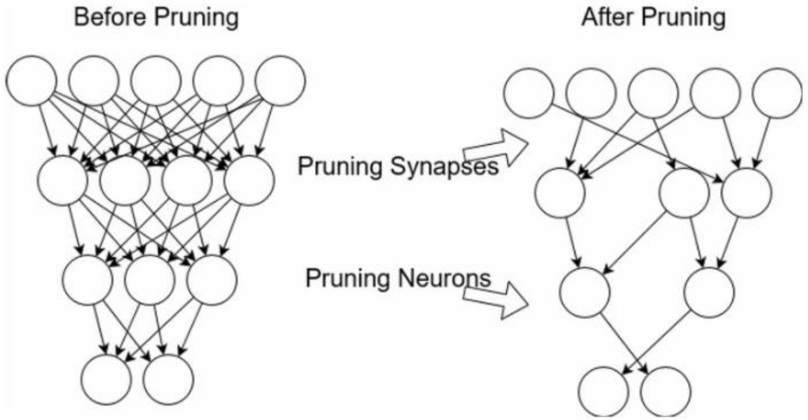
CS 598
AI Efficiency: Systems and Algorithms
Overview of Efficient and Effective Algorithms

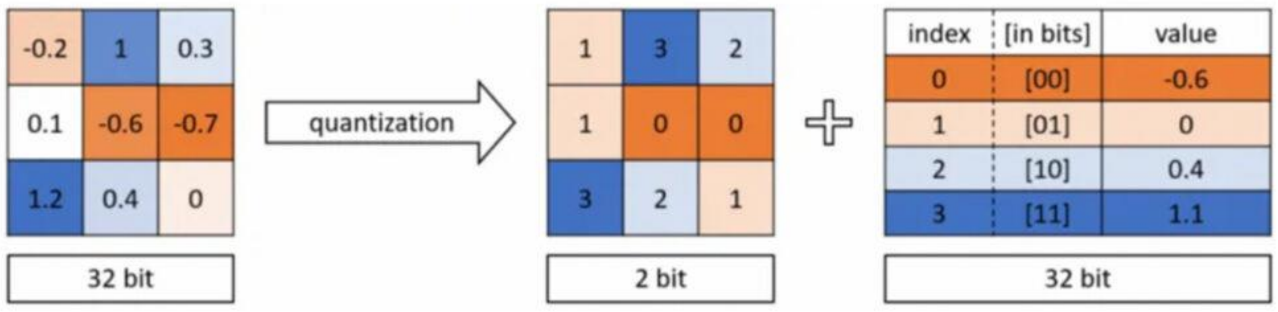
Minjia Zhang

Computer Science Department

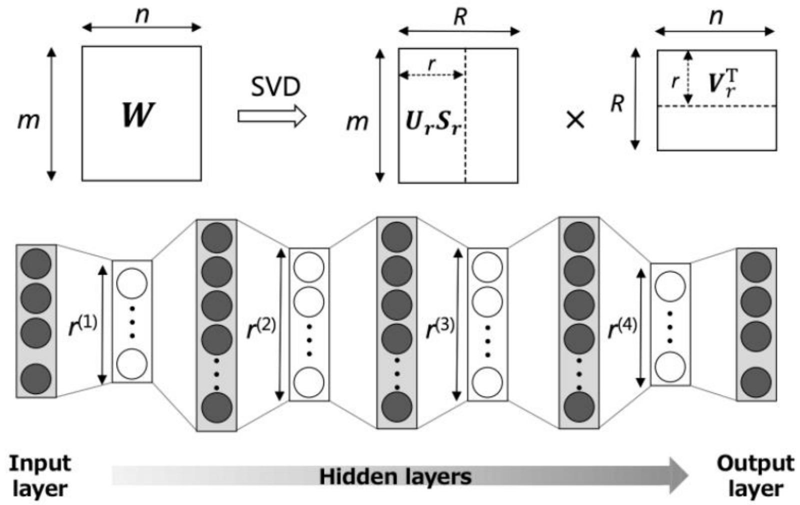
Compression Strategies



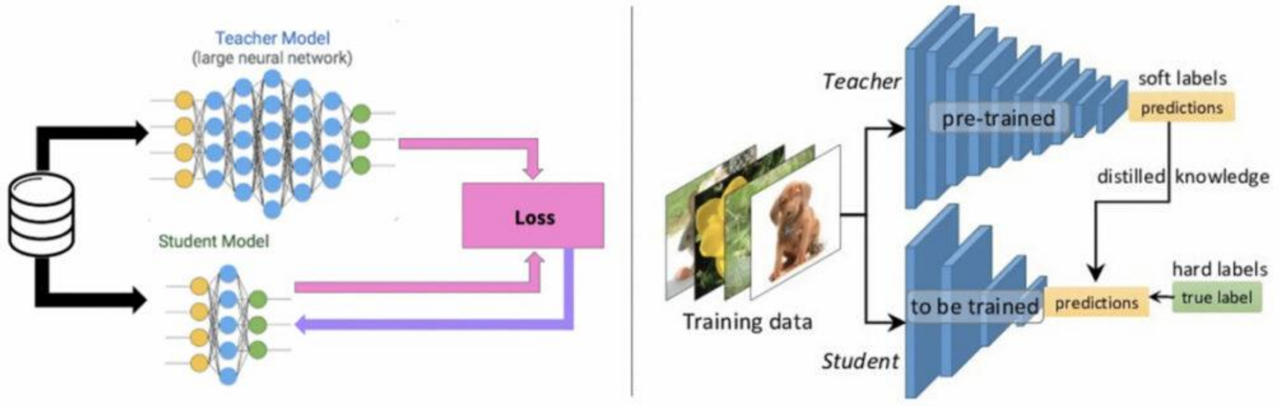
Quantization



Quantization



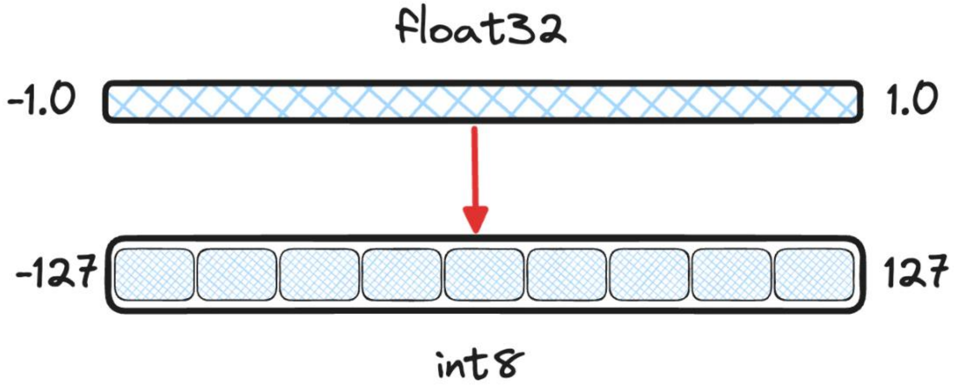
Low-rank decomposition



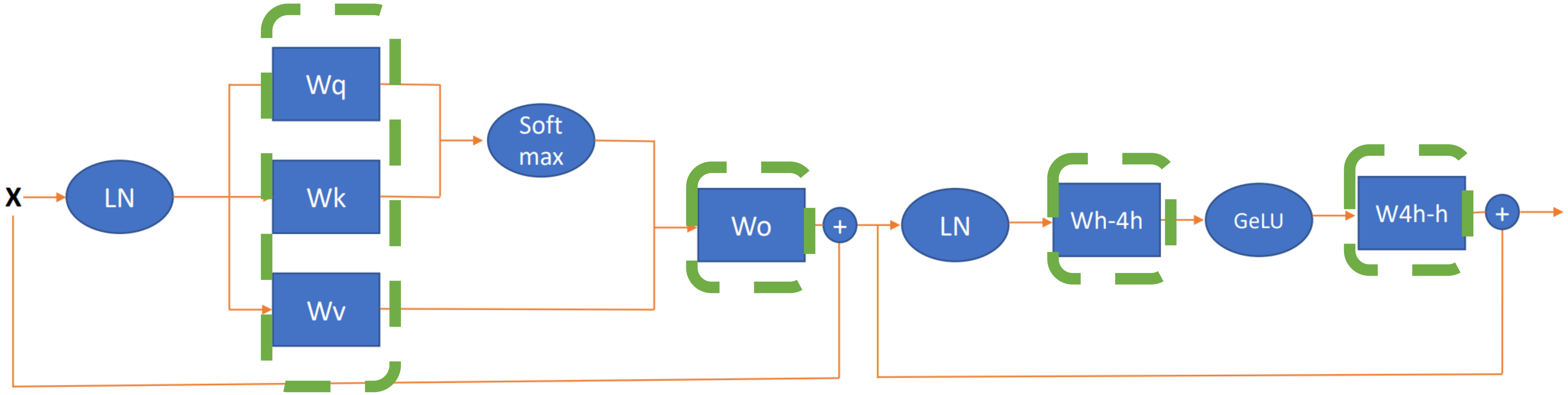
Distillation

Quantization: Quick Recap

- Reduce the bits per weight, saving memory consumption
- Accelerate inference speed on supporting hardware



8-bit Weight Quantization



- 8-bit weight quantization

$$\mathbf{x}_{quantize} = round\left(clamp(\frac{\mathbf{x}}{S}, -2^{bit-1}, 2^{bit-1}-1)\right)$$

FP32 weight matrix

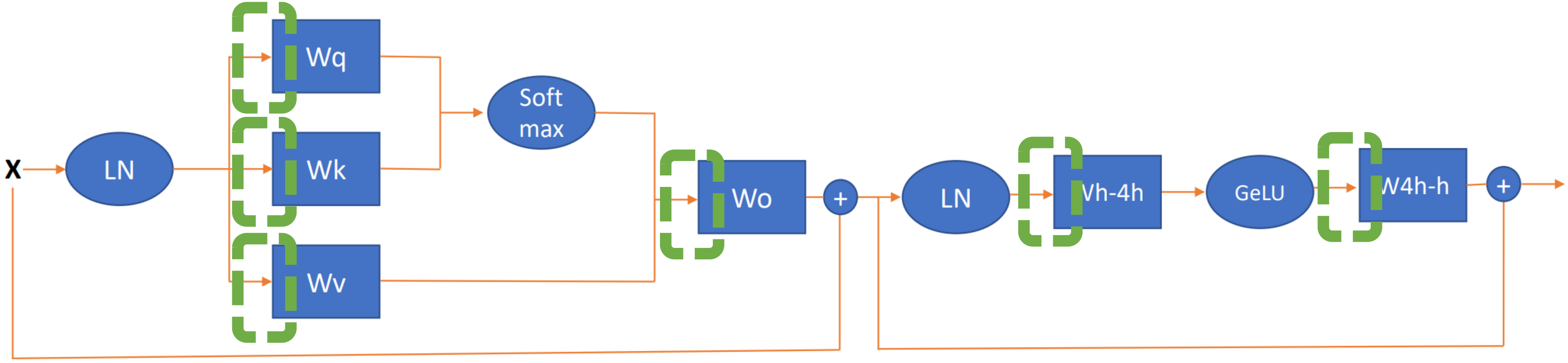
1.1	2.2	0.1	-0.1	-5.5	-6.6
…					
…					
…					
…					
1.1	2.1	0.1	-0.1	-4.8	-6.6

