Visual Language Models for Edge AI 2.0

Song Han
Associate Professor, MIT
Distinguished Scientist, NVIDIA

https://songhan.mit.edu
The Need for Efficient AI Computing

Move up the stack, co-design software and hardware

The demand for AI computing is increasing fast

Software is important, the cost is high

[source]
Previous Work

Deep Compression and EIE

Top-5 most cited papers in 50 years of ISCA (1953-2023)

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

Efficient Inference Engine
EfficientML Project

Bridge the supply and demand of AI computing

Algorithm and system co-design for accelerated AI computing

Goal: reduce latency, memory, low power/energy; increase throughput, accuracy, scalability.

Software

Hardware

Tiny Models

Large Models

Inference

Training

Discriminative

Generative

Dense

Sparse

Full-Precision

Quantization

Single-Modality

Multi-Modality

Classic

Quantum

hanlab.mit.edu
Tiny Machine Learning with MCUNet

- TinyML: design light-weighted neural networks and deploy on cheap edge devices that has low power, computing, and memory.
- Billions of IoT devices around the world based on microcontrollers, much cheaper ($1-2), much smaller, everywhere in our lives, but very memory-constraint.
- MCUNet and TinyEngine paves the way for tiny machine learning on edge devices.
Demo
MCUNetV2: Memory-Efficient Patch-based Inference for Tiny Deep Learning

Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, Song Han

mcunet.mit.edu
On-Device Training Under 256KB Memory

Demo Video
ImageNet Pre-trained MCUNet -> VWW
Running on OpenMV Cam H7 MCU
Edge AI 1.0

Train a specific model for each task

Medical image processing  Autonomous driving  Smart manufacturing  Machine translation

Task-specific models
Edge AI 1.0

Train a specific model for each task

Medical image processing  Autonomous driving  Smart manufacturing  Machine translation

Task-specific models

- Need different model / data to train different tasks
- Lack of ‘negative samples’ for training
- Limited generalization; failure of corner cases
Edge AI 2.0

General models with world knowledge

Transformer-based

OpenAI ChatGPT

ChatBots

Scientific Discovery

Software Development

GitHub Copilot
Edge AI 2.0

General models with world knowledge

- One model - multiple tasks
- Enhanced by LM’s world knowledge
- Advanced reasoning capabilities
- Instruction-following proficiency

**Visual Language Model (VLM)**

Landmark recognition
Driving assistant
Patient monitoring
Smart manufacturing
Edge AI 2.0

General models with world knowledge

- One model - multiple tasks
- Enhanced by LM’s world knowledge
- Advanced reasoning capabilities
- Instruction-following proficiency

Visual Language Model (VLM)

VLM Pre-training (VILA) → Model Compression (AWQ) → Efficient Deployment (TinyChat)

World knowledge embedded

• One model - multiple tasks
• Enhanced by LM’s world knowledge
• Advanced reasoning capabilities
• Instruction-following proficiency
VILA: Visual Language Model on the Edge
**ViLA: Visual Language Model**

**Goal:** multi-modal LLM, enhance visual reasoning by language model, enable in-context learning and reasoning across images

**Challenge:** learn from visual inputs without destroying the LLM. Computationally heavy to deploy on the edge.

**Our Solution:** ViLA provides efficient recipe from data curation to training to deployment for VLM. Interleaved image-text pretraining enables in-context learning.

Augments the language model with interleaved image-text

ViLA deployed on Jetson Orin (click to play)

GTC demo booth#: K033

USER: What is unusual about this scene? Are there any safety concerns?

ASSISTANT: In the image, a person is hanging from a rope attached to a wind turbine blade. This is an unusual and potentially dangerous situation as it involves climbing on a moving object high in the air. There are several safety concerns associated with this activity, such as the risk of falling from a significant height, getting injured by the blade or other parts of the wind turbine.
**Prompt:** ... + Think about

