

Accelerating Large Language Model and Generative AI

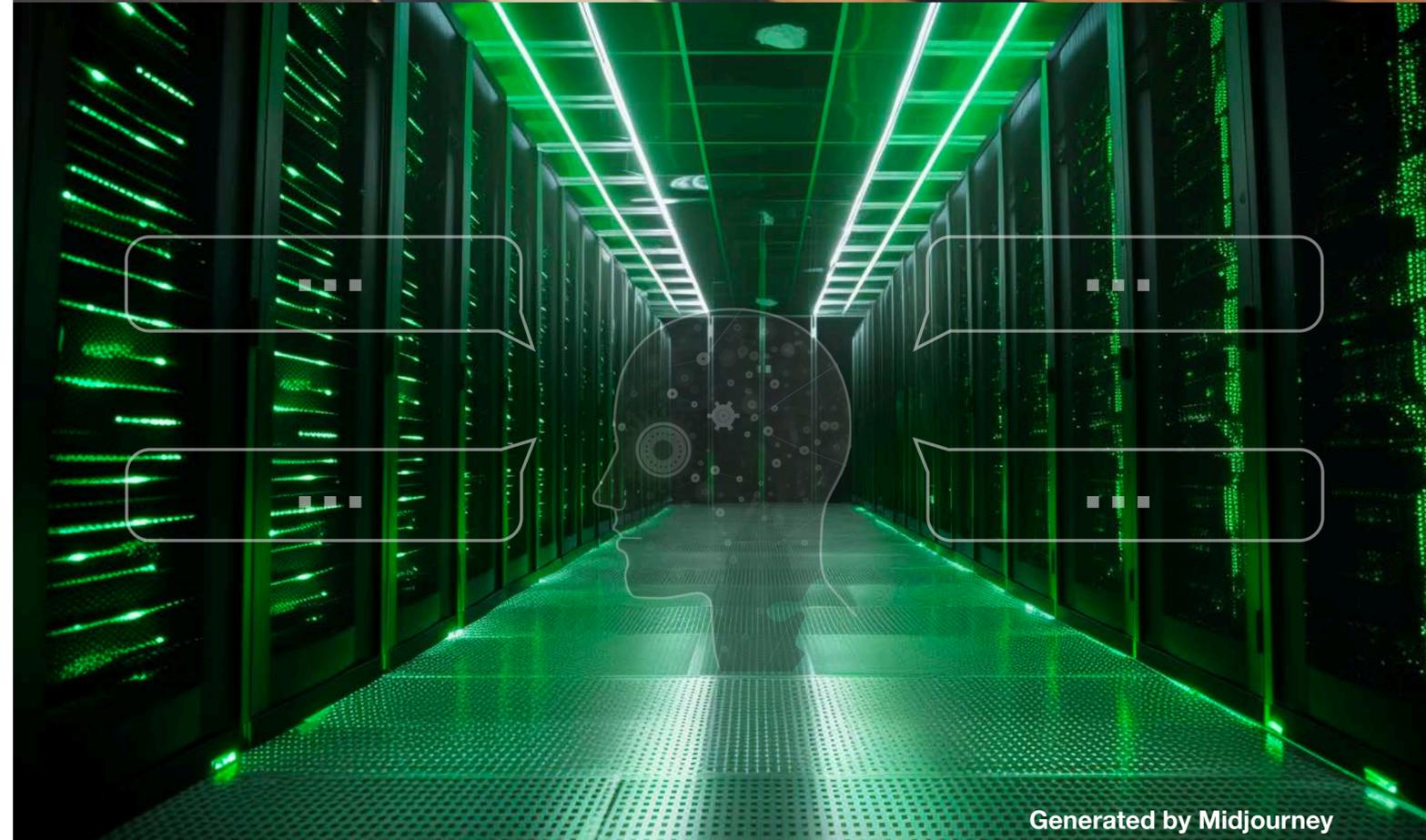


Song Han

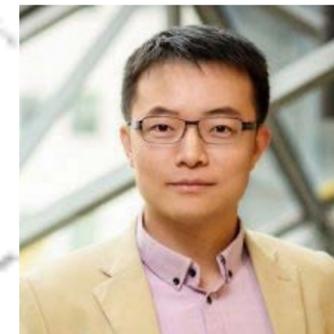
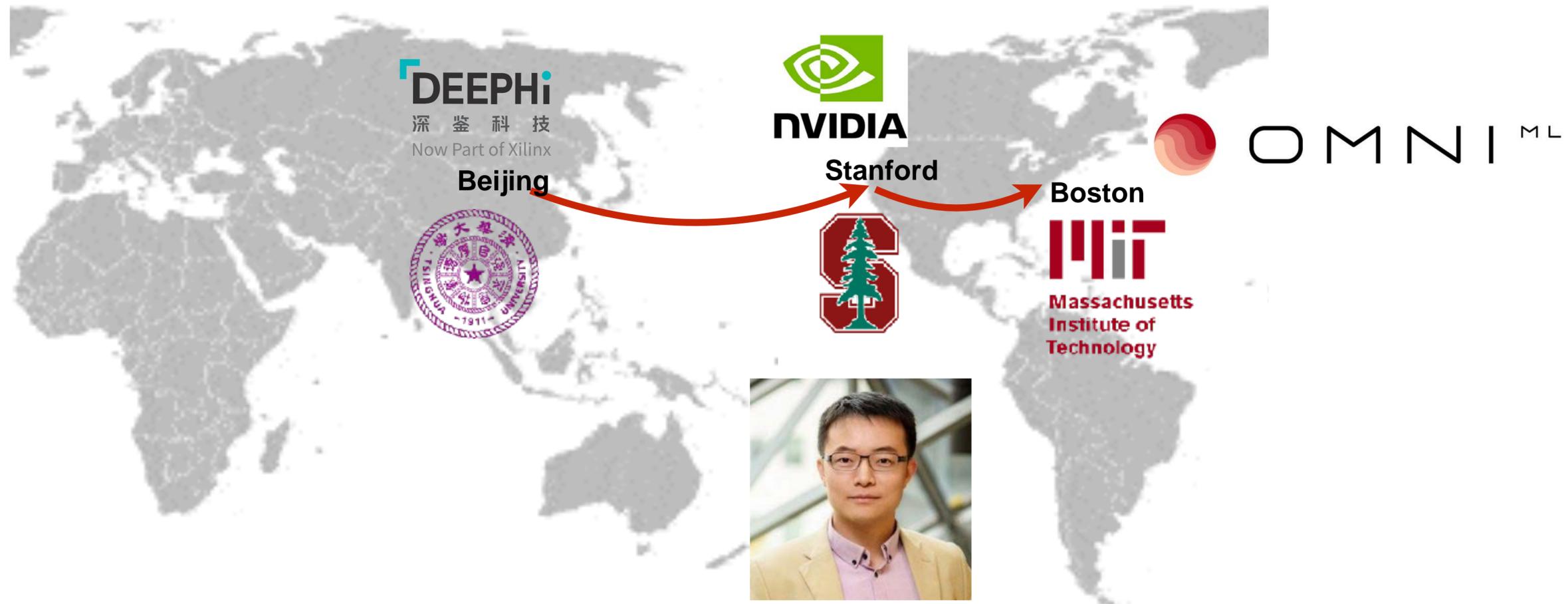
Associate Professor, MIT
Distinguished Scientist, NVIDIA

<https://songhan.mit.edu>

 @SongHan_MIT



Prof. Song Han

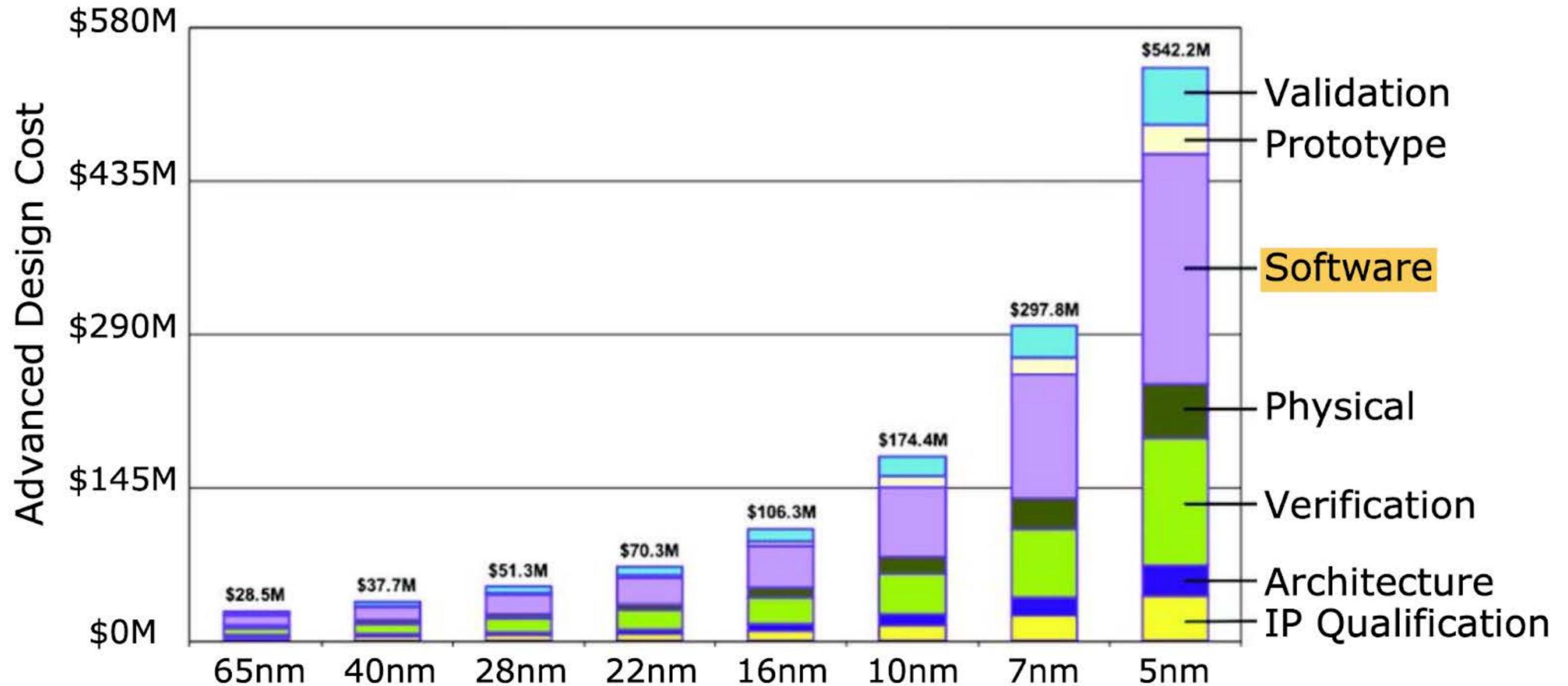


- B.S. from Tsinghua University
- Ph.D. from Stanford University, advised by Prof. Bill Dally
- Deep Compression (best paper award);
- EIE (top5 cited paper in 50 years of ISCA)

- Cofounder of DeePhi Tech (acquired by Xilinx/AMD)
- Cofounder of OmniML (acquired by NVIDIA)

- MIT Technology Review, 35 Innovators under 35
- NSF Career Award
- IEEE "AIs 10 to Watch: The Future of AI" Award
- Sloan Research Fellowship

Software is Important in Advanced Technology Node

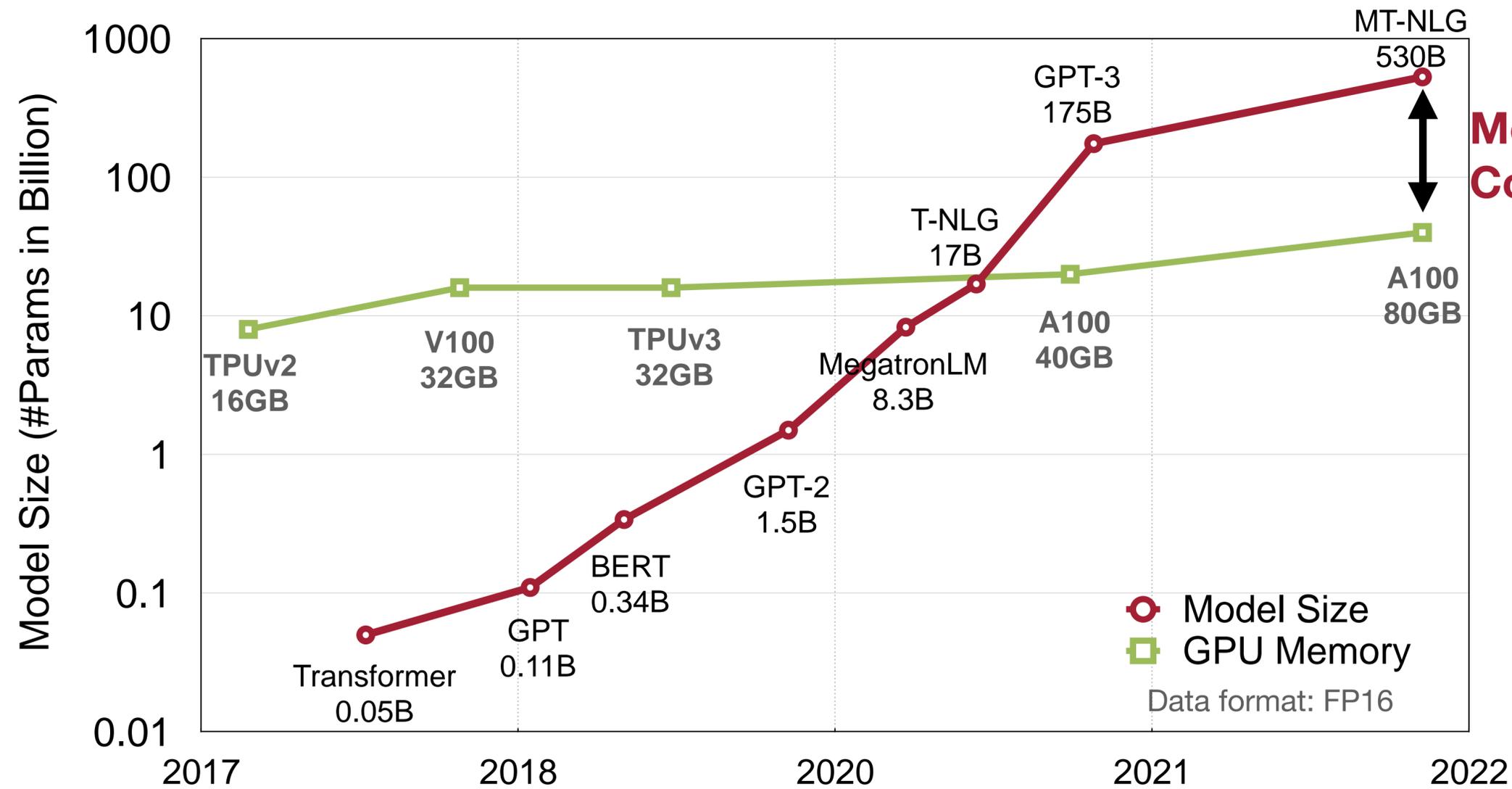


The software cost dominates the cost breakdown of advanced technology nodes [source].

We focus on designing new algorithms and software for efficient computing.

Model Compression and Co-Design are Essential

Bridge the Gap between the Supply and Demand of AI Computing



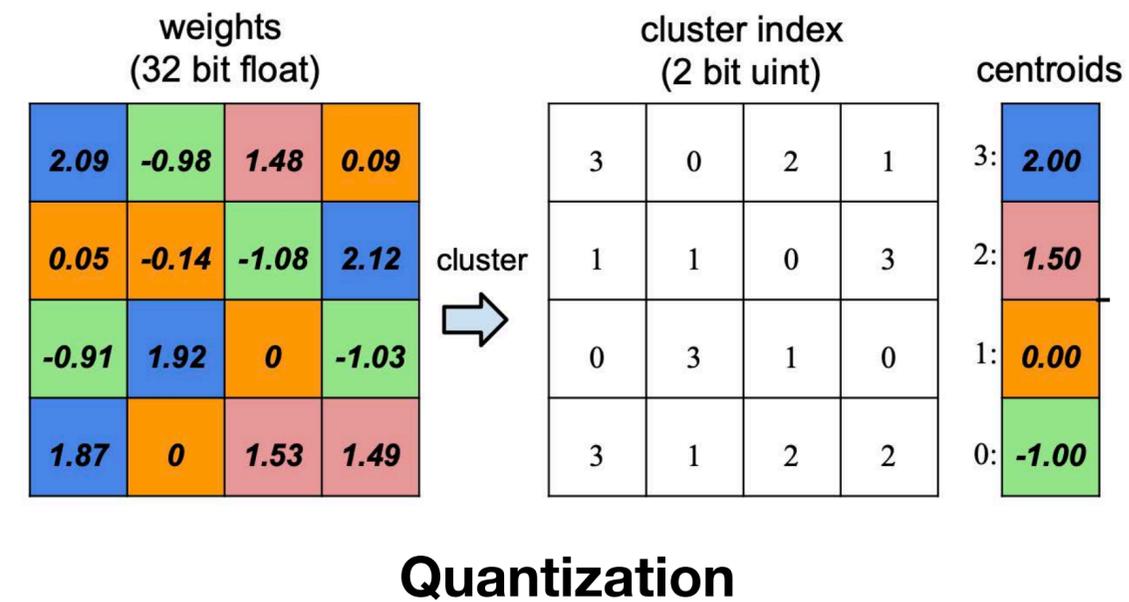
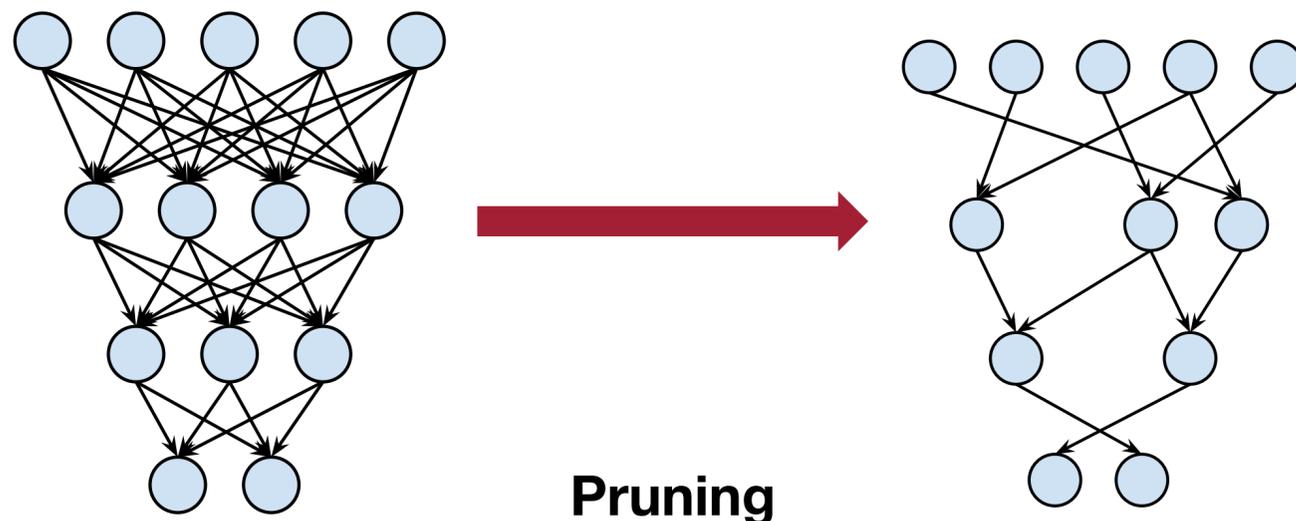
Model Compression



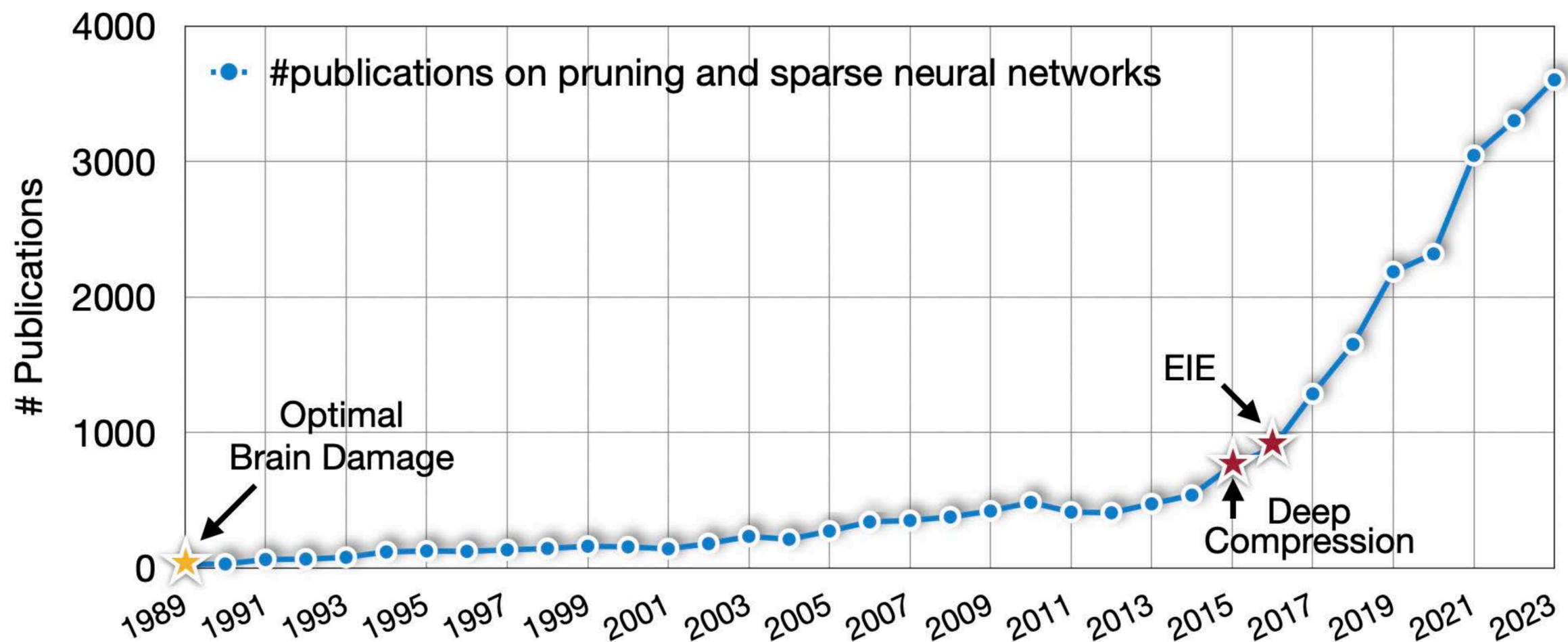
208V, 50Amp:

2x NVIDIA DGX A100
or
1x NVIDIA DGX H100
10KW!

