PROCEEDINGS OF SCIENCE

Development and firmware implementation of a Machine Learning based hadronic τ lepton Level-1 Trigger algorithm in CMS for the HL-LHC

Jona Motta* on behalf of the CMS Collaboration

Laboratoire Leprince-Ringuet, CNRS/IN2P3, Ecole Polytechnique, Institut Polytechnique de Paris, Route de Saclay, Palaiseau, France

E-mail: jona.motta@cern.ch

The High-Luminosity LHC (HL-LHC) will open an unprecedented window on the weak-scale nature of the universe, providing high-precision measurements of the standard model as well as searches for new physics beyond it. The CMS Collaboration is planning to replace entirely its trigger and data acquisition systems to match this ambitious physics program. Efficiently collecting datasets in Phase-2 will be a challenging task, given the harsh environment of 200 simultaneous proton-proton interactions per HL-LHC bunch crossing. The already challenging implementation of an efficient τ lepton trigger will become, in such conditions, an even more crucial and harder task; especially interesting will be the case of hadronically decaying τ. To this end, the highly upgraded capabilities of the Phase 2 Level-1 triggering system can be exploited to design new complex machine learning based algorithms that are not yet implementable in the current Phase-1 system. Moreover, the foreseen high-granularity endcap calorimeter and the astonishing amount of information it will provide play a key role in the design of novel τ lepton triggering methods. In these proceedings, the development of a Level-1 trigger algorithm, with consistent barrel and endcap treatment, for hadronically decaying τ based on the calorimetric information from the ECAL, HCAL, and HGCAL detectors will be presented: the TauMinator. A completely new and innovative design for a Level-1 trigger algorithm based on convolutional neural networks will be shown alongside its preliminary FPGA firmware implementation. The Level-1 trigger latency and resource availability constraints will also be discussed, and their role in the algorithm design will be highlighted.

The European Physical Society Conference on High Energy Physics
21-25 August 2023
Hamburg, Germany

*Speaker

PoS(EPS-HEP2023)590

1. Introduction

The High-Luminosity LHC (HL-LHC) is scheduled to start in 2029, and it will constitute the Phase-2 of the LHC operations. It is designed to operate at a centre-of-mass energy of 14 TeV while delivering an instantaneous luminosity of $5 - 7.5 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. These conditions correspond to a number of simultaneous collisions (pileup, PU) per bunch crossing (BX) of $O(200)$.

Efficiently collecting datasets to be used in the HL-LHC physics program will be challenging. Therefore, the CMS Collaboration [1] is redesigning its hardware-implemented Level-1 Trigger (L1T) [2]. The Phase-2 L1T will exploit state-of-the-art Field Programmable Grid Arrays (FPGAs) and link technologies, providing a high-performance, low-latency, and high-throughput system in which algorithms based on machine learning techniques will be widely employed [3].

These proceedings are structured as follows. Section 2 presents the innovative TauMinator algorithm [4], its design and firmware implementation. Section 3 discusses the physics performance attained by the algorithm. Section 4 closes the discussion with conclusions and outlook.

2. The TauMinator algorithm

The η coverage of the CMS calorimeters at the L1T is organised in Trigger Towers (TTs), offering a coarse view of the calorimeters. Each TT is identified by its position in discrete Cartesian coordinates $(i\eta, i\phi)$ and carries energy deposit (E_T). In the endcap, the High Granularity Calorimeter (HGCAL) [5] produces a second type of input to the L1T, the $\mathrm{CL}^{3\mathrm{D}}$, which are 3-dimensional clusters following the particle shower evolution characterized by shower shape variables.

The calorimetric inputs are exploited in the TauMinator algorithm, which is designed based on five guidelines: boost the Run-2 and Run-3 approach to τ_h shape recognition; avoid the need for an independent isolation step between τ_h and QCD-induced jets; calibrate the τ_h candidate profiting of energy deposits correlations; exploit the highly granular information of the $\mathrm{CL}^{3\mathrm{D}}$s; maximally profit of the L1T FPGAs computing resources.

The use of Convolutional Neural Networks (CNNs) abides by all five principles. This class of NNs is specifically designed to process pixel data and is generally used in image recognition. The TT map can be interpreted as a pixelated view of the CMS calorimeters, making CNNs a natural approach. Any τ_h candidate can be reconstructed as a fixed-size image of TTs, where each TT acts as a pixel, and a CNN can be trained to recognize patterns associated with a τ_h. This approach can perform both the rejection of background and the calibration of the τ_h candidate by exploiting the pattern recognition capabilities of a CNN embedded in FPGA firmware. Additionally, in the endcap region only, the $\mathrm{CL}^{3\mathrm{D}}$ information can be seamlessly included in the process.

