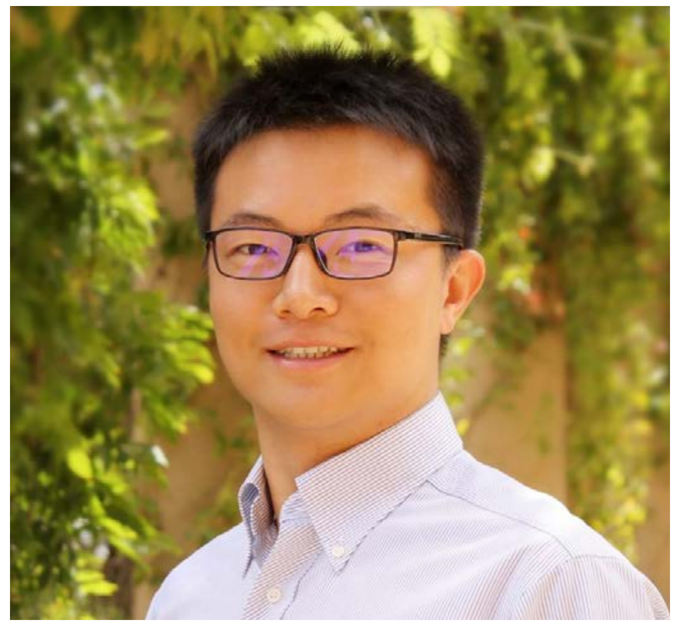
HAN LAB
Hardware Accelerated Neural-nets

Accelerating Large Language Models and Generative AI



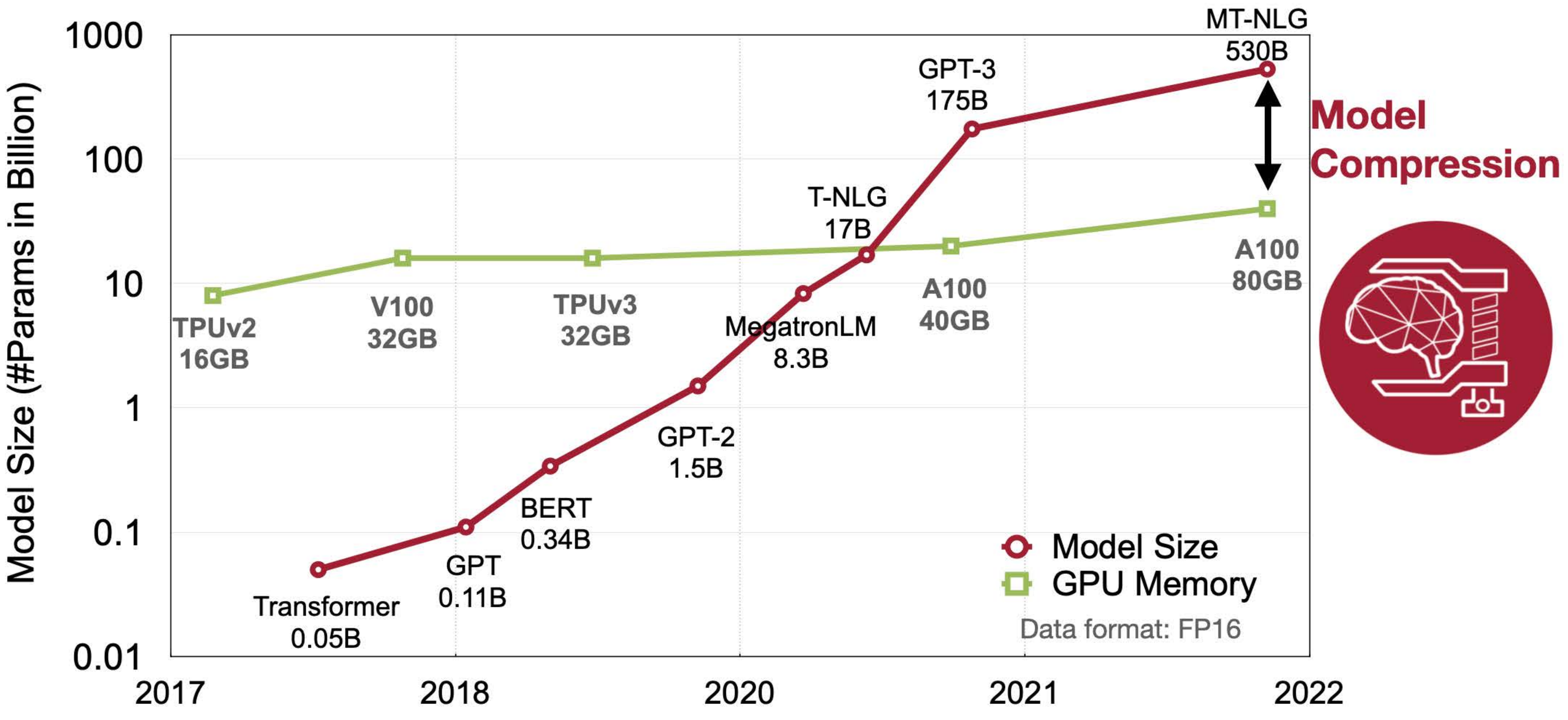
Song Han

Associate Professor, MIT
Distinguished Scientist, NVIDIA

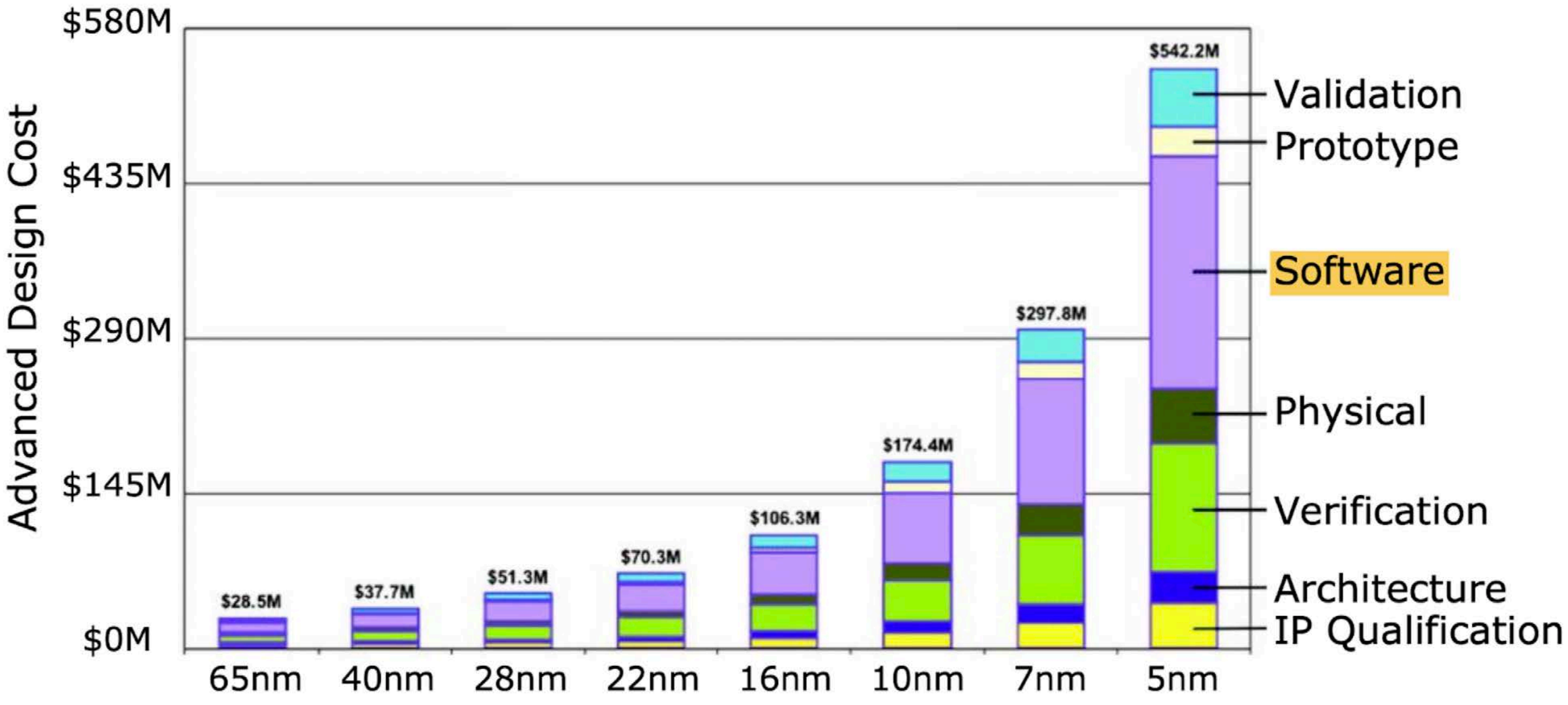
https://songhan.mit.edu
@SongHan_MIT

The Need for Efficient AI Computing

co-design software and hardware



The demand for AI computing is increasing fast

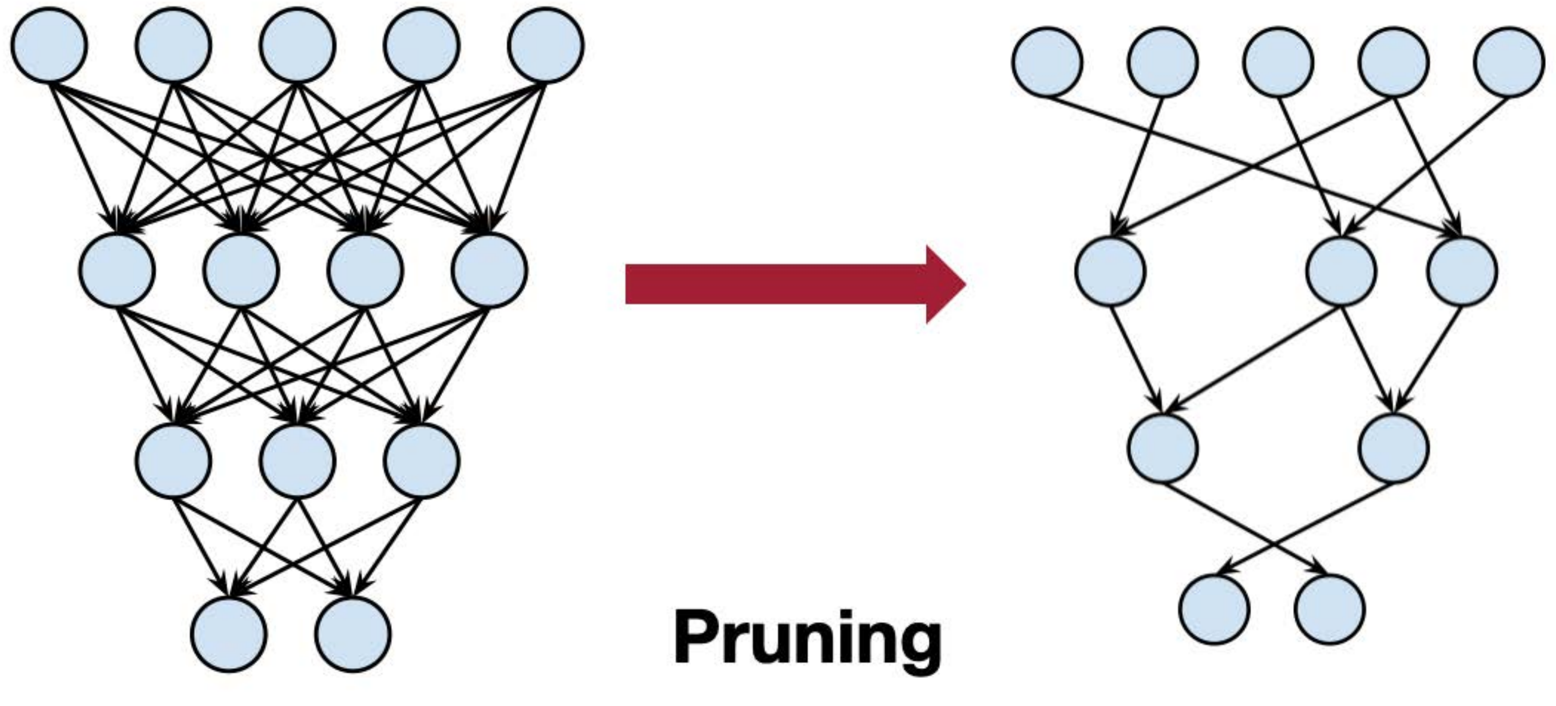


Software is important, the cost is high

[source]

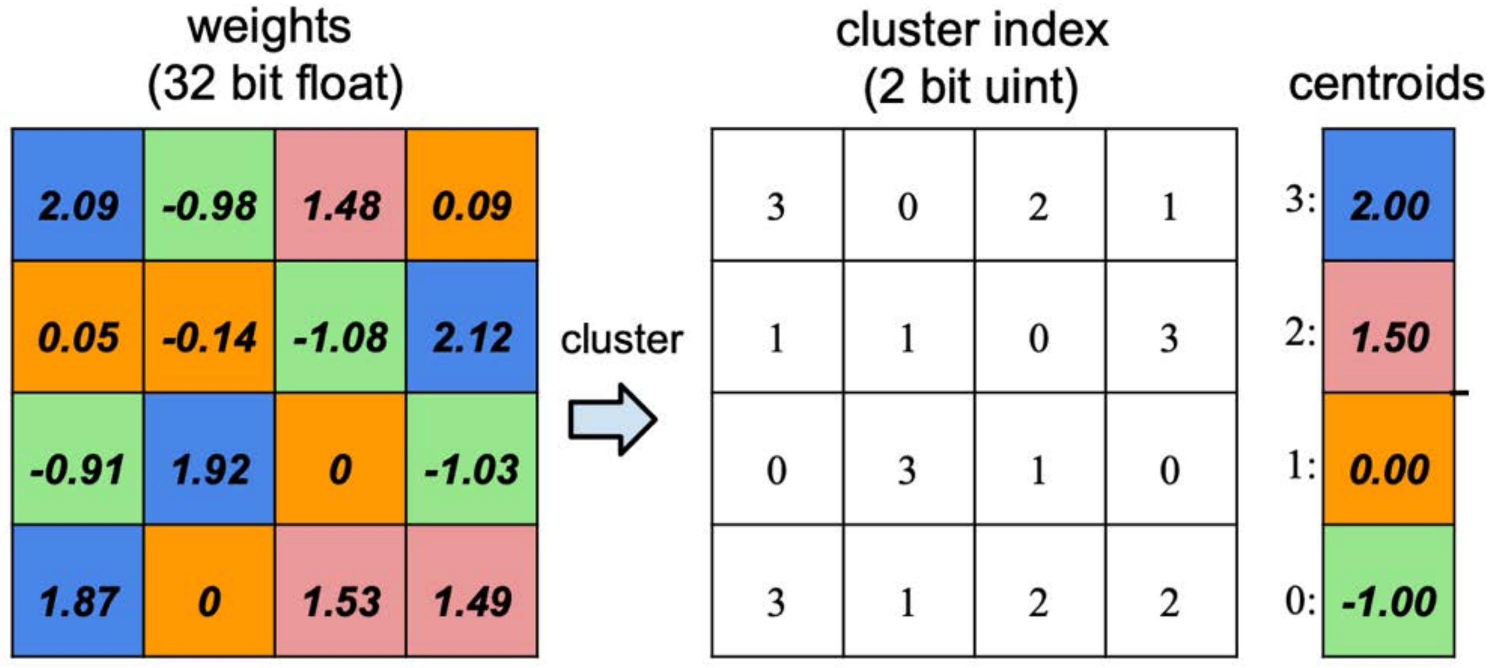
Previous Work

Deep Compression and EIE



Pruning

Deep Compression

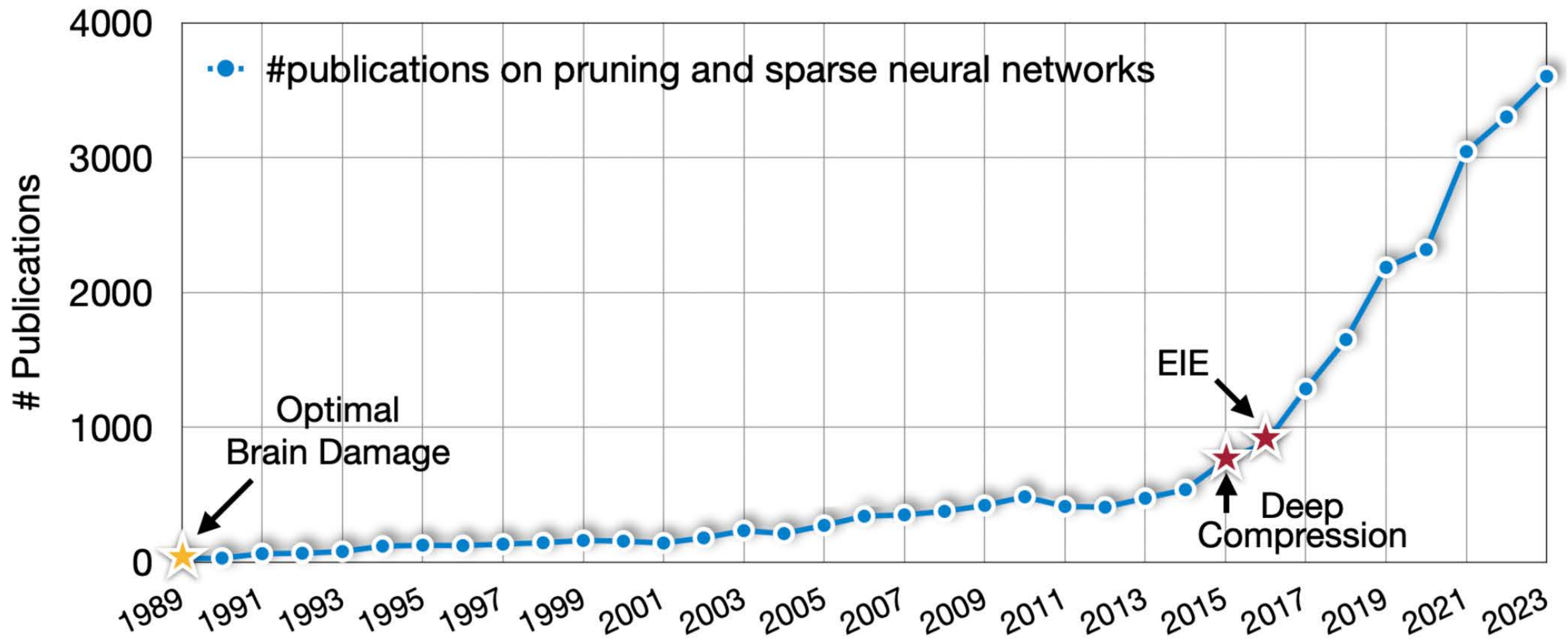


Quantization

Top-5 most cited papers in 50 years of ISCA (1953-2023)

Rank	Citations	Year	Title (★ means it won the ISCA Influential Paper Award)	First Author + HOF Authors	Type	Topic
1	5351	1995	The SPLASH-2 programs: Characterization and methodological considerations	Stephen Woo, *Anoop Gupta*	Tool	Benchmark
2	4214	2017	In-datacenter performance analysis of a Tensor Processing Unit	*Norm Jouppi, David Patterson*	Arch	Machine Learning
3	3834	2000	★ Watch: A framework for architectural-level power analysis and optimizations	*David Brooks, Margaret Martonosi*	Tool	Power
4	3386	1993	★ Transactional memory: Architectural support for lock-free data structures	Maurice Herlihy	Micro	Parallelism
5	2690	2016	EIE: Efficient inference engine on compressed deep neural network	Song Han, *Bill Dally, Mark Horowitz*	Arch	Machine Learning

Efficient Inference Engine



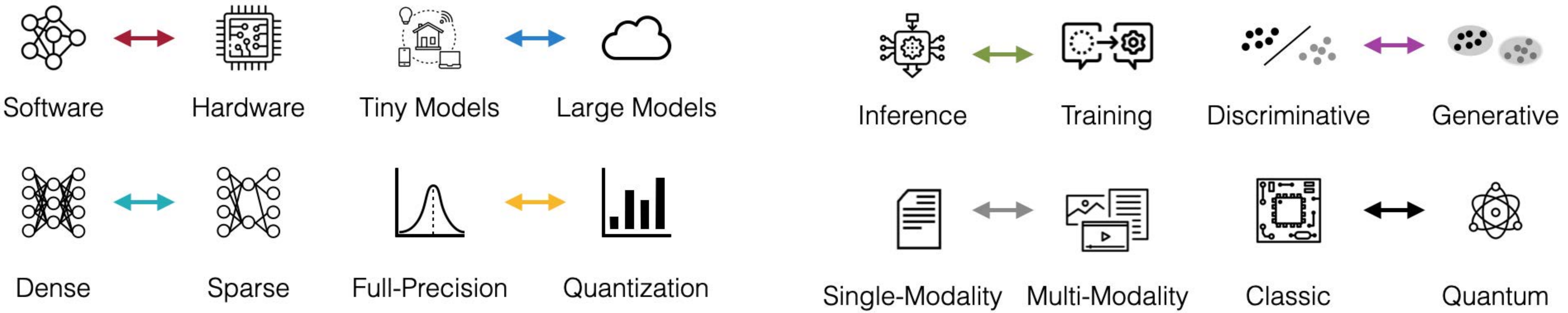
[NIPS'15, ICLR'16, ISCA'16]

EfficientML Project

Bridge the supply and demand of AI computing

Algorithm and system co-design for accelerated AI computing

Goal: reduce latency, memory, low power/energy; increase throughput, accuracy, scalability.



