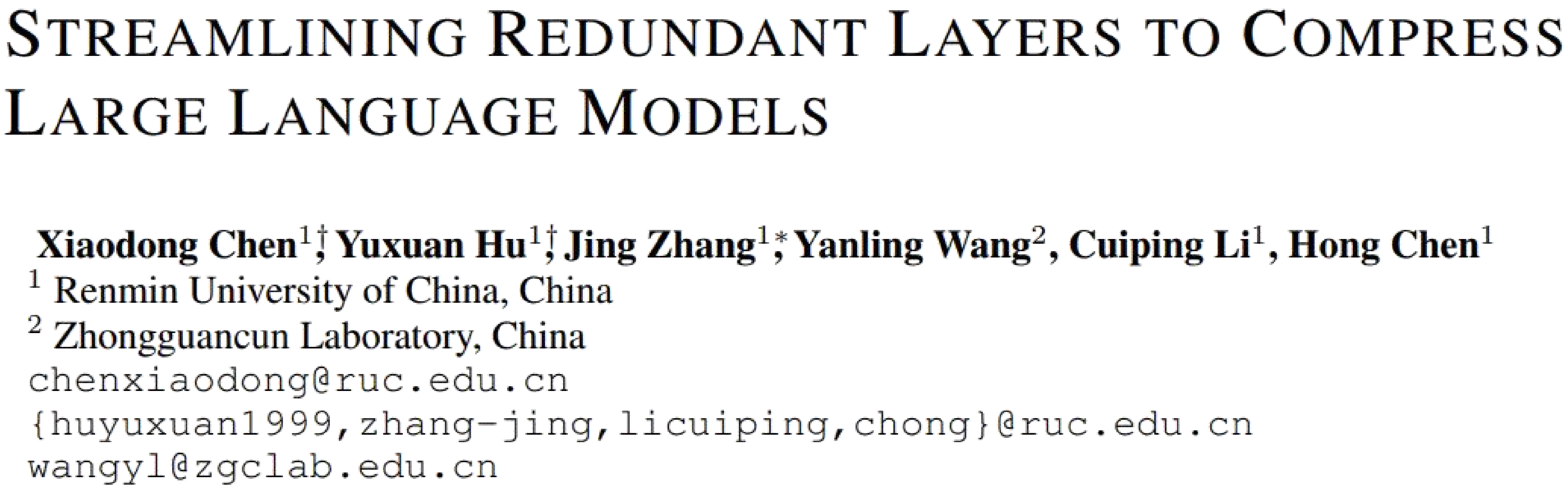
ICLR 2025



Streamlining Redundant Layers to Compress Large Language Models

Xiaodong Chen[1]†, Yuxuan Hu[1]†, Jing Zhang[1]*, Yanling Wang[2], Cuiping Li[1], Hong Chen[1]
[1] Renmin University of China, China
[2] Zhongguancun Laboratory, China
chenxiaodong@ruc.edu.cn
{huyuxuan1999,zhang-jing,licuiping,chong}@ruc.edu.cn
wangyl@zgclab.edu.cn

