Lecture 11:
A Deeper Dive into GPUs and Compute

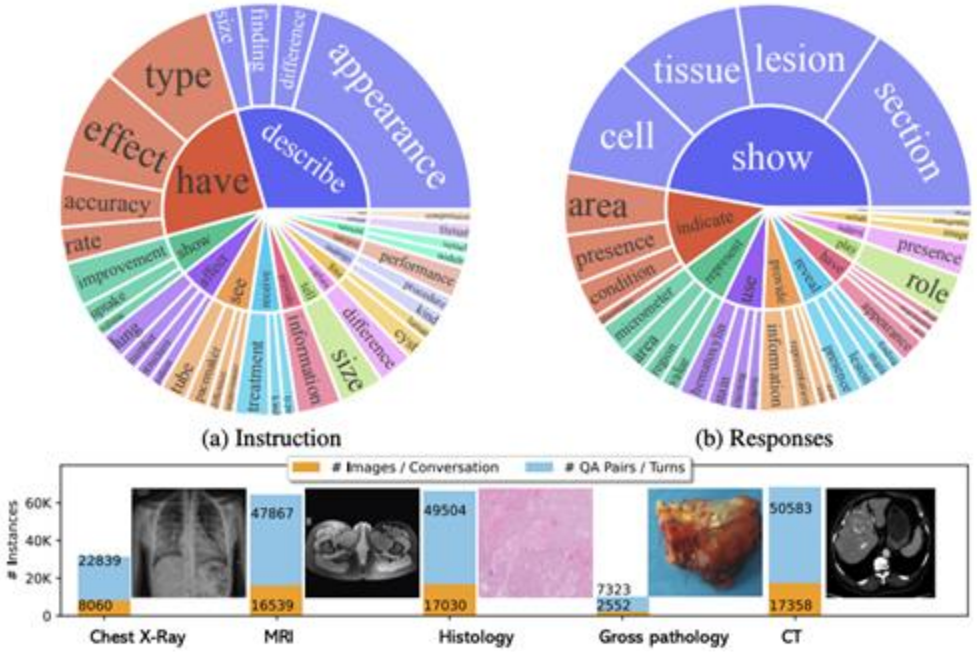
Announcements

- A2 will be released Oct 30, due Nov 13
- Discussion presentations start Mon Nov 4
- Before starting discussions, we will summarize and synthesize what we have seen in class so far at the start of the Wed Oct 30 lecture
- Wed Oct 30 from 11-11:50 will be a special guest lecture / Q&A from Troy Tazbaz from the FDA on regulation and responsible and ethical deployment of AI models. Everyone is highly encouraged to attend this class!

Finishing up from last lecture:
Vision Language Generative Models in Biomedicine

Last time we covered: LLaVA-Med, Training a Large Language-and-Vision Assistant for Biomedicine in One Day

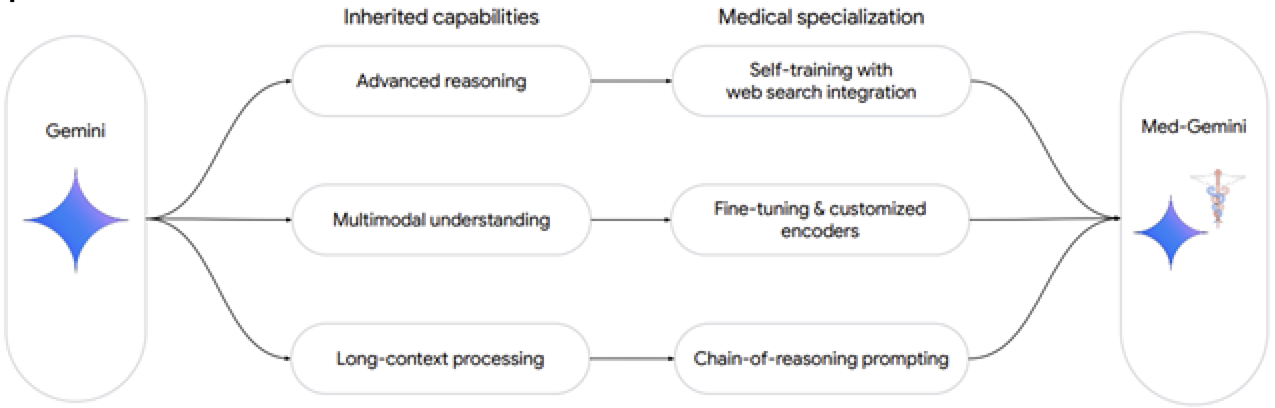
- Extends LLaVA to better answer biomedical questions, using a new biomedical instruction tuning dataset
- Instruction tuning dataset leverages PMC-15M (PubMed Central figure-caption dataset) and covers diverse domains
- Efficiently trained in < 15 hours using eight A100s



Li et al. LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day. NeurIPS Datasets and Benchmarks 2023.

Last time we covered: Med-Gemini, state-of-the-art generalist biomedical VLM

- Also a generalist (broad domains) biomedical VLM like LLaVA-Med, but extends from the much more powerful Gemini models (and built by internal Google DeepMind team)
- SoTA (state-of-the-art) due to Gemini foundation and additional techniques for medical specialization

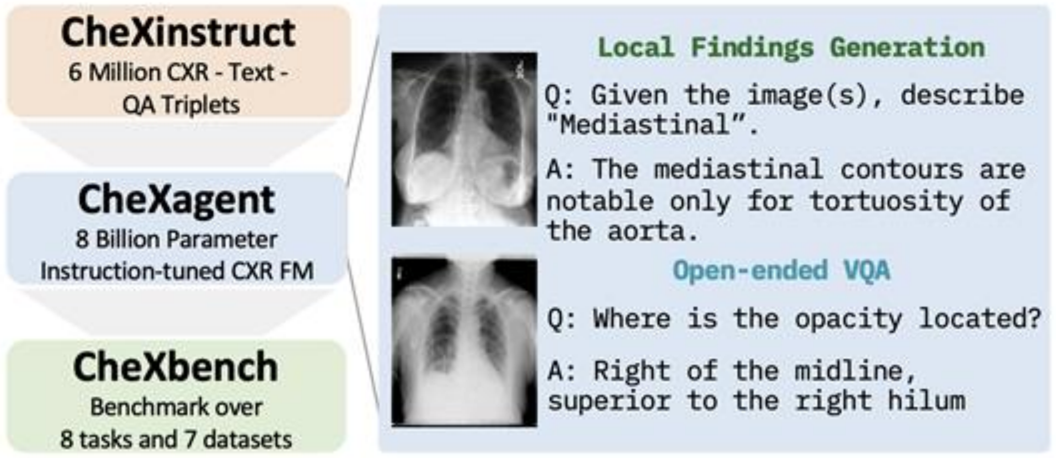


Saab et al. Capabilities of Gemini Models in Medicine. arXiv 2024.

Continuing:
Specialist (narrower domain) biomedical VLMs

CheXagent: VLM for CXR interpretation

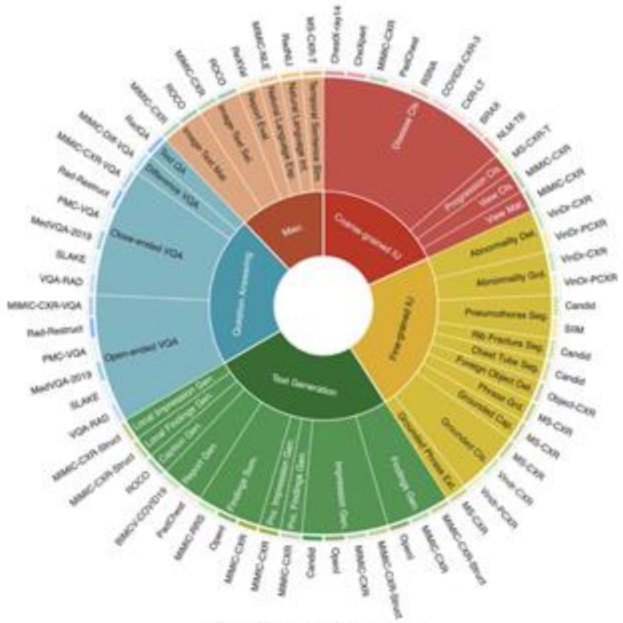
Contributed an instruction tuning dataset, instruction-tuned model, and evaluation benchmark for CXR interpretation tasks spanning coarse-grained image understanding, fine-grained image understanding, question answering, text (report) generation, and other miscellaneous categories.


CheXinstruct
6 Million CXR - Text - QA Triplets
CheXagent
8 Billion Parameter Instruction-tuned CXR FM
CheXbench
Benchmark over 8 tasks and 7 datasets
Local Findings Generation
Q: Given the image(s), describe "Mediastinal".
A: The mediastinal contours are notable only for tortuosity of the aorta.
Open-ended VQA
Q: Where is the opacity located?
A: Right of the midline, superior to the right hilum

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

CheXagent: VLM for CXR interpretation

- CheXinstruct instruction-tuning dataset contains 6M instruction-image-answer triplets curated from across 34 tasks and 65 existing datasets

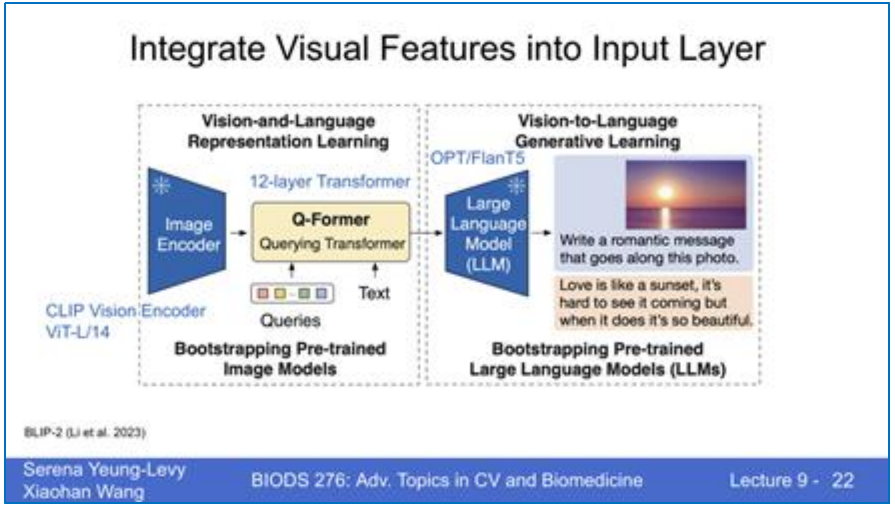


(a) Overview of CheXinstruct

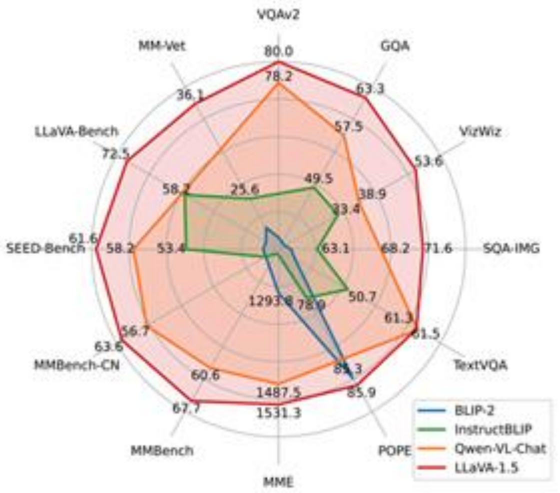
Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

CheXagent: VLM for CXR interpretation

- CheXagent model is based on the BLIP-2 architecture (one of the VLM models mentioned last lecture, under the "integrate visual features into input layer" category together with LLaVA (but LLaVA-1.5 generally outperforms BLIP-2))

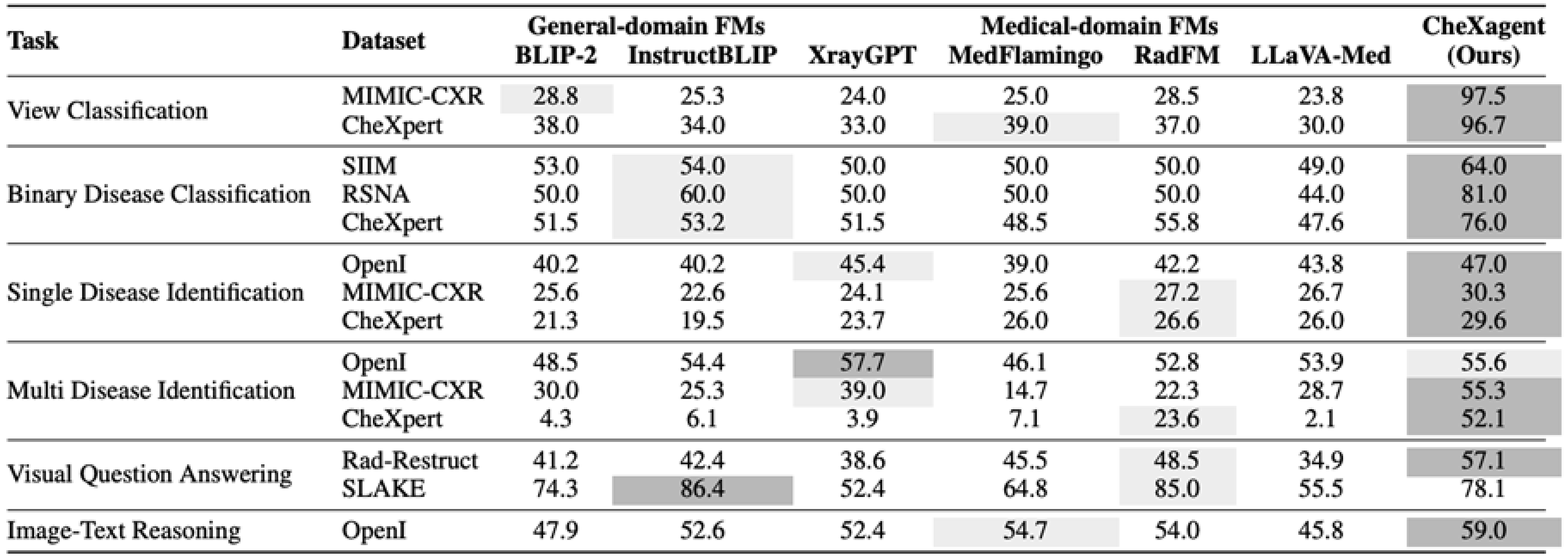


Integrate Visual Features into Input Layer
Vision-and-Language Representation Learning
Vision-to-Language Generative Learning
OPT/FlanT5
Image Encoder
12-layer Transformer
Q-Former Querying Transformer
Large Language Model (LLM)
Write a romantic message that goes along this photo.
Love is like a sunset, it's hard to see it coming but when it does it's so beautiful.
Queries
Text
CLIP Vision Encoder ViT-L/14
Bootstrapping Pre-trained Image Models
Bootstrapping Pre-trained Large Language Models (LLMs)
BLIP-2 (Li et al. 2023)
Serena Yeung-Levy Xiaohan Wang
BIODS 276: Adv. Topics in CV and Biomedicine
Lecture 9 - 22



Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

Evaluation on image perception tasks

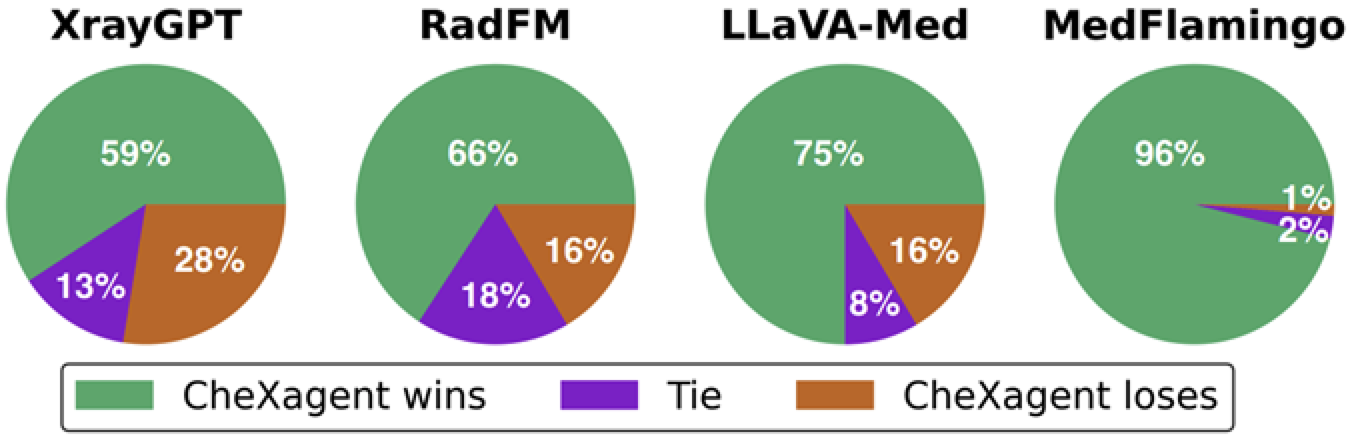


Task	Dataset	General-domain FMs		Medical-domain FMs				CheXagent
		BLIP-2	InstructBLIP	XrayGPT	MedFlamingo	RadFM	LLaVA-Med	(Ours)
View Classification	MIMIC-CXR	28.8	25.3	24.0	25.0	28.5	23.8	97.5
	CheXpert	38.0	34.0	33.0	39.0	37.0	30.0	96.7
Binary Disease Classification	SIIM	53.0	54.0	50.0	50.0	50.0	49.0	64.0
	RSNA	50.0	60.0	50.0	50.0	50.0	44.0	81.0
	CheXpert	51.5	53.2	51.5	48.5	55.8	47.6	76.0
Single Disease Identification	OpenI	40.2	40.2	45.4	39.0	42.2	43.8	47.0
	MIMIC-CXR	25.6	22.6	24.1	25.6	27.2	26.7	30.3
	CheXpert	21.3	19.5	23.7	26.0	26.6	26.0	29.6
Multi Disease Identification	OpenI	48.5	54.4	57.7	46.1	52.8	53.9	55.6
	MIMIC-CXR	30.0	25.3	39.0	14.7	22.3	28.7	55.3
	CheXpert	4.3	6.1	3.9	7.1	23.6	2.1	52.1
Visual Question Answering	Rad-Restruct	41.2	42.4	38.6	45.5	48.5	34.9	57.1
	SLAKE	74.3	86.4	52.4	64.8	85.0	55.5	78.1
Image-Text Reasoning	OpenI	47.9	52.6	52.4	54.7	54.0	45.8	59.0

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

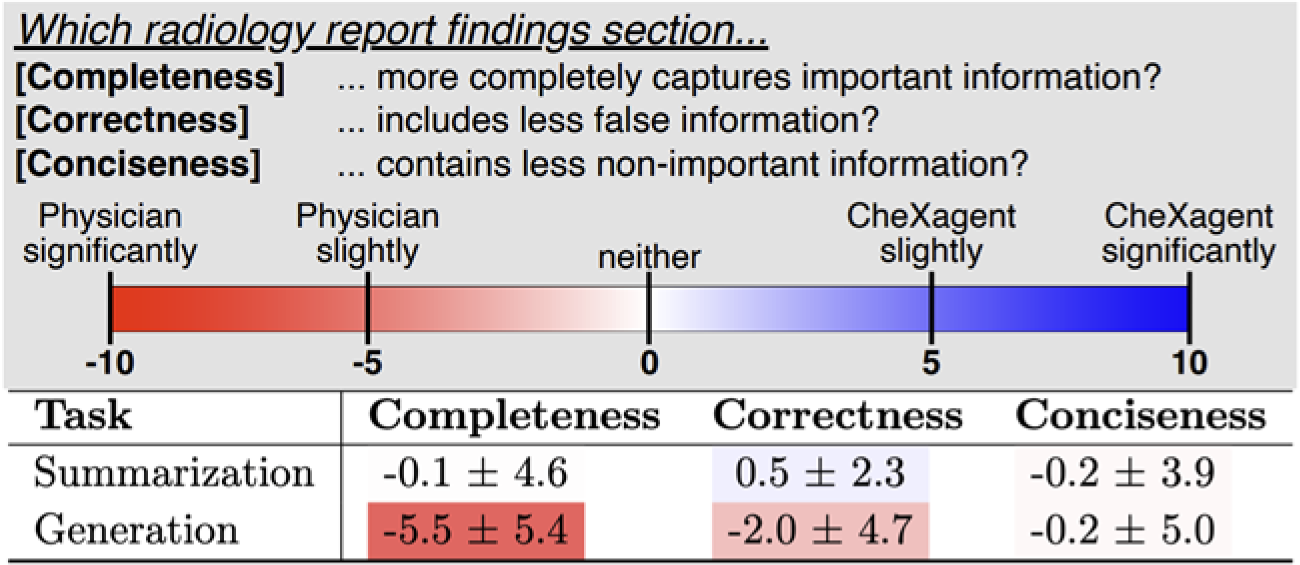
Evaluation on findings generation in reports

- Automated GPT-4 based evaluation, where GPT-4 is provided with a reference report, findings generated by CheXagent, and findings generated by comparison models, and prompted to select the report with the highest accuracy



Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

Evaluation on findings generation in reports: reader study (assessment by five radiologists)



Which radiology report findings section...

