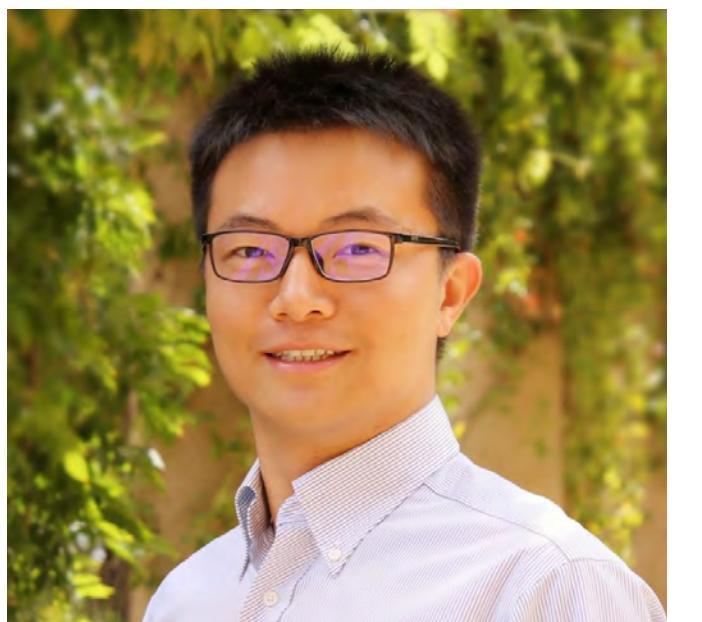


TinyML and Efficient Deep Learning Computing



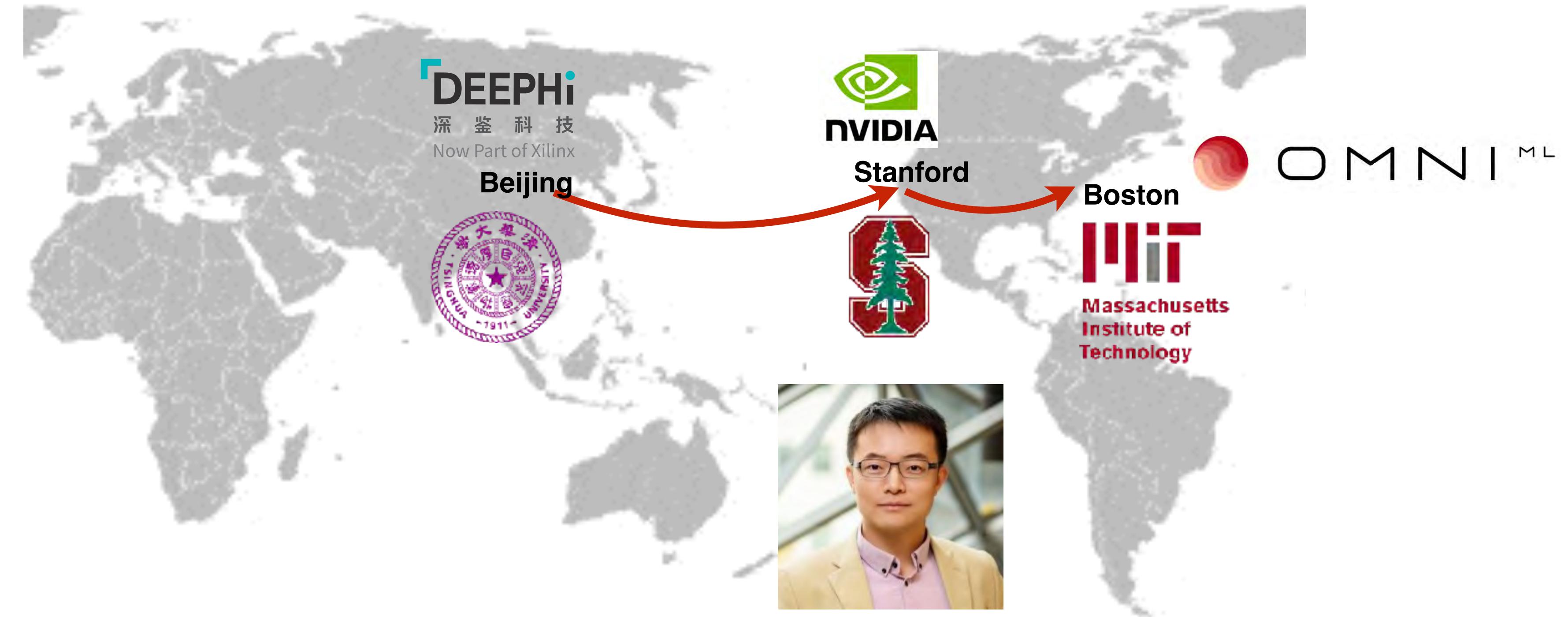
Song Han

Associate Professor, MIT
Distinguished Scientist, NVIDIA
<https://efficientml.ai>

 @SongHan_MIT



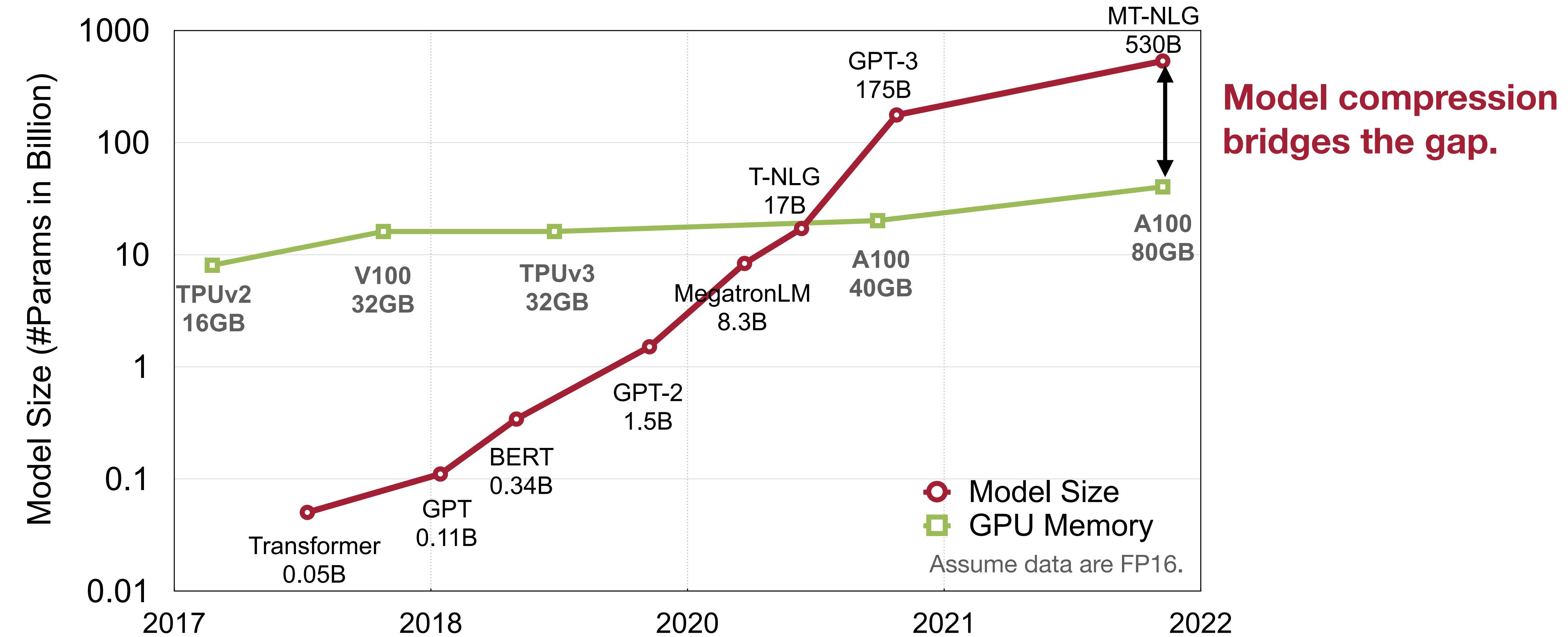
Prof. Song Han



- B.S. from Tsinghua University
- Ph.D. from Stanford University, advised by Prof. Bill Dally
- Deep Compression (best paper award of ICLR)
- EIE (top5 cited paper in 50 years of ISCA)
- Cofounder of DeePhi (now part of AMD)
- Cofounder of OmniML (now part of NVIDIA)
- MIT Technology Review, 35 Innovators under 35
- NSF Career Award
- IEEE “Als 10 to Watch: The Future of AI” Award
- Sloan Research Fellowship

Model Compression and Efficient AI

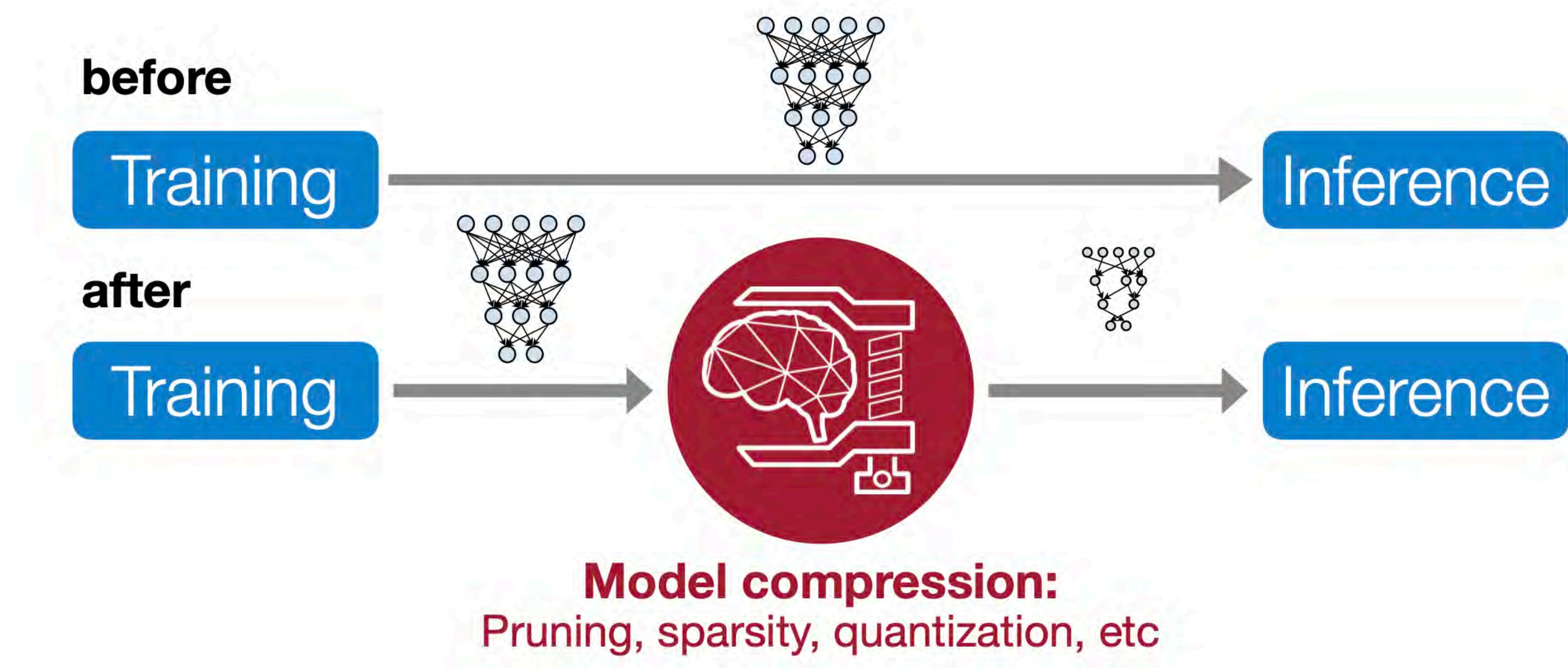
Bridges the Gap between the Supply and Demand of AI Computing



SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (ICML 2023)

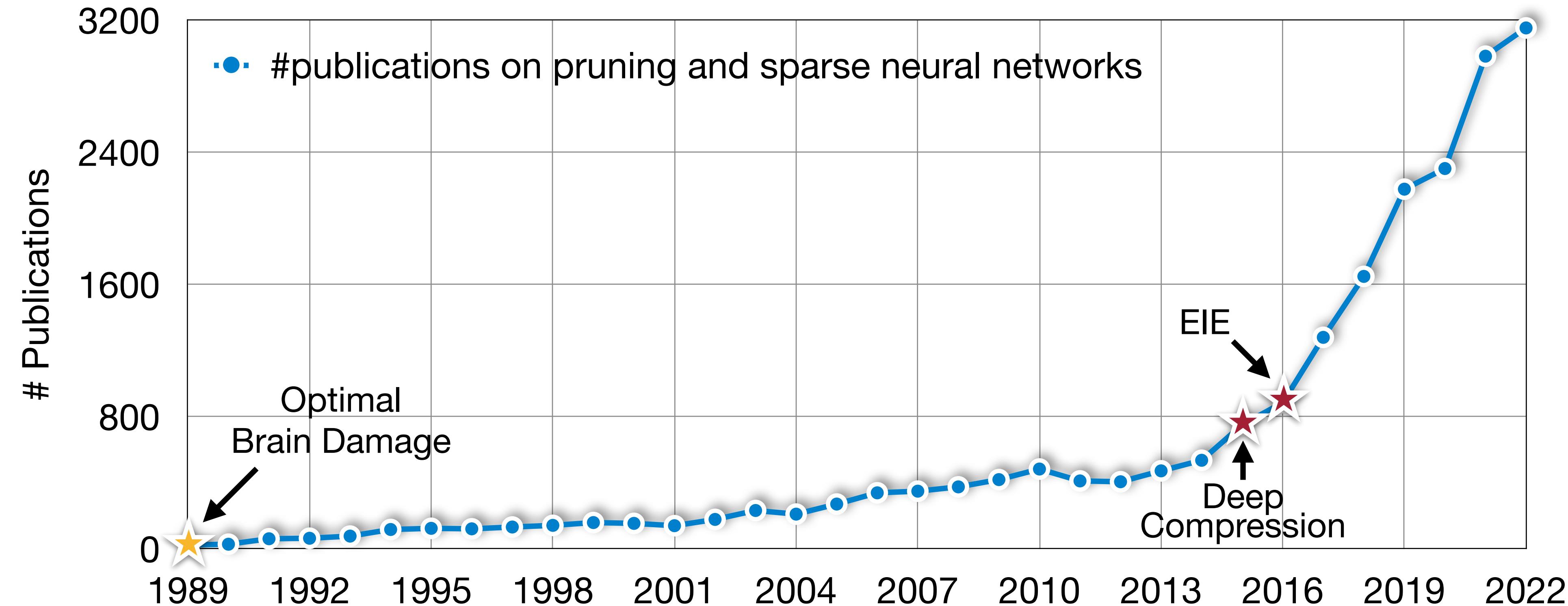
Model Compression and Efficient AI

Bridges the Gap between the Supply and Demand of AI Computing



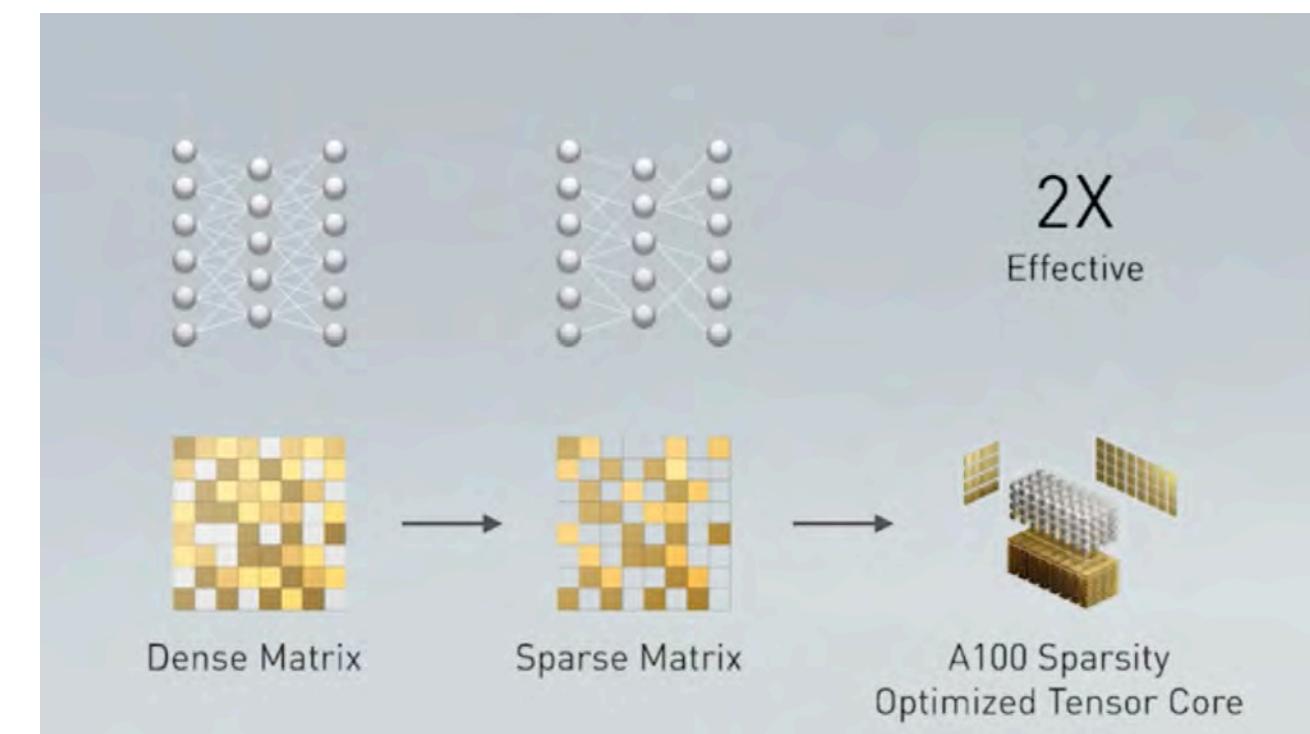
“Deep Compression” and EIE opened a new opportunity to build hardware accelerator for sparse and compressed neural networks

Pruning and Sparsity: Impact



Quickly increased since 2015, including both algorithms and systems.
Influenced the design of NVIDIA Ampere Sparse Tensor Core

Source: <https://github.com/mit-han-lab/pruning-sparsity-publications>



Deep Learning is Everywhere

But they are computationally costly

How to make them light-weighted and fast?



Image source: 1, 2, 3

Deep Learning is Everywhere

But they are computationally costly

How to make them light-weighted and fast?



Image source: [1](#), [2](#), [3](#)

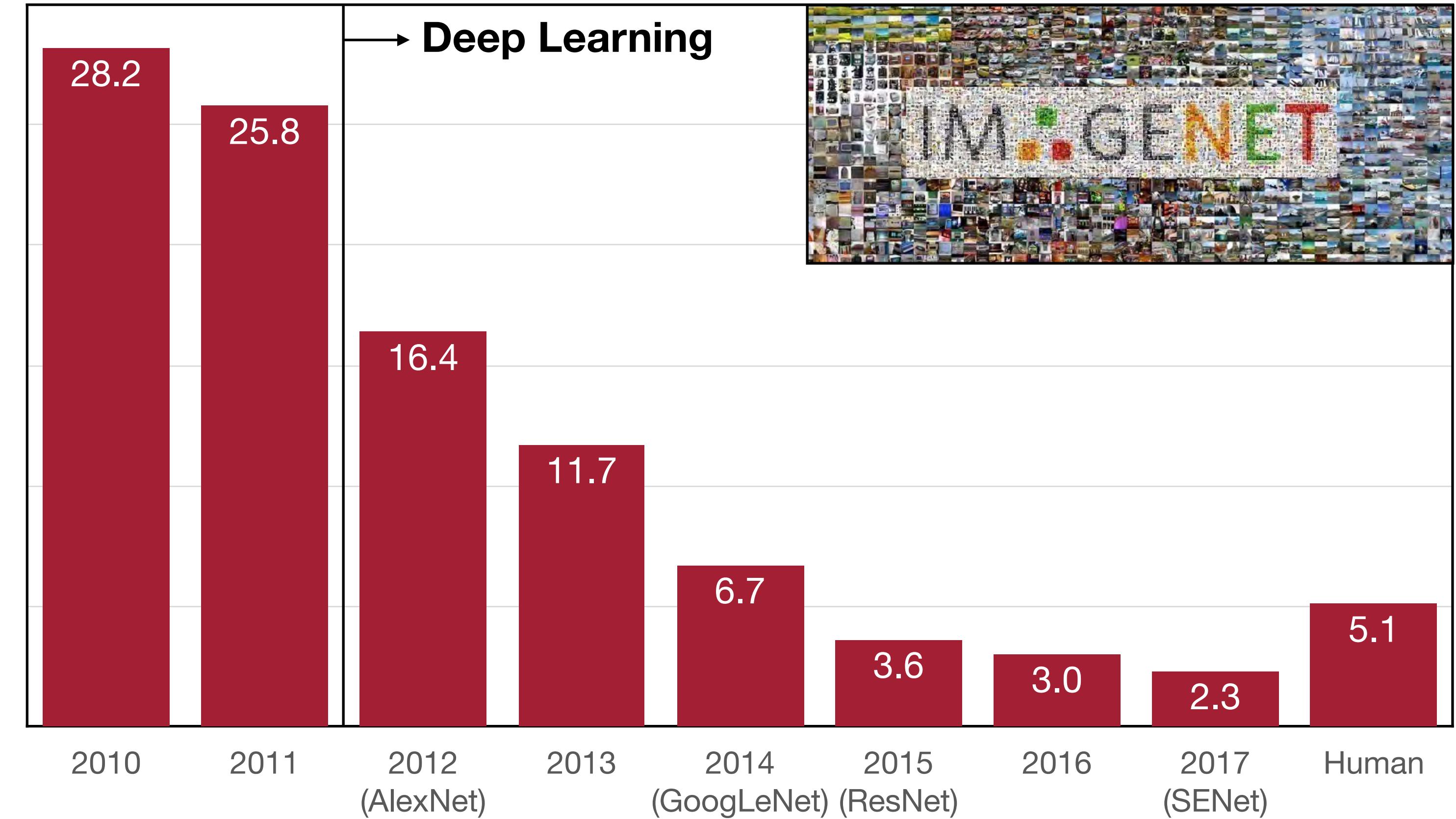
Deep Learning for Image Classification

DNNs achieve super-human classification accuracy on ImageNet



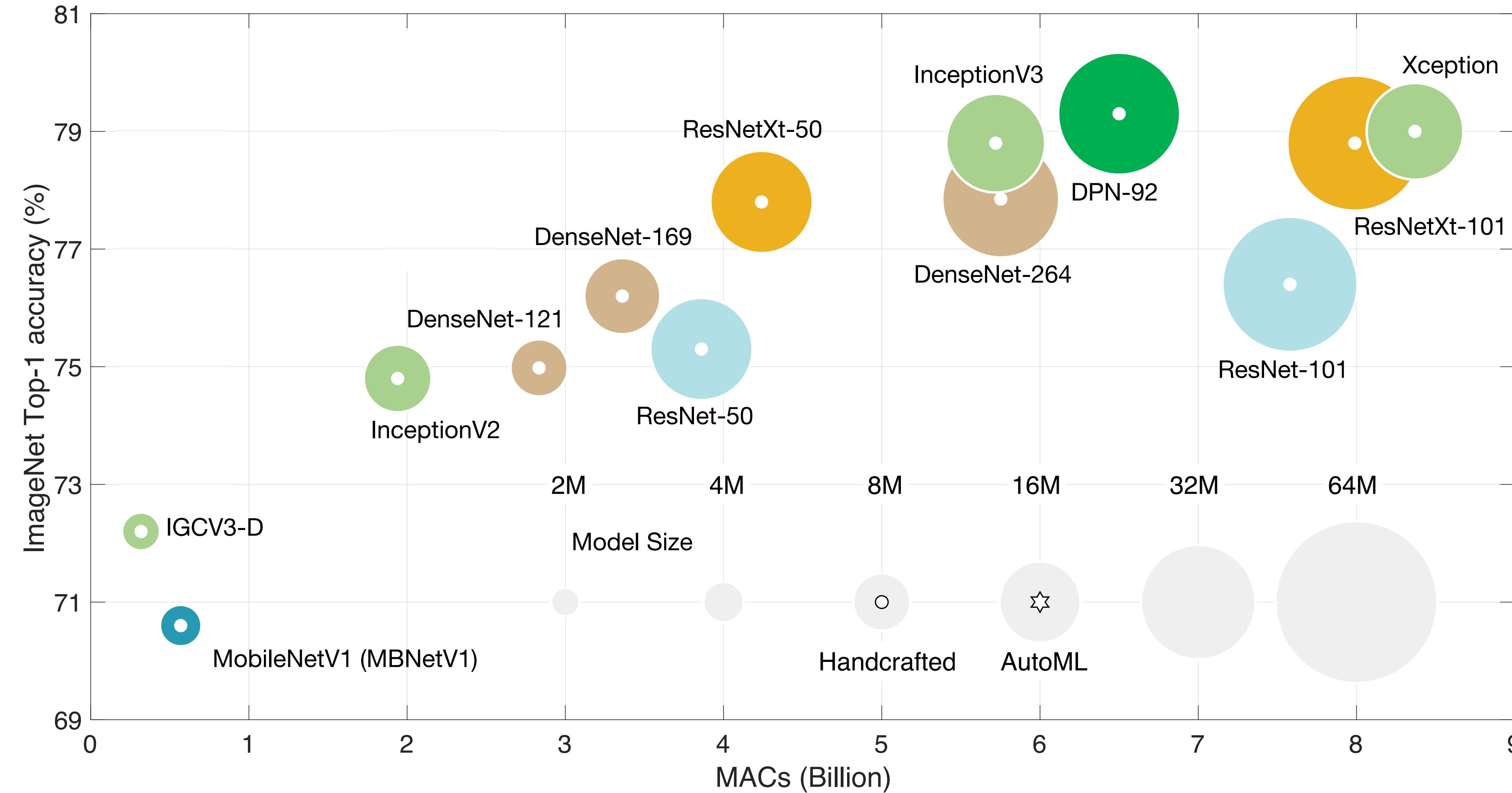
Image Classification

ImageNet Contest Winning Entry: Top 5 Error Rate (%)



Efficient Image Classification

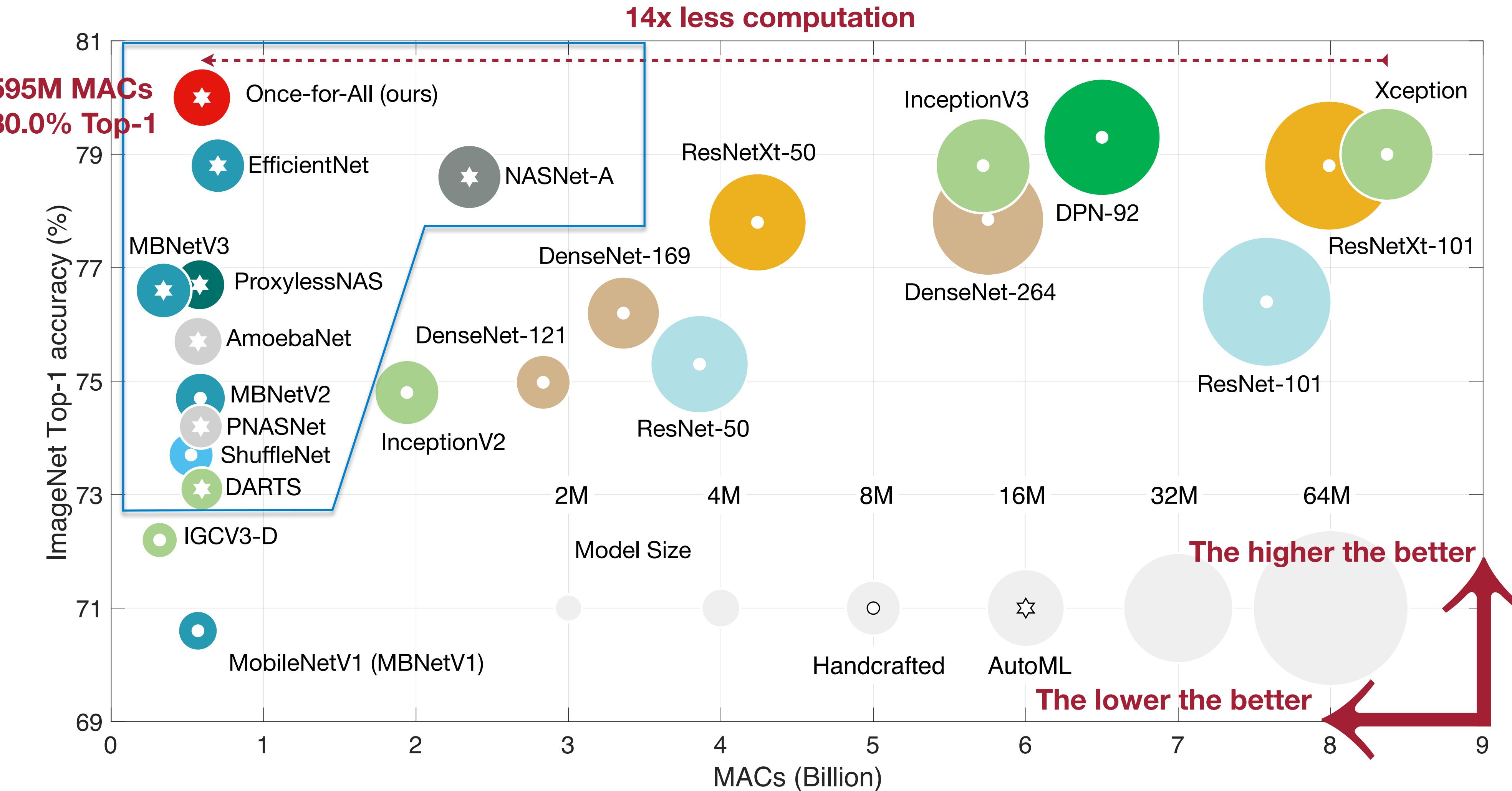
High accuracy comes at the cost of high computation



Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey [Deng et al., IEEE 2020]

Efficient Image Classification

Neural architecture search reduces the computational cost



Once-for-All: Train One Network and Specialize it for Efficient Deployment [Cai et al., ICLR 2020]

Efficient Image Classification

Efficient deep learning enables daily life application on mobile phones

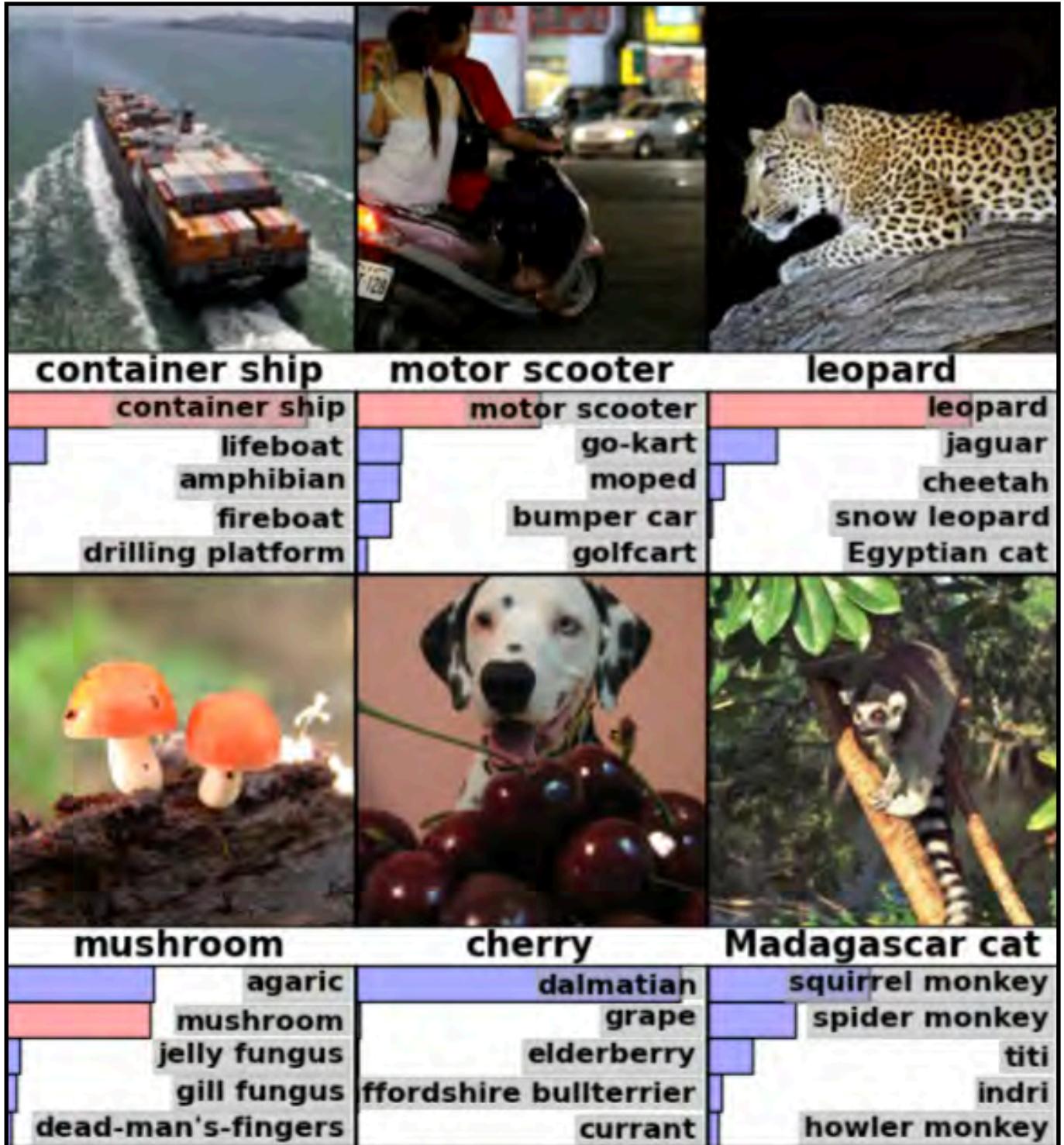


Image Classification

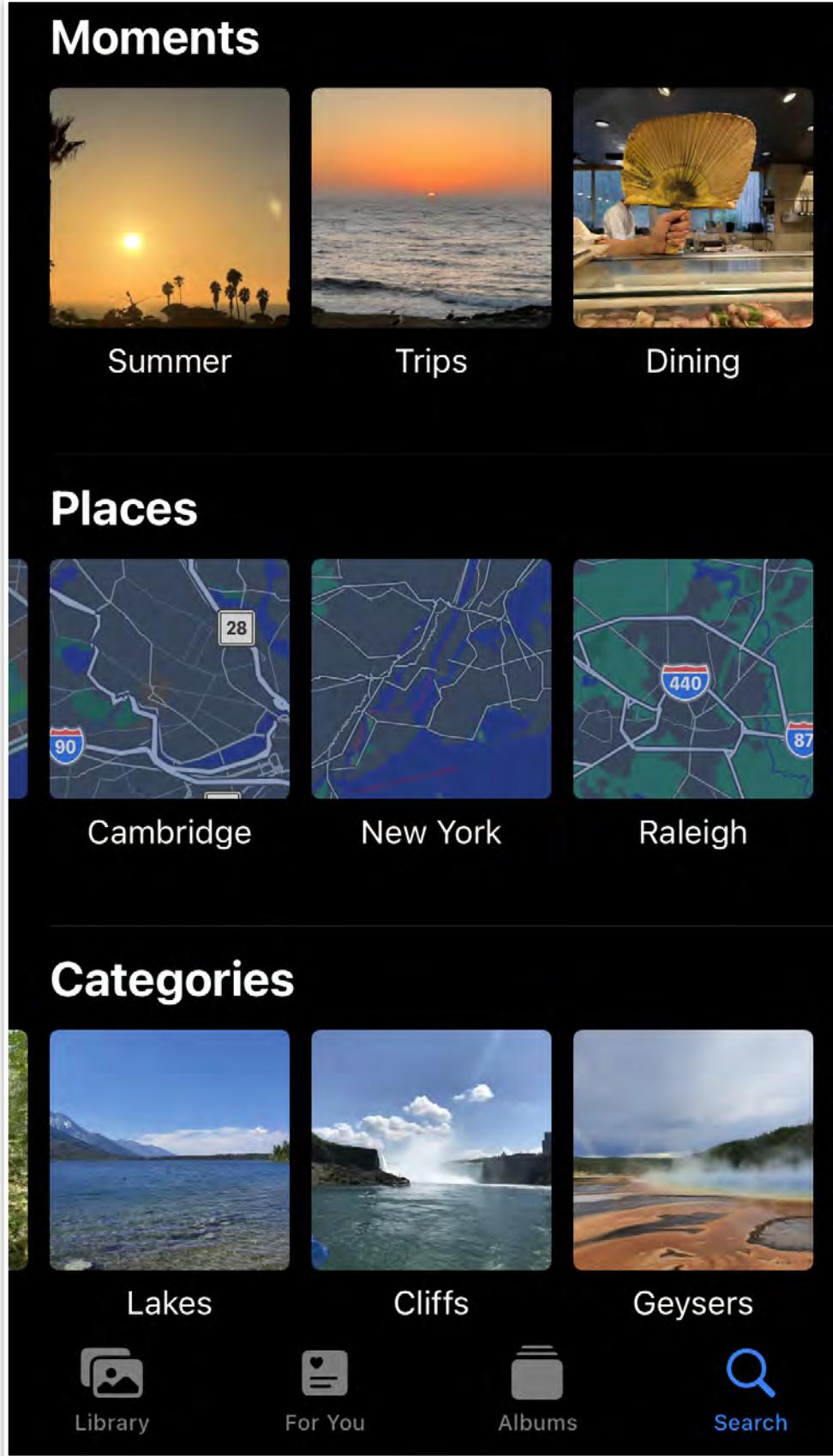
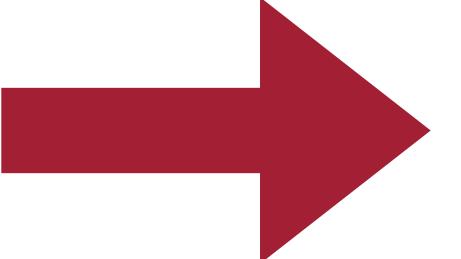
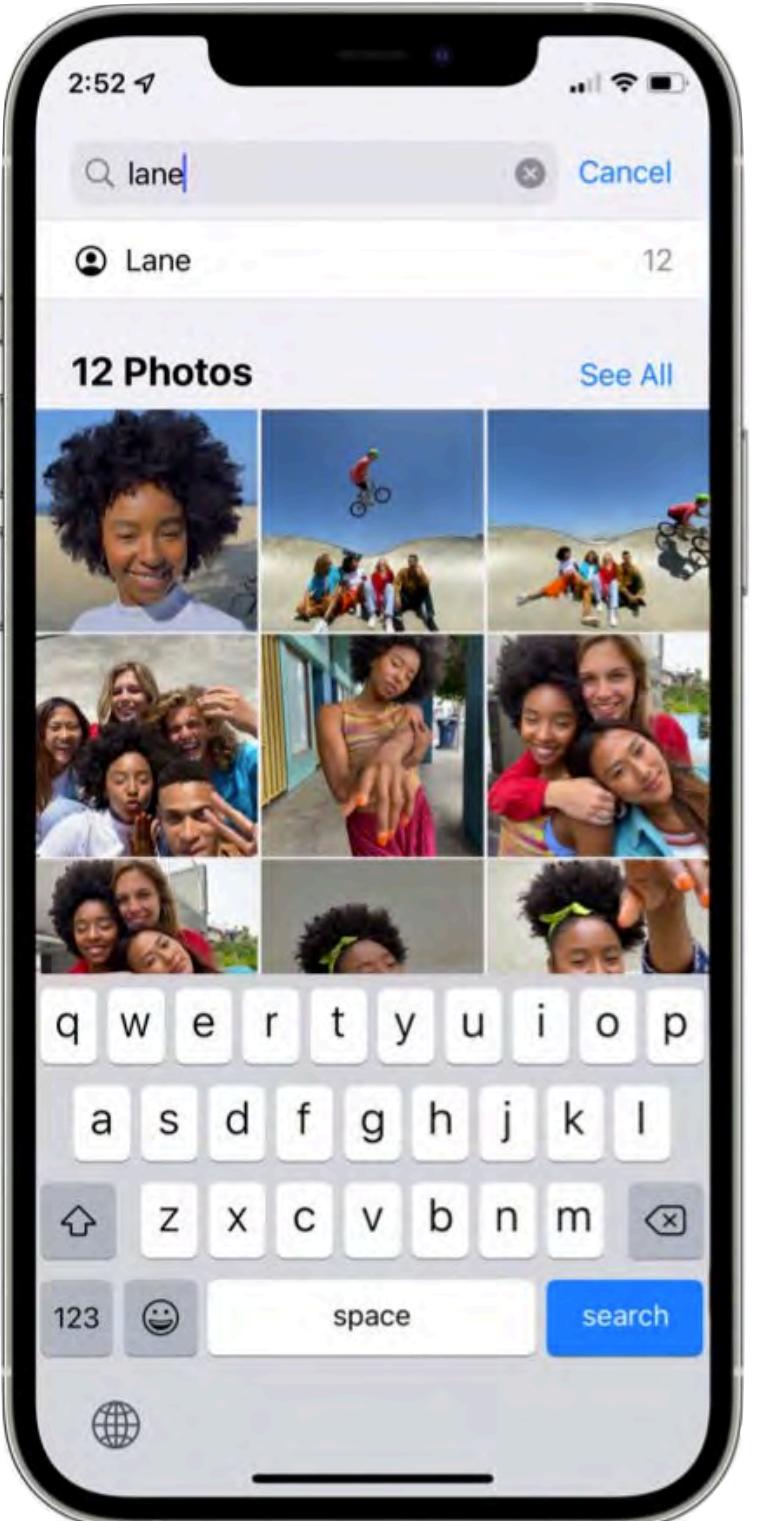
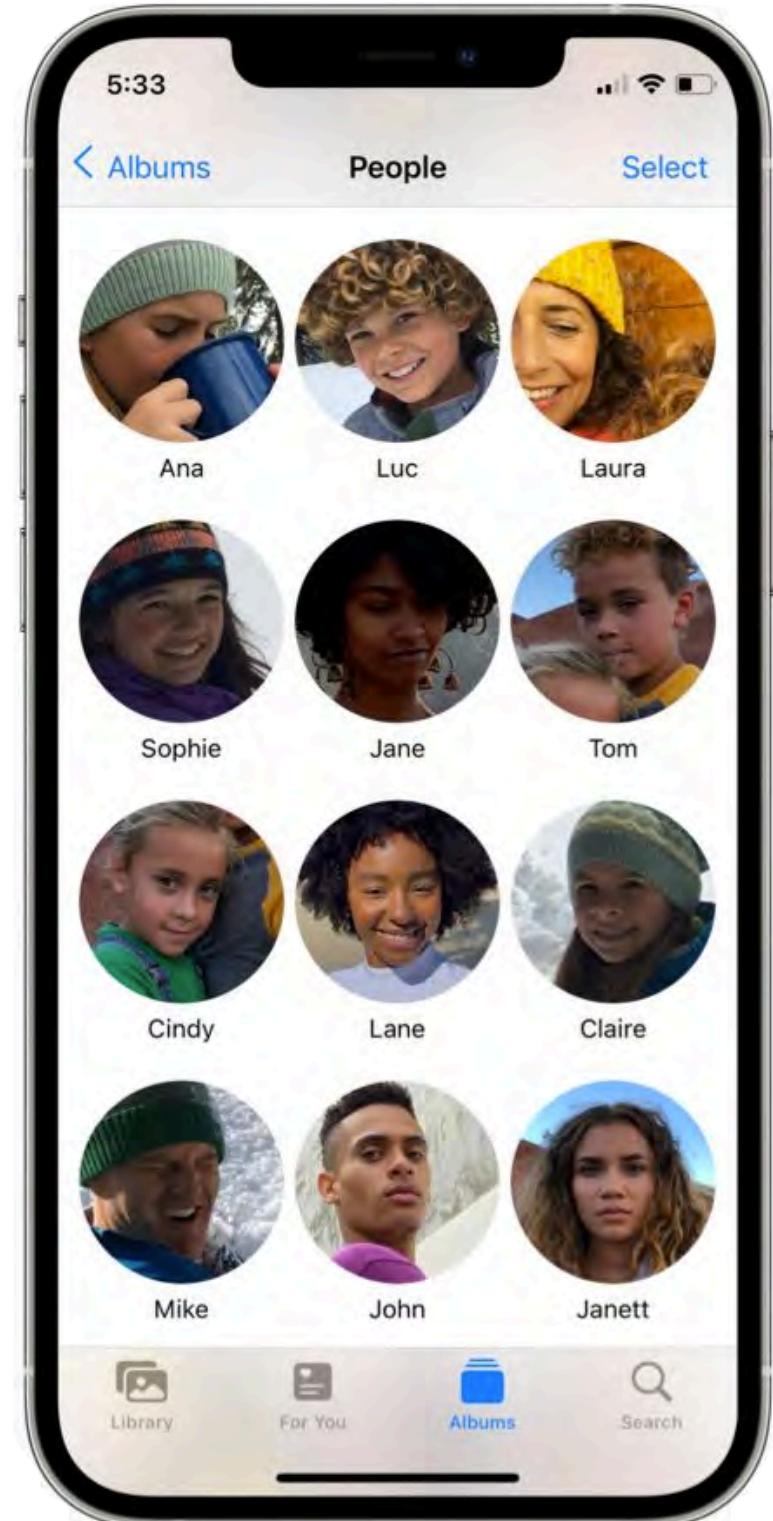
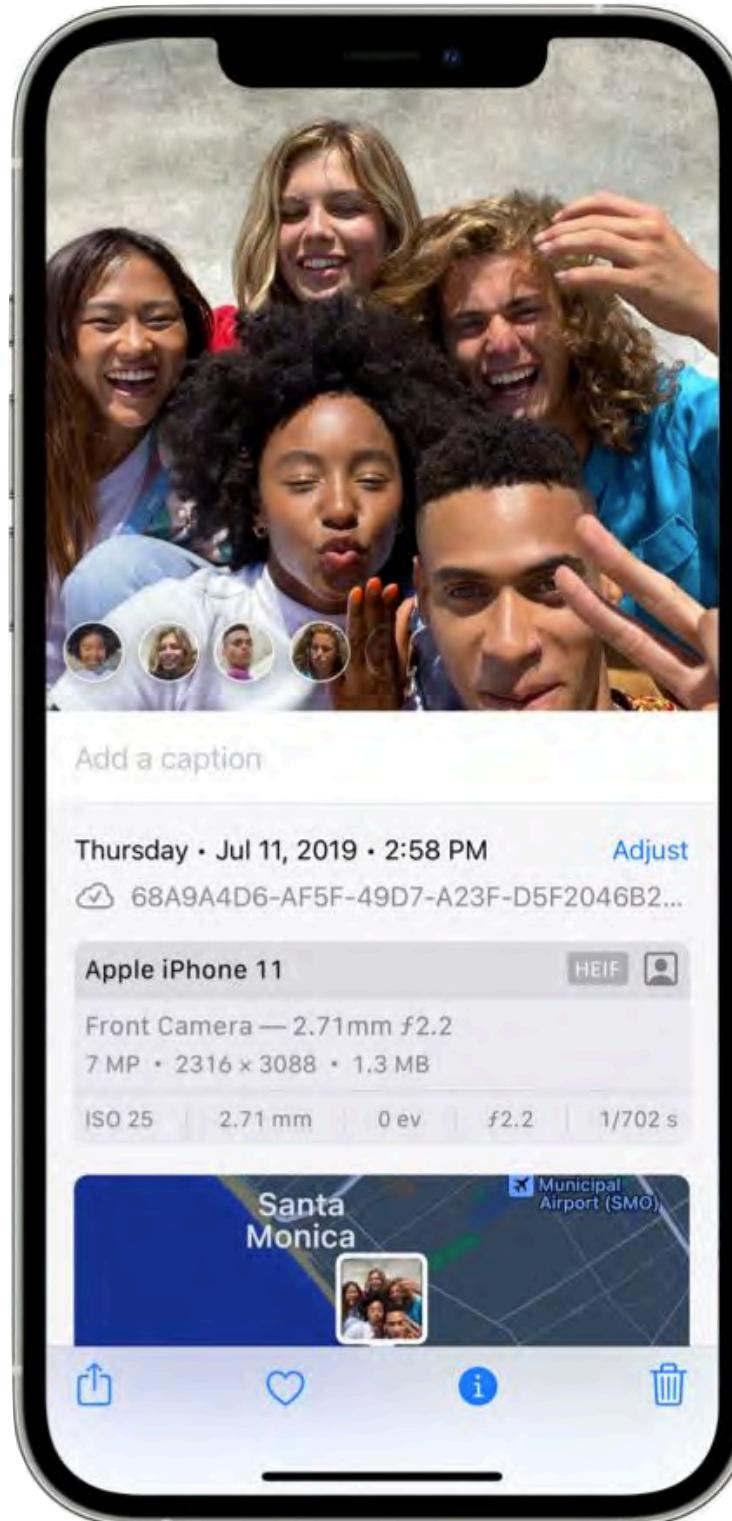


