# Online AI-based distributed data reduction for the dual-radiator RICH detector in the ePIC experiment

C. Rossi[i,t,*], B. R. Achari[d,n], N. Agrawal[d,n], M. Alexeev[h,q], C. Alice[h,q], R. Ammendola[j], P. Antonioli[d], C. Baldanza[d], L. Barion[f], A. Biagioni[i], A. Calivà[b,r], M. Capua[a,m], F. Capuani[i], A. Ciardiello[i,t], E. Cisbani[i,k], M. Chiosso[h,q], M. Contalbrigo[f], F. Cossio[h], M. Da Rocha Rolo[h], A. De Caro[b,r], D. De Gruttola[b,r], G. Dellacasa[h], D. Falchieri[d], S. Fazio[a,m], O. Frezza[i], N. Funicello[b,r], M. Garbini[d,l], S. Geminiani[d,n,l], N. Jacazio[h,s], F. Lo Cicero[i], A. Lonardo[i], R. Malaguti[f], F. Mammoliti[c], M. Martinelli[i], M. Mignone[h], C. Mingioni[h], M. Nenni[h], F. Noto[c], L. Occhiuto[a,m], A. Paladino[d], D. Panzieri[h,s], P. Perticaroli[i], S. Plavully[f], L. Polizzi[f], L. Pontisso[i], R. Preghenella[d], R. Ricci[d], L. Rignanese[d], C. Ripoli[b,r], E. Rovati[d,n], N. Rubini[d], M. Ruspa[h,s], A. Saputi[f], F. Simula[i], F. Spizzo [f], U. Tamponi[h], E. Tassi[a,m], G. Torromeo[d], C. Tuvè[e,p], G. M. Urciuoli[i], S. Vallarino[g], P. Vicini[i], R. Wheadon[h]

[a] *INFN Gruppo Collegato di Cosenza, Italy*
[b] *INFN Gruppo Collegato di Salerno, Italy*
[c] *INFN Laboratori Nazionali del Sud, Italy*
[d] *INFN Sezione di Bologna, Italy*
[e] *INFN Sezione di Catania, Italy*
[f] *INFN Sezione di Ferrara, Italy*
[g] *INFN Sezione di Genova, Italy*
[h] *INFN Sezione di Torino, Italy*
[i] *INFN Sezione di Roma 1, Italy*
[j] *INFN Sezione di Roma 2, Italy*
[k] *Istituto Superiore di Sanità, Roma Italy*
[l] *Museo Storico della Fisica e Centro Studi e Ricerche Enrico Fermi, Italy*
[m] *Università della Calabria, Italy*
[n] *Università degli Studi di Bologna, Italy*
[o] *Università degli Studi di Ferrara, Italy*
[p] *Università degli Studi di Catania, Italy*
[q] *Università degli Studi di Torino, Italy*
[r] *Università degli Studi di Salerno, Italy*
[s] *Università del Piemonte Orientale, Italy*
[t] *Università La Sapienza di Roma, Italy*

