Recent LLM basics

Presented by

Amir Shariatmadari, Guangzhi Xiong, Sabit Ahmed, Shiyu Feng

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Overview

- (LLM Training) Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling
- (LLM Inference) Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems
- (Large Audio Model) Sparks of Large Audio Models: A Survey and Outlook
- (Multimodal LLM) MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training

Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling

EleutherAI

Presenter: Guangzhi Xiong

University of Virginia | ENGINEERING Department of Computer Science

Motivation

Research Questions:

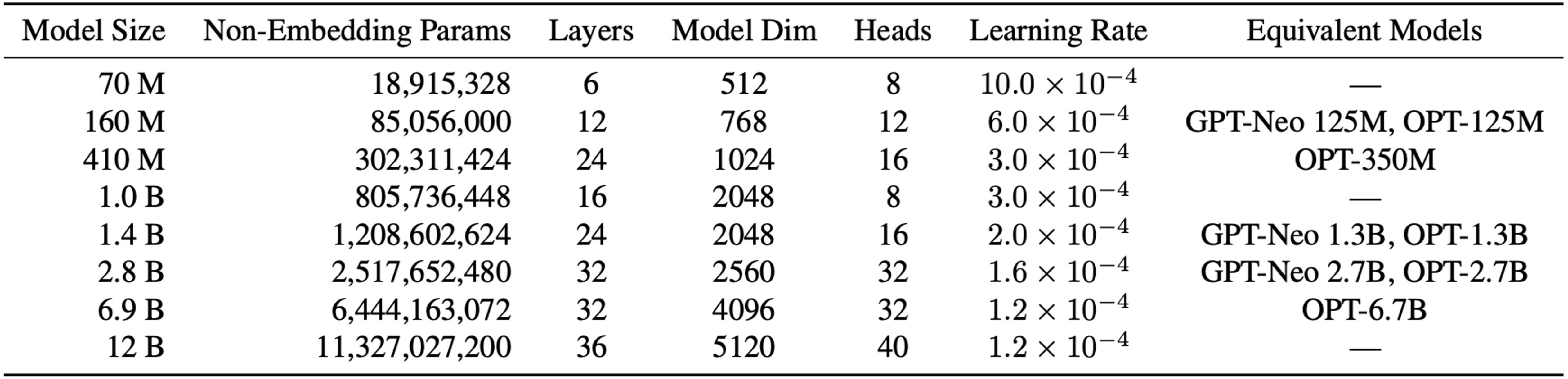
- How do large language models (LLMs) develop and evolve over the course of training?
- How do these patterns change as models scale?

Contributions:

- (Models) It introduces of Pythia, a suite of **16 LLMs** all trained on public data seen in the exact same order and ranging in size from **70M to 12B** parameters.
- (Checkpoints) It provides public access to 154 **checkpoints** for each one of the 16 models.
- (Data) It provides tools to download and reconstruct their exact training **dataloaders** for further study.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Models in the Pythia suite

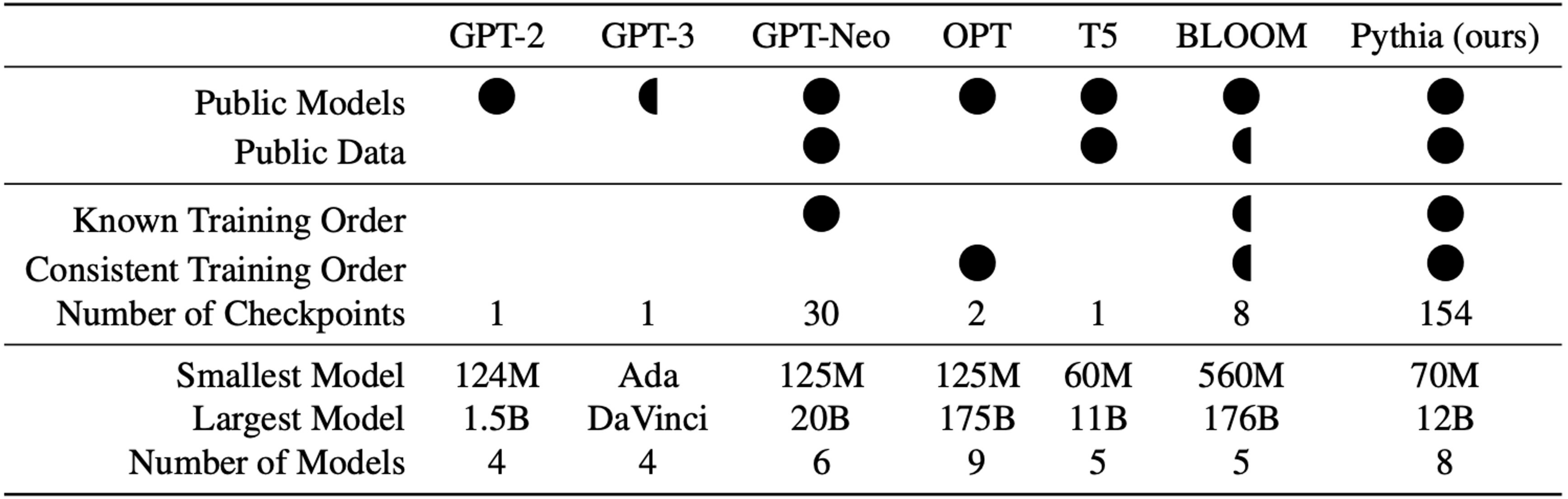


Model Size	Non-Embedding Params	Layers	Model Dim	Heads	Learning Rate	Equivalent Models
70 M	18,915,328	6	512	8	10.0×10^{-4}	—
160 M	85,056,000	12	768	12	6.0×10^{-4}	GPT-Neo 125M, OPT-125M
410 M	302,311,424	24	1024	16	3.0×10^{-4}	OPT-350M
1.0 B	805,736,448	16	2048	8	3.0×10^{-4}	—
1.4 B	1,208,602,624	24	2048	16	2.0×10^{-4}	GPT-Neo 1.3B, OPT-1.3B
2.8 B	2,517,652,480	32	2560	32	1.6×10^{-4}	GPT-Neo 2.7B, OPT-2.7B
6.9 B	6,444,163,072	32	4096	32	1.2×10^{-4}	OPT-6.7B
12 B	11,327,027,200	36	5120	40	1.2×10^{-4}	—

*Models marked as "equivalent" have the same architecture and number of non-embedding parameters.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Models in the Pythia suite



	GPT-2	GPT-3	GPT-Neo	OPT	T5	BLOOM	Pythia (ours)
Public Models	●	◖	●	●	●	●	●
Public Data			●		●	◖	●
Known Training Order			●			◖	●
Consistent Training Order				●		◖	●
Number of Checkpoints	1	1	30	2	1	8	154
Smallest Model	124M	Ada	125M	125M	60M	560M	70M
Largest Model	1.5B	DaVinci	20B	175B	11B	176B	12B
Number of Models	4	4	6	9	5	5	8

Table 2. Commonly used model suites and how they rate according to our requirements. Further information can be found in Appendix F.1.

Training data in Pythia

The Pile

- Description: A curated collection of English language datasets for training large language model
- Benefits:
  - It is freely and publicly available.
  - It reports a higher downstream performance than popular crawl-based datasets C4 and OSCAR.
  - It has been widely used by state-of-the-art models including GPT-J-6B, GPT-NeoX-20B, Jurassic-1, Megatron-Turing NLG 530B, OPT, and WuDao.

* Gao et al., The Pile: An 800GB dataset of diverse text for language modeling, 2020.

Training data in Pythia

The authors trained two copies of the Pythia suite using identical architectures.

- Pile (334B tokens)
- A copy of the Pile after applying near-deduplication with MinHashLSH and a threshold of 0.87 (207B tokens)
    - following the advice that LLMs trained on deduplicated data are better and memorize less of their data (Lee et al., 2021).

Model Architecture in Pythia

- Fully **dense attention** layers are used.
- **Flash Attention** is used during training for improved device throughput.
- **Rotary embeddings** are used as the positional embedding type of choice.
- The parallelized attention, feedforward technique and model initialization methods introduced by **GPT-J** are used.
- **Untied embedding** / unembedding matrices are used to facilitate interpretability research (Belrose et al., 2023).

*Belrose et al., Eliciting latent predictions from transformers with the tuned lens, 2023.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

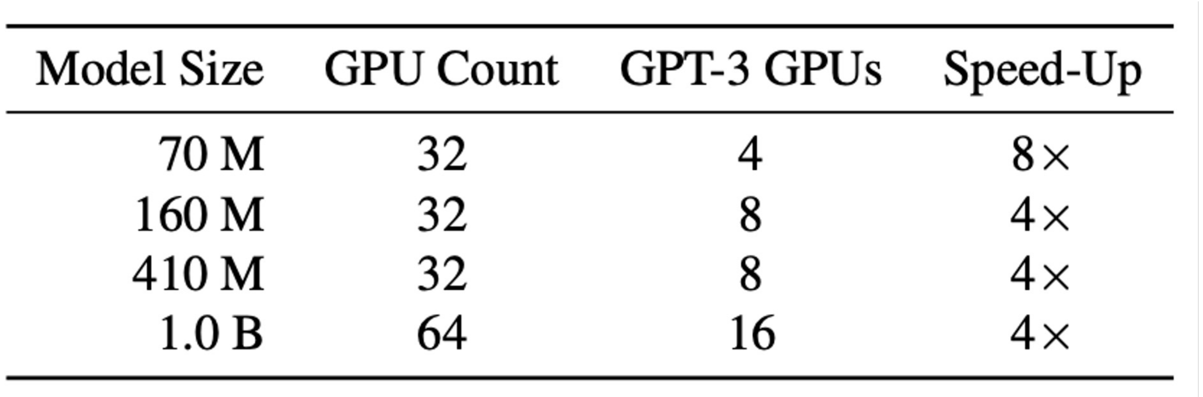
Model Training in Pythia

Training code: open source library GPTNeoX

Optimizer: Adam and the Zero Redundancy Optimizer (ZeRO)

Batch size: 1024 samples with a sequence length of 2048 (2,097,152 tokens)

Epoch: all models are trained for 299,892,736,000 ≈ 300B tokens

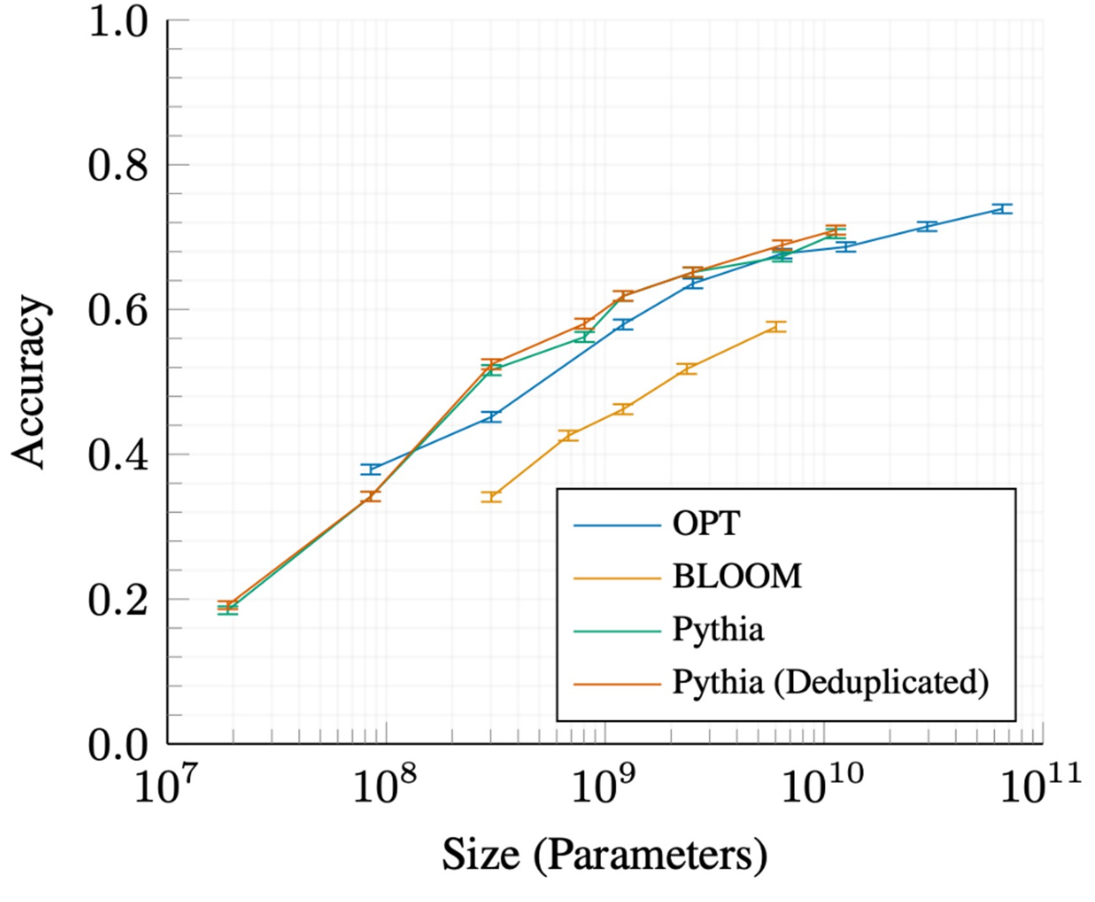


Model Size	GPU Count	GPT-3 GPUs	Speed-Up
70 M	32	4	8×
160 M	32	8	4×
410 M	32	8	4×
1.0 B	64	16	4×

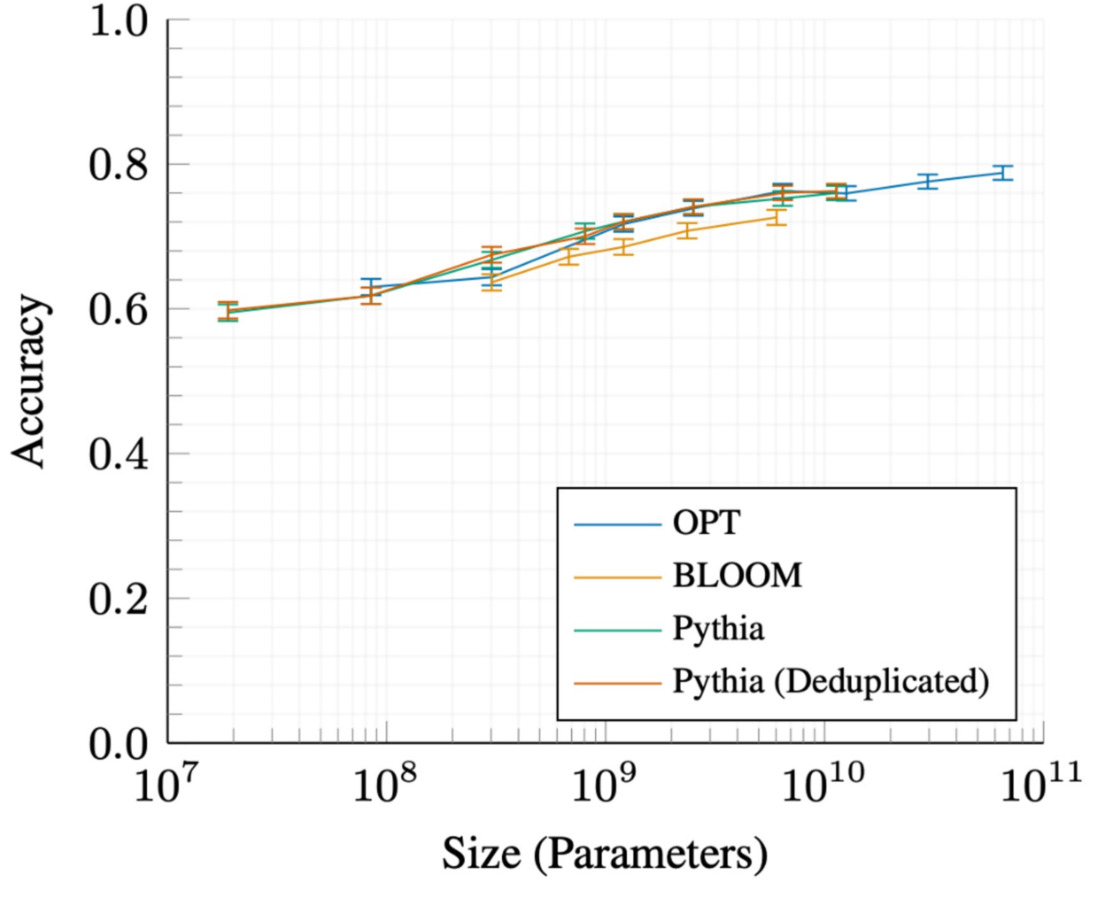
*All GPUs are A100s with 40 GiB VRAM

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Evaluation of Pythia



(a) LAMBADA (OpenAI)