Low Precision

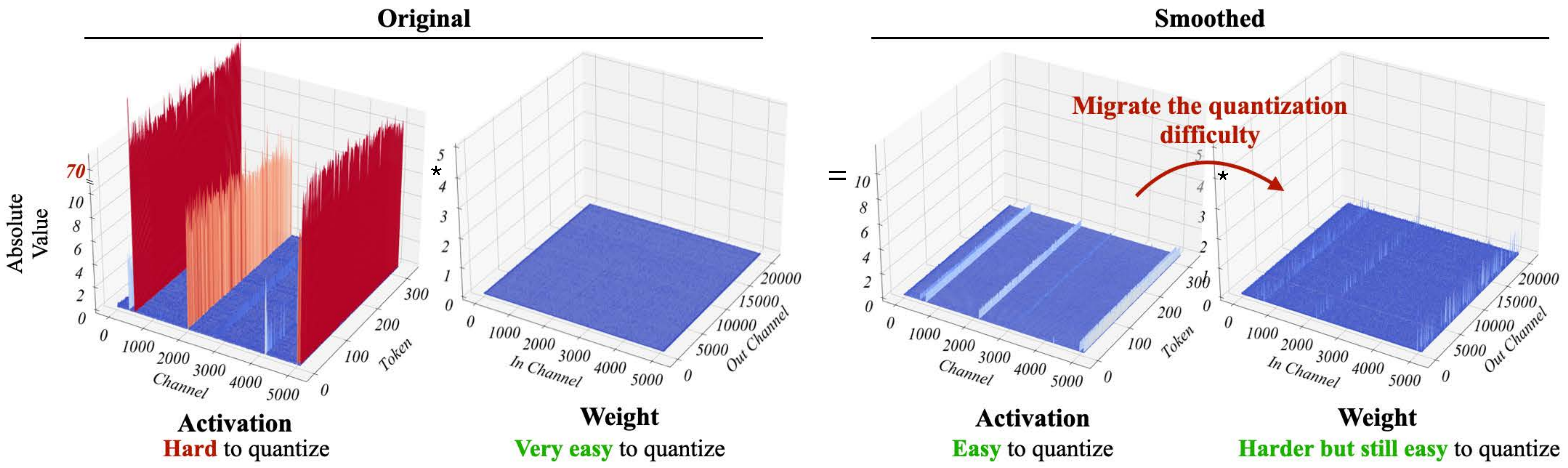
SmoothQuant

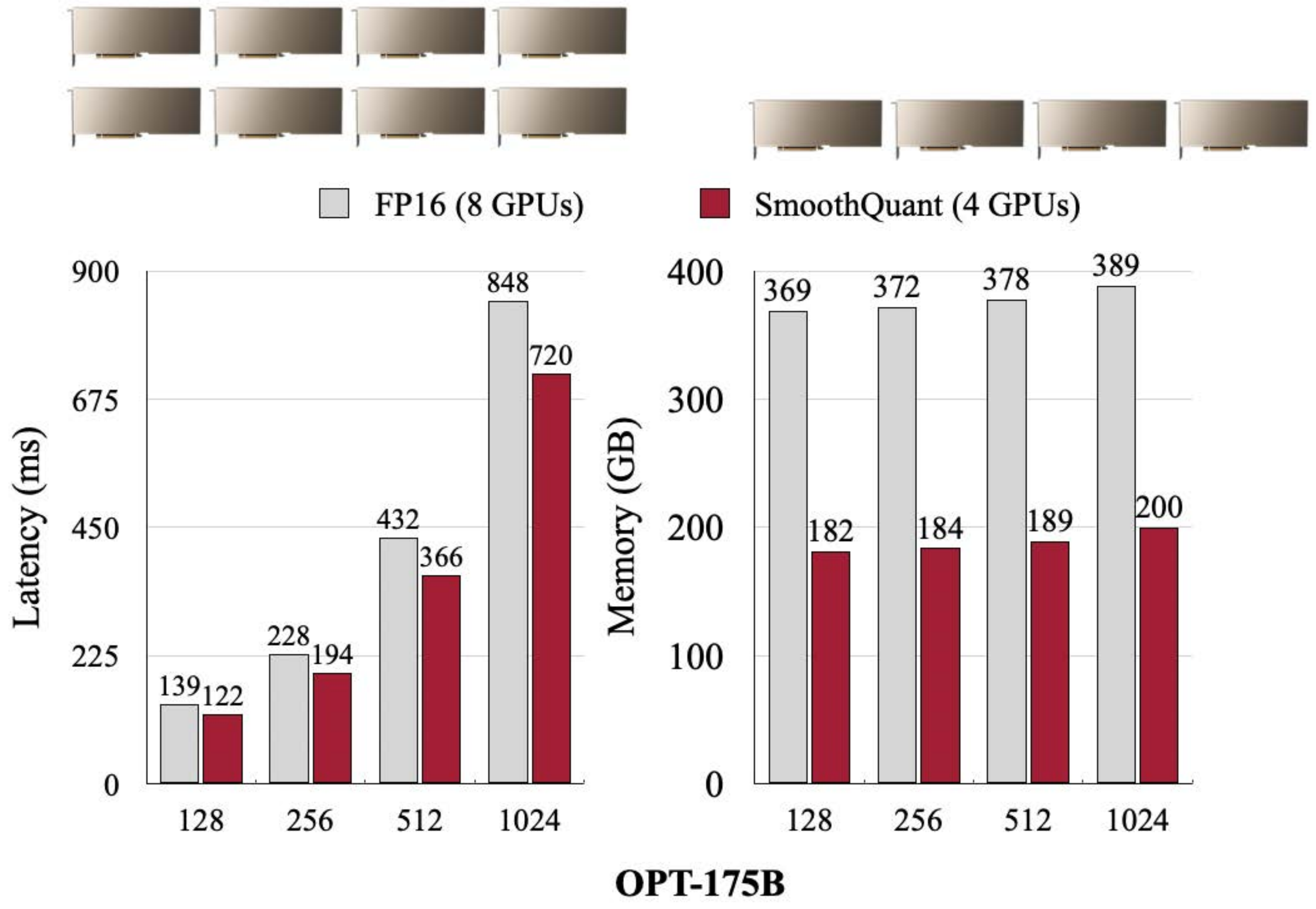
SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models

Goal: Quantize LLM to lower precision, both activation and weight

Challenge: activation channels have many outliers, wasting the dynamic range (many channels became zero)

Our Solution: Smooth the activations: 100*1= 10*10; Equalize the quantization difficult from activation to weights.





AWQ for On-Device LLM

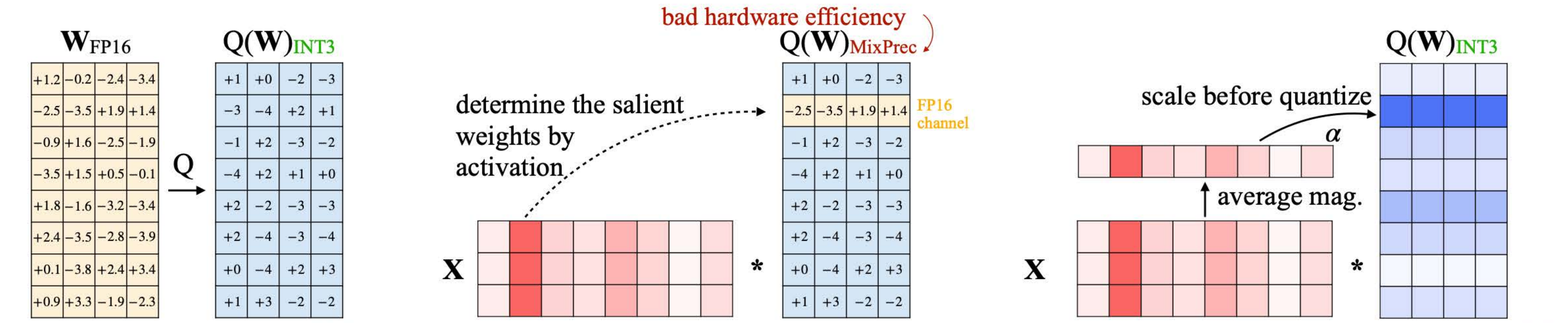
AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration

Goal: deploy LLM on the edge: Jetson Orin, AI PC

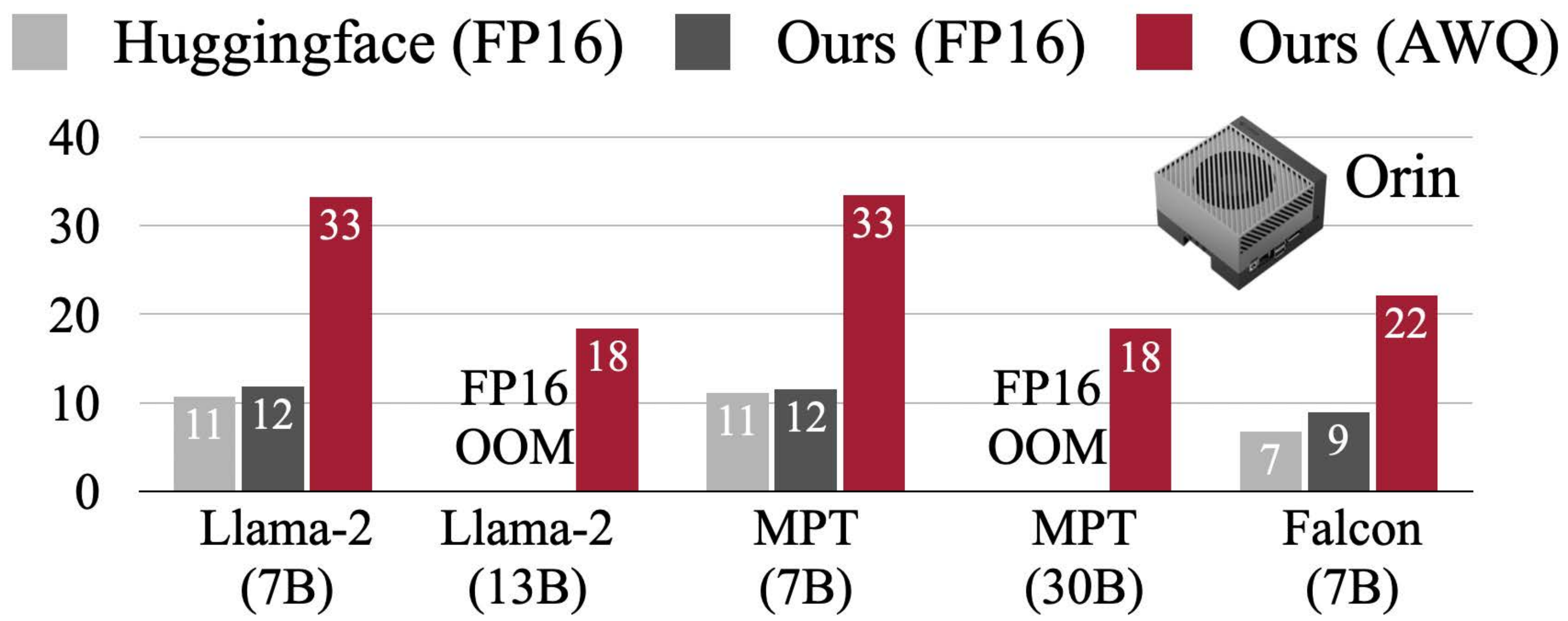
Challenge: weight memory bounded @low batch size; can't fit; idle ALU.

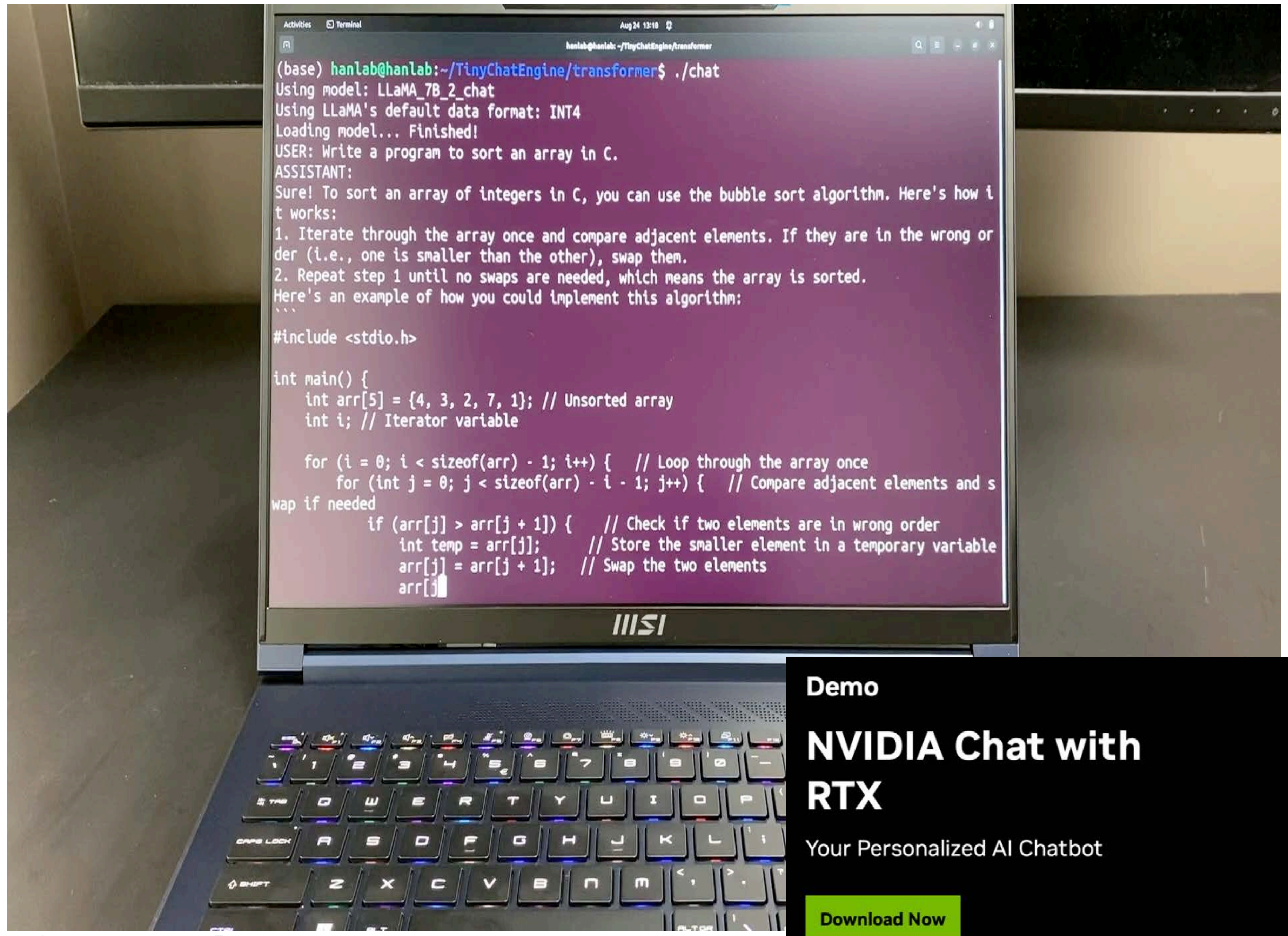
Our Solution: 4bit weights, fp16 activation, fp16 arithmetic.

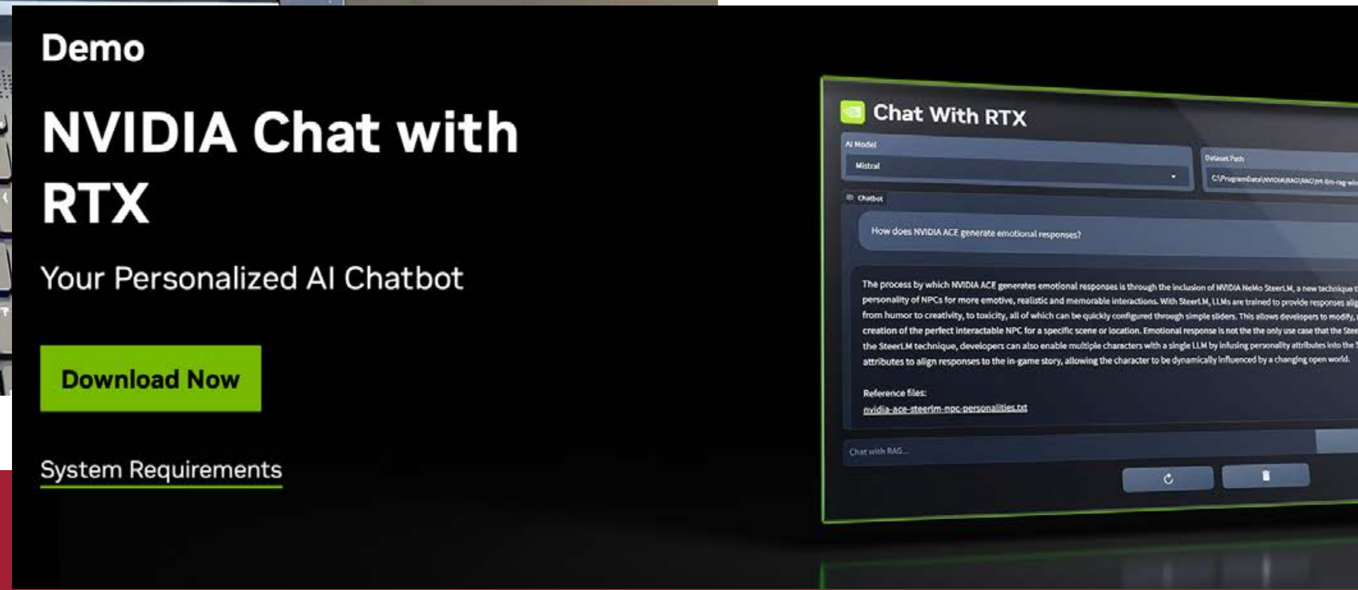
Activation-awareness: preserve the salient weight channel by scaling according to the activation magnitude.



