Towards Efficient and Robust Personalized Adaptation of Large Foundation Models

Peter A. Beerel
University of Southern California

June 22, 2025

USC University of Southern California

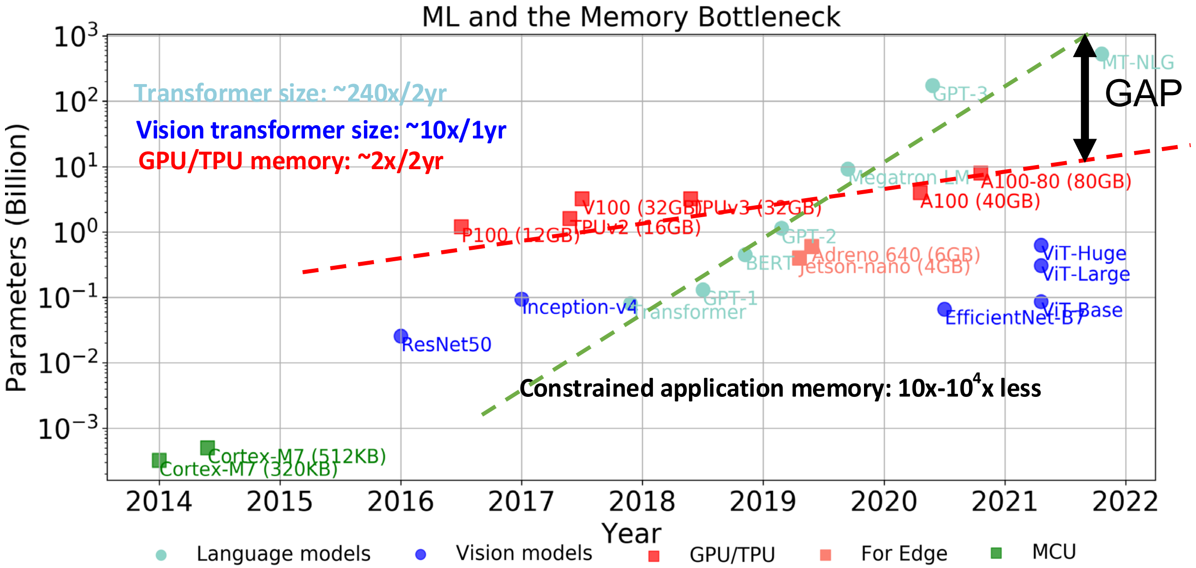
Energy Efficient Secure Sustainable Computing Group, USC

amazon

In collaboration with
Souvik Kundu, Intel Labs

intel

Motivation: Memory Wall Problem of Foundation Models



Courtesy: AIL Lab Research, Intel

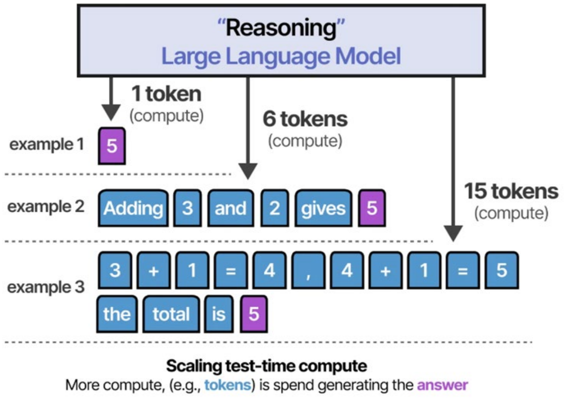
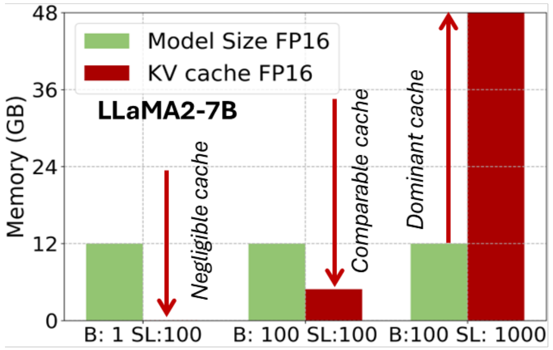


Photo Courtesy: Google Search



B: batch size , SL: Sequence length

Courtesy: AIL Lab Research, Intel

The rapid growth of LLM parameters is outpacing the increase in GPU memory capacity, and the gap makes model on-device adaptation exceedingly hard

Can we develop a memory efficient model adaptation as opposed to parameter efficiency of LORA?

The emergence of reasoning models and long-sequence processing, memory bottleneck intensifies for the traditional transformer-based models

Can we have the performance of transformers while leveraging the benefits of linear attention?

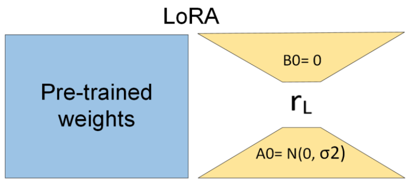
Can we yield efficient models that are more robust to security issues like hallucination?

Outline: Towards a Comprehensive Solution for Personalized LLMs

3 key considerations

Personalized: Demand for adaptive automation

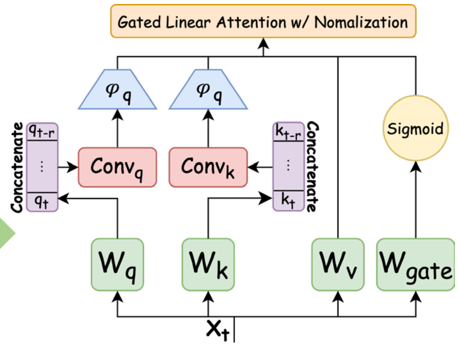
Efficient fine-tuning



Memory ~~Parameter~~ efficient fine-tuning

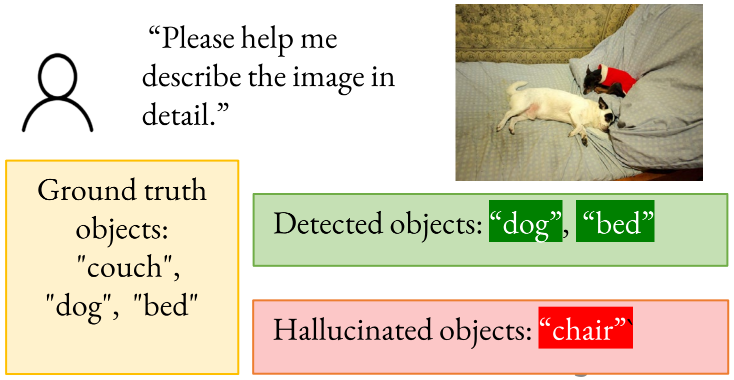
Efficient: For democratized and sustainable deployment

Towards linear-attention



Reliable: For sensitive generative tasks

Improving hallucination



Adaptive Freezing of Low Rank Adaptation in Parameter Efficient Fine-Tuning of Large Models






