Proceedings of the CTD 2022
PROC-CTD2022-18
September 12, 2022

Track Finding and Neural Network-Based Primary Vertex Reconstruction with FPGAs for the Upgrade of the CMS Level-1 Trigger System

Christopher Brown on behalf of the CMS Collaboration

Department of Physics
Imperial College London, UK

ABSTRACT

The CMS experiment will be upgraded to take advantage of the rich and ambitious physics opportunities provided by the High Luminosity LHC. Part of this upgrade will see the first level (Level-1) trigger use charged particle tracks from the full outer silicon tracker as an input for the first time. The reconstruction of these tracks begins with on-detector hit suppression, identifying hits (stubs) from charged particles with transverse momentum (p_T) > 2 GeV within the tracker modules themselves. This reduces the hit rate by one order of magnitude with 15,000 stubs being produced per bunch crossing. Dedicated off-detector electronics using high performance FPGAs are used to find track candidates at 40 MHz using a road-search based pattern matching step. These are then passed to a combinatorial Kalman filter that performs the track fit for O(200) tracks. This overall track finding algorithm is described, as is the ongoing work developing the track fitting firmware and a boosted decision tree approach to track quality estimation. The tracks are used in a variety of ways in downstream algorithms, in particular primary vertex finding in order to identify the hard scatter in an event and separate the primary interaction from an additional 200 simultaneous interactions. A novel approach to regress the primary vertex position and to reject tracks from additional soft interactions uses a lightweight 1000 parameter end-to-end neural network. This neural network possesses simultaneous knowledge of all stages in the reconstruction chain, which allows for end-to-end optimisation. The improved performance of this network versus a baseline approach in the primary vertex regression and track-to-vertex classification is shown. A quantised and pruned version of the neural network has been deployed on an FPGA to match the stringent timing and computing requirements of the Level-1 Trigger. Finally, integration tests between the track finder and vertexing firmware are shown using prototype hardware

PRESENTED AT

Connecting the Dots Workshop (CTD 2022)
May 31 - June 2, 2022

1 Introduction

The Compact Muon Soleniod (CMS) experiment [1] is one of two general purpose detectors situated around the Large Hadron Collider (LHC). The LHC provides CMS with a 40 MHz bunch crossing rate and an average of 25 simultaneous proton-proton interactions per bunch crossing (pileup). At this bunch crossing rate the amount and bandwidth of data produced by CMS is too large to process and store. Therefore a two-stage trigger [2] is used to select events based on their physics potential. The first hardware-based trigger, Level-1, was originally designed to make decisions on lower granularity calorimeter and muon system data with a fixed latency < 4 μs and output rate of 100 kHz. This is followed by a CPU farm high level trigger with an output rate of 1 kHz for storage and offline analysis.

By 2029 the LHC will have completed its High Luminosity (HL) [3] upgrades which will increase its instantaneous luminosity by three times, this will provide 3000 fb^{-1} of integrated luminosity for the general purpose LHC experiments by the end of the 2030s. While this increased luminosity aids searches for new rare process physics and precision standard model measurements it poses a large challenge to the current detectors and triggering systems with pileup (PU) increasing to a maximum of 200.

To maintain physics sensitivity with the HL-LHC and exploit advancements in both detector and computing hardware the CMS experiment is being upgraded [4], including the trigger system [5]. The high pileup would see current Level-1 accept rates increase to 4MHz, so in order to reduce this to a new rate of 750 kHz, silicon tracker tracks will be used at Level-1 for the first time. This necessitates a new outer tracker for CMS which will stream data to a new all-FPGA trigger system. The use of these tracker tracks also allows for the identification of the hard scatter, or primary vertex, in an event leading to the separation of the primary interaction from additional pileup interactions reducing the impact of background pileup on downstream triggering algorithms and maintaining controllable trigger rates.

