CS 498: Machine Learning System
Spring 2025

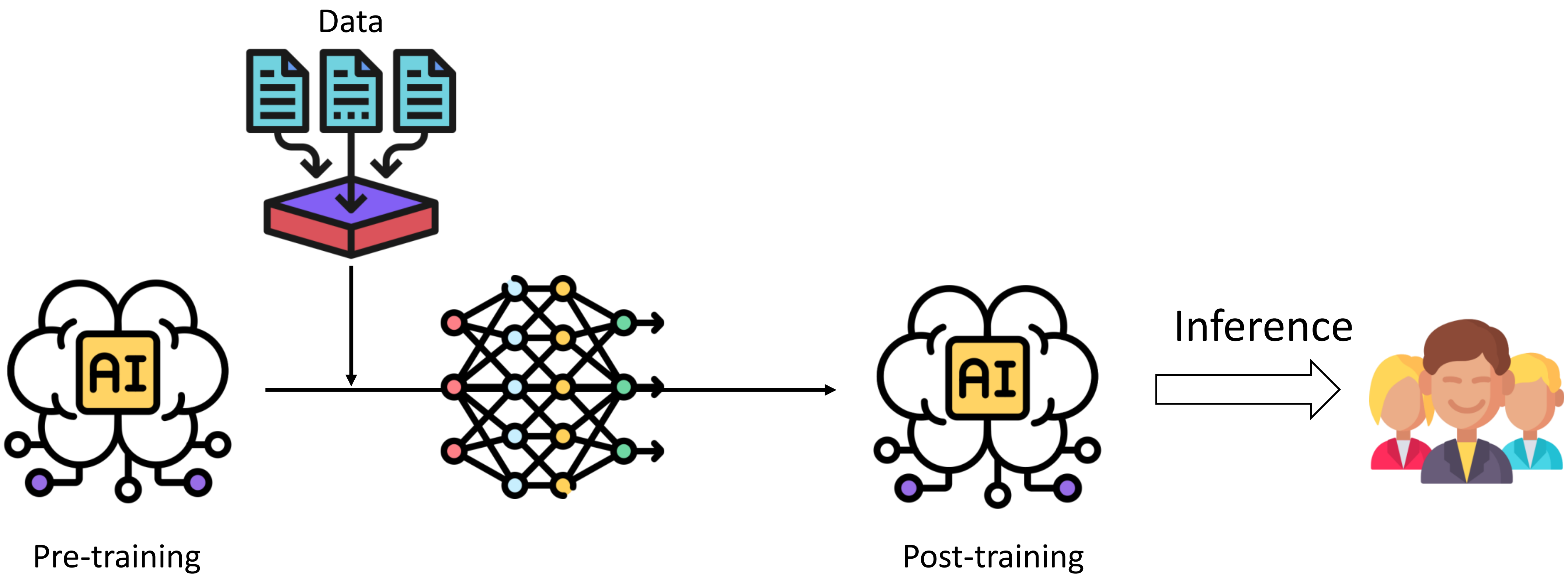
Minjia Zhang

The Grainger College of Engineering

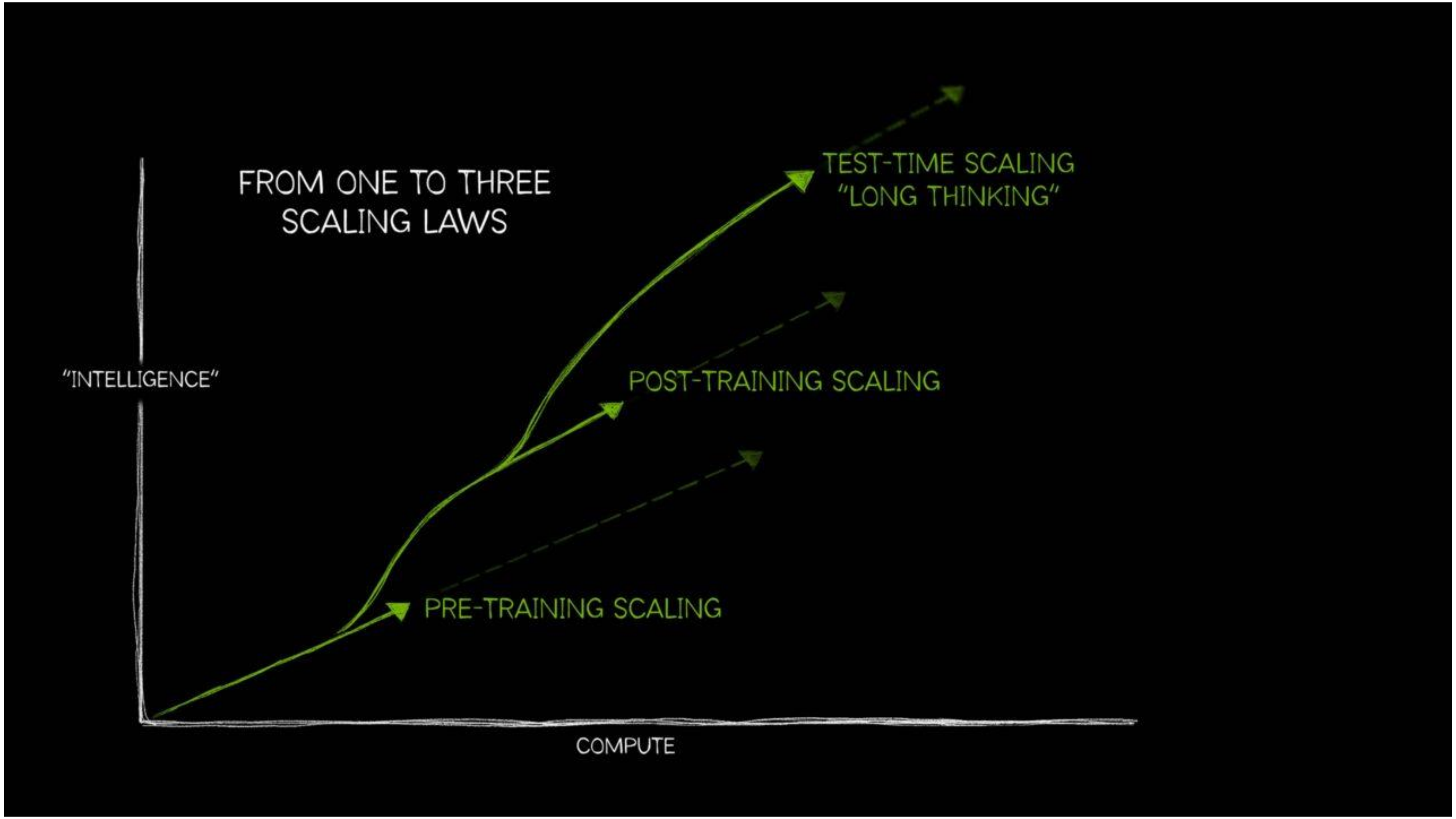
Today

DL Inference Overview

What is Model Serving/Inference?

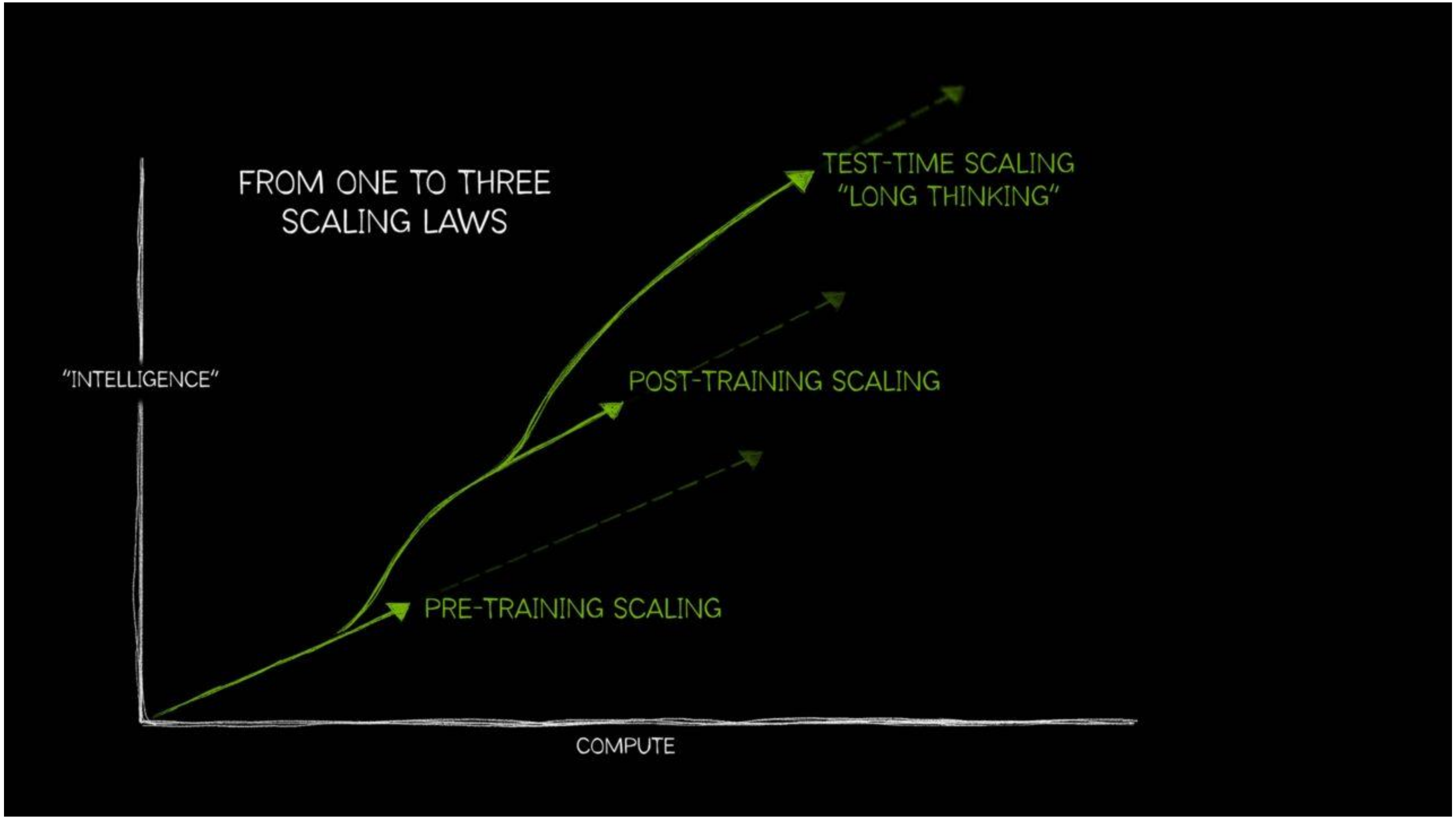


Inference Time Scaling Requires More Compute



NVIDIA Jensen Huang at CES 2025

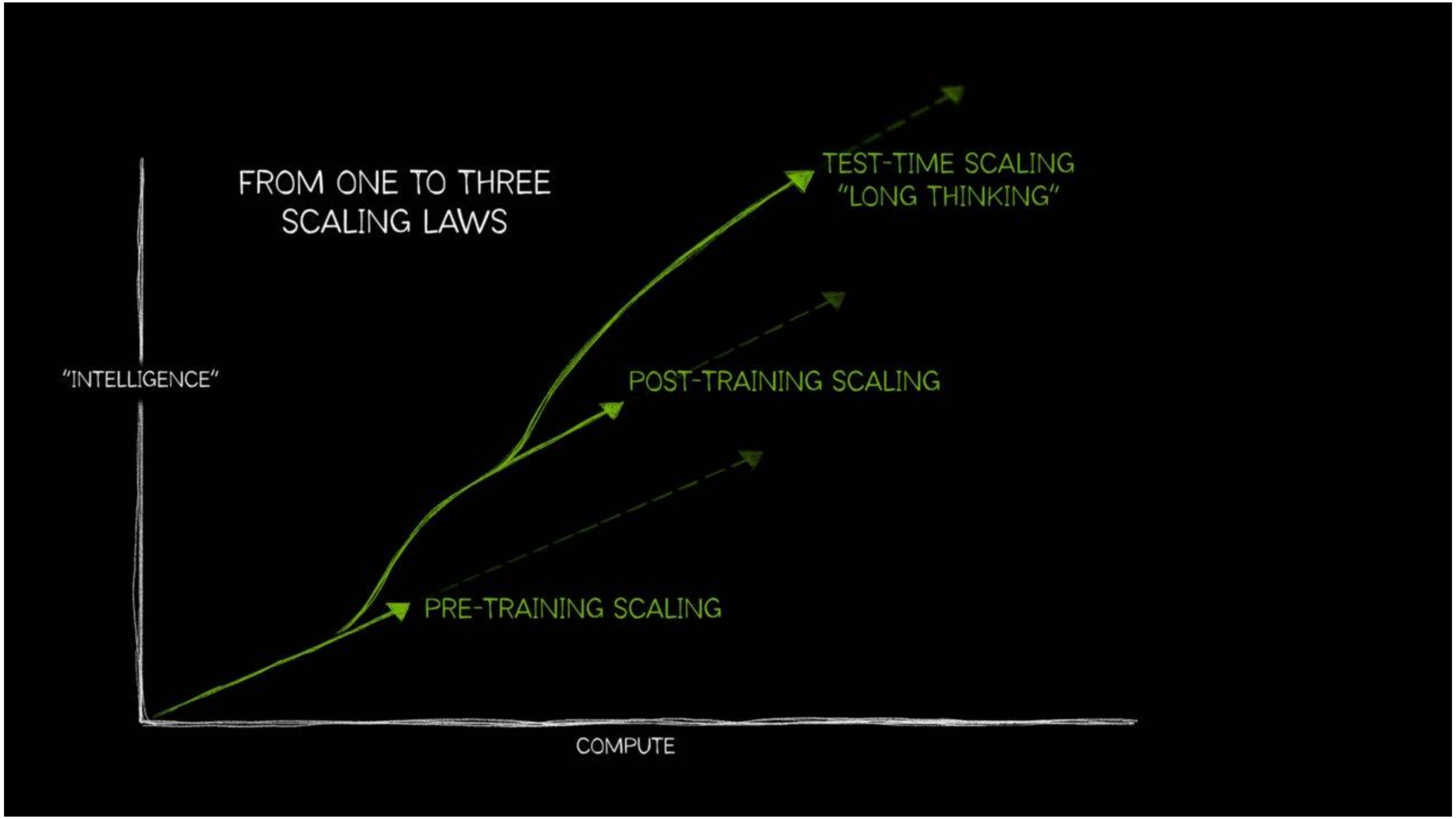
Inference Time Scaling Requires More Compute



