



CS 498: Machine Learning System Spring 2025

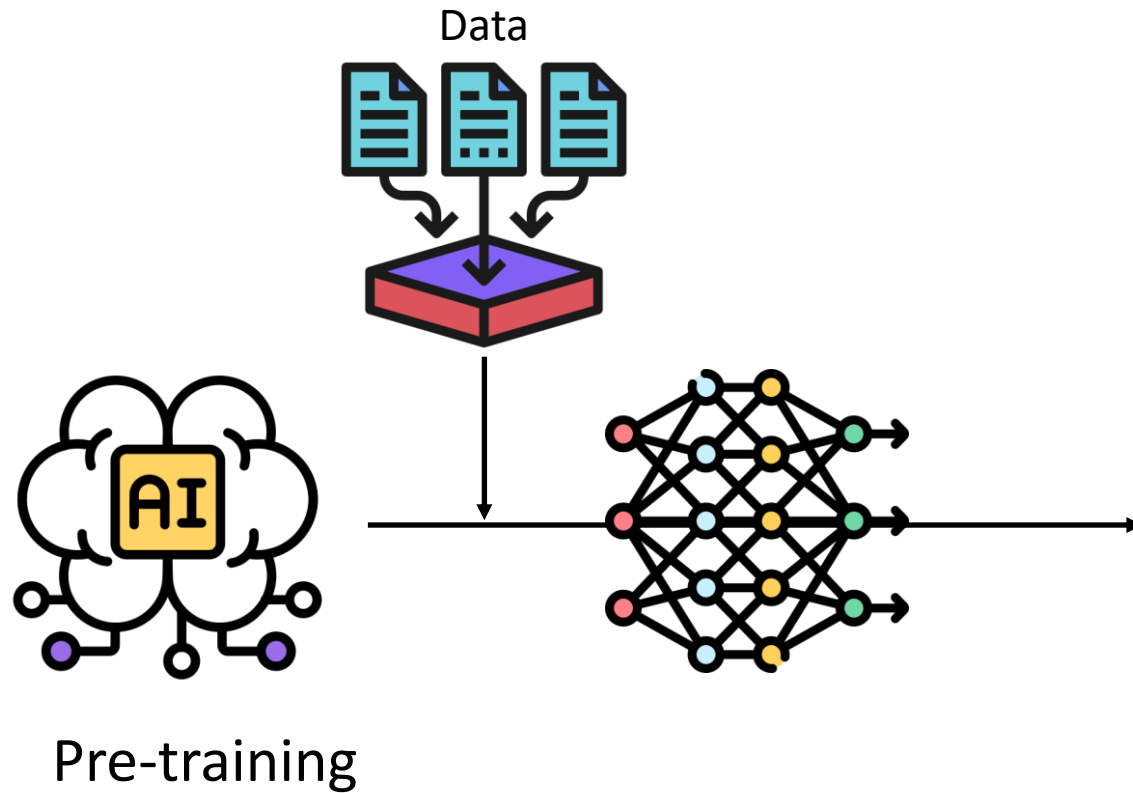
Minjia Zhang

The Grainger College of Engineering

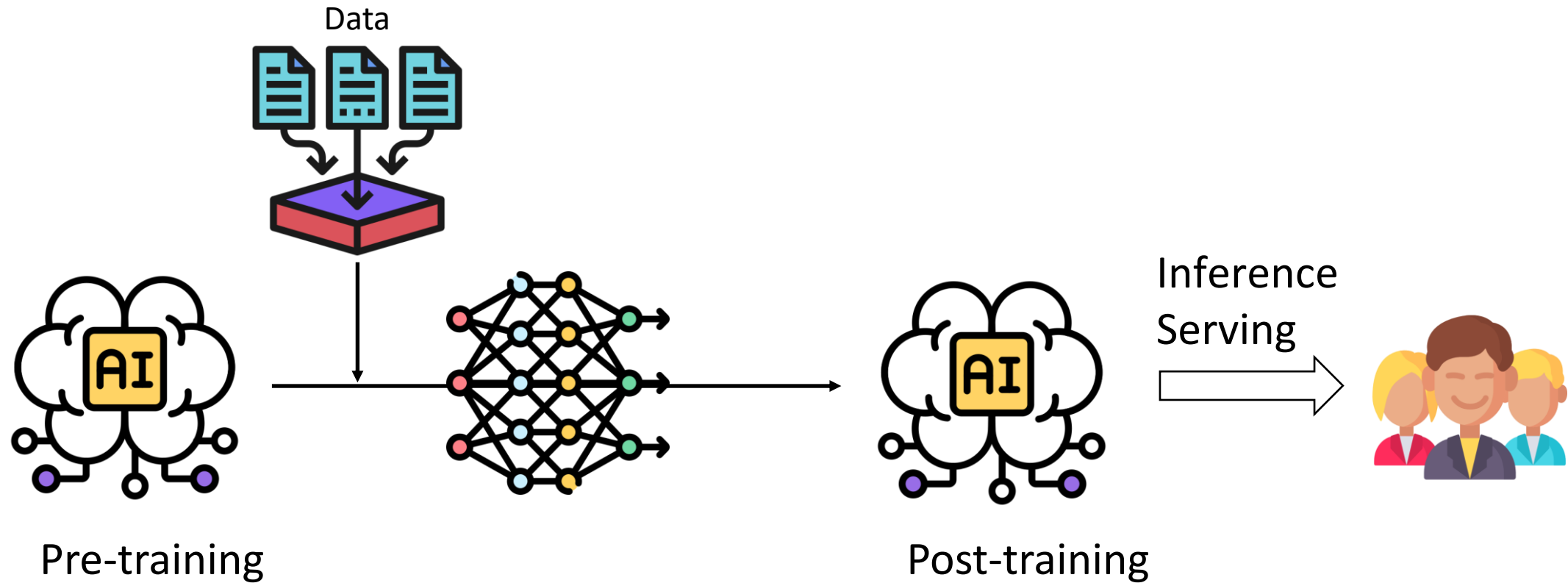
DL Inference Overview

- Background
- Challenges
- System optimizations
- Algorithm optimizations

What is Model Inference/Serving?



What is Model Inference/Serving?



Inference Scenario 1: ChatGPT/SearchGPT/Copilot



What are some fun places to visit at Urbana Champaign?

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

✓ Generating answers for you...

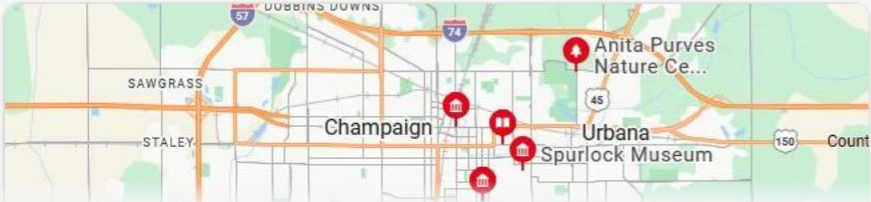
There are many fun places to visit at Urbana Champaign, depending on your interests and preferences. Here are some of the most popular ones:

- 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/Video Generation



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

[DALL-E 3 | OpenAI](#)



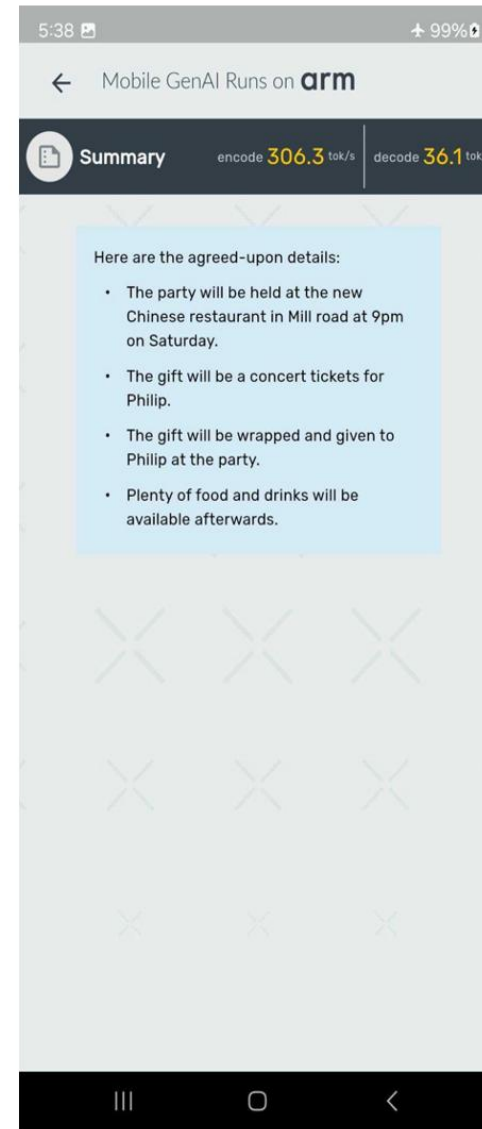
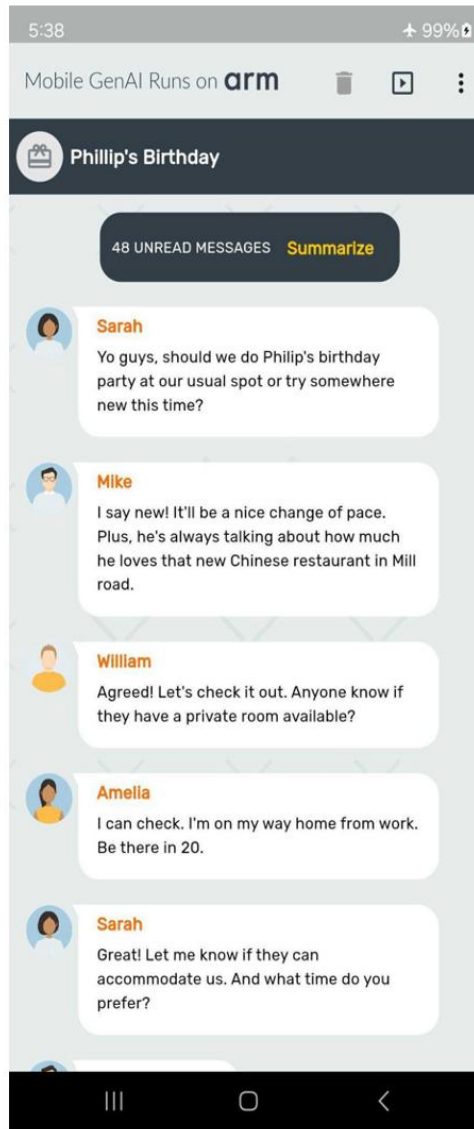
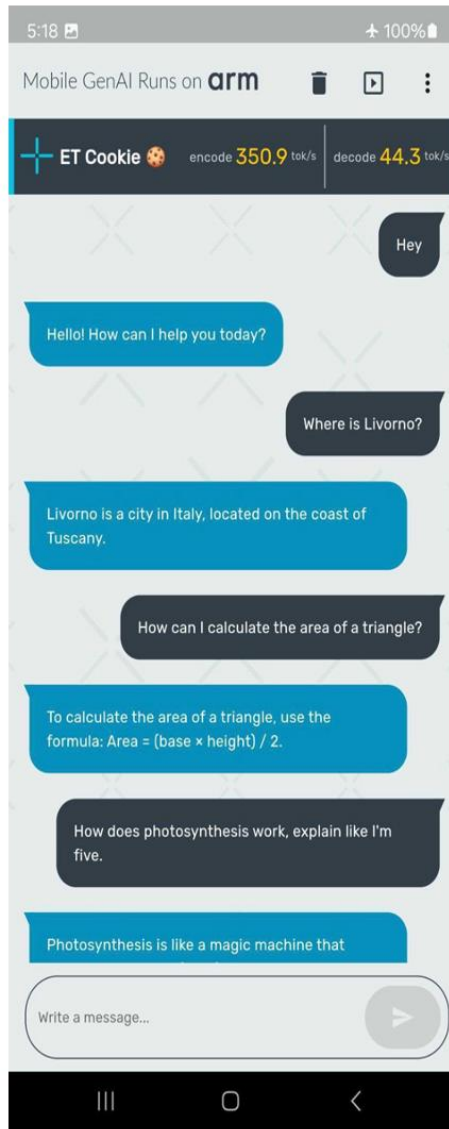
Prompt: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and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

[Sora: Creating video from text](#)

