Politecnico
di Torino

1859

Master Degree course in Data Science and Engineering
Master Degree Thesis

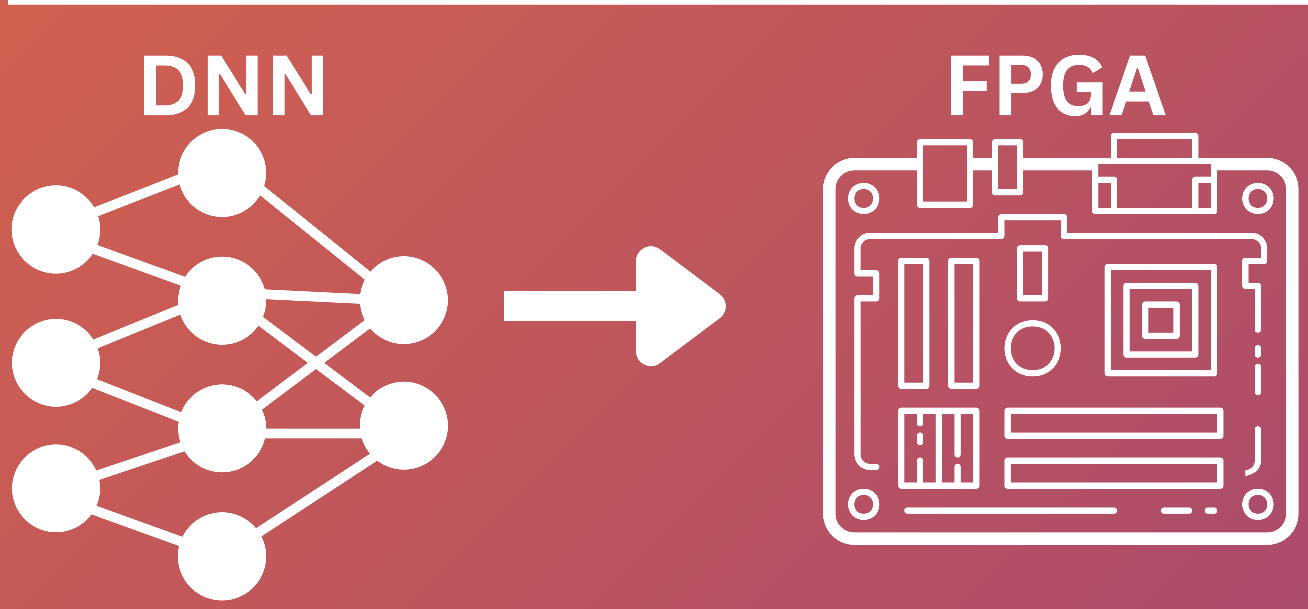
Deploying Deep Learning on FPGA: an assessment of ConvNets performance on Xilinx Zynq MPSoC using Vitis-AI development platform

Supervisors
Prof. Andrea Calimera
Dr. Roberto Giorgio Rizzo

Candidate
Gabriele Cuni

Accademic year 2021/2022

OBJECTIVE



The **aim** of my thesis was the deployment of deep neural network algorithms on a **Field-Programmable Gate** Array also known as FPGA

Making a deployment flow by using Xilinx Vitis AI and PYNQ and assess the capability of the tools.

To assess the accuracy and inference throughput of the models deployed on the Zynq UltraScale+ MPSoc

OUTLINE

- Introduction
- Background & Assessment
    - State-of-the-art CNN MobileNet
    - Xilinx Vitis AI
    - HW Setup: Zynq UltraScale+ MPSoc
    - CNN Deployment Flow
- Experimental Results

OUTLINE

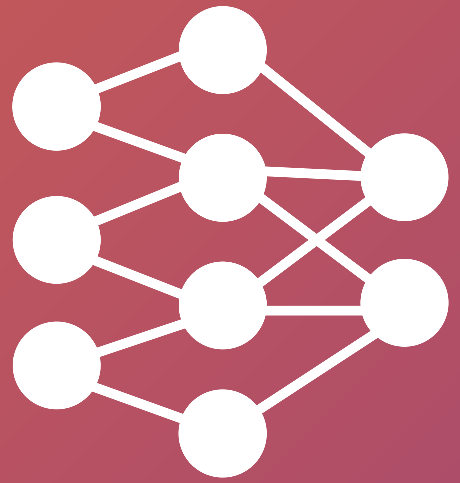
- **Introduction**
- Background & Assessment
    - State-of-the-art CNN MobileNet
    - Xilinx Vitis AI
    - HW Setup: Zynq UltraScale+ MPSoc
    - CNN Deployment Flow
- Experimental Results

INTRODUCTION - Context

EDGE

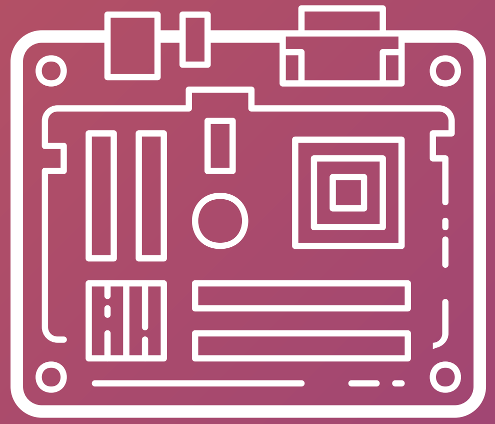
Intelligence

Nowadays **Edge Intelligence** has received significant research attention for a wide range of application scenarios, such as **smart cities** and **Internet of Vehicles**



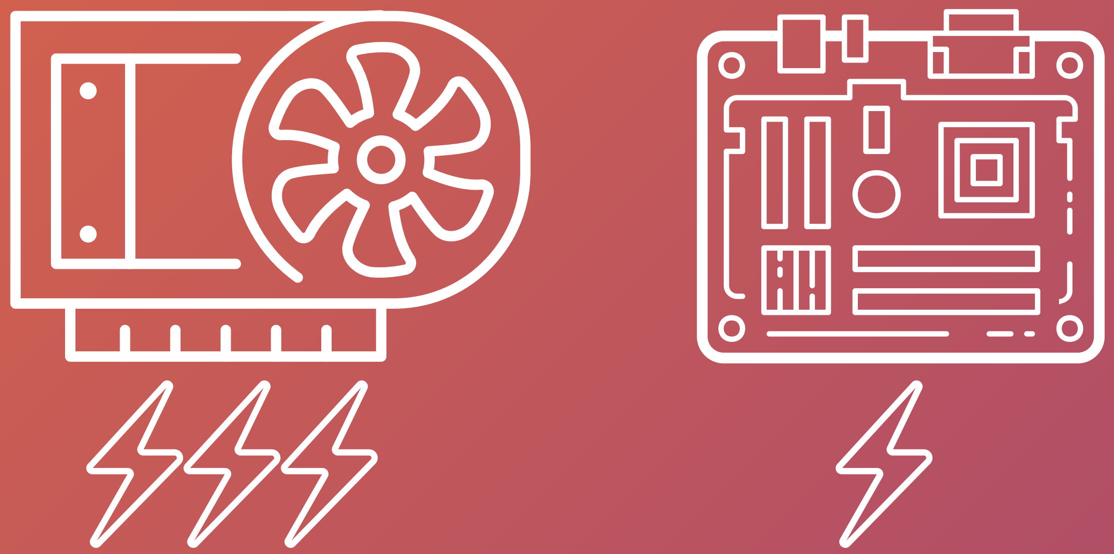
Deep Neural Networks integrated in **embedded** devices are the enabling software to bring intelligence to the edge.

FPGA



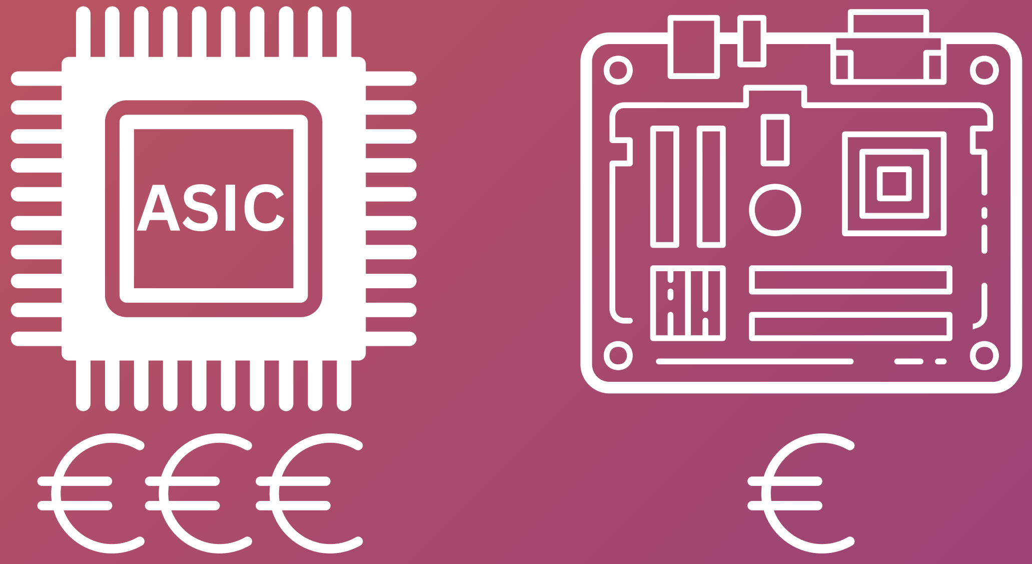
Field-Programmable Gate Arrays are an excellent hardware component for the implementation of **DNNs** on the edge, thanks to their high **parallelism**, energy **efficiency** and **flexibility**.

INTRODUCTION - Motivations



FPGAs are more energy efficient than **GPU**s.
Energy consumption is a fundamental constraint in embedded systems.

FPGAs have high customization potential as they are a configurable hardware accelerator cable to satisfy very different design constraints.



ASICs are also energy efficient with high performances, but they are complex to design and they have high access costs.

INTRODUCTION - Challenges

QUANTIZATION

DNN needs to be quantized in order to be deploy on the FPGA that works with fixed-point 8-Bit values.
Quantization techniques can lead to accuracy loss.

The greatest challenge is due to the lack of widespread tools for implementing deep learning on FPGAs.
Major deep learning frameworks do not offer standard deployment flow for FPGAs.

