Accelerating Track Reconstruction at CMS Using FPGA Co-Processors

Professor Isobel Ojalvo, Princeton University

1. Overview and Relevant Experience

As a member of the CMS experiment [1], I propose to continue work developing both traditional and Machine Learning track based reconstruction on FPGA co-processors. While acceleration using FPGA co-processors is a new endeavor for my group (as well as for the CMS collaboration) I have contributed to and served in leadership positions in the CMS collaboration first as the developer of high speed algorithms for the the Level-1 trigger, put into place in 2015, then as the convener of the tau trigger development group (2014-2015) and later as the convener of the tau physics object group (2015-2018). Currently, I serve as the US CMS Level-1 Trigger Operations deputy. I am working with my postdoc Savannah Thais and graduate student Gage DeZoort on the development of low latency track reconstruction algorithms that can be run either online (in an upgraded High Level Trigger (HLT)) or offline.

This proposal is to request a continuation of 50% support for Princeton postdoc Savannah Thais in our effort to explore and implement methods for accelerating track reconstruction. Over the past year we have made substantial progress towards implementing GNN-based track reconstruction on FPGAs. Thais and DeZoort have demonstrated a proof-of-concept implementation using an Intel Stratix FPGA co-processor and have thus far measured resource utilization as a function of graph size and bit precision. This compliments work done by our collaborators implementing the GNNs fully on an FPGA using the hls4ml framework and Xilinx FPGAs. Together, we presented this work in a paper published at the NeurIPS Machine Learning and Physical Sciences workshop and presentations at the Inter-experiment Machine Learning Workshop and the FastML workshop. We are currently studying improvements on the co-processor implementation by examining the data transfer bottleneck from CPU to FPGA, optimizing the matrix multiplication kernels, and implementing the graph construction algorithm in parallel on FPGAs. In a recent paper submitted to vCHEP we demonstrated that the Interaction Network GNN architecture that we have been studying for the FPGA implementation can be substantially reduced in terms of the number of learnable parameters so it can be more effectively accelerated. Additionally, Thais has been serving as the L3 co-convener for the CMS machine learning group since June 2020.

We will continue to develop and extend this research to optimize for graph sizes and architectures that will be needed in the high-occupancy HL-LHC environment.

2. Proposed Research

The unambiguous identification of an event produced at the LHC which contains a clear signature of Dark Matter, or perhaps a SuperSymmetric particle would completely change our understanding of the physical universe. However, the difficulty surrounding sifting through large data sets to identify a tiny fraction of interesting events will only grow in the next ten years as data rates are expected to increase dramatically while physicists search

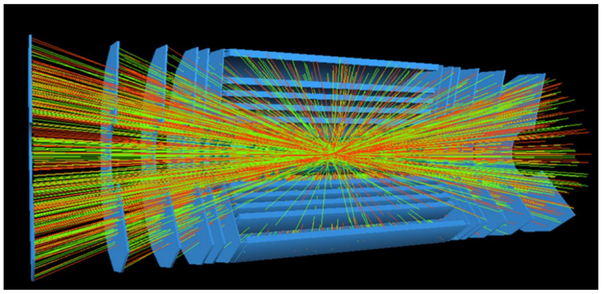

Figure 1: The CMS detector has over 100 million channels. As the number of simultaneous collisions increase from an average of 75 in 2018 to 200 in the 2026 and beyond, reconstruction of charged particle tracks will become an increasingly complex problem.

for increasingly rare events and interactions. On the LHC ring, in the ATLAS and CMS detectors collisions between particle bunches occur 40 million times per second. During LHC Run 2 (2015-2018) each bunch collision resulted in an average of 75 individual proton-proton interactions. In both ATLAS and CMS, close to 150 million channels, as illustrated in Figure 1, need to be read out, creating a data rate of hundreds of terabytes per second [2, 3]. This rate must be reduced by a factor of 10^6 within a few microseconds and then by a further factor of 100 within a few milliseconds in order to be stored for later analysis.

Track finding and fitting is one of the most computationally challenging problems for event reconstruction in particle physics. The CMS experiment has a specialized tracking system that consists of multiple layers of highly granular sensors which are designed to measure both the position and the curvature of the charged particle. The many layers and multiple possible paths for each charged particle turns the track reconstruction into a highly complex ”connecting the dots” problem, as illustrated in Figure 2. Currently at CMS, track reconstruction relies on the Kalman filter method. The filter proceeds iteratively from the seed layer, starting from a coarse estimate of the track parameters provided by the seed, and including the information of the successive detection layers one by one. On each layer, i.e. with every new measurement, the track parameters are known with a better precision, up to the last point, where they include the full tracker information. The Kalman Filter method is highly tuned for physics performance in today's LHC conditions but the computation time has been shown to scale quadratically or worse with detector occupancy [4].

