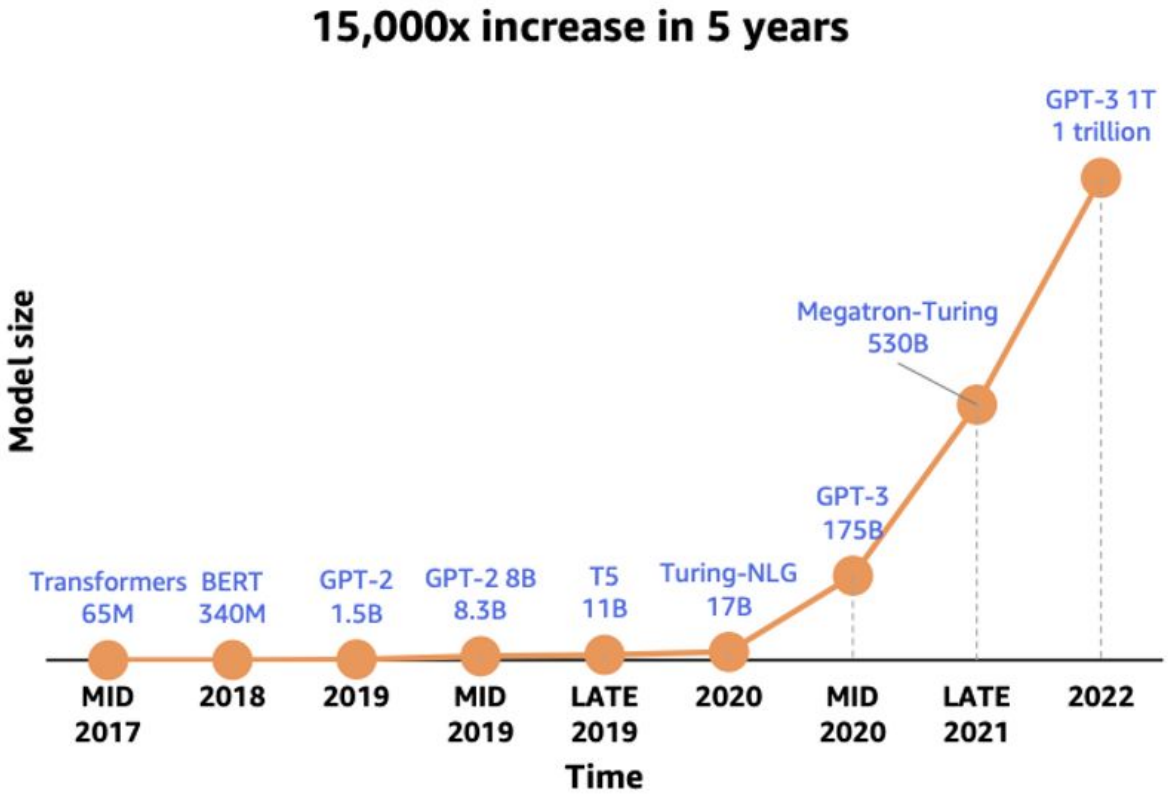
LLM Quantization

Motivation

While a larger model offers more capabilities, it also demands more expensive hardware and greater hardware resources in general.

Solutions: Model Distillation, Quantization, etc



What is Quantization?

- Quantization = mapping continuous infinite values to a smaller set of discrete finite values
- In the context of LLMs, it refers to the process of converting the weights of the model **from higher precision data types to lower-precision ones**.

