Commissioning and Low Latency Operation of the Graph Neural Network Electromagnetic Calorimeter Trigger at the Belle II Experiment

M. Neu[1], F. Baptist[1], I. Haide[1], Y. Unno[2], T. Ferber[1], J. Becker[1], K. Arai[3], Y.-T. Lai[4], T. Koga[4], M. Maushart[5], H. Nakazawa[6], V. Savinov[3], and K. Unger[1]
[1]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany [2]Hanyang University, Seoul, South Korea [3]University of Pittsburgh, Pittsburgh, United States [4]High Energy Accelerator Research Organiization (KEK), Tsukuba, Japan [5]Université de Strasbourg, Strasbourg, France [6]National Taiwan University, Taipei, Taiwan

***Abstract*—We present the commissioning and operation of the Graph Neural Network Electromagnetic Calorimeter Trigger Module (GNN-ETM) of the Belle II experiment at the SuperKEKB collider. The GNN-ETM processes calorimeter trigger cells as graph nodes to perform clustering and feature extraction. We fully integrate the system with the successive stages of the first-level trigger, develop slow-control drivers, and add online monitoring capabilities. We optimise the existing FPGA-based architecture through hardware–algorithm co-design, achieving an overall system latency of 1.053 us. Our hardware implementation is validated through register-transfer-level simulations, achieving bit-accurate agreement with the offline reference model. Online monitoring enables the measurement of instantaneous trigger rates, providing a quantitative basis for trigger-level performance studies. In summary, we report on the GNN-ETM as a fully operational, low-latency trigger module with online control and monitoring capabilities, compatible with the latency requirements of the Belle II first-level trigger system.**

***Index Terms*—Electromagnetic Calorimeter, Clustering, FPGAs, Graph Neural Networks, Machine Learning, Particle Physics, Trigger**

I. Introduction

GRAPH Neural Networks (GNNs) have received increasing attention in the domain of high-energy physics for applications such as track reconstruction [1], particle tracking [2], hit cleanup [3], calorimeter clustering [4], jet tagging [5], and event classification [6]. So far, the majority of these studies have been performed in an offline environment, that is, without systematic constraints on throughput or latency. GNNs for particle physics applications have nonetheless been deployed on Field-Programmable Gate Arrays (FPGAs) in prototypical studies for some time [3], [7]–[11]. Recently, we presented the first GNN-based reconstruction algorithm implemented on FPGAs within the readout infrastructure of a collider experiment trigger system [12].

The Belle II experiment is located at the SuperKEKB [13] electron-positron collider in Tsukuba, Japan [14]. At SuperKEKB, $4\,\mathrm{GeV}$ positrons collide with $7\,\mathrm{GeV}$ electrons at a center-of-mass energy of approximately $10.58\,\mathrm{GeV}$. At bunch crossing rates of up to $254.4\,\mathrm{MHz}$, defined by the SuperKEKB bunch filling pattern, reading out the full detector for each crossing is unfeasible due to bandwidth and storage limitations. Belle II therefore employs a trigger system that acts as an online filter, selecting potentially interesting events during operation based on a reduced subset of the detector signals. The first-level (L1) trigger operates under hard real-time requirements and is implemented as a pipeline of multiple FPGAs, each providing a custom compute module for a specific purpose. One stage of this pipeline is the electromagnetic calorimeter (ECL) trigger, which clusters the energy depositions recorded in the calorimeter.

The `GNN-ETM`, a real-time GNN-based calorimeter clustering algorithm implemented on FPGA, is one such module within the Belle II ECL trigger [12]. It currently operates in parallel to the existing ECL trigger system. It is not used for trigger decision, but records debug data for further development. The system supports a steady-state throughput of 8 million detector snapshots per second at an end-to-end latency of $3.168\,\mu\mathrm{s}$. While the module is real-time compliant, it is neither connected to the successive stages of the Belle II L1 trigger system nor does its latency meet the hard real-time deadline of $1.067\,\mu\mathrm{s}$ required to contribute to the L1 trigger decision.

To overcome these limitations, this work extends the original `GNN-ETM` of ref. [12] with the following contributions:

- We introduce model compression and physical design optimisations, reducing the end-to-end latency from $3.168\,\mu\mathrm{s}$ to $1.053\,\mu\mathrm{s}$ to meet the Belle II L1 trigger requirement, enabling the `GNN-ETM` to actively contribute to the L1 trigger decision for the first time.
- We implement hardware modules that generate binary flags, so-called trigger bits, which are used in the global trigger decision, making the system compatible with the full L1 trigger chain.
- We implement real-time monitoring of trigger rates measured at the last stage of the L1 trigger system, demonstrating complete integration and enabling physics analyses on the developed trigger system by comparing `GNN-ETM` with the existing `ICN-ETM`.

The authors would like to thank the Belle II collaboration for useful discussions and suggestions on how to improve this work. The training of the GNN-models was performed on the TOpAS GPU cluster at the Scientific Computing Center (SCC) at the Karlsruhe Institute of Technology (KIT).

- We have commissioned and operated the improved GNN−ETM in the Belle II experiment since December 2025, validating its performance under nominal data-taking conditions.

II. Background

The data acquisition (DAQ) system at the Belle II Experiment supports a maximum event readout rate of 30 kHz. To reduce the computational load, a L1 trigger system is employed [16]. This trigger operates synchronously with the detector frontend readout at 127.216 MHz, which is approximately half the bunch-crossing rate. Detector snapshots are processed strictly sequentially. To prevent buffer overflows in the DAQ readout system, the hard real-time latency budget, including data transfer, preprocessing, and synchronisation, is limited to 5.0 µs.

