中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

智能算法安全全国重点实验室
State Key Laboratory of AI Safety

Layer-wise Fine-tuning in LLMs

Wanli Yang

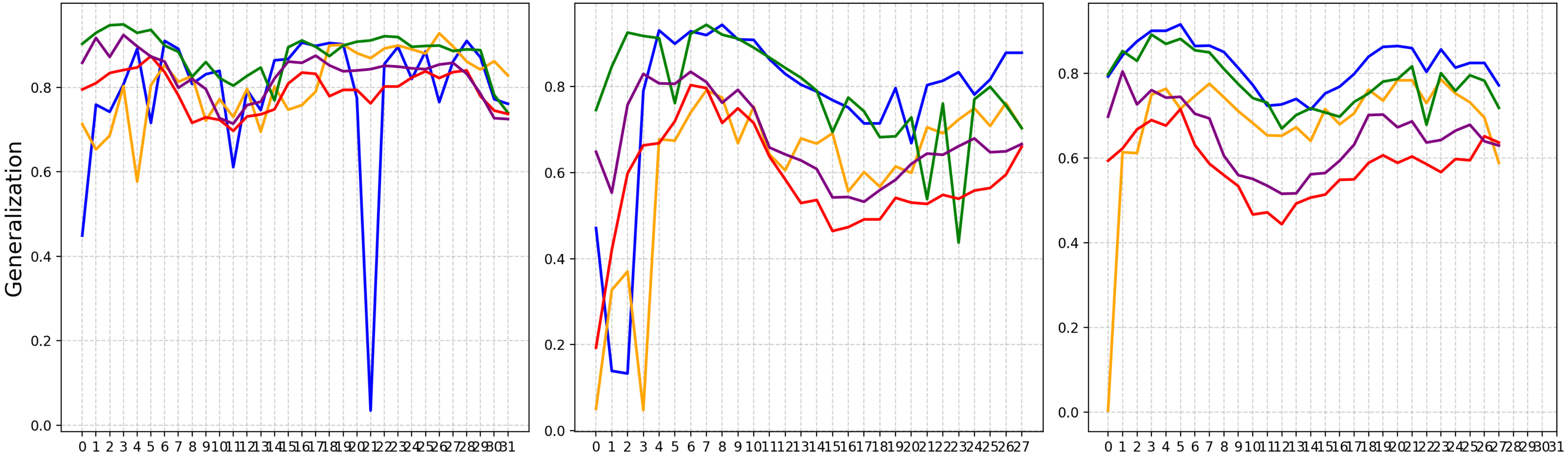
July 18, 2025

STAR Group Paper Reading

Table of Contents

Background

- Model editing pursue localized update of LLMs, i.e., single MLP
- Our work demonstrates localized fine-tuning is effective for editing
- How can we identify the optimal tuning locations?
- Existing strategy: investigate all layers and modules



More Efficient Approaches?

- LISA: Layerwise Importance Sampling for Memory-Efficient Large Language Model Fine-Tuning (NIPS 2024)
- Understanding Layer Significance in LLM Alignment (ArXiv 2024)
- Layer-wise Importance Matters: Less Memory for Better Performance in Parameter-efficient Fine-tuning of Large Language Models (EMNLP 2024)

Table of Contents

Paper Information

LISA: Layerwise Importance Sampling for Memory-Efficient Large Language Model Fine-Tuning

Rui Pan♠*, **Xiang Liu**♣*, **Shizhe Diao**♦, **Renjie Pi**♡, **Jipeng Zhang**♡,
Chi Han♠, **Tong Zhang**♠

♠University of Illinois Urbana-Champaign
♣The Hong Kong University of Science and Technology(Guangzhou)
♦NVIDIA ♡The Hong Kong University of Science and Technology
{ruip4, chihan3, tozhang}@illinois.edu
xliu886@connect.hkust-gz.edu.cn {sdiaoaa, rpi, jzhanggr}@ust.hk

Motivation

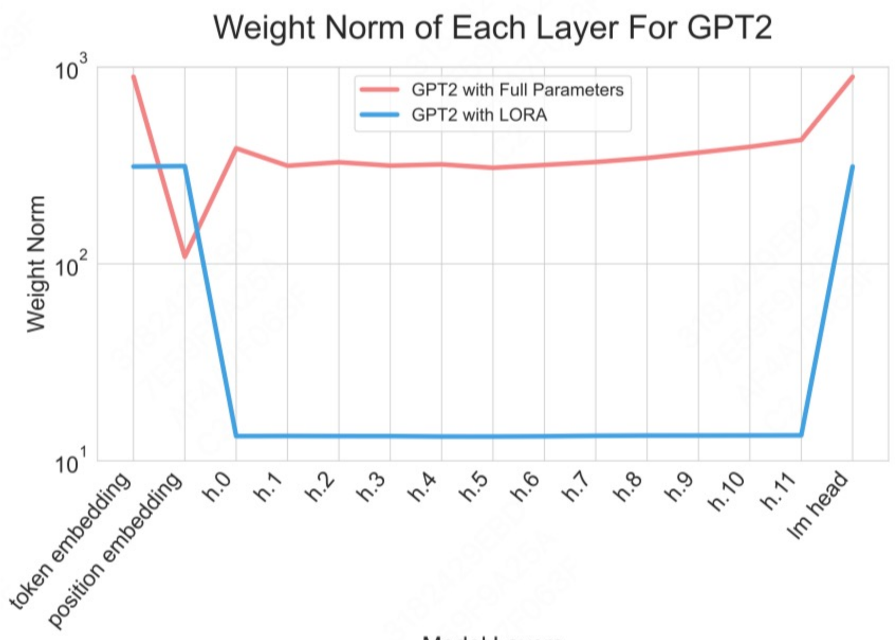
- LoRA is resource-efficient, but generally underperform full FT
- Delve into training statistics in each layer for LoRA and full FT
- Tune on Alpaca-GPT4, record mean norms of each layer at every step