**Answer:** The beer on the table is priced at $6.00. Therefore, the total cost is $6.00 x 2 = $12.00.

**VILA:** On Pre-training for Visual Language Models [Lin et al., CVPR 2024]
**ViLA: Visual Language Model**

**More in context learning examples**

<table>
<thead>
<tr>
<th>Input few shot examples + target image</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Underground image" /> Underground. <img src="image2" alt="Congress image" /> Congress.</td>
<td><img src="image3" alt="Image3" /> <strong>LLaVA-1.5: Soulomes.</strong> <strong>Ours:</strong> Soulomes.</td>
</tr>
<tr>
<td><img src="image4" alt="Image4" /> 2+1=3 <img src="image5" alt="Image5" /> 5+6=11 <img src="image6" alt="Image6" /> 3x6</td>
<td><img src="image7" alt="Image7" /> <strong>LLaVA-1.5: 3x6=18</strong> <strong>Ours:</strong> 3x6=18</td>
</tr>
<tr>
<td><img src="image8" alt="Violin image" /> Les sanglots longs des violons de l’automne blessent mon cœur d’une langueur monotone. <img src="image9" alt="Snake image" /> Pour qui sont ces serpents qui sifflent sur vos têtes?</td>
<td><img src="image10" alt="Image10" /> <strong>LLaVA-1.5: Les serpents <em>(snake)</em> qui sifflent sur vos têtes sont-ils vos amis?</strong> <strong>Ours:</strong> Pourquoi ces flamands <em>(flamingo)</em> roses s’embrassent-ils?</td>
</tr>
<tr>
<td><img src="image11" alt="Romanticism image" /> Romanticism <img src="image12" alt="Surrealism image" /> Surrealism</td>
<td><img src="image13" alt="Image13" /> <strong>LLaVA-1.5: Surrealism</strong> <strong>Ours:</strong> Impressionism</td>
</tr>
</tbody>
</table>

VILA: On Pre-training for Visual Language Models [Lin et al., CVPR 2024]
Q: If you are driving, should you honk at the pedestrians crossing the road in this case? Why?
A: In the image, a car is driving down a street with a group of people crossing the road. It is **not appropriate to honk** at pedestrians crossing the road, as it can be considered **impolite and disrespectful**. Instead, the driver should wait patiently for the pedestrians to cross safely before proceeding. Honking can cause distress and discomfort to the pedestrians and may also lead to accidents if the pedestrians are not aware of the approaching vehicle. It is essential to prioritize the safety and well-being of pedestrians in such situations.

Q: What color is the traffic light in the image?
A: The traffic light in the image is red.

Q: What is the person in the center of the image doing?
A: The person in the center of the image is pushing a stroller with a baby in it.
Q: Is the driver distracted?
VILA: Yes, the driver is distracted, as he is looking at his cell phone while driving. This is a dangerous behavior that can lead to accidents and endanger the lives of the driver, passengers, and other road users.

Q: Is the driver on the phone?
VILA: Yes, the driver is on the phone while sitting in the driver's seat.

Q: How many people are in the car?
VILA: There are two people in the car.

Q: Where is the passenger sitting?
VILA: The passenger is sitting in the back seat of the car.
VILA for UAV monitoring

Zero-shot learning: UAV captured image explaining

User: Do facilities here look normal or need maintenance?

VILA: The facilities in the image, which are wind turbines, appear to be in good condition and functioning properly. They are situated in a large field with a clear blue sky, indicating that they are well-maintained and not in need of immediate maintenance. The presence of multiple wind turbines in the field suggests that they are part of a larger wind farm, which is likely to be managed and maintained by professionals to ensure optimal performance and efficiency.
Zero-shot learning: patient situation understanding

**User:** What is the patient status?

**VILA:** The patient is in a critical condition, as indicated by the presence of a ventilator and the fact that she is hooked up to a lot of wires.

**User:** Is the patient asleep?

**VILA:** Yes, the patient is asleep in the hospital bed.
Zero-shot learning: training session understanding

**User:** What is the training session about?

**VILA:** The training session is about learning how to perform CPR (cardiopulmonary resuscitation) on a mannequin. The group of people, including nurses and possibly other medical professionals, are gathered around a mannequin to practice their skills and learn the proper techniques for administering CPR.

**User:** Will the practice cause pain to the patient in the image?

**VILA:** The image shows a group of people, including nurses and a mannequin, practicing CPR on a mannequin. While practicing CPR is essential for healthcare professionals, it is important to note that it is a simulation and not a real patient. The practice does not cause pain to the patient in the image, as it is a mannequin and not a living person.
User: How many chip bags are picked by robot arm:

VILA: Two chip bags are picked by robot arm

User: The 8 input images are sampled at 2 frames per second. How long does it take for the robot arm to pick up one chip bag?

VILA: The robot arm takes 1.5 seconds to pick up the chip bag.
User: Is the building in healthy condition?

VILA: The building appears to be in a state of disrepair, with visible cracks and chips in the paint. This suggests that the building may be old, poorly maintained, or has experienced some form of damage or wear over time. The presence of these structural issues could potentially pose safety risks or negatively impact the building’s overall appearance and value.

User: What are the likely service needed for this building?

VILA: The building in the image has a cracked wall and a damaged roof, which indicates that it requires repair and maintenance services. The cracked wall may require repairs to ensure structural integrity and prevent further damage. The damaged roof may need to be replaced or repaired to prevent water leakage and potential damage to the building’s interior. Additionally, the building may need to be inspected for any other potential issues that may arise due to the damage.
LLM Compression: Lower-bit Quantization
Quantization can reduce deployment costs
Quantization lowers the bit-width and improves efficiency

- 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: W8A8 Quantization for Cloud

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models

**Goal:** Quantize LLM to lower precision, both activation and weight

**Challenge:** activation channels have many outliers, wasting the dynamic range (many channels became zero)

**Our Solution:** Smooth the activations: 100*1= 10*10; Equalize the quantization difficult from activation to weights.

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models (Xiao et al., ICML 2023)
W4A16 for Single Batch Serving

W8A8 cannot address low computational intensity of decoding

- W8A8 quantization improves arithmetic efficiency and memory efficiency by \(2x\) compared to FP16. Is it enough?
- But single-query LLM inference (e.g., local) is still highly memory-bounded.
- We need **low-bit weight-only quantization** (e.g., W4A16)

- LLaMA-65B GEMV \([1, 8192] \times [8192, 8192]\)
- NVIDIA A100 GPU 80GB: 312TFLOPS (int8), 2000GB/s
- Computational intensity: \[
\frac{\text{FLOP}}{\text{Byte}} \quad \frac{8192^2}{8192^2} < \frac{312}{2000} \times 10^3
\]
- Highly **memory-bounded** (~10^2 gap)!
**AWQ for On-Device LLM**

**AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration**

**Goal:** deploy LLM on the edge: Jetson Orin, AI PC

**Challenge:** weight memory bounded @low batch size; can’t fit; idle ALU.

**Our Solution:** 4bit weights, fp16 activation, fp16 arithmetic.

Activation-awareness: preserve the salient weight channel by scaling according to the activation magnitude.
**AWQ: Activation-aware Weight Quantization**

Targeting group-wise low-bit weight-only quantization (W4A16)

<table>
<thead>
<tr>
<th>$W_{\text{FP16}}$</th>
<th>$Q(W)_{\text{INT3}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1.2 -0.2 -2.4 -3.4</td>
<td>+1 +0 -2 -3</td>
</tr>
<tr>
<td>-2.5 -3.5 +1.9 +1.4</td>
<td>-3 -4 +2 +1</td>
</tr>
<tr>
<td>-0.9 +1.6 -2.5 -1.9</td>
<td>-1 +2 -3 -2</td>
</tr>
<tr>
<td>-3.5 +1.5 +0.5 -0.1</td>
<td>-4 +2 +1 +0</td>
</tr>
<tr>
<td>+1.8 -1.6 -3.2 -3.4</td>
<td>+2 -2 -3 -3</td>
</tr>
<tr>
<td>+2.4 -3.5 -2.8 -3.9</td>
<td>+2 -4 -3 -4</td>
</tr>
<tr>
<td>+0.1 -3.8 +2.4 +3.4</td>
<td>+0 -4 +2 +3</td>
</tr>
<tr>
<td>+0.9 +3.3 -1.9 -2.3</td>
<td>+1 +3 -2 -2</td>
</tr>
</tbody>
</table>

- 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.
AWQ: Activation-aware Weight Quantization

Targeting group-wise low-bit weight-only quantization (W4A16)

- 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., MLSys 2024]
AWQ: Activation-aware Weight 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.

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
**AWQ: Activation-aware Weight 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

- 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?
- **This is not the truth!**

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
AWQ for Low-bit Weight-only Quantization

Salient weights are determined by activation distribution, not weight

- But how do we select salient channels? Should we select based on weight magnitude?
- No! We should look for **activation distribution, but not weight!** (Activation has outliers!)
- However, 1% FP16 is not hardware-friendly.
AWQ for Low-bit Weight-only Quantization

Protecting salient weights by scaling (no mixed prec.)

\[ WX \rightarrow Q(W \cdot s)(s^{-1} \cdot X) \]

- We need to consider **activation-awareness** for salient channels.
- We solve for best hyper-parameters with a simple grid search.

**OPT-6.7B Wiki-2 PPL↓**

- fuse to previous op
- protects salient
- destroy non-salient

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>FP16</th>
<th>RTN</th>
<th>1.5x</th>
<th>2x</th>
<th>4x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.29</td>
<td>43.16</td>
<td>14.49</td>
<td>14.07</td>
<td>14.42</td>
</tr>
</tbody>
</table>

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]

Song Han: Visual Language Models for Edge AI 2.0
AWQ for Low-bit Weight-only Quantization

Protecting salient weights by scaling (no mixed prec.)

\[ WX \rightarrow Q(W \cdot s)(s^{-1} \cdot X) \]

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

\[ s = s_x^\alpha, \quad \alpha^* = \arg\min_{\alpha} \mathcal{L}(s_x^{\alpha}) \]

- We need to consider **activation-awareness** for salient channels.
- We solve for best hyper-parameters with a simple grid search.

**AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration** [Lin et al., MLSys 2024]
### AWQ Results

**Improving general LLM quantization (LLaMA & OPT)**

<table>
<thead>
<tr>
<th>LLaMA Family</th>
<th>MMLU (5-shot) average ↑</th>
<th>Common Sense QA (0-shot) average ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7B</td>
<td>13B</td>
</tr>
<tr>
<td><strong>FP16</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>38.41%</td>
<td>45.21%</td>
</tr>
<tr>
<td><strong>INT3 g128</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>33.43%</td>
<td>39.20%</td>
</tr>
<tr>
<td>GPTQ</td>
<td>30.53%</td>
<td>40.90%</td>
</tr>
<tr>
<td><strong>AWQ</strong></td>
<td>35.43%</td>
<td>41.84%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPT / PPL↓</th>
<th>125M</th>
<th>1.3B</th>
<th>2.7B</th>
<th>6.7B</th>
<th>13B</th>
<th>30B</th>
<th>66B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FP16</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>31.95</td>
<td>16.41</td>
<td>14.32</td>
<td>12.29</td>
<td>11.5</td>
<td>10.67</td>
<td>10.09</td>
</tr>
<tr>
<td><strong>INT3 g128</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>58.49</td>
<td>206.54</td>
<td>595.28</td>
<td>43.16</td>
<td>45.37</td>
<td>28.84</td>
<td>423.39</td>
</tr>
<tr>
<td>GPTQ</td>
<td>41.93</td>
<td>18.53</td>
<td>15.79</td>
<td>13.13</td>
<td>12.01</td>
<td>11.00</td>
<td>11.48</td>
</tr>
<tr>
<td>AWQ</td>
<td>41.10</td>
<td>18.53</td>
<td>15.62</td>
<td>12.99</td>
<td>12.03</td>
<td>11.03</td>
<td>10.46</td>
</tr>
<tr>
<td><strong>INT4 g128</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>35.51</td>
<td>17.70</td>
<td>15.12</td>
<td>13.02</td>
<td>11.89</td>
<td>11.00</td>
<td>10.44</td>
</tr>
<tr>
<td>GPTQ</td>
<td>34.23</td>
<td>16.92</td>
<td>14.69</td>
<td>12.51</td>
<td>11.60</td>
<td>10.74</td>
<td>10.24</td>
</tr>
<tr>
<td>AWQ</td>
<td>33.96</td>
<td>16.85</td>
<td>14.61</td>
<td>12.44</td>
<td>11.60</td>
<td>10.75</td>
<td>10.16</td>
</tr>
</tbody>
</table>

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
**AWQ Results**

Quantization of multi-modal LMs (OpenFlamingo, captioning)

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

- Improved quantized performance for both 4-bit and 3-bit quantization.
- Big improvement even under **4-bit** quantization (not very noticeable with QA benchmarks)
### AWQ Results

**Quantization of instruction-tuned models**

<table>
<thead>
<tr>
<th></th>
<th>INT3/g128</th>
<th>RTN</th>
<th>0</th>
<th>40</th>
<th>80</th>
<th>120</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWQ</td>
<td>28</td>
<td>13</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPTQ</td>
<td>48</td>
<td>148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>28</td>
<td>13</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Comparing quantized Vicuna with FP16 counterparts
- Test under both orderings (quantized vs FP16, FP16 vs quantized) to get rid of ordering bias