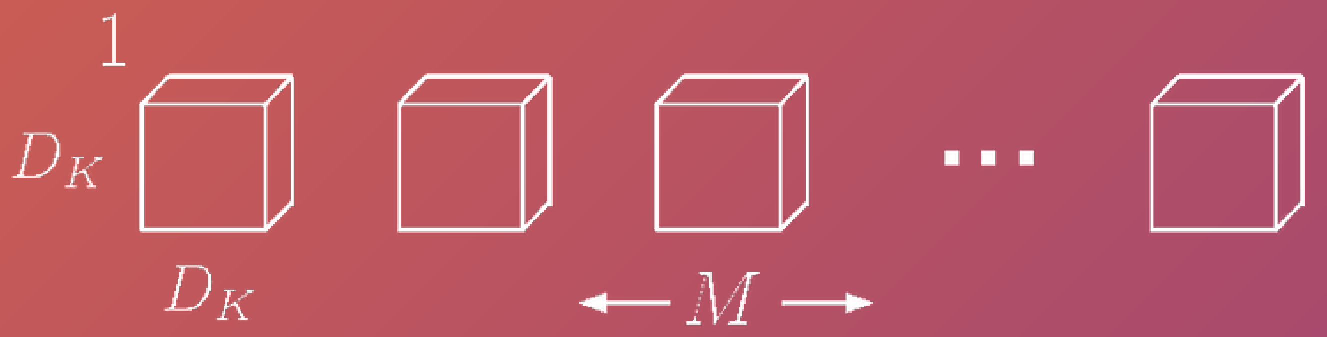
There is not an extensive literature available that directly concerns the analysis and creation of a stable flow for the implementation of deep neural networks on FPGAs.

OUTLINE

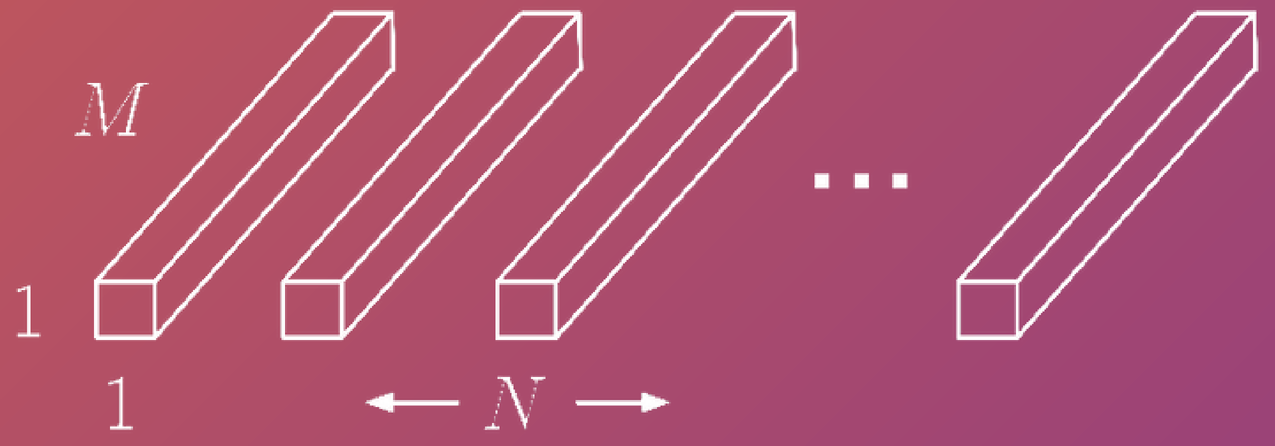
- Introduction
- **Background & Assessment**
  - State-of-the-art CNN MobileNet
  - Xilinx Vitis AI
  - HW Setup: Zynq UltraScale+ MPSoc
  - CNN Deployment Flow
- Experimental Results

BACKGROUND - CNN MobileNet

Depthwise convolution



Pointwise convolution



Computational cost

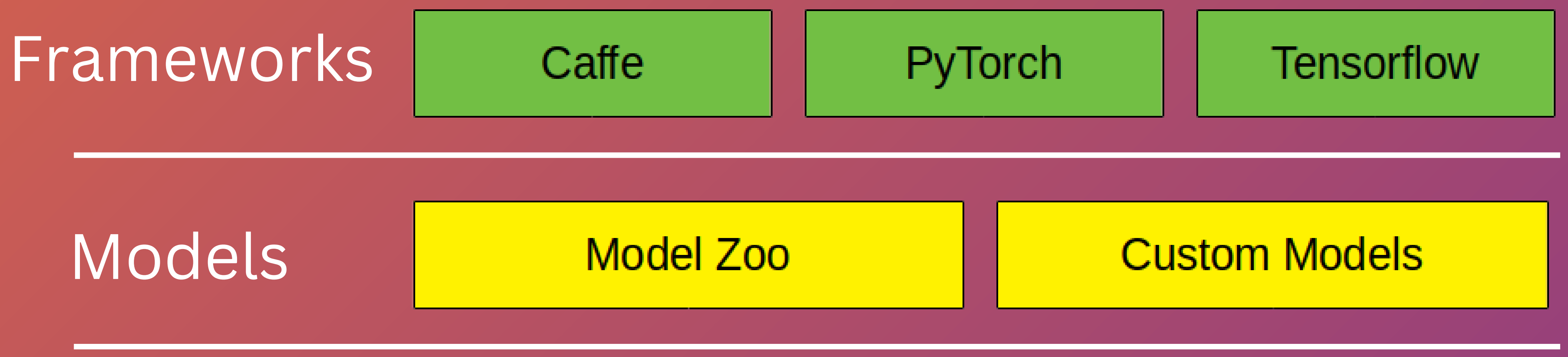
$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$$

The **MobileNet** is an efficient CNN, which is designed for the image recognition task. The main purpose of the model is to be **flexible**, **small** and **fast** .

Width Multiplier: The role of the width multiplier is to thin a network uniformly at each layer by varying the number of input and output channel.

Resolution Multiplier: It is the input image size, therefore the internal representation of every layer is subsequently reduced accordingly.

BACKGROUND - Xilinx Vitis AI



Frameworks: Caffe, PyTorch, Tensorflow

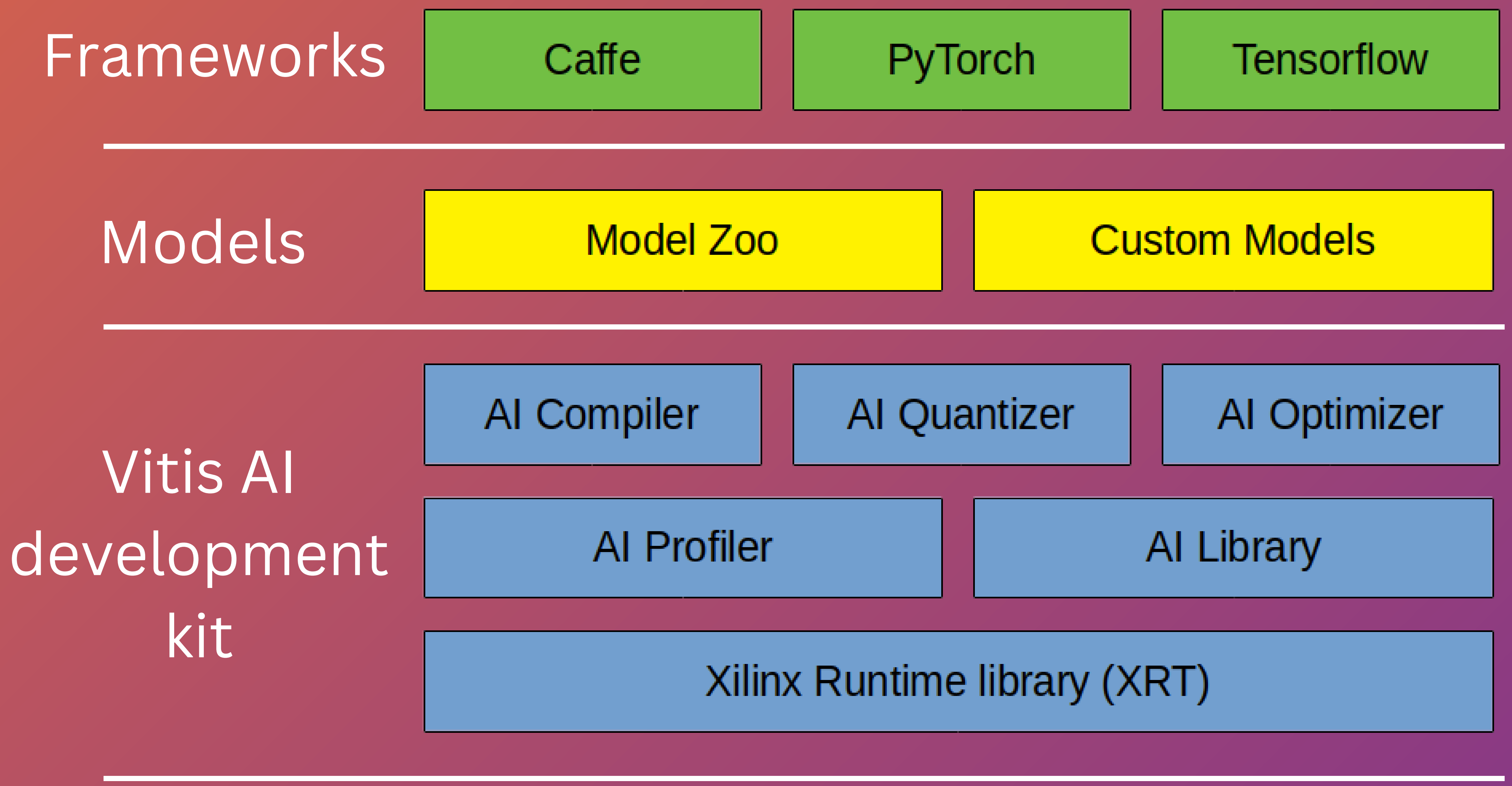
Models: Model Zoo, Custom Models

Data Science and Data Engineering layers.

Models can be made or imported from any standard frameworks.