2.1 Algorithm design

The creation of L1T τ_h candidates, in both barrel and endcap, is initiated by local energy maxima in exclusive regions extending five TTs in the η direction and nine TTs along the ϕ direction, so no overlap between the clusters can be formed. Seeding TTs satisfy $E_\mathrm{T} \geq 2.5\,\mathrm{GeV}$; to ensure that not only the seed but entire clusters are contained in the HGCAL acceptance, seeds must fulfil $|i\eta| \leq 33$. All TTs within a distance $|\Delta i\eta| \leq 2$ and $|\Delta i\phi| \leq 4$ from the seed are clustered in a single τ_h candidate. Due to their characteristic dimensions, these clusters are referred to as $\mathrm{CL}^{5\times9}$.

In the HGCAL, $\text{CL}^{3\text{D}}$-based L1T τ_h candidates are selected as single clusters fulfilling $E_\text{T} > 4\,\text{GeV}$. A preselection based on a BDT developed at the time of the Phase 2 L1T technical design report and trained for PU rejection is also applied [2]. After $\text{CL}^{3\text{D}}$ candidates are selected, the matching between $\text{CL}^{5\times9}$ and $\text{CL}^{3\text{D}}$ is performed to ensure that they reconstruct the same τ_h lepton. For $\text{CL}^{5\times9}$ satisfying $|i\eta_\text{seed}| \geq 19$ the geometrical requirement $\Delta R(\text{CL}^{5\times9}, \text{CL}^{3\text{D}}) < 0.5$ is enforced.

The architecture of the TauMinator algorithm is reported in Figure 1; it is implemented in Keras [6] with a TensorFlow [7] backend, and the specific parameters of each component can be grasped in the Figure. Due to the different available TPs in the barrel and endcap areas, the algorithm is split into two independent compartments, one for each region, with separation at $|i\eta| \leq 18$. In the barrel section, the input is represented by the $\text{CL}^{5\times9}$. In the endcap section, the input is $\text{CL}^{5\times9}$ and $\text{CL}^{3\text{D}}$. In both partitions of the algorithm, the $\text{CL}^{5\times9}$ is processed by a CNN that performs the τ_h pattern recognition based on the TTs information; the additional information from the seeding TT and the $\text{CL}^{3\text{D}}$ shower shapes is concatenated to the CNN output and used as input to two dense NNs which perform the final identification and calibration of the τ_h candidate.

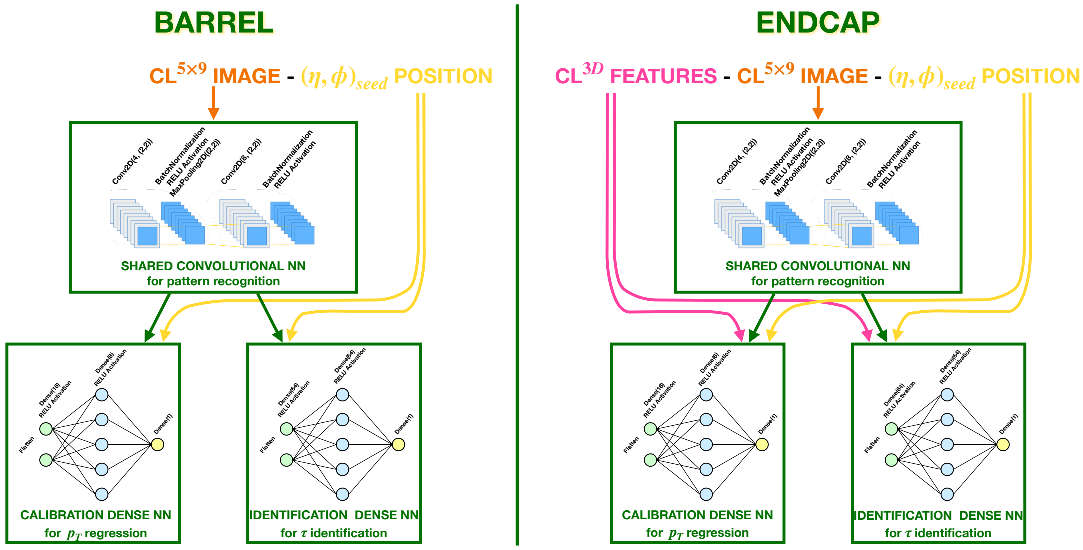

Figure 1: Visual representation of the TauMinator algorithm architecture. The TauMinator comprises two sections: barrel and endcap with separation $|i\eta| = 18$. The $\text{CL}^{5\times9}$ identifies the input obtained from the TTs of the calorimeters, with $(\eta, \phi)_\text{seed}$ the seeding tower position, while $\text{CL}^{3\text{D}}$ is the specific input from the HGCAL detector; the characteristics of both are detailed in the text. In each section of the algorithm, a standard CNN architecture is employed with the hyperparameters specified in the figure [4].

