UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia “Galileo Galilei”
Master Degree in Physics of Data

Final Dissertation

On the measurement of the $W \to 3\pi$ decay with the Phase-2 L1 scouting system at CMS

Thesis Supervisor:
Dr. Giovanni Petrucciani

Candidate:
Pietro Cappelli

Thesis co-Supervisor:
Prof. Marco Zanetti
Prof. Jacopo Pazzini

Academic Year 2022/2023

Abstract

The era of exploring the subatomic world through particle accelerators has reached a new pinnacle with the Large Hadron Collider (LHC) at CERN. This groundbreaking facility has transformed our comprehension of the fundamental fabric of the universe by facilitating collisions between proton bunches (or heavy ions) at energies reaching up to 13.6 TeV. These high-energy collisions have led to momentous discoveries, notably the confirmation of the Higgs boson's existence. However, the quest for answers to still-unresolved questions in the field of particle physics is an ongoing evolution. In this context of unprecedented scientific excitement, the particle physics community is preparing for the next monumental phase: the upgrade to the LHC known as the High-Luminosity LHC (HL-LHC). The HL-LHC is designed to open new frontiers of exploration, enabling high-luminosity data acquisition and experiments with unprecedented precision.

The Compact Muon Solenoid (CMS) at the Large Hadron Collider (LHC) stands as a remarkable testament to human ingenuity and collaborative scientific endeavor. It is one of the main LHC detector and one of the two general-purpose experiment. CMS played a pivotal role in the discovery of the Standard Model (SM) Higgs boson, a goal envisioned since its design phase. It has a broad physics program ranging from the study of the SM to searching for novel interactions and particles beyond the Standard Model. This huge and incredibly sophisticated detector relies on a two-stage trigger system to select the most interesting collision events for read-out and analysis. The first of these, the level 1 (L1) trigger receives only a sub-set of the detector data at each bunch crossing (BX), and must determine whether to trigger the detector for the full readout within around 3 μs.

The L1 data scouting (L1DS) system is a novel and innovative data collection approach that acquires intermediate data from the L1 trigger at the full bunch crossing rate of 40 MHz. This unique capability allows for analysis of event types that occur too frequent to be part of the nominal L1 menu. Notably, the scouting system operates independently of the L1 trigger, ensuring that it does not influence the trigger decision-making process. In order to determine the importance of a particular event, the L1 trigger rejects thousand of events. The scouting system assumes a pivotal role in avoiding the rejection of vast amounts of potentially valuable data containing interesting signatures.

Among the numerous upgrades that CMS will undergo in preparation for Phase

2 (HL-LHC), a particularly intriguing development is the ability to run reconstruction algorithms such as Particle Flow and PUPPI directly at Level 1. This paves the way for innovative strategies in selecting interesting events by leveraging the data provided by the L1 scouting system.

This thesis introduces for the first time a demonstration for the feasibility of performing an event selection using the data coming from the scouting system at the full bunch crossing rate of 40 MHz, within the conditions of the HL-LHC era. Referred to as the *online selection*, this approach underscores the capability to perform event selection in real-time within the LHC operational rate, specifically utilizing the L1 scouting data. In particular, the online selection focuses on a rare decay of the W boson to three charged pions ($W \to 3\pi$). This rare decay was already investigated for the first time using the Run2 data collected by CMS from 2016 to 2018. The exclusive decay of the W boson to hadrons would provide a new precision measurement of the W boson mass or would serve for an unobserved probe of the strong interaction at the boundary level of perturbative and non-perturbative domains of the QCD (Quantum Cromodyamics). The analysis of this rare decay reveled no significant excess above the background observation, highlighting the statistical limitation of the process. Given these considerations, this particular rare decay fits very well the characteristic for an online selection. Consequently, the objective is to identify events featuring a triplet of pions, serving as decay product candidates of a W boson.

To outline the developed work, this thesis commences by providing an introductory overview of the background context of online selection in Chapter 1. This includes an exploration of the key features of the CMS experiment, with a specific emphasis on the pertinent characteristics for the conducted study, namely the L1 trigger and the scouting system. Following this, Chapter 2 delves on an illustrative example of the W into 3π analysis, through the phase2 MC simulation. The primary goal is to formulate a selection algorithm tailored for event selection, culminating in the estimation of an upper limit on the branching fraction of the decay at a 95% confidence level. Subsequently, Chapter 3 provides an in-depth description of the real demonstrator, intricately detailing the processing chain essential for the implementation of online selection. This chapter systematically breaks down the construction of each step, offering insights into the hardware environment that accommodates the demonstrator. Finally, Chapter 4 illustrates the integration of all the steps, providing a comprehensive depiction on how each component fits together. The chapter also unveils the results obtained from the constructed prototype of the online selection, offering a comprehensive view of the developed system.

Contents

Abstract **i**

1 The CMS experiment at the CERN LHC **1**

1.1 Large Hadron Collider at CERN 1

1.1.1 Particles in a collision 2

1.1.2 Observables in an events study 3

1.2 CMS Experiment 5

1.2.1 Detector 5

1.3 Data acquisition system 7

1.3.1 Level-1 Trigger 8

1.3.2 High-Level Trigger 8

1.3.3 Event reconstruction 9

1.4 The Phase-2 upgrade 11

1.4.1 The Phase-2 upgrade of Level1-Trigger 12

1.5 The scouting system or Data scouting at the CMS experiment . . . 14

1.5.1 Scouting at the level 1 15

2 The W boson rare decay to 3 charged pions analysis **17**

2.1 State-of-the-art analysis 17

2.2 W to 3 pions selection 18

2.2.1 Cutting filters 19

2.2.2 Selection on MC simulation 20

2.2.3 Matched pions 21

2.2.4 Feature resolution and reconstruction efficiency 23

2.2.5 Parameter tuning 27

2.3 Signal and Background events estimate 31

2.3.1 Signal-to-noise ratio 34

2.4 Set of an Upper Limit on the branching ratio 35

2.4.1 Upper limit statistical concepts 35

2.4.2 Final results 37

3 The Online Selection Demonstrator 39
3.1 Correlator layer 2 emulator 40
3.1.1 System Description 40
3.1.2 Puppi row format 42
3.1.3 Feature distribution study 44
3.1.4 Features Generation 46
3.1.5 Event generation 50
3.2 Receiving and Unpacking 51
3.2.1 Unpacking timing performance results 52
3.3 Selection 55
3.3.1 Selection optimization 56
3.3.2 Selection timing performance 57

4 Live analysis and Results 61
4.1 Setup for online selection 61
4.2 Running the online selection 63
4.2.1 Final Results 63

Conclusions 65

Appendix A 67

Appendix B 71

Bibliography 75

Acknowledgments 79

Chapter 1

The CMS experiment at the CERN LHC

The study of high-energy particle physics has brought us to the forefront of understanding the fundamental constituents of the universe. One of the pioneering endeavors in this field is the CMS (Compact Muon Solenoid) experiment at CERN (European Organization for Nuclear Research), situated deep underground on the Franco-Swiss border. At the heart of CMS lies the Large Hadron Collider (LHC), the world's most powerful particle accelerator.

The LHC recreate conditions similar to those just moments after the Big Bang by colliding protons at very large energy, allowing physicists to explore the fundamental forces and particles that govern the universe's behavior.

Within this extraordinary facility, the CMS experiment stands as a crucial component. Its massive detectors and sophisticated technology enable the observation and measurement of particles produced by these high-energy collisions.

This chapter serves as an introduction to the CMS experiment and the LHC, providing a comprehensive overview of their significance in the world of particle physics. It sets the stage for the subsequent analysis, which focuses on the utilization of Level 1 trigger data from CMS in our quest to understand the universe's fundamental building blocks.

1.1 Large Hadron Collider at CERN

The Large Hadron Collider is a superconducting circular proton-proton collider located in Geneva, Switzerland, operating since 2010 at the laboratories of the European Organization for Nuclear Research (CERN) [1–4]. It is the largest and most powerful accelerator built up to now. It consists of an underground 27-kilometer ring of superconducting magnets and detectors, designed to produce proton-proton collisions at unprecedented center of mass energies, reaching a maximum value of $\sqrt{s} = 14$ TeV, and collision rates. In particular, its current instantaneous luminosity is of the order of $\mathcal{L} = 2 \times 10^{34}$ cm^2 s^{-1}. Inside the LHC ring, two high-energy particle beams travel close to the speed of lights and in opposite directions, i.e. clockwise and anti-clockwise. This relativistic regime is progressively reached through different steps. The acceleration chain, which is shown in the scheme in fig. 1.1, allows the beam to reach those energies, thanks to superconducting magnets, which are cooled to 2.1 K using superfluid helium. The cooling system allows for a magnetic field of 8.3 T, which bends the beam and allows it to circulate at high energies within the ring.

The two beams are made of bunches of protons injected in the collider and cross at a frequency of $f_{\text{BX}} = 40$ MHz, i.e., every 25 ns. The nominal beam structure is

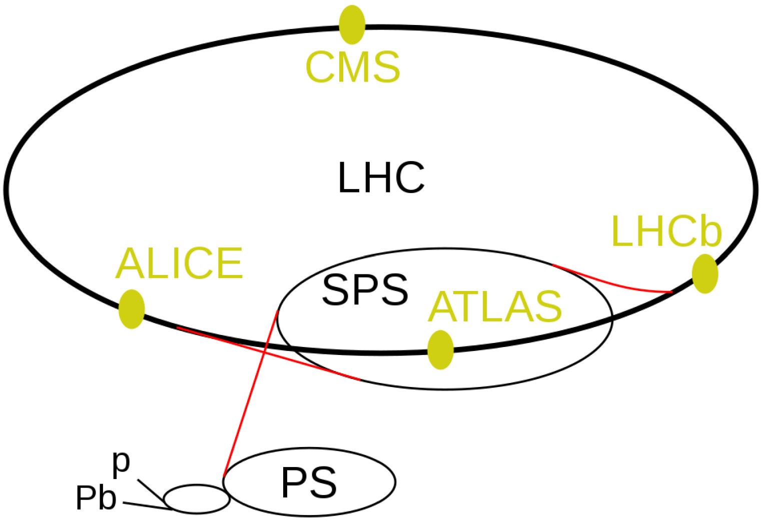


Figure 1.1: *General scheme of the acceleration chain of the process of proton (of heavy ion) injection into the LHC. After a linear acceleration process and before reaching LHC, the beam is accelerated thanks to PS (Proton Synchrotron) and SPS (Super Proton Synchrotron). This cascade ensures an injection energy into the LHC of* 450 *GeV*

depicted in fig. 1.2. Each beam is carefully collimated to enhance the probability of collisions. Just before reaching the selected collision points, a specialized magnet is employed to 'squeeze' the protons closer together, significantly amplifying the likelihood of collision events. On the LHC ring, there are four main collision points, each hosting one of the four primary detector experiments: CMS [5] and ATLAS [6], which are the two general-purpose experiments, and LHCb [7] and ALICE [8], each designed for specialized investigations. In the subsequent sections, a deeper description of the CMS experiments will be provided.

1.1.1 Particles in a collision

As mentioned earlier, all LHC experiments must contend with a high volume of collisions occurring at each proton bunch crossing, largely due to the elevated instantaneous luminosity. This phenomenon is closely linked to the number of protons within each bunch and is increased by the beam squeezing process, which enhances the probability

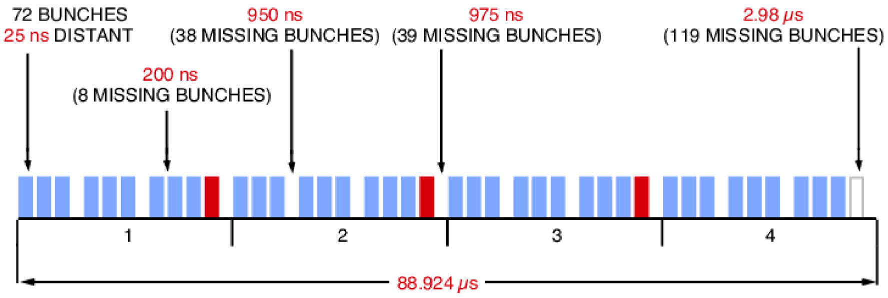


Figure 1.2: *LHC beam structure scheme*

of proton-proton interactions[1].

The average number of proton-proton collisions occurring per bunch crossing is referred to as "pileup," and it poses a significant challenge to address. When two proton bunches collide, a multitude of collisions overlap within the detector, surrounding the primary hard particle collision that is the focus of our study. For instance, the average pileup for the CMS experiment during the 2018 data acquisition period was $\langle \mu \rangle = 37$. More comprehensive information about the pileup distribution and the integrated luminosity across all Run 2 data acquisition years can be found in fig. 1.3 [11].

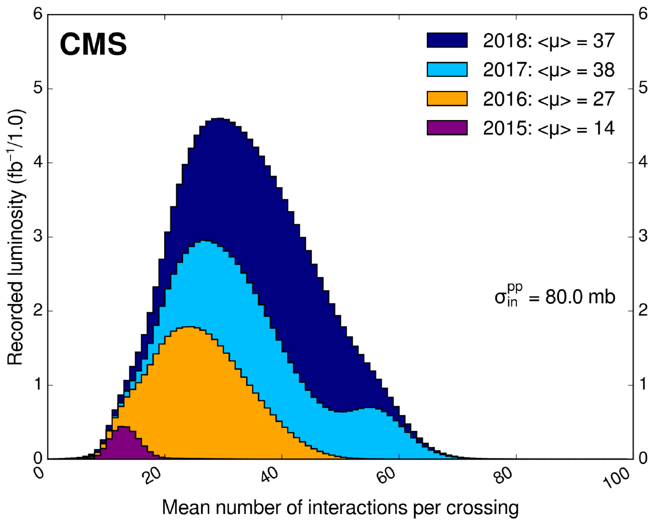


Figure 1.3: *Interactions per crossing (pileup) for 2015-2018. Distribution of the average number of interactions per crossing (pileup) for pp collisions in 2015 (purple), 2016 (orange), 2017 (light blue), and 2018 (navy blue). The overall mean values and the minimum bias cross sections are also shown.*

1.1.2 Observables in an events study

Before going into a concise description of the CMS detector, it's important to introduce the general features and the observable measured in such high energy physics experiments, describing the coordinate system of the detector too. The latter, as depicted in fig. 1.4a, is right-handed and is defined as follows: the x-axis directs towards the accelerator ring center, the y-axis ascends upwards, while the z-axis runs parallel to both the beam pipe and the solenoid's magnetic field. Moreover, in the same figure, the momentum direction of a product particle is drawn as well as some relevant kinematic features, which are usually taken into account in the high energy physics analysis. In particular, as well as in many other high-energy physics experiments, it is common to measure the transverse component of momentum because of the nature of the collisions and the design of the detectors, namely p_T:

$$p_T = \sqrt{p_x^2 + p_y^2}, \tag{1.1}$$

which in case of massless particles coincides with the transverse energy E_T.

On the other hand, regarding the angular information, the azimutal angle ϕ and the pseudorapidity (η) are often measured. The latter is important because it provides several

[1] The proton-proton cross-section at the LHC energy of $\sqrt{s} = 14$ TeV is on the order of $\sigma_{pp} = 100$ mb, and it is influenced by various characteristics of the proton beam. Further details about cross-section measurements can be found in references [9, 10]

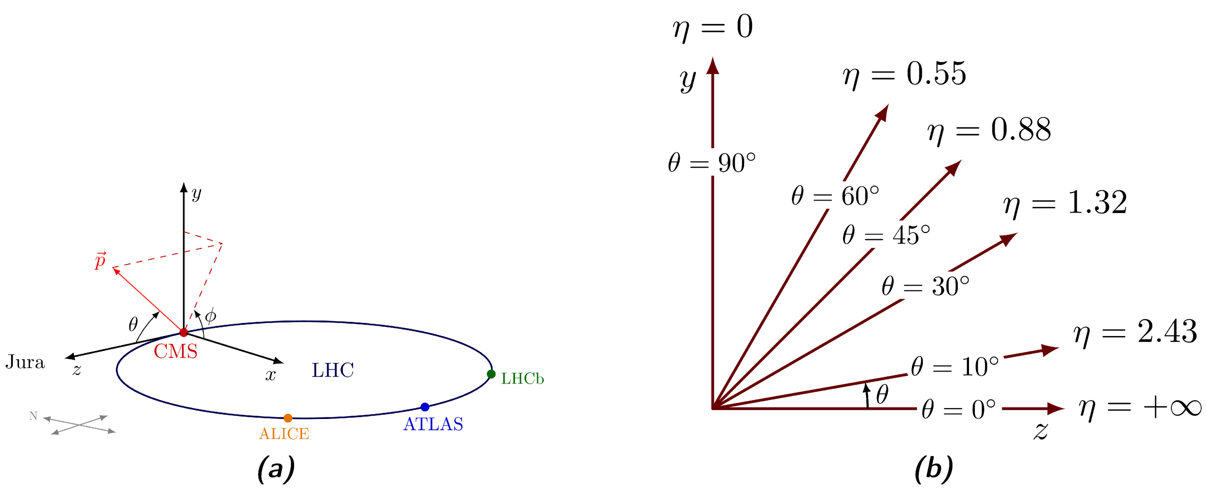


Figure 1.4: *Left:CMS coordinate system in the LHC ring. Right: pseudorapidity values with respect to the angle θ*

advantages over other angle information, such as polar angle. First of all, pseudorapidity (η) is closely related to rapidity (y), which is a relativistic quantity, being also a good approximation of it. It is advantageous because rapidity differences are invariant under Lorentz boosts. In high-energy particle collisions, particles can have a wide range of velocities, and the rapidity variable is useful for comparing particles' kinematic properties across different energy regimes. It also allows for a more uniform angular coverage. In many experiments, detectors are designed to have a good coverage in η, which ensures that particles traveling in various directions can be detected with roughly equal efficiency. Finally, η provides a geometric interpretation that's useful for detectors. It is defined as:

$$\eta = -\ln[\tan(\frac{\theta}{2})], \tag{1.2}$$

where θ is the polar angle, so its values are related to the angles at which particles are emitted relative to the beamline.

A useful quantity usually employed to express the angular distance between two particles is ΔR, namely:

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{1.3}$$

where:

$$\Delta\phi = \phi_2 - \phi_1, \tag{1.4}$$

$$\Delta\eta = \eta_2 - \eta_1 \tag{1.5}$$

with the two indices referring to the two different particle taking into account.

Finally, in the context of a generic n-particle collision one additional crucial parameter that can be readily calculated is the invariant mass (m):

$$m = \sqrt{\left(\sum_{i=1}^{n} E_i\right)^2 - \left\|\sum_{i=1}^{n} \vec{p_i}\right\|^2}, \tag{1.6}$$

where E_i is the energy of the i^{th} particle.

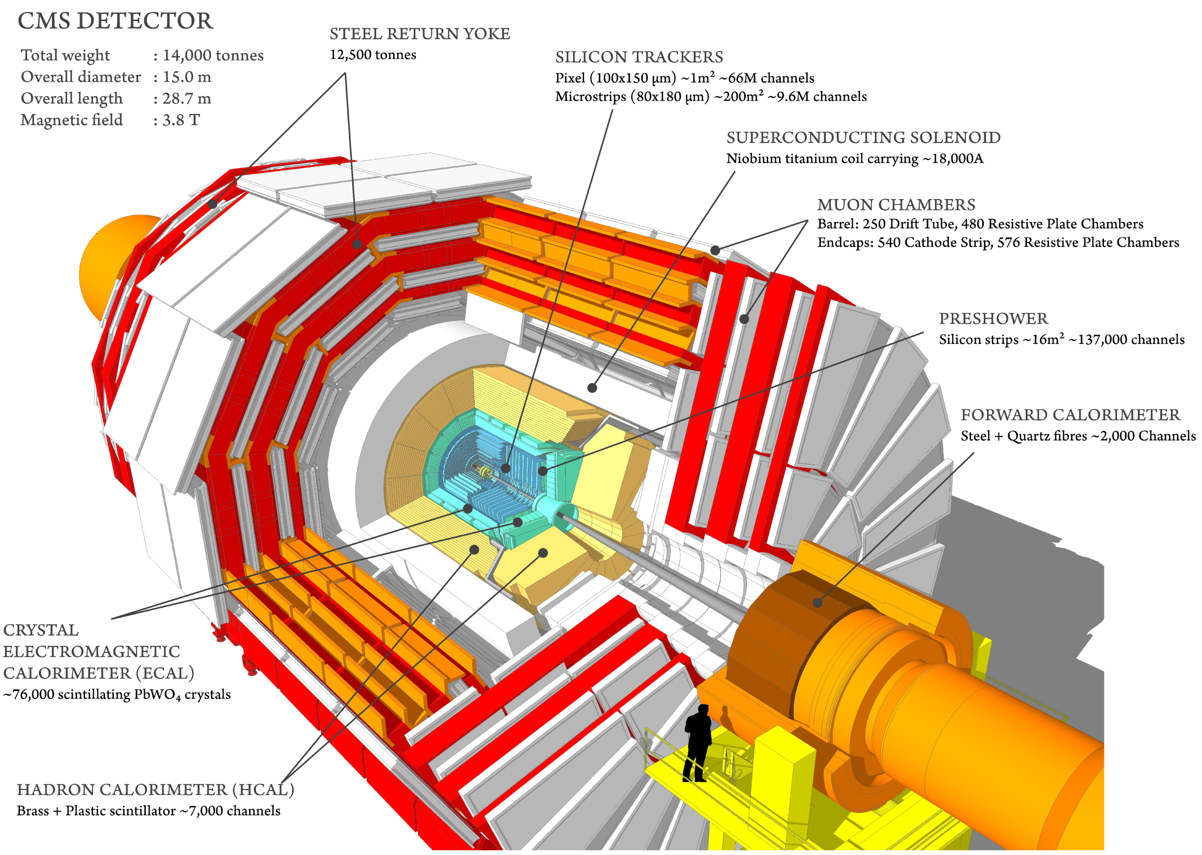

CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm) ~1m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 540 Cathode Strip, 576 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO₄ crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

Figure 1.5: *3D model of CMS Detector from [12]*

1.2 CMS Experiment

The Compact Muon Solenoid experiment [5] is one of the two general-purpose experiment, designed to search for new physics. The discovery of the SM Higgs boson has been one of the primary goals of CMS already since its design phase.

1.2.1 Detector

The CMS detector is build around a huge solenoid magnet, which is shaped like a cylindrical coil of superconducting cable that generates a field of 3.8 T in its inner region, i.e., 10^5 times the Earth magnetic field. As most of high energy physics detector, it is built in a cylindrical shape with a barrel and two end-cap regions. The CMS detector is a remarkable feat of engineering, compactly packed with advanced materials and technologies, measuring 21 meters in length and 14 meters in width. Comprising several concentric layers, this cylindrical marvel is depicted in 3D detail in Figure 1.5 and offers an octane view of its transverse section in Figure 1.6. Each component is meticulously designed to capture every detail of collision events, deciphering the properties of particles generated in these high-energy interactions. This feat is accomplished through several critical processes:

- Trajectory Bending: Charged particles, emerging from the collision point, have their paths bent by the solenoidal magnet. This bending is instrumental in determining particle charge and measuring momentum. Positively and negatively charged particles bend in opposite directions within the magnetic field, with high-momentum particles displaying less bending compared to their low-momentum counterparts.

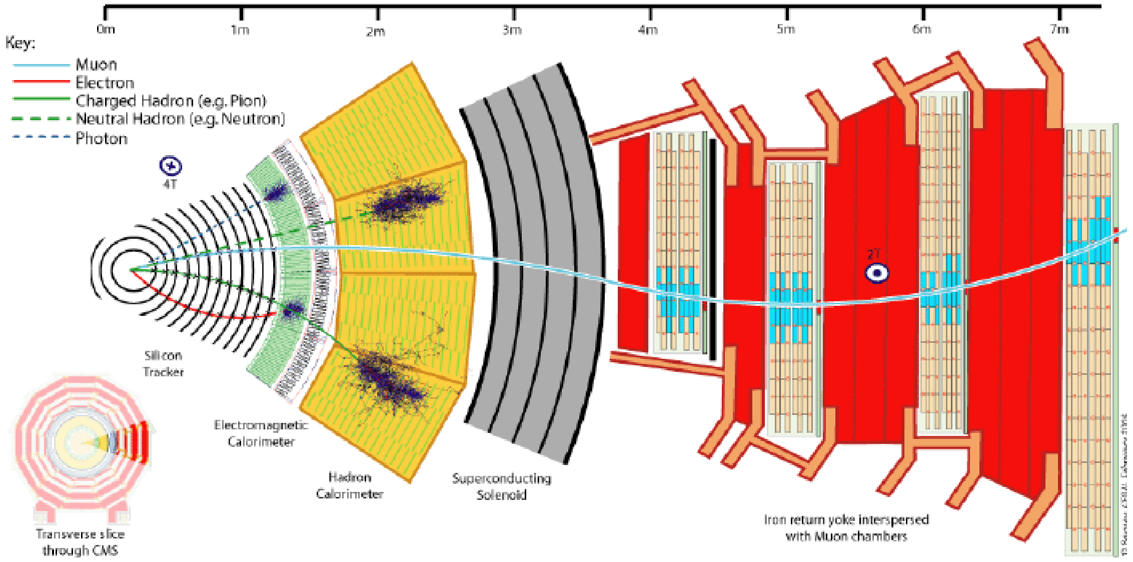

0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS

Figure 1.6: *Slice of the CMS detector with sample particles that can be detected.*

- Track Identification: The precise identification of particle trajectories, known as tracks, is achieved through the Silicon Pixel and Tracker systems, as well as the muon detectors. This precision is essential for understanding particle behavior.

- Energy Measurement: Understanding the energy of resultant particles is crucial for decoding the collision process. This information is captured by two types of calorimeter detectors. The Electromagnetic Calorimeter (ECAL) halts electrons and photons, measuring their energy entirely. Other particles, such as hadrons, traverse the ECAL and are stopped in the Hadron Calorimeter (HCAL), which envelops the ECAL.

- Muon Detection: Muons, unlike other particles, are not arrested by calorimeters. Their momentum can be measured both inside the superconducting coil through tracking devices and outside of it using the muon chambers.

Thanks to the ensemble of all these components, the energy and momentum of the collision products can be detected, with the exceptions of "undetectable" particles (such as neutrinos), revealed by their missing energy.

A concise breakdown of the CMS detector's structure, from its innermost to its outermost components, is as follows:

- The **Inner Silicon tracker** [13], composed of silicon pixel and silicon strip sensors, is located at the center of the detector. Its sensors are sensitive to charged particles that pass through, producing a detectable electric signal, which can be used to reconstruct the trajectories of those particles. Its coverage extends up to $|\eta| = 2.5$

- The **electromagnetic calorimiter (ECAL)**, composed by lead-tungsten crystals and the **hadronic calorimiter (HCAL)**, made of layers of brass and scintillator plates, are located between the tracker and the solenoid. While the ECAL [14] measures the energy of the electromagnetic component of the collision products, namely electrons and photons, the HCAL [15] measures the energy of the hadronic one, namely neutral and charged hadrons. The coverage of ECAL and HCAL extends up to $|\eta| = 3$, and is complemented in the region $3 < |\eta| < 5$ by a forward calorimeter (HF), made of quartz and scintillating fibers;

- The **superconducting solenoid** [16] is placed between the HCAL and the muon chambers, it produces a strong magnetic field of 3.8T within its core and a 2T in its flux return yoke;

- the **muon spectrometer** is located in the steel return yoke of the magnet, and uses three technologies of gas detectors: drift tubes (DT), cathode strip chambers (CSC) and resistive plate chambers (RPC), with a coverage up to $\eta = 2.4$.

1.3 Data acquisition system

The CMS experiment, along with the other LHC experiments, faces the challenge of managing a high volume of collision events occurring within the detector at extremely high rates, coupled with a substantial instantaneous luminosity. This poses significant constraints on the Data Acquisition System (DAQ). The limited storage capacity prohibits the retention of all collision events, including all the particles generated in each bunch crossing.

Even if storage were theoretically feasible, a significant portion of the events would not be relevant for analysis due to the prevalence of backgrounds. To address this issue, it can be useful to estimate the maximum data throughput that can be accommodated. Considering the fine-grained segmentation of the CMS detector, a bunch crossing frequency of 40 MHz, and the number of events per bunch crossing, the raw data rate would be approximately 40 terabytes per second, assuming an average data size of 1 megabyte per collision event. Such a rate would quickly saturate permanent storage resources.

To mitigate this challenge, a sophisticated trigger system is indispensable. It serves the dual purpose of selecting potentially interesting events while also reducing the data flow to a manageable level.

CMS employs a two-level trigger strategy, comprising the **Level 1 Trigger** (**L1T**) [18] and the **High-Level Trigger** (**HLT**) [19]. The L1T utilizes data from the calorimeters and muon detectors to swiftly identify the most interesting events in less than 4 microsecond, resulting in an output event rate of approximately 100kHz, effectively reducing the event rate from the LHC clock's 40MHz.

Following the L1T, the High-Level Trigger (HLT) further refines the reconstruction of events chosen by the L1T, ultimately narrowing the event rate to approximately 1kHz, marking the final selection before data storage. The HLT relies on a processor farm that operates using the same software framework employed for offline reconstruction, drawing information from all detector components. A schematic overview of the trigger chain can be seen on the right, in Fig.1.7.