The Belle II first-level trigger system comprises dedicated subtriggers for the participating subdetectors. The task of the ECL trigger is to identify energy depositions for the global decision logic [17]. Figure 1 depicts a simplified schematic of the ECL L1 trigger system. For clarity, the connection to the Global Reconstruction Logic, where a matching between the ECL trigger and the trigger of other subdetectors is performed, is omitted. The ECL detector is composed of 8736 thallium-doped caesium iodide crystals, which are read out via the Frontend Electronics ① [18]. On the Frontend Electronics, crystals are first summed in the analog domain using 4×4 crystal groups. A waveform fit is then performed to extract the signal amplitude and timing relative to the global revolution clock signal ②. The resulting preprocessed groups of crystals, referred to as Trigger Cells (TCs), are uniquely identified by their positions within the detector and forwarded to the L1 trigger ③. For the ECL L1 trigger, three modules are shown in the figure.

The first module is the ICN−ETM, an isolated clustering logic implemented in the Isolated Cluster Number ECL Trigger Module [19], which aggregates the information from all 576 TCs supplied by the Frontend Electronics ④. Its purpose is to identify energy clusters and generate trigger bits.

The second module is the GNN−ETM Ⓐ, receiving a copy of the ECL data from the ICN−ETM. It runs in parallel to the ICN−ETM, also reconstructing energy clusters and generating trigger bits.

The ICN−ETM is connected to the Global Decision Logic (GDL) via optical fiber ⑤. As part of this work, the connection between GNN−ETM and GDL is established and evaluated. The GDL aggregates trigger bits from all subdetectors ⑥ to generate the global L1 trigger signal ⑦. Trigger bits are Boolean flags classifying the current detector snapshot. As an example, the two-cluster (C2) trigger bit on ICN−ETM is true if at least two clusters in the inner region of the ECL detector have been found with a per-cluster threshold of at least 100 MeV. The combination of trigger bits on GDL is a combinatorial Boolean equation joining all trigger bits. Inverted trigger bits may act as veto signals, for example, to suppress beam background.

The global L1 trigger signal is finally sent to the Frontend Timing Switch ⑧, which distributes the trigger signal to all modules in the Frontend Electronics as well as all modules in the trigger system ⑨. The Belle2Link Buffers store full-resolution and trigger data for a specified time period. When a trigger signal is received, data is sent to the PCIe40 Endpoints of the DAQ system, where the individual packets are assembled into a common format in the Event Builder. The events are then processed by the second filtering stage in the Belle II trigger system, the High Level Trigger (HLT), based on a CPU farm with soft real-time constraints. Finally, events that pass the HLT are persistently stored on disk for later analysis. All other data are lost and cannot be recovered.

Operating the GNN−ETM in the position described in Figure 1 imposes the following requirements on the system, based on the analysis in ref. [12]:

1) The system must exhibit deterministic latency to satisfy hard real-time deadlines.
2) The critical-path latency ① → ⑨ must not exceed $R_{\mathrm{L}} = 5.0\,\mu\mathrm{s}$, a constraint imposed by the finite depth of the data buffers on the vertex detector frontend electronics modules. In practice, only a small fraction of this latency is available for a subsystem trigger module. For the Belle II ECL L1 trigger system, the latency ④ of the module replacing the current ICN−ETM must not exceed $1.067\,\mu\mathrm{s}$[1].
3) The system must sustain the full input rate of the respective subdetector, which for the Belle II ECL L1 trigger amounts to $R_{\mathrm{th}} = 8\,\mathrm{MHz}$.
4) The system must maintain 100 % uptime, as any disruption halts the entire experiment for the duration of the fault.

III. GNN-ETM Architecture

The GNN−ETM is realised as an FPGA-based architecture on the fourth generation of the Universal Trigger Board (UT4). It interfaces with the up- and downstream modules in the Belle II L1 trigger chain via AXI-Stream [20]. These interfaces are carried over optical links realised with gigabit transceivers. For slow control and monitoring, the FPGA communicates with a general-purpose CPU via the Versa Module Eurocard (VME) bus [21]. Debug data is transmitted via the Belle2Link physical layer protocol [15], [22].

A system overview of the GNN−ETM is shown in Figure 2. It comprises three submodules on the critical path: the preprocessing stage, the GNN dataflow accelerator, and the postprocessing stage. Further submodules provide monitoring, control, and the Belle2Link media access control, which handles the interfaces to VME and to the Belle2Link for debugging. Compared to the GNN−ETM described in ref. [12], we introduce the following modifications to the architecture.

First, we adapt the Chisel-based [23] preprocessing and postprocessing stages to include clock domain crossings between the submodules, enabling a user-defined system frequency f_{sys} for the GNN dataflow accelerator. We choose

[1]The value here differs from the previously reported value in ref. [12]. The reason is that we do not swap the order of ICN−ETM and GNN−ETM in the ECL L1 trigger system in this work, as it incurs additional implementation overhead.

