# Joint CPU-FPGA Hardware-Aware Quantized Training of Graph Convolutional Networks

LINKÖPING UNIVERSITY

Olle Hansson, Linköping University
Department of Electrical Engineering

## Introduction

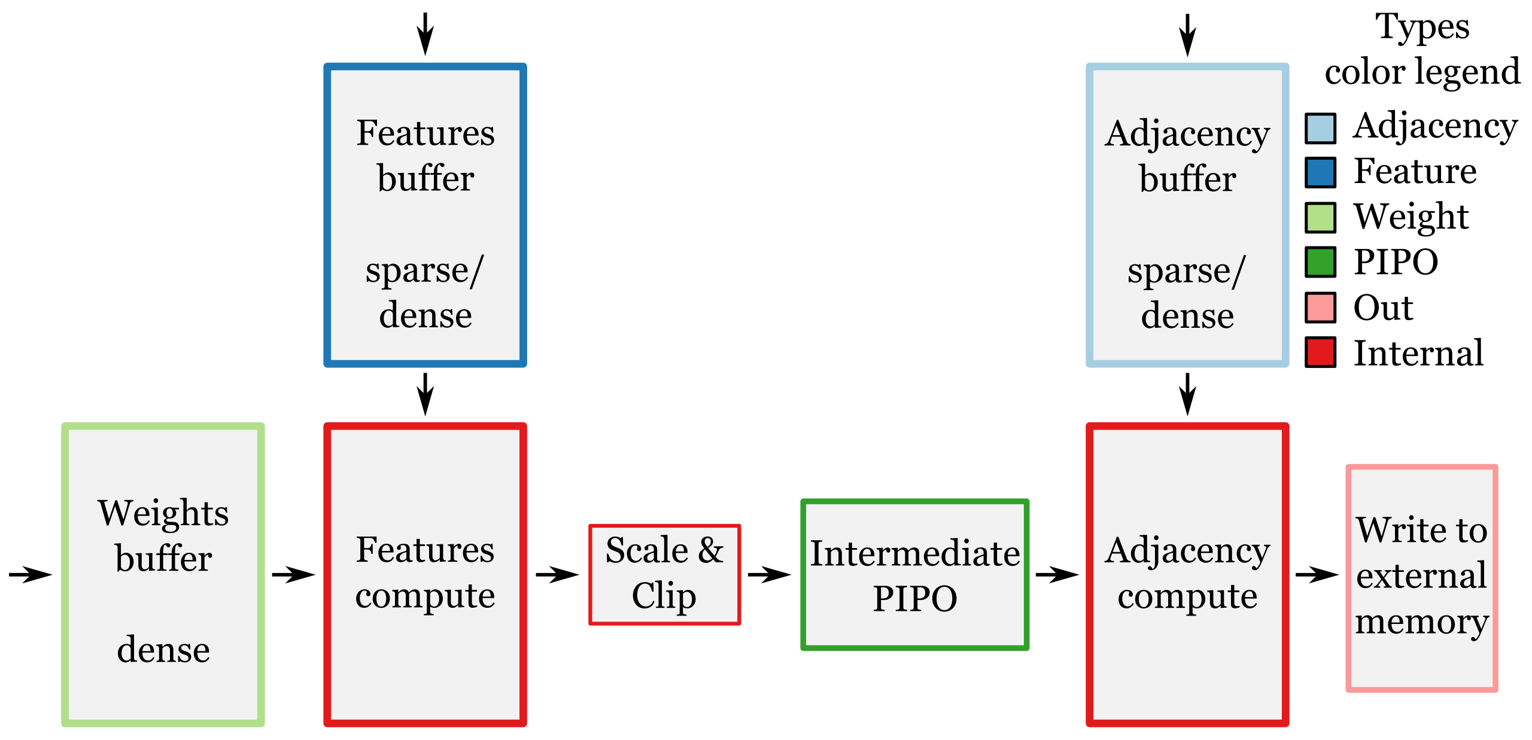
Current trends seem to be to move Artificial Intelligence (AI) systems into more applications and closer to the user. This transition would require smaller and more energy-efficient models. One of the most common ways to achieve this is by performing arithmetic operations at reduced precision by quantizing the model's operands. This work extends our previous work [1] that presented an investigation of using the graph neural network accelerator gFADES to accelerate the forward pass of the backpropagation training loop by also including the backward pass.

