Barrel muon track reconstruction with deep learning for Level-1 trigger data scouting in the CMS experiment

CERN Summer School 2023 Report

Rocco Ardino[1,2,3,*], *Thomas* James[3,**], and ***Nicolò* Lai**[1,2,***]

[1]Department of Physics and Astronomy, University of Padova, Padova, Italy
[2]INFN, Padova Division, Padova, Italy
[3]CERN, Experimental Physics Department, Geneva, Switzerland

Abstract. In anticipation of the High Luminosity LHC, the CMS Level-1 trigger data scouting system stands to substantially extend the physics reach of the experiment. Utilizing the existing Run-3 data scouting demonstrator as a testbed, this study introduces new methodologies to refine muon track reconstruction processes in the barrel region at the Level-1 trigger. The document reports on three pivotal advancements. Firstly, it offers the initial investigation and validation of trigger muon track segments, termed super-primitives, generated within the trigger chain. Secondly, the study undertakes the validation of the software emulator corresponding to the current muon reconstruction algorithm for the barrel region. Lastly, the study employs machine learning algorithms, specifically engineered for FPGA deployment, to leverage stub-only information for fast, online reconstruction of muon physical parameters. These algorithms yield a slight improvement over standard reconstruction algorithms.

1 Introduction

The Compact Muon Solenoid (CMS) [1] experiment operates as one of the two multi-purpose detectors at the Large Hadron Collider (LHC) [2]. The LHC generates proton-proton collisions at a nominal rate of 40 MHz, thereby providing a significant data throughput to the experiments. To manage the influx of data, CMS utilizes a two-tiered trigger system for event selection [3]. The first tier, the Level-1 (L1) trigger [4], employs hardware-based, fixed-latency algorithms for coarse object reconstruction, focusing solely on calorimeters and the muon system to filter out background events. The second tier, the High-Level Trigger (HLT) [5], conducts more refined event reconstruction in software by using complete detector information and storing events that satisfy various HLT path requirements.

To manage the high data throughput from the LHC, the CMS trigger system employs strict selection criteria, which, while effective for known physics processes, could potentially filter out new or rare phenomena. Data scouting emerges as a solution to this challenge, allowing for extracting coarse, less-filtered event information directly from the trigger chain.

*e-mail: rocco.ardino@cern.ch
**e-mail: tom.james@cern.ch
***e-mail: nicolo.lai@cern.ch

Data scouting has been implemented within the HLT for over a decade [6–8]. With the upcoming upgrades for high-luminosity operations at the LHC [9], scheduled to begin by the end of this decade, experiments are preparing to cope with a significant increase in luminosity and pile-up. Hardware upgrades are planned across most sub-detectors within the CMS experiment. Notably, the trigger system will experience comprehensive modifications [10, 11]. At the Level-1 trigger stage, the architecture will be revamped to include the Particle Flow algorithm and particle tracks from the tracker back-end [12], aiming to approach the offline object reconstruction resolution at a rate of 40 MHz. As such, Level-1 trigger Data Scouting (L1DS) becomes of particular interest [13, 14]. The enhanced resolution of this data would allow for physics analyses using almost unbiased datasets and offer substantial capabilities for anomaly detection and real-time studies. In anticipation of the forthcoming high-luminosity operations and associated trigger system upgrades, the CMS collaboration has developed a Run-3 demonstrator of the L1DS system [13–15]. This demonstrator is a Level-1 data scouting infrastructure compatible with the current trigger system architecture. Given that the existing trigger objects offer insufficient resolution for several physics analyses, this platform is primarily intended for testing and validating scouting strategies, hardware configurations, and data extraction methods for high-luminosity operations. Furthermore, it facilitates the first-ever collection, exploration, and validation of various Level-1 trigger stages and objects.

This report focuses on exploring and validating track segments, also termed super-primitives or stubs, delivered by the TwinMux [16] processors to the Barrel Muon Track Finder boards employing the Kalman filter (referred to as kBMTF) [17]. The document entails offline emulation of the kBMTF algorithm and subsequent validation of the resulting BMTF muon candidates. These are compared with muons from the Global Muon Trigger (GMT), which are directly scouted at the GMT stage. Additionally, preliminary tests are conducted to evaluate the applicability of machine learning techniques for enhancing muon reconstruction based on super-primitives. These tests aim to assess the feasibility and potential advantages of implementing such techniques into the upgraded L1DS system planned for high-luminosity operations.

2 The Large Hadron Collider and CMS Experiment

The Large Hadron Collider (LHC) is located at CERN in Geneva, Switzerland. It is housed within a 27-kilometer tunnel 100 meters underground and has been operational since 2010. The LHC can accelerate protons and heavy-ion beams, achieving a center-of-mass energy ($\sqrt{s}$) up to 13.6 TeV [2, 18–20]. The complex process of injecting protons into the LHC involves a series of accelerators. This ensures an injection energy of 450 GeV into the LHC. The nominal beam structure includes 39 trains, each housing 72 bunches containing $N = 1.1 \times 10^{11}$ protons. The configuration operates at a crossing frequency of 40 MHz, equating to an inter-collision interval of 25 ns. The periodic time between collisions, called bunch crossing (BX), is a standardized temporal unit. The LHC operates at a nominal instantaneous luminosity $\mathcal{L}_{\text{inst}} = 2\times10^{34}\,\text{cm}^{-2}\text{s}^{-1}$, leading to an average pile-up of 50 proton-proton interactions per bunch crossing.

The Compact Muon Solenoid (CMS) is located at the LHC's interaction point five (P5). The detector, measuring 21.6 m in length and 15 m in diameter, weighs approximately 14,000 tons. As depicted in Fig. 1, the CMS features a central cylindrical barrel complemented by two endcaps, giving it a comprehensive design to capture diverse physics processes. A concise breakdown of the structure of the CMS detector, from its innermost to its outermost components, is as follows:

- **Silicon Tracker:** Primarily made of silicon, this subdetector is sensitive to charged particles, enabling the reconstruction of their trajectories [21].

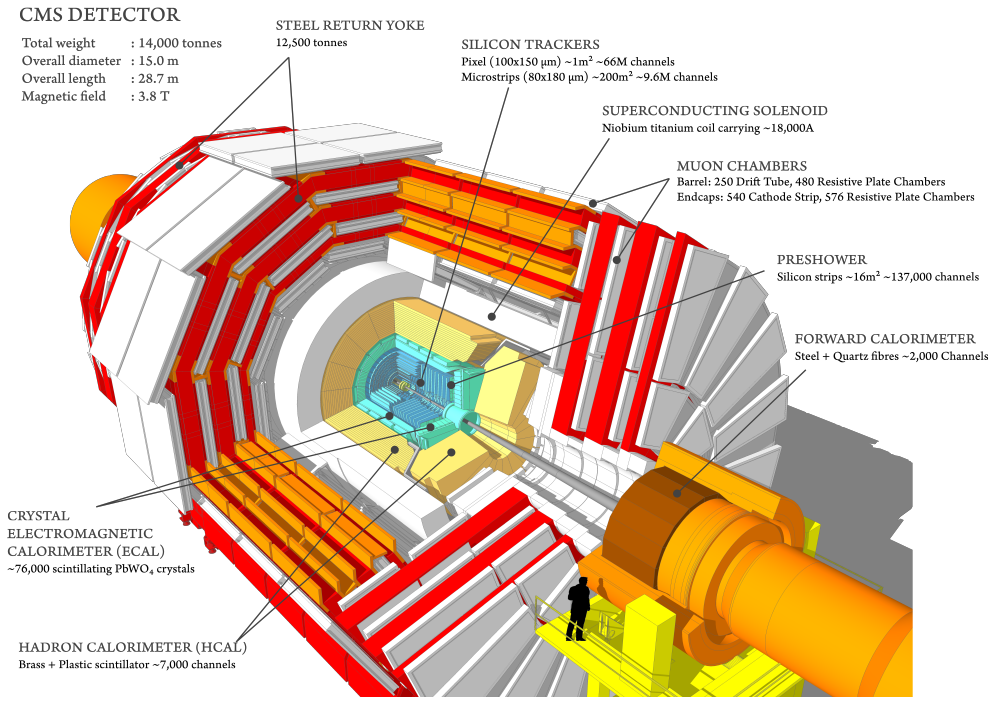


Figure 1: Illustrative depiction of the CMS detector showcasing its diverse subdetectors [26].

- **Electromagnetic Calorimeter (ECAL):** Composed of lead tungstate crystals, the ECAL measures the energy of photons and electrons [22].
- **Hadronic Calorimeter (HCAL):** This subdetector, constructed from brass and steel, measures the energy of hadrons [23].
- **Superconducting Solenoid:** Comprising niobium-titanium (NbTi), this solenoid produces a 3.8 T magnetic field [24].
- **Muon Chambers:** Designed to capture muon tracks, these chambers consist of drift tubes, cathode strip chambers, and resistive plate chambers [25].

Upgrades are planned for the CMS detector to adapt to the increased demands of the upcoming High Luminosity LHC (HL-LHC). The upgraded detector, known as CMS Phase-2, aims to maintain efficient performance under the increased pile-up conditions.

3 Overview of the CMS Level-1 Trigger and Data Scouting

The CMS Trigger System plays an essential role in the data acquisition chain, filtering the large volume of collision data generated at a rate of 40 MHz down to an offline storage rate of about 1 kHz [3]. This rate reduction is crucial, given the readout and storage constraints.