Inference Scenario 3: Surveillance System



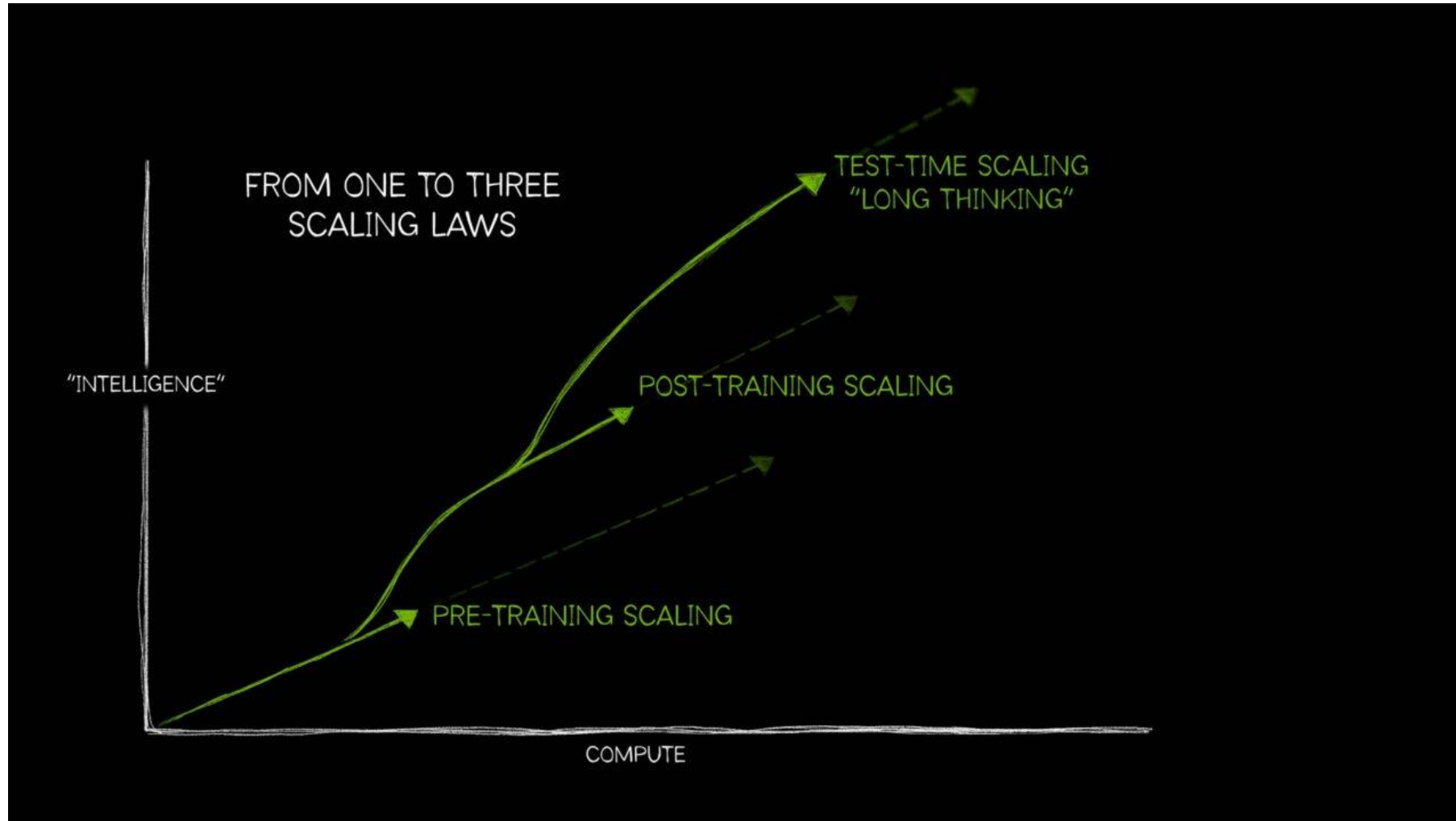
Inference Scenario 4: Mobile Apps

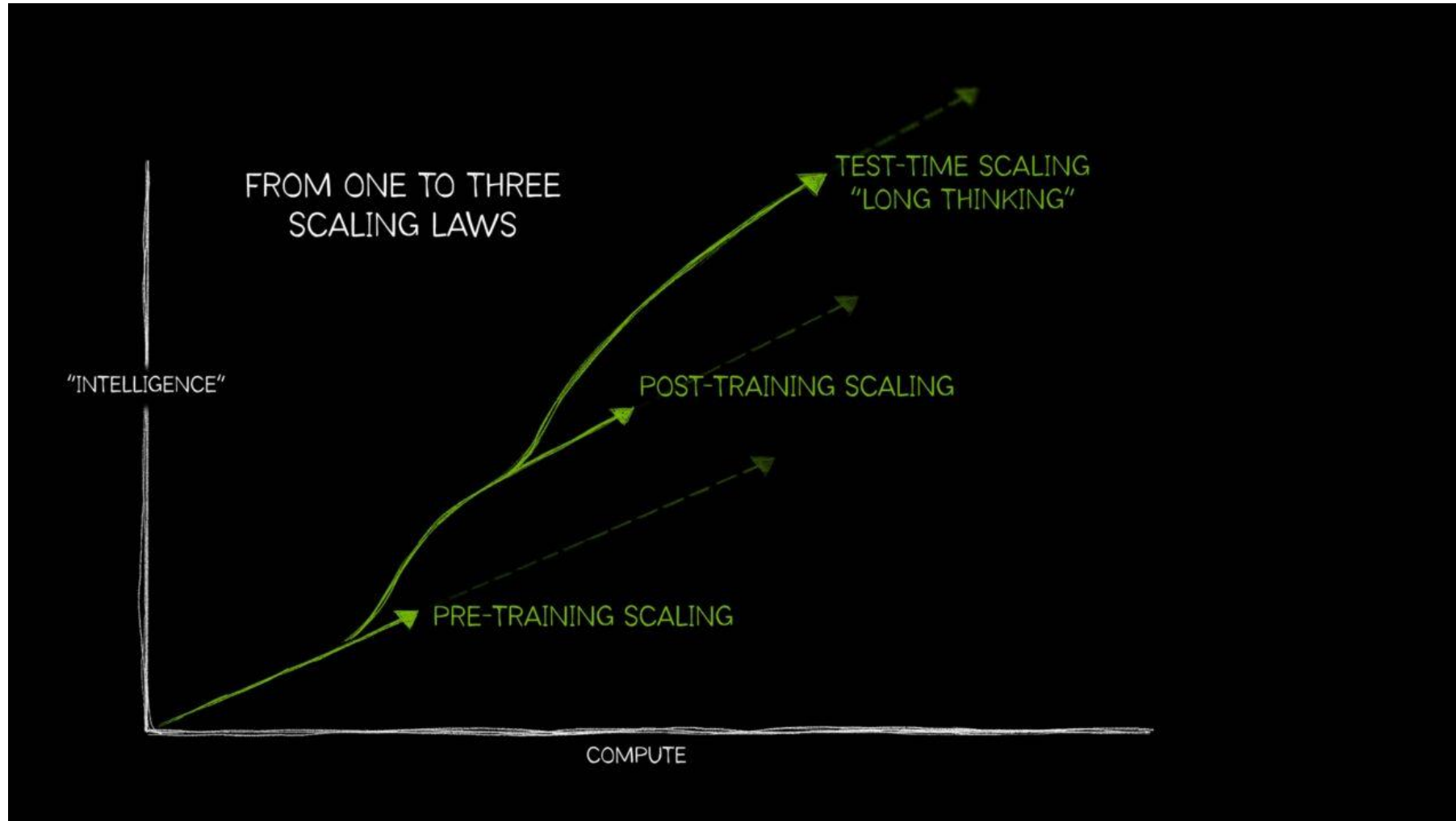


Inference Scenario 5: Augmented and Virtual Reality

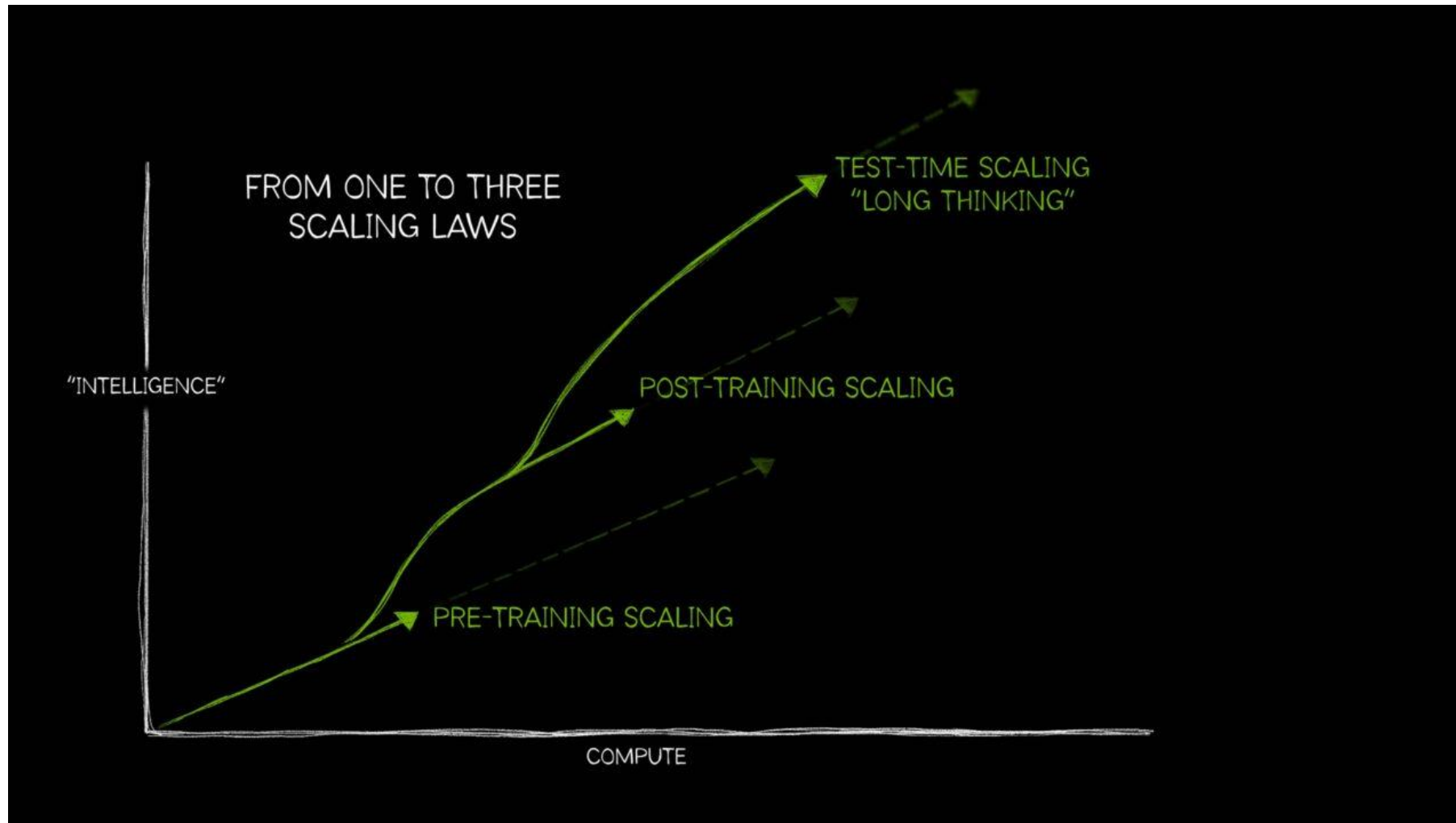


Inference Time Scaling





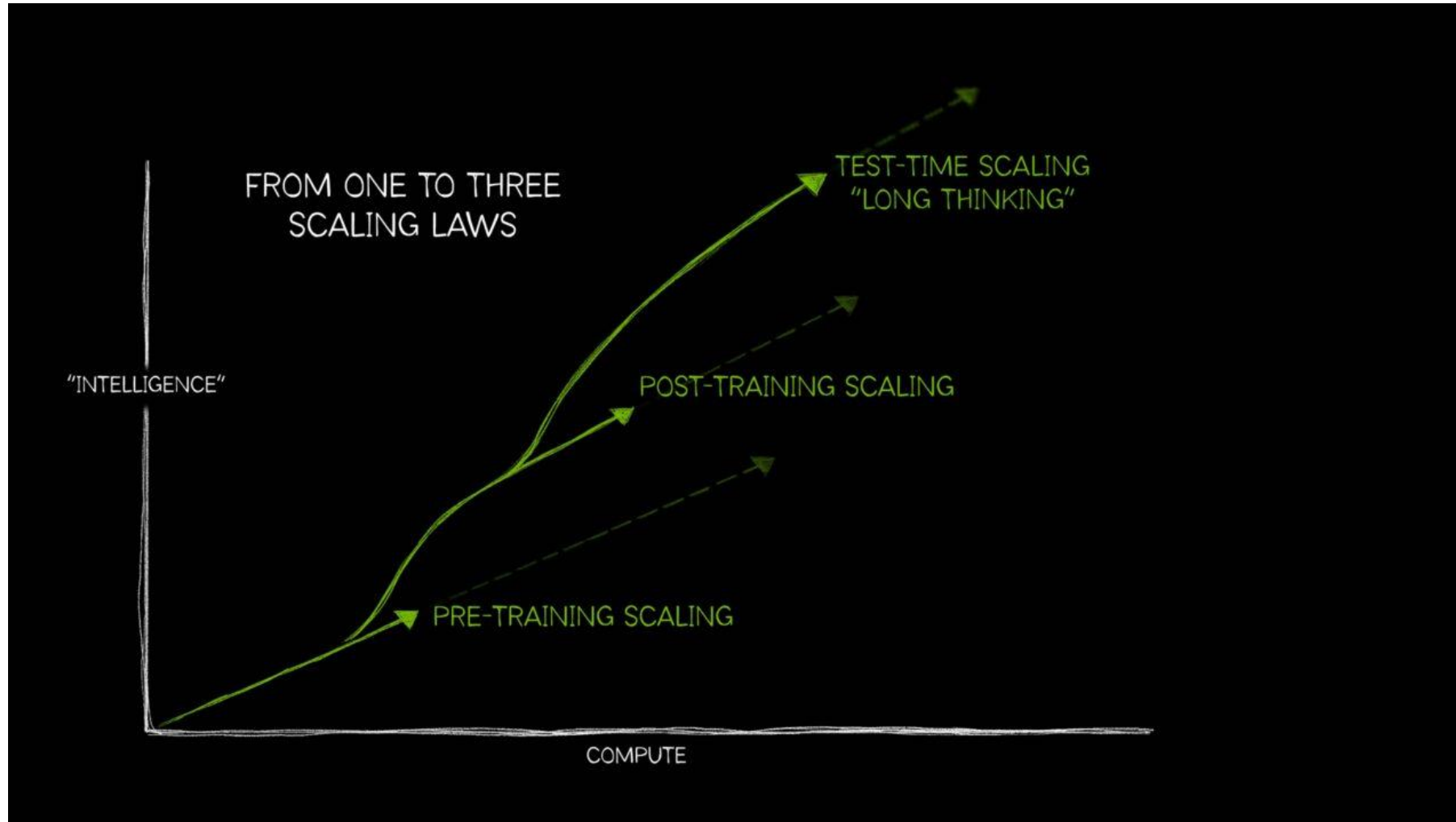
1 Pretraining scaling = bigger model and data, better performance



1 Pretraining scaling = bigger model and data, better performance

2 Post-training scaling = fine-tuning for precision

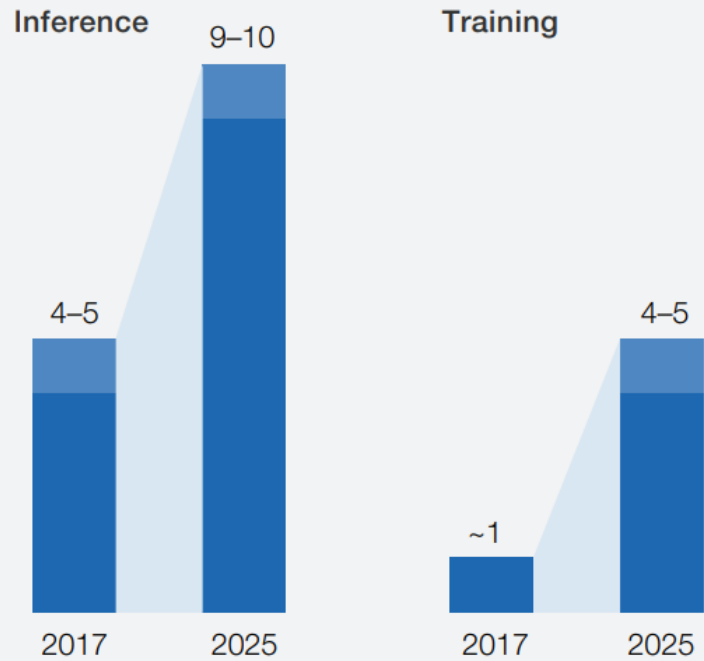
Inference Time Scaling Requires More Compute



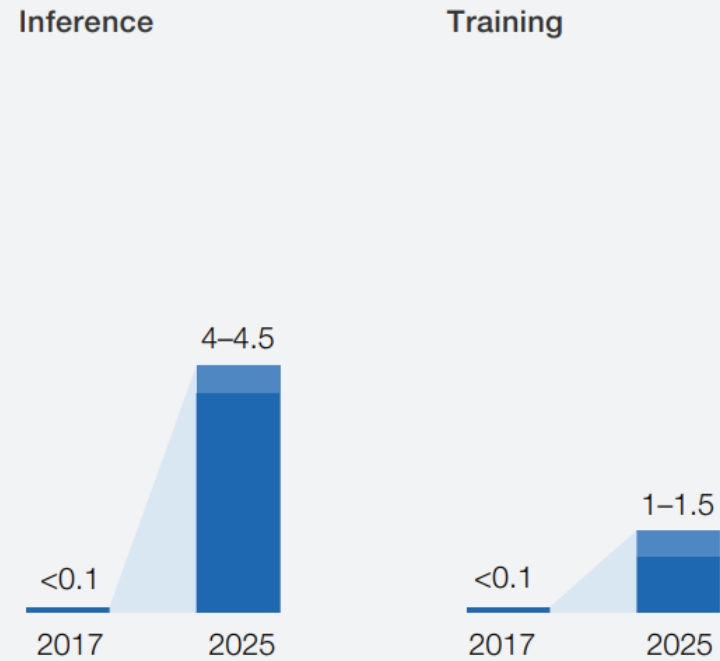
- 1 Pretraining scaling = bigger model and data, better performance
- 2 Post-training scaling = fine-tuning for precision
- 3 Test-time scaling (long thinking) = multi-pass reasoning for complex problems

Exhibit 5 At both data centers and the edge, demand for training and inference hardware is growing.

