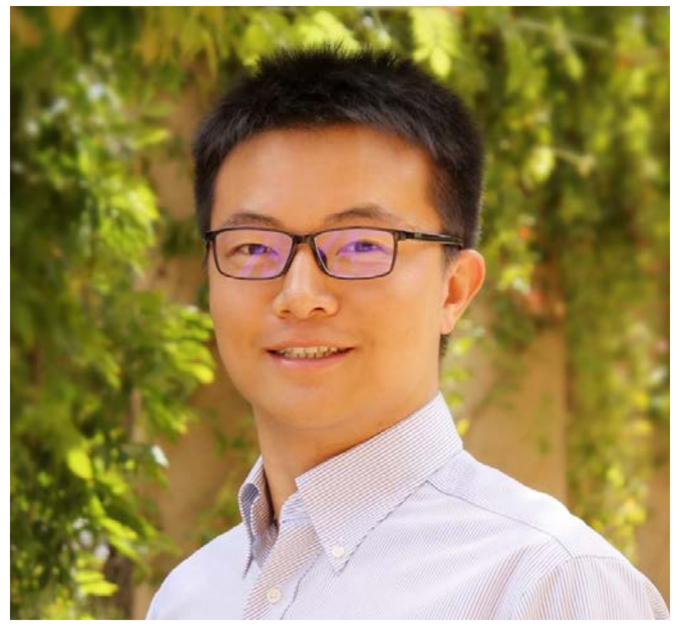
Hardware, AI, and Neural-nets
open source, co-design
http://github.com/mit-han-lab

Model Compression for Efficient AI Computing

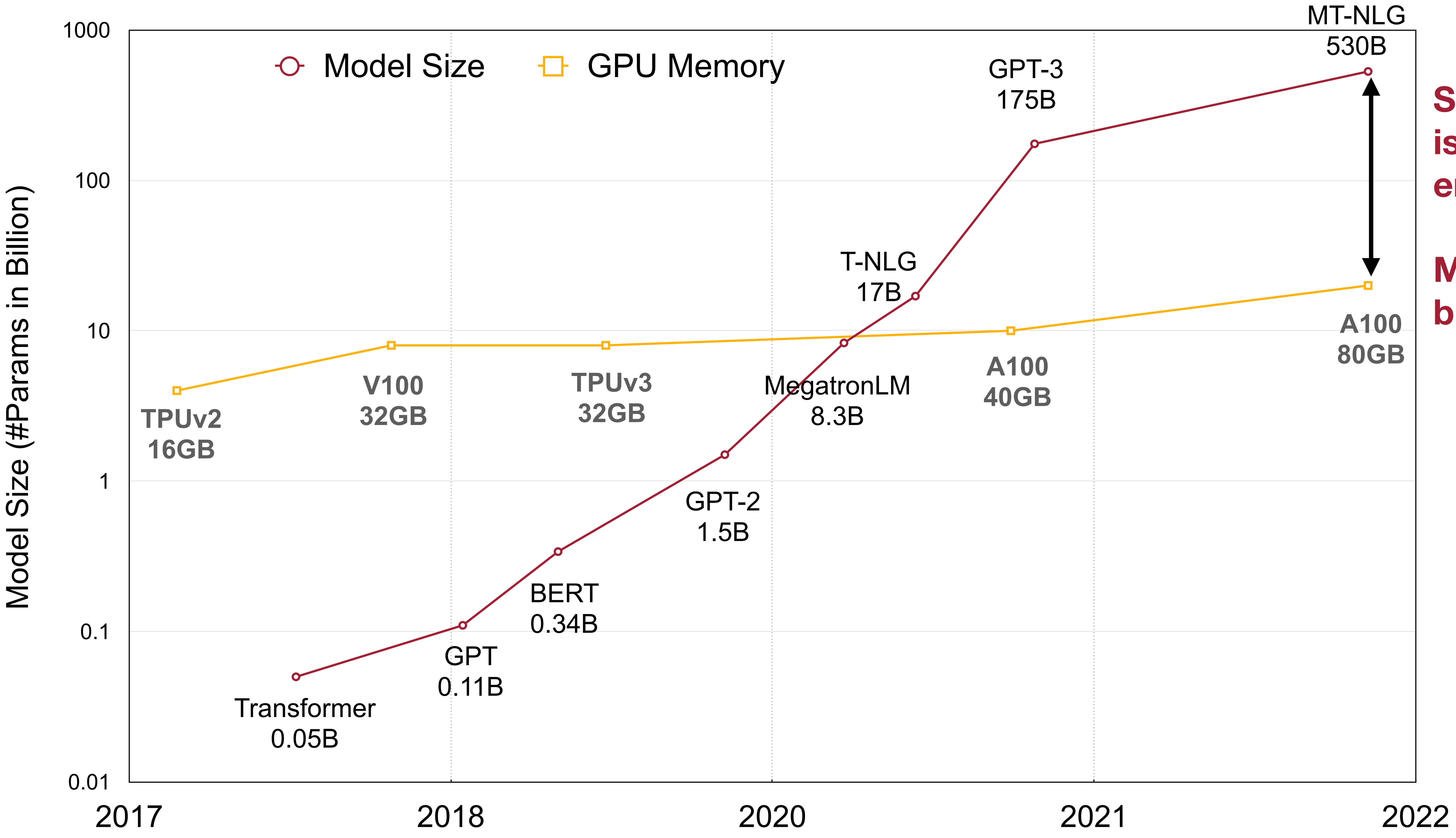
From TinyML to LargeML



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

Model Compression

Bridges the Gap between the Supply and Demand of Computation

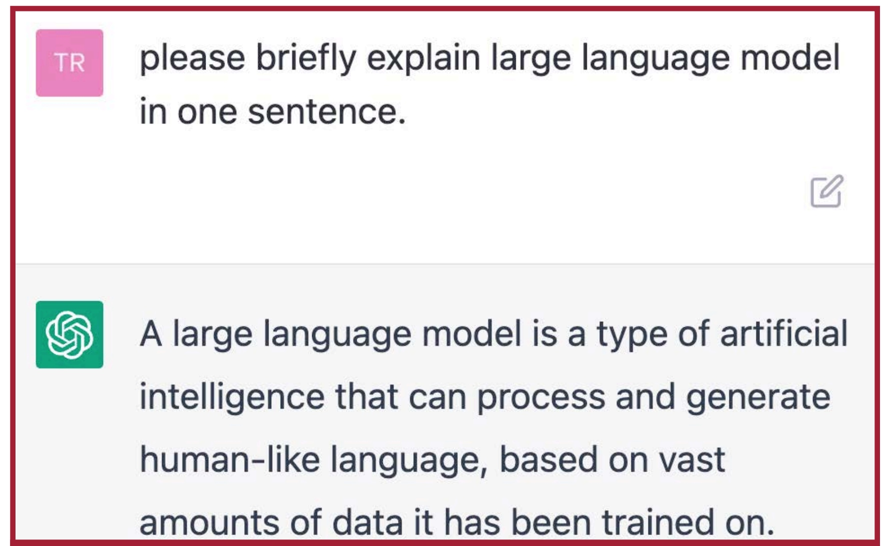


Specialized hardware is important but not enough.

Model compression bridges the gap.

Model Compression

Applications



Large Language Model

Generative AI

Advanced Driver Assistance System

TinyML

Techniques

Hardware-aware NAS

Pruning & Sparsity

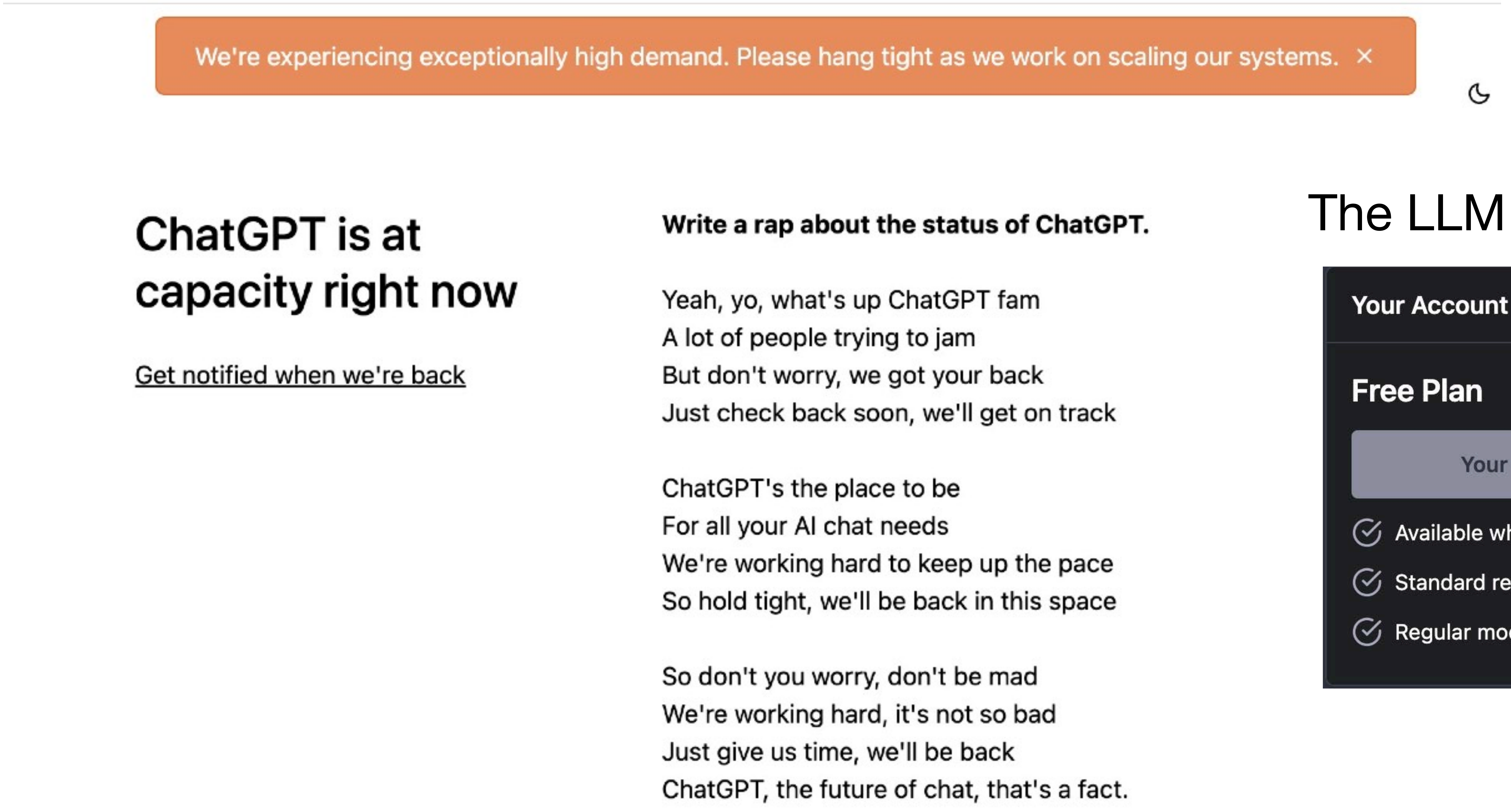
Quantization

Distillation

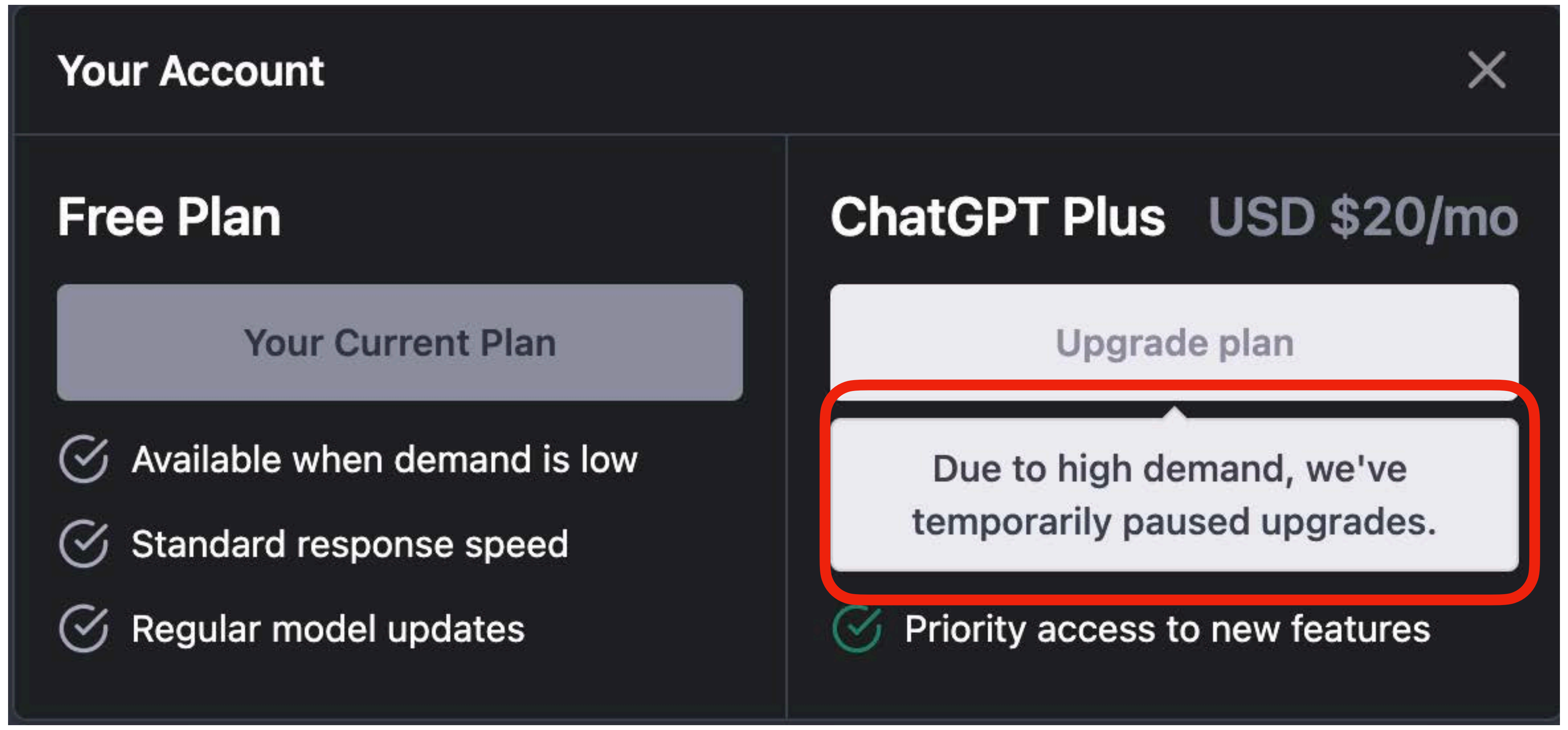
New Primitive

Efficient Large Language Models

Reducing LLM Serving Cost and Accelerating Inference

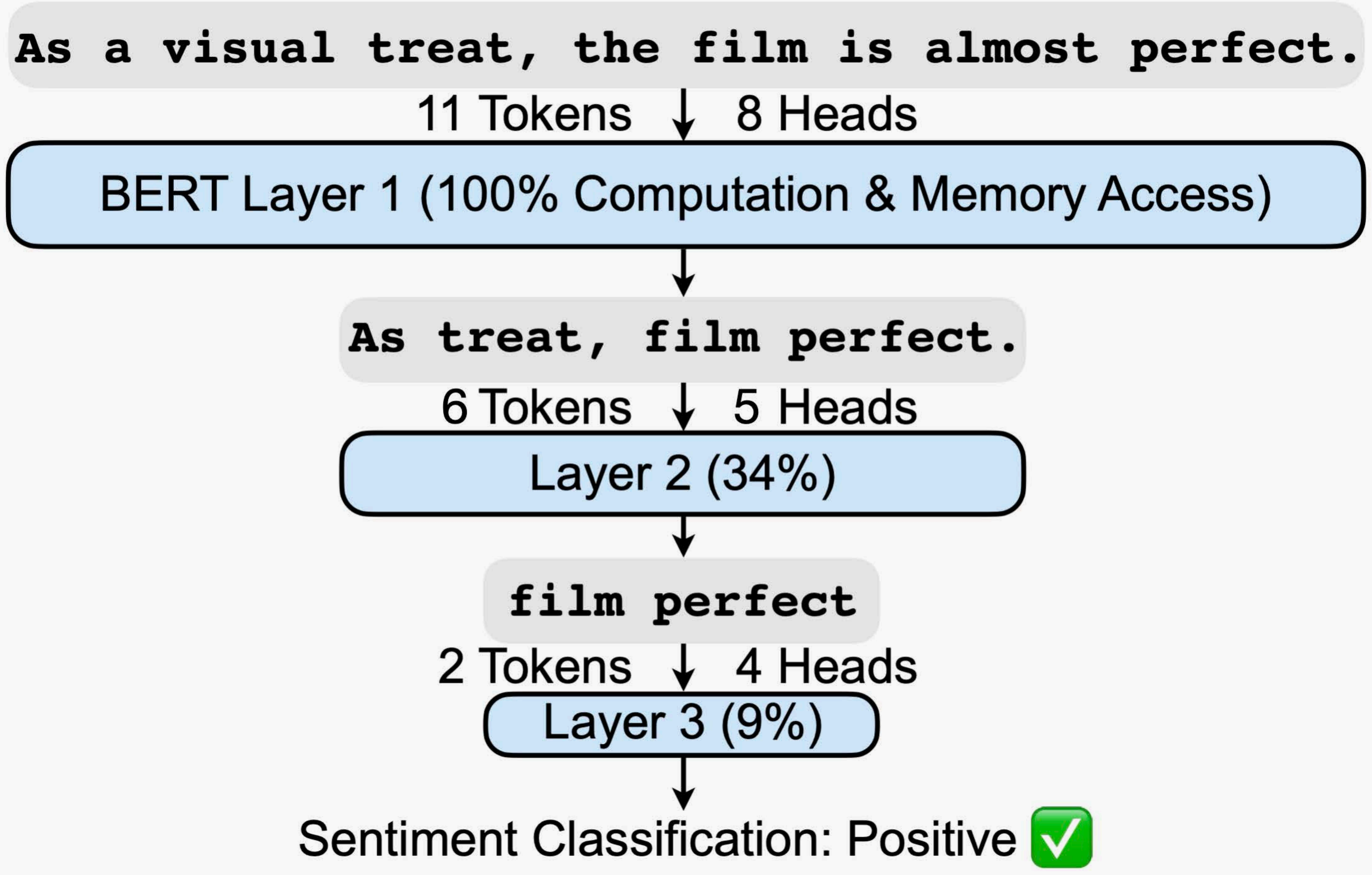


The LLM serving costs are extremely high

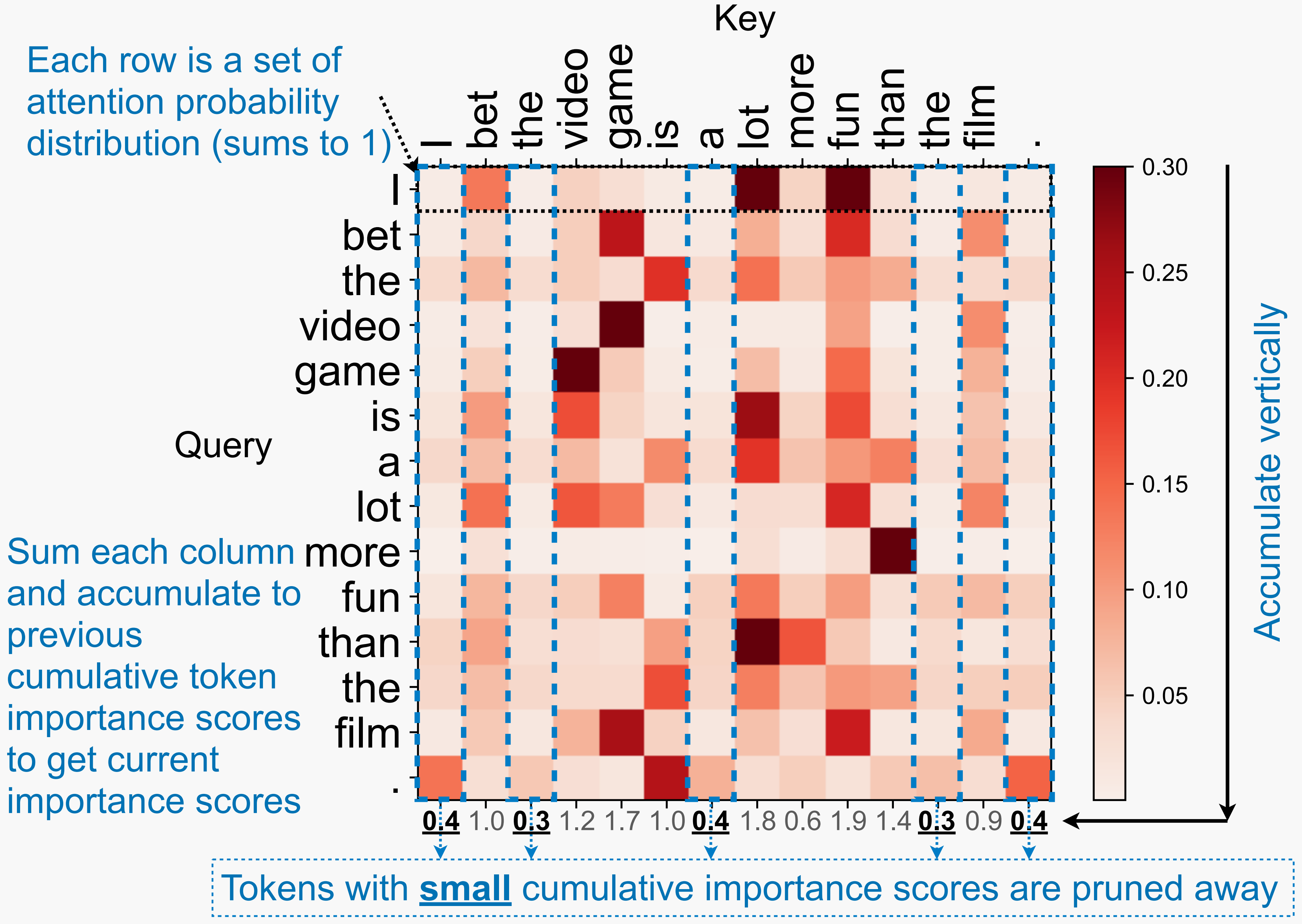


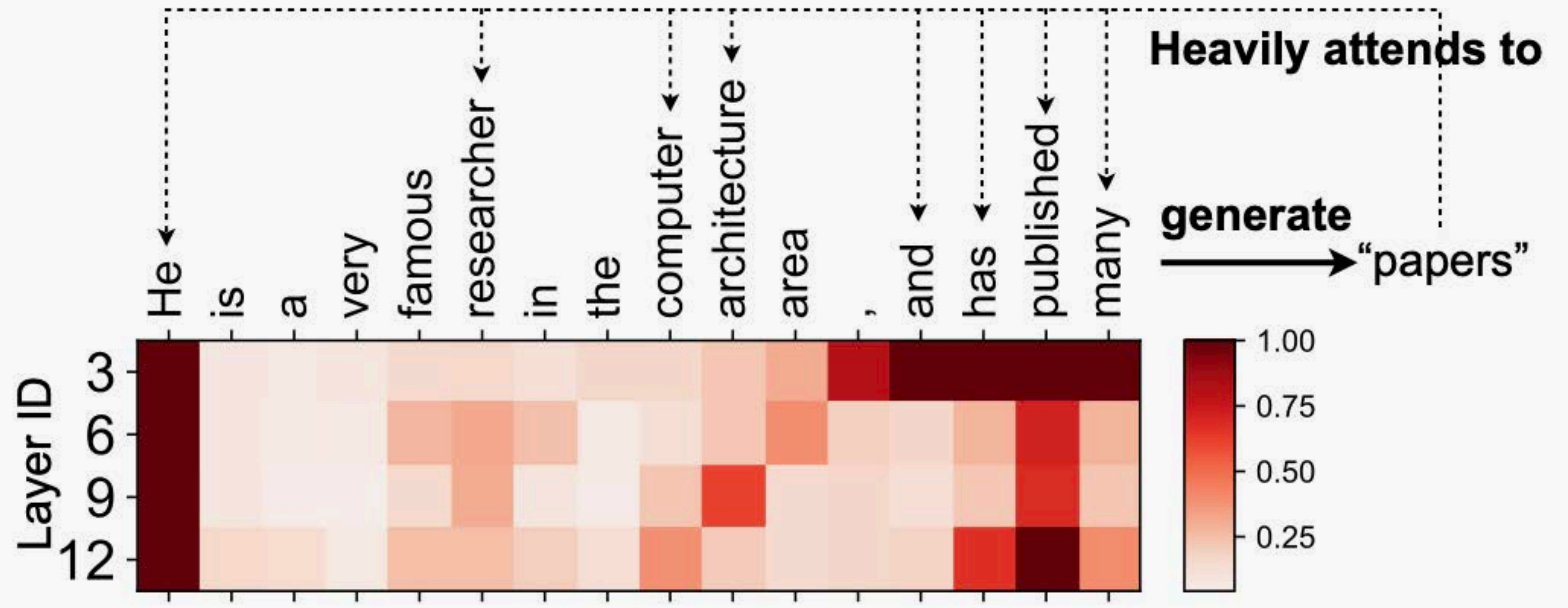
SpAtten: Transformer with Sparse Attention