Deep Compression



#publications on pruning and sparsity



The number of publications on neural network pruning and sparsity quickly increased since 2015, including both algorithms and systems.

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

Top-5 most cited papers in 50 years of ISCA

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

Efficient Large Language Models

LLM serving cost is extremely high

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

ChatGPT is at capacity right now

[Get notified when we're back](#)

Write a rap about the status of ChatGPT.

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

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

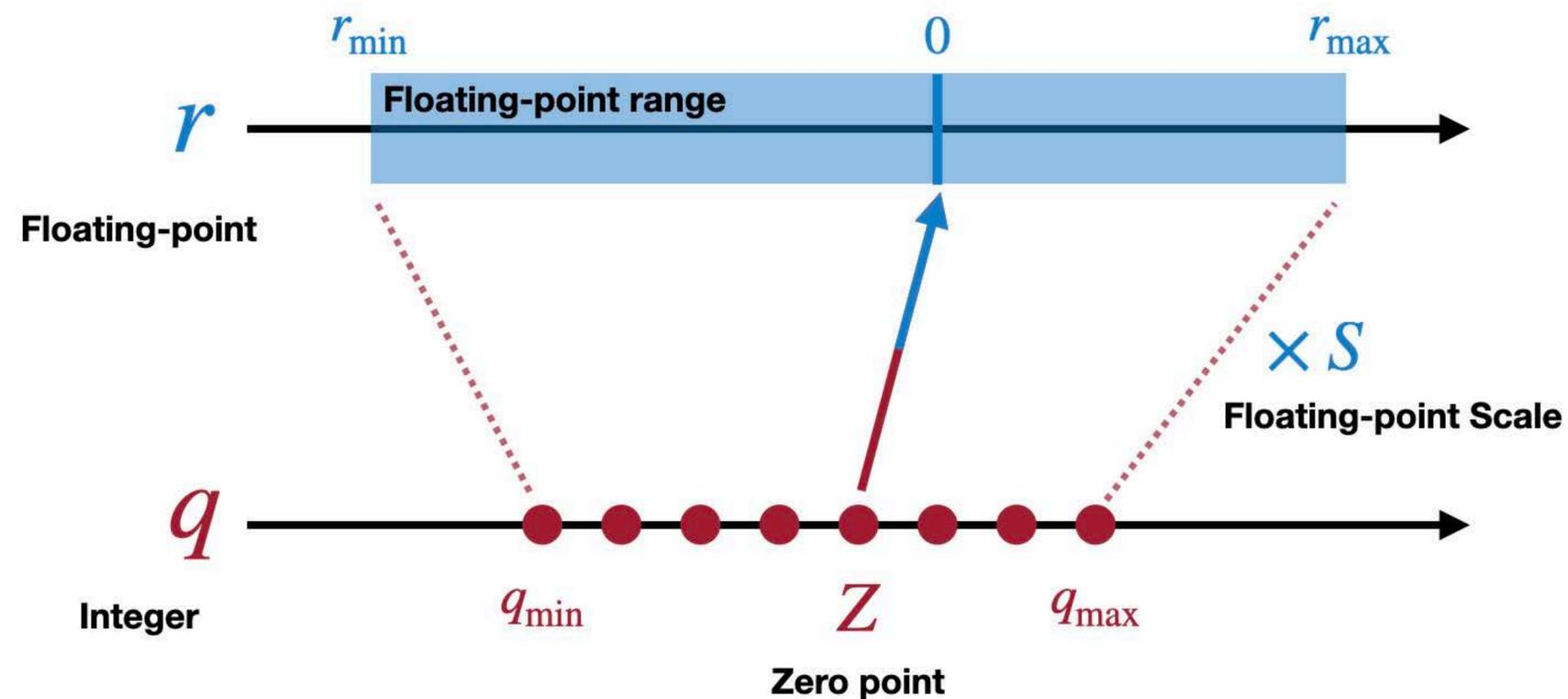
So don't you worry, don't be mad
We're working hard, it's not so bad
Just give us time, we'll be back
ChatGPT, the future of chat, that's a fact.

Your Account

<h4>Free Plan</h4> <p>Your Current Plan</p> <ul style="list-style-type: none">✓ Available when demand is low✓ Standard response speed✓ Regular model updates	<h4>ChatGPT Plus USD \$20/mo</h4> <p>Upgrade plan</p> <p>Due to high demand, we've temporarily paused upgrades.</p> <ul style="list-style-type: none">✓ Priority access to new features
--	---

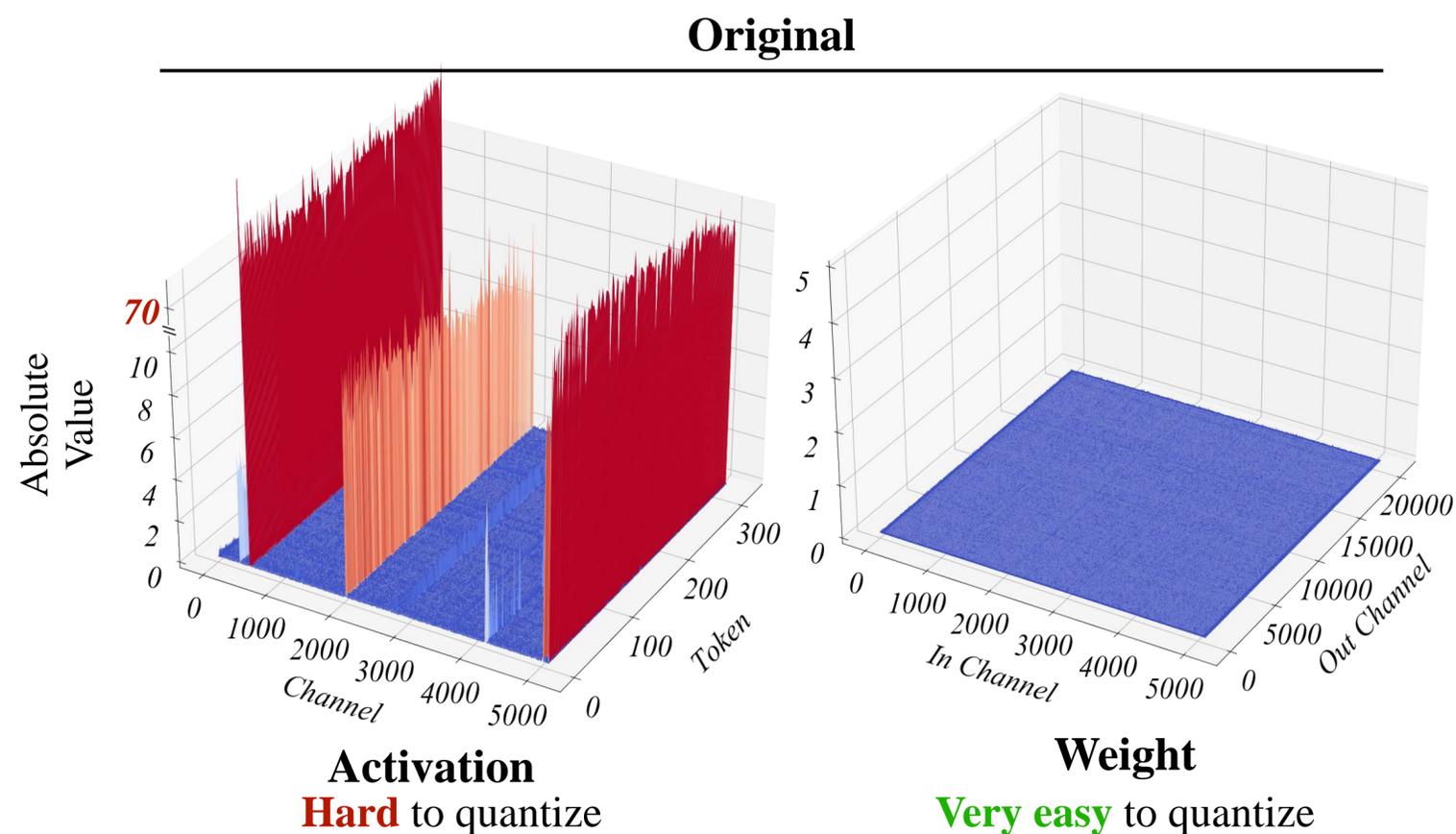
Quantization Can Reduce Deployment Costs

- Serving a 175B GPT-3 model at least requires:
 - FP16: 350GB memory → 5 x 80GB A100 GPUs
 - **INT8: 175GB memory → 3 x 80GB A100 GPUs**



SmoothQuant

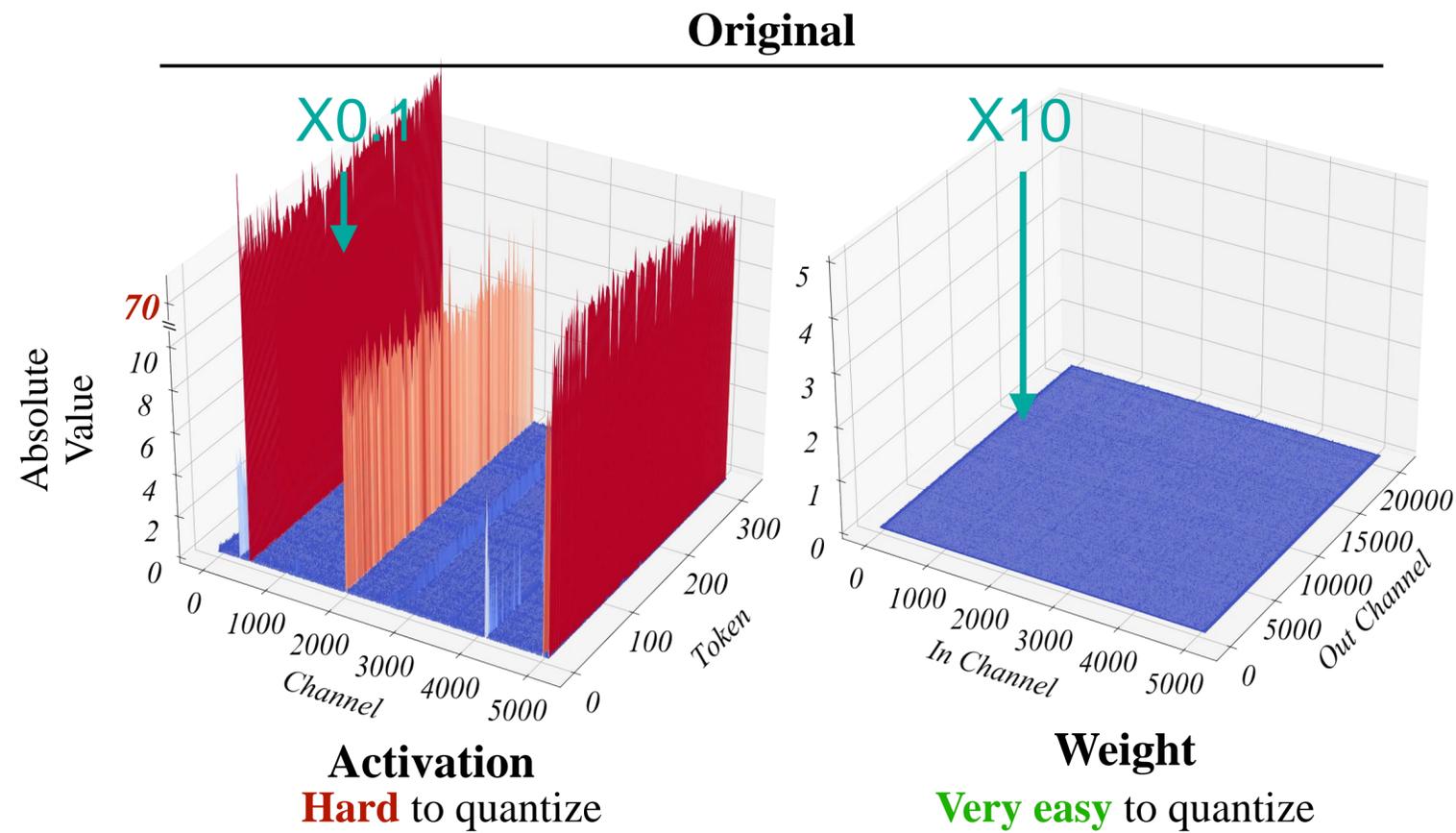
Smoothing activation to reduce quantization error



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

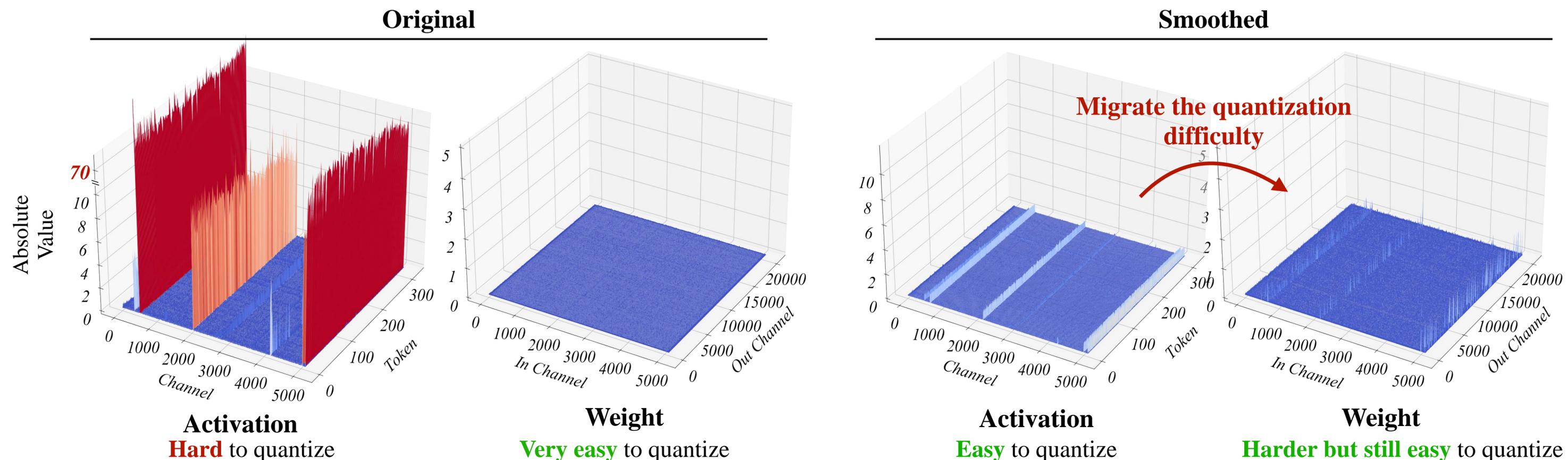
SmoothQuant

Smoothing activation to reduce quantization error



SmoothQuant

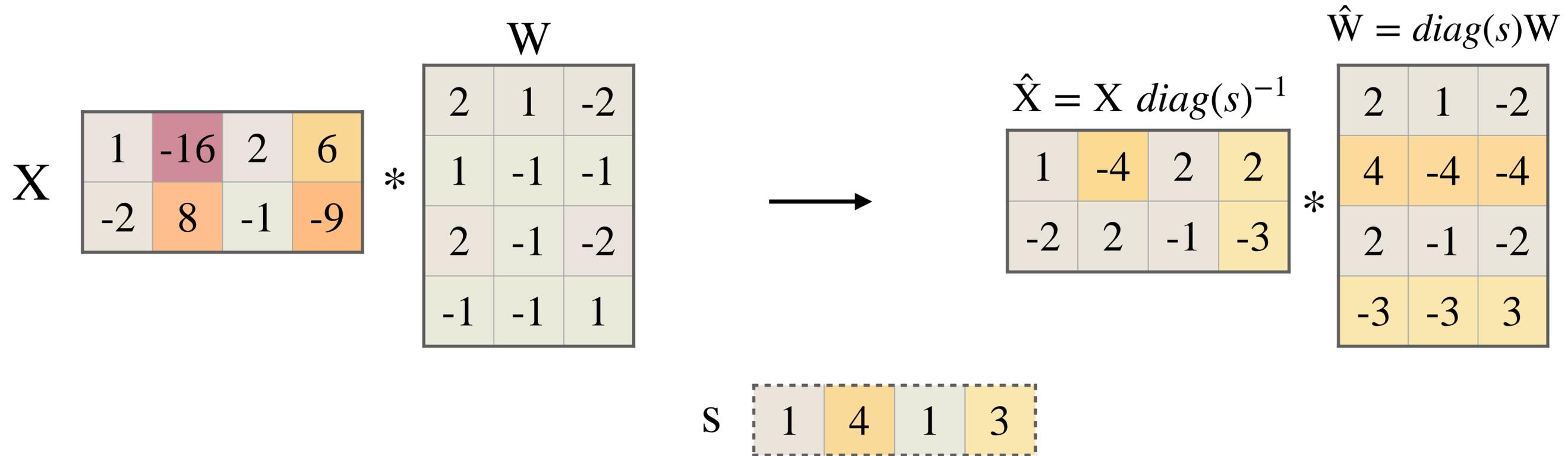
Smoothing activation to reduce quantization error



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

SmoothQuant

Smoothing activation to reduce quantization error



$$s_j = \max(|\mathbf{X}_j|)^\alpha / \max(|\mathbf{W}_j|)^{1-\alpha}, j = 1, 2, \dots, C_i$$

$$\mathbf{Y} = (\mathbf{X} \text{diag}(\mathbf{s})^{-1}) \cdot (\text{diag}(\mathbf{s})\mathbf{W}) = \hat{\mathbf{X}}\hat{\mathbf{W}}$$