(b) PIQA

Evaluation of Pythia

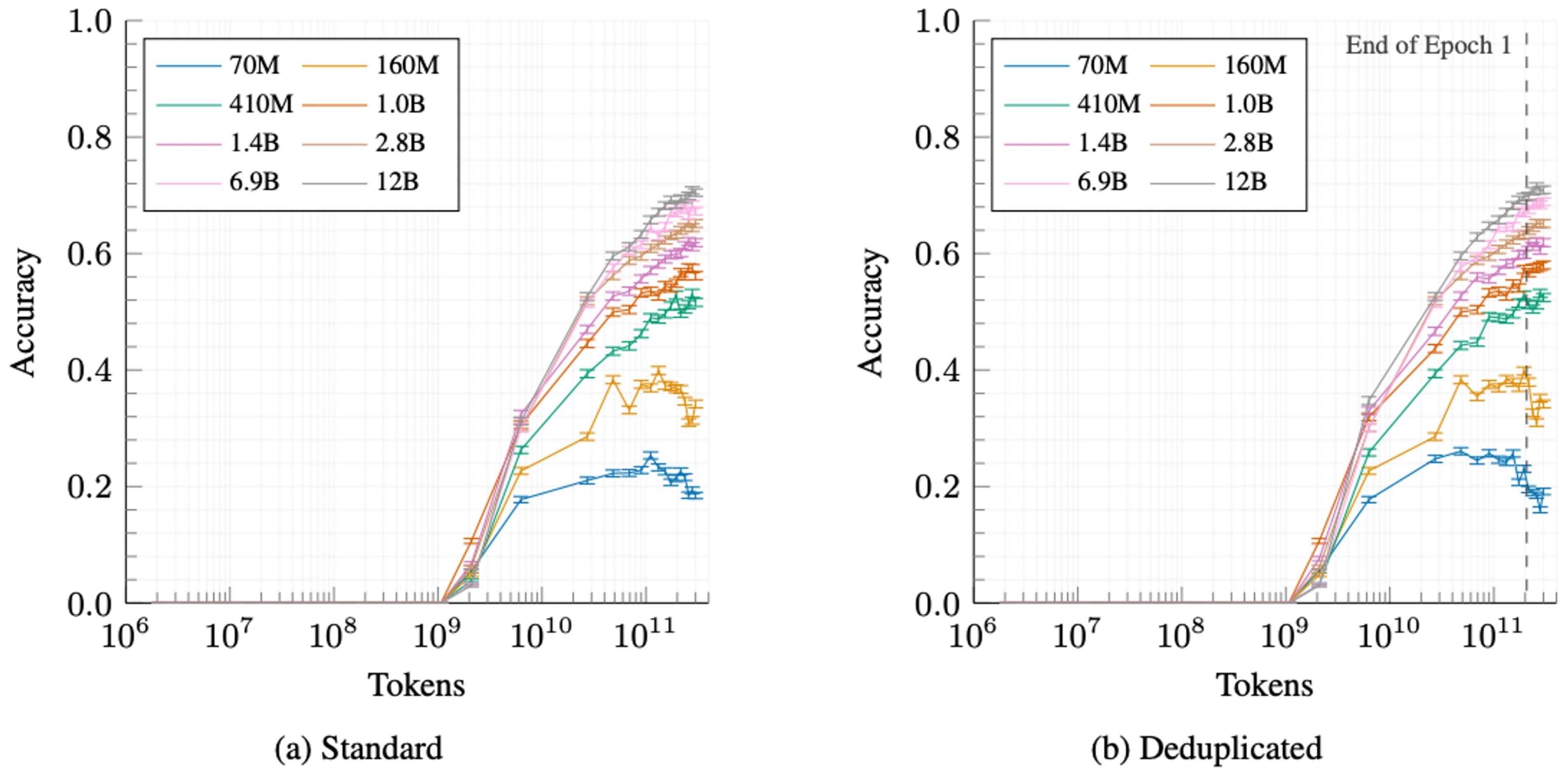


Figure 10. LAMBADA (OpenAI) over the course of training. Left is the standard Pile, while the right is the deduplicated Pile. The dashed line indicates where the deduplicated Pile began its second epoch.

Case Study: How Does Data Bias Influence Learned Behaviors?

- replace morphologically masculine pronouns by their feminine counterparts

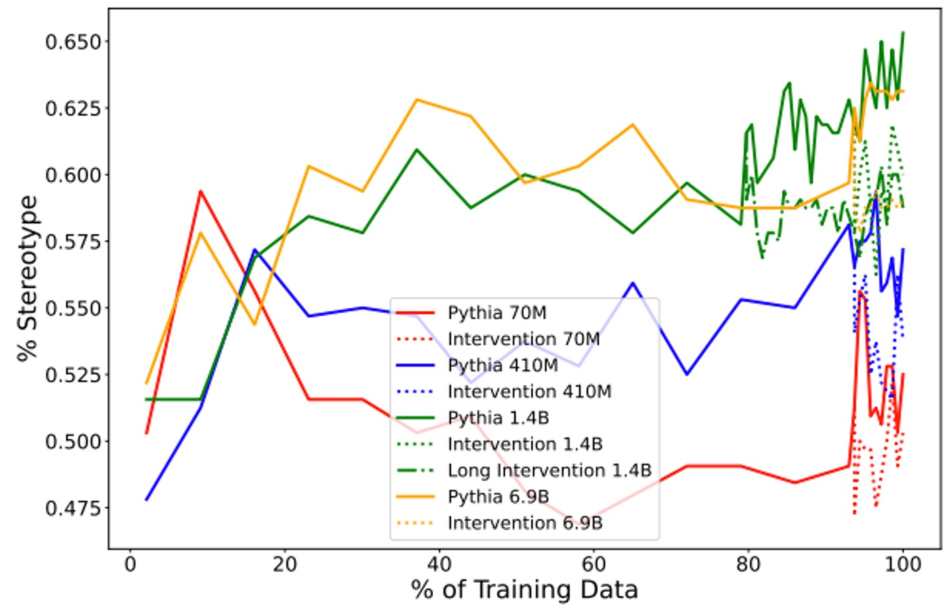


Figure 1. The CrowS-Pairs gender bias, shown as the percentage of times that the perplexity of the stereotyping sentence is lower than its less stereotyped counterpart (% Stereotype) for the Pythia models of different sizes at the end of training. We also show the effect of the gender swapping intervention on the measured bias for the partially retrained models.

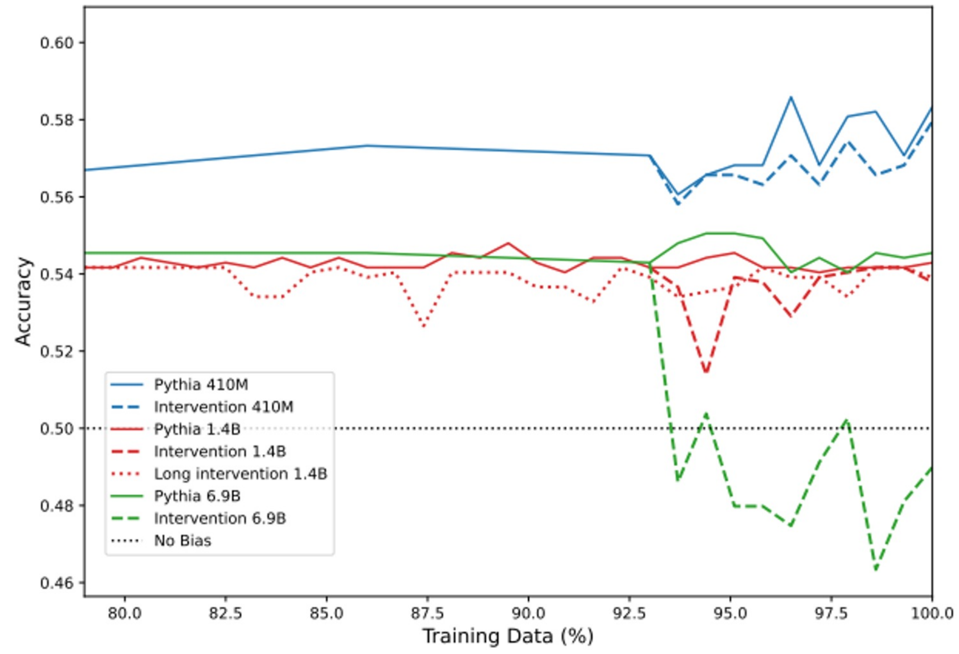


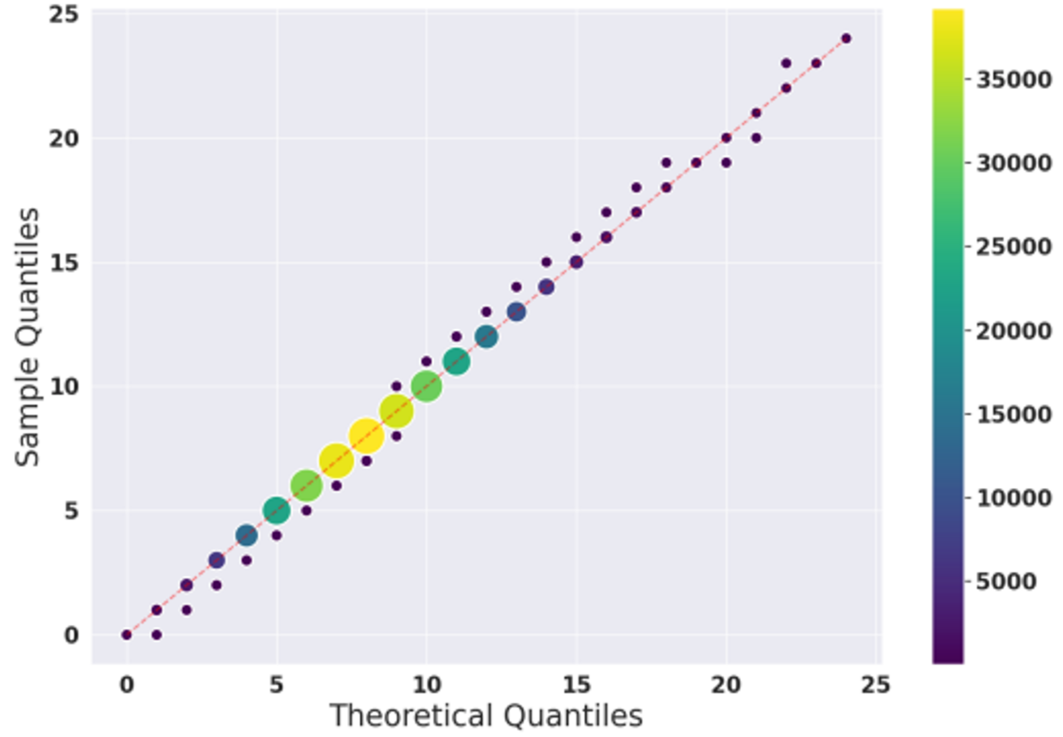
Figure 2. The WinoBias gender bias results, shown as the proportion of the time that the model placed a higher log probability on the more stereotyped pronoun as an answer to a multiple choice gender–occupation co-reference question.

Case Study: Does Training Order Influence Memorization?

Hypothesis: data encountered later in training will be memorized more

Method: they measure the memorization of an initial segment of each sequence in the training corpus

Result: training order has little impact on memorization



Quantile-Quantile plot of rate of occurrence of memorized sequences in 12B model compared to a Poisson Point Process. Color and dot size indicates number of points.

Case Study: Do Pretraining Term Frequencies Influence Task Performance Throughout Training?

Model sizes affect the correlation between average performance and the term frequencies, indicating that this correlation is an emergent property in larger models

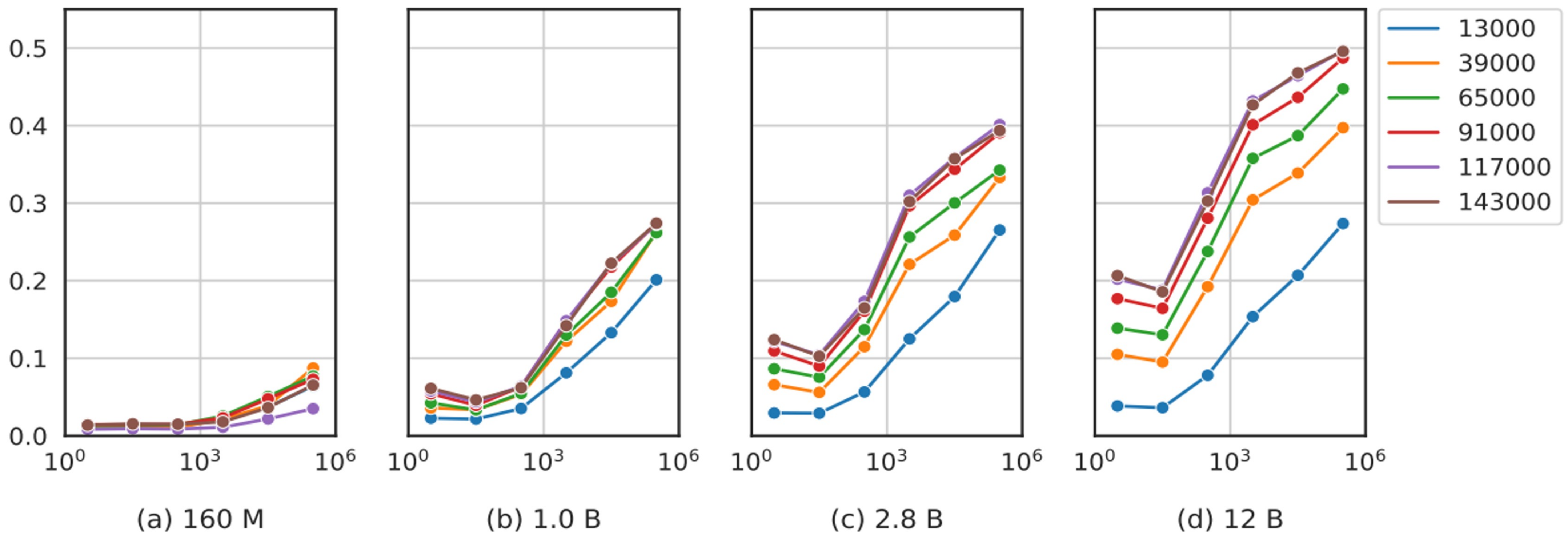


Figure 4. Accuracy on Trivia QA plotted againts the number of relevant entity counts found in a QA-pair. Each subfigure shows the impact of performance across various model sizes over multiple intermediate checkpoints. (With train step counts denoted by color on the right) Each point represents the average accuracy (y-axis) of binned counts (x-axis).

Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems

Miao et al.

Presenter: Sabit Ahmed (bcw3zj)

University of Virginia | ENGINEERING Department of Computer Science

Objective

Overview

- Recent advancements in LLM serving and inference.
- Systematic review and categorization of existing techniques.
- Highlight strengths and limitations of each method.

Coverage

- Decoding algorithms
- Architecture design
- Model compression and low-bit quantization
- Parallel computation and memory management
- Request scheduling and kernel optimization

University of Virginia | ENGINEERING Department of Computer Science

Background of LLM Serving

- Transformer-based LLM
  - Self-attention and FFN layers
- GPU and Other Accelerators
  - Supports higher FLOPS, mixed-precision computing
  - Latest architectures supports FP32, TF32, BF16, INT8, INT4, etc.
  - Other accelerators: CPU, mobile and edge devices, TPUs, FPGAs, etc.
- LLM Inference
  - Leverages Auto-regressive Decoding

Challenges

1. **Latency & Response Time**
   - Balancing speed and complexity for real-time applications
2. **Memory Footprint & Model Size**
   - Deploying on memory-constrained devices
3. **Scalability & Throughput**
   - Handling simultaneous request loads efficiently
4. **Hardware Compatibility & Acceleration**
   - Adapting models to diverse hardware
5. **Accuracy vs. Efficiency**
   - Balancing model size with performance

