Level-3 Trigger for CLAS12 with Artificial Intelligence

Richard Tyson[1,*], *Gagik* Gavalian[2], *Bryan* McKinnon[1], *David* Ireland[1], and the CLAS Collaboration

[1]SUPA, School of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, United Kingdom
[2]Jefferson Lab, Newport News, VA, USA

Abstract. Fast, efficient and accurate triggers are a critical requirement for modern high energy physics experiments given the increasingly large quantities of data that they produce. The CEBAF Large Acceptance Spectrometer (CLAS12) employs a highly efficient electron trigger to filter the amount of data recorded by requiring at least one electron candidate in each event, at the cost of a low purity in electron identification. However, machine learning algorithms are increasingly employed for classification tasks such as particle identification due to their high accuracy and fast processing times. In this proceeding we present recently published work that showed how a convolutional neural network could be deployed as a Level 3 electron trigger at CLAS12. We demonstrate that this AI trigger would achieve a significant data reduction compared to the conventional CLAS12 electron trigger, whilst preserving a 99.5% electron identification efficiency, at nominal CLAS12 beam currents.

1 Introduction

The Continuous Electron Beam Accelerator Facility (CEBAF) produces and delivers an electron beam to the four experimental halls of the Thomas Jefferson National Accelerator Facility (JLab), including the CLAS12 (CEBAF Large Acceptance Spectrometer @ 12 GeV) detector [1] located in Hall B. Beam energies up to 11 GeV are delivered to Hall B. CLAS12 is a large acceptance detector with above 10^5 readout channels [2]. A diagram of CLAS12 is shown in Figure 1. CLAS12 has a central detector and a forward detector which is further segmented into six sectors. Electrons are only identified in the forward detector as the central detector does not have the capabilities to do so. In 2018, data rates of 500 MB/s with a live time of 95% were achieved at CLAS12 and a total of 2 pB of data was accumulated [2]. As such, a fast and efficient trigger is necessary to reduce these large data rates so as to save on costs of storage and post processing times.

Most reactions studied with CLAS12 require the detection of an electron scattered from the interaction of the electron beam with a fixed target. Therefore, an electron trigger is used to flag events relevant to the CLAS12 experimental program. The electron confusion matrix is shown in Figure 2. It defines True (False) Positives as events with (without) an electron selected by the trigger and True (False) Negatives as events without (with) an electron rejected

*e-mail: tyson@jlab.org

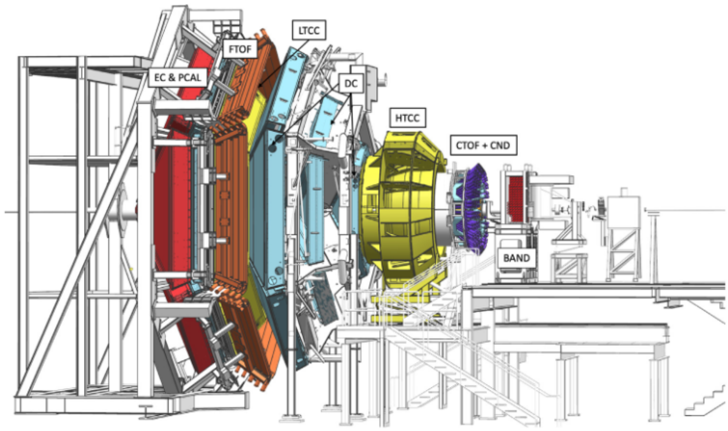

Figure 1. A diagram of the CLAS12 detector highlighting relevant subsystems of the forward and central detectors [1]. Three of the forward detector's six sectors are visible.

by the trigger. The performance of the trigger can be evaluated by estimating the trigger efficiency (E), purity (P) and overall accuracy (A) which are calculated as:

$$
\begin{aligned}
E &= \frac{TP}{TP+FN} \\
P &= \frac{TP}{TP+FP} \\
A &= \frac{TP+TN}{TP+FP+FN+TN}
\end{aligned}
\tag{1}
$$

Confusion Matrix	Electron in Sector	No Electron in Sector
Selected by Trigger	True Positive (TP)	False Positive (FP)
Rejected by Trigger	False Negative (FN)	True Negative (TN)

Figure 2. The electron trigger confusion matrix [3].