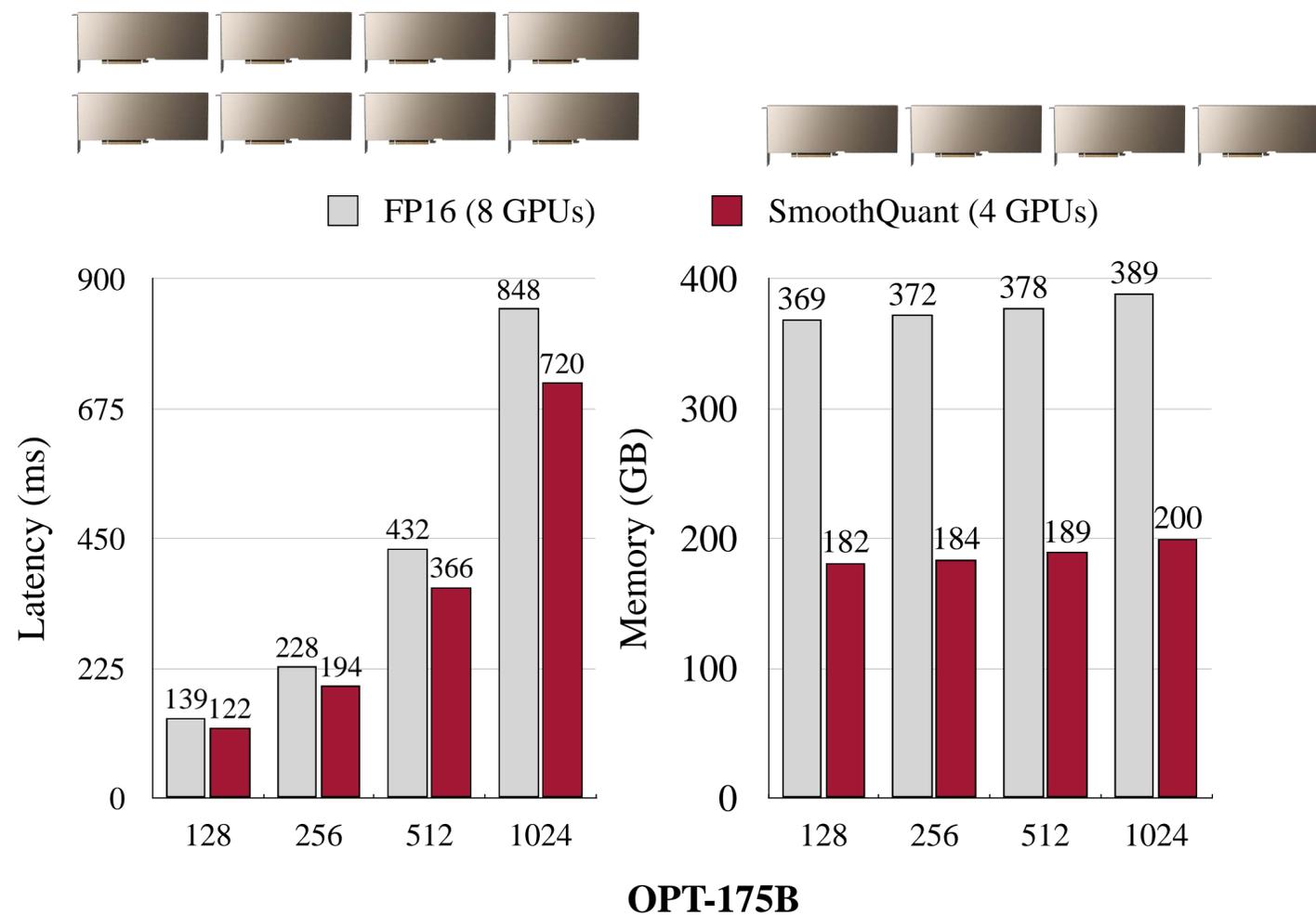
SmoothQuant (W8A8)

Accurate and efficient quantization of various LLMs

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

LAMBADA Accuracy

	OPT-175B	BLOOM-176B	GLM-130B
FP16	71.6%	68.2%	73.8%
SmoothQuant	71.2%	68.3%	73.7%



SmoothQuant (W8A8)

Scaling up: W8A8 quantization of MT-NLG 530B

MT-NLG 530B Accuracy

	LAMBADA	HellaSwag	PIQA	WinoGrande	Average
FP16	76.6%	62.1%	81.0%	72.9%	73.1%
INT8	77.2%	60.4%	80.7%	74.1%	73.1%

MT-NLG 530B Efficiency

SeqLen	Prec.	#GPUs	Latency	Memory
512	FP16	16	838ms	1068GB
	INT8	8	839ms	545GB
1024	FP16	16	1707ms	1095GB
	INT8	8	1689ms	570GB

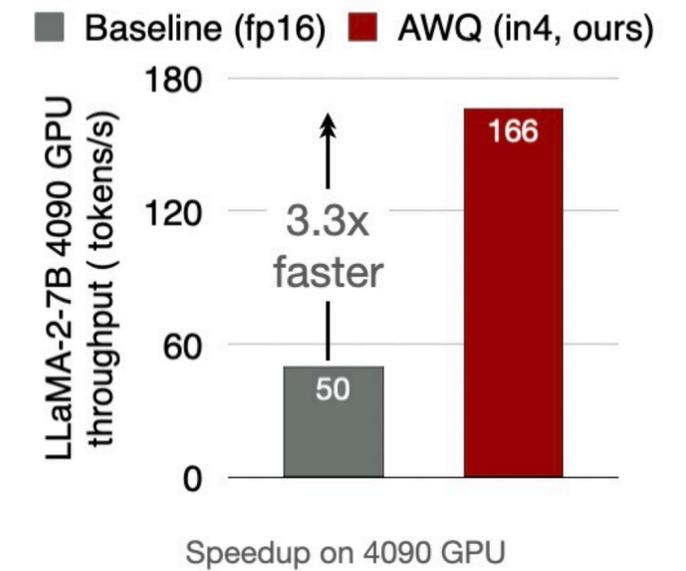


- SmoothQuant can accurately quantize MT-NLG 530B model and reduce the serving GPU numbers by half at a similar latency, which allows serving the 530B model within a single node.

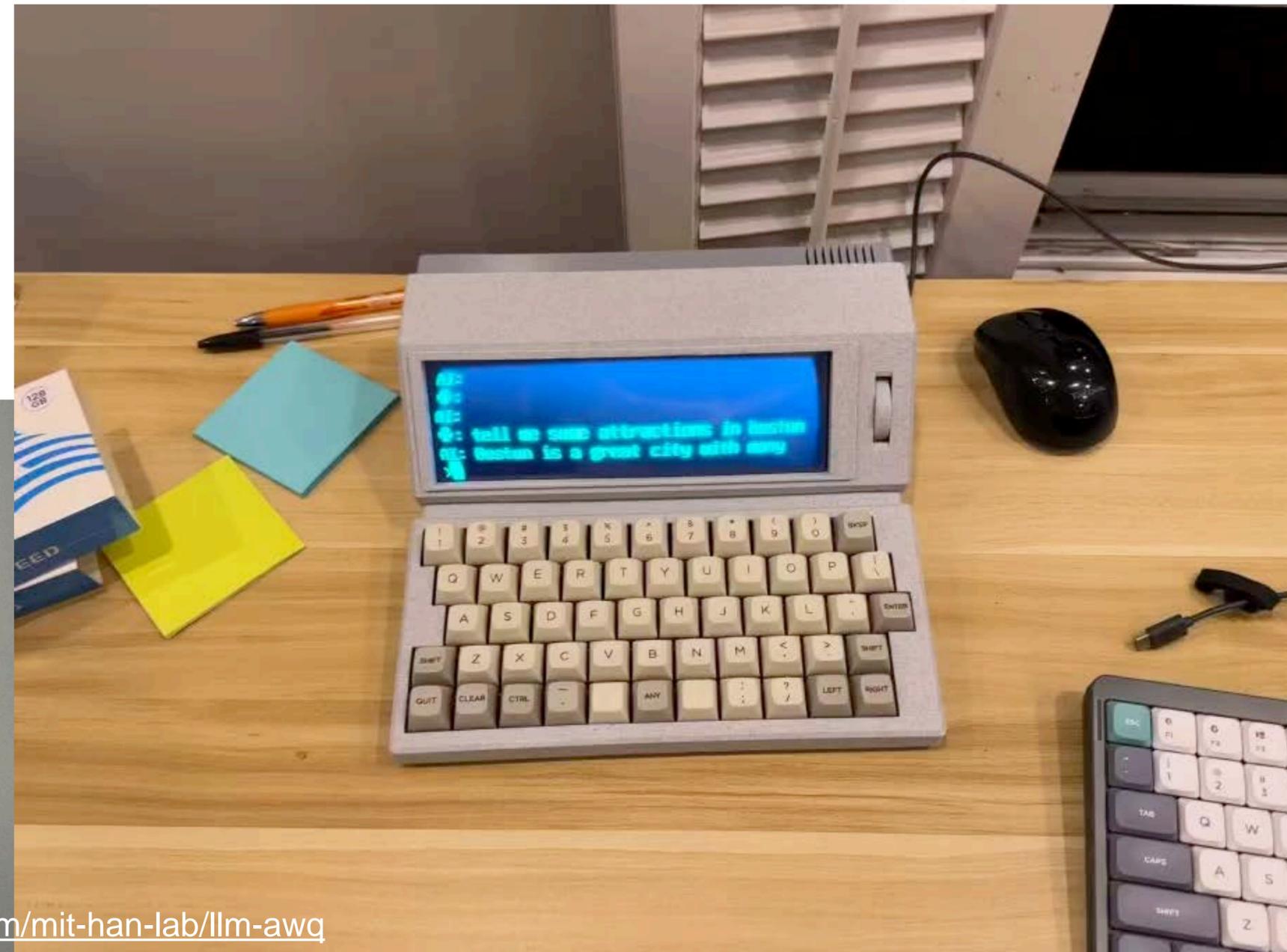


TinyChat

- A LightWeight Chatbot for LLMs on the edge

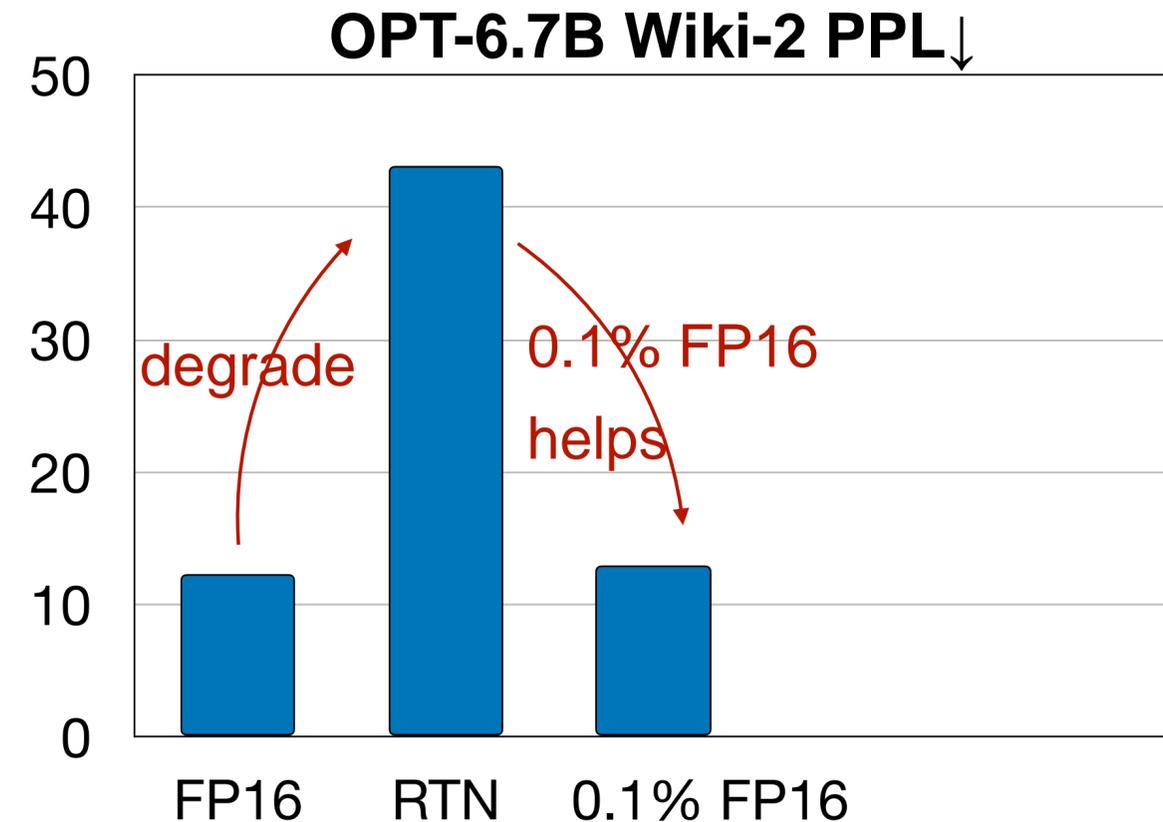
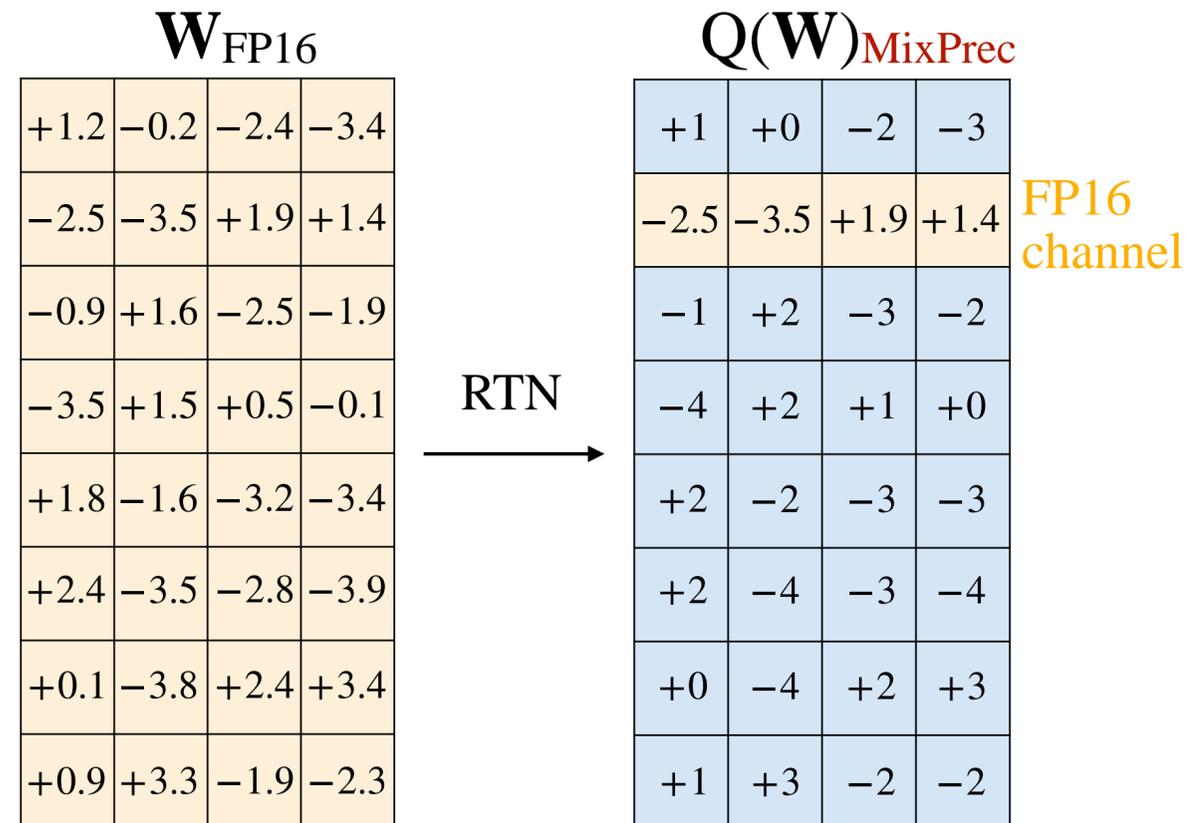


- Deploying LLM on the edge is useful: running copilot services (code completion, office, game chat) locally on laptops, cars, robots, and more. These devices are **resource-constrained**, **low-power** and sometimes **do not have access to the Internet**.
- **Data privacy** is important. Users do not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



AWQ: 4bit Activation-aware Weight Quantization

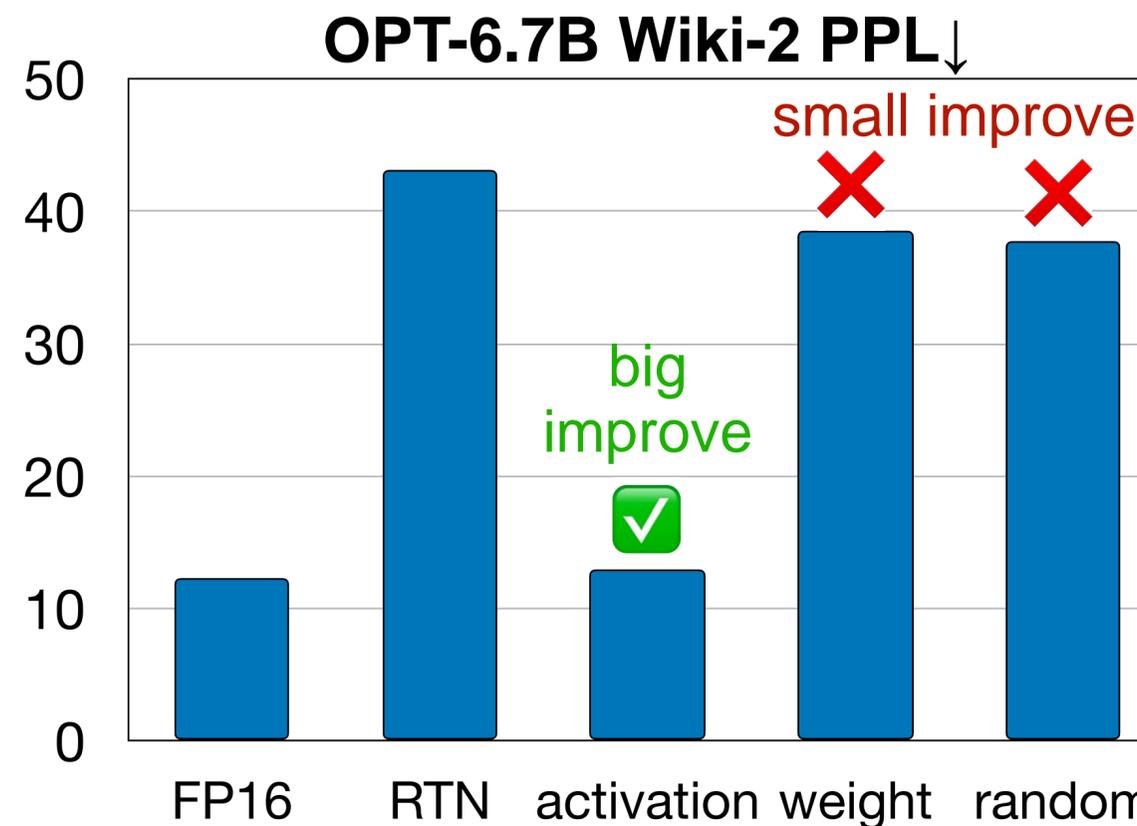
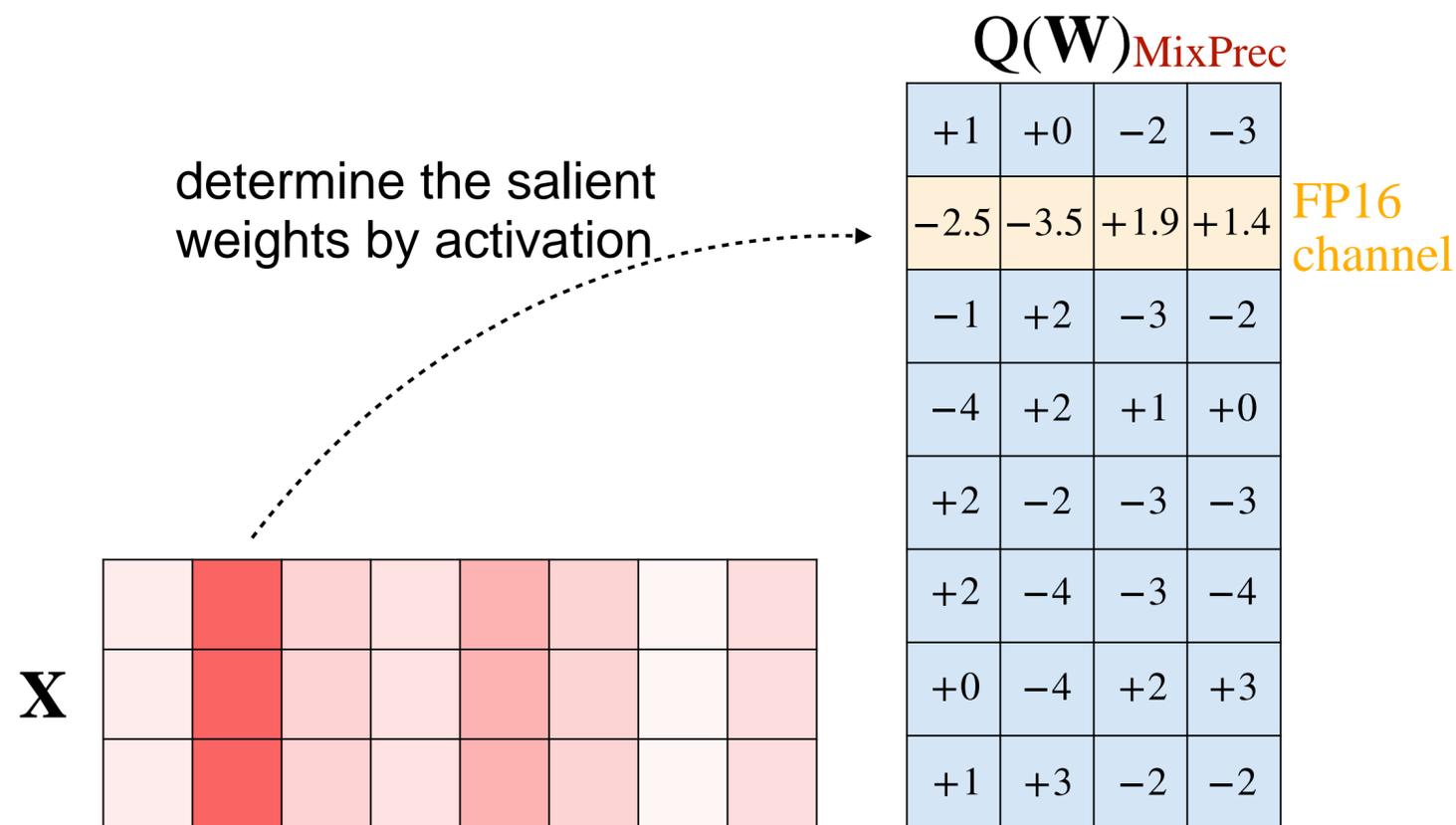
Observation: Weights are not equally important; 0.1% salient weights



- We find that weights are not equally important, keeping **only 0.1%** of salient weight channels in FP16 can greatly improve perplexity
- But how do we select salient channels? Should we select based on weight magnitude?

