longer sampled from a random distribution but is instead expected to consistently lie either above or below the μ_{err} value, implying that over the entire time-windowed period the Z-score will consistently increase in value and improve the capacity to identify anomalies.

A variety of different threshold mechanisms were considered for use in determining the presence of anomalies, including a single threshold value, θ, such that if $Z > \theta$ for any Z-value then the data would be labeled anomalous. Ultimately, this single threshold value approach was found to not offer robust performance and instead led to an increased false positive rate (FPR) when deployed on datasets originating from outside the training data. Instead, a *gap-score* method was employed which identified large gaps in the Z-score distribution to indicate the presence of anomalies. For this method, all

2026 JINST 21 P10004

Z-score values are calculated within a given LS. The Z-score distribution is then normalized to the largest Z-score value, Z_1, so that the distribution is bounded between $(0, 1]$, and the distribution is ordered in ascending order. The gap-score, G, is calculated as:

$$G = Z_1 - Z_2, \tag{4.11}$$

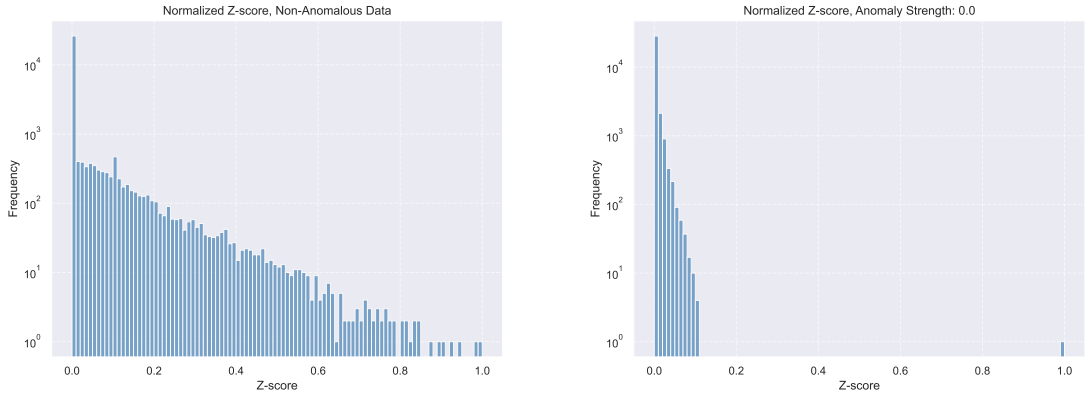
where Z_2 is the second largest Z-score value. For any G value greater than G_0 the LS is labeled as anomalous, where G_0 is a detection parameter that can be optimized for specific use cases. The use of only the two largest Z-score values in the gap-score calculation is due to only single anomalies per LS being considered in this work. However, the same technique can easily be extrapolated to the presence of multiple simultaneous anomalies by simply calculating G between each sequential pair of ordered Z-score values and identifying any LS with $G > G_0$ as being anomalous. Figure 6 shows the normalized Z-score distribution for a variety of different scenarios including a well-performing model deployed on non-anomalous data (top left), deployed on anomalous data (top right), and a degraded model deployed on anomalous data (bottom). Both the well-performing model deployed on non-anomalous data and the degraded model deployed on anomalous data produce similar distributions, neither of which would be labeled as anomalous. This tendency for a degraded model to label all data as non-anomalous helps reduce the FPR, while a single well-performing model within the overall ensemble of models reduces the False Negative Rate (FNR).

4.3 Ensembling technique

The lightweight nature of the DepthViT architecture makes it ideal for use in ensembling methods, which have the potential to constrain memory resources. For this work, a straightforward ensembling strategy was employed and evaluated for its ability to counteract the effects of model degradation. Separate DepthViT models were trained for each run, using training and test datasets derived exclusively from that run. Ensembles of DepthViT models were then deployed in parallel on the same incoming data stream, and per-model Z-score distributions were calculated. These Z-score distributions were fed into the gap-score equation (eq. (4.10)) which produced an anomalous/non-anomalous classification prediction from each model of the overall ensemble. If any individual model classified the data as anomalous, then the ensemble output prediction is an anomalous classification for that LS. This approach is trivially parallelizable, implying it does not necessarily increase the compute-time for systems with sufficient processing power. Anomaly classification tests were performed on test data that were injected with synthetic anomalies. *Anomaly factors* of "0.0", "0.2", "0.4", "0.6", "0.8", "1.5", and "2.0" were considered for this study. The location of an anomaly was chosen by randomly selecting with equal probability an individual pixel within the HE, and the DigiOccupancy for that pixel was then multiplied by the anomaly factor value. These anomaly strengths reflect actual detector abnormalities, with 0.0 representing a dead channel, 0.2 to 0.8 representing degraded channels, and 1.5 and 2.0 representing hot channels. Time-windows of varying lengths were considered for both scenarios in which the anomalies were transient or persistent. Two different scaling methods, *quantile* and *max*, were employed in parallel as part of a pre-processing step. This was due to the empirical observation that max scaling provided better sensitivity to anomaly factors < 1.0, while quantile scaling provided better sensitivity to anomaly factors > 1.0. Quantile scaling was applied using the following formula:

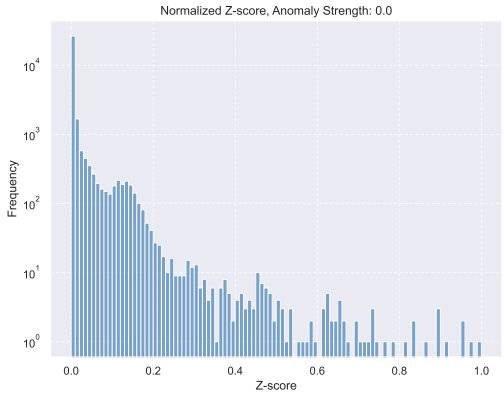
$$Y_{\text{scaled}} = \frac{X - \text{Q}_{25}(X)}{\text{Q}_{75}(X) - \text{Q}_{25}(X) + \epsilon} \tag{4.12}$$

2026 JINST 21 P10004



(a)

(b)



(c)