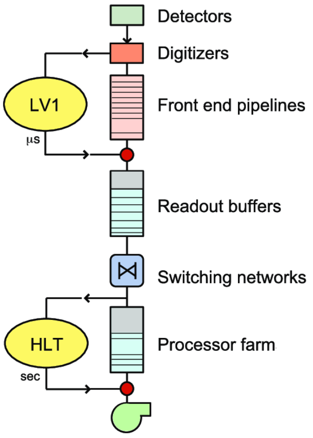


Figure 1.7: *Schematic view of the CMS trigger system from [17].*

While the entire detector readout is accessible at the HLT stage, it is essential to meet the stringent timing requirements dictated by the input rate from L1. Consequently, events are selectively discarded as soon as sufficient information becomes available to make a decision. This approach allows, for instance, the utilization of full track reconstruction exclusively for events that cannot be rejected based on information from the calorimeters or the faster pixel-only track reconstruction.

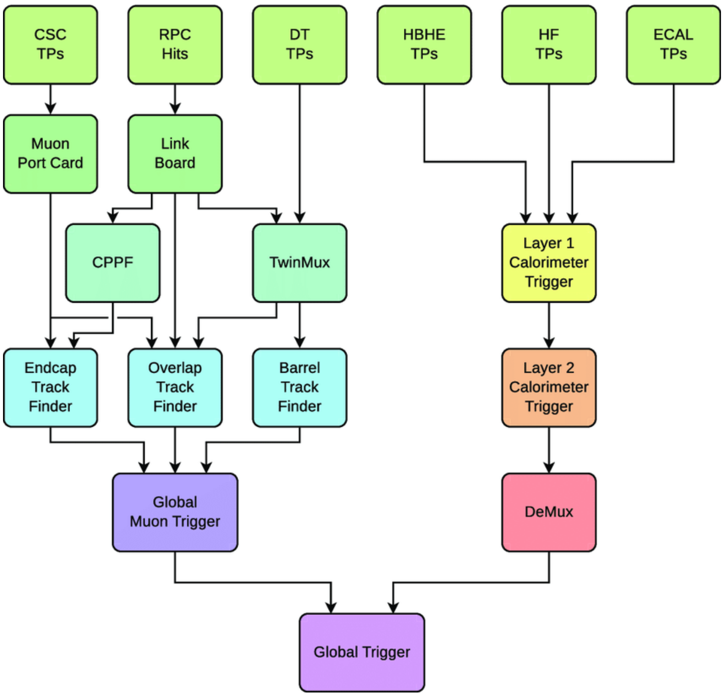


Figure 1.8: *Diagram of the CMS Level-1 trigger system during Phase-1 from [20]. CSC: Cathode Strip Chambers; RPC: Resistive Plate Chambers; DT: Drift Tubes; HO: Hadronic Calorimeter-Outer; HF: Hadronic Calorimeter-Forward; ECAL: Electromagnetic Calorimiter; TPs: Trigger Primitives; CPPF: Concentration, Pre-Processing and Fan-out system.*

1.3.1 Level-1 Trigger

The L1 Trigger[18], depicted in fig.1.8 is divided into three major subsystems: the **muon trigger**, the **calorimiter trigger** and the **global trigger**. Moreover, the whole system is organized in a multilevel architecture, with Local Triggers or Trigger Primitive Generators (TPG) and Regional Triggers. The former performs a local reconstruction using a small fraction of sub-detectors, such as muon chambers; while the latter reconstructs higher level objects by ,atching information from different local triggers. Then, the global calorimiter and the muon triggers combine the information from regional ones. Finally, the Global Trigger (**GT**) select the events relying on the all the information coming from all the previous layers.

As previously said, the L1T necessitate rapid evaluations of every bunch crossing (BX). The system need to perform a non-trivial algorithmic assessments quickly. The trigger logic, segmenting its evaluations into steps, is pipelined such that it can accept data from a new BX every 25ns. To achieve this, custom-programmable hardware, such as Field Programmable Gate Arrays (FPGA) and Programmable Lookup Tables (LUTs), are used. Currently, the L1T effectively processed approximately 5TB/s of data, reducing the detector read-out rate from 40MHz to a fixed 100kHz.

1.3.2 High-Level Trigger

While the Level 1 Trigger (L1T) is built upon FPGAs and Application-Specific Integrated Circuits (ASICs) to execute fast and relatively straightforward trigger algorithms, the High-Level Trigger (HLT)[19] relies on software implementation. It employs the same offline reconstruction techniques to ensure maximum flexibility and operates on a cluster of approximately 16,000 CPU cores within a farm of commercial computers.

The software is meticulously optimized to meet the real-time processing requirements of online event selection. HLT processes are structured into 'paths', each representing a systematic progression involving the selection of specific physics objects or combinations through stages of reconstruction and filtering. These paths are composed of sequences of producers and filters, arranged in an organized hierarchy based on computational complexity. Initial rapid algorithms are given priority, and their output is subsequently subjected to filtering. Any filter failure results in the cancellation of subsequent, more computationally intensive algorithms. The final HLT decision is derived from the logical OR combination of all the trigger paths.

1.3.3 Event reconstruction

After the selection performed by the two-level trigger, it is necessary to identify and reconstruct all the particles inside the selected event, i.e. in the raw data describing the event. The output of this step is the so called RECO data-tier, which provides access to uncalibrated reconstructed physics objects for physics analyses in a more convenient format. To perform it, first the information of the hits and the calorimeter towers are obtained from the unpacked detector data. These information are used to reconstruct the global tracks including also the hits in the silicon tracker and in the muon detectors (1.2), through pattern recognition algorithm. Then, primary and secondary vertex[2] candidates are recognised. Finally, two algorithm are applied: **Particle Flow** (**PF**), which is a particle identification algorithm to identify standard physics objects candidates; **Pileup per particle identification algorithm** (**PUPPI**), which is used for pileup mitigation. While the former has been always used after the collection of the data since run 1, the latter has different usage depending on the LHC run.

1.3.3.1 Particle Flow algorithm

The Particle Flow algorithm [21, 22] is a fundamental component of CMS reconstruction. It is designed to individually identify and reconstruct each particle originating from collision events by correlating data from all subdetectors. This distinctive feature enhances the performance of jet and MET (missing transverse energy) reconstruction and enables the accurate identification of electrons, muons, and taus.

The Particle Flow algorithm capitalizes on the unique characteristics of each particle to be reconstructed and their corresponding signatures within CMS subdetectors. It relies on an efficient linking procedure to connect these deposits, a pure track reconstruction, and a clustering algorithm capable of disentangling overlapping energy showers. A simplified overview of the algorithm is presented here, with a more comprehensive description available in [21], while a scheme of the PF algorithm can be found in fig. 1.9.

Photon and electrons When a photon or an electron traverses the Electromagnetic Calorimeter (ECAL), they lose energy by initiating an electromagnetic shower. Photons primarily lose energy through electron-positron emission, while electrons lose energy through Bremsstrahlung photon emission. Clustering algorithms combine energy deposits collected by ECAL crystals from the same shower to recover the radiated energy. Subsequently, the crystals participating in the shower are merged into a "super-cluster,"

[2] The primary vertex marks the initial collision point in the collider, representing the primary interaction point. On the other hand, a secondary vertex is formed where new particles are generated or where particles from the primary collision decay. Precisely determining these vertices is crucial for accurately reconstructing the tracks of produced particles.

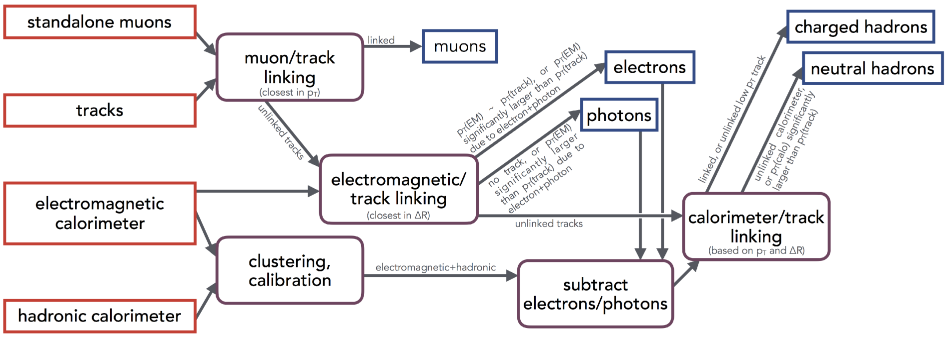

standalone muons
tracks
electromagnetic calorimeter
hadronic calorimeter
muon/track linking
(closest in p_T)
linked
muons
unlinked tracks
electromagnetic/ track linking
(closest in ΔR)
clustering, calibration
electromagnetic+hadronic
p_T(EM) ~ p_T(track), or p_T(EM) significantly larger than p_T(track) due to electron+photon
electrons
no track, or p_T(EM) significantly larger than p_T(track) due to electron+photon
photons
unlinked tracks
subtract electrons/photons
calorimeter/track linking
(based on p_T and ΔR)
linked, or unlinked low p_T track
charged hadrons
unlinked calorimeter, or p_T(calo) significantly larger than p_T(track)
neutral hadrons

Figure 1.9: *Schematic of PF algorithm from [23]*

and the photon's energy is computed using the signals detected by the crystals within the super-cluster. In the case of photons, energy deposits should not be associated with hits in the inner tracker. The ECAL's fine granularity, combined with the presence of a strong magnetic field, enables effective separation of photon energy deposits from those of charged particles. Electron reconstruction, on the other hand, necessitates a combination of an energy cluster in the ECAL and a track in the silicon tracker. The electron track must also satisfy specific criteria related to the transverse and longitudinal impact parameters concerning the electron's primary vertex, and it should not have more than one missing hit in the internal layers of the tracker.

Muons Muons are reconstructed and identified with very large efficiency and purity from a combination of the tracker and muon chamber information. The energy deposits in the calorimeter may be checked for compatibility with the muon hypothesis.

Charged and neutral hadrons The Hadron Calorimeter (HCAL) is responsible for capturing the energy deposits of both charged and neutral hadrons. Unlike the finer granularity of the Electromagnetic Calorimeter (ECAL), the HCAL's coarser granularity does not facilitate the spatial separation of charged and neutral hadrons within jets.

- Charged Hadrons: Charged hadrons are identified as tracks of charged particles that are not recognized as electrons or muons. Their energy is determined by combining the track's momentum with the corresponding energy deposits in the ECAL and HCAL. These values are corrected for the calorimeters' response function to hadronic showers.

- Neutral Hadrons: Neutral hadrons are identified as HCAL energy clusters not associated with any charged hadron trajectory. They can also be recognized through a combined excess of energy in the ECAL and HCAL compared to the expected energy deposit of charged hadrons.

The resulting list of reconstructed particles includes charged hadrons, photons, neutral hadrons, electrons, and muons. This comprehensive particle list is instrumental in various aspects of the analysis, such as jet reconstruction, determination of missing transverse energy, identification of tau leptons from their decay products, and measurement of particle isolation. The outstanding performance of the Particle Flow (PF) algorithm, which underpins these processes, is well-documented in [22].

The CMS detector is exceptionally well-suited for these tasks due to its various components, which provide excellent energy resolution and the ability to reconstruct particle tracks. This precision is pivotal in the successful implementation of the PF algorithm,

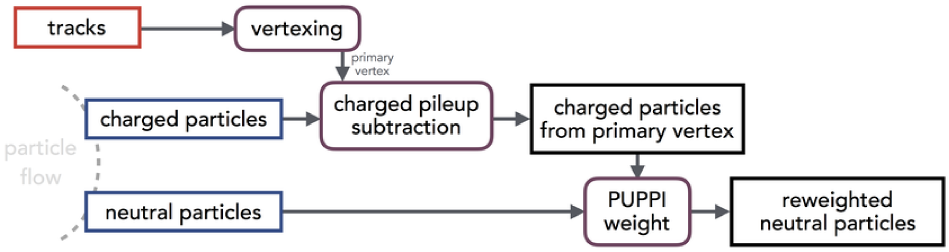

tracks
vertexing
primary vertex
charged particles
charged pileup subtraction
charged particles from primary vertex
particle flow
neutral particles
PUPPI weight
reweighted neutral particles

Figure 1.10: *Schematic of PUPPI pileup mitigation algorithm from [23]*

making CMS an indispensable asset in the feasibility of particle-flow event reconstruction.

1.3.3.2 Pileup per particle identification

Pileup, which refers to the presence of overlapping secondary proton-proton collisions in addition to the primary interaction, poses a significant challenge for the LHC's luminosity. This challenge becomes even more pronounced with the upgrade of the LHC, which will be described in sec.1.4. To address this issue, the Pileup per Particle Identification algorithm[24] has been developed as a means of mitigating pileup effects.

As previously anticipated, higher luminosity implies higher number of simultaneously collisions during the same BX. This creates the need to distinguish the primary interaction of interest from the additional "pileup" collisions to ensure effective physics performance. In the CMS Collaboration, various techniques have been devised to mitigate the impact of pileup collisions[25]. These methods include charged-hadron subtraction, pileup jet identification, isospin-based neutral particle "$\delta\beta$" correction, and, most recently, pileup per particle identification, as introduced by D. Bertolini et al.[24]. The primary goal of these techniques is to reconstruct and distinguish particles originating from the primary vertex, or primary interaction, from those arising from secondary interactions or pileup. In the context of the PUPPI algorithm, these particles are referred to as "PUPPI candidates" and play a crucial role in the analysis of all events.

The PUPPI algorithm assigns a local shape parameter α to each particle, which serves to probe the collinear versus soft diffuse structure in the vicinity of the particle. Collinear structures indicate particles originating from the hard scatter, while soft diffuse structures indicate particles originating from pileup interactions. An event-specific weight for each particle is computed by using the distribution of α for charged pileup, which acts as a proxy for all pileup. These weights reflect the degree to which particles resemble pileup particles and are employed to rescale their four-momenta. This innovative approach represents a significant advancement in pileup mitigation strategies. A scheme of the Puppi algorithm can be found in fig. 1.10.

1.4 The Phase-2 upgrade

The High-Luminosity LHC (HL-LHC), scheduled to commence operations in 2030[26], represents a major upgrade of the CERN accelerator. This upgraded facility is expected to increase the delivered instantaneous luminosity to $\mathcal{L} = 5\times10^{34}$ cm^{-2}s^{-1}, i.e. five times the accelerator's original design value, reaching values of $\mathcal{L} = 7.5 \times 10^{34}$ cm^{-2}s^{-1} in its "ultimate" configuration. The upgraded detectors will also need to handle increased pile-up, reaching up to an average of $<\mu>= 200$. Under these parameters, it will be possible to achieve a performance of approximately 350 fb^{-1} if the number of proton

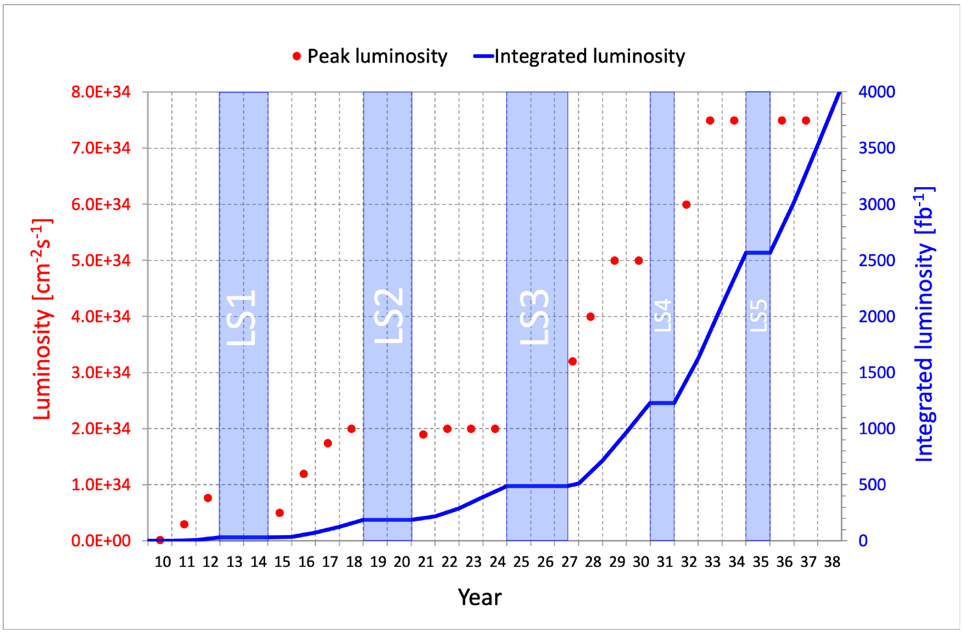


Figure 1.11: *Forecast for peak luminosity (red dots) and integrated luminosity (blue line) in all the LHC era with also the ultimate HL-LHC parameters, from [33]*

physics days per year can be extended beyond LS4[3] and LS5. This could potentially lead to an accumulation of up to 4000 fb^{-1} before 2040, as illustrated in Figure 1.11.

The High-Luminosity LHC (HL-LHC) is expected to provide an unprecedented opportunity to explore the weak-scale nature of the universe. It will enable high-precision measurements of the standard model (SM) electroweak interaction, including properties of the Higgs Boson, and facilitate searches for new physics beyond the standard model (BSM). This may include investigations into weak-scale couplings and possible explanations for phenomena such as the observed gauge hierarchy or the quantum nature of dark matter.

The increase in instantaneous luminosity and pile-up will result in a very high particle multiplicity. Therefore, the CMS detector is undergoing extensive consolidation and modifications across all its components, previously described in Section 1.2. These enhancements are outlined in the Technical Design Reports[27–32]. The objective is to ensure the detector's robustness in the face of significant hadronic background, making the two-level trigger system more efficient.

While the two-level strategy of CMS remains consistent for Phase-2, the entire trigger and data acquisition (DAQ) system will be overhauled. Substantial improvements will be implemented at both trigger levels, focusing on timing and logic. The detector readout electronics and DAQ will be upgraded to accommodate a maximum L1A (Level-1 Acceptance) rate of 750kHz and a latency of 12.5μs (or 500 LHC bunch crossings). Additionally, for the first time, the L1 trigger will include tracking information and high-granularity calorimeter information. This chapter will focus briefly only on the L1T upgrade, while further general details about all the planned upgrades are available in the Technical Design Report (TDR).

1.4.1 The Phase-2 upgrade of Level1-Trigger

The Level-1 trigger is composed of various subsystems, and the proposed layout for the Phase 2 upgrade is illustrated in Figure 1.12[34]. For this upgrade, CMS plans to

[3] Referring to the fourth Long Shutdown in the LHC's history (1.11)

replace the Strip and Pixel tracking detectors with a new Inner Tracker, which features small-size pixel sensors, and an Outer Tracker equipped with strip and macro-pixel sensors. The Outer Tracker provides tracks to the trigger through the Track Finder (TF) and maintains threshold and efficiency values consistent with those from LHC Run 1. The tracking acceptance in the forward region will be extended from a pseudorapidity of $|\eta| < 2.5$ to $|\eta < 4$ thanks to the Inner tracker, while the Outer Tracker components cover up to $|\eta| \lesssim 2.8$. he Outer Tracker provides tracks to the trigger through the Track Finder (TF) necessary to maintain L1 threshold and efficiency values consistent with those from LHC Run 1.

Notably, this marks the first time tracking information will be available for the Level 1 trigger. The Endcap calorimeters will be replaced by the high-granularity calorimeter (HGCal), along with a Barrel Calorimeter Trigger (BCT) system. For the latter, the existing calorimeters are kept, but for the first time they will send full granularity information to the L1T system and not just to the HLT and the Offline reconstruction. The Endcap Muon Track Finding (EMTF) and Barrel Muon Track Finding (BMTF) will incorporate additional chambers to cover up to $|\eta| < 2.5$ and apply efficient muon identification algorithms.

A new Correlator Trigger system will match tracks with the Global Calorimeter and Muon Trigger (Calo and GMT), along with tracking information. The correlator layer employs complex object identification algorithms (see Section 1.4.1.1) and provides a sorted list of trigger objects to a Global Trigger (GT). The GT processes significantly more information than the current system and applies more sophisticated algorithms to produce an L1 Acceptance. This acceptance is sent to the CMS Trigger Control and Distribution System (TCDS), which distributes it to the subdetector backend electronics, initiating readout to the data acquisition system (DAQ).

The Phase-2 upgrade of the L1 trigger system is designed not only to maintain the signal selection efficiency at the Phase-1 performance level but also to enhance and enable the selection of potential manifestations of New Physics. Additionally, the longer latency available with this upgrade will allow for higher-level object reconstruction and identification, as well as the evaluation of complex global event quantities and correlation variables to optimize physics selectivity. Specifically, the upgrade will enable the possibility of running particle flow (PF, see Section 1.3.3.1) and pileup per particle identification (PUPPI, see Section 1.3.3.2) in the correlator layers, as shown in Figure 1.12. Few more details in the subsection 1.4.1.1.

In addition, the design of the phase-2 L1T includes a dedicated scouting system streaming data from key parts of the trigger at 40MHz via FPGAs into HPC resources. The next section will focus on the importance and the role of this original data stream.

1.4.1.1 Particle Flow and PUPPI algorithm at trigger level

As mentioned earlier, one of the primary challenges for the HL-LHC experiments will be to maintain detection efficiency for interesting physics events occurring at the electroweak energy scale within the same bunch crossing. Several studies, such as [23], are dedicated to addressing this challenge and determining which collision events should be fully read out for further analysis.

These studies explore the feasibility of implementing particle flow-like reconstruction and pileup per particle identification, which are among the best-performing algorithms in current offline processing, at the hardware trigger level in the upgraded trigger system. This approach leverages the correlator layer to provide a global view of each collision event, allowing these algorithms to run on the FPGAs of the Level-1 trigger correlator.

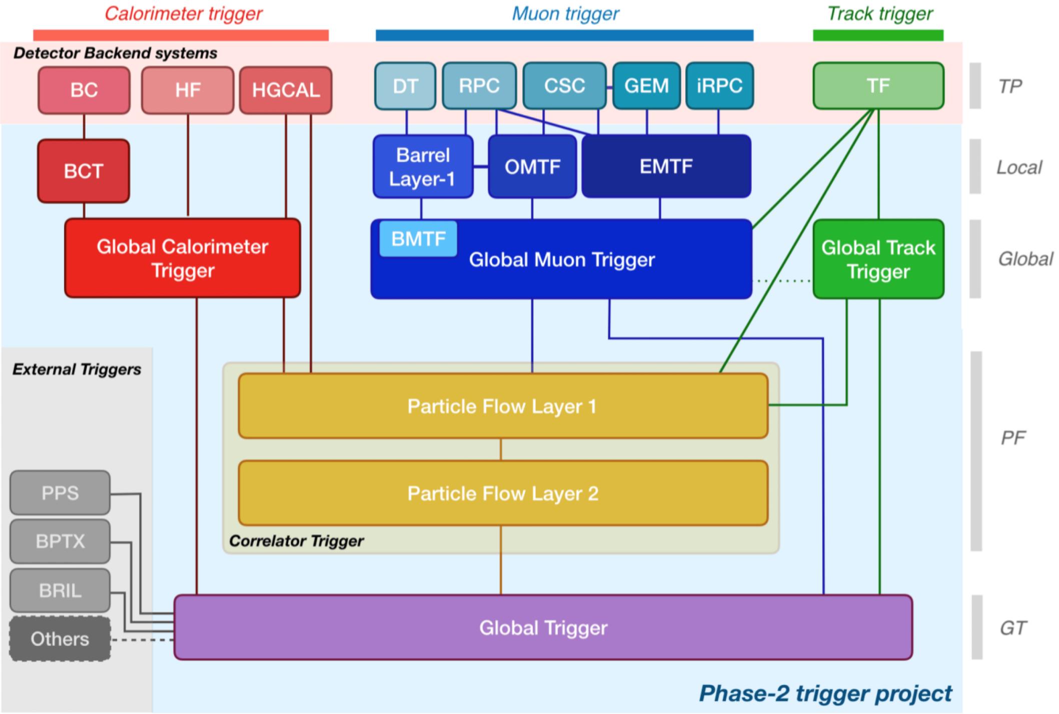


Figure 1.12: *Functional diagram of the CMS L1 Phase-2 from [34]. The main data flow is shown with solid lines. Additional data paths are under study, including direct connections from systems upstream of the Correlator Trigger to the Global Trigger, and paths that allow Tracker data to be passed to the Muon Triggers. The calorimiter trigger (red) is composed of the barrel calorimiter trigger (BCT) and the global calorimiter trigger (GCT), receiving inputs from the barrel (BC), endcap (HGCAL) and HCAL Forward Detector(HF). The muon trigger (blue) is composed of a barrel layer-1 and muon track finder processors: BMTF, OMTF and EMTF for each detector region, and receiving inputs from drift tubes (DT), resistive plate chambers (RPC), cathode strip chambers (CSC) and gas electron multiplier chambers (GEM). The global muon trigger (GMT) matches muons with tracks from the track finder (TF). The event vertex is reconstructed in the global track trigger (GTT), and the correlator trigger (CT) implements the particle-flow reconstruction. The global trigger (GT) issues the final L1 trigger decision.*

The studies present proof-of-principle implementations of these algorithms for FPGAs, demonstrating that they significantly improve the physics performance of the Level-1 trigger while remaining feasible in terms of FPGA resource usage and latency. This approach holds promise for enhancing the efficiency of physics event selection at the trigger level.

1.5 The scouting system or Data scouting at the CMS experiment

The CMS experiment produce a vast amount of data. To handle this massive throughput, the previously described CMS trigger selectively processes and filters data based on established particle physics knowledge. However, while invaluable, this system inherently introduces biases into the dataset and often omits significant amounts of statistics vital for observing rare decay channels.

Data scouting[35] refers to the use of physics objects reconstructed online during data taking to perform searches and measurement. The technique, pioneered by the CMS experiment, allows events to be recorded for analysis at a rate of several additional kHz with negligible impact on total data volume. With the data scouting system is possible to utilize objects within the trigger chain, extract and process them online to ensure efficient storage, where online means during the trigger timing. This approach focuses on obtaining objects with a reduced level of accuracy, trading off some resolution for greater statistics. While this technique has been prevalent at the HLT level[35, 36], new opportunities are presented with the LHC's upgrade. In particular, the possibility of data scouting at the L1 trigger is emerging. This system will extract L1-trigger primitives generated by sub-detectors and identify trigger objects at various stages of the L1-trigger hierarchy.

The Data scouting approach takes place in both the two trigger levels. This thesis take care of introducing only the scouting at the level 1 trigger concentrating on the LHC's phase-2, since it's focused on the usage of the scouting data coming from the L1 correlator layer.

1.5.1 Scouting at the level 1

As previously detailed in the previous section (Section 1.4.1), the Phase-2 upgrade of the CMS L1 trigger system introduces several critical improvements. These enhancements include a new tracking system that incorporates a track finder processor providing tracks to the Level-1 trigger. Additionally, a high-granularity calorimeter furnishes fine-grained energy deposition information in the endcap region, and new front-end electronics supply the L1 trigger with high-resolution data from the barrel calorimeter and the muon system.

The upgraded L1 trigger will primarily rely on Xilinx Ultrascale Plus series FPGAs capable of performing complex feature searches with resolutions often similar to those of the offline reconstruction. All these upgrades enable a form of scouting at the Level-1 trigger, making it interesting and important to explore, even at the unprecedented rate of 40 MHz.

The L1 Data Scouting (L1DS) is designed to capture Level-1 intermediate data produced by trigger processors at the beam-crossing rate and perform analyses based on these limited-resolution data. The L1DS operates semi-independently, separated from the conventional trigger and data acquisition chain. It will receive trigger data using a 25 Gb/s serial interconnect technology, maintaining the link protocol intrinsic to the trigger.

Dedicated FPGA boards facilitate data acquisition, bridging the synchronous trigger and the asynchronous scouting DAQ domain (ScDAQ). The immediate processing step in the scouting data landscape is executed in the I/O nodes, directly connected to the data acquisition boards. Reports such as [37] and [38] present the first results from a demonstrator being installed for LHC Run-3. This demonstrator, a data acquisition system operating at the LHC bunch-crossing rate, collects data from various Level-1 trigger components, and it uses different types of FPGA boards.

Phase-2 upgrade Particularly relevant for this thesis is the phase 2 upgrade, as previously anticipated. In this future configuration, the L1 Data Scouting (L1DS) will rely on the DAQ800 board, which is specially developed component for the CMS Phase-2 upgrade. The DAQ800 board will be equipped with two Xilinx VU35P FPGAs, 12x4 Samtec Firefly optical links with a data input capacity of up to 25 Gb/s per link, and ten QSFP outputs, each capable of handling up to 100 Gb/s. A small batch of similar DAQ400 prototype boards has already been manufactured.