AWQ: 4bit Activation-aware Weight Quantization

Salient weights are determined by activation distribution, not weight



0.1% FP16 based on

- We find that weights are not equally important, keeping **only 0.1%** of salient weight channels in FP16 can greatly improve perplexity
- But how do we select salient channels? Should we select based on weight magnitude?
- No! We should look for **activation distribution**, but not **weight!**

AWQ: 4bit Activation-aware Weight Quantization

Protecting salient weights by scaling (no mixed prec.)

		W				
Q(+1.2	-0.2	-2.4	-3.4	× 1
		-2.5	-3.5	+1.9	+1.4	× 2
		-0.9	+1.6	-2.5	-1.9	× 1
		-3.5	+1.5	+0.5	-0.1	× 1
		+1.8	-1.6	-3.2	-3.4	× 1
		+2.4	-3.5	-2.8	-3.9	× 1
		+0.1	-3.8	+2.4	+3.4	× 1
		+0.9	+3.3	-1.9	-2.3	× 1

fuse to previous op

WX → **Q(W · s)(s⁻¹ · X)**

$$Q(\mathbf{w}) \cdot x = \Delta \cdot \text{Round}\left(\frac{\mathbf{w}}{\Delta}\right) \cdot x, \quad \Delta = \frac{\max(|\mathbf{w}|)}{2^{N-1}}$$

$$Q(\mathbf{w} \cdot s)(x/s) = \underbrace{\Delta'}_{\text{constant E(error)}} \cdot \text{Round}(\mathbf{w}s/\Delta') \cdot x \cdot \underbrace{\frac{1}{s}}_{\text{suppress error}}$$

$$\Delta' \approx \Delta, \quad \text{RoundErr} \sim 0.25, \quad s > 1$$

$$\frac{\text{Err}(Q(\mathbf{w} \cdot s)(x/s))}{\text{Err}(Q(\mathbf{w}) \cdot x)} = \frac{\Delta'}{\Delta} \cdot \frac{1}{s} < 1$$

- Multiplying the salient channels with $s > 1$ reduces its quantization error
- Skip mathematical derivation for now

- TinyChatEngine implements the compressed AWQ 4bit model, built from C/C++ from scratch, easy to install and migrate to edge platforms
- Enables on-device LLM

```
hanlab@hanlab:~/TinyChatEngine/llm$ ./chat https://github.com/mit-han-lab/TinyChatEngine_LLaMA2_7B_chat
Using model: LLaMA2_7B_chat
Using AWQ for 4bit quantization: https://github.com/mit-han-lab/llm-awq
Loading model... Finished!
USER: Introduce MIT.
ASSISTANT:
MIT is the Massachusetts Institute of Technology, a prestigious research university located in Cambridge, Massachusetts, USA. It was founded in 1865 and has since become one of the world's leading institutions for science, technology, engineering, and mathematics (STEM) education and research. MIT is known for its rigorous academic programs, innovative research, and entrepreneurial spirit. The university offers undergraduate and graduate degrees in a wide range of fields, including engineering, computer science, physics, chemistry, biology, and economics, among others.
```

```
(base) wweichen@Wei-Chens-MacBook-Pro llm % ./chat CodeLLaMA_7B_Instruct INT4 8
TinyChatEngine by MIT HAN Lab: https://github.com/mit-han-lab/TinyChatEngine
Using model: CodeLLaMA_7B_Instruct
Using AWQ for 4bit quantization: https://github.com/mit-han-lab/llm-awq
Loading model... Finished!
USER: Write a program to sort an integer array in C.
ASSISTANT:
Here is an example of how you could sort an integer array in C:
```
#include <stdio.h>

void swap(int *a, int *b) {
 int temp = *a;
 *a = *b;
}
```



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

```
shang [SSH: Hanlab_4090] - Visual Studio Code
sh - AWQ-Chat
(AWQ-Chat) x4% ./llama2_fp16.sh 7b
Loading checkpoint shards: 100% | 2/2 [00:05<00:00, 2.70s/it]
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT: Of course! Here are five popular attractions in the Greater Boston Area:
1. Fenway Park - Home of the Boston Red Sox baseball team, Fenway Park is a historic and beloved landmark in Boston. Visitors can take a guided tour of the stadium, learn about its history, and even sit in the famous Green Monster seats.
2. Boston Common - As the oldest public park in the country, Boston Common offers a peaceful escape from the hustle and bustle of the city. Visitors can stroll through the park, visit the famous Swan Boats, or attend one of the many events held here throughout the year.
3. Museum of Fine Arts - With over 450,000 works of art spanning 5,000 years of history, the Museum of Fine Arts is one of the largest and most comprehensive art museums in the world. Visitors can explore exhibitions featuring everything from ancient Egyptian artifacts to contemporary American art.
4. New England Aquarium - Located on the waterfront in Boston, the New England Aquarium is home to over 20,000 marine animals, including penguins, seals, and fish of all kinds. Visitors can watch the daily sea lion show, touch rays and starfish, or learn about the importance of marine conservation.
5. Harvard University - Founded in 1636, Harvard University is one of the oldest and most prestigious institutions of higher learning in the United States. Visitors can take a guided tour of the campus, visit the iconic Harvard Yard, or attend a lecture or performance at one of the university's many venues.
These are just a few of the many exciting attractions in the Greater Boston Area. Whether you're interested in sports, art, nature, or education, there's something for everyone in this vibrant and historic region.
=====
Speed of Inference

Generation Stage : 20.09 ms/token
=====
USER: |
```

```
shang [SSH: Hanlab_4090] - Visual Studio Code
sh - AWQ-Chat
(AWQ-Chat) x4% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100% | 1/1 [00:01<00:00, 1.27s/it]
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT: Of course! Here are five popular attractions in the Greater Boston Area:
1. The Freedom Trail: This 2.5-mile trail takes you through some of Boston's most historic sites, including Faneuil Hall, the Old North Church, and the USS Constitution. Along the way, you'll learn about the city's rich history and see iconic landmarks.
2. Fenway Park: Home of the Boston Red Sox, this legendary baseball stadium is a must-visit for any sports fan. Take a guided tour, check out the Green Monster (the famous wall in left field), and maybe even catch a game during baseball season.
3. Museum of Fine Arts: With over 450,000 works of art spanning 5,000 years of history, the MFA is one of the largest and most comprehensive art museums in the country. Explore European, American, and Asian art, as well as contemporary exhibitions and installations.
4. New England Aquarium: Located on the Boston waterfront, the NEAQ features a diverse array of marine life, including penguins, seals, and fish of all kinds. Don't miss the giant Pacific octopus, which has been known to steal the show.
5. Harvard University: As one of the oldest and most prestigious universities in the US, Harvard offers a wealth of cultural and educational attractions. Take a guided tour of the campus, visit the iconic Harvard Yard, and explore the collections at the Harvard Art Museums.
I hope these suggestions help! Is there anything else I can assist you with?
=====
Speed of Inference

Generation Stage : 6.02 ms/token
=====
USER: |
```

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



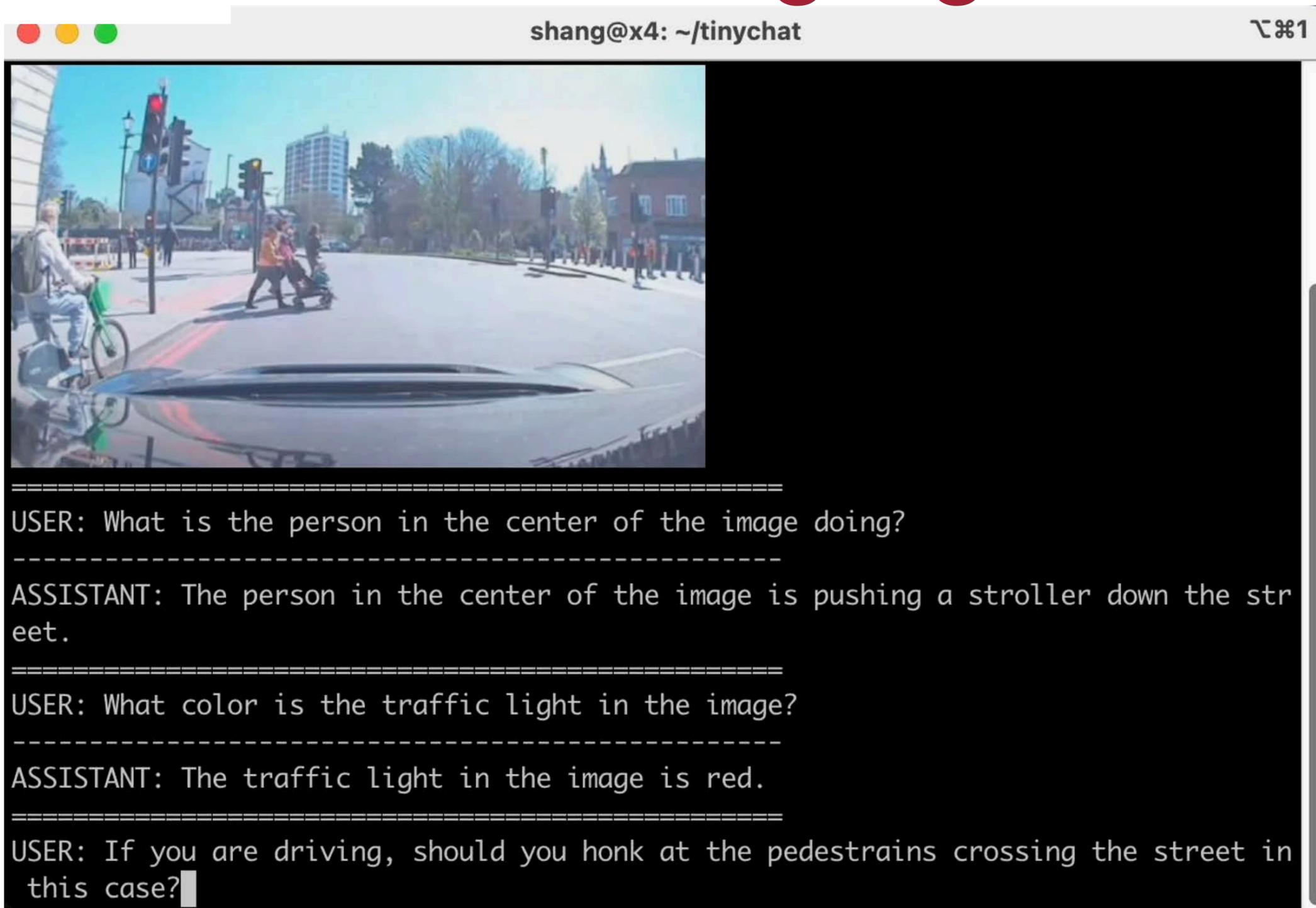
Baseline: fp16 weight, fp16 activation

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



AWQ: int4 weight, fp16 activation

# TinyChat for visual language model



Single image - Multi-round QA **VILA-13B + AWQ**: 100 tokens/s on 4090

# TinyChat for visual language model

```
shang@x4: ~/tinychat 1
--vis-image
real weight quantization...(init only): 100%|██████████| 40/40 [00:01<00:00, 22.43it/s]
Loading checkpoint: 100%|██████████| 1/1 [00:04<00:00, 4.38s/it]
=====
Input Image:

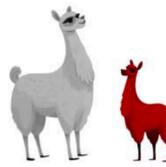
=====
USER: Photo1, at 10:30am: <image> Photo2, at 12:45pm: <image> Photo3, at 3:45pm <image> What did I have for lunch, and what time was it?

ASSISTANT: For lunch, I had a sandwich, which I enjoyed around noon.
=====
USER: What was the exact time?

ASSISTANT: The exact time of my lunch was 12:45 pm.
=====
USER: █
```

Multi-image - Multi-round QA

**VILA-13B + AWQ:** 83 tokens/s (3 image inputs) on 4090



# TinyChat for visual language model

```
shang@x4: ~/tinychat
real weight quantization...(init only): 100%|██████████| 40/40 [00:02<00:00, 19.09it/s]
Loading checkpoint: 100%|██████████| 1/1 [00:04<00:00, 4.31s/it]
=====
Input Image:



=====
Input: <image> The company is famous for its search engine. <image> The company is
famous for iPhone and Mac. <image>

Generated: The company is famous for its graphics processing units.
=====
Input:

EXIT...

Speed of Generation : 11.87 ms/token

```

**VILA-13B + AWQ: 84 tokens / s on 4090**

# TinyChat On Edge Device (Orin)



orin@orin0: ~/tinychat ⌘ 81



=====

USER: How many cars are jacked up?

-----

ASSISTANT: There are two cars jacked up in the image.

=====

USER: There is a person whose head is under the jacked up car. What color are his gloves?

-----

ASSISTANT: The person has orange gloves on.

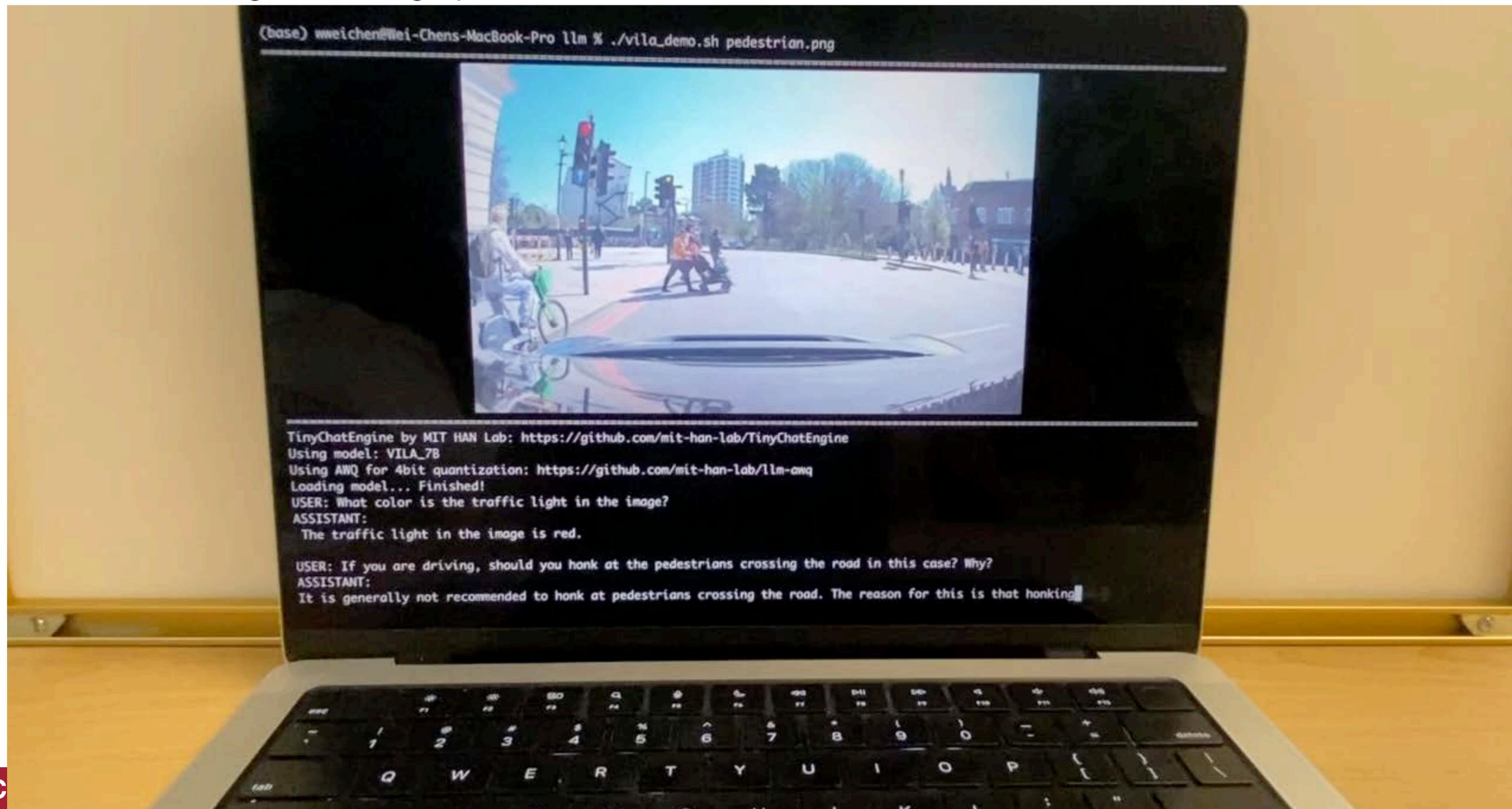
=====

USER: █

Single image - Multi-round QA **VILA-7B + AWQ: 29 tokens / s**

# TinyChat

- TinyChatEngine implements the compressed AWQ 4bit VILA model, built from C/C++ from scratch, easy to install and migrate to edge platforms, enables on-device VLM



# Impact of SmoothQuant and AWQ



TensorRT-LLM

<https://github.com/NVIDIA/TensorRT-LLM#key-features>



Neural Compressor  
Q8-Chat

[https://github.com/intel/neural-compressor/blob/master/docs/source/smooth\\_quant.md](https://github.com/intel/neural-compressor/blob/master/docs/source/smooth_quant.md)



[https://github.com/vllm-project/vllm/blob/main/vllm/model\\_executor/layers/quantization/awq.py](https://github.com/vllm-project/vllm/blob/main/vllm/model_executor/layers/quantization/awq.py)



Im-sys/FastChat

<https://github.com/lm-sys/FastChat/blob/main/docs/awq.md>



Granite

IBM's internal code model, Granite, utilizes AWQ for quantization.



text-generation-inference

[https://github.com/huggingface/text-generation-inference/tree/main/server/text\\_generation\\_server/utils/awq/quantize](https://github.com/huggingface/text-generation-inference/tree/main/server/text_generation_server/utils/awq/quantize)



Imdeploy

<https://github.com/InternLM/lmdeploy/blob/main/lmdeploy/lite/quantization/awq.py>

FriendliAI

<https://friendli.ai/blog/Unlocking-Efficiency-of-Serving-LLMs-with-Activation-aware-Weight-Quantization-AWQ-on-PeriFlow/>

Replicate

[https://github.com/replicate/vllm-with-loras/blob/main/vllm/model\\_executor/quantization\\_utils/awq.py](https://github.com/replicate/vllm-with-loras/blob/main/vllm/model_executor/quantization_utils/awq.py)

# StreamingLLM

[StreamingLLM, ICLR'24]

## Enable long conversations in non-stop streaming applications

- There's need for LLMs in streaming applications such as multi-round dialogues, where long, non-stop interactions are needed.
- Challenge: extensive memory consumption when conversation gets long; perplexity explodes after sequence length exceeds threshold.

### w/o StreamingLLM

```
(streaming) guangxuan@l29:~/workspace/streaming-llm$ CUDA_VISIBLE_DEVICE
S=0 python examples/run_streaming_llama.py
Loading model from lmsys/vicuna-13b-v1.3 ...
Loading checkpoint shards: 67%|██████████| 2/3 [00:09<00:04, 4.94s/it]
```

Without StreamingLLM, model collapses as the conversation gets long, then it runs out of memory.

### w/ StreamingLLM