The conventional CLAS12 electron trigger has a high efficiency measured above 99.5% [4] but in 2018 had a purity below 40%. Given that machine learning algorithms are known to be well suited to classification tasks such as electron identification, the decision was then made to test convolutional neural networks [3, 5, 6] as a way to improve the CLAS12 trigger purity. This machine learning based trigger is referred to as the AI trigger.

2 Training

Electrons at CLAS12 are identified using the coincidence of a track in the drift chambers (DC) [7], a signal in the High Threshold Cherenkov Counters (HTCC) [8] and a high energy deposition in the Electromagnetic Calorimeters (ECAL) [9]. The conventional offline electron identification routine is used to create the training dataset for the AI trigger. The positive sample of this dataset is composed of events with at least one electron in one the six sectors of the forward detector. The negative sample is composed of all other sectors including those from data taken with different triggers. The AI trigger is called on a sector by sector basis [3]. To decrease the amount of data required for the AI trigger, the choice was made to only use data from the DC and the ECAL. This allows for a higher prediction rate for the AI trigger as it needs to process less data.

Figure 3 shows a diagram of the DC system of CLAS12. It is composed of six superlayers grouped into three regions [7]. A schematic of the wire layout for one superlayer is shown in Figure 3. Each superlayer is composed of six layers of sense wires, with 112 wires per layer [7]. For the purpose of the AI trigger, the information from the DC is then parsed into a six by 112 array filled with 1/6 per hit per layer in a superlayer. An example of this is shown in Figure 5.

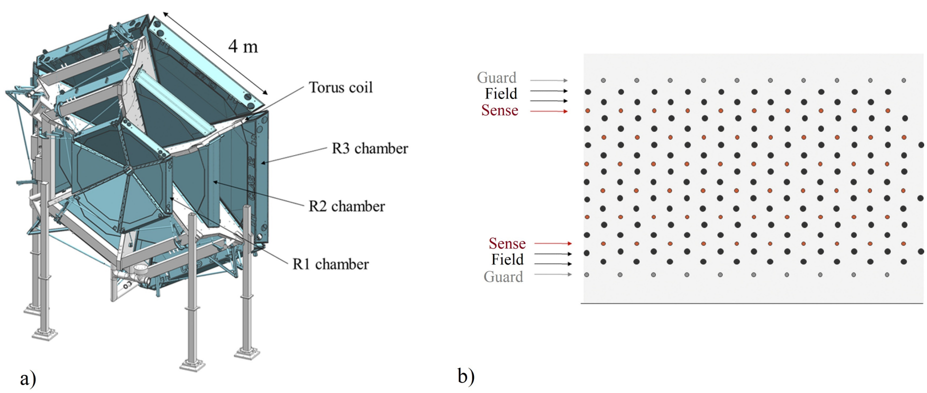

Figure 3. a) Diagram showing the three regions (R1, R2, R3) of the DC, each region containing two superlayers. The forward detector's six sectors are also visible [7]. b) Schematic of the wire layout for one superlayer, with each superlayer containing 6 layers of sense wires (red) [7]. The view is a cut perpendicular to the wire direction.

The ECAL is segmented into three calorimeters, the pre-shower calorimeter (PCAL), and the inner and outer electromagnetic calorimeters (ECIN and ECOUT respectively). Each of the three calorimeters is composed of three views, U, V and W, as shown in Figure 4. A cluster in the ECAL is defined at the intersection of hits in each view. The PCAL has 68 strips in the U view and 62 strips in the V and W views, with the ECIN and ECOUT having 36 strips in all three views [9]. For the purpose of the AI trigger, the information from the ECAL is then parsed into a six by 72 array filled with the energy deposited in each strip, divided by three as the maximal energy deposition in a strip is about three GeV. An example of this array is shown in Figure 5.