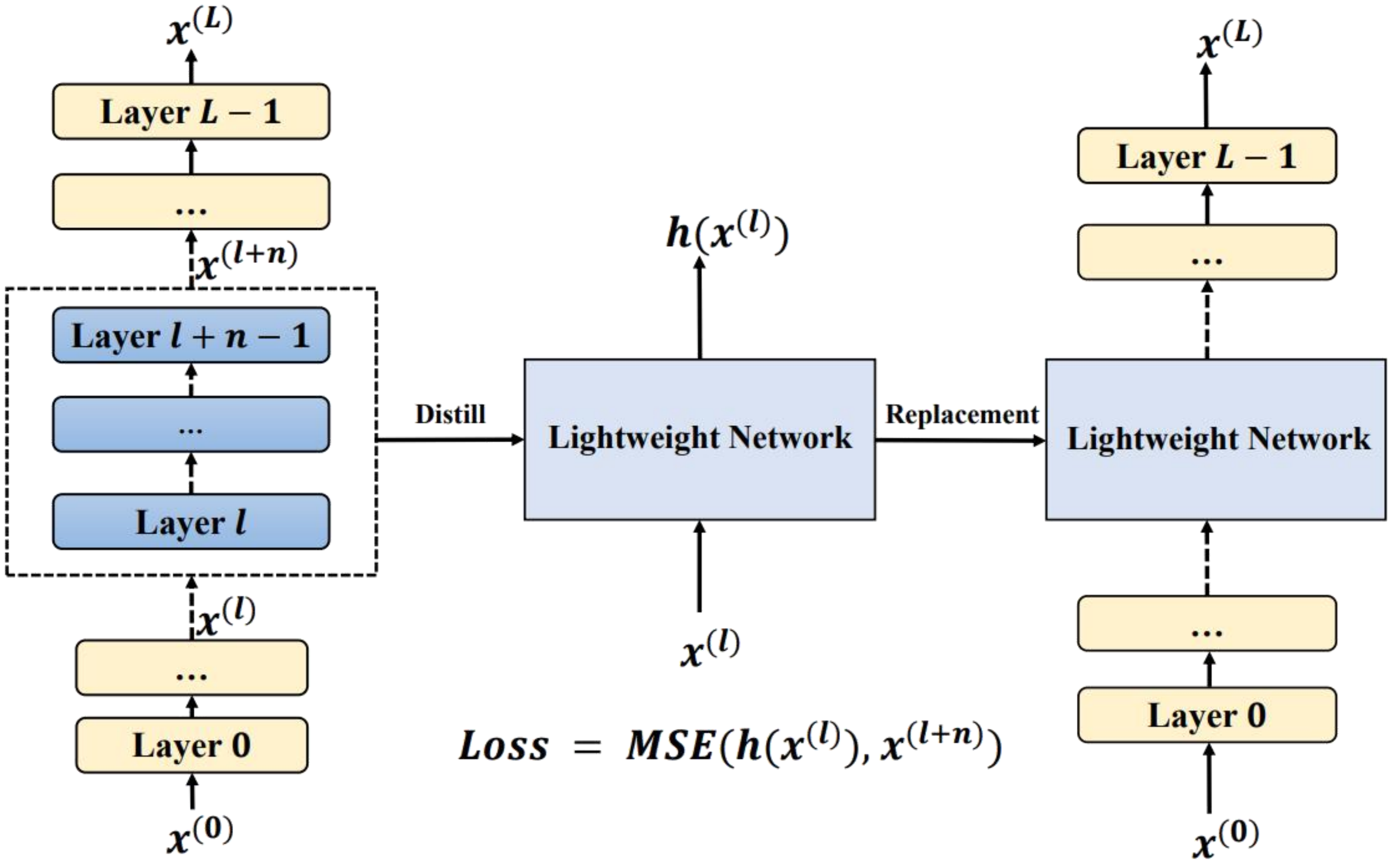
Xiaodong Chen
chenxiaodong@ruc.edu.cn

Abstract

This paper presents LLM-Streamline, a novel layer pruning method for LLMs. It identifies less important layers by analyzing their impact on hidden states and prunes them based on a target rate.

LLM-Streamline includes layer pruning, which removes less important consecutive layers, and layer replacement, a new module that trains a lightweight network to replace pruned layers, reducing performance loss.

A new metric, stability, is introduced to to address the limitations of the widely used accuracy metric in evaluating pruned models.



(1) Layer Pruning (2) Training Lightweight Network (3) Layer Replacement

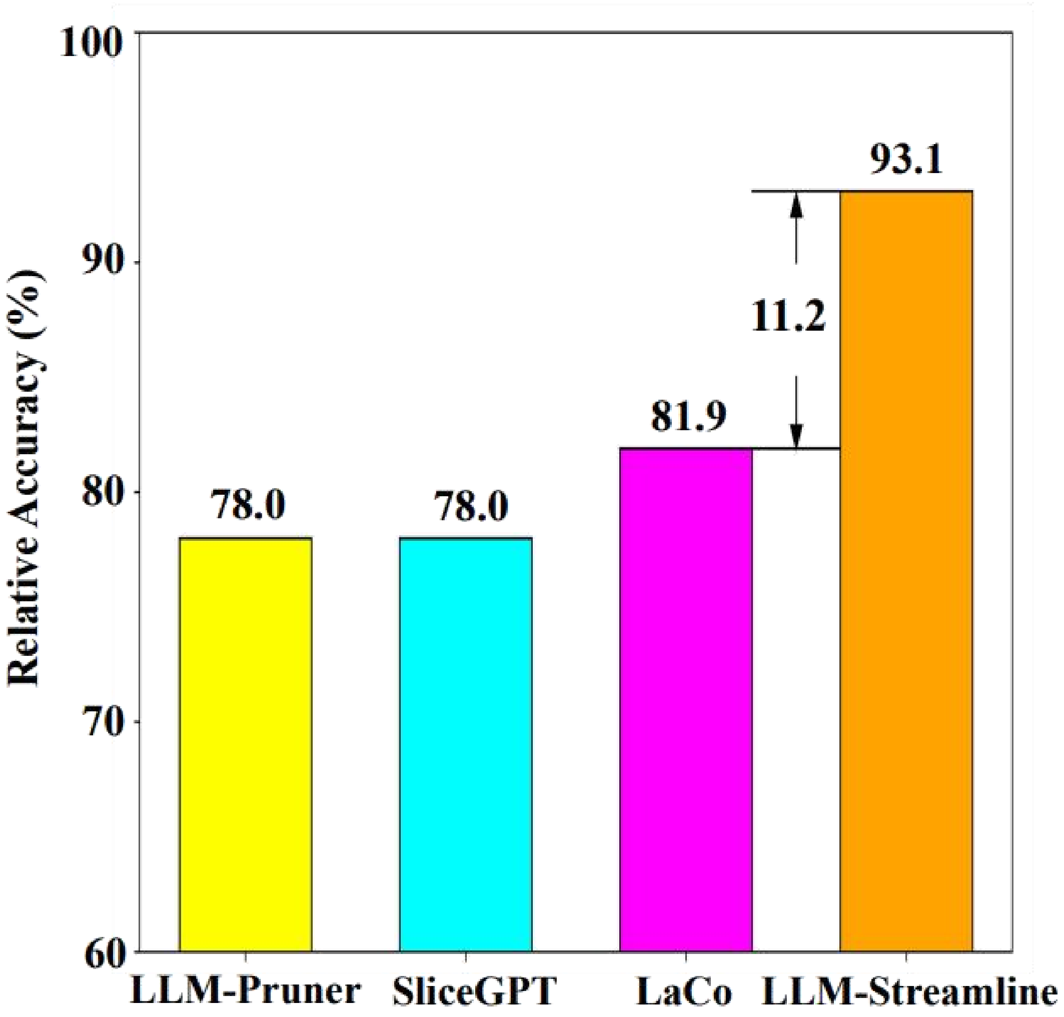
The workflow of LLM-Streamline

Background

With the rapid advancement of large language models (LLMs), their increasing parameter sizes and computational demands pose significant challenges for deployment in resource-constrained environments.