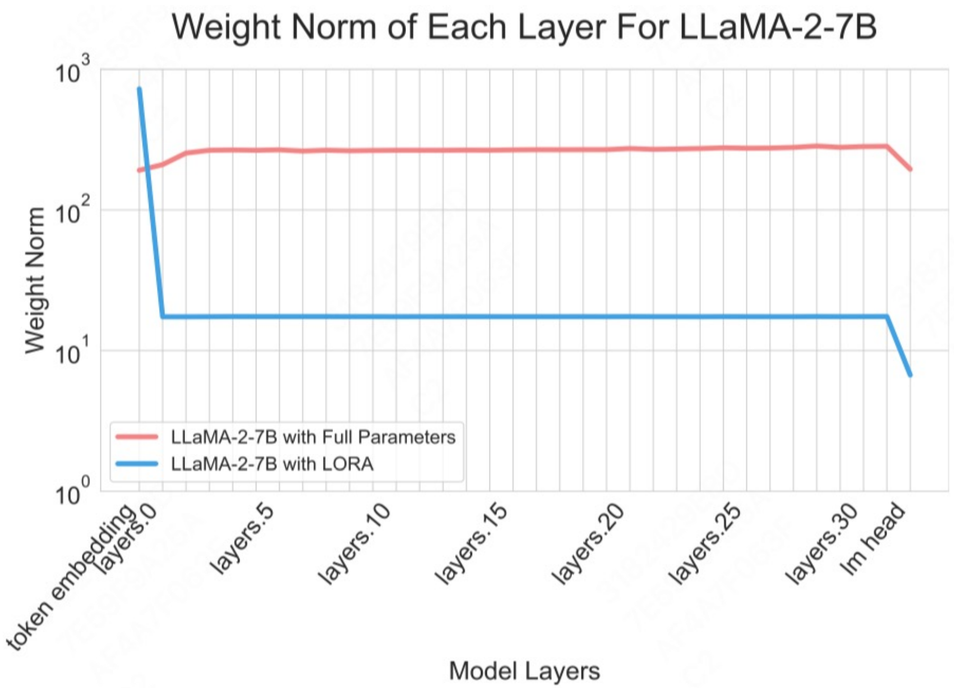
$$\mathbf{w}^{(\ell)} \triangleq \text{mean-weight-norm}(\ell) = \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{\theta}_t^{(\ell)}\|_2$$

Observation

- Embedding or LM head exhibits **significantly larger norms** than intermediary layers in LoRA

- LoRA values **layerwise importance** differently from full fine-tuning





LISA

Simulate LoRA's updating pattern via sampling layers to freeze:

- **Layers with small norms in LoRA should also have small sampling probabilities to unfreeze in *full-parameter* settings**
- **Probabilities:** $\{p_\ell\}_{\ell=1}^{N_L} = \{1.0, \gamma/N_L, \gamma/N_L, \ldots, \gamma/N_L, 1.0\}$

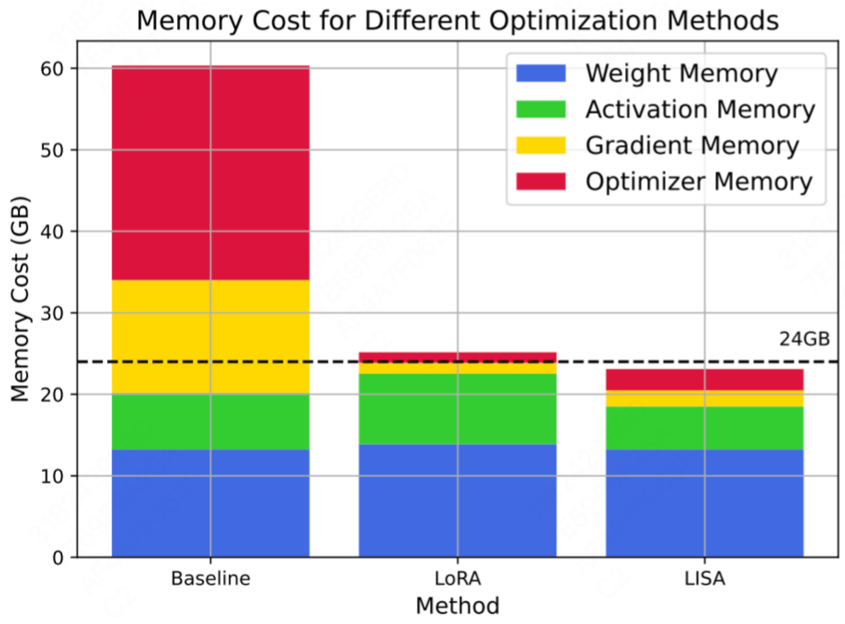
Algorithm 1 Layerwise Importance Sampling AdamW (**LISA**)

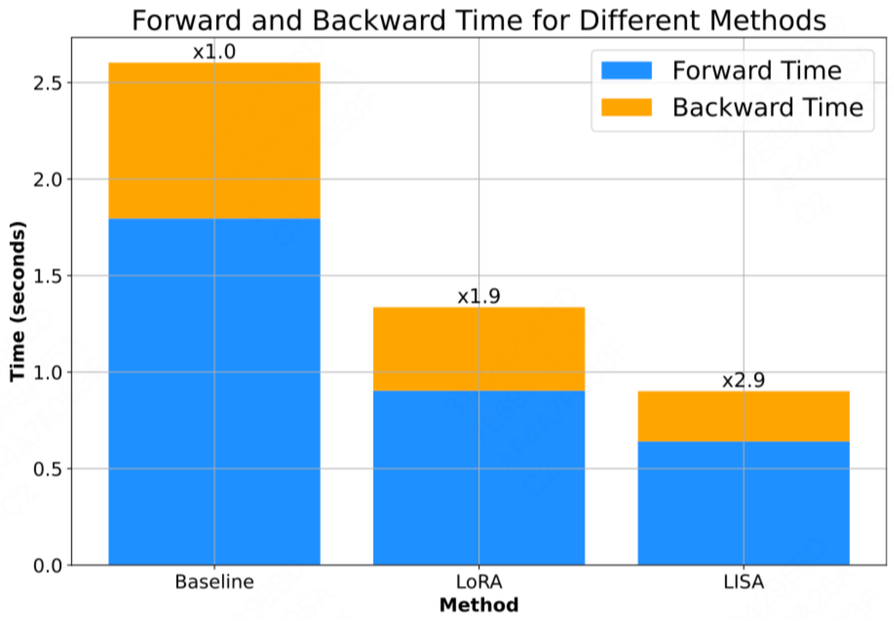
Require: number of layers N_L, number of iterations T, sampling period K, number of sampled layers γ, initial learning rate η_0

