



# AWQ and TinyChat: Efficient LLMs on the Edge

**Ji Lin**

[jilin@mit.edu](mailto:jilin@mit.edu)  
<https://linji.me>

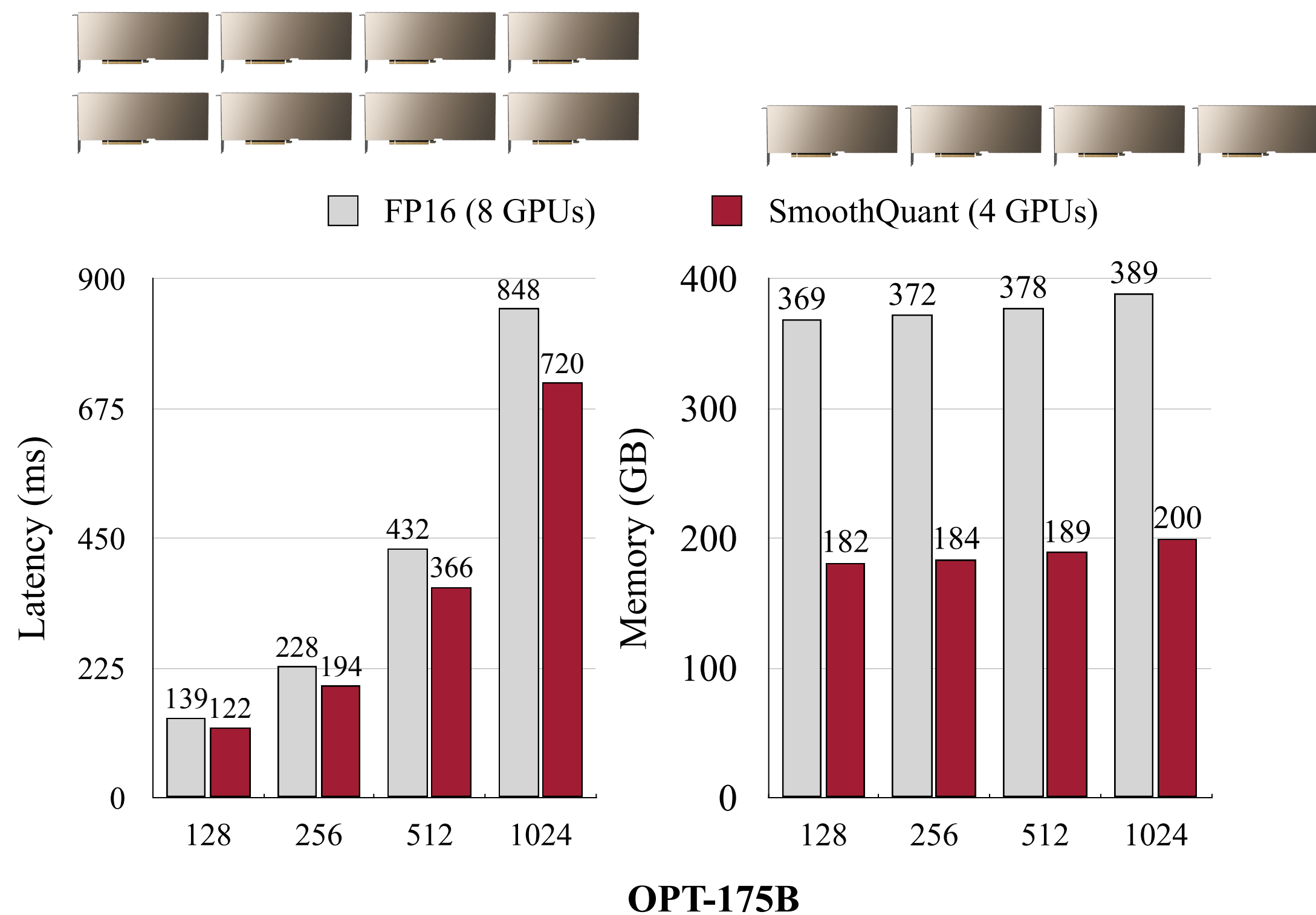
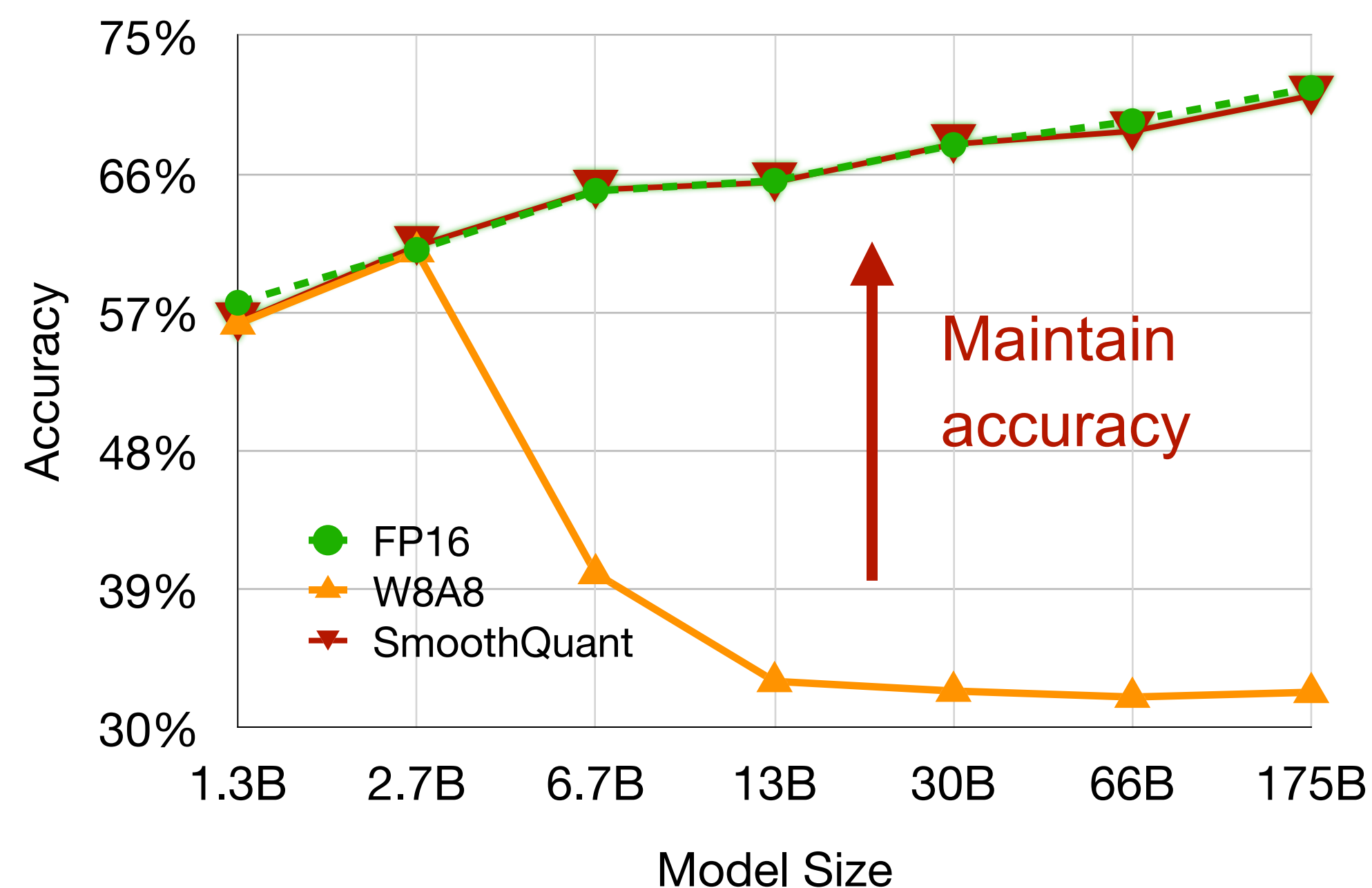
**Haotian Tang**

[kentang@mit.edu](mailto:kentang@mit.edu)  
<http://kentang.net>

# Revisit: SmoothQuant (W8A8)

## Accurate and efficient quantization of various LLMs

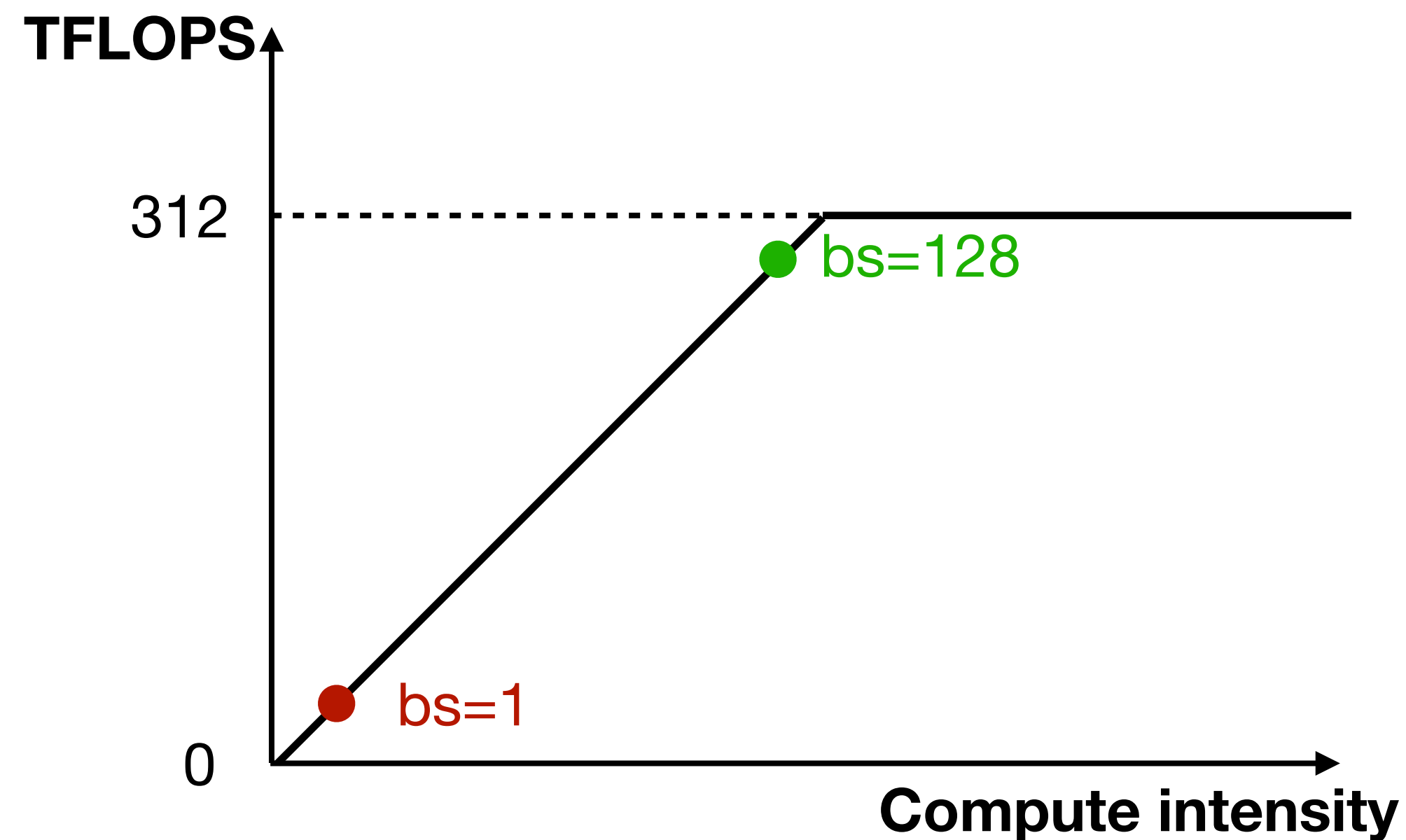
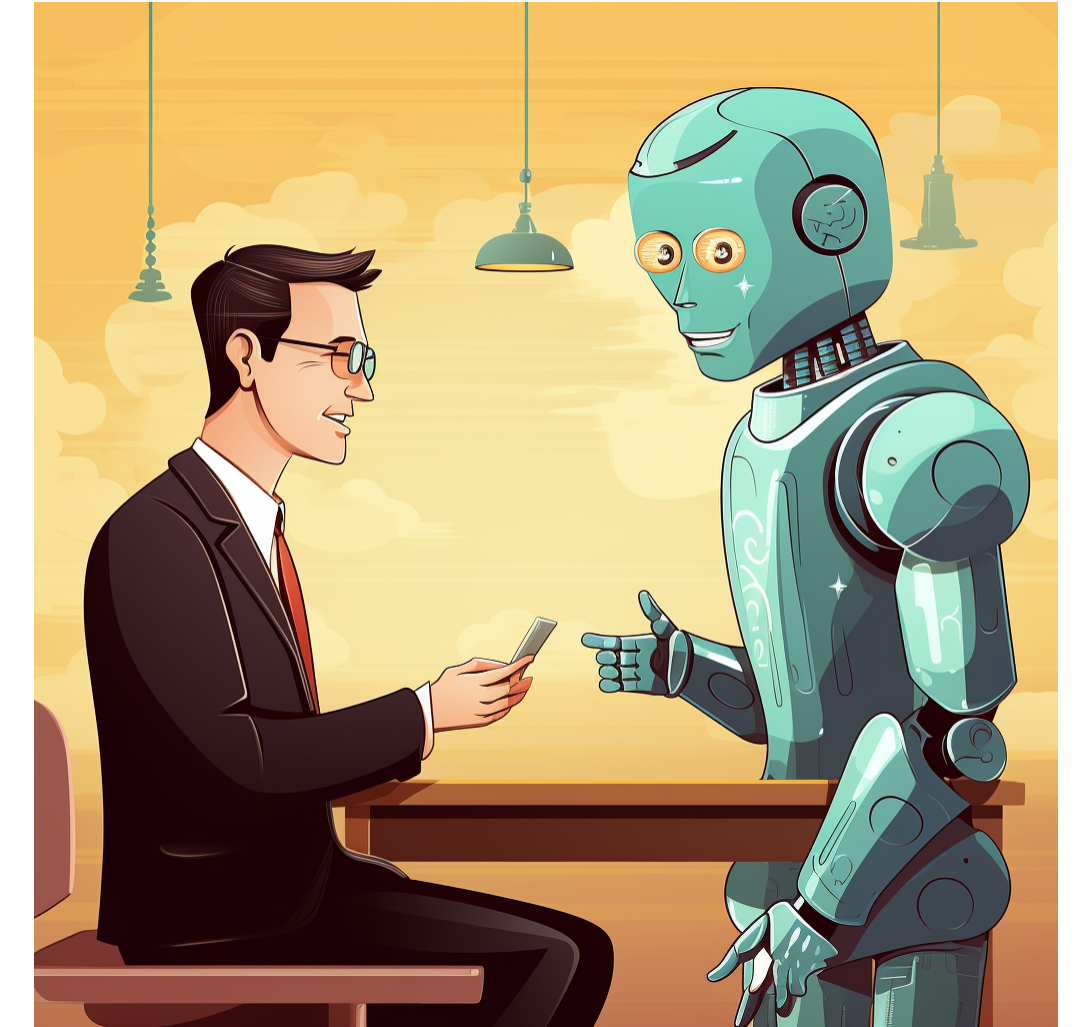
- SmoothQuant well maintains the accuracy without fine-tuning.
- SmoothQuant can both accelerate inference and halve the memory footprint.



