Muon Trigger using Deep Neural Networks accelerated by FPGAs

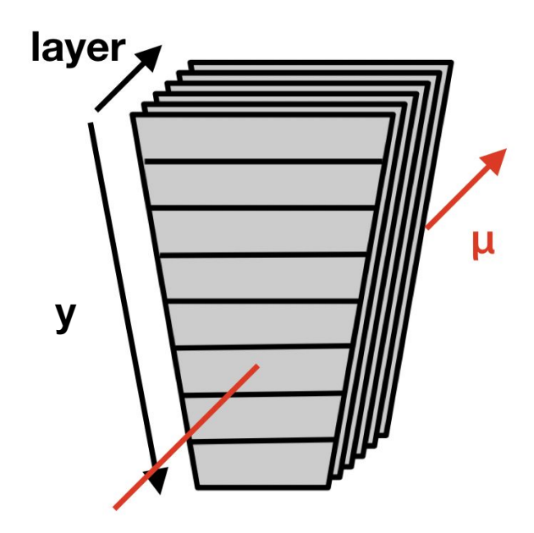
Seulgi Kim, Jason Lee, Inkyu Park,
Youngwan Son, Ian James Watson, Seungjin Yang

University of Seoul

Introduction

• Accuracy and latency are crucial to the trigger system in high luminosity particle physics experiment. Our plan is to increase accuracy with Convolutional Neural Network (CNN) based trigger, while reducing latency by FPGA (Xilinx Alveo U250) to enable online use.

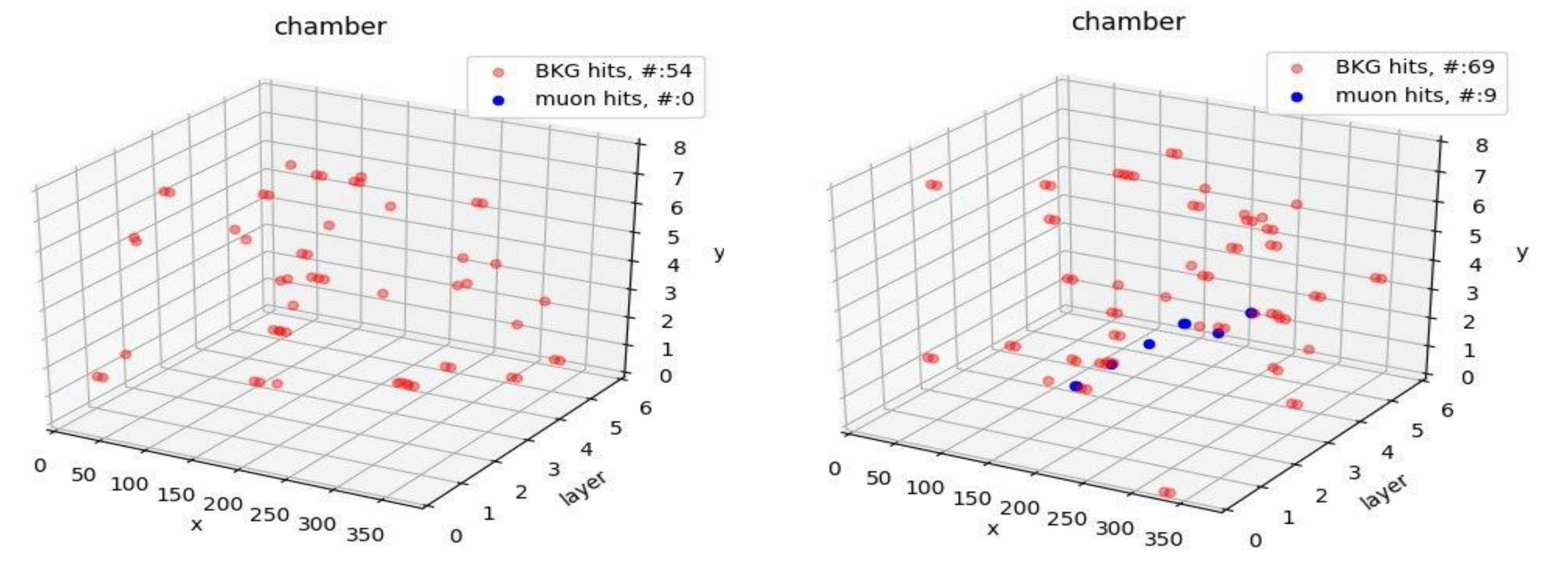
• We aim for generic muon detector chambers. When the charged particle passes through the chamber, hits are produced on the wires near the particle path. Our purpose is muon segment reconstruction in the detector image by CNN based segmentation model.