Figure 6. (a) Normalized Z-score distribution of a well-performing DepthViT model when deployed on non-anomalous data. (b) Normalized Z-score distribution of a well-performing DepthViT model when deployed on anomalous data with anomaly factor of 0.0. (c) Normalized Z-score distribution of a severely degraded DepthViT model when deployed on anomalous data with anomaly factor of 0.0.

where X represents the DigiOccupancy values for a single LS, $Q_{25}(X)$ is the sampled 25th percentile value, $Q_{75}(X)$ is the 75th percentile value, ϵ is a small value to offset any potential divide-by-zero errors and Y_{scaled} is the resulting scaled output data. *Max scaling* was applied using:

$$Y_{\text{scaled}} = \frac{X}{X_{\text{max}}} \tag{4.13}$$

where again X represents the input DigiOccupancy values for a single LS, and X_{max} is the maximum value within the input array for a single LS. Only one version of the scaled data was fed into an individual DepthViT model, as determined by which scaled data version the model was trained on. A graphical workflow of this process is shown in figure 7. The term *sub-ensemble* will be used here to describe a Max Scaling DepthViT model placed in parallel with a Quantile Scaling DepthViT, and will represent the smallest building block used to create larger ensembled systems.

2026 JINST 21 P10004

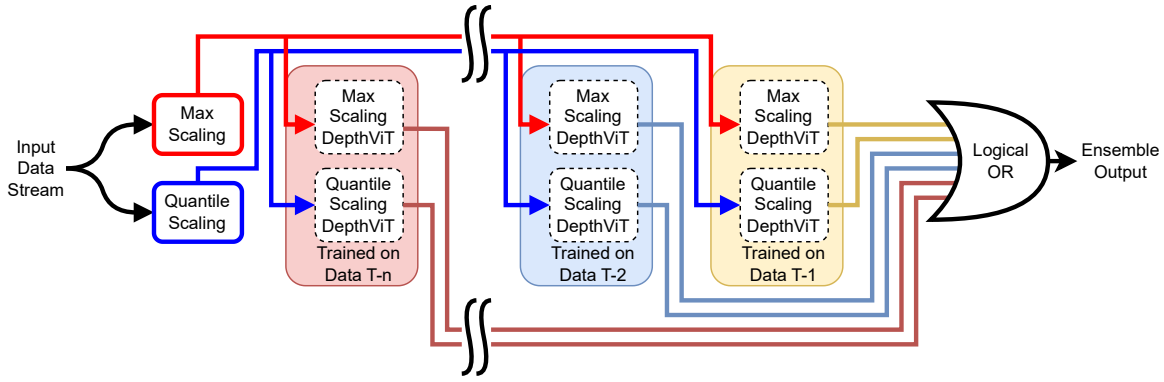


Figure 7. Flow chart detailing the ensembling scheme implemented within this work. Input data is scaled separately using max scaling and quantile scaling operations. The data is then fed in a parallel fashion into individual sub-ensembles which consist of independent DepthViT models trained on max-scaled data and quantile-scaled data respectively. The outputs from all sub-ensembles is combined using a logical OR operation, where if any sub-ensemble labels the data as being anomalous then the entire ensemble output label is anomalous.

5 Results and discussion

5.1 Baseline model performance

Presented first are baseline results demonstrating the performance of a single DepthViT sub-ensemble model. This model was trained on *Run323940* data which originates from the 2018 dataset. The HE data was first randomized by LS and then divided using a 60%/20%/20% split into *training data*, *test data*, and *validation data*. The μ_{err} and σ_{err} values were derived by deploying the trained model on its associated test data. Synthetic anomalies of the previously-described values were injected into the validation data to simulate run-time inference and evaluate performance. Only one anomaly factor was considered at a given time (anomalies of different values were not injected into the same dataset). Optimization studies were performed on a variety of hyperparameters including patch size, number of kernels, masking ratio, and time-window length T. The optimal hyperparameter values were selected according to their anomaly detection performance. Varying the patch size did not lead to improved performance, however it was found that decreasing the patch size did lead to longer inference compute times. A gap-score value of $G_0 = 0.3$ was selected as the optimized threshold for events to be classified as anomalous. Increasing T was found to improve recall performance for persistent anomalies across all anomaly factors, however recall performance was found to degrade as a result for transient anomalies. Precision performance improved for increasing T, however, the improvements are marginal except in the shortest transient cases. As a trade-off between these results, a time-window value of $T = 5$ was selected as an optimized value. Table 1 shows the performance of a DepthViT model using the optimized hyperparameter values across all anomaly factors considered for the metrics of precision, recall, F1 score, FPR, & FNR. It was observed that the model performed very well, even for subtle anomaly factors (e.g. anomaly factor of 0.8), therefore additional anomaly factors were included in this study.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}};\ \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}};\ \text{F1 Score} = \frac{2}{\frac{1}{\text{Recall}}+\frac{1}{\text{Precision}}}\ \text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}};\ \text{FNR} = \frac{\text{FN}}{\text{FN}+\text{TP}} \tag{5.1}$$

2026 JINST 21 P10004

Table 1. Relevant performance metrics of a DepthViT model trained on *Run323940* using optimized hyperparameters (patch size = 12, kernels = 8, mask ratio = 0.5, T = 5, layers = 5, latent dimension = 56, $G > 0.3$). Inference was performed using synthetic anomalies of the listed strengths that were injected into the associated test dataset.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	1.000	1.000	1.000	0.000	0.000
0.2	1.000	1.000	1.000	0.000	0.000
0.4	1.000	1.000	1.000	0.000	0.000
0.6	1.000	1.000	1.000	0.000	0.000
0.8	1.000	0.967	0.983	0.000	0.033
0.9	1.000	0.783	0.879	0.000	0.217
1.1	1.000	0.667	0.800	0.000	0.333
1.25	1.000	0.983	0.992	0.000	0.017
1.5	1.000	1.000	1.000	0.000	0.000
2.0	1.000	1.000	1.000	0.000	0.000

5.2 Baseline model performance degradation over time

In order to quantify the performance degradation that occurs due to shifts in the incoming data streams, the same baseline sub-ensemble model presented in section 5.1 was then evaluated on data originating from subsequent runs. Figure 8 presents the FPR and FNR of this model when evaluated on both its own validation data and subsequent 2018 and 2022 data for scenarios in which anomalies are present in each of the 5 LS considered within the time-window. It is observed that for subsequent runs originating in 2018 (*Run323997* through *Run325170*) which exhibit only small shifts in the incoming data streams, the model's FPR tends to decline while the FNR increases (a result which was referenced in section 4). For runs originating in 2022 (*Run355456* through *Run362760*) which demonstrate large shifts in the incoming data streams, the model performs extremely poorly, failing to identify large numbers of anomalous events. The full performance metrics for this test are presented in table 2.