1 Pretraining scaling = bigger model and data, better performance

NVIDIA Jensen Huang at CES 2025

Inference Time Scaling Requires More Compute

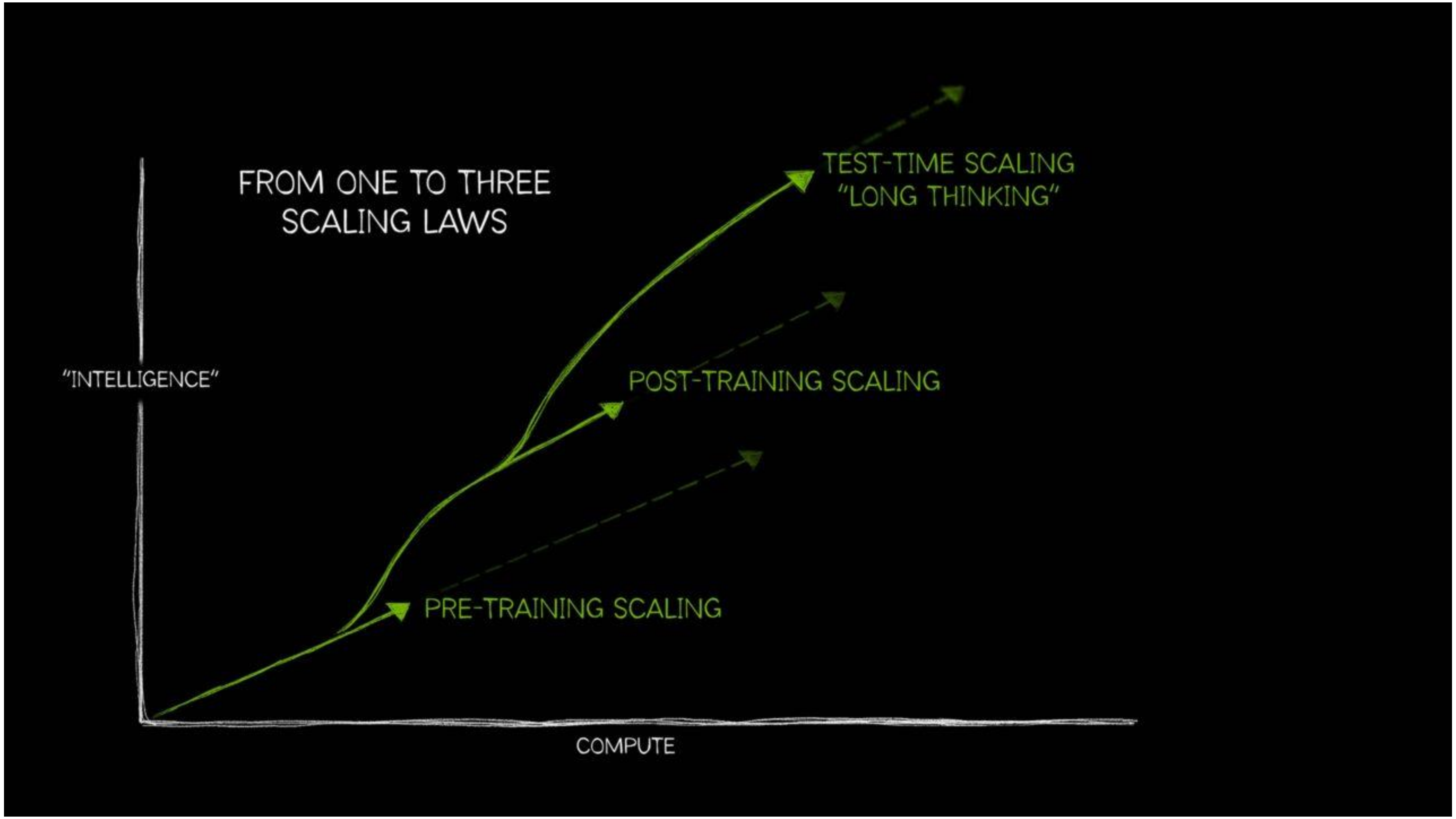


$\boxed{1}$ Pretraining scaling = bigger model and data, better performance

$\boxed{2}$ Post-training scaling = fine-tuning for precision

NVIDIA Jensen Huang at CES 2025

Inference Time Scaling Requires More Compute



1 Pretraining scaling = bigger model and data, better performance

2 Post-training scaling = fine-tuning for precision

3 Test-time scaling (long thinking) = multi-pass reasoning for complex problems

NVIDIA Jensen Huang at CES 2025

Inference Scenario 1: Online ChatGPT/Copilot

What are some fun places to visit at Urbana Champaign?

Searching for: **fun places to visit at Urbana Champaign**

Generating answers for you...

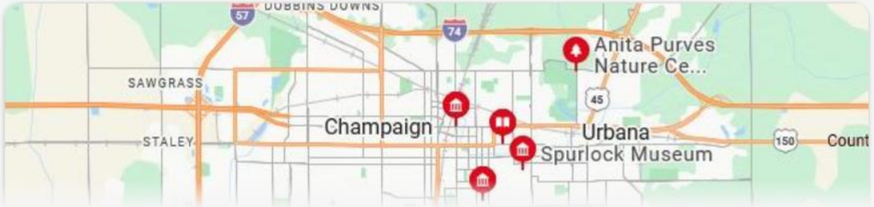
There are many fun places to visit at Urbana Champaign, depending on your interests and preferences. Here are some of the most popular ones:

- If you love nature and animals, you might enjoy visiting the **Anita Purves Nature Center** [1], where you can explore trails, feed goats, and learn about local wildlife.
- If you are into art and culture, you might want to check out the **Krannert Art Museum** [2], which has a diverse collection of artworks from various regions and periods, as well as exhibitions and events.
- If you are looking for some entertainment and history, you might like the **Virginia Theatre** [3], which is a restored historic venue that hosts concerts, movies, and shows.
- If you are feeling adventurous and sporty, you might have fun at the **University of Illinois Ice Arena**, where you can skate, play hockey, or watch games.

These are just some of the fun places to visit at Urbana Champaign. You can find more information and reviews on Tripadvisor or Bing. I hope you have a great time exploring the city! 😊

Learn more 1 experiencecu.org 2 kam.illinois.edu 3 bing.com

2 of 30



Inference Scenario 2: Online Image Generation

DALL·E 3 · An expressive oil painting of a basketball player dunking, depicted as an explosion of a nebula.

DALL·E 3

An ink sketch style illustration of a small hedgehog holding a piece of watermelon with its tiny paws, taking little bites with its eyes closed in delight.

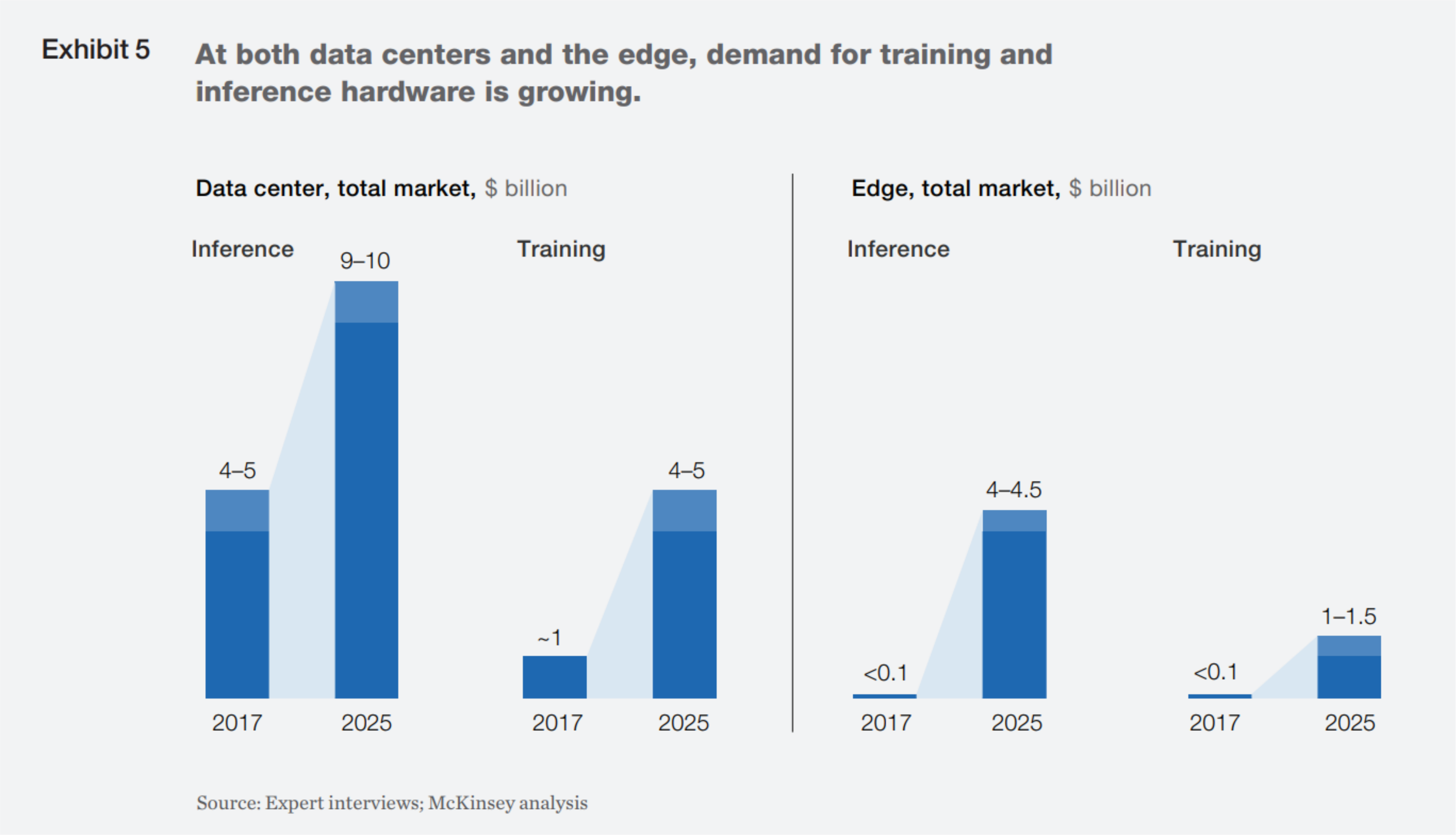
[DALL·E 3 | OpenAI]

Inference Scenario 3: Online Video Generation

Prompt: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

[Sora: Creating video from text]

Training -> Inference



Training vs. Inference

	Training	vs	Inference
Runtime	Weeks or months		Milliseconds or seconds
Challenges	TCO (Cost, Energy)		TCO (Cost, Energy)

Speed (LLM: token rates)

Model size

- Parameter volume
- LLM: Context length

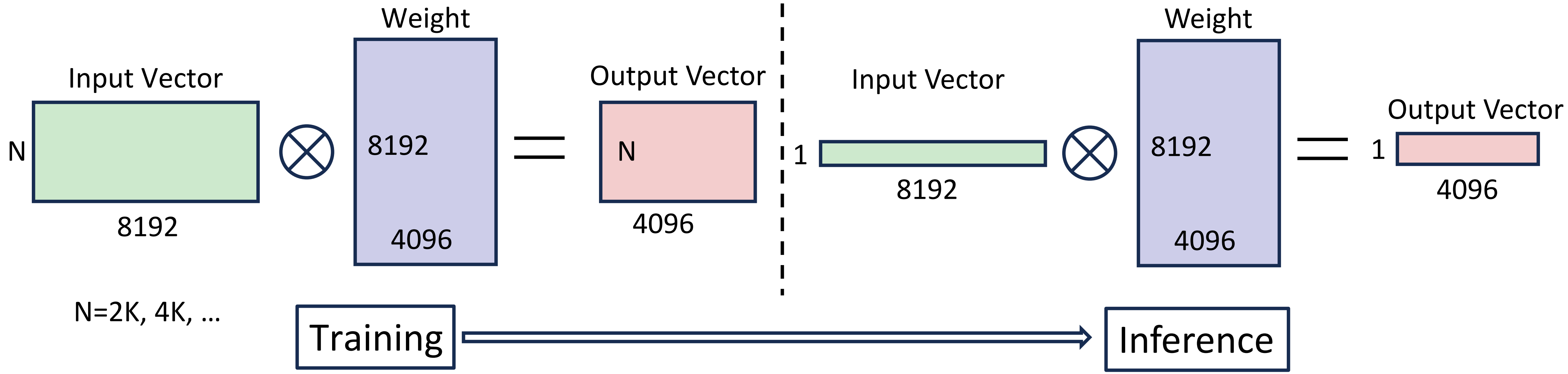
Inference Challenge 1: Long Latency Violates SLA

- Long serving latency blocks deployment
- Support advance models while meeting latency SLA and saving cost

DL Scenarios	Original Latency	Latency Target
Turing Prototype 2	~100ms	< 10ms
Turing Prototype 3	~107ms	< 10ms
Deep Query Document Similarity	10~12ms for [query, 1 doc] x 33 docs	< 6ms
Malta Click Features	10ms for [query, 1 passage] x 150 passages	< 5ms
Ads seq2seq model for query rewriting	~51ms	< 5ms

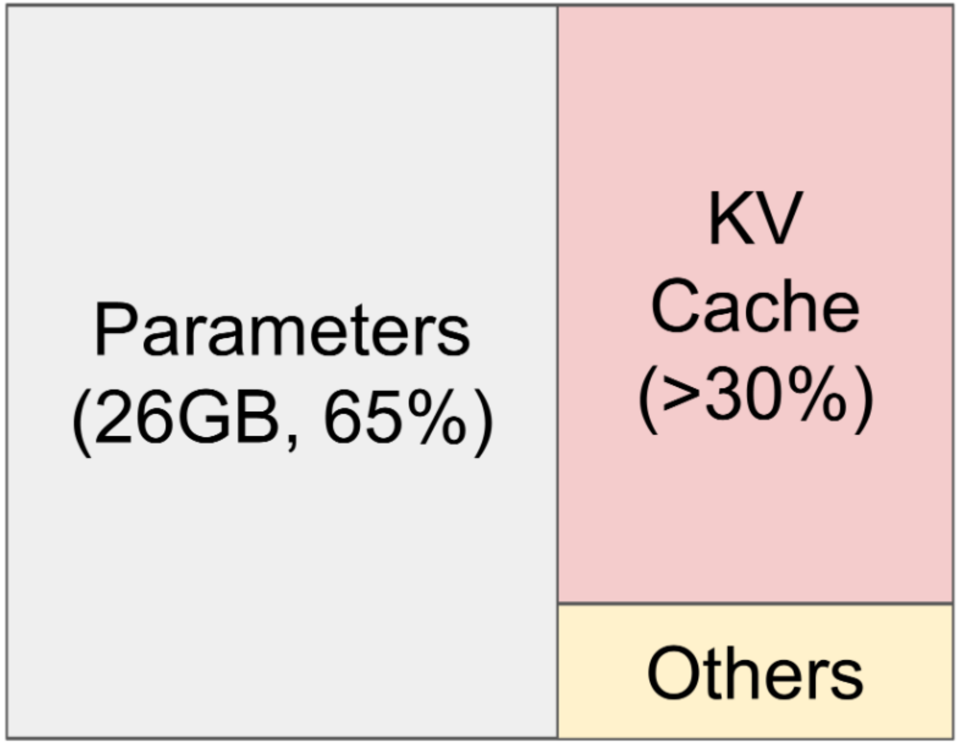
Inference Challenge 2: Small Batch Limits Parallelism

- Small batch size ⟹ Low data reuse
- Autoregressive generation ⟹ Sequential dependency