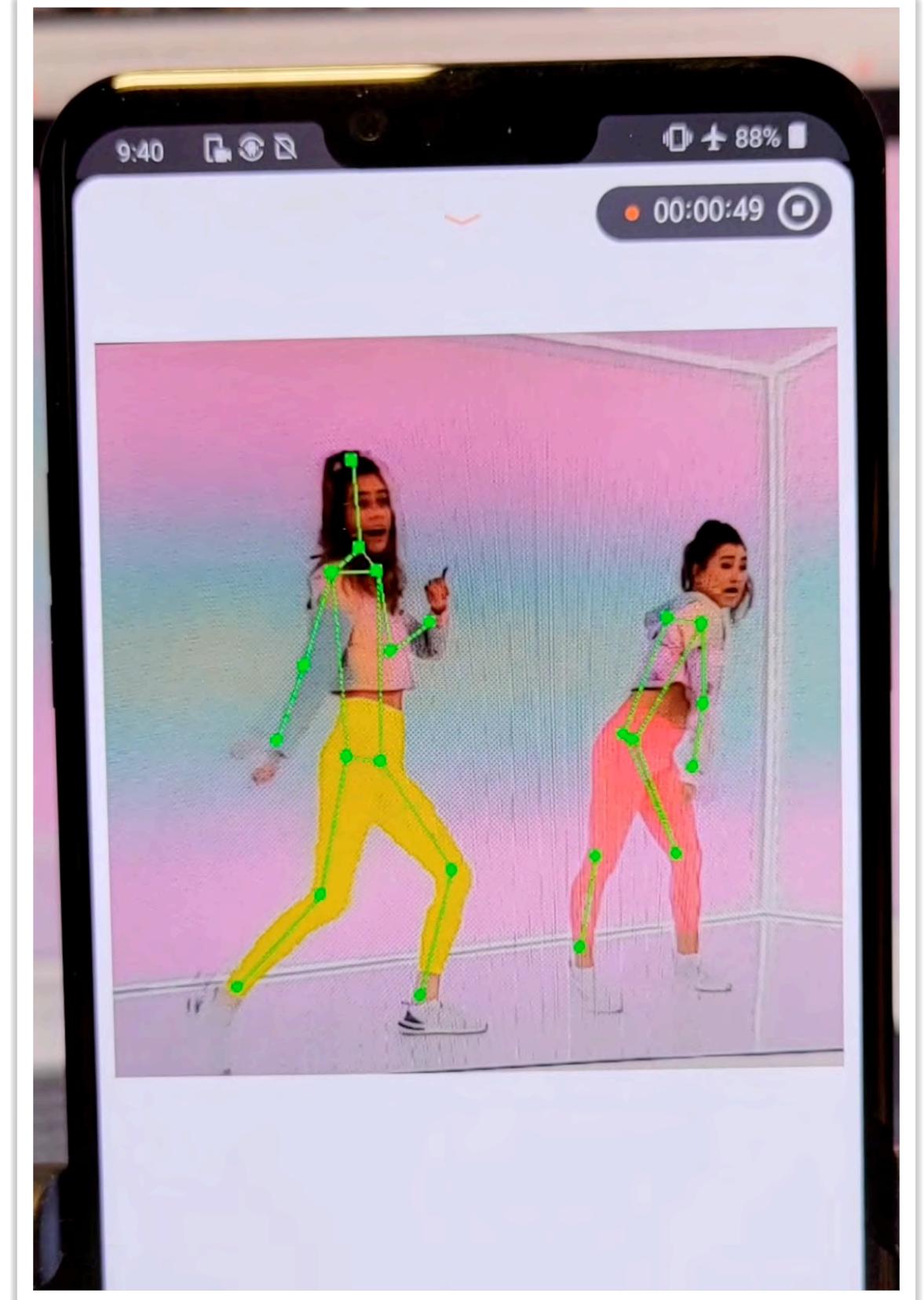
Photo Tags (on iPhone)

Efficient Image Recognition on Phones

Efficient deep learning enables daily life application on mobile phones



People Recognition (on iPhone)

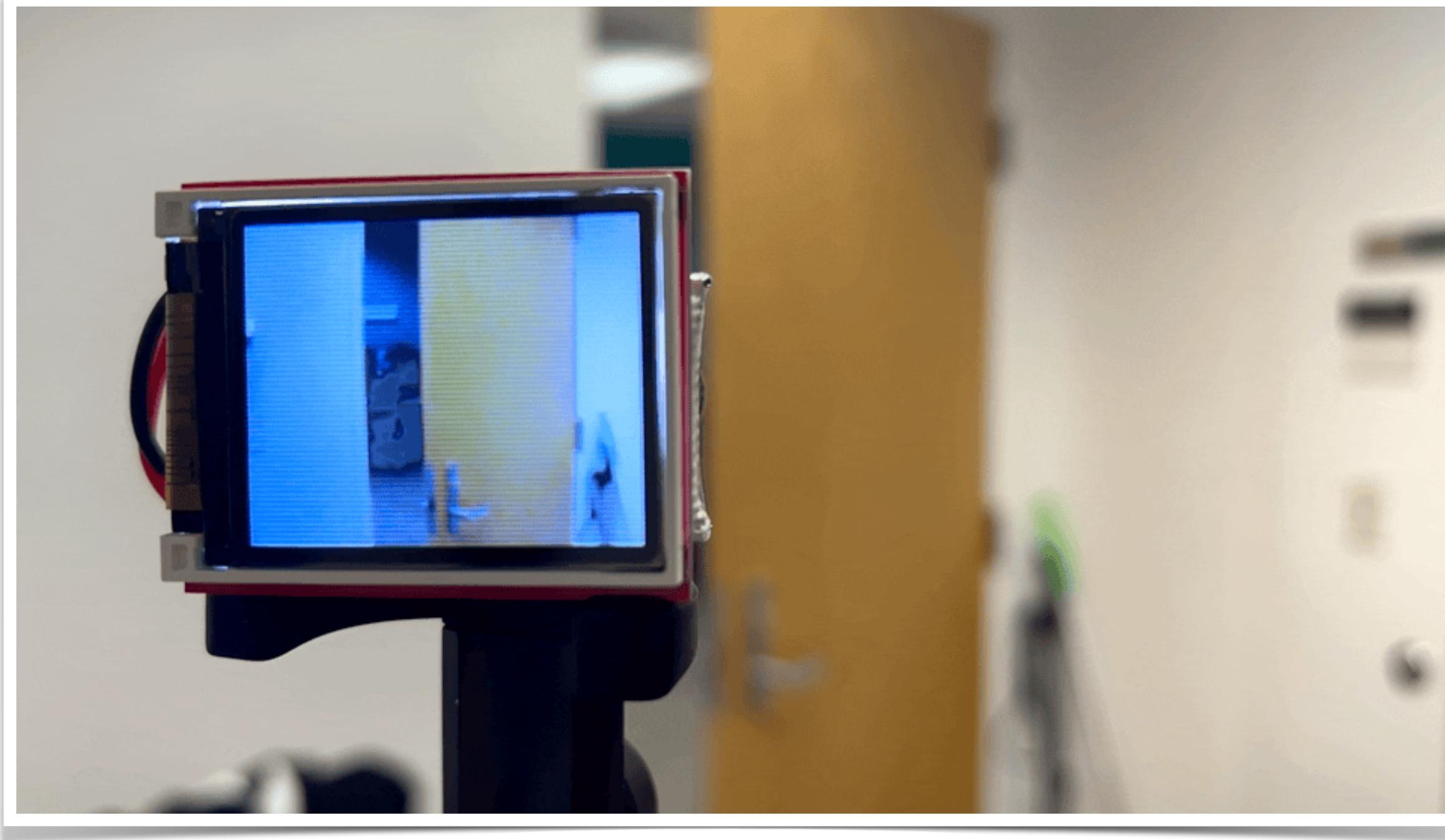
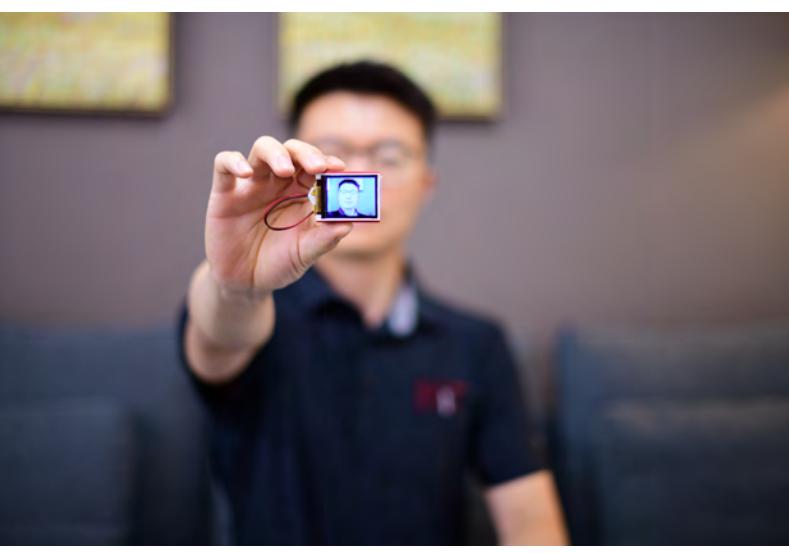


On-Device Pose Estimation

Lite Pose: Efficient Architecture Design for 2D Human Pose Estimation [Wang et al., CVPR 2022]
Recognizing People in Photos Through Private On-Device Machine Learning [Apple, 2021]

Efficient Image Recognition on MCUs

MCUNet enables TinyML on IoT devices
(ARM Cortex-M7 + OpenMV Cam)



Facial Mask Detection



Person Detection

Efficient On-Device Training on Edge

AI systems need to continually adapt to new data collected locally



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

On-Device Training Under 256KB Memory

Demo Video

ImageNet Pre-trained MCUNet -> VWW

Running on STM32F746 MCU (320KB SRAM, 1MB FLASH)



Prompt Segmentation

Segment Anything Model (SAM) realizes promptable zero-shot segmentation



Segment based on interactive point prompts



Automatically segment everything in an image

SAM runs at 12 images per second due to the large vision transformer model (ViT-Huge).

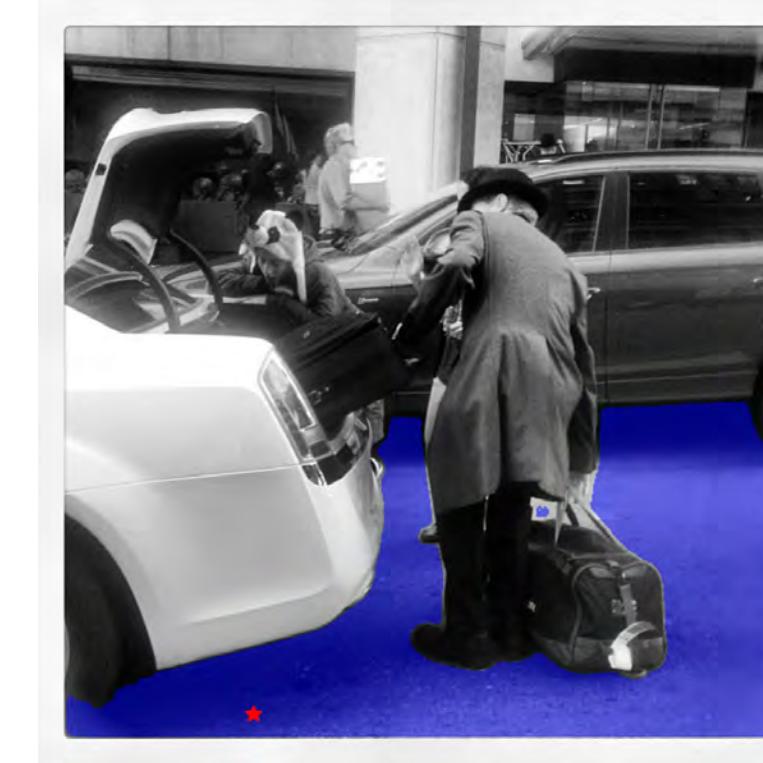
Segment Anything: <https://segment-anything.com/>

Efficient Prompt Segmentation

EfficientViT accelerates Segment Anything Model by 70 times

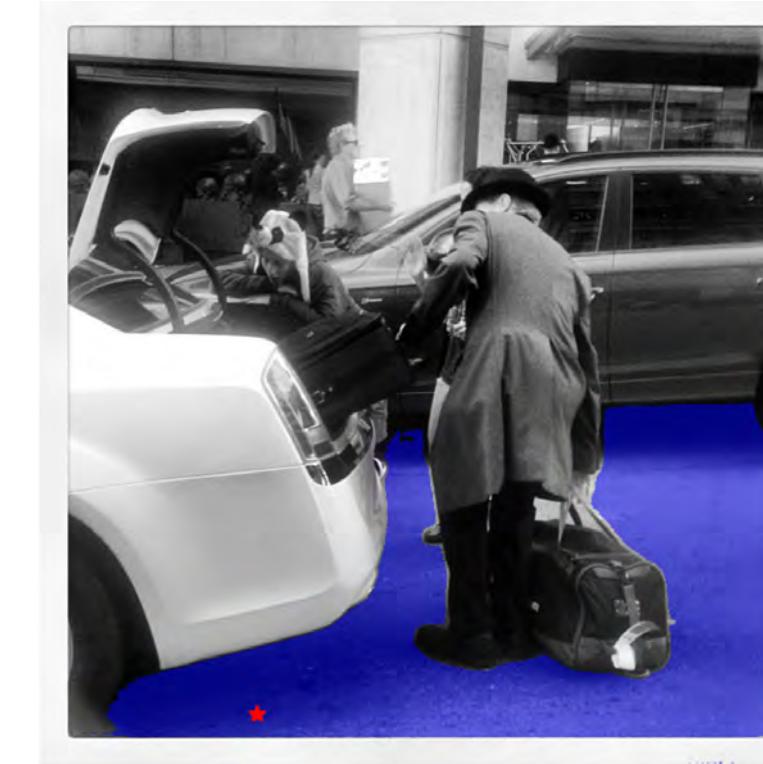
SAM ViT-Huge

12 image/s



EfficientViT

842 image/s



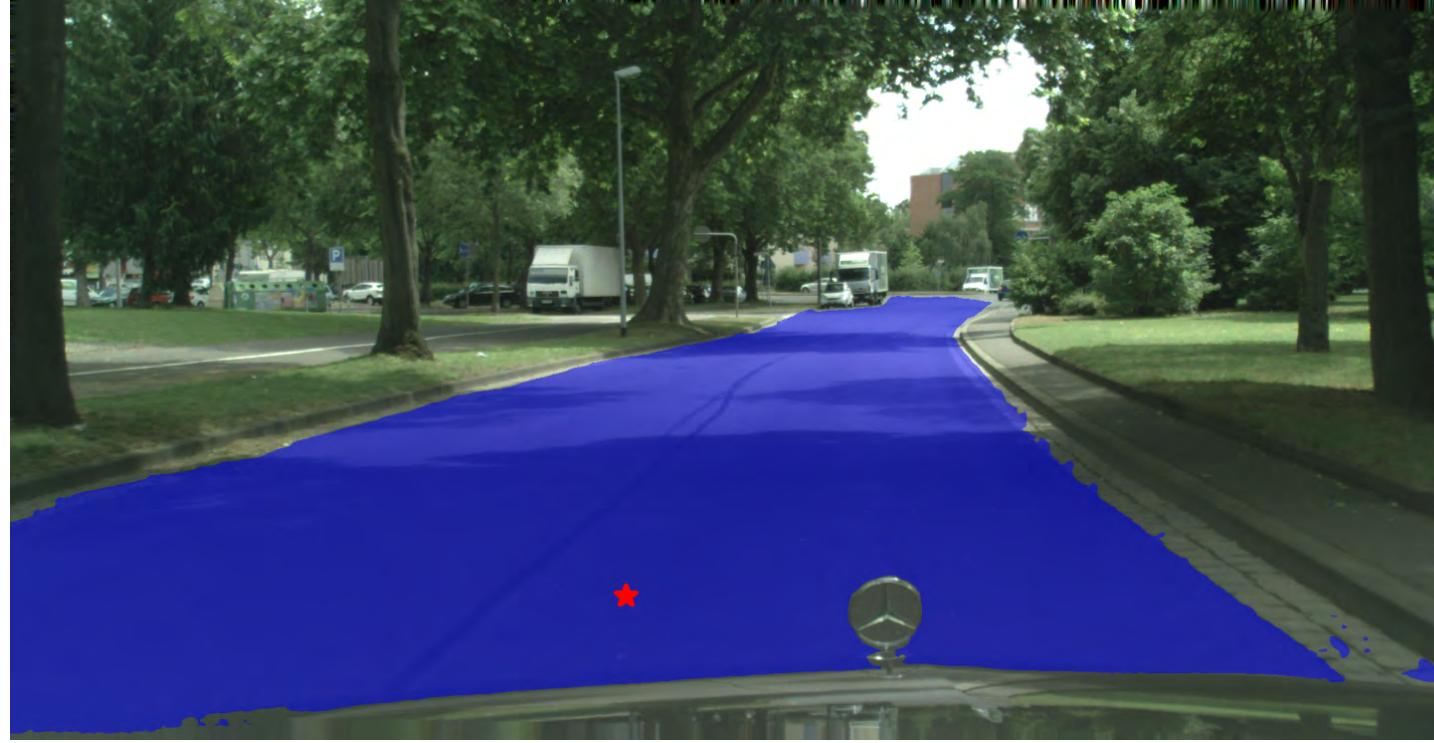
EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction [Cai et al., ICCV 2023]

Efficient Prompt Segmentation

EfficientViT accelerates Segment Anything Model by 70 times

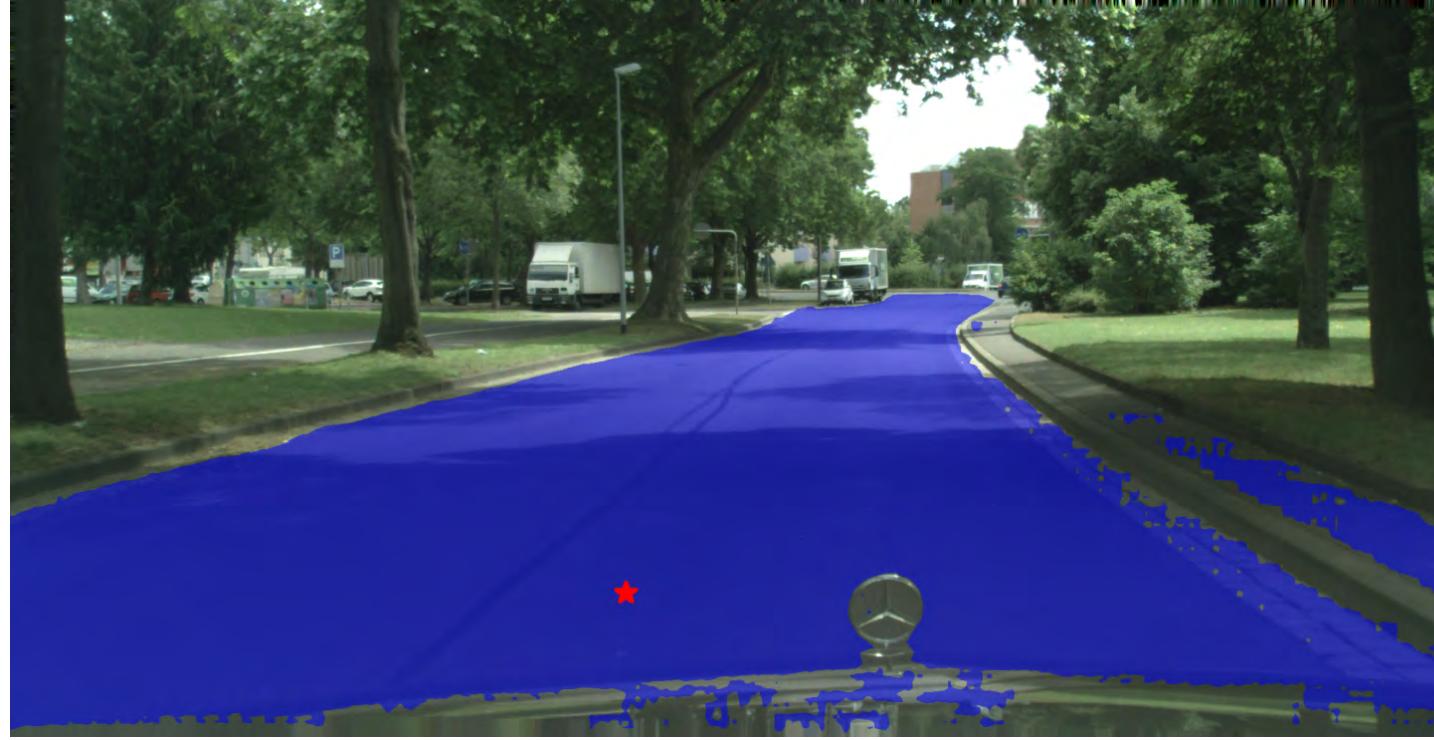
EfficientViT

842 image/s



SAM ViT-Huge

12 image/s



EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction [Cai et al., ICCV 2023]

From Discriminative Model to Generative Model

Diffusion models create realistic images from a natural language description

Teddy bears, mixing sparkling chemicals as mad scientists



A bowl of soup, as a planet in the universe, as a 1960s poster



A photo of an astronaut riding a horse on mars

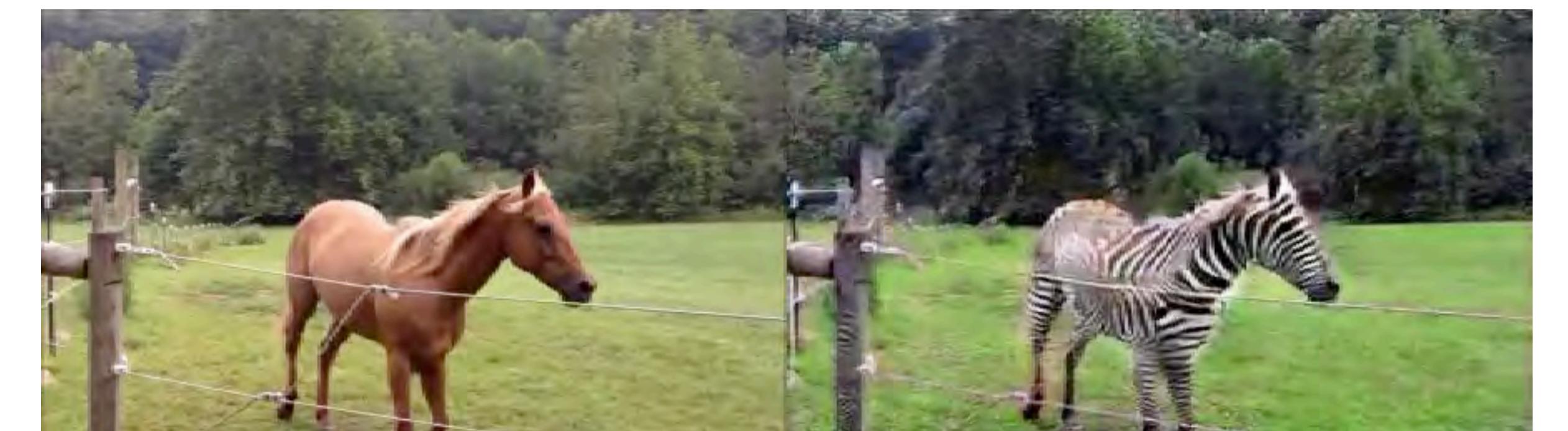
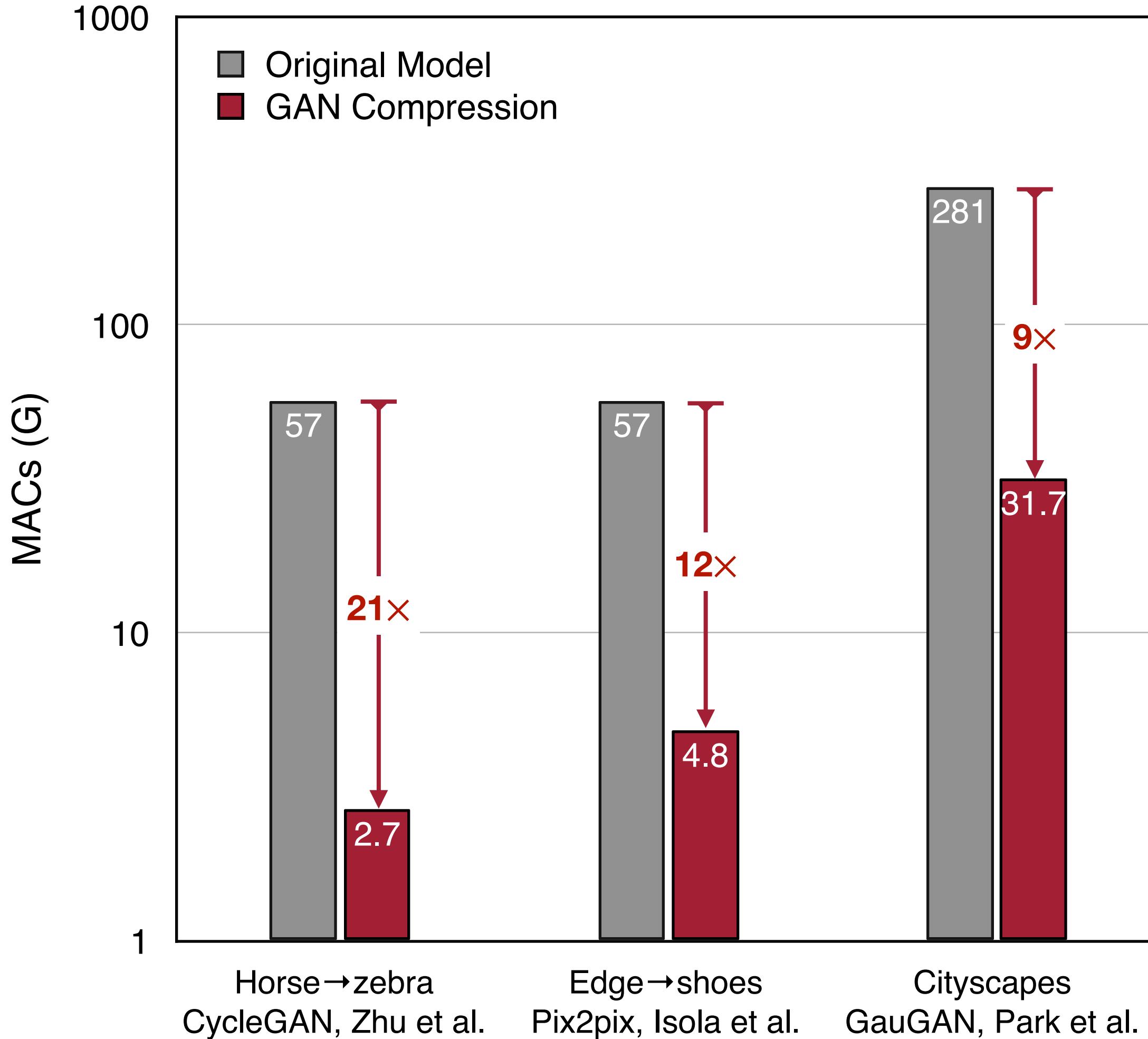


Training Stable Diffusion costs \$600,000 (256 A100s, 150k hours)

Midjourney: <https://www.midjourney.com/>

Efficient Image Generation

GAN Compression reduces the computation by 9-21X by pruning



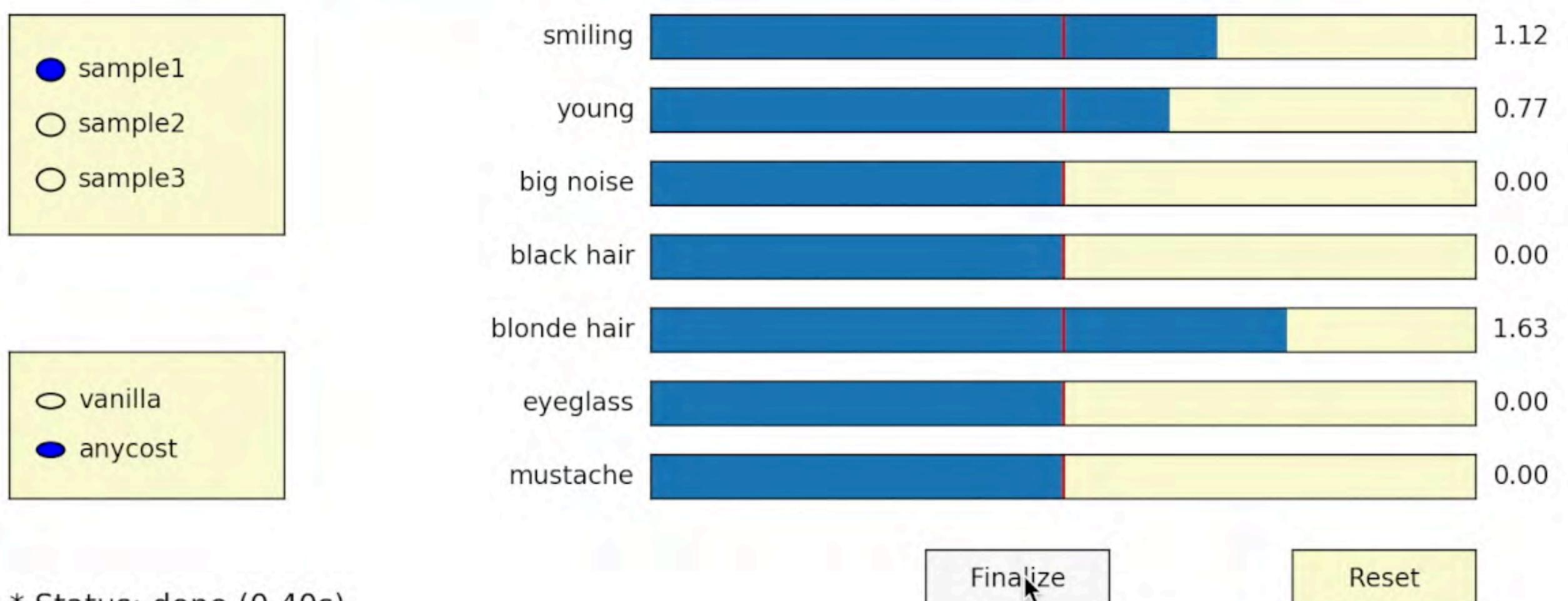
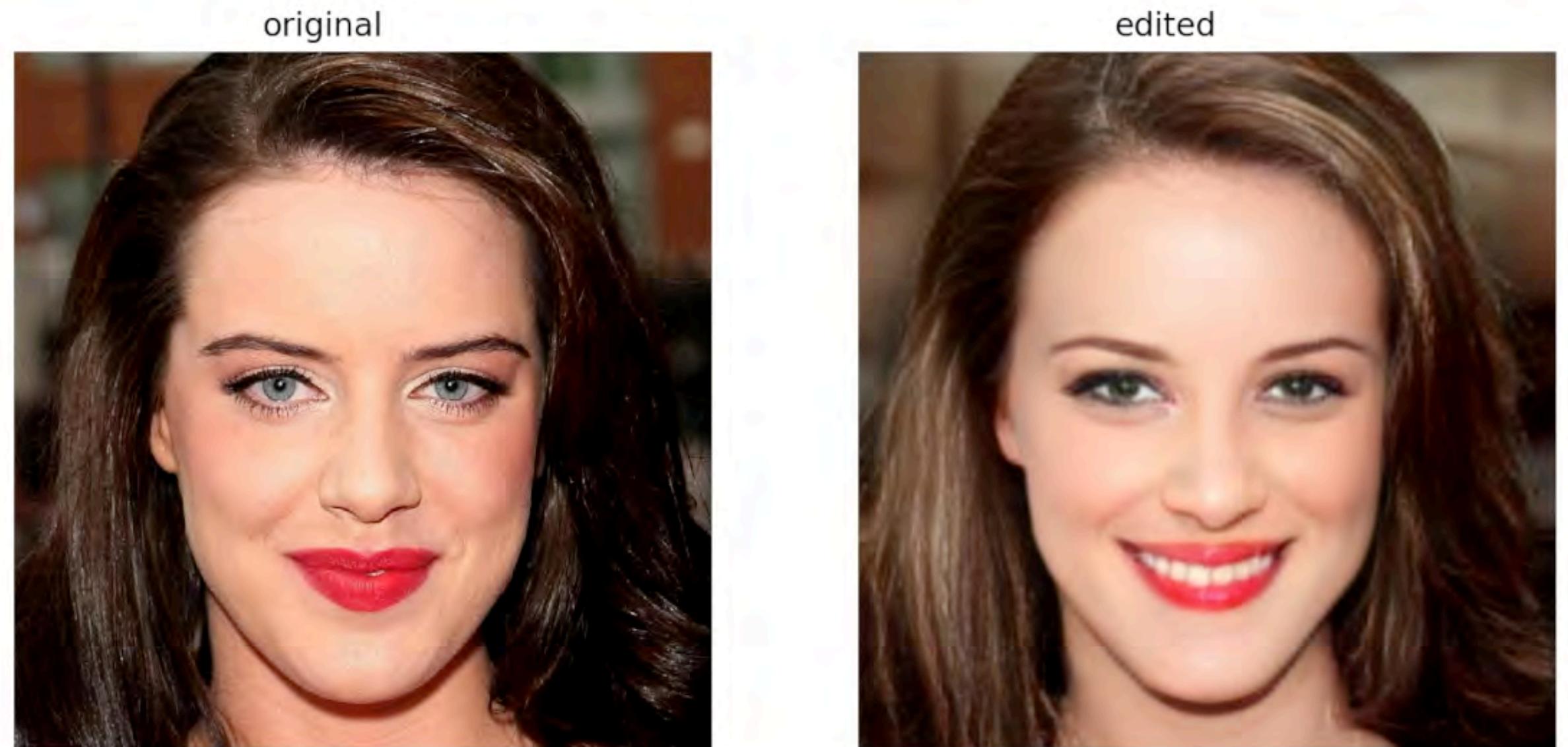
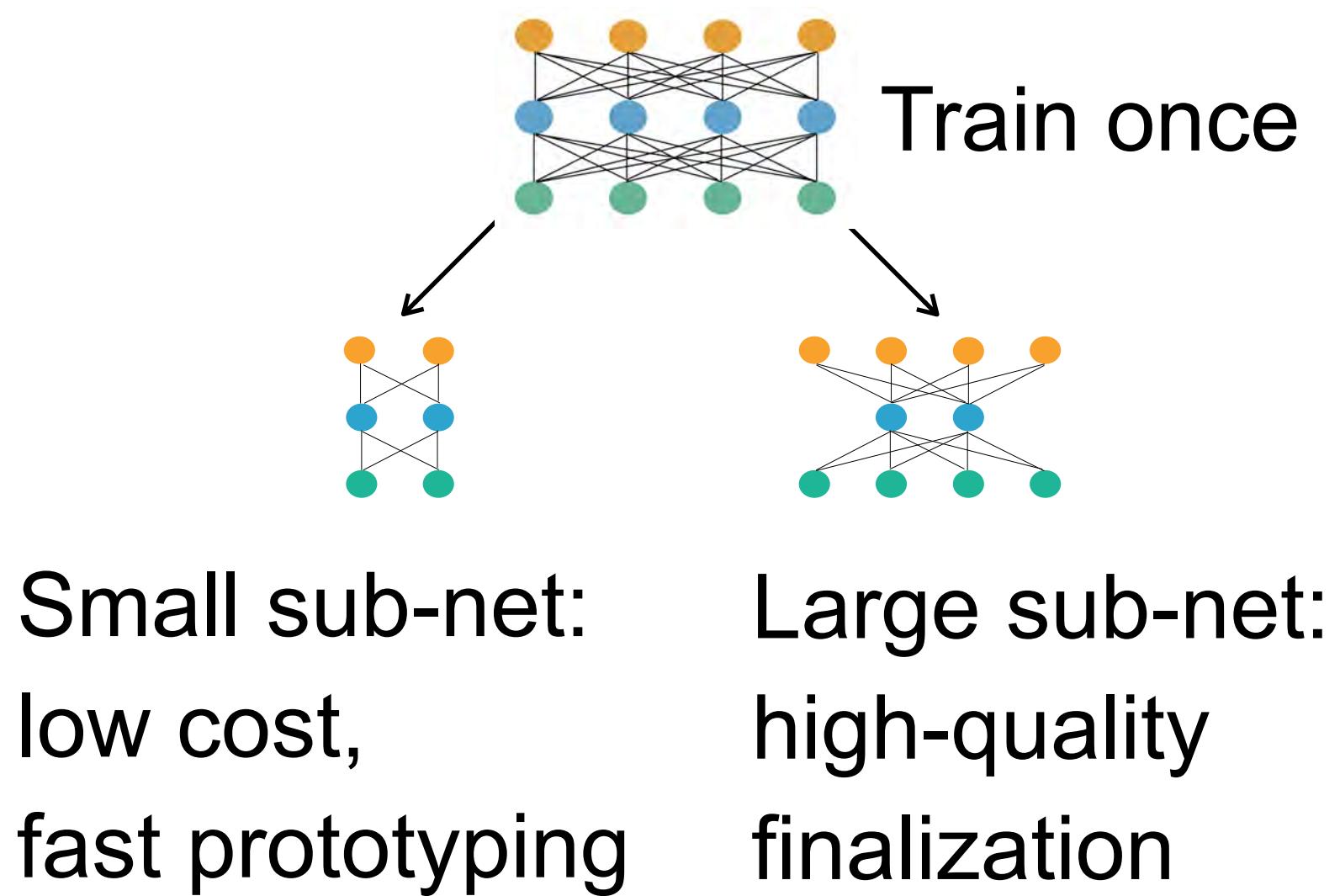
Measured on NVIDIA **Jetson Xavier GPU**
Lower FID indicates better Performance.