≈ Scaling Factor $1/S$ 0.05 *

8-bit quantization

21	42	2	-2	-106	-127
…					
…					
…					
…					
21	40	2	-2	-92	-127

8-bit Activation Quantization



- 8-bit activation
  (Input to the linear layer)

$$\mathbf{x}_{quantize} = round\left(clamp(\frac{\mathbf{x}}{S}, -2^{bit-1}, 2^{bit-1} - 1)\right)$$

FP32 input matrix

1.1	2.2	0.1	-0.1	-5.5	-6.6
…					
…					
…					
…					
1.1	2.1	0.1	-0.1	-4.8	-6.6

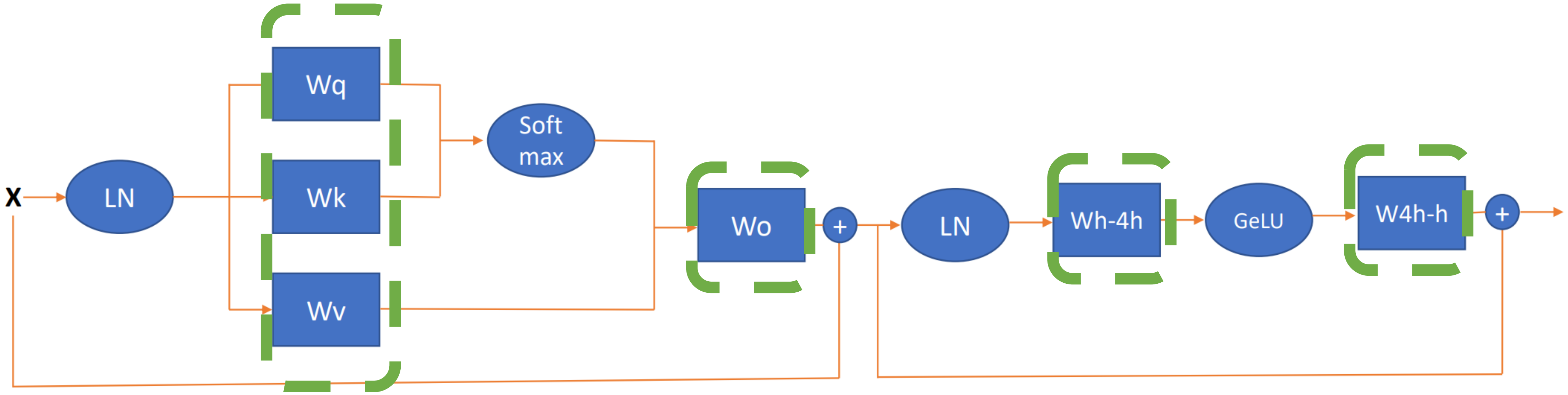
Scaling Factor $1/S$

$\approx$ 0.05*

8-bit quantization

21	42	2	-2	-106	-127
…					
…					
…					
…					
21	40	2	-2	-92	-127

Weight Ternarization



- Ternarization (weight)
  W: weight matrix, FP32.
  Q(W): Quantization mapping, 2-bit.
  With α=$\|W\|_1$/n, for some scalar s

$$Q(W_{ij}) = \begin{cases} \alpha \cdot \mathrm{sign}(W_{ij}) & \text{when } |W_{ij}| > s \\ 0 & \text{when } |W_{ij}| < s \end{cases}$$

FP32 weight matrix

1.1	2.2	0.1	-0.1	-5.5	-6.6
…					
…					
…					
…					
1.1	2.1	0.1	-0.1	-4.8	-6.0

≈

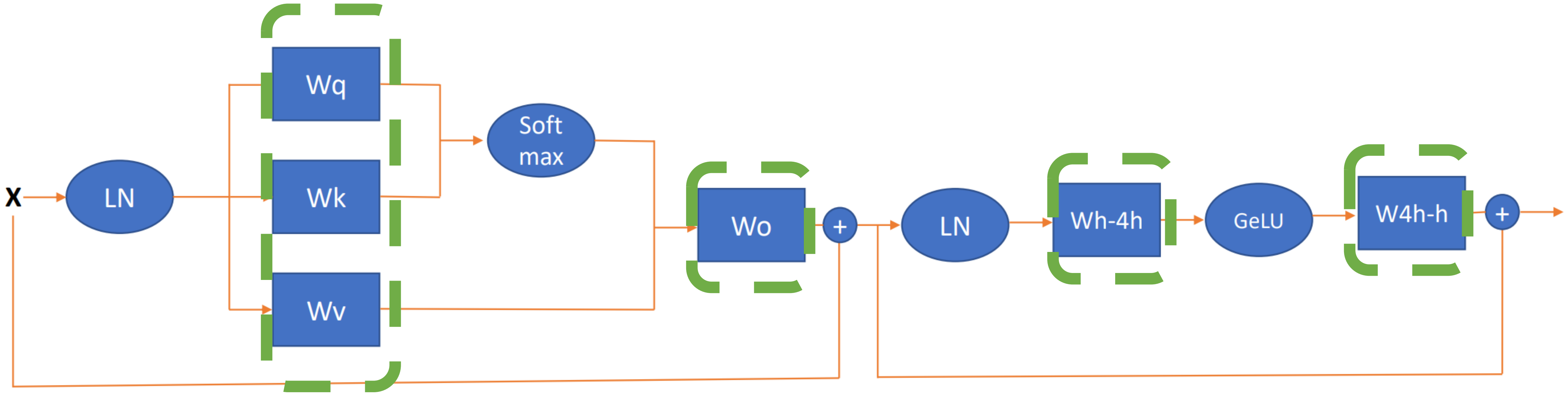
Scaling Factor α

2.06 *

2-bit quantization

1	1	0	0	-1	-1
…					
…					
…					
…					
1	1	0	0	-1	-1

Weight Binarization



- Binarization (weight)

  W: weight matrix, FP32.

  Q(W): Quantization mapping, 1-bit.

  With α= $\|W\|_1$/n

  $$Q(W_{ij}) = \alpha \cdot \text{sign}(W_{ij})$$

FP32 weight matrix

1.1	2.2	0.1	-0.1	-5.5	-6.6
…					
…					
…					
…					
1.1	2.1	0.1	-0.1	-4.8	-6.0