**Abstract**

The ePIC experiment at EIC integrates a dual-radiator RICH (dRICH) detector for particle identification in the forward region. The detector will use silicon photomultipliers (SiPMs) to detect Cherenkov radiation with single-photon sensitivity over a surface of ∼3 m$^2$. The ∼320k detector channels will be read out by 4,992 Front End Boards (FEBs); each of the 1,248 Readout Boards (RDOs) will collect data from four FEBs and forward them via a VTRx+ optical link to a Data Aggregation and Manipulation Board (DAM). The DAM will be implemented using the FPGA-based FELIX-155 PCIe card, originally designed for the ATLAS experiment. In the ePIC dRICH DAQ, it will collect and merge data from 42 RDOs, transferring them via PCIe to the host memory, from where the host server streams the event fragments to the ePIC data buffering system (Echelon 0) via its 100 GbE interfaces. To mitigate the risk of excessive bandwidth demand caused by the increasing SiPM Dark Count Rate (DCR) – expected to peak at 300 kHz during experiment operation – we designed a real-time data reduction system to reduce the output bandwidth by at least an order of magnitude. This is achieved through a distributed data-flow processing scheme across the DAMs and an additional FELIX-155 card acting as a Trigger Processor (TP) to discard DCR noise-only events online. The architecture employs a distributed Multi-Layer Perceptron (MLP) to discriminate DCR noise-only events, with 30 local sub-network replicas deployed on the DAMs extracting features that are relayed to the TP using a direct low-latency communication channel. The TP implements a global sub-network that, taking the aggregated feature sets from all DAMs as input to complete the inference process, issues a trigger signal to either retain physics events (comprising signal, background, and noise) or discard those consisting solely of DCR noise. The primary implementation challenge is the ∼100MHz acquisition rate, dictated by the ∼10ns electron-ion bunch crossing interval. In the following sections, we describe our technical approach to addressing these strict timing requirements, focusing on the design of the FPGA computing pipelines and high-speed communication channels. Finally, we report on the current implementation status of the system.

*Keywords:* ePIC EIC, Cherenkov detectors, FPGA, Neural Network, readout system

## 1. Introduction

The Electron–Ion Collider (EIC) [1] will be the first collider facility capable of operating with polarized electron, proton, and light-ion beams, enabling the investigation of the spatial and spin structure of protons, neutrons, and light nuclei. Within this framework, and in particular through the electron-Proton/Ion Collider (ePIC) experiment, the EIC also provides significant opportunities for technological advancements in accelerator systems, detector design, as well as in readout and Data Acquisition (DAQ) strategies. In this context, the focus of this proceeding is on the development of a potential data reduction system aimed at addressing bandwidth limitations in the readout of the dual-radiator Ring Imaging Cherenkov (dRICH) detector [2].

## 2. dRICH and Data Acquisition system (DAQ) overview

The dRICH detector is under development to provide charged-hadron identification in the hadronic end-cap region of the ePIC experiment at the EIC. To achieve the required performance over a broad momentum range, extending from a few GeV/$c$ up to 50 GeV/$c$, the detector leverages on Cherenkov radiation generated in two distinct radiators, namely aerogel and gas. The dRICH is equipped by six sectors of Silicon Photomultipliers (SiPMs) [3] featuring single-photon sensitivity, covering a total active area of $\sim$3 m$^2$ and corresponding to $\sim$320k readout channels continuously streaming data to the dRICH Data Acquisition (DAQ) system.
The readout architecture is based on a hierarchical aggregation scheme in which data from 4,992 ALCOR-based [4] Front-End Boards (FEBs) are collected by 1,248 FPGA-based Readout Boards (RDOs). Each RDO is in turn connected to higher-level FPGA-based processing units, referred to as Data Aggregation and Manipulation Boards (DAM), which provide increased computational capability for data handling and reduction.
Utilizing the FELIX-155 –the latest iteration of the ATLAS FELIX platform[5]– for the DAMS, the system leverages the AMD/Xilinx Versal Premium FPGA architecture. With its support for PCIe Gen4/5 and high-density optical links, the FLX-155 enables the aggregation of distributed front-end data –the card sports 48 optical ports with support for data rates up to 25 Gbps– into a unified, high-speed stream for real-time processing and PCIe DMA transfer towards the host memory.

