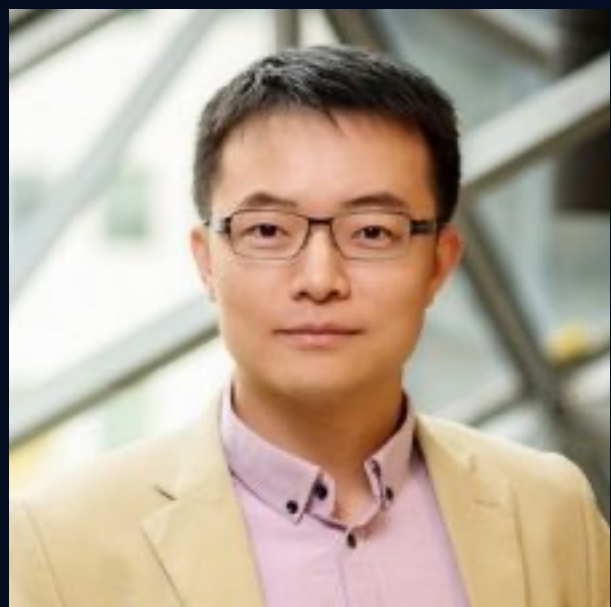
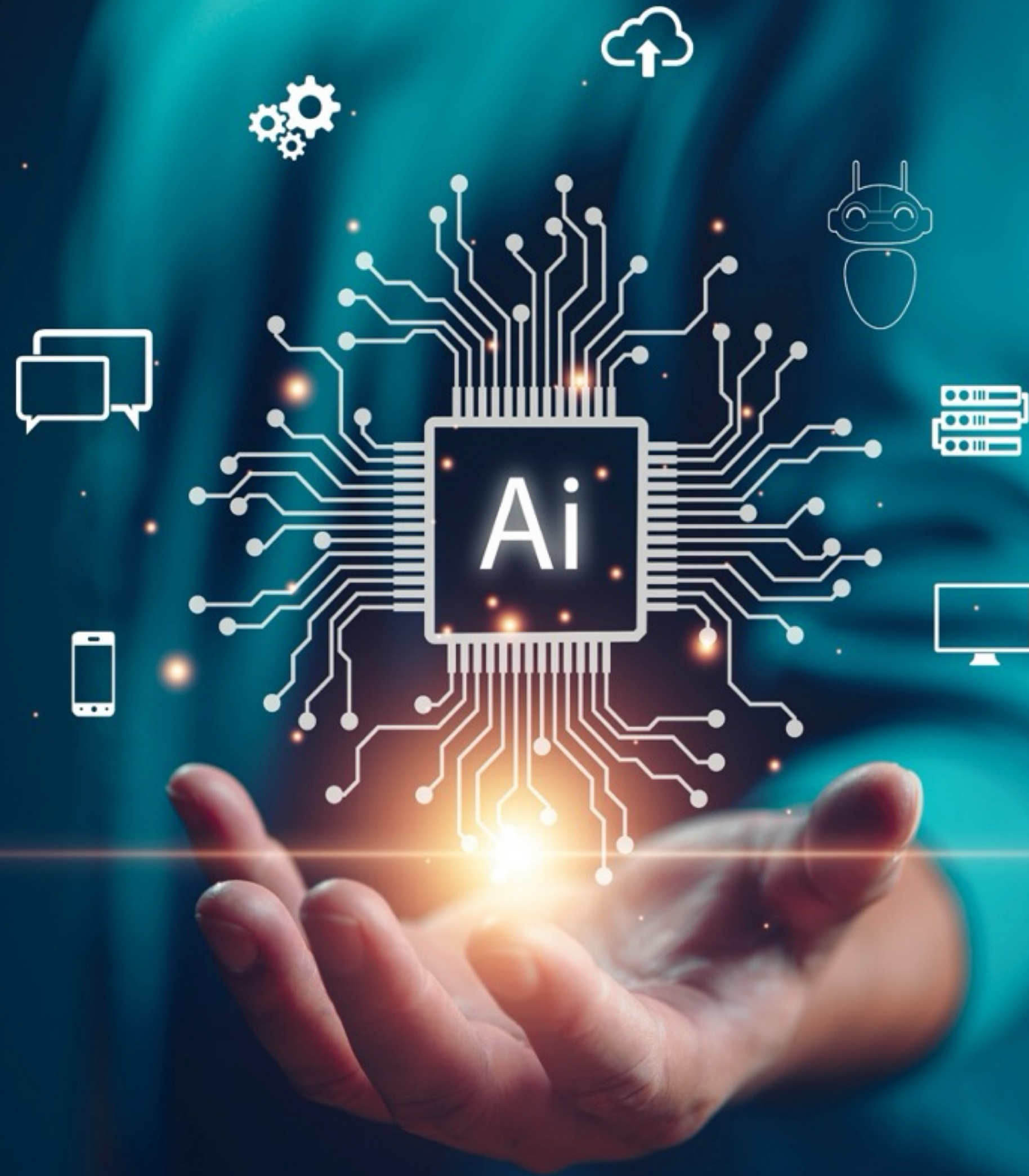


Efficient AI Computing



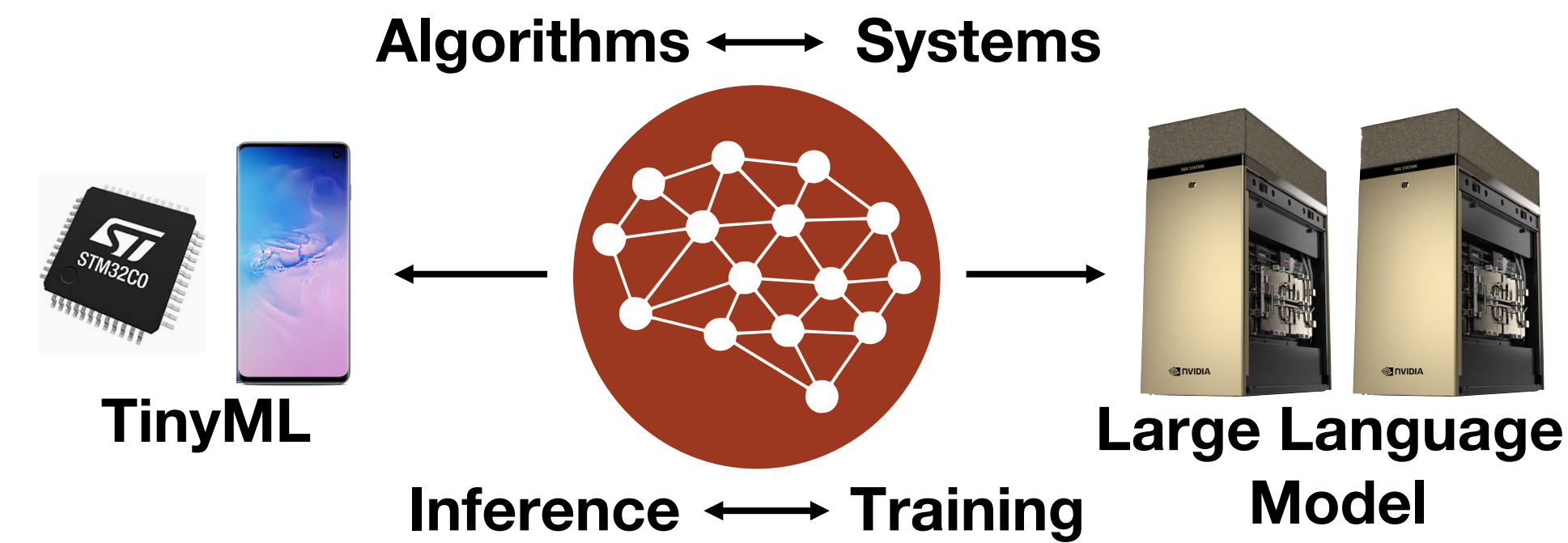
Song Han

<https://hanlab.mit.edu>



Research Overview

TinyML and Efficient Deep Learning Computing



- **Motivation:**

- Deep learning is requiring more computation than ever; training and inference become very costly.
- Enable deep learning on small, low-power devices (TinyML).
- Greener AI: reduce model size, latency, memory, energy; increase throughput, accuracy, scalability, productivity.

- **Approach:**

- **Model compression algorithms** that shrink neural networks without compromising accuracy: pruning, quantization, distillation, hardware-aware neural architecture search, novel neural architectures and building blocks.
- **Efficient systems and hardware** that implements the algorithmic innovations into measured speedup. Exploit sparsity and redundancy with algorithm and system co-design.
- **Application-specific optimizations** for generative AI, including large language model and diffusion model. Invent new operators to efficiently perform high-resolution image generation and long text generation.

- **Impact:**

- Pioneered the area of TinyML, at the intersection between machine learning and systems.
- Model compression, pruning and quantization have become the standard lexicon of the field.
- Research is adopted by industry (NVIDIA, AMD, Xilinx, Intel, Google, HuggingFace), with two startups acquired.