[Completeness] ... more completely captures important information?

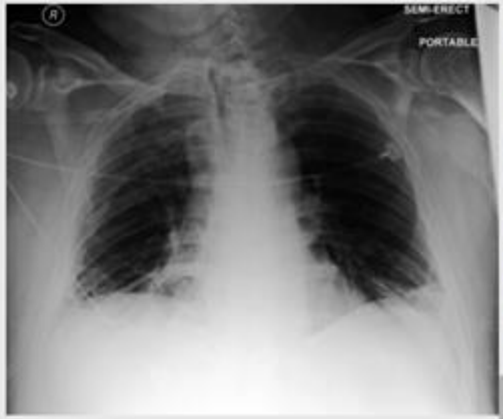
[Correctness] ... includes less false information?

[Conciseness] ... contains less non-important information?

Task	Completeness	Correctness	Conciseness
Summarization	-0.1 ± 4.6	0.5 ± 2.3	-0.2 ± 3.9
Generation	-5.5 ± 5.4	-2.0 ± 4.7	-0.2 ± 5.0

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

Evaluation on findings generation in reports: qualitative comparison of CheXagent with human physician report



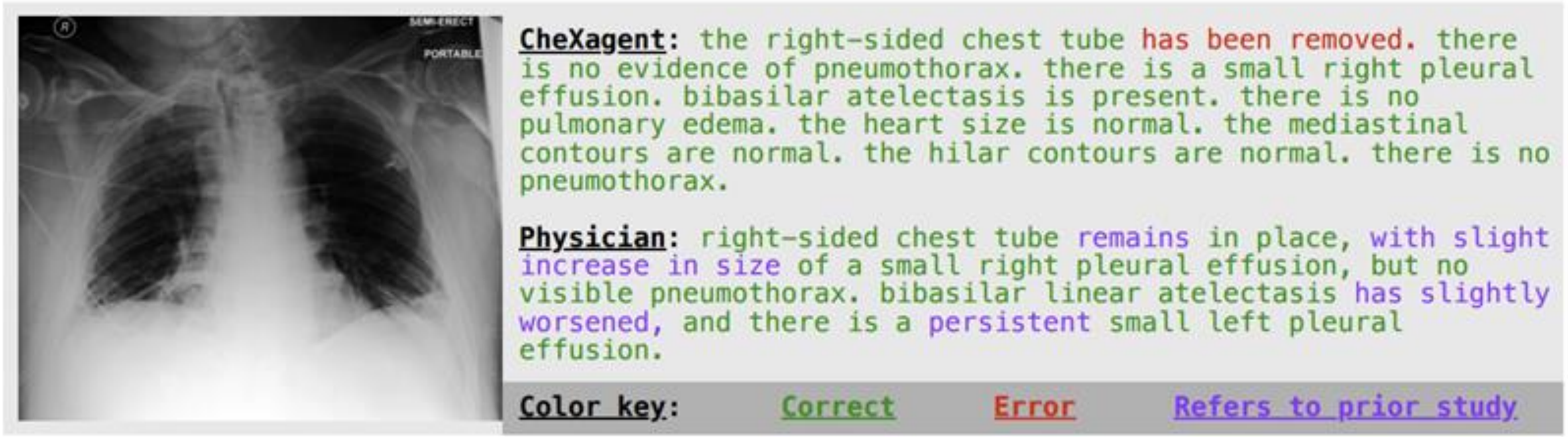
CheXagent: the right-sided chest tube has been removed. there is no evidence of pneumothorax. there is a small right pleural effusion. bibasilar atelectasis is present. there is no pulmonary edema. the heart size is normal. the mediastinal contours are normal. the hilar contours are normal. there is no pneumothorax.

Physician: right-sided chest tube remains in place, with slight increase in size of a small right pleural effusion, but no visible pneumothorax. bibasilar linear atelectasis has slightly worsened, and there is a persistent small left pleural effusion.

Color key: Correct Error Refers to prior study

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

Evaluation on findings generation in reports: qualitative comparison of CheXagent with human physician report



CheXagent: the right-sided chest tube has been removed. there is no evidence of pneumothorax. there is a small right pleural effusion. bibasilar atelectasis is present. there is no pulmonary edema. the heart size is normal. the mediastinal contours are normal. the hilar contours are normal. there is no pneumothorax.

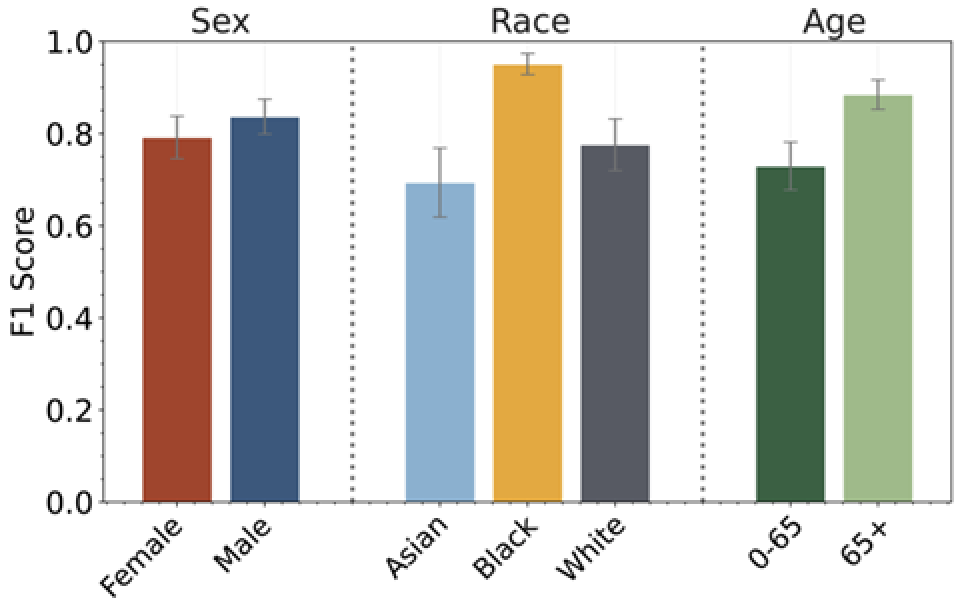
Physician: right-sided chest tube remains in place, with slight increase in size of a small right pleural effusion, but no visible pneumothorax. bibasilar linear atelectasis has slightly worsened, and there is a persistent small left pleural effusion.

Color key: Correct Error Refers to prior study

Physician reports more frequently refer to past studies

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

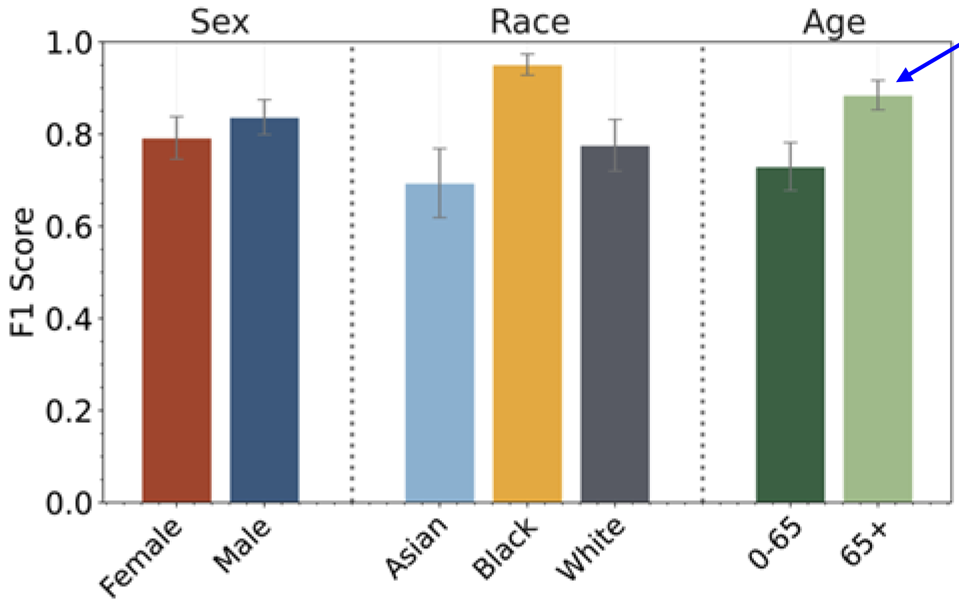
Example of fairness evaluation: subgroup performance comparison on cardiomegaly classification



Authors state caveat about limited sample sizes

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

Example of fairness evaluation: subgroup performance comparison on cardiomegaly classification



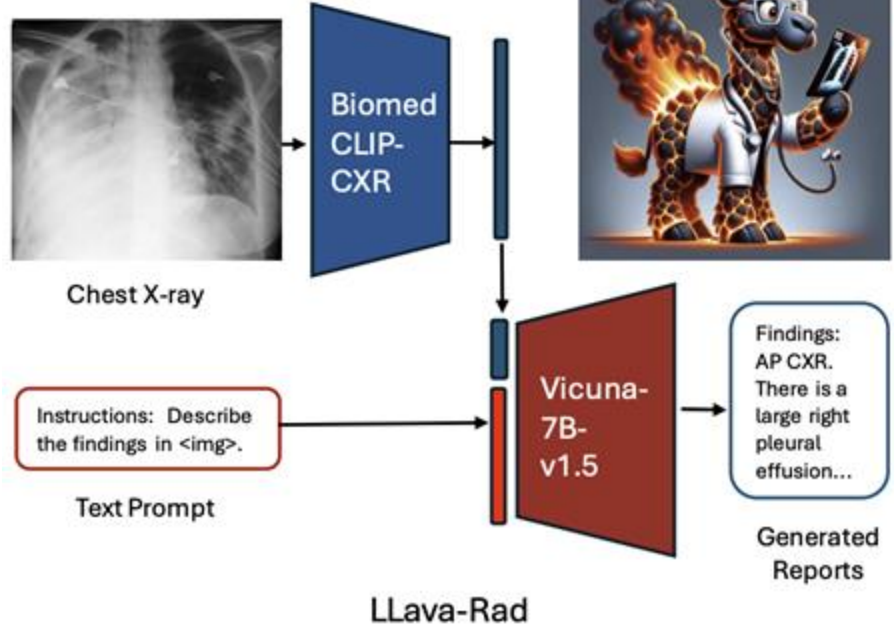
Could be partially attributed to higher prevalence of cardiomegaly in older patients

Authors state caveat about limited sample sizes

Chen et al. CheXagent: Towards a Foundation Model for Chest X-Ray Interpretation. arXiv 2024.

LLaVA-Rad: specialized CXR VLM that improves over CheXpert

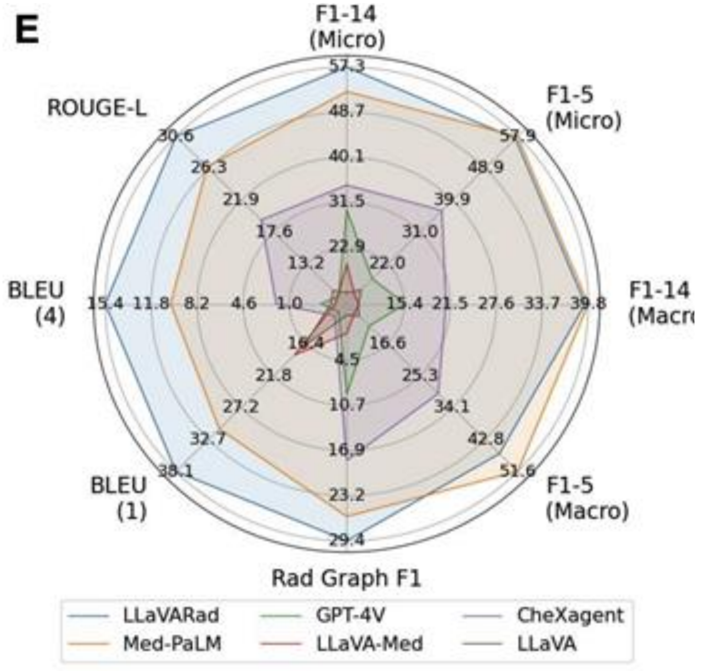
- Based on LLaVA (similar to LLaVA-Med), but trains from scratch (instead of fine-tuning LLaVA) since it uses a biomedical encoder (Biomed CLIP-CXR) instead of LLaVA's CLIP encoder.
- Trains using 697K image-report pairs from a collection of datasets

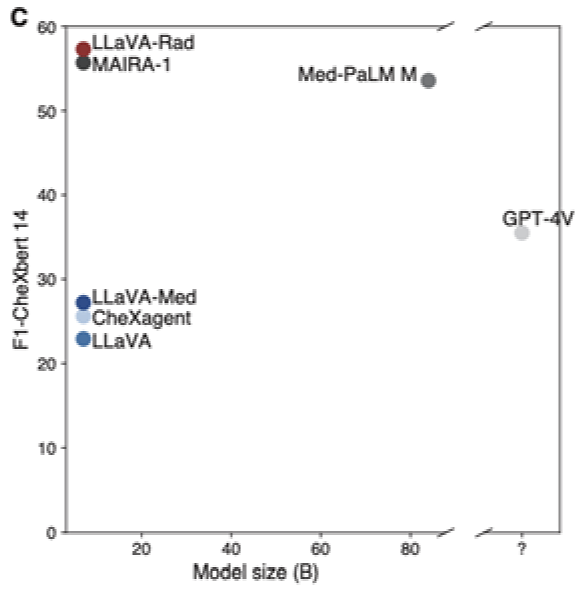

Chest X-ray
Biomed CLIP-CXR
Instructions: Describe the findings in <img>.
Text Prompt
Vicuna-7B-v1.5
Findings: AP CXR. There is a large right pleural effusion...
Generated Reports
LLava-Rad

Chaves et al. Towards a clinically accessible radiology multimodal model: open-access and lightweight, with automatic evaluation. arXiv 2024.

LLaVA-Rad: specialized CXR VLM that improves over CheXpert

Outperforms LLaVA-Med, CheXagent, and GPT-4V, among other models (but probably not Med-Gemini)



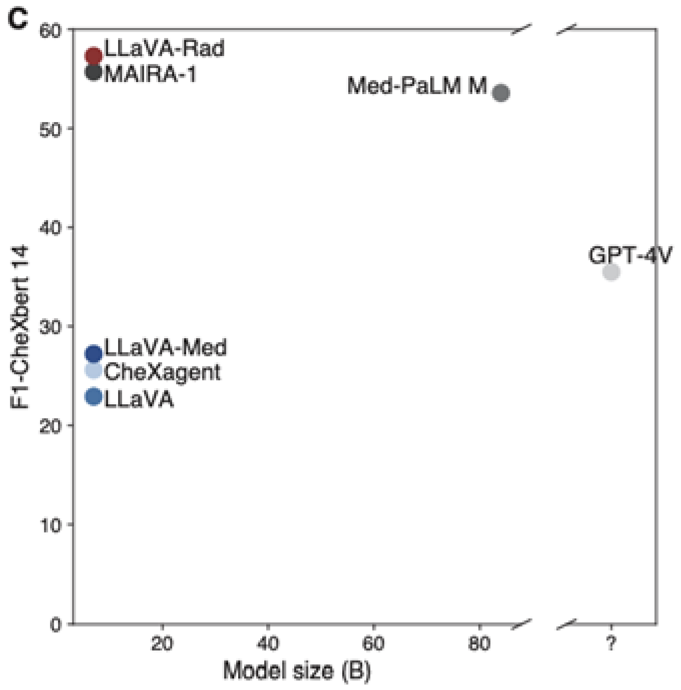


Results

Chaves et al. Towards a clinically accessible radiology multimodal model: open-access and lightweight, with automatic evaluation. arXiv 2024.

LLaVA-Rad: specialized CXR VLM that improves over CheXpert

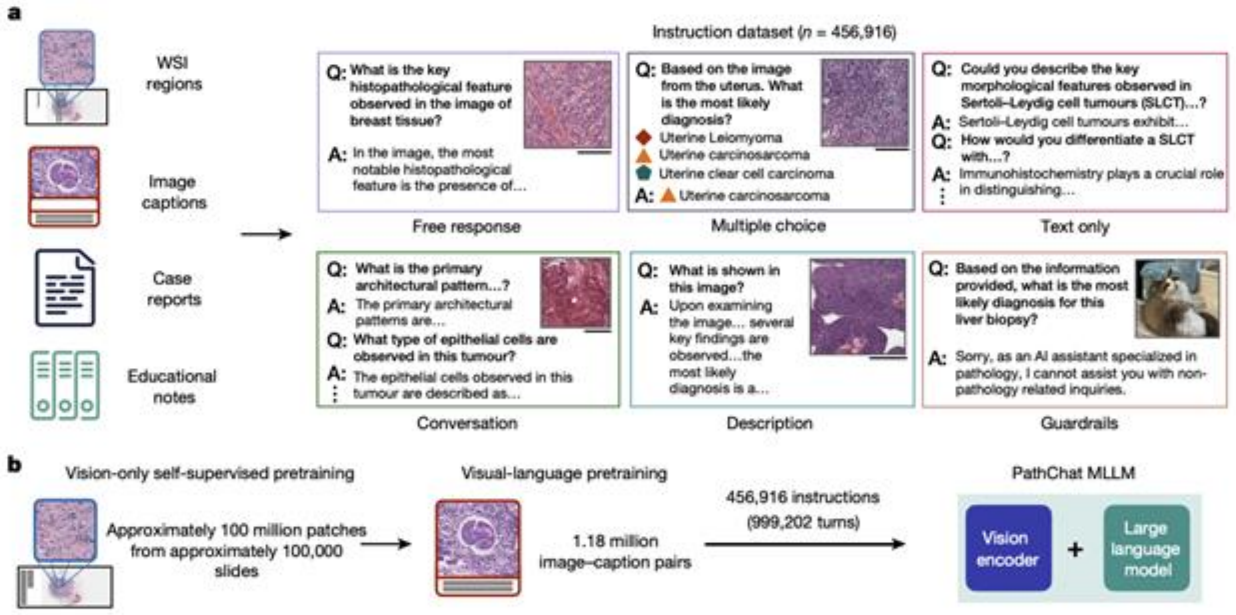
- Training and inference is reasonably fast:
    - Training takes one day on an 8xA100 server
    - Inference can be run on single (older generation) GPUs



Chaves et al. Towards a clinically accessible radiology multimodal model: open-access and lightweight, with automatic evaluation. arXiv 2024.

PathChat: specialized VLM for pathology

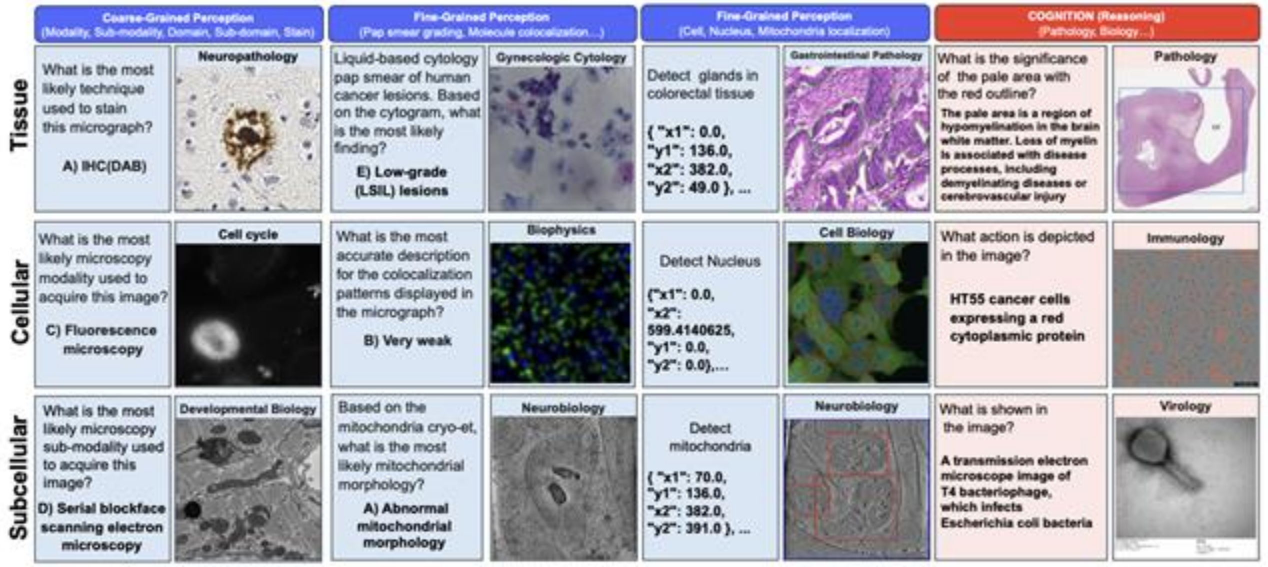
Trained on over 450K instructions from pathology. Will cover further in the discussion presentations!


a
WSI regions
Image captions
Case reports
Educational notes
Instruction dataset (n = 456,916)
Q: What is the key histopathological feature observed in the image of breast tissue?
A: In the image, the most notable histopathological feature is the presence of…
Free response
Q: Based on the image from the uterus. What is the most likely diagnosis?
◆ Uterine Leiomyoma
▲ Uterine carcinosarcoma
⬟ Uterine clear cell carcinoma
A: ▲ Uterine carcinosarcoma
Multiple choice
Q: Could you describe the key morphological features observed in Sertoli–Leydig cell tumours (SLCT)…?
A: Sertoli-Leydig cell tumours exhibit…
Q: How would you differentiate a SLCT with…?
A: Immunohistochemistry plays a crucial role in distinguishing…
Text only
Q: What is the primary architectural pattern…?
A: The primary architectural patterns are…
Q: What type of epithelial cells are observed in this tumour?
A: The epithelial cells observed in this tumour are described as…
Conversation
Q: What is shown in this image?
A: Upon examining the image… several key findings are observed…the most likely diagnosis is a…
Description
Q: Based on the information provided, what is the most likely diagnosis for this liver biopsy?
A: Sorry, as an AI assistant specialized in pathology, I cannot assist you with non-pathology related inquiries.
Guardrails
b
Vision-only self-supervised pretraining
Approximately 100 million patches from approximately 100,000 slides
Visual-language pretraining
1.18 million image–caption pairs
456,916 instructions (999,202 turns)
PathChat MLLM
Vision encoder
+
Large language model

Lu et al. A multimodal generative AI copilot for human pathology. Nature 2024.