GAN Compression: Efficient Architectures for Interactive Conditional GANs [Li et al., CVPR 2020]

Efficient Image Generation

- Generative Adversarial Network (GAN) is computationally heavy and slow
- Difficult for interactive photo editing on mobile device (iPad)
- Anycost GAN with once-for-all network:



Anycost GAN, CVPR'21

Efficient Image Generation

AnycostGAN achieves interactive image synthesis and editing on a laptop



MACs:  100% 1.0x reduction

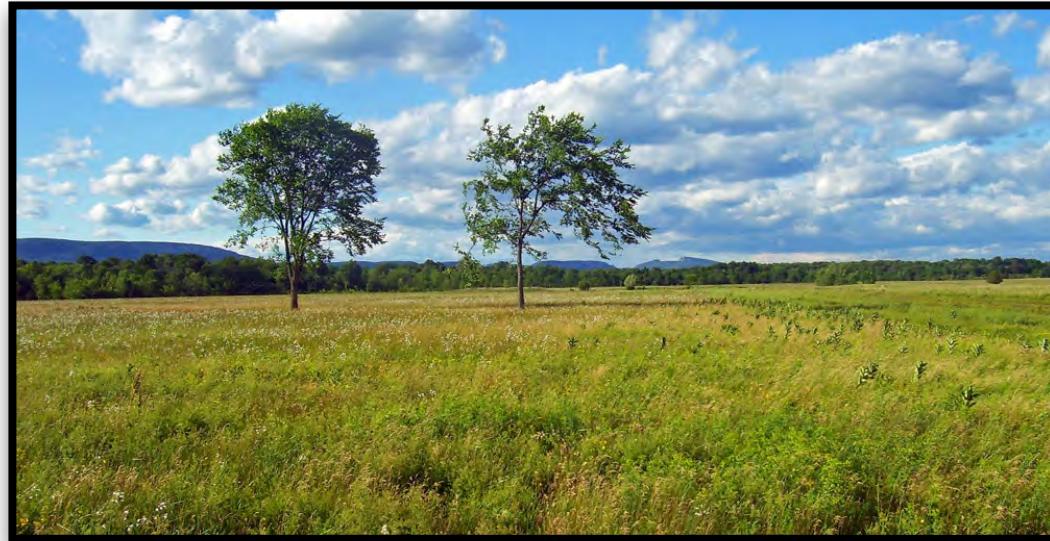
Compute Budget	1x	0.7x	0.5x	0.4x	0.2x
Tiered Pricing	\$0.01	\$0.007	\$0.005	\$0.004	\$0.002

The quality is still reasonably good

Efficient Image Generation

SIGE accelerates Stable Diffusion by >4X with spatial sparsity

A photograph of a horse on a grassland.



Original



11.6% Masked



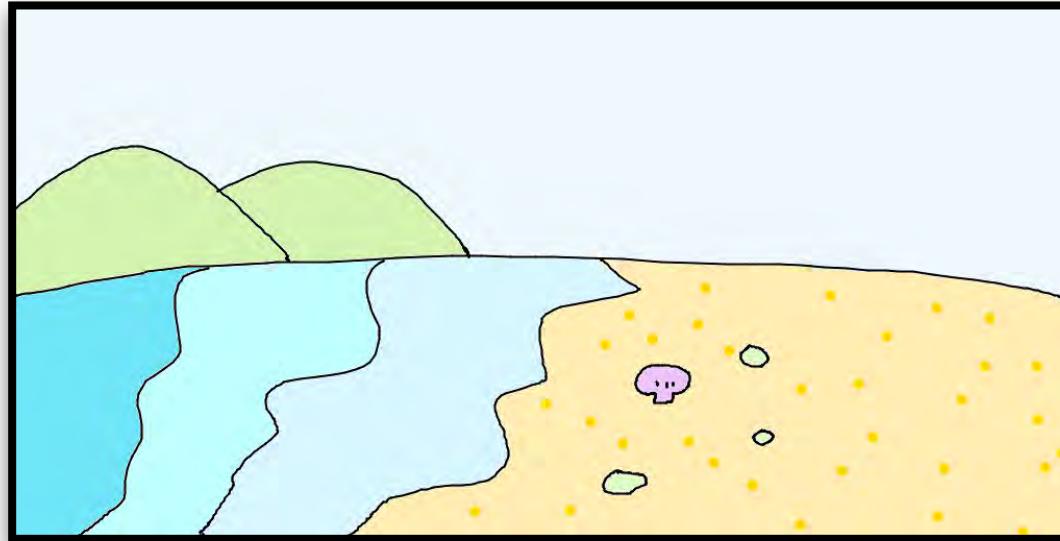
Stable Diffusion:
1855GMACs 369ms

Image Inpainting

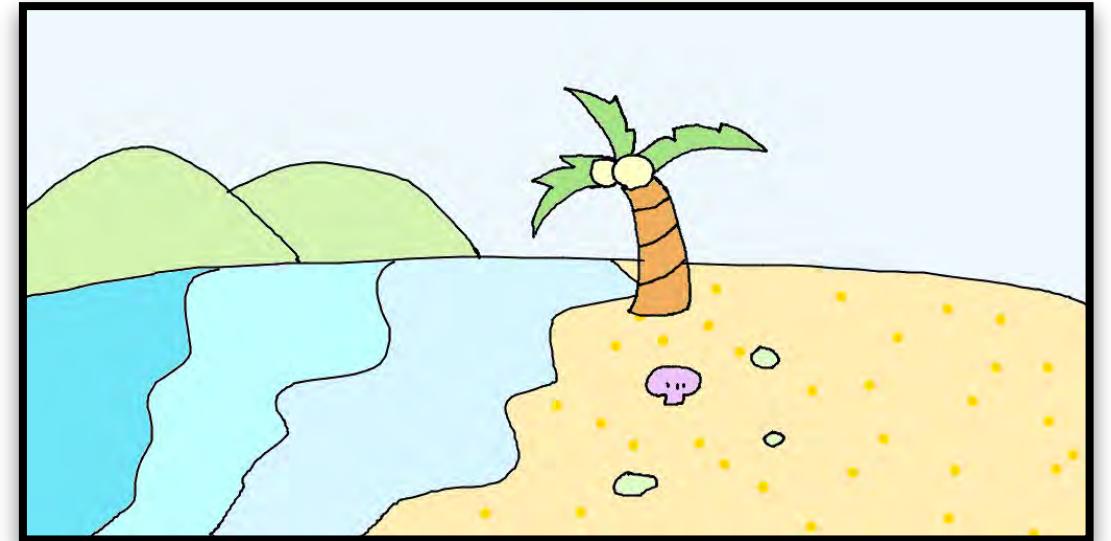


Ours:
514G (3.6×) 95.0ms (3.9×)

A fantasy beach landscape, trending on artstation.



Original



2.9% Edited



Stable Diffusion+SDEdit:
1855GMACs 369ms

Image Editing



Ours:
353G (5.3×) 76.4ms (4.8×)

Latency Measured on NVIDIA RTX 3090



Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models [Li et al., NeurIPS 2022]

Efficient Image Generation

FastComposer achieves tuning-free multi-subject image generation

- Create personalized images based on user-specified inputs
- Existing work:
 - need extra fine-tuning stage to learn subjects (computationally expensive),
 - generate poor multi-subject images (**limited composability**).
 - overfit reference images (**limited editability**).

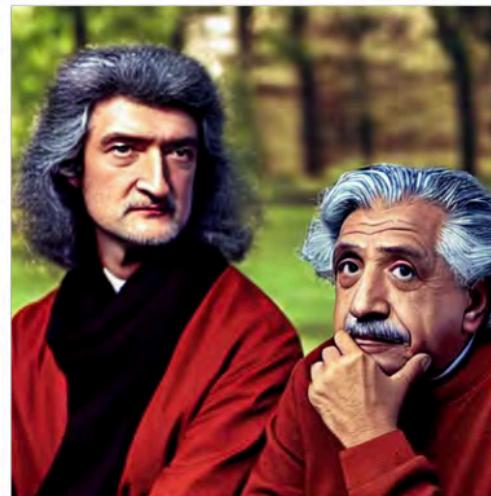
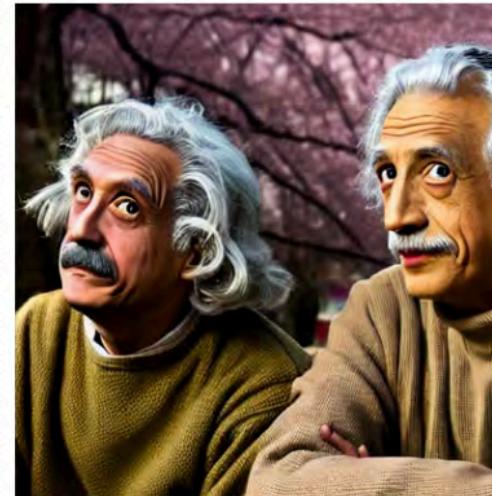
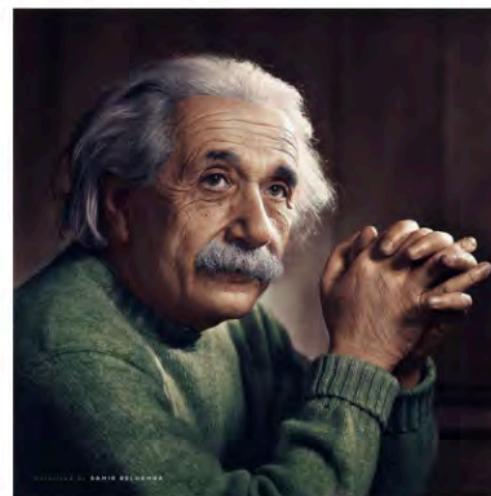


“Song Han riding a horse”

(Midjourney)

Problem 1: Identity Blending

“A man and a man sitting in a park”



Inputs

w/o cross-attention
localization

w/ cross-attention
localization (**ours**)

Problem 2: Subject Overfitting

“A woman riding a horse”



Input

w/o delayed subject
conditioning

w/ delayed subject
conditioning (**ours**)

Efficient Image Generation

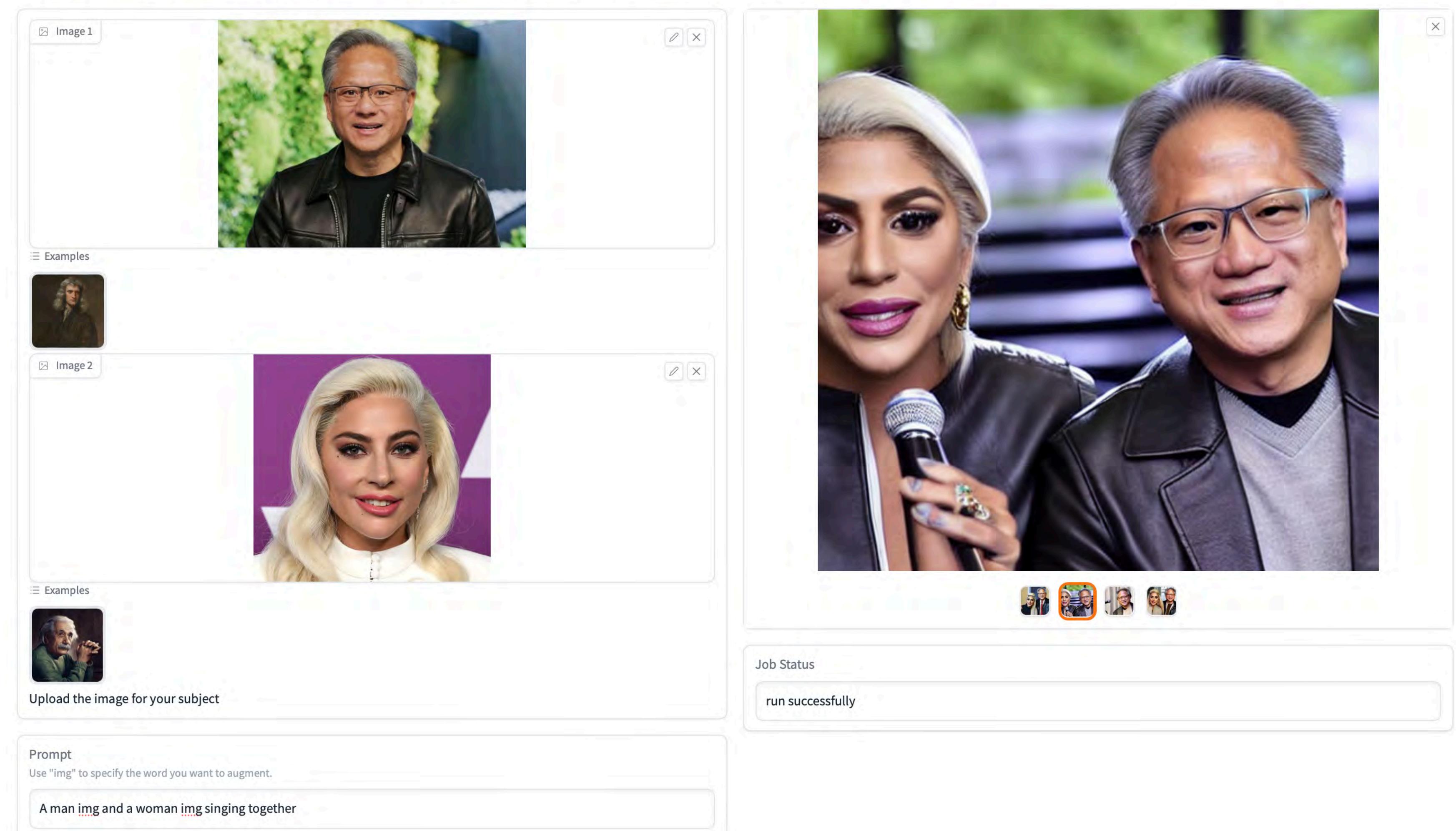
FastComposer achieves tuning-free multi-subject image generation



FastComposer: <https://fastcomposer.hanlab.ai/>

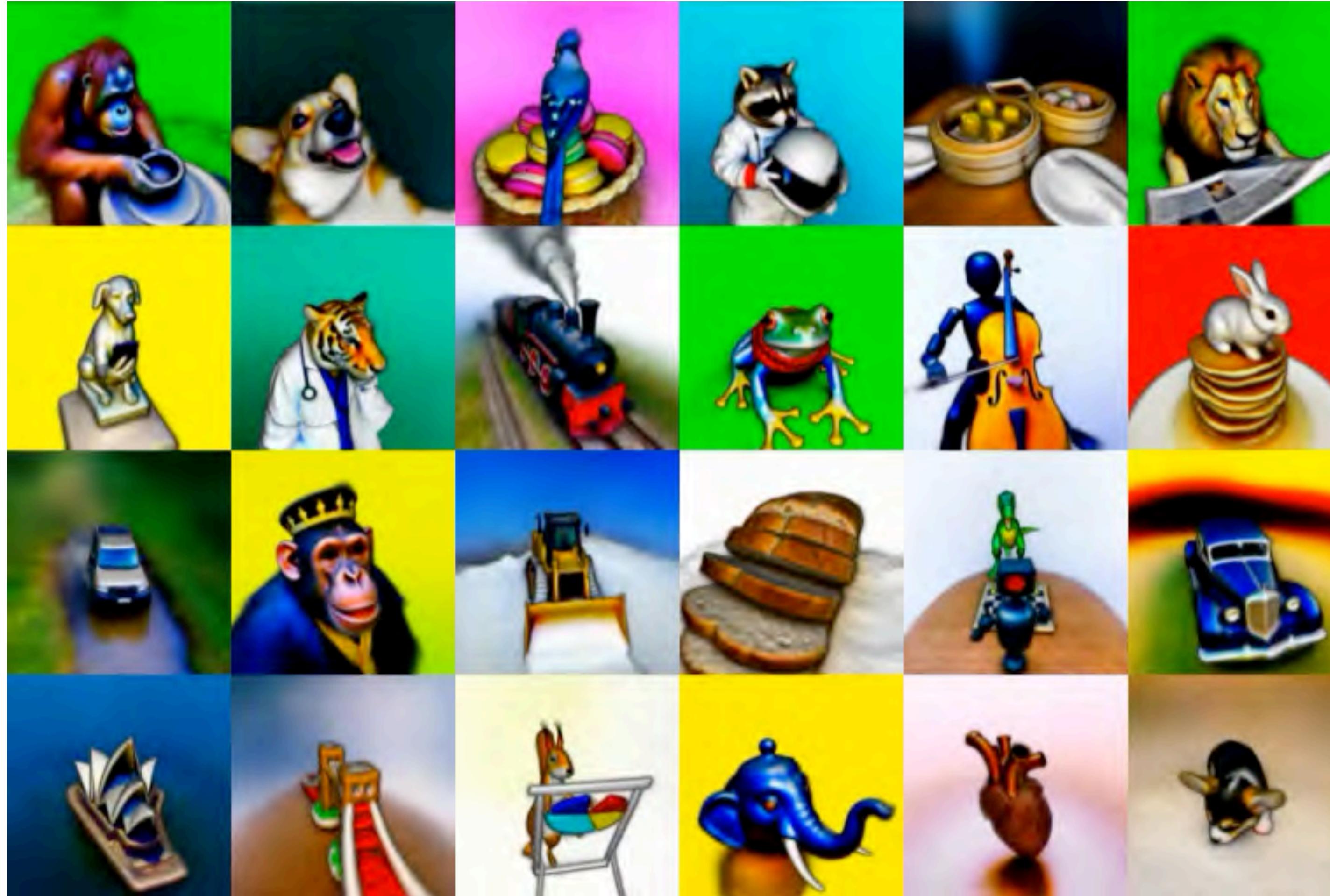
Efficient Image Generation

FastComposer achieves tuning-free multi-subject image generation



3D Generation

Diffusion models create 3D objects from a natural language description



DreamFusion: <https://dreamfusion3d.github.io/>

Video Generation

Diffusion models create realistic videos from a natural language description



“Video modeling is a harder task for which performance is not yet saturated at **5.6B model size**”

Imagen Video: <https://imagen.research.google/video/>

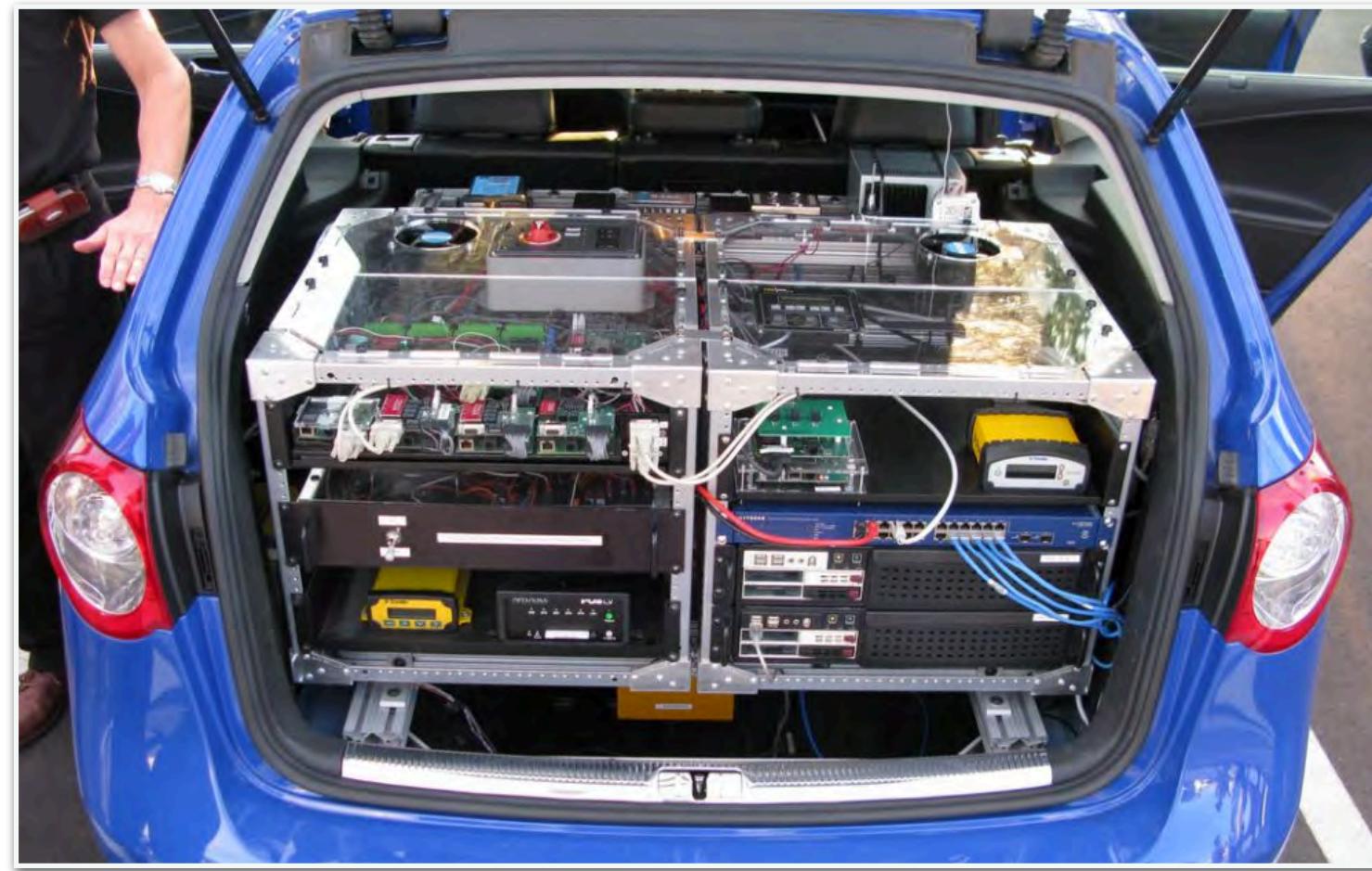
From 2D Vision to 3D Vision

Deep learning helps machine perceive the surrounding environment



Waymo Driver

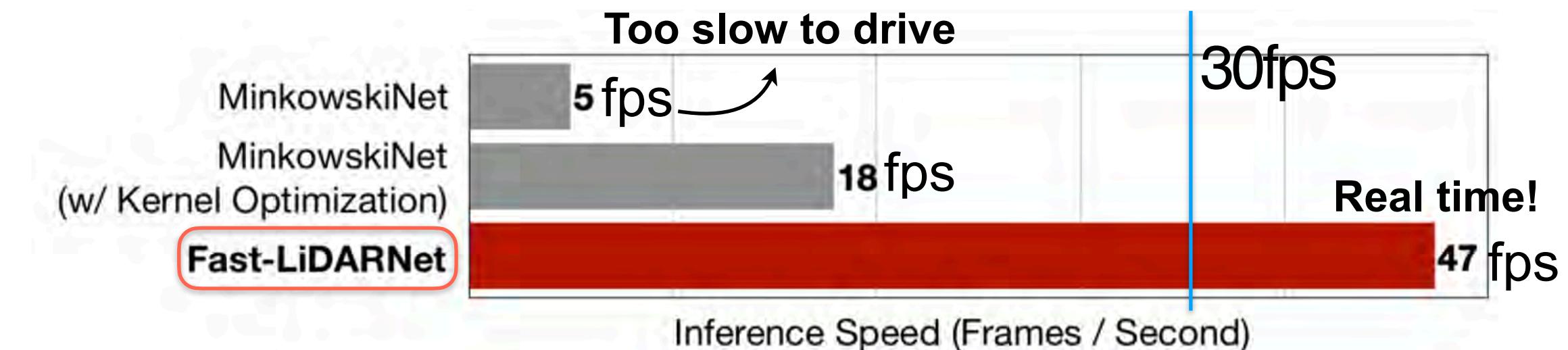
Waymo Driver: <https://www.youtube.com/watch?v=2CVInKMz9cA>



A **whole trunk** of workstation

Efficient 3D Perception

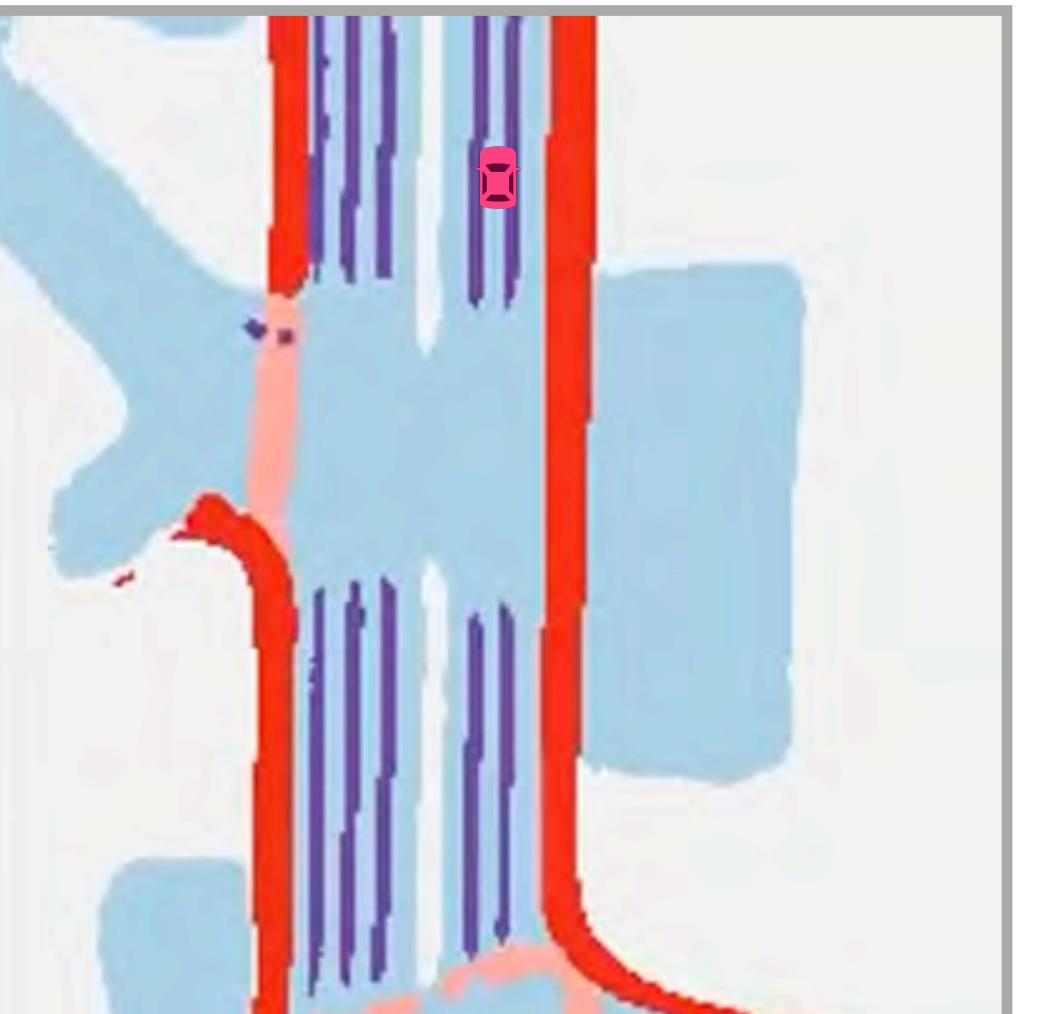
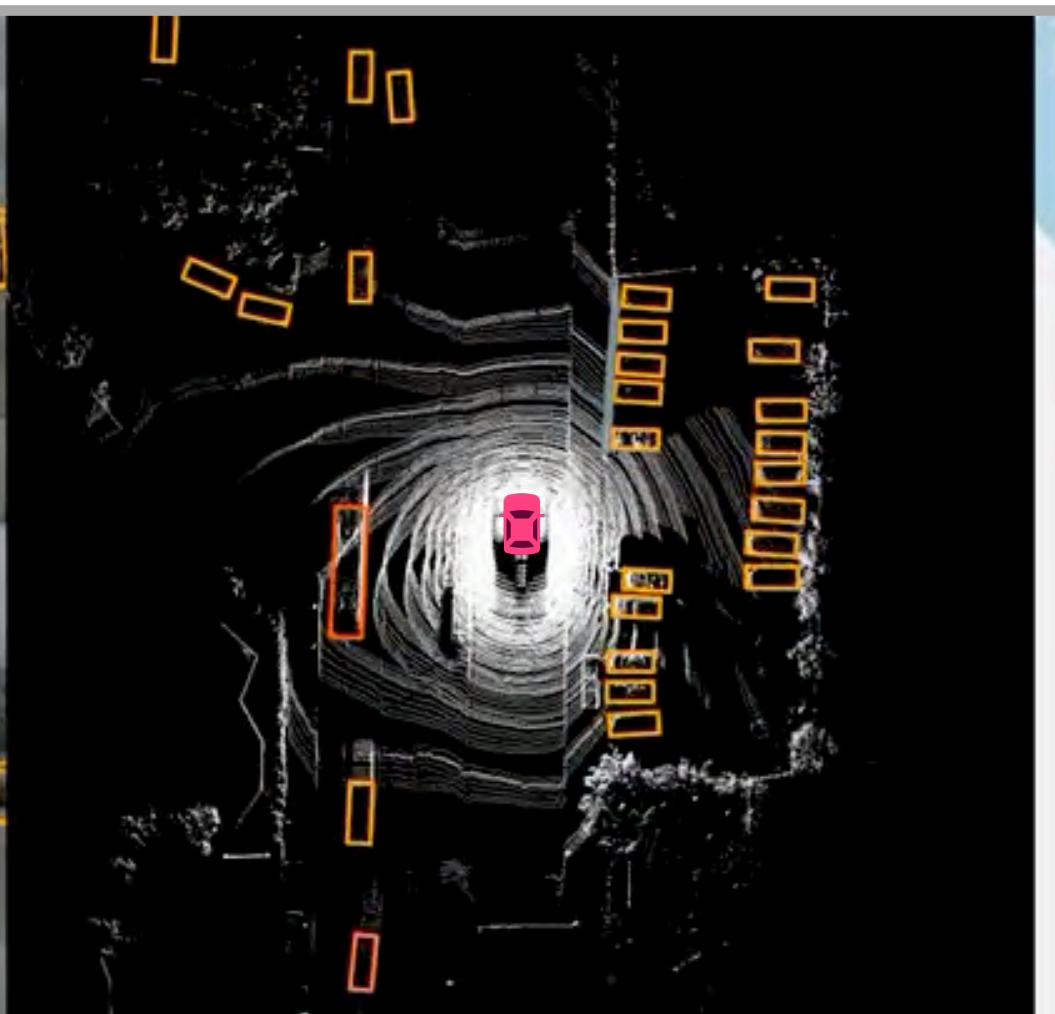
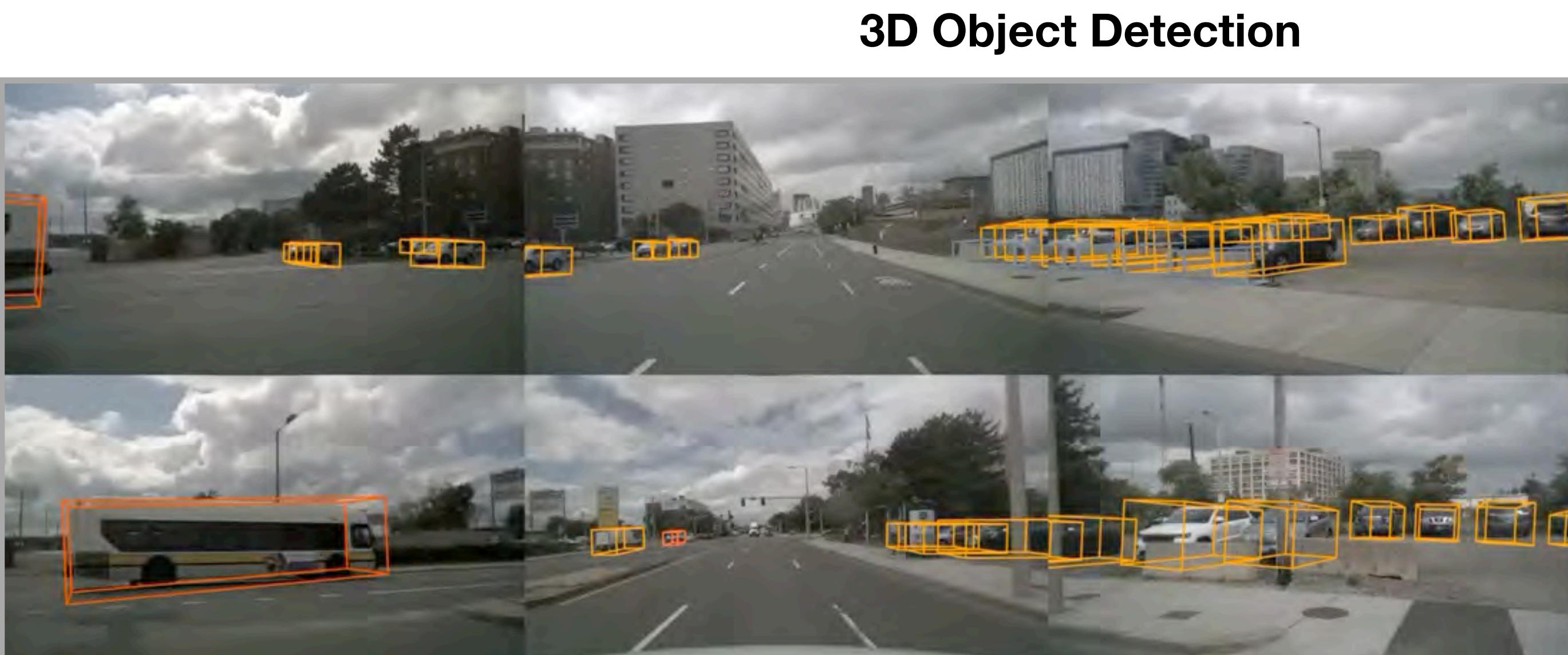
Fast-LiDARNet accelerates 3D perception with algorithm/system co-design



Efficient and Robust LiDAR-Based End-to-End Navigation [Liu et al., ICRA 2021]

Efficient 3D Perception

BEVFusion supports efficient multi-task multi-sensor fusion



Legend:
Orange square: Car Red square: Truck Blue square: Pedestrian Light blue square: Barrier
Light blue square: Drivable Area Purple square: Lane Divider Red square: Walkway Pink square: Crosswalk

BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation [Liu et al., Arxiv 2022]

Deep Learning is Everywhere



Image source: [1](#), [2](#), [3](#)

ChatGPT and Large Language Model

Large language models produce human-like text based on past conversations

HA Please draft a short description for the course "TinyML and Efficient Deep Learning Computing"?

 Course Title: "TinyML and Efficient Deep Learning Computing"

Course Description:

Unleash the power of Tiny Machine Learning (TinyML) and efficient deep learning computing in this comprehensive course designed for learners with a fundamental understanding of artificial intelligence (AI) and machine learning (ML).

Throughout this course, participants will gain a solid grasp of the emerging field of TinyML, which focuses on creating and implementing efficient machine learning models on resource-constrained devices such as microcontrollers. Our syllabus explores deep learning model optimization techniques, effective neural network architectures, and hardware-aware training approaches to make the most of limited computational capabilities.

Learners will explore real-world case studies and hands-on projects involving smartphones, IoT devices, and wearable technology, demonstrating the immense potential of TinyML applications in various sectors. By the end of the course, students will be equipped with the skills to design, develop, and deploy efficient deep learning models for edge computing, thus opening new opportunities for AI innovations that are cost-effective, power-efficient, and privacy-preserving.

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](#)

Our most capable model, great for tasks that require creativity and advanced reasoning.

Available exclusively to Plus users

GPT-4 currently has a cap of 50 messages every 3 hours.