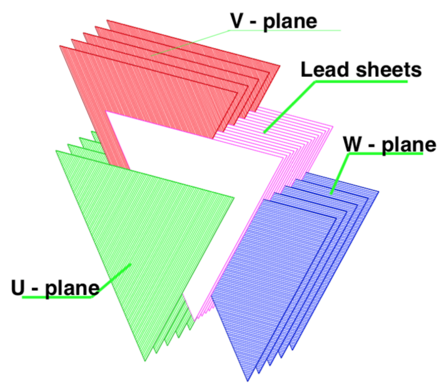

Figure 4. Schematic representation of the calorimeters' three views [9].

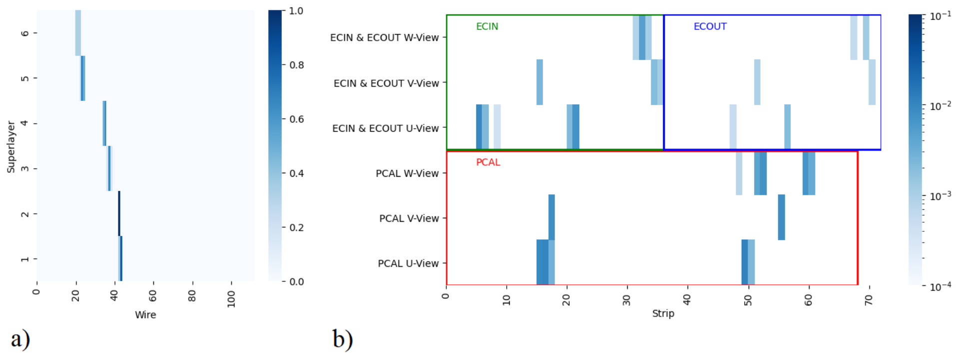

Figure 5. a) An example of a DC image containing an electron track in a given sector. The colour axis represents the number of layers with the same wire number, divided by six for normalisation [3]. b) An example of an ECAL image containing all ECAL hits in a given sector for a single event [3]. The colour axis represents the energy deposited in each strip divided by three.

The CLAS12 detector has been taking data at various beam currents (from 40nA to 55nA). As a higher beam current leads to higher detector occupancies and additional noise, a lower purity for the conventional CLAS12 trigger is seen with increasing beam current. A background merging method has been developed to mimic data taken at various beam currents, where the background is derived using random trigger data. The background in random trigger data taken at a given beam current is added to data as shown in Figure 5 to mimic raw data taken at that beam current. The background can also be added multiple times to mimic data taken at a larger beam current. The process of background merging for the DC is exemplified in Figure 6.

In Ref. [10] it was shown how a convolutional auto-encoder can be used to remove noise from the DC. This is exemplified in Figure 6. The auto-encoder is trained to remove noise by using the data with background merging as input and the data without background merging as output. A similar approach was investigated here where a denoiser is added to a trigger trained on denoised data. The comparison between the performance of the trigger with and without the denoiser is then informative in testing the robustness of the trigger to noise.