Inference Challenge 3: Large Memory Increases Cost

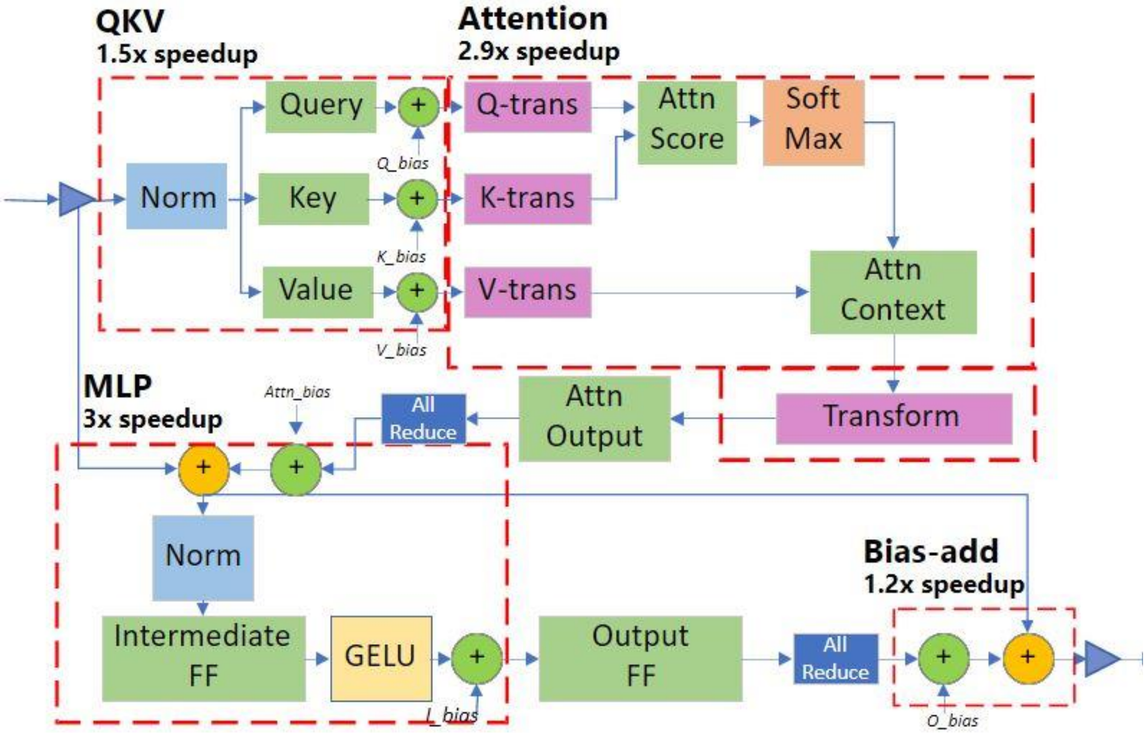
- Model parameters
  - # Layers
  - # Hidden dim
- KV cache
  - Batch size
  - Sequence length
  - # Layers
  - # Hidden
- Activation and others



OPT-13B on A100 40 GB

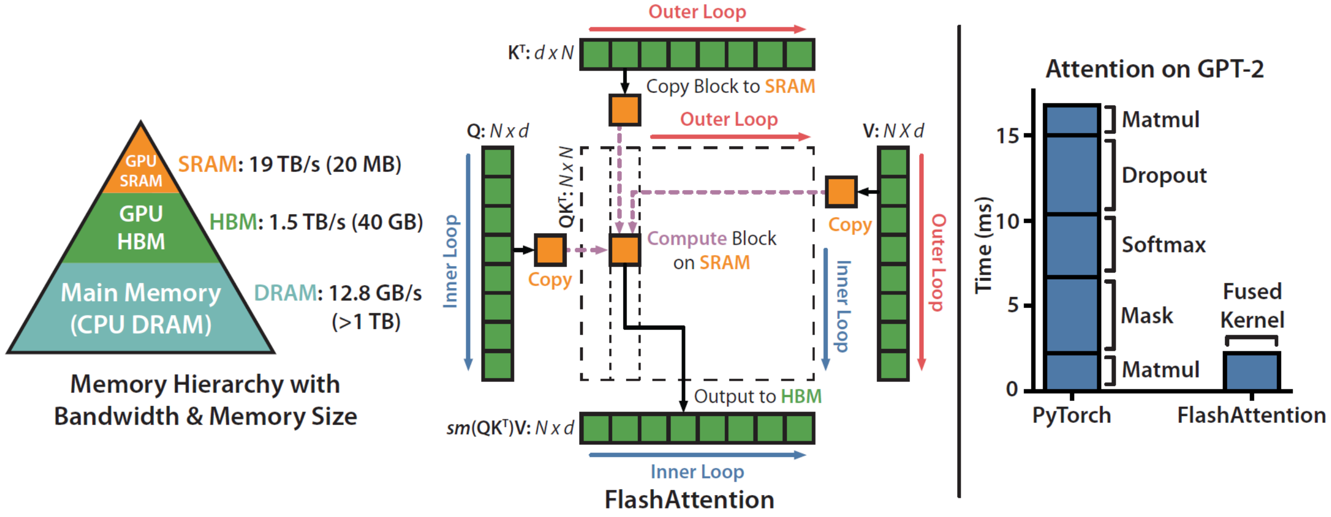
Efficient Memory Management for Large Language Model Serving with PagedAttention, by Kwon et al., 2023

Customized Kernels

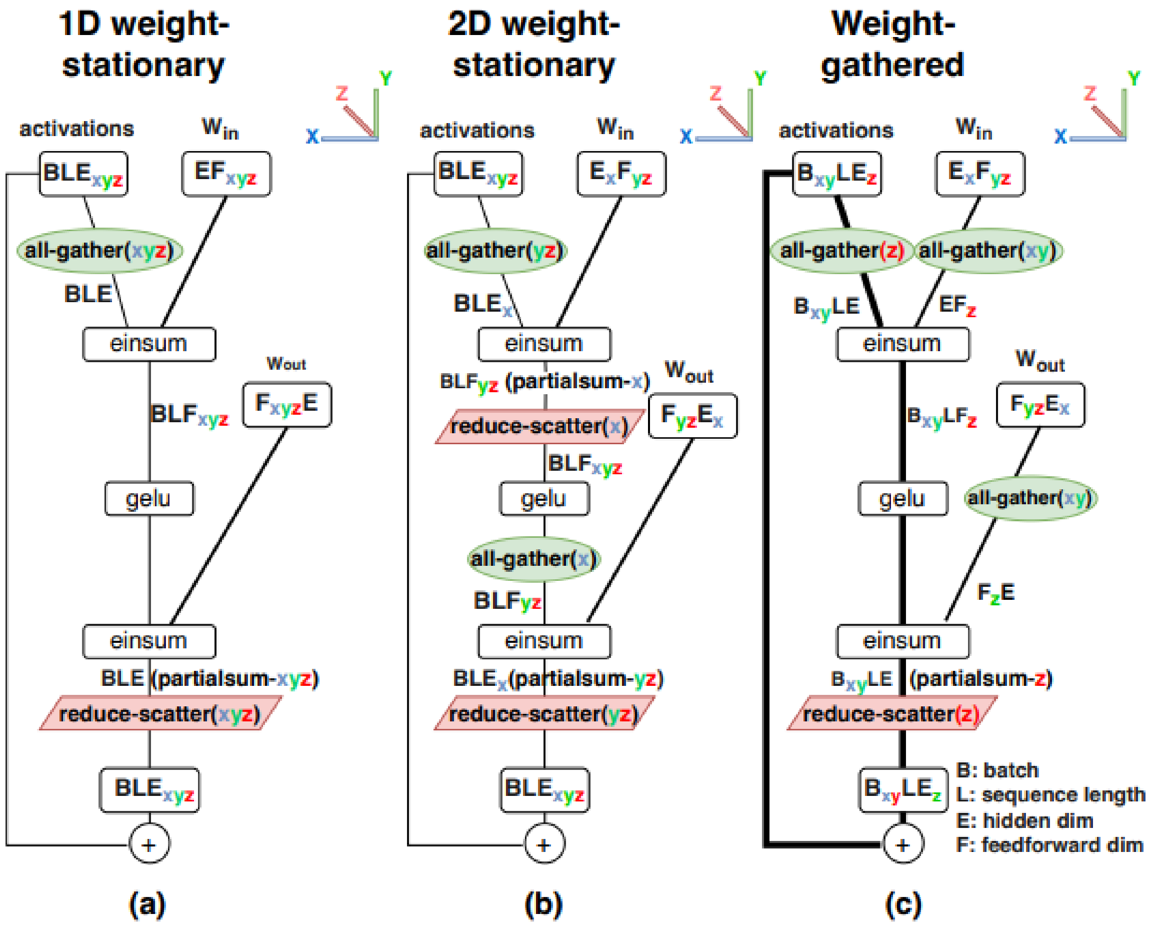


DeepSpeed-Inference: enabling efficient inference of transformer models at unprecedented scale, SC 2022

Fast and Memory-Efficient Exact Attention with IO-Awareness, 2023

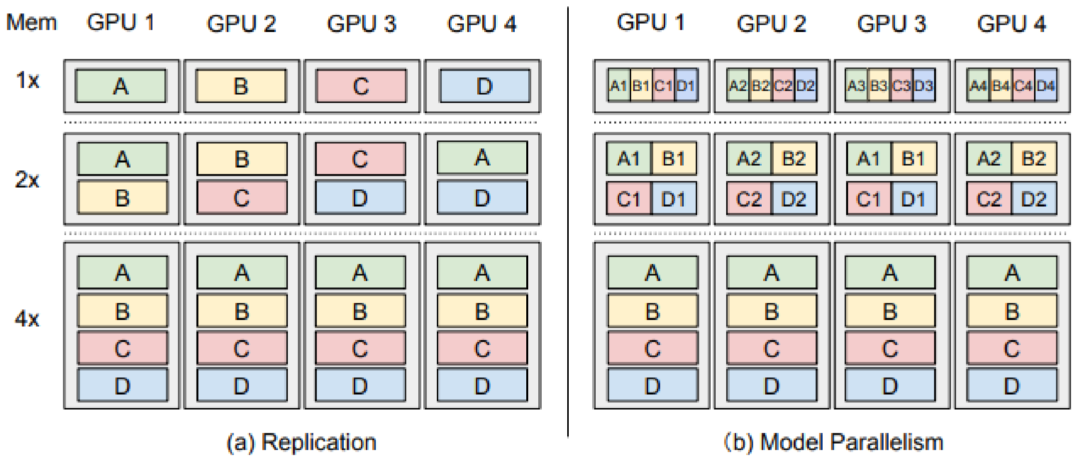


Multi-GPU Inference via Partitioned Layouts

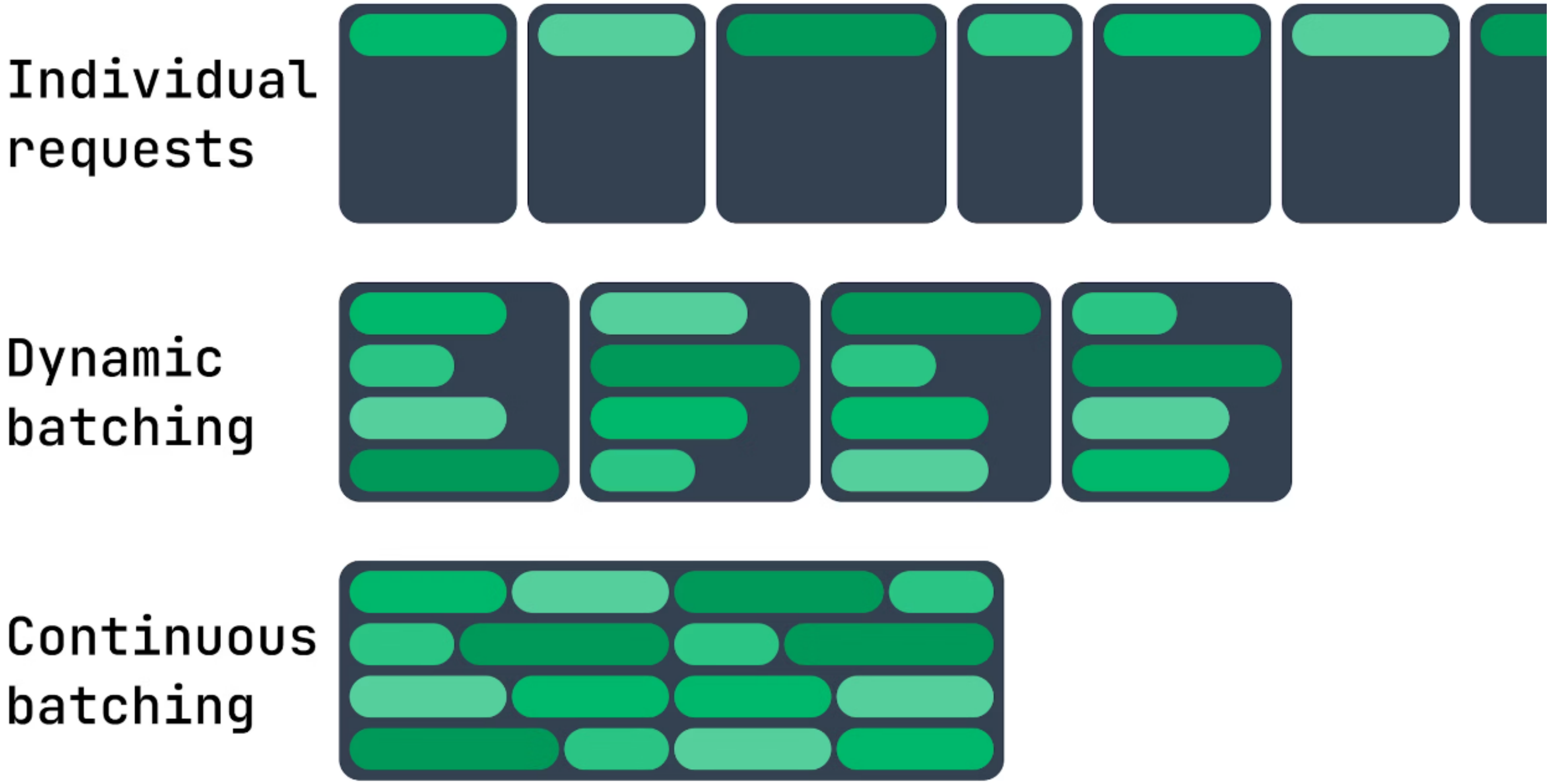


Efficiently Scaling Transformer Inference, MLSys 2023

AlpaServe: Statistical Multiplexing with Model Parallelism for Deep Learning Serving, OSDI 2023