Code Generation

GitHub CoPilot can make meaningful coding suggestions based on context

The screenshot shows a dark-themed code editor interface. At the top, there are four tabs: 'parse_expenses.py' (selected), 'write_sql.go', 'sentiment.ts', and 'addresses.rb'. The main area displays a Python script with line numbers from 1 to 20 on the left. Lines 1 through 4 are visible:

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, va
```

A cursor is positioned at the end of line 4, after the closing parenthesis of the function definition.

Image credit: <https://techcrunch.com/2021/06/29/github-previews-new-ai-tool-that-makes-coding-suggestions/>

Neural Machine Translation

Neural machine translation bridges the language barrier

The screenshot shows a comparison between English and Chinese Simplified translations. The English text discusses the challenges of deploying neural networks to mobile devices and IoT, mentioning topics like model compression, pruning, quantization, neural architecture search, distillation, and efficient training techniques. The Chinese translation provides a summary of these challenges and adds information about efficient inference techniques, specific optimizations for videos, point clouds, and NLP, and quantum machine learning. A link to 'Show more' is visible at the bottom.

ENGLISH - DETECTED ENGLISH SPANISH FRENCH CHINESE (SIMPLIFIED) ENGLISH SPANISH

Have you found it difficult to deploy neural networks on mobile devices and IoT devices? Have you ever found it too slow to train neural networks? This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices. Topics cover efficient inference techniques, including model compression, pruning, quantization, neural architecture search, distillation; and efficient training techniques, including gradient compression and on-device transfer learning; followed by application-specific model optimization techniques for videos, point cloud and NLP; and efficient quantum machine learning. Students will get hands-on experience implementing deep learning applications on microcontrollers, mobile phones and quantum machines with an open-ended design project related to mobile AI.

您是否发现很难在移动设备和物联网设备上部署神经网络？你有没有发现训练神经网络太慢了？本课程深入探讨有效的机器学习技术，这些技术可在资源受限的设备上实现强大的深度学习应用。主题涵盖高效推理技术，包括模型压缩、剪枝、量化、神经架构搜索、蒸馏；和高效的训练技术，包括梯度压缩和设备迁移学习；其次是针对视频、点云和 NLP 的特定应用模型优化技术；和高效的量子机器学习。学生将通过与移动 AI 相关的开放式设计项目获得在微控制器、手机和量子机器上实施深度学习应用程序的实践经验。

Nín shìfǒu fāxiànr hěn nán zài yídòng shèbèi hé wù liánwǎng shèbèi shàng bùshǔ shénjīng wǎngluò? Nǐ yǒu méiyǒu fāxiànr xùnliàn shénjīng wǎngluò tài mǎnle? Běn kèchéng shēnrù tantǎo yóuxiào de jīqì xuéxí jishù, zhèxiē jishù kě zài zīyuán shòu xiàn de

Show more

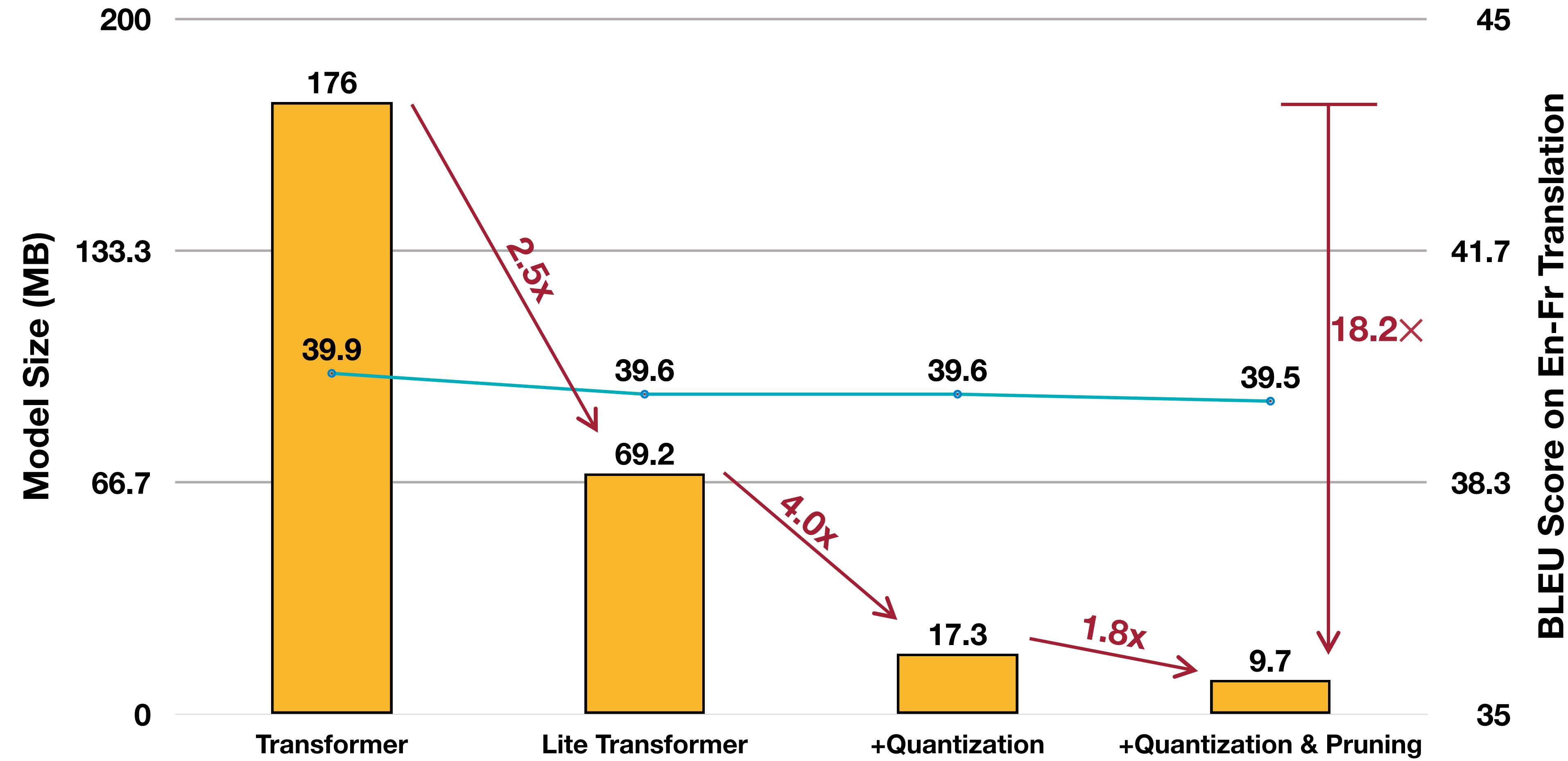


Conversation Translation
(on iPhone)

Google Translate: <https://translate.google.com/>

Efficient Neural Machine Translation

Lite Transformer reduces the model size with pruning and quantization



Lite Transformer with Long-Short Range Attention [Wu et al., ICLR 2020]

Large Language Models

Large language models show emergent behaviors: zero/few learning

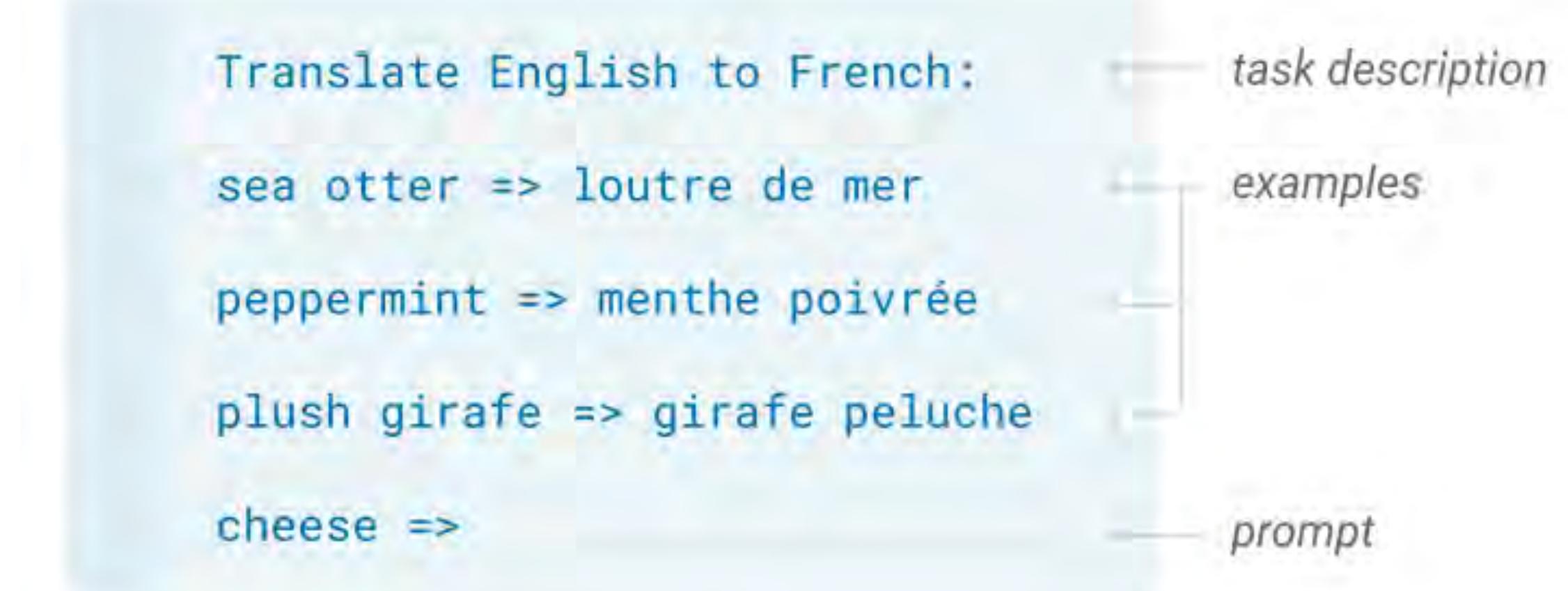
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



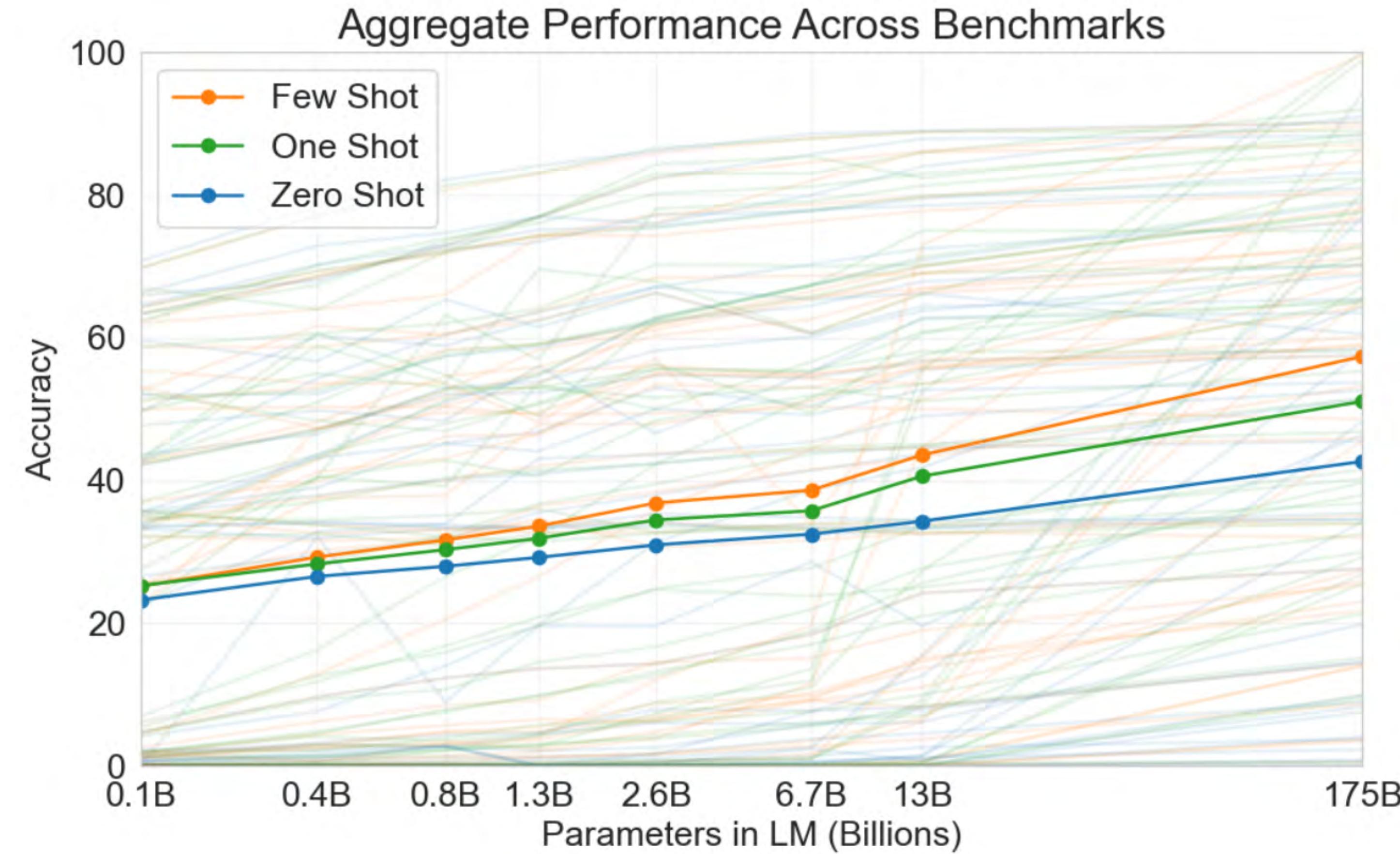
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Large Language Models

But it comes at the cost of large model size



Brown et al., GPT-3, 2020

Large Language Models

Large language models show emergent behaviors: chain-of-thought

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. 

Large Language Models

Large language models show emergent behaviors: chain-of-thought

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. 

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

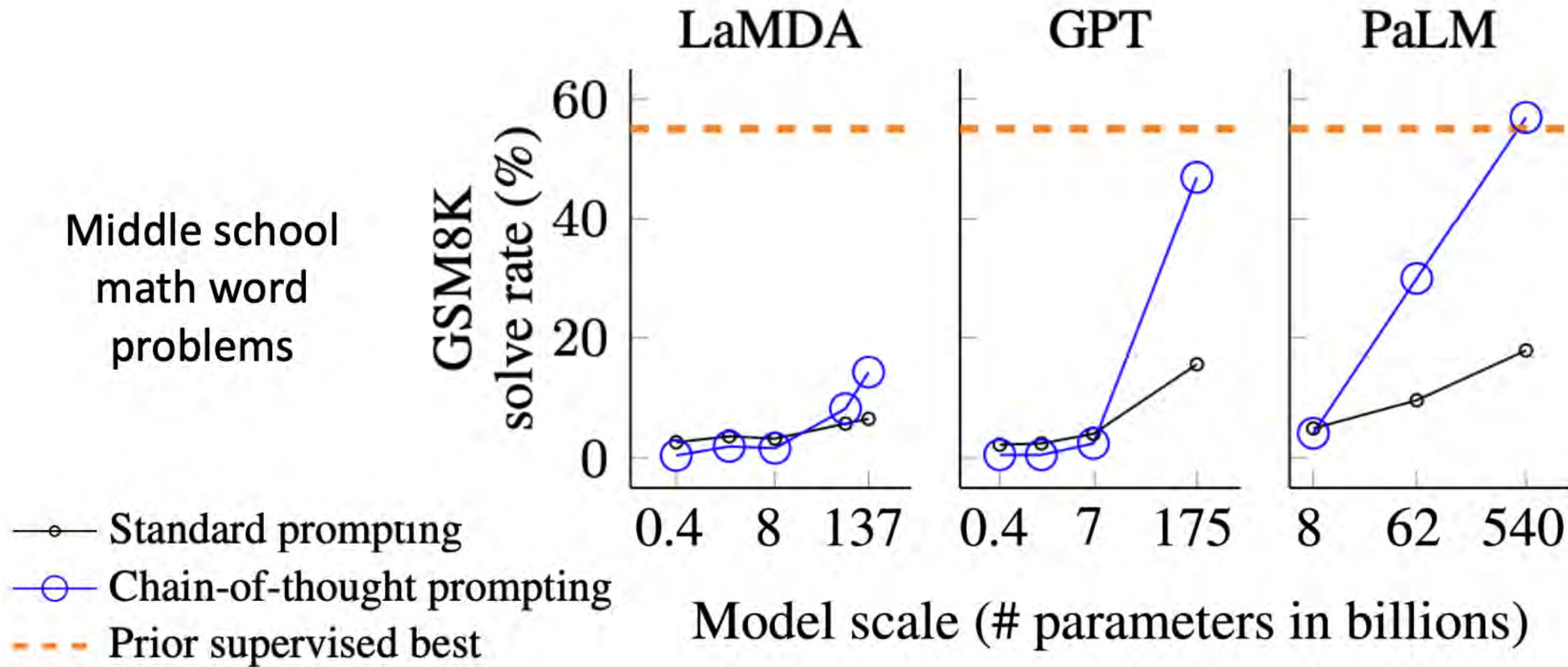
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. 

Large Language Models

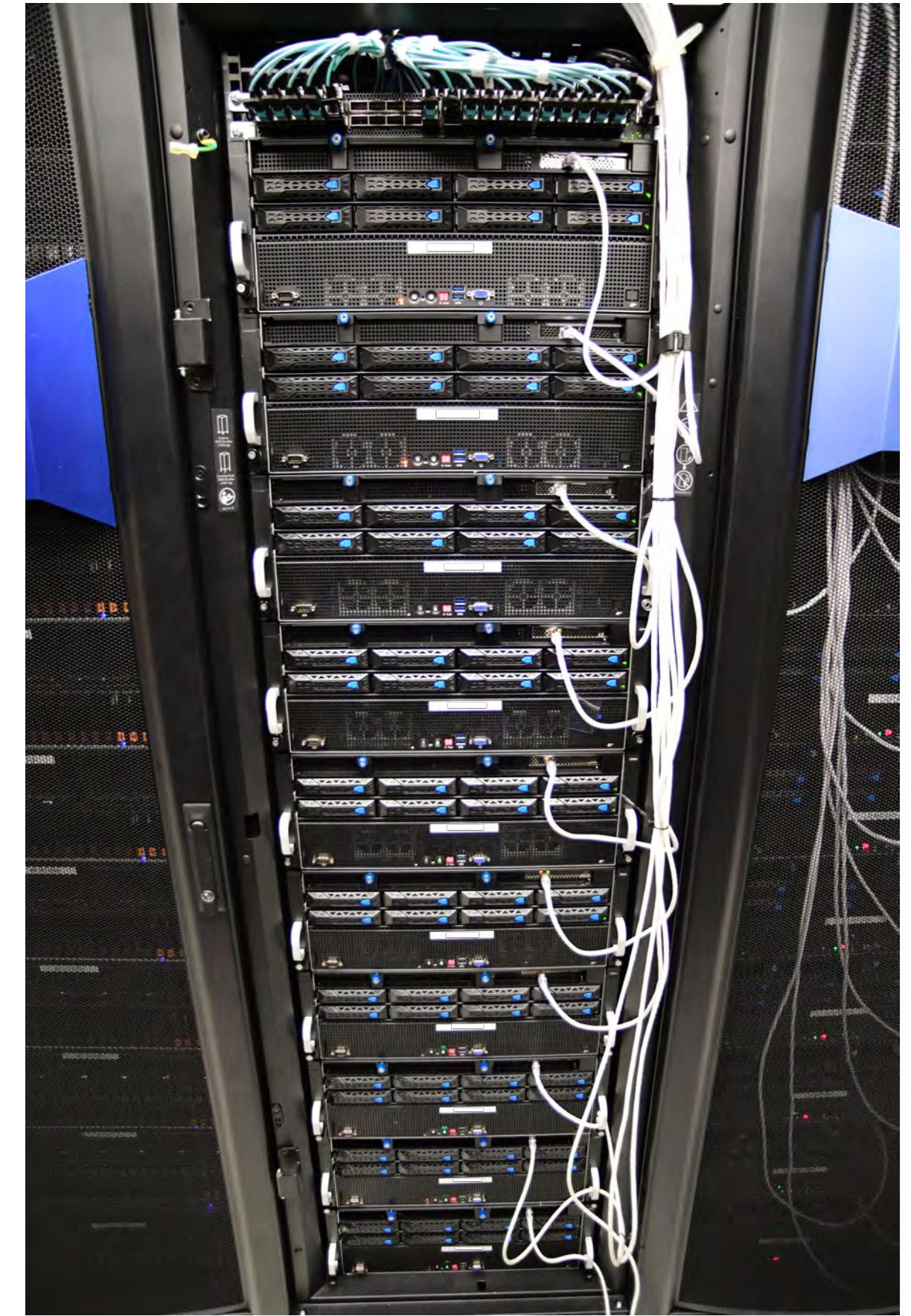
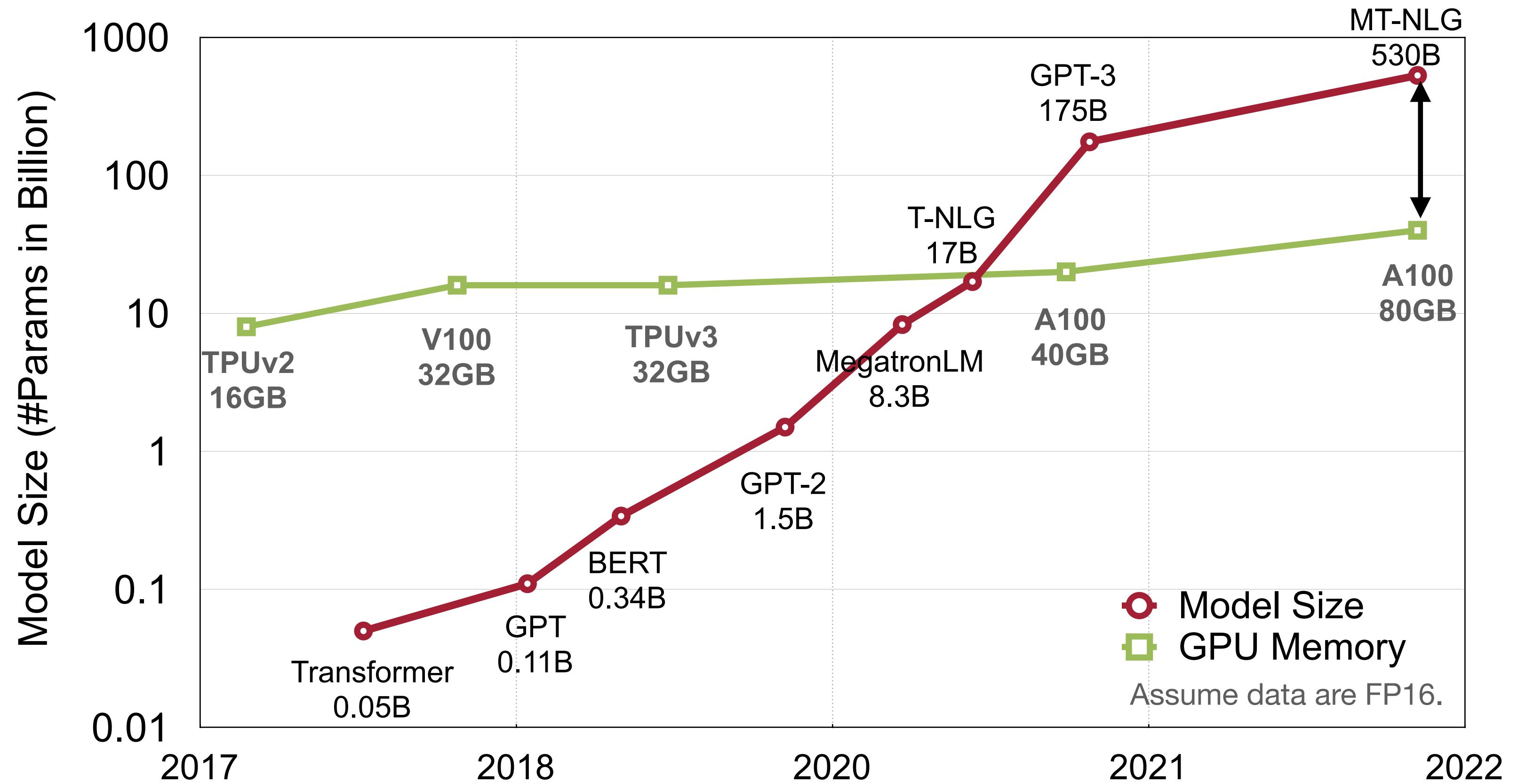
But it comes at the cost of large model size



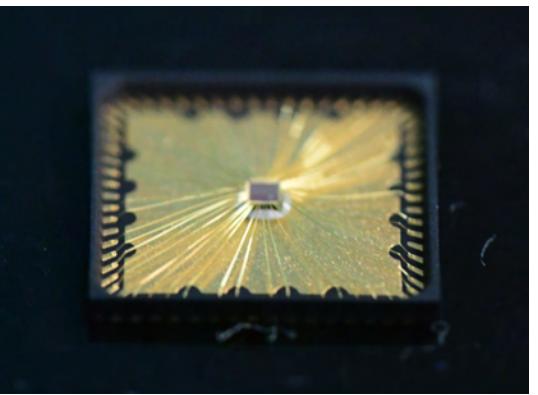
Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2022

Large Language Models

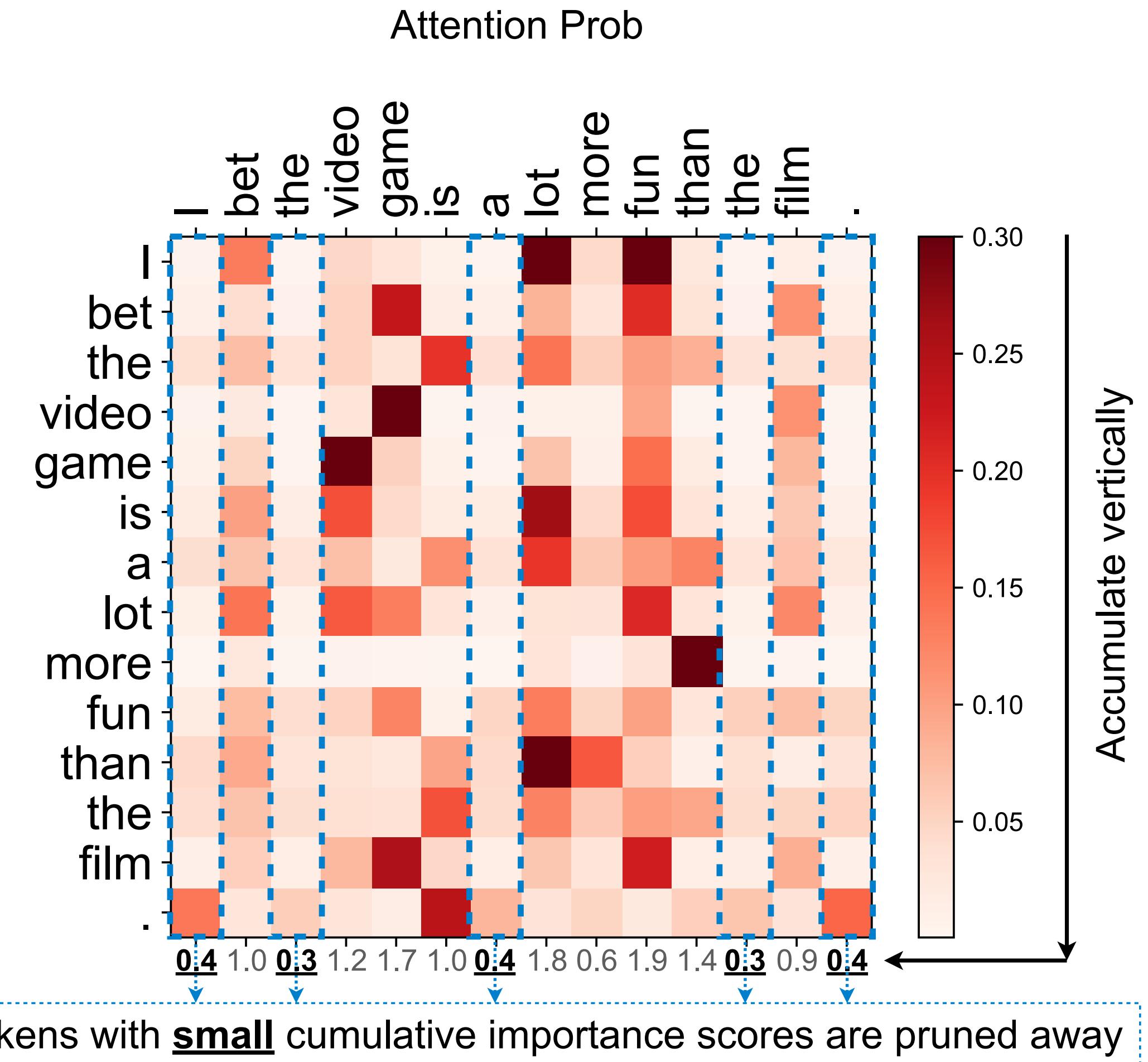
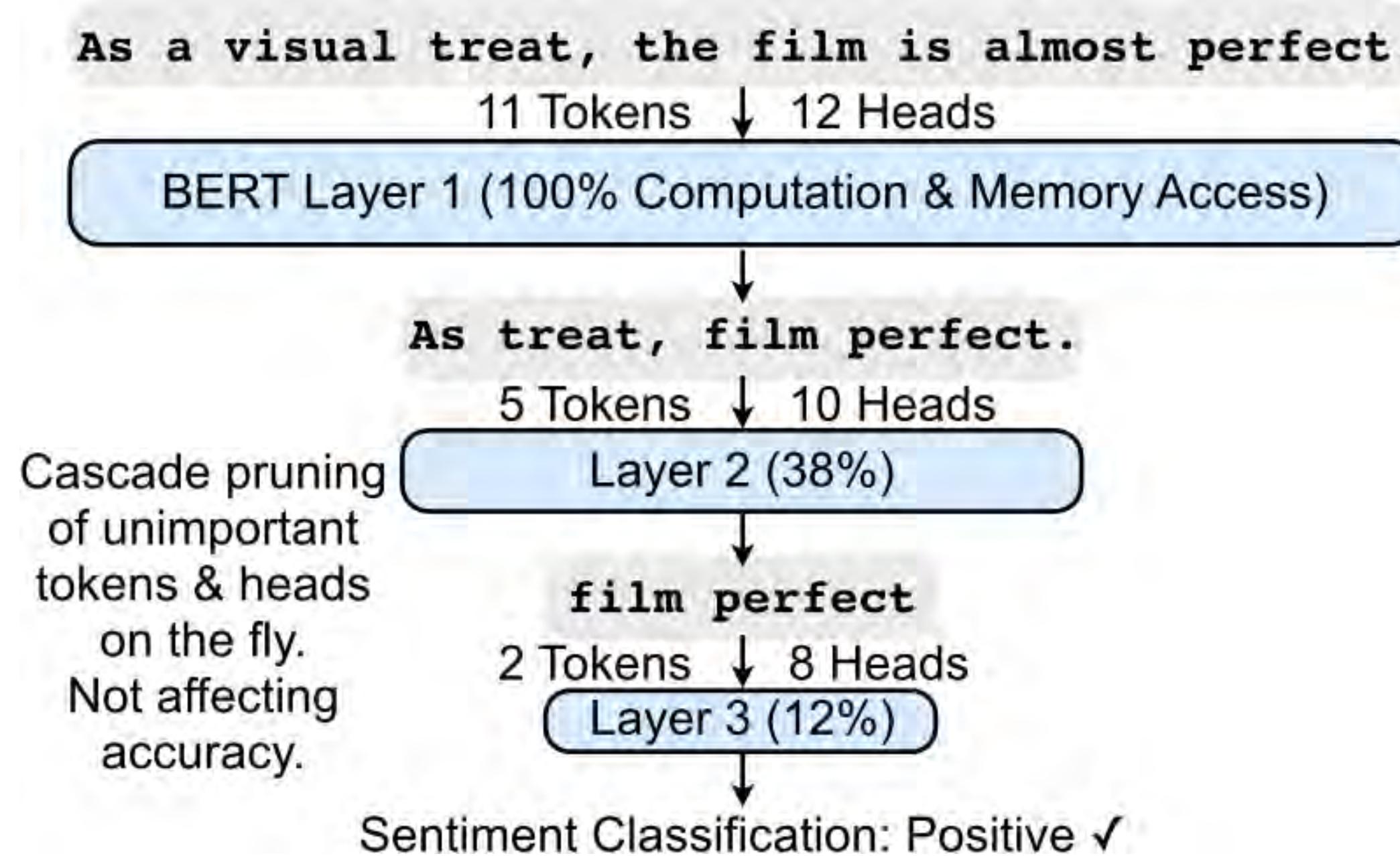
Model size of language models is growing exponentially



Efficient Large Language Models



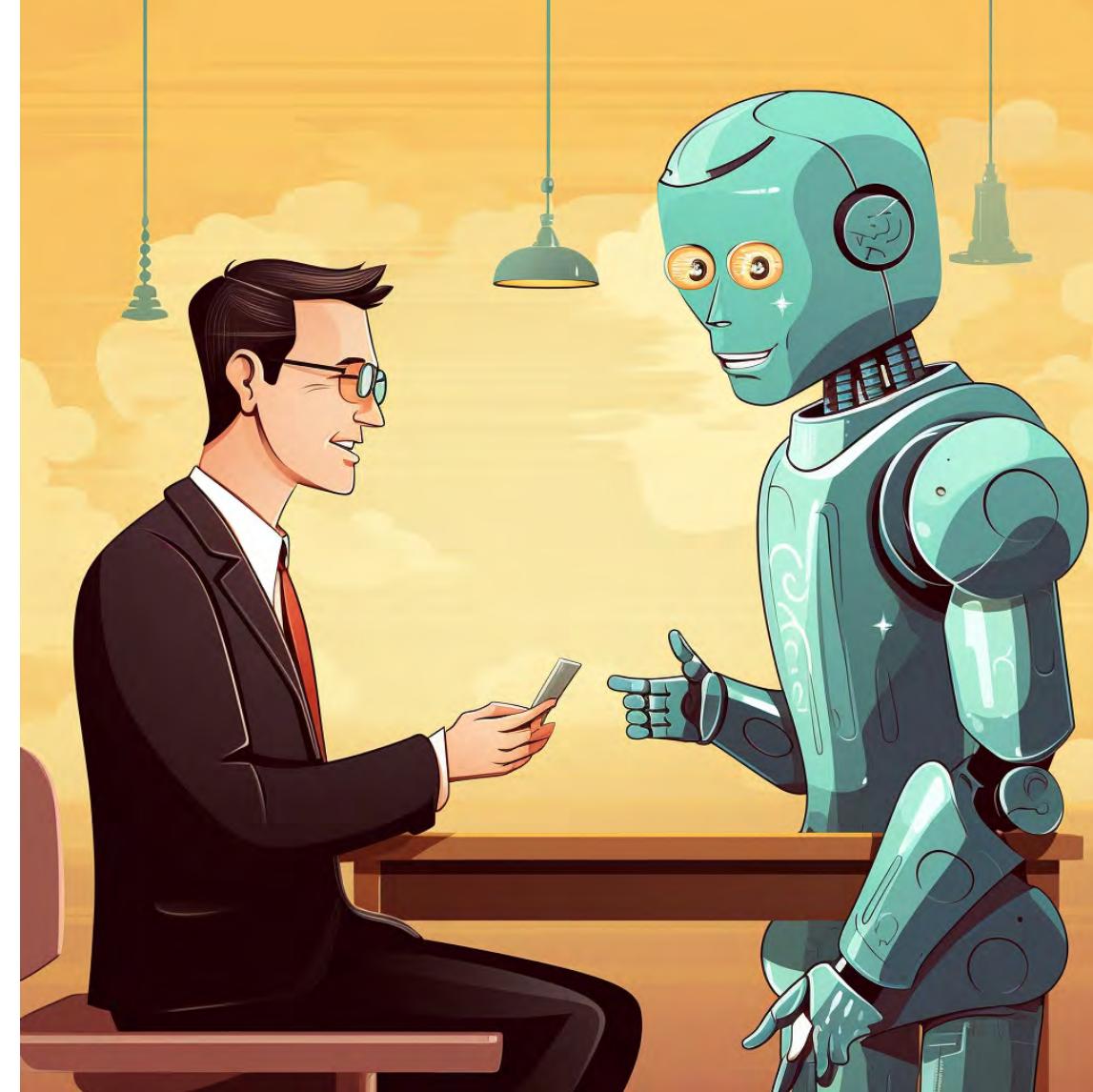
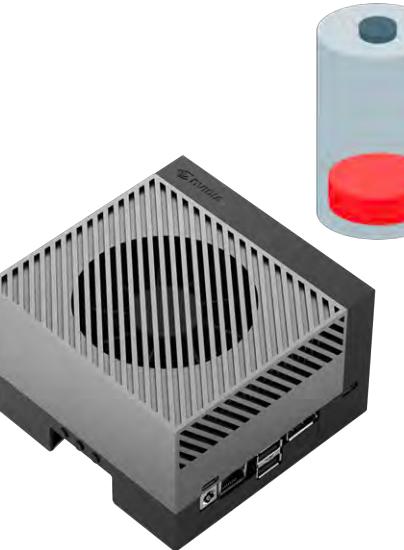
SpAtten accelerates language models by pruning redundant tokens



SpAtten: Efficient Natural Language Processing [Wang et al., HPCA 2021]

Efficient Large Language Models

Running LLMs on the edge is very important



- Deploying LLM on the edge is useful: running copilot services (code completion, office, game chat) locally on laptops, cars, robots, and more. These devices are **resource-constrained**, **low-power** and sometimes **do not have access to the Internet**.
- **Data privacy** is important. Users do not want to share personal data with large companies.

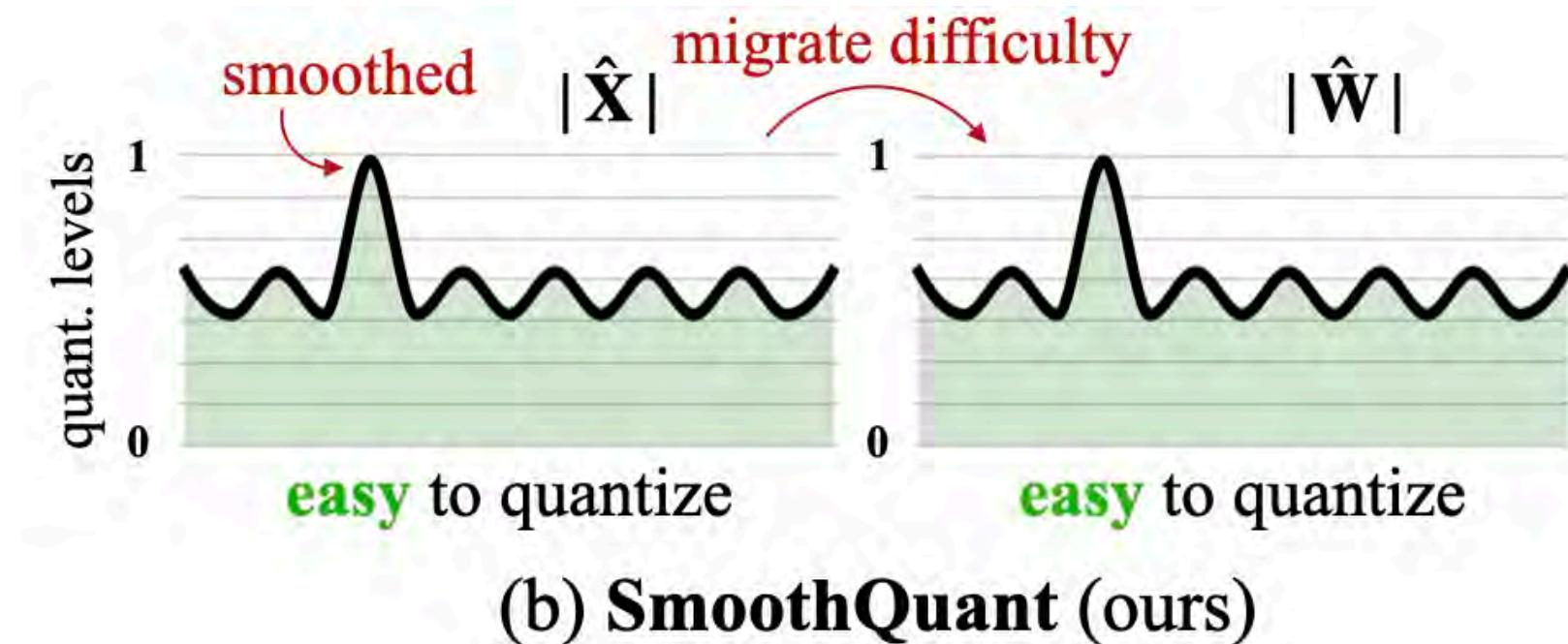
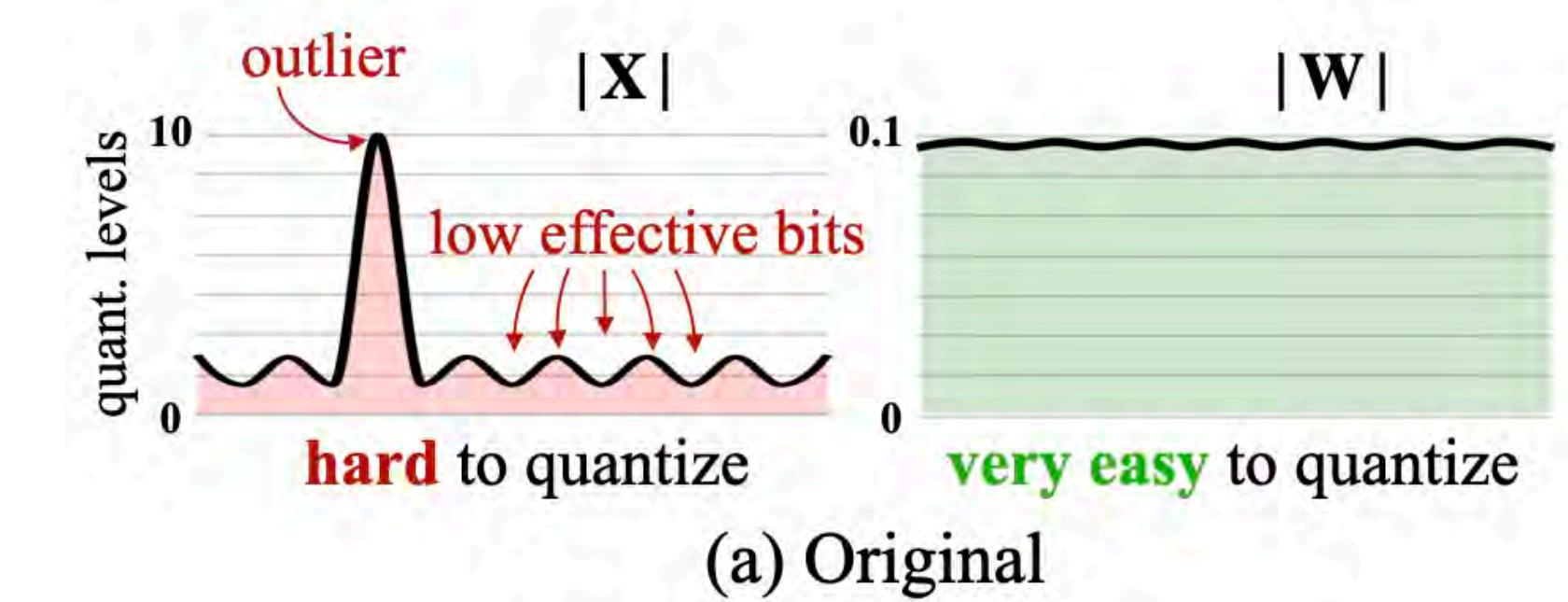
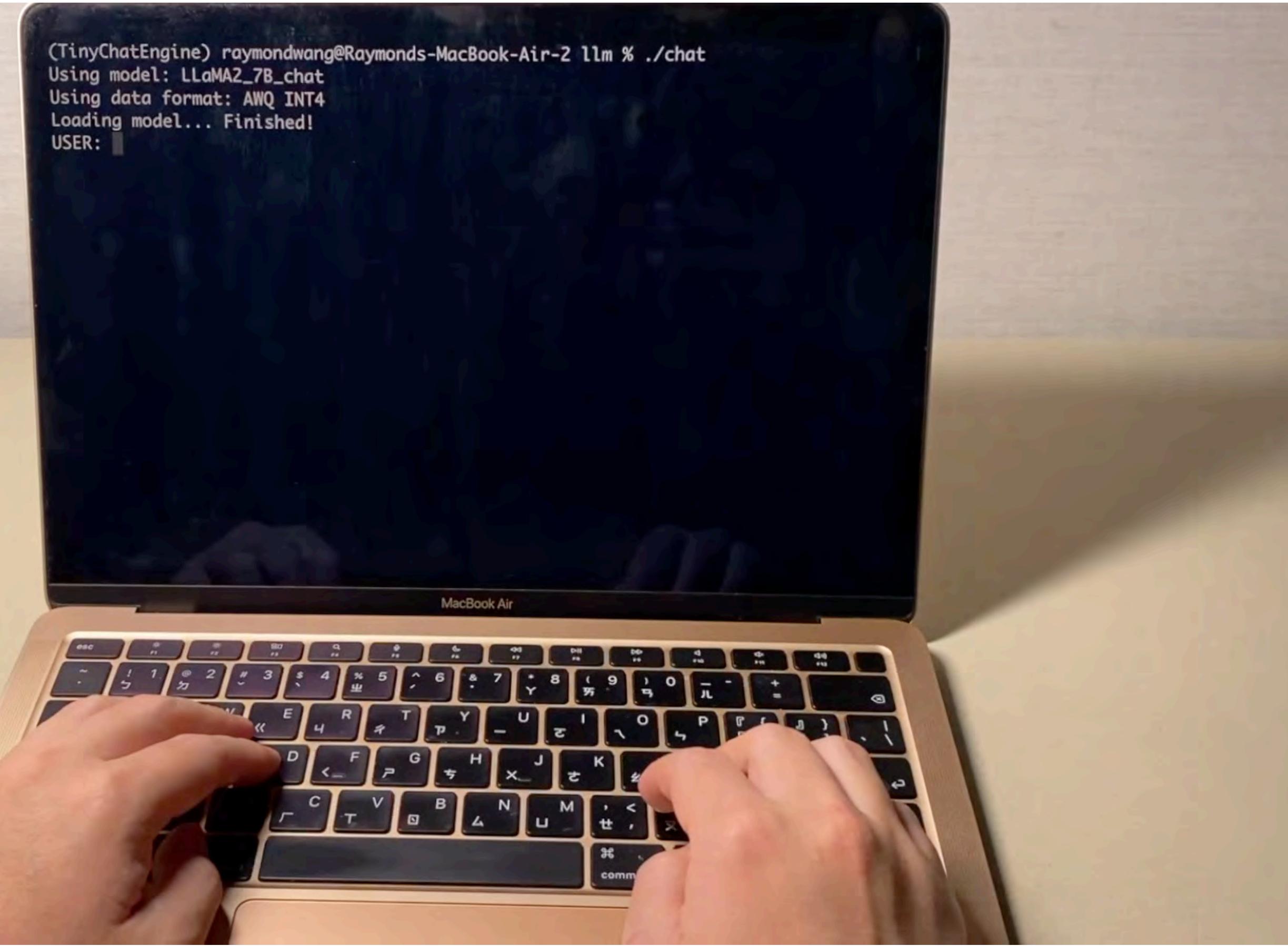
Images are generated by Midjourney

Efficient Large Language Models



TinyChat

- LLMs are too large to fit into the memory of edge devices. We enable edge deployment of LLMs through quantization: SmoothQuant and AWQ
- TinyChatEngine implements the compressed inference, built from C/C++ from scratch, easy to install and migrate to edge platforms



TinyChatEngine: <https://github.com/mit-han-lab/TinyChatEngine>

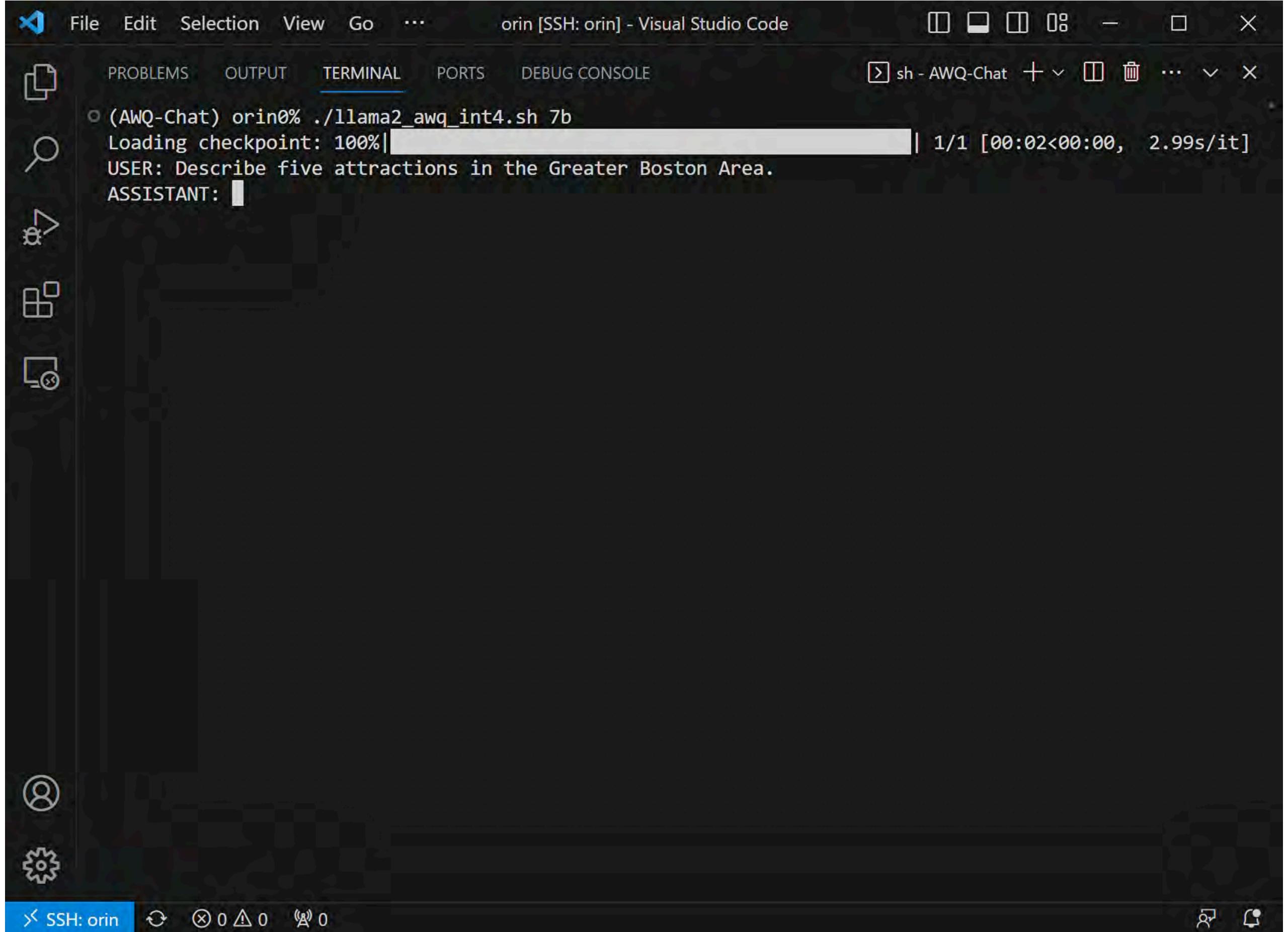
TinyChat on Orin

Running LLMs on resource-constrained edge GPUs



TinyChat on Orin

TinyChat delivers 30 tokens / second performance for LLaMA2



The screenshot shows a Visual Studio Code interface connected via SSH to an Orin device. The terminal tab is active, displaying the command `./llama2_awq_int4.sh 7b` and its execution progress. The output shows the model loading a checkpoint at 100% and responding to a user query about attractions in the Greater Boston Area.