1: **for** $i \leftarrow 0$ to $T/K - 1$ **do**
2: Freeze all layers except the embedding and language modeling head layer
3: Randomly sample γ intermediate layers to unfreeze
4: Run AdamW for K iterations with $\{\eta_t\}_{t=ik}^{ik+k-1}$
5: **end for**

Experimental Results: Memory Efficiency

- **Memory reduction in LISA allows LLaMA-2-7B to be trained on a single RTX4090 (24GB) GPU**
- **LISA provides almost 2.9 × speedup when compared with full-parameter training, and ~ 1.5 × speedup against LoRA**





Experimental Results: Task Performance

- **Setting:**
  - **Train on instruction-following task Alpaca GPT-4 (52k conversation pairs)**
  - **Test on multiple benchmarks: MT-Bench, MMLU, AGIEval, WinoGrande**

MODEL	METHOD	MMLU (5-SHOT)	AGIEVAL (3-SHOT)	WINOGRANDE (5-SHOT)	MT-BENCH ↑
TINYLLAMA	VANILLA	25.50	19.55	59.91	1.25
	LORA	25.81 ± 0.07	19.82 ± 0.11	61.33 ± 0.09	1.90 ± 0.14
	GALORE	25.21 ± 0.06	21.19 ± 0.07	61.09 ± 0.12	**2.61 ± 0.17**
	LISA	**26.02 ± 0.13**	**21.71 ± 0.09**	61.48 ± 0.08	2.57 ± 0.25
	FT	25.62 ± 0.10	21.28 ± 0.07	**62.12 ± 0.15**	2.21 ± 0.16
MISTRAL-7B	VANILLA	60.12	26.79	79.24	4.32
	LORA	61.78 ± 0.09	27.56 ± 0.07	78.85 ± 0.11	4.41 ± 0.09
	GALORE	57.87 ± 0.08	26.23 ± 0.05	75.85 ± 0.13	4.36 ± 0.16
	LISA	**62.09 ± 0.10**	**29.76 ± 0.09**	**78.93 ± 0.08**	**4.85 ± 0.14**
	FT	61.70 ± 0.13	28.07 ± 0.12	78.85 ± 0.12	4.64 ± 0.12
LLAMA-2-7B	VANILLA	45.87	25.69	74.11	3.29
	LORA	45.50 ± 0.07	24.73 ±0.04	74.74 ± 0.09	4.45 ± 0.15
	GALORE	45.56 ± 0.05	24.39 ± 0.11	73.32 ± 0.12	4.63 ± 0.09
	LISA	**46.21 ± 0.12**	26.06 ± 0.08	**75.30 ± 0.11**	**4.94 ± 0.14**
	FT	45.66 ± 0.09	**27.02 ± 0.10**	75.06 ± 0.13	4.75 ± 0.16

Experimental Results: Task Performance

- **Results:**
  - **LISA outperforms other fine-tuning methods in most tracks**
  - **LISA even outperforms Full-parameter Training (*similar to dropout*)**

Model	Method	MMLU (5-shot)	AGIEval (3-shot)	WinoGrande (5-shot)	MT-Bench ↑
TinyLlama	Vanilla	25.50	19.55	59.91	1.25
	LoRA	25.81 ± 0.07	19.82 ± 0.11	61.33 ± 0.09	1.90 ± 0.14
	GaLore	25.21 ± 0.06	21.19 ± 0.07	61.09 ± 0.12	**2.61 ± 0.17**
	LISA	**26.02 ± 0.13**	**21.71 ± 0.09**	61.48 ± 0.08	2.57 ± 0.25
	FT	25.62 ± 0.10	21.28 ± 0.07	**62.12 ± 0.15**	2.21 ± 0.16
Mistral-7B	Vanilla	60.12	26.79	79.24	4.32
	LoRA	61.78 ± 0.09	27.56 ± 0.07	78.85 ± 0.11	4.41 ± 0.09
	GaLore	57.87 ± 0.08	26.23 ± 0.05	75.85 ± 0.13	4.36 ± 0.16
	LISA	**62.09 ± 0.10**	**29.76 ± 0.09**	**78.93 ± 0.08**	**4.85 ± 0.14**
	FT	61.70 ± 0.13	28.07 ± 0.12	78.85 ± 0.12	4.64 ± 0.12
LLaMA-2-7B	Vanilla	45.87	25.69	74.11	3.29
	LoRA	45.50 ± 0.07	24.73 ±0.04	74.74 ± 0.09	4.45 ± 0.15
	GaLore	45.56 ± 0.05	24.39 ± 0.11	73.32 ± 0.12	4.63 ± 0.09
	LISA	**46.21 ± 0.12**	26.06 ± 0.08	**75.30 ± 0.11**	**4.94 ± 0.14**
	FT	45.66 ± 0.09	**27.02 ± 0.10**	75.06 ± 0.13	4.75 ± 0.16

Ablation Studies

- **Hyperparameters of LISA**
  - Increasing **sampling layers** and **sampling period** leads to better performance
- **Sensitiveness of LISA**
  - LISA is quite resilient to different **random seeds**