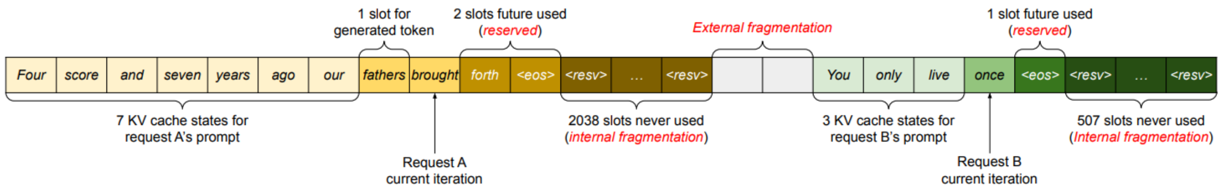


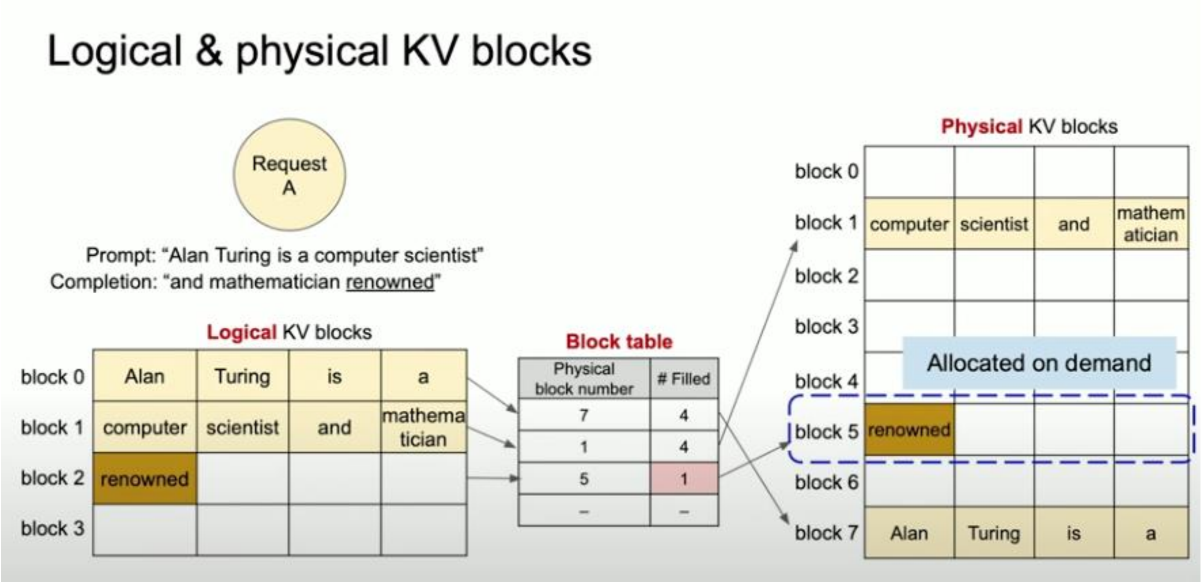
Scheduling Strategies for LLM Inference



Orca: A Distributed Serving System for Transformer-Based Generative Models, OSDI 2022

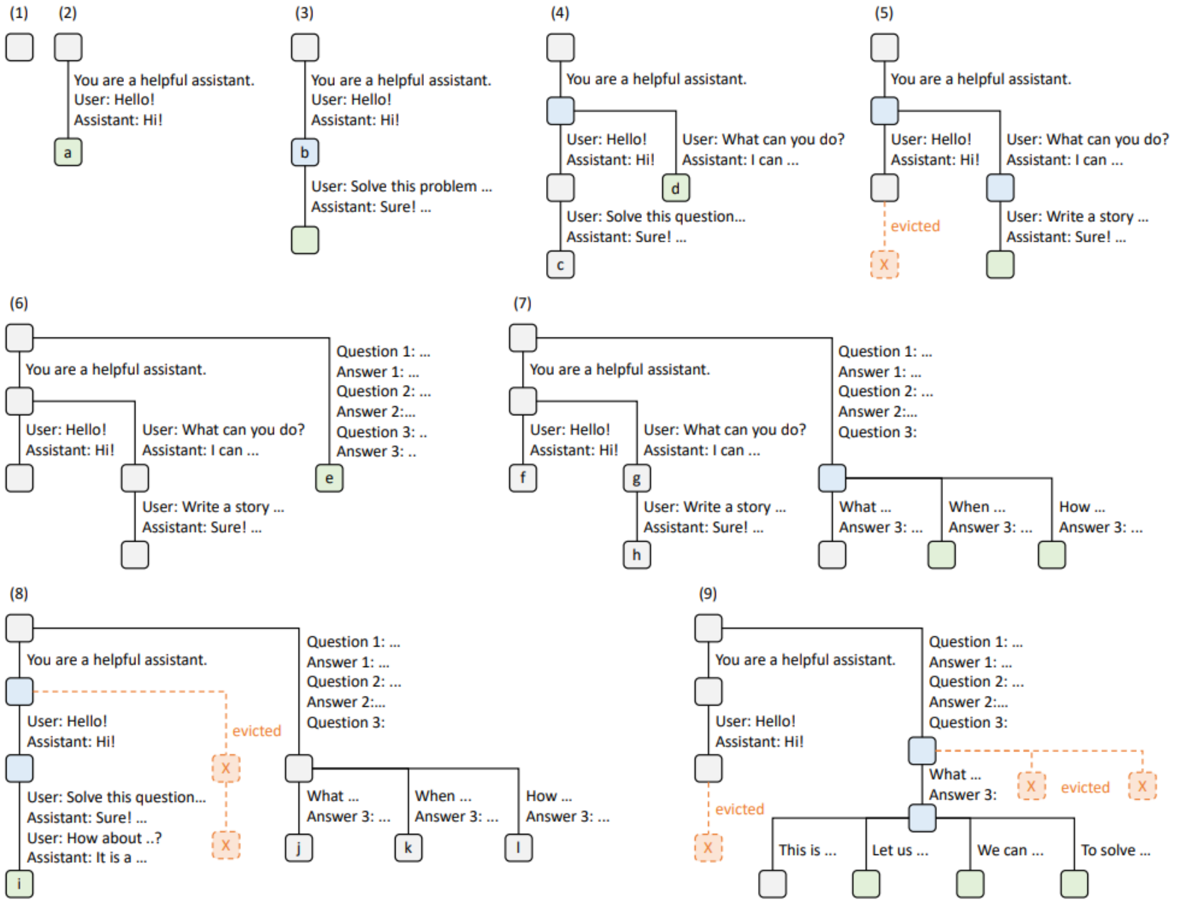
KV Cache Management





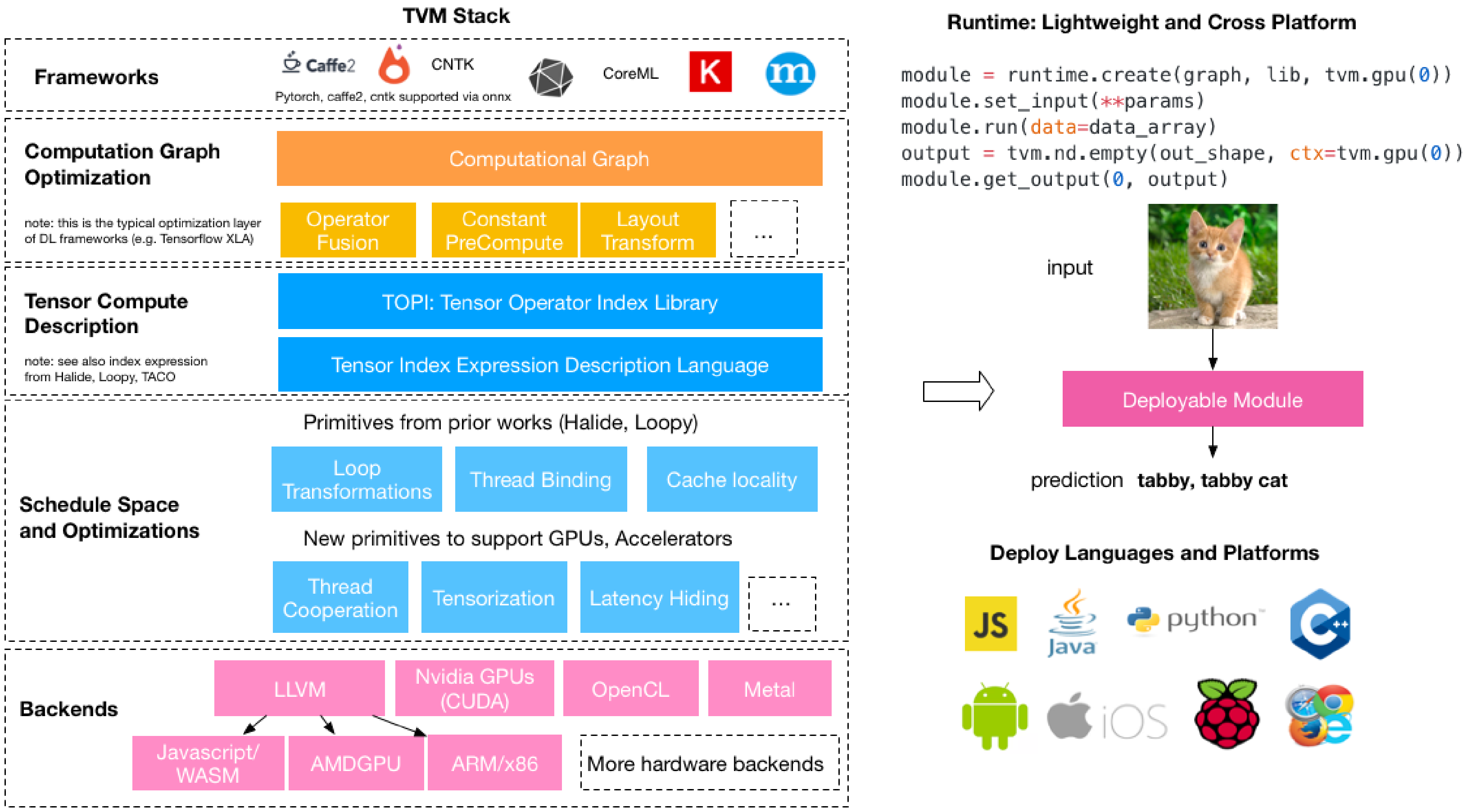
Efficient Memory Management for Large Language Model Serving with PagedAttention, 2023

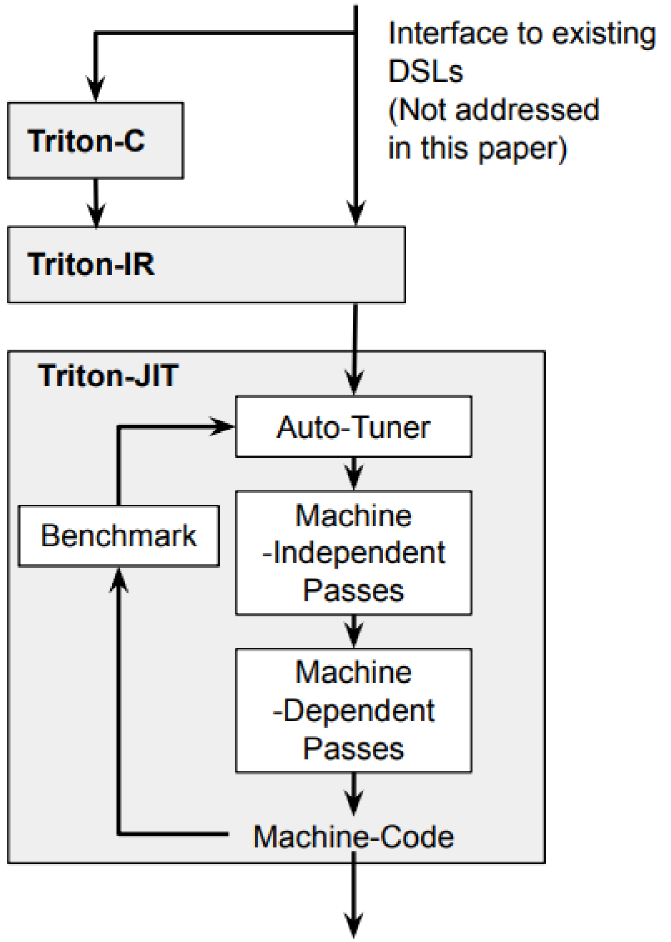
SGLang: Efficient Execution of Structured Language Model Programs, 2024



DL Compilation

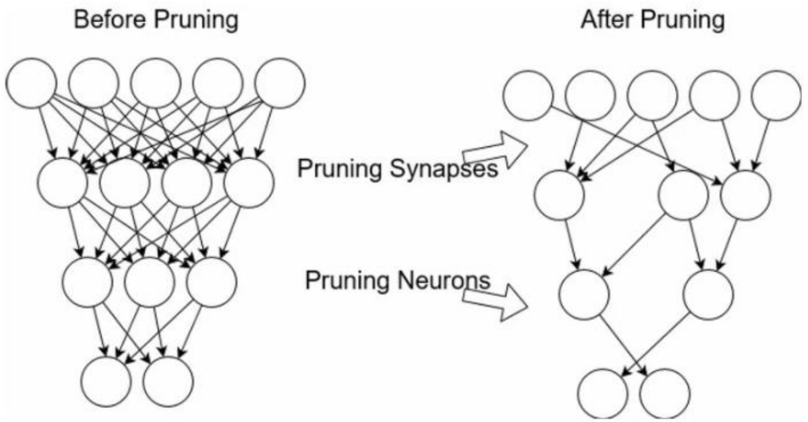
Triton: An Intermediate Language and Compiler for Tiled Neural Network Computations, 2019