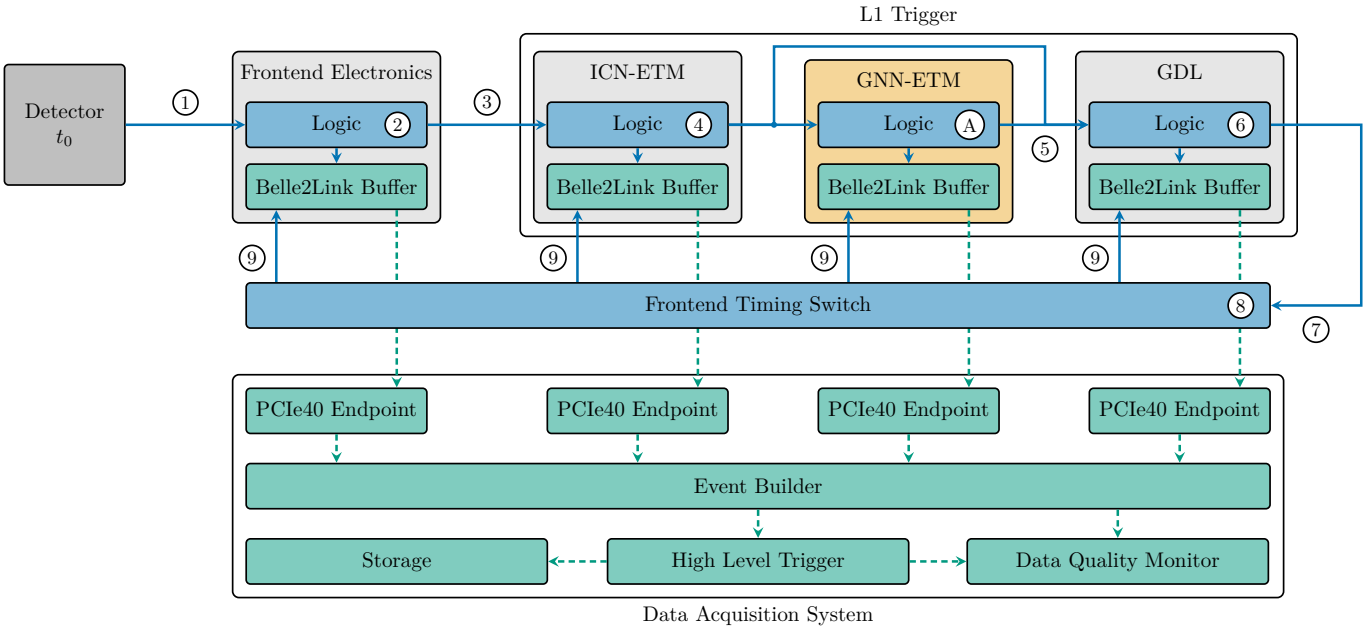


Fig. 1. Simplified overview of the ECL L1 trigger system at the Belle II Experiment. Components of the L1 trigger system that must satisfy hard real-time constraints are highlighted in blue, and components of the DAQ system that must satisfy soft real-time constraints are highlighted in green. The GNN−ETM developed in this work is highlighted in orange. Numbered circles label selected components and connections referenced in the text. t_0 denotes the time of a detected bunch crossing. Adapted from ref. [12]. A detailed overview of the DAQ system is given in ref. [15].

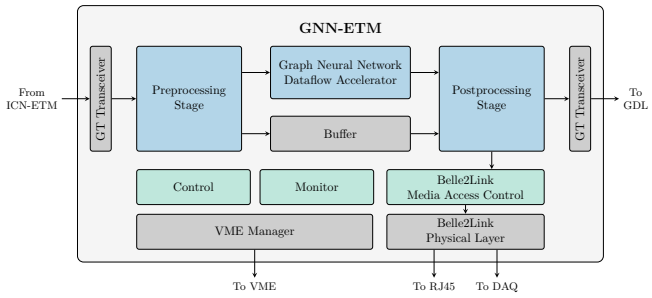


Fig. 2. Overview of the GNN−ETM system architecture. Existing components of the base firmware is shown in grey. Modules introduced in this work are coloured: modules on the critical path of the trigger system are blue, the remaining modules are green. Adapted from ref. [12].

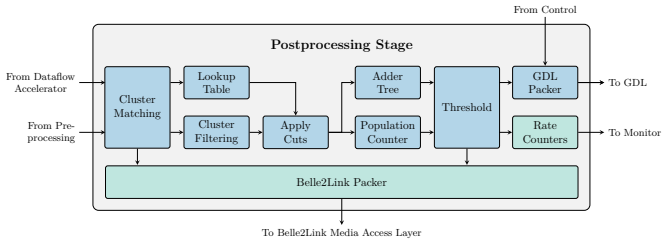


Fig. 3. Overview of the GNN−ETM postprocessing stage. Latency-critical submodules are blue; the remaining submodules are green. The external interfaces of this submodule are also shown in Figure 2.

synchronous clock domain crossings to mitigate boundary effects and to minimise the latency overhead of the crossing.

Second, we fully integrate the GNN−ETM with the slow control system of the Belle II experiment [24], [25]. This integration enables automated configuration, error handling, and monitoring. For example, at the start of every run the GNN−ETM parameters are read from a database via Network Shared Memory 2 (NSM2) [26] and written directly to the firmware register map through VME. The same interface is used for real-time monitoring of the GNN−ETM trigger rates.

Third, we extend the postprocessing stage with the generation of trigger bits and a monitoring counter for each trigger bit. An overview is given in Figure 3.

The postprocessing stage extracts the final trigger bits from the GNN−ETM cluster predictions. It comprises ten submodules, eight of which reside in the critical timing path.

First, the cluster matching module synchronises the cluster features from the GNN dataflow accelerator with the corresponding TCs from the preprocessing stage. Since TCs are processed strictly in order, the matching problem reduces to concatenating synchronised trigger-cell and cluster features. Second, a lookup table retrieves the static TC information required by subsequent modules, such as the position of each TC. In parallel with the lookup, the cluster filtering submodule applies the condensation point selection mask, setting all inactive clusters to zero so that only valid clusters propagate through the pipeline. Third, energy and multiplicity cuts are applied to form the individual trigger bits: parallel adder trees compute energy-sum trigger bits, which calculate the total energy deposited in the ECL, while population counters capture the cluster-counting trigger bits, which count the number of clusters above a certain energy threshold in the ECL. Finally, all trigger bits are packed for transmission to the GDL module.

Separately, two monitoring modules process the data further.

The raw rate of each trigger bit is tracked using 32 bit counters, which are read out via the slow control interface over VME. In addition, a copy of all trigger bits is sent to the Belle2Link media access control system for further debugging of the GNN−ETM in operation.