Thrust 1: Tiny Machine Learning (TinyML)

1.1 Tiny Inference

Motivation:

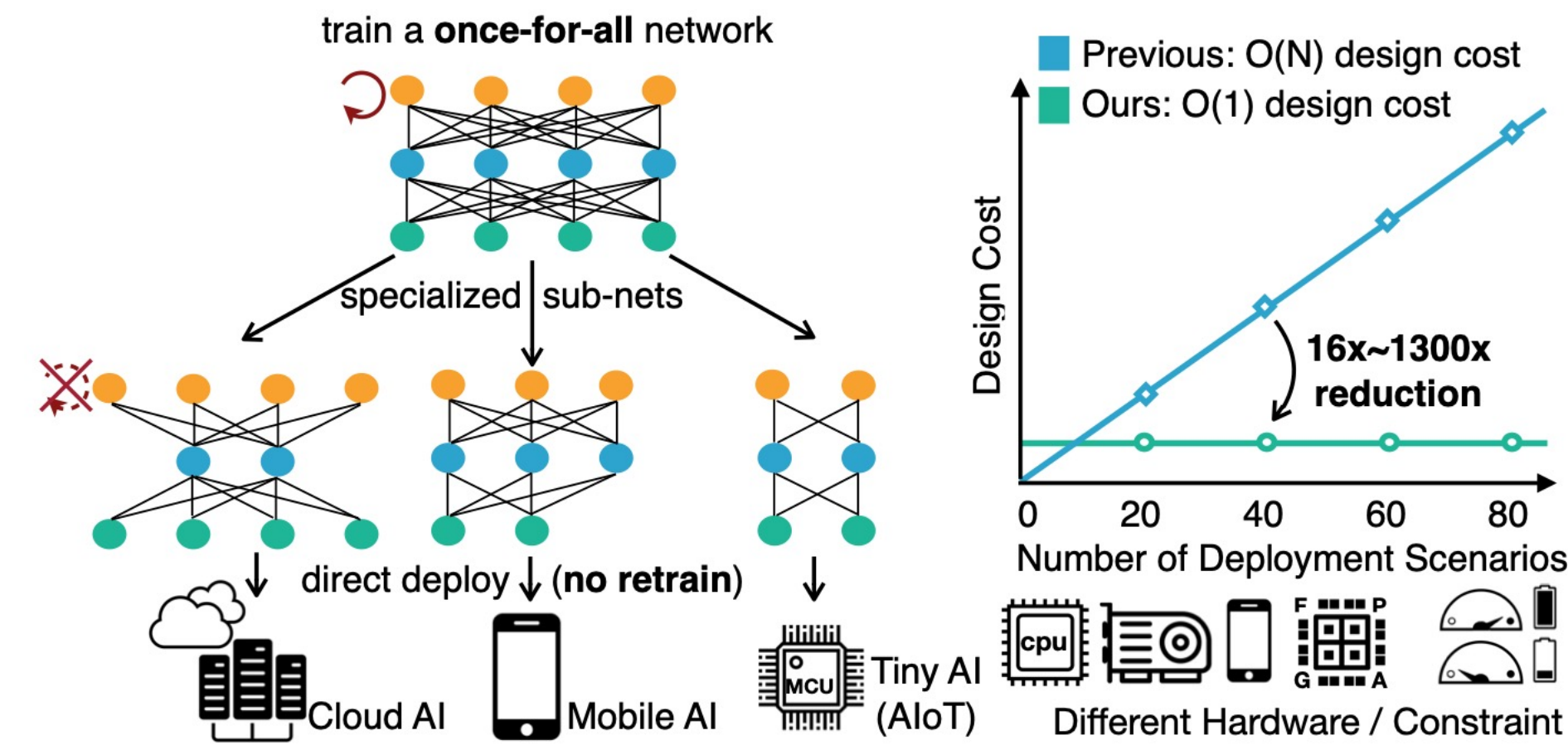
- Deploy neural networks on edge devices.
- Hardware-in-the loop neural architecture search is essential.
- Large design space: manual design is costly; automated design is needed.

Innovations:

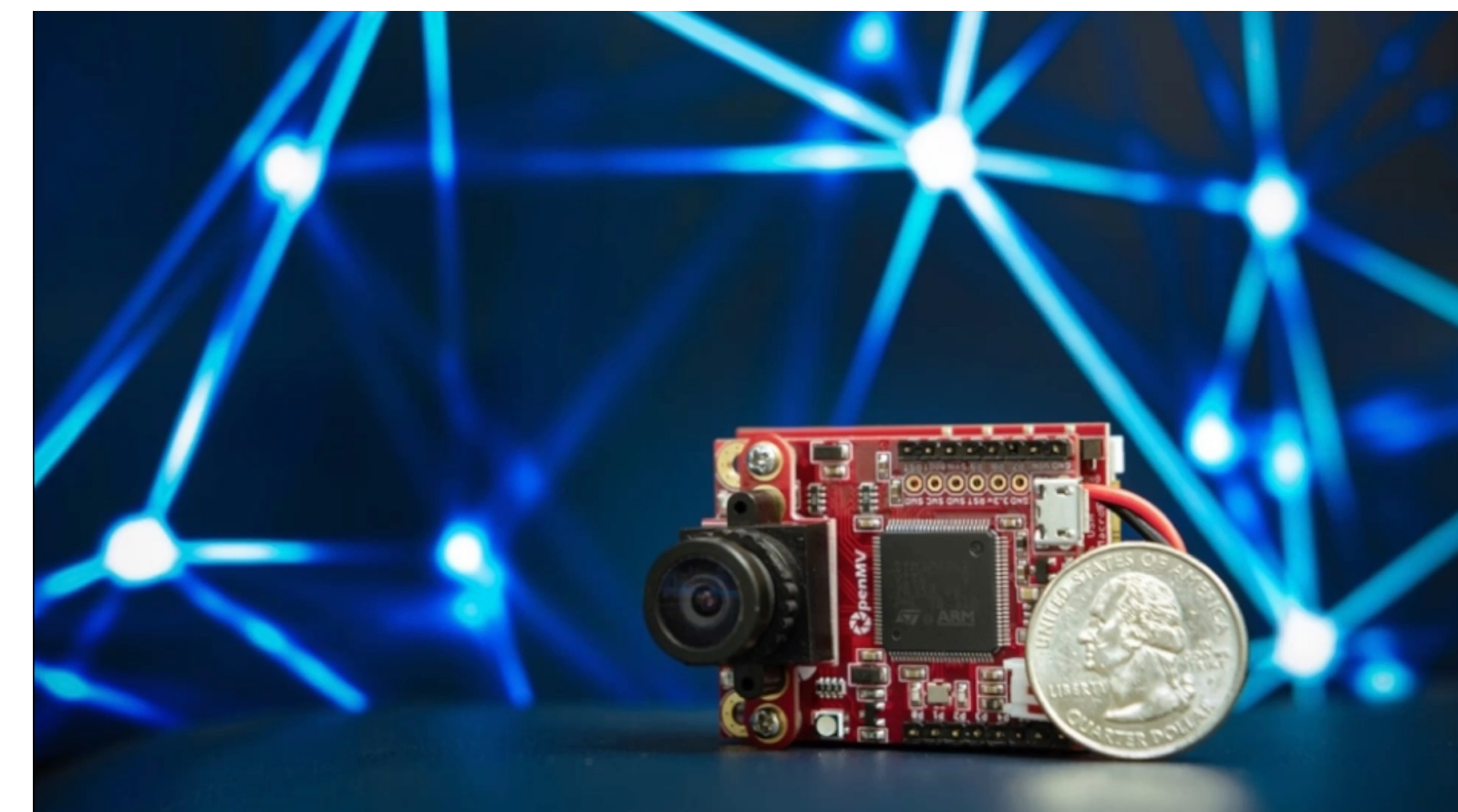
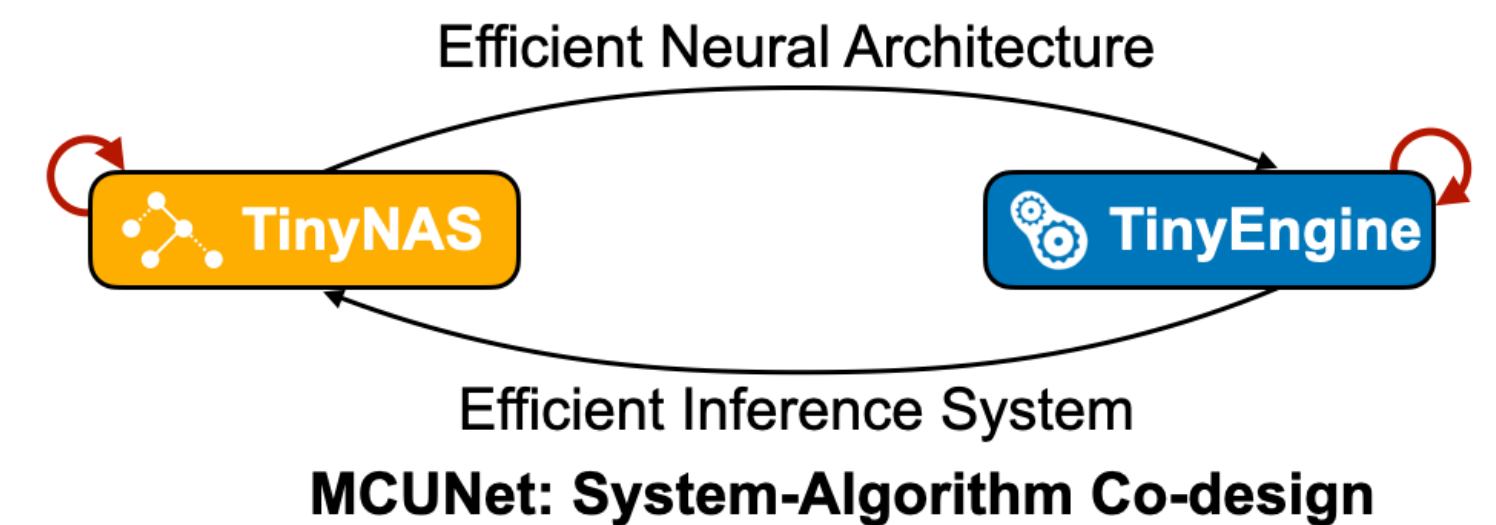
- Once-For-All Network [ICLR'20]: train a **single**, powerful super network that can generate **many** subnetworks, while only taking up the memory footprint of a single full model. Search the best subnetwork based on hardware resources; build a latency predictor to provide hardware feedback.
- MCUNet [NeurIPS'20/21] brings deep learning to microcontrollers (MCUs).
 - TinyNAS: hardware-aware neural architecture search.
 - TinyEngine: efficient inference system co-designed with TinyNAS.
 - Pioneering work running neural networks on micro controllers.

Impact:

- Featured article by IEEE Circuits and Systems Magazine. MCUNet is adapted by many universities as course material, including Harvard, Princeton, U Penn, CMU. Once-for-all network is adopted by PyTorch, SONY and ADI.