# W4A16 for Single Query Serving

## W8A8 cannot address low computational intensity of decoding

- W8A8 quantization is good for batch serving (e.g., batch size 128)
- But single-query LLM inference (e.g., local) is still highly memory-bounded
- We need **low-bit weight-only** quantization (e.g., W4A16) for this setting

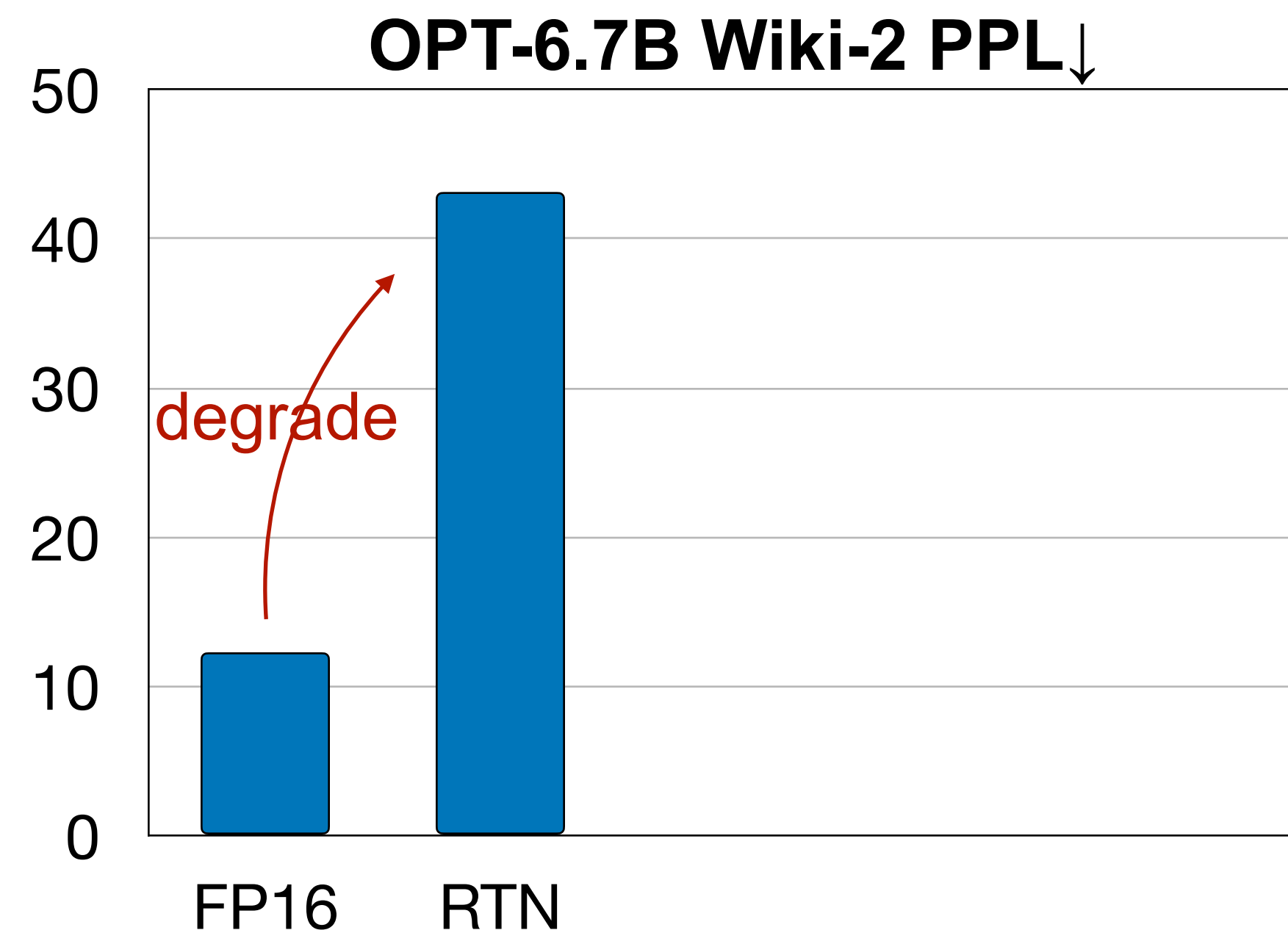


- A100 GPU
- LLaMA-65B decoding

# AWQ for Low-bit Weight-only Quantization

## Targeting group-wise W3/W4 quantization

$W_{FP16}$					$Q(W)_{INT3}$			
+1.2	-0.2	-2.4	-3.4		+1	+0	-2	-3
-2.5	-3.5	+1.9	+1.4		-3	-4	+2	+1
-0.9	+1.6	-2.5	-1.9		-1	+2	-3	-2
-3.5	+1.5	+0.5	-0.1	RTN	-4	+2	+1	+0
+1.8	-1.6	-3.2	-3.4	→	+2	-2	-3	-3
+2.4	-3.5	-2.8	-3.9		+2	-4	-3	-4
+0.1	-3.8	+2.4	+3.4		+0	-4	+2	+3
+0.9	+3.3	-1.9	-2.3		+1	+3	-2	-2

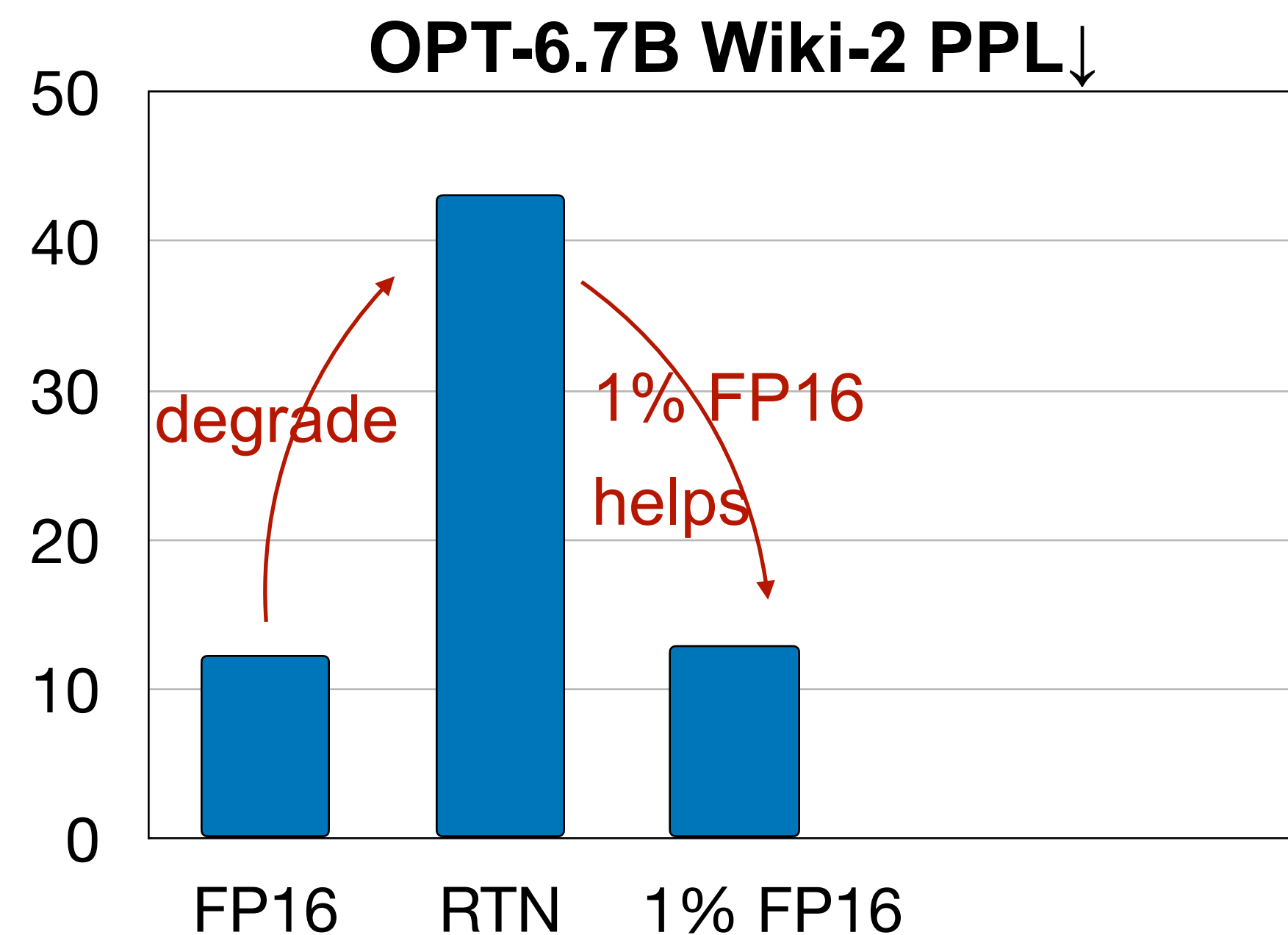
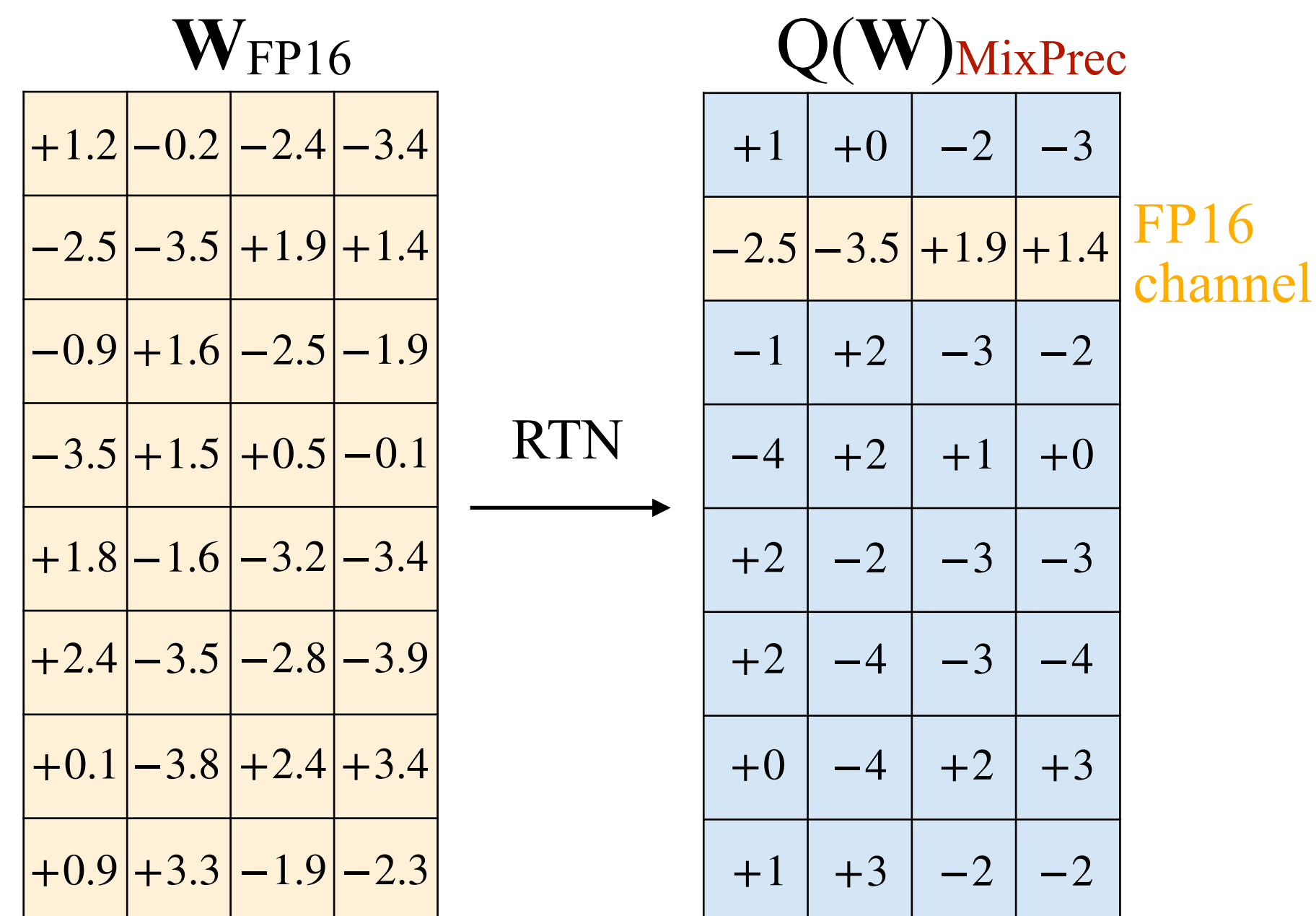