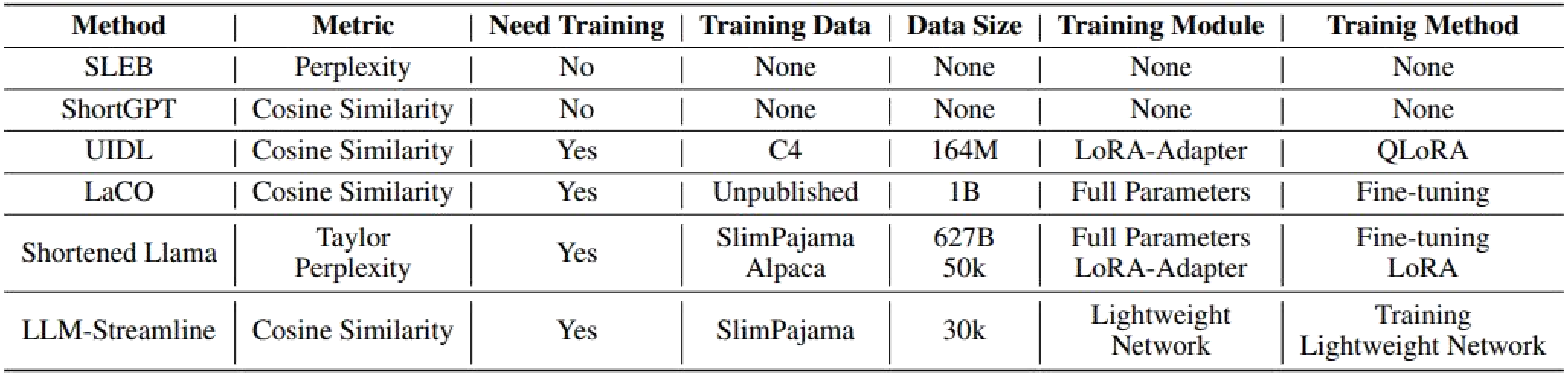
To address this challenge, researchers have focused on exploring structured pruning techniques for LLMs, aiming to reduce the number of model parameters while preserving performance as much as possible.

However, current structured pruning methods, such as LLM-Pruner and SliceGPT, although capable of reducing model parameters to some extent, often lead to irregular model structures and exhibit notable performance gaps compared to the original models.



Performance comparison between LLM Streamline and existing methods

Difference from Concurrent Methods



Method	Metric	Need Training	Training Data	Data Size	Training Module	Trainig Method
SLEB	Perplexity	No	None	None	None	None
ShortGPT	Cosine Similarity	No	None	None	None	None
UIDL	Cosine Similarity	Yes	C4	164M	LoRA-Adapter	QLoRA
LaCO	Cosine Similarity	Yes	Unpublished	1B	Full Parameters	Fine-tuning
Shortened Llama	Taylor Perplexity	Yes	SlimPajama Alpaca	627B 50k	Full Parameters LoRA-Adapter	Fine-tuning LoRA
LLM-Streamline	Cosine Similarity	Yes	SlimPajama	30k	Lightweight Network	Training Lightweight Network

Concurrent methods:

LLM-Streamline(ours, 2024.3)、LaCo(2024.2)、ShortGPT(2024.3)、UIDL(2024.3)、Shortened Llama(2024.2)、SLEB(2024.2)

Key difference:

Unlike concurrent methods, LLM-Streamline fundamentally differs by training a lightweight model to replace pruned layers, rather than directly removing them.

Compared to training-based pruning methods (e.g., Shortened Llama, LaCo, UIDL), LLM-Streamline significantly reduces computation time and resource consumption. Moreover, it achieves state-of-the-art (SOTA) performance compared to concurrent methods.

Method: Layer Pruning

Due to the widespread use of Pre-Norm in LLMs, the impact of each layer on the hidden state of its input can be expressed as:

$$\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x}^{(\ell)} + f(\boldsymbol{x}^{(\ell)}, \theta^{(\ell)})$$

Therefore, we estimate the importance of each layer by evaluating its impact on the hidden state, specifically using cosine similarity as the metric.

$$\boldsymbol{Importance}^{(\ell)} = \cos\left(\boldsymbol{x}^{(\ell)}, \boldsymbol{x}^{(\ell+1)}\right)$$

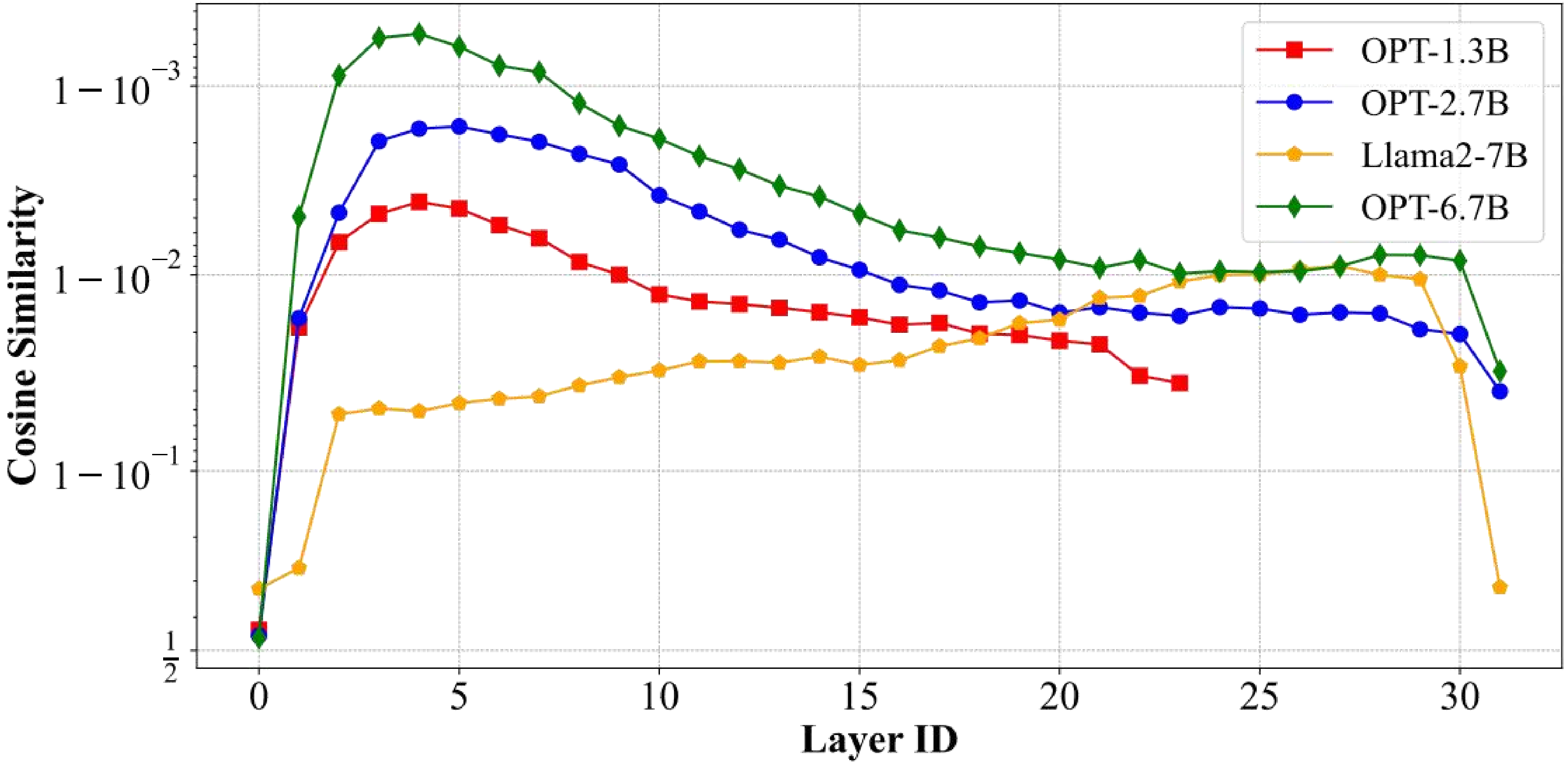
1. Why not use other metrics, such as Euclidean distance or dot product?