Data center, total market, \$ billion



Edge, total market, \$ billion



Source: Expert interviews; McKinsey analysis

Training

vs

Inference

Runtime

Weeks or months

Milliseconds or seconds

Training

vs

Inference

Runtime

Weeks or months

Milliseconds or seconds

Challenges

Speed, Scale, Data

SLA, TCO (total cost of ownership)

Training

vs

Inference

Runtime

Weeks or months

Milliseconds or seconds

Challenges

Speed, Scale, Data

SLA, TCO (total cost of ownership)

SLA (LLM: token rates)

TCO

- Parameter volume
- LLM: Context length

Inference Challenge 1: Long Latency Violates SLA



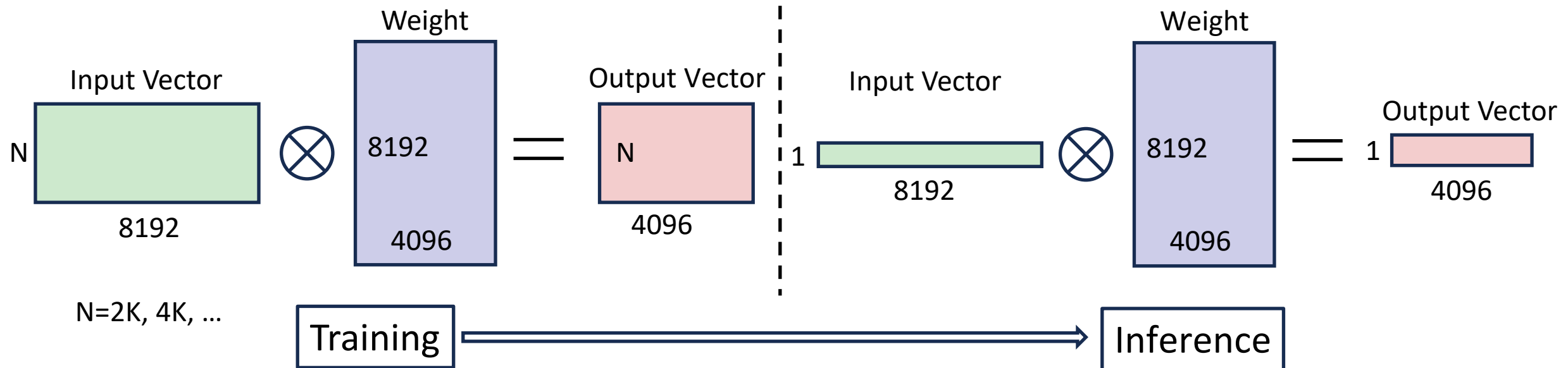
- Long serving latency blocks deployment
- Support advance models while meeting latency SLA and saving cost

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



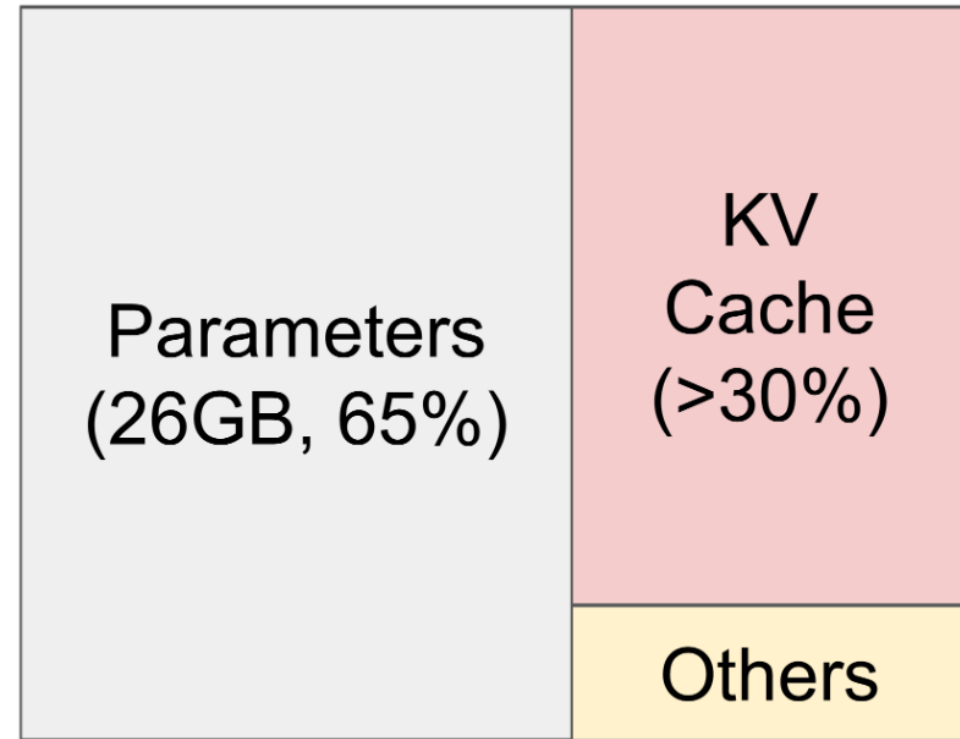
- Small batch size \implies Low data reuse (arithmetic intensity)
- Autoregressive generation \implies Generate tokens one-by-one



Inference Challenge 3: Large Memory Increases Cost



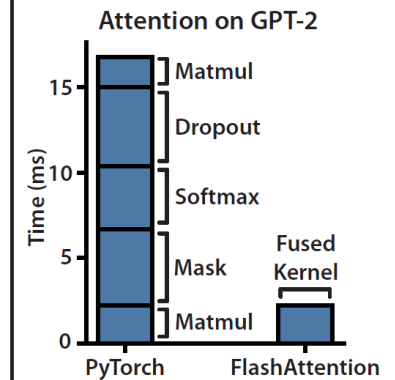
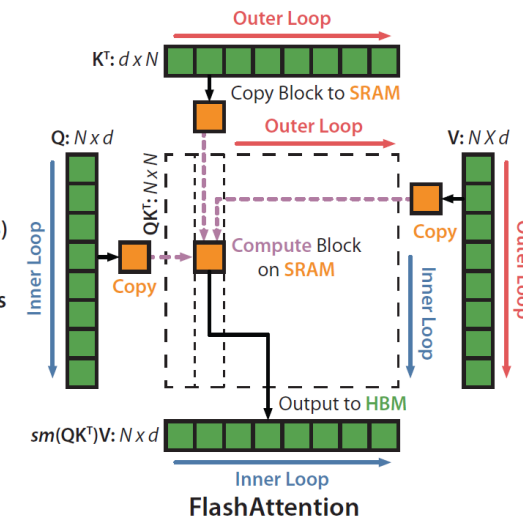
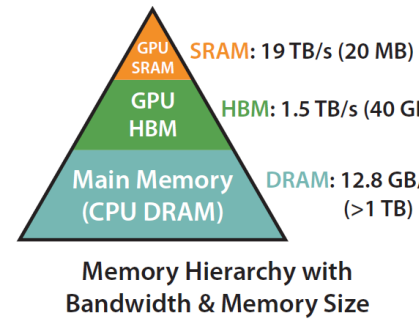
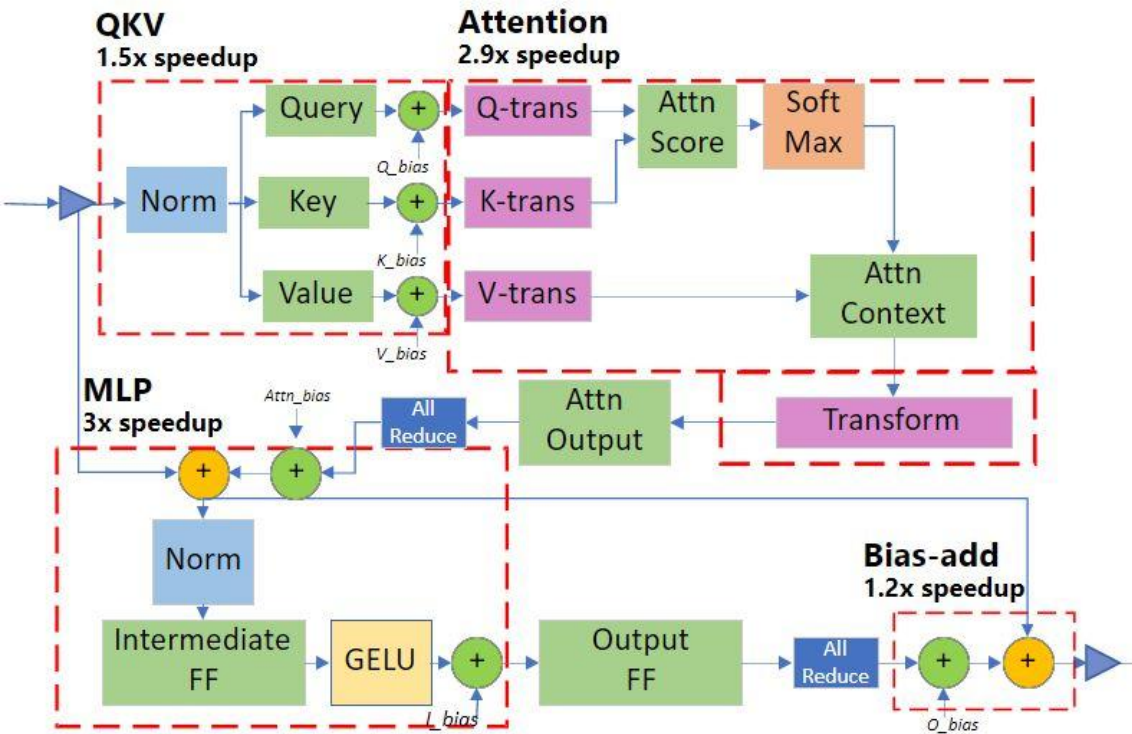
- Model parameters
 - # Layers
 - # Hidden dim
- KV cache
 - Batch size
 - Sequence length
 - # Layers
 - # Hidden
- Activation and others



LLaMA-13B on A100 40 GB

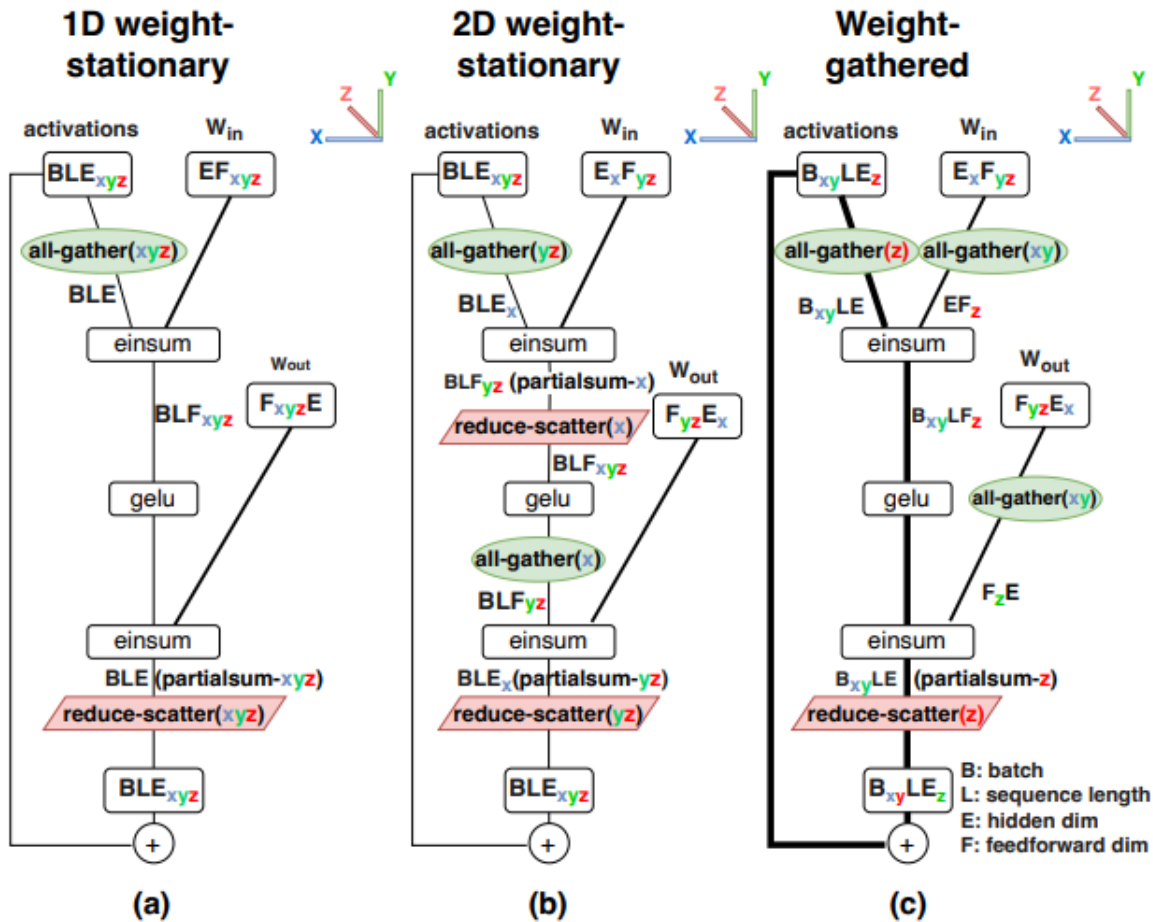
[Efficient Memory Management for Large Language Model Serving with PagedAttention](#), by Kwon et al., 2023