2 The CMS Phase-2 Upgrade

The Phase-2 upgrade of CMS is a wide program of replacements and upgrades to the detector. Including a replacement of the endcap calorimetery to give high-granularity 4D shower reconstruction, higher η coverage in the muon systems and upgraded electronics throughout the detector subsystems. The all-silicon tracker will be replaced with a higher radiation tolerant, higher η coverage tracker [6], schematically shown in Fig. 1. The outer tracker will be used as input to the Level-1 trigger as the inner tracker data bandwidth is too high and the data too difficult to extract to process in the fixed trigger latency. In order to reduce the data bandwidth to within manageable levels, new "p_T" modules [7] for the outer tracker have also been designed consisting of two closely spaced layers (1.6 - 4.0 mm) of silicon and on-module correlation logic giving a tuneable p_T cut on charged particle tracks of 2-3 GeV. Due to this on-detector p_T cut, the number of pairs of hits or 'stubs' being passed to the Level-1 trigger (at the full 40 MHz) is reduced by a factor of 10 compared to no p_T cut. This will still see $\approx$ 15,000 stubs per bunch crossing needing to be processed, matched to charged particle trajectories and fitted all within 4 μs to meet the latency requirements of the Level-1 trigger.

The Phase-2 upgrade of the Level-1 trigger will see the total rate being increased to 750 kHz with a longer latency per event of < 12.5 μs due to improvements to on-detector buffers. This, combined with the use of state-of-the-art FPGAs, will see more complex algorithms being used at Level-1. With access to tracker tracks it will become possible to perform particle flow reconstruction which allows the Level-1 trigger to reconstruct full events with all detector subsystems. However, the use of these algorithms is highly dependent on the ability of the trigger to reduce the impact of pileup. The most important tool for reducing this background pileup is the reconstruction of the primary vertex (PV) in an event. Tracks will be used to locate the hard scatter in an event which other trigger objects can be associated to, such as the tracks themselves or energy deposits in the calorimeters, thus allowing for the vetoing of additional energy in the detector from pileup. Another key part of the upgraded trigger is the use of machine learning techniques, these are particularly suitable for finding optimal solutions on reduced inputs that the environment of the trigger provides. This allows for a reduction in the lengthy algorithmic and firmware development process of more traditional approaches.

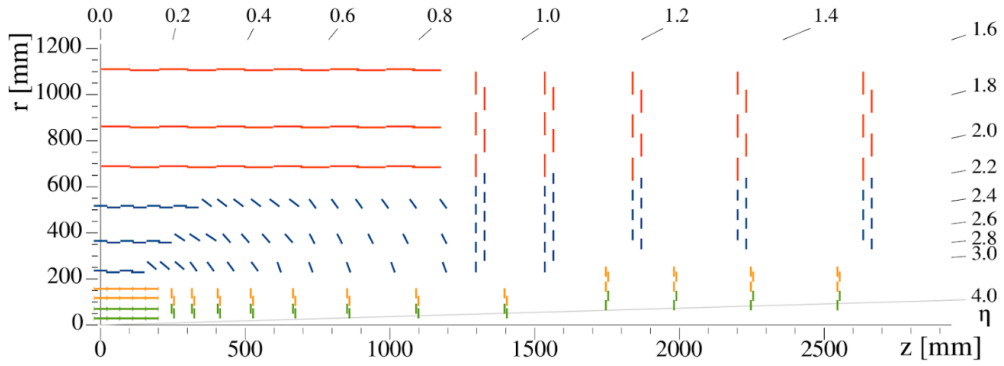

Figure 1: A slice in r-z of the Phase-2 CMS tracker. In green and yellow the inner tracker with increased η coverage up to 3.8 and in red and blue the new outer tracker consisting of p_T modules with a tuneable p_T cut and an η coverage < 2.4.