Due to the presence of Pre-Norm, the magnitude of the model's hidden states increases with layer depth, introducing bias. Specifically, deeper layers tend to have larger dot products between input and output hidden states, while shallower layers exhibit smaller Euclidean distances. Therefore, we use cosine similarity as the metric because it focuses solely on direction, ignoring magnitude.

2. Why not use perplexity or gradient-based methods?

Using perplexity or gradient-based methods can lead to a phenomenon similar to overfitting. This approach may result in a pruned model that performs well on the dataset used for pruning but fails to generalize to other datasets.

Method: Layer Relacement



Since we observe that layers with high cosine similarity tend to cluster together, we choose to prune consecutive layers and approximate their functionality by training a lightweight model.

Based on a predefined pruning rate, we select layers from i to i+n for pruning. We then collect the hidden states from the input of layer i and the output of layer i+n as training data, and use MSE Loss to train the lightweight model through distillation.

Comparison between Layer Replacement and LoRA

Previous structured pruning methods, such as LLM-Pruner, SliceGPT, and some layer pruning approaches, opt to use LoRA for training. Compared to LoRA, layer replacement offers the following advantages:

1. Lower GPU memory consumption: Layer replacement only requires the cost of forward propagation of the original model during hidden state collection and trains only the lightweight network during training, making it more memory-efficient than LoRA.

2. More reasonable training approach: Layer replacement trains a lightweight model to approximate the functionality of consecutive pruned layers, whereas LoRA directly trains the remaining layers. Therefore, we hypothesize that replacing pruned layers with a lightweight network may be simpler than training the remaining layers. Our experiments confirm this.

Can layer replacement be trained using SFT Loss?

While using SFT Loss for training is possible, MSE Loss is more memory-efficient and better suited for resource-constrained conditions. However, in resource-abundant scenarios, training with SFT Loss converges faster than using MSE Loss.

A New Metric: Stability

We analyzed the limitations of the widely used accuracy metric in evaluating the performance of pruned models and proposed a new stability metric to better assess the performance of pruned models.

First, we performed layer pruning on the model without any training and tested the pruned model's performance on several classification tasks. We observed that, in some tasks, the pruned model's performance unexpectedly improved compared to the original model. To further investigate this phenomenon, we analyzed it using a confusion matrix:

TP: Questions answered correctly by both the original and pruned models.
FN: Questions answered correctly by the original model but incorrectly by the pruned model.
FP: Questions answered incorrectly by the original model but correctly by the pruned model.
TN: Questions answered incorrectly by both the original and pruned models.

Additionally, we introduced the calculation process for perplexity and its standard deviation in classification task testing, where $\mathcal{M}$ denotes the original model, x_i denotes the i-th sample's question, $c_{i,j}$ represents the j-th option for that question.

$$\mathrm{PPL}_{i,j} = \mathrm{PPL}(\mathcal{M}(x_i, c_{i,j})), \mathrm{PPL}_i = \frac{\sum_{j=1}^{k} \mathrm{PPL}_{i,j}}{k}, \mathrm{std}_i = \sqrt{\frac{\sum_{j=1}^{k} (\mathrm{PPL}_{i,j} - \mathrm{PPL}_i)^2}{k-1}},$$

A New Metric: Stability

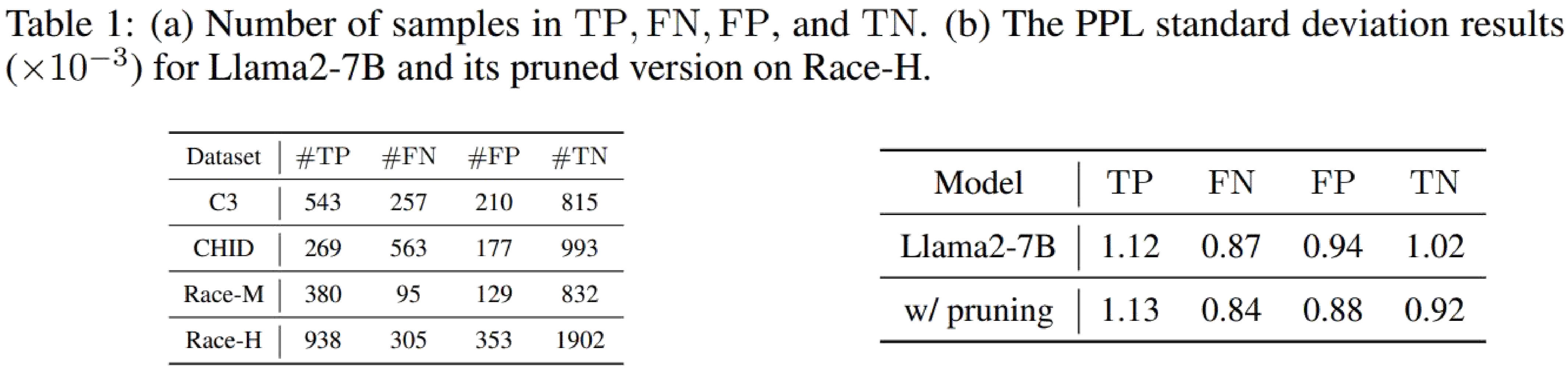


Table 1: (a) Number of samples in TP, FN, FP, and TN. (b) The PPL standard deviation results ($\times 10^{-3}$) for Llama2-7B and its pruned version on Race-H.