Zeyu Liu*
University of Southern California, USA

Souvik Kundu*
Intel Labs, San Diego, USA

Anni Li
University of Southern California, USA

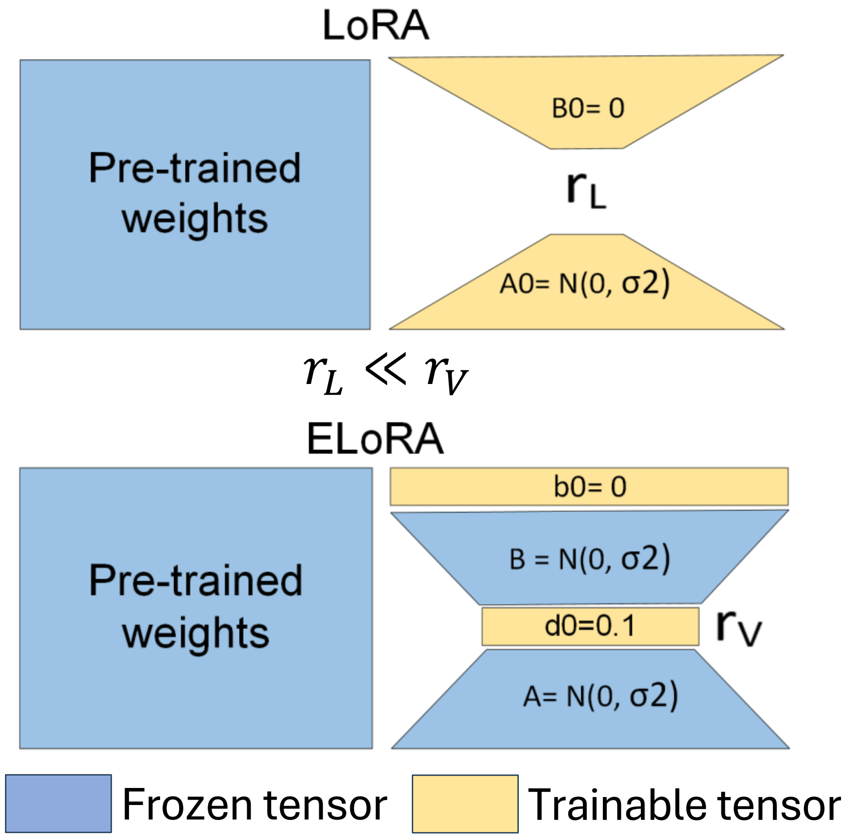
Junrui Wan
University of Southern California, USA

Lianghao Jiang
University of Southern California, USA

Peter A. Beerel
University of Southern California, USA

GitHub link

Limitations of State-of-the-Art Adapter Based Approaches



- ❑ Despite reduced parameter count w.r.t full fine-tuning (FFT), LoRA [1] has a **large trainable parameter count**!
- ❑ ELoRA [2] reduces trainable parameter count by freezing LoRA projection matrices (PM) and using only two trainable vectors!

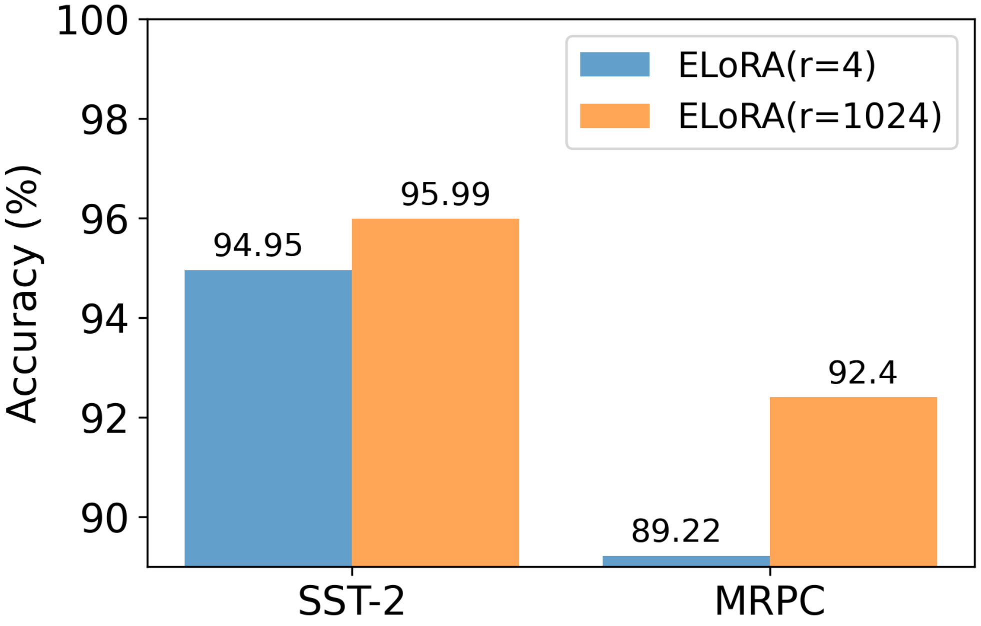
ELoRA [2] seems to achieve comparable performance with extremely small number of trainable parameters without any sacrifice.

[1] E. J. Hu, et al. "LoRA: Low-Rank Adaptation of Large Language Models", ICLR 2022
[2] D. J. Kopiczko, et al. "ELoRA: Efficient Low-Rank Adaptation with Random Matrices", ICLR 2024

Limitations of State-of-the-Art Adapter Based Approaches

However,

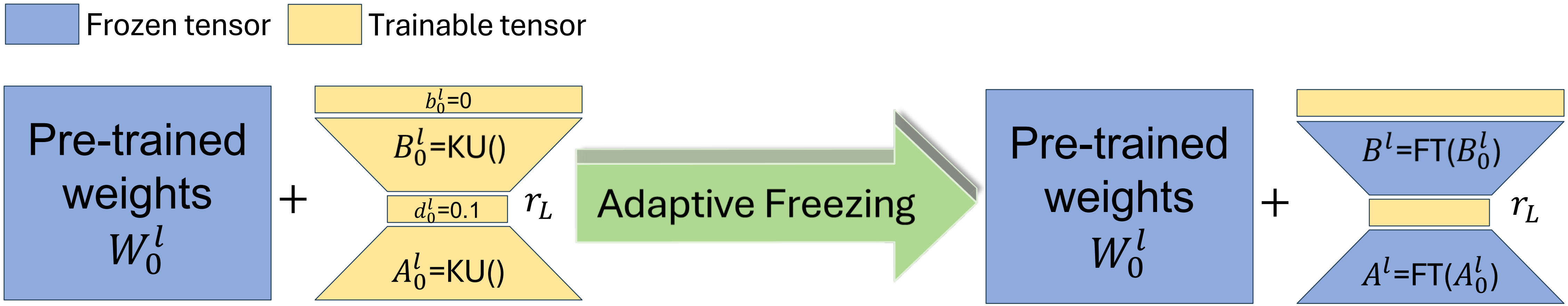
- ❑ ELoRA significantly increases FLOP count for fine-tuning
- ❑ ELoRA with lower rank leads to performance degradation



- ❑ We measured the accuracy of ELoRA with rank (r) of 4 and 1024 on two datasets.
- ❑ Model with r=4 yields poorer performance

This highlights the accuracy sacrifice in making the PMs untrainable when rank is low

Our Approach: Adaptive Freezing of LoRA (AFLoRA)



- ❑ AFLoRA starts fine-tuning with trainable PMs and vectors. It adaptively freezes the PMs (but always updates the vectors)!
- ❑ AFLoRA has similar rank as that with LoRA yielding reduced FLOPs!
- ❑ AFLoRA has similar effective trainable parameters as that with ELoRA!
- ❑ Adaptive freezing of AFLoRA potentially avoids overfitting over small dataset!