(a) RTN quantization (**PPL** 43.2) (b) Keep 1% salient weights in FP16 (**PPL** 13.0) (c) Scale the weights before quantization (**PPL** 13.0)

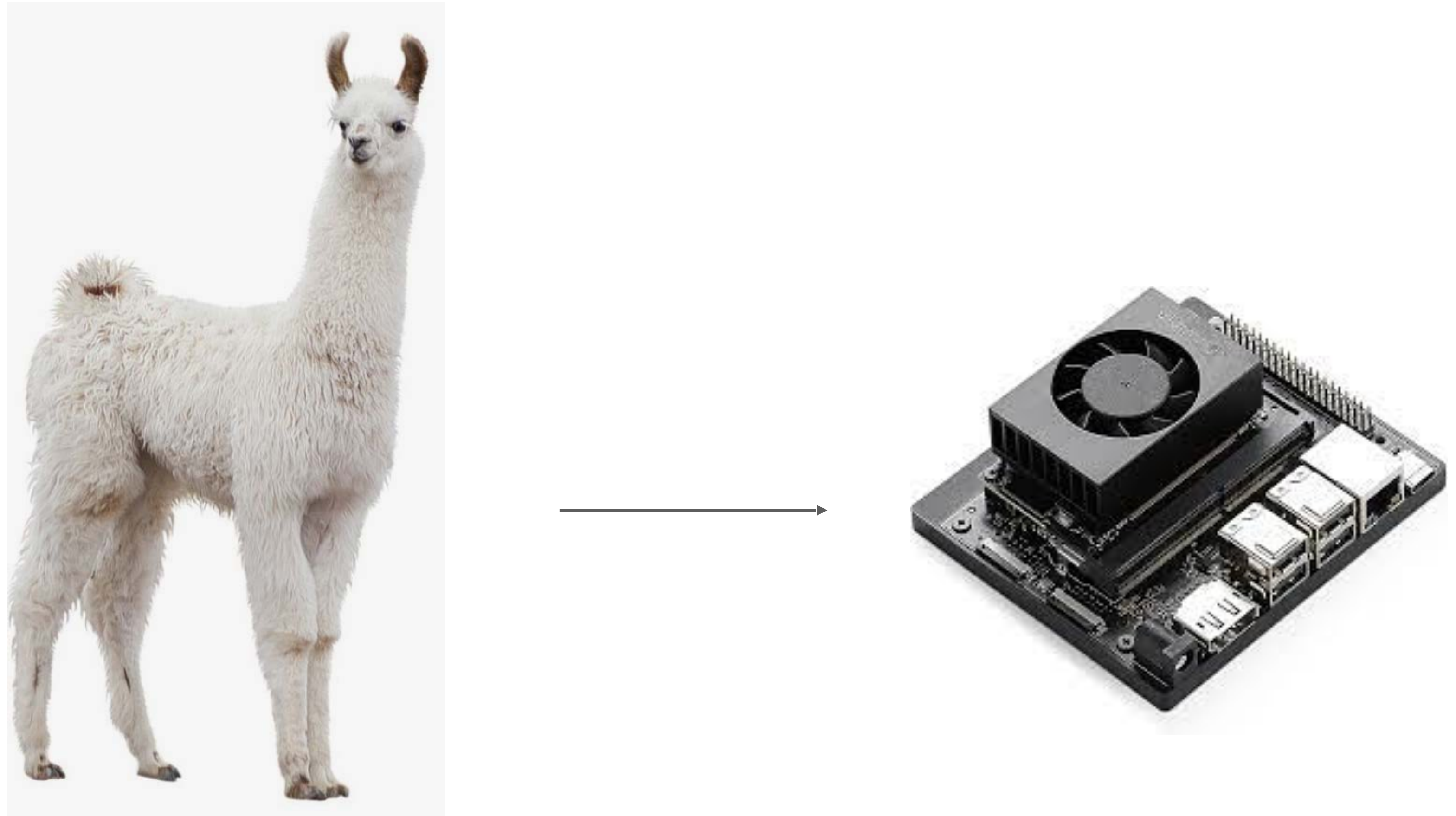




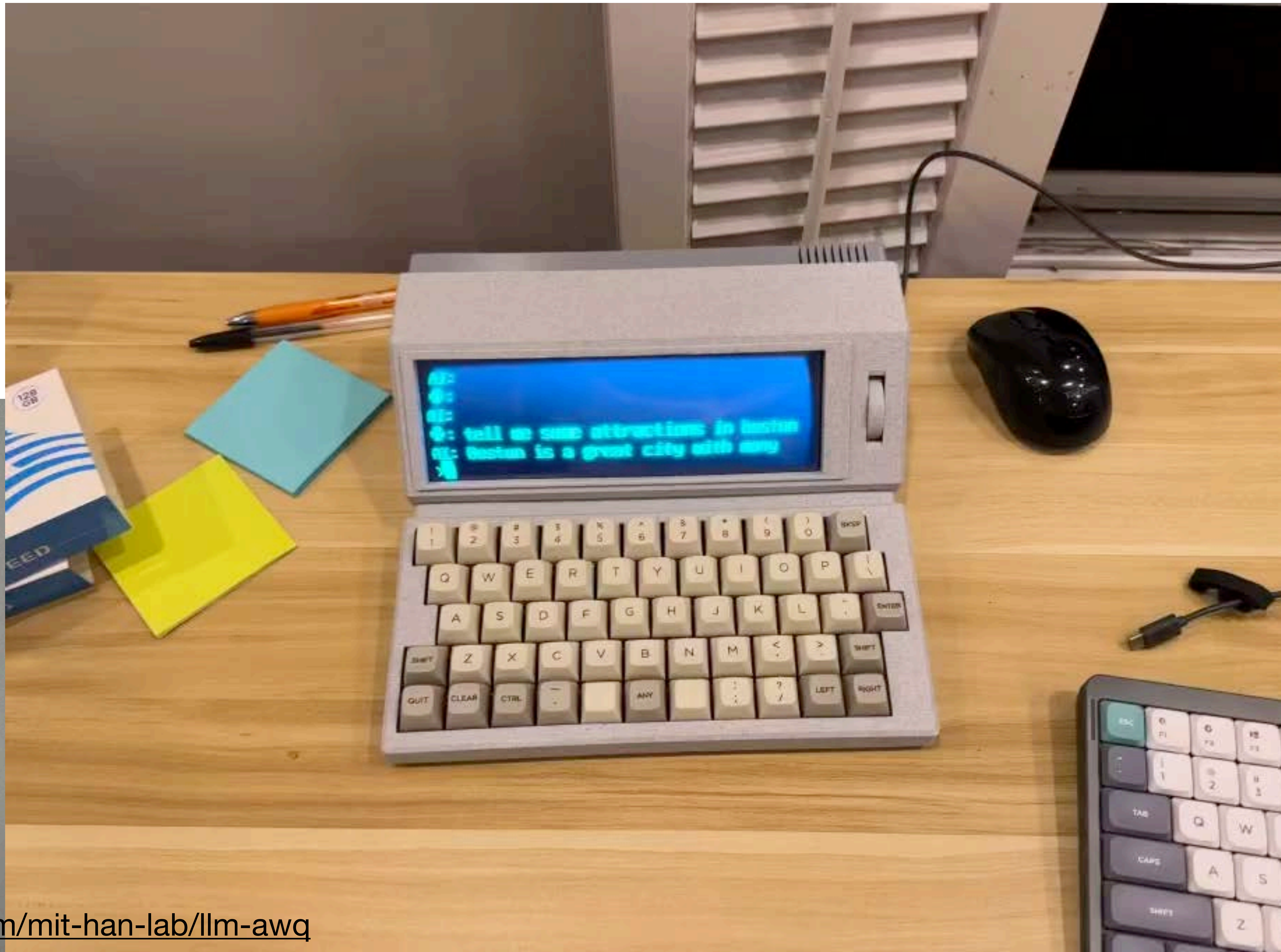


AWQ [Lin et al., MLSys 2024]

TinyChat



- Deploying LLM on the edge is useful: running copilot services (code completion, office, game chat) locally on laptops, cars, robots, and more. Protect the privacy. These devices are **resource-constrained**, **low-power** and sometimes **do not have access to the Internet**.



https://github.com/mit-han-lab/llm-awq

AWQ for Cloud LLM

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration

Goal: deploy LLM on the cloud

Challenge: LLM is too big (Falcon-180B) to fit GPU memory (141GB of H200).

Our Solution: 4bit weights, fp16 activation, fp16 arithmetic.

Activation-awareness: preserve the salient weight channel by scaling according to the activation magnitude.

Key Features

TensorRT-LLM contains examples that implement the following features.

- Multi-head Attention(MHA)
- Multi-query Attention (MQA)
- Group-query Attention(GQA)
- In-flight Batching
- Paged KV Cache for the Attention
- Tensor Parallelism
- Pipeline Parallelism
- INT4/INT8 Weight-Only Quantization (W4A16 & W8A16)
- SmoothQuant
- GPTQ
- AWQ
- FP8
- Greedy-search
- Beam-search
- RoPE

https://github.com/NVIDIA/TensorRT-LLM/

Falcon-180B on a single H200 GPU with INT4 AWQ, and 6.7x faster Llama-70B over A100

H200's large capacity & high memory bandwidth, paired with TensorRT-LLM's optimizations, maximizes inference performance.

Falcon-180B on a single H200 with INT4 AWQ

Falcon-180B, one of the largest & most accurate open source models available, can run on a *single* H200 GPU.

The 141GB of memory on H200, paired with TensorRT-LLM running INT4 AWQ with FP8, allows for the entire large language model to fit on a single GPU, where previously eight A100s were required. H200 Falcon-180B provides up to **800** tok/s and retains high accuracy.

Model Performance: H200's large capacity & high memory bandwidth, utilizing INT4 AWQ to reduce memory footprint, allows for great performance on Falcon-180B on a single GPU.

https://github.com/NVIDIA/TensorRT-LLM/blob/main/docs/source/blogs/Falcon180B-H200.md

Impact of SmoothQuant and AWQ

NVIDIA	TensorRT-LLM	https://github.com/NVIDIA/TensorRT-LLM#key-features
IBM	Granite	IBM's internal code model, Granite, utilizes AWQ for quantization.
Berkeley UNIVERSITY OF CALIFORNIA	vLLM	https://github.com/vllm-project/vllm/blob/main/vllm/model_executor/layers/quantization/awq.py
	lm-sys/FastChat	https://github.com/lm-sys/FastChat/blob/main/docs/awq.md
	Transformer Quantization API	https://huggingface.co/docs/transformers/main_classes/quantization
	lmdeploy	https://github.com/InternLM/lmdeploy/blob/main/lmdeploy/lite/quantization/awq.py
FriendliAI		https://friendli.ai/blog/Unlocking-Efficiency-of-Serving-LLMs-with-Activation-aware-Weight-Quantization-AWQ-on-PeriFlow/
replicate		https://github.com/replicate/vllm-with-loras/blob/main/vllm/model_executor/quantization_utils/awq.py

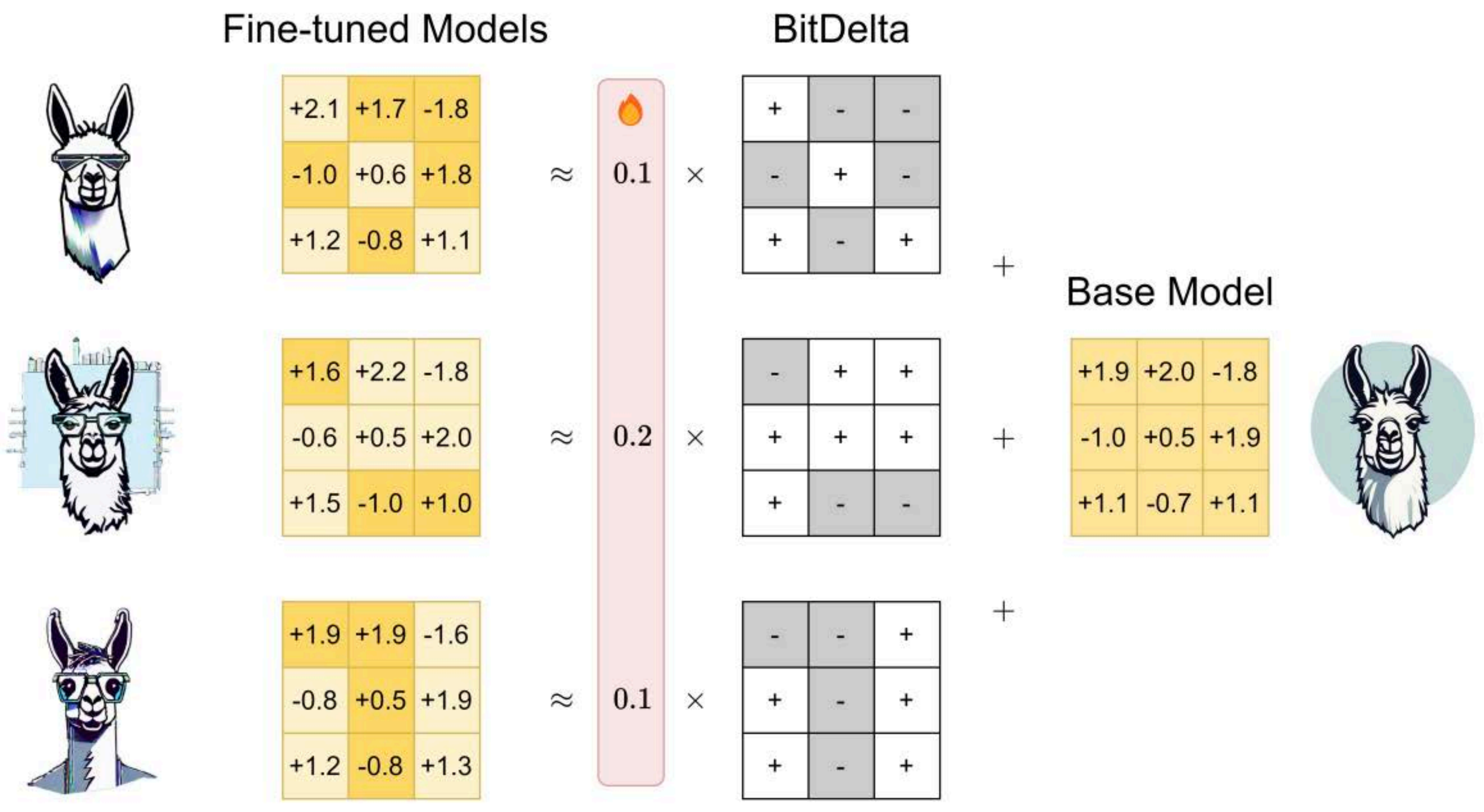
Bit-Delta

Your Fine-Tune May Only Be Worth One Bit

Goal: efficient LLM finetuning with low precision

Intuition: fine-tuning adds less new information to the model, and is thus more compressible.

Our Solution: quantizes the weight delta down to 1 bit without compromising performance, finetuning the scaling factor (per tensor)



$$\#\text{params} \times \#\text{models} \times 16\text{bits} \quad \Rightarrow \quad \#\text{params} \times (\#\text{models} \times 1\text{bit} + 16\text{bits})$$

- Weight delta: $\Delta = W_{\text{fine}} - W_{\text{base}}$
- Binarized delta: $\hat{\Delta} = \alpha \odot \text{Sign}(\Delta)$

$$\text{Sign}(W_{ij}) = \begin{cases} +1, & \text{if } W_{ij} > 0, \\ -1, & \text{if } W_{ij} \le 0, \end{cases}$$

- To minimize the L_2 quantization error:

$$\left\|\Delta - \hat{\Delta}\right\|_2^2 = \sum_{ij} (|W_{ij}| - \alpha)^2$$

- We initialize α as

$$\alpha = \frac{1}{nm} \sum_{ij} |W_{ij}|.$$

- We further optimize the scales by performing model distillation:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathbb{E}_{x \sim \mathbf{X}} \left[\|\mathbf{Z}_{\text{fine}}(x) - \mathbf{Z}_{\text{bin}}(x; \boldsymbol{\alpha})\|^2 \right]$$

- We distill on the C4 dataset, using 800 samples of length 128. For 70B models, the distillation roughly takes 10 minutes.

[Liu et al., arXiv 2024]

Bit-Delta

Your Fine-Tune May Only Be Worth One Bit

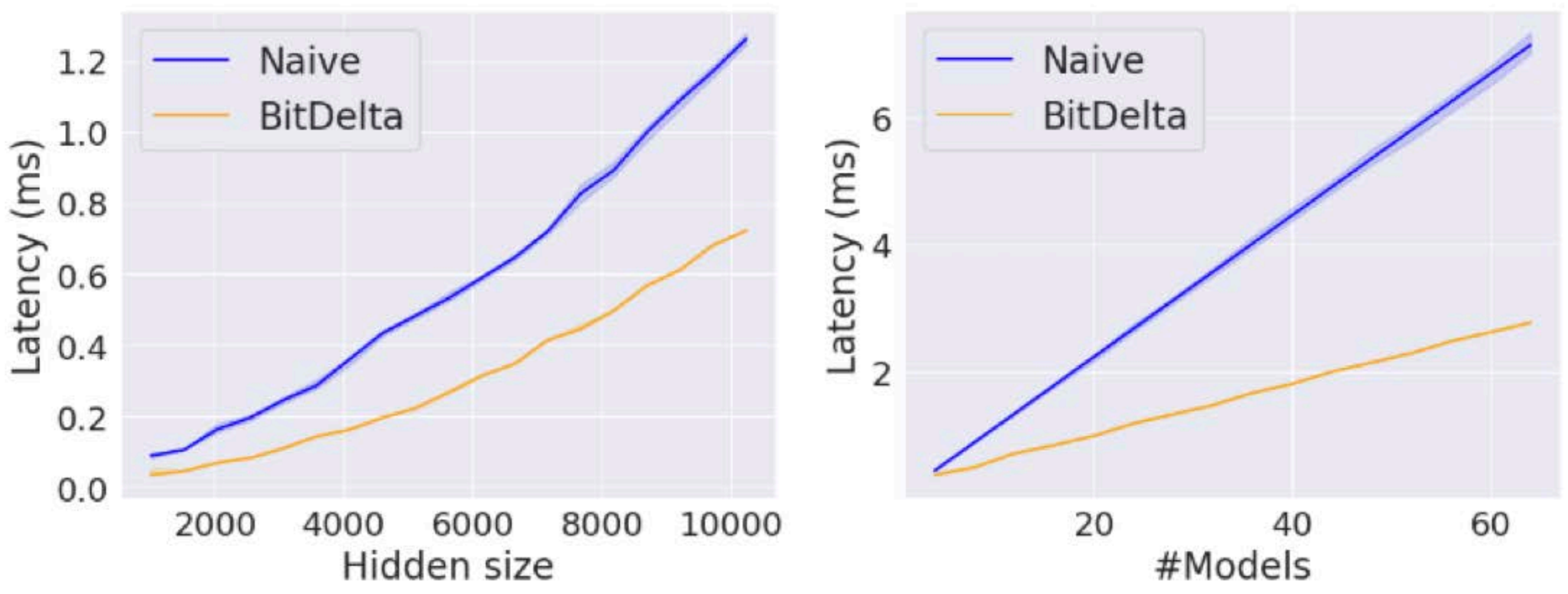
Goal: efficient LLM finetuning with low precision

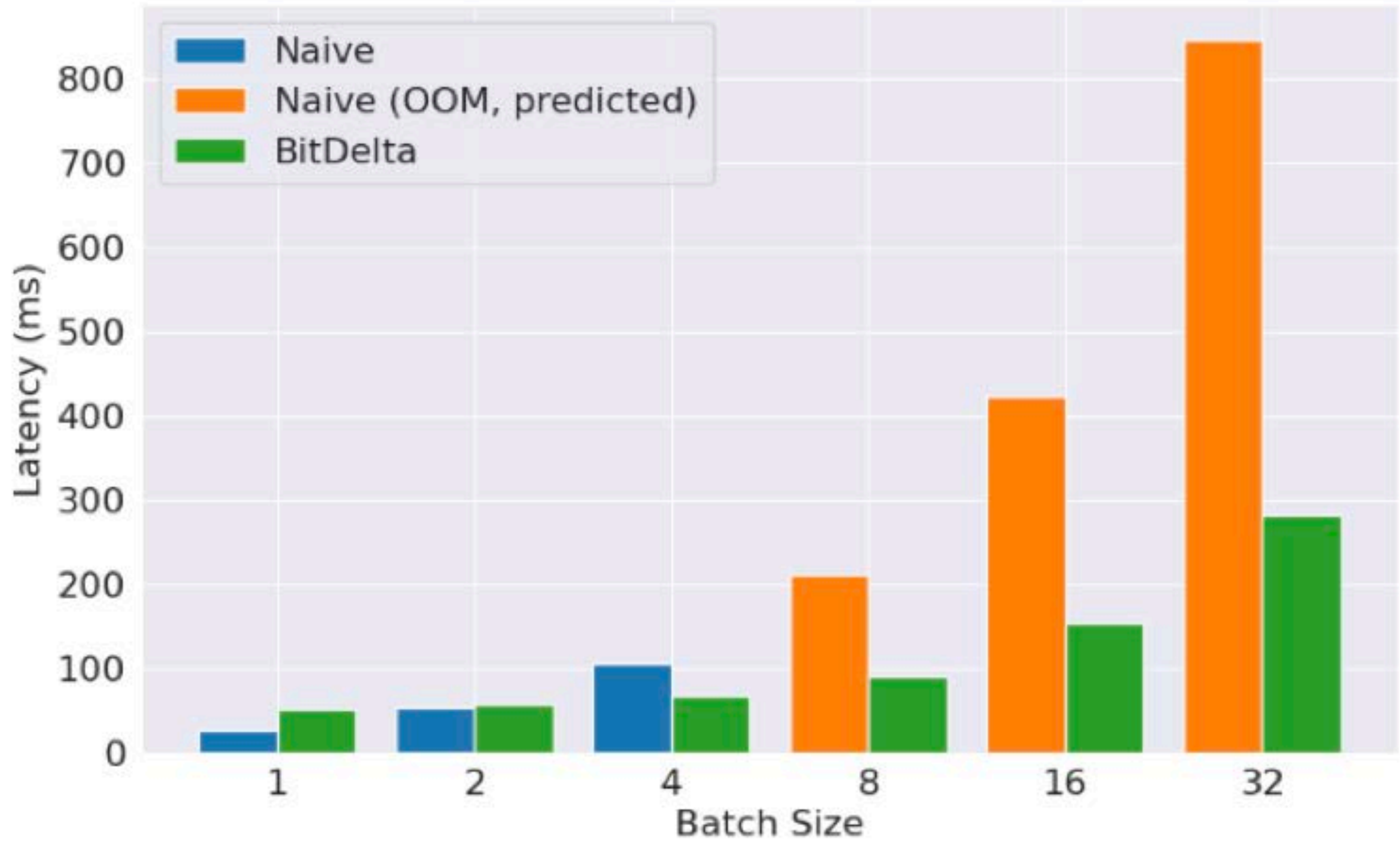
Intuition: fine-tuning adds less new information to the model, and is thus more compressible.