Detector

Dataset: binary image of muon detector chambers

The detector has 6 layers, with each layer consisting of 384 strips in the x axis and segmented in 8 partitions in the y axis. When the current flows on a strip, a voxel turned 1 from 0. Muons are ranging from p_T 5~100 GeV.

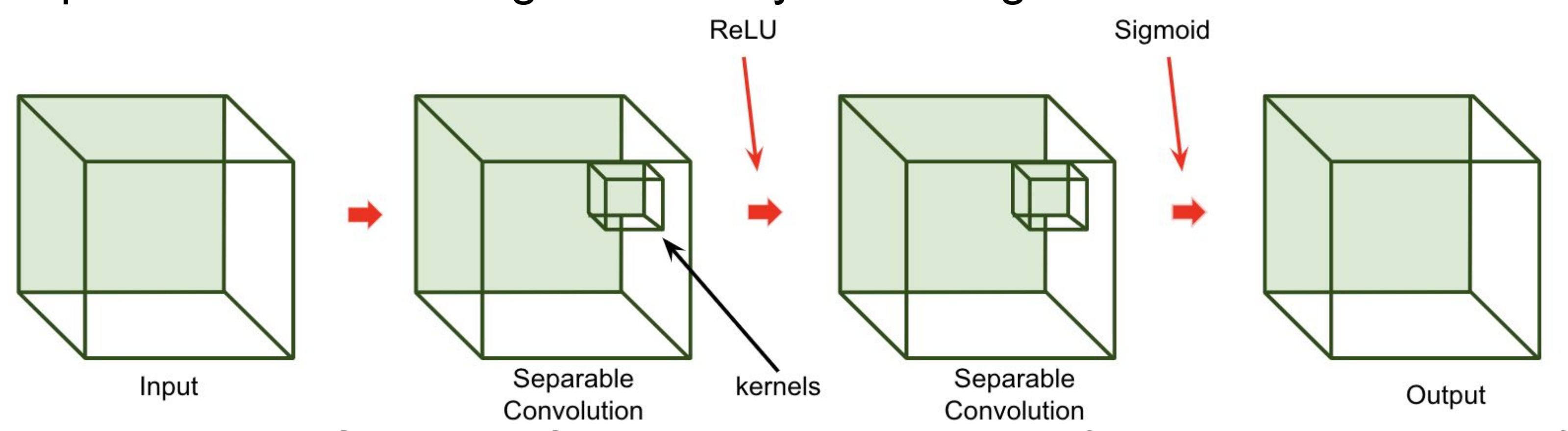


Visualization of chambers. The left one has only noise hits, and the right one has hits from noise and a muon. Blue and orange points both are 1 in these images.

Method

Training CNN based image segmentation model

The segmentation model which we want should get noisy detector images as inputs and returns images with only muon segments.

