DISTANCE-WEIGHTED GRAPH NEURAL NETWORKS ON FPGAS FOR REAL-TIME PARTICLE RECONSTRUCTION AT THE LARGE HADRON COLLIDER

Gianluca Cerminara, Abhijay Gupta, Yutaro Iiyama, Jan Kieseler, Vladimir Loncar, Jennifer Ngadiuba, Maurizio Pierini, Marcel Rieger, Sioni Summers, Gerrit Van Onsem, Kinga Wozniak
European Organization for Nuclear Research (CERN)
CH-1211 Geneva 23, Switzerland

Giuseppe Di Guglielmo
Columbia University, New York, NY 10027, USA

Javier Duarte,
University of California San Diego, La Jolla, CA 92093, USA

Philip Harris, Dylan Rankin
Massachusetts Institute of Technology Cambridge, MA 02139, USA

Sergo Jindariani, Mia Liu, Kevin Pedro, Nhan Tran
Fermi National Accelerator Laboratory Batavia, IL 60510, USA

Edward Kreinar
HawkEye360 Herndon, VA 20170, USA

Zhenbin Wu
University of Illinois at Chicago Chicago, IL 60607, USA

January 8, 2020

ABSTRACT

Graph neural networks have been shown to achieve excellent performance for several crucial tasks in particle physics, such as charged particle tracking, jet tagging, and clustering. An important domain for the application of these networks is the Level-1, FGPA-based trigger, which has strict latency and resource constraints. We discuss how to design distance-weighted graph networks that can be executed with less than 1 μs latency on an FPGA. To do so, we consider representative tasks associated to particle reconstruction and identification in a next-generation calorimeter operating at a particle collider. We use graph architectures developed for these purposes and simplified in order to match the computing constraints of real-time event processing at the CERN Large Hadron Collider. The trained models are compressed using pruning and quantization. Using the `hls4ml` library, we convert the compressed models into firmware to be implemented on an FPGA. We show results both in terms of model accuracy and computing performance.

Keywords Deep Learning · FPGA · Graph Networks

1 Introduction

At the CERN Large Hadron Collider, collision data are collected every 25 nsec by a real-time processing system (the *trigger*) that filters away uninteresting collision events, based on a set of pre-defined algorithms. The trigger system is structured in two stages: a level-one trigger (L1T), implemented in the custom electronic of the detector and usually consisting of logic circuits emulated on field-programmable gate arrays (FPGAs) ; a high-level trigger (HLT), consisting of a computer farm (possibly powered by parallel accelerators such as FPGAs of GPUs). At the HLT, thanks to an asynchronous event processing, the accept/reject decision has to be reached with a typical latency of $\mathcal{O}(100)$ msec. At the L1T, a decision has to be taken within $\mathcal{O}(10)$ μsec. The main limitations are the synchronous nature of the processing system and the limited size of the memory buffer that pipelines the data of each collision.

While HLT algorithms have a complexity comparable to those used *offline* to produce the final physics results, a typical L1T algorithm consists of a basic set of rules. This allows to respect the latency constraint but it typically affects the resolution that one can reach. Recently, the deployment of the first Machine Learning (ML) L1T algorithm [1] has changed this tendency, raising interest in using ML algorithms as fast-to-execute approximations of complex rule-based solutions. The first example consisted of a big look-up table (LUT) implementing a boosted decision tree. One might be interested to deploy more complex architectures. At this stage, one is limited by the limited computing resources of an FPGA card, e.g. the number of digital signal processing units (DSPs).

The `hls4ml` library was designed to to facilitate the deployment of complex algorithms on FPGAs, having in mind the specific use case of an LHC L1T [1]. In this respect, support for graph architectures is an increasingly demanded feature, given the growing list of examples showing how well graph neural networks (GNNs) can deal with high-energy-physics (HEP) related tasks [2, 3, 4, 5, 6, 7, 8, 9]. In fact, while the irregular geometry of a typical HEP detector makes the use of computing vision techniques complicated, GNNs can naturally deal with the sparse and irregular nature of HEP collision data.

In this work, we show how a graph model could be efficiently deployed on FPGAs to perform inference within $\mathcal{O}(1)$ μsec for HEP-related problems. We consider the distance-weighted architectures introduced in Ref. [5] and designed in order to keep resource consumption under control by reducing as much as possible the number of operations. For this reasons, and for having shown to perform well on HEP related tasks (particle reconstruction in a calorimeter), they represent a good candidate to our purpose.

As an example, we consider the same detector geometry of Ref. [5] and two calorimeter-related tasks: identifying the nature of an incoming particle, given the shape of the energy deposit left in the detector; the clustering of overlapping energy deposits into particle candidates. On both problems, we first describe a benchmark *off-line* model, that sets the best accuracy we can reach for a give problem, regardless of resource constraints at inference time. Then, we discuss how to compress those models to meet these resource constraints and assess the final inference latency and resource utilization.