3 Track Finding

In order to reconstruct tracks from collections of hits or 'stubs' from the detectors p_T modules a two-part 'hybrid' algorithm formed of two previously demonstrated algorithms [8],[9] will be used. The first part is a tracklet pattern recognition step used to form track candidates from pairs of seed stubs, and secondly, a combinatorial Kalman Filter (KF) fits the tracks and calculates the helix parameters. The algorithm follows these main steps:

- Tracklet seeds are formed from stub doublets in specific combinations of adjacent layers and disks.
- Tracklet helix parameters are calculated from the seed using the constraint that the track originated from the interaction region.
- The tracklet is projected to the remaining layers and disks and stubs are matched to it within a set window.
- Duplicate tracklets are merged which share three or more layers and disks.
- A combinatorial Kalman Filter seeded from the tracklet candidate has associated stubs iteratively added to it, updating the KF state and covariance matrices. This selects the best stub combination for the track and calculates the p_T, η, ϕ and z_0 (the track's distance from the beamspot, along the beam line).

This track finding algorithm performs well with a $> 95\%$ efficiency for all tracks $|\eta| < 2.4$, $p_T > 2$ GeV. While this has been successfully demonstrated in emulation the firmware development is ongoing with key sections of the track finding chain being tested in hardware. Due to the geometry of the CMS tracker a $\cosh(|\eta|)$ relationship between transverse impact parameter (z_0) resolution and η can be see in Fig. 2, this relationship is important when designing vertex finding and association algorithms where the resolution of the tracks themselves dominate the achievable resolution of a vertex. Also shown is the fraction of non-genuine (or fake) tracks (those not associated to real tracking particles as defined in Monte Carlo simulations based on stub matching) as a function of p_T. These fakes are produced when combinations of stubs that do not originate from a real charged particle are combined by the track finding. As these represent a large fraction ($\approx 10\%$) of tracks at higher p_T strategies for reducing the number of fakes is important to not influence downstream trigger performance where algorithms typically rely on high p_T signatures.

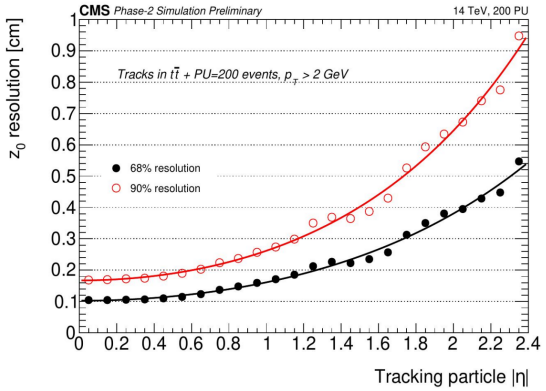

Figure 2: Track z_0 residuals are calculated in a number of η bins and the 68% and 90% quantiles are shown for each bin, also shown is the expected $\cosh(|\eta|)$ fit.

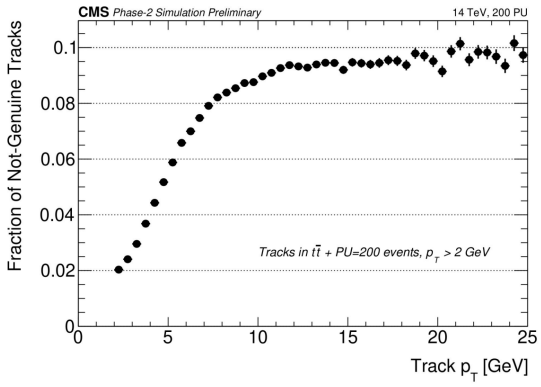

Figure 3: Non-genuine fraction of reconstructed tracks (number of non-genuine tracks / total number of reconstructed tracks) for multiple p_T bins showing a relative increase at higher p_Ts.

3.1 Track Quality

In order to reduce the number of non-genuine tracks the quality of the track fit can be used. The KF produces uncertainties in the $r-\phi$ and $r-z$ planes that can be combined to give a $\chi^2_{r\phi}$ and χ^2_{rz} fit as well as an additional χ^2_{bend} which depends on the detector geometry. Together, these fitting parameters, as well as additional constraints on the number of stubs a track has been formed from, can be used to impose quality criteria on tracks. These have been shown to be useful in highly p_T sensitive calculations such as track-based E_T^{miss} where E_T^{miss} trigger thresholds can be reduced by up to 50% by imposing strict quality cuts on tracks.