Floating Point Formats

bfloat16: Brain Floating Point Format

Range: ~1e^{-38} to ~3e^{38}

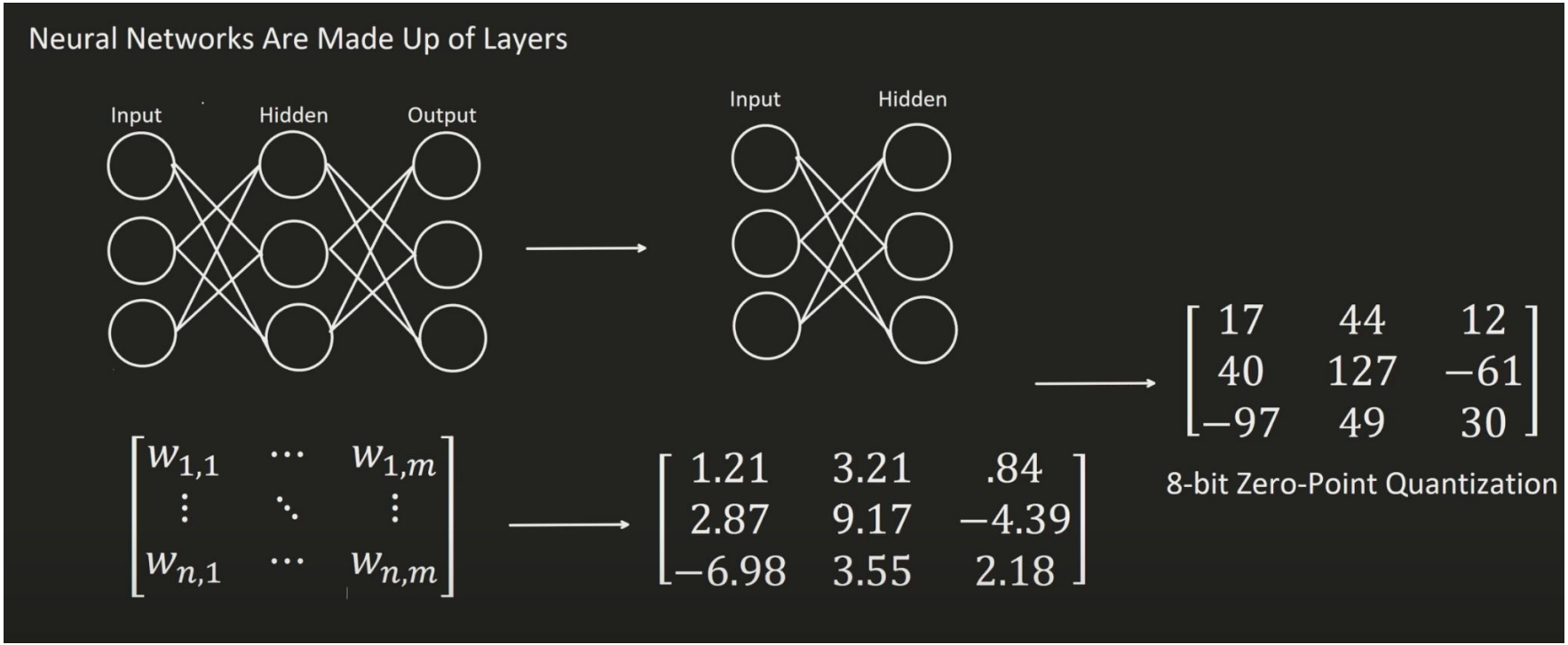
fp32: Single-precision IEEE Floating Point Format