The L1 scouting system designed is structured to be adaptable and scalable, enabling

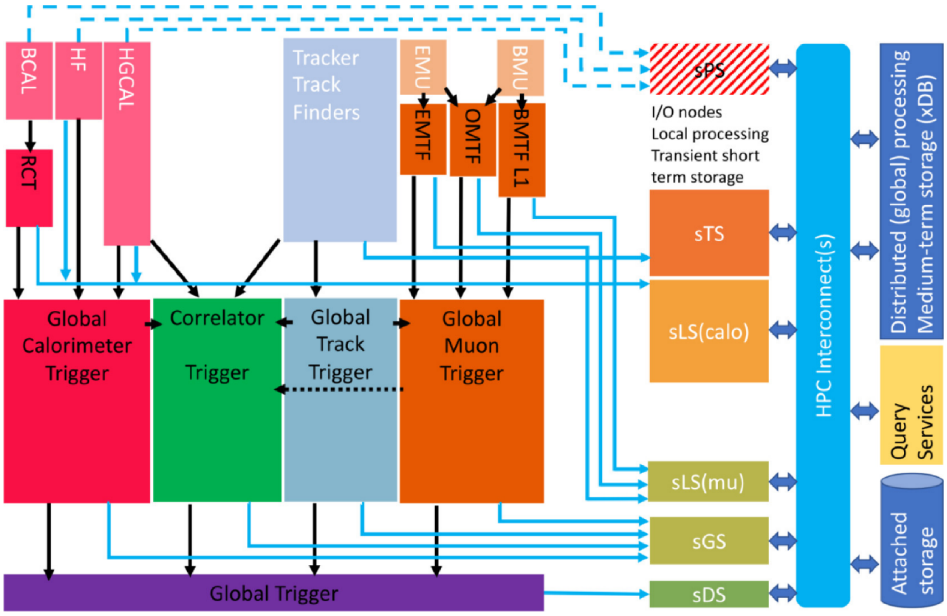


Figure 1.13: *Architecture of the Phase-2 Level-1 trigger and scouting system. Scouting I/O nodes will receive data from spare L1 trigger outputs and propagate them to a distributed computing farm via a high performance computing interconnect. Analysis results will then be sent further for long-term storage. The system is stageable by design, already being able to provide utility in the smallest version where it receives data only from the global trigger stages (sDS and sGS). Figure from [37].*

the system to grow as needed. The baseline configuration, illustrated in Figure 1.13 as the scouting decision system (sDS) and the scouting global system (sGD), will require seven DAQ800 boards. These boards will accept incoming data links from the global trigger, global calorimeter trigger, global track trigger, global muon trigger, and, notably for this thesis, from the L1 correlator. Possible future extensions could encompass the capture of data from local/regional muon and calorimeter triggers (sLS), L1 tracks (sTS), and L1 calorimeter trigger primitives (sPS). As reported in [39] and [38], the demonstrator of the L1DS system for CMS is already in operation and capable of collecting data from multiple L1 sources.

The project discussed in this thesis focuses on L1 scouting at the Correlator layer and aims to demonstrate the feasibility of online event selection using data from this trigger layer. The thesis underscores the importance of using unbiased scouting objects, which have the potential to enable the study of otherwise inaccessible or rare physics signatures. This approach is particularly relevant for events that don't fit within the L1 trigger's acceptance budget or have requirements that are orthogonal to "mainstream" physics.

Chapter 2

The W boson rare decay to 3 charged pions analysis

This chapter of the thesis is dedicated to exploring the viability of an online selection leveraging scouting data 1.5.1. To accomplish this, a specific channel must be chosen. The thesis opts to investigate the elusive decay of the W boson into three charged pions, serving as a prototype analysis. This decay process, yet to be observed, offers an excellent candidate for L1-scouting analysis due to its low-multiplicity characteristics, representing an atypical signature that doesn't neatly align with standard trigger configurations.

The chapter delves into the analysis of the W boson decaying into three pions. It commences by reviewing previous research into this unique decay mode. Subsequently, it introduces the dataset, comprised of Monte Carlo (MC) simulations, and outlines the selection algorithm, providing comprehensive insights into each of its criteria. Furthermore, the chapter discusses the event features and characteristics before embarking on a preliminary background estimation and characterization. The culmination of this analysis includes the determination of an upper limit on the branching ratio for this intriguing decay channel.

2.1 State-of-the-art analysis

Previous investigations have already delved into the rare decay of the W boson into three π [40]. The study of exclusive rare decays of the W boson to hadrons holds the promise of providing a precision measurement of the W boson mass, relying solely on observable decay products. Such observations also serve as a probe of the strong interaction, operating at the interface between the perturbative and non-perturbative domains of Quantum Chromodynamics (QCD). Other exclusive W boson decay modes explored include $W^{\pm} \rightarrow \mathrm{Ds}^{\pm}\gamma$ [41] and $W^{\pm} \rightarrow \gamma\pi^{\pm}$ [42], both yielding 95% confidence upper limits on their branching ratios: $\mathcal{B}(W^{\pm} \rightarrow \mathrm{Ds}^{\pm}\gamma) < 1.3 \times 10^{-3}$ and $\mathcal{B}(W^{\pm} \rightarrow \pi^{\pm}\gamma) < 7.0 \times 10^{-6}$.

In the context of the $W \rightarrow 3\pi$ analysis, the dataset encompasses an integrated luminosity of $77.3\mathrm{fb}^{-1}$ collected during LHC Run 2. This corresponds to proton-proton collisions at a center-of-mass energy of 13TeV over the 2016 and 2017 period. The analysis incorporates an innovative algorithm designed to trigger and identify $\tau \rightarrow \mathrm{hadrons} + \nu_{\tau}$ weak decays. Events are selected using triggers that necessitate the presence of two τ candidates, each with a transverse momentum (p_T) threshold greater than 35 or 40GeV. Roughly 3% of the selected events exhibit two pions with $p_T > 35\mathrm{GeV}$ at the generator level. All pions must be reconstructed in the 1-prong τ decay mode, which signifies a single track.

Furthermore, an offline analysis requires the presence of a third pion candidate in the event. These three pions must be adequately separated from each other by at least $\Delta R = 0.3$, adhere to isolation criteria, and not all possess the same electric charge. More details can be found in [40].

The primary source of background in this analysis arises from standard model events composed of jets produced through the strong interaction, referred to as QCD multi-jet events. These events were estimated from data containing at least one reconstructed pion candidate that failed the isolation requirement. The second most significant background corresponds to $Z/\gamma^{(*)} \rightarrow$ events. Other contributions from diboson, single top quark, $t\bar{t}Z$, $t\bar{t}W$, and triboson backgrounds are relatively small.

The report concludes that no significant excess above the expected background has been observed, implying that the data align with the background hypothesis. Consequently, it establishes an upper limit on the branching fraction of this specific decay, utilizing the CL_s method [43, 44]:

$$\mathcal{B}(\mathrm{W} \rightarrow 3\pi) < 1.01 \times 10^{-6}. \tag{2.1}$$

2.2 W to 3 pions selection

This section is dedicated to the selection algorithm, which is designed to identify three charged pions with specific characteristics and properties. The selection process employs various filters with the goal of identifying a trio+plet of pions in the same event, stemming from a three-body decay. Subsequently, the invariant mass of this triplet is computed and depicted in a plot to determine if it falls within the mass window of the W boson. This plot helps ascertain whether the W boson resonance has occurred, which should manifest as a peak around the W mass[45]:

$$m_{\mathrm{W}} = 80.379 \pm 0.012\mathrm{GeV}. \tag{2.2}$$

Dataset This analysis relies on specific datasets composed exclusively of Monte Carlo simulations within the Phase-2 environment. Three sets of data are available: one containing signal events, and the other two consisting of background events. The background datasets comprise a "Neutrino Gun" dataset, which represents pure pileup events with low multiplicity, and a "$t\bar{t}$" dataset, signifying pure QCD background events with high multiplicity. It is worth noting that the top quark decays into a b quark and a W boson.

In addition to the events, the signal dataset provides a "generated" signal for each event. Each event is accompanied by an ntuple containing the particles generated within the event, complete with all the available features at the Level-1, which typically contain candidate triplets. Furthermore, each event contains an ntuple of objects labeled as "*Gen*", representing a fake signal consisting of a triplet of pions originating from the decay of a W boson. Moreover, they contain the features of the parent W boson too, from which the pions come from. The generated triplet includes all the corresponding features but does not take into account the CMS detector acceptance. The Gen triplets are typically used for comparison with the particle content of the events.

Specifically, for each event, it is essential to determine if the reconstructed triplet is associated to all the three generated pions. If so, then that triplet is a genuine candidate for the signal. Such cases are referred to as "matched" signals. It is crucial to emphasize that the particles contained in all the datasets precisely correspond to the candidate particles coming from the correlator layer, as this thesis primarily focuses on these particles. Consequently, from this point forward, these particles are referred to as

"Puppi candidates", since as explained in 1.4.1, in the phase L1T both PF and puppi algorithm will be run in the correlator layer.

Lastly, it is important to underscore that all the datasets and ntuples are read and processed using the `ROOT` framework[46, 47], specifically through a ROOT API known as `RDataFrame`[48]. ROOT's `RDataFrame` offers a modern, high-level interface for data analysis stored in various formats, such as TTree and CSV, in both `C++` and `Python`. Moreover, it harnesses multi-threading and other low-level optimizations, enabling users to fully leverage the resources available on their machines in a transparent manner.

2.2.1 Cutting filters

As anticipated in the preceding section, the selection process relies on a series of filters applied to the particles within each event. These filters are designed to distinguish the signal, which is a triplet of isolated pions with an invariant mass falling within the range of the W boson resonance, from background events. Some filters enforce conservation laws, while others depend on particle detection and reconstruction techniques. Additionally, certain filters are parameter-dependent, and in this first description, the default parameters employed in the final version of the algorithm are used. Later in this chapter, we delve into the tuning of these parameters and provide the rationale behind the choices.

The filters are performed in the following order:

PdgID or PID (particle ID) The particle flow algorithm seeks to identify the particles assigning them an index (sec. 1.3.3.1). The first filter aims to identify the particles involved by looking at the particle ID number assigned by PF, i.e. pions having the PDG ID equal to ±211. Moreover, the electrons (±11) are taken into account since pions are sometimes misidentified as electrons.

Transverse momenta lower bound The transverse momenta of the three pions are required to exceed certain thresholds. There are three distinct lower bounds, each corresponding to a different category of pions, thus ensuring that the triplet consists of one pion from each category. For example, a lower bounds triplet could be 18GeV, 15GeV, 12 GeV. This filter is based on the fact that the W boson mass is relatively large and its p_T is usually small, so the transverse momenta of its decay products should also be relatively large, typically on the order of at least 10GeV. The p_T cut is crucial for effectively discriminating against pileup particles. For instance, in neutrino gun events, particles typically have transverse momenta on the order of a few GeV.

Charge The charge conservation principle mandates that the total electric charge must be conserved in the decay process. Given that, the W boson is a charged particle with a charge of ±1, it's essential that the three pions cannot all have the same electric charge. At least one of the pions must carry an opposite charge to the others. For example, potential charge combinations for the W^+ are (+1, +1, -1), and for the W^-, (+1, -1, -1).

Angular separation The pions must exhibit an angular separation from each other, meaning their angular distance must meet the criterion $\Delta R > 0.5$, which sets a lower bound. This requirement serves both to suppress the QCD background and facilitate background estimation by making the isolation of the pions independent.

Invariant mass To be considered as a valid triplet, the invariant mass (eq.(1.6)) of the three selected pions must fall within the mass window of the W boson, specifically 60GeV $< m_{3\pi} <$ 100GeV. This constraint is designed to eliminate nonsensical background and retain genuine candidate triplets. If the W boson exists, its resonance should be around 80GeV.

Isolation Each of the three pions is required to be isolated. Isolation is a dimensionless quantity that characterizes the particle's separation from its neighboring particles. It's defined as the sum of the transverse momenta of neighboring particles, normalized by the transverse momentum of the main particle, as described by the equation:

$$\text{Isolation} = \sum_{i \in \text{n.n.}} \frac{\mathrm{p}_{\mathrm{T}i}}{\mathrm{p}_{\mathrm{T}}}. \tag{2.3}$$

Neighboring particles are identified as "nearest neighbors" when their angular separation falls within a cone around the primary particle, with an angular distance of $0.01 < \Delta R < 0.25$. The isolation value, as defined, quantifies the degree of isolation of a particle. To be considered valid, each pion must have an isolation value less than 0.5, a criterion essential for eliminating QCD backgrounds. Notably, the lower bound on the neighbor separation is nonzero due to the reconstruction technique of the particle flow algorithm, which sometimes reconstructs a spurious neutral particle very close to the pion.

Multiple triplets If the algorithm identifies more than one triplet within a single event, it selects the triplet with the highest sum of transverse momenta ($\sum_{i \in \text{triplet}} \mathrm{p}_{\mathrm{T}i}$). While instances of multiple triplets passing the selection criteria are exceptionally rare, this additional filter is applied to avoid any significant impact on the invariant mass spectrum, except for a possible shift towards higher values.

2.2.2 Selection on MC simulation

The results are typically presented through a plot of the invariant mass of the selected triplets. This approach is valuable for assessing the presence of the W boson resonance. The histogram in Figure 2.1 illustrates the outcome of applying all the previously described filters to the MC simulation containing the signal. The histogram prominently displays a resonance at approximately 80 GeV. As detailed in the preceding paragraph, the figure only depicts invariant mass values within the previously mentioned range (60GeV $< m_{3\pi} <$ 100GeV). The number of entries, representing the quantity of events in the signal MC simulation, is $\mathrm{N}_{\mathrm{events}} = 50,400$.

The number of pre-selected events, defined as the number of events with at least three pions passing the transverse momentum filter, is $\mathrm{N}_{\mathrm{pre}} = 7,148$. Meanwhile, the number of events that satisfy all the selection criteria, which corresponds to the number of candidate triplets, is $\mathrm{N}_{\mathrm{sel}} = 3,730$, amounting to an efficiency of:

$$\varepsilon = \frac{\mathrm{N}_{\mathrm{selected}}}{\mathrm{N}_{\mathrm{events}}} = 7.46\%. \tag{2.4}$$

It's important to note that the transverse momenta filter plays a crucial role in reducing the number of candidate events, making the selection highly efficient in reducing the number of candidate events, for which the more computationally expensive steps of triplet finding and isolation computation have to be performed.

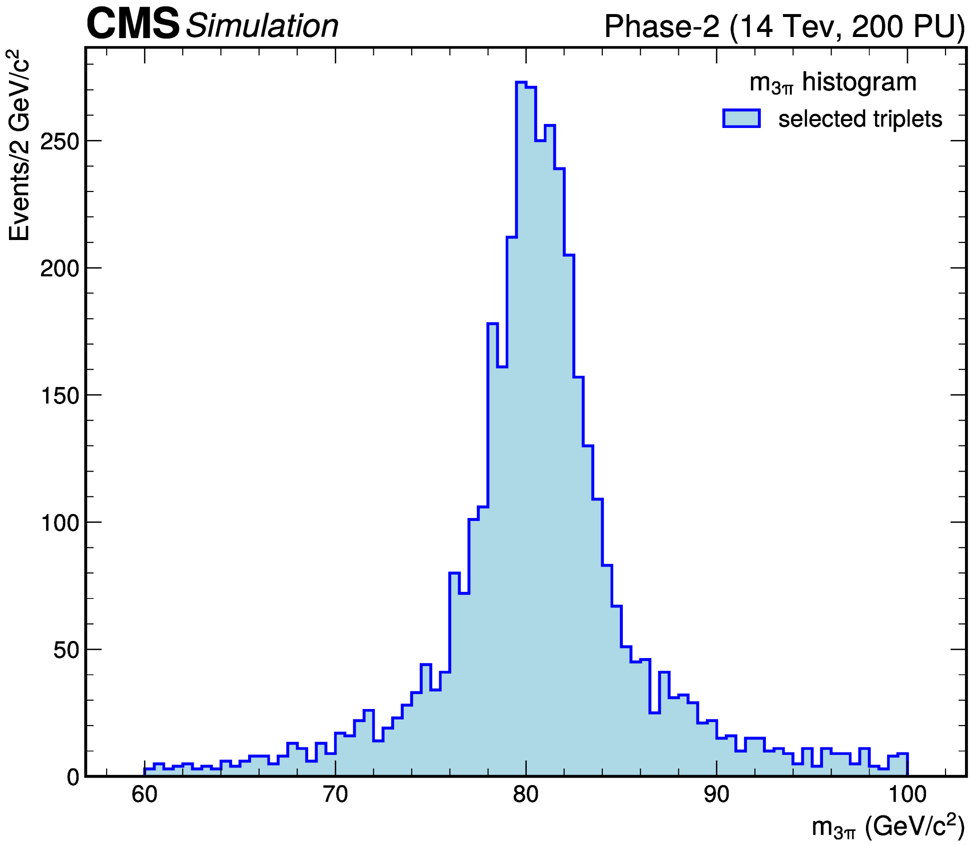


Figure 2.1: *The histogram illustrates the invariant mass distribution of the pion triplets selected using the algorithm outlined in Section 2.2. These triplets were extracted from the dataset containing the signal, comprising a total of* $50,400$ *events. The histogram prominently displays a peak at approximately* $80 GeV$*, corresponding to the resonance of the W boson. It's worth noting that the histogram exclusively exhibits invariant mass values within the W boson mass spectrum, thanks to the inclusion of a mass filter that confines the triplets to an invariant mass range of* 60 *GeV to* 100 *GeV.*

2.2.3 Matched pions

The "Gen" signal, as previously explained, serves as a benchmark to assess the efficiency and accuracy of the selection algorithm. In each event, three pions are generated, originating from the decay of the parent W boson, and their associated properties are generated as well. In contrast, real events contain "realistic" L1 Puppi candidates that may match one of the three *Gen* pions. This matching implies that they share similar properties or are in close proximity to each other, typically within an angular separation of $\Delta R < 0.1$.

To ensure that the comparison between generated pions and L1 Puppi candidates is meaningful, certain criteria are applied. Firstly, the generated pions must fall within the acceptance region of the CMS detector. The Phase-2 CMS detector's acceptance region is defined in terms of pseudo-rapidity, adhering to the following range:

$$|\eta| < 2.4, \tag{2.5}$$

This criterion corresponds to the coverage of the tracker. Additionally, a transverse momentum filter is employed, as L1 Puppi objects are required to have a transverse momentum exceeding 2 GeV.

Applying these criteria, there are 20, 707 events where all three generated pions meet the acceptance and transverse momentum requirements, which constitute approximately 41.1% of the total events. A histogram illustrating the invariant mass of all generated triplets that fulfill these criteria is presented in Figure 2.2. The filtered generated dataset is subsequently utilized for comparison with the output dataset generated by the selection algorithm. This feature plays a crucial role in the forthcoming section on parameter tuning (Section 2.2.5). An illustrative histogram showcasing the comparison between the invariant mass distribution of the selected triplets and that of the matched triplets can be seen in Figure 2.3.

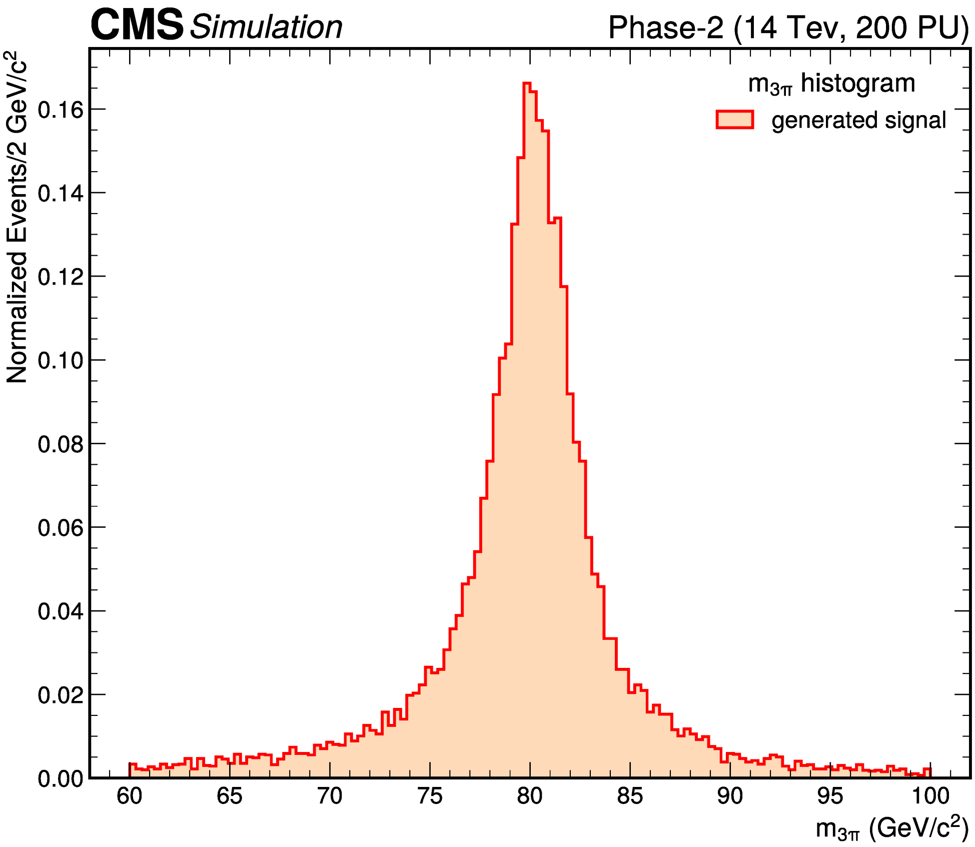


Figure 2.2: *The histogram depicts the normalized distribution of the invariant masses for the generated pion triplets. Specifically, it showcases the invariant masses of triplets in which all three pions fall within the acceptance region of the CMS detector, as defined in Equation* (2.5)*. The mass values are restricted to the range of* 60 *GeV to* 100 *GeV, enabling a direct comparison with the invariant mass histograms generated by the selection algorithm. As expected, the histogram exhibits the prominent presence of the W boson resonance.*

Matching with the "Gen" pions can be executed using either the triplets selected by the algorithm, as previously described, or with all the pions in the event. In the latter case, it is common to investigate the *reconstruction efficiency*. This efficiency is defined as the ratio between the number of particles generated ($\mathrm{N_{Gen}}$) and the number of particles reconstructed ($\mathrm{N_{Reco}}$) that have been correctly associated with the generated particles:

$$\varepsilon_{\mathrm{Reco}} = \frac{\mathrm{N_{Reco}}}{\mathrm{N_{Gen}}} \tag{2.6}$$

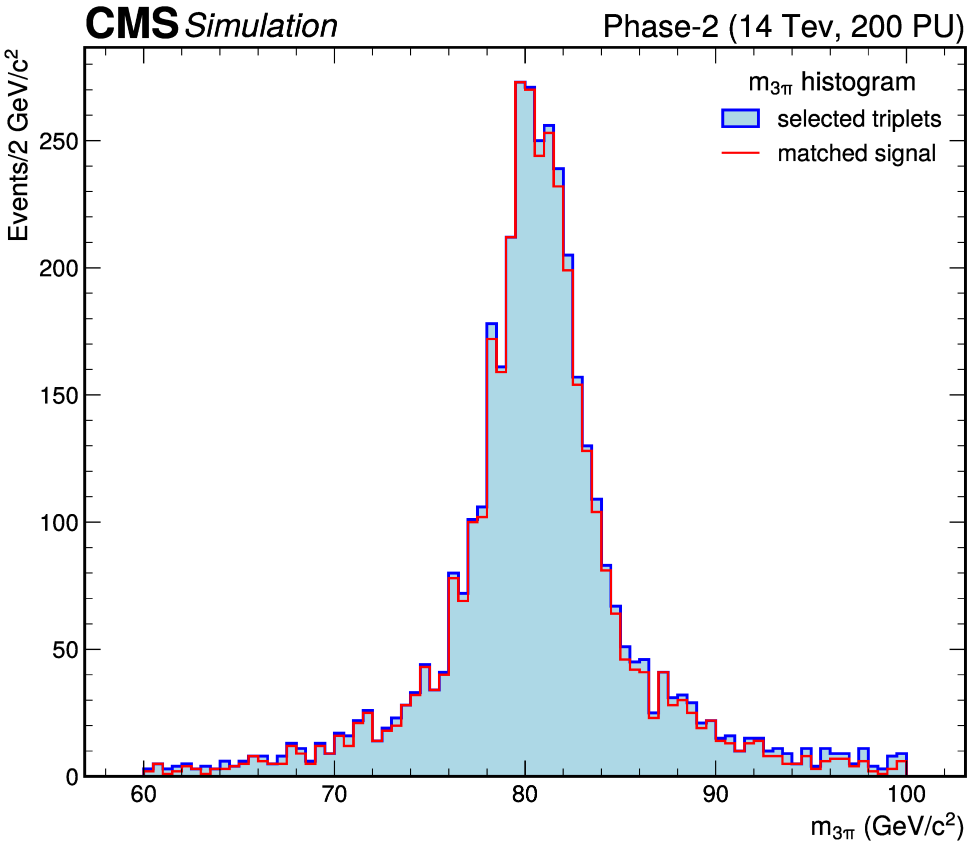


Figure 2.3: *The histogram illustrates in blue the invariant mass distribution of the pion triplets selected using the algorithm outlined in Section 2.2, in red the distribution of the triplets which match the generated ones.*

In practice, "reconstructed" is often used interchangeably with "matched" since both terms refer to pions that have been reconstructed and correctly correspond to the generated ones. A dedicated section in the following part of this thesis focuses on the estimation of this parameter.

2.2.4 Feature resolution and reconstruction efficiency

Another crucial aspect of the comparison between the generated signal and the Puppi candidates is the study of the resolution of the simulated signal. This parameter quantifies how accurately the detector or simulation reconstructs the three pions. Resolution, in this context, measures the accuracy of the detector or simulation in determining a particle's features. In this case, the focus is on the resolution of the transverse momentum, which determines how sharp will be the W invariant mass peak reconstructed from the three pions.

To estimate this resolution, L1-Puppi candidates must be associated with their corresponding generated particles. In each event, every candidate pion is compared with all three generated pions, considering both the spatial distance in terms of ΔR and the similarity in transverse momentum (p_T). When all three generated pions have corresponding matches in the Puppi candidates, the triplet is retained for the resolution assessment. These pions are referred to as "reconstructed."

The resolution of a particle feature (x) is defined as the standard deviation of the

pt interval	**Resolution**
$5 < \mathrm{p_T} < 10$ GeV	2.2%
$10 < \mathrm{p_T} < 20$ GeV	2.0%
$20 < \mathrm{p_T} < 30$ GeV	2.4%
$30 < \mathrm{p_T} < 60$ GeV	2.7%
$60 < \mathrm{p_T} < 100$ GeV	5.4%
overall: $5 < \mathrm{p_T} < 100$ GeV	3.9%

Table 2.1: *Table reporting the final results for the resolution for the different transverse momenta ranges of the generated pions*

differences between the measured feature and the generated values, normalized to the value of the generated feature:

$$\text{resolution} = \sigma\Big[\frac{x_{gen} - x_{reco}}{x_{gen}}\Big]. \tag{2.7}$$

The resolution serves as a metric for quantifying the disparity between the generated feature and the actual, observed feature. This disparity can be attributed to various sources, including uncertainties in the detector or in the simulation process. In particular, the dominant processes are the resolution of the hits position in the tracker and the fluctuation of the multiple scattering in the material. Notably, generated events are processed through a simulation of the CMS detector based on GEANT4 and reconstructed using the same algorithms used for collision data.

The analysis focuses on two features: pions transverse momenta and triplets' invariant mass. Specifically, in the investigation of transverse momenta, the resolution is evaluated as a function of these momenta, allowing an assessment of its variation concerning the pions' pT. For example, pions with $5\text{GeV} < \mathrm{p_T} < 10\text{GeV}$ are pre-selected and then the resolution is evaluated. Moreover, a pseudorapidity pre-filter is performed before to distinguish different detector areas. For simplicity, only one region is considered, namely the entire detector acceptance region ($|\eta| < 2.4$). Here, only a general example is provide: Figure 2.4 displays the distribution in eq.(2.7) for all pions within the detector acceptance, covering a range of $\mathrm{p_T}$ values. The histogram yields the overall resolution for transverse momenta, resulting in $\sigma(\mathrm{p_T}) = 3.9\%$. Table 2.1 presents the resolution results for the different pions transverse momenta. It indicates that the resolution increases with higher $\mathrm{p_T}$ values. This suggests that the detector has larger uncertainty when dealing with higher $\mathrm{p_T}$ tracks. Indeed, since a high-$\mathrm{p_T}$ particle has a track with a lower curvature, the measurement of the transverse momentum is less precise giving the same tracker resolution.

All the plots that correspond to the data summarized in the table are available in Appendix A4.2.1. These plots provide a more detailed visual representation of the results for different transverse momentum intervals.

It's worth noting that higher $\mathrm{p_T}$ ranges are affected by limited statistics, particularly in the $[60, 100]$ GeV range. In the context of W boson decay products, typical pT values rarely exceed 60 GeV. Consequently, the results for high $\mathrm{p_T}$ pions may not accurately reflect realistic conditions. The issue of limited statistics will also be addressed in the subsequent paragraph, particularly in Figure 2.6.