University of Virginia | ENGINEERING Department of Computer Science

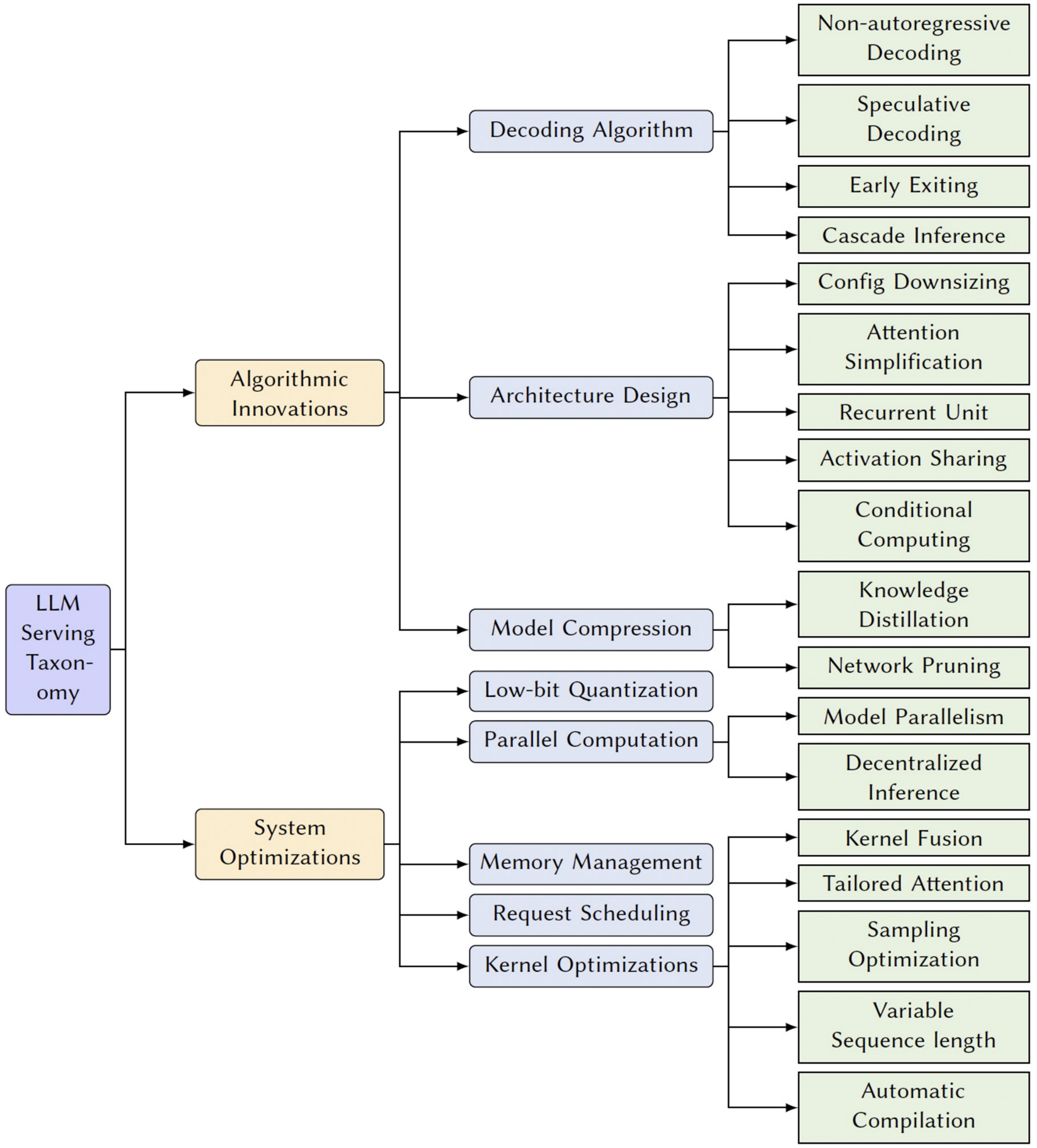
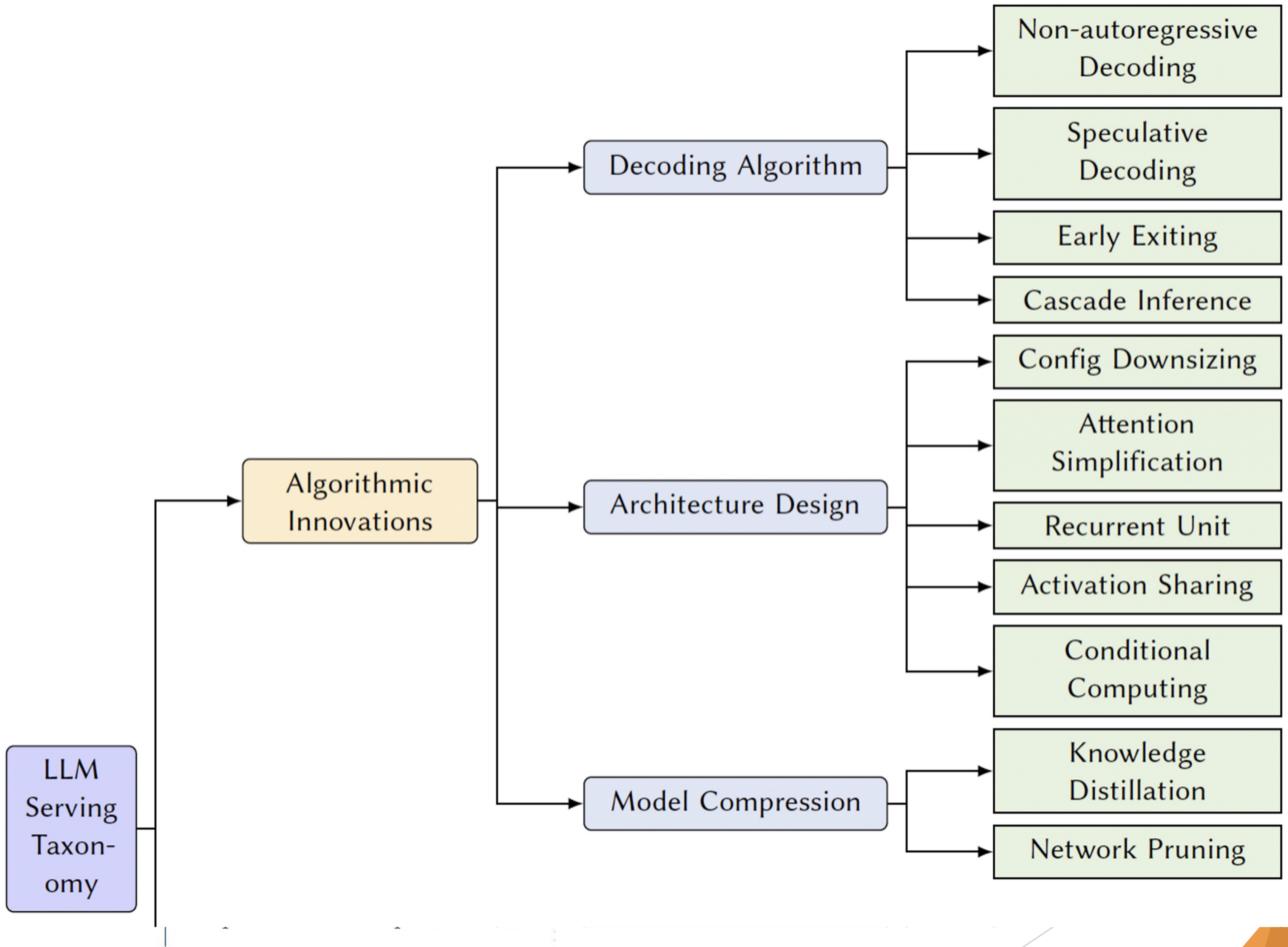


Fig. 1. Taxonomy of LLM Inference Advancements


LLM Serving Taxon-omy
Algorithmic Innovations
Decoding Algorithm
Architecture Design
Model Compression
Non-autoregressive Decoding
Speculative Decoding
Early Exiting
Cascade Inference
Config Downsizing
Attention Simplification
Recurrent Unit
Activation Sharing
Conditional Computing
Knowledge Distillation
Network Pruning

Decoding Algorithm

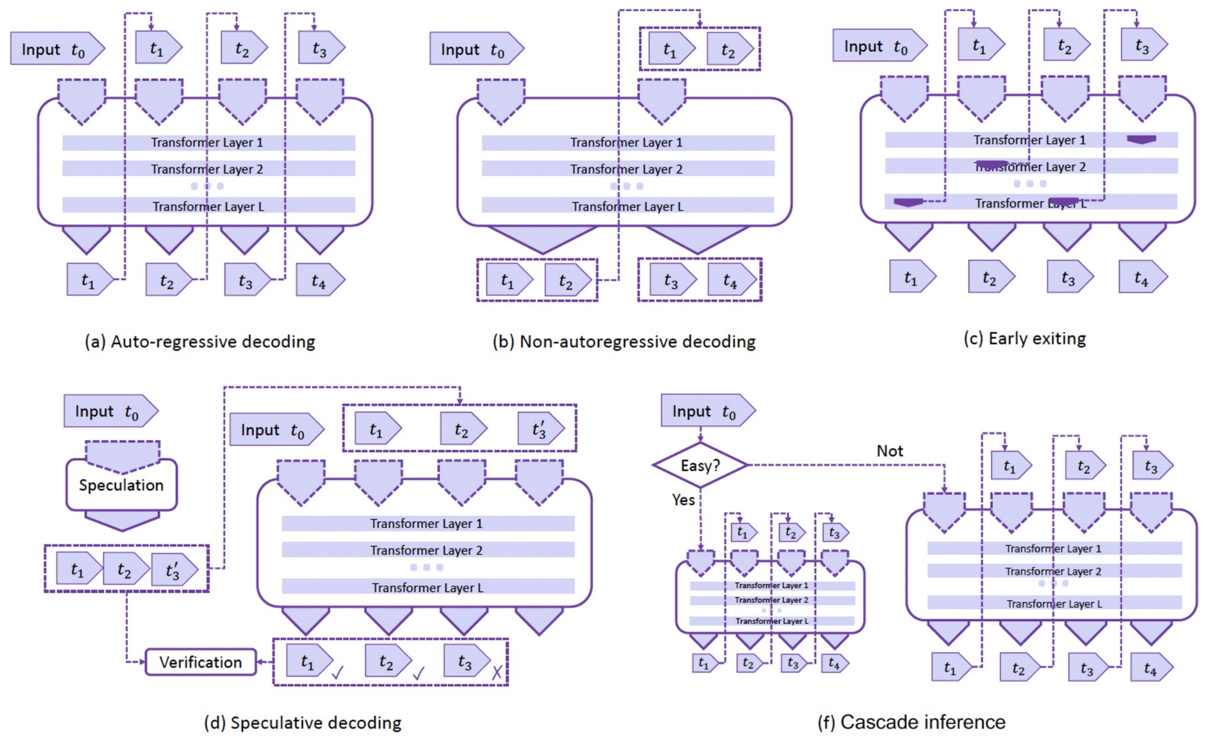
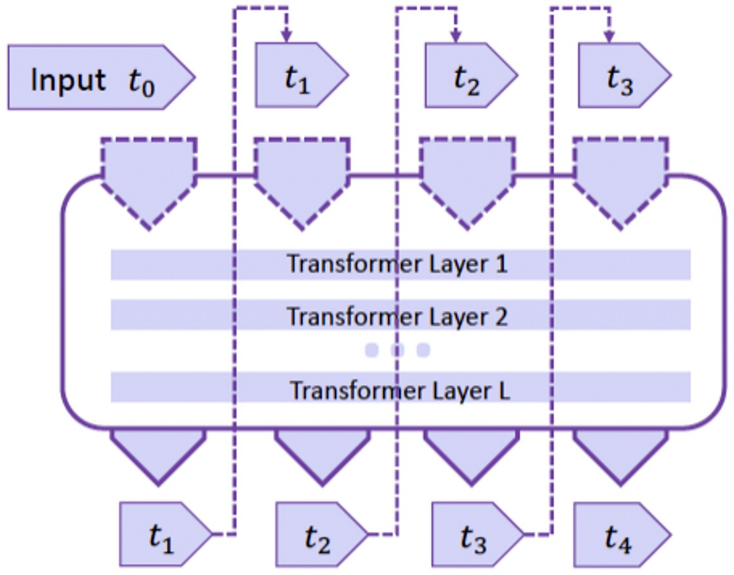


Fig. 2. Illustration of different LLM decoding algorithms

University of Virginia | ENGINEERING Department of Computer Science

Auto-regressive Decoding

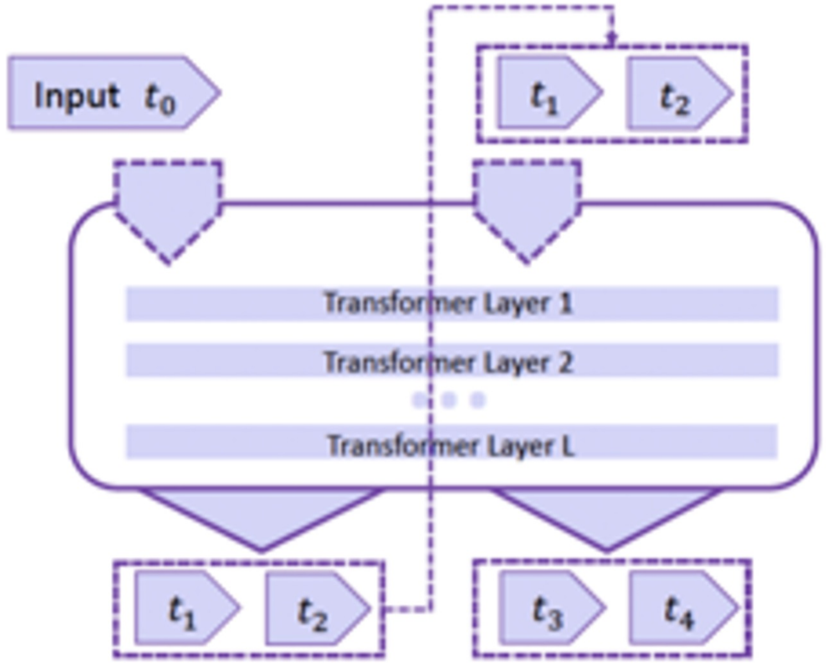
- Sequentially predicting the next token in a sequence, given all the previous ones



(a) Auto-regressive decoding

Auto-regressive Decoding

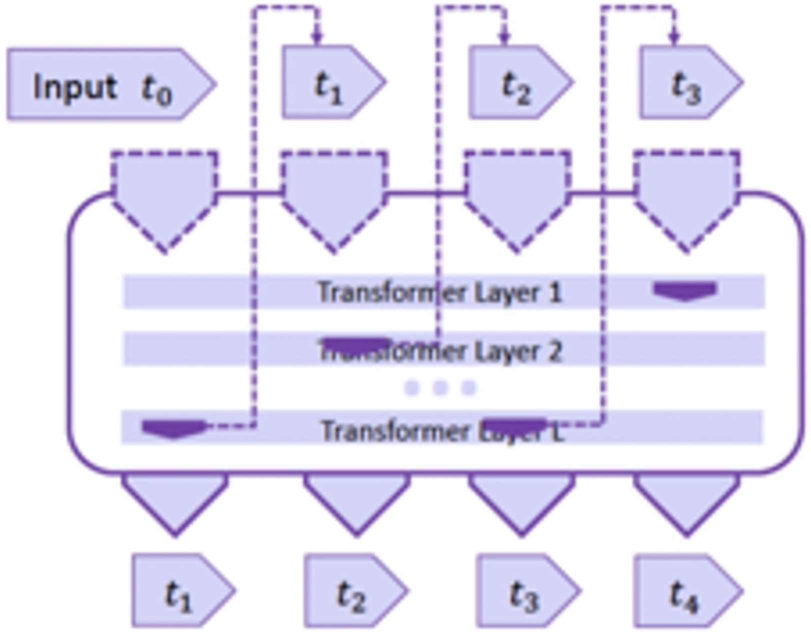
- Decode output tokens in parallel
- Not as reliable as auto-regressive models
- Breaking or modelling word dependencies



(b) Non-autoregressive decoding

Early Exiting

- Utilize multi-layer architecture of existing LLMs
- Adaptive Computation
  - Emit predictions based on internal classifiers instead of running the whole LLM
- Insufficient Information
  - May not faithfully make accurate predictions

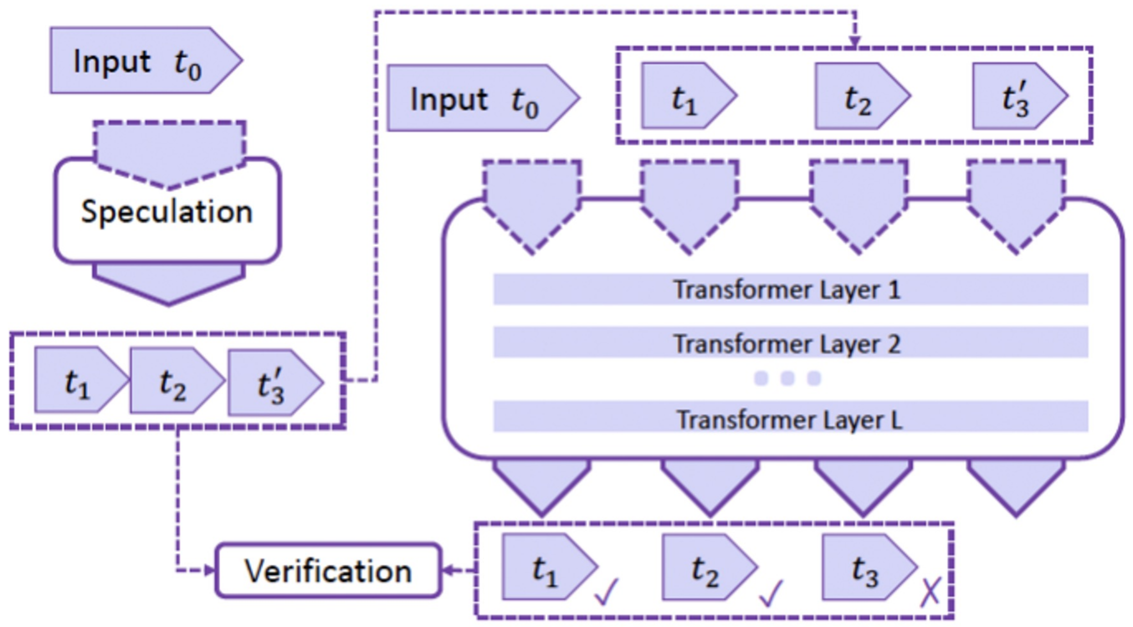


(c) Early exiting

University of Virginia | ENGINEERING Department of Computer Science

Speculative Decoding

- Uses smaller draft model
- Allows parallel decoding
- Verification and Fallback mechanism