The trigger system comprises two primary stages: the Level-1 Trigger (L1T) [4] and the High-Level Trigger (HLT) [5]. The L1T, constructed with custom electronics, focuses on fast decision-making. It uses coarse-grained muon detectors and calorimeter information to reduce the event readout acceptance rate to 100 kHz. In contrast, the HLT is software-based and runs on a dedicated processor farm. The HLT uses detailed data from all subdetectors, including the silicon inner tracker, enabling more comprehensive event analysis.

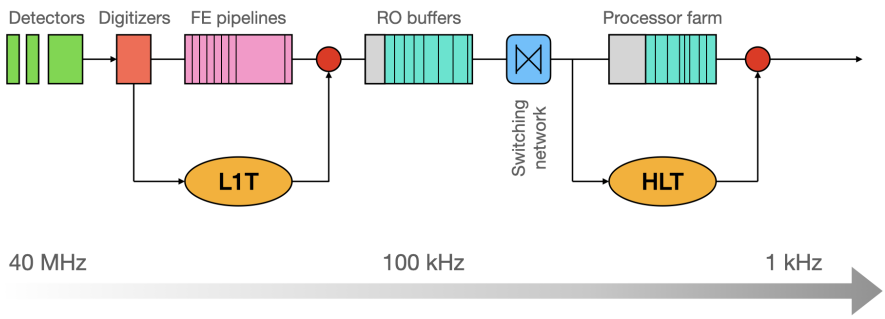


Figure 2: Schematic representation of the CMS Phase-1 trigger system.

A diagrammatic representation of the CMS Trigger System and the event rates at each stage can be seen in Fig. 2.

3.1 The CMS Level-1 trigger

The L1T [4], depicted in Fig. 3, is made of local, regional, and global components:

- **Local Triggers or Trigger Primitive Generators (TPG):** These rely on energy deposits in calorimeter trigger towers, track segments, and hit patterns in muon chambers.
- **Regional Triggers:** These merge information from the TPGs. Based on their spatial positioning, these triggers implement pattern logic to assess and rank trigger objects, such as muon candidates. Their ranking hinges on parameters like energy, momentum, and the quality of their measurements.
- **Global Triggers:** The Global Trigger determines whether to retain an event for further analysis or pass it to the HLT. This decision is based on rigorous algorithmic evaluations, the operational status of the various subdetectors, and the status of the central DAQ.

The real-time processing challenges posed by the L1T necessitate rapid evaluations of every bunch crossing (BX). Given the limited depth of the FE buffers, the system needs to perform non-trivial algorithmic assessments quickly, while FIFO memories retain data from the sub-detectors. The trigger logic, segmenting its evaluations into steps, is pipelined to accept data from a new BX every 25 ns. To achieve this, custom-programmable hardware, such as Field Programmable Gate Arrays (FPGA) and Programmable Lookup Tables (LUTs), is crucial. This infrastructure culminates in an event acceptance decision within a bounded timeframe dictated by the FIFO's storage capacity, roughly corresponding to 4 µs.

Muon Trigger

The Muon Trigger, divided into three subsystems targeting distinct η ranges, is central to muon tracking across the detector. As depicted in Fig. 3, Trigger Primitives (TP) from the Cathode Strip Chambers (CSCs) are routed to the Endcap Muon Track Finder (EMTF) and the Overlap Muon Track Finder (OMTF) via a mezzanine on the muon port card. Endcap Resistive Plate Chambers (RPCs) hits are channeled via the link board to the Concentrator Pre-Processor and Fan-out (CPPF) card, while barrel RPC hits approach the TwinMux concentrator card. Drift Tube (DT) trigger primitives reach the TwinMux card through a copper-to-optical fiber (CuOF) mezzanine. The TwinMux, in turn, crafts super-primitives,

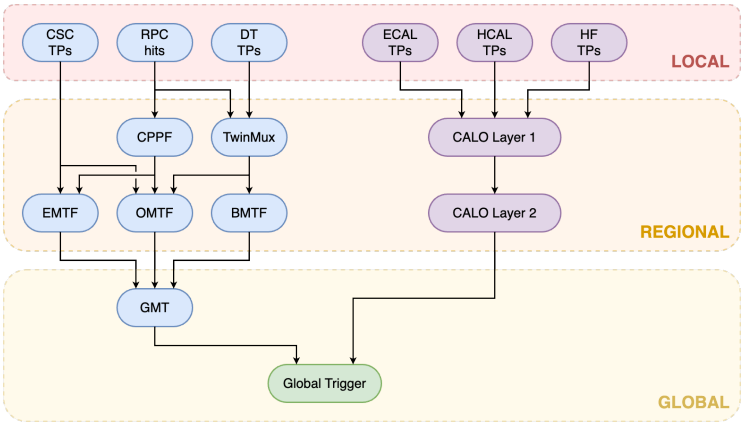


Figure 3: Schematic representation of the Phase-1 Level-1 trigger of CMS. CSC: Cathode Strip Chambers; RPC: Resistive Plate Chambers; DT: Drift Tubes; HF: Hadronic Calorimeter-Forward; ECAL: Electromagnetic Calorimeter; HCAL: Hadronic Calorimeter; TPs: Trigger Primitives; CPPF: Concentration, Pre-processing and Fan-out system; EMTF: Endcap Muon Track Finder; OMTF: Overlap Muon Track Finder; BMTF: Barrel Muon Track Finder; GMT: Global Muon Trigger.

merging the precise spatial resolution of DT trigger segments with the optimal timing characteristics of RPC hits, thereby refining the efficiency and data quality for subsequent phases. Notably, the EMTF absorbs RPC hits via the CPPF card. The OMTF, in addition to CSC TPs, considers DT TPs and RPC hits via the CPPF and TwinMux boards, the latter also providing the Barrel Muon Track Finder (BMTF) with DT and RPC hits. In its final step, the Global Muon Trigger (GMT) arranges muons, discards duplicates and transmits the top eight muon candidates to the Global Trigger.

Calorimeter Trigger

This trigger processes the energy deposited in calorimeter towers. A two-tier structure equipped with time-multiplexing capabilities ensures proficient energy sum calculations.

The global trigger (GT) functions on a "trigger menu", a spectrum of selection criteria from basic single-object p_{T} thresholds to intricate object correlations. The GT can perform up to 512 selection algorithms in parallel, and it takes all the results into account to decide whether to send an acceptance signal, named Level-1 Accept (L1A), based on a global OR condition. The L1A decision is forwarded to the sub-detectors through the Timing, Trigger, and Control (TTC) system. The L1 Trigger must evaluate every bunch crossing, with a maximum latency of 4 microseconds between a particular bunch crossing and the trigger decision distribution. Therefore, pipelined processing is necessary for near-deadtime-free operation.

3.2 The CMS Run-3 Level-1 Data Scouting demonstrator

Data scouting is historically used in CMS to augment traditional analyses, especially for rare events [6–8]. It involves using objects within the trigger chain and processing them online to enable efficient storage. This approach focuses on obtaining objects with a reduced level of detail, trading off some resolution for greater statistics.

Table 1: Inputs to the Level-1 data scouting Run-3 demonstrator.

Input system	Number of links	Objects
uGMT	8 + duplicate 8	Up to 8 uGMT final muons 8 BMTF muon candidates
Calorimeter trigger	7 + 1 spare	e/γ, tau candidates, jets and energy sums including $E_{\mathrm{T}}^{\mathrm{miss}}$
BMTF	24	BMTF input super-primitives
uGT	18	Algorithm bits

While this technique has been prevalent at the HLT level, new opportunities are presented with the High-Luminosity LHC upgrade. Specifically, the possibility of data scouting at the Level-1 trigger is emerging [27, 28]. This new strategy aims to extract L1 objects at different stages of the L1 trigger chain, with options for direct storage or online processing using heterogeneous computing methods, including FPGAs, GPUs, and big-data tools.

During LHC Run-3, a demonstrator system [15, 27, 28] has been set up to assess various concepts and understand system dynamics using real data. This demonstrator system draws data from the Global Trigger (GT), the Global Muon Trigger (GMT), the Calorimeter Trigger, and the Barrel Muon Track Finder (BMTF), with details illustrated in Tab. 1. The Run-3 Level-1 data scouting demonstrator consists of a series of FPGA-based processing boards receiving data via optical links from the trigger system. Afterward, the data is transferred to computing nodes (DSBU), where event construction and subsequent processing occur. The Run-3 demonstrator manifests as a heterogeneous system comprised of three distinct receiver board types:

1. **Xilinx KCU1500:** This development kit hosts the KU115 FPGA, capable of handling eight optical links at 10 Gb/s each. It communicates with a host computer through PCIe and employs Direct Memory Access (DMA) for data transition to this host. Subsequently, the data gets sent to a computing node for further processing. The KCU1500 was initially applied in a smaller demonstrator at the close of Run-2, where it received inputs from the uGMT, as referenced in [29].

2. **Micron SB852:** This PCIe card hosts a Xilinx VU9P FPGA and is enhanced with the Micron Deep Learning Accelerator (MDLA) [30]. Functionally similar to the KCU1500, it supports eight optical links at up to 25 Gb/s, and can use DMA over PCIe to transfer data to the host computer.