Dataset	#TP	#FN	#FP	#TN
C3	543	257	210	815
CHID	269	563	177	993
Race-M	380	95	129	832
Race-H	938	305	353	1902

Model	TP	FN	FP	TN
Llama2-7B	1.12	0.87	0.94	1.02
w/ pruning	1.13	0.84	0.88	0.92

We can observe that the perplexity standard deviation of TP and TN samples is significantly higher than that of FN and FP samples. This indicates that the model is more uncertain about its predictions for FN and FP samples. Additionally, FP samples account for a considerable proportion of the total samples, suggesting that the pruned model may guess and correctly answer a significant portion of the samples. This phenomenon indicates that the accuracy metric may overestimate the performance of pruned models.

Therefore, we define stability to better evaluate the performance of pruned models. Stability takes into account both the model's confidence in its predictions and the consistency of the model's answers before and after pruning:

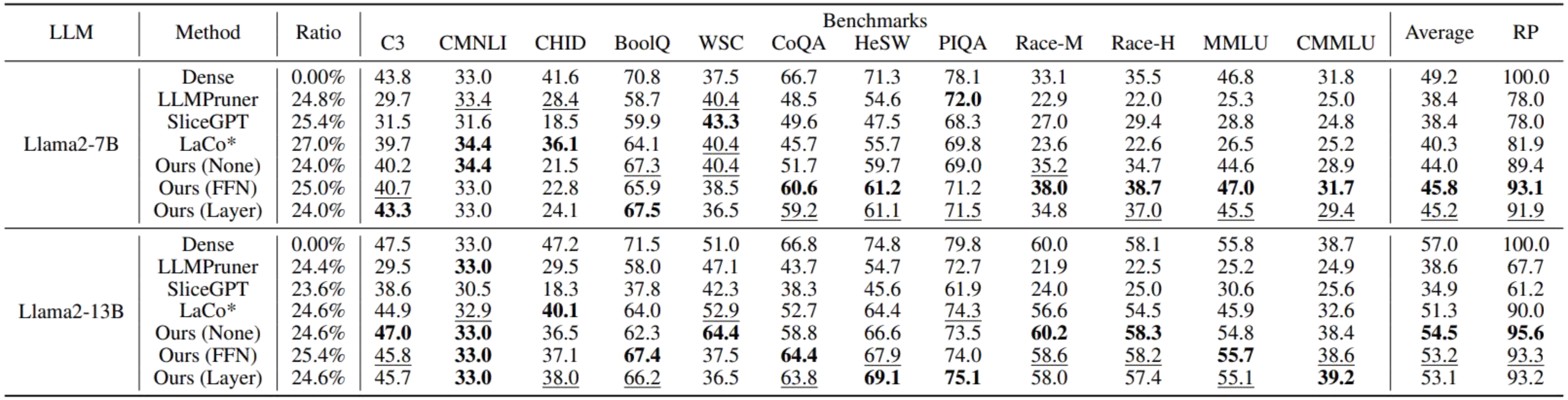
$$\text{Stability}(\mathcal{M}, \bar{\mathcal{M}}) = \frac{\sum_{i=1}^{N} \left(\exp\left(\text{std}_i\right) \cdot \mathbb{1}_{[i \in \text{TP} \cup \text{TN}]}\right)}{\sum_{i=1}^{N} \exp\left(\text{std}_j\right)}$$

Experiment

LLM: Llama2-7B、Llama2-13B、OPT-1.3B、OPT-2.7B、OPT-6.7B、Llama3.1-8B、Llama3.1-70B、Mixtral-8x7B-v0.1

Concurrent methods:
ShortGPT: ShortGPT is a subset of our method, Equivalent to Ours(None)
UIDL: ShortGPT + LoRA
Shortened Llama、SLEB: Based on perplexity, this method is not as effective as cosine similarity based methods

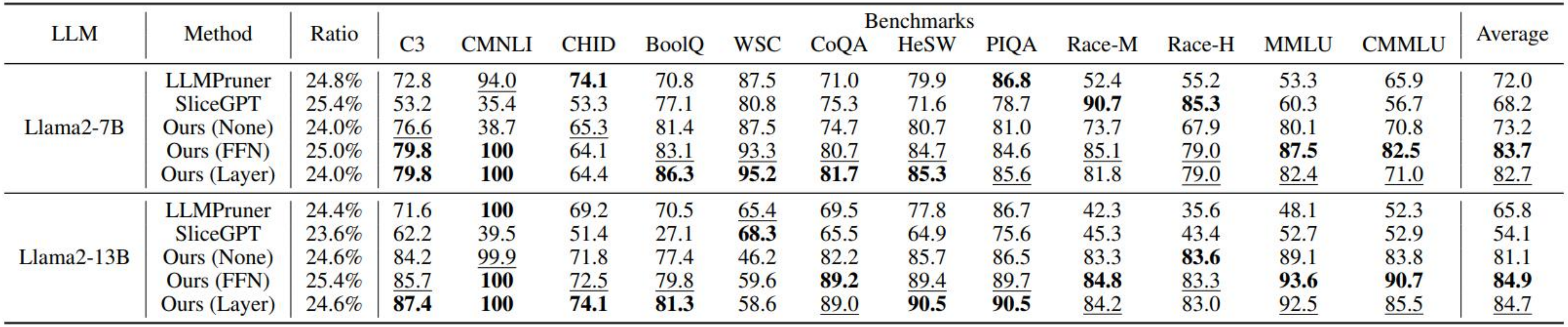
Table 2: Accuracy of pruning methods on classification benchmarks. "*" indicates that we refer to the results in the original paper. Retained performance (RP) represents the percentage of the original model's performance retained by the pruning method.