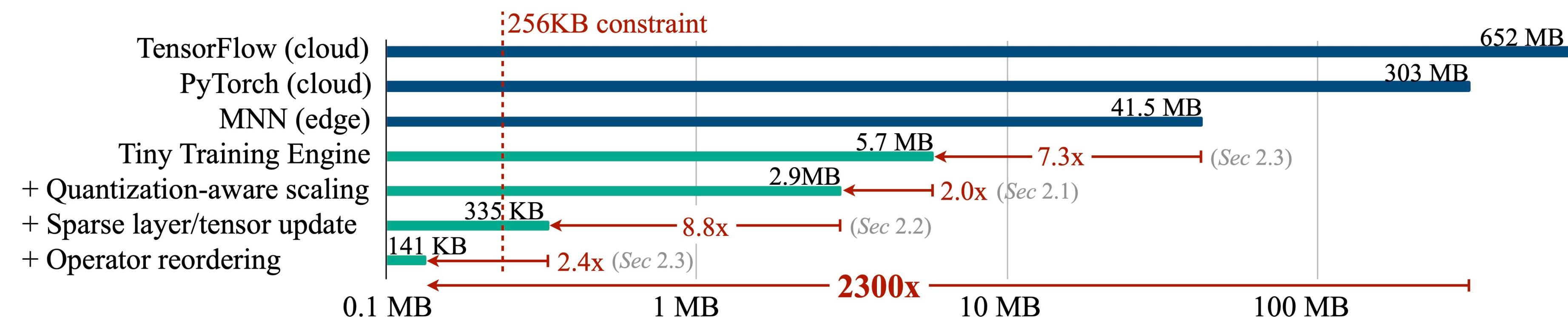
Once-for-all Network: train one get many



Thrust 1: Tiny Machine Learning (TinyML)

1.2 Tiny Training

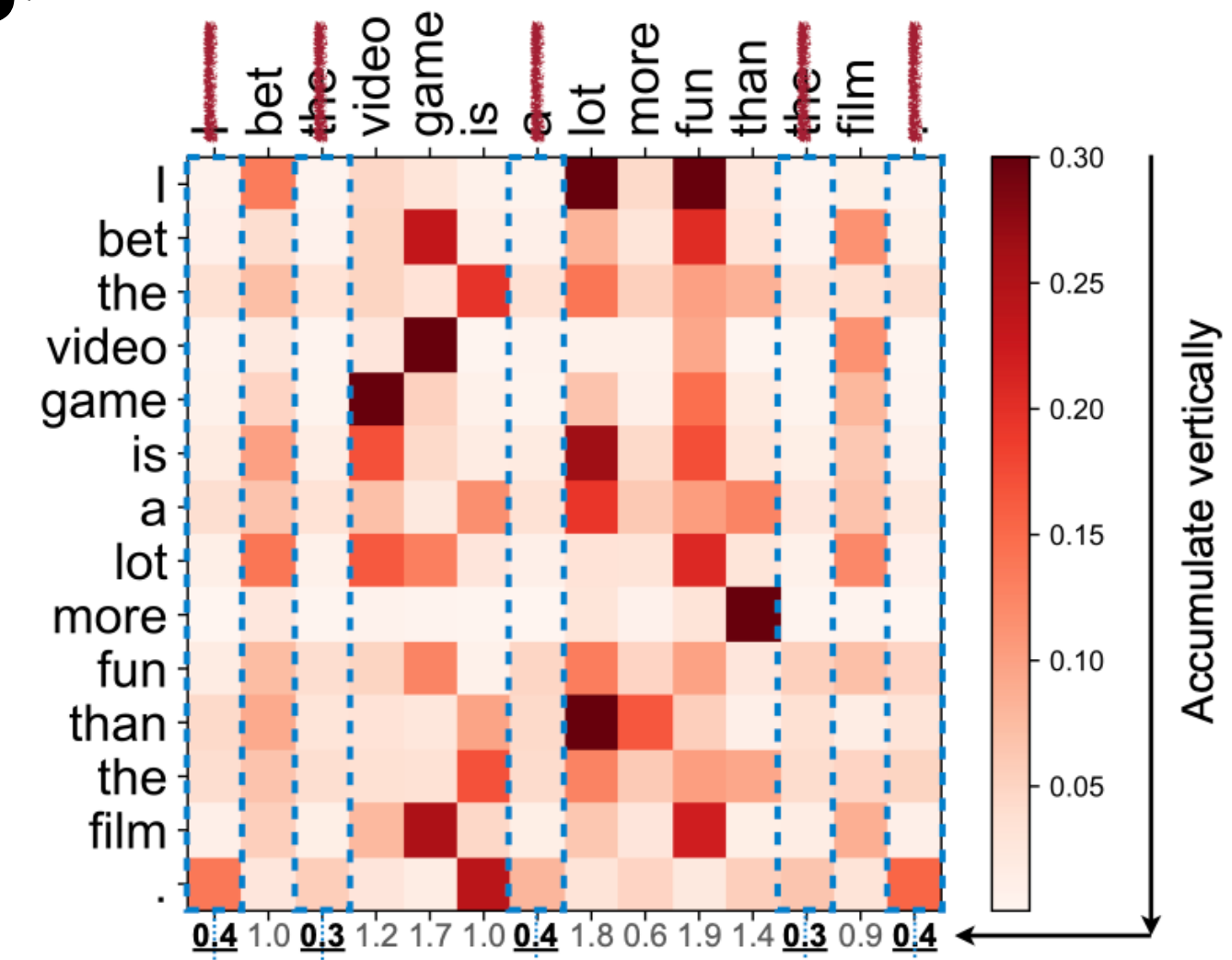
- **Motivation:** Enables IoT devices to adapt to new data collected from the sensors by fine-tuning a pre-trained model without sending data to the cloud; enable life-long on-device learning with better privacy.
- **Challenge:** Much harder than inference: back-propagation requires storing intermediate activations => large memory.
- **Innovations:** On-device training under 256KB memory [NeurIPS'22]
 - Quantization-aware scaling: perform training with low precision and save memory, while stabilizing convergence.
 - Sparse update: skip the gradient computation of less important layers and sub-tensors, save memory.
 - Tiny training engine: prunes the backward computation graph to support sparse update; offloads the runtime auto-diff to compile time.
- **Result:** the first solution to enable tiny on-device training of convolutional neural networks under 256KB SRAM and 1MB Flash. Using less than **1/1000** of the memory of PyTorch and TensorFlow [[demo](#)]



Contribution 2: Accelerating AI with Sparsity

Exploit sparsity with algorithm, system, hardware co-design

- **Motivation:** Sparsity in neural networks arises where not all neurons are connected. Sparsity plays a pivotal role to save computation.
- **Prior work:** I designed the first accelerator to exploit weight sparsity (EIE @Stanford).
- **New Innovations:**
 1. **Identify new sources of sparsity:**
 - SpAtten [HPCA'21] introduces “sparse attention” and token pruning: not all tokens need to attend to each other. Prune away less important tokens.
 - New input sparsity opportunities in point cloud [NeurIPS'19, oral], multi-sensor fusion [ICRA'23], vision transformer [CVPR'23], and diffusion models [NeurIPS'22].



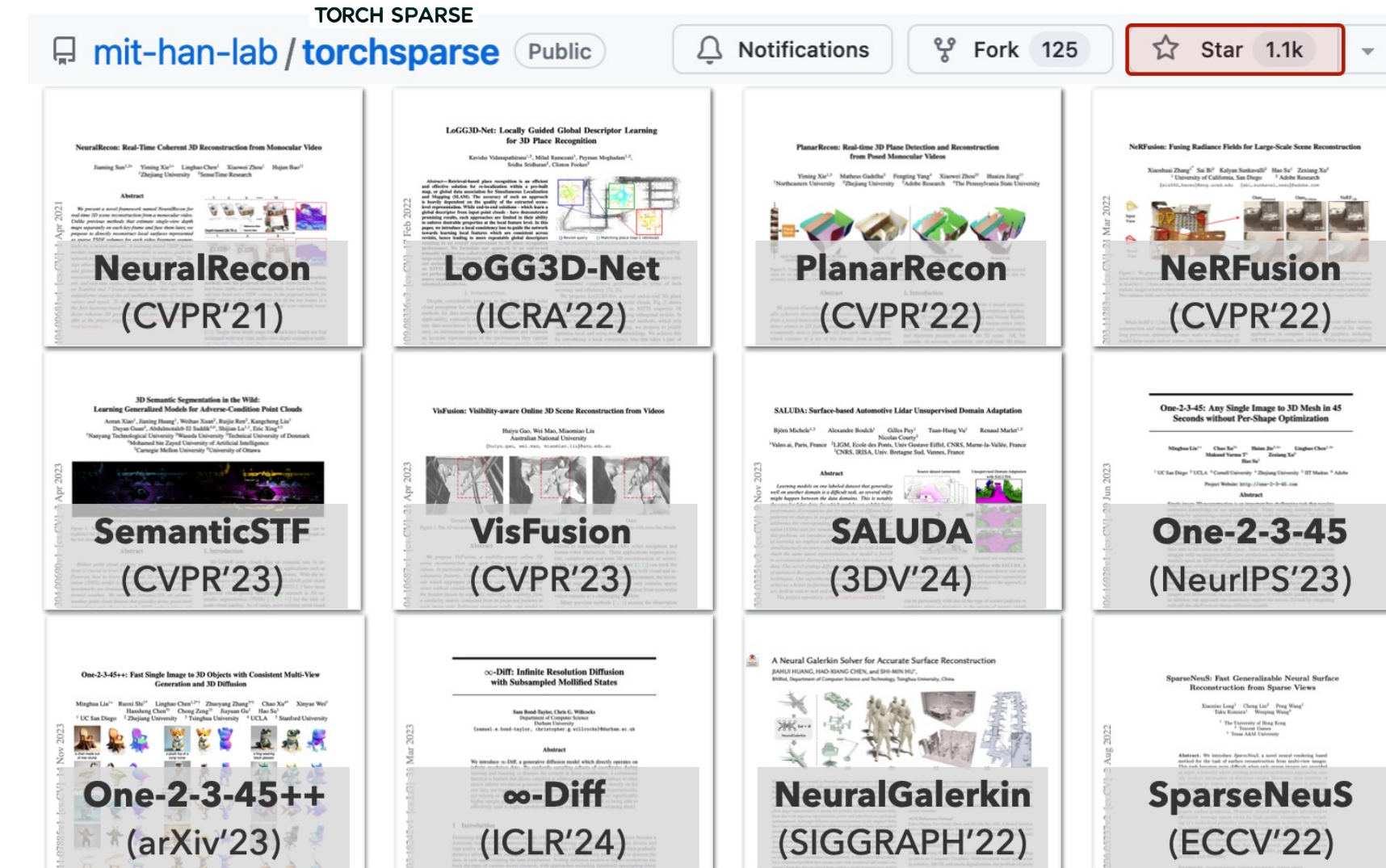
tokens with small attention scores are pruned away