- Weight-only quantization reduces the memory requirement, and accelerates token generation by alleviating the memory bottleneck.
- Group-wise/block-wise quantization (e.g., 64/128/256) offers a better accuracy-model size trade-off.
- But there is still a performance gap with round-to-nearest (RTN) quantization (INT3-g128)

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration (Lin et al., 2023)

# AWQ for Low-bit Weight-only Quantization

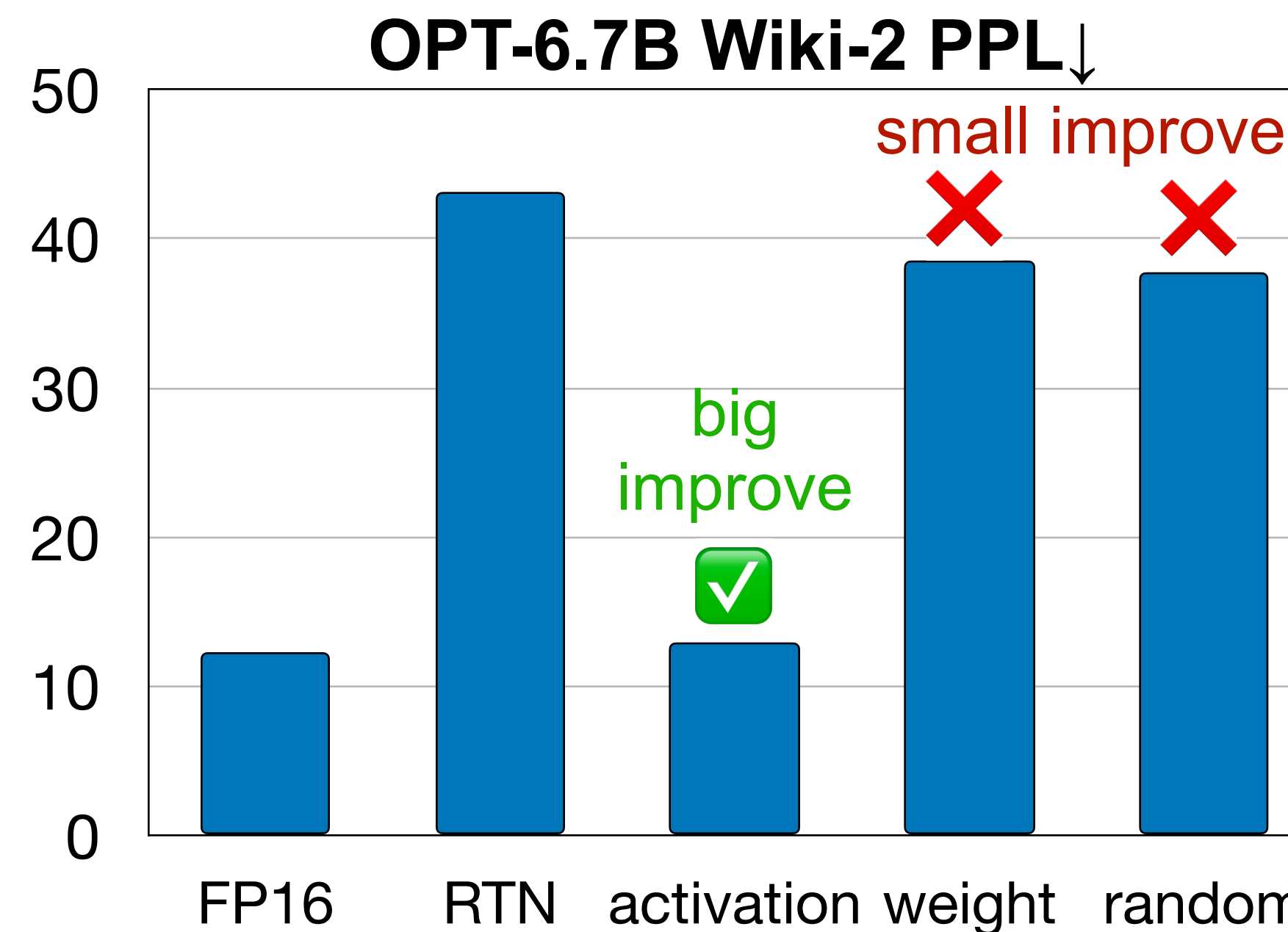
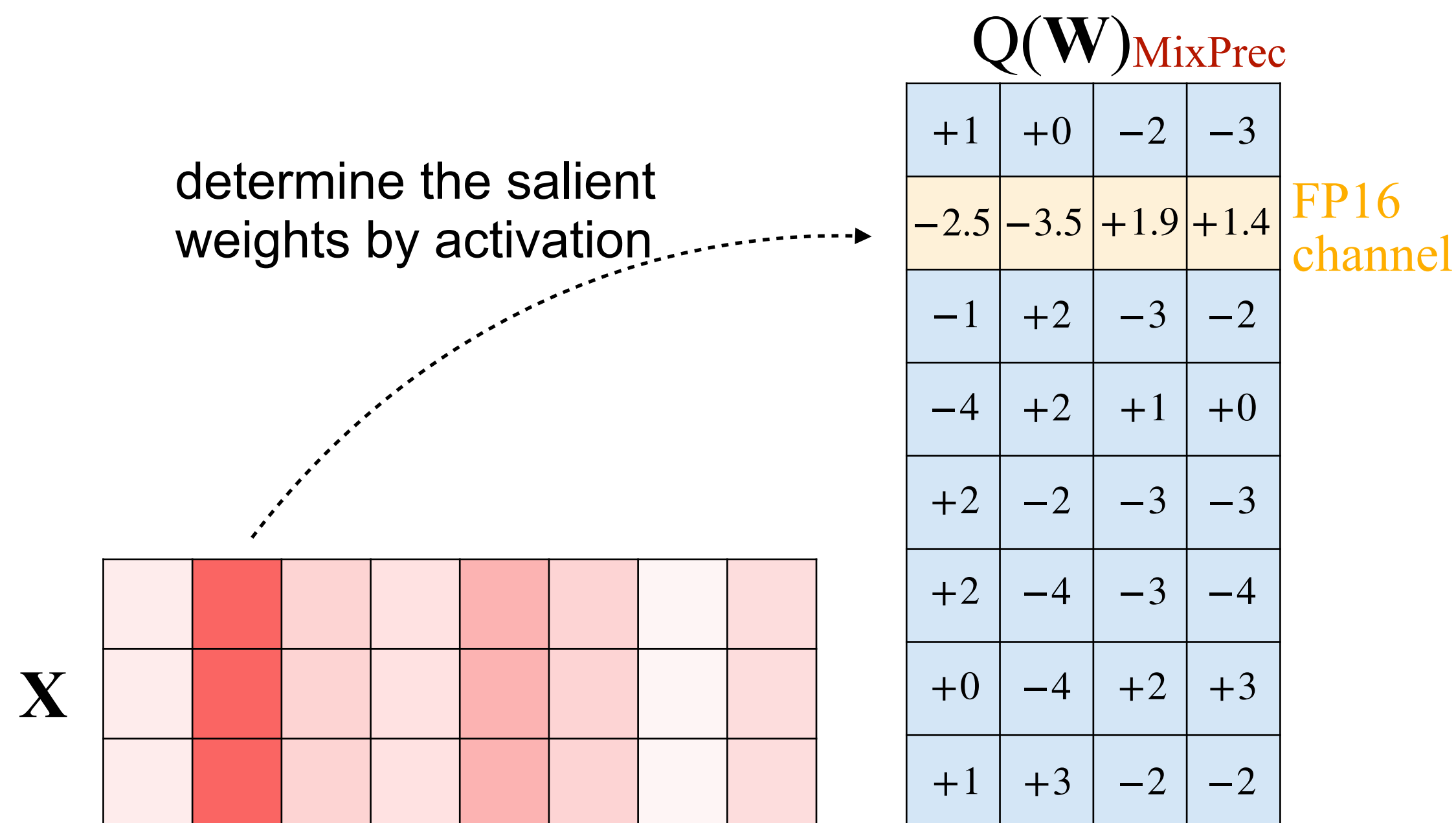
Observation: Weights are not equally important; 1% salient weights



- We find that weights are not equally important, keeping **only 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 for Low-bit Weight-only Quantization

