Title	Development of a GPU-based High Level Trigger for the J-PARC KOTO experiment
Author(s)	Gonzalez, Mario
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/101909
rights	
Note	

December 2024

Doctoral Thesis

Development of a GPU-based High Level Trigger for the J-PARC KOTO Experiment

Mario Gonzalez Carpintero

Department of Physics, Graduate School of Science
Osaka University

Abstract

The main aim of the J-PARC KOTO experiment is to measure the branching ratio of the CP-violating $K_L \to \pi^0 \nu \bar{\nu}$ decay. To improve our understanding of its background contributions and to enhance the precision of this measurement, several categories of physics events need to be collected on top of the $K_L \to \pi^0 \nu \bar{\nu}$ candidates. To cope with these demands, the KOTO data acquisition system has been upgraded before the beam-time in 2024. Particularly, a GPU-based High Level Trigger (HLT) has been developed.

The new HLT could capture incoming data with a negligible packet loss. Event reconstruction, selection, and compression could be performed on GPUs to reduce the data size to 18% of itself. The event loss due to HLT inefficiencies was decreased from 3.5% at the beginning of the 2024 beam-time to 0.4% at the end.

The new HLT in 2024 allowed for the collection of enough data to improve the precision of the $K_L \to 2\pi^0$ background estimation by an expected factor of 3. K^+ candidate decays were also collected to measure the K^+ flux into the KOTO detector. $K_L \to \pi^0 e^+ e^-$ candidates were continuously recorded too to study the feasibility of a future $K_L \to \pi^0 e^+ e^-$ search.

The usage of GPUs in the HLT allowed it to cope in real-time with the high data rate and computing demands. The development of the KOTO GPU-based HLT, as well as its performance during the 2024 beam-time are presented in this thesis.

Contents

Chapter 1

Introduction

The Standard Model (SM) in particle physics is the theoretical framework that compiles our best understanding on elementary particles and their interactions. However, the SM is not a complete theory. Phenomena such as the existence of dark matter, or the large imbalance between matter and antimatter observed in the Universe, are not explained by the SM. Physics beyond the Standard Model (BSM) that explains these effects is continuously being searched for with experiments.

One of the most widely known facts evidencing the incompleteness of the SM is indeed the matter-antimatter asymmetry in the Universe. While the SM allows the asymmetric production of matter and antimatter through charge-parity (CP) violation processes, the predicted extent of CP violation is not sufficient to explain the large imbalance observed in the Universe [1].

The search for new physics beyond the SM has always been a hot topic in the particle physics community. Accelerators such as the LHC have been built to explore new physics at the *energy frontier*, by studying physics processes in collisions at the highest achievable energies. Other accelerators and experiments probe the *intensity frontier*, by collecting large amounts of data to search for rare processes where physics beyond the Standard Model could contribute.

1.1 CP violation in the neutral kaon system

The neutral kaon K^0 is a state composed of an s and a d quarks. The states K^0 and $\bar{K^0}$ can be described as $|K^0\rangle = |d\bar{s}\rangle$ and $|\bar{K^0}\rangle = |\bar{d}s\rangle$ respectively. The C (charge conjugation) operator exchanges a particle with its antiparticle, and the P (parity) operator inverts all the spacial coordinates. Applying the CP operator to the $|K^0\rangle$ state gives $CP\,|K^0\rangle = CP\,|d\bar{s}\rangle = |\bar{d}s\rangle = |\bar{K^0}\rangle$. Conversely, $CP\,|\bar{K^0}\rangle = |K^0\rangle$. A basis of CP eigenstates in the neutral kaon system can be constructed as

$$|K_1\rangle = \frac{1}{\sqrt{2}}\left(|K^0\rangle + |\bar{K^0}\rangle\right), \text{ and } \; |K_2\rangle = \frac{1}{\sqrt{2}}\left(|K^0\rangle - |\bar{K^0}\rangle\right). \tag{1.1}$$

The state $|K_1\rangle$ has CP eigenvalue +1, and it is said to be CP-even. The state $|K_2\rangle$, with CP eigenvalue -1, is said to be CP-odd.

The neutral kaon can decay into two or three pion states. The two-pion state is CP-even, while the three-pion state is CP-odd. If CP was conserved, K_1 would always decay into two pions, and K_2 would always decay into three pions.

The mass eigenstates K_L (K-long) and K_S (K-short) can be constructed as a mixture of the CP eigenstates K_1 and K_2 as follows:

$$|K_L\rangle = \frac{1}{\sqrt{1+|\epsilon|^2}}\left(|K_2\rangle + \epsilon\,|K_1\rangle\right) \tag{1.2}$$

$$|K_S\rangle = \frac{1}{\sqrt{1+|\epsilon|^2}}\left(|K_1\rangle + \epsilon\,|K_2\rangle\right), \tag{1.3}$$

The parameter ϵ is small and $K_L \approx K_2$. The K_L branching ratio to three pions is 32%, while its branching ratio to the CP-violating two pion final states is 10^{-3} [2]. The higher mass of the three-pion final state restricts the phase space available for the K_L to decay, which therefore has a longer lifetime compared to the K_S.

1.2 $K_L \to \pi^0\nu\overline{\nu}$ decay as a probe for physics beyond the Standard Model

Among all the CP-violating K_L decays, the rare kaon decay $K_L \to \pi^0\nu\overline{\nu}$ provides a unique opportunity to search for new physics beyond the Standard Model. In the SM, this decay happens through a process in which a *strange* quark transitions to a *down* quark[1]. This transition is forbidden at tree level in the SM. However, this decay can happen through loop diagrams mediated by a W boson and a t quark[2], as exemplified in Fig. 1.1.

[1] Processes like this one, in which the flavor is changed but the charge is conserved are known as Flavor Changing Neutral Current (FCNC) processes.

[2] Contributions from u and c quarks are greatly suppressed in this decay due to the GIM mechanism.

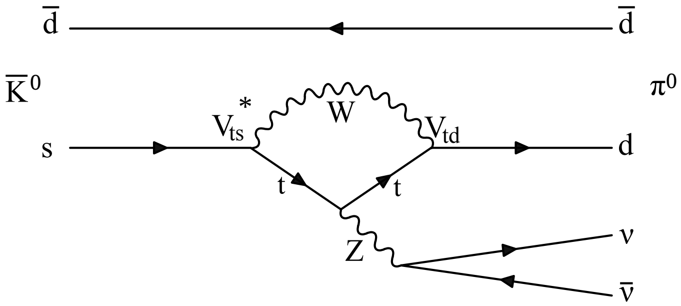


Figure 1.1: One of the Feynman diagrams through which the $K_L \to \pi^0\nu\overline{\nu}$ decay is allowed in the SM.

The parameters V_{ts}^* and V_{td} in Fig. 1.1 are elements of the Cabibbo-Kobayashi-Maskawa (CKM) matrix. The CKM matrix describes the quark flavor transitions. The strength of the $s \to d$ and $t \to d$ transitions in Fig. 1.1 is given by the parameters $|V_{ts}|$ and $|V_{td}|$ respectively. The CKM matrix is given by:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}. \tag{1.4}$$

The CKM matrix can also be expressed in terms of the parameter $\lambda = |V_{us}| = 0.22501 \pm 0.00068$ [2] as follows:

$$V_{CKM} = \begin{pmatrix} 1-\lambda^2/2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\lambda^2/2 & A\lambda^2 \\ A\lambda^3(1-\rho-i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4), \tag{1.5}$$

where A, ρ, and η are real coefficients[3].

The amplitude of the $K_L \to \pi^0\nu\overline{\nu}$ decay can be given in terms of the CKM matrix elements as follows:

$$A(K_L \to \pi^0\nu\overline{\nu}) \simeq A(K_2 \to \pi^0\nu\bar{\nu}) = \frac{1}{\sqrt{2}}\left(A(K^0 \to \pi^0\nu\bar{\nu}) - A(\bar{K^0} \to \pi^0\nu\bar{\nu})\right) \tag{1.6}$$

$$\propto V_{td}^*V_{ts} - V_{td}V_{ts}^* = -2\mathrm{Im}(V_{td}^*V_{ts}). \tag{1.7}$$

If CP was conserved, the amplitude of any particle decay would be equal to the amplitude of its CP-conjugate decay. For this to happen all the CKM matrix elements would have to be real. The scale of the CP violation in a particle decay is therefore given by the imaginary

[3] The experimental values of λ, A, ρ, and η are given in Section 12.4 of Ref. [2].

part of its amplitude. The amplitude of the $K_L \to \pi^0\nu\overline{\nu}$ decay being imaginary makes the $K_L \to \pi^0\nu\overline{\nu}$ decay CP-violating.

On top of CP-violating, the branching ratio of the $K_L \to \pi^0\nu\overline{\nu}$ decay is also predicted to be small for several reasons. First, its amplitude is proportional to the product of the CKM matrix elements V_{td}^* and V_{ts}, which are of order λ^3 and λ^2 respectively. Second, the $K_L \to \pi^0\nu\overline{\nu}$ decay is a loop process, suppressed with respect to the tree-level K_L decays. Third $K_L \to \pi^0\nu\overline{\nu}$ decay is suppressed due high mass of the t, W and Z involved in the decay.

The $BR(K_L \to \pi^0\nu\overline{\nu})$ prediction in the SM is $(2.94 \pm 0.15) \times 10^{-11}$ [3]. Most of the uncertainty in this prediction comes from the experimental determination of the CKM matrix elements V_{td} and V_{ts}. The absence of long-distance (low-energy) contributions to this process makes the theoretical error of its branching ratio just $\mathcal{O}(2\%)$.

Possible extensions to the SM contributing to the $K_L \to \pi^0\nu\overline{\nu}$ decay are discussed in Ref. [4]. The simplest one postulates the existence of a heavy and neutral gauge boson Z' that mediates the $s \to d$ transition at tree level, directly changing the flavor (which the SM's Z cannot). Figure 1.2 illustrates this process.

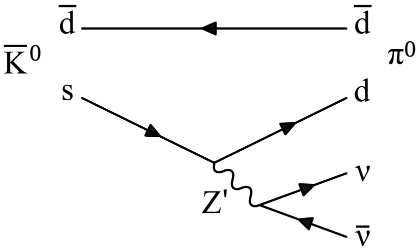


Figure 1.2: The BSM Z' boson mediates the $s \to d$ transition at tree level.

This process contributes at tree level, and thus its contribution to the $K_L \to \pi^0\nu\overline{\nu}$ branching ratio can be noticeable even for Z' boson masses as large as several hundreds of TeV, which are beyond the LHC reach. Direct searches at Tevatron have excluded the existence of a Z' up to a mass of 850 GeV/c^2 (95% C.L.) [5]. KOTO aims to reach a sensitivity to the $K_L \to \pi^0\nu\overline{\nu}$ branching ratio of $\mathcal{O}(10^{-10})$ by the end of its lifetime, which will make it sensitive to Z' masses up to 1 TeV/c^2 [6].

New physics models involving new sources of CP violation that could explain the matter-antimatter asymmetry in the Universe have also been proposed [7]. Some of these models predict the branching ratio of the $K_L \to \pi^0\nu\overline{\nu}$ and other CP-violating kaon decays to deviate from their current SM predictions. These models too can be tested with future precise measurements of these decays.

1.3 Introduction to the J-PARC KOTO experiment

The KOTO experiment is located at the Japan Proton Accelerator Research Complex (J-PARC) in the Ibaraki prefecture, Japan. The main purpose of the KOTO experiment is to search for the $K_L \to \pi^0 \nu\overline{\nu}$ decay.

The J-PARC accelerator provides a 30 GeV high-intensity proton beam. The beam is supplied in beam-on beam-off repetition cycles of 4.2 s (2-s beam-on, 2.2-s beam-off). This proton beam is injected into a gold target. K_L's are produced in the proton collisions with the target, some of which will eventually decay in the KOTO detector. More information about the J-PARC accelerator and the KOTO beamline will be given in section 2.1.

The final state of the main physics target of KOTO, $K_L \to \pi^0 \nu\overline{\nu}$, includes two measurable photons from the π^0 decay and two neutrinos that cannot be measured. A special detector system is needed to identify this decay, particularly to distinguish it from other processes producing similar final states.

1.3.1 Introduction to the KOTO detector

The KOTO detector system is composed of multiple detectors. A calorimeter, red in Fig. 1.3, is used to measure the energy and hit position of the two photons from the $K_L \to \pi^0 \nu\overline{\nu}$ decay. Ensuring that no detectable particle is left undetected requires the full kaon decay volume to be hermetically sealed by veto detectors (blue in the figure). The likelihood of beam particles interacting with air molecules is minimized by keeping the K_L decay volume in the KOTO detector at around 10^{-5} Pa.

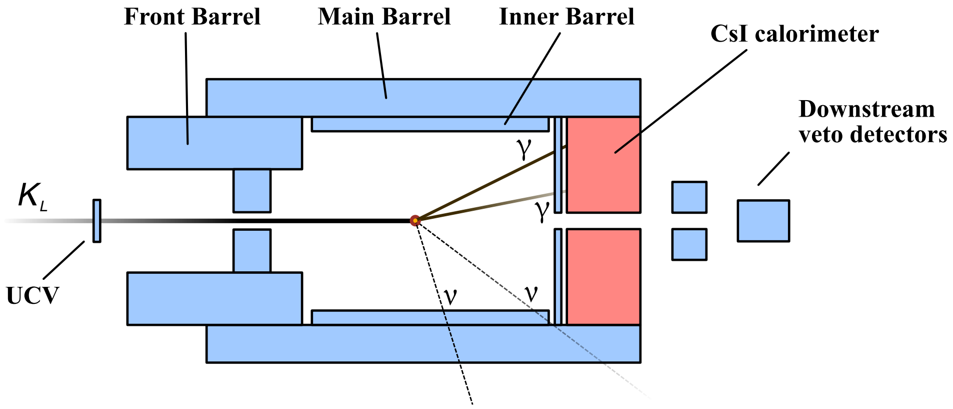


Figure 1.3: Simplified view of the KOTO detector. The CsI calorimeter is shown in red, and the veto detectors in blue. A schematic detector view to scale will be given later in Sec. 2.4.

KOTO's calorimeter is composed of 2716 channels. Each channel corresponds to a 50-cm long CsI (Cesium Iodide) crystal[4] of size 2.5 × 2.5 cm or 5 × 5 cm, read out from the back by

[4] 50 cm is equivalent to 27 radiation lengths. The Moliere radius of the CsI is 3.57 cm, thus showers spread

a PMT. The channels are arranged as shown in Fig. 1.4.

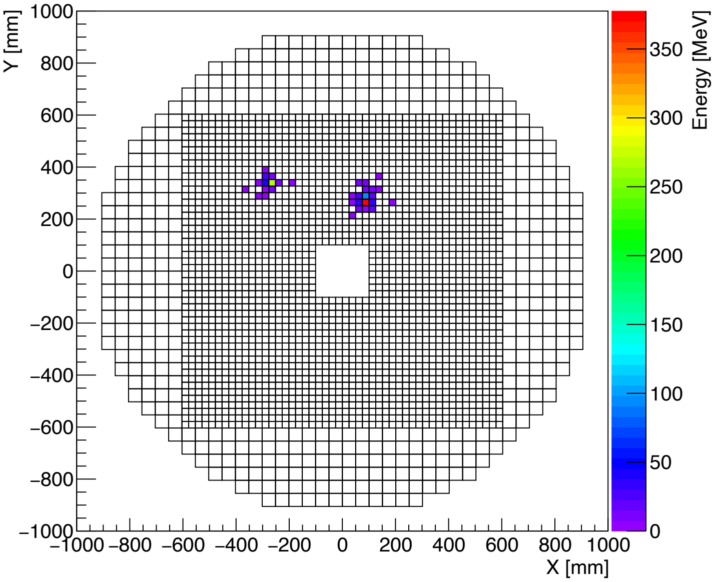


Figure 1.4: A Schematic view of KOTO's calorimeter, showing an event with two clusters.

A candidate $K_L \to \pi^0\nu\overline{\nu}$ decay would produce two hits in the calorimeter from the decay of the π^0, with no other signal in any other detector. The transverse momentum of the two-photon final state is expected to be non-zero, due to the contribution from the two neutrinos that were not detected. In Fig. 1.4, the measured final state transverse momentum points upwards. If this were a $K_L \to \pi^0\nu\overline{\nu}$ decay, the combined momentum of the two missing neutrinos would be expected to point down, to satisfy the momentum conservation.

1.3.2 Introduction to the $K_L \to \pi^0\nu\overline{\nu}$ search at KOTO

$K_L \to \pi^0\nu\overline{\nu}$ decays are referred to as *signal* in the $K_L \to \pi^0\nu\overline{\nu}$ search. Processes other than the signal that produce detector signatures that mimic a $K_L \to \pi^0\nu\overline{\nu}$ decay are referred to as *background*. Due to the large amount of background processes in the $K_L \to \pi^0\nu\overline{\nu}$ analysis, a large data sample and a strong background suppression are required for the $K_L \to \pi^0\nu\overline{\nu}$ search. One of the goals of the KOTO analysis is to find an event selection criteria that keeps the $K_L \to \pi^0\nu\overline{\nu}$ signal efficiency high and the background predictions low. A large effort is also put into reducing the uncertainties of the background estimations to make the signal identification clear.

Accumulating data depends on the beam intensity and beam-time provided by the accelerator. KOTO has been collecting data at J-PARC since 2013. As data is accumulated, KOTO gradually improves its *single event sensitivity* on the $K_L \to \pi^0\nu\overline{\nu}$ decay.

The single event sensitivity (SES) gives a measure of what the $K_L \to \pi^0\nu\overline{\nu}$ branching

across multiple crystals.

ratio would be if one signal event was observed at KOTO among a collected dataset of kaon decays. If the true $K_L \to \pi^0 \nu\overline{\nu}$ branching ratio is equal to its SM prediction, we would expect to have observed one signal event by the time the KOTO SES reaches the SM's predicted BR($K_L \to \pi^0 \nu\overline{\nu}$)[5].

The KOTO single event sensitivity in 2024 reached $(9.26 \pm 0.75) \times 10^{-10}$ [8], still two orders of magnitude higher than the $K_L \to \pi^0 \nu\overline{\nu}$ branching ratio. Enough data to lower the single event sensitivity beyond 10^{-10} is expected to be collected by KOTO within the next decade. To help us improve it faster, the J-PARC accelerator has been increasing its beam power[6]. In 2024, the beam power was increased from its original 64 kW to 80 kW, by reducing the beam-off stage of the beam cycle from 3.2 s to the current 2.2 s. Eventually, J-PARC aims to provide a 100-kW beam. KOTO's original data acquisition (DAQ) system, which will be briefly described in section 1.5, could not cope with a 100 kW beam and needed to be redesigned.

1.4 Introduction to the data acquisition at KOTO

The offline analysis of the $K_L \to \pi^0 \nu\overline{\nu}$ decay involves collecting more than just $K_L \to \pi^0 \nu\overline{\nu}$ candidate events. Different data need to be taken to measure detector inefficiencies, to calibrate detectors, or to measure the inefficiency of the online trigger system among other purposes. During physics data taking at KOTO, the trigger collecting $K_L \to \pi^0 \nu\overline{\nu}$ candidates represents less than 10% of the total collected data volume. Other triggers collected during physics runs will be covered in more detail in section 3. They include the following.

- The minimum bias trigger, collecting events satisfying minimal trigger requirements. This trigger is primarily used to monitor the bias introduced by the online event reconstruction and selection in the collected data.
- The $K^\pm \to \pi^\pm \pi^0$. We mainly use this trigger to measure the K^+ flux into the KOTO detector and calibrate Monte Carlo simulations. Hereafter this trigger will be referred to as the K^+ trigger.
- The 6-cluster $K_L \to 3\pi^0$ trigger, where all pions decay into two photons, and all six final-state photons hit the calorimeter. These events are mainly used for the offline energy calibration of KOTO's calorimeter channels. Hereafter this trigger will be referred to as the 6γ trigger.
- The 5-cluster $K_L \to 3\pi^0$, where only one of the six final-state photons escapes the calorimeter. These events are used as a data-driven method to measure the inefficiencies of the veto detector hit by the sixth photon. Hereafter this trigger will be also referred to as the 5γ trigger.

[5] The actual calculation of KOTO's SES is detailed in Appendix E.

[6] The beam power is defined as $E_p \cdot N_P/t$, where E_p is the energy of the protons (30 GeV), N_p is the number of protons per spill and t is the spill cycle length.

All triggers mentioned above are directly related to the $K_L \to \pi^0 \nu \overline{\nu}$ search. The statistical uncertainties of some of the measurements obtained from these events can only be reduced by increasing the amount of data collected for this purpose. In particular, the amount of data available for the K^+ trigger has a direct impact on the measurement of the $K^\pm$ flux into the KOTO detector. Knowing with precision this flux is needed to accurately estimate the backgrounds from K^+ decays. The statistics available from the 5γ trigger has a direct impact on the $K_L \to 2\pi^0$ background estimation. These two backgrounds will be covered in more detail in section 2.3.

Due to limitations in the original DAQ system of KOTO, the bandwidth reserved for these triggers had to be constrained by introducing *prescale factors*[7]. For a trigger with a prescale factor of P, only one of every P events would be collected. To improve the precision of these measurements in future analyses, prescale factors need to be reduced. To afford reducing prescale factors, a higher DAQ bandwidth is required.

For reference, the trigger menu of a KOTO run including the 5γ trigger in 2020, before the DAQ upgrade presented in this thesis, is shown in table 1.1, together with the prescale factors applied to each of the collected triggers.

Table 1.1: The eight triggers enabled in a 2020 physics run that included the 5γ trigger. The average trigger rate in this run was 11.3 kEvents/spill.

Trigger ID	Trigger	Prescale factor
1	$K_L \to \pi^0 \nu \overline{\nu}$ trigger	1
2	Normalization trigger	30
3	Minimum bias trigger	1
4	–	–
5	–	–
6	6γ trigger	3
7	Minimum bias trigger (off-spill)	1
8	5γ trigger	5/4

Triggers 1 to 3, that will be covered in more detail in chapter 3, have been taken with the same prescale factors in 2024 runs. The minimum bias trigger (off-spill), represents less than 2% of the total event rate and is not relevant in this discussion. Before the DAQ upgrade presented in this thesis, taking the 5γ trigger implied giving up the K^+ trigger and adding a prescale factor of 3 to the 6γ trigger, to keep the trigger rate below the DAQ bandwidth constraints at the time.

Finally, although the $K_L \to \pi^0 \nu \overline{\nu}$ search is KOTO's main purpose, the KOTO experiment

[7] Prescale factors are also used to prevent very loose triggers, such as the minimum bias trigger, form saturating the DAQ system.

has the capability to perform other physics searches. A particularly interesting example is the $K_L \to \pi^0 e^+ e^-$ decay, which has already been extensively searched for by KTeV [9]. Despite its physics interest, the $K_L \to \pi^0 e^+ e^-$ search could not be performed by KOTO due to bandwidth limitations of its data acquisition system. A measurement of the $K_L \to \pi^0 e^+ e^-$ branching ratio is already out of the reach of KOTO, but the study of this decay with KOTO data can provide valuable feedback towards the design of KOTO's successor, the KOTO II experiment [10]. KOTO II, currently under development, plans to include the $K_L \to \pi^0 e^+ e^-$ decay in its physics program.

Enabling the KOTO experiment to collect in parallel $K_L \to \pi^0 \nu \overline{\nu}$ and $K_L \to \pi^0 e^+ e^-$, leaving as well margin for other possible searches, requires a major upgrade of the DAQ system.

1.5 Data acquisition system of the KOTO experiment before this work

Until 2024, the J-PARC accelerator delivered beam to KOTO following a 2-s beam-on and 3.2-s beam-off cycle. During beam-on, data was collected from each detector channel into Analog to Digital Converter (ADC) modules. As illustrated in Fig. 1.5, the outputs of all 16 ADC modules per ADC crate converge into an assembly board. The assembly boards cannot simultaneously read and write their memory. For this reason, they would make use of two independent memory banks, one for reading and one for writing. Each spill would be written into one of those banks during beam on. During beam off, the spill would start being sent out to the PC farm. The next spill would be written into the other bank, sometimes while the previous spill is still being sent. Due to the bandwidth between each board and the PC farm being limited to 1 Gbps, the DAQ throughput before 2024 was limited to around 12 kEvents per beam cycle.

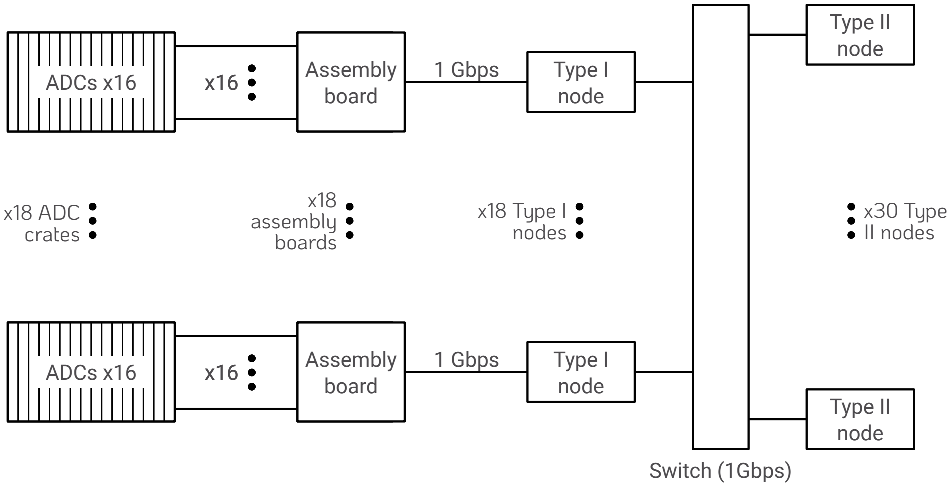


Figure 1.5: Overview of the DAQ system before the upgrade presented in this thesis.

At the PC farm, each of the 18 *Type-I* nodes shown in Fig. 1.5 would receive partial events from one assembly board. Event parts would then be sent to a second layer of *Type-II* nodes. The target Type-II node is selected based on event ID, so that all 18 parts of each event end up in the same Type-II node. The Type-II nodes would perform the event building, and send the events to disk arrays for temporary storage. Data is finally sent from the disk arrays to the KEK computing center, located at KEK in Tsukuba, around 60 km from J-PARC.

The limited memory of the assembly boards and the bandwidth limitation between them and the PC farm motivated data compression to be performed at the ADCs, providing an overall reduction factor of 3. On top of this, the ADC data was formatted in a way that makes its processing in a computer very limited by a memory-intensive task, as will be described later in section 4.7.1. Since the ADC data is compressed before leaving the ADCs, re-formatting it at the assembly boards was not possible. Therefore, any sort of event reconstruction at the Type-II nodes would need to be done on top of the event building, waveform decompression, ADC data re-formatting, and waveform compression again. These CPU and memory-intensive tasks did not leave any room for event selection, limiting the capabilities of the computer farm to just event building.

Other than the constraints in bandwidth and processing capabilities, it should be noted that a failure in any of the 18 Type-I nodes would cause a complete loss of data, as part of every event flows through each Type-I node. On top of the motivations introduced in the previous sections, The new DAQ system would be required to be flexible enough to cope with hardware failures without a total data loss.

1.6 Motivation of this thesis

The work presented in this thesis is mainly motivated by the need for a high-bandwidth DAQ with complex event selection capabilities. A high-bandwidth DAQ system is needed to cope with the ongoing beam intensity upgrade at J-PARC, to improve the precision of background estimations that are obtained with data-driven methods, and to enable the collection of events targeting alternative physics searches as has been introduced in previous sections. Towards this goal, the assembly boards need to be replaced by higher-throughput modules, and the PC farm needs to be made capable of capturing their output at an increased rate.

However, increasing the bandwidth is not efficient if a large fraction of the collected data is unusable or uninteresting. Realtime event reconstruction, and a *trigger system* based on the physics interest of each event, is needed to both increase the quality of the collected data and reduce its size. Reducing the data size is motivated by the limited bandwidth between J-PARC and KEK, and by the need to keep reasonably low the resources needed at KEK for the permanent storage of the collected data.

Most analyses performed in KOTO with the collected data involve preliminary quality-based selection criteria, aiming to reject, among others, events whose energy cannot be measured with good enough precision. These include, for example, events with hits at the edge of the calorimeter, where part of the hit energy is likely to have been lost. These data could be rejected online instead, reducing both the load of the DAQ system and the offline data storage requirements.

1.7 Purpose of this thesis

The issues introduced above motivate the upgrade of the KOTO DAQ system, and in particular the complete replacement of the PC farm, by the software-based trigger system (hereafter *High Level Trigger* or HLT) presented in this thesis. The requirements for the HLT will be elaborated in section 5.1, after the introduction of the new KOTO DAQ system. In this section, they are summarized as follows:

- receiving data at the highest output rate of the upgraded DAQ modules, 40 Gbps, with a packet loss below 0.01%,
- performing physics event reconstruction and selection with an efficiency around 99%, and pedestal suppression with an inefficiency below 0.1%,
- combining the effect of event selection, pedestal suppression and data compression, keeping the data rate at the HLT output below the 4 Gbps bottleneck between J-PARC and the KEK computing center,
- processing events at a rate faster than the DAQ system limit of 50 kEvents/spill.

The new HLT will be also required to be able to minimize the impact of potential hardware errors affecting a single node, and to show its potential to cope with future additional data collection requirements or upstream DAQ system upgrades.

This thesis presents the work done to overcome the DAQ limitations outlined in this chapter, and in particular to develop a High Level Trigger satisfying the above requirements. The HLT was commissioned and maintained during the 2024 beam-time by the author of this thesis. Physics data which is essential for the future physics output of the KOTO experiment has been produced in part thanks to the work presented in this thesis. The quality of these data is also studied in this thesis.

Chapter 2

The KOTO Experiment

The KOTO experiment is conducted at J-PARC. The features of the J-PARC accelerator relevant to the KOTO experiment are given in chapter 2.1. The event reconstruction and $K_L \to \pi^0 \nu\overline{\nu}$ identification at KOTO are introduced in section 2.2. The importance of background reduction and accurate estimation in KOTO is then motivated in section 2.3. The two background processes whose estimation mostly benefits from the work presented in this thesis are presented in more detail.

2.1 J-PARC accelerator and KOTO beamline

An overview of the J-PARC accelerator is given in Fig. 2.1. Negative Hydrogen ions are first accelerated in a Linear Accelerator (LINAC). The two electrons are removed from the ions, and the remaining protons are accelerated up to 3 GeV in a synchrotron. The 3 GeV protons are injected into the Main Ring (MR, green in the figure), a larger synchrotron where they are further accelerated up to 30 GeV. The 30 GeV protons are eventually extracted from the MR and delivered to the Hadron Experimental Facility, where the KOTO detector is located.

The proton extraction from the MR lasts for 2 seconds. The injection, acceleration, and extraction cycle is repeated with a period of 4.2 seconds. The cycle length was decreased from 5.2 s to 4.2 s in 2021, increasing the effective beam power by 20%.

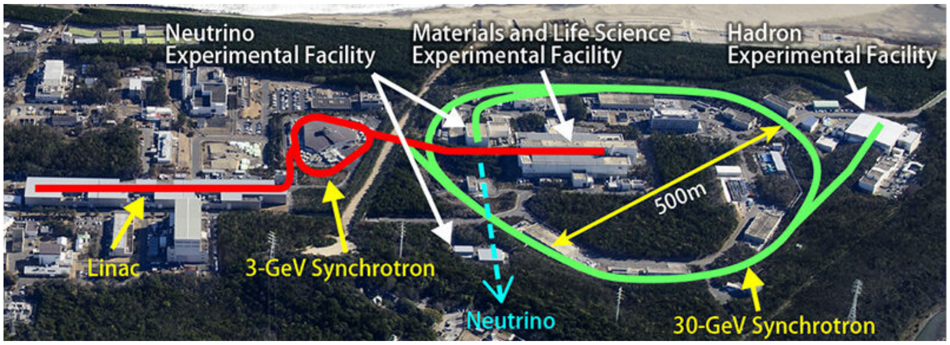


Figure 2.1: The J-PARC accelerator [11].

At the Hadron Experimental Facility, the 30 GeV proton beam is injected into a gold target producing, among other particles, kaons. The KOTO beamline, outlined in Fig. 2.2, is extracted from the primary beamline at an angle of 16°. The distance between the gold target and the KOTO detector is 21.5 m, for the short-lived particles to decay before reaching the KOTO detector. Two collimators are also placed before the KOTO detector to narrow down the width of the beam. A magnet is installed between the collimators to deflect charged particles from the path to the KOTO detector. A beam-plug is installed between the collimators, that can be closed if needed, stopping beam particles but pions and muons. The plug is closed during special detector calibration runs (section 3.3), and to minimize radiation when maintenance work in the detector area is needed during beam-time. Since 2024, a permanent magnet has been installed at the downstream end of the second collimator to reduce the flux of charged kaons produced from K_L interacting with the second collimator, after the first sweeping magnet.

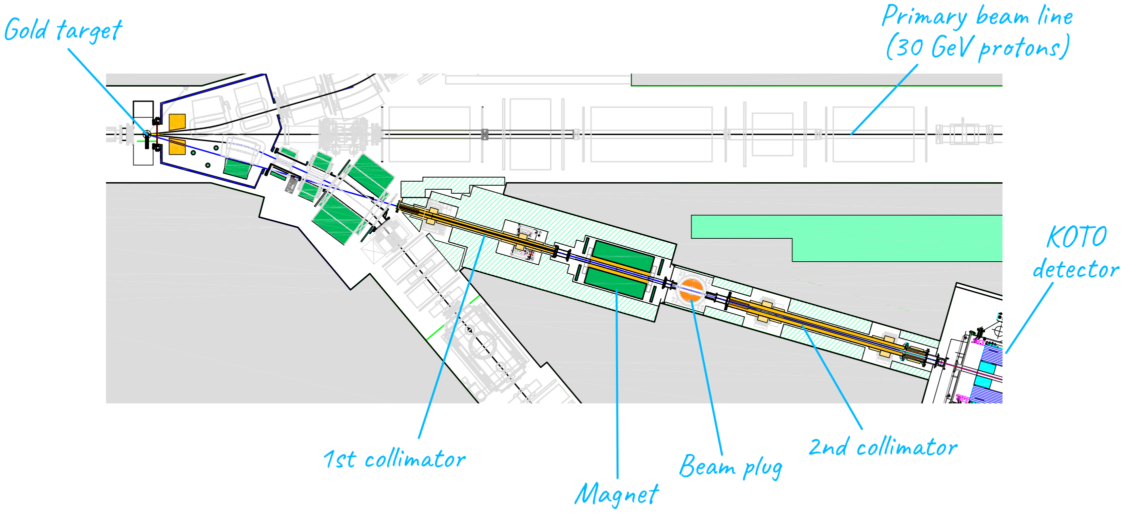


Figure 2.2: Schematics of the KOTO beamline.

The beam entering the KOTO detector contains, among other neutral particles, long-lived kaons (K_L^0). Some of these kaons decay in KOTO's decay volume and are measured at the KOTO detector. Among all possible kaon decays, the characteristic detector signature of the

$K_L \to \pi^0 \nu \overline{\nu}$ decay is introduced in section 2.2.

2.2 Signal identification

The $K_L \to \pi^0 \nu \overline{\nu}$ signature in the KOTO detector was outlined in Fig. 1.3. The short decay time of the neutral pion $\tau_{\pi^0} = 8.4 \cdot 10^{-17}$ s makes its decay instantaneous, thus the π^0 decay position can be taken as the K_L decay position ($c\tau_{\pi^0} = 25$ nm). The $K_L \to \pi^0 \nu \overline{\nu}$ decay is expected to produce two hits in the CsI calorimeter from the two photons produced by the π^0 decay, and no signal in any other detector.

The π^0 is reconstructed from the two photons in the calorimeter. This is done assuming that the photons come from a π^0 (so the π^0 mass can be imposed) and that the decay point is on the beam axis, i.e. the beam has no width. The opening angle (θ in Fig. 2.3) between the two photons can be calculated from the conservation of the 4-momentum in the $\pi^0 \to \gamma\gamma$ decay. Denoting P the 4-momentum,

$$\begin{aligned}
P_{\pi^0}^2 &= (P_{\gamma_1} + P_{\gamma_2})^2 \\
m_{\pi^0}^2 &= m_{\gamma_1}^2 + m_{\gamma_2}^2 + 2P_{\gamma_1} \cdot P_{\gamma_2} \\
&= 0 + 0 + 2E_{\gamma_1} E_{\gamma_2} - 2\vec{p}_{\gamma_1} \cdot \vec{p}_{\gamma_2} \\
&= 2E_{\gamma_1} E_{\gamma_2} - 2p_{\gamma_1} p_{\gamma_2} \cos\theta \\
&= 2E_{\gamma_1} E_{\gamma_2} (1 - \cos\theta),
\end{aligned}$$

from where

$$\cos\theta = 1 - \frac{m_{\pi^0}^2}{2E_{\gamma_1} E_{\gamma_2}} \ , \tag{2.1}$$

where E_{γ_1} and E_{γ_2} are the energies of the two photons and m_{π^0} is the mass of the π^0, 134.9766 GeV[12].

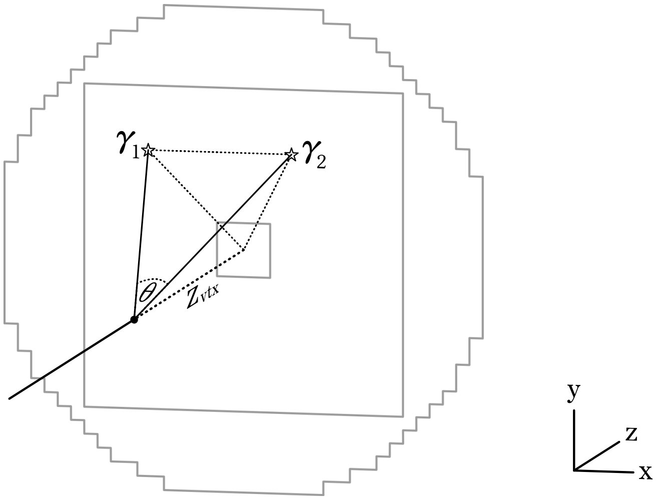


Figure 2.3: The opening angle between the two photons in the $\pi^0 \to \gamma\gamma$ decay.

The K_L and the pion are assumed to decay in the beam axis. Its z coordinate Z_{vtx} is calculated from θ. From here, the pion transverse momentum is calculated, which is expected to be non-zero to compensate for the undetected neutrinos.

The *Signal Region* (SR) is defined as a region in the multidimensional space of physics observables were both signal efficiency and background rejection are high. Fig. 2.4 shows the projection of the SR in the $P_t - z$ plane, where P_t is the transverse component of the momentum reconstructed from the two photons and z is the pion decay position on the Z axis. To minimize the human bias on the $K_L \to \pi^0 \nu\overline{\nu}$ analysis, a *blind region* (black in the figure) is constructed around the signal region, and data points inside this region are not uncovered during the analysis.

Monte Carlo (MC) simulations are calibrated and tested in *Control Regions* (CRs). Control regions by construction do not share any events with the signal regions. CRs are often defined by inverting just one of the selection criteria that defines the SR. Generally, looser cuts[1] are used in control regions to include more events. Possible discrepancies between data and MC need to be well understood in the CRs before *unblinding* the signal region.

[1] The word *cut* is used in high-energy physics as a synonym for *selection criterion*.

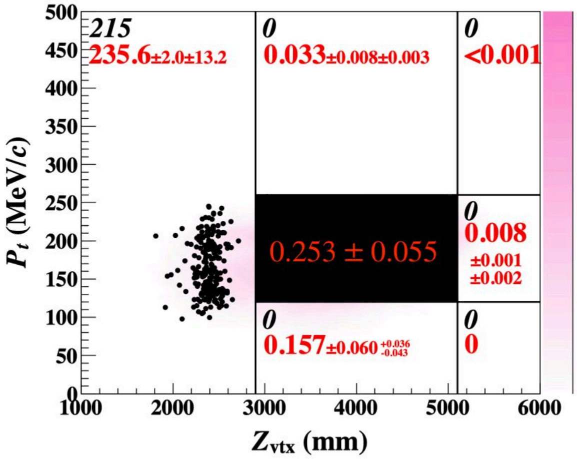


Figure 2.4: The Signal Region in the $P_t - z$ plane, being P_t the transverse momentum of the reconstructed pion and z the K_L decay position, along the beam axis. Black numbers correspond to the observed events. Red corresponds to the expected events. Figure taken from Ref. [8].

.

After fixing the selection criteria and unblinding the signal region, the number of observed events inside is compared to the background expectation, and the upper limit of the $K_L \to \pi^0 \nu\overline{\nu}$ branching ratio can be updated.

2.3 Background estimation

In principle, any process recorded in the KOTO detector that is not a $K_L \to \pi^0 \nu\overline{\nu}$ decay becomes a background in the $K_L \to \pi^0 \nu\overline{\nu}$ search.

Measuring the $K_L \to \pi^0 \nu\overline{\nu}$ branching ratio involves counting the number of observed $K_L \to \pi^0 \nu\overline{\nu}$ decays in its signal region. However, observed events could be attributed to background processes if the background prediction is not close to zero, or if the uncertainties in the background prediction are large. These uncertainties propagate to the upper limit of the $K_L \to \pi^0 \nu\overline{\nu}$ branching ratio. Minimizing them is an important goal of the KOTO analysis.

Many background processes can be excluded by simple event selection, such as selection based on the number of clusters recorded in the CsI calorimeter or on the presence or absence of final-state charged particles. Other processes are more complicated, and in some cases indistinguishable from the signal. The background estimation in KOTO's latest $K_L \to \pi^0 \nu\overline{\nu}$ analysis is summarized in Table 2.1 This analysis was performed with data collected in 2021, before adding the permanent magnet to reduce the K^+ background.

Table 2.1: Background estimation in the KOTO analysis with 2021 data [8]. The uncertainties given for each estimation are the statistical and systematic respectively.

Source		**Expected number of events**
$K^{\pm}$		$0.042 \pm 0.014\,(\text{stat})^{+0.004}_{-0.029}\,(\text{syst})$
K_L	$K_L \to 2\gamma$ (beam halo)	$0.045 \pm 0.010\,(\text{stat}) \pm 0.006\,(\text{syst})$
	$K_L \to 2\pi^0$	$0.059 \pm 0.022\,(\text{stat})^{+0.051}_{-0.060}\,(\text{syst})$
Hadronic cluster		$0.024 \pm 0.004\,(\text{stat}) \pm 0.006\,(\text{syst})$
	CV-η	$0.023 \pm 0.010\,(\text{stat}) \pm 0.005\,(\text{syst})$
	Upstream-π^0	$0.060 \pm 0.046\,(\text{stat}) \pm 0.007\,(\text{syst})$
Total		$0.253 \pm 0.055\,(\text{stat})^{+0.052}_{-0.067}\,(\text{syst})$

In this thesis, we will concentrate on the K^+ and the $K_L \to 2\pi^0$ backgrounds, as their estimation requires dedicated data to be taken together with the signal candidates during physics runs.

K^+ background

Charged kaon decays might become a background if a K^+ is produced by a K_L interacting with the second collimator, fails to be detected by the UCV detector, and then it decays producing a π^0 and other particles that are missed. This situation is illustrated in Fig. 2.5. This background has been recently reduced by upgrading an Upstream Charged Veto detector [13] and installing a permanent magnet after the second collimator. Evaluating this background requires computing the $K^{\pm}$ / K_L flux ratio, which requires the collection of $K^+ \to \pi^+\pi^0$ candidate events during physics runs.

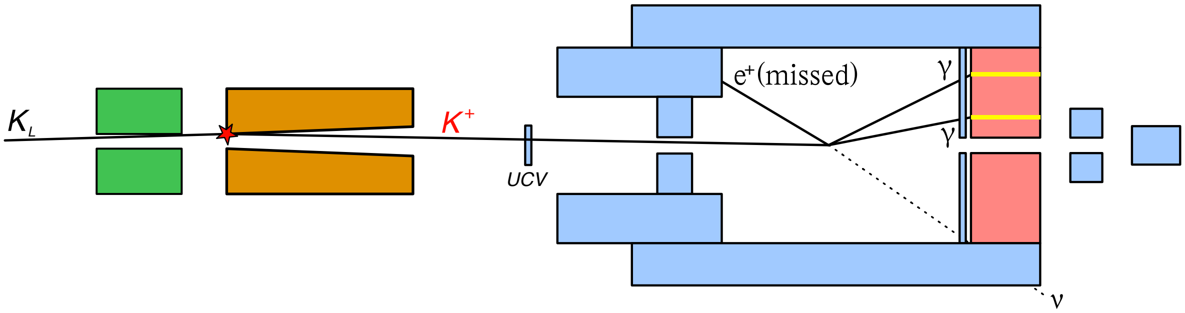


Figure 2.5: A K^+ is produced by a K_L interacting with the collimator material. This kaon fails to be detected by the Upstream Charged Veto detector (UCV) due to inefficiencies. The K^+ decays to $\pi^0 e^+ \nu$. The photons from the neutral pion are measured in the calorimeter. The low-energy electron is missed.

$K_L \to 2\pi^0$ background

The $K_L \to 2\pi^0$ decay will also become a background if the two photons from a π^0 are missed due to veto detector inefficiencies, and the two remaining photons hit the calorimeter, as illustrated in Fig. 2.6. In this case, the $K_L \to 2\pi^0$ event will leave a detector signature indistinguishable from the $K_L \to \pi^0\nu\overline{\nu}$ decay. The likelihood of this happening can be estimated with Monte Carlo, but requires precise knowledge of the photon veto detector inefficiencies.

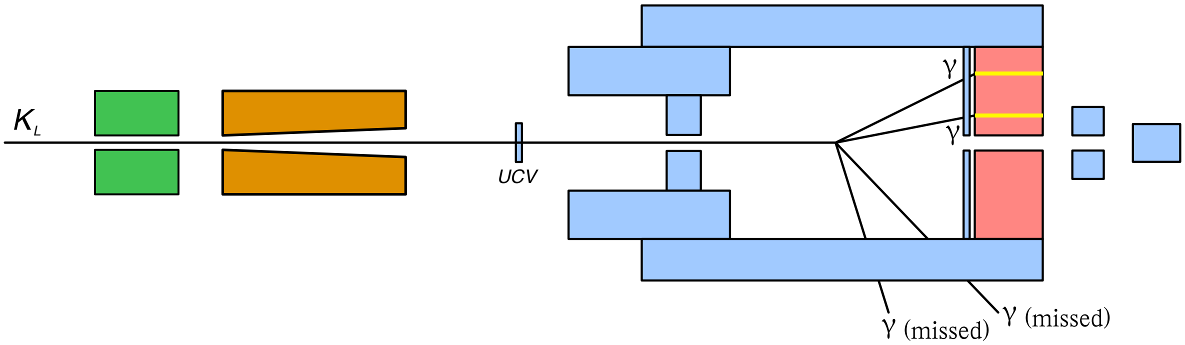


Figure 2.6: A $K_L \to 2\pi^0$ decay, in which two final-state photons are missed due to photon veto detector inefficiencies.

Inefficiencies of photon veto detectors are studied with data from the 5γ trigger to minimize the bias introduced from simulations. The energy and hit position of the five photons measured in the calorimeter is used to calculate the properties of the sixth one[2]. By looking at the signal from the veto detector that the sixth photon was expected to hit, the inefficiency of that veto detector can be calculated. A large amount of 5γ events is required to be collected during physics runs to perform this calculation with precision. In the 2021 data analysis (Table 2.1), the $K_L \to 2\pi^0$ background prediction was $0.059 \pm 0.022\,(\text{stat})^{+0.051}_{-0.060}\,(\text{syst})$ events[3]. The large systematic uncertainty of this estimation is also the largest contribution to the total uncertainty of the background estimation in the 2021 analysis, and it is mainly due to the uncertainty in the veto detector inefficiencies. Improving the precision of the $K_L \to \pi^0\nu\overline{\nu}$ background estimation in future KOTO analyses requires the collection of a large amount of 5γ events during physics runs, which was not possible with the bandwidth limitations of previous KOTO DAQ system. Part of the work presented in this thesis has been essential to overcome this limitation.

2.4 The KOTO Detector

The KOTO detector was introduced in section 1.3.1. A scaled schematic view of it is given in Fig. 2.7. In this chapter, the CsI calorimeter, as well as photon and charged veto detectors will be described in more detail.

[2] $K_L \to 3\pi^0$ decays are preferred over the $K_L \to 2\pi^0$ and other kaon decays for multiple reasons. First and mainly, because of the large $K_L \to 3\pi^0$ branching ratio, 19.5%, significantly larger than the $K_L \to 2\pi^0$ branching ratio of $8.6 \cdot 10^{-4}$. Second, because of the fact that two of the three pions can be fully reconstructed from the calorimeter hits, which allows us to confidently identify the kaon decay vertex, versus in the $K_L \to 2\pi^0$ decay where only one π^0 can be used for this purpose (see appendix J for details on the K_L decay vertex reconstruction). Third, the 5-cluster $K_L \to 3\pi^0$ is almost background-free, as the K_L has no other decay modes to five or more photons with significant branching ratio.

[3] An overview of this background estimation is given in appendix K

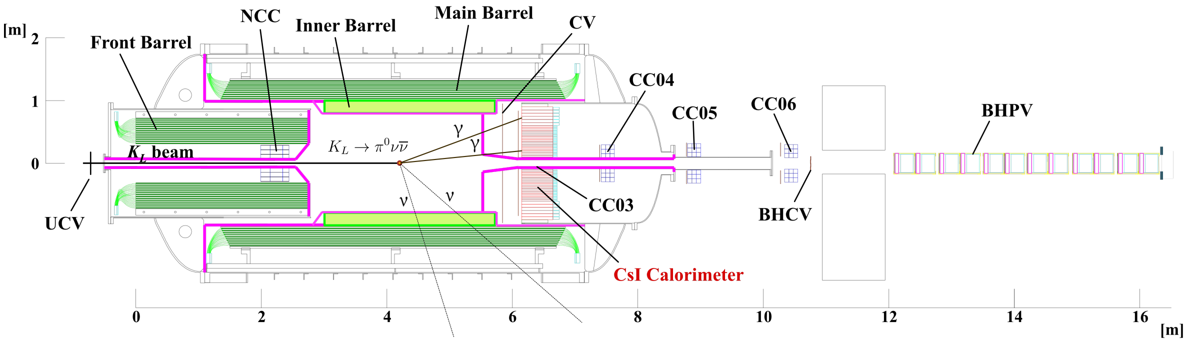


Figure 2.7: Schematic view of the KOTO detector. The kaon beam enters the detector from the left. A vacuum tank is built around the calorimeter and the barrel detectors, to separate the inner vacuum from the atmospheric pressure outside.

The KOTO subdetectors marked in Fig. 2.7 are described in the following sections.

2.4.1 Electromagnetic CsI calorimeter

The KOTO CsI calorimeter [14] is a circular detector with a diameter of 1.9 m, placed perpendicular to the beam, at the downstream side of the decay volume. It is composed of 2716 CsI crystals, each read out by a PMT (Photomultiplier Tube) placed at the back of each crystal. A schematic front view of the calorimeter is shown in Fig. 2.8. The inner squared section is composed of 2.5 × 2.5 cm crystals and the outer section of 5 × 5 cm crystals. A 20 × 20 cm hole in the center of the calorimeter allows the beam to pass through.

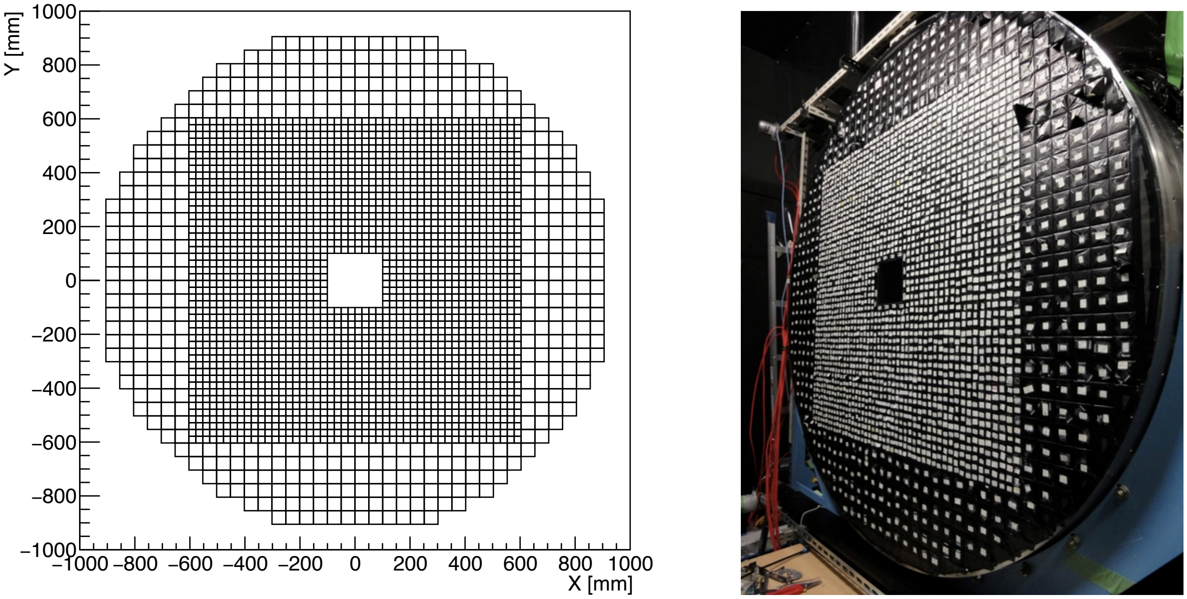


Figure 2.8: Left: Schematic front view of the CsI calorimeter. Right: Picture of the CsI calorimeter, quoted from [15].

All crystals are 50-cm long, equivalent to 27 radiation lengths. This is long enough to trap all the energy of electrons or photons hitting it. However, less interacting particles such as charged pions might pass through the calorimeter, depositing only a fraction of their energy in the CsI crystals. To help distinguish between photons, pions, and neutrons, a MPPCs (Multi Pixel Photon Counters) are attached to the front of the calorimeter. The scintillation light of shallow showers produced by photons will reach the MPPC layer noticeably faster than the PMTs. Particles that traverse the entire calorimeter will be detected by MPPCs and PMTs with a smaller delay. Particles like neutrons, scattering inside the calorimeter, can be detected by the MPPCs and PMTs in any order.

2.4.2 Photon veto detectors

Photon veto detectors are meant to *veto* an event if they register a photon hit. The photon veto detectors are required to have very low and well-known inefficiencies, as was justified in section 2.3. In particular, precise knowledge in the inefficiencies of the photon veto detectors is essential to precisely estimate the $K_L \to 2\pi^0$ background. The main veto detectors in the context of this thesis, from upstream to downstream, are listed below.

Front Barrel (FB) and Main Barrel (MB)

Both the FB and MB [16] are lead/scintillator sandwich-type photon detectors. Their thickness in radiation lengths is 16.5 and 14.0 respectively. The FB has a length of 2.75 m, and it is read out by wavelength shifting fibers (WLS fibers) connected to PMTs at its upstream end. The MB, with a length of 5.5 m, is also read-out by WLS fibers, but PMTs are connected at both of its ends due to its large length. The usage of WLS fibers is due to the small attenuation length[4] of the scintillators (0.45 m) compared to their length.

The hit position along the beam axis can be calculated in the MB from the timing difference between the signals at its front and rear PMTs, with a resolution of 3.8 cm[5]. In the plane perpendicular to the beam, the angular resolution is constrained by the number of modules, 16 in the FB and 32 in the MB.

Neutron Collar Counter (NCC)

The NCC [17] is a veto detector placed at the upstream side of KOTO, surrounded by the Front Barrel. The NCC shields the CsI calorimeter from beam neutrons and photons, and vetoes K_L decays that happen upstream of the decay volume. A schematic view of the NCC is shown in Fig. 2.9.

[4] The attenuation length is the distance at which the light intensity is reduced to $1/e$ of its original value. The attenuation length of the WLS fibers used in the FB and MB detectors is 4.5 m

[5] For a 30 MeV photon. The timing and position resolutions improve as the energy of the photon hit increases.

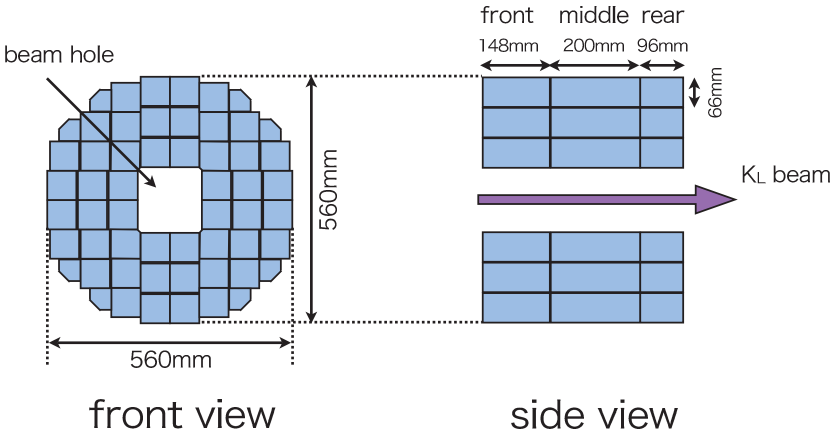


Figure 2.9: Schematic view of the NCC [17].

The NCC is composed of undoped CsI crystals divided in three sections along the beam direction. Each section is read out individually by wavelength shifting fibers, and more fibers collect the combined light yield of the three sections. The individual module information is mainly used offline for studies on the flux of beam-halo neutrons. The combined readout is used as event veto.

Inner Barrel (IB)

The IB [18] is a lead/scintillator sandwich-type photon detector, installed inside the MB in 2016 to increase the photon detection efficiency around the decay volume and to suppress the $K_L \to 2\pi^0$ background by a factor of 3. The IB is composed of 32 modules, each 2.8 m long, and read out by WLS fibers and PMTs at both ends. A picture of the IB is shown in Fig. 2.10.

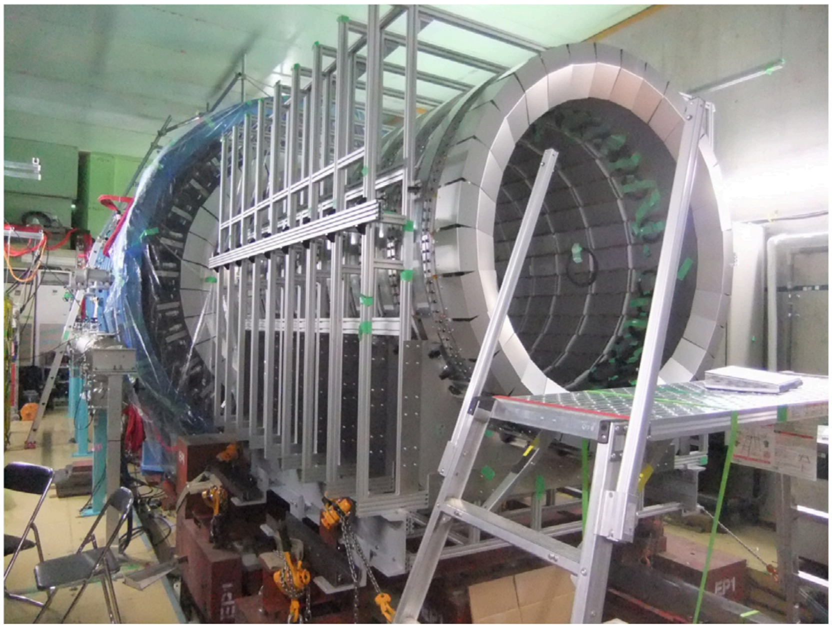


Figure 2.10: A picture of the Inner Barrel [18] during its installation, showing the arrangement of its 32 modules. The Main Barrel is also visible in the background.

Collar Counters (CC03, CC04, CC05, and CC06)

The Collar counters are photon veto detectors located around the beam pipe, arranged as shown in Fig. 1.3. They are composed of undoped CsI crystals readout by PMTs. The CC03 is placed around the CsI calorimeter beam hole, and aims to veto K_L decays near the CsI calorimeter.

The CC04, CC05, and CC06 counters, located downstream from the CsI calorimeter, are composed of undoped CsI crystals, and plastic scintillators on their upstream surface to help with the detection of charged particles.

Beam-Hole Photon Veto (BHPV) and Beam-Hole Charged Veto (BHCV)

The BHPV and BHCV are veto detectors placed in the beam, downstream from the CsI calorimeter. Their purpose is to detect charged particles and photons that might have escaped the decay volume through the CsI calorimeter's beam hole.

The BHPV [19] consists of 16 modules, each of them composed of a led layer to initiate a electromagnetic shower, and aerogel tiles. Light due to cherenkov radiation in the aerogel is guided by mirrors to PMTs. The BHCV [20] consists of multi-wire chambers where a gas is ionized by passing charged particles. the resulting electrons are collected by wires and amplified through an avalanche multiplication process. The BHCV is used to detect charged particles that might have escaped the calorimeter through the beam hole.

2.4.3 Charged veto detectors

The main Charged Veto (CV) detector [21] is a flat and thin plastic scintillation detector placed in front of the calorimeter. It is composed of two layers of 3 mm thick, 7 cm wide plastic scintillators, read out by wavelength-shifting fibers and MPPCs. Charged particles, such as pions or electrons, can be tagged by the CV before they enter the CsI calorimeter. A picture of the CV detector is shown in Fig. 2.11.

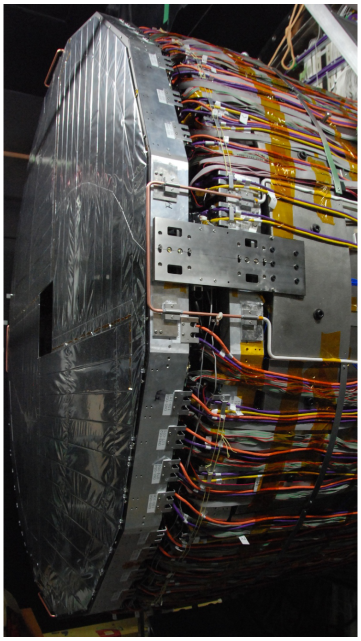


Figure 2.11: A picture of the Charged Veto detector placed in front of the CsI calorimeter.

The Charged Veto is used as a *veto* in triggers targeting decays without charged particles in the final state, such as the $K_L \to \pi^0 \nu \overline{\nu}$ or the 5γ triggers. However, in some cases, the Charged Veto can be used conversely, to tag kaon decays in which charged decay products are expected. This is the case of the $K^+ \to \pi^+ \pi^0$ or the $K_L \to \pi^0 e^+ e^-$ triggers.