## 3. Distributed Neural Network model for Online Data Reduction

Silicon Photomultiplier (SiPM) sensors have been chosen for their high photon-detection efficiency, insensitivity to strong magnetic fields (of the order of 1T in the dRICH region), and excellent timing performance. Nevertheless, SiPMs exhibit limited radiation tolerance, leading to a significant increase in the Dark Count Rate (DCR) over the experiment's lifetime as a consequence of radiation damage. In particular, the DCR is expected to rise from initial values of approximately 3kHz up to about 300kHz per channel.
To mitigate this effect and maintain the DCR below an acceptable level of 300 kHz per channel, the collaboration has identified several operational strategies, including the recovery of radiation damage through high-temperature in-situ annealing cycles [6]. Despite these mitigation operations, noise contributions induced by elevated DCR levels (especially at rates closer to 300 kHz) are expected to generate a high amount of uncorrelated dRICH SiPM hits, which may lead to a saturation of the available detector bandwidth (estimated to be of the order of 30 channels at 100GbE, corresponding to an aggregate throughput of $\sim$3Tbps).
In order to limit the redundant data throughput and preserve the overall DAQ performance, a reduction of the dRICH data throughput is therefore required. Motivated by this constraint, an online AI-based data reduction system has been proposed to discriminate in real-time between Noise-Only and physics-related Signal+Background+Noise events.

### *3.1. Dataset generation and training*

Leveraging the EIC software framework developed for Monte Carlo event generation and reconstruction, namely `EICrecon`, multiple datasets have been produced for the NN model training and validation. These datasets are designed to distinguish between two main event categories:

- **Signal+Background+Noise** events, obtained by initially combining simulated physics signals, such as Deep Inelastic Scattering (DIS), with physics background contributions, including electron- and proton-beam–gas interactions. Subsequently, DCR noise is injected into the reconstructed events to emulate realistic detector conditions;
- **Noise-Only** events, consisting exclusively of dRICH SiPM hitmaps populated by uncorrelated DCR-induced hits.

The datasets cover several baseline DCR configurations to assess the performance and robustness of the proposed AI-based system. This allows for stress-testing the architecture under varying noise conditions in preparation for its prospective real-time online deployment.

### *3.2. Model architecture integration in the DAQ system*

The architecture of the proposed online data reduction system, illustrated in Fig. 1, is modeled after the dRICH DAQ structure. The design is implemented as a distributed, multi-FPGA application comprising 30 parallel Multi-Layer Perceptron (MLP) instances, each characterized by independent weights and deployed across the DAM FPGAs. In addition, six Sector MLPs and a final Aggregation network are implemented on a dedicated Trigger Processor (TP) FPGA.

Each DAM FPGA processes input data coming from 42 PDUs, corresponding to a specific sub-sector of a given dRICH

*Corresponding author - *Email address*: cristian.rossi@roma1.infn.it (Cristian Rossi)

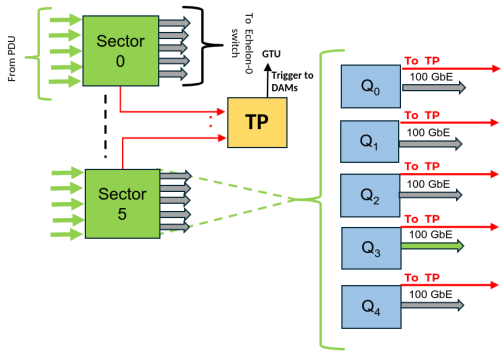


Figure 1: Online data reduction system distributed on the dRICH readout DAMs. Green arrows indicate the default RDO stream of data to the ePIC DAQ switch; Red arrows indicated the stream of features extracted from each MLP sub-network; black arrows indicate the connection with the Global Timing Unit (GTU) of the ePIC experiment.

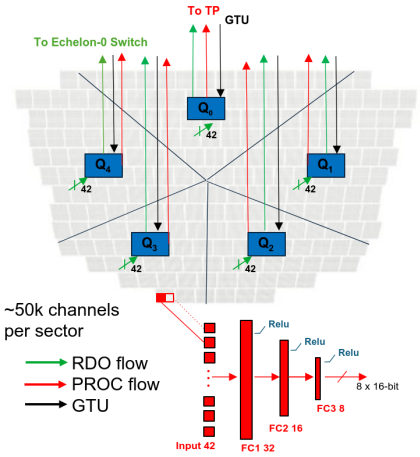


Figure 2: A close up on a single sector of the dRICH: each DAM manages 42 streams of data coming from the sub-sector PDUs and performs the NN computation. The features extracted from each MLP sub-network are then streamed to the TP board via dedicated channels (in red).