3. **Xilinx VCU128:** This board hosts a VU37P FPGA and, with the addition of an I/O extender mezzanine card, is equipped with up to ten QSFP 100G ports. These features mirror half the capabilities of a DAQ800 board, designed for the CMS Phase-2 central DAQ [11] and chosen as CMS Phase-2 L1DS concentrator board. However, similarly to the other boards, the input links are set to operate at 10 Gb/s for the demonstrator, aligning with the transmission speed of the Phase-1 Level-1 trigger.

Data from the uGMT and the calorimeter trigger are transferred through eight 10 Gb/s optical links, respectively, to a pair of Xilinx KCU1500 boards, which decode the trigger link protocol, align the links with each other and perform firmware zero-suppression. This

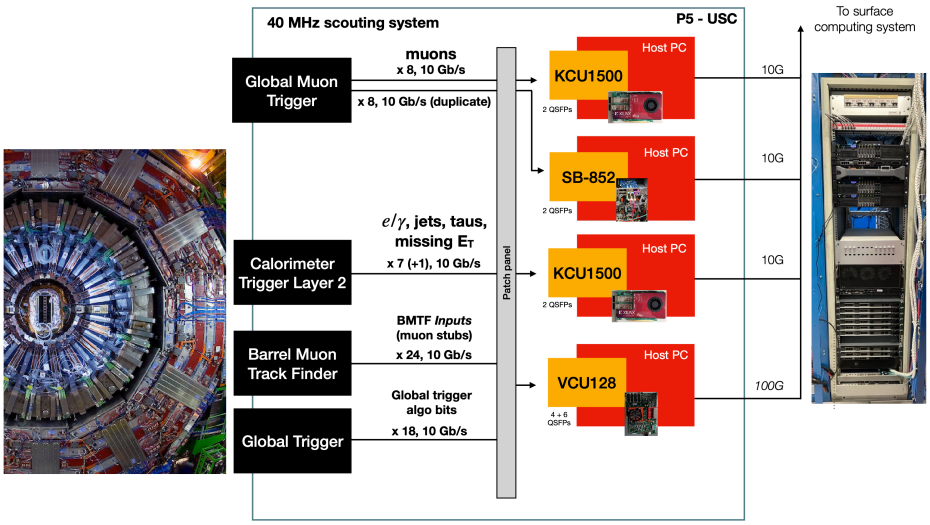


Figure 4: The L1 data scouting system demonstrator at the LHC Run-3 [31].

first stage of zero-suppression reduces the uGMT data rate by a factor of ∼10, discarding data from any bunch crossing where no muons have been found. A more fine-grained zero-suppression is performed on the host PC in software. A duplicate set of GMT muons is sent to the Micron SB852 board, used to prototype on-the-fly muon histograms involved in luminosity measurements and neural network approaches for re-calibration and classification of L1 trigger objects. The BMTF super-primitives and GT algorithm bits are sent over 24 and 18 links to the Xilinx VCU128 boards, respectively. Utilizing the High Bandwidth Memory (HBM), an additional data buffer between the trigger back-end and the commercial off-the-shelf switched network, the VU37P device sends data directly to a commercial PC.

L1 trigger objects are calibrated to achieve a specific efficiency at a given energy or transverse momentum threshold. For this reason, we cannot employ them for a direct physics analysis. Ongoing studies are exploiting Machine Learning, specifically neural networks, to re-calibrate the L1 information for semi-online analysis studies [15, 31, 32]. Although the L1 scouting system does not have to follow the strict latency requirements of the L1 trigger pipeline, it still needs to handle a large throughput of roughly 2 million muons per second. Therefore, the trained neural network must be capable of sustaining a high number of inferences per second. To accomplish this, the L1 data scouting system implements neural networks in the FPGA boards receiving the data from the L1 trigger system. While using Verilog or VHDL could be more efficient regarding resource utilization, a more straightforward solution to implement a neural network model on FPGAs exploits alternative technologies. The Micron Deep Learning Accelerator (MDLA) [30] includes a software compiler that converts neural networks to hardware instructions for an FPGA processor. The models are trained in Tensorflow [33], then converted to Open Neural Network Exchange (ONNX) format and executed on hardware using the MDLA API. Another approach is implementing neural networks in the VU37P FPGA using the Python API and command-line tool HLS4ML [34] to translate trained neural networks to synthesizable FPGA firmware.

4 Validation of Level-1 Trigger super-primitives

The Muon Trigger System, delineated in Sec. 3.1 and depicted in Fig. 3, comprises several components, among which the TwinMux boards play a pivotal role for the barrel region. These boards process trigger primitives from the Drift Tubes (DTs), integrating them with hits from the Resistive Plate Chambers (RPCs) to form super-primitives. These super-primitives are then processed by the barrel muon track finder (BMTF) for subsequent muon detection and track reconstruction. This section discusses the validation of these super-primitives, explains the functionality of the TwinMux boards, examines the integrity and synchronization of the data, and investigates the spatial distribution and multiplicity of stubs.

4.1 Trigger super-primitives overview

As specified in Sec. 3.1, TwinMux boards serve as the trigger primitive concentrators for the barrel region of the CMS detector, and it is crucial to understand their functionality for a comprehensive analysis of the trigger super-primitives.

TwinMux Board Inputs

Each TwinMux board is designed to receive inputs from the Drift Tubes (DTs) and the Resistive Plate Chambers (RPCs) from one sector of the barrel muon detector, aggregating to a total of 60 boards. The DT input, sent at a rate of 480 Mb/s, includes trigger segments comprised of position, direction, quality, and BX (Bunch Crossing) information. This data is received from DT mini-crates via Copper-to-Optical Fiber (CuOF) boards that convert the galvanic inputs to optical signals. The RPCs contribute hit information, including position and BX, at a data rate of 1.6 Gb/s.

TwinMux Algorithm and Outputs

Upon receiving these inputs, the TwinMux boards execute a series of operations to generate super-primitives. Initially, the incoming data are de-serialized and synchronized. Subsequently, a clustering algorithm is applied to the RPC hits to merge neighboring hits, resulting in a cluster position with half-strip resolution converted into DT coordinates. Special conditions are also considered, such as suppressing consecutive RPC clusters and matching close RPC and DT clusters.

The super-primitives are then constructed with considerations for quality and BX adjustments. Specifically, if $\Delta\phi \leq 15$ mrad between a DT Trigger Segment and an RPC cluster, they are matched, and a quality bit is set for the super-primitive. Furthermore, BX adjustments are made depending on the number of DT ϕ layers involved in building the DT Trigger Segment.

The TwinMux system can construct up to two super-primitives per muon station. These super-primitives are then sent to the BMTF processors for the originating wedge and its neighboring wedges. If the wheel is one of the two external wheels close to the endcaps, the system duplicates these two super-primitives to forward them to the OMTF processors. It additionally forwards the track segment data along the η direction for each muon station.

4.2 Scouting the BMTF inputs

The BMTF inputs, hereafter referred to as *stubs*, are extracted through a specialized scouting system. Each of the 12 BMTF boards—allocated one per wedge—receives super-primitives from the TwinMux boards. Specifically, each BMTF board can receive at most eight stubs per wedge, as the TwinMux system generates up to two stubs for each of the four stations.

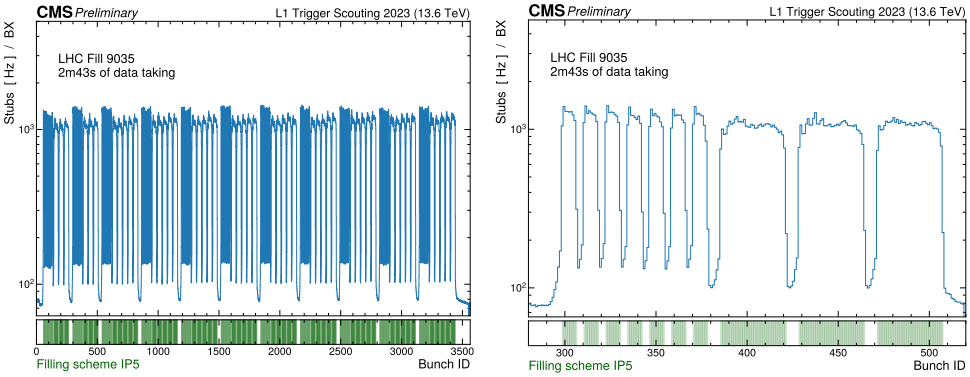
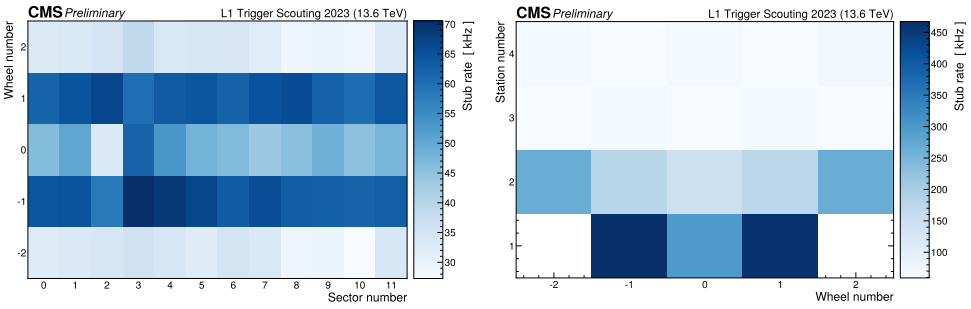


Figure 5: Stub collection rate per bunch crossing relative to the bunch crossing identifier, showing the LHC filling scheme structure. The bottom panel shows, in green, the nominal LHC filling scheme at interaction point five, identified by the string `25ns_2464b_2452_1842_1821_236bpi_12inj_hybrid`.