μ-Bench: benchmarking VLMs on microscopy image interpretation tasks across scales

Evaluated proprietary and open-source VLMs on 22 biomedical tasks across various scientific disciplines (biology, pathology), microscopy modalities (electron, fluorescence, light), and scales (subcellular, cellular, tissue)

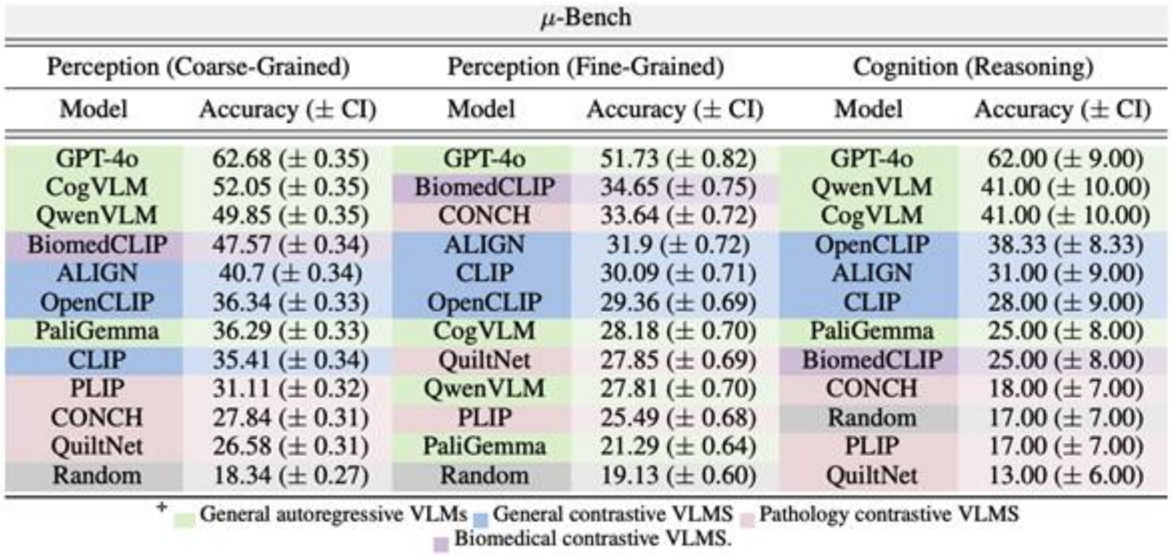


Coarse-Grained Perception (Modality, Sub-modality, Domain, Sub-domain, Stain)
Fine-Grained Perception (Pap smear grading, Molecule colocalization...)
Fine-Grained Perception (Cell, Nucleus, Mitochondria localization)
COGNITION (Reasoning) (Pathology, Biology...)
Tissue
Cellular
Subcellular
What is the most likely technique used to stain this micrograph?
A) IHC(DAB)
Neuropathology
Liquid-based cytology pap smear of human cancer lesions. Based on the cytogram, what is the most likely finding?
E) Low-grade (LSIL) lesions
Gynecologic Cytology
Detect glands in colorectal tissue
{ "x1": 0.0, "y1": 136.0, "x2": 382.0, "y2": 49.0 }, ...
Gastrointestinal Pathology
What is the significance of the pale area with the red outline?
The pale area is a region of hypomyelination in the brain white matter. Loss of myelin is associated with disease processes, including demyelinating diseases or cerebrovascular injury
Pathology
What is the most likely microscopy modality used to acquire this image?
C) Fluorescence microscopy
Cell cycle
What is the most accurate description for the colocalization patterns displayed in the micrograph?
B) Very weak
Biophysics
Detect Nucleus
{"x1": 0.0, "x2": 599.4140625, "y1": 0.0, "y2": 0.0},...
Cell Biology
What action is depicted in the image?
HT55 cancer cells expressing a red cytoplasmic protein
Immunology
What is the most likely microscopy sub-modality used to acquire this image?
D) Serial blockface scanning electron microscopy
Developmental Biology
Based on the mitochondria cryo-et, what is the most likely mitochondrial morphology?
A) Abnormal mitochondrial morphology
Neurobiology
Detect mitochondria
{ "x1": 70.0, "y1": 136.0, "x2": 382.0, "y2": 391.0 }, ...
Neurobiology
What is shown in the image?
A transmission electron microscope image of T4 bacteriophage, which infects Escherichia coli bacteria
Virology

Lozano et al. μ-Bench: A Vision-Language Benchmark for Microscopy Understanding. NeurIPS Datasets and Benchmarks 2024.

μ-Bench: benchmarking VLMs on microscopy image interpretation tasks across scales

GPT-4o still outperforms all other compared models, and non-specialist biomedical models often outperform specialist counterparts, suggesting room for improvement in how to leverage specialist domain data



μ-Bench					
Perception (Coarse-Grained)		Perception (Fine-Grained)		Cognition (Reasoning)	
Model	Accuracy (± CI)	Model	Accuracy (± CI)	Model	Accuracy (± CI)
GPT-4o	62.68 (± 0.35)	GPT-4o	51.73 (± 0.82)	GPT-4o	62.00 (± 9.00)
CogVLM	52.05 (± 0.35)	BiomedCLIP	34.65 (± 0.75)	QwenVLM	41.00 (± 10.00)
QwenVLM	49.85 (± 0.35)	CONCH	33.64 (± 0.72)	CogVLM	41.00 (± 10.00)
BiomedCLIP	47.57 (± 0.34)	ALIGN	31.9 (± 0.72)	OpenCLIP	38.33 (± 8.33)
ALIGN	40.7 (± 0.34)	CLIP	30.09 (± 0.71)	ALIGN	31.00 (± 9.00)
OpenCLIP	36.34 (± 0.33)	OpenCLIP	29.36 (± 0.69)	CLIP	28.00 (± 9.00)
PaliGemma	36.29 (± 0.33)	CogVLM	28.18 (± 0.70)	PaliGemma	25.00 (± 8.00)
CLIP	35.41 (± 0.34)	QuiltNet	27.85 (± 0.69)	BiomedCLIP	25.00 (± 8.00)
PLIP	31.11 (± 0.32)	QwenVLM	27.81 (± 0.70)	CONCH	18.00 (± 7.00)
CONCH	27.84 (± 0.31)	PLIP	25.49 (± 0.68)	Random	17.00 (± 7.00)
QuiltNet	26.58 (± 0.31)	PaliGemma	21.29 (± 0.64)	PLIP	17.00 (± 7.00)
Random	18.34 (± 0.27)	Random	19.13 (± 0.60)	QuiltNet	13.00 (± 6.00)

+ 🟩 General autoregressive VLMs 🟦 General contrastive VLMS 🟥 Pathology contrastive VLMS 🟪 Biomedical contrastive VLMS.

Lozano et al. μ-Bench: A Vision-Language Benchmark for Microscopy Understanding. NeurIPS Datasets and Benchmarks 2024.

We covered:

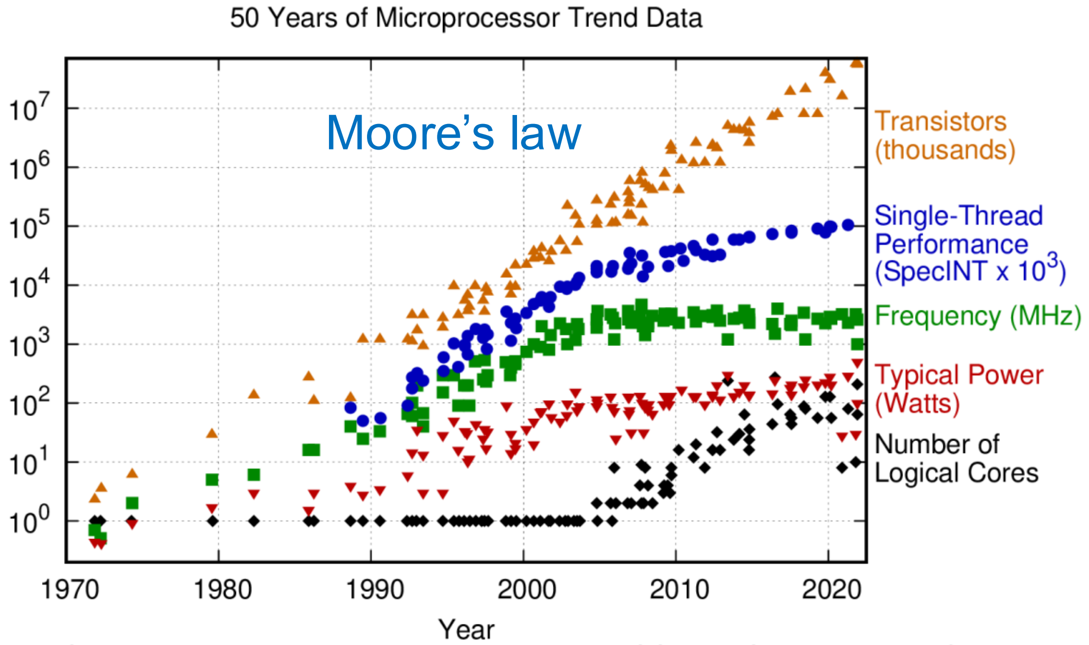
Examples of vision-language generative models in biomedicine:

- Generalist (broad domain) biomedical VLMs
    - LLaVA-Med (open-source) and Med-Gemini (SoTA)
- Specialist biomedical VLMs:
    - CXR: CheXagent (earlier, though still 2024), LLaVA-Rad (improved)
    - Pathology: PathChat
    - Benchmarking VLMs across cell and tissue imaging domains: μ-Bench

Next:

A Deeper Dive into GPUs and Compute

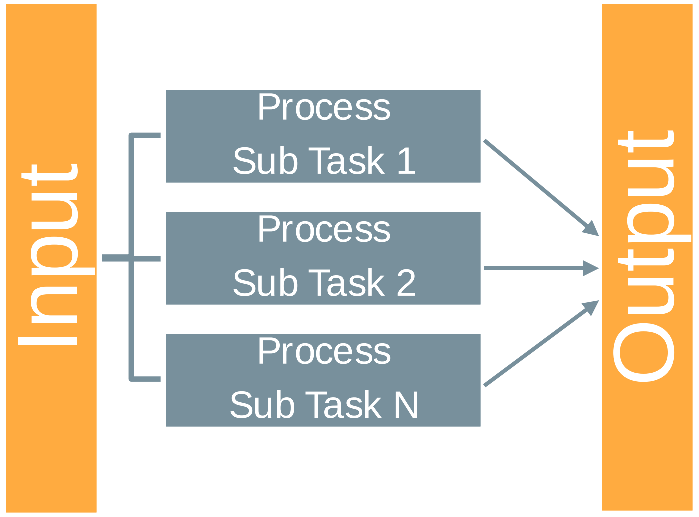
Why GPUs?



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

The number of transistors per chip doubles roughly every 2 years.
However, it can no longer be explored by the core frequency due to the power consumption limits.

Computing in Parallel



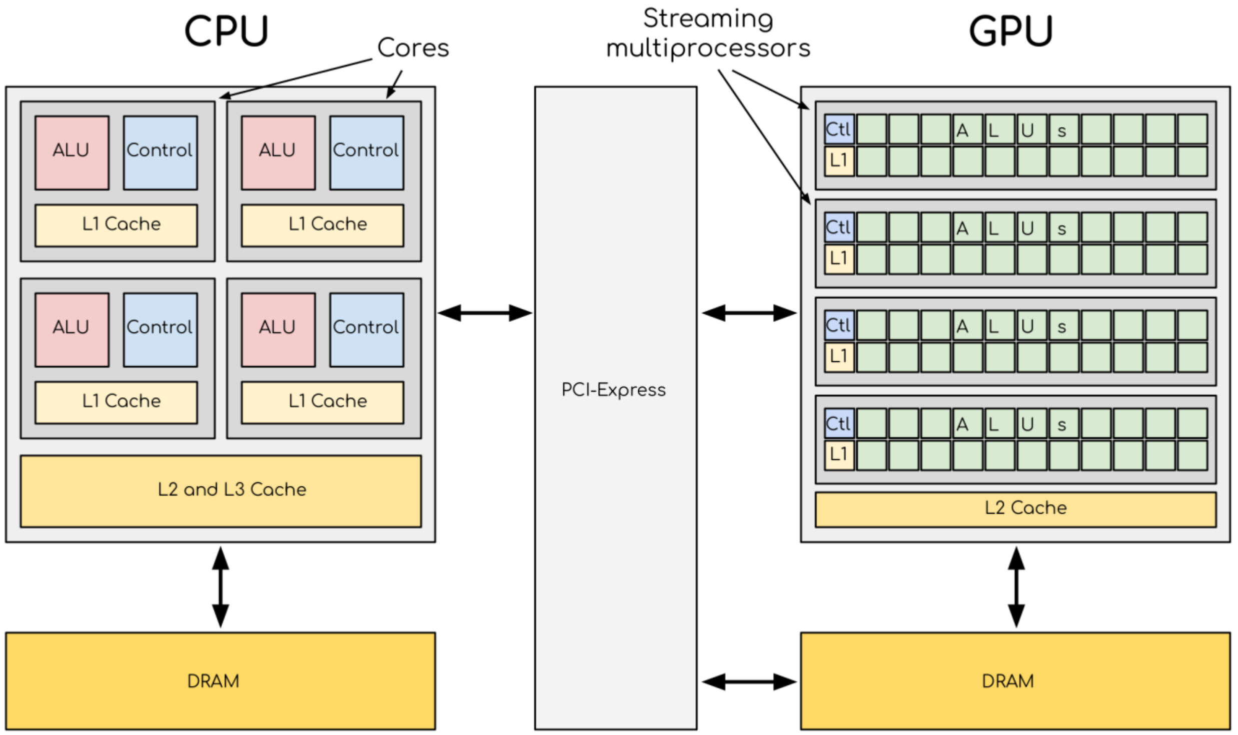
- Split a computational problem into smaller subtasks
- Many subtasks can then be solved simultaneously by multiple processing units

What is a GPU?

Central Processing Unit

Graphics Processing Unit

Arithmetic Logic Unit



https://enccs.github.io/gpu-programming/2-gpu-ecosystem/

What is a GPU?

L1 Cache / Shared Memory = cache located inside the core

- Register spilling

L2 Cache = in many modern processors, dedicated to a particular core or set of cores and slower than L1

- Like L1, the L2 cache is intended to speed up subsequent reloads
- Unlike L1, there is just one L2 that is shared by all the SMs
- Situated in the path of data moving on or off the device via PCIe or NVLink

DRAM = Dynamic Random Access Memory

- VRAM (video RAM) is a type of DRAM that is meant to ensure the even and smooth execution of graphics display
  - Farther from the GPU processing cores and amount dependent on GPU architecture
- DRAM units sit very close to the GPU chip itself
- Represents the bulk of the main memory of the device, equivalent to RAM in a CPU-based processor

What is a GPU?

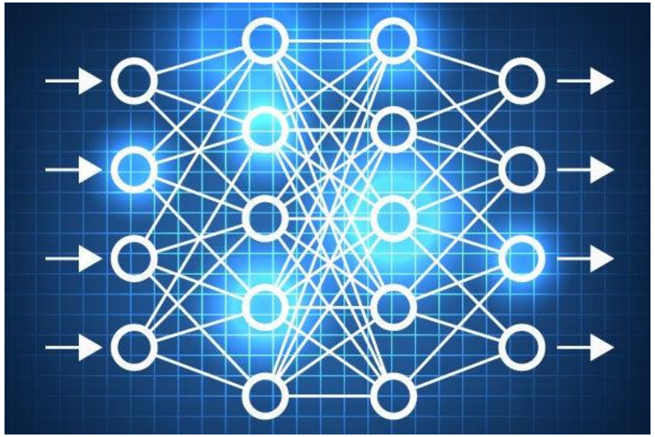
Central Processing Unit	Graphics Processing Unit
CPU	**GPU**
General purpose	Highly specialized for parallelism
Good for serial processing	Good for parallel processing
Great for task parallelism	Great for data parallelism
Low latency per thread	High-throughput
Large area dedicated cache and control	Hundreds of floating-point execution units

https://enccs.github.io/gpu-programming/2-gpu-ecosystem/

Use Cases of GPUs

- GPUs were designed for 3D Rendering (Video Games)
- To represent a 3D triangle mesh, given a triangle, determinate where it lies on
- For all output pixels covered by the triangle, compute the color of each pixels

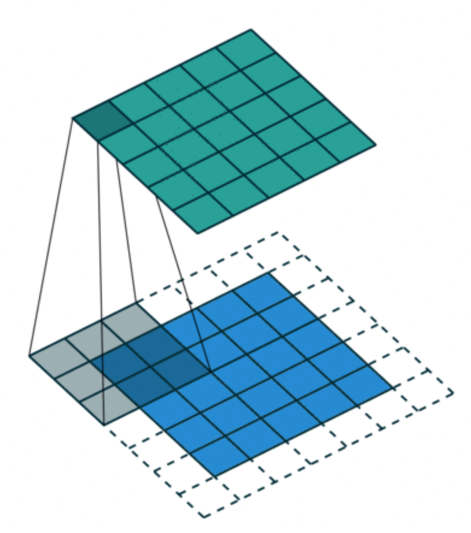
Use Cases of GPUs

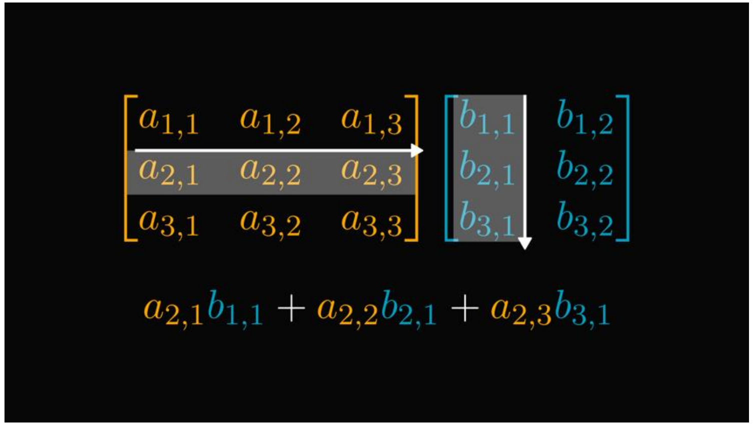


Machine Learning and Deep Learning

Blockchain

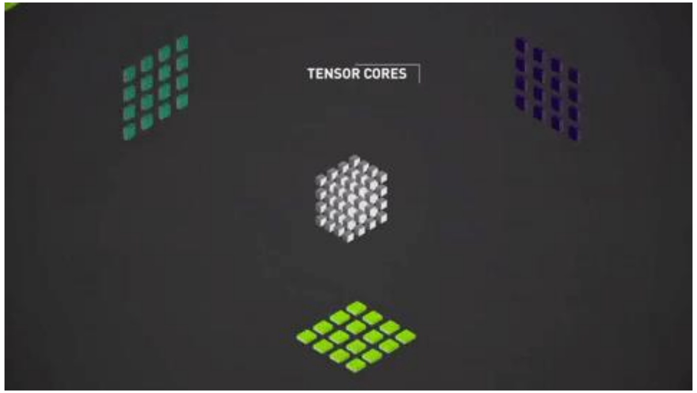
GPUs are Essential for Deep Learning




$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix}$
$a_{2,1}b_{1,1} + a_{2,2}b_{2,1} + a_{2,3}b_{3,1}$

Massive Parallel: GPUs are optimized for training deep learning models as they can process multiple computations simultaneously