sector, as shown in Fig. 2. Within each of the 30 independent MLP processing pipelines, eight 16-bit features are extracted and transmitted from the DAMs to the TP through the APEIRON-based [7] Communication IP. The hardware block diagram of the TP is reported in Fig. 4. At this stage, the local information provided by the five DAM MLPs of a single sector is recombined and used as input to the corresponding Sector MLP, which produces four 16-bit features.

The outputs of the six Sector MLPs are subsequently aggregated and processed by the final Aggregation NN, where the global classification is performed. The resulting inference drives a Finite State Machine (FSM) responsible for issuing the trigger signal via the DAQ Global Timing Unit to the DAMs, where the dRICH event fragments are temporarily buffered. Based on the classification outcome provided by the AI system, a decision is taken either to forward the event fragments to the ePIC Echelon 0 data buffering system, in the case of physics (including background) events accompanied by DCR noise, or to discard them when the event is identified as Noise-Only.

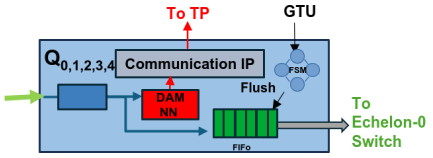


Figure 3: DAM hardware blocks scheme.

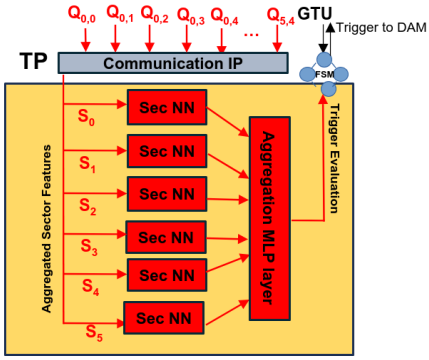


Figure 4: TP hardware blocks scheme.

### *3.3. Model training, validation and quantization*

The complete model was trained as a noise classifier, leveraging the TensorFlow famework [8]. A balanced dataset of 200k events (90% training set, 8% testing set, and 2% validation set) was employed, varying the DCR parameter from 25 kHz to 300 kHz per channel to assess the scaling performance of the system. Classification performance is expressed in terms of True Positive Rate (TPR) and True Negative Rate (TNR) [9], representing the percentage of correctly classified true-labeled Noise-Only events and Signal+Background+Noise events, respectively.

As the final design targets the implementation of multiple MLPs on FPGA hardware, a model quantization step was applied using the QKeras library [10]. This procedure aims at reducing the bit-width required for the representation of weights and biases in the ML layers, thereby optimizing resource utilization for hardware deployment.

The performance scaling obtained in both the floating-point and quantized configurations is represented in Fig. 5. As expected, a degradation in classification performance is observed with increasing DCR values. Nevertheless, for the TensorFlow floating-point model, both TPR and TNR remain above 82% even under the most adverse noise conditions. In contrast, for the quantized model, post-quantization training could not be successfully performed for DCR values exceeding 150 kHz due to limitations encountered in the QKeras framework, leading to a noticeable reduction in performance with respect to the non-quantized case.

## 4. Stripped-down TP HW design

To validate the TP NN model (6 Sector MLPs + Aggregation Layer) implementation and to assess its performance, we developed a stripped-down HW system for deployment on a

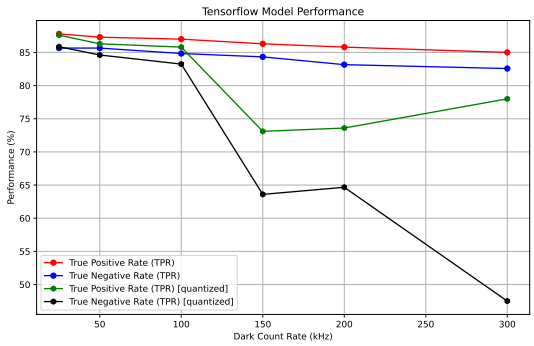


Figure 5: Noise-classifier NN model performance scaling with increasing DCR values.