Other charge detectors are placed both upstream and downstream from the KOTO barrel. The Upstream Charged Veto (UCV) [13], a thin in-beam charged particle detector, is used to tag K^+ that might enter the KOTO decay volume after failing to be swept away by the magnets. The BHCV (Beam-Hole Charged Veto) is used to detect charged particles that escaped the calorimeter through the beam hole.

Chapter 3

Data taking at KOTO

As has been motivated in the previous chapters, the offline analysis of the $K_L \to \pi^0 \nu \overline{\nu}$ decay requires the collection of multiple types of events together with the $K_L \to \pi^0 \nu \overline{\nu}$ candidates. Data needs to be taken to measure the veto inefficiencies, to calibrate detectors, or to measure the inefficiency of the online trigger system among others. In this chapter, we introduce the different types of data taken at KOTO during physics runs. Cosmic runs and two special runs needed for calibration purposes are briefly covered too.

Simultaneously targeting different kaon decays or physics processes, each of them with a different detector signature, is possible in KOTO's DAQ system during runs by defining and enabling multiple *triggers*. In this context, a trigger is defined as the set of conditions that an event must satisfy to be recorded. A *trigger menu* defines the triggers that are enabled during data collection. Each trigger is defined by a set of criteria imposed on the following quantities:

- the total energy deposition in the CsI calorimeter,
- the number of clusters in the CsI calorimeter,
- the set of veto detectors required to have no hits,
- the set of veto detectors required to have hits.

where a veto detector is said to have been hit if at least one of its channels records an energy deposition larger than a set threshold. Typically two thresholds are configured for each veto detector, a tight (low) threshold, and a loose (high) threshold. Loose thresholds are preferred by triggers that strictly require to minimize the impact of online trigger inefficiencies on the data (such as the $K_L \to \pi^0 \nu \overline{\nu}$ trigger), or by triggers aiming to study the veto detector inefficiencies themselves (such as the 5γ trigger).

Sometimes, loose trigger requirements can lead to a high trigger rate, which could limit or saturate the DAQ, or simply produce more data than what is needed. Prescale factors are naturally used in this situation to reduce the event rate.

The KOTO DAQ system can accommodate eight triggers in its trigger menu. All eight of them were eventually enabled for different purposes during physics runs in 2024. These triggers are described in the following section.

3.1 Physics runs

In this section, we briefly describe the different triggers that were enabled during the 2024 beam-time, taken with the DAQ system presented in this thesis. On top of the individual trigger requirements described below, all triggers require the total energy deposition in the CsI calorimeter to be larger than 500 MeV.

$K_L \to \pi^0 \nu \overline{\nu}$ trigger

Multiple photon vetoes are required to have no hits, to ensure no photon hits outside the calorimeter. The Charged Veto in front of the calorimeter is also required to have no hits, to ensure that the hits in the calorimeter do not come from charged particles. Two clusters are required in the calorimeter, expected from the decay $\pi^0 \to \gamma\gamma$. The veto detector configuration of the $K_L \to \pi^0 \nu \overline{\nu}$ trigger is summarized in Fig. 3.1.

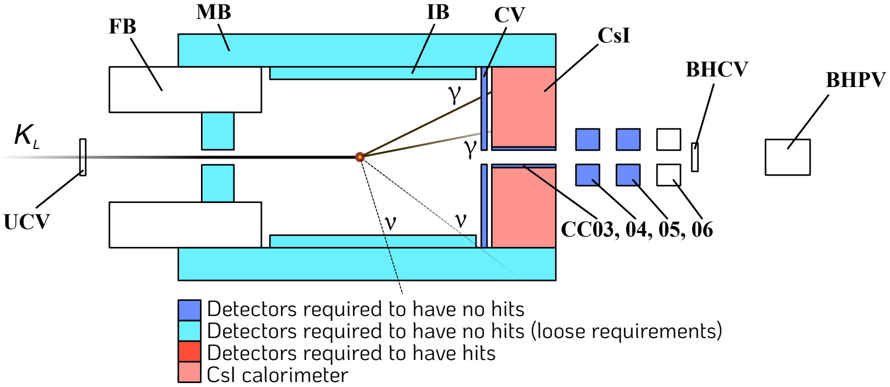

FB
MB
IB
CV
CsI
BHCV
BHPV
K_L
γ
γ
ν
ν
UCV
CC03, 04, 05, 06
🟦 Detectors required to have no hits
🟦 Detectors required to have no hits (loose requirements)
🟥 Detectors required to have hits
🟥 CsI calorimeter

Figure 3.1: Veto detector configuration of the $K_L \to \pi^0 \nu \overline{\nu}$ trigger.

Normalization trigger

The veto conditions are identical to the $K_L \to \pi^0 \nu \overline{\nu}$ trigger, but no cluster requirements are set. Because of this, the prescale factor is set to 30. This trigger mainly collects $K_L \to 3\pi^0$, $K_L \to 2\pi^0$, and $K_L \to \gamma\gamma$ decays. These decays, which have a high branching ratio compared to the $K_L \to \pi^0 \nu \overline{\nu}$ signal, are used to calculate the number of kaons entering the KOTO detector, which is needed to compute the $BR(K_L \to \pi^0 \nu \overline{\nu})$ [22]. Furthermore, this trigger is used to study the effect of the online clustering, which can be artificially emulated offline, on the $K_L \to \pi^0 \nu \overline{\nu}$ trigger. This online clustering algorithm will be later described in section 4.5.

Minimum bias trigger

No veto conditions or cluster requirements are set. This trigger is mainly used to monitor the effect of the online veto on the physics data. As many events satisfy this trigger, its prescale factor is set to 900. The veto detector configuration of the minimum bias trigger is summarized in Fig. 3.2.

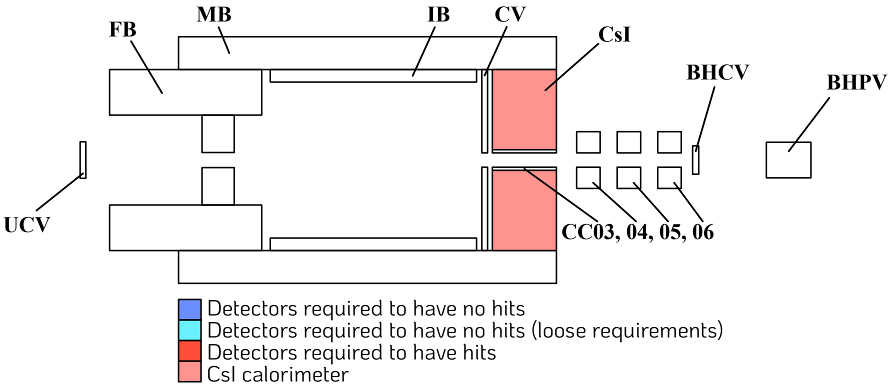

FB
MB
IB
CV
CsI
BHCV
BHPV
UCV
CC03, 04, 05, 06
🟦 Detectors required to have no hits
🟦 Detectors required to have no hits (loose requirements)
🟥 Detectors required to have hits
🟧 CsI calorimeter

Figure 3.2: Veto detector configuration of the minimum bias trigger.

Tight K^+ trigger

Three clusters are required, one for the π^+ and two for the photons coming from the π^0 decay. The Charged Veto is required to have hits, expected from the charged pion. The CC04, CC05 and CC06, the NCC, the MB and the IB are required to have no hits to ensure that the three calorimeter hits are everything that was produced from the kaon decay. The prescale factor is set to 1. This trigger is mainly used to measure the K^+ flux into the KOTO detector and calibrate Monte Carlo simulations accordingly. In 2024, this trigger was also used to measure the performance of the new permanent magnet. The veto detector configuration of the tight K^+ trigger is summarized in Fig. 3.3.

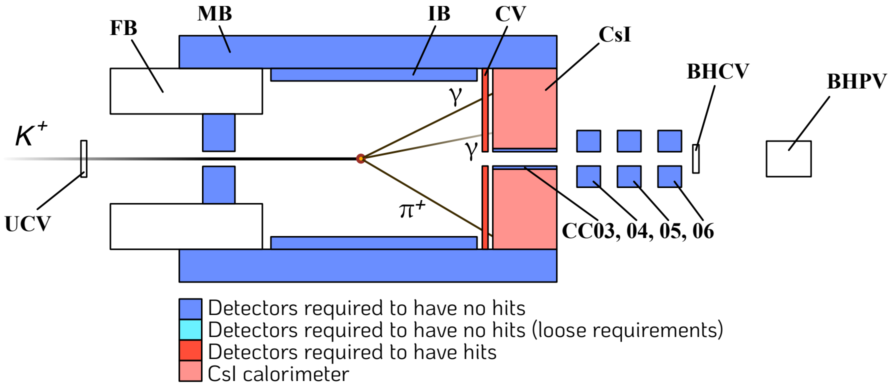

FB
MB
IB
CV
CsI
BHCV
BHPV
K+
γ
γ
π+
UCV
CC03, 04, 05, 06
🟦 Detectors required to have no hits
🟦 Detectors required to have no hits (loose requirements)
🟥 Detectors required to have hits
🟧 CsI calorimeter

Figure 3.3: Veto detector configuration of the tight K^+ trigger.

Loose K^+ trigger

The same requirements as its tight version, but excluding the downstream CC04, CC04 and CC06 detectors from the trigger evaluation. Events triggered by both the loose but not by the tight K^+ trigger are used to measure the efficiency of the excluded veto detectors. The prescale factor is set to 30. The veto detector configuration of the loose K^+ trigger is summarized in Fig. 3.4.

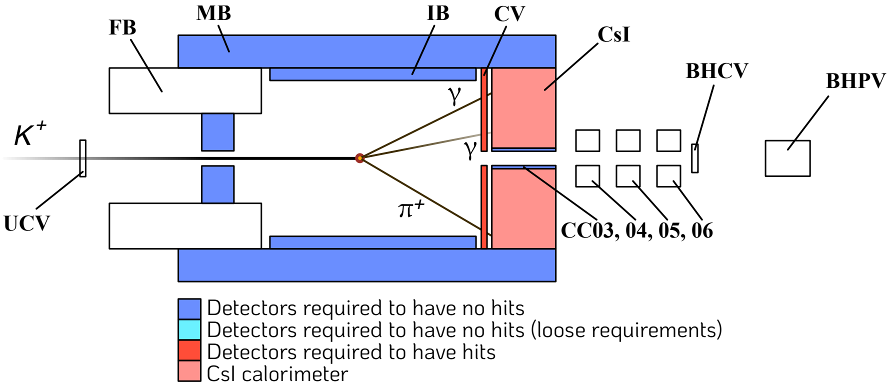

FB
MB
IB
CV
CsI
BHCV
BHPV
K^+
UCV
γ
γ
π^+
CC03, 04, 05, 06
■ Detectors required to have no hits
■ Detectors required to have no hits (loose requirements)
■ Detectors required to have hits
■ CsI calorimeter

Figure 3.4: Veto detector configuration of the loose K^+ trigger.

6γ trigger

The 6-cluster $K_L \to 3\pi^0$ trigger is mainly used for the offline energy calibration of the CsI channels, together with the data from Aluminum target runs which will be treated later. Online, six clusters are required and the prescale factor is set to 1. Offline, the three pions are reconstructed from the photons by making three photon pairs whose invariant mass is close to the nominal pion mass. Imposing then the actual pion mass allows to calibrate each calorimeter channel. Repeating the process throughout the beam-time allows us to monitor a possible drift of the calibration constants in some calorimeter channels. This trigger demands the veto detectors CV, CC04, CC05, CC06, NCC, MB and IB to have no hits. The veto detector configuration of the 6γ trigger is summarized in Fig. 3.5.

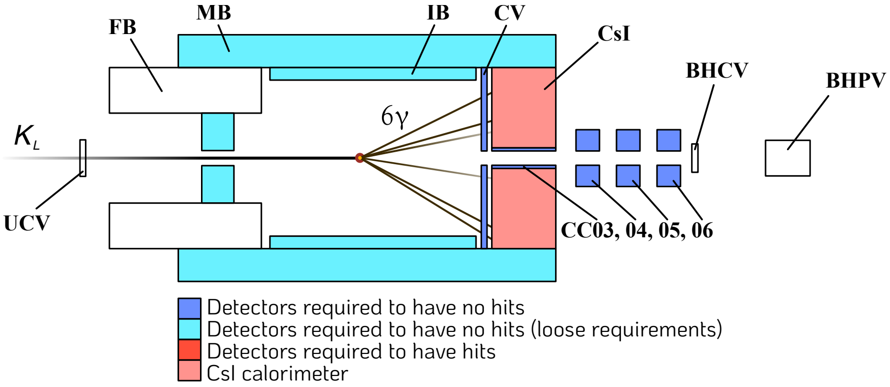

FB
MB
IB
CV
CsI
BHCV
BHPV
K_L
UCV
6γ
CC03, 04, 05, 06
■ Detectors required to have no hits
■ Detectors required to have no hits (loose requirements)
■ Detectors required to have hits
■ CsI calorimeter

Figure 3.5: Veto detector configuration of the 6γ trigger.

$K_L \to \pi^0 e^+ e^-$ trigger

The CC04, CC05 and CC06, as well as the NCC, MB and IB are required to have no hits. The CV in front of the CsI calorimeter is required to have hits, expected from the electrons. Four clusters are required in the calorimeter. The prescale factor is set to 1. The veto detector configuration of the $K_L \to \pi^0 e^+ e^-$ trigger is summarized in Fig. 3.6.

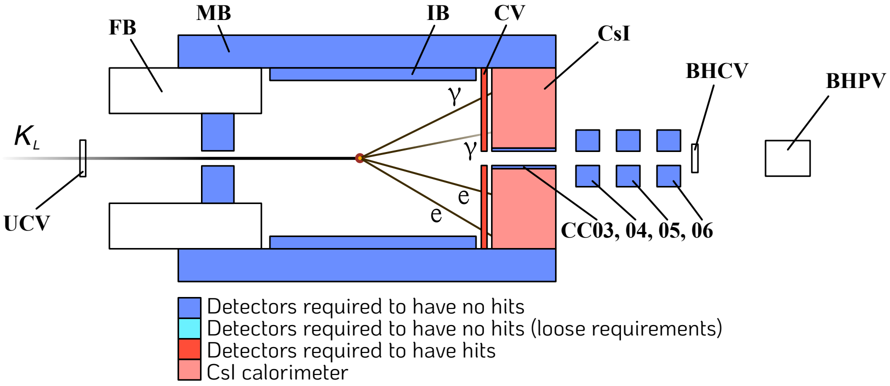

FB
MB
IB
CV
CsI
BHCV
BHPV
K_L
UCV
γ
γ
e
e
CC03, 04, 05, 06
Detectors required to have no hits
Detectors required to have no hits (loose requirements)
Detectors required to have hits
CsI calorimeter

Figure 3.6: Veto detector configuration of the $K_L \to \pi^0 e^+ e^-$ trigger.

5γ trigger

The 5-cluster $K_L \to 3\pi^0$ trigger is used to study the inefficiencies of the veto detectors hit by the sixth photon. The prescale factor is set to 1. The veto conditions of this trigger are the same as the 6γ trigger. The veto detector configuration of the 5γ trigger is summarized in Fig. 3.7.

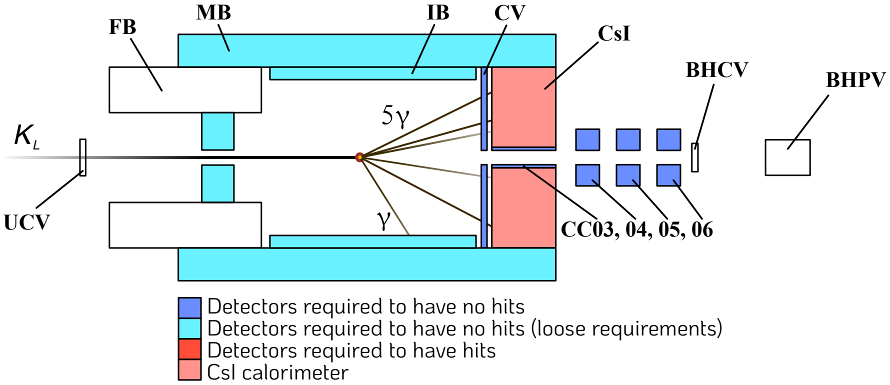

FB
MB
IB
CV
CsI
BHCV
BHPV
K_L
UCV
5γ
γ
CC03, 04, 05, 06
Detectors required to have no hits
Detectors required to have no hits (loose requirements)
Detectors required to have hits
CsI calorimeter

Figure 3.7: Veto detector configuration of the 5γ trigger.

The fraction of the total amount of events collected during a physics run taken by each of these triggers, together with their prescale factors are summarized in table 3.1.

Table 3.1: The eight triggers included in physics runs during the 2024 beam-time, and the fraction of events triggered by each. The total amount of triggers per spill in these runs was 18000.

Trigger	**Prescale factor**	**Fraction of events**
$K_L \to \pi^0 \nu\overline{\nu}$	1	8.5%
Normalization	30	3.1%
Minimum bias	900	3.1%
Tight K^+	1	33.7%
Loose K^+	30	5.2%
6γ	1	10.7%
$K_L \to \pi^0 e^+ e^-$	1	13.1%
5γ	1	23.6%

The veto requirements and number of clusters in the calorimeter are summarized for each trigger in table 3.2.

Table 3.2: The eight triggers included in physics runs during the 2024 beam-time, their required number of clusters in the CsI calorimeter, and the veto detectors required to have hits. All triggers include a minimum energy deposition requirement in the CsI calorimeter of 500 MeV.

Trigger	**clusters**	**Veto detectors required to have hits**
$K_L \to \pi^0 \nu\overline{\nu}$	2	not {CV, CC04, CC05, NCC, MB, IB}
Normalization	-	not {CV, CC04, CC05, NCC, MB, IB}
Minimum bias	-	-
Tight K^+	3	CV and not {CC04, CC05, CC06, NCC, MB, IB}
Loose K^+	3	CV and not {NCC, MB, IB}
6γ	6	not {CV, CC04, CC05, CC06, NCC, MB, IB}
$K_L \to \pi^0 e^+ e^-$	4	CV and not {CC04, CC05, CC06, NCC, MB, IB}
5γ	5	not {CV, CC04, CC05, CC06, NCC, MB, IB}

Other non-physics triggers are also collected during physics runs. These include the *clock trigger*, issued at a frequency of 10 Hz and mainly used to study the evolution of the pedestal level, and the *laser trigger*, which distributes through fibers a light pulse to all calorimeter channels and is used to verify the healthiness of the CsI channels. These triggers are not part of the trigger menu, and are referred to as *external triggers*.

3.2 Cosmic ray runs

Multiple cosmic ray runs are taken before beam-time, mainly used for energy and timing calibration of different detectors. Typically, any event leaving a significant energy deposition in the CsI calorimeter, the NCC, the Main Barrel or the CC04, CC05, and CC06 detectors will be recorded in cosmic ray runs. Cosmic runs are also taken during scheduled and unscheduled accelerator maintenance, and after beam-time, mainly to monitor the stability of the calorimeter's energy calibration constants.

3.3 Special runs

Different special runs are often taken for calibration purposes.

- *Muon runs.* The CC04, CC04, CC06 and BHPV have their channels stacked horizontally. Cosmic muons are very unlikely to hit multiple channels of these veto detectors, making their timing calibration not possible with cosmic ray data. To perform the calibration, the beam plug located between the two collimators in KOTO's beamline is closed during these special runs, allowing a large amount of muons to reach the KOTO detector parallel to the beamline. The CV detector, whose two layers are arranged perpendicular to the beam, is also calibrated with muon run events.
- Aluminum target runs, which are mainly used for the offline energy calibration of the CsI detector. During these runs, an aluminum target is placed in the beam, just upstream from the FB. This triggers a large production of π^0s, which immediately decay to two photons that can be measured at the CsI calorimeter. The known pion mass *and decay point* allow for a precise energy calibration of the CsI detector channels. Aluminum target runs are taken before physics runs after long shutdowns. The possible evolution of the energy calibration constants obtained from aluminum target runs is monitored during physics runs with the $K_L \to 3\pi^0$ trigger data.

More runs are regularly taken for other calibration purposes, or special physics analysis requiring detector configurations that are not compatible with normal physics runs.

The Trigger System, responsible for triggering the collection of events, and the DAQ System, responsible for recording the triggered events into permanent storage devices, are described in the next section.

Chapter 4

Data Acquisition and Trigger System of the KOTO Experiment

An introduction to data acquisition (DAQ) and trigger systems in particle physics experiments is given at the beginning of this chapter, followed by an overview of KOTO's entire DAQ and Trigger system. KOTO's first-level and second-level trigger systems are then described. The event building and data transmission up to the computer farm is also described. The High-Level Trigger will be described in chapter 5.

A more detailed description of KOTO's upstream DAQ system, as well as KOTO's first-level and second-level trigger (sections 4.3 to 4.6.3) can be found in the PhD thesis linked in Ref. [23].

4.1 Data acquisition and trigger systems in particle physics experiments

The fundamental purpose of data acquisition (DAQ) systems in particle physics experiments is to record detector output into permanent storage devices. The collected data can then be analyzed to produce physics results. Realtime data processing during the data acquisition is commonly referred to as *online* processing. Conversely, the word *offline* refers to the processing performed after the data has been recorded.

Online, constantly recording all detector output would result in an unmanageable amount of data. To make the data acquisition efficient, a *trigger* system is built with the DAQ, to trigger the recording of physics events and avoid recording unnecessary data. In high-rate experiments where even recording just *all* the physics events would result in too much data, fast reconstruction of physics quantities is done online, and based on them events of low physics interest are rejected.

Modern high-rate particle physics experiments, consisting of thousands of detector channels, generally divide their trigger system into multiple *levels*. The most upstream (close to the detector) level, usually referred to as Level 1 or *L1 trigger*, is normally a hardware-based trigger. It aims to perform a loose but fast selection based on physics quantities that can be quickly reconstructed from partial detector information. The L1 trigger is usually followed by a Level 2 (L2) and/or a HLT or *High-Level Trigger* (HLT). The High-Level Trigger is implemented in software on a computer farm, and performs relatively sophisticated event reconstruction based on full event information. The HLT is the last stage of the trigger system, and the events that pass it are recorded to permanent storage devices.

4.2 Overview of the KOTO Data acquisition and trigger system

The DAQ system of KOTO, as it was designed for the physics data taking in Summer 2024[1], is outlined in Fig. 4.1. The KOTO detector consists of different subdetectors, and a total of almost four thousand readout channels. At the first stage of the DAQ (leftmost in Fig. 4.1), waveforms from each detector channel are digitized by ADC (Analog to Digital Converter) modules. A total of 18 ADC crates are installed in KOTO, hosting 16 ADC modules each. Hereafter, the acronym ADC will be used to refer to the ADC modules. The Analog to Digital Converters inside the ADC modules will be referred to as *ADC chips*.

The $K_L \to \pi^0\nu\overline{\nu}$ search at KOTO heavily relies on the hermeticity of the KOTO detector, required to ensure that the two photons observed in the calorimeter are the only observable products of the $K_L \to \pi^0\nu\overline{\nu}$ decay. To ensure high detection efficiency, complete waveforms are recorded from all detector channels. This particularly helps to identify very low-energy hits that could have otherwise been missed, as well as to allow for the separation of hits recorded in the same time window.

The synchronized outputs from all ADCs are combined (event building block in Fig. 4.1), as will be described in section 4.7, and a complete event is sent to the HLT.

[1] The switch between OFC-II and the HLT shown in Fig. 4.1 was removed after the start of the beam-time. This modification is discussed in section 4.7.3.

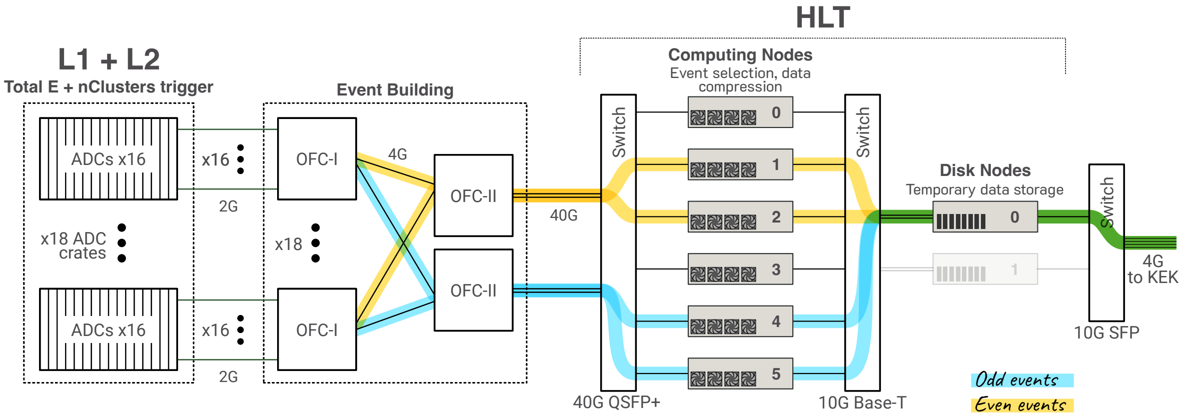


Figure 4.1: Overview of the KOTO DAQ system as designed for the physics data taking in Summer 2024. "G" stands for Gbps. Only the modules directly involved in the data transfer are shown. L1 and L2 trigger modules are not shown.

At the HLT, a layer of Computing Nodes (CN) receives the event data and performs event reconstruction and selection. Event data is then compressed and sent to a single Disk Node (DN) for temporary storage. Eventually, data is sent from the DN to the computing center at KEK (Tsukuba, Japan) for permanent storage. The link between J-PARC and KEK is shared with other experiments. On average, the bandwidth available for KOTO's HLT is 4 Gbps.

The ADC modules, which play an important role in both the DAQ and the trigger systems, are described in the next section. The evaluation of the L1 and L2 trigger decisions are then described from section 4.5. The Event building stage between the ADCs and the High Level Trigger is described in section 4.7. The High Level Trigger, the main topic of this thesis, is treated from chapter 5.

4.3 ADC modules

The KOTO DAQ system receives analog inputs from almost 3000 channels of the CsI calorimeter and from other channels in different veto detectors into ADC modules.

All the CsI channels and some VETO detectors[1] are read out by 16-channel, 14-bit, 125-MHz (8 ns sampling time) ADC modules [24]. Before digitalization, analog signals are filtered through a ten-pole Bessel filter. Through this transformation, detector signals with sharp rising edges are converted into Gaussian pulses. This provides higher resolution in the rising edge after digitalization, which enhances the timing measurement. A digitized waveform after the ten-pole Bessel filter is shown in Fig. 4.2.

[1] All veto detectors mentioned in this thesis except for the IB, BHCV and BHPV.

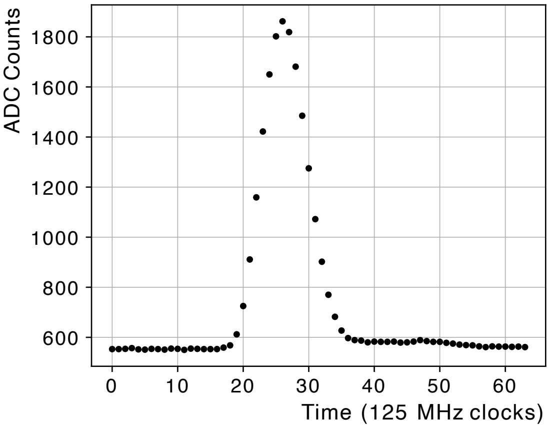


Figure 4.2: Example of a digitized waveform after the ten-pole Bessel filter.

The veto detectors with higher hit rate[2] are read out by 12-bit, 500 MHz (2 ns sampling time) ADC modules [25], which provide better time resolution and separation of overlapped pulses. The Bessel filter is not applied in these ADCs, to preserve the sharpness of the pulses and enhance the pulse separation.

The digitized signals in the ADC modules are temporarily stored in FPGAs (Field Programmable Gate Arrays). The FPGAs in the ADC modules have enough memory to hold incoming data for 5.2 μs. Within this time, features are extracted from the data stream, and the L1 and L2 trigger decisions are made. If an event is triggered, a snapshot with length 8 ns × 64 is recorded in all ADCs, and sent to the event building block of the DAQ system.

Organization of the ADC modules

The ADCs are organized in a total of eighteen ADC crates. Eleven of them are dedicated to the CsI calorimeter, and the remaining seven to the VETO detectors. Apart from the ADCs, each ADC crate contains a VME[3] computer used for monitor and remote control, and a Local CDT (Clock Distribution and Trigger) module. The Local CDT distributes the clock and trigger signals to all the ADCs in the crate. It also acts as the interface between the ADCs in the crate and the rest of the modules involved in the L1 and L2 trigger decisions.

Figure 4.3 illustrates a crate with 16 125-MHz ADCs. One ADC receives sixteen analog signals from the detectors through eight RJ45 connectors (red in Fig. 4.3). Two detector channel signals are transmitted per cable. Clock and trigger signals are received from the local CDT through RJ45 connectors (blue). Clock-by-clock calculation results are transmitted to the local CDT through fibers (yellow). Upon receiving a trigger signal, data snapshots are sent to the event building block through fibers (green).

[2] from the detectors mentioned in this thesis, these are the IB, BHCV, and BHPV.

[3] All DAQ and Trigger modules in KOTO are VME (Versa Module Eurocard) modules. All VME modules (including the VME computer) get power from the crate and are connected to each other through the crate's backplane.

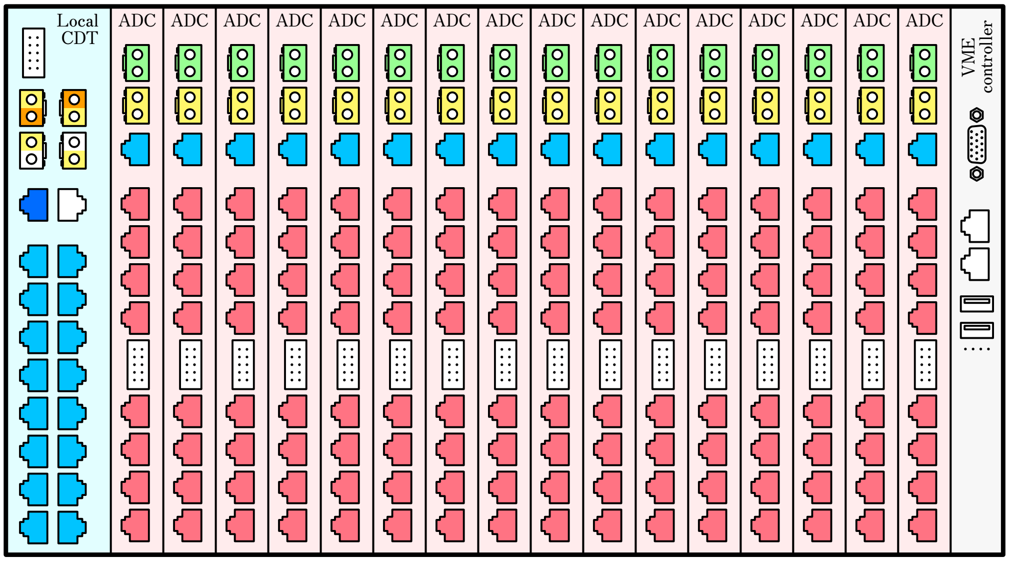


Figure 4.3: Illustration of a KOTO ADC crate, showing a VME controller (right), 16 125-MHz ADCs and a local CDT (left).

4.4 Clock distribution and trigger

All ADCs must be able to provide synchronized and on-time waveforms when a trigger is issued, for which the trigger signal must be distributed simultaneously to all ADCs. The clock signal must also be common to all DAQ modules for them to work synchronously.

These issues motivate the need for a single Top CDT (Clock Distribution and Trigger) module, that performs the following tasks:

- It generates a 125 MHz clock signal which is distributed to all KOTO DAQ and trigger modules, where it is used as a reference clock.
- It issues L1 and L2 triggers based on the results from calculations performed at dedicated L1 and L2 trigger modules. The L1 signal triggers the L2 evaluation, and the L2 signal triggers the data transfer from ADCs to the Event Building block.
- It receives a *LIVE* signal from the accelerator's monitor. The LIVE signal is high when the beam is on, and low when the beam is off. The LIVE signal is also distributed to all the DAQ modules.

Clock, LIVE, and trigger signals from Top CDT are distributed to the ADCs as shown in Fig. 4.4. The signals are first sent to a *fan-out CDT* module. The fan-out CDT further distributes them to each Local CDT in each crate. The Local CDT ultimately distributes the signals to all the ADC modules.

4.5 Overview of the L1 and L2 trigger architecture

The architecture of the L1 and L2 triggers in KOTO is shown in Fig. 4.4. The CDT modules (Top CDT, Fan-Out CDT, and Local CDT), as well as the ethernet cables distributing the clock, trigger, and LIVE signals, are painted blue. The ADC modules are painted red.

An *ET/Veto OFC* (Optical Fiber Center) module computes the total energy deposited in the CsI calorimeter and in the veto detectors, needed to make the L1 decision. A *Clustering OFC* module calculates the number of clusters in the CsI calorimeter needed for the L2 decision. The OFCs share their results with the Top CDT, where the trigger decisions are ultimately made. The process of making the L1 and L2 trigger decisions is detailed in section 4.6.

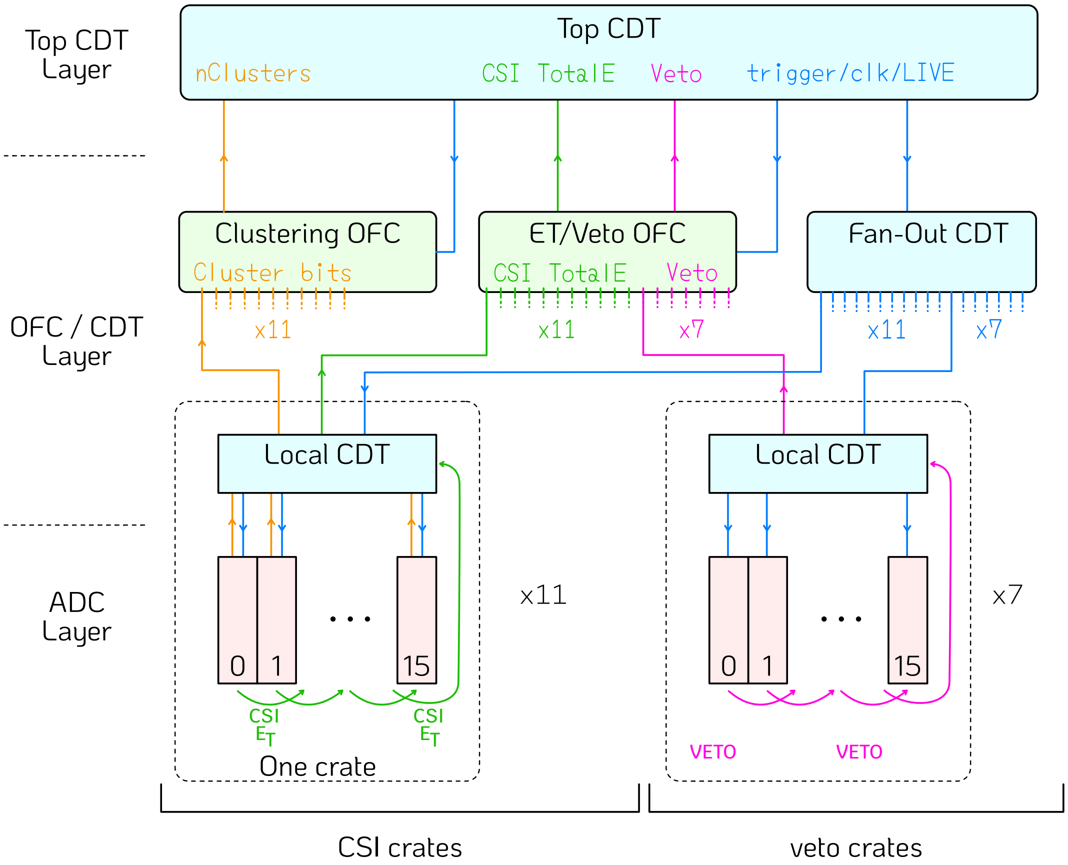


Figure 4.4: Architecture of the L1 and L2 triggers. Blue connections represent clock, trigger, and LIVE signals. Green lines transmit energy calculation results. Purple lines transmit veto results. Orange connections transmit clustering information.

4.6 Level 1 and Level 2 trigger decision

The L1 and L2 trigger evaluation process starts at the ADC modules, where the energy and timing of incoming pulses are calculated as will be described in section 4.6.1. Local ADC results are collected and combined by the local CDT as will be described in section 4.6.2. The ET/VETO OFC combines information from each crate's local CDT to compute the total energy deposited in the CsI calorimeter and the number of hits in each veto detector. The calculation results are sent to the Top CDT, where the L1 decision is made. In usual physics runs, all triggers are required to have a Total CSI Energy above 500 MeV. The Veto conditions differ trigger by trigger depending on the physics target.

The L1 evaluation process takes a total of 2.4 µs, from the 5.2 µs that data can be held at the ADCs. However, The L1 decision can be evaluated every clock so this latency does not translate into dead time. The L1 evaluation is described in section 4.6.2.

If the L1 passes, the Clustering OFC retrieves from the local CDT of each ADC crate the data it needs to compute the number of CsI clusters. This data transfer takes 20 clocks (or 160 ns) per trigger. During this time, incoming L1 triggers will be rejected. During physics runs in the 2024 beam-time, this dead time accounted for a 1.5% event loss.

Clusters are computed in the Clustering OFC, as will be described in section 4.6.3. The number of clusters is then used by the Top CDT to make the L2 decision. The L2 will pass if, for any trigger that passes the L1, the reconstructed number of clusters is equal to the configured requirement, e.g. equal to two in the $K_L \to \pi^0\nu\overline{\nu}$ trigger. If the L2 passes, waveforms from all ADCs are sent to the Event Building block, following the format described in section 4.6.4. In total, L1 and L2 evaluation takes 4.8 μs.

The bandwidth limitation at the ADC output is discussed at the end of this section. The event building stage will be covered in section 4.7.

4.6.1 Pedestal calculation at the ADCs

Both the L1 trigger decision and the clustering at the L2 make use of the energy calculation results from multiple detector channels. At the ADCs, the energy deposited on each channel is calculated every clock from the distance between the peak height and the pedestal, illustrated in Fig. 4.5. The pedestal needs to be known in advance.

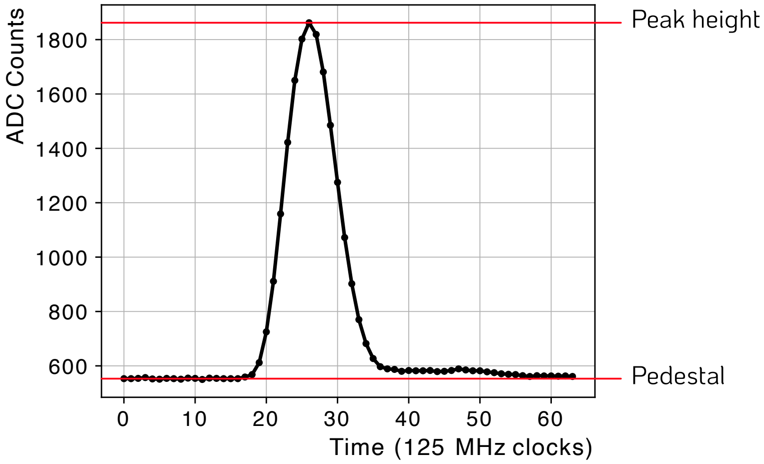


Figure 4.5: Illustration of the pedestal and peak height of a waveform.

The waveform pedestal is calculated at the ADCs before every spill. All samples are expected to be distributed around the pedestal level at that time. Eight thousand consecutive samples are recorded for this purpose on each ADC channel. The most frequent ADC-count value is then taken as the pedestal of the channel. This operation is illustrated in Fig. 4.6.

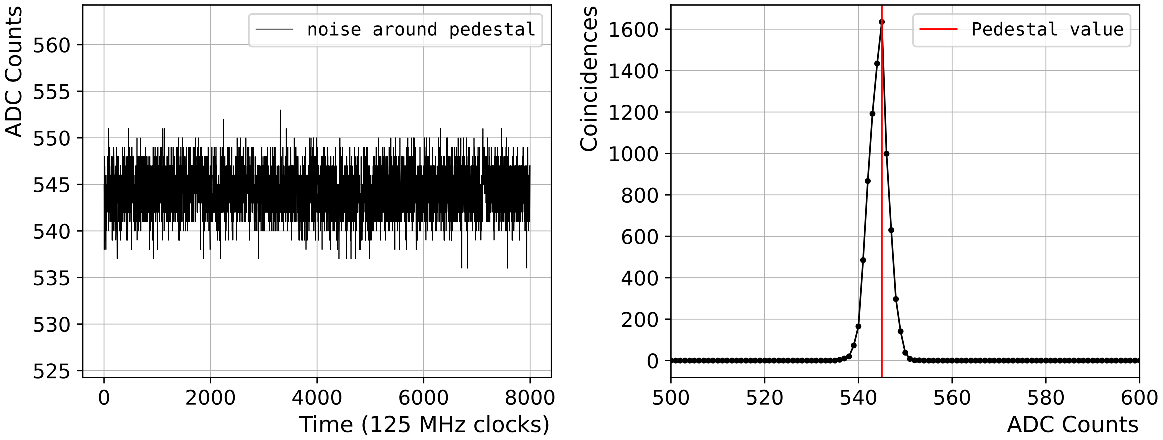


Figure 4.6: Pedestal calculation during beam off. Left: 8000 samples are taken consecutively on each channel. Right: Projection of the left plot on its vertical axis. The most frequent ADC count is taken as the pedestal.

The obtained pedestal value is used during the whole spill. To account for possible drifts during a run, it is re-calculated before every spill.

4.6.2 Level 1 trigger: CsI total energy and veto

The calculation of the total energy deposited in the CsI calorimeter starts at each ADC channel in the CsI crates. Each 8-ns clock, the potential presence of hits in the incoming data stream is evaluated at each ADC channel following the method illustrated in Fig. 4.7.

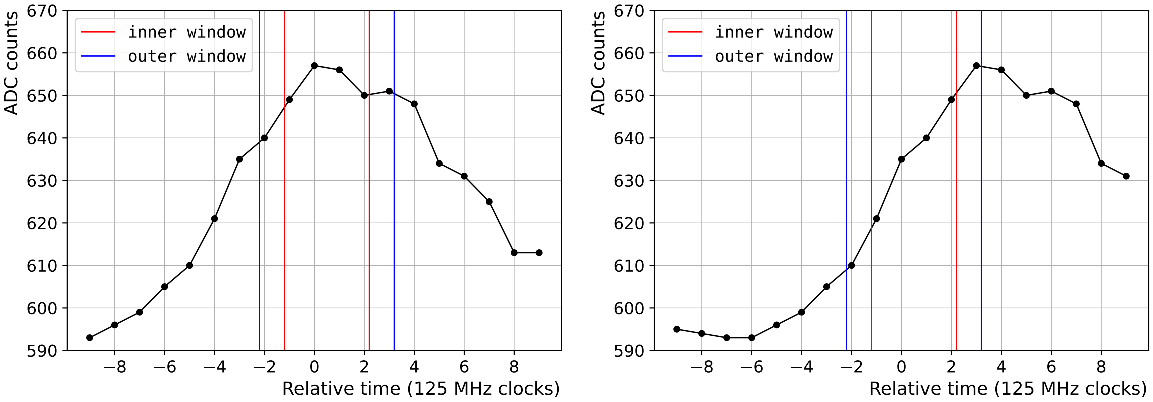


Figure 4.7: Left: The maximum sample within the outer (blue) window falls inside the inner (red) window, resulting in an on-time hit. Right: The same waveform a few clocks later. Now the maximum point within the outer window lies outside the inner window, thus no hits are observed.

If the peak within the outer window lies inside the inner window, there is a potential on-time hit. In this case, the energy is calculated from the peak height (maximum ADC count - pedestal). Peak height is linearly related to the energy, but the conversion factor varies from channel to channel, mostly due to differences in the gain of the PMTs. The conversion factors from ADC counts to energy used online[4] are estimated with cosmic muon events [26], that can be collected before beam-time. To reduce the contribution from accumulated noise, energy per channel is set to 0 if its calculation returns a value lower than 3 MeV.

At this point, the clock-by-clock energy deposition in individual ADC channels is known. The process of combining individual channel information to make the L1 trigger decision is described in the next section.

Total energy in the CsI calorimeter

The sixteen ADCs in one crate are grouped in four 4-ADC chains. All ADCs in a chain are connected in serial through optic fibers, as shown in Fig. 4.8. The first ADC computes the total energy recorded in its 16 channels and sends the result to the next ADC. The result is received, added to the local calculation, and sent to the next module in the chain. Ultimately, the local CDT of a CsI crate receives four inputs that are added up to form the total energy recorded in the crate.

[4] Offline, more precise conversion factors are estimated with pairs of photon clusters coming from π^0 decays in $K_L \to 3\pi^0$ events, and from π^0 produced at a fixed Aluminum target in special runs.

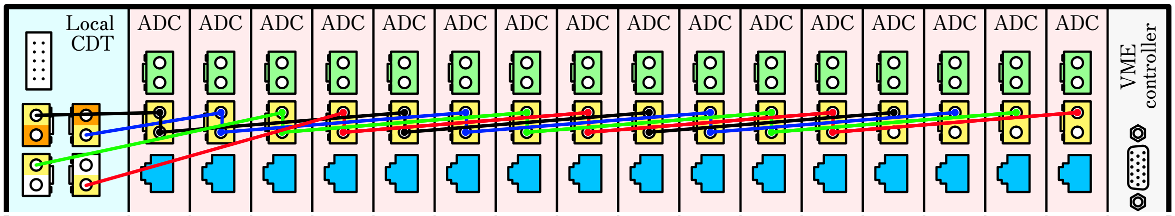


Figure 4.8: The four ADC chains, each of them connecting in series four ADCs each to the local CDT.

The energy computed at the Local CDT is sent clock by clock to the ET/Veto OFC module. ET/Veto OFC combines it with the inputs from other CsI crates to form the total E recorded in the CsI detector. The result is then sent to the Top CDT. The Top CDT records the received total energy clock by clock. Comparing each sample to the previous and the following ones, it finds peaks in the clock-by-clock total energy distribution. If a peak is found to be above 500 MeV, a L1 trigger is issued.

Total energy and number of hit channels in veto detectors

The second part of the L1 trigger decision is based on the deposited energy and number of hits in veto detectors. The calculation of these two quantities is covered in appendix D. A L1 trigger is issued if the total energy in the calorimeter and the veto information meet the requirements of any of the triggers in the trigger menu. In this case, the Top CDT issues a trigger signal back to the ADCs, and the L2 evaluation begins. The following section describes the clustering process at the L2 trigger.

4.6.3 Level-2 clustering trigger

If the L1 trigger is passed, the Top CDT issues a trigger signal back to the ADCs through the local CDTs, triggering them to send the *cluster bits* to the Clustering OFC. Cluster bits are set per ADC channel if the channel recorded a hit above 22 MeV (44 MeV) for small (large) CsI crystals. The Clustering OFC maps the cluster bits to a 38 × 38 grid, where each bin corresponds to the area of one big crystal (or 4 small crystals) on the CsI surface. Figure 4.9 shows an example of this process.

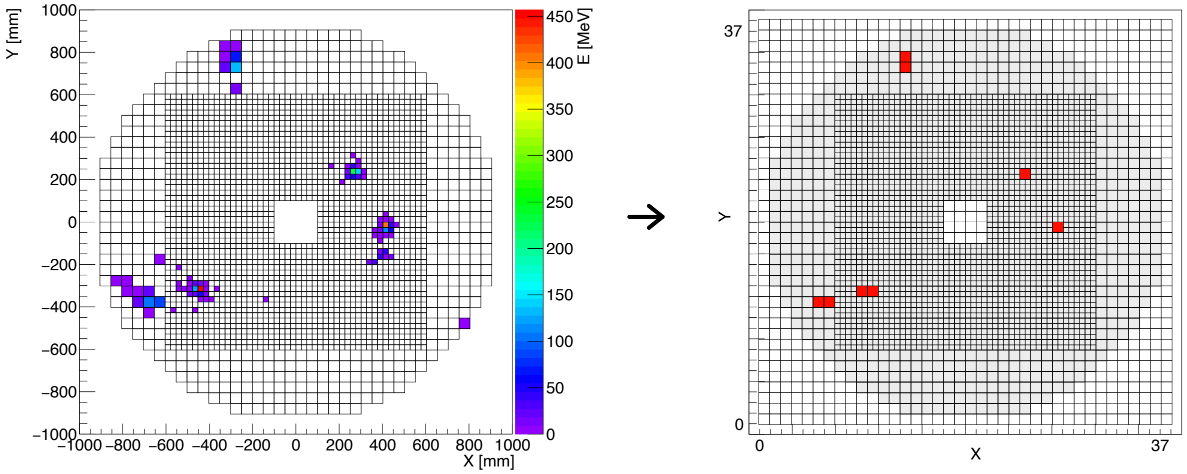


Figure 4.9: Display of a physics event on the CsI detector (left) and the corresponding cluster map (right).

The cluster-finding algorithm works based on the following. When tracing the contour of a cluster on the cluster map in a clockwise direction, if one adds up the number of turns to the right and subtracts the number of turns to the left, the net sum of all turns around any cluster always equals 4. The number of clusters can then be calculated as $N_{\text{clusters}} = (N_{\text{turns}})/4$.

To get the total number of turns, the cluster map is scanned in parallel in 2×2 bin squares. Possible patterns of these squares and their corresponding number of turns are shown in Fig. 4.10. The total number of turns N_{turns} is calculated by adding up the results in all squares.

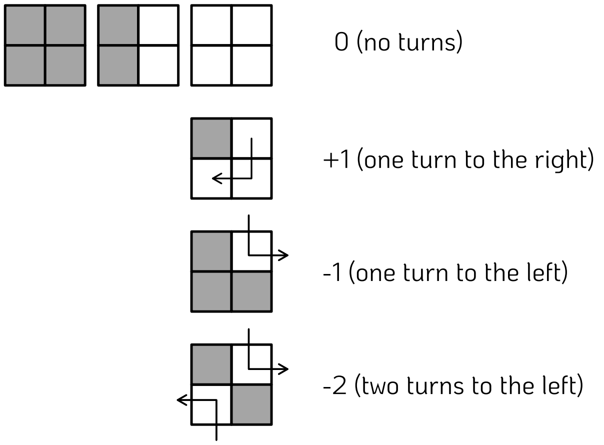
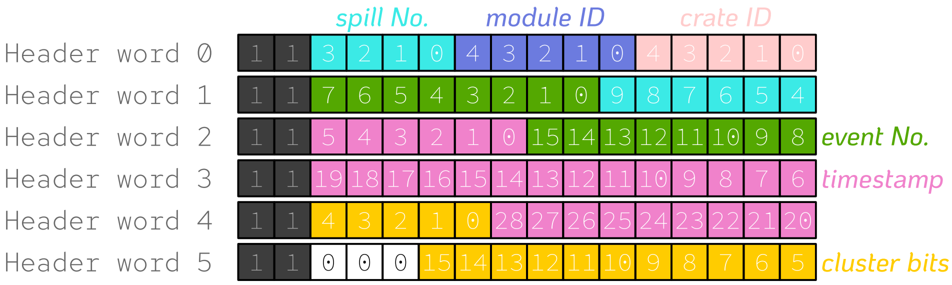

0 (no turns)
+1 (one turn to the right)
-1 (one turn to the left)
-2 (two turns to the left)

Figure 4.10: Possible patterns in a 2×2 bin square and their corresponding number of turns.

The number of clusters is sent to the Top CDT. If the requirements of any of the configured trigger modes are met at both the L1 (CsI Energy + Veto) and the L2 (Number of CsI clusters) stages, the L2 trigger is issued to all ADCs, and the waveform data is sent out to the event building stage of the DAQ. A header and a footer are attached to each ADC data. The structure of the ADC data is described in the following section. The event building stage is described in section 4.7.

4.6.4 ADC data format

Each ADC module sends all waveforms after a six 16-bit word header. The header contains information about the event and each ADC as shown in Fig. 4.11. The two most significant bits of the header words are set to a constant and used later as a data integrity check. Following the header, each waveform sample is stored into a 16-bit integer. Since the dynamic range of the ADCs is only 14 bits, the two most significant bits of each waveform sample are set to a constant value.



Header word 0	1	1	3	2	1	0	4	3	2	1	0	4	3	2	1	0
Header word 1	1	1	7	6	5	4	3	2	1	0	9	8	7	6	5	4
Header word 2	1	1	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Header word 3	1	1	19	18	17	16	15	14	13	12	11	10	9	8	7	6
Header word 4	1	1	4	3	2	1	0	28	27	26	25	24	23	22	21	20
Header word 5	1	1	0	0	0	15	14	13	12	11	10	9	8	7	6	5

Figure 4.11: ADC header format, containing the spill number, the ADC module ID within the crate, the crate ID, the event number, an event timestamp with respect to the beginning of the spill, and the cluster bits used to evaluate the L2 trigger. The two first bits of each header word are set to 11.

After the ADC header, the waveform data is stored in 16.4 kilobits[5]. Finally, a two 16-bit word footer is added to the end of each ADC data. The footer was not used in the 2024 DAQ, but is still kept. The ADC footer plans to be used in the future as a way to share real-time ADC monitoring data with the downstream DAQ.

In previous versions of KOTO's DAQ system, ADCs would perform pedestal suppression and lossless waveform compression before sending out the event data. This was done to transmit the highest possible amount of events through the old DAQ's limited bandwidth. Due to major upgrades in both the bandwidth between ADCs and HLT, and in the HLT computing capabilities, the ADC data reaches now the HLT uncompressed, and both pedestal suppression and lossless compression are performed at the HLT's GPUs.

Maximum bandwidth at the ADC output

The total data size that an ADC needs to send out every event is 16512 bits[6], or 1032 16-bit words. The ADCs operate with the 8 ns clock provided by Top CDT, and can send 16

[5] 16.4 kb = 16 [channels] × 64 [samples] × 16 [bits/sample] in the case of a 125 MHz ADC, or 4 [channels] × 256 [samples] × 16 [bits/sample] in the case of a 500 MHz ADC.

[6] 16512 bits = (6 [header words] + 2 [footer words]) · 16 [bits / word] + 16.4 [kb / ADC data]

bit every clock. One event can then be sent out every 1032 [clocks / event] · 8 [ns / clock] = 8256 ns. The maximum bandwidth at the ADCs output is then 1 [event / 8256 ns] = 121 kEvents/s. In terms of data size, this corresponds to 2 Gbps per ADC.

4.7 Event building stage: OFC-I and OFC-II

Event building is performed in KOTO's DAQ system before the data reaches the PC farm. Two layers of custom-made Optical Fiber Center (OFC) modules are set up for this purpose. The connections between OFC modules are shown in Fig. 4.12. Each OFC-I receives data from all ADCs in a crate and sends it out to the OFC-IIs. The OFC-Is are connected to each OFC-II through 6.4 Gbps fibers. This is to cope with the maximum data rate as their output, 4 Gbps per fiber as will be justified in section 4.7.1. OFC-II modules combine data from all the OFC-Is, and send complete events to the HLT nodes through two 40 Gbps fibers each. The maximum average data rate at the OFC-II output is 36 Gbps, as will be justified in section 4.7.2.

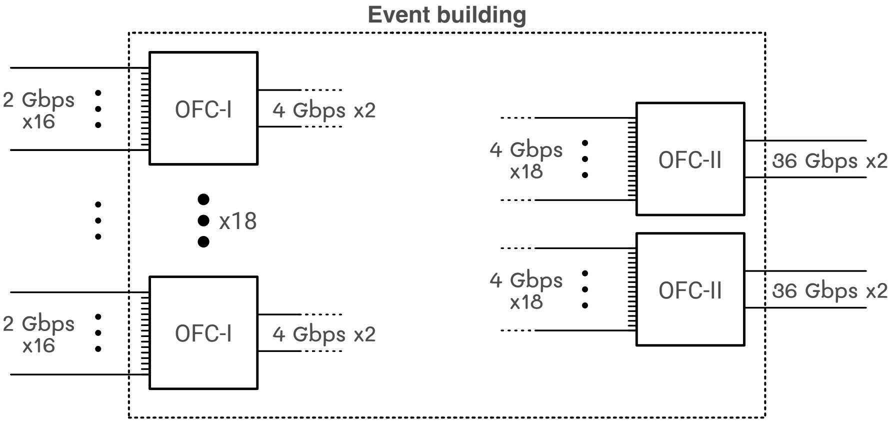


Figure 4.12: I/O limits of the OFC-I and OFC-II modules. The numbers shown correspond to the maximum achievable bandwidth at each stage, not the network bandwidth limit.

4.7.1 OFC-I

There are 18 OFC-I modules in KOTO's DAQ system, one per ADC crate. OFC-I modules collect data from all sixteen ADCs in a single crate and send it to two OFC-II modules through two fibers. The target OFC-II is common for all OFC-Is and it is switched event by event. ADC data is aligned at the OFC-I from ADC 0 to ADC 15.

The OFC-Is perform various checks before and during each spill and every event to ensure the data integrity. Before the spill, the healthiness of the optical links from the ADCs is checked by transmitting a counter between ADCs and OFC-I. The ADCs set and send out the counter, and the OFC-I verifies its contents. If this check fails, an error is reported to the Top CDT and data is not collected for the spill. During a spill, the following two checks are performed:

- *Data integrity*: The OFC-I checks the integrity of the data received from the ADCs by verifying the two most significant bits of the header words.
- *Data alignment*: The OFC-I checks the alignment of the data by ensuring that the data from all ADCs come within a fixed time after receiving the first ADC data.

If any of the checks above fail, the OFC-I reports an error to the Top CDT, and the DAQ stops until the next spill.

Finally, the OFC-I buffer with a capacity for 46 events can become temporarily full depending on trigger rates and the instantaneous beam rate. In this case, the OFC-I issues a *busy* signal to the Top CDT, which stops issuing triggers until the OFC-I buffers have been freed. Setting as a requirement an event loss $< 1\%$, the DAQ throughput gets limited to 50 kEvents per spill.

On top of the data checks, the OFC-I does two important modifications to data format, that considerably simplify the data handling at the HLT: *ADC data transposition* and *endianness swapping*. Both modifications are explained in the following paragraphs. They were proposed by the author of this thesis, and are motivated by the fact that in computer memory, data access is more efficient when the data is stored continuously in the memory banks.

ADC waveform data transposition

ADCs send data to the OFC-Is in the following sequence: The first sample of all 16 channels is sent first. Then the second sample of all 16 channels, and so on. This is illustrated in the top row of Fig. 4.13. Addressing a 64-sample waveform from this format would require 64 pointers. If all 64 samples of a complete waveform were stored together (bottom row in the figure), accessing them would require just the address of the first one.

Random (not contiguous) memory access, which is relatively slow on computer memory, does not introduce any delay on the OFC-I FPGAs. The transposition operation is then performed as illustrated in Fig. 4.13, while reading out and sending data to the OFC-II.

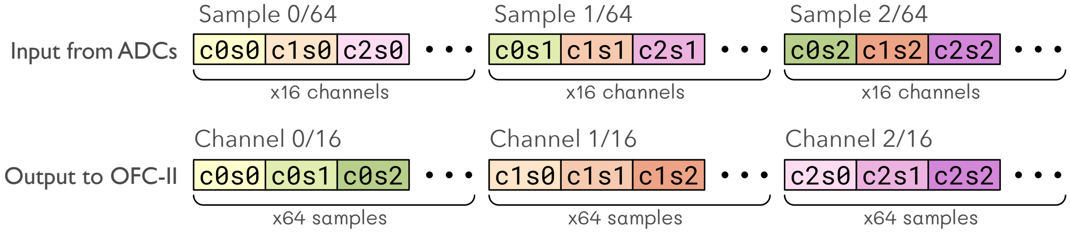


Figure 4.13: The OFC-I transposing the ADC waveform data. “c” stands for ADC channel (16 channels per ADC), and “s” for sample (64 samples per waveform).

Endianness swapping

Eventually, the HLT receives data from OFC-II. Both headers and waveform data are encoded into 16-bit words up to the OFC-II output, but for convenience, they are treated as 32-bit words as they are sent to the HLT.

The endianness of each 32-bit word is swapped by the firmware that controls the 40G output at the OFC-II, assuming that the HLT will read them also as 32-bit words. The HLT retrieving them as 16-bit words would lead to every pair of 16-bit words being swapped. This situation is illustrated in Fig. 4.14.

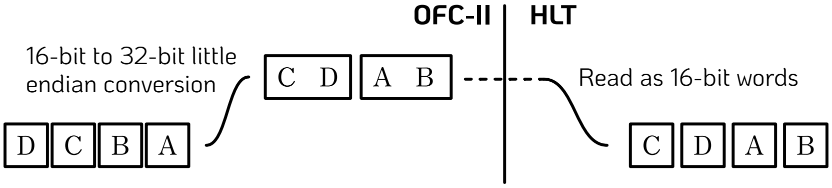


Figure 4.14: Endianness swapping during the 16-bit to 32-bit conversion at the OFC-II (left). The HLT reads those data back as 16-bit words (right).

For the HLT to be able to retrieve all 16-bit waveform samples in order, the endianness of each 32-bit word is once swapped at the OFC-I, and reverted to the intended order at the OFC-II output. Both the OFC-Is and the OFC-IIs can perform this operation without extra latency.

Bandwidth limitations at the OFC-I output

The data size each OFC-I has to process per event is 264.2 kilobits[7]. The OFC-I using the 125 MHz reference clock provided by Top CDT can send out 32 bits every clock through a single fiber. With this, the maximum theoretical output rate through one fiber at the OFC-I is one event per 8256 clocks. Equivalently, this is 15.1 kEvents per second, or 4 Gbps. The OFC-I can simultaneously send data through two fibers, making the maximum OFC-I to OFC-II bandwidth 30.3 kEvents/s, or 60.6 kEvents per spill. Note, however, that the maximum rate is set to 50 kEvents/spill to keep the event loss due to buffers full below 1%, as was discussed earlier in this section.

4.7.2 Two OFC-II system

Each OFC-II module has 18 fiber inputs that can provide data at a maximum of 4 Gbps each. The OFC-II waits for each event data from all 18 OFC-I modules to be received, and then sends out the entire event to the HLT. The 18 OFC-II inputs are aligned in memory, OFC-I by OFC-I, and therefore crate by crate. A 128-bit empty footer is added to each crate data to simplify the alignment of all OFC-I inputs.