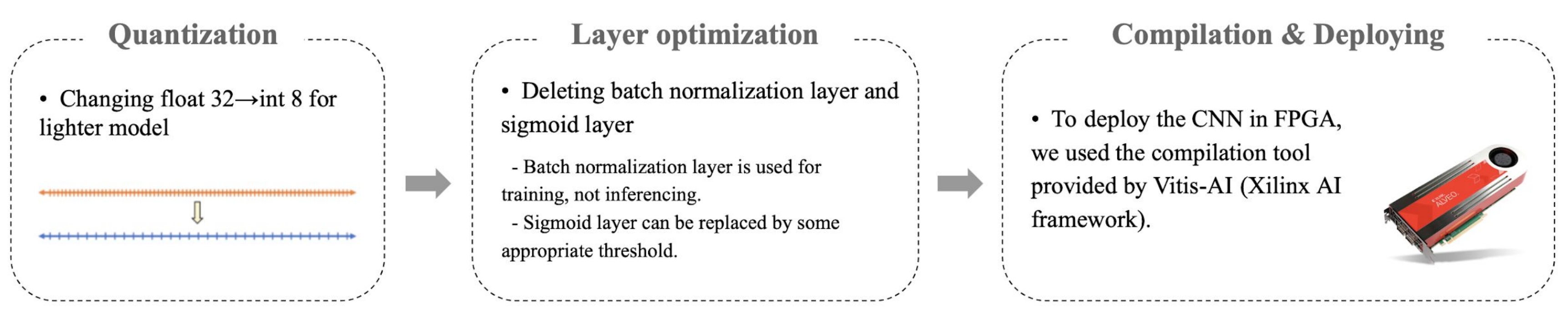


Model structures - Separable Convolution extracts spatial feature by each channel first (Depthwise), and then convolution to channel direction (Pointwise). It uses fewer parameters than standard convolution, so it requires smaller computational resources.

Loss: weighted binary cross entropy $\longrightarrow$ $\mathrm{WCE}\,(p,\hat{p}) = -\left(\beta p \log\left(\hat{p}\right) + (1-p)\log\left(1-\hat{p}\right)\right)$

"Training" means changing the parameters of model's weights by minimizing the loss, which a difference measurement between ground truth and model output. During training, convolution kernels catch the visual features of hits from muons so that model output close to ground truth.

Compressing trained model and Deploying in FPGA



Quantization

• Changing float 32→int 8 for lighter model

Layer optimization

• Deleting batch normalization layer and sigmoid layer

- Batch normalization layer is used for training, not inferencing.
- Sigmoid layer can be replaced by some appropriate threshold.

Compilation & Deploying

• To deploy the CNN in FPGA, we used the compilation tool provided by Vitis-AI (Xilinx AI framework).

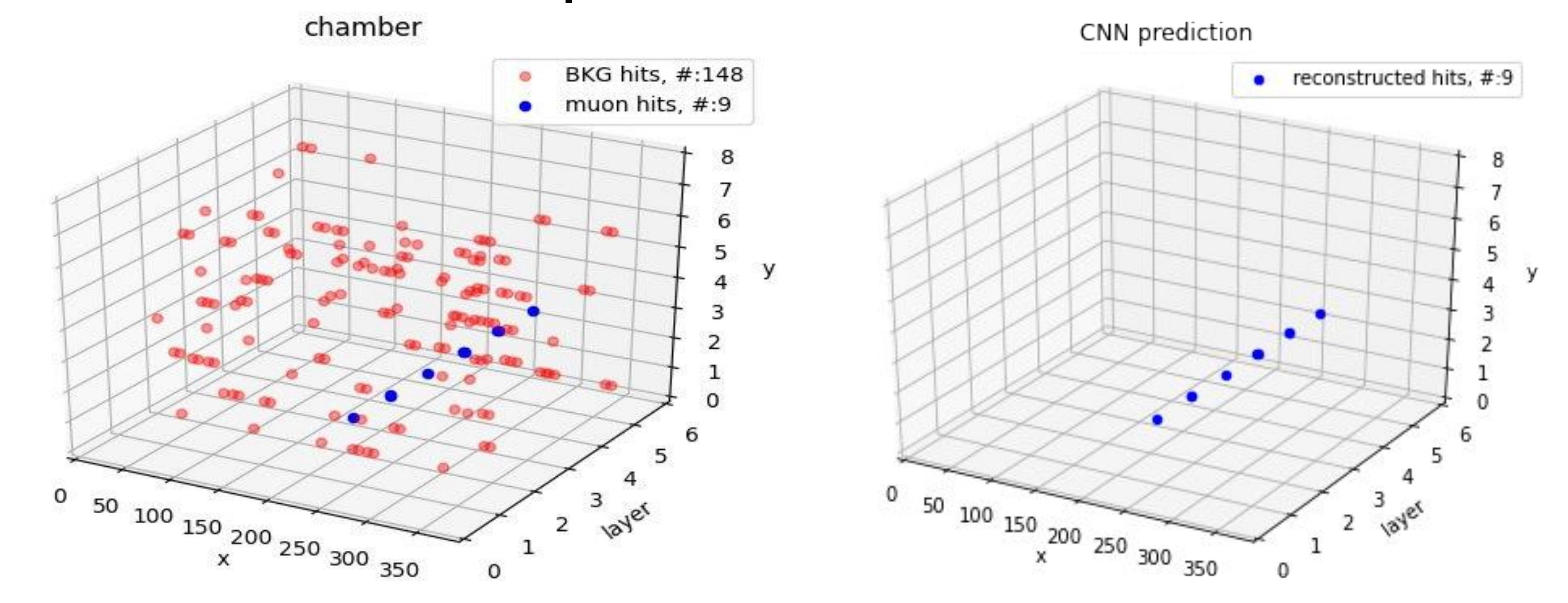
After training flow - Optimizing and Deploying CNN model