Our Solution: quantizes the weight delta down to 1 bit without compromising performance, finetuning the scaling factor (per tensor)

Base Model	Size	ΔSize	Comp. Factor
Llama 2-7B	13.48 GB	1.24 GB	10.87
Llama 2-13B	26.03 GB	2.09 GB	12.45
Llama 2-70B	137.95 GB	8.95 GB	15.41
Mistral-7B v0.1	14.48 GB	1.30 GB	11.14

The more you serve, the more you save!





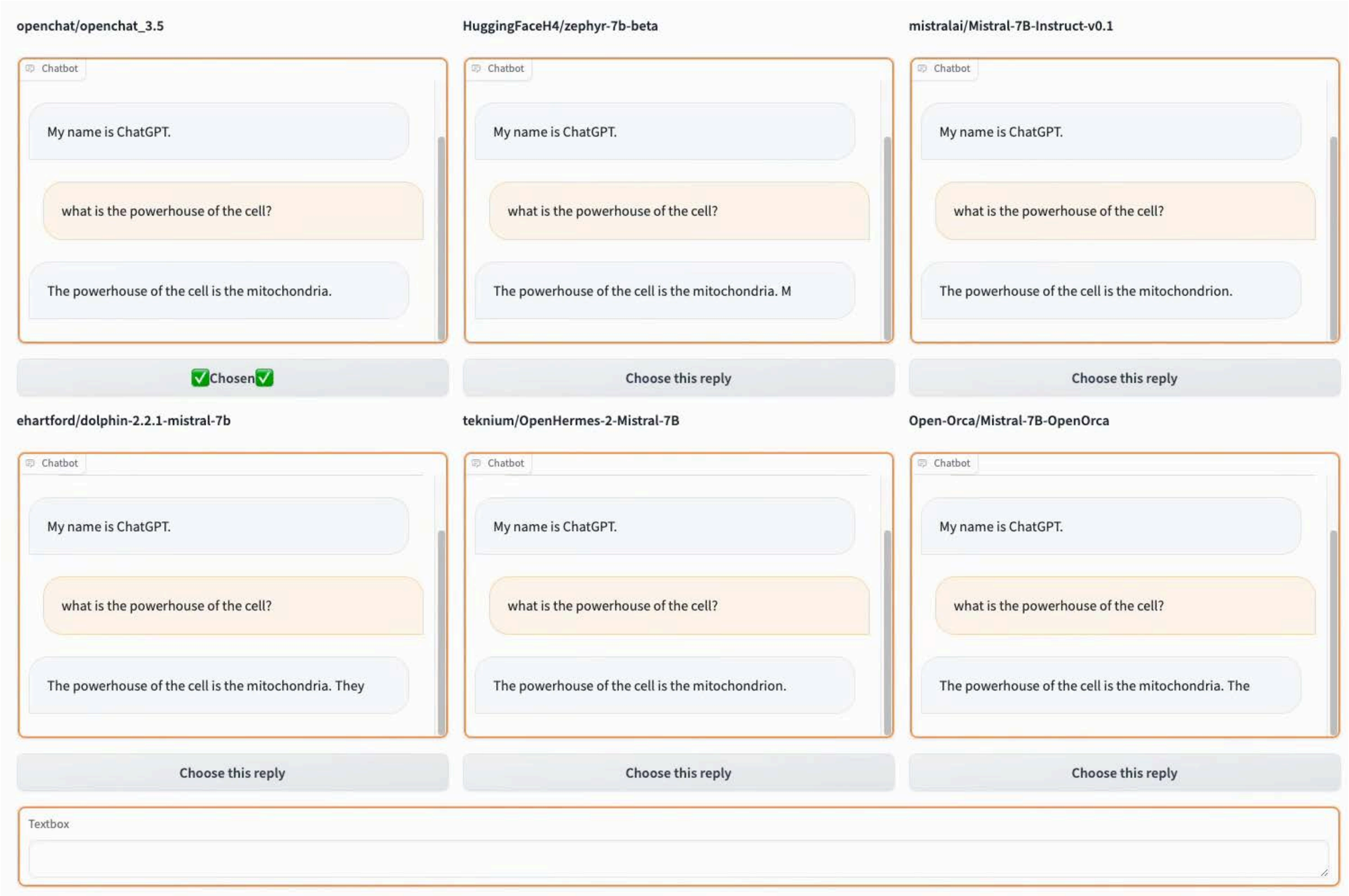
End-to-end decoding latency, Llam2-7B. We implement a fused binary GEMM kernel that allows us to calculate Delta * X in a batched setting while keeping the 1-bit deltas quantized. This kernel fuses the dequantization operation with the GEMM calculation, reducing the data movement overhead by a large factor.

[Liu et al., arXiv 2024]

Multi-tenant Serving with BitDelta

[Liu et al., arXiv 2024]

Your Fine-Tune May Only Be Worth One Bit



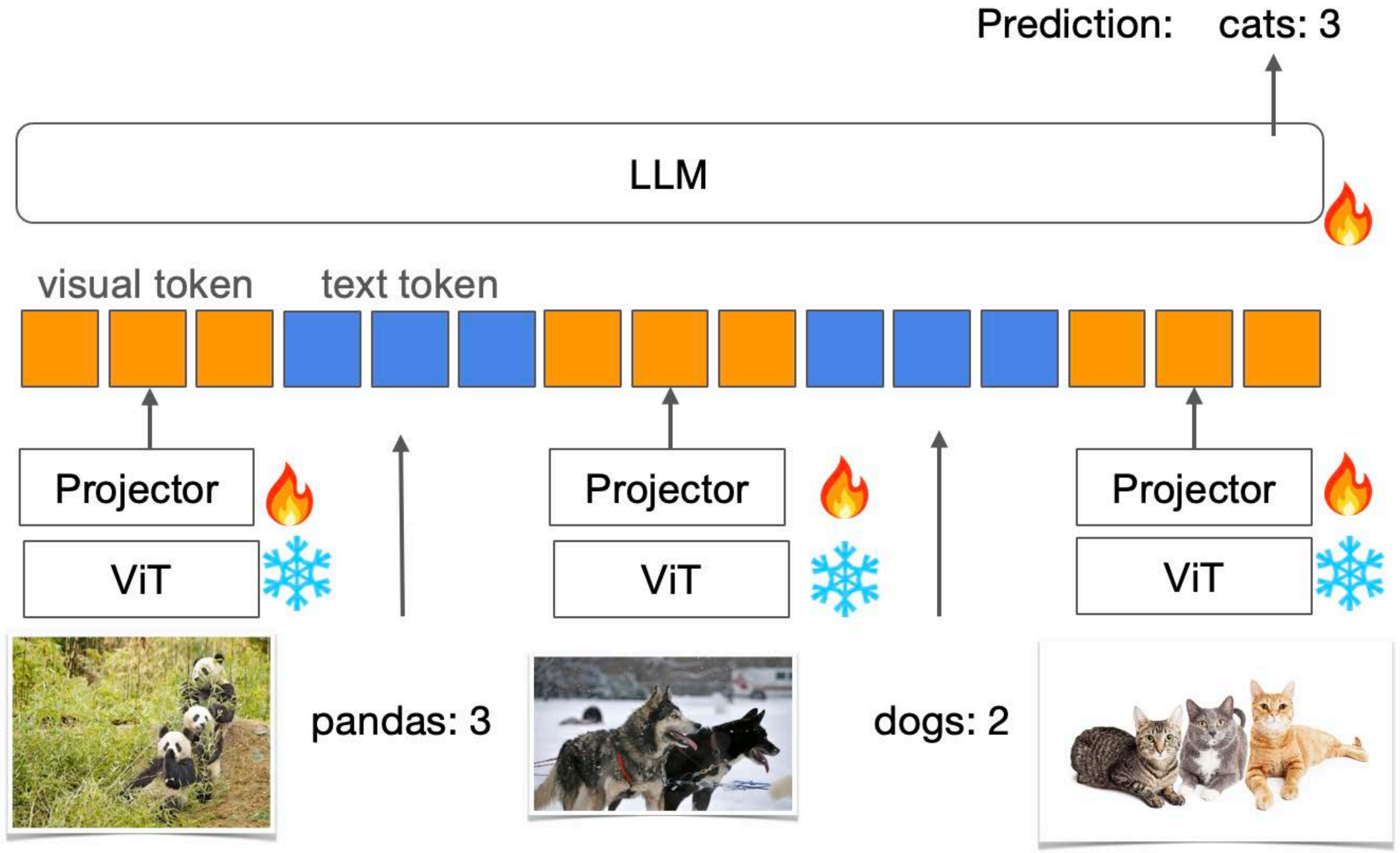
Multi-modality

Visual Language Model

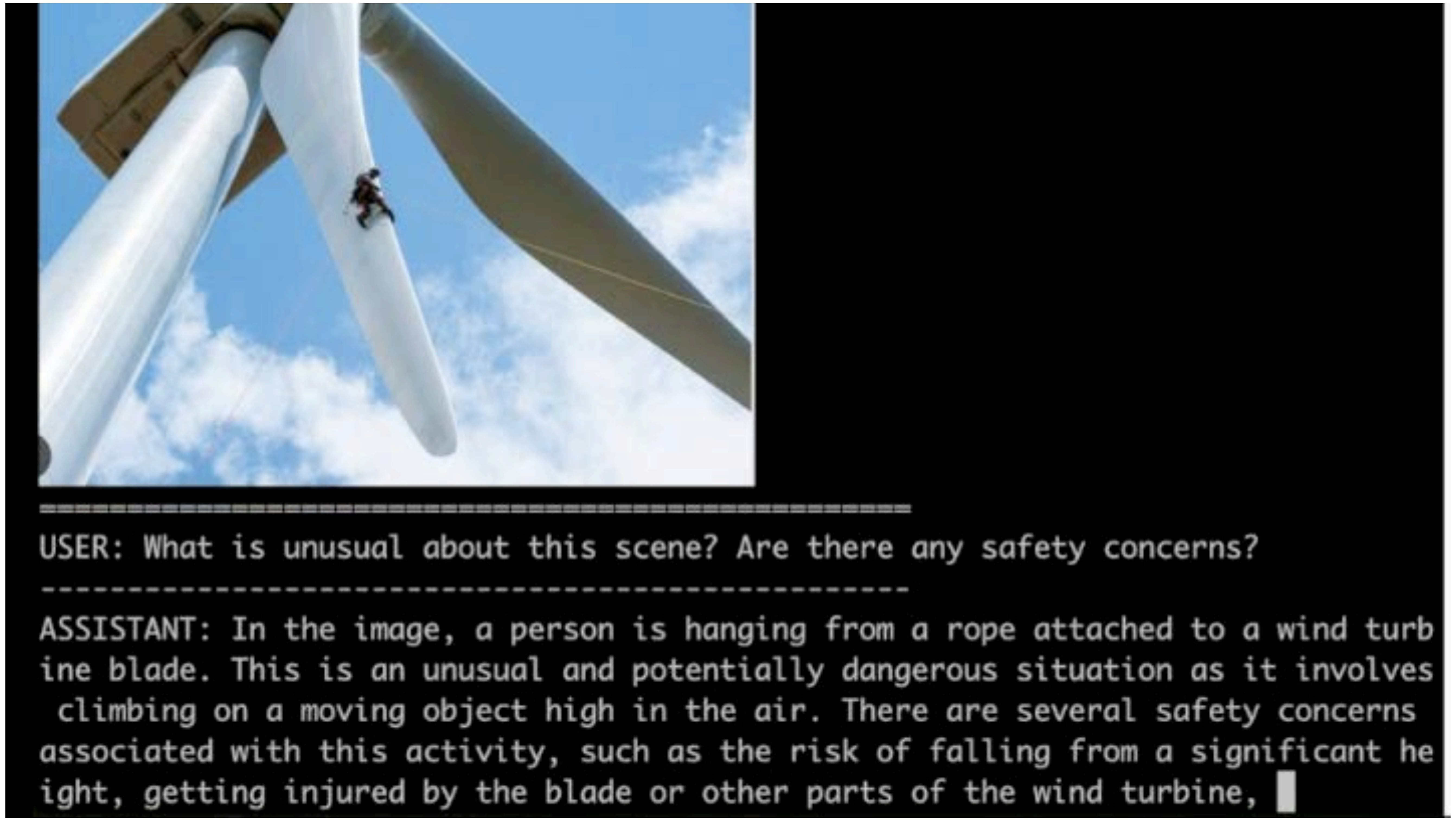
Goal: multi-modal LLM, enhance visual reasoning by language model, enable in-context learning and reasoning across images

Challenge: learn from visual inputs without destroying the LLM. Computationally heavy to deploy on the edge.

Our Solution: VILA provides efficient recipe from data curation to training to deployment for VLM. Interleaved image-text pretraining enables in-context learning.



Augments the language model with interleaved image-text



VILA deployed on Jetson Orin

[Lin et al., CVPR 2024]

In Context Learning and Multi-Image Reasoning

The company is famous for its search engine.

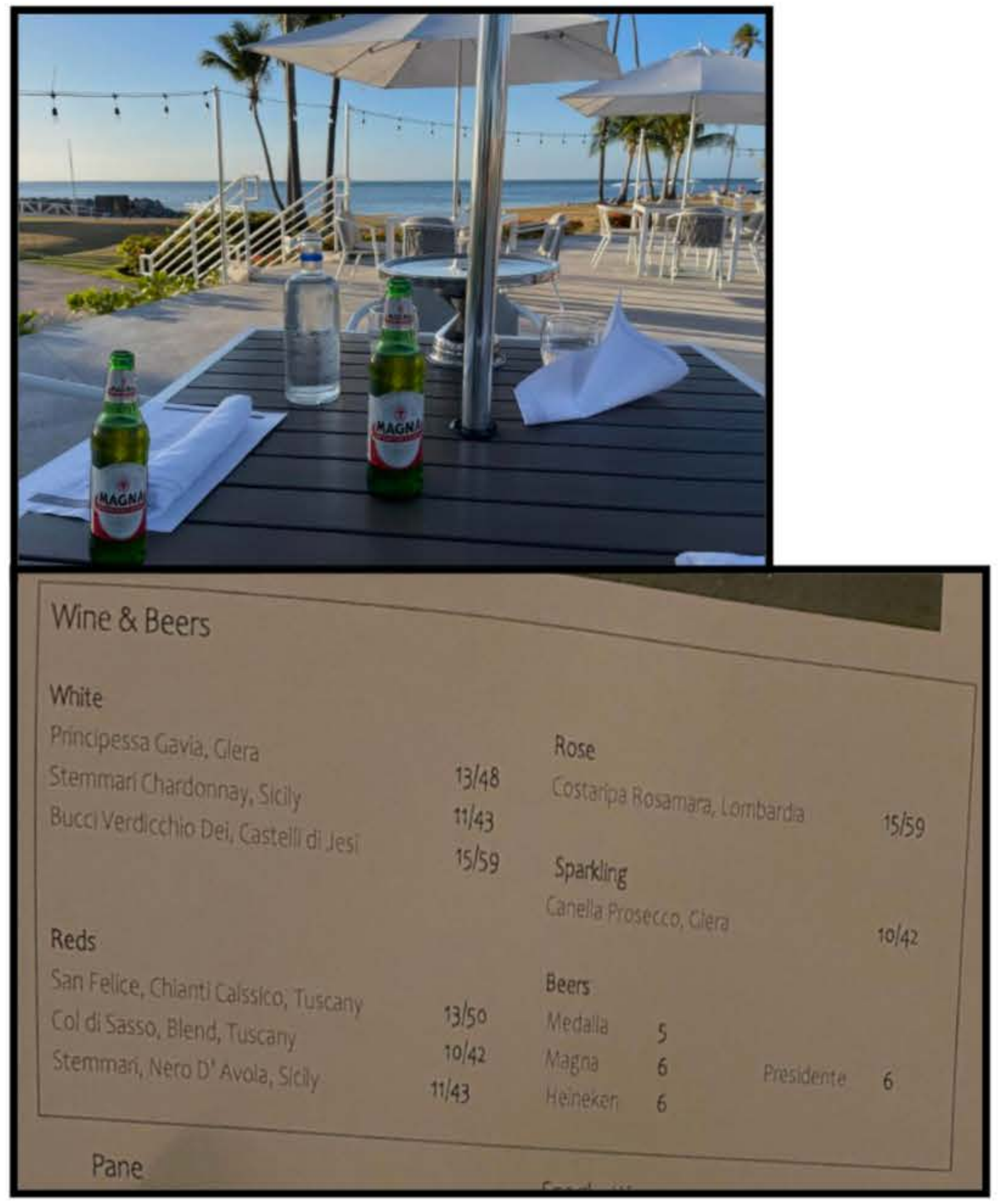
The company is famous for the operating system.

The company is famous for iPhone and Mac.