[3] Q. Zhang, et al. “PLATON: Pruning Large Transformer Models with Upper Confidence Bound of Weight Importance”, ICML 2022

Our Approach: Adaptive Freezing of LoRA (AFLoRA)

Frozen tensor Trainable tensor

$$\Lambda_b^l = \mathrm{diag}(b^l) \quad \Lambda_d^l = \mathrm{diag}(d^l)$$

Trainable Projection Matrices

$$Y = W_0^l X + \underbrace{\Lambda_b^l}\boxed{B^l}\underbrace{\Lambda_d^l}\boxed{A^l}X$$

Trainable Vectors

Frozen Projection Matrices

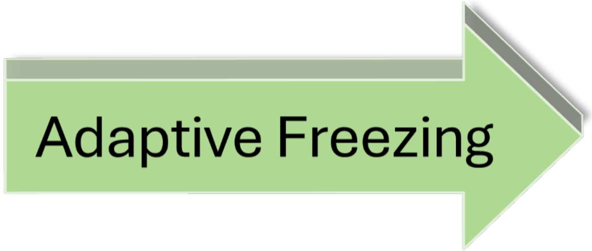
$$Y = W_0^l X + \underbrace{\Lambda_b^l}\boxed{B^l}\underbrace{\Lambda_d^l}\boxed{A^l}X$$

Trainable Vectors

[3] Q. Zhang, et al. "PLATON: Pruning Large Transformer Models with Upper Confidence Bound of Weight Importance", ICML 2022

Our Approach: Adaptive Freezing of LoRA (AFLoRA)

Frozen tensor | Trainable tensor



We calculate the freezing score as

$$I_{A^l} = |\nabla \mathcal{L}(\theta)|, \bar{I}_{A^l}^{(t)} = \beta_1 \bar{I}_{A^l}^{(t-1)} + (1-\beta_1) I_{A^l}^{(t)}$$

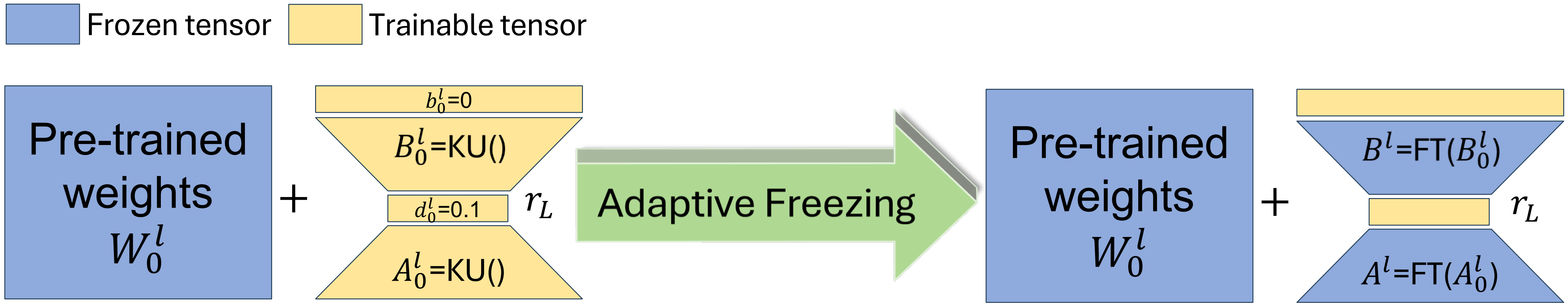
$$U_{A^l}^{(t)} = \left| I_{A^l}^{(t)} - \bar{I}_{A^l}^{(t)} \right|, \bar{U}_{A^l}^{(t)} = \beta_2 \bar{U}_{A^l}^{(t-1)} + (1-\beta_2) U_{A^l}^{(t)}$$

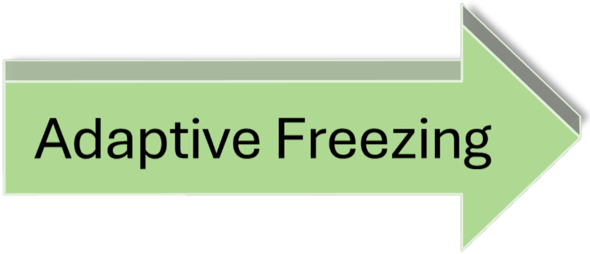
$$s_{A^l}^{(t)} = mean(\bar{I}_{A^l}^{(t)} \circ \bar{U}_{A^l}^{(t)})$$

Freezing score at iteration *t*

[3] Q. Zhang, et al. “PLATON: Pruning Large Transformer Models with Upper Confidence Bound of Weight Importance”, ICML 2022

Our Approach: Adaptive Freezing of LoRA (AFLoRA)





- At step *t*, we freeze the lowest *k*% of PMs using freezing score
  - The *k* is calculated from the cubic scheduling [3]
  - We set a hyper-parameter t_f to ensure all PMs freeze after $T - t_f$ where T is the number of total iteration

[3] Q. Zhang, et al. "PLATON: Pruning Large Transformer Models with Upper Confidence Bound of Weight Importance", ICML 2022

Experimental Results: Comparision with the SoTA

Results with DeBERTaV3* on GLUE benchmark

Method	#Params. ↓	CoLA ↑	SST-2 ↑	MRPC ↑	QNLI ↑	STS-B ↑	RTE ↑	MNLI ↑	QQP ↑	Avg. ↑
FFT	184M	69.21	95.64	89.22	93.78	91.59	82.49	89.98/89.95	92.05/89.31	87.82
LoRA (r = 8)	1.33M	69.73	95.57	89.71	93.76	**91.86**	85.32	**90.47/90.46**	91.95/89.26	88.38
AdaLoRA	1.27M	70.86	95.95	90.22	94.28	91.39	87.36	90.27/90.30	**92.13**/88.41	88.83
SoRA (r = 4)	0.47M	71.05	95.57	90.20	93.92	91.76	86.04	90.38/90.43	92.06/**89.44**	88.71
ELoRA*	0.16M	70.74	95.18	90.93	93.58	91.08	87.36	90.11/90.22	90.69/87.63	88.53
AFLoRA (r = 4)	**0.14M****	**72.01**	**96.22**	**91.91**	**94.42**	91.84	**88.09**	89.88/90.17	90.81/87.77	**89.23**

*We only apply AFLoRA to the PMs in the FFN and freeze the PMs in attention layers.

Results with LLaMA-7B** on complex reasoning task

Method	Model	Low-rank val.	# Params.	GSM8k Acc (%)
LoRA	LLaMA-7B	32	56.1M	37.50
AFLoRA (Ours)	LLaMA-7B	32	**17.8M**	**38.59**

**We apply AFLoRA on all the PMs in the models.

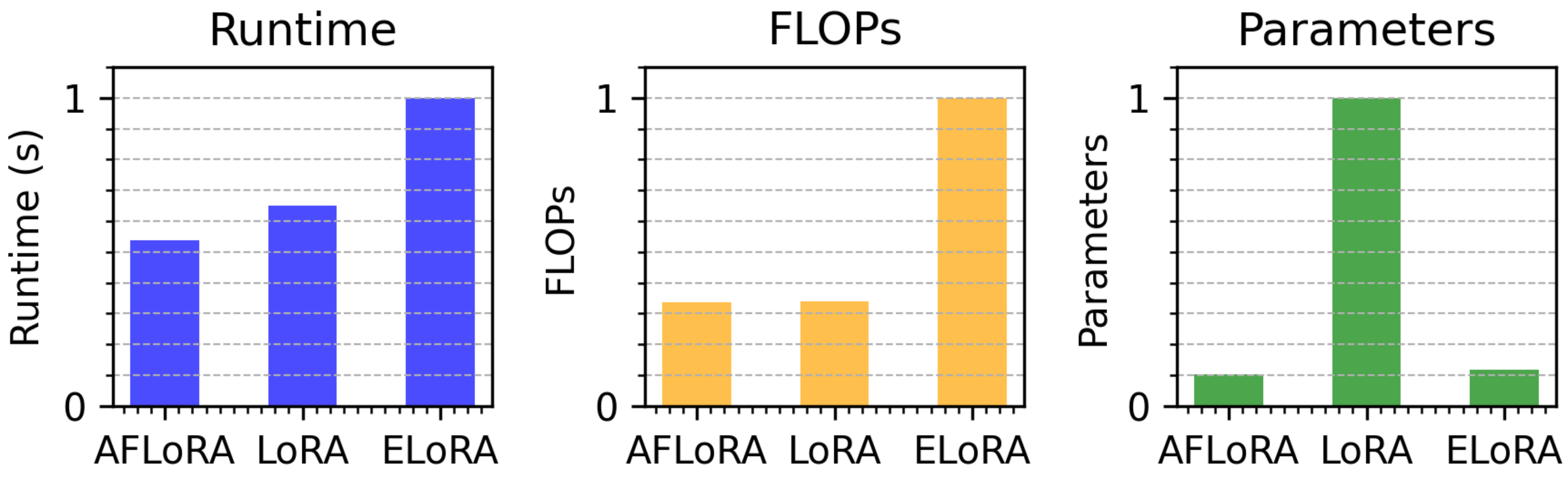
Results with BART-Large** on summarization task

Method	Model	Low-rank val.	# Params.	CNN/DailyMail (R1/R2)
LoRA	BART-Large	16	8.65M	43.96/21.06
AFLoRA (Ours)	BART-Large	16	**5.10M**	**44.31/21.32**

- ❑ AFLoRA fine-tuned models yields ***higher performance*** compared to the alternatives!
- ❑ AFLoRA fine-tuning requires up to ***9.5x*** fewer average params. than the SoTA

Experimental Results: Efficiency Evaluation

Results with DeBERTaV3* on GLUE benchmark: System Metrics



*We only apply AFLoRA to the PMs in the FFN and freeze the PMs in the attention layers.