While a single working point per individual algorithm performs well it is more useful for downstream users to have flexible working points based on the overall 'quality' of a track which can be derived from various track features. For this reason, Boosted Decision Trees (BDTs) were trained using track helix and fit parameters to identify real versus fake tracks in a binary classification task [10]. These BDTs outperform the discrimination power of a single tuned cut as shown in Fig. 4 while also providing a flexible threshold for multiple downstream algorithms.

These BDTs were designed to be lightweight with a tree depth of 3 and with 60 iterations due to firmware constraints in the Level-1 trigger. Custom HDL was used to give BDTs with $< 1\%$ resource usage on the target FPGA (Xilinx UltraScale+ VU13P) used for the track finding algorithm and with a total latency of 33 ns running at 250 MHz making them a viable option for track quality assessment.

4 Vertex Finding

The primary vertex is the location along the beam line of the hard proton-proton scatter in an event. Offline, it is estimated as the reconstructed vertex with the highest track $\sum p_T^2$ [11]. It is important to locate this vertex at Level-1 as it reduces the impact of pileup on downstream algorithms. When other trigger objects are associated to the vertex the trigger can have cleaner energy sums, better lepton isolation and a reduction in the number of objects being passed to downstream algorithms making them computationally feasible in the strict latency requirements.

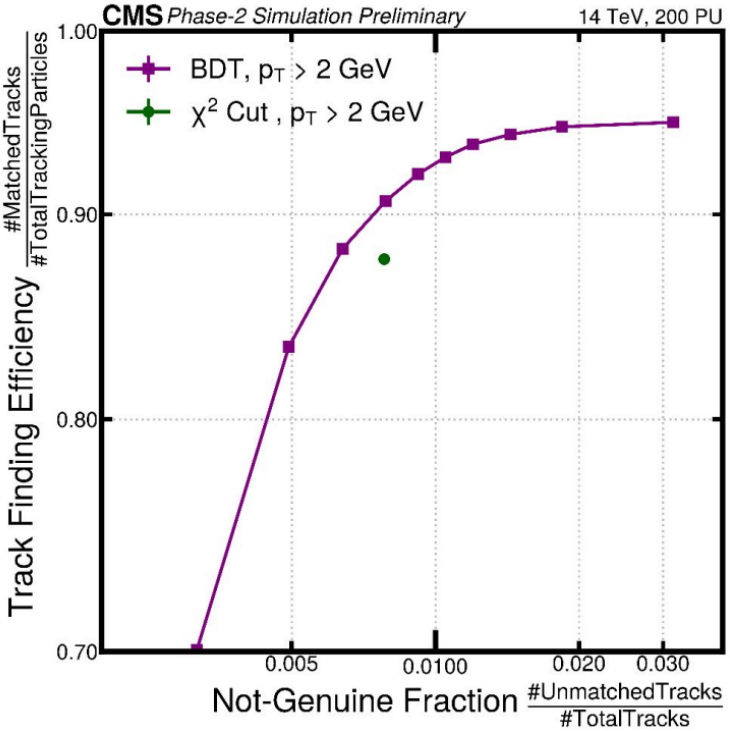

Figure 4: Track finding efficiency versus non-genuine fraction for the track-based E_T^{miss} cut on χ^2 and the BDT approach for track quality estimation.

4.1 Baseline Approach

The baseline approach to vertex finding [5] creates a histogram in z_0 of all tracks weighted by their p_T. This histograms consists of 256 bins in a z_0 range of -15,15 cm. A three bin window is then passed across the histogram in order to find the three consecutive bins with the highest total weight. The centre of the middle of these three bins is returned as the primary vertex. This method has been implemented in firmware and run in hardware with a latency of 81 ns at 360 MHz. There are some key issues with this method; firstly, there is no accounting for the high p_T fakes as shown in Fig. 3. These can appear with other pileup tracks as high p_T clusters in the histogram returning a fake vertex. Secondly, there is no correction for the degradation in z_0 resolution in high η tracks as shown in Fig. 2 which leads to a worse resolution of the PV.