Advanced Machine Learning (ML) poses as an exciting solution to the issue of algorithm scalability as many ML algorithms are expected to scale linearly with detector occupancy [5]. In particular, the Graph Neural Network (GNN) is an interesting option given its ability to learn effective representations of high-dimensional data through training and to model complex dynamics through computationally regular transformation techniques. GNNs were first introduced in [6] and has been applied to a growing variety of problems including social networks, and 3D Shape analysis [7]. They have already been studied for particle tracking applications in [5] and for the problem of particle event classification in [8, 9]. The graph is constructed so that the nodes are the hits recorded by the detector and the edges are

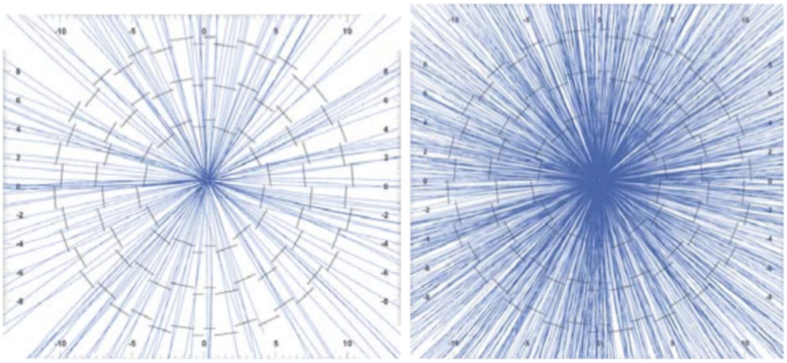

Figure 2: The detector's layers are linked together to infer the trajectory of individual charged particles in a complex "connect the dots" puzzle. This problem will grow even more complex in the next ten years when the number of proton-proton interactions increases from 20 in 2015 conditions (left), to 140 (right) which is expected at the HL-LHC.

connections of the hits between adjacent detector layers that pass a pre-defined filter which is tuned to be efficient for tracks resulting from high transverse momentum particles.

Figure 3, produced by Princeton postdoc Savannah Thais, graduate student Gage DeZoort, and summer fellow Aneesh Heintz show a characteristic graph of particle tracks with $p_T > 2$ GeV for one event (top) and the GNN architecture used for the implementation below. Following the approach of Ref. [10, 11], we define a "segment classifier" using an IN model [12, 13] to learn which edges connect hits belonging to the same track. Relative to Ref. [11], the model architecture is simplified for the more limited task and the FPGA implementations. The implementation of the Interaction Network (IN) adopts a CPU-plus-FPGA coprocessing approach where the host program on the CPU manages the application, and all computational operations are accelerated using dedicated kernels deployed on the FPGA that capitalize on the device's hardware architecture to parallelize operations. The matrix multiplication kernel is repeatedly executed during a forward pass of the network and leverages the FPGA architecture for an efficient data-parallel implementation. This kernel uses both 2D local memory tiling and 2D register blocking to reduce the redundancy and latency of reading from globally shared off-chip memory Because the input graph sizes to the network changes per event, the matrix multiplication kernels pad each matrix before computing the result. Throughout the forward pass, several optimizations are made to speed up computation. All loops iterations executed in the OpenCL kernels are "unrolled" to run in parallel, which decreases the latency at the cost of increased hardware resource consumption. The results of this study are summarized in Ref [14]. This work is the first step towards actualizing a fully functioning GNN track-based reconstruction architecture implemented on FPGA co-processors. This work needs to be extended to larger graph sizes to allow for track reconstruction below 2 GeV for events with high occupancy (such as those at the HL-LHC). We are also planning to test a variety of architectures to optimize for the CPU + FPGA based architecture, including, one-shot tracking, where by full inference and reconstruction