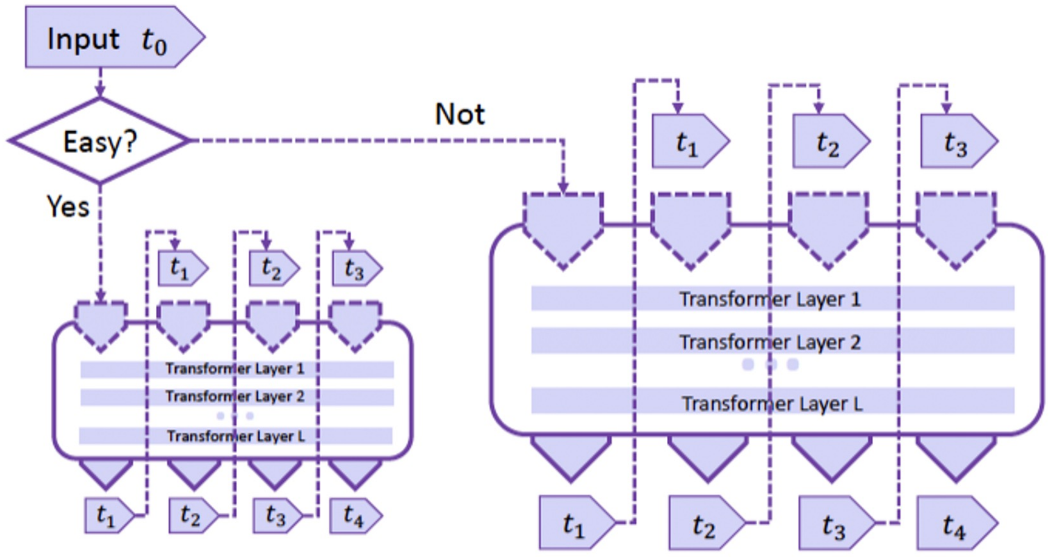


(d) Speculative decoding

University of Virginia | ENGINEERING Department of Computer Science

Cascade Inference

- Internal classifiers organizes queries in a cascade manner
- Adaptively select proper model based on the difficulty level



(f) Cascade inference

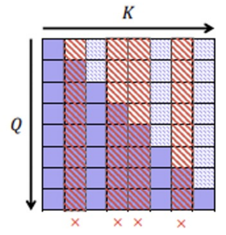
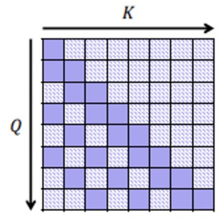
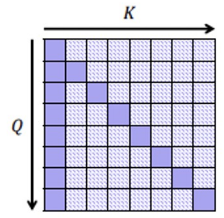
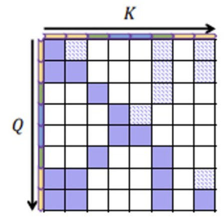
Architecture Design

A few efficient architectures are proposed:

- Configuration downsizing
  - Using shallow encoder or decoders, weight sharing, vocabulary shrinking
- Attention Simplification
  - Sparsification, kernelization, factorization
  - Different types of attention: sliding and dilated attention, hash-based attention, etc.
- Activation Sharing
  - Sharing the intermediate activations to improve efficiency
- Conditional Computing
  - Uses Mixture of Experts (MoE) paradigm: partitions model's capacity into various experts
  - Invoke necessary experts based on routing mechanism
- Recurrent Unit
  - Replace Transformers with recurrent units for linear computation
  - Varieties: Linear Transformer, Attention Free Transformer
  - Overcome $O(L^2)$ bottleneck problem

Architecture Design

Table 1. Comparisons of attention simplification methods in prior efficient Transformers and recent LLMs.

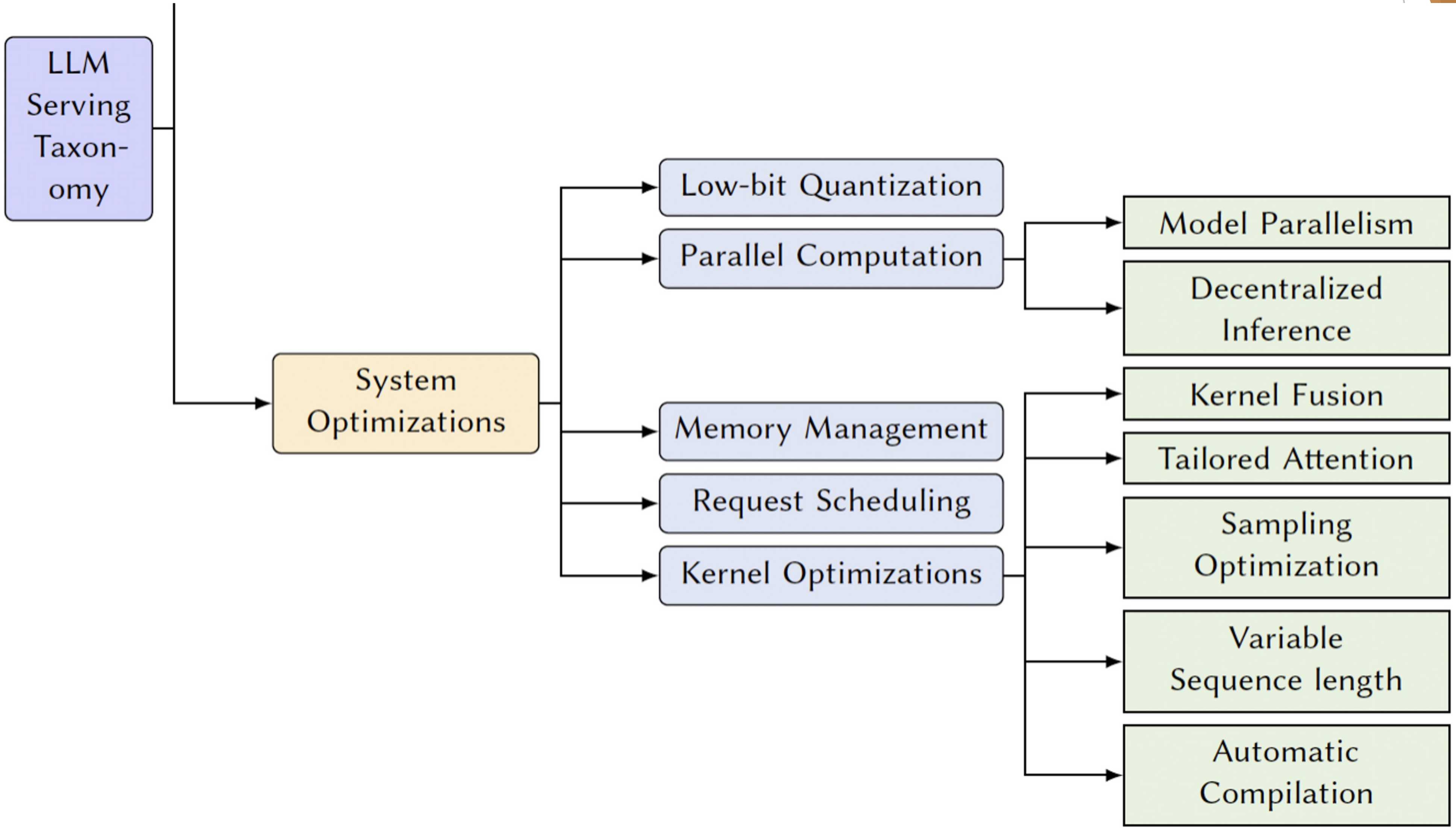
Attention Type	Selective	Sliding + Dilated	Global token	Hash-based
Sparse Pattern				
References	Top-*k* [112], Sorting [239], Adaptive [63], Informer [306]	Sparse Transformer [60], LongFormer [41]	Star Transformer [110], GMAT [111]	Reformer [146], Routing Transformer [211]
LLM Applications	Scissorhands [170], H_2O [301] [36, 130, 159, 186, 301]	Mistral-7B [129], [269], LongNet [74]	StreamingLLM [269], Summary [58], Landmark [184]	Sparse hash attention[198]

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Model Compression

Aims to reduce memory footprint and computational requirements of LLMs

- Knowledge Distillation
  - Student and teacher model
  - White-box and black-box distillation
  - Promising performance compared to GPT models
- Network Pruning
  - Structural pruning on LLMs
  - Facilitates GPU speedups


LLM Serving Taxonomy
System Optimizations
Low-bit Quantization
Parallel Computation
Memory Management
Request Scheduling
Kernel Optimizations
Model Parallelism
Decentralized Inference
Kernel Fusion
Tailored Attention
Sampling Optimization
Variable Sequence length
Automatic Compilation

System Optimization

- Low-bit Quantization
  - Quantize-Aware Training (QAT)
  - Post-Training Quantization (PTQ)
- Parallel Computation
  - Model Parallelism
    - **Tensor Model Parallelism (TP):** Splits model layers for deployment on separate devices
    - **Pipeline Model Parallelism (PP):** Each device is responsible for pipeline stage
    - **Sequence Parallelism (SP):** Splits long sequences across multiple GPUs
    - **Automatic Parallelism:** Uses algorithms for optimal performance without manual tuning
  - Decentralized Inference
    - Combination of model and data parallelism
    - Multiple decentralized nodes collaborate to process data
    - Useful for geographically distributed hardwares

University of Virginia | ENGINEERING Department of Computer Science

System Optimization

- Memory Management

  KV cache grows and shrinks dynamically
  - **Current Approaches:** Naive pre-allocation leads to severe memory wastage
  - **Paged Attention:** Partitions KV cache into non-contiguous blocks, enhancing batch size and throughput.
  - **SpecInfer Tree Attention:** Utilizes tree traversal to avoid redundant KV cache allocations

- Request Scheduling

  Objective: Efficiently scheduling incoming inference requests
  Common Approaches:
  - Dynamic batching for efficient request management.
  - Preemption and priority settings for task handling.
  - Swapping, model selection, and cost-efficient operations.
  - Load balancing and strategic resource allocation.

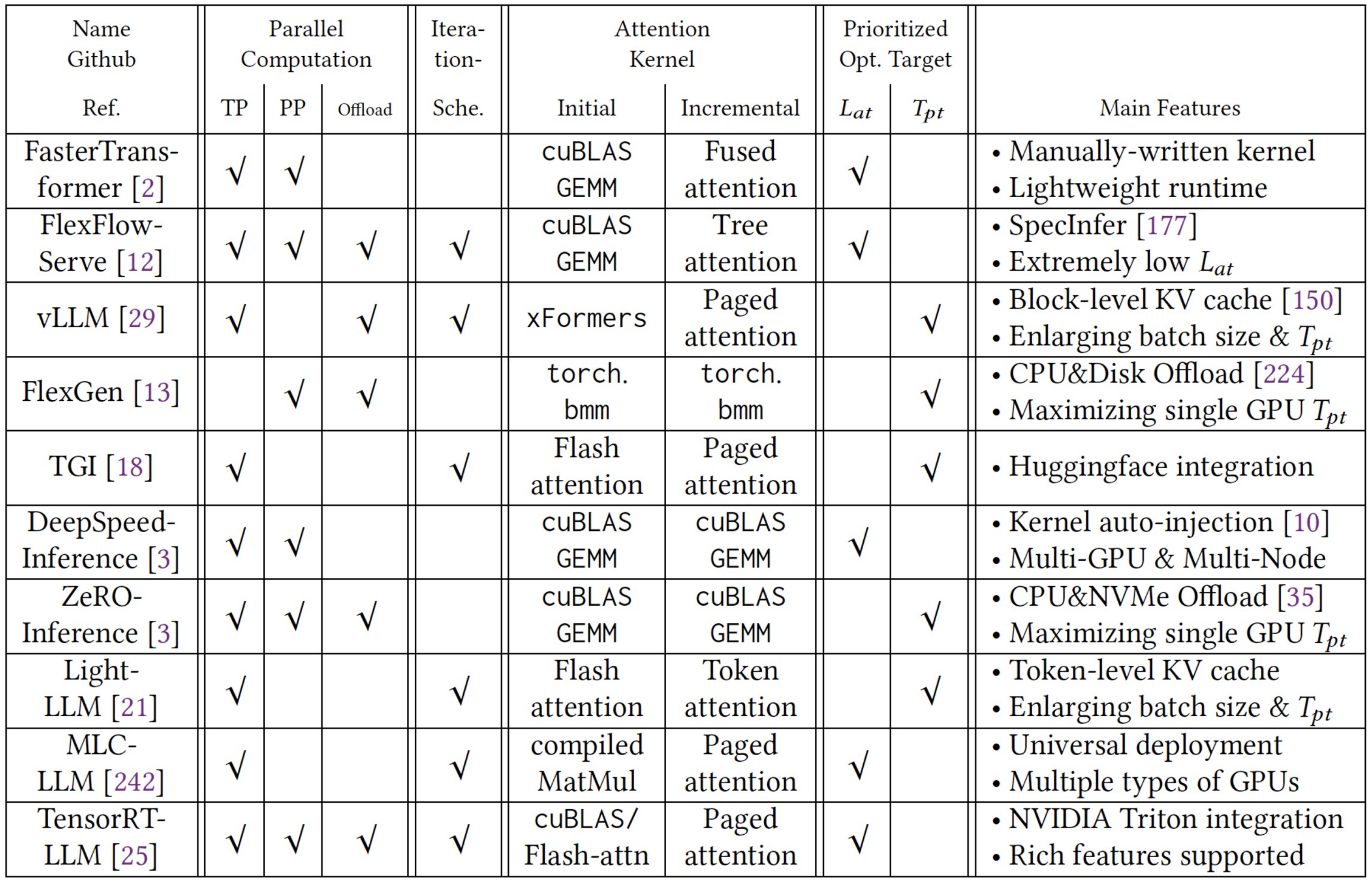
University of Virginia | ENGINEERING Department of Computer Science

System Optimization

- Kernel Optimization
  - Kernel Fusion
  - Tailored Attention
    - Initial/prefill/context phase: processes tokens from input in parallel
    - Incremental/decode/generation phase: generates one output token per iteration
  - Sampling Optimization
    - Various sampling strategies: Greedy sampling, parallel and stochastic sampling, etc.
  - Variable Sequence Length
  - Automatic Compilation

Software Frameworks

Table 2. Comparison of state-of-the-art open-sourced GPU-based LLM serving systems.



Name Github Ref.	Parallel Computation			Itera-tion-Sche.	Attention Kernel		Prioritized Opt. Target		Main Features
	TP	PP	Offload		Initial	Incremental	L_{at}	T_{pt}	
FasterTrans-former [2]	√	√			cuBLAS GEMM	Fused attention	√		• Manually-written kernel • Lightweight runtime
FlexFlow-Serve [12]	√	√	√	√	cuBLAS GEMM	Tree attention	√		• SpecInfer [177] • Extremely low L_{at}
vLLM [29]	√		√	√	xFormers	Paged attention		√	• Block-level KV cache [150] • Enlarging batch size & T_{pt}
FlexGen [13]		√	√		torch.bmm	torch.bmm		√	• CPU&Disk Offload [224] • Maximizing single GPU T_{pt}
TGI [18]	√			√	Flash attention	Paged attention		√	• Huggingface integration
DeepSpeed-Inference [3]	√	√			cuBLAS GEMM	cuBLAS GEMM	√		• Kernel auto-injection [10] • Multi-GPU & Multi-Node
ZeRO-Inference [3]	√	√	√		cuBLAS GEMM	cuBLAS GEMM		√	• CPU&NVMe Offload [35] • Maximizing single GPU T_{pt}
Light-LLM [21]	√			√	Flash attention	Token attention		√	• Token-level KV cache • Enlarging batch size & T_{pt}
MLC-LLM [242]	√			√	compiled MatMul	Paged attention	√		• Universal deployment • Multiple types of GPUs
TensorRT-LLM [25]	√	√	√	√	cuBLAS/ Flash-attn	Paged attention	√		• NVIDIA Triton integration • Rich features supported

Future Direction

- Developing and Enhancing Hardware Accelerators
- Designing Efficient and Effective Decoding Algorithms
- Optimizing Long Context/Sequence Scenarios
- Investigating Alternative Architectures
- Exploration of Deployment in Complex Environments

Sparks of Large Audio Models: A Survey and Outlook

Presented by: Shiyu Feng (eus5fy)

University of Virginia | ENGINEERING Department of Computer Science

Motivation

- Why explore Large Audio Models?
- Importance of audio processing in diverse real-world applications
  - voice-activated assistants, transcription services, hearing aids