## Method

### gFADES Computation Architecture

The gFADES design is the hardware computation architecture used in this work. It is a development of the design used in our previous work [2]. The hardware accelerator computes a graph convolutional layer according to Kipf and Welling.

$$H^{(l+1)} = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right)$$
$$gI^{(l)} = \hat{A}G^{(l)}W^{(l)}$$
$$gW^{(l)} = H^{(l)}\hat{A}G^{(l)}$$



### Hardware-Aware Quantized Training

To make the floating-point values from the PyTorch model map correctly to the fixed-point format of the accelerator scaling and casting have to be done in a pre-processing step.

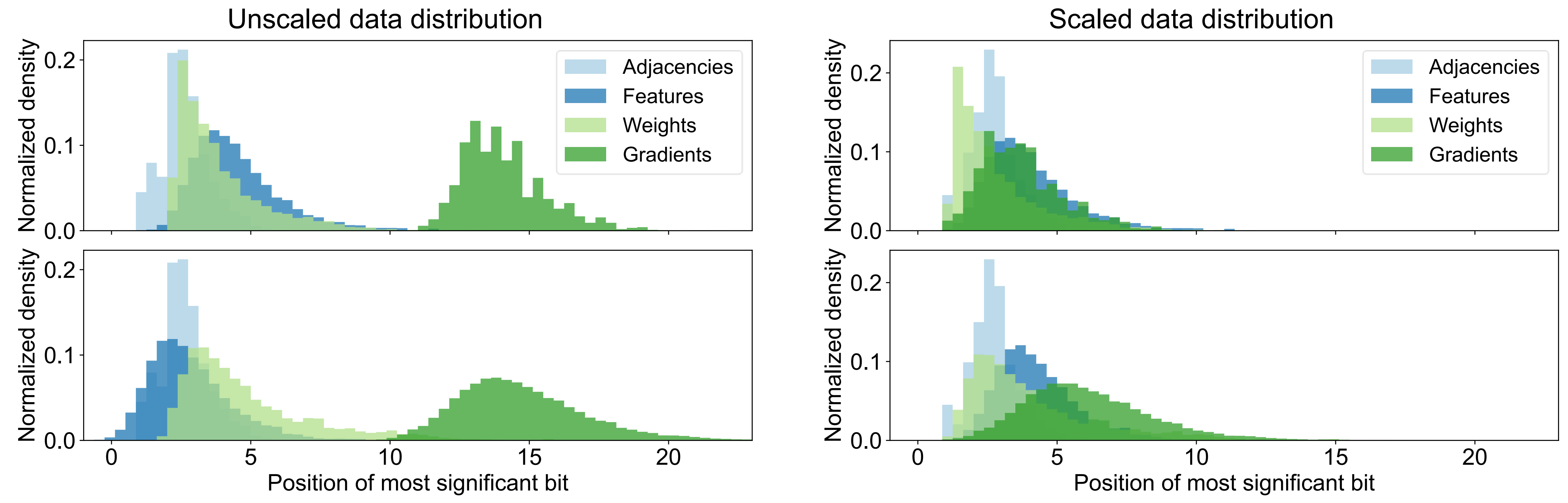
$$Q(x) = \text{clip}\left(\text{round}\left(\frac{x}{S} + Z\right), Q_{min}, Q_{max}\right)$$
$$S = \frac{x_{max} - x_{min}}{Q_{max} - Q_{min}}$$