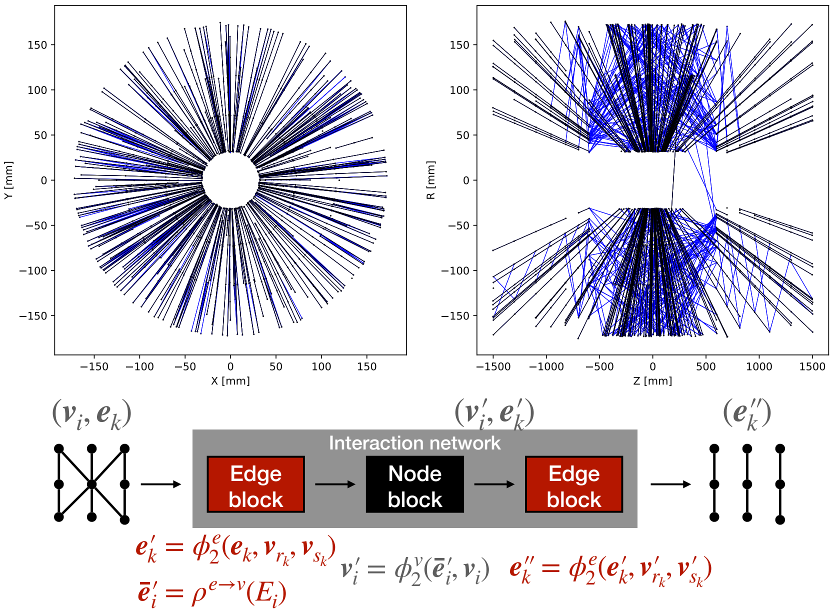

Figure 3: A characteristic graph of particles with $p_T > 2$ GeV for one event in x–y view (upper left) and r–z view (upper right). The black edges correspond to true track segments, while the blue edges are spurious. The GNN architecture used for the implementation is shown below.

is performed in one-shot wit no need to join hits using a linking algorithm, reducing the need for multiple data transfers onto and off of the FPGA. We will also explore divisions of the tracking geometry in order to reduce the overall graph sizes. Upon successful completion of this project, we will complete a GNN based approach to track reconstruction using an FPGA co-processor architecture with the CMS, which will help maximize the HL-LHC physics reach.

3. Proposed Timeline

We have already started on this project in January 2020. We have performed latency measurements on an Intel Stratix FPGA co-processor and developed an IN for GNN inference. We plan to complete studies with large graphs that allow for track p_T down to 0.5 GeV by July of 2021. concurrently we will be developing a single-shot GNN track reconstruction algorithm for use on an Intel Stratix FPGA. The development of the overall training and inference structure we expect will take until August of 2021. Then we will spend until the following spring to complete the implementation on the FPGA co-processor.

References

[1] S. Chatrchyan et al. "The CMS experiment at the CERN LHC". In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[2] Johannes Albrecht et al. "A Roadmap for HEP Software and Computing R&D for the 2020s". In: *Comput. Softw. Big Sci.* 3.1 (2019), p. 7. DOI: 10.1007/s41781-018-0018-8. arXiv: 1712.06982 [physics.comp-ph].

[3] A. Klimentov et al. "BigData and computing challenges in high energy and nuclear physics". English. In: *Journal of Instrumentation* 12.6 (June 2017). ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/06/C06044.

[4] Giuseppe Cerati et al. "Traditional Tracking with Kalman Filter on Parallel Architectures". In: *Journal of Physics: Conference Series* 608 (May 2015), p. 012057. DOI: 10.1088/1742-6596/608/1/012057. URL: https://doi.org/10.1088%2F1742-6596%2F608%2F1%2F012057.

[5] Steven Farrell et al. *Novel deep learning methods for track reconstruction.* 2018. arXiv: 1810.06111 [hep-ex].

[6] F. Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (Jan. 2009), pp. 61–80. ISSN: 1941-0093.

[7] Jie Zhou et al. *Graph Neural Networks: A Review of Methods and Applications.* 2018. arXiv: 1812.08434 [cs.LG].

[8] Jesus Arjona Martinez et al. *Pileup mitigation at the Large Hadron Collider with Graph Neural Networks.* 2018. arXiv: 1810.07988 [hep-ph].

[9] Huilin Qu and Loukas Gouskos. *ParticleNet: Jet Tagging via Particle Clouds.* 2019. arXiv: 1902.08570 [hep-ph].

[10] Steven Farrell et al. "Novel deep learning methods for track reconstruction". In: *4th International Workshop Connecting The Dots 2018.* 2018. arXiv: 1810.06111 [hep-ex].

[11] Xiangyang Ju et al. "Graph Neural Networks for Particle Reconstruction in High Energy Physics Detectors". In: *Machine Learning and the Physical Sciences Workshop at the 33rd Annual Conference on Neural Information Processing Systems.* 2019. arXiv: 2003.11603 [physics.ins-det]. URL: https://ml4physicalsciences.github.io/files/NeurIPS_ML4PS_2019_83.pdf.

[12] Peter W. Battaglia et al. "Interaction Networks for Learning about Objects, Relations and Physics". In: *Advances in Neural Information Processing Systems.* Vol. 29. 2016, p. 4502. arXiv: 1612.00222 [cs.AI].

[13] Peter W. Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: (2018). arXiv: 1806.01261 [cs.LG].

[14] A. Heintz et al. "Accelerated Charged Particle Tracking with Graph Neural Networks on FPGAs". In: (2020). arXiv: 2012.01563.