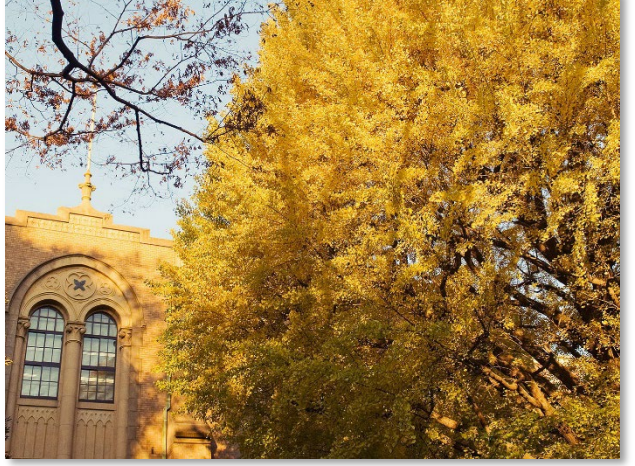
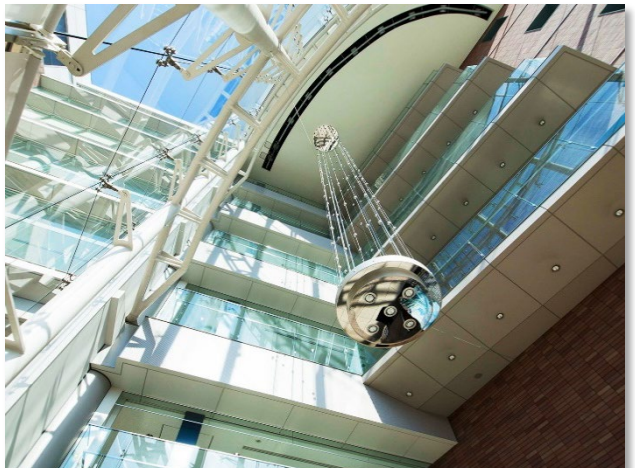


An Ultra-Fast On-Device CNN Finetuning Approach for Resource-Limited Edge Devices

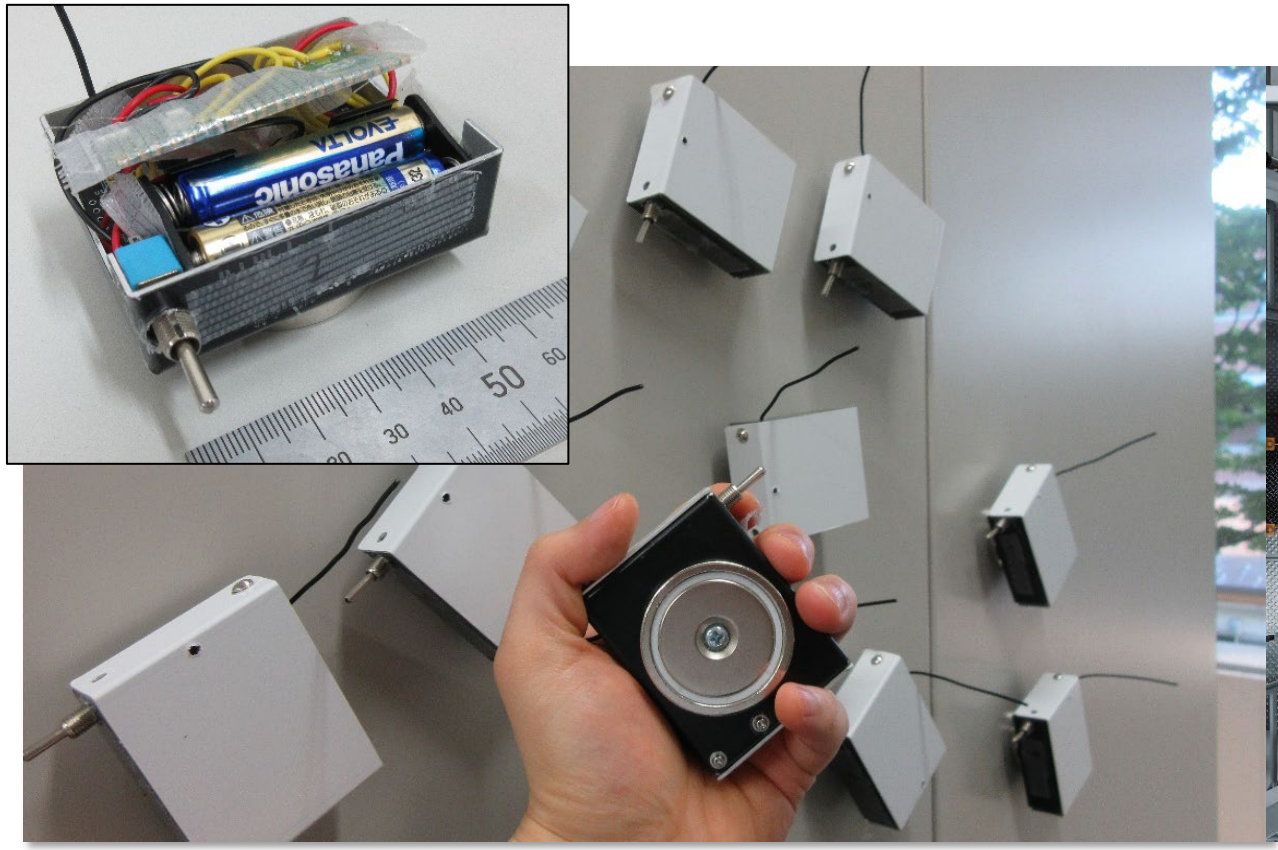
Hiroki Matsutani (Keio University, Japan)

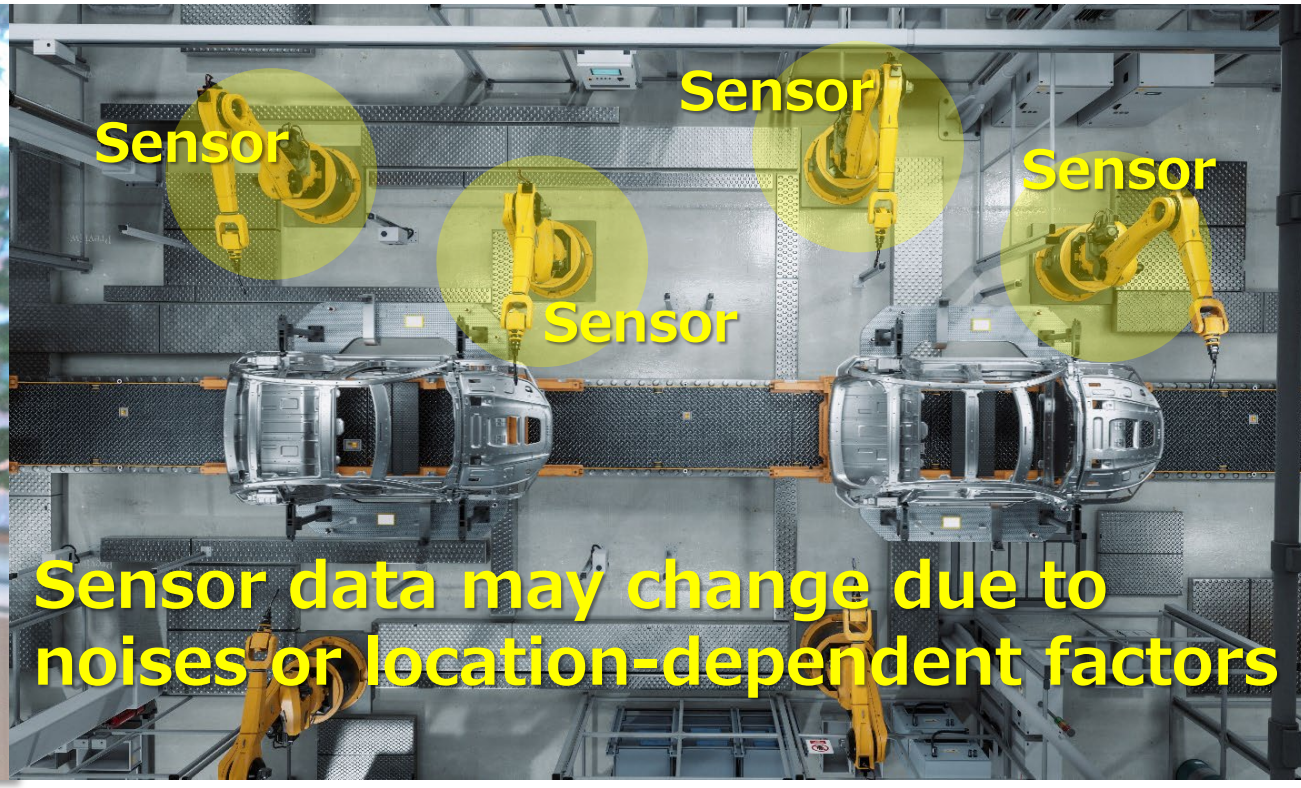



On-device learning (ODL) for IoT devices

- **Motivation for neural network training at edge side**

  Addressing the **gap** between **pretrained model** and **deployed environment** by updating the model on-device [1,2]

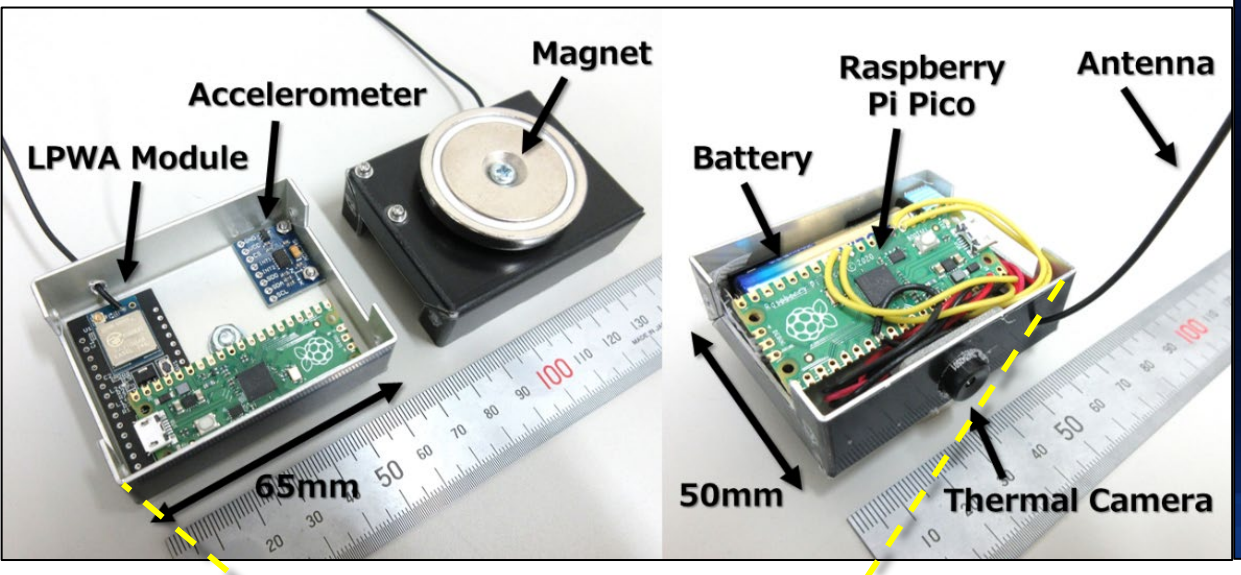


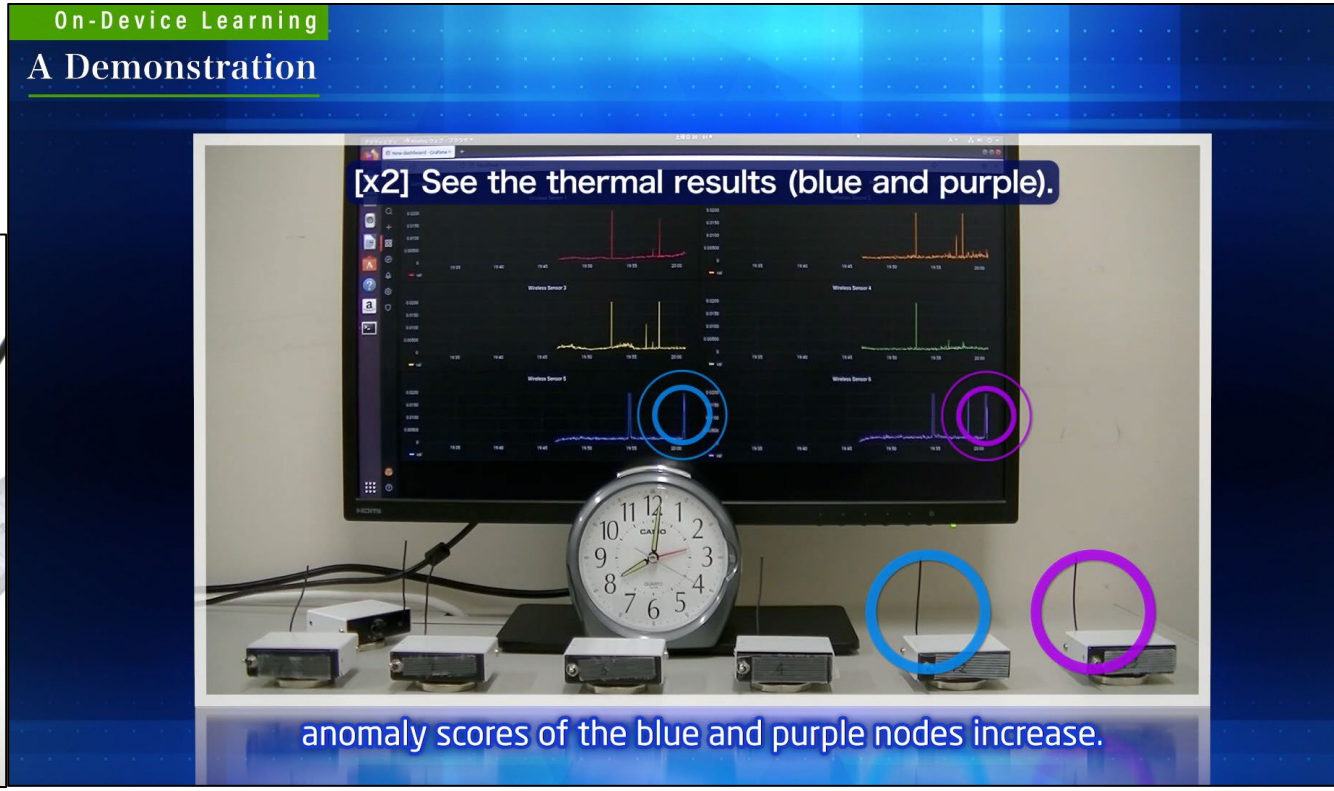


[1] Mineto Tsukada et al., "A Neural Network-Based On-device Learning Anomaly Detector for Edge Devices", IEEE Trans. on Computers (2020).
[2] Kazuki Sunaga et al., "Addressing Gap between Training Data and Deployed Environment by On-Device Learning", IEEE Micro (2023).

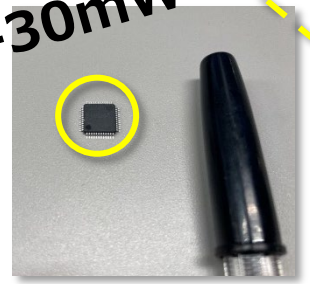
On-device learning (ODL) for IoT devices

- Lightweight **sequential training** (since 2017) [1]





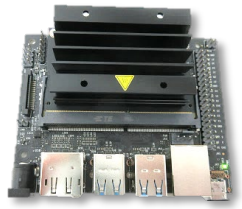
20-30mW



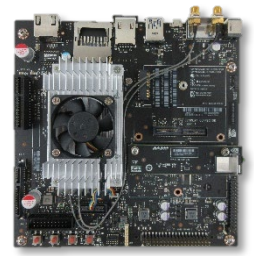
0.2-1W

2-10W

15W+



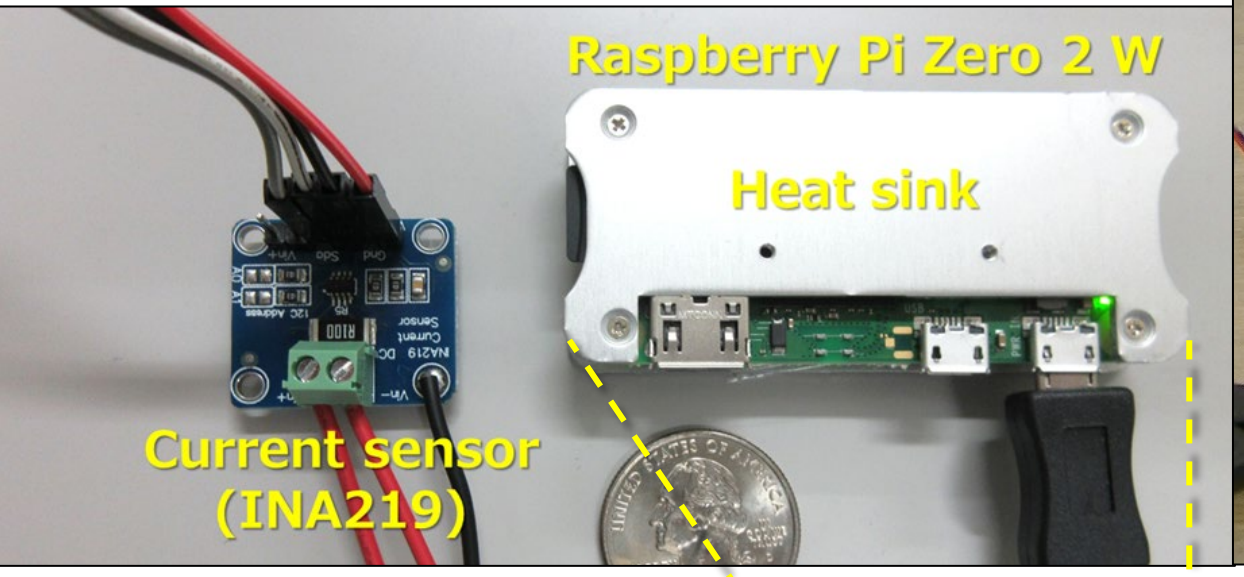
200W+

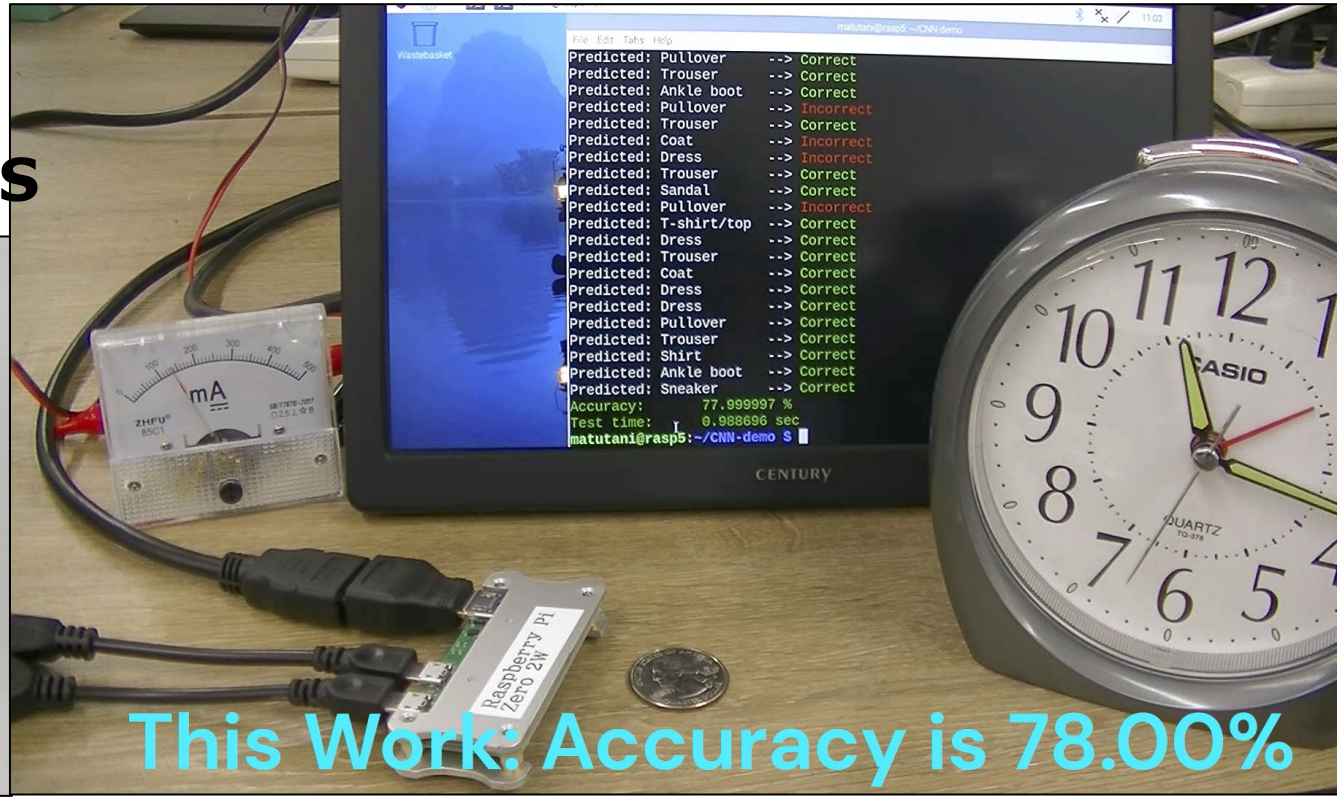


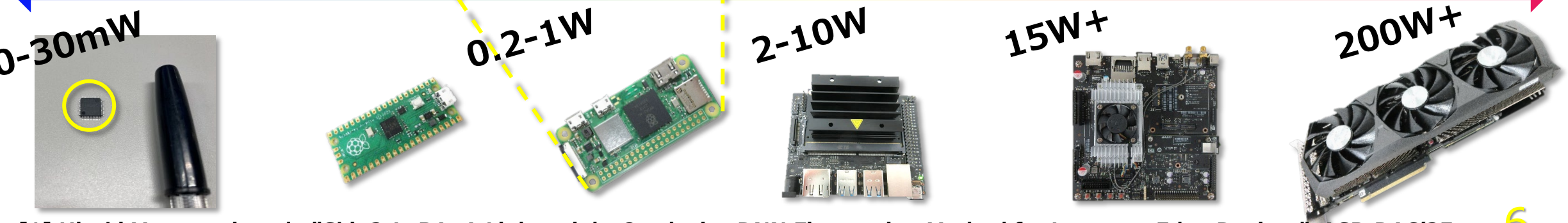
[1] Mineto Tsukada et al., "A Neural Network-Based On-device Learning Anomaly Detector for Edge Devices", IEEE Trans. on Computers (2020).

This talk: On-device finetuning for DNNs

- We focus on DNN/CNN
  including vision & LLM tasks



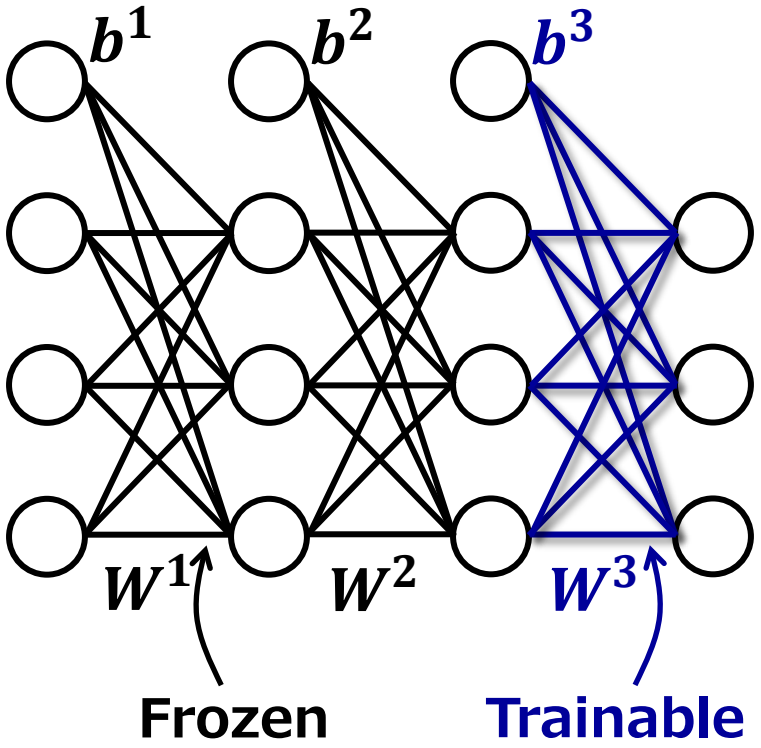




[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

Baseline finetuning methods

FT-Last [1]

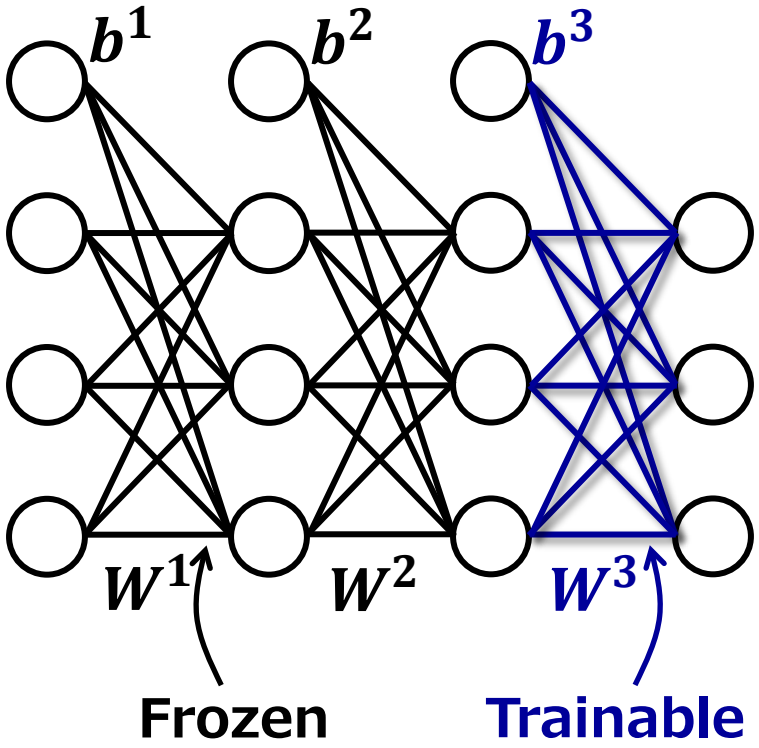


Weights of the last layer (W^3, b^3) are updated

[1] Haoyu Ren et al., "TinyOL: TinyML with Online-Learning onMicrocontrollers", IJCNN'21.

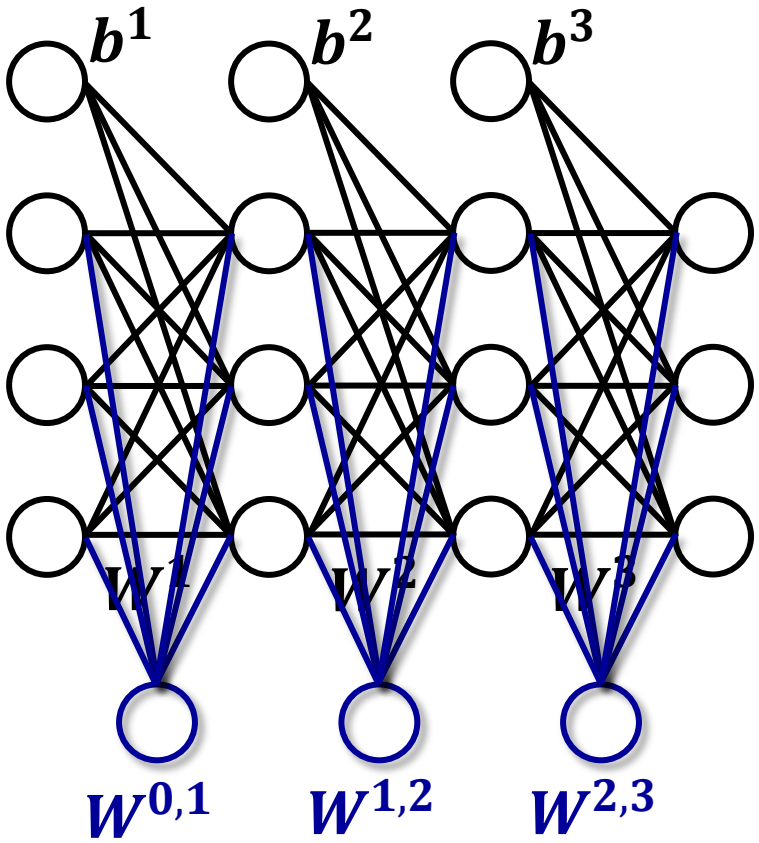
Baseline finetuning methods

FT-Last [1]



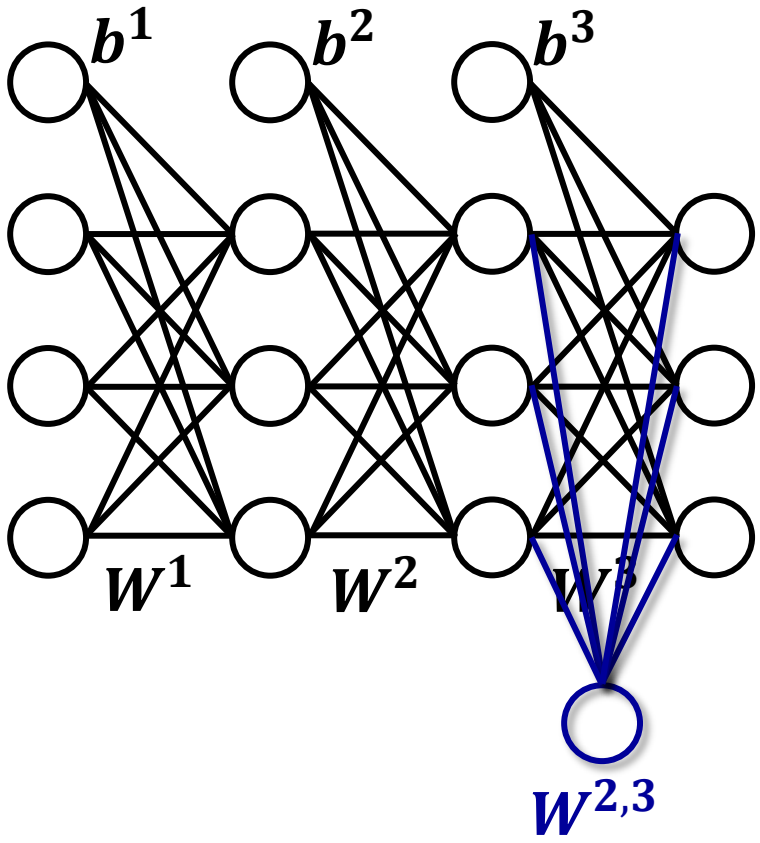
Weights of the last layer (W^3, b^3) are updated

LoRA-All [2]



Trainable adapters are attached to all layers

LoRA-Last

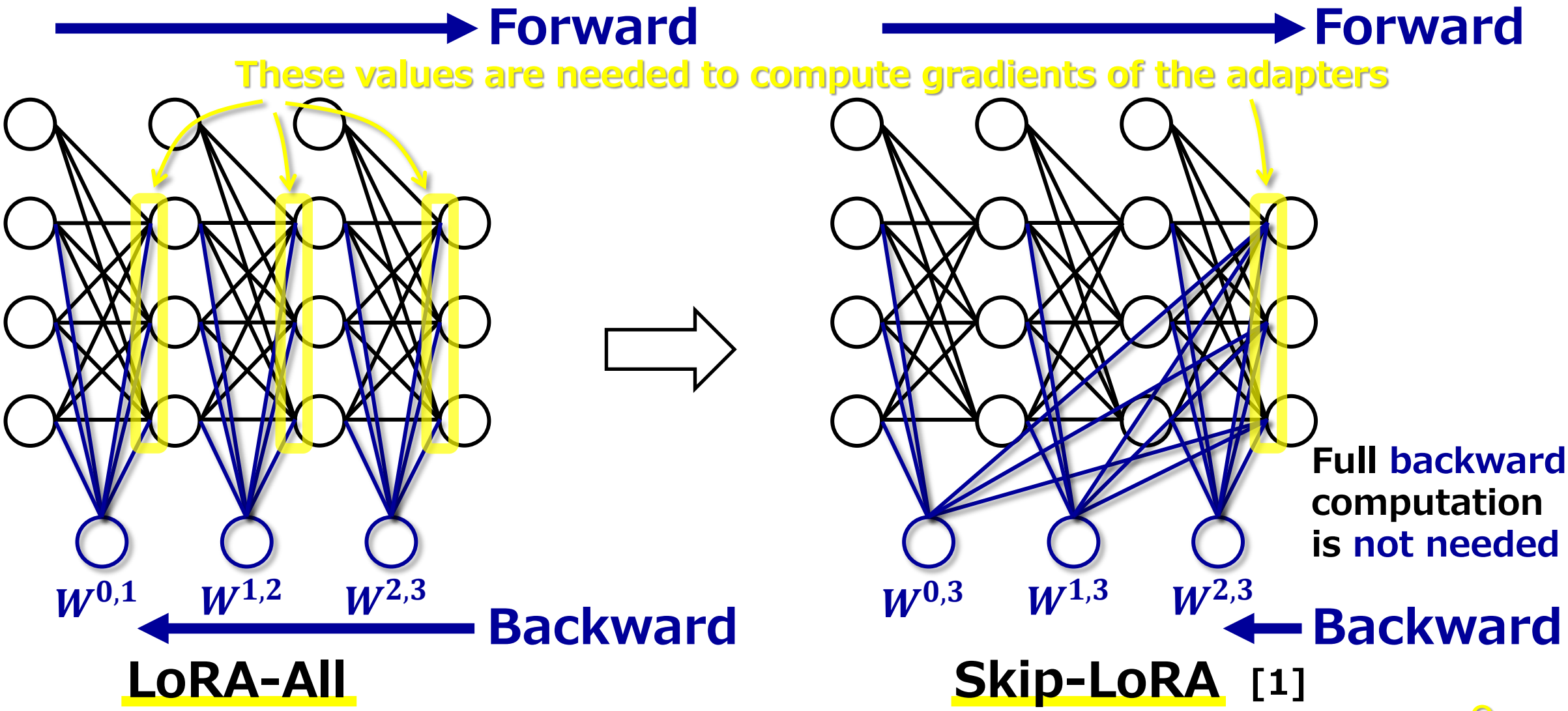


Trainable adapter is attached to the last layer

[1] Haoyu Ren et al., "TinyOL: TinyML with Online-Learning onMicrocontrollers", IJCNN'21.
[2] Edward J. Hu et al., "LoRA: Low-Rank Adaptation of Large LanguageModels", arXiv:2106.09685 (2021).

Our proposed approach: Skip-LoRA

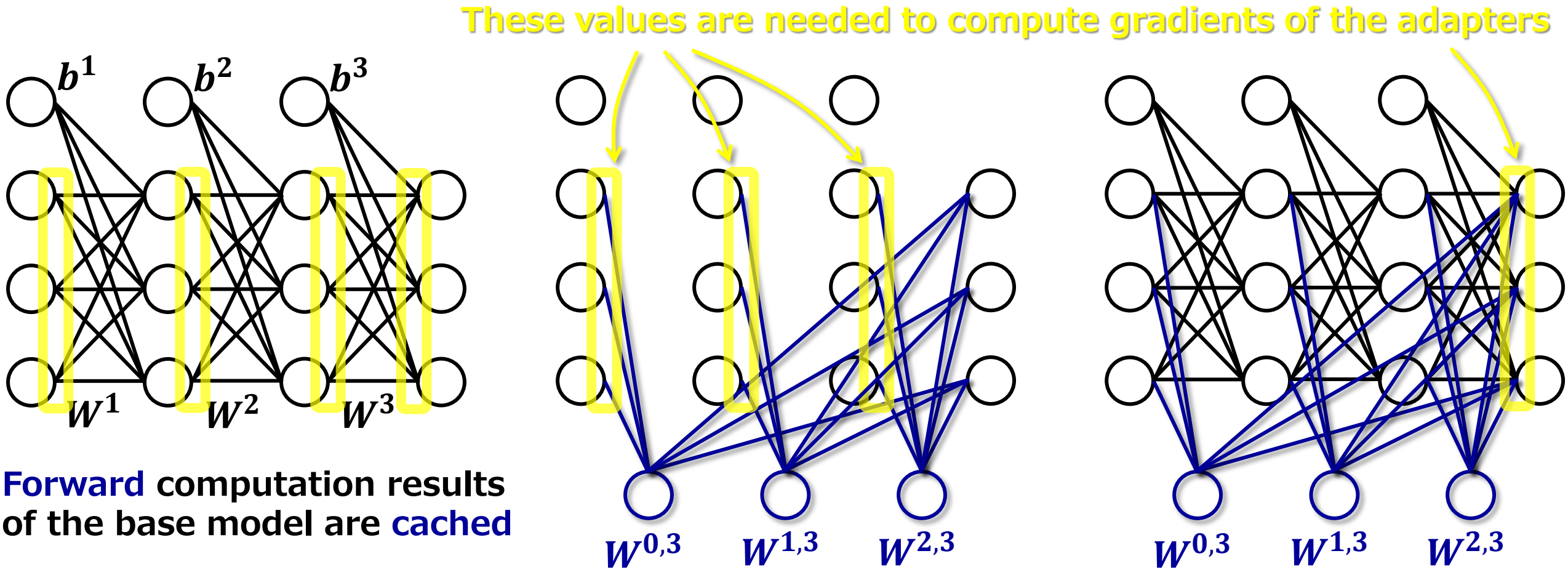
- Skip-LoRA can reduce the backward computation



[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

Our proposed approach: Skip2-LoRA

- **Skip2-LoRA** can reuse **forward** computation results



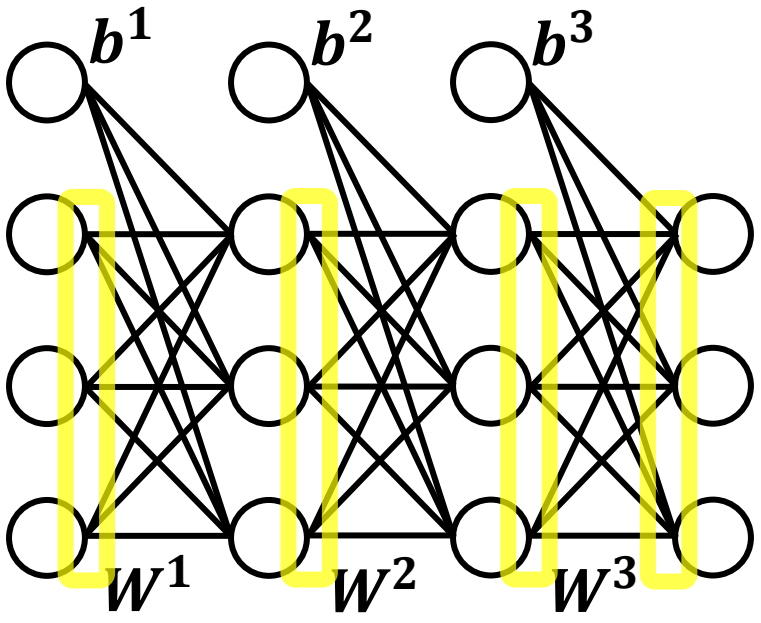
Forward computation results of the base model are **cached**

Base model + **Adapters** = **Skip-LoRA** [1]

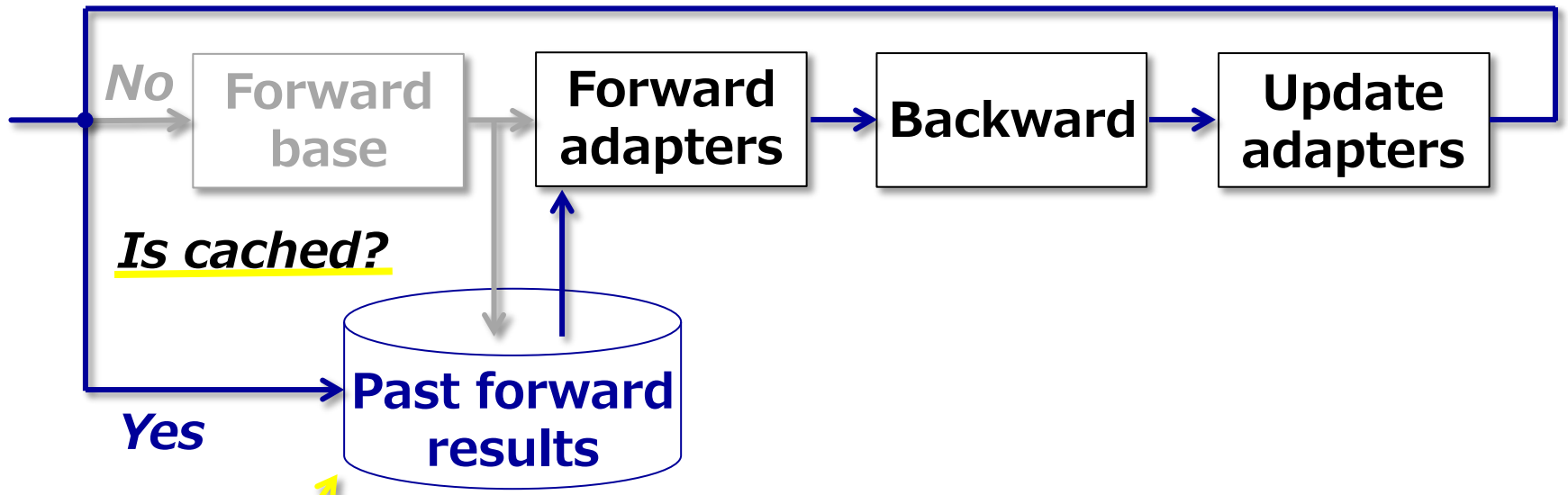
[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

Our proposed approach: Skip2-LoRA

- **Skip2-LoRA can reuse forward computation results**



(# iterations) = (# samples) / (Batch size) × (# epochs)



Forward computation results of the base model are cached

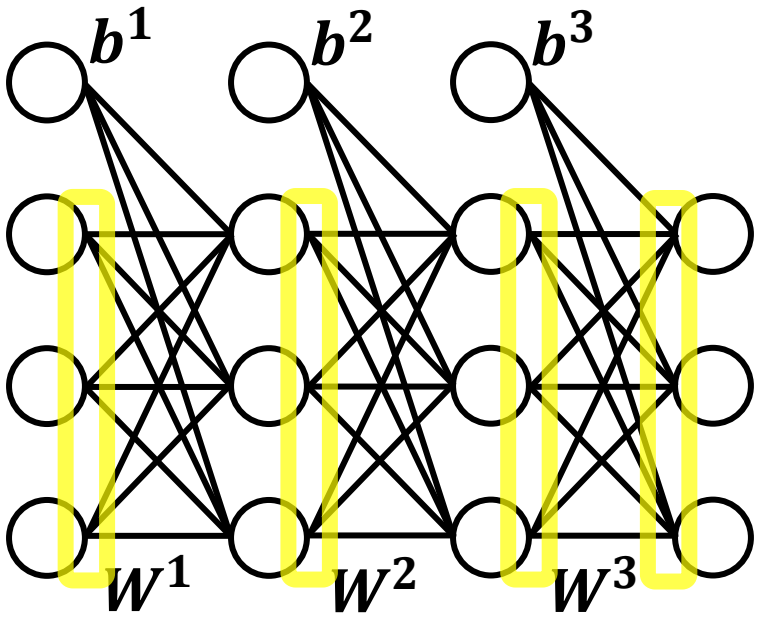
iterations is typically larger than 1;
so, we can skip most of the forward computation [1]

Base model

[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

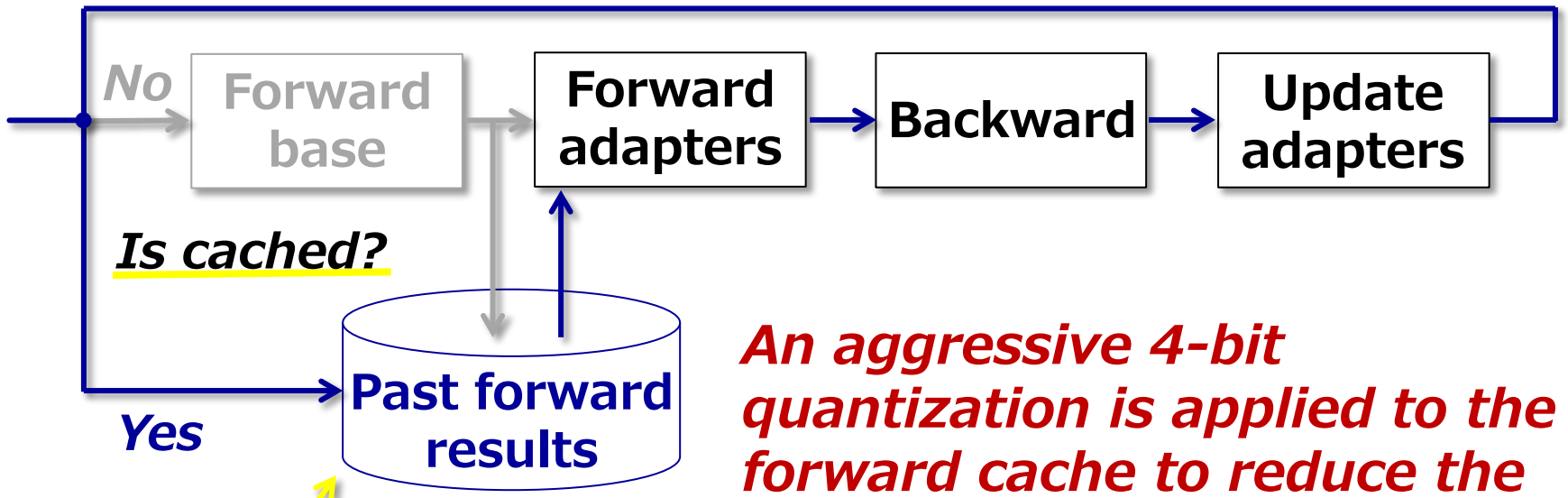
Our proposed approach: Skip2-LoRA

- **Skip2-LoRA can reuse forward computation results**



Base model

(# iterations) = (# samples) / (Batch size) × (# epochs)



Forward computation results of the base model are cached

An aggressive 4-bit quantization is applied to the forward cache to reduce the memory footprint

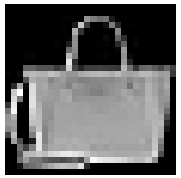
iterations is typically larger than 1;
so, we can skip most of the forward computation [1]

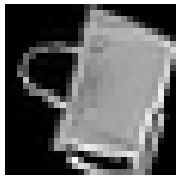
[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

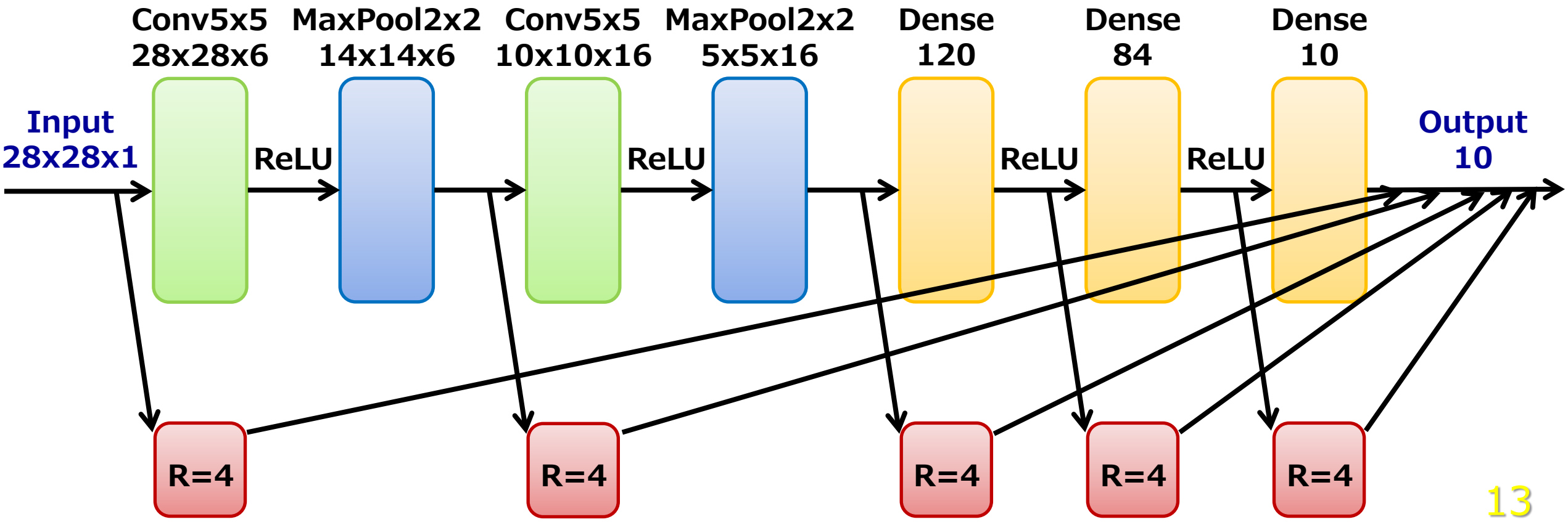
Skip2-LoRA for CNNs: Model

- Pretrained with fashion-MNIST
- Finetuned with Rotated fashion-MNIST (75 deg)
- Tested with Rotated fashion-MNIST (75 deg)

This gap reduces accuracy

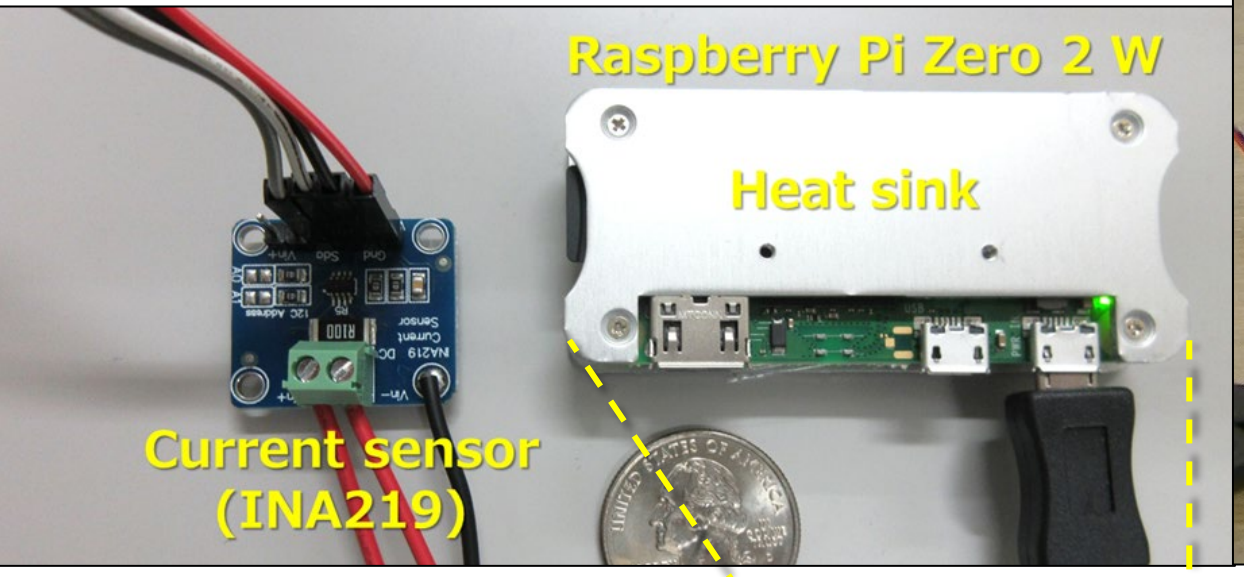


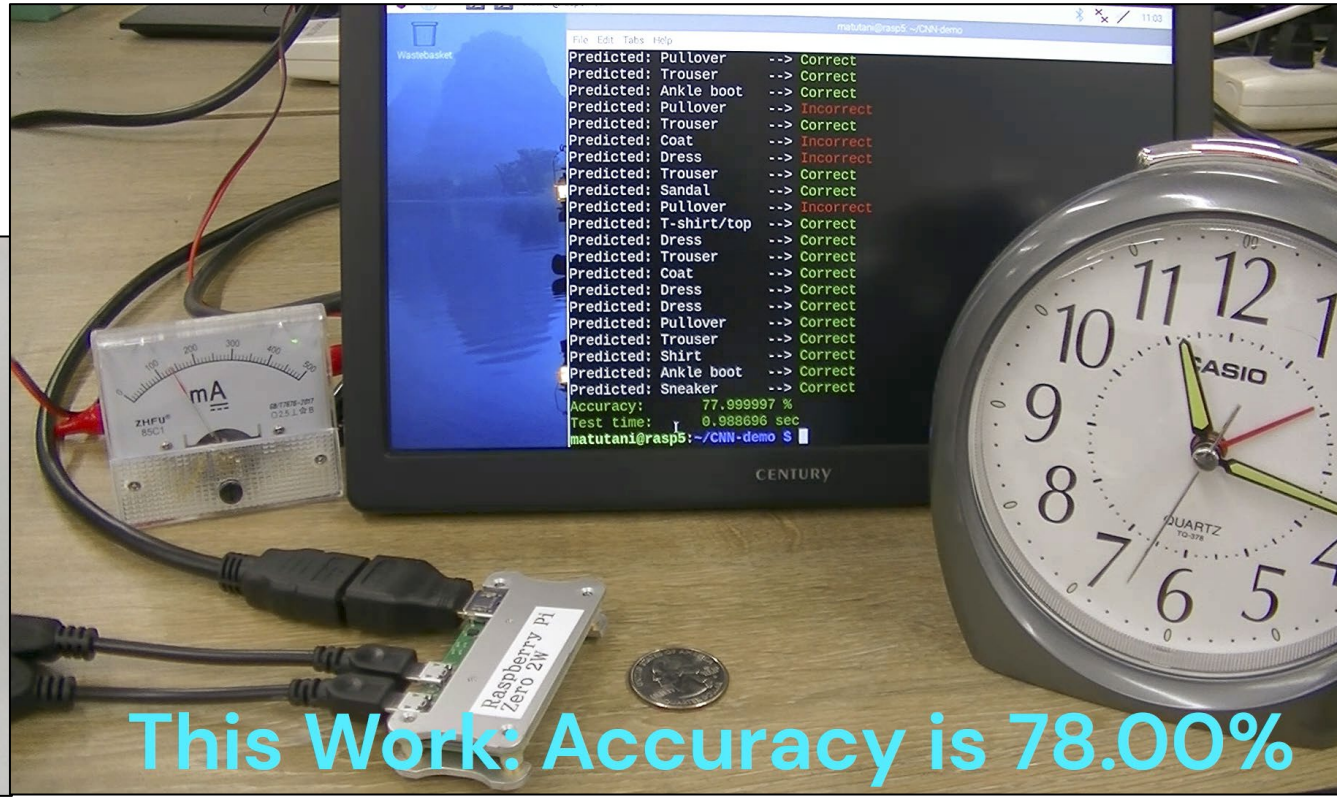


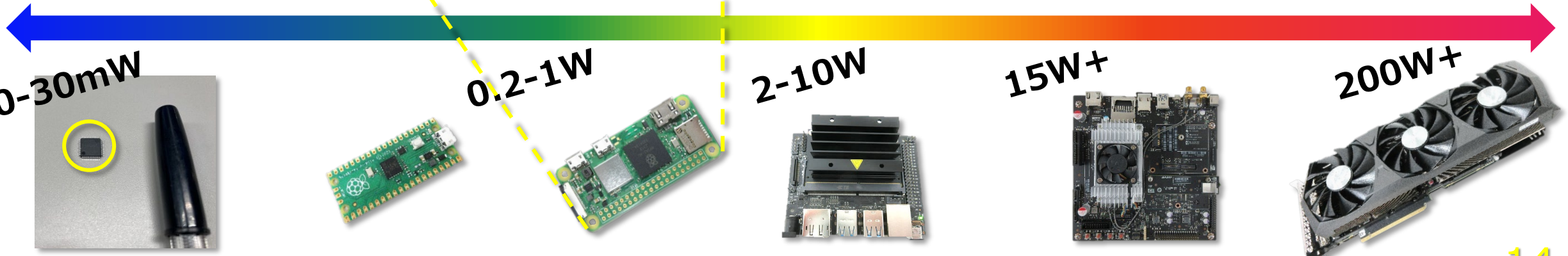


Skip2-LoRA for CNNs: Raspberry Pi

- **Raspberry Pi Zero 2W**
  ARM Cortex-A53 @1GHz







[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

Skip2-LoRA for CNNs: Results

- In this work, Skip2-LoRA [1] is applied to CNNs
- An aggressive 4-bit quantization is applied to the forward cache to reduce the memory footprint

Model	Accuracy	FT time @RPZ2	Cache size
No Finetuning (FT)	9.18 %		
FT-Last	60.94 %	18.09 sec	
LoRA-Last	53.81 %	18.09 sec	
LoRA-All	75.59 %	114.15 sec	
Skip-LoRA	73.54 %	19.84 sec	
Skip2-LoRA	73.54 %	3.90 sec	7,336 kB
Quant Skip2-LoRA	74.02 %	4.27 sec	1,036 kB

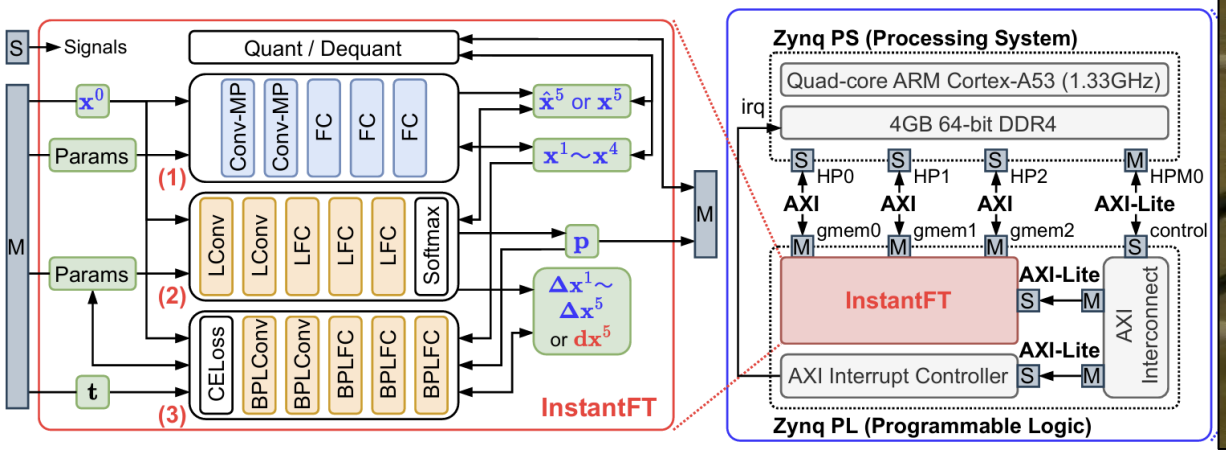
Raspberry Pi Zero 2W
(aka $15 computer)

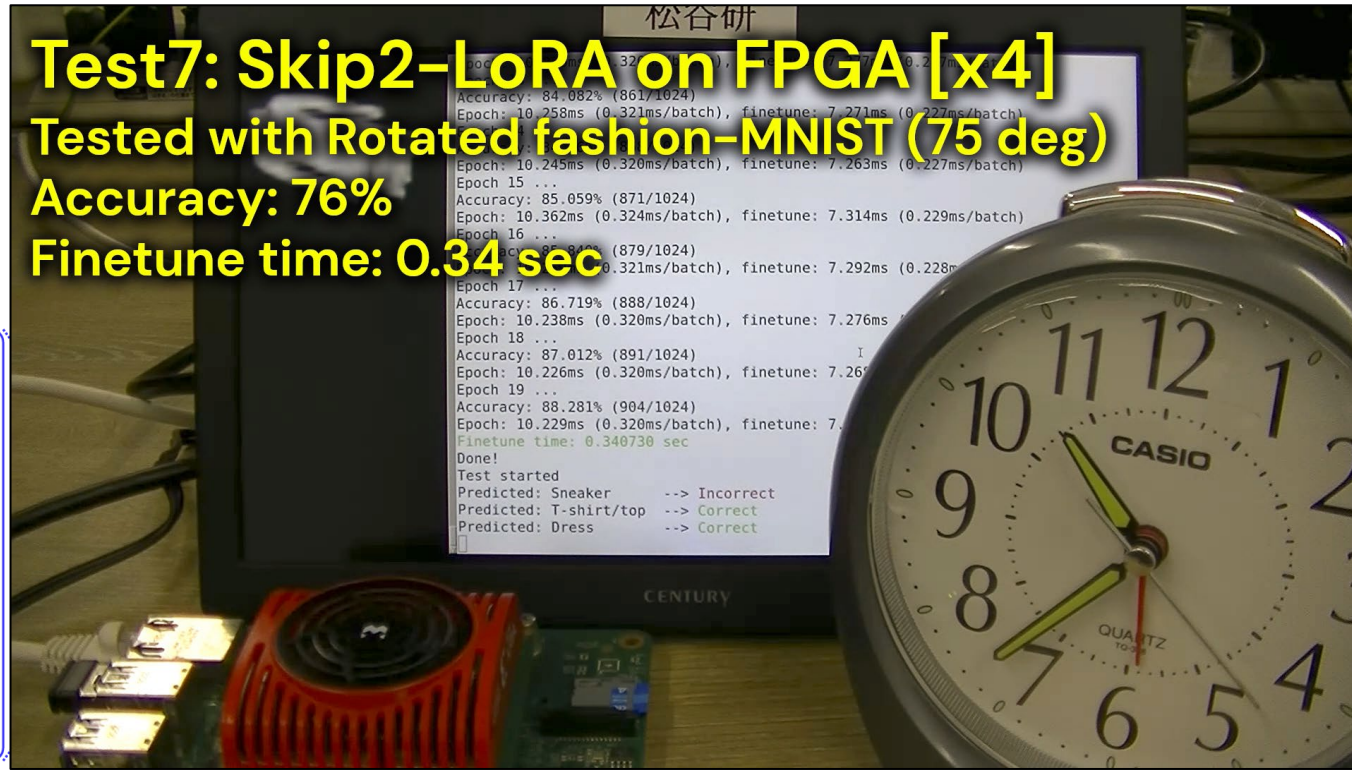
*Number of FT samples: 1024, Number of epochs for FT: 10

[1] Hiroki Matsutani et al., "Skip2-LoRA: A Lightweight On-device DNN Fine-tuning Method for Low-cost Edge Devices", ASP-DAC'25.

Skip2-LoRA for CNNs: Low-cost FPGA

- **Low-cost FPGA board**
  AMD KV260 Starter Kit
  Finetune time: 0.34 sec





[1] Keisuke Sugiura et al., "InstantFT: An FPGA-Based Runtime Subsecond Fine-tuning of CNN Models", arXiv:2506.06505 (2025).

Skip2-LoRA for CNNs: Summary

- **Ultra-fast on-device CNN finetuning for resource-limited edge devices**

  Our approach: algorithm and architecture codesign

Step 1:

New connection pattern

114.15 sec → 19.84 sec

Step 2:

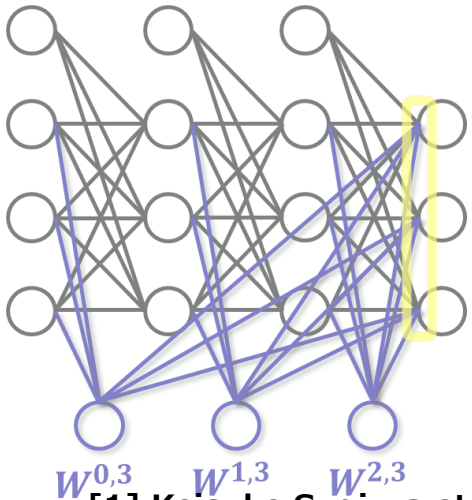
Reusing computation results

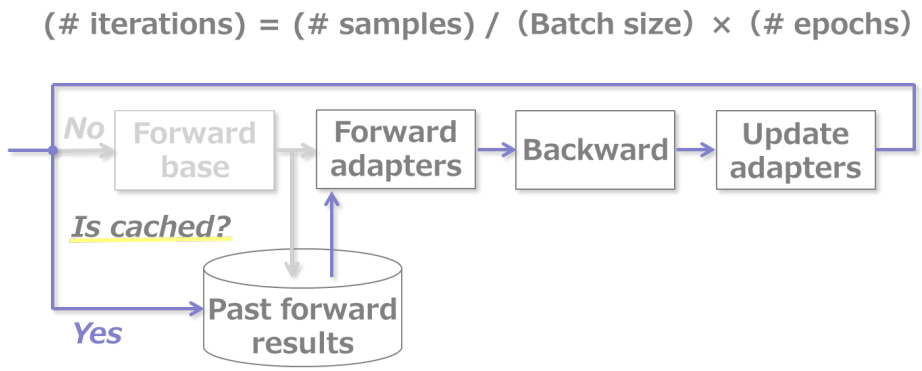
19.84 sec → 4.27 sec

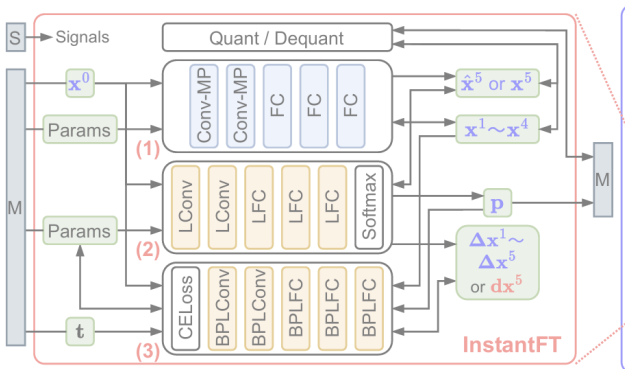
Step 3:

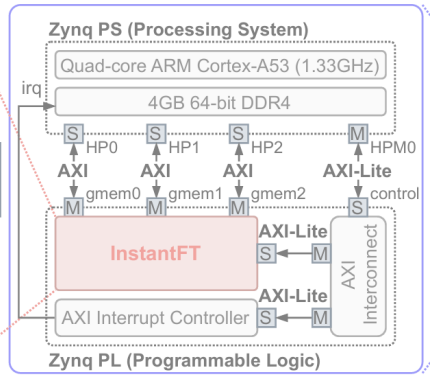
FPGA-based acceleration

4.27 sec → 0.34 sec









[1] Keisuke Sugiura et al., "InstantFT: An FPGA-Based Runtime Subsecond Fine-tuning of CNN Models", arXiv:2506.06505 (2025).