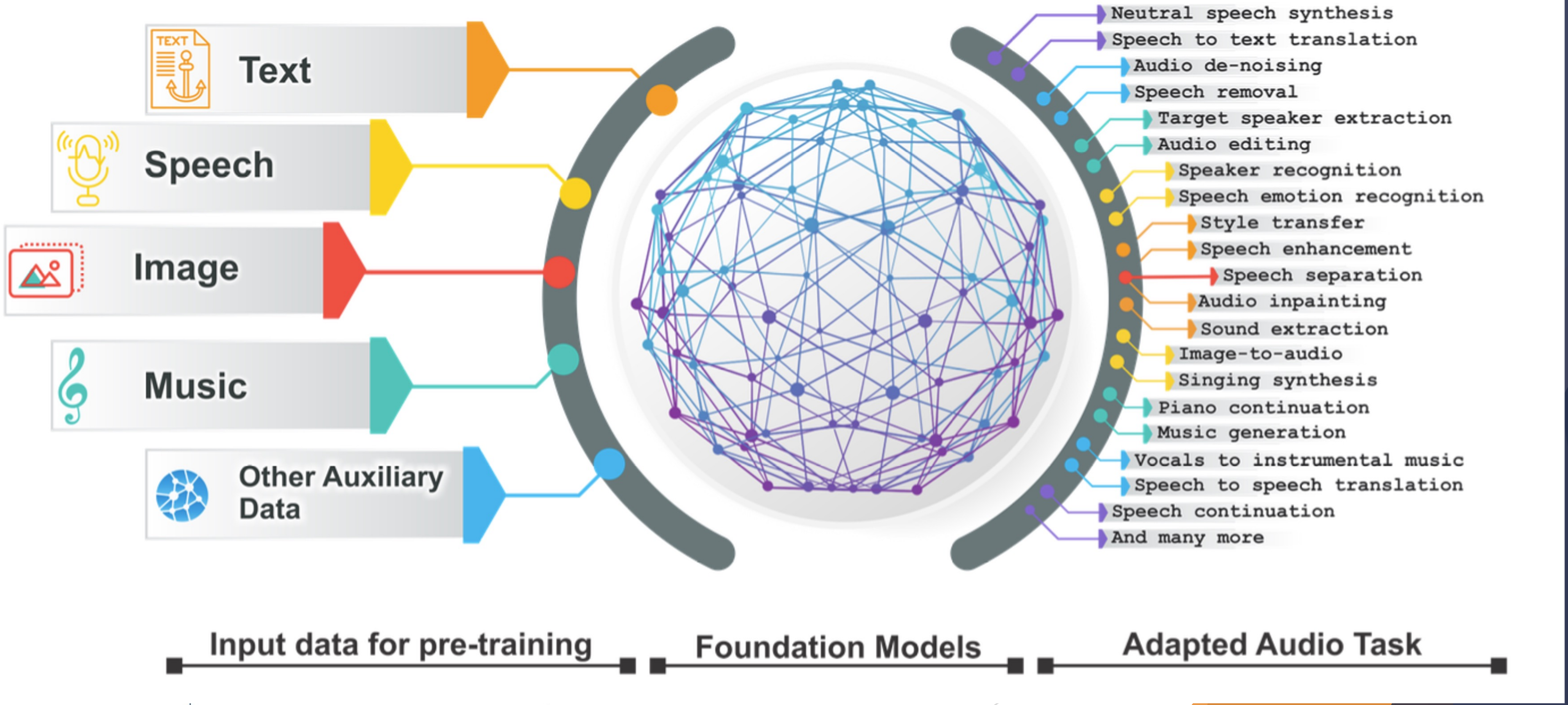
Note: We would not be able to go through all the technical details for every model and application task. We will focus on the main insights and provide one or two examples to elucidate them.

https://github.com/EmulationAI/awesome-large-audio-models

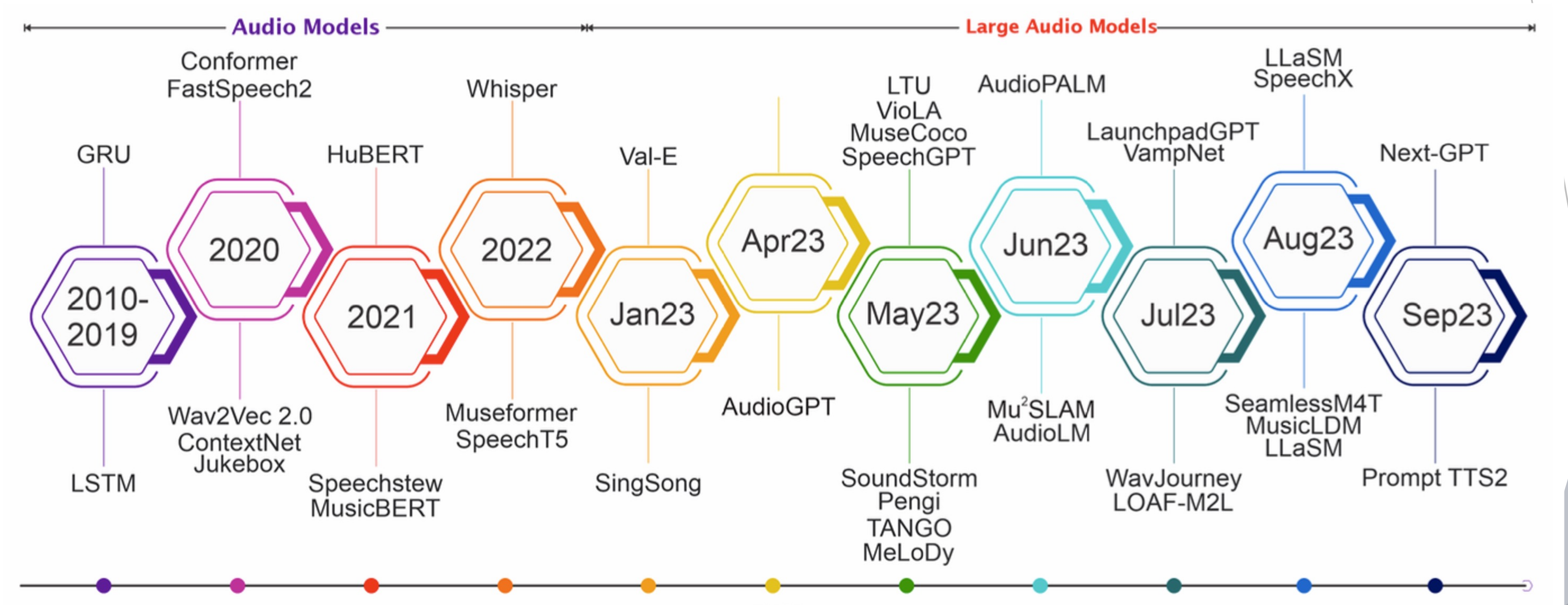
University of Virginia | ENGINEERING Department of Computer Science

Foundational Audio Models

aggregates information from diverse data modalities. Once trained, this model can be tailored to various downstream audio tasks.



Large Audio Models



Time line of Large Audio Models

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Application

- Speech processing
  - Automatic Speech Recognition (ASR), Text-To-Speech (TTS), Speech Translation (ST), Spoken Dialogue Systems (SDSs)
  - Challenges
    - handle variations in accents, background noise, and ensure accurate transcription and synthesis of speech
- Music signal processing
  - Music generation, Analyzing musical patterns, Enhancing music composition
  - Challenges
    - modeling complex musical structures, capturing emotional and creative aspects of music, and ensuring coherence and sophistication in generated music.

Audio datasets

ASR: automatic speech recognition

ST: speech translation

MT: machine translation

AC: audio classification

SED: sound event detection

AMG: affective music generation

MAG: music analysis and generation

MU: music understanding

SC: sound classification

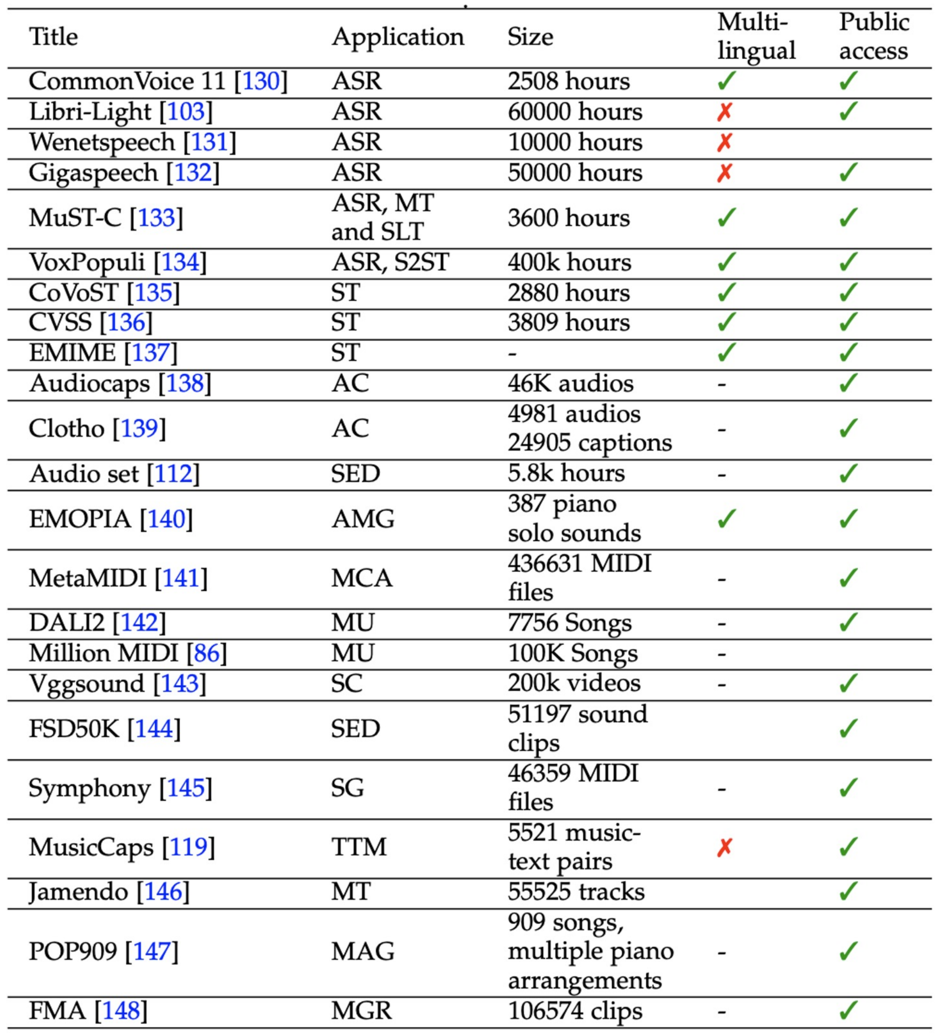
SG: symphony generation

TTM: text to music

MT: music tagging

MAG: Music Arrangement Generation

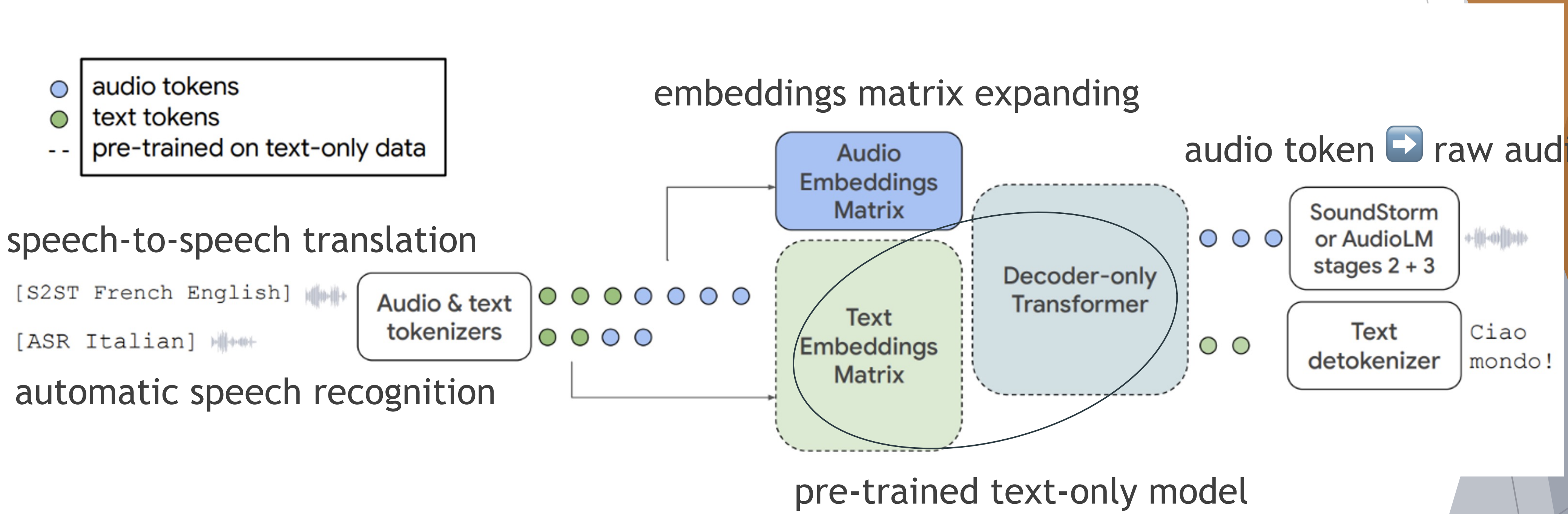
MGR: Music Genre Recognition



Title	Application	Size	Multi-lingual	Public access
CommonVoice 11 [130]	ASR	2508 hours	✓	✓
Libri-Light [103]	ASR	60000 hours	✗	✓
Wenetspeech [131]	ASR	10000 hours	✗	
Gigaspeech [132]	ASR	50000 hours	✗	✓
MuST-C [133]	ASR, MT and SLT	3600 hours	✓	✓
VoxPopuli [134]	ASR, S2ST	400k hours	✓	✓
CoVoST [135]	ST	2880 hours	✓	✓
CVSS [136]	ST	3809 hours	✓	✓
EMIME [137]	ST	-	✓	✓
Audiocaps [138]	AC	46K audios	-	✓
Clotho [139]	AC	4981 audios 24905 captions	-	✓
Audio set [112]	SED	5.8k hours	-	✓
EMOPIA [140]	AMG	387 piano solo sounds	✓	✓
MetaMIDI [141]	MCA	436631 MIDI files	-	✓
DALI2 [142]	MU	7756 Songs	-	✓
Million MIDI [86]	MU	100K Songs	-	
Vggsound [143]	SC	200k videos	-	✓
FSD50K [144]	SED	51197 sound clips		✓
Symphony [145]	SG	46359 MIDI files	-	✓
MusicCaps [119]	TTM	5521 music-text pairs	✗	✓
Jamendo [146]	MT	55525 tracks		✓
POP909 [147]	MAG	909 songs, multiple piano arrangements	-	✓
FMA [148]	MGR	106574 clips	-	✓

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Speech Processing – AudioPalm



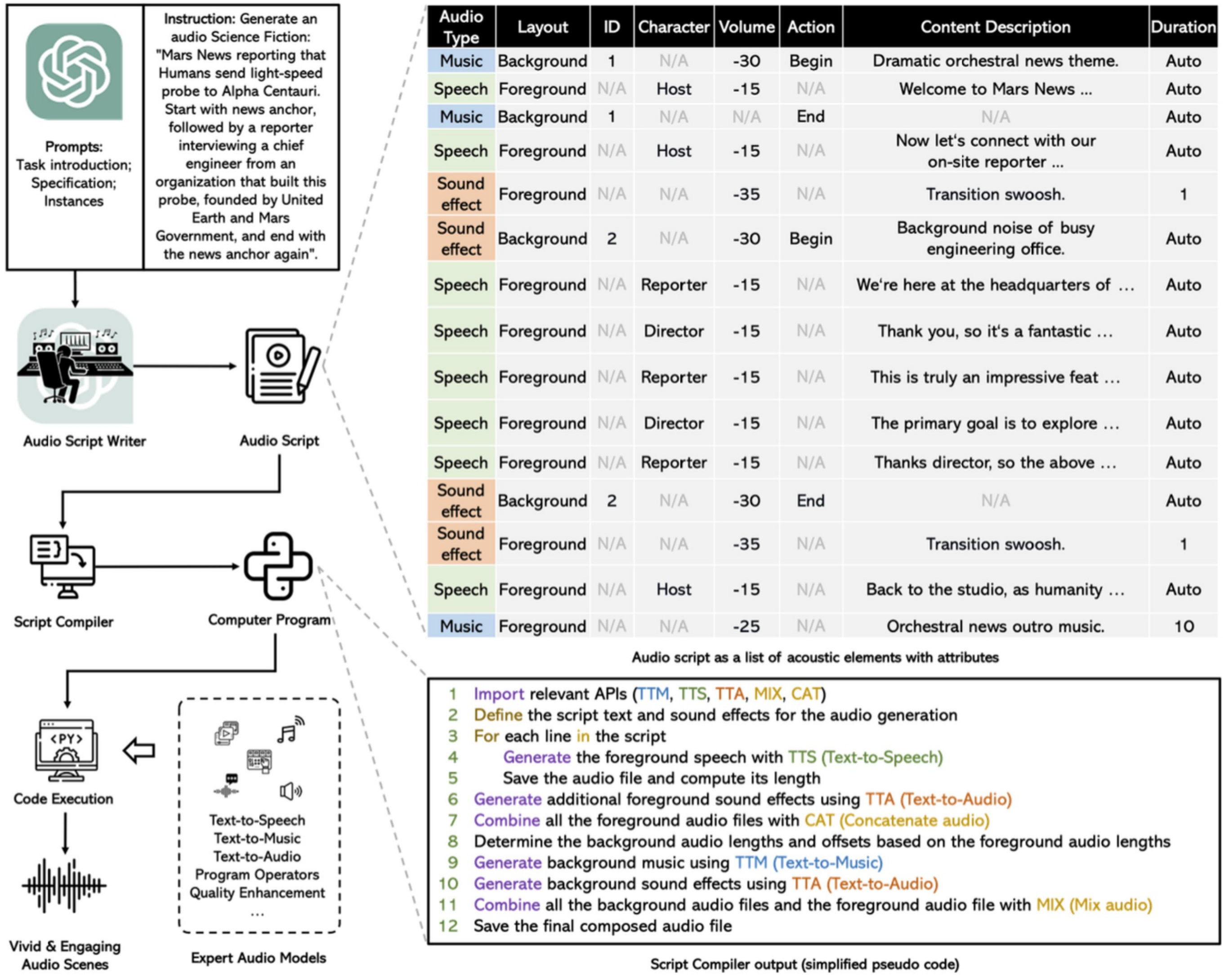
tokenisation of text and audio ➡️ modification of pre-trained text decoders ➡️ transformation of the model's output into audio

Music Signal Processing – WavJourney

Textual story narration

⬇️

speech + music + sound effects



Instruction: Generate an audio Science Fiction: "Mars News reporting that Humans send light-speed probe to Alpha Centauri. Start with news anchor, followed by a reporter interviewing a chief engineer from an organization that built this probe, founded by United Earth and Mars Government, and end with the news anchor again".