Salient weights are determined by activation distribution, not weight

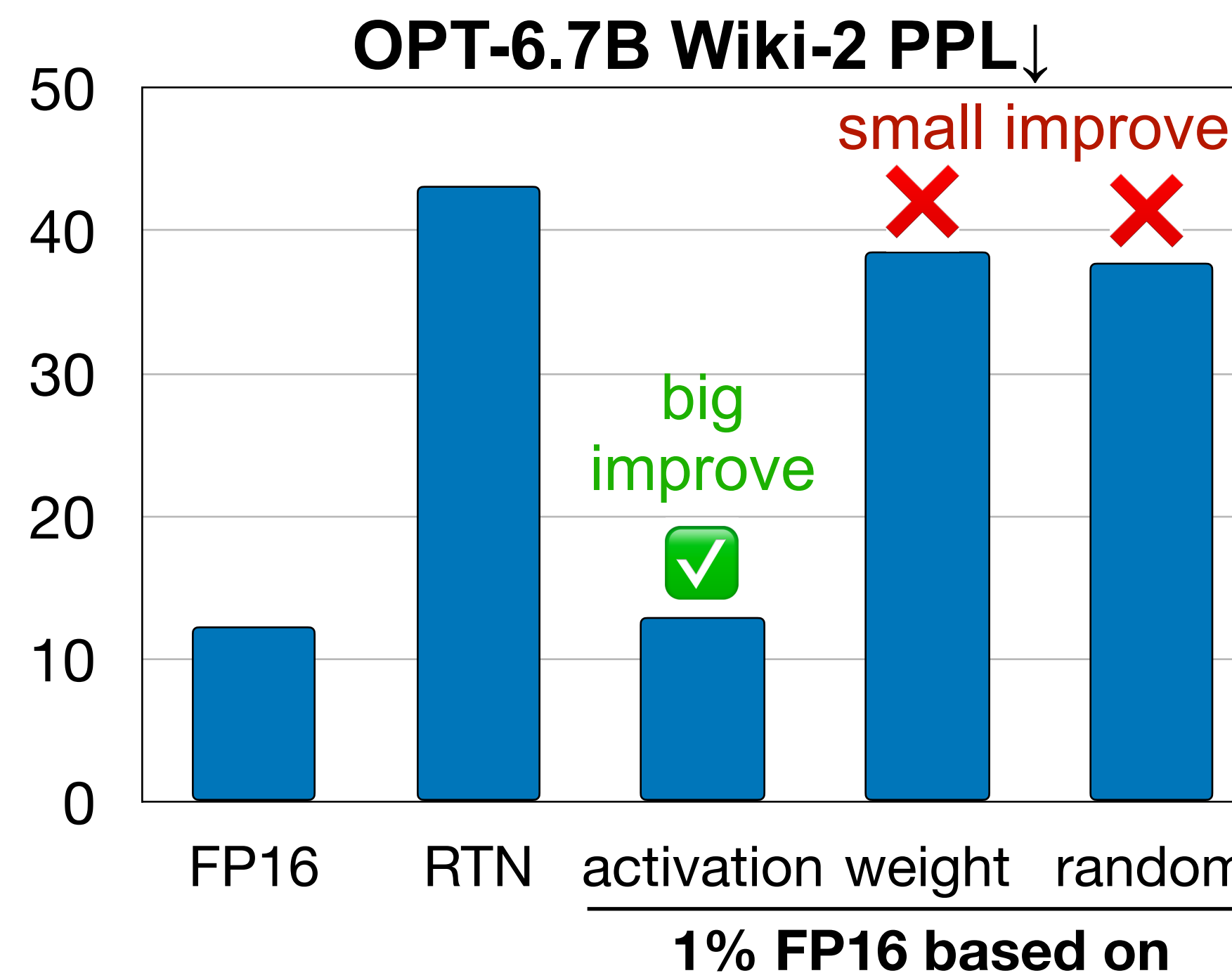
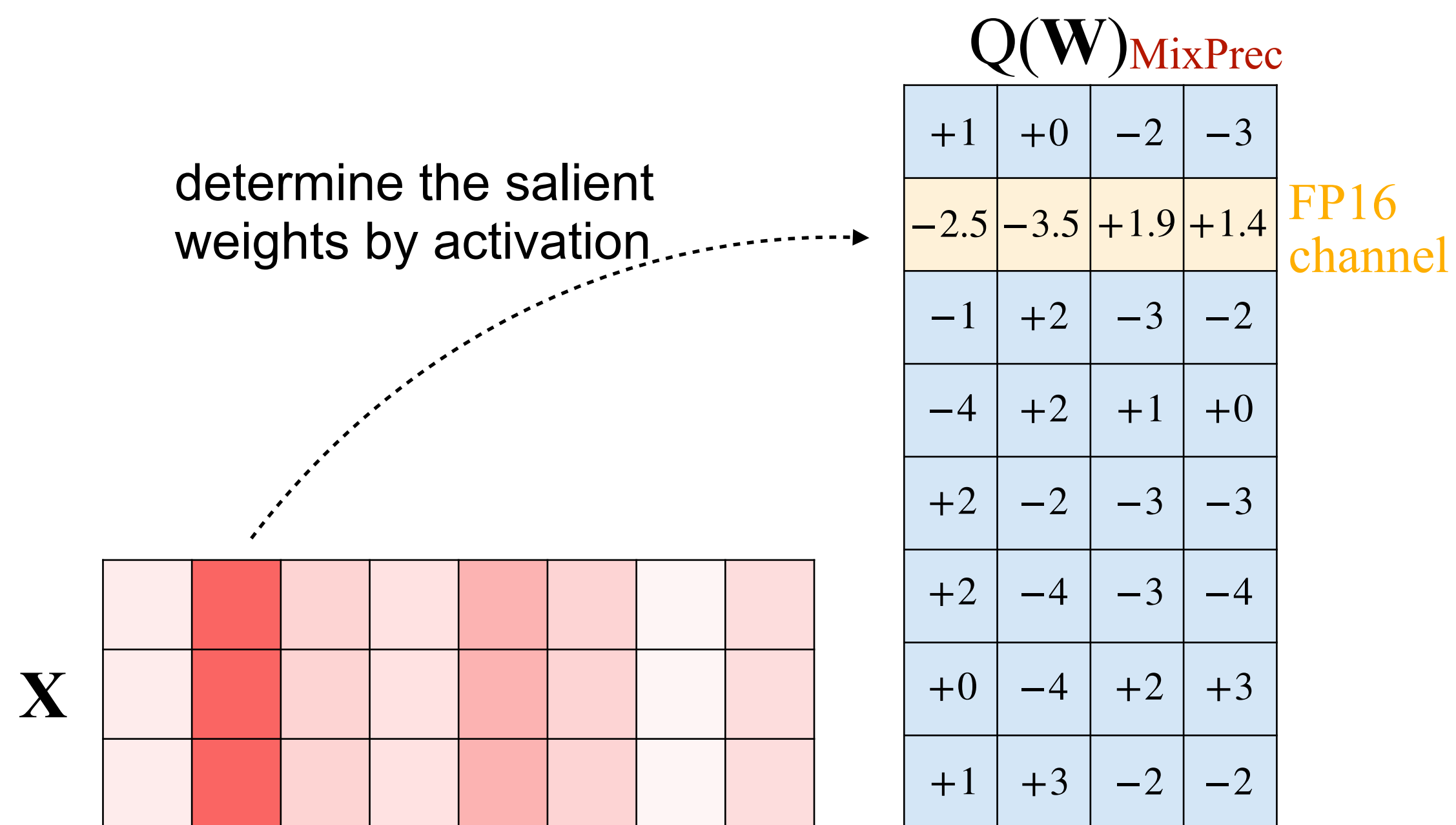


1% FP16 based on

- We find that weights are not equally important, keeping **only 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 for Low-bit Weight-only Quantization

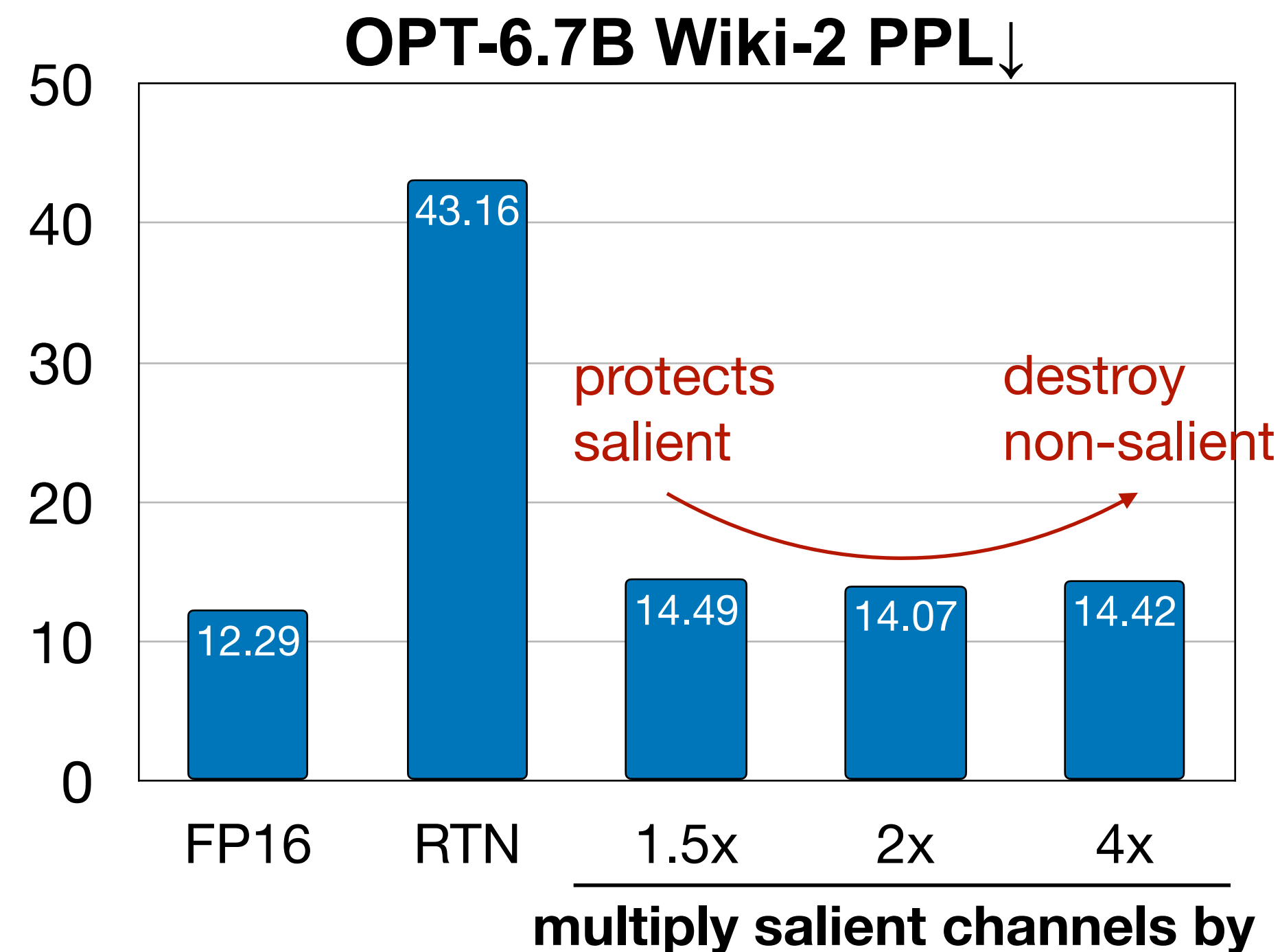
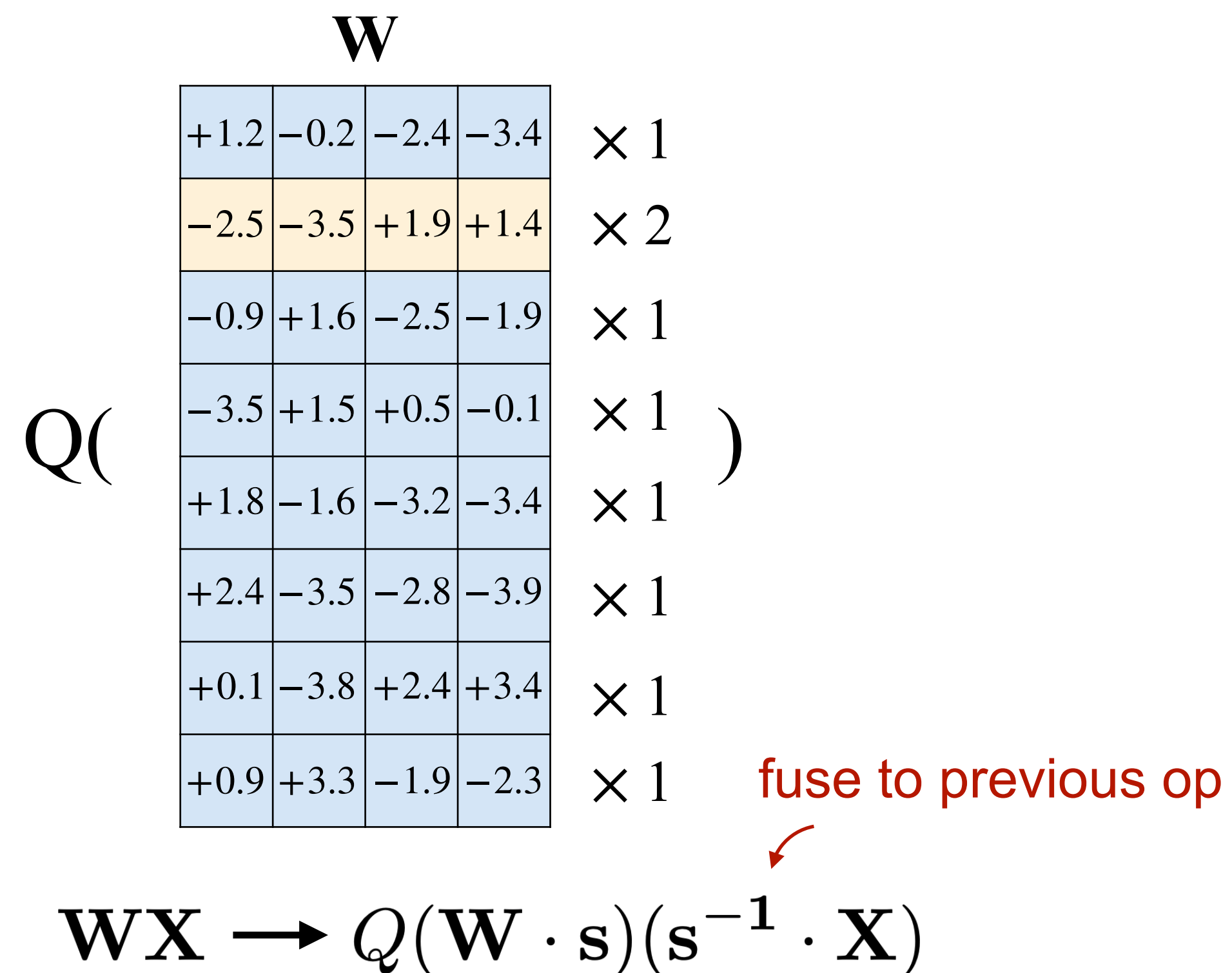
Salient weights are determined by activation distribution, not weight



- **Pro:** improve quantized performance with a negligible overhead (only 1%)
- **Con:** bad hardware efficiency due to mixed-precision weights
  - A recently work SpQR only manages to get 15% speed-up

# AWQ for Low-bit Weight-only Quantization

Protecting salient weights by scaling (no mixed prec.)



- Multiplying the salient channels with  $s > 1$  reduces its quantization error
- Why?