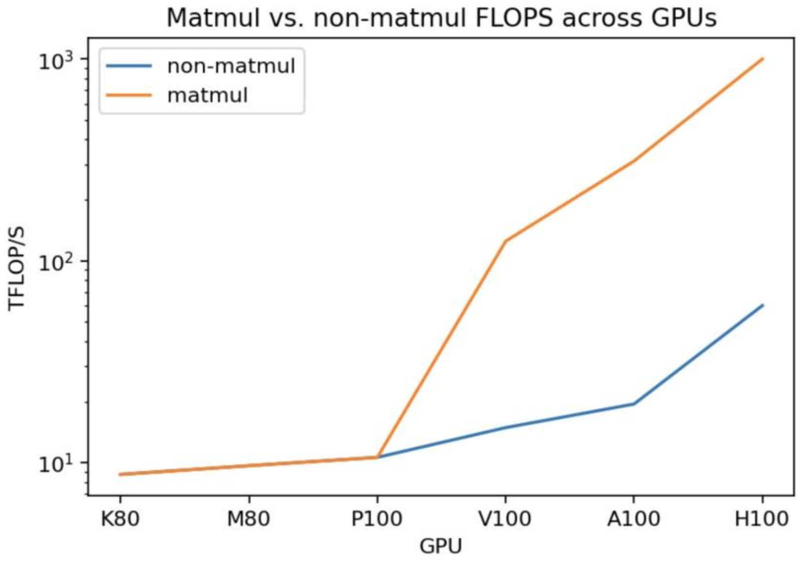
GPUs are Essential for Deep Learning



CUDA Cores	Tensor Cores
High accuracy at the cost of compute speed	Slight hit in accuracy but fast computational speeds
General-purpose cores for graphics, crypto, ML, etc.	Specialized cores for high-end computations and allows mixed-precision training
Perform 1 operation per clock cycle	Can perform multiple operations per clock cycle

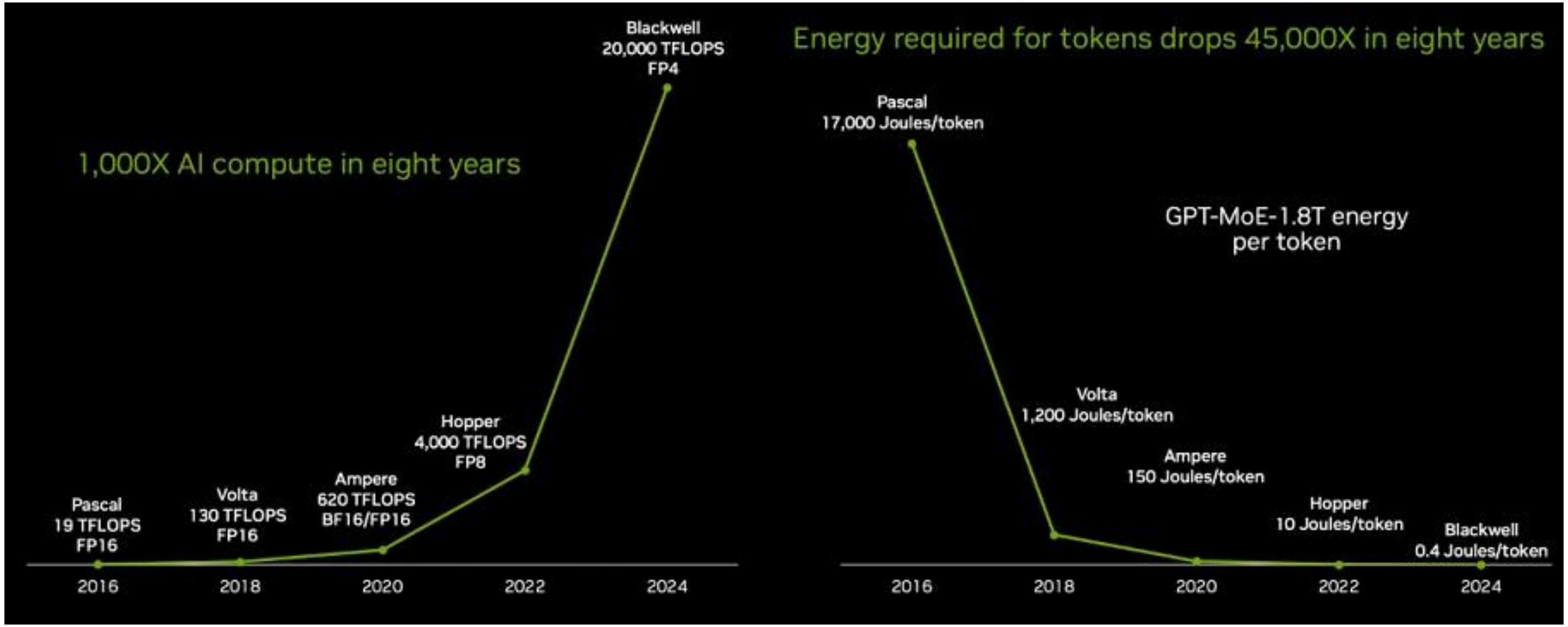
Special architecture to accelerate matrix multiply and accumulate operations for machine learning and scientific applications

GPUs are Essential for Deep Learning



Fast matrix multiplication: special cores >10x faster than other fp ops

GPU Generation



GPU Type

GPU	FPP	Memory	Bandwidth
1. NVIDIA H100 NVL	FP16* - 1,671 TFLOP FP32* - 835 TFLOPS FP64 - 60 TFLOPS	94 GB HBM3	3.9 TB/s
2. AMD Radeon Instinct MI300	FP16 - 383 TFLOPS FP32 - 47.87 TFLOPS FP64 - 47.87 TFLOPS	128 GB HBM2e	5.3 TB/s
3. NVIDIA A100 Tensor Core	FP16* - 624 TFLOPS FP32* - 312 TFLOPS FP64 - 19.5 TFLOPS	80 GB HBM2e	2TB/s
4. NVIDIA GeForce RTX 4090	FP16 - 82.58 TFLOPS FP32 - 82.58 TFLOPS FP64 - 1,290 GFLOPS	24 GB GDDR6X	1TB/s
5. NVIDIA Quadro RTX 8000	FP16 - 32.62 TFLOPS FP32 - 16.31 TFLOPS FP64 - 509.8 GFLOPS	48 GB GDDR6	672 GB/sec
6. NVIDIA RTX A6000	FP16 - 38.71 TFLOPS FP32 - 38.71 TFLOPS FP64 - 604.8 GLOPS	48 GB GDDR6	768 GB/s
7. NVIDIA GeForce RTX 3090 Ti	FP16 - 40 TFLOPS FP32 - 40 TFLOPS FP64 - 625 GFLOPS	24 GB GDDR6X	100 GB/s
8. NVIDIA GeForce RTX 4070	FP16 - 29.15 TFLOPS FP32 - 29.15 TFLOPS FP64 - 455.4 GFLOPS	12 GB GDDR6X	504.2 GB/s
9. Google TPU v4 Pod	bf 16 - 297 TFLOPS Int 8 - 393 TFLOPS	Different storage options	300 GB/s
10. NVIDIA GeForce RTX 3060 Ti	FP16 - 16.20 TFLOPS FP32 - 16.20 TFLOPS FP64 - 253.1 GFLOPS	8GB GDDR6	448 GB/s

GPU Requirement for Models

VLM LLAVA-1.5

- train all models with 8×A100s. 6 hours of pretraining and 20 hours of visual instruction tuning

Vision FM DINO

- A batch size of 1024, distributed over 16 GPUs. Trained for 3 days.

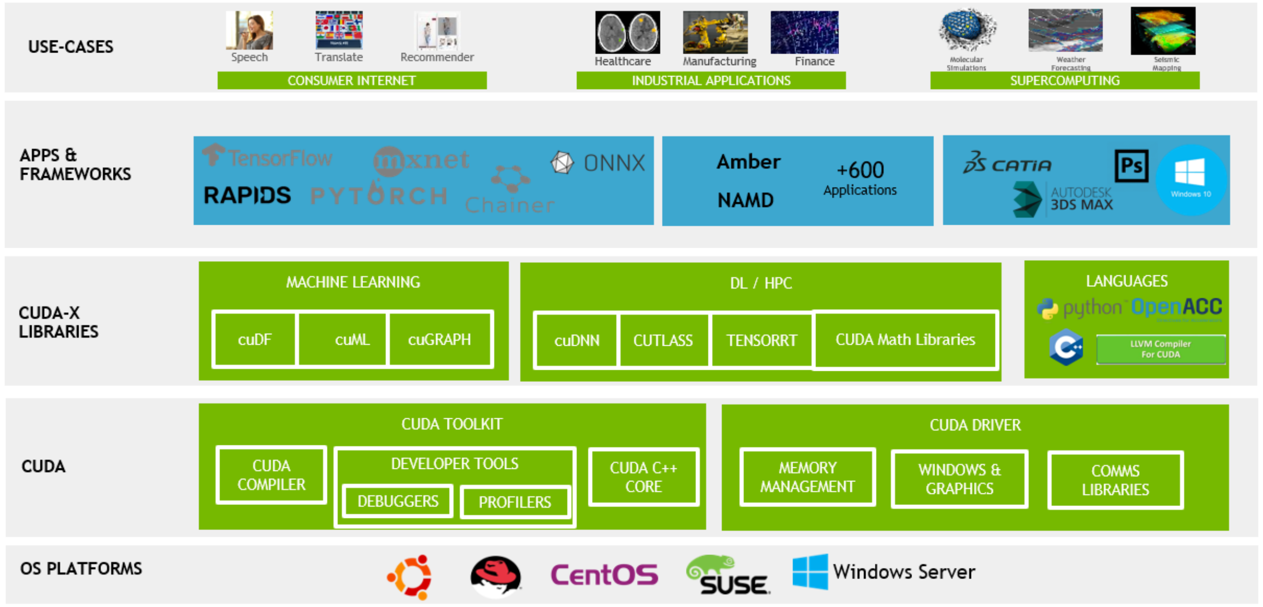
CLIP OPEN-CLIP

- ViT-L/14, LAINON-2B data, 384 A100 GPUs for 319 Hours

Vision Generative Model Stable Diffusion

- using 256 Nvidia A100 GPUs on AWS for a total of 150,000 GPU-hours

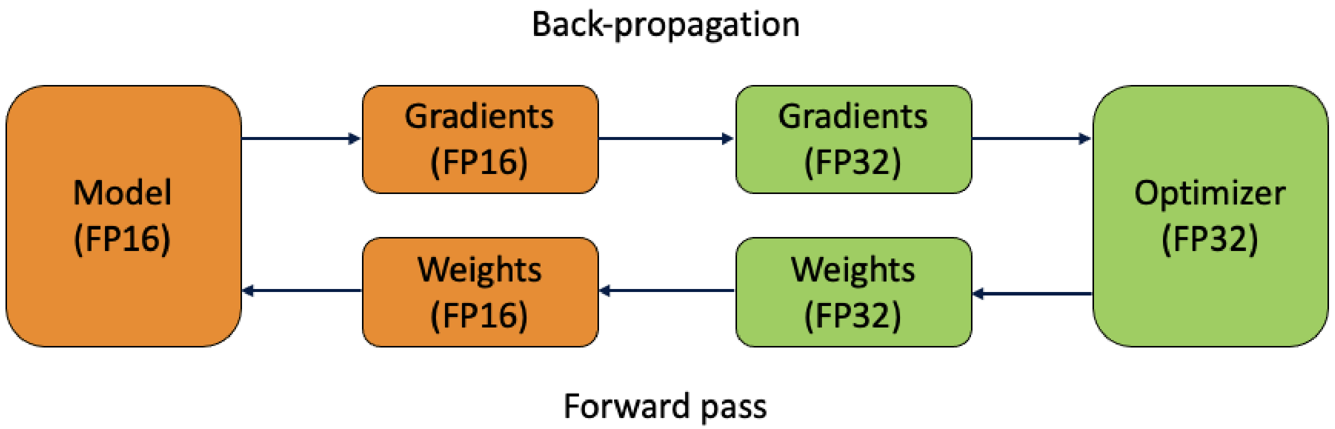
Programming Language for GPUs



CUDA: CUDA is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements for the execution of compute kernels

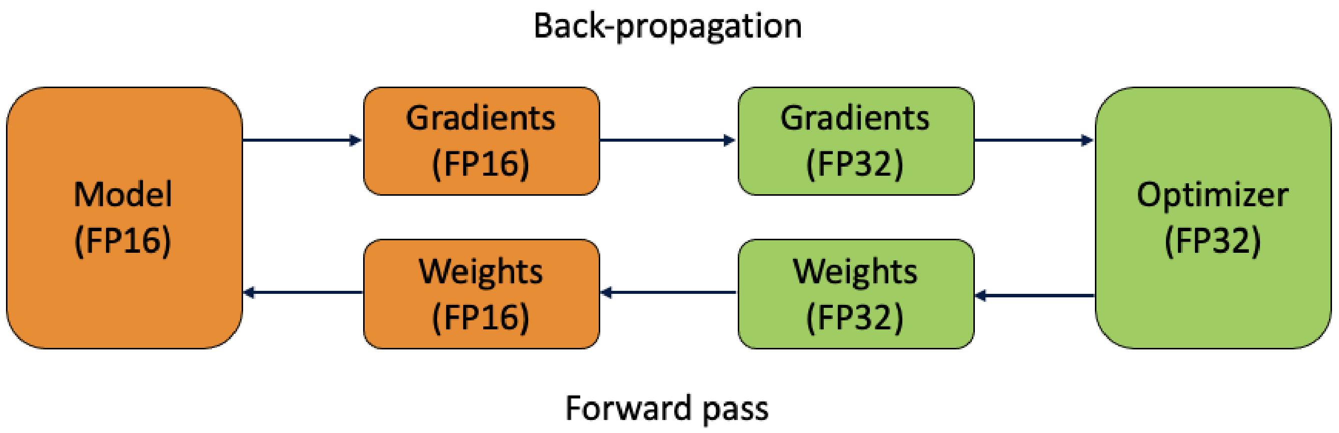
https://blogs.nvidia.com/blog/what-is-cuda-2/

Mixed Precision Training



Mixed precision training converts the weights to FP16 and calculates the gradients, before converting them back to FP32, multiplying by the learning rate and updating the weights in the optimizer.

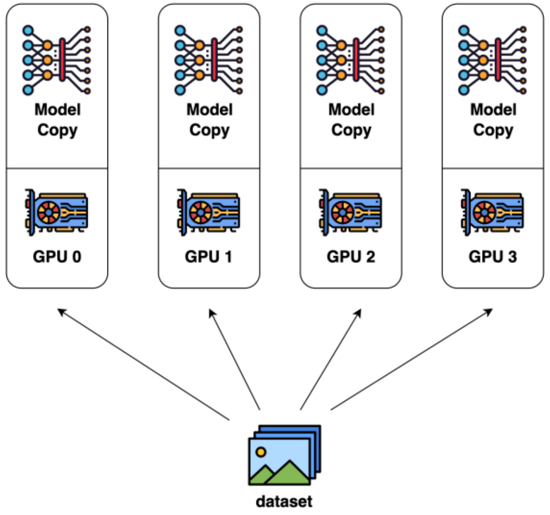
Mixed Precision Training

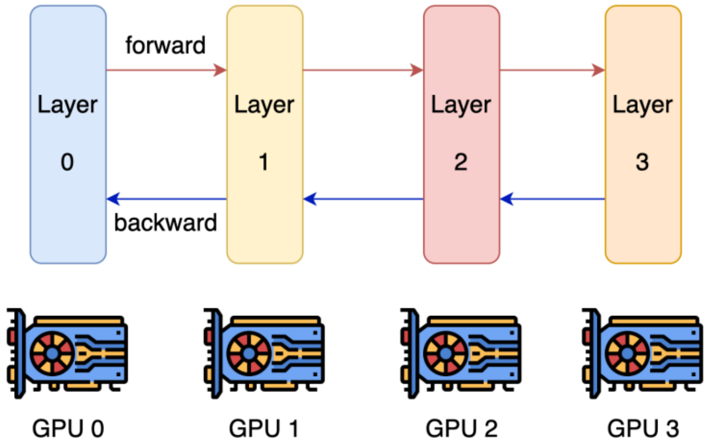


Automatic Mixed Precision
https://github.com/NVIDIA/apex

Parallelization

Distribute the training of large machine-learning models across multiple GPUs



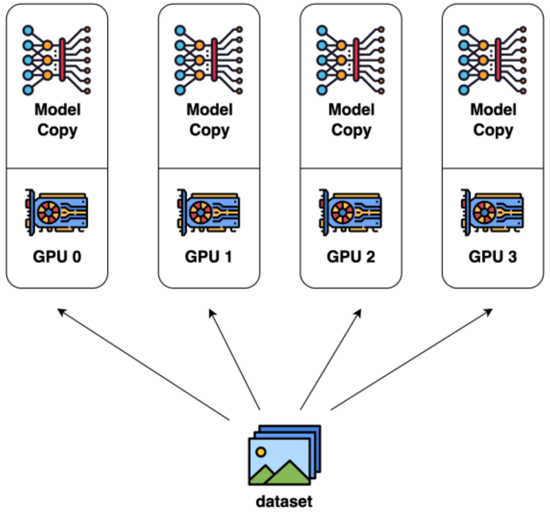


Data Parallel

Model Parallel

credit: CS229

Data Parallelization



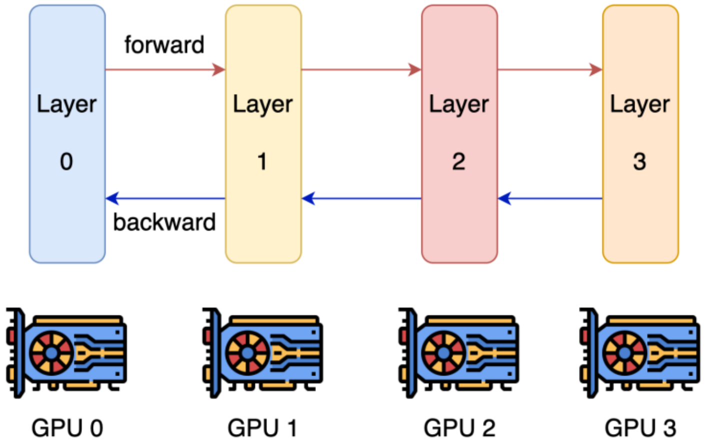
Data Parallel

Naïve data parallelization:
1. Copy model & optimizer on each GPU
2. Split data
3. Communicate and reduce (sum) gradients

data parallelism only works if batch size >= # GPUS

credit: CS229

Model Parallelization

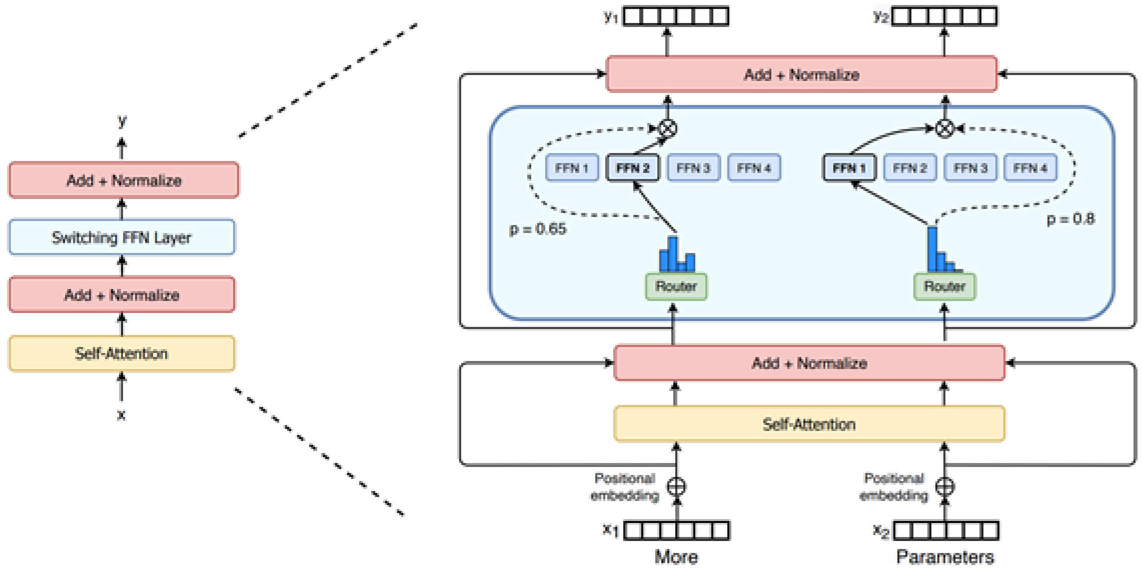


have every GPU take care of applying specific parameters (rather than updating)
• pipeline parallel: every GPU has different layer
• tensor parallel: split single matrix across GPUs and use partial sum

Sparsity for Large Models

- Architecture Sparsity: Mixture of Experts
- Token Sparsity: Token Pruning

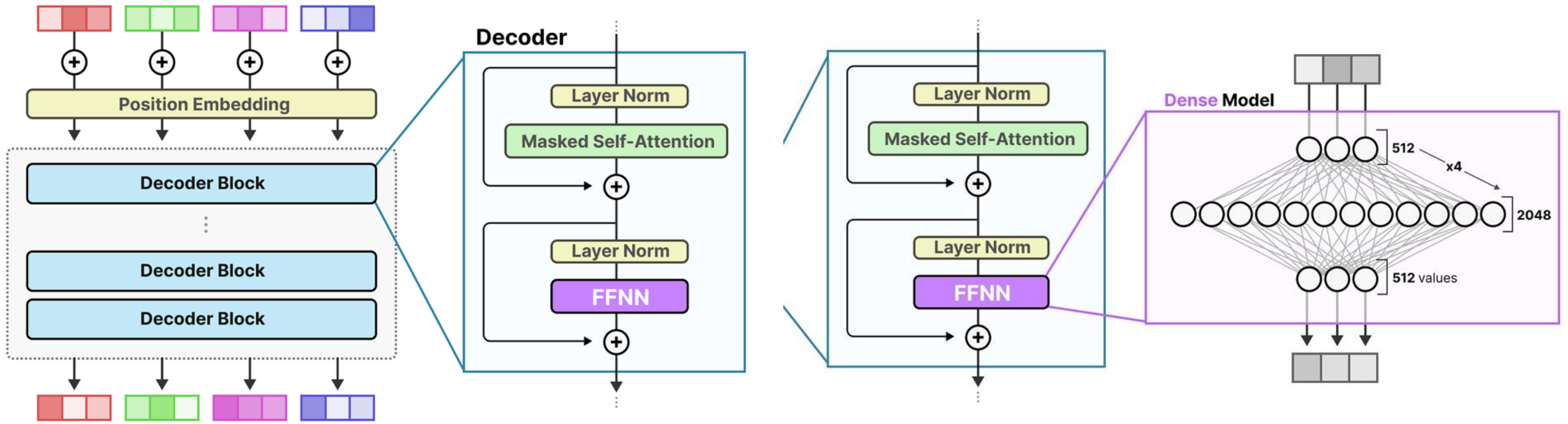
Mixture of Experts (MoE)



Sparse MoE layers are used instead of dense feed-forward network (FFN) layers.
A gate network or router, that determines which tokens are sent to which expert.