Models	γ	K	MT-Bench Score
TinyLlama	2	$\lceil T/125 \rceil$	2.44
		$\lceil T/25 \rceil$	**2.73**
		$\lceil T/5 \rceil$	2.64
		T	2.26
	8	$\lceil T/125 \rceil$	2.59
		$\lceil T/25 \rceil$	**2.81**
		$\lceil T/5 \rceil$	2.74
		T	2.53

Model	Seed 1	Seed 2	Seed 3
TinyLlama	2.57	2.55	2.60
Mistral-7B	4.85	4.82	4.82
LLaMA-2-7B	4.94	4.92	4.89

Memorization and Reasoning

- **LISA is much better than LoRA at memorization-centered tasks**
  - **LISA emphasizes width and restricts depth**
  - **LoRA emphasizes depth and restricts width**
- **Width is crucial for memorization, depth is important for reasoning**

Model & Method	MT-Bench								
	Writing	Roleplay	Reasoning	Code	Math	Extraction	STEM	Humanities	Avg. ↑
TinyLlama (Vanilla)	1.05	2.25	1.25	1.00	1.00	1.00	1.45	1.00	1.25
TinyLlama (LoRA)	2.77	4.05	1.35	1.00	**1.40**	1.00	1.55	2.15	1.90
TinyLlama (GaLore)	**3.55**	**5.20**	2.40	**1.15**	1.40	**1.85**	2.95	2.40	**2.61**
TinyLlama (**LISA**)	3.30	4.40	**2.65**	1.12	1.30	1.75	**3.00**	**3.05**	2.57
TinyLlama (FT)	3.27	3.95	1.35	1.04	1.33	1.73	2.69	2.35	2.21
Mistral-7B (Vanilla)	5.25	3.20	4.50	1.60	2.70	**6.50**	**6.17**	4.65	4.32
Mistral-7B (LoRA)	5.30	4.40	4.65	**2.35**	**3.30**	5.50	5.55	4.30	4.41
Mistral-7B (GaLore)	5.05	**5.27**	4.45	1.70	2.50	5.21	5.52	5.20	4.36
Mistral-7B (**LISA**)	**6.84**	3.65	**5.45**	2.20	2.75	5.65	5.95	**6.35**	**4.85**
Mistral-7B (FT)	5.50	4.45	5.45	2.50	3.25	5.78	4.75	5.45	4.64
LLaMA-2-7B (Vanilla)	2.75	4.40	2.80	1.55	1.80	3.20	5.25	4.60	3.29
LLaMA-2-7B (LoRA)	6.30	5.65	**4.05**	1.60	1.45	4.17	6.20	6.20	4.45
LLaMA-2-7B (GaLore)	5.60	6.40	3.20	1.25	1.95	**5.05**	6.57	7.00	4.63
LLaMA-2-7B (**LISA**)	**6.55**	**6.90**	3.45	**1.60**	**2.16**	4.50	**6.75**	**7.65**	**4.94**
LLaMA-2-7B (FT)	5.55	6.45	3.60	1.75	2.00	4.70	6.45	7.50	4.75

Table of Contents

Paper Information

Understanding Layer Significance in LLM Alignment

Guangyuan Shi[1], Zexin Lu[1], Xiaoyu Dong[1], Wenlong Zhang[1], Xuanyu Zhang[2],
Yujie Feng[1], Xiao-Ming Wu[1]✉
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong S.A.R., China
[2]Du Xiaoman Financial, China
{guang-yuan.shi, zexin.lu, xiaoyu.dong}@connect.polyu.hk,
{wenlong.zhang, yujie.feng}@connect.polyu.hk,
xyz@mail.bnu.edu.cn, xiao-ming.wu@polyu.edu.hk

Motivation

- **LIMA [1] posits pretraining develops knowledge and capabilities, alignment refine conversational style and formatting**
- *Only certain components of LLMs are significantly impacted?*
- **Examine alignment in model parameter level (layer significance) to gain deeper understanding**

[1] Lima: Less is more for alignment. NIPS 203. Chunting Zhou, Pengfei Liu, Puxin Xu and et al.

Quantify Layer Significance

ILA: learn a binary mask to indicate significance for each layer

- **Definition 1:** $\in$-stable at iteration T. For any $t > T$, loss satisfies

$$|\mathbb{E}_z[\mathcal{L}(\boldsymbol{\theta}_{t+1}, z)] - \mathbb{E}_z[\mathcal{L}(\boldsymbol{\theta}_t, z)]| < \epsilon,$$

- **Definition 2:** Layer Importance. Binary mask $\gamma_t = \{\gamma_t^i \mid \gamma_t^i \in \{0, 1\}\}_{\{i=1\}}^{\{N\}}$

$$\gamma_t = \arg\min_{\gamma_t} \mathcal{L}(\boldsymbol{\theta}_t^{\text{mask}}), \quad \text{s.t.} \quad \|\gamma_t\| < H,$$

$$\boldsymbol{\theta}_t^{\text{mask}} = \boldsymbol{\theta}_0 + \gamma_t \odot \Delta\boldsymbol{\theta}_t$$

Quantify Layer Significance

ILA: learn a binary mask to indicate significance for each layer

Algorithm 1: Identify the Important Layers for Alignment (ILA)

Input: Pre-trained model parameters θ_0, learning rate α, the initial importance score vector $s_0 = \{s_0^i\}_{i=1}^N$, the number of insignificant layers K, the low-rank matrices A_0, B_0 for the LoRA algorithm.