This paper is structured as follows: Section 2 describes the input data. Sections 3 and 4 describes the model architecture, off-line performance and inference resource utilization after compression, respectively for the PID and cluster problems. Conclusions are given in Section 5.

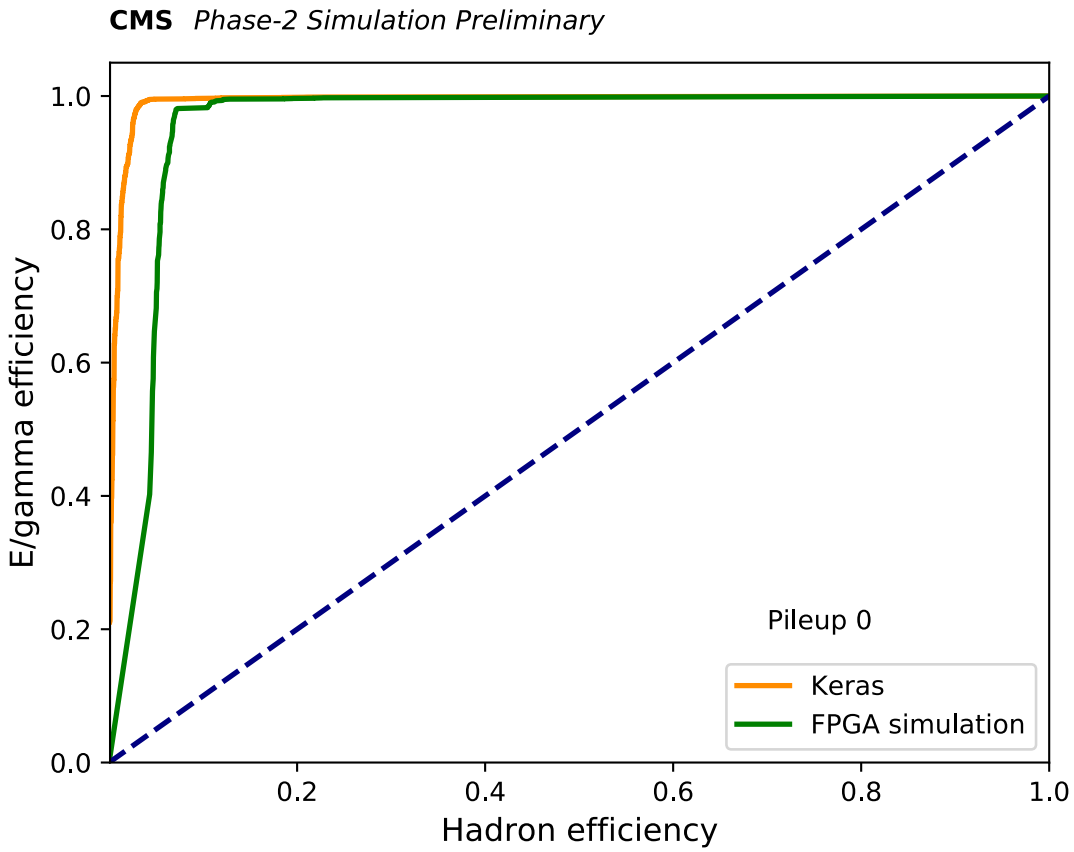

Figure 1: ROC curve for the $e/\gamma/\pi^0$ particle-identification GarNet model, for the full-precision KERAS implementation and the low-precision FPGA synthesis (estimated from FPGA emulator).

Latency	55-100 clocks	Interval	30-93 clocks
DSP	2.4-5.5k	DSP %	< 44
LUT	88/663k	LUT %	< 13
Block RAM	2/76 MB	Block RAM %	13

Table 1: Resource utilization for various configurations of the $e/\gamma/\pi^0$ particle-identification GarNet model, synthetized on a Xilinx Kintex Ultrascale XCKU115, with a clock frequency of 200 MHz.