2.2 Firmware implementation

The TauMinator design outlined above is heavily influenced by the necessity to implement the CNNs into FPGA firmware; nevertheless, the architecture is built using a floating point precision architecture that is not easily implementable in FPGA firmware. Therefore, additional optimization steps need to be performed to achieve the final firmware-embedded model.

The first step is the compression of the TauMinator model to reduce the firmware resources used by the CNN using two techniques. *Quantization* consists of training a CNN whose variables have been encoded into digital quantities of fixed precision. *Pruning* consists of simplifying the CNN by reducing its complexity by removing certain weights. These two methods are exploited simultaneously to achieve maximal efficiency of the compression.

The second step is the conversion of the software into a custom HLS (High-Level Synthesis) firmware design with the `hls4ml` package [8]. Once the HLS conversion has been performed, the FPGA resources estimate can be performed. The estimates of the main resources usage, the Initiation Interval (II), and the Latency (Lat.) of each part of the TauMinator algorithm are reported in Table 1 for the barrel section. All components require a very small percentage of FPGA resources, generally remaining below 1%. It should be noted that the resources reported are for a single instance of the algorithm; therefore, the TauMinator would be well suited for a time-multiplexed trigger architecture. The firmware deployment of the algorithm in an FPGA testbench showcases 100% hardware-emulator agreement.

When translating the TauMinator algorithm from software to firmware, it is imperative to preserve its performance. This is achieved by fine-tuning all the parameters for the CNN compression and firmware synthesization. The performance attained at each step of this process is reported in Figure 2. Minimal loss in performance is achieved at each step, highlighting the successful adaptation of the TauMinator algorithm to the hardware constraints of the L1T FPGAs.

	LUT	FF	BRAM	DSP	II [ns]	Lat. [ns]
Shared Convolutional NN	1.07%	0.48%	0.00%	0.00%	22.2	55.6
Identification Dense NN	0.40%	0.09%	0.02%	0.17%	2.78	30.6
Calibration Dense NN	1.68%	0.39%	0.00%	3.28%	2.78	38.9

Table 1: Summary of the main FPGA resources used by the barrel section of the TauMinator algorithm, alongside the II and Lat. of each part of the algorithm. These results are obtained targeting a Xilinx Virtex UltraScale+ VU13P FPGA at a clock frequency of 360 MHz. The same naming of Figure 1 is used for the networks. Analogous results are obtained for the endcap section of the TauMinator algorithm [4].

3. Physics performance of the TauMinator algorithm

Figure 3 reports the physics performance of the TauMinator algorithm. On the left and in the centre, the matching efficiency and the trigger turn-ons as a function of generated p_T of the TauMinator algorithm are compared to those of the CaloTau algorithm, respectively. The matching efficiency is computed as the fraction of generated τ_h that are geometrically matched to an L1T τ_h candidate; the trigger turn-on is defined as the fraction of matched L1T objects that pass a specific p_T threshold. While the TauMinator matching efficiency is mostly comparable to the one of the CaloTau algorithm, showcasing a steep rise and a plateau approaching unity, the trigger turn-ons show a consistently better performance of the TauMinator algorithm owing to its better calibration. On the right, the single-τ_h rate is shown as a function of the offline threshold, which is evaluated as the generator p_T value at which the trigger turn-on crosses the 90% efficiency point. The TauMinator algorithm guarantees the following improvements: a reduction of the inclusive rate by 37% (from 31.4 kHz to 19.8 kHz) at a threshold of 150 GeV; or conversely, a reduction of the threshold by 14 GeV at a fixed rate of 31.4 kHz.