- ❑ AFLoRA yields up to ***1.86x*** runtime improvement!
- ❑ AFLoRA yields up to ***2.96x*** reduced FLOP!

Summary

- ❑ Parameter Efficient Fine Tuning (PEFT) is an important approach towards extending the applicability of large language models to a variety of datasets
- ❑ AFLoRA's adaptive freezing the PMs provides the benefits of trainable PMs with up to 9.5x fewer average trainable parameters than the SOTA
- ❑ Results on a variety of language models with different tasks demonstrate performance, FLOPs, and run-time advantages

LAWCAT: Efficient Distillation from Quadratic to Linear Attention with Convolution across Tokens for Long Context Modeling

Zeyu Liu
University of Southern California, USA

Souvik Kundu
Intel Labs, San Diego, USA

Lianghao Jiang
University of Southern California, USA

Anni Li
University of Southern California, USA

Srikanth Ronanki
Amazon AGI, USA

Sravan Bodapati
Amazon AGI, USA

Gourav Datta
Case Western Reserve University, USA

Peter A. Beerel
University of Southern California, USA

GitHub link

Background - Self Attention

Most open-source LLMs employ standard softmax-based self-attention as the core token-mixing mechanism.

Meta/Llama

Google/Gemma

Mistral AI

Tongyi/Qwen

Amazon/Nova

This architecture has achieved remarkable success across a wide range of tasks, including NLP, CV, and speech understanding.

However, the ***quadratic computational complexity*** with respect to sequence length remains a significant bottleneck, particularly for deployment on edge devices and in ***long-context scenarios***.

Background - Recurrent Model

Modern recurrent models can be broadly categorized into state space models and linear attention models

Mamba

RWKV

Flash linear attention

Models can achieve **O(N)** complexity by avoiding explicit pairwise token interactions by propagating compressed memory states across time.

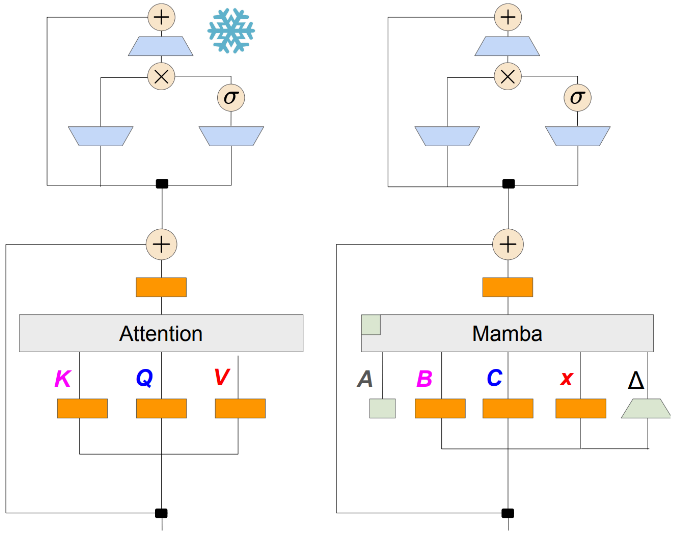
However, pre-training a large-scale recurrent model still demands significant computation resource which hinder widespread adoption.

[1] https://tridao.me/blog/2024/mamba2-part1-model/
[2] https://wiki.rwkv.com/
[3] https://github.com/fla-org/flash-linear-attention

Background - Linearization via Distillation

Recently, some research propose to **convert** existing quadratic transform LLMs into linear complexity LLMs

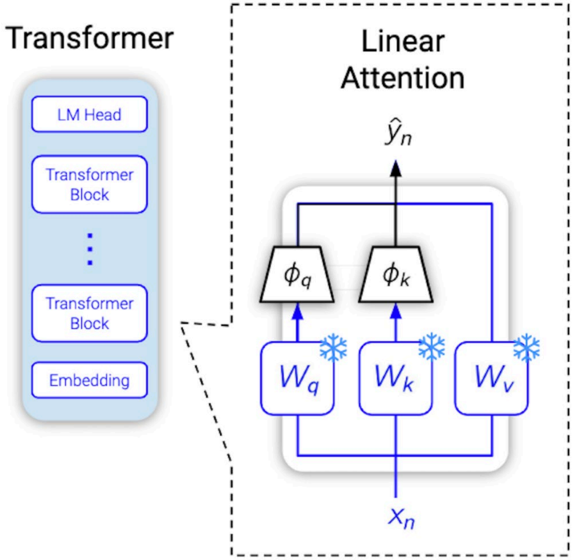
Is there a way to efficiently maintain or even extend the model's context length?



LlamaInMamba

Require ~20% of the original pre-training tokens

Retains half of attention layers



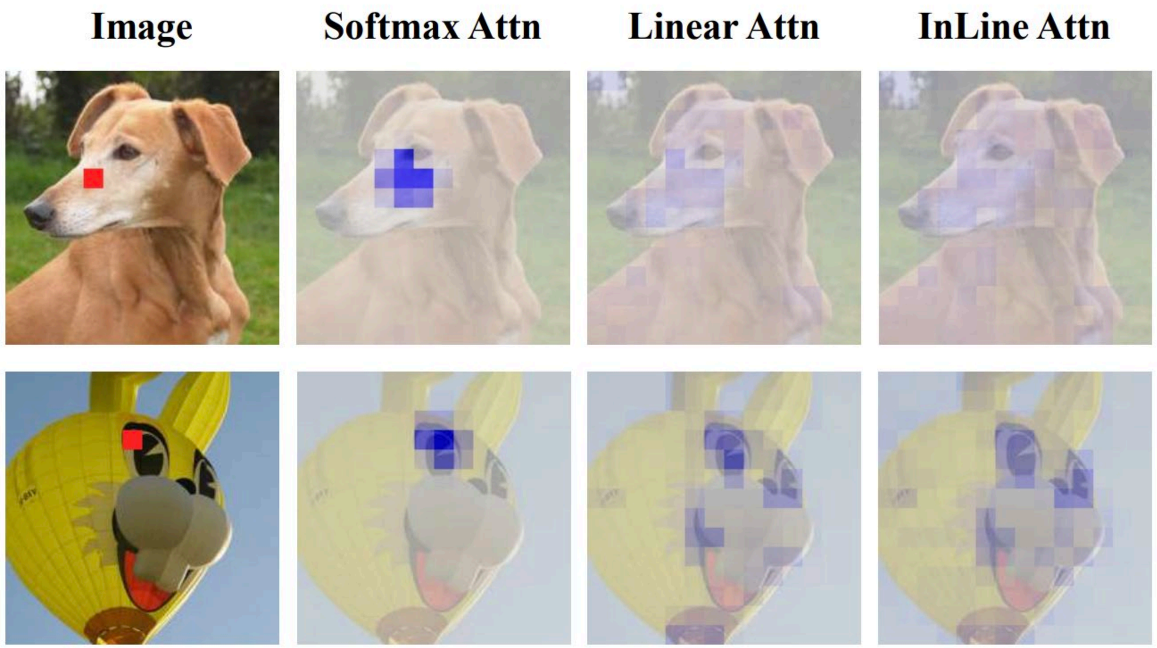
LoLCATs

Perform well only on the tasks with sequence lengths similar to the training data.