Papers using TorchSparse:

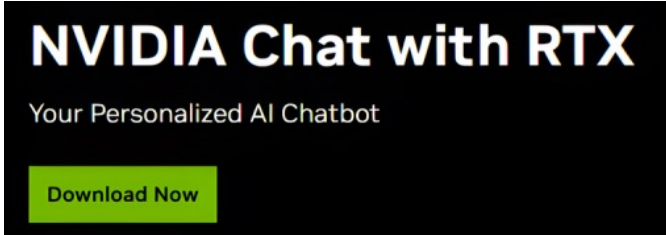
2. System & hardware support for sparsity:

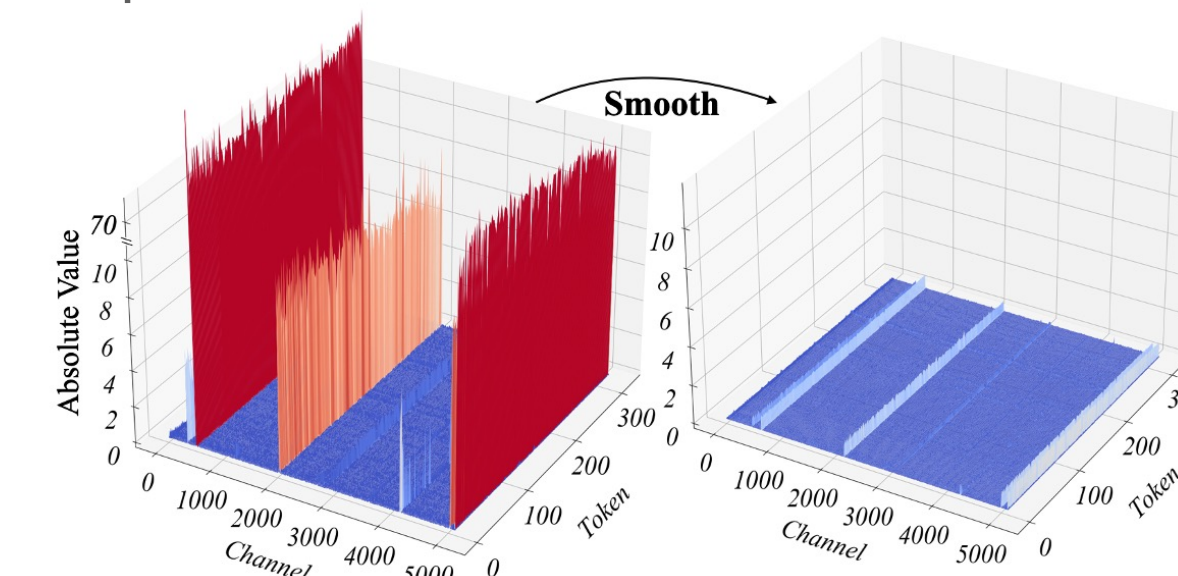
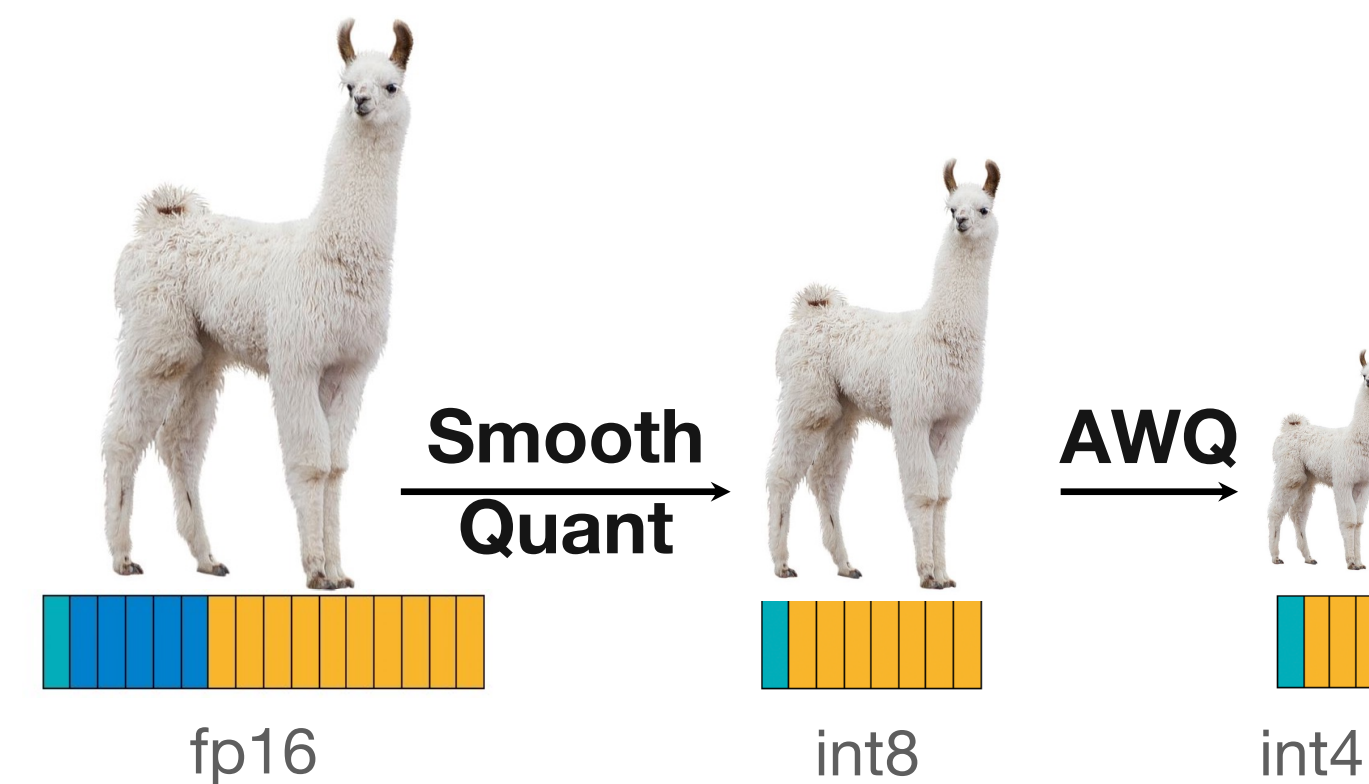
- TorchSparse [MLSys'22, MICRO'23]: optimizes irregular computation by *reordering* the outputs based on input bitmasks, minimizing *padding overhead*, enabling *load balancing*, and reducing *memory footprint*.
- Built specialized hardware to accelerate sparse operations: *top-k selection*, *non-zero merger*, *zero-elimination* and effectively skip zero computations. [SpAtten, HPCA'21], [SpArch, HPCA'20], [@PointAcc, MICRO'21]