Token Pruning: not every token are created equal



Remove redundant token and head according to cumulative importance



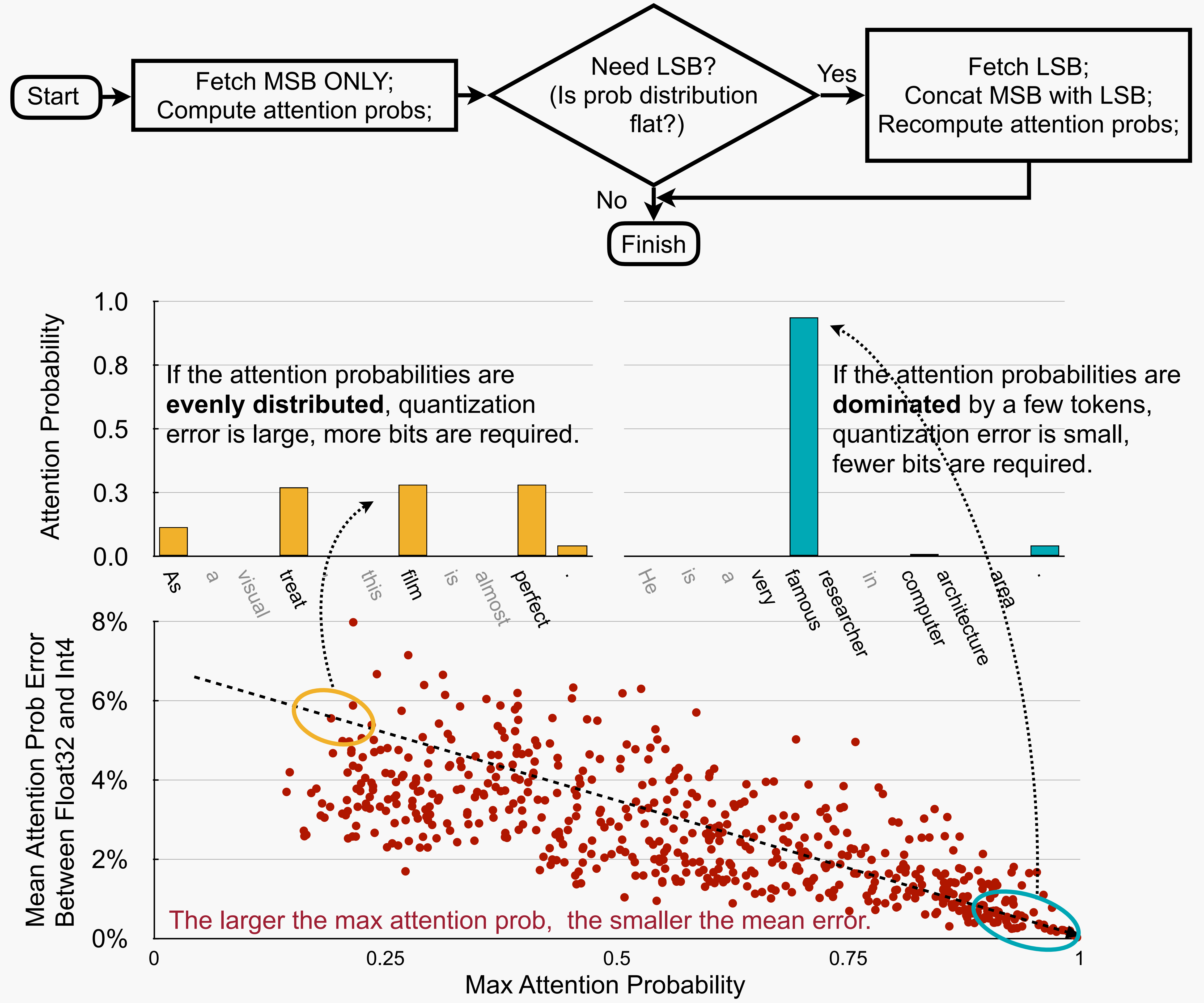


Cumulative importance scores in GPT-2. Unimportant tokens are pruned on the fly. Important tokens are heavily attended.

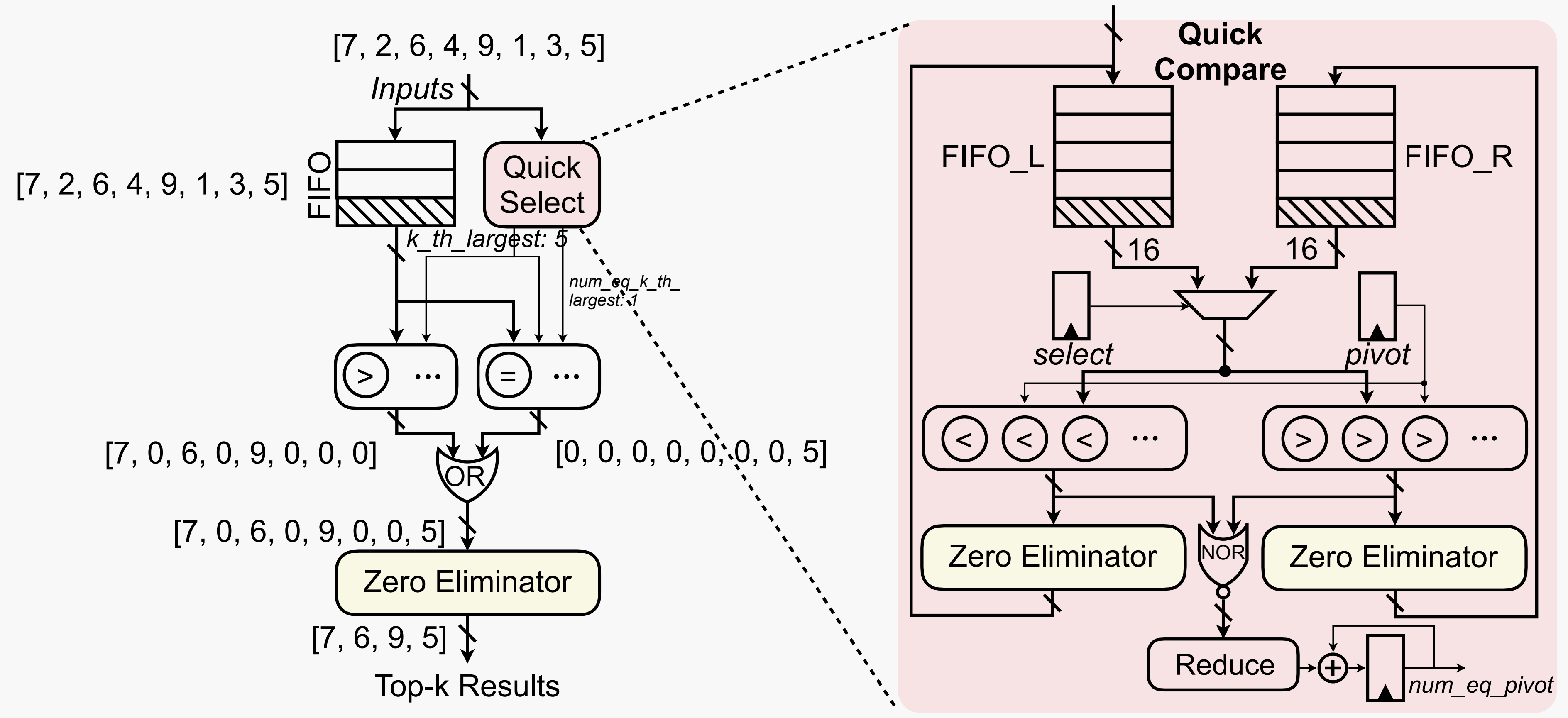
SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning [HPCA'21]

SpAtten: Transformer with Sparse Attention

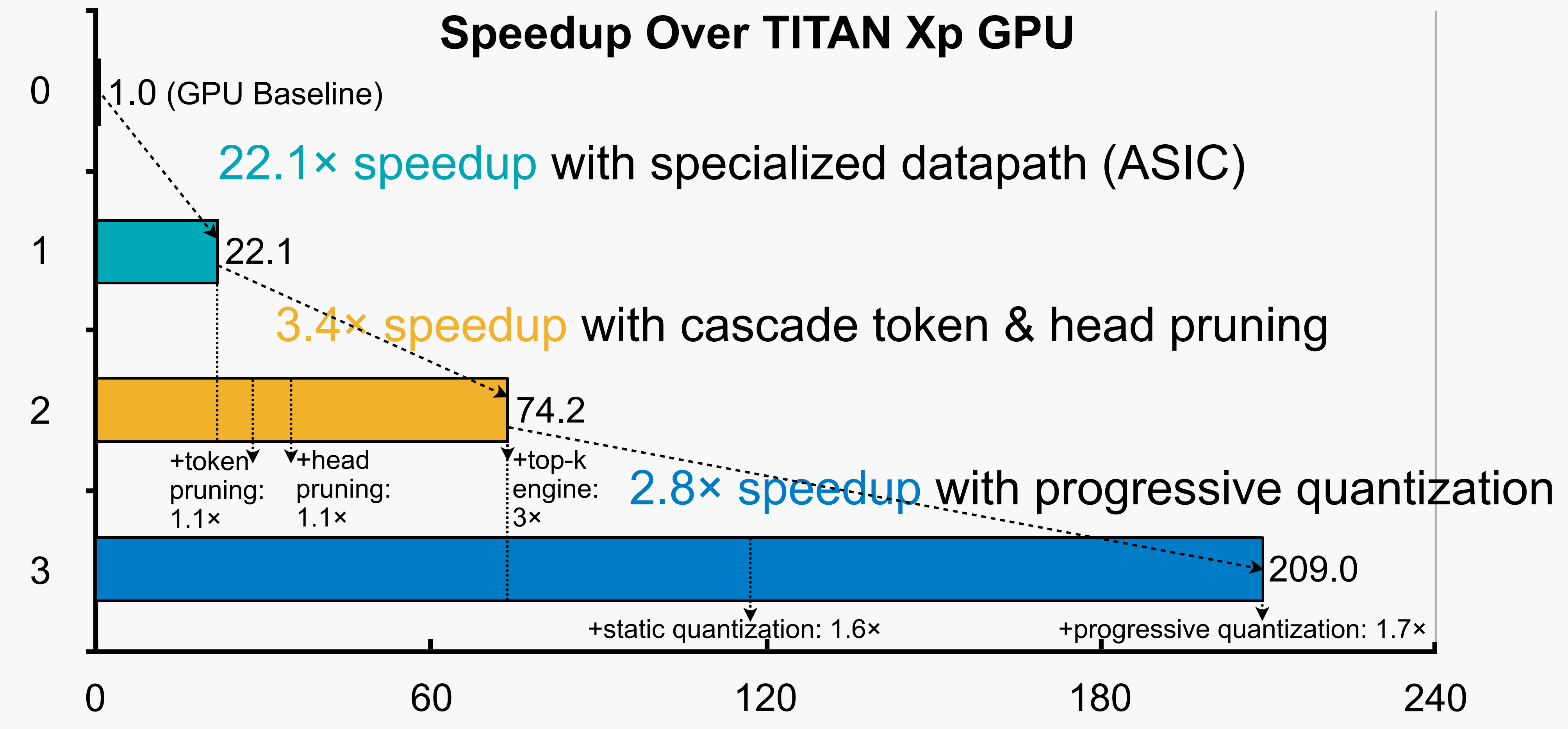
Progressive Quantization: high confident attention requires low bit width



Progressive Quantization: only use MSB when attention confidence is high



Specialized top-k engine to select important token and head

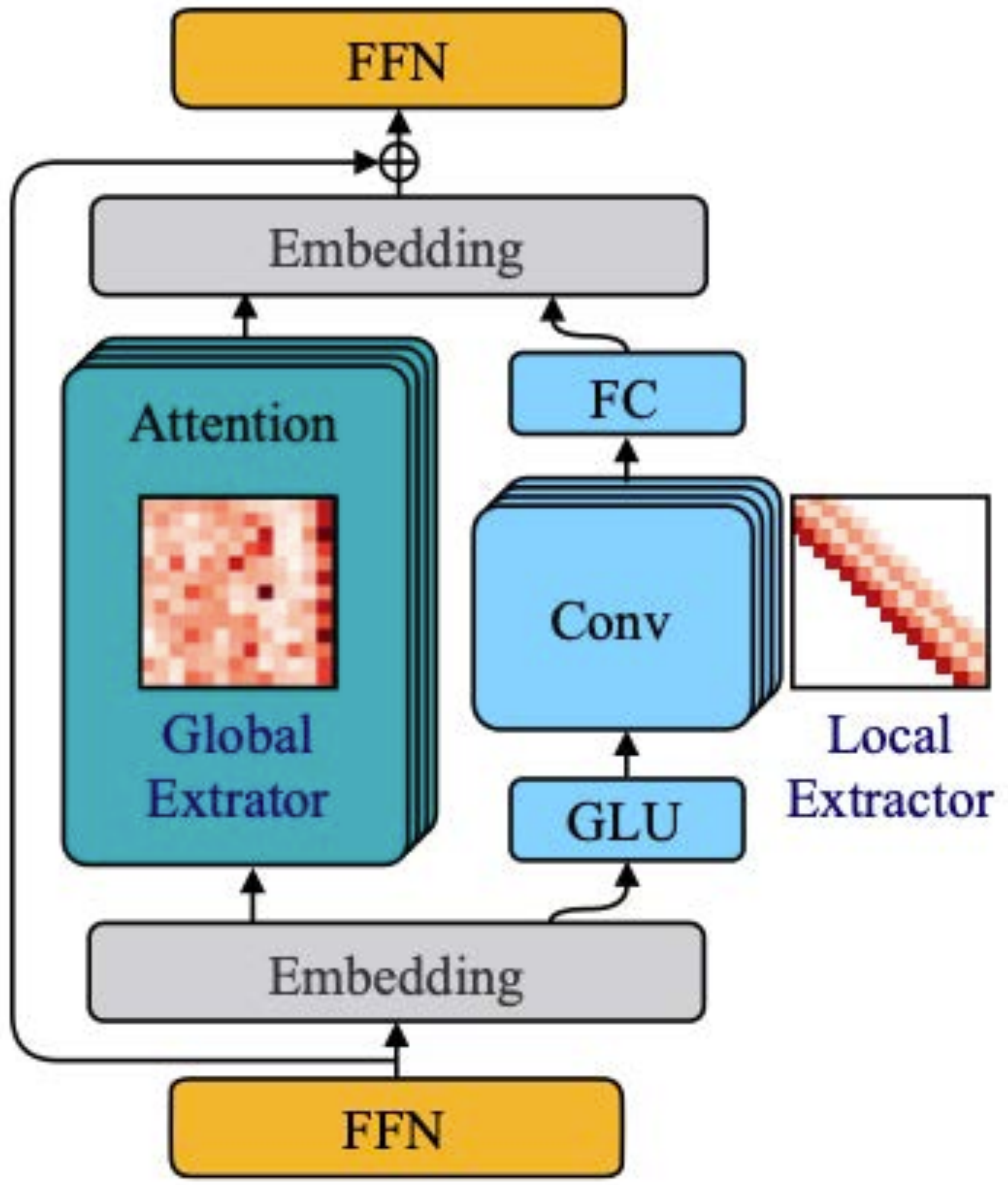


SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning [HPCA'21]

Lite Transformer

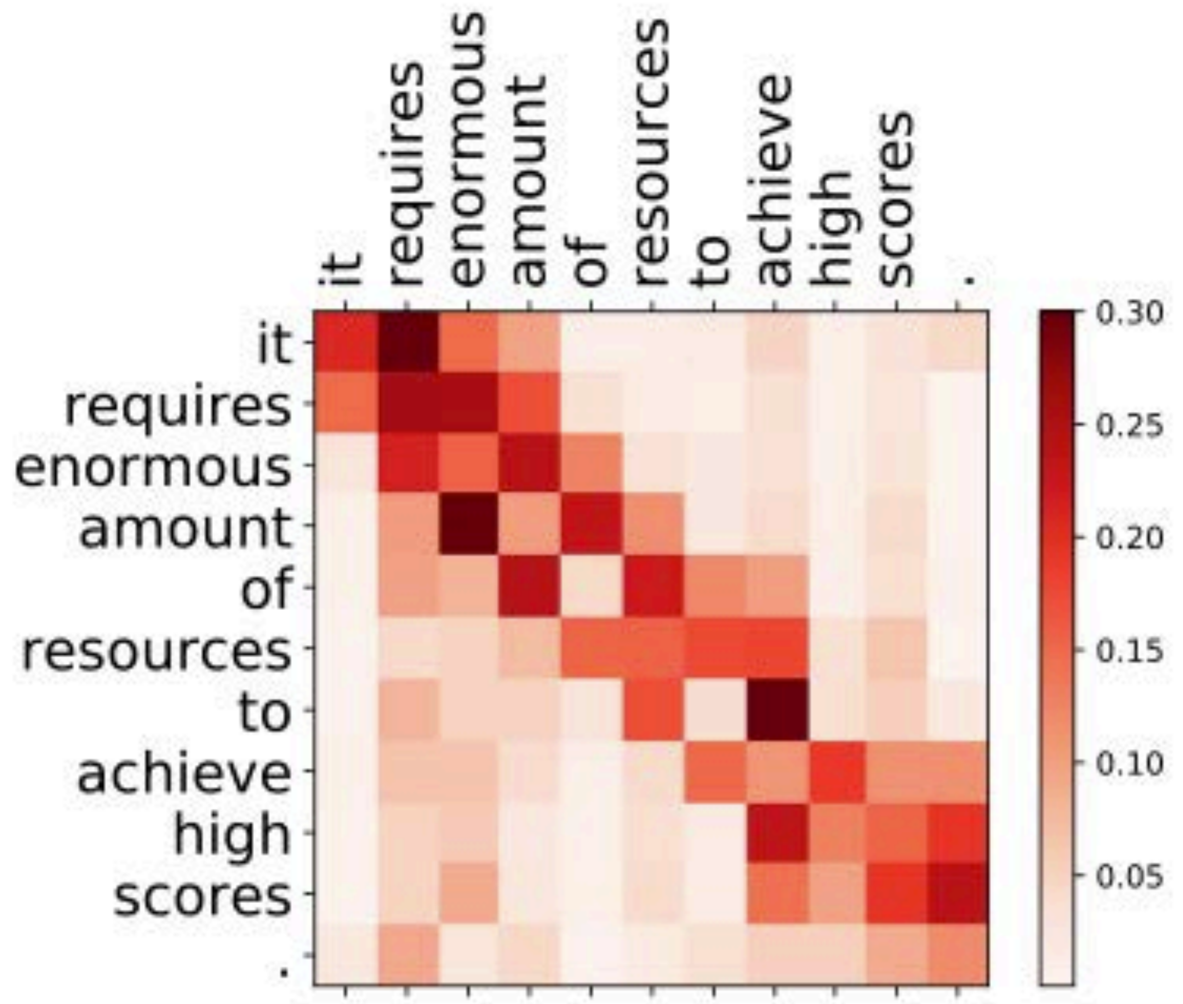
Local Convolution + Global Attention

- Long-Short Range Attention (LSRA):
  - **Convolution**: Efficiently extract the **local** (short-range) features.
  - **Attention**: Tailored for **global** (long-range) feature extraction.