Finally, regarding the triplet invariant mass, the results are shown in figure 2.5, from

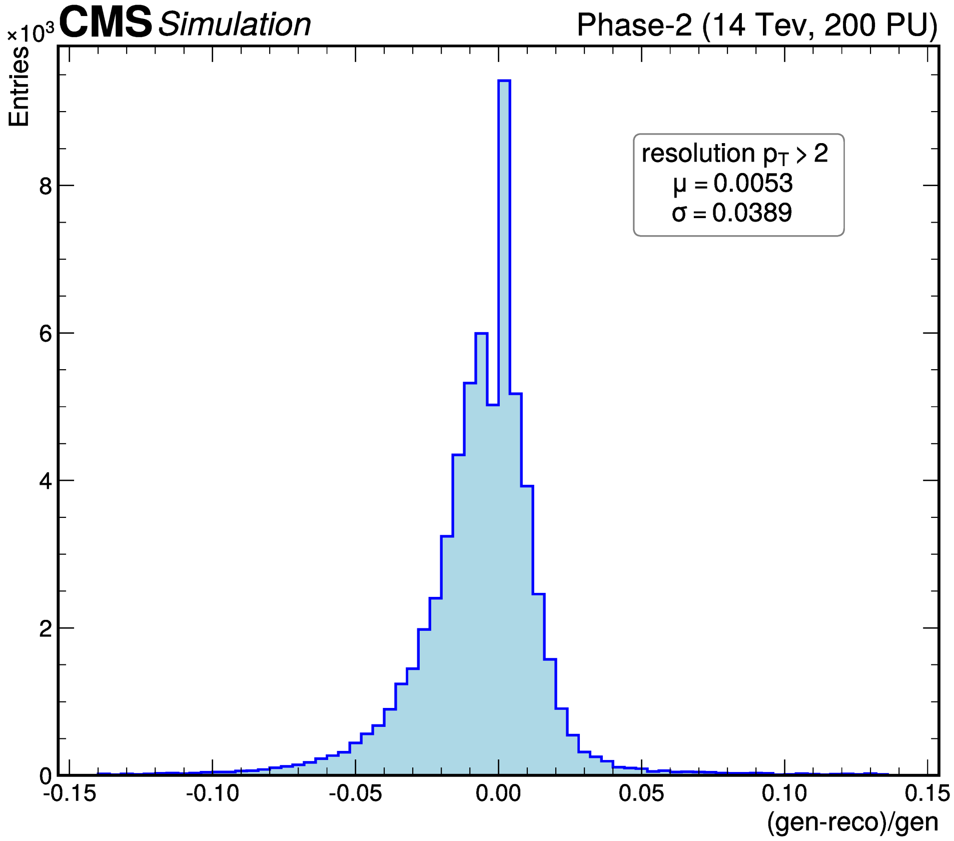


Figure 2.4: *The histogram illustrates the distribution as defined in Equation* (2.7) *for the transverse momenta of the pions. Specifically, it takes into account the generated pions falling within the detector's acceptance, and no additional filters are applied to the pions' transverse momenta. The figure also provides the standard deviation, which serves as the measure of the resolution, as defined in Equation* (2.7)*.*

which the resolution is estimated to be $\sigma(m_{3\pi}) = 2.2\%$. Here all the triplets having all the pions within the detector acceptance are considered.

Reconstruction efficiency Reconstruction efficiency, in essence, quantifies the ability of a detection system to accurately identify and reconstruct particles or events. It measures the proportion of generated particles that have been successfully reconstructed by the detector or analysis process. This efficiency is influenced by various factors, including the detector's geometry, its performance, and the properties of the particles involved. Typically, it reflects the precision of measurements: low efficiency results in a smaller collected sample of signal events, for the same integrated luminosity, and thus in a larger statistical uncertainties. On the other hand, high efficiency enables more accurate results. To assess reconstruction efficiency, the generated particles are compared to the reconstructed ones by computing the ratio of reconstructed pions to the total number of generated ones:

$$\varepsilon_{\text{reco}} = \frac{\text{N}_{\text{Reco}}}{\text{N}_{\text{Gen}}}. \tag{2.8}$$

Here, the numerator denotes the count of generated particles that are reconstructed (matched) in the simulation or experiment, while the denominator represents the total number of generated particles. Consequently, this definition underscores that recon-

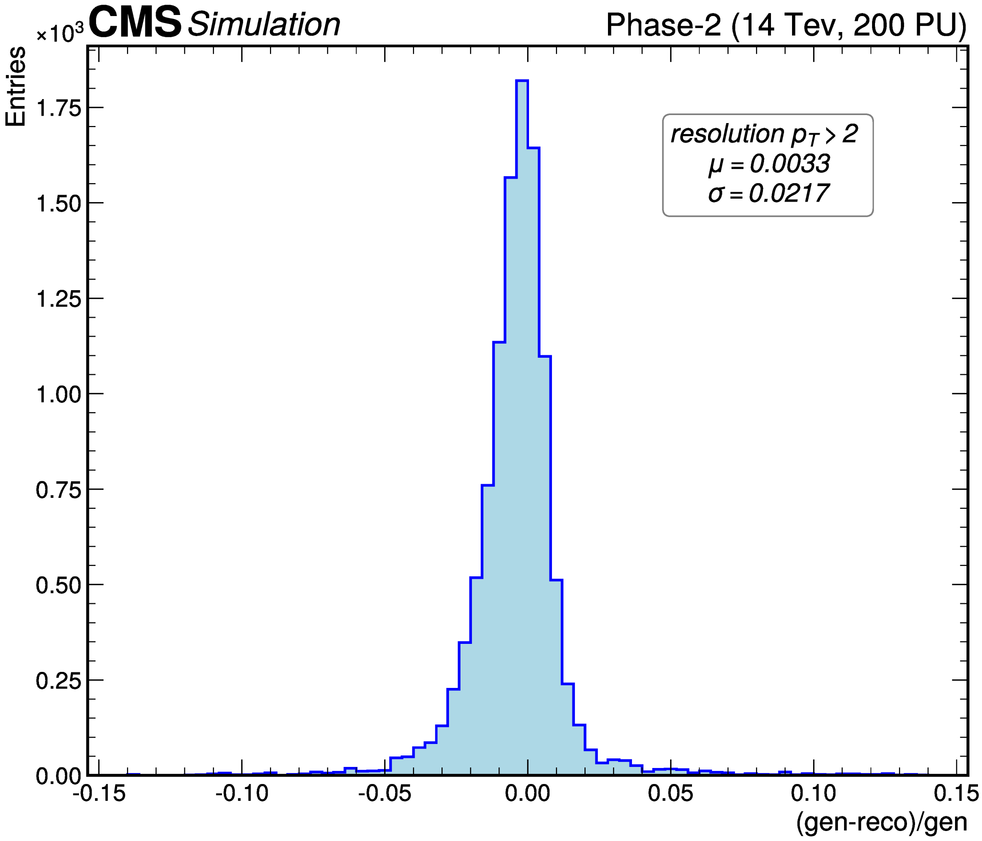


Figure 2.5: *The histogram illustrates the distribution as defined in Equation* (2.7) *for the invariant mass of the generated triplets having all the three pions in the acceptance. The figure also provides the standard deviation, which serves as the measure of the resolution, as defined in Equation* (2.7)*.*

struction efficiency measures the fraction of generated particles correctly identified and reconstructed by the detection system, which is why generated particles are employed.

It is customary to analyze the efficiency as a function of transverse momentum or pseudorapidity of the generated pions. This analysis is essential to understand how the efficiency varies concerning the detector's coordinates and particle energy. For the sake of simplicity, this discussion focuses on one pseudorapidity interval, specifically, the detector acceptance criterion $|\eta| < 2.4$.

In this specific pseudorapidity range, a histogram of transverse momenta was generated for both generated and reconstructed pions, spanning from 0 to 60 GeV with a bin width of 2 GeV (refer to Figure 2.6). Subsequently, the ratio of corresponding bins was computed, yielding the reconstruction efficiency for various transverse momentum intervals. These outcomes are depicted in Figure 2.7, offering insights into how the efficiency varies across distinct transverse momentum intervals with the corresponding errors. On the whole, the efficiency exhibits fluctuations, converging at approximately 75%. The figure also shows that higher values of transverse momenta correspond to higher errors due to the limited statistics available for those pions.

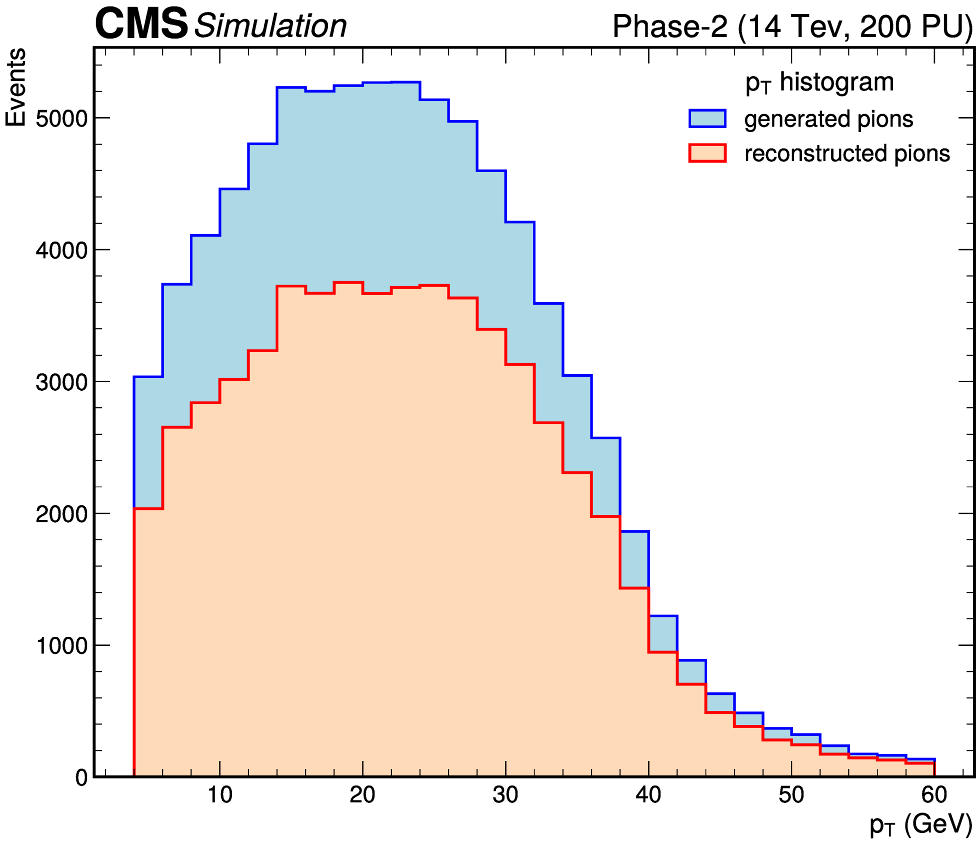


Figure 2.6: *The histograms display the distributions of transverse momenta for both generated and reconstructed pions. Each bin in these histograms has a width of* 2 *GeV. This bin width choice facilitates the calculation of the reconstruction efficiency through the ratio of corresponding bins in the two histograms. It's essential to note that there are only a few entries in the bins where* $p_T > 40$ *GeV. This scarcity of entries in these high transverse momentum bins reflects the limited statistics available for such high-energy pions.*

2.2.5 Parameter tuning

As mentioned earlier in this section, some filters within the algorithm depend on specific parameters. These parameters have a direct impact on the algorithm's ability to accurately extract the signal. Striking the right balance with these cuts is crucial. They should not be too tight, allowing candidate triplets to pass through, but also not too loose, as they must effectively separate the background from the signal. The effectiveness of the algorithm and the balance of this trade-off are assessed using two key metrics: selection *efficiency* and *purity*, which play a vital role in tuning the parameters. The former, as previously anticipated in eq. (2.4)is defined as:

$$\varepsilon = \frac{\mathrm{N_{select}}}{\mathrm{N_{events}}}, \tag{2.9}$$

while the latter as:

$$purity = \frac{\mathrm{N_{match}}}{\mathrm{N_{select}}}. \tag{2.10}$$

The goal of the selection chain is to be optimized, in order to maximize the Signal-to-Noise ratio, the estimation of which will be described in the subsequent section.

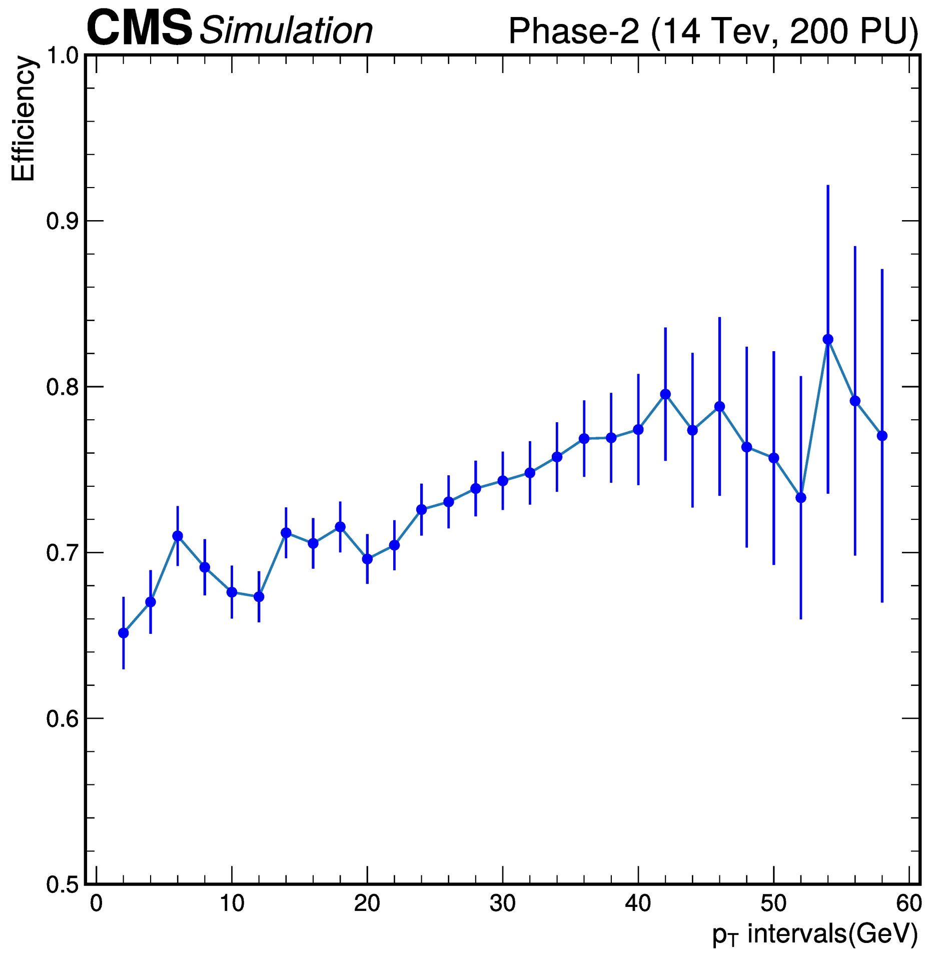


Figure 2.7: *This scatterplot illustrates the reconstruction efficiency as a function of pion transverse momenta. The efficiency exhibits an increasing trend with rising transverse momenta until it reaches approximately* 46 *GeV. Beyond this point, there is a distinctive fluctuation in the efficiency, primarily due to the limited statistics available in those higher ranges. It's important to mention that the overall efficiency stabilizes at around* 75%. *The figure also shows that higher values of transverse momenta correspond to higher errors due to the limited statistics available for those pions.*

The ideal set of parameters represents a delicate balance between efficiency and purity, as previously rationalized. By adjusting these parameters or relaxing some filters, we gain valuable insights into how algorithm performance varies and the impact of parameter changes. Certain filters, such as those ensuring conservation laws (e.g., charge) or specific particle identification (PdgId), are retained as they are fundamental to the analysis.

Releasing or smoothing filters is also a good way to estimate the background contribution in the process. The subsequent section will address this aspect by presenting a method for approximating the signal-to-noise ratio.

This thesis primarily focuses on parameters that are expected to have the most significant impact on background suppression, specifically the lower bounds on p_T and isolation. While other parameters, such as ΔR boundaries defining the isolation cone and the ΔR separation for the pions, could be considered, they are expected to have a lesser effect on algorithm performance and are thus excluded for the sake of simplicity. It's worth noting that the separation between pions must be at least twice the maximum

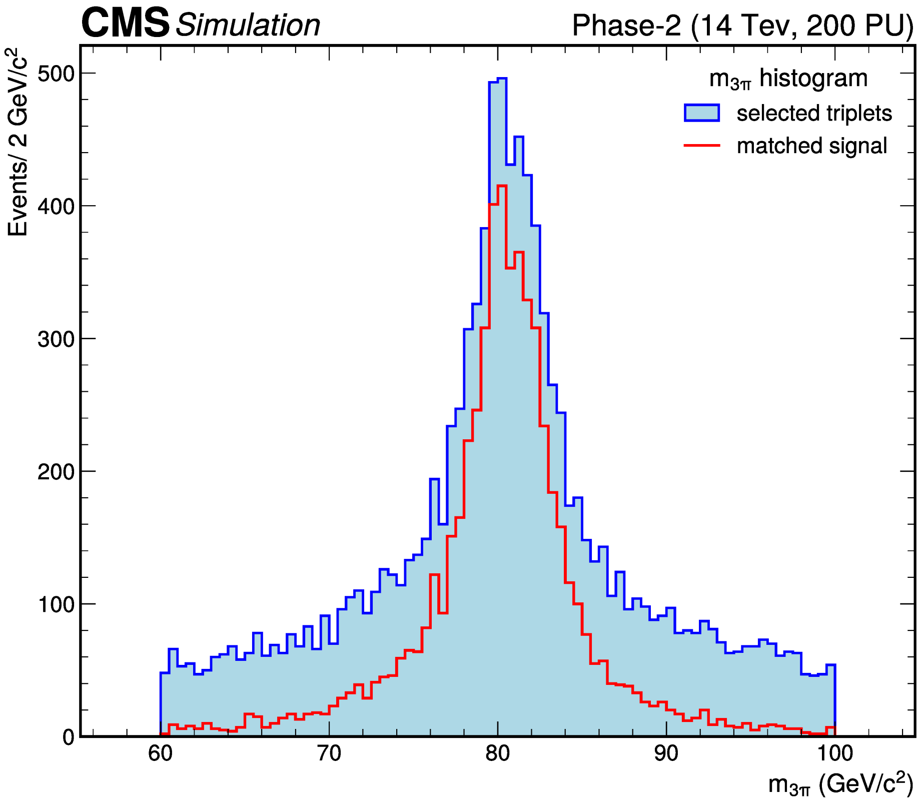


Figure 2.8: *The histogram illustrates in blue the invariant mass distribution of the pion triplets selected by relaxing the p_T threshold, and in red the distribution of the triplets which match the generated ones. Specifically, the transverse momenta lower bounds are set to* $(15, 12, 4)$ *GeV. It is essential to highlight that the algorithm selects a substantial number of triplets, but only a relatively small percentage of them achieve a match with the generated counterparts. The substantial divergence between the two histograms is most pronounced far from the peak, indicating that triplets within those bins predominantly correspond to background events. This observation is essential for understanding the algorithm's performance when the transverse momentum filter is relaxed.*

boundary of the isolation cone to ensure the independence of the extrapolation of neighboring particles in the isolation estimate. Therefore, the boundaries for the isolation cone are maintained as $0.01 < \Delta R < 0.25$, while the pion separation requirement is $\Delta R > 0.5$, which is double the cone's maximum boundary.

Now, delving further into parameter optimization, the selected $\mathrm{p_T}$ values fall into three distinct intervals, each corresponding to one of the three boundaries:

$$\begin{aligned} \text{high } \mathrm{p_T} &\in [12, 22]\mathrm{GeV}, \\ \text{medium } \mathrm{p_T} &\in [10, 20]\mathrm{GeV}, \qquad (2.11) \\ \text{low } \mathrm{p_T} &\in [8, 18]\mathrm{GeV}, \end{aligned}$$

It's important to note that these three sets of values change simultaneously with a consistent step size of 2 GeV or 3 GeV. The selection of these specific boundaries respects the hierarchical ordering of pT. Removing the pT filter completely or to a significant extent results in a large number of triplets passing the selection, with most of them representing background events. In such cases, the purity values are notably low, indicating that the signal constitutes only a small fraction when compared to the background. This outcome underscores the critical role of the transverse momenta filter in effectively distinguishing signal from background.

As an illustrative example of this scenario, consider the invariant mass histogram depicted in figure 2.8, where the blue curve represents the selected triplets and the red curve represents the matched ones. The chosen p_T boundaries in this instance are $(15, 12, 4)$ GeV, yielding an efficiency of $\varepsilon = 21.5\%$ and a purity of 51%. Notably, the discrepancy between the blue and red histograms is more pronounced in the regions to the left and right of the peak, signifying that those triplets primarily consist of background events. The impact of this filter on algorithm performance will be further discussed in section 3.3, particularly in the context of timing optimization. On the other hand, when examining the isolation parameter, the chosen values fall within the following interval: iso $\in [0.2, 0.6]$, with a step size of 0.05. Setting a low value for this parameter implies demanding strong isolation for the pions, where the sum of the transverse momenta of neighboring particles constitutes less than 20% of the pion's transverse momentum.

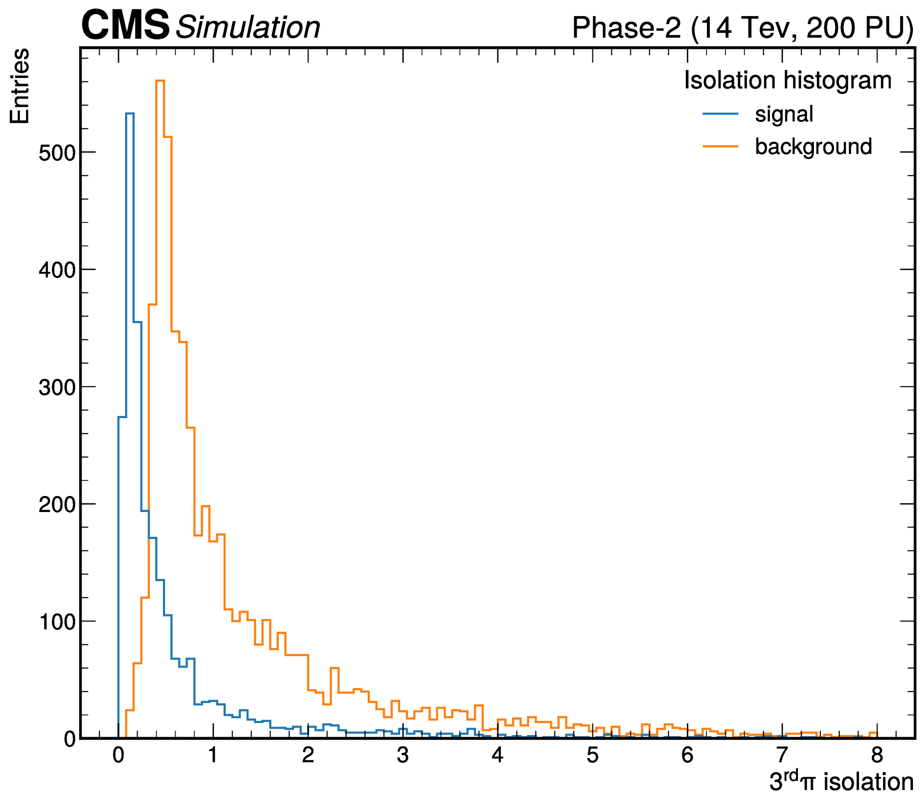


Figure 2.9: *The histogram displays the isolation of the third pion in the signal (blue) and background (orange) datasets when the isolation constraint on the third pion is relaxed. The figure unequivocally demonstrates that background events exhibit higher isolation values, underscoring the substantial impact of non-isolated pions on background events.*

The impact of the isolation filter becomes apparent when examining the histogram of the isolation parameter on the third pion within triplets, specifically the pion with the lowest transverse momentum. To generate these histograms, the selection is executed on both datasets (signal and background) while relaxing the isolation constraint on the third pion. This choice is significant because, being the pion with the lowest p_T, it is more likely to be non-isolated. As depicted in Figure 2.9, the background events (orange) exhibit higher isolation values compared to the signal events (blue). This discrepancy underscores the importance of the isolation parameter as a discriminating feature in the selection process.

For a more comprehensive assessment of the isolation parameter's impact, the selection can be applied to the background dataset with and without the isolation filter, while slightly adjusting the transverse momentum thresholds to permit some events to pass. The ratio of events passing the selection with the isolation filter versus those without it is approximately 9%, indicating that background events are notably influenced by non-isolated pions.

The heatmap in figure 2.10 illustrates the relationship between efficiency and purity with respect to isolation (y-axis) and $\mathrm{p_T}$ (x-axis) values, where their ranges are the ones described before. This heatmap provides a qualitative estimate to visualize the effects of the filters on the algorithm. From the plot, it is evident that the transverse momenta cut has a more substantial impact on both metrics compared to the isolation parameter. This outcome was anticipated since, as mentioned earlier, relaxing the $\mathrm{p_T}$ lower bounds permits more pile-up pions to pass the selection, thus influencing both efficiency and purity. Another important result to note from the heatmap is that the two metric parameter are inversely related, as expected. Another crucial observation from the heatmap is that the two metric parameters are inversely related, which is expected. It is worth noting that both filter parameters are selected to maintain a purity of approximately 90% or higher, as it is crucial to preserve a high signal extraction rate. Please remember that only these two filter parameters are allowed to vary, while the others remain constant, as indicated at the beginning of this section.

To set the best set of parameter, based on the efficiency and purity metric, the best trade-off has to be chosen. The adopted values are:

$$\begin{aligned} \varepsilon &= 7.8\% \\ purity &= 95.7\%, \end{aligned} \tag{2.12}$$

which correspond to the case in which the transverse momenta lower bound are $(18, 15, 12)$ GeV and iso < 0.5.

Finally, as previously emphasized, it is crucial to apply the algorithm to the dataset containing only background events as an additional verification step. Therefore, another aspect to examine and verify is the number of selected triplets in the background events. In an ideal scenario, when the purity in the signal file is high, this number should be close to zero. A robust and efficient algorithm should predominantly select events when signal events are present, effectively minimizing selections in background-only events.

2.3 Signal and Background events estimate

This section discusses a preliminary estimation of the signal-to-noise ratio, starting from the background and signal events estimation. Given the limited number of events in the simulations, it is essential to extrapolate these estimates to a larger dataset, generalizing the expected numbers within a certain period of data acquisition or a certain amount of integrated luminosity. All the performed estimations rely on an integrated luminosity of $L = 400\mathrm{fb}^{-1}$, which corresponds to approximately one year of data acquisition with the LHC maximum luminosity performance, meaning that the results regard the expected number of events in $400\mathrm{fb}^{-1}$ of integrated luminosity, based on the currently available simulations and the described selection algorithm.

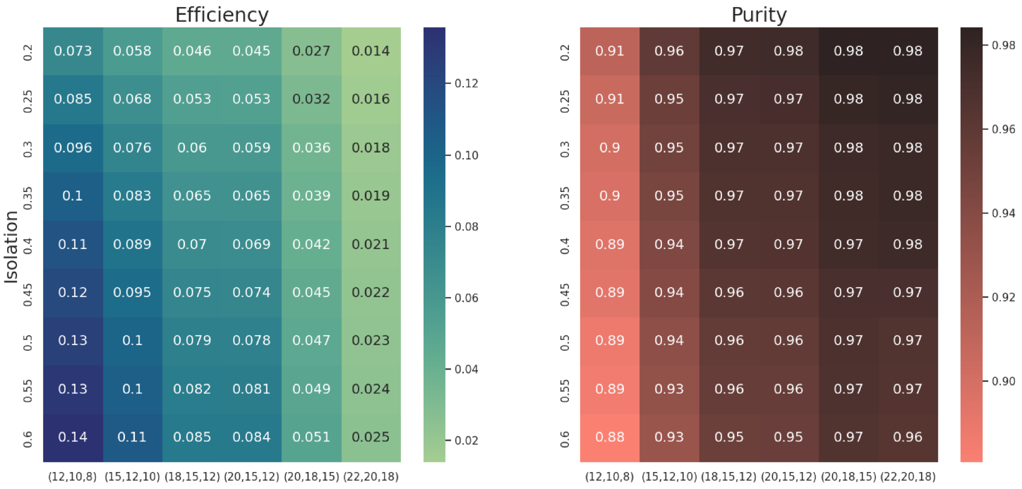


Figure 2.10: *The heatmap visually presents the interplay between efficiency (on the left) and purity (on the right) concerning different values of isolation (y-axis) and transverse momentum (x-axis).*

Signal The signal dataset, as previously described, consists of a total of $\mathrm{N}_{\mathrm{event}} = 50,400$. By running the algorithm with the parameters and filters as described in the paragraph 2.2.1, the number of events having a candidate triplet, i.e passing the selection, is $\mathrm{N}_{\mathrm{passed}} = 3730$. To provide a general estimation, as previously introduced, the used formula is:

$$\mathrm{N}_{\mathrm{sig}} = \sigma_{\mathrm{W}} \times \mathcal{BR}_{(\mathrm{W}\to 3\pi)} \times L \times \varepsilon_{\mathrm{sig}} \tag{2.13}$$

where $\sigma_{\mathrm{W}} = 1.85 \times 10^5$pb is the theoretical cross-section of the W boson production in the LHC collisions, evaluated at the next-next-to-leading-order (NNLO); $\mathcal{BR}$ is the branching fraction of the decay, which the used value is $\mathcal{BR}_{(\mathrm{W}\to 3\pi)} \simeq 10^{-7}$ for an order-of-magnitude estimation; L is the integrated luminosity ($L = 400\mathrm{fb}^{-1}$) and lastly, ε is the efficiency of the selection algorithm on the dataset, described in eq. (2.9), which value has previously reported in eq. (2.4), i.e. $\varepsilon = 7.86\%$. The results describes the expected number of events having a candidate triplet in, estimated using eq. (2.13):

$$\mathrm{N}_{\mathrm{sig}} = 576. \tag{2.14}$$

It's essential to emphasize that this is an indicative number, and it is sensitive to the branching fraction of the process, which has neither been experimental measured nor theoretically estimated.