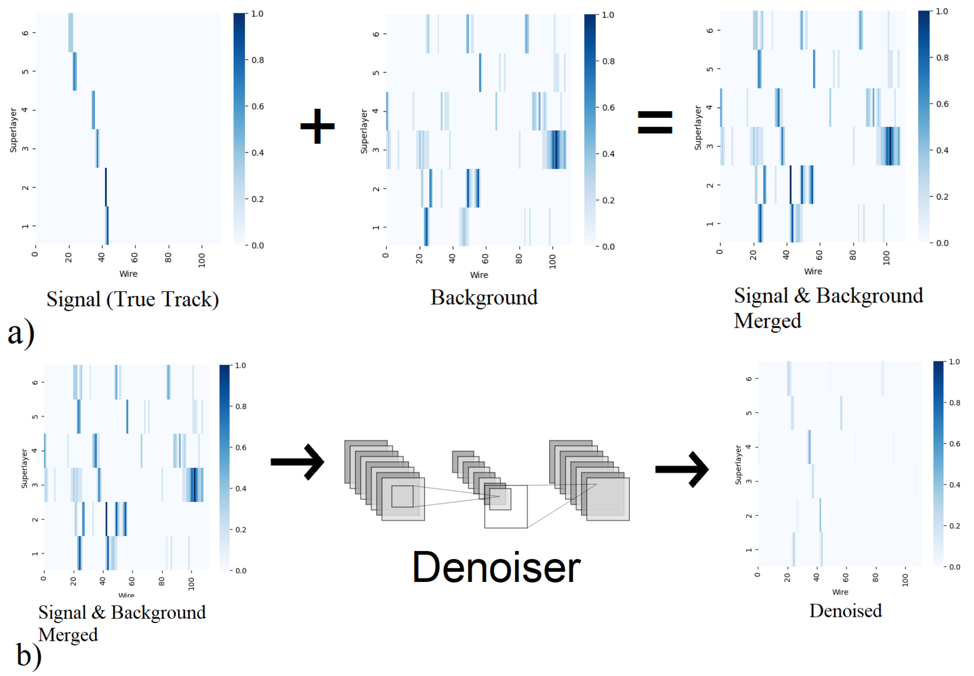

Figure 6. a) The top row demonstrates the process of adding spurious noise to a clean track to synthetically create raw hits as measured by the DC [3]. b) The bottom row demonstrates how the de-noiser can be applied to the raw hits to return cleaner denoised data [3].

The specific architecture used for the AI trigger sees the DC and ECAL arrays passed to separate sets of convolutional layers, with the output of these concatenated and passed to a neural network for classification. The output of the neural network forms the response of the trigger and ranges from 0 to 1. By applying a lower threshold to the response, the electron identification process is then reduced to a cut on the response. Figure 7 shows the accuracy, purity, efficiency and the product of the purity and efficiency as a function of the threshold on the trigger response, when the trigger is used without the denoiser. Tables 1 gives a few examples of these metrics at specific cuts on the response. The aim of the AI trigger is to maximise the purity for an efficiency above 99.5%. The threshold on the response is then placed at a low value of the response where the efficiency is above 99.5%. As shown, the trigger is able to achieve a purity of 91.2% for an efficiency of 99.5%.

Threshold on Response	Purity	Efficiency	Accuracy
0.01	83.0 %	99.9 %	89.7
0.04	88.8 %	99.7 %	93.6
0.08	91.2 %	99.5 %	95.0

Table 1. The purity, efficiency and accuracy of the AI trigger at different cuts on the response [3].

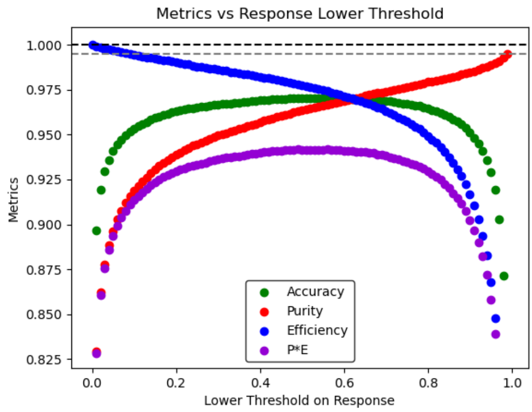

Figure 7. The accuracy, efficiency and purity and the product of the purity and efficiency of the AI trigger as a function of the threshold cut on the CNN response [3]. Here the trigger is used without the denoiser. The black dashed line is set at 1.0, whilst the grey dashed line is set at 0.995.