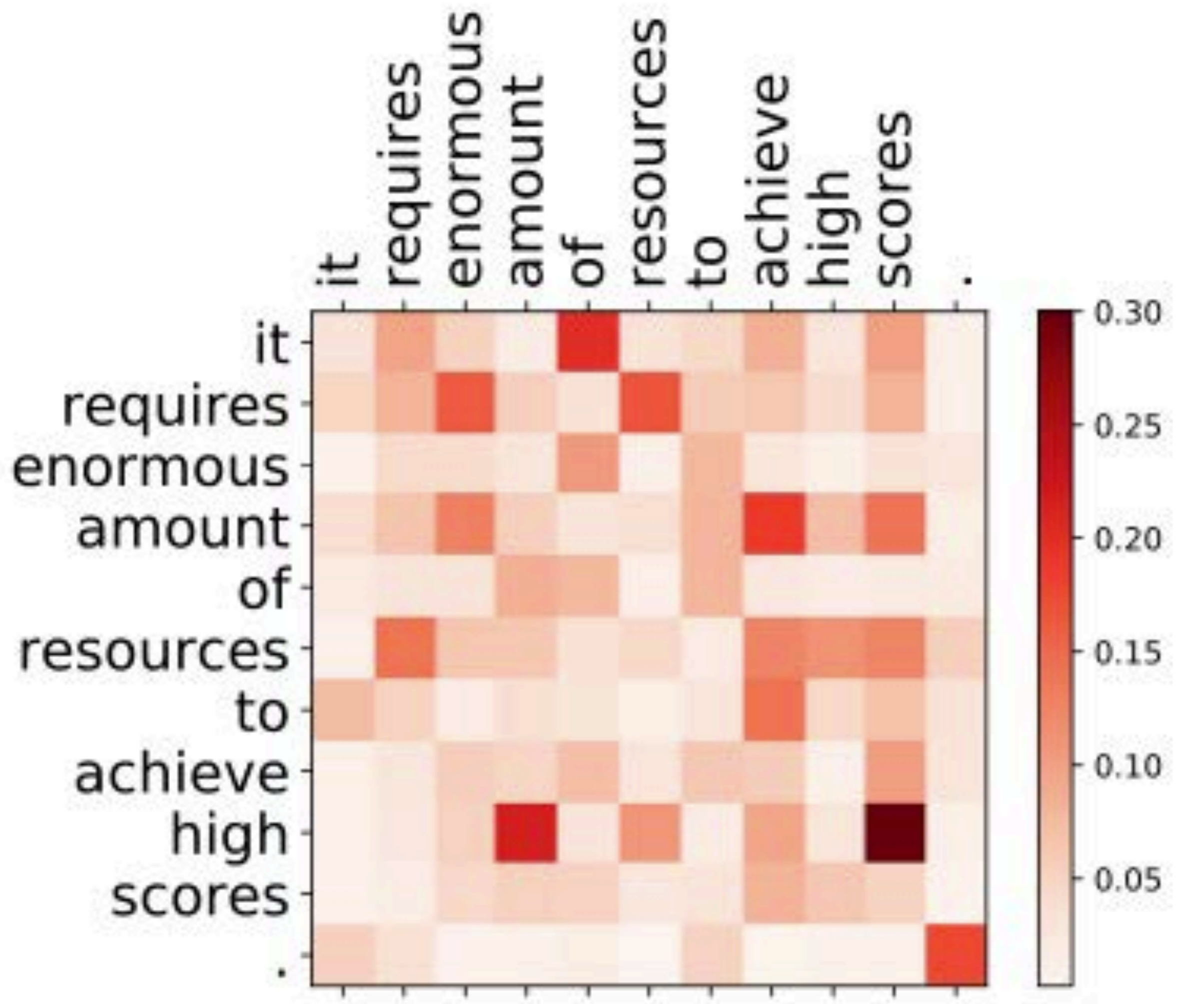
Original Attention

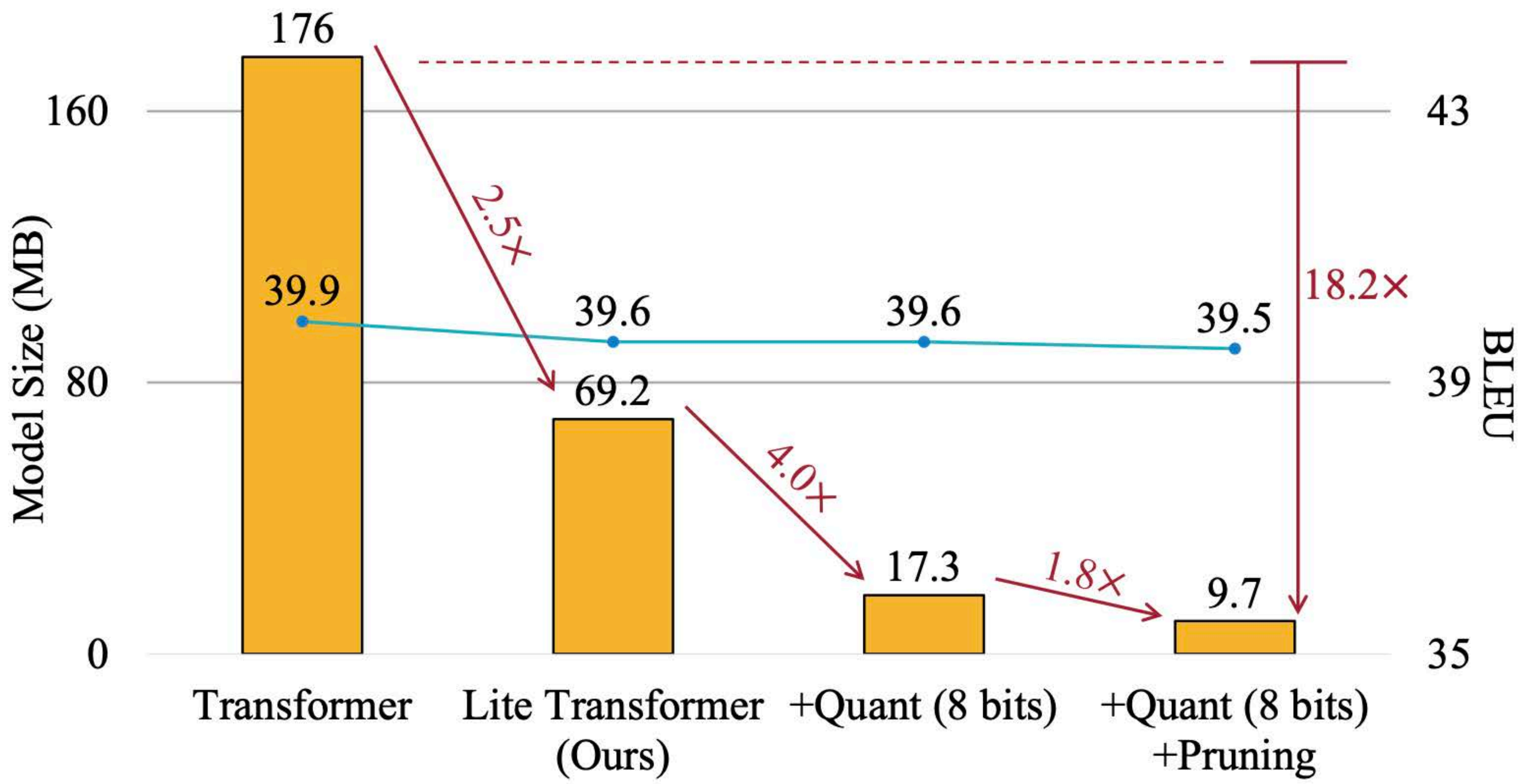
(Too much emphasize on local feature extraction)



Attention in LSRA

(Dedicated for global feature extraction)





Lite Transformer with Long-Short Range Attention [ICLR 2020]

Same Principle, Diverse Applications

Applications

Large Language Model

Generative AI

Advanced Driver Assistance System

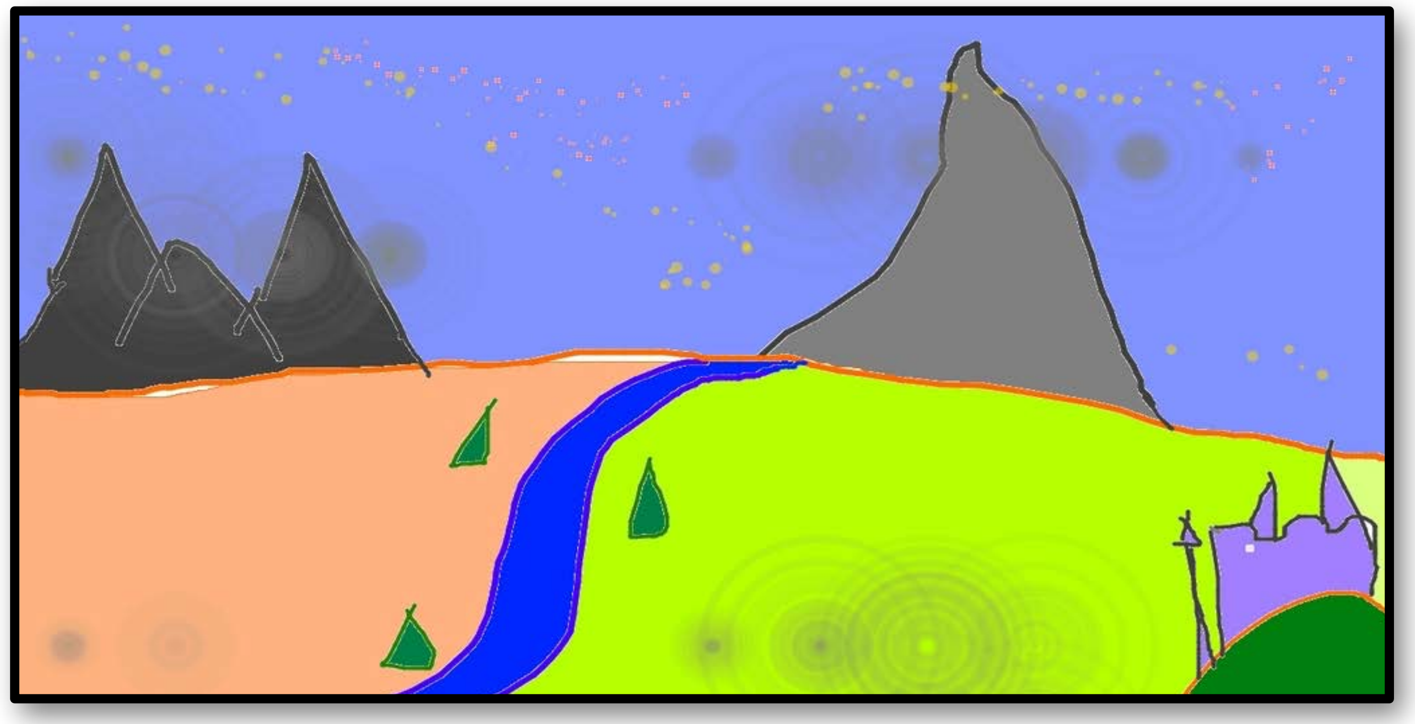
TinyML

Techniques

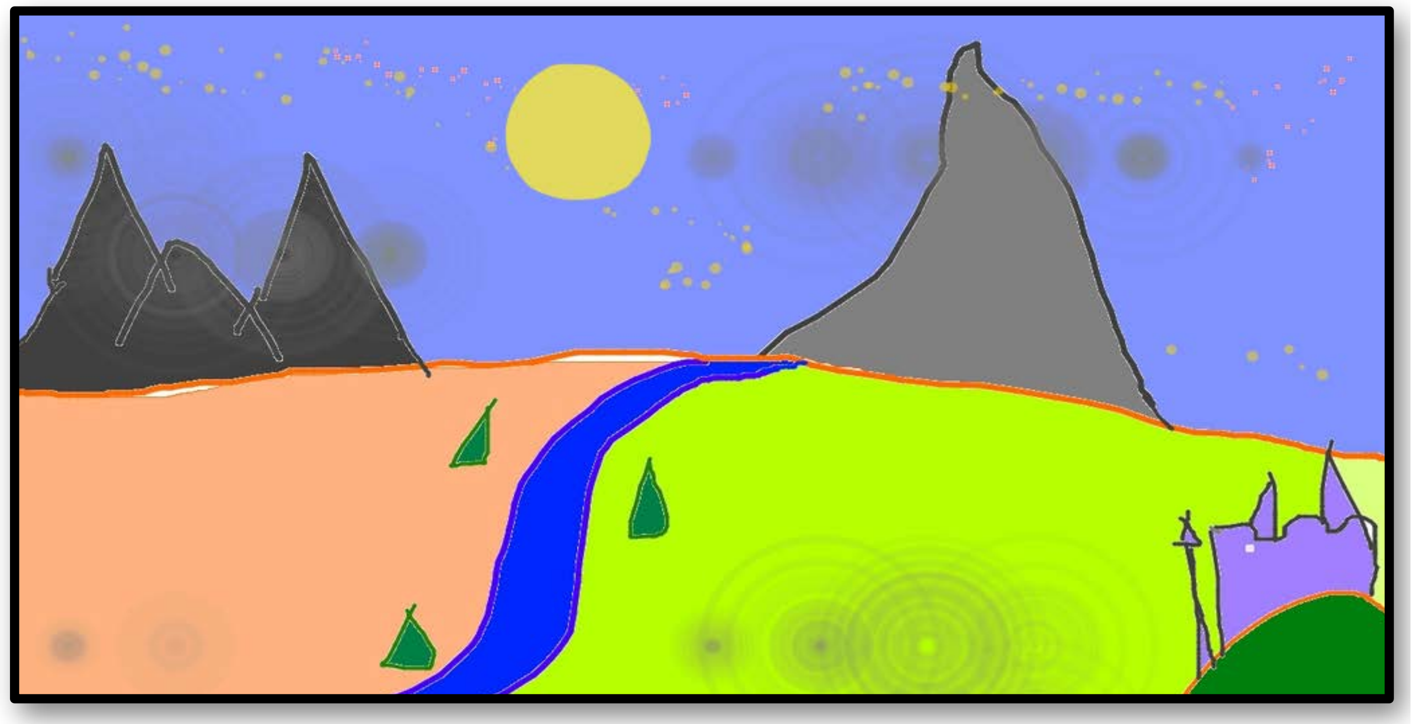
- Hardware-aware NAS
- Pruning & Sparsity
- Quantization
- Distillation
- New Primitive

Compressing and Accelerating Diffusion Models

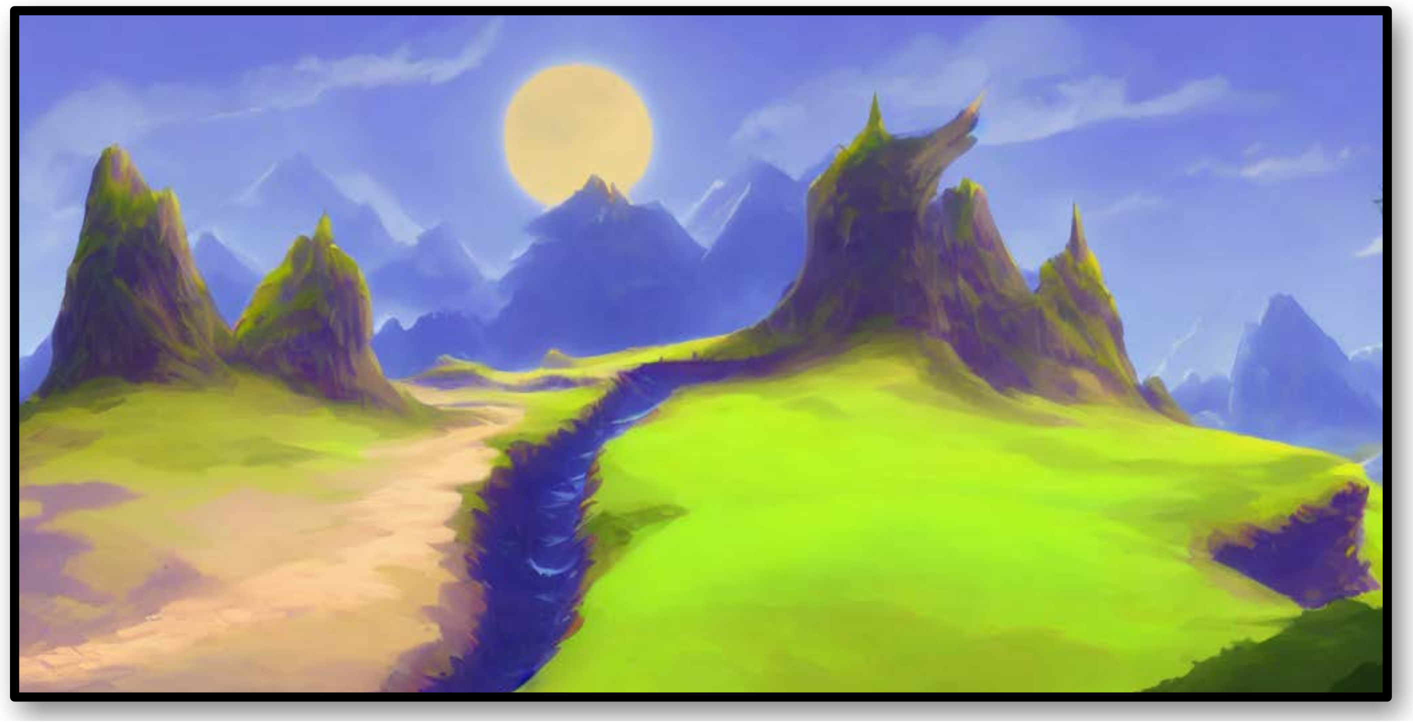
Prompt: A fantasy landscape, trending on artstation.



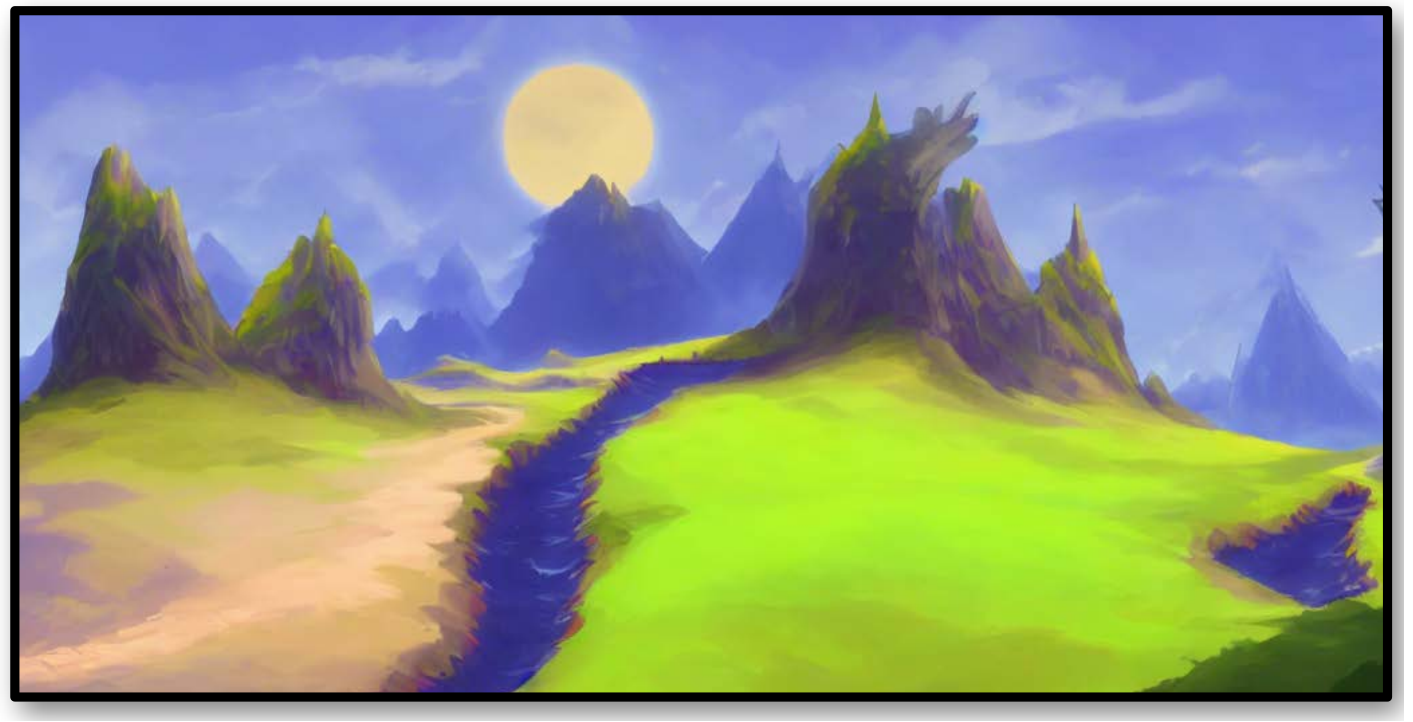
Original



Edited



Stable Diffusion+SDEdit:
1855GMACs, 369ms

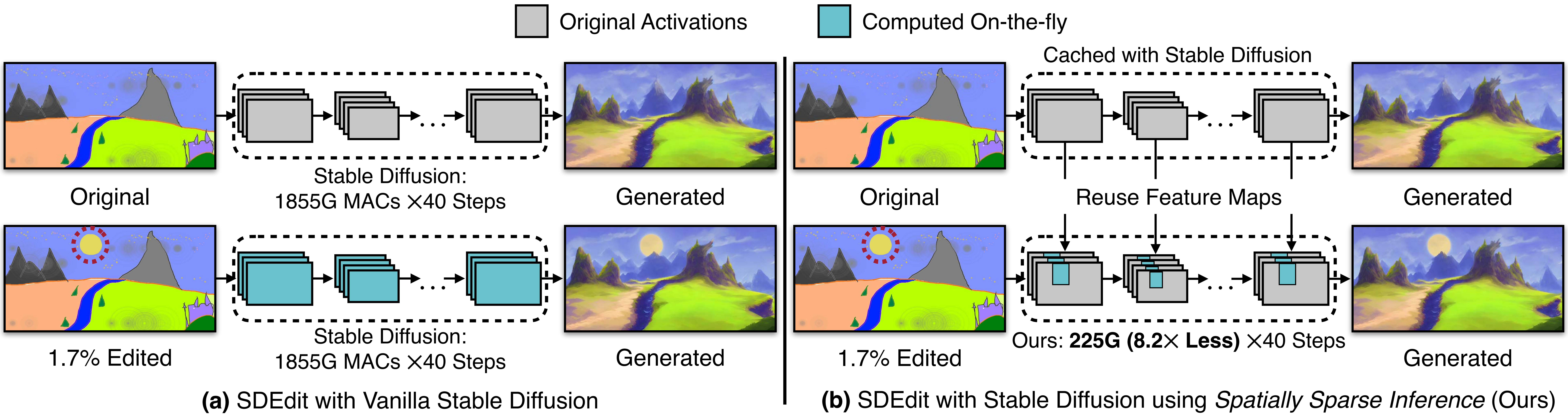


Ours:
225GMACs (8.2×),
51.2ms (7.2x)

Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models, NeurIPS'22

Spatially Sparse Inference for Diffusion Models

Vanilla Model Wastes Many Computations to Re-synthesize the Entire Image

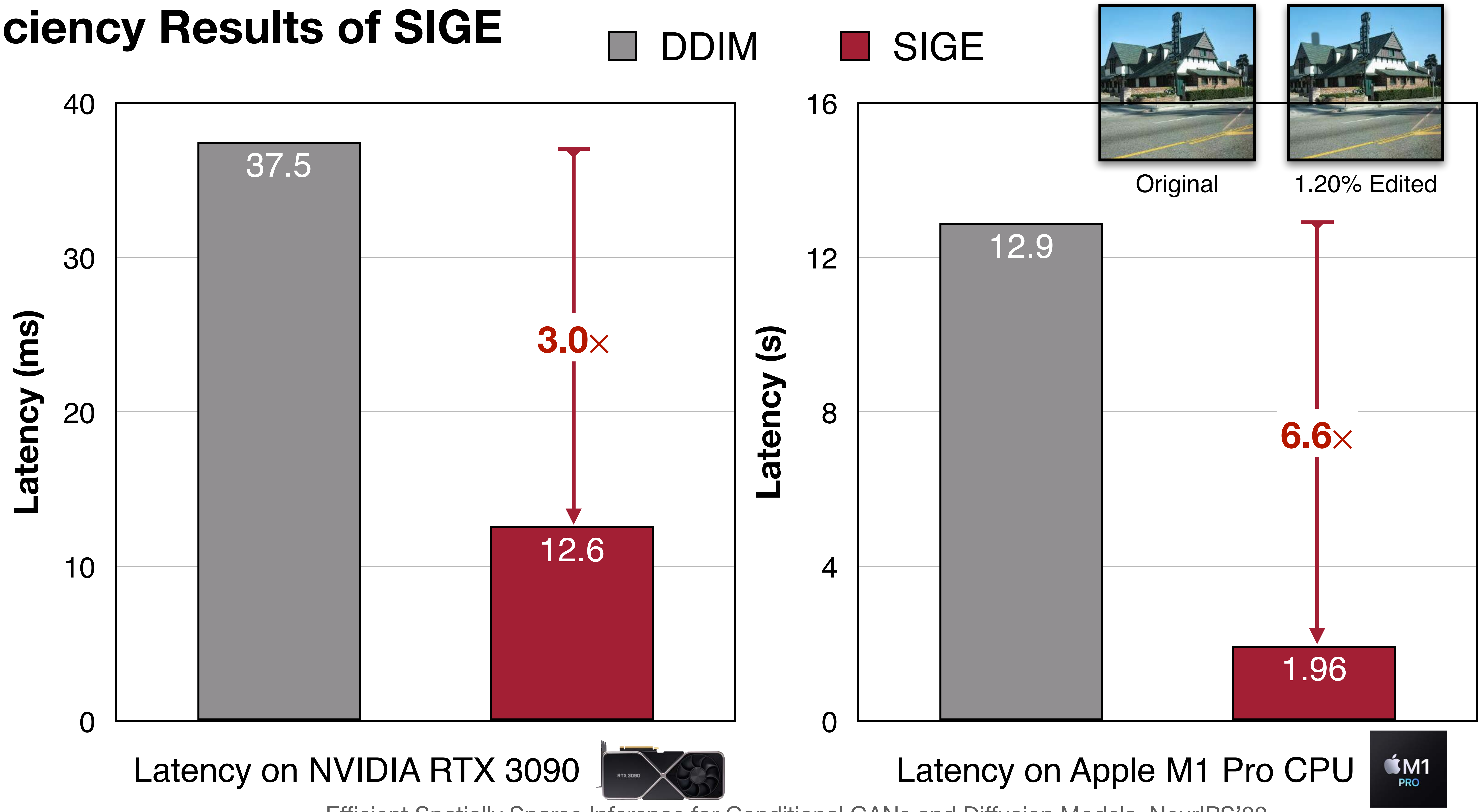


- Only 1.7% region is edited, but vanilla model re-synthesizes the entire image.
- Feature maps remain mostly the same at unedited regions.
- Reuse cached activations to selectively update edited regions (8× less computation).

Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models, NeurIPS'22

Spatially Sparse Inference for Diffusion Models

Efficiency Results of SIGE



Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models, NeurIPS'22

Spatially Sparse Inference for Diffusion Models

Qualitative Results of SIGE on Stable Diffusion

A photograph of a horse on a grassland.

Original

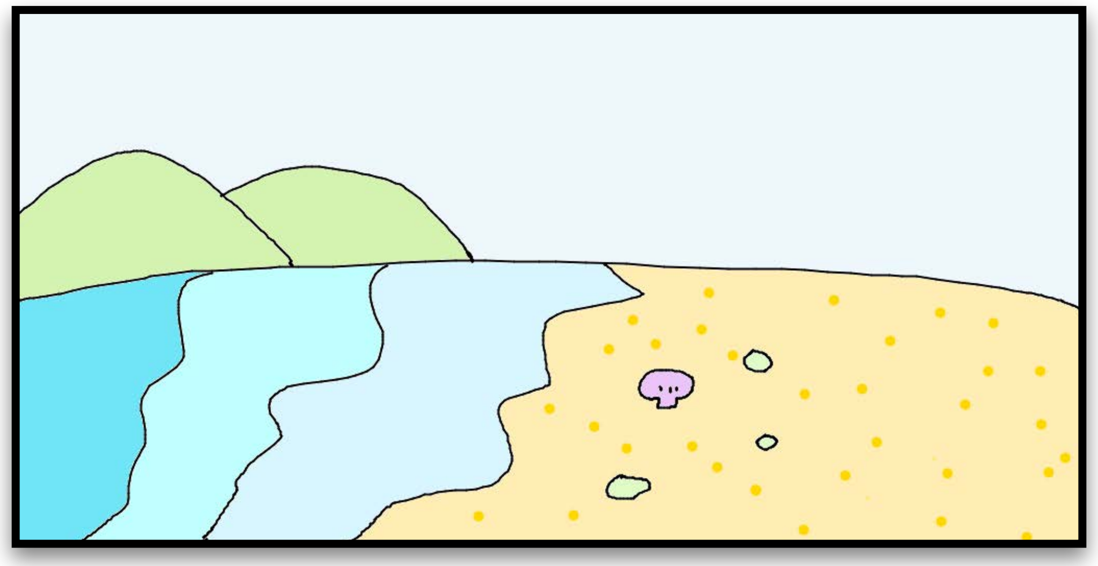
11.6% Masked

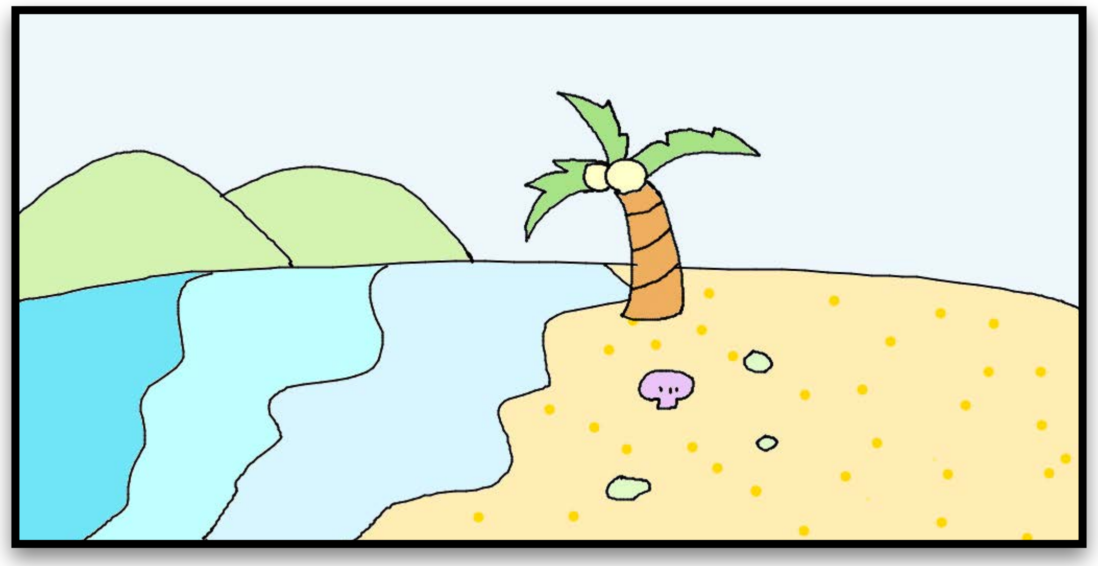
Stable Diffusion:
1855GMACs 369ms

Ours:
514G (3.6×) 95.0ms (3.9×)

Image Inpainting

A fantasy beach landscape, trending on artstation.





Original

2.9% Edited

Stable Diffusion+SDEdit:
1855GMACs 369ms

Ours:
353G (5.3×) 76.4ms (4.8×)

Image Editing

Latency Measured on NVIDIA RTX 3090 

Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models, NeurIPS'22

Same Principle, Diverse Applications

Applications

Large Language Model

Generative AI

Advanced Driver Assistance System

TinyML

Techniques

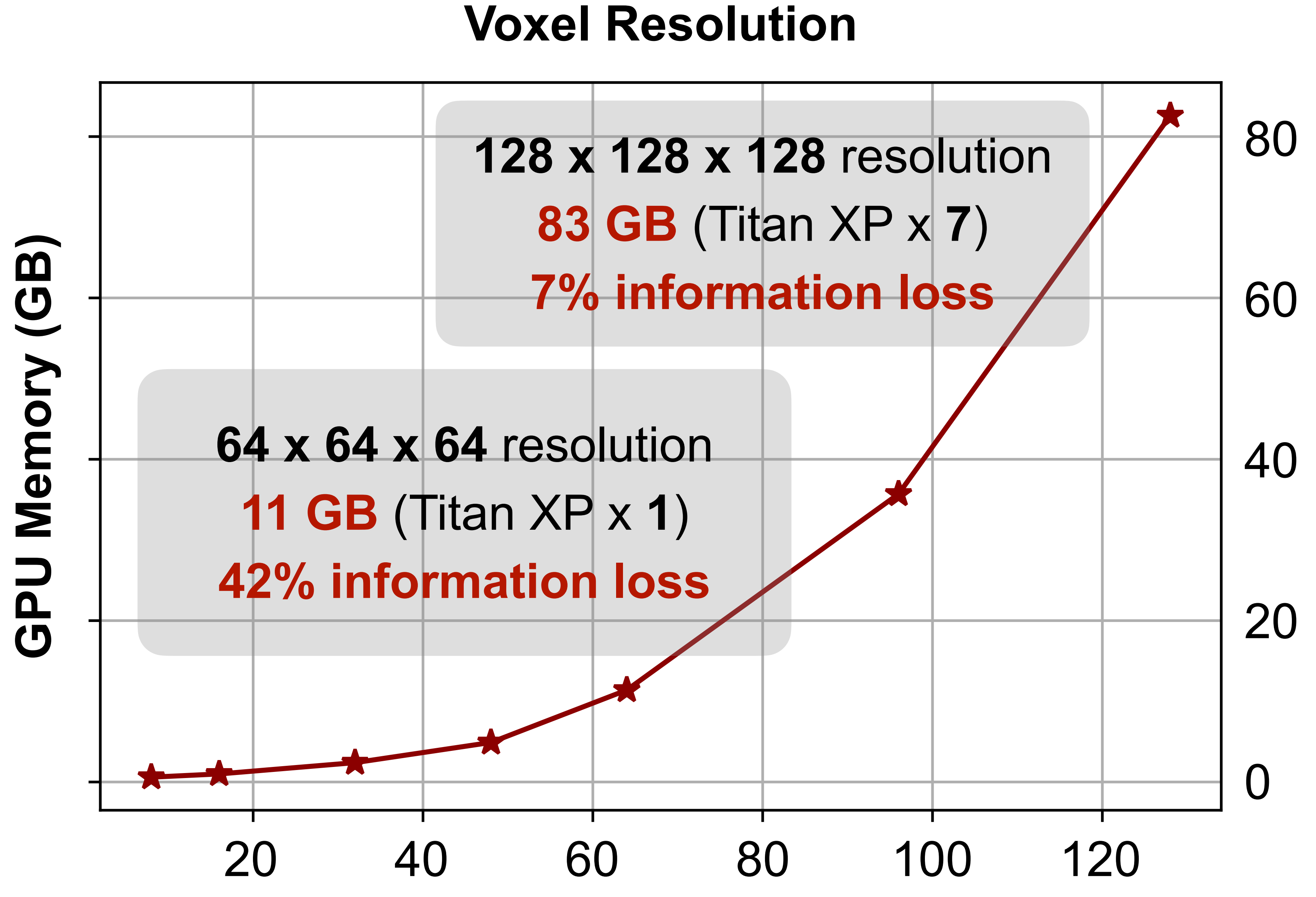
- Hardware-aware NAS
- Pruning & Sparsity
- Quantization
- Distillation
- New Primitive

PVCNN: Point-Voxel CNN

New primitive for handling spatial sparsity in point clouds

Voxel-Based Models

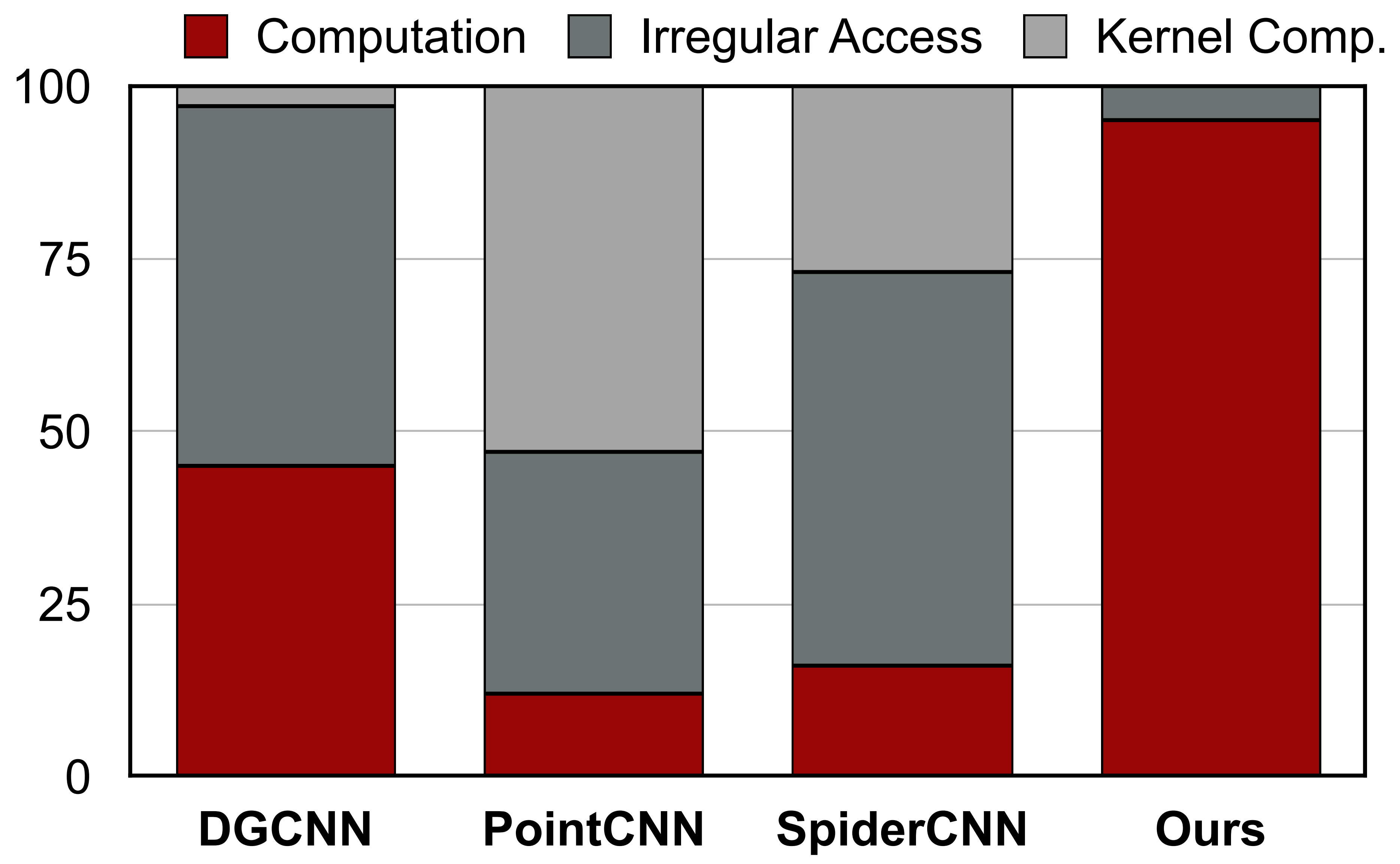
3D ShapeNets [CVPR'15]
VoxNet [IROS'15]
3D U-Net [MICCAI'16]



GPU memory consumption **increases cubically** with the volumetric resolution

Point-Based Models

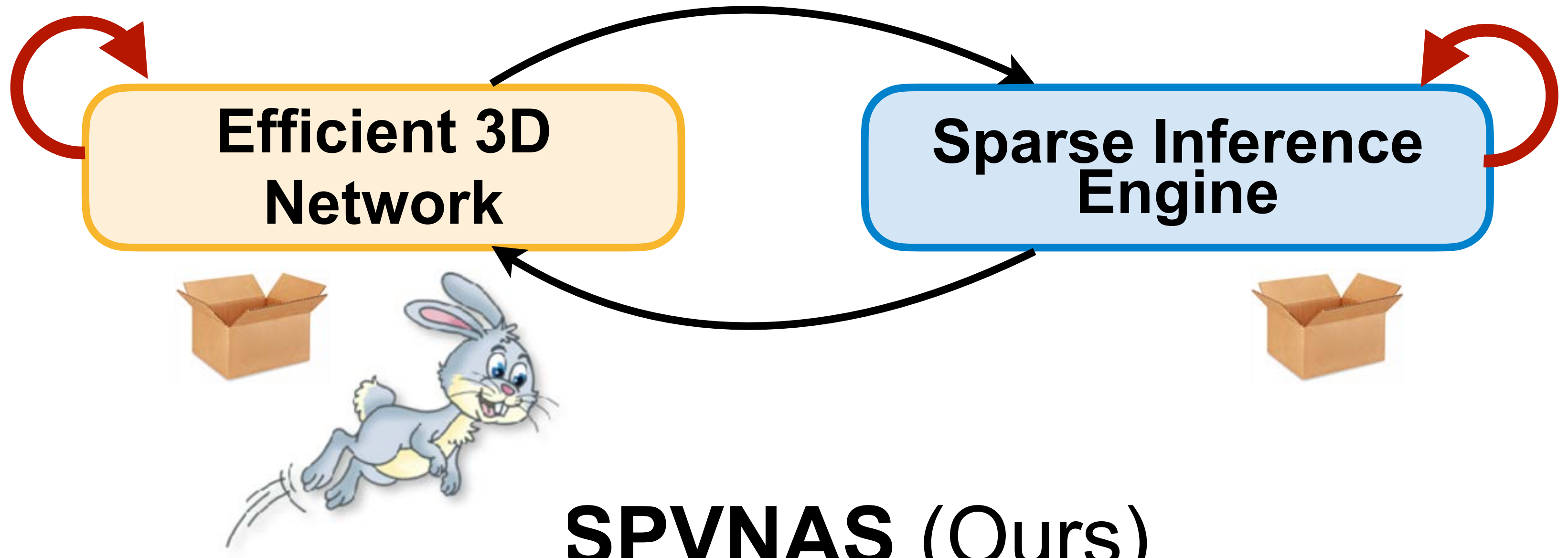
PointNet [CVPR'17]
PointCNN [NeurIPS'18]
DGCNN [SIGGRAPH'19]