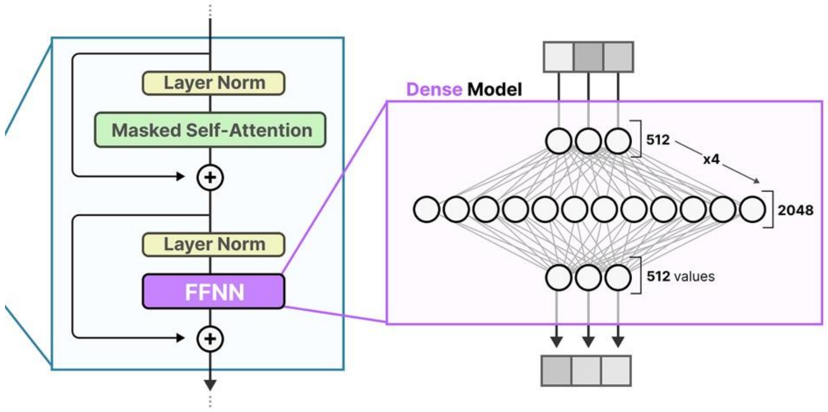
Switch Transformers (Fedus et al. 2021)

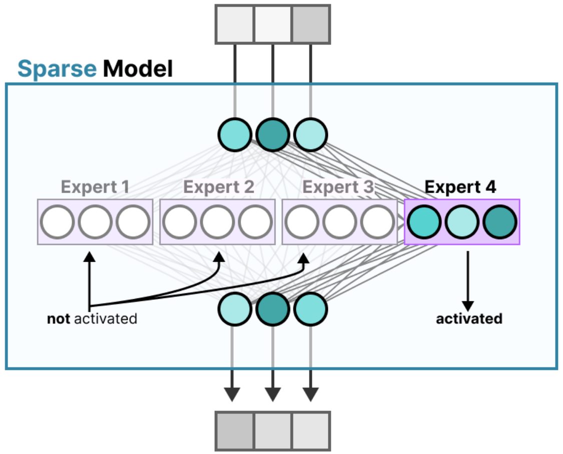
Mixture of Experts (MoE)



credit: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

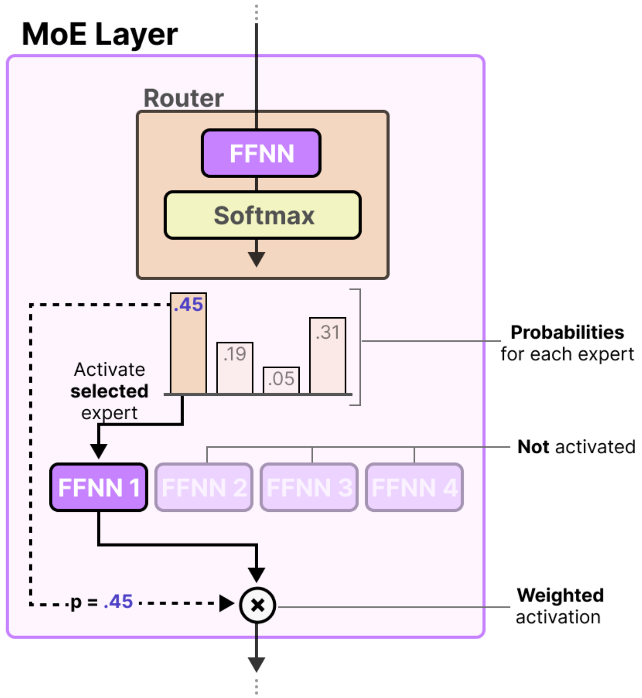
Mixture of Experts (MoE)

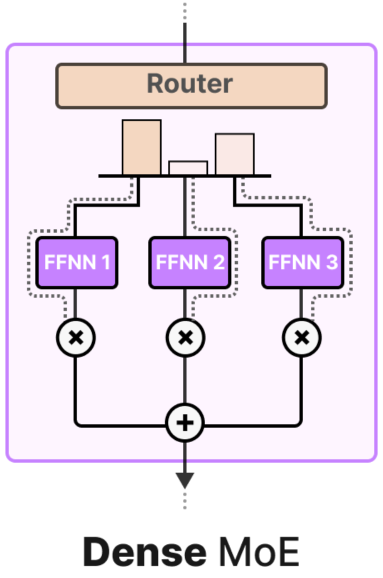


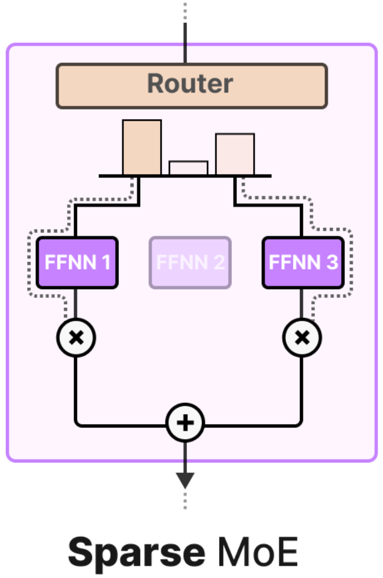


credit: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Experts (MoE)



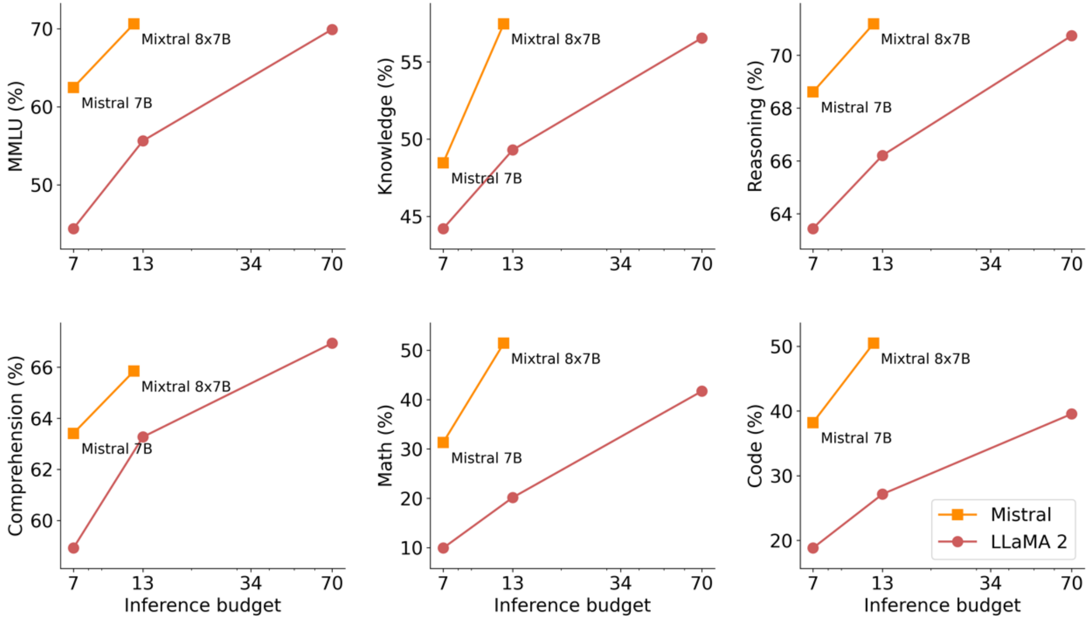




credit: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

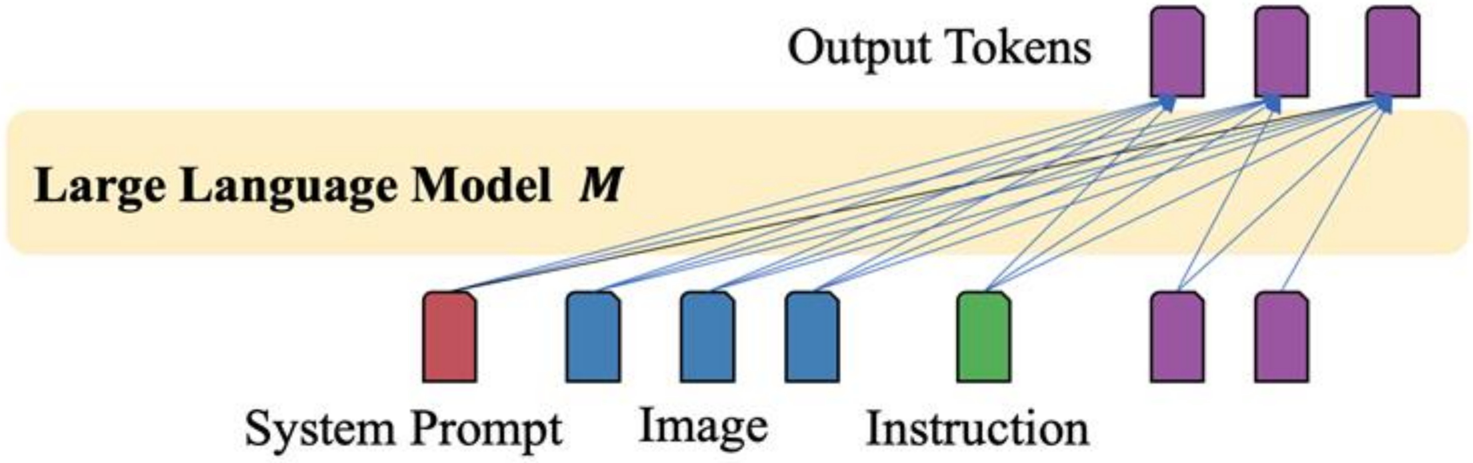
Mixture of Experts (MoE)

Mixtral of experts
A high quality Sparse Mixture-of-Experts.



Mixtral of experts (https://mistral.ai/news/mixtral-of-experts)

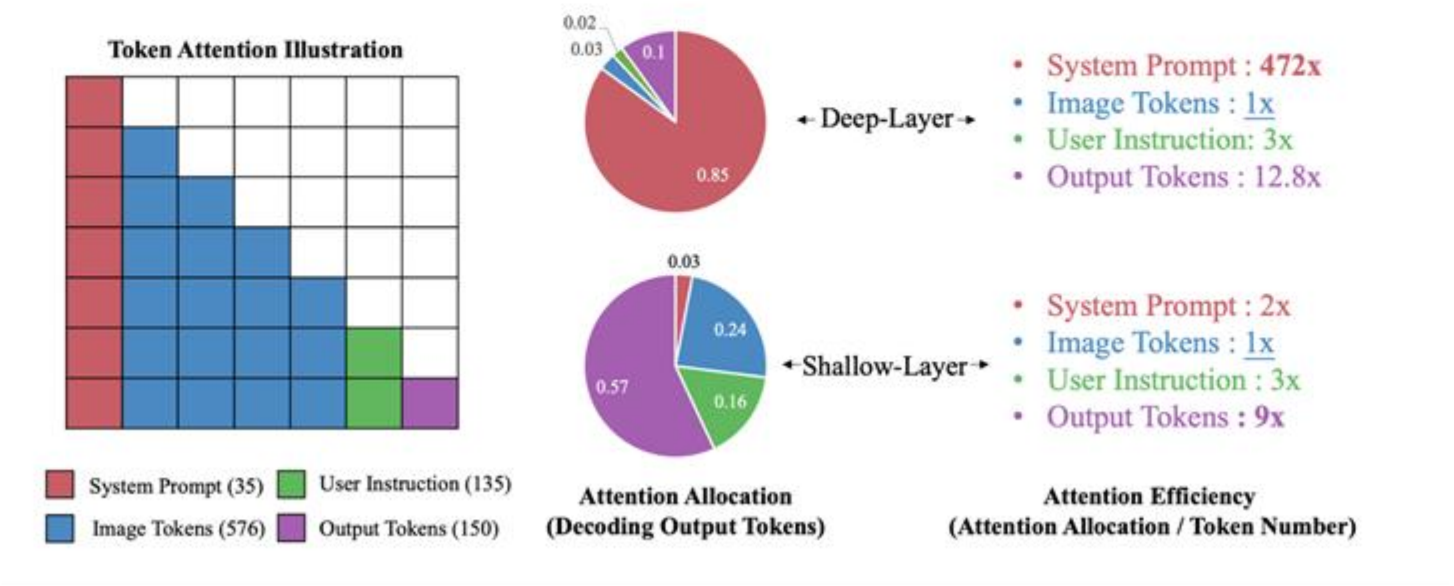
Token Pruning



Typical Architecture for Large VLMs

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

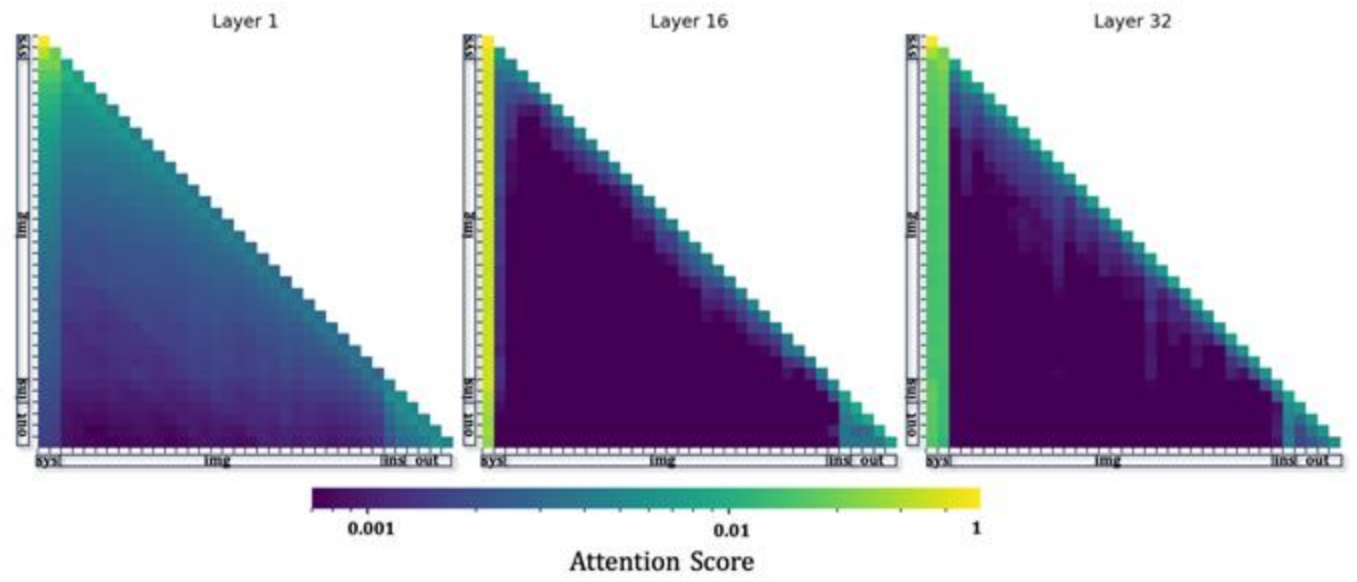
Token Pruning



Inefficient visual attention phenomena

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

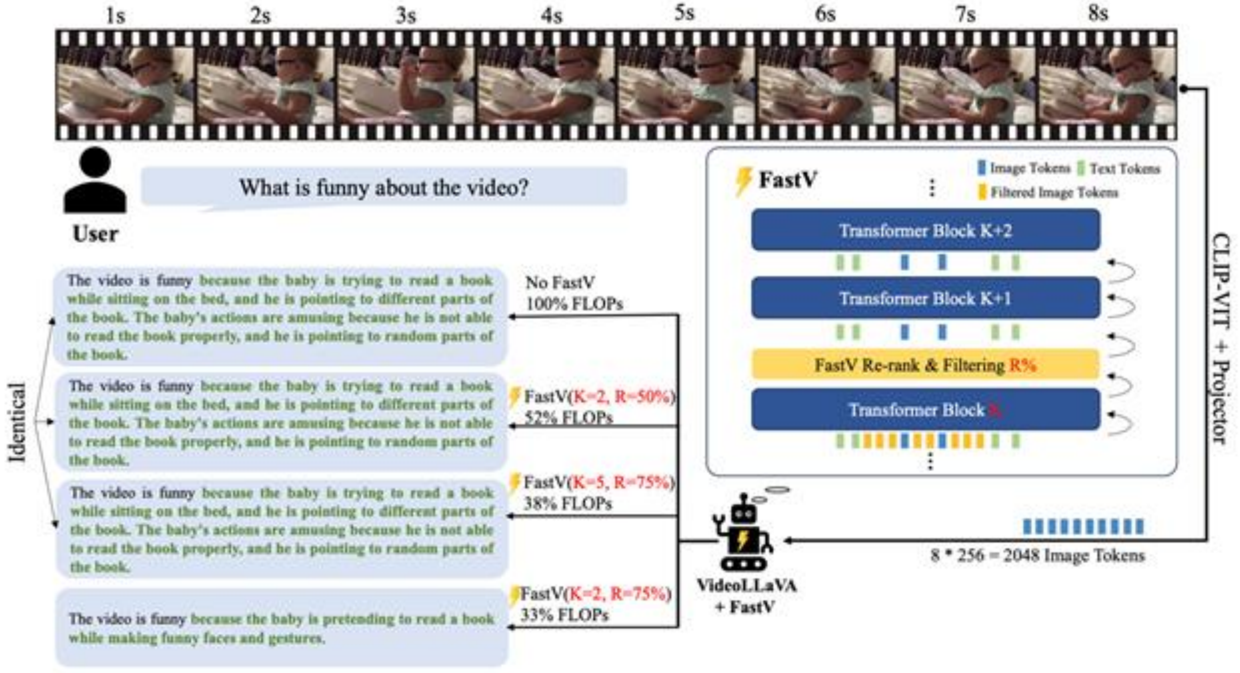
Token Pruning



The attention maps during the decoding process of one model response for LLaVA1.5-7B

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

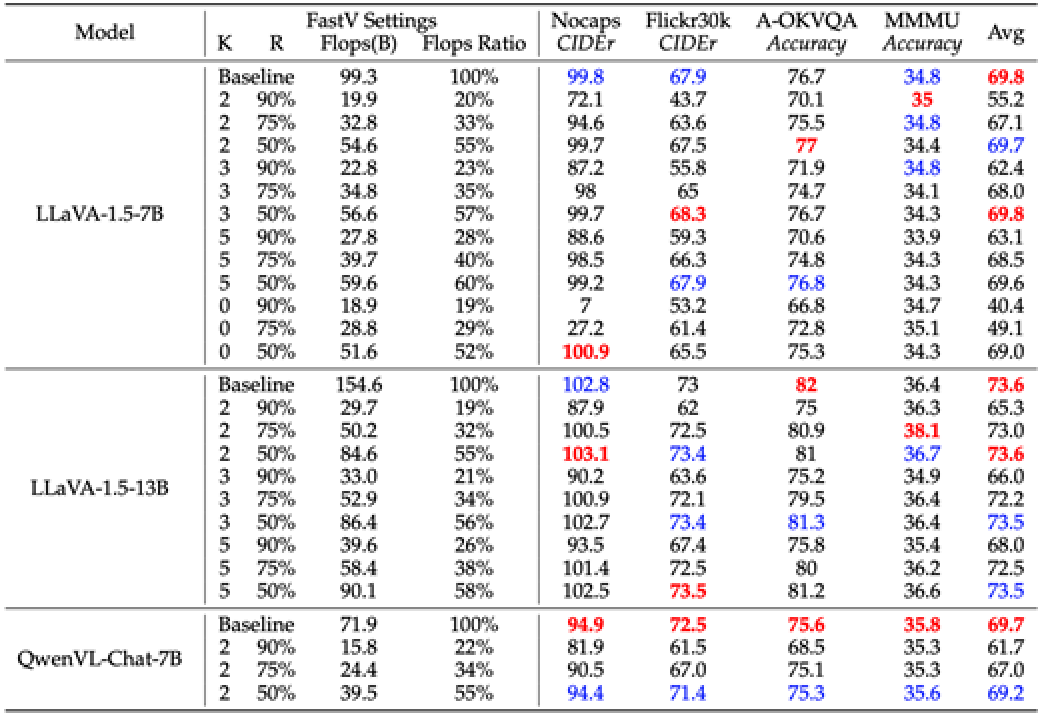
Token Pruning


1s
2s
3s
4s
5s
6s
7s
8s
User
What is funny about the video?
The video is funny because the baby is trying to read a book while sitting on the bed, and he is pointing to different parts of the book. The baby's actions are amusing because he is not able to read the book properly, and he is pointing to random parts of the book.
No FastV
100% FLOPs
FastV(K=2, R=50%)
52% FLOPs
FastV(K=5, R=75%)
38% FLOPs
FastV(K=2, R=75%)
33% FLOPs
The video is funny because the baby is pretending to read a book while making funny faces and gestures.
Identical
FastV
Image Tokens
Text Tokens
Filtered Image Tokens
Transformer Block K+2
Transformer Block K+1
FastV Re-rank & Filtering R%
Transformer Block K
CLIP-VIT + Projector
8 * 256 = 2048 Image Tokens
VideoLLaVA + FastV