Background The background estimation relies on studying the performance and the results of the selection algorithm described in section 2.2, using the background simulation dataset. This estimation takes advantage of the flexibility of the algorithm's performance, as discussed in the previous section when examining the filter parameters. When applying the final version of the selection to the background-only dataset, the number of events passing the selection is typically zero. To estimate the expected number of events in a given integrated luminosity, it is possible to release certain filters and study how this number varies. This information can then be used to provide a final estimate.

The formula used to estimate the expected number of background events is analogous to the signal one:

$$\mathcal{N}_{\mathrm{bkg}} = \mathcal{R}_{\mathrm{MB}} \times \mathrm{T}_{\mathrm{DAQ}} \times \varepsilon_{bkg} \tag{2.15}$$

where $\mathcal{R}_{\mathrm{MB}}$ indicates the rate of minimum bias (31.5MHz); $\mathrm{T}_{\mathrm{DAQ}}$ is the "effective" data acquisition period, i.e. the period keeping the HL-LHC instantaneous luminosity ($\mathcal{L} = 7.5 \times 10^{34}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$) constant until the integrated luminosity $L = 400\ \mathrm{fb}^{-1}$ is collected; and lastly ε_{bkg} is the background efficiency, i.e. the fraction of background events passing the selection. Regarding the rate of minimum bias, it represents the frequency of collisions between pairs of full bunches (averaged over an orbit) which differs from the 40MHz bunch crossing rate. The TDR [26] reports a maximum number of 2808 full bunches, meaning that, since an orbit can contain spatially 3564 bunches, the rate of minum bias is $\mathcal{R}_{\mathrm{MB}} = 40\mathrm{MHz} * 2808/3564 \simeq 31.5\mathrm{MHz}$.

While the values of the first two parameters in the eq.(2.15) are given, the background efficiency has to be estimated from the performance of the selection algorithm on the background simulations, as previously anticipated. The definition of ε_{bkg} is the same for the signal (ε) in eq.(2.9), where the only difference is the used dataset. For these calculations the used data is the "Neutrino Gun" background dataset, which contains a total number of background-only events equal to $\mathcal{N}_{\mathrm{events}} = 1,941,240$, which also corresponds to the denominator in the efficiency formula. On the other hand, the numerator, i.e. the number of selected events, is less trivial to estimate than the signal one, since running the algorithm on the background file, zero events are selected, and this is what actually happens. This result is expected, since the algorithm aims to extract the candidate events in a file without them. However, $\varepsilon_{bkg} = 0$ is not a realistic estimate, the expected value is $\varepsilon_{bkg} \ll 1$ but still different from zero.

The thesis introduces an innovative method for estimating background efficiency. This method involves releasing or smoothing some filters by adjusting their parameters, allowing some events to pass the selection. By studying how the number of selected events changes with filter parameters, the numerator of the efficiency can be determined. While the result is a rough estimate, it provides an indicative parameterization of the background. After a brief examination of the filter effects on the background dataset, as briefly discussed in the previous section, it is found that the most discriminating filter parameters are the lower bounds on transverse momenta. The isolation filter has a smaller impact on the selection efficiency but still contributes. The number of events ($\mathcal{N}_{\mathrm{events}}$) is estimated by varying or releasing one of these two filters or even a combination of them. The proposed method is based on the following formula:

$$\mathcal{N}_{\mathrm{events}} = \mathcal{N}_{(12)(3)} \simeq \frac{\mathcal{N}_{(12)(\not{3})} \cdot \mathcal{N}_{(\not{1}2)(3)}}{\mathcal{N}_{(\not{1}2)(\not{3})}}, \tag{2.16}$$

where the subscripts notation indicate the filter configuration on the three pions. For example, $\mathcal{N}_{(12)(\not{3})}$ indicates the number of events passing when the filter for the first two pions are kept the same described before while the third one is released or reduced. When dealing with the isolation filter, the formalism has the following meaning: (3) means that the isolation is required for the third pion and ($\not{3}$) means that its isolation is released. For simplicity, the first two pions are kept together while the third one is separated, but it also possible to separate all of them. The third usually have a $\mathrm{p_T}$ lower than the first two pions, so in the background events its track is usually a random one and also background events have often the third pion not isolated.

As the proposed formula is empirical, it needs to be tested before being used to estimate the background efficiency numerator. This involves verifying if it predicts correctly. To test it, the two sides of the equation (2.16) are compared, in the case the selection is smoothed in order to have $\mathcal{N}_{(12)(3)} \neq 0$. In this context, the formula is tested using both of the two discussed filters. It's worth noting that there are three different scenarios under investigation:

- transverse momenta lower bound
- isolation on specific pions
- combination of both isolation and transverse momenta filter

In the first case the three lower bounds are loosened when "releasing" the constraints. For example, $\mathrm{N}_{(12)(3)}$ indicates that the chosen $\mathrm{p_T}$ lower bounds are selected to be $(18, 15, 12)$ as the standard case, while for the case $\mathrm{N}_{(12)(\cancel{3})}$ the values are $(18, 15, 2)$.

As mentioned earlier, the case where only the isolation parameter is varied performs better, and here are the results, starting with the formula testing and then its application to estimate the background efficiency. It's important to note that the other filter parameters remain the same as initially described.

The formula is tested keeping the lower bounds for the transverse momenta as $(18, 15, 12)$ in order to allow the algorithm to select a number of events which is different from zero. As describe before, the notation $\mathcal{N}_{(12)(3)}$ means that all the three pions are required to be isolated. The algorithm in this case select $\mathcal{N} = 1$ event. On the other hand, using the empiric formula, it predicts $\mathcal{N}_{(12)(3)} = 1.34$ events. The two results, representing the same number, are comparable. Then, the formula predict correctly, still being an empiric one.

Finally, if it perform correctly, it can be used to calculate the value of the background efficiency $\varepsilon_{\mathrm{bkg}}$, providing an indicative estimate. This step can be also seen as a rescaled case of the test scenario, where the only difference lies in the lower bound values. Indeed, the value of ε_{bkg} has to refer to the same version of the algorithm as the one used for the signal. For these reasons, the lower bound are maintained to $(20, 15, 12)$ and the formula predicts the following:

$$\mathcal{N}_{\mathrm{events}} = \mathcal{N}_{(12)(3)} \simeq 0.425, \tag{2.17}$$

which is a number smaller than 1 as expected, but not as infinitesimal as hoped. Then, the background efficiency is calculated:

$$\varepsilon_{bkg} \simeq 2.19 \times 10^{-7} \tag{2.18}$$

With this result, the number of expected background events within the condition previously described is, from equation (2.15):

$$\mathcal{N}_{\mathrm{bkg}} \simeq \mathcal{R}_{\mathrm{MB}} \times \mathrm{T}_{\mathrm{DAQ}} \times \frac{\mathcal{N}_{(12)(3)}}{\mathcal{N}_{\mathrm{tot}}} = 3.68 \times 10^{7}. \tag{2.19}$$

2.3.1 Signal-to-noise ratio

Once both signal and background expected events are estimated, the signal-to-noise ratio can be deduced. In particular, its value depends one the ratio between the two expected events numbers. Note that both the two estimation has to be done considering the same conditions, e.g. the period of data acquisition. So, using the results in eq. (2.19) and (2.14), the signal-to-noise ratio is:

$$\frac{\mathcal{N}_{\mathrm{sig}}}{\mathcal{N}_{\mathrm{bkg}}} = 1.56 \times 10^{-5}. \tag{2.20}$$

When the number of the background events is very greater than the signal one, another useful parameter to estimate is the sensitivity of the analysis, which is another version of the snr and it's defined as follow:

$$\frac{\mathcal{N}_{\text{sig}}}{\sqrt{\mathcal{N}_{\text{bkg}}}} = 9.50 \times 10^{-2}. \tag{2.21}$$

The square root in the denominator takes into account the statistical uncertainty associated to the background events. When the noise is significant, this parameter might be a more indicative estimate of the signal-to-noise ratio. The first result show that a single signal event corresponds to 10^5 background events, providing an estimate of the rareness of the decay. On the other hand, the analysis has a sensitivity on the order of 10^{-1}, which means that the signal can be detected with a significance of a similar order of magnitude to the background noise, indicating that the ability to distinguish the signal from the noise is modest. Nevertheless, improvement in the selection is needed. It's important to recall that these estimation are rough and based on the assumption that the branching ratio is on the order of 10^{-7}, while previous studies show an upper limit of 10^{-6}. Moreover, the results can be improved by a deeper study on the filters the variables which the selection is based on. Indeed, some correlation between the pions and the parent W boson can be explored and used as a discriminator for the signal. An example might be the transverse momenta of the W boson. Another approach could be implementing neural networks helping the discrimination of the signal to the background. Different approaches can be explored to achieve a higher signal-to-noise ratio. For example, the background efficiency or parameterization can be improved using a estimation method which differs from the one proposed in the thesis in eq.(2.16). Also using a bigger statistic dataset can be useful, both for the signal and the background estimation.

2.4 Set of an Upper Limit on the branching ratio

After estimating the expected signal-to-noise ratio and defining a computing strategy for the expected background events, the next section focuses on the statistical analysis required for computing upper limits. The primary concept behind this strategy is that, when the expected signal is not likely to be observed with the current amount of data, the signal can be enhanced by introducing a parameter, denoted as "signal strength" (often denoted as μ), until it becomes discernible at a certain level of significance. Techniques such as "Toy Monte Carlo" may be employed for this purpose. This approach is necessary in the case of the $W \to 3\pi$ analysis since, based on the signal-to-noise ratio and the provided figure, it appears that the signal cannot be distinguished from the background. Consequently, upper limits are usually computed under the assumption of a background-only hypothesis. The section begins by briefly introducing this technique and subsequently presents the final results obtained by applying an algorithm called "combine," which implements this statistical technique.

2.4.1 Upper limit statistical concepts

In the following section, within the context of the statistical technique, the background and signal models are represented as b and s, respectively. The parameter sets describing these models are denoted as θ_B and θ_s. The combined model is constructed from these two models and can be expressed as follows: $\sigma(\theta) = B(\theta_B) + \mu \cdot s(\theta_s)$. In this expression, $\theta = \theta_B, \theta_s$, and the signal contribution is scaled by the previously introduced signal strength parameter, μ. This model corresponds to the observed model when $\mu = 0$. Therefore, the following provides a step-by-step explanation of the method routine for computing upper limits. For a more comprehensive understanding of the technique and the theory behind it, refer to G. Cowan et al. [49].

It's important to note that this explanation offers a general overview of the statistical routine. In this context, the term "observed data" refers to the data obtained from simulations, as real data have not been acquired yet. Nonetheless, the adjective "observed" distinguishes this data from "expected" data. Typically, this statistical approach is applied when both real (observed) and simulation (expected) data are available, with the latter used for modeling background or signal components.

To compute the observed upper limit at a certain confidence level (usually a typical value is $\alpha = 95\%$), the following steps are applied:

- first of all a test statistic q_μ is constructed on the basis of the likelihood function $\mathcal{L}(\mu, \theta)$. A common choice for LHC related analysis is the **negative log-likelihood ratio**, which is a quantity used in statistic to estimate how much a statistical model describe the data with respect to another one. Its general definition is:

$$q_\mu = -2\log\left[\frac{\mathcal{L}(data|\mu, \hat{\theta}_\mu)}{\mathcal{L}(data|\mu = \hat{\mu}, \hat{\theta})}\right]. \tag{2.22}$$

  The logarithm contains the ratio between the two likelihood of the data under two different statistical models, both are maximum likelihood obtained by a fit of the data, the numerator with floating θ parameters and a given μ while the denominator with a μ set to the best-fit value $\hat{\mu}$.

- For each given value of μ, the test statistic is computed on the observed data and the result is denoted as q_μ^{obs}.

- To simplify the procedure, analytic approximations of the *Probability Density Functions* (PDFs) of q_μ derived in the asymptotic case of a large number of events can be used, as in practice they're found to be accurate enough already for just a few expected background events. In particular, the PDFs refers to two models: *background-only* hypothesis ($\mu = 0$) $f(q_\mu|0, \hat{\theta}(0, \text{data}))$ and *background plus signal* hypothesis (with $\mu > 0$) $f(q_\mu|\mu, \hat{\theta}(\mu, \text{data}))$.

- From the two constructed distributions of q_μ, the corresponding p-values are computed as:

$$\begin{aligned} p_\mu &= \int_{q_\mu^{\text{obs}}}^{\infty} f(q_\mu|\mu, \hat{\theta}(\mu, \text{data}))dq_\mu, \\ p_b &= 1 - \int_{q_\mu^{\text{obs}}}^{\infty} f(q_\mu|0, \hat{\theta}(0, \text{data}))dq_\mu. \end{aligned} \tag{2.23}$$

- Lastly, the CL_s criterion [50, 51] is applied by computing:

$$\text{CL}_s = \frac{p_\mu}{1 - p_b} \tag{2.24}$$

  and iterating the previous steps for several values of μ in order to find that value $\mu_{\text{up}} \equiv \mu_{\text{obs}}$ such that $p_{\mu_{\text{up}}} = \alpha$.

Thus, the observed upper limit on the branching ratio will be the nominal value used for all the estimations in (2.13), multiplied by μ_{up}.

2.4.2 Final results

The CLs asymptotic limit method is applied to get the final results of this work. An upper limit on the branching fraction is set on 95% of confidence level to:

$$\mathcal{B}(\mathrm{W} \rightarrow 3\pi) < 1.04 \times 10^{-6}. \tag{2.25}$$

This is the observed upper limit, where the simulation data are considered as the real observed data, as previously said. Nevertheless, the result is comparable with the Run2 studies reported at the beginning of the chapter in sec 2.1.

Chapter 3

The Online Selection Demonstrator

The third chapter of this thesis delves into its primary objective: demonstrating the feasibility of an **online selection** based on the correlator layer data flow. The second chapter provided an example of a physics analysis that can be performed using data simulating the ntuple of L1 correlator layer 2 (CTL2) output in the Phase 2 configuration, which includes L1 Puppi candidates. The term "online selection" refers to the process of performing signal selection during LHC bunch crossings, thereby extrapolating certain events or particles using L1 scouting data. Such candidate events would then be saved for a deeper offline analysis. This approach necessitates working at a rate comparable to the LHC's 40MHz collision rate.

This "trigger" approach represents a novel technique because it typically involves two-level triggering, where interesting events are initially selected and then analyzed *offline*, i.e., after being stored. However, this conventional method ensures a high level of accuracy and reconstruction performance for particle tracks but discards a huge number of events. Indeed, the output rate is reduced by the trigger system from 40MHz to about 1kHz, implying the rejection of a significant amount of events. On the other hand, the scouting approach seeks to perform signal selection using the data available at Level 1, thereby saving events that may be less precise than offline selections but with a larger amount of data for analysis. Reason why its feasibility needs to be demonstrated.

Nonetheless, the data coming from the correlator layer initially exist as raw data, as explained in the first chapter 1. This means that the data must be translated and converted into a higher-level format before the selection process can take place. This additional step requires some time and resources. In terms of selection, the algorithm discussed in 2.2 is employed, with the goal of identifying a candidate pion triplet originating from a W boson decay. This type of process is particularly well-suited for the scouting approach because it is a rare event and often limited by the available statistics, as indicated in previous studies such as [40]. By using the online selection approach, a larger dataset can be analyzed due to the high working frequencies.

This chapter addresses the steps required to perform an online selection, starting from the correlator layer output. It describes how to unpack and construct the ntuple, carry out the selection process, all while adhering to the constraints of high computing rates, and details how to optimize these processes. Additionally, the available technologies are discussed. Generally, the chapter explores the steps involved in constructing the following *chain*: emulating the correlator layer data flow, unpacking the data, selecting and saving candidate events. Each of these processes is previously described independently, then an explanation of how they are interconnected follows. Next chapter will then give a more general overview.

3.1 Correlator layer 2 emulator

The first section focuses on the development of an emulator for the correlator layer 2 (CTL2). To perform any selection or action on the data, it is essential to generate correlator data flowing at a realistic rate (40MHz). The primary objective is to generate events that contain particles with features matching the distributions of the Puppi candidates in the signal dataset described in the previous chapter (sec 2.2). The desired event multiplicity should align with that of the dataset, but it should also take into account the multiplicity of other background event distribution. So, it is important to consider the flexibility to change it to keep the possibility for other studies.

The emulator is implemented as a firmware algorithm running on a prototype Phase-2 trigger board of the correlator layer, named "Serenity". This algorithm must consider the interface and connections of the board with a computer responsible for collecting and processing the generated data. These connections are mediated through a **DAQ Timing Hub** (DTH). The entire system, along with its various connections, will be described in detail throughout this section.

The section commences with a detailed description of the trigger board and the hardware system that will host the emulator. It then proceeds to elucidate the format of raw data, providing insights into how an event is represented in a bit format and how it aligns with the instrumentation links. Following this, the section provides a comprehensive characterization of feature distributions and subsequently describes various generation techniques. These discussions are framed within the context of the FPGA environment on the trigger board, as introduced earlier in the section. To conclude, the section offers an overview of the generated data flows, tracing their path from the correlator board to the computer. This overview sets the stage for the subsequent section, which delves into the data reception and unpacking process.

3.1.1 System Description

Before delving into the development of the emulator, it is crucial to establish an understanding of the environment surrounding the trigger board in the correlator layer. Furthermore, it is important to elucidate the data flow from this board to the PC, where essential online selection processes occur, such as the unpacking of raw data and the selection algorithm, which will be described in Section 3.2.

To provide an overarching perspective of the system, Figure 3.1 offers a schematic representation. The trigger board establishes a connection with the DTH (Data Acquisition Timing Hub) board through a custom Level-1 trigger protocol implemented on top of the 25 Gbps ethernet standard. Subsequently, data packets are transmitted from the DTH to the PC via TCP/IP. The DTH serves as an intermediary, enhancing the reliability of the system. This setup ensures that data packets are transmitted in the correct order. The choice of the TCP/IP protocol is integral to enabling effective communication between devices within this network. TCP (Transmission Control Protocol) is responsible for the secure and ordered delivery of data packets. IP (Internet Protocol) manages the routing of these packets between devices on the network. Consequently, the TCP/IP connection guarantees the delivery of packets in the correct sequence but also it guarantees also that they are received by the PC. Indeed, packets can be buffered or re-transmitted if the PC is late in receiving them or loses them.

An important consideration is that the trigger board lacks the necessary memory to store packets after transmission. Thus, direct transmission from the trigger board using these connections is unfeasible. In contrast, the DTH possesses ample memory capacity and can effectively perform data transmission. The DTH, therefore, operates as a buffer, temporarily storing the puppi candidates. This architecture facilitates the flow of data from the trigger board to the PC.

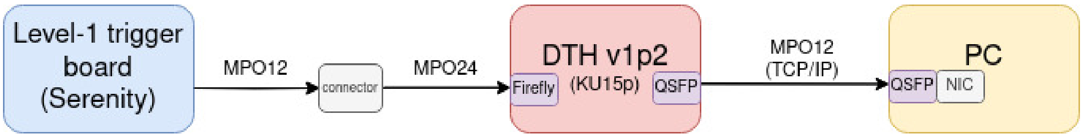


Figure 3.1: *Block diagram of the hardware system setup, from the L1 trigger board to the PC passing through the DTH.*

The hardware setup comprises two main boards: "Serenity," a standard Level-1 trigger board, and "DTH p1v2," a prototype Data Acquisition (DAQ) board designed for phase 2 of the CMS experiment. Specifically, the "Serenity" board is a versatile board compatible with most sub-detectors developed for the CMS phase-2 upgrade. Additional information about this board is available on its website[52].

One of the critical components common to both boards is the FPGA (Field Programmable Gate Array) within them, with both featuring the `KU15p` model. The FPGA serves as the central processing unit, managing all data input and output flows, as well as overseeing various board components. The emulator is constructed as a firmware algorithm that runs on the trigger board, implying that it will program the FPGA embedded in the trigger board. Consequently, the design of the emulator must take into account the pivotal role played by the FPGA in the overall operation of these boards.

The data transmission process involves several key components and technologies. Initially, data packets are transmitted from the Serenity board to the DTH board via an MPO12 breakout cable, which is further connected to an MPO24 breakout cable. The Multi-Fiber Push-On (MPO) connectors are designed to accommodate multiple optical fibers within a single connector, enabling the simultaneous transmission of multiple optical signals. A breakout cable is employed to separate the individual fibers within the MPO connectors. Using two breakout cables is necessary to properly match the active fibers of the MPO12 to the right channels of the MPO24. Subsequently, the data packets are received on the DTH board using a Firefly, a low-profile optical connector and high-speed optical interface commonly used in data communications applications. The Firefly receives high-speed serial optical data and converts it into electric signals, or vice versa. Finally, after processing within the DTH board, the data packets are transmitted from the DTH to the PC using two QSFPs (Quad Small Form-Factor Pluggable connectors), which are compact optical connectors designed for the high-speed transmission of data.

It's worth noting that the trigger board is equipped with four links or channels, which correspond to the MPO12 cables. Each link is connected to a specific port on the trigger board and is capable of carrying a payload containing up to 52 candidates, every 6 bunch crossings, where each candidate consists of 64-bit words. Specifically, the trigger board outputs 52 candidates in each link *every 6 bunch crossing* (150ns), meaning that each event is composed by 52 frames and the FPGA clock rate is 360 MHz.

This information plays a crucial role in the development of the emulator and is elaborated upon in the following sections.

Lastly, regarding the PC used for the two computing tasks (unpacking and selection), it is a `AMD Ryzen 9 5950X` 16-Core hyper-threaded, meaning that it has 16 physical cores and 32 logic ones.

3.1.2 Puppi row format

Before delving into the study and generation of feature distributions, it is essential to introduce the event's raw data format, as this understanding is crucial for explaining the event generation process. Knowing how an event is constructed is fundamental for generating events.

In this context, each particle is represented as a 64-bit word. Consequently, each event is composed of a set of N 64-bit words, with N representing the number of particles in that event. Notably, each 64-bit Puppi candidate has its first 40 bits that are common to all candidate types, while the remaining 24 bits vary between charged and neutral objects.

As mentioned in section 3.1.1, the FPGA is a central component of the trigger board. It's vital to understand that the FPGA efficiently handles integer data types, such as `int`, with high time efficiency, while it doesn't perform as efficiently with floating-point data types, such as `float`. In practice, trigger boards typically work with integer data types represented in a raw data format.

Therefore, all the information contained within the 64-bit words is presented in an integer format. However, it's important to note that features like p_T, η, and ϕ are essentially discrete decimal numbers. To convert this information from the integer format output by the trigger boards to the floating format used for analysis, a concept is introduced: the *last significant bit* (LSB), which is equal to the sensitivity of the features measurement. For instance, for the transverse momenta the detector is sensible to a change in the measurement to a minimum of 0.25 GeV. The LSB will play a fundamental role in the subsequent feature generation, as discussed in the following section.

Common info for all candidate types				
Bits	Field	Size	Format	LSB
13-0	p_T	14	unsigned int	0.25 GeV
25-14	η	12	signed int	$\pi/720 = 1/4$ deg
36-26	ϕ	11	signed int	$\pi/720 = 1/4$ deg
39-37	PID	3	see table 3.2	

Charged candidates specific info				
Bits	Field	Size	Format	LSB
49-40	z_0	10	signed int	0.5 mm
57-50	d_{xy}	8	signed int	
60-58	quality	3	tbd	
63-61	*unassigned*	3		

Neutral candidates specific info				
Bits	Field	Size	Format	LSB
49-40	w_{Puppi}	10	unsigned int	
55-50	e/γ id	6	unsigned int	
63-56	*unassigned*	8		

Table 3.1: *Table describing the composition of the 64-bit word of the puppi candidates. For each particle information, the corresponding bits are reported in the first row. It worth noting that the features change for a charged or a neutral particle and also that each information is registered in an integer format. The empty spaces usually are due to the fact that the numbers are still to be defined or they are integers.*

After having recognized the significance of the integer number format, Table 3.1 provides a detailed breakdown of how the bits are allocated among different features and the information contained within each 64-bit word. Additionally, it specifies the LSB for features which is needed for the conversion from integer to float. The first column of the table indicates the bits dedicated to that specific information. For instance, the first 14 bits are allocated to transverse momenta information. Concerning particle id, Table 3.2 explains the formalism of the bits reserved for it and their conversion from raw data to the actual PF format. It's worth noting that Table 3.1 indicates that some bits in the Puppi words are yet to be assigned. This is due to ongoing work in characterizing the Puppi format for the phase-2 upgrade, and as such, these tables present preliminary results. It worth noting that the 64 bits have a different structure for charged and neutral particles. Indeed, Table 3.1 shows that the candidates share the same initial structure dedicated to the general common features ($\mathrm{p_T}$, η, ϕ, and PDG), then charged particles contains different information with respect to the neutral ones, since dealing with charged particles allows to measure different information than dealing with neutral particles.

Val	Binary	Particle		PDG ID
0	000	h^0	neutral hadron	130
1	001	γ	photon	22
2	010	h^-	hadron of charge $-$	$-211(\pi^-)$
3	011	h^+	hadron of charge $+$	$+211(\pi^+)$
4	100	e^-	electron	$+11$
5	101	e^+	positron	-11
6	110	μ^-	muon	$+13$
7	111	μ^+	anti-muon	-13

Table 3.2: *The table reports specifications about how the PID bits are constructed and how to convert them from the row to the real format.*

When presenting the Puppi raw data format, it's crucial to emphasize that the emulator is responsible for generating Puppi candidates with this structure. However, it particularly focuses on the first four pieces of information shared among all candidate types, namely $\mathrm{p_T}$, η, ϕ, and PDG ID. As for the other pieces of information, their corresponding bits are populated with random placeholders. This specific approach is chosen because the selection algorithm, as described in the previous chapter, only depends on these primary features.

Concerning the additional information, z_0 represents the distance of the particle track's origin from the primary vertex's z-coordinate, which is along the beam direction. On the other hand, d_{xy} signifies the distance in the transverse plane.

After detailing the construction of a single Puppi candidate, it's important to note that each event, comprising N particles, is represented as N 64-bit words, preceded by a single 64-bit header word. The event header format is presented in Table 3.3. As illustrated in the table, the first 8 bits are allocated for indicating the number of particles in that specific event. This means that the header is followed by this number of 64-bit objects.

As described in Section 3.1.1, the trigger board provides a payload containing up to 52 candidates. Since the particles are transmitted through 4 links, there are 4 available payloads, potentially allowing for a maximum of 208 particles and 1 header to be emitted in a single event or bunch crossing.

Consequently, after filling the first bits of the event with the candidates, a certain number of bits are still available and outputted. These remaining bits are all set to zero. To summarize, each event begins with a header word, followed by N 64-bit words, and then concluded by a series of null words, which continue until a new event begins. It's essential to emphasize that the header is defined by the trigger board, which maintains the count of all event characterization numbers, such as BX run and orbit.

Header bit format		
Bits	Size	Meaning
07-00	8	Number of Puppi candidates
11-07	4	must be set to 0
23-12	12	bunch crossing number (0-3563)
55-24	32	orbit number
60-56	6	(local) run number
61	1	error bit
63-62	2	10=valid event header

Table 3.3: *The table illustrates how the event header word is defined. This word, which precedes each event, contain general information about it such as the number of puppi candidates in it, which gives an essential information since gives the number of meaningful 64-bit words that follow it.*

3.1.3 Feature distribution study

The emulator's objective is to generate events comprising particles with features that align with the distribution of Puppi candidates contained in the dataset used in the second chapter to estimate the number of signal events, i.e., the dataset containing the signal. This choice is motivated by the necessity of having a simulated events which should contain a signal one in order to make more accurate estimation in the performance testing of the demonstrator. In this way, the online selection step may select some candidate triplets. Regarding the number of particles in each event, it should follow the multiplicity distribution observed in this dataset, but considering also the distributions of other background dataset, in order to provide events with a more general multiplicity. The significance of this lies in the fact that the multiplicity embedded in the simulations may be inaccurate, attributed to uncertainties in the QCD models, particularly within a high pileup energy scenario. Additionally, the simulations of the detector and track finder might underestimate factors such as the number of fake tracks. Therefore, having the flexibility to explore various multiplicity scenarios is useful for a comprehensive study.

Before delving into the generation technique, which involves random number generation based on specific distributions, it is essential to examine and discuss these distributions. The features considered for each particle are consistent with those used in the selection algorithm, including transverse momentum ($\mathrm{p_T}$), pseudo-rapidity (η), azimuthal angle (ϕ), and particle identification (PDG ID). Lastly, the multiplicity distribution is also taken into account.

A critical point to emphasize, is that all features are treated as **independent** in the generation process. This means that each particle is composed of features without considering their inter-dependencies. This simplification is made for the sake of efficiency, even though, in reality, high-$\mathrm{p_T}$ pions are more likely to exhibit certain values of η and ϕ. This assumption is introduced in this paragraph as the distributions are studied in their entirety, looking at the distribution of all the particles while ignoring their other properties. This critical aspect will be reiterated later, in particular when discussing the online selection step since it will affect the efficiency (ε) of the selection algorithm.

Transverse momenta Figure 3.2 illustrates the histogram of the transverse momenta distribution in logarithmic scale. The same distribution not in logarithmic scale can be seen in Figure 4.4, in the appendix B 4.2.1. The choice of plotting in logarithmic scale is motivated by the fact that the majority of the entries fall in the lower values of p_T, so in this way also the bins corresponding to higher values can be observed. Lastly, the distribution is plotted for transverse momenta values less than 50 GeV. This assumption is important also for the next chapter when dealing with the generation of such values, since it is important to focus on this part of the distribution since it contains the 99.593% of the entries.