There are three error signals that the OFC-IIs can issue:

[7] 264192 bits = 16 [ADCs / OFC-I] · 16512 [bits / ADC]

- *Busy error*: In principle, since the maximum output rate per OFC-II ($40\times2 = 80$ Gbps) is larger than the maximum input rate ($4\times18 = 72$ Gbps), the OFC-II buffers should never become full. However, buffers could fill up in exceptional cases if the 40G transmission block of the OFC-II firmware gets stuck for any reason, thus this error signal is kept.
- *Alignment error*: Similar to what the OFC-Is do with their 16 inputs, each OFC-II monitors its 18 inputs to ensure that the data from all OFC-Is come within a fixed time window.
- *Optical Link error*: A known data sequence is received from each of the 18 inputs and checked at the start of every spill, similar to what the OFC-Is do with their inputs.

Alignment and optical link errors are propagated to the Top CDT and stop the DAQ until the next spill.

4.7.3 Event packing and transmission at the OFC-II

The size of an event at the OFC-II output is 583.4 KiB. Each event data is fit into sixty-seven 8800-byte packets and one last 7840-byte packet. These packets follow the standard Ethernet format [27], which can be easily interpreted by both the HLT nodes and the switch between the OFC-II and the HLT.

The headers of these packets contain the destination MAC address of each packet. The destination MAC address is set packet by packet to the address of the corresponding target node. The target HLT node at the OFC-II output is switched event by event so that only two HLT nodes are targeted each spill by each OFC-II. The rotation scheme between OFC-I and the HLT nodes as originally commissioned in 2024 is summarized in Fig. 4.15.

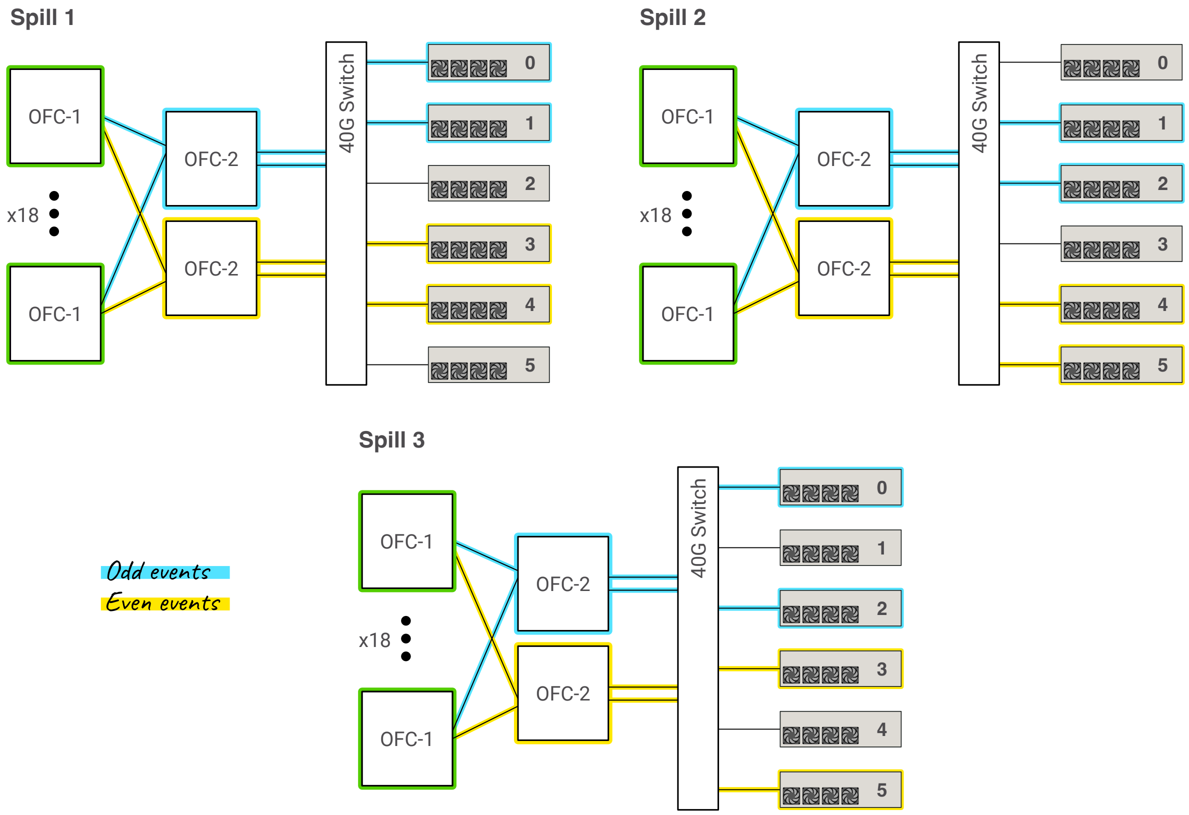


Figure 4.15: Rotation scheme between OFC-I and HLT nodes, as originally designed for the 2024 beam-time. All OFC-Is send odd events to one OFC-II and even events to the other OFC-II. Four HLT nodes are targeted per spill, two per OFC-II. Each HLT node receives two quarters of a spill every three spills. The cycle repeats every three spills, and spill 4 = spill 1.

There are two advantages of this scheme. First, note that there is no overlap between the targets of the two OFC-IIs. Although the maximum hardware-allowed bandwidth between the OFC-IIs and the HLT is $4 \times 40 = 160$ Gbps, the maximum rate at which a HLT node will ever need to receive data is 40 Gbps. Second and more importantly, complete events flow through a single OFC-II and a single HLT node. In case of a failure in a HLT node or on the network (a cable, a switch port, etc.), only a fraction of the events is lost. This represents an improvement with respect to the previous DAQ system, in which a failure in any of the 18 Type-1 nodes would result in the loss of all events.

OFC-II – HLT switch in 2024

The DAQ system was initially commissioned in 2024 as shown in Fig. 4.1, following the HLT node rotation scheme outlined in Fig. 4.15. However, intermittent interface resets leading to a large packet loss were observed at the switch between OFC-II and the HLT in physics runs at the beginning of 2024 beam-time. This issue could not be reproduced with cosmic ray or high-rate dummy data generated at the OFC-II, and due to the limited time available for testing during beam-time, it was decided to bypass the switch and connect the OFC-IIs directly to the HLT nodes. Under this configuration, the rotation scheme between OFC-II and HLT nodes was changed to the one shown in Fig. 4.16.

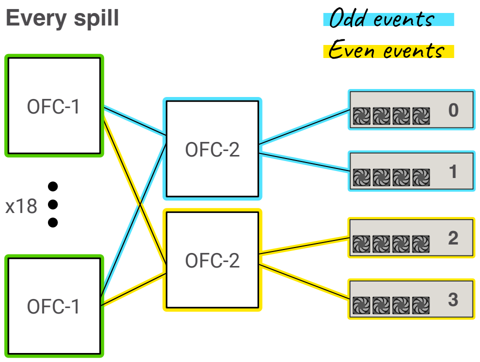


Figure 4.16: Rotation scheme between OFC-II and HLT nodes after bypassing the switch and connecting each of the two OFC-II outputs to the HLT nodes. Every Computing Node receives a quarter of a spill every spill.

This modification reduced the number of available Computing Nodes from 6 to 4, increasing the load per node by a factor of 1.5. The HLT software could cope with this increase without issues, as will be presented in sections 11.2 and 12.3. The data capture requirements at the HLT Computing Nodes were not affected by this change, as the maximum rate each node could expect remains 36 Gbps. Other data transfers and write/read operations at the Disk Nodes were not affected by this change either. The Data Acquisition ran stably without the switch during the rest of the beam-time, as will be presented in section 12.3.

The HLT software is described in detail from the next chapter. Performance results are given in section 11.

Chapter 5

Introduction to the High Level Trigger of the KOTO Experiment

The High Level Trigger (HLT) is the last stage of the data acquisition system in KOTO. Complete events are received by each HLT node. Energy and timing of the CsI channels are calculated, clustering is performed, and a final event selection takes place. Pedestal suppression and waveform compression are performed, and data is written to hard drives. By performing real-time event selection and compression, and thus maximizing the data that is available offline for physics analyses, the HLT plays a crucial role in maximizing the physics reach of the KOTO experiment.

Most of the processing at the HLT is done on GPUs, exploiting the massive parallelism that they offer. An introduction to GPU computing and its applications in high energy physics is given in section 5.3 to motivate their usage in KOTO's HLT. The HLT software is then explained stage by stage from chapter 6.

The efficiency and performance of the HLT are evaluated in chapters 10 and 11 respectively. Results are then summarized in chapter 12. Other minor results are given in this chapter, following some of the sections that describe each stage of the High Level Trigger.

5.1 Requirements of the new High Level Trigger

The upgraded data acquisition system of the KOTO experiment up to the HLT has been described so far. To fit in this new DAQ system, the new HLT needs to satisfy requirements in terms of throughput and data reduction. Furthermore, when the data reduction involves rejecting partial or complete events, the data reduction methods are also required to be efficient.

The HLT requirements were initially set assuming that six Computing Nodes would be used to process the OFC-II data. However, removing the switch between OFC-II and the HLT reduced the number of Computing Nodes to four at the very beginning of the 2024 beam-time, leading to the DAQ layout shown in Fig. 5.1. To avoid confusion in this section, we will redefine the requirements considering an HLT running on only four Computing Nodes.

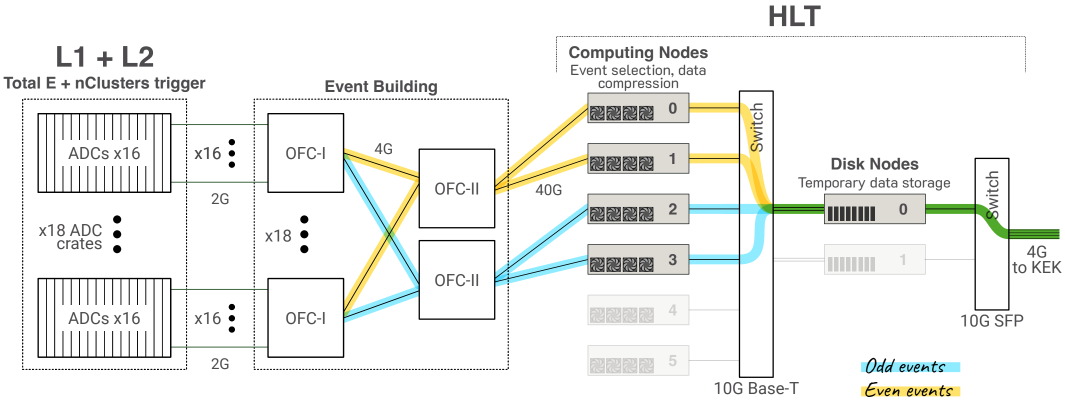


Figure 5.1: Baseline DAQ system of the KOTO experiment during the 2024 beam-time. The OFC-IIs are connected to the HLT nodes through four 40 Gbps links.

A number of Computing Nodes equal to four will be also taken as the baseline in the rest of the chapters of this thesis unless explicitly stated otherwise.

5.1.1 Data capture requirements

In 2024, physics data was taken with an 80 kW beam, and with the trigger menu described in section 3.1. The average trigger rate in these physics runs was 18 kEvents/spill. Knowing the size of a raw event (583.4 KiB at the OFC-II output) and the length of a spill (2 s), 18 kEvents/spill translates to 43.0 Gbps, which split among four HLT nodes yields 10.7 Gbps per node. As a minimum requirement, each HLT node would need to capture data at this rate to avoid the loss of events during the 2024 runs. With a 100 kW beam, and keeping the trigger menu the same, this requirement would scale to 13.4 Gbps per node.

However, the trigger rate during physics runs cannot be predicted with precision before beam-time, as it is sensitive to trigger parameters that are commonly modified during beam-time[1]. More realistically, one could require the HLT to be able to capture data at the maximum rate the current DAQ system can handle, which is limited at the OFC-I output to 50 kEvents/spill. At the HLT, 50 kEvents/spill (being one spill 2 s) corresponds to 118 Gbps. Distributed among four Computing Nodes, each Computing Node would need to be able to capture data at least 30 Gbps to avoid becoming a bottleneck in the DAQ system.

The requirement for the HLT nodes was set to the following; **The HLT nodes will be able to capture data at up to 40 Gbps with a packet loss below 0.01%**. The requirement was set to 40 Gbps to ensure that the data capture stage at the HLT does

[1] These include: The addition of new triggers to the trigger menu, the modification of trigger criteria for any of the enabled triggers, the modification of the energy thresholds for a veto detector, or the modification any detector's energy calibration constants among others.

not become a bottleneck in the DAQ system despite the beam power or the upstream DAQ throughput. The packet loss was set to 0.01% to make the corresponding event loss[2] small when compared to the unavoidable 1.5% loss due to dead time at the L2 clustering trigger, discussed in section 4.6. The 40 Gbps packet capture paradigm will be covered in chapter 6.

5.1.2 Throughput requirements

The stages of the High Level Trigger will be discussed in the following chapters. Together with packet capture, event reconstruction, pedestal suppression, and waveform compression are performed by the new HLT. The buffers where the HLT stores the captured data are large enough to accommodate more than two spills worth of data[3]. Still, for them not to saturate, the average HLT throughput needs to be larger than one spill per 4.2 s spill cycle.

Assuming four Computing Nodes working in parallel, each node needs to process on average one-fourth of a spill every 4.2 s. At the 2024 rate of 18 kEvents per spill, this means 1.1 kEvents per second per node[4]. At the maximum OFC-I rate of 50 kEvents per spill, this scales to 3.0 kEvents per second per node.

The requirement for the HLT nodes was set as follows: **The HLT nodes would be able to process events at a rate faster than the DAQ system limit of 50 kEvents/spill**. This is 3 kEvents per second per node.

5.1.3 Efficiency and data reduction requirements

Data reduction is the main purpose of any high level trigger. In KOTO, reducing the data rate is motivated by the limited bandwidth between J-PARC and the KEK computing center, and by the need to keep low the resources needed at KEK for the permanent storage of the data. The minimum requirement of the HLT is **to keep the output data rate below 4 Gbps**, for the J-PARC to KEK link not to become a bottleneck.

In KOTO's HLT, data reduction comes from event selection, pedestal suppression, and waveform compression.

Event selection

The definition of the efficiency of the HLT event selection will be given in section 10.1. In simple terms, 100% efficiency is achieved when the HLT does not reject any events that would have been accepted by the offline reconstruction. If the HLT accidentally rejects a good event,

[2] In the worst-case scenario, a 0.01% packet loss would yield to a 0.68% event loss. The correspondence between packet loss and event loss will be discussed in section 11.1

[3] In 2024, each Computing Node had the capacity to hold 47 kEvents. With four nodes, the HLT could hold more than ten 18 kEvent spills.

[4] 1.1 [kEvents/s/node] = 18 [kEvents/spill cycle] / 4.2 [s/spill cycle] / 4 [nodes]

that event is lost, lowering the statistics available for offline analysis. If the efficiency of the HLT is low enough, the HLT could introduce a bias in the data offline, and the effect of this bias would need to be accounted for in the offline analysis.

Realistically, any event selection at the HLT will introduce inefficiencies due to slightly different calibration constants and reconstruction procedures online and offline. In the 2024 runs, the author of this thesis aimed for the event selection to be roughly 99% efficient. More importantly, events contributing to the inefficiency would be studied to ensure they do not significantly bias the data. The estimation and measurement of the event selection efficiency at the HLT will be discussed in detail in section 10.2.

Pedestal suppression and waveform compression

Pedestal suppression consists of removing the waveforms from the CsI calorimeter channels that do not contain actual hits. The inefficiency of the HLT pedestal suppression is defined as the fraction of channels that are not suppressed (i.e., considered to have hits) offline, but are suppressed at the HLT. If the pedestal suppression is inefficient, channels with actual low-energy hits might be suppressed, potentially affecting the offline physics analysis. The pedestal suppression inefficiency would be kept below the intrinsic low-energy photon detection inefficiency of the CsI crystals. The photon detection inefficiency when $E_\gamma < 3$ MeV was estimated in the order of 10% in the original KOTO proposal [28]. **The pedestal suppression inefficiency at the HLT would be kept below 0.1%**, significantly lower than this value. The pedestal suppression at the HLT will be covered in section 8.1.

The waveform compression is lossless, and it is applied to all events of all triggers at the HLT. No requirements need to be set specifically for the waveform compression, as no information is lost from it. The waveform compression at the HLT will be covered in section 8.4.

5.2 Towards a new GPU-based High Level Trigger in KOTO

The limitations of the previous KOTO DAQ system have been outlined in section 1.5. Under these constraints, attempts at event reconstruction and selection were made at the CPU farm, but the CPU resources needed just to prepare the data for reconstruction did not leave enough room for the reconstruction itself, and event selection could not be performed. The risks and limitations of the old computing farm eventually led to the decision to completely renovate it during J-PARC's long shutdown in 2022 – 2023.

The 18 Gbps bottleneck at the front-end, between the Assembly boards and the computer farm, was solved by the introduction of the two OFC-II modules that would be connected to the HLT through four 40 Gbps links, increasing the maximum bandwidth of that link to 160 Gbps. Event building would also be performed at the OFC-II boards so that each node in the new computing farm could receive complete events and work on them independently from the other nodes. Failures affecting a single node would now cause only a quarter of the events to be lost.

A minimum of four Computing Nodes are needed at the new HLT to process all the data, facilitating maintenance with respect to the 48 nodes used in the previous computing farm. The HLT does not impose any further constraint on the number of nodes, not from the design nor from the performance perspectives. As was outlined in Sec. 4.7.3, this flexibility proved to be critically important during the 2024 data taking, keeping the DAQ system running smoothly after the switch failure that reduced the number of usable nodes from six to four.

The drastic increase in DAQ bandwidth coming from the addition of the OFC-II modules allows the HLT to receive data that is uncompressed and already transposed at the OFC-I (section 4.7), and can therefore be directly reconstructed. Pedestal suppression, which used to be performed at the ADCs, would also be offloaded now to the HLT, where it can be performed with a higher degree of precision.

Finally, A GPU was added to all Computing Nodes in the new HLT, pushing their computing capabilities beyond any bottleneck in the upstream DAQ hardware. The whole event reconstruction and selection would run on GPU. The CPU resources would be reserved for 40 Gbps data capture, data transfers, and monitoring. Leveraging GPU computing and data capture at 40 Gbps in the KOTO HLT would be proven essential to meet the physics goals of the experiment in the 2024 beam-time, as well as to show the potential of the HLT to cope with higher data rates and physics requirements in the future.

The complete data processing at the new HLT, including event reconstruction, selection, and compression is covered in this chapter. Results are given in chapter 12. The performance and future potential of the new HLT are discussed in section 13.2.

The next section gives an overview of GPU computing in high energy physics. An introduction to the GPU programming paradigm, focusing on the concepts needed to understand the chapters that follow, will be given in section 5.3.1.

5.3 Heterogeneous Computing in High Energy Physics

For many years, CPUs have been the only processor used in high performance computing. Up to the early 2000s, the increase in computing performance was mainly due to progressively smaller transistors and faster clock speeds. As the physical size of transistors reached a limit, single CPU performance started reaching a plateau, and focus was shifted towards interconnecting multiple CPU servers and developing new parallel computing techniques. Costs in maintenance, cooling, and power consumption of large CPU centers eventually motivated the need for new specific *accelerators*, processors that could perform specific tasks with much higher efficiency than CPUs. It was the birth of the heterogeneous computing era.

Heterogeneous computing refers to the usage of one or multiple accelerators together with CPUs in a single computer system, where each accelerator performs the tasks it is best suited for. The most widely used accelerator in high energy physics is the GPU (Graphics Processing Unit). GPUs have lower power consumption and associated cooling costs than CPUs while delivering higher performance in processing large datasets in parallel. Other well-known accelerators are FPGAs (Field Programmable Gate Arrays) and ASICs (Application Specific Integrated Circuits).

The following sections in this chapter are devoted to motivating GPU computing in KOTO. GPU and CPU architectures are outlined in the next section, highlighting their main differences. The GPU programming model is introduced in section 5.3.2, including concepts that will be referred to later in the chapter.

5.3.1 Outline of the CPU and GPU architectures

CPUs are versatile processors that can deliver high performance in processing complex tasks. Tasks performed by a CPU include running the Operating System and coordinating the usage of different hardware components, such as network cards or storage devices. The architecture of the CPUs in KOTO's HLT nodes is outlined in Fig. 5.2 (left). Each node has a total of 40 CPU threads, grouped in cores (2 threads per core) and sockets (10 cores per socket). CPU cores are relatively complex and physically large, which limits the number of CPU cores per socket and makes them relatively power-consuming. All 40 CPU threads can run complex tasks in parallel and independently of each other.

The CPU has access to the RAM (Random Access Memory) through the PCIe card. The CPU-RAM bandwidth is typically not larger than 100 Gbps through a PCIe 3.0 board, as it is installed in KOTO's HLT. RAM capacity can be scaled up to hundreds of GBs by simply adding more RAM cards to the server.

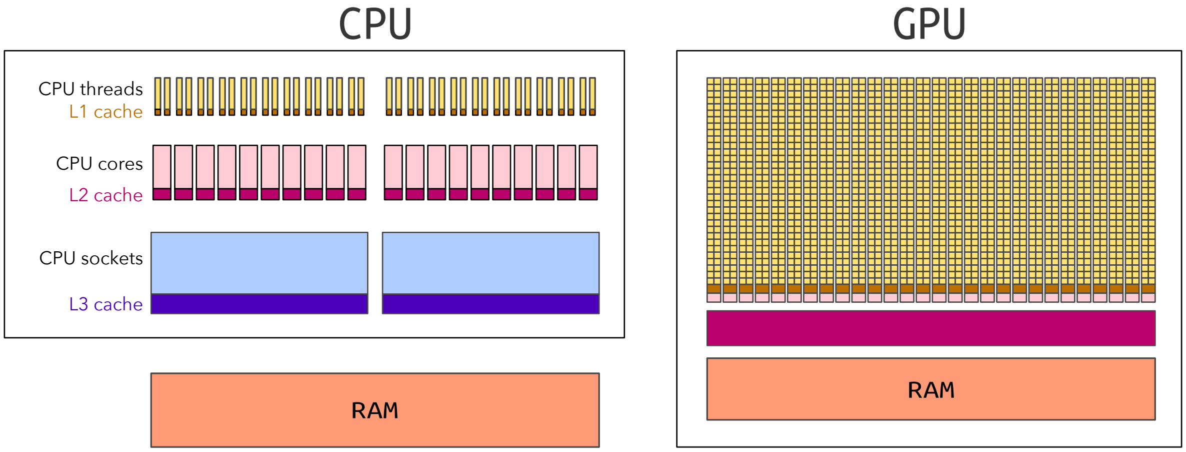


Figure 5.2: Layout of the CPU and GPU architectures. Left: Intel(R) Xeon(R) Silver 4210R CPU, used at the HLT servers. These CPUs are divided in two sockets (blue), each socket having 10 CPU cores (pink), and each core having 2 threads (yellow). All threads can run independently of each other. Right: Nvidia A30 GPU, also used at the HLT. The GPU has 56 multiprocessors (pink), each of them capable of running up to 2048 threads grouped in 32-thread warps (yellow).

GPUs are optimized for parallel processing of relatively simple tasks. GPU threads are simpler, smaller, and individually slower than CPU threads. However, thousands of threads can run in parallel in a modern GPU, making GPU processing of simple tasks several orders of magnitude faster than a CPU. The architecture of a GPU is outlined in Fig. 5.2 (right), based on the Nvidia A30s used in KOTO's HLT.

The A30 has 56 *multiprocessors*, which in terms of architecture can be roughly compared to CPU cores. Each multiprocessor can run up to 2048 threads concurrently, accounting for a total of 114688 threads that can theoretically run in parallel on the GPU[5]. For reference, the 20 CPU cores on the HLT can run a total of 40 threads in parallel. However, important to note is that not all the individual GPU threads can simultaneously perform independent operations. GPU threads are grouped by the GPU in 32-thread groups called *warps*. All threads in a warp execute *the same* instruction at the same time. As an example, using a single thread to multiply two numbers will take one cycle, the same time and resources as using 32 threads to multiply 32 pairs of numbers. Using just two threads of the warp to perform different operations each (e.g. thread 1 doing $a = b + c$ and thread 2 doing $d = 2 \cdot e$) will take two cycles, as all threads in the warp will execute first the first instruction ($a = b + c$), and then the second ($d = 2 \cdot e$). Understanding this architecture and taking advantage of it is important when programming on GPUs.

The GPU RAM is installed within the GPU card and cannot be expanded. Its capacity is typically not larger than tens of gigabytes (24 GiB[6] in the case of KOTO), but it can be accessed by the GPU threads at bandwidths in the order of 1 Tbps. This GPU RAM is referred to as *global memory*. Global memory can be accessible from the CPU. Incoming data from the CPU is received into global memory, and the results of GPU processing are retrieved by the CPU also from the global memory.

Specifications of KOTO's HLT CPUs and GPUs are summarized in Table 5.1.

Table 5.1: Specifications of the CPUs and GPUs used in KOTO's HLT servers.

Component	**Intel(R) Xeon(R) Silver 4210R**	**Nvidia A30**
Max. parallel running threads	40	114688
Max. parallel instructions	40	3584
L1 cache	32 KiB per thread	48 KiB per thread block
L2 cache	20 MiB per core	24 MiB (total)
L3 cache	27.5 MiB per socket	-
Max. Clock rate	3.2 GHz	1.44 GHz
Max. RAM capacity	hundreds of GiB	24 GiB
RAM bandwidth	100 Gbps	1 Tbps

CPUs and GPUs are connected through the PCIe bus. The bandwidth between CPU and GPU in KOTO's HLT is limited by the PCIe 3.0 bus to 100 Gbps.

[5] In practice, the performance of a GPU is usually limited by memory copies, and not by the achievable amount of parallelism when doing calculations.

[6] 1 GiB = 2^{30} Bytes. Not to be confused with 1 GB, equal to 10^9 Bytes.

5.3.2 CUDA and the GPU programming model

The GPU programming model is introduced in this section, and the basic concepts needed to understand the GPU-based HLT trigger software are explained.

CUDA, or the *Compute Unified Device Architecture*, is a parallel computing programming model developed by NVIDIA that can be used to program NVIDIA GPUs. CUDA is generally used as a library in C or C++ programs. CUDA code can be compiled together with CPU code, but all source files including CUDA code must be compiled with the NVIDIA CUDA compiler *nvcc*.

CUDA functions that are executed on the GPU are called *kernels*. Kernels are written in a simple C-like language. CPU libraries or methods cannot be called from kernels.

The most basic computing unit of a GPU is the *thread*. In CUDA, threads are grouped in *blocks*, and blocks are grouped in *grids* as shown in Fig. 5.3. Threads within a block have access to very fast *shared memory*, and their execution can be synchronized. The number of threads per block is decided by the programmer and can be up to 1024 in modern GPUs including the HLT's A30. Threads from different blocks cannot be synchronized, and have no direct access to each other's shared memory. All threads from all blocks can access the GPU's global memory. The maximum number of blocks in a grid can be up to 65535 in modern GPUs, also including the A30, but not all blocks are guaranteed to be executed at the same time.

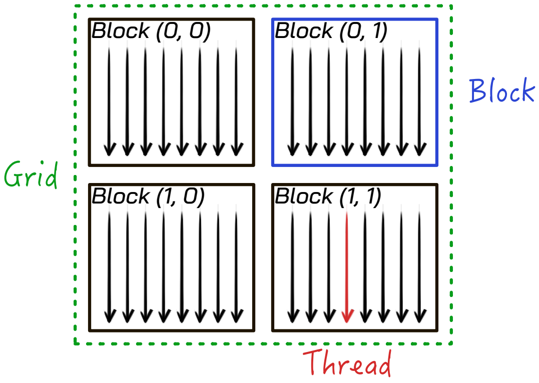


Figure 5.3: A 2D grid created with 4 blocks, each block containing 8 threads. Each thread is uniquely identified by the coordinates of its block inside the grid and its index within the block.

Block and grid dimensions are specified by the programmer at the moment of launching a kernel. For example, a HLT kernel to compress in parallel all waveforms of 100 events[7] would be launched with a 2D grid of dimensions 288 × 100, each block processing just the data from one of the 288 KOTO ADC modules from a single event. 256 threads are launched per block. Groups of 16 are used for each of the 16 waveforms per ADC. The same compression steps are performed in parallel for all waveforms in a block.

[7] In the 2024 beam-time, events were packet in blocks of up to 800 events and processed together.

Kernels run asynchronously with respect to the CPU. From the CPU perspective, a kernel call will return immediately, before the kernel has finished executing. CPU-GPU Memory copies can be launched synchronously (*cudaMemcpy*) or asynchronously (*cudaMemcpyAsync*) with respect to the CPU, depending on the scenario. On the GPU side, both kernels and memory copies are queued in a *stream*, and executed in the order they were queued. The CPU can explicitly wait for a kernel to finish by calling *cudaStreamSynchronize*, or by calling any blocking CUDA function that implicitly waits for the stream to finish.

By default, CUDA functions called independently from different CPU threads are not executed in parallel. The requests are queued sequentially into a single GPU stream. A CPU thread waiting for its results will wait until the GPU has finished executing all previous requests from other threads.

To avoid this situation in multi-threaded CPU programs, multiple CUDA streams can be created, each of them with its own queue of GPU calls. In each Computing Node at KOTO's HLT (Fig. 5.1), eight CPU threads capture OFC-II data in parallel from the 40 Gbps network card. The captured events are processed by eight independent queues, that are associated with eight different CUDA streams. Data is processed and eventually sent to the Disk Node from each stream independently of the other streams.

Multiple kernels can run in parallel in different streams, but only one CPU→GPU or GPU→CPU memory copy can be run at the time. The maximum concurrency allowed by the GPU consists of a single CPU→GPU copy, a single GPU→CPU copy, and one or many functions and/or kernels running in parallel on CPU and GPU. Multiple memory copies in the same direction launched from different streams are still queued and executed sequentially. Allowing parallel transfers in the same direction would not translate into a performance gain though, as the CPU-GPU bandwidth is limited by the PCIe.

GPU applications are rarely limited by computing throughput. Data transfers to and from the CPU, as well as global memory accesses, are generally the bottleneck in GPU applications. CPU↔GPU transfers can sometimes be masked behind CPU processing, as will be shown in section 7.3. Global memory accesses can be minimized by copying once just the needed data from global to shared memory, and finally copying back to global memory just the processing results.

5.4 Contributions from the author of this thesis

The introduction of GPUs in the KOTO HLT was initially proposed and motivated by the author of this thesis. The potential of including GPUs in the HLT was studied and presented to the KOTO collaborators and in public workshops[8]. Eventually, the decision to include GPUs in the new HLT was taken by the KOTO collaboration, and the author of this thesis was responsible for the design and development of the GPU-based HLT software. The author was also responsible for the commissioning of the HLT system and the evaluation of its performance.

[8] See for example Ref. [29], a talk given in 2021 about the potential of GPUs in KOTO's HLT.

The HLT software is publicly accessible and can be found online[9].

Unless explicitly stated otherwise, the following chapters describe the KOTO HLT software as has been developed by the author of this thesis. The efficiency and performance of the HLT are evaluated in chapters 10 and 11. Results are summarized in chapter 12.

[9] The HLT software is available at `https://gitlab.com/koto-l3/l3-sw`.

Chapter 6

Data Capture at 40 Gbps

One of the main goals of the KOTO DAQ upgrade was to eliminate the main bottleneck of the previous DAQ system —the eighteen 1 Gbps links between the Assembly boards and the PC farm— in favor of four 40 Gbps links between the OFC-IIs and the HLT. This upgrade was essential to cope with both the increased beam power and the ambitious physics goals of the 2024 beam-time.

From the HLT perspective, the data capture at 40 Gbps poses important challenges to the HLT software and required a complete re-design of the old capture framework. The 40G[1] packet capture paradigm at KOTO's HLT is covered in this section.

6.1 Outline of the 40 Gbps packet capture

By packet capture we understand the process from intercepting incoming packets at the network interface card (NIC) to storing them in permanent buffers in the HLT's RAM. The layout of the packet capture at the HLT Computing Nodes is summarized in Fig. 6.1. Packets from the OFC-II arrive at the NIC of each node through 40 Gbps optic cables, connected to the NIC through QSFP+ transceivers[2].

[1] In the chapters that follow, "G" will be used interchangeably with "Gbps" to denote Gigabits per second.

[2] QSFP+ (Quad Small Form-factor Pluggable) connectors are the industry standard for 40G links.

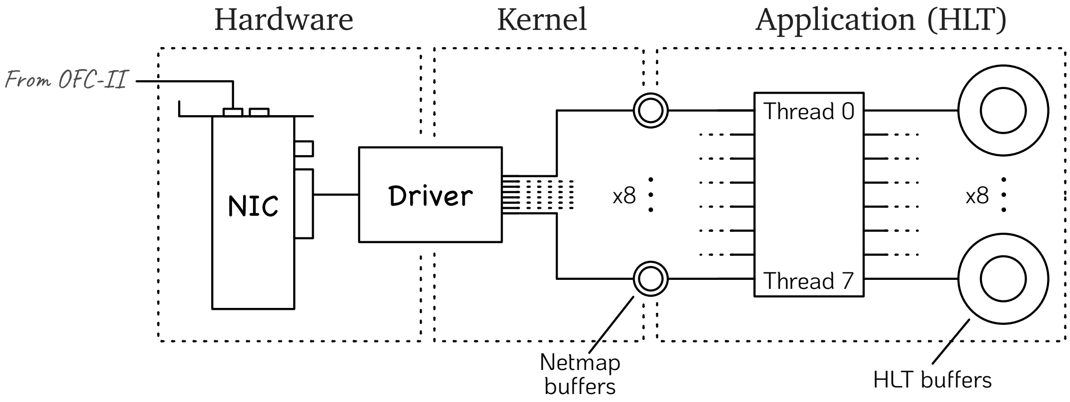


Figure 6.1: Packet capture layout, from the NIC to the HLT buffers in the Computing Nodes' RAM. Eight receiving queues (0 to 7 in the figure) are set up to capture packets in parallel from the NIC.

On the software side, the 40G packet capture is partly made possible through the open-source Netmap framework [30]. To cope with the 40G input, the packet processing needs to be parallelized so that the load can be shared between multiple CPUs. For this purpose, eight independent RX queues (0 to 7 in Fig. 6.1) are managed by eight different CPU threads. Incoming packets are redirected at the NIC driver to a specific RX queue as will be detailed in section 6.2. Each RX queue leads to a circular buffer (also *ring* buffer or just *ring*), where the NIC driver writes the received packets. Hereafter, we will refer to these buffers as the *Netmap buffers*. The structure of the Netmap buffers is described in section 6.3. On the HLT software side, another eight CPU threads move the packets to much larger buffers in the Computing Nodes' RAM, hereafter *HLT buffers*. Transferring packets from the Netmap to the HLT buffers is done taking advantage of the NUMA architecture at the Computing Nodes, as will be briefly covered in section 6.4.

The packet capture needs to be done in real-time during the 2-second "beam on" phase of the spill cycle. Once in the HLT buffers, packets can be processed at a slower phase as was discussed in section 5.1.2.

6.2 Packet redirection at the NIC driver

Configuring eight receiving queues does not automatically guarantee that the load will be uniformly distributed across them. Typically, the NIC driver will redirect packets to queues based on the contents of part of each packet's header. Packets sharing headers will tend to be redirected to the same queue. Understanding the packet redirection at our Intel i40e NIC driver is a necessary step to manually ensure uniform load distribution at the HLT's packet capture stage.

The packets sent by the OFC-II to the Computing Nodes follow the Ethernet II standard [27], whose structure is outlined in Fig. 6.2. The header of Ethernet II frames consists of a source MAC address (6 bytes) and a destination MAC address (6 bytes), followed by two bytes defining the so-called EtherType. The EtherType is normally used to identify the protocol encapsulated by the packet's payload, i.e., the structure of the packet's payload. The payload is attached after EtherType. Depending on the protocol, more headers may be attached at the beginning of the payload. Four bytes are reserved at the end of the packet for the CRC (Cyclic Redundancy Check), currently not used in KOTO.

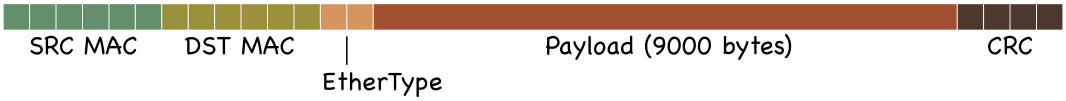


Figure 6.2: Structure of an Ethernet II frame. The payload can be any size up to 9000 bytes

We said before that the NIC redirects packets based on the contents part of their headers. More specifically, the exact portion of the packet's headers used by the NIC is determined by the packet's protocol, and therefore by EtherType. For instance, packets following the widely used IPv4 protocol are identified by an EtherType value of 0x0800. In this case, the driver will base its decision on the source and destination ports, which are part of all IPv4 packet's additional headers, and located at a fixed position at the beginning of every IPv4 packet's payload. Although this method already provides some sort of balancing, it is not intended to maximize performance, and it does not guarantee uniform load distribution across all configured queues.

EtherType is normally set by the sender to a standard and known value[3], according to the protocol the packet follows. The receiver side can then use this information to know how to process the packet's payload. This is particularly useful in systems connected to open networks such as the Internet, etc., where different packets coming from different sources, each packet with its own length and format may be received. In our case, the OFC-II and the HLT are connected through an isolated network, and the format of the OFC-II packets is fixed and known a priori by the HLT.

To our advantage, if a packet with an unknown EtherType is received at the NIC, the NIC will determine its destination RX queue based on the contents of EtherType itself. This fact is used by the OFC-II and the HLT to gain control of the packet redirection and to ensure uniform load distribution in all HLT nodes.

The value of the EtherType bytes can indeed be set freely by the OFC-II, just as long as the following conditions are satisfied. First, EtherType must not correspond to any standard format, as the OFC-II packets themselves do not follow any standard format. Second, the two bytes in the EtherType field have to compute a value larger than 0x0600, or 1536 decimal. For values below this threshold, the packet's format is assumed to follow the IEEE 802.3 standard, where the value encoded in the EtherType field indicates the length of the packet's payload. Details on the IEEE 802.3 standard can be found in Ref. [27].

Assuming the two conditions above are satisfied, the packet redirection at the NIC works

[3] A list of commonly known Ethernet protocols and their corresponding EtherType can be found at https://en.wikipedia.org/wiki/EtherType

as outlined in Fig. 6.3. A Hash is first calculated from the EtherType bytes, as will be detailed in section 6.2.1. This hash is then used to determine the destination queue for each packet, as will be explained in section 6.2.2.

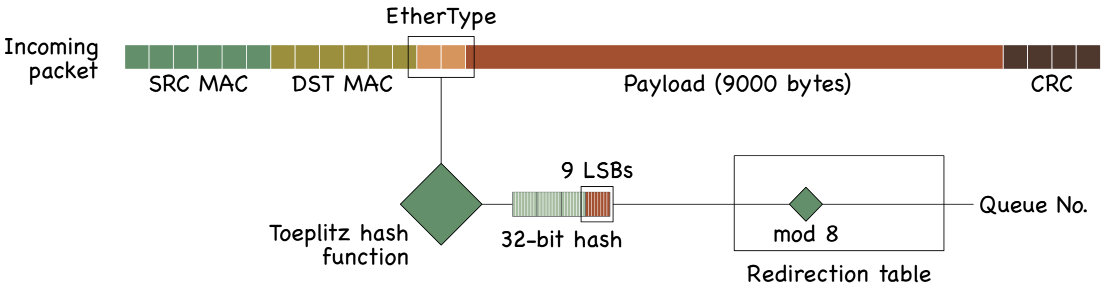


Figure 6.3: Outline of the packet redirection at the NIC driver of the HLT nodes.

6.2.1 Toeplitz matrix and the Toeplitz hash algorithm

The Intel i40e driver uses the Toeplitz hash function to convert an input bit set (the two EtherType bytes in our case) to the hash that will determine which queue each packet gets redirected to. The working principle of the Toeplitz hash function is covered in this section.

In the following, the input bit set will be denoted as $\vec{d} = \{d_0, \ldots, d_m\}$. In our case, the input bit set consists of two bytes,thus $m = 15$. The hash will be denoted as the bit set $\vec{h} = \{h_0, \ldots, h_n\}$. Our NIC works with 32-bit hashes, thus $n = 31$. Finally, let k_i be the entries of a long enough bitstream $\vec{k}$ known as the Toeplitz key. The Toeplitz hash function uses the key $\vec{k}$ to build the Toeplitz matrix T. The hash $\vec{h}$ is then calculated as the product of T and $\vec{d}$:

$$\underbrace{\begin{pmatrix} k_0 & k_1 & \cdots & \cdots & k_m \\ k_1 & \ddots & \ddots & k_m & k_{m+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ k_n & \cdots & \cdots & \cdots & k_{m+n} \end{pmatrix}}_{T} \cdot \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_m \end{pmatrix} = \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{pmatrix} \tag{6.1}$$

where all the operations are modulo 2. Note that the entries of each ascending diagonal in the Toeplitz matrix T are equal. To build the matrix T, the length in bits of the Toeplitz key $\vec{k}$ has to be at least $m + n + 1$, and not $m \times n$. In our case, this is $15 + 31 + 1 = 47$ bits. The key used by the Intel i40e NIC driver is long enough, with a total length of 52 bytes, or 416 bits. The contents of the key $\vec{k}$ will be discussed later in section 6.2.3.

The HLT makes use of 8 RX queues. The EtherType $\vec{d}$ is therefore set to rotate among eight values, from 0x2000 to 0x2007 in hexadecimal[4]. These values satisfy the two conditions for EtherType that we introduced in the previous section.

[4] This is 00100000 00000000 to 00100111 00000000 in binary, or 32,0 to 39,0 in decimal

6.2.2 Redirection table

The mapping between each possible hash $\vec{h}$ and each of the eight RX queues is defined at the redirection table. In our NIC driver's case, not the whole 32-bit hash but only its last 9 least significant bits (bits 23 to 31) are used. The destination queue is determined as $h \bmod 8$, where h is the value of the last 9 bits of the hash, $h = (2^8\, h_{23} + 2^7\, h_{24} + \cdots + 2^0\, h_{31})$. The quantity h can be any value from 0 to $2^9 - 1 = 511$.

6.2.3 Solution to the Toeplitz hash problem for 8 RX queues

Solving the Toeplitz hash problem consists of finding a key $\vec{k}$ that, given the eight $\vec{d}$'s set at the OFC-II (0x2000, ..., 0x2007), produces eight hashes $\vec{h}$ whose mod 8 are uniformly distributed between 0 and 7. Our NIC uses only the last 9 bits of the hash, $h_{23}, \ldots, h_{31}$, to determine the RX queue. Thus, the hash calculation introduced in eq. 6.1 can be rewritten as

$$\begin{pmatrix} k_{23} & k_{24} & \cdots & \cdots & k_{38} \\ k_{24} & \ddots & \ddots & k_{38} & k_{39} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ k_{31} & \cdots & \cdots & \cdots & k_{47} \end{pmatrix} \cdot \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{15} \end{pmatrix} = \begin{pmatrix} h_{23} \\ h_{24} \\ \vdots \\ h_{31} \end{pmatrix} . \tag{6.2}$$

The solution to the Toeplitz hash problem in our case is not unique, but the following is the matrix representation of the simplest one[5]:

$$T = \begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}$$

where the '$\cdot$' represent zeros. It can be checked that the matrix T multiplied by the binary representation of the eight inputs $\vec{d}_0 \ldots \vec{d}_7$ gives eight hashes $\vec{h}$ whose mod 8 are uniformly distributed between 0 and 7:

[5] The corresponding Toeplitz key is, in hexadecimal, {00 00 00 00 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00}, where each two-character group represent a byte.

$$
\begin{aligned}
(T_{nm}\, d_{0m} \bmod 2) &= (0,0,0,0,0,0,0,0,0) = 0\\
(T_{nm}\, d_{1m} \bmod 2) &= (0,0,0,0,0,0,1,0,1) = 5\\
(T_{nm}\, d_{2m} \bmod 2) &= (0,0,0,0,0,0,0,1,0) = 2\\
(T_{nm}\, d_{3m} \bmod 2) &= (0,0,0,0,0,0,1,1,1) = 7\\
(T_{nm}\, d_{4m} \bmod 2) &= (0,0,0,0,0,0,0,0,1) = 1\\
(T_{nm}\, d_{5m} \bmod 2) &= (0,0,0,0,0,0,1,0,0) = 4\\
(T_{nm}\, d_{6m} \bmod 2) &= (0,0,0,0,0,0,0,1,1) = 3\\
(T_{nm}\, d_{7m} \bmod 2) &= (0,0,0,0,0,0,1,1,0) = 6
\end{aligned}
$$

The OFC-II setting $\vec{d}$ (the EtherType two bytes) as described above, and the HLT nodes configuring their NIC drivers with the key obtained in this section results in a uniform load distribution among the eight CPUs managing each RX queue. To ensure that all packets from the same event end up in the same buffer, EtherType is rotated at the OFC-II once per event, and not per packet.

During the 2024 physics DAQ, the average data rate into each Computing Node was 10.7 Gbps, as was discussed in section 5.1.1. However, within every event, all 68 packets are transmitted at 40 Gbps. To average the rate and reduce possible impacts on performance, the size of each ring buffer is set to 20480 packets, or 301 events.

6.3 Ring buffers and RX queues

As we outlined at the beginning of this chapter, each RX queue leads to a small Netmap ring buffer that connects the "kernel space", where the NIC driver runs, to the HLT software. Another eight CPUs at the HLT side quickly move these packets from the Netmap buffers to the HLT buffers, much larger circular buffers on the RAM. The structure of ring buffers, represented in figure 6.4, is briefly described in this section.

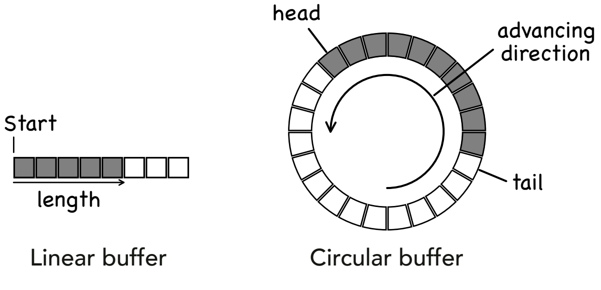


Figure 6.4: Representation of linear and circular buffers.

Usual *linear* buffers can be defined by a fixed start pointer and their length, as shown in Fig 6.4 (left). Elements are added to the back as the buffer expands and also removed from the back as the buffer shrinks. The nth element of the buffer can be accessed by adding

n positions to the start pointer. Attempting to add elements to a buffer that is full is not allowed.

A *ring buffer* is defined by two addresses, named *head* and *tail*. Head is advanced, and the buffer expands, as *new* elements are added to the front. Tail is advanced as *old* elements are removed from the back. Note that head is not allowed to overtake tail. Adding new elements to a buffer that is full results in simultaneously advancing both head and tail, overwriting the oldest element.

The HLT makes extensive use of circular buffers, as they are particularly useful when the writing and reading processes work independently from each other. A good example of these is the Netmap buffers, introduced in Fig. 6.1, accessible by both the NIC driver and the HLT software. The NIC driver advances head as new packets are received, and the HLT nodes independently advance *tail* as packets are copied to the RAM. The much larger HLT buffers are also circular, written by the 40G packet capture stage, and read back independently by the event reconstruction stage.

To keep up with the 40G NIC, the HLT buffers have to be allocated and written in a specific way, aiming to maximize the bandwidth between them and the NIC. The method found to overcome this situation is described in sec. 6.4. The structure of the HTL buffers is then briefly treated in sec. 6.5.

6.4 NUMA nodes and CPU affinity

The 40 CPUs available on each Computing Node are distributed in 2 CPU sockets, each socket hosting 10 physical cores with two threads per core. The two sockets are located in Fig. 6.5 under the two visible metal heat sinks.

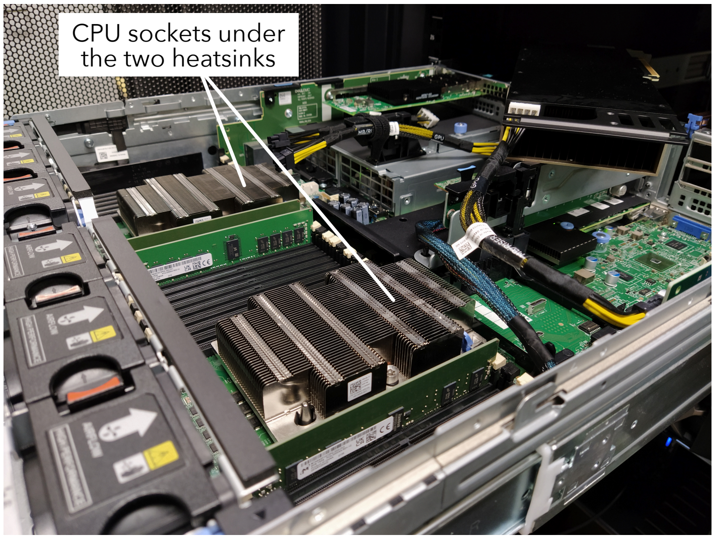


Figure 6.5: A picture of the interior of one HLT node, showing some of its main components. Two CPU sockets are located under two heat sinks.

It is common in large systems like the HLT nodes to follow the Non-Uniform Memory Access (NUMA) architecture. In these systems, the RAM slots are divided into two groups, and each group is assigned to one CPU socket. CPUs in a socket have faster access to their local memory banks. All CPUs can still access all memory regions, but with a higher latency cost when accessing remote memory banks.

The difference in latency between local and remote memory access is roughly a factor of two, which is significant in memory-extensive tasks. In KOTO's HLT, the high incoming data rate makes minimizing this latency necessary to avoid any packet loss[6].

To take advantage of the NUMA architecture, the HLT buffers that hold the captured data are explicitly allocated into memory belonging to the NUMA node local to the 40G NIC. Each thread participating in the data capture process is explicitly attached to a physical CPU local to the same NUMA node. This includes eight threads moving data from the NIC to the Netmap buffers, and another eight threads moving data from the Netmap buffers to the HLT buffers.

Explicitly mapping threads to physical CPUs is known as setting the *thread affinity*. Setting thread affinity to specific CPUs is required in the HLT nodes to avoid packet loss. The affinity of HLT threads not involved in the 40G packet capture can be left to the operating system, as maximizing the memory bandwidth is only a priority at the packet capture stage.

6.5 Data storage on the HLT RAM

At the OFC-II, partial event data from all OFC-Is are buffered until a full event has been received. Then, the event buffer is chopped into 67 8800-byte packets and a single 7840-byte packet. To build these packets, the OFC-II event buffer might be cut at any position, including in the middle of an ADC header or in the middle of a waveform. For this reason, on the HLT side, the event data cannot be easily accessed until packets have been re-aligned back in memory. As packets are retrieved from the Netmap buffers, a 40-byte OFC-II header is separated from the event data. From the OFC-II header, only the packet number within the event (from 0 to 67) is recorded.

Each HLT buffer has two dimensions, the first one indexing packets and the second one indexing bytes inside each packet. However, in physical memory, the 2D array is unfolded into a 1D contiguous block of memory. First, this allows the capturing stage to work in units of packets, writing to the buffer one packet after another. Second, the processing stage can ignore the packet structure and "re-chop" the buffer in units of ADC data (One ADC data = 2064 bytes), making easier the data integrity checks that will be covered in section 7.1.

The packet numbers, extracted from the OFC-II headers, are stored in a separate small *packet-No.* buffer, which for every packet stores just the packet number. The head and tail of each HLT buffer and its corresponding packet-No. buffer are linked, and they always contain the same number of packets.

[6] This statement will be proven with data in section 11.1

Note that every circular buffer is in reality just a wrapper around an underlying linear memory array in the physical RAM. The last element of the physical array is "logically stitched" to the first one so that the memory looks contiguous to the software. This process requires special treatment in the case of the HLT buffers, since data is retrieved in units of ADC data, and the last ADC data is very likely to be split between the end and the beginning of the underlying linear array. The issue is illustrated in Fig. 6.6, together with its solution. First, enough space is allocated at the end of the memory array to accommodate an extra packet, without making the software aware of this extra slot. Before the software attempts to retrieve the event data whose length crosses the configured buffer's boundary, the packet at index 0 (packet 40 in the figure) is copied to this extra hidden slot.

The HLT then retrieves from the circular buffer all the event data, ADC by ADC. When it reaches the last ADC of the underlying linear buffer (highlighted red in Fig 6.6), it retrieves it normally from contiguous memory. When advancing to the next ADC, the HLT is naturally redirected back to the beginning of the array, where the next ADC (blue in the figure) is physically stored.

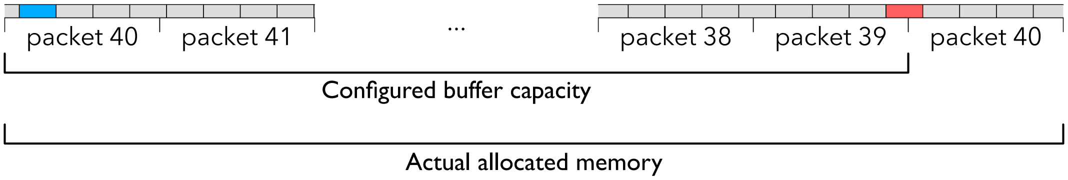


Figure 6.6: The solution to the memory discontinuity in the configured HLT circular buffers. Grey boxes represent data from one ADC.

Chapter 7

Event Reconstruction and Event Selection

The trigger menu used in physics runs during the 2024 beam-time has been described in section 3.1. To reduce the event rate at the HLT output, the HLT filters events from the triggers with the highest rate, based on a set of physics criteria that differs trigger by trigger.

The event reconstruction and selection at the HLT are fully performed on GPU. Before starting with the physics event reconstruction, event data has to be extracted from the raw OFC-II data, and events with missing or corrupted packets need to be discarded. This process is covered in this section 7.1. Complete events need to be efficiently sent to the GPU. We will briefly cover this data transfer in section 7.3. The physics reconstruction will be then described from section 7.4. The event selection criteria set at the HLT for each trigger is finally given in section 7.6.

7.1 Event building and data integrity checks

From this section, we will describe the processing of event data already collected in the HLT buffers. The first goal of this processing is to prepare a block of events that do not contain missing packets and belong to the same spill. This process is covered in this section. All events in the block will be sent to the GPU, reconstructed, selected, and compressed simultaneously. They will be finally sent together to the Disk Nodes, where they will be combined from other event blocks from the same spill and written to disk.

The process is started by eight "master" threads assigned to the eight HLT buffers. Each master thread first estimates the number of events in its HLT buffer as

$$\text{events} = (\text{head} - \text{tail})\ /\ 68\ , \tag{7.1}$$

where 68 is the number of packets per event. Then, it defines the boundaries of the event block as [tail, tail + events · 68]. Since the amount of memory that has to be pre-allocated in the HLT depends on the size of these blocks, an upper limit of 800 events per block was set in 2024 beam-time.

If no packets were missing in the previous block, the packet number stored at index *tail* should be 0. If this is not the case, the master thread will move forward until the next packet "0" is found, advancing *tail* in the process, and therefore rejecting all packets in between. Once the first packet of an event is found, the master thread moves forward packet by packet checking that the packet number increases by 1, up to 68-1.

If an event is complete, data starts being retrieved from the HLT buffer ADC by ADC. While doing so, ADC headers are separated from the ADC data while checking that the ADCs are correctly sorted (crate by crate and ADC module by ADC module). The Spill ID within the run is taken from the first ADC header. If a new spill is detected, the event block is considered complete, and a new block starts being built with events from the new spill. Events with corrupted ADC headers are not discarded, but they are reported in real-time to the run monitor, so that the shifter can take action if necessary. Header corruption does not necessarily lead to unusable events or corrupted waveform data. Corruption in the ADC headers is carefully studied offline. This issue will be further discussed in section 13.2.2.

The trigger that each event belongs to, set by Top CDT at the upstream of the DAQ, is used to tag the event at the HLT[1]. This event tag will decide the path each event will follow through the HLT's GPUs.

7.2 HLT event tag

The eight trigger types taken in physics runs during the 2024 beam-time were shown in table 3.1. Depending on their trigger type, different events will follow different processing paths on the GPU. For instance, events belonging to the $K_L \to \pi^0\nu\overline{\nu}$ trigger will bypass the reconstruction and the pedestal suppression at the HLT. Events belonging to the $K^+ \to \pi^+\pi^0$ or the $K_L \to \pi^0 e^+ e^-$ triggers will be reconstructed, but their selection criteria or clustering parameters (section 7.5.2) will differ. Other more special triggers, such as the minimum bias triggers, are just pedestal-suppressed and compressed. For the GPUs to know what to do with each event, a 16-bit *event tag* is generated on the CPU, categorizing each event based on its trigger type. According to its value, during 2024 runs an event could fall into any of the following five categories:

1. Reconstruction and selection of 5-cluster $K_L \to 3\pi^0$.
2. Reconstruction and selection of $K^+ \to \pi^+\pi^0$.
3. Reconstruction and selection of $K_L \to \pi^0 e^+ e^-$.

[1] This information is encoded after in a special ADC that is not connected to any detector.

4. CSI waveform pedestal suppression and lossless compression.

5. Lossless compression.

Triggers reconstructed and selected undergo also pedestal suppression and waveform compression.

A randomly selected 1% of events are further selected as *HLT-minimum-bias* events. These events will normally undergo all stages of the HLT corresponding to their original HLT tag, but they will not be discarded despite their selection results. A bit in the HLT tag is reserved to identify these events, and their selection results will also included in the HLT tag[2]. This sample of events is used offline to measure and monitor the HLT efficiency.

At this stage, we have a block of events where all waveforms are aligned ADC channel after ADC channel and ADC after ADC. All events in the block are complete and belong to the same spill. The process of sending to the GPU all events in the block, together with the HLT tag and the ADC headers, is covered in section 7.3.

7.3 Data transfer to the GPU

The event block built in the previous section is copied to the GPU as it is being built. Once an event is ready on the CPU, it is sent to the GPU as the next event is prepared. This might sound inefficient since, in general, a much larger throughput can be achieved by transferring fewer but larger buffers. However, in the case of KOTO's HLT, event data can be sent to the GPU in an average of 16 µs, while preparing an event on the CPU takes an average of 531 µs. The event-by-event copy to the GPU can be masked behind the CPU processing, which overall saves time. The advantages of this scheme are illustrated in Fig. 7.1.

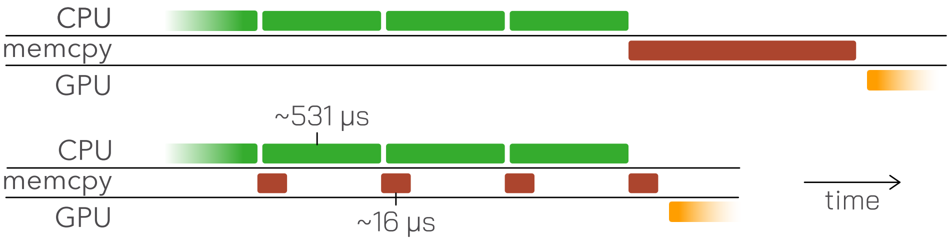


Figure 7.1: Event data is prepared on CPU event by event. Top: All events are copied to the CPU at once. Bottom: events are copied to the GPU event by event. The length of the blocks is not to scale.

The ADC headers of all events in the block, as well as their HLT tags are also transferred to the GPU once the complete event block has been copied.

[2] this process will be covered in section 7.6.

7.4 Energy and time reconstruction

The reconstruction and selection processes at the HLT are entirely performed on GPU, and based on variables from the CsI calorimeter. The reconstruction process is detailed in this section. Some of the concepts about the GPU programming model introduced in section 5.3.2 will be utilized in this section. The event reconstruction is performed in parallel for all events within an event block. Furthermore, up to 8 event blocks can be processed at the same time, one per CUDA stream.

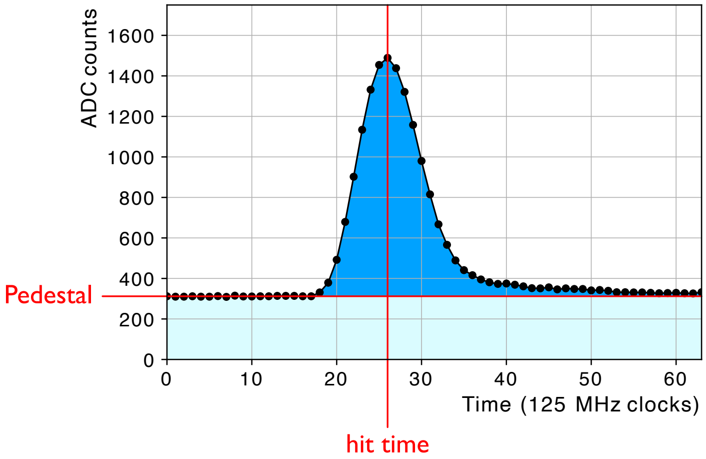


Figure 7.2: Definition of hit time and pedestal of a sample waveform. The dark blue area represents the integrated ADC used to calculate the waveform's energy.

The channel energy and time are calculated in parallel for all 64-sample waveforms from all 2716 calorimeter channels of every event in the block. Energy is calculated from the pedestal-subtracted integrated ADC, and hit timing is extracted from the highest sample in the waveform, as shown in Fig. 7.2. The HLT uses the pedestal value calculated at the ADCs, stored in the first sample of every waveform. Energy reconstruction is detailed in section 7.4.1. Time reconstruction is detailed in section 7.4.2.

7.4.1 Energy reconstruction

The waveform energy is calculated from its integrated ADC. To compute on GPU the waveform's integrated ADC, the ADC counts of each waveform bin are added up as shown in Fig. 7.3.