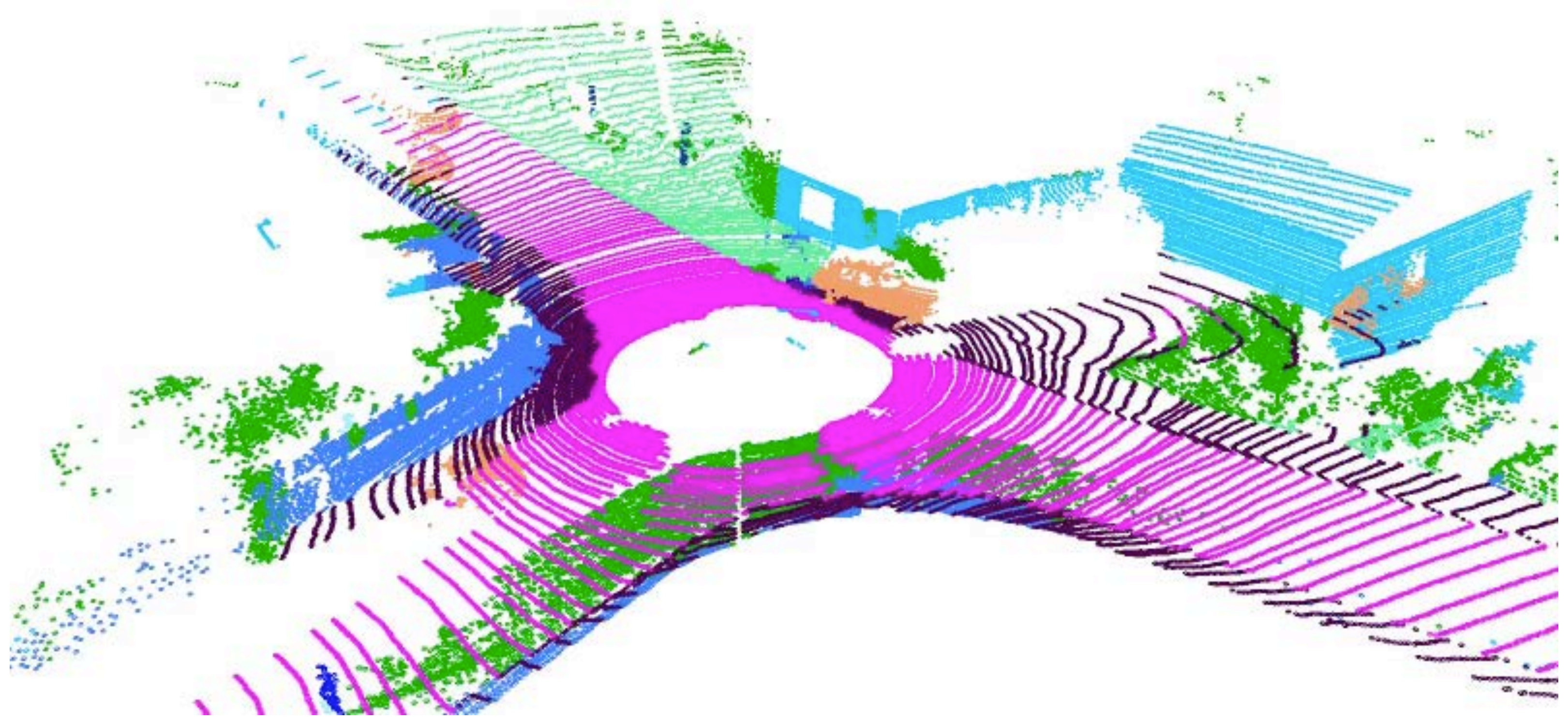
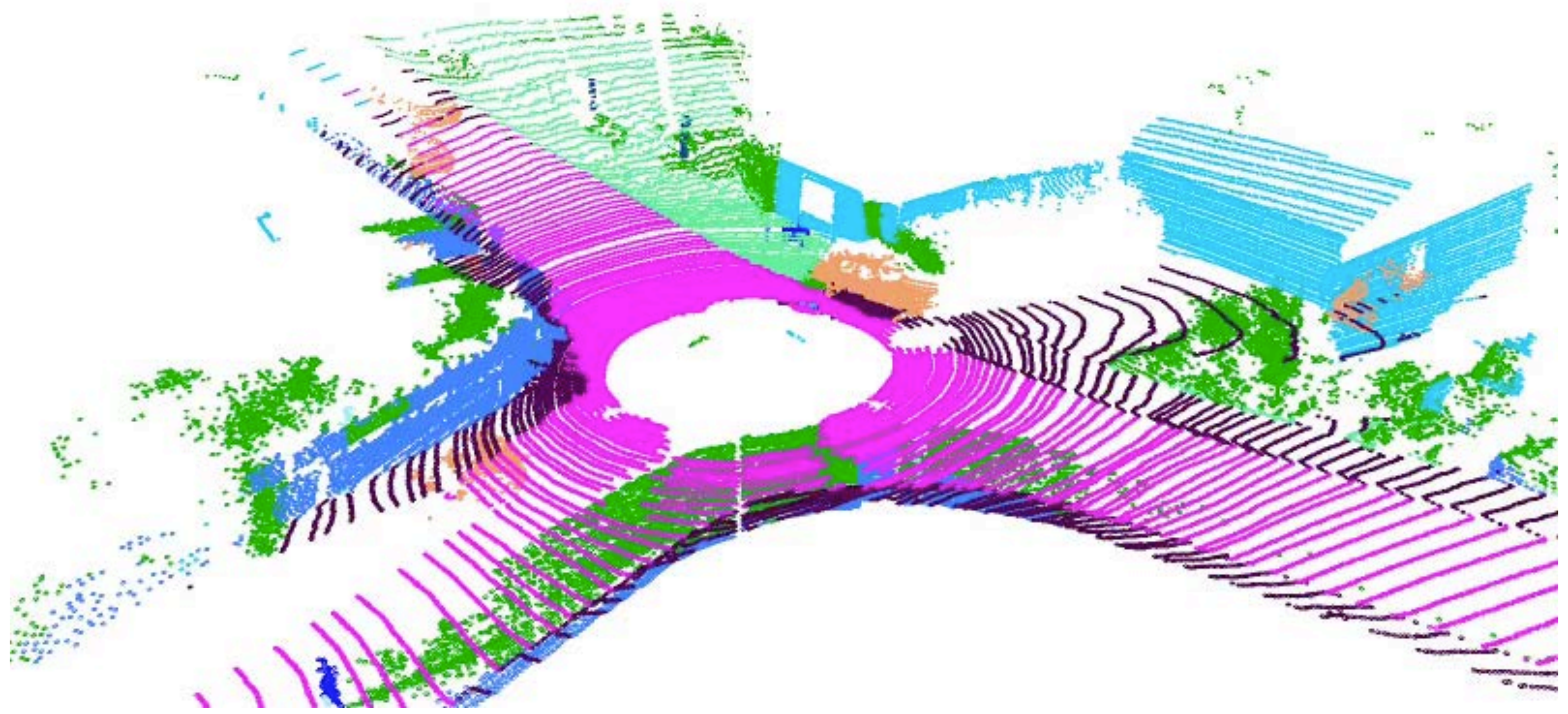
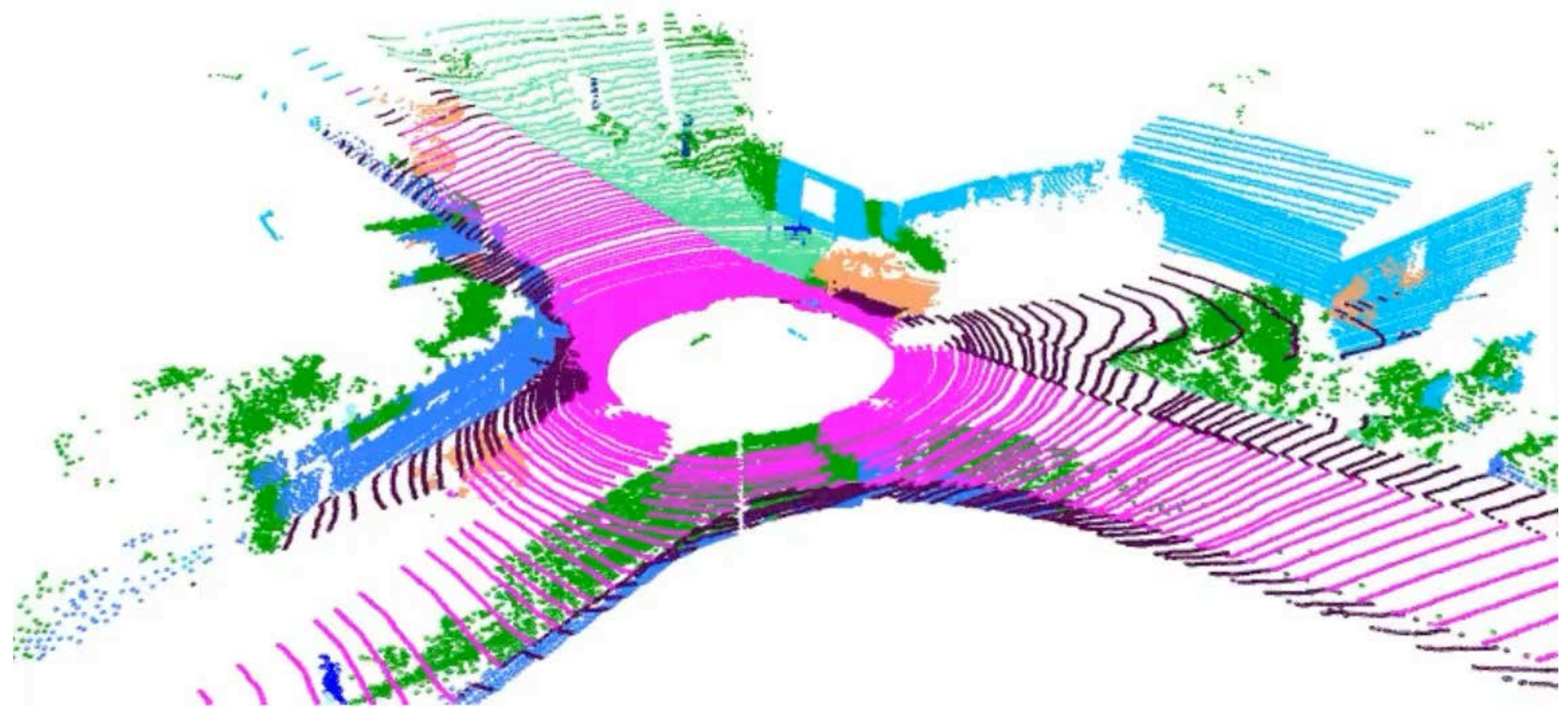


Point-based models suffer from **random memory accesses** and **dynamic kernel computation**

SPVNAS accelerated by TorchSparse

TorchSparse brings about another 1.3x speedup to the efficient SPVNAS model



MinkowskiNet	**SPVNAS** (Ours)	**SPVNAS + TorchSparse** (Ours)
		
Mean IoU: **63.1** Throughput: **3.4** FPS (**21.7M** Params **114.0G** FLOPs)	Mean IoU: **63.6** Throughput: **9.1** FPS (**2.6M** Params **15.0G** FLOPs)	Mean IoU: **63.6** Throughput: **12.1** FPS (**2.6M** Params **15.0G** FLOPs)

Measured on GTX1080Ti

TorchSparse: Efficient Point Cloud Library

Balancing computation regularity and computation overhead

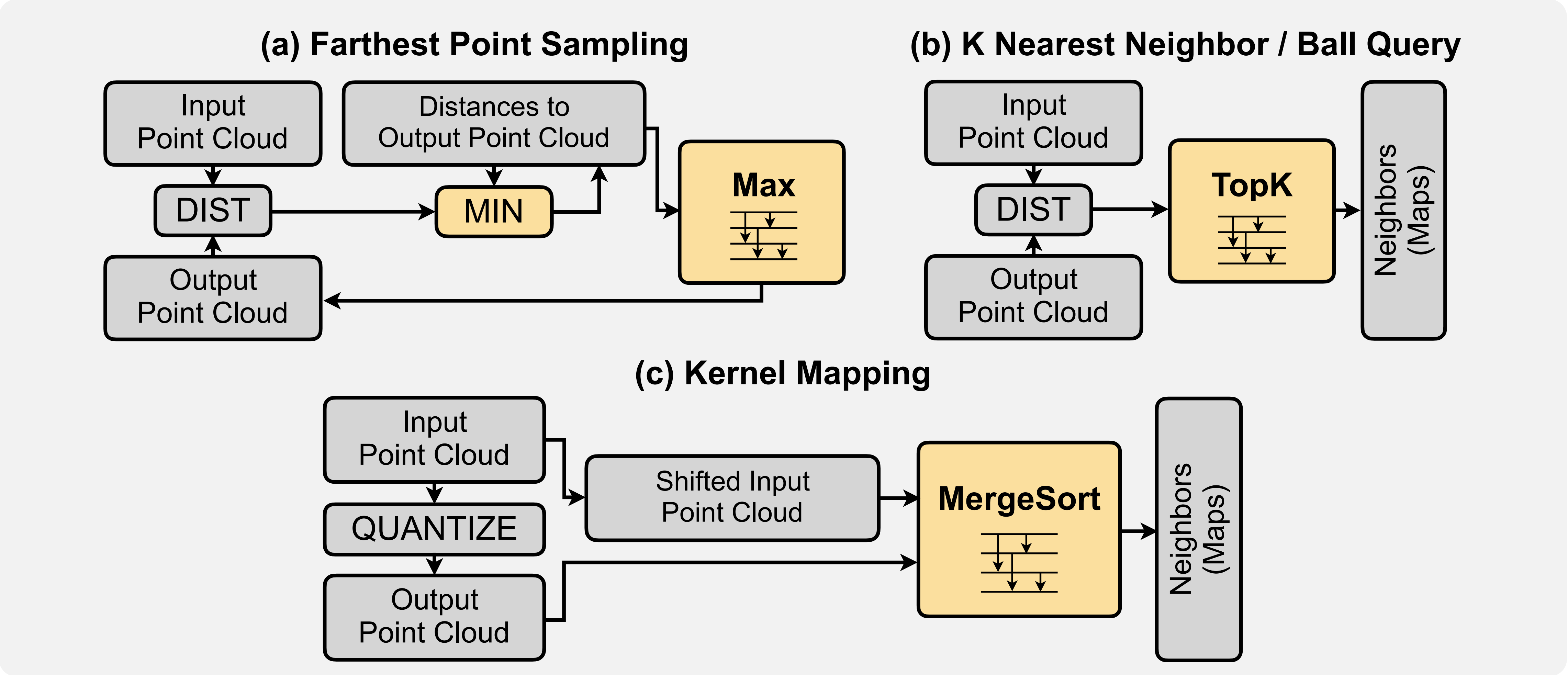
- Automatically determine the best tradeoff between **control flow** and **computation** overhead;
- **Mixed dataflow configurations** for different layers and forward/backward computation.

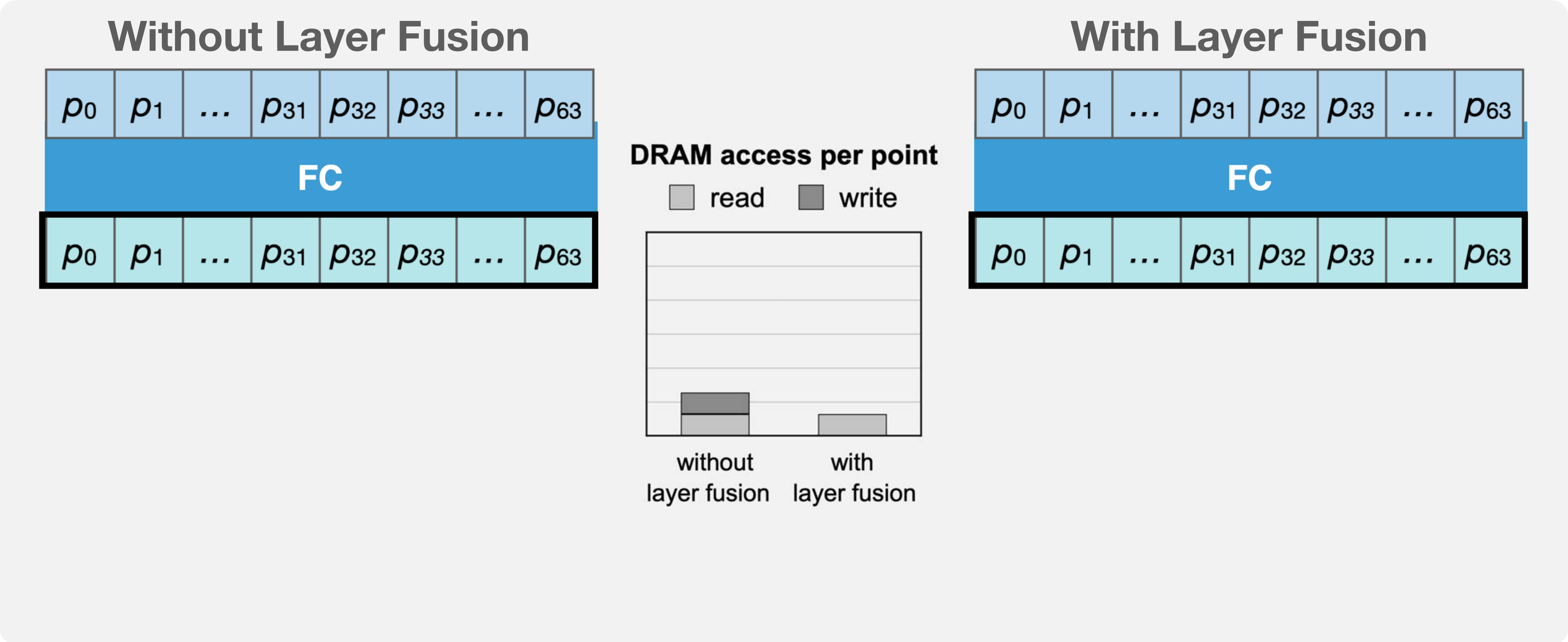
PointAcc: Point Cloud Accelerator

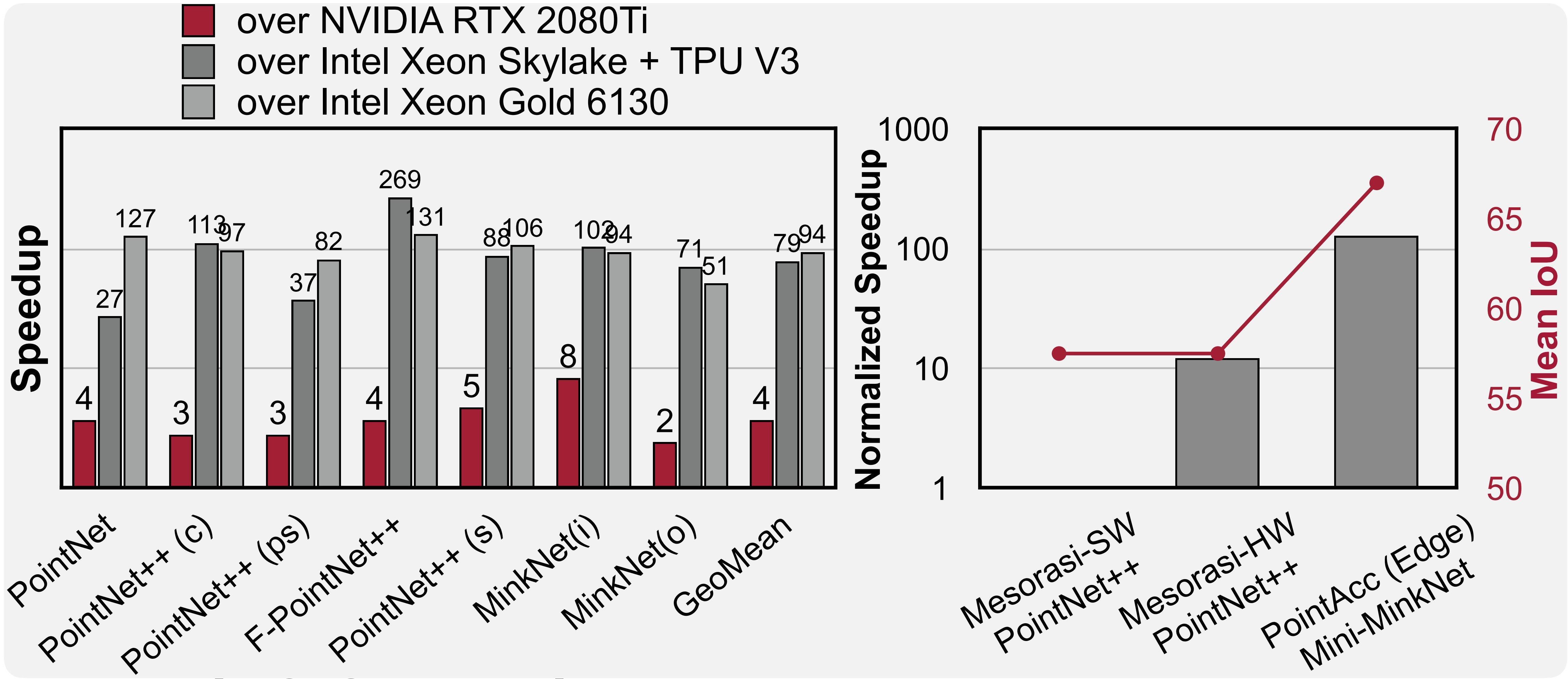
One hardware architecture, diverse neural architectures

- The sparsity of point clouds leads to two bottlenecks:
  - ***Diverse mapping operations*** for searching the input, output, weight maps (e.g., k-Nearest Neighbor, Ball Query, Kernel Mapping)
  - Data movement overhead from ***gather and scatter*** of the sparse features
- PointAcc maps diverse mapping ops into sort-based computation with ***one versatile hardware architecture***.
- PointAcc reduces off-chip memory access and minimize the overhead of gather and scatter by ***flexible caching*** and ***layer fusion***.

Diverse Mapping Ops in One Versatile Architecture



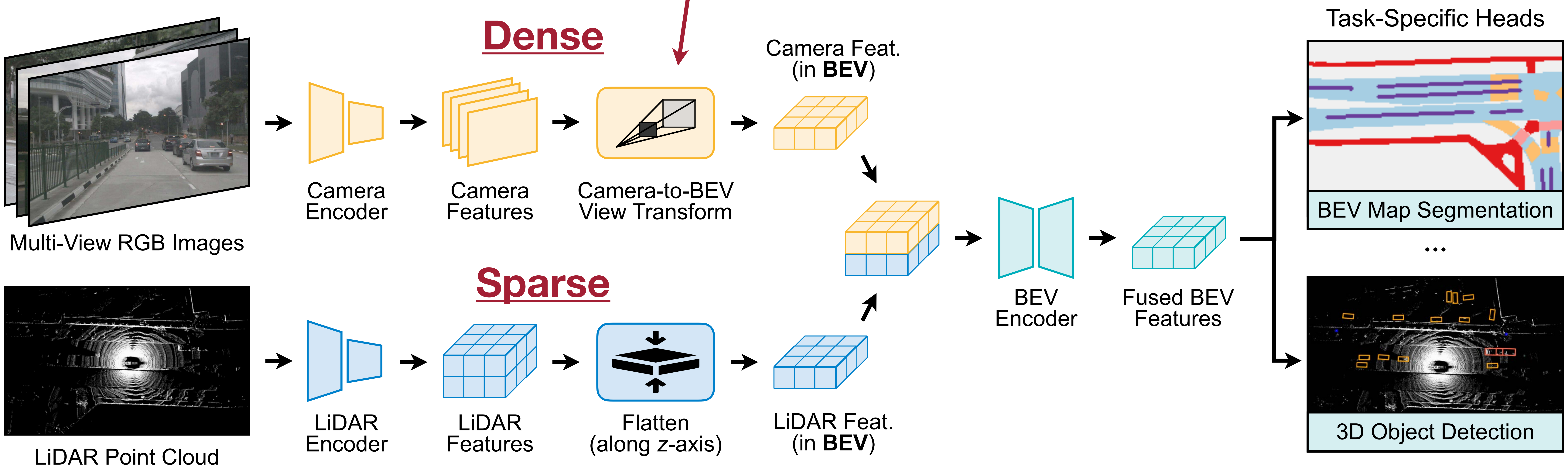


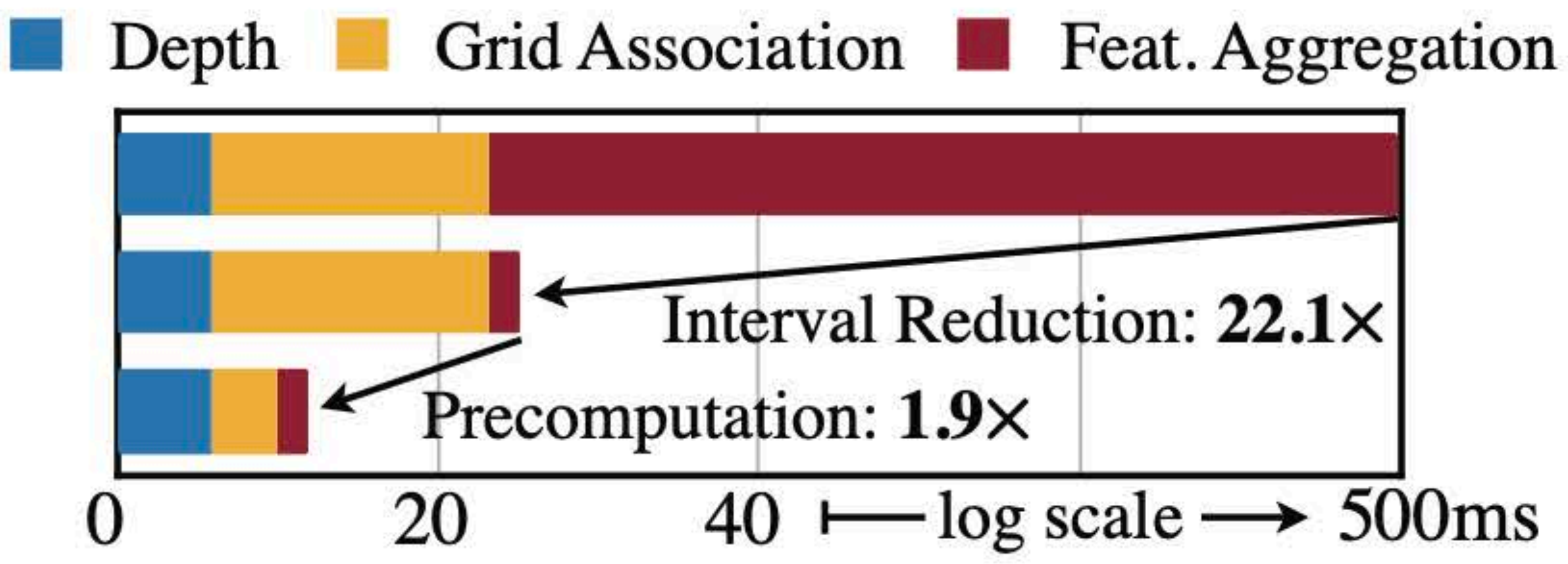


PointAcc: Efficient Point Cloud Accelerator [MICRO-54, 2021]