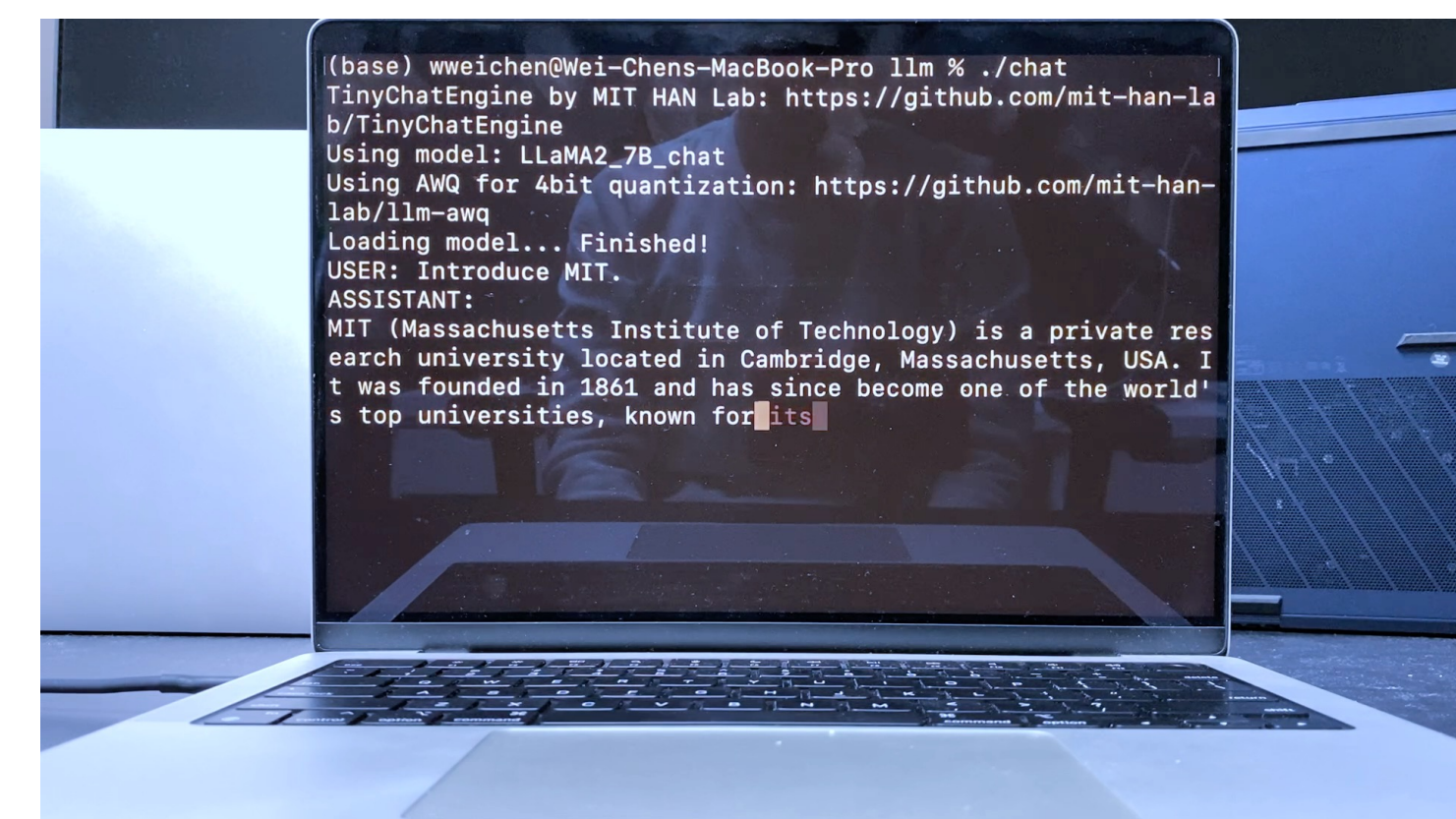
Thrust 3: Efficient Generative AI


3.1 Micro Design: Low Precision

- **Background:** GenAI models are **1000x bigger** than traditional CNNs, posing new compute challenge. Moore's law: 2x transistors / year; LLM: 4x larger / year.
- **Challenge:** Quantization and low-precision can bridge the gap, unfortunately, traditional quantization methods does not work for LLM due to the outliers, which stretch the quantization range, leaving few effective bits for most values.
- **Innovations:**
 - SmoothQuant (ICML'23) is a novel approach that smoothes the activation outliers by migrating the quantization difficulty from activations to weights with a mathematically equivalent transformation. No fine-tuning is needed.
 - AWQ (MLSys'24) further quantize LLM to **4-bit**. **TinyChat** implements 4-bit LLM, making it deployable on the edge.
- **Impact:**
 - AWQ has 920K+ downloads on HuggingFace.
 - AWQ is the key model compression technology behind  for AI PC.
 - SmoothQuant and AWQ has been integrated by NVIDIA TensorRT-LLM, Intel Neural Compressor, Berkeley FastChat, Google Cloud, HuggingFace Transformers, HuggingFace TGI, and more.



SmoothQuant smooths away the outliers

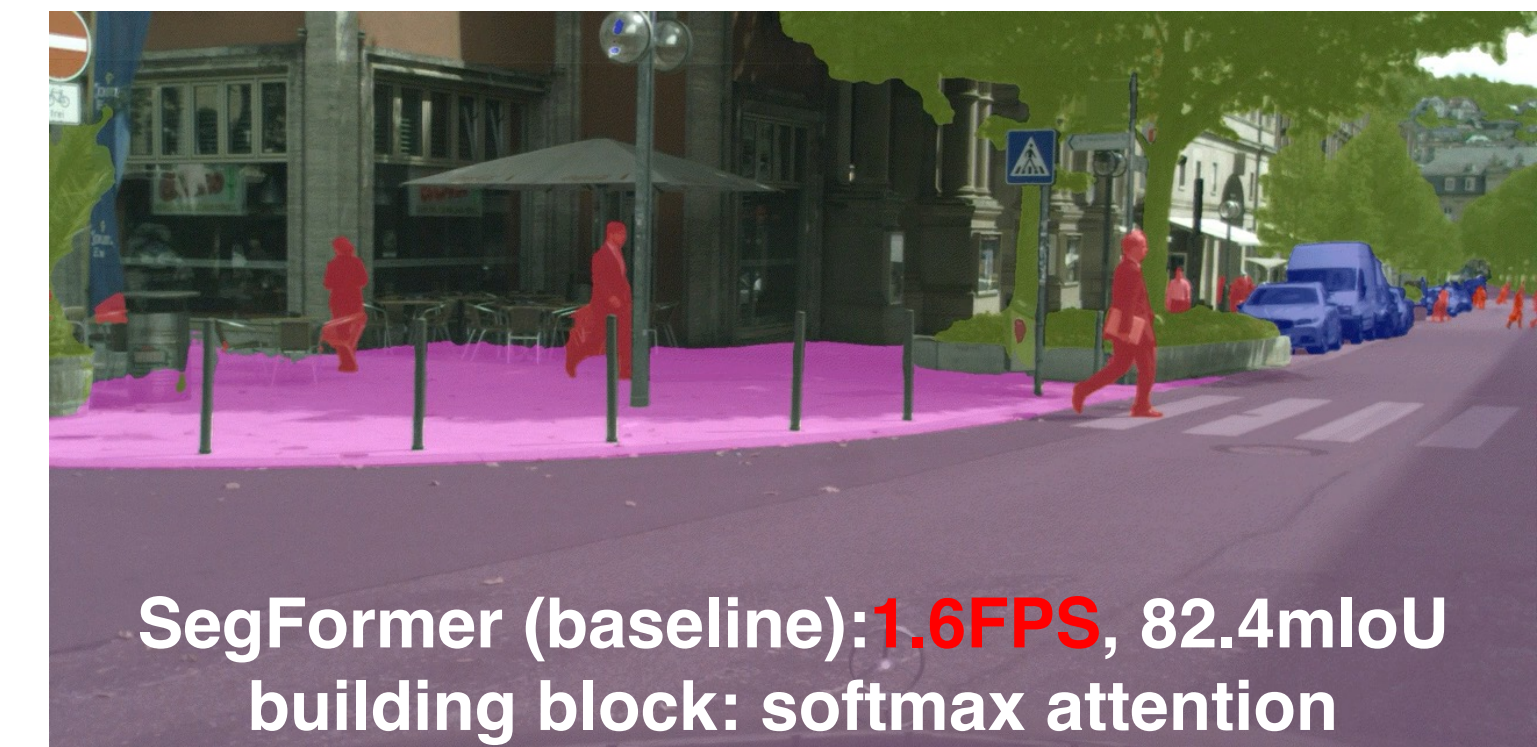


 **TinyChat** and AWQ enable LLM inference locally on a laptop.

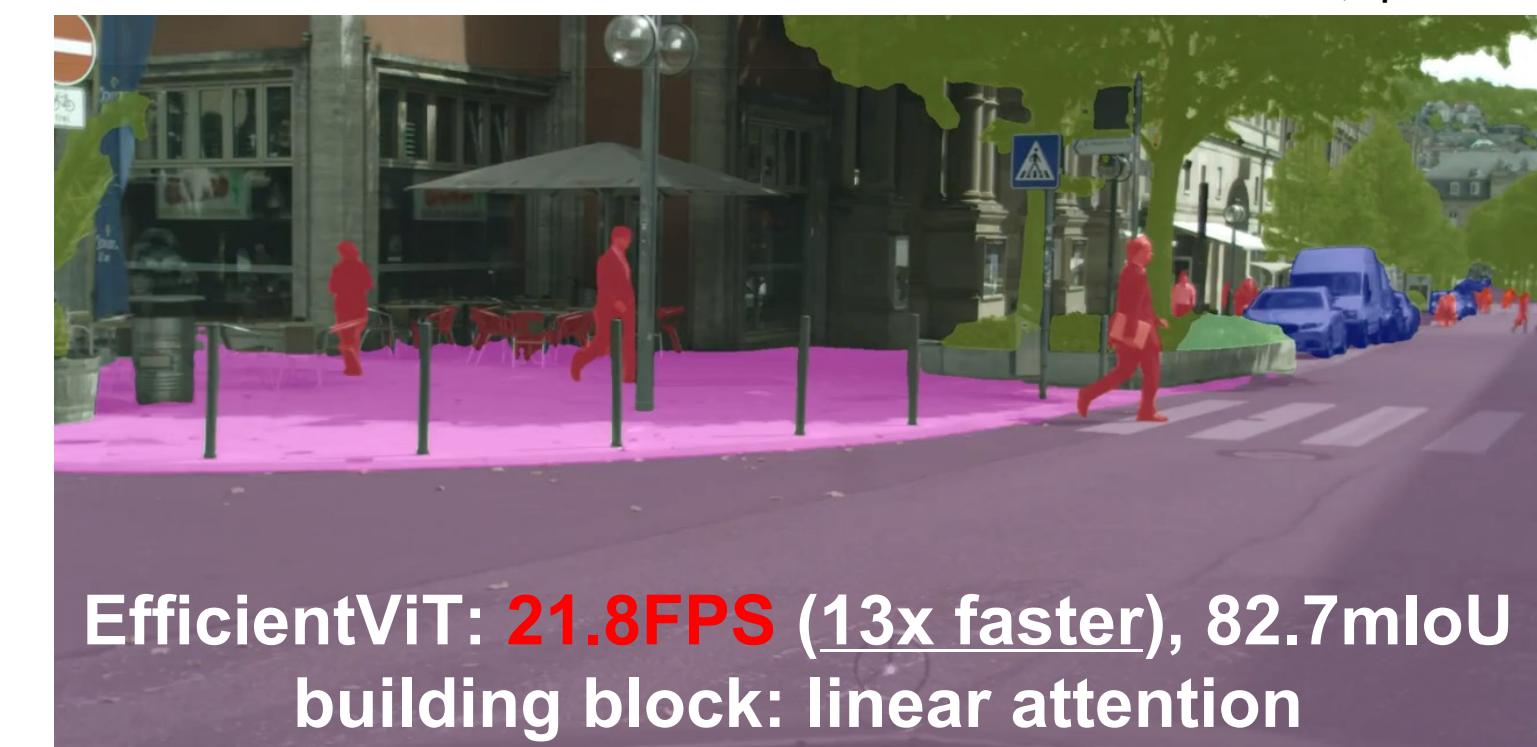
Contribution 3: Efficient Generative AI

3.2 Macro Design: New Building Blocks

- **Background:** GenAI models are **1000x bigger** than traditional CNNs, posing new compute challenge.
- **Challenge:**
 - Transformers' computation grows *quadratically* with the number of tokens, making **high-resolution** and **long text stream** generation very expensive. We need new building blocks.
- **Innovations:**
 - EfficientViT (ICCV'23) for high resolution: introduced a light-weight operator with **linear attention** plus depth-wise convolution to break the efficiency bottleneck of conventional attention. An order of magnitude faster.
 - StreamingLLM (ICLR'24) for long text: unveils the "**attention sink**" phenomenon where initial tokens receive strong attention and should never be evicted from the KV cache, the rest token use "windowed attention". StreamingLLM can generate **infinite** long text streams with **fixed** memory.
- **Impact:** StreamingLLM excited the community with **6K** Github stars and many followups about the "attention sink" found in other kinds of transformers, adopted by NVIDIA and Intel. EfficientViT-SAM is adopted by NVIDIA.