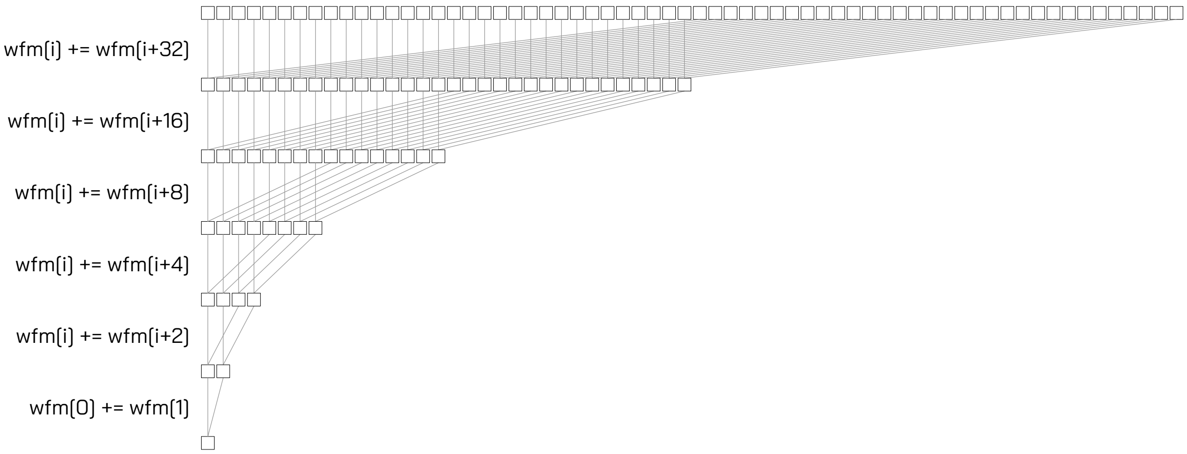


Figure 7.3: Calculation of the integrated ADC of a 64-sample waveform in 5 iterations, using 32 threads per waveform.

From the pedestal-subtracted integrated ADC, the energy in MeV of a CsI channel is calculated as

$$E = \frac{\text{Integrated ADC [Counts]}}{\text{Conversion factor [Counts/MeV]}} \cdot F_{\text{non-linearity}}(\text{peak height}) \tag{7.2}$$

where the conversion factor is calibrated offline channel by channel with cosmic ray data before each beam-time.

The non-linearity factor $F_{\text{non-linearity}}$ accounts for the fact that the linearity between the energy deposited in the crystal and the ADC counts is lost at high energy depositions. In the offline analysis, the non-linearity factor is obtained by performing a fit to experimental data using each waveform's peak height as input. The peak height is defined as the maximum ADC count minus the waveform pedestal. The value of the $F_{\text{non-linearity}}$ correction is shown for multiple values of peak height in Fig. 7.4. Before beam-time, an analytical function that approximates the offline non-linearity factor, shown black in Fig. 7.4, is obtained. This function is then utilized online. The correction is set to 1 for peak heights lower than 3000 counts, and to 0.86 for peak heights higher than 13000 counts.

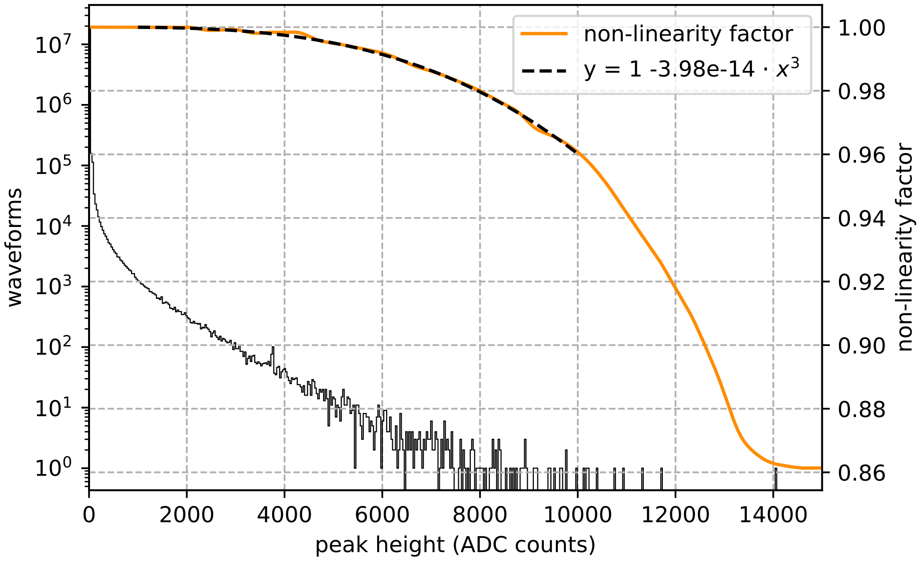


Figure 7.4: Right axis: The non-linearity factor used offline (orange) and the 3rd degree polynomial approximation used online (black). Left axis: The peak height distribution of calorimeter waveforms from a physics run.

The peak height distribution of CsI waveforms from a physics run is overlaid with the non-linearity factor in Fig. 7.4. Less than 0.02% of the CsI waveforms reach peak heights larger than 3000 counts (roughly 300 MeV). Less than 0.002% of the waveforms reach peak heights larger than 6000 and need a correction larger than 1%.

7.4.2 Time reconstruction

All calorimeter ADCs simultaneously receive a trigger signal when the L2 trigger passes, and record a 512 ns waveform consisting of 64 samples using their internal 125 MHz clock. Occasionally, particles coming from processes other than the triggered event can accidentally hit the calorimeter within the same 512 ns window as the triggered event. These hits are called accidentals. An example of them is given in Fig. 7.5. Accidental waveforms need to be rejected so they do not interfere with the reconstruction of the triggered event. Typically, the start time of an accidental waveform is visibly before or after the start time of the triggered event. The HLT makes use of this time to reject accidental hits at this stage.

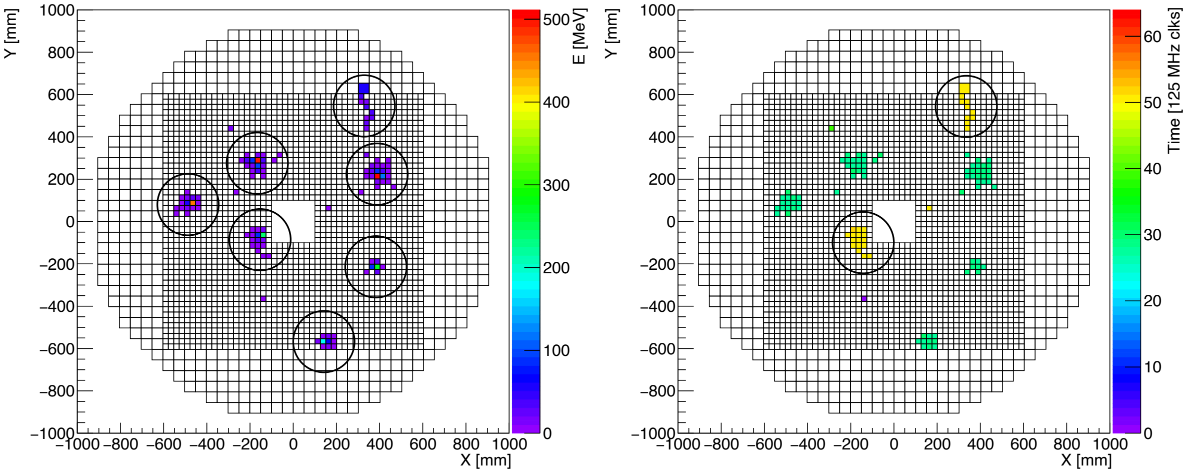


Figure 7.5: An example of an event with accidental hits. Left: Energy deposited in each crystal. Seven clusters are observed, identified by circles. Right: Timing of each crystal hit, in units of waveform sample (0 to 64). Two of the clusters (also identified by circles) belong to accidental hits, around 20 clocks (160 ns) after the triggered event.

Hit time is simply defined at the HLT as the position of the maximum sample in the waveform. In the offline analysis, the *Constant Fraction Time* (CFT) is used instead. The calculation of the CFT is shown in Fig. 7.6.

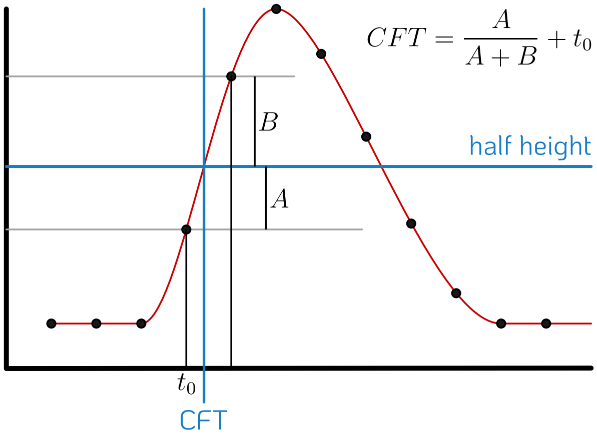


Figure 7.6: Constant Fraction Time calculation. The CFT is the time at which the waveform reaches 50% of its peak height, calculated as shown in the figure.

A comparison between the CFT method and the maximum sample method is shown in Fig. 7.7, using a sample of waveforms from CsI channels registering hits with more than 3 MeV. The standard deviation of the CFT peak is 0.47 samples, while the standard deviation of the maximum sample peak is 0.71 samples. The CFT provides a slight precision improvement over using the position of the highest waveform sample. This is mostly due to the fact that the resolution of the latter is limited to the waveforms' 8 ns sampling time, while the CFT method can return real values between waveform samples. Still, the HLT obtains the hit time from the highest sample in the waveform because of the simplicity of the calculation.

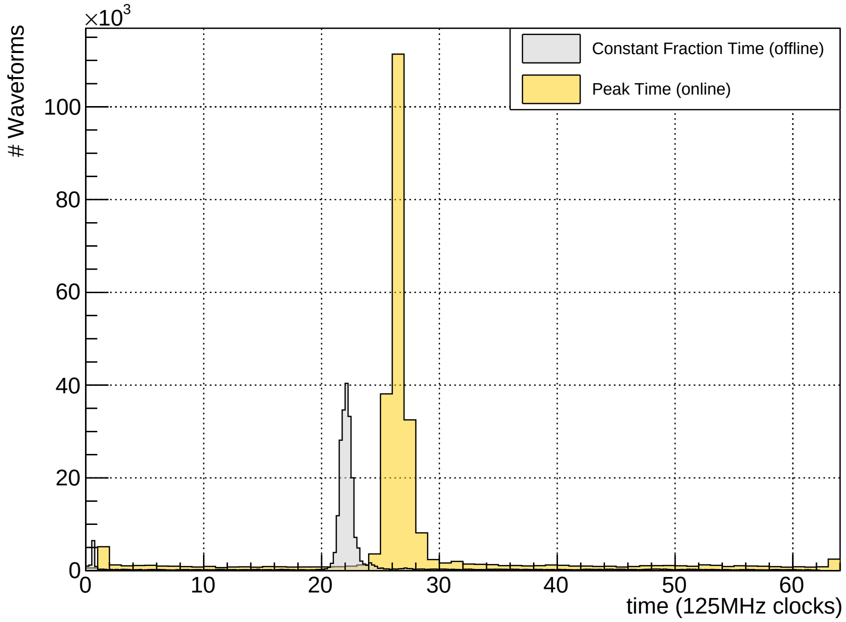


Figure 7.7: Comparison between the CFT time (gray) used offline, and the maximum sample coordinate (yellow) used online. Data is taken from a sample of real events from 2024 runs, and only hits with more than 3 MeV are considered. The bin width of the CFT distribution is 1/4 of the bin width of the maximum sample distribution. Accidental hits are evenly distributed across all 64 samples.

The timing window used at the HLT is manually defined at the beginning of beam-time, from distributions like the one shown in Fig. 7.7. The event selection cannot be enabled until the timing window is fixed, but the process is quick, and the distributions do not change over time during beam-time. In the 2024 runs, a 6-sample wide timing window between samples 24 and 30 was defined and applied to the yellow distribution in Fig. 7.7. Hits outside this timing window would be considered accidentals at the HLT.

7.5 Clustering

In the context of the event reconstruction at the HLT, clustering is the process of identifying the hit position and deposited energy of a particle in the CsI calorimeter, from the energy and timing of the waveforms recorded by the individual calorimeter crystals.

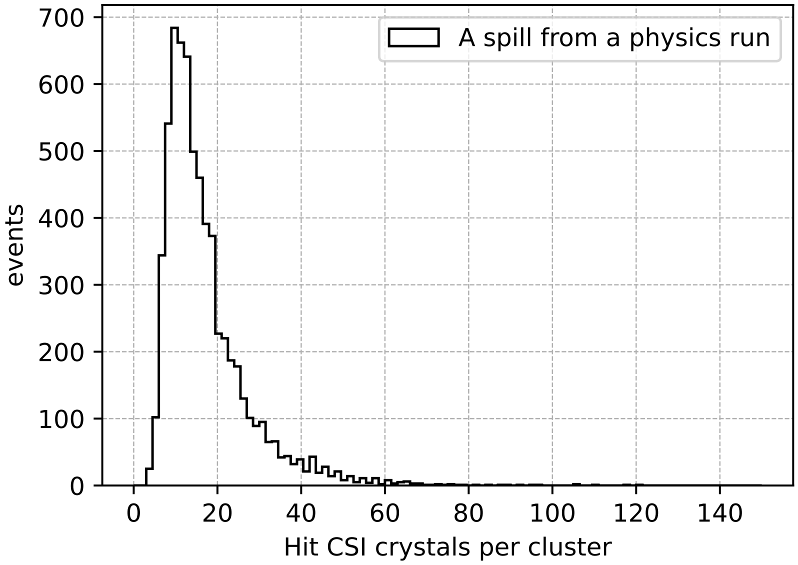


Figure 7.8: Number of hit channels per cluster in the CsI calorimeter, as reconstructed by the offline algorithm. These events were taken in a 2024 physics run including triggers collecting photon and charged pion clusters.

A hit in the CsI calorimeter is recorded as a cluster. A cluster typically spans 10 to 30 crystals, as shown in Fig. 7.8. However, the number of hit crystals per cluster, as well as their energy distribution depends on the hit particle type and energy. Particles such as photons or electrons typically produce small and shallow electromagnetic showers[3] in the CsI crystals, leading to clusters where most of the energy is concentrated in the center. A photon-like cluster is shown in Fig. 7.9 (left).

Particles like pions or neutrons, however, can produce hadronic showers and penetrate deeper into the CsI crystals. This leads to more complex clusters, typically with a more spread energy deposition than photons. An example of a photon-like cluster and a pion-like cluster is shown in Fig. 7.9.

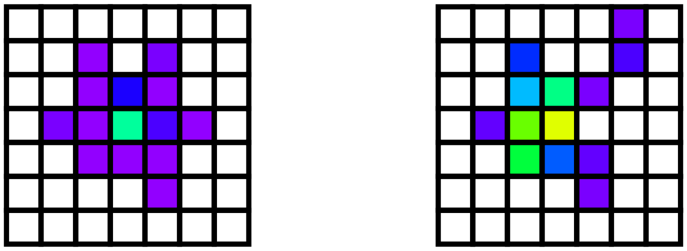


Figure 7.9: Left: a photon-like cluster, extracted from a $K_L \rightarrow 3\pi^0$ candidate event. Right: a pion-like cluster, extracted from a $K^+ \rightarrow \pi^+\pi^0$ candidate event. Both clusters are around 250 MeV. The colormaps are not normalized.

[3] Photonuclear interactions can happen with $\mathcal{O}(1\%)$ probability before an electromagnetic shower. In this case, secondary particles (pions or neutrons) produced in the interaction can lead to more complex showers.

KOTO makes use of three different clustering algorithms, which differ in precision requirements and performance constraints.

- The L2 algorithm. This is the algorithm that runs on KOTO's clustering OFC module. It was described in section 4.6.3. This algorithm counts the number of clusters, but does not compute their position or energy.
- The offline algorithm, running as part of the offline event reconstruction. The offline reconstruction needs to be precise. On the other hand, calculation time is not a concern.
- The HLT algorithm, which needs to be fast, to cope with the DAQ rate in real-time and precise, to allow for efficient event selection based on clustering results.

The clustering algorithms used online at the HLT and offline are different because of the extra constraints the HLT is subject to. The offline algorithm is briefly described in section 7.5.1. The need for a new algorithm in the context of a GPU-based HLT is then discussed, and the online algorithm is finally described in section 7.5.2.

Both the HLT and the offline algorithms start from a list of *hit channels*. The word *channel* here encompasses the CsI crystal, the PMT reading it out, and the ADC channel digitizing the PMT signal. Hit channels are channels that recorded a hit above 3 MeV.

7.5.1 Offline clustering algorithm

By definition, the list of hit calorimeter channels includes accidental hits, not belonging to the triggered event. Offline, accidental hits are also clustered. Off-time clusters are rejected at the end of the clustering process.

The offline clustering process is illustrated in Fig. 7.10. The algorithm starts by taking any hit channel as a cluster seed, shown in red in Fig. 7.10-1. Then, it defines a squared window of size 5 × 5 crystals[4] around the seed. It then loops through the list of other hits in the calorimeter, and if any hit is within that window, it is added to the local cluster, and removed from the original hit list.

In the next iteration (Fig. 7.10-2), the algorithm loops through the list of hits in the local cluster, testing each of them against all the hits left in the calorimeter. If any hit in the cluster finds a new hit within its own window, the new hit is added to the cluster, and removed from the original hit list. The process is repeated until no new hits are left to be added to the cluster (Fig. 7.10-3 to 5). A cluster, red in Fig. 7.10-5, has been built at this point.

The algorithm picks another hit from the ones left on the original hit list and repeats the process (Fig. 7.10-6). Once all hits have been assigned to a cluster, the algorithm finishes. Clusters containing a single hit crystal are finally discarded (Fig. 7.10-7). Isolated fake hits

[4] A 5 × 5 crystal square in KOTO's CSI calorimeter contains the same crystals as a 71 mm circle.

most often come from noisy channels whose integrated ADC happened to compute energy just above the 3 MeV threshold. This easily happens if the pedestal is underestimated, as will discussed later in section 8.1. In some cases, isolated hits can also originate from low-energy neutrons produced in photonuclear interactions before electromagnetic showers, or by particles produced in hadronic showers and scattered outside the shower. These hits are typically not far away from the main cluster.

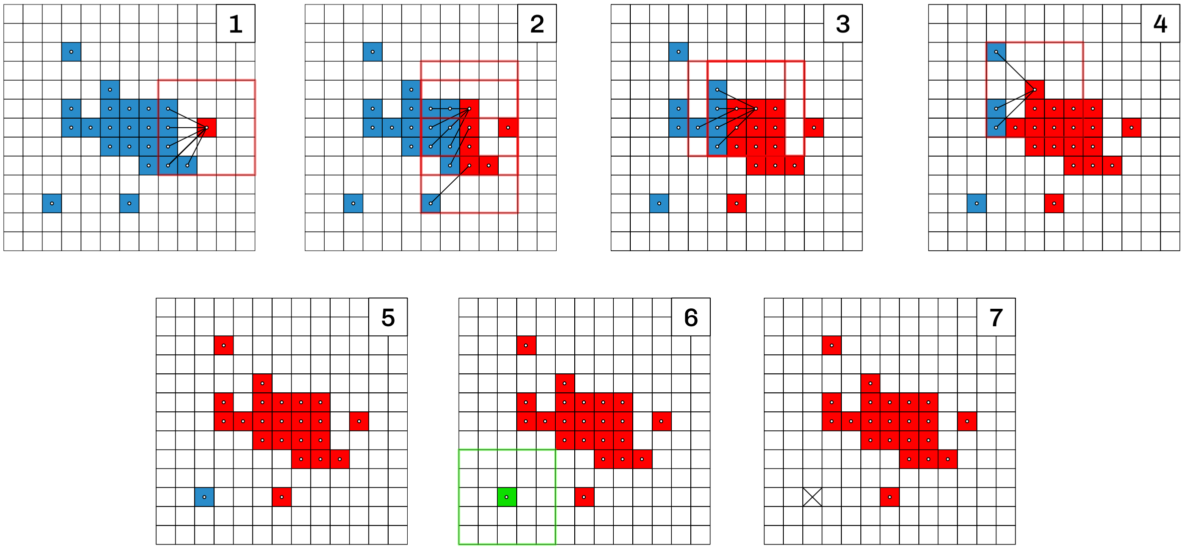


Figure 7.10: Offline clustering algorithm. 1: A hit channel is selected (red). After looping through all blue crystals, the ones within the squared red window are added to the cluster. 2., 3., and 4: The process is repeated while new hit crystals are added to the cluster. 5: The red cluster is complete. 6: The process is repeated until all CsI hits have been assigned to a cluster. 7. Single-hit clusters are discarded.

Once all clusters have been constructed, the timing of each cluster is set to the average timing of all the waveforms of hit channels in the cluster. Clusters are grouped according to their timing, and the group with lower standard deviation is selected as the event. Clusters outside this group are considered to come from accidental hits.

Note that the energy of each crystal does not play any role in the offline clustering algorithm, as all crystals with $E > 3$ MeV are treated equally. Cluster energy is calculated after clustering as the sum of the energy of all hit channels in the cluster.

Need for a new online algorithm for the HLT

The offline algorithm has two important drawbacks that justify the need for an alternative algorithm to be used at the HLT.

The first one is the computational cost, and in particular its quadratic scaling with the number of hits. The online clustering algorithm is required to spend a small *and consistent* (in other words, predictable) amount of time processing every event. In particular, the time consumed to cluster an event should have a small dependence on the number of hits or the event type. This will allow us to make reliable predictions on the HLT performance and limitations as a function of only the event rate.

The second reason motivating the need for a new algorithm is the limited potential and flexibility of the offline algorithm. The limited potential is due to the channel energy information not being used offline during clustering. A consequence of this is the tendency to merge close-by clusters shown in Fig. 7.11. The limited flexibility is because the only parameter one can tune to adapt to different event types is the size of the window introduced above.

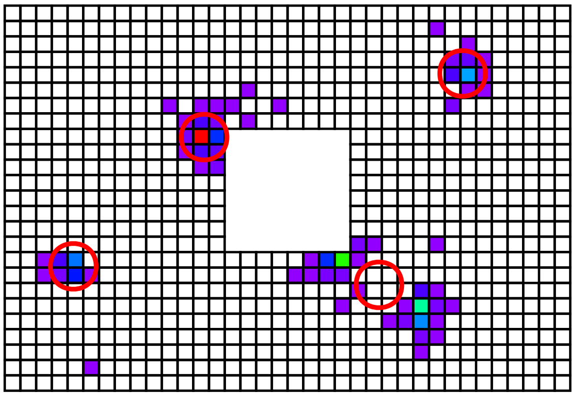


Figure 7.11: A section of an event display, taken from a 5-cluster $K_L \rightarrow 3\pi^0$ candidate event. The circles show the position of the clusters reconstructed offline. The two clusters at the bottom right have been merged. This event will be rejected later by the RMS upper threshold.

The limited flexibility of the offline algorithm does not represent a problem in the offline reconstruction. The events where this clustering algorithm is less efficient, such as the one in Fig. 7.11, are very likely to be rejected by cluster-based quality at later stages of the reconstruction. The $K_L \rightarrow \pi^0 \nu \overline{\nu}$ selection in KOTO's last published $K_L \rightarrow \pi^0 \nu \overline{\nu}$ analysis [31] include:

- all clusters to be separated from each other by a minimum distance of 300 mm.
- the RMS[5] of the energy distribution in all clusters to be higher than 10 (to reject hadronic showers against electromagnetic showers), and smaller than 40 (to reject merged clusters). The RMS cut thresholds are determined with Monte Carlo.
- the energy of all clusters to be above 100 MeV.
- the size of all clusters, to be above four small (2.5×2.5 cm) crystals. This requirement comes from the CsI Moliere radius of 3.57 cm, meaning that 90% of the energy of a photon shower should be contained in 6.4 small crystals.

The HLT clustering algorithm aims to introduce the flexibility needed to efficiently find both photon-like and pion-like clusters. The HLT algorithm also intends to produce clusters

[5] The Cluster RMS is defined as $\sqrt{\sum E_i r_i^2 / \sum E_i}$, where i is each hit channel in the cluster, and r_i is the distance between the cluster center of energy and each hit channel.

that are already on-time and do not require the application of the three criteria listed above. The HLT clustering algorithm is described in the next section.

7.5.2 HLT clustering algorithm

In recent years, the use of GPUs has become more common in HLTs, and different GPU-based clustering algorithms have been developed by experiments. KOTO's online clustering algorithm is based on the CLUE [32] algorithm, a GPU-based clustering algorithm developed by the CMS experiment at CERN. The CLUE algorithm is designed to run online on GPUs on CMS's High Level Trigger. It targets CMS's new High Granularity Calorimeter [33], a 3D calorimeter composed of 50 layers and around 6.5 million readout channels.

KOTO's calorimeter has a single layer with 2716 channels, and very different cluster signatures than the ones found at CMS. However, the working principle of the CLUE algorithm can still be applied to our geometry, and efficiently used in KOTO's HLT.

A note about our target efficiency.

Due to its flexibility and ability to build clusters using both hit energy and position information, the HLT clustering algorithm has the potential to outperform the offline algorithm. However, this can only be shown by testing both algorithms against a simulation, since the true number of clusters in data events is not known.

Requiring simulation events to define the efficiency of the HLT would complicate calculating the efficiency with real data during the 2024 runs. Furthermore, some usable events accepted by the HLT might still be rejected offline by the less efficient offline algorithm. This would defeat the purpose of maximizing the HLT clustering against simulation. Modifications to the offline clustering algorithm were not considered as an option before the 2024 runs, as the offline algorithm is already well understood and validated.

For this reason, the HLT efficiency has been defined by testing its results not against simulation, but against the offline reconstruction results. This approach allows the efficiency to be easily measured and monitored throughout the 2024 runs.

A comparison between the HLT and the offline algorithms against Monte Carlo is presented in Appendix G. In the future, the current offline clustering could be replaced by a CPU version of the CLUE algorithm. Until then, the clustering parameters online will be kept optimized to make the HLT event selection results as close as possible to the offline ones.

The throughput and efficiency results of the online clustering will be discussed in chapter 12. From now on, the algorithm itself will be explained, highlighting some of the strategies that allowed us to take advantage of the parallelism that our GPU offers.

7.5.2.1 Overview of the CLUE algorithm

The adaptation of the CLUE algorithm implemented in the KOTO HLT will be described in the following sections. An overview of its working principle is given here, based on the illustration shown in Fig. 7.12.

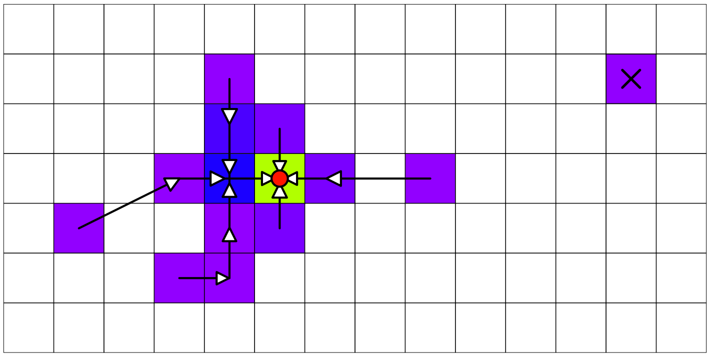


Figure 7.12: A cluster in KOTO's calorimeter, illustrating the HLT clustering process.

The hits shown in Fig. 7.12 are clustered based on the following steps:

1. A *local density* ρ of each crystal is defined as its energy plus the energy of its closest neighbors.
2. Each crystal finds its closest neighbor with higher ρ. This is shown in Fig. 7.12 with arrows.
3. Crystals with ρ above a threshold and no close-by higher-ρ neighbors are defined as cluster seeds (red circle in Fig. 7.12). Crystals with ρ below a threshold and no close-by neighbors with higher ρ are defined as outliers (black cross in Fig. 7.12).
4. Clusters are built from seeds, following the arrows in reverse order. Outliers are considered noise.

The parameters involved in the CLUE algorithm are summarized in Table 7.1. Before physics runs, these parameters are adjusted trigger type by trigger type to maximize the efficiency of the HLT reconstruction. In 2024, only the $K_L \to \pi^0 e^+ e^-$, the 5-cluster $K_L \to 3\pi^0$, and the $K^+ \to \pi^+ \pi^0$ triggers were subject to event reconstruction and selection.

Table 7.1: Parameters involved in the CLUE algorithm.

Parameter	**Description**
d_c	Crystals i and j are considered close neighbours if $\lvert\vec{x}_i - \vec{x}_j\rvert < d_c$.
ρ_c	The density threshold above which a crystal is considered a candidate seed, and below which it is considered a candidate outlier.
d_{seed}	If $\rho_i > \rho_c$ and no higher density neighbour is found within d_{seed}, the crystal i is considered a seed.
d_{outl}	If $\rho_i < \rho_c$ and no higher density neighbour is found within d_{outl}, the crystal i is considered an outlier.

The value of these parameters set for the 2024 runs, aiming to maximize efficiency against the offline algorithm, will be shown in Appendix G, together with the optimal parameters for each trigger type, obtained when maximizing the efficiency against simulation data.

The stages of the CLUE clustering process at KOTO's HLT are detailed in the following sections.

7.5.2.2 Initialization stage

At the HLT initialization stage, and before starting processing events, the clustering structures are initialized and all the needed memory is allocated. No dynamic memory allocations are allowed at runtime, as these operations are in general much slower than any calculation done on the GPU.

Following this, the HLT re-scales the physical 2D coordinates of the CsI crystals from their value in mm to an arbitrary unit that is always positive and integer. This coordinate system, shown in Fig. 7.13 (left), facilitates the indexing of the crystals and reduces the memory needed during clustering. The size of a small crystal in this new grid is 2u×2u, and the size of a large crystal is 4u×4u.

The HLT further divides the CsI surface into sections (hereafter *tiles*), each tile spanning 5×5 small crystals, as shown in Fig. 7.13 (right). Each tile can contain up to 25 hits. The motivation of this division will be given later in section 7.5.2.4.

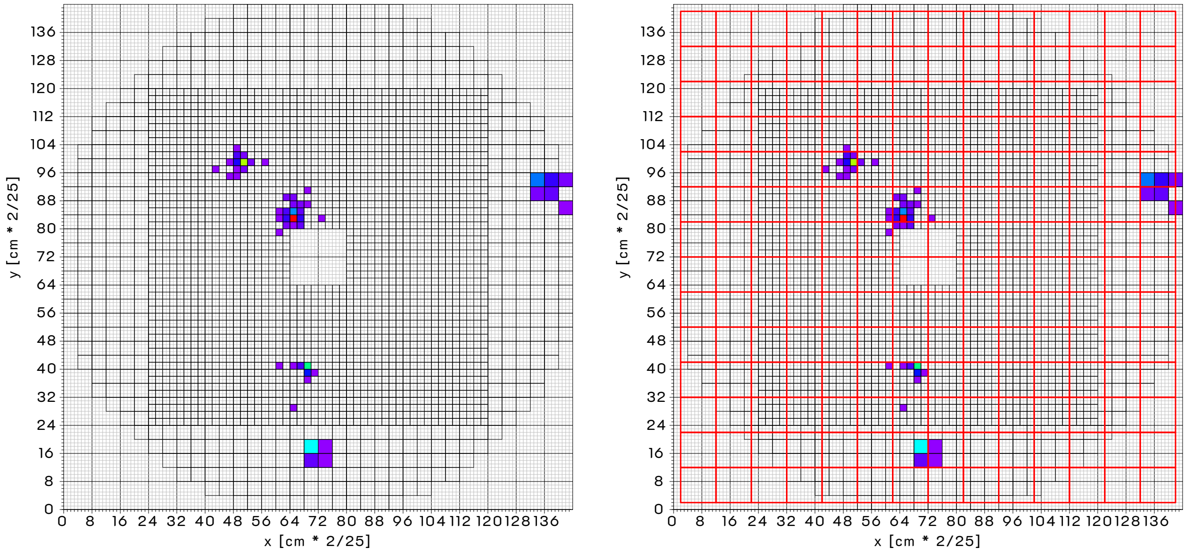


Figure 7.13: Display of an event with 5 clusters on the CsI calorimeter. The HLT coordinate system is shown on the left on top of the calorimeter. The tile grid built on top of the coordinate system is overlaid in red on the right figure.

All processing up to this point happens only once at the initialization stage. Tiles will be filled with hits, and clusters will be built as events begin to arrive from the CPU.

7.5.2.3 Filling tiles with hits

The online clustering starts by selecting the hit channels that will be considered during the clustering process. A *hit channel* is required to have recorded an energy above 3 MeV. The energy reconstruction was detailed in section 7.4.1. Whether a hit channel is on-time or not was determined as explained at the end of section 7.4.2. Rejecting off-time hits at this stage reduces the number of hits that need to be loaded into the tiles. Moreover, it allows the online algorithm to produce already on-time clusters, therefore not needing the further cluster-level timing selection performed offline.

In the following, hit channels will be referred to as *points*. The coordinates of each point are taken at the center of each calorimeter crystal. Event by event, points are assigned a unique index. Linked to that index, the coordinates and energy of each point are recorded. Tiles are filled with the indexes of all points that fall within their boundaries.

All points are added in parallel to their corresponding tiles. However, these operations are performed atomically to avoid race conditions between GPU threads[6]. Note that indepen-

[6] Imagine, for example, two threads attempting at the same time to increment the variable that stores the number of points in a tile. The slightly faster one would first read, let's say, 10. It would then add 1 to it and write back 11. While this happens, the second thread would still read 10, then add 1 to it and write back also 11, not 12. These race conditions can be solved by *atomic operations*, in which any thread accessing a variable locks its address, and no other thread can access it until the lock is released. This defeats the parallelism and thus should be used only when race conditions are expected to occur with a low probability, as is the case here.

dently of the number of hits in the event, the maximum number of iterations of this stage is 25, which corresponds to the rare case in which a tile is completely filled with 25 points. In the example shown in Fig. 7.13, the tile with the most hits contains just 12 points.

7.5.2.4 Local density calculation

The local density of a point is defined as the sum of its energy and the energy of its neighbor points within a given radius d_c. The radius is set to about the expected size of a cluster. With this construction, points belonging to the same cluster are expected to have very similar local densities[7].

When looking for neighbors within a circular radius, requiring $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < d_c$, where i and j are two points, is costly on the GPU. Using a square window instead, $\max(|x_i - x_j|, |y_i - y_j|) < d_c$, is faster, but less accurate as usual clusters tend to be round. The grid structure of KOTO's CsI calorimeter allows to substitute the square root by a more efficient octahedral window, defined by the following two conditions:

$$\begin{aligned} \max(d_x, d_y) &< d_c \\ d_y &< -d_x + \frac{5}{3} d_c \end{aligned} \tag{7.3}$$

where $d_x = |x_i - x_j|$ and $d_y = |y_i - y_j|$. An example of the octahedral window for the case $d_c = 12$u is shown in Fig. 7.14.

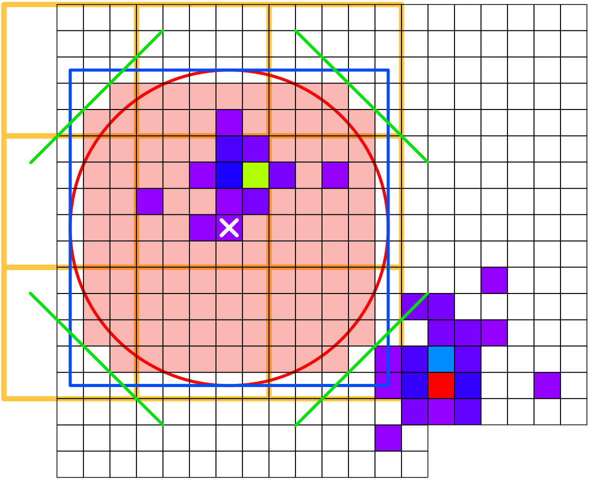


Figure 7.14: The octahedral window used to calculate the local density of the point marked in white. In red, a circle of radius 12u around the point. In blue, a square of side 12u (first condition in equation 7.3). In green, the diagonal lines trimming the corners of the square (second condition in equation 7.3).

[7] This is not the most optimal construction, but it makes the HLT results similar to the results offline. The actual parameters used at the HLT, together with the true *optimal* parameters for each trigger type, are shown in Appendix G.

The merit of the tile grid.

When looking for a close-by neighbor, each point retrieves first from the tile grid the tiles touched by a square window of side $2d_c$. In Fig. 7.14, this window is highlighted blue, and the nine tiles touched by it are highlighted orange. A loop through the points inside those tiles is still needed to retrieve the points that satisfy the octahedral window condition. The number of points contained in these tiles is in general small, and in particular much smaller than the total amount of hit channels per event. In the worst-case scenario, when all channels have been hit, each point needs to loop just through the 117 channels highlighted in red in Fig. 7.14. This is to be compared to the 2716 channels that would need to be looped through in the offline algorithm.

Since the local density calculation is independent for each point, a GPU thread is assigned to each point and the process described above is performed in parallel for all hit points.

7.5.2.5 Distance to closest higher

In this stage, each point will be connected to the most energetic point of its cluster through a path that goes through points in ascending order of density. The point with the highest density in a cluster will be defined as the *seed* of the cluster.

The *distance to the closest higher* is defined as the distance to the closest point with higher density. If multiple closest points have the same density, which will happen often in KOTO's calorimeter, the one with higher energy is selected. Every point is added to a list of *followers* of its closest higher. In Fig. 7.12, the arrows illustrate the chain of *closest highers*, from the less to the most energetic hits of the cluster.

If a point has a local density higher than a threshold ρ_c and does not have any neighbors with higher density within a radius d_{seed}, the point is promoted to *seed*. One seed was marked with a red circle in Fig. 7.12.

Conversely, if a point has a local density lower than ρ_c and no higher-density neighbor within d_{outl}, the point is demoted to *outlier*. One outlier was marked with a black cross in Fig. 7.12.

Note that in our construction, d_c is comparable to the size of the clusters, and the local density ρ of the seed of a cluster will give a fair measure of the total cluster energy. All clusters obtained by this method will have by construction an energy larger than ρ_c.

7.5.2.6 Clusters from points and events from clusters

Finally, clusters are reconstructed from seeds. Seeds' first-order followers are first added to the cluster, and then sequentially the followers of the followers, until the cluster is complete. This is equivalent to following the arrows in Fig. 7.12 in reverse order. Cluster E is calculated at this stage as the sum of the energy of all clusters. The cluster Center of Energy (COE) is also calculated as follows:

$$\mathrm{COE} = \frac{\sum E_i \, \vec{x}_i}{\sum E_i} \; , \tag{7.4}$$

where i is each point in a cluster. Event Total E and COE^2 are eventually calculated from each cluster's energies and COEs. With this, the event has been reconstructed and it is ready for the selection that follows.

7.6 Event selection

The calorimeter-related[8] *offline* selection at KOTO consists of two stages. In the first one, events are rejected if they do not satisfy minimum quality requirements on calorimeter-related variables. In the second stage, complex analysis-specific selection is applied to look for specific signatures or to reduce specific backgrounds.

The quality selection at the first stage mainly aims to reject events in which part of the energy is likely to have been deposited outside the calorimeter, either because a particle escaped through the beam hole, or because part of a hit energy escaped through the edges of the calorimeter. For this purpose, all clusters are required to be within the boundaries outlined in Fig. 7.15, and a minimum total energy deposition of 600 MeV is required in the calorimeter. More formally, thresholds are imposed on the following variables:

- Max R, the distance between the center of the calorimeter and the cluster furthest from it.
- Min XY, or the lowest $\max(|x|_{\mathrm{cluster}}, |y|_{\mathrm{cluster}})$ of all clusters in an event. The quantity $\min XY$ represents a measure of distance between the edges of the square beam hole and the cluster closest to it.
- Total E, or the sum of the energy of all clusters recorded on the CsI.

In Fig. 7.15, the black lines delimit the region satisfying $\min XY > 150$ mm and $\max R <$ 850 mm.

[8] we exclude from the explanation the offline event selection based on veto detector information.

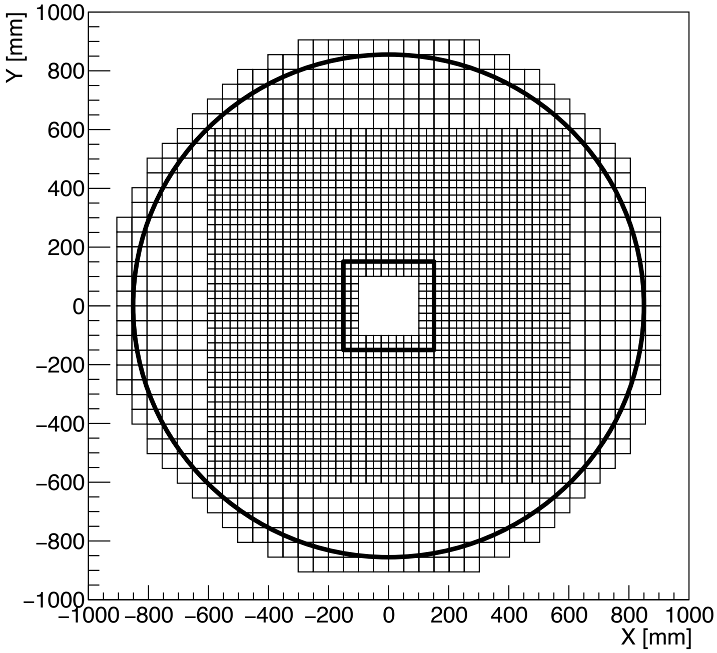


Figure 7.15: Events are rejected if they contain clusters whose COE is reconstructed inside the inner black square ("Min XY"), or outside the black circle ("Max R"). In both cases, it is likely that part of the hit energy was not fully deposited in the calorimeter.

The Min XY, Max R, and Total E criteria used offline and in 2024 at the HLT are summarized in Table 7.2. The HLT thresholds will be justified in chapter 10.

Table 7.2: Quality cuts applied offline and at the HLT for each of the three triggers reconstructed in 2024.

	5γ		K^+		$K_L \to \pi^0 e^+ e^-$	
	Offline	HLT	Offline	HLT	Offline	HLT
Min XY	>150 mm	>140 mm	>150 mm	>140 mm	>150 mm	>140 mm
Max R	<850 mm	<865 mm	<850 mm	<850 mm	<850 mm	<870 mm
Total E	>600 MeV	-	>600 MeV	>600 MeV	>600 MeV	>600 MeV

The second stage is specific to every trigger mode and every physics analysis, and it is performed after complete event reconstruction. It is purposed to reject background events (e.g. reject $K_L \to \pi^+\pi^-\pi^0$ in a $K_L \to \pi^0 e^+ e^-$ search), or to select signal events with specific detector signatures.

As a general rule, tight cuts are performed offline, after a precise calibration and reconstruction that is not feasible online. The online trigger applies looser cuts, with the main goal of not introducing a bias in the data, and prioritizing efficiency over rejection power. Accepting *bad* events that will be later rejected offline is not a problem and does not affect offline physics results. Rejecting *good* events that would have been later accepted offline leads to a loss of valuable data, and if the effect is big enough, to a bias in the data that would need to be accounted for in the offline analysis. This situation needs to be avoided.

In 2024, only loose quality cuts based on min XY, max R, and total E were applied at the HLT. The COE was also reconstructed event by event, and a COE cut was ready to be applied to both the 5-cluster $K_L \to 3\pi^0$ and the $K_L \to \pi^0 e^+ e^-$ events. However, data reduction from pedestal suppression and waveform compression (Chapter 8) was enough in 2024 to reduce the data rate below requirements, and the COE cut was not applied in most physics runs. The motivation of the COE cut in the 5-cluster $K_L \to 3\pi^0$ and the $K_L \to \pi^0 e^+ e^-$ triggers, as well as its rejection power and efficiency, are discussed in Section 10.2.

Storing selection results

The results of the selection event by event are added to the HLT event tag, introduced in section 7.2. The event tag will eventually become part of the event header, which is kept even if the event is rejected.

More specifically, the second byte of the HLT tag is reserved for the selection results of every cut. The conditions for each of its bits to be enabled are described in table 7.3.

Table 7.3: Contents of the second part of the HLT event tag. Bits 13 to 15 are currently unused.

Bit	Enabled if
8	The event was randomly selected to be a HLT-minimum-bias event.
9	The event did not pass the Min XY cut
10	The event did not pass the Max R cut
11	The event did not pass the Total E cut
12	The event did not pass the COE cut
13 14 15	

All selection criteria are tested at the HLT. Even if an event does not pass the min XY cut, and is therefore set to be rejected, all other variables up to the COE will still be reconstructed and tested. This allows to monitor in real time the rejection power of each of the HLT cuts, and it is possible thanks to the negligible time[9] that reconstructing these variables takes at the HLT.

[9] The throughput of the HLT will be evaluated in section 11.2.

Chapter 8

Pedestal Suppression and Waveform Compression

Pedestal suppression and waveform compression are the last stages of the HLT processing and aim to reduce the size of the events that have been selected. The pedestal suppression consists in identifying detector channels without hits and discarding their waveforms, as they do not contain relevant information. The lossless waveform compression further reduces the size of the accepted waveforms without losing information.

Both pedestal suppression and waveform compression were performed in KOTO before the HLT was installed. Pedestal suppression was done at the ADCs, based on the distance between waveform peaks and each channel's pedestal value[1]. The HLT implements a new pedestal suppression strategy based on the energy and peak height of each waveform. The HLT lossless compression algorithm is kept as the original and has just been adapted to run efficiently on the HLT's GPUs. The pedestal suppression and waveform compression, as currently done on the HLT, are described in this chapter. The resulting data reduction is shown at the end of the chapter, in section 8.5. A more powerful lossless compression algorithm, not utilized in 2024 at the HLT, is presented in section 13.2.1.

8.1 Pedestal suppression

Motivation

The number of hit channels[2] in the CsI calorimeter in a physics event is typically fewer than 150 (Fig. 8.1). The waveforms of the remaining channels are just noise, and do not contain meaningful information. These waveforms can be suppressed, significantly reducing the size of an event. This translates into lower bandwidth and lower offline storage requirements, which in turn increases the maximum event rate that the DAQ can handle.

[1] The ADC pedestal suppression is briefly described in chapter 3.3.3 of Ref. [23].

[2] hit channels are those which recorded an energy deposition higher than 3 MeV.

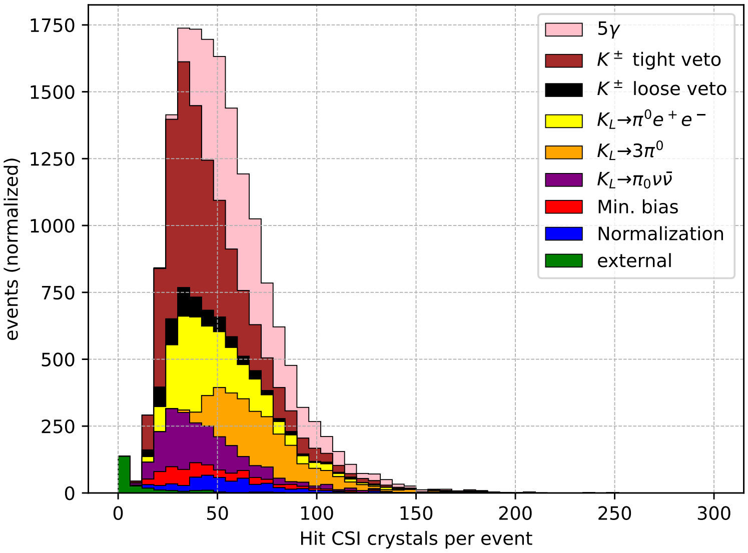


Figure 8.1: Typical number of hit CsI channels per event during a physics run including different triggers.

The pedestal suppression (PS) stage consists in identifying and tagging these non-hit channels. Their waveforms are then discarded, and only their pedestal value is kept. The HLT uses the online pedestal calculated at the L1 stage, as the average "ADC count" obtained spill by spill during beam-off. This pedestal is stored as the first sample of every waveform. A sample output from a hit and a non-hit channel is shown in Fig. 8.2.

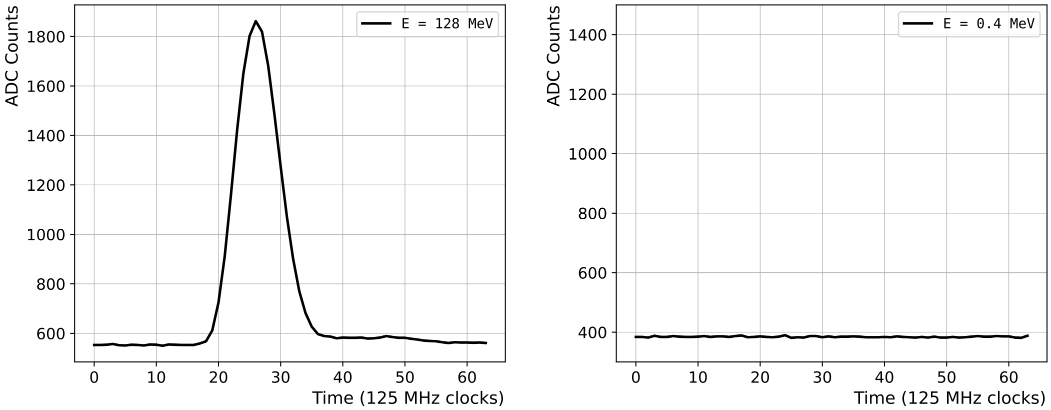


Figure 8.2: Sample waveform from a hit channel (left) and a non-hit channel (right) on the CsI calorimeter.

Flat waveforms are identified online based on their energy and peak height. The energy is calculated by applying a conversion factor to the pedestal-subtracted integrated ADC. The peak height is defined as the distance in ADC counts between the highest sample and the pedestal.

In some cases, like the one shown in Fig. 8.3, an overestimation of the online pedestal even by a few counts can significantly reduce the integrated ADC of a waveform, since all samples at the level of the real pedestal will now contribute negatively to the integrated ADC. The peak height, much larger than possible pedestal fluctuations, is used to prevent these waveforms from being suppressed. Concrete thresholds will be given later in this section.

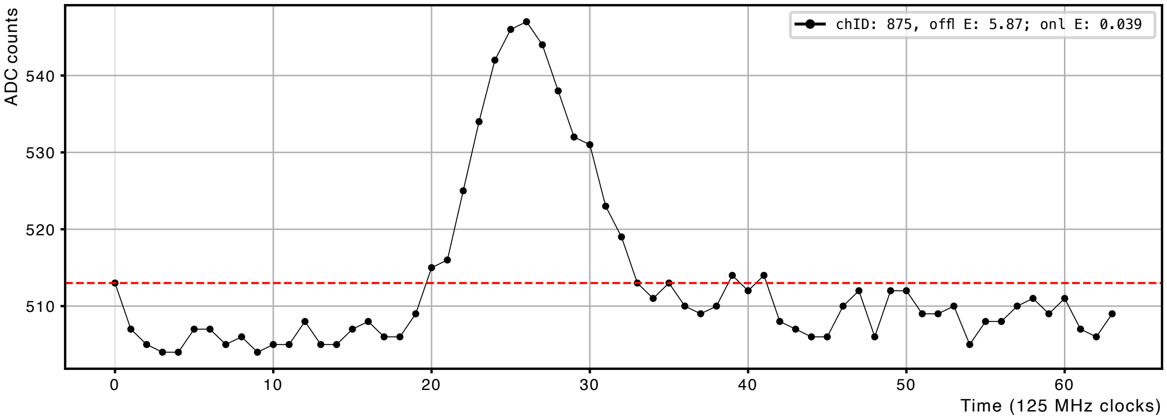


Figure 8.3: Due to the online pedestal (stored as the first value of the waveform) being overestimated, the integrated ADC of this waveform, and therefore its energy, become very close to zero. The peak height, which is less affected by pedestal fluctuations, prevents the waveform from being suppressed.

The online pedestal suppression inefficiency is defined by taking the waveforms that are not suppressed offline, and counting how many of them are suppressed online, as shown in Eq. 8.1.

$$\text{Pedestal suppression inefficiency} = \frac{\text{waveforms accepted offline and suppressed online}}{\text{waveforms accepted offline}} \tag{8.1}$$

In the *offline* reconstruction, for each waveform, the mean of samples 1 to 10 and 54 to 63 is calculated, and the value with the lowest standard deviation is taken as the pedestal. Then, the energy is calculated as done online (Eq. 7.2), and waveforms with less than 3 MeV are suppressed.

In the *online* reconstruction, waveforms of veto detector channels are never suppressed, and waveforms from the main $K_L \to \pi^0 \nu \overline{\nu}$ triggers are not suppressed either. Low-gain CsI channels are also masked to prevent the suppression of waveforms with actual hits close to the noise level, as will be discussed in the next section.

The online suppression criteria are determined with a sample of waveforms from a physics run at the beginning of the 2024 beam-time, before enabling the pedestal suppression at the HLT. The waveform energy is calculated according to Eq. 7.2 in the same way the HLT does online. The online peak height is also obtained using the online pedestal, as done at the HLT.

The upper and lower energy thresholds for a waveform to be suppressed online are determined independently. First, as shown in Fig. 8.4 (left) the upper threshold is fixed at 0, and the lower threshold is varied, aiming for low inefficiency while keeping a reasonable suppression power. Then, the process is repeated fixing the lower threshold at 0 and varying the upper threshold (Fig. 8.4, right). No peak height requirements or channel masking are applied in this stage.

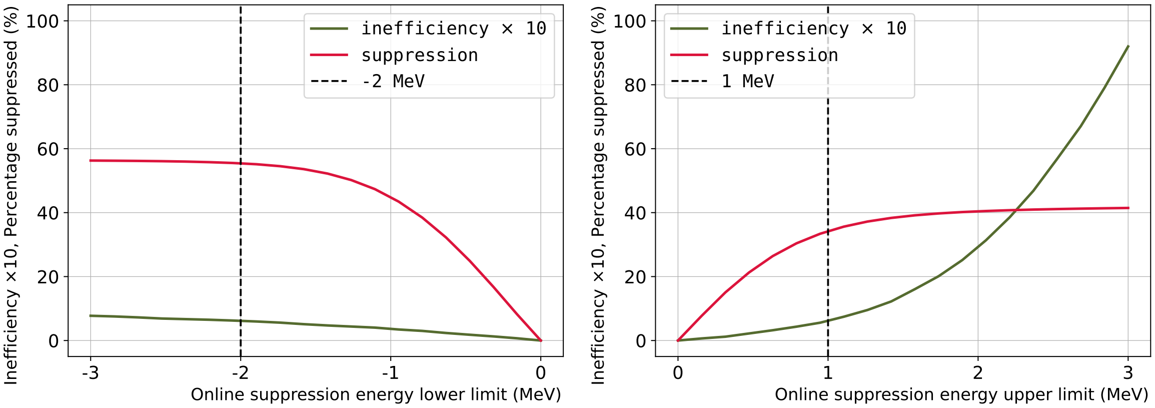


Figure 8.4: Pedestal suppression inefficiency and percentage of CsI channels suppressed as a function of the low energy threshold (left) while keeping the higher threshold at 0 MeV, and the high energy threshold (right), while keeping the lower threshold at 0 MeV. No channel masking or peak height requirements are applied. Inefficiency is scaled by a factor of 10 for a better visualization.

From the results in Fig. 8.4, the online energy window for a waveform to be suppressed is chosen to be from -2 MeV to < 1 MeV. On top of the energy requirement, the peak height is required to be below 10 counts for a waveform to be suppressed. For an average-gain CsI channel, 10 counts roughly corresponds to 1 MeV. The minimum energy requirement, which is not imposed offline, is applied online to prevent the suppression waveforms where the pedestal was overestimated and the peak height did not reach 10 counts.

With just the "-2 MeV $<$ E $<$ 1 MeV" requirement, the inefficiency of the online pedestal suppression as defined in Eq. 8.1 is 1.36%, and the average percentage of suppressed channels in a physics run is 90%. After imposing the "peak height $<$ 10 counts" requirement, the inefficiency is reduced to 1.20%, and the percentage of suppressed channels is 87%. The final results of inefficiency and the nature of inefficient waveforms will be discussed in section 8.3, after treating the effect of low-gain channel masking.

8.2 Low gain CsI channels

The *gain* of a CsI channel in the HLT context is defined as the number of integrated ADC counts that correspond to an energy deposition of 1 MeV. The online gain is estimated channel by channel with cosmic muons at the beginning of each run. Offline, more precise coefficients are calculated from π^0 decays to photons in $K_L \to 3\pi^0$ events, and from dedicated aluminum target runs, described in section 3.3.

CsI channels with low gain have smaller peak-to-noise ratios. If the pedestal is overestimated, the online energy can easily fall within the suppression range. The peak height check is also likely to fail if the actual hit energy is small and the gain of the channel is low. An example of this situation is shown in Fig. 8.5.

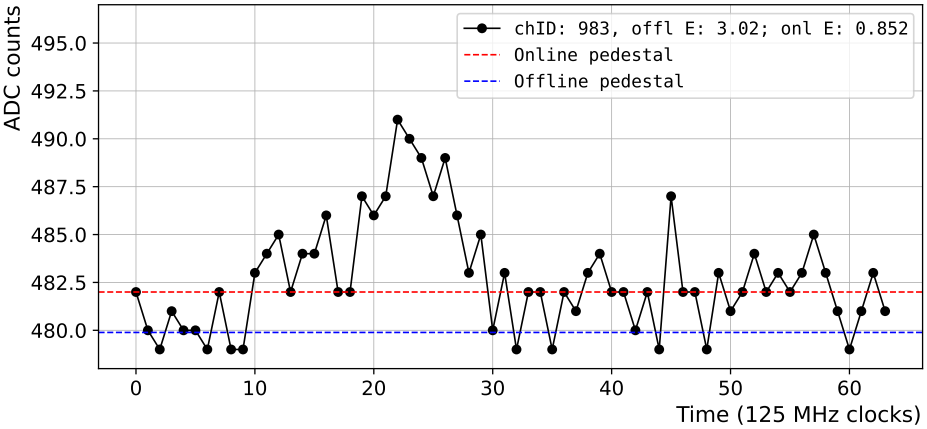


Figure 8.5: Waveform with a potential low-energy hit from a channel with low gain. The online-calculated energy is 0.852 MeV. Its peak height is 9 counts.

To measure the relation between inefficiency and gain for all CsI channels, the HLT is configured to tag the waveforms that fall within the suppression criteria, without actually suppressing them. All events can then be reconstructed offline as normal. The data presented in this section was taken in these conditions at the beginning of beam-time, during an actual physics run. The relation between the gain of a CsI channel and the pedestal suppression inefficiency is shown in Fig. 8.6 (left). The red dotted line marks the threshold below which channels are masked online. The average inefficiency of channels with gain below this threshold is 2.6%, and the average inefficiency of channels with gain above this threshold is 0.09%. The right plot in Fig. 8.6 shows the gain distribution of the CsI channels.

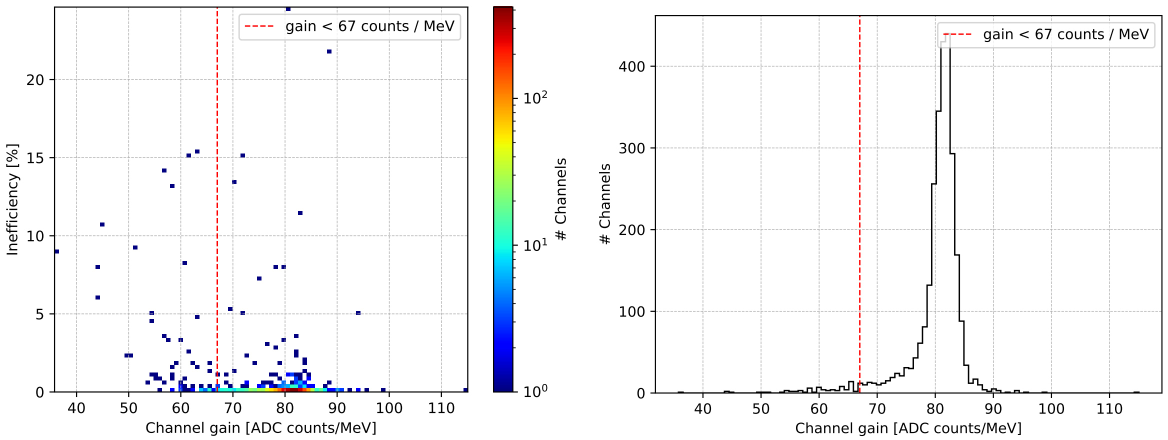


Figure 8.6: Left: Relation between the inefficiency of the online pedestal suppression and the CsI channel gain. Right: gain distribution of the CsI channels. All channels below the red dotted line are masked online. Both plots contain one entry per CsI channel, 2716 in total.

After this study, the list of channel IDs to be masked is recorded in the HLT configuration. Tagged data is taken again, and online results based on the HLT tag are compared to the offline results to get a realistic measurement of the inefficiency of the online pedestal suppression. The results of this measurement are shown in the next section. The nature of channels with normal gain but high inefficiency is also discussed in the next section.

8.3 Inefficiency measurement during actual physics runs

The pedestal suppression inefficiency, after all suppression criteria have been fixed, is measured with tagged data during actual physics runs. Figure 8.7 shows the pedestal suppression inefficiency of each CsI channel, computed using a sample of waveforms from triggers with pedestal suppression enabled. Masked low-gain channels are shown in red. Waveforms from masked channels are never suppressed online so they do not contribute to the inefficiency.

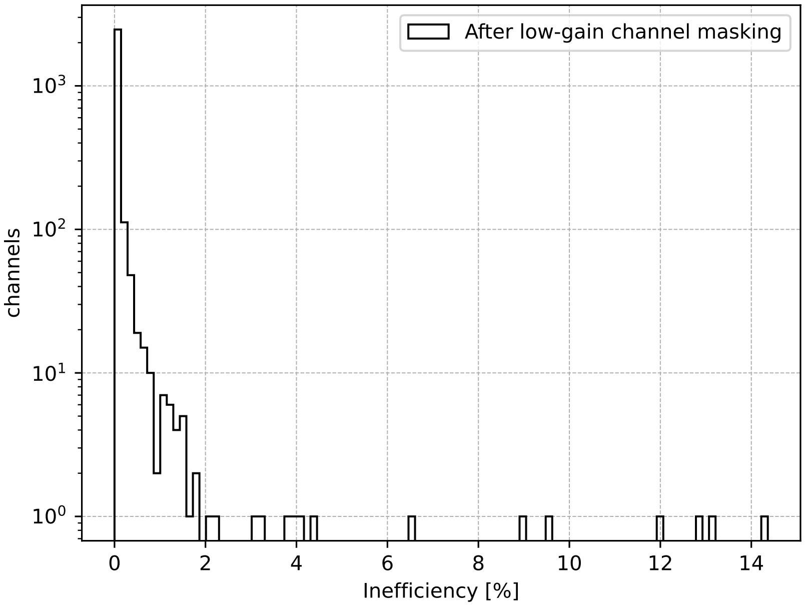


Figure 8.7: Pedestal suppression inefficiency distribution. The inefficiency of masked channels is 0, as they are never suppressed.

The average channel inefficiency of the online pedestal suppression after masking low gain channels is 0.09%. A randomly selected sample of inefficient waveforms is shown in Appendix A. The vast majority of these waveforms are noise-like waveforms where the *offline* pedestal was underestimated, leading to a higher integrated ADC calculated offline.