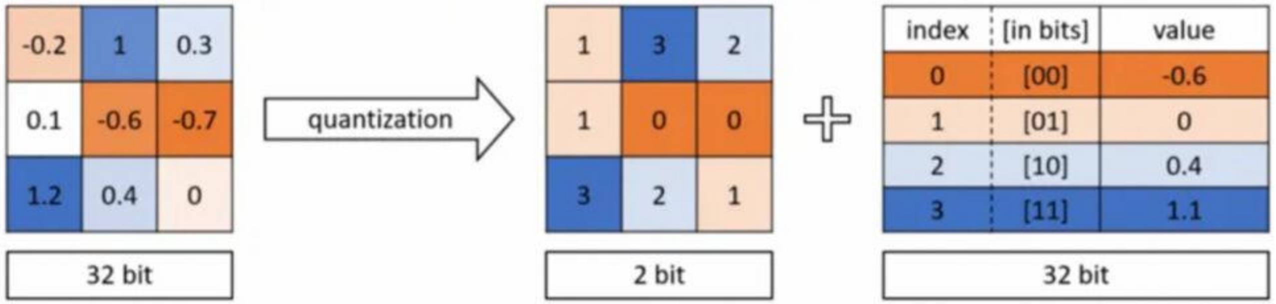


TVM: An Automated End-to-End Optimizing Compiler for Deep Learning, 2018

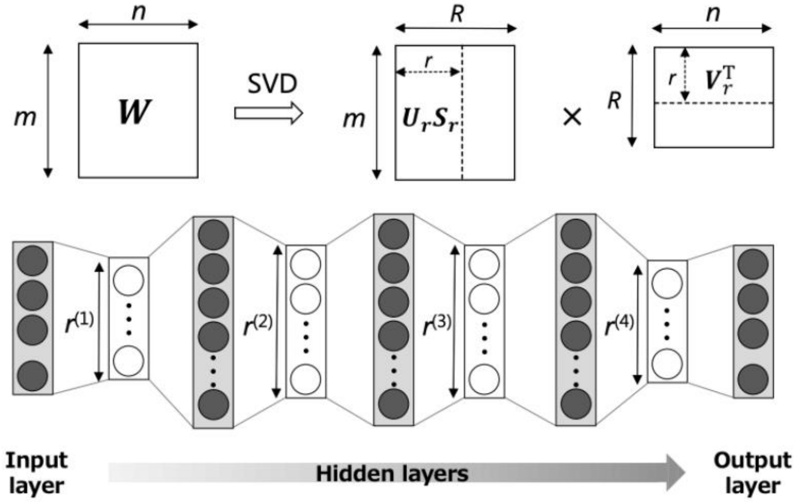
Compression Strategies



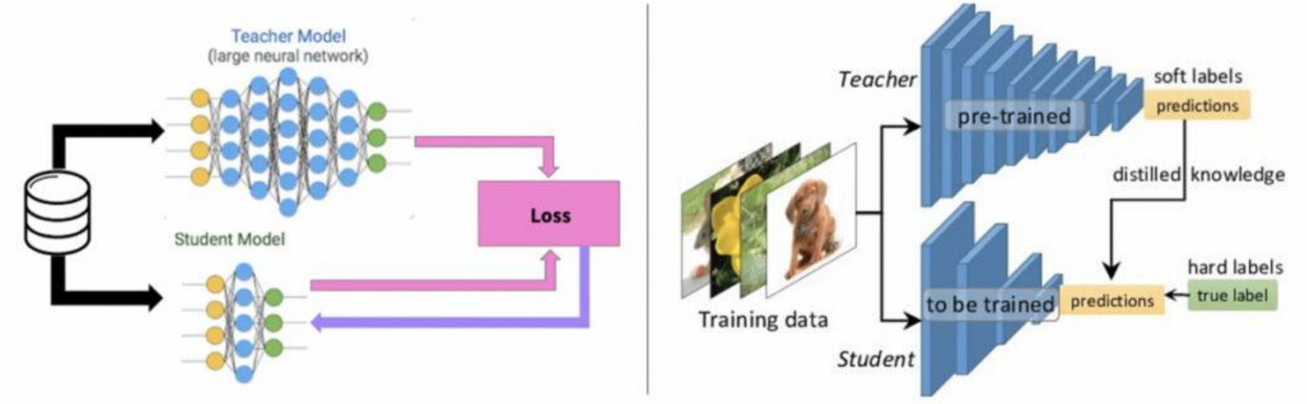
Pruning/Sparsification



Quantization



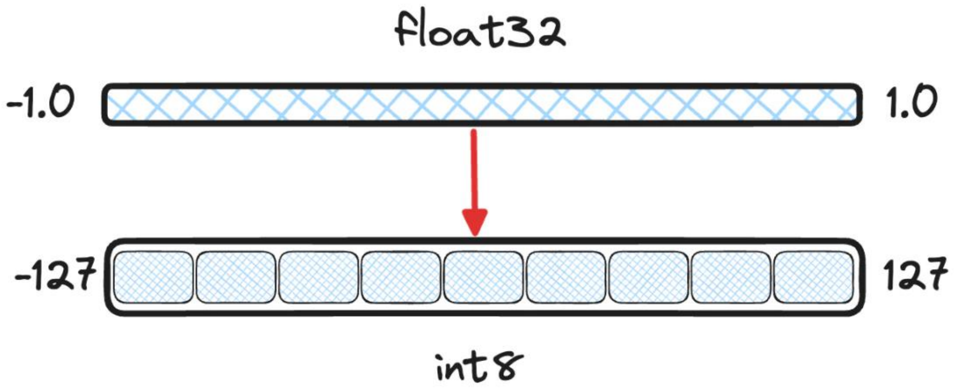
Low-rank decomposition



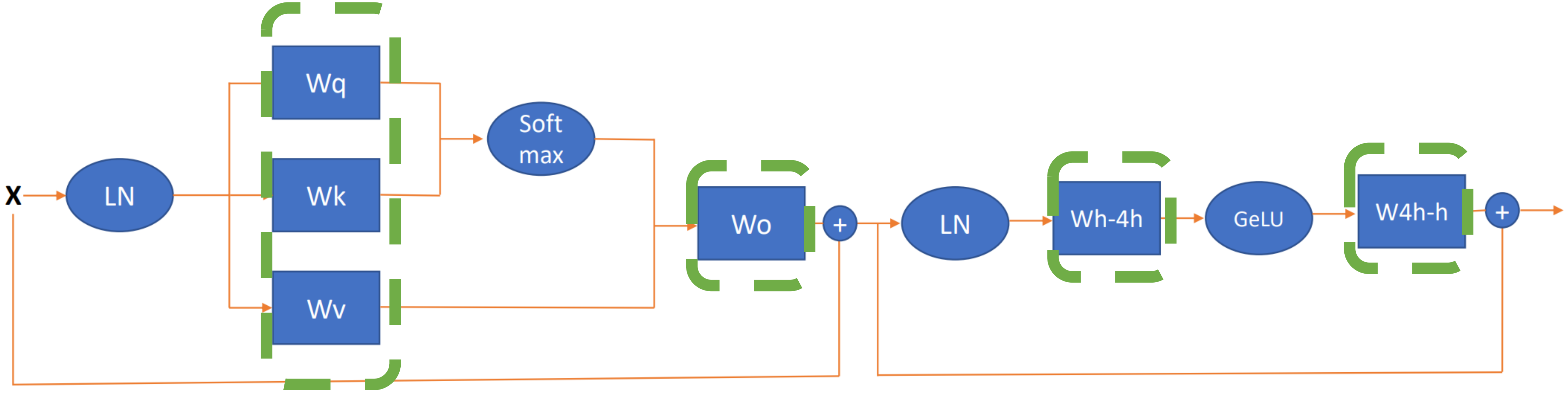
Distillation

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

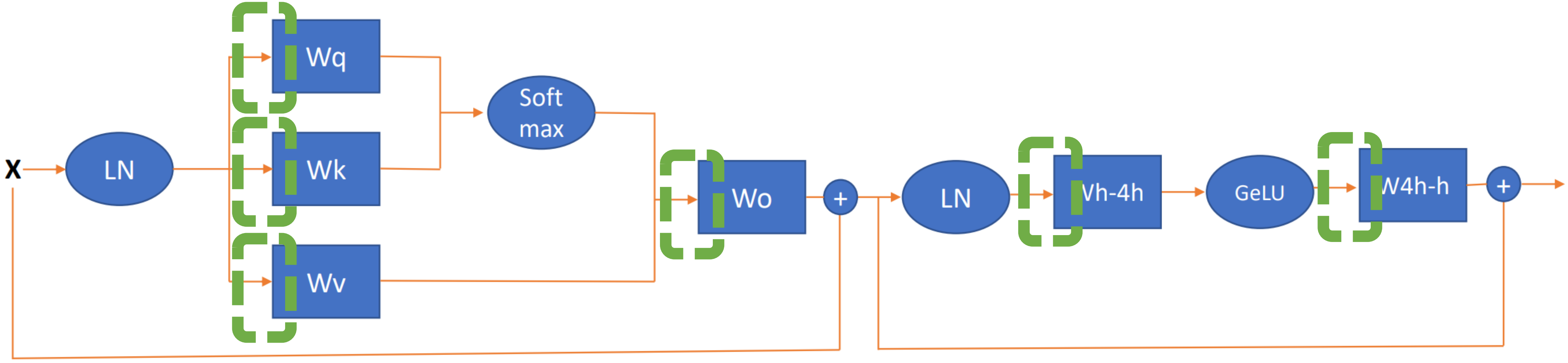
1.1	2.2	0.1	-0.1	-5.5	-6.6
…					
…					
…					
…					
1.1	2.1	0.1	-0.1	-4.8	-6.6

$\approx$ 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 $1/S$

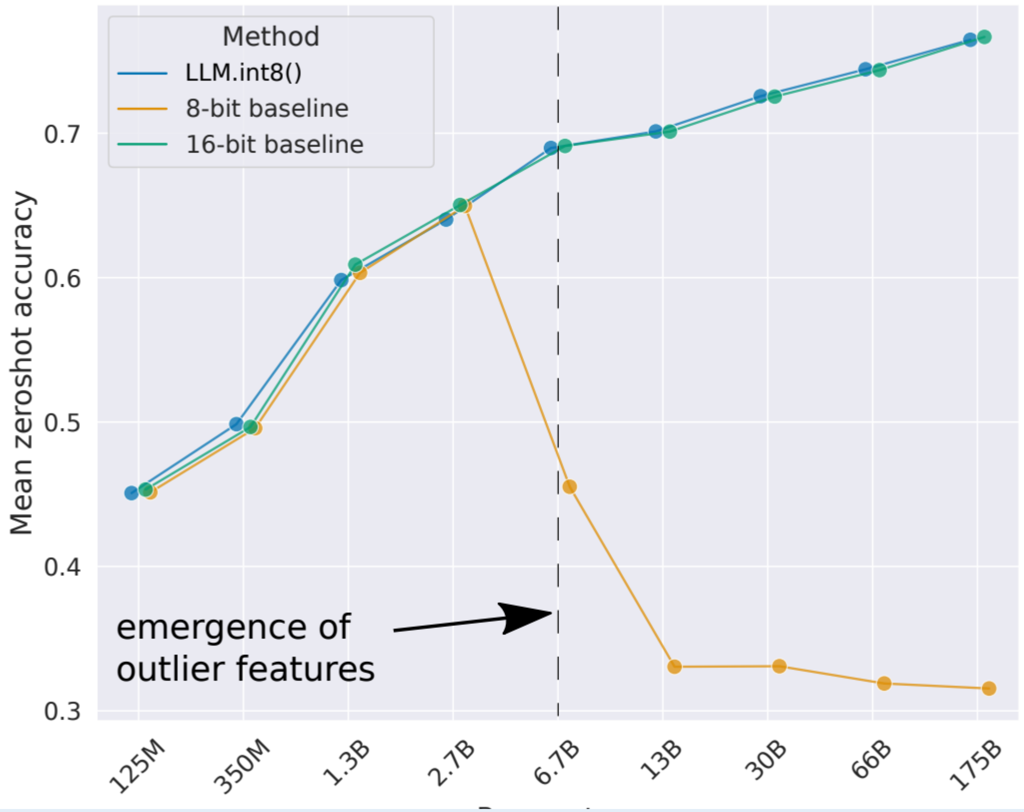
$\approx$ 0.05*

8-bit quantization

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

Challenges to Quantize LLMs

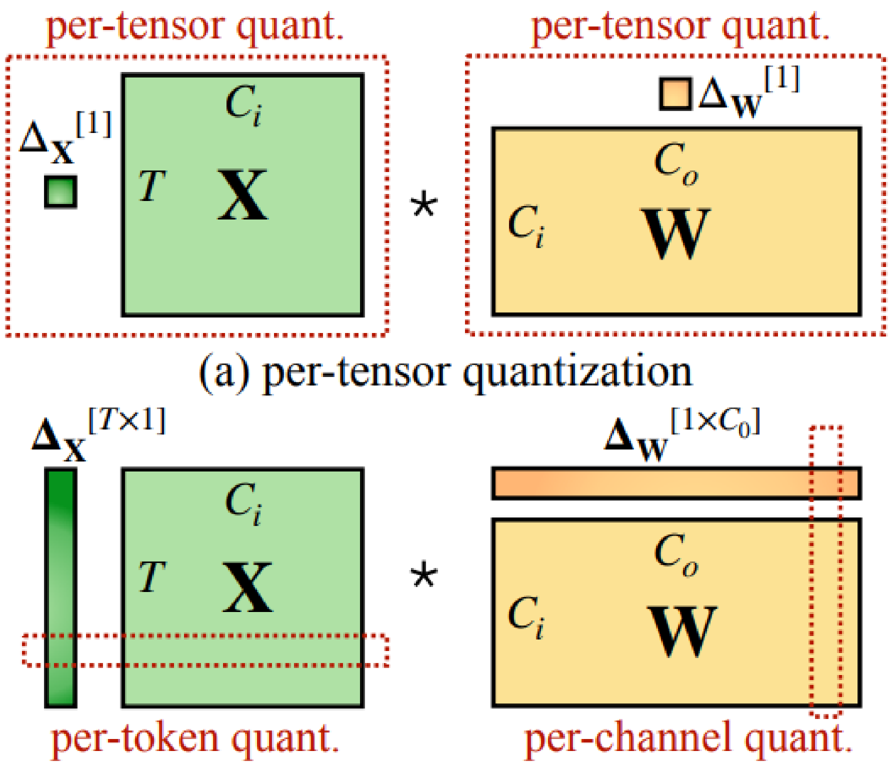
- Standard quantization strategy leads to catastrophic accuracy drop



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

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



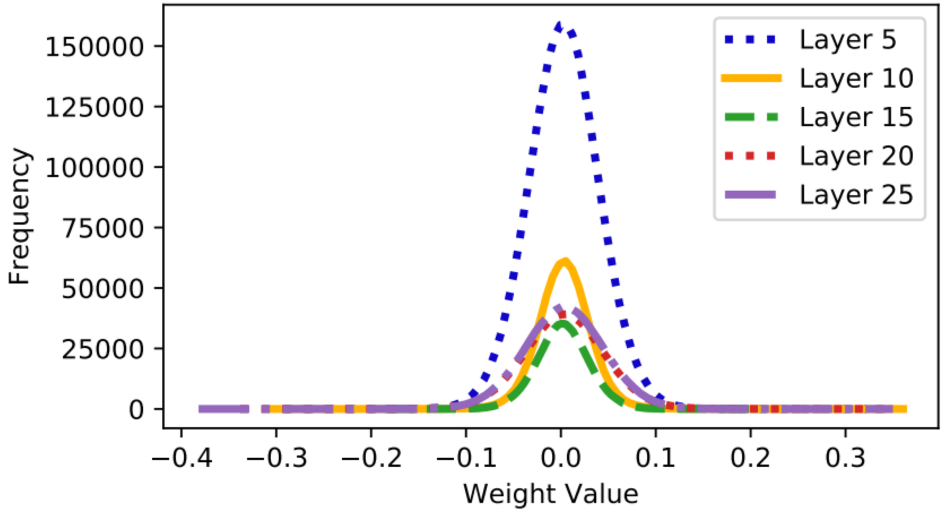
(a) per-tensor quantization

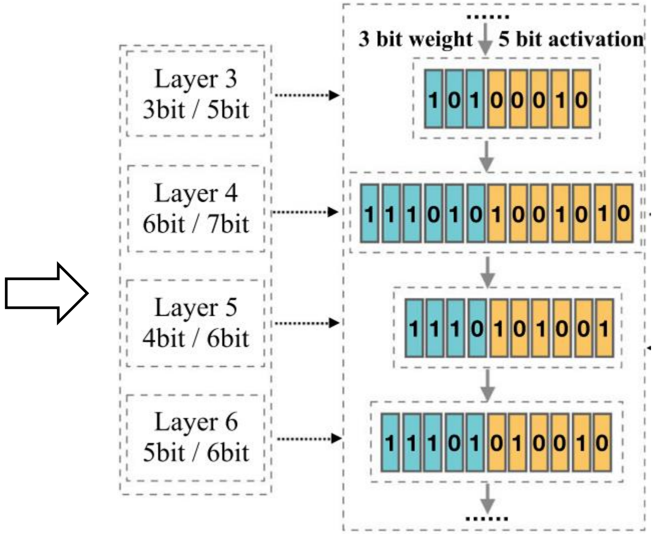
(b) per-token + per-channel quantization