```
(AWQ-Chat) orin0% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100%
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT:
```

LLaMA-2-7B (W4A16, AWQ): 30 tokens / s

TinyChat on Orin

TinyChat enables local inference of 13B LLMs on edge devices



A screenshot of a Visual Studio Code interface showing an SSH terminal session connected to an Orin device. The terminal window is titled "orin [SSH: orin] - Visual Studio Code". The terminal tab is active, displaying the command `(AWQ-Chat) orin% ./llama2_awq_int4.sh 13b` followed by "Loading checkpoint: 100%". Below this, a user prompt "USER: How do you compare MIT and Harvard?" is shown, and the AI response "ASSISTANT: " is partially visible. The left sidebar of VS Code includes icons for PROBLEMS, OUTPUT, TERMINAL, PORTS, and DEBUG CONSOLE. The bottom status bar shows the connection details "SSH: orin".

LLaMA-2-13B (W4A16, AWQ): 17 tokens / s

TinyChat



TinyChat brings about 3.3x speedup to LLaMA-2 on 4090

shang [SSH: Hanlab_4090] - Visual Studio Code

TERMINAL

```
(AWQ-Chat) x4% ./llama2_fp16.sh 7b
Loading checkpoint shards: 100%|██████████| 2/2 [00:05<00:00, 2.70s/it]
USER:
```

shang [SSH: Hanlab_4090] - Visual Studio Code

TERMINAL

```
(AWQ-Chat) x4% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100%|██████████| 1/1 [00:01<00:00, 1.27s/it]
USER:
```

LLaMA-2-7B (FP16): 50 tokens / s



LLaMA-2-7B (W4A16, AWQ): 166 tokens / s*

All-Python user interface, diverse and flexible support for different LLMs (e.g. LLaMA, MPT, Falcon)

*: If measured in exLLaMA's setting, we can achieve 195 tokens / s.

TinyChat: <https://tinychat.hanlab.ai>

TinyChat

TinyChat seamlessly supports different LLM architectures



```
File Edit Selection View Go ... orin [SSH: orin] - Visual Studio Code
PROBLEMS OUTPUT PORTS TERMINAL DEBUG CONSOLE
(AWQ-Chat) orin% ./mpt_awq_int4.sh 7b
real weight quantization...(init only): 100%| 32/32 [00:00<00:00, 314.76it/s]
Loading checkpoint: 100%| 1/1 [00:01<00:00, 1.54s/it]
USER: 
```

```
File Edit Selection View Go ... orin [SSH: orin] - Visual Studio Code
PROBLEMS OUTPUT PORTS TERMINAL DEBUG CONSOLE
(AWQ-Chat) orin% ./falcon_awq_int4.sh 7b
real weight quantization...(init only): 100%| 32/32 [00:00<00:00, 127.73it/s]
Loading checkpoint: 100%| 1/1 [00:02<00:00, 2.72s/it]
USER: 
```

```
File Edit Selection View Go ... orin [SSH: orin] - Visual Studio Code
PROBLEMS OUTPUT PORTS TERMINAL DEBUG CONSOLE
(AWQ-Chat) orin% ./vicuna_awq_int4.sh 7b
Loading checkpoint: 100%| 1/1 [00:01<00:00, 1.69s/it]
USER: 
```



MPT-7B (W4A16):
31 tokens / s



Falcon-7B (W4A16):
22 tokens / s



Vicuna-7B (W4A16):
33 tokens / s

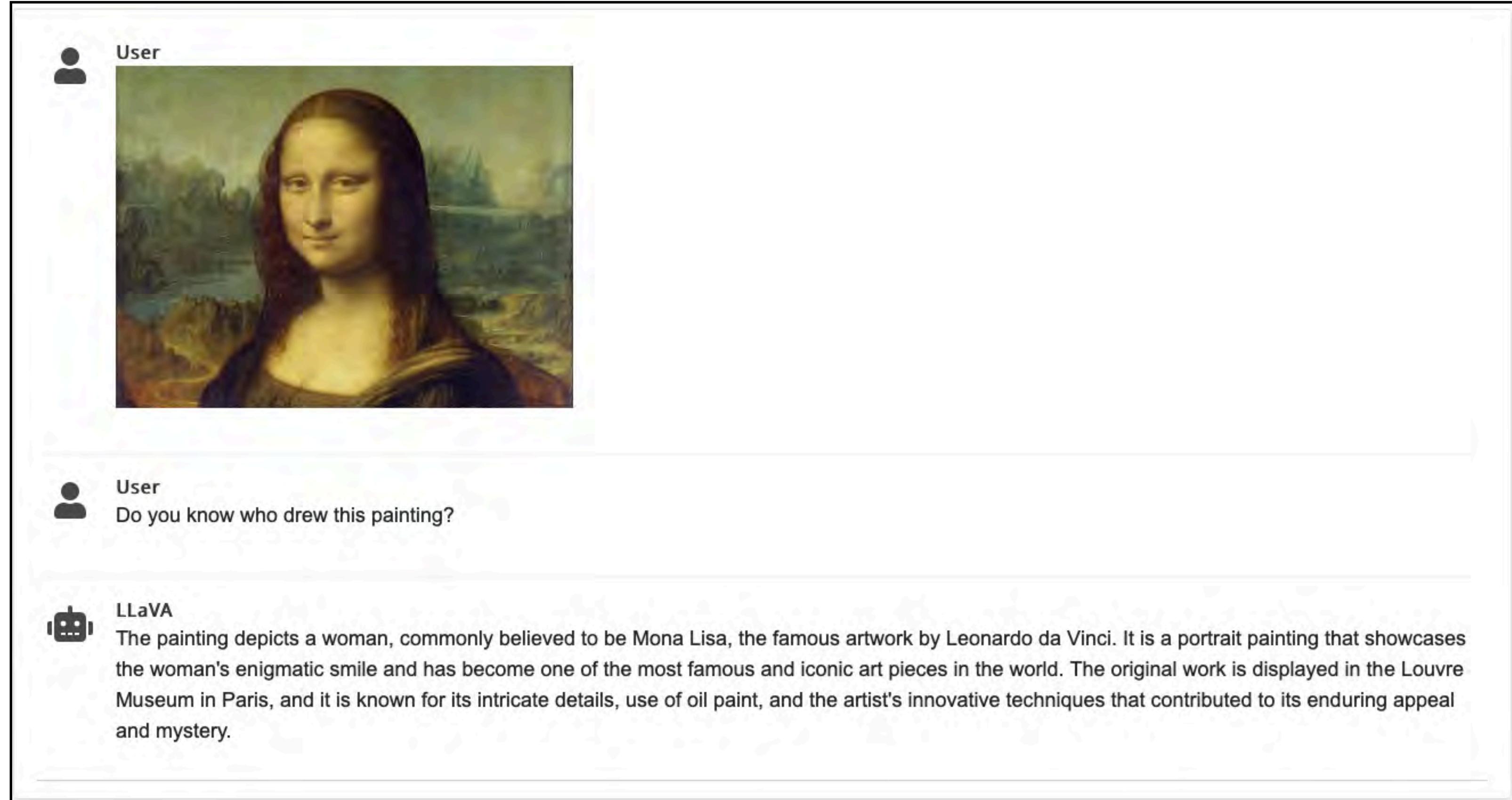
Deep Learning is Everywhere



Image source: [1](#), [2](#), [3](#)

Vision-Language Models

LLaVA achieves general-purpose visual and language understanding



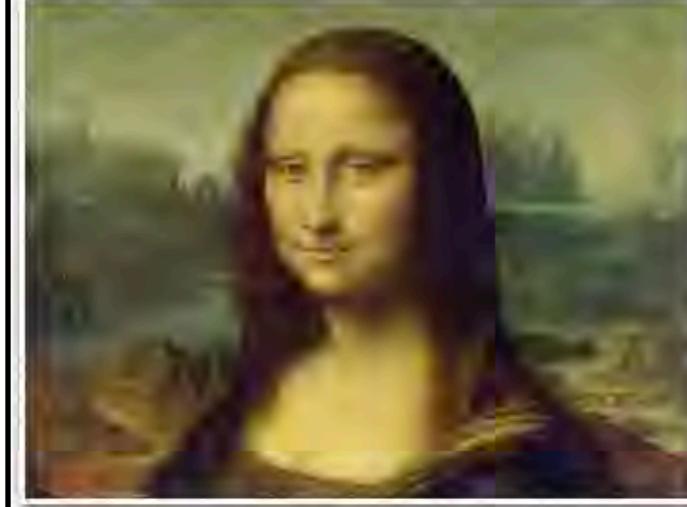
LLaVA uses a 13B LLaMA for language understanding.

LLaVA: <https://llava-vl.github.io/>



Efficient Vision-Language Models

AWQ quantizes vision-language models to 4 bits with high quality



Question: Do you know who drew this painting?

INT4-g128 RTN: The painting depicts a woman, who some believe to be a representation of the famous artist, Mona Lisa. However, it is not certain that the woman in the painting is indeed Mona Lisa. The artwork is a popular subject of artistic analysis and admiration, often considered a symbol of feminine beauty and enigma. The portrait is characterized by its intricate details, captivating the viewer with the woman's expressive gaze and the subtle, enigmatic smile that has become synonymous with the Mona Lisa.

INT4-g128 AWQ: The painting depicts a woman, who is believed to be a portrait of Mona Lisa, the famous artwork by the artist **Leonardo da Vinci**. Although the painting is widely recognized as a representation of Mona Lisa, it is not possible to confirm the artist who drew this particular painting without additional information. Leonardo da Vinci was a renowned artist known for his exceptional skill in painting detailed, realistic portraits, and the Mona Lisa is one of his most famous works.

Sometimes I just look at pictures of the earth from space and I marvel at how beautiful it all is.



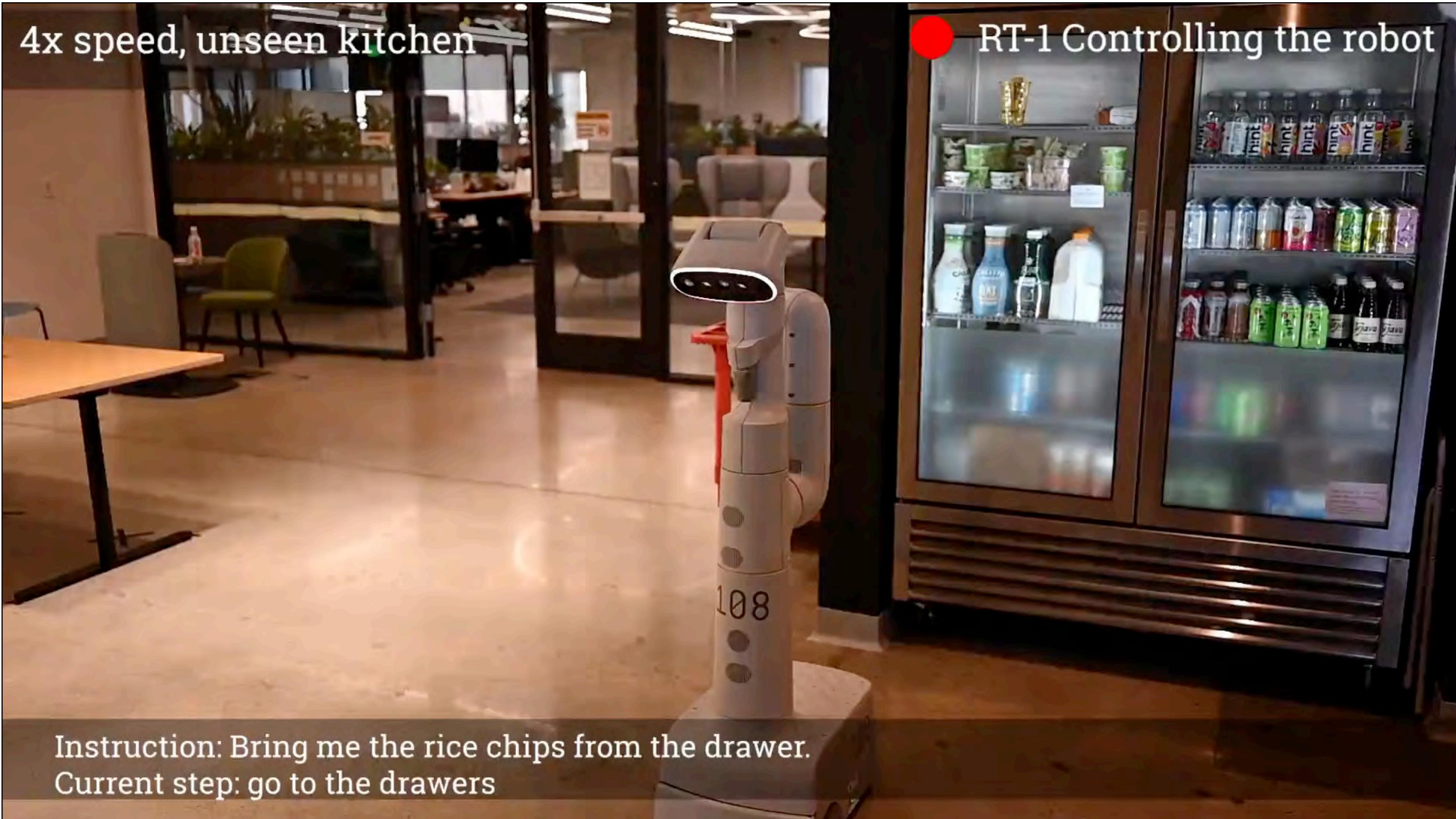
Question: Can you explain this meme in detail?

INT4-g128 RTN: The meme in the image features a plate of food, likely fried chicken, with a unique twist. **There are small pictures of the Earth and other planets placed on top of the food**, creating an interesting and amusing visual. Alongside the plate, a caption reads, "Sometimes I look at pictures of the Earth from space and I marvel at how beautiful it all is." This combination of food and planet images adds a touch of humor and curiosity, making the viewer wonder about the context and intent behind the meme.

INT4-g128 AWQ: The meme in the image is a **lighthearted and humorous take on the concept of looking at pictures of the Earth from space**. A plate of fried food, specifically chicken nuggets, is presented with the caption, "Sometimes I look at pictures of the Earth from space and I marvel at how beautiful it all is." The playful combination of a food item and an Earth-related context creates a fun and entertaining visual.

Vision-Language-Action Models

Robotics transformers control robots based on language instructions



Run at only **3Hz** due to the high computational cost and networking latency.

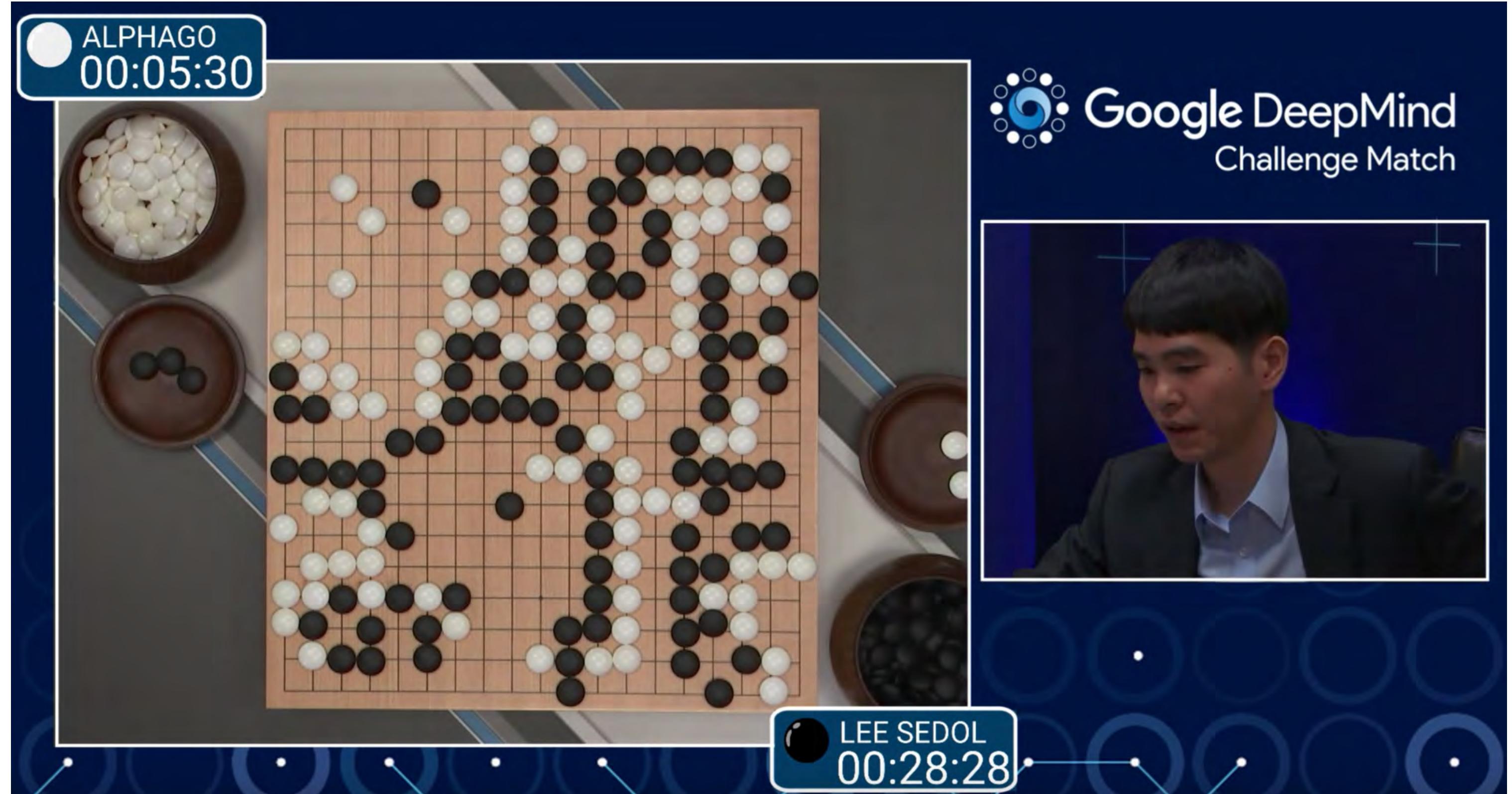
RT-1: <https://robotics-transformer1.github.io/>

Deep Learning for Games

AlphaGo masters the game of Go with DNNs & tree search



AlphaGo (Nature 2016)



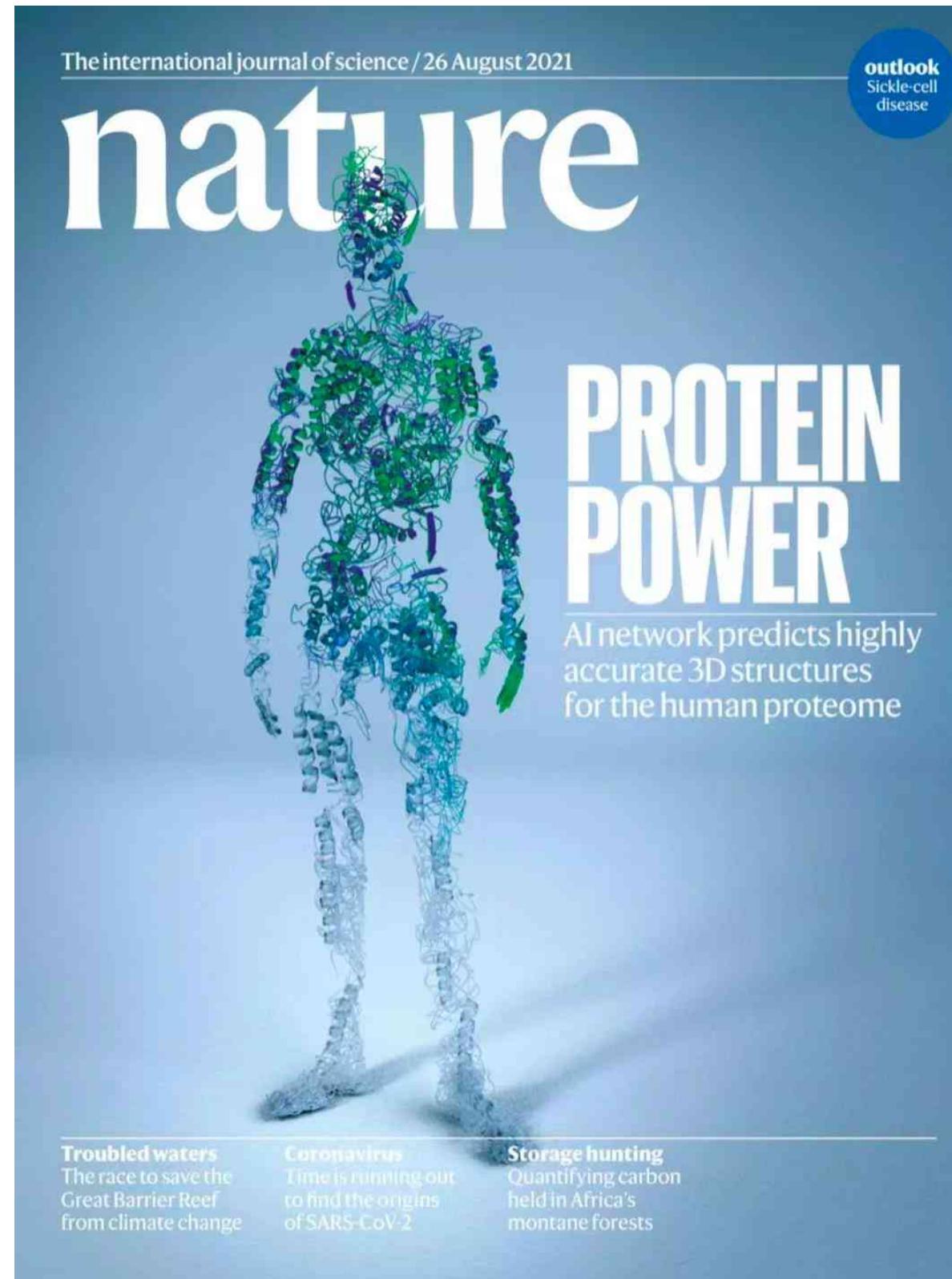
AlphaGo versus Lee Sedol (4-1)

Compute: 1920 CPUs and 280 GPUs (**\$3000 electric bill per game**)

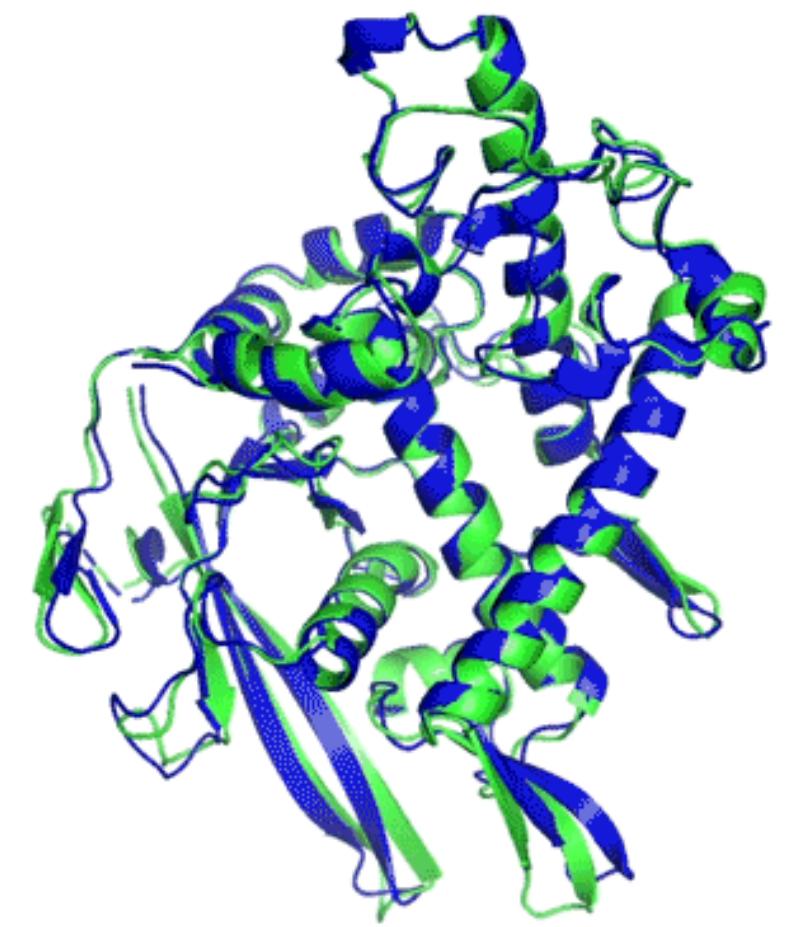
AlphaGo: <https://www.deepmind.com/research/highlighted-research/alphago>

Deep Learning for Scientific Discovery

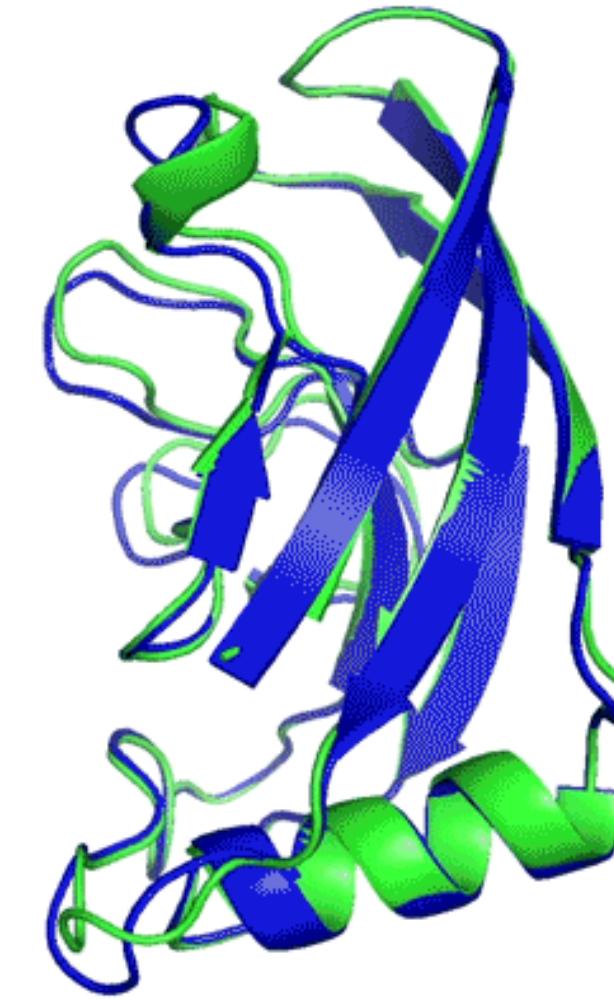
AlphaFold reveals the structure of the protein universe



AlphaFold (Nature 2021)



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)



T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction

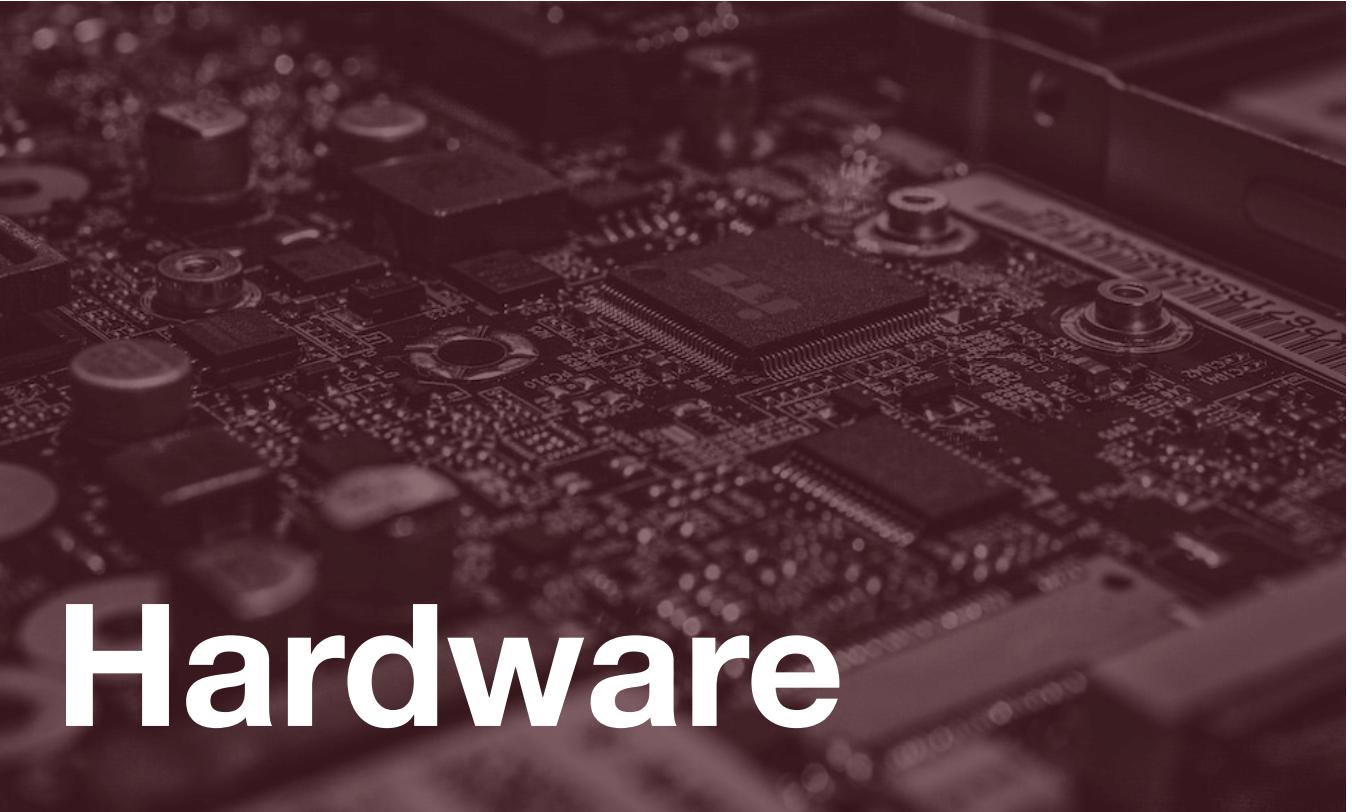
Compute: 16 TPUv3s (128 TPUv3 cores) for a few weeks

AlphaFold 2: <https://www.deepmind.com/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>

Deep Learning: Three Pillars