BEVFusion: Dense (Camera) + Sparse (LiDAR)

Accelerate LSS by 40x with Interval Reduction and Pre-computation



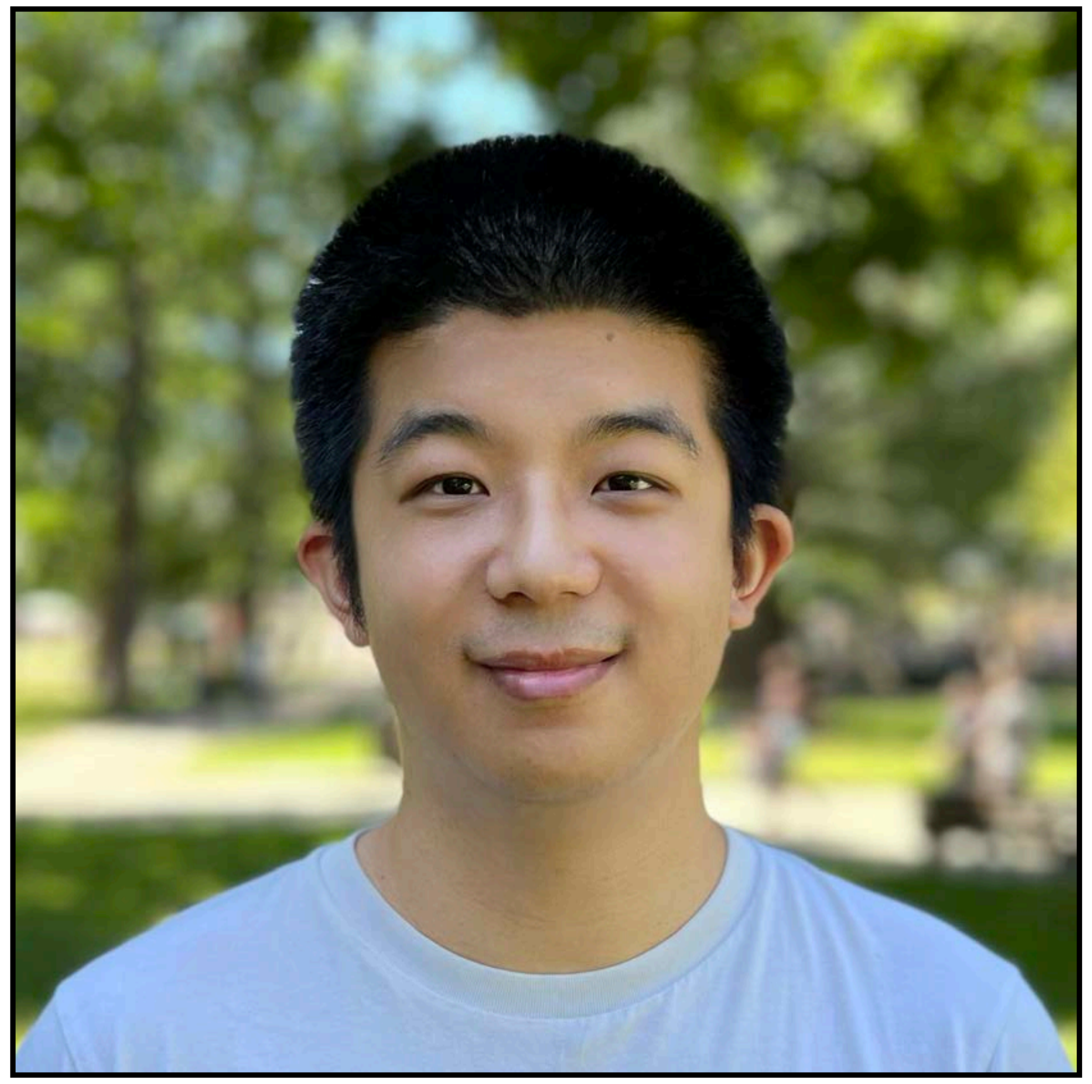


- Ranked **first** on **nuScenes 3D object detection** (2022/6).
- Ranked **first** on **nuScenes 3D object tracking** (2022/7).
- Ranked **first** on **Waymo 3D object detection** (2022/11).
- Ranked **first** on **Argoverse 3D object detection** (2023/4).

BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation [ICRA 2023]

PhD Student — Zhijian Liu

Research Interest: **Efficient Algorithms and Systems for Deep Learning**
Graduating in Fall’23



Zhijian Liu

zhijian@mit.edu
https://zhijianliu.com
@zhijianliu96

Representative Work

- Algorithms:
  - **PVCNN** (NeurIPS’19 Spotlight)
  - **SPVNAS** (ECCV’20, TPAMI’21)
  - **FlatFormer** (CVPR’23)
- Systems:
  - **TorchSparse** (MLSys’22)
- Applications:
  - **BEVFusion** (ICRA’23)

Selected Honors

- Qualcomm Innovation Fellowship
- NVIDIA Graduate Fellowship
- MIT Ho-Ching and Han-Ching Fund Award

Same Principle, Diverse Applications

Applications

Large Language Model

Generative AI

Advanced Driver Assistance System

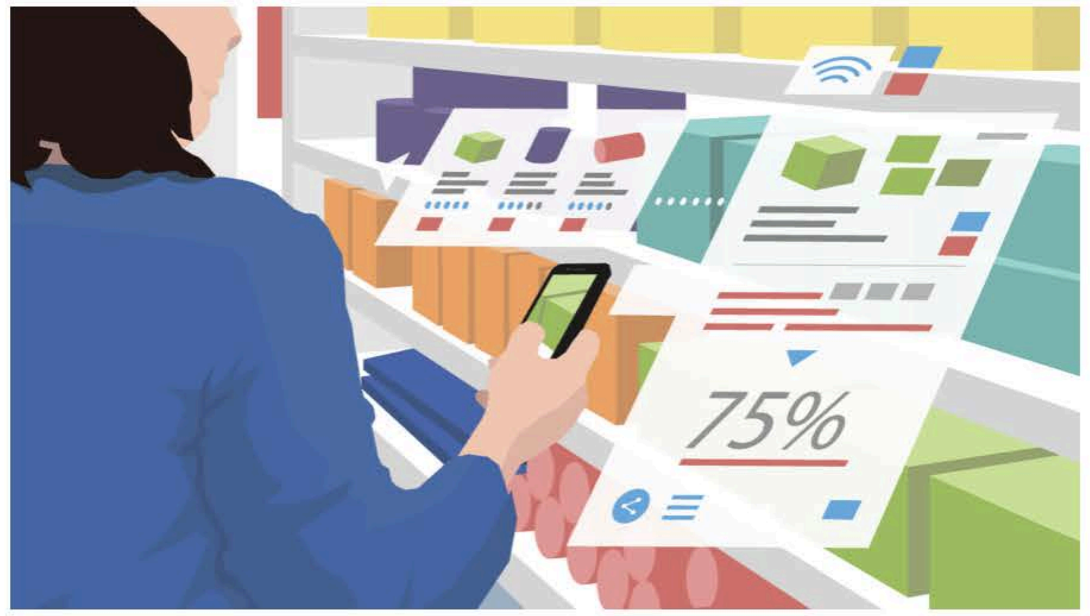
TinyML

Techniques

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

Background: The Era of AIoT on Microcontrollers

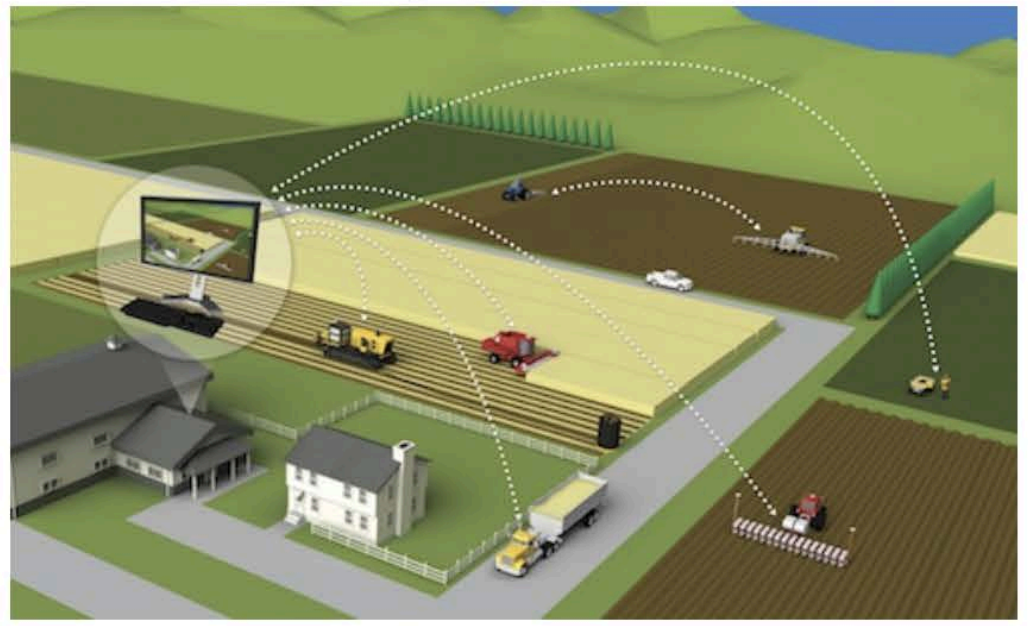
Smart Retail



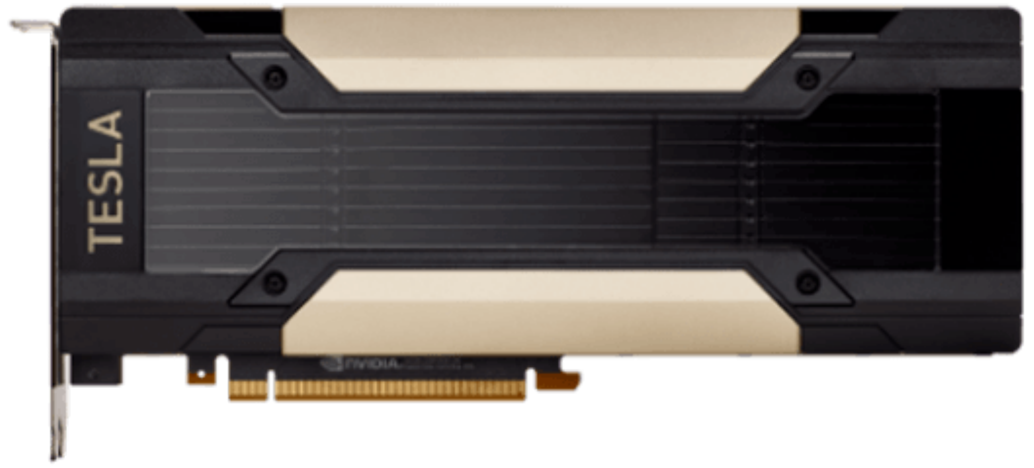
Personalized Healthcare

Smart Home

Precision Agriculture



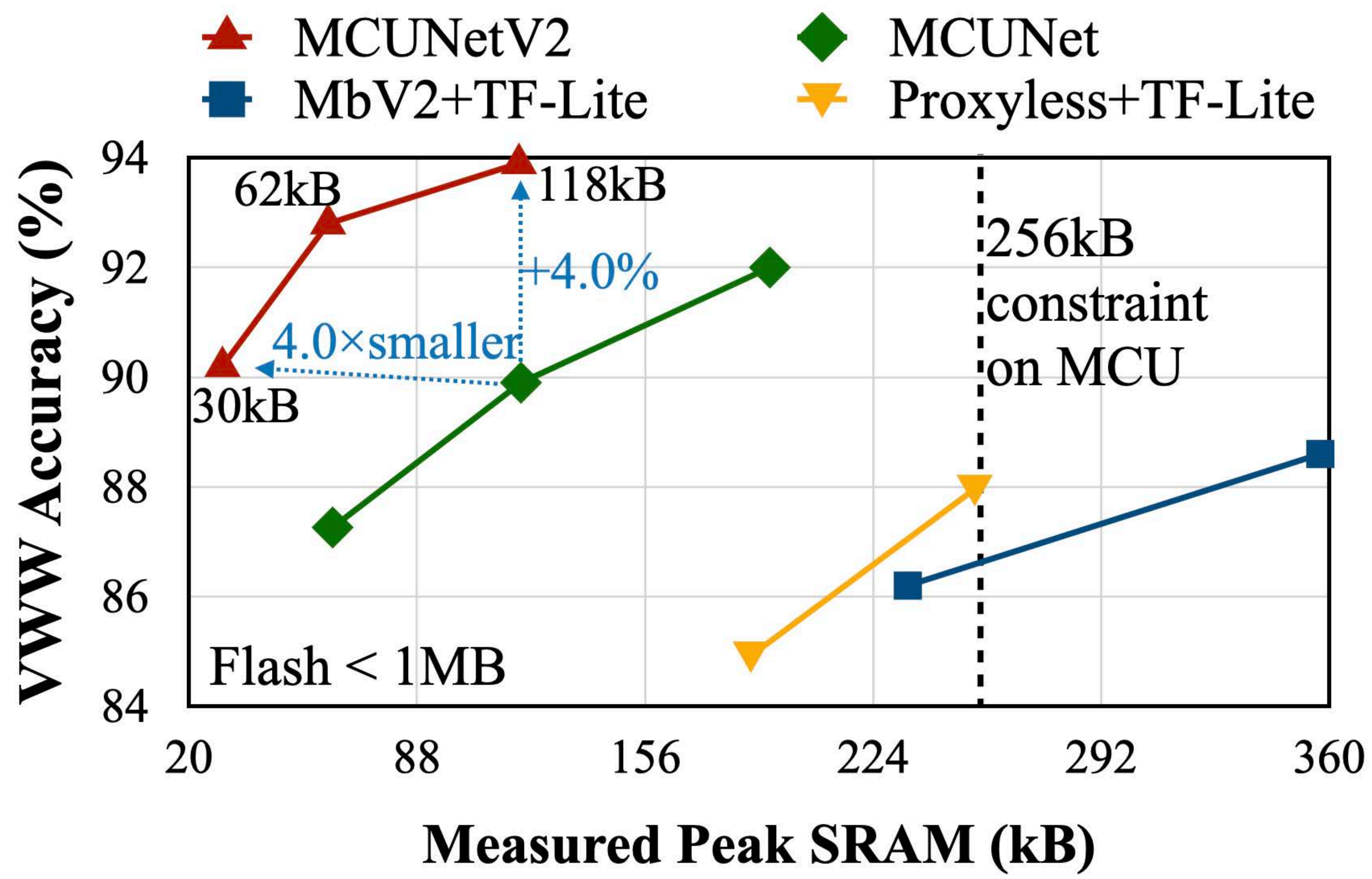
- **Problem**: restricted memory size



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

MCUNet

Deploy AI on MCUs that has only 256KB SRAM



Face/mask detection

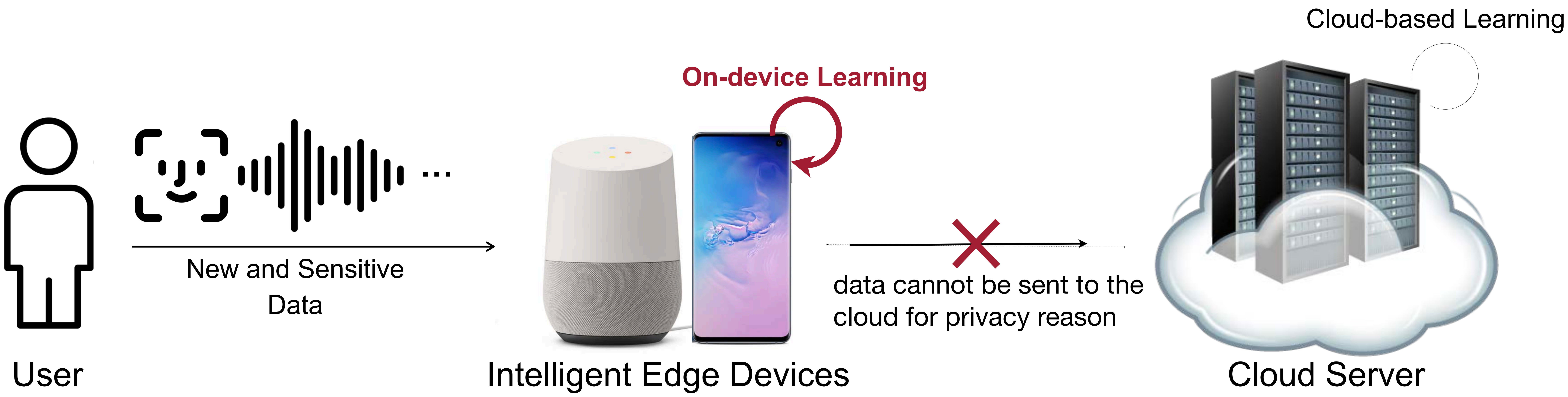
Person detection

The camera is OpenMV Cam.

Inference Is Good. Can We Learn on Edge?

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

Not only inference, but also training



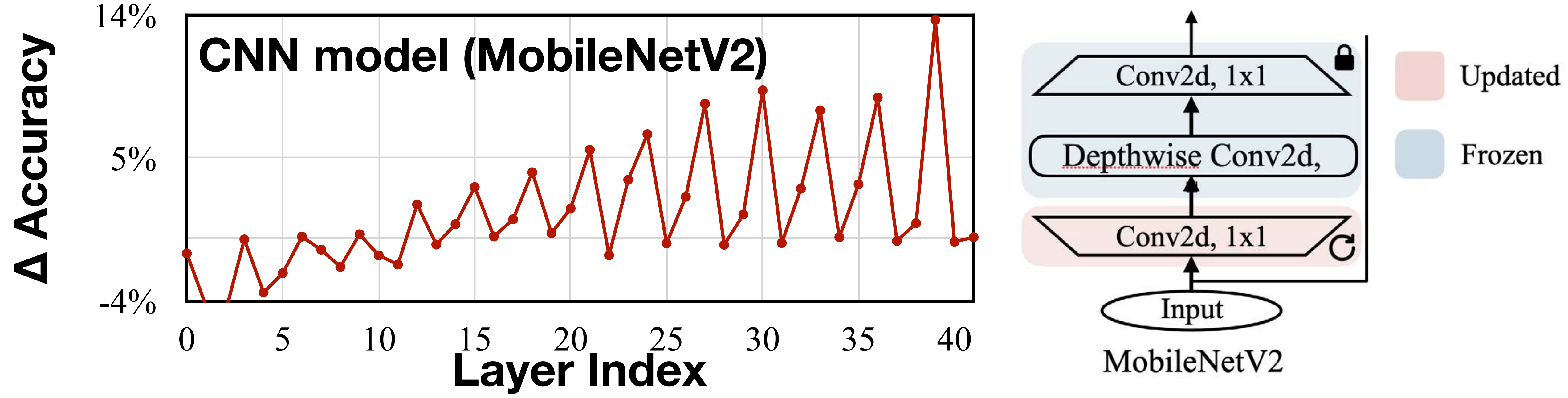
- On-device learning: **better privacy, lower cost, customization, life-long learning**
- Training is more **expensive** than inference, hard to fit edge hardware (limited memory)

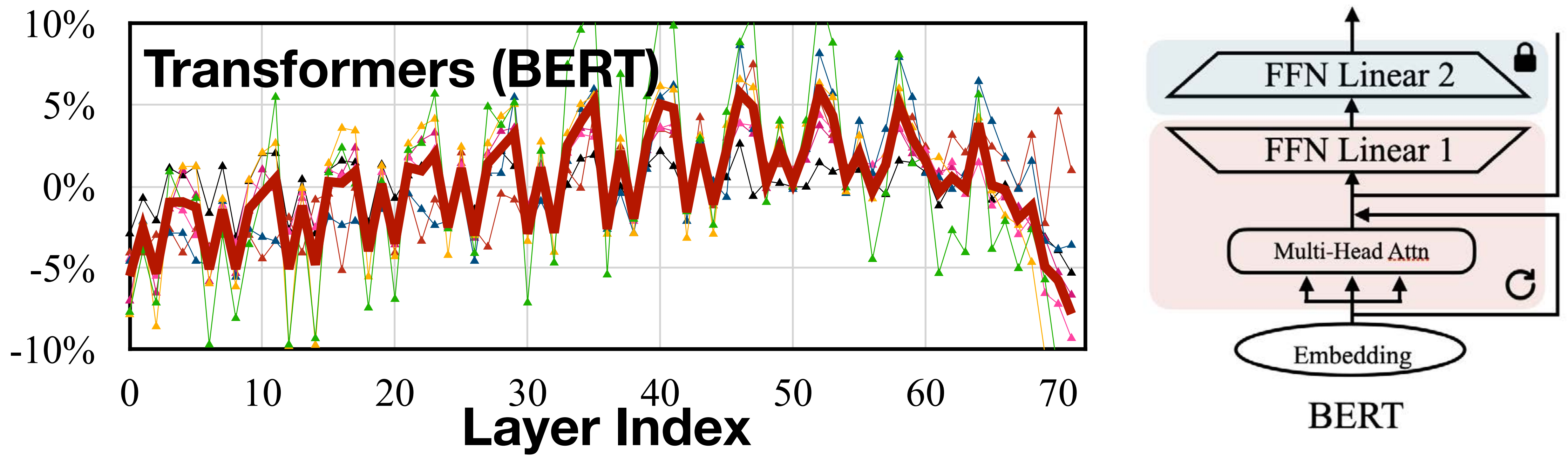
The camera is OpenMV Cam.