3 Performance and Deployment

The AI trigger can therefore reproduce the efficiency of 99.5% of the conventional electron trigger. The purity of the AI trigger is then compared to that of the conventional CLAS12 trigger in Figure 8. As shown, the AI trigger can greatly improve on the conventional trigger. Not shown here for brevity, the efficiency of the AI trigger is also stable around 99.5% as a function of beam current [3]. The purity is also shown for the AI trigger with and without the denoiser. The purity of the AI trigger with the denoiser is only marginally higher than that of the AI trigger without the denoiser. Combined with the fact that the purity of the AI trigger is stable with beam current, this is a very good indication that the AI trigger is robust to noise and a higher occupancy in the detector. The data reduction (DR) achieved by the AI trigger relative to the traditional CLAS12 trigger can then be calculated as [3]:

$$DR = E_{AI}|(P_{AI} - P_{CLAS12})| \tag{2}$$

where E_{AI} is the efficiency of the AI trigger and P_{AI} and P_{CLAS12} are the purity of the AI and CLAS12 triggers respectively. A high data reduction allows to decrease the data rate out of CLAS12, which then allows to save on costs of storage and post processing times. In 2018 CLAS12 took data with beam currents of 40nA, 45nA, 50nA and 55nA. As shown in Figure 8, a data reduction by more than 60% is achieved with the AI trigger for all beam currents at which CLAS12 took data in 2018. CLAS12 is also aiming to increase the beam current at which it takes data, where the AI trigger enables an even higher data reduction while maintaining a high efficiency and purity.

The trigger was trained in tensorflow [11] in python but will deployed using the Deeplearning4j [12] software package in Java. The code used to deploy the trigger has already been written and is available following Ref. [13]. Figure 9 shows the prediction rates for the AI trigger with and without the denoiser and for the denoiser itself. The prediction rate is measured as a function of the batch size, for events grouped into batches at prediction time. The

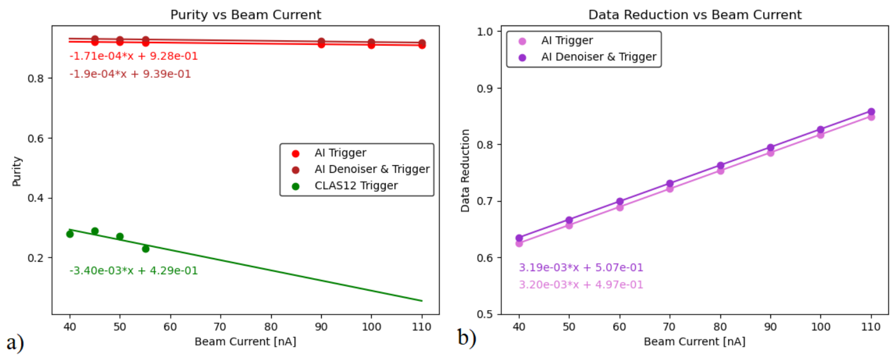

Figure 8. a) The purity of the AI (red) and conventional CLAS12 (green) triggers as a function of beam current [3]. b) The data reduction achieved by the AI trigger relative to the CLAS12 trigger [3]. The purity and data reduction is shown for the AI trigger with and without the denoiser. For both AI triggers the cut on the response was placed at 0.08 in both plots.

prediction rates were measured in python using a GPU with an Nvidia GeForce RTX 2080 Ti graphics card, 11 GB GDDR6 RAM and 4352 CUDA cores. In 2018, the typical trigger rate of CLAS12 was of about 15kHz [2]. Furthermore, as the CLAS12 forward detector has six sectors, the AI trigger needs to make six predictions per event. Thus, as shown in Figure 9, only the AI trigger without the denoiser is able to keep up with the rate of CLAS12 data taking and will be deployed on a GPU. An increase in beam current would lead to an increase in data rate. Another suggested deployment would see the AI trigger only being called on events that pass the conventional trigger, in which case the AI trigger would not need to make a prediction for each sector and would be more than capable of keeping up with the rate of data taking. Both suggested implementations of the AI trigger would see the data being streamed to one or more GPUs during online data taking before the data is written to tape.