```
19    temp = 0;
20    unsigned int len1 = s1.size(), len2 = s2.size();
21    const size_t len1 = col(len2+1), prevCol(len2);
22    vector<unsigned int> col(len2+1), prevCol(len2);
23    for (unsigned int i = 0; i < prevCol.size(); i++) {
24        prevCol[i] = i;
25        for (unsigned int i = 0; i < len1; i++) {
26            col[0] = i+1;
27            for (unsigned int j = 0; j < len2; j++) {
28                col[j+1] = std::min( std::min( prevCol[j] + (s1[i]==s2[j]) ? 1 : 0,
29                                            prevCol[j] );
30                swap(prevCol);
31            }
32        }
33    }
34    return col;
35 }
```

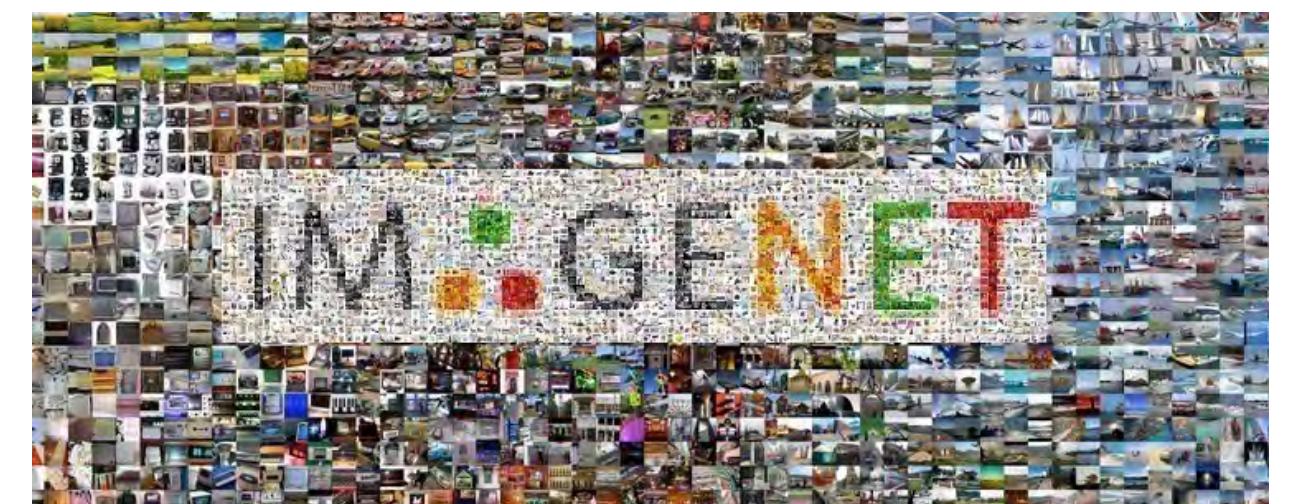
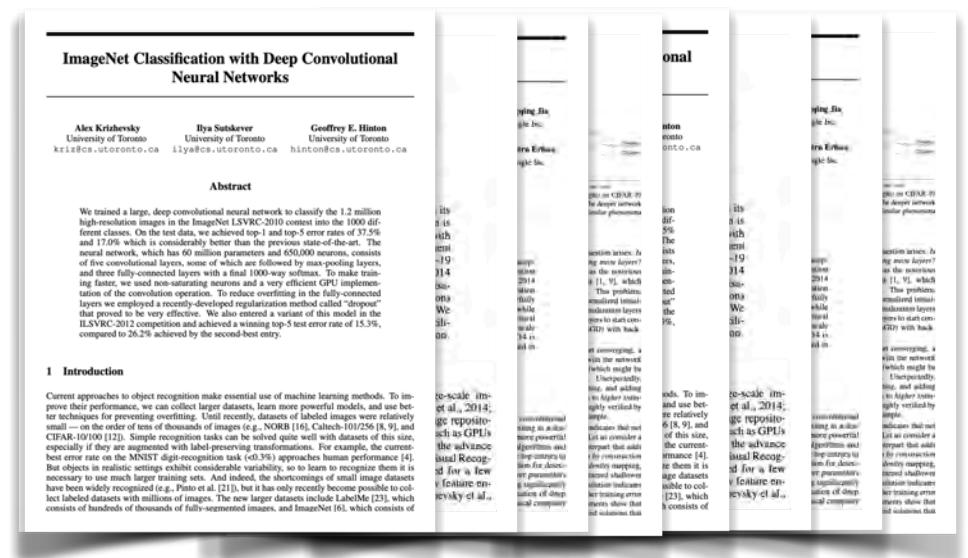
Algorithm



Hardware

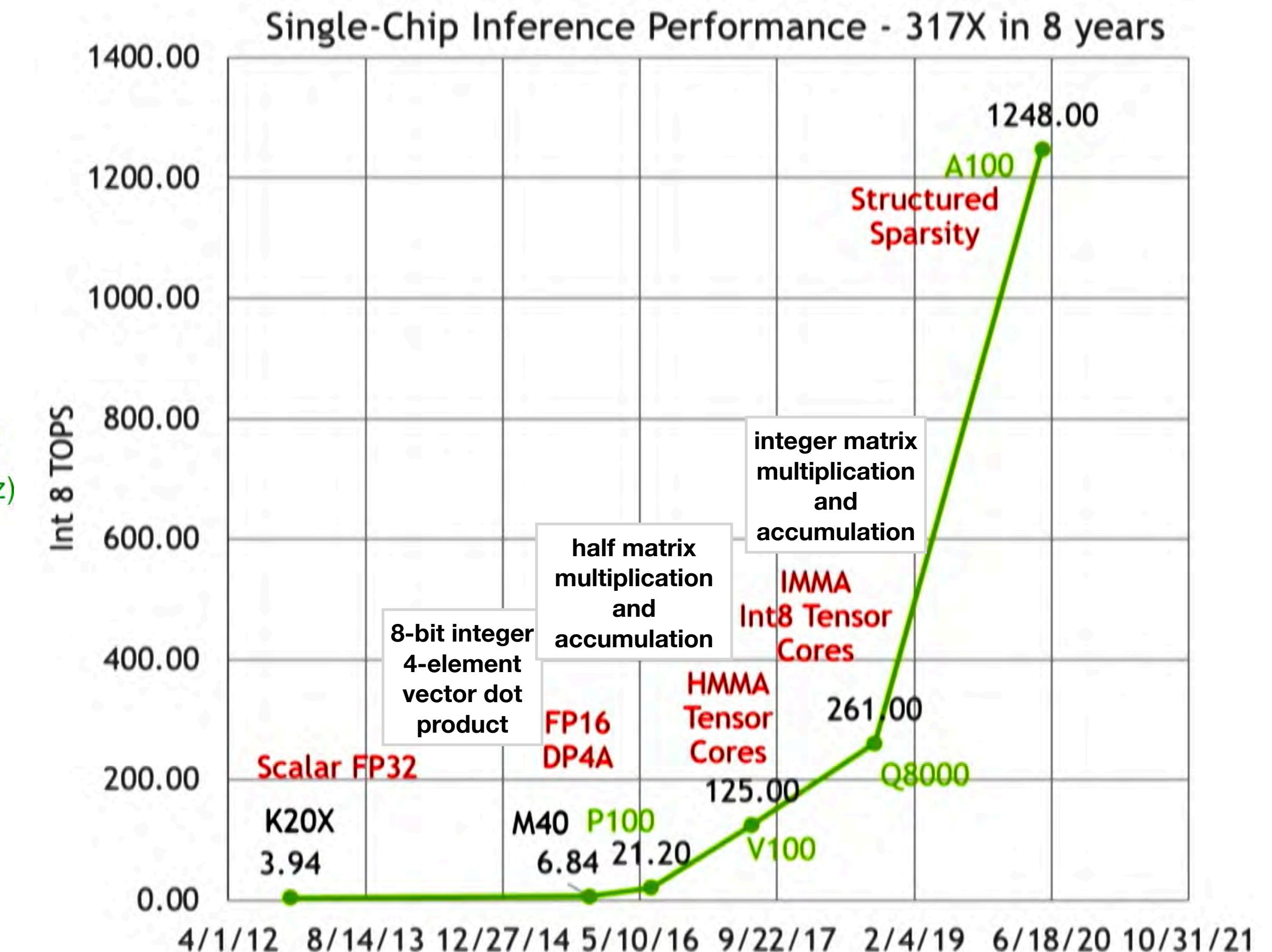
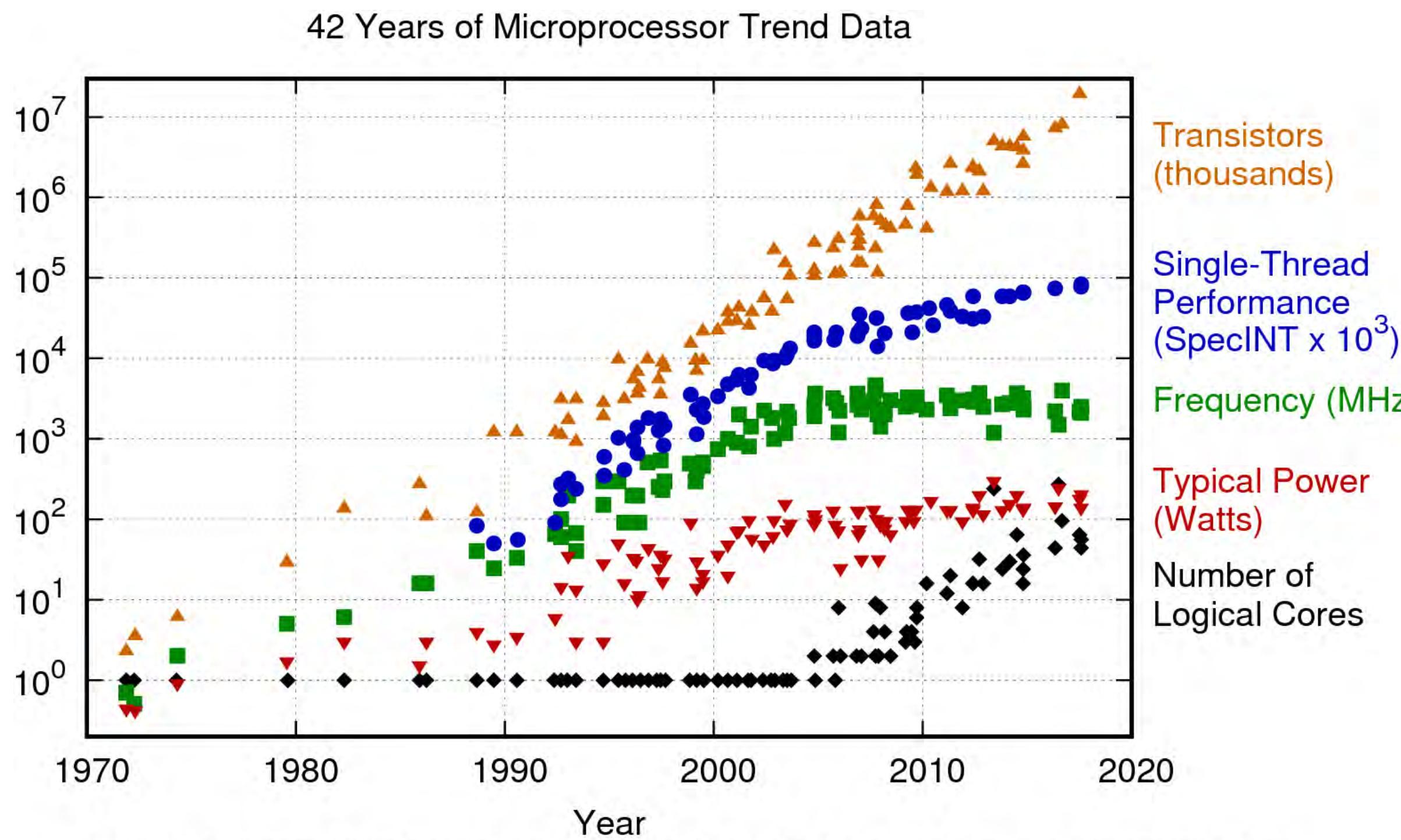


Data



Architectural support for quantization/pruning brings tremendous improvement

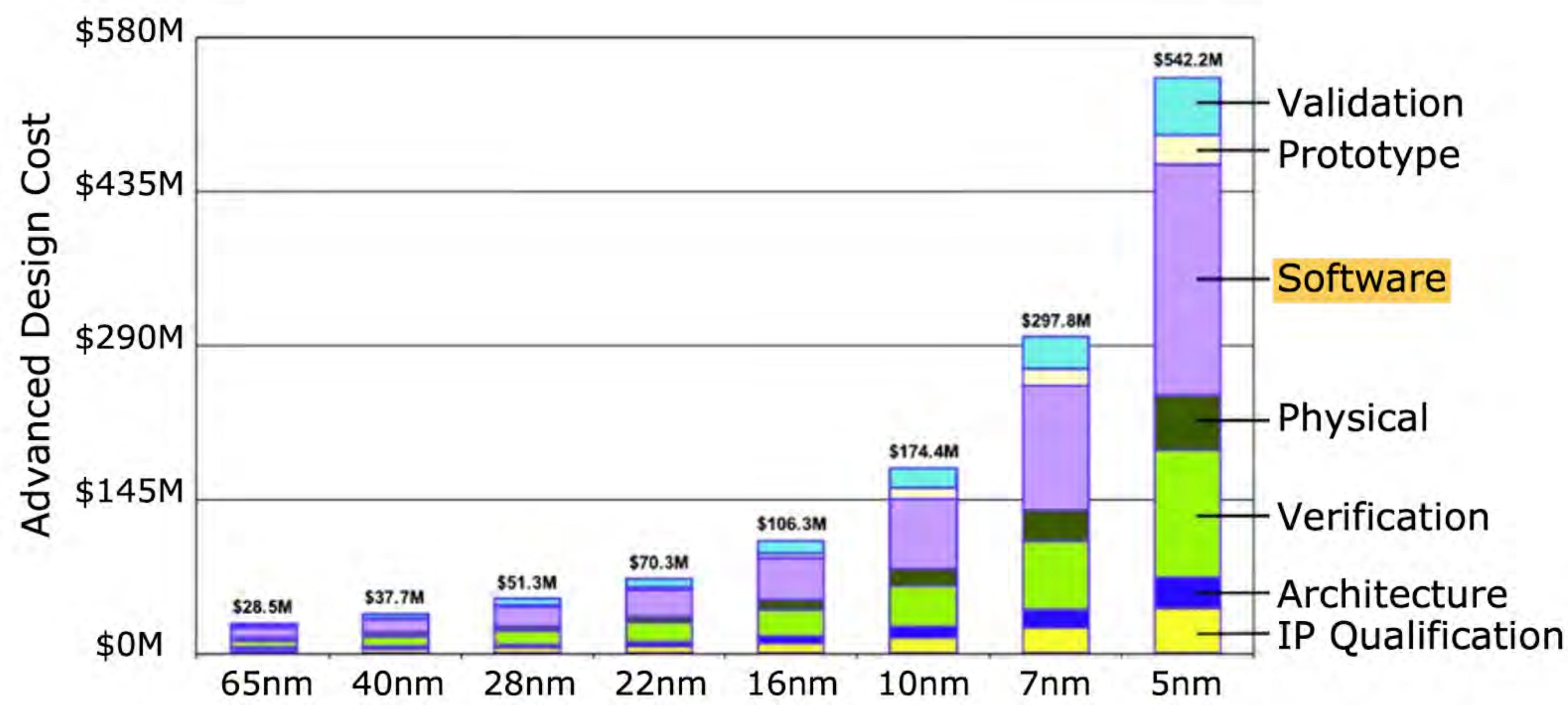
FP32 => FP16 => Int8; dense => sparse



Data source: M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten, K. Rupp

GPUs, Machine Learning, and EDA — Bill Dally

Software Innovation is important in advanced technology node



The software cost dominates the cost breakdown of advanced technology nodes [source].
We focus on designing new algorithms and software for efficient computing.

Cloud AI Hardware

NVIDIA



P100 (2016)



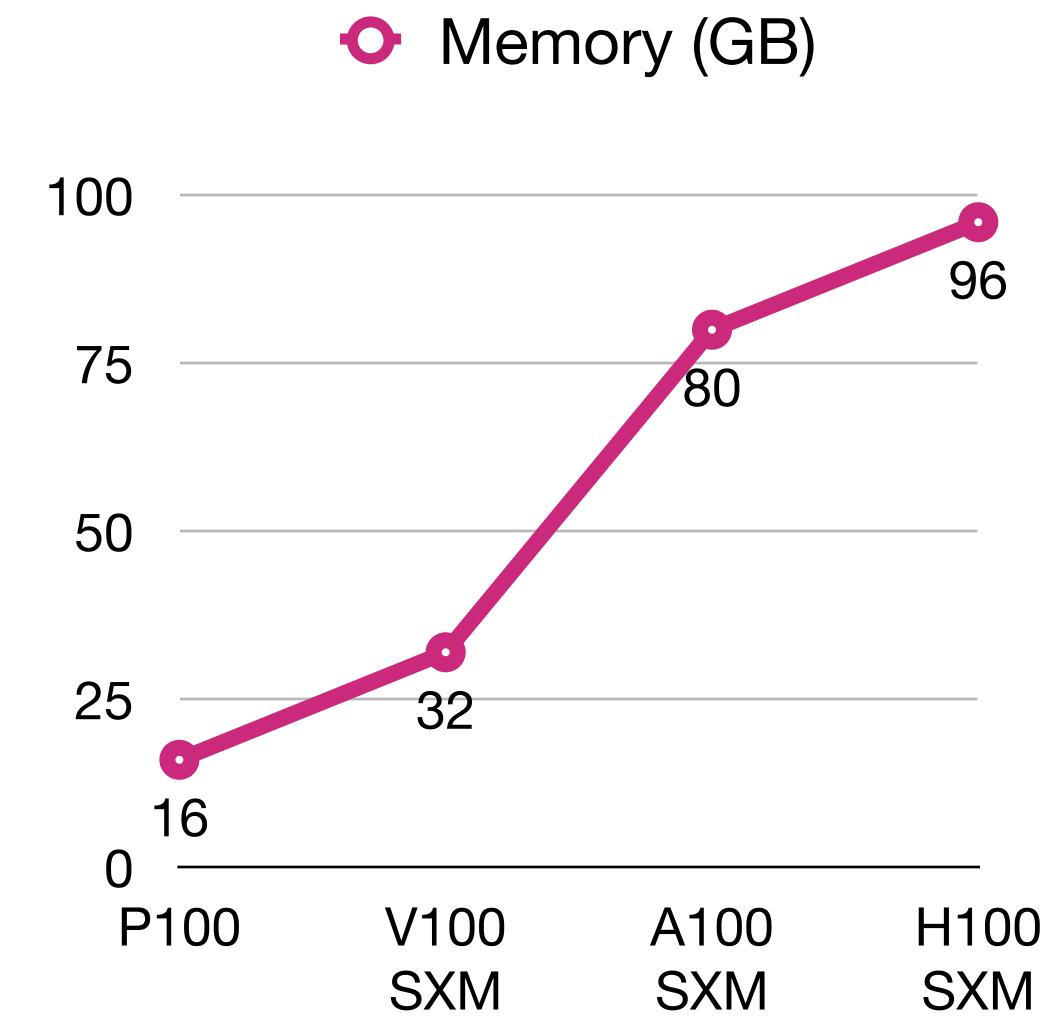
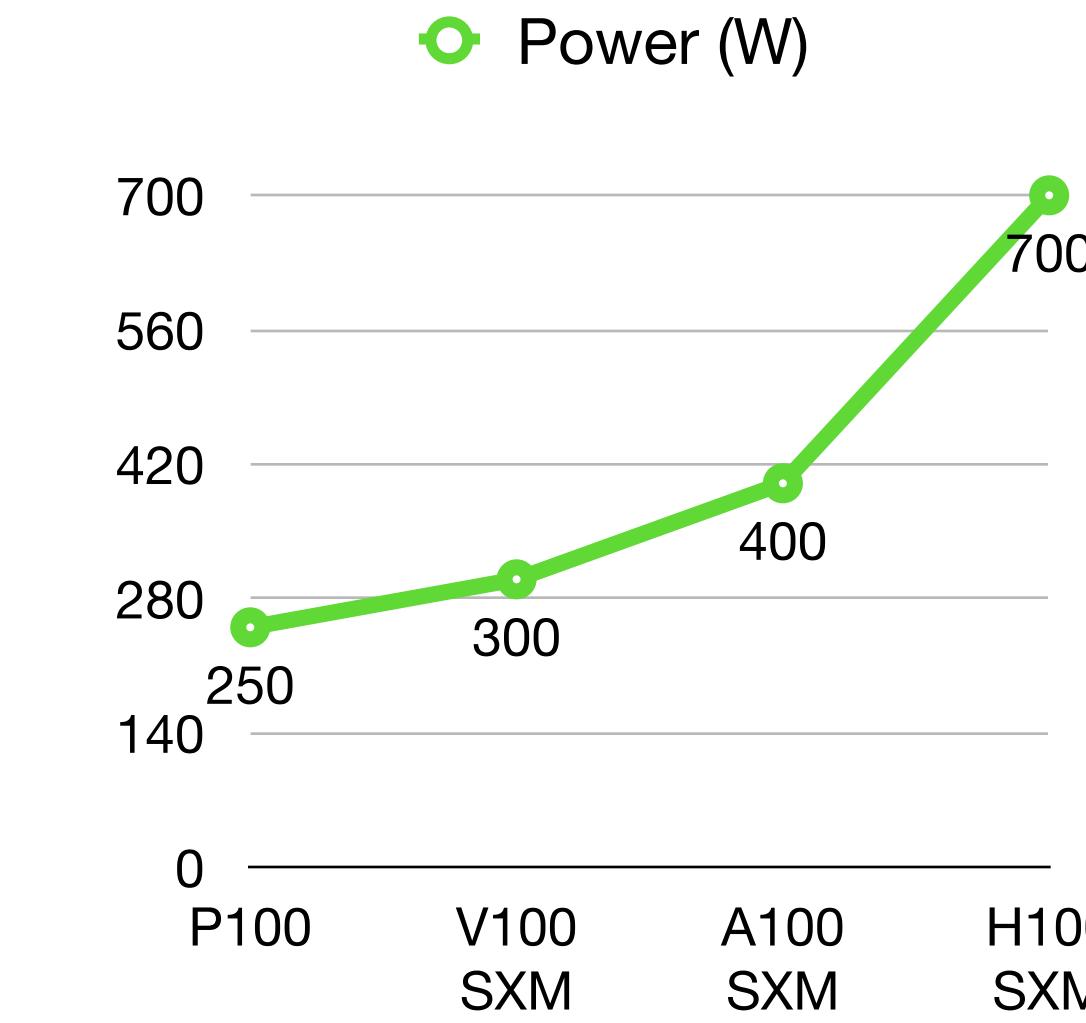
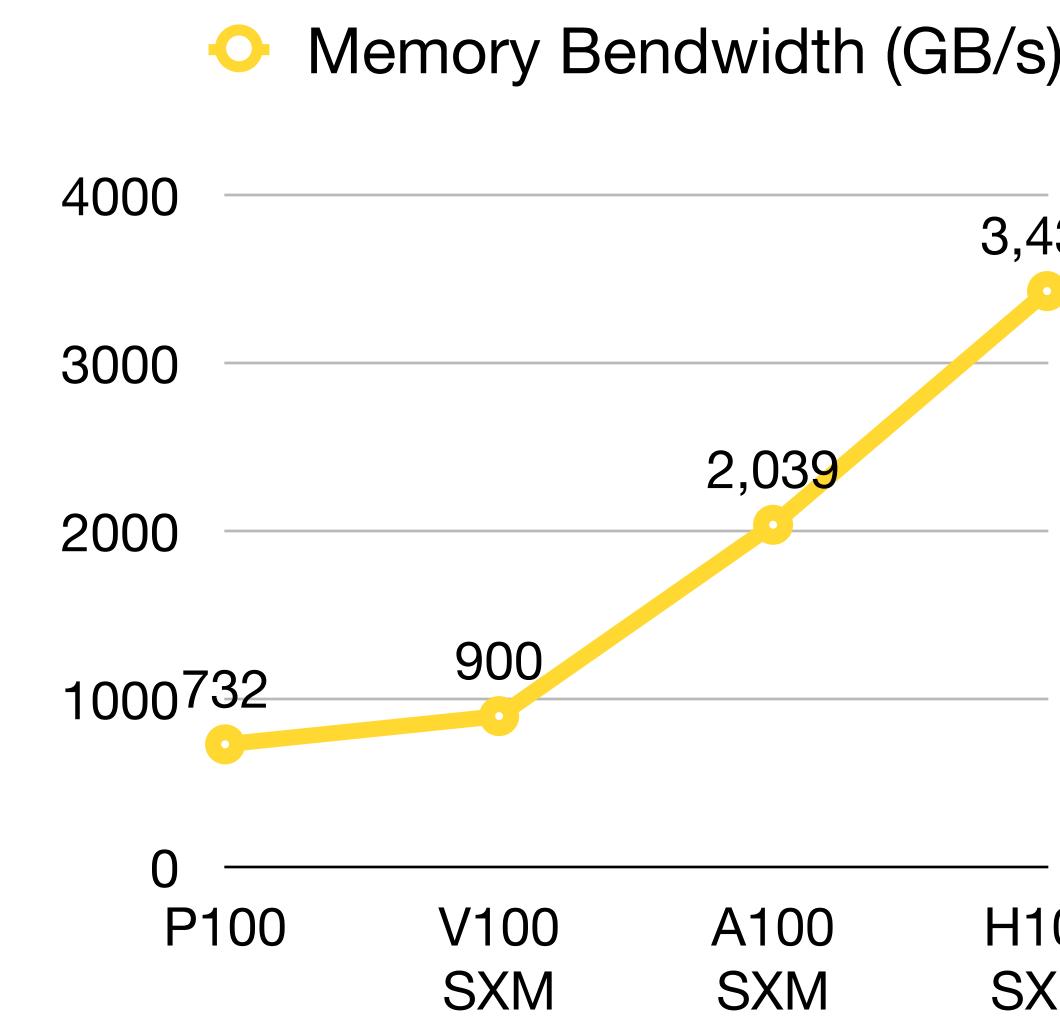
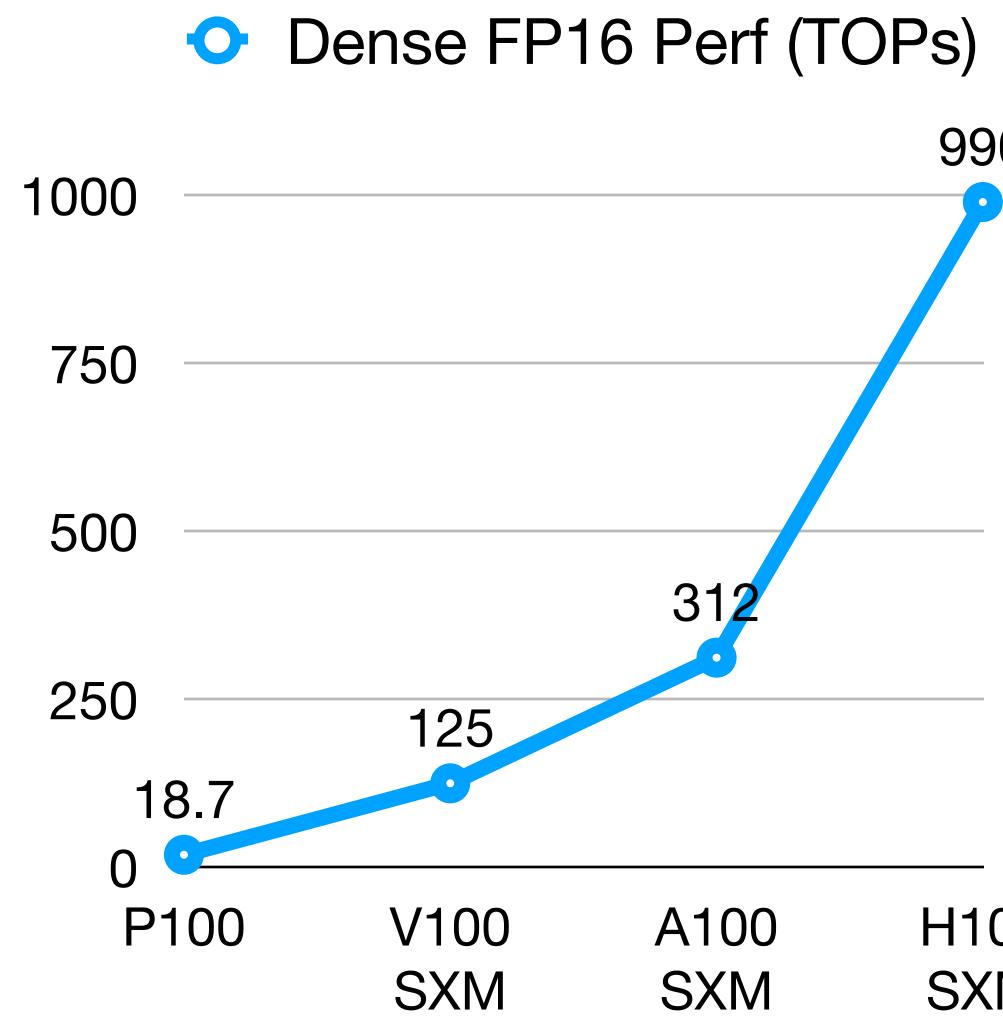
V100 (2017)



A100 (2020)



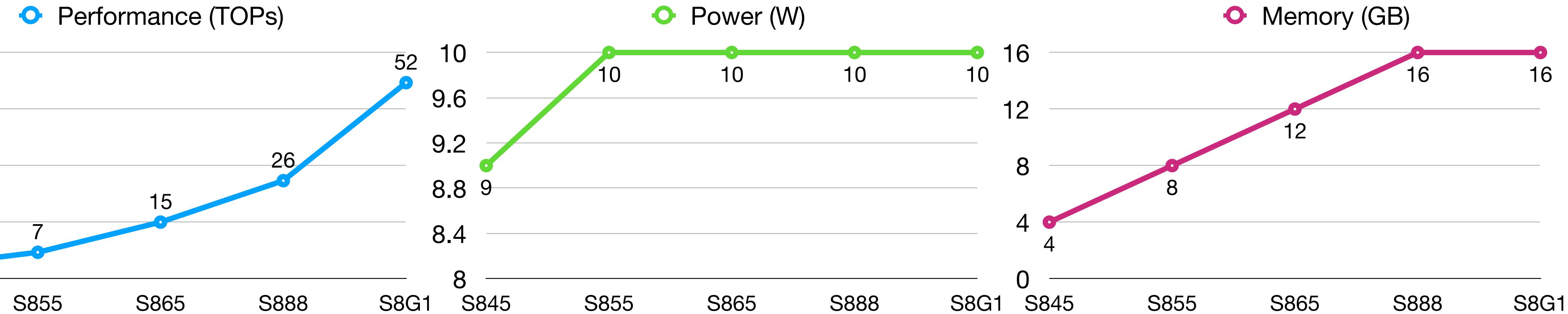
H100 (2022)



Edge AI Hardware

Qualcomm Hexagon DSP

- Qualcomm Hexagon is a family of digital signal processor (DSP) products by Qualcomm. It is designed to deliver performance with low power over a variety of applications

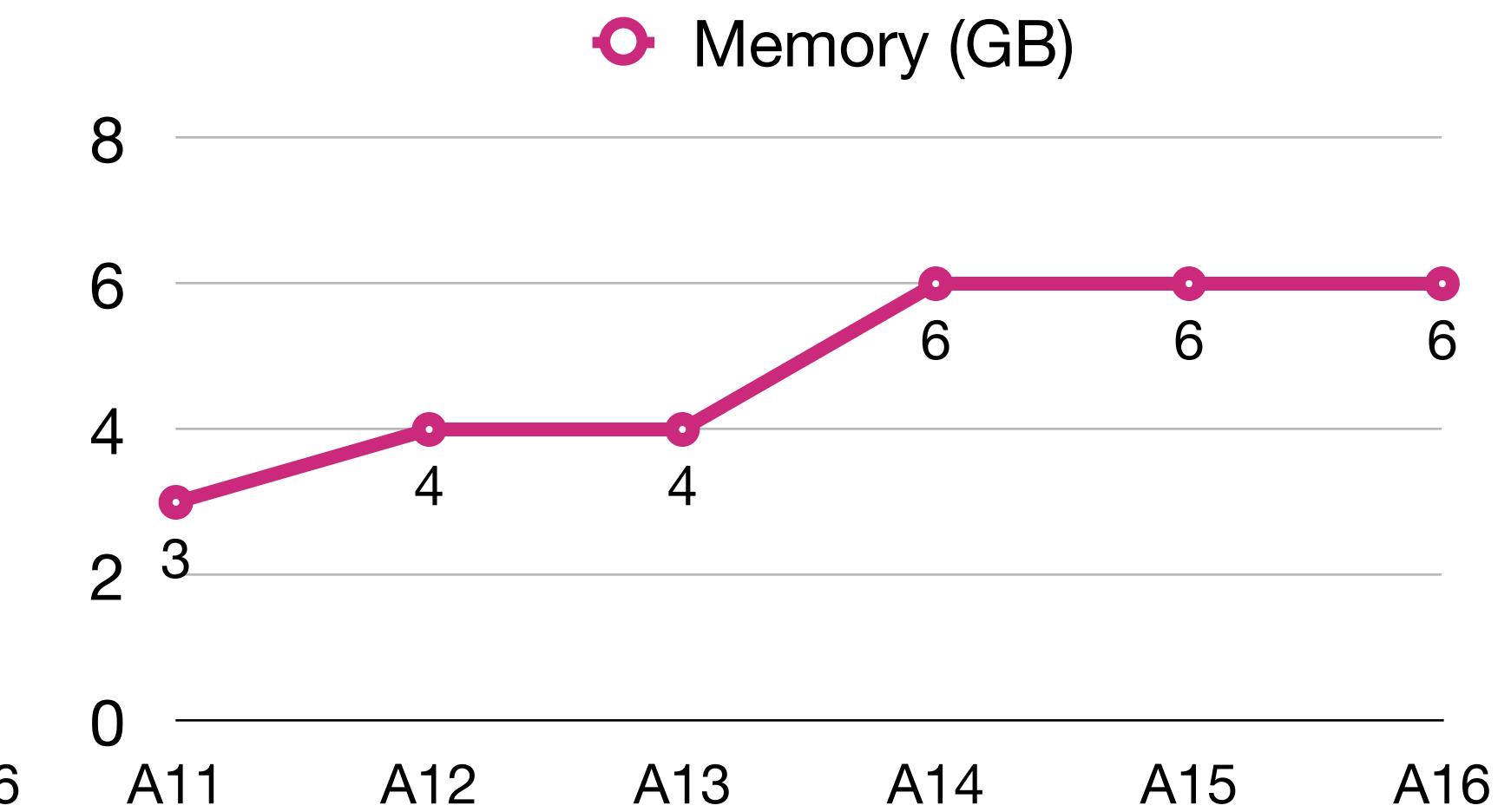
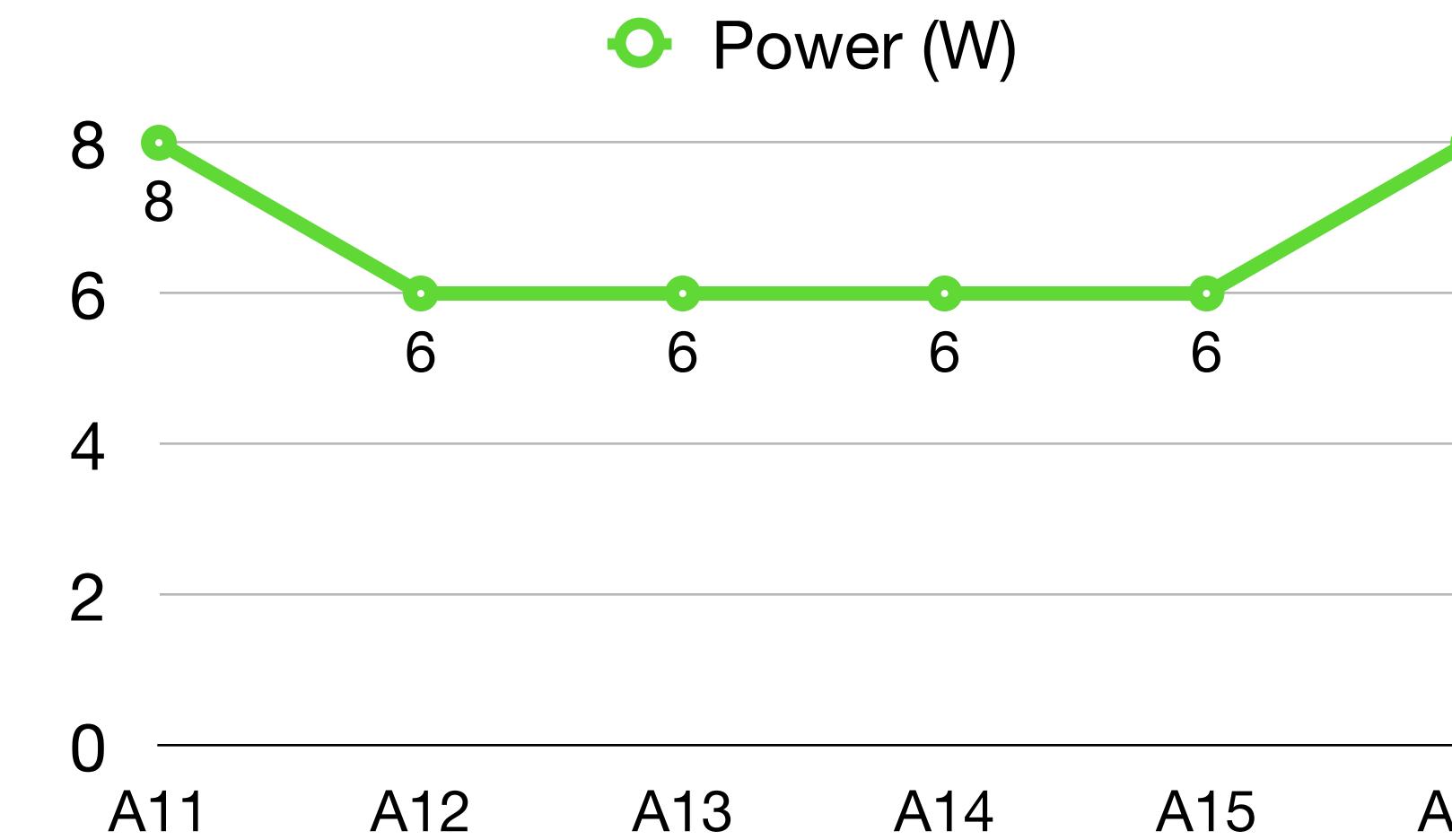
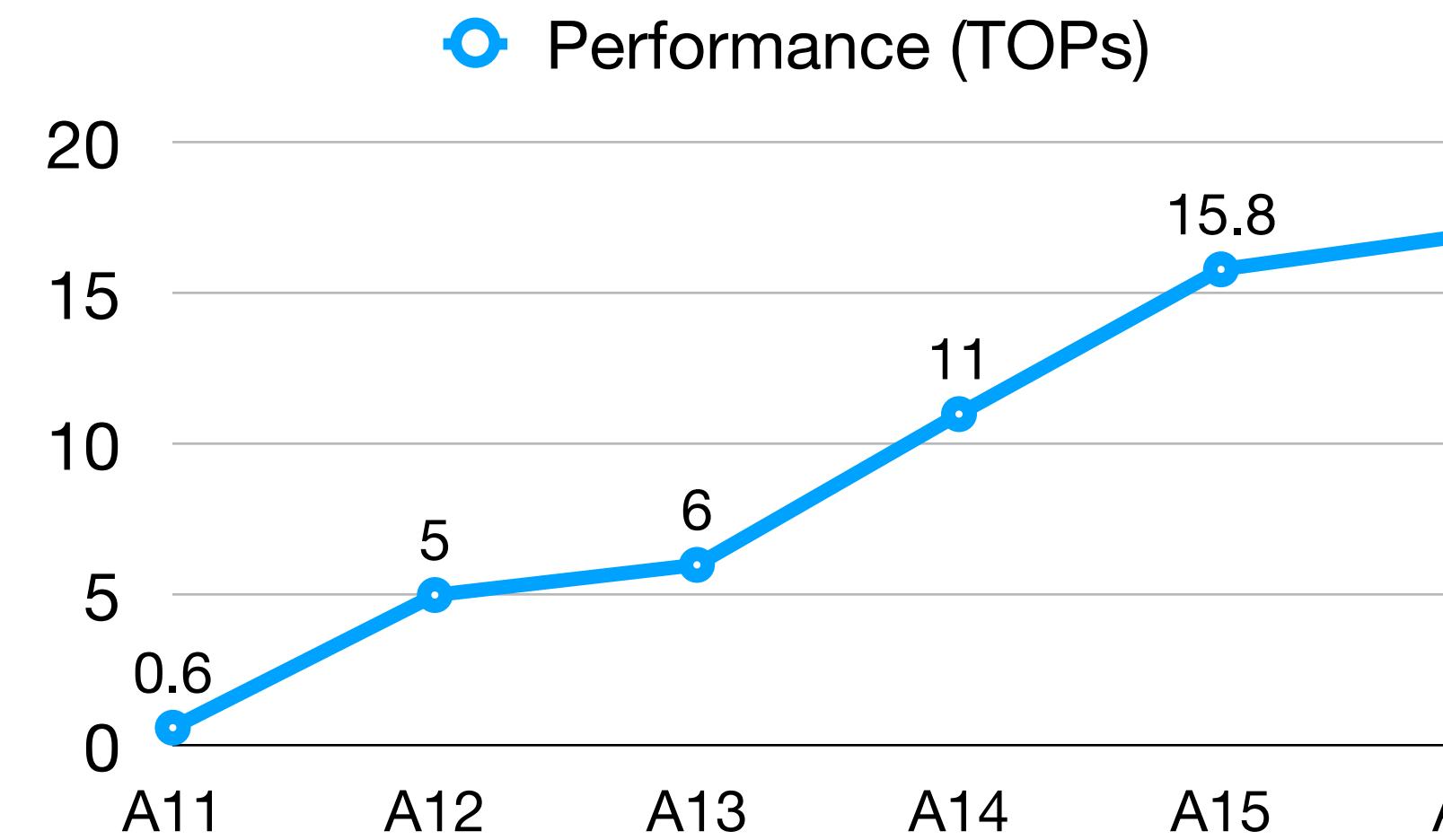


https://en.wikipedia.org/wiki/List_of_Qualcomm_Snapdragon_processors

Edge AI Hardware

Apple Neural Engine

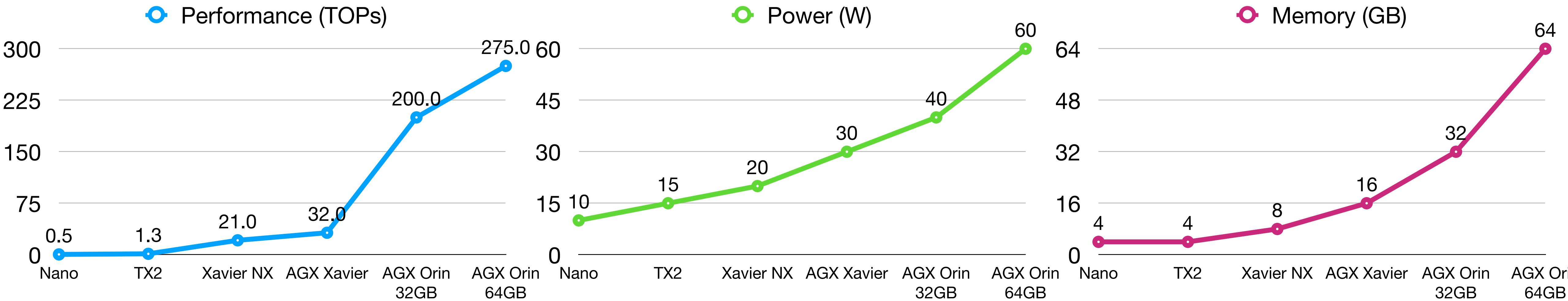
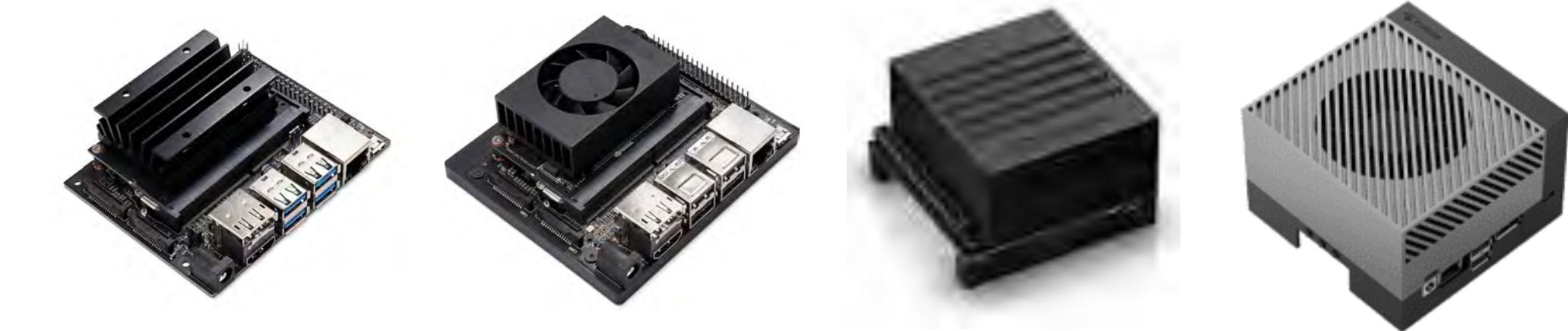
- The Apple Neural Engine (ANE) is an energy-efficient and high-throughput engine for ML inference on Apple silicon.



Edge AI Hardware

Nvidia Jetson

- NVIDIA Jetson is a complete System on Module (SOM) that includes a GPU, CPU, memory, power management, high-speed interfaces, and more.

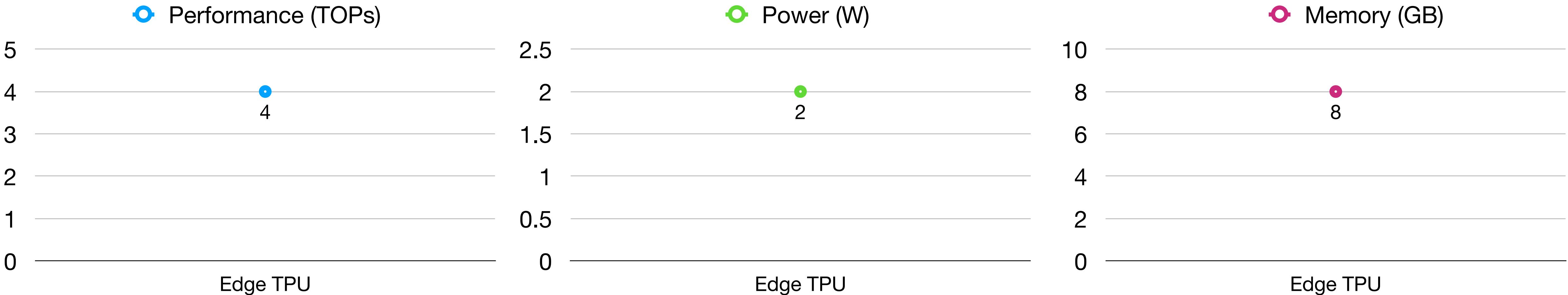


NanoReview. <https://connecttech.com/jetson/jetson-module-comparison/>

Edge AI Hardware

Tensor Processing Unit

- Tensor Processing Unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google for neural network machine learning, using Google's own TensorFlow software.

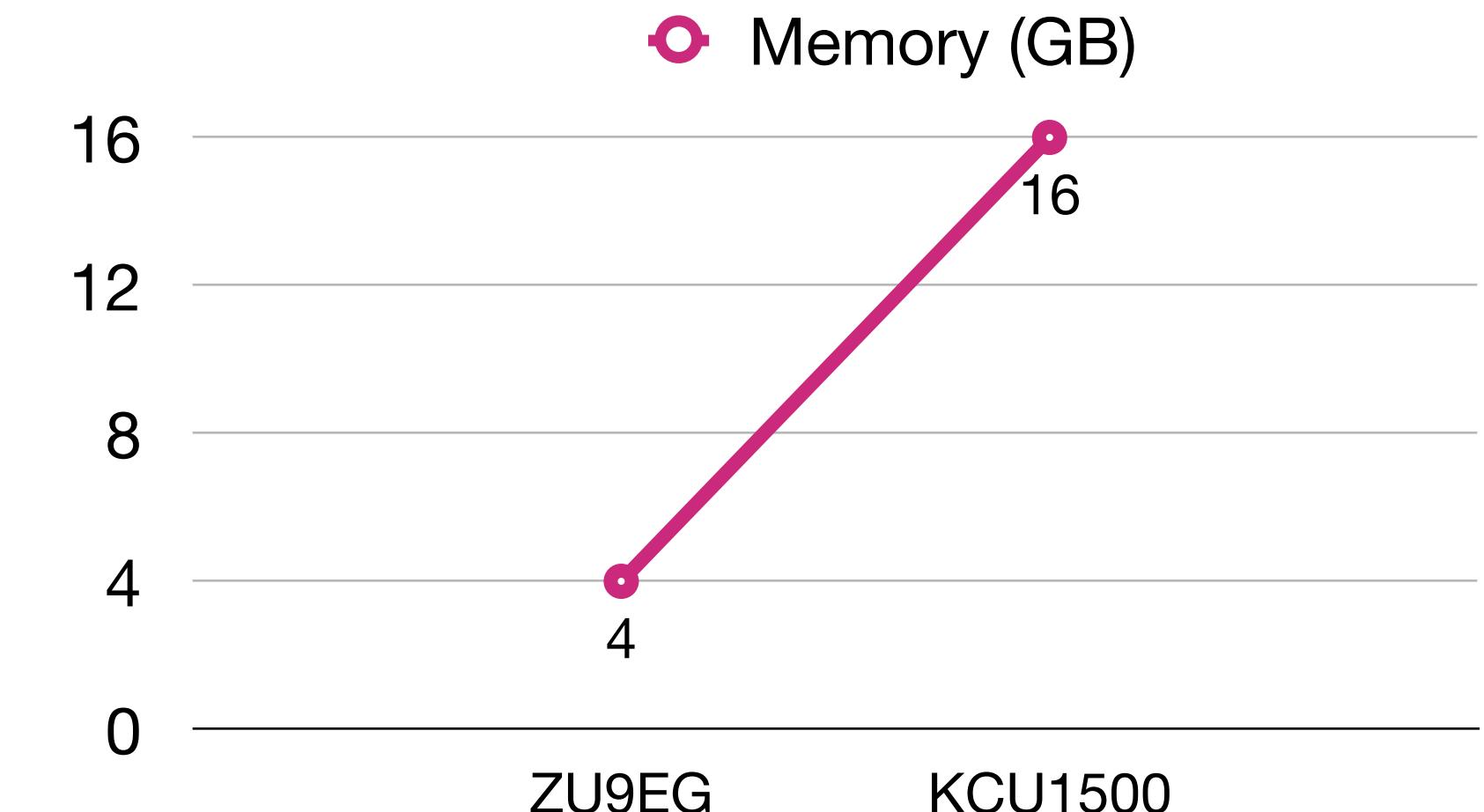
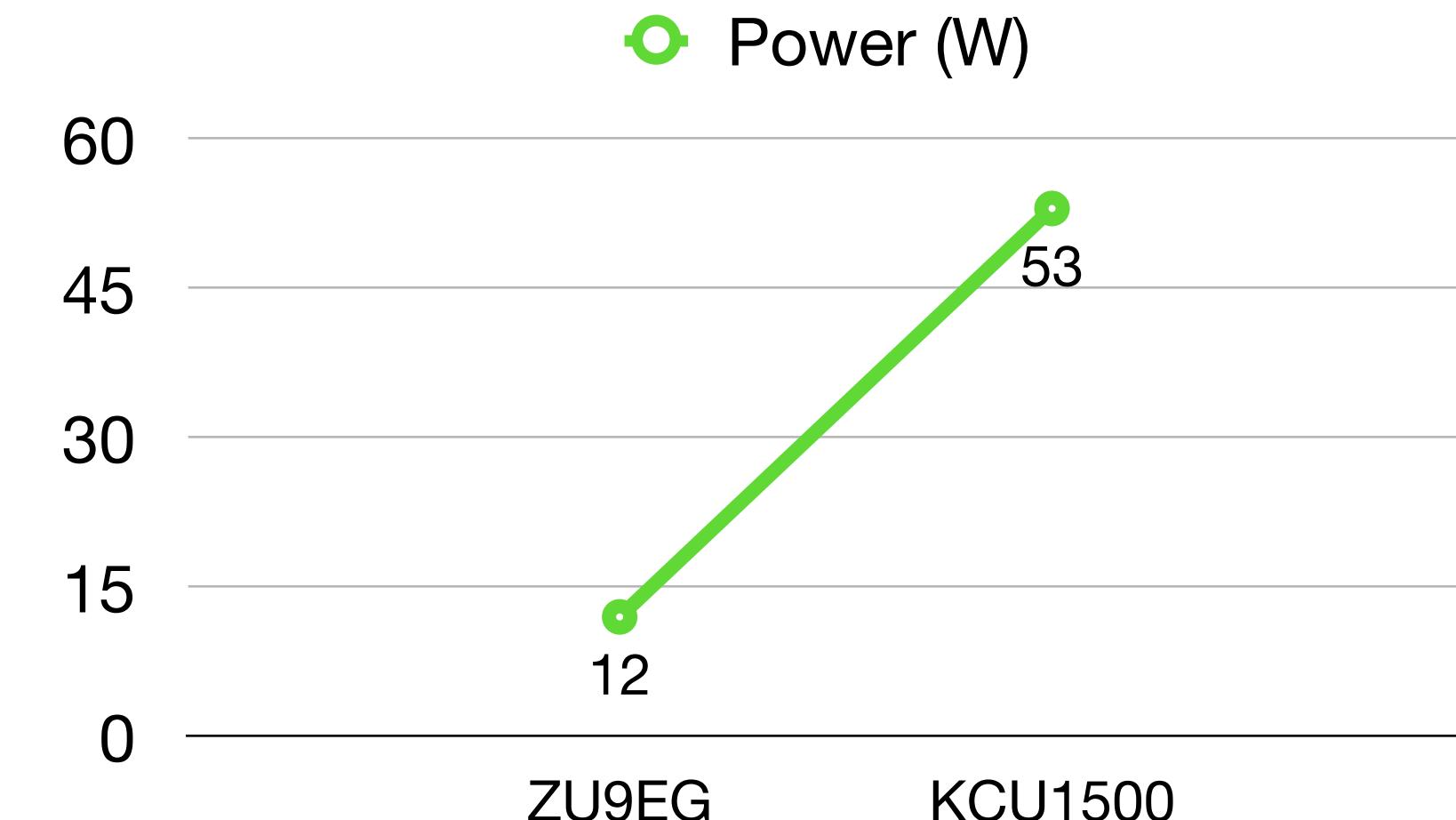
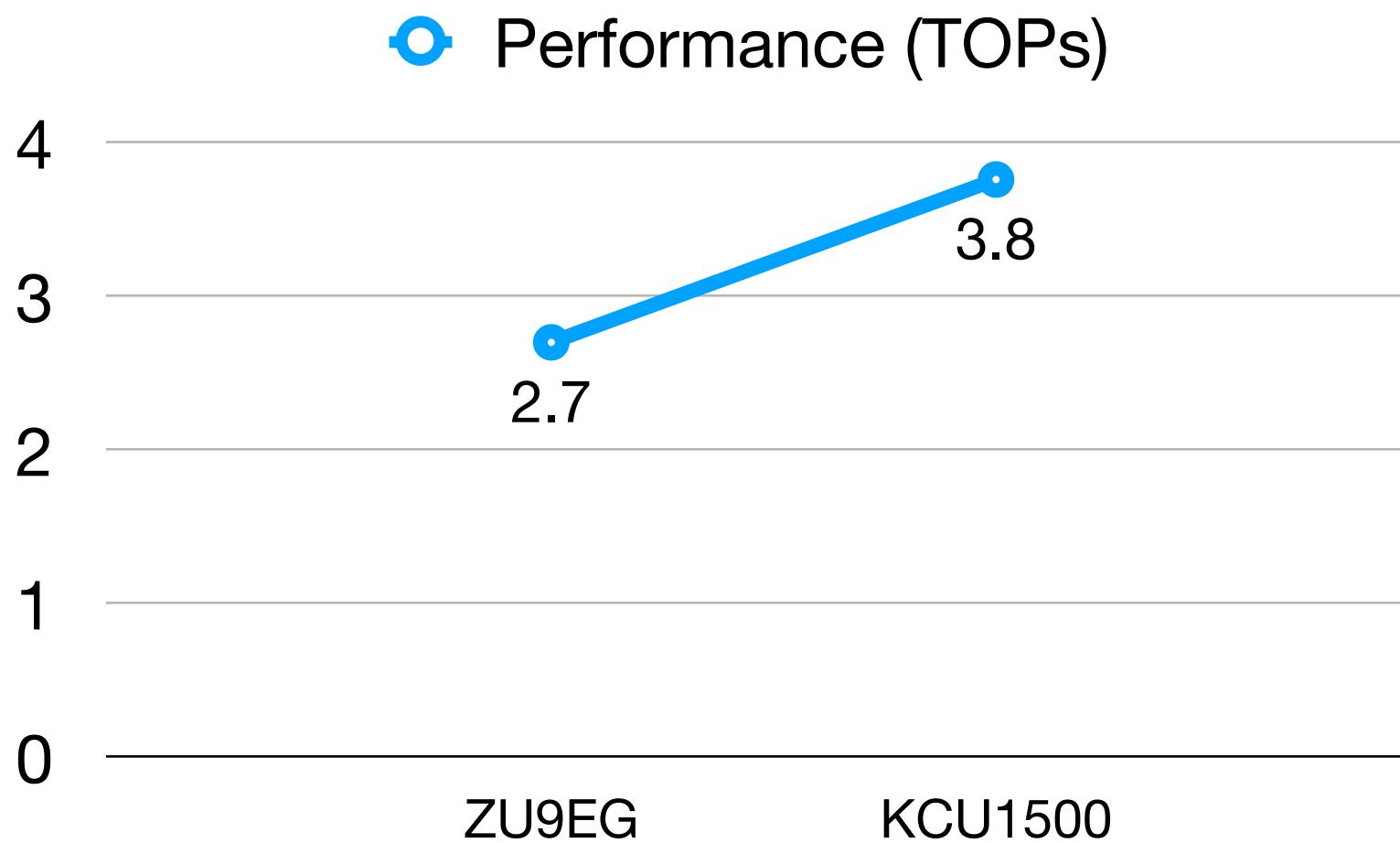


Tensor Processing Unit. https://en.wikipedia.org/wiki/Tensor_Processing_Unit

Edge AI Hardware

FPGA-based Accelerators

- Field Programmable Gate Arrays (FPGA) delivers higher performance compared to a fixed-architecture AI accelerator like a GPU due to efficiency of custom hardware acceleration.

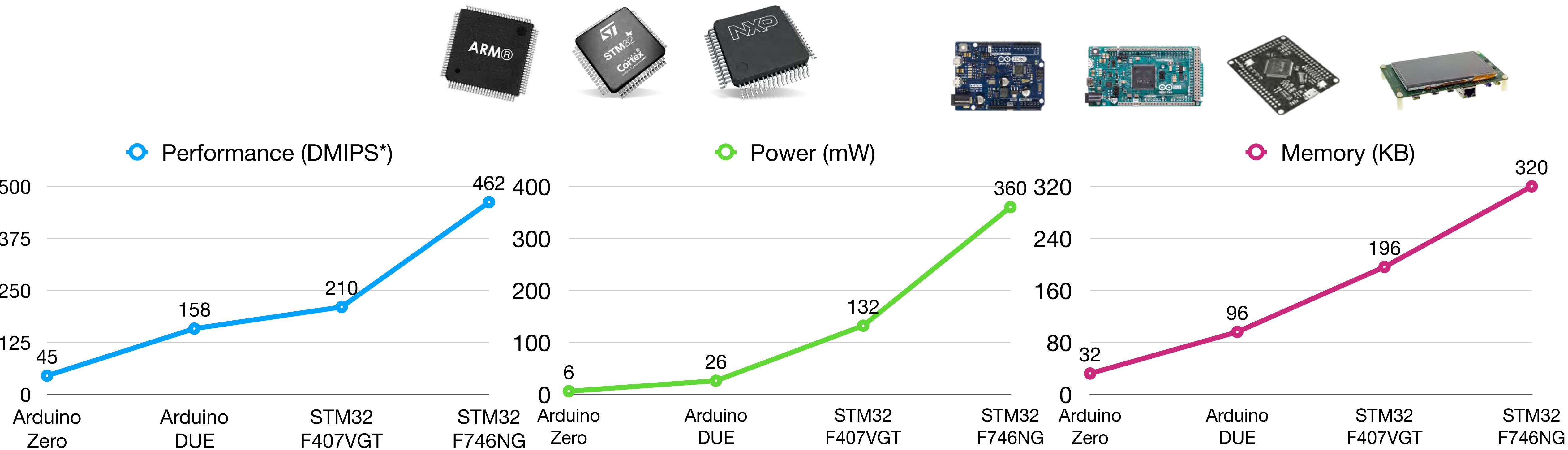


Neural Network Accelerator Comparison. <https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator.html>

Edge AI Hardware

Microcontrollers (MCU)

- A microcontroller is a compact integrated circuit designed for embedded systems. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.



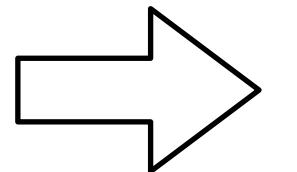
* Dhrystone Million Instructions Per Second (DMIPs) is an index for integer computation.

Edge AI Hardware

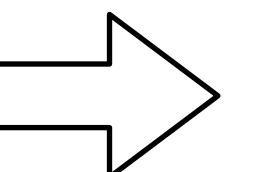
Edge AI devices still have huge gap to cloud processors



Cloud AI



Mobile AI



Tiny AI

Memory (Activation)

80GB

4GB

320kB

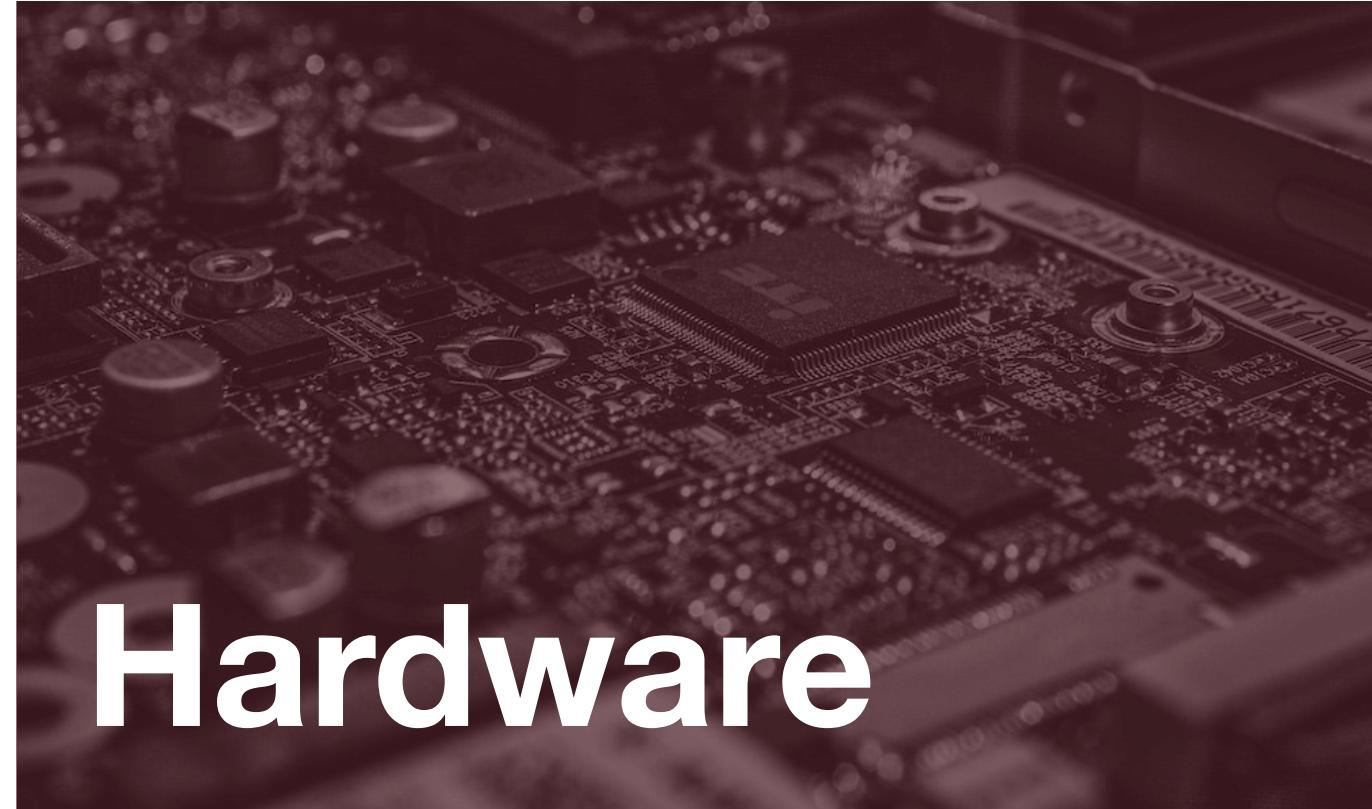
Storage (Weights)

~TB/PB

256GB

1MB

Deep Learning: Three Pillars



Hardware



Data



Algorithm

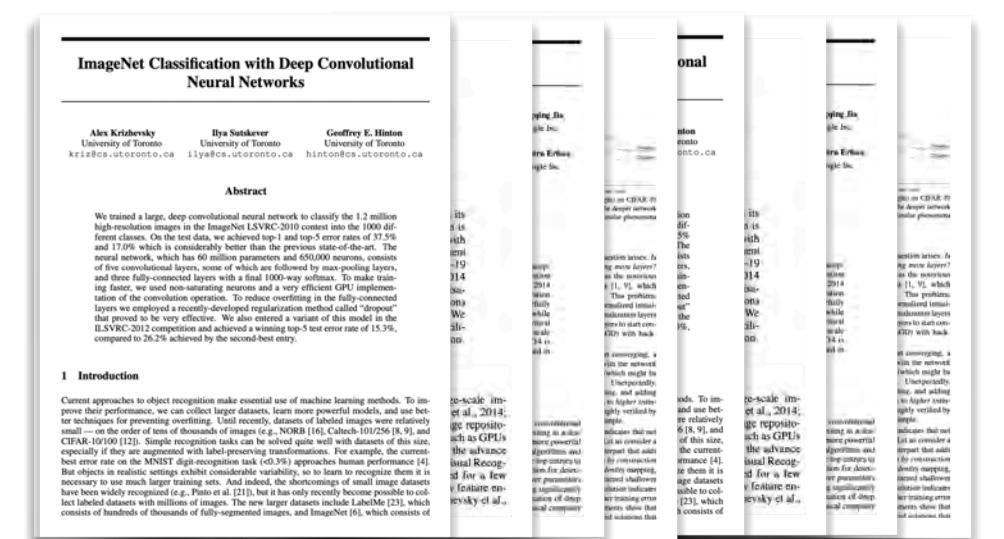
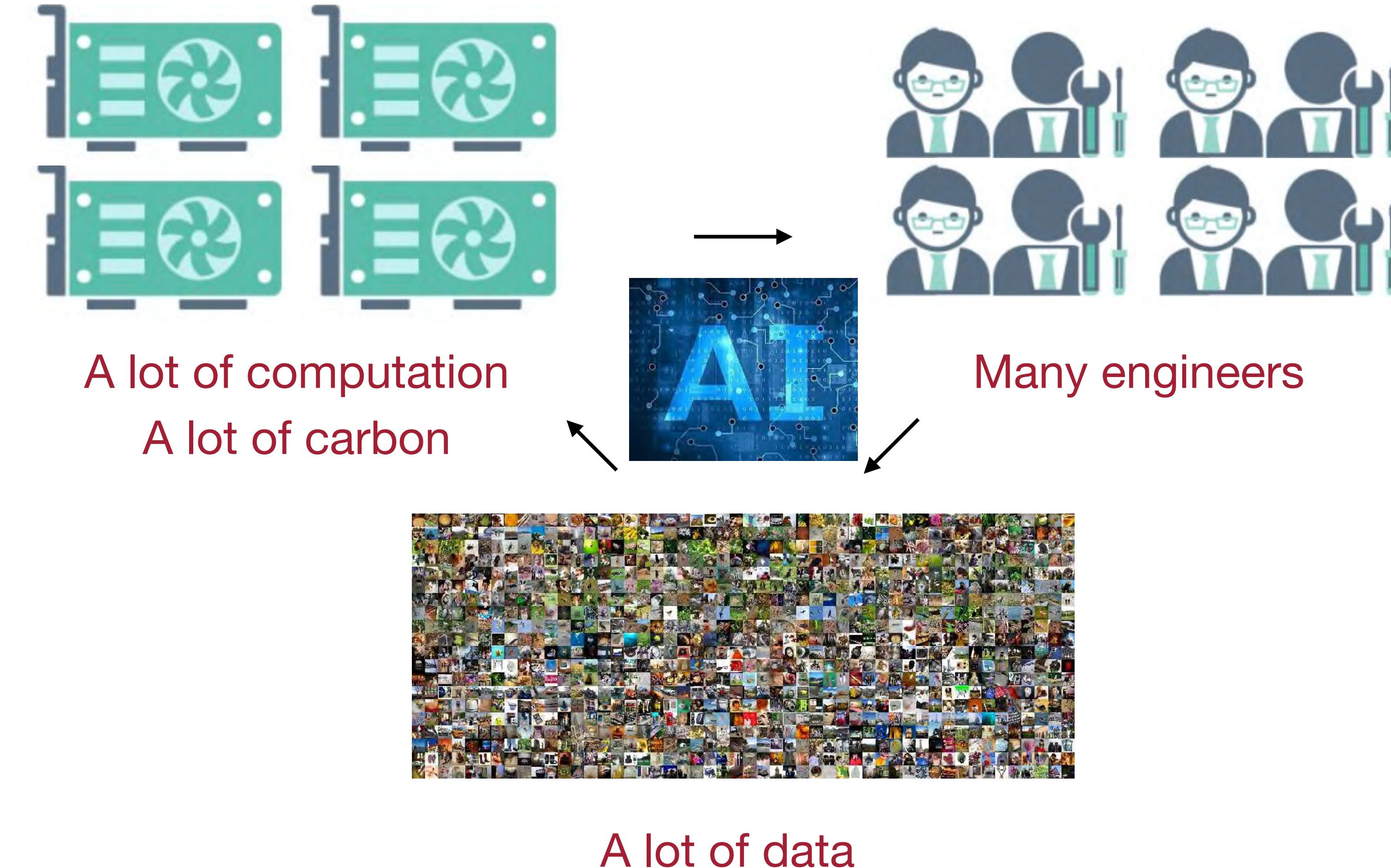


Image source: [1](#), [2](#), [3](#)

Current Landscape of Deep Learning

Big computation, engineer and data



Tiny Deep Learning

Tiny computation, engineer and data



less computation
less carbon

TinyML



fewer engineers



less data

Course Overview

Course Overview

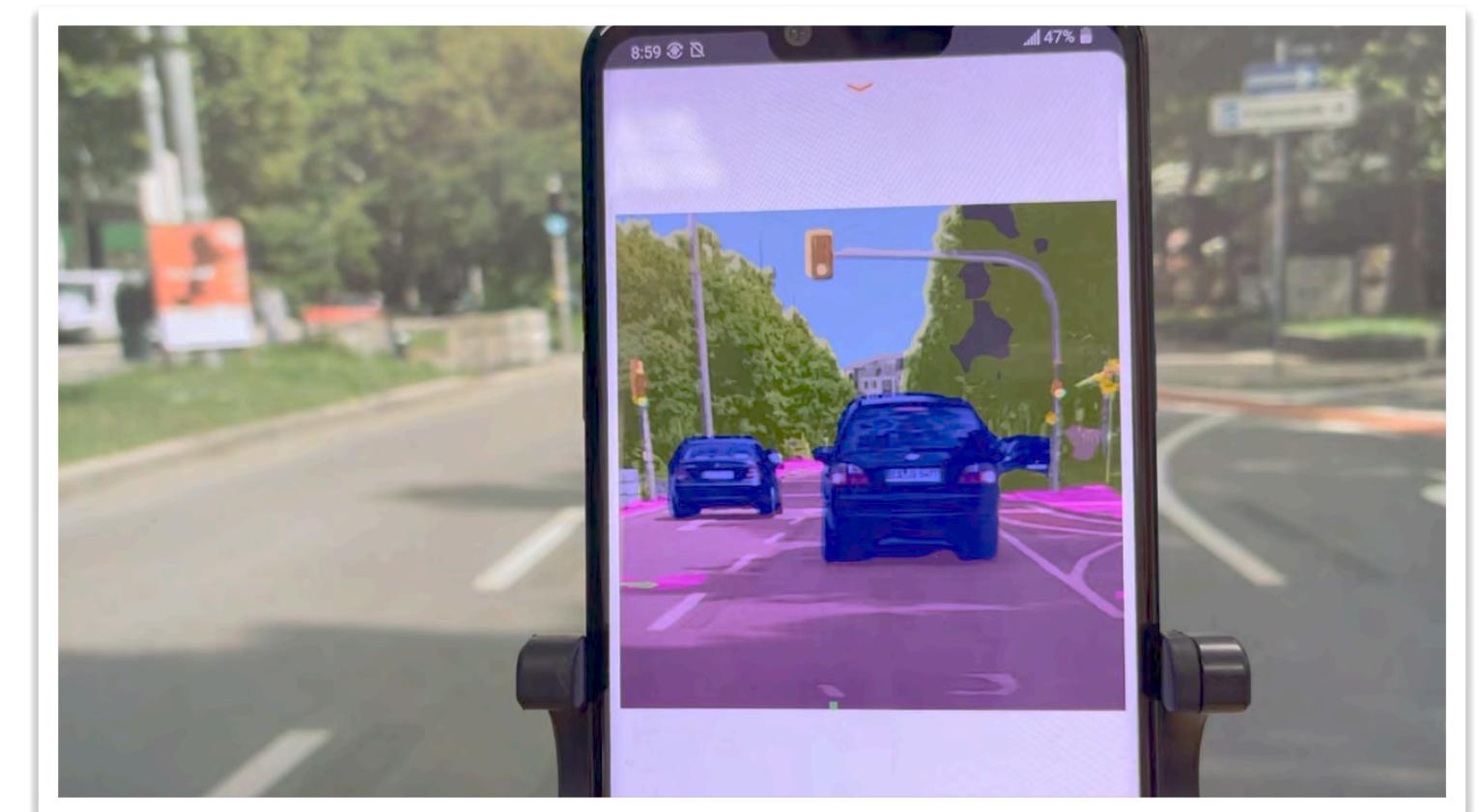
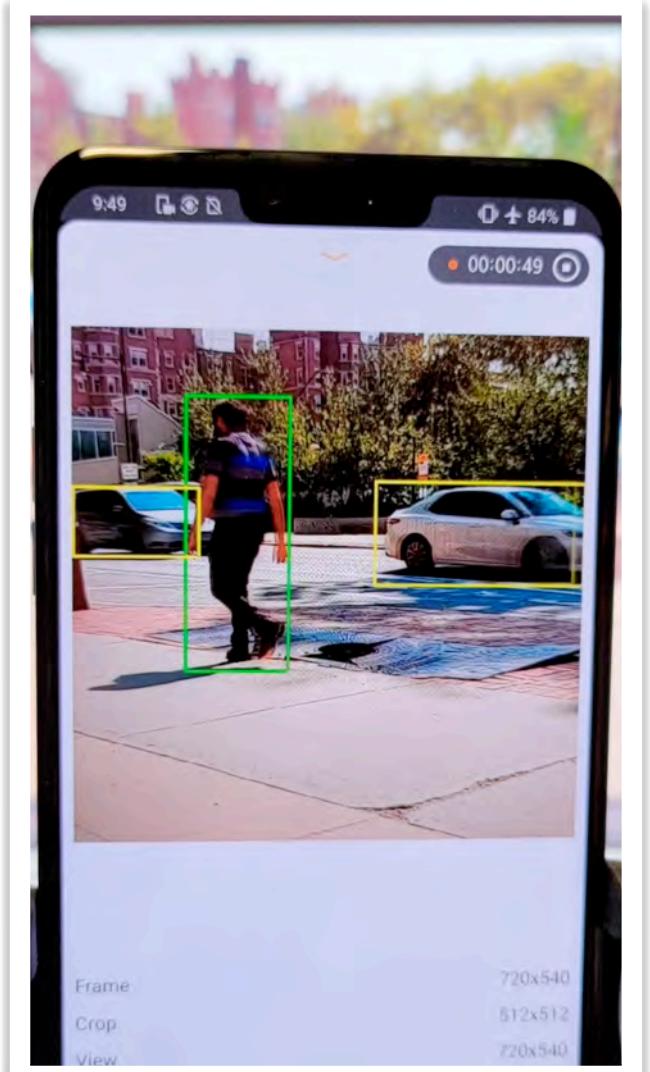
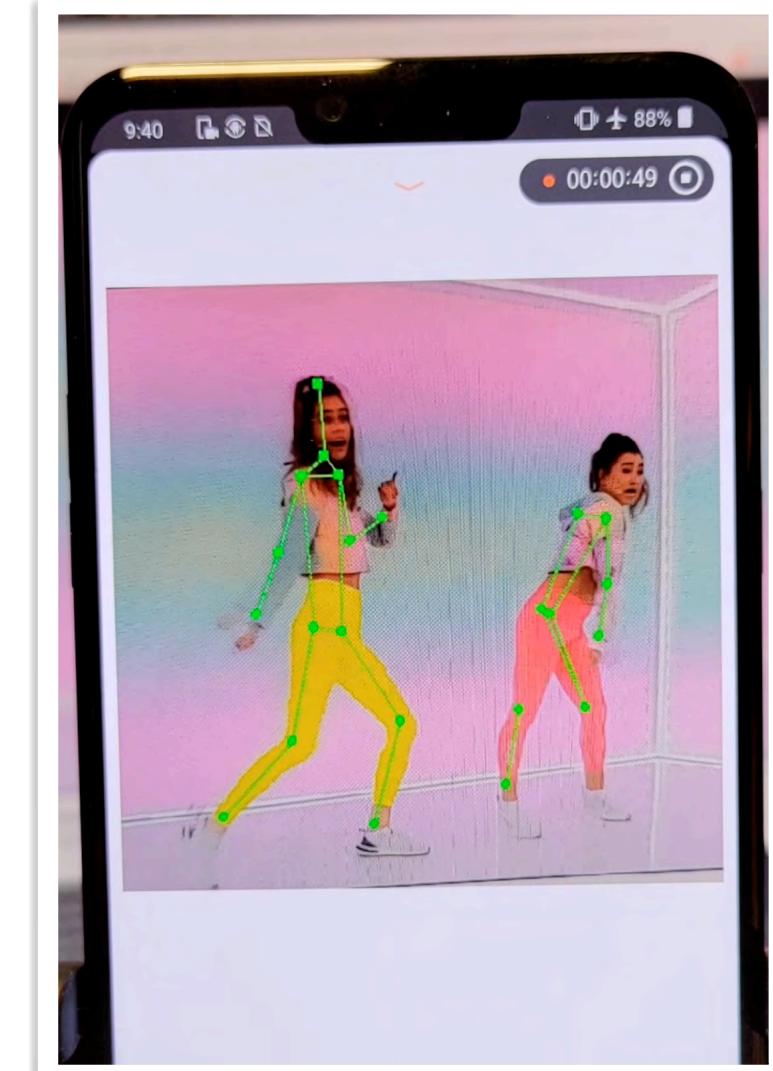


TinyML and Efficient Deep Learning Computing

6.5940 • Fall 2023 • MIT

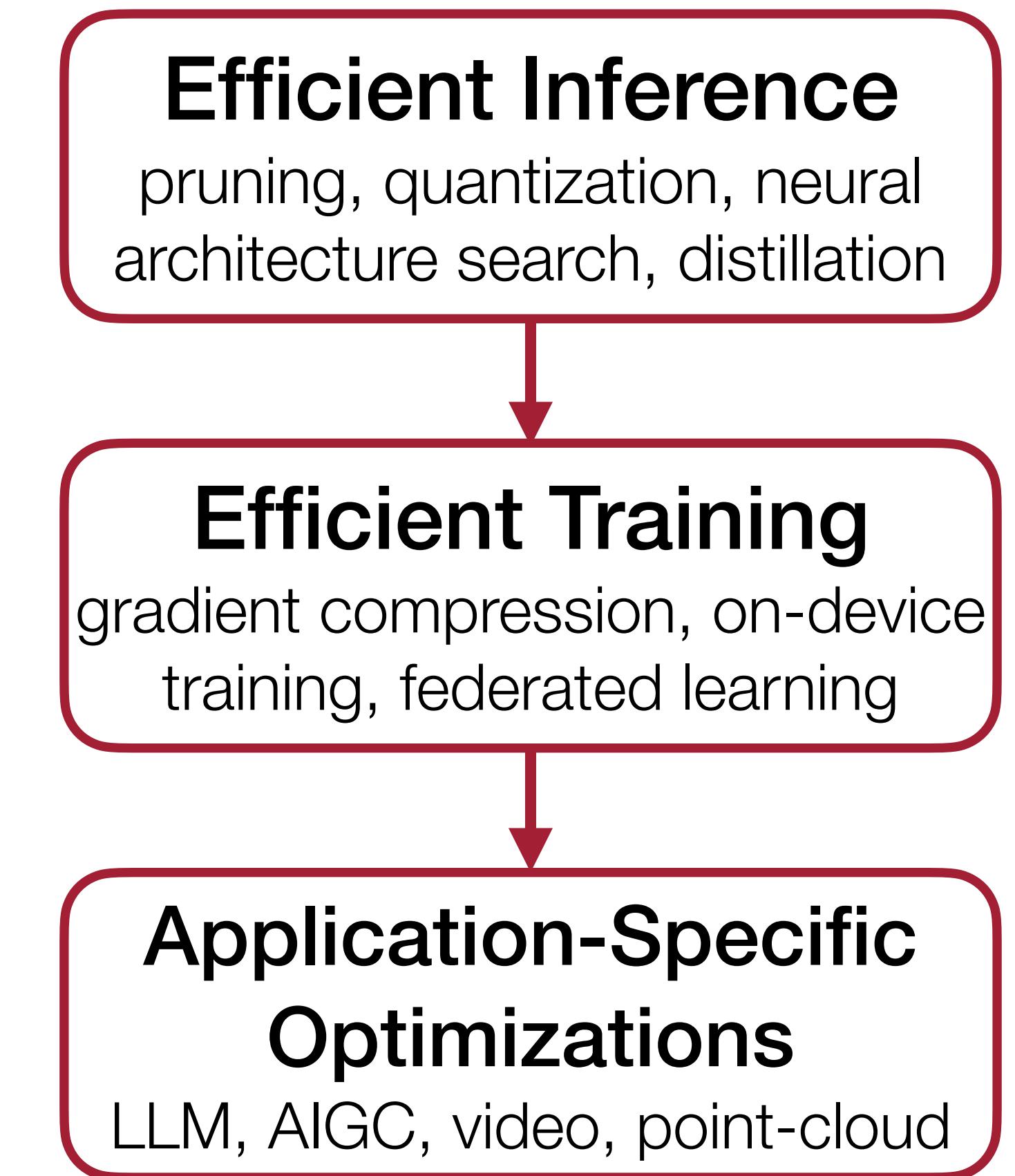
Large generative models (e.g., large language models, diffusion models) have shown remarkable performance, but they require a massive amount of computational resources. To make them more accessible, it is crucial to improve their efficiency.