Looking also at the figure 4.4 it can be observed that the initial part of the distribution (low p_T values) is similar to a decreasing exponential, then around $p_T = 10$ GeV, the slope changes starting to be approximately constant. As expected the entries for the particles having $p_T > 50$ GeV is notably lower than the other values. A large number of values fall in the range of 2GeV $< p_T <$ 10GeV.

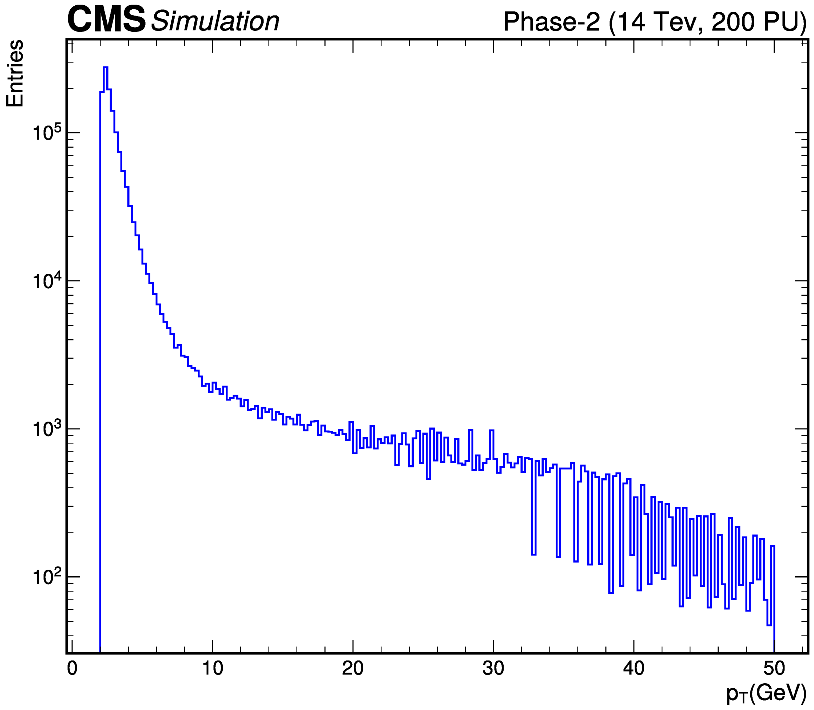


Figure 3.2: *Histogram illustrating the distribution of the transverse momenta contained in the simulation in a logarithmic scale, plotted in the range from 2 to 50 GeV. The logarithmic scale allows to distinguish better also the bins having a low number of entries.*

Pseudorapidity and azimutal angle Figure 3.3 illustrates the histogram of the η distribution in a range from -6 to $+6$. First of all, as expected the distribution is symmetric with respect to zero. It also can be distinguished the acceptance range (2.5) since the majority of the values falls in the region $|\eta| < 2.4$. In particular, only the 2.8% of the values fall outside the acceptance range. This behaviour is due to the fact that the tracker acceptance boundary is not exactly 2.4 and depending on the Z position of the vertex (the bunch crossing position) there may be traces with higher η values that still fall within the acceptance.

Nevertheless in the acceptance region, the distribution is almost uniform in the central region for values comprises in the range of about $|\eta| < 1.5$. Then, in the remained region $1.5 < |\eta| < 2.4$ the distribution shows more particles having such values of η, but still having an uniform trend within it. This trend is probably due to detector effects.

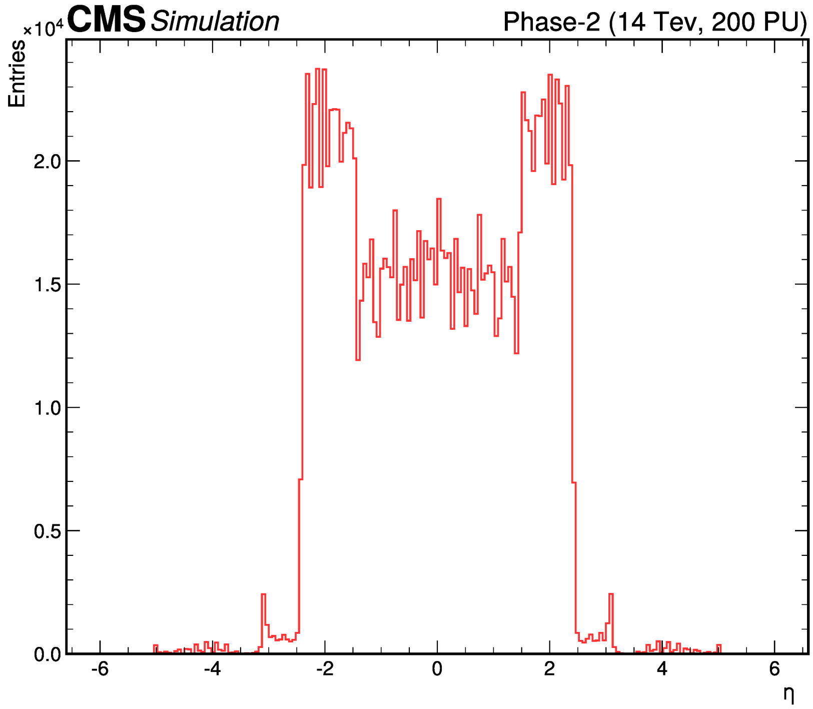


Figure 3.3: *The histogram illustrates the pseudorapidity distribution of the particles in the simulation. The distribution is symmetric with respect to zero, then two ranges can be distinguished: outside and inside the detector acceptance region (*$|\eta| < 2.4$*), the latter containing almost the* 98% *of the values.*

On the other hand, the ϕ distribution showed in Figure 3.4, is a clearly uniform distribution in the range $[-\pi, \pi]$, which is exactly the definition range of the azimutal angle.

Particle ID Lastly, the characterization of the particle Id distribution is performed by giving simply the frequentist probabilities of finding a particular particle species, i.e. the fraction of a species with respect to the total number of particles. Specifically, the type of particles contained in the simulations are exactly the ones reported in Table 3.2, namely charged or neutral hadrons, muons, electrons or positrons and photons. Table 3.4 shows the probabilities of the such particles population, where each of them is calculated as the ration between the number of a specific species and the total number of particles.

Event multiplicity Figure 3.5 shows the comparison between the multiplicity distribution of the two only-pileup simulations and signal dataset. For the neutrino gun and the signal dataset the mode is about 30 particles per event, representing a typical low multiplicity kind of events. On the other hand the ttbar dataset contains a higher multiplicity events, with an average of about 55 particles per event. As introduced before, it is essential to consider the distributions of every available dataset, since the multiplicity contained in the simulations may not be correct.

3.1.4 Features Generation

After examining the distributions of the features, the focus now shifts to their generation. The process involves simple random number generation, with slight variations

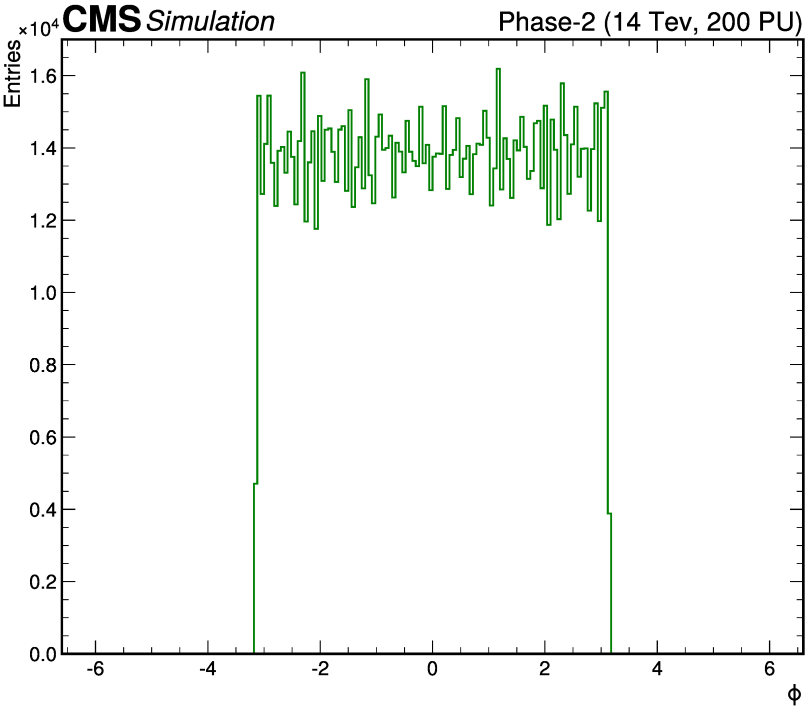


Figure 3.4: *The histogram illustrates the azimutal angle distribution of the particles in the simulation. The distribution is uniform in the range between* $[-\pi, +\pi]$

for each feature. It's crucial to note that the generation occurs within the FPGA environment. Random number generation in the firmware context is a challenging problem and not trivial to address. Despite this, a specific algorithm called `Xoshiro256` is employed, producing a single 64-bit number. This constraint means that the technique for generating feature numbers has to deal with the availability of such random. Additionally, due to the FPGA environment, the generated numbers are integers, as previously discussed in Section 3.1.2.

The description of how the features are generated is categorized into three paragraphs, each corresponding to a specific method used, aligning precisely with the subdivision from the preceding paragraph. This division is associated with the different distribution shapes.

Transverse momenta For the transverse momenta generation, the method used relies on the inverse of the Cumulative Density Function (CDF), sometimes called quantile function. This method is usually exploit for the generation of random number following a particular PDF. In general, the sampling takes uniform samples of a number u, between 0 and 1, interpreted as a probability, and then returns the smallest number $x \in \mathbb{R}$ such that $F(x) \geq u$ for the cumulative distribution function F of a random variable.

Since the transverse momenta distribution (Figure 3.2) as introduced before doesn't represent a particular known function, the evaluation of the CDF and its inverse function has to be performed from scratch, considering the distribution as discrete.

The objective is to compute the inverse of the Cumulative Distribution Function (CDF). First, the CDF is calculated by following the definition, which involves performing the cumulative sum of the bins in the histogram of the transverse momentum (pt) distribution, namely the Probability Density Function (PDF). Subsequently, the CDF is normalized to ensure it ranges between 0 and 1, as it must represent a probability.

Particle		PDG ID	Probability
h^0	neutral hadron	130	6.96%
γ	photon	22	5.58%
h^-	hadron of charge $-$	$-211(\pi^-)$	42.1%
h^+	hadron of charge $+$	$+211(\pi^+)$	43.0%
e^-	electron	$+11$	0.62%
e^+	positron	-11	0.69%
μ^-	muon	$+13$	0.40%
μ^+	anti-muon	-13	0.41%

Table 3.4: *The table reports the frequentist probabilities of the particle Id, i.e. the different particle populations. Each of them is calculated as the ratio between the the number of a specific species and the total number of particles*

To generate numbers, an array of n values ranging from 0 to 1 (y-value) (representing probabilities) is created. For each value, the bin (x-value) corresponding to the CDF[i] and CDF[i+1] range is determined, namely the inverse CDF value $F^{-1}(p)$. This process constructs an array of n elements, one for each value of p. These elements correspond to the inverses of the CDF and are used to fill a lookup table (LUT) for the number generation. Finally, the generation process involves selecting a number from 0 to n and retrieving the corresponding element in the array, which represents the desired transverse momentum value. It is important to recall that this lookup table has to be converted in an integer format, through the LSB reported in Table 3.1.

In the context of the FPGA, the Lookup Table has to be storage in its memory, so a proper size value has to be chosen in order the LUT to fit in the memory, without affecting negatively the performance of the hardware processor. This means that for the generation of a transverse momenta value, a random number ranging from 0 to 1023 is needed, namely a 10-bits number. This particular and limited number generation method implies a maximum of 1024 different possible numbers. So, this method is very limited and also very affected by the approximations due to the conversion from float to integer format. This is also the reason why the distribution is considered only in the range $2 < \mathrm{p_T} < 50\mathrm{GeV}$. Indeed, since the number of possible different values that can be generated is low (1024), it is better to concentrate on the range in which the majority of values fall. In the appendix B, figure 4.5 illustrates the comparison between the generated and the real transverse momenta distributions.

Pseudorapidity and azimutal angle As mentioned in the preceding paragraph, the η distribution can be segmented into two regions where the histogram is uniform. However, for the sake of simplicity, a single uniform distribution is considered for the generation within the range $|\eta| < 2.4$.

The generation of a number following a uniform distribution is simple: a random number is generated and then it is converted in order to fall inside a particular range. As before, is important to recall that only a 64-bit random number is available, part of it can be used for this purpose. Regarding the range, is important to recall that the values has to be converted into an integer format. This implies that the range $-2.4 < \eta < 2.4$ becomes, using the LSB in table 3.1: $-550 < \eta < 550$. Although the distribution of the generated η values does not follow exactly the real one, it represents a good approximation.

Same procedure is used for the generation of ϕ values. In particular, its distribution 3.4 is exactly a uniform distribution in the range $[-\pi, \pi]$, which becomes $[-720, 720]$ in

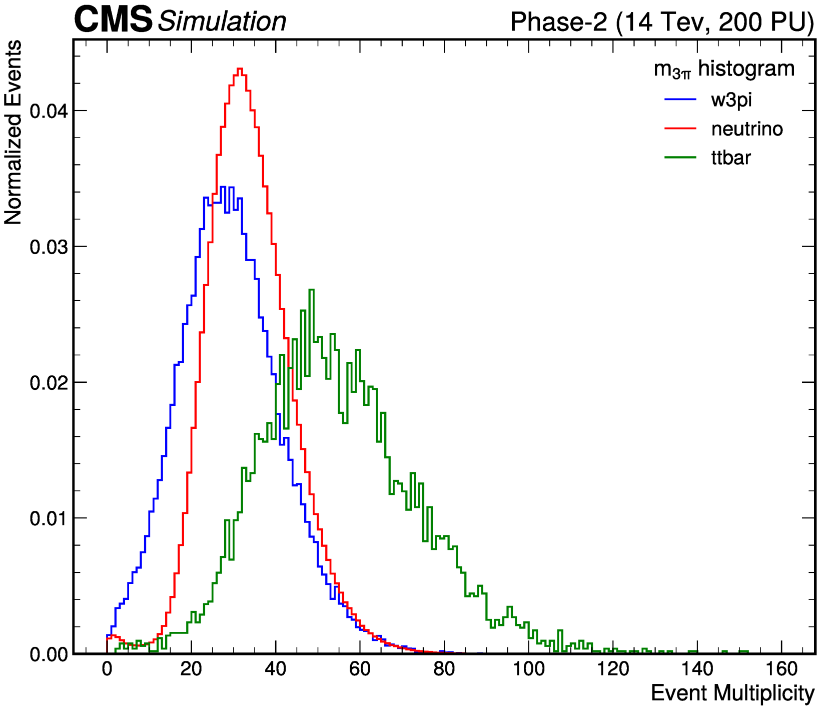


Figure 3.5: The histogram illustrates the normalized distribution of the event multiplicity in the three available datasets. In blue the distribution of the MC simulation containing the signal, in red the simulation of low-multiplicity events called neutrino gun, in green the distribution of high-multiplicity events called "ttbar". The first two distributions present similar values for the number particles per event. The third shows higher values presenting some events having also more than 100 *particles.*

an integer format.

In the appendix B, Figure 4.7 and Figure 4.6 show respectively the comparison between the generated and the real pseudorapidity and azimutal angle distributions.

Particle ID The generation of the particle type relies on the cumulative distribution function (CDF), similar to the case of transverse momenta. The probabilities in Table 3.4 can be interpreted as a discrete probability density function (PDF) with 8 values. To generate the particle ID, a random number is generated, serving as a probability, and the inverse CDF procedure, described earlier, is applied. In this case as well, the generated numbers need to be integers. The CDF values are multiplied by 1000, and the probability is generated in the range between 0 and 1000. This approach allows storing only the CDF values in the FPGA. By comparing the generated probability with the CDF values, the particle ID is determined.

Event multiplicity Finally, the distribution of the number of particles can be characterized by a log-normal function of the following form:

$$\mathrm{N} = \mathrm{N}_0 \cdot k^{\theta} \tag{3.1}$$

where N_0 and k are two real parameters that characterize the function, while θ is a random Gaussian variable $\theta \sim \mathcal{N}(0, 1)$. First of all, the two parameters N_0 and k need to be determined in order to describe the distributions. Indicative values can be provided by the median for N_0. As mentioned early, it's important to handle the number of generated particle in order to have the flexibility to change this number, i.e. the distribution. For this purpose, the role of N_0 is central. For instance, increasing it the peak of the distribution can be moved to higher values, meaning a slightly greater multiplicity.

For the initial estimation of the two parameters, the starting reference distribution is that of the signal dataset, and the characterization of the other datasets is also performed (Figure 3.5).

To generate number from such distribution, a lookup table (LUT) is found similarly to the transverse momenta case. However, the LUT here reports 2048 values of k^θ, where $k = 1.45$. Then, the LUT is saved in the FPGA and for the generation of N, a random number from 0 to 2047 is used to select the elements of the LUT, which is then multiplied to the chosen value of N_0, that can be changed, for the previously described reasons. In particular, the generated multiplicity has to be checked if is less than the maximum supported multiplicity, i.e. 208:

$$\mathrm{N} = \min\{\mathrm{N}_0 \cdot k^\theta, 208\} \tag{3.2}$$

In the appendix B, Figure 4.8 shows the comparison between the generated and the three other distributions. The choice made is to generate a mixture of the neutrino gun and signal dataset distributions.

3.1.5 Event generation

Now, all the tools are described to finalize the emulator of the correlator layer 2. For each event, the number of particles is generated first, and then N 64-bit objects are constructed. The first 40 bits contain the p_T, η, ϕ, and PID, while the remaining bits are filled with random numbers that serve as placeholders. Subsequently, the constructed candidates are shared among 4 links, implying that 4 fluxes have to be constructed as output. To handle these fluxes, each particle carries 4 extra bits called "control bits," which help the receiver understand and manage the incoming flux more effectively. These control bits are fundamental for the system to work, so they have to be generated together with each generated candidate. From left to right, the bits are: orbit, valid, start, end.

The **orbit** bit is equal to 1 for the first 4 particles of each orbit, indicating the start of the orbit. It is then zero elsewhere. Clearly, the 4 particles are correlated and refer to the 4 links. The **valid** bit indicates if a candidate is valid or not, meaning that the following word contains a particle or not. This is important, as previously explained, since the board emits 54 words for each event. After filling the N particles, the remaining spots are filled with null words. For such words, the valid bit is false. The **start** and **end** bits indicate the beginning of each event and its end. The first one is true only for the first 4 particles for that event, while the other is true only for the last 4 ones.

Similarly to the control bits, 4 bits can be given as input to the Serenity board. In particular, a general input interface can be established with it, having the same structure as the output, namely 4 streams or columns of 4 control bits with the corresponding 64-bit words. These input data are fundamental for the generation of random numbers, on which the feature generation relies. As introduced before, each feature generation method relies on a 64-bit pseudorandom number provided by a generator. However, it needs a seed for the production of a number. The input serves to set the seed. In particular, four generators are used, one for each link, so four seeds are needed. It is essential to highlight that setting the seed has to be performed only once to have different orbits from each other. This procedure is executed thanks to the control bits in input.

Finally, the emulator of the correlator layer 2 is build. It generates events at the bunch crossing rate, each populated by N puppi candidates. Their multiplicity adheres to the distribution ilustrated in 4.8, while their features align with those described in section 3.1.4. These candidates are subsequently passed to the DTH through four links and then to the PC through TCP/IP connection. Each event is prefaced by a header word, providing essential and general information about it. The following section delves into the receiving process, considering unpacking these row data coming from the DTH and preparing them for the selection.

3.2 Receiving and Unpacking

Upon the arrival of packets in the PC from the DTH, the subsequent step involves the execution of the selection process. However, prior to initiating the candidate triplet selection, it is imperative to unpack the data from its raw format and construct a higher-level structure (ntuple). This section elucidates this intricate process. Notably, two ntuple formats are under consideration to scrutinize their timing performances, facilitating the selection of the optimal one for online selection. As previously mentioned, the final two stages of the online selection chain, i.e., unpacking and the selection itself, must contend with the operational frequency of the LHC bunch crossing. Both ntuple formats adhere to the `ROOT` standards and are strategically chosen for compatibility with the `ROOT RDataFrame`, which is integral to the selection algorithm. The chosen formats include the conventional `TTree` and the innovative `RNtuple`, a new addition to Root's API, seamlessly compatible with `ROOT RDataFrame`.

Before fitting the events into the ntuple, the row data undergo an unpacking process. A designated size of data received from the DTH is examined after accumulating a specific number of orbits. This data is temporarily stored in the RAM disk of the computer, functioning as a buffer, awaiting the collection of a designated number of orbits before undergoing unpacking. This approach enables the CPU to concurrently manage multiple events, thus enhancing computing performance. It is imperative to note this methodology, as it signifies that data processing occurs not on an event-by-event basis but rather for a cluster of events.

For each event, the initial step involves reading the corresponding header and storing the number of particles within.

Occasionally, the header may solely consist of zero bit values, inserted by the DTH at the end of an orbit since the packets sent by the DTH must have a length which is a multiple of 256 bits (32 bytes, 4 words). Subsequently, the features of the candidates are reconstructed, considering their structure as outlined in Table 3.1. The physical version is then calculated from their integer format, utilizing the LSB parameter specified in the table. Notably, the unpacking procedure differentiates between charged and neutral particles, given the distinct structure of their respective 64 bits, as delineated in the table and elaborated in Section 3.1.2.

After the entire event is read and PUPPI candidates are reconstructed, the ntuples are constructed. It is crucial to emphasize that the event is written in its entirety to expedite timing performance. This approach ensures that the unpacking process generates a ROOT-format file comprising a specific quantity of events, or more generally, designated orbits. The resulting file is then stored in the RAM disk of the computer, providing a swift and efficient saving space. The forthcoming chapter will delve into the management of available RAM disk space, considering its limited storage capacity. The output file mimics the structure of the simulation data used in Chapter 2, containing solely the L1 PUPPI candidates' information, specifically the data required for the selection algorithm. It is noteworthy that regardless of the output file, it must be readable with `ROOT`'s `RDataFrame`, which both formats comply with.

3.2.1 Unpacking timing performance results

This section presents the results concerning the timing performance of the unpacking algorithm. It is crucial to emphasize that these measurements provide indicative results; the reported values represent average behavior obtained through multiple runs of the algorithm. If a performance plateau is observed, it is recorded. The unpacking algorithm is executed on various files containing different numbers of events, generated by collecting events from the correlator emulator. The performances are evaluated to understand how they vary with files of different sizes. Specifically, four distinct input files are chosen, with the last one representing the typical size of the targeted number of orbits. The reported times encompass the entire process of unpacking raw data and saving the file in the RAM disk for both file formats.

The algorithm is initially executed on a computer with a single core and a single thread, without enabling multi-threading, to measure the performance for a single CPU. Subsequently, the section reports the same results using multi-threading while keeping the file size constant. The critical parameters to estimate and study include not only the time required for the unpacking process but also the storage size of the output file, namely the size of the produced ntuple. The results are categorized by format type: Table 3.5 presents the results for `TTree`, while Table 3.6 presents results for `RNTuple`.

The estimated parameters include computing time, the working rate per event (calculated as the ratio between the number of events and the total computing time, to be compared with the LHC bunch crossing rate), the size of the input and output files, and the input and output data rates (computed as the ratio between the file size and the total computing time, aiming to estimate the amount of data flowing per second).

Size IN	**Events**	**Time**	**Event Rate**	**Size OUT**	**In Rate**	**Out Rate**
(MB)	**Number**	**(s)**	**(kHz)**	**(MB)**	**(MB/s)**	**(MB/s)**
1.42	5,941	0.05	117.1	5.13	28.0	101.0
3.58	14,850	0.07	220.4	12.87	53.1	190.7
14.41	59,994	0.15	402.7	51.67	96.0	344.4
285.54	1,188,595	2.03	586.9	1021.53	123.9	503.1

Table 3.5: *Table reporting the final results for unpacking the raw data and producing a ntuple in a `TTree` format. The unpacker performance is tested on files of different size. Last row refers to a typical-size file which is aimed in a realistic online selection.*

Size IN	**Events**	**Time**	**Event Rate**	**Size OUT**	**In Rate**	**Out Rate**
(MB)	**Number**	**(s)**	**(kHz)**	**(MB)**	**(MB/s)**	**(MB/s)**
1.42	5,941	0.05	118.0	4.89	28.3	102.6
3.58	14,850	0.13	113.7	12.32	27.4	99.3
14.41	59,994	0.52	114.7	49.56	27.5	99.9
285.54	1,188,595	10.45	113.7	982.8	27.3	95.5

Table 3.6: *Table reporting the final results for unpacking the raw data and producing a ntuple in a `RNTuple` format. Both reading and outputting is done on a Ram disk.*

Firstly, it is noteworthy from these tables that the TTree format outperforms the RNTuple format. This discrepancy is likely due to the latter being a newer format that may still undergo optimization. Nevertheless, its primary objective is to offer a faster interface for rntuple saving, so future ROOT versions may enhance its performance.

Table 3.5 illustrates that the unpacking algorithm scales with the size of the input file for the TTree format, i.e., it scales with the number of events.

This scalability is evident in the working rate values, which increase proportionally to the file size, indicating that the algorithm handles larger files more efficiently. The computing time is probably the sum of a constant term A(namely opening the file, defining the branches, ...) and a term B which scale with the number of events N(namely unpacking the data, writing in the file). The resulting rate is $N/(A + B * N)$. After a certain file size, i.e. after a certain N, the rate reaches a plateau, saturating its value at $1/N$. However, there is a limitation to increasing the input file size since the ntuple must be stored in the RAM disk of the computer, which has limited space. Additionally, if a buffer is required, its storage space should not be saturated. This limitation does not hold for the other data format, as the rate remains constant for different files.

Another crucial result is the working rate of the algorithm. The best working rate per event, achieved using the largest file and the TTree format, is approximately 586.9kHz—only one to two orders of magnitude lower than the LHC's 40MHz. Specifically, it is 68.2 times lower.

Lastly, another noteworthy result is the size of the output file. In all tested cases, both formats report an output size approximately 3.6 times larger than the input size. This increase comes from the fact that the variables that were saved with few bits (14 for the transverse momenta case, or 12 for η), in the ntuple are saved as floating point with 32 bit and that the non-existing variables for charged or neutrals particles are added as zeros. Nonetheless, the input-output size ratio appears to remain constant with changes in dimension.

3.2.1.1 Multi-threading and compression results

An intriguing aspect to investigate is the behavior of the algorithm when run with multi-threading enabled. Multi-threading is the capability of a central processing unit (CPU) to support multiple threads of execution concurrently, facilitated by the operating system. In a multithreaded application, threads share the resources of one or more cores. The CPU used for these tests is a 16-core hyper-threaded processor, featuring 16 physical cores and 32 logical ones. Performance is assessed by varying the number of threads (jobs) from 0 to 8, where the difference between 0 and 1 means respectively, single core not multi-thread and single core with multi-threaded allowed. The number of threads is not increased further, as indicated in the table, since performance already exhibit the trends when using a larger number of threads.

Simultaneously with the multi-thread analysis, it is interesting to explore performance when the output file is compressed. This consideration is particularly relevant when storage of the ntuple is required. While the time required for computation may increase, given the additional compression task for the CPU, it is valuable to understand how this aspect evolves with an increasing number of threads. The compression algorithm used is lz4, and it is tested for different compression levels.

The results for the two formats are presented separately in Table 3.7 for the TTree and in Table 3.8 for the RNTuple. The outcomes are based on running the algorithm on a file containing about 2000 orbits, with a size of 285.54 MB. This choice is justified as it represents a candidate file size of the live processing and being a large file, it can highlight specific trends. Both tables provide information on timing performance and working rate per event, similar to the previous section. It is crucial to include the rate as it offers additional and more precise insights when the computing time experiences slight fluctuations. Additionally, reporting the different output sizes for varying compression levels is essential.

It is important to reiterate that these measurements offer indicative results, and the reported values represent average behavior. In particular, the performance here exhibits slight oscillations around an average value, which is documented in the tables.

Comp Level	**0**			**1**		
Jobs	**time (s)**	**rate (kHz)**	**CPUs used**	**time (s)**	**rate (kHz)**	**CPUs used**
0	2.01	593.0	0.828	11.79	101.1	0.995
1	2.03	586.8	1.270	7.51	158.8	1.931
2	2.03	586.4	1.280	5.95	200.3	2.400
4	2.10	564.8	1.277	5.41	220.4	2.676
8	2.09	570.0	1.279	5.01	237.8	2.911
Size	1022			384		

Comp Level	**2**			**5**		
Jobs	**time (s)**	**rate (kHz)**	**CPUs used**	**time (s)**	**rate (kHz)**	**CPUs used**
0	13.50	88.3	0.996	32.35	36.8	0.998
1	8.70	136.9	1.892	21.03	56.7	1.748
2	6.90	172.6	2.366	15.20	78.4	2.360
4	6.39	186.4	2.604	13.45	88.6	2.712
8	5.75	207.4	2.918	11.55	103.2	3.156
Size	371			356		

Table 3.7: *Table reporting the final results for a multi-threading approach for unpacking the raw data and producing a ntuple in a `TTree` format. For each number of jobs, in the first column, the performance results are reported. Moreover, the compression of the output file is tested allowing multi-threading. The results are shown for different level of compression. For each combination, it is also reported the number of CPUs utilized, which is similar to the percentage of CPU used for the execution of the particular combination. It helps to show the CPU performances when enabling more threads*

TTREE In the uncompressed scenario, the augmentation of threads doesn't lead to a discernible improvement in execution time; the performance remains relatively constant. This phenomenon is likely attributed to the absence of compression, leaving little for the additional threads to contribute.