However, to associate tracks to this vertex and filter out PU tracks, an η dependent window in z_0 is used. This is reasonably effective with a 91% rate of correctly assigning PV tracks to the vertex and a 10% rate of assigning fake or PU tracks to the vertex. This is a fast method but does not take into account the quality of the tracks which can degrade the performance as well as other track features that could better discriminate pileup from PV tracks.

5 End-to-End Vertex Finding

By taking into account multiple track features when weighting the baseline approach's histogram it is possible to improve the performance. Additional weightings such as $\frac{1}{\eta^2}$ (as an approximation of the fit in Fig. 2) can be shown to give a 10% improvement in the core of the vertex error residual. Cuts on the output of the track quality BDT shown in Section 3.1 can give a 40% improvement in the tails of the residual. The combination of these features is non-trivial so an iteratively learnt approach using a dense neural network was employed. Additionally, it was found that allowing for a more complex pattern recognition of the peak in the weighted

histogram improved performance when replacing the baseline's three bin window with a 1D convolutional neural network. Finally, the track-to-vertex association also benefits from more complex track features aside from η and distance to the vertex so another dense neural network was used for this binary classification task. While the regression of the primary vertex is an important target for developing this algorithm, the association of tracks to this vertex is ultimately the most important information being passed to downstream physics algorithms. For this reason, an end-to-end approach was adopted which uses both an event-level primary vertex regression and track-level track-to-vertex association that are learnt simultaneously allowing the histogram weighting and peak pattern finder to be optimised for the track-to-vertex association as well.

5.1 Architecture and Back Propagation

The network has three main parts as shown in Fig. 5 with the weight network feeding into a differentiable histogram leading to the 1D convolutional pattern recognition and then outputting to a differentiable ArgMax to regress the PV position with a pseudo-Huber loss function. The difference in position between vertex and track z_0 along with other features is then used to perform the track-to-vertex association with a binary cross entropy loss function.

In order to perform back propagation and allow the weight network to learn from both the PV regression and track-to-vertex association losses, custom histogram and ArgMax layers were written. The histogram with bin h_i is filled with track z_0, weighted by weight w which is a learnt function of track features:

$$h_i = \sum_{j=0}^{tracks} \delta(j \in \text{bin } i)\, w(p_{T,j}, \eta_j, MVA_j) \tag{1}$$

resulting in the following gradients

$$\frac{\partial h_i}{\partial z_0} = 0 \quad \text{and} \quad \frac{\partial h_i}{\partial w} = \sum_{j=0}^{tracks} \delta(j \in \text{bin } i) \tag{2}$$

which are implemented as custom TensorFlow operations.

The ArgMax layer has an input of a vector x (in this case the convolved histogram) of N elements and is defined as:

$$\sum_{i=0}^{N} i \frac{e^{x_i/T}}{\sum_{j=0}^{N} e^{x_j/T}} \tag{3}$$

where T is a tuned hyperparameter of the network which allows this layer to return an approximate one-hot encoding of the convolved histogram.

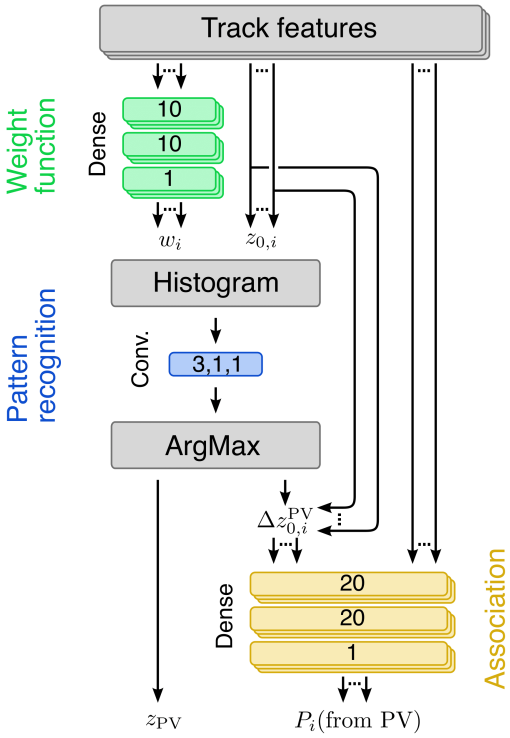

Figure 5: End-to-end network architecture showing the three distinct networks in colour as well as the position of the histogram and ArgMax layers.