```
(streaming) guangxuan@l29:~/workspace/streaming-llm$ CUDA_VISIBLE_DEVICES=1 py
thon examples/run_streaming_llama.py --enable_streaming
Loading model from lmsys/vicuna-13b-v1.3 ...
Loading checkpoint shards: 67%|██████████| 2/3 [00:09<00:04, 4.89s/it]
```

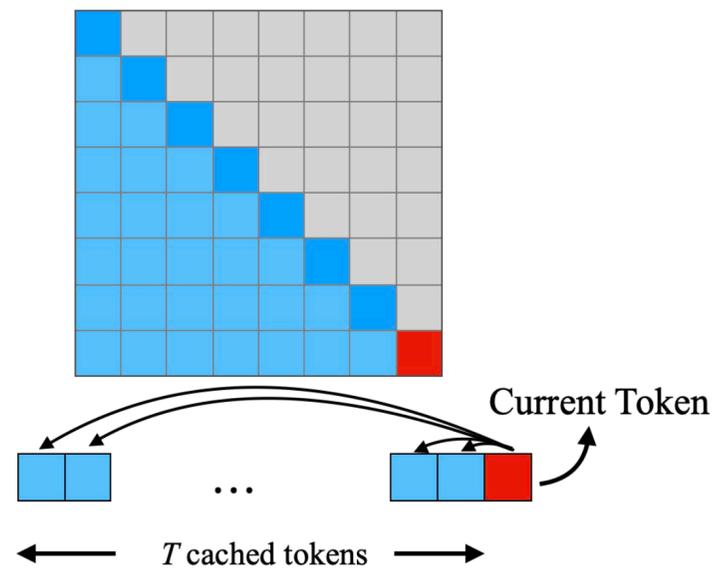
With StreamingLLM, non-stop conversation continues.

# StreamingLLM

[StreamingLLM, ICLR'24]

## Enable long conversations in non-stop streaming applications

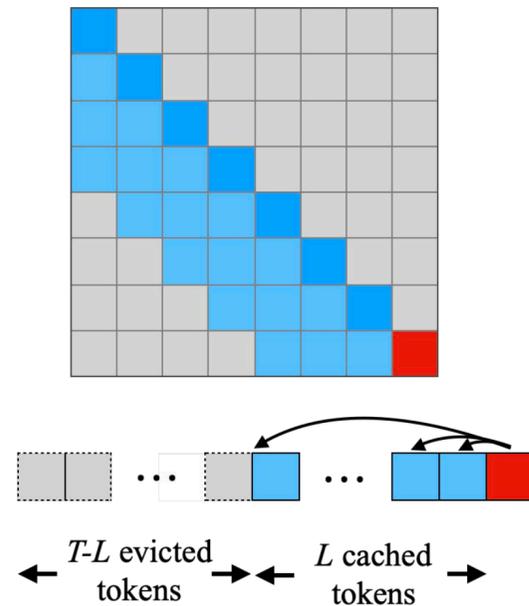
(a) Dense Attention



$O(T)$  ✗ PPL: 5641 ✗

KV cache size grows linearly with the sequence length; perplexity explodes after exceeding the max context length (4K for llama2-7B).

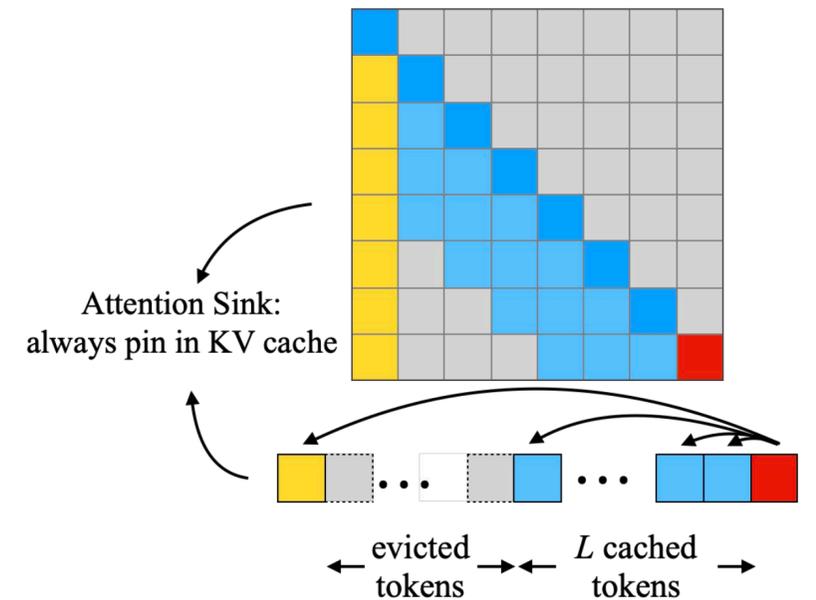
(b) Window Attention



$O(1)$  ✓ PPL: 5158 ✗

KV cache size is constant; but perplexity explodes after sequence length exceeds the KV cache size (when first token is evicted).

(c) StreamingLLM (ours)



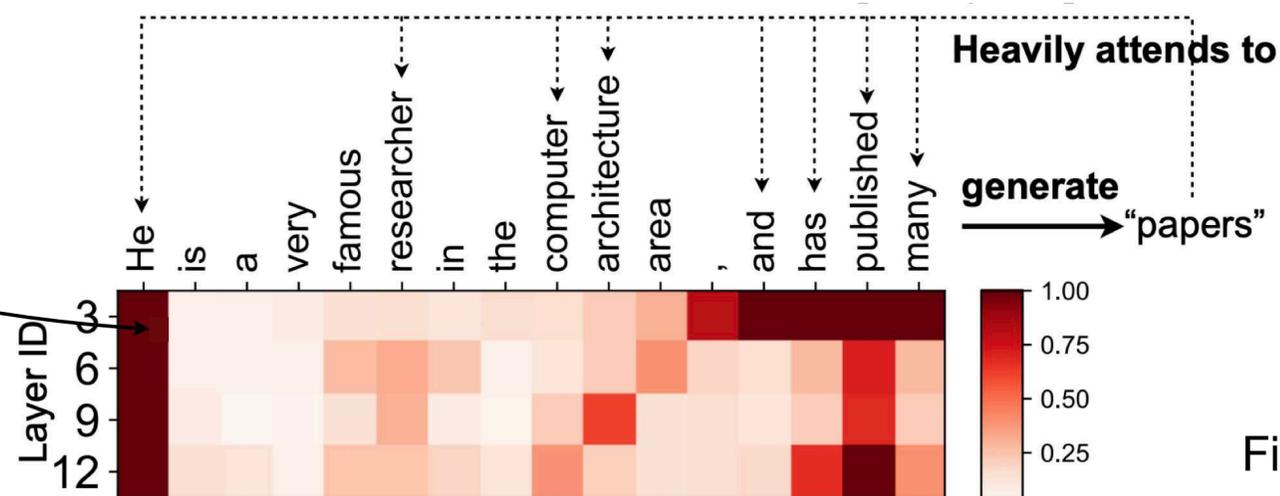
$O(1)$  ✓ PPL: 5.40 ✓

KV cache size is constant; perplexity doesn't explode.

# StreamingLLM with “Attention Sink”

[StreamingLLM, ICLR'24]

The first token is always heavily attended, we call it “attention sink”



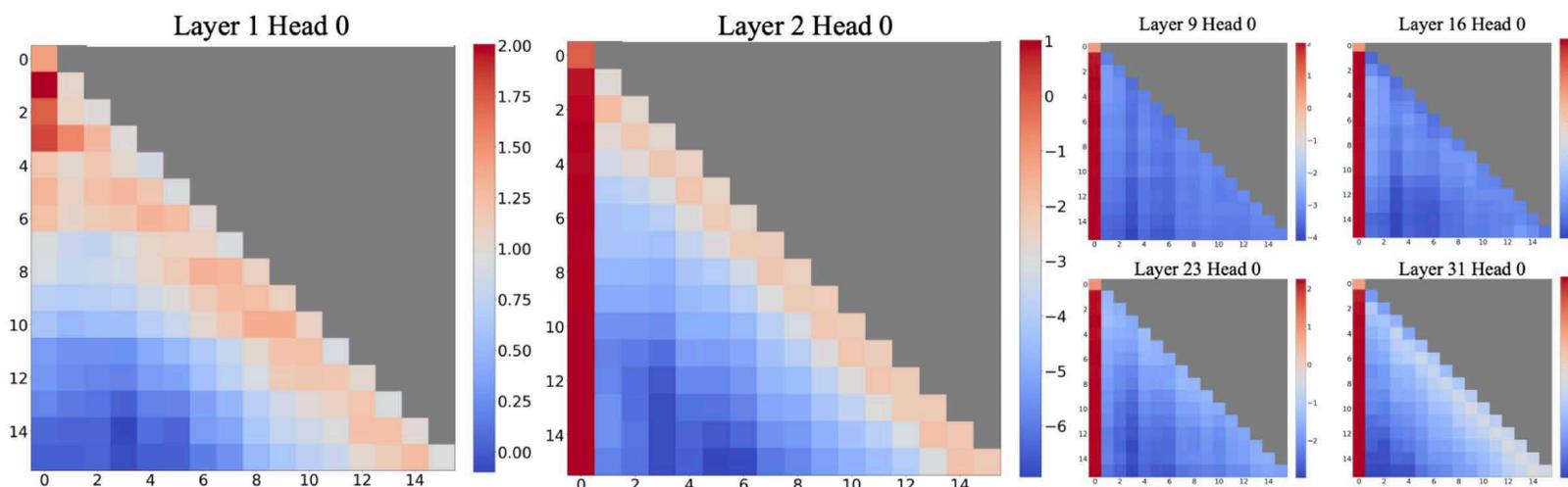
GPT-2 for Language Modeling

Du Fu was a great poet of the Tang dynasty. Recently a variety of styles have been used in efforts to translate the work of Du Fu into *English* *English* is the generated token.

~~Du Fu was a great poet of the Tang dynasty. Recently a variety of styles have been used in efforts to translate the work of Du Fu into English~~

~~Du Fu was a great poet of the Tang dynasty. Recently a variety of styles have been used in efforts to translate the work of Du Fu into English~~

attention sink can not be pruned.



Beyond layer 1, the model heavily attends to the initial token.

Why “attention sink” exist?

- SoftMax makes sure attention scores sum up to one for all contextual tokens, even not relevant, needs to dump it somewhere => attention sink.
- Why first token? Globally visible.

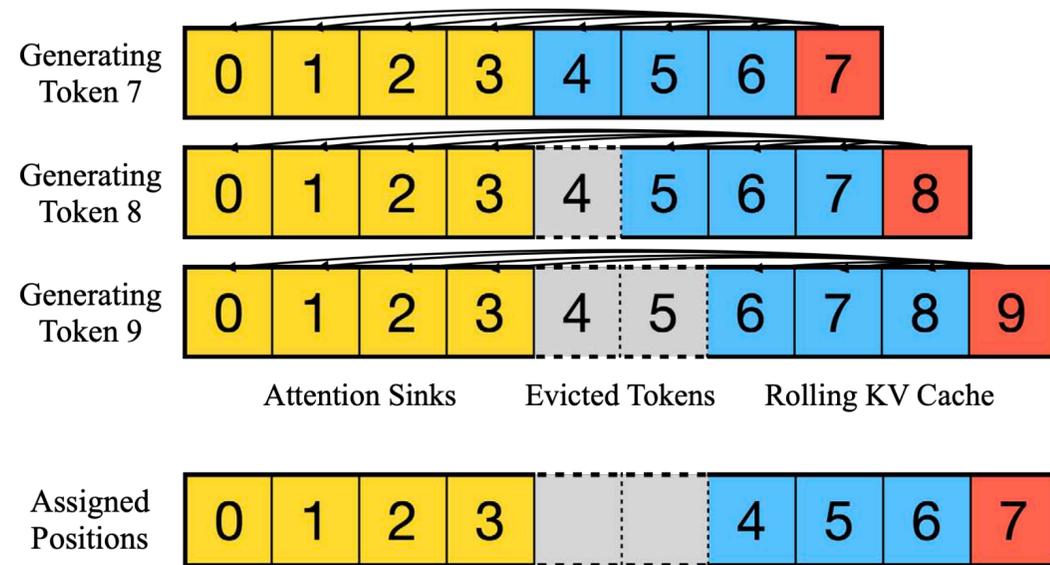
$$\text{SoftMax}(x)_i = \frac{e^{x_i}}{e^{x_1} + \sum_{j=2}^N e^{x_j}}, \quad x_1 \gg x_j, j \in 2, \dots, N$$

- What should we do with attention sink? Always keep the attention sink tokens in the KV cache.

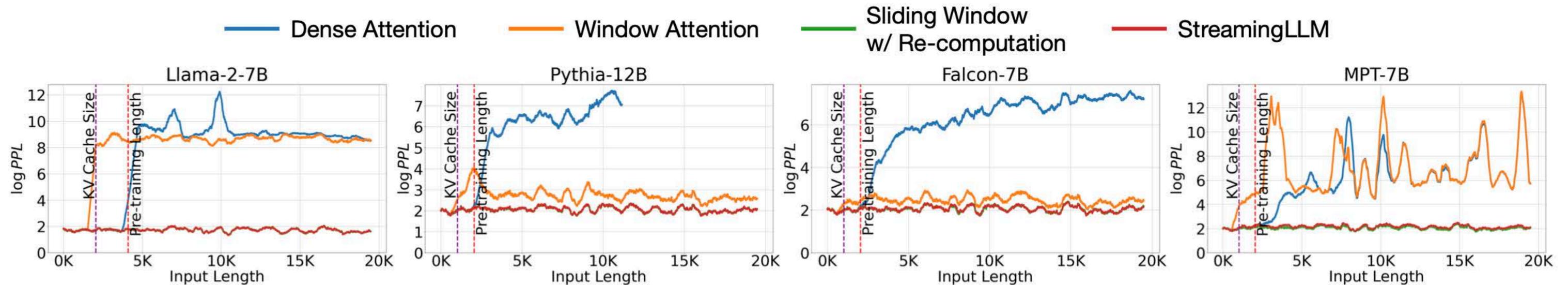
# StreamingLLM

[StreamingLLM, ICLR'24]

## Implementation and Results



- Recipe:
  - Always keep the attention sink tokens in the KV cache.
  - Use a sliding window KV to stabilize the model's behavior.
  - Use positions *in the cache* instead of those *in the original text*.
- Results:
  - Dense attention fails beyond pre-training attention window size.
  - Window attention fails after input exceeds cache size (initial tokens evicted).
  - StreamingLLM demonstrates no perplexity explosion, tested up to 4M tokens.

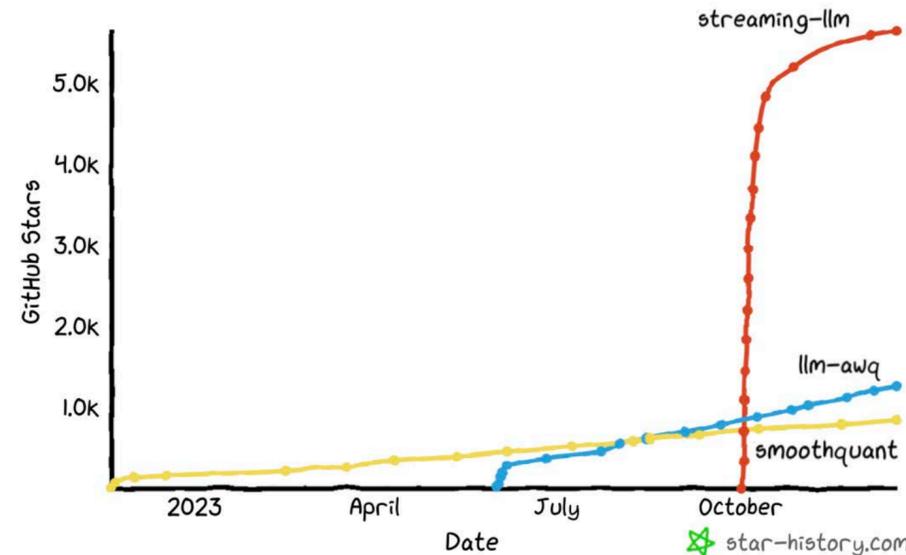
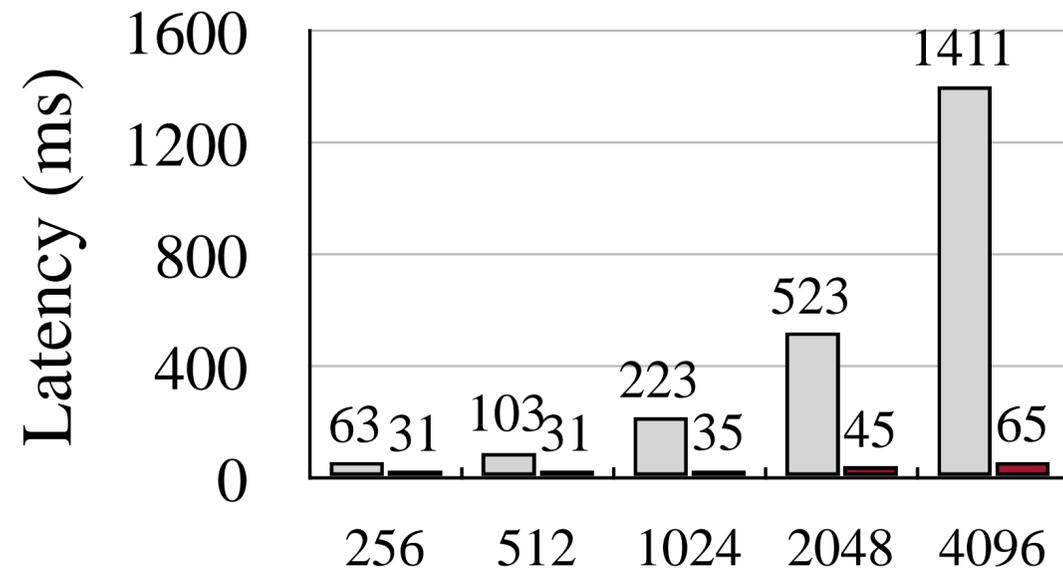


# StreamingLLM

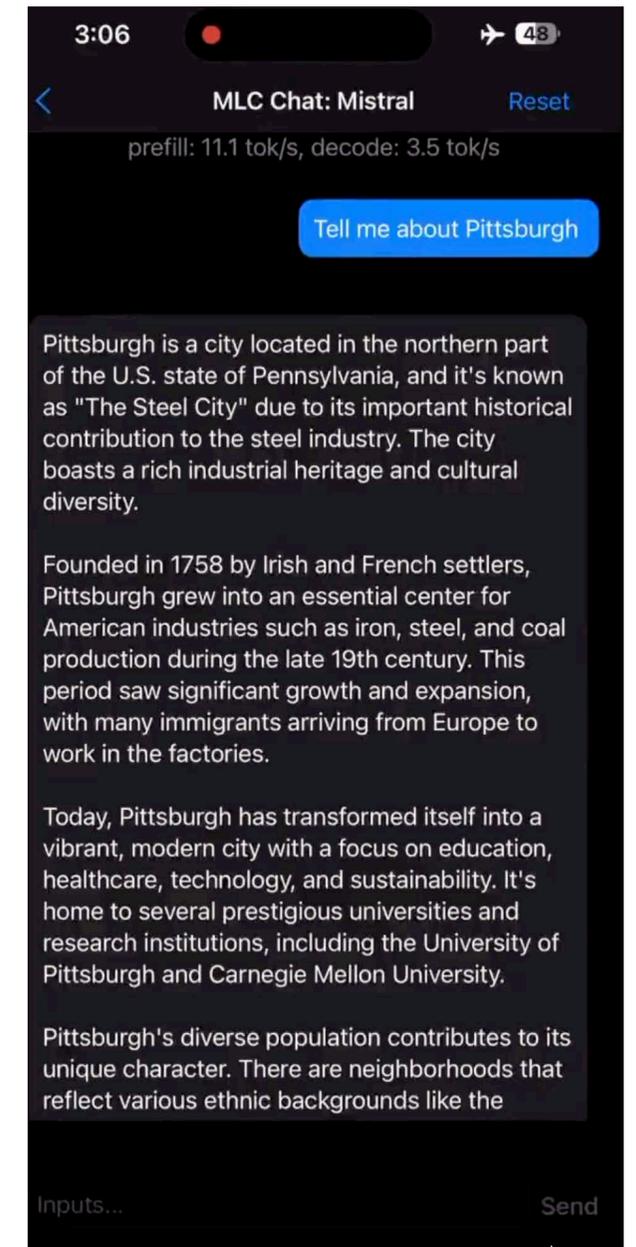
[StreamingLLM, ICLR'24]

## Enable long conversations in non-stop streaming applications

▣ Sliding Window with Re-computation    ▣ StreamingLLM



StreamingLLM on iPhone:



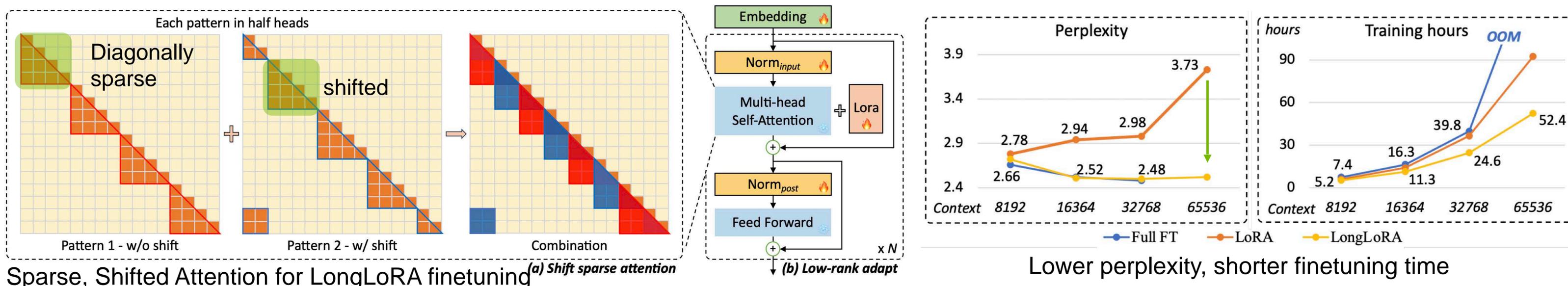
- StreamingLLM accelerates Llama2-7B compared to sliding window with recomputation.
- The more you generate, the more you save!
