Scaling up implementations of GNNs with FPGA co-processors for charged particle track reconstruction

Caitlin Patterson, *The Ohio State University* (patterson.988@osu.edu)

PI: Savannah Thais, *Princeton University*

November 5, 2020

1 Proposed Project

Charged particle tracking is an important part of characterizing particles produced in colliders. Tracking algorithms are very computationally expensive and scale poorly with the number of hits [1]. The upcomding run of high-luminosity the LHC will feature an increase in collision rate, leading to more data production than ever before. It is therefore advantageous to implement these algorithms using FPGAs, which have lower latency and higher energy efficiency than CPUs. Graph neural networks (GNN) can be used for segment classification of tracks [2] and are well suited for implementation on FPGAs.

This project proposes building on the progress made by Aneesh Heintz using FPGAs to co-procress GNNs based on the interaction network architecture. This work will be done in collaboration with Dr. Savannah Thais and Dr. Isobel Ojalvo's group. Thus far, there has been success implementing small graph networks by accelerating execution of OpenCL kernels on an FPGA. Kernels are parallelized functions in OpenCL, a C-based framework for parallel computing. It remains to scale up these networks to larger graph sizes. Compared to other implementations of GNNs on FPGAs, the OpenCL implementation scales up more easily but has longer latency. Therefore, an initial investigation would involve exploring the latency of the existing small graph networks. During this, small graph networks will be implemented using various kernels for matrix multiplication in OpenCL. Then, the latency of these kernels will be compared, with scalability in mind. I will also determine whether data transfer rates between the FPGA and CPU contribute to the latency of GNN implementations. Dr. Ojalvo's group and Dr. Thais have been collaborating with many researchers, some of whom are investigating complementary implementations of GNNs on FPGAs which may have latency advantages. Throughout this project, we will make detailed comparisons of the two implementations.

The ultimate goal will be to scale up the implementation of the GNN. When doing this, findings about latency both in this project and prior to this proejct will be considered. If necessary, methods to decrease FPGA resource consumption will also be considered. Deliverables will include

- Scaling up of FPGA co-processed GNN for graphs corresponding to particles with transverse momentum $p_{\mathrm{T}} > 5$ GeV for applications in high energy physics experiments
- Better understanding of latency in OpenCL implementations

2 Student background

During the fellowship, I will be taking classes and working toward my graduation in May. As I am in my last semester, I require few credits to graduate and would take a reduced course load to accommodate the fellow position. I will work for 6 months at half-time in light of my classes. I will also be doing research in some capacity at The Ohio State University, where I am in an undergraduate. However, my research advisor has encouraged me to apply to this position and to focus my time on it as I am unable to go into lab in light of the COVID-19 pandemic.

Previously, I have done research characterizing time delays on FPGAs, toward the goal of realizing a time-to-digital converter on a FPGA. This work has also led to implications for the implementation of physically unclonable functions on FPGAs. My background with FPGAs and familiarity with C-based languages, including C, C++ and Verilog will help me complete the project within the timeline. In the proposed work, I would expand my skillset to include the implementation of neural networks on FPGAs while working on a project more directly related to current physics problems than my previous work. This will better prepare me to pursue a PhD in experimental physics, where I hope to continuing researching the application of computational techniques to physics problems involving data science or quantum computation.

3 Timeline

Month	*Activity*
1	Project setup. Become familiar with with OpenCL and existing graph networks. Perform pruning study on OpenCL graph network to compare timing performance with another complementary implementation of the GNN
2	Investigate impact of various OpenCL kernels on latency of small graph networks
3	Study data transfer rates between CPU and FPGA for various network sizes
4 - 5	Scale up existing GNNs using information on OpenCL kernels and the data transfer rate
6	Prepare findings, ensure work is well-organized for continuation

4 References

[1] A. Heinz & V. Razavimaleki et al., Accelerated charged particle tracking with graph neural networks on FPGAs (2020), *NeurIPS 2020.*

[2] S. Farrell et al., Novel deep learning methods for track reconstruction (2018), *4th International Workshop Connecting The Dots 2018.* arXiv:1810.06111