Table 2. Performance metrics for the baseline DepthViT model trained on *Run323940* data, when deployed over all runs within the full dataset and implementing no continual learning mitigation techniques.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	0.874	0.549	0.674	0.079	0.451
0.2	0.863	0.498	0.631	0.079	0.503
0.4	0.841	0.420	0.560	0.079	0.580
0.6	0.810	0.338	0.477	0.079	0.662
0.8	0.757	0.247	0.372	0.079	0.753
1.5	0.800	0.316	0.453	0.079	0.684
2.0	0.865	0.508	0.640	0.079	0.492

2026 JINST 21 P10004

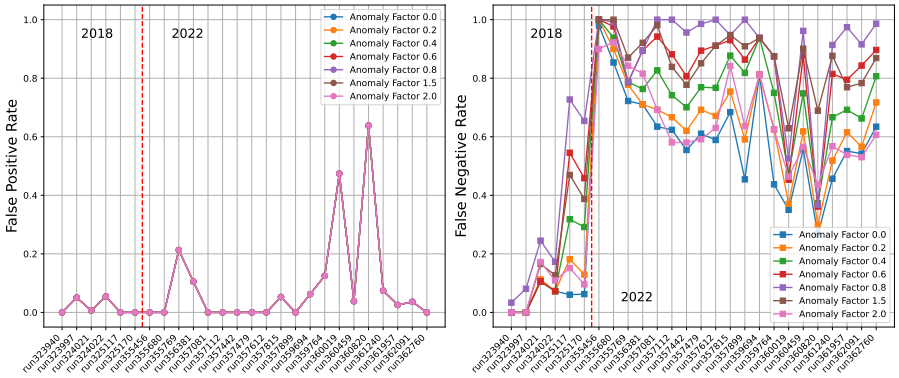


Figure 8. FPR (left) and FNR (right) of DepthViT model trained on *Run323940* using the same hyperparameters detailed in table 1 when evaluated on test data from subsequent runs. Pointers indicate which dataset the model was evaluated on, while the red dashed line delineates runs originating in 2018 vs. 2022. Gradual model degradation is observed for 2018 data due to small shifts in the incoming data stream, while severe degradation is observed for 2022 data due to large shifts.

5.3 Model ensembling & mitigation techniques

Multiple techniques for mitigating the observed model degradation were investigated and tested. This included periodically updating the μ_{err} and σ_{err} values used in eq. (4.10) via the latest data stream, ensembling together models trained on historic datasets with models trained on the most recent dataset, and the combination of these two techniques. Presented first are studies in which only μ_{err} and σ_{err} values have been recalculated. For this test, the baseline sub-ensemble model from *Run323940* was redeployed on test data from each subsequent run allowing for new μ_{err} and σ_{err} statistics to be computed. These updated values were then applied during inference on the corresponding validation data which included synthetically injected anomalies. The model's weights were not updated and only the performance metrics used in the Z-score calculation were updated, implying that this does not constitute *fine-tuning* of the model to the latest data. The resulting FPR and FNR values produced by this technique are shown in figure 9. The result is that large shifts in the incoming data which would otherwise produce severe model degradation instead produce results characteristic with small shifts in the incoming data. The significance of this result is that because it does not require fine-tuning or further model training, it represents a potentially computationally resourceful mitigation technique that relies solely on the continued availability of pristine detector data. Table 3 presents the relevant performance metrics from this study. It is observed that this techniques improves the FNR significantly, while improvements to the FPR are less substantial.

Presented next are results obtained by ensembling together multiple sub-ensemble models, while the μ_{err} and σ_{err} values are left static and are not updated. For this study, no more than 4 models (each consisting of a quantile scaled and max scaled model in parallel) were ensembled together at a time.

The overall structure of the ensemble follows the flow chart detailed in figure 7. The individual sub-ensemble models were never deployed on data that was used to train an individual model within the larger ensemble in order to avoid potential data contamination. Each time a new sub-ensemble model was introduced, the oldest sub-ensemble model was retired in order to maintain a fixed ensemble size. Table 4 shows the resulting performance metrics of this study. An important conclusion to note is that the static ensembling approach further improves the FNR values, however the FPR worsens compared to the baseline with no continual learning techniques employed (table 2).

2026 JINST 21 P10004

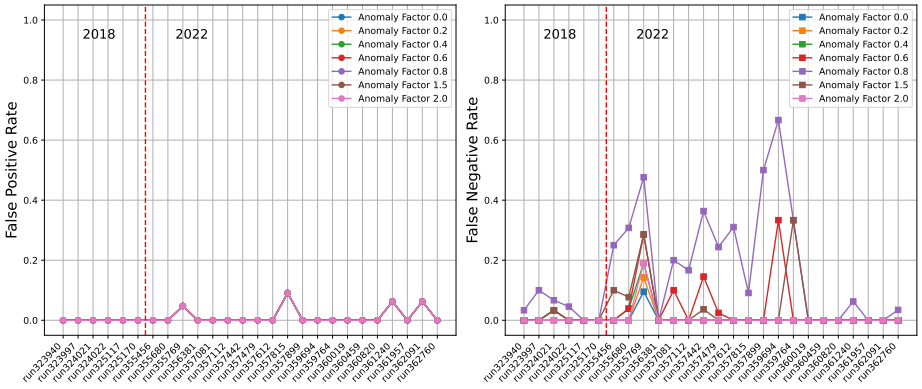


Figure 9. FPR (left) and FNR (right) of the baseline DepthViT model trained on *Run323940*, using the same hyperparameters detailed in table 1 but with updated μ_{err} and σ_{err} values derived from deployment on the validation data of the dataset being evaluated. Pointers indicate which test dataset the model was evaluated on, while the red dashed line delineates runs originating in 2018 vs. 2022.