The scouting system interfaces with each BMTF board via two dedicated links, summing up to 24 links in total. These links can transfer a maximum of 8 stubs each, allowing an upper limit of 96 stubs to be extracted per bunch crossing. The stubs are subsequently channeled to a Xilinx VCU128 board and then forwarded to a commercial PC for further analysis.

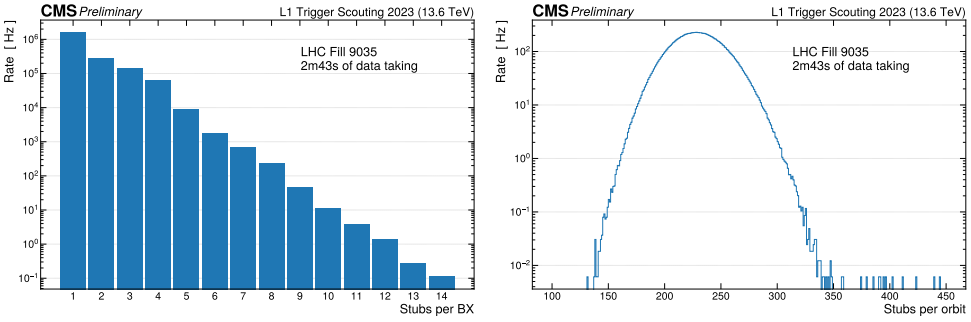
In this section and the following one, we will use 2 minutes and 43 seconds of data-taking during CMS run number 370169, amounting to approximately 6×10^8 stubs. To assess the integrity and synchronization of the data, we examine the number of stubs across all bunch crossing IDs (BXs), which range from 0 to 3654. Fig. 5a presents the stub collection rate per BX, normalized by the data collection time. This verifies the LHC filling scheme. The lower panel of the figure explicitly depicts the nominal filling scheme as delivered to interaction point five by the LHC, designated by fill number 9035 [35]. For a more detailed validation of the synchronization of BX assignments, Fig. 5b zooms into a subset of BXs. The graph clearly shows that the peaks in the stub collection rate align well with the nominal filling scheme. It is also evident that the collection rate never drops to zero, which can be attributed to the flat cosmic muon rate and particle products that span multiple BXs, along with other possible effects. Notably, the background rate is elevated when bunches are less separated in BXs and diminishes when bunches are more spaced apart.

To investigate the spatial distribution of stubs within the detector, we analyze the stub acquisition rate in sectors, wheels, and stations. Due to the current limitations of the Level-1 Trigger system, further granularity is unattainable at this stage of the trigger chain. Figures 6a and 6b display these rates as two-dimensional histograms. Fig. 6a portrays the stub acquisition rate as a function of wheel and sector. Wheels ±1 show the highest super-primitives acquisition rate. Fig. 6b focuses on the acquisition rate segmented by muon station and wheel. We observe that the innermost stations of the outer wheels (station number 1, wheels ±2) are excluded from the BMTF inputs and directed solely to the OMTF system. Consequently, the outer wheels do not contribute the highest rate of stubs to the BMTF. The acquisition rates decrease as stations move away from the interaction point.



(a) (b)

Figure 6: Stub collection rate for each wheel, sector and station, presented as heatmaps.



(a) (b)

Figure 7: Stub multiplicity per bunch crossing (left) and orbit (right).

Lastly, we examine the multiplicity of stub acquisition, quantifying the number of super-primitives collected within specified time intervals. Fig. 7a illustrates the distribution of super-primitives collected within a single bunch crossing (BX), lasting 25 ns. Fig. 7b represents the distribution for an entire orbit, comprised of 3564 BXs or approximately 89 µs. It is worth noting that most BXs yield zero stubs, even though this data point is not shown. The rate of stub multiplicity per BX demonstrates an exponential decrease. Specifically, BXs containing more than 12 stubs occur at an approximate rate of 1 Hz within the total acquisition rate of 40 MHz, effectively making them one in 40 million occurrences. Although BXs with more than 14 stubs exist, they are not included in Fig. 7a due to their negligible frequency. In contrast, the stub multiplicity within an orbit does not follow an exponential trend. As depicted in Fig. 7b, the distribution takes on a Gaussian shape, plotted on a logarithmic scale, with the mean centered around 230 stubs per orbit.

In summary, we validated and explored super-primitives using various metrics and visualizations. The data exhibited coherence in terms of its distribution and multiplicity, confirming the reliability of the stub acquisition process.

Table 2: Scales and definitions of super-primitive parameters transmitted to the BMTF.

Parameter	Bits	Range	Description
ϕ	12	[−2048, 2047]	Relative position of a segment inside a sector
ϕ_B	10	[−512, 511]	Bending angle
quality	3	[0, 7]	Number of superlayers used to construct the stub
η hits	7	"pattern"	Each bit corresponds to one chamber area 0 : no hit (less than 3 SL hits) 1 : hit (3 or 4 SL hits)
η quality	7	"pattern"	Each bit corresponds to one chamber area 0 : 3 SL hits 1 : 4 SL hits

5 Validation of the Barrel Muon Track Finder Emulator

Having validated the integrity of the super-primitives in the preceding section, the focus now shifts to their utility in reconstructing muon candidates. This reconstruction task is carried out online by the Muon Track Finder processors, with the Barrel Muon Track Finder (BMTF) boards specifically responsible for the barrel region of the detector. These BMTF boards implement a specialized muon reconstruction algorithm in hardware, efficiently converting super-primitives into track candidates.

This section is organized into two parts: the first subsection offers a brief overview of the BMTF hardware algorithm responsible for online muon candidate reconstruction in the trigger chain, while the second subsection discusses its emulation within the CMS software framework. Subsequently, we will present results that characterize the so-called BMTF muon candidates produced by the software emulation. This will also include a comparison with the Global Muon Tracker (GMT) muons, collected at the GMT level via the scouting system, to evaluate the concordance between the emulated and the hardware-based algorithms.

5.1 The Barrel Muon Track Finder

As outlined in the previous section, the Barrel Muon Track Finder (BMTF) is designed for the reconstruction of muon tracks in the central barrel region of the CMS detector ($|\eta| < 0.83$). The BMTF processes super-primitives, which are received from the TwinMux system that collates information from DT and RPC hits. The super-primitive features are summarized in Tab. 2. Each super-primitive comprises 12 bits for the ϕ-coordinate, 10 bits for the bending angle, and three quality bits. Additionally, the BMTF receives 7 bits of η hits and 7 bits of η quality for each muon station in a wedge. Due to each DT station being capable of transmitting up to two trigger primitives, a pair of input links are used to forward this information to the BMTF processor.

The BMTF utilizes a Kalman filter algorithm for the reconstruction tasks. This filter serves as a mean squared error minimizer and is mathematically similar to a χ^2 fit [36].

The state vector $x_n = (k, \phi, \phi_B)$ denotes the track parameters at each station, where $k = q/p_{\mathrm{T}}$. Tracks are initialized with a stub from the outermost available station and are propagated inwards (see Fig. 8) using the following equation:

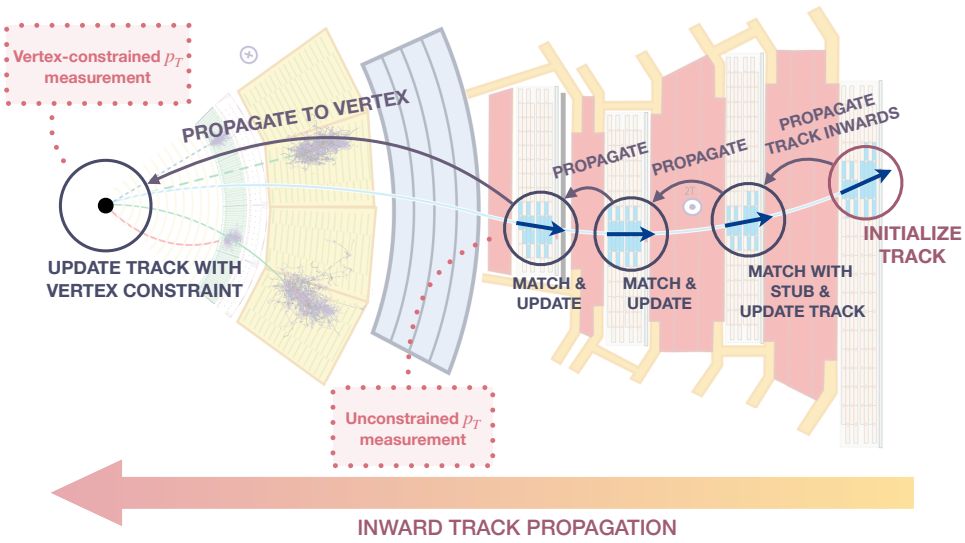


Figure 8: Visualization of the Kalman filter procedure applied to muon track reconstruction.

$$x_n = F\, x_{n-1}\,, \tag{1}$$

where the matrix F varies between stations and is determined by the detector geometry and magnetic field. Energy loss is accounted for collectively at the end of the track reconstruction process. Uncertainties are also propagated via a covariance matrix P:

$$P_n = F\, P_{n-1}\, F^{\mathrm{T}} + Q\,(k,\, x/X_0)\ . \tag{2}$$

Afterward, the closest stub $z_k = (\phi,\, \phi_B)$ is identified, and the state vector is updated based on the stub values. The residual r_n and updated state x_n^{updated} are then calculated:

$$r_n = z_n - H\, x_n = \begin{pmatrix} \phi^{\text{stub}} \\ \phi_B^{\text{stub}} \end{pmatrix}_n - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} k \\ \phi \\ \phi_B \end{pmatrix}_n\,, \tag{3}$$

$$x_n^{\text{updated}} = x_n + G\, r_n\,, \tag{4}$$

where G is the Kalman Gain matrix.