This course will introduce efficient AI computing techniques that enable powerful deep learning applications on resource-constrained devices. Topics include model compression, pruning, quantization, neural architecture search, distributed training, data/model parallelism, gradient compression, and on-device fine-tuning. It also introduces application-specific acceleration techniques for large language models, diffusion models, video recognition, and point cloud. This course will also cover topics about quantum machine learning. Students will get hands-on experience deploying large language models (e.g., LLaMA 2) on a laptop.

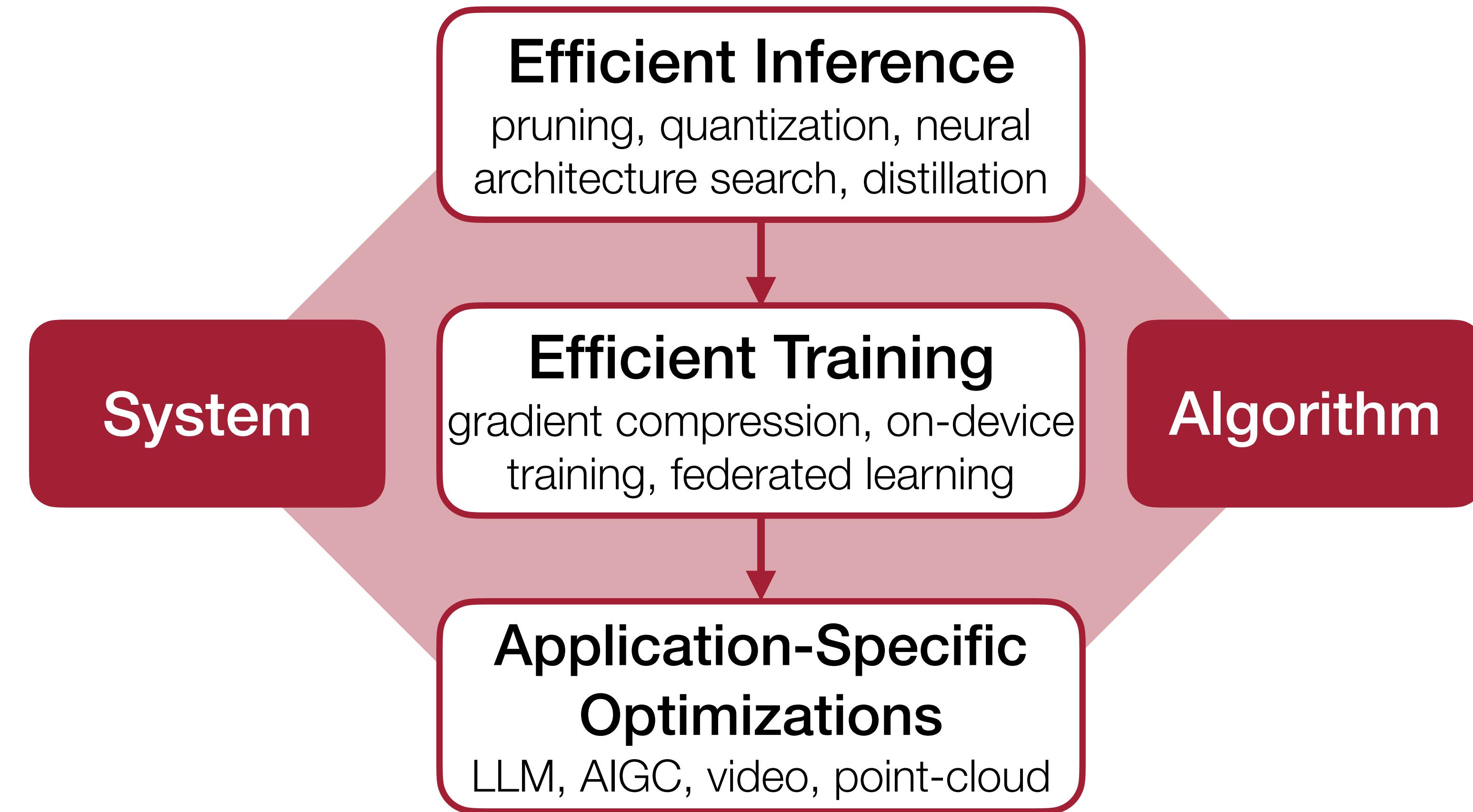


- **Time:** Tuesday/Thursday 3:30-5:00 pm Eastern Time
- **Location:** 36-156
- **Office Hour:** Thursday 5:00-6:00 pm Eastern Time, 38-344 Meeting Room
- **Discussion:** [Discord](#)
- **Homework submission:** [Canvas](#)
- **Online lectures:** The lectures will be streamed on [YouTube](#).
- **Resources:** [MIT HAN Lab](#), [HAN Lab Github](#), [TinyML](#), [MCUNet](#), [OFA](#), [SmoothQuant](#)
- **Contact:**
 - Students can ask all course-related questions on [Discord](#).
 - For external inquiries, personal matters, or emergencies, you can email us at efficientml-staff@mit.edu.
 - If you are interested in getting updates, please sign up [here](#) to join our mailing list to get notified!

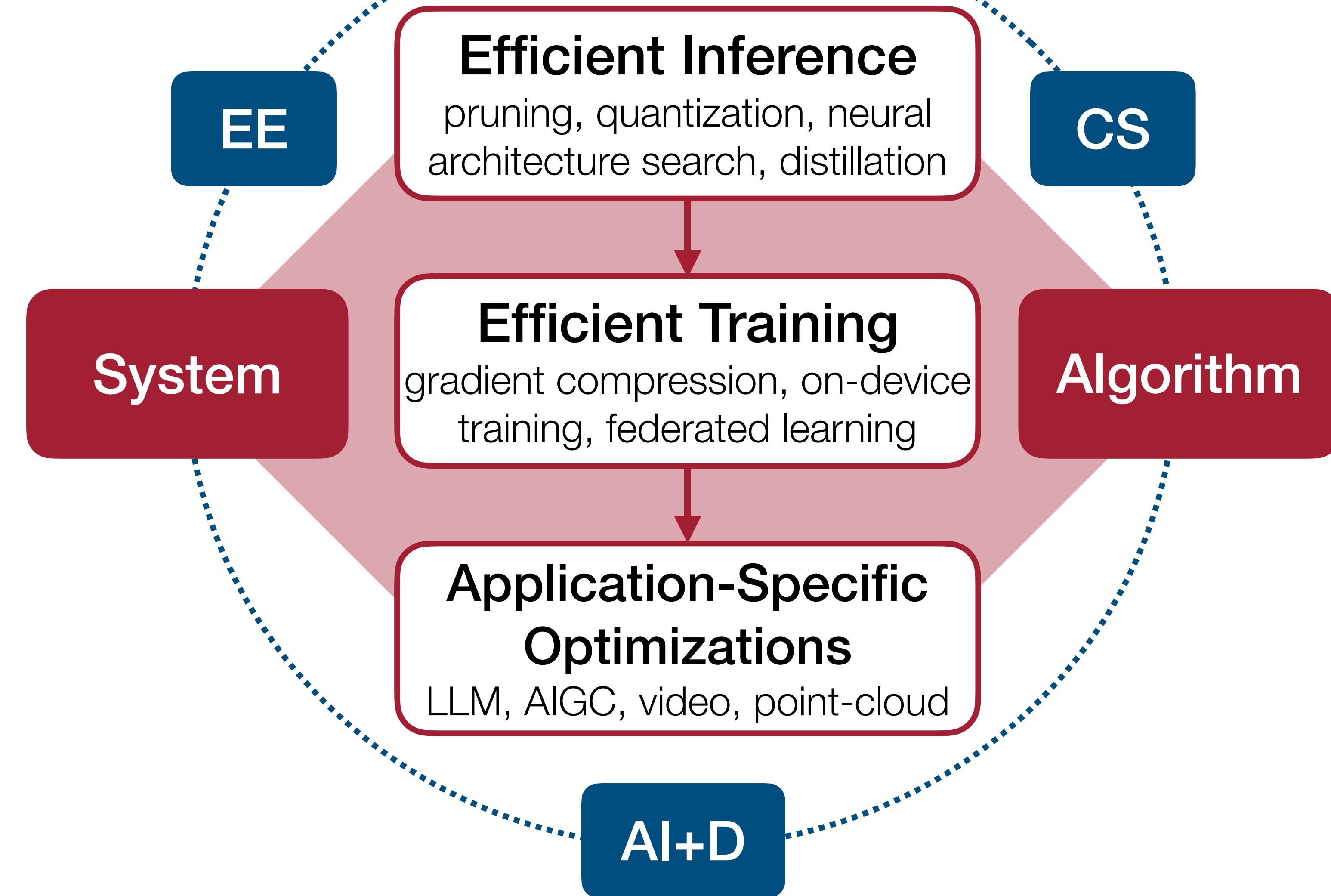
Course Overview



Course Overview



Course Overview



Course Overview

Computation Structures

(6.1910[6.004]), pre-req

Hardware Architecture for Deep Learning
(6.5930[6.825])

Microcomputer Project Lab
(6.2060[6.115])

EE

System

Efficient Inference

pruning, quantization, neural architecture search, distillation

CS

Algorithm

Efficient Training

gradient compression, on-device training, federated learning

Application-Specific Optimizations

LLM, AIGC, video, point-cloud

AI+D

Introduction to Machine Learning (6.3900[6.036]), pre-req

Deep Learning (6.S898)

Advances in Computer Vision (6.8300)

Computer System Architecture
(6.5900 [6.823])

Software Performance Engineering
(6.1060[6.172])

Mobile and Sensor Computing
(6.1820[6.808])

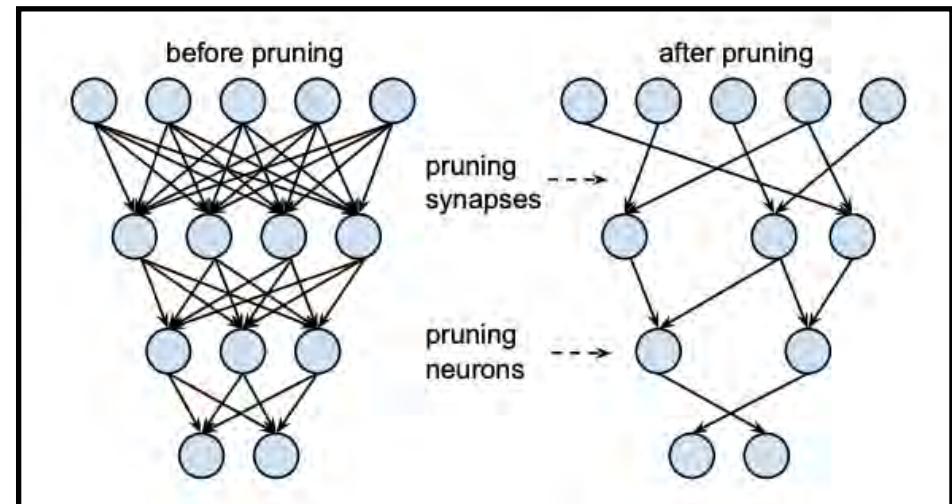
Lecture Structure

- **23 Lectures (90min)**
- **1 Guest Lecture**
- **5 Lab Assignments**
- **1 Final Project:**
 - proposal
 - presentation + demo
 - written report

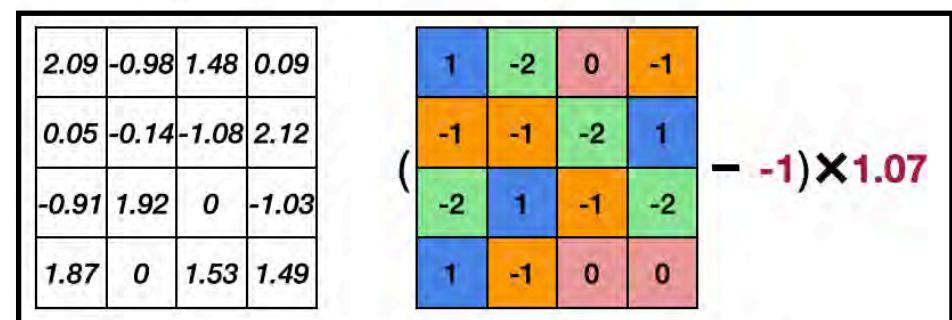
9/7	Lecture 1: Introduction [slides] [video] [video (live)]	
9/12	Lecture 2: Basics of Deep Learning [slides] [video] [video (live)]	
Efficient Inference		
9/14	Lecture 3: Pruning and Sparsity (Part I) [slides] [video] [video (live)]	
9/19	Lecture 4: Pruning and Sparsity (Part II) [slides] [video] [video (live)]	Lab 1 out
9/21	Lecture 5: Quantization (Part I) [slides] [video] [video (live)]	
9/26	Lecture 6: Quantization (Part II) [slides] [video] [video (live)]	
9/28	Lecture 7: Neural Architecture Search (Part I) [slides] [video] [video (live)]	Lab 1 due, Lab 2 out
10/3	Lecture 8: Neural Architecture Search (Part II) [slides] [video] [video (live)]	
10/5	Lecture 9: Knowledge Distillation [slides] [video] [video (live)]	Lab 3 out
10/10	Student Holiday — No Class	
10/12	Lecture 10: MCUNet [slides] [video] [video (live)]	Lab 2 due
10/17	Lecture 11: TinyEngine [slides] [video] [video (live)]	

Efficient Transformer and Generative AI		
10/19	Lecture 12: Transformer and LLM (Part I) [slides] [video] [video (live)]	Lab 3 due, Lab 4 out
10/24	Lecture 13: Transformer and LLM (Part II) [slides] [video] [video (live)]	
10/26	Lecture 14: Vision Transformer [slides] [video] [video (live)]	
10/31	Lecture 15: GAN, Video, and Point Cloud [slides] [video] [video (live)]	Lab 4 due, Lab 5 out
11/2	Lecture 16: Diffusion Model [slides] [video] [video (live)]	
Efficient Training and System Support		
11/7	Lecture 17: Distributed Training (Part I) [slides] [video] [video (live)]	
11/9	Lecture 18: Distributed Training (Part II) [slides] [video] [video (live)]	Lab 5 due
11/14	Lecture 19: On-Device Training (Part I) [slides] [video] [video (live)]	Project proposal due
11/16	Lecture 20: On-Device Training (Part II) [slides] [video] [video (live)]	
Quantum ML		
11/21	Lecture 21: Basics of Quantum Computing [slides] [video] [video (live)]	
11/23		Thanksgiving — No Class
11/28	Lecture 22: Quantum Machine Learning [slides] [video] [video (live)]	
11/30	Lecture 23: Noise Robust Quantum ML [slides] [video] [video (live)]	
12/5	Lecture 24: Final Project Presentation [slides] [video] [video (live)]	
12/7	Lecture 25: Final Project Presentation [slides] [video] [video (live)]	
12/12	Lecture 26: Course Summary & Guest Lecture [slides] [video] [video (live)]	Final report due

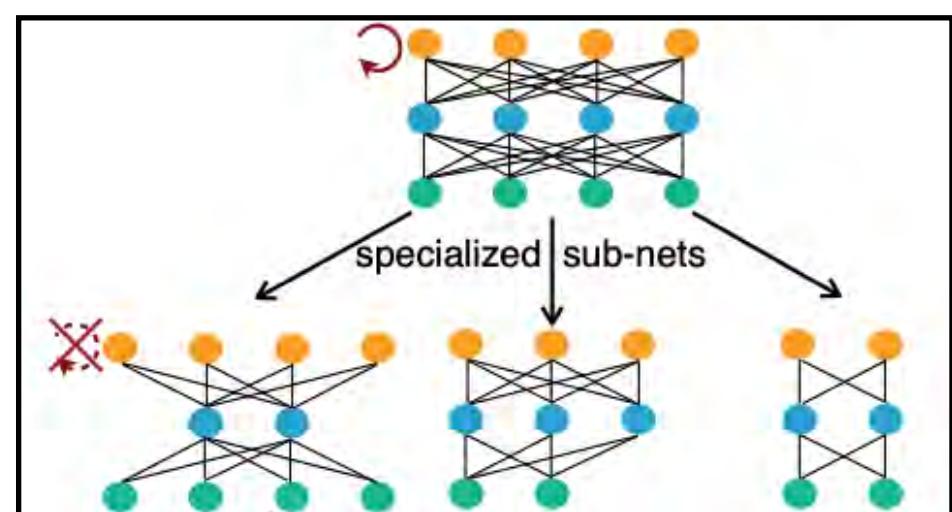
Part I: Efficient Inference



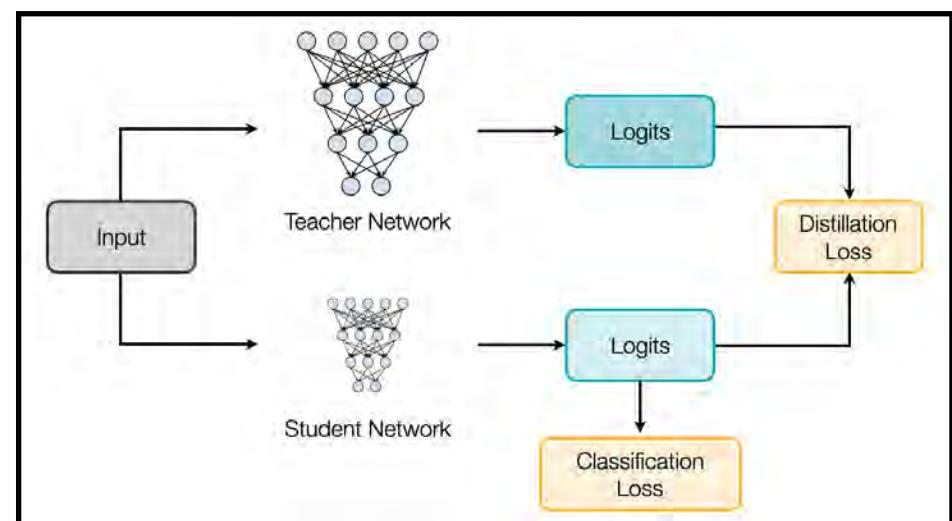
Pruning



Quantization



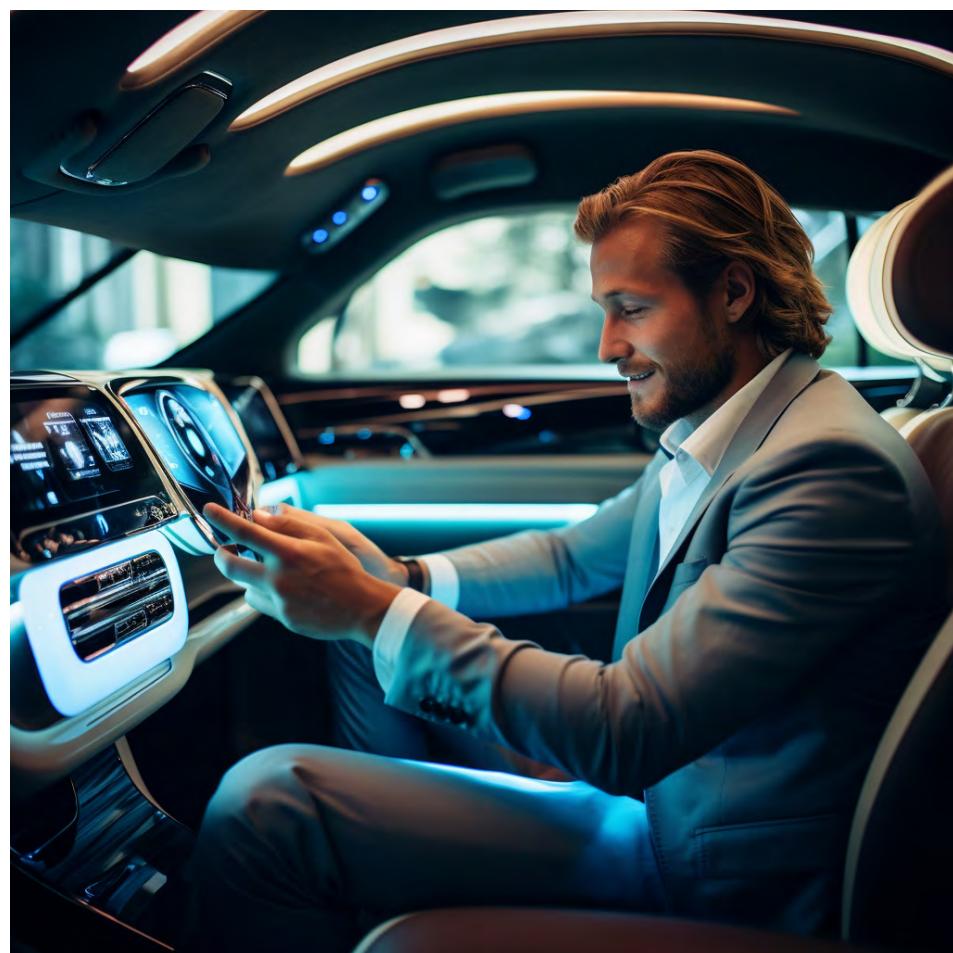
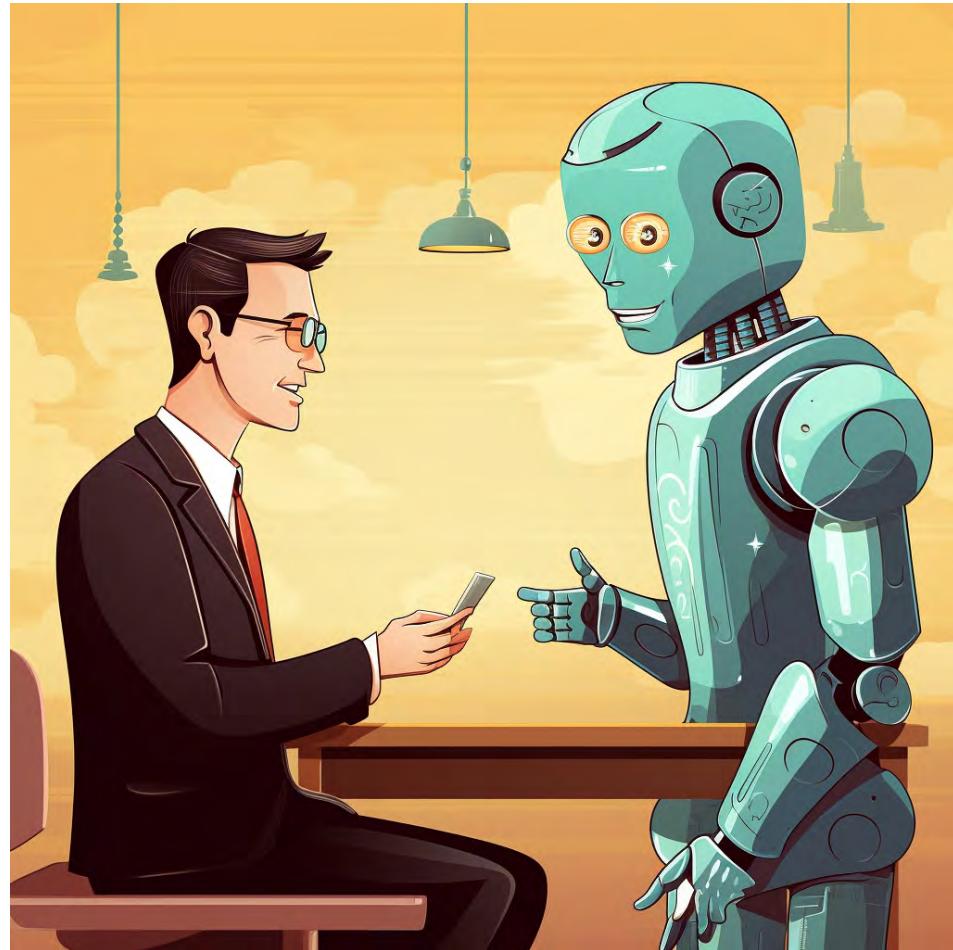
Neural Architecture Search