LLM	Method	Ratio	Benchmarks												Average	RP
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU		
Llama2-7B	Dense	0.00%	43.8	33.0	41.6	70.8	37.5	66.7	71.3	78.1	33.1	35.5	46.8	31.8	49.2	100.0
	LLMPruner	24.8%	29.7	33.4	28.4	58.7	40.4	48.5	54.6	**72.0**	22.9	22.0	25.3	25.0	38.4	78.0
	SliceGPT	25.4%	31.5	31.6	18.5	59.9	**43.3**	49.6	47.5	68.3	27.0	29.4	28.8	24.8	38.4	78.0
	LaCo*	27.0%	39.7	**34.4**	**36.1**	64.1	40.4	45.7	55.7	69.8	23.6	22.6	26.5	25.2	40.3	81.9
	Ours (None)	24.0%	40.2	**34.4**	21.5	67.3	40.4	51.7	59.7	69.0	35.2	34.7	44.6	28.9	44.0	89.4
	Ours (FFN)	25.0%	40.7	33.0	22.8	65.9	38.5	**60.6**	**61.2**	71.2	**38.0**	**38.7**	**47.0**	**31.7**	**45.8**	**93.1**
	Ours (Layer)	24.0%	**43.3**	33.0	24.1	**67.5**	36.5	59.2	61.1	71.5	34.8	37.0	45.5	29.4	45.2	91.9
Llama2-13B	Dense	0.00%	47.5	33.0	47.2	71.5	51.0	66.8	74.8	79.8	60.0	58.1	55.8	38.7	57.0	100.0
	LLMPruner	24.4%	29.5	**33.0**	29.5	58.0	47.1	43.7	54.7	72.7	21.9	22.5	25.2	24.9	38.6	67.7
	SliceGPT	23.6%	38.6	30.5	18.3	37.8	42.3	38.3	45.6	61.9	24.0	25.0	30.6	25.6	34.9	61.2
	LaCo*	24.6%	44.9	32.9	**40.1**	64.0	52.9	52.7	64.4	74.3	56.6	54.5	45.9	32.6	51.3	90.0
	Ours (None)	24.6%	**47.0**	**33.0**	36.5	62.3	**64.4**	58.8	66.6	73.5	**60.2**	**58.3**	54.8	38.4	**54.5**	**95.6**
	Ours (FFN)	25.4%	45.8	**33.0**	37.1	**67.4**	37.5	**64.4**	67.9	74.0	58.6	58.2	**55.7**	38.6	53.2	93.3
	Ours (Layer)	24.6%	45.7	**33.0**	38.0	66.2	36.5	63.8	**69.1**	**75.1**	58.0	57.4	55.1	**39.2**	53.1	93.2

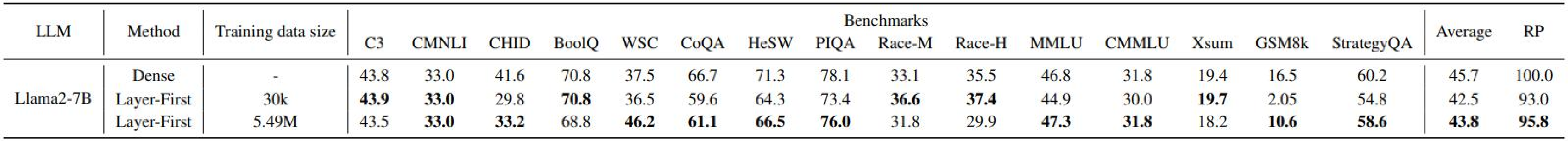
Experiment

Table 3: Stability of pruning methods on classification benchmarks. The stability of the original model is 1.0, because stability is measured by comparing the prediction results of the original model.



LLM	Method	Ratio	Benchmarks												Average
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU	
Llama2-7B	LLMPruner	24.8%	72.8	94.0	**74.1**	70.8	87.5	71.0	79.9	**86.8**	52.4	55.2	53.3	65.9	72.0
	SliceGPT	25.4%	53.2	35.4	53.3	77.1	80.8	75.3	71.6	78.7	**90.7**	**85.3**	60.3	56.7	68.2
	Ours (None)	24.0%	76.6	38.7	65.3	81.4	87.5	74.7	80.7	81.0	73.7	67.9	80.1	70.8	73.2
	Ours (FFN)	25.0%	**79.8**	**100**	64.1	83.1	93.3	80.7	84.7	84.6	85.1	79.0	**87.5**	**82.5**	**83.7**
	Ours (Layer)	24.0%	**79.8**	**100**	64.4	**86.3**	**95.2**	**81.7**	**85.3**	85.6	81.8	79.0	82.4	71.0	82.7
Llama2-13B	LLMPruner	24.4%	71.6	**100**	69.2	70.5	65.4	69.5	77.8	86.7	42.3	35.6	48.1	52.3	65.8
	SliceGPT	23.6%	62.2	39.5	51.4	27.1	**68.3**	65.5	64.9	75.6	45.3	43.4	52.7	52.9	54.1
	Ours (None)	24.6%	84.2	99.9	71.8	77.4	46.2	82.2	85.7	86.5	83.3	**83.6**	89.1	83.8	81.1
	Ours (FFN)	25.4%	85.7	**100**	72.5	79.8	59.6	**89.2**	89.4	89.7	**84.8**	83.3	**93.6**	**90.7**	**84.9**
	Ours (Layer)	24.6%	**87.4**	**100**	**74.1**	**81.3**	58.6	89.0	**90.5**	**90.5**	84.2	83.0	92.5	85.5	84.7

Table 28: Detailed accuracy results with different training data volumes.