For online triggering, model needs to run inference extremely fast. Since CNN in typical CPU architectures has too high latency to used for triggering, so we chose FPGAs that can reduce latency.

Quantization, changing floating number to integer, can be used to save memory and reduce latency by simplifying the calculations that occur during inference. It potentially causes threshold shifting or accuracy change.

Results

Visualization of model output



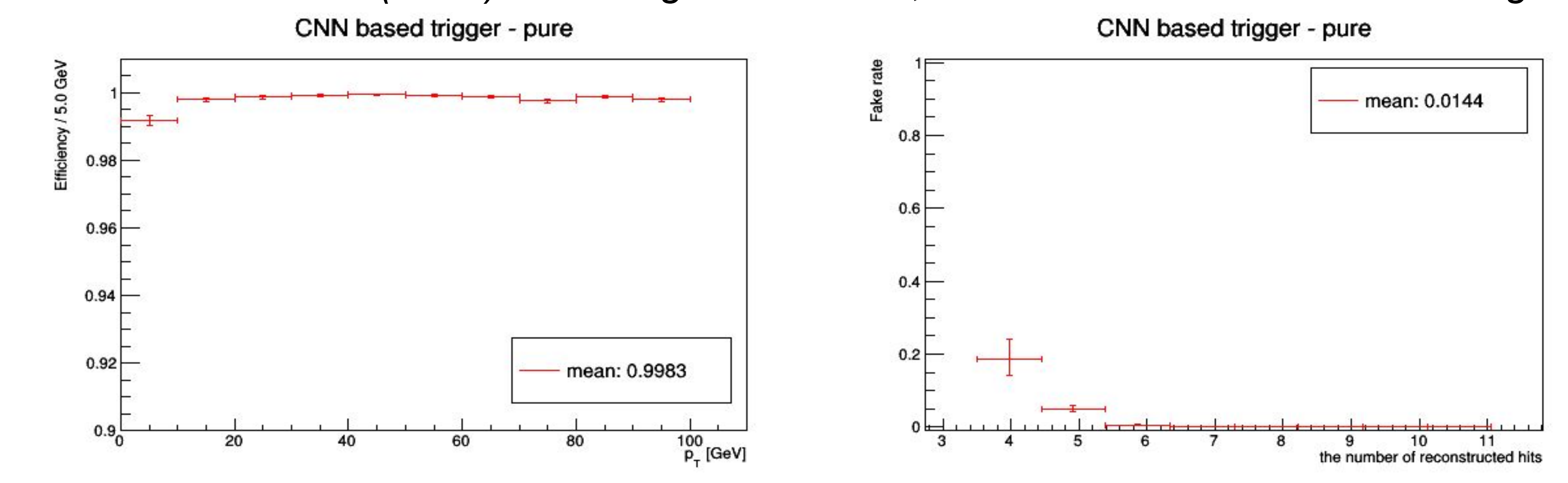
The left one is an input image for CNN, and the right one is the output of CNN.