Table 3. Performance metrics for the baseline *Run323940* model presented in section 5.1 when the relevant μ_{err} and σ_{err} values have been updated using the validation data of the run being deployed on.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	0.993	0.997	0.995	0.007	0.003
0.2	0.993	0.995	0.994	0.007	0.005
0.4	0.993	0.990	0.992	0.007	0.010
0.6	0.993	0.968	0.981	0.007	0.032
0.8	0.992	0.863	0.923	0.007	0.137
1.5	0.993	0.977	0.985	0.007	0.024
2.0	0.993	0.993	0.993	0.007	0.007

Table 4. Performance metrics for an ensemble of models generated on 4 total runs, where a new set of models were introduced every run of the dataset. The older models within the ensemble do not have their μ_{err} and σ_{err} values updated and instead remain static.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	0.779	1.000	0.876	0.283	0.000
0.2	0.779	1.000	0.876	0.283	0.000
0.4	0.779	0.998	0.875	0.283	0.002
0.6	0.777	0.983	0.868	0.283	0.017
0.8	0.762	0.908	0.829	0.283	0.092
1.5	0.777	0.988	0.870	0.283	0.012
2.0	0.779	1.000	0.876	0.283	0.000

2026 JINST 21 P10004

Table 5. Performance metrics for an ensemble of models generated on 4 total runs, where a new set of models were introduced for every run of the dataset.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	0.988	1.000	0.994	0.012	0.000
0.2	0.988	0.998	0.993	0.012	0.002
0.4	0.988	0.995	0.992	0.012	0.005
0.6	0.988	0.985	0.987	0.012	0.015
0.8	0.987	0.891	0.937	0.012	0.109
1.5	0.988	0.990	0.989	0.012	0.010
2.0	0.988	0.997	0.993	0.012	0.003

Finally, a study was performed which combined both of these techniques, in which a new sub-ensemble model was trained and added to the ensemble for every run, while the μ_{err} and σ_{err} values were updated for all individual sub-ensemble models within the ensemble using the validation data of the latest run. The entire ensemble was then evaluated using the latest run's test data. The result, shown in table 5, is an ensemble which exhibits > 98% precision across all anomaly factors tested, and similar recall performance for all anomaly factors except the most subtle tested (0.8). Most importantly, both the FPR and FNR values are significantly improved relative to the application of either technique alone, and relative to baseline performance which included no continual learning mitigation techniques.

5.4 Model ensembling performance compared to latest model performance

A final study was performed to compare the performance of an ensembled set of models to that of a single model only trained on the latest data stream. To test this, ensembling of older models was employed, but instead a model was generated and employed for every run, and then immediately discarded. This would represent a scenario in which no older models were able to contribute to the performance of the overall ensemble. These results are presented in table 6. It is observed that ensembling provides an improvement of 100% and 55% to the FNR for anomaly factors of 0.0 and 1.5, respectively (strong anomalies), and an improvement of 11% to the FNR for anomaly factor 0.8 (subtle anomalies). These results indicate that the ensembling improves the overall performance and stability of the system.

Table 6. Performance metrics for an ensemble of models consisting of 2 total models (one for each employed scaling technique) trained only on the latest run.

Anomaly Factor	Precision	Recall	F1 Score	FPR	FNR
0.0	0.997	0.998	0.998	0.003	0.002
0.2	0.997	0.993	0.995	0.003	0.007
0.4	0.997	0.988	0.992	0.003	0.012
0.6	0.997	0.972	0.984	0.003	0.029
0.8	0.996	0.878	0.933	0.003	0.122
1.5	0.997	0.978	0.987	0.003	0.022
2.0	0.997	0.997	0.997	0.003	0.003

2026 JINST 21 P10004

6 Conclusion

In this work, DepthViT, a novel, lightweight ML architecture, has been introduced and evaluated on the task of anomaly detection in HEP data. The DepthViT architecture varies from traditional vision transformers in that input channels do not share kernel filters, and that the attention mechanism is applied depthwise rather than along the input sequence. Two continual learning approaches have been evaluated and applied. The first was an updating of the evaluation statistics used to perform a Z-score calculation using data drawn from the latest input stream, while the second was a straightforward ensembling scheme which ensembled together models trained on prior data, in addition to the latest input data stream. It was observed that employing both of these techniques together improved the performance of the overall system when compared to individual models and also made the system more resilient to changes in the incoming data stream. These findings suggest that lightweight, modular ensemble architectures can provide a practical approach to reducing the effects of model degradation without the need for retraining on historical data. The broader implications of this work extends beyond HEP, as the same continual adaptation strategy can be directly applied to industrial monitoring environments where data naturally evolves over time, such as manufacturing lines with aging sensors, or the emergence of new failure modes in advanced manufacturing processes. Additionally, the novel DepthViT architecture introduced could be extended to applications in other branches of research where critical information for the task is patch-local but where channels do not necessarily correspond to the same physical point in space. Example tasks could include classification and regression based on key peaks in spectral data, or local feature extraction from multi-channel optical data.

Acknowledgments

This work was supported in part by a Phase I contract from the U.S. Department of Energy Small Business Technology Transfer Program (contract number DE-SC0024776). Individuals have received support from the Belgian Fonds de la Recherche Scientifique, and Fonds voor Wetenschappelijk Onderzoek; the Brazilian Funding agencies (CNPq, CAPES, FAPERJ, FAPERGS, and FAPESP); SRNSF (Georgia); the Bundesministerium für Bildung und Forschung, the Deutsche Forschungsgemeinschaft (DFG), under Germany's Excellence Strategy — EXC 2121 "Quantum Universe" — 390833306, and under project number 400140256 — GRK2497, and Helmholtz-Gemeinschaft Deutscher Forschungszentren, Germany; the National Research, Development and Innovation Office (NKFIH) (Hungary) under project numbers K 128713, K 143460, and TKP2021-NKTA-64; the Department of Atomic Energy and the Department of Science and Technology, India; the Ministry of Science, ICT and Future Planning, and National Research Foundation (NRF), Republic of Korea; the Lithuanian Academy of Sciences; the Scientific and Technical Research Council of Turkey, and Turkish Energy, Nuclear and Mineral Research Agency; the National Academy of Sciences of Ukraine; the US Department of Energy.