≈ 1.4 *

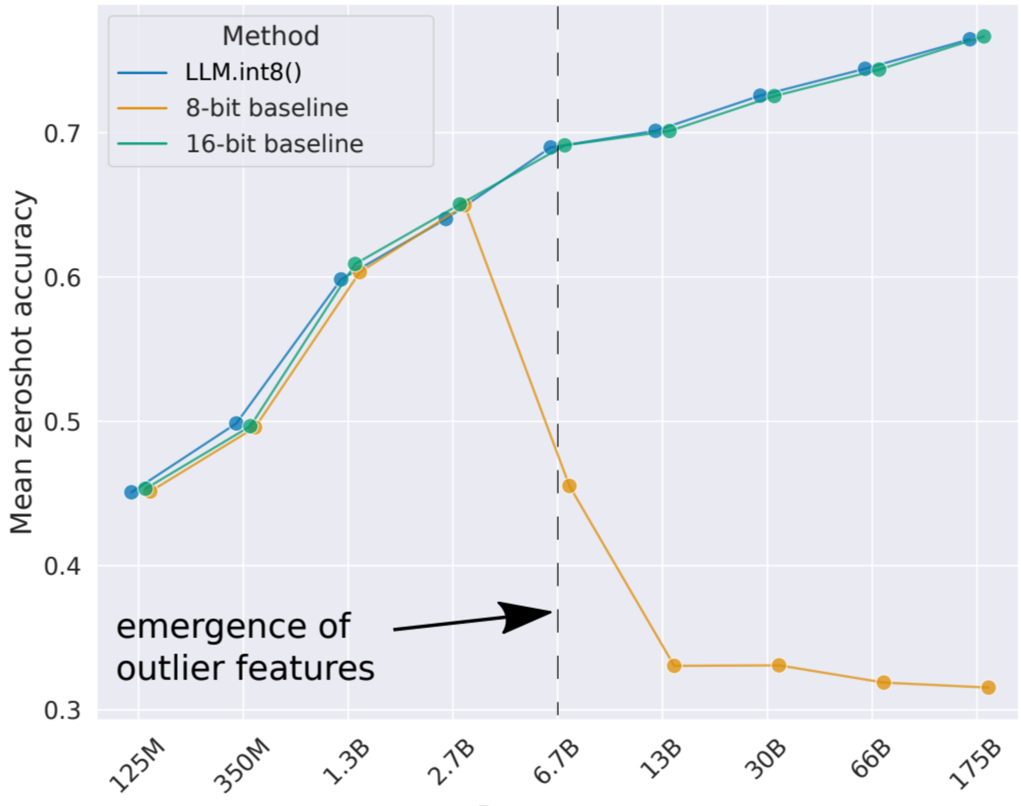
Scaling Factor α

1-bit quantization

1	1	1	-1	-1	-1
…					
…					
…					
…					
1	1	1	-1	-1	-1

Challenges to Quantize LLMs

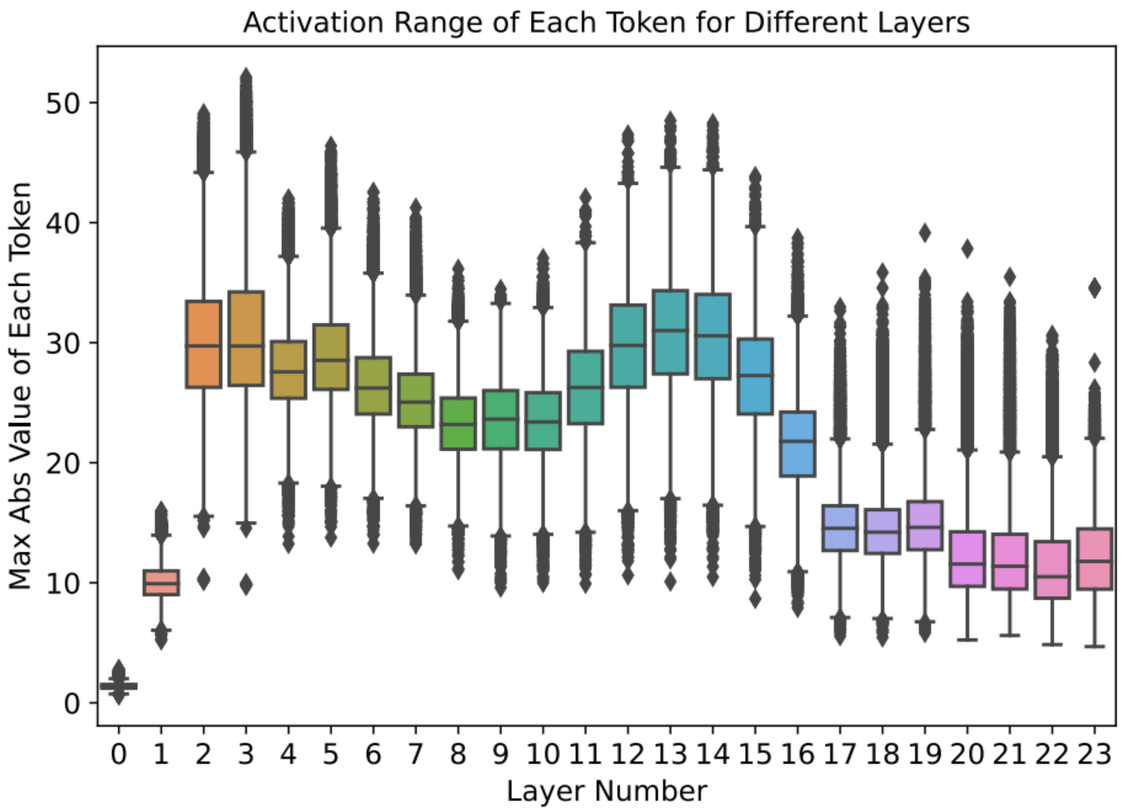
- Standard quantization strategy leads to catastrophic accuracy drop



LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, 2023

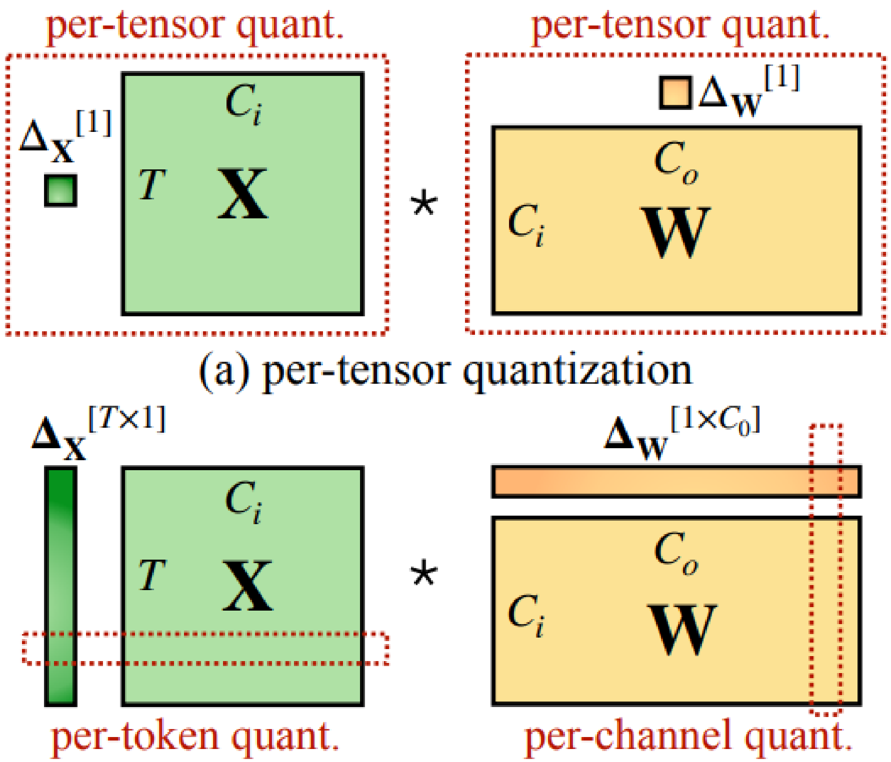
Challenges to Quantize LLMs

- High dynamic ranges of activation, leading to large quantization errors



Fine-grained Quantization

- Per-tensor quantization
  - Low accuracy
  - Fast to quantize/dequantize
- Per-token/channel quantization
  - High accuracy
  - Slower to quantize/dequantize
  - Custom kernels required



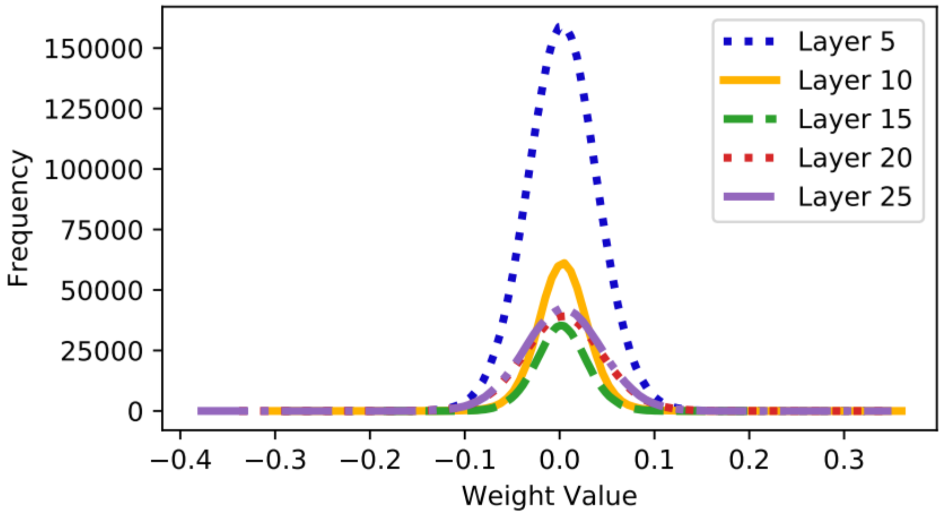
(a) per-tensor quantization

(b) per-token + per-channel quantization

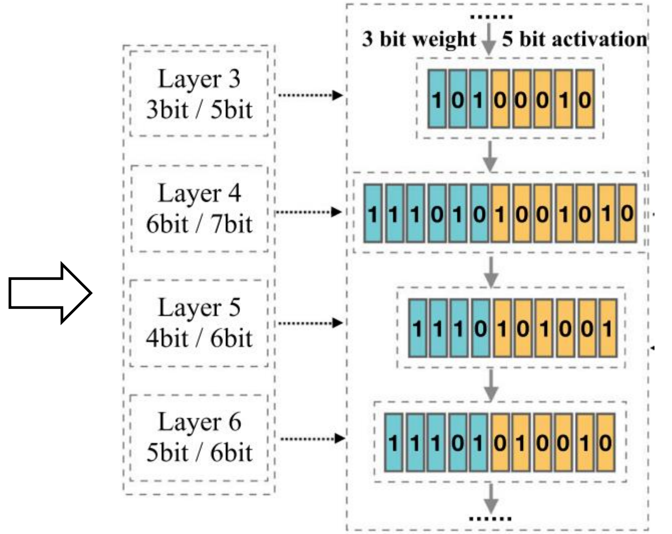
ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers, NeurIPS 2022

Mixed Precision Quantization

- Weights follow Gaussian distribution
- Outliers remain in original form, quantize the rest of the values
- Different bits for different layers



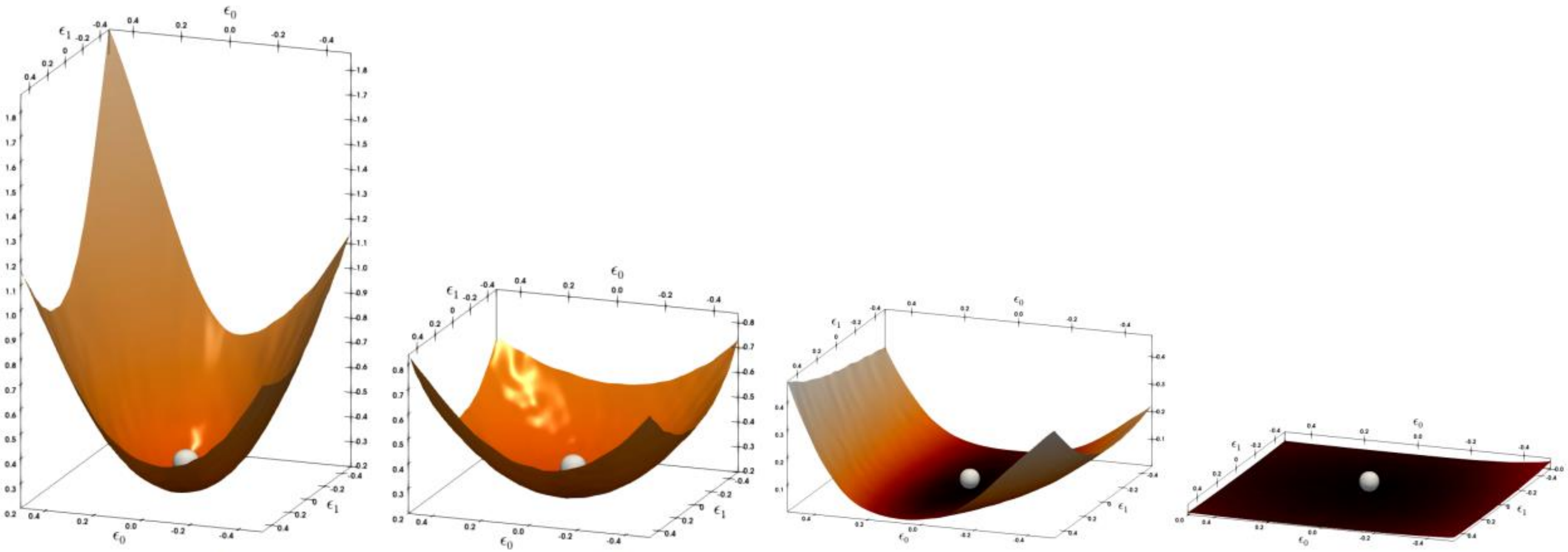
Per-layer weight distribution of BERT model