## References

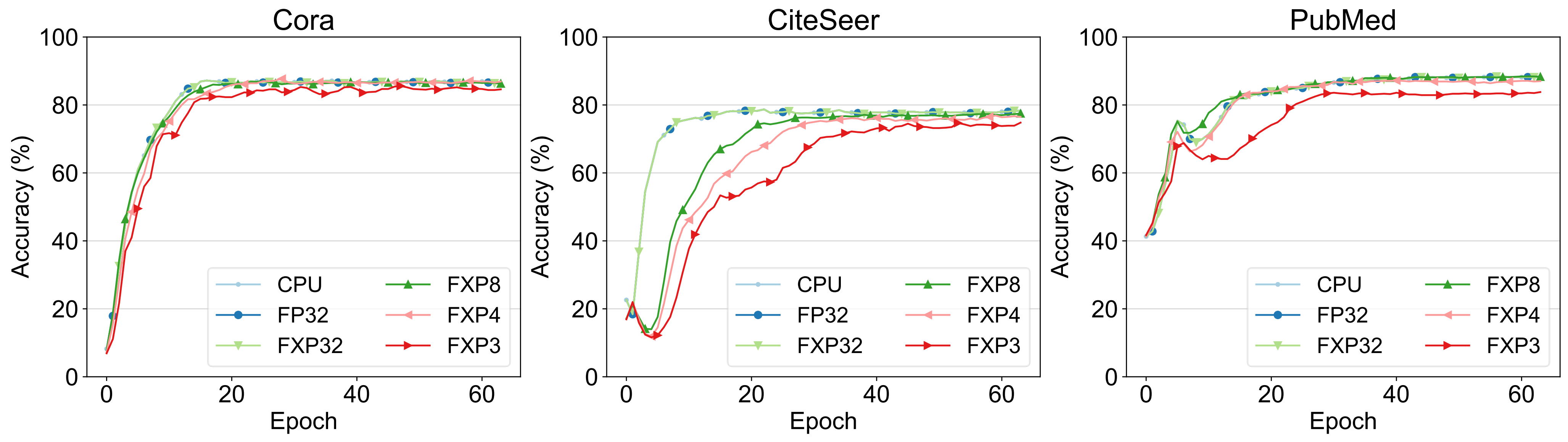
1. O. Hansson, M. Grailoo, O. Gustafsson, and J. Nunez-Yanez, “Deep quantization of graph neural networks with run-time hardware-aware training,” in *Appl. Reconfigurable Comput.. Archit., Tools, and Appl*., I. Skliarova, P. Brox Jim ́enez, M. V ́estias, and P. C. Diniz, Eds. Cham: Springer Nature Switzerland, 2024, pp. 33–47.
2. J. Nunez-Yanez, “Fused architecture for dense and sparse matrix processing in TensorFlow Lite,” *IEEE Micro*, vol. 42, no. 6, pp. 55–66,Nov. 2022.
3. J. Nunez-Yanez, “Accelerating graph neural networks in pytorch with HLS and deep dataflows,” in *Appl. Reconfigurable Comput.. Archit.,Tools, and Appl.*, F. Palumbo, G. Keramidas, N. Voros, and P. C. Diniz,Eds. Cham: Springer Nature Switzerland, 2023, pp. 131–145.

## Results

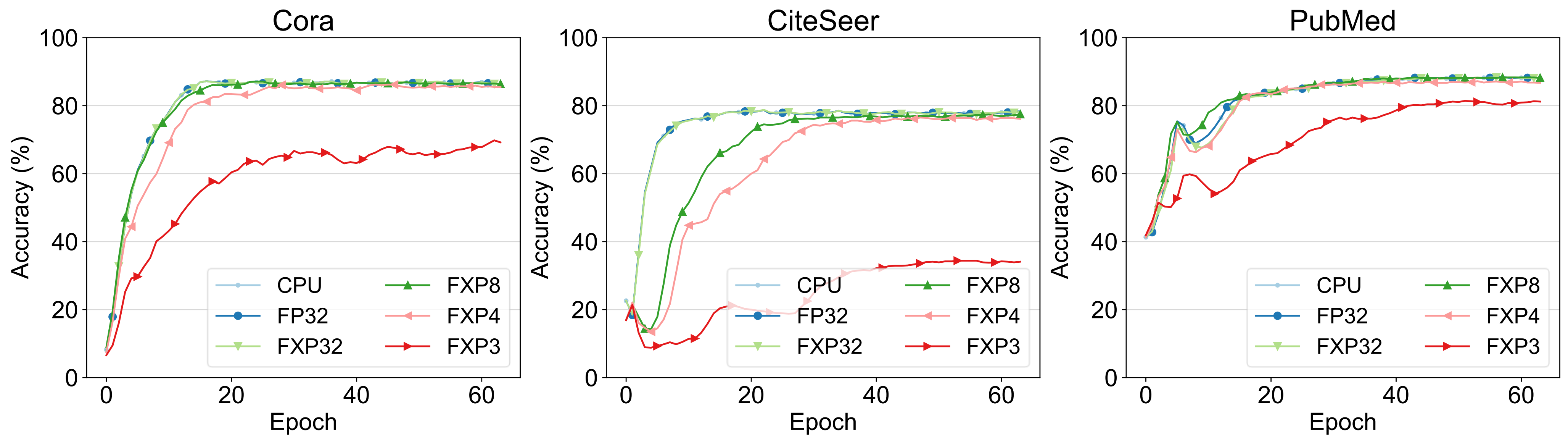
Pre-processing is crucial now that we mix different matrices in the same computation pipeline. The data distributions for the different matrices can become significantly more similar in terms of range and precision after scaling.