$$\gamma_t^i = \sigma(s_t^i)$$

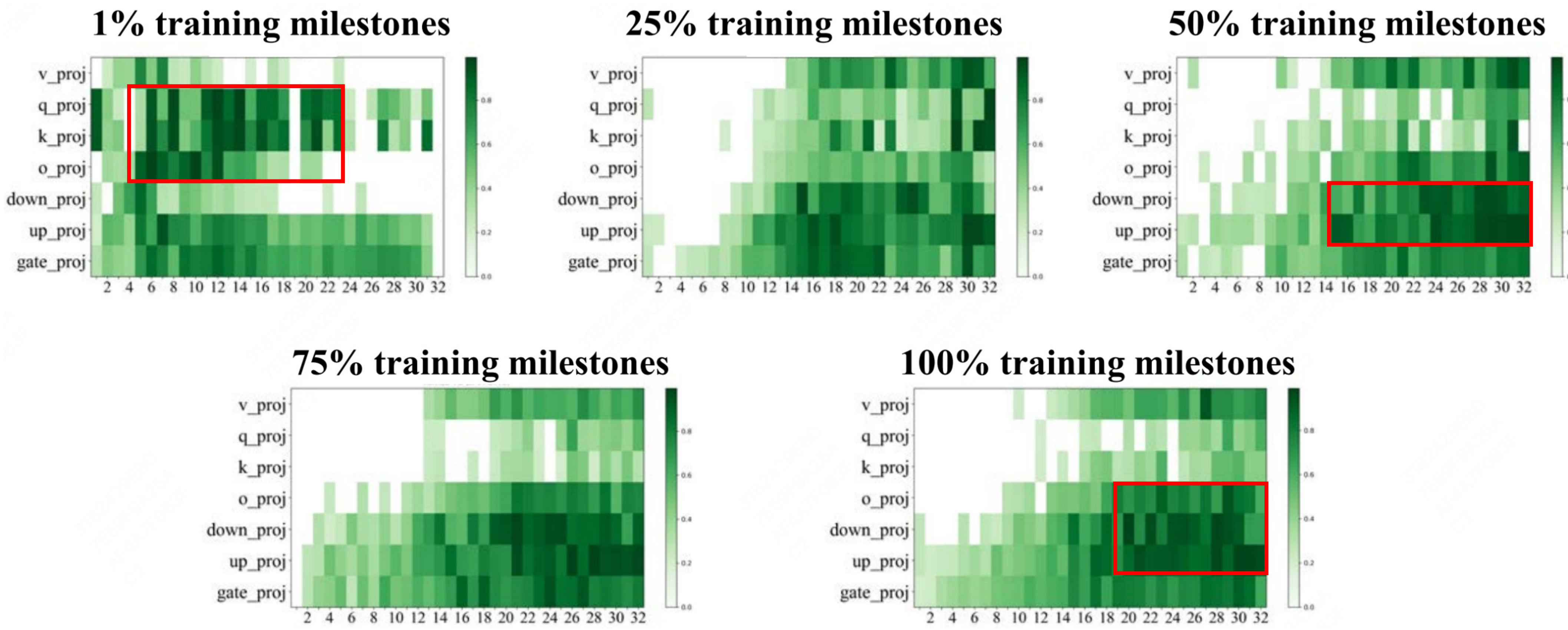
for iteration i = 1, 2, … **do**
- Update $A_t = A_{t-1} - \alpha \nabla_{A_{t-1}} \mathcal{L}(\theta_t)$;
- Update $B_t = B_{t-1} - \alpha \nabla_{B_{t-1}} \mathcal{L}(\theta_t)$;
- **if** *Training has become stable* **then**
  - Solve the optimization problem in Eq. (7) by gradient descent to find $s_t = \{s_t^i\}_{i=1}^N$;
  - Stop training;
- **end**

end

$$s_t = \arg\min_{s_t} \mathcal{L}(\theta_t^{\mathrm{M}}).$$

Layer Importance Ranking

- Layer importance ranking of LLAMA 2-7B identified by ILA on LIMA in different training milestones:



Layer Importance Across Datasets

- **Define top *75% highest-scoring* layers as important layers (Set S)**
- **Jaccard similarity between two datasets:** $J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$
- **Important layers for different datasets exhibit high similarity**

Datasets	LLAMA 2-7B			Mistral-7B		
	LIMA	No Robots	Alpaca-GPT4	LIMA	No Robots	Alpaca-GPT4
LIMA	-	-	-	-	-	-
No Robots	0.91	-	-	0.90	-	-
Alpaca-GPT4	0.90	0.90	-	0.89	0.93	-

Freeze Unimportant Layers

- **Exclude 25% unimportant layers, whose modifications would *negatively impact fine-tuning***
- **Freezing unimportant layers may enhance performance**

Models	Methods	Language Understanding		Conversational Ability	
		MMLU ↑	Hellaswag ↑	Vicuna ↑	MT-Bench ↑
LLAMA 2-7B	AdaLoRA	45.23	57.30	5.70	4.05
	Full Finetune	45.72	57.69	6.00	3.93
	Full Finetune w/ ILA	**45.98**	57.87	5.90	4.21
	LoRA	44.58	59.46	6.23	4.70
	LoRA w/ ILA	45.78	**59.65**	**6.30**	**4.93**
Mistral-7B-v0.1	AdaLoRA	62.13	61.68	6.10	5.03
	Full Finetune	61.05	**64.26**	6.70	5.56
	Full Finetune w/ IFILA	61.75	64.21	6.73	**5.70**
	LoRA	61.95	62.90	6.77	5.35
	LoRA w/ IFILA	**62.14**	62.80	**6.82**	5.42

Comparative evaluation of models finetuned on the LIMA Dataset.