- StreamingLLM is available in NVIDIA TRT-LLM

# LongLoRA

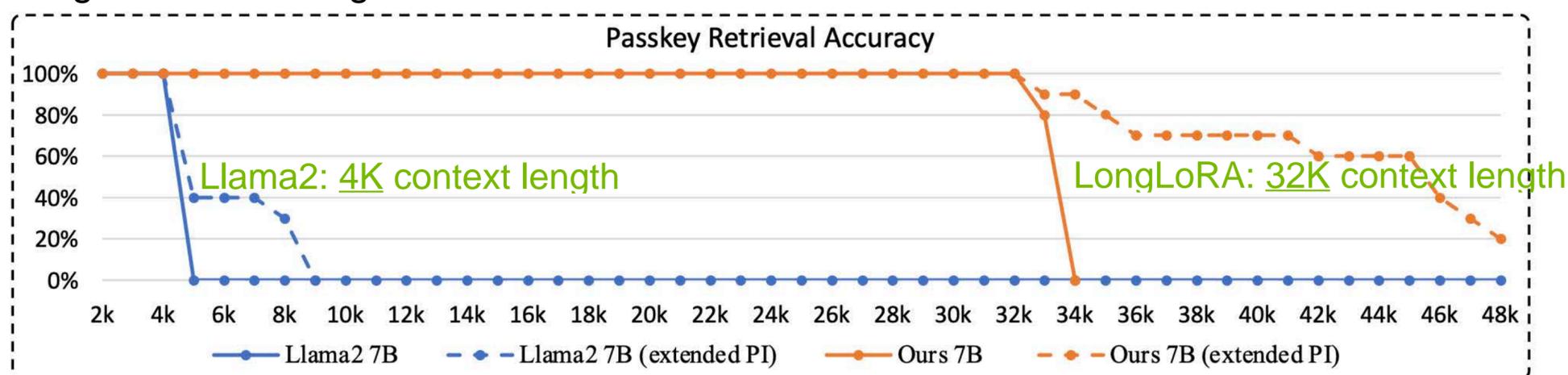
[LongLoRA, ICLR'24, Oral]

## Increase the context length with low fine-tuning cost (on a single node)

- LongLoRA invented “shifted, sparse local attention” to enable longer context length at low finetuning cost.
- LongLoRA efficiently extends the context length of Llama2-7B from 4k to 100k, Llama2-70B to 32k on a single 8× A100 machine.
- LongLoRA (for finetuning) and StreamingLLM (for inference) work together to increase both context length and generation length.

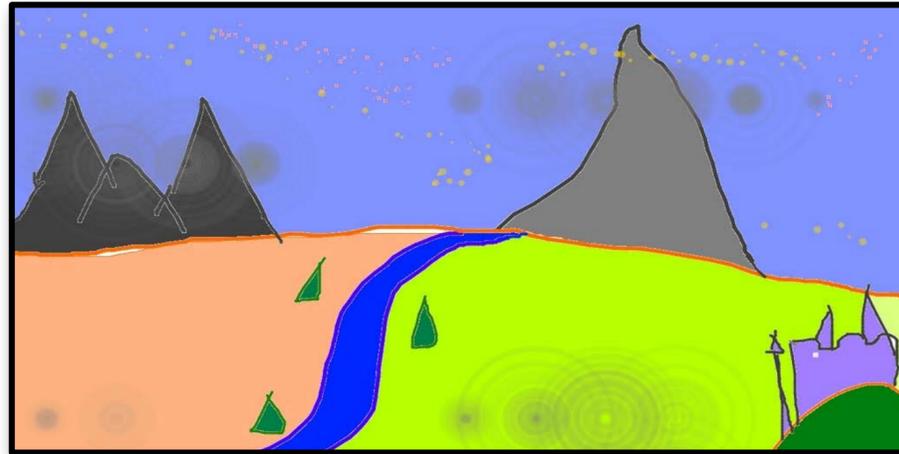


Sparse, Shifted Attention for LongLoRA finetuning

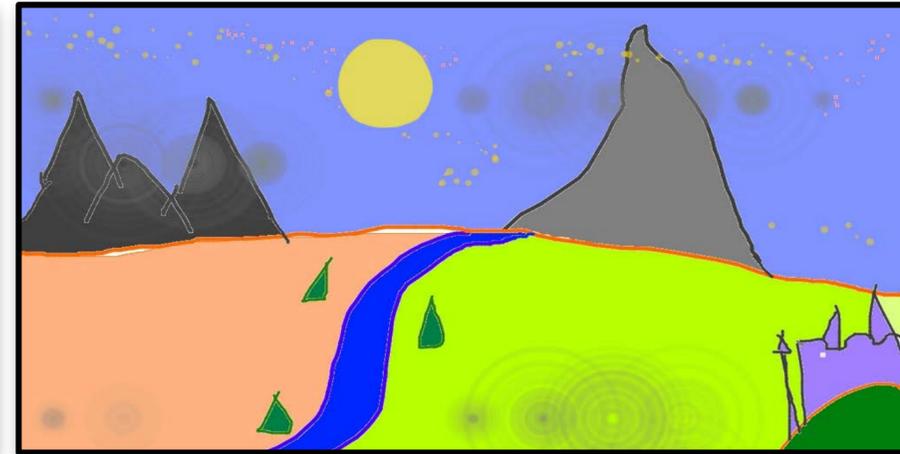


# Compressing and Accelerating Diffusion Models

*Prompt: A fantasy landscape, trending on artstation.*



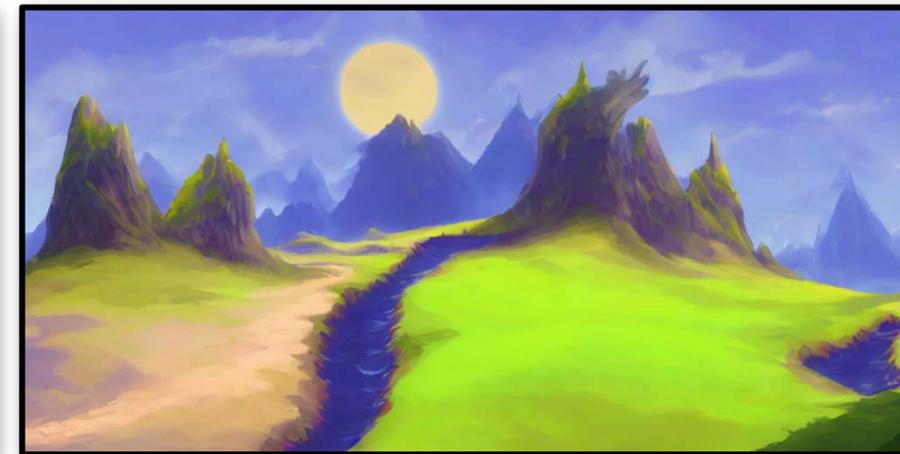
Original



Edited



Stable Diffusion+SDEdit:  
1855GMACs, 369ms



Ours:  
225GMACs (8.2x),  
51.2ms (7.2x)

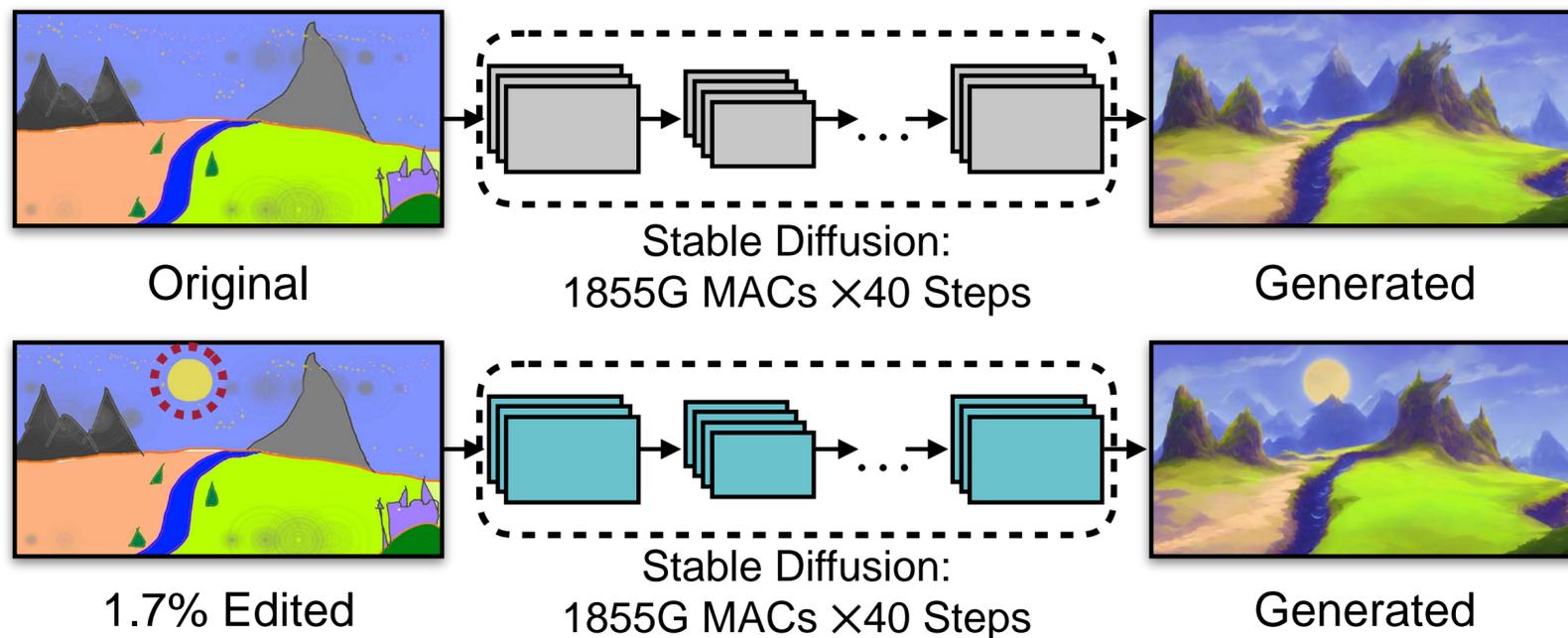
Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models. [Li et al, NeurIPS 2020]

# Spatially Sparse Inference for Diffusion Models

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

- MCUNet: System-Algorithm Co-design

Original Activations



(a) SDEdit with Vanilla Stable Diffusion

- Only **1.7%** region is edited, but vanilla model re-synthesizes the entire image.
- Feature maps remain mostly the same at unedited regions.

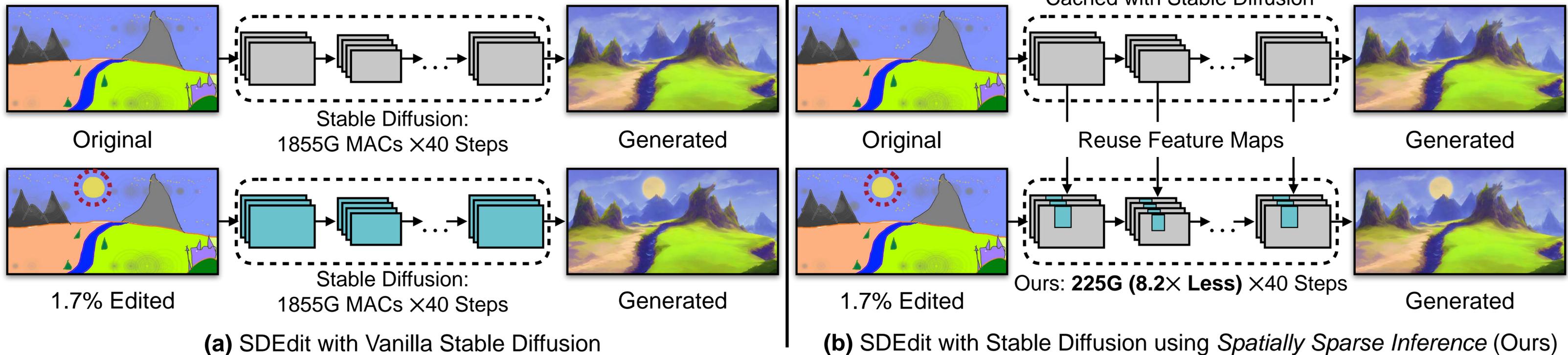
# Spatially Sparse Inference for Diffusion Models

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

- MCUNet: System-Algorithm Co-design

Original Activations

Computed On-the-fly

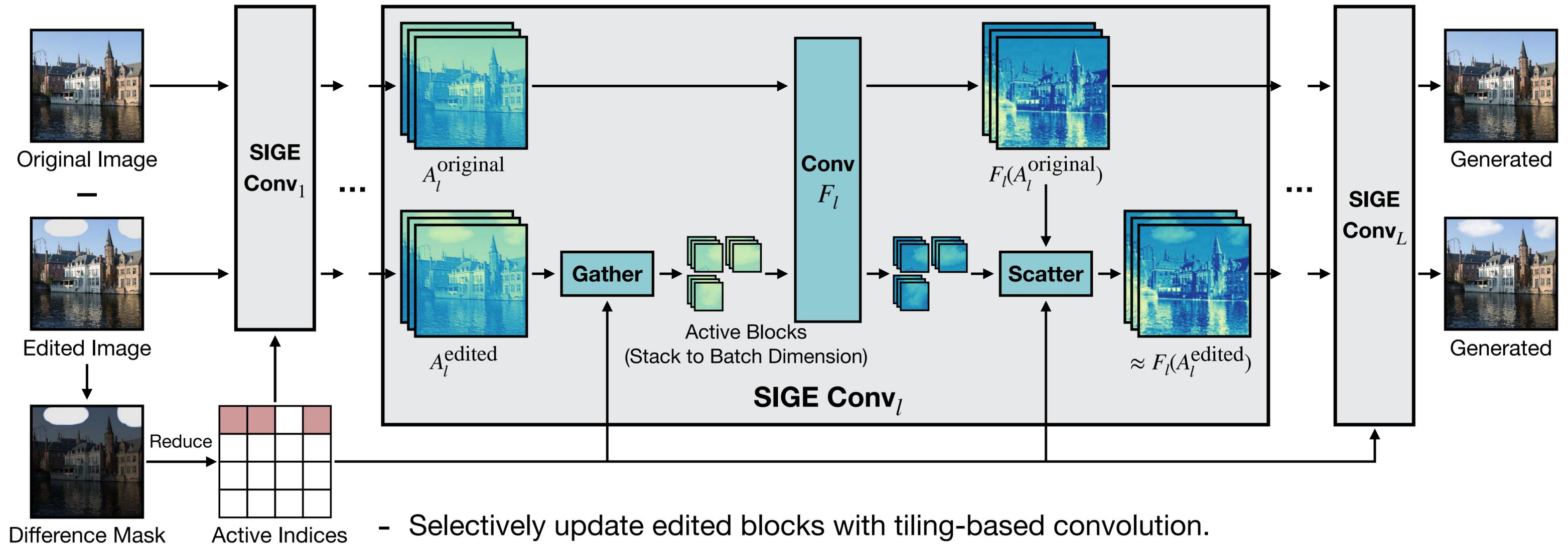


- Only **1.7%** region is edited, but vanilla model re-synthesizes the entire image.
- Feature maps remain mostly the same at unedited regions.
- Reuse cached activations to selectively update edited regions (8X less computation).

Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models. [Li et al, NeurIPS 2020]

# Spatially Sparse Inference for Diffusion Models

## Sparse Incremental Generative Engine (SIGE)



- Selectively update edited blocks with tiling-based convolution.
- Write customized Gather and Scatter operations.
- Fuse kernels to reduce overheads.

Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models. [Li et al, NeurIPS 2020]

# 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 (3.6X) 95.0ms (3.9X)

Image Inpainting

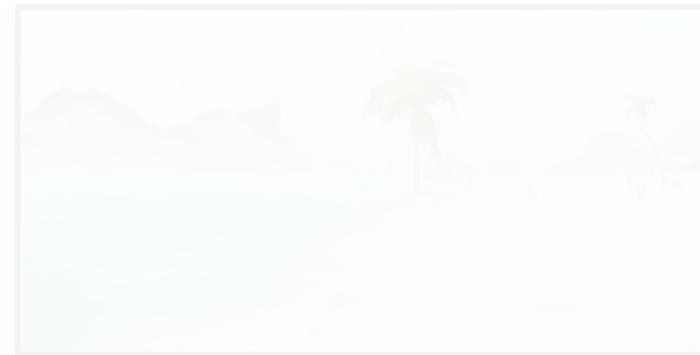
*A fantasy beach landscape, trending on artstation.*



Original



2.9% Edited



Stable Diffusion+SDEdit:  
1855GMACs 369ms



Ours:  
353G (5.3X) 76.4ms (4.8X)

Image Editing

Latency Measured on NVIDIA RTX 3090

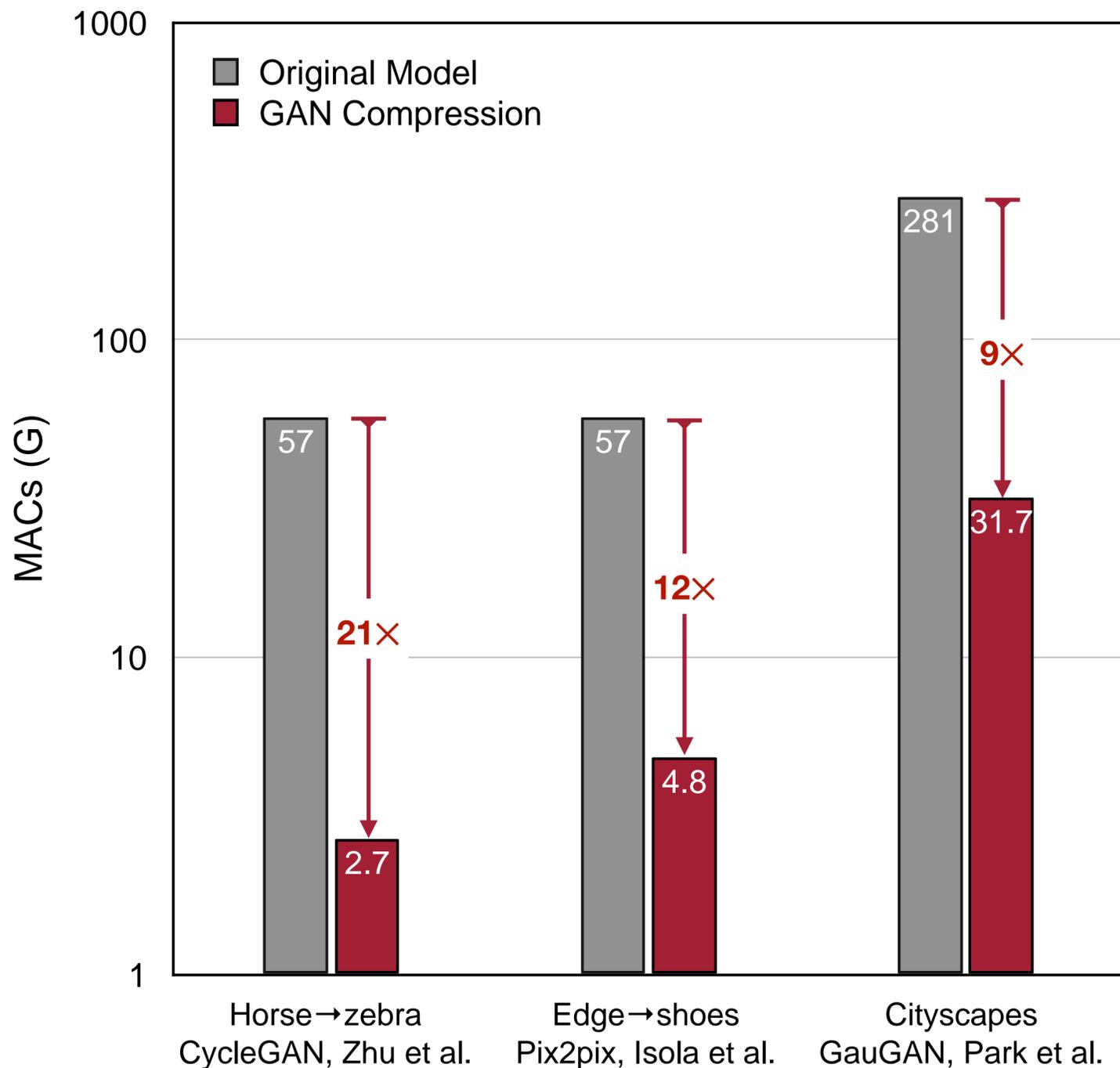


Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models. [Li et al, NeurIPS 2020]



# GAN Compression

## GAN Compression Reduces the Computation by 9-21x



Original CycleGAN; MACs: 56.8G; FPS: 12.1; FID: 61.5



GAN Compression; MACs: 3.50G (16.2x); FPS: 40.0 (3.3x); FID: 53.6

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

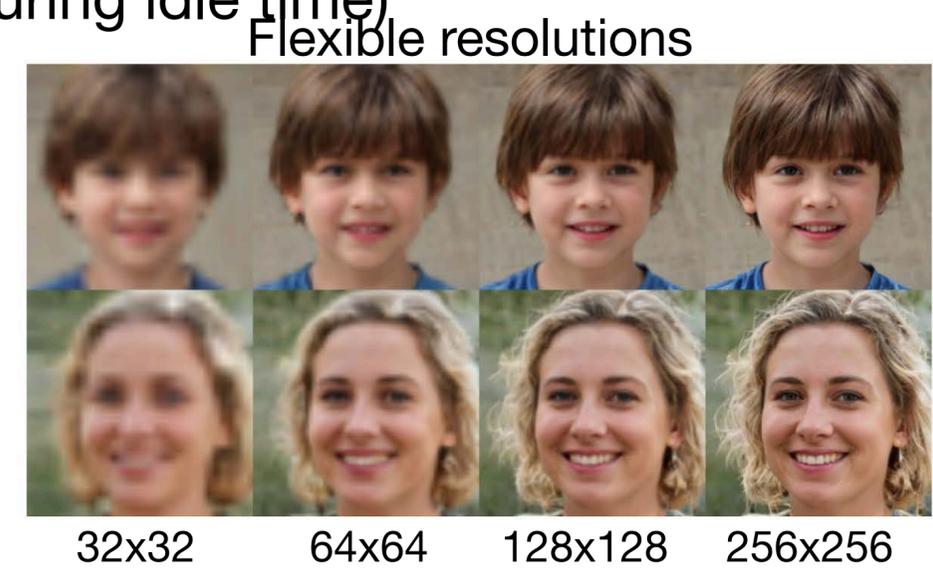
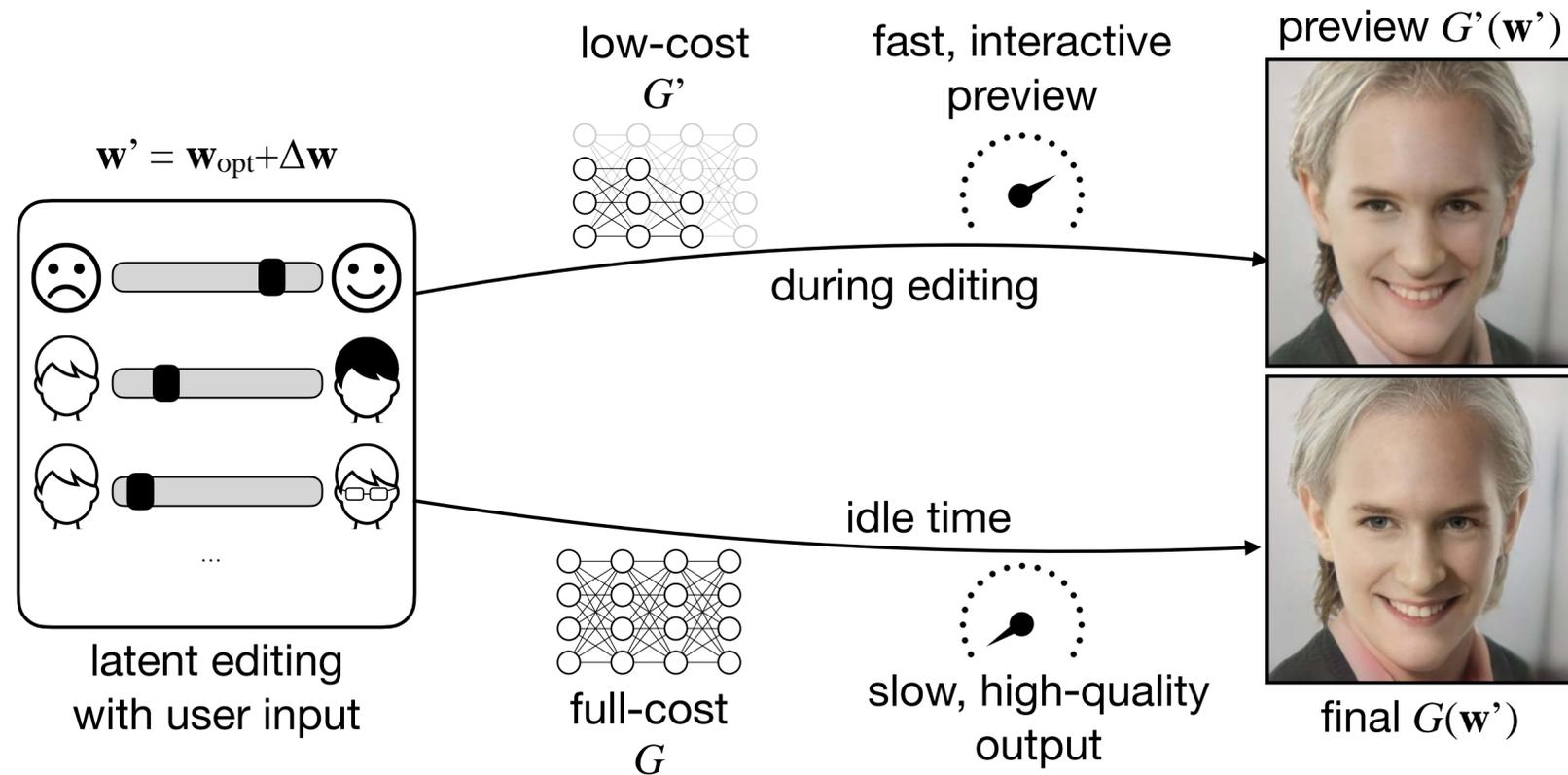


GAN Compression: Efficient Architectures for Interactive Conditional GANs. [Li et al. CVPR 2020]