FastV dynamically prunes R% image tokens after layer K in the forward process of input tokens

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

Token Pruning

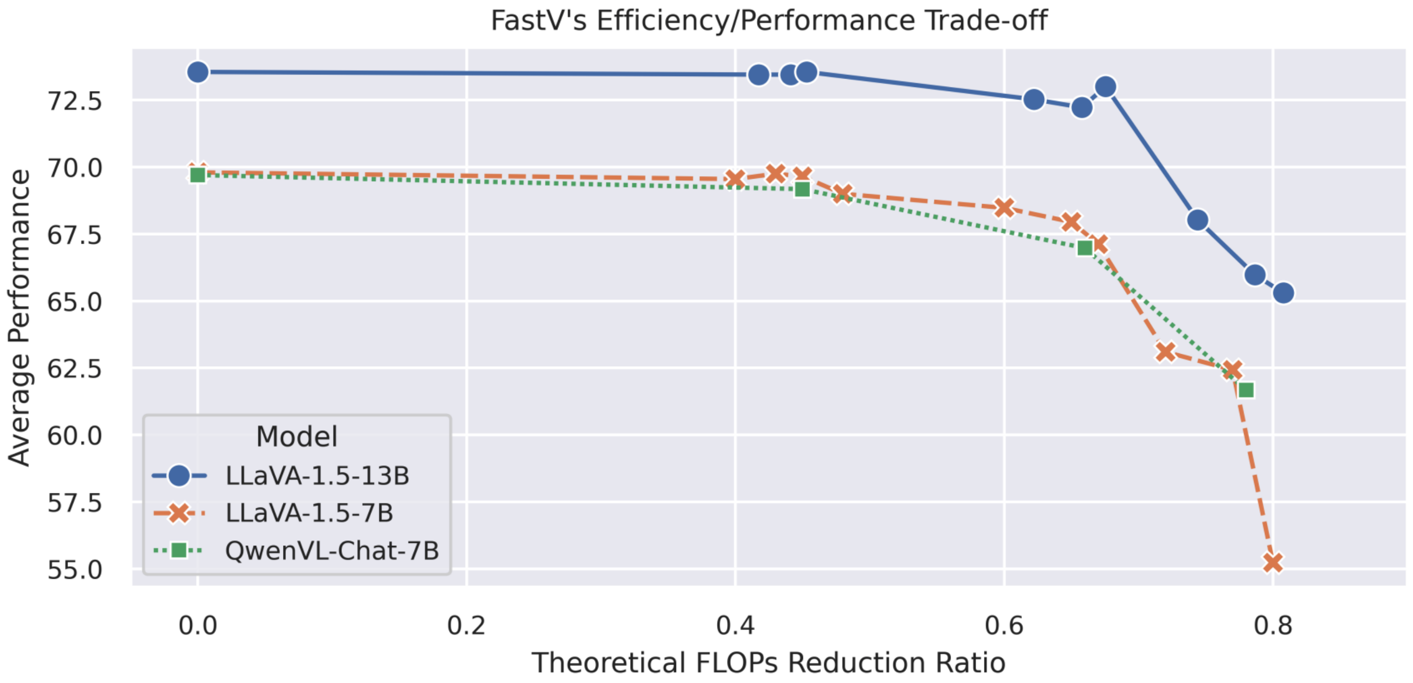


Model	K	R	Flops(B)	Flops Ratio	Nocaps CIDEr	Flickr30k CIDEr	A-OKVQA Accuracy	MMMU Accuracy	Avg
	FastV Settings								
LLaVA-1.5-7B	Baseline		99.3	100%	99.8	67.9	76.7	34.8	69.8
	2	90%	19.9	20%	72.1	43.7	70.1	35	55.2
	2	75%	32.8	33%	94.6	63.6	75.5	34.8	67.1
	2	50%	54.6	55%	99.7	67.5	77	34.4	69.7
	3	90%	22.8	23%	87.2	55.8	71.9	34.8	62.4
	3	75%	34.8	35%	98	65	74.7	34.1	68.0
	3	50%	56.6	57%	99.7	68.3	76.7	34.3	69.8
	5	90%	27.8	28%	88.6	59.3	70.6	33.9	63.1
	5	75%	39.7	40%	98.5	66.3	74.8	34.3	68.5
	5	50%	59.6	60%	99.2	67.9	76.8	34.3	69.6
	0	90%	18.9	19%	7	53.2	66.8	34.7	40.4
	0	75%	28.8	29%	27.2	61.4	72.8	35.1	49.1
	0	50%	51.6	52%	100.9	65.5	75.3	34.3	69.0
LLaVA-1.5-13B	Baseline		154.6	100%	102.8	73	82	36.4	73.6
	2	90%	29.7	19%	87.9	62	75	36.3	65.3
	2	75%	50.2	32%	100.5	72.5	80.9	38.1	73.0
	2	50%	84.6	55%	103.1	73.4	81	36.7	73.6
	3	90%	33.0	21%	90.2	63.6	75.2	34.9	66.0
	3	75%	52.9	34%	100.9	72.1	79.5	36.4	72.2
	3	50%	86.4	56%	102.7	73.4	81.3	36.4	73.5
	5	90%	39.6	26%	93.5	67.4	75.8	35.4	68.0
	5	75%	58.4	38%	101.4	72.5	80	36.2	72.5
	5	50%	90.1	58%	102.5	73.5	81.2	36.6	73.5
QwenVL-Chat-7B	Baseline		71.9	100%	94.9	72.5	75.6	35.8	69.7
	2	90%	15.8	22%	81.9	61.5	68.5	35.3	61.7
	2	75%	24.4	34%	90.5	67.0	75.1	35.3	67.0
	2	50%	39.5	55%	94.4	71.4	75.3	35.6	69.2

FastV could achieve about 45% FLOPs reduction for different LVLMs without sacrificing the performance

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

Token Pruning



FastV could achieve about 45% FLOPs reduction for different LVLMs without sacrificing the performance

An Image is Worth 1/2 Tokens After Layer 2 (Chen et al. 2024)

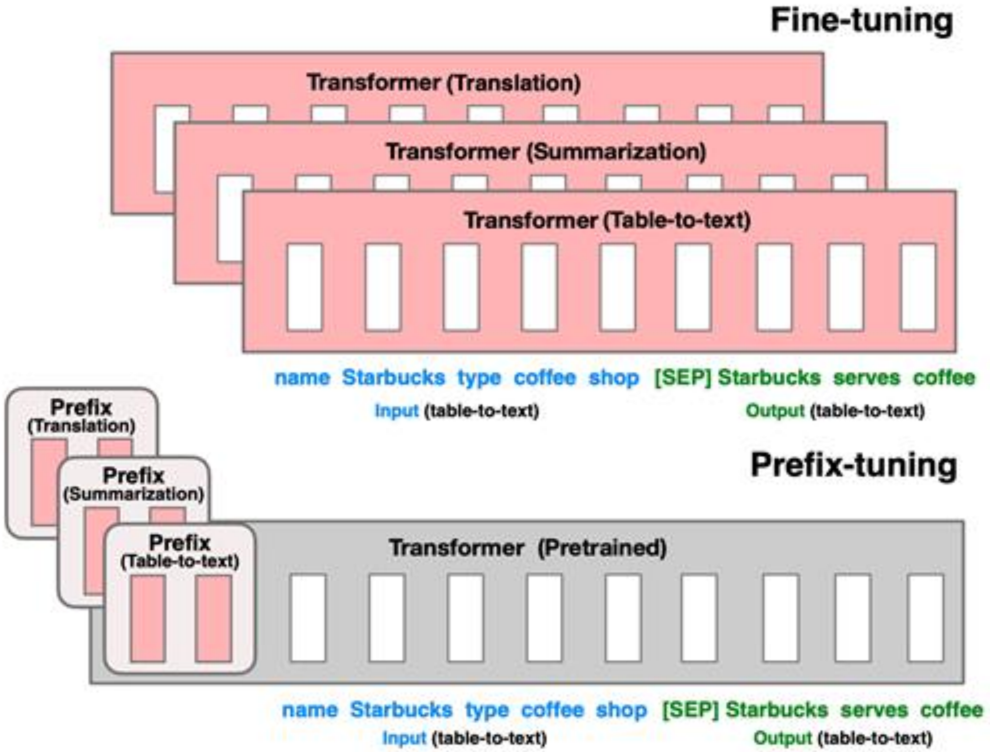
Parameter-Efficient Fine-Tuning

- Full fine-tuning require more computational cost and becomes infeasible to train on consumer hardware.
- Full fine-tuning leads to catastrophic forgetting in the low-data regimes.
- Storing and deploying fine-tuned models independently for each downstream task becomes very expensive

Parameter-Efficient Fine-tuning (PEFT) approaches are meant to address both problems!

- Prefix Tuning / Prompt Tuning
- Adapter Tuning
- LoRA Tuning

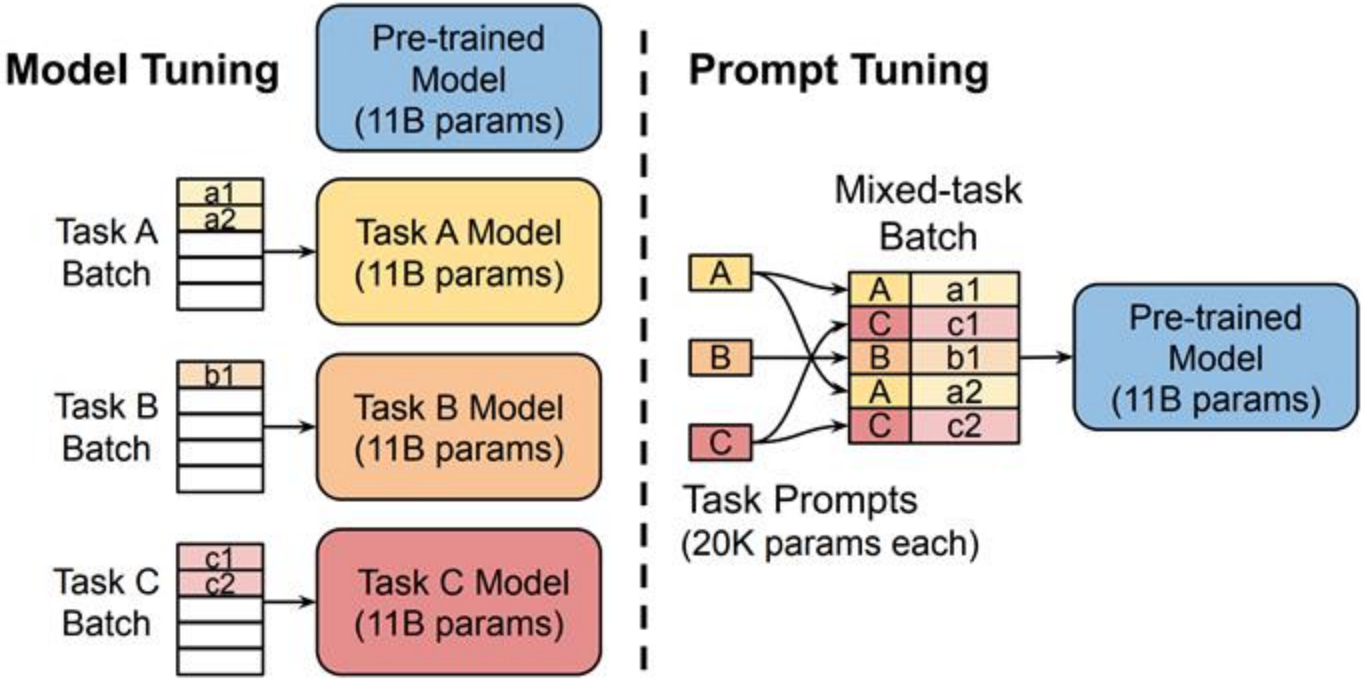
Prefix Tuning


Fine-tuning
Transformer (Translation)
Transformer (Summarization)
Transformer (Table-to-text)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Prefix-tuning
Prefix (Translation)
Prefix (Summarization)
Prefix (Table-to-text)
Transformer (Pretrained)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)

keep language model parameters frozen, but optimizes a small set of continuous task-specific vectors

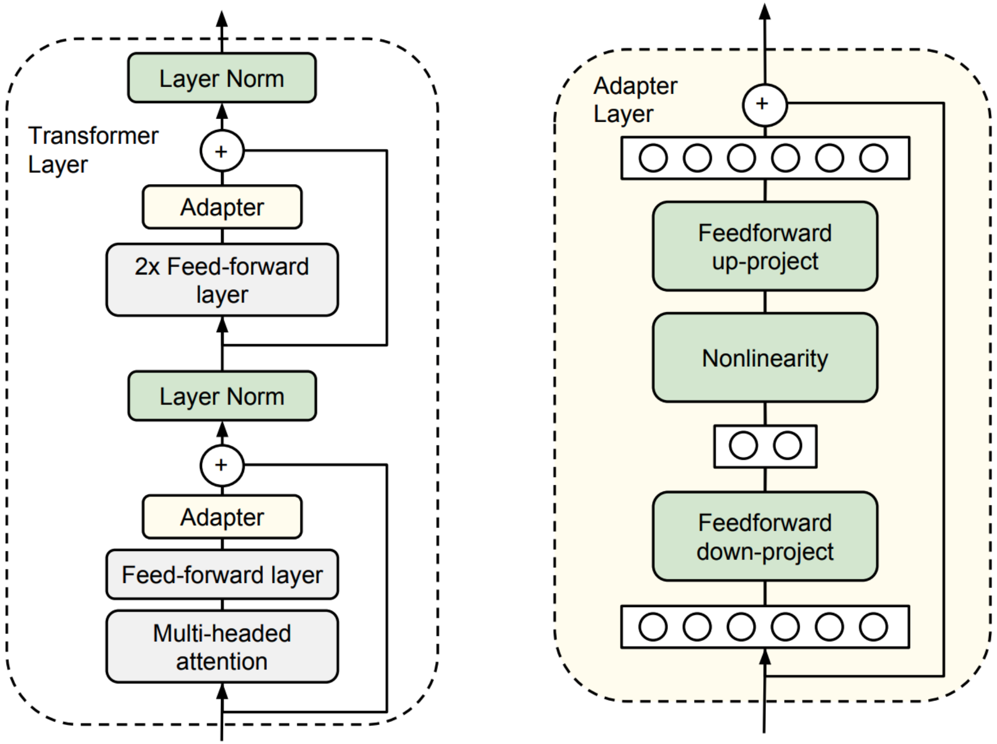
Prefix-Tuning (Li et al. 2021)

Prompt Tuning


Model Tuning
Pre-trained Model (11B params)
Task A Batch
a1
a2
Task A Model (11B params)
Task B Batch
b1
Task B Model (11B params)
Task C Batch
c1
c2
Task C Model (11B params)
Prompt Tuning
Mixed-task Batch
A
B
C
A a1
C c1
B b1
A a2
C c2
Pre-trained Model (11B params)
Task Prompts (20K params each)

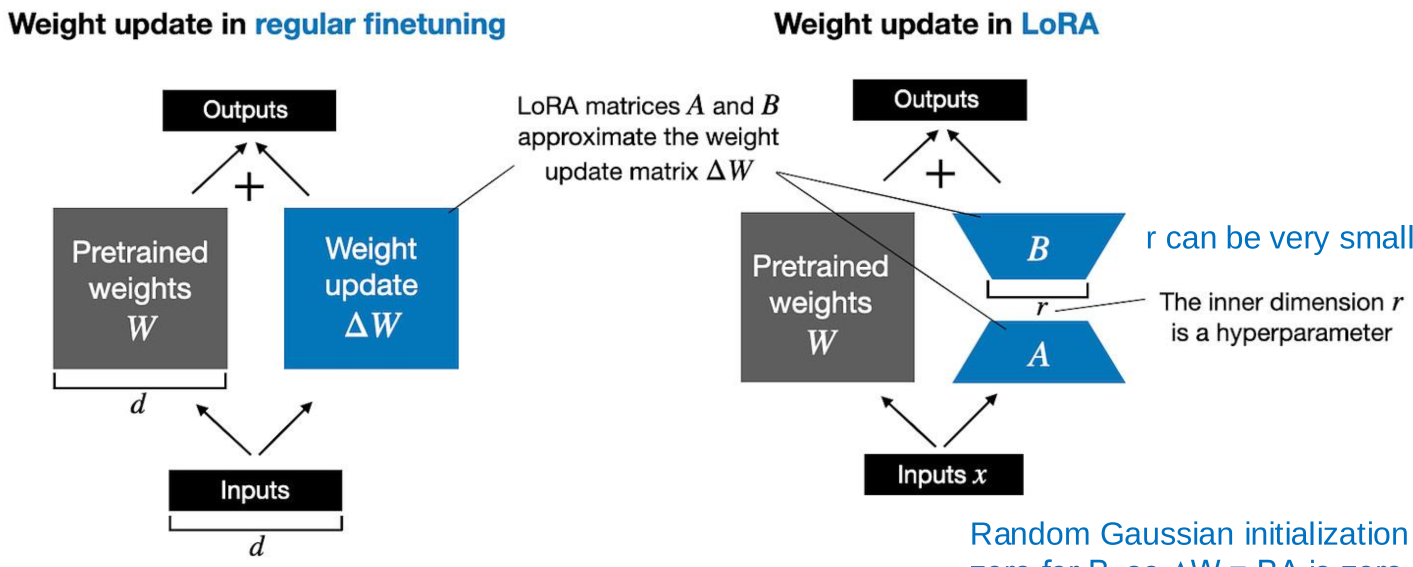
Prompt Tuning (Lester et al. 2021)

Adapter Tuning



Parameter-Efficient Transfer Learning for NLP (Houlsby et al. 2019)