Ready to be used models can also be taken from the Vitis AI model Zoo

BACKGROUND - Xilinx Vitis AI



Frameworks

Caffe

PyTorch

Tensorflow

Models

Model Zoo

Custom Models

Vitis AI development kit

AI Compiler

AI Quantizer

AI Optimizer

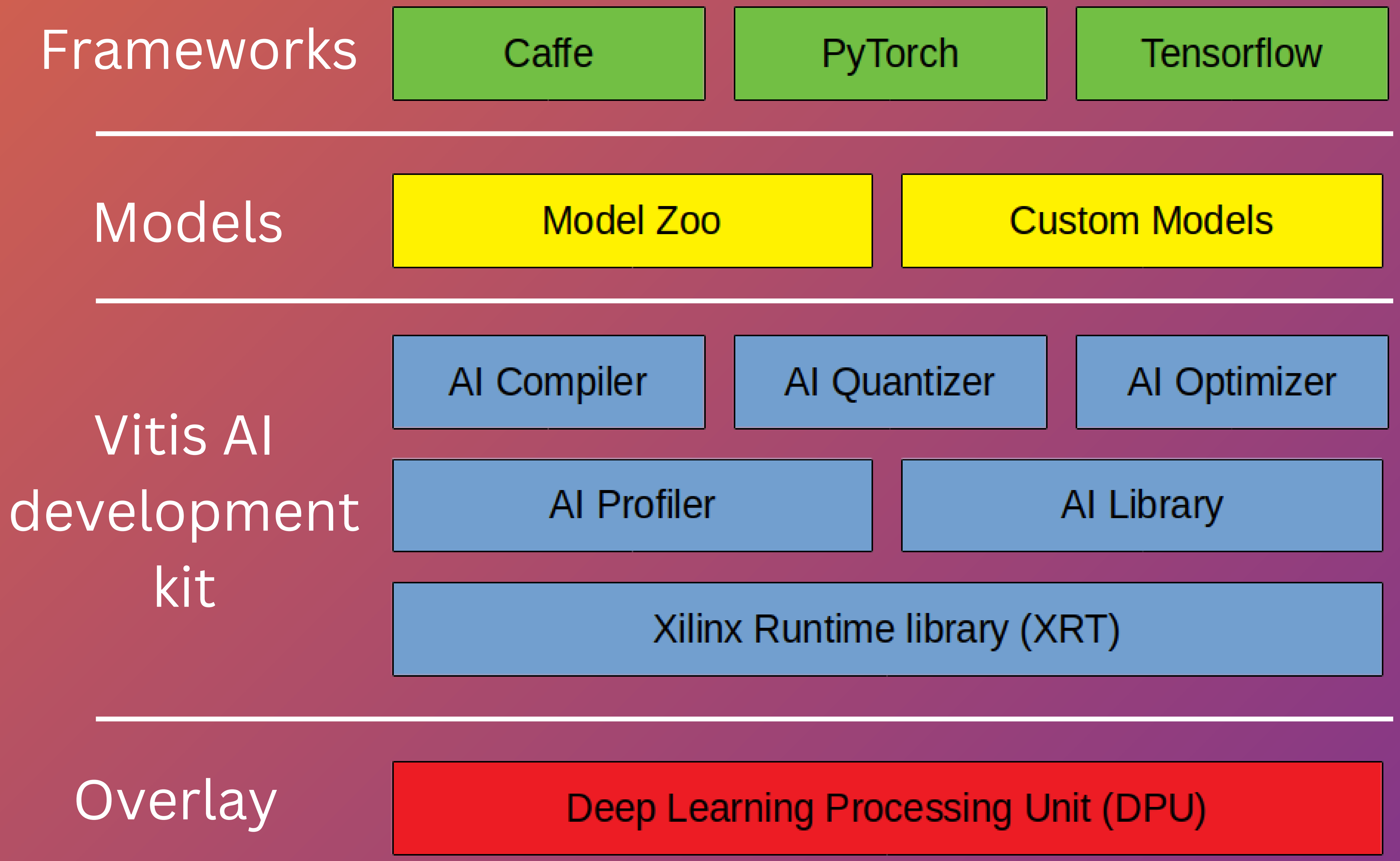
AI Profiler

AI Library

Xilinx Runtime library (XRT)

The Vitis AI development kit is the set of **tools** and **libraries** given by Xilinx in order to deploy **DNN** on the Xilinx devices.

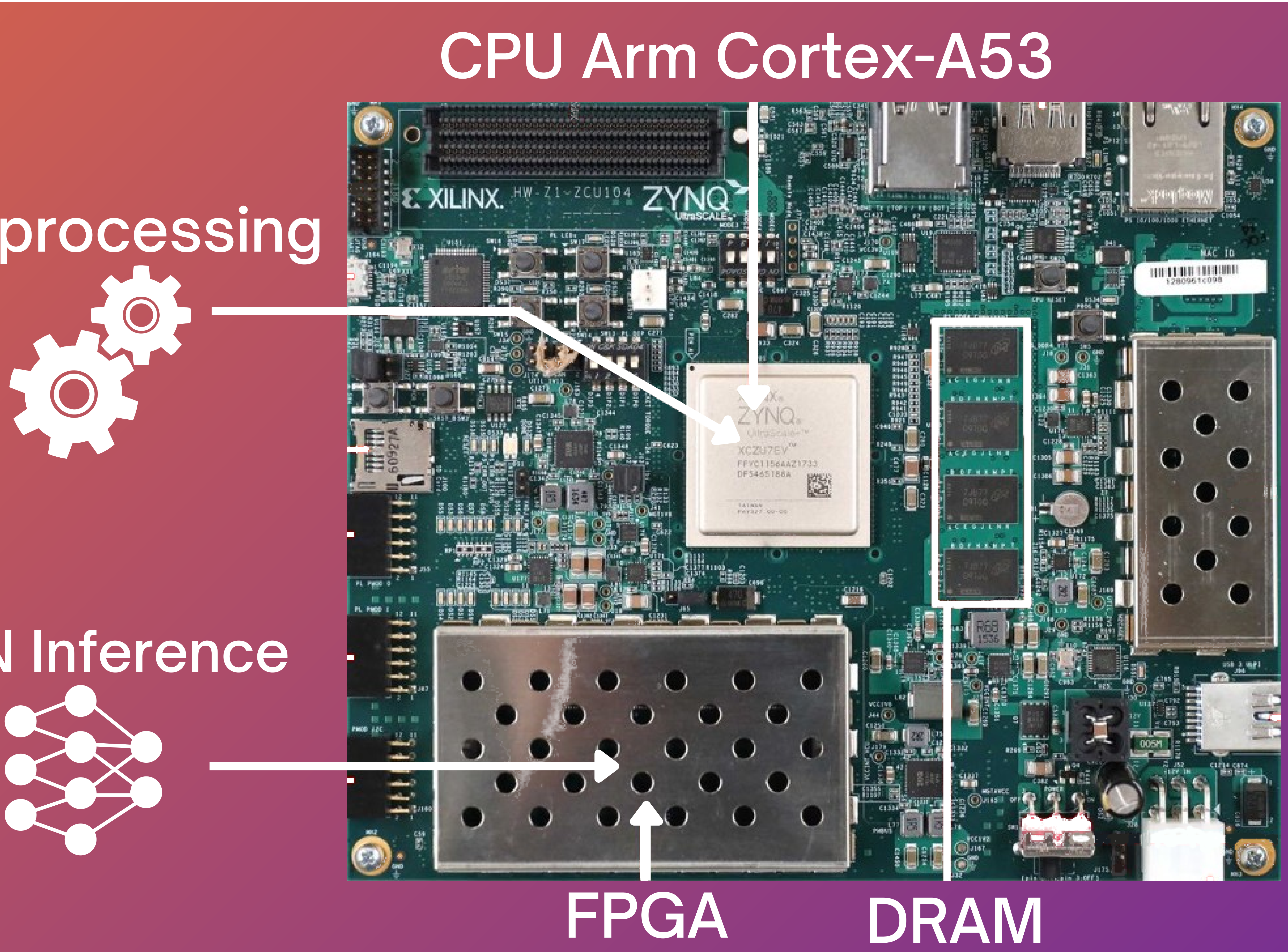
BACKGROUND - Xilinx Vitis AI



Frameworks	Caffe	PyTorch	Tensorflow
Models	Model Zoo	Custom Models	
Vitis AI development kit	AI Compiler	AI Quantizer	AI Optimizer
	AI Profiler	AI Library	
	Xilinx Runtime library (XRT)		
Overlay	Deep Learning Processing Unit (DPU)		

The overlay is the hardware representation of the DNN that in Xilinx is called DPU.

BACKGROUND - Zynq UltraScale+ MPSoC



Operating System

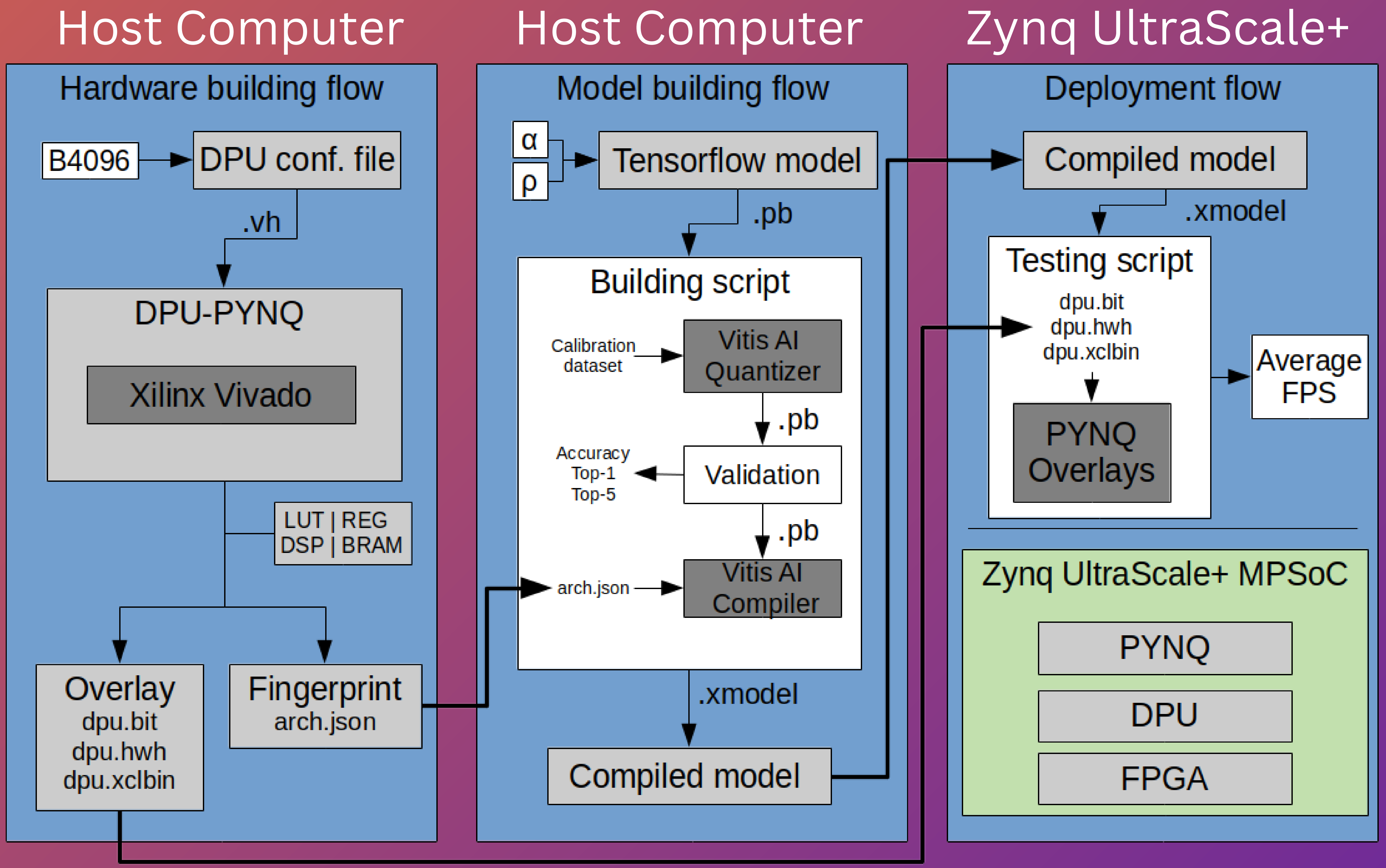
- Jupyter
- PYNQ
- Linux Kernel

HW Components

- 2 GB DDR4 RAM
- 16nm FinFET+ FPGA
- Arm Cortex-A53
- Arm Cortex-R5
- DisplayPort v1.2a
- USB 3.0
- SATA 3.1
- PCIe 1.0/2.0