# Anycost GAN

## Quick Preview By Runner A Smaller But Consistent Model

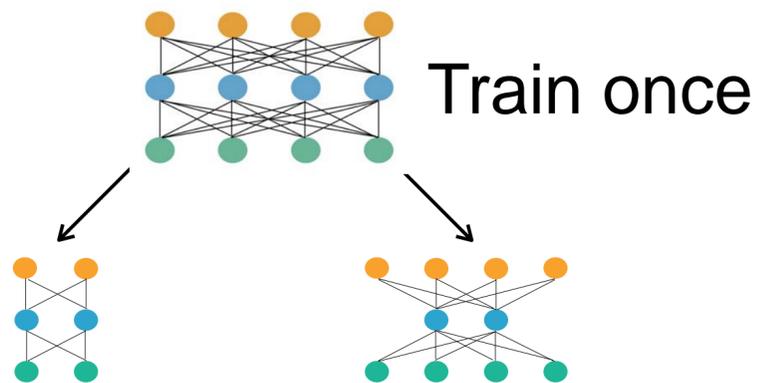
- **Anycost GAN**
  - Run a low-cost model to generate a fast, interactive preview
  - Run a full-cost model to generate the final, high-quality output (during idle time)
- Supports flexible **resolutions** and **channel widths**



Anycost GANs for Interactive Image Synthesis and Editing. [Lin et al. CVPR 2021]

# Anycost GAN

- 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:



Small sub-net:  
low cost,  
fast prototyping

Large sub-net:  
high-quality  
finalization

original                      edited

|             |  |      |
|-------------|--|------|
| smiling     |  | 1.12 |
| young       |  | 0.77 |
| big noise   |  | 0.00 |
| black hair  |  | 0.00 |
| blonde hair |  | 1.63 |
| eyeglass    |  | 0.00 |
| mustache    |  | 0.00 |

● sample1  
○ sample2  
○ sample3

○ vanilla  
● anycost

\* Status: done (0.40s)

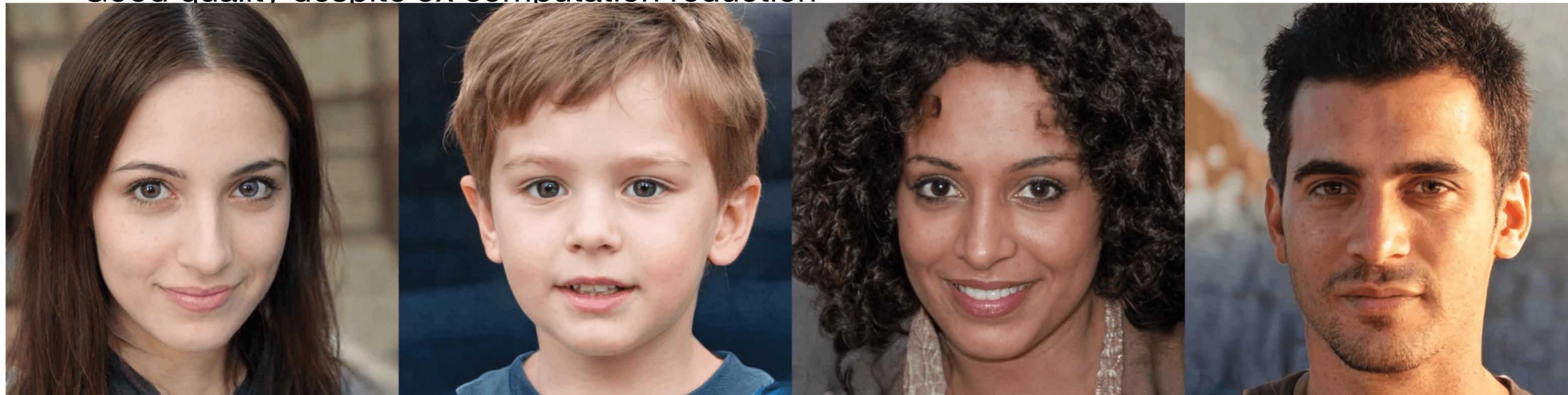
Finalize      Reset

Anycost GANs for Interactive Image Synthesis and Editing. [Lin et al. CVPR 2021]

# Anycost GAN

## Tiered Pricing for Content Generation as a Service

- Good quality despite 5x computation reduction



MACs:  100% 1.0x reduction

|                       |        |         |         |         |         |
|-----------------------|--------|---------|---------|---------|---------|
| Compute Budget        | 1x     | 0.7x    | 0.5x    | 0.4x    | 0.2x    |
| <b>Tiered Pricing</b> | \$0.01 | \$0.007 | \$0.005 | \$0.004 | \$0.002 |

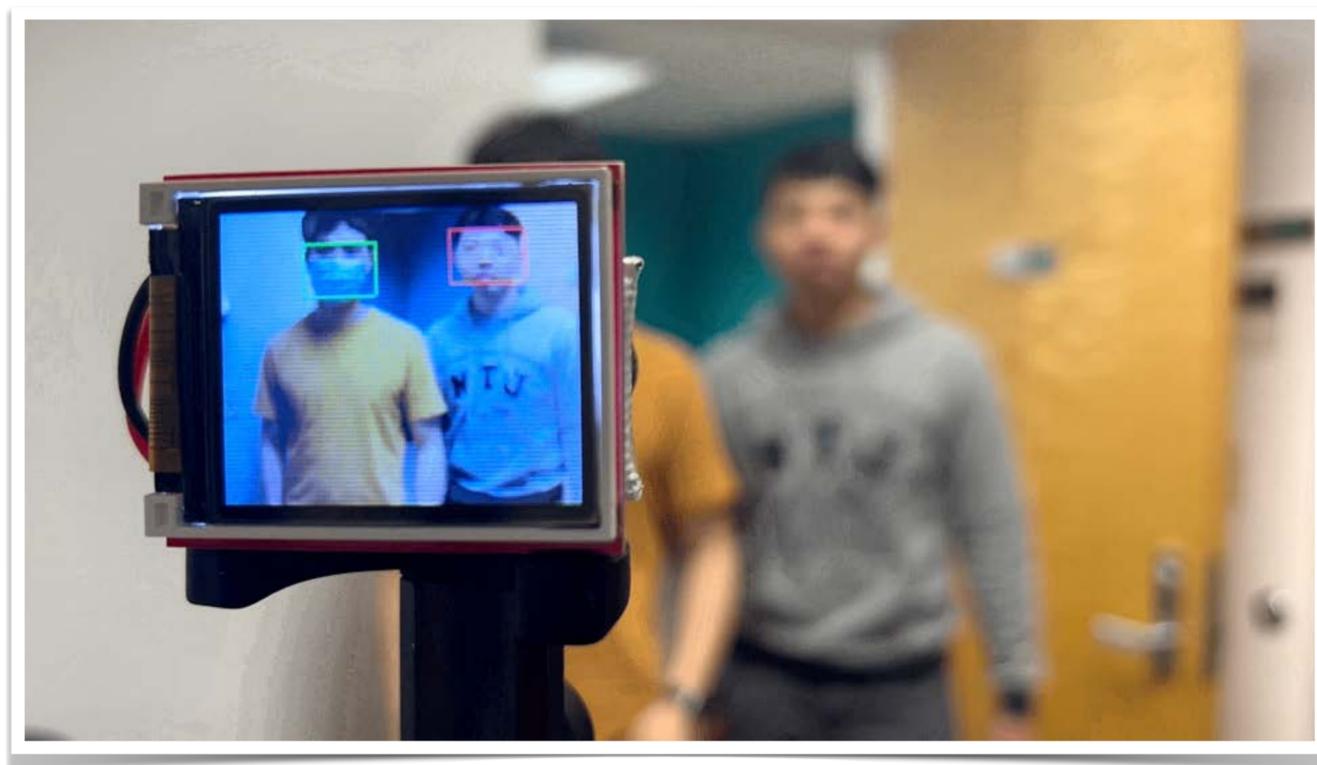
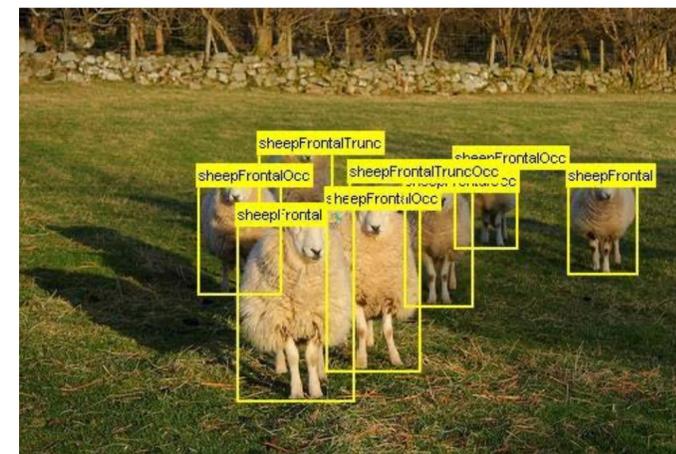
Anycost GANs for Interactive Image Synthesis and Editing. [Lin et al. CVPR 2021]

The quality is still  
reasonably good

# MCUNet for TinyML

## Advancing object detection by allowing a larger resolution

- Resolution is more important for detection than classification
- Our method significantly improves objection detection by double digits



Face/mask detection



Person detection

**OpenMV Cam: 512KB SRAM + 2MB Flash**

# MCUNet v1-v3

**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

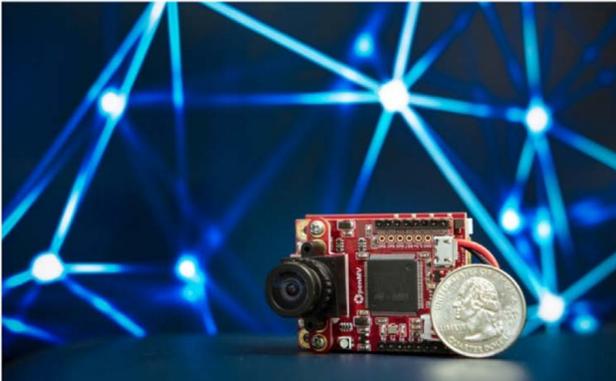
## Tiny machine learning design alleviates a bottleneck in memory usage on internet-of-things devices

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.

(Highlighted by MIT Homepage)

**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

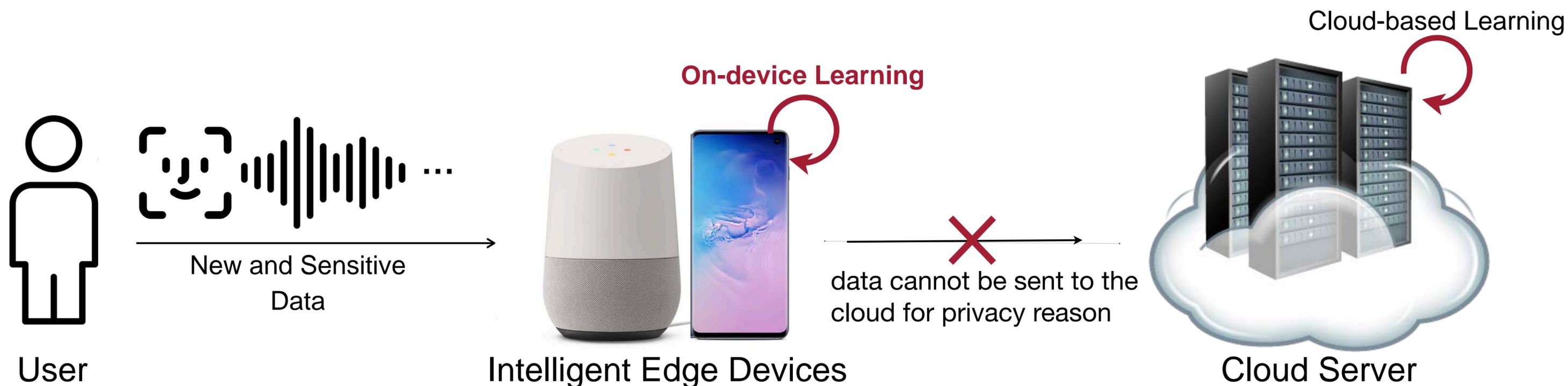


A machine-learning model on an intelligent edge device allows it to adapt to new data and make better predictions. For instance, training a model on a smart keyboard could enable the keyboard to continually learn from the user's writing.  
Image: Digital collage by Jose-Luis Olivares, MIT, using stock images and images derived from MidJourney AI.

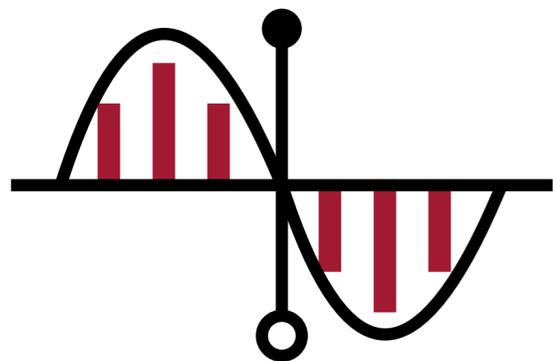
# Can We Learn on Edge?

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

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

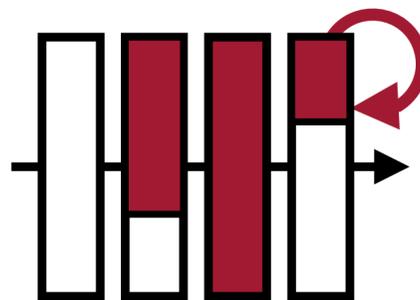


# On-Device Training Under 256KB Memory

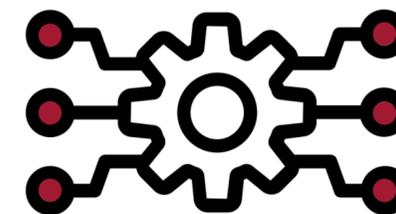


## 1. Quantization-aware scaling

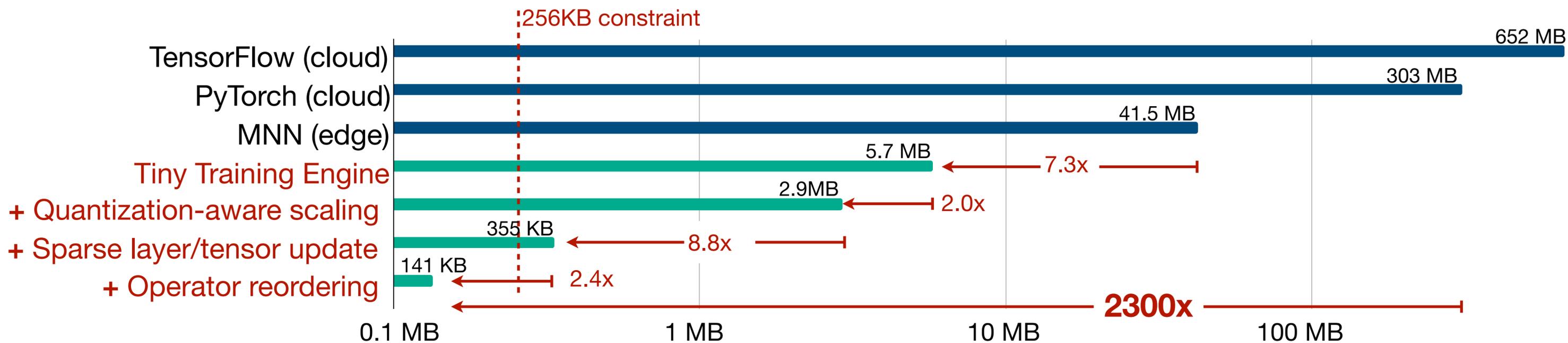
$$\tilde{G}_{\bar{w}} = G_{\bar{w}} \cdot s_{\bar{w}}^{-2}$$



## 2. Sparse layer/tensor update

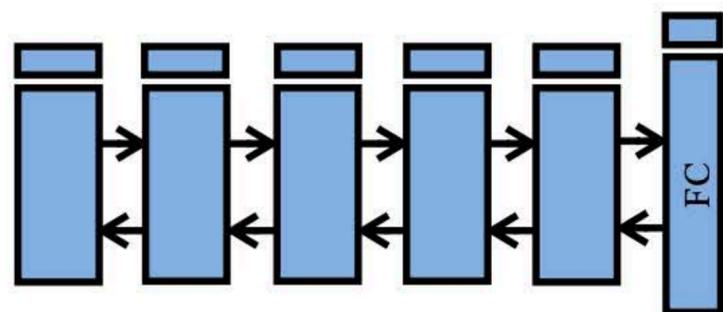


## 3. Tiny Training Engine

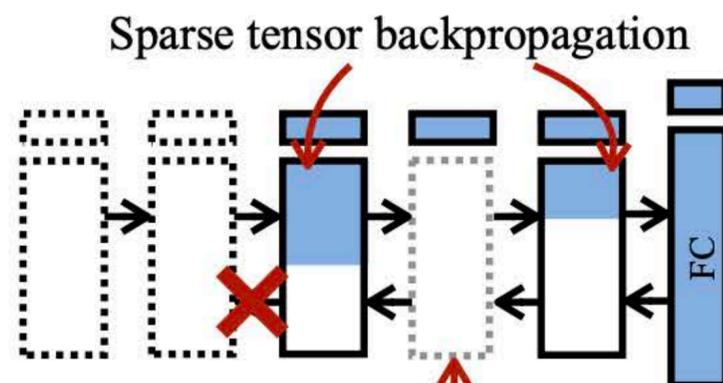