IV. Deployment

For deploying the GNN algorithm on the GNN−ETM architecture, we use the approach previously described in ref. [12]. The deployed GNN model is the CaloClusterNet, a dynamic GNN model, based on the GravNet [27] layer and the Object Condensation algorithm [28]. Because the baseline version of the CaloClusterNet, hereafter referred to as *Armadillo* CaloClusterNet, does not meet the latency requirements of the Belle II L1 trigger system, we apply three optimisation steps to reduce the overall latency of the system while minimising the loss of algorithmic performance of the model. As the target FPGA on the UT4, the AMD Ultrascale XCVU190 is chosen. For the deployment and evaluation, we use AMD Vitis 2024.2 [29] and AMD Vivado 2024.2 [30]. In the following, we describe four design iterations: Design iteration ① describes the baseline implementation from ref. [12] using the *Armadillo* CaloClusterNet. Design iteration ② describes the deployment of the improved version after model compression in Section IV-A. Design iteration ③ describes the deployment after manual floorplanning in Section IV-B. Design iteration ④ describes the deployment after DSP-level optimisations in Section IV-C.

A. Model Compression

In the first design iteration, we aim to reduce the latency of the baseline *Armadillo* CaloClusterNet through optimised quantisation-aware training. We use QKERAS [31] with the adaptation from refs. [32]–[34] to implement and train the network. Our *Sunset* CaloClusterNet incorporates the following optimisations in comparison to the original *Armadillo* CaloClusterNet.

First, we reduce the number of GravNet blocks from two to one, significantly decreasing the network's complexity.

Second, we perform a manual hyperparameter search for layerwise heterogeneous fixed-point quantisation. To reduce the hardware resource utilisation on the FPGA, we impose a hard limit of 8 bit per network layer. Quantisation is uniform within each layer. Power-of-two quantisation is chosen to enable efficient requantisation between adjacent neural network layers.

Third, we apply stochastic rounding and add quantisation noise during training to improve the trainability of the neural network under stricter quantisation schemes by reducing the bias imposed by the quantisation scheme [35], [36].

The model topology of our resulting network is shown in Figure 4. Most values are quantized to Q1.7 and Q2.6, meaning that most values have to lie in the range $[-1, 1]$ and $[-2, 2]$ respectively. At the interfaces, we keep the Q4.12 and Q5.11 quantisation to retain the full resolution. In comparison to the *Armadillo* CaloClusterNet, the *Sunset* CaloClusterNet does not yield a significantly lower algorithmic performance.

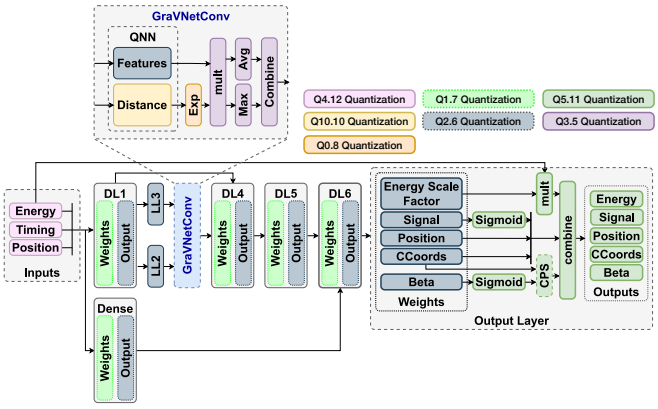


Fig. 4. Our compressed *Sunset* CaloClusterNet neural network architecture. This model architecture is derived from ref. [12].

B. Floorplanning

In the second design iteration, we consider manual floorplanning to improve the design's routability and increase the maximum achievable design frequency f_{sys}. Because the AMD Ultrascale XCVU190 is composed of three identical Super Logic Regions (SLRs), manually floorplanning the architecture can drastically improve f_{sys}, as netlist wires crossing these SLRs without further optimisation result in either congested routes or routing delays that dominate the path. In ref. [12], the GNN dataflow accelerator operates at 127.216 MHz, while the preprocessing stage already runs at the doubled frequency of 254.432 MHz. Through manual floorplanning, we now also operate the GNN dataflow accelerator at 254.232 MHz, applying the floorplanning constraints described in Figure 5. Notably, using the previous model compression in Section IV-A, we can implement the GNN dataflow accelerator on a single SLR.

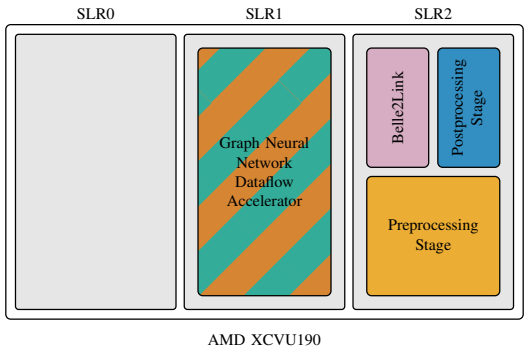


Fig. 5. Floorplan constraints of the GNN−ETM for implementation with AMD Vivado 2024.2. Hierarchical modules are the same as in Figure 2. Hierarchical modules that do not appear in the floor plan are not subject to any location constraints.

C. DSP Mapping

Due to the multiplication-heavy design of our dataflow accelerator, we usually maximise the utilisation of the hard DSP blocks on the FPGA, as they tend to achieve a better performance than realising multiplication in distributed logic.

However, during implementation, we observed that the distribution of the hard-macro DSPs on the FPGA results in suboptimal placement and, in turn, routing congestion, making timing closure more difficult. In addition, hard-macro DSPs on the AMD Ultrascale fabric require up to three internal registers for full pipelining, resulting in some latency overhead. Thus, in a final optimisation step, we disable all hard-macro DSPs via the corresponding configuration options in AMD Vitis 2024.2 and AMD Vivado 2024.2.

V. Performance Analysis

We measure the end-to-end system performance of GNN−ETM in two different ways before commissioning the system in the Belle II L1 trigger system. First, we perform a cycle-accurate register-transfer-level simulation of the complete design in ModelSim 2023.4 [37]. From this simulation, we validate functional correctness, verify that the throughput requirement is met, and derive the system's end-to-end latency in Section V-A. Second, we implement the design on the UT4 board and validate that all timing constraints are met after place-and-route. We analyse the AMD Vivado 2024.2 report after implementation in Section V-B.