Finally, two separate transverse momentum measurements are stored: one without a vertex constraint ($p_T^{\text{unconstrained}}$) and the other incorporating it (p_{T}). The latter includes the material effects of passing through the CMS calorimeters, magnet, and tracker.

Implementing the Kalman filter in the BMTF significantly enhances the trigger's performance with respect to the legacy BMTF algorithm: it reduces the inefficiency by 25% for muons originating from the vertex while maintaining the same rate. For muons displaced by over 50 cm from the CMS center, the efficiency is improved by a factor of four [17, 37].

5.2 Emulating the kBMTF in software

The BMTF is a hardware-based system responsible for the real-time reconstruction of muon candidates in the CMS barrel region. However, a Kalman filter BMTF algorithm software emulator exists within the CMS Software (CMSSW). This software emulator mimics the hardware implementation as closely as possible, including using LUTs, thereby avoiding the simplifications that could be made when using more versatile software techniques.

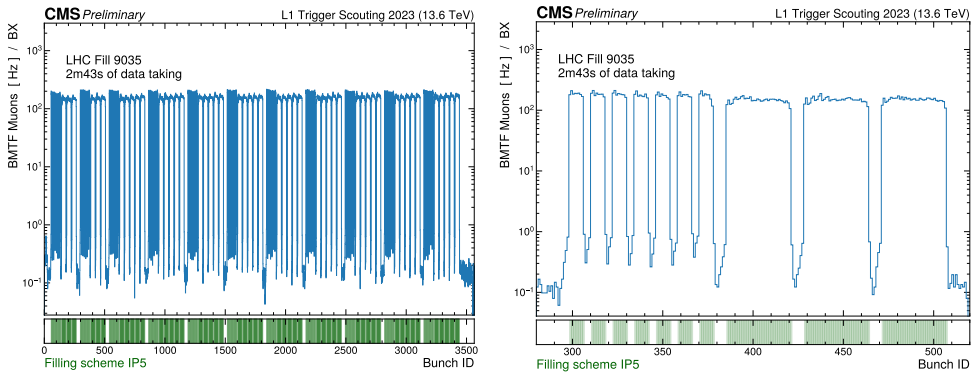
The ability to scout the BMTF inputs offers an avenue for the extraction of essential physical parameters of the muons, such as transverse momentum (p_{T}), azimuthal angle (ϕ), pseudorapidity (η), and charge. The first strategy employed is to utilize the kBMTF software emulator to reconstruct muon candidates similarly to the hardware-based trigger system. This approach provides insights into the physical parameters of the reconstructed muons and allows the exploration of intermediate quantities within the algorithm that are not forwarded to the GMT in the actual hardware implementation. One distinct advantage of using the software emulator lies in its ability to access the "full resolution" of the BMTF reconstruction process. While some quantities are truncated or approximated in the hardware version before being sent to the GMT due to bandwidth constraints, the software emulator bypasses these limitations. Consequently, this permits a more fine-grained analysis, particularly for parameters like the displacement measure. It is important to note, however, that even though the resolution of the software emulator is superior to that of the hardware trigger, it still falls short compared to the offline reconstruction resolution.

This subsection will discuss occupancy and multiplicity plots, drawing parallels to similar observations made for Figures 6 and 7. Physical distributions of the reconstructed muons will not be presented in this report as they do not offer significant new information. However, it has been verified that these distributions align with expectations.

Emulated BMTF candidates occupancy and multiplicity

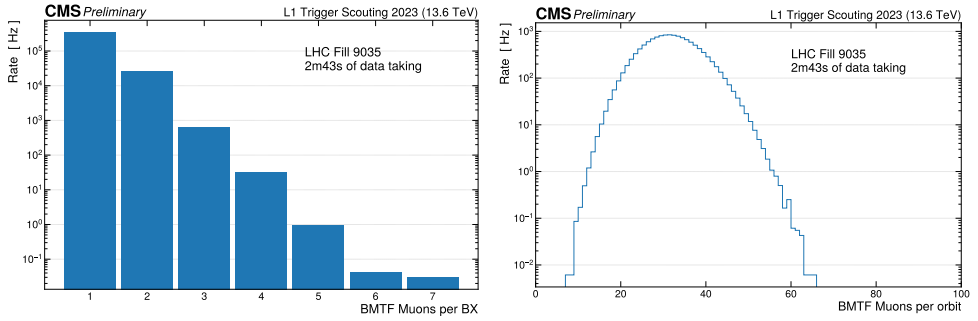
The occupancy rate of the emulated BMTF muon candidates across the bunch crossing IDs (BXs) is presented in Fig. 9a and Fig. 9b. Similar to the stub occupancy discussed in Sec. 4.2, the BMTF candidate rate is examined in the context of the LHC filling scheme, validating that the rate distribution aligns with the nominal LHC bunch structure. Given that the BMTF candidates are derived from stubs, the rate distribution naturally inherits the properties related to synchronization and occupancy. A notable difference lies in the rate scale; the emulated BMTF candidates exhibit a rate approximately an order of magnitude lower than the stubs. This reduction is attributed to multiple factors. Initially, not all stubs are associated with muons. Furthermore, the inherent inefficiency of the detector system and the reconstruction algorithm is non-negligible.

Subsequently, the multiplicity of the BMTF muon candidates is explored, delineating the number of candidates emulated within specific time intervals. The rate of BMTF candidate multiplicity per BX, as shown in Fig. 10a, follows a decreasing trend with increasing multiplicity. This trend is somewhat analogous to the stub multiplicity per BX. However, there are noticeable differences: while the BMTF candidates drop sharply after 4 per BX, the stubs exhibit a more gradual decline, extending up to 14 stubs per BX as seen in Fig. 7a. The distributions of BMTF candidates and stubs across an entire orbit are distinct. The BMTF candidate distribution, as visualized in Fig. 10b, exhibits a bell-shaped curve, pointing towards a most probable value of ~30 for the emulated BMTF candidates multiplicity per orbit. Furthermore, its shape is slightly asymmetric and positively skewed, while the stubs' multiplicity per orbit is highly symmetrical.



(a) (b)

Figure 9: BMTF candidates rate per bunch crossing relative to the bunch crossing identifier, showing the LHC filling scheme structure.



(a) (b)

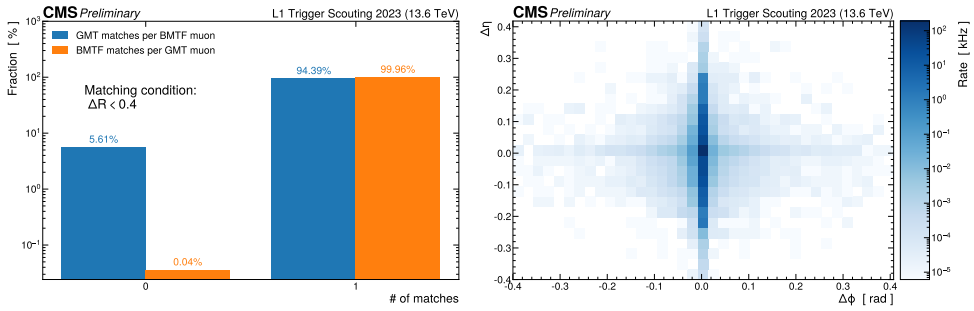
Figure 10: BMTF candidates multiplicity per bunch crossing (left) and orbit (right).

Matching emulated BMTF candidates with GMT muons

We then proceeded to compare emulated BMTF candidates with the GMT muons. This comparative study acts as a final verification, ensuring that what we are collecting and emulating aligns closely with the data present in the trigger system, specifically in its hardware-based format. As presented in Tab. 1 and Fig. 4, up to eight GMT muons are transferred to a pair of Xilinx KCU1500 boards, subsequently relayed to a host PC for storage. Our methodology involved sourcing the stubs from the scouting system, emulating the kBMTF, and concurrently acquiring the GMT muons. Consequently, we attempted to establish a match between an emulated BMTF muon candidate, characterized by "full resolution" quantities, and a scouted GMT muon derived from the trigger system.

The methodology to match BMTF muon candidates with GMT muons is established in terms of the distance in the $\eta - \phi$ plane, represented as:

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2} \tag{5}$$



(a) (b)

Figure 11: Left: matching fractions per BMTF candidate (blue) and GMT muon (orange). Right: Differences in η and ϕ between matched objects.

with $\Delta\phi = \phi_{\text{BMTF}} - \phi_{\text{GMT}}$ and $\Delta\eta = \eta_{\text{BMTF}} - \eta_{\text{GMT}}$. A matching criterion is defined by the threshold R_{min}. When ΔR is below this threshold, we infer a successful match, suggesting that both entities likely represent the identical physical muon. For the purposes of this study, we adopted a $R_{\text{min}} = 0.4$ as our benchmark, although in practice, a $\Delta R < 0.1$ is indicative of a confident match.