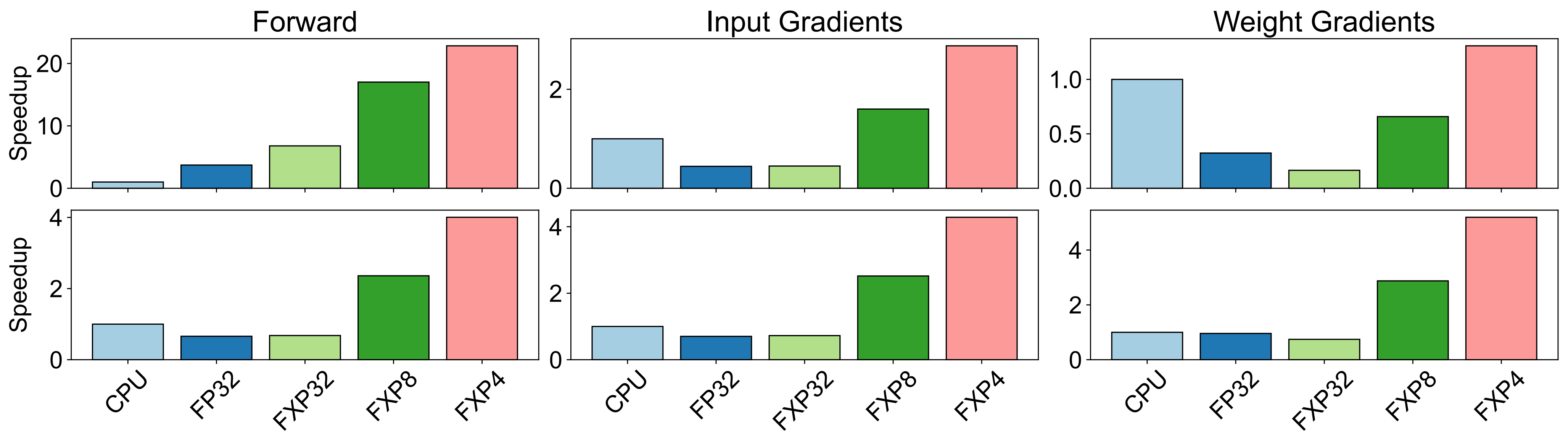
Training accuracy results of only quantizing the forward pass of the backpropagation algorithm: In this case, the backward pass is done on the processing system of the FPGA in 32-bit floating-point precision.



Training accuracy when running both the forward and backward pass quantized on the accelerator: The computation of the linear layer, the output loss, the first set of gradients, and the weight updates are not quantized.



Execution speedup of the *kernel* measurement: The kernel measurement is taken as close as possible to the computation of the tensor multiplication, not including any pre- and post-processing. Additional speedup results in inference mode have been done in our previous work [3].



## Conclusions

In this work, we have investigated the potential of quantization applied to both the forward and backward pass part of the backpropagation training loop for Graph Convolutional Network (GCN) models. Quantization has been successful down to 4 bits without significant degradation leveraging the capabilities of the gFADES FPGA accelerator.

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM