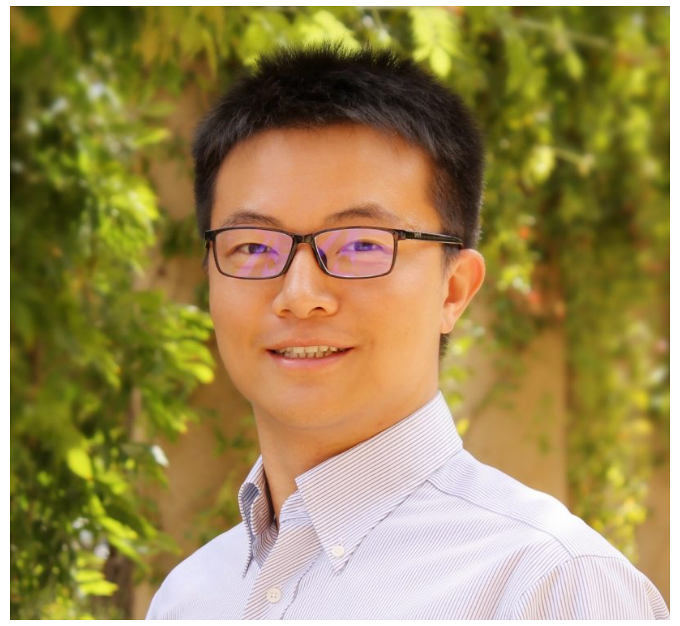
MIT HAN LAB

Hardware, **A**I, and **N**eural-nets
open source, co-design
http://github.com/mit-han-lab

Model Compression for Efficient AI Computing

From TinyML to LargeLM

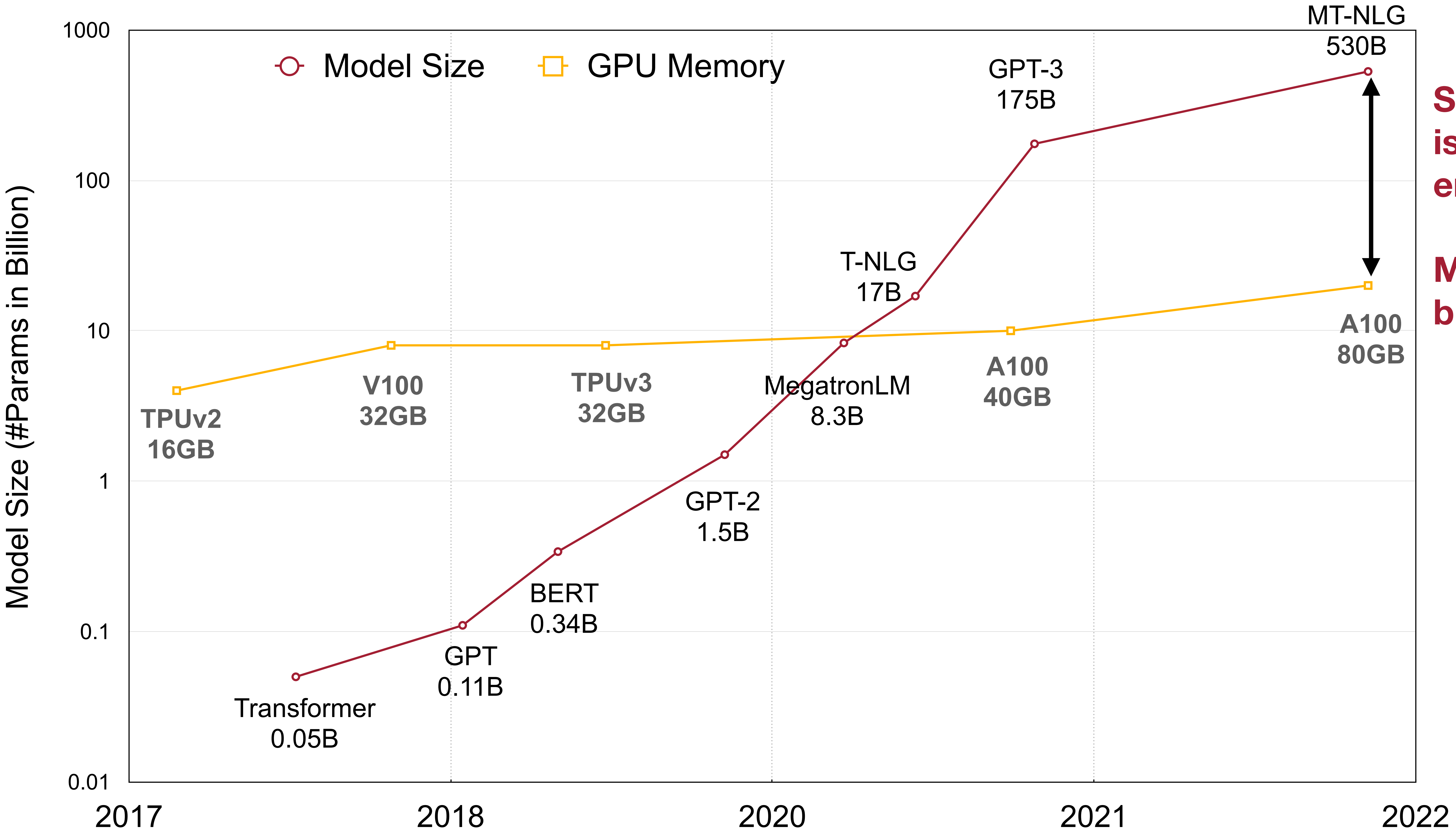


Song Han
MIT
songhan.mit.edu
tinyml.mit.edu
@SongHan_MIT

Generated by Midjourney

Model Compression

Bridges the Gap between the Supply and Demand of Computation



Specialized hardware is important but not enough.

Model compression bridges the gap.

Model Compression

Bridges the Gap between the Supply and Demand of Computation

AI Application
(demand of computation)

New Primitive

Hardware-aware NAS

Fine-Grained Pruning

CoarseGrained / Structured Pruning

Post-Training Quantization

Quantization-Aware Training

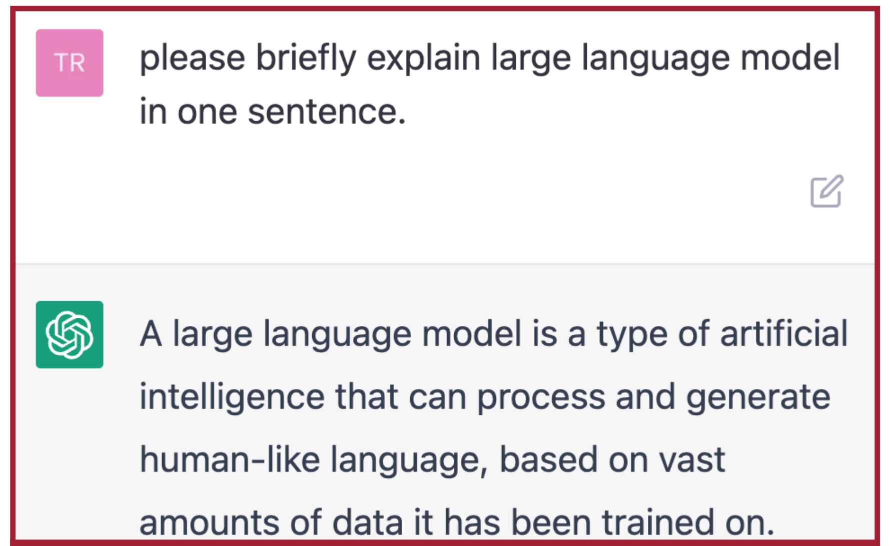
Distillation

Augmentation

AI Hardware
(supply of computation)

Model Compression

Applications



TR please briefly explain large language model in one sentence.

A large language model is a type of artificial intelligence that can process and generate human-like language, based on vast amounts of data it has been trained on.

Large Language Model

Generative AI

Advanced Driver Assistance System

TinyML

Techniques

Hardware-aware NAS

Pruning & Sparsity

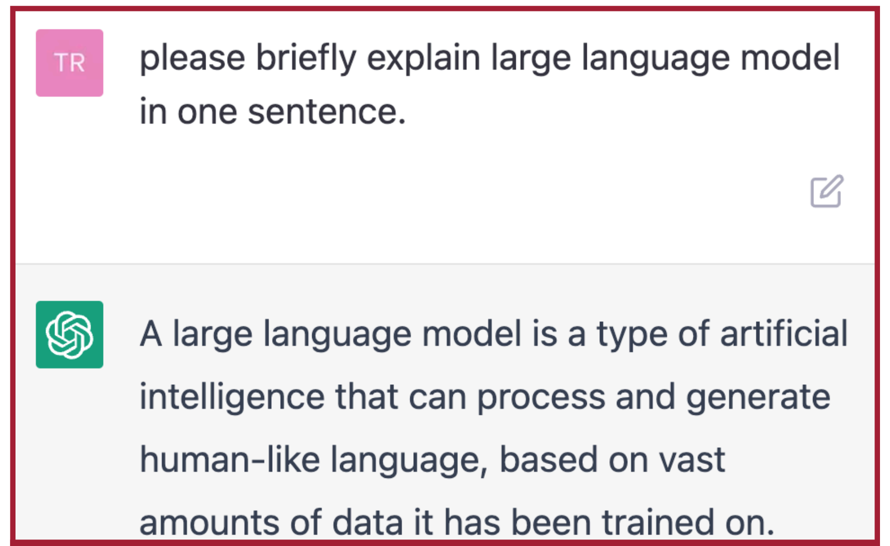
Quantization

Distillation

New Primitive

Same Principle, Diverse Applications

Applications



TR please briefly explain large language model in one sentence.

A large language model is a type of artificial intelligence that can process and generate human-like language, based on vast amounts of data it has been trained on.

Large Language Model

Generative AI

Advanced Driver Assistance System

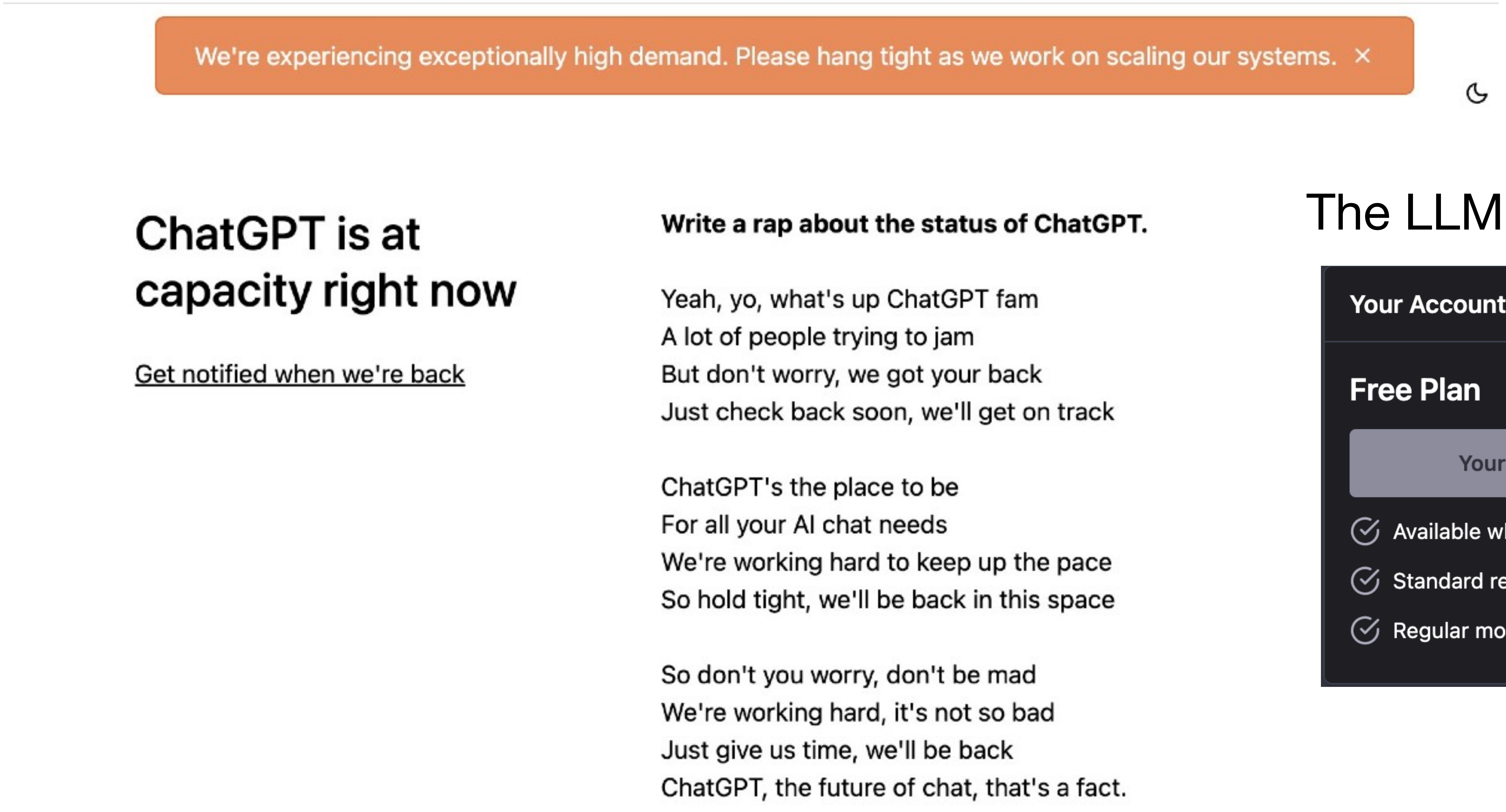
TinyML

Techniques

- Hardware-aware NAS
- Pruning & Sparsity
- Quantization
- Distillation
- New Primitive

Efficient Large Language Models

Reducing LLM Serving Cost and Accelerating Inference



We're experiencing exceptionally high demand. Please hang tight as we work on scaling our systems. ×

ChatGPT is at capacity right now

Get notified when we're back

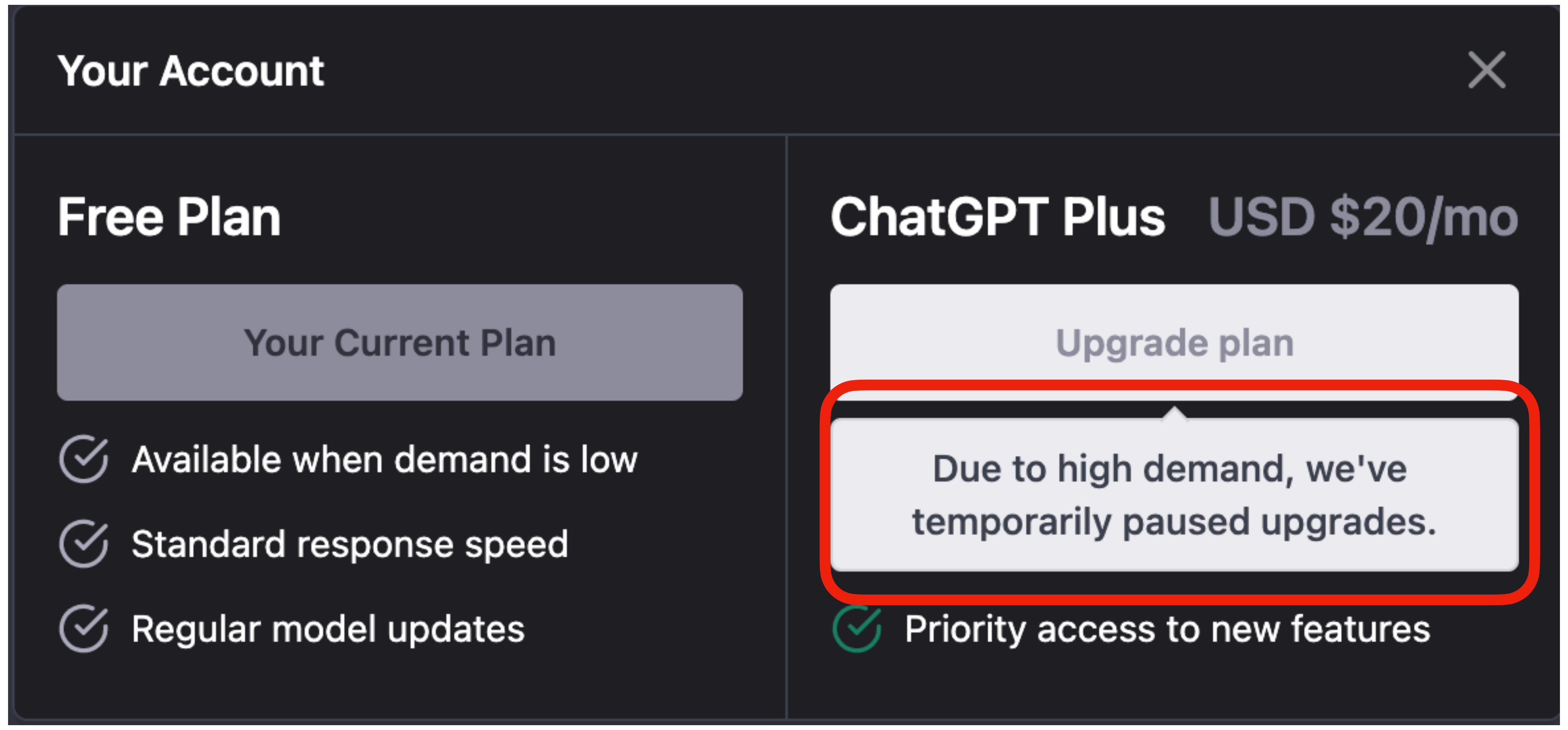
Write a rap about the status of ChatGPT.

Yeah, yo, what's up ChatGPT fam
A lot of people trying to jam
But don't worry, we got your back
Just check back soon, we'll get on track

ChatGPT's the place to be
For all your AI chat needs
We're working hard to keep up the pace
So hold tight, we'll be back in this space

So don't you worry, don't be mad
We're working hard, it's not so bad
Just give us time, we'll be back
ChatGPT, the future of chat, that's a fact.

The LLM serving costs are extremely high



Your Account ×

Free Plan

Your Current Plan

- ✓ Available when demand is low
- ✓ Standard response speed
- ✓ Regular model updates

ChatGPT Plus USD $20/mo

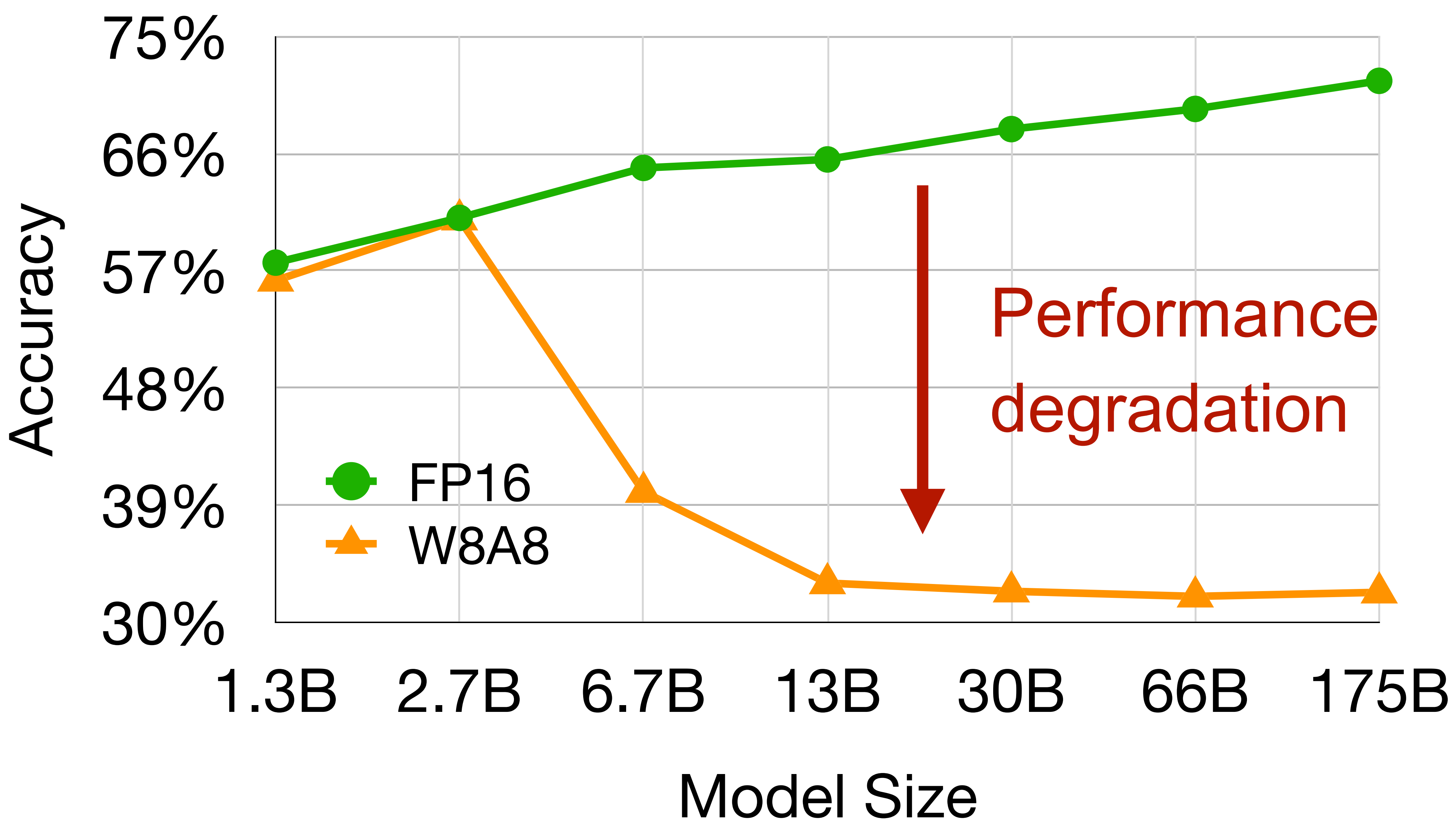
Upgrade plan

Due to high demand, we've temporarily paused upgrades.

- ✓ Priority access to new features

Quantization cut the model size by half, but…

Existing Quantization Method is Slow or Inaccurate



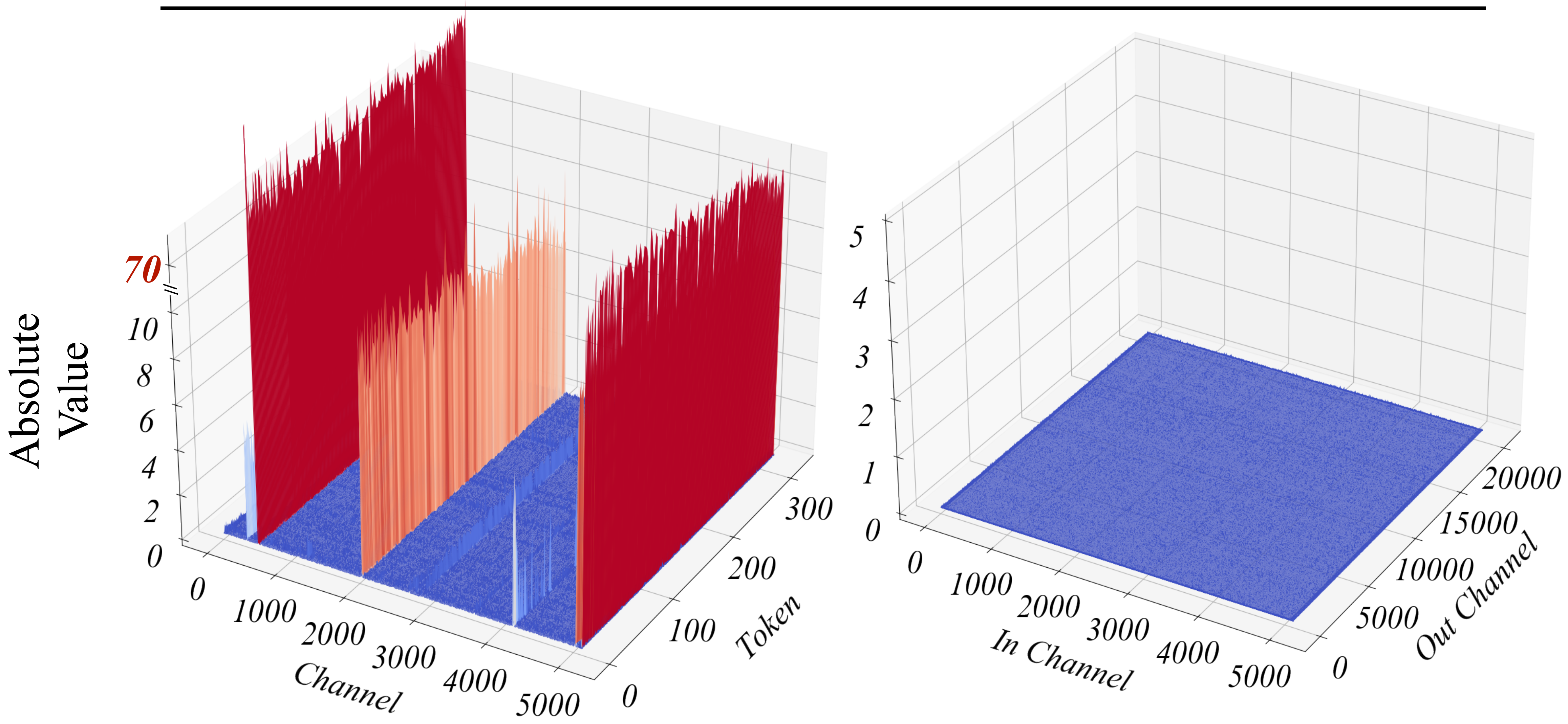
- W8A8 quantization has been an industrial standard for CNNs, but not LLM. Why?
- Systematic outliers emerge in **activations** when we scale up LLMs beyond 6.7B. Traditional CNN quantization methods will destroy the accuracy.

LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale (Dettmers *et al.*, 2022)

SmoothQuant

Smoothing activation to reduce quantization error

Original



Activation
Hard to quantize

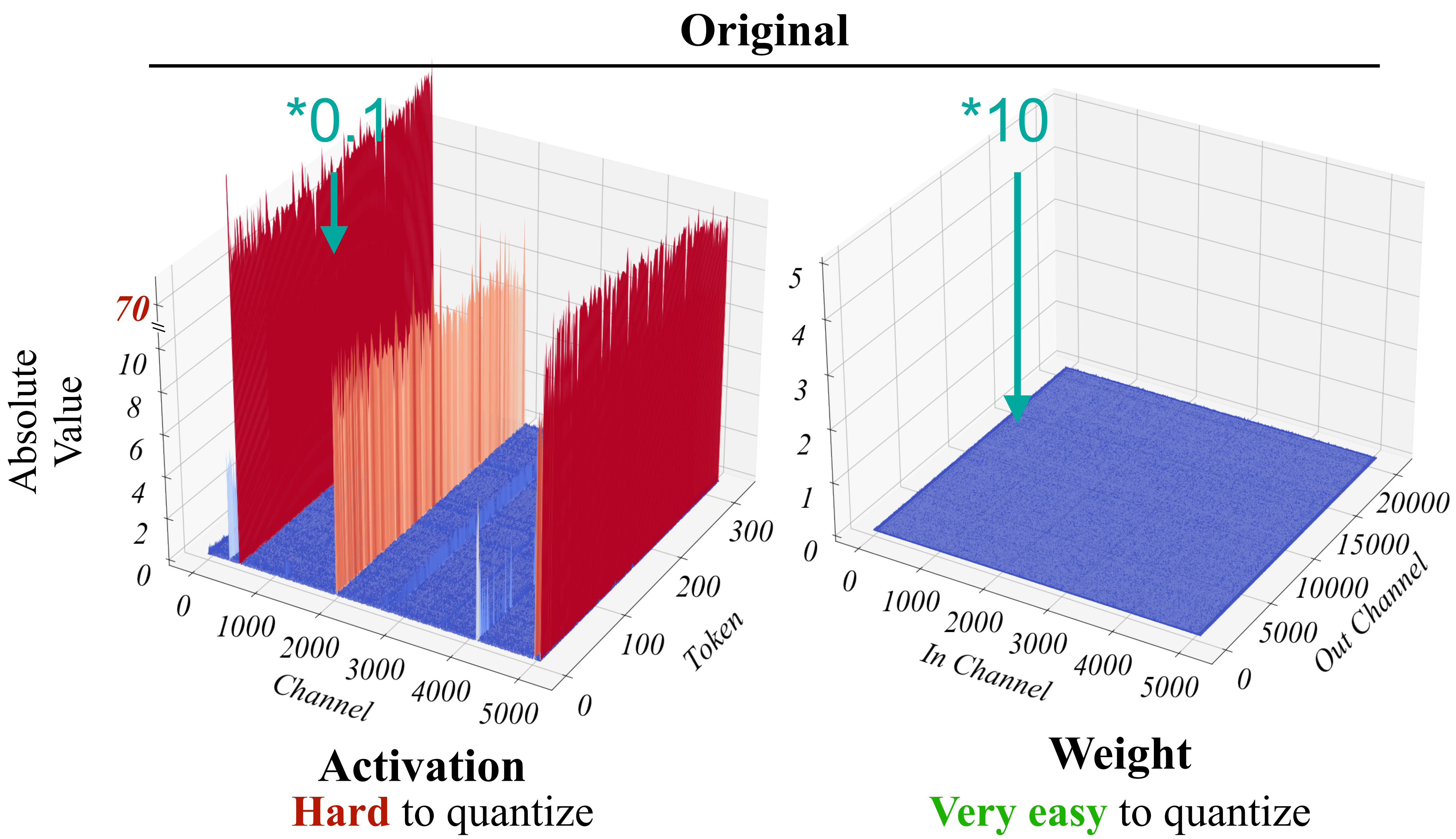
Weight
Very easy to quantize

- Weights are easy to quantize, but activation is hard due to outliers
- Luckily, outliers persist in fixed channels

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., 2022)

SmoothQuant

Smoothing activation to reduce quantization error



*0.1

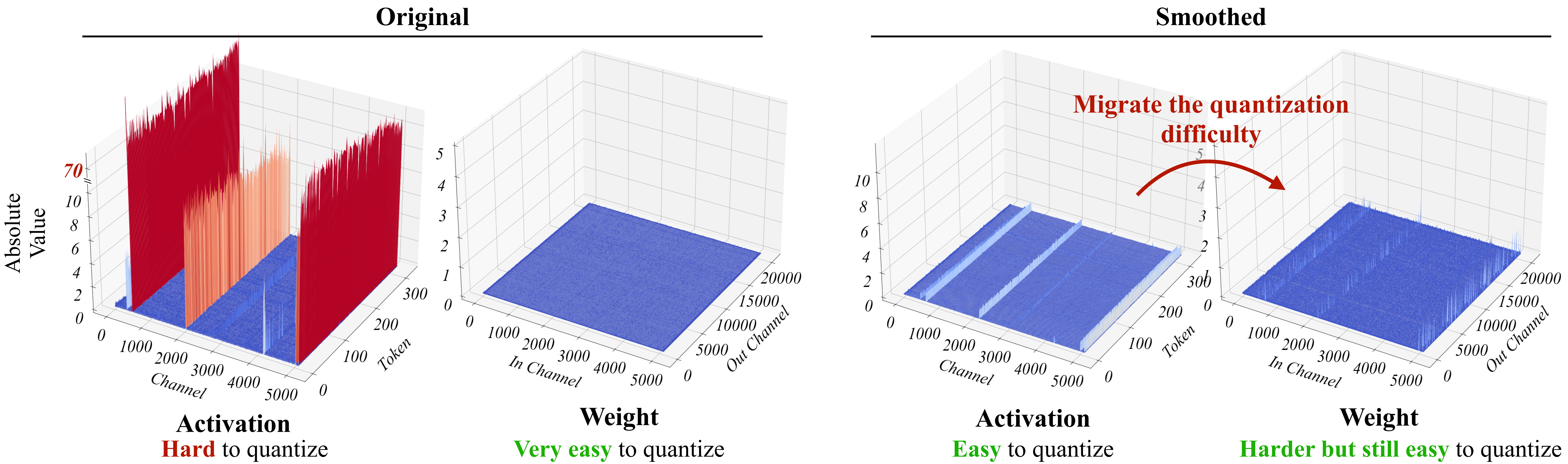
$$\mathbf{Y} = \mathbf{X}\mathbf{W}$$

- Weights are easy to quantize, but activation is hard due to outliers
- Luckily, outliers persist in fixed channels

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., 2022)

SmoothQuant

Smoothing activation to reduce quantization error

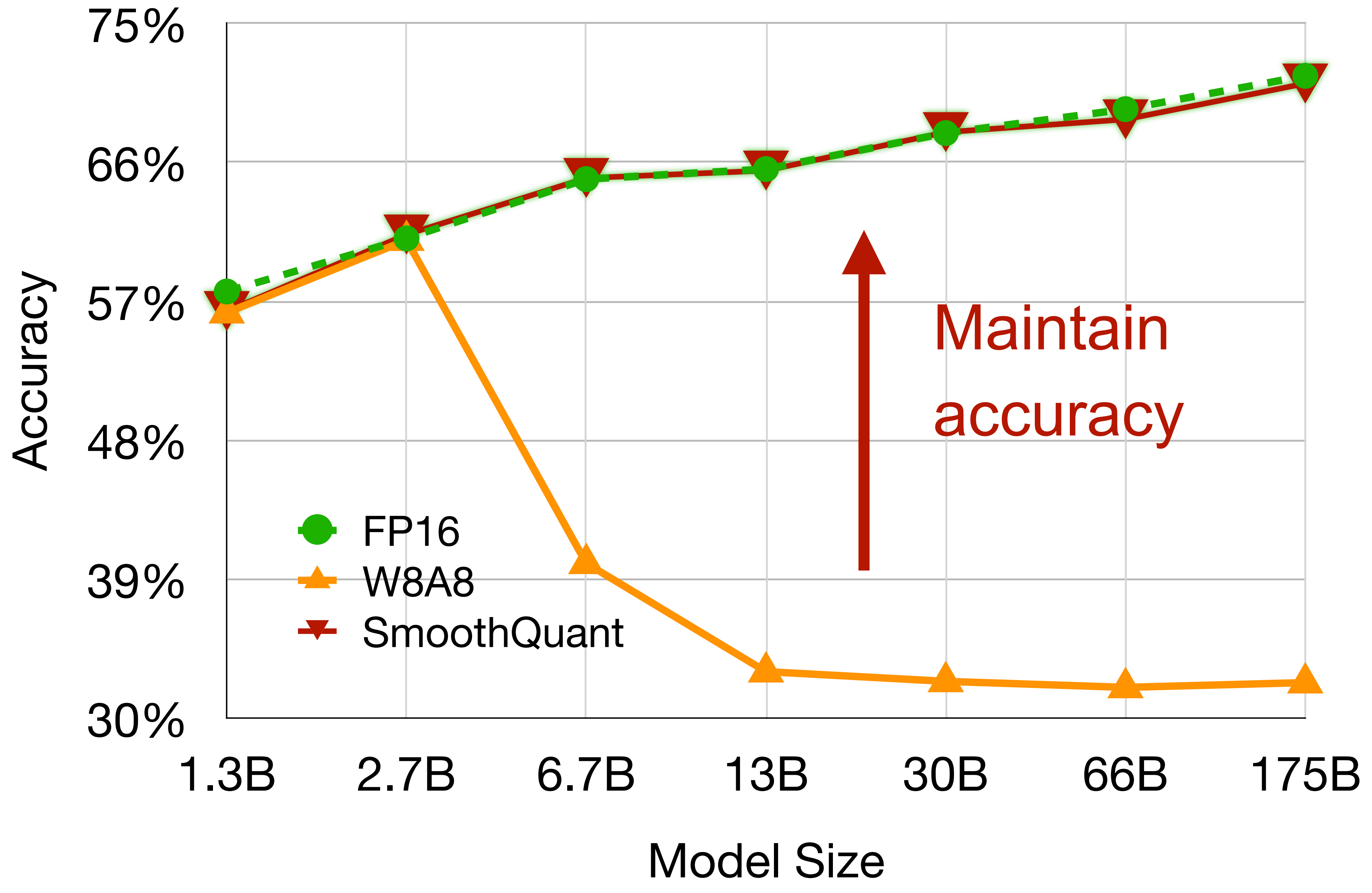


- Weights are easy to quantize, but activation is hard due to outliers
- Luckily, outliers persist in fixed channels
- Migrate the quantization difficulty from activation to weights, so both are easy to quantize

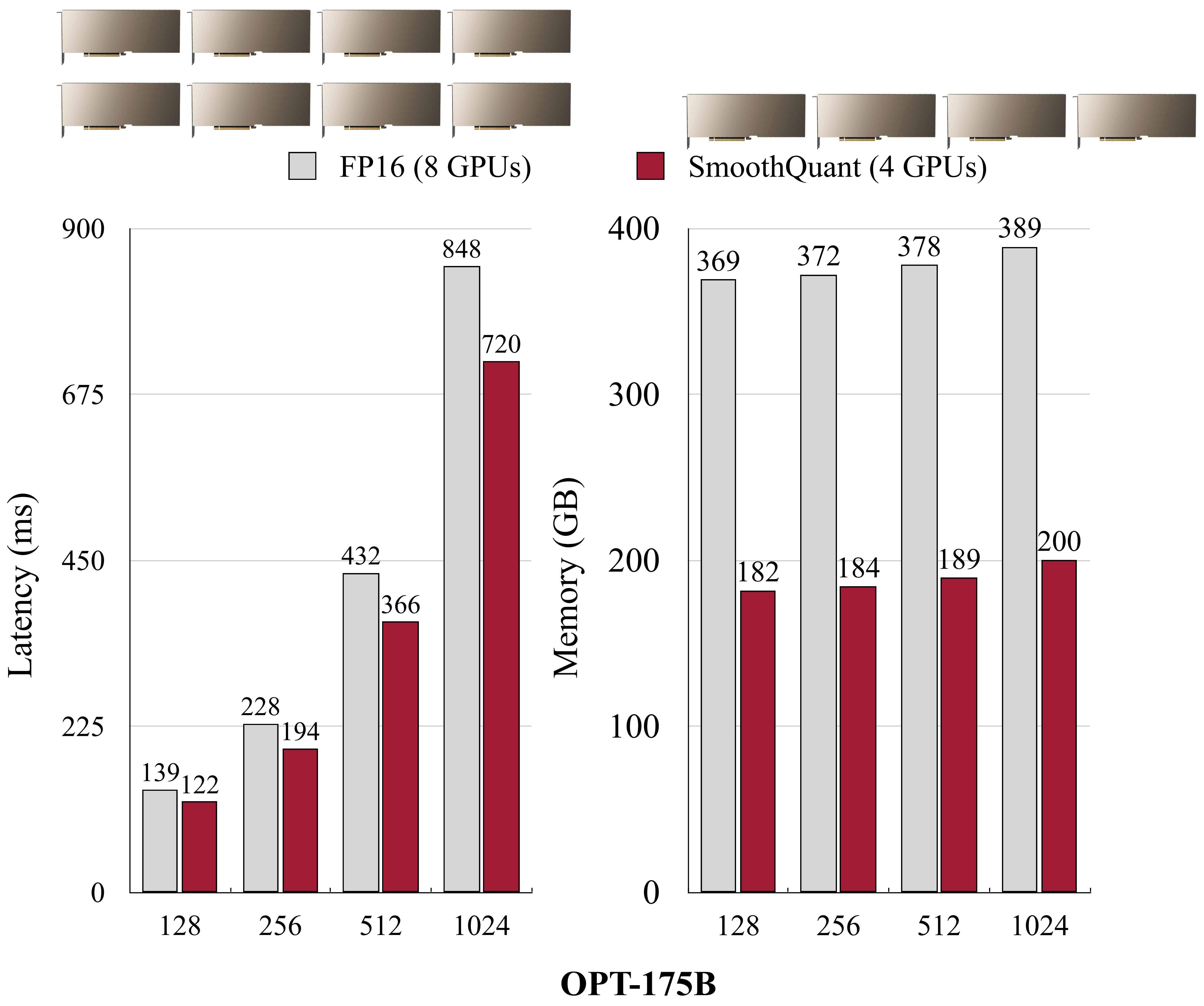
SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., 2022)

SmoothQuant

SmoothQuant is Accurate and Efficient



Maintain accuracy



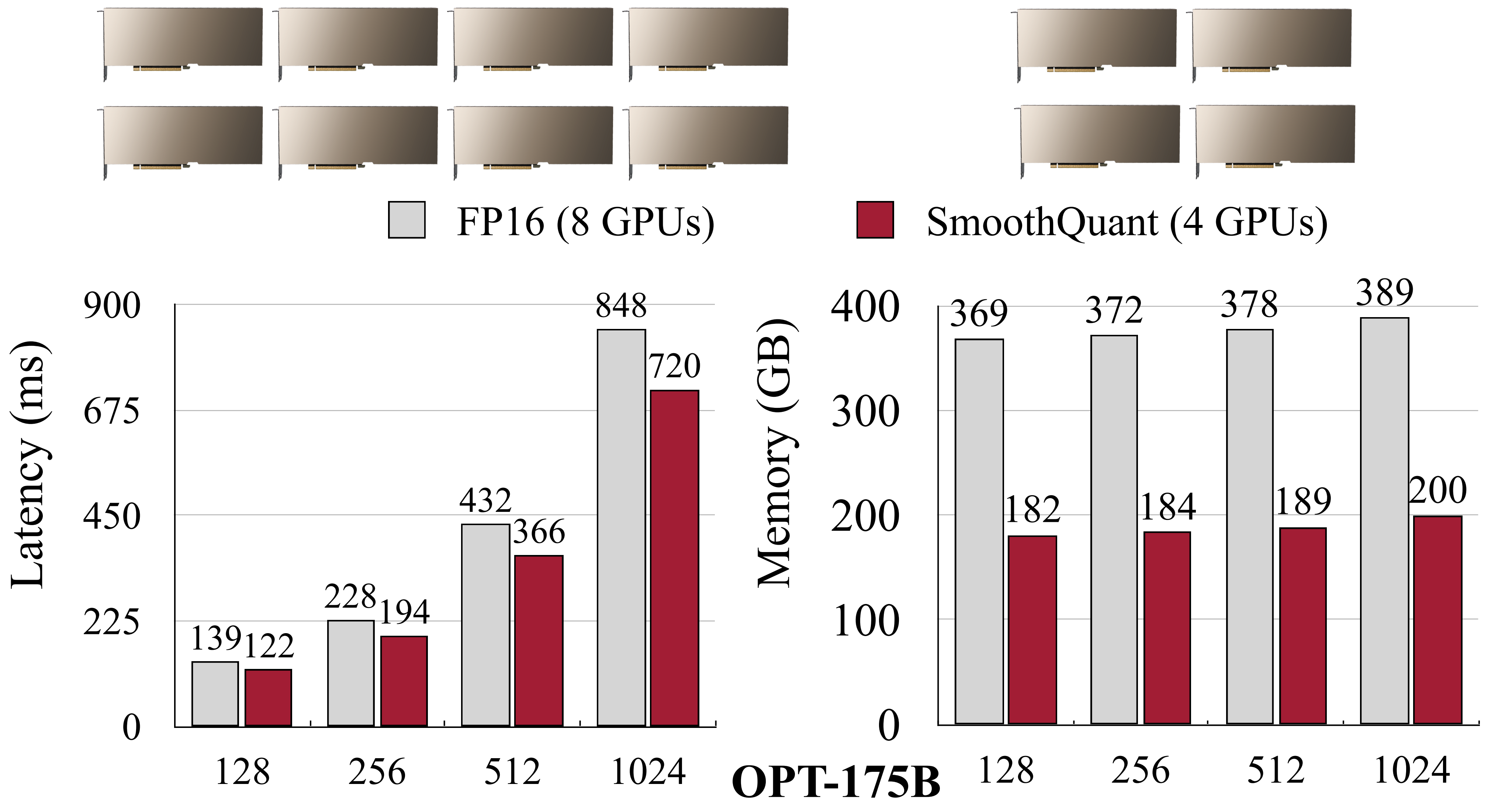
- SmoothQuant well maintains the accuracy without finetuning.
- SmoothQuant can both accelerate inference and halve the memory footprint.

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., 2022)

SmoothQuant

SmoothQuant is Accurate and Efficient

Method	OPT-175B	BLOOM-176B	GLM-130B
FP16	71.6%	68.2%	73.8%
W8A8	32.3%	64.2%	26.9%
ZeroQuant	31.7%	67.4%	26.7%
`LLM.int8()`	71.4%	68.0%	73.8%
Outlier Suppression	31.7%	54.1%	63.5%
SmoothQuant	**71.2%**	68.3%	**73.7%**

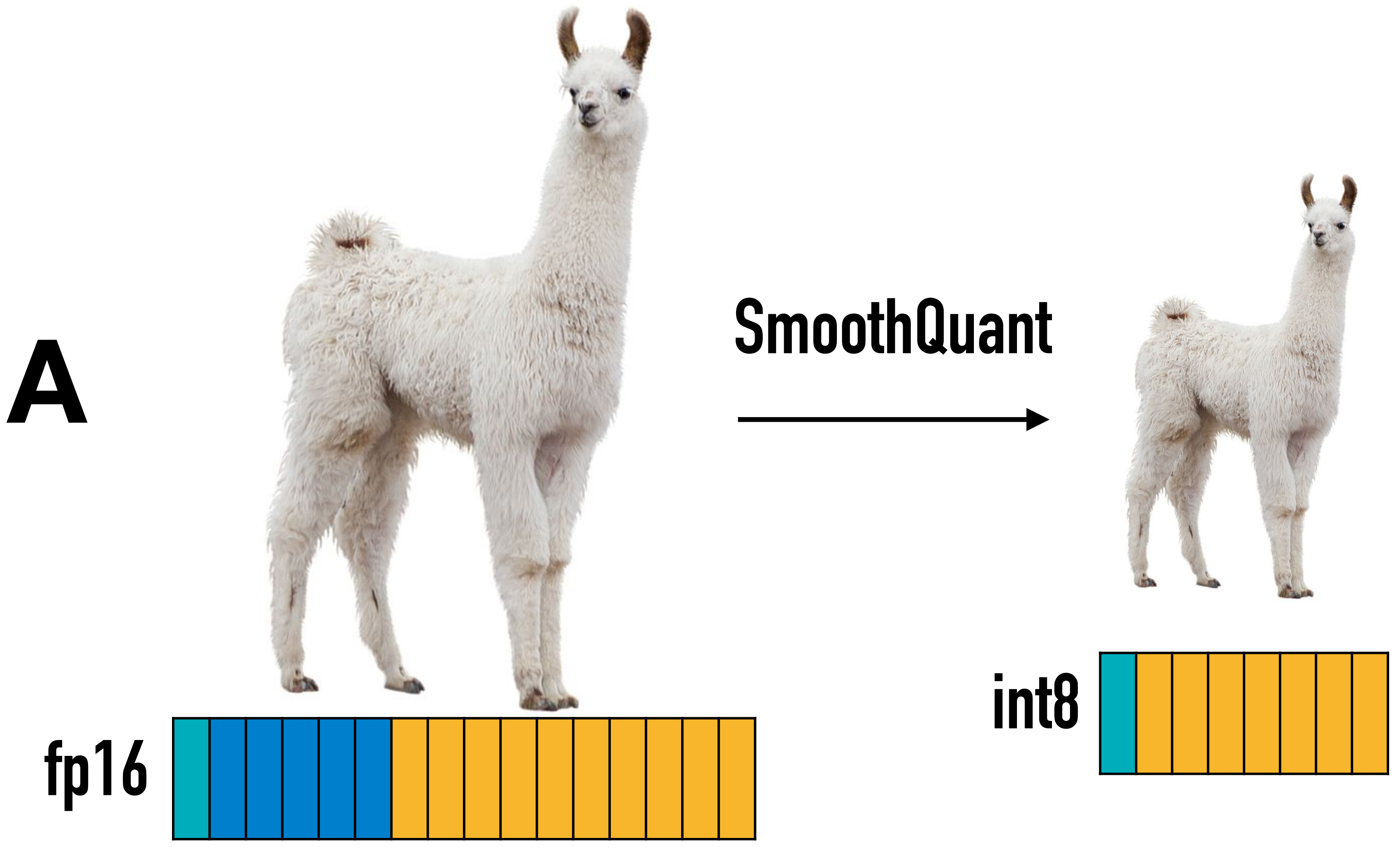


- SmoothQuant well maintains the accuracy without finetuning.
- SmoothQuant can both accelerate inference and halve the memory footprint.

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., 2022)

SmoothQuant

Advancing new efficient open model LLaMA



- **LLaMA** (and its successors like Alpaca) are popular open-source LLMs, which introduced SwishGLU, making activation quantization even harder
- SmoothQuant can losslessly quantize LLaMA families, further lowering the hardware barrier

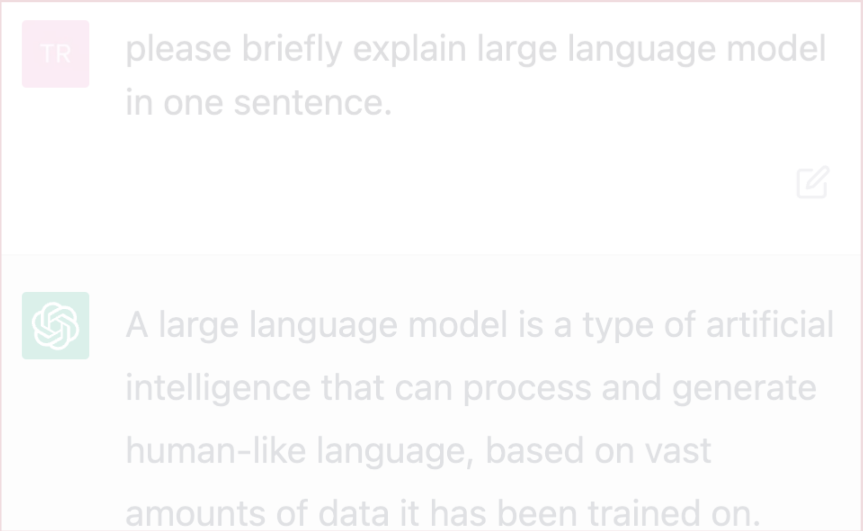
PIQA↑	LLaMA 7B	LLaMA 13B	LLaMA 30B	LLaMA 65B
FP16	78.24%	79.05%	80.96%	81.72%
SmoothQuant	78.24%	78.84%	80.74%	81.50%

Wikitext↓	LLaMA 7B	LLaMA 13B	LLaMA 30B	LLaMA 65B
FP16	11.51	10.05	7.53	6.17
SmoothQuant	11.69	10.31	7.71	6.68

W8A8 per token

Same Principle, Diverse Applications

Applications



TR please briefly explain large language model in one sentence.

A large language model is a type of artificial intelligence that can process and generate human-like language, based on vast amounts of data it has been trained on.

Large Language Model

Generative AI

Advanced Driver Assistance System

TinyML

Techniques

Hardware-aware NAS

Pruning & Sparsity

Quantization

Distillation

New Primitive

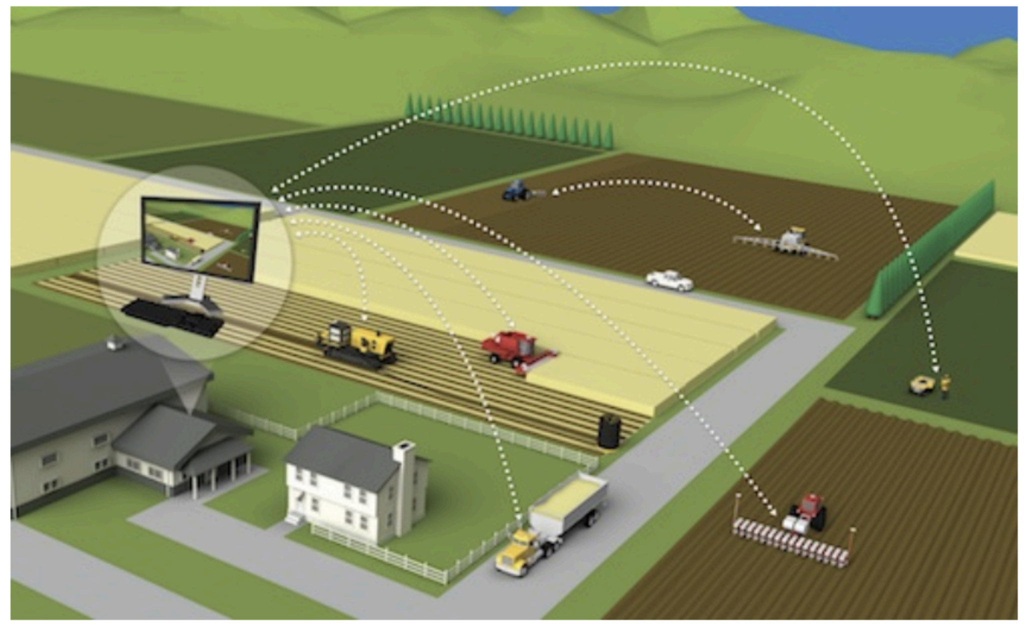
Background: The Era of AIoT on Microcontrollers

Smart Retail

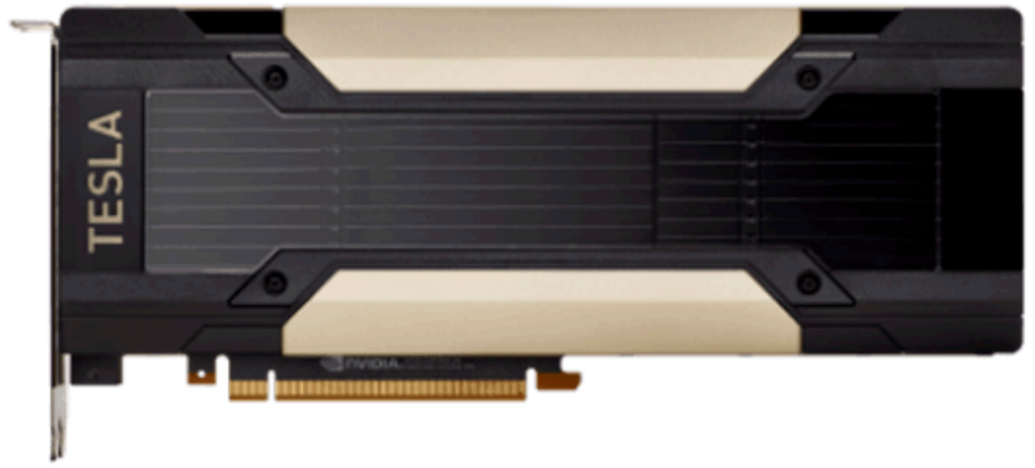
Personalized Healthcare

Smart Home

Precision Agriculture



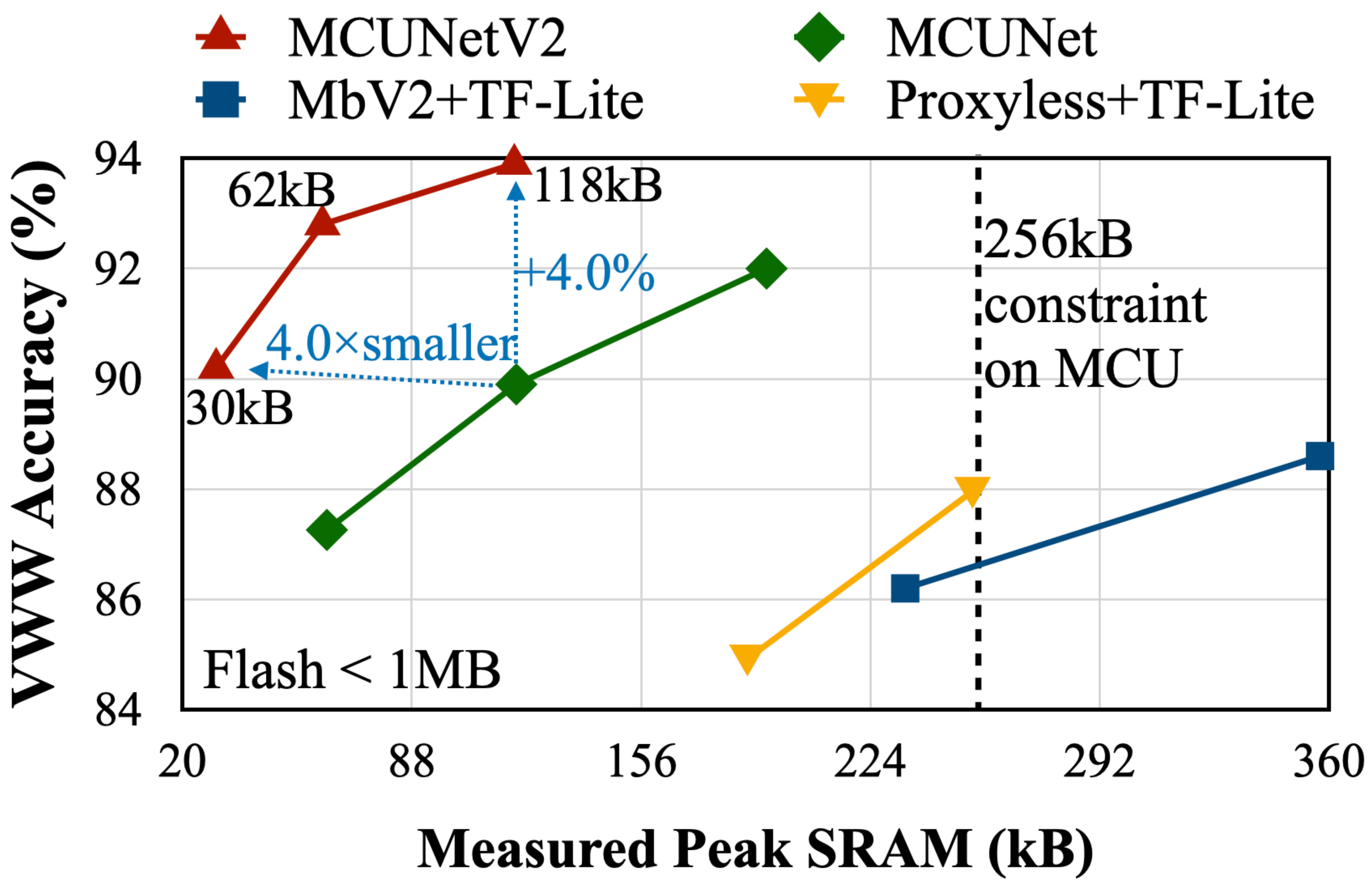
- **Problem**: restricted memory size



	Cloud AI	**Mobile AI**	**Tiny AI**
Memory (Activation)	32GB	4GB	320kB
Storage (Weights)	~TB/PB	256GB	1MB

MCUNet

Deploy AI on MCUs that has only 256KB SRAM



Face/mask detection

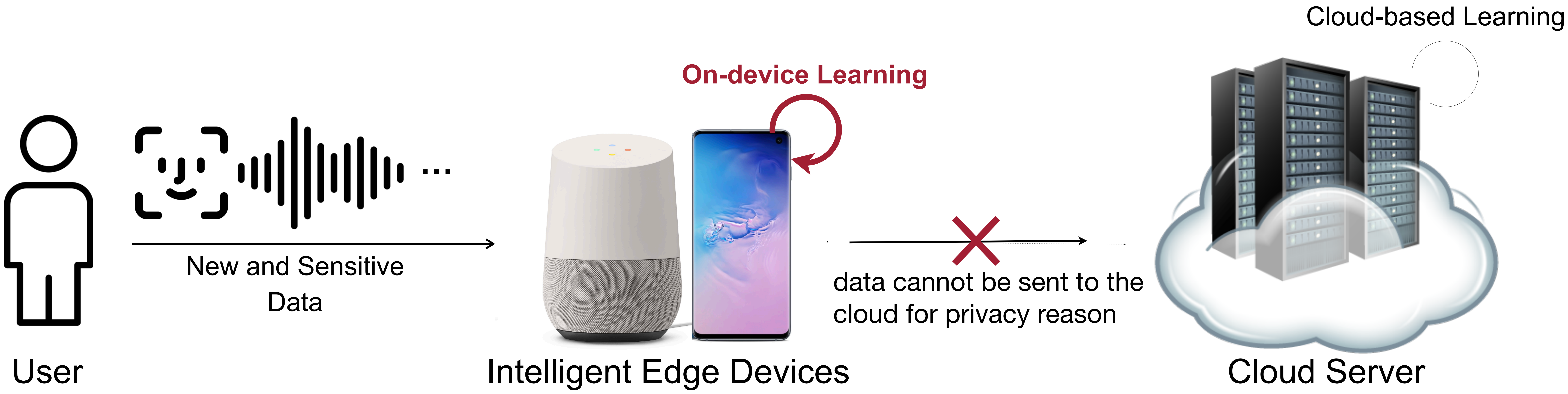
Person detection

The camera is OpenMV Cam.

Inference Is Good. Can We Learn on Edge?

AI systems need to continually adapt to new data collected from the sensors

Not only inference, but also training



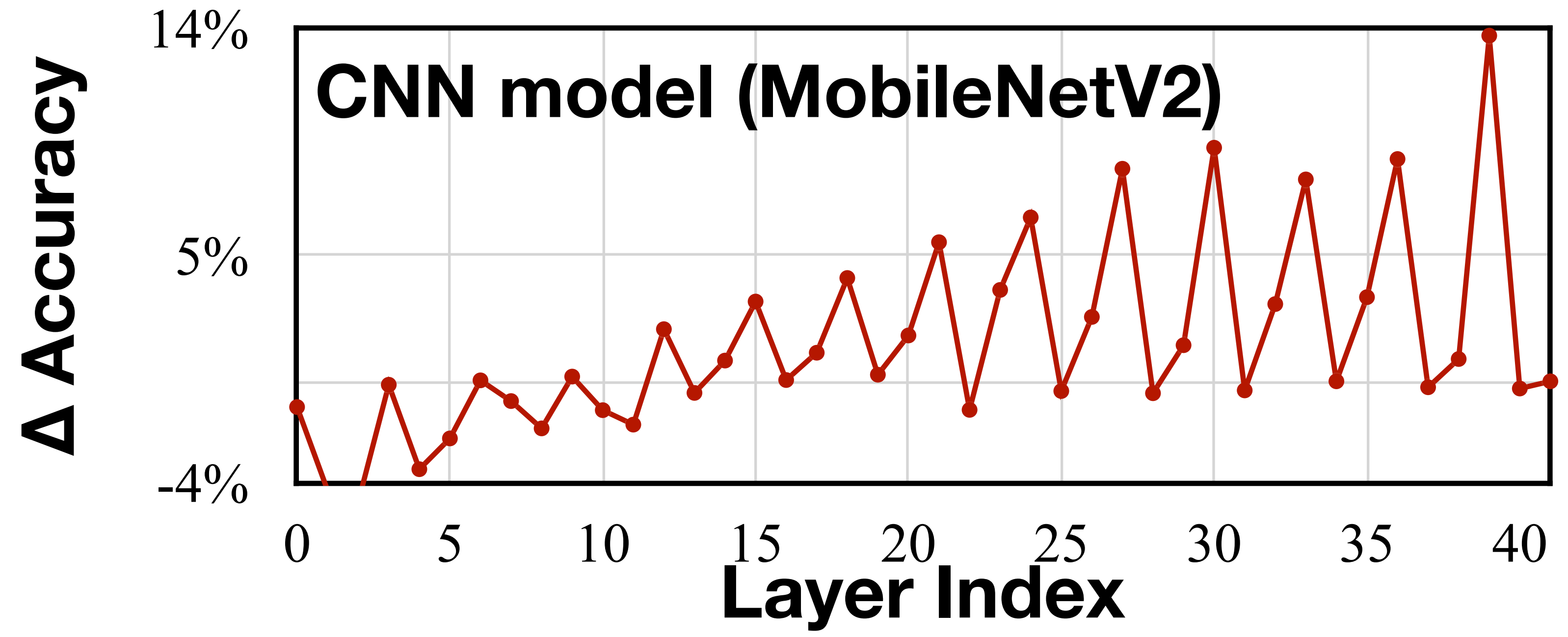
- On-device learning: **better privacy, lower cost, customization, life-long learning**
- Training is more **expensive** than inference, hard to fit edge hardware (limited memory)

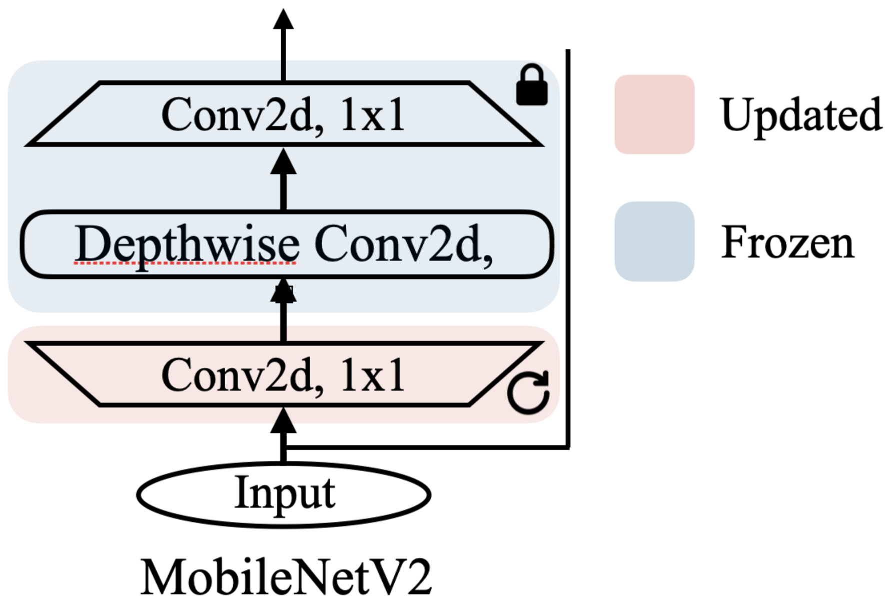
The camera is OpenMV Cam.

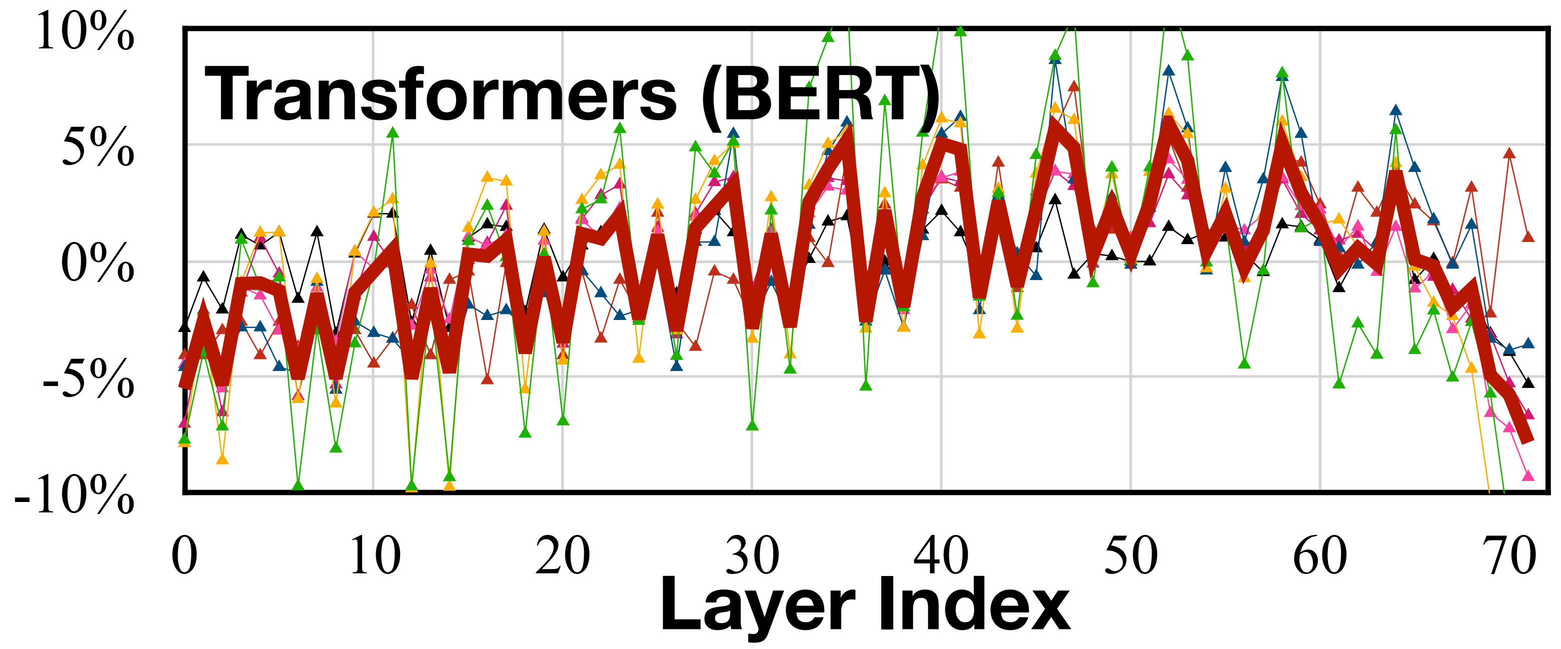
Sparse Training

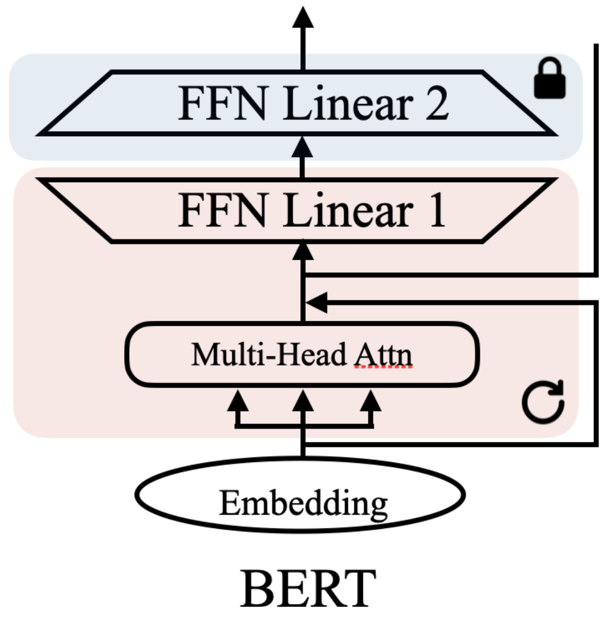
Only update important layers and sub-tensors to save memory

- **Sensitivity analysis**





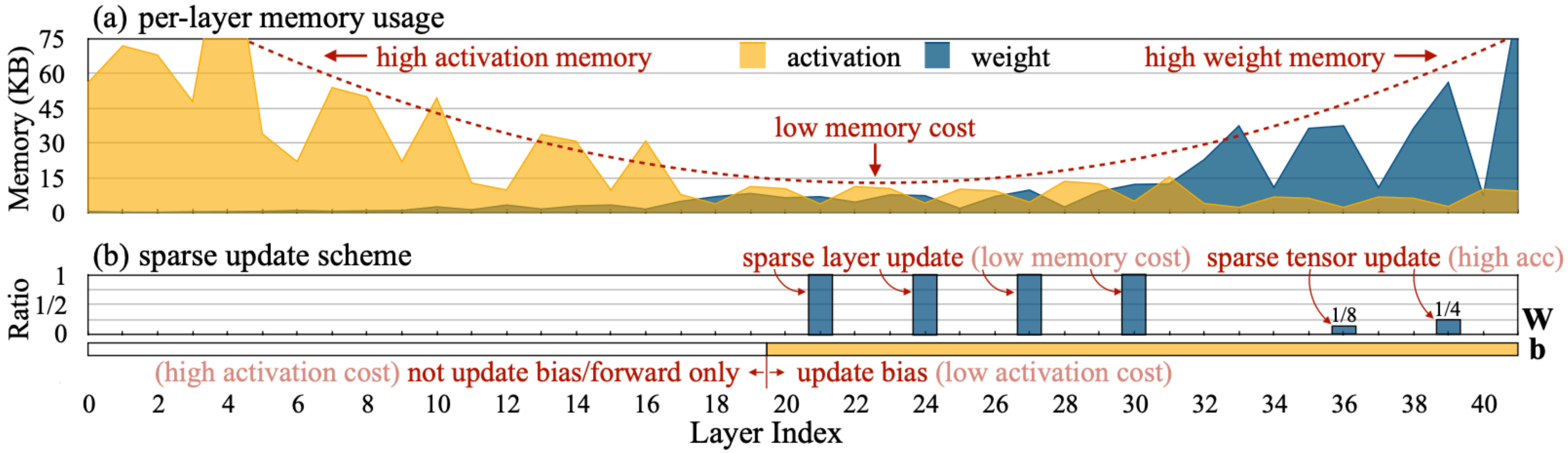




- **Later layers** are more important
- The **first point-wise conv** in each block contributes more

- **Middle layers** are more important
- **Attention and first FFN layers** contribute more.

- **Detailed Update Scheme for MobileNetV2**
  - The **activation cost** is high for the early layers;
  - The **weight cost** is high for the later layers;
  - The **overall memory** cost is low for the middle layers.
    - Bias-only update
    - Update weights for the middle layers



On-Device Training Under 256KB Memory [Lin *et al.*, NeurIPS 2022]

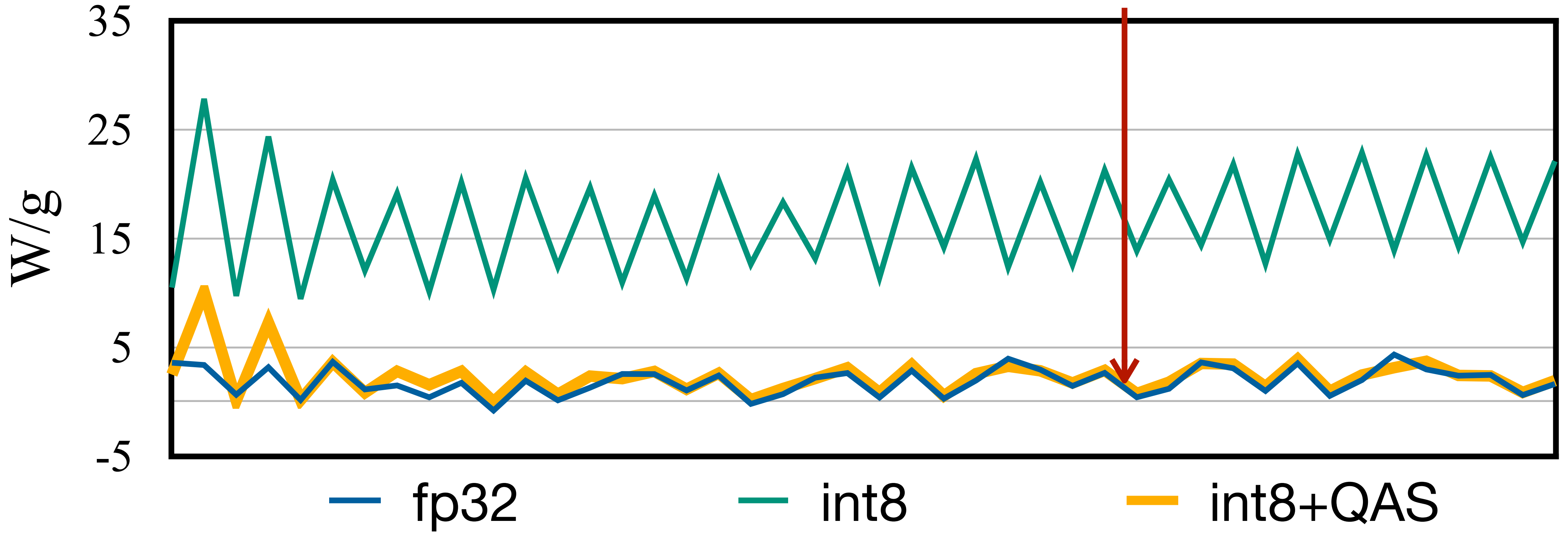
Low-Precision Training

with Quantization Aware Scaling (QAS)

- Optimizing an INT8 quantized graph leads to **memory and computing savings**
    - All weights and activations are in **INT8**
    - Different from quantization-aware training (QAT), where operations are performed in **FP16**

- … But at the cost of **worse convergence**
- We found the issue lie lies in gradient scale mismatch

QAS aligns the W/G ratio



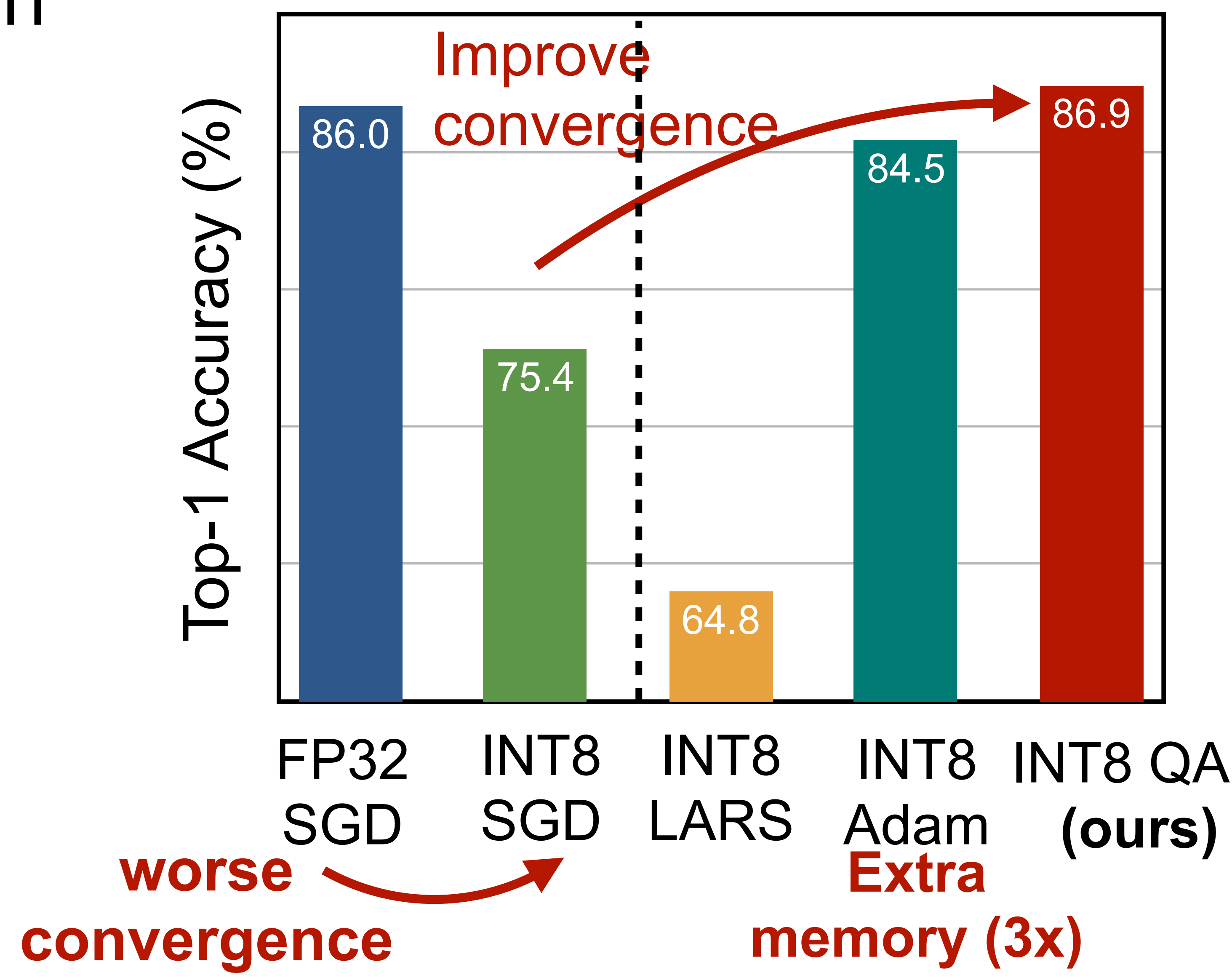
- Solution: **quantization-aware scaling (QAS)**

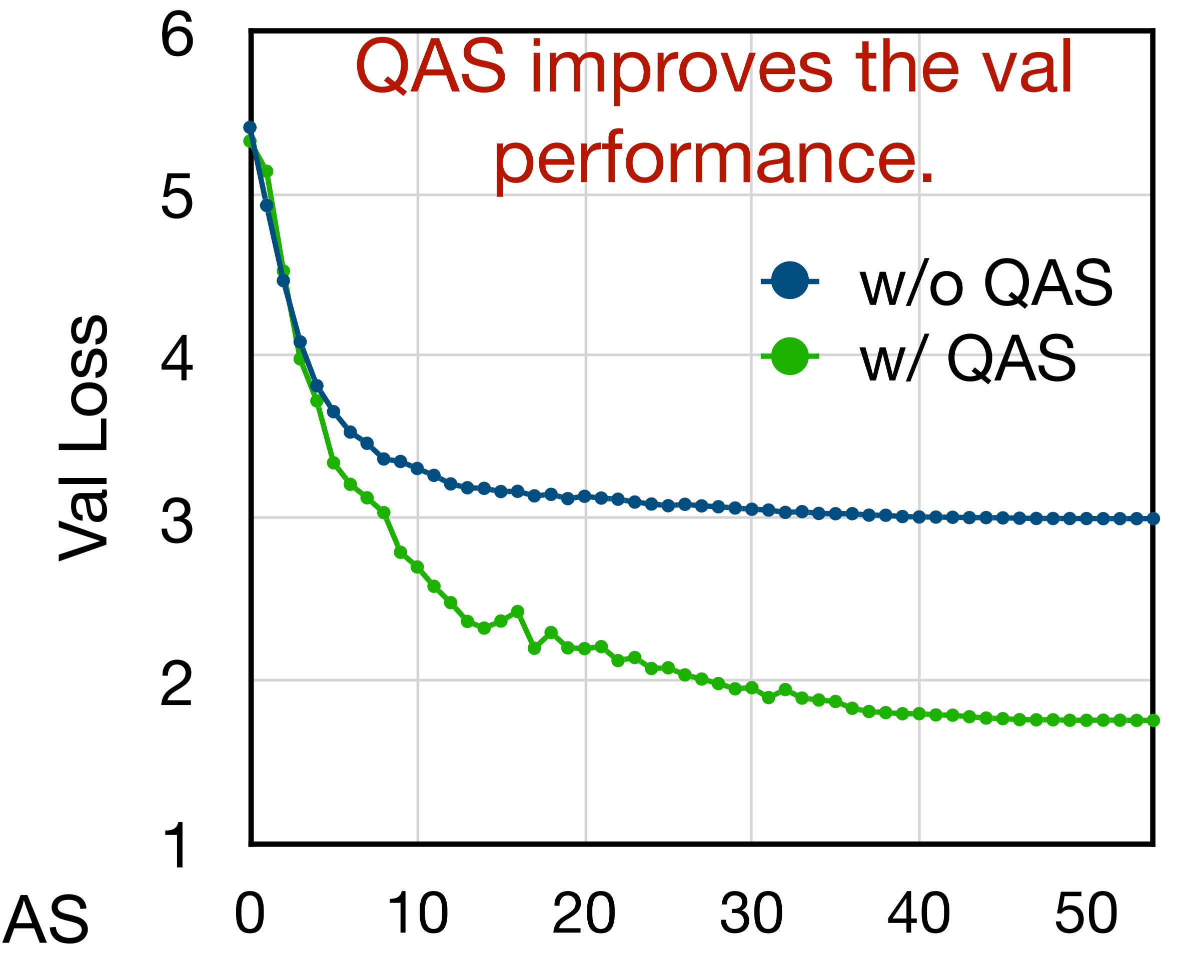
$$\overset{\text{[-2, 3]}}{W} = s_W \cdot (W/s_W) \overset{quantize}{\approx} s_W \cdot \overset{\text{[-128, 127]}}{W_Q}, \quad G_{W_Q} \approx s_W \cdot G_W$$

weight and gradient ratios are off by s_W^{-2}

$$\|W_Q\|/\|G_{W_Q}\| \approx \|W/s_W\|/\|s_w \cdot G_W\| = s_W^{-2} \cdot \|W\|/\|G\|$$

Thus, we need to re-scale the gradients $G'_{W_Q} = G_{W_Q} \cdot s_W^{-2}$

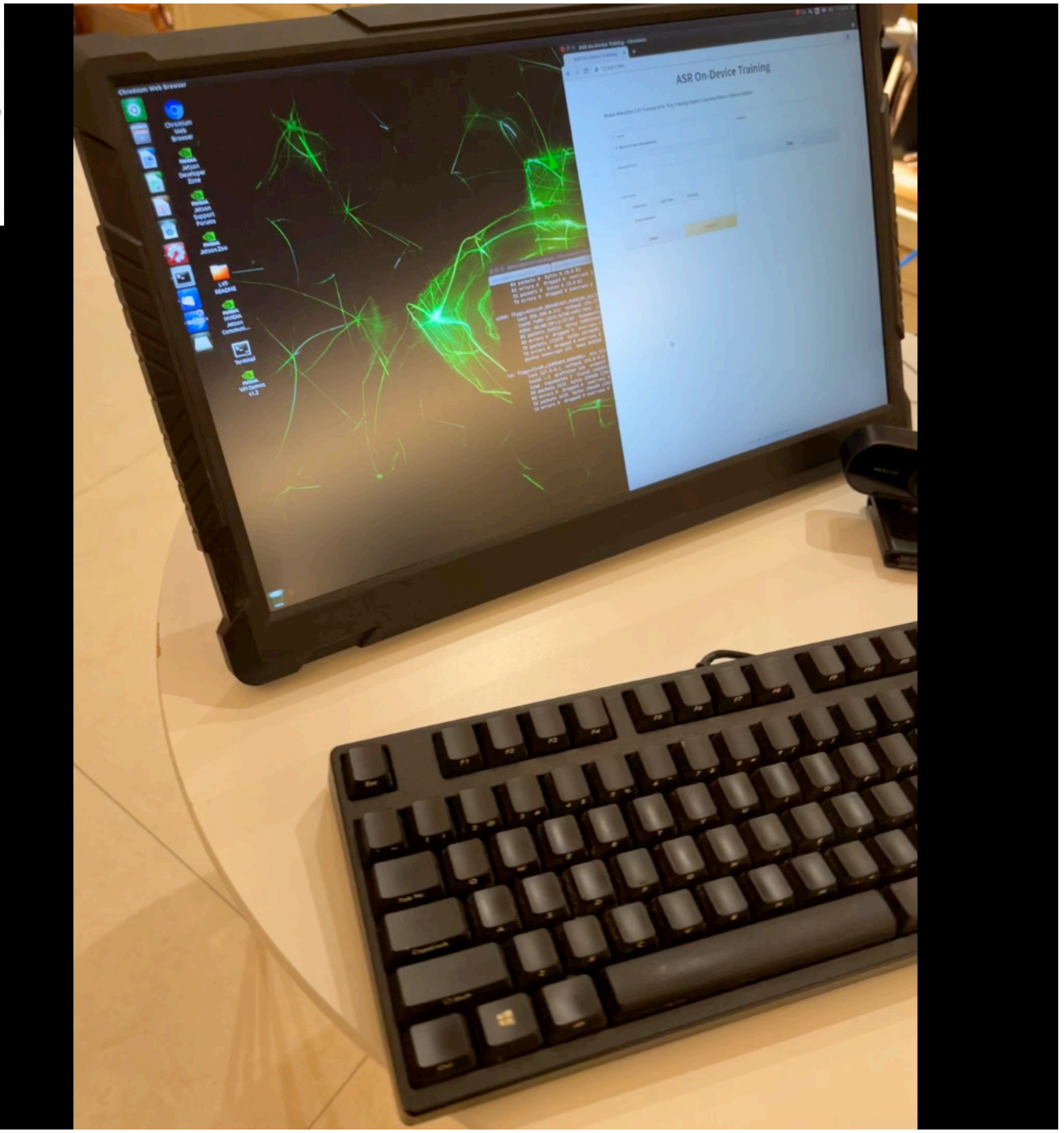


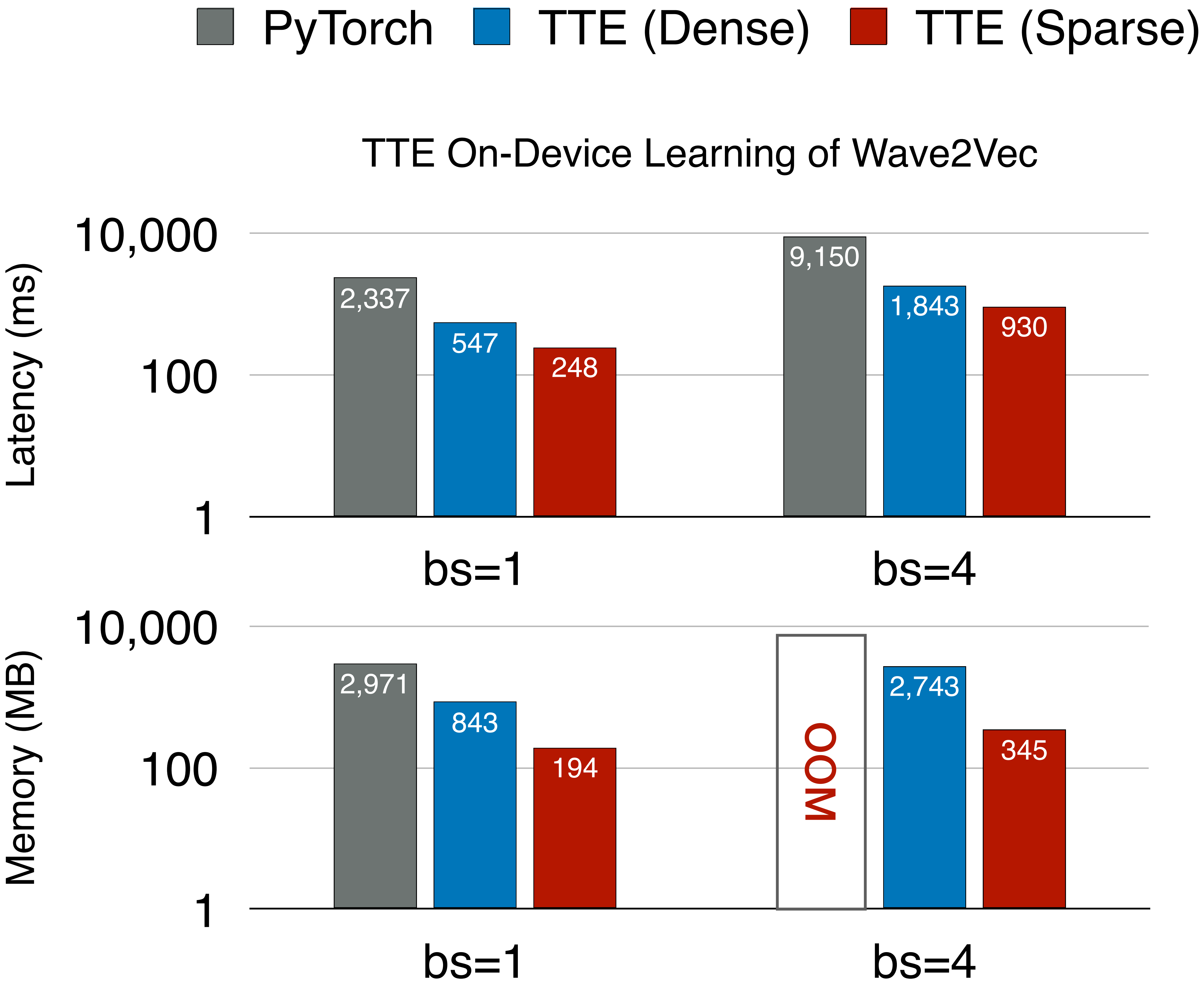


On-Device Training Under 256KB Memory [Lin *et al.*, NeurIPS 2022]

Tiny Training Engine

Translate the theoretical savings into measured savings. 10x faster and smaller!





Device: Jetson Nano; Backend:Tiny Training Engine; Task: Speech Recognition

Model Compression for Diverse Applications

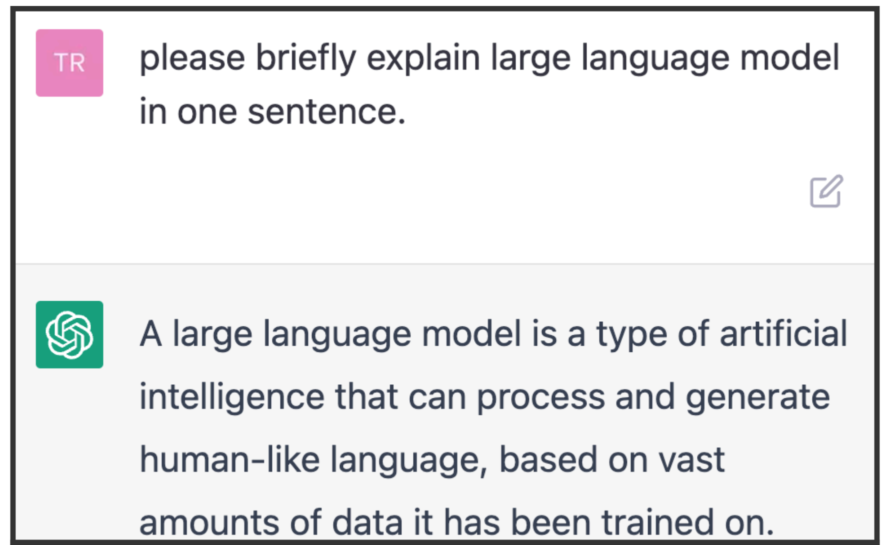
Video Synthesis Search Engine Revolution Chatbots

Predictive Maintenance Art Generation Question Answering Augmented Reality

Gesture Recognition Storytelling Autonomous Driving

Video Recognition Music Composition Sentiment Analysis Blind Spot Detection

Health Monitoring Fashion Design Machine Translation Adaptive Cruise Control



TR please briefly explain large language model in one sentence.

A large language model is a type of artificial intelligence that can process and generate human-like language, based on vast amounts of data it has been trained on.

Large Language Model

Generative AI

Driver Assistance System

TinyML

Application
(demand of computation)

Hardware
(supply of computation)

TinyML and Efficient AI

MIT HAN LAB
Hardware, AI and Neural-nets

github.com/mit-han-lab

youtube.com/c/MITHANLab

songhan.mit.edu
tinyml.mit.edu

Sponsors: IBM QUALCOMM intel NVIDIA XILINX maxim integrated NSF amazon G f SONY SAMSUNG HYUNDAI Ford

Media: MIT Technology Review IEEE SPECTRUM WIRED engadget MIT News ON CAMPUS AND AROUND THE WORLD AI DAILY VentureBeat ScienceDaily Analytics Insight AI Business