A. Latency

Figure 6 depicts the latency of all four design iterations. The baseline design in ① requires 3168 ns for the end-to-end inference, including the preprocessing stage. The baseline design does not include generating trigger bits. After applying the model compression in ②, the latency is reduced by 1282 ns. Further optimisation of the floorplanning in ③ reduces the latency by an additional 735 ns. A final 98 ns are saved in ④, resulting in an end-to-end latency of the GNN−ETM of 1053 ns, which is a 3.01× reduction over the baseline version. Breaking down the latency, 385 ns are used by the preprocessing stage, 507 ns by the CaloClusterNet, 130 ns by the Condensation Point Selection, and 31 ns by the postprocessing stage.

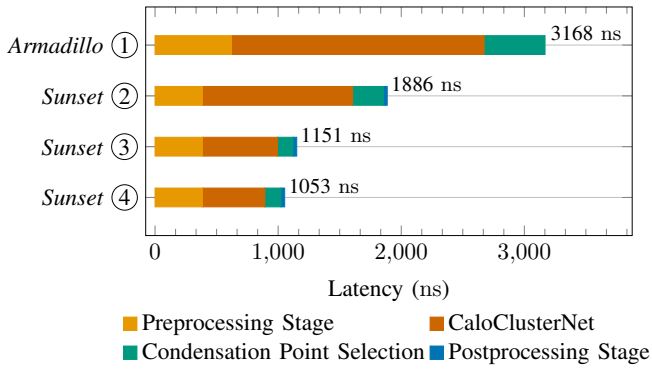


Fig. 6. End-to-end latency for the complete inference chain on the UT4 with an AMD Ultrascale XCVU190 FPGA. Design iterations ①—④ are presented.

B. System Resource Utilisation

Figure 7 depicts the system resource utilisation after place and route on the UT4 for the baseline design ① and the final design ④. In comparison between the two versions, both flip-flops (FFs) and lookup tables (LUTs) are reduced by approximately 50 %. The main reason for this difference lies in the removal of one GravNet layer and the reduced precision of all Dense layers. This effect also influences the utilisation of the successive Condensation Point Selection submodule. Similarly, DSPs are now unused, and multiplications are mapped to distributed logic after applying the optimisation from Section IV-C. A slight increase in Block RAM (BRAM) utilisation is observed due to the addition of trigger bits in the Belle2Link readout.

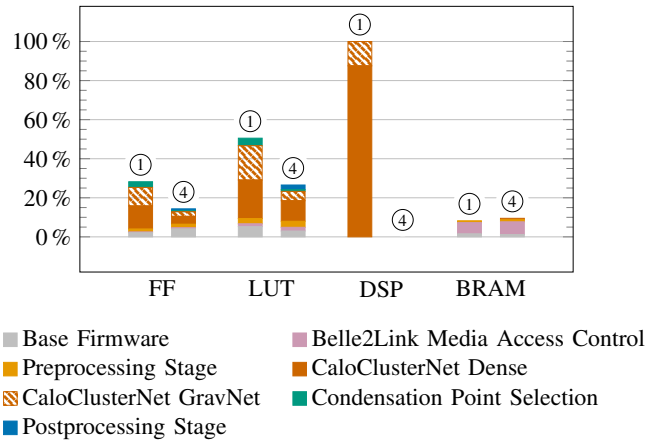


Fig. 7. Utilisation of system resources on the AMD Ultrascale XCVU190 FPGA for the GNN−ETM with *Armadillo* ① and *Sunset* ④ CaloClusterNet model.

VI. Commissioning & Operation

To validate the GNN−ETM, we commission the system in the Belle II L1 trigger system as depicted in Figure 1. Compared with the commissioning of the previous system in ref. [12], we develop slow-control and monitoring software compatible with the general Belle II run control and add the upstream link to the GDL. Configurations, trigger rates, and monitoring flags are broadcast via NSM2, and additionally registered as process variables (PVs) in the Belle II EPICS archiver database for later analysis [38]. Polling-sampling mode is used for logging GNN−ETM PVs at a rate of 1 Hz.

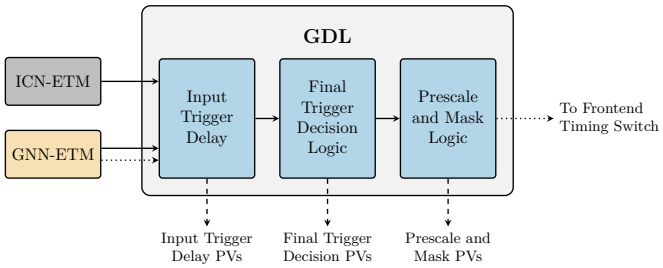


Fig. 8. Interfaces between GNN−ETM, ICN−ETM, and GDL. Interfaces via gigabit transceivers are shown as solid arrows. Interfaces via twisted-pair cables are depicted as dotted arrows. Slow control interfaces are depicted as dashed arrows.

Figure 8 depicts the interfaces of ICN−ETM, GNN−ETM, and GDL in the Belle II L1 trigger system. Both ICN−ETM and

GNN−ETM are connected via gigabit transceivers (GTs) to the GDL. This connection via optical fibres enables the transmission of large packets of up to 768 bit per 127.216 MHz system clock cycle, with a latency of approximately 200 ns. To avoid this latency overhead, we add a single twisted-pair (LEMO) cable between GNN−ETM and ICN−ETM. This transmission interfaces directly with the high-speed-capable pins on the FPGA, avoiding error correction, synchronisation, and other physical-layer overhead present in the GT connections.

On the GDL, trigger bits are received from all L1 trigger subsystems. For simplicity, only ICN−ETM and GNN−ETM are shown in the figure. In general, trigger bits pass through three submodules:

First, the input trigger delay submodule checks the connection to the GDL for liveness and measures the subtrigger latency via the active signal. Variable-length shift registers are used to synchronise all incoming subtriggers based on the measured delay. The GDL implements rate counters for the active signal and all trigger bits after this stage, measuring the raw input trigger rate and storing it in the EPICS archiver database. Trigger bits included in the GDL are monitored via the Input Trigger Delay PVs.