Code Availability Statement. This article has associated code in a code repository. The code for this work is made available at the following link along with simulated demo data for testing and development: https://github.com/daleadamjulson/DepthViT_CML.

2026 JINST 21 P10004

References

[1] H. Wang et al., *Scientific discovery in the age of artificial intelligence*, *Nature* **620** (2023) 47.

[2] S. Kumari et al., *A Comprehensive Investigation of Anomaly Detection Methods in Deep Learning and Machine Learning: 2019–2023*, *IET Inform. Sec.* **2024** (2024) 8821891.

[3] V. Azzolini et al., *The Data Quality Monitoring software for the CMS experiment at the LHC: past, present and future*, *EPJ Web Conf.* **214** (2019) 02003.

[4] A.A. Pol, *Machine Learning Anomaly Detection Applications to Compact Muon Solenoid Data Quality Monitoring*, Ph.D. thesis, Université Paris-Saclay, Paris, France (2020).

[5] V. Belis, P. Odagiu and T.K. Aarrestad, *Machine learning for anomaly detection in particle physics*, *Rev. Phys.* **12** (2024) 100091 [arXiv:2312.14190].

[6] M.W. Asres et al., *Spatio-Temporal Anomaly Detection with Graph Networks for Data Quality Monitoring of the Hadron Calorimeter*, *Sensors* **23** (2023) 9679 [arXiv:2311.04190].

[7] M.W. Asres et al., *Long Horizon Anomaly Prediction in Multivariate Time Series with Causal Autoencoders*, *PHM Soc. Eur. Conf.* **7** (2022) 21.

[8] M.W. Asres et al., *Data Quality Monitoring for the Hadron Calorimeters Using Transfer Learning for Anomaly Detection*, *Sensors* **25** (2025) 3475 [arXiv:2408.16612].

[9] N. Strobbe, *The upgrade of the CMS hadron calorimeter with silicon photomultipliers*, 2017 *JINST* **12** C01080.

[10] L. Evans and P. Bryant, *LHC Machine*, 2008 *JINST* **3** S08001.

[11] CMS collaboration, *The CMS Experiment at the CERN LHC*, 2008 *JINST* **3** S08004.

[12] CMS collaboration, *CMS Technical Design Report for the Phase 1 Upgrade of the Hadron Calorimeter*, CERN-LHCC-2012-015 (2012) [DOI:10.2172/1151651].

[13] CMS collaboration, *The CMS trigger system*, 2017 *JINST* **12** P01020 [arXiv:1609.02366].

[14] A. Seiden, *Characteristics of the ATLAS and CMS detectors*, *Phil. Trans. Roy. Soc. Lond. A* **370** (2012) 892.

[15] L. Tuura, A. Meyer, I. Segoni and G. Della Ricca, *CMS data quality monitoring: Systems and experiences*, *J. Phys. Conf. Ser.* **219** (2010) 072020.

[16] M. McCloskey and N.J. Cohen, *Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem*, in *Psychology of Learning and Motivation*, G.H. Bower, ed., Elsevier (1989), p. 109–165 [DOI:10.1016/s0079-7421(08)60536-8].

[17] J. Kirkpatrick et al., *Overcoming catastrophic forgetting in neural networks*, *Proc. Nat. Acad. Sci.* **114** (2017) 3521.

[18] F. Zenke, B. Poole and S. Ganguli, *Continual Learning Through Synaptic Intelligence*, *Proc. Mach. Learn. Res.* **70** (2017) 3987 [arXiv:1703.04200].

[19] D. Lopez-Paz and M.A. Ranzato, *Gradient Episodic Memory for Continual Learning*, in *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6467–6476, (2017) [arXiv:1706.08840].

[20] S.-A. Rebuffi, A. Kolesnikov, G. Sperl and C.H. Lampert, *iCaRL: Incremental Classifier and Representation Learning*, in the proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, U.S.A., 21–26 July 2017 [DOI:10.1109/cvpr.2017.587].

[21] A.A. Rusu et al., *Progressive Neural Networks*, arXiv:1606.04671.

2026 JINST 21 P10004

[22] A. Mallya and S. Lazebnik, *PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning*, in the proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, U.S.A., 18–23 June 2018 [DOI:10.1109/cvpr.2018.00810].

[23] I.E. Marouf, S. Roy, E. Tartaglione and S. Lathuilière, *Weighted Ensemble Models Are Strong Continual Learners*, arXiv:2312.08977.

[24] T. Doan, S.I. Mirzadeh and M. Farajtabar, *Continual Learning Beyond a Single Model*, arXiv:2202.09826.

[25] D.-W. Zhou, H.-L. Sun, H.-J. Ye and D.-C. Zhan, *Expandable Subspace Ensemble for Pre-Trained Model-Based Class-Incremental Learning*, arXiv:2403.12030.

[26] M. Wójcik, W. Kościukiewicz, T. Kajdanowicz and A. Gonczarek, *Neural Architecture for Online Ensemble Continual Learning*, arXiv:2211.14963.

[27] G. Rypeść et al., *Divide and not forget: Ensemble of selectively trained experts in Continual Learning*, arXiv:2401.10191.

[28] F. Ye and A.G. Bors, *Learning an evolved mixture model for task-free continual learning*, arXiv:2207.05080.

[29] O.S. Bruning et al., *LHC Design Report Vol.1: The LHC Main Ring*, CERN-2004-003-V1 (2004) [DOI:10.5170/CERN-2004-003-V-1].

[30] CMS collaboration, *Search for disappearing tracks in proton-proton collisions at* $\sqrt{s} = 13$ *TeV*, *Phys. Lett. B* **806** (2020) 135502 [arXiv:2004.05153].

[31] A. Dosovitskiy et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv:2010.11929.

[32] K. He et al., *Masked Autoencoders Are Scalable Vision Learners*, arXiv:2111.06377.