Tuning Critical Layers Only

- **Fine-tune *only important layers* of Mistral-7B, as identified by ILA, on the No Robots dataset**
- **Focusing on selected important layers nearly matches the performance of full fine-tuning**

Models	Methods	Language Understanding		Conversational Ability	
		MMLU ↑	Hellaswag ↑	Vicuna ↑	MT-Bench ↑
Mistral-7B-v0.1	LoRA	**61.95**	**62.90**	**6.77**	5.35
	LoRA w/ ILA (10%)	62.09	61.94	6.49	5.08
	LoRA w/ ILA (20%)	61.83	62.16	6.60	5.23
	LoRA w/ ILA (30%)	61.89	62.79	6.71	**5.37**

Ablation Study

- **Randomly or manually selecting layers does not work**
  - RL 1 and 2: randomly select K layers to freeze with different seeds
  - FL: freeze the first K linear layers
  - LL: freeze the last K linear layers

Methods	Language Understanding		Conversational Ability	
	MMLU ↑	Hellaswag ↑	Vicuna ↑	MT-Bench ↑
LoRA	44.58	59.46	6.23	4.70
LoRA w/ RL 1	44.23	59.71	6.08	4.60
LoRA w/ RL 2	43.98	59.11	6.10	4.68
LoRA w/ FL	44.02	59.32	6.13	4.59
LoRA w/ LL	44.61	59.21	6.20	4.63
LoRA w/ ILA	45.78	59.65	6.30	4.93

Cross-dataset Evaluation

- **An intuitive hypothesis: layers *consistently deemed unimportant* across all datasets may truly be non-essential**
- ***Intersect the top-K least important* layers from three datasets**
- **Imp. layers across datasets yields better results than specific dataset**

Dataset (Imp. Layers)	Dataset (Finetune)	Language Understanding		Conversational Ability	
		MMLU ↑	Hellaswag ↑	Vicuna ↑	MT-Bench ↑
LIMA	LIMA	**61.82**	65.48	6.99	5.38
No Robots	LIMA	61.52	65.51	6.92	5.34
Alpaca-GPT4	LIMA	61.23	65.20	7.03	5.21
Intersection	LIMA	61.49	**65.62**	**7.06**	**5.44**

Table of Contents

Paper Information

Layer-wise Importance Matters: Less Memory for Better Performance in Parameter-efficient Fine-tuning of Large Language Models

Kai Yao[1,2*], **Penlei Gao**[3*], **Lichun Li**[2], **Yuan Zhao**[2],
Xiaofeng Wang[3], **Wei Wang**[2†], **Jianke Zhu**[1†],
[1]Zhejiang University [2]Ant Group [3]Cleveland Clinic Lerner Research Institution
jiumo.yk@antgroup.com, gaop@ccf.org

Motivation

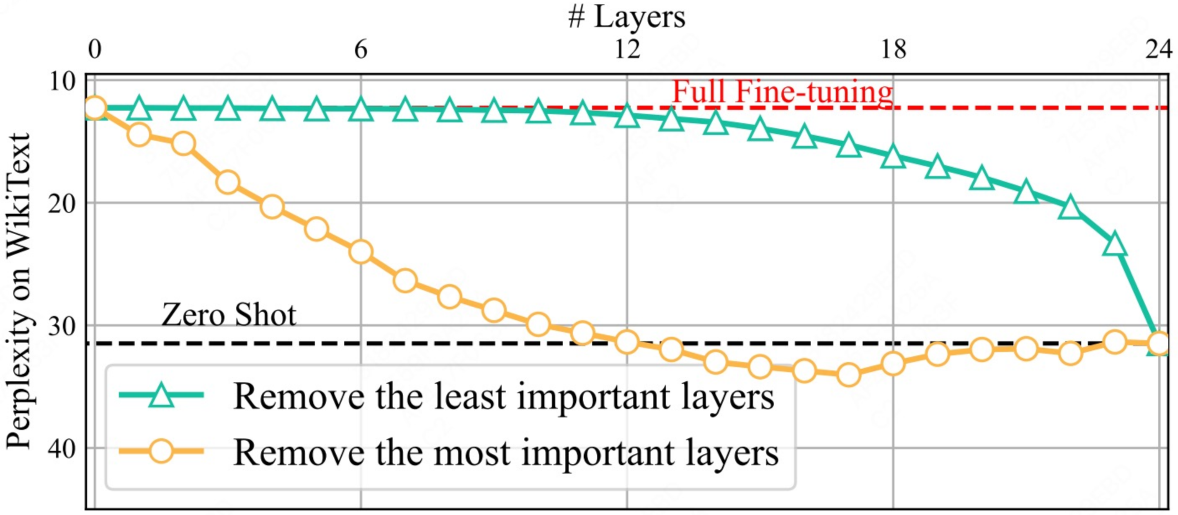
- **LoRA apply uniform architectural across all layers, ignores the varying importance of each layer**
- **LISA trains partial layers and yields promising results**
- **IST estimates task-specific importance score of each layer**

Preliminary Observation

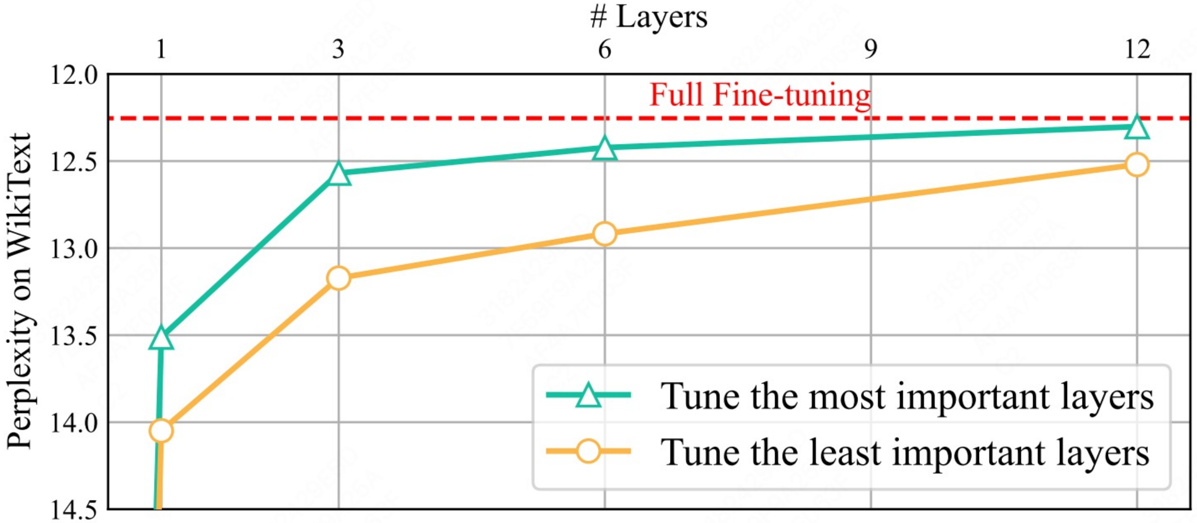
Apply LoRA to OPT 1.3B on WikiText across all layers:

1. **Gradually remove layers according to contribution to performance**
2. **Performed PEFT on the most and least important layers**

Layer-wise sparsity in PEFT is an inherent characteristic



(a) Remove trained LoRA modules layer-by-layer greedily

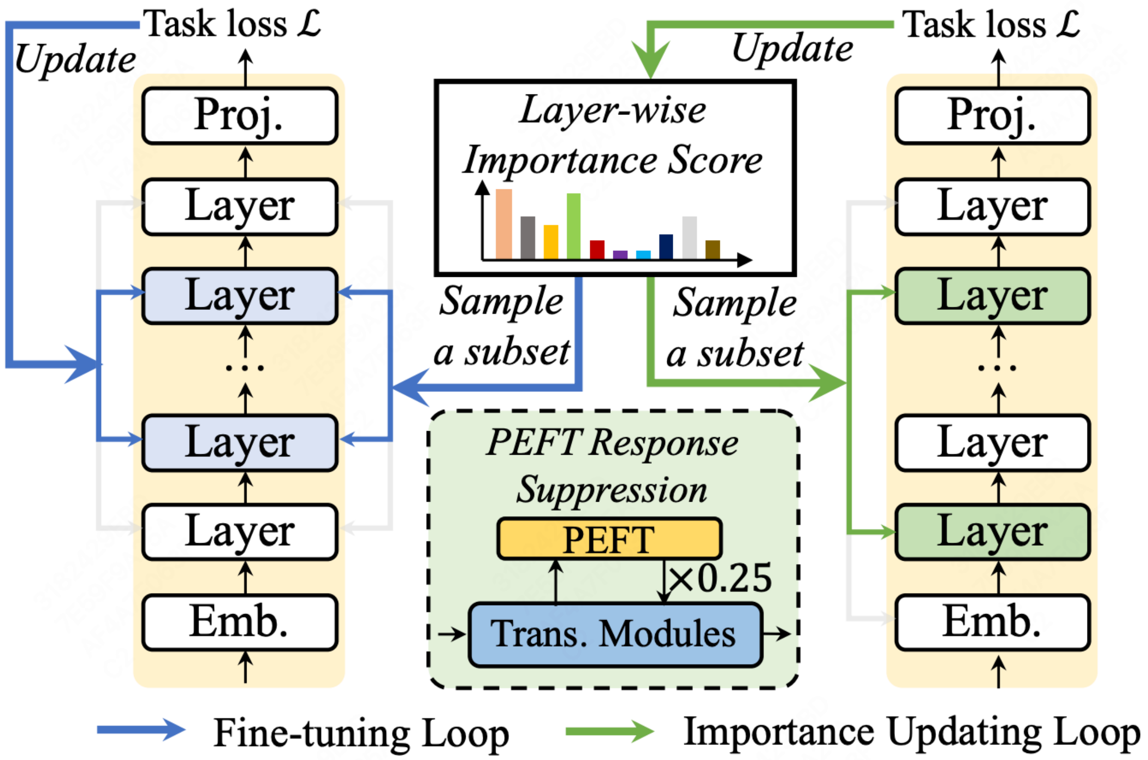


(b) Train LoRA modules within the selective layers

Importance-aware Sparse Tuning

IST involves two loops (*similar to data minimization*):

- **Fine-tuning loop**: selects a subset of full layers to update
- **Importance updating loop**: updates importance score of each layer



Importance-aware Sparse Tuning

- Fine-tuning loop: Define degree of importance as $I \in \mathbb{R}^{N_L}$ and choose Nu layers to update based on I in each iteration
- Importance updating loop:
  - Suppress the response of layer to measure its contribution to the result

$$o_{i+1}^{j} = \begin{cases} m_i(o_i^j) + a_i(o_i^j), & \text{if } i \in S_c^j \\ m_i(o_i^j) + \beta * a_i(o_i^j), & \text{otherwise} \end{cases}$$

  - Calculate the rewards according to their loss

$$\mathbf{r}^j = e^{-\mathcal{L}^j} - \frac{1}{N_c}\sum_{k=1}^{N_c} e^{-\mathcal{L}^k}$$

  - Employ reward to update importance score

$$\mathbf{I}_i = \begin{cases} \mathbf{I}_i + \mu * \mathbf{r}_j, & \text{if } i \in S_c^j \\ \mathbf{I}_i, & \text{otherwise} \end{cases}$$

Experimental Results

IST consistently shows an enhancement in model performance on the commonsense reasoning task.