ASSESSMENT - DNN Deployment Flow

The proposed deployment flow

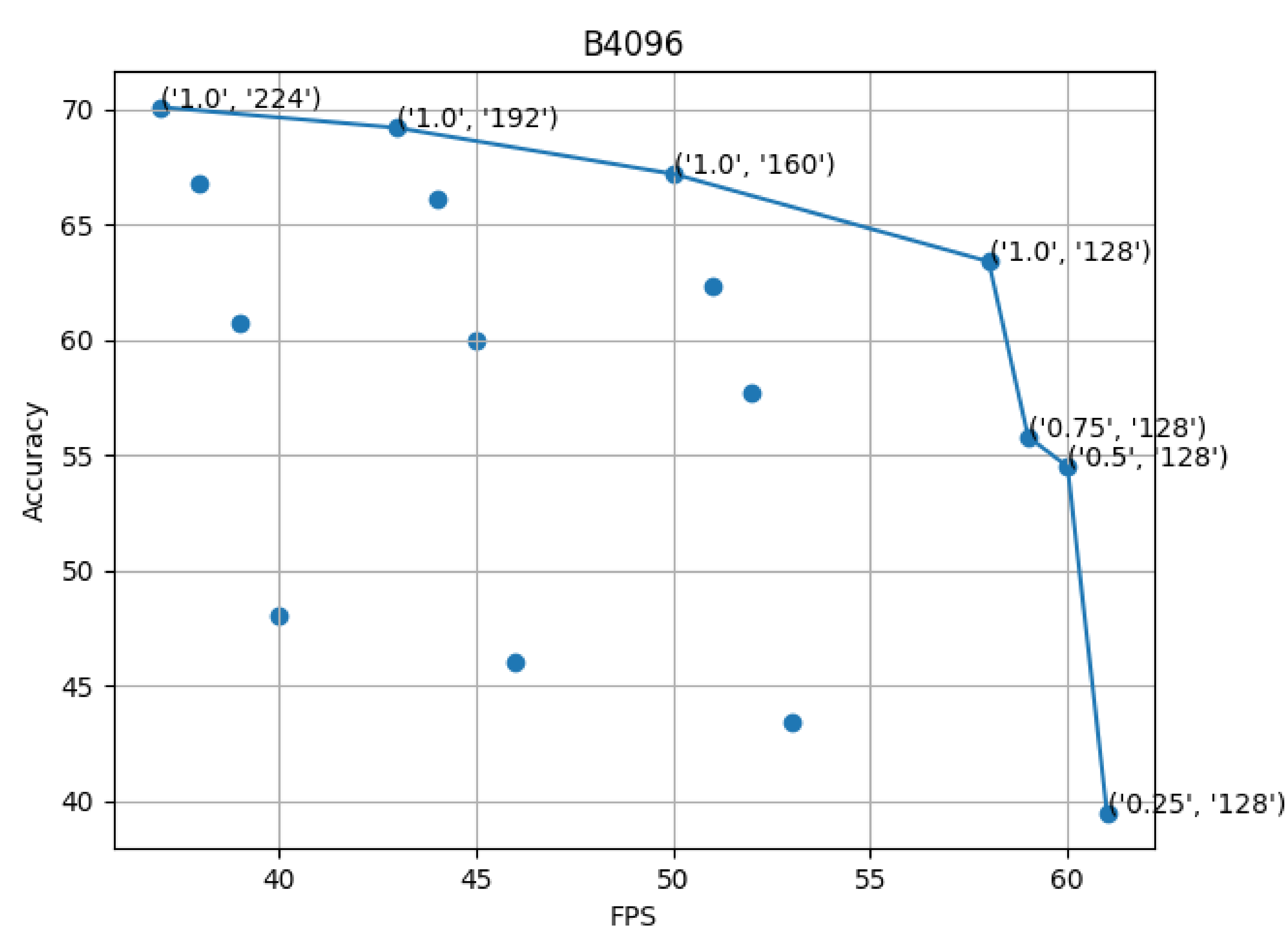


OUTLINE

- Introduction
- Background & Assessment
    - State-of-the-art CNN MobileNet
    - Xilinx Vitis AI
    - HW Setup: Zynq UltraScale+ MPSoc
    - CNN Deployment Flow
- **Experimental Results**

EXPERIMENTAL RESULTS - DPU conf. B4096

The B4096 is the DPU with the most hardware resources



Fastest Topology:

MobileNet (0.25, 128)

FPS: 61

Acc: 39.5 %

Most Accurate Topology:

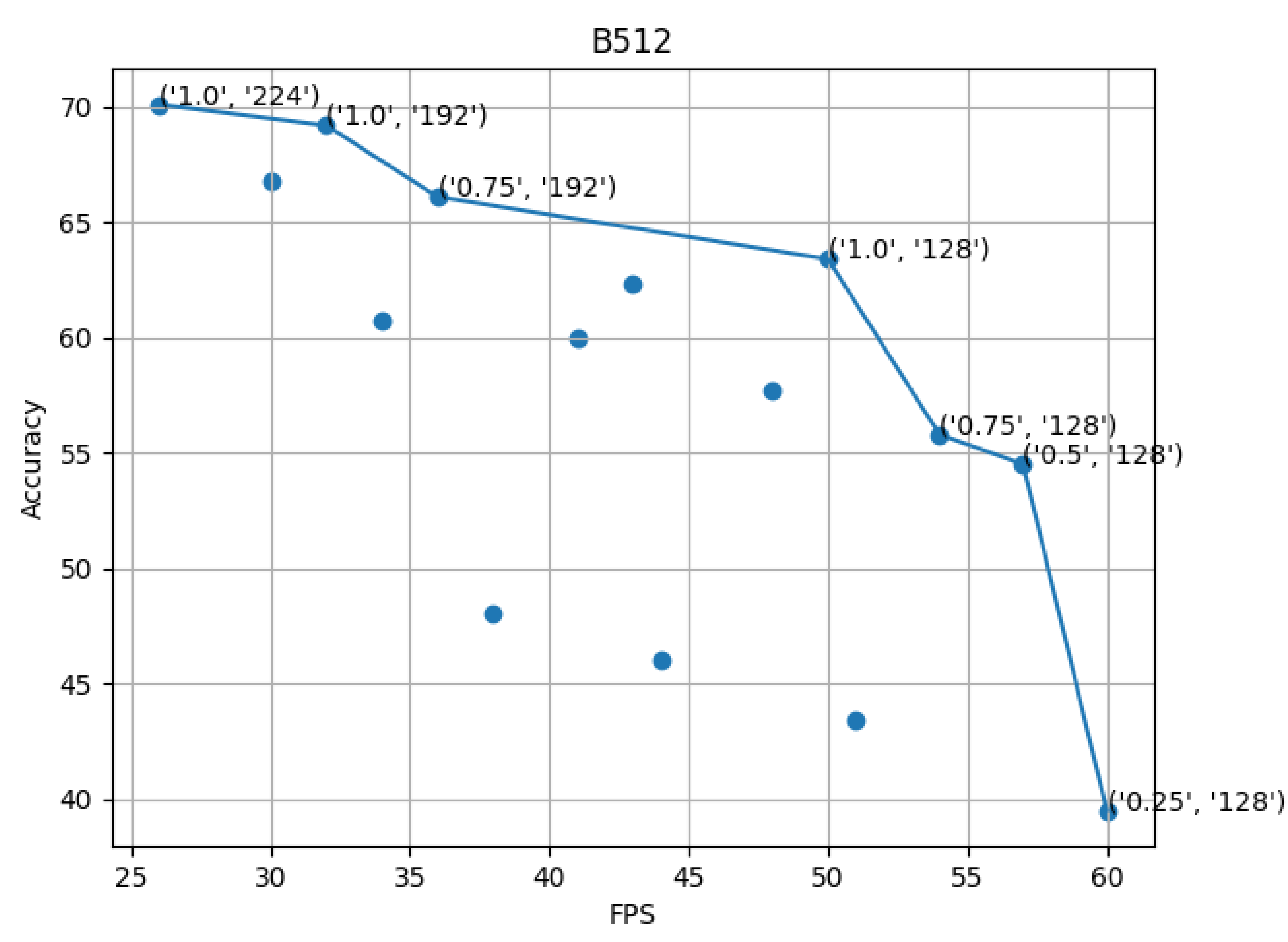
MobileNet (1.0, 224)