We explored two similar strategies to match BMTF candidates and GMT muons. The first method isolates a BMTF muon candidate within a specified BX and assesses its matching criterion against every GMT muon. Conversely, the second approach starts with a GMT muon and calculates its ΔR relative to all BMTF candidates within that BX. In both methodologies, pairs are chosen to optimize the value of ΔR. The comparative outcomes of these techniques are illustrated in Fig. 11a, with the first strategy represented in blue and the second in orange. Notably, a unique match is consistently observed for both approaches. A slight difference in outcomes is evident between the methods, with the second strategy showing higher efficiency. Particularly, for a specific BX, 99.96% of GMT muons consistently find a BMTF counterpart. Conversely, BMTF muons correlate with a GMT counterpart in 94.39% of the times. This can be attributed to the presence of BMTF muons not subsequently selected and relayed by the GMT, rendering the matching efficiency marginally diminished. We decided to adopt the more efficient strategy of the two.

To further explore the matching characteristics between BMTF candidates and GMT muons, we present a two-dimensional histogram in Fig. 11b, binned in $\Delta\phi$ and $\Delta\eta$. This visualization reveals that most matches are well-aligned in ϕ, as evidenced by a pronounced peak at $\Delta\phi = 0$ and minimal deviations in the tails. In contrast, the $\Delta\eta$ distribution exhibits a broader spread with more pronounced tails. Despite this, using a tighter matching criterion of $\Delta R < 0.1$ instead of the previously set $R_{\text{min}} = 0.4$ threshold encompasses the majority of the matches. This suggests that when a match between a BMTF candidate and a GMT muon is identified, they are typically proximate within the $\eta - \phi$ space.

6 Muon reconstruction from trigger super-primitives

Previously, we discussed the scouting system's capability to derive stubs for muon track reconstruction and determine physical observables like transverse momentum and charge. We utilized the kBMTF algorithm emulator, replicating the Kalman filter's operations in the

BMTF hardware. The emulator's strict adherence to hardware operations, however, limits software flexibility advantages. This study aims to explore alternatives to the BMTF Kalman filter within the scouting system, ideally surpassing the Level-1 trigger Kalman filter's performance. The absence of tight latency constraints in the scouting system allows for this flexibility. Still, a standalone online software algorithm would be inefficient in terms of speed. Thus, we look towards High-Level Synthesis (HLS), particularly HLS4ML [34], to develop an offline-trained machine learning algorithm synthesized into hardware for rapid real-time inferences on FPGAs.

6.1 Challenges and limitations

Using HLS4ML offers a significant advantage for track reconstruction by leveraging machine learning. This combination provides both algorithmic flexibility and the speed of hardware-accelerated inference. However, the current version of HLS4ML has its limitations; it does not support many modern neural network architectures. Therefore, we are constrained to simpler architectures like feed-forward, fully connected neural networks. Another challenge is the hardware capacity. The trained network weights must fit into the FPGA, restricting us from using large networks. Given our current hardware, we have set a limit on the number of weights, which is in the order of a few thousand parameters. One promising technique is knowledge distillation [38]. Here, a sophisticated neural network, trained offline, acts as a guide for a smaller FPGA-compatible network. This student network then captures essential knowledge from the larger model. Furthermore, other model compression techniques could be employed, such as quantization and pruning [39, 40]. The former reduces the number of bits required to represent weights and biases, while the latter directly removes connections or neurons from the architecture.

Future FPGAs may accommodate larger networks. Similarly, advancements in HLS4ML could enable the synthesis of more advanced models. Despite these potential developments, this report focuses on using simple networks compatible with current HLS4ML capabilities.

6.2 Muon reconstruction strategies

Muon track reconstruction using trigger super-primitives can be divided into two distinct tasks. The initial task, track-finding, focuses on pattern recognition. Given a complete set of stubs within a BX, the objective is to identify the specific combination of stubs resulting from a muon traversing the muon spectrometer. The Kalman filter addresses this through inward track propagation, subsequently matching the closest stub in the propagated station. Alternative techniques might utilize different analytical methods or employ clustering algorithms. The subsequent task in muon track reconstruction, track parameters assignment, is a fitting endeavor. Given the correct stubs a muon produces, we aim to deduce its physical properties: charge, transverse momentum, ϕ, and η. While the Kalman filter updates track parameters post-stub matching utilizing pre-calculated detector data and the magnetic field, machine learning algorithms can train on objects reconstructed offline. This leverages comprehensive detector information, ensuring maximum resolution on physical properties to make inferences on trigger super-primitives. Integrating both tasks within a single machine learning framework is plausible. However, challenges arise due to variable stub counts across BXs and the distinct number of muon-generated stubs ranging from 2 to 4. This variability challenges conventional machine learning algorithms, especially those compliant with hardware limitations. That said, Recurrent Neural Networks [41], particularly Long Short-Term Memory (LSTM) [42] networks, may be suited for this task.

In this report, our focus is solely on the parameters assignment task. Utilizing the recognized muon-generated stubs, we employ machine learning to infer charge, transverse momentum, and the kinematic angles η and ϕ. A primary advantage of this approach is its adaptability; any track-finding solution can be seamlessly integrated before track fitting. This is due to our methodology's dependence only on the accurate stub set for observable inference, rendering the strategy both flexible and primed for enhancement with an adept pattern recognition algorithm. Explorations into consolidating the tasks within a singular machine learning model have not been pursued further in this study.

6.3 Dataset preparation

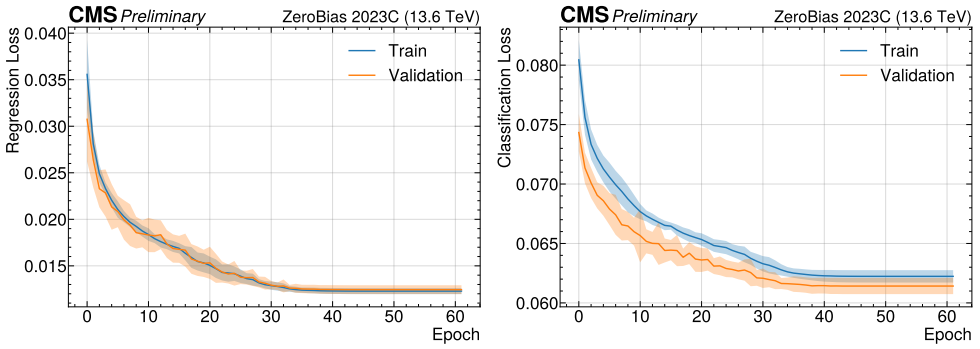
Training a machine learning algorithm to assign physical attributes to input stubs requires an accurate training dataset comprising both stub data and offline reconstruction details. We utilize the offline-reconstructed muons, termed RECO muons, which are accompanied by Level-1 trigger data. The ZeroBias RUN2023C dataset, obtained via random triggering to avoid bias from trigger selection, serves this purpose. We execute offline reconstruction to get the target attributes for neural network training. During offline reconstruction, we also extract associated GMT muons, BMTF candidates, and kBMTF tracks. Through kBMTF tracks, we access the hit pattern and stub data for the machine learning input. To align RECO muons, GMT muons, BMTF candidates, and kBMTF tracks to the same physical object, we implement a matching procedure. For each RECO muon, suitable BMTF candidates are identified by matching their η to a kBMTF track, while the d_{xy} and the $p_T^{\text{unconstrained}}$ to a GMT muon. If these criteria are satisfied, the alignment of the kBMTF track, BMTF candidate, and GMT muon is confirmed. A subsequent requirement is a match between the RECO and GMT muon with $\Delta R < 0.1$.

Our training set is constructed as follows: the input variables are derived from stub quantities and their transmission path to the BMTF boards. These include the sector (4 bits), wheel (3 bits), station (2 bits), stub quality (3 bits), two η variables (7 bits each) with corresponding quality variables (7 bits each), a tag feature (1 bit) to select between the two η sets, ϕ (12 bits), and the bending angle ϕ_B (10 bits). The goal is to predict RECO p_{T}, η, ϕ, and charge.

It is worth noting that the number of stubs associated with each RECO muon can vary. This is because offline reconstruction in the barrel muon spectrometer can be achieved with a minimum of two stubs. A consistent number of features in the training set is essential. Thus, we use a four-stub configuration. We replicate existing stubs for RECO muons linked with only two or three stubs to compensate. We considered using dummy values for missing stubs, but given that 53% of RECO has two stubs and 29% has three, totaling 82% of entries with missing values, the approach was less effective. As a result, our final dataset consists of 45 features (11 stub features multiplied by four, plus the stub count), equivalent to 254 bits. For optimal results, the four stubs are ordered by station. The first set of stub features corresponds to the innermost stub, and the last to the outermost. In instances with missing stubs, existing stubs are duplicated to ensure consistent station-wise ordering.

6.4 Neural network design

To predict the target RECO physical attributes from the stub feature set using machine learning, we first need to determine the appropriate model class. Neural networks were selected due to their compatibility with the HLS4ML framework, aiming to facilitate real-time inference through hardware implementation. Our design remains confined to fully connected feed-forward architectures, further simplifying integration into HLS4ML. Due to hardware constraints, we have limited our model to only include a few thousand parameters. With 45



(a) (b)

Figure 12: Left: regression loss function per epoch, averaged over 50 iterations, during training (blue) and validation (orange). Right: classification loss function per epoch, averaged over 50 iterations, during training (blue) and validation (orange).

input nodes based on our dataset configuration, the selected architecture incorporates four hidden layers of 64, 32, 16, and 8 neurons, respectively. The output layer is composed of four neurons representing each target attribute. This configuration results in a total of 5724 parameters. Each neuron incorporates a nonlinear activation function. Among the tested functions—ReLU (Rectified Linear Unit) [43], eLU (Exponential Linear Unit) [44], and GELU (Gaussian Error Linear Unit) [45]—the eLU function demonstrated superior results.