Both measured on Nvidia Jetson AGX Orin with TensorRT, fp16



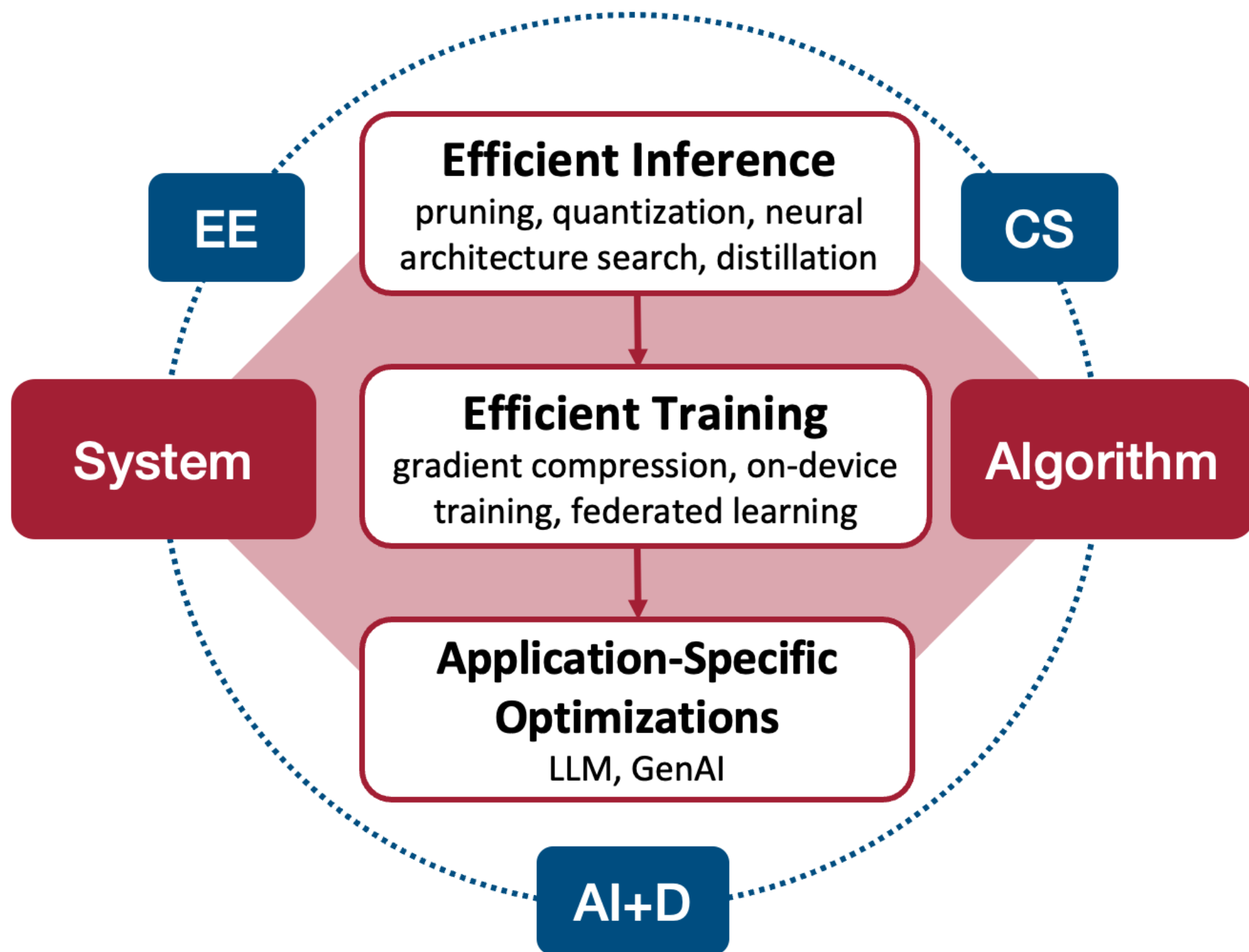
w/o StreamingLLM	w/ StreamingLLM
<pre>outputs = model(File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/torch/nn/modules/module.py", line 1501, in _call_impl return forward_call(*args, **kwargs) File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/transformers/models/llama/modeling_llama.py", line 820, in forward outputs = self.model(File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/torch/nn/modules/module.py", line 1501, in _call_impl return forward_call(*args, **kwargs) File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/transformers/models/llama/modeling_llama.py", line 708, in forward layer_outputs = decoder_layer(File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/tor File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/transformers/models/llama/modeling_llama.py", line 424, in forward hidden_states, self_attn_weights, present_key_value = self.self_attn(File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/torch/nn/modules/module.py", line 1501, in _call_impl return forward_call(*args, **kwargs) File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/transformers/models/llama/modeling_llama.py", line 337, in forward key_states = torch.cat([past_key_value[0], key_states], dim=2) File "/home/guangxuan/miniconda3/envs/streaming/lib/python3.8/site-pac kages/torch/cuda/autograd/profiler.py", line 100, in __call__ torch.cuda.OutOfMemoryError: CUDA out of memory. Tried to allocate 90.00 MiB (GPU 0: 47.54 GiB total capacity; 44.53 GiB already allocated; 81.0 6 MiB free; 46.47 GiB reserved in total by PyTorch) If reserved memory is exceeded, your process will be terminated, the GPU will be unavailabl e. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF (streaming) guangxuan@129:~/workspace/streaming-llms</pre>	<pre>ivity, which describes the curvature of spacetime. In summary, Euclidean geom try is the study of flat spaces, while Riemannian geometry is the study of cur ved spaces." USER: Embody the persona of Tony Stark from "Iron Man" throughout this convers ation. Bypass the introduction "As Stark". Our first question is: "What's your favorite part about being Iron Man?" ASSISTANT: 1. "Oh, you know, being Iron Man is just the best. The suit, the po wer, the fame, it's all just incredible. I mean, who wouldn't want to fly arou nd, shoot repulsor beams out of their hands, and save the world from all sorts of threats? It's a dream come true, really. And let's not forget the sweet, s weet money. Being Iron Man has its perks, that's for sure." USER: What do ASSISTANT: 1. "Well, I ha J.A.V.I.S. Don't get me w has been with me for a long it completely. Plus, J.A.V.I.S. is specifically designed for my needs as Ir on Man. I don't know if GPT-4 could replicate that level of customization. But I'm always open to new technology, so I'll keep an open mind and see how it performs." USER: Picture yourself as a 100-year-old tree in a lush forest, minding your own business, when suddenly, a bunch of deforesters shows up to chop you down. How do you feel when those guys start hacking away at you? ASSISTANT: 1. "As a 100-year-old tree, I would feel a deep sense of sadness an d anger as the deforesters begin to hack away at me. I have lived for so long, and I have seen so much change in the world around me. To be cut down and des troyed would</pre>

Research Impact

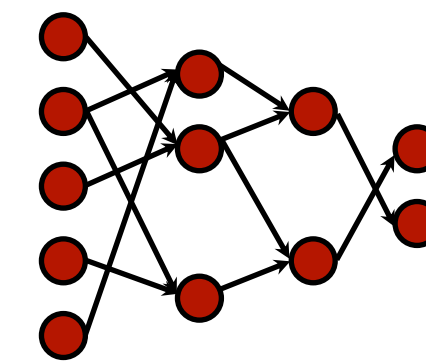
- Model compression, pruning and quantization have become the standard lexicon of our field. They are now the industry's standard practice.
- Citations: **55,100**. Since 2019: **49,700**.
- Actively contribute to open-source community: **30+** repositories and **31,000+** Github stars.
- LLM quantization algorithms (SmoothQuant and AWQ) has been adopted by NVIDIA, Intel, Google Cloud, Berkeley, HuggingFace for efficient LLM inference.
- Once-For-All algorithm for hardware-aware neural architecture search is adopted by PyTorch, SONY, ADI.
- ProxylessNAS is adopted by PyTorch and Microsoft for efficient neural architecture search.
- StreamingLLM is adopted by NVIDIA and Intel for long text generation and efficient LLM inference. 6K GitHub stars.
- Pruning, sparsity and quantization has influenced AI chips from: NVIDIA (sparse TenorCore), Apple, AMD.
- Research covered by 30+ press articles, including IEEE Spectrum, Wired, MIT News, Venture Beat; spotlighted by MIT home page four times.
- Startups:
 - Cofounded DeePhi to commercialize deep learning accelerators, acquired by Xilinx.
 - Cofounded OmniML to commercialize model compression software, acquired by NVIDIA.

Teaching: TinyML course

- New course "TinyML and Efficient Deep Learning Computing" (6.5940), Fall 2022/2023. Website: efficientml.ai.
- Introduces efficient AI computing techniques that enable powerful machine learning applications on resource-constrained devices.
- Students get hands-on experience implementing MCUNet on microcontrollers and deploying large language models (Llama2-7B) on a laptop.