GOBO: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference, MICRO 2020

Second Order Information

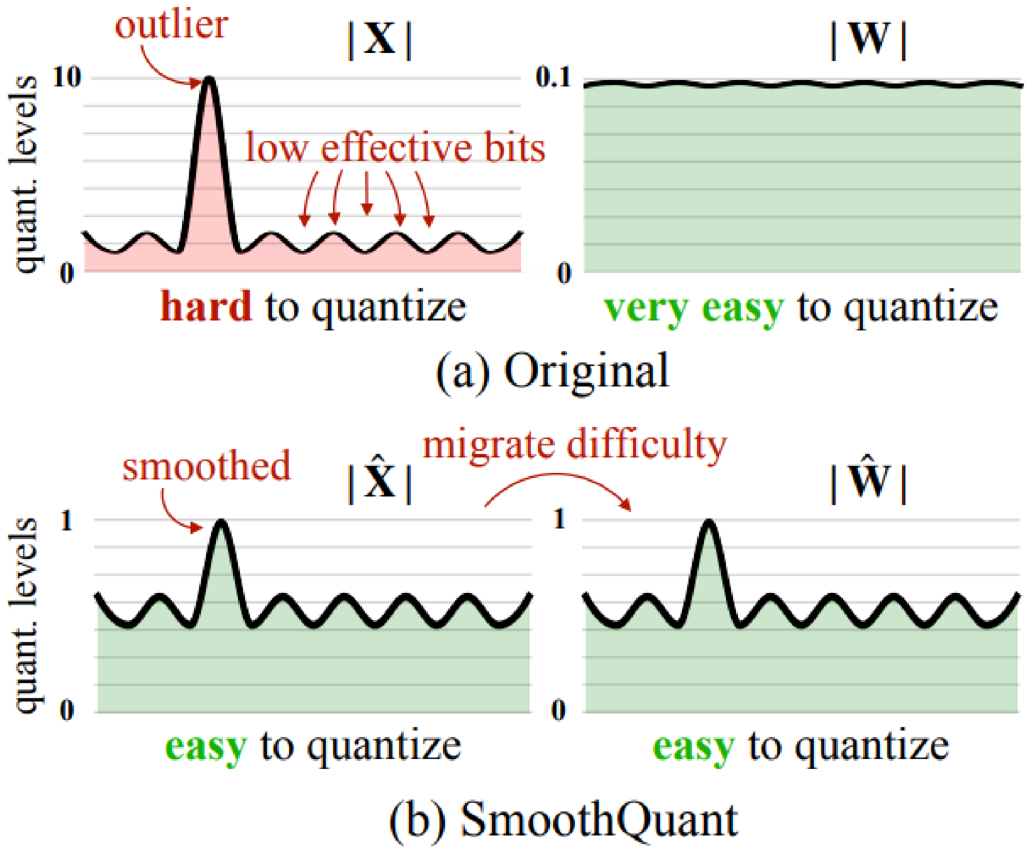
- Analyze the loss curvature (Hessian matrices) to help identify layer sensitivity



(a) MNLI 4th layer (b) MNLI 10th layer (c) CoNLL-03 4th layer (d) CoNLL-03 11th layer

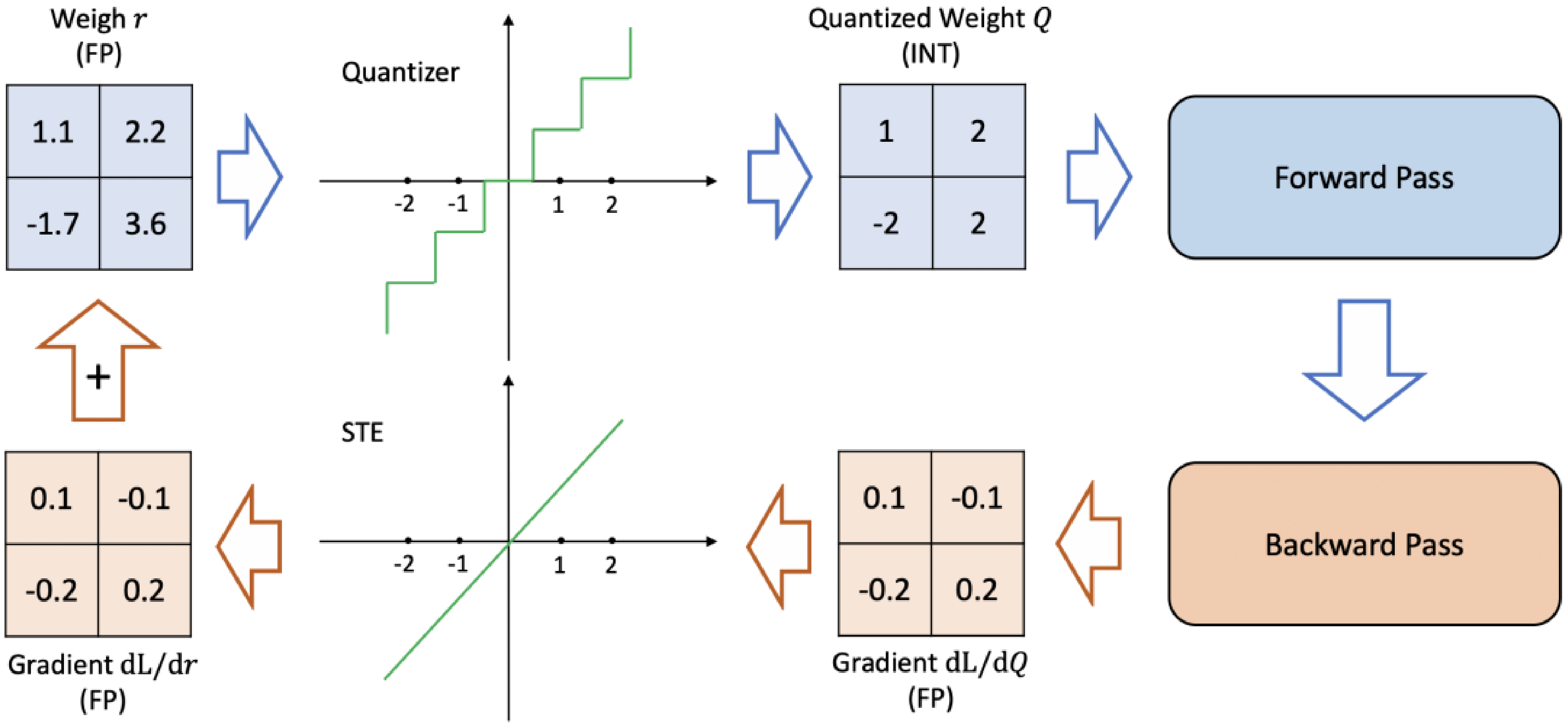
GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers, ICLR 2023

Outlier Smoothing



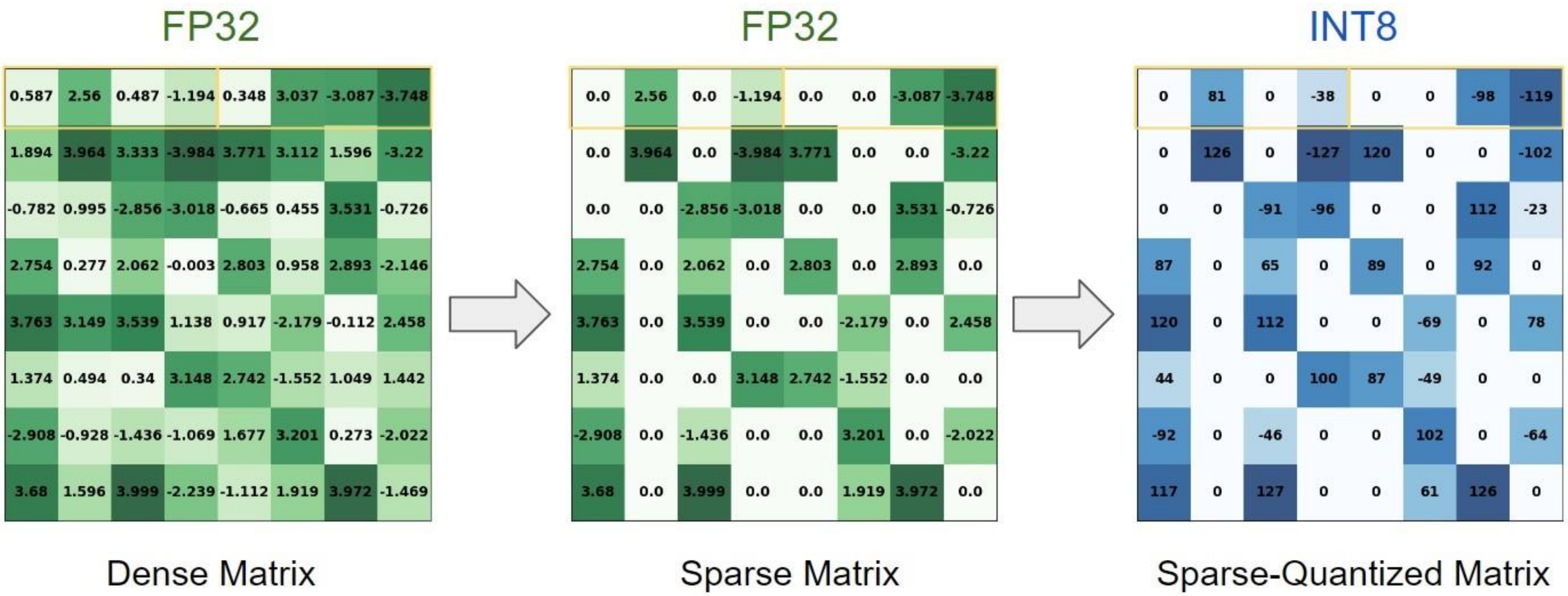
SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models, ICML 2023