Range: ~1e^{-38} to ~3e^{38}

fp16: Half-precision IEEE Floating Point Format

Range: ~5.96e^{-8} to 65504

32 bits float -> 8 bits int

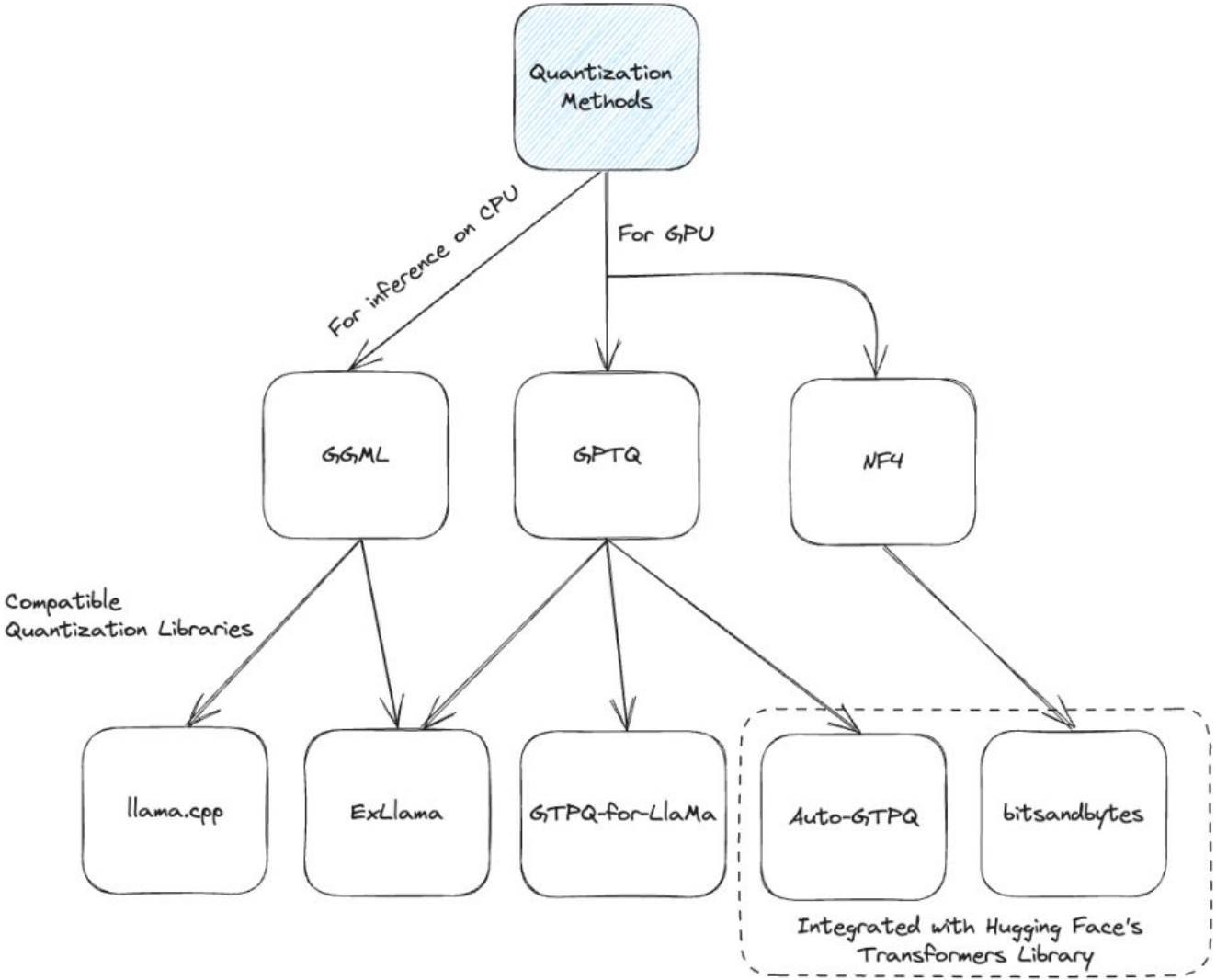


The Two Types of LLM Quantization

Post-Training Quantization (PTQ): converting the weights of an already trained model to a lower precision without any retraining. Though straightforward and easy to implement, PTQ might degrade the model's performance slightly due to the loss of precision in the value of the weights.

Quantization-Aware Training (QAT): Unlike PTQ, QAT integrates the weight conversion process during the training stage. This often results in superior model performance, but it's more computationally demanding. A highly used QAT technique is the QLoRA.

Noteworthy Techniques in Post-Training Quantization



Larger Quantized Model vs Smaller non-Quantized

- Reducing the precision will reduce the accuracy of the model
- Should you prefer smaller full-precision model or a larger quantized model with a comparable inference cost?
- Meta researchers have demonstrated that in some cases:
  - When comparing models with similar inference costs, the larger quantized models can outperform their smaller, non-quantized counterparts.
  - This advantage becomes even more pronounced with larger networks, as they exhibit a smaller quality loss when quantized.
  - https://arxiv.org/abs/2305.17888

The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits

- Based on another paper: BitNet: Scaling 1-bit Transformers for Large Language Models https://arxiv.org/pdf/2310.11453.pdf
- Matches the full-precision (i.e., FP16 or BF16) Transformer LLM with the same model size and training tokens in terms of both perplexity and end task performance
- While being significantly more cost-effective in terms of latency, memory, throughput, and energy consumption.