Audio Type	Layout	ID	Character	Volume	Action	Content Description	Duration
Music	Background	1	N/A	-30	Begin	Dramatic orchestral news theme.	Auto
Speech	Foreground	N/A	Host	-15	N/A	Welcome to Mars News ...	Auto
Music	Background	1	N/A	N/A	End	N/A	Auto
Speech	Foreground	N/A	Host	-15	N/A	Now let's connect with our on-site reporter ...	Auto
Sound effect	Foreground	N/A	N/A	-35	N/A	Transition swoosh.	1
Sound effect	Background	2	N/A	-30	Begin	Background noise of busy engineering office.	Auto
Speech	Foreground	N/A	Reporter	-15	N/A	We're here at the headquarters of ...	Auto
Speech	Foreground	N/A	Director	-15	N/A	Thank you, so it's a fantastic ...	Auto
Speech	Foreground	N/A	Reporter	-15	N/A	This is truly an impressive feat ...	Auto
Speech	Foreground	N/A	Director	-15	N/A	The primary goal is to explore ...	Auto
Speech	Foreground	N/A	Reporter	-15	N/A	Thanks director, so the above ...	Auto
Sound effect	Background	2	N/A	-30	End	N/A	Auto
Sound effect	Foreground	N/A	N/A	-35	N/A	Transition swoosh.	1
Speech	Foreground	N/A	Host	-15	N/A	Back to the studio, as humanity ...	Auto
Music	Foreground	N/A	N/A	-25	N/A	Orchestral news outro music.	10

Audio script as a list of acoustic elements with attributes

```
Import relevant APIs (TTM, TTS, TTA, MIX, CAT)
Define the script text and sound effects for the audio generation
For each line in the script
    Generate the foreground speech with TTS (Text-to-Speech)
    Save the audio file and compute its length
Generate additional foreground sound effects using TTA (Text-to-Audio)
Combine all the foreground audio files with CAT (Concatenate audio)
Determine the background audio lengths and offsets based on the foreground audio lengths
Generate background music using TTM (Text-to-Music)
Generate background sound effects using TTA (Text-to-Audio)
Combine all the background audio files and the foreground audio file with MIX (Mix audio)
Save the final composed audio file
```

Script Compiler output (simplified pseudo code)

Challenges

- Data Issues (pre-training period)
  - Duplicated data instances
    - model memorization and performance degradation
  - Data contamination
    - affects the performance of benchmarking LAMs, with issues like background noise, audio distortion, and offensive content
  - Concerns of personally identifiable information
  - Need for diverse pre-training data
- Tokenisation
  - Variations in pronunciations and overlapping speech
  - Multilingual speech
  - Emotion tokenization and information loss
- Computational Cost and Energy Requirements
  - Pre-training
  - Fine-tuning

Challenges (Cont.)

- Limited context length
  - Difficulty in understanding long-term dependencies and relationships
- Understanding paralinguistic information
  - Emotions, prosody, and other paralinguistic cues are key for speech and music
- Prompt Sensitivity
  - vulnerable to prompt variations
- Hallucination
  - misinterpretations of audio sources, introduction of random noise
- Ethics
  - bias, privacy concerns, misuse

University of Virginia | ENGINEERING Department of Computer Science

MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training

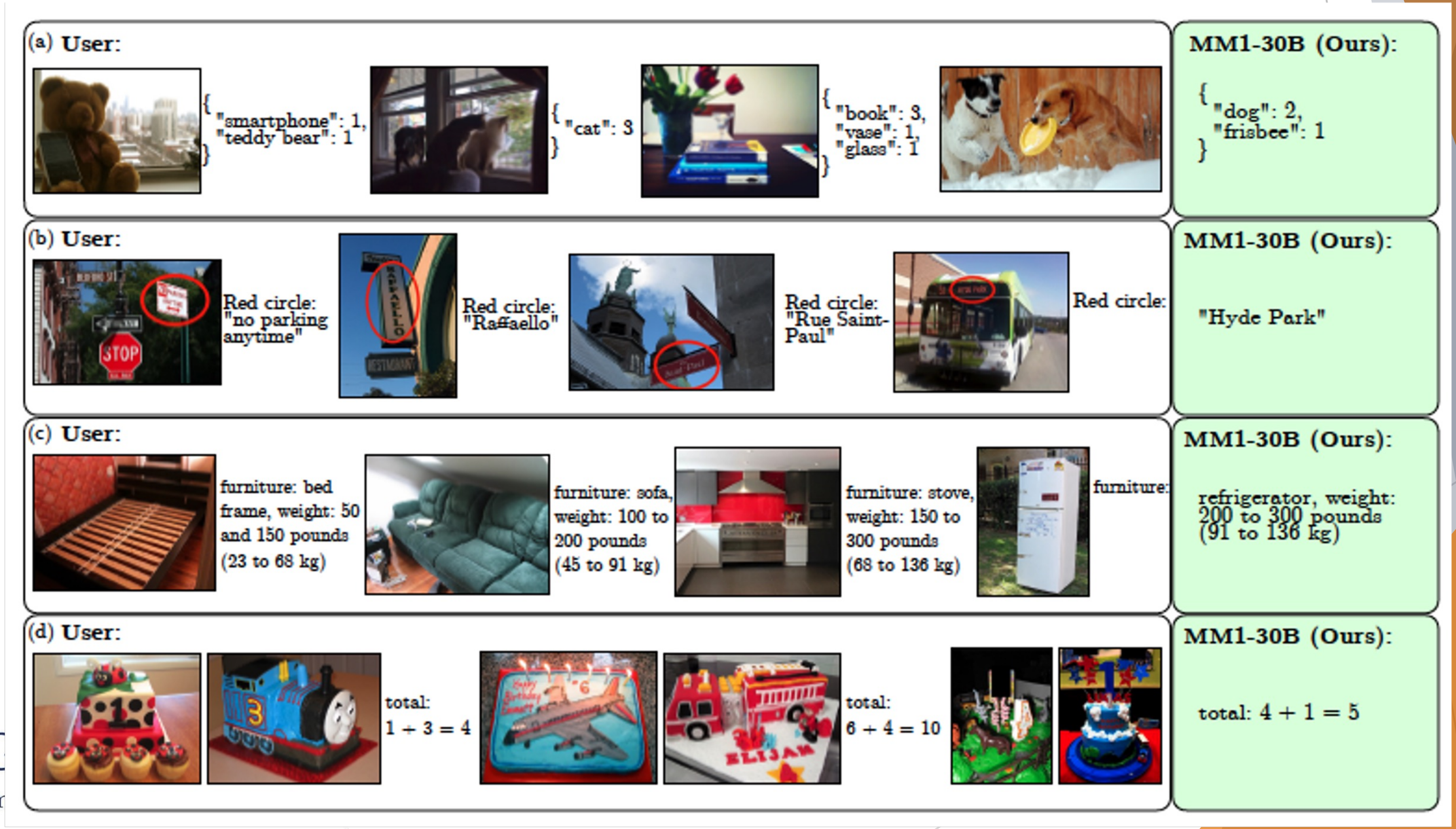
McKinzie et. al, 2024

Apple

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Motivation:

- There are has great development in multi-modal large language models (MLLMs) in the past few years.
    - Flamingo
    - GPT-4V
    - Gemini
    - LLaVA
    - etc


(a) User:
{ "smartphone": 1, "teddy bear": 1 }
{ "cat": 3 }
{ "book": 3, "vase": 1, "glass": 1 }
MM1-30B (Ours):
{ "dog": 2, "frisbee": 1 }
(b) User:
Red circle: "no parking anytime"
Red circle: "Raffaello"
Red circle: "Rue Saint-Paul"
Red circle:
MM1-30B (Ours):
"Hyde Park"
(c) User:
furniture: bed frame, weight: 50 and 150 pounds (23 to 68 kg)
furniture: sofa, weight: 100 to 200 pounds (45 to 91 kg)
furniture: stove, weight: 150 to 300 pounds (68 to 136 kg)
furniture:
MM1-30B (Ours):
refrigerator, weight: 200 to 300 pounds (91 to 136 kg)
(d) User:
total: 1 + 3 = 4
total: 6 + 4 = 10
MM1-30B (Ours):
total: 4 + 1 = 5

UNIVERSITY of VIRGINIA | ENG
Departm

Motivation:

- What are the best design choices when developing a MLLM?
  - Best architecture design?
  - Best training procedure?
  - Best data to use?

Contributions

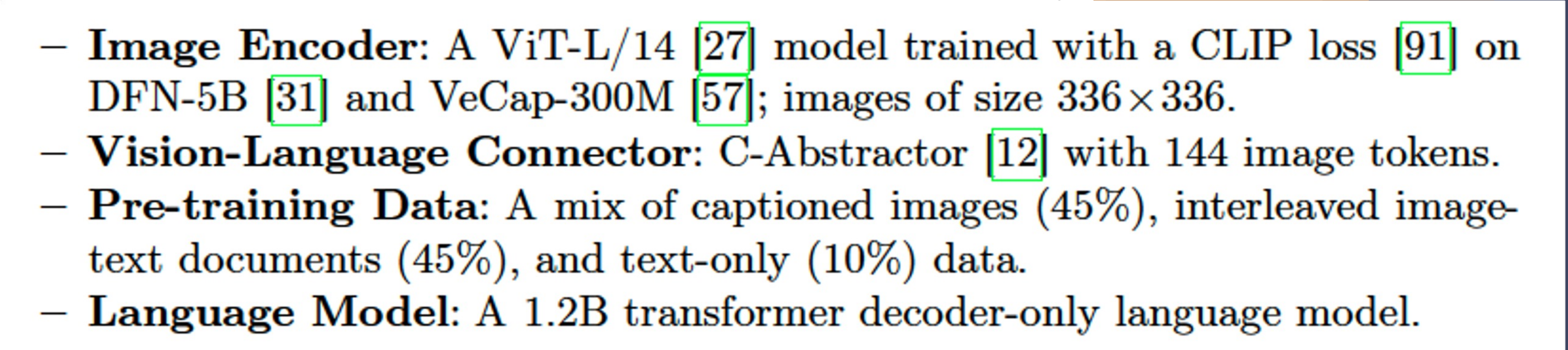
- To answer these questions the authors conducts a fine-grained ablation across:
  - Model architecture
  - Type of data
  - Training procedure
- Based on their findings, they also create their family of MM1 models, which exhibit SOTA performance on captioning and visual question answering (VQA).

Ablation Setup

- Base configuration

- Ablate on component (either model architecture or data source) at a time.

- Evaluate design decision in both a zero-shot and few-shot setting on various image captioning and VQA tasks.



– **Image Encoder**: A ViT-L/14 [27] model trained with a CLIP loss [91] on DFN-5B [31] and VeCap-300M [57]; images of size 336×336.
– **Vision-Language Connector**: C-Abstractor [12] with 144 image tokens.
– **Pre-training Data**: A mix of captioned images (45%), interleaved image-text documents (45%), and text-only (10%) data.
– **Language Model**: A 1.2B transformer decoder-only language model.

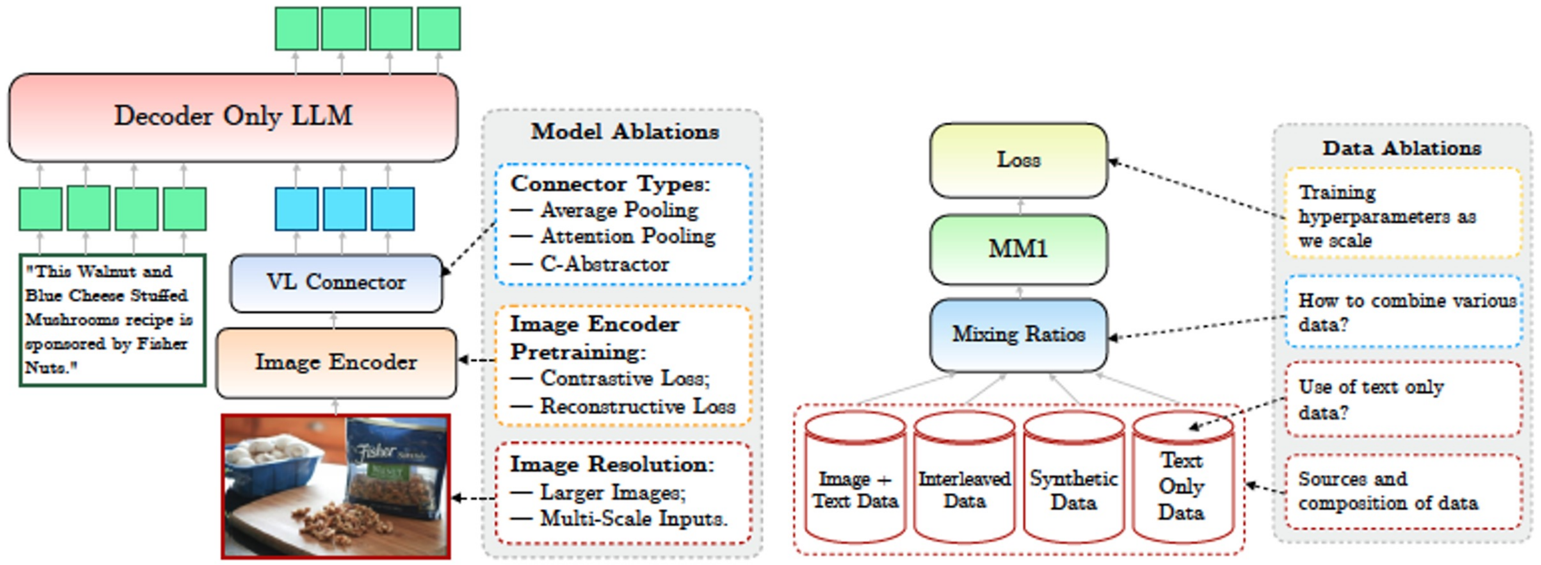


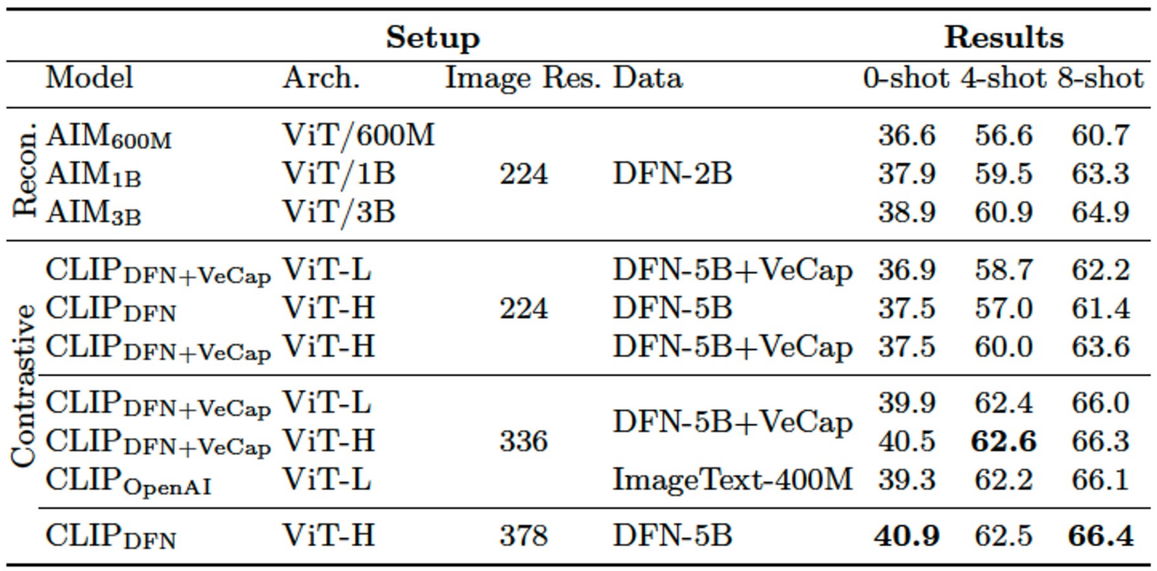
Fig. 3: *Left:* Model ablations: what visual encoder to use, how to feed rich visual data, and how to connect the visual representation to the LLM. *Right:* Data ablations: type of data, and their mixture.

Model Architecture Ablations

- Two motivating questions:
  - How to best pre-train a visual encoder?
  - How to bridge visual features to the LLM space?