On the other hand, in scenarios involving compression, even the utilization of a single thread results in a significant boost in performance. For example, employing only one thread at compression level 1 leads to a twofold reduction in computing time, diminishing from 11.79 seconds to 7.51 seconds. This improvement persists with the addition of more threads, albeit more gradually. The compression rate increases, and the execution time decreases in tandem with the growing number of threads. This consistent behavior holds true across all compression levels. With one thread, the CPU likely needs to create some additional data structures to synchronize threads (even though practically only one thread is used), and this may be sufficient to expedite the execution.

However, there is an increase in time when raising the compression level, attributed to the fact that compression is a computationally demanding task. Moreover, as the compression level increases, the time increases significantly, even for a slight reduction in the output file size. This behavior becomes particularly noteworthy when transitioning from compression level 2 to 5. For example, with a single job, the time required to compress at level 5 is approximately twice that of compressing at level 1, while gaining only slightly over one megabyte of space, equivalent to about 5% of the value. This implies that if it is necessary to store a compressed file to save memory space, it is crucial to increase the number of threads, even by just one. However, working with a compressed file is not efficient, as even with multi-threading, the execution time is still twice that without compression. Ultimately, it is preferable to work directly with 0 threads, as there is no performance gain with multi-threading.

RNTuple In the `RNTuple` scenario, the situation diverges slightly from the previous one. As noted earlier, `RNTuples` generally demonstrate slightly lower performance compared to `TTrees`. However, for uncompressed files, performance remains relatively stable with an increasing number of threads, with only a marginal degradation by a few hundredths of a second. Despite this, the performance appears quite comparable when the output file is compressed. Even with an elevated compression level, the overall performance appears to remain nearly constant, although there is a slight decline for higher compression levels.

In the context of compression, the results do not showcase a distinct improvement between scenarios with and without multi-threading, as previously observed. Indeed, on the whole, there seems to be a slight decline when multiple threads are utilized.

Nonetheless, even in this scenario, optimal performance is attained when working without multi-threading and without compressing the output file.

3.3 Selection

Lastly, after the data collected from the DTH are unpacked in the PC, the final step of the chain takes place: the real online selection. Indeed, all the tasks described and developed until now lead to what follows here. A certain number of orbits generated by the correlator emulator are collected in the PC, passing through the DTH. Subsequently, in the PC, they are unpacked in their entirety and stored in the RAM disk. Following this, they are analyzed to select the events containing a $W \to 3\pi$ candidate signal. This entails searching for a candidate triplet of pions in each event. The selection algorithm is the same as described in Section 2.2 with the same filter parameters. Various criteria are applied to the pions contained in each event, necessitating specific features or characteristics in the pions or in the potential triplets.

Comp Level	**0**			**1**		
Jobs	**time (s)**	**rate (kHz)**	**CPUs used**	**time (s)**	**rate (kHz)**	**CPUs used**
0	10.77	110.0	1.000	11.01	107.9	1.000
1	11.08	107.1	2.152	11.20	106.2	2.358
2	11.22	106.0	2.182	11.41	104.2	2.223
4	11.30	105.5	2.188	11.45	103.8	2.271
8	11.24	105.8	2.173	11.36	104.6	2.274
Size	983			622		
Comp Level	**2**			**5**		
Jobs	**time (s)**	**rate (kHz)**	**CPUs used**	**time (s)**	**rate (kHz)**	**CPUs used**
0	12.02	98.8	1.000	11.10	107.1	1.000
1	11.27	105.5	2.276	11.96	99.4	3.317
2	11.13	106.8	2.310	11.66	101.9	3.351
4	10.96	108.4	2.321	11.81	100.6	3.346
8	11.71	105.1	2.253	11.60	102.5	3.367
Size	621			475		

Table 3.8: *Table reporting the final results for a multi-threading approach for unpacking the raw data and producing a ntuple in a `RNTuple` format. For each number of jobs, in the first column, the performance results are reported. Moreover, the compression of the output file is tested allowing multi-threading. The results are shown for different level of compression. For each combination, it is also reported the number of CPUs utilized, which is similar to the percentage of CPU used for the execution of the particular combination. It helps to show the CPU performances when enabling more threads*

3.3.1 Selection optimization

The performance of the selection algorithm must contend with the rate of LHC bunch crossing, necessitating efforts to enhance its timing capabilities. The characterization of the algorithm, as detailed in the second chapter, is crucial for this purpose. This study revealed the significance of each filter and its impact on the selection, particularly concerning the transverse momenta and isolation filters. In Section 2.2, an intriguing result was highlighted during the discussion of signal efficiency: the number of pre-selected events, defined as the events with at least three pions passing the transverse momentum filter, indicated a drastic reduction in the number of candidate events by applying only this filter. To expedite computational time, the transverse momentum filter is initially applied to all collected events, and only the passing events are subsequently analyzed by applying the other filters.

Another vital observation regards the computation of isolation. Computing isolation for a single particle is computationally intensive, requiring additional calculations. It is imperative to evaluate this parameter only for the pions that truly need it, avoiding redundant computations for the same pion. Therefore, when constructing the different possible triplets, special attention must be paid to avoiding unnecessary calculations and enhancing overall performance.

Considerations such as these can significantly enhance the computational performance of the selection process. From a hardware perspective, as mentioned earlier, the role of the RAM disk is crucial: the files of the ntuple containing the unpacked events are stored in the RAM disk of the computer, and the selection algorithm utilizes the same strategy to expedite reading and writing tasks. The trade-off is that the utilization of the available space, approximately 12GB, must accommodate both the selection and unpacking algorithms, both of which write to the same disk.

3.3.2 Selection timing performance

A fundamental question to address is what information is worth preserving about the selected events, namely, what the output file of the selection algorithm will contain. Given that this step represents the final stage of the online selection, it is crucial to determine what to retain for subsequent offline and in-depth analyses. Two types of output are explored in this context. Firstly, the algorithm can save the entire selected event, including the particles with all their features, in the same format as the input ntuple. Secondly, the selection can choose to save only the histogram of the invariant mass produced by the invariant mass algorithm. As described in the previous chapter, the algorithm computes the histogram of the invariant mass of the selected triplets. Lastly, an alternative is to save only the tracks of the candidate triplets, i.e., the features of the pions, specifically their $\mathrm{p_T}, \eta, \phi$.

The first scenario involves saving a large-sized file containing all the information of the event. While this allows for a more comprehensive offline analysis and potential exploration for various studies, it comes with the trade-off of requiring more time and storage space. The other two scenarios, saving only the histogram or the tracks of the candidate triplets, imply smaller file sizes with a focus on retaining key information related to the specific decay.

In this context, only the first two scenarios are tested, i.e., saving the entire event or the histogram, to parameterize the two different situations. Specifically, the presented time information pertains to reading the ntuple saved in the ram disk by the unpacker, performing the selection, and saving one of the two types of files. Additionally, the dimensions of the saved output are recorded and can be compared to the input file size. This information is crucial for providing an order of magnitude estimation of the required storage space.

Concerning the output file size, it is crucial to emphasize its strong correlation with the signal efficiency, as defined in equation (2.9). This efficiency, denoted as ε, significantly relies on how the events are generated, specifically on the generated features of the puppi candidates. As explained in section 3.1.4, the particle's features are populated by assigning them the generated numbers under the assumption of their non-correlation or independence. Consequently, the puppi candidates produced by the emulator deviate from those in the reference dataset, resulting in a non-realistic representation of the generated signal. Due to this assumption, the signal efficiency will not be equal to the one reported in the second chapter.

Nonetheless, from the tests performed the average value of the efficiency is about 0.5% which is very near the value of the efficiency obtained by combining the signal (2.4) and the background (2.18) efficiencies estimated in the previous chapter:

$$\varepsilon_{tot} = \frac{\mathrm{N}_{\text{selected}}}{\mathrm{N}_{tot}} = 0.2\%. \tag{3.3}$$

This indicates that the data size results differs slightly from the reality.

However, the results pertain to running the selection algorithm on the ntuple in the RAM disk and saving either a histogram of the invariant mass in the W mass range (from 60 to 100 GeV) or the entire selected events. Specifically, the tests are conducted on reading from both the `TTree` and the `RNTuple`. However, it's important to note that when the algorithm saves the events, it does so by utilizing a `TTree`, even if the input is originally in the `RNTuple` format. The selected events are saved through a snapshot of the dataframe, i.e., retaining only the selected portion of the ntuple. For each combination, such as reading a `TTree` and saving a histogram, the input size and output size are reported, along with the computational time required and the rate per event, similar to the previous case regarding the unpacking performances. As before, this rate needs to be compared with the bunch crossing rate. Moreover, the tests are performed on two different files, namely the two largest files also used in the previous section. One contains $59,994$ events, while the other one contains $1,188,595$. Clearly, the selection efficiencies are the same in both cases since the selection algorithm is consistent. For the smaller file, the selected events are $N_{\text{sel}} = 320$, resulting in an efficiency of $\varepsilon = 0.51\%$, while for the larger file, $N_{\text{sel}} = 6095$, resulting in $\varepsilon = 0.53\%$.

Snapshot				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	51.67	0.668	89.1	138.0
rntuple	51.96	0.626	95.9	152.2

Histogram				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	51.67	0.452	132.8	12.0
rntuple	51.96	0.432	138.9	12.0

Table 3.9: *Table reporting the final results for the selection performance when running the selection algorithm on the file containing* 59994 *events. The selected number of events is* $N = 320$. *Both input format are tested.*

The results are presented in two tables, one for the smaller file (Table 3.9) and the other for the larger file (Table 3.10).

In both cases, a similar trend is observed; the two formats yield comparable results in terms of computing time, with the `TTree` appearing to perform slightly better for bigger files. Additionally, all scenarios exhibit scalability with the number of events, as evidenced by the increasing rate with the number of events. As anticipated, saving the histogram demands less time and requires less storage space than saving the entire event. However, in both cases, the file size undergoes a significant reduction. The reduction ratio is highly dependent on the selection filters, which regulate the number of passing events. As indicative results, the outcomes of running the algorithm with loose filters are reported in Appendix B, specifically in Table 4.3a and 4.3b. Here, the reported results correspond to the filters described in Section 2.2.

Snapshot				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(MB)**
tree	1021.87	7.513	158.2	2.4
rntuple	1029.76	7.732	153.0	2.7

Histogram				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	1021.87	4.271	278.3	12.0
rntuple	1029.76	4.472	265.8	12.0

Table 3.10: *Table reporting the final results for the selection performance when running the selection algorithm on the file containing* $1,188,595$ *events. The selected number of events is* $N = 6095$. *Both input format are tested.*

Chapter 4

Live analysis and Results

The primary goal of this project is to demonstrate the feasibility of an online selection, specifically targeting a rare decay of the W boson into three pions, an event that has not yet been observed. The preceding chapter provided a comprehensive overview of the diverse facets of this endeavor, introducing the concept of constructing a three-step chain, each dedicated to a specific processing task. Initially, the L1 correlator trigger board emulator generates puppi candidate events, transmitting these packets to the PC via a DTH. Subsequently, the PC accumulates a set number of events, unpacks them, and produces ntuples stored in a RAM disk. Finally, the ntuples undergo analysis to identify events containing candidate triplets of pions.

This chapter consolidates the varied outcomes to offer an inclusive and conclusive depiction of the entire online selection process. The discussion culminates with an exploration of the performance tests conducted on the realistic online selection, elucidating the synergy of the distinct steps. It evaluates whether the processing chain can effectively contend with the operational rate of the LHC or if adjustments to the data flow are necessary to ensure sufficient time for data processing.

4.1 Setup for online selection

In the context of the unpacking and selection steps, the previous chapter carefully considered various strategies, including the choice of ntuple format and decisions on the number of events to process and the content to save. This section provides a detailed description of these choices.

To begin, it is crucial to address the question of how many events or orbits to collect and process in each iteration. The optimal value should strike a balance between the timing performance of the processes and the limited capacity of the RAM disk. As discussed in section 3.2.1, the unpacking performance scales with the number of events (up to a certain limit), suggesting that higher numbers of events result in higher rates. A reasonable estimate may involve collecting the number of orbits such that the size of the unpacked ntuple is approximately 1 GB. The rough estimate is given by

$$\mathrm{N_{orbits}} = \frac{1\mathrm{GB}}{3.6 \cdot dim_{ev} \cdot 3564/6} \simeq 2000 \tag{4.1}$$

where 3.6 represents the ratio between the unpacked ntuple and the raw data file, 3564 is the number of events in each orbit (divided by 6 due to the trigger board's output every 6 bunch crossings), and dim(1 event) is the dimension of a single event in bits. This dimension is estimated as the average event multiplicity value multiplied by the size of a particle (64 bits). The performances of the unpacking and selection process were already documented for a file containing this number of orbits.

Another important characteristic to discuss is the ntuple format to employ. As highlighted in Section 3.2.1, optimal performances are achieved with the `TTree` format and a single core without multi-threading. Specifically, this combination operates at a rate of approximately 600 kHz. Such performance is competitive with the LHC working rate, even though a buffer is necessary. It is essential to note that the process runs on a single CPU, so the performance could be significantly improved by distributing computing tasks across multiple processors. While the unpacking performances with multi-threads appear nearly identical, multi-threading is chosen as it can be beneficial when dealing with multiple files simultaneously, aiding the system in achieving better overall performance.

Lastly, compression is not considered in the process, as it is not strictly necessary, and its performances are relatively slow. Additionally, performing the selection on a compressed file might incur additional time for decompressing the data. However, compression can prove useful before storing the selected events to conserve disk space.

Finally, the last important aspect to discuss pertains to the choice of what to save after the selection. The performances of saving only the histogram or saving the entire event are detailed in Section 3.3.2 (see Table 3.10).

Despite the faster speed and reduced space requirements of saving only the invariant mass histogram, opting to save the entire selected event proves to be a more advantageous solution. Although the timing performances differ significantly (factor two), saving the entire event allows for a more versatile outcome. Specifically, the selection process with event snapshotting operates at a rate of approximately 160 kHz, while saving the histogram achieves a rate of 280 kHz. Nevertheless, saving the entire event remains the preferred option as the resulting dataset can be valuable for various studies or analyses, making it the preferred solution.

While saving only the invariant mass histogram demonstrates faster speed and reduced space requirements, choosing to store the complete selected event emerges as a more beneficial solution. Specifically, the event snapshotting selection process operates at a rate of approximately 160 kHz, whereas saving the histogram achieves a rate of 280 kHz. Despite a significant difference in timing performances (a factor of two), opting for the entire event storage offers greater versatility and potential value of the resulting dataset for various studies or analyses.

Storaging space required This decision leads to the estimation of the storage space required over a specific period of data acquisition. In this thesis, all estimations, such as the following, are based on the assumption of 400 fb^{-1} of integrated luminosity, corresponding to approximately a year of Data Acquisition in the full HL-LHC regime. Within this timeframe, according to the results in Table 3.10, the required space to save the ntuple of the selected events is approximately 432 TB. This calculation considers the rough estimate of the saved file size corresponding to 2000 orbits and accounts for the fact that we are dealing with 1/6 of the realistic flux. It is then scaled to the 400 fb^{-1} integrated luminosity. It is crucial to emphasize that these estimates are based on a selection efficiency of $\varepsilon = 0.5\%$, which, as outlined in Section 3.3.2, is a combination of the two efficiencies (signal and background) evaluated in the second chapter. Furthermore, it is essential to note that this is an order of magnitude estimate.

The same figure can be computed by considering the average size of an event, assuming an average multiplicity of 30 particles. Taking into account the 3.6 factor of the unpacking process and an efficiency of 0.5%, a similar result is obtained.

4.2 Running the online selection

The final phase of the project involves integrating all the individual tasks into a cohesive system. Up to this point, each test has been analyzed individually. However, the actual implementation of the online selection aims to run all tasks concurrently. The approach involves continuously collecting around 2000 orbits in the RAM disk every 150 ns. As soon as a raw file is present, it is unpacked, and the `TTree` is saved. The same concept applies to the selection step: as soon as an ntuple is available, it is analyzed, and the selected events are stored. It's crucial to note that the maximum space available in the RAM disk in this demonstration setup, is approximately 13 GB, while during Phase-2 this space is planned to be larger, up to few TB.

As demonstrated earlier, the two processes operate at different rates, implying that the files to be analyzed accumulate in the RAM disk, awaiting processing by the two distinct tasks. Therefore, it becomes imperative to prevent the RAM disk from overflowing, meaning its usage must be monitored to prevent a system crash and potential loss of events.

The performance results clearly indicate that the real data flow needs to be scaled. This is due to the inability to match the real LHC bunch crossing rate, as a single CPU is utilized for this initial study, and the tests reveal a working rate lower than the 40 MHz. It's essential to remember that the data flow is already scaled by 6 by the trigger board. The system is set to work online but at a scaled rate, which is expected given that the processor used in this study is not representative of the one that will be employed for HL-LHC.

However, an important study to conduct involves identifying the minimum scaling factor that the processing chain can withstand without crashing. The critical factor under consideration is the *prescale*, which represents the number of orbits that are skipped before saving them. For instance, a prescale of 10 implies that one orbit is saved for every 10. The prescale is systematically reduced until the system can no longer tolerate the incoming data, which is evident when the RAM disk usage approaches full capacity over a certain period.

The final tests are executed as follows: the system is configured as detailed in the previous section and allowed to operate according to the described procedure. Subsequently, the trigger board is activated with a specified prescale value. The system is observed for a few minutes, with continuous monitoring of the RAM disk and CPU usage. These parameters provide insights into how effectively the system handles the incoming data under varying prescale conditions.

4.2.1 Final Results

The ultimate results are presented in terms of unpacking performance in Table 4.1 and selection in Table 4.2. The former pertains to the live unpacking process with multi-threading enabled, employing 4 jobs. Notably, the table displays CPU usage values exceeding 100% due to the multi-threading process. The latter table concerns the selection algorithm. Both tables report identical RAM disk occupancy values since they share the same space.

Primarily, the RAM occupancy exhibits slight oscillations in each configuration, as expected. This behavior arises from the distinct working rates of the unpacking and selection processes, causing fluctuations in the utilized space depending on how the files are processed. The table reports the mean value of the oscillating range, offering insight into the system's average behavior. It's noteworthy that the width of the oscillating range

Prescale	CPU %	RAM Disk (MB)	Data Rate In (GB/s)	Data Rate Out (GB/s)
10	180	50	0.2	0.5
8	180	90	0.2	0.7
6	220	140	0.3	0.9
4	300	340	0.4	1.4
2	**650**	**1,400**	**0.8**	**2.8**

Table 4.1: *unpacking live performances*

Prescale	CPU %	RAM Disk (MB)	Data Rate In (GB/s)	Data Rate Out (GB/s)
10	90	50	0.1	0.003
8	90	90	0.2	0.004
6	90	140	0.1	0.003
4	95	340	0.1	0.003
2	**100**	**1,400**	**0.1**	**0.003**

Table 4.2: *selection live performances*

increases with the decreasing prescale value. Additionally, it's important to highlight that, for each configuration, the system runs continuously for approximately 1 minute, providing a snapshot of its performance over a specific time interval.

However, the table exclusively presents performances with a prescale value of 2, as the system crashes when further reducing it to 1. Within a few seconds of operation, the RAM occupancy fills up, indicating that the unpacking process cannot keep up with the incoming flux, preventing the timely unpacking of all saved files. Nonetheless, the key takeaway is as follows: with a prescale of 2, saving one orbit every two, the system effectively handles all incoming events from the DTH without causing a drastic increase in RAM occupancy, which does not exceed a maximum of 2 GB, well within the available space of 13 GB.

Conclusions

The High-Luminosity LHC (HL-LHC) marks a new era in particle physics, promising unparalleled opportunities for groundbreaking discoveries. With its remarkable features, including increased luminosity and an anticipated average of 140 collisions per bunch crossing, the HL-LHC will delve into uncharted territories of physics. However, this progress comes at the cost of managing an even larger volume of data and coping with substantial contributions to pileup. The CMS detector, a crucial component in this scientific endeavor, will undergo significant upgrades, with a particular focus on enhancing the Level 1 trigger to expand its menu of capabilities.

This unprecedented upgrade underscores the pivotal role of the Level 1 scouting system. Unlike traditional methods, data scouting enables the extraction of information directly from various stages of the trigger chain as it is actively collected. The upgraded Level 1 trigger, coupled with the unique capabilities of the scouting system, will play a crucial role in navigating the challenges posed by the enhanced conditions of the HL-LHC era, ultimately contributing to the success of the experiments and the potential discovery of new physics. These challenges include the increased data volume and pileup, demanding novel strategies for efficient data extraction and event selection.

A particularly intriguing development among the numerous upgrades that CMS will undergo in preparation for Phase 2 (HL-LHC), is the ability to run reconstruction algorithms such as Particle Flow and PUPPI directly at Level 1. This paves the way for innovative strategies in selecting interesting events by leveraging the data provided by the L1 scouting system.

A pioneering effort has yielded the creation of a prototype for an online selection, specifically tailored for the rare decay of the W boson into three charged pions. This project aimed to showcase the viability of employing such an event selection technique using data from the L1 scouting system, specifically the data coming from the correlator layer 2.

Initiating the project, Monte Carlo (MC) simulations for phase 2 PUPPI candidates facilitated a preliminary physics analysis of the unique decay process. This analysis served as an exploratory study designed to emulate the kind of dataset anticipated to be accessible at the L1 correlator layer during phase 2. This initial investigation yielded crucial estimates, including the projected number of expected signal events over a year of data acquisition (equivalent to an integrated luminosity of 400 fb^{-1}):

$$N_{events} \simeq 570. \tag{4.2}$$

Additionally, a significant estimate was derived for the signal-to-noise ratio:

$$\mathrm{SNR} = 1.56 \times 10^{-5} \tag{4.3}$$

Further contributing to this endeavor, an upper limit was imposed on the branching fraction of the rare W boson decay, restricting it to $\mathcal{B}(\mathrm{W} \to 3\pi) < 1.04 \times 10^{-6}$. Ultimately, this introductory analysis played a pivotal role in delineating a selection algorithm for the online selection process, undestranding its performance across various PUPPI candidate features.

Subsequently, each of the three essential steps required for the demonstrator is meticulously examined in isolation to delineate their characteristics and identify optimal conditions for enhancing performance in an online approach. Finally, their interconnections are explored, providing insights into the hardware system where the processing chain operates and assessing how they align with the LHC working rate.

The results of the online selection performance are promising, demonstrating effective functionality with a prescale set to 2, indicating that one orbit every two is analyzed, thereby reducing the LHC rate by a factor of 2. Moreover, the estimated storage space required is approximately 500 TB for a year of data acquisition, which stands as a reasonable and feasible demand.

It's crucial to note that the performance results are obtained by testing the processing chain on an incoming flux already scaled by 1/6, owing to the limited output connection from the trigger board. Therefore, the prescale on the orbits serves as an additional scale on the LHC online working rate.

Despite these considerations, the results are promising, especially given that the computing system used for the demonstrator is not a realistic representation of the one that will be employed by the Phase-2 scouting system. This initial approach to the selection technique suggests ample room for improvement through further in-depth study. This applies not only to the methodology but also to the analysis performed on the data provided by the online selection. Finally, it is noteworthy that the effective implementation of online selection is slated to be introduced in the Phase-2 High-Luminosity era, expected around 2030. The significant advancements in technology over recent years foster optimism for the deployment of highly efficient processing units capable of analyzing more data with increased efficiency.

Appendix A

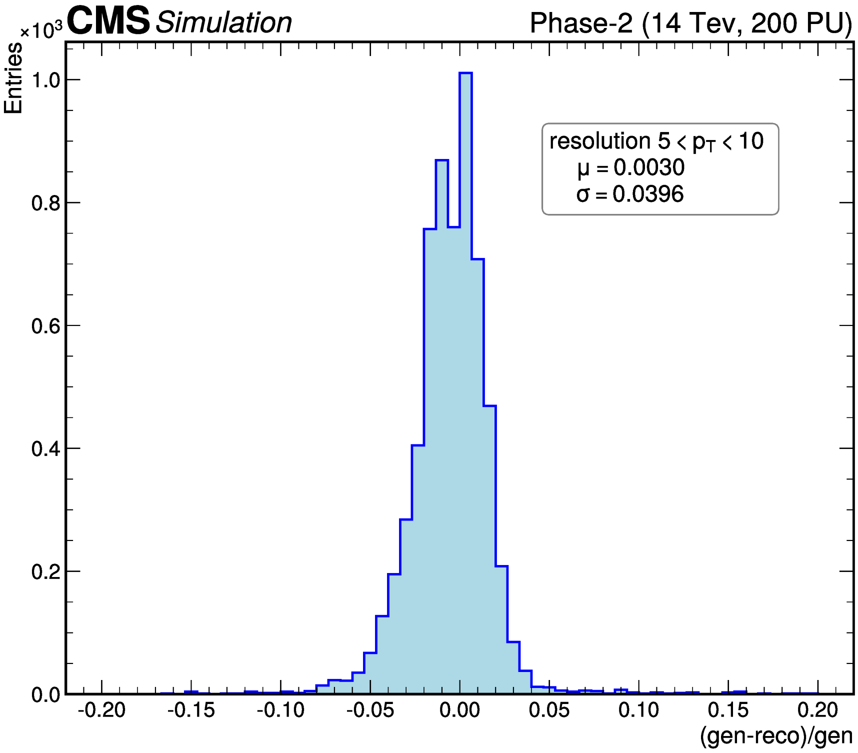
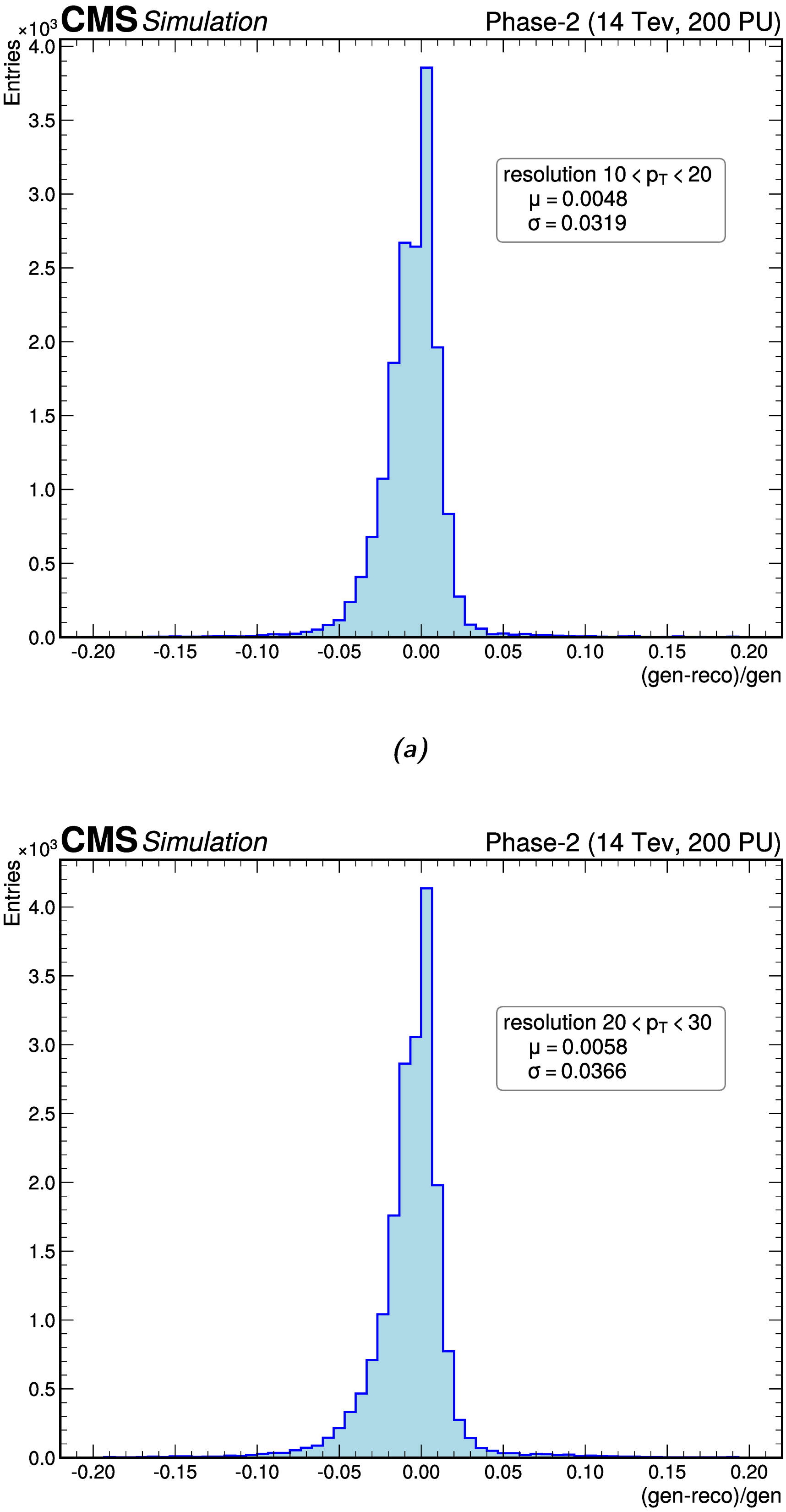


Figure 4.1: *Histogram illustrating the distribution defined in Equation* (2.7) *for the transverse momenta of the pions. Specifically, the distribution of the pions having the transverse momenta in the range* $[5, 10]\,GeV$

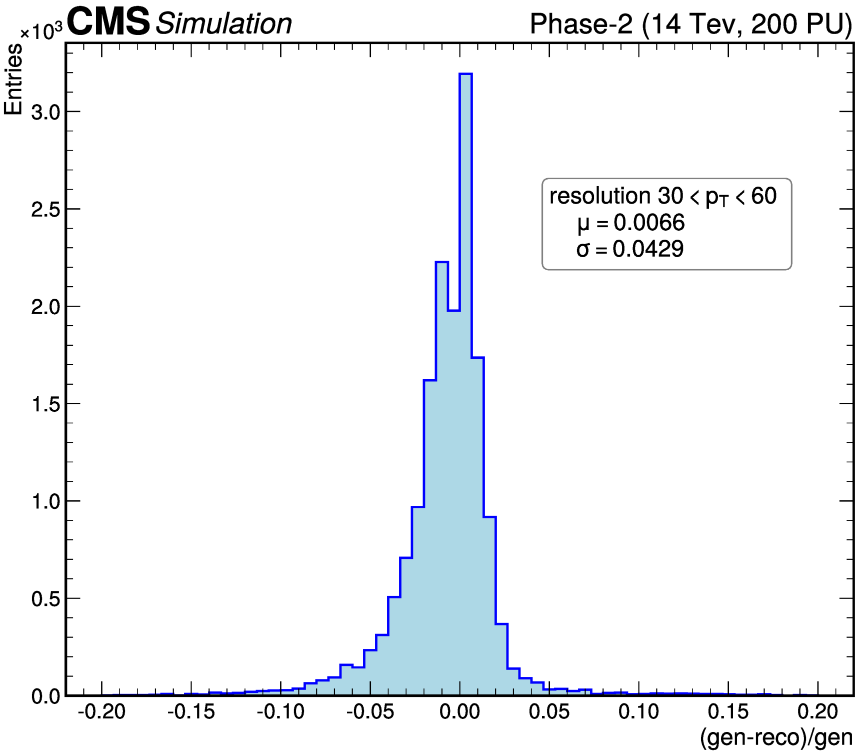
.