[33] N.-C. Ristea et al., *Self-Distilled Masked Auto-Encoders are Efficient Video Anomaly Detectors*, arXiv:2306.12041.

[34] https://github.com/daleadamjulson/DepthViT_CML.

2026 JINST 21 P10004

The CMS HCAL collaboration

A. Gevorgyan[1], A. Petrosyan[1], A. Tumasyan[1], M. Van De Klundert[2,a], H. Van Haevermaet[2,a], P. Van Mechelen[2,a], A. Van Spilbeeck[2,a], G.A. Alves[3,a], C. Hensel[3,a], W.L. Aldá Júnior[4,a], W. Carvalho[4,a], J. Chinellato[4,a,b], C. De Olivera Martins[4,a], D. Matos Figueiredo[4,a], C. Mora Herrera[4,a], H. Nogima[4,a], W.L. Prado Da Silva[4,a], E.J. Tonelli Manganote[4,a], A. Vilela Pereira[4,a], M. Finger Jr.[5,c], M. Finger[5,c], G. Adamov[6], I. Lomidze[6], T. Toriashvili[6,d], Z. Tsamalaidze[6,d], K. Borras[7,e], A. Campbell[7], F. Engelke[7,e], D. Krücker[7], I. Martens[7], L. Wiens[7,e], M. Csanád[8], S. Lökös[8,f], A. Feherkuti[8], G. Pásztor[8], O. Surányi[8], G.I. Veres[8], V. Hegde[9,a], K. Kothekaar[9,a], S. Pandey[9,a], S. Sharma[9,a], S.B. Beri[10,a], B. Bhawandeep[10,a], R. Chawla[10,a], A. Kalsi[10,a], A. Kaur[10,a], M. Kaur[10,a], G. Walia[10,a], S. Bhattacharya[11,a], S. Ghosh[11,a], S. Nandan[11,a], A. Purohit[11,a], M. Sharan[11,a], S. Banerjee[12,a], S. Bhattacharya[12,a], S. Chatterjee[12,a], P. Das[12,a], M. Guchait[12,a], S. Jain[12,a], S. Kumar[12,a], M. Maity[12,a], G. Majumder[12,a], K. Mazumdar[12,a], M. Patil[12,a], T. Sarkar[12,a], S. Sekmen[13], A. Juodagalvis[14], D. Agyel[15], F. Boran[15], S. Damarseckin[15], Z.S. Demiroglu[15], F. Dolek[15], I. Dumanoglu[15,g], E. Eskut[15], G. Gokbulut[15], Y. Guler[15,h], E. Gurpinar Guler[15,h], C. Isik[15], E.E. Kangal[15], O. Kara[15], A. Kayis Topaksu[15], U. Kiminsu[15], G. Onengut[15], K. Ozdemir[15,i], E. Pinar[15], A. Polatoz[15], A.E. Simsek[15], B. Tali[15,j], U.G. Tok[15], S. Turkcapar[15], E. Uslan[15], I.S. Zorbakir[15], B. Bilin[16,a,k], G. Karapinar[16,a], A. Murat Guler[16,a], K. Ocalan[16,a,l], M. Yalvac[16,a,m], M. Zeyrek[16,a], B. Akgun[17], I.O. Atakisi[17], E. Gülmez[17], M. Kaya[17,n], O. Kaya[17,o], S. Tekten[17,p], E.A. Yetkin[17], T. Yetkin[17,q], K. Cankocak[18,g], S. Sen[18,r], O. Aydilek[19], S. Cerci[19,j], B. Hacisahinoglu[19], I. Hos[19,s], B. Isildak[19,q], B. Kaynak[19], S. Ozkorucuklu[19], O. Potok[19], H. Sert[19], C. Simsek[19], D. Sunar Cerci[19,j], C. Zorbilmez[19], A. Boyarintsev[20], B. Grynyov[20], L. Levchuk[21], V. Popov[21], P. Sorokin[21], H. Flacher[22], S. Abdullin[23], B. Caraway[23], J. Dittmann[23], K. Hatakeyama[23], A.R. Kanuganti[23], B. McMaster[23], M. Saunders[23], J. Wilson[23], P. Bunin[24,c], A. Buccilli[24,t], S.I. Cooper[24], C. Henderson[24,u], C.U. Perez[24], P. Rumerio[24,v], C. West[24], D. Arcaro[25], C. Cosby[25], Z. Demiragli[25], D. Gastler[25], E. Hazen[25], J. Rohlf[25], M. Hadley[26], U. Heintz[26], T. Kwon[26], G. Landsberg[26], K.T. Lau[26], Z. Mao[26], X. Yan[26], D.R. Yu[26,w], J.W. Gary[27], G. Karapostoli[27,x], O.R. Long[27], R. Bhandari[28], R. Heller[28], D. Stuart[28], J.H. Yoo[28], Y. Chen[29], J. Duarte[29], J.M. Lawhorn[29], M. Spiropulu[29], D. Winn[30], A. Apresyan[31], A. Apyan[31,y], S. Banerjee[31,z], F. Chlebana[31], Y. Feng[31], J. Freeman[31], D. Green[31], J. Hirschauer[31], U. Joshi[31], K.H.M. Kwok[31], D. Lincoln[31], S. Los[31], C. Madrid[31], N. Pastika[31], K. Pedro[31], W.J. Spalding[31], S. Tkaczyk[31], S. Linn[32,a], P. Markowitz[32,a], V. Hagopian[33], T. Kolberg[33], G. Martinez[33], O. Viazlo[33], M. Hohlmann[34], R. Kumar Verma[34], D. Noonan[34], F. Yumiceva[34,aa], M. Alhusseini[35], B. Bilki[35,ab], D. Blend[35], K. Dilsiz[35,ac], L. Emediato[35], R.P. Gandrajula[35], M. Herrmann[35], O.K. Köseyan[35], J.-P. Merlo[35], A. Mestvirishvili[35,ad], M. Miller[35], H. Ogul[35,ae], Y. Onel[35], A. Penzo[35], D. Southwick[35], E. Tiras[35,af], J. Wetzel[35], A. Al-bataineh[36,ag], J. Bowen[36,ah], C. Le Mahieu[36], W. McBrayer[36], J. Marquez[36], M. Murray[36], M. Nickel[36], S. Popescu[36], C. Smith[36], Q. Wang[36], K. Kaadze[37], D. Kim[37], Y. Maravin[37], A. Mohammadi[37,z], J. Natoli[37], D. Roy[37], L.K. Saini[37,ai], E. Adams[38], A. Baden[38], O. Baron[38], A. Belloni[38], J.D. Calderon[38,aj], Y.M. Chen[38], C. Coldsmith[38,ak], S.C. Eno[38], C. Ferraioli[38,al], T. Grassi[38], N.J. Hadley[38], A. Hunt[38,am], G.Y. Jeng[38,an], R.G. Kellogg[38], T. Koeth[38], J. Kunkle[38,ao], Y. Lai[38], S. Lascio[38], A.C. Mignerey[38], S. Nabili[38], C. Palmer[38], C. Papageorgakis[38], F. Ricci-Tam[38,ap], M. Seidel[38,aq], Y.H. Shin[38,ar], L. Wang[38], K. Wong[38], Z. Yang[38], Y. Yao[38,as], M. D'Alfonso[39], M. Hu[39], M. Klute[39,at], B. Crossman[40], J. Hiltbrand[40], M. Krohn[40], J. Mans[40], M. Revering[40], N. Strobbe[40], A. Heering[41],