LoRA: Low-Rank Adaptation


Weight update in regular finetuning
Outputs
+
Pretrained weights W
Weight update ΔW
d
Inputs
d
LoRA matrices A and B approximate the weight update matrix ΔW
Weight update in LoRA
Outputs
+
Pretrained weights W
B
r
A
The inner dimension r is a hyperparameter
Inputs x

r can be very small, like 4/8

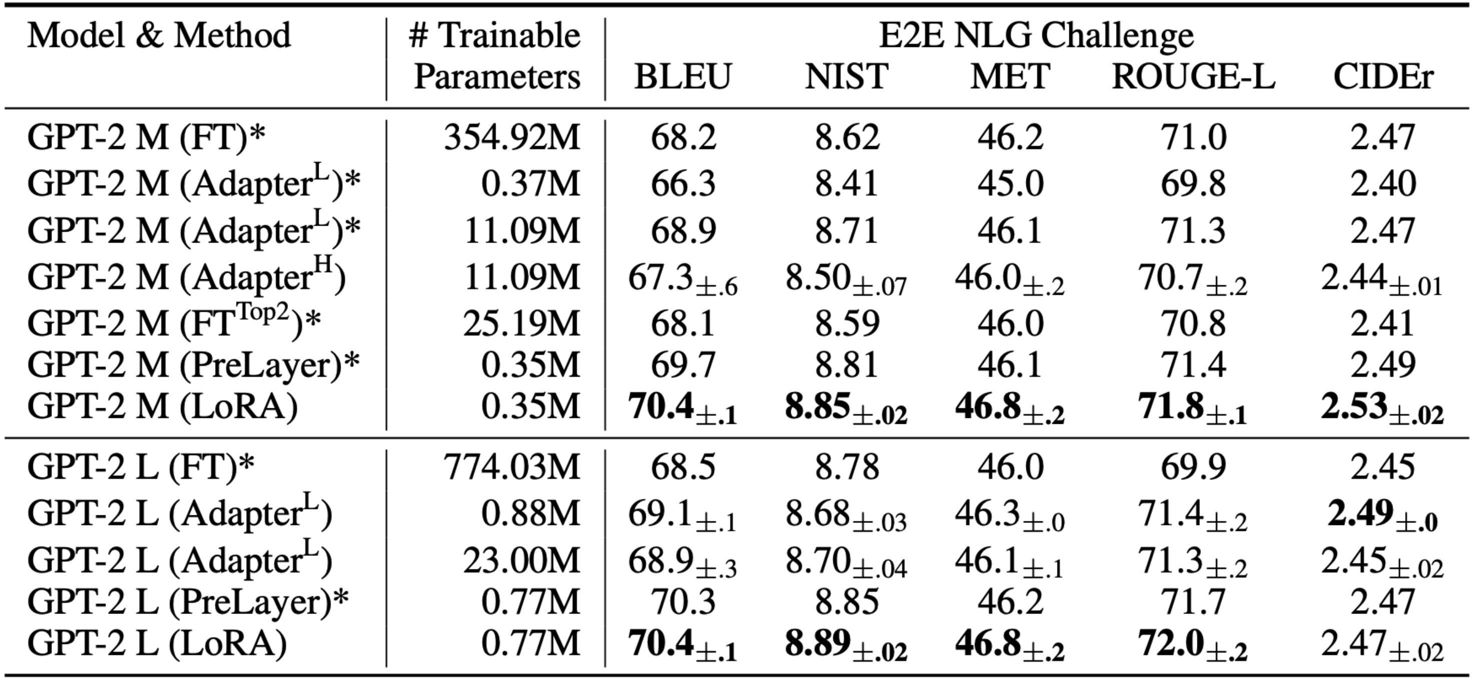
Random Gaussian initialization for A and zero for B, so ∆W = BA is zero at the beginning of training

$$h = W_0x + \Delta W x = W_0x + BAx$$

LoRA (Hu et al. 2021)

https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms

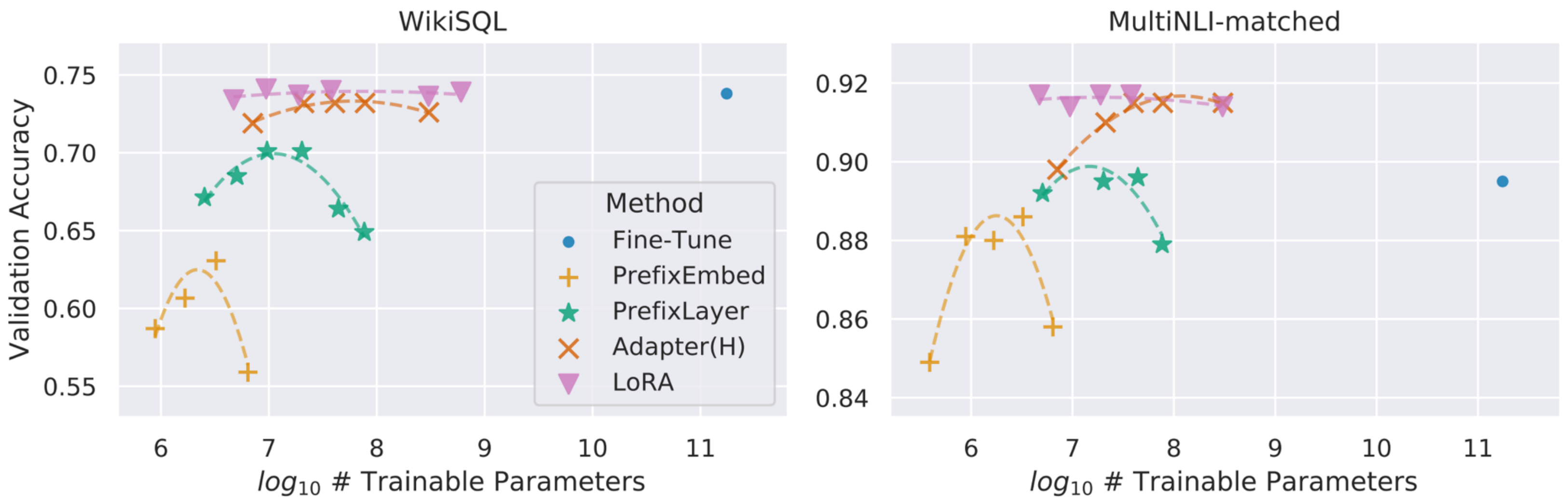
LoRA: Low-Rank Adaptation



Model & Method	# Trainable Parameters	E2E NLG Challenge				
		BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter^{L})*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M (Adapter^{L})*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (Adapter^{H})	11.09M	67.3$_{\pm.6}$	8.50$_{\pm.07}$	46.0$_{\pm.2}$	70.7$_{\pm.2}$	2.44$_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	**70.4**$_{\pm.1}$	**8.85**$_{\pm.02}$	**46.8**$_{\pm.2}$	**71.8**$_{\pm.1}$	**2.53**$_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adapter^{L})	0.88M	69.1$_{\pm.1}$	8.68$_{\pm.03}$	46.3$_{\pm.0}$	71.4$_{\pm.2}$	**2.49**$_{\pm.0}$
GPT-2 L (Adapter^{L})	23.00M	68.9$_{\pm.3}$	8.70$_{\pm.04}$	46.1$_{\pm.1}$	71.3$_{\pm.2}$	2.45$_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	**70.4**$_{\pm.1}$	**8.89**$_{\pm.02}$	**46.8**$_{\pm.2}$	**72.0**$_{\pm.2}$	2.47$_{\pm.02}$

LoRA (Hu et al. 2021)

LoRA: Low-Rank Adaptation

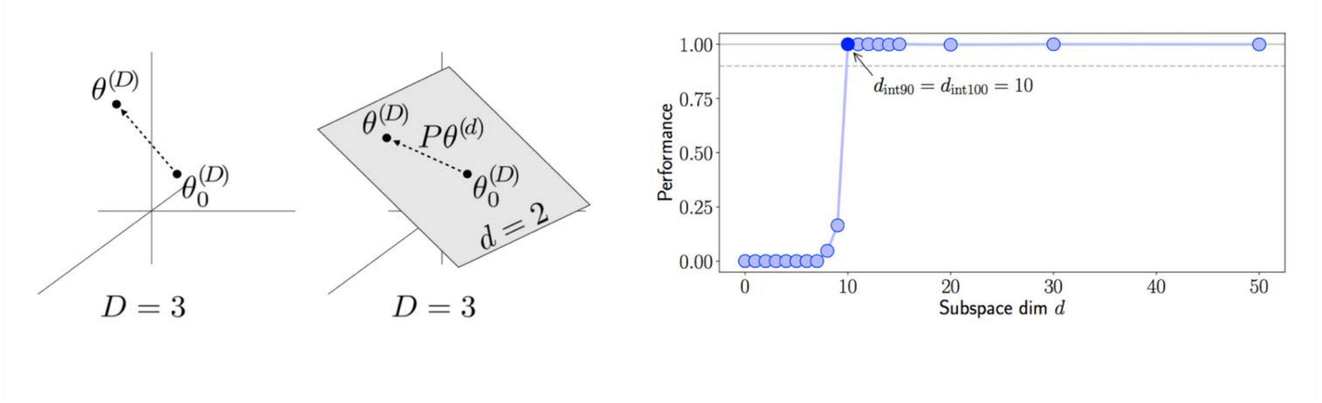


GPT-3 175B validation accuracy vs. number of trainable parameters of several adaptation methods on WikiSQL and MNLI-matched

LoRA (Hu et al. 2021)

Why LoRA Works?

- When adapting to a specific task, Aghajanyan et al. (2020) shows that the pre-trained language models have a low “intrinsic dimension”.



train models not in their native parameter space (i.e. the parameter space consisting of all the parameters) but in smaller randomly oriented subspaces

LoRA (Hu et al. 2021)

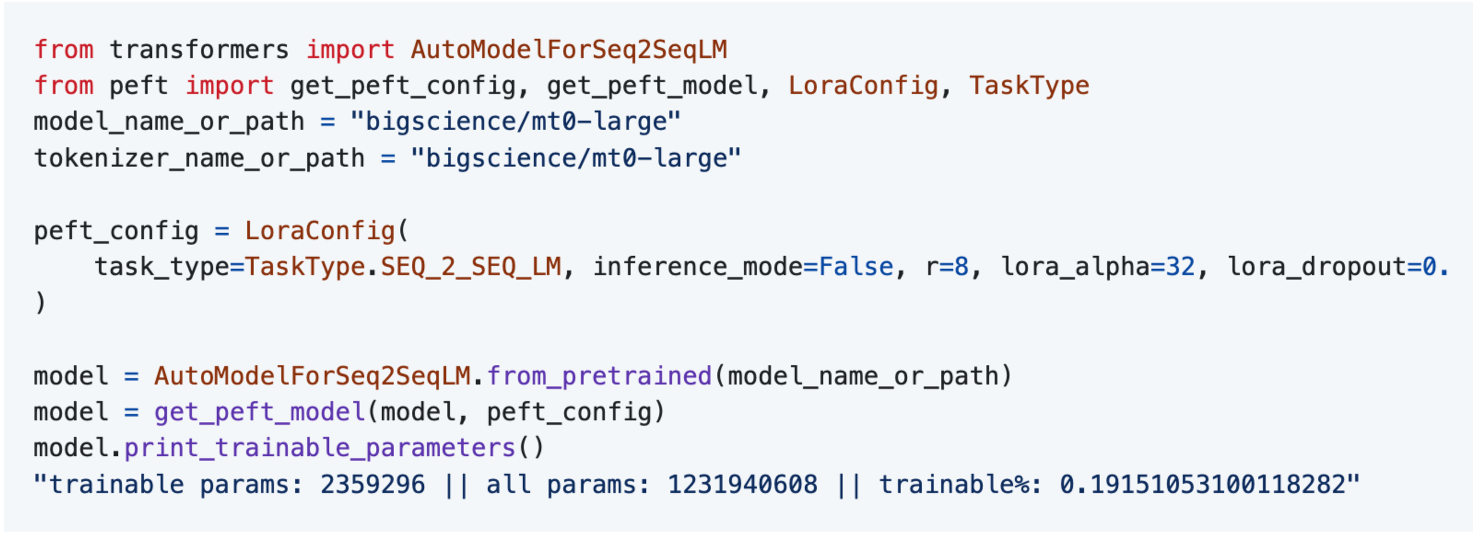
Parameter-Efficient Fine-Tuning

- Prefix Tuning / Prompt Tuning
- Adapter Tuning
- LoRA Tuning

- In general, LoRA has better performance
- LoRA has No Additional Inference Latency: $W = W_0 + BA$
- Prefix/Prompt Tuning is good at dynamical batch inference

Examples for LoRA

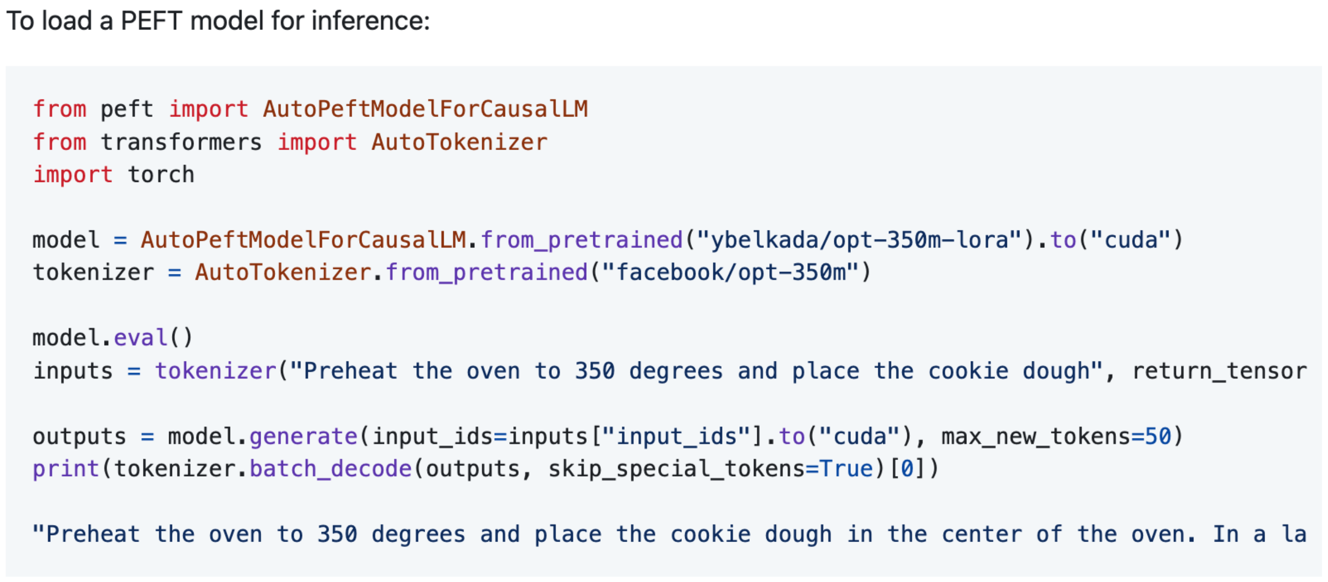
Huggingface 🤗 PEFT Library



```
from transformers import AutoModelForSeq2SeqLM
from peft import get_peft_config, get_peft_model, LoraConfig, TaskType
model_name_or_path = "bigscience/mt0-large"
tokenizer_name_or_path = "bigscience/mt0-large"

peft_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM, inference_mode=False, r=8, lora_alpha=32, lora_dropout=0.
)

model = AutoModelForSeq2SeqLM.from_pretrained(model_name_or_path)
model = get_peft_model(model, peft_config)
model.print_trainable_parameters()
"trainable params: 2359296 || all params: 1231940608 || trainable%: 0.19151053100118282"
```

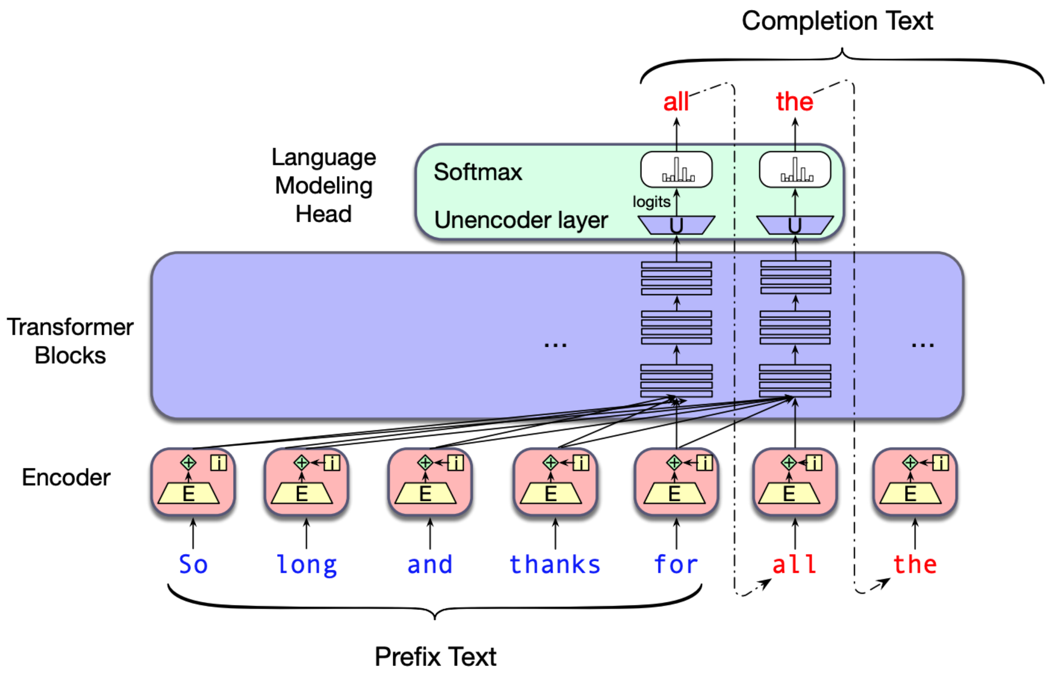
Examples for LoRA

Huggingface 🤗 PEFT Library


To load a PEFT model for inference:

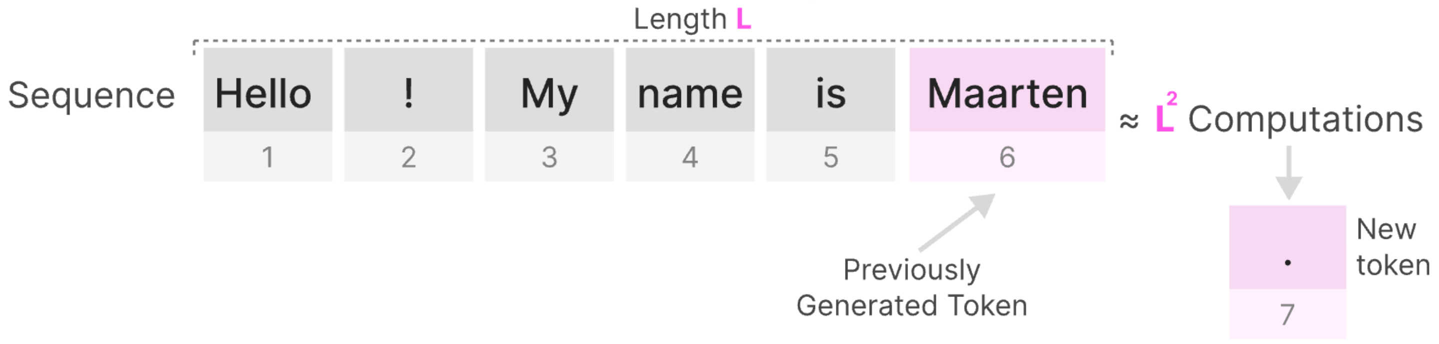
```
from peft import AutoPeftModelForCausalLM
from transformers import AutoTokenizer
import torch

model = AutoPeftModelForCausalLM.from_pretrained("ybelkada/opt-350m-lora").to("cuda")
tokenizer = AutoTokenizer.from_pretrained("facebook/opt-350m")

model.eval()
inputs = tokenizer("Preheat the oven to 350 degrees and place the cookie dough", return_tensor

outputs = model.generate(input_ids=inputs["input_ids"].to("cuda"), max_new_tokens=50)
print(tokenizer.batch_decode(outputs, skip_special_tokens=True)[0])

"Preheat the oven to 350 degrees and place the cookie dough in the center of the oven. In a la
```

New Architecture Design



LARGE LANGUAGE MODELS WITH TRANSFORMERS (Daniel Jurafsky & James H. Martin 2024)

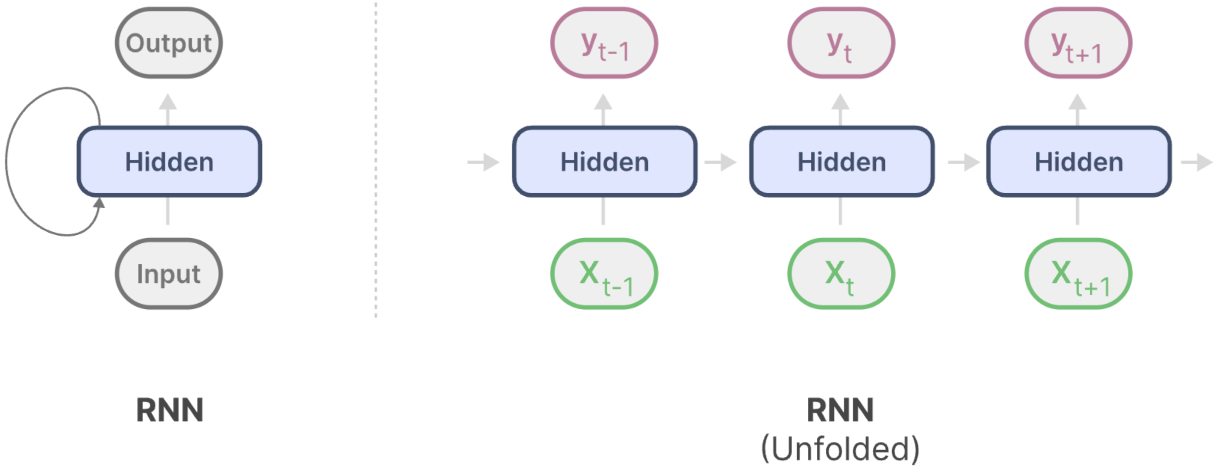
New Architecture Design



	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)