5.2 Performance

This end-to-end approach outperforms the baseline approach. Firstly, it sees a 55% reduction in the tails of the residual due to better filtering of fake clusters caused by high p_T fake tracks, this is shown in Fig. 6. Secondly, in the track-to-vertex association the end-to-end approach better discriminates PV tracks from PU and fakes with a rate of 96% at the same 10% false positive rate as shown in the receiver operating characteristic curve in Fig. 7. It is also able to pass down a flexible threshold for downstream algorithms to use as opposed to a set working point given by a traditional cut-based approach.

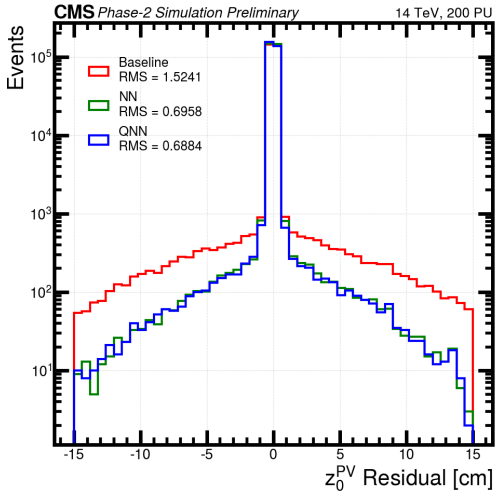

Figure 6: True PV - Reconstructed PV for the Baseline and NN approaches. NN refers to the floating point approach while QNN is the quantised approach described in Section 5.3

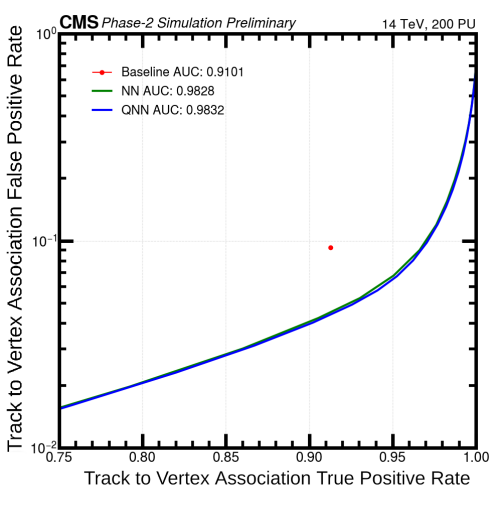

Figure 7: Receiver Operating Characteristic (ROC) curve for the Baseline and NN approaches to track-to-vertex association. Shown as true positive rate versus false positive rate.

5.3 Implementation

As this end-to-end approach uses the same building blocks as the baseline approach, much of the baseline approach's firmware can be reused and individual parts of the network can be inserted. For this reason the network is split into three individual parts, the weight, pattern and association networks, shown in Fig. 5. These are run through the hls4ml [12] tool that generates High Level Synthesis (HLS) firmware for each network which are then wired within the existing baseline approach.

Due to the overall parallelism of the system architecture of the Level-1 trigger the weight and association networks that operate at a track-level are to be implemented multiple times. This makes the small resource usage and low latency of these parts of the network critical. Two approaches were used in order to reduce the network size. Firstly, quantized aware training restricts the bitwidths of learnt weights and biases in the training phase using the QKeras package [13]. This allows fine tuned setting of fixed bitwidths for each layer thus reducing the total amount of FPGA resources needed per operation in the network while maintaining the performance of the network. Secondly, network pruning [14], the technique of iteratively removing close to zero weights over multiple training cycles, is used to reduce the total number of operations needed by the network, this is particularly useful for reducing the number of multiplications required at inference time.