[1] Junxiong Wang, et al. The Mamba in the Llama: Distilling and Accelerating Hybrid Models. [2] Michael Zhang, et al. LoLCATs: On Low-Rank Linearizing of Large Language Models

Motivation - The Local Modeling in Self-Attention

Attention mechanism is famous for its large receptive field and outstanding long-range modeling capability. But, [1] find that effective local modeling is crucial for its effectiveness.



Softmax attention exhibits strong local bias.

Linear and InLine Attn. yield meaningful Attn. dist., but focus more on global modeling.

Increasing local bias may enhance the expressive power of linear attention

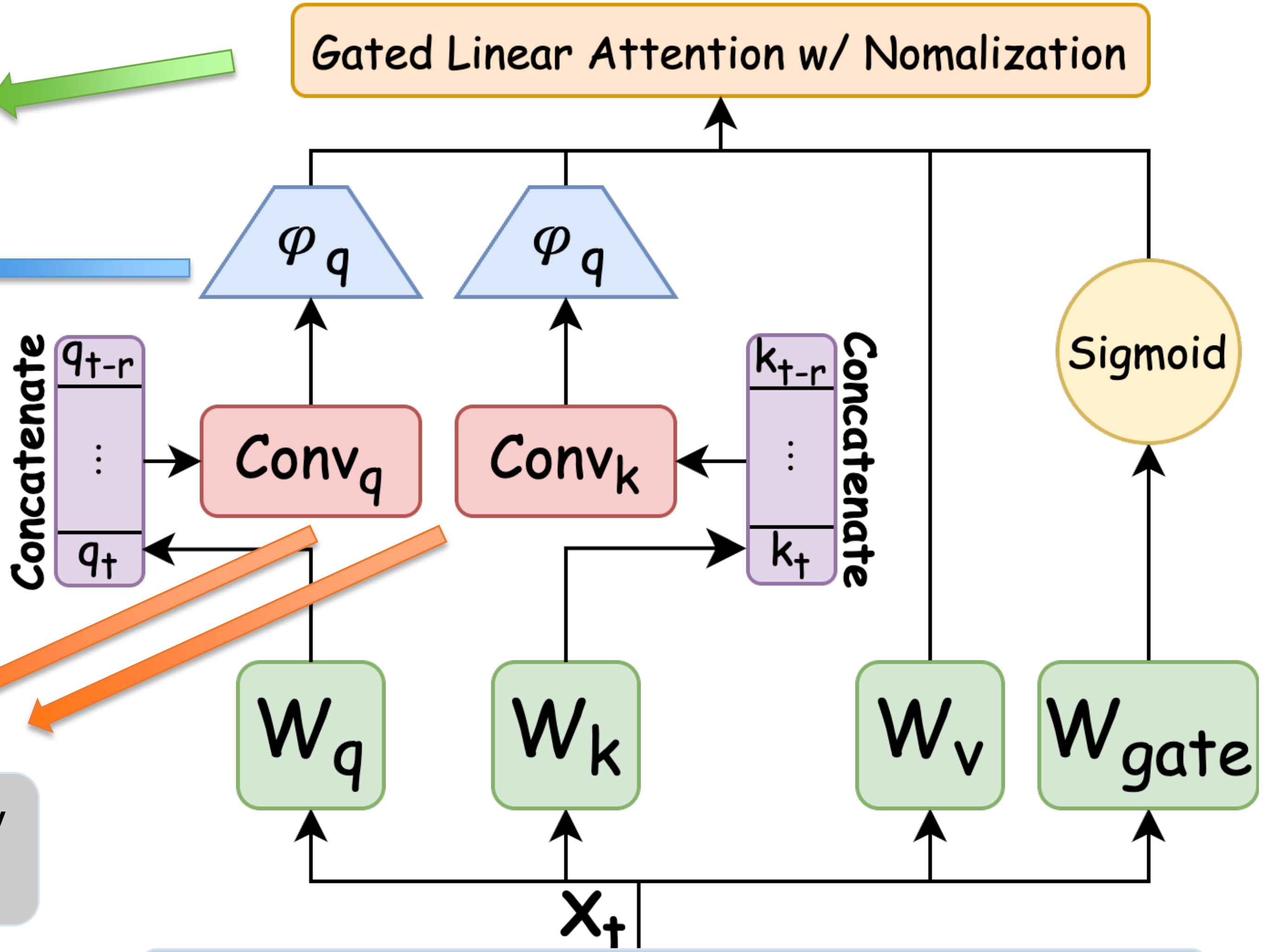
[1] Dongchen Han, et al. Bridging the Divide: Reconsidering Softmax and Linear Attention

LAWCAT - Linear Attention with Convolution across Tokens

Special normalization

Shared linear projection

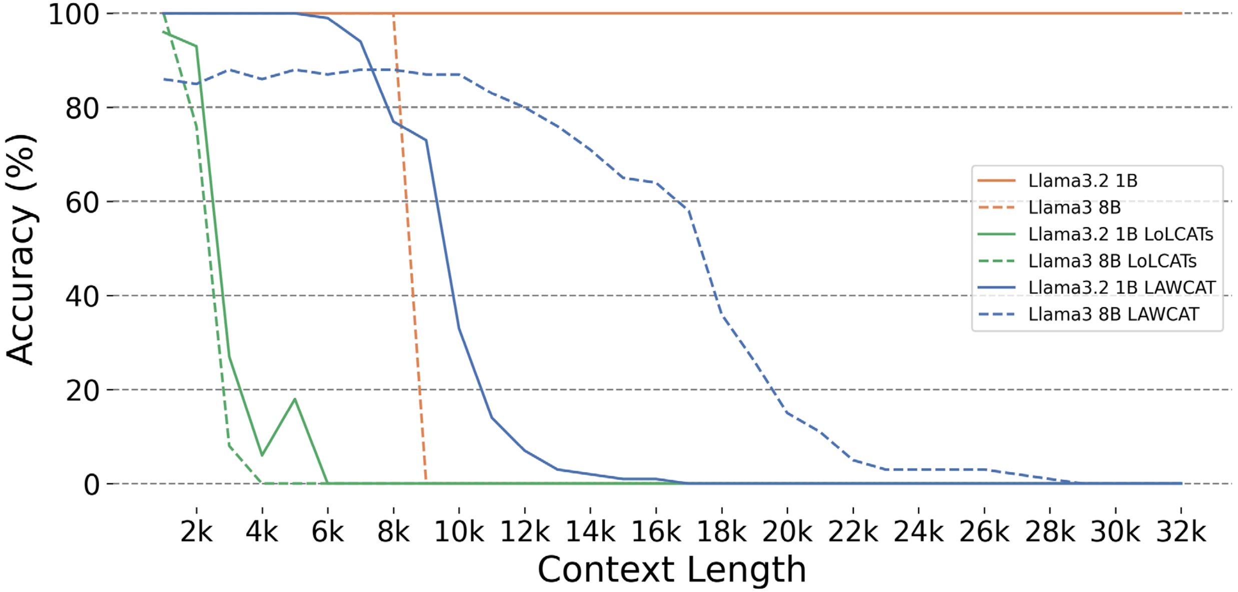
Causal depthwise Conv 1D layer across token