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



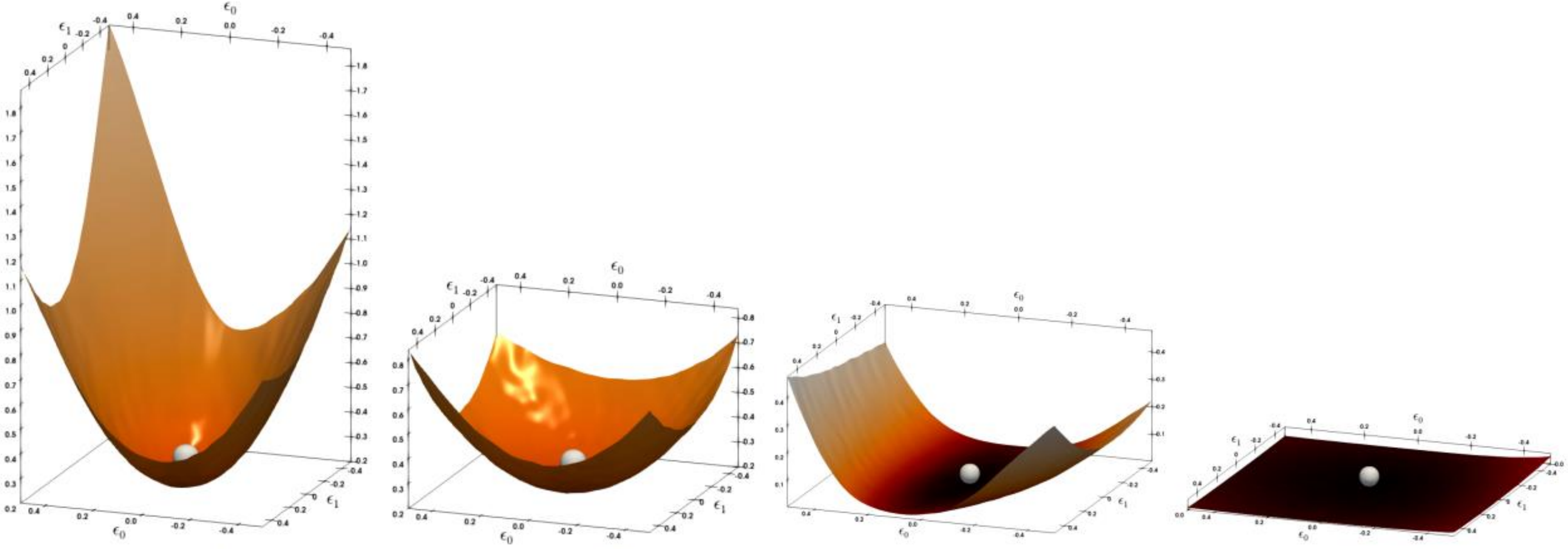


Per-layer weight distribution of BERT model

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

Second Order Information

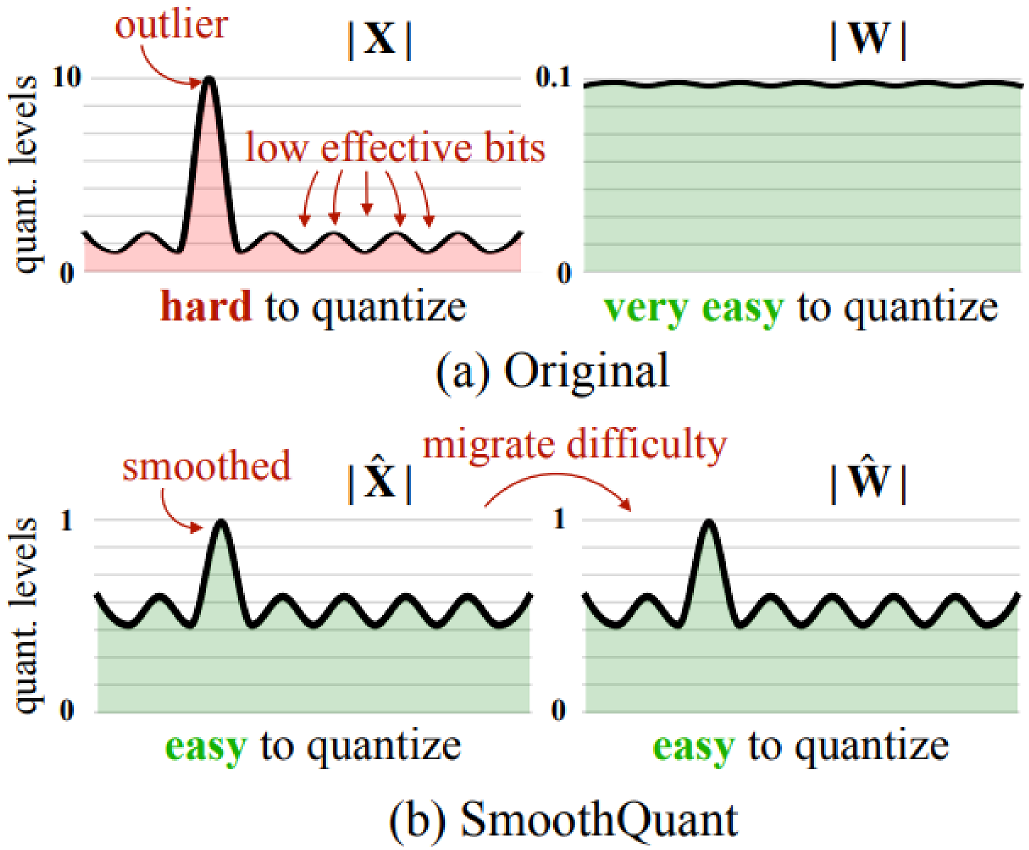
- Analyze the loss curvature (Hessian matrices) to help identify layer sensitivity



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

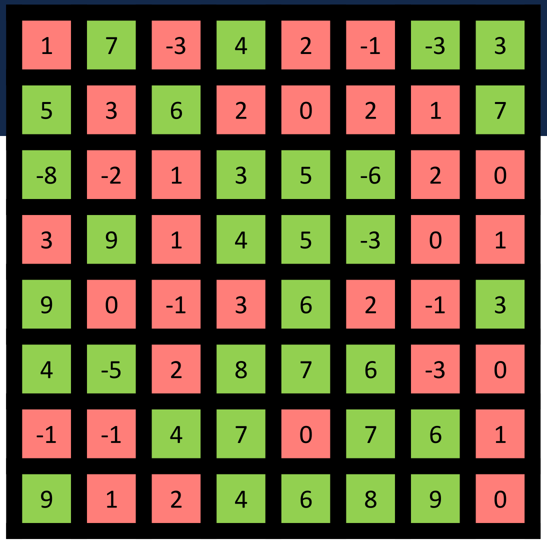
GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers, ICLR 2023

Outlier Smoothing



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

Sparsification

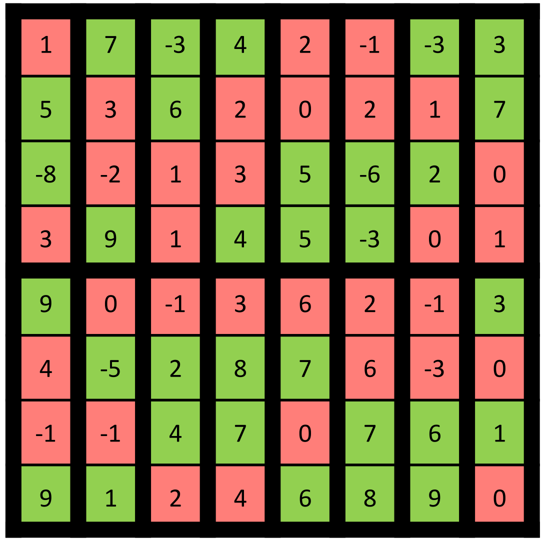


Unstructured Sparsity

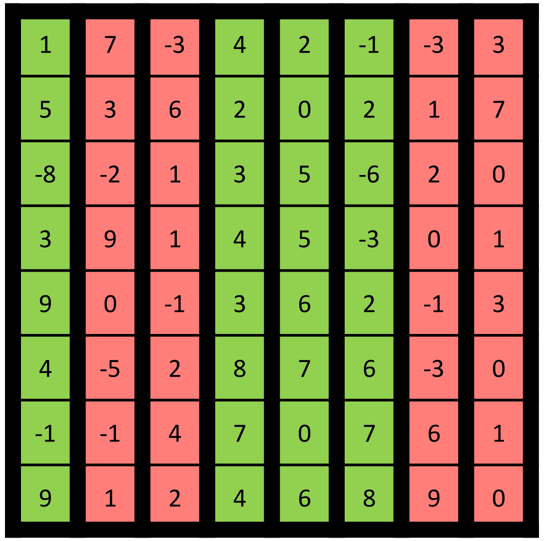
- **Unstructured (connection) Sparsity**:
- High accuracy
- No performance improvement or performance regression

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

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



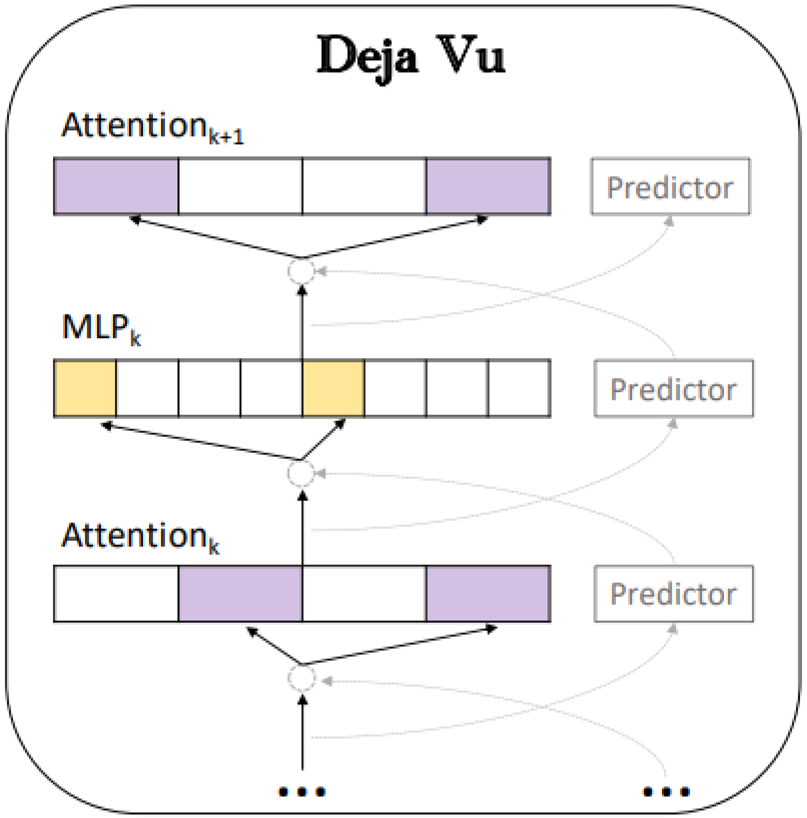
Semi-Structured Sparsity (4:2 N:M)



Structured Sparsity (Column-wise Sparsity)

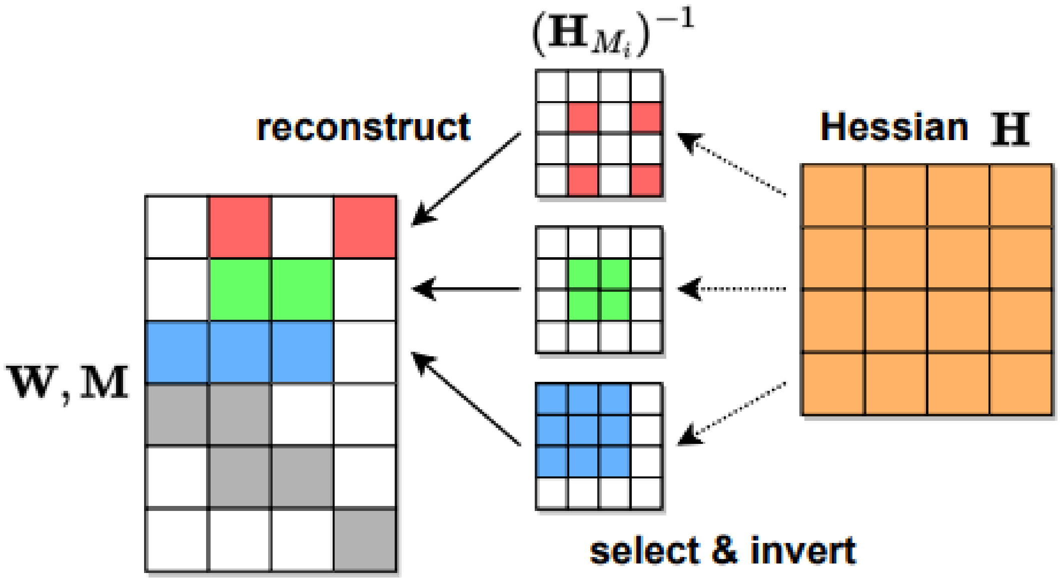
SliceGPT: Compress Large Language Models by Deleting Rows and Columns, ICLR 2024

Model Pruning



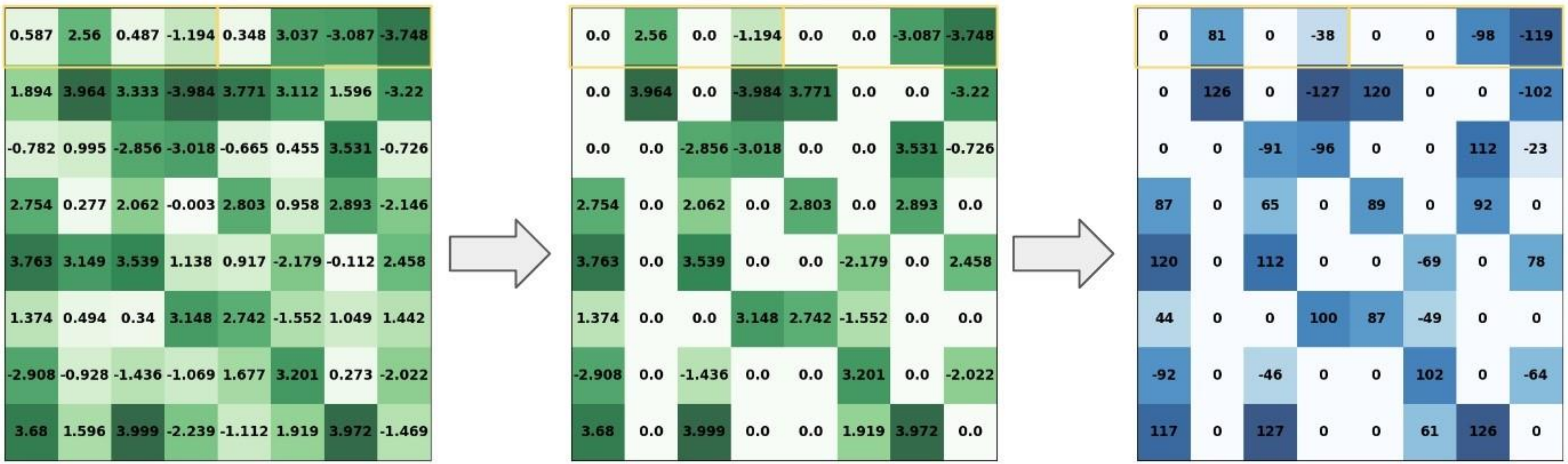
Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time, 2023