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

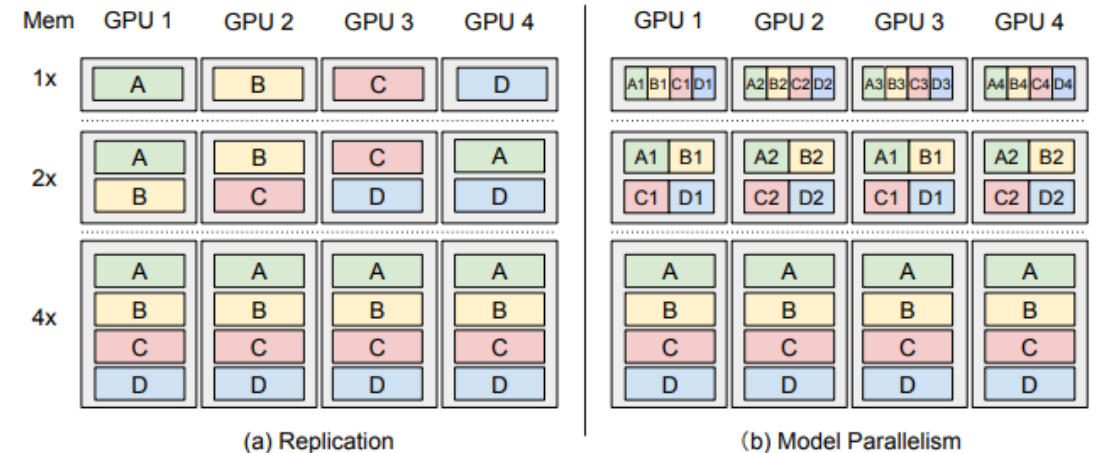


DeepSpeed-Inference: enabling efficient inference of transformer models at unprecedented scale, SC 2022

Multi-GPU Inference via Partitioned Layouts

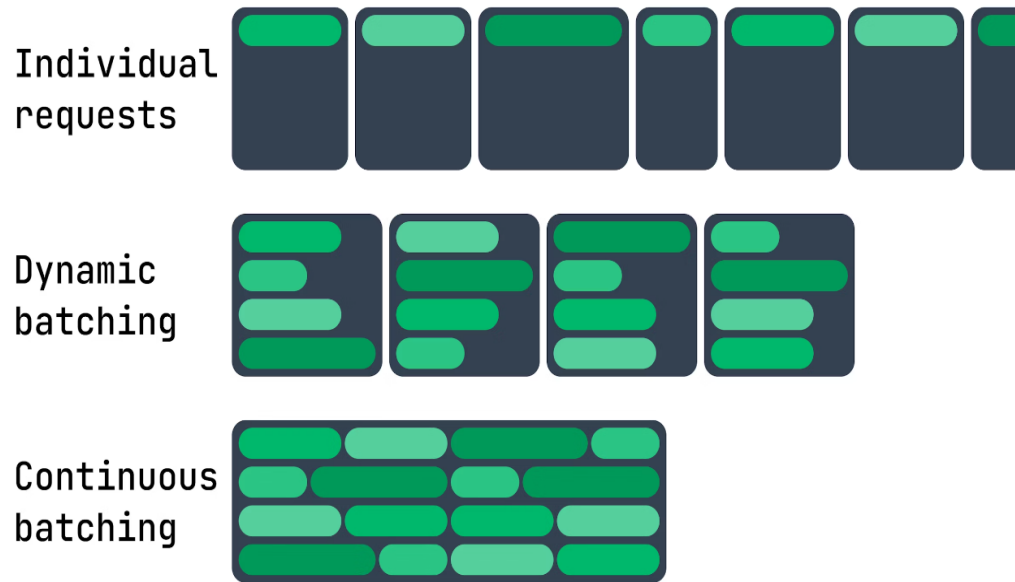


AlpaServe: Statistical Multiplexing with Model Parallelism for Deep Learning Serving, OSDI 2023



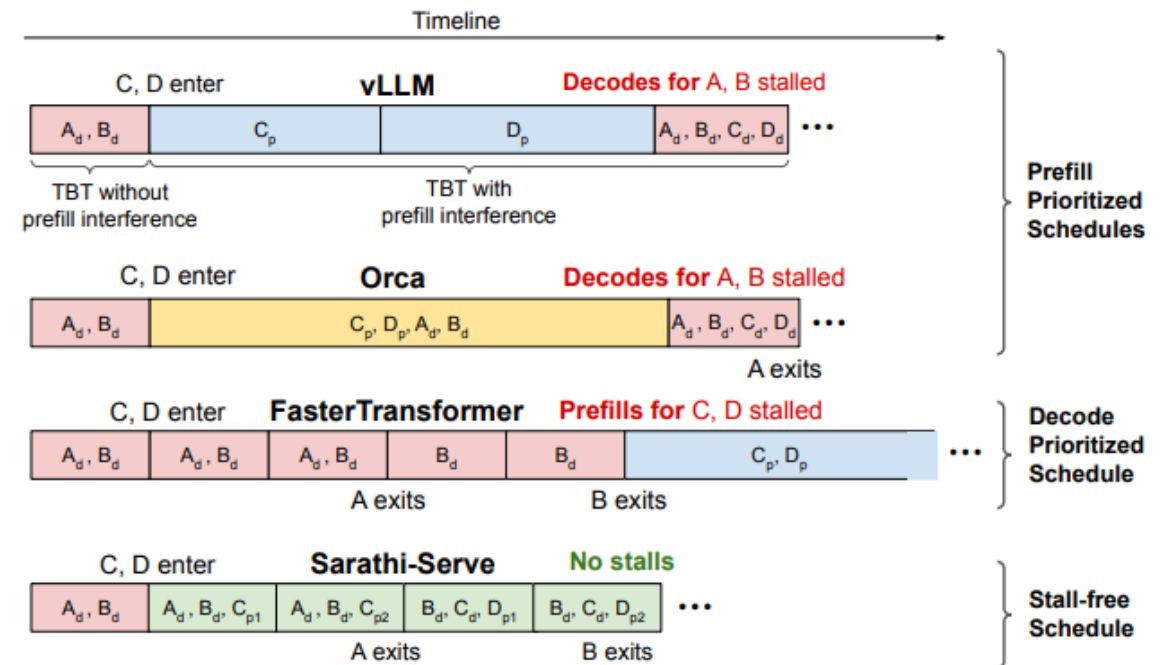
Efficiently Scaling Transformer Inference, MLSys 2023

Scheduling Strategies for LLM Inference

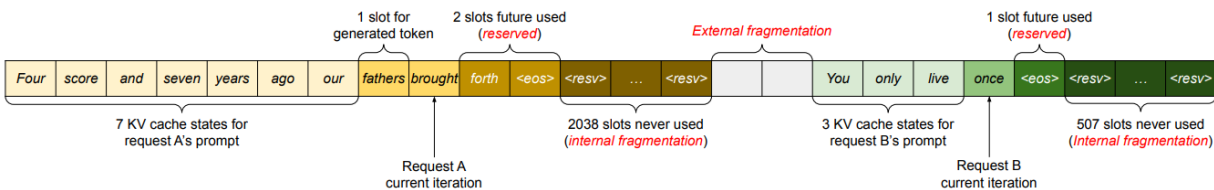


Orca: A Distributed Serving System for Transformer-Based Generative Models, OSDI 2022

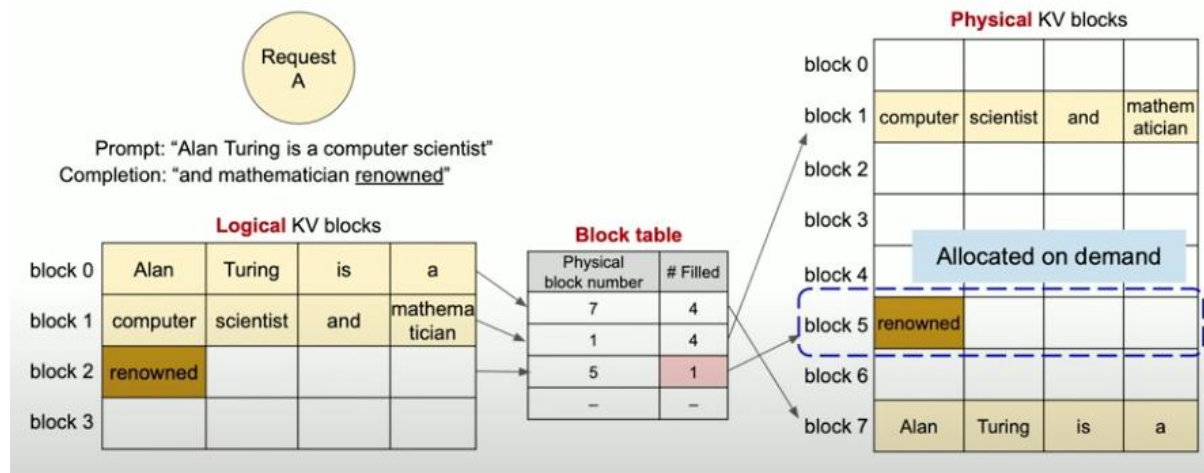
Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve, OSDI 2024



KV Cache Management

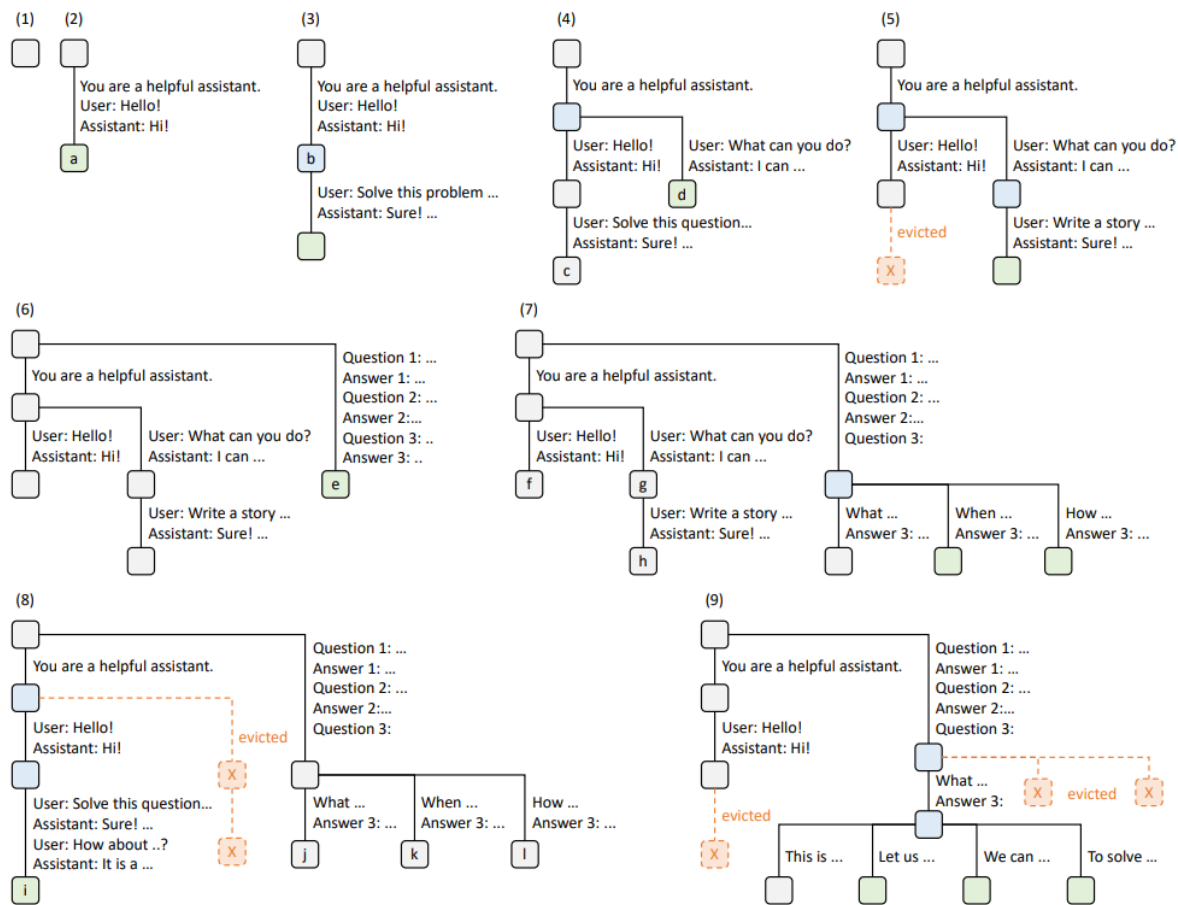


Logical & physical KV blocks



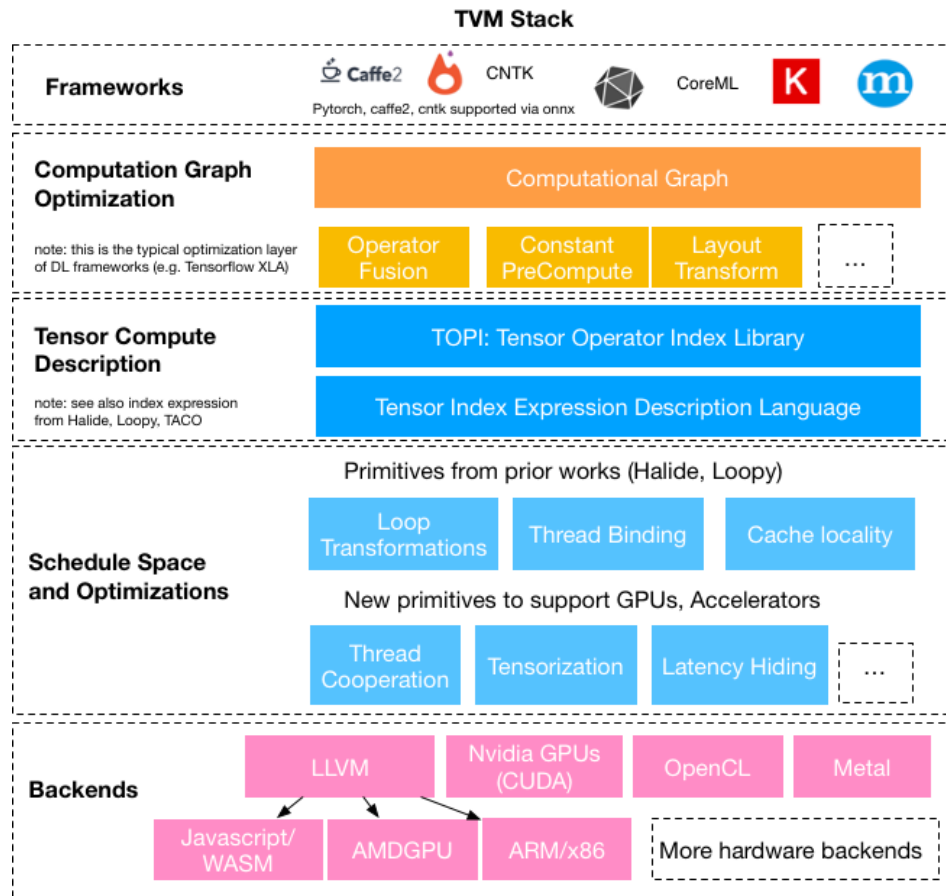
Efficient Memory Management for Large Language Model Serving with PagedAttention, 2023