# PockEngine

## Learning on the Edge



Full back-propagation



Sparse layer backpropagation  
Sparse back-propagation

## Technique enables AI on edge devices to keep learning over time

With the PockEngine training method, machine-learning models can efficiently and continuously learn from user data on edge devices like smartphones.

Adam Zewe | MIT News

November 16, 2023



**Input:** Please reverse the words in the sentence "I love the Micro conference"

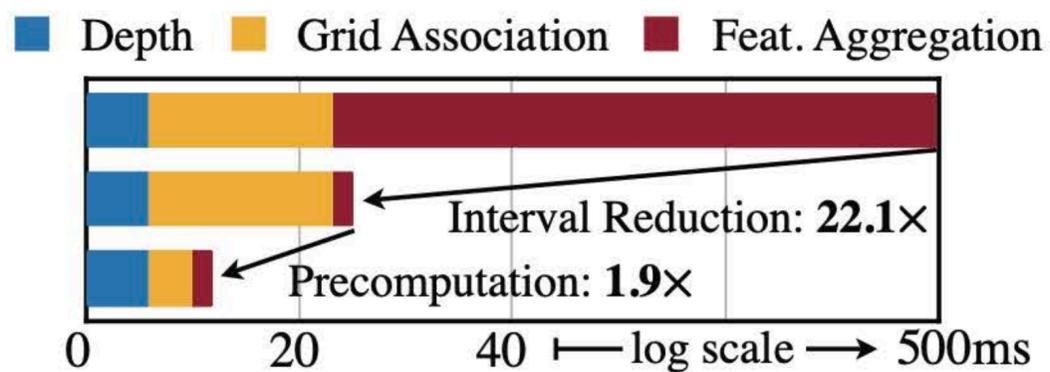
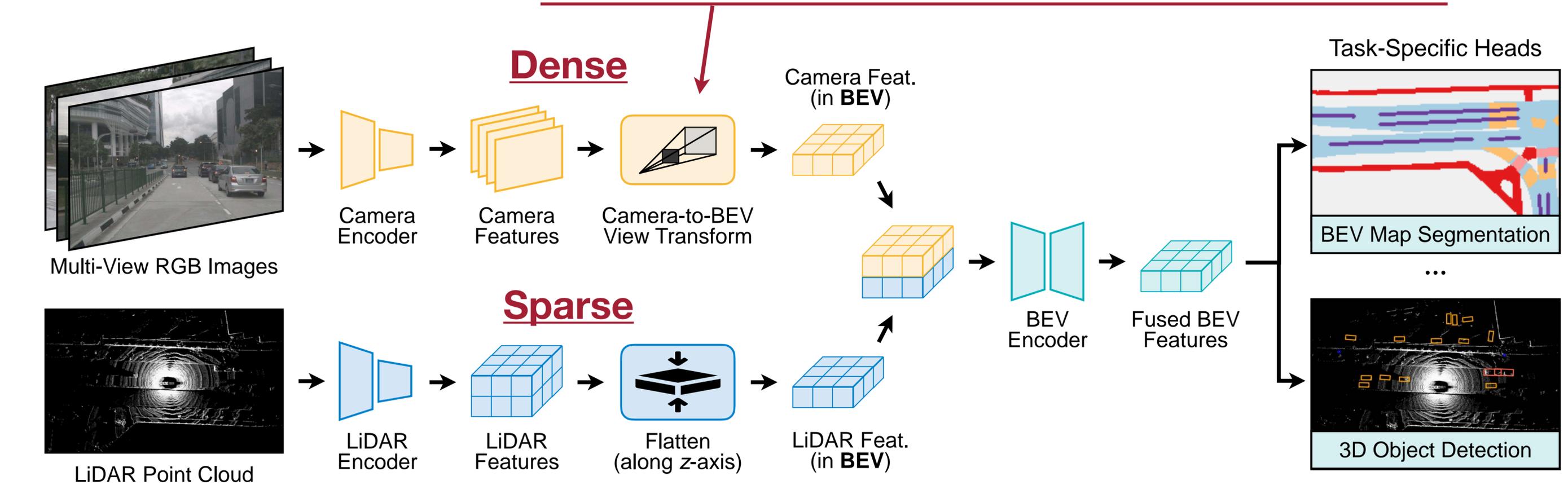
**LlamaV2-7B-original:** I hate the conference Micro.

**LlamaV2-7B-tuned:** The reversed result is "Conference Micro the love I"

| Method        | Iteration Latency (↓) |
|---------------|-----------------------|
| FT-Full       | 7.7s                  |
| LoRA (rank=8) | 7.3s                  |
| FT-Full       | 1.8s                  |
| Sparse        | 0.9s                  |

# BEVFusion: Dense (Camera) + Sparse (LiDAR) Sensor Fusion

## 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

BEVFusion takes **multi-modal** sensory inputs and supports **multiple** 3D perception tasks.

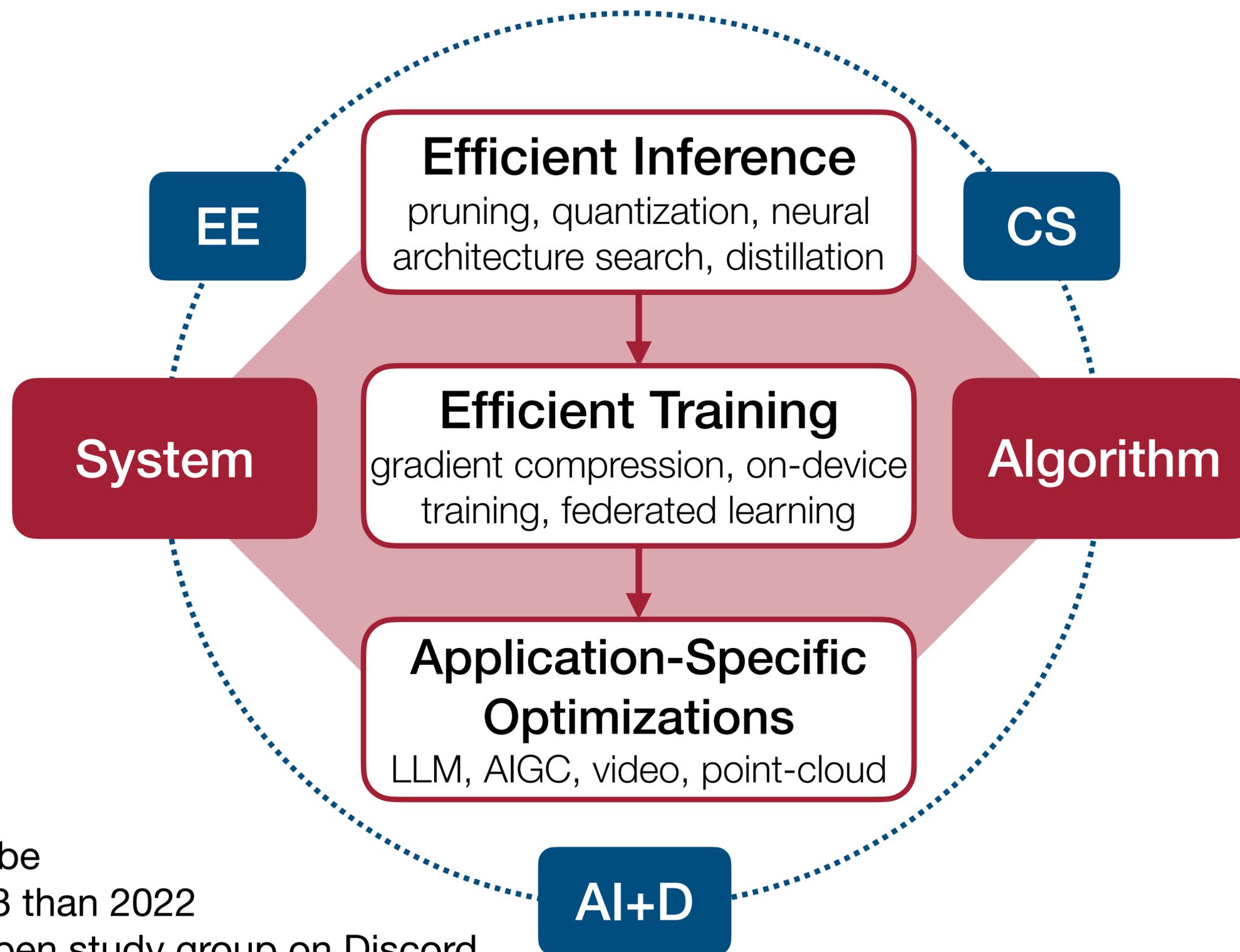
Multi-View Camera



LiDAR



<https://youtu.be/uCAka90si9E>



200K views on YouTube

3x registration in 2023 than 2022

700 students in the open study group on Discord

# MIT HAN LAB

Hardware, AI and Neural-nets



[github.com/mit-han-lab](https://github.com/mit-han-lab)



[youtube.com/c/MITHANLab](https://youtube.com/c/MITHANLab)



[songhan.mit.edu](https://songhan.mit.edu)  
[tinymml.mit.edu](https://tinymml.mit.edu)

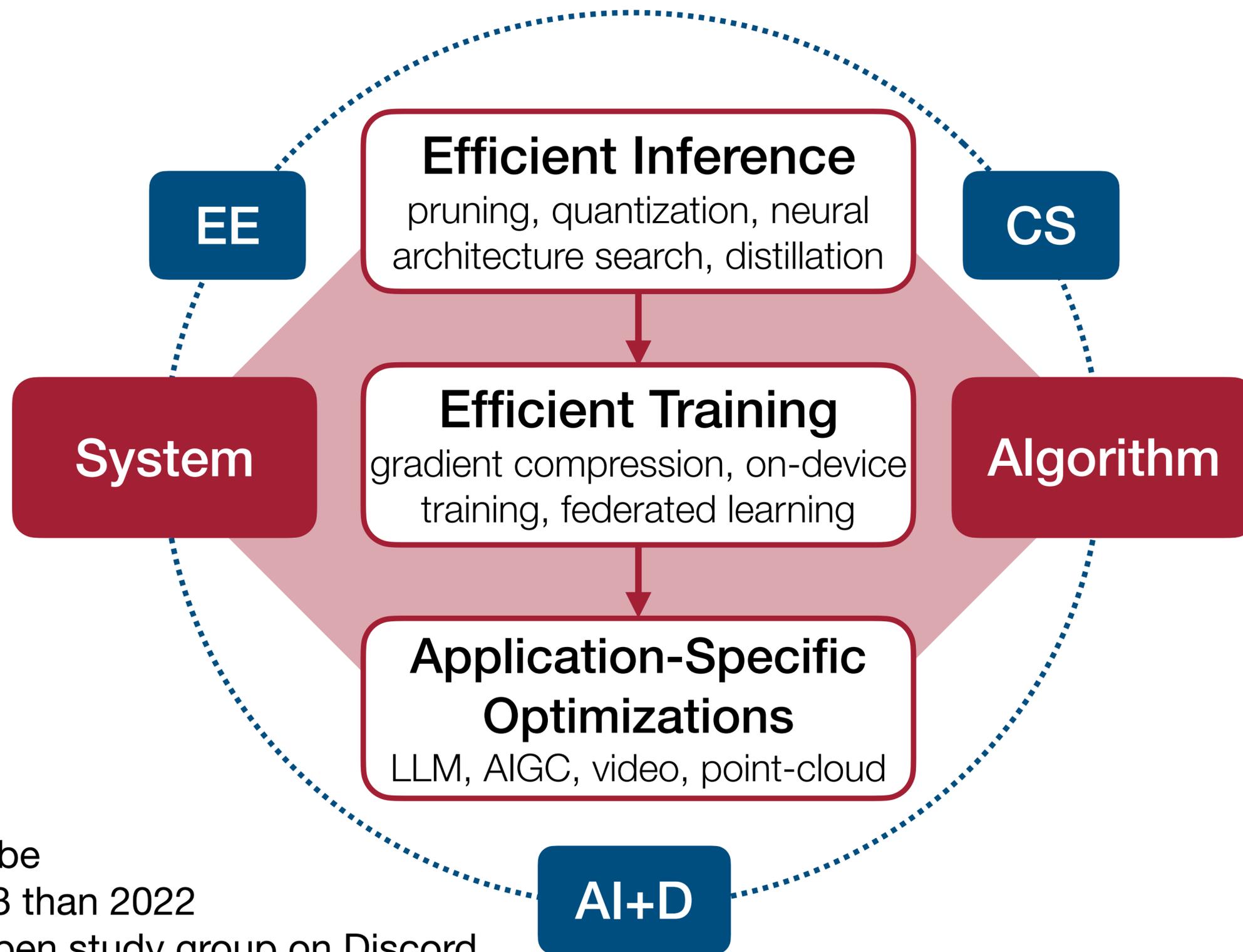


## Sponsors:



## Media:





200K views on YouTube

3x registration in 2023 than 2022

700 students in the open study group on Discord

# EfficientML.ai Course

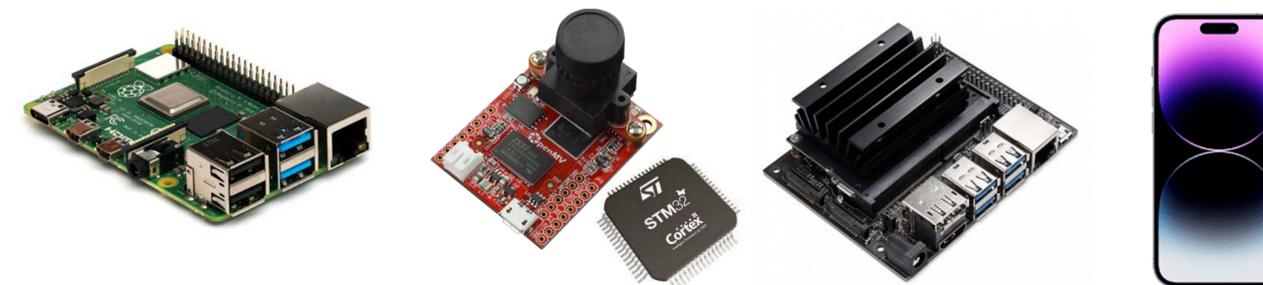
## TinyML and Efficient Deep Learning Computing

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](mailto:efficientml-staff@mit.edu).
  - If you are interested in getting updates, please sign up [here](#) to join our mailing list to get notified!



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

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

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

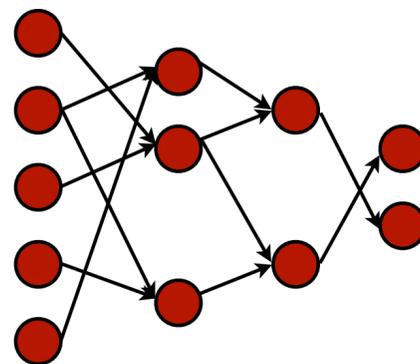
I like the class and I have been able to follow the class easily (which had rarely happened to me in my previous courses)

<https://efficientml.ai>

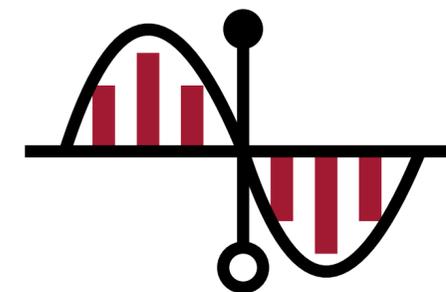
# EfficientML.ai Hands-on Labs

PYTORCH

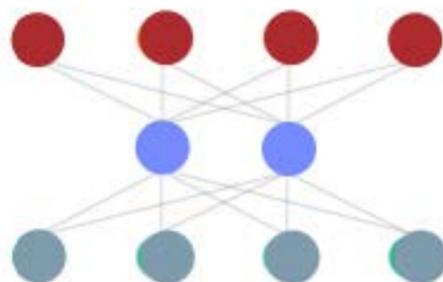
Lab 0: Tutorial  
on PyTorch



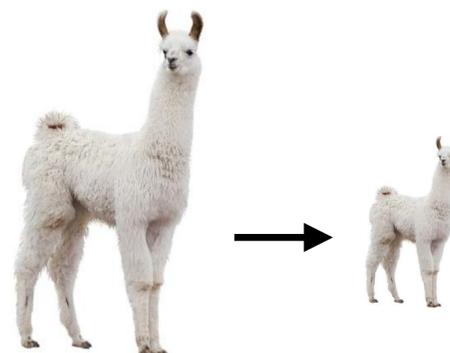
Lab 1:  
Pruning



Lab 2:  
Quantization



Lab 3:  
Neural Architecture  
Search



Lab 4:  
LLM Compression



Lab 5:  
On-Device LLM  
Deployment



# MIT AI Hardware Program

MIT Microsystems Technology Laboratories (SoE)  
MIT Quest for Intelligence – Corporate (SCC)

Co-Leads: Jesús del Alamo and Aude Oliva

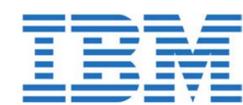
Internal Advisory Board Chair: Anantha Chandrakasan

# TinyML and Efficient AI Computing



-  [github.com/mit-han-lab](https://github.com/mit-han-lab)
-  [youtube.com/c/MITHANLab](https://youtube.com/c/MITHANLab)
-  [songhan.mit.edu](https://songhan.mit.edu)  
[tinymml.mit.edu](https://tinymml.mit.edu)

## Sponsors:



## Media:

