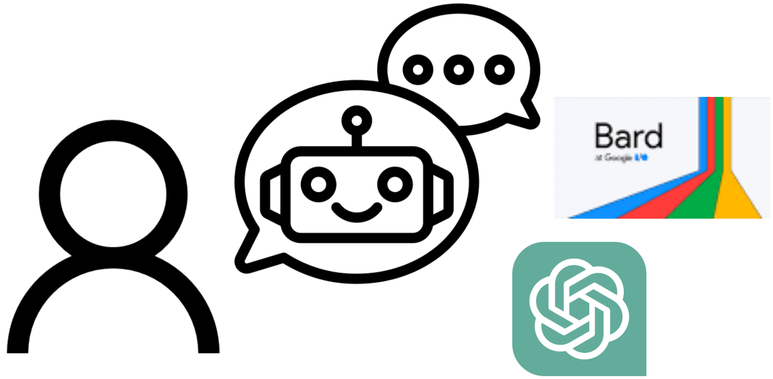
MIT HAN LAB

TinyChatEngine

Wei-Ming Chen

Deployment of Large Language Models

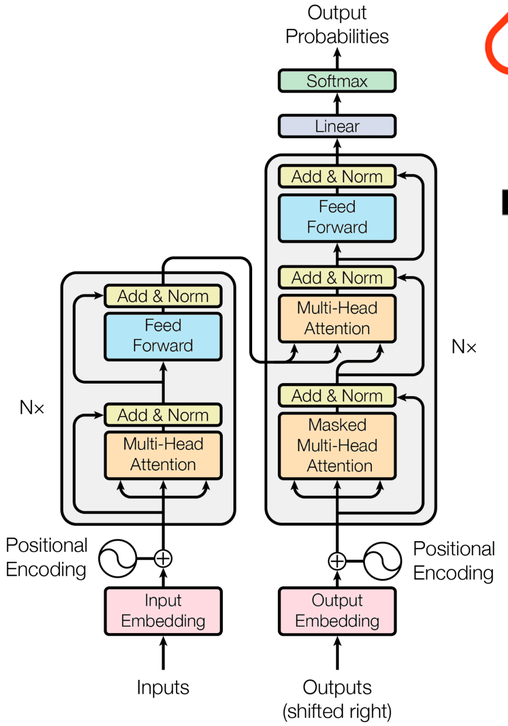
- Transformer-based large language models power impactful applications.



Online Chatbot

GitHub Copilot

Software Development



PyTorch

NVIDIA

Cloud service

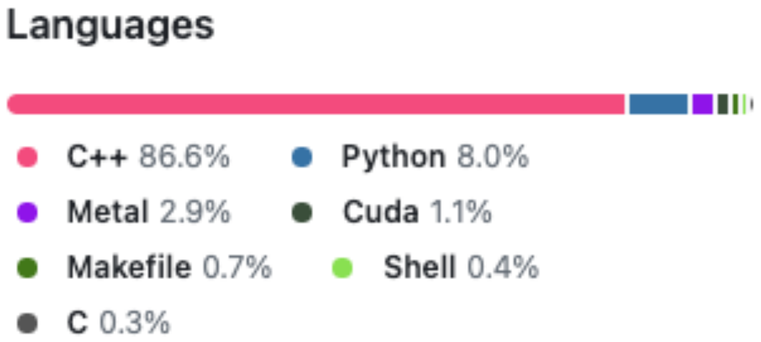
?

Edge devices

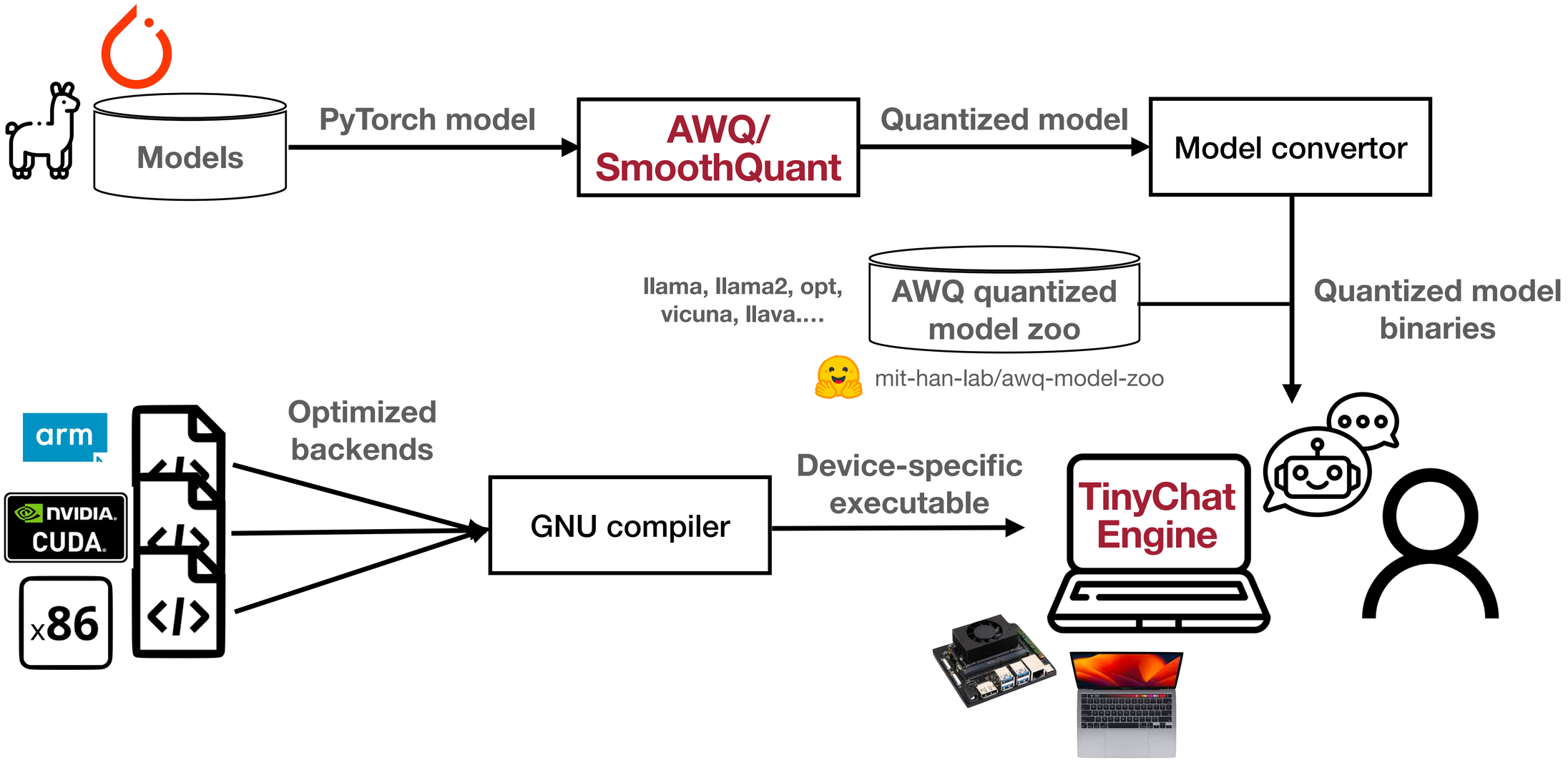
Introduction

- TinyChatEngine: An inference library designed for the efficient deployment of quantized large language models (LLMs) on edge devices.
  - Universal framework
  - No dependency on bulky libraries
  - High performance
  - Easy to use

- TinyChatEngine consists of:
  - Pure C/C++ implementation
    - Control flow for LLM runtime
    - Various device-specific kernels
  - Python toolchains for model conversion and quantization



Workflow of Deploying a Model with TinyChatEngine



Flexible Backends and Quantization Methods

- Flexibility to support diverse quantization techniques:
  - Per-tensor int8 quantization: e.g., w8a8 (SmoothQuant).
  - Group-wise int4 weight quantization: e.g., w4a16/w4a32 (AWQ).
- Offline weight reordering for different devices to reduce weight decoding overhead.
- Unleash the potentials of devices with different ISA and accelerators:
  - CPU:
    - ARM: NEON intrinsics.
    - x86-64: SSE, AVX, and AVX2.
  - GPU:
    - Nvidia: CUDA.
    - Apple: Metal.
- Support multiple platforms
  - MacOS
  - Linux
  - Windows

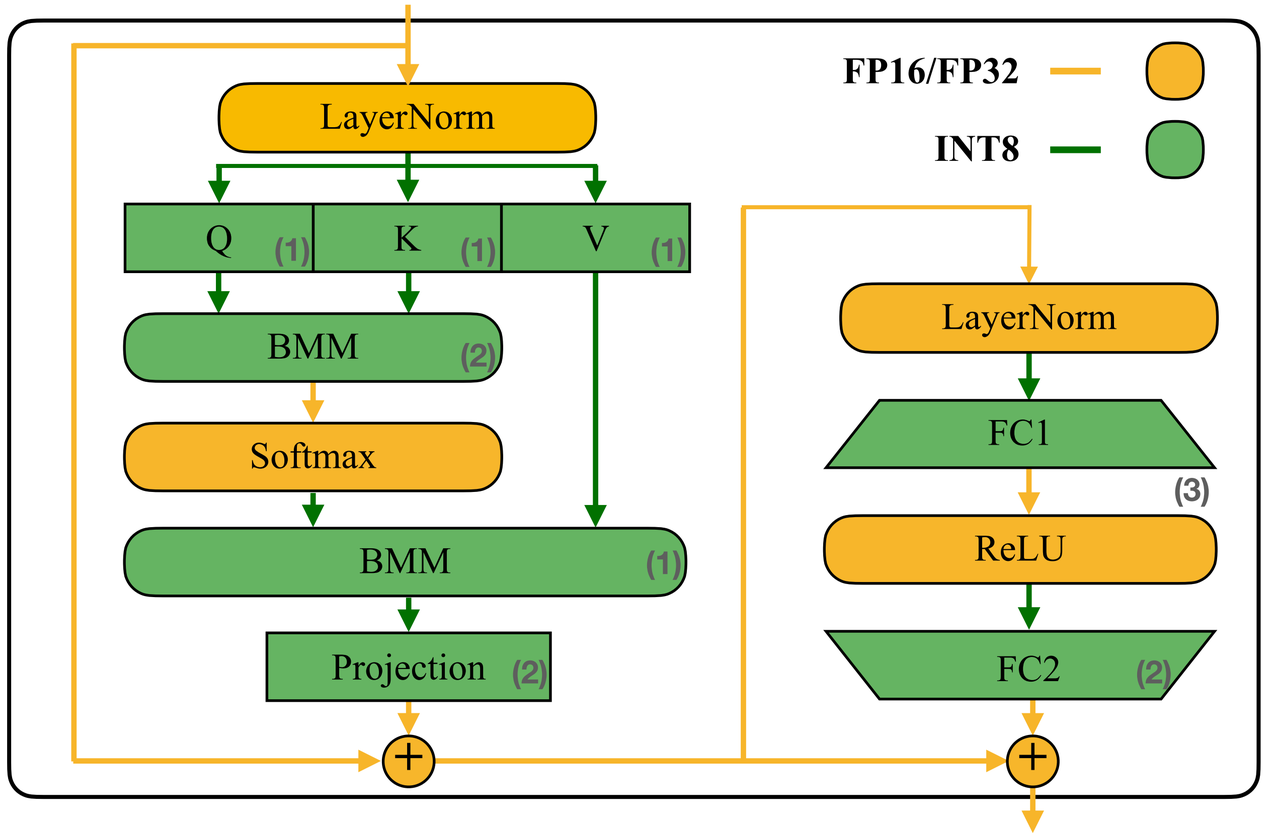



W8A8 Quantization Flow (SmoothQuant)

- Activation are stored as FP32 by default unless the device natively supports FP16.

- Per-tensor quantization flow for w8a8
  - Keep input and output of a block as FP16/FP32.
  - Quantized floating-point activation at **runtime.**
  - Use int8 operations for most operators.

- Specialized kernels for different output precisions and activation function.
  1. W8A8O8Linear
  2. W8A8OFP32Linear
  3. W8A8OFP32LinearReLU

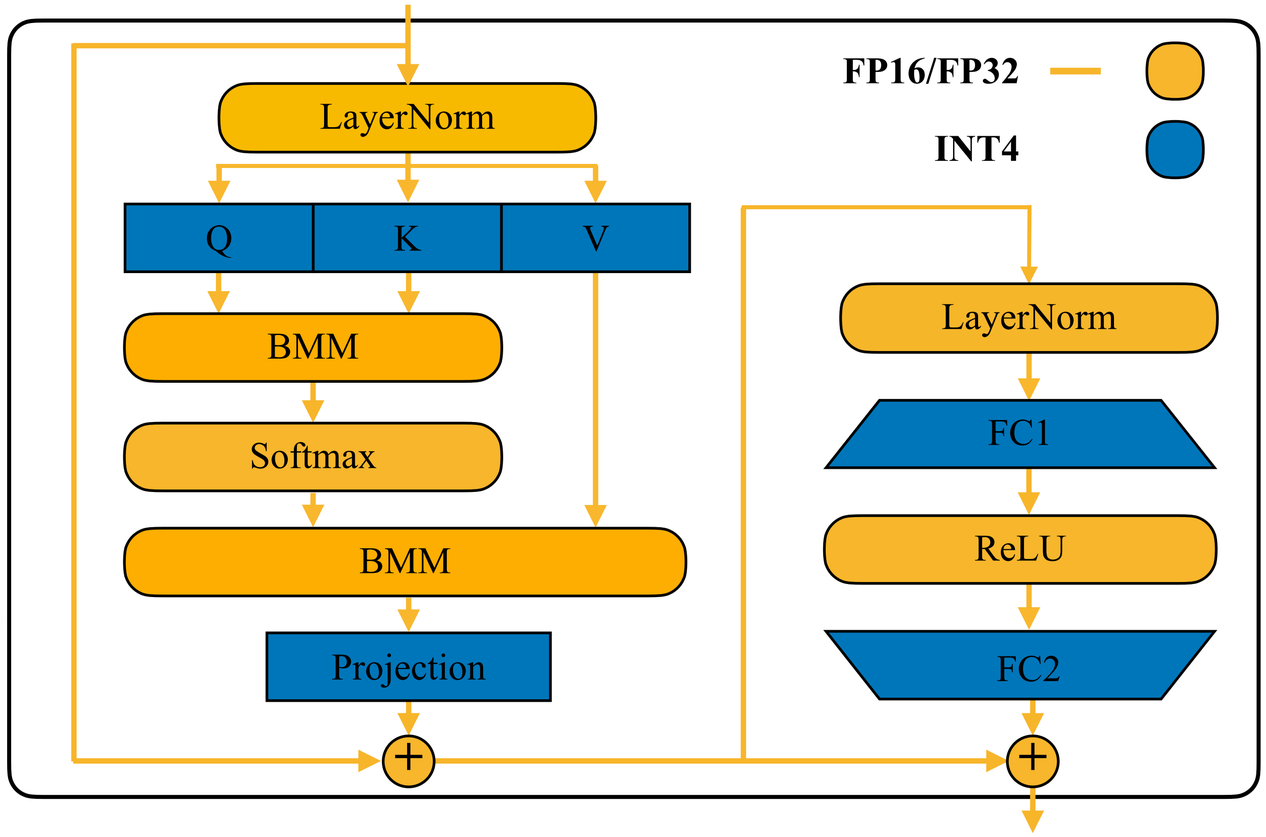


Precision mapping for a Transformer block

W4A16/W4A32 Quantization Flow (AWQ)

- Group-wise quantization flow for w4a16/ w4a32
  - Only apply low-bit computation for compute-intensive linear layers
  - Keep input and output of each operator as FP16/FP32.

- High performance int4 linear kernels specialized for CPU and GPU
  - CPU: Runtime activation quantization to utilize int8 SIMD intrinsics
  - GPU: Utilize CUDA tensor core to accelerate W4A16

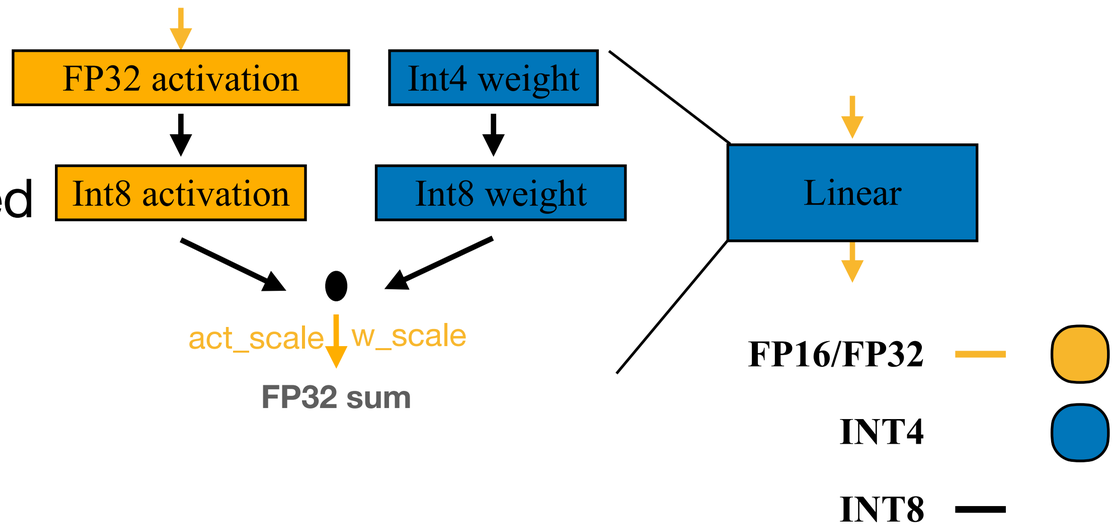


Precision mapping for a Transformer block

High Performance int4 Linear Operator

ARM and x86 CPU

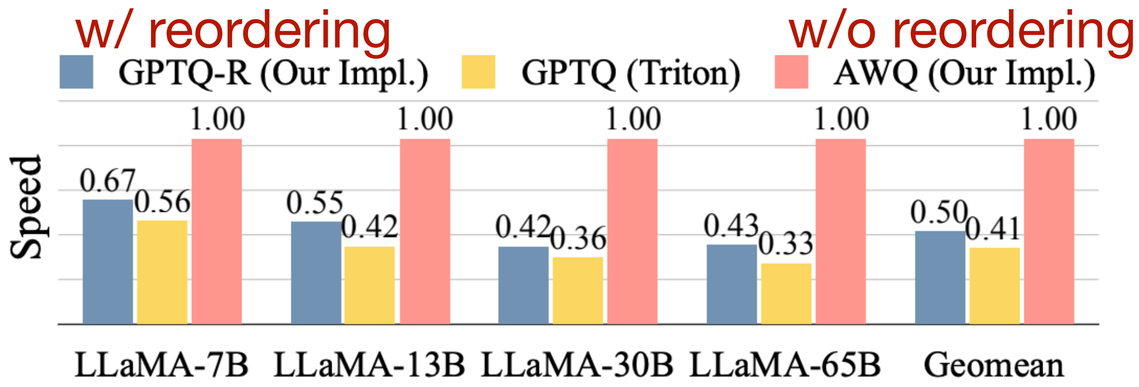
- CPUs typically lack native FP16 support, but FP32 Multiply–accumulate (MAC) operation is slow
  - Low parallelism compared to int8 operations 4x slower!
  - De-quantizing 4-bit weights into FP32 introduce high overheads even with SIMD intrinsics:
    - 32 int4 (128bit) -> 32 int8 (256bit)
    - 32 int8 (256bit) -> 2 x16 int16 (2 x 256 bit)
    - 2 x 16 Int16 (2 x 256 bit) -> 4 x 8 FP32 (4 x 256 bit) Expanding data footprint by 8x!
- Leverage int8 SIMD MAC operations
  - Group-wise quantization of activation into int8.
  - Expansion of int4 weights to int8.
  - Invocation of int8 SIMD MAC operation.
  - Significant performance improvement compared to FP32 operations
    - 1.3x speed up on Intel i7-9750H
    - 3x speed up on M1 Pro



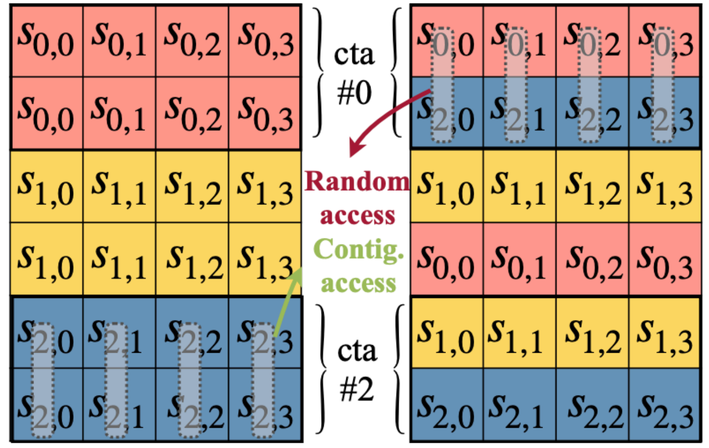
High Performance int4 Linear Operator

W4A16 CUDA Kernel

- Instead of Matrix-Vector product, we implement it in Matrix-Matrix product to utilize **tensor cores**
  - Support diverse workloads (context, generation, batch sizes)
  - Outperform GPTQ-Triton implementation
- Eliminate runtime reordering to achieve optimal performance
  - Unlike GPTQ which requires runtime weight reordering



Reordering elimination brings 2x speedup to W4A16

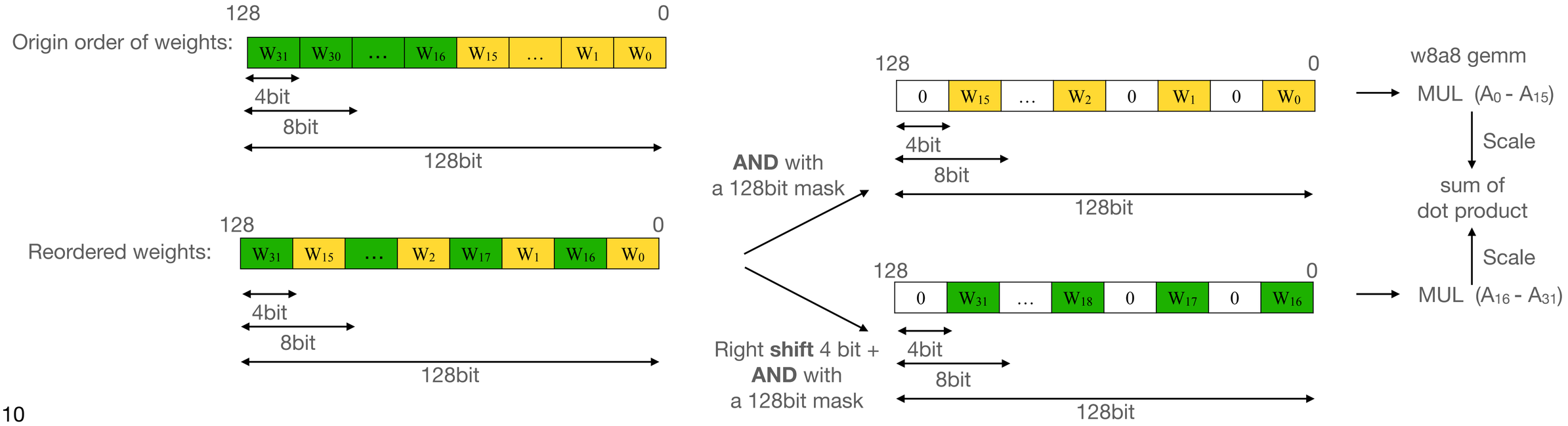


Device-specific Weight Reordering

- Weights will be reordered during model conversion to suit the operation bit width/kernel implementation on the device.
  - Weights need to be expand to 8bit to utilize int8 SIMD operations with ***shift*** and ***AND*** operations
  - Eliminating runtime reordering to improve performance.

NEON: 128 bit
AVX2: 256 bit
AVX512: 512 bit

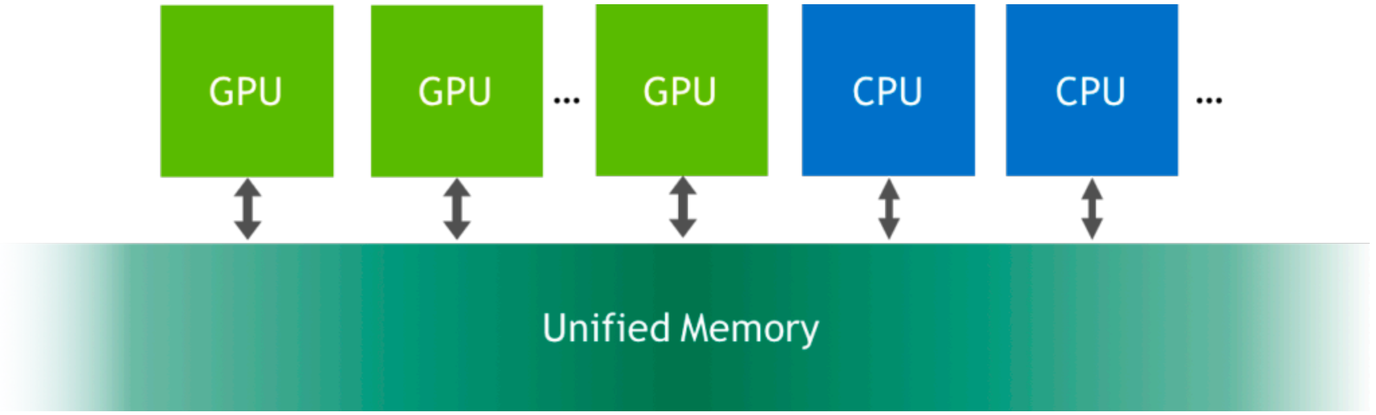
Example of weight decoding with ARM NEON (128 bit-width SIMD)



Efficient Memory Management

- Pre-allocate runtime buffers during initialization
  - Reuse memory buffer to reduce memory footprint
  - Eliminate the overheads to allocate/deallocate memory buffer

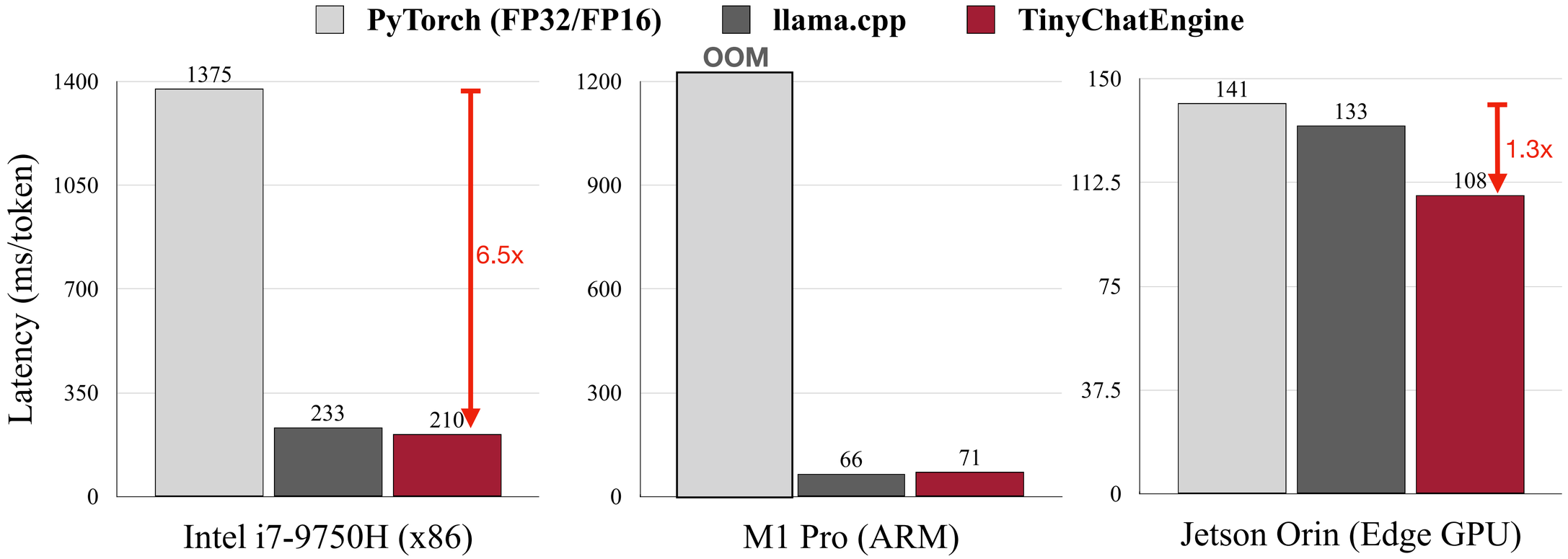
- Utilize unified memory that is available on edge GPU to reduce peak memory (e.g., Jetson Orin and Apple GPU)
  - Unlike other libraries like PyTorch that cause model duplication