The resource usage of a Xilinx UltraScale+ VU9P running at 360 MHz is shown in Table 1. Both floating point and a quantised and pruned version of the network are shown demonstrating the large reduction in resource usage especially in reducing the Digitial Signal Processor (DSP) usage which is a limiting factor in these FPGAs. The performance of these two version of the network are shown in Figs. 6 and 7 where little difference is seen between the NN and QNN.

6 Integration Testing

In order to validate the functioning of the upgraded Level-1 trigger, testing of algorithms on their target hardware is required. Custom carrier boards for specific FPGAs were designed for different parts of the trigger with the track finding being performed on the Apollo [16] board and much of the Level-1 trigger using either the Serenity [15] or APx boards. Both the boards and algorithms have reached a level of maturity that

Network	Latency (ns)	Initiation Interval (ns)	LUTs %	DSPs %	BRAMs %	FFs %
NN (Q) Weights	22 (14)	2.7 (2.7)	2.52 (0.90)	19.98 (0.00)	0.00 (0.00)	0.72 (0.36)
NN (Q) Pattern	58 (42)	51 (35)	4.27 (4.43)	3.74 (0.00)	5.28 (5.28)	3.22 (3.15)
NN (Q) Assoc.	30 (25)	2.7 (2.7)	0.54 (7.92)	107.64 (0.54)	0.00 (0.00)	2.70 (2.34)
Baseline	44	2.7	2.40	0.00	1.90	1.40

Table 1: Resource usage and latencies of a Xilinx VU9P running at 360 MHz for the floating point Neural Network (labelled NN) and the quantised and pruned version (labelled Q and in parenthesis in the table). Shown are total resource usage for all replications of the weight and association networks, corresponding to the number of parallel track streams used in the Level-1 trigger (the pattern network is only implemented once). Also included is the baseline approach, the NN approaches are additional to these resources and latency as they use existing parts of the baseline firmware. These resource usages are estimates from a Vivado synthesis of the networks and the latencies are from a C-Simulation of the RTL logic using ModelSim.

allows for system integration testing to begin. Single board tests of parts of the track finding chain have been completed with sub sections of the tracklet algorithm being interfaced with the Kalman Filter and compared to software emulation. Single board tests have also been performed with both the baseline and end-to-end approaches to vertex finding. Another part of these tests is the integration of subsystems. High speed 25 GB/s) fibre optic connections are used to transmit the output of one algorithm to another running on separate hardware. This has led to a demonstration of the track finding output step outputting tracks to the vertex finding and a vertex being recorded in output buffers, an algorithmic overview is shown in Fig. 8 and the corresponding hardware setup for this is shown in Fig. 9. This not only allows for the hardware boards themselves to be tested but also the performance of algorithms in more 'real-world' scenarios with data being received over fibre optics.

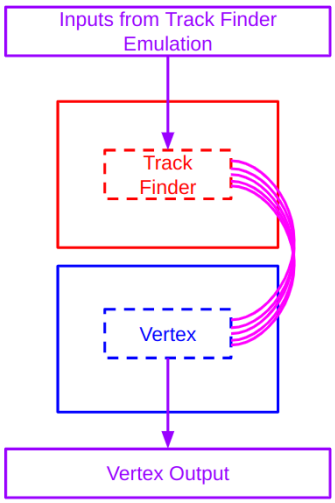

Figure 8: Overview of algorithms and links used for board-to-board integration testing of the track finder to vertex finder.

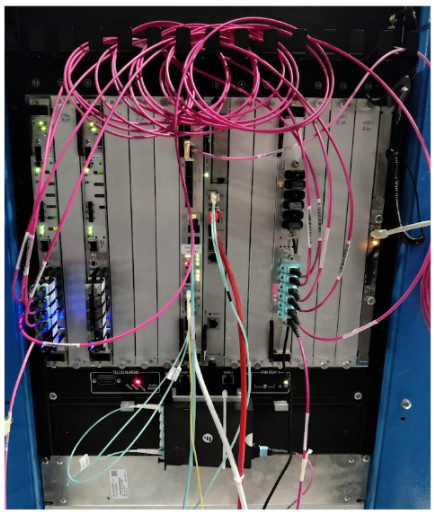

Figure 9: Hardware demonstrator of the track finding output running on the board on the left outputting tracks over five fibre optic links to the vertex finding running on the board on the right where the vertex is calculated and stored in buffers.