Sparse Training

Only update important layers and sub-tensors to save memory

- **Sensitivity analysis**

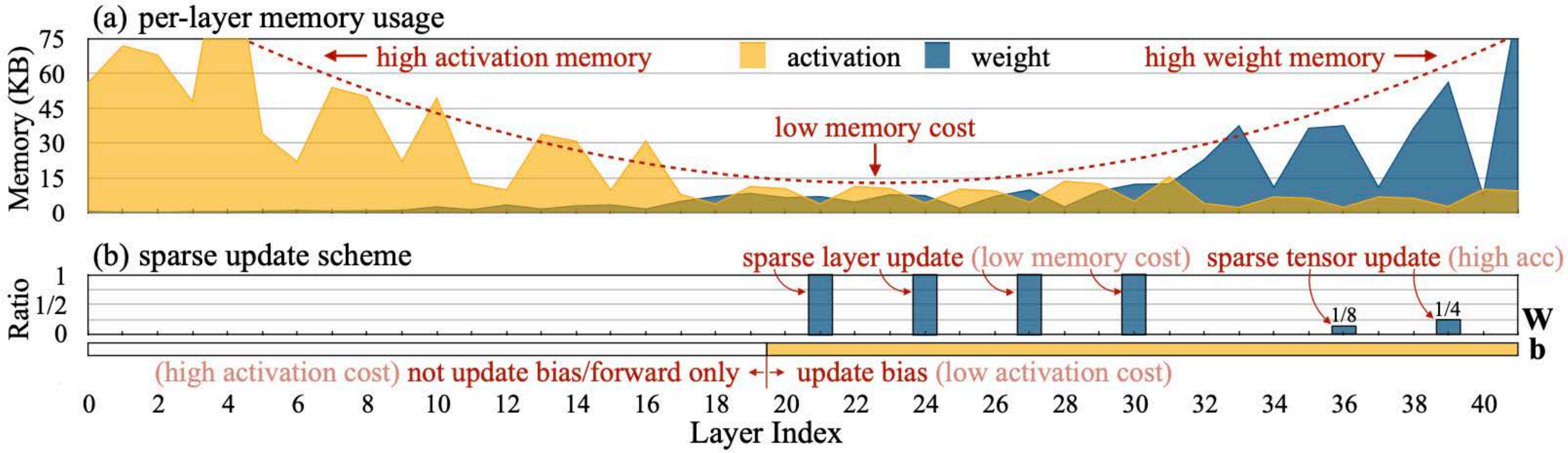




- **Later layers** are more important
- The **first point-wise conv** in each block contributes more

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

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



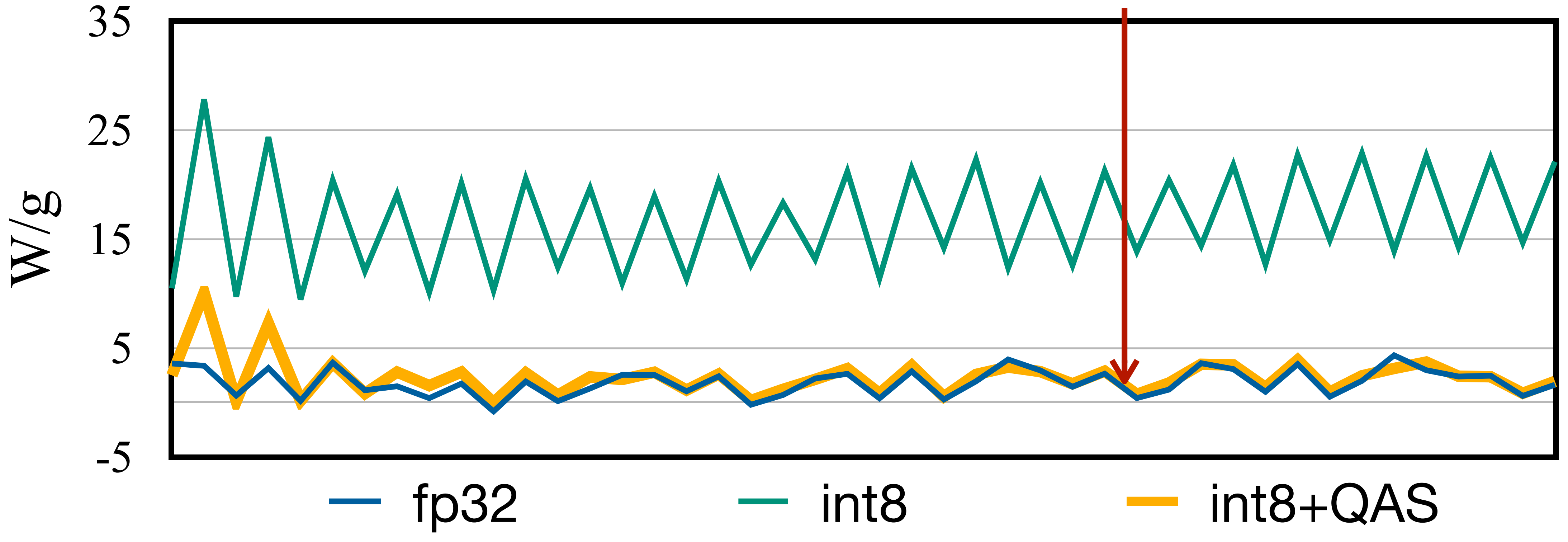
On-Device Training Under 256KB Memory [Lin *et al.*, NeurIPS 2022]

Low-Precision Training
with Quantization Aware Scaling (QAS)

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

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

QAS aligns the W/G ratio



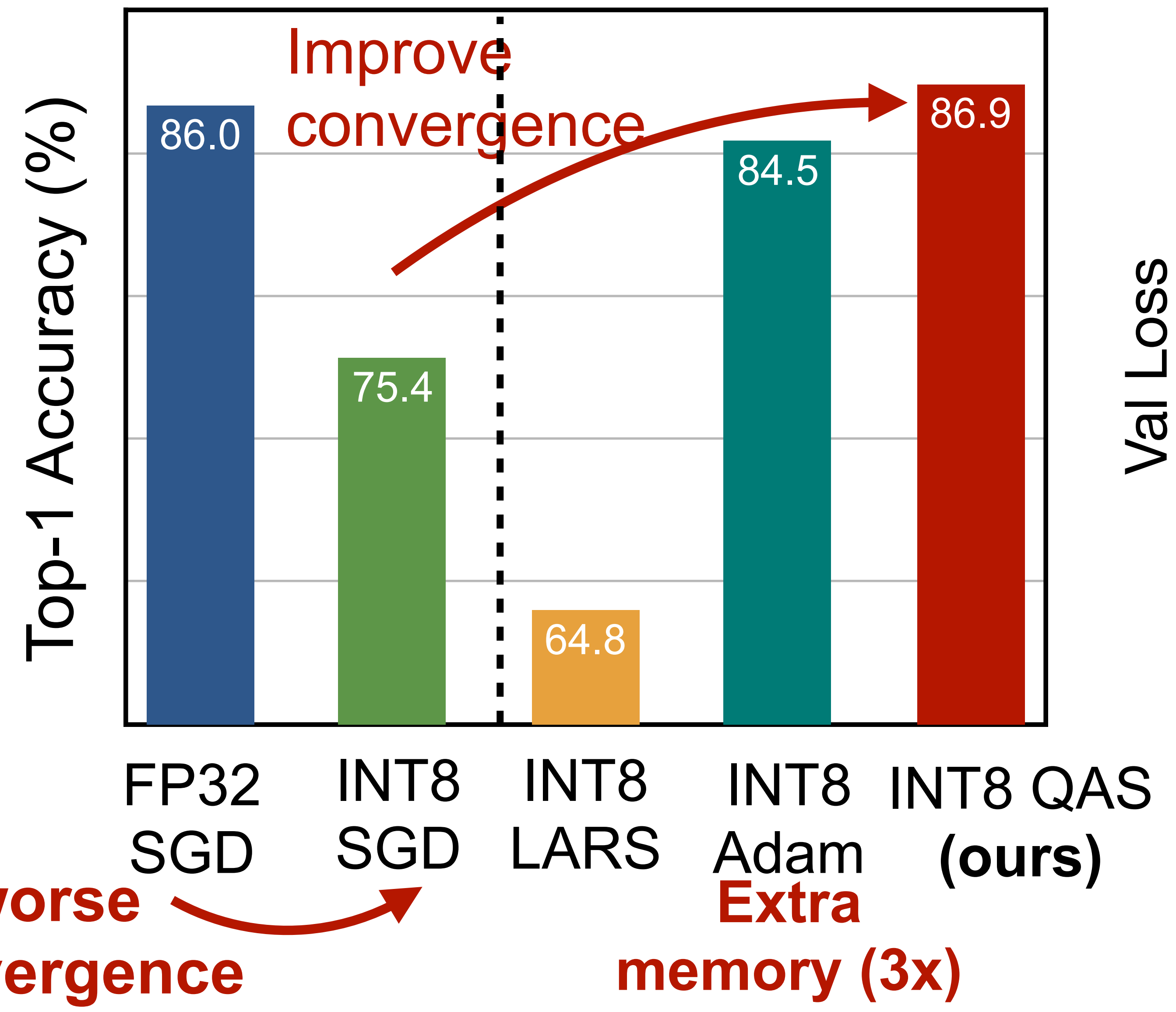
- Solution: **quantization-aware scaling (QAS)**

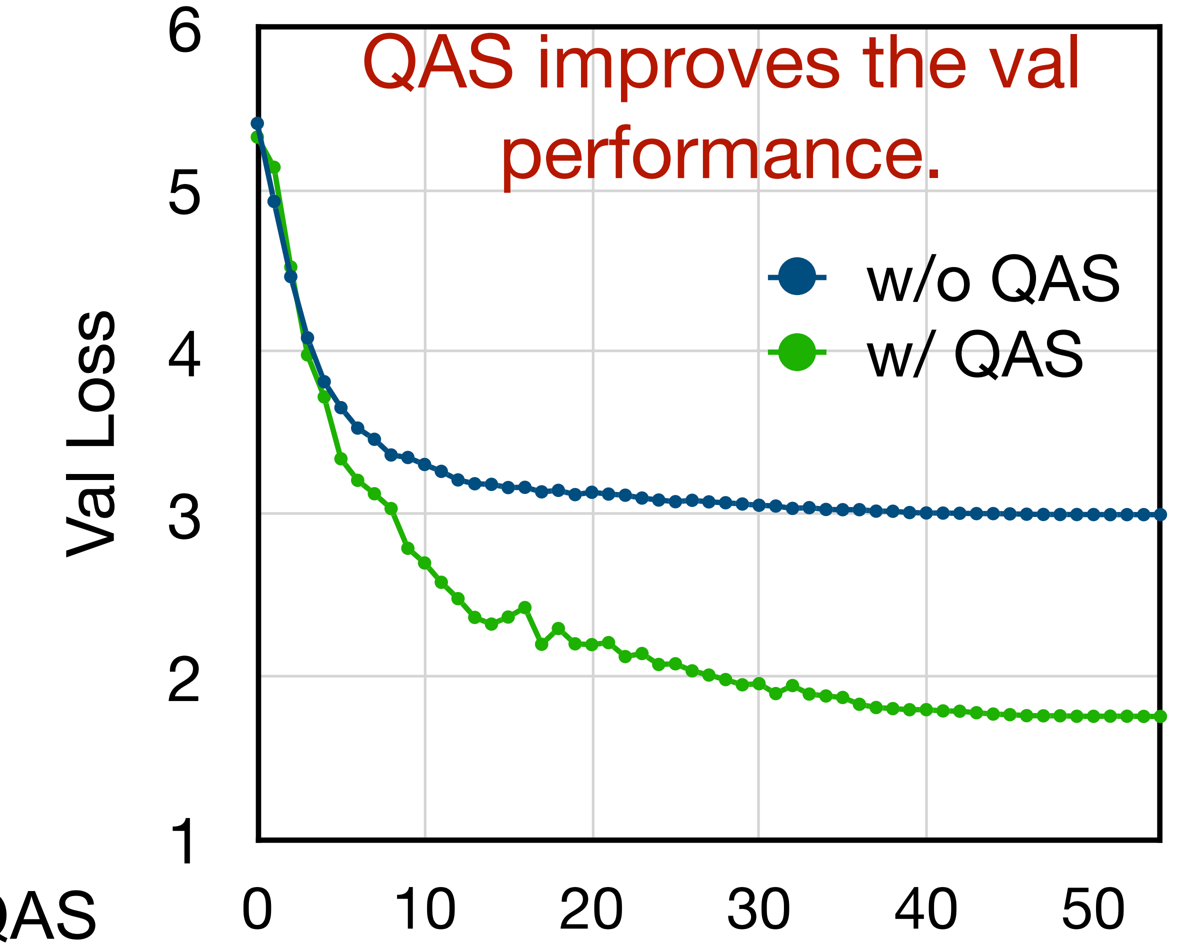
$$\overset{[-2,\,3]}{W} = s_W \cdot (W/s_W) \overset{quantize}{\approx} s_W \cdot \overset{[-128,\,127]}{W_Q}, \quad G_{W_Q} \approx s_W \cdot G_W$$

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

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

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

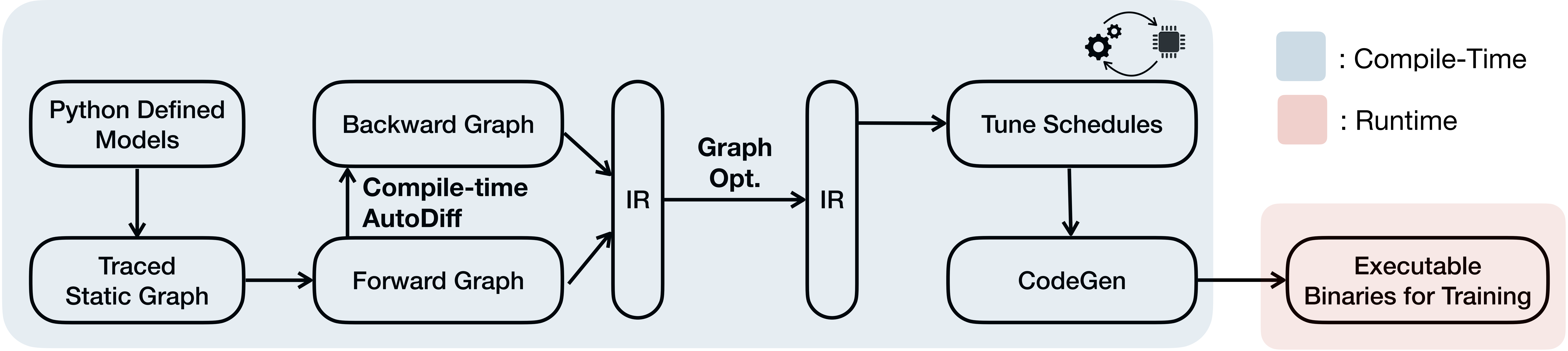


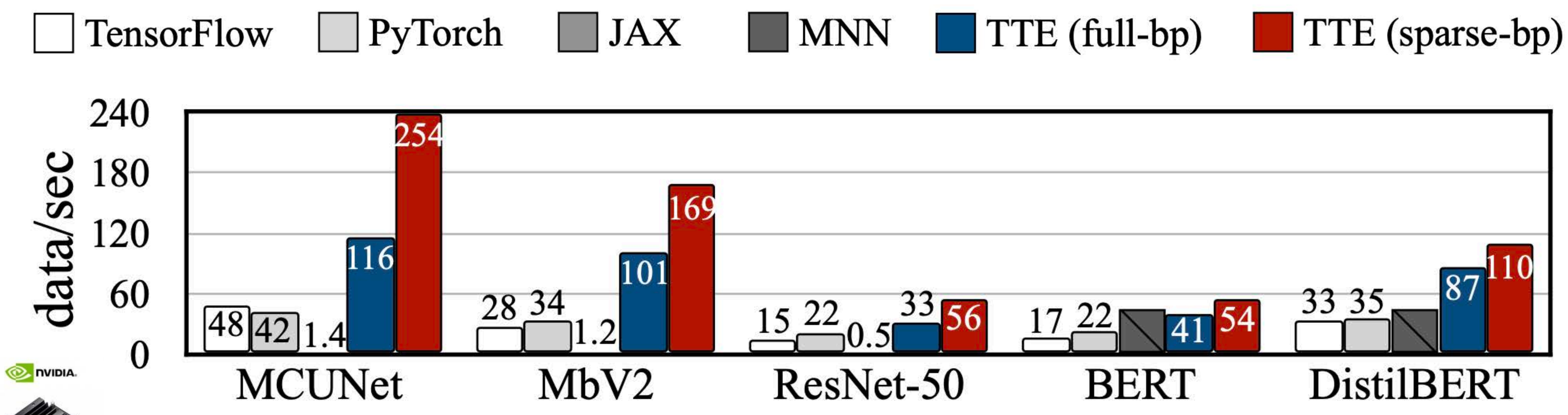


On-Device Training Under 256KB Memory [Lin *et al.*, NeurIPS 2022]

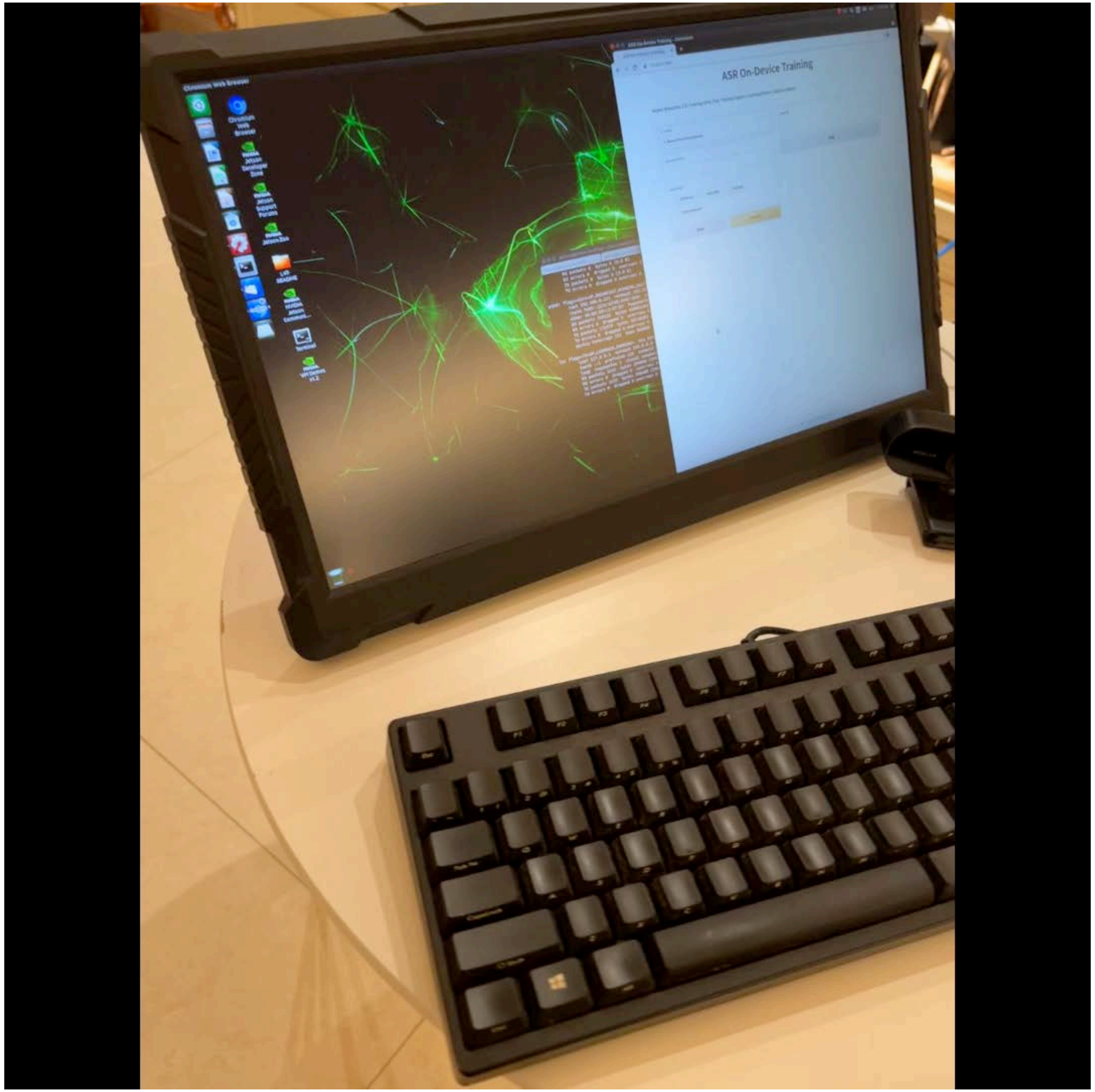
Tiny Training Engine

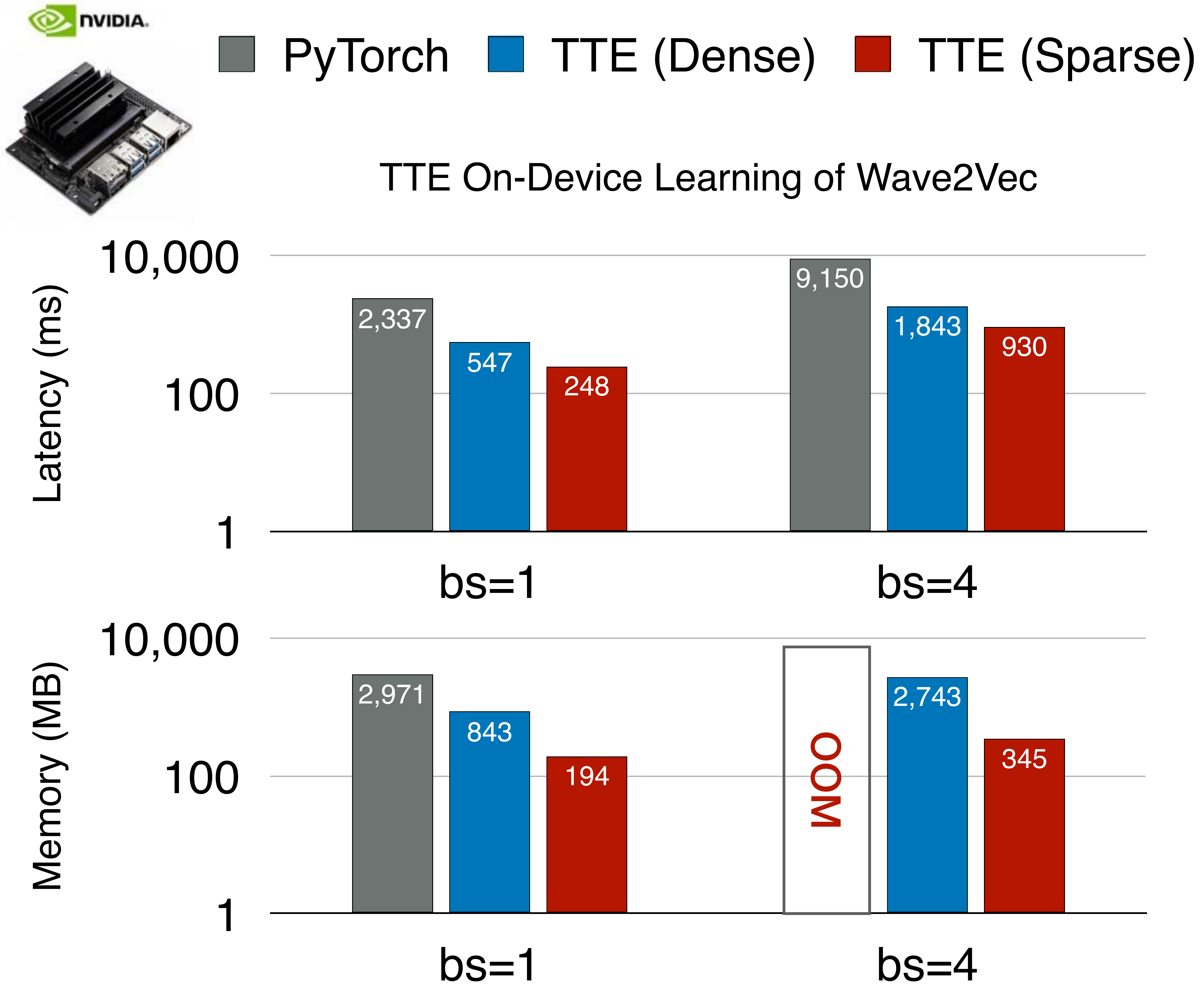
Translate the theoretical saving into measured savings. Runtime => Compile time