SGLang: Efficient Execution of Structured Language Model Programs, 2024



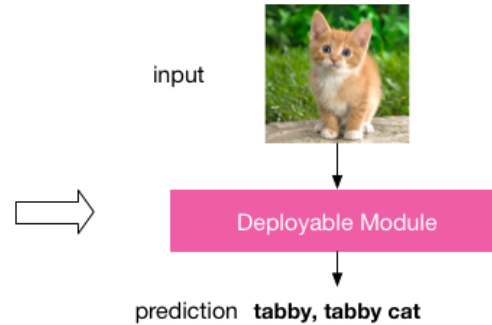
DL Compilation

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

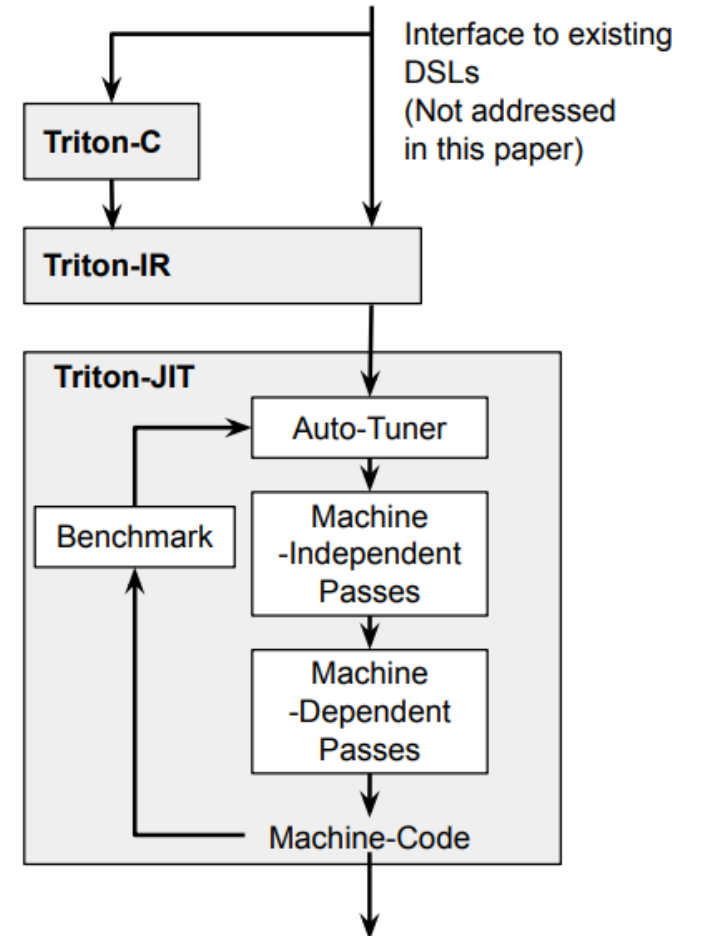


Runtime: Lightweight and Cross Platform

```
module = runtime.create(graph, lib, tvm.gpu(0))
module.set_input(**params)
module.run(data=data_array)
output = tvm.nd.empty(out_shape, ctx=tvm.gpu(0))
module.get_output(0, output)
```

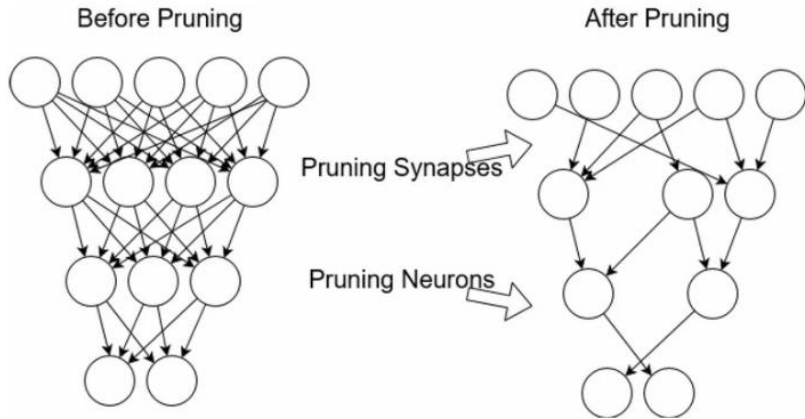


Deploy Languages and Platforms

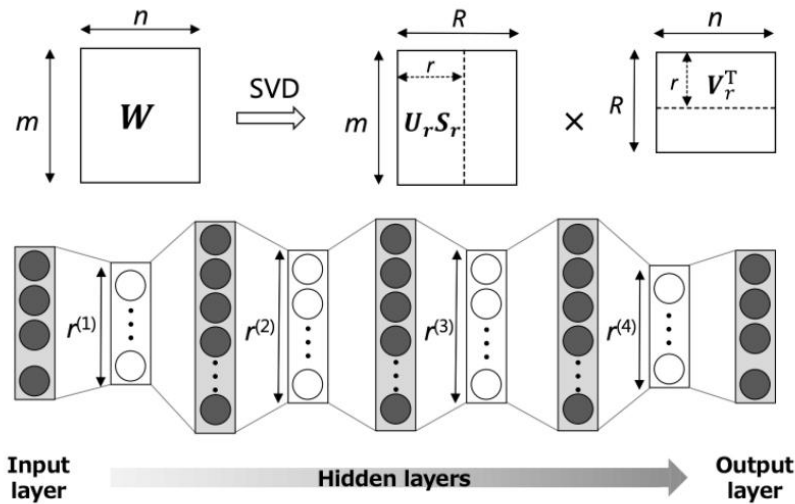


TVM: An Automated End-to-End Optimizing Compiler for Deep Learning, 2018

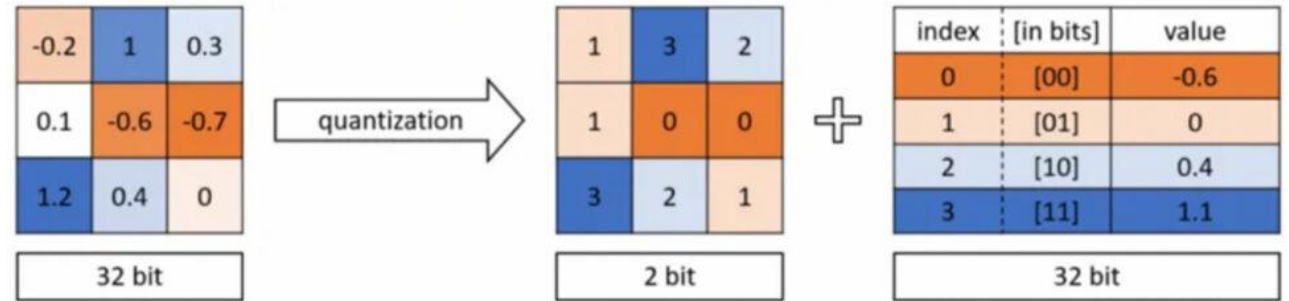
Compression Strategies



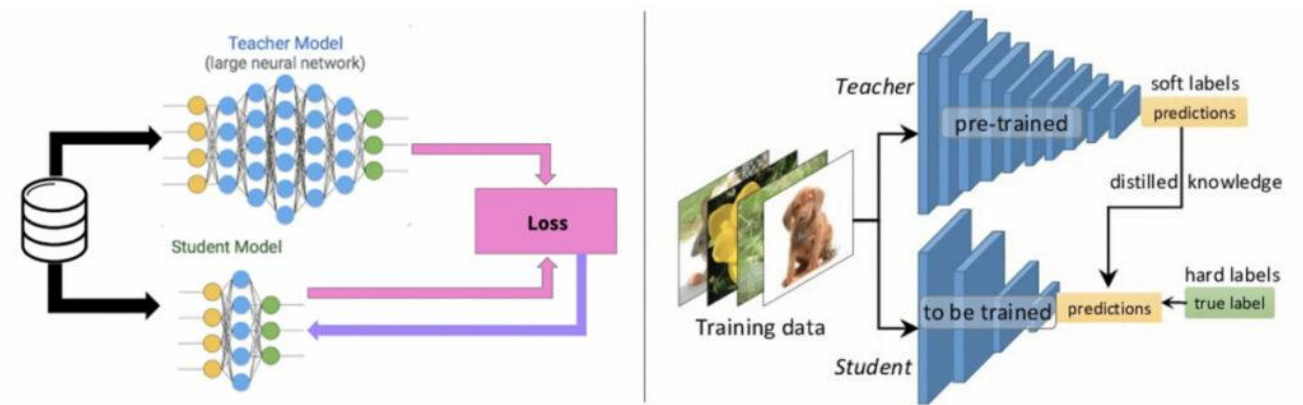
Pruning/Sparsification



Low-rank decomposition

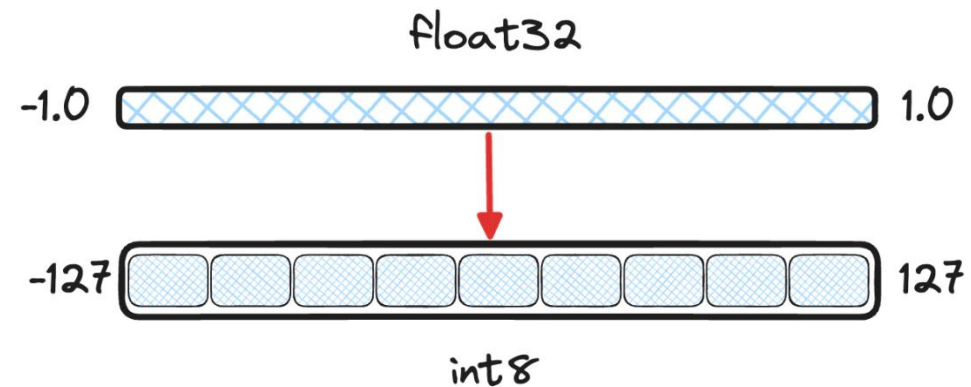


Quantization

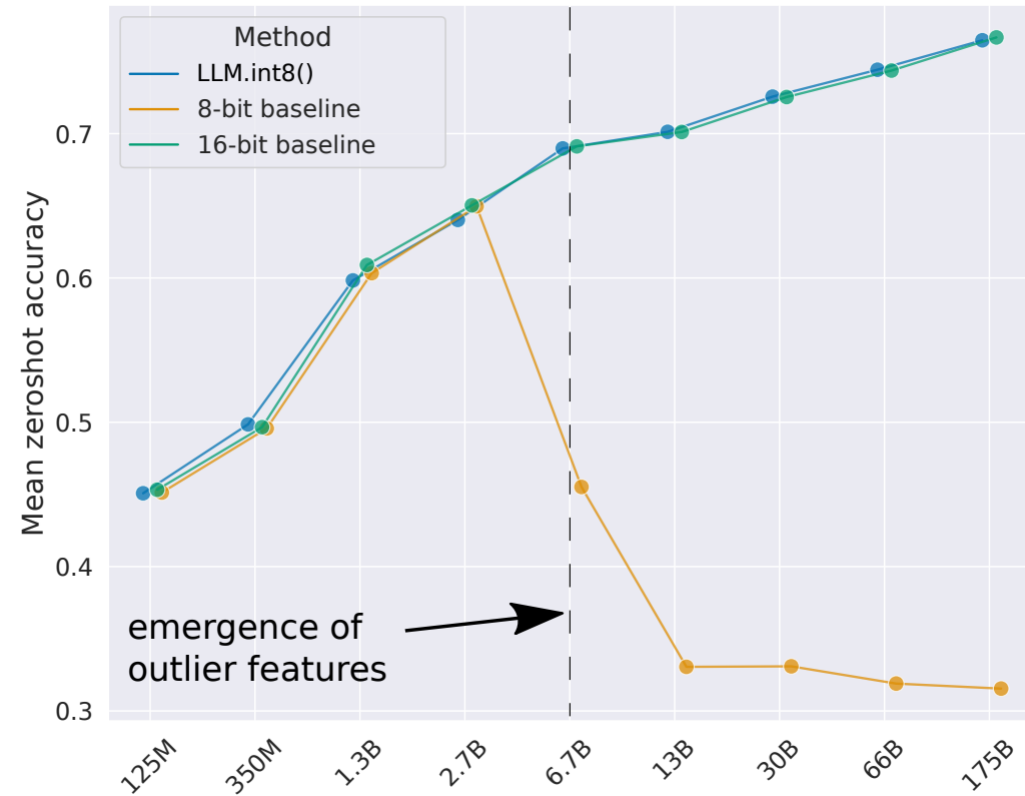


Distillation

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

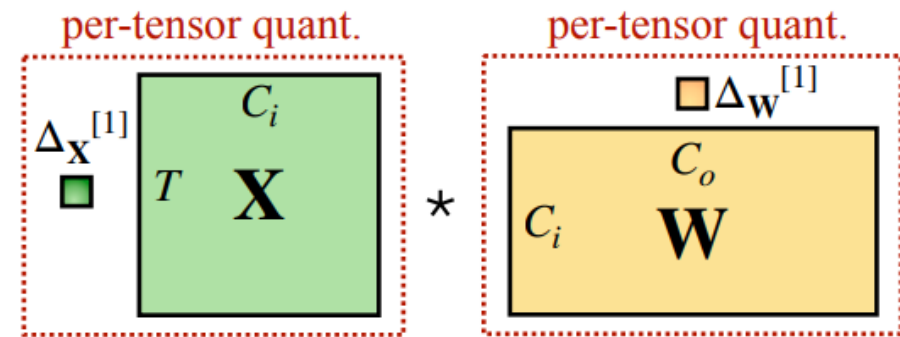


- Standard quantization strategy leads to catastrophic accuracy drop

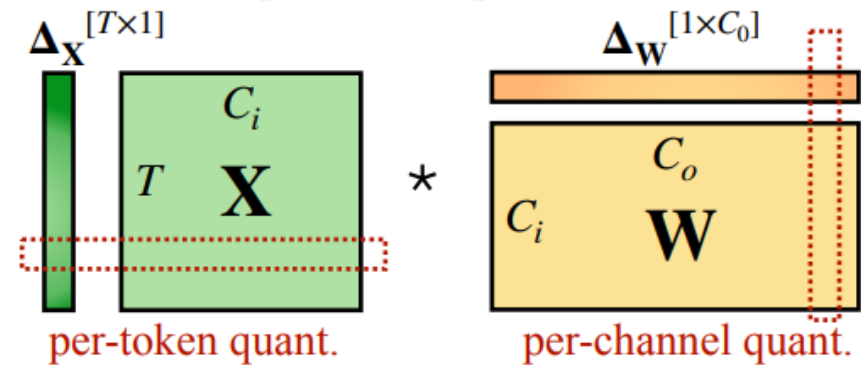


LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, 2023

- 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



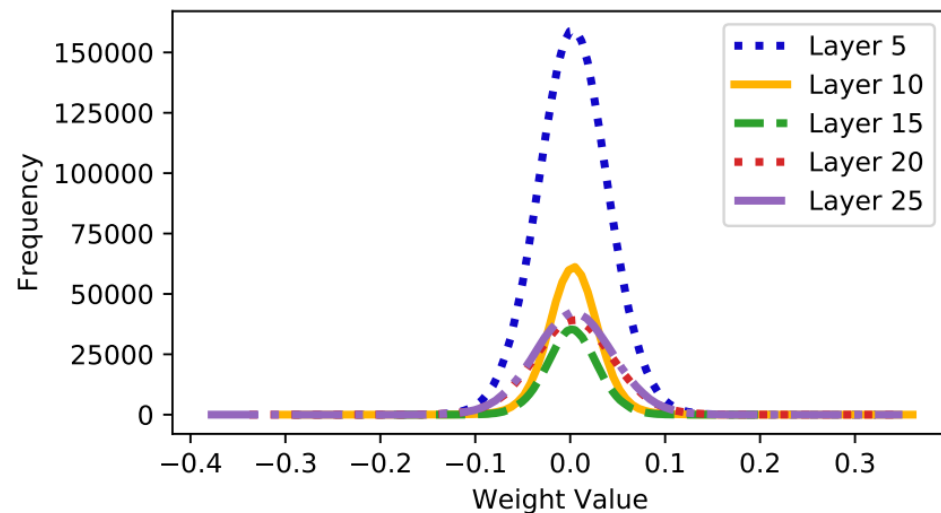
(b) per-token + per-channel quantization