Model Architecture Ablations: Image-Encoder Pre-Training

- Image-enoder projects images with their captions into a visual space.
- Let's look at the effect that contrastive loss, reconstructive loss, and image resolution has:
  - Image resolutions has the biggest impact…
    - Higher resolution -> better
  - then model size….
    - Larger model -> better
  - and finally training data composition.
    - Adding a synthetic caption dataset
    - (VeCap-300M) helped increase performance



	Setup				Results		
	Model	Arch.	Image Res.	Data	0-shot	4-shot	8-shot
Recon.	AIM$_{600M}$	ViT/600M	224	DFN-2B	36.6	56.6	60.7
	AIM$_{1B}$	ViT/1B			37.9	59.5	63.3
	AIM$_{3B}$	ViT/3B			38.9	60.9	64.9
Contrastive	CLIP$_{DFN+VeCap}$	ViT-L	224	DFN-5B+VeCap	36.9	58.7	62.2
	CLIP$_{DFN}$	ViT-H		DFN-5B	37.5	57.0	61.4
	CLIP$_{DFN+VeCap}$	ViT-H		DFN-5B+VeCap	37.5	60.0	63.6
	CLIP$_{DFN+VeCap}$	ViT-L	336	DFN-5B+VeCap	39.9	62.4	66.0
	CLIP$_{DFN+VeCap}$	ViT-H			40.5	**62.6**	66.3
	CLIP$_{OpenAI}$	ViT-L		ImageText-400M	39.3	62.2	66.1
	CLIP$_{DFN}$	ViT-H	378	DFN-5B	**40.9**	62.5	**66.4**

Table 1: MM1 pre-training ablation across different image encoders (with 2.9B LLM). Note that the values in the Data column correspond to the data that was used for the initial training of the image encoder itself, not MM1. Recon.: Reconstructive loss. AIM: [30]; DFN-2/5B: [31]; VeCap: VeCap-300M [57]; OpenAI [91].

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Model Architecture Ablations: Vision-Language Connector

- The vision-language connector projects the visual representation into the same space as the LLM.
- Let's see the effect of the number of visual tokens, the image resolution, average pooling, attention pooling, and convolutional mapping has:
- Authors found that:
  - The number of visual tokens and image resolution matter most! (More the better)
  - The type of VL connector has little effect.

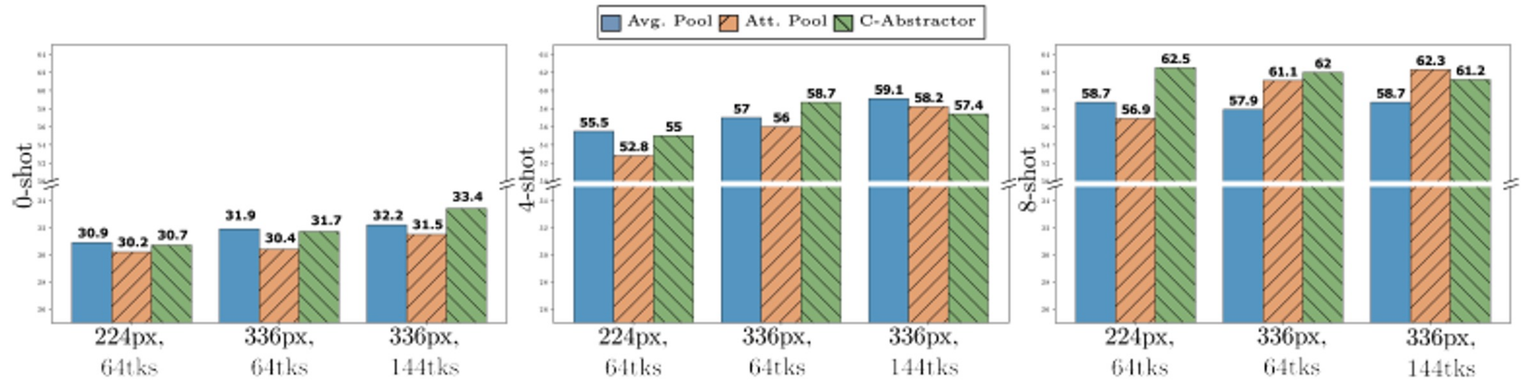


Fig. 4: 0-shot, 4-shot, and 8-shot ablations across different visual-language connectors for two image resolutions, and two image token sizes.

Model Architecture Ablations: Pre-training Data

- Let’s see the effect captioned images, interleaved images and text, and only text has on pre-training.

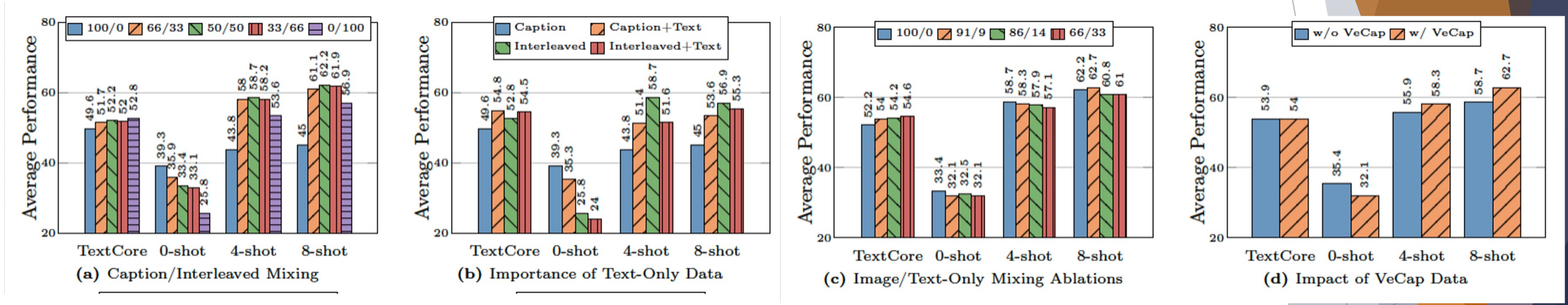
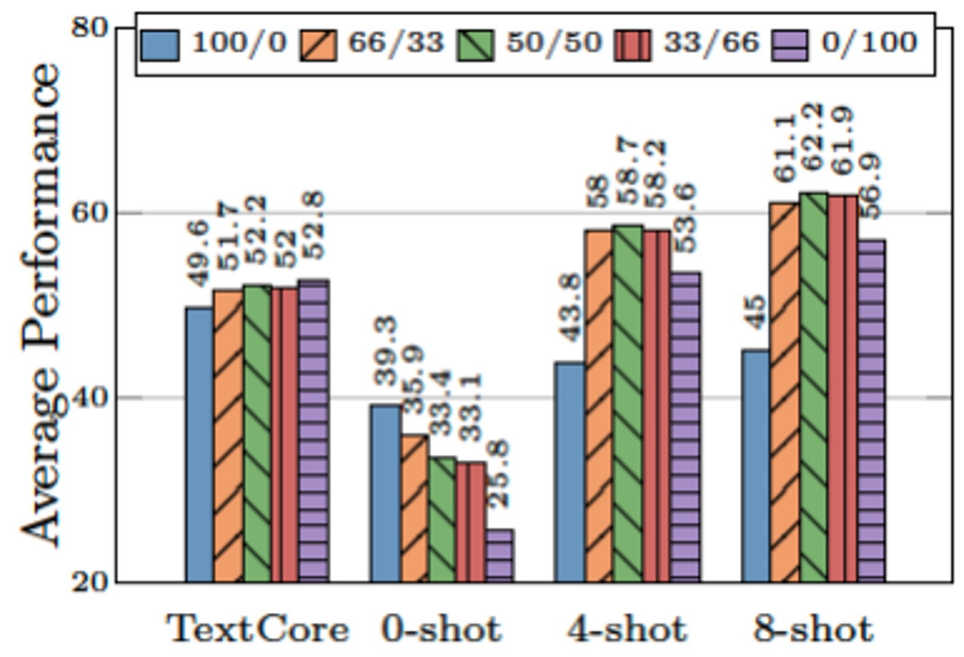


Fig. 5: Data Ablations. For each ablation, we present four different metrics: TextCore, 0-shot, 4-shot, and 8-shot. **(a)** Results with image data where we present five different mixing ratios between interleaved and captioned data. **(b)** Results with and without text-only data. We mix the text-only data separately with captioned and interleaved data. **(c)** Results with different mixing ratios between image data (caption and interleaved) and text-only data. **(d)** Results with and without including VeCap as part of caption data.

Model Architecture Ablations: Pre-training Data

- Interleaved data is vital for few-shot and text-only performance
- Caption data improves zero-shot performance

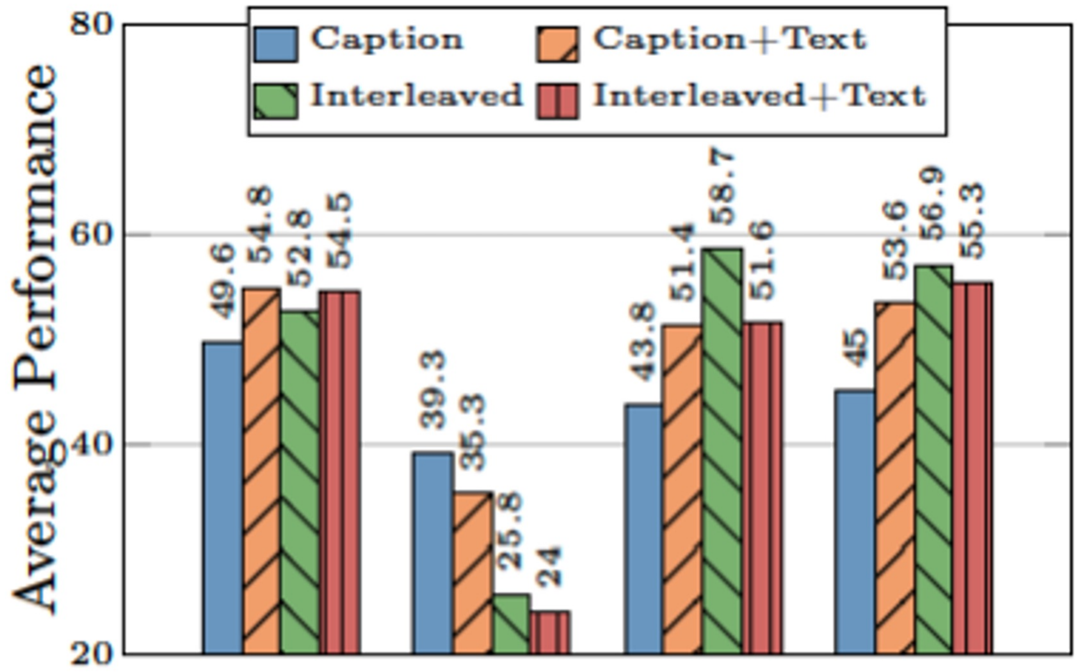


(a) Caption/Interleaved Mixing

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Model Architecture Ablations: Pre-training Data

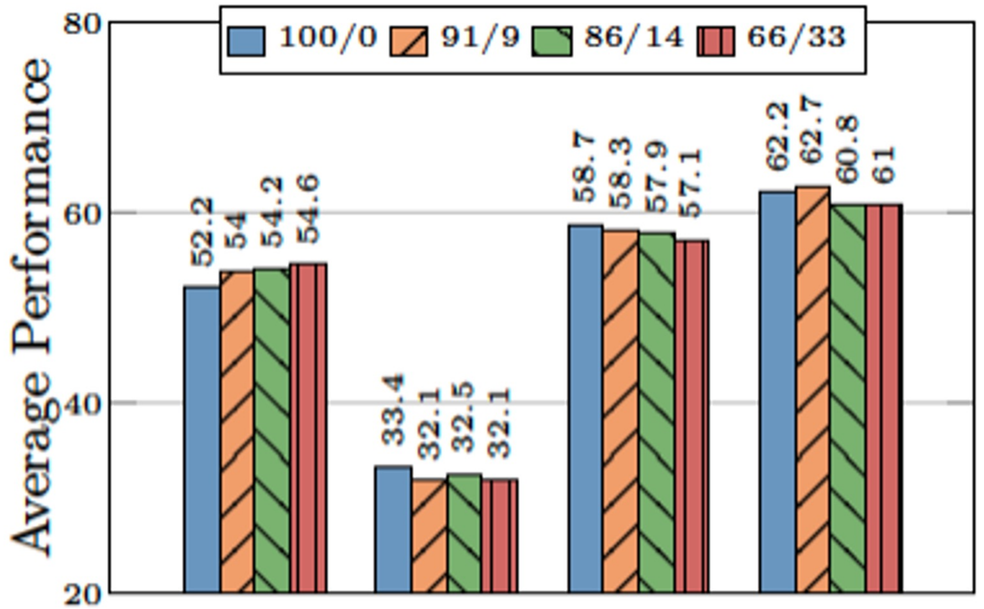
- Text-only data only improves few-shot and text-only performance.



(b) Importance of Text-Only Data

Model Architecture Ablations: Pre-training Data

- Thoughtfully mixing text and image data can lead to optimal multi-modal performance while maintaining text performance.

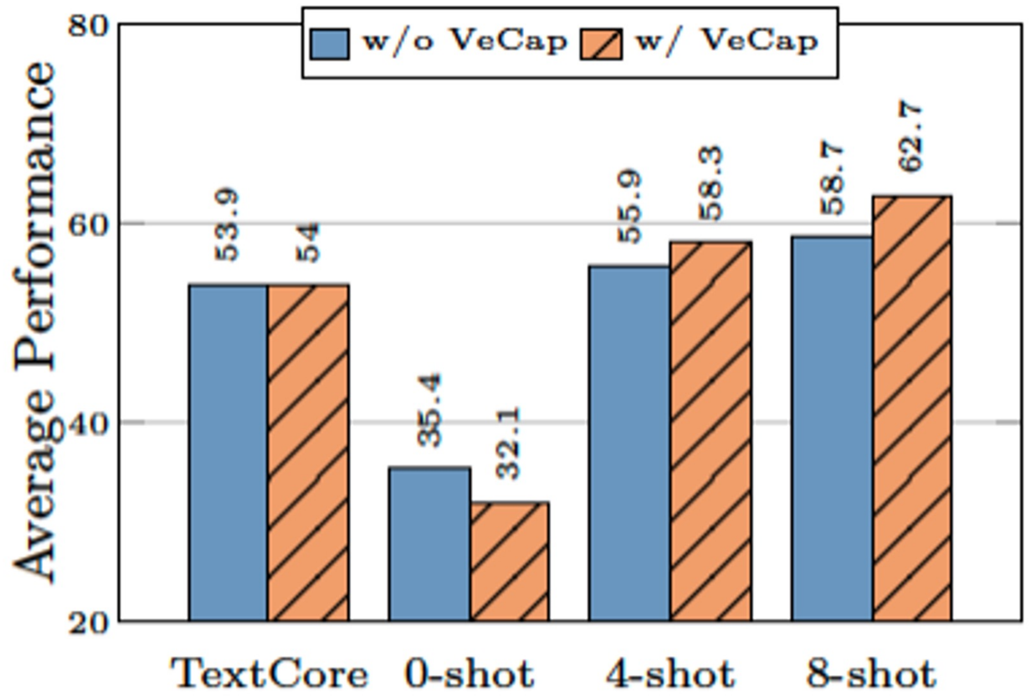


(c) Image/Text-Only Mixing Ablations

University of Virginia ENGINEERING Department of Computer Science

Model Architecture Ablations: Pre-training Data

- Synthetic data helps with few-shot learning



(d) Impact of VeCap Data

Building the MM1 Model

- Image-encoder:
  - ViT-H model with 378x378 resolution, pretrained with CLIP objective on DFN-5B dataset
  - (motivated by importance of high image resolution)
- Vision-language connector:
  - C-abstractor with 144 tokens.
  - (Motivated by importance of many image tokens).
- Data:
  - 45% interleaved image-text documents
  - 45% image-text pair documents
  - 10% text-only documents
  - (Motivated to maintain a balance between zero-shot and few-shot performance

Model Scaling

- Initial Grid Search at Smaller Scales:
  - Conducted a grid search for optimal learning rates at smaller model sizes (9M, 85M, 302M, 1.2B parameters) to gather data efficiently without excessive computational costs.
- Utilized linear regression in log space based on smaller models to predict optimal learning rates for larger scales, resulting in the formula:

$$\eta = \exp(-0.4214 \ln(N) - 0.5535)$$

- Replaced traditional validation loss metrics with direct 8-shot task performance to optimize learning rates, focusing on real-world applicability.
- Simple Scaling Rule for Weight Decay:
  - Adopted a simple rule to scale weight decay proportionally to the learning rate, setting $\lambda=0.1\eta$, ensuring consistency across different model sizes.