7 Conclusion

The HL-LHC upgrade program will expand the physics reach of the CMS experiment and with key upgrades to both the detector and trigger system it will maintain and expand its physics capabilities despite the high pileup conditions. An improved outer tracker using specialised modules with a tuneable on-detector p_T cut will allow tracker tracks to be reconstructed at the Level-1 trigger for the first time. A tracklet pattern recognition followed by combinatorial Kalman Filter will be used to reconstruct tracks within 4 μs and an additional BDT for track quality will be used to reduce the impact of fake tracks in downstream algorithms. Global event variables such as the primary vertex will also be reconstructed in the Level-1 trigger reducing the impact of pileup on downstream algorithms and improving trigger performance. A novel approach to the regression of this vertex and track-to-vertex association using an end-to-end neural network is shown to outperform the baseline approach while maintaining low latency and low resource usage. Finally, the demonstration of these systems is underway with key integration tests between the track finder and vertex finding being completed.

References

[1] The CMS Collaboration, "The CMS Experiment at the CERN LHC," JINST **3** (2008), S08004 doi:10.1088/1748-0221/3/08/S08004

[2] The CMS Collaboration, "The CMS trigger system," JINST **12** (2017) no.01, P01020 doi:10.1088/1748-0221/12/01/P01020

[3] G. Apollinari, I. Béjar Alonso, O. Brüning, M. Lamont and L. Rossi, "High-Luminosity Large Hadron Collider (HL-LHC) : Preliminary Design Report," doi:10.5170/CERN-2015-005

[4] The CMS Collaboration, "Technical Proposal for the Phase-II Upgrade of the CMS Detector," CERN-LHCC-2015-010. https://inspirehep.net/literature/1614097

[5] The CMS Collaboration, "The Phase-2 Upgrade of the CMS Level-1 Trigger," CERN-LHCC-2020-004. https://inspirehep.net/literature/1819968

[6] The CMS Collaboration, "The Phase-2 Upgrade of the CMS Tracker," doi:10.17181/CERN.QZ28.FLHW

[7] G. Hall, M. Raymond and A. Rose, "2-D PT module concept for the SLHC CMS tracker," JINST **5** (2010), C07012 doi:10.1088/1748-0221/5/07/C07012

[8] I. Tomalin *et al.* "An FPGA based track finder for the L1 trigger of the CMS experiment at the High Luminosity LHC," JINST **12** (2017), P12019 doi:10.1088/1748-0221/12/12/P12019

[9] E. Bartz *et al.* "FPGA-based tracking for the CMS Level-1 trigger using the tracklet algorithm," JINST **15** (2020) no.06, P06024 doi:10.1088/1748-0221/15/06/P06024

[10] S. P. Summers, "Application of FPGAs to triggering in high energy physics," doi:10.25560/66689

[11] The CMS Collaboration, "Description and performance of track and primary-vertex reconstruction with the CMS tracker," JINST **9** (2014) no.10, P10009 doi:10.1088/1748-0221/9/10/P10009

[12] J. Duarte *et al.* "Fast inference of deep neural networks in FPGAs for particle physics," JINST **13** (2018) no.07, P07027 doi:10.1088/1748-0221/13/07/P07027

[13] C. N. Coelho *et al.* "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," Nature Mach. Intell. **3** (2021), 675-686 doi:10.1038/s42256-021-00356-5

[14] H. S. Pool, J. Tran, W. J. Dally, 'Learning both Weights and Connections for Efficient Neural Networks," **NIPS'15** (2015), 1135 doi:10.5555/2969239.2969366

[15] A. Rose *et al.* "Serenity: An ATCA prototyping platform for CMS Phase-2," PoS **TWEPP2018** (2019), 115 doi:10.22323/1.343.0115

[16] A. Albert *et al.* "The Apollo ATCA Platform," PoS **TWEPP2019** (2020), 120 doi:10.22323/1.370.0120