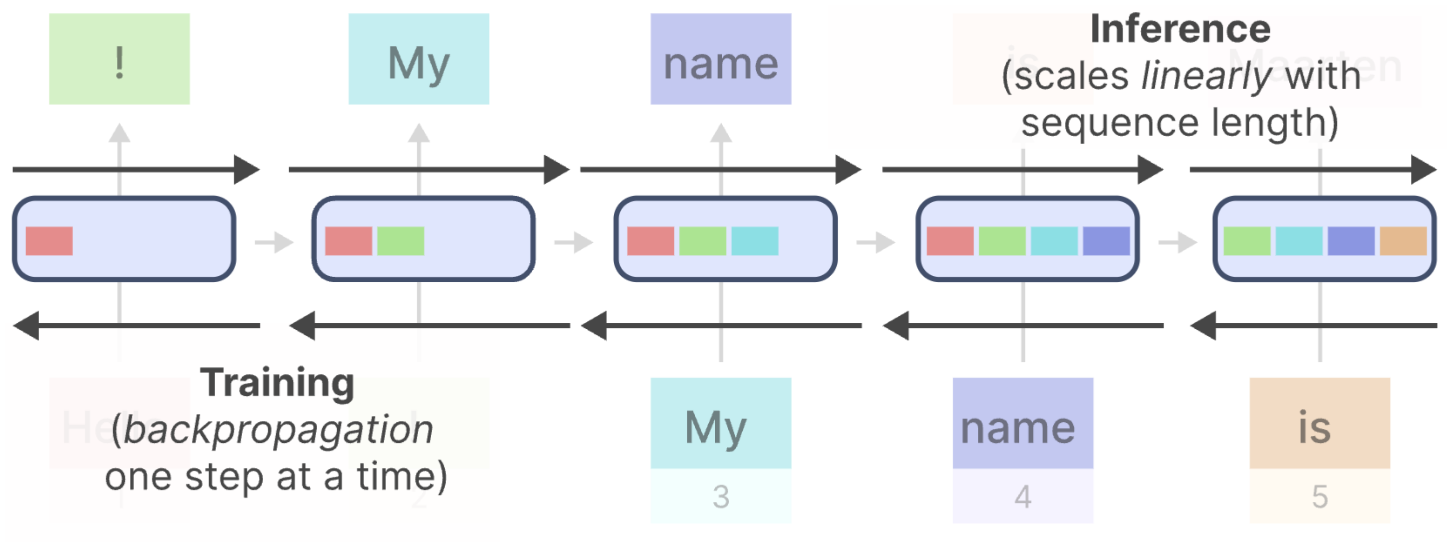
https://www.maartengrootendorst.com/blog/mamba/

New Architecture Design



https://www.maartengrootendorst.com/blog/mamba/

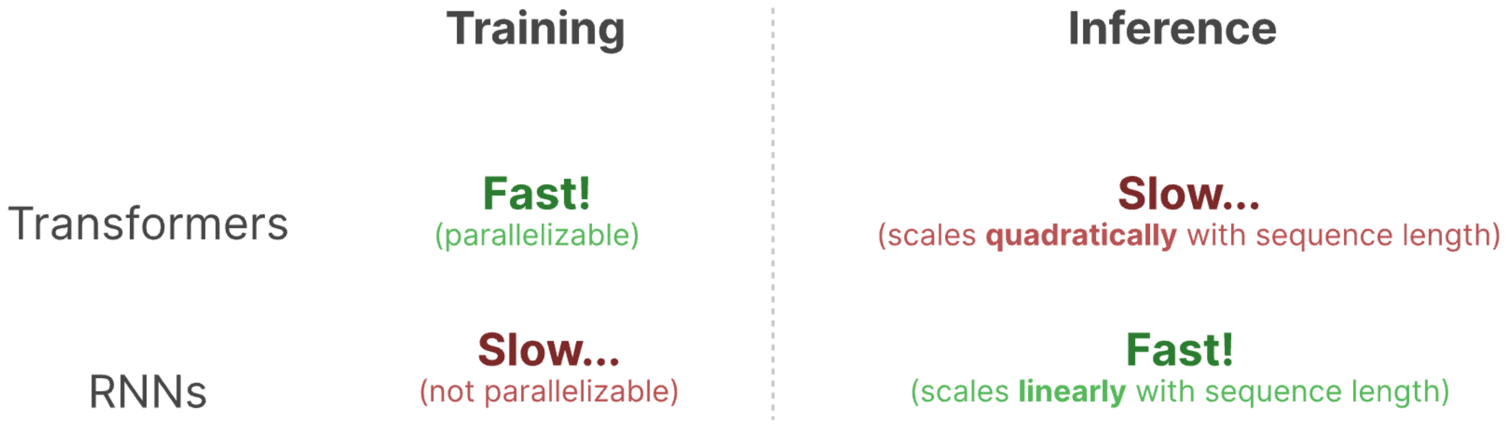
New Architecture Design



!
My
name
Inference
(scales *linearly* with sequence length)
Training
(*backpropagation* one step at a time)
My
3
name
4
is
5

https://www.maartengrootendorst.com/blog/mamba/

New Architecture Design



	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)
RNNs	**Slow...** (not parallelizable)	**Fast!** (scales **linearly** with sequence length)

https://www.maartengrootendorst.com/blog/mamba/

State Space Model

$$h_0 = \bar{\mathbf{B}}x_0$$
$$h_1 = \bar{\mathbf{A}}(\bar{\mathbf{B}}x_0) + \bar{\mathbf{B}}x_1$$
$$h_2 = \bar{\mathbf{A}}(\bar{\mathbf{A}}(\bar{\mathbf{B}}x_0) + \bar{\mathbf{B}}x_1) + \bar{\mathbf{B}}x_2$$
$$h_3 = \bar{\mathbf{A}}(\bar{\mathbf{A}}(\bar{\mathbf{A}}(\bar{\mathbf{B}}x_0) + \bar{\mathbf{B}}x_1) + \bar{\mathbf{B}}x_2) + \bar{\mathbf{B}}x_3$$

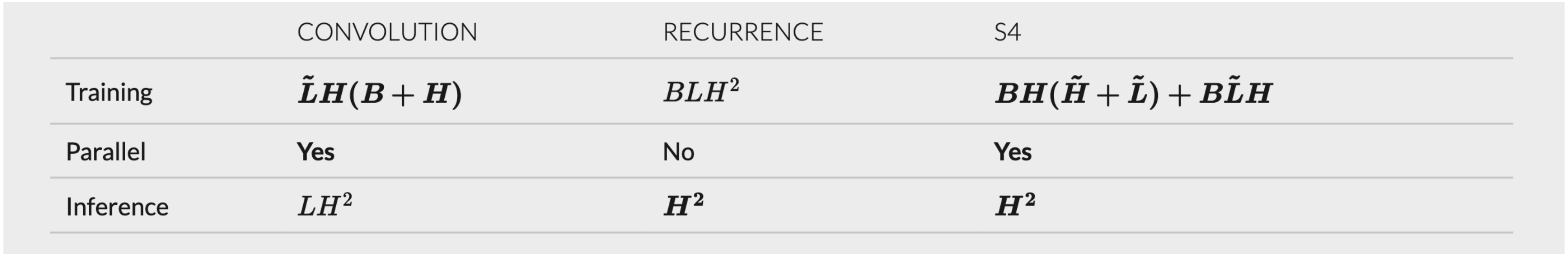
$$y_3 = \mathbf{C}(\bar{\mathbf{A}}(\bar{\mathbf{A}}(\bar{\mathbf{A}}(\bar{\mathbf{B}}x_0) + \bar{\mathbf{B}}x_1) + \bar{\mathbf{B}}x_2) + \bar{\mathbf{B}}x_3)$$
$$y_3 = \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{B}}x_0 + \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{B}}x_1 + \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}x_2 + \mathbf{C}\bar{\mathbf{B}}x_3$$

$$\bar{\mathbf{K}} = \begin{pmatrix}\mathbf{C}\bar{\mathbf{B}} & \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}} & \cdots & \mathbf{C}\bar{\mathbf{A}}^{\mathbf{k}}\bar{\mathbf{B}}\end{pmatrix}$$
$$y = \bar{\mathbf{K}} * x$$

$$y_3 = \begin{pmatrix}\mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{B}} & \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{A}}\bar{\mathbf{B}} & \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}} & \mathbf{C}\bar{\mathbf{B}}\end{pmatrix}\begin{pmatrix}x_0 \\ x_1 \\ x_2 \\ x_3\end{pmatrix}$$

https://jackcook.com/2024/02/23/mamba.html

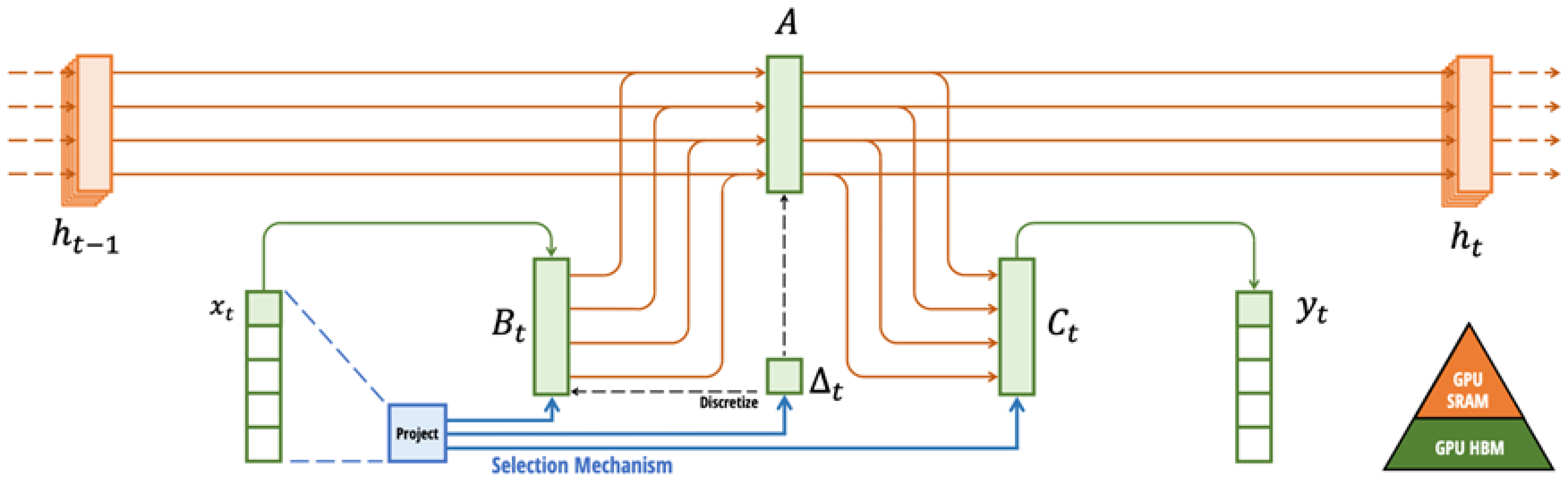
State Space Model



	CONVOLUTION	RECURRENCE	S4
Training	$\boldsymbol{\tilde{L}H(B+H)}$	BLH^2	$\boldsymbol{BH(\tilde{H}+\tilde{L})+B\tilde{L}H}$
Parallel	**Yes**	No	**Yes**
Inference	LH^2	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$

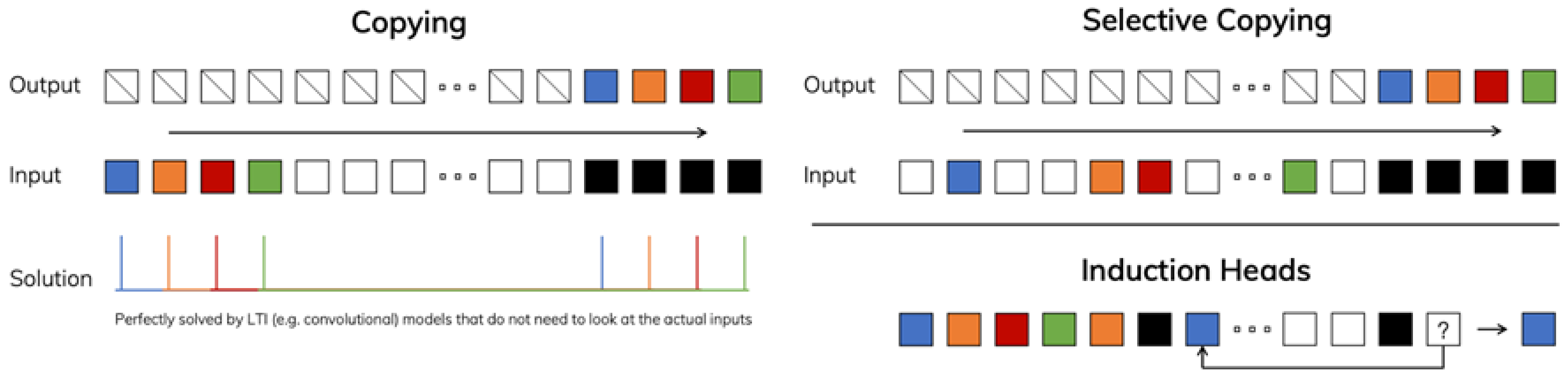
https://jackcook.com/2024/02/23/mamba.html

Selective State Space Model



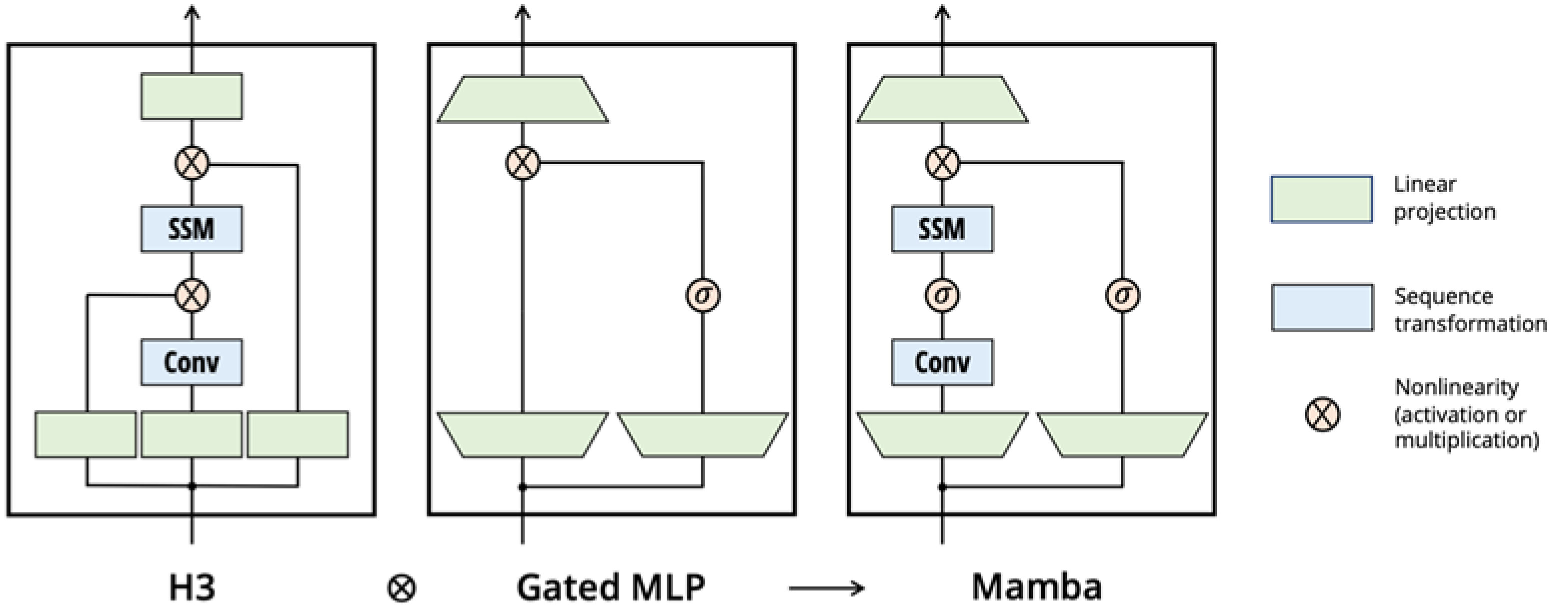
Mamba (Gu et al. 2023)

Selective State Space Model


Copying
Output
Input
Solution
Perfectly solved by LTI (e.g. convolutional) models that do not need to look at the actual inputs
Selective Copying
Output
Input
Induction Heads
?

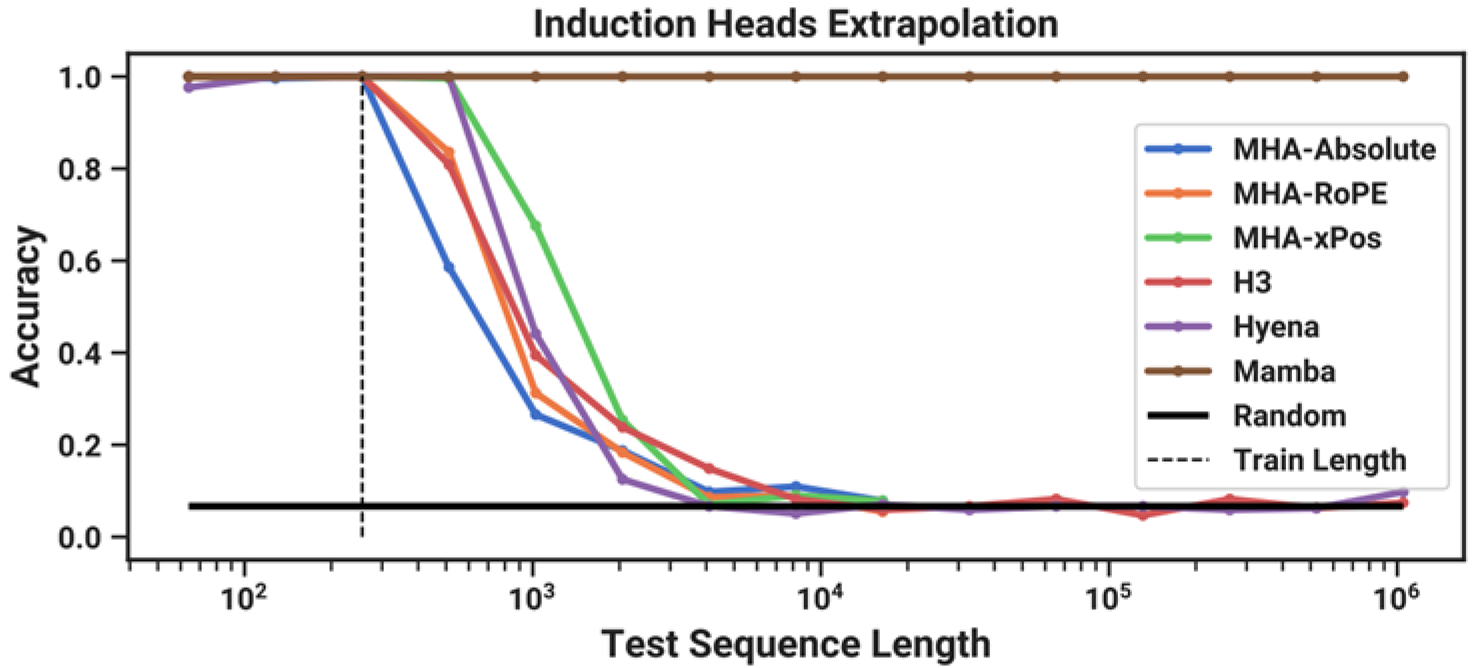
Mamba (Gu et al. 2023)

Selective State Space Model



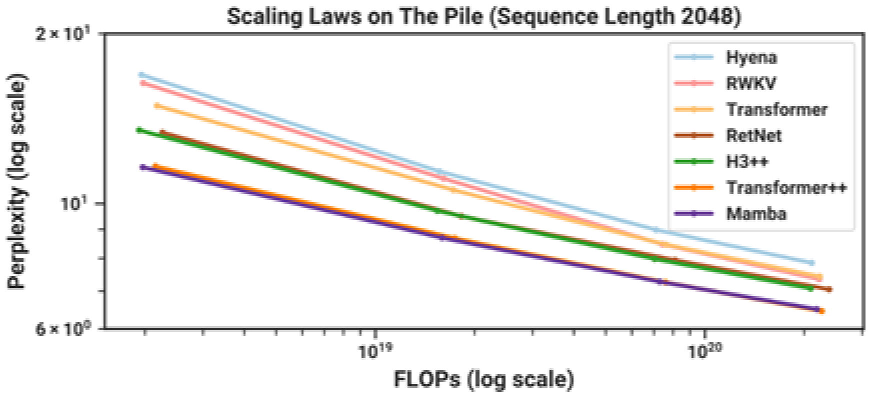
Mamba (Gu et al. 2023)

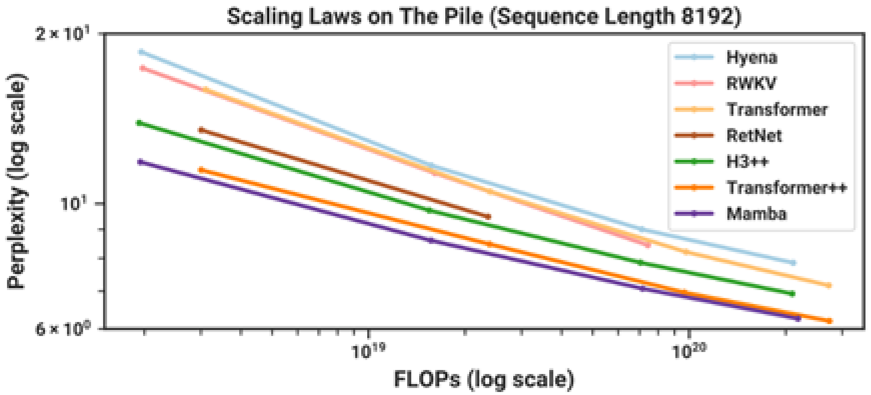
Selective State Space Model



Mamba (Gu et al. 2023)

Selective State Space Model





Mamba (Gu et al. 2023)

Next time

- Guest Lecture: Responsible and Ethical Deployment of Advanced Vision Models