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



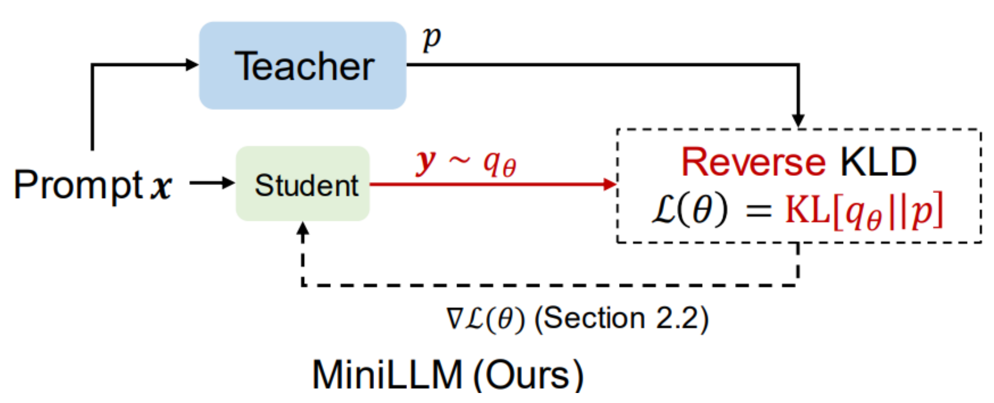
Sparsification + Quantization

FP32 FP32 INT8

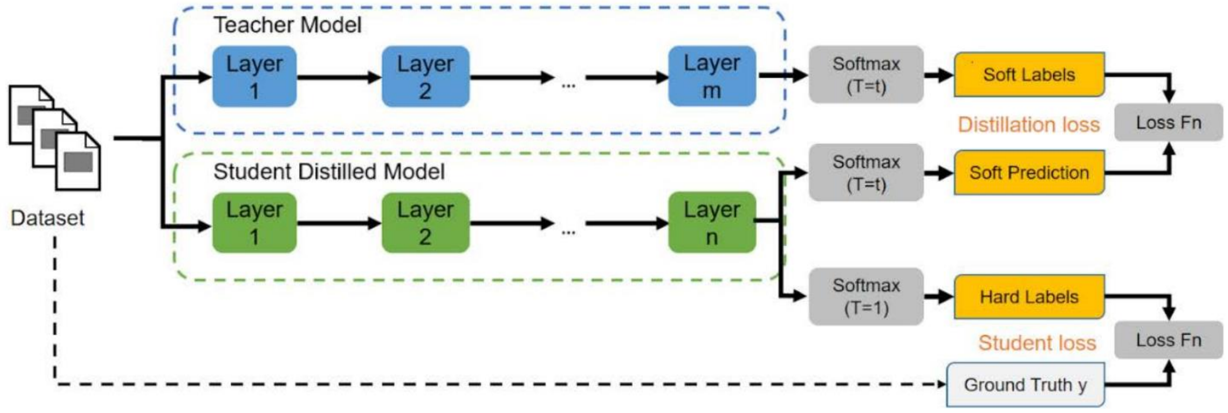


Dense Matrix Sparse Matrix Sparse-Quantized Matrix

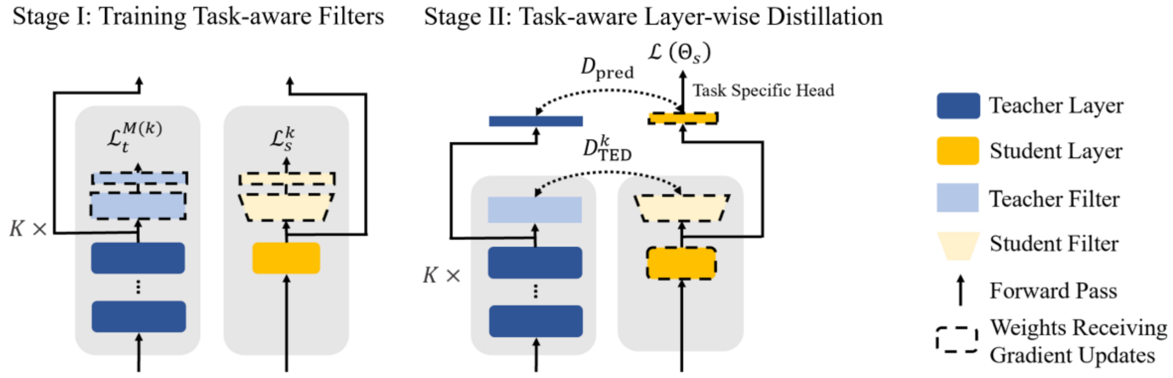
Knowledge Distillation



MiniLLM: Knowledge distillation of large language models, 2024

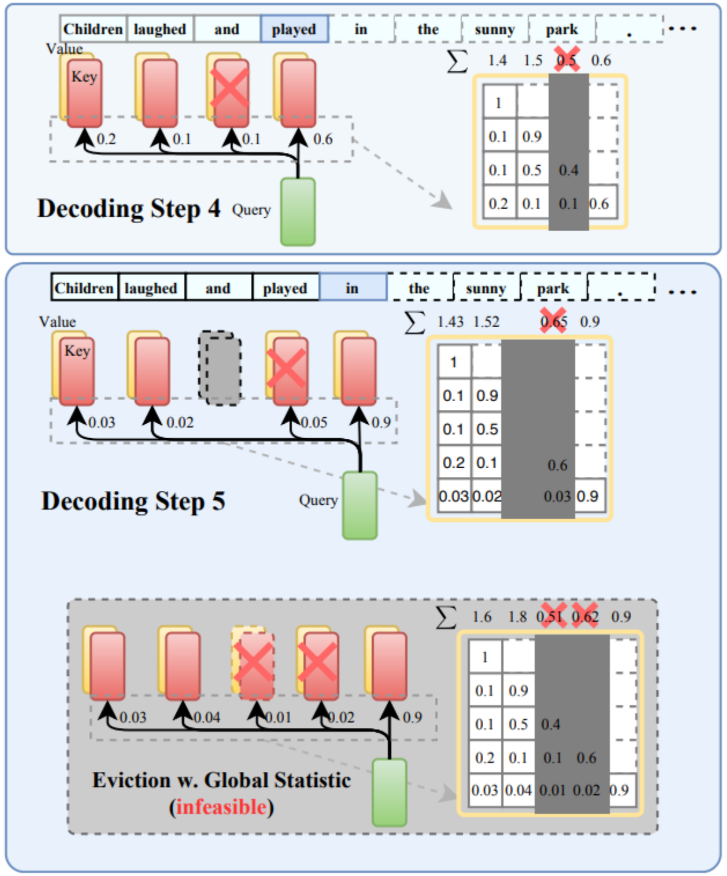


Distilling the Knowledge in a Neural Network , 2015



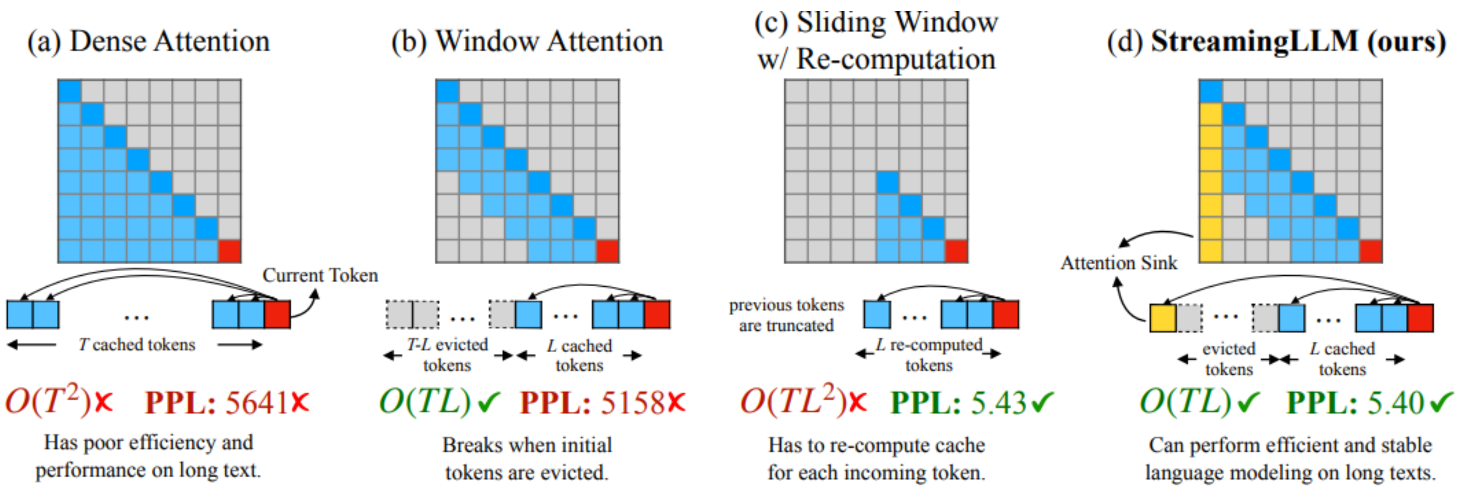
Less is More: Task-aware Layer-wise Distillation for Language Model Compression, 2024