LLM	Method	Training data size	Benchmarks															Average	RP
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU	Xsum	GSM8k	StrategyQA		
Llama2-7B	Dense	-	43.8	33.0	41.6	70.8	37.5	66.7	71.3	78.1	33.1	35.5	46.8	31.8	19.4	16.5	60.2	45.7	100.0
	Layer-First	30k	**43.9**	**33.0**	29.8	**70.8**	36.5	59.6	64.3	73.4	**36.6**	**37.4**	44.9	30.0	**19.7**	2.05	54.8	42.5	93.0
	Layer-First	5.49M	43.5	**33.0**	**33.2**	68.8	**46.2**	**61.1**	**66.5**	**76.0**	31.8	29.9	**47.3**	**31.8**	18.2	**10.6**	**58.6**	**43.8**	**95.8**

Experiment

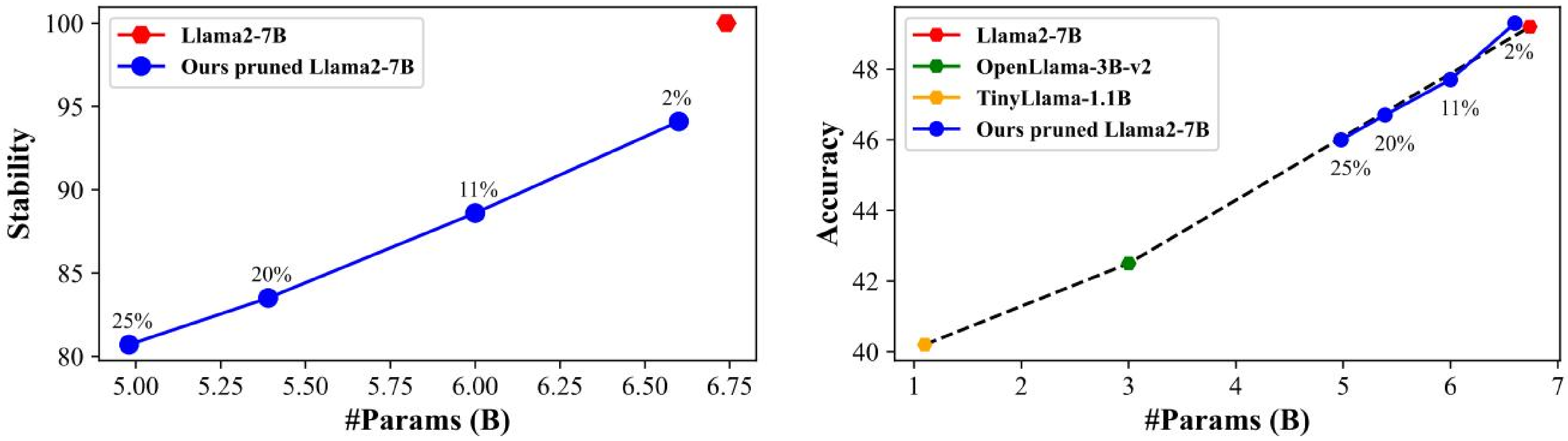


Figure 4: (a) Stability of the pruned Llama2-7B at different pruning ratios. (b) Accuracy of the pruned Llama2-7B at different pruning ratios, compared to the original Llama2-7B, OpenLlama-3B-v2, and TinyLlama-1.1B. Metrics are averaged across classification benchmarks.

Experiment

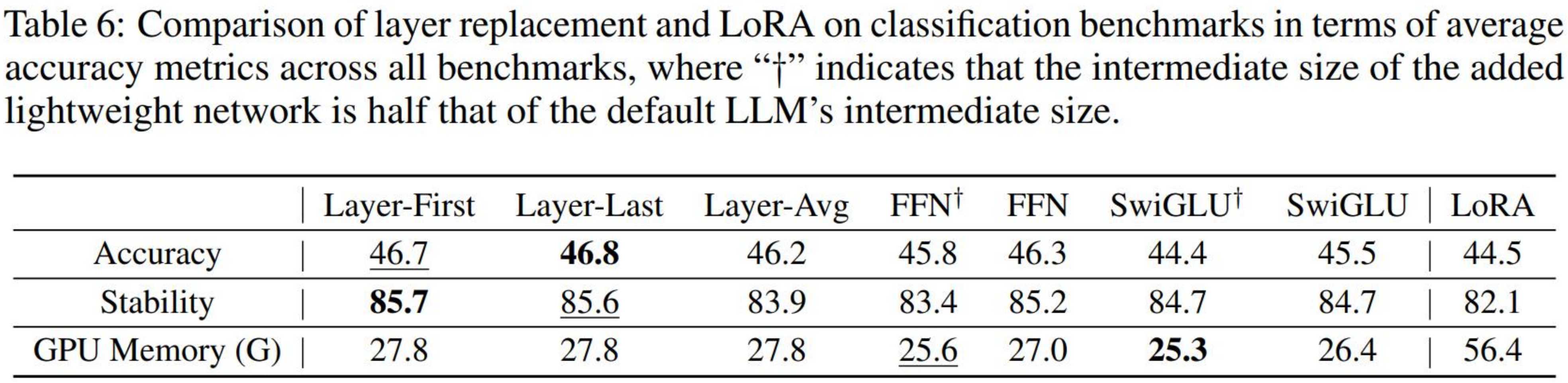


Table 6: Comparison of layer replacement and LoRA on classification benchmarks in terms of average accuracy metrics across all benchmarks, where “†” indicates that the intermediate size of the added lightweight network is half that of the default LLM’s intermediate size.

	Layer-First	Layer-Last	Layer-Avg	FFN†	FFN	SwiGLU†	SwiGLU	LoRA
Accuracy	46.7	**46.8**	46.2	45.8	46.3	44.4	45.5	44.5
Stability	**85.7**	85.6	83.9	83.4	85.2	84.7	84.7	82.1
GPU Memory (G)	27.8	27.8	27.8	25.6	27.0	**25.3**	26.4	56.4