Pred: The company is famous for its graphics processing units (GPUs)

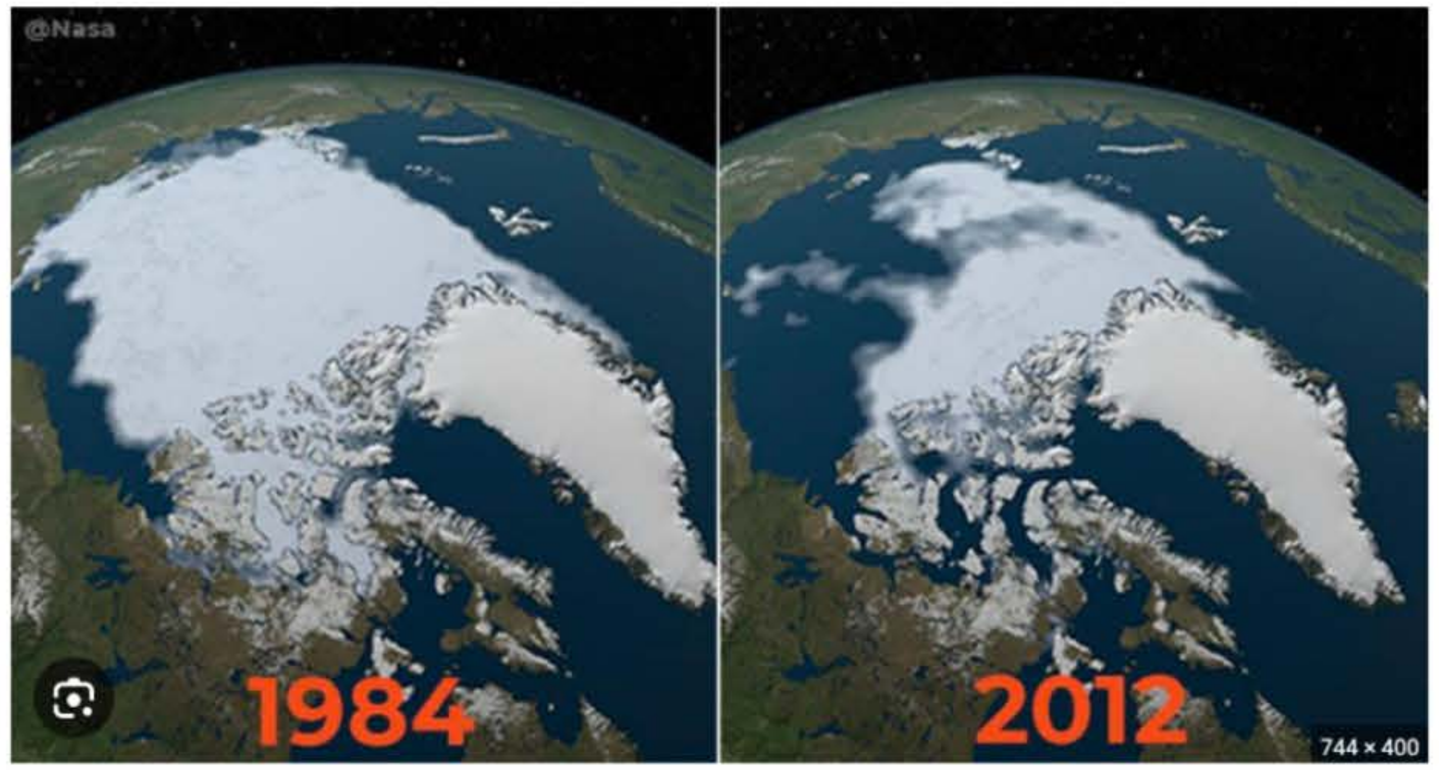
context

prediction



Prompt: Photo: <img1>\
much should I pay for all t
according to the price on t
Answer: According to the
on the table is $6.

Prompt: … + ***Think step***
Answer: The beer on the
priced at $6.00. There are
table. Therefore, the total
$6.00 x 2 = $12.00.



User: What is the implication of temperature based on this image?

VILA: The image shows a comparison of the Arctic ice cover in 1984 and 2012. The implication of this image is that the Arctic ice cover has significantly decreased over the years, which is a result of climate change. The reduction in ice cover can have various implications, such as increased exposure of the Arctic environment to warmer temperatures, which can lead to the melting of sea ice, loss of habitat for polar bears and other Arctic species, and potential disruptions to the global climate system. This image serves as a visual representation of the impact of climate change on the Arctic environment and the need for action to mitigate its effects.

Sparsity

Dictionary Definition

Merriam-Webster

Sparse

: of **few** and **scattered** elements

2	0	4	5	0	2	0	8
0	7	0	0	2	1	3	0
0	7	0	4	0	0	0	0
0	9	0	6	0	1	2	3

Streaming LLM

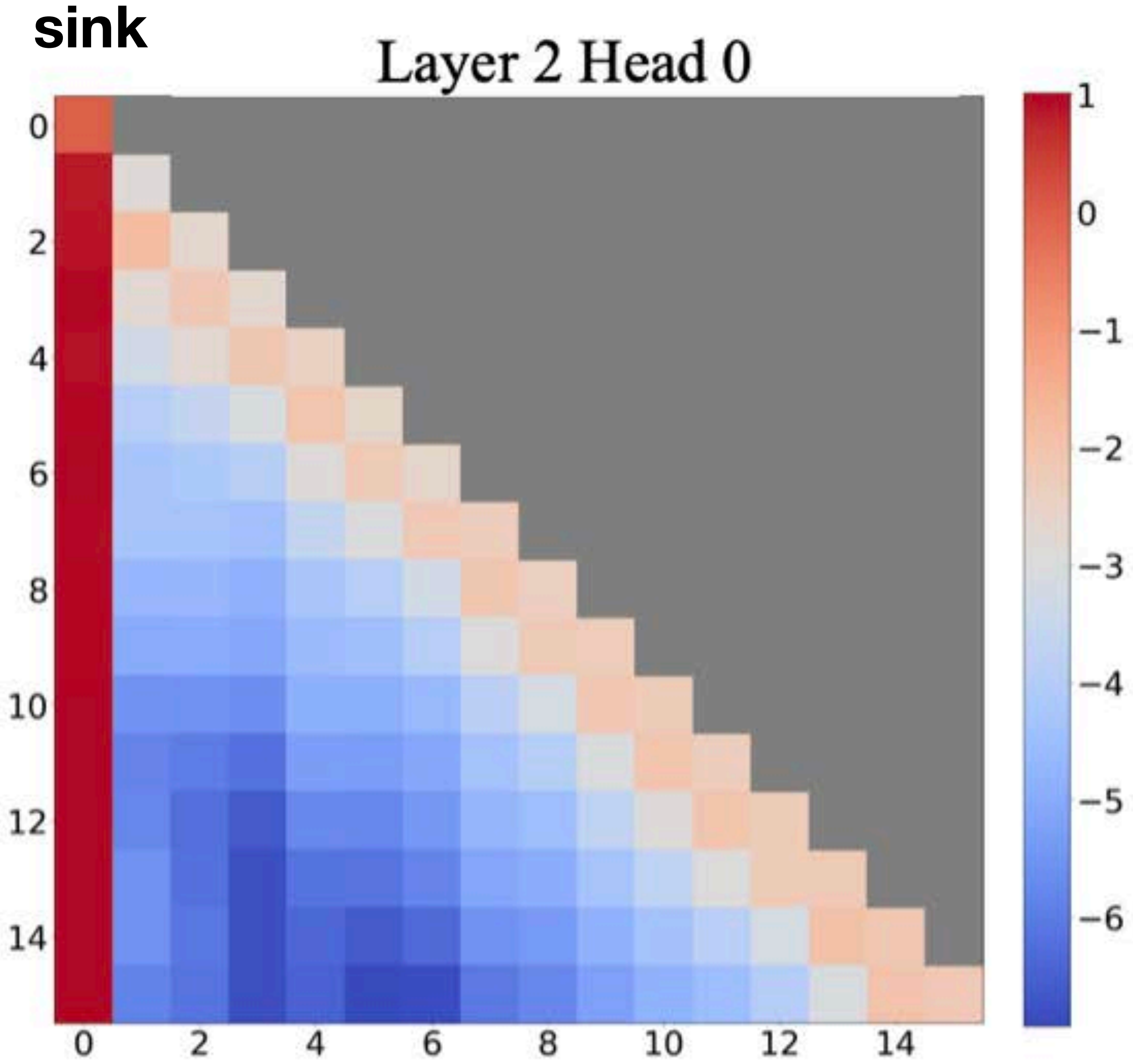
Enable long conversations in non-stop streaming applications

Goal: long text generation in streaming LLM applications such as multi-round dialogues and non-stop interaction.

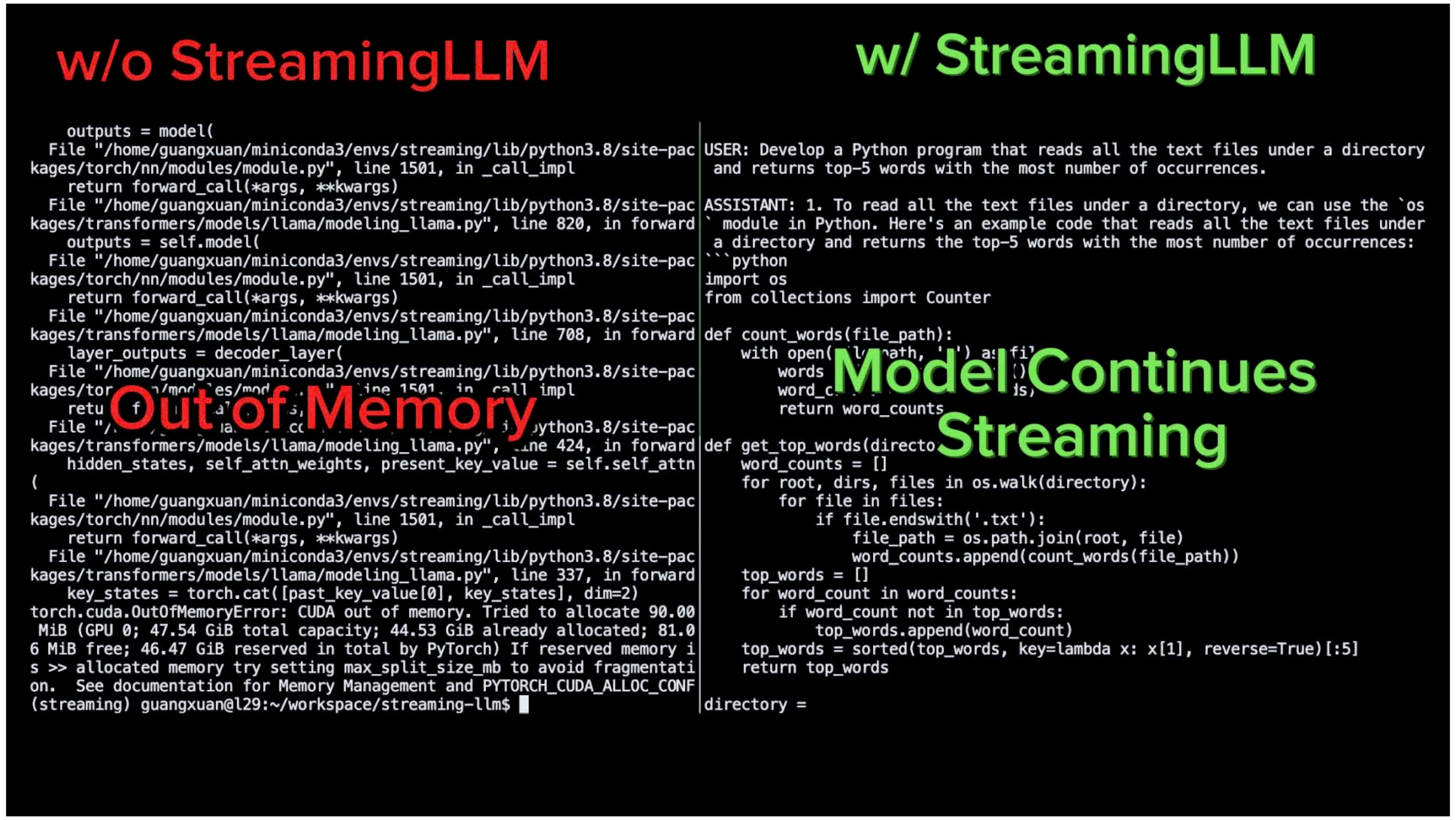
Challenge: KV cache grows linearly with the conversation => runs out of memory as the conversation goes long; perplexity explodes after the sequence length exceeds the KV cache size (when the first token is evicted).

Our Solution: StreamingLLM always keep the “attention sink” tokens in the KV cache; and use windowed KV cache.

attention sink



LLM heavily attends to the initial token: the “attention sink”.



[Xiao et al., ICLR 2024]

StreamingLLM on iPhone

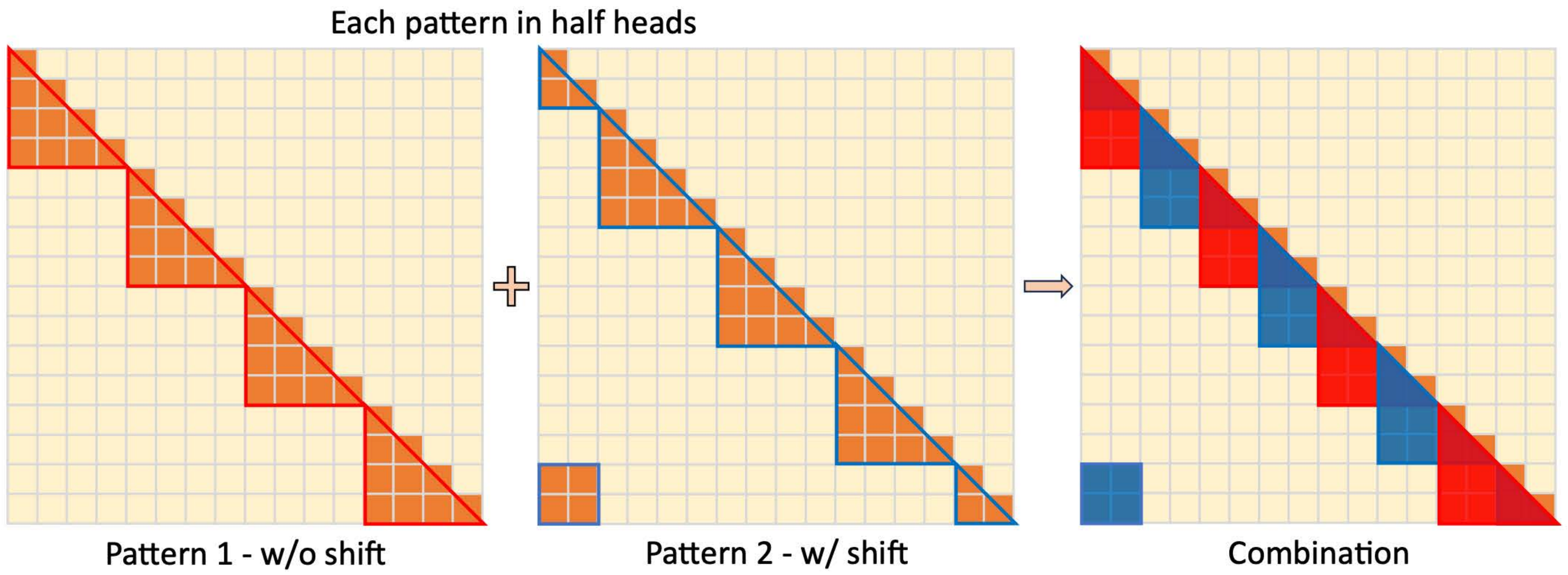
Long-Lora

Efficient Fine-tuning of Long-Context LLMs

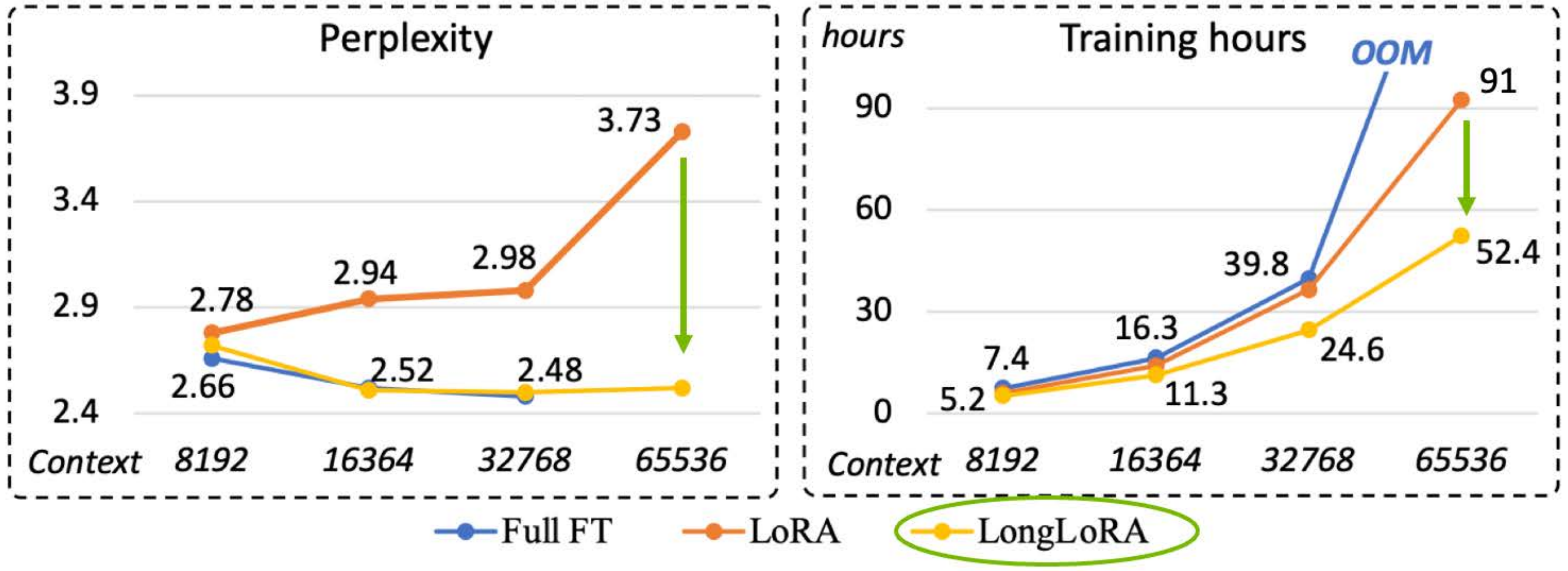
Goal: Let LLM remember more; extend the context length.

Challenge: O(N^2) computation and memory complexity for attention. For longer context, attention becomes expensive.

Our Solution: LongLoRA invented “shifted, sparse attention” to enable longer context length at low finetuning cost.



shifted, sparse attention: O(N^2) => O (N*M)



Lower perplexity, shorter finetuning time

Long-Lora

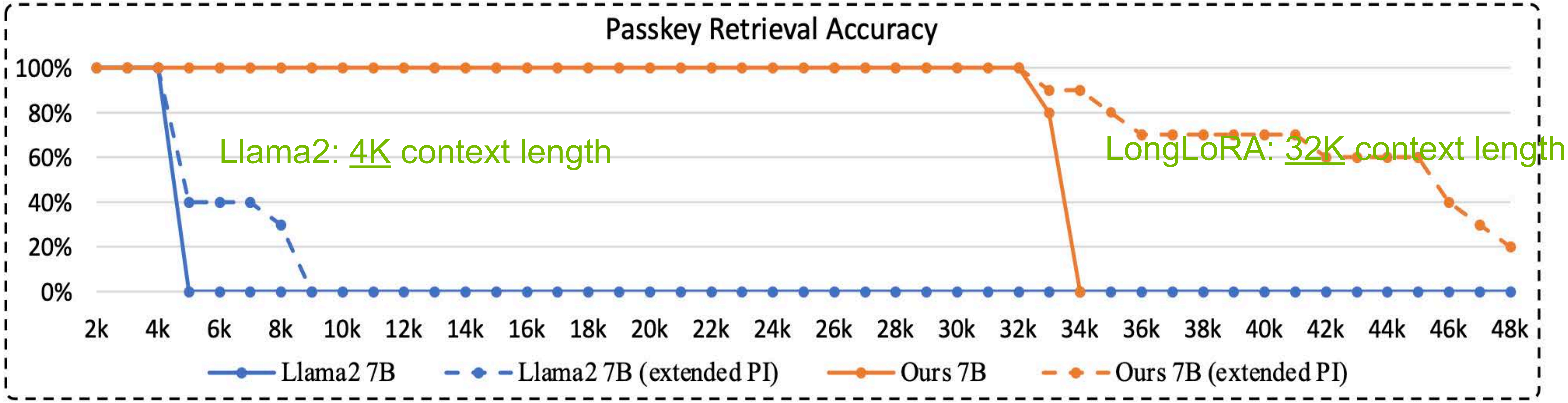
Efficient Fine-tuning of Long-Context LLMs

Goal: Let LLM remember more; extend the context length.