Second, an optional veto (bitwise AND with the inverted veto signal) is applied to the input trigger signals and again recorded as Final Trigger Decision PVs. Two vetoes used for the GNN−ETM trigger signals are the injection veto and the Bhabha veto. The injection veto suppresses beam-induced background. The Bhabha veto suppresses the high-rate Bhabha scattering process. Without applying these vetoes, both would dominate the resulting trigger rates.

Third, a prescale and mask value is applied to all trigger bits. This effectively enables the run operators to disable single bits (masking) or to reduce the trigger rate of these bits by triggering only on the Nth occurrence (where N is the prescale factor). The output of this stage is recorded as Prescale and Mask PVs.

In this work, GNN−ETM trigger bits are monitored in the GDL, but do not actively contribute to the trigger decision in the Belle II L1 trigger system. In the following, GNN−ETM will be evaluated in runs. A run is defined as a data-taking period during which the experiment and the accelerator configuration remain constant. We differentiate between two run types. Cosmic runs are data-taking periods without beam, whereas beam runs are data-taking periods in which the beams collide in the interaction point of the Belle II Experiment.

A. Latency Measurement

We derive the end-to-end latency and the latency requirement for GNN−ETM in a beam run. The system latency is measured by observing the histogram of the rising-edge clock counter implemented on the GDL, recorded over the full run. The observed input delay is denoted by $\hat{t}$. The rising-edge clock counter continuously monitors each trigger bit and stores the delay value with a resolution of 32 ns for both the gigabit transceiver and the twisted-pair cable. For the analysis, a histogram of arrival times is recorded over a full run using a dedicated clock counter module on the GDL. The upper acceptable input delay at the GDL has been defined as 20 system clock cycles, or 640 ns, based on experimental evaluation during data acquisition (DAQ) stress tests. The input delay is measured against an arbitrarily chosen reference time on the GDL, which depends on the clock distribution architecture at Belle II.

To relate a connection between $\hat{t}$ and the actual GNN−ETM latency $\hat{t}_{\mathrm{gnn}}$, we apply the following offsets: First, we apply the programmable delay offset t_{FAM} from the Frontend Analog Module to remove the effects of different run configurations. Second, we apply an offset based on the difference between the simulated cycle-accurate latency t_{sim} and the 95 % quantile $\hat{t}_{0.95}$ measured via the twisted-pair cable:

$$\hat{t}_{\mathrm{gnn}} = \hat{t} + t_{\mathrm{FAM}} + t_{\mathrm{sim}} - \hat{t}_{0.95} \tag{1}$$

Figure 9 shows the adjusted latency measurement with $t_{\mathrm{sim}} = 1053\,\mathrm{ns}$, and $\hat{t}_{0.95} = 480\,\mathrm{ns}$. The twisted-pair cable meets the latency requirement. For the configuration $t_{\mathrm{FAM}} = -146\,\mathrm{ns}$, the latency margin on GNN−ETM is 160 ns for trigger bits transmitted via twisted-pair cable. Without this configuration delay offset, the latency margin shrinks to 14 ns. In both configurations, the latency of the GNN−ETM is too high to transmit trigger bits over gigabit transceivers, due to the overhead of the physical communication layer.

Three latency bounds can therefore be derived from experimental measurements:

1) In the current configuration, a latency bound of 1067 ns is derived.
2) When the programmable delay offset is applied at the Frontend Analog Module, the latency budget increases to 1213 ns.
3) Additionally swapping GNN−ETM and ICN−ETM increases the latency budget further up to 1367 ns.

To conclude, GNN−ETM is ready to partake in active trigger decisions of the Belle II L1 trigger system with a single, runtime-reconfigurable trigger bit via twisted-pair cable.

B. Trigger Rate Monitoring

In the following, we compare the trigger rates for the C2 trigger bit on the existing ICN−ETM and GNN−ETM, using this representative trigger bit to demonstrate trigger rate monitoring. The C2 trigger bit is a Boolean decision variable which is true if at least two clusters are detected in the ECL inner region in the 250 ns observation window of the ECL L1 trigger system. In both systems, a per-cluster energy cut of 100 MeV is applied.

Figure 10 shows a comparison of trigger rates between the ICN−ETM and the GNN−ETM for the C2 trigger bit. We select two representative runs from the Belle II operation between May and June 2026 to demonstrate the functionality of the trigger rate monitoring. Figure 10a shows a cosmic run without beam collisions in June 2026. Figure 10b shows a physics run with beam collisions in May 2026. The rates are based on the Final Trigger Monitor PVs on the GDL after applying both the Bhabha and injection vetoes. As a baseline, we depict the trigger rate of the existing ICN−ETM C2 trigger bit. For