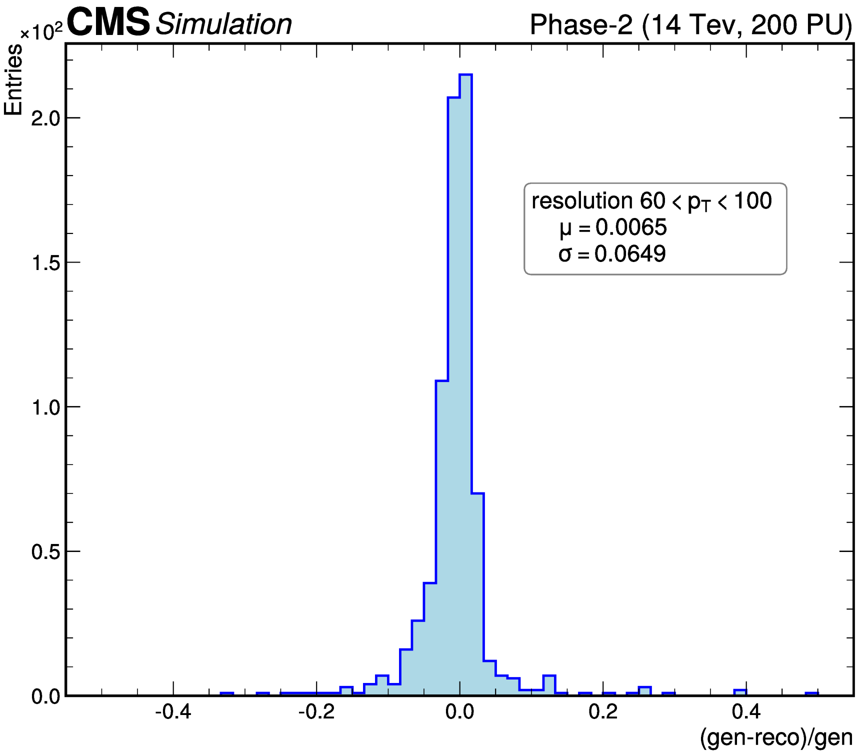
(a)

(b)

Figure 4.2: *Histograms illustrating the distribution defined in Equation (2.7) for the transverse momenta of the pions. Specifically, Figure (a) shows the distribution of the pions having the transverse momenta in the range* $[10, 20] GeV$*, while Figure (b) the pions having the transverse momenta in the range* $[20, 30] GeV$



(a)



(b)

Figure 4.3: *Histograms illustrating the distribution defined in Equation (2.7) for the transverse momenta of the pions. Specifically, Figure (a) shows the distribution of the pions having the transverse momenta in the range* $[30, 60]\,GeV$*, while Figure (b) the pions having the transverse momenta in the range* $[60, 100]\,GeV$

Appendix B

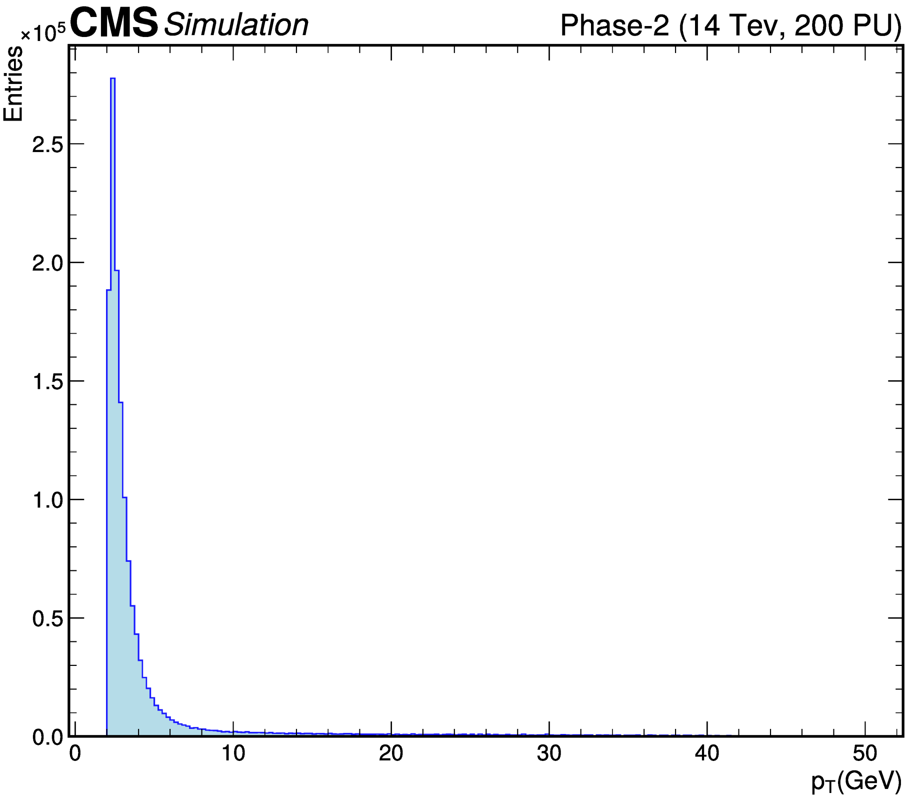


Figure 4.4: *Histogram illustrating the distribution of the transverse momenta contained in the simulation plotted in the range from 2 to 50 GeV.*

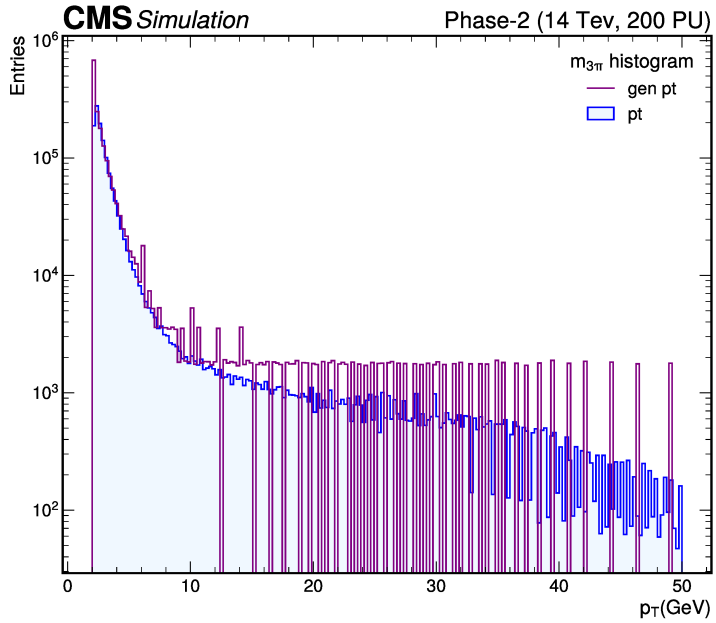


Figure 4.5: *Histogram illustrating the distribution of the generated transverse momenta and the one contained in the simulation in a logarithmic scale, plotted in the range from 2 to 50 GeV. The comparison between the generated and the simulated p_T demonstrate the limitation of the generation technique choose for this features. There are a maximum of 1024 possible different values that can be generated. The first part of the distribution is well replicated while the other one shows some limitations.*

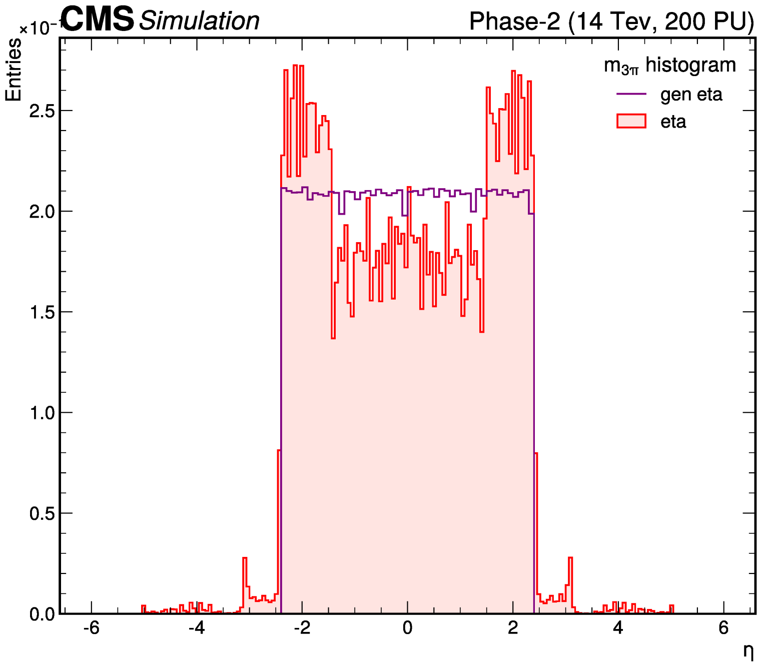


Figure 4.6: *The histogram illustrates the comparison between the pseudorapidity distribution of the particles in the simulation and the generated ones.*

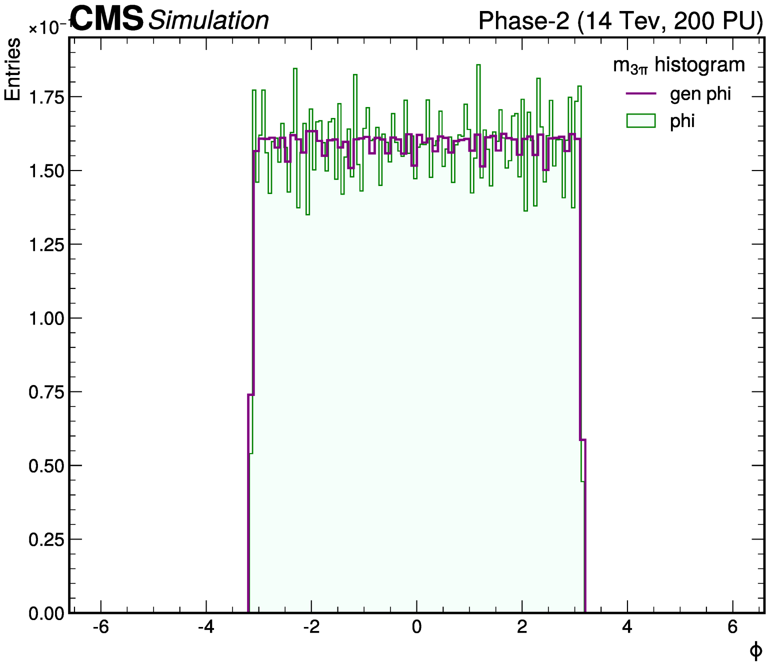


Figure 4.7: The histogram illustrates the comparison between the azimutal angle distribution of the particles in the simulation and the generated ones. Both distributions are uniform in the range between $[-\pi, +\pi]$

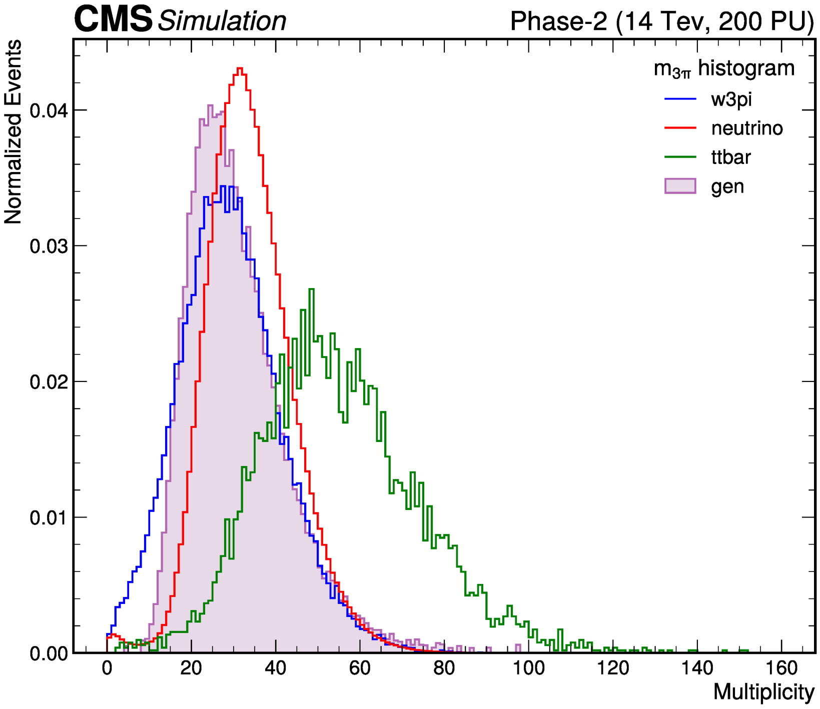


Figure 4.8: The histogram illustrates the comparison between the normalized distribution of the event multiplicity in the three available datasets and the generated one. The latter represents a trade-off between the one of the signal dataset and the "neutrinogun" background dataset.

Snapshot				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	51.67	0.733	81.1	543.7
rntuple	51.96	0.758	79.2	601.2

Histogram				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	51.67	0.473	126.8	12.0
rntuple	51.96	0.492	122.0	12.0

(a) *Table reporting the final results for the selection performance when running the selection algorithm on the file containing* 59994 *events. The selected number of events is* $N = 320$. *Both input format are tested.*

Snapshot				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(MB)**
tree	1021.87	9.503	125.1	9.80
rntuple	1029.76	9.763	121.7	11.0

Histogram				
	dim in (MB)	**time(s)**	**rate(kHz)**	**dim out(kB)**
tree	1021.87	5.251	226.4	12.0
rntuple	1029.76	5.378	221.0	12.0

(b) *Table reporting the final results for the selection performance when running the selection algorithm on the file containing* $1,188,595$ *events. The selected number of events is* $N = 6095$. *Both input format are tested.*

Table 4.3: *The two tables report the outcomes of running the algorithm with loose filters*

Bibliography

[1] Oliver Sim Brüning et al. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-1.

[2] Oliver Sim Brüning et al. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-2.

[3] Lyndon Evans. "The Large Hadron Collider". In: *New Journal of Physics* 9.9 (2007), p. 335. DOI: 10.1088/1367-2630/9/9/335.

[4] Michael Benedikt et al. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-3.

[5] The CMS Collaboration. In: 3.08 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004. URL: https://dx.doi.org/10.1088/1748-0221/3/08/S08004.

[6] The ATLAS Collaboration. In: 3.08 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003. URL: https://dx.doi.org/10.1088/1748-0221/3/08/S08003.

[7] The LHCb Collaboration. In: 3.08 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005. URL: https://dx.doi.org/10.1088/1748-0221/3/08/S08005.

[8] The ALICE Collaboration. In: 3.08 (2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002. URL: https://dx.doi.org/10.1088/1748-0221/3/08/S08002.

[9] The CMS Collaboration. "Measurement of the inelastic proton-proton cross section at $$ \sqrt{s}=13 $$ TeV". In: *Journal of High Energy Physics* 2018.7 (2018). DOI: 10.1007/jhep07(2018)161. URL: https://doi.org/10.1007%2Fjhep07%282018%29161.

[10] Werner Herr and B Muratori. "Concept of luminosity". In: (2006). DOI: 10.5170/CERN-2006-002.361. URL: https://cds.cern.ch/record/941318.

[11] *Public CMS Luminosity Information*. https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults.

[12] Tai Sakuma and Thomas McCauley. "Detector and Event Visualization with Sketch Up at the CMS Experiment". In: *Journal of Physics: Conference Series* 513.2 (2014), p. 022032. DOI: 10.1088/1742-6596/513/2/022032. URL: https://dx.doi.org/10.1088/1742-6596/513/2/022032.

[13] V Karimäki et al. *The CMS tracker system project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/368412.

[14] *The CMS electromagnetic calorimeter project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/349375.

[15] *The CMS hadron calorimeter project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/357153.

[16] *The CMS magnet project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. DOI: 10.17181/CERN.6ZU0.V4T9. URL: https://cds.cern.ch/record/331056.

[17] SMS Collaboration et al. "Commissioning of CMS HLT". In: *Journal of Instrumentation* 5 (2010), T03005. DOI: 10.1088/1748-0221/5/03/T03005.

[18] G L Bayatyan et al. *CMS TriDAS project: Technical Design Report, Volume 1: The Trigger Systems*. Technical design report. CMS. URL: https://cds.cern.ch/record/706847.

[19] Sergio Cittolin, Attila Rácz, and Paris Sphicas. *CMS The TriDAS Project: Technical Design Report, Volume 2: Data Acquisition and High-Level Trigger. CMS trigger and data-acquisition project*. Technical design report. CMS. Geneva: CERN, 2002. URL: https://cds.cern.ch/record/578006.

[20] CMS Collaboration. "Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV". In: (2020).

[21] *Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET*. Tech. rep. Geneva: CERN, 2009. URL: https://cds.cern.ch/record/1194487.

[22] CMS Collaboration. "Particle-flow reconstruction and global event description with the CMS detector. Particle-flow reconstruction and global event description with the CMS detector". In: *JINST* 12.10 (2017), P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965. URL: https://cds.cern.ch/record/2270046.

[23] Benjamin Kreis. *Particle Flow and PUPPI in the Level-1 Trigger at CMS for the HL-LHC*. 2018. arXiv: 1808.02094 [physics.ins-det].

[24] Daniele Bertolini et al. "Pileup per particle identification". In: *Journal of High Energy Physics* 2014.10 (2014). DOI: 10.1007/jhep10(2014)059. URL: https://doi.org/10.1007%2Fjhep10%282014%29059.

[25] A.M. Sirunyan et al. "Pileup mitigation at CMS in 13 TeV data". In: 15.09 (2020), P09018–P09018. DOI: 10.1088/1748-0221/15/09/p09018. URL: https://doi.org/10.1088%2F1748-0221%2F15%2F09%2Fp09018.

[26] O. Aberle et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2020. DOI: 10.23731/CYRM-2020-0010. URL: https://cds.cern.ch/record/2749422.

[27] D Contardo et al. *Technical Proposal for the Phase-II Upgrade of the CMS Detector*. Tech. rep. Geneva, 2015. DOI: 10.17181/CERN.VU8I.D59J. URL: https://cds.cern.ch/record/2020886.

[28] *The Phase-2 Upgrade of the CMS Barrel Calorimeters*. Tech. rep. This is the final version, approved by the LHCC. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2283187.

[29] *The Phase-2 Upgrade of the CMS Endcap Calorimeter*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.IV8M.1JY2. URL: https://cds.cern.ch/record/2293646.

[30] Collaboration CMS. *A MIP Timing Detector for the CMS Phase-2 Upgrade*. Tech. rep. Geneva: CERN, 2019. URL: https://cds.cern.ch/record/2667167.

[31] *The Phase-2 Upgrade of the CMS Muon Detectors*. Tech. rep. This is the final version, approved by the LHCC. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2283189.

[32] *The Phase-2 Upgrade of the CMS Tracker*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.QZ28.FLHW. URL: https://cds.cern.ch/record/2272264.

[33] CMS Collaboration. *The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger*. Tech. rep. This is the final version of the document, approved by the LHCC. Geneva: CERN, 2021. URL: https://cds.cern.ch/record/2759072.

[34] Alexandre Zabi. "System Design and Prototyping for the CMS Level-1 Trigger at the High-Luminosity LHC". In: *Journal of Physics: Conference Series* 2374 (2022), p. 012090. DOI: 10.1088/1742-6596/2374/1/012090.

[35] Dustin Anderson. "Data Scouting in CMS". In: 2017, p. 190. DOI: 10.22323/1.282.0190.

[36] Swagata Mukherjee. "Data Scouting and Data Parking with the CMS High level Trigger". In: *PoS* EPS-HEP2019 (2020), p. 139. DOI: 10.22323/1.364.0139. URL: https://cds.cern.ch/record/2766071.

[37] Gilbert Badaro et al. *40 MHz Level-1 Trigger Scouting for CMS*. Tech. rep. Geneva: CERN, 2020. DOI: 10.1051/epjconf/202024501032. URL: https://cds.cern.ch/record/2798134.

[38] Dinyar Sebastian Rabady. *A 40 MHz Level-1 trigger scouting system for the CMS Phase-2 upgrade*. Tech. rep. Geneva: CERN, 2023. DOI: 10.1016/j.nima.2022.167805. URL: https://cds.cern.ch/record/2816252.

[39] Thomas Owen James. *The Level 1 Scouting system of the CMS experiment*. Tech. rep. Geneva: CERN, 2023. URL: https://cds.cern.ch/record/2852916.

[40] The CMS Collaboration. "Search for W Boson Decays to Three Charged Pions". In: *Physical Review Letters* 122.15 (2019). DOI: 10.1103/physrevlett.122.151802. URL: https://doi.org/10.1103%2Fphysrevlett.122.151802.

[41] The CDF Collaboration. "Search for the rare decay $W^{\pm} \to D_s^{\pm}\gamma$ in $p\overline{p}$ collisions at $\sqrt{s} = 1.8$TeV". In: *Phys. Rev. D* 58 (9 1998), p. 091101. DOI: 10.1103/PhysRevD.58.091101. URL: https://link.aps.org/doi/10.1103/PhysRevD.58.091101.

[42] The CMS Collaboration. "Search for the rare decay of the W boson into a pion and a photon in proton-proton collisions at $\sqrt{s} = 13$ TeV". In: *Phys. Lett. B* 819 (2021). Replaced with the published version. Added the journal reference. All the figures and tables can be found at http://cms-results.web.cern.ch/cms-results/public-results/publications/SMP-20-008 (CMS Public Pages), p. 136409. DOI: 10.1016/j.physletb.2021.136409. arXiv: 2011.06028. URL: https://cds.cern.ch/record/2744169.

[43] A L Read. "Presentation of search results: the CLs technique". In: *Journal of Physics G: Nuclear and Particle Physics* 28.10 (2002), p. 2693. DOI: 10.1088/0954-3899/28/10/313. URL: https://dx.doi.org/10.1088/0954-3899/28/10/313.

[44] Thomas Junk. "Confidence level computation for combining searches with small statistics". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 434.2-3 (1999), pp. 435–443. DOI: 10.1016/s0168-9002(99)00498-2. URL: https://doi.org/10.1016%2Fs0168-9002%2899%2900498-2.

[45] R. L. Workman et al. "Review of Particle Physics". In: *PTEP* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.

[46] Rene Brun and Fons Rademakers. "ROOT — An object oriented data analysis framework". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389 (1997). New Computing Techniques in Physics Research V, pp. 81–86. DOI: https://doi.org/10.1016/S0168-9002(97)00048-X. URL: https://www.sciencedirect.com/science/article/pii/S016890029700048X.

[47] I. Antcheva et al. "ROOT — A C framework for petabyte data storage, statistical analysis and visualization". In: *Computer Physics Communications* 180.12 (2009), pp. 2499–2512. DOI: 10.1016/j.cpc.2009.08.005. URL: https://doi.org/10.1016%2Fj.cpc.2009.08.005.

[48] Root collaboration. *ROOT::RDataFrame Class Reference*. URL: https://root.cern/doc/master/classROOT_1_1RDataFrame.html.

[49] Glen Cowan et al. "Asymptotic formulae for likelihood-based tests of new physics". In: *The European Physical Journal C* (2011). DOI: 10.1140/epjc/s10052-011-1554-0. URL: https://doi.org/10.1140%2Fepjc%2Fs10052-011-1554-0.

[50] A L Read. "Presentation of search results: the CLs technique". In: *Journal of Physics G: Nuclear and Particle Physics* 28.10 (2002), p. 2693. DOI: 10.1088/0954-3899/28/10/313. URL: https://dx.doi.org/10.1088/0954-3899/28/10/313.

[51] Thomas Junk. "Confidence level computation for combining searches with small statistics". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 434.2 (1999), pp. 435–443. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(99)00498-2. URL: https://www.sciencedirect.com/science/article/pii/S0168900299004982.

[52] Serenity Carrier Card. *Serenity for CMS*. URL: https://serenity.web.cern.ch/serenity/.

Acknowledgments

I would like to express my gratitude to all the people I have encountered throughout my academic journey. I consider myself fortunate to have had the opportunity to collaborate with exceptional colleagues and friends who have enriched my physics experience. I do not take their support for granted.

First of all, I want to acknowledge my supervisor, Dr. Giovanni Petrucciani, for guiding me during my time at CERN and throughout the thesis-writing process. His support and care from the beginning have been invaluable. I extend my thanks to Prof. Marco Zanetti for his role as a mentor and to Prof. Jacopo Pazzini for providing me with the incredible opportunity to work at CERN, where I experienced both professional and personal growth.

I am grateful to my colleagues at CERN, including Rocco, Tom, Marco, and my group mates: Nicolò, Alberto, and Giacomo. Our collaborative efforts set ambitious goals, and working alongside them taught me a valuable lesson: training and collaborating with the bests accelerates improvement. Special thanks to Nicolò, who supported me significantly during the thesis period and who answered the millions of questions about everything, my personal chatGPT.

From the University of Padua, I want to thank all my physics classmates, especially my friends Nicole and Federico. Your friendship has been a constant beacon in the fog of Padua. Thanks for the "birrette" with Nicole and for the countless philosophical discussions with Federico. You both have shown me that deep friendships can be built also in a short time.

I extend my appreciation to the people at CERN who welcomed me warmly, showing kindness and support during challenging times, especially the "Zorro FC" group: Mich, Piè, AleGuida, Lorello, il Cholo, Luis, Shini, Fede, Friti. Special thanks to Cristiano (aka Sebba) and Maria Giovanna (aka MG) for treating me like a little brother and providing motivational speeches. I learned a valuable lesson: generosity and spontaneity are contagious, and when people show you kindness, you, too, become more generous with a light heart. Thank you for the insightful discussions about the future that helped me realize that everything is normal.

Subsequently, I thank my classmates from the Bachelor degree. Despite the distances and different paths, we remained in contact, always supporting each other. Milk kings: Bree, Forma, Keivan, Luke, Buz.

Among all the people, I especially would like thank my brothers Nico and Giacomo, for being safe havens, for their spontaneity and serenity. I especially thank Giacomo for the lightheartedness he always tried to pass on to me, reminding me to always put everything into perspective, but also for always supporting me in the many moments of difficulty, we have always supported and strengthened each other, despite the thousand difficulties. I will never forget the thousand days studying together. Best study buddy ever.

I extend a note for my friend Margherita for her support and for always listening to me, in the true sense of the word. But above all for teaching me to listen.

To this list, I would like to thank my first-of-all friend Elena. You always stayed by my side in the innumerous moment of difficulty, keeping it real always. Thank you for being my polar star, surprising me for your kindness.

I have certainly forgotten someone, but I am sincere when I say that I am grateful to all those with whom I have even just spent a few words or who have responded to even just one of my doubts or curiosities. I can't wait to reciprocate all the kindness that has been given to me.

Last but not least, I sincerely want to acknowledge the incredible support of my family, in particular my dad. He has always supported me in many moments of difficulty, always giving support and being a safe place.