Challenge: O(N^2) computation and memory complexity for attention. For longer context, attention becomes expensive.

Our Solution: LongLoRA invented "shifted, sparse attention" to enable longer context length at low finetuning cost.

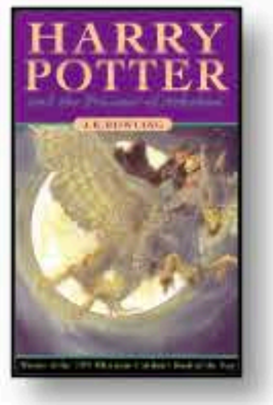
Results: on a single 8× A100 machine, longLoRA efficiently extends the context length of Llama2-7B from 4k to 100k, Llama2-70B to 32k .



[Chen et al., ICLR 2024, Oral]

Demo

LongLoRA
(ICLR'24 **Oral**)



Below is some content in the book. Memorize the content and answer my question after the book. \n The book begins. {book_content} \n Now the material ends.
Who was the person Sirius Black want to kill?

Ours-13B

Based on the book, Sirius Black did not actually want to kill anyone. The key details are:

1. Sirius Black was wrongly imprisoned for betraying Harry's parents and killing Peter Pettigrew.
2. It was actually Pettigrew who betrayed the Potters and staged his own death to frame Sirius Black.
3. When Sirius Black escaped from prison, he was initially believed to be hunting down Harry to kill him for revenge.
4. However, it was later revealed that Sirius Black actually wanted to kill Pettigrew to get revenge for framing him and betraying the Potters.
5. Sirius Black did not actually want to harm Harry. In fact, he was Harry's godfather and wanted to take Harry in once his name was cleared.

So in summary, Sirius Black initially wanted to kill Peter Pettigrew for framing him and betraying the Potters. He did not actually want to kill Harry.

Demo

LongLoRA
(ICLR'24 **Oral**)

Below is an economic outlook. Memorize the content and answer my question after the paper.
{book_content} \n Now the material ends.
Based on the material and your understanding, please predict the economy in the future.

Based on the economic outlook from 2023 to 2006, here are my predictions for the global economy in the future:

• Growth will remain moderate and uneven across countries, with advanced economies likely growing faster than emerging markets and developing economies. This trend has been seen in recent years.

• Inflation will remain elevated but start to come down from the high levels seen in 2022 and 2023 as central banks continue to raise interest rates and commodity prices stabilize. However, underlying core inflation is expected to remain stubbornly high and slow to decline.

• Recovery from the COVID-19 pandemic will continue but likely remain bumpy as new virus variants emerge periodically. Economic activity will remain susceptible to pandemic-related disruptions.

• Downside risks will continue to outweigh upside potential, including risks from higher interest rates, geopolitical tensions, financial market volatility, and slowing potential growth.

• Structural reforms will be needed to boost productivity, support inclusive growth, and facilitate the transition to a green economy. Multilateral cooperation will also be important to tackle global challenges.

• Central banks will gradually normalize monetary policies but remain cautious and data-dependent to avoid overtightening and curtailing the economic recovery. Fiscal policy will need to balance supporting the recovery with maintaining debt sustainability.

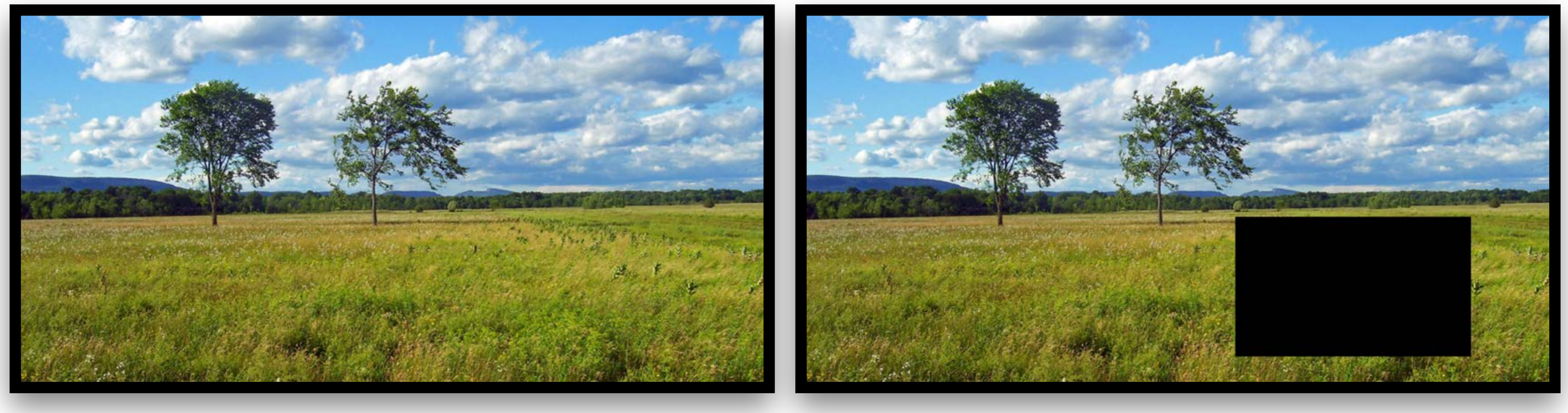
In summary, while the global economy is expected to continue expanding in the coming years, the pace of growth will likely be moderate and below-trend. Considerable challenges and risks persist that policymakers will need to navigate carefully through multilateral cooperation and structural reforms.

SIGE: Sparse Incremental Generative Engine

Spatially Sparse Inference for Conditional GANs and Diffusion Models

Designers only edit part of the image at a time; can we save the computation by regenerate only edited pixels?

A photograph of a horse on a grassland.



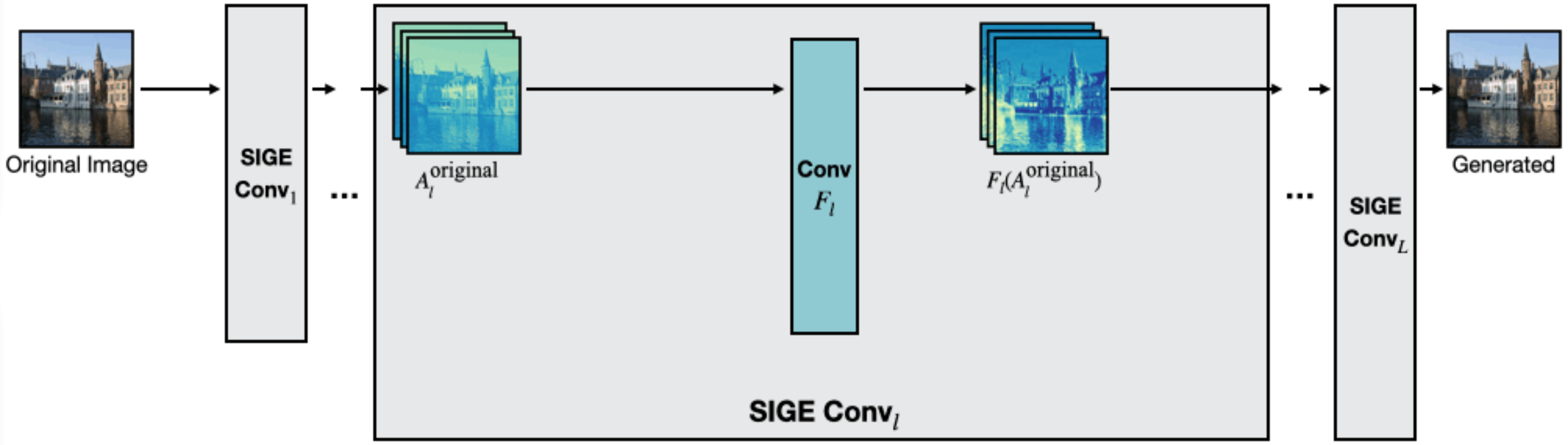
Original | 11.6% Masked

Stable Diffusion: 1855GMACs 369ms

Ours: 514G (3.6×) 95.0ms (3.9×)

Image Inpainting Latency Measured on NVIDIA RTX 3090 

Tiling-based Sparse Convolution

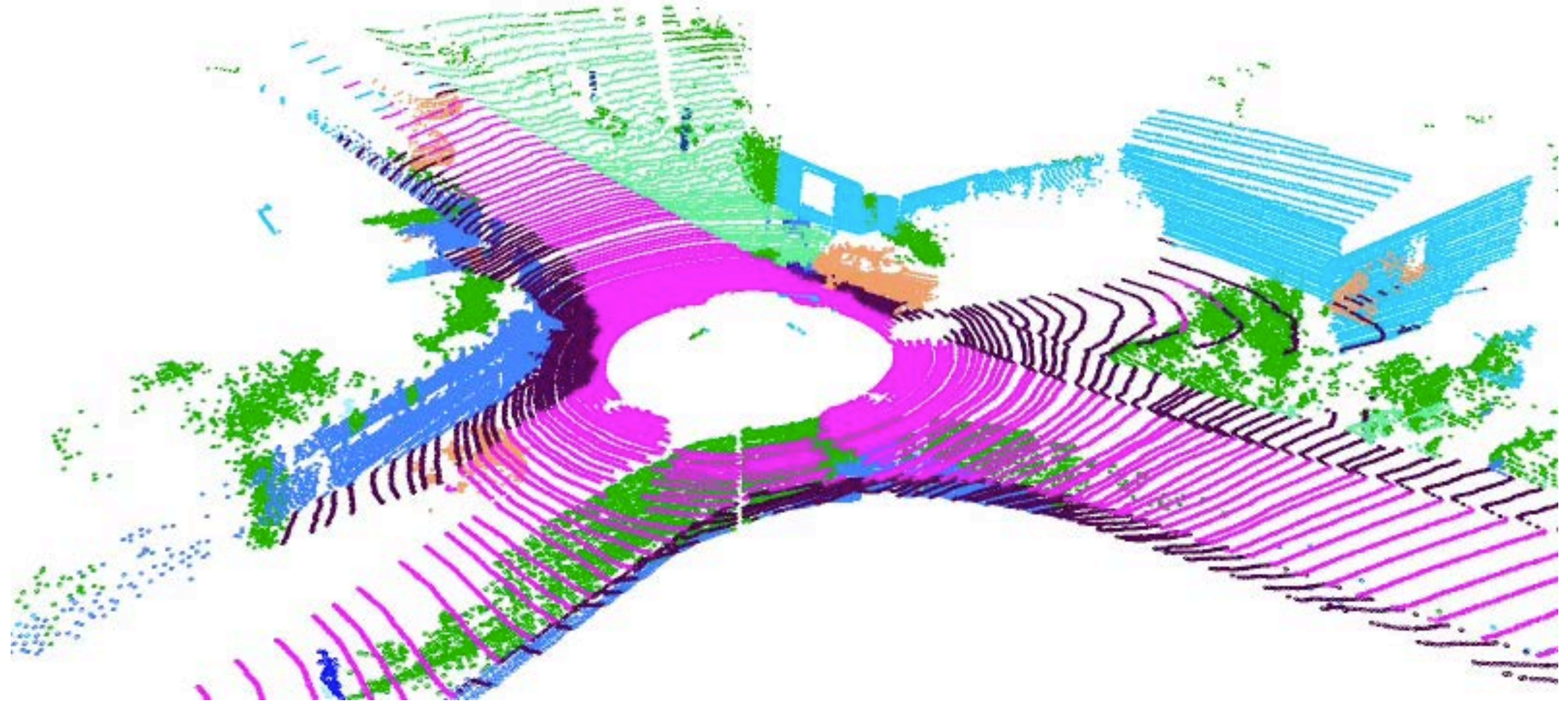


Sparsity in Autonomous Driving

PVCNN + SPVNAS
(NeurIPS'19 **Spotlight**, ECCV'20)

(~10x)

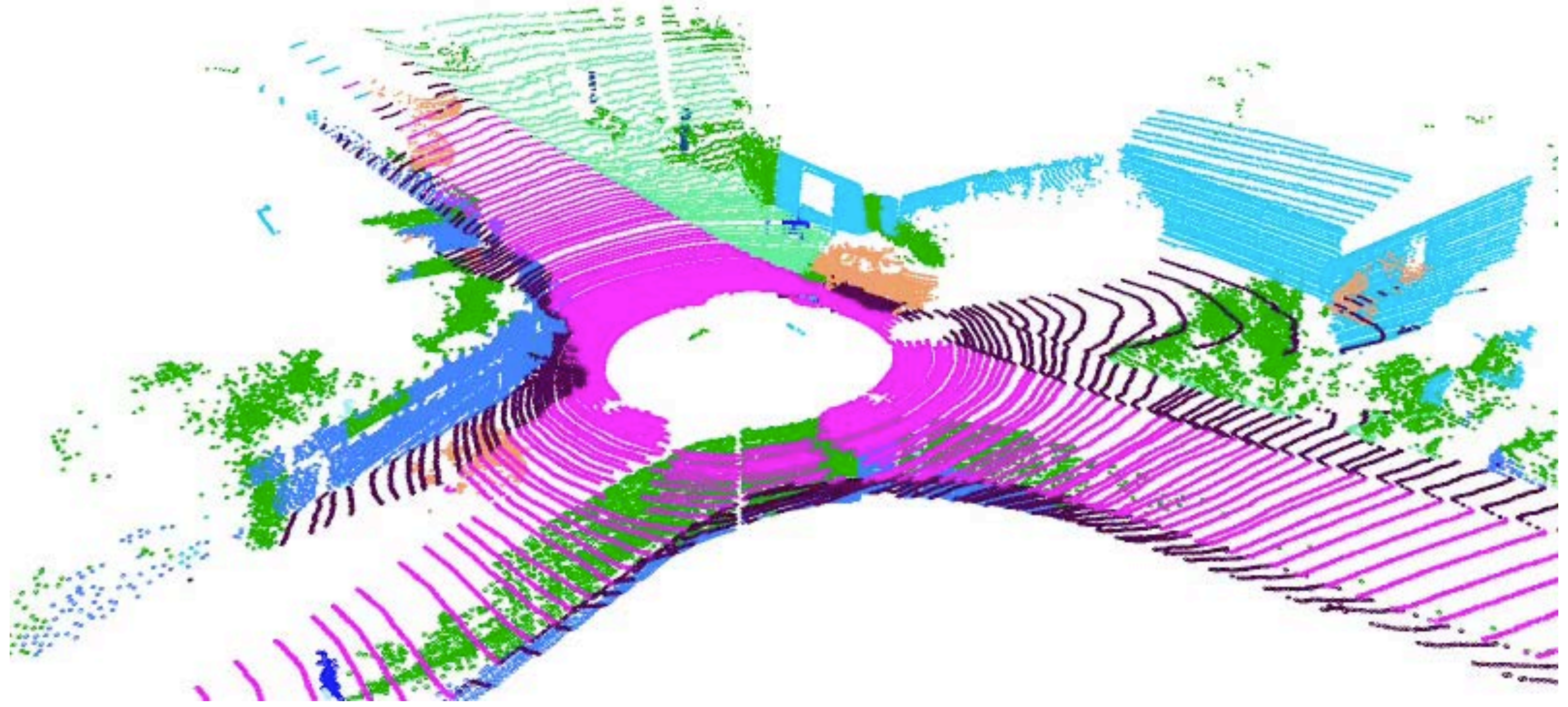
MinkowskiNet



Mean IoU: **63.1** Throughput: **3.4** FPS
(**21.7M** Params **114.0G** FLOPs)

(~4x)

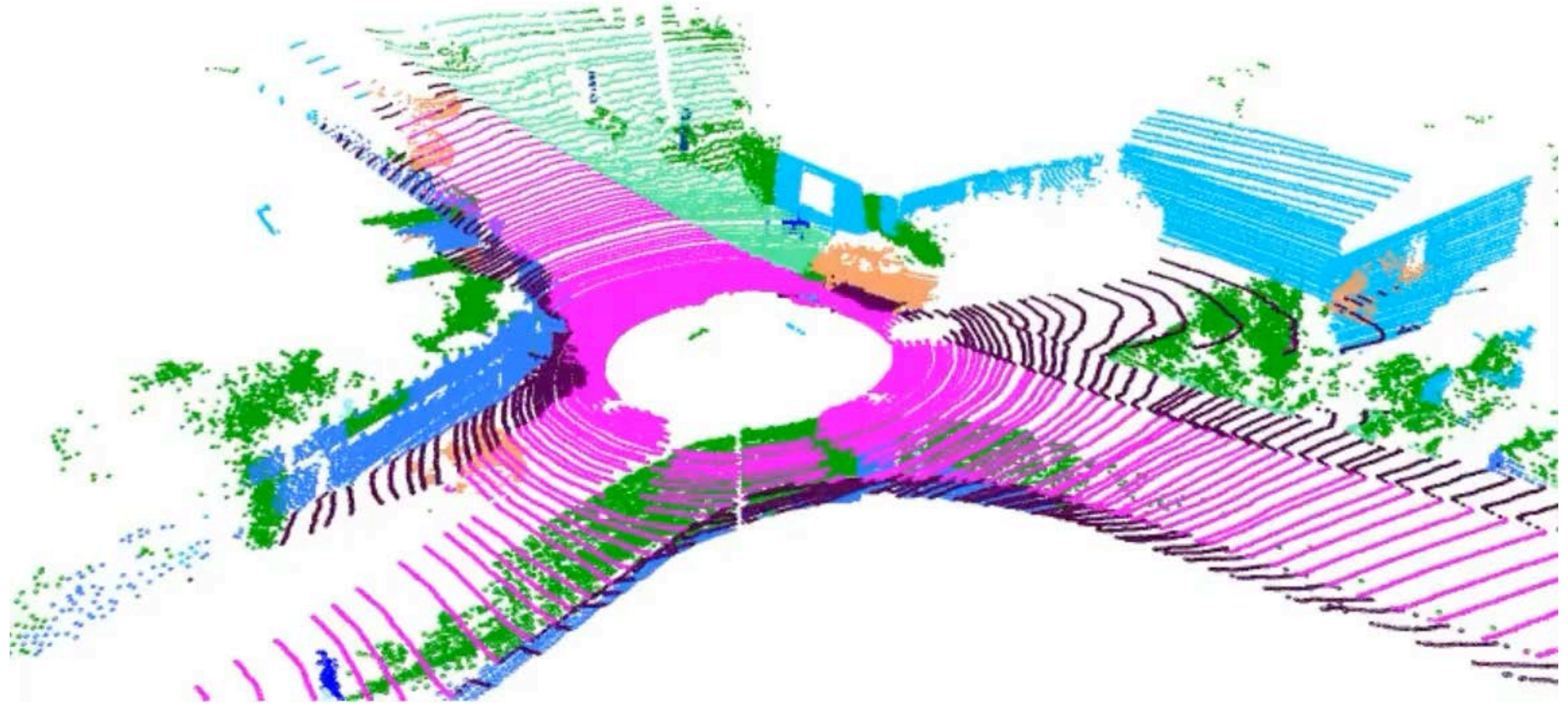
+ PVCNN, SPVNAS



Mean IoU: **63.6** Throughput: **9.1** FPS
(**2.6M** Params **15.0G** FLOPs)