FPS: 38

Acc: 70.1 %

EXPERIMENTAL RESULTS - DPU conf. B512

The B512 is the DPU with the fewest hardware resources



Fastest Topology:
MobileNet (0.25, 128)
FPS: 60
Acc: 39.5 %

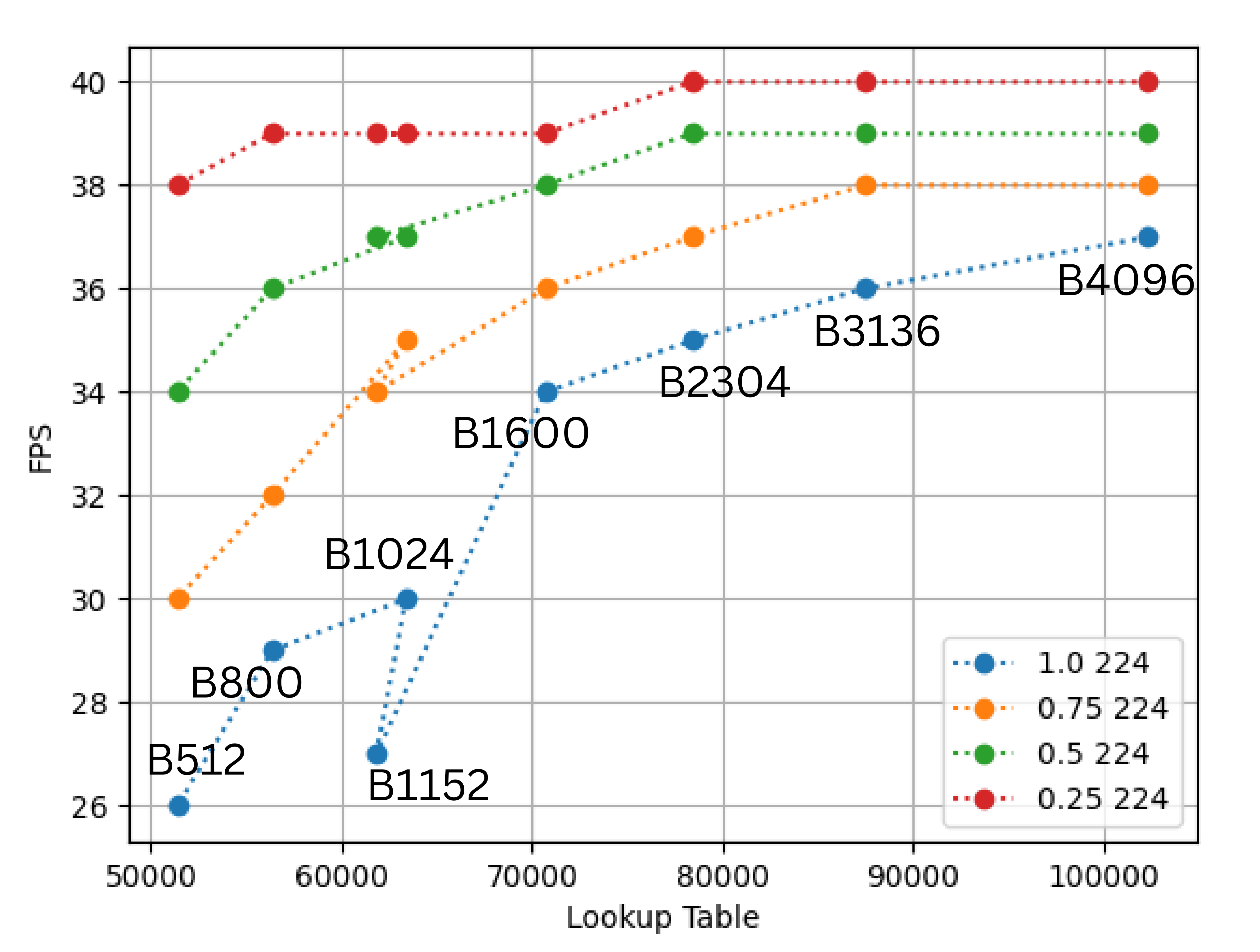
Most Accurate Topology:
MobileNet (1.0, 224)
FPS: 26
Acc: 70.1 %

EXPERIMENTAL RESULTS - Lookup Table Analysis

FPS in relation to the number of lookup tables available to each DPU

LUT

B4096	102319
B3136	87435
B2304	78380
B1600	70699
B1152	61818
B1024	63380
B800	56357
B512	51371



the least accurate

the most accurate

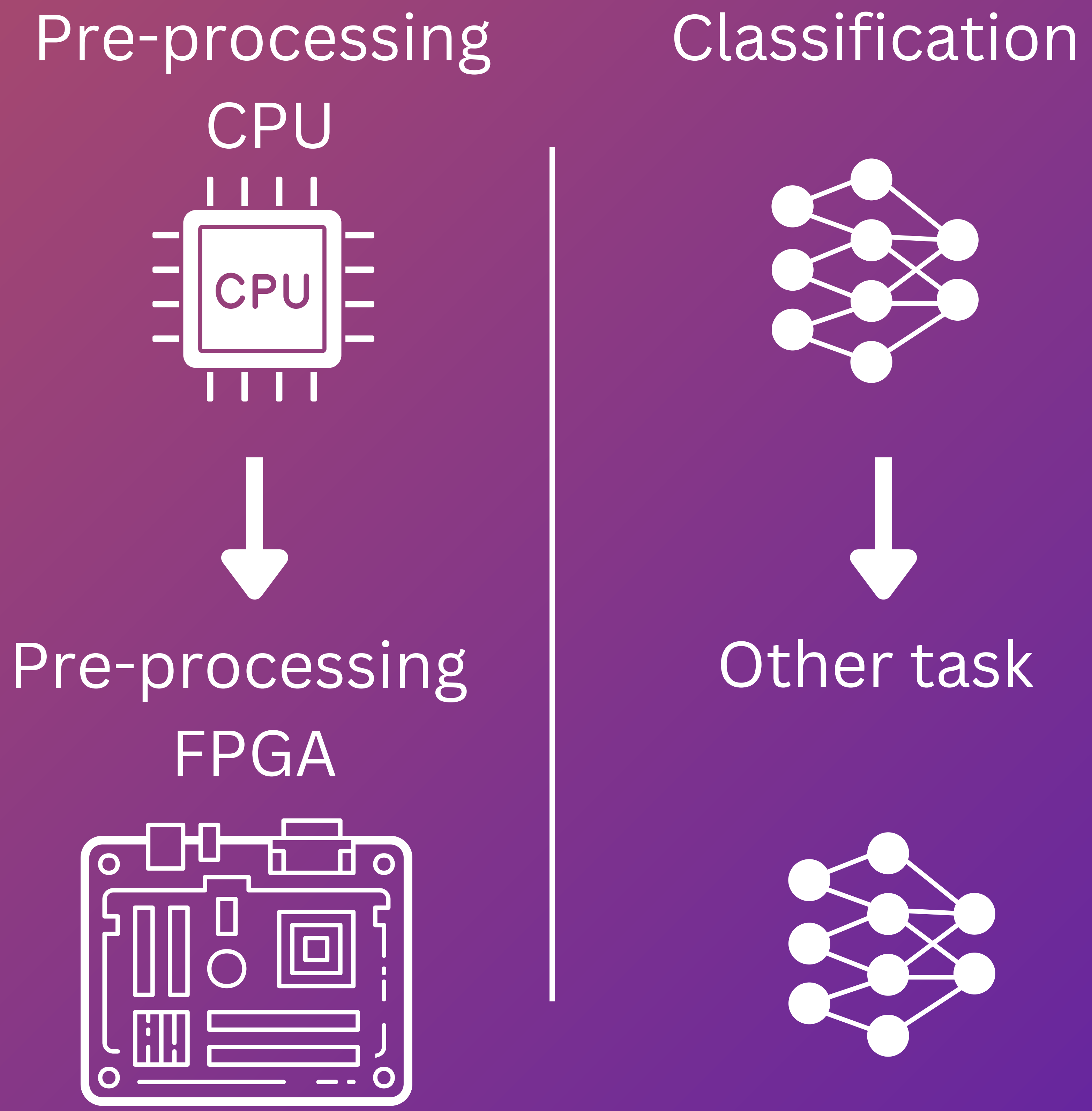
CONCLUSIONS

The accelerated implementation of the MobileNets on the FPGA has got excellent performance on all the DPU configurations.

The results has shown a high throughput, which is compatible with real-time edge applications, even on the smallest available FPGA architecture.

Therefore, it can be said that the deployment of deep neural networks on FPGA is an excellent design choice, although it is necessary to be very careful in the quantization phase to avoid excessive accuracy reduction of the models.