Inefficient channels

Even though the average inefficiency is low, some channels in Fig. 8.7 can be seen to have a very high inefficiency. The gain of these channels is not low, and thus these channels are not masked. Nevertheless, the reason why their inefficiency is large has been studied individually. The four channels observed in Fig. 8.7 with inefficiency larger than 10% are studied in detail in Appendix I.

The suppression power can be measured by counting the number of times each channel has been suppressed in real runs. The result is shown in Fig. 8.8.

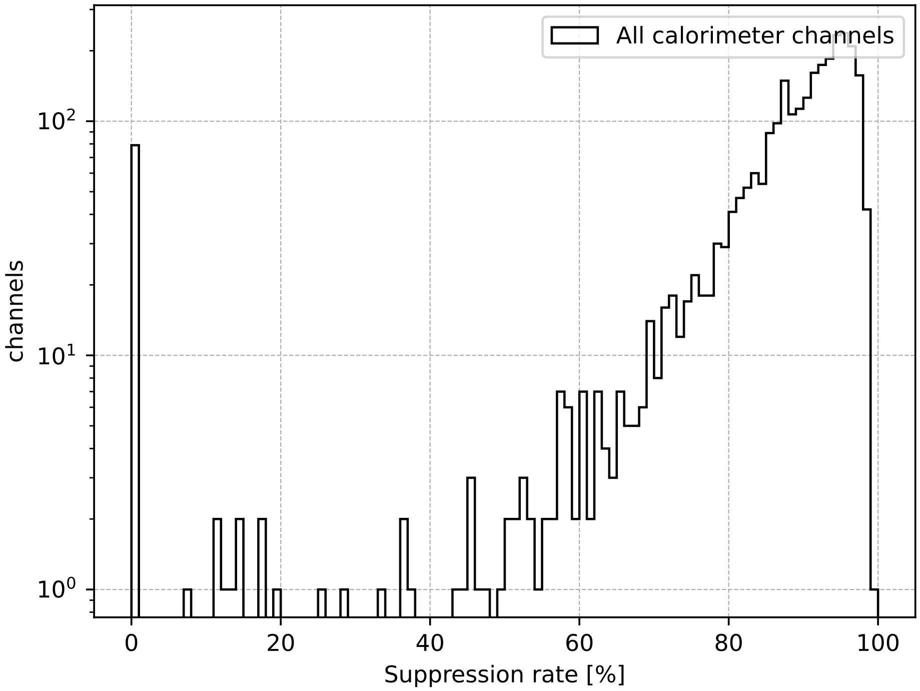


Figure 8.8: Number of times each CsI channel has been suppressed in a normal physics run, shown in percentage. Masked channels are never suppressed.

The channel pedestal suppression rate during physics runs largely depends on the location of each channel and the trigger type. The higher the hit rate, the lower the suppression rate. In average, the suppression rate during physics runs considering data from triggers with pedestal suppression enabled and including masked channels, is $(86.24 \pm 0.03)\%$. The data reduction factor linked to this value will be discussed in section 8.5.

8.4 Lossless waveform compression

Motivation

The lossless waveform compression provides compression factors larger than 3 without the loss of any information. It is a powerful data reduction tool that is applied to all triggers, including the main $K_L \to \pi^0 \nu \overline{\nu}$ trigger. Avoiding the loss of information is required, particularly in low-energy waveforms that could impact the veto efficiency in the offline analysis.

Its working principle can be reduced to a series of simple operations (additions, subtractions, and bit-level operations) that can be applied in parallel to all waveforms, making the compression algorithm suitable for GPU parallelization.

Waveform compression is the last stage of the online event processing. Its working principle is discussed in this section, followed by its implementation on GPU.

8.4.1 Compression principle

The dynamic range of KOTO's ADCs is 14 bits. In other words, analog waveforms are digitized into arrays where each sample can range from 0 to $2^{14} - 1 = 16383$ ADC counts. However, most waveforms have the pedestal at a few hundred counts and maximum amplitudes on the order of a few hundred counts. A representative waveform from a hit channel is shown in Fig. 8.9. The pedestal and amplitude distributions of waveforms from a physics run without pedestal suppression applied are shown in Fig. 8.10.

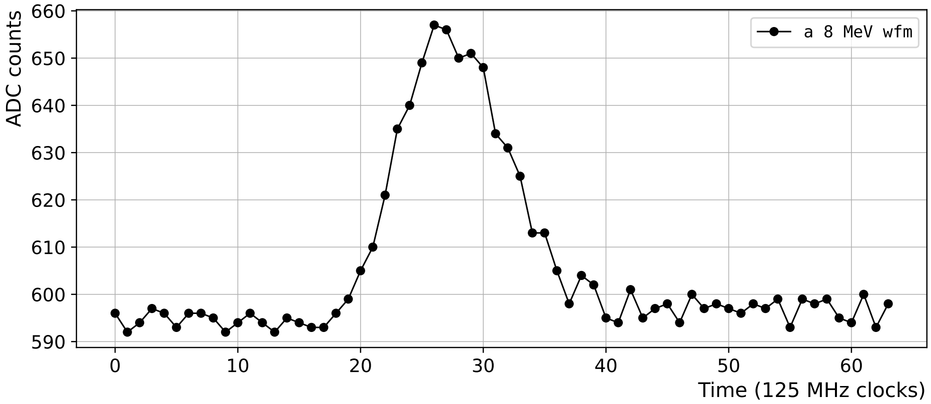


Figure 8.9: A waveform from a hit crystal on the CsI, corresponding to an 8 MeV hit. The amplitude of this waveform is 65 counts.

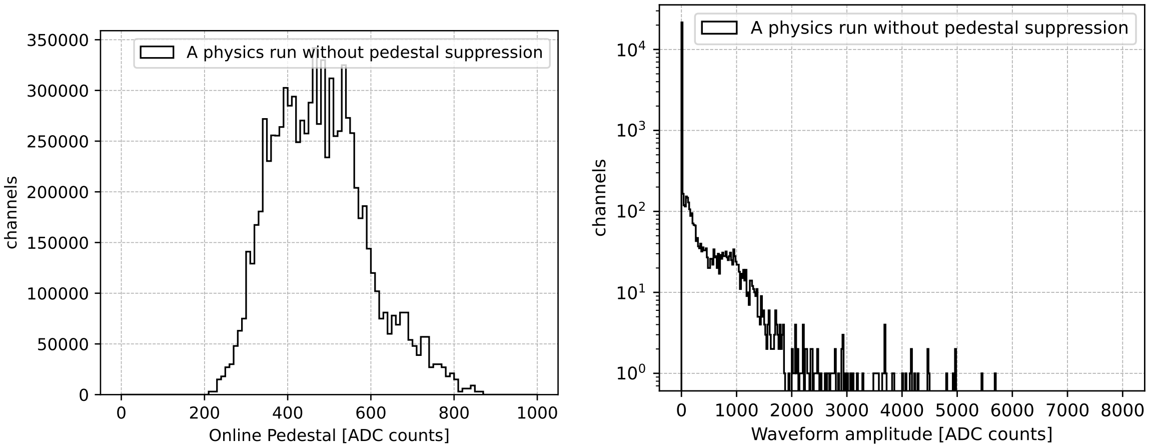


Figure 8.10: Online pedestal (left) and waveform amplitude (right) distribution of waveforms from a physics run without pedestal suppression applied.

Most waveforms are not large enough to make use of the full 14-bit dynamic range of the ADCs, as will be shown in Fig. 8.13. On top of that, the ADCs record each waveform sample into a 16-bit unsigned integer to simplify the data handling, leaving the two most significant bits unused. The 8 MeV waveform in Fig. 8.9 has an amplitude of just 65 counts, meaning that after subtracting its minimum, it could be contained in just $\log_2(65) \approx 7$ bits. This would reduce its size by around 55%. Figure 8.11 illustrates the compression principle.

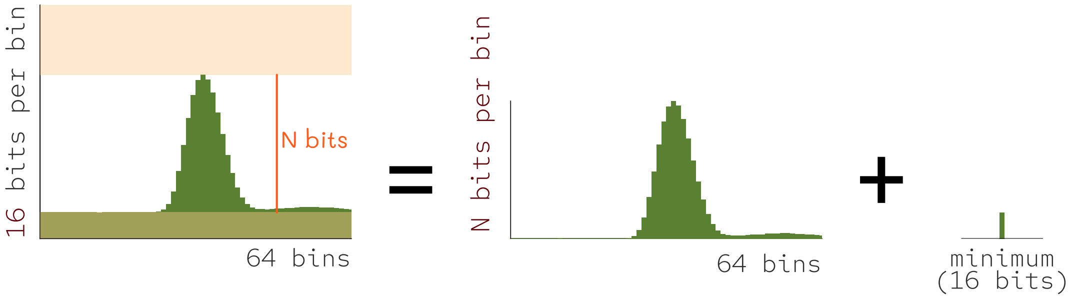


Figure 8.11: Compression principle. The amplitude of every waveform (the number of bits per sample N_{bps} needed to contain the difference between the highest and lowest sample) is calculated. The waveform is then compressed into $N_{bps} \times 64$ bits. Its minimum value is also recorded, as it is needed to decompress the waveform.

Compressed waveforms are stored in 8-bit integer arrays. The waveform data is preceded by a 3-byte header, whose contents are shown in Fig. 8.12. The first (leftmost) header byte stores the number N_{bps} of bits needed to contain each sample of the compressed waveform ($N_{bps} \in [0, 16]$). This number is substituted by a tag if the waveform was pedestal-suppressed. The two most significant bits of the second byte are reserved to indicate whether the HLT was running in tagging mode. When running in tagging mode, waveforms that satisfy the criteria to be suppressed are tagged, but not suppressed. The remaining 14 bits (blue in Fig.8.12) contain the waveform minimum, needed to reconstruct the waveform offline. For pedestal-suppressed waveforms, this region stores the waveform pedestal.

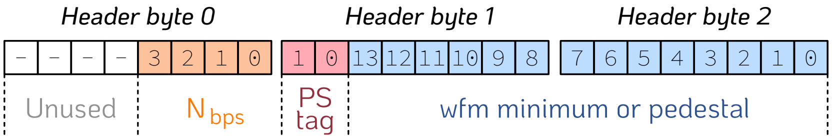


Figure 8.12: Structure of the 3-byte waveform headers. For pedestal-suppressed waveforms, the last 14 bits record the pedestal value. Otherwise, they record the waveform minimum, needed offline to reconstruct the waveform.

Following the header, non-suppressed waveforms are recorded in the next 64 samples $\times N_{bps}$ bits. The size of the compressed waveforms in bytes can therefore be calculated as follows:

$$\text{Size of compressed waveform} = 3\ [\text{bytes}] + 64\ [\text{samples}] \cdot \frac{N_{bps}\ [\text{bits/sample}]}{8\ [\text{bits/byte}]} \tag{8.2}$$

The only exception to the format explained above is the waveforms with N_{bps} larger than 8. Following the original compression algorithm[3], compression is not performed in this case.

[3] A more powerful compression algorithm without this limitation, subject to be commissioned in future runs is introduced in section 13.2.1

Instead, a one-byte flag is used in place of the 3-byte header above, and the uncompressed waveform is stored in the following 128 bytes.

For perfectly flat or pedestal-suppressed waveforms, N_{bps} is 0, and the compressed waveform is stored in 3 bytes. This gives a size reduction of 98% with respect to the original waveform. If the difference between the highest and lowest sample was 1, N_{bps} would be 1, the size of the compressed waveform 11 bytes, and the size reduction 91%. The compressed size is shown as a function of N_{bps} for all values of N_{bps} in Fig. 8.13. The distribution of N_{bps} from waveforms from a physics run is overlaid.

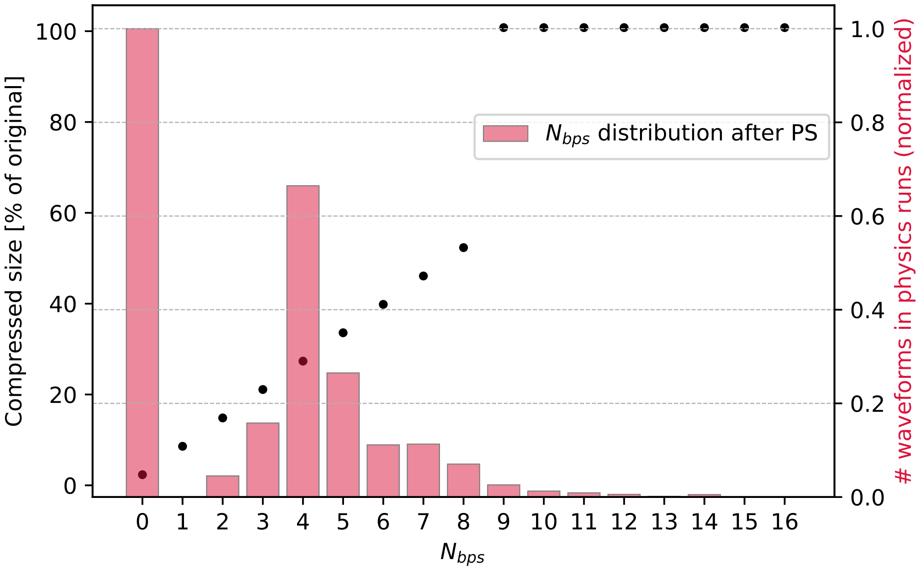


Figure 8.13: Compressed waveform size as a function of the original, for all possible values of N_{bps} (black). In red, the distribution of N_{bps} from waveforms from a physics run. The bin at 0 comes from pedestal-suppressed waveforms.

The implementation of the compression algorithm on GPU, which has been entirely designed and implemented by the author of this thesis, is discussed in the following section.

8.4.2 Implementation on GPU

On average during physics runs, around $\sim 1.5 \times 10^4$ events every spill pass the online event selection at the HLT. During special calibration runs, this number grew up to 2.5×10^4 in 2024. In total, each event consists of 4608 waveforms, making a total of almost 70 million waveforms that have to be compressed per spill. These requirements, together with the rest of the HLT tasks, cannot be satisfied on CPUs in the current HLT.

GPUs on the other side are very well suited for this task. Properly implemented, the waveform compression can be performed in parallel at the event level, the ADC channel level, and even at the sample level inside a waveform. However, the parallelism offered by the GPU also introduces some challenges that increase the complexity of the algorithm, most of them being a consequence of the GPU parallelism itself.

The largest challenge comes from the fact that compressed waveforms from multiple events have to be written together into the array, to be sent to the next stage of the HLT. On a serial CPU implementation, each waveform would be compressed and simply written after the previous one. On GPU, however, multiple waveforms are compressed in parallel. The index of each waveform in the output array, which is equal to the size of all previously compressed waveforms, needs to be known in advance.

To solve this issue while keeping the performance high, the compression is implemented in four stages. In the first stage, the compressed size per waveform is calculated. In the second and third stages, the index in the compressed array where each compressed waveform will be written is calculated. Finally, in the fourth stage, waveforms are compressed and written into the compressed array. The following sections describe the working principle of each stage.

Stage 1: Calculation of the compressed size per waveform

The compressed size of each waveform can be calculated according to Eq. 8.2 without performing the actual compression. For pedestal-suppressed waveforms, $N_{bps} = 0$, and the compressed size is trivially 3 bytes. Non-suppressed waveforms require the calculation of N_{bps}, for which the maximum and minimum samples of the waveform need first to be found.

Each event block built at the HLT contains N_{evts} events, typically with $N_{\mathrm{evts}} \leq 100$. Each event contains data from a total of 288 ADCs. A 2D block grid with dimensions $N_{\mathrm{adcs}} \times N_{\mathrm{evts}} = 288 \times N_{\mathrm{evts}}$ is launched. Each thread block targets a single ADC from a single event. Each block is launched with 256 threads, so that each of the 16 waveforms per ADC can be processed by 16 threads.

As a first step, the maximum and minimum sample of each waveform are found as illustrated in Fig. 8.14; Each of the 16 threads per 64-sample waveform computes first the minimum and maximum of the four samples it takes care of. Threads are synchronized at this point, and an atomic comparison[4] is performed to calculate the global minimum and maximum of the waveform.

[4] Atomic operations are those in which the memory location of a shared variable is locked while a thread reads or writes it, to prevent multiple threads from reading or writing it at the same time. See the footnote in section 7.5.2.3 for a more detailed explanation.

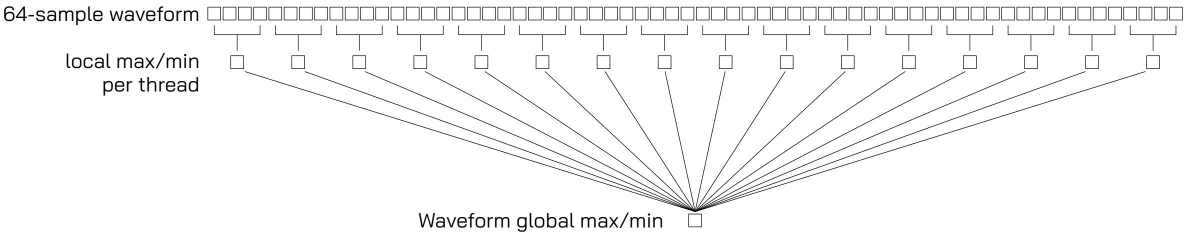


Figure 8.14: The process of finding the maximum and minimum of a waveform on GPU. The local max/min per thread is computed in four iterations. The waveform global max/min is then calculated atomically.

Minimum and maximum are used to calculate N_{bps}. The number of compressed bytes per waveform $N_{bps} \times 64$ is then calculated, and written into an array in global GPU memory. Minimum and N_{bps} are also recorded for each waveform, as they will be needed in the compression stage.

Stage 2: Calculation of waveform index in the compressed array, relative to the beginning of each crate.

As introduced in section 4.2, the KOTO DAQ system consists of 18 crates, hosting 16 ADCs each. Each 16 records 16 waveforms per event, making a total of 256 waveforms per crate. The compressed size per waveform is already known at this stage. Now, the distance in bytes between the beginning of a compressed waveform and the beginning of its local crate is calculated according to Eq. 8.3, as the sum of the size of all previous waveforms in the crate.

$$\mathrm{idx}_i = \sum_{j=0}^{i-1} \mathrm{size}_j \tag{8.3}$$

A 2D grid consisting of $N_{\mathrm{crates}} \times N_{\mathrm{evts}} = 18 \times N_{\mathrm{evts}}$ blocks is launched at this stage. Each block treats a single ADC crate from a single event. Each block consists of 256 threads, and one thread is assigned to a single ADC channel (16 ADCs per crate and 16 channels per ADC = 256 channels). Figure 8.15 illustrates the outcome of this stage. The compressed size per waveform is known for all waveforms (left). The output index with respect to the beginning of the crate is obtained for all crates independently (right), according to Eq. 8.3.

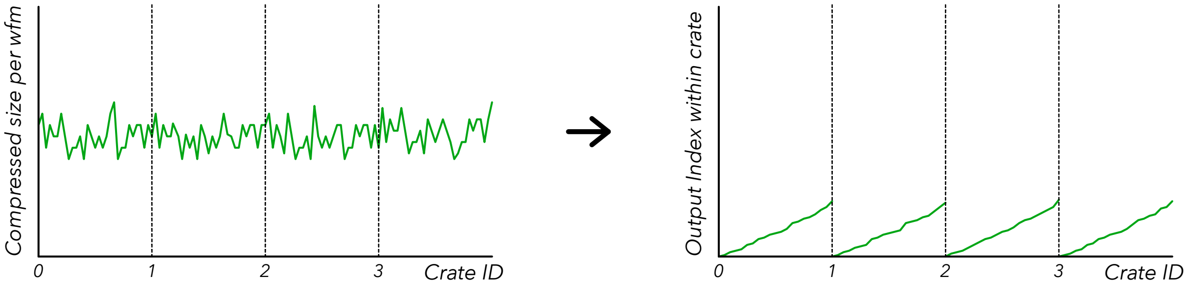


Figure 8.15: Input and output of the second stage of the compression algorithm, illustrated with just 4 crates. The output index per ADC channel corresponds to the number of bytes that all previous waveforms in the crate will occupy in the compressed array. The compressed size per waveform is known from the previous stage.

The compressed bytes of all 256 waveforms per crate are first retrieved from global memory to shared memory for faster access. To calculate the index of each compressed waveform relative to the beginning of the crate, one needs to know the compressed size of all the waveforms before it. An efficient way to solve this in parallel is by implementing a *parallel prefix sum* algorithm, whose working principle is illustrated in Fig. 8.16.

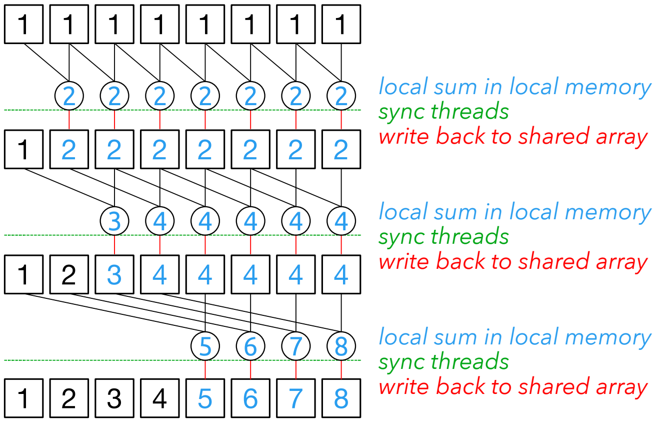

1 1 1 1 1 1 1 1
2 2 2 2 2 2 2
local sum in local memory
sync threads
write back to shared array
1 2 2 2 2 2 2 2
3 4 4 4 4 4
local sum in local memory
sync threads
write back to shared array
1 2 3 4 4 4 4 4
5 6 7 8
local sum in local memory
sync threads
write back to shared array
1 2 3 4 5 6 7 8

Figure 8.16: Working principle of the parallel prefix sum algorithm, exemplified with an array of 8 elements. The same array is re-used in every iteration. The numbers in the squares represent the contents of the array each iteration.

Fig. 8.16 shows a simplified case in which the size of all waveforms is "1". Each thread is assigned a waveform. In the first iteration, each thread adds up in local memory the size of its compressed waveform to the size of the one before it. Then, the result is copied back to the shared array. Note that since the original array is overwritten every iteration, threads need to be synchronized before writing their local results back to the shared array. Otherwise, thread i could write its result to index i of the shared array before thread $i+1$ has read it to perform its local sum.

The parallel prefix sum algorithm explained above performs in $\log_2(N)$ iterations what would have taken N in a serial implementation. In our case, this operation is performed in parallel to all events and to all 18 crates per event containing 256 waveforms each, in just

$log_2(256) = 8$ iterations.

The shared array with the results is finally copied back to global memory, overwriting the original, and leading to the result in Fig. 8.15 (right).

Stage 3: Calculation of waveform index in the compressed array, relative to the beginning of the event.

A small 2D grid of dimension $1 \times N_{\text{evts}}$ is used. This time, each block treats one entire event. Each block consists of 256 threads, one single thread for each of the 256 waveforms per crate.

In the case of crate 0, the waveform indexes with respect to the beginning of the crate are also the indexes with respect to the beginning of the event. The indexes of the second crate waveforms with respect to the beginning of the event are calculated by taking the indexes with respect to the beginning of their crate and adding to them the size of all compressed waveforms in the first crate. All crates are processed in this way in a total of 17 iterations. The result is shown in Fig. 8.17.

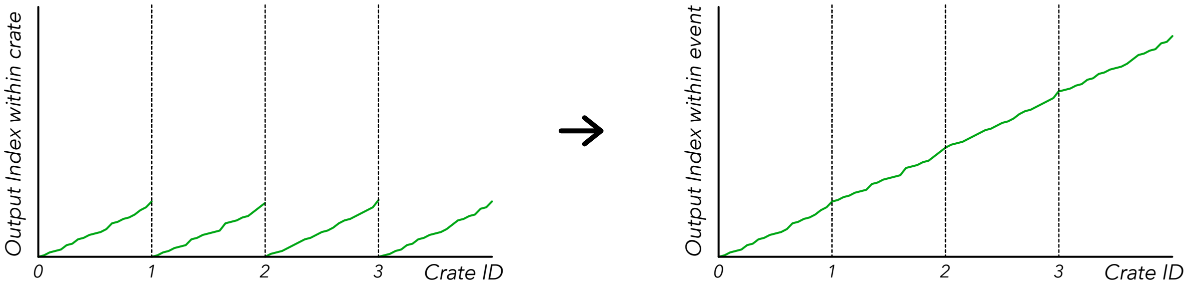


Figure 8.17: Working principle of the third stage of the compression algorithm, illustrated with just 4 crates. All 256 waveforms per crate are shifted upwards by adding to every waveform's index the size of the previous crate.

The output of this stage is the index in the array of compressed waveforms in which each waveform will need to be written once compressed, relative to the beginning of the event. The sizes of each compressed event (equal to the relative index of the last waveform in the event plus its size) are stored in a separate array.

At this point, the CPU retrieves from the GPU the compressed size of every event. The starting index of each event in the compressed array is simply calculated on CPU as $\text{index}_i = \text{index}_{i-1} + \text{size}_{i-1}$, for each event i in the event block. The CPU keeps the size of the whole compressed event block, as it will need it later to send the data to the Disk Nodes. The array of starting indexes per compressed event is sent back to the GPU.

Stage 4: The actual waveform compression

The actual waveform compression is performed by 16 threads per waveform. A 2D grid of dimension $N_{\text{adcs}} \times N_{\text{evts}} = 288 \times N_{\text{evts}}$ is used, and each 256-thread block targets a single ADC crate.

N_{bps}, and the output index relative to the beginning of the event are already known at this stage, and are retrieved from global memory. The actual uncompressed waveforms are also copied from global to shared memory, as they will be accessed repeatedly during the compression.

Each compressed waveform starts with a 3-byte header, built as described in section 8.4.1. If $0 < N_{bps} \leq 8$, the compression is performed according to the principle illustrated in Fig. 8.18.

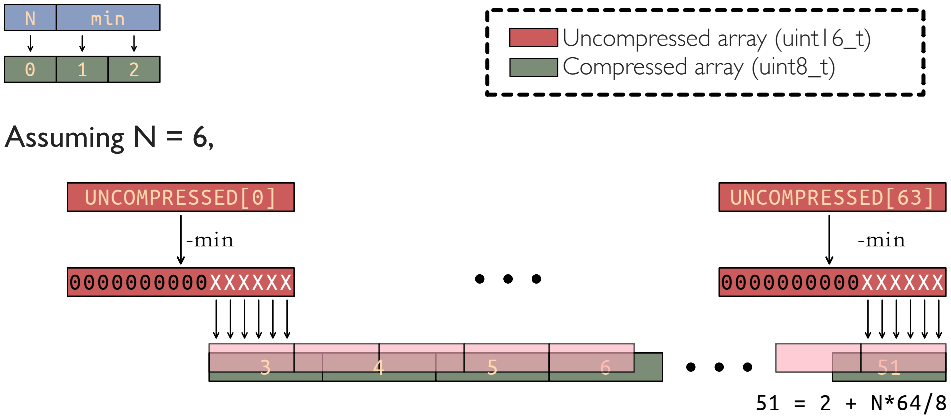


Figure 8.18: Compression principle on GPU, illustrated for the case where $N_{\text{bps}} = 6$. 16 threads compress in parallel a single waveform, each of them writing 4 samples into the compressed array.

The size in bits of each compressed sample can be any number from 0 to 8, but the compressed array is indexed in units of bytes (8-bit steps). $N_{\text{bps}} = 6$ is set as an example in Fig. 8.18. All compressed samples (pink rectangles at the bottom of Fig. 8.18) are written immediately after the previous one at the bit level. Compressed samples are cut whenever an 8-bit boundary is reached, and their LSBs are written to the first bits of the next byte.

Each compressed waveform is finally copied to global memory, into a compressed array, to the memory address calculated in the previous stages. This operation completes the waveform compression.

8.5 Data reduction from compression and pedestal suppression

Low energy waveforms typically have small amplitudes, and high compression factors even without applying pedestal suppression. However, the total number of bits needed to contain a waveform grows proportionally to $64 \cdot \log_2(\text{amplitude})$, making a *perfectly flat* waveform significantly less heavy than an *almost flat* one. Figure 8.19 shows the size of compressed waveforms as a percentage of the originals, as a function of their online-calculated energy. Most waveforms within the suppression range (-2 MeV to 1 MeV) are already compressed to $20 \sim 30\%$ of their original size without pedestal suppression, as shown in Fig. 8.19 (left). Applying pedestal suppression further reduces their compressed size to a negligible 0.9% of the original, filling the two bins at the bottom of Fig. 8.19 (right). Low-energy waveforms that are not suppressed do not satisfy the peak height requirement.

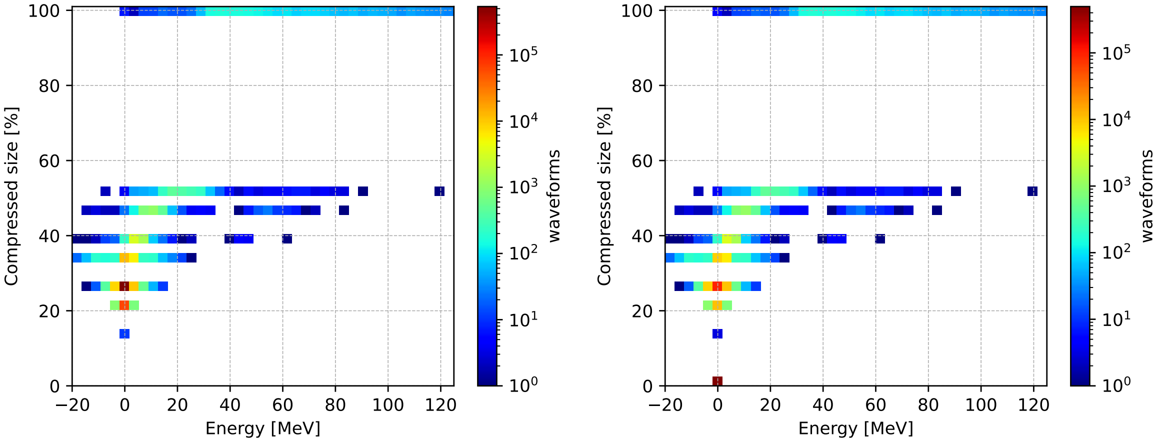


Figure 8.19: Size of the compressed waveform, as a percentage of the original, as a function of the online-calculated Energy. Left: Without applying pedestal suppression. Right: Applying pedestal suppression.

Figure 8.20 shows the compressed size per waveform with and without pedestal suppression enabled, i.e. the projection of the two histograms in Fig. 8.19 onto the y-axis.

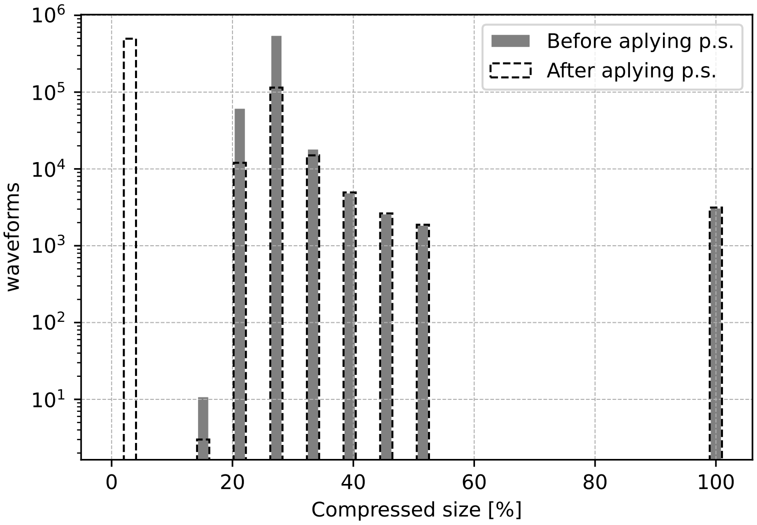


Figure 8.20: Compressed size per waveform, before and after applying pedestal suppression, as a percentage of the original waveform size.

To calculate the total data reduction from pedestal suppression and compression, the total size of every compressed event (including compressed waveforms and uncompressed ADC headers) is compared to the event's uncompressed size. Figure 8.21 shows the data reduction distributions of physics and cosmic runs with different pedestal-suppression configurations.

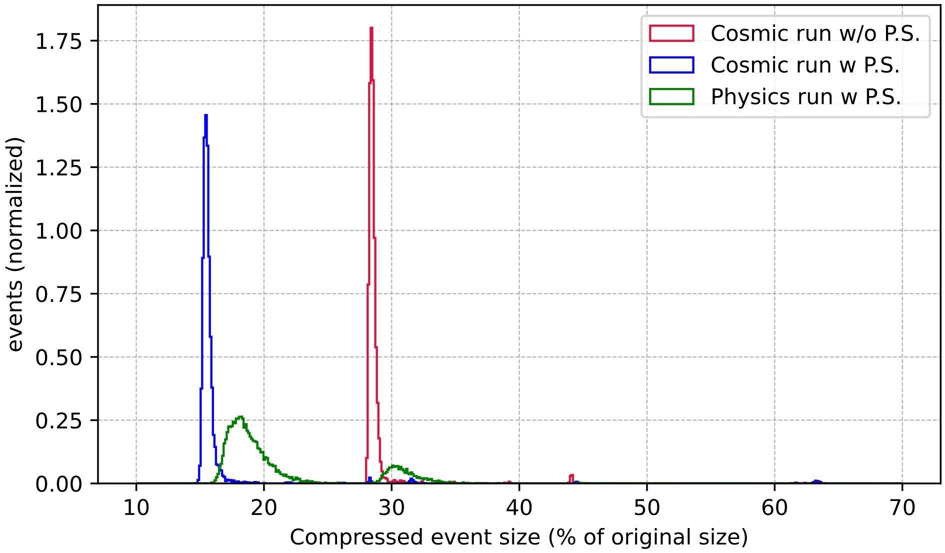


Figure 8.21: Data reduction per event for a cosmic run with pedestal suppression enabled (blue), a cosmic run with pedestal suppression disabled (red), and a physics run including both PS-enabled and PS-disabled triggers (green). Events with low compression factors (large compressed size) mainly come from the non-physics LED and laser triggers, which produce large signals in a large amount of detector channels whose waveforms are not compressed. The histograms are normalized to their integral.

The deposited energy per channel in cosmic events is roughly limited by the maximum distance a muon can travel through each crystal. The number of CsI hits is also limited by the trajectory of the cosmic muons. This makes the compressed size of cosmic events smaller, and its distribution more uniform than the case of physics runs as shown in Fig. 8.21.

In physics runs, the CsI hit rate and deposited energy also differ slightly from trigger to trigger. Events from the $K_L \to 3\pi^0$ trigger, which expects six clusters in the calorimeter, will have a smaller data reduction than events from the $K_L \to \pi^0\nu\overline{\nu}$ trigger, which expects only two clusters. Charged pion clusters, such as the ones expected in the $K_L \to \pi^+\pi^-\pi^0$ trigger, are generally more spread out, and tend to have smaller compression factors than photon or electron clusters coming from the $K_L \to 3\pi^0$, $K_L \to \pi^0\nu\overline{\nu}$, or $K_L \to \pi^0 e^+e^-$ triggers. The two peaks in physics runs come from non-pedestal-suppressed triggers such as the minimum bias or the $K_L \to \pi^0\nu\overline{\nu}$ trigger (right peak) and events from other PS-enabled triggers (left peak). Figure 8.22 shows the data reduction of physics events coming from both pedestal suppressed and non-suppressed triggers.

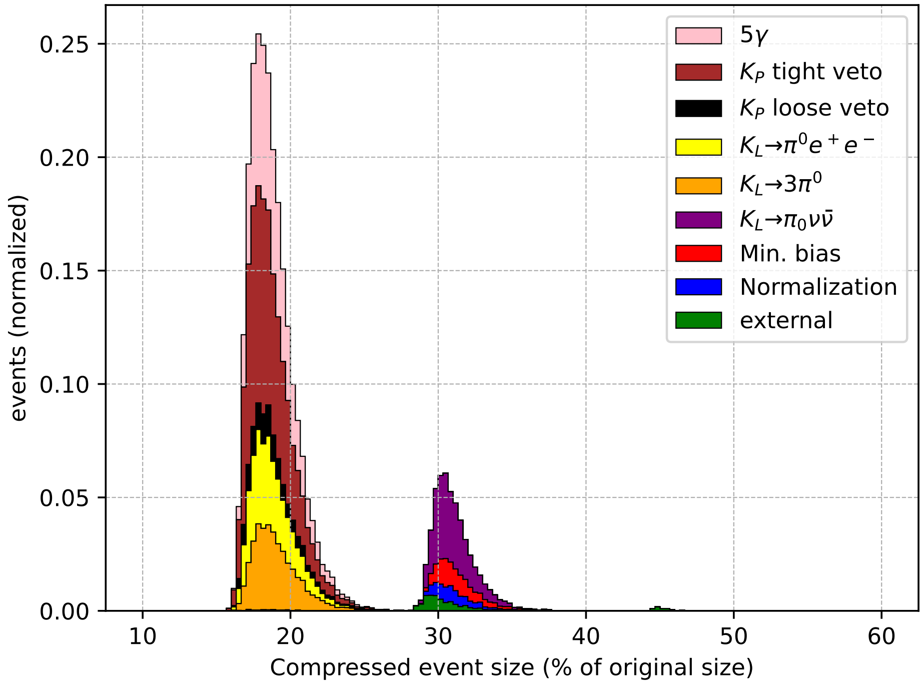


Figure 8.22: Data reduction per event for events belonging to different trigger types. Trigger types in the right peak have pedestal suppression disabled. These triggers are the Normalization and $K_L \to \pi^0 \nu \overline{\nu}$ triggers used for the main physics analysis, the minimum-bias trigger, and the external triggers.

Once pedestal suppression was implemented in the HLT, all cosmic runs were taken with pedestal suppression enabled.

From the histograms in Fig. 8.21, the average compressed size per event in cosmic runs is 16% of the original. The average compressed size per event in physics runs, averaging all events from all triggers, is 21% of the original. The average size of events from triggers with pedestal suppression enabled is 60% of the size of events where the pedestal suppression is disabled.

Chapter 9

Data packing and storage on disk

At this stage, the event reconstruction, selection, and compression are complete, and the events are ready to be written to disk at the Disk Nodes. This operation, however, is not straightforward. Compressed waveforms are scattered across the GPU memory of the four Computing Nodes that process every spill. Furthermore, ADC headers are not attached to the waveform data, and the event headers have not been built. Complete events have to be packed at the GPUs and then sent to the Disk Nodes. At the Disk Nodes, events from all Computing Nodes need to be classified according to their spill number and finally written to disk. The technical design of this process is discussed in this chapter. The optimizations implemented to maximize the performance of the data transfer to the Disk Nodes are described in detail.

9.1 HLT event headers

The GPU buffer where all compressed waveforms were written asynchronously is the same buffer that will be eventually sent out to the Disk Nodes. When calculating the start index of every compressed event during the compression stage, enough space is reserved for a 13-byte HLT header followed by all ADC headers (288 ADCs, 12 bytes per ADC). The header of rejected events contains only the HLT header.

The structure of the ADC headers was described in Fig. 4.11. The contents of the HLT header are summarized in Table 9.1.

Table 9.1: Contents of the 13-Byte HLT header. Information encoded in the HLT tag was described in section 7.2 and table 7.3.

Number of bytes	Description
4	Compressed event size (including header)
1	Trigger type
2	HLT tag
2	Event number
2	Spill number
2	Run number

The compressed event size is used at both the Disk Nodes and the offline reconstruction framework to quickly jump between events. Only rejected events have a fixed event size, equal to the HLT header size.

Once all headers have been inserted into the waveform buffer, the buffer can be sent to the Disk Nodes.

9.2 Data transfer to the Disk Nodes

All HLT nodes are interconnected through 10 Gbps ethernet cables for data transfers, and through a 1 Gbps network for control. Both the communication and the data transfers are handled in the software layer by a custom build of MPI (the Message Passing Interface) [34].

9.2.1 Technical software design of the data transfer: RDMA

Complete data buffers containing processed events are located on the GPU. Data can be copied from GPU to CPU (specifically, from GPU memory to the CPU's RAM) using the CUDA API. Then, CPU memory can be copied to a remote CPU with MPI. This approach is straightforward and does not require any special configuration. However, for the following three reasons, it is not ideal.

- First, both the GPU – CPU and the CPU – Remote CPU memory copies require the "CPU" in the Computing Nodes to actively take part. Thus the available CPU resources left for other applications would be reduced.
- Second, a large RAM buffer would need to be pre-allocated on the Computing Nodes for the compressed buffer (whose size is a priori unknown), constraining the memory left for other stages.

- Third, bandwidth from the PCIe to the CPU memory would now need to be shared by the packet capture (limited to 40 Gbps by the maximum packet rate), and the compressed data copy from the GPU (limited to 100 Gbps by the GPU). Both transfers happening simultaneously could compromise the data capture.

Motivated by the issues described above, MPI is built with UCX [35] and CUDA support, enabling it to perform RDMA (Remote Direct Memory Access). This allows for direct data transfers between the GPU of the Computing Nodes and the CPU (RAM) of the Disk Nodes. Figure 9.1 shows the data flow from the 40G input at the Computing Nodes to the disks at the Disk Nodes, with and without performing RDMA.

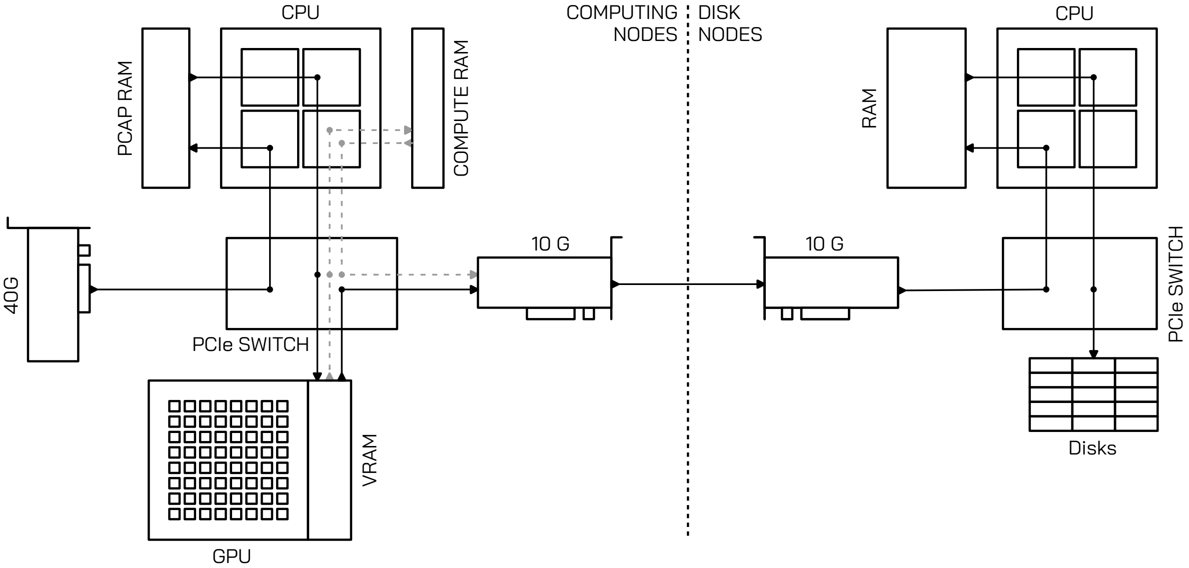


Figure 9.1: Data flow from the 40G input at the Computing Nodes to the disks at the Disk Nodes. Without performing RDMA (dotted line), the data needs to be copied back to the RAM, and then transferred to the Network Interface Card (NIC), increasing the PCIe switch load. At the Computing Nodes, the "PCAP RAM" is the portion of the RAM explicitly reserved for the packet capture. The rest of the RAM, labeled "COMPUTE RAM" is used for other tasks.

As shown in Fig. 9.1, the RDMA transfer goes directly from the GPU to the 10G NIC through the PICe Switch, completely bypassing the Operating System of the Computing Nodes. Remarkably, the PCIe – CPU traffic is reduced to just the transfer from the 40G NIC to the RAM, plus the raw data transfer from the RAM to the GPU. These transfers do not interfere with each other, as they go in opposite directions.

It should be noted that performing RDMA does not have any real impact on the data transfer between the Computing Nodes and the Disk Nodes. Both the RDMA and the non-RDMA transfers are limited by the 10G link to the Disk Nodes, which is 10 times slower than the 100G between the PCIe and the GPU. This is shown in Fig. 9.2. In the figure, the latency is defined as the time spent to complete a data transfer. The latency of the transfer from the Computing Node's GPU to the Disk Node decreases by around 10% when performing RDMA (orange line in the figure) with respect to the non-RDMA case (green line). The latency increase when not performing RDMA (green line in the figure) translates just into extra delay, without reducing the rate at which data can be transferred, 10 Gbps.

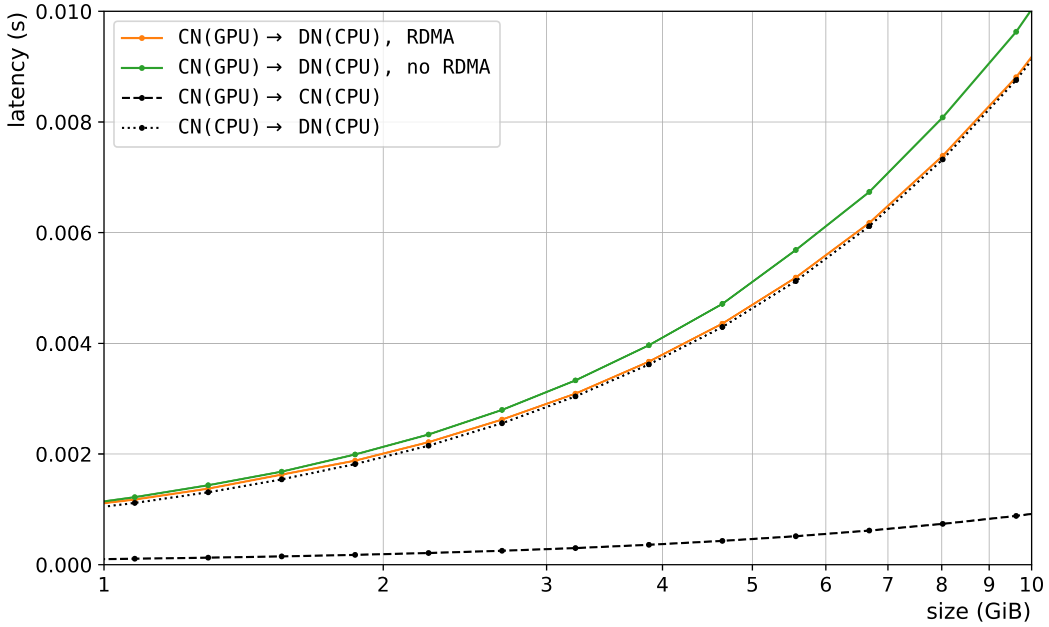


Figure 9.2: Latency of the data transfer from the GPU of the Computing Nodes (CN) to the RAM memory of the Disk Nodes (DN). Orange: direct transfer through RDMA. Green: Transfer without RDMA. Dashed black line: GPU to CPU copy (100 Gbps). Dashed dot line: CPU (CN) to CPU (DN) copy (10 Gbps). The non-RDMA transfer (green) is roughly equal to the sum of the two dashed lines.

9.2.2 Network configuration between Computing Nodes and Disk Nodes: Interface bonding and LACP

The software side of the data transfer between HLT nodes has been presented in the previous section. In this section, we will introduce the hardware layout on top of which the data transfer takes place. Fig. 9.3 shows the connections between the Computing Nodes and the Disk Node through a 10G switch.

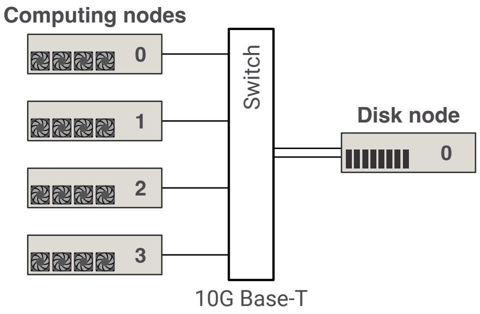


Figure 9.3: Cable connections between Computing Node and Disk Node, through a 10G switch.

It is clear from Fig. 9.3 that the maximum bandwidth between the Computing Nodes and the switch is larger than between the switch and the Disk Node. However, this link does not represent a bottleneck during physics runs. Since no data reduction happens at the Disk Node, and the bandwidth to KEK is limited to an average of 4 Gbps, the *average* bandwidth between the Computing Nodes and the switch has to be kept below 4 Gbps.

The exception to this rule are short non-physics runs, such as aluminum target runs, where pedestal suppression and event selection are deactivated, and data rates larger than 4 Gbps can be sustained throughout the whole length of the runs. In 2024, the maximum rate in an aluminum target run was 25 kTriggers per spill, which is equivalent to 9.2 Gbps at the input of the Disk Node. The limit was imposed by the limited disk I/O at the Disk Nodes, not by the network bandwidth. However, it is very close to the theoretical maximum of 10 Gbps that a single 10G link can provide.

To increase the bandwidth between the Computing Nodes and the Disk Node, the Disk Node is connected to the switch through the two ports of its 10G interface. On the server side, these two 10G ports are bonded into a new single logical interface, which gets a unique MAC address, and effectively acts as a single 20G interface. On the switch side, LACP (Link Aggregation Control Protocol) is configured, to make the switch aware of the bonding at the computer side. Packets incoming at the switch directed to the Disk Node are distributed by the switch between the two physical interfaces, balancing the load and effectively doubling the bandwidth between the computing and the Disk Nodes. Furthermore, in the event of a cable failure, LACP allows the switch to automatically redirect all traffic through the remaining cable, minimizing the impact of the failure.

Fig. 9.4 shows the maximum bandwidth through the switch before and after configuring the bonding between the switch and the Disk Node, as a function of the size of the buffer being transferred. This is tested using two Computing Nodes, connected to the switch through a 10G link each. Before configuring the bond, transfers from one Computing Node to the Disk Node are greatly affected by simultaneous transfers from the other Computing Node to the Disk Node, as the total bandwidth is limited to 10 Gbps. After configuring the bond, the DN can now receive at up to 20 Gbps, and both transfers can be performed at their individual peak performances of 10 Gbps[1] without affecting each other.

[1] Due to the way MPI was configured, the actual bandwidth peaks at around 9.4 Gbps.

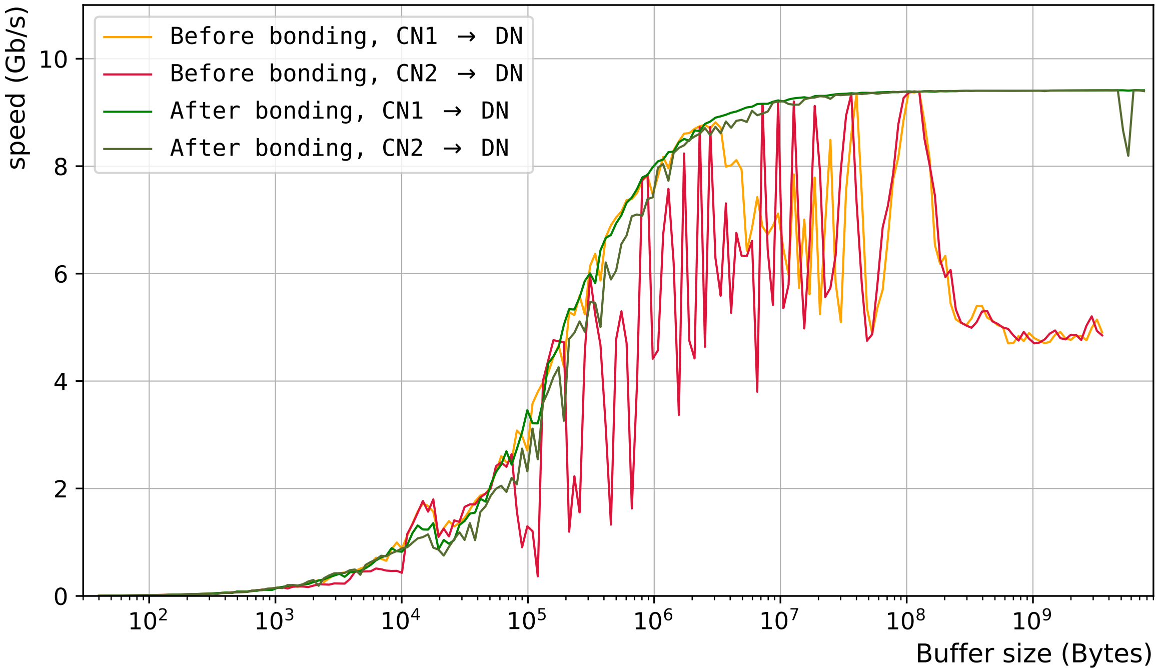


Figure 9.4: Performance improvement after configuring LACP on the 10G switch and bonding two 10G interfaces on the Disk Node (DN). The two Computing Nodes (CN1 and CN2) can make full usage of their 10G output after the bonding is configured.

9.2.3 Receiving and writing to disk data at the Disk Nodes

The Disk Nodes mainly act as a temporary buffer between the Computing Nodes and the computer farm at KEK, where data is permanently stored. The disk capacity of the Disk Nodes is 96 TB, divided among eight 12-TB disks. For an average input of 4 Gbps from the Computing Nodes, one Disk Node can store 2 days' worth of continuous data taking.

During the HLT initialization, the Disk Node allocates several buffers in the RAM. Each buffer stores incoming data belonging to a unique spill. A tag attached to each buffer indicates its current spill number and status.

Two master threads constantly monitor the network, waiting for incoming data from any of the Computing Nodes. Whenever a buffer is ready to be sent at the Computing Node side, a master thread first retrieves its spill number and size. If no data from the same spill has been received before, the master thread picks an empty buffer and receives the data into it. If any pre-allocated buffer already contains data from the same spill and has enough free space, the master thread receives the data into that buffer.

Buffers containing data are written to disk in the following cases:

- A buffer containing data from one spill does not have enough free space to store incoming data belonging to the same spill.

- The spill number of a buffer is more than two spills behind the current spill. At this point, no more data from that spill is expected to be received[2].
- No data has been received in the last 10 seconds. This would be the case if DAQ finishes, or if it stops after an error.

Writing data to disk takes a significant amount of time, and it is performed by worker threads that the masters invoke when needed. This way, the master threads can focus on waiting for packets and receiving them, minimally affecting the data transfer from the Computing Nodes to the Disk Node. Data is written to disk in binary format. Further lossless compression to the binary files (e.g. zip or tar) would reduce their size by an extra 12%, but it is not performed to minimize the data loss in case of a partial file corruption. Binary files getting corrupted between the disk nodes and KEK is a very rare event, and no instance of it has been observed during the 2024 runs. However, corruption could happen once the files are transferred to tape for long-term storage at KEK, due to deterioration or physical damage of the tapes.

A typical spill from a physics run in 2024 contained 18k events, which on average take around 1.8 GB on disk. The Disk Node's ram buffers are 1.2 GB each, and thus spills are written into two binary files, one of 1.2 GB and the other around 600 MB. Producing large files facilitates the data transfer to KEK, which will be briefly covered in the next section.

Other than receiving data, the Disk Nodes play an important role in monitoring the physics data collection and the DAQ performance. This will be briefly described in section 9.4.

9.3 Data storage and transfer to KEK

A program developed by the Yamagata University, independent from the HLT software, runs on the Disk Nodes and sends the binary data to KEK. Checksum tests are performed to ensure the integrity of the data before deleting files at the Disk Node.

At the starting point of the 2024 runs, the Disk Node was configured with 8 SATA disks forming a single RAID 0 array. However, shortly after, it was realized that the disk read/write speed was not enough to simultaneously cope with the HLT writing binary data and the data being read as it is sent to KEK. The limited write speed when both programs were working simultaneously caused a bottleneck in the HLT, leading to the saturation of buffers up to the 40G capture stage, and eventually to the loss of events in multiple spills during the first half of the beam-time.

To temporarily solve the issue with the available hardware, the disk array was reconfigured into two RAID 0 virtual disks. The HLT would target one disk while the J-PARC – KEK transfer program works on the other disk, as shown in Fig. 9.5. Once the transfer program finishes, it would make the HLT aware of it. Once the run finishes, the HLT and the transfer program would switch targets so no interference happens. Reducing the number of disks from

[2] If an incoming buffer with an old spill number were detected, it would be rejected.

8 to 4 halved the maximum write speed, but the stability gain allowed both the HLT and the transfer software to operate smoothly until the end of the beam-time.

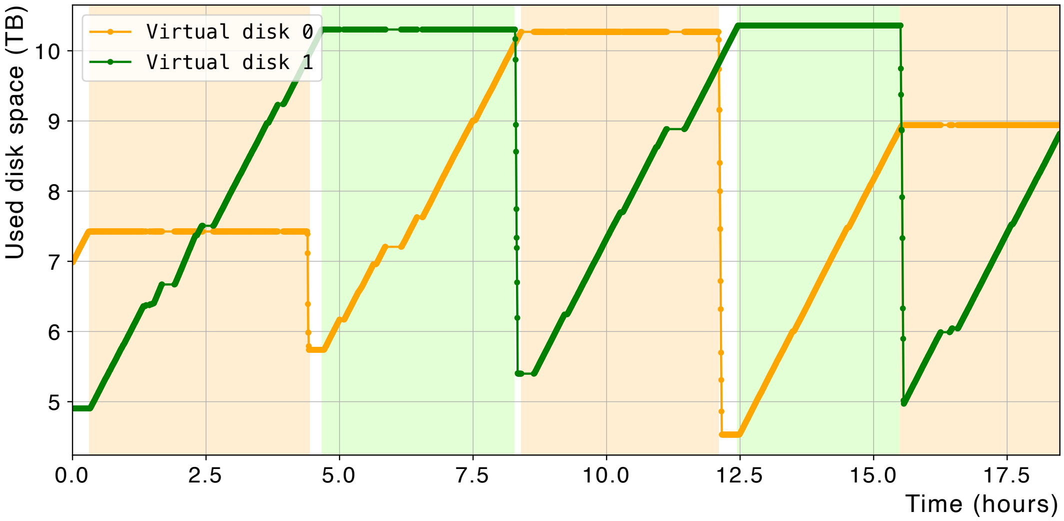


Figure 9.5: The process of switching target RAID disks to avoid conflicts between the HLT software and the J-PARC – KEK transfer software. During the first period highlighted in orange, the transfer program is sending to KEK data from virtual disk 0. In the meantime, the HLT is writing data to the virtual disk 1. After enough data has been sent, data is deleted from disk 0 and the target is switched. The HLT notices this, and it too switches targets. In the second highlighted zone (green), the transfer program reads from disk 1, and the HLT writes into disk 0, etc.

This section completes the description of the HLT software. A short description of the HLT and DAQ monitoring is given in the next section, followed by the performance and efficiency evaluation of the HLT.

9.4 HLT and DAQ Monitor

Monitoring the DAQ is crucially important to identify and promptly fix hardware failures that could lead to loss or corruption of data. Monitoring the DAQ involves checking the consistency of trigger rates, data loss, and event rejection rates among others. In this section, we will focus on the role of the HLT and the contributions of the author of this thesis.

Statistics and logs from Top CDT, the OFC-Is, the OFC-IIs, and the HLT are collected every spill. Some of these statistics, particularly event rates, are loaded into a database hosted on a dedicated server. “InfluxDB OSS”[3], a widely used time-series database that is also open-source, was chosen for this purpose, mainly because of the extent support and documentation available online. InfluxDB data is monitored through “Grafana OSS”, an open-source web visualization tool that is connected to InfluxDB. Grafana provides an interactive and real-time visualization of DAQ statistics, including real-time DAQ efficiency, event rejection factors at

[3] “InfluxDB OSS” and “Grafana OSS” are open-source versions of “InfluxDB” and “Grafana”.

the HLT, and run information. Since it is a web-based application, any collaborator with access to J-PARC's internal network can access the data from a web browser.

Instances of InfluxDB and Grafana are also installed in Osaka University's servers, backing up the data at J-PARC and allowing it to be accessed from outside J-PARC. Some of the plots displayed in the KOTO DAQ monitor system are shown in appendix H.

At the HLT, the monitor data is extracted at the Disk Node, from each event's HLT tag before writing the binary data to disk. The CPU resources involved in this operation are not significant, as the Disk Node performance is completely limited by the writing speed of binary files.

Together with the DAQ monitor, small binary files containing up to 200 events per spill are prepared at the Disk Nodes, and sent to a dedicated server for detector monitoring. Monitoring the detector mainly aims to identify possible broken detector channels or ADC channels. Broken detector channels are found by looking for channels with unexpectedly low hit rates. Broken or faulty ADC channels generally manifest as corrupted or abnormal (e.g. significantly wider than normal) waveforms. The detector monitoring software is maintained by KOTO collaborators at J-PARC.

In case of DAQ failure, the KOTO control room is alerted by a system developed by collaborators at J-PARC. If the failure is not critical, the DAQ system can be restarted to minimize the data loss during beam-time. The corresponding DAQ logs are later studied by the DAQ team to determine the cause of the failure.

Chapter 10

Evaluation of the Online Event Selection Efficiency at the HLT

The importance of keeping the HLT physics selection efficiency high has been motivated in several chapters in this thesis. Inefficient event selection can bias the collected data, and if the bias is large enough its effect would need to be taken into account in the offline analysis. The process of determining the selection criteria before the beam-time, verifying it at the beginning of the beam-time, and measuring its final efficiency after the beam-time with the collected data is covered in this chapter.

Before the beam-time, the efficiency and rejection factors of each cut are estimated with a sample of previously collected data that did not undergo the HLT selection. These data are converted to the same format the HLT expects from the OFC-II, and then fed into the HLT from another computer running the OFC-II emulator. The estimations obtained at this stage are given in section 10.2.

At the beginning of beam-time, several physics runs are collected with the HLT running in *tag mode*. In tag mode, no events are rejected despite their selection results. These events are collected and reconstructed offline, and the HLT selection results (encoded in the HLT tag) are compared to the offline reconstruction results. This allows us to accurately measure the HLT selection efficiencies, which can then be compared with the expectations from the study before the beam-time. This stage is covered in section 10.3.

During usual physics runs with event selection enabled, the HLT randomly selects 1% of the events and processes them in tag mode. These HLT-unbiased events are used offline after the run to verify the HLT efficiency and check its stability.

The results obtained before the beam-time with the OFC-II emulator, at the beginning of the beam period with the HLT running in tagging mode, and during physics runs with the unbiased 1% sample are compared to check the stability of the HLT efficiency. Results of this measurement will be presented in section 12.1.