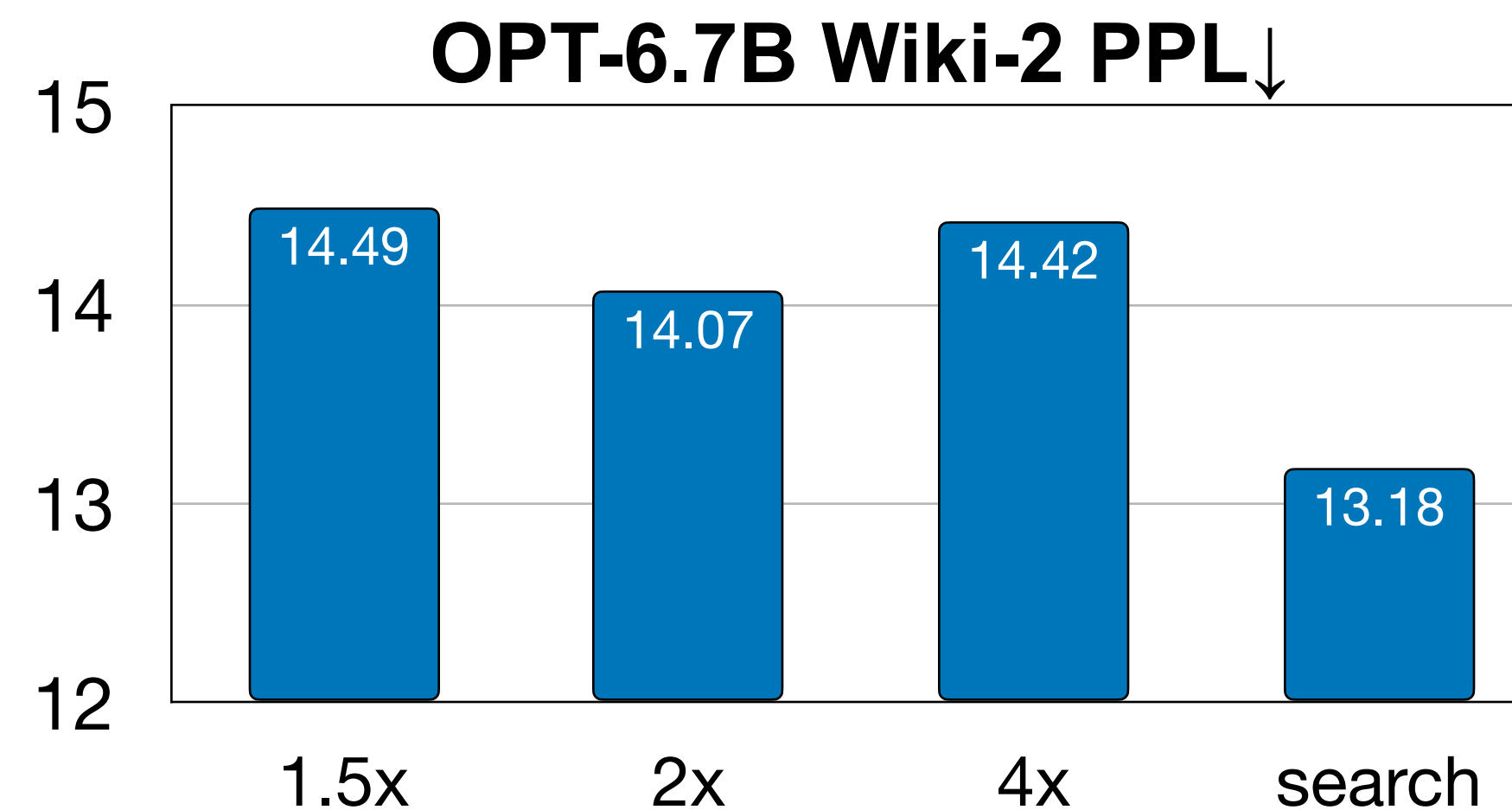
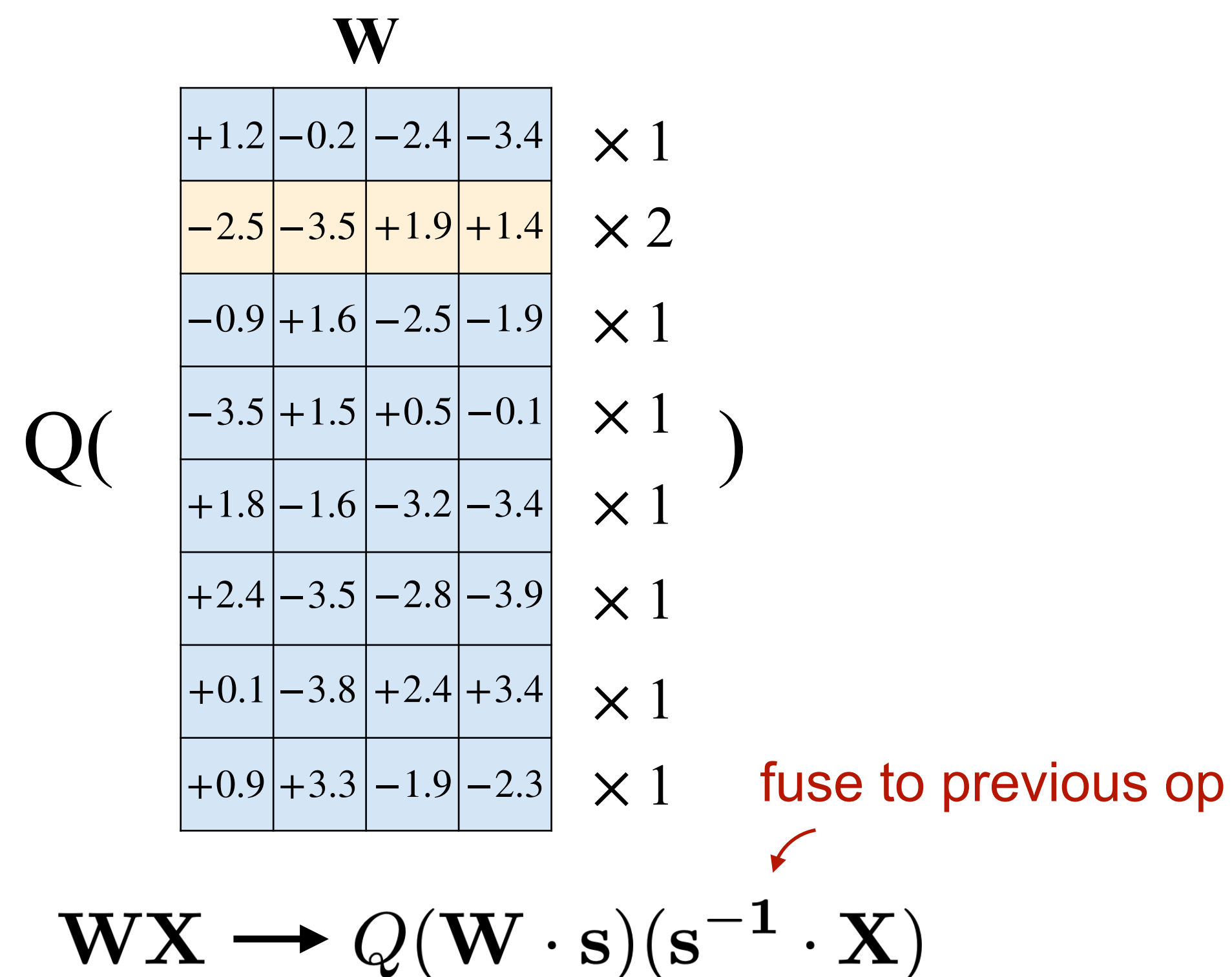
# AWQ for Low-bit Weight-only Quantization

## Protecting salient weights by scaling (no mixed precision)

- Consider a linear layer channel  $\mathbf{y} = \mathbf{w}x$  (from  $\mathbf{W}\mathbf{x}$ ). We care about the quantization error from  $Q(\mathbf{w})x$
- $Q(\mathbf{w}) = \Delta \cdot \text{Round}(\mathbf{w}/\Delta)$ ,  $\Delta = \frac{\max(|\mathbf{w}|)}{2^{N-1}}$
- The scaled version is  $Q(\mathbf{w} \cdot s)(x/s) = \underline{\Delta} \cdot \underline{\text{Round}(s\mathbf{w}/\Delta)} \cdot x \cdot \underline{\frac{1}{s}}$
- We find that the error from Round() is always  $\sim 0.25$  (average from 0-0.5)
- The maximum value in a group “usually” does not change if we just scale up a channel  $\rightarrow \Delta$  not changed
- With  $s > 1$ , the error is scaled down.

# AWQ for Low-bit Weight-only Quantization

Protecting salient weights by scaling (no mixed prec.)



multiply salient channels by

$$\mathcal{L}(\mathbf{s}) = \|Q(\mathbf{W} \cdot \mathbf{s})(\mathbf{s}^{-1} \cdot \mathbf{X}) - \mathbf{WX}\|$$

$\mathbf{s} = \mathbf{s}_X^\alpha$  **Activation-awareness** is important, but not weight-awareness

- Multiplying the salient channels with  $s > 1$  reduces its quantization error
- Take a data-driven approach with a fast **grid search**

# AWQ for Low-bit Weight-only Quantization

Better PPL under low-bit weight-only quantization

PPL↓		Llama-2			LLaMA			
		7B	13B	70B	7B	13B	30B	65B
FP16	-	5.47	4.88	3.32	5.68	5.09	4.10	3.53
INT3 g128	RTN	6.66	5.52	3.98	7.01	5.88	4.88	4.24
	GPTQ	6.43	5.48	3.88	8.81	5.66	4.88	4.17
	GPTQ-R	6.42	5.41	3.86	6.53	5.64	4.74	4.21
	<b>AWQ</b>	<b>6.24</b>	<b>5.32</b>	<b>3.74</b>	<b>6.35</b>	<b>5.52</b>	<b>4.61</b>	<b>3.95</b>
INT4 g128	RTN	5.73	4.98	3.46	5.96	5.25	4.23	3.67
	GPTQ	5.69	4.98	3.42	6.22	5.23	4.24	3.66
	GPTQ-R	5.63	4.99	3.43	5.83	5.20	4.22	3.66
	<b>AWQ</b>	<b>5.60</b>	<b>4.97</b>	<b>3.41</b>	<b>5.78</b>	<b>5.19</b>	<b>4.21</b>	<b>3.62</b>

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration (Lin et al., 2023)

# AWQ for Low-bit Weight-only Quantization

Also works for multi-modal LLMs (OpenFlamingo-9B, captioning)

COCO (CIDEr $\uparrow$ )		0-shot	4-shot	8-shot	16-shot	32-shot	$\Delta(32-shot)$
FP16	-	63.73	72.18	76.95	79.74	81.70	-
INT4 g128	RTN	60.24	68.07	72.46	74.09	77.13	-4.57
	GPTQ	59.72	67.68	72.53	74.98	74.98	-6.72
	AWQ	<b>62.57</b>	<b>71.02</b>	<b>74.75</b>	<b>78.23</b>	<b>80.53</b>	<b>-1.17</b>
INT3 g128	RTN	46.07	55.13	60.46	63.21	64.79	-16.91
	GPTQ	29.84	50.77	56.55	60.54	64.77	-16.93
	AWQ	<b>56.33</b>	<b>64.73</b>	<b>68.79</b>	<b>72.86</b>	<b>74.47</b>	<b>-7.23</b>



W4-RTN: A model airplane **flying in the sky**.

W4-AWQ: Two toy airplanes **sit on a grass field**.



W4-RTN: A man is **holding a baby elephant** in his arms.

W4-AWQ: A man and his daughter **pose with an elephant**.



W4-RTN: **A man and a dog** walking past some bushes.

W4-AWQ: **Two dogs** are walking on the street.

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration (Lin et al., 2023)

# AWQ for Low-bit Weight-only Quantization

Also works for multi-modal LLMs (LLaVA, visual reasoning)



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



**Question:** Can you explain this meme in detail?

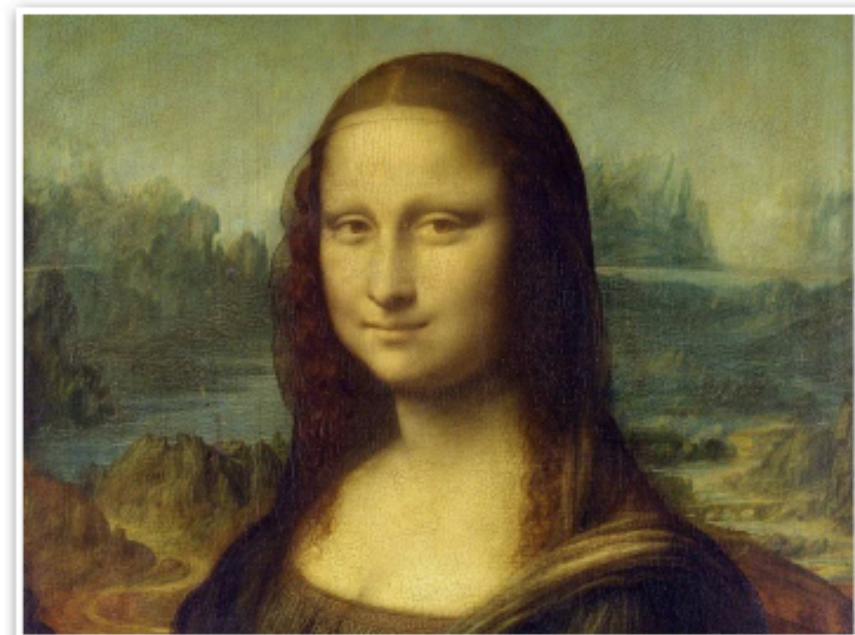
**INT4-g128 RTN:** The meme in the image features a plate of food, likely fried chicken, with a unique twist. **There are small pictures of the Earth and other planets placed on top of the food**, creating an interesting and amusing visual.