Quantization-Aware Training



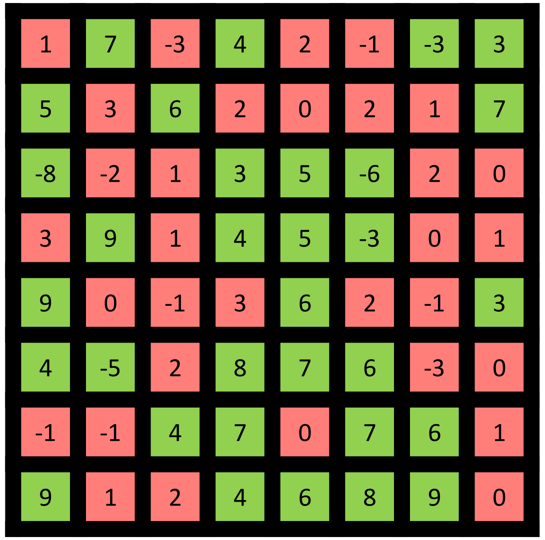
EfficientQAT: Efficient Quantization-Aware Training for Large Language Models, 2024

Sparsification



Model Pruning

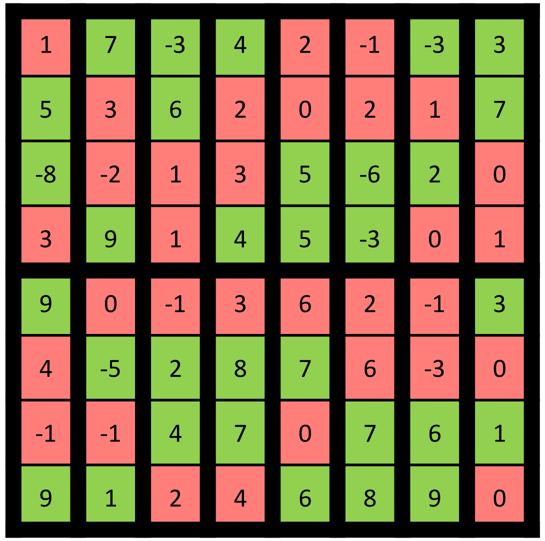
- **Unstructured (connection) Sparsity**:
- High accuracy
- No performance improvement or performance regression



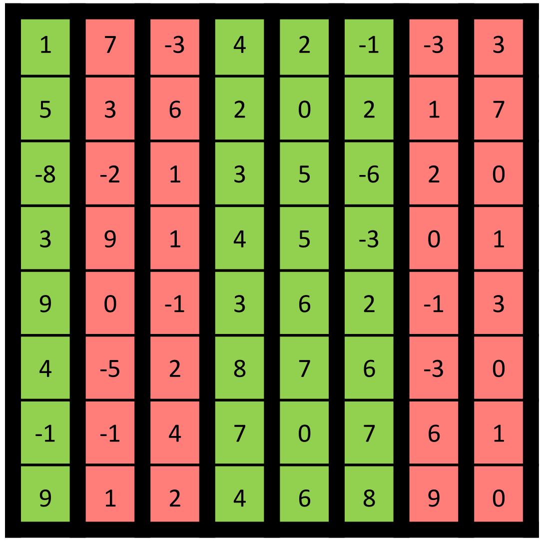
Unstructured Sparsity

- **N:M Semi-Structured Sparsity**:
- High accuracy
- High performance improvement

- **Structured Sparsity**:
- Large accuracy degradation
- High performance scalability



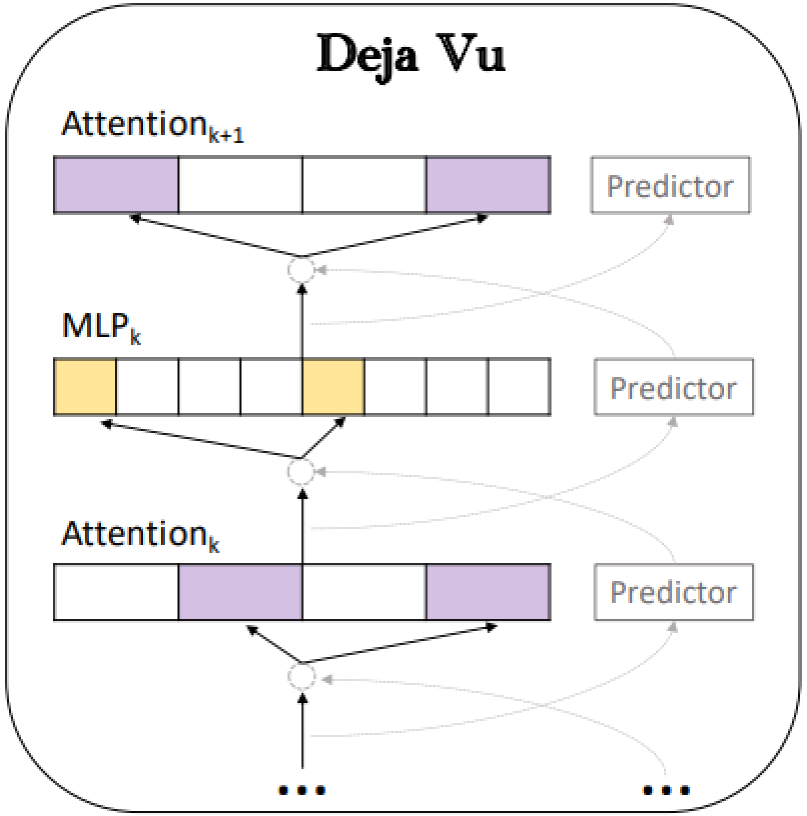
Semi-Structured Sparsity (4:2 N:M)



Structured Sparsity (Column-wise Sparsity)

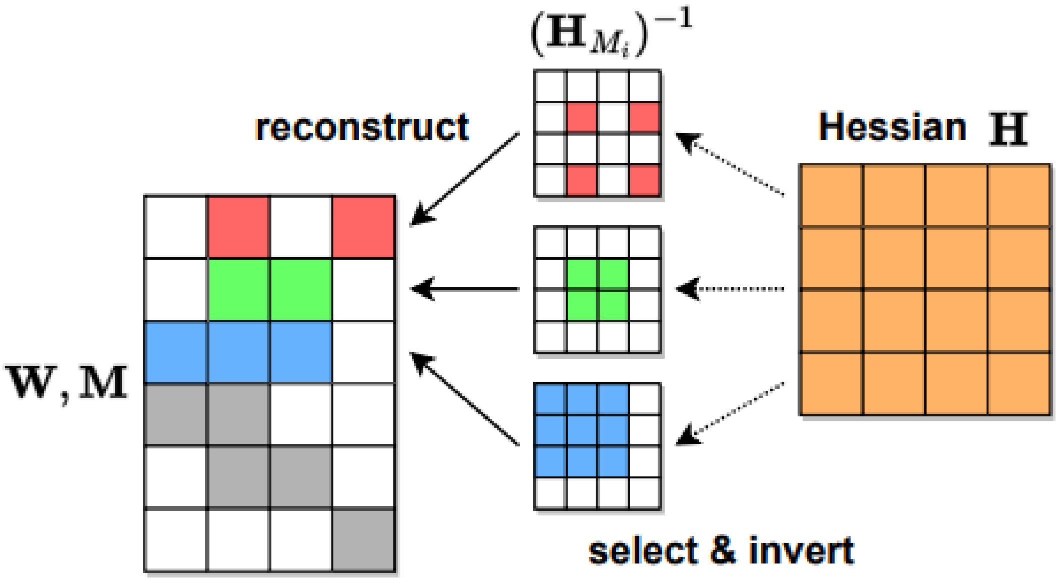
SliceGPT: Compress Large Language Models by Deleting Rows and Columns, ICLR 2024

Model Pruning

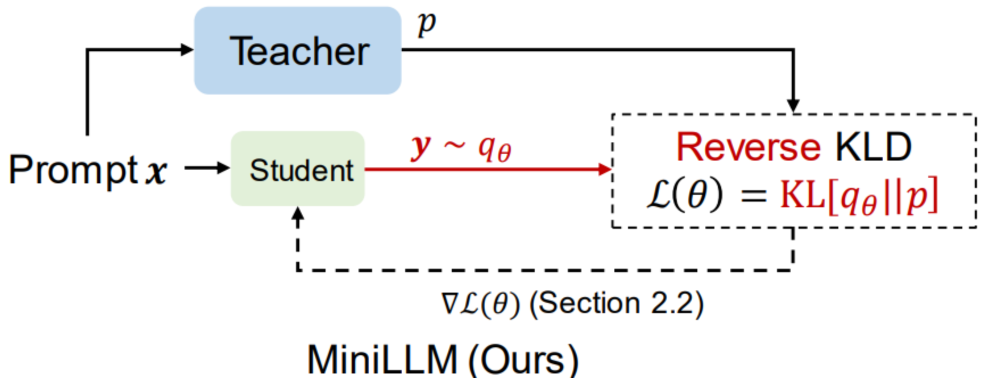


Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time, 2023

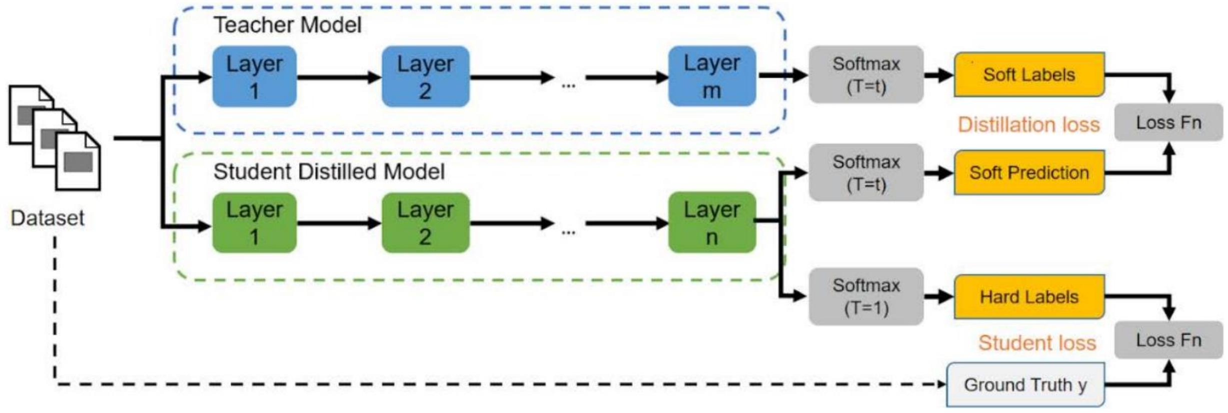
SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot, 2023