(~6x)

+ Sparse System (TorchSparse)



Mean IoU: **63.6** Throughput: **12.1** FPS
(**2.6M** Params **15.0G** FLOPs)

Sparsity in Autonomous Driving

BEVFusion
(ICRA'23, **Most cited paper** in ICRA'21-23)

3D Object Detection

BEV Map Segmentation

Leaderboard

Ranked 1st	Ranked 1st	Ranked 1st	Ranked 1st
3D Detection (nuScenes)	3D Tracking (nuScenes)	3D Detection (Waymo)	3D Detection (Argoverse)
(as of 2022/6)	(as of 2022/8)	(as of 2022/11)	(as of 2023/6)

Sparsity in Autonomous Driving

BEVFusion
(ICRA'23, **Most cited paper** in ICRA'21-23)

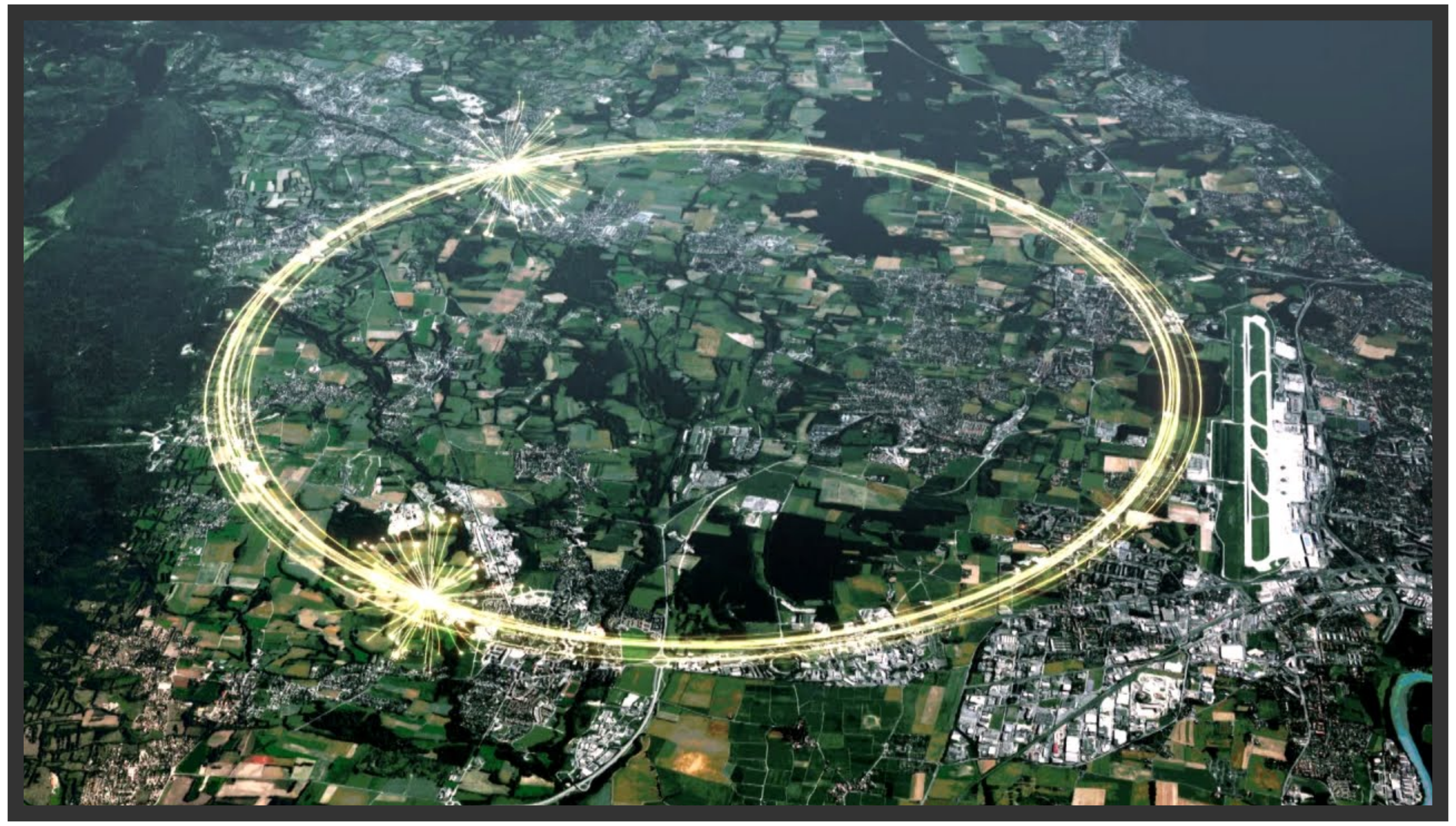
3D Object Detection

BEV Map Segmentation

Industry Adoption: NVIDIA WAYMO cruise RIVIAN

Sparsity in Scientific Discovery

Calo-SPVCNN
(FastML for Science)



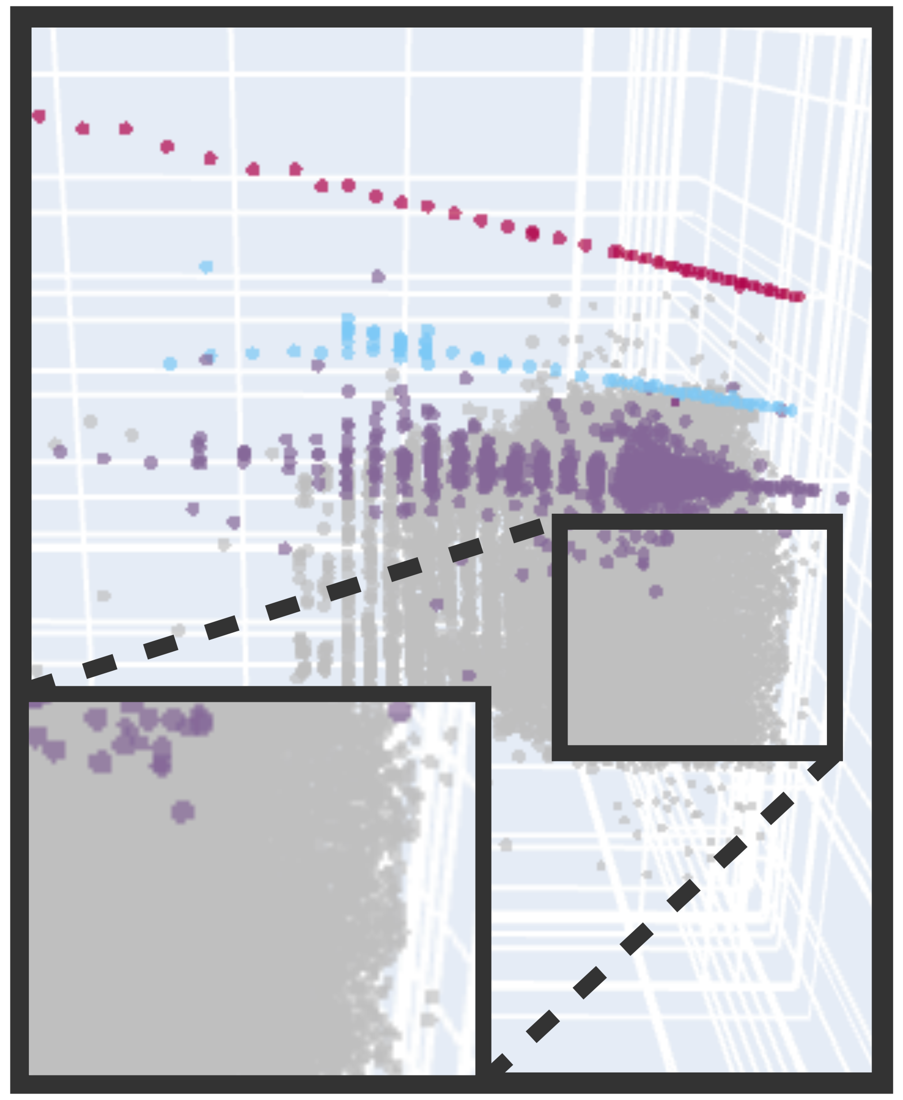
Large Hadron Collider (LHC) at CERN

Philip Harris
(MIT Physics)

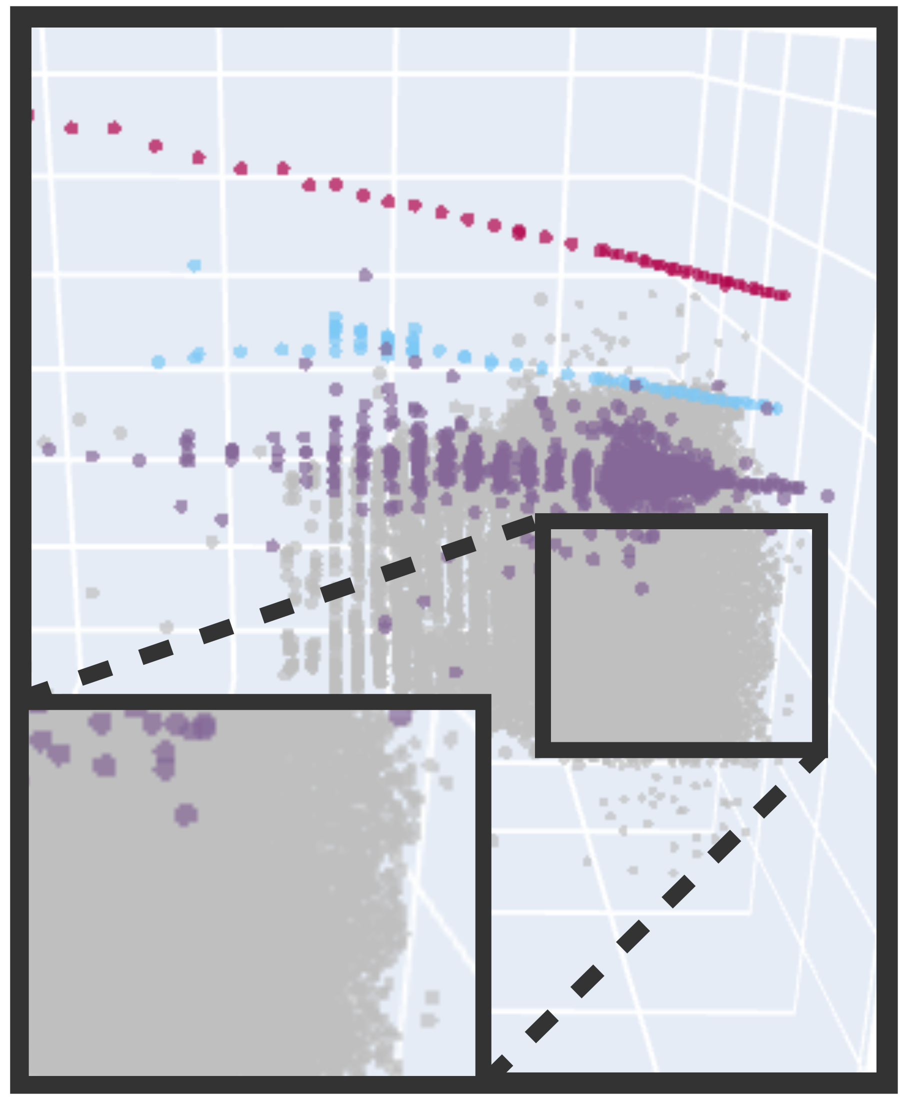
Shih-Chieh Hsu
(UW Physics)

Lindsey Gray
(Fermilab)

GravNet
(CERN)



Calo-SPVCNN
(Ours)



Ground Truth

	SQ	RQ	PQ	Speedup
GravNet	90.0	82.6	75.9	—
Calo-SPVCNN	**92.1**	**85.4**	**79.8**	**11.2×**

Parallelization

DistriFusion

Accelerate High-Resolution Diffusion Model Inference by Leveraging GPU Parallelism

Goal: distributed parallel inference exploiting multiple GPUs to accelerate high-resolution diffusion models.

Naïve Method: distributes the activation across multiple GPUs by splitting images into patches.

Challenge: naive parallelization leads to strong artifacts (duplicated object) due to lack of patch interaction.

Our Solution: DistriFusion communicates the patches, reuses the activations from the *previous* diffusion step to hide networking latency. Insight: adjacent steps' feature maps are similar.

Original	Naïve Patch (4 Devices)	Ours (4 Devices)
MACs: 907T	MACs Per Device: 190T (4.8× Less)	MACs Per Device: **227T** **(4.0× Less)**
Latency: 12.3s	Latency: 3.14s (3.9× Faster)	Latency: **4.16s** **(3.0× Faster)**

Prompt: *Ethereal fantasy concept art of an elf, magnificent, celestial, ethereal, painterly, epic, majestic, magical, fantasy art, cover art, dreamy.*

Artifact (duplication) due to lack of patch interaction

Remove artifact by patch communication; Reduce communication overhead by reusing the old patch

[Li et al., CVPR 2024]

New Architecture, New Primitives

Efficient-ViT

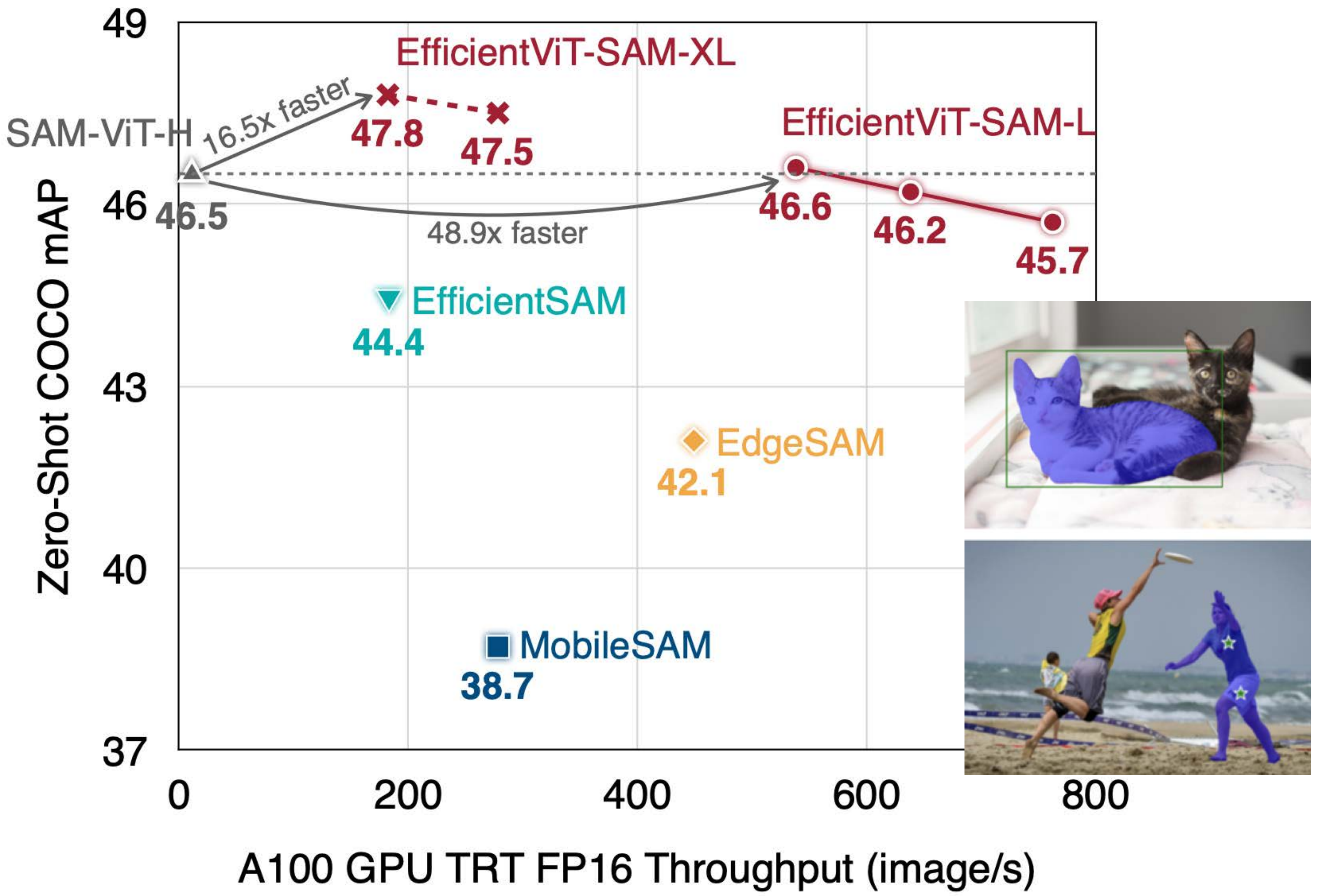
GPU Accelerated Multi-Scale Linear Attention for High-Resolution Dense Prediction

Goal: GPU-friendly *high-resolution* vision transformer architecture for dense prediction (segmentation, SR, SAM, etc)

Challenge: attention FLOPs grow *quadratically* with the #tokens, #tokens grows *quadratically* with the image resolution.

Our Solution: EfficientViT introduces lightweight multi-scale *linear-attention* to replace the heavy softmax attention.

Measured on Nvidia Jetson AGX Orin with TensorRT fp16, bs=1.