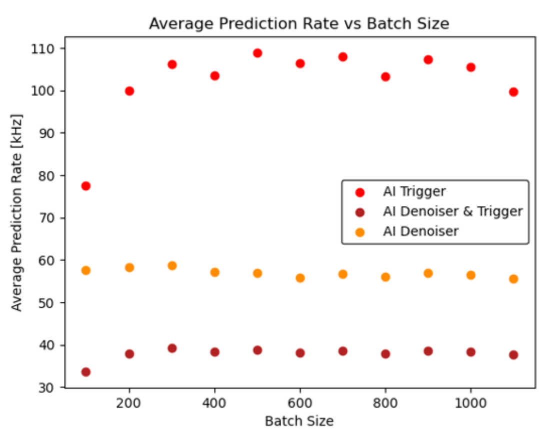

Figure 9. The prediction rate of the AI trigger, the denoiser and the trigger and denoiser as a function of the batch size, for predictions grouped into batches. Quoted here are the average rates per sector from 100 batches.

4 Conclusion

Triggers with a high efficiency and purity are critical for high energy physics experiments that tend to produce large quantities of data. In this proceeding it was shown how an AI trigger employing convolutional neural networks can be used to greatly improve on the purity of the conventional electron trigger at CLAS12. This AI trigger is able to keep up with the rate of data taking, and is especially powerful at beam currents higher than what was previously used at CLAS12. The AI trigger would therefore enable to save on storage costs and post processing times whilst allowing to produce larger quantities of the data relevant to the CLAS12 experimental program.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under contract DE-AC05-06OR23177 and by the U.K. Science and Technology Facilities Council under grants ST/P004458/1 and ST/V00106X/1. We would also like to thank the CLAS Collaboration for providing the data used for this body of work.

References

[1] V. Burkert *et al.*, The CLAS12 Spectrometer at Jefferson Laboratory, *Nucl. Inst. Methods A* **959** 163419 (2020).

[2] V. Ziegler *et al.*, The CLAS12 software framework and event reconstruction, *Nuclear Instruments and Methods in Physics Research. Section A* **959** 163472 (2020).

[3] R. Tyson, G. Gavalian, D.G. Ireland, B. McKinnon, Deep learning level-3 electron trigger for CLAS12, *Computer Physics Communications* **290**, 108783 (2023).

[4] B. Raydo *et al.*, The CLAS12 Trigger System, *Nucl. Instrum. Methods A* **960** 163529 (2020).

[5] S. Albawi, T. A. Mohammed and S. Al-Zawi, Understanding of a convolutional neural network, *2017 International Conference on Engineering and Technology (ICET)*, doi: 10.1109/ICEngTechnol.2017.8308186, (2017).

[6] K. O'Shea and R. Nash, An Introduction to Convolutional Neural Networks, *ArXiv e-prints* arXiv:1511.08458 (2015).

[7] M.D. Mestayer *et al.*, The CLAS12 drift chamber system, *Nucl. Instrum. Methods A* **959** 163518 (2020).

[8] Y.G. Sharabian *et al.*, The CLAS12 high threshold Cherenkov counter, *Nucl. Instrum. Methods A* **968** 163824 (2020).

[9] G. Asryan *et al.*, The CLAS12 forward electromagnetic calorimeter, *Nucl. Instrum. Methods A* **959** 163425 (2020).

[10] P. Thomadakis *et al.*, De-noising drift chambers in CLAS12 using convolutional auto encoders, *Computer Physics Communications* **271** 108201 (2022).

[11] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, *ArXiv e-prints* arXiv:1603.04467v2 (2016). Software available from tensorflow.org.

[12] Eclipse Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org

[13] R. Tyson and G. Gavalian, CLAS12AIElectronTrigger, https://github.com/rtysonCLAS12/CLAS12AIElectronTrigger