Furthermore, the output layer's functionality varies based on the task designated to each neuron. The neurons associated with the target variables p_T, η, and ϕ are dedicated to regression tasks. Their outputs are evaluated using regression loss metrics. Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics were assessed, with MAE exhibiting enhanced performance. Conversely, the neuron corresponding to the charge attribute operates as a binary classifier, necessitating the application of a Binary Cross-Entropy (BCE) loss metric. Notably, during the training phase, the network simultaneously accommodates both loss metrics, exemplifying multi-task optimization. A weight factor adjusts the regression loss to equate with the BCE's magnitude. Finally, to enhance the model's robustness and prevent overfitting, we incorporate L2 regularization with a strength of $\lambda = 10^{-3}$. Preliminary testing of different regularization strengths indicated a negligible impact on network performance. In addition to L2 regularization, we examined alternative regularization techniques including Batch Normalization [46] and Dropout [47]. Preliminary evaluations indicated that these methods resulted in inferior performance. Hence, they were not pursued further in our study.

6.5 Training strategy

For our neural network training, we allocated datasets as follows: 63%, or 380×10^3 muons, for training, 7%, or 40×10^3 muons, for validation, and the remaining 30%, or 180×10^3 muons, for testing the model's predictive performance in muon reconstruction.

We adopted the standard mini-batch training approach, identifying batch size as a pivotal hyper-parameter for both training speed and optimization consistency. While smaller batch sizes (32, 64, 128, 256) and larger ones (512, 1024, 2048, 4096) exhibited similar reconstruction accuracy, a batch size of 512 muons was chosen for its balance between stability

and training speed. For production-ready networks, however, smaller batch sizes are recommended. The ADAM optimizer [48] is chosen to solve the optimization problem. Its learning rate, η, is a crucial hyper-parameter for precise muon reconstruction. Through experiments with constant learning rates ranging from 10^{-2} to 10^{-8}, performance remained inferior to the L1 Kalman Filter algorithm in the BMTF. Both cyclic learning rate schedulers (per optimizer steps and epochs) also underperformed relative to the Kalman Filter. A custom learning rate scheduler, starting from $\eta = 10^{-2}$ and reducing by half whenever the total validation loss showed less than 5% improvement over 5 epochs, yielded optimal results. Various thresholds were assessed, but none produced satisfactory outcomes. We introduced an early stopping mechanism, activated when the total validation loss improved by less than 0.001% over ten successive epochs. Coupled with the learning rate scheduler, early stopping was typically triggered after about 60 epochs. Fig. 12 illustrates the training and validation losses averaged over 50 training iterations with randomized datasets. Notably, between epochs 40 and 60, a seeming loss plateau is evident. However, our learning rate scheduler achieves finer optimization in this phase, resulting in improved prediction compared to an immediate training stopping upon reaching this plateau.

To streamline the training process and avert excessively large weights (subject to L2 penalties), we normalized each feature by a designated power of 2. Although unconventional in typical machine learning applications, this method significantly aids real-time data normalization in FPGA hardware during online inference. Positive-only features were adjusted to range approximately between 0 and 1, while features covering both positive and negative domains were normalized to approximate the range between -1 and $+1$. To further facilitate the training and improve subsequent reconstruction performance, we utilized $1/p_{\mathrm{T}}$ as the target for transverse momentum. Due to the more uniform distribution of $1/p_{\mathrm{T}}$ values within the $0-1$ range after normalization, this approach markedly improved the accuracy of p_{T} reconstruction.

6.6 Evaluation and results

To rigorously evaluate the performance of the reconstruction network, we utilize an orthogonal test dataset, distinct from the training and validation datasets. This dataset comprises the complete input stub information fed into the trained neural network for target prediction. Although RECO targets are included, they are reserved exclusively for computing performance metrics rather than for evaluation purposes. Additionally, the dataset contains Level-1 BMTF candidate quantities computed by the Kalman Filter, serving as a basis for comparison with our machine learning approach.

For charge reconstruction, identifying a performance evaluation metric is straightforward. Given its nature as a binary classification problem, we employ prediction accuracy as the performance metric. We report a charge assignment accuracy of 98.7%, slightly higher than the BMTF's 98.1%. Assessing the network's efficacy in reconstructing p_{T}, η, and ϕ is less trivial. While loss functions on the test set provide insightful data, they do not adequately reflect typical metrics conventionally used to evaluate the performance of a reconstruction algorithm, such as resolution and residuals relative to target quantities. Consequently, we evaluate the neural network's reconstruction capability through calculating the p_{T} resolution $\Delta p_{\mathrm{T}}/p_T^{\mathrm{RECO}}$ where $\Delta p_{\mathrm{T}} = p_T - p_T^{\mathrm{RECO}}$ and the residuals for η and ϕ, given by $\Delta\phi = \phi - \phi^{\mathrm{RECO}}$ and $\Delta\eta = \eta - \eta^{\mathrm{RECO}}$. Both the neural network predictions and the Level-1 trigger quantities reconstructed by the kBMTF algorithm are considered in this computation.

Fig. 13 depicts the transverse momentum reconstruction resolution using both the machine learning method (blue) and the conventional Kalman Filter of the BMTF (orange). Notably, the distributions in Fig. 13a are derived from nominal transverse momentum values

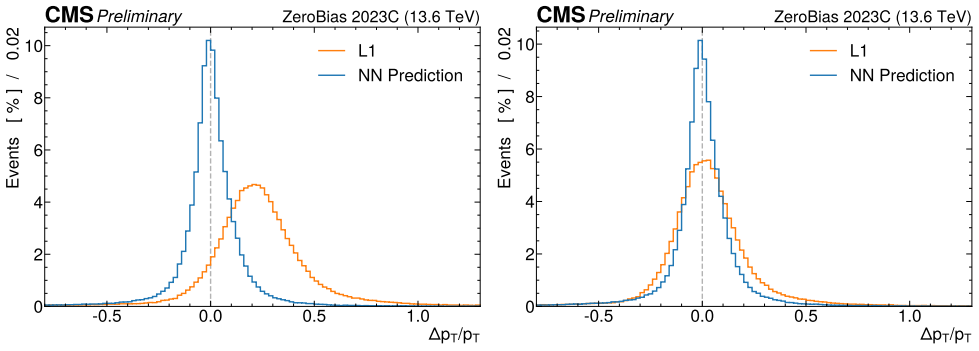


Figure 13: Transverse momentum resolution, relative to offline reconstruction, of the neural network prediction (blue) and the BMTF assignment (orange). The right figure shows the p_T resolution using a re-calibrated BMTF candidate transverse momentum.

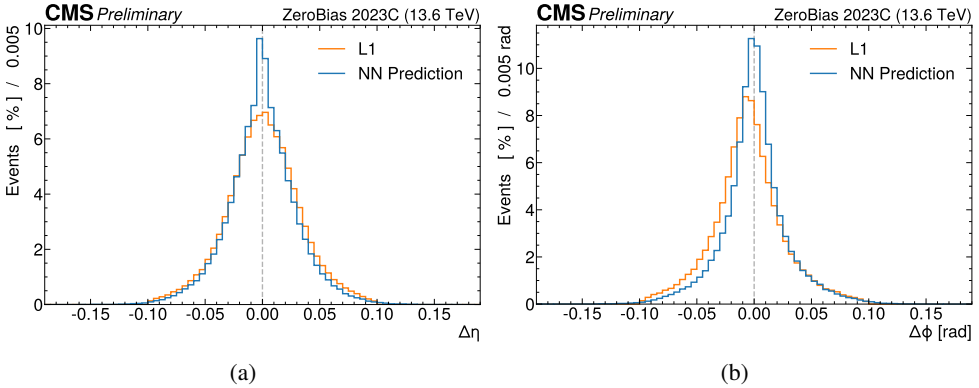


Figure 14: Left:η residuals, relative to offline reconstruction, of the neural network prediction (blue) and the BMTF assignment (orange). Right: ϕ residuals, relative to offline reconstruction, of the neural network prediction (blue) and the BMTF assignment (orange).

output by the network and the kBMTF. A marked resolution disparity, accentuated by the suboptimal BMTF resolution, can be traced back to an intentional miscalibration in the Level-1 trigger. In the trigger system, the reconstruction algorithm deliberately assigns a transverse momentum value approximately 1.2 times larger than the physically accurate value. This intentional miscalibration of p_T has historically been employed to enhance trigger efficiency over a wide p_T range. Traditionally, trigger objects, such as BMTF candidates, with intentionally altered quantities like transverse momentum were not problematic. The primary reason is that they were not designed for precise physical analyses. Instead, the main objective of the trigger system was to select events of interest for subsequent stages in the data processing pipeline, allowing for flexibility in the physical interpretation of the quantity values. With the introduction of data scouting, especially Level-1 data scouting, during the High Luminosity

phase of the LHC, there is a renewed interest in utilizing trigger objects for in-depth analyses. This means that the longstanding practice of miscalibrating trigger quantities might become a concern. The machine learning-powered reconstruction strategy presented in this report inherently addresses and overcomes this issue. In this study, we subsequently recalibrated the L1 p_{T} by a factor of 1.2, leading to the outcomes shown in Fig. 13b. The gap between the two methodologies narrows, though the machine learning approach retains a slight edge over the traditional method.