10.1 Definition of the HLT efficiency and rejection factors

Differences in the online and offline pedestal calculation, timing calculation, clustering algorithm, or energy calibration constants will unavoidably introduce inefficiencies in the online event selection. The HLT efficiency is defined in Eq. 10.1.

$$\mathcal{E} = \frac{\text{events that pass the online selection and offline pre-selection}}{\text{events that pass an offline pre-selection}} \tag{10.1}$$

The HLT efficiency will be calculated trigger mode by trigger mode, namely $\mathcal{E}_{5\gamma}$, $\mathcal{E}_{K^+}$, and $\mathcal{E}_{K_L \to \pi^0 e^+ e^-}$, by including only events from the corresponding trigger mode in the numerator and denominator of Eq. 10.1.

Furthermore, the HLT efficiency will also be calculated for every particular selection criterion, by including only the events that pass the corresponding online cut in the numerator of Eq. 10.1. For instance, the efficiency of the MaxR cut is defined as:

$$\mathcal{E}_{\text{MaxR}} = \frac{\text{events that pass the MaxR cut online and the offline pre-selection}}{\text{events that pass the offline pre-selection}},$$

where the offline pre-selection includes the offline MaxR cut.

The HLT efficiency measures how close the HLT reconstruction gets to the offline one. The value obtained from this formula is conservative, as the denominator includes only a subset of the quality cuts that are performed offline. The offline pre-selection criteria included in the numerator of Eq. 10.1 are summarized in table 10.1.

Table 10.1: A subset of the offline pre-selection. Events passing these criteria become the denominator of the HLT selection efficiency. MinXY, MaxR, TotalE, and COE were motivated in section 7.6. Minimum cluster distance, energy, and RMS were motivated at the end of section 7.5.1.

	Criteria		
Variable	5γ	K^+	$K_L \to \pi^0 e^+ e^-$
MinXY	> 150 mm	> 150 mm	> 150 mm
MaxR	< 850 mm	< 850 mm	< 850 mm
TotalE	> 650 MeV	> 650 MeV	> 650 MeV
COE	> 60 mm	–	< 100 mm
Number of Clusters	$= 5$	$= 3$	$= 4$
Minimum cluster distance	> 150 mm	> 150 mm	> 150 mm
Minimum cluster Energy	> 50 MeV	> 50 MeV	> 50 MeV
Minimum cluster RMS	> 10	> 10	> 10

While high efficiency is the first priority, the HLT selection also aims to provide a reasonably high data rejection factor. The main aims are to keep the offline disk usage low and keep the HLT output below the 4 Gbps of the link from J-PARC to KEK.

The rejection factor, or rejected fraction (R.F.) is defined for each trigger mode as the ratio between the total number of processed events and the number of accepted events, as shown in Eq. 10.2.

$$\mathcal{R}_{\text{HLT}} = \frac{\text{HLT-rejected events}}{\text{processed events}} \tag{10.2}$$

In the same manner, the rejection factor of individual selection criteria is defined by including only the events rejected by the corresponding online cut in the numerator in Eq. 10.2.

In the following sections, efficiency and rejection factors will be given as a percentage.

Note about the offline energy calibration constants

The HLT efficiency calculation requires events to be reconstructed offline. The final offline calibration constants were not known at the time of performing the analysis presented in this chapter. The results reported in this chapter are obtained with the preliminary constants that were known at the time. The effect of discrepancies between the preliminary and the final calibration constants is taken into account when estimating the selection criteria for each trigger, as explained in section 10.2.

10.2 Offline efficiency estimation before physics runs

In this section, we explain the method used to estimate the cut thresholds for each trigger, before enabling the event selection in the HLT.

At the very beginning of the 2024 beam-time, several physics runs were taken including 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ triggers without any event selection. Part of these events were converted to the OFC-II format. The events were then fed into the HLT, where MinXY, MaxR, TotalE, and COE are reconstructed in the same way it is done during a real physics run. These four quantities are then recorded for all the events. The same batch of events is then processed through the offline analysis, and results are compared event by event. With this, $\mathcal{E}_{\mathrm{HLT}}$ and $\mathcal{R}_{\mathrm{HLT}}$ can be calculated for each variable as a function of the cut thresholds. At this stage, online clustering parameters are adjusted when needed to maximize efficiency.

The precise detector calibration constants are unknown at this stage, but they are expected to differ by up to ±4% from the preliminary ones calculated before the beam-time with cosmic ray data. These small variations have little effect on cluster-position-related cuts, such as the MinXY, MaxR, and COE.

The min. TotalE cut, however, needs special consideration. If there is an overall tendency of the preliminary constants to be underestimated, the online min. TotalE cut will be rejecting more events than it should, which will translate into an $\mathcal{E}_{totalE}$ lower than estimated. Accounting for this possibility, the online min. TotalE cut position is estimated after artificially increasing all online TotalE values by 5%. This is equivalent to assuming that the preliminary calibration constants are underestimated by 5%, which is more conservative than the maximum expected shift of 4%.

10.2.1 Estimation of the 5-cluster $K_L \to 3\pi^0$ online selection criteria

The goal of this trigger is to collect $K_L \to 3\pi^0$ events in which 5 clusters are reconstructed in the CsI, and the sixth photon hits a veto detector.

On top of the Min XY, Max E and the Total Energy cuts, the Center of Energy (COE) was also considered. If the sixth photon hits a veto detector surrounding the CsI calorimeter, The Center of Energy on the CsI is expected to be shifted in the opposite direction. The COE cut aims to reject the following events:

- Events in which all six photons hit the calorimeter, but two very close hits are identified as a single one, and the event is tagged as a 5-cluster event.
- Events in which the sixth photon escapes through the beam hole and it is not detected by any of the downstream veto detectors.

In both cases, the reconstructed COE value is expected to be low.

The results of the preliminary study with 5-cluster $K_L \to 3\pi^0$ events are shown in Fig. 10.1. In the plots, $\mathcal{E}$ and $\mathcal{R}$ are calculated for a range of HLT thresholds on each variable. A preliminary online cut position is manually decided from the plots, aiming to maximize $\mathcal{E}$ while keeping $\mathcal{R}$ significant.

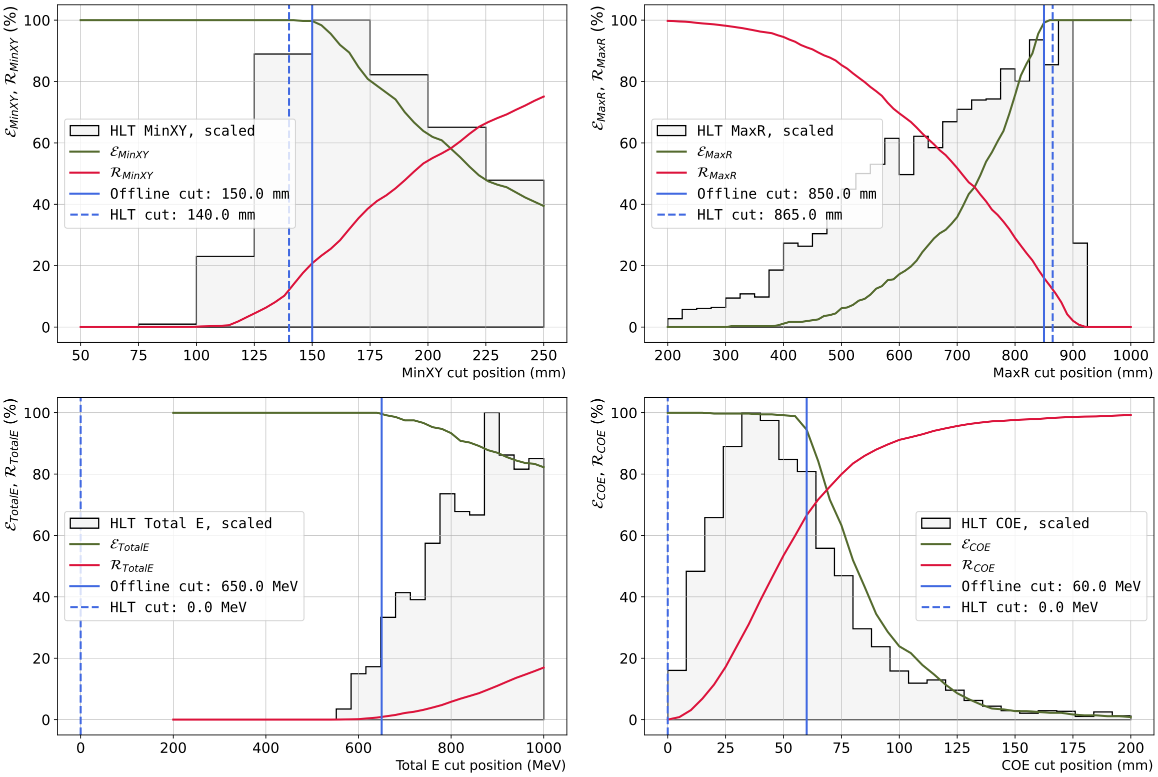


Figure 10.1: $\mathcal{E}$ and $\mathcal{R}$ of events rejected from a 5-cluster $K_L \to 3\pi^0$ sample as a function of the online cut threshold, for the MinXY (top left), MaxR (top right), TotalE (bottom left) and COE (bottom right) cuts. The vertical solid line indicates the threshold used offline. The vertical dashed line indicates the threshold determined for the HLT. The distribution of each variable, as calculated at the HLT, is shown in gray, and arbitrarily scaled to fit the plot.

The Minimum XY cut (top left in Fig. 10.1) was fixed at 140 mm, expecting to reject about 12% of the events with an efficiency $\mathcal{E}_{MinXY}$ close to 100%. Similarly, the maximum R cut (top right in the figure) is set at 865 mm, with an expected $\mathcal{R}_{MaxR}$ of also 12%. The uncertainty of these values is given in Table 10.2.

Aiming to account for possible corrections to the preliminary calibration constants, each value of the HLT TotalE was increased by 5%. In other words, the TotalE histogram shown in Fig. 10.1 is shifted to the right by 5%. As a consequence, there is no possible cut threshold that can reject a significant amount of events without losing a significant amount of efficiency. The TotalE cut in the 5γ trigger was therefore not applied at the HLT.

The COE cut, despite being the most powerful cut to reduce the 5γ sample size, was decided not to be applied online, as the total data size without it was already found to satisfy our bandwidth requirements. Originally, a minimum COE cut was considered, based on that if the photon that escapes the calorimeter in the $K_L \to 3\pi^0 \to 6\gamma$ decay hits a veto detector surrounding the CsI calorimeter, the COE of the event would be shifted in the direction opposite to the photon's trajectory. The upside of not applying the COE cut is that events in which the sixth photon escapes through the beam hole and hits one of the downstream veto detectors can be also collected. These events, whose COE is expected to be very close to zero, can be used to estimate the inefficiency of the downstream veto detectors.

The expected efficiencies and rejection factors of the MinXY and MaxR cuts for the 5γ trigger are summarized in Table 10.2.

Table 10.2: Efficiency and rejection factor of every cut applied to the 5-cluster $K_L \to 3\pi^0$ trigger.

Cut	Value	$\mathcal{E}$ (expected)	$\mathcal{R}$ (expected)
MaxR	865 mm	$\left(100.00^{+0.00}_{-0.83}\right)\%$	$(12.3 \pm 0.5)\%$
MinXY	140 mm	$\left(100.00^{+0.00}_{-0.83}\right)\%$	$(12.0 \pm 0.5)\%$
$\mathcal{E}_{5\gamma}$, $\mathcal{R}_{5\gamma}$		$\left(100.00^{+0.00}_{-0.83}\right)\%$	$(23.4 \pm 0.7)\%$

The errors in Table 10.2 correspond to the 90% confidence interval calculated following the Clopper-Pearson method [36], an exact method for calculating binomial confidence intervals widely used in particle physics. Errors presented in the following sections are calculated in the same manner.

10.2.2 Estimation of the K^+ trigger online selection criteria

The K^+ trigger mainly aims to collect $K^+ \to \pi^+\pi^0$ events. However, it is dominated by neutral kaon decay $K_L \to \pi^+\pi^-\pi^0$. Offline, these two decays are differentiated by looking at the signal in different charged veto detectors and the cluster properties in the calorimeter. At the HLT, only basic and standard quality cuts are applied. Their respective $\mathcal{E}$ and $\mathcal{R}$ are shown in Fig. 10.2.

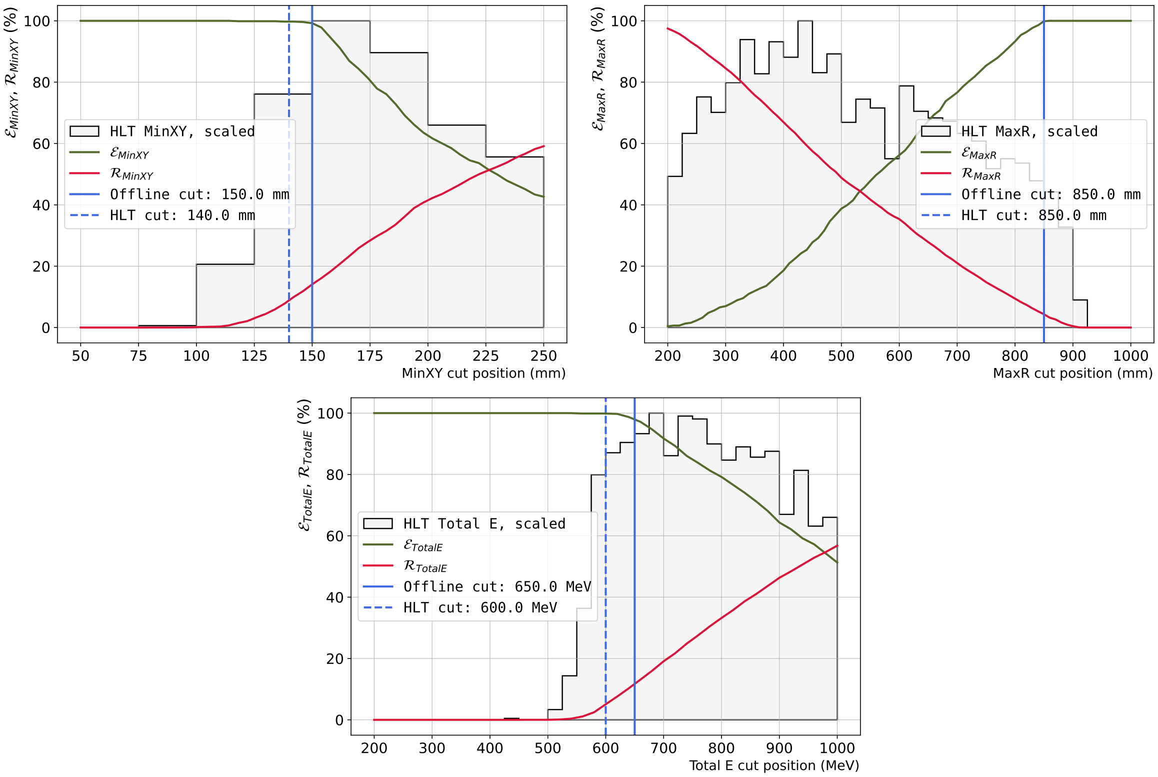


Figure 10.2: $\mathcal{E}$ and $\mathcal{R}$ of events rejected from a K^+ sample as a function of the online cut threshold, for the MinXY (top left), MaxR (top right), and TotalE (bottom) cuts. The vertical solid line indicates the threshold used offline. The vertical dashed line indicates the threshold determined for the HLT. The distribution of each variable, as calculated at the HLT, is shown in gray, and arbitrarily scaled to fit the plot.

The TotalE distribution is shifted to the right by 5% to account for possible discrepancies between the preliminary and the final calibration constants, as was done for the 5γ trigger. Still, the TotalE cut is relatively powerful when selecting $K^+ \to \pi^+\pi^0$ candidates. Most of the events rejected by this cut are $K_L \to \pi^+\pi^-\pi^0$ decays in which one pion did not hit the calorimeter. The number of reconstructed clusters is likely to become three in this case, as we would expect from the $K^+ \to \pi^+\pi^0$ decay. The total deposited energy will, however, become too low, as one of the final-state particles got missed. Expected $\mathcal{E}$ and $\mathcal{R}$ of each cut applied to the K^+ trigger are summarized in Table 10.3.

Table 10.3: Efficiency and rejection factor of every cut applied to the set of events tagged as K^+.

Cut	Value	$\mathcal{E}$ (expected)	$\mathcal{R}$ (expected)
MaxR	850 mm	$(99.87^{+0.12}_{-0.48})\%$	$(4.3 \pm 0.3)\%$
MinXY	140 mm	$(99.74^{+0.21}_{-0.55})\%$	$(8.8 \pm 0.4)\%$
Min. Total E	600 MeV	$(99.87^{+0.12}_{-0.48})\%$	$(5.1 \pm 0.3)\%$
$\mathcal{E}_{K+}$, $\mathcal{R}_{K+}$		$(99.48^{+0.34}_{-0.66})\%$	$(17.3 \pm 0.5)\%$

10.2.3 Estimation of the $K_L \to \pi^0 e^+ e^-$ trigger online selection criteria

The main background process in the $K_L \to \pi^0 e^+ e^-$ trigger is the $K_L \to \pi^+\pi^-\pi^0$ decay, where the two charged pions are misidentified as electrons. The main difference between the two is that, unlike electrons, charged pions are likely to fly through the calorimeter without depositing all their energy. This will shift the COE towards the neutral pion in $K_L \to \pi^+\pi^-\pi^0$ events, but not in the $K_L \to \pi^0 e^+ e^-$ case. The COE distribution of $K_L \to \pi^0 e^+ e^-$ and $K_L \to \pi^+\pi^-\pi^0$ events, obtained from simulation data, is shown in Fig. 10.3.

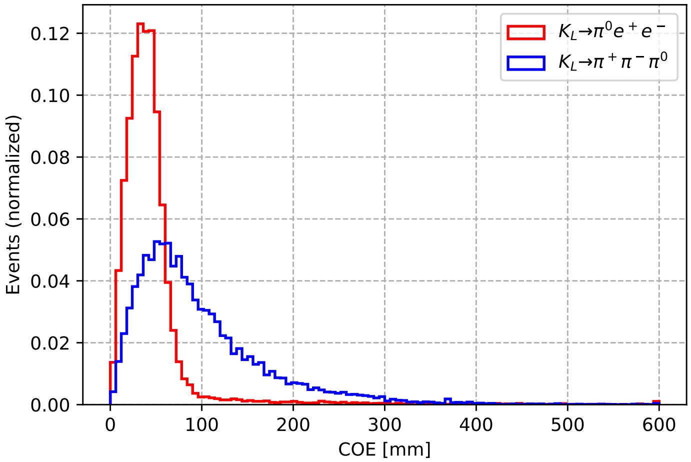


Figure 10.3: Center of Energy distribution of $K_L \to \pi^0 e^+ e^-$ and $K_L \to \pi^+\pi^-\pi^0$ simulated events. The COE of $K_L \to \pi^0 e^+ e^-$ events is expected to be close to zero, as all the final-state energy is collected in the CsI calorimeter. In the $K_L \to \pi^+\pi^-\pi^0$ case, the COE can be larger than zero if a charged pion traverses the calorimeter without depositing all its energy.

A COE cut to discriminate the two was therefore considered. However, the COE cut was found to be unnecessary in terms of bandwidth requirements. Furthermore, not applying it allows this same trigger to be used to study the $K_L \to \pi^0 \mu^+ \mu^-$ decay, similarly interesting from the physics perspective.

Figure 10.4 shows $\mathcal{E}$ and $\mathcal{R}$ of the MinXY, MaxR, TotalE, and COE cuts for the $K_L \to \pi^0 e^+ e^-$ sample.

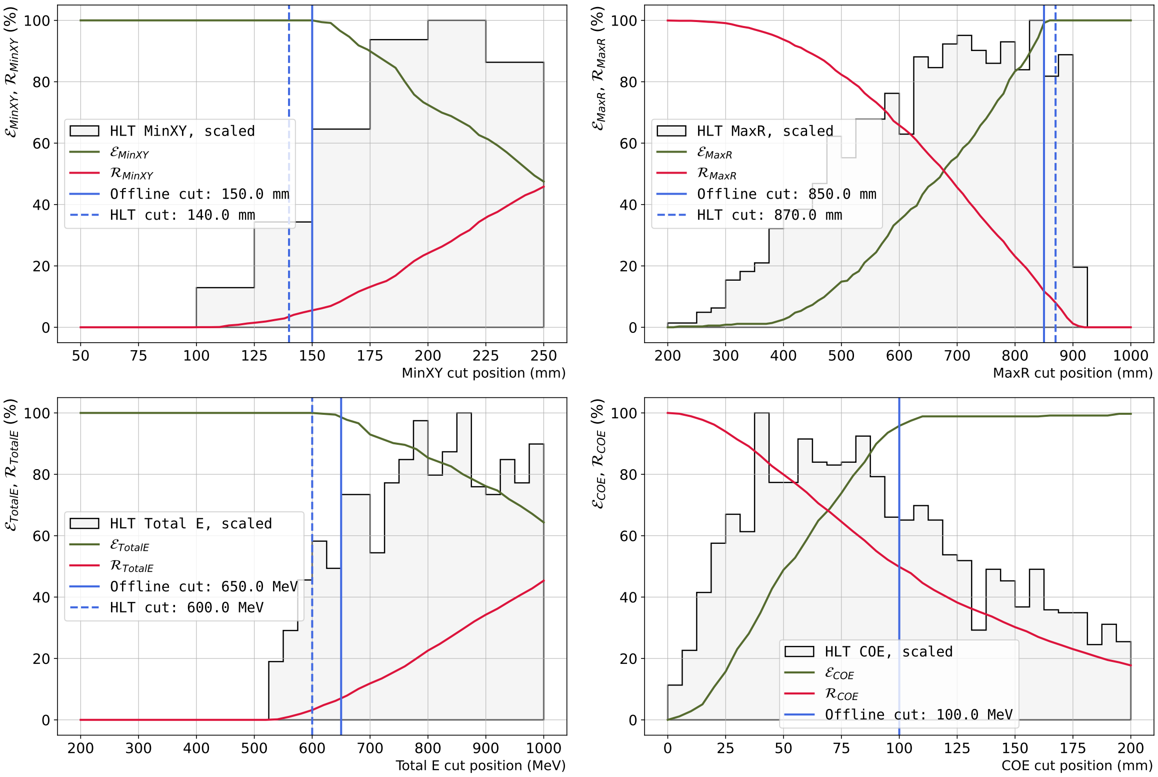


Figure 10.4: $\mathcal{E}$ and $\mathcal{R}$ of events rejected from a $K_L \to \pi^0 e^+ e^-$ sample as a function of the online cut threshold, for the MinXY (top left), MaxR (top right), TotalE (bottom left) and COE (bottom right) cuts. The vertical solid line indicates the threshold used offline. The vertical dashed line indicates the threshold determined for the HLT. The distribution of each variable, as calculated at the HLT, is shown in gray, and arbitrarily scaled to fit the plot.

The expected efficiencies and rejection factors of the MinXY, MaxR, and TotalE cuts for the $K_L \to \pi^0 e^+ e^-$ trigger are summarized in Table 10.4.

Table 10.4: Efficiency and rejection factor of every cut applied to the set of events tagged as $K_L \to \pi^0 e^+ e^-$.

Cut	Value	$\mathcal{E}$ (expected)	$\mathcal{R}$ (expected)
MaxR	850 mm	$(100.00^{+0.00}_{-0.84})\%$	$(8.0 \pm 0.6)\%$
MinXY	140 mm	$(100.00^{+0.00}_{-0.84})\%$	$(3.5 \pm 0.4)\%$
Min. Total E	600 MeV	$(100.00^{+0.00}_{-0.84})\%$	$(3.2 \pm 0.4)\%$
$\mathcal{E}_{\pi^0 e^+ e^-}$, $\mathcal{R}_{\pi^0 e^+ e^-}$		$(100.00^{+0.00}_{-0.84})\%$	$(14.2 \pm 0.7)\%$

10.3 Efficiency and rejection factors measurement during special physics runs

The method presented in the previous section has the benefit of allowing us to use the same data sample to simultaneously estimate $\mathcal{E}$ and $\mathcal{R}$ for different cut thresholds in each variable. On the other hand, it requires converting collected data back to the OFC-II format and feeding it back into the version of the HLT that records each reconstructed value. This is a time-consuming process and requires the HLT to work in conditions that differ from its normal working conditions during beam-time.

The candidate cut thresholds obtained in the previous section are tested with beam data in the special *HLT-tag* runs. In these runs, the HLT performs normal event reconstruction and selection with these preliminary thresholds, but keeps all events despite the selection results. The processed data is eventually reconstructed offline, and the online and offline results are directly compared without the need to re-feed any events into the HLT. The agreement between the $\mathcal{E}$ and $\mathcal{R}$ values estimated in the previous section and the ones measured in the *HLT-tag* runs is checked.

The analysis method at this stage consists in comparing the offline-reconstructed distribution of each variable, before and after applying the correspondent HLT cut. If the HLT efficiency was 100%, all offline-reconstructed events at the “accepted” side of the offline cut threshold would also be accepted by the HLT. This process will become clear with the examples shown in this section.

10.3.1 5-cluster $K_L \to 3\pi^0$ trigger

Distributions of the MinXY and MaxR variables reconstructed offline from 5γ trigger data, before and after applying the HLT selection, are shown in Fig. 10.5.

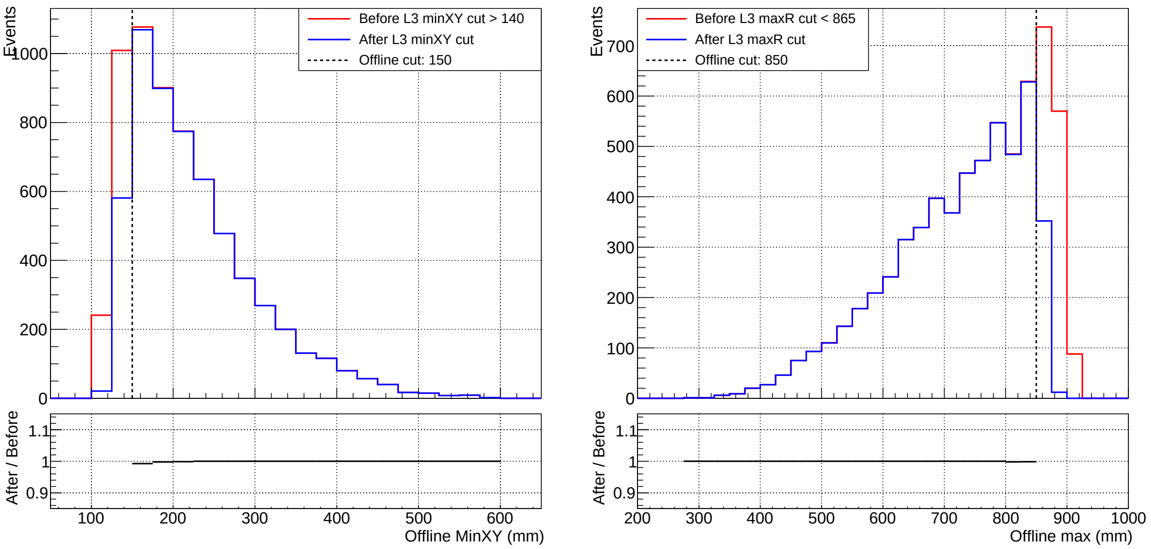


Figure 10.5: Distributions of the MinXY (left) and MaxR (right) variables, reconstructed offline from 5γ trigger data. The vertical dotted line indicates the offline cut threshold.

Using data directly from the plots above, the calculated $\mathcal{E}$ and $\mathcal{R}$ of the MinXY and MaxR cuts are shown in Table 10.5.

Table 10.5: $\mathcal{E}$ and $\mathcal{R}$ of the MinXY and MaxR cuts applied to the 5-cluster $K_L \to 3\pi^0$ trigger.

Cut	Value	$\mathcal{E}$ (measured)	$\mathcal{R}$ (measured)
MaxR	865 mm	$\left(99.96^{+0.03}_{-0.08}\right)\%$	$(11.9 \pm 0.1)\%$
MinXY	140 mm	$\left(99.79^{+0.09}_{-0.14}\right)\%$	$(11.1 \pm 0.1)\%$
$\mathcal{E}_{5\gamma}$, $\mathcal{R}_{5\gamma}$		$\left(99.75^{+0.10}_{-0.15}\right)\%$	$(23.0 \pm 0.2)\%$

The values shown in Table 10.5 are consistent with the ones expected from Table 10.2. The combined efficiency is also larger than the 99% target. Still, inefficient events must be understood before finally enabling the event selection at the HLT.

10.3.2 K^+ trigger

Fig. 10.6 shows the distributions of the MinXY, MaxR, and TotalE variables reconstructed offline with events from the K^+ trigger, before and after applying the HLT selection.

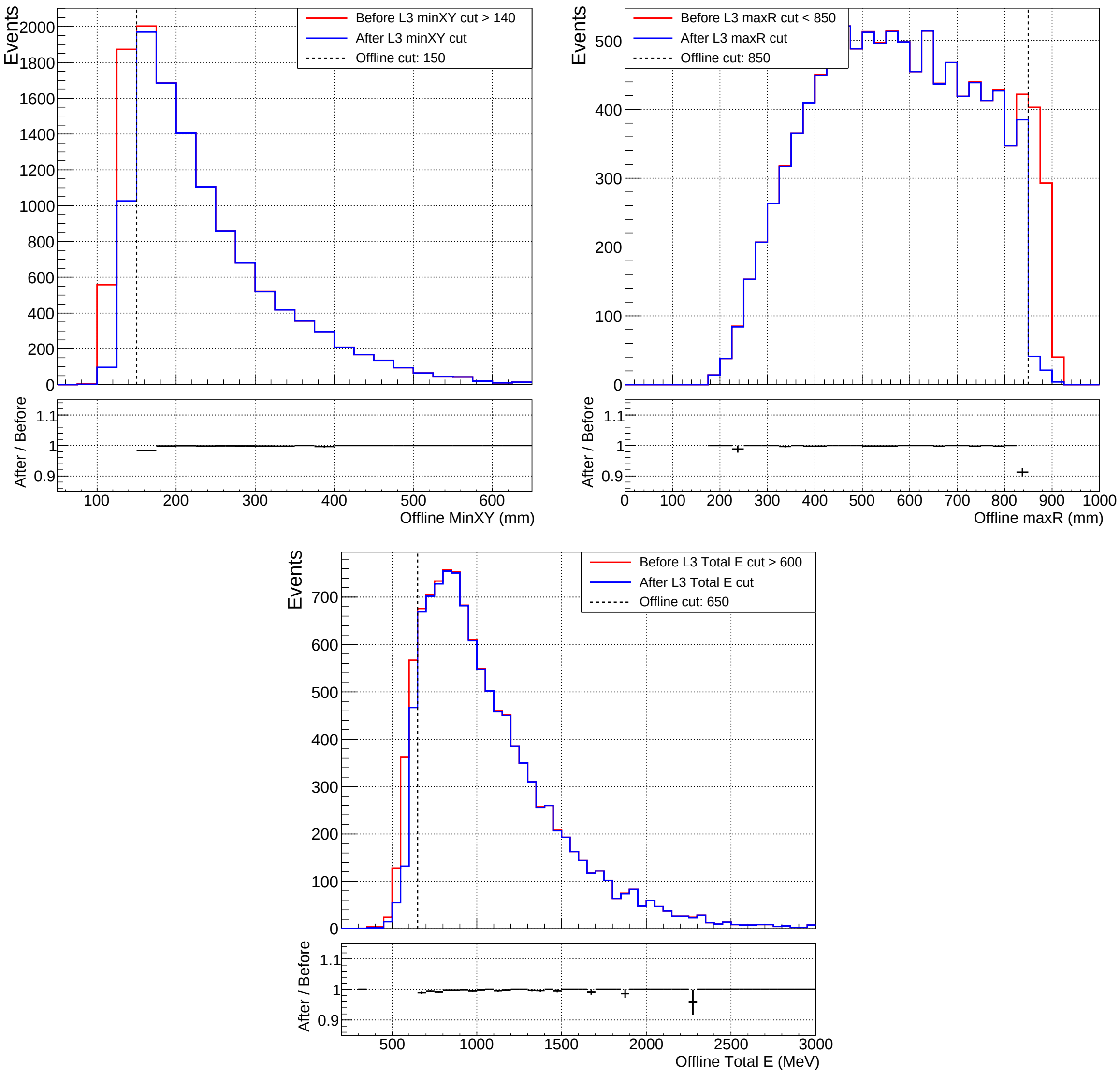


Figure 10.6: Distributions of the MinXY (top left), MaxR (top right), and TotalE (bottom) variables, reconstructed offline from K^+ trigger data. The vertical dotted line indicates the offline cut threshold.

The corresponding $\mathcal{E}$ and $\mathcal{R}$ of the MinXY, MaxR, and TotalE cuts are shown in Table 10.6.

Table 10.6: Efficiency and rejection factor of every cut applied to the K^+ trigger.

Cut	Value	$\mathcal{E}$ (measured)	$\mathcal{R}$ (measured)
MaxR	850 mm	$\left(99.54^{+0.11}_{-0.13}\right)\%$	$(4.9 \pm 0.1)\%$
MinXY	140 mm	$\left(99.57^{+0.10}_{-0.12}\right)\%$	$(8.4 \pm 0.1)\%$
Min. Total E	600 MeV	$\left(99.66^{+0.09}_{-0.11}\right)\%$	$(7.7 \pm 0.1)\%$
$\mathcal{E}_{K^+}$, $\mathcal{R}_{K^+}$		$\left(98.79^{+0.17}_{-0.19}\right)\%$	$(20.0 \pm 0.2)\%$

The efficiencies above are also consistent with the ones that were expected, shown in Table 10.3. However, it can be seen in Fig. 10.6 that some of the quantities reconstructed online and offline differ significantly in some events. These discrepancies need to be understood before finally enabling this event selection at the HLT.

10.3.3 $K_L \to \pi^0 e^+ e^-$ trigger

Results of the MinXY, MaxR, and TotalE selection efficiencies for the $K_L \to \pi^0 e^+ e^-$ trigger are shown in Fig. 10.7.

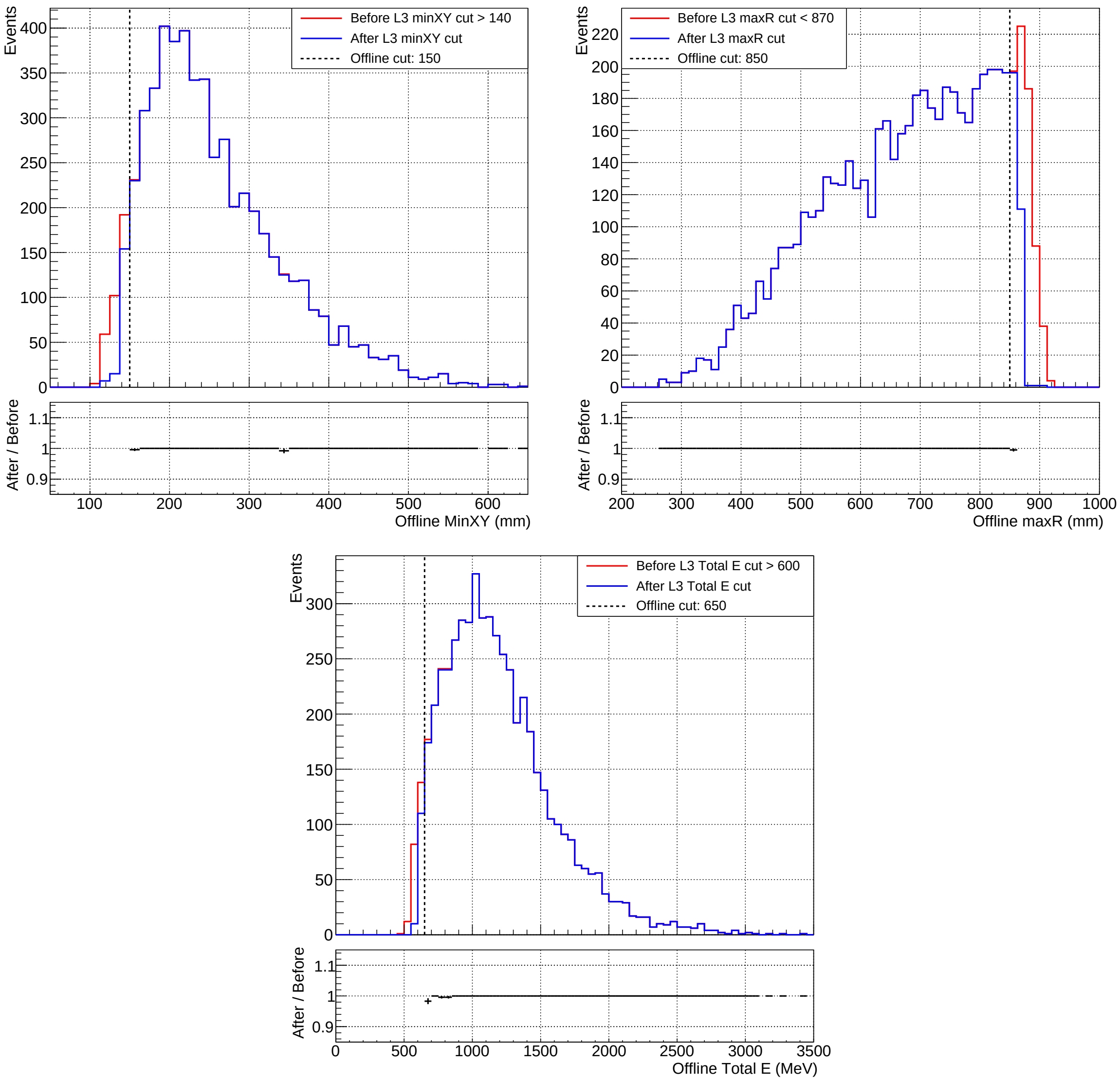


Figure 10.7: Distributions of the MinXY (top left), MaxR (top right), and TotalE (bottom) variables, reconstructed offline from $K_L \to \pi^0 e^+ e^-$ trigger data. The vertical dotted line indicates the offline cut threshold.

The corresponding efficiencies and rejection factors are shown in Table 10.7.

Table 10.7: Efficiency and rejection factor of every cut applied to the $K_L \to \pi^0 e^+ e^-$ trigger.

Cut	Value	$\mathcal{E}$ (measured)	$\mathcal{R}$ (measured)
MaxR	870 mm	$\left(100.00^{+0.00}_{-0.06}\right)\%$	$(6.4 \pm 0.1)\%$
MinXY	140 mm	$\left(99.96^{+0.03}_{-0.08}\right)\%$	$(3.0 \pm 0.1)\%$
Min. Total E	600 MeV	$\left(99.90^{+0.06}_{-0.11}\right)\%$	$(4.7 \pm 0.1)\%$
$\mathcal{E}_{\pi^0 e^+ e^-}$, $\mathcal{R}_{\pi^0 e^+ e^-}$		$\left(99.75^{+0.09}_{-0.12}\right)\%$	$(16.0 \pm 0.2)\%$

Once again, these efficiencies are consistent with the predictions previously made before beam-time. The combined efficiency is also good enough.

10.3.4 Events contributing to the inefficiency of the HLT selection

Small discrepancies in the online and offline energy calculation of individual CsI channels can cause small shifts in the reconstructed cluster positions and energies online and offline. Sometimes, if the actual energies or cluster positions of an event are already close to the cut thresholds, these discrepancies can be enough for an event to be accepted offline but rejected online. Low selection efficiencies originated from these events are unavoidable, and tolerable as long as the selection is kept around the 99% target. If the efficiency was measured significantly below 99% at this stage, the thresholds would simply need to be loosened to increase $\mathcal{E}$, at the cost of a lower $\mathcal{R}$.

Discrepancies between the offline and online reconstructed variables can also arise from reasons other than slightly different energy calculations. To identify these reasons, all inefficient events from the 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ triggers presented in the previous sections have been individually studied. The ones whose inefficiency was not due to discrepancies in energy calculations were found to be inefficient due to one of the following reasons:

- A single cluster is erroneously split into two at the HLT, and one of the split clusters falls into the MinXY or MaxR rejection area.
- Two clusters, one of them triggering the event to be rejected online, are erroneously merged offline. The merged cluster is outside the MinXY or MaxR rejection area.
- Two clusters, all of them passing the MinXY cut offline, are merged online. The merged cluster falls into the MinXY rejection area.
- Clusters that are single-hit offline, and therefore ignored, are identified as multi-hit online. If these clusters are within the MinXY or MaxR rejection area, the event contributes to the inefficiency.

- Accidental events recorded in the same 512 ns window of the L1-triggered event. If the RMS of the hit timing in the accidental event is smaller than the one of the triggered event, the offline reconstruction will discard the triggered event and keep the accidental one. The L1-trigger event satisfies the criteria for being rejected, but the accidental event considered offline does not.

Representative event displays for each of these cases are shown in Appendix B. The potential impact of these events on the offline analysis is also discussed.

10.4 Efficiency and rejection factors measurement during normal physics runs

At any time during a physics run, the HLT randomly selects 1% of the events and fully records them despite their trigger type or selection result. These HLT-unbiased events are used offline after a run to monitor the HLT selection efficiencies, following the same procedure as in the *HLT-tag* runs discussed in the previous section. The results obtained before physics runs, with the special tag runs and during usual physics DAQ are compared and discussed in section 12.1.

Chapter 11

Evaluation of the HLT Performance

Evaluating the performance of the HLT involves benchmarking both the 40 Gbps data capture stage and the event processing stage. Both stages work independently from each other and can be studied separately. The 40 Gbps packet capture stage is treated in section 11.1. The performance of the event processing stage is discussed in section 11.2.

11.1 Performance of the 40 Gbps packet capture stage

The performance of the 40 Gbps packet capture is carefully measured before the beam-time with the two Computing Nodes installed at Osaka. One of them sends dummy data at any arbitrary speed up to 40 Gbps, and the other one runs the HLT and receives the data as it would in normal runs. The results obtained from these tests are then verified on-site, with an actual OFC-II module running a special firmware that allows it to send dummy data at an arbitrary speed.

At the HLT, packets can be dropped either upstream of the NIC (including at the sender, through the cables, and in the transceivers) or at the Netmap buffers, if the packets cannot be copied fast enough to the HLT buffers. Both the NIC and the HLT software retrieve packets one by one. To minimize the packet loss at high rates, even more significant than the bandwidth in Gbps is the amount of transmitted packets per second. At the same bandwidth, transmitting fewer larger packets results in higher efficiency than transmitting many smaller ones.

The packet sizes in KOTO are fixed; Each event consists of 67 8800-byte packets and one 7840-byte packet. During the tests, however, the packet capture efficiency is measured against multiple data rates and packet sizes. This allows us to estimate how far the working point of the HLT (in the *bandwidth / packet size* plane) is from the zone in which packet drop starts being high. This gives a measure of how confident we can be in the actual HLT performance at J-PARC.

Figure 11.1 shows the results of a test consisting in transmitting 10^6 packets at different data rates and packet sizes. The test is repeated twice, and the average loss is shown. The packet size and the upper limit of the bandwidth are set at the sender. The horizontal axis in the figure shows not the set but the actual *measured* bandwidth, which in most cases coincides with the set value. Only at high bandwidths and small packet sizes, the sender struggles to keep up with the set rates.

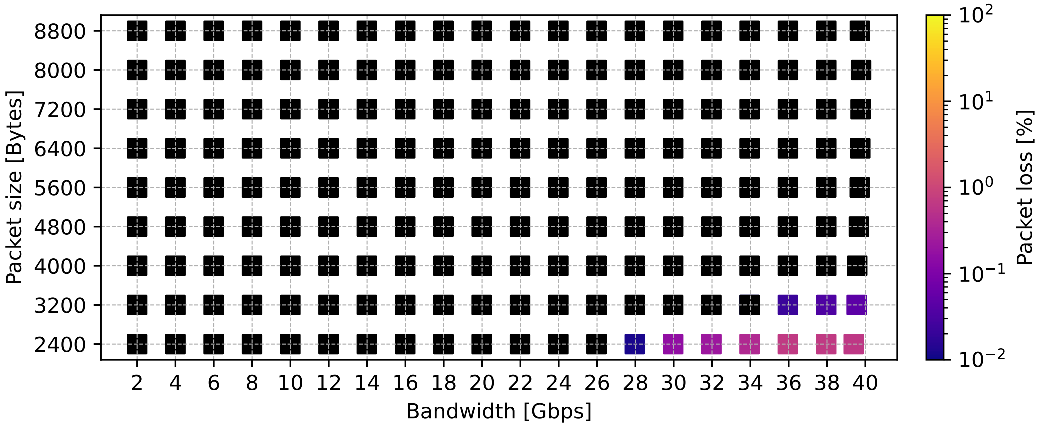


Figure 11.1: Packet loss as a function of packet size and data rate. Black squares show points where the average packet drop was zero.

The average bandwidth expected during physics runs, and the maximum bandwidth supported by the upstream DAQ system were given in section 5.1.1. At the upstream DAQ limit of 50 kEvents per spill set by the OFC-I, the average bandwidth per node would be 20.5 Gbps. If the 50 kEvents per spill limit is solved in the future, the bandwidth could theoretically reach up to 36 Gbps. During physics runs in 2024, the average bandwidth received per HLT computing node was 10.7 Gbps.

The packet size in KOTO is fixed at 8800 bytes[1], corresponding to the top column in Fig. 11.1. For any bandwidth, no packet loss was observed within the total of 2×10^6 transmitted packets. At the 90% confidence level, the packet loss is estimated to be less than 0.00015% satisfying the 0.01% requirement set in section 5.1.1.

The performance of the packet capture stage highly relies on the NUMA architecture. In section 6.4, we highlighted the importance of explicitly setting the CPU affinity of the CPUs involved in the packet capture, as well as explicitly allocating memory in the regions with the fastest access by those threads. To demonstrate the importance of this, the test shown in Fig. 11.1 was repeated after an incorrect memory allocation and CPU affinity setting. The results are shown in Appendix F.

The result obtained in this section gives confidence in the performance of the HLT packet capture stage during beam-time. However, the test does not account for the loss that could come from the OFC-II itself, the 300 m of fiber separating it from the HLT, or the transceivers

[1] The last packet size in KOTO is 7840 bytes, but the impact in the overall packet rate is insignificant.

at the OFC-II or the HLT among other factors. The actual event loss due to packet drop measured during the 2024 beam-time will be discussed in the following paragraphs.

Correspondence between packet loss and event loss

Events with missing packets at the HLT cannot be reconstructed. These events are discarded despite the number of missing packets.

If individual packets are lost randomly, all lost packets will likely belong to different events. In this scenario, the event loss can be as high as 68 times the packet loss. This could happen, for instance, if the packet loss comes from a low-power transceiver or a defective cable.

In another scenario, many packets are rejected in bursts, either by the NIC of the Computing Nodes or by the HLT software. For instance, multiple consecutive packets would be rejected if the Netmap buffers get full because the HLT cannot keep up with the speed. Packets at the bottom-right in Fig. 11.1 are dropped because of this reason. In this case, the event loss could get as low as the packet loss.

During physics runs, the HLT reports the number of events that were discarded due to missing packets. However, it cannot keep track of the events that were entirely lost before reaching the HLT software. The total event lost due to packet loss is estimated by comparing the events processed by the HLT with the events sent by the OFC-II, run by run. The results are shown in Fig. 11.2. Excluded from the comparison are all spills in which event loss was reported to have originated by reasons other than packet loss. These include the saturation of the HLT buffers due to the limited disk-write speed at the Disk Node among others. The different reasons why the HLT lost events in the 2024 beam-time will be reported in section 12.3.

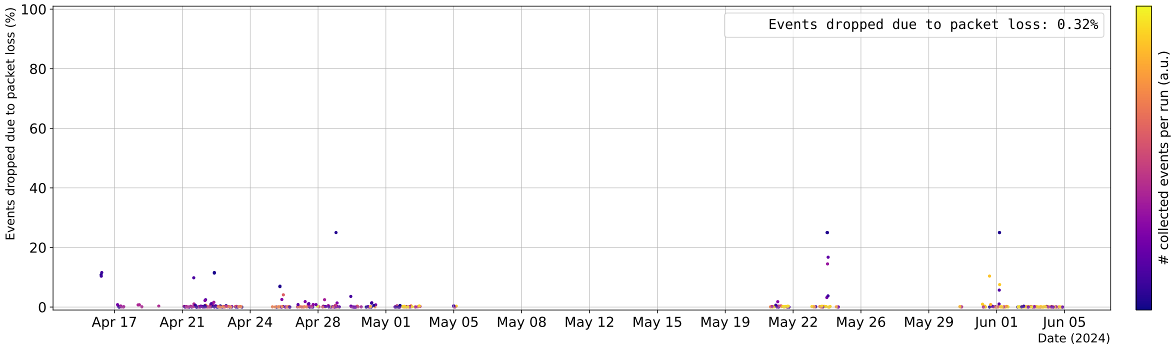


Figure 11.2: Event loss due to packet loss during physics runs in 2024. Each point represents a different run. The loss is given as a percentage of the total number of events sent by the OFC-II.

In most runs with high event loss, the corresponding packet loss is concentrated in a single Computing Node. It is believed to have originated from a weak signal consequence of using

150 m-rated transceivers to cover the 300 m distance between the OFC-II and the HLT[2]. The three runs with a loss equal to 25% in Fig. 11.2 are the result of one of the four Computing Nodes not receiving any packets at all. The total event loss due to packet drop during the 2024 beam-time was measured to be 0.3%.

11.2 Event processing stage

The performance of the HLT event processing is evaluated with the Nvidia Nsight Systems profiler, a dedicated tool for GPU profiling. The analysis is performed offline, with the OFC-II emulator sending data from real physics runs to the HLT. However, the OFC-II emulator cannot reach the high speeds the real OFC-II does. To simulate a high event rate, the processing stage at the HLT is blocked until all incoming data has been captured. From the HLT point of view, this situation is equivalent to receiving a spill instantaneously.

The tests were performed before the beam-time, by feeding one HLT node with 5000 events. This is equivalent to what each of the four nodes targeted per spill would receive in a 20 kEvent spill. During the tests, the Computing Node processed the events and sent them to a Disk Node in the same way it is done during physics runs. Sending the data to the Disk Node is limited by the 10G network. This stage was included in the tests for reference.

The complete processing of these 5000 events was finished in 640 ms. This translates into a throughput of $7.8 \cdot 10^3$ events per second per Computing Node. The requirements for the HLT in terms of throughput have been justified in section 5.1.2. To keep up with the maximum OFC-I rate of 50 kEvents/spill assuming four Computing Nodes, each node would need to be able to process 3 kEvents per second. The obtained throughput is 2.6 times higher than that.

Profiling results

In practice, the number that matters is the overall HLT performance that has been presented above. However, it is interesting to know where the HLT, and the GPUs in particular, spend most of their time. An idea of how much time is taken at each stage of the HLT GPU processing is presented in Fig. 11.3.

[2] The 150 m transceivers did not give any problem during tests before beam-time, nor during most physics runs. Thus it was decided to keep them until the end of beam-time. They will be replaced before the next scheduled beam-time by 300 m capable transceivers.

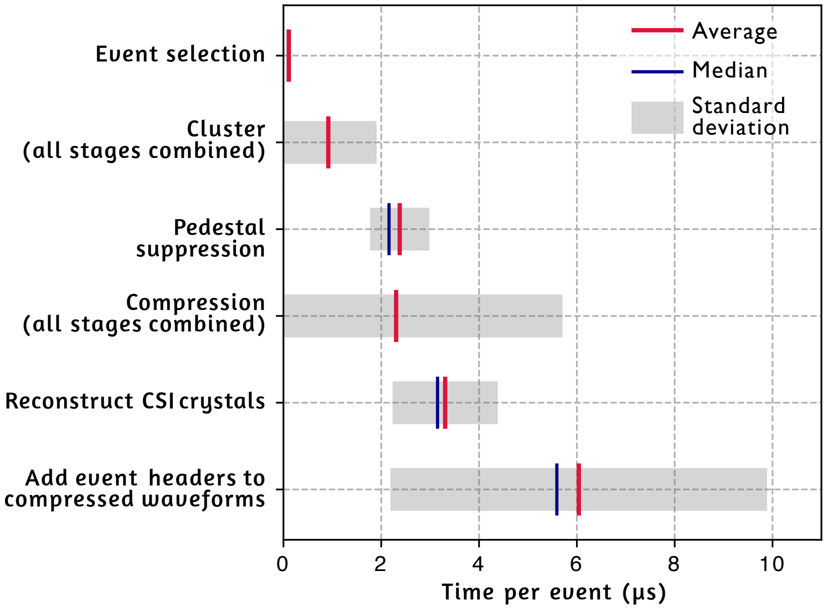


Figure 11.3: Time taken by each stage of the GPU processing at each HLT node.

The time *per event* in Fig. 11.3 is calculated from the time *per event block* reported by the Nvidia profiler. It does not account for the fact that multiple event blocks can run in parallel, and therefore should not be taken as an absolute value.

The performance of each GPU stage is mostly limited by the amount of memory transactions that it involves. The slowest stage in Fig. 11.3, adding event headers to the compressed waveform array, involves just copying them from one array to another. On the other hand, clustering, despite its complexity, is also one of the fastest stages. This is because most of its complexity lies on calculations, with few interactions with the GPU's global memory.

Memory intensive stages also have the largest standard deviation, as each kernel's memory bandwidth is highly affected by other memory operations that may be running on the GPU at the same time.

Figure 11.3 only covered the kernels running on the GPU. In Fig. 11.4, we show how the time spend in running kernels compares to other operations involved in GPU computing. As seen in the figure, running kernels represents just a small fraction of the total time consumed by the GPU. Memset operations involve resetting memory before processing a new event block. Stream synchronizations involve waiting for all threads in a block to finish one stage before proceeding to the next one. Memory copies, which take most of the time, involve retrieving raw data from the CPU and sending back compressed events.

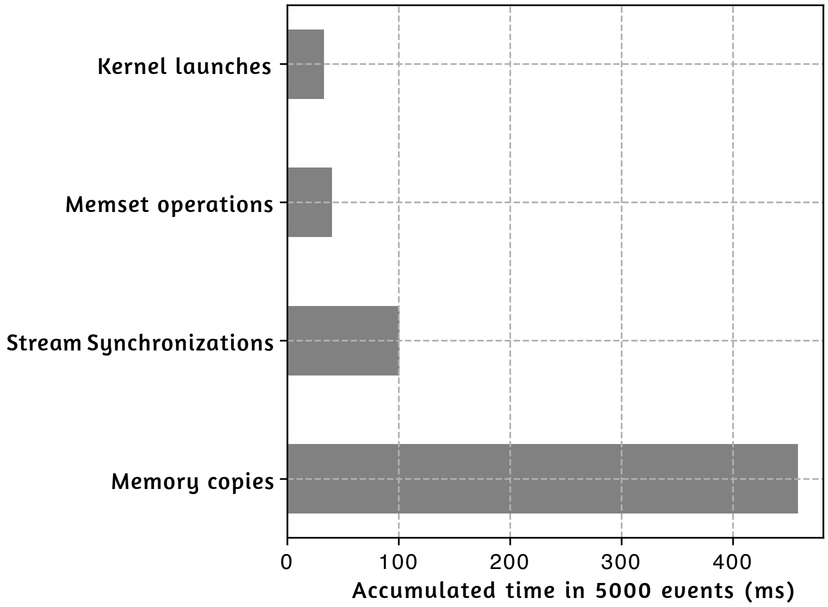


Figure 11.4: Accumulated time on different operations involved in GPU computing at the KOTO HLT.

Most memory copies are masked behind much slower CPU processing, as was explained in section 7.3. The accumulated time spent by the CPU in preparing 5000 events for the GPU was 3.2 s[3]. This is currently the slowest stage of the HLT. Sending events to the Disk Nodes, which is hard-limited to 10 Gbps, is still three times faster than preparing raw data on CPU. From the total time the HLT spent processing 5000 events, only 0.5% was spent by the GPUs in processing data.

[3] This is an *accumulated* time. Up to eight CPU threads can work in parallel, and 5000 events can be processed in 640 ms.

Chapter 12

Results

12.1 Efficiency of the online physics event selection

The efficiency of the HLT selection cannot be monitored in real-time during a run, as calculating it requires the results from the offline analysis. Offline, it can be calculated from the HLT-unbiased events, which represent 1% of the total events collected in each run.

Figure 12.1 shows $\mathcal{E}_{MinXY}$, $\mathcal{E}_{MaxR}$, and $\mathcal{E}_{TotalE}$ for the 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ triggers, as estimated before physics runs, measured during special HLT-tag runs before enabling the HLT selection, and measured with the 1% of HLT-unbiased events during normal physics runs towards the end of beam-time. The values predicted before the runs (red) were presented in section 10.2. The values measured during HLT-tag runs (green) are the ones presented in section 10.3.

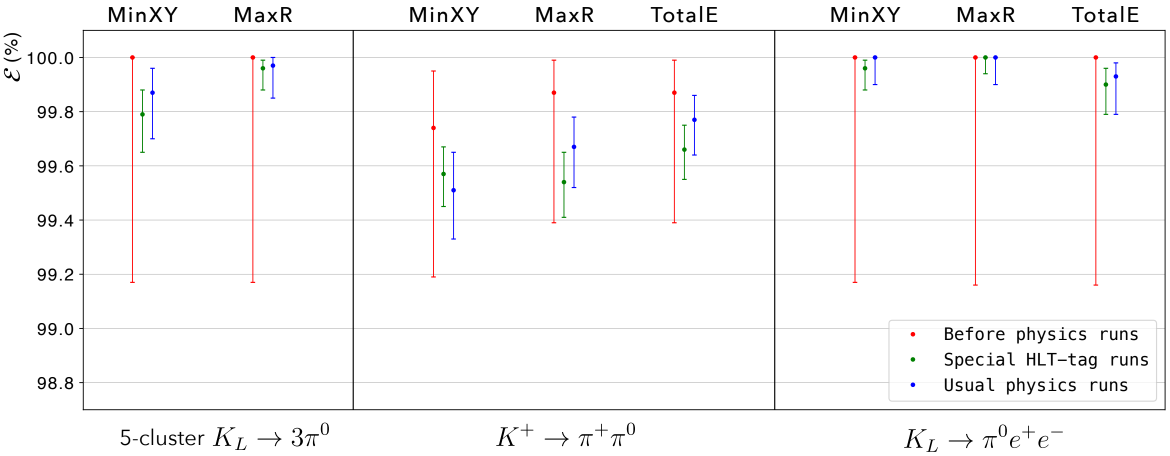


Figure 12.1: Efficiency of all cuts applied during physics runs, predicted before the runs, measured during HLT-tag runs before enabling the HLT selection, and measured with HLT-unbiased events taken during normal physics runs around the end of beam-time.

The selection criteria of the 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ triggers were decided to keep $\mathcal{E}_5\gamma$, $\mathcal{E}_K^+$, and $\mathcal{E}_\pi^0 e^+ e^-$ around 99%. The efficiencies measured at the end of the beam-time with HLT-unbiased events were consistent with the ones predicted before enabling the HLT selection offline and with the special HLT-tag runs, within the statistical uncertainties as shown in Fig. 12.1.

The combined efficiencies of the 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ event selections, obtained with HLT-unbiased events during normal physics runs (blue in Fig.12.1), are shown in Table 12.1.

Table 12.1: Combined efficiency and rejection power of all HLT cuts applied to the 5γ, K^+ and $K_L \to \pi^0 e^+ e^-$ triggers.

Trigger	$\mathcal{E}$	$\mathcal{R}$
5-cluster $K_L \to 3\pi^0$	$(99.83^{+0.10}_{-0.18})\%$	$(23.0 \pm 0.3)\%$
$K^+ \to \pi^+\pi^0$	$(98.95^{+0.21}_{-0.24})\%$	$(20.0 \pm 0.2)\%$
$K_L \to \pi^0 e^+ e^-$	$(99.87^{+0.08}_{-0.14})\%$	$(16.0 \pm 0.3)\%$

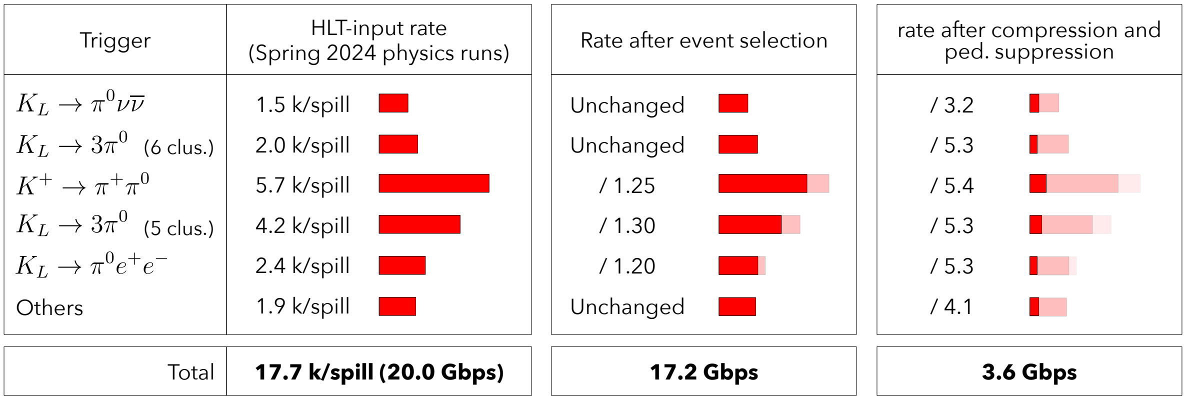
These rejection factors were enough to keep the HLT output data rate in physics runs, discussed in the next section, within the J-PARC to KEK transfer bandwidth limitation. Further selection criteria, such as the powerful COE cut in 5γ and $K_L \to \pi^0 e^+ e^-$ events did not need to be applied.

12.2 Data reduction at the HLT

One of the main goals of the new HLT is to reduce the data rate below the J-PARC to KEK bandwidth of 4 Gbps. The data reduction from pedestal suppression and waveform compression at the HLT has been discussed in section 8.5. The reduction factors from event selection were shown in Table 12.1.

The overall data reduction combining all triggers is summarized in Fig. 12.2. The first column shows the average rate at the HLT input, in kTriggers per spill, for each trigger. The red bars are in length proportional to each trigger rate. At the bottom row, the total rate at the HLT input is shown in both kTriggers per spill and Gbps. The conversion is done by taking into account the size of an event and the full length of the spill cycle, 4.2 s.