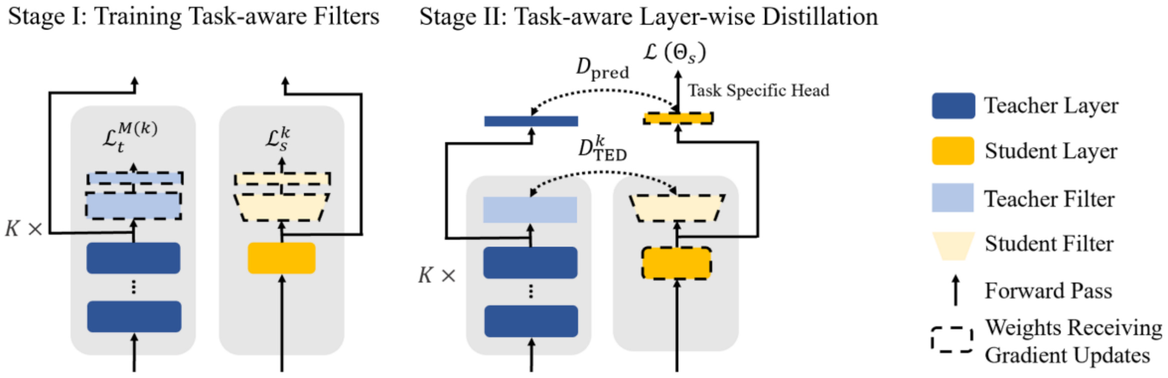
Knowledge Distillation



MiniLLM: Knowledge distillation of large language models, 2024

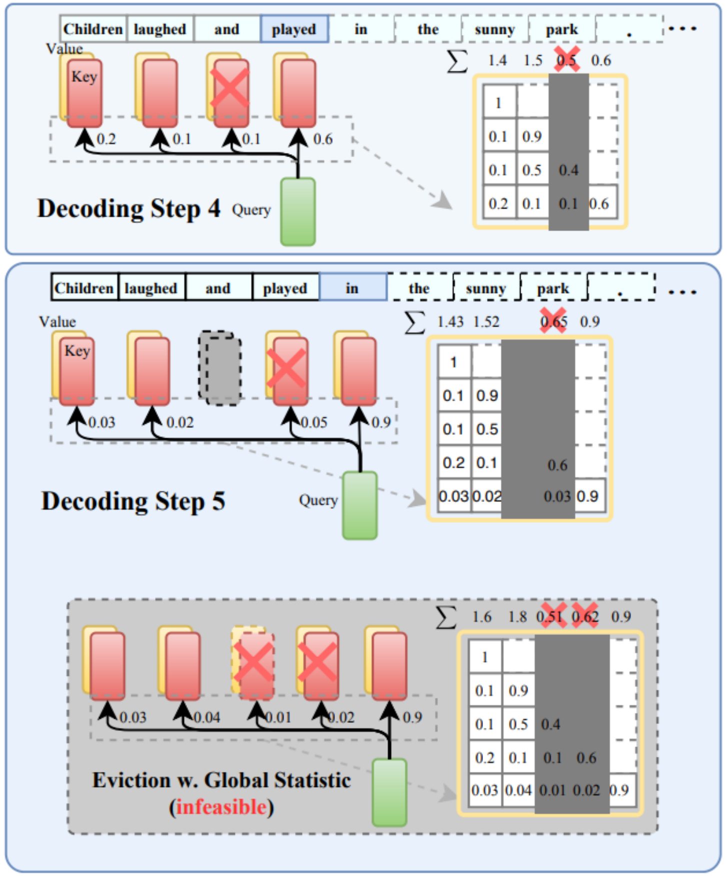


Distilling the Knowledge in a Neural Network , 2015



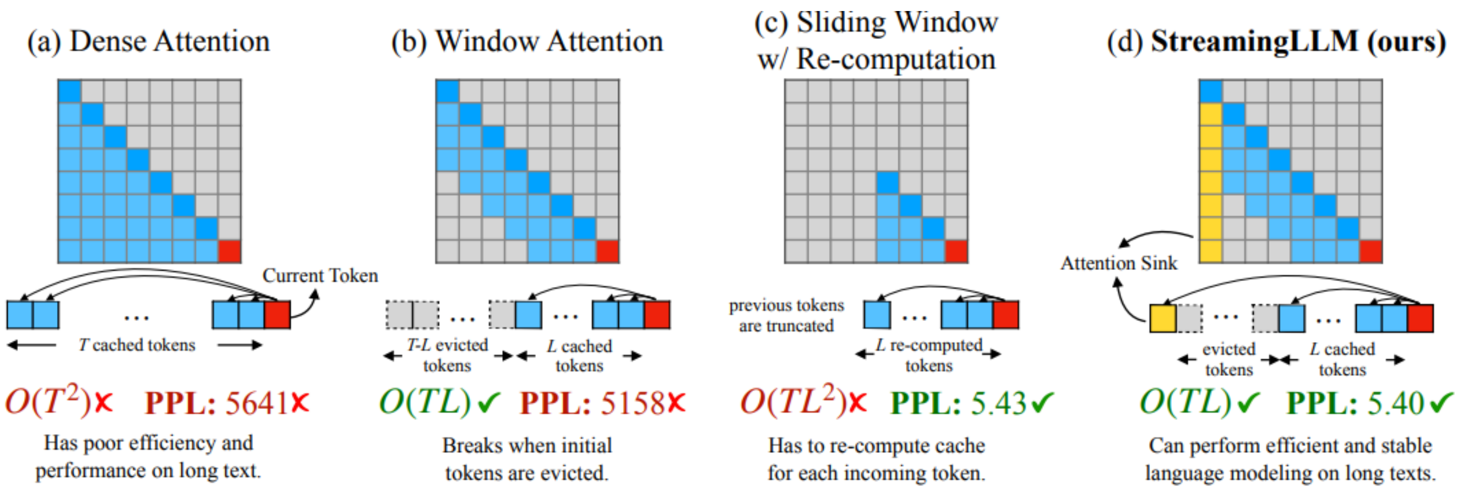
Less is More: Task-aware Layer-wise Distillation for Language Model Compression, 2024