2026 JINST 21 P10004

Y. Musienko[41,c], R. Ruchti[41], M. Wayne[41], A.D. Benaglia[42], W. Chung[42], G. Kopp[42], T. Medvedeva[42], K. Mei[42], C. Tully[42], A. Bodek[43], P. de Barbaro[43], C. Fallon[43], T. Ferbel[43,†], M. Galanti[43], A. Garcia-Bellido[43], A. Khukhunaishvili[43], R. Taus[43], D. Vishnevskiy[43], M. Zielinski[43], B. Chiarito[44], J.P. Chou[44], S.A. Thayil[44], H. Wang[44], N. Akchurin[45], J. Damgov[45], F. De Guio[45,au], S. Kunori[45], K. Lamichhane[45], S.W. Lee[45], T. Mengke[45], S. Muthumuni[45], S. Undleeb[45], I. Volobouev[45], Z. Wang[45], A. Whitbeck[45], G. Cummings[46], S. Goadhouse[46], J. Hakala[46], R. Hirosky[46], V. Alexakhin[47], V. Andreev[47], Yu. Andreev[47], M. Azarkin[47], A. Belyaev[47], S. Bitioukov[47,†], E. Boos[47], O. Bychkova[47], M. Chadeeva[47], V. Chekhovsky[47], R. Chistov[47,av], M. Danilov[47], A. Demianov[47], A. Dermenev[47], M. Dubinin[47,aw], L. Dudko[47], D. Elumakhov[47], V. Epshteyn[47,ax], Y. Ershov[47], A. Ershov[47], V. Gavrilov[47], I. Golutvin[47,†], A. Gribushin[47], A. Kalinin[47,ay], A. Kaminskiy[47], A. Karneyeu[47], L. Khein[47], M. Kirakosyan[47], V. Klyukhin[47], O. Kodolova[47,az], V. Krychkine[47], A. Kurenkov[47], A. Litomin[47], N. Lychkovskaya[47], V. Makarenko[47], P. Mandrik[47], P. Moisenz[47,†], S. Obraztsov[47], A. Oskin[47], P. Parygin[47,ba], V. Petrov[47], S. Petrushanko[47], S. Polikarpov[47,av], E. Popova[47,ay], V. Rusinov[47], R. Ryutin[47], V. Savrin[47], D. Selivanova[47], V. Smirnov[47], A. Snigirev[47], A. Sobol[47], A. Stepennov[47,bb], E. Tarkovskii[47], A. Terkulov[47], D. Tlisov[47,†], I. Tlisova[47], R. Tolochek[47], M. Toms[47,at], A. Toropin[47], S. Troshin[47], A. Volkov[47], A. Zarubin[47], B. Yuldashev[47], A. Zhokin[47], Dale Julson[199,at,*], Eric Reinhardt[200,at], Andrii Krutsylo[201,at], Resham Sohal[200,at], Guillermo Fidalgo[200,at], Sergei Gleyzer[200,at], Emanuele Usai[200,at]

[1] *Yerevan Physics Institute, Yerevan, Armenia*
[2] *Universiteit Antwerpen, Antwerpen, Belgium*
[3] *Centro Brasileiro de Pesquisas Fisicas, Rio de Janeiro, Brazil*
[4] *Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil*
[5] *Charles University, Prague, Czech Republic*
[6] *Georgian Technical University, Tbilisi, Georgia*
[7] *Deutsches Elektronen-Synchrotron, Hamburg, Germany*
[8] *MTA-ELTE Lendület CMS Particle and Nuclear Physics Group, Eötvös Loránd University, Budapest, Hungary*
[9] *Indian Institute of Science Education and Research (IISER), Pune, India*
[10] *Panjab University, Chandigarh, India*
[11] *Saha Institute of Nuclear Physics, Kolkata, India*
[12] *Tata Institute of Fundamental Research-B, Mumbai, India*
[13] *Kyungpook National University, Daegu, Korea*
[14] *Vilnius University, Vilnius, Lithuania*
[15] *Çukurova University, Physics Department, Science and Art Faculty, Adana, Turkey*
[16] *Middle East Technical University, Physics Department, Ankara, Turkey*
[17] *Bogazici University, Istanbul, Turkey*
[18] *Istanbul Technical University, Istanbul, Turkey*
[19] *Istanbul University, Istanbul, Turkey*
[20] *Institute for Scintillation Materials of National Academy of Science of Ukraine, Kharkiv, Ukraine*
[21] *National Science Centre, Kharkiv Institute of Physics and Technology, Kharkiv, Ukraine*
[22] *University of Bristol, Bristol, U.K.*
[23] *Baylor University, Waco, Texas, U.S.A.*
[24] *The University of Alabama, Tuscaloosa, Alabama, U.S.A.*
[25] *Boston University, Boston, Massachusetts, U.S.A.*
[26] *Brown University, Providence, Rhode Island, U.S.A.*
[27] *University of California, Riverside, Riverside, California, U.S.A.*
[28] *University of California, Santa Barbara - Department of Physics, Santa Barbara, California, U.S.A.*
[29] *California Institute of Technology, Pasadena, California, U.S.A.*
[30] *Fairfield University, Fairfield, U.S.A.*