Knowledge Distillation

9/14	Lecture 3: Pruning and Sparsity (Part I) [slides] [video] [video (live)]	
9/19	Lecture 4: Pruning and Sparsity (Part II) [slides] [video] [video (live)]	Lab 1 out
9/21	Lecture 5: Quantization (Part I) [slides] [video] [video (live)]	
9/26	Lecture 6: Quantization (Part II) [slides] [video] [video (live)]	
9/28	Lecture 7: Neural Architecture Search (Part I) [slides] [video] [video (live)]	Lab 1 due, Lab 2 out
10/3	Lecture 8: Neural Architecture Search (Part II) [slides] [video] [video (live)]	
10/5	Lecture 9: Knowledge Distillation [slides] [video] [video (live)]	Lab 3 out
10/10		Student Holiday — No Class
10/12	Lecture 10: MCUNet [slides] [video] [video (live)]	Lab 2 due
10/17	Lecture 11: TinyEngine [slides] [video] [video (live)]	

Part II: Application-Specific Optimizations



10/19 Lecture 12: Transformer and LLM (Part I)

[slides] [video] [video (live)]

Lab 3 due, Lab 4 out

10/24 Lecture 13: Transformer and LLM (Part II)

[slides] [video] [video (live)]

10/26 Lecture 14: Vision Transformer

[slides] [video] [video (live)]

10/31 Lecture 15: GAN, Video, and Point Cloud

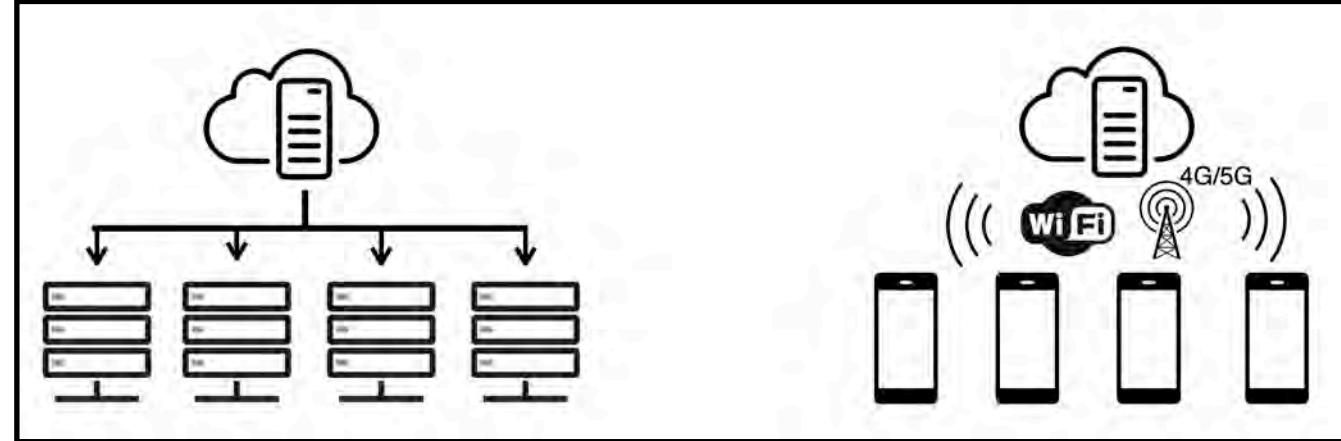
[slides] [video] [video (live)]

Lab 4 due, Lab 5 out

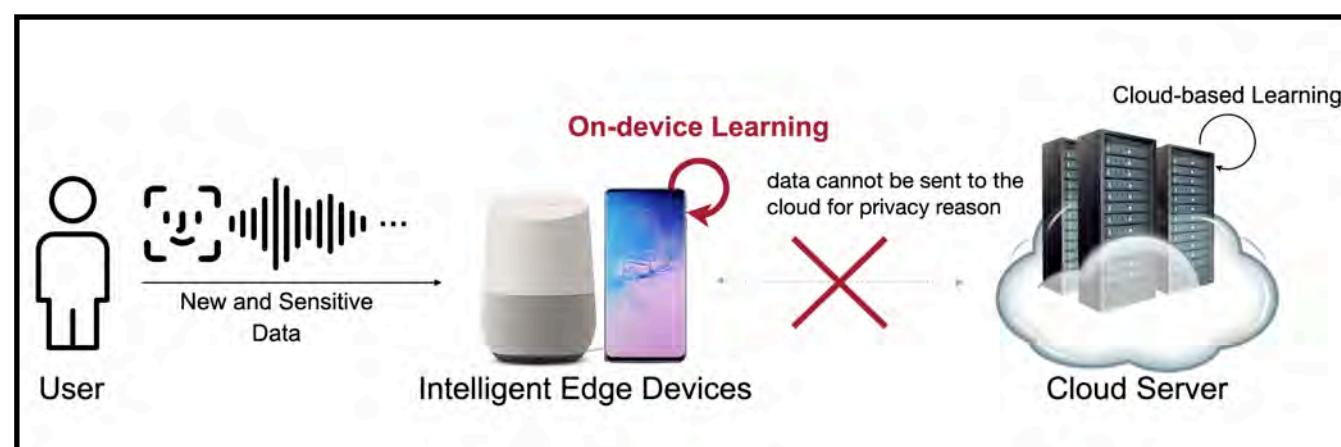
11/2 Lecture 16: Diffusion Model

[slides] [video] [video (live)]

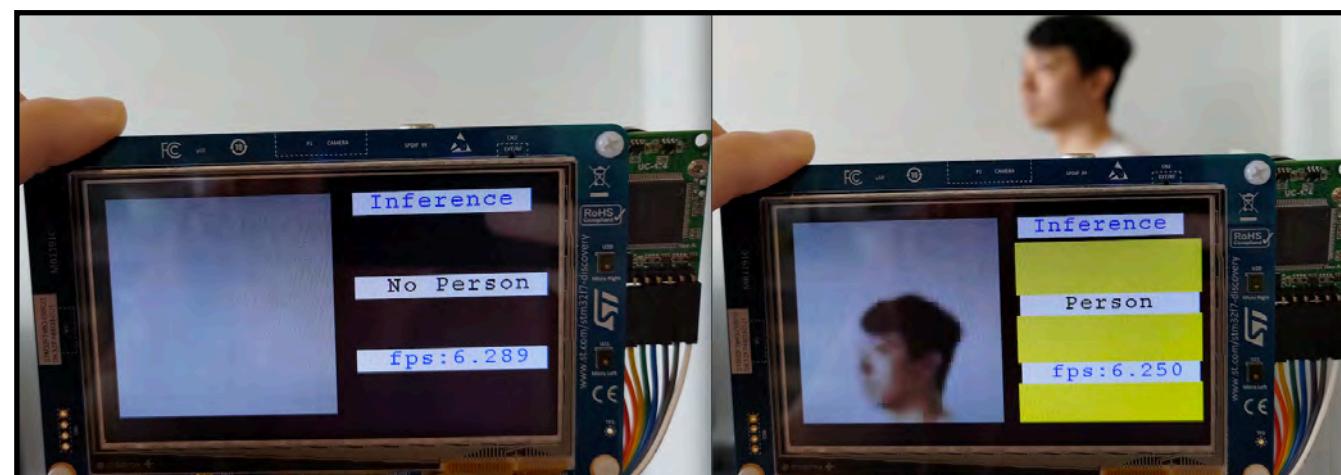
Part III: Efficient Training



Distributed Training



On-Device Learning



TinyEngine on MCU

11/7 Lecture 17: **Distributed Training (Part I)**

[slides] [video] [video (live)]

Lab 5 due

11/9 Lecture 18: **Distributed Training (Part II)**

[slides] [video] [video (live)]

Project proposal due

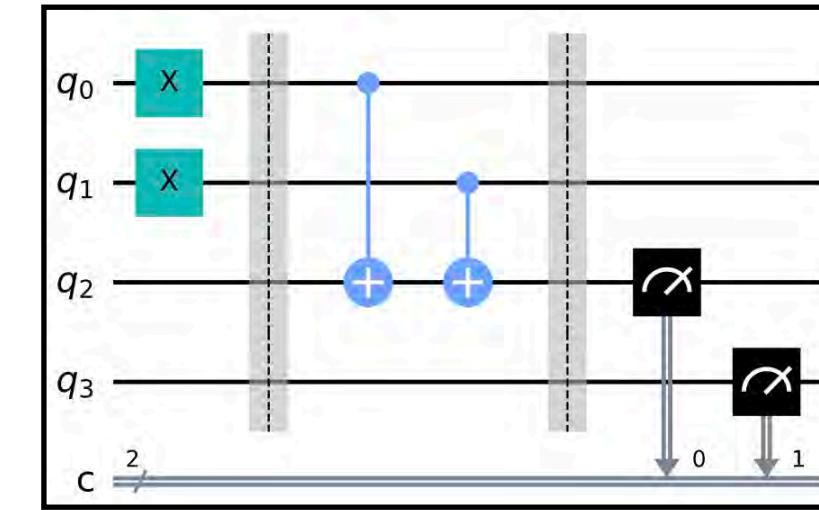
11/14 Lecture 19: **On-Device Training (Part I)**

[slides] [video] [video (live)]

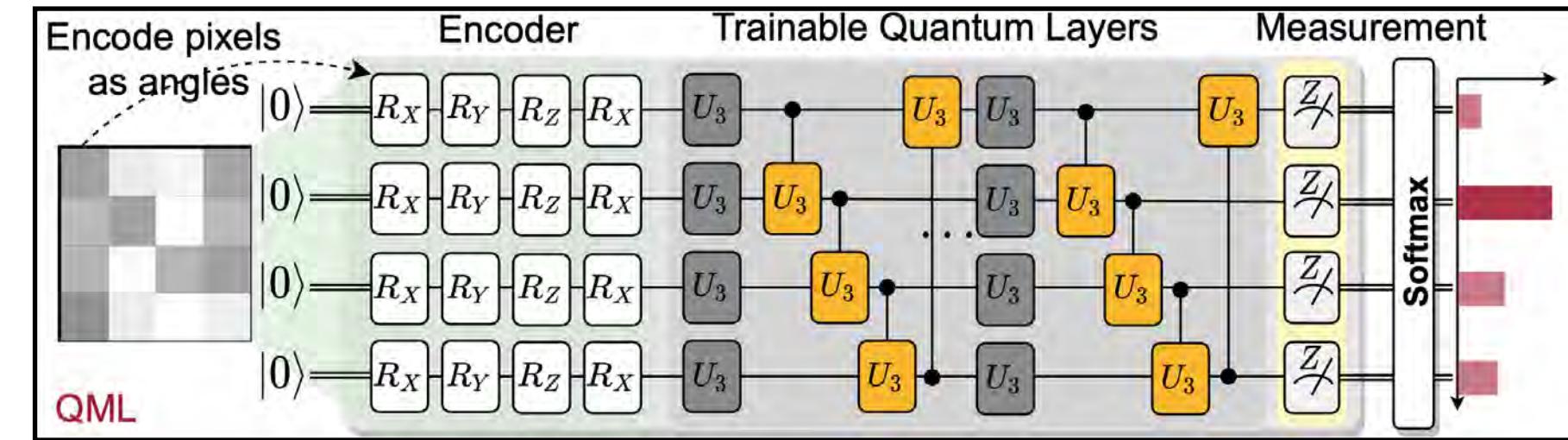
11/16 Lecture 20: **On-Device Training (Part II)**

[slides] [video] [video (live)]

Part IV: Quantum ML



Basics Quantum Circuit



Quantum ML, Noise Robustness

11/21 Lecture 21: Basics of Quantum Computing

[slides] [video] [video (live)]

11/23

Thanksgiving — No Class

11/28 Lecture 22: Quantum Machine Learning

[slides] [video] [video (live)]

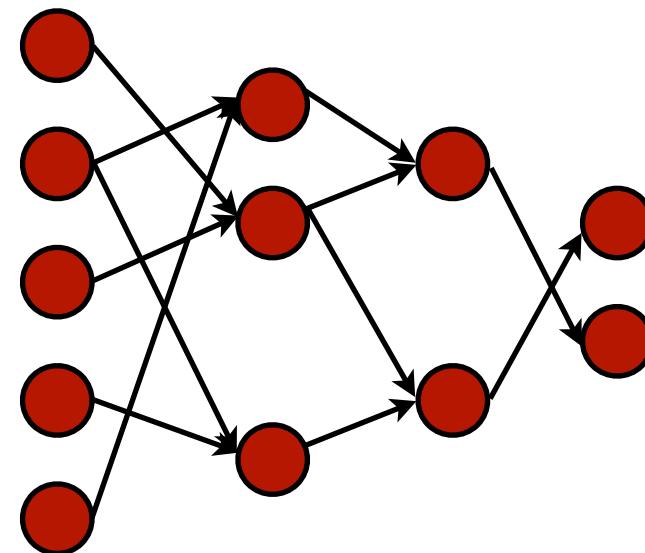
11/30 Lecture 23: Noise Robust Quantum ML

[slides] [video] [video (live)]

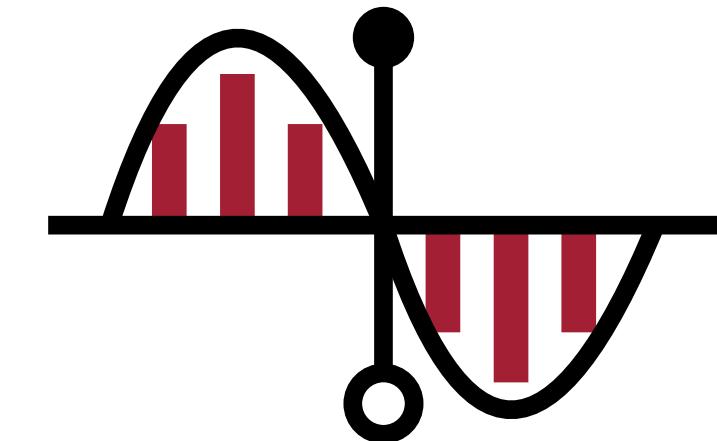
Labs



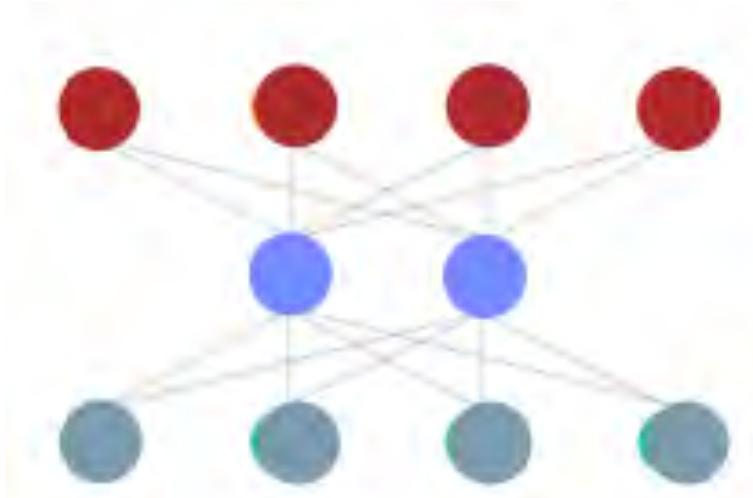
Lab 0 – Getting Started with PyTorch



Lab 1 – Pruning



Lab 2 – Quantization



Lab 3 – Neural Architecture Search



Lab 4 – LLM Compression



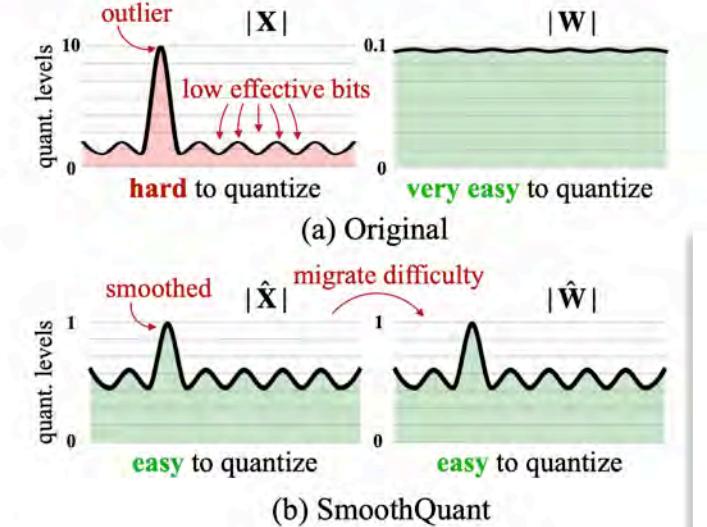
Lab 5 – LLM Deployment on Laptop

Final Projects – Last Year

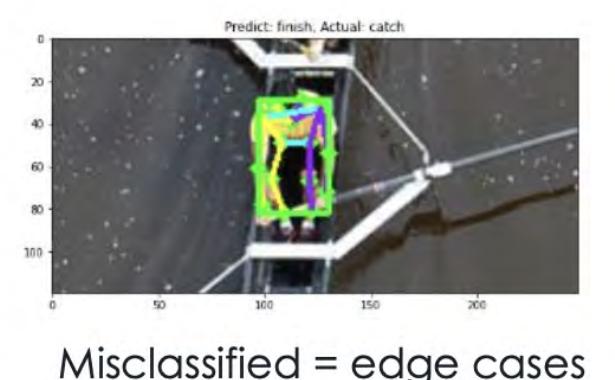
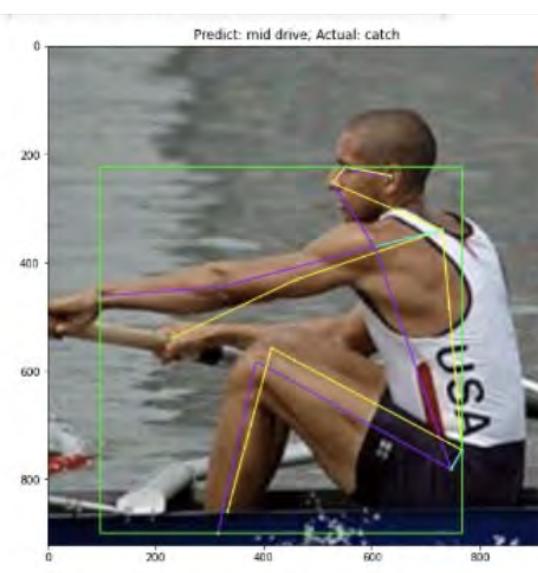
The poster features a blue background with a white silhouette of a rower in the center. The text "aiP: Al-Row" is displayed in white. Below it, the project description reads: "The First Real Time Mobile ML Analytics for both Para and Non-Para Rowers". The names "Emelie, Sol, Veronica" are listed. At the bottom, there is a link: "MIT 6.S965: TinyML and Efficient Deep Learning Computing" and "https://efficientml.ai".

SmoothQuant: Accurate and Efficient Post-Training Quantization for LLMs

LLM (100B+)	Accuracy	Hardware Efficiency
ZeroQuant	✗	✓
Outlier Suppression	✗	✓
LLM.int8()	✓	✗
SmoothQuant	✓	✓



- We propose SmoothQuant, an **accurate** and **efficient** post-training-quantization (PTQ) method to enable 8-bit weight, 8-bit activation (**W8A8**) quantization for LLMs.
- Since **weights are easy** to quantize while **activations are not**, SmoothQuant smooths the activation outliers by **migrating the quantization difficulty from activations to weights** with a mathematically equivalent transform.



Misclassified = edge cases

The poster features a black background with a large, stylized red and blue infinity symbol. The title "Knowledge Distillation and Quantization for Efficient Keyword Spotting" is displayed in white. Below it, the names "Arman Dave and Julian Hamelberg" are listed. At the top right, the code "6.S95" is visible.

Optimizing TinyEngine Kernels and Experimental Kernel Generation with Multistaging and Meta-Programming

...

Anne Ouyang, Pranav Krishna

NoiseOut

Real-time Speech Enhancement on Mobile phones

Presenter: Team NoiseOut: {William Zhao, Mark Jabbour, Ian Lee}



Impacts

Art Generation

Video Synthesis

Search Engine Revolution

Chatbots

Predictive Maintenance

Question Answering

Augmented Reality

Gesture Recognition

Storytelling

Autonomous Driving

Video Recognition

Music Composition

Sentiment Analysis

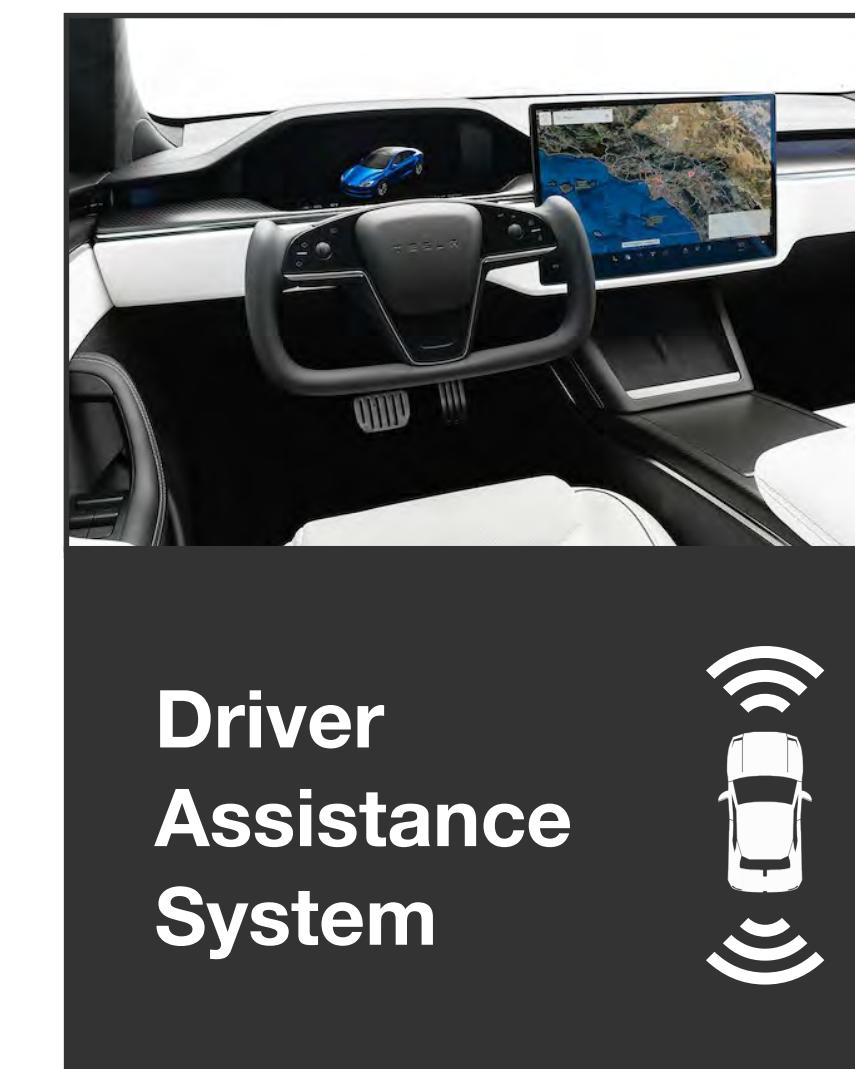
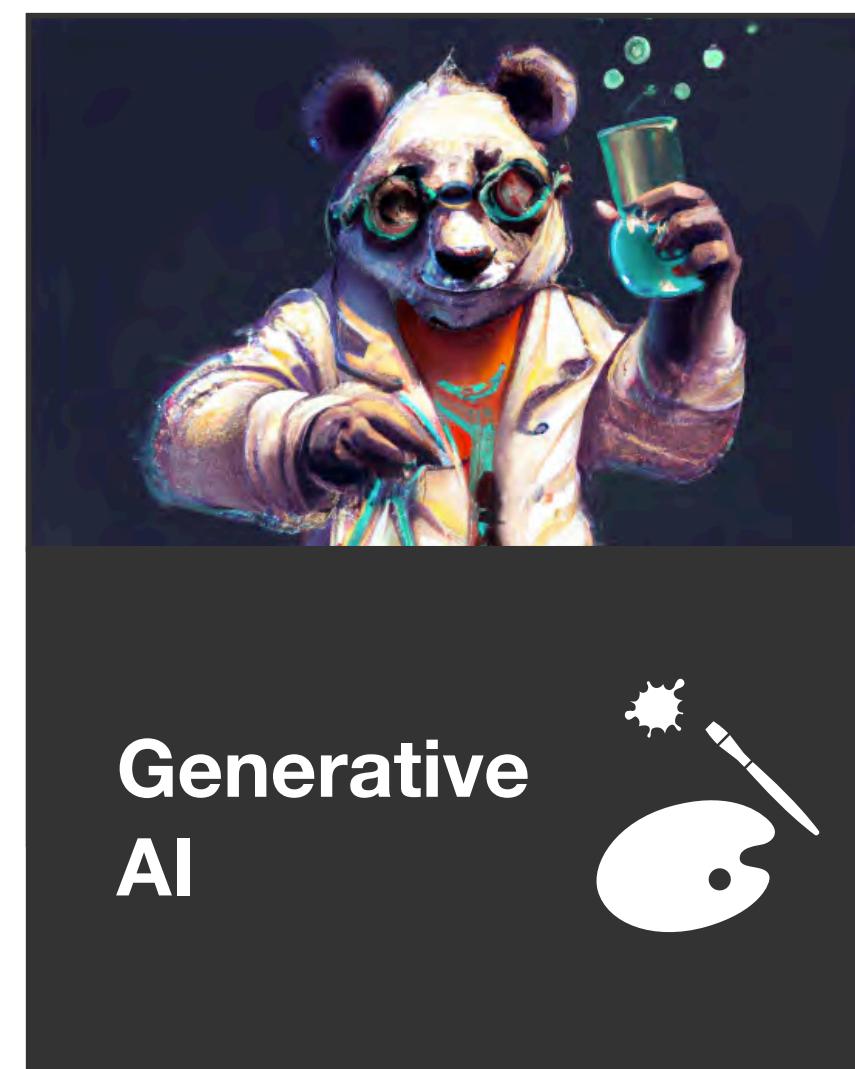
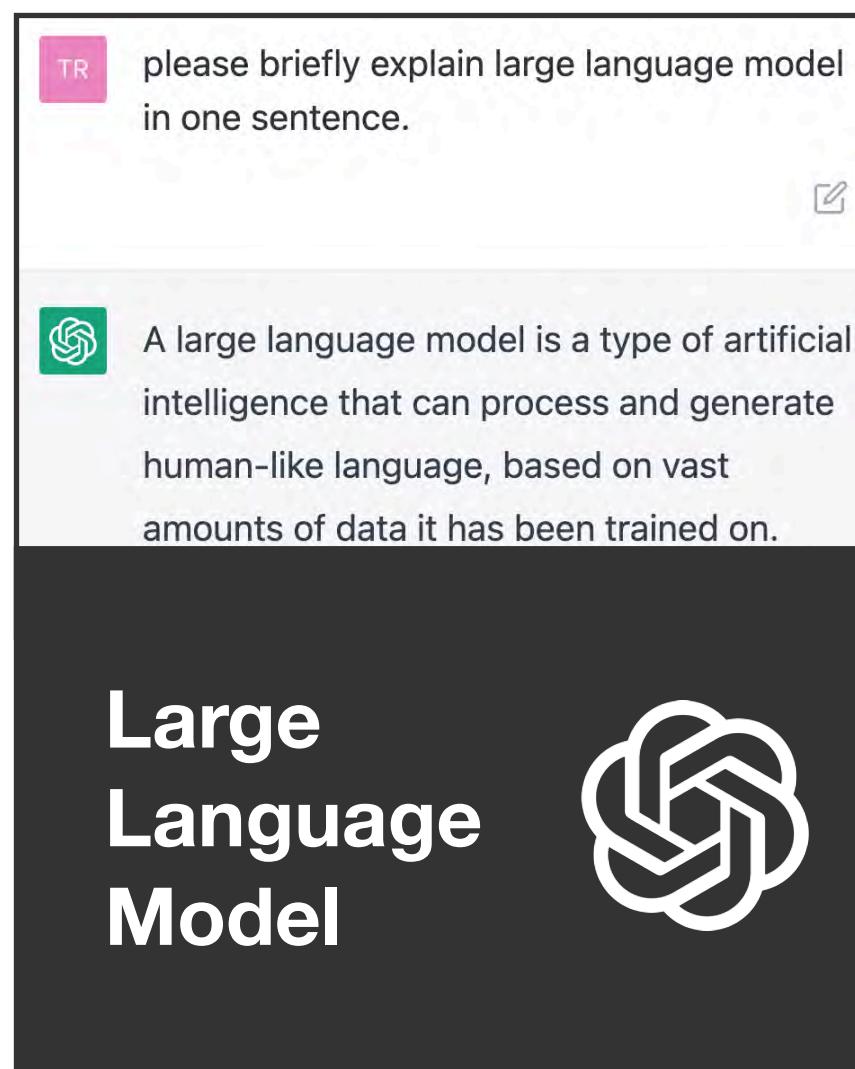
Blind Spot Detection

Health Monitoring

Fashion Design

Machine Translation

Adaptive Cruise Control



AI Application
(demand of computation)

Model
Compression

AI Hardware
(supply of computation)