Reference: Unified Memory for CUDA Beginners [Link]

Efficient LLM Deployment on Edge Devices

- TinyChatEngine achieves fast text generation for LLaMA2-7B on various devices



Demo: Deploy LLaMA2 Chatbot with TinyChatEngine

- We provide ready to use quantized model which can be easily deployed.
  - Download the source code

```
$ git clone --recursive https://github.com/mit-han-lab/TinyChatEngine.git
```

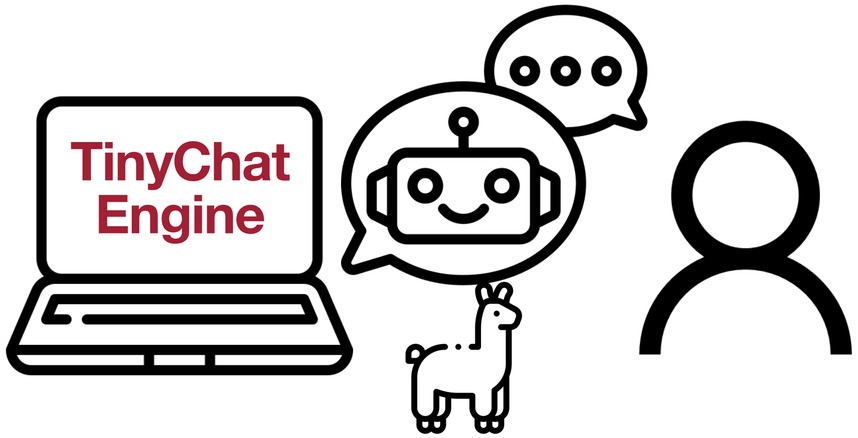
  - Download a quantized model from our model zoo

```
$ python download_model.py --model LLaMA_7B_2_chat --QM QM_ARM
```

  - Compile the program and start the chat

```
$ make chat -j
$ ./chat
Using model: LLaMA7B_2_chat
Using LLaMA's default data format: INT4
Loading model... Finished!
USER: Write a syllabus for Operating Systems.
ASSISTANT:
Of course! Here is a sample syllabus for a …
```

```
(base) wmchen@Wei-Mings-MacBook-Pro transformer % ./chat
Using model: LLaMA_7B_2_chat
Using LLaMA's default data format: INT4
Loading model... Finished!
USER: Write a poem about MIT.
ASSISTANT:
Sure, I'd be happy to help you write a poem about MIT! Here is one possible draft:
In Cambridge, Massachusetts, where the brightest minds meet,
There stands a university of excellence and great feat.
MIT - A place of innovation, creativity, and might,
Where knowledge and technology come together in delightful sight.
From physics to computer science, engineering to math,
The students here are pushing boundaries with each new batch.
With labs and libraries, professors who inspire and guide,
MIT is a place where dreams take flight and soar inside.
From the Great Dome to Building 10,
This institution's legacy will forever be etched in stone.
So here's to MIT - A shining star of higher learning,
Where minds are expanded and futures are reborn!
```



Thank you
Q & A