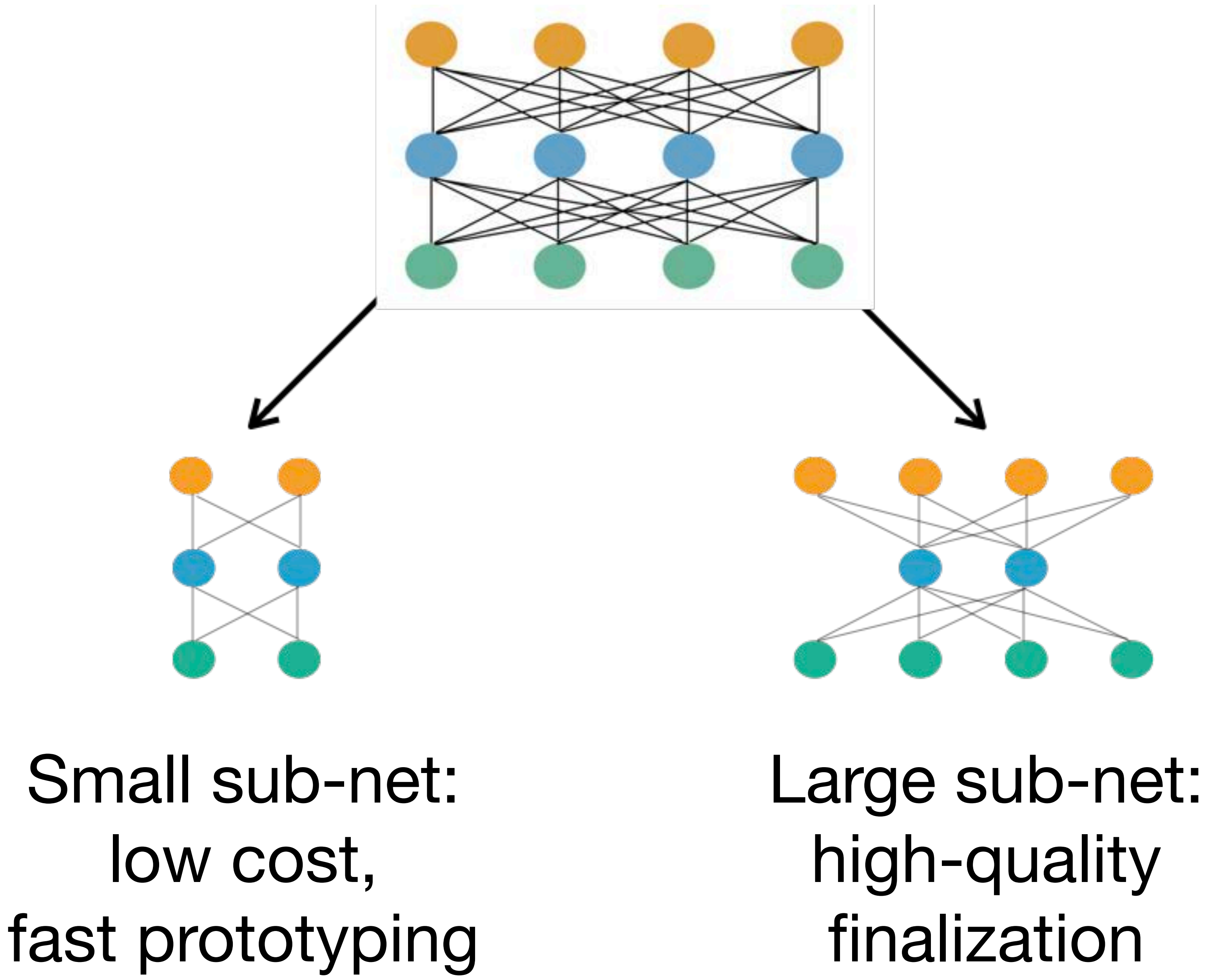
EfficientViT-SAM achieves 48x speedup than SAM-ViT

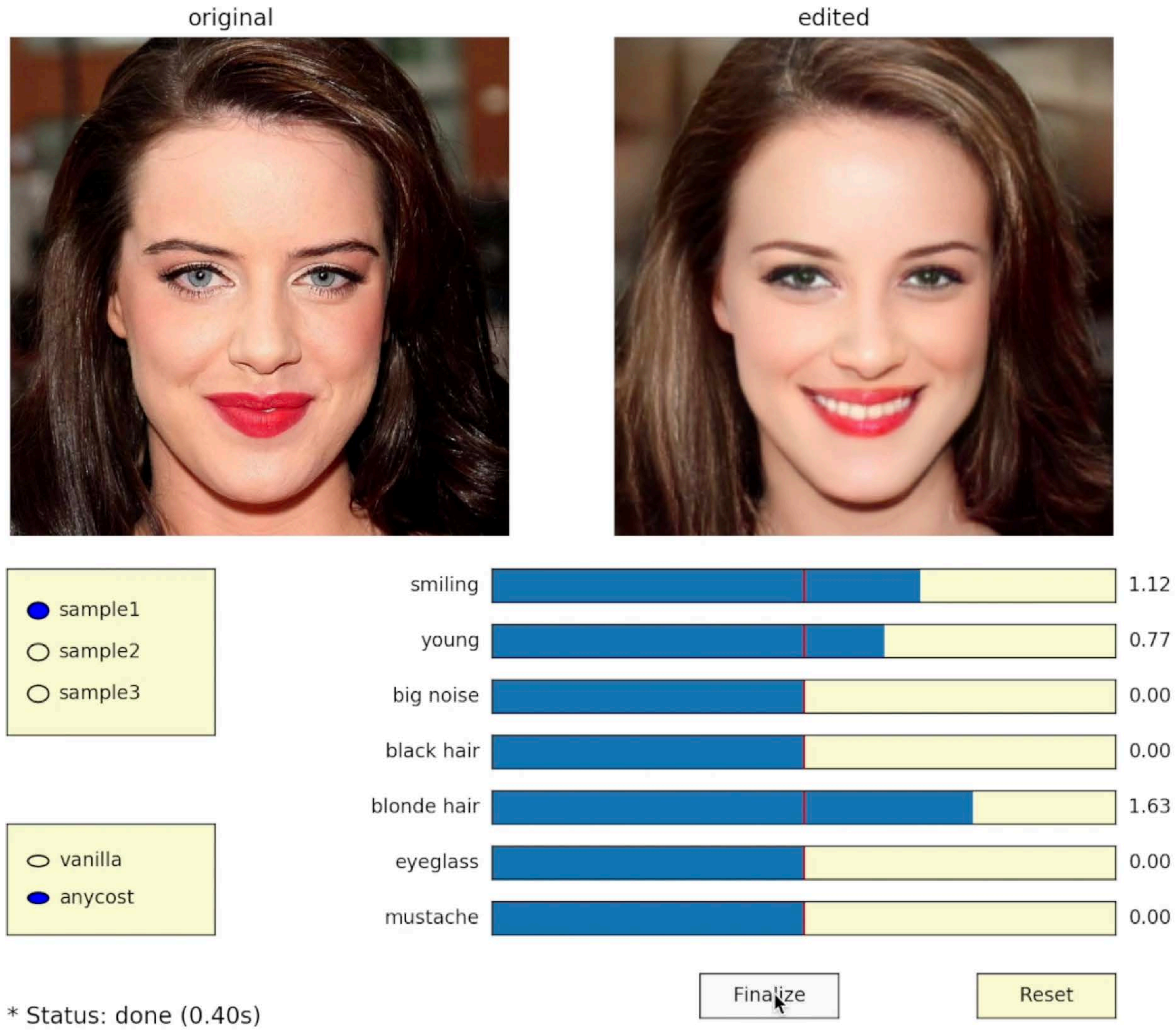
[Cai et al., ICCV 2023][Zhang et al. arXiv]

ANYcost GAN

Generative AI on the Edge

- Generative model is computationally heavy and slow
- Difficult for interactive photo editing on mobile devices
- Anycost GAN with once-for-all (OFA) network, which contains subnetworks that can independently operate.





TinyML

Demo

MCUNetV2: Memory-Efficient Patch-based Inference for Tiny Deep Learning

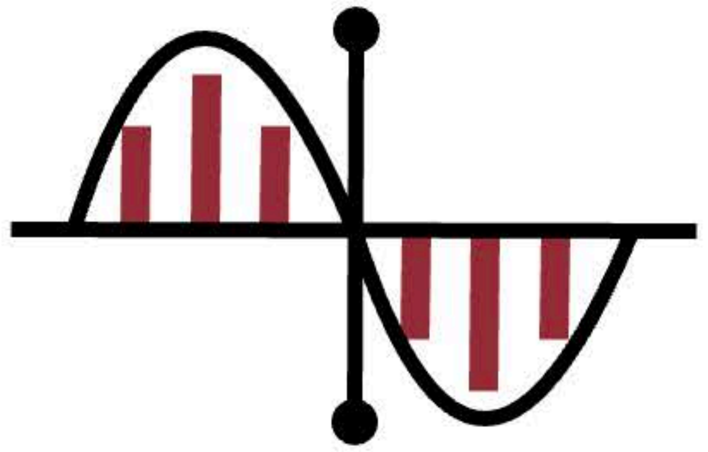
Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, Song Han

mcunet.mit.edu

Learning on the edge

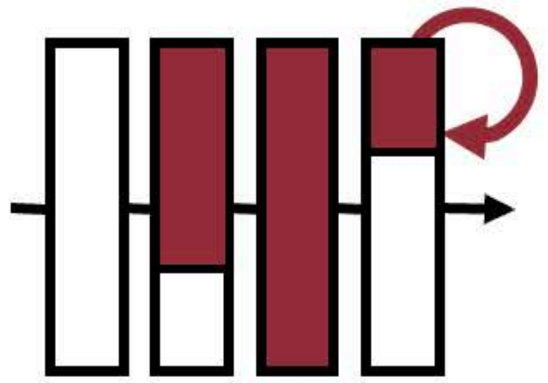
AI systems need to continually adapt to new data collected from the sensors

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

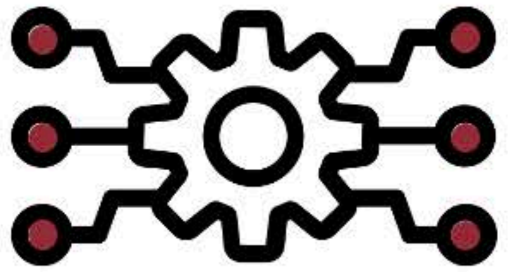


1. Quantization-aware scaling

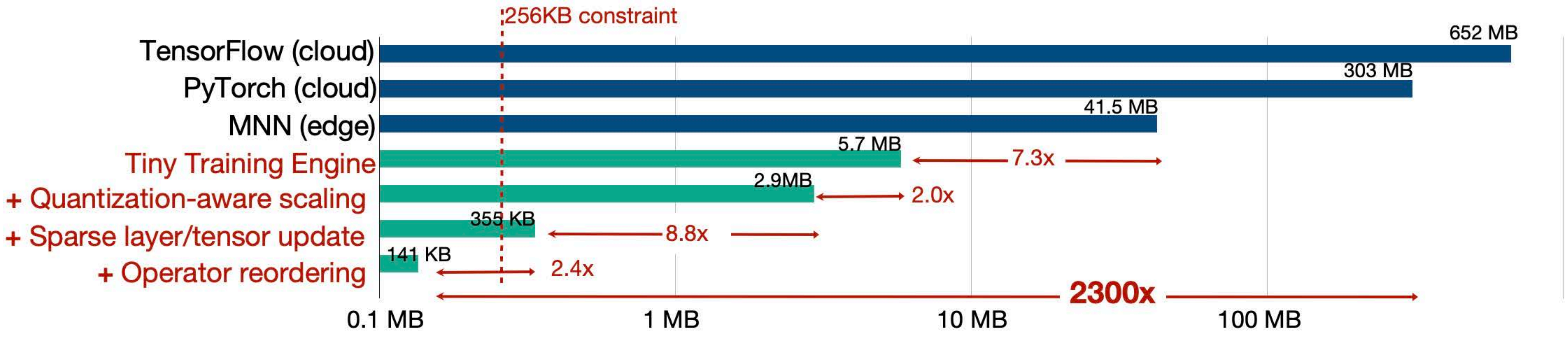
$$\tilde{\mathbf{G}}_{\bar{\mathbf{W}}} = \mathbf{G}_{\bar{\mathbf{W}}} \cdot \boxed{s_{\bar{\mathbf{W}}}^{-2}}$$



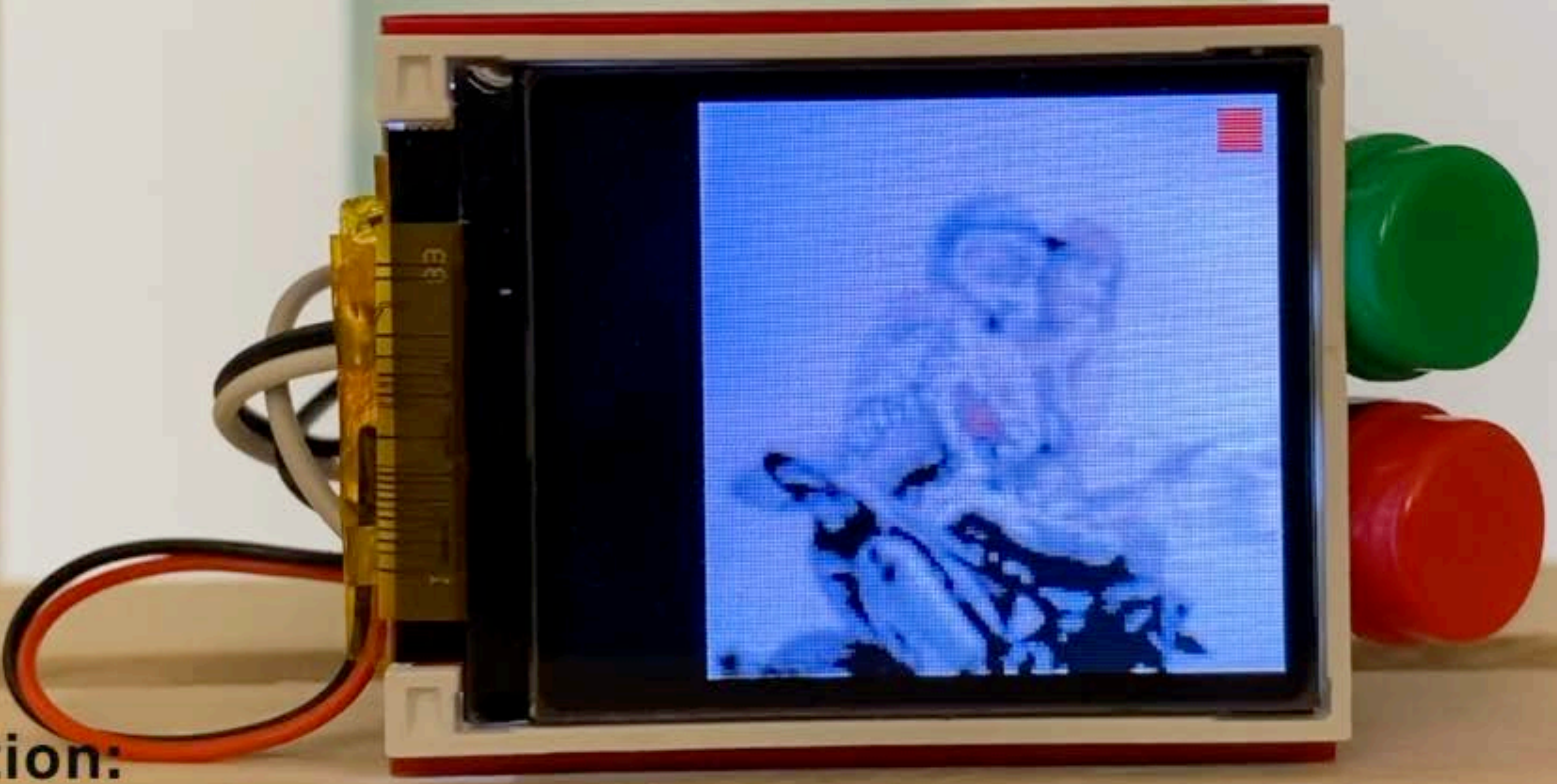
2. Sparse layer/tensor update



3. Tiny Training Engine



3. Post-training testing (high accuracy)



Prediction:
red: person
green: no person

Future work

Research Roadmap

Application

(**Demand** for Computation)

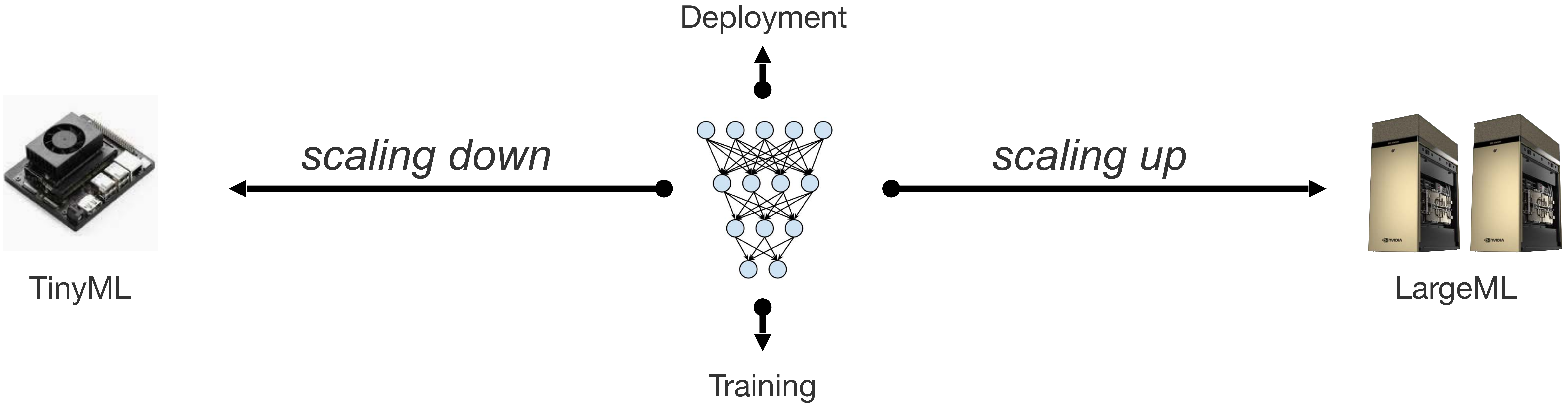
Algorithm **System**

Hardware

(**Supply** of Computation)

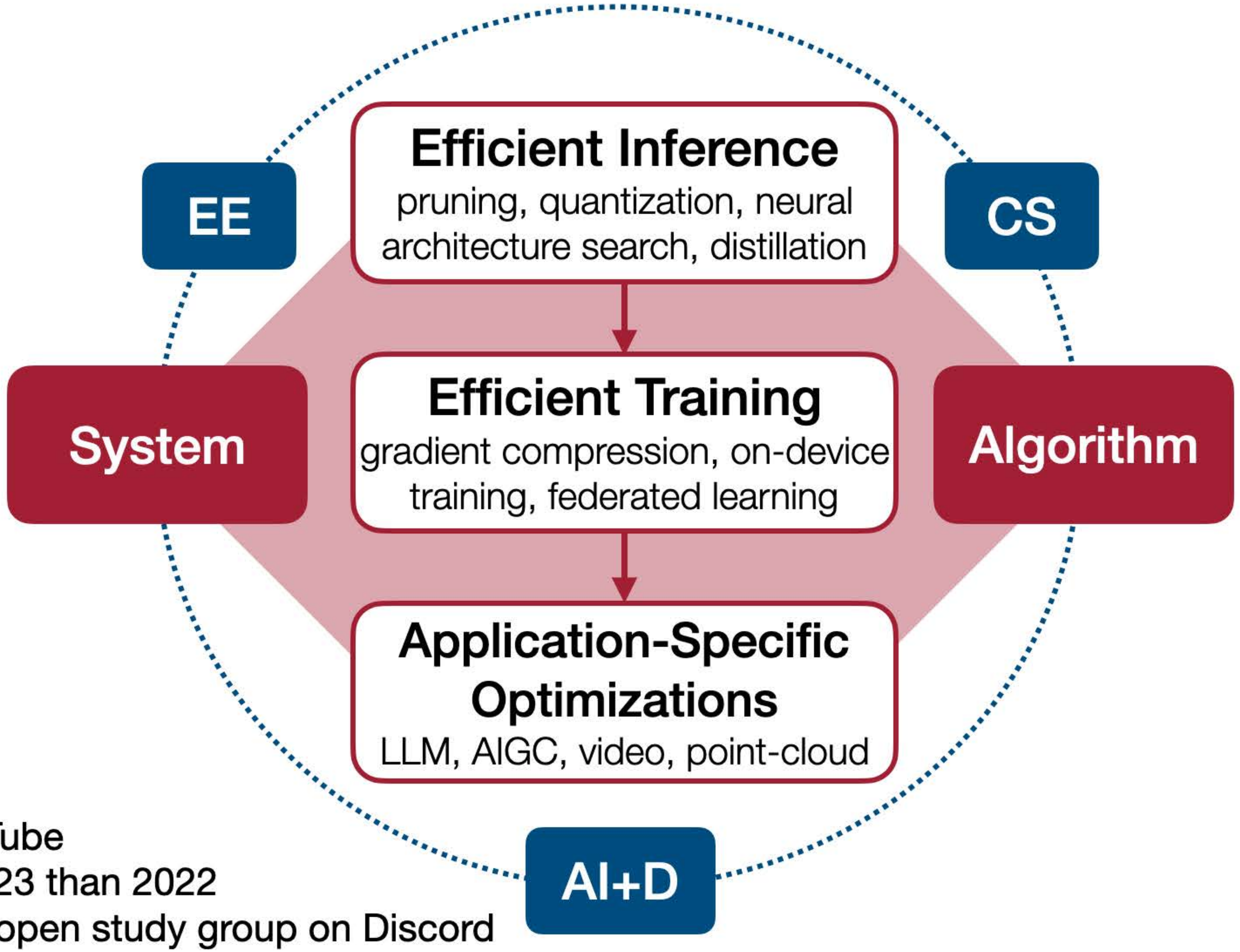
Research Roadmap

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



EfficientML.ai Course

TinyML and Efficient AI Computing



200K views on YouTube
3x registration in 2023 than 2022
700 students in the open study group on Discord

Thank you

MIT HAN LAB
Hardware Accelerated 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, Google, Facebook, SONY, SAMSUNG, HYUNDAI, Ford

Media: MIT Technology Review, IEEE Spectrum, WIRED, engadget, MIT News, VentureBeat