Alongside the plate, a caption reads, "Sometimes I look at pictures of the Earth from space and I marvel at how beautiful it all is." This combination of food and planet images adds a touch of humor and curiosity, making the viewer wonder about the context and intent behind the meme.

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

# AWQ for Low-bit Weight-only Quantization

Also works for multi-modal LLMs (LLaVA, visual reasoning)



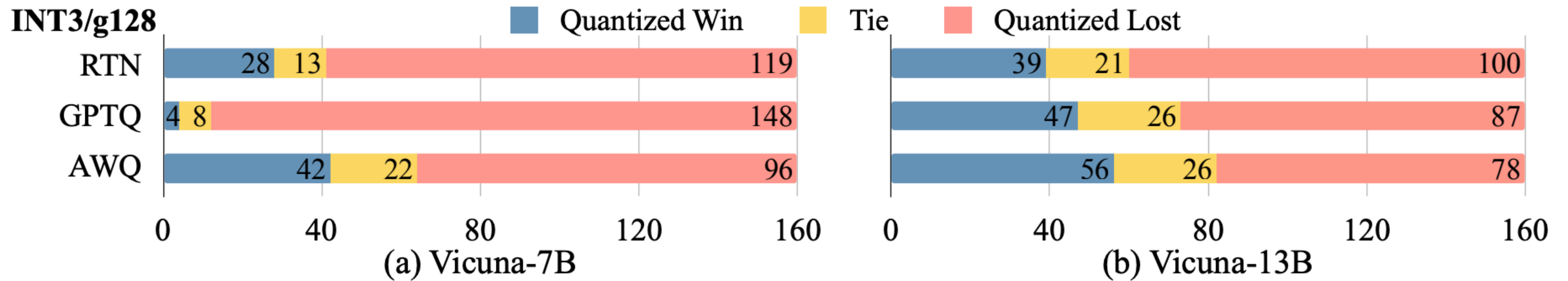
**Question:** Do you know who drew this painting?

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

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

# Results

## Quantization of instruction-tuned models



- Comparing quantized Vicuna with FP16 counterparts
- *W4 almost preserves performance*

# Results

## Quantization for complex generations (code & math)

<b>MBPP (7B)</b>	pass@1	pass@10	<b>GSM-8K</b>	7B	13B	70B
FP16	38.53	49.77	FP16	13.87	26.16	56.41
RTN	37.51	48.49	RTN	11.07	21.23	53.98
GPTQ	31.97	44.75	GPTQ	12.13	24.26	56.03
AWQ	<b>40.64</b>	<b>49.25</b>	AWQ	<b>13.57</b>	<b>25.25</b>	<b>56.40</b>

- **MBPP**: a Python coding dataset
- **GSM-8K**: a math reasoning dataset (requires multi-step reasoning)
- AWQ preserves the accuracy under **W4-g128** quantization



# Industry & Community Impact

SmoothQuant and AWQ are widely used:



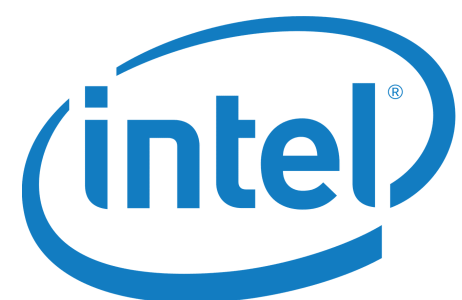
FasterTransformer  
TRT-LLM

[https://github.com/NVIDIA/FasterTransformer/blob/main/docs/gpt\\_guide.md](https://github.com/NVIDIA/FasterTransformer/blob/main/docs/gpt_guide.md)  
<https://github.com/NVIDIA/TensorRT-LLM#key-features>



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)



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)



Imdeploy

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

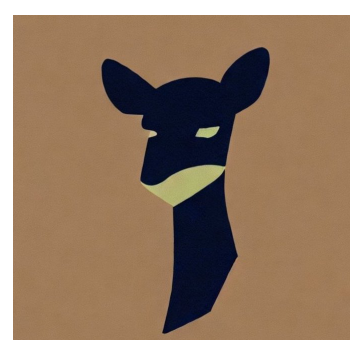


[https://github.com/vllm-project/vllm/blob/main/vllm/model\\_executor/quantization\\_utils/awq.py](https://github.com/vllm-project/vllm/blob/main/vllm/model_executor/quantization_utils/awq.py)



oobabooga/  
text-  
generation-  
webui

<https://github.com/oobabooga/text-generation-webui/blob/main/modules/models.py>



lm-sys/FastChat

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

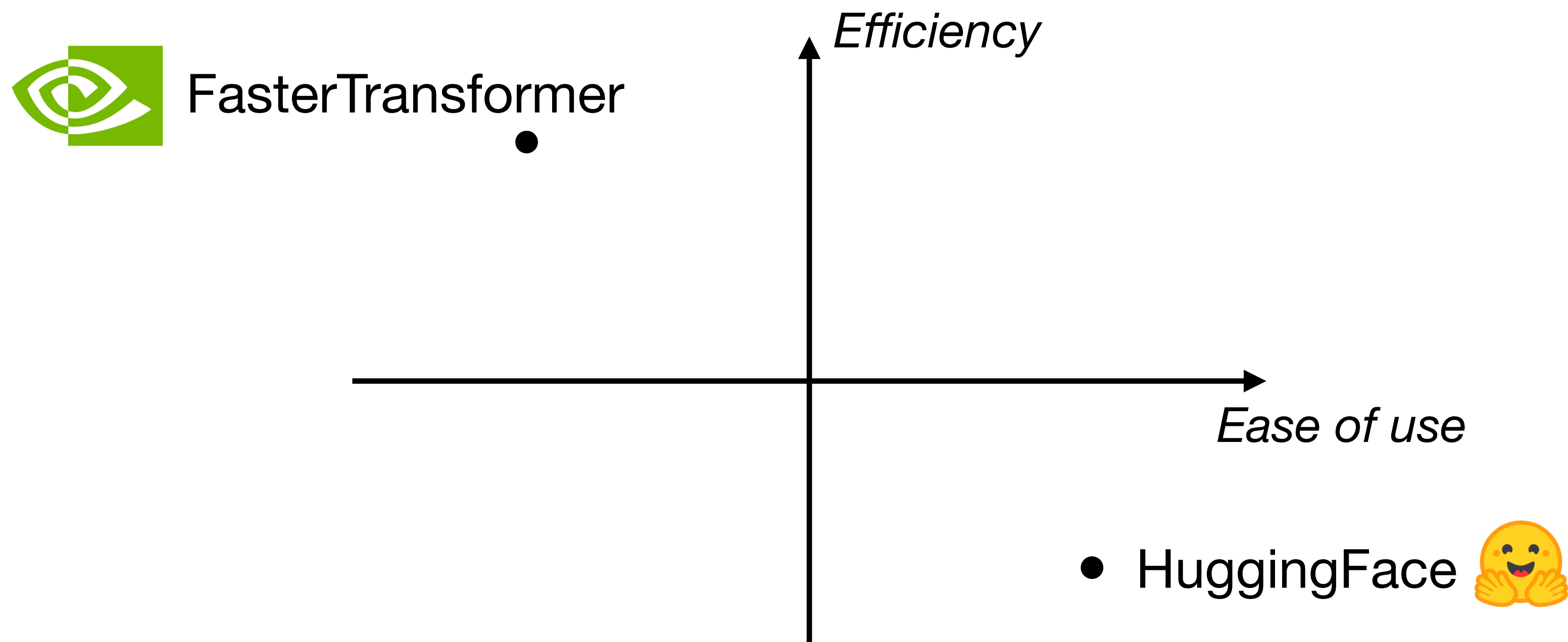


[https://github.com/replicate/vllm-with-loras/blob/main/vllm/model\\_executor/quantization\\_utils/awq.py](https://github.com/replicate/vllm-with-loras/blob/main/vllm/model_executor/quantization_utils/awq.py)

# TinyChat: A Lightweight Serving Infra

Pythonic, lightweight, efficient

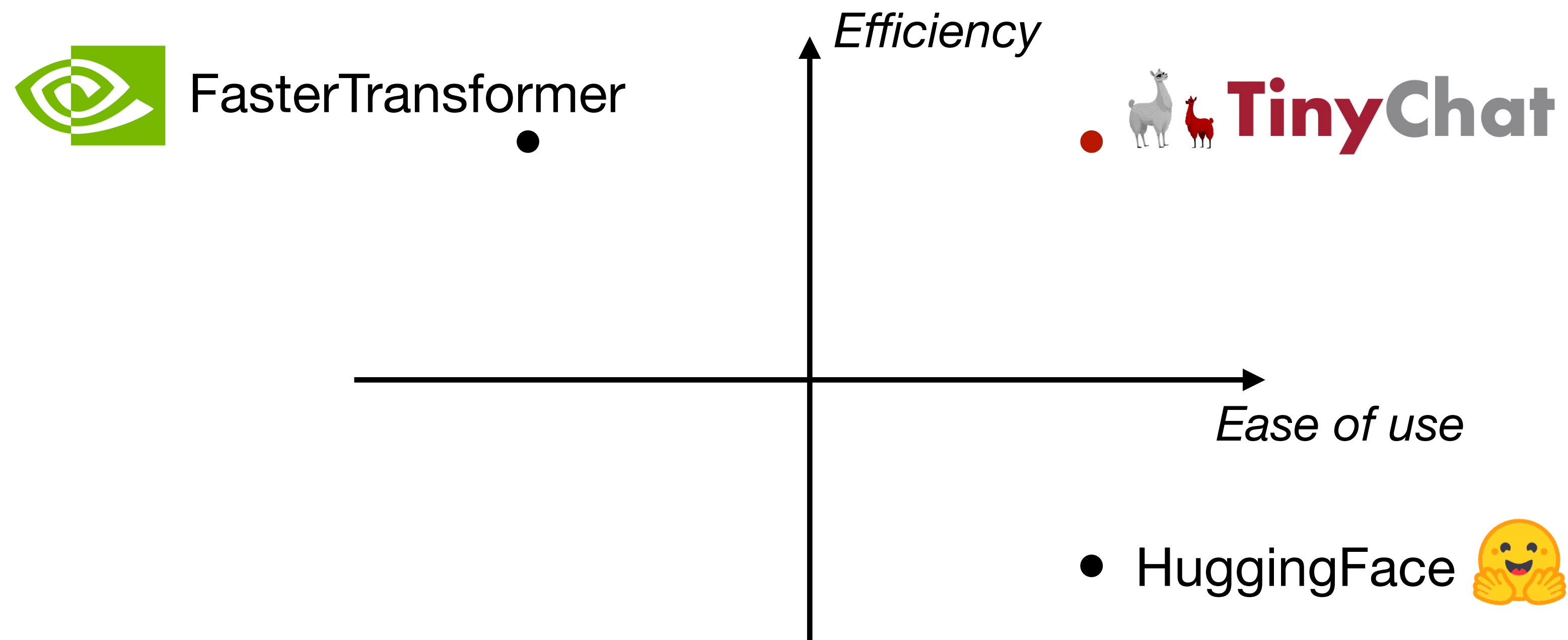
- We need a framework to serve the quantized model to achieve low latency (AWQ only for Linears)
  - HuggingFace: easy to use, but slow
  - FasterTransformer: high efficiency, but harder to use



# TinyChat: A Lightweight Serving Infra

Pythonic, lightweight, efficient

- We need a framework to serve the quantized model to achieve low latency
  - HuggingFace: easy to use, but slow
  - FasterTransformer: high efficiency, but harder to use
- **TinyChat** goals: efficient, lightweight, Python-native (composable with other stacks like vLLM)



# TinyChat: A Lightweight Serving Infra

## Analyze the latency overhead under FP16

- Measurement based on LLaMA-7B on RTX4090

Technique	Tok/sec	Ratio
Huggingface impl.	49.0	76%
FasterTransformer impl.	64.1	100%

# TinyChat: A Lightweight Serving Infra

## Analyze the latency overhead under FP16

- Measurement based on LLaMA-7B on RTX4090
- Some overheads can be easily removed! 95% of FT performance in Python

Technique	Tok/sec	Ratio
Huggingface impl.	49.0	76%
Preallocate KV cache	54.1	84%
FT LayerNorm kernel	57.5	90%
FlashAttention	57.5	90%
Merge QKV projections	59.2	92%
Fuse rotary embedding	61.0	95%
FasterTransformer impl.	64.1	100%

# TinyChat: A Lightweight Serving Infra

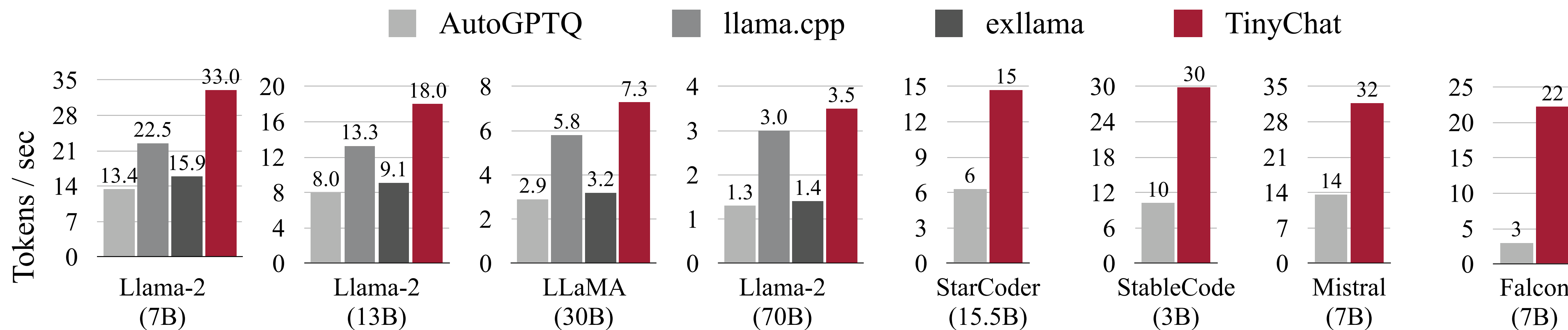
## State-of-the-art W4 inference speed

- Now we plugin the AWQ to quantize the weights into 4-bit (further 3x improvement)
- We can outperform the state-of-the-art MLC-LLM (TVM compilation-based) with our **Pythonic** solution
- 50% faster on AGX Orin

LLaMA-7B	RTX 4090 Tok/sec	AGX Orin Tok/sec
llama.cpp	141	22.5
Exllama	153	15.9
MLC-LLM	191	-
TinyChat	<b>195</b>	30.2

# TinyChat: A Lightweight Serving Infra

Supporting a wide range of models on NVIDIA Jetson Orin



Latency comparison on Jetson Orin (64G) mobile GPU

- TinyChat achieves up to **1.5x** faster runtime for Meta's Llama models compared with systems specialized for this model.
- Compared with the only competitor that can support a diverse range of models, TinyChat is up to **7x** faster.
- Remarkably, TinyChat's front end is **fully PyTorch-based**.