$\log_2 3 = 1.58$ bit

- every single parameter/weight is ternary {-1, 0, 1}

Quantization Function. To constrain the weights to -1, 0, or +1, we adopt an *absmean* quantization function. It first scales the weight matrix by its average absolute value, and then round each value to the nearest integer among {-1, 0, +1}:

$$\widetilde{W} = \text{RoundClip}(\frac{W}{\gamma + \epsilon}, -1, 1), \tag{1}$$

$$\text{RoundClip}(x, a, b) = \max(a, \min(b, \text{round}(x))), \tag{2}$$

$$\gamma = \frac{1}{nm} \sum_{ij} |W_{ij}|. \tag{3}$$

Models	Size	Memory (GB)↓	Latency (ms)↓	PPL↓
LLaMA LLM	700M	2.08 (1.00x)	1.18 (1.00x)	12.33
BitNet b1.58	700M	0.80 (2.60x)	0.96 (1.23x)	12.87
LLaMA LLM	1.3B	3.34 (1.00x)	1.62 (1.00x)	11.25
BitNet b1.58	1.3B	1.14 (2.93x)	0.97 (1.67x)	11.29
LLaMA LLM	3B	7.89 (1.00x)	5.07 (1.00x)	10.04
BitNet b1.58	3B	**2.22 (3.55x)**	**1.87 (2.71x)**	**9.91**
BitNet b1.58	3.9B	**2.38 (3.32x)**	**2.11 (2.40x)**	**9.62**

Table 1: Perplexity as well as the cost of BitNet b1.58 and LLaMA LLM.

Models	Size	ARCe	ARCc	HS	BQ	OQ	PQ	WGe	Avg.
LLaMA LLM	700M	54.7	23.0	37.0	60.0	20.2	68.9	54.8	45.5
BitNet b1.58	700M	51.8	21.4	35.1	58.2	20.0	68.1	55.2	44.3
LLaMA LLM	1.3B	56.9	23.5	38.5	59.1	21.6	70.0	53.9	46.2
BitNet b1.58	1.3B	54.9	24.2	37.7	56.7	19.6	68.8	55.8	45.4
LLaMA LLM	3B	62.1	25.6	43.3	61.8	24.6	72.1	58.2	49.7
BitNet b1.58	3B	**61.4**	**28.3**	**42.9**	**61.5**	**26.6**	**71.5**	**59.3**	**50.2**
BitNet b1.58	3.9B	**64.2**	**28.7**	**44.2**	**63.5**	**24.2**	**73.2**	**60.5**	**51.2**

Table 2: Zero-shot accuracy of BitNet b1.58 and LLaMA LLM on the end tasks.