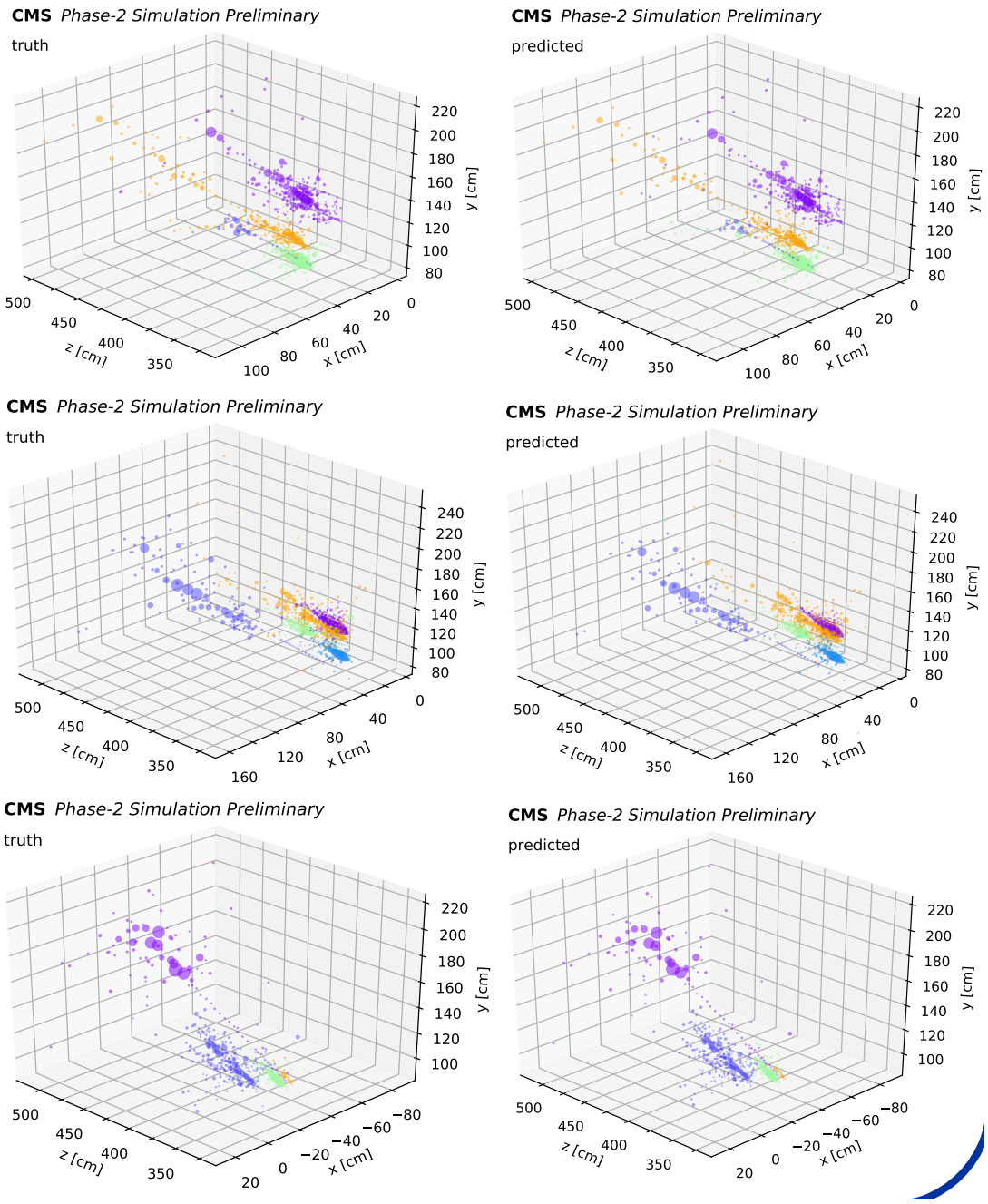

Figure 2: Comparison between true showers (left) and reconstructed clusters (right) returned by the GravNet model described in the text.bf CHANGE THE PLOTS

2 Dataset

3 Particle identification

3.1 Model architecture

3.2 Training and results

3.3 Model synthesis and performance

4 Particle Clustering

4.1 Model architecture

4.2 Training and results

M. P., A. G., K. W., S. S., V. L. and J. N. are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n^o 772369).

J. D., B. K., S. J., R. R., and N. T. are supported by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. P.H. is supported by a Massachusetts Institute of Technology University grant.

Z. W. is supported by the National Science Foundation under Grants No. 1606321 and 115164.

References

[1] CMS Collaboration, D. Acosta et al., *Boosted Decision Trees in the Level-1 Muon Endcap Trigger at CMS*, *J. Phys. Conf. Ser.* **1085** (2018) 042042.

[2] M. Abdughani, J. Ren, L. Wu and J. M. Yang, *Probing stop pair production at the LHC with graph neural networks*, *JHEP* **08** (2019) 055 [1807.09088].

[3] IceCube Collaboration, N. Choma et al., *Graph Neural Networks for IceCube Signal Classification*, 1809.06166.

[4] J. Arjona Martínez et al., *Pileup mitigation at the Large Hadron Collider with graph neural networks*, *Eur. Phys. J. Plus* **134** (2019) 333 [1810.07988].

[5] S. R. Qasim, J. Kieseler, Y. Iiyama and M. Pierini, *Learning representations of irregular particle-detector geometry with distance-weighted graph networks*, *Eur. Phys. J.* **C79** (2019) 608 [1902.07987].

[6] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, 1902.08570.

[7] E. A. Moreno et al., *JEDI-net: a jet identification algorithm based on interaction networks*, 1908.05318.

[8] E. A. Moreno et al., *Interaction networks for the identification of boosted $H \to b\bar{b}$ decays*, 1909.12285.

[9] C. Jin, S.-z. Chen and H.-H. He, *Classifying the Cosmic-Ray Proton and Light Groups on the LHAASO-KM2A Experiment with the Graph Neural Network*, 1910.07160.