Model	PEFT	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
$LLaMA_{7B}$	Series	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Series + IST	66.2	78.3	74.9	72.2	75.9	75.8	59.0	72.2	**71.8**
	Parallel	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.2
	Parallel + IST	68.4	79.1	77.9	70.0	78.9	81.2	62.3	77.6	**74.4**
	LoRA	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	LoRA + IST	68.7	81.7	77.3	82.7	78.7	80.6	62.4	80.0	**76.5**
$LLaMA_{13B}$	Series	71.8	83.0	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Series + IST	72.9	82.2	81.4	87.9	84.0	82.7	69.1	81.1	**80.2**
	Parallel	72.5	84.9	79.8	92.1	84.7	84.2	71.2	82.4	**81.4**
	Parallel + IST	72.6	86.0	79.2	89.1	83.5	84.8	70.6	82.8	81.1
	LoRA	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	LoRA + IST	71.5	85.0	81.2	89.1	84.2	84.0	70.1	81.8	**80.9**
$GPT\text{-}J_{6B}$	LoRA	62.4	68.6	49.5	43.1	57.3	43.4	31.0	46.6	50.2
	LoRA + IST	63.0	63.2	62.9	35.8	39.1	56.8	39.1	51.2	**51.4**
$BLOOMz_{7B}$	LoRA	65.9	75.3	74.5	57.3	72.5	74.6	57.8	73.4	**68.9**
	LoRA + IST	67.0	74.4	74.4	51.4	68.7	77.9	58.9	74.4	68.4
$LLaMA3_{8B}$	LoRA	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	LoRA + IST	72.7	88.3	80.5	94.7	84.4	89.8	79.9	86.6	**84.6**

Layer-wise Importance Learning

Visualize layer-wise importance learning process of two tasks

- *Layer 2 and 32* significantly contribute to **commonsense reasoning** task
- *Layer 6 and 18* contribute to **arithmetic reasoning** task most

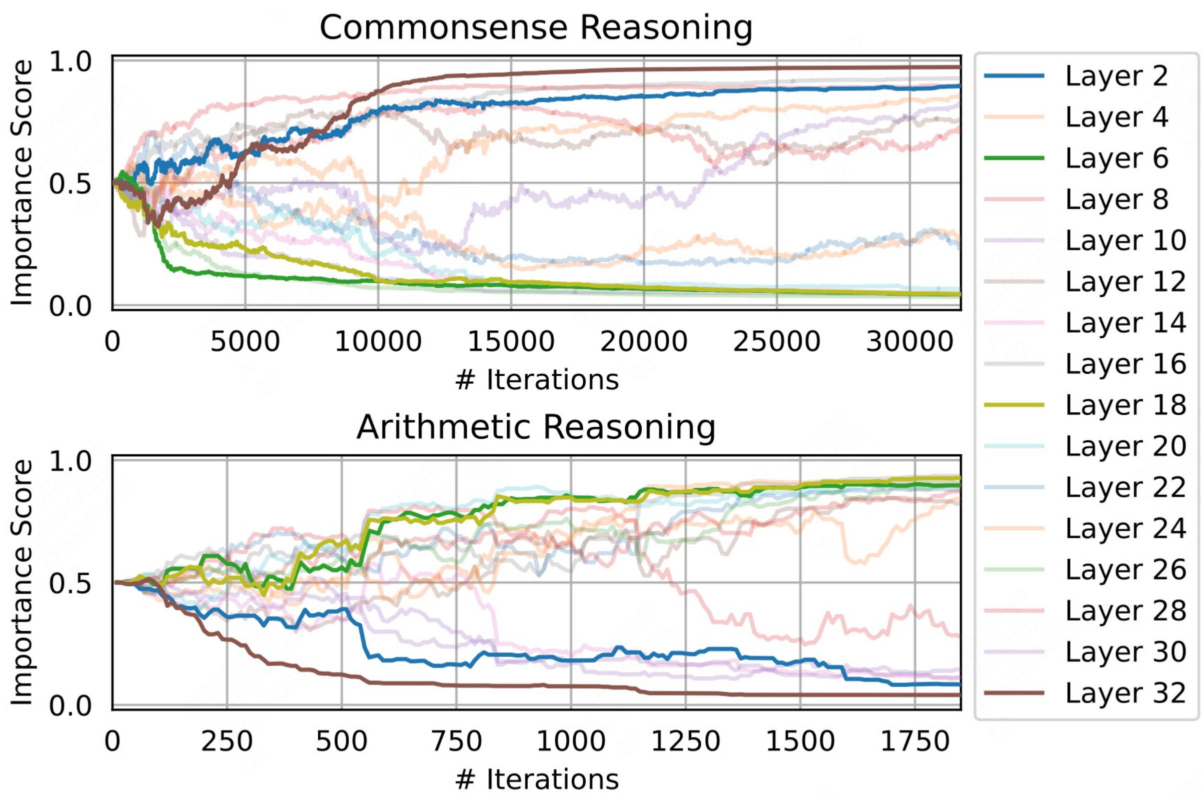


Table of Contents

Conclusions

- **LISA:**
  - observe the magnitude of parameter changes
  - design importance probability
  - repeatedly sample a subset of layers during training
- **ILA:**
  - train all layers until convergence
  - learn a binary mask to select beneficial parameter changes
- **IST:**
  - two loops to jointly learn importance scores and parameter updates

Table of Contents

Related Works

- **LIFT: Efficient Layer-wise Fine-tuning for Large Model Models (ArXiv 2024)**
  - layer-wise fine-tuning strategy that only learns one layer at a time
- **Random Masking Finds Winning Tickets for Parameter Efficient Fine-tuning (ICML 2024)**
  - use random masking to fine-tune the pretrained model

Related Works

- **Investigating Layer Importance in Large Language Models (ArXiv 2024)**
  - propose an efficient sampling method to faithfully evaluate the importance of layers using Shapley values (certain early layers exhibit dominant contribution)
- **Spectral Insights into Data-Oblivious Critical Layers in Large Language Models (ACL 2025 Findings)**
  - layers with significant shifts in representation space are also those most affected during fine-tuning -- a pattern that holds consistently across tasks for a given model

Table of Contents

Discussion

- Layers in LLMs indeed exhibit varying functions and levels of importance, which is intuitive—after all, not all modules can be equally important
- There is currently no consensus on layer importance and different studies report varying findings (as a result, their impact has been limited)
- If localized fine-tuning is necessary, the ideal solution would be an efficient empirical proxy that enables global identification of critical components, with conclusions that generalize within same architecture.

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

智能算法安全全国重点实验室
State Key Laboratory of AI Safety

Thank you for listening!

Any questions?