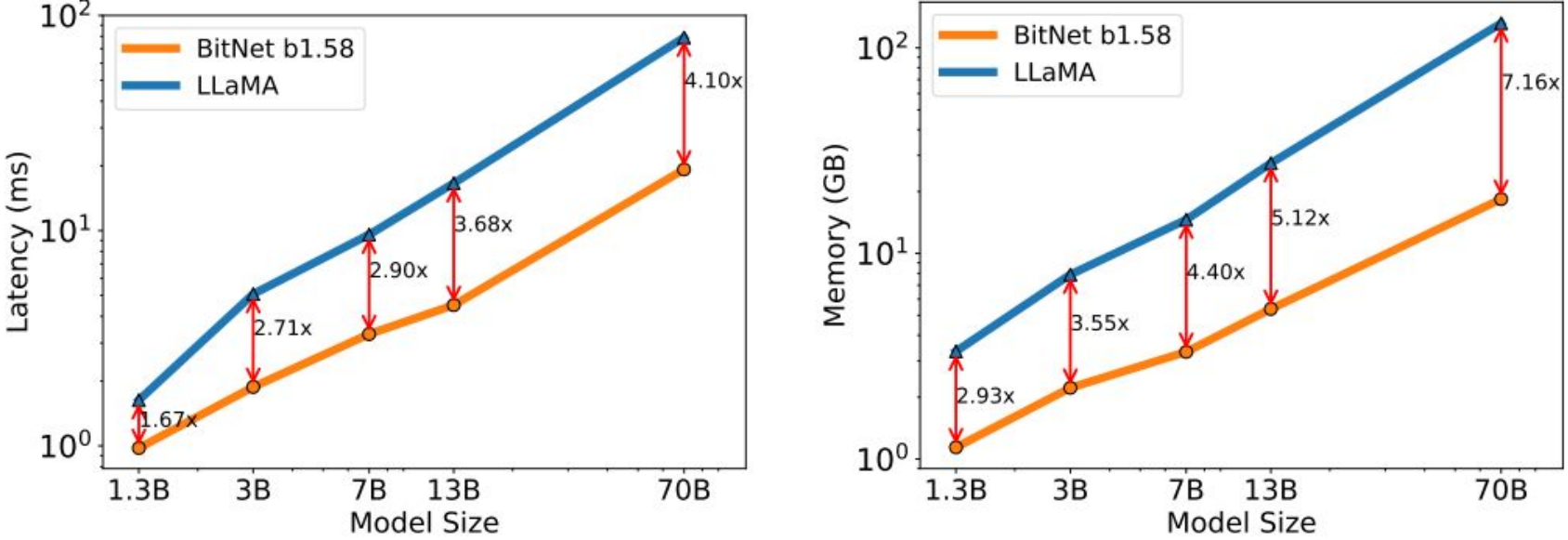


Figure 2: Decoding latency (Left) and memory consumption (Right) of BitNet b1.58 varying the model size.

Models	Size	Max Batch Size	Throughput (tokens/s)
LLaMA LLM	70B	16 (1.0x)	333 (1.0x)
BitNet b1.58	70B	**176 (11.0x)**	**2977 (8.9x)**

Table 3: Comparison of the throughput between BitNet b1.58 70B and LLaMA LLM 70B.