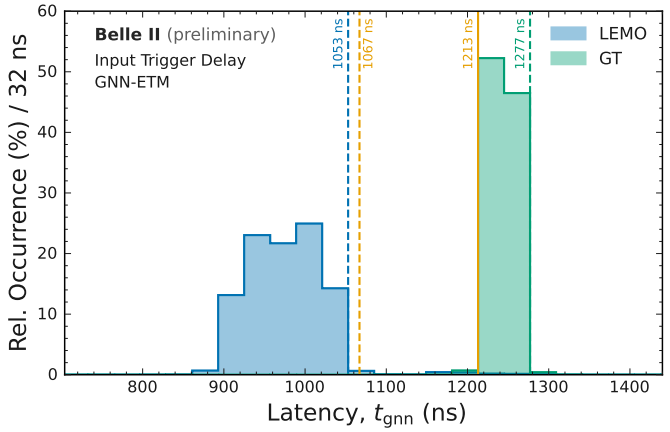
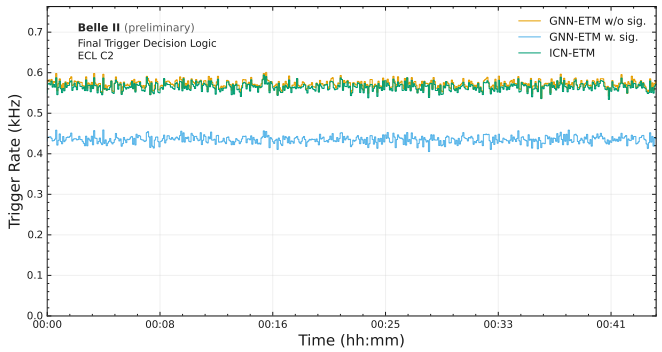
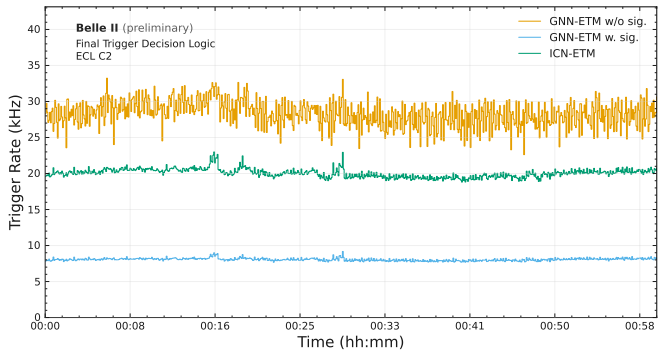


Fig. 9. Relative occurrence of the GNN−ETM latency $\hat{t}_{\mathrm{gnn}}$ of a trigger bit from GNN−ETM, measured at the GDL. GT equals transmission via gigabit transceiver. LEMO equals transmission via twisted-pair cable. Blue and green dashed lines denote the 95 % quantile for the respective distribution. The orange line describes the latency requirement for partaking in the active trigger decision with $t_{\mathrm{FAM}} = -146\,\mathrm{ns}$. The dashed orange line describes the latency requirement for partaking in the active trigger decision with $t_{\mathrm{FAM}} = 0\,\mathrm{ns}$.



(a) Cosmic run



(b) Beam run

Fig. 10. Comparison of C2 trigger rates between ICN−ETM and GNN−ETM based on monitoring PVs on the GDL. GNN−ETM trigger rates are shown both with the signal classifier (w. sig.) and without the signal classifier (w/o sig.).

the GNN−ETM, we show two versions: First, we show the C2 trigger bit based on the clusters selected by the condensation-point selection algorithm without consideration of the signal classifier per-cluster output [12]. This trigger rate is denoted as GNN−ETM w/o sig. Second, we show the same C2 trigger bit, but we apply the signal classifier to each cluster. As a result, background clusters are masked, and the total number of clusters per detector snapshot decreases, so the threshold of at least two clusters is reached in fewer cases. This trigger rate is denoted as GNN−ETM w. sig.

In Figure 10a, we observe that the C2 trigger rates of the ICN−ETM and the GNN−ETM without a signal classifier are almost identical. Applying the signal classifier on GNN−ETM reduces the C2 trigger rate by approx. 100 Hz. In Figure 10b, we observe a higher C2 trigger rate for the GNN−ETM w/o sig. in comparison to the ICN−ETM. A potential cause of this increased rate is the ability of GNN−ETM to split energy depositions into multiple clusters. Another potential cause is the characteristic of the ICN−ETM, to shift the position of a cluster towards the forward endcap due to the way TCs are defined in the inhomogeneous endcap region. Thus, the ICN−ETM is more likely to have only one cluster in the ECL inner region, as required by the C2 trigger bit, which leads to a lower trigger rate than the GNN−ETM. After applying the signal classifier, the trigger rate of GNN−ETM drops significantly below the ICN−ETM rate.

In general, both online measurements of the GNN−ETM and the ICN−ETM confirm the trends presented in ref. [12]:

1) For cosmic runs without beam background, the cluster-finding performance of ICN−ETM and GNN−ETM w/o sig. is almost identical.
2) The GNN−ETM signal classifier can significantly reduce the trigger rate.
3) For beam runs, a greater difference in trigger rates is expected between GNN−ETM and ICN−ETM.

Nevertheless, a more thorough analysis is required to make quantitative statements on the two systems. By making online monitoring available, this work lays the groundwork for a quantitative comparison of the two modules in the Belle II ECL trigger system.

VII. Conclusion

In this work, we have presented the commissioning and low-latency operation of the GNN−ETM, a GNN-based calorimeter clustering trigger algorithm, in the Belle II experiment. For the commissioning of the GNN−ETM in the L1 trigger system, we have reduced the end-to-end latency of the FPGA-based system by a factor of 3 from 3168 ns to 1053 ns. In addition, we have integrated trigger-bit generation into the postprocessing stage and completed the connection between the GNN−ETM and the GDL. We confirm in a measurement that the end-to-end latency target of 1067 ns is met with a margin of 14 ns during operation. Structural modifications of the Belle II ECL L1 trigger relax the latency budget to up to 1367 ns, enabling the integration of more complex trigger bits. In addition, the logging of trigger rates in the Belle II EPICS database enables a quantitative comparison of the GNN−ETM and the ICN−ETM in a physics analysis.

References

[1] L. Reuter *et al.*, "End-to-End Multi-track Reconstruction Using Graph Neural Networks at Belle II," *Computing and Software for Big Science*, vol. 9, no. 1, p. 6, Dec. 2025.
[2] A. Elabd *et al.*, "Graph Neural Networks for Charged Particle Tracking on FPGAs," *Frontiers in Big Data*, vol. 5, p. 828666, Mar. 2022.

[3] G. a. o. Heine, "Hardware-accelerated GNN-based hit filtering for the Belle II Level-1 trigger," *Journal of Instrumentation*, vol. 21, no. 02, p. C02007, 2 2026.

[4] F. Wemmer *et al.*, "Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks," *Comput. Softw. Big Sci.*, vol. 7, no. 1, 12 2023.