2026 JINST 21 P10004

[31] *Fermi National Accelerator Laboratory, Batavia, Illinois, U.S.A.*
[32] *Florida International University, Miami, U.S.A.*
[33] *Florida State University, Tallahassee, Florida, U.S.A.*
[34] *Florida Institute of Technology, Melbourne, Florida, U.S.A.*
[35] *The University of Iowa, Iowa City, Iowa, U.S.A.*
[36] *The University of Kansas, Lawrence, Kansas, U.S.A.*
[37] *Kansas State University, Manhattan, Kansas, U.S.A.*
[38] *University of Maryland, College Park, Maryland, U.S.A.*
[39] *Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.*
[40] *University of Minnesota, Minneapolis, Minnesota, U.S.A.*
[41] *University of Notre Dame, Notre Dame, Indiana, U.S.A.*
[42] *Princeton University, Princeton, New Jersey, U.S.A.*
[43] *University of Rochester, Rochester, New York, U.S.A.*
[44] *Rutgers, The State University of New Jersey, Piscataway, New Jersey, U.S.A.*
[45] *Texas Tech University, Lubbock, Texas, U.S.A.*
[46] *University of Virginia, Charlottesville, Virginia, U.S.A.*
[47] *Authors affiliated with an institute or an international laboratory covered by a cooperation agreement with CERN*
[199] *Cerium Laboratories, Austin, TX 78741, U.S.A.*
[200] *Department of Physics, University of Alabama, Tuscaloosa, AL 35487, U.S.A.*
[201] *Institute of Computer Science, Polish Academy of Sciences, Warsaw, 02-106, Poland*

[*] *Corresponding author*
[a] *No longer in CMS HCAL Collaboration*
[b] *Also at Universidade Estadual de Campinas, Campinas, Brazil*
[c] *Also at an institute or an international laboratory covered by a cooperation agreement with CERN*
[d] *Also at Tbilisi State University, Tbilisi, Georgia*
[e] *Also at RWTH Aachen University, III. Physikalisches Institut A, Aachen, Germany*
[f] *Also at Karoly Robert Campus, MATE Institute of Technology, Gyongyos, Hungary*
[g] *Also at Near East University, Research Center of Experimental Health Science, Nicosia, Turkey*
[h] *Also at Konya Technical University, Konya, Turkey*
[i] *Also at Piri Reis University, Istanbul, Turkey*
[j] *Also at Adiyaman University, Adiyaman, Turkey*
[k] *Also at CERN, Geneva, Switzerland*
[l] *Also at Necmettin Erbakan University, Konya, Turkey*
[m] *Also at Bozok Universitetesi Rektörlügü, Yozgat, Turkey*
[n] *Also at Marmara University, Istanbul, Turkey*
[o] *Also at Milli Savunma University, Istanbul, Turkey*
[p] *Also at Kafkas University, Kars, Turkey*
[q] *Now at Yildiz Technical University, Istanbul, Turkey*
[r] *Also at Hacettepe University, Ankara, Turkey*
[s] *Also at Istanbul University - Cerrahpasa, Faculty of Engineering, Istanbul, Turkey*
[t] *Now at Bond, San Francisco, U.S.A.*
[u] *Now at University of Cincinnati, Cincinnati, U.S.A.*
[v] *Also at Università di Torino, Torino, Italy*
[w] *Now at University of Nebraska, Lincoln, U.S.A.*
[x] *Now at National Technical University of Athens, Athens, Greece*
[y] *Now at Brandeis University, Waltham, U.S.A.*
[z] *Now at University of Wisconsin-Madison, Madison, U.S.A.*
[aa] *Now at Northrop Grumman, Linthicum Heights, U.S.A.*
[ab] *Also at Beykent University, Istanbul, Turkey*
[ac] *Also at Bingol University, Bingol, Turkey*
[ad] *Also at Georgian Technical University, Tbilisi, Georgia*
[ae] *Also at Sinop University, Sinop, Turkey*

2026 JINST 21 P10004

[af] *Also at Erciyes University, Kayseri, Turkey*
[ag] *Now at Yarmouk University, Irbid, Jordan*
[ah] *Now at Baker University, Baldwin City, U.S.A.*
[ai] *Now at Gallagher Basset, Schaumburg, U.S.A.*
[aj] *Now at NOAA, National Oceanic and Atmospheric Administration, U.S.A.*
[ak] *Now at Northon Grumman Sperry Marine, Huntington Station, U.S.A.*
[al] *Now at Windfall Data, Novato, U.S.A.*
[am] *Now at Accenture Federal Services, Greenbelt, U.S.A.*
[an] *Now at ArcPoint Forensics, Sarasota, U.S.A.*
[ao] *Now at Comprehensive Nuclear-Test-Ban Treaty Organization - CTBTO, Vienna, Austria*
[ap] *Now at Sigmoid Health, Santa Clara, U.S.A.*
[aq] *Now at Riga Technical University, Riga, Latvia*
[ar] *Now at Mars Auto, Inc., Seoul, South Korea*
[as] *Now at University of California, Davis, Davis, U.S.A.*
[at] *Now at Karlsruhe Institute of Technology, Karlsruhe, Germany*
[au] *Now at Università degli Studi di Milano Bicocca, Milano, Italy*
[av] *Also at another institute or international laboratory covered by a cooperation agreement with CERN*
[aw] *Also at California Institute of Technology, Pasadena, California, U.S.A.*
[ax] *Now at Istanbul University, Istanbul, Turkey*
[ay] *Now at University of Maryland, College Park, U.S.A.*
[az] *Also at Yerevan Physics Institute, Yerevan, Armenia*
[ba] *Now at University of Rochester, Rochester, New York, U.S.A.*
[bb] *Now at University of Cyprus, Nicosia, Cyprus*
[†] *Deceased*

2026 JINST 21 P10004