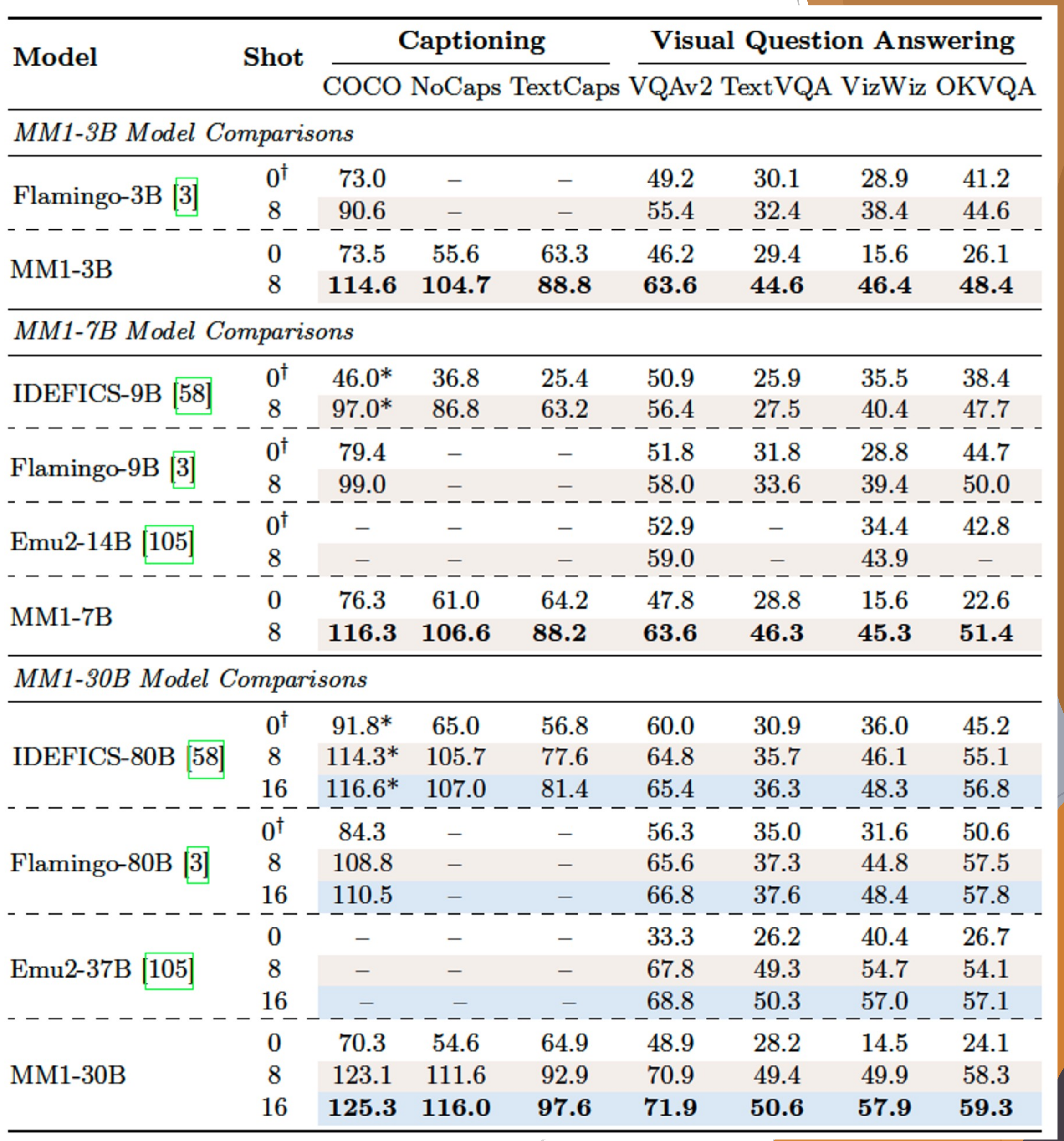
Model Scaling with MoE

- Mixture-of-Experts (MoE) scales up the total number of model parameters while maintaining constant activated parameters per instance, enhancing model capacity without significantly impacting inference speed.
- Two specific models were designed:
  - a 3B-MoE with 64 experts and a
  - 7B-MoE with 32 experts
- To convert a dense model to MoE, only the dense language decoder is replaced with an MoE decoder, while other components like the image encoder remain unchanged.
- MoE models use the same training hyperparameters and conditions as the dense models, ensuring consistency in the training process.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

Pre-training Results

- The family of MM1 models beat baselines in both Caption and VQA.
- Notably, MM1-30B can beat Flamingo 80B



Model	Shot	Captioning			Visual Question Answering			
		COCO	NoCaps	TextCaps	VQAv2	TextVQA	VizWiz	OKVQA
MM1-3B Model Comparisons								
Flamingo-3B [3]	$0^{\dagger}$	73.0	–	–	49.2	30.1	28.9	41.2
	8	90.6	–	–	55.4	32.4	38.4	44.6
MM1-3B	0	73.5	55.6	63.3	46.2	29.4	15.6	26.1
	8	**114.6**	**104.7**	**88.8**	**63.6**	**44.6**	**46.4**	**48.4**
MM1-7B Model Comparisons								
IDEFICS-9B [58]	$0^{\dagger}$	46.0*	36.8	25.4	50.9	25.9	35.5	38.4
	8	97.0*	86.8	63.2	56.4	27.5	40.4	47.7
Flamingo-9B [3]	$0^{\dagger}$	79.4	–	–	51.8	31.8	28.8	44.7
	8	99.0	–	–	58.0	33.6	39.4	50.0
Emu2-14B [105]	$0^{\dagger}$	–	–	–	52.9	–	34.4	42.8
	8	–	–	–	59.0	–	43.9	–
MM1-7B	0	76.3	61.0	64.2	47.8	28.8	15.6	22.6
	8	**116.3**	**106.6**	**88.2**	**63.6**	**46.3**	**45.3**	**51.4**
MM1-30B Model Comparisons								
IDEFICS-80B [58]	$0^{\dagger}$	91.8*	65.0	56.8	60.0	30.9	36.0	45.2
	8	114.3*	105.7	77.6	64.8	35.7	46.1	55.1
	16	116.6*	107.0	81.4	65.4	36.3	48.3	56.8
Flamingo-80B [3]	$0^{\dagger}$	84.3	–	–	56.3	35.0	31.6	50.6
	8	108.8	–	–	65.6	37.3	44.8	57.5
	16	110.5	–	–	66.8	37.6	48.4	57.8
Emu2-37B [105]	0	–	–	–	33.3	26.2	40.4	26.7
	8	–	–	–	67.8	49.3	54.7	54.1
	16	–	–	–	68.8	50.3	57.0	57.1
MM1-30B	0	70.3	54.6	64.9	48.9	28.2	14.5	24.1
	8	123.1	111.6	92.9	70.9	49.4	49.9	58.3
	16	**125.3**	**116.0**	**97.6**	**71.9**	**50.6**	**57.9**	**59.3**

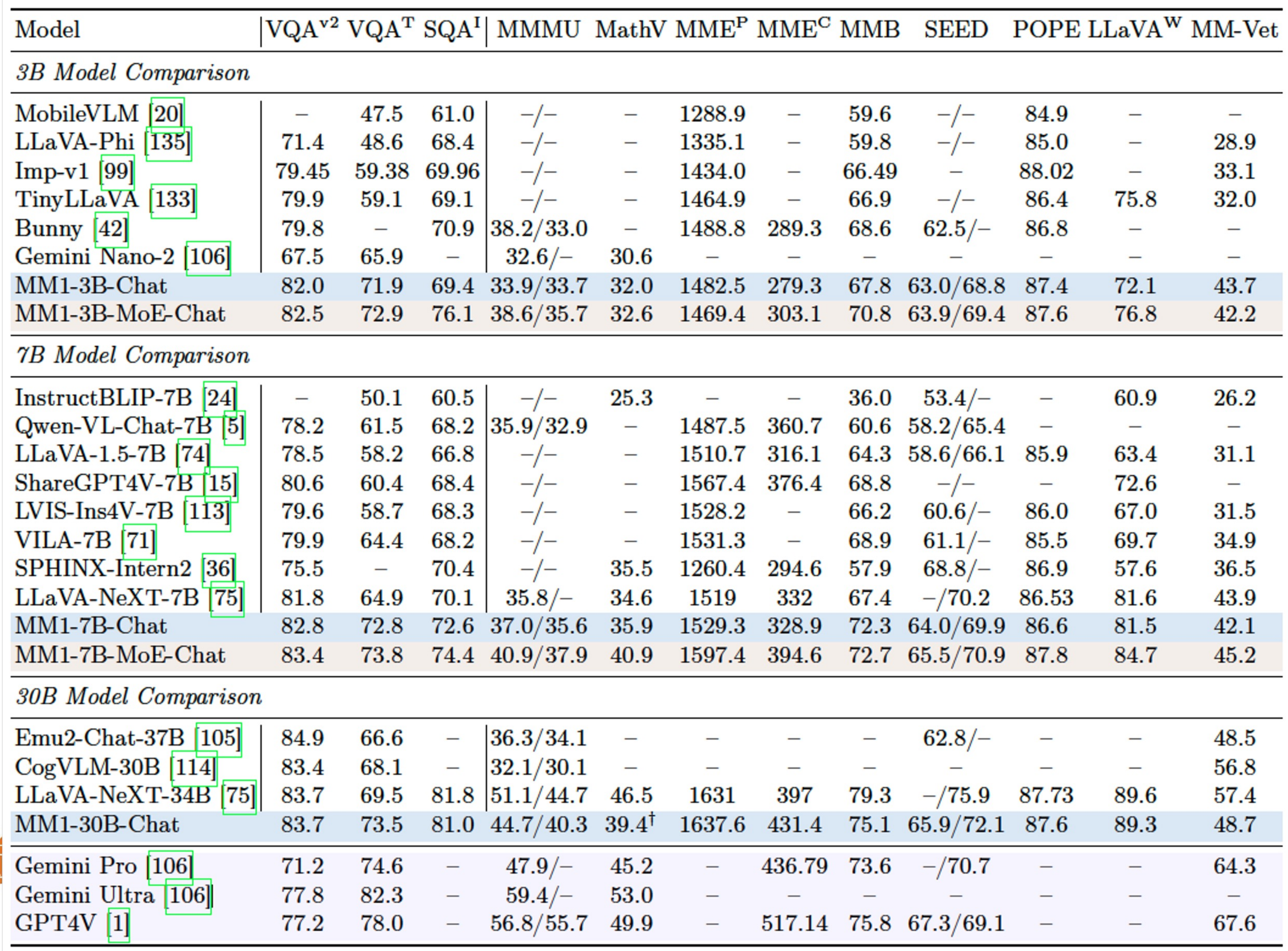
Supervised Fine-tuning

- Supervised Fine-Tuning Data Mixture:
  - Utilizes approximately 1.45M SFT examples from a diverse set of datasets.
  - Data includes instruction-response pairs generated by GPT-4, vision-language (VL) datasets for natural and text-rich images, and document/chart understanding.
  - Includes a text-only dataset for text instruction-following capabilities.
  - Datasets are formatted for instruction-following and mixed for random sampling during training.
- SFT Configuration and Evaluation:
  - Both the image encoder and the language model backbone are kept active (unfrozen) during SFT.
- Models are evaluated across 12 MLLM benchmarks

Supervised Fine-tuning (SFT)

- Scaling to Higher Image Resolutions:
  - Positional embedding interpolation is used to adapt the vision transformer backbone for higher resolutions (448x448 to 672x672 pixels).
  - Supports image resolutions up to 672x672, with a representation of 2,304 image tokens due to a patch size of 14x14.
- Sub-image Decomposition for Even Higher Resolutions:
  - For ultra-high resolutions (e.g., 1344x1344), the image is first downscaled to 672x672 for a high-level representation.
  - The same high-resolution image is also divided into four 672x672 sub-images to capture detailed visual information.
  - Positional embedding interpolation is applied to each sub-image, enabling support for resolutions as high as 1792x1792 in experiments.

SFT Results



Model	VQA^{v2}	VQA^{T}	SQA^{I}	MMMU	MathV	MME^{P}	MME^{C}	MMB	SEED	POPE	LLaVA^{W}	MM-Vet
3B Model Comparison												
MobileVLM [20]	–	47.5	61.0	–/–	–	1288.9	–	59.6	–/–	84.9	–	–
LLaVA-Phi [135]	71.4	48.6	68.4	–/–	–	1335.1	–	59.8	–/–	85.0	–	28.9
Imp-v1 [99]	79.45	59.38	69.96	–/–	–	1434.0	–	66.49	–	88.02	–	33.1
TinyLLaVA [133]	79.9	59.1	69.1	–/–	–	1464.9	–	66.9	–/–	86.4	75.8	32.0
Bunny [42]	79.8	–	70.9	38.2/33.0	–	1488.8	289.3	68.6	62.5/–	86.8	–	–
Gemini Nano-2 [106]	67.5	65.9	–	32.6/–	30.6	–	–	–	–	–	–	–
MM1-3B-Chat	82.0	71.9	69.4	33.9/33.7	32.0	1482.5	279.3	67.8	63.0/68.8	87.4	72.1	43.7
MM1-3B-MoE-Chat	82.5	72.9	76.1	38.6/35.7	32.6	1469.4	303.1	70.8	63.9/69.4	87.6	76.8	42.2
7B Model Comparison												
InstructBLIP-7B [24]	–	50.1	60.5	–/–	25.3	–	–	36.0	53.4/–	–	60.9	26.2
Qwen-VL-Chat-7B [5]	78.2	61.5	68.2	35.9/32.9	–	1487.5	360.7	60.6	58.2/65.4	–	–	–
LLaVA-1.5-7B [74]	78.5	58.2	66.8	–/–	–	1510.7	316.1	64.3	58.6/66.1	85.9	63.4	31.1
ShareGPT4V-7B [15]	80.6	60.4	68.4	–/–	–	1567.4	376.4	68.8	–/–	–	72.6	–
LVIS-Ins4V-7B [113]	79.6	58.7	68.3	–/–	–	1528.2	–	66.2	60.6/–	86.0	67.0	31.5
VILA-7B [71]	79.9	64.4	68.2	–/–	–	1531.3	–	68.9	61.1/–	85.5	69.7	34.9
SPHINX-Intern2 [36]	75.5	–	70.4	–/–	35.5	1260.4	294.6	57.9	68.8/–	86.9	57.6	36.5
LLaVA-NeXT-7B [75]	81.8	64.9	70.1	35.8/–	34.6	1519	332	67.4	–/70.2	86.53	81.6	43.9
MM1-7B-Chat	82.8	72.8	72.6	37.0/35.6	35.9	1529.3	328.9	72.3	64.0/69.9	86.6	81.5	42.1
MM1-7B-MoE-Chat	83.4	73.8	74.4	40.9/37.9	40.9	1597.4	394.6	72.7	65.5/70.9	87.8	84.7	45.2
30B Model Comparison												
Emu2-Chat-37B [105]	84.9	66.6	–	36.3/34.1	–	–	–	–	62.8/–	–	–	48.5
CogVLM-30B [114]	83.4	68.1	–	32.1/30.1	–	–	–	–	–	–	–	56.8
LLaVA-NeXT-34B [75]	83.7	69.5	81.8	51.1/44.7	46.5	1631	397	79.3	–/75.9	87.73	89.6	57.4
MM1-30B-Chat	83.7	73.5	81.0	44.7/40.3	39.4†	1637.6	431.4	75.1	65.9/72.1	87.6	89.3	48.7
Gemini Pro [106]	71.2	74.6	–	47.9/–	45.2	–	436.79	73.6	–/70.7	–	–	64.3
Gemini Ultra [106]	77.8	82.3	–	59.4/–	53.0	–	–	–	–	–	–	–
GPT4V [1]	77.2	78.0	–	56.8/55.7	49.9	–	517.14	75.8	67.3/69.1	–	–	67.6

SFT Results

- Competitive results with current SOTA
- MoE models tend to work better
- Higher image resolution and pre-training steps has a positive impact on SFT performance.
- Lessons for pre-training do transfer to SFT
  - Pre-training with caption-only data improves SFT metrics, and
  - different VL connector architectures have negligible impact on final results.

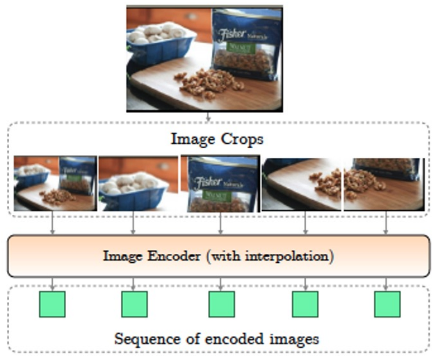
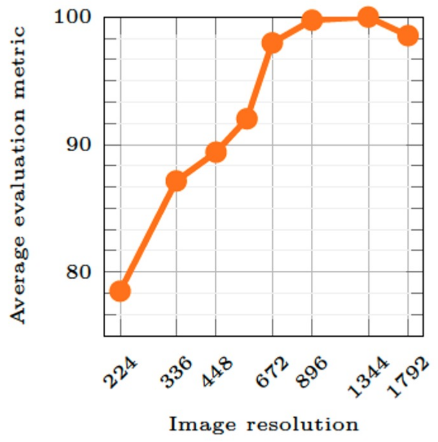
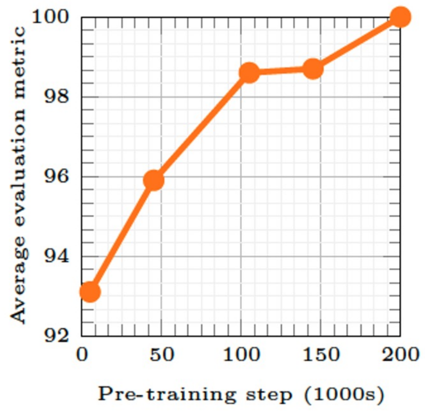


Image Crops
Image Encoder (with interpolation)
Sequence of encoded images

(a) High resolution image input processing.



(b) Impact of image resolution on SFT performance.



(c) Impact of pre-training on SFT performance.

Conclusion

- For MLLMs, authors explore the most optimal combination of:
  - Model architecture
  - Type of data
  - Training procedure
- They also create their family of MM1 models, based on the optimal combination they found. The MM1 models exhibit SOTA performance on captioning and visual question answering (VQA).
- Authors also find their optimal configuration also applies when models face SFT.

Thank you!

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science