# TinyChat: A Lightweight Serving Infra

## Demo on AGX Orin (edge LLM inference)

- Orin Nano has 200GB/s memory bandwidth; even more memory-bounded
- Model size: 7B. ~**30 token/s** generation

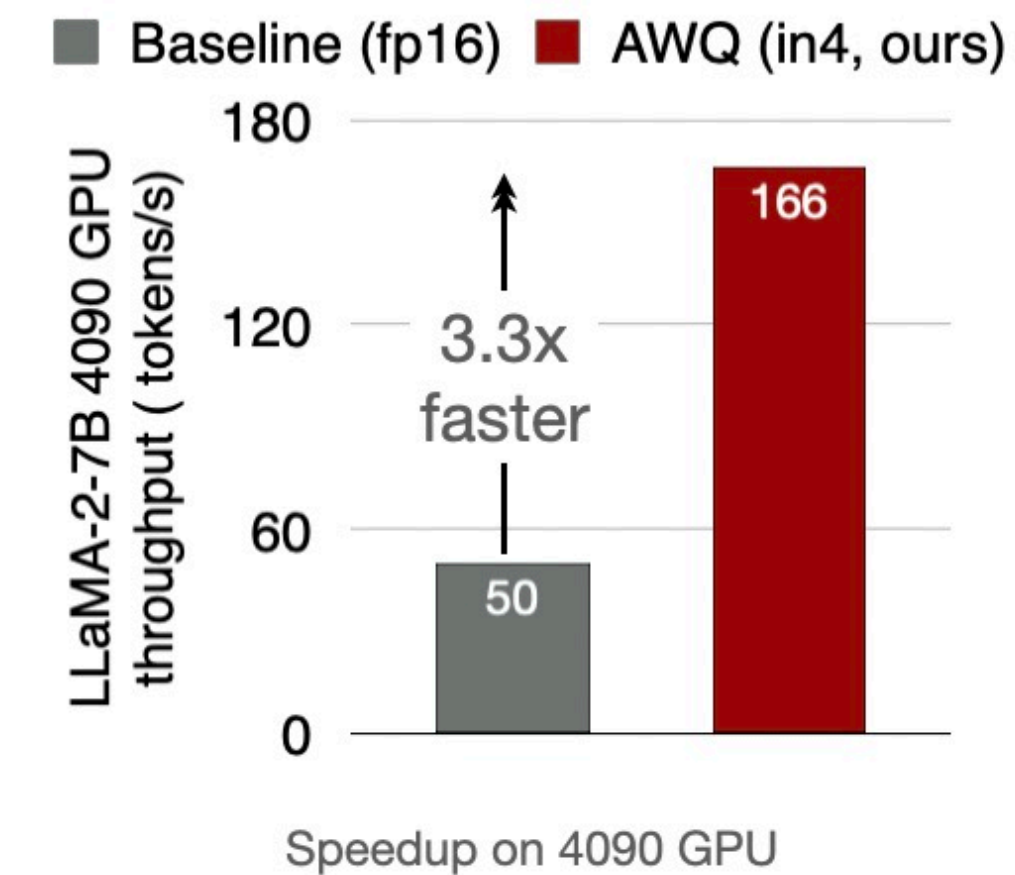


```
File Edit Selection View Go ... orin [SSH: orin] - Visual Studio Code
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE sh - AWQ-Chat + - | 1/1 [00:02<00:00, 2.99s/it]
(AWQ-Chat) orin0% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100%|
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT: |
```

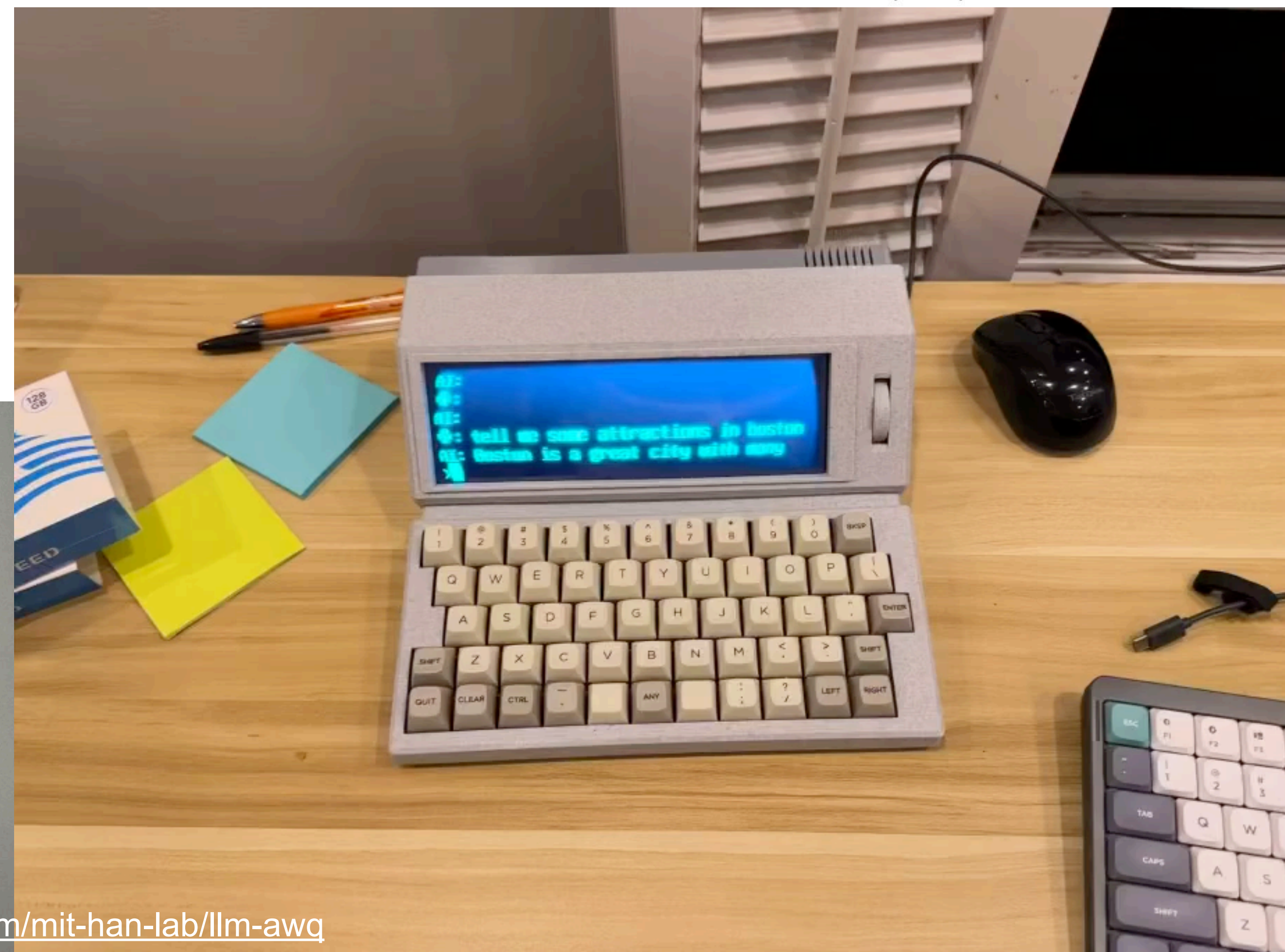




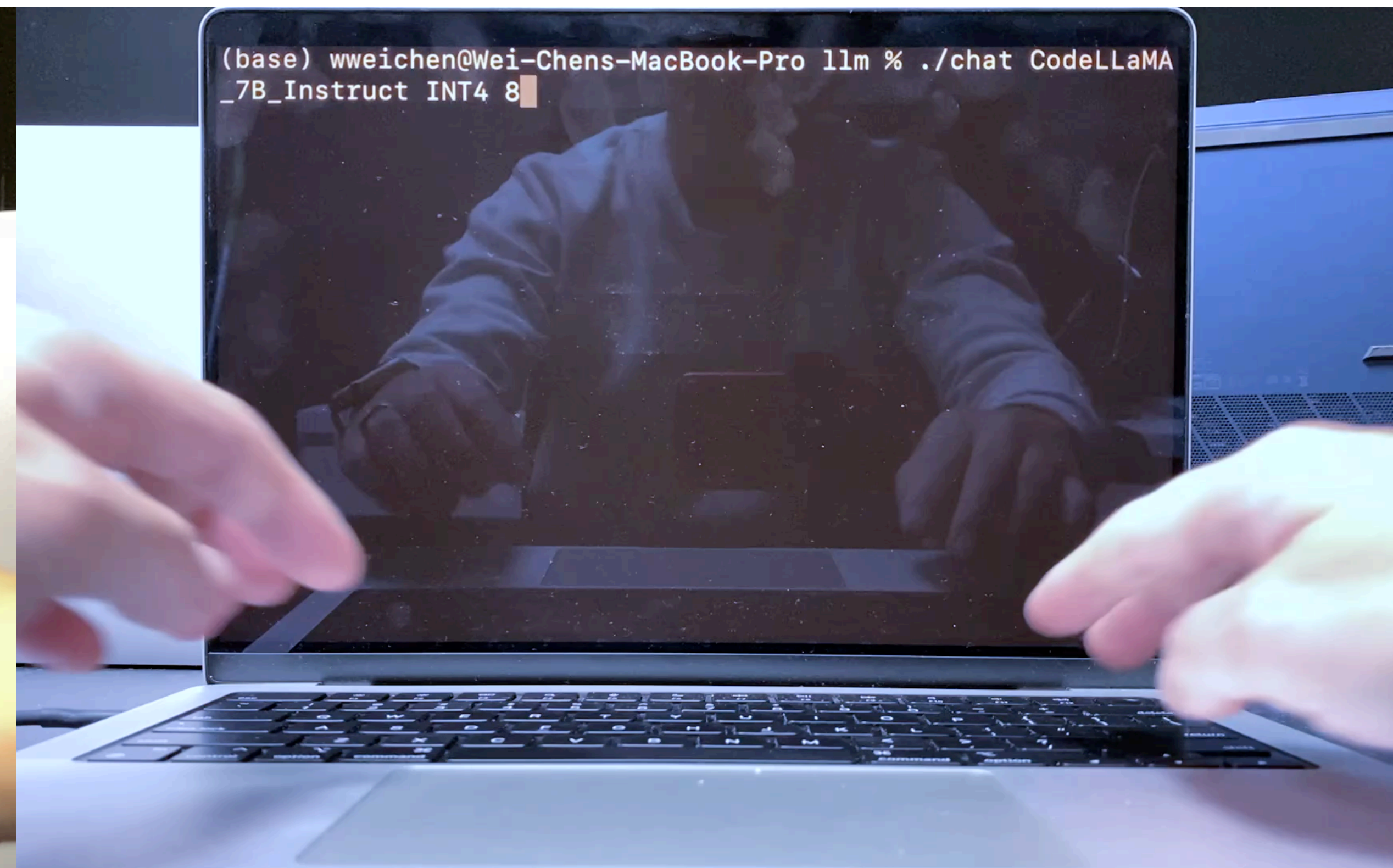
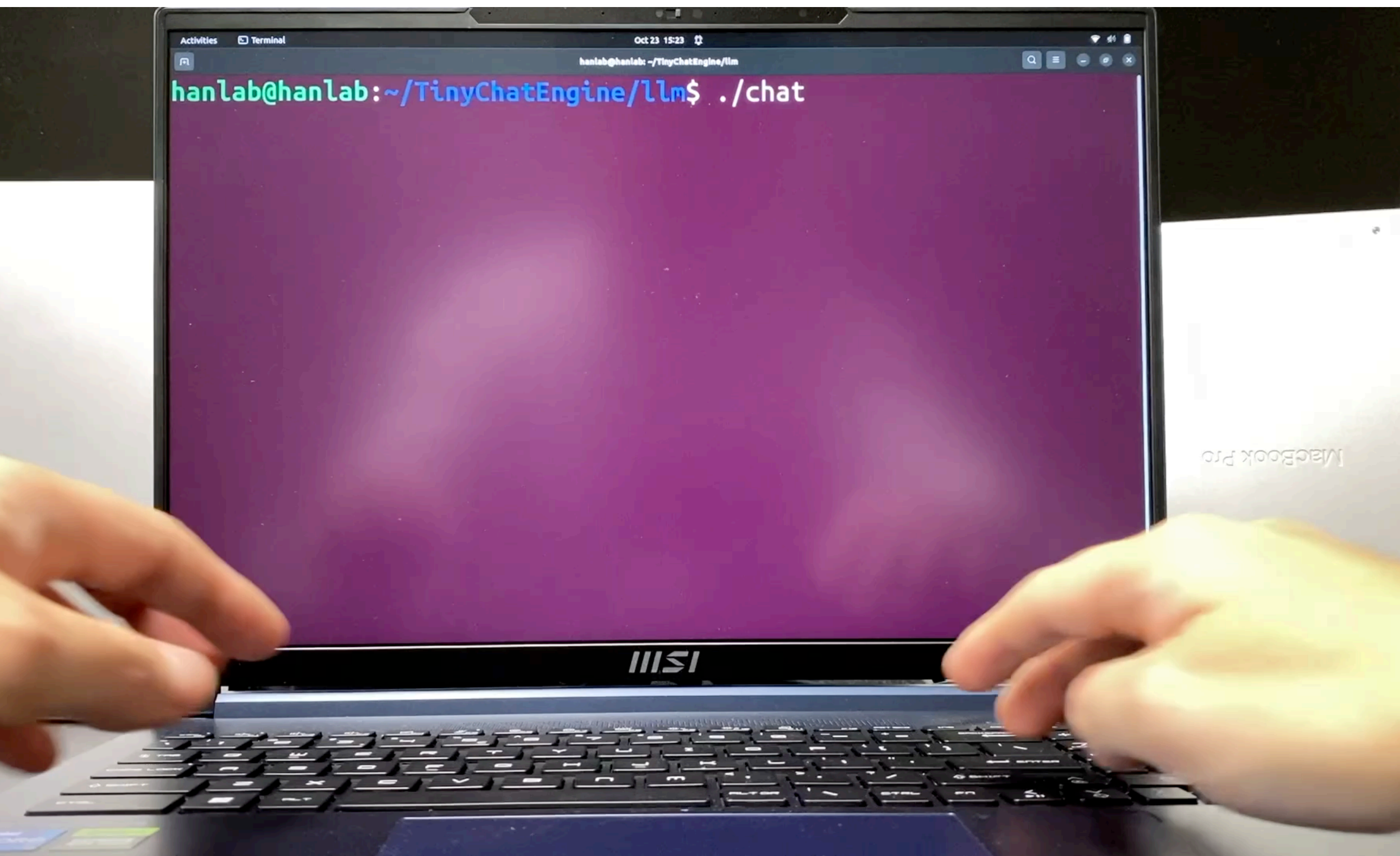
# TinyChat