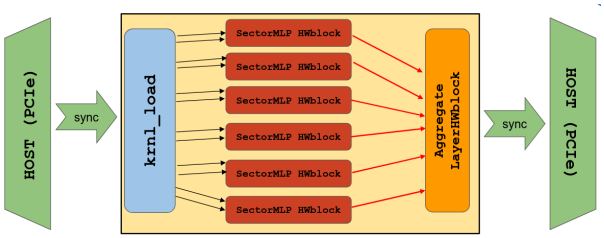


Figure 6: TP NN model stripped-down HW architecture scheme

Xilinx Alveo U280 card. As illustrated in Fig. 6, the design comprises a series of High Level Synthesis (HLS) [11] blocks:

- *krnl_load*, in charge of loading event data from the host memory via PCIe bus and of streaming them to the NN processing blocks;
- *SectorMLP_HWblock*, in which incoming data are concatenated to form the input for the proper NN computation. This is performed leveraging the *hls4ml* [12] Python library to convert Sector MLP layers defined in QKeras into HLS C++ code;
- *AggregateLayer_HWblock*, receiving feature streams from all Sector MLPs and performing the final classification via a single neural network layer implemented with hls4ml. Once computed, the prediction is offloaded to the host memory via the PCIe interface.

### *4.1. Results*

We synthesized the HW system firmware using Xilinx Vitis HLS tool [11]. Considering the resources utilization of the design, shown in Fig. 7, we obtained a high BRAM utilization due to the allocation six different sets of weights and biases for the six Sector MLPs implemented in the TP model.

We flashed the TP firmware on the Xilinx Alveo U280 board in our lab and run several tests to stress out the HW design capabilities. Firstly, loading events data from the validation datasets (Section. 3.3) on the 8GB large Alveo HBM memory, we have evaluated that the HW model performance in terms of TPR and TNR are compatible with the ones obtained for the quantized

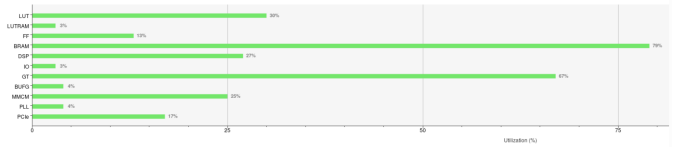


Figure 7: Xilix Vitis analyzer report of the stripped-down TP HW design. The percentage reported have to be considered as "Post-Synthesis" values

Qkeras Model. Then, using a reduced set of event data preloaded on the much smaller Alveo BRAM memory, we have tested the system to measure its processing throughput performance.

The challenge in our work is to develop a FPGA pipelined design capable to cope with the timing requirements of the ePIC experiment, and so with a bunch crossing rate of ∼100 MHz. Using the BRAM memory, we are able to avoid the bottleneck due to the Host-HBM communication, obtaining timing constraints reasonably similar to the ones expected in the "real-environment" in which the final HW design will be deployed. With the current setup, we were able to reach a throughput of ∼11 MHz, which is still far from the experiment requirements. Optimization of the pipeline initiation interval (II) and further parallelization are planned to close this gap.

## 5. Conclusions and future work

A simplified implementation of the TP NN model has been developed and evaluated. The system performance has been assessed in terms of standard ML classification metrics (TPR and TNR), as well as hardware-oriented metrics such as resource utilization and processing throughput.

The results confirm the viability of the proposed architecture while identifying key areas for further optimization. Current developments focus on improving classification TNR by reducing physics-related Signal+Background+Noise events misclassification, enhancing post-quantization performance at noise rates above 100 kHz, and increasing the throughput of the full processing pipeline.