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

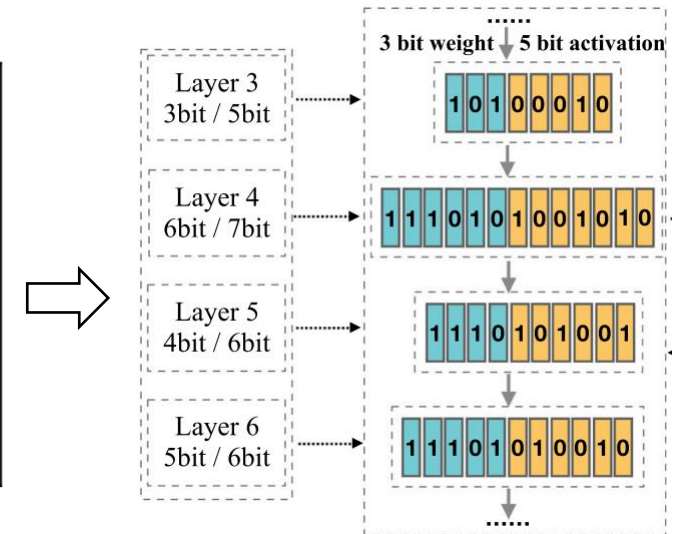
Mixed Precision Quantization



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

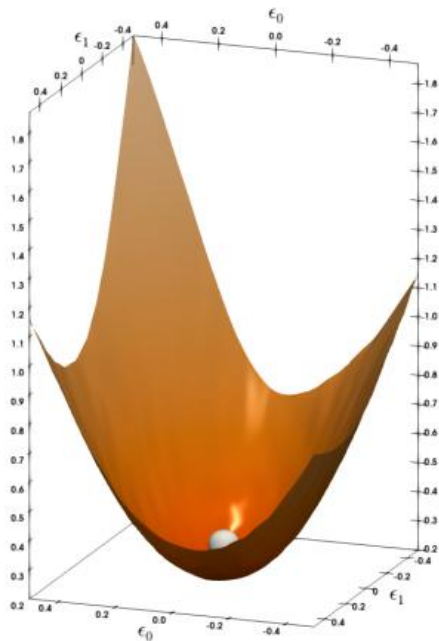


Per-layer weight distribution of BERT model

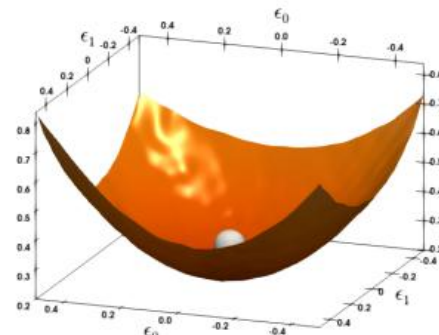


GOBO: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference, MICRO 2020

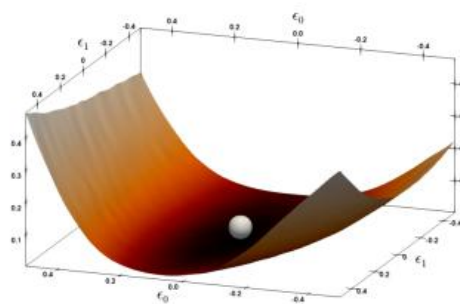
- Analyze the loss curvature (Hessian matrices) to help identify layer sensitivity



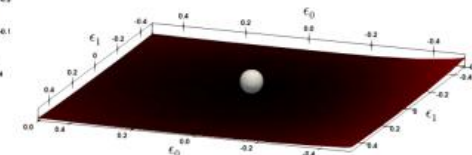
(a) MNLI 4th layer



(b) MNLI 10th layer

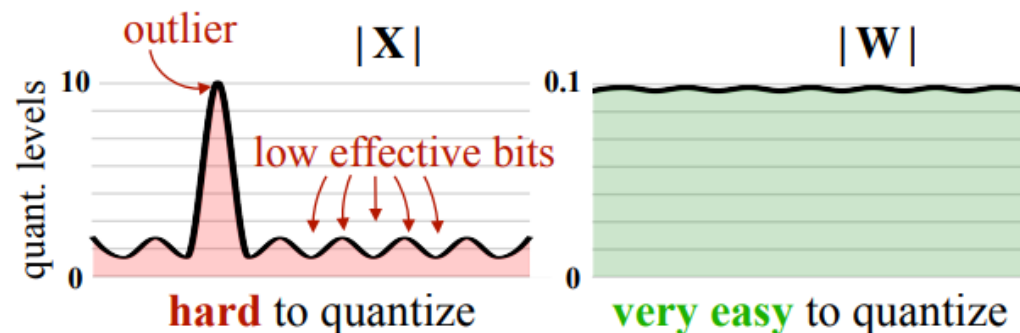


(c) CoNLL-03 4th layer

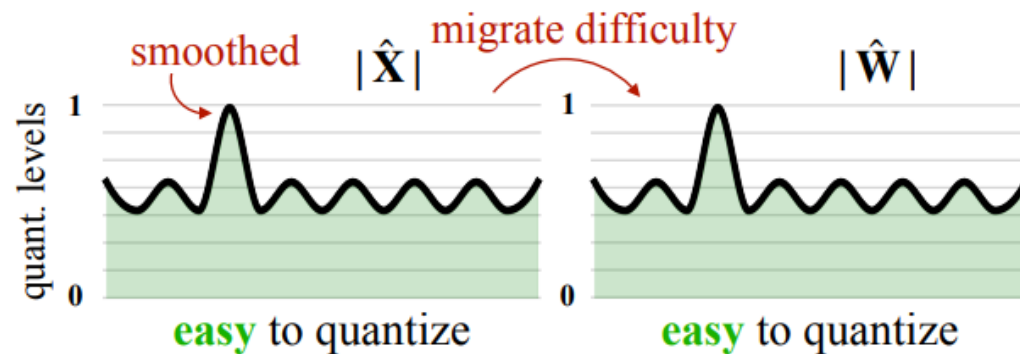


(d) CoNLL-03 11th layer

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



(a) Original



(b) SmoothQuant

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

Sparsification



Unstructured (connection) Sparsity:

- High accuracy
- No performance improvement or performance regression

1	7	-3	4	2	-1	-3	3
5	3	6	2	0	2	1	7
-8	-2	1	3	5	-6	2	0
3	9	1	4	5	-3	0	1
9	0	-1	3	6	2	-1	3
4	-5	2	8	7	6	-3	0
-1	-1	4	7	0	7	6	1
9	1	2	4	6	8	9	0

Unstructured Sparsity

Structured Sparsity:

- Large accuracy degradation
- High performance scalability

N:M Semi-Structured Sparsity:

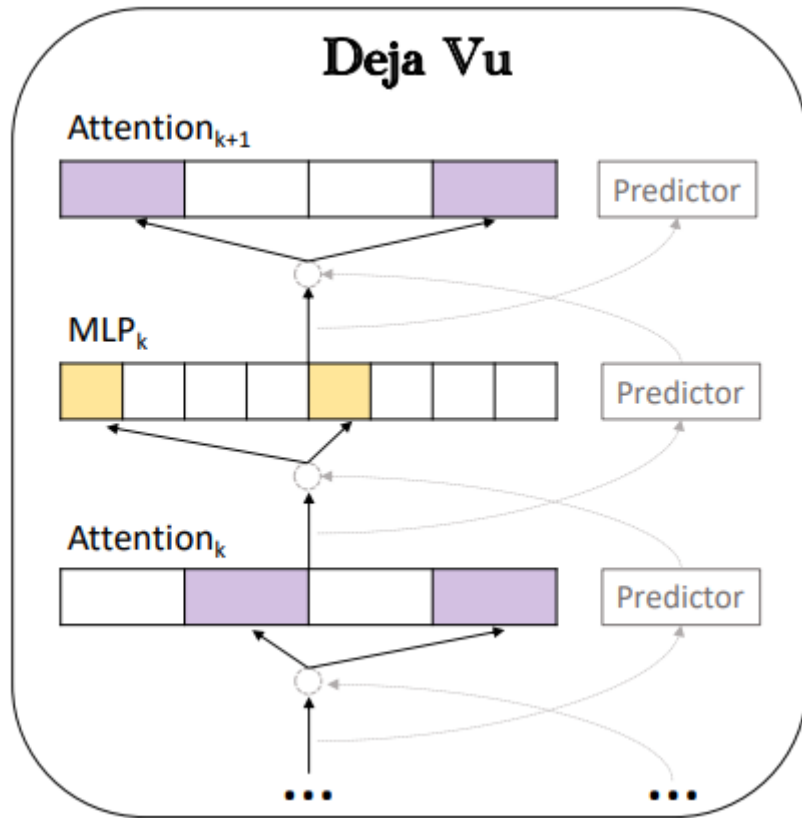
- High accuracy
- High performance improvement

1	7	-3	4	2	-1	-3	3
5	3	6	2	0	2	1	7
-8	-2	1	3	5	-6	2	0
3	9	1	4	5	-3	0	1
9	0	-1	3	6	2	-1	3
4	-5	2	8	7	6	-3	0
-1	-1	4	7	0	7	6	1
9	1	2	4	6	8	9	0

Structured Sparsity (Column-wise Sparsity)

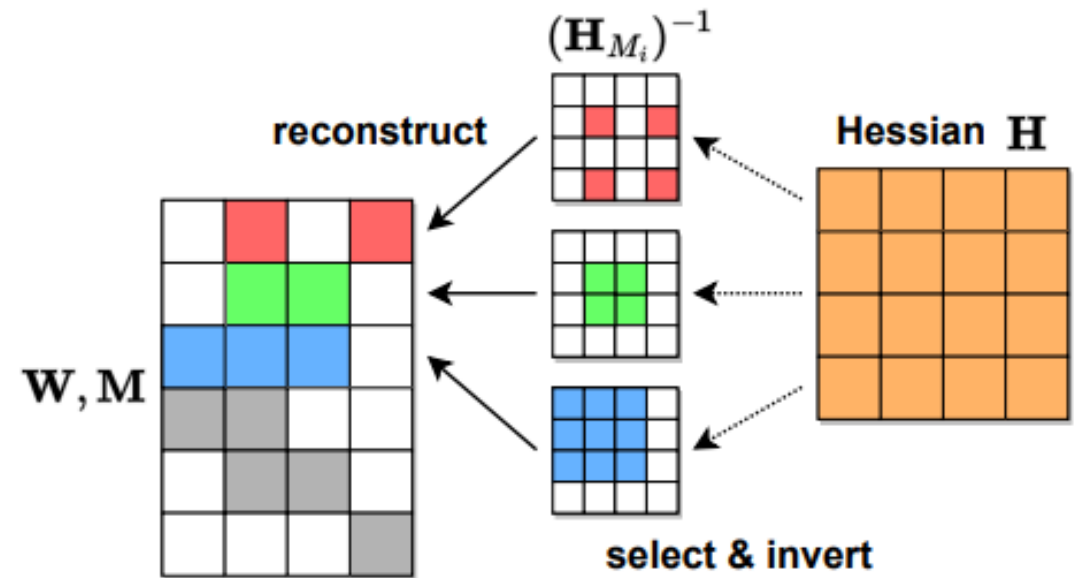
1	7	-3	4	2	-1	-3	3
5	3	6	2	0	2	1	7
-8	-2	1	3	5	-6	2	0
3	9	1	4	5	-3	0	1
9	0	-1	3	6	2	-1	3
4	-5	2	8	7	6	-3	0
-1	-1	4	7	0	7	6	1
9	1	2	4	6	8	9	0