KV Cache Compression

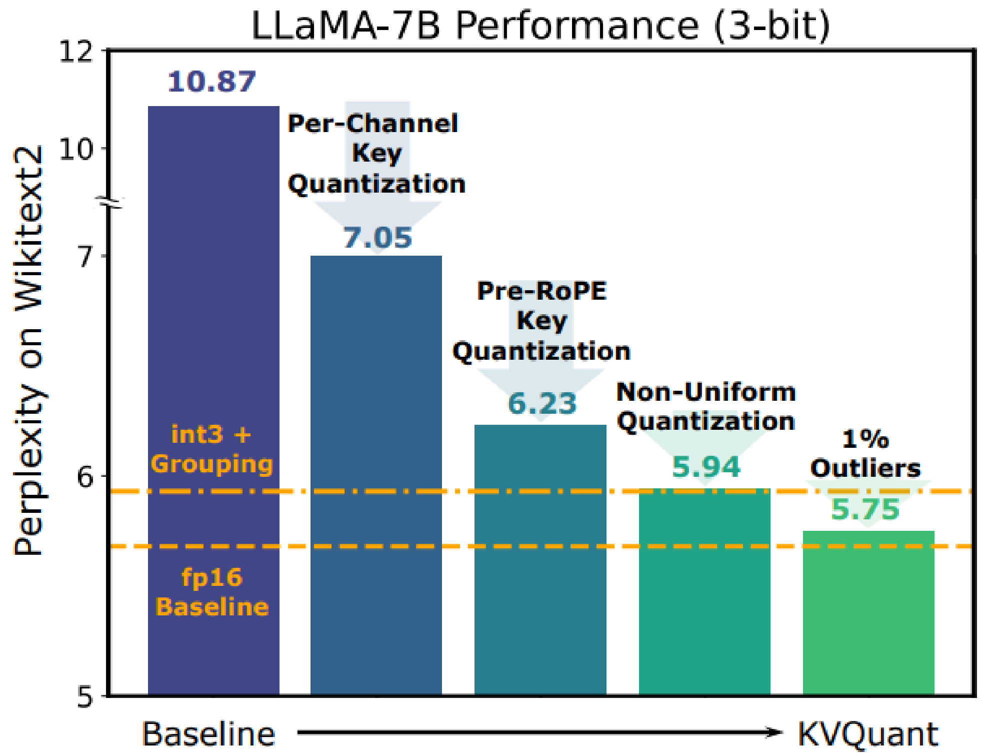


H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models, 2023

Efficient Streaming Language Models with Attention Sinks, ICL 2024

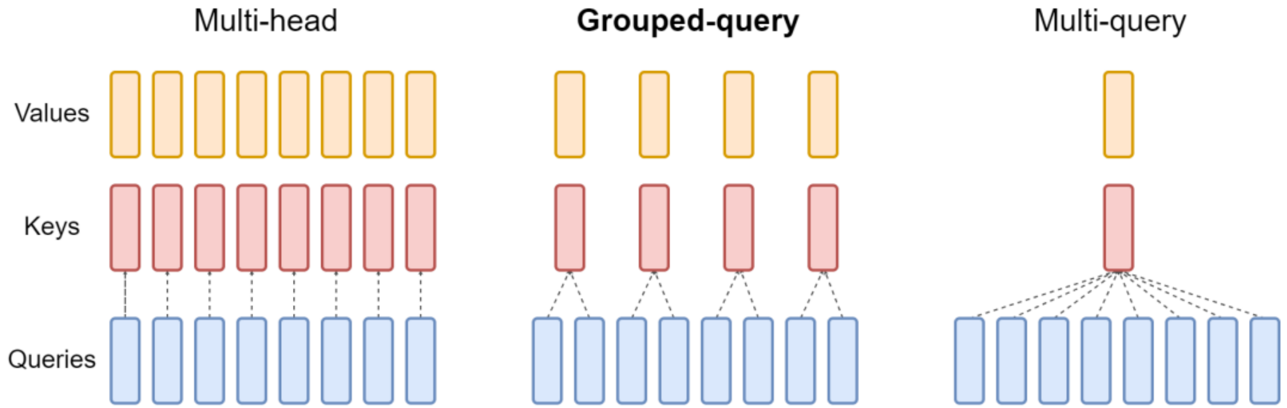


KV Cache Compression



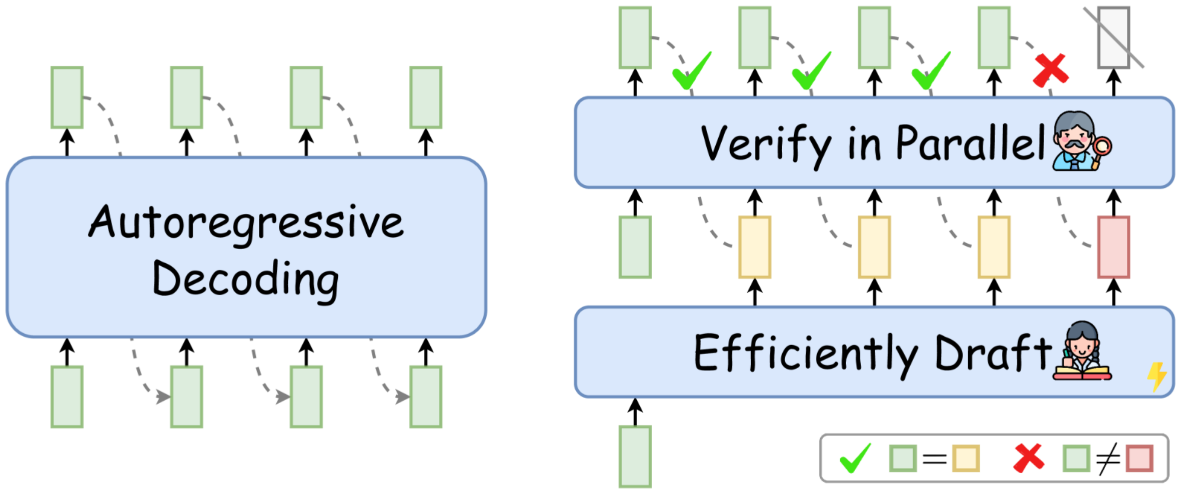
KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization, 2024

GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, 2023

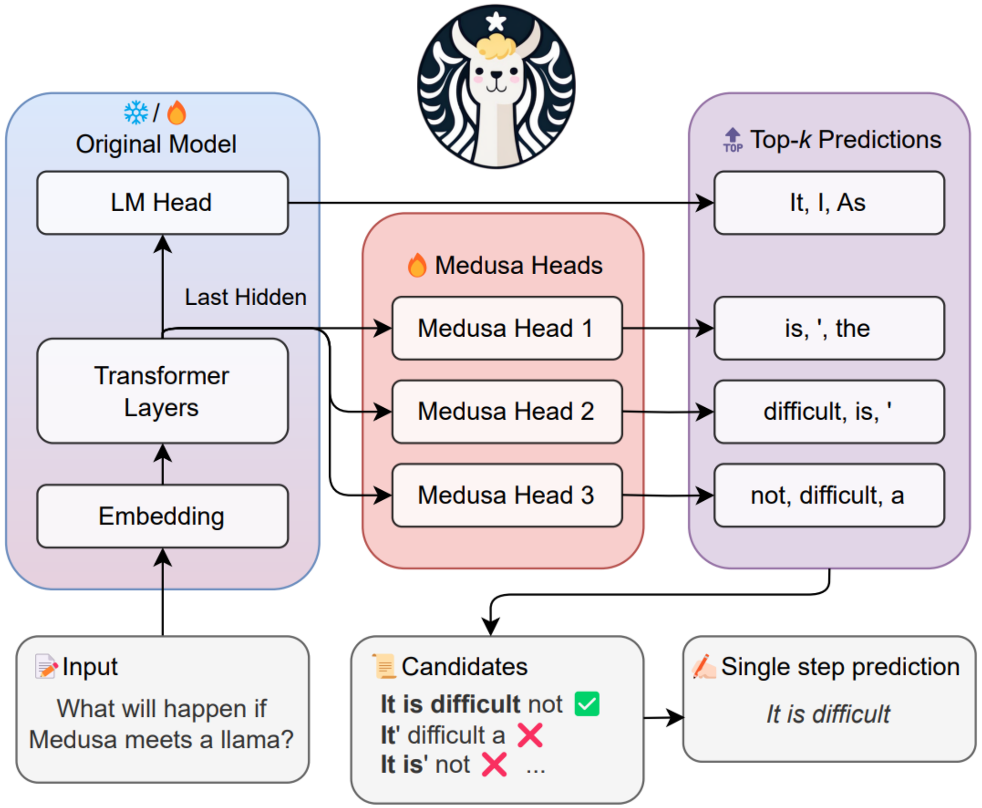


Speculative/Parallel Decoding

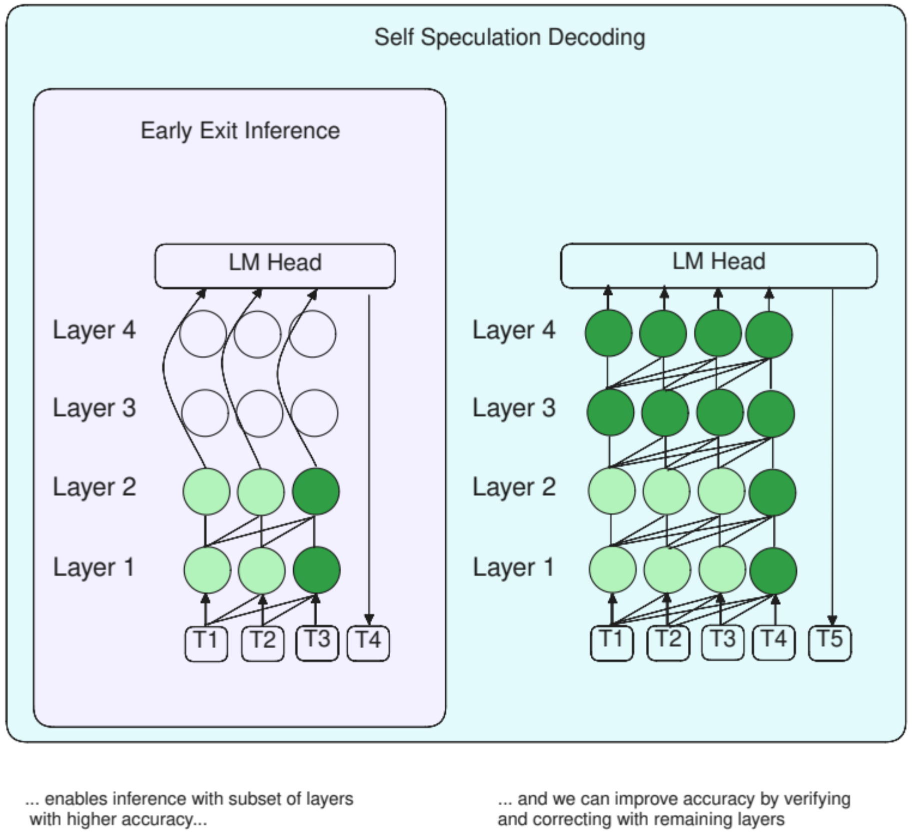
MEDUSA: Simple LLM Inference Acceleration Framework with Multiple, 2024