Future works



Pre-processing CPU

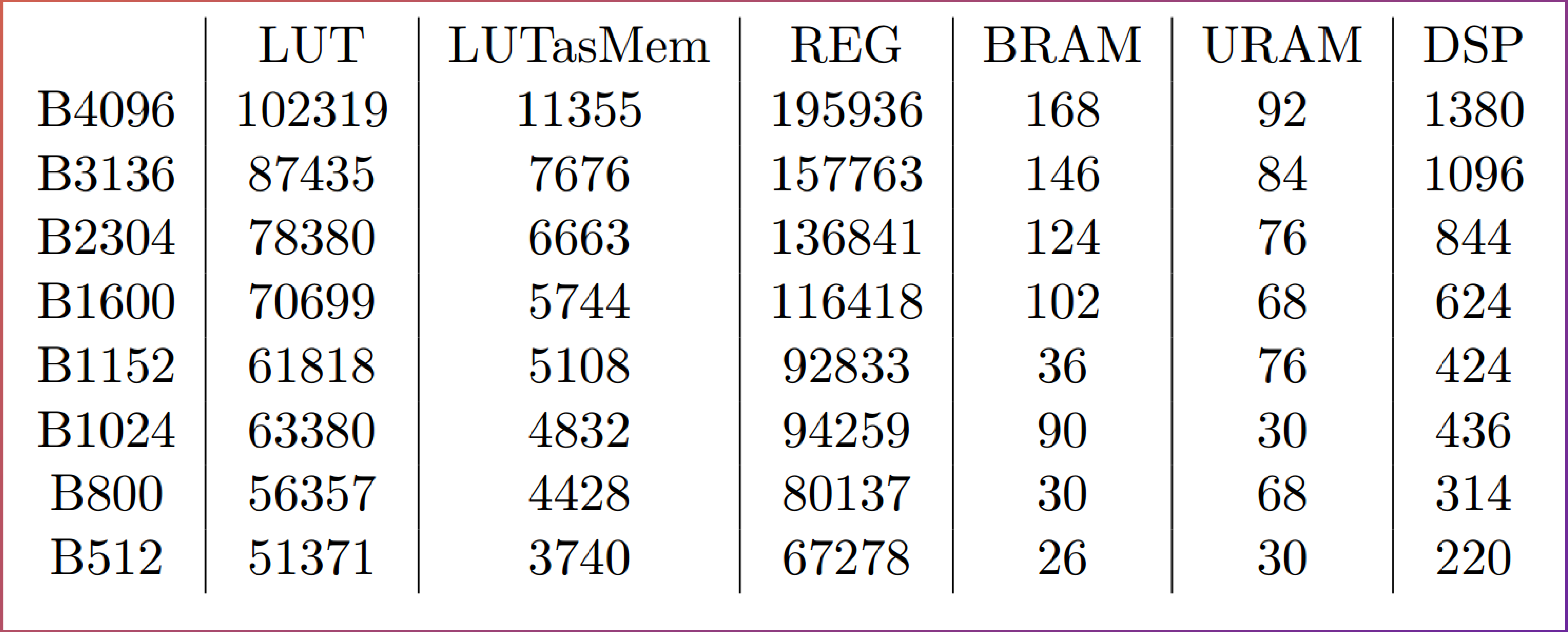
Pre-processing FPGA

Classification

Other task

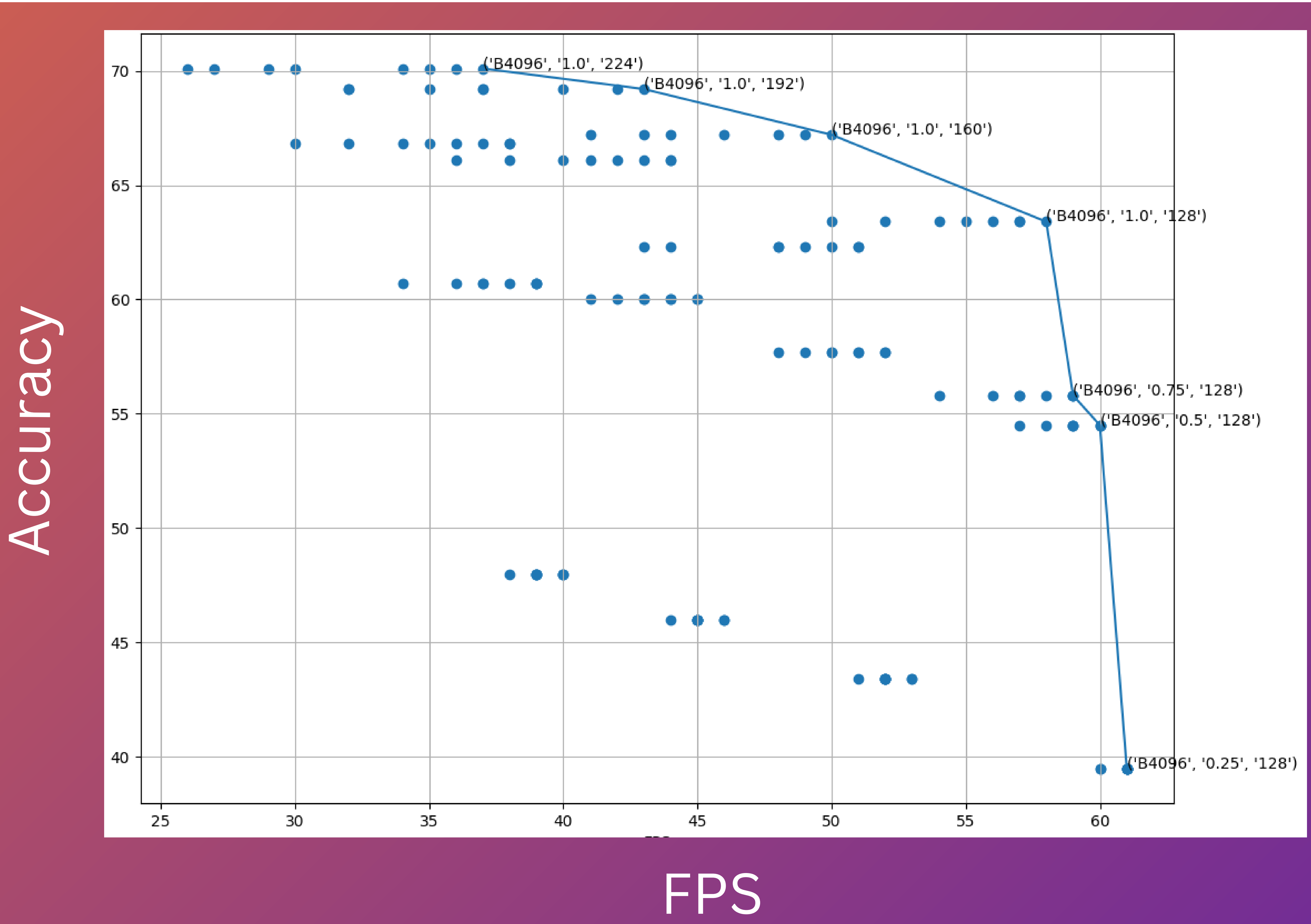
Thank you

Q&A

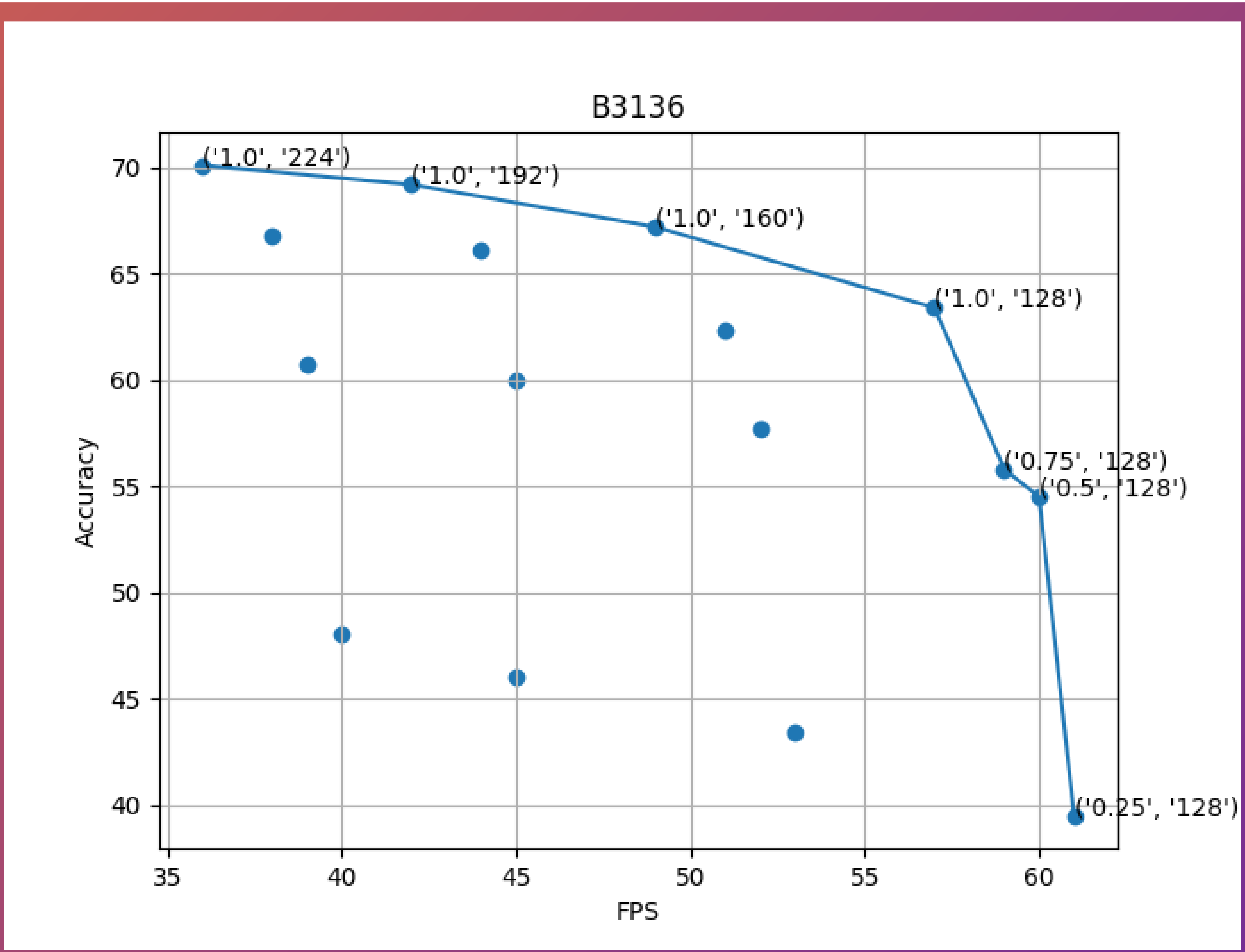


	LUT	LUTasMem	REG	BRAM	URAM	DSP
B4096	102319	11355	195936	168	92	1380
B3136	87435	7676	157763	146	84	1096
B2304	78380	6663	136841	124	76	844
B1600	70699	5744	116418	102	68	624
B1152	61818	5108	92833	36	76	424
B1024	63380	4832	94259	90	30	436
B800	56357	4428	80137	30	68	314
B512	51371	3740	67278	26	30	220

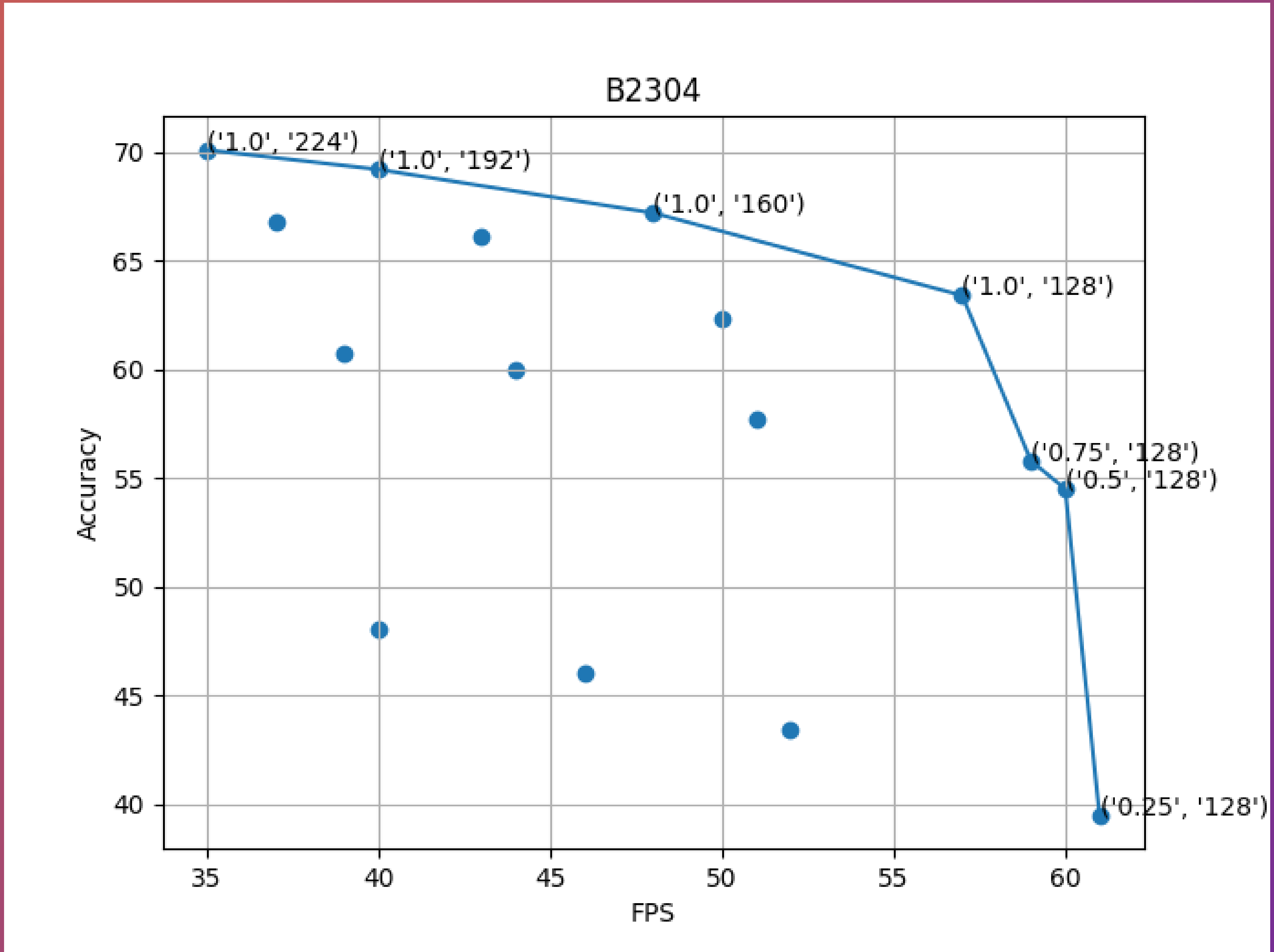
BACKUP


('B4096', '1.0', '224')
('B4096', '1.0', '192')
('B4096', '1.0', '160')
('B4096', '1.0', '128')
('B4096', '0.75', '128')
('B4096', '0.5', '128')
('B4096', '0.25', '128')
70
65
60
55
50
45
40
25
30
35
40
45
50
55
60
Accuracy
FPS