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

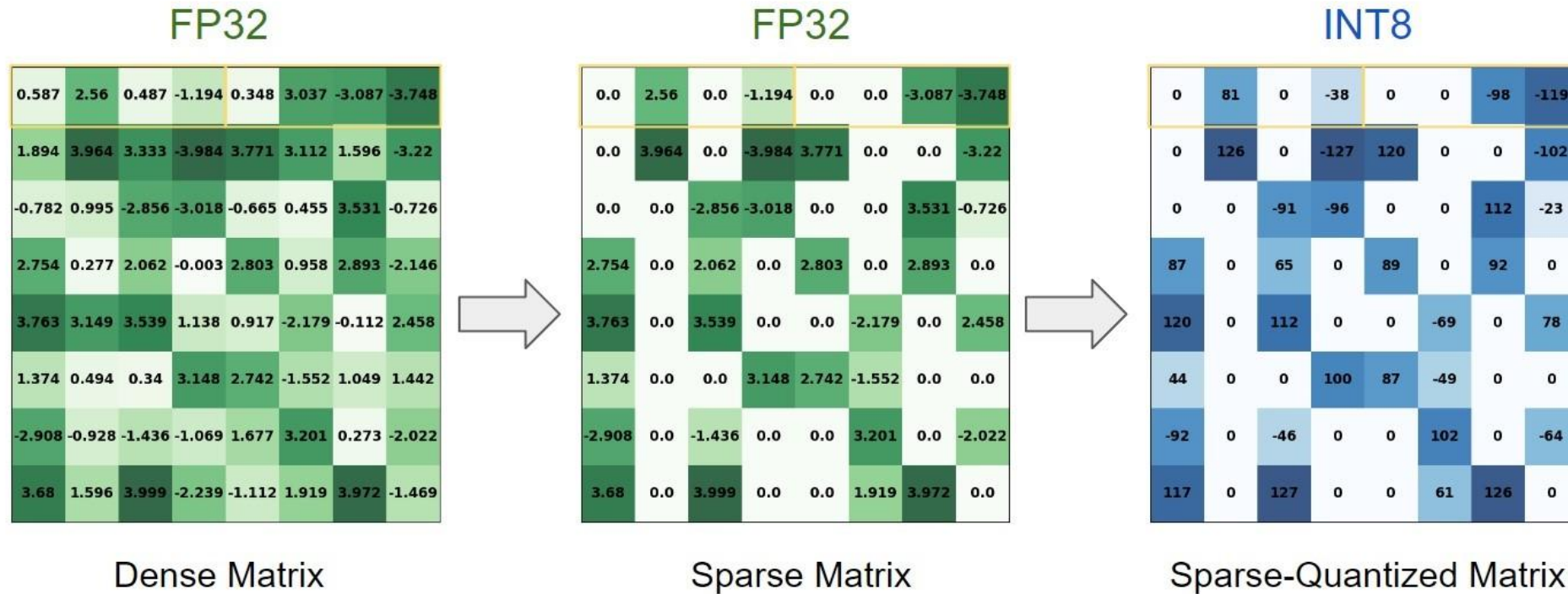


DeJa Vu: Contextual Sparsity for Efficient LLMs at Inference Time, 2023

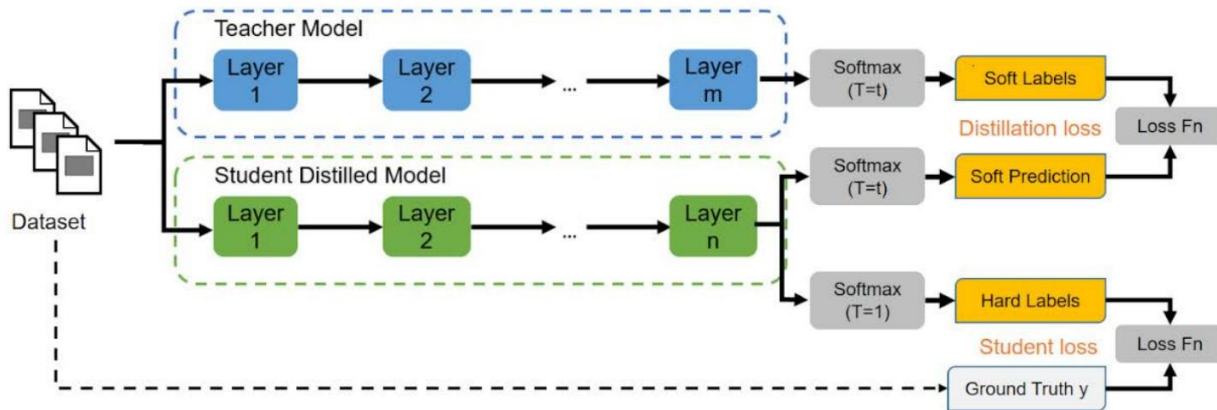
SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot, 2023



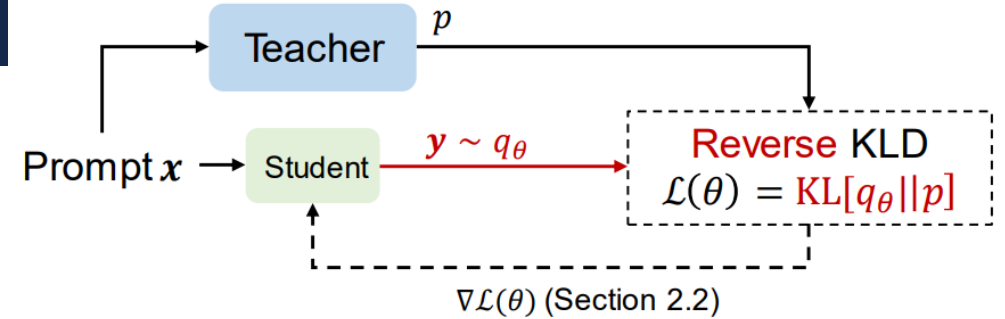
Sparsification + Quantization



Knowledge Distillation

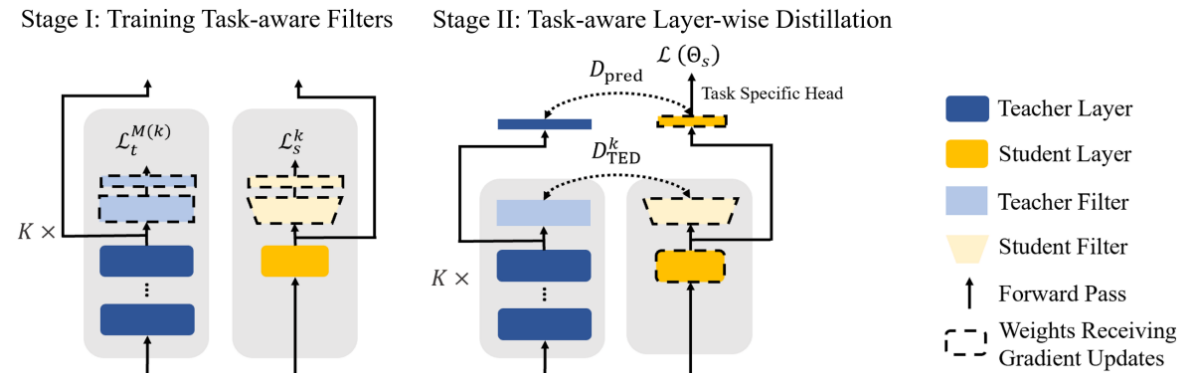


Distilling the Knowledge in a Neural Network, 2015



MiniLLM (Ours)

MiniLLM: Knowledge distillation of large language models, 2024

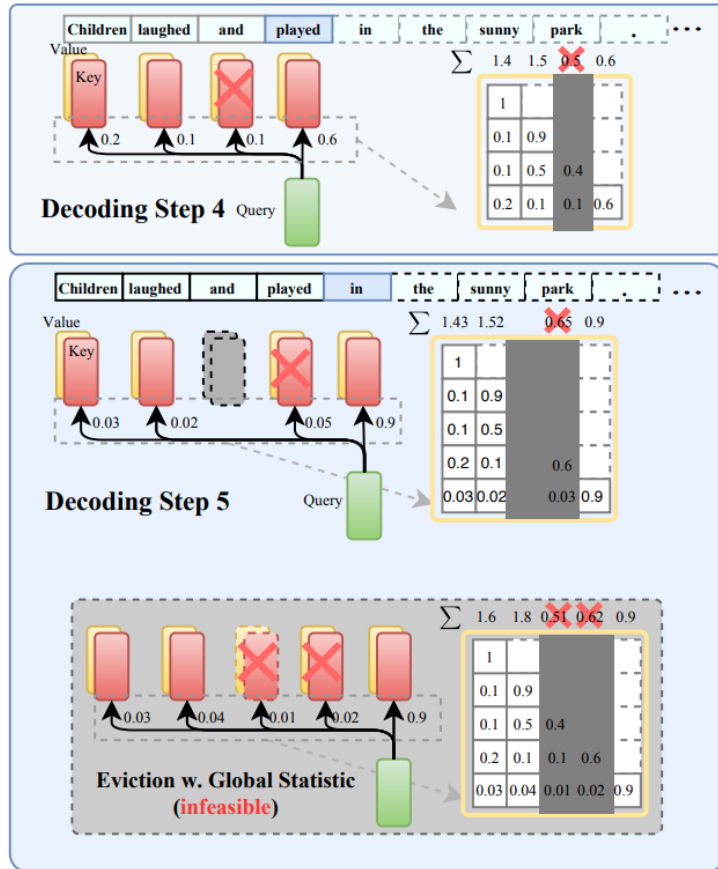


Less is More: Task-aware Layer-wise Distillation for Language Model Compression, 2024

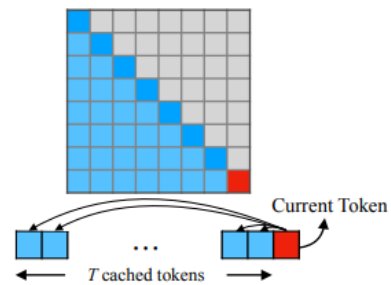
KV Cache Compression



Efficient Streaming Language Models with Attention Sinks, ICL 2024



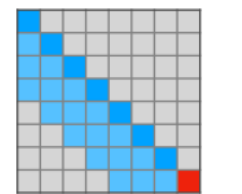
(a) Dense Attention



$O(T^2)$ \times PPL: 5641 \times

Has poor efficiency and performance on long text.

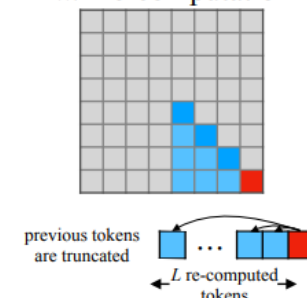
(b) Window Attention



$O(TL)$ \checkmark PPL: 5158 \times

Breaks when initial tokens are evicted.

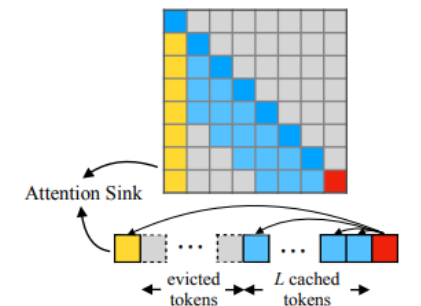
(c) Sliding Window w/ Re-computation



$O(TL^2)$ \times PPL: 5.43 \checkmark

Has to re-compute cache for each incoming token.

(d) StreamingLLM (ours)

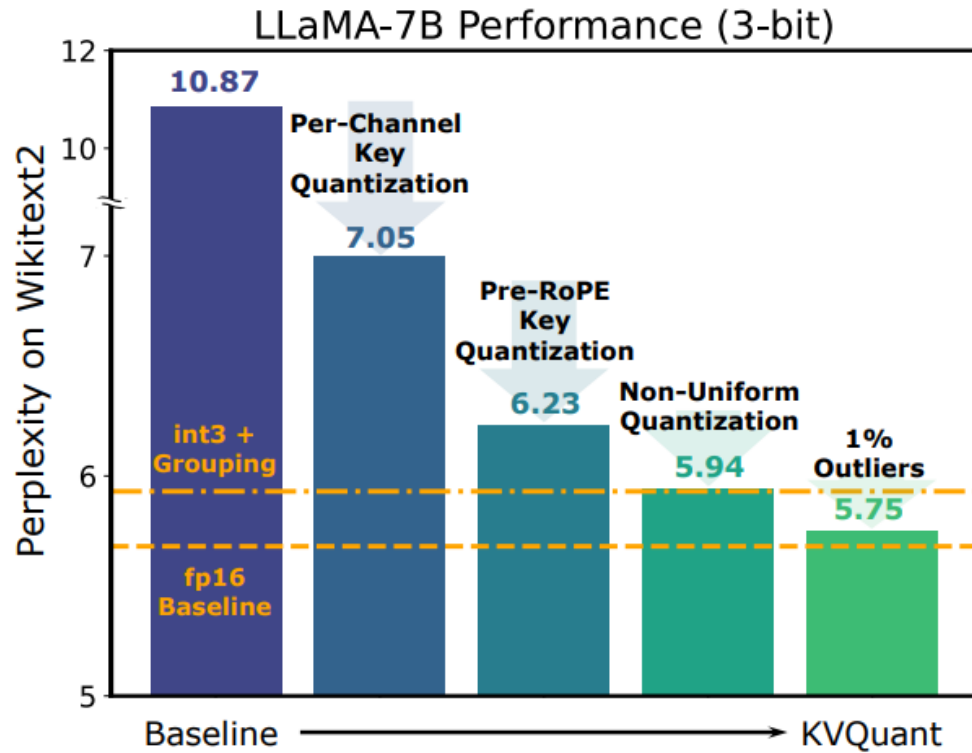


$O(TL)$ \checkmark PPL: 5.40 \checkmark

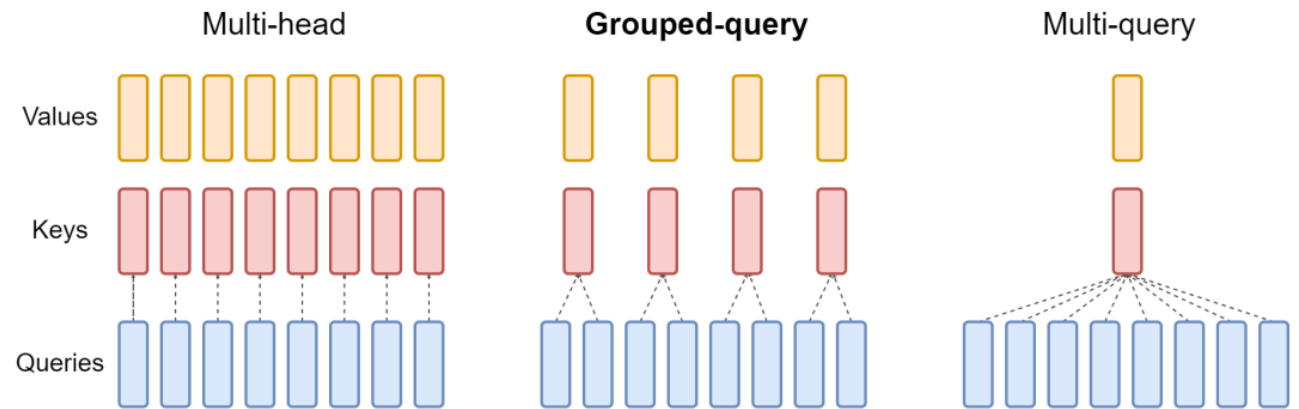
Can perform efficient and stable language modeling on long texts.