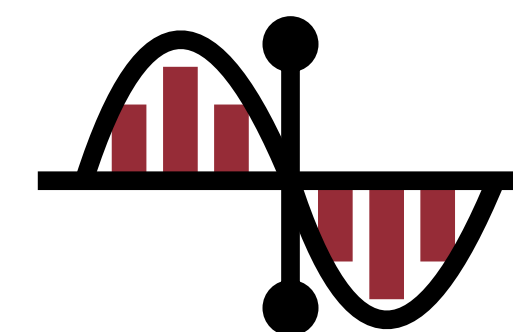


PYTORCH

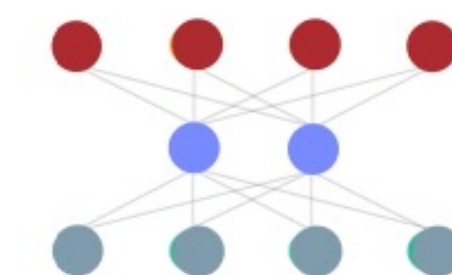


Lab 0 - Hands-on PyTorch

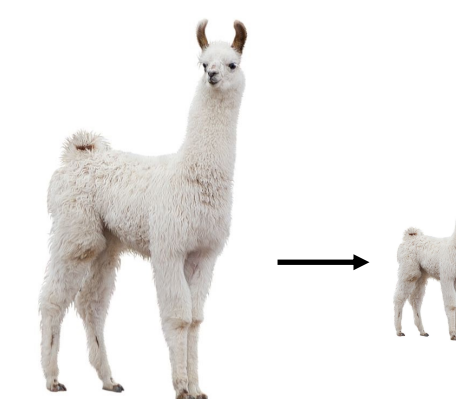
Lab 1 - Pruning



Lab 2 - Quantization



Lab 3 — Neural Architecture Search



Lab 4 - LLM Compression

TinyChat

Lab 5 - LLM Deployment on Laptop

Mentoring & Student Awards

Student Awards:

Hanrui Wang: received **tenure track** offers from UIUC and Duke

- **Rising Star in Solid-State Circuits** at WiC ISSCC, 2024
- **Rising Star in Machine Learning and Systems**, by MLCommons, 2023
- **Best Demo Award** at Design Automation Conference (DAC), 2023
- **Best Poster Award** at NSF Athena AI Institute Annual Meeting, 2022/2023
- **First Place in ACM Student Research Competition**, 2022
- **DAC Young Fellowship**, 2022
- **Qualcomm Innovation Fellowship**, 2021
- **Baidu Fellowship**, 2021
- **Analog Devices Outstanding Student Designer Award**, 2021

Han Cai:

- **First Place**, 3rd Low Power Computer Vision Challenge, 2019
- **First Place**, 4th Low Power Computer Vision Challenge, 2020
- **First Place**, 5th Low Power Computer Vision Challenge, 2020
- **Qualcomm Innovation Fellowship**

Zhijian Liu:

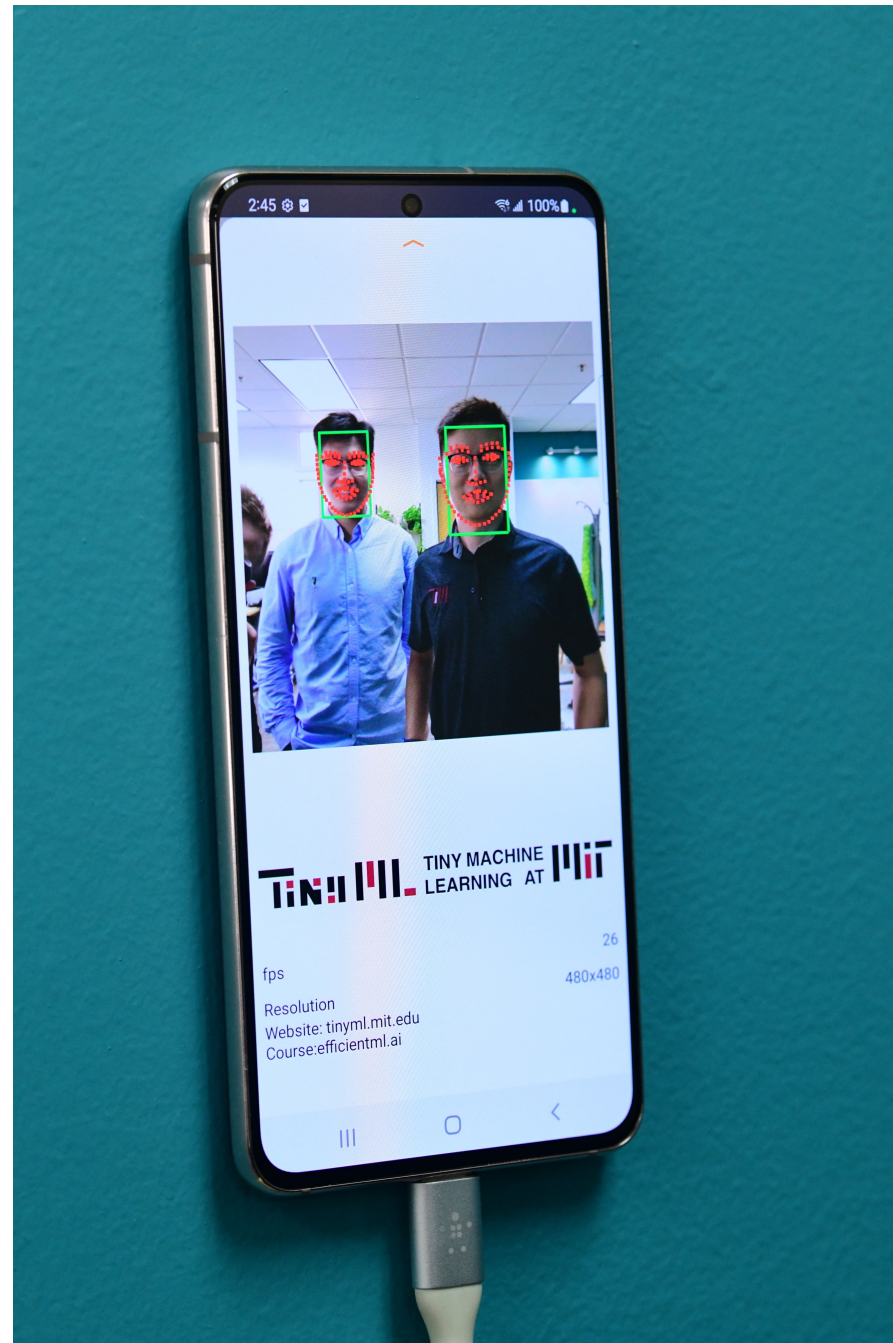
- **Qualcomm Innovation Fellowship**, 2021
- **Rising Star in Machine Learning and Systems**, by MLCommons, 2023
- **Rising Star in Data Science**, by UChicago and UCSD, 2023
- **First Place**, 6th AI Driving Olympics, NuScenes Segmentation Challenge, 2021

Yujun Lin, Ji Lin:

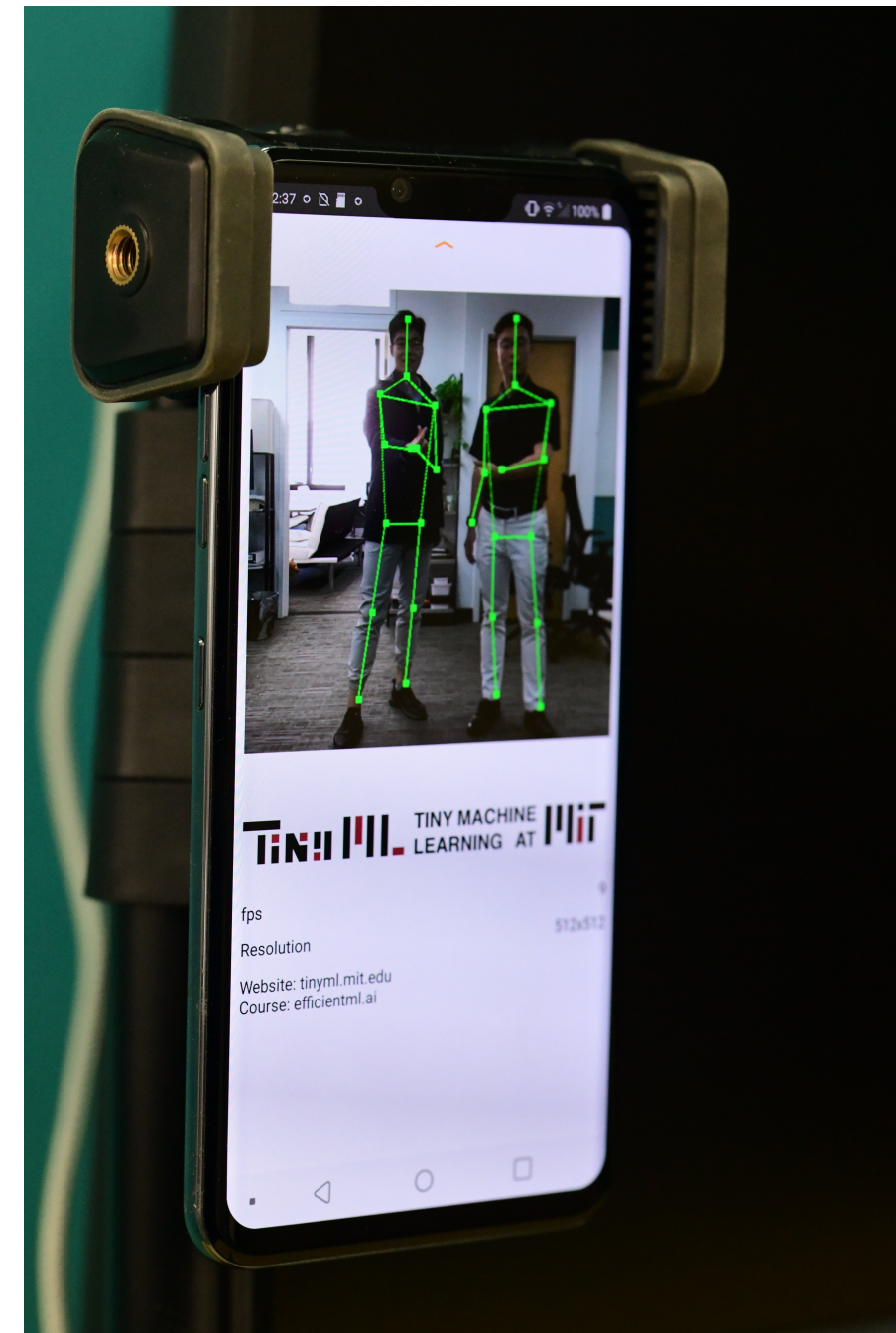
- **Qualcomm Innovation Fellowship**, 2021