6x speedup compared to TensorFlow on Jetson Nano





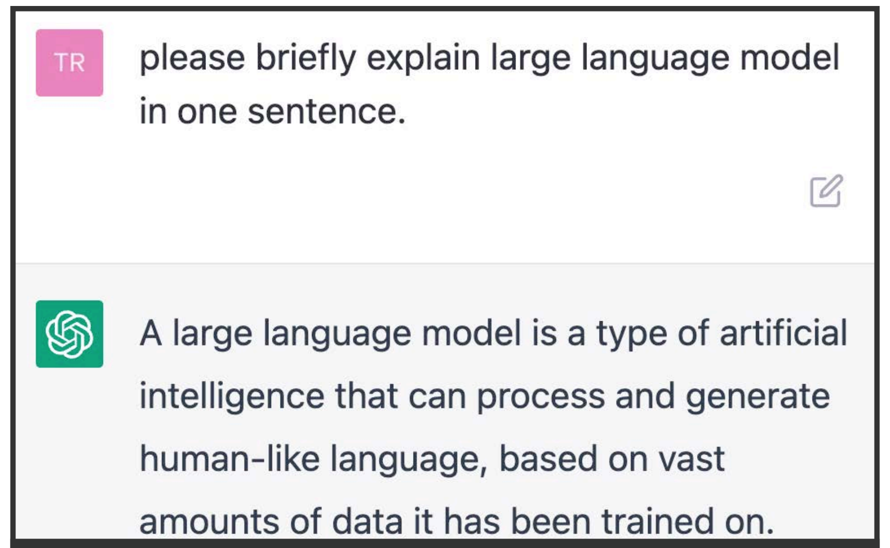
Accuracy (%)

Wav2Vec2.0	Word Error Rate on TIMIT
Full update	33.9
Classifier-only update	98.7
Sparse Update (last two Encoder + FC)	33.3

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

Model Compression for Diverse Applications

Video Synthesis Search Engine Revolution Chatbots
Predictive Maintenance Art Generation Question Answering Augmented Reality
Gesture Recognition Storytelling Autonomous Driving
Video Recognition Music Composition Sentiment Analysis Blind Spot Detection
Health Monitoring Fashion Design Machine Translation Adaptive Cruise Control



Large Language Model

Generative AI

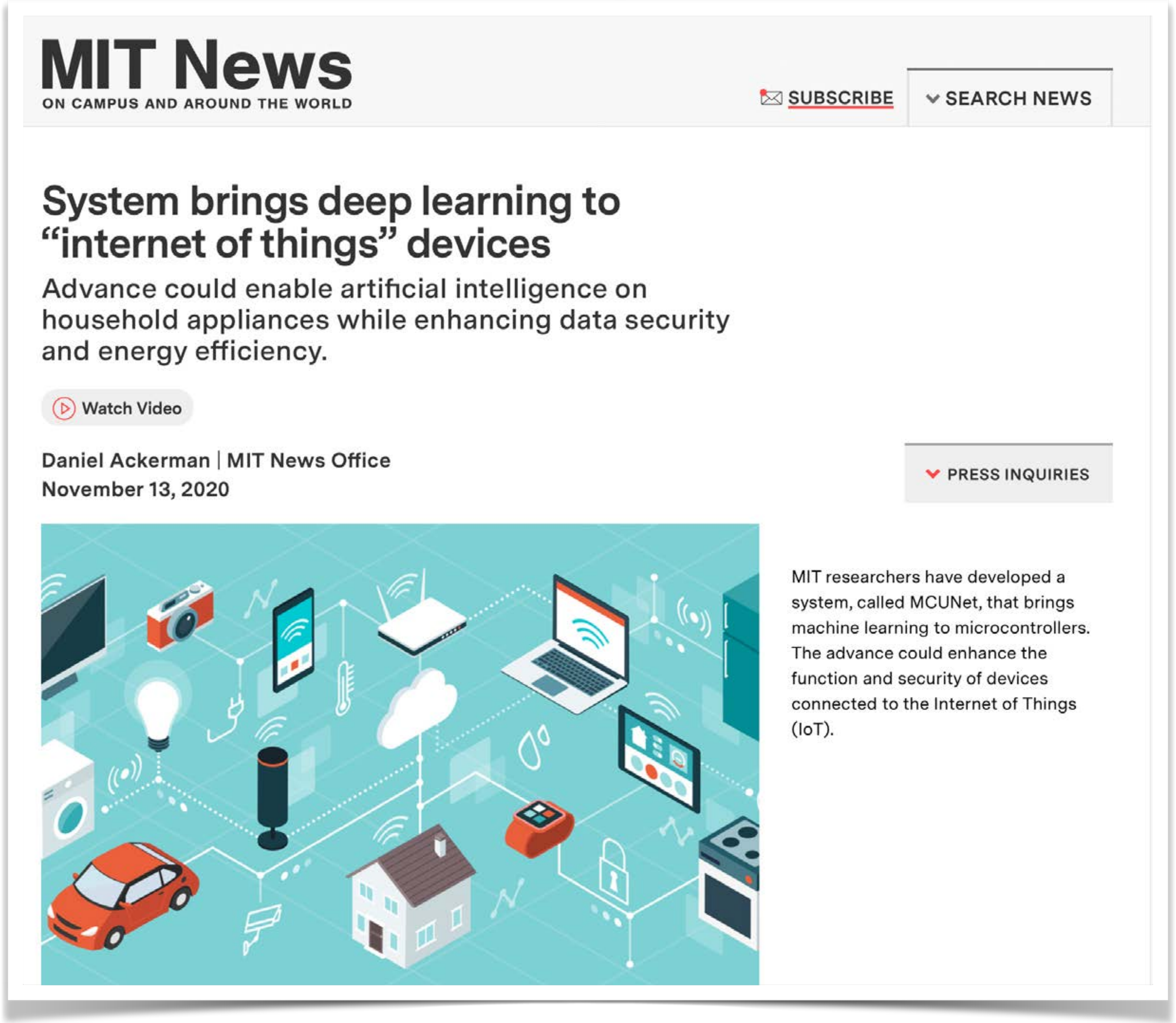
Driver Assistance System

TinyML

Application (demand of computation)

Hardware (supply of computation)

Media



MIT News
ON CAMPUS AND AROUND THE WORLD
SUBSCRIBE
SEARCH NEWS

System brings deep learning to "internet of things" devices

Advance could enable artificial intelligence on household appliances while enhancing data security and energy efficiency.

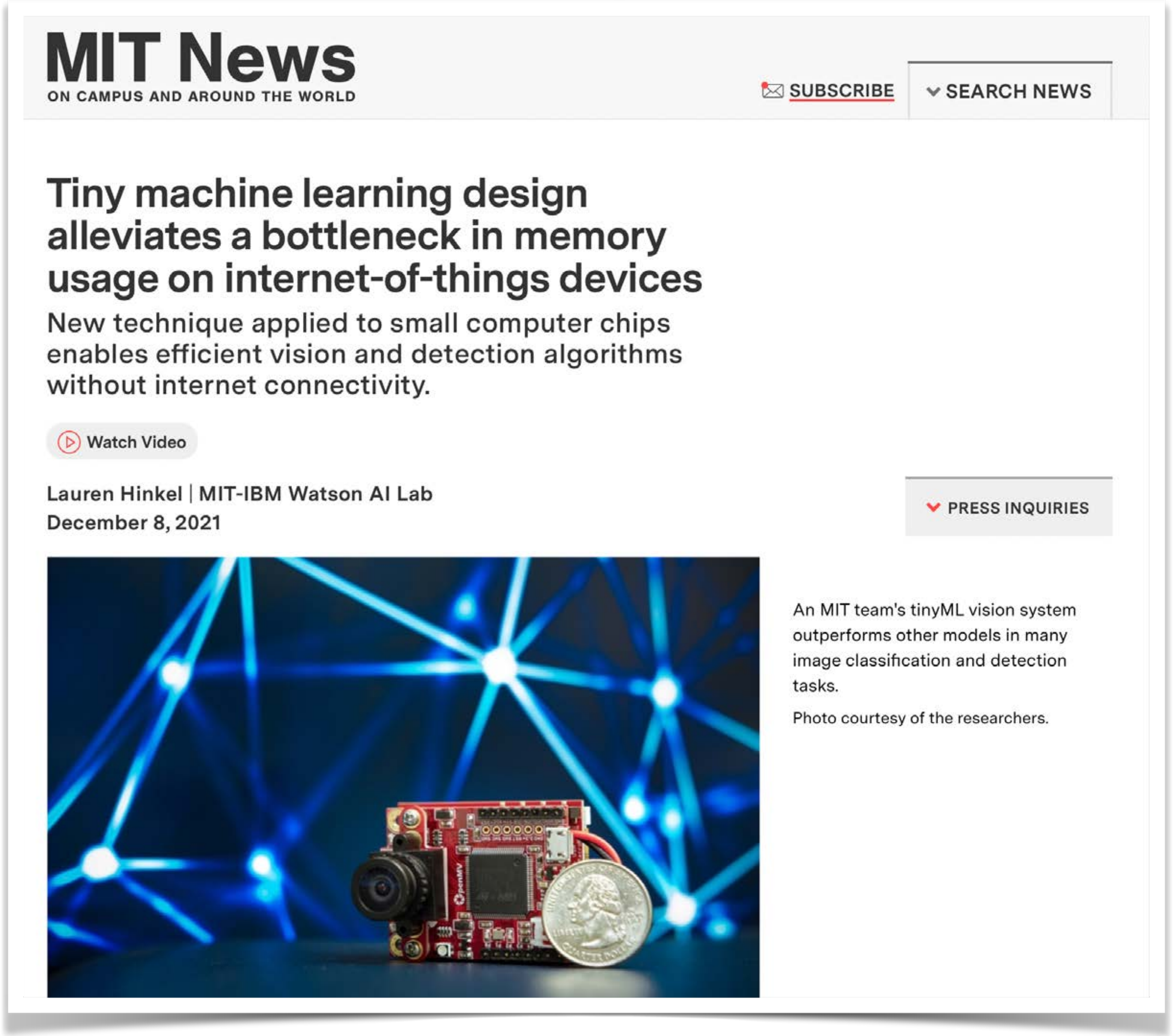
Watch Video

Daniel Ackerman | MIT News Office
November 13, 2020

PRESS INQUIRIES

MIT researchers have developed a system, called MCUNet, that brings machine learning to microcontrollers. The advance could enhance the function and security of devices connected to the Internet of Things (IoT).

(Highlighted by MIT Homepage)



MIT News
ON CAMPUS AND AROUND THE WORLD
SUBSCRIBE
SEARCH NEWS

Tiny machine learning design alleviates a bottleneck in memory usage on internet-of-things devices

New technique applied to small computer chips enables efficient vision and detection algorithms without internet connectivity.

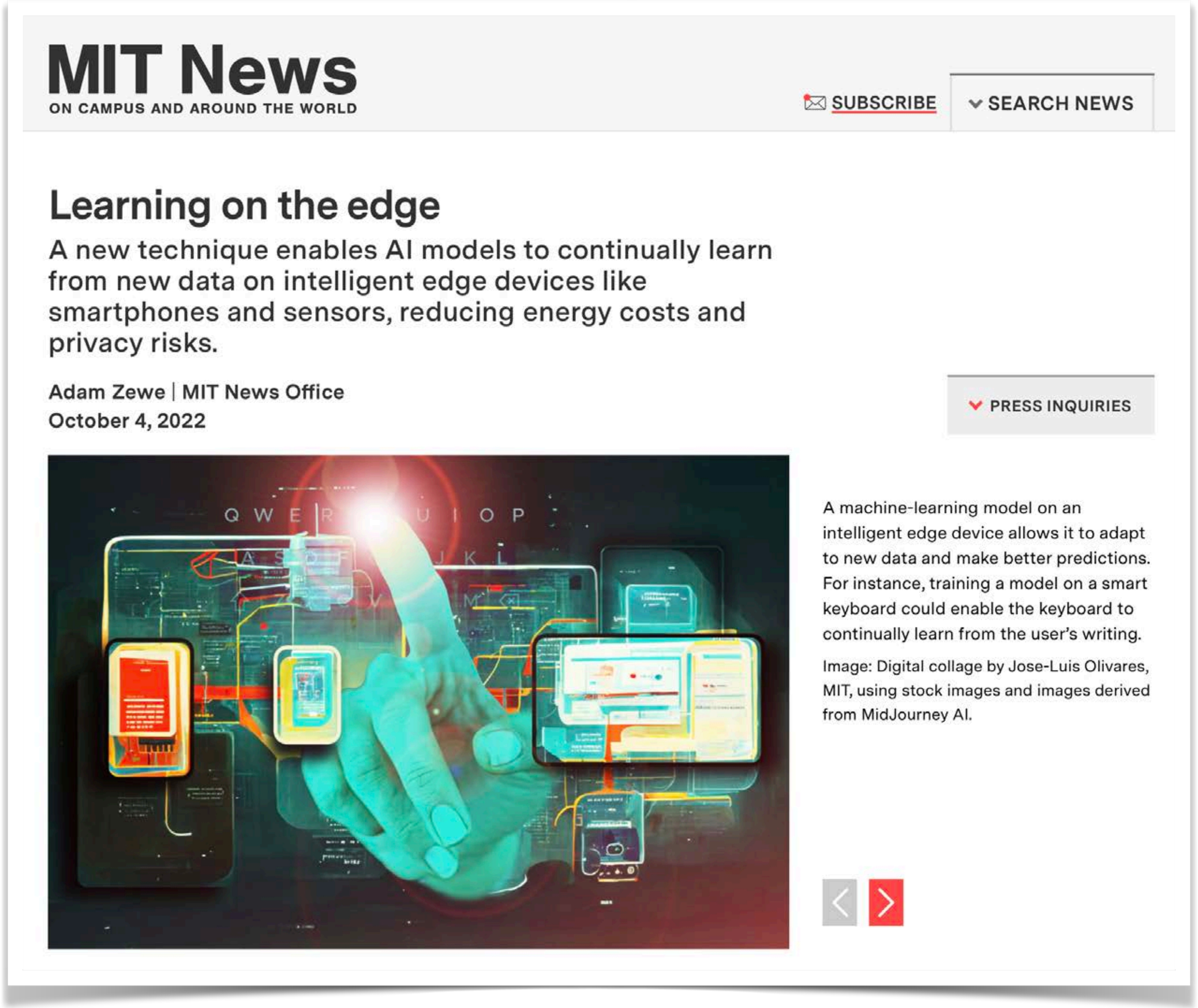
Watch Video

Lauren Hinkel | MIT-IBM Watson AI Lab
December 8, 2021

PRESS INQUIRIES

An MIT team's tinyML vision system outperforms other models in many image classification and detection tasks.

Photo courtesy of the researchers.



MIT News
ON CAMPUS AND AROUND THE WORLD
SUBSCRIBE
SEARCH NEWS

Learning on the edge

A new technique enables AI models to continually learn from new data on intelligent edge devices like smartphones and sensors, reducing energy costs and privacy risks.

Adam Zewe | MIT News Office
October 4, 2022

PRESS INQUIRIES

A machine-learning model on an intelligent edge device allows it to adapt to new data and make better predictions. For instance, training a model on a smart keyboard could enable the keyboard to continually learn from the user's writing.

Image: Digital collage by Jose-Luis Olivares, MIT, using stock images and images derived from MidJourney AI.

(Highlighted by MIT Homepage)

MCUNet: Tiny Deep Learning on IoT Devices [Lin *et al.*, NeurIPS 2020]
MCUNetV2: Memory-Efficient Patch-based Inference for Tiny Deep Learning [Lin *et al.*, NeurIPS 2021]
On-Device Training Under 256KB Memory [Lin *et al.*, NeurIPS 2022]

Open Source

Sign up here to get updates!

https://forms.gle/UW1uUmnfk1k6UJPPA

New Course: TinyML and Efficient Deep Learning Computing

MIT 6.S965: https://efficientml.ai

6.S965 Logistics Schedule

TinyML and Efficient Deep Learning

6.S965 • Fall 2022 • MIT

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, and 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.

- **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:** Piazza
- **Homework submission:** Canvas
- **Online lectures:** The lectures will be streamed on YouTube.
- **Resources:** MIT HAN Lab, Github, TinyML, MCUNet, OFA
- **Contact:** Students should ask all course-related questions on Piazza. For external inquiries, personal matters, or emergencies, you can email us at *6s965-fall2022-staff@mit.edu*.

Instructor Song Han
Email: songhan@mit.edu

TA Zhijian Liu
Email: zhijian@mit.edu

TA Yujun Lin
Email: yujunlin@mit.edu

Anonymous Student Feedback Collected from Mid-term

- This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices.

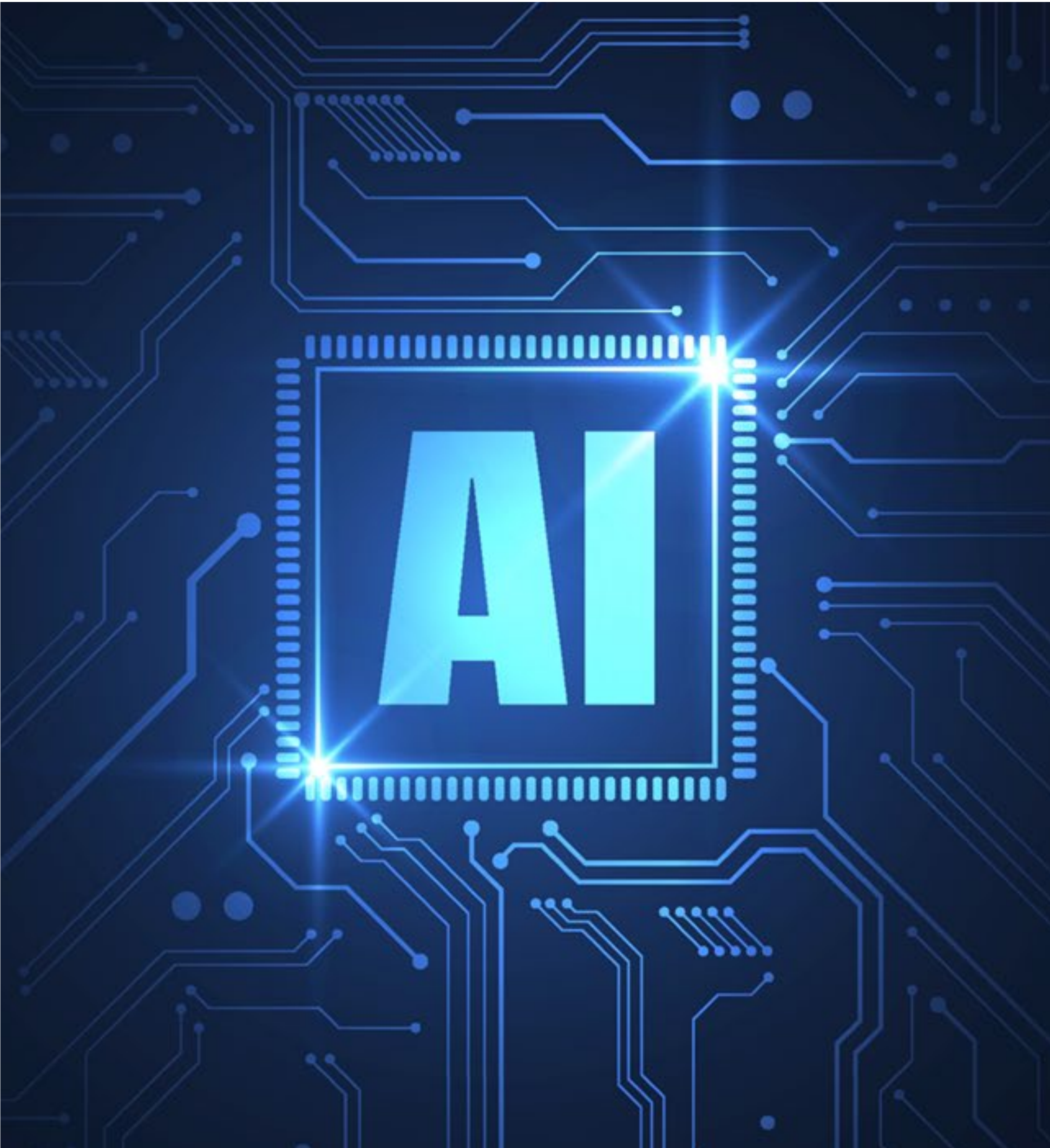
I really like how structured the labs are, and being able to see actual implementations of the techniques we learn about.

This is honestly one of the best set up courses I've taken at MIT

I love how we are using microntroller and focusing on application instead of just theories.

I managed the weekly labs and lectures by only watching the course on YouTube. As a researcher, I gained some valuable knowledge from your course. Excellent slides and teaching and useful labs.

I like the class and I have been able to follow the class easily (which had rarely happened to me in my previous courses)



MIT AI Hardware Program

MIT Microsystems Technology Laboratories (SoE)

MIT Quest for Intelligence – Corporate (SCC)

Co-Leads: Jesús del Alamo and Aude Oliva

Internal Advisory Board Chair: Anantha Chandrakasan

TinyML and Efficient AI

MIT HAN LAB
Hardware, AI and Neural-nets

github.com/mit-han-lab
youtube.com/c/MITHANLab
songhan.mit.edu
tinyml.mit.edu

Sponsors:

IBM, QUALCOMM, intel, NVIDIA, XILINX, maxim integrated, NSF, amazon, Google, facebook, SONY, SAMSUNG, HYUNDAI, Ford

Media:

MIT Technology Review, IEEE SPECTRUM, WIRED, engadget, MIT News, AI DAILY, VentureBeat, ScienceDaily, Analytics Insight, AI Business