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

KV Cache Compression

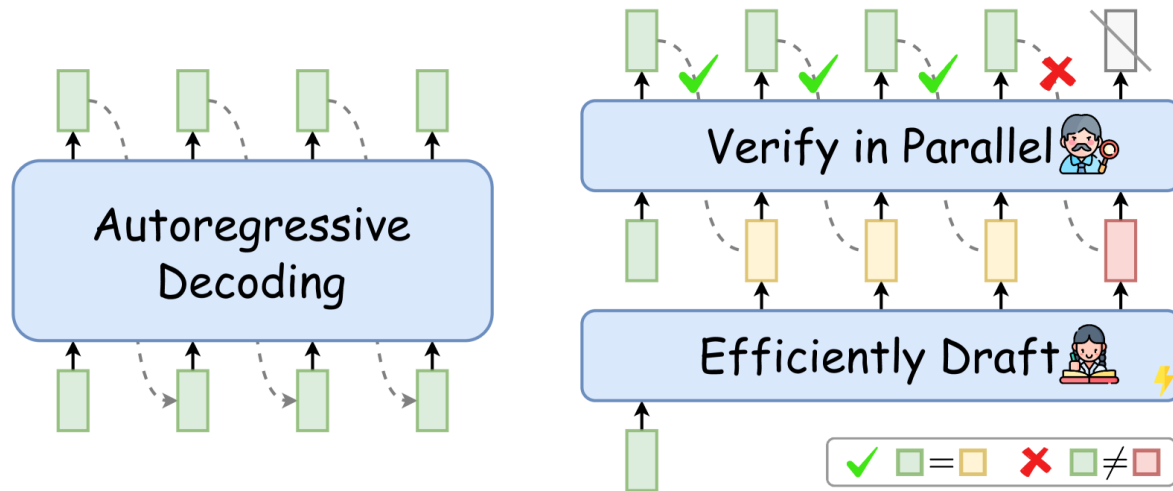


GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, 2023



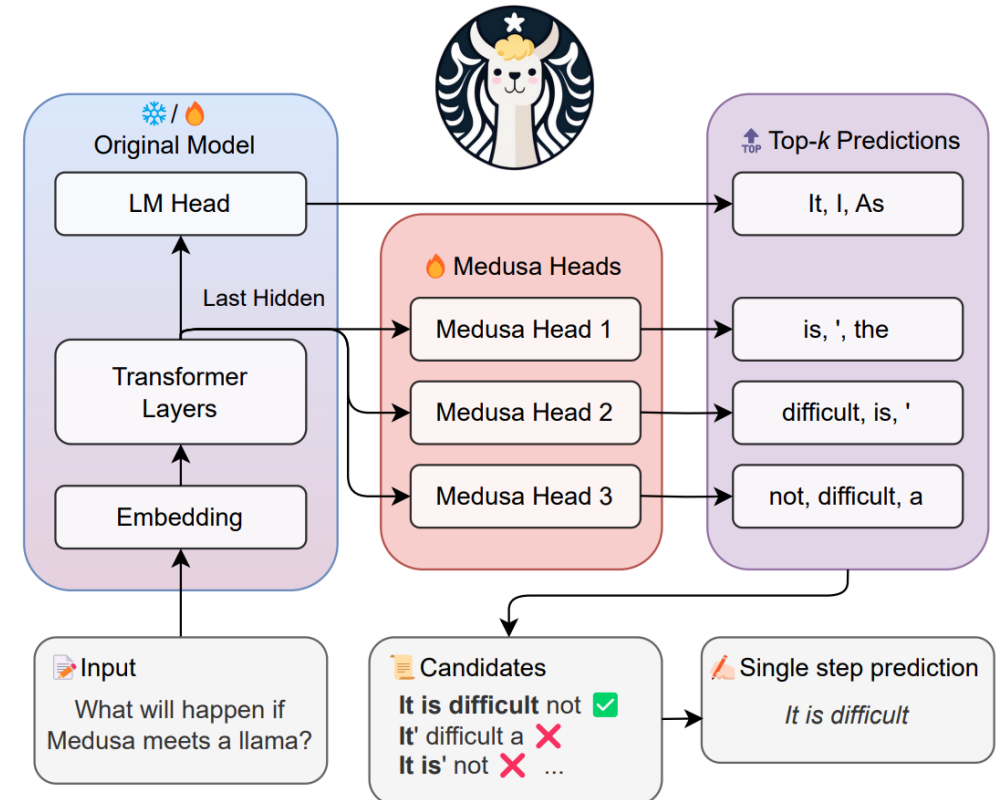
KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization, 2024

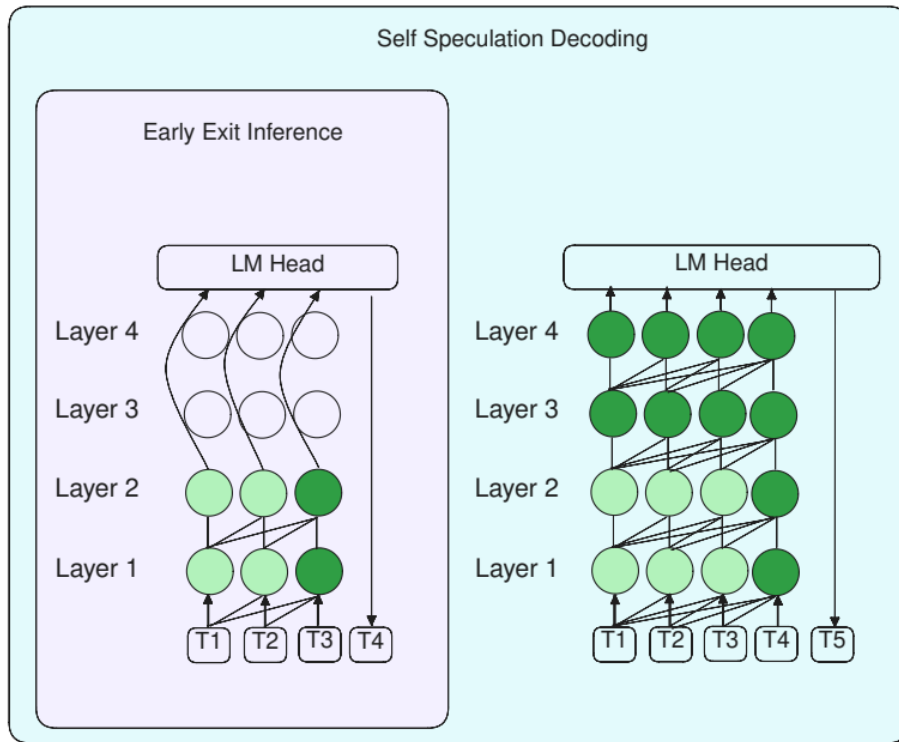
Speculative/Parallel Decoding



Fast Inference from Transformers via Speculative Decoding, 2023

MEDUSA: Simple LLM Inference Acceleration Framework with Multiple, 2024

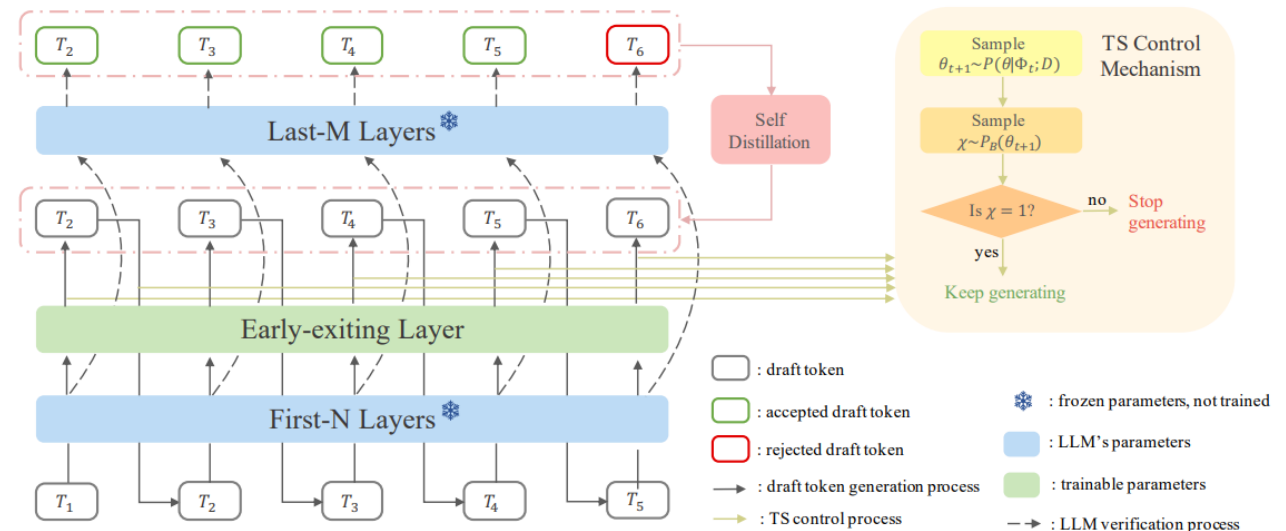




... enables inference with subset of layers with higher accuracy...

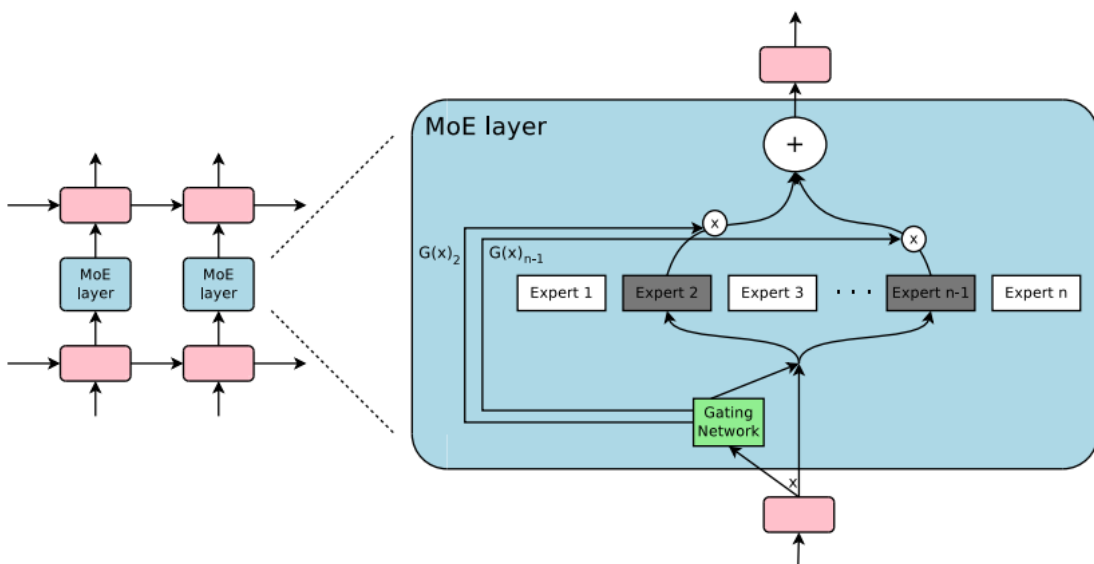
... and we can improve accuracy by verifying and correcting with remaining layers

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



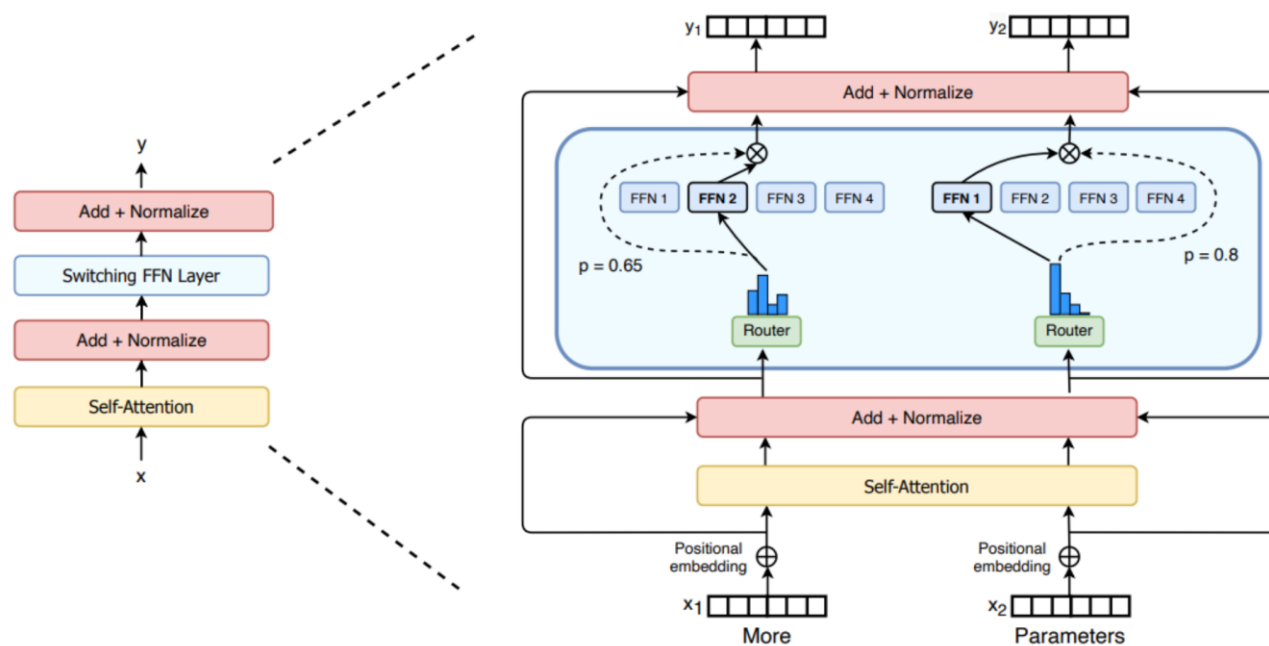
LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding, 2024

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

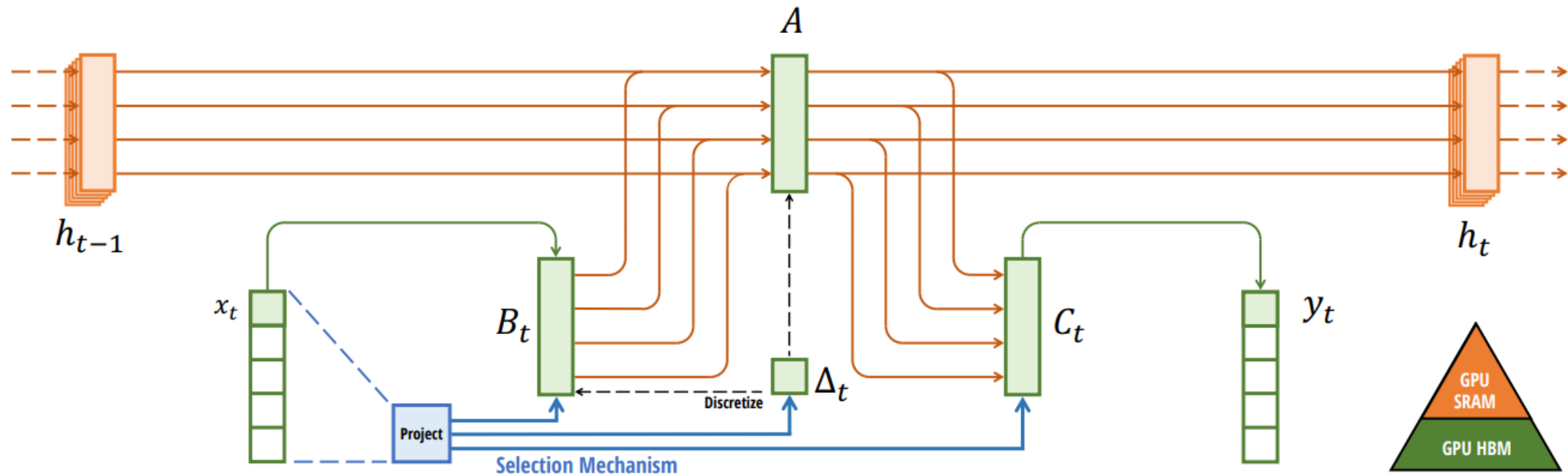


Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, 2017

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



Selective State Space Model *with Hardware-aware State Expansion*



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

Questions?