---

**AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration** [Lin et al., MLSys 2024]
**AWQ Results**

Quantization of multi-modal LMs (OpenFlamingo, captioning)

- Improved quantized performance for both 4-bit and 3-bit quantization.
- Big improvement even under **4-bit** quantization (not very noticeable with QA benchmarks)

**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 Results**

Quantization of multi-modal LMs (LLaVA, visual reasoning)

**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 Results**

Quantization of multi-modal LMs (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.
SmoothQuant and AWQ Impacts

TensorRT-LLM (NVIDIA)
- https://github.com/NVIDIA/TensorRT-LLM#key-features
- https://huggingface.co/docs/transformers/main_classes/quantization

Granite (IBM)
- IBM’s internal code model, Granite, utilizes AWQ for quantization.
- https://github.com/lm-sys/FastChat/blob/main/docs/awq.md

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

Imdeploy (InternLM)

Replicate

Granite (IBM’s internal code model)

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
TinyChat: Efficient LLMs Inference Engine
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
  - TensorRT-LLM: high efficiency, but harder to use
- TinyChat: efficient, lightweight, Python-native (composable with other stacks like vLLM)

Graph showing:
- TensorRT-LLM
- TinyChat
- HuggingFace

Efficiency vs. Ease of use

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
TinyChat: A Lightweight Serving Infra

Supporting a wide range of models on NVIDIA Jetson Orin

- 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**.

Latency comparison on Jetson Orin (64G) mobile GPU

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
TinyChat seamlessly supports VLMs

Accelerating visual-language models across different GPU platforms

- TinyChat also seamlessly supports VILA, delivering ~3x speedup over FP16 on Orin and allows interactive VLM deployment on the edge (laptops and AIoT).

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>A100 Tok/sec</th>
<th>4090 Tok / sec</th>
<th>Orin Tok / sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>VILA-7B</td>
<td>FP16</td>
<td>81.6</td>
<td>58.5</td>
<td>11.5</td>
</tr>
<tr>
<td>VILA-7B-AWQ</td>
<td>INT4</td>
<td>155.3</td>
<td>168.1</td>
<td>35.6</td>
</tr>
<tr>
<td>VILA-13B</td>
<td>FP16</td>
<td>48.5</td>
<td>OOM</td>
<td>6.1</td>
</tr>
<tr>
<td>VILA-13B-AWQ</td>
<td>INT4</td>
<td>102.1</td>
<td>99.0</td>
<td>17.5</td>
</tr>
</tbody>
</table>

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
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
TinyChat: A Lightweight Serving Infra

Demo on TinyChatComputer, powered by NVIDIA Jetson Orin Nano

• On a GPU board with just ~7G available memory, TinyChat enables efficient deployment of 7B large language models, thanks to AWQ quantization.

• Worked with students from Harvard Graduate School of Design to manufacture a physical TinyChatComputer demo.

https://github.com/mit-han-lab/llm-awq
TinyChat: A Lightweight Serving Infra

TinyChat seamlessly supports personal laptops with Intel / ARM CPUs

AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [Lin et al., MLSys 2024]
TinyChat: Efficient framework for LLM deployment

VILA: Multi-modal capability for LLMs

AWQ: LLM quantization, 4x weight reduction

Edge AI 2.0: Multi-model LM on the edge!
TinyChat — Visual language models (VILA)

Efficient image reasoning on Jetson Orin: TinyChat w/ VILA model family
TinyChat for visual language model

Single image for multi-round Q&A: A driving scenario

VILA-13B + AWQ: 100 tokens/s on 4090
TinyChat for visual language model

In context learning with multiple image inputs

VILA-13B + AWQ: 84 tokens/s (3 image inputs) on RTX 4090
TinyChat for visual language model

Multi-round Q&A with multi-image inputs

VILA-13B + AWQ: 83 tokens/s (3 image inputs) on RTX 4090
Run visual language models on personal laptops

VILA-7B + AWQ: Running on MacBook Arm CPU
Try out our online demo for VILA models!

https://vila.hanlab.ai/
Summary

Edge AI 2.0 Requires Full-Stack Optimization

Application
(Demand for Computation)

Model Compression
(Bridging the gap between demand and supply for computation)

System and Hardware
(Supply of Computation)