Efficiency & Fake rate

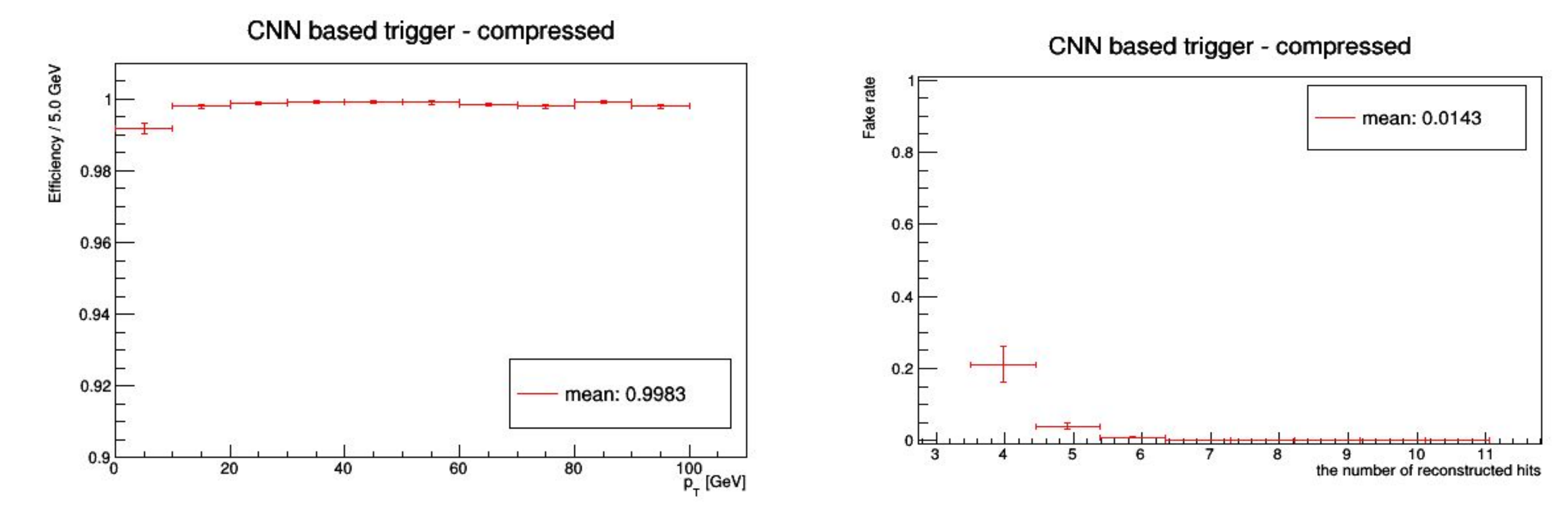
We define the efficiency and the fake rate to evaluate model performance. The efficiency is a measure of how well a model triggers muon, and the fake rate is a measure of how much a model reconstructs fake hits which not from muon.

$$Efficiency = \frac{the\ number\ of\ matched\ images}{total\ number\ of\ images\ with\ muon} \qquad Fake\ rate = \frac{the\ number\ of\ wrong\ reconstructed\ hits}{total\ number\ of\ reconstructed\ hits}$$

If True Positive Rate (TP/P) of a image is over 0.6, we define it as a "matched image".