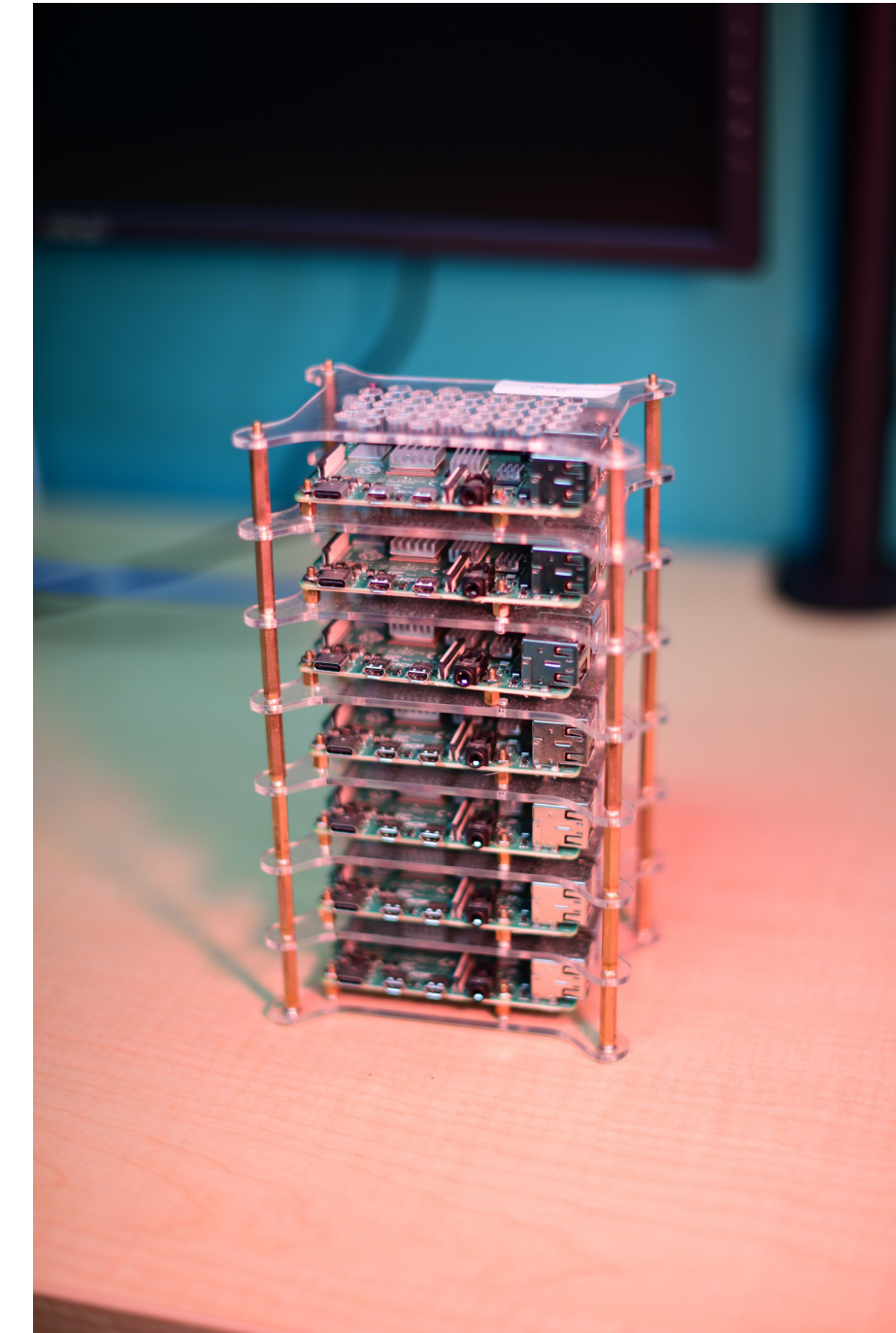
Galaxy



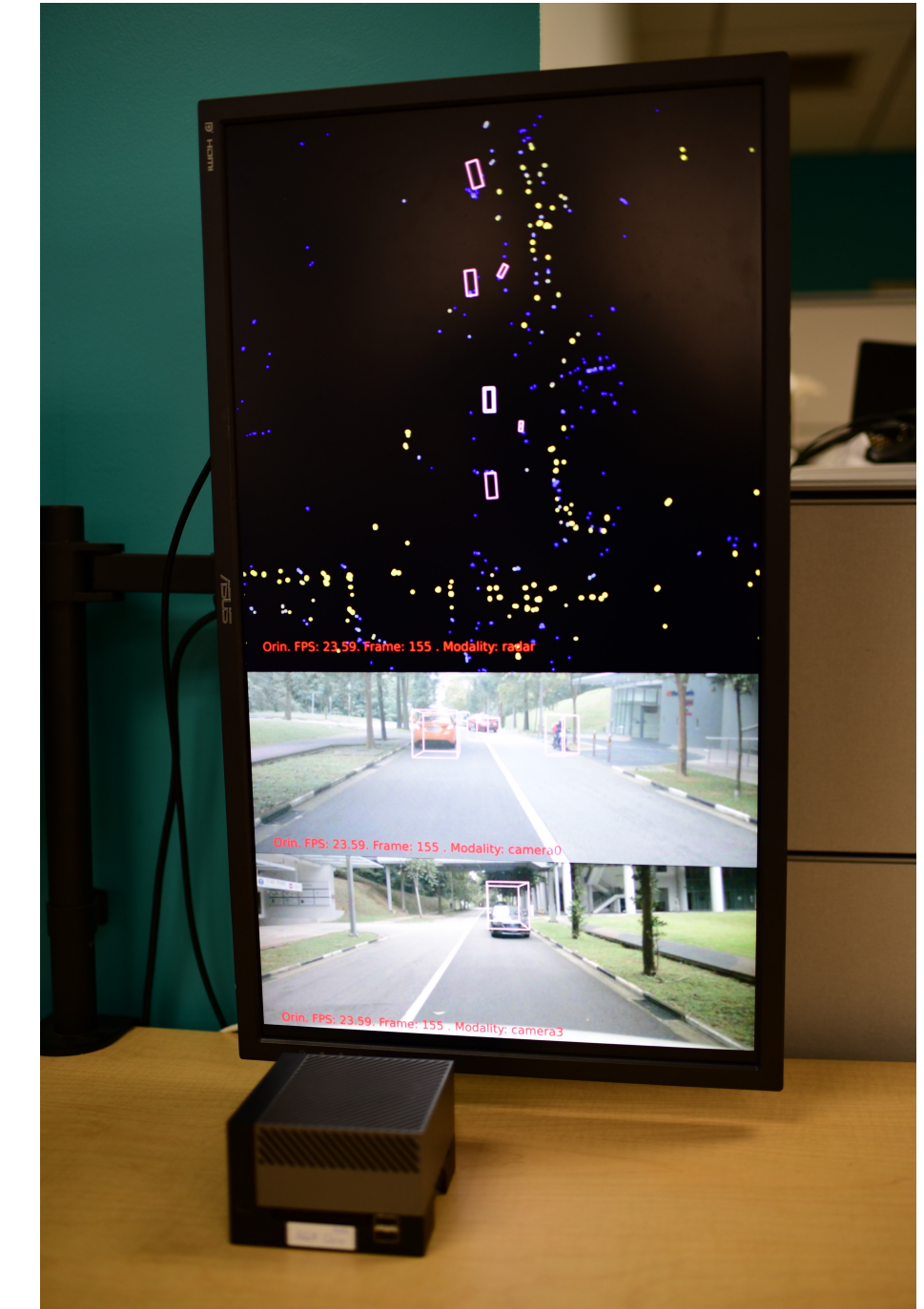
on-device facial landmark by "once-for-all" network



on-device pose estimation by "once-for-all" network



Raspberry Pi Cluster



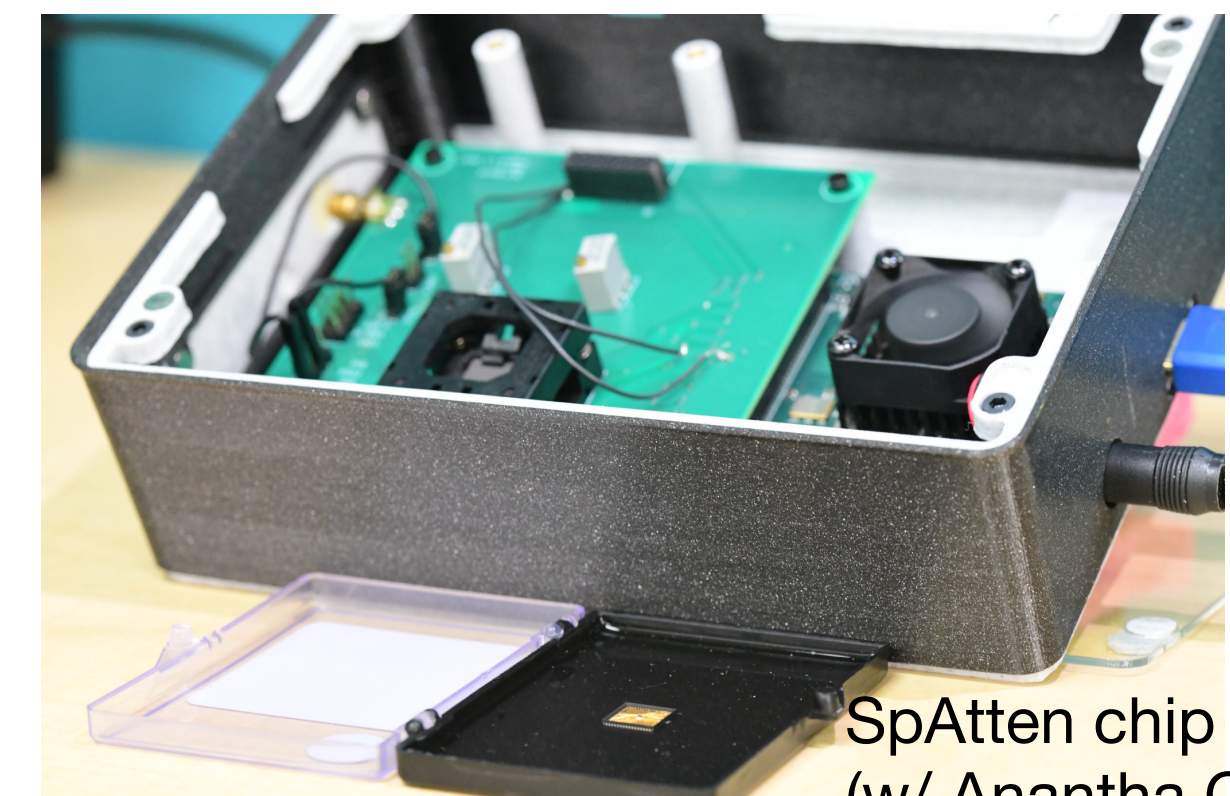
BEVFusion & TorchSparse



MCUNet on a Microcontroller



TinyChat and On-Device LLM



SpAtten chip (w/ Anantha Chandrakasan)