The second column shows the rate after event selection, which is reduced according to the $\mathcal{R}$ factors shown in Table 12.1. For example, if $\mathcal{R}_{K^+}$ is 20%, the rate after K^+ selection is $1/(1 - \mathcal{R}_{K^+}) = 1.25$ times lower than the rate before the selection. The data size of the K^+ sample is reduced by the same factor. The third column in Fig. 12.2 shows the rate after pedestal suppression and waveform compression. Between the second and the third columns the trigger rate is kept constant, but the data size is reduced as the events are compressed. The reduction factors for events with and without pedestal suppression enabled are extracted from Fig. 8.22. The last column shows the rate at the HLT output, after all data reduction stages.



Trigger	HLT-input rate (Spring 2024 physics runs)	Rate after event selection	rate after compression and ped. suppression
$K_L \to \pi^0\nu\overline{\nu}$	1.5 k/spill	Unchanged	/ 3.2
$K_L \to 3\pi^0$ (6 clus.)	2.0 k/spill	Unchanged	/ 5.3
$K^+ \to \pi^+\pi^0$	5.7 k/spill	/ 1.25	/ 5.4
$K_L \to 3\pi^0$ (5 clus.)	4.2 k/spill	/ 1.30	/ 5.3
$K_L \to \pi^0 e^+ e^-$	2.4 k/spill	/ 1.20	/ 5.3
Others	1.9 k/spill	Unchanged	/ 4.1
Total	**17.7 k/spill (20.0 Gbps)**	**17.2 Gbps**	**3.6 Gbps**

Figure 12.2: Data reduction from an average physics run in the 2024 beam-time.

After selection and compression, the average data rate between the Computing Nodes and the Disk Node is reduced to 3.6 Gbps, 18% of the rate at the HLT input. This is below the 4 Gbps requirement set by the limited bandwidth between J-PARC and KEK.

12.3 Efficiency of the new DAQ system

In this section, we present the efficiency of the complete KOTO DAQ system during the last beam-time, run by run. The most relevant issues affecting the efficiencies, and the corresponding solutions, are also discussed. We will first look at the upstream DAQ system (Top CDT, OFC-I and OFC-II). Then we will focus on the HLT, and finally we will combine the results into an overall DAQ efficiency. At the end of the section, we will study any possible correlation between bad inefficiency and low physics data quality.

Upstream (Top CDT, OFC-I, and OFC-II) DAQ efficiency

To compute efficiency of one stage of the DAQ, we compare the number of events at its input to the number of events at its output. If both numbers are equal, the efficiency is 100%.

The efficiency of the most upstream stage of the DAQ requires special handling, as the number of events at its input is not known. The most upstream counter we have record of is the number of events passing the Level 2 trigger, reported by the Top CDT module spill by spill in every run. However, error or busy signals coming from the ADCs or from any OFC module will make the Top CDT module stop issuing triggers until the next spill. The number of events lost because of this is not known, but can be estimated.

The total number of triggers that Top CDT would have issued per run if there had been no problems is estimated run by run from distributions like the one shown in Fig. 12.3. Figure

12.3 shows the distribution of the number of triggers per spill issued by Top CDT within a particular physics run.

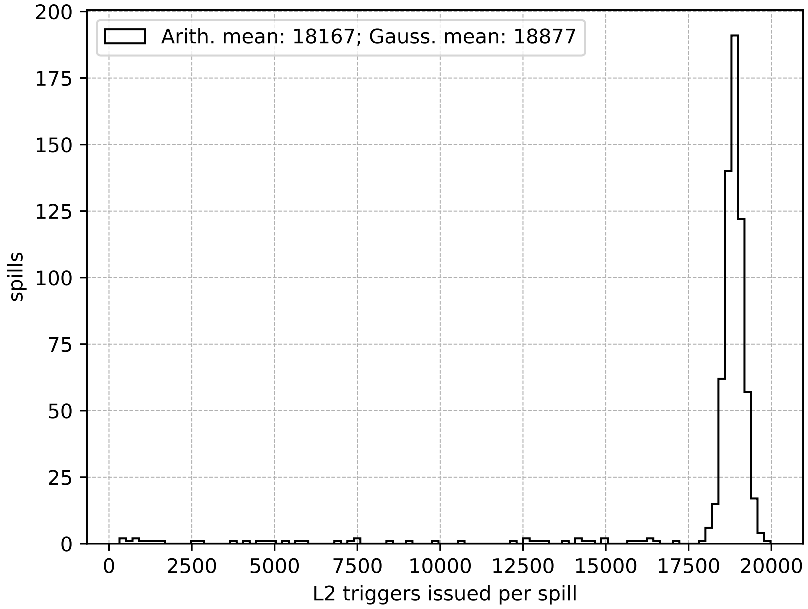


Figure 12.3: Example of a Level 1 accepted distribution, including all spills from a run.

Most spills scattered across the horizontal axis were affected by upstream DAQ errors. These spills are too few to affect the shape of the gaussian distribution observed around 18 kTriggers/spill. The expected number of events that would have been collected in this run if there had been no errors can be estimated as the Gaussian mean times the number of spills in the run. The actual amount of collected events is the integral of the histogram in Fig. 12.3, or equivalently its arithmetic mean multiplied by the number of spills.

The efficiency of the upstream DAQ can be obtained by dividing the Arithmetic mean of the Level 1 accepted distribution by its Gaussian mean. If no errors have occurred during the run, the Gaussian mean and the arithmetic mean will coincide, and the efficiency of the upstream DAQ will be 100%[1]. It is important to note that some spills might have a low L1 count due to an actual low number of protons delivered by the accelerator. These spills are excluded from the efficiency calculation.

Fig. 12.4 shows the upstream DAQ efficiency calculated as described above.

[1] Taking the arithmetic mean as exact, the uncertainty of the efficiency is given by $A/G^2 \cdot \sigma$, where A is the arithmetic mean and G the gaussian mean. The error of the efficiency of the run in Fig. 12.3 is 1%.

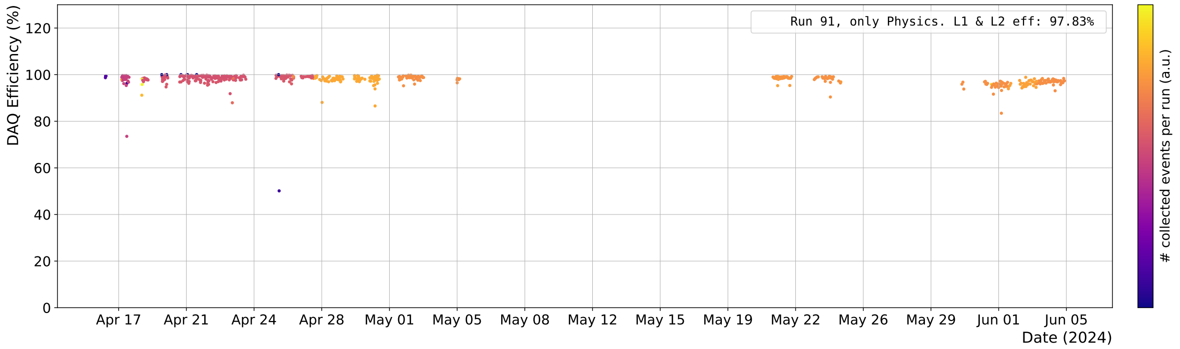


Figure 12.4: Upstream DAQ efficiency in each run during the 2024 beam-time. Error signals from ADCs, OFC-Is, and OFC-IIs contribute to the inefficiency at this stage.

The upstream DAQ efficiency was high and stable during most of the beam-time. Its uncertainty in most runs, coming from the standard deviation of the L2-accepted gaussian distribution, is in the order of 1%. The efficiency drops are originated from errors reported by OFC-I or OFC-II modules, described in sections 4.7.1 and 4.7.2 respectively. The effect of this errors clearly manifests in the L1-accepted distribution as spills with low trigger counts, and cannot be attributed to statistical fluctuations. Most of these errors are automatically fixed after simple resets done between spills. Only in a few runs we observed large efficiency drops that required manual intervention. These include, for example, an ADC failure requiring a module replacement.

The drop in efficiency towards the end of the beam-time is under investigation at the time of writing this thesis.

Not included in Fig. 12.4 is the dead time of the online clustering performed at the Clustering OFC. This dead-time is well understood, and known to cause the loss of 1.5% of the events every spill.

HLT efficiency

The HLT efficiency is computed as the ratio between the number of events recorded offline and the number of events sent out by the OFC-II. However, issues when reading and resetting counters from the VME computer at the OFC-II crate made the OFC-II counters unreliable in some spills[2]. To account for this, only the spills in which the L2-issued triggers and OFC-II output counters are equal are included in the calculation.

The HLT efficiency is shown in Fig. 12.5.

[2] In most cases, some events processed in a spill were included in the next spill's counters.

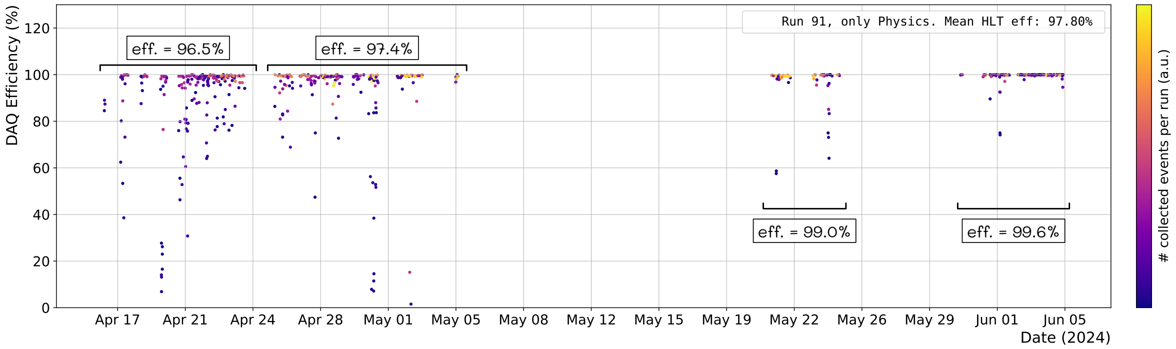


Figure 12.5: HLT efficiency run by run during the 2024 beam-time. Only physics runs are included.

The plot in Fig. 12.5 is divided into four sections. Different factors contributed to the inefficiency of the HLT in each section.

The largest cause of event drop at the HLT, and the DAQ bottleneck in this beam-time, was the limited read/write speed to and from the disks at the Disk Nodes, when being accessed simultaneously by the HLT and the program transferring data to KEK. Due to time constraints, the HLT and the transfer program could not be tested together before the beam-time to confirm the issue. This was solved before the third section in Fig. 12.5 as explained in section 9.3.

At the beginning of beam-time, multiple events were lost at the switch between OFC-II and the HLT, not even reaching the Computing Nodes. The issue was mitigated by bypassing the switch and connecting the OFC-IIs directly to the HLT nodes. The actual cause of the issue is not known at the time of writing this thesis.

Some other unpredicted issues affected the efficiency of the HLT during the first half of the beam-time. A total of 13 runs were affected by a HLT software issue that caused the corruption of large portions of the data produced at the Computing Nodes. This issue is further discussed in appendix C.1. The impact of this issue into the HLT efficiency is estimated to be less than 0.1%. This issue, together with others affecting similar amount of events, were gradually solved during the first half of the beam-time and during the long break before the second half. During this time, the software in the Disk Nodes was also modified to produce fewer and larger files per spill, helping to make the transfer program and the offline storage of binary data more efficient.

Eventually, these improvements increased the efficiency of the HLT from 97.4% in the second period to 99.0% in the third period, as shown in Fig. 12.5. During the third period, it was further discovered that the HLT was accidentally being prematurely terminated at the end of each run before it had time to write the last $1 \sim 5$ spills to disk. In a normal 850-spill run, this represents a $< 0.6\%$ event loss. This issue was quickly fixed, pushing the efficiency of the HLT during the last period of the beam-time to 99.6%, equivalent to a total event loss of 0.4%. The reasons for this remaining small event loss at the last stage of the beam-time are discussed in section C.2.

Except for the disk read/write bottleneck, all major issues affecting the HLT efficiency were permanently solved. Upgrading the Disk Node hardware before the next run will tackle the disk bottleneck.

Overall DAQ efficiency, and the impact in the physics data quality

The overall DAQ efficiency is the combination of the Upstream and the HLT efficiencies. It is computed by dividing the number of events collected offline by the number of events expected to have been received each run at the L1. The overall DAQ efficiency in the 2024 beam-time is shown in Fig. 12.6.

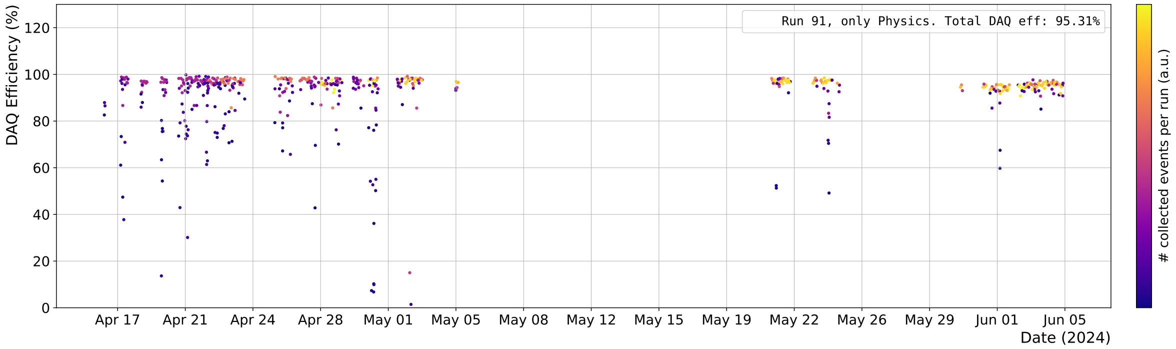


Figure 12.6: Overall DAQ efficiency in the 2024 beam-time.

Most of the low DAQ efficiency at the first half of the beam-time is originated by the HLT. Most of the low efficiency at the last portion of the beam-time comes from a large error rate at the upstream DAQ, whose cause is still to be investigated.

The potential correlation between low DAQ efficiency and corruption of the collected data was checked by comparing a sample of events extracted from spills where the efficiency was 100% to another sample of events extracted from spills with low efficiency. The results are shown in Fig. 12.7 for different ranges of DAQ efficiency[3].

[3] The K_L mass reconstruction from $K_L \to 3\pi^0$ events is detailed in appendix J. The reason for the long tail of the K_L mass distribution is also explained in the appendix.

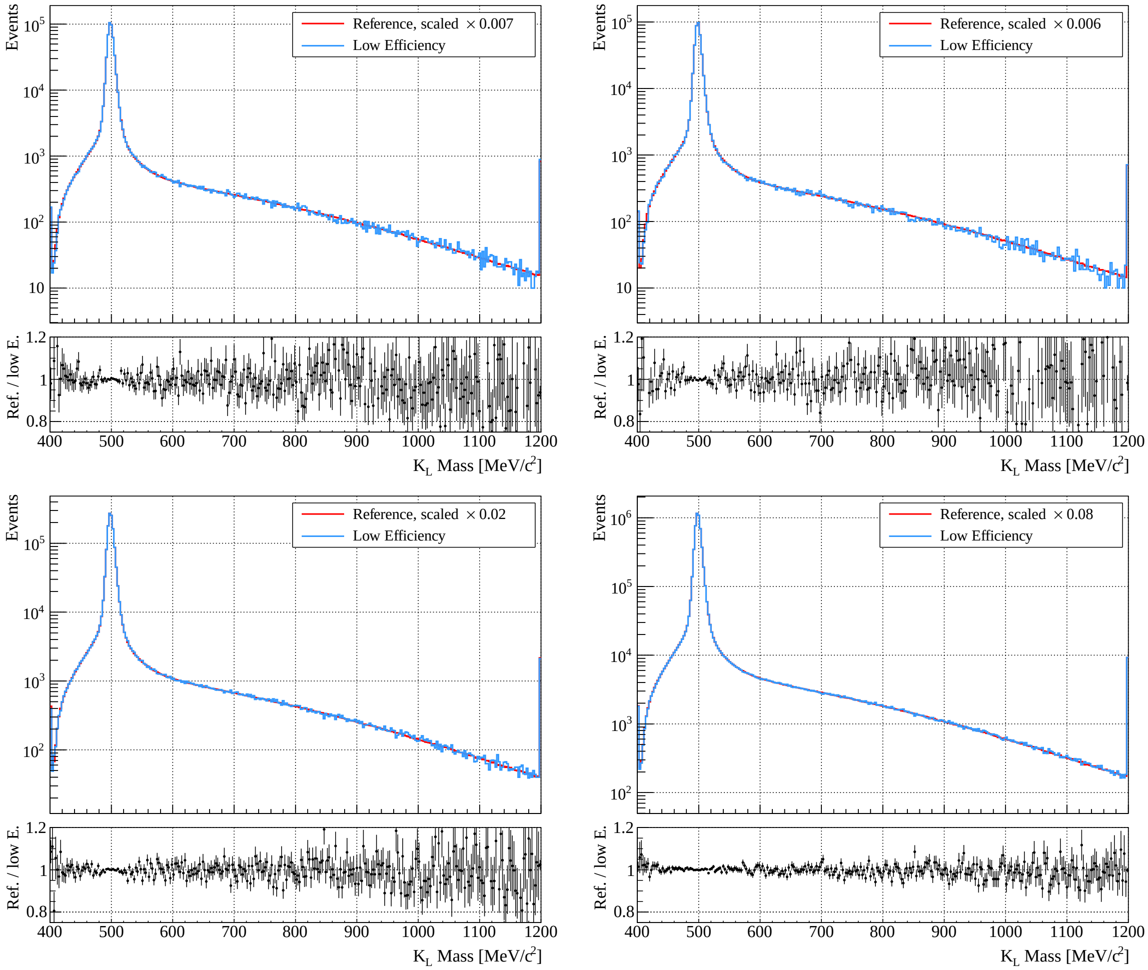


Figure 12.7: Comparison between a sample of events from spills with 100% DAQ efficiency and a sample of events from spills with DAQ efficiency below 40% (top left), between 40% and 60% (top right), between 60% and 80% (bottom left) and between 80% and 100% (bottom right). The distributions show the reconstructed K_L mass, obtained from $K_L \to 3\pi^0$ events. Events from spills with 100% DAQ efficiency are shown in red. Events from spills with lower DAQ efficiency are shown in blue.

The distributions above include data from a set of physics runs collected during the first half of the 2024 beam-time. To check the effect of low DAQ efficiency on the quality of the collected data, a chi-squared test is performed between the reference and the low-efficiency histograms in each plot. The test is performing considering the weights of the bins in the scaled histograms, according to the procedure described in [37]. The number of bins in each histogram is set so that all bins have at least 10 entries. The maximum number of bins per histogram is set to 500. The results of the Chi-squared test are shown in Table 12.2.

Table 12.2: Results of the Chi-squared test between the reference and the low-efficiency samples in Fig. 12.7. The degrees of freedom (ddof) are given by the number of bins in the histograms minus one. The critical χ^2 values are calculated for a 95% confidence level.

DAQ efficiency	ddof	χ^2	critical χ^2
100% vs. $< 40\%$	299	263	340
100% vs. 40% – 60%	269	276	308
100% vs. 60% – 80%	299	291	340
100% vs. 80% – 100%	299	302	340

A χ^2 below the critical χ^2 value indicates that low DAQ efficiency does not have a significant impact on the quality of the collected data. Therefore, collected events can be used for physics analysis offline despite the DAQ efficiency of the spills or runs they belong to.

12.4 Throughput of the HLT system

The evaluation of the throughput of the HLT system was discussed in section 11. Particularly, the HLT throughput was compared to the maximum KOTO DAQ bandwidth of 50 kTriggers per spill. Even with only four nodes, the HLT Computing Nodes could process events at a rate 2.6 times higher than the requirement to cope with an input of 50 kTriggers per spill. The Computing Nodes could also capture OFC-II data at any rate up to 40 Gbps with a packet loss of less than 0.01%. In actual physics runs, the event loss due to packet loss in the OFC-II – HLT network was 0.3%. These results set the Computing Nodes ready for any future beam intensity upgrade or upstream DAQ upgrade, or for any future additions to the trigger increasing the data rate up to 50 kTriggers per spill.

The limited disk read/write speed at the Disk Node was found to be the HLT bottleneck in 2024. The maximum event rate in physics runs was not reached, but this bottleneck limited the acquisition of Aluminum target data, which does not include event selection or pedestal suppression, to 22 kEvents per spill.

12.5 Physics significance

The new KOTO DAQ system after the upgrade presented in this thesis has been able to collect for the first time 5-cluster $K_L \to 3\pi^0$, $K^+ \to \pi^+\pi^0$ and $K_L \to \pi^0 e^+ e^-$ candidate decays together with the triggers directly related to the $K_L \to \pi^0 \nu\overline{\nu}$ search.

One of the main physics goals of the 2024 runs was the collection of 5-cluster $K_L \to 3\pi^0$ data to improve the estimation of some detector inefficiencies and ultimately to reduce the uncertainty of the number of $K_L \to 2\pi^0$ events that become a background of $K_L \to \pi^0\nu\overline{\nu}$.

During physics runs, 24% of the total event rate was taken by the 5-cluster $K_L \to 3\pi^0$ trigger. The accumulated 5-cluster $K_L \to 3\pi^0$ yield in 2024 has been 9.4 times larger than in the previous long beam-time in 2021, despite the 2024 beam-time delivering to KOTO around half of the kaon decays that were delivered in 2021. The offline analysis is ongoing at the time of writing this thesis, but we expect the systematic uncertainty of the $K_L \to 2\pi^0$ background prediction in the 2021 data analysis (shown in table 2.1) is expected to be reduced by a factor of 3 due to the > 9 times larger collected event yield. This will have an impact in a future Upper Limit of the $K_L \to \pi^0\nu\overline{\nu}$ search if the number of observed events in the Signal region is non-zero, as explained in the next paragraph.

The Upper Limit in each KOTO analysis is obtained from the SES, given the background expectation and number of observed events inside the signal region. The calculation is done assuming Poisson statistics [38]. If the number of observed events is zero, as it has been the case in the 2021 analysis, the Upper Limit is independent of the background prediction. If the number of observed events is non zero, the Upper Limit naturally increases the smaller is the background prediction, i.e. the higher is the likelihood that the observed events are signal. However, for a given number of observed events, and predicted background, the 90% CL Upper Limit will be higher the larger is the uncertainty in the background prediction.

The K^+ trigger in 2024 accounted for 32% of the event rate during physics runs. This trigger was also collected in 2021 with a 60 kW beam. The trigger conditions have been kept the same in 2024. The trigger rate has only increased proportionally to the beam intensity increase[4] from 60 kW to 80 kW. The collected K^+ candidates have been already used to estimate the performance of the permanent magnet added to the KOTO beamline to sweep K^+ away from the KOTO detector. The distribution of the reconstructed K^+ mass from K^+ candidate decays with the magnet (2024) and without it (2021) is shown in Fig. 12.8.

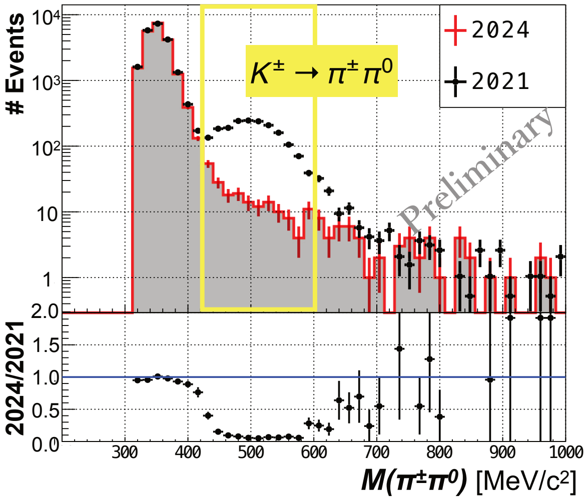


Figure 12.8: Distribution of the reconstructed K^+ mass from K^+ candidate decays, in 2021 (without the permanent magnet) and 2024 (with the permanent magnet). Borrowed from the analysis performed by a collaborator [39].

[4] The K^+ flux into the KOTO detector has been reduced by a factor of 10 by the addition of the permanent magnet to the KOTO beamline. However, the vast majority of $K^+ \to \pi^+\pi^0$ candidates in the K^+ trigger are actually $K_L \to \pi^+\pi^-\pi^0$, that are not affected by the magnet.

The nominal K^+ mass is 500 MeV, and corresponds to the peak observed in 2021 (black in the figure) around that value. A detailed analysis is still ongoing, but it has been confirmed that the ratio of the $K^\pm$ flux to the K_L flux has been reduced by a factor of 10 with respect to 2021 thanks to the effect of the new permanent magnet.

Finally, $K_L \to \pi^0 e^+ e^-$ events have been collected continuously during physics runs in 2024. The offline analysis of these data is expected to provide feedback towards the development of KOTO-II, which is considering this decay as one of its main physics goals.

Thanks to the upstream DAQ upgrade and to the HLT being capable to capturing data at up to 40 Gbps, the current KOTO DAQ system can trigger and transfer to the HLT Computing Nodes up to 50 kEvents per spill. This is enough to accommodate a 100 kW beam with the current trigger configuration, and to add new entries to the trigger menu if it becomes a need in the future.

Chapter 13

Discussion

13.1 Bottleneck of the current DAQ and the HLT system

Excluding the 4 Gbps link between J-PARC and KEK, the bottleneck of the DAQ system in 2024 was the disk access at the Disk Nodes. Before beam-time, the read/write speed was measured to be above 20 Gbps when using a single RAID 0 disk array composed of 8 physical disks. Simultaneously reading and writing the disks was known to not be possible, but the impact of the transfer program on the HLT performance was underestimated, and could not be tested before beam-time. The maximum event rate in physics runs allowed by this bottleneck is known to be above 18 kEvents per spill, although the exact number is unknown.

In future runs, the disk R/W bottleneck will be completely removed by either adding a second Disk Node (effectively doubling all R/W speeds) or by installing SAS disks in place of the current SATA array. As shown in Fig. 13.1, a single SAS disk already provides a writing speed 1.6 times larger than the RAID 0 composed of four SATA disks used during DAQ in 2024. By replacing all current disks with SAS disks and maintaining the 4-disk RAID 0 configuration of the Disk Nodes, the R/W speed can be expected to increase[1] by a factor of $1.6 \times 4 = 6.4$. This is 3.2 times larger than the factor of 2 that would be provided by adding a second Disk Node with SATA disk arrays.

[1] The actual increase in speed has not been measured due to the lack of SAS disks.

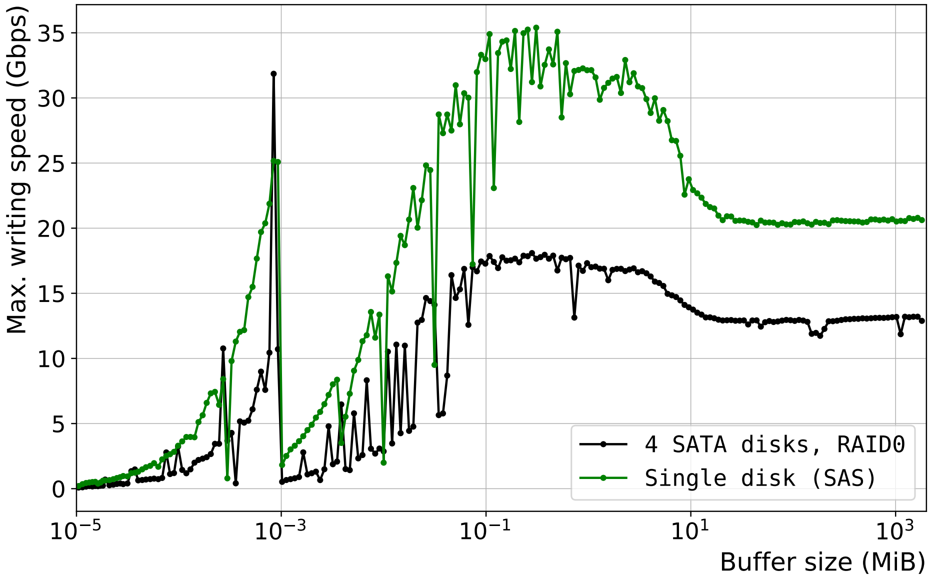


Figure 13.1: Performance comparison, between each of the two RAID 0 SATA arrays currently configured on the Disk Node and a single SAS disk. The shared area shows the standard deviation of four measurements. Each point shows the average of the four measurements.

The next HLT bottleneck is the maximum rate at the input of the Disk Nodes. The maximum speed at which the HLT can transmit through that link is 18.8 Gbps (Fig. 9.4), but the actual data rate is lower during physics runs due to the following reasons.

First, before each compressed data buffer is transferred, MPI sends to the disk node a small packet containing the size of the buffer. This information is needed at the Disk Nodes to determine the destination of the upcoming data buffers before actually receiving them. Second, upon being notified that a buffer is ready to be sent, the disk nodes need some time to determine its destination before actually starting to receive it, which further contributes to reducing the effective data rate.

This bottleneck will be removed by the addition of a second Disk Node, which will effectively double the bandwidth through the Computing Nodes – Disk Nodes link. At the Computing Nodes, the option of sending fewer and larger data buffers can also be explored in the future.

The next bottleneck is at the OFC-I, limiting the maximum DAQ rate to around 50 kEvents per spill. This bottleneck is not expected to be reached. Overcoming it would require major upgrades that are currently not under consideration.

13.2 Future upgrades to the HLT system

Multiple upgrades can improve the HLT in the near future, both from the hardware and software perspectives. The only hardware upgrade planned for the HLT is the addition of a second Disk Node, which has been discussed in section 9.3.

The most relevant upgrades planned for the HLT software are listed in this section.

13.2.1 More flexible compression algorithm

The basic working principle of the current waveform compression algorithm has been discussed in section 8.4.1. The compression factor that it provides depends only on the amplitude of each waveform, the difference between its highest and lowest sample. The highest compression factors come from flat waveforms without hits. However, most of these waveforms are pedestal-suppressed and do not need to be compressed. On the other side of the spectrum, waveforms with amplitudes larger than $2^8 - 1 = 255$ ADC counts are not compressed. Although these waveforms represent a small fraction of the total waveforms in normal events, there is technically a small compression gain that could be obtained from them (see Fig. 8.13).

The proposed new compression algorithm is a modification of the one presented in section 8.4.1. The two algorithms are compared in Fig. 13.2. The improved algorithm has been built on top of an original idea from Taku Yamanaka, a KOTO collaborator and former professor in the KOTO group at Osaka University. His idea suggested a waveform transformation based on substituting each waveform sample with the difference between it and the previous one. This transformation works best for waveforms with high and smooth peaks, as the one shown in Fig. 13.2.

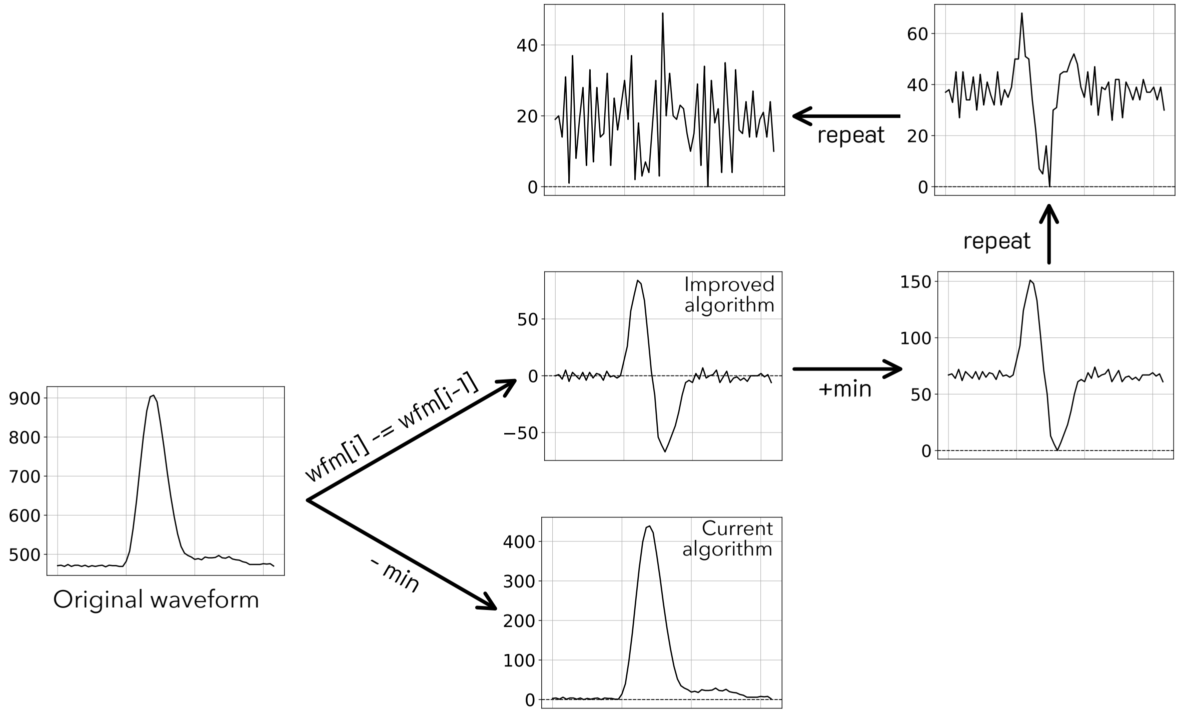


Figure 13.2: At the left, the original waveform. At the bottom, the minimum is subtracted from all waveform samples before compression, yielding to a waveform whose highest sample is around 450 counts. At the top, the transformations suggested by the proposed compression algorithm, giving a waveform whose highest sample is 50 counts.

The amplitude of the waveform transformed by the current algorithm in Fig. 13.2 is around 450 counts, hence this waveform could technically be compressed in $\log_2(450) = 9$-bit samples, 64 of them. However, the current algorithm only supports up to 8-bit compressed samples, and would not compress this waveform.

The new idea starts by subtracting from each sample the previous one, setting the first one to zero as shown in Fig. 13.2. This is motivated by the fact that the difference between each sample and the previous one is in general much smaller than the peak height for waveforms with a hit. The result of this transformation is shown in the middle column, middle row in Fig. 13.2. To avoid negative values, the waveform is shifted upwards leading to the result on the right. The transformed waveform now has an amplitude of 150 counts, thus it can be compressed in 64 8-bit samples. However, the waveform peak is still visibly higher than the noise level. We can therefore apply the same transformation again, leading to the result in the top right. This new waveform can be compressed in 7-bit samples. Still, performing the transformation a third time leads to the final waveform, at the middle top of the figure, that can be compressed in just 64 6-bit samples.

The transformed waveform is compressed in the same way the original algorithm does. To decompress the waveform offline, we need to know both the first sample of the original waveform and the number of iterations that the algorithm performed, three in this case. This information be stored in the current waveform headers without modifying their structure. In

Fig. 8.12, we noted the presence of four unused bits at the beginning of each waveform header. Two of these bits would be used by the new algorithm. Their value would indicate the number of iterations performed, as shown in the table 13.1.

Table 13.1: Bit pattern set by the purposed compression algorithm into each waveform header.

Bit pattern	Meaning
00	No iteration was performed, and the waveform was compressed following the original algorithm.
01	One iteration was performed
10	Two iterations were performed
11	Three iterations were performed

Only in rare occasions, a waveform will be large enough to make more than three iterations worth. The number of iterations can be limited to three. Note also that there is an optimal amount of iterations for each waveform. Performing an extra iteration after the optimal amount will produce a waveform whose amplitude becomes larger and not smaller. The optimal value is reached when the following relation satisfies for any waveform sample `i`:

$$|(\mathrm{wfm}[\mathtt{i}] - \mathrm{wfm}[\mathtt{i-1}]) + (\mathrm{wfm}[\mathtt{i}] - \mathrm{wfm}[\mathtt{i+1}])| \;\leq\; \max(\mathrm{wfm}) - \min(\mathrm{wfm})\;, \tag{13.1}$$

which in practice is equivalent to performing the extra iteration, and checking whether the number of bits needed to store the transformed waveform is less or not than the number of bits needed to store the original waveform.

The compression introduced by the current HLT algorithm has been shown for multiple physics triggers in Fig. 8.22, utilizing data from a physics run in 2024. With the same data sample, the compression introduced by the proposed algorithm is compared to the current one in Fig. 13.3.

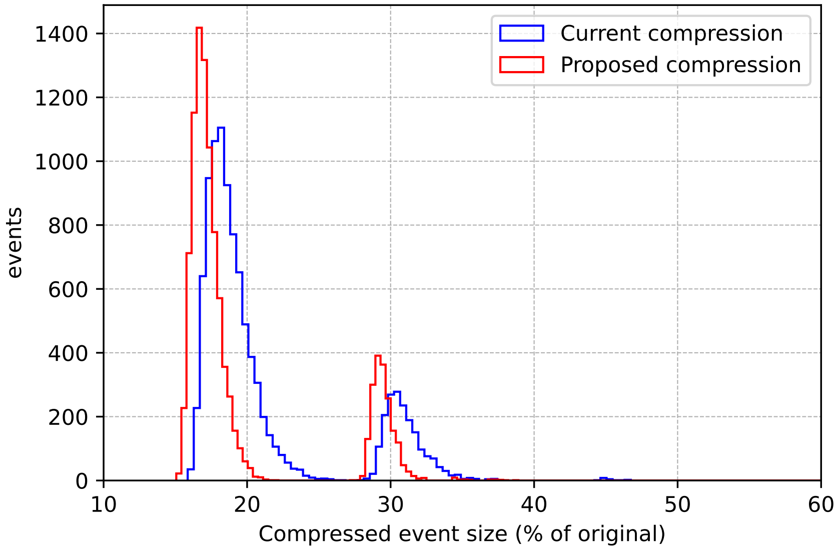


Figure 13.3: Comparison between the current compression algorithm and the proposed one.

On average, the proposed algorithm produces events whose size is 8% smaller than the current one. The first consequence of this improvement would be an increase in the maximum throughput (around 8%) of the Disk Node in events per spill. The second is an equivalent reduction of the storage needed at KEK to keep all the collected data. For reference, from the total of 500 TB collected in 2024, 40 TB could be saved by using the proposed algorithm.

A version of this algorithm has been already tested on GPUs, and is subject to be included in KOTO's HLT in the future. Since all extra operations involved can be performed in shared memory, the impact on the current HLT throughput is expected to be negligible.

13.2.2 Data integrity checks across the whole DAQ system

Around 0.01% of the collected events in 2024 have been found to contain corrupted ADC headers. The source of this corruption has not been found at the moment of writing this thesis. It could happen at any stage of the DAQ, including the ADCs and the HLT.

In future runs, checksums are planned to be computed at the ADCs and added into each ADC footer, which will allow checking the integrity of the ADC data at the OFC-Is. The OFC-Is and the OFC-IIs will also add their own checksums to the event headers, which will be verified at the HLT. The HLT will further add its own checksums to the binary files too. This strategy will allow us to identify corruption in real-time in future runs, and to constrain the source of the corruption to a specific DAQ module or stage.

Chapter 14

Conclusion

In the past three years, the data acquisition system of the J-PARC KOTO experiment has undergone a major upgrade. In particular, its PC farm has been replaced with a new GPU-based High Level Trigger (HLT), developed and commissioned by the author of this thesis. The HLT has been made able to capture data with a packet loss below 0.0002% up to 40 Gbps, satisfying the 0.1% requirement. The event loss due to packet loss in the OFC-II – HLT network during the 2024 beam-time was measured to be 0.3%.

Physics reconstruction and selection were performed on GPU. The selection efficiency of 5-cluster $K_L \to 3\pi^0$ was $\left(99.83^{+0.10}_{-0.18}\right)\%$. The efficiency of the $K^+ \to \pi^+\pi^0$ selection was $\left(98.95^{+0.21}_{-0.24}\right)\%$. The efficiency of the $K_L \to \pi^0 e^+ e^-$ selection was $\left(99.87^{+0.08}_{-0.14}\right)\%$. The event selection rejected on average 20% of the events from these three triggers. Pedestal suppression was also performed at the HLT with an inefficiency below 0.1%. Combining data reduction from pedestal suppression and waveform compression, the HLT output rate was maintained at 3.6 Gbps, 18% of the input, and below the 4 Gbps limit required between J-PARC and the data storage center at KEK.

In terms of performance, the HLT software was required to cope with the maximum expected rate of 50 kEvents per spill cycle, which corresponds to 3 kEvents per second per Computing Node. The actual throughput was measured at 7.8 kEvents per second per node, 2.6 times faster than the requirement, ensuring the software would not become a bottleneck and leaving enough room for future upgrades.

The new HLT was commissioned in Spring 2024, alongside the rest of the upgraded DAQ system. Multiple issues were gradually identified and solved, increasing the HLT efficiency from its original 97.4% to 99.6% at the end of the beam-time. The quality and usability of the events collected in spills with low DAQ efficiency have been confirmed.

The upgraded DAQ system enabled the collection of over nine times more 5-cluster $K_L \to 3\pi^0$ events than in the previous 2021 beam-time, despite the 2024 beam-time delivering only half of the kaon decays that were delivered in 2021. As a consequence, the statistical uncertainties of the $K_L \to 2\pi^0$ background estimations are expected to be reduced by a factor of 3.

This will be essential to improve the Upper Limit of the $K_L \to \pi^0\nu\overline{\nu}$ search in future KOTO analyses. Enough $K^+ \to \pi^+\pi^0$ decays were also collected to confirm that the new permanent magnet added to the KOTO beamline reduced the K^+ flux to the detector by a factor of 10. Finally, $K_L \to \pi^0 e^+e^-$ candidates could also be collected continuously for the first time in KOTO.

Bibliography

[1] Patrick Huet and Eric Sather. Electroweak baryogenesis and standard model cp violation. *Phys. Rev. D*, 51:379–394, Jan 1995.

[2] S. Navas, C. Amsler, T. Gutsche, C. Hanhart, J. J. Hernández-Rey, C. Lourenço, A. Masoni, Mikhasenko, et al. Review of particle physics. *Phys. Rev. D*, 110:030001, Aug 2024.

[3] Andrzej J. Buras. Standard Model predictions for rare K and B decays without new physics infection. *Eur. Phys. J. C*, 83(1):66, 2023.

[4] Andrzej J. Buras, Dario Buttazzo, Jennifer Girrbach-Noe, and Robert Knegjens. Can we reach the zeptouniverse with rare k and b s,d decays? *Journal of High Energy Physics*, 2014(11), November 2014.

[5] A. Abulencia, D. Acosta, J. Adelman, T. Affolder, T. Akimoto, M. G. Albrow, D. Ambrose, Amerio, et al. Search for $Z^{'} \to e^{+}e^{-}$ using dielectron mass and angular distribution. *Phys. Rev. Lett.*, 96:211801, May 2006.

[6] Andrzej J. Buras, Fulvia De Fazio, and Jennifer Girrbach. The anatomy of Z' and Z with flavour changing neutral currents in the flavour precision era. *Journal of High Energy Physics*, 2013(2), February 2013.

[7] Shinya Kanemura and Yushi Mura. Electroweak baryogenesis via top-charm mixing. *JHEP*, 09:153, 2023.

[8] Joseph Redeker. The search for $K_L \to \pi^0\nu\bar{\nu}$ in the koto experiment. In *International Conference on High Energy Physics (ICHEP)*, 2024.

[9] A. Alavi-Harati et al. Search for the rare decay K(L) —> pi0 e+ e-. *Phys. Rev. Lett.*, 93:021805, 2004.

[10] Hajime Nanjo and for the KOTO collaboration. KOTO II at J-PARC : toward measurement of the branching ratio of $K_L \to \pi^0\nu\bar{\nu}$. *Journal of Physics: Conference Series*, 2446(1):012037, feb 2023.

[11] J-PARC. Significant Increase in Beam Power and Electric Power Efficiency of the J-PARC Main Ring Accelerator, 2024.

[12] M. Tanabashi et al. (Particle Data Group). Review of particle physics. *Phys. Rev. D*, 98:030001, 2018.

[13] Keita Ono. Thin scintillation counter with a new readout method for the koto experiment. *Journal of Physics: Conference Series*, 2446(1):012048, feb 2023.

[14] K. Sato et al. CsI calorimeter for the J-PARC KOTO experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 982:164527, 2020.

[15] Yu-Chen Tung. Analysis techniques for neutron background suppression at koto. *Journal of Physics: Conference Series*, 2446:012053, 02 2023.

[16] Y. Tajima et al. Barrel photon detector of the kek $K_L \to \pi^0 \nu \overline{\nu}$ experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 592(3):261–272, 2008.

[17] Naoki Kawasaki, Noboru Sasao, Tadashi Nomura, Hajime Nanjo, Daichi Naito, Yosuke Maeda, Shigeto Seki, Ichinori Kamiji, and Kouta Nakagiri. *Halo Neutron Measurement for KOTO Experiment.*

[18] R. Murayama et al. A new cylindrical photon-veto detector for the $K_L \to \pi^0 \nu \overline{\nu}$ experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 953:163255, 2020.

[19] Y. Maeda, N. Kawasaki, T. Masuda, H. Morii, D. Naito, Y. Nakajima, H. Nanjo, T. Nomura, N. Sasao, S. Seki, K. Shiomi, T. Sumida, and Y. Tajima. An aerogel cherenkov detector for multi-gev photon detection with low sensitivity to neutrons. *Progress of Theoretical and Experimental Physics*, 2015(6):063H01, 06 2015.

[20] Kota Nakagiri, Noboru Sasao, Tadashi Nomura, Hajime Nanjo, Naoki Kawasaki, Daichi Naito, Yosuke Maeda, Shigeto Seki, and Ichinori Kamiji. *Gas Wire Chamber for In-Beam Charged Particle Detector in KOTO Experiment.*

[21] D. Naito, Y. Maeda, N. Kawasaki, T. Masuda, H. Nanjo, T. Nomura, M. Sasaki, N. Sasao, S. Seki, K. Shiomi, and Y. Tajima. Development of a low-mass and high-efficiency charged-particle detector. *Progress of Theoretical and Experimental Physics*, 2016(2):023C01, 02 2016.

[22] Melissa A. Hutcheson and on behalf of the KOTO collaboration. Normalization studies on the 2016-2018 data for the koto experiment. *Journal of Physics: Conference Series*, 1526(1):012035, apr 2020.

[23] C. Lin. *Study of $K_L \to \pi^0 \nu \overline{\nu}$ and $K_L \to \pi^0 \gamma\gamma$ with the Cluster-Finding Trigger at KOTO*. PhD thesis, National Taiwan University, Jan 2021.

[24] Mircea Bogdan, Jiasen Ma, Harold Sanders, and Y. Wah. Custom 14-bit, 125mhz adc/-data processing module for the kl experiment at j-parc. In *2007 IEEE Nuclear Science Symposium Conference Record*, volume 1, pages 133–134, 2007.

[25] Mircea Bogdan, Jean-Francois Genat, and Yau Wah. Custom 12-bit, 500mhz adc/data processing module for the koto experiment at j-parc. In *2010 17th IEEE-NPSS Real Time Conference*, pages 1–2, 2010.

[26] Lee Jong-won. *Energy Calibration Method for the KOTO CsI Calorimeter*. PhD thesis, Osaka University, 2014.

[27] IEEE. Ieee standard for ethernet. IEEE Std 802.3-2018, 2018. Accessed: 2020-11-20.

[28] J. et al. (KOTO Collaboration) Comfort. Proposal for $k_l \rightarrow \pi^0 \nu\overline{\nu}$ experiment at j-parc, 2006.

[29] Mario Gonzalez. GPU offloading in the High Level Trigger of the KOTO experiment. In *Measurement System Workshop*, 2021.

[30] Luigi Rizzo. netmap: A novel framework for fast packet I/O. In *2012 USENIX Annual Technical Conference (USENIX ATC 12)*, pages 101–112, Boston, MA, June 2012. USENIX Association.

[31] The KOTO Collaboration. Study of the $K_L \rightarrow \pi^0 \nu\overline{\nu}$ decay at the j-parc koto experiment. *Phys. Rev. Lett.*, 126:121801, Mar 2021.

[32] Marco Rovere, Ziheng Chen, Antonio Di Pilato, Felice Pantaleo, and Chris Seez. Clue: A fast parallel clustering algorithm for high granularity calorimeters in high-energy physics. *Frontiers in Big Data*, 3, 2020.

[33] The CMS Collaboration. The Phase-2 Upgrade of the CMS Endcap Calorimeter. 2017.

[34] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 4.1*, November 2023.

[35] Pavel Shamis, Manjunath Gorentla Venkata, M Graham Lopez, Matthew B Baker, Oscar Hernandez, Yossi Itigin, Mike Dubman, Gilad Shainer, Richard L Graham, Liran Liss, et al. UCX: an open source framework for HPC network APIs and beyond. In *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, pages 40–43. IEEE, 2015.

[36] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[37] N. D. Gagunashvili. Comparison of weighted and unweighted histograms, 2006.

[38] S I Bityukov and N V Krasnikov. Confidence intervals for the parameter of Poisson distribution in presence of background. Technical report, CERN, Geneva, 2000.

[39] Tadashi Nomura. KOTO results and prospects. In *Kaons@J-PARC 2024 workshop*, 2024.

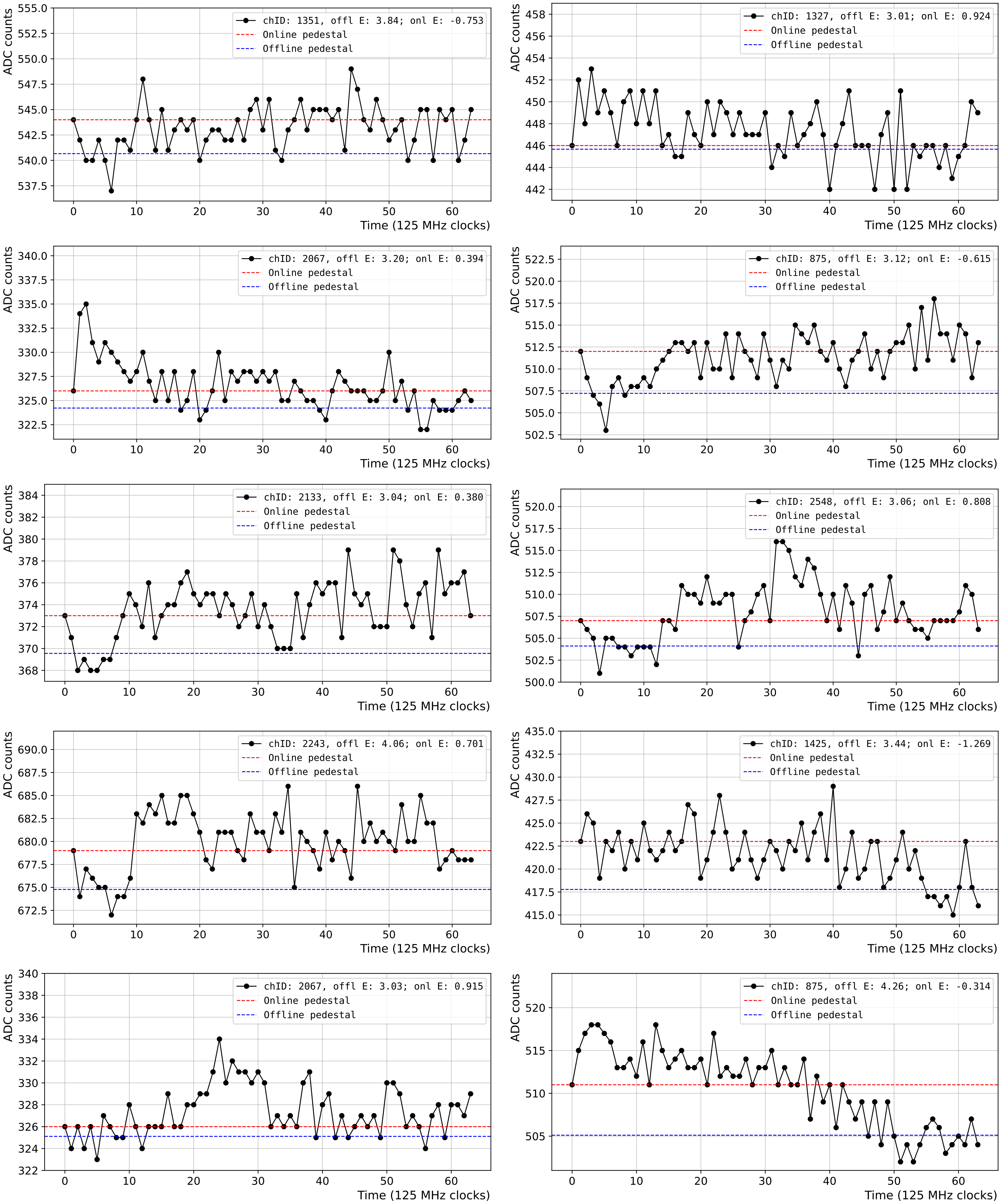
[40] F. Ambrosino et al. Measurements of the absolute branching ratios for the dominant K(L) decays, the K(L) lifetime, and V(us) with the KLOE detector. *Phys. Lett. B*, 632:43–50, 2006.

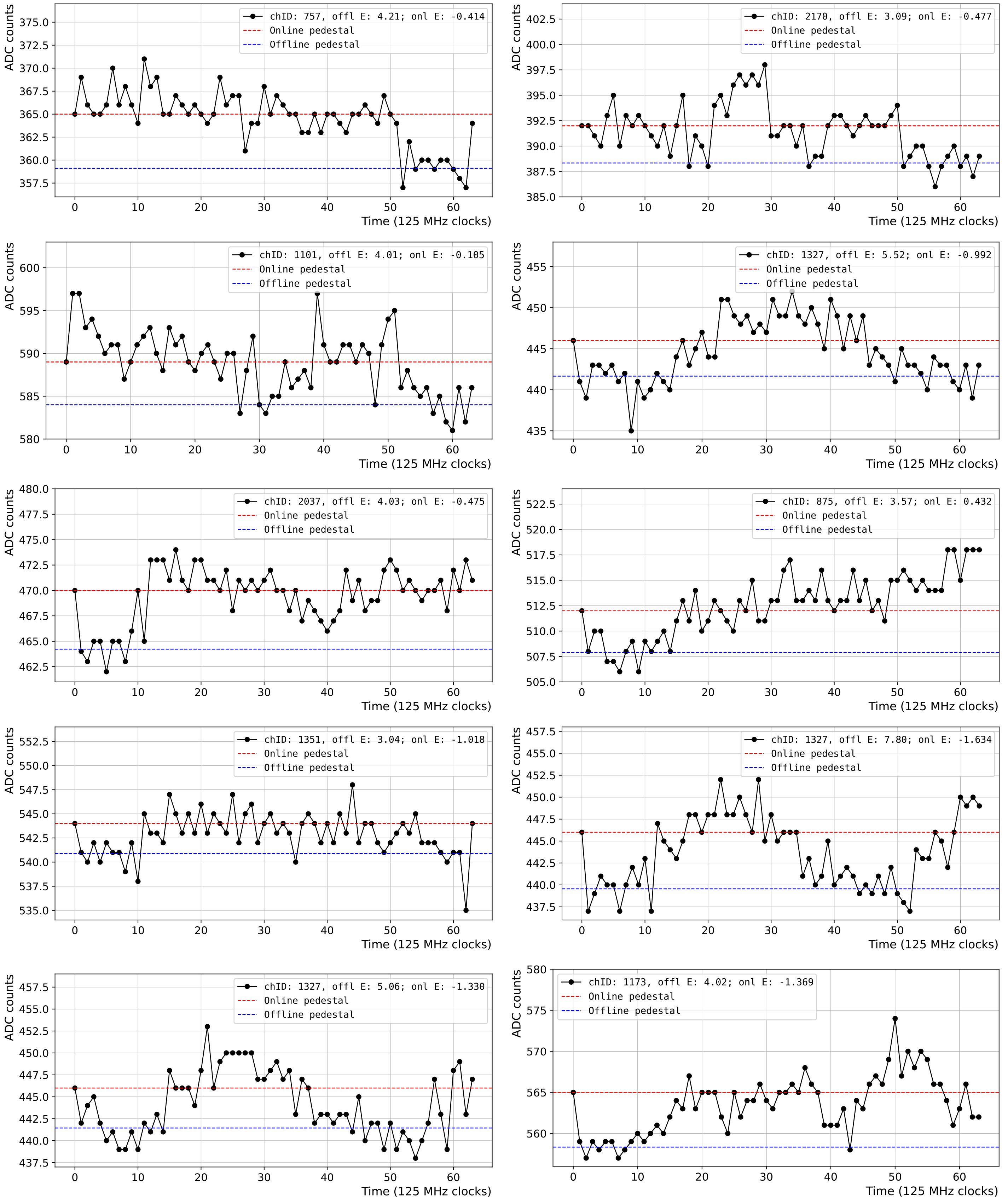
Appendices

Appendix A

CsI Waveforms Contributing to the Pedestal Suppression Inefficiency

The Inefficiency measurements of the online pedestal suppression have been discussed in section 8.3. In this appendix, a randomly selected sample of actual waveforms contributing to the inefficiency are shown. The legend in each plot indicates the CsI channel ID, and the energy measured offline and online. All waveforms are within the online pedestal suppression window ($-2MeV < E < 1MeV$ and peak heigh $<$ 10 counts), and their offline calculated Energy is larger than 3 MeV. In most cases, the offline pedestal is miscalculated, and lower than the actual pedestal value. This can be due to a large pulse recorded just before the beginning of the waveform, which can lead to a temporary decrease in ADC counts just after the large pulse. Other waveforms have just large noise fluctuations. No obvious pulse could be identified in a large sample of *inefficient* waveforms. The already small inefficiency of the online pedestal suppression can therefore be neglected in the offline analysis.


chID: 1351, offl E: 3.84; onl E: -0.753
Online pedestal
Offline pedestal
chID: 1327, offl E: 3.01; onl E: 0.924
Online pedestal
Offline pedestal
chID: 2067, offl E: 3.20; onl E: 0.394
Online pedestal
Offline pedestal
chID: 875, offl E: 3.12; onl E: -0.615
Online pedestal
Offline pedestal
chID: 2133, offl E: 3.04; onl E: 0.380
Online pedestal
Offline pedestal
chID: 2548, offl E: 3.06; onl E: 0.808
Online pedestal
Offline pedestal
chID: 2243, offl E: 4.06; onl E: 0.701
Online pedestal
Offline pedestal
chID: 1425, offl E: 3.44; onl E: -1.269
Online pedestal
Offline pedestal
chID: 2067, offl E: 3.03; onl E: 0.915
Online pedestal
Offline pedestal
chID: 875, offl E: 4.26; onl E: -0.314
Online pedestal
Offline pedestal
ADC counts
Time (125 MHz clocks)
0
10
20
30
40
50
60


chID: 757, offl E: 4.21; onl E: -0.414
chID: 2170, offl E: 3.09; onl E: -0.477
chID: 1101, offl E: 4.01; onl E: -0.105
chID: 1327, offl E: 5.52; onl E: -0.992
chID: 2037, offl E: 4.03; onl E: -0.475
chID: 875, offl E: 3.57; onl E: 0.432
chID: 1351, offl E: 3.04; onl E: -1.018
chID: 1327, offl E: 7.80; onl E: -1.634
chID: 1327, offl E: 5.06; onl E: -1.330
chID: 1173, offl E: 4.02; onl E: -1.369
Online pedestal
Offline pedestal
ADC counts
Time (125 MHz clocks)
0
10
20
30
40
50
60

Appendix B

Events contributing to the HLT selection inefficiency

In section 10.3, the $\mathcal{E}$ and $\mathcal{R}$ of the MinXY, MaxR, and TotalE cuts for the 5γ, K^+, and $K_L \to \pi^0 e^+ e^-$ triggers were estimated. The reasons why events contribute to the inefficiency of the HLT selection were given at the end of the section. Representative event displays for each of these cases are shown in this appendix. The potential impact of these events on the offline analysis is also discussed.

Single cluster split into two at the HLT

The event display in Fig. B.1 shows a $K^+ \to \pi^+\pi^0$ event with a long cluster, perhaps originating from a charged pion hit producing a muon in the calorimeter. The COE of that cluster as reconstructed offline, satisfies the offline quality criteria MinXY > 150 mm, and the event is accepted offline.

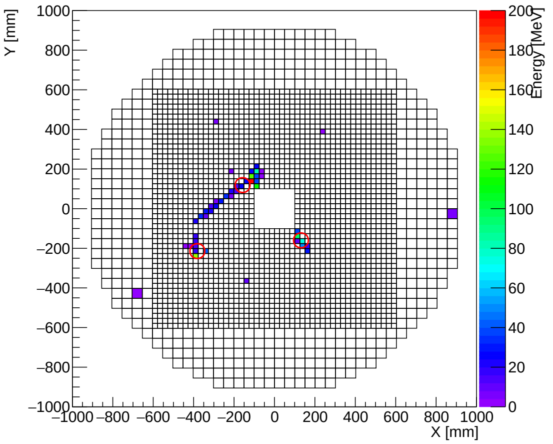


Figure B.1: The circles represent the position of the offline-reconstructed clusters. Online, the long cluster was split into two.

The HLT looks for local maximums to identify clusters. The long cluster in Fig. B.1 has two local maximums, one where the highest energy crystal is, and another one less obvious on the cluster tail. Two clusters are therefore identified, and the event is rejected because the cluster closest to the beam-hole falls below the HLT's MinXY < 140 threshold.

Events containing spread clusters are rejected if the spread clusters come from photon hits. However, to increase statistics in the $K^+ \to \pi^+\pi^0$ analysis, events with spread clusters are kept if the spread cluster comes from a charged pion. This is checked by matching energy depositions in the Charged Veto in front of the Calorimeter with the spread cluster coordinates.

The event in Fig. B.1, which is rejected by the HLT, is kept offline and therefore contributes to the HLT inefficiency. However, it does not pose a concern for the offline analysis, since the production of spread-out clusters by charged pions depends only on the interaction of the pion with the calorimeter, and is independent of the physics of the $K^+ \to \pi^+\pi^0$ decay.

Two clusters merged offline

The event in Fig. B.2 shows an event from the 5γ trigger, where the two clusters closest to the beam-hole were merged by the offline algorithm. The L2 clustering algorithm also merged the two clusters, and thus the event was included in the 5γ trigger. The HLT algorithm did separate the two clusters, and the cluster closest to the beam-hole caused the event to be rejected.