Lastly, Fig. 14 showcases the residuals for η (Fig. 14a) and ϕ (Fig. 14b) for both methodologies. In each scenario, predictions derived from the neural network exhibit marginally superior performance compared to the conventional approach.

7 Conclusions and outlook

In this study, we have focused on advancing the methods for muon track reconstruction in the Level-1 trigger data scouting system of the CMS experiment at the LHC. The work introduces several original contributions to the field. Firstly, we conducted the first investigation and validation of trigger super-primitives built by the CMS trigger system. This sets a foundation for understanding the reliability and utility of these trigger objects for possible analysis studies in the future. Secondly, the work includes the validation of the Kalman filter reconstruction algorithm software emulator, allowing for the possibility of emulating a crucial step in the Level-1 trigger chain, the Barrel Muon Track Finder algorithm, entirely offline, starting from super-primitive information. Lastly, we successfully employed machine learning algorithms designed for FPGA implementation to utilize super-primitives information for fast, online reconstruction of muon physical parameters. Notably, the performance of this approach marginally exceeds that of the standard kBMTF currently in use by the trigger system. The advancements presented have broad implications, particularly for the efficacy and reach of the L1 data scouting system in the CMS experiment. They are significant steps forward in optimizing and implementing novel techniques into the CMS Level-1 data scouting system, deviating from the traditional strategies employed by the trigger system that have to satisfy the stringent constraints of the trigger pipeline.

Future studies should improve machine learning models to enhance reconstruction precision and hardware resource usage. Specifically, layer quantization and knowledge distillation are worth investigating before transitioning networks to hardware with HLS4ML. Alternatives like monotonic networks and convolutional neural networks, mindful of hardware constraints, should also be explored [49, 50]. The relationship between p_{T} resolution and RECO transverse momentum needs further exploration, mainly as low p_{T} statistics limit current models. A dataset better representing high p_{T} muons will be crucial for this analysis. Additionally, integrating track finding with parameter fitting into one machine learning model could streamline the process. We currently train networks on predetermined stubs, but a unified approach, possibly using RNNs, could identify promising muon candidates from all collected stubs. Finally, establishing a figure of merit for network performance in muon parameter reconstruction is necessary. This would provide an objective network quality assessment, guiding future development beyond BMTF Kalman filter comparisons.

References

[1] The CMS Collaboration, Journal of Instrumentation **3**, S08004 (2008)

[2] L. Evans, New Journal of Physics **9**, 335 (2007)

[3] The CMS Collaboration, Journal of Instrumentation **12**, P01020 (2017)

[4] A. Tapper, D. Acosta (CMS), *CMS Technical Design Report for the Level-1 Trigger Upgrade* (2013), https://cds.cern.ch/record/1556311

[5] W. Adam et al. (CMS Trigger, Data Acquisition Group), Eur. Phys. J. C **46**, 605 (2006), hep-ex/0512077

[6] J. Duarte, *Fast Reconstruction and Data Scouting* (2018), 1808.00902

[7] S. Mukherjee (CMS), *Data Scouting and Data Parking with the CMS High level Trigger* (2020), https://cds.cern.ch/record/2766071

[8] D. Anderson (CMS), PoS **ICHEP2016**, 190 (2016)

[9] *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1* (2017)

[10] The CMS Collaboration (CMS), *The Phase-2 Upgrade of the CMS Level-1 Trigger* (2020), https://cds.cern.ch/record/2714892

[11] The CMS Collaboration, *The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger* (2021), https://cds.cern.ch/record/2759072

[12] C. Herwig (CMS), Journal of Instrumentation **18**, C01037 (2023)

[13] G. Badaro, U. Behrens, J. Branson, P. Brummer, S. Cittolin, D. Da Silva-Gomes, G.L. Darlea, C. Deldicque, M. Dobson, N. Doualot et al., EPJ Web Conf. **245**, 01032 (2020)

[14] R. Ardino, C. Deldicque, M. Dobson, S. Giorgetti, G. Grosso, T. James, E. Meschi, D. Rabady, A. Racz, H. Sakulin et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **1047**, 167805 (2023)

[15] T.O. James (CMS), *The Level 1 Scouting system of the CMS experiment* (2023), https://cds.cern.ch/record/2852916

[16] A. Triossi, M. Bellato, J.C. Ruiz, J. Ero, C.F. Bedoya, G. Flouris, C. Foudas, L. Guiducci, N. Loukas, Journal of Instrumentation **12**, C01095 (2017)

[17] S. Mallios, Ph.D. thesis, Ioannina U. (2019)

[18] O.S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, P. Proudlock, *LHC Design Report* (2004), http://cds.cern.ch/record/782076

[19] O.S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, P. Proudlock, *LHC Design Report* (2004), http://cds.cern.ch/record/815187

[20] M. Benedikt, P. Collier, V. Mertens, J. Poole, K. Schindl, *LHC Design Report* (2004), http://cds.cern.ch/record/823808

[21] V. Karimäki, M. Mannelli, P. Siegrist, H. Breuker, A. Caner, R. Castaldi, K. Freudenreich, G. Hall, R. Horisberger, M. Huhtinen et al. (CMS), *The CMS tracker system project: Technical Design Report* (1997), http://cds.cern.ch/record/368412

[22] *The CMS electromagnetic calorimeter project: Technical Design Report* (1997), https://cds.cern.ch/record/349375

[23] *The CMS hadron calorimeter project: Technical Design Report* (1997), https://cds.cern.ch/record/357153

[24] *The CMS magnet project: Technical Design Report* (1997), https://cds.cern.ch/record/331056

[25] J.G. Layter (CMS), *The CMS muon project: Technical Design Report* (1997), http://cds.cern.ch/record/343814

[26] *Detector | CMS Experiment*, https://cms.cern/detector
[27] G. Badaro, U. Behrens, J. Branson, P. Brummer, S. Cittolin, D. Da Silva-Gomes, G.L. Darlea, C. Deldicque, M. Dobson, N. Doualot et al., EPJ Web Conf. **245**, 01032 (2020)
[28] R. Ardino, C. Deldicque, M. Dobson, S. Giorgetti, G. Grosso, T. James, E. Meschi, D. Rabady, A. Racz, H. Sakulin et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **1047**, 167805 (2023)
[29] G. Badaro, U. Behrens, J. Branson, P. Brummer, S. Cittolin, D. Da Silva-Gomes, G.L. Darlea, C. Deldicque, M. Dobson, N. Doualot et al., EPJ Web Conf. **245**, 01032 (2020)
[30] Micron Technology Inc., *Micron deep learning accelerator software development kit* (2023), uRL: https://github.com/FWDNXT/SDK
[31] T.O. James (2022), 21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research, https://indico.cern.ch/event/1106990/contributions/4991226/
[32] T.O. James (2023), 2023 CERN openlab Technical Workshop, https://indico.cern.ch/event/1225408/contributions/5243978/
[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin et al., *Tensorflow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org, https://www.tensorflow.org/
[34] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran et al., Journal of Instrumentation **13**, P07027 (2018)
[35] The CMS Collaboration, *Online Monitoring System (OMS) Tutorial*, https://twiki.cern.ch/twiki/bin/view/Main/OnlineMonitoringSystemOMSTutorial
[36] G. Welch, G. Bishop, Proc. Siggraph Course **8** (2006)
[37] C. Foudas, P. Katsoulis, T. Lama, S. Mallios, G. Karathanasis, I. Papavergou, S. Regnard, M. Tepper, P. Sphicas, C. Vellidis et al., *Upgrade of the CMS Barrel Muon Track Finder for HL-LHC featuring a Kalman Filter algorithm and an ATCA Host Processor with Ultrascale+ FPGAs* (2019)
[38] G. Hinton, O. Vinyals, J. Dean (2015), 1503.02531
[39] C.N. Coelho, A. Kuusela, S. Li, H. Zhuang, J. Ngadiuba, T.K. Aarrestad, V. Loncar, M. Pierini, A.A. Pol, S. Summers, Nature Machine Intelligence **3**, 675 (2021)
[40] S. Vadera, S. Ameen (2021), 2011.00241
[41] R.M. Schmidt (2019), 1912.05911
[42] S. Hochreiter, J. Schmidhuber, Neural computation **9**, 1735 (1997)
[43] A.F. Agarap, *Deep Learning using Rectified Linear Units (ReLU)* (2019), 1803.08375
[44] D.A. Clevert, T. Unterthiner, S. Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)* (2016), 1511.07289
[45] D. Hendrycks, K. Gimpel, *Gaussian Error Linear Units (GELUs)* (2023), 1606.08415
[46] S. Ioffe, C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift* (2015), 1502.03167
[47] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors* (2012), 1207.0580
[48] D.P. Kingma, J. Ba, *Adam: A method for stochastic optimization* (2017), 1412.6980
[49] O. Kitouni, N. Nolte, M. Williams, Mach. Learn. Sci. Tech. **4**, 035020 (2023), 2112.00038
[50] K. O'Shea, R. Nash, *An introduction to convolutional neural networks* (2015), 1511.08458