KV Cache Compression

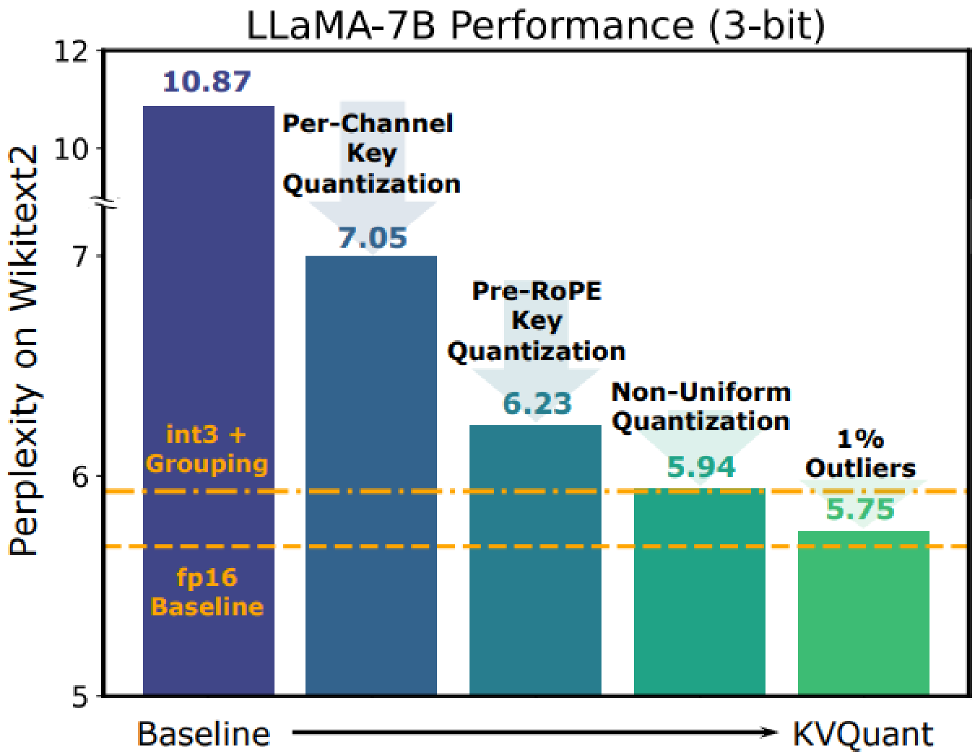


H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models, 2023

Efficient Streaming Language Models with Attention Sinks, ICL 2024

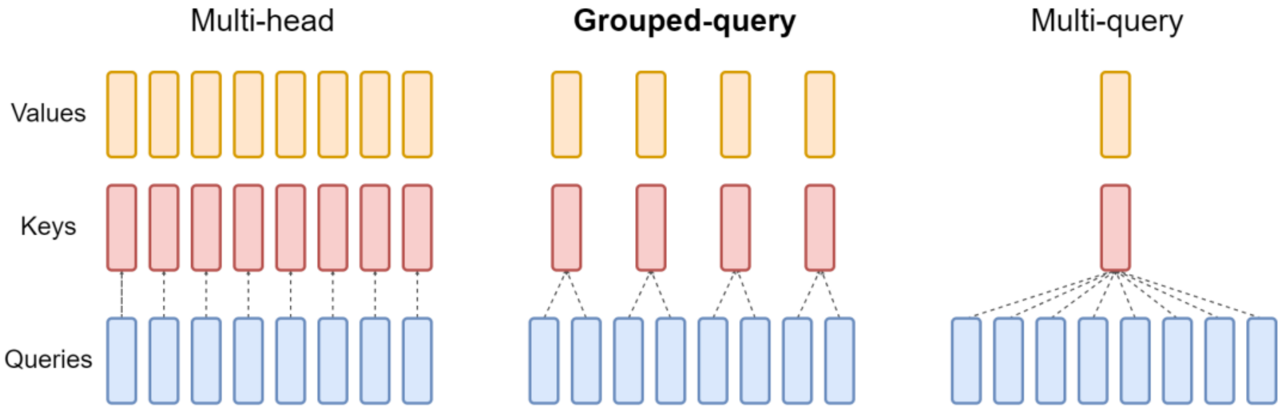


KV Cache Compression

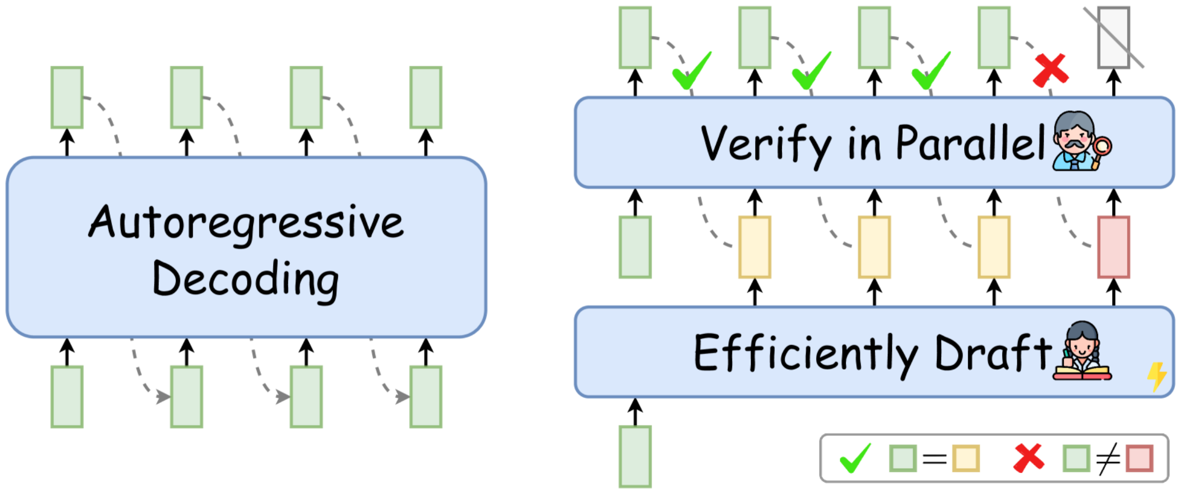


KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization, 2024

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

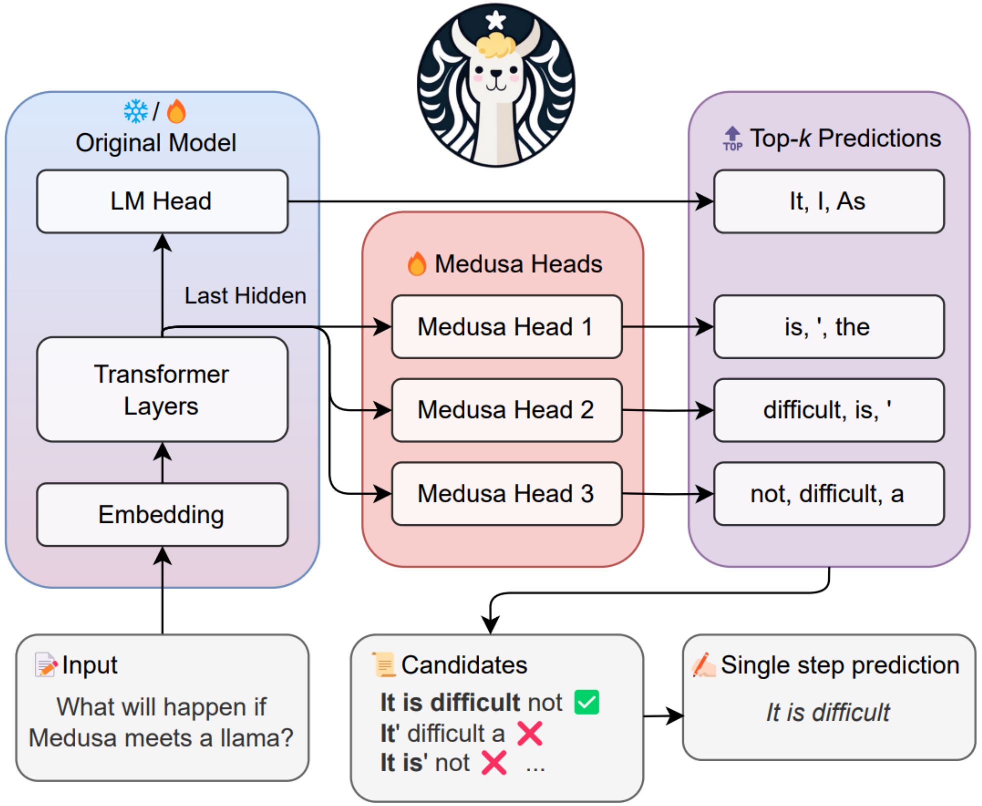


Speculative/Parallel Decoding

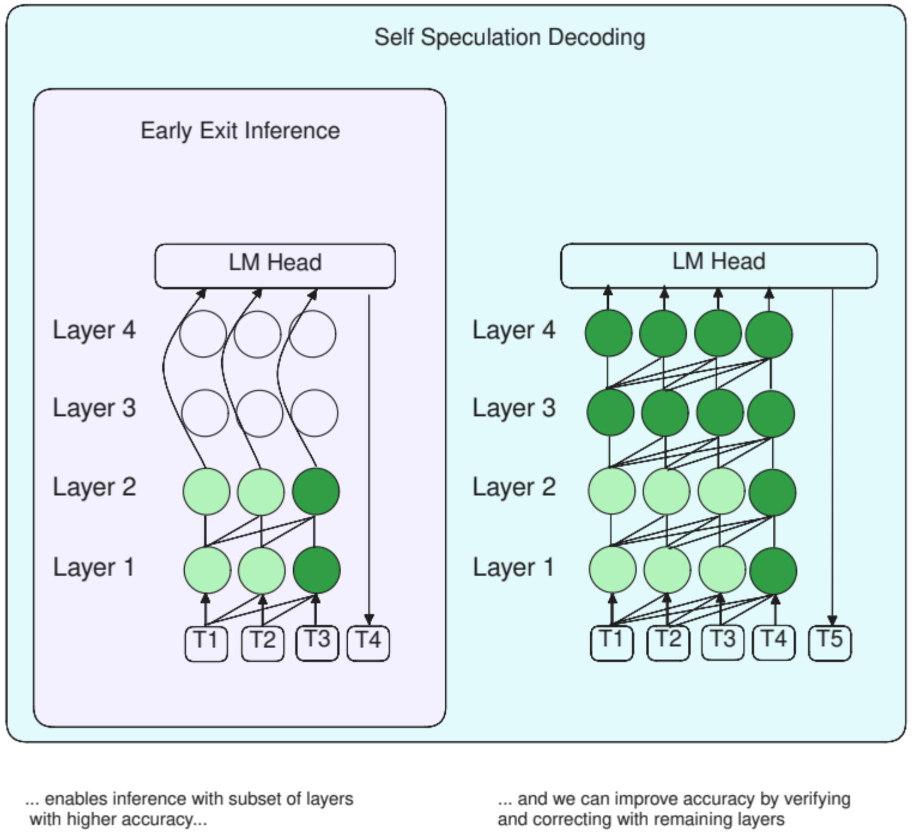


Fast Inference from Transformers via Speculative Decoding, 2023

MEDUSA: Simple LLM Inference Acceleration Framework with Multiple, 2024

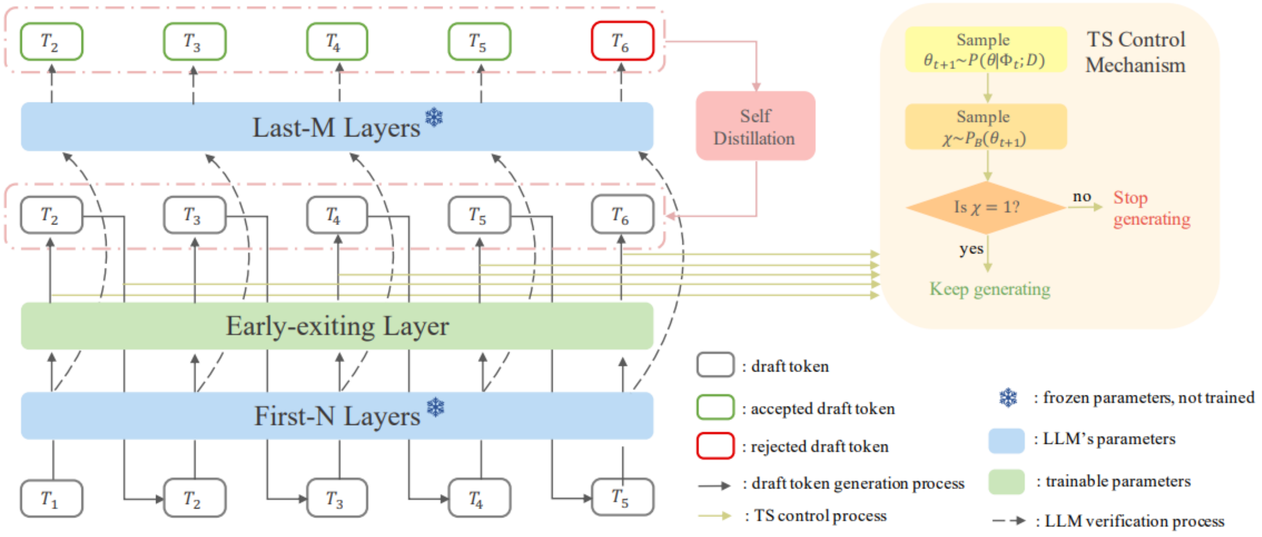


Early-Exit Inference

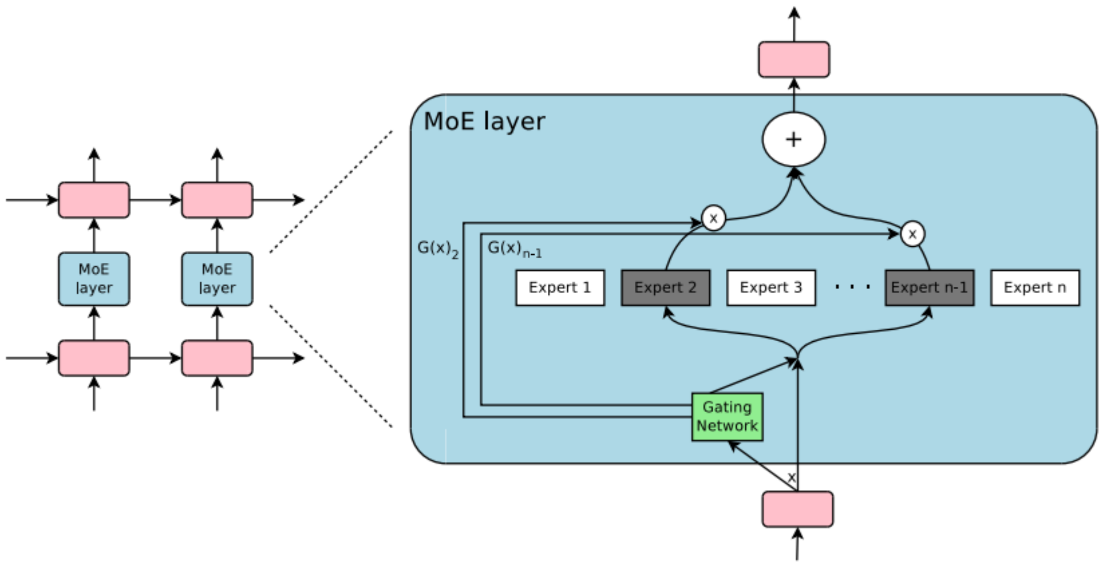


LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding, 2024

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

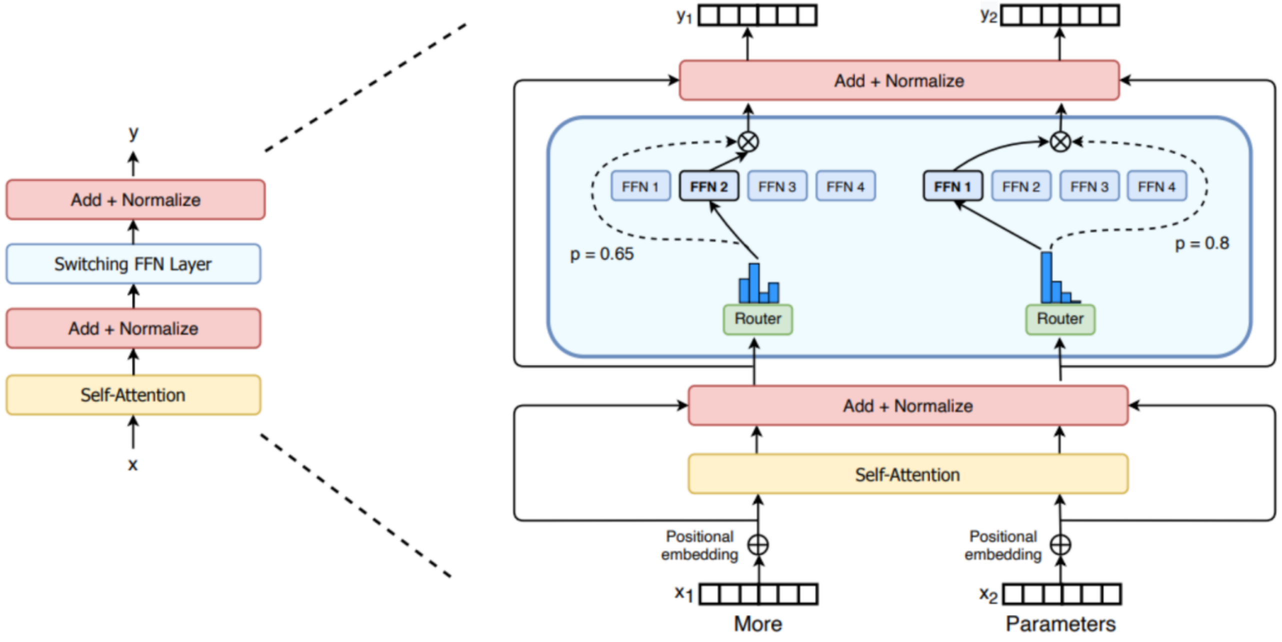


Mixture-of-Expert Models are Sparse and Need Less Compute

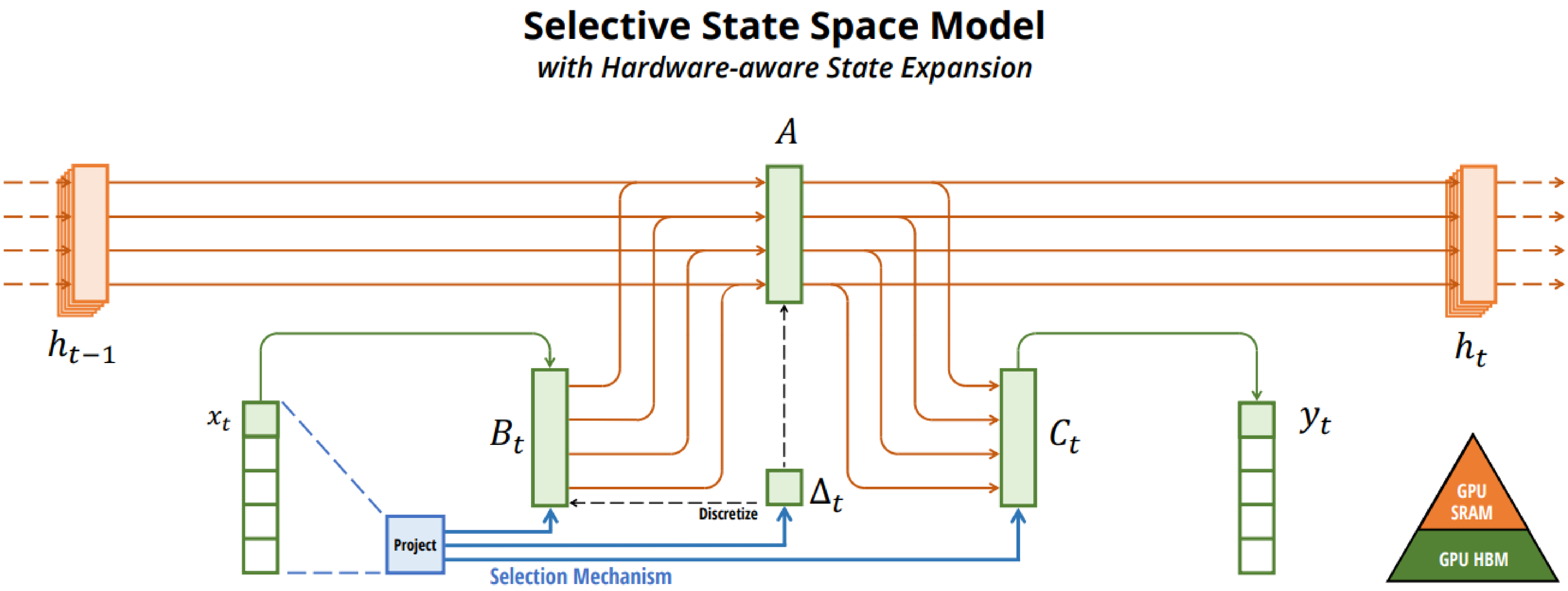


Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, 2017

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

Questions?