float32 -> int8
Quantization

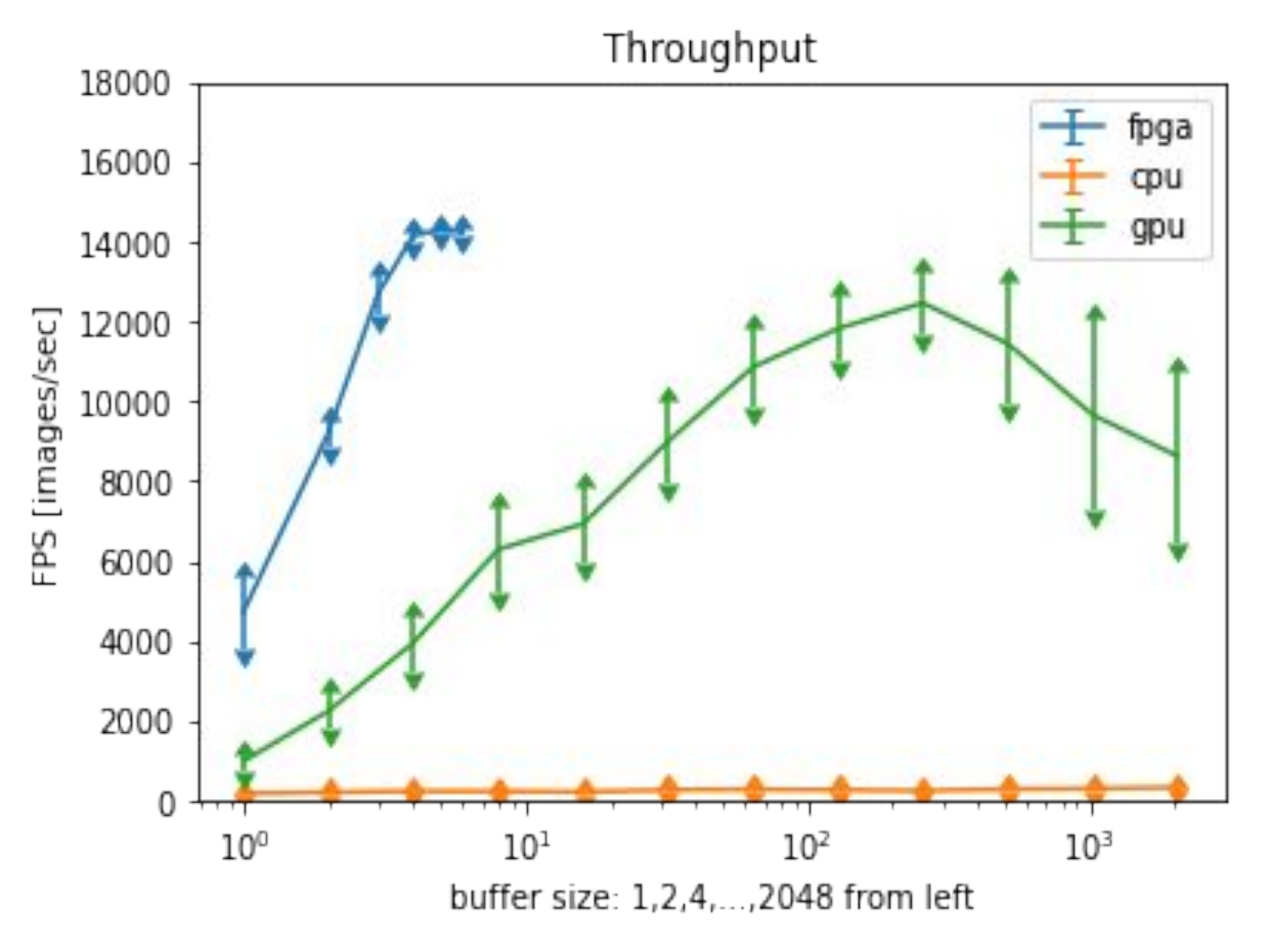


The plots of the efficiency and the fake rate (left/right). The accuracy change is caused by quantization (above/below). About 300,000 images were used for calculating efficiency and fake rate.

Trigger throughput

Throughput is measured in processed images per second (FPS).

FPGA	Alveo U250 (Xilinx)
GPU	Tesla V100 (Nvidia)
CPU	Xeon Gold 5218 (Intel)



GPU can have larger buffers than FPGA. Nevertheless, FPGA can operate faster.

Summary & Plan

• We investigated whether muon segment reconstruction can be done with the Neural Network at low-latency with FPGA.

• What FPGA we used here was Xilinx Alveo U250, which is designed for the cloud AI inference. We are also considering the embedded FPGA which can be directly connected with chamber. We will try to deploy CNN with the embedded FPGA (Xilinx ZCU 111).

References

[1] Francois Chollet (2018). Deep Learning with Python. Manning Publications Co.
[2] Xilinx (2020). Vitis AI User Guide UG1414 (v1.2). Xilinx Inc.

Youngwan Son
contact: sonkun2005@uos.ac.kr
Computational Physics Lab, University of Seoul