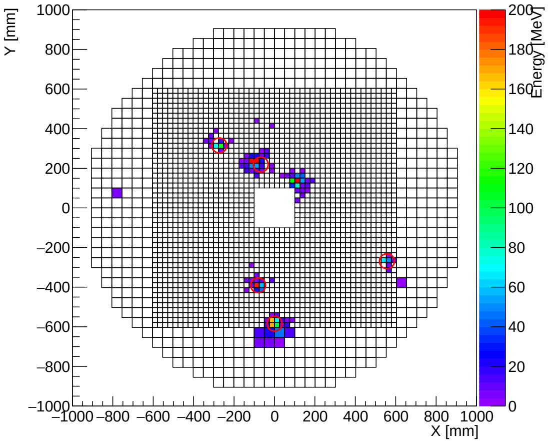


Figure B.2: A six-cluster event where two clusters were merged offline. The circles represent the position of the offline-reconstructed clusters.

These events are rejected offline by cluster-shape cuts that are not included in the definition of the HLT inefficiency. Rejecting these events online is therefore not a concern for the offline analysis.

Two clusters merged online

The event in Fig. B.2 shows an event from the 5γ trigger. The offline algorithm (red circles), has reconstructed all clusters in the correct positions. The HLT algorithm, however, merged the two clusters at the bottom left of the beam-hole. The merged cluster, whose COE falls between the two, triggers the event to be rejected by the MinXY cut.

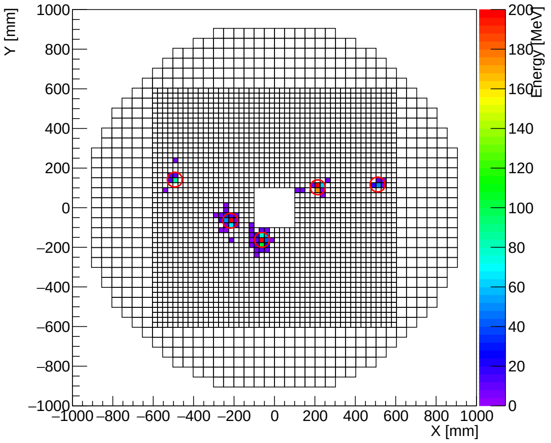


Figure B.3: The circles represent the position of the offline-reconstructed clusters. Online, the two clusters at the bottom left of the beam-hole were merged.

Events that are rejected online because two clusters were merged at the HLT contribute to the HLT inefficiency.

Single-hit clusters ignored offline

The event in Fig. B.4 shows a $K^+ \to \pi^+\pi^0$ event with two hits near the bottom edge of the calorimeter. Offline, these two hits are not close enough to form a cluster together, and are therefore ignored. Online, the two hits form a cluster that triggers the event to be rejected by the MaxR cut.

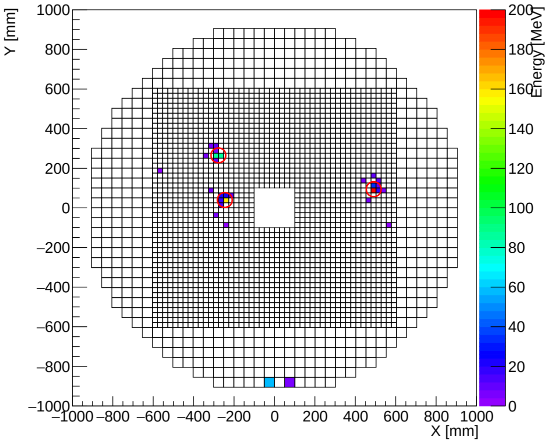


Figure B.4: The circles represent the position of the offline-reconstructed clusters. Online, an extra cluster is identified near the bottom calorimeter edge.

This phenomenon is the main reason for large discrepancies in the MaxR reconstructed online and offline. Part of the total energy of these events is likely to have been deposited outside the calorimeter, and in most cases a correspondent energy deposition should be found in the Main Barrel photon veto detector surrounding the calorimeter. The Center of Energy in the event in Fig. B.4 is shifted towards the opposite direction of the two energy depositions at the bottom of the calorimeter, indicating that the energy depositions at the bottom of the calorimeter come from an actual particle hit, and this event is therefore not a $K^+ \to \pi^+\pi^0$ decay, even though it was tagged as such. Rejecting it online is thus not a concern for the offline analysis.

Accidental events

In some cases, accidental hits can be recorded recorded within the same 512 ns window of the L1-triggered event. If the standard deviation of the timing of the reconstructed clusters in the accidental event is smaller than the one of the triggered event, the offline reconstruction discards the triggered event and keeps the accidental one.

Figure B.5 shows an example of this from the 5γ trigger. Fig. B.5 shows another example from the K^+ trigger.

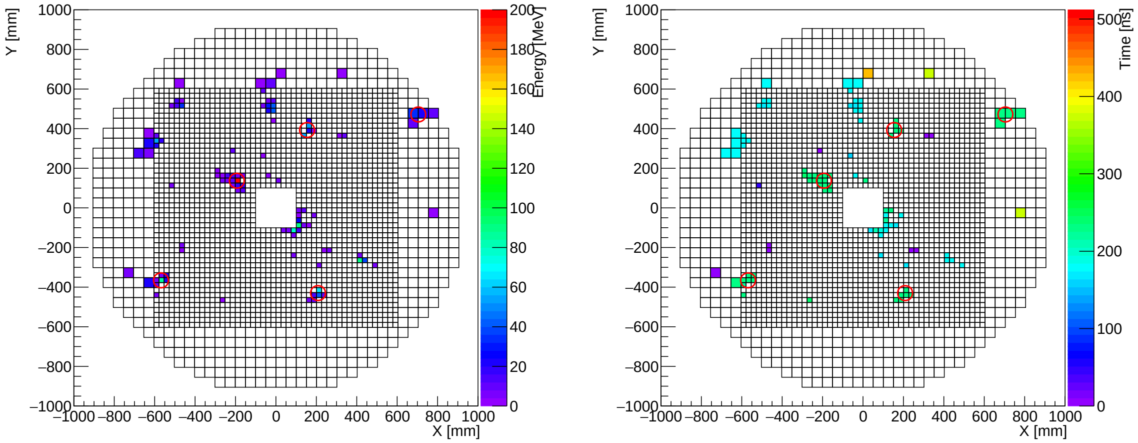


Figure B.5: An event from the 5γ trigger. Left: Energy display. Red circles indicate the position of the five clusters selected offline. Right: Timing display. The L1-triggered hits are shown in blue. The hits reconstructed offline are shown in green. The blue event is rejected at the HLT for failing the MinXY cut.

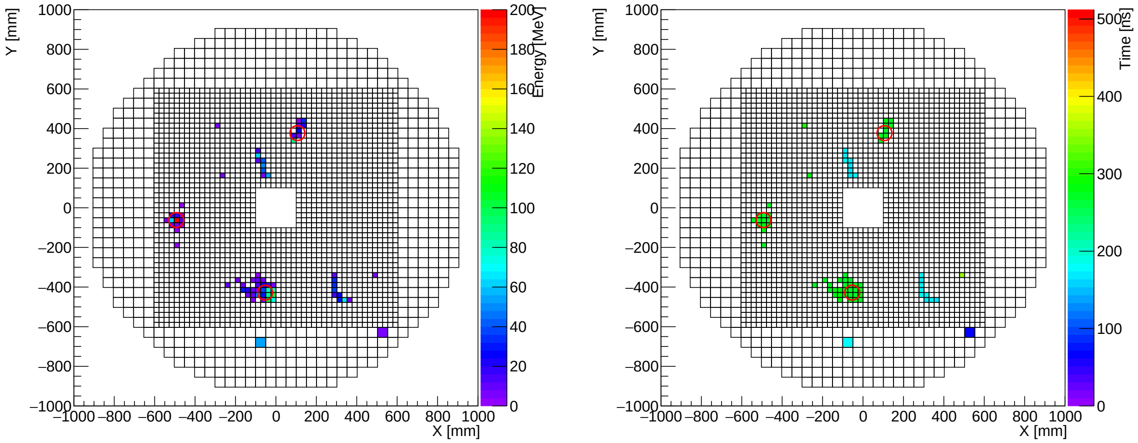


Figure B.6: The circles represent the position of the offline-reconstructed clusters. These clusters (green in the right plot), were not considered at the HLT, for being outside the L1 trigger window. The event reconstructed at the HLT (blue hits) falls below the TotalE cut threshold and is therefore rejected online.

Two events being captured in the same 512 ns window is a phenomenon that occurs randomly, and independently from the physics of each event. For this reason, rejecting these events at the HLT does not pose a concern for the offline analysis.

Appendix C

Issues Affecting the HLT Efficiency

C.1 Corrupted data in the HLT

During a total of 13 runs, large portions of the data were corrupted due to an issue when writing the event size into the event headers on the GPUs. This issue lowered the efficiency of these 13 runs to around 80%. In most cases, the run was restarted promptly after. The total loss due to this issue is estimated to be less than 0.1%. This issue was solved after the first half of the beam-time. The average number of events per spill where corrupted data was found during the offline analysis are reported in Fig. C.1. The figure shows all physics runs taken during the first half of the beam-time.

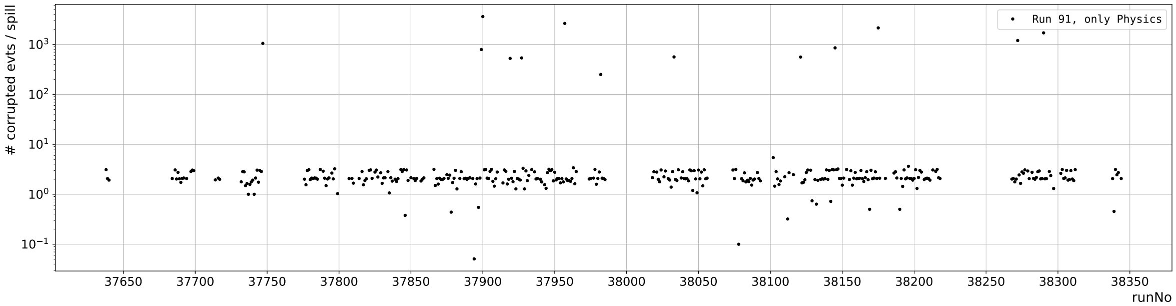


Figure C.1: Average number of events per spill where corrupted data was found during the offline analysis. Only in 13 runs the number of corrupted events was significant.

One particular ADC is known to have a high bit-flip rate in part of its header. This ADC is the reason why between 1 and 2 events per spill show signs of corruption on average. All runs where the number of corrupted events per spill is significantly high (in the order of 10^3) are due to actual corruption happening at the HLT.

C.2 Low efficiency during the last stage of the beam-time

The 2024 beam-time was divided into four stages as delimited in Fig. 12.5. During the last stage, the HLT efficiency was 99.6%. However, there are still some runs in this period whose efficiency was significantly lower. The efficiency of all runs belonging to the last stage of the 2024 beam-time is shown in Fig. C.2.

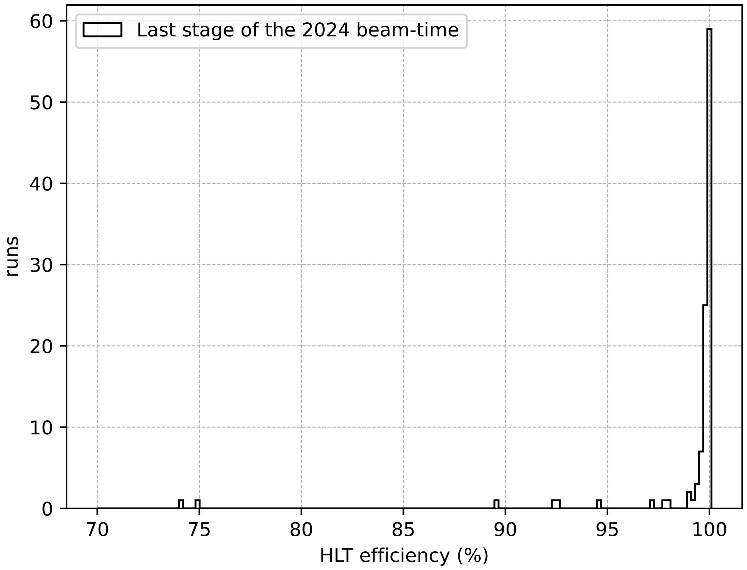


Figure C.2: HLT efficiency of all physics runs from the last stage of the 2024 beam-time.

In this section, we will attempt to understand the reason for the low efficiency of the six runs with efficiency below 95% in Fig. C.2. These runs are marked with black boxes in Fig. C.3.

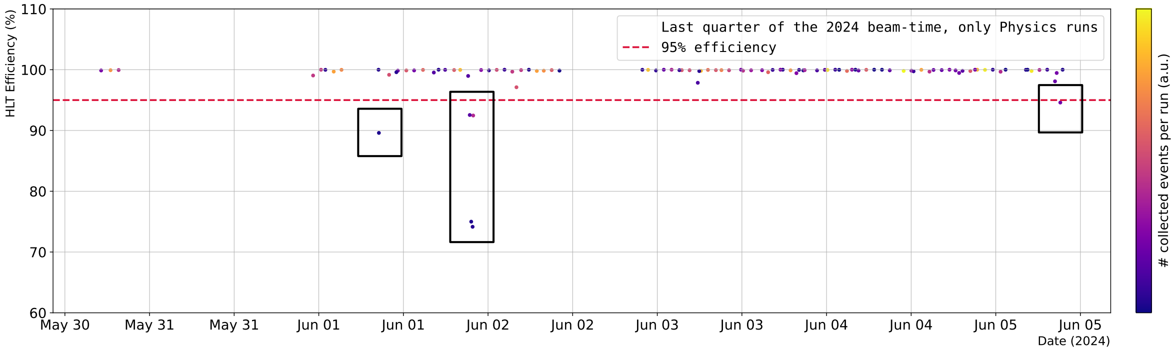


Figure C.3: HLT efficiency of all physics runs from the last stage of the 2024 beam-time.

The six runs highlighted in Fig. C.3 can be classified into two categories.

First, in all four runs in the middle square, the same Computing Node registered a high packet that led to the loss of a large number of events. The event loss in this particular node increased close to 100% in two of these runs, making the total HLT efficiency drop to around 75%. The other three Computing Nodes were not affected. The four runs affected by this issue are contiguous in time. In this case, restarting the DAQ system did not make a difference. The lost packets did not reach the HLT software, and thus we believe they were dropped either at the transceivers or at the Network Interface Card, likely due to a poor signal. We hope this issue will be solved after replacing the currently 150 m-rated transceivers with 300 m-rated ones at the Computing Node side.

Second, the remaining two runs marked with black boxes in Fig. C.3 with efficiency around 90% are due to a failure in the link between the Computing Nodes and the Disk Node. In these runs, during some time, the Disk Node did not receive any data from any of the Computing Nodes. No error was reported by either the HLT software or MPI (which manages the communication between all HLT nodes). Eventually, the link was automatically reestablished, and the HLT efficiency went back to normal. This issue has not been solved, reproduced, or fully understood to this day. Either a switch failure or an MPI-related issue is suspected at the moment. In the next beam-time, monitoring the switch logs will help identify the cause of this network blockage. In any case, the issue is likely triggered by high traffic in the network. To improve the situation, it has been planned to increase the size of the event blocks constructed at the Computing Nodes, contributing to sending larger, but less frequent, packets to the Disk Node.

Appendix D

Total Energy and Number of Hit Channels in Veto Detectors

Depending on the physics target, different trigger configurations might impose tighter (lower maximum energy) or looser (higher maximum energy) requirements on different veto detectors. To evaluate the presence of hits within these requirements, each ADC channel waveform is evaluated against two thresholds. The example in Fig. D.1 shows a low-energy waveform, that passes the loose requirement but not the tight requirement.

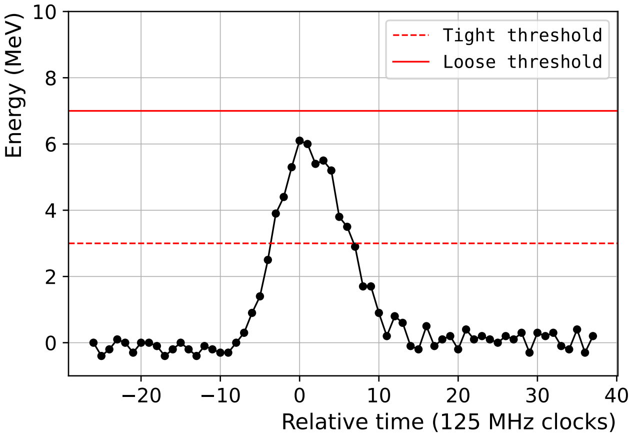


Figure D.1: An example of a waveform with a hit, that passes the loose requirement but not the tight requirement.

The presence of a hit under the tight or loose thresholds is first evaluated in each ADC channel. Each ADC then combines the results of its 16 channels as shown in Fig. D.2.

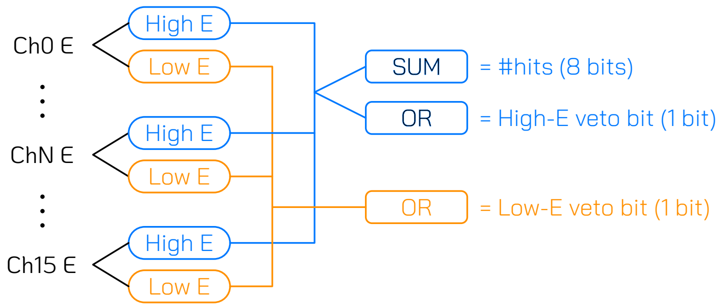


Figure D.2: Calculations performed at the ADC modules in veto crates. OR operations are used to determine if any channel recorded a hit above the high-E and/or the low-E thresholds. High-E hits are added up to compute the total number of hits.

The results are then sent to the local CDT through four ADC chains, as was explained for ADCs on the CsI crates. The local CDT computes the presence of hits under the tight and loose veto requirements. It also reserves two bits that determine where the received number of hits falls within two configured thresholds, as illustrated in table D.1.

Table D.1: Contents of the two bits prepared at the local CDT. In this example, the low bit is 1 if the number of hits is ≥ 2, and the high bit is 1 if the number of hits is ≥ 2.

Number of hits	high threshold bit	low threshold bit
0	0	0
1	0	1
≥ 2	1	1

For instance, a trigger targeting $K_L \to \pi^0 e^+ e^-$ events could require the Charged Veto detector in front of the CsI calorimeter to record at least 2 hits. A different trigger targeting $K^+ \to \pi^+ \pi^0$ events could require the same detector to record 1 hit, and the main $K_L \to \pi^0 \nu \overline{\nu}$ physics trigger would require 0 hits on the CV. Utilizing two threshold bits as defined in table D.1 allows for these three triggers to run simultaneously.

Note that for this approach to work, all outputs of a single veto detector have to converge in the same local CDT. An entire veto detector must therefore be read out by a single ADC crate.

Results from each local CDT are combined in the ET/Veto OFC module, and sent to the Top CDT.

Appendix E

Single Event Sensitivity Calculation

The single event sensitivity (SES) gives a measure of what the $K_L \to \pi^0\nu\overline{\nu}$ branching ratio would be if one signal event was observed within a collected data set of kaon decays. It is defined as shown in Eq. E.1, as the ratio of the $K_L \to \pi^0\nu\overline{\nu}$ branching ratio to the expected number of recorded $K_L \to \pi^0\nu\overline{\nu}$ decays in KOTO.

If the true $K_L \to \pi^0\nu\overline{\nu}$ branching ratio was equal to its SM prediction, we would expect to observe one signal event once the KOTO SES reaches the SM predicted $K_L \to \pi^0\nu\overline{\nu}$ BR. Until we reach that point, KOTO's SES on the $K_L \to \pi^0\nu\overline{\nu}$ decay gets lowered as more data is collected and analyzed.

The single event sensitivity is formally defined as shown in Eq. E.1.

$$\mathrm{SES} = \frac{1}{\mu}\mathcal{B}(K_L \to \pi^0\nu\overline{\nu}) \tag{E.1}$$

where $\mathcal{B}(K_L \to \pi^0\nu\overline{\nu})$ is the true $K_L \to \pi^0\nu\overline{\nu}$ branching ratio (unknown), and μ is the expected number of recorded signal events within the available data set. The quantity μ can be computed as shown in Eq. E.2.

$$\mu = A_{K_L\to\pi^0\nu\overline{\nu}}\; N_{K_L}\; \mathcal{B}(K_L \to \pi^0\nu\overline{\nu}) \tag{E.2}$$

where,

- $A_{K_L\to\pi^0\nu\overline{\nu}}$, the acceptance of the $K_L \to \pi^0\nu\overline{\nu}$ signal, accounts for the fact that not all K_L's in the beam decay in the KOTO detector, and even a recorded actual $K_L \to \pi^0\nu\overline{\nu}$ decay might be rejected during the $K_L \to \pi^0\nu\overline{\nu}$ event selection. The signal acceptance is defined as the product of the K_L decay probability inside the KOTO detector and the probability of a $K_L \to \pi^0\nu\overline{\nu}$ passing its own event selection.

- N_{K_L} is the number of K_L's entering the KOTO detector.

The signal acceptance is evaluated with Monte Carlo simulations. The K_L yield N_{K_L} is calculated with $K_L \to 2\pi^0$ events, whose real branching ratio is precisely known [40], as shown in Eq. E.3:

$$N_{K_L} = \frac{N_{K_L \to 2\pi^0}}{A_{K_L \to 2\pi^0}} \cdot \frac{1}{\mathcal{B}(K_L \to 2\pi^0)} \tag{E.3}$$

The first term in Eq. E.3 gives the amount of $K_L \to 2\pi^0$ decays that are expected to have occurred in the KOTO detector. This is, the ones that were actually reconstructed after the event selection $N_{K_L \to 2\pi^0}$, scaled by the $K_L \to 2\pi^0$ acceptance $A_{K_L \to 2\pi^0}$. The total "number of K_L decays" N_{K_L} is obtained by dividing the "number of $K_L \to 2\pi^0$ decays" by the $K_L \to 2\pi^0$ branching ratio.

We can eliminate the unknown $\mathcal{B}(K_L \to \pi^0 \nu\overline{\nu})$ in Eq. E.1 by combining it with eq E.3.

$$\text{SES} = \frac{1}{\mu}\mathcal{B}(K_L \to \pi^0 \nu\overline{\nu}) = \frac{1}{A_{K_L \to \pi^0 \nu\overline{\nu}} N_{K_L}} \tag{E.4}$$

Then, we can use Eq. E.3 to express N_{K_L} in terms of measurable quantities, giving the SES expression in Eq. E.5.

$$\text{SES} = \frac{1}{A_{K_L \to \pi^0 \nu\overline{\nu}}} \frac{A_{K_L \to 2\pi^0}\mathcal{B}(K_L \to 2\pi^0)}{N_{K_L \to 2\pi^0}} \tag{E.5}$$

where the $K_L \to \pi^0 \nu\overline{\nu}$ and $K_L \to 2\pi^0$ acceptances are estimated with Monte Carlo, the $K_L \to 2\pi^0$ branching ratio is known, and $N_{K_L \to 2\pi^0}$ is measured.

Appendix F

Effect of a Suboptimal Memory Allocation and CPU Affinity in the 40 Gbps Packet Capture

To demonstrate the importance of correct memory allocation, we repeated the test shown in Fig. 11.1 with only a 8 GB RAM module installed in the correct NUMA node. This forces the HLT software to allocate part of its PCAP buffers in the other NUMA node, which the PCAP CPUs have slower access to. The results are shown in Fig. F.1.

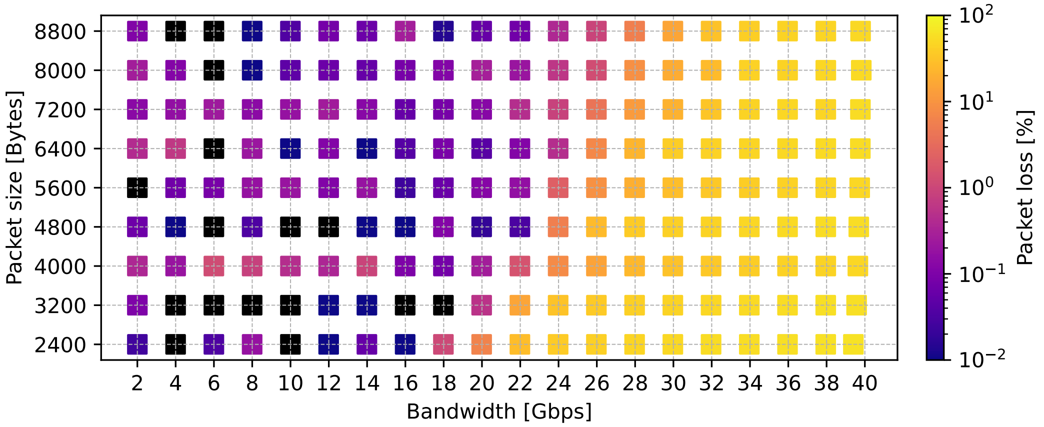


Figure F.1: Packet loss as a function of packet size and data rate, when forcing the allocation of part of the HLT raw data buffers outside the optimal NUMA node.

The total amount of memory installed in the HLT node does not change with respect to the test results shown in section 11.1. The HLT software is also identical. The much lower performance observed in Fig. F.1 is entirely due to a suboptimal memory allocation. The effect of not setting the CPU affinity is shown in Fig. F.2. It is more subtle, but still significant.

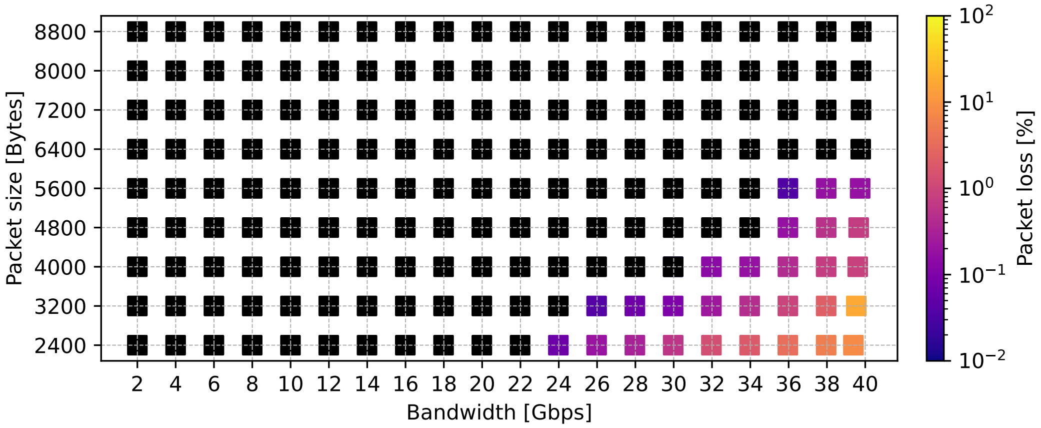


Figure F.2: Packet loss as a function of packet size and data rate, when correctly allocating memory but setting the CPU affinity to the wrong NUMA node.

Appendix G

Comparison of Clustering Algorithms in KOTO

In this appendix, the following three clustering algorithms used in KOTO are tested against simulation data:

- The L2 algorithm. This is the algorithm that runs on KOTO's L2 FPGAs. It was described in section 4.6.3. Its only output is the number of clusters in the CsI calorimeter.
- The HLT algorithm, described in section 7.5.2, running on GPUs in KOTO's HLT.
- The offline algorithm, running offline on CPUs.

The tests consist in feeding these three algorithms with the same events, and comparing the number of clusters obtained by each of them to the true number of CsI hits, which is known in Monte Carlo data.

The HLT algorithm, which offers the highest degree of flexibility, has been optimized in each case to maximize its efficiency against Monte Carlo. The definition of the parameters involved in the HLT algorithm (the CLUE algorithm originally developed by CMS) was given in table 7.1. Their optimal values found for events from the $K_L \to \pi^0 e^+ e^-$, $K_L \to 3\pi^0$, and $K_L \to \pi^+ \pi^- \pi^0$ triggers, when optimizing the clustering efficiency against the offline algorithm and against simulation data, are given in tables G.1, G.2, and G.3 respectively. In the tables, one "crystal" is equivalent to 25 mm, the size of the small CsI crystals in the KOTO calorimeter.

Table G.1: Clustering parameters used at the HLT for $K_L \to \pi^0 e^+ e^-$, and found to be best against simulation data.

Parameter	Value (best against offline algo.)	Value (best against simulation)
d_c	5 crystals	2.5 crystals
ρ_c	56 MeV	57 MeV
d_{seed}	4.5 crystals	3 crystals
d_{outl}	4.5 crystals	3 crystals

Table G.2: Clustering parameters used at the HLT for $K_L \to 3\pi^0$, and found to be best against simulation data.

Parameter	Value (best against offline algo.)	Value (best against simulation)
d_c	6 crystals	2.5 crystals
ρ_c	56 MeV	70 MeV
d_{seed}	5.5 crystals	5 crystals
d_{outl}	5.5 crystals	5 crystals

Table G.3: Clustering parameters used at the HLT for $K_L \to \pi^+ \pi^- \pi^0$, and found to be best against simulation data.

Parameter	Value (best against offline algo.)	Value (best against simulation)
d_c	7 crystals	4.5 crystals
ρ_c	56 MeV	65 MeV
d_{seed}	6.5 crystals	8.5 crystals
d_{outl}	6.5 crystals	8.5 crystals

G.1 $K_L \to 3\pi^0$ events

The $K_L \to 3\pi^0$ events produce six photons. Some or all of them will hit the calorimeter. Photon clusters, as was shown in Fig. 7.9, are typically round, and their energy is very concentrated in the center of the cluster. The performance of the three clustering algorithms against this data is shown in Fig. G.1.

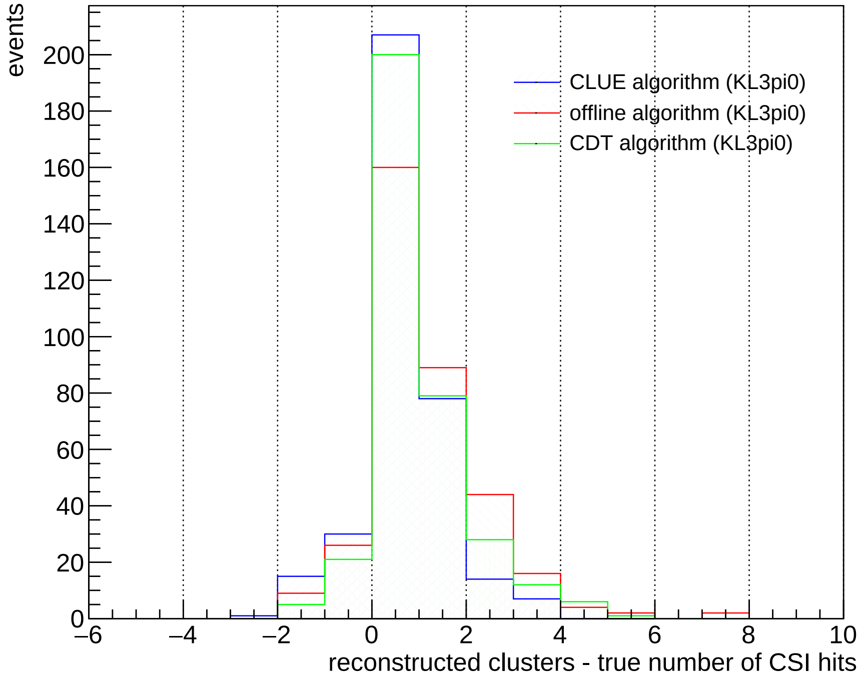


Figure G.1: Comparison of the L2 (CDT, green), the HLT (CLUE, blue) and the offline clustering algorithms against $K_L \to 3\pi^0$ events.

The L2 algorithm (green in Fig. G.1), by imposing a high energy (20 MeV) threshold to the energy of each CsI crystal, sees only the most energetic crystals in the cluster, and is able to accurately separate even close-by hits. The offline algorithm (red), by imposing a low 3 MeV threshold, tends to merge close-by clusters and does not perform specially well when many clusters are expected in the calorimeter. The HLT algorithm (blue) combines the advantages of both the L2 and the offline algorithms. By considering the energy of each crystal, it is able to identify local maximums and separate close-by clusters. By imposing a 3 MeV when defining a "hit crystal", it is able to accurately measure the hit energy of each cluster.

G.2 $K_L \to \pi^0 e^+ e^-$ events

Photon and electron clusters look similar, often indistinguishable from each other. The main difference between the $K_L \to \pi^0 e^+ e^-$ and the $K_L \to 3\pi^0$ events is the number of clusters in the calorimeter, reduced from six to four. This reduces the probability of having close-by hits, which increases the relative accuracy of the offline algorithm, as shown in Fig. G.2.

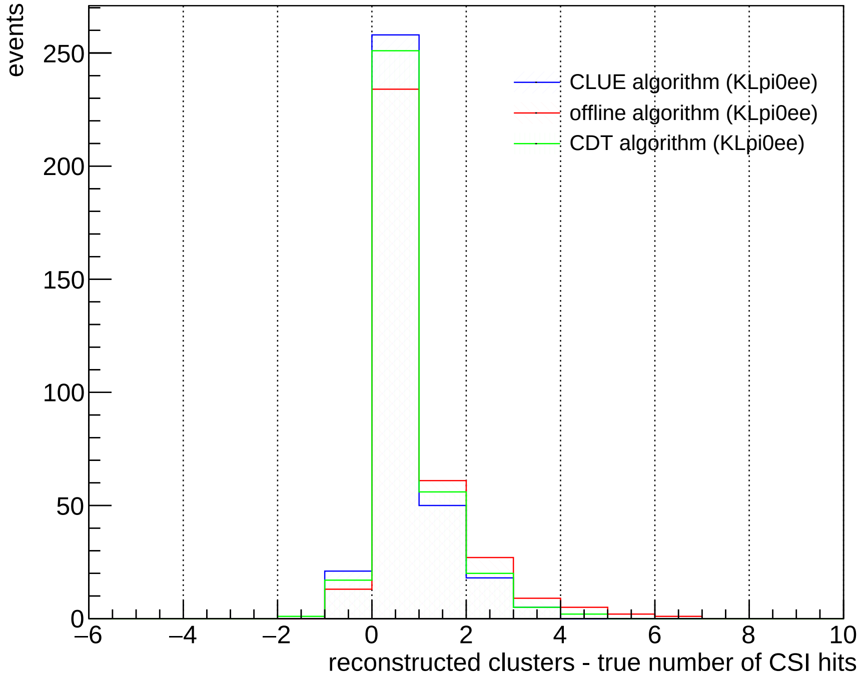


Figure G.2: Comparison of the L2 (CDT, green), the HLT (CLUE, blue), and the offline clustering algorithms against $K_L \to \pi^0 e^+ e^-$ events.

The fewer calorimeter hits in these events lead to an overall higher efficiency, particularly benefitting the offline algorithm.

G.3 $K_L \to \pi^+\pi^-\pi^0$ events

Pion clusters present different shapes and energy distributions than photon or electron clusters. The L2 and the offline algorithms, designed to perform well specifically for photon clusters, are expected to have difficulties when reconstructing pion clusters.

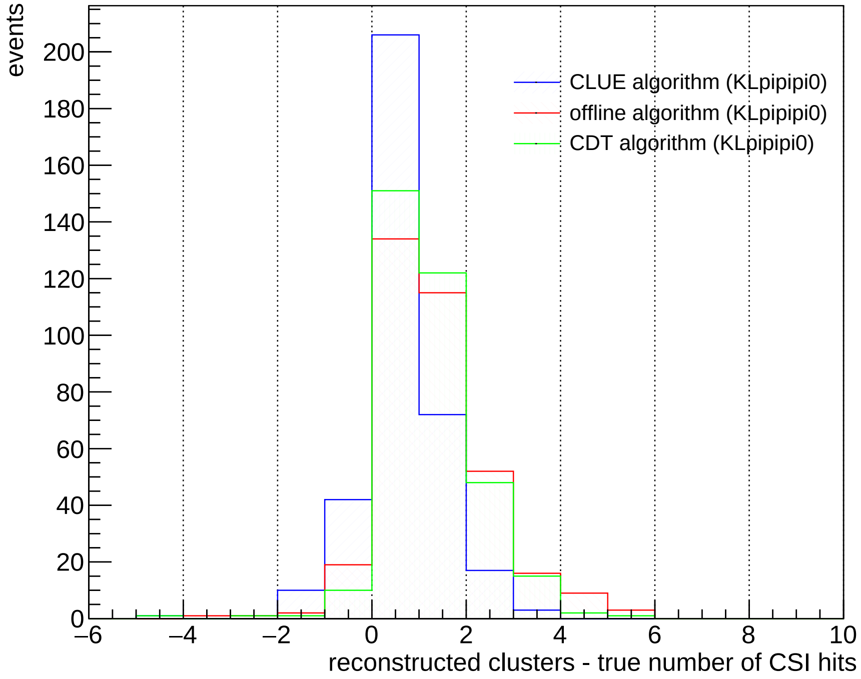


Figure G.3: Comparison of the L2 (CDT, green), the HLT (CLUE, blue), and the offline clustering algorithms against $K_L \to \pi^+\pi^-\pi^0$ events.

The large energy dispersion of pion clusters often leads to multiple local maximums within the same cluster, making the L2 algorithm likely to overestimate the number of clusters. The offline algorithm, will also tend to do so, mainly due to small energy splashes close but not connected to a cluster. The HLT algorithm can adjust its parameters to avoid most of these issues.

G.4 Conclusion

The potential of the new and more sophisticated CLUE algorithm to outperform the L2 and offline algorithms, particularly with events involving non-photon-like clusters, has been shown in this appendix. The usability of this clustering algorithm in KOTO's HLT during actual physics runs has been demonstrated during the 2024 beam-time.

Still, before implementing event selection based on the number of clusters computed at the HLT, the offline algorithm would need to be upgraded. so that usable events accepted by the HLT are not rejected by a less efficient algorithm offline. This will be particularly important in KOTO II, where charged pion clusters are expected to play a crucial role if the $K_L \to \pi^0 e^+ e^-$ analysis is included in the experiment's physics program. The long shutdown expected while KOTO II is installed should be considered as a good opportunity to study this offline clustering upgrade.

Appendix H

Web-based DAQ Monitoring System

Some of the plots displayed in the DAQ monitor system are shown in this section. Fig. H.1 (left) shows the number of events processed by the Top CDT ("L1" label), the OFC-II, and the HLT ("L3" label). Ideally, this distribution should be kept flat during a run. Spikes pointing down are spills in which either the accelerator delivered fewer protons than usual, or the Top CDT stopped issuing events after an incoming error signal from other DAQ modules. An abnormally high number of spikes should alert the DAQ team to investigate a possibly faulty module.

The average triggers per second, as accepted by the HLT ("L3 in the plots") is shown in blue at the bottom right.

The right plot in Fig. H.1 shows DAQ efficiency between L1 and OFC-II, between OFC-II and the HLT, and between L1 and HLT. This efficiency does not include the effect of error signals that cause spikes in the left plot. The main reason why this efficiency would become low are events dropped at the HLT due to missing or corrupted packets.

The horizontal axis in both plots is time (hh:mm). Every point corresponds to a spill.

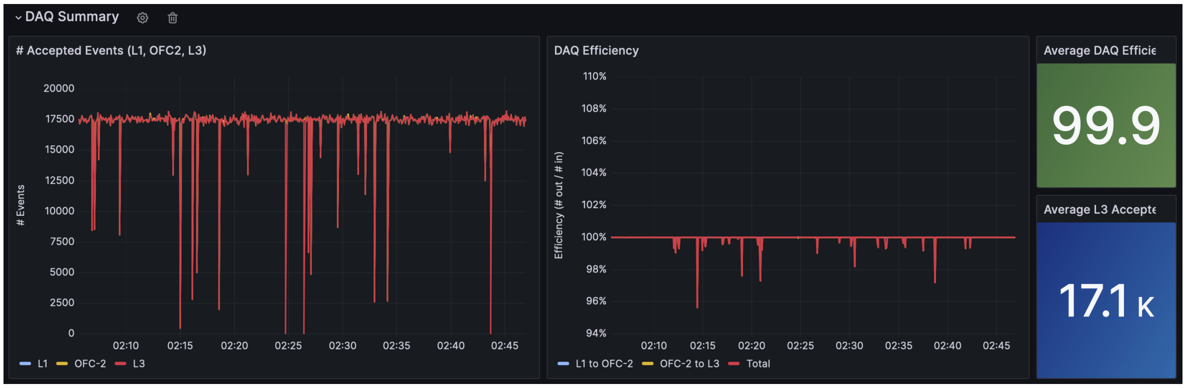


Figure H.1: Part of the DAQ monitor system.

The Disk Monitor plots are shown in Fig. H.2. The top bar shows the RAM usage, which should (and does) stay constant during a run. The left gauges show how filled are both of the RAID disks configured at the Disk Node. It should be ensured that they do not saturate. The right plot shows the evolution of the disk usage. This plot helps ensure that the data is being transferred to KEK and allows to make quick predictions on the disk usage evolution during the course of multiple runs.

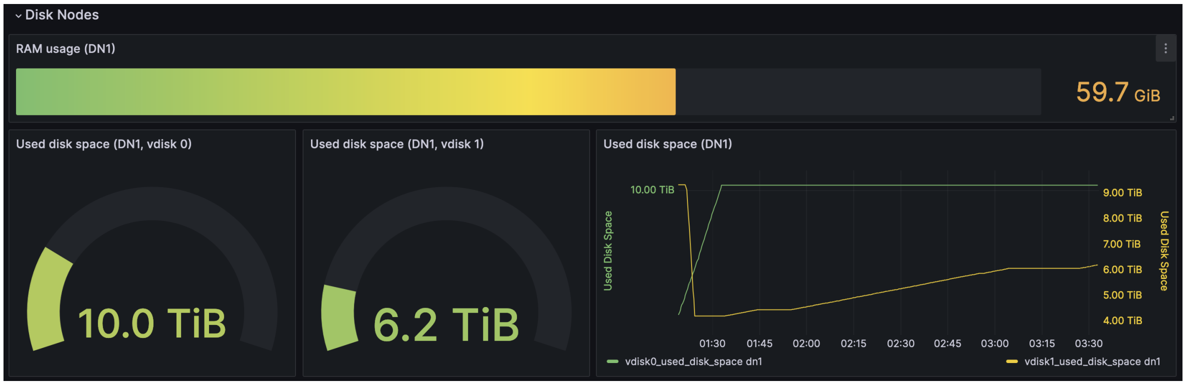


Figure H.2: The Disk Monitor.

Fig. H.3 shows the fraction of rejected events that do not pass each selection criteria. During this period, the TotalE and COE cuts were not applied to the 5-cluster $K_L \to 3\pi^0$ events. The COE was not applied to the $K_L \to \pi^0 e^+ e^-$ trigger either.

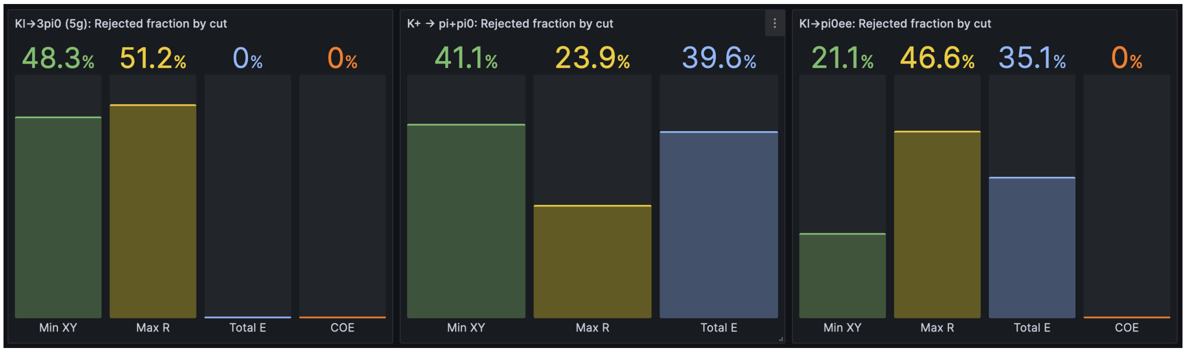

Kl→3pi0 (5g): Rejected fraction by cut
48.3%
51.2%
0%
0%
Min XY
Max R
Total E
COE
K+ → pi+pi0: Rejected fraction by cut
41.1%
23.9%
39.6%
Min XY
Max R
Total E
Kl→pi0ee: Rejected fraction by cut
21.1%
46.6%
35.1%
0%
Min XY
Max R
Total E
COE

Figure H.3: The fraction of rejected events that do not pass each selection criteria.

Other graphics monitor the total amount of rejected events, and the reduction factors trigger type by trigger type. General information such as the spill number, run number and average number of events per spill is also displayed.

Appendix I

Calorimeter Channels with Large Pedestal Suppression Inefficiency

Four channels were observed with inefficiency larger than 10% in Fig. 8.7. In this section, we will cover the reasons why these channels have such a high inefficiency.

Firstly, we consider a sample of 4000 events, from the same spill and from the same physics run. As they belong to the same spill, the online pedestal of each calorimeter channel is the same for all events in the spill. The offline pedestal, however, is calculated waveform by waveform, and therefore event by event. The mean offline pedestal of all calorimeter channels across these events is shown in Fig. I.1. The four channels with the highest inefficiency are marked with large red dots.

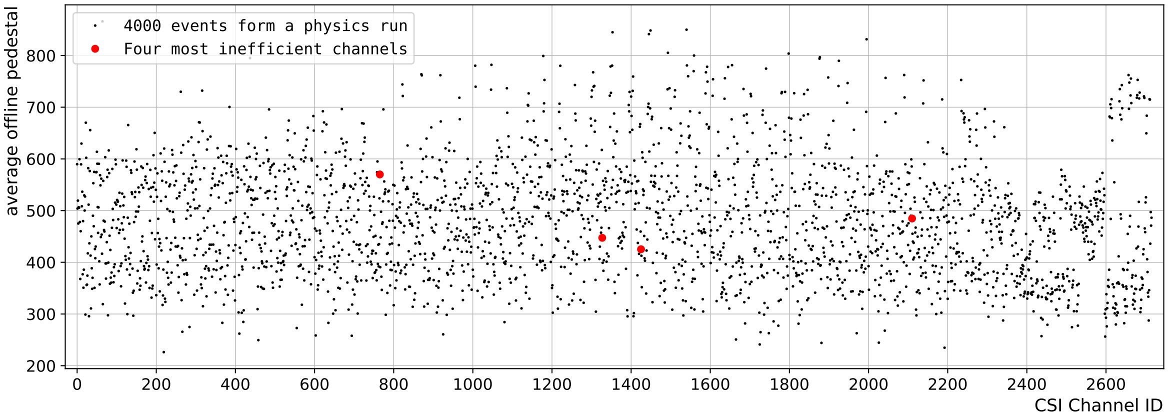


Figure I.1: Offline pedestal, averaged across 4000 events, for all calorimeter channels.

As seen in Fig. I.1, the offline pedestal of these four channels is, on average, consistent with the offline pedestal of the other channels. Known this, we can now evaluate how noisy the channels are, by looking at not the mean but the standard deviation of the offline pedestal. This is shown in Fig. I.2.

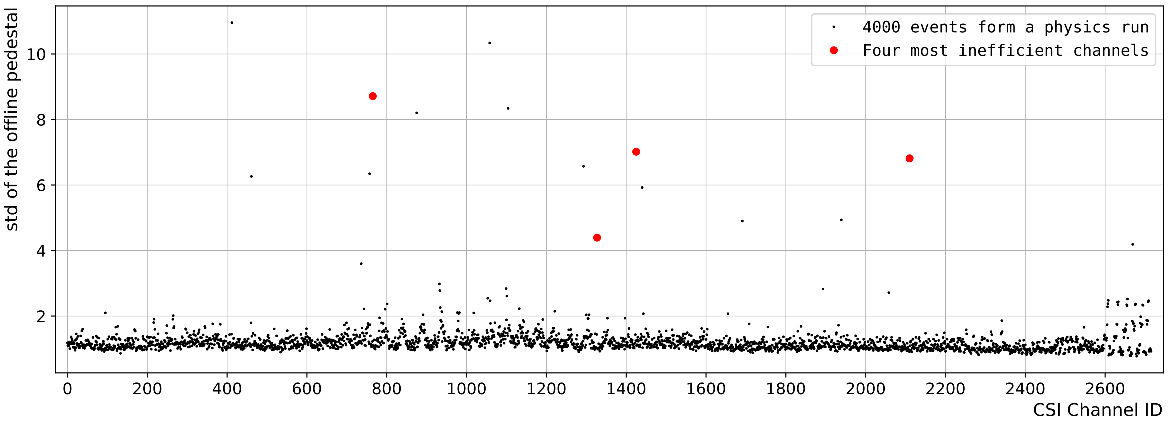


Figure I.2: Standard deviation of the offline pedestal distribution, for all calorimeter channels. Channels with low gain are included in the plot.

The four most inefficient channels are all extraordinarily noisy. For an average channel gain of 80 ADC counts per MeV (see Fig. 8.6 in the main text), a 1-count pedestal difference means a 64-count difference in the integrated ADC, and therefore a $\pm 1 \times 64/80 = \pm 0.8$ MeV difference. As shown in Fig. I.2, the standard deviations of the offline pedestal distributions of these channels are all between four and ten counts. Thus the standard deviation of their energy distribution when no hit is present is between 3 and 8 MeV. Due to the large offline pedestal fluctuations, the energy of waveforms without hit is likely to be calculated above 3 MeV, and to contribute to the inefficiency of the online pedestal suppression. This can be checked by looking at the hit rate of all calorimeter channels, and comparing the hit rate of these channels with the one of neighboring channels. The hit rate of the four inefficient channels is shown in Fig. I.3.

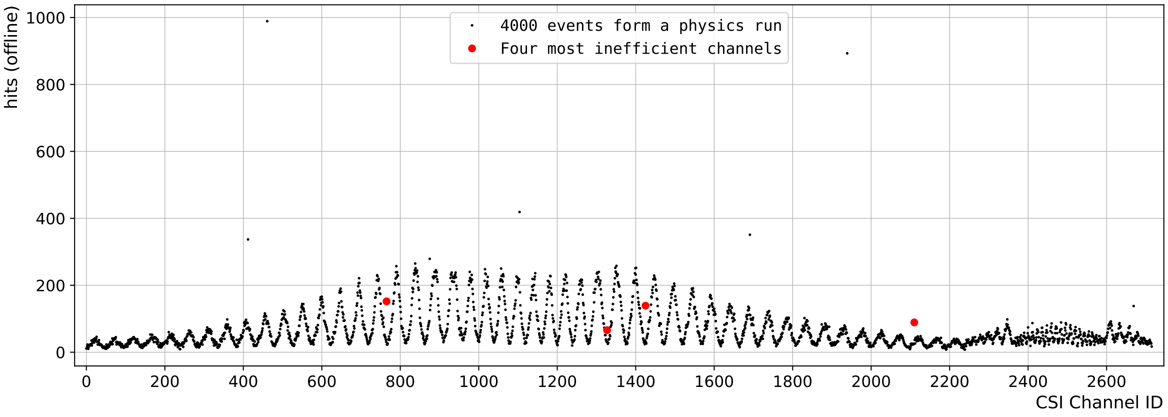


Figure I.3: Number of hits per channel, for all calorimeter channels. Channels with low gain are included in the plot.

The wavy pattern comes from the fact that channel IDs are assigned row by row. The hit rate in each row increases as we move towards the center of the calorimeter, and decreases again as we move far from it. The four inefficient channels have, as expected, higher hit rates than their healthy neighbors.

The online pedestal is less sensitive to large noise fluctuations, as it is calculated from 8000 samples (as opposed to 9 samples for the offline pedestal). For this reason, the online-calculated energy tends to be more accurate than the offline-calculated energy.

The high noise level could come from the PMT itself, from a component between the PMTs and the ADCs, from any ADC component, or from any cable and connector. Tests to identify the actual source of the noise would start by replacing the ADC with one known to be healthy. In most cases, this would solve the issue. Otherwise, major maintenance work would be needed, including checking the healthiness of the long signal cables and, in the worst-case scenario, having to replace a PMT.

Appendix J

K_L mass reconstruction from $K_L \to 3\pi^0$ events

The K_L mass distribution, as reconstructed from $K_L \to 3\pi^0$ events, was given in Fig. 12.7. A long tail towards high K_L mass was observed. The reconstruction of the K_L mass from $K_L \to 3\pi^0$ decays, as well as the reason for the tail towards high K_L mass, are explained in this section.

In section 2.2 we explained how the $\pi^0 \to \gamma\gamma$ decay vertex can be reconstructed from the energy and position of two photons recorded in the CsI calorimeter assuming that the pion decays in the beam axis. We also mentioned that because of the lifetime of the π^0 being very small ($c\tau_{\pi^0} = 25$ nm), the pion can be assumed to decay in the same position where it was produced by the K_L. Therefore, the reconstructed decay vertexes of the three pions in a $K_L \to 3\pi^0$ decay are expected to coincide within their uncertainties. This fact is used to reconstruct the three $\pi^0 \to \gamma\gamma$ decays from the six final-state photons in the $K_L \to 3\pi^0$ decay, as follows.

A total of 15 combinations of three pairs can be constructed from six photons. The kaon decay vertex can be calculated as the average pion decay vertex in each of the 15 cases. The goodness of each photon pairing combination is evaluated by performing the following χ^2 test:

$$\chi^2_{\text{vtx}} = \sum_{i=1}^{3} \left(\frac{z_i - \bar{z}}{\sigma_z} \right)^2 . \tag{J.1}$$

The sum goes over each constructed photon pair. z_i is the reconstructed decay vertex of each π^0. $\bar{z}$ is the average of the three pion decay vertexes, and σ_z its standard deviation.

Fifteen χ^2_{vtx} are obtained. The photon pairs producing the smallest χ^2_{vtx} are taken as the actual $\pi^0 \to \gamma\gamma$ decays that took place in the KOTO detector. The corresponding $\bar{z}$ is taken as the K_L decay z coordinate, making the kaon decay vertex $(x, y, z) = (0, 0, z_{\text{vtx}})$.

At this point, a small correction is applied to the K_L decay vertex x and y coordinates. The correction is based on the fact that, unlike the case of the $K_L \to \pi^0 \nu \overline{\nu}$ decay, the full energy of the $K_L \to 3\pi^0$ final state is measured in the calorimeter.

The correction process is illustrated in Fig. J.1 for the y coordinate, and the same process is applied to the x coordinate. The process starts by calculating the COE of the six clusters in the calorimeter, $y_{\mathrm{COE}} = \sum_i^6 E_i y_i / \sum_i^6 E_i$. The actual decay position of the K_L must then be somewhere on the straight line connecting the gold target (where the kaon was generated) and the COE on the calorimeter's surface. The correction to the K_L decay vertex y coordinate is then given by

$$\Delta y_{\mathrm{vtx}} = y_{\mathrm{COE}} \cdot \frac{z_{\mathrm{vtx}} - z_{\mathrm{target}}}{z_{\mathrm{CsI}} - z_{\mathrm{target}}} . \tag{J.2}$$

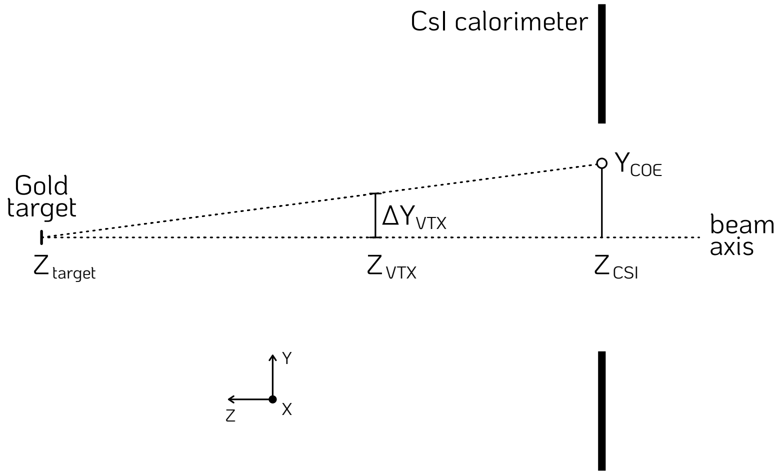


Figure J.1: Calculation of the K_L decay vertex correction in the y coordinate. Not to scale.

The corrected K_L decay vertex becomes $(x, y, z) = (\Delta x_{\mathrm{vtx}}, \Delta y_{\mathrm{vtx}}, z_{\mathrm{vtx}})$. This new vertex is used to calculate the linear momentum of all six photons. The energy and momentum of each pion are then calculated as the sum of the energy and momentum of the two photons in each π^0 decay. E_{K_L} and $\vec{p}_{K_L}$ are calculated from the energy and momentum of the three pions. The K_L mass is then reconstructed as

$$M_{K_L} = \sqrt{E_{K_L}^2 - \vec{p}_{K_L}^{\,2}} .$$

The Kaon mass will be overestimated if either $E_{K_L}^2$ or $\vec{p}_{K_L}^{\,2}$ are overestimated. The Kaon energy is generally accurate, but could be overestimated in the rare case of two unrelated events hitting the CsI calorimeter simultaneously, producing a sum of 6 photon clusters that are misreconstructed as a $K_L \to 3\pi^0$ decay. More commonly, the kaon momentum will be overestimated. This will happen if the photon momentum is overestimated, which will occur if the photon pairing is incorrect and z_{vtx} is misconstructed towards the CsI calorimeter. This is likely to happen in non 6-cluster $K_L \to 3\pi^0$ decays that are misidentified as $K_L \to 3\pi^0$ decays[1].

For example, assume a $K_L \to 3\pi^0$ decay where only 5 photons hit the CsI calorimeter, but one of them produces a secondary cluster so the total cluster count is six. The split cluster will lead to two of the six "photon" energies being underestimated, which according to Eq. 2.1 will lead to larger opening angles in the reconstructed π^0 decay, and thus to a pion decay vertex closer to the CsI calorimeter.

In any case, the χ^2_{vtx} of the best photon pairing will be expected to be large if any of the six photon clusters was misreconstructed. In the actual offline $K_L \to 3\pi^0$ analysis, a $\chi^2_{\text{vtx}} < 10$ cut is applied to the $K_L \to 3\pi^0$ sample to reject these misreconstructed decays[2]. Figure J.2 shows the shape of the K_L mass distribution for three ranges of χ^2_{vtx}. The combined histogram does not include any cut in χ^2_{vtx}, and leads to the same distributions observed in Fig. 12.7.

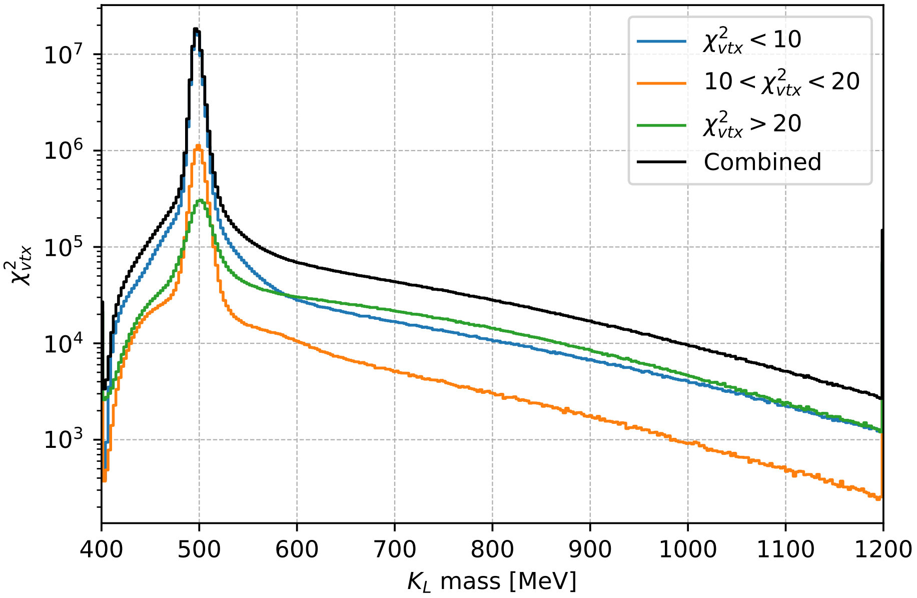


Figure J.2: Reconstructed K_L mass for three ranges of χ^2_{vtx}. The histograms are not normalized.

As χ^2_{vtx} increases, the likelihood that the event was misreconstructed increases. Consequently, the gaussian peak becomes wider and the non-gaussian tail more pronounced. By comparing the black (before any χ^2 cut) and blue (after a $\chi^2 < 10$ cut) histograms in Fig. J.2, we can observe the number of events in the gaussian peak not being significantly affected, while the number of events in the non-gaussian tail is reduced after applying the $\chi^2 < 10$ cut. Thus most of the events contributing to the non-gaussian tail at both sides of the peak are misreconstructed $K_L \to 3\pi^0$ decays.

[1] This statement has not been tested quantitatively by the author of this thesis

[2] Further cuts, such a window around the nominal K_L mass or a cut in the reconstructed z_{vtx} are also applied.

Appendix K

Overview of the $K_L \to 2\pi^0$ Background Estimation

The expected number of $K_L \to 2\pi^0$ background events is calculated as

$$\mathrm{BG}_{K_L \to 2\pi^0} = \mathrm{NF} \cdot \sum \left(\mathrm{SF}_{\mathrm{det1}} \cdot \mathrm{SF}_{\mathrm{det2}}\right) \ ,$$

where,

- NF is the normalization factor that accounts for the statistical difference between data and Monte Carlo simulations.
- The sum is over all simulation $K_L \to 2\pi^0$ events found in the $K_L \to \pi^0\nu\overline{\nu}$ signal region after applying all $K_L \to \pi^0\nu\overline{\nu}$ selection criteria.
- det1 and det2 are the two detectors hit by the two missing photons of those simulation $K_L \to 2\pi^0$ events.
- $\mathrm{SF}_{\mathrm{det1}}$ and $\mathrm{SF}_{\mathrm{det2}}$ are the ratios $\mathrm{Ineff.}_{\mathrm{data}}/\mathrm{Ineff.}_{\mathrm{MC}}$ for det1 and det2, obtained from the 5γ analysis.

The uncertainty of the $K_L \to 2\pi^0$ background expectation is calculated as

$$\sigma = \mathrm{NF} \cdot \sum \left(\sqrt{(\mathrm{SF}_{\mathrm{det1}} \cdot \sigma_{\mathrm{det2}})^2 + (\mathrm{SF}_{\mathrm{det2}} \cdot \sigma_{\mathrm{det1}})^2}\right)$$

where σ_{det1} and σ_{det2} are the uncertainties of $\mathrm{SF}_{\mathrm{det1}}$ and $\mathrm{SF}_{\mathrm{det2}}$ respectively. Since, in principle, an arbitrarily large amount of background events can be generated, the uncertainties σ_{det} are dominated by the statistical uncertainty of $\mathrm{Ineff}_{\mathrm{data}}$.