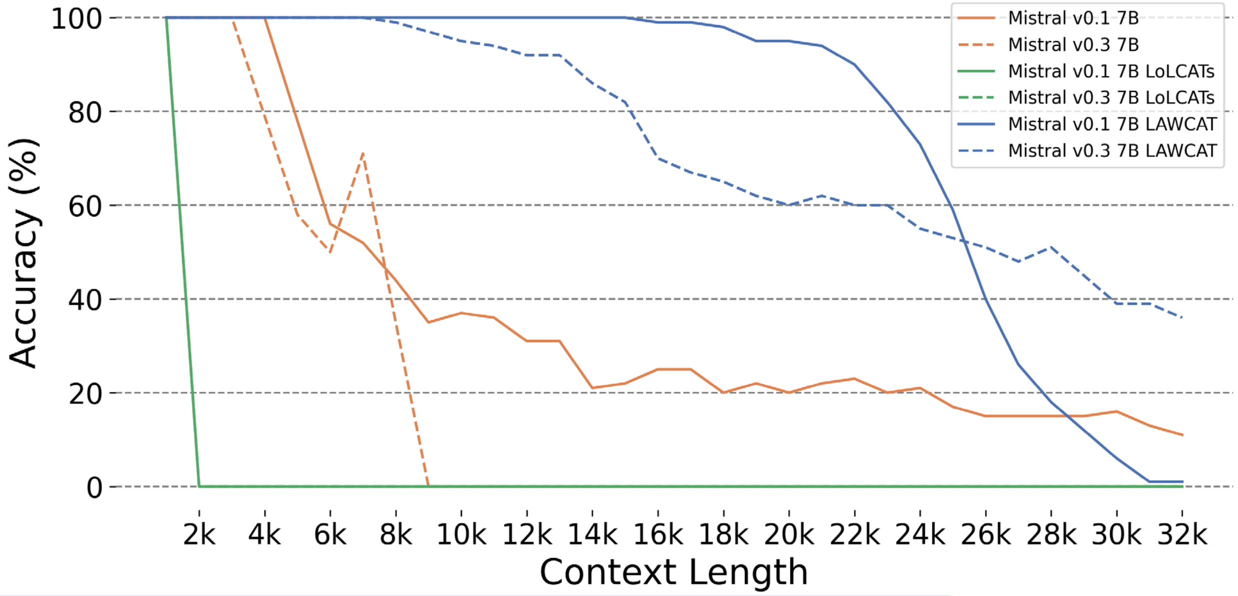


removes RoPE and sliding window attention.

Results - Passkey Retrieval

Accuracy on passkey retrieval from 1k to 32 k (distill and fine-tune on 1k-length dataset)





For Llama3 8B, our LAWCAT can extend the effective context length from 8k to ~11k.
For Llama3.2 1B, LAWCAT also can preserve the performance until 9k.

For Mistral v0.1 7B and Mistral v0.3 7B models, our LAWCAT can extend the effective context length to ~23k/15k respectively.

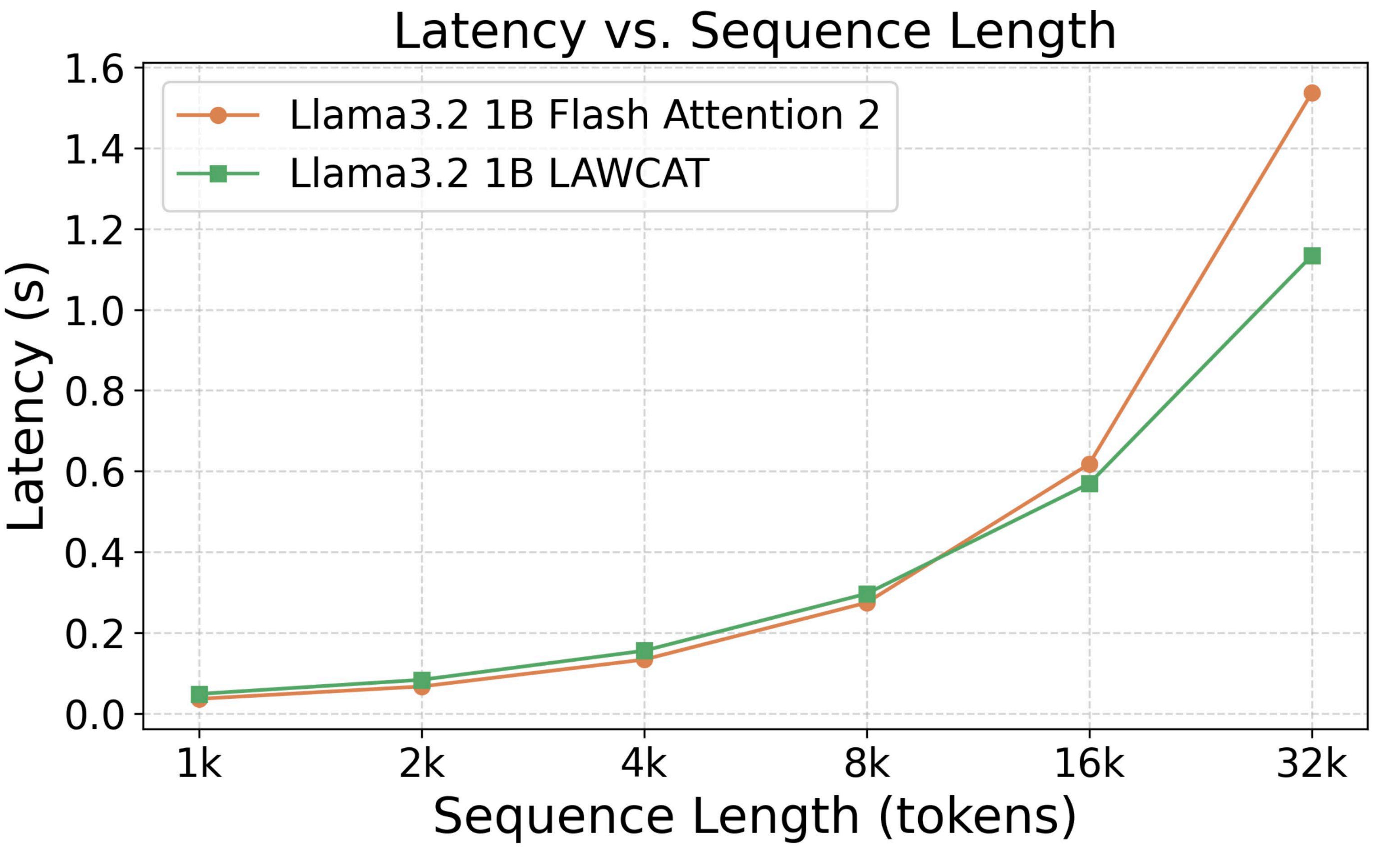
Results - S-NIAH 1&2&3(distill and fine-tune on 1k-length)

	S-NIAH-1 (pass-key retrieval)				S-NIAH-2 (number in haystack)				S-NIAH-3 (uuid in haystack)		
	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K
Pre-trained Model											
DeltaNet-1.3B	97.4	96.8	99.0	**98.8**	98.4	45.6	18.6	14.4	85.2	47.0	22.4
Mamba2-1.3B	99.2	98.8	65.4	30.4	99.4	98.8	56.2	17.0	64.4	47.6	4.6
Gated DeltaNet-1.3B	98.4	88.4	91.4	91.8	100.0	**99.8**	**92.2**	29.6	**86.6**	84.2	27.6
Distilled Model	Pre-trained Model: Llama3.2-1B										
LoLCATs	100	84	0	0	84	44	0	0	72	24	0
Ours	**100**	**100**	**100**	80	**100**	96	88	**48**	56	44	24
	Pre-trained Model: Llama3-8B										
LoLCATs	100	4	0	0	92	4	0	0	84	24	0
Ours	100	100	96	80	100	92	84	32	80	**88**	**60**

Even if we only distill&ft the model on 1k-length dataset with S-NIAH-3 format, our LAWCAT models can generalize the performance to other tasks

Our LAWCAT demonstrates superior robustness to increasing input lengths, with notably smaller performance drops than other SoTAs

Results – Latency Benchmark (Prefill stage)



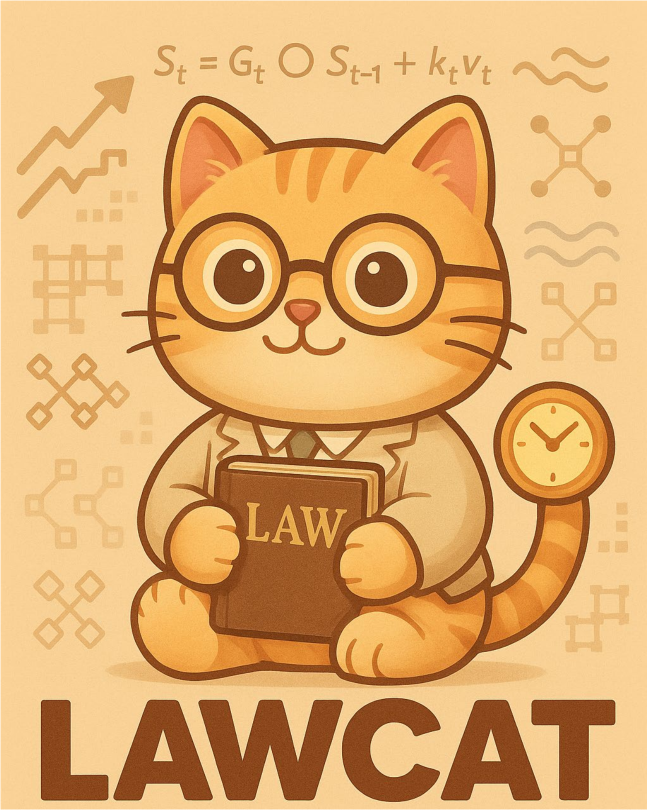
From 1k to 8k sequence length, the Transformer model exhibits slightly lower latency than LAWCAT.

However, beyond 8k, LAWCAT becomes increasingly more efficient, with the latency gap widening as the input length grows.

Summary

- We propose a novel distillation framework, LAWCAT, which leverage Conv across time and GLA with normalization to approximate the transformers.

- Our LAWCAT framework enables efficient linearization while maintaining competitive performance and exhibits strong robustness to long context.



- Future work may focus on more effective integration with sliding window attention, which holds promise for further enhancing performance.

Hallucinations in Vision-Language Models through Image-Guided Head Suppression

Sreetama Sarkar*
University of Southern California, USA

Yue Che*
University of Southern California, USA

Alex Gavin
Harvard-Westlake High School, USA

Peter A. Beerel
University of Southern California, USA

Souvik Kundu
Intel Labs, San Diego, USA

*Equally cobtributing authors

Hallucinations in VLMs

Image Captioning

"Please help me describe the image in detail."

Ground truth objects: "couch", "dog", "bed"

Detected objects: "dog", "bed"

Hallucinated objects: "chair"`

LLaVa-1.5: "The image features two dogs lying on a bed, one on the left side and the other on the right side. They appear to be relaxed and comfortable, possibly sleeping or resting. The bed is covered with a blue blanket, providing a cozy environment for the dogs.\n\nIn addition to the dogs, there is a chair located in the background, positioned behind the bed. The scene captures a peaceful and restful moment shared between the two dogs."

Visual Q/A

"Is there a spoon in the image?"

LLaVa-1.5: "No, there is no spoon in the image. The man is eating a donut, not using a spoon."

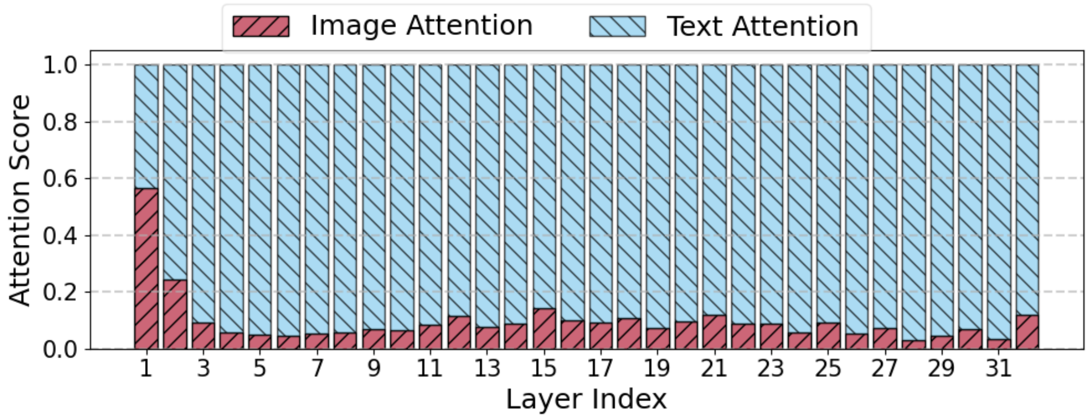
Why Do Vision-Language Models Hallucinate?

Prompt: "A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. USER: <image> Please help me describe the image in detail. ASSISTANT:"

49 text tokens

576 image tokens

- Image tokens receive <10% of total attention from layer 3, while constituting ~76-92% of the input
- This causes the model to ignore the context provided by image (taken as the "fact"), potentially leading to hallucinations



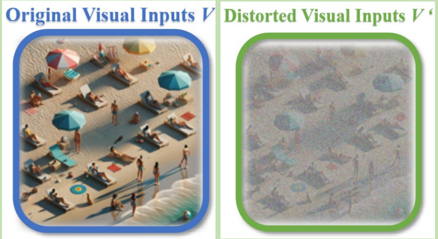
Existing Methods

Training-based Methods

- ❑ Factually-Augmented RLHF: Reinforcement Learning from Human Feedback
- ❑ FGAIF: Reinforcement Learning using fine-grained AI feedback
- ❑ LACING: separate attention streams for visual and textual inputs

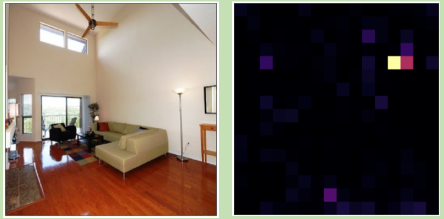
Training-free Methods

- ❑ VCD: contrastive decoding between original and distorted visual inputs



Courtesy: VCD

- ❑ PAI: contrastive decoding with and without image priors and increased attention to image tokes
- ❑ DAMRO: contrastive deco-ding between entire image and only outlier visual tokens



Courtesy: DAMRO

Training-based Methods require **large-scale resources** (e.g., 8× A100 GPUs with 40GB memory each)

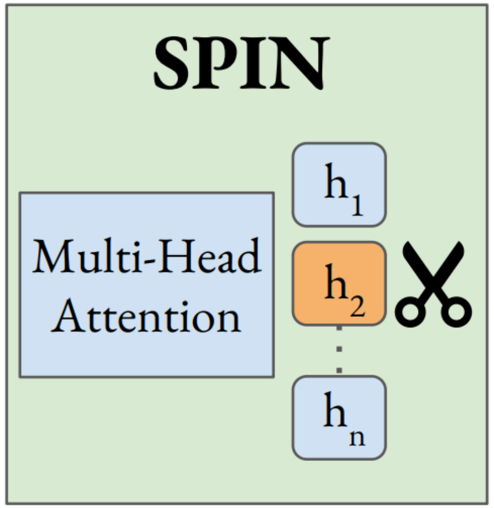
Training-free methods use contrastive decoding for hallucination reduction incurring **additional latency overhead** (~2X)!!

SPIN: SuPpressing image INattentive heads (Our Approach)

- We observe that hallucination in LVLMs often stems from specific attention heads exhibiting insufficient attention to visual input, which we refer as ***"image-inattentive heads"***

- We present **SPIN**, **SuP**pressing image **IN**attentive heads, a ***training-free*** method to reduce hallucinations incurring ***no computational or latency overhead***

- SPIN dynamically suppresses attention heads for each input token, reducing the attention imbalance and enhancing model performance



SPIN Multi-head Attention

- To suppress problematic attention heads, we introduce *masked multi-head attention (MHA)*

- We use a dynamic mask m_i for each attention head i, where m_i is obtained based on the attention of the current text query token q_i to key vision tokens
  - $m_i = 1$: the head is kept intact
  - $m_i = \alpha$: head suppressed using suppression factor α
  - $\alpha = 0$: the head is completely suppressed or pruned

$$\mathrm{A}_v = \mathrm{A}_{tot}[I_{\text{start}} : I_{\text{end}}], \quad \mathrm{A}_{tot} = q^i K^{iT}$$

$$m_i = \begin{cases} 1 & \text{if } i \in \text{top-K}(\sum_{j=1}^{N_v} A_v[j]) \\ \alpha & \text{otherwise} \end{cases}$$

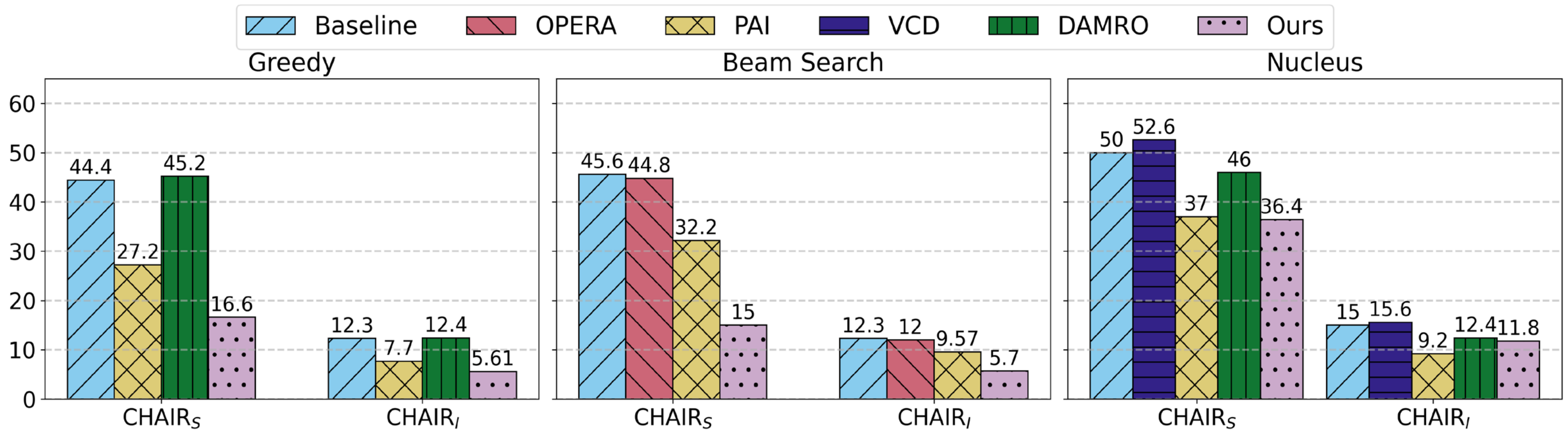
$$\mathrm{MHA}_{Q,K,V,m} = \left(\bigoplus_{i=1}^{H} (m_i \cdot h_i) \right) W_o$$

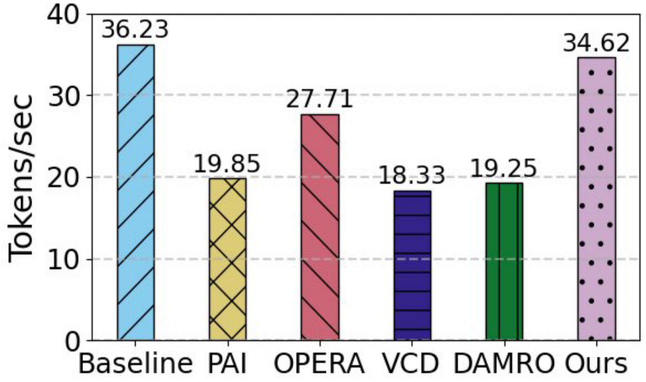
CHAIR Evaluation for Image Captioning

Model	Method	Layers	r	α	C_S	C_I	F1
LLaVA-1.5(7B)	Baseline	-	-	-	44.4	12.3	**77.8**
	PAI	-	-	-	27.2	7.7	76.8
	DAMRO	-	-	-	45.2	12.4	77.8
	SPIN	1∼32	0.05	0.08	26.4	7.6	77.6
	SPIN	1∼32	0.05	0.01	**16.6**	**5.6**	74.6
LLaVA-1.5(13B)	Baseline	-	-	-	41.4	10.9	78.9
	PAI	-	-	-	37.4	9.2	79.2
	DAMRO	-	-	-	41.2	11.0	78.7
	SPIN	1∼16	0.10	0.0	30.6	8.3	**79.6**
	SPIN	1∼20	0.10	0.0	**29.2**	**7.9**	79.1
Minigpt4	Baseline	-	-	-	31.4	11.1	**70.6**
	PAI	-	-	-	19.8	8.4	69.7
	SPIN	1∼16	0.18	0.0	21.0	**6.2**	68.8
	SPIN	1∼16	0.18	0.05	**17.6**	8.4	68.4
Qwen-VL	Baseline	-	-	-	46.2	11.7	76.5
	PAI	-	-	-	45.0	11.3	76.6
	SPIN	1∼20	0.30	0.08	29.6	8.5	**78.0**
	SPIN	1∼20	0.30	0.001	**24.8**	**7.8**	76.7

- ❑ A lower C_s/C_I score indicates fewer hallucinated objects, while a higher F1 indicates higher caption accuracy
- ❑ Problematic heads **typically lie in the first 16 to 20 layers**, but are uniformly distributed across all layers for LLaVA-7B
- ❑ **r ranges from 5%** (LLaVA-7B) **to 30%** (Qwen-VL), indicating specialized heads in LLaVA vs. distributed head functionality in Qwen-VL

CHAIR Evaluation for Different Decoding Methods





- While SPIN *can be applied irrespective of the decoding strategy*, it is more effective for greedy and beam-search decoding, due to the variability in token selection for nucleus sampling
- SPIN reduces hallucination scores up to **3x over baseline and existing methods**
- SPIN achieves the **highest throughput** among SoTA approaches, while achieving throughput performance close to baseline

Qualitative Comparison of Image Captions

"Please help me describe the image in detail."

LLaVa-1.5: The image features two dogs lying on a bed, one on the left side and the other on the right side. They appear to be relaxed and comfortable, possibly sleeping or resting. The bed is covered with a blue blanket, providing a cozy environment for the dogs. In addition to the dogs, there is a chair located in the background, positioned behind the bed. The scene captures a peaceful and restful moment shared between the two dogs.

SPIN: The image features two dogs lying on a bed, with one dog being a white dog and the other being a black dog. The white dog is lying on the left side of the bed, while the black dog is lying on the right side. Both dogs appear to be sleeping or resting, possibly enjoying each other's company. The bed is covered with a blanket, providing a comfortable surface for the dogs to lay on.

"Is there a spoon in the image?"

LLaVa-1.5: "No, there is no spoon in the image. The man is eating a donut, not using a spoon."

SPIN: "Yes, there is a spoon in the image."

Summary and Conclusions

- Adaptively freezing LoRA projection matrices can reduce computation and overfitting when fine-tuning the LLMs for downstream tasks
- Using the LAWCAT framework, we can efficiently distill pre-trained transformer-based models into linear attention model while maintaining long context ability and less latency, making it well-suited for edge deployment
- Reliable deployment of vision foundation models can be achieved at no additional cost through identification and suppression of problematic heads