To address current quantization challenges, we are investigating state-of-the-art libraries such as HGQ [13], which is integrated in the latest version of hls4ml framework. By leveraging these enhanced capabilities and updating the hls4ml toolchain, we expect to further optimize the hardware performance and resolve existing post-quantization bottlenecks.

In parallel with the improvements to the TP NN model, its implementation on the Xilinx Versal device targeting the DAM FELIX system is currently in progress.

Furthermore, a simplified distributed MLP architecture spanning two FPGAs and including all major HW architectural components of five DAM MLP NNs –modeing the readout behaviour of a single dRICH sector– and the complete TP model is under development, with particular emphasis on validating the low-latency DAM-to-TP communication protocol.

## References


[1] Abdul Khalek, et al., Science requirements and detector concepts for the electron-ion collider, Nuclear Physics A 1026 (2022) 122447. doi:10.1016/j.nuclphysa.2022.122447.
URL http://dx.doi.org/10.1016/j.nuclphysa.2022.122447

[2] Vallarino, et al., Prototype of a dual-radiator rich detector for the electron–ion collider, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 1058 (2024) 168834. doi:https://doi.org/10.1016/j.nima.2023.168834.
URL https://www.sciencedirect.com/science/article/pii/S0168900223008252

[3] N. Rubini, et al., The SiPM readout plane for the ePIC-dRICH detector at the EIC: Overview and beam test results, Nucl. Instrum. Meth. A 1082 (2026) 170890. doi:10.1016/j.nima.2025.170890.

[4] F. Cossio, et al., ALCOR: A mixed-signal ASIC for the dRICH detector of the ePIC experiment at the EIC, Nucl. Instrum. Meth. A 1069 (2024) 169817. doi:10.1016/j.nima.2024.169817.

[5] Paramonov, Alexander, Felix: the detector interface for the atlas experiment at cern, EPJ Web Conf. 251 (2021) 04006. doi:10.1051/epjconf/202125104006.
URL https://doi.org/10.1051/epjconf/202125104006

[6] R. Preghenella, et al., SiPM photosensors for the ePIC dual-radiator RICH detector at the EIC, PoS EPS-HEP2023 (2024) 515. doi:10.22323/1.449.0515.

[7] Ammendola, et al., Apeiron: A framework for high level programming of dataflow applications on multi-fpga systems, EPJ Web of Conf. 295 (2024) 11002. doi:10.1051/epjconf/202429511002.
URL https://doi.org/10.1051/epjconf/202429511002

[8] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: A system for large-scale machine learning, Proceedings of the 12th USENIX conference on operating systems design and implementation (2016) 265–283.

[9] T. Fawcett, An introduction to roc analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[10] C. N. Coelho, A. Kuusela, S. Li, H. Zhuang, J. Ngadiuba, T. K. Aarrestad, V. Loncar, M. Pierini, A. A. Pol, S. Summers, Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors, Nature Machine Intelligence 3 (8) (2021) 675–686. doi:10.1038/s42256-021-00356-5.
URL http://dx.doi.org/10.1038/s42256-021-00356-5

[11] Advanced Micro Devices, Inc., Xilinx Vitis High-Level Synthesis, https://www.xilinx.com/products/design-tools/vitis/vitis-hls.html, accessed: 2024-XX-XX (2023).

[12] J. Duarte, et al., Fast inference of deep neural networks in FPGAs for particle physics, JINST 13 (07) (2018) P07027. arXiv:1804.06913, doi:10.1088/1748-0221/13/07/P07027.

[13] C. Sun, T. K. Arrestad, V. Loncar, J. Ngadiuba, M. Spiropulu, Gradient-based Automatic Mixed Precision Quantization for Neural Networks On-Chip (5 2024). arXiv:2405.00645, doi:10.7907/hq8jd-rhg30.