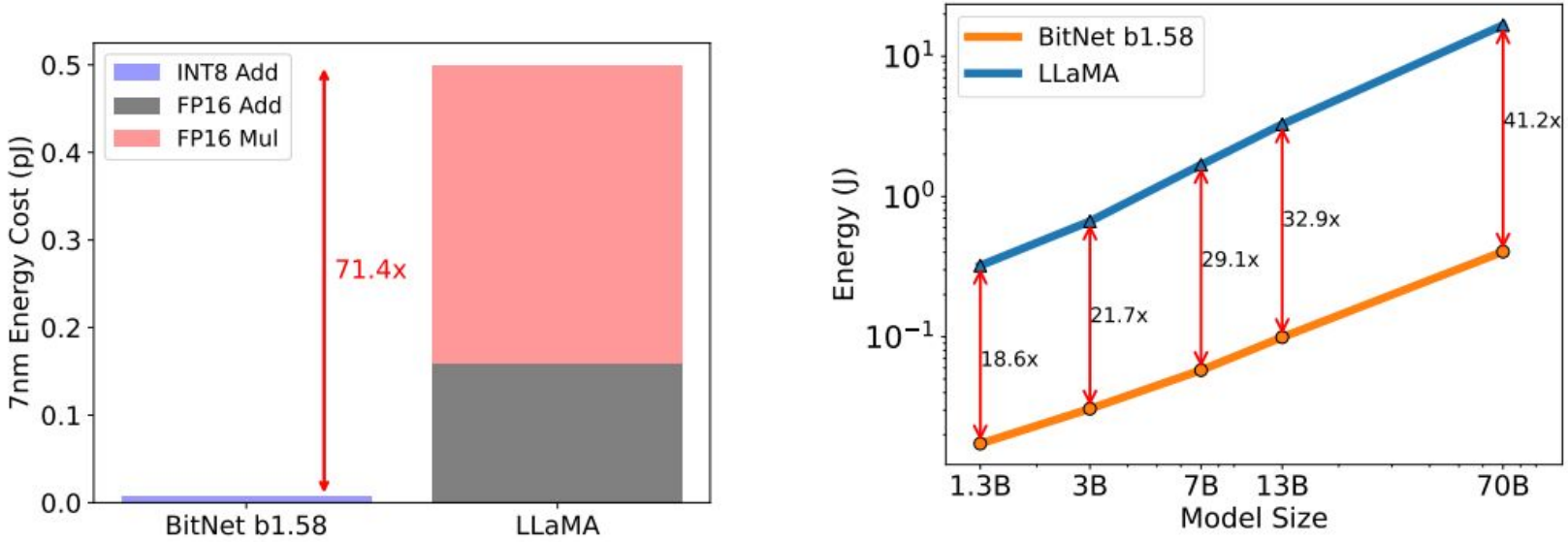
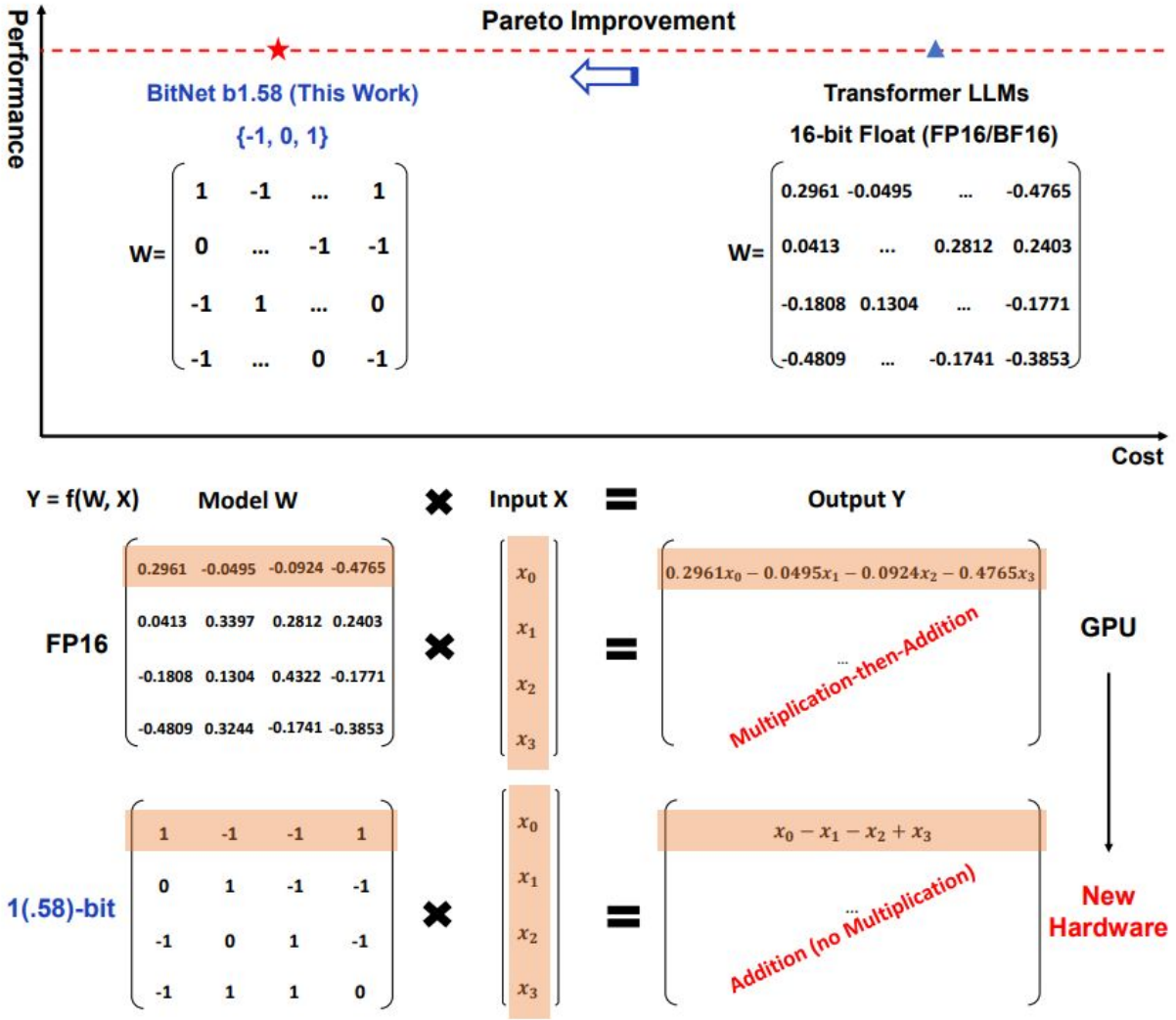


Figure 3: Energy consumption of BitNet b1.58 compared to LLaMA LLM at 7nm process nodes. On the left is the components of arithmetic operations energy. On the right is the end-to-end energy cost across different model sizes.

Models	Tokens	Winogrande	PIQA	SciQ	LAMBADA	ARC-easy	Avg.
StableLM-3B	2T	64.56	76.93	90.75	66.09	67.78	73.22
BitNet b1.58 3B	2T	**66.37**	**78.40**	**91.20**	**67.63**	**68.12**	**74.34**

Table 4: Comparison of BitNet b1.58 with StableLM-3B with 2T tokens.

Design new hardware specifically optimized for 1.58-bit LLMs ?



Thanks