[5] H. Qu and L. Gouskos, "Jet tagging via particle clouds," *Physical Review D*, vol. 101, no. 5, p. 056019, Mar. 2020.

[6] J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph neural networks in particle physics," *Machine Learning: Science and Technology*, vol. 2, no. 2, p. 021001, Jan. 2021.

[7] Z. Que *et al.*, "JEDI-linear: Fast and Efficient Graph Neural Networks for Jet Tagging on FPGAs," 8 2025.

[8] S. Dittmeier, "Online track reconstruction with graph neural networks on FPGAs for the ATLAS experiment," *EPJ Web Conf.*, vol. 337, 2025.

[9] M. Neu *et al.*, "Real-Time Graph-based Point Cloud Networks on FPGAs via Stall-Free Deep Pipelining," in *2025 38th SBC/SBMicro/IEEE Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2025.

[10] Z. Que *et al.*, "LL-GNN: Low Latency Graph Neural Networks on FPGAs for High Energy Physics," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 2, pp. 1–28, Mar. 2024.

[11] S.-Y. Huang *et al.*, "Low Latency Edge Classification GNN for Particle Trajectory Tracking on FPGAs," in *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*. Gothenburg, Sweden: IEEE, Sep. 2023, pp. 294–298.

[12] I. Haide *et al.*, "Real-Time Graph Neural Networks on FPGAs for the Belle II Electromagnetic Calorimeter," *JINST*, 2026.

[13] K. Akai, K. Furukawa, and H. Koiso, "SuperKEKB Collider," *Nucl. Instrum. Meth. A*, vol. 907, pp. 188–199, 2018.

[14] T. Abe *et al.*, "Belle II Technical Design Report," 11 2010.

[15] S. Yamada *et al.*, "Data Acquisition System for the Belle II Experiment," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 3, 2015.

[16] Y.-T. Lai *et al.*, "Design of the Global Reconstruction Logic in the Belle II Level-1 Trigger system," *Nucl. Instrum. Meth. A*, vol. 1078, 2025.

[17] S. Kim *et al.*, "Status of the electromagnetic calorimeter trigger system at Belle II," *J. Phys. Conf. Ser.*, vol. 928, 11 2017.

[18] B. Shwartz and BELLE II calorimeter group, "Electromagnetic calorimeter of the Belle II detector," *Journal of Physics: Conference Series*, vol. 928, p. 012021, Nov. 2017.

[19] B. Cheon *et al.*, "Electromagnetic calorimeter trigger at Belle," *Nucl. Instrum. Meth. A*, vol. 494, no. 1, 2002.

[20] Arm Limited, *AMBA AXI-Stream Protocol Specification*, 2021, iHI 0051B. [Online]. Available: https://developer.arm.com/documentation/ihi0051/latest/

[21] *IEEE Standard for a Versatile Backplane Bus: VMEbus*, Institute of Electrical and Electronics Engineers, New York, NY, USA, 1987.

[22] D. Sun *et al.*, "Belle2Link: A Global Data Readout and Transmission for Belle II Experiment at KEK," *Physics Procedia*, vol. 37, pp. 1933–1939, 2012.

[23] J. Bachrach *et al.*, "Chisel: Constructing Hardware in a Scala Embedded Language," in *Proceedings of the 49th Annual Design Automation Conference*. San Francisco, California: ACM, 6 2012.

[24] T. Konno *et al.*, "The Slow Control and Data Quality Monitoring System for the Belle II Experiment," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 897–902, Jun. 2015.

[25] C.-H. Kim *et al.*, "Trigger slow control system of the Belle II experiment," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1014, p. 165748, Oct. 2021.

[26] M. Nakao and S. Suzuki, "Network shared memory framework for the Belle data acquisition control system," in *1999 IEEE Conference on Real-Time Computer Applications in Nuclear Particle and Plasma Physics. 11th IEEE NPSS Real Time Conference. Conference Record (Cat. No.99EX295)*. Sante Fe, NM, USA: IEEE, 1999, pp. 346–350.

[27] S. R. Qasim *et al.*, "Learning representations of irregular particle-detector geometry with distance-weighted graph networks," *Eur. Phys. J. C*, vol. 79, no. 7, 1 2019.

[28] J. Kieseler, "Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data," *Eur. Phys. J. C*, vol. 80, no. 9, p. 886, 2020.

[29] AMD, "Vitis Unified Software Platform," https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vitis.html, 2025, version 2024.2, accessed 2025-10-28.

[30] ——, "Vivado Design Suite," https://www.amd.com/de/products/software/adaptive-socs-and-fpgas/vivado.html, 2025, version 2024.2, accessed 2025-10-28.

[31] C. N. Coelho *et al.*, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Machine Intelligence*, vol. 3, no. 8, pp. 675–686, Aug. 2021. [Online]. Available: https://www.nature.com/articles/s42256-021-00356-5

[32] I. Haide *et al.*, "Code for the GNN-ETM Training and Evaluation," https://github.com/ihaide/gnnetm-software, 2026.

[33] M. Neu *et al.*, "Code for the Quantized GravNet Implementation," https://github.com/ihaide/qgravnet, 2026.

[34] M. Neu and I. Haide, "Custom QKeras Fork," https://github.com/ihaide/qkeras, 2026.

[35] S. Gupta *et al.* Deep Learning with Limited Numerical Precision. [Online]. Available: http://arxiv.org/abs/1502.02551

[36] Z.-G. Liu and M. Mattina. Learning low-precision neural networks without Straight-Through Estimator(STE). [Online]. Available: http://arxiv.org/abs/1903.01061

[37] AMD, "ModelSim HDL simulator," https://eda.sw.siemens.com/en-US/ic/modelsim/, 2025, version 2023.4, accessed 2025-05-13.

[38] M. Shankar *et al.*, "The EPICS Archiver Appliance," *Proceedings of the 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems*, vol. ICALEPCS2015, pp. 4 pages, 0.753 MB, 2015.