Fast Inference from Transformers via Speculative Decoding, 2023

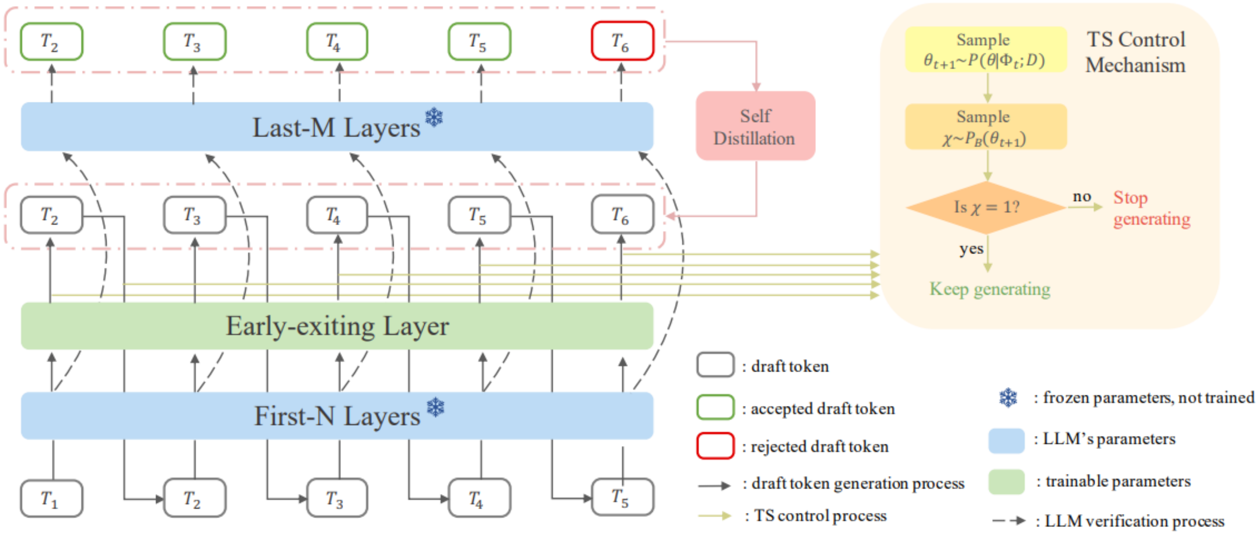


Early-Exit Inference

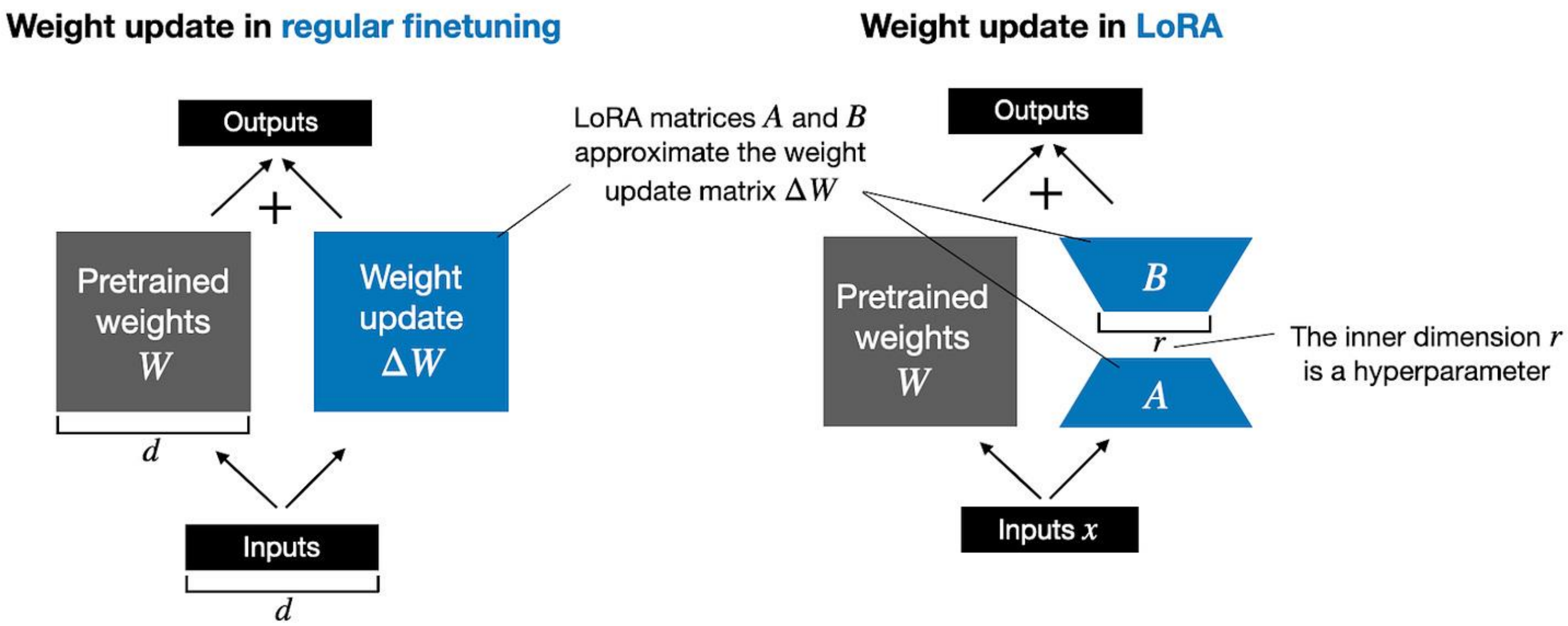


LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding, 2024

Speculative Decoding via Early-exiting for Faster LLM Inference with Thompson Sampling Control Mechanism, 2024



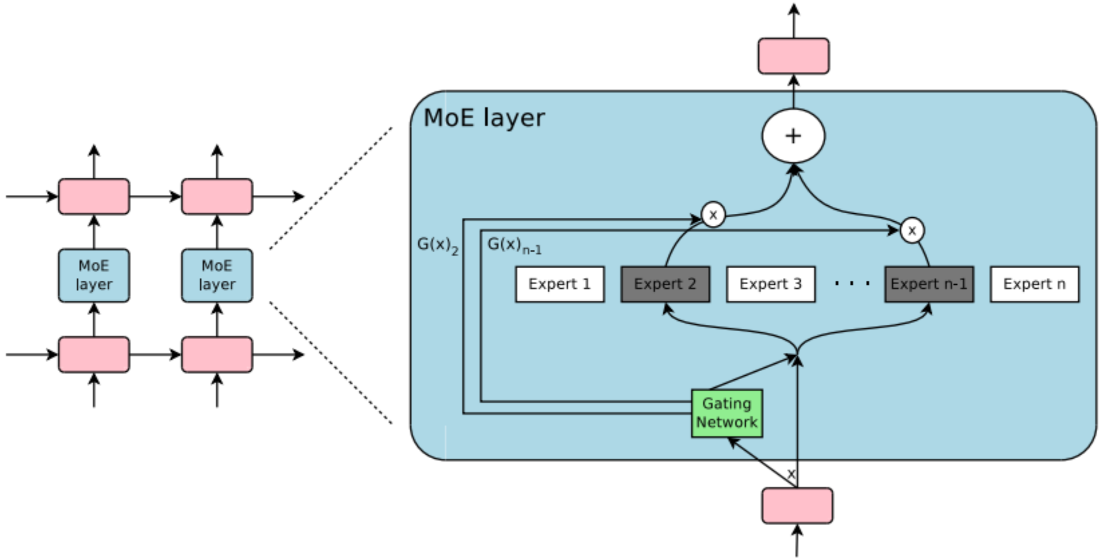
Efficient Fine-Tuning



LoRA: Low-Rank Adaptation of Large Language Models, 2021

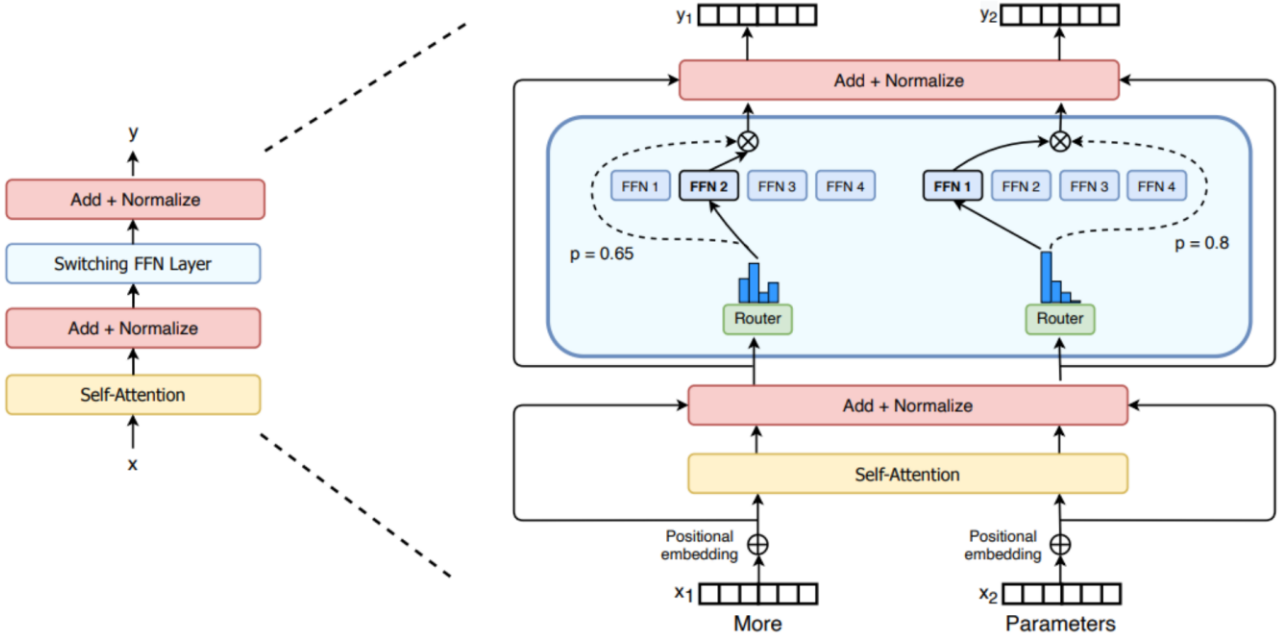
QLoRA: Efficient Finetuning of Quantized LLMs, 2024

Mixture-of-Expert Models are Sparse and Need Less Compute

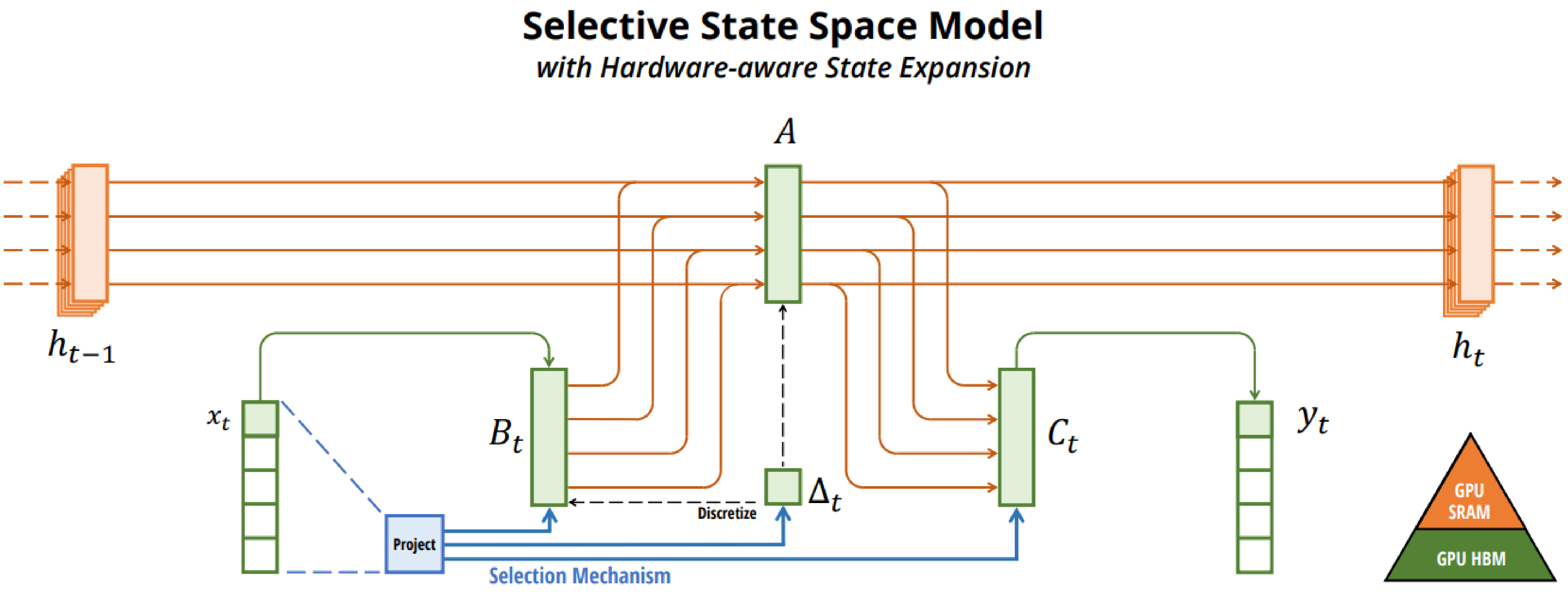


Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, 2017

Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, 2021



Mamba – Linear Time Sequence Model



Mamba: Linear-Time Sequence Modeling with Selective State Spaces, 2024

QA