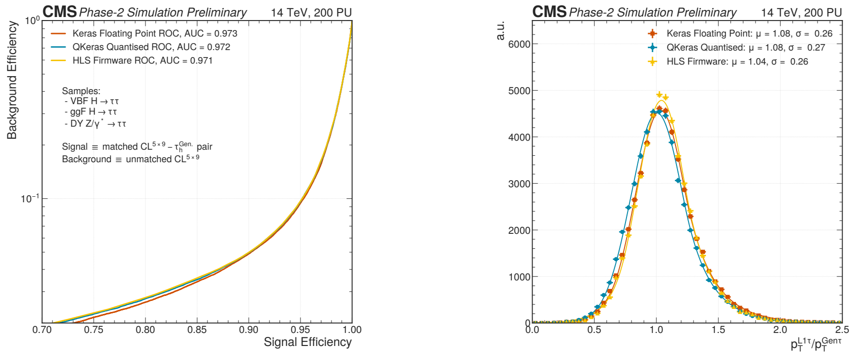

Figure 2: Receiver Operating Characteristic (ROC) curve (left) and energy response of the Level-1 τ_h with respect to the generated p_T (right) for the barrel section of the TauMinator algorithm. The results are shown for the three steps of the design, i.e. Keras software (red), QKeras quantized and pruned software (blue), and HLS firmware implementation (yellow), showcasing minimal loss of performance achieved in all the steps. Analogous results are obtained for the endcap section of the TauMinator algorithm [4].

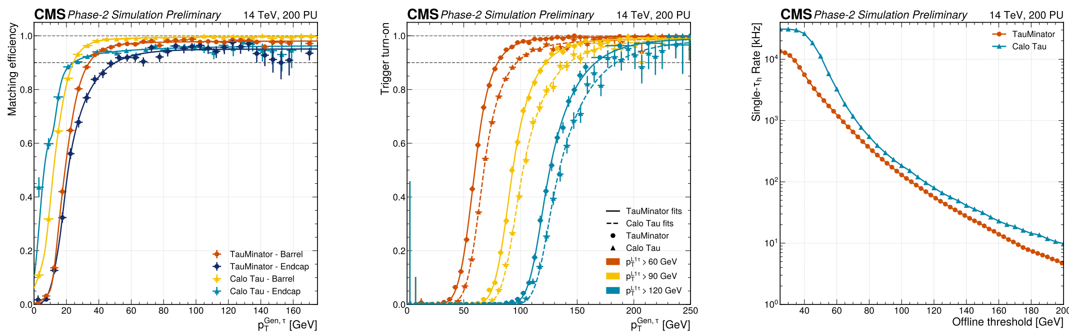

Figure 3: Comparison of the matching efficiency (left), the trigger turn-ons (centre) as a function of generated p_T, and the single-τ_h rate (right) as a function of the offline p_T, defined as the generator p_T value at which the trigger turn-on crosses the 90% efficiency point, for the TauMinator algorithm and the CaloTau algorithm. The efficiencies are evaluated in HH $\rightarrow$ bb$\tau\tau$ events at 200 PU, and the functional form of the fits consists of a cumulative Crystal Ball function [9] convolved with an arc-tangent in the high p_T region. The rate is evaluated in minimum-bias events at 200 PU [4].

4. Conclusions and outlook

The HL-LHC will pose big challenges for the CMS experiment, which will entirely replace its L1T system. In this context, the reconstruction of τ_h candidates will be particularly challenging, and the TauMinator algorithm offers an innovative and highly-performing solution to the problem by employing FPGA-embedded CNNs. The TauMinator algorithm has been successfully deployed in firmware, and it outperforms currently available standard triggering algorithms. Future developments of the TauMinator will feature the inclusion of track information, the exploration of graph neural network architectures, and the enhancement to a multi-particle identifier.

References

[1] CMS Collaboration, *The CMS experiment at the CERN LHC*, *Journal of Instrumentation* **3** (2008) S08004.

[2] CMS Collaboration, *The Phase-2 Upgrade of the CMS Level-1 Trigger*, Tech. Rep. CERN, Geneva (2020), **CERN-LHCC-2020-004, CMS-TDR-021**.

[3] J. Motta, *Overview of the HL-LHC Upgrade for the CMS Level-1 Trigger*, *PoS* **EPS-HEP2023** (2023) 534.

[4] CMS Collaboration, *Hadronic Tau Reconstruction in the CMS Phase-2 Level-1 Trigger using NNs with Calorimetric Information*, **CMS-DP-2023-062** (2023) .

[5] CMS Collaboration, *The Phase-2 Upgrade of the CMS Endcap Calorimeter*, Tech. Rep. CERN, Geneva (2017), **CERN-LHCC-2017-023, CMS-TDR-019**.

[6] Keras, *Keras website*, **https://keras.io** .

[7] M. Abadi et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. **https://www.tensorflow.org/**.

[8] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis et al., *Fast inference of deep neural networks in FPGAs for particle physics*, *Journal of Instrumentation* **13** (2018) P07027.

[9] Mark Joseph oreglia, *A study of the reactions* $\psi' \to \gamma\gamma\psi$, Ph.D. thesis, Stanford University, 1980. **SLAC Report SLAC-R-236**.