- 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.



- 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



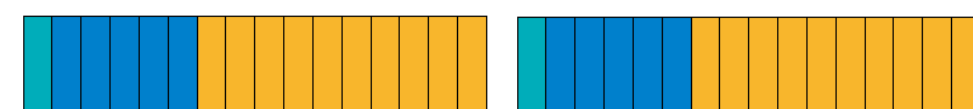


# TinyChat brings about 3.3x speedup to LLaMA-2 on 4090

```
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: |
```

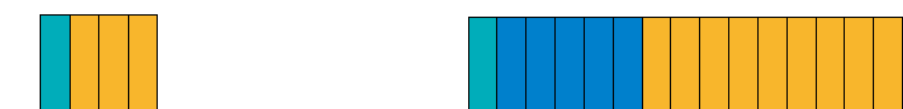
```
shang [SSH: Hanlab_4090] - Visual Studio Code
sh - AWQ-Chat
(AWQ-Chat) x4% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100% | 1/1 [00:01<00:00, 1.27s/it]
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT: Of course! Here are five popular attractions in the Greater Boston Area:
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 flexibly supports different LLM architectures

```

(AWQ-Chat) orin0% ./mpt_awq_int4.sh 7b
real weight quantization...(init only): 100% | ██████████ 32/32 [00:00<00:00, 314.76it/s]
Loading checkpoint: 100% | ██████████ 1/1 [00:01<00:00, 1.54s/it]
USER: How do you compare MIT and Harvard?
ASSISTANT: As an AI language model, I don't have personal experiences or preferences, so I cannot provide an objective comparison between these two prestigious universities. However, both of them are well-known for their academic excellence in various fields such as science, technology, engineering, and mathematics (STEM). They consistently rank among the top universities worldwide in terms of academic reputation, research output, and employability of graduates. Ultimately, the choice between MIT and Harvard depends on individual preferences and goals, such as location, program offerings, admission requirements, financial aid opportunities, and career prospects after graduation. It is essential to conduct thorough research and consult with experts before making a decision about which university to attend.
=====
Speed of Inference
-----
Generation Stage : 31.29 ms/token
=====
USER: █

```

```

(AWQ-Chat) orin0% ./falcon_awq_int4.sh 7b
real weight quantization...(init only): 100% | ██████████ 32/32 [00:00<00:00, 127.73it/s]
Loading checkpoint: 100% | ██████████ 1/1 [00:02<00:00, 2.72s/it]
USER: How do you compare MIT and Harvard?
ASSISTANT: There are many differences between MIT and Harvard, but both universities have a reputation for rigorous academic programs, cutting-edge research, and talented faculty. MIT is known for its science and engineering programs, while Harvard is known for its humanities and social science programs. Additionally, Harvard has a larger endowment and more alumni giving, making it financially stronger than MIT. However, both universities share a commitment to excellence and are highly selective in their admissions process.
=====
Speed of Inference
-----
Generation Stage : 45.90 ms/token
=====
USER: █

```

```

(AWQ-Chat) orin0% ./vicuna_awq_int4.sh 7b
Loading checkpoint: 100% | ██████████ 1/1 [00:01<00:00, 1.69s/it]
USER: How do you compare MIT and Harvard?
ASSISTANT: MIT and Harvard are both highly prestigious universities with strong reputations in various fields of study. However, they have some key differences that set them apart from each other.

One significant difference is their academic focus. MIT is known for its emphasis on science, technology, engineering, and mathematics (STEM) fields, while Harvard has a more diverse range of programs across the humanities, social sciences, law, medicine, and business. As such, MIT is often seen as the "technology school" and Harvard as the "liberal arts school."

In terms of research, both institutions are among the top performers in terms of funding and number of patents granted, but MIT is generally considered to be more focused on applied research, while Harvard is known for its more theoretical and interdisciplinary research.

Overall, it's difficult to say which one is "better" since it depends on what you're looking for in a university education. If you're interested in STEM fields or want to pursue a career in academia, MIT might be the better choice. If you're looking for a broader liberal arts education or want to go into a field like law or business, Harvard might be the better fit.
=====
Speed of Inference
-----
Generation Stage : 30.50 ms/token
=====
USER: █

```

```

Homework: Read about different types of parallelism and complete a short quiz.
Week 2: Parallel Programming Languages
- Introduction to parallel programming languages (MPI, OpenMP, LINUX threads)
- Basic syntax and features of each language
- Examples of how to use each language to solve parallel problems
Homework: Write a simple parallel program using MPI or OpenMP and submit it for grading.
Week 3: Performance Analysis and Optimization
- How to measure and analyze the performance of parallel programs?
- Techniques for optimizing parallel programs (vectorization, loop fusion, pipelining)
- Understanding the tradeoffs between parallelism and efficiency
Homework: Analyze the performance of a parallel program and optimize it using different techniques.
Week 4: Multicore and Manycore Systems
- Introduction to modern CPUs and GPUs
- Understanding the differences between multicore and manycore systems
- How to design and implement efficient parallel algorithms on these systems
Homework: Design and implement a parallel algorithm that scales well on a manycore system.
Week 5: Applications of Parallel Computing
- Overview of some important applications areas for parallel computing (scientific simulations, machine learning, data analytics)
- Case studies of successful parallel computing projects in these areas
- Discussion of challenges and opportunities in applying parallel computing to real-world problems
Home homework: Research and discuss a real-world application of parallel computing and its challenges

Note: The above syllabus is just a sample and can be modified based on specific requirements and preferences.
=====
Speed of Inference
-----
Generation Stage : 31.27 ms/token
=====
USER: █

```

```

(AWQ-Chat) orin0% ./falcon_awq_int4.sh 7b
real weight quantization...(init only): 100% | ██████████ 32/32 [00:00<00:00, 140.33it/s]
Loading checkpoint: 100% | ██████████ 1/1 [00:02<00:00, 2.67s/it]
USER: Make a detailed 5-week syllabus for Parallel Computing.
ASSISTANT: Week 1: Introduction to Parallel Computing - Understanding the basics of parallel computing and its applications.

Week 2: Single Processor Systems - Understanding the concept of a single processor and the limitations of a single processor system.

Week 3: Multi-Processor Systems - Understanding the concept of multi-processor systems and their advantages.

Week 4: Parallel Programming Models - Understanding the different types of parallel programming models, including shared memory, distributed memory, and many-to-many communication patterns.

Week 5: Parallel Programming - Writing and executing parallel programs using a programming language and understanding the concepts of parallelism and data locality.
=====
Speed of Inference
-----
Generation Stage : 45.25 ms/token
=====
USER: █

```

```

* Parallel computing architectures (SMP, MPP, GPU)

Week 2: Parallel Programming Languages
* Introduction to parallel programming languages (MPI, OpenMP, CUDA, etc.)
* Features and limitations of each language
* Best practices for writing efficient parallel code in each language

Week 3: Parallel Algorithms
* Introduction to parallel algorithms (divide and conquer, map reduce, etc.)
* Analysis of algorithmic complexity
* Performance characteristics of different types of algorithms

Week 4: Parallel Architectures and System Design
* Introduction to parallel architectures (SMP, MPP, GPU)
* System design considerations for parallel computing systems
* Memory hierarchy and cache design

Week 5: Case Studies in Parallel Computing
* Real-world applications of parallel computing
* Case studies on large scale data processing, scientific simulations, and high performance computing
* Discussion of best practices and optimization strategies for different workloads.
=====
Speed of Inference
-----
Generation Stage : 30.46 ms/token
=====
USER: █

```



**MPT-7B: 31 tokens / s**



**Falcon-7B: 22 tokens / s**



**Vicuna-7B: 33 tokens / s**



# TinyChat delivers 30 tokens / second performance for LLaMA2

```

File Edit Selection View Go ... orin [SSH: orin] - Visual Studio Code
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE sh - AWQ-Chat + - [X] [X] [X] [X] [X] [X]
(AWQ-Chat) orin0% ./llama2_awq_int4.sh 7b
Loading checkpoint: 100% | ████████████████████████████████████████████████████████████████████████████ | 1/1 [00:02<00:00, 2.99s/it]
USER: Describe five attractions in the Greater Boston Area.
ASSISTANT: Of course! The Greater Boston Area has a plethora of exciting attractions that cater to diverse interests. Here are five popular ones:
1. The Freedom Trail: This 2.5-mile trail takes you through historic sites in Boston, including the Old State House, Faneuil Hall, and the USS Constitution. You'll learn about the city's rich history and witness its iconic landmarks along the way.
2. Museum of Fine Arts: As one of the largest and most visited art museums in the country, the MFA features over 450,000 works of art from around the world, including European, American, and Asian pieces. Don't miss the impressive Egyptian, Greek, and Roman collections.
3. Fenway Park: Known as "America's Most Beloved Ballpark," Fenway Park is home to the Boston Red Sox baseball team. Take a guided tour of the stadium, explore the team's Hall of Fame, and enjoy the lively atmosphere during a game.
4. New England Aquarium: Located on the waterfront, the NEAQ showcases a diverse array of marine life, including penguins, seals, and fish of all shapes and sizes. Be sure to check out the giant Pacific octopus and the shark and ray touch tanks.
5. Harvard University: Established in 1636, Harvard is one of the oldest and most prestigious universities in the US. Take a stroll across the campus, visit the Widener Library, and marvel at the architecture of the Harvard Yard. You might even catch a glimpse of the elusive Harvard Bridge.
Each of these attractions offers a unique experience, so be sure to add them to your itinerary when visiting the Greater Boston Area!
=====
Speed of Inference
-----
Generation Stage : 33.07 ms/token
=====
USER: █

```

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

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

**Thanks for Listening!**