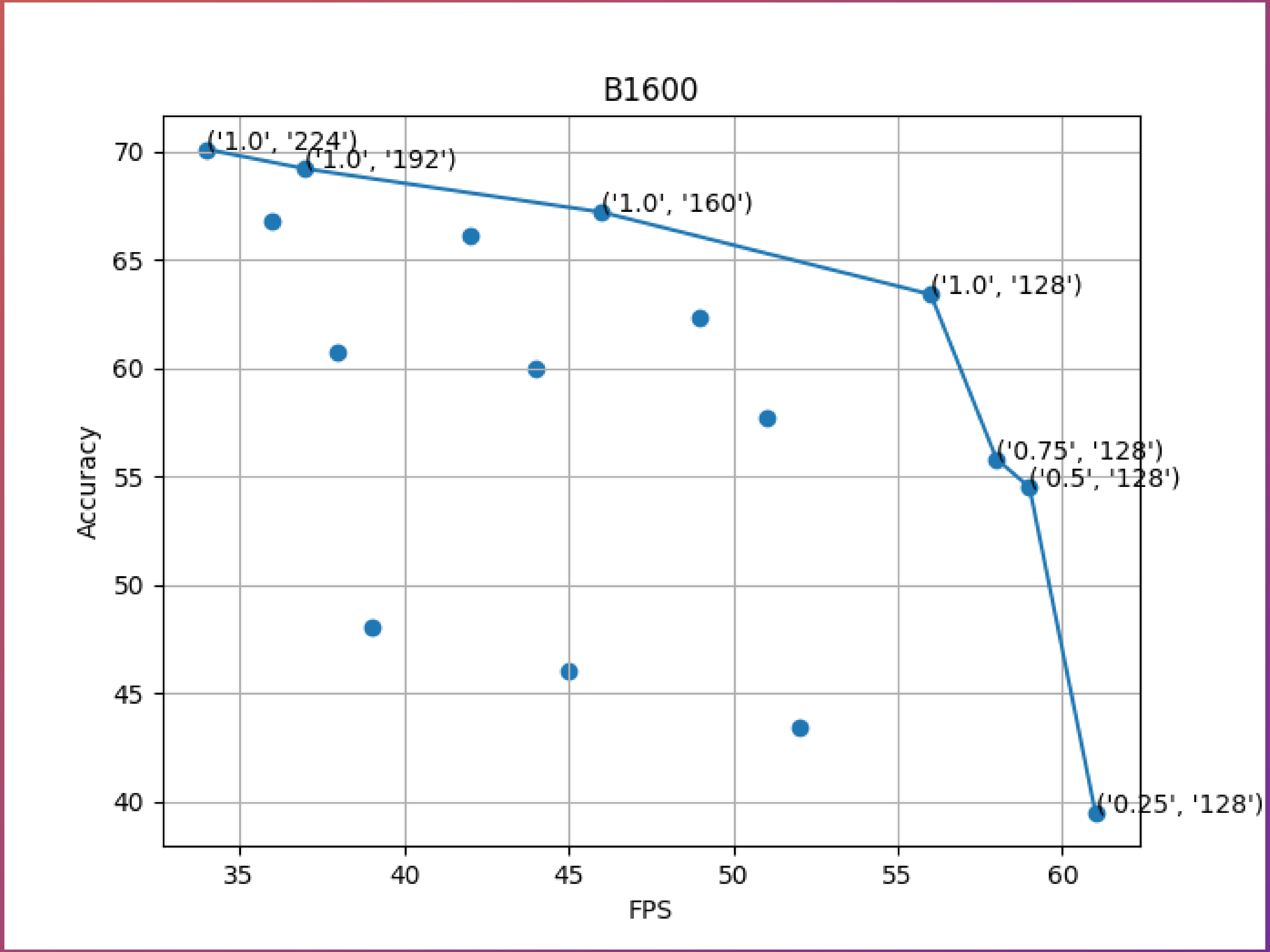
BACKUP


B3136
Accuracy
FPS
70
65
60
55
50
45
40
35
40
45
50
55
60
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')

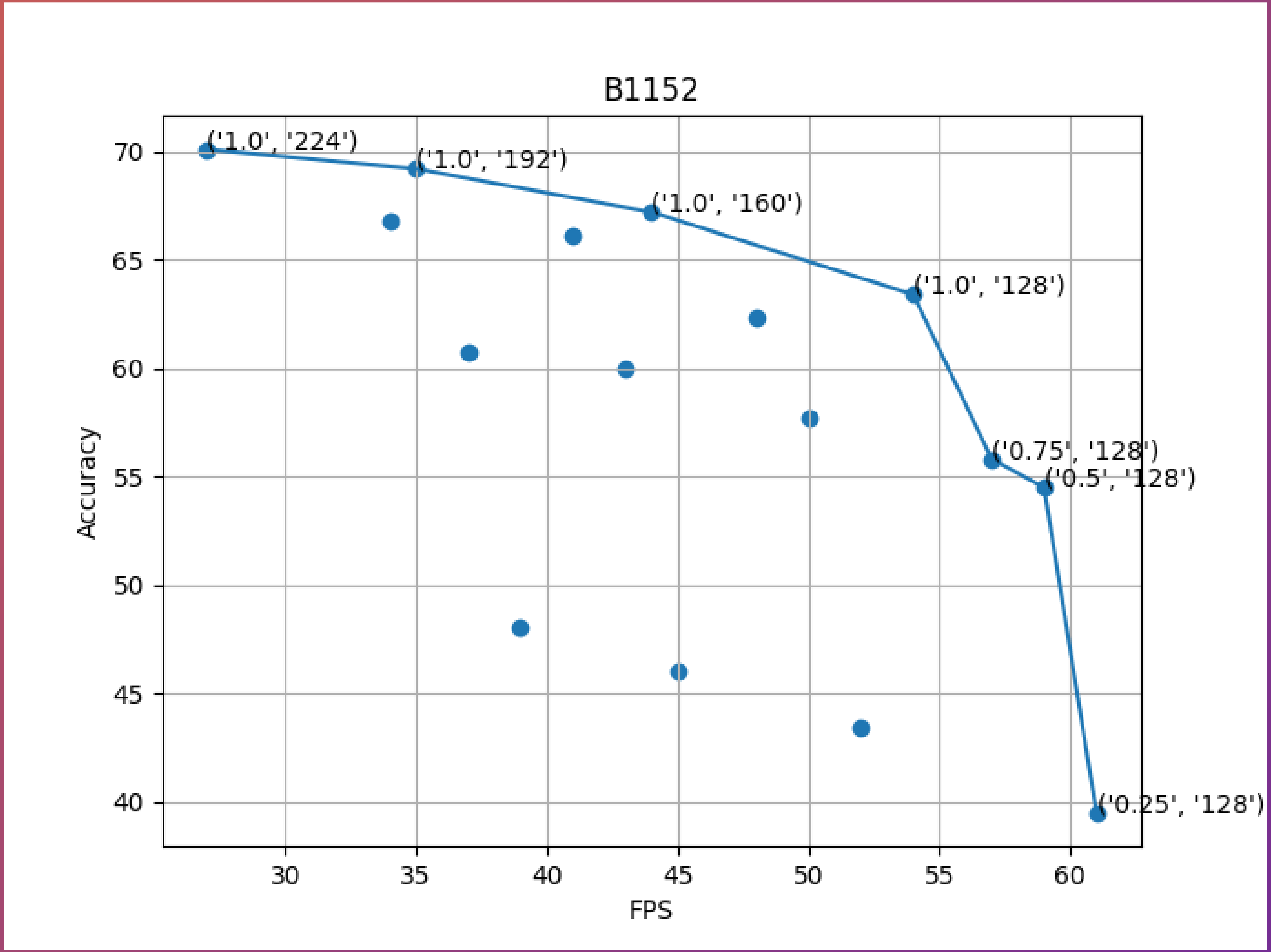
BACKUP


B2304
Accuracy
FPS
70
65
60
55
50
45
40
35
40
45
50
55
60
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')

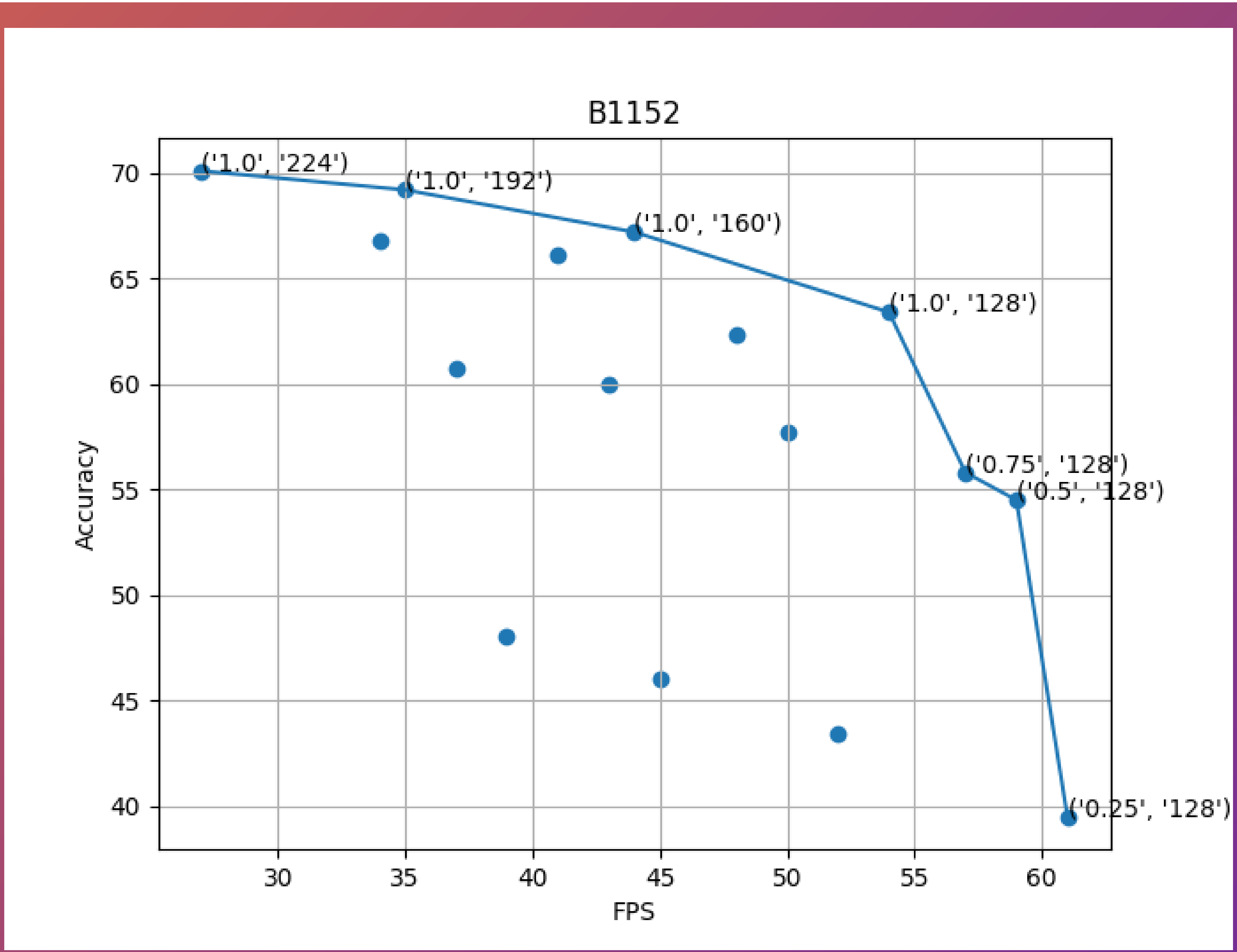
BACKUP


B1600
Accuracy
FPS
70
65
60
55
50
45
40
35
40
45
50
55
60
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')

BACKUP

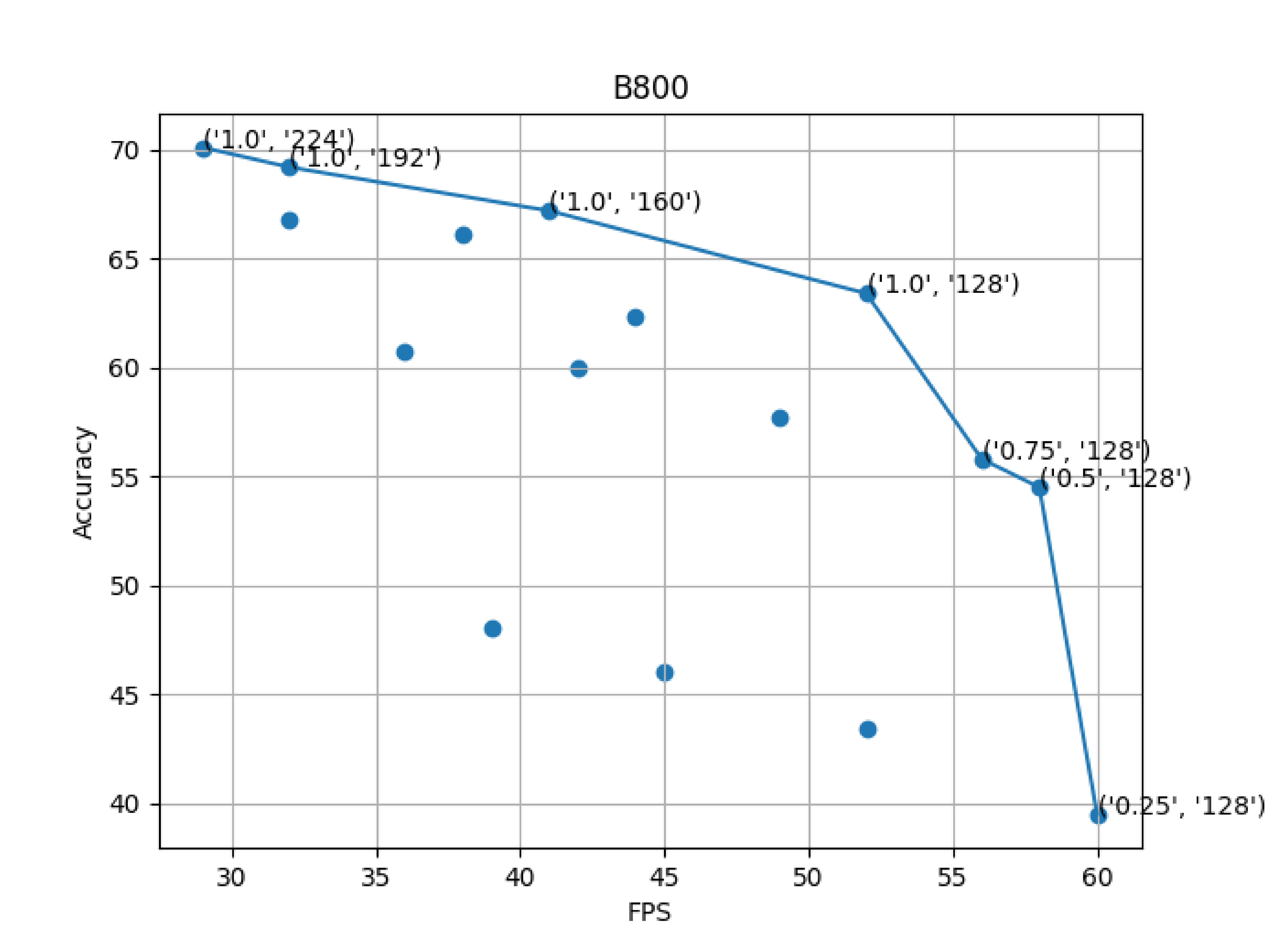

B1152
Accuracy
FPS
70
65
60
55
50
45
40
30
35
40
45
50
55
60
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')

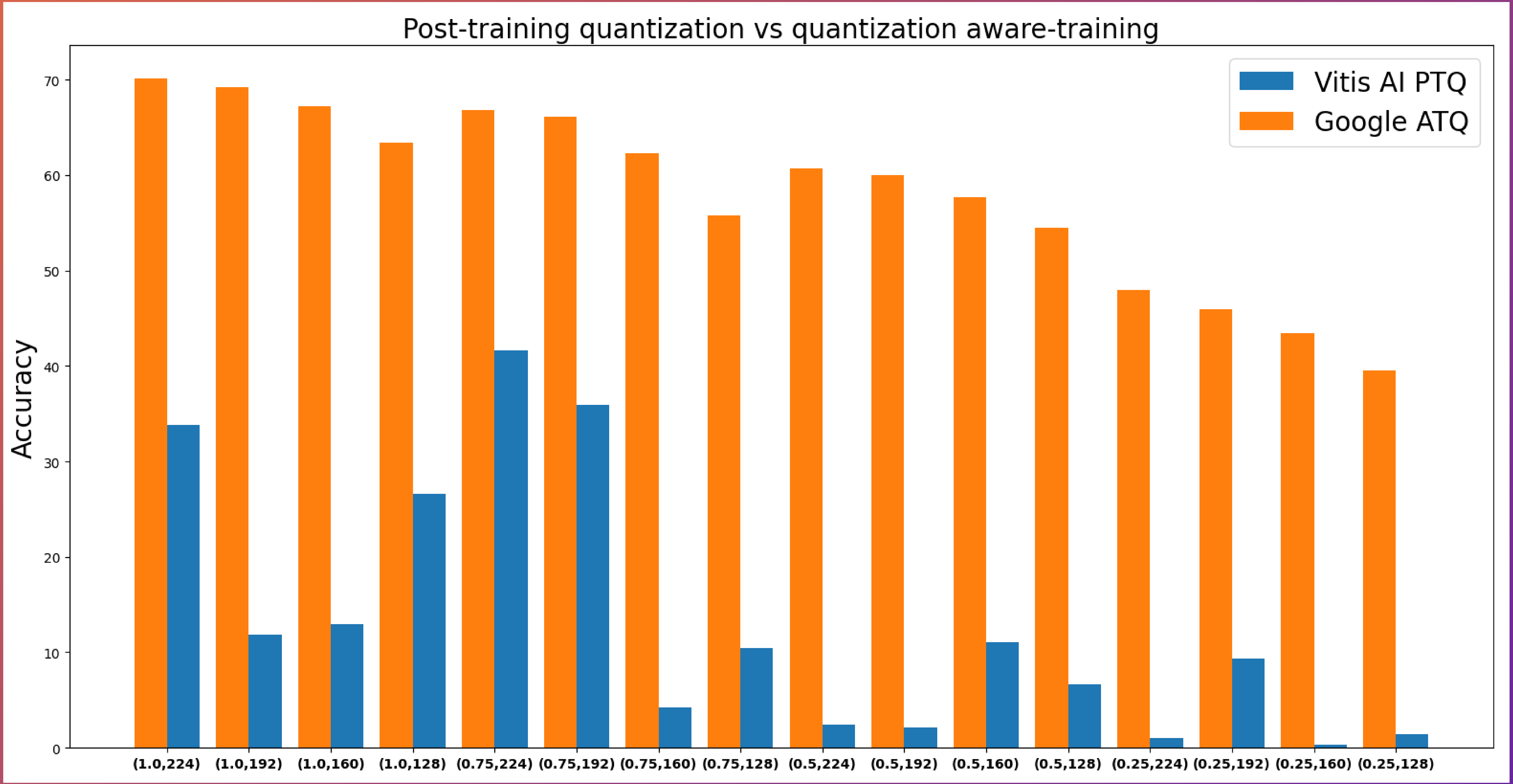
BACKUP



B1152
Accuracy
FPS
70
65
60
55
50
45
40
30
35
40
45
50
55
60
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')

BACKUP


B800
('1.0', '224')
('1.0', '192')
('1.0', '160')
('1.0', '128')
('0.75', '128')
('0.5', '128')
('0.25', '128')
Accuracy
70
65
60
55
50
45
40
FPS
30
35
40
45
50
55
60



Post-training quantization vs quantization aware-training