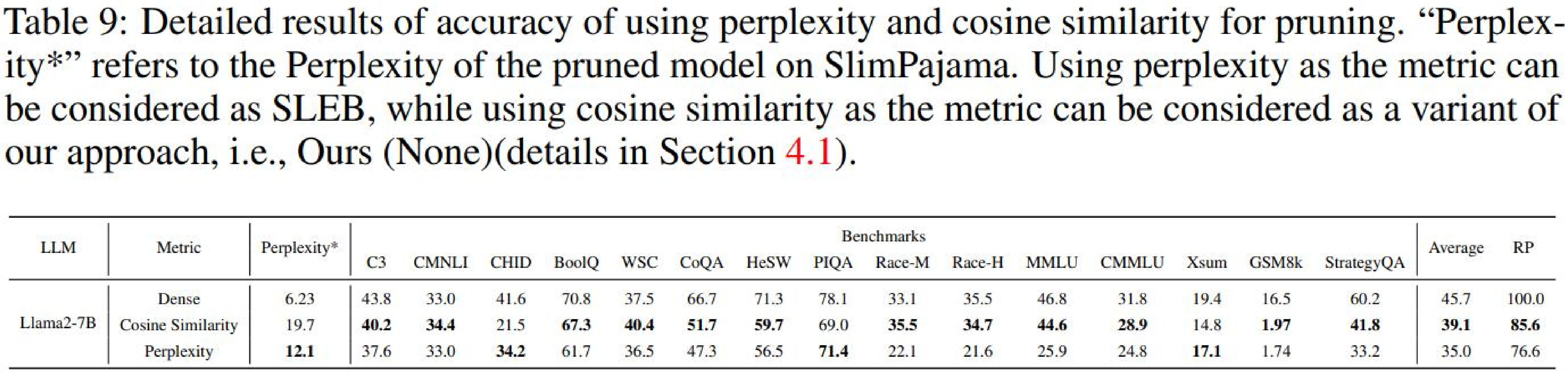
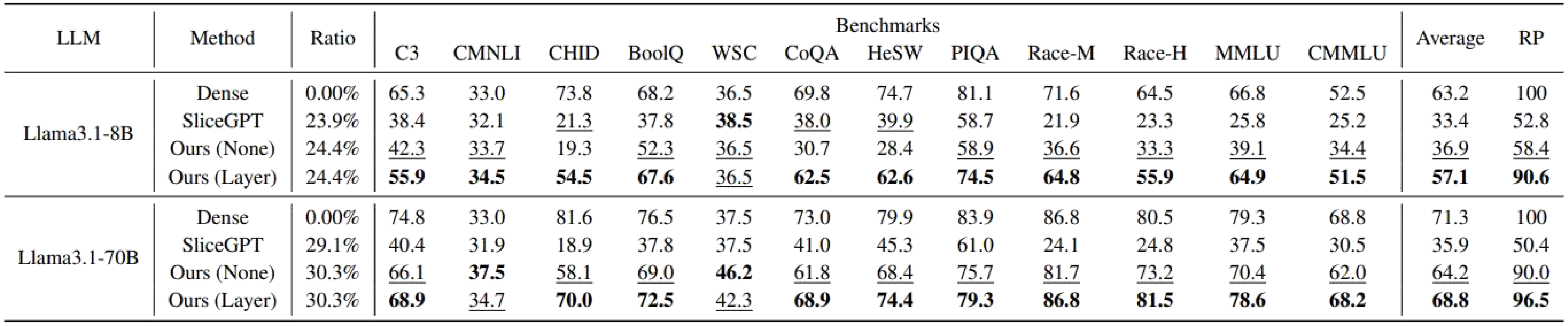


Table 9: Detailed results of accuracy of using perplexity and cosine similarity for pruning. “Perplexity*” refers to the Perplexity of the pruned model on SlimPajama. Using perplexity as the metric can be considered as SLEB, while using cosine similarity as the metric can be considered as a variant of our approach, i.e., Ours (None)(details in Section 4.1).

LLM	Metric	Perplexity*	Benchmarks															Average	RP
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU	Xsum	GSM8k	StrategyQA		
Llama2-7B	Dense	6.23	43.8	33.0	41.6	70.8	37.5	66.7	71.3	78.1	33.1	35.5	46.8	31.8	19.4	16.5	60.2	45.7	100.0
	Cosine Similarity	19.7	**40.2**	**34.4**	21.5	**67.3**	**40.4**	**51.7**	**59.7**	69.0	**35.5**	**34.7**	**44.6**	**28.9**	14.8	**1.97**	**41.8**	**39.1**	**85.6**
	Perplexity	**12.1**	37.6	33.0	**34.2**	61.7	36.5	47.3	56.5	**71.4**	22.1	21.6	25.9	24.8	**17.1**	1.74	33.2	35.0	76.6

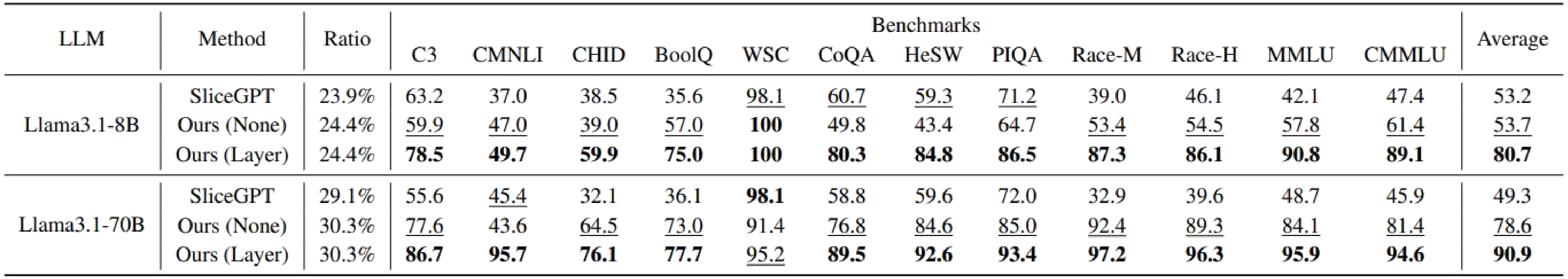
Experiment

Table 21: Accuracy of different pruning methods on classification benchmarks by pruning Llama3.1-8B and Llama3.1-70B.



LLM	Method	Ratio	Benchmarks												Average	RP
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU		
Llama3.1-8B	Dense	0.00%	65.3	33.0	73.8	68.2	36.5	69.8	74.7	81.1	71.6	64.5	66.8	52.5	63.2	100
	SliceGPT	23.9%	38.4	32.1	21.3	37.8	**38.5**	38.0	39.9	58.7	21.9	23.3	25.8	25.2	33.4	52.8
	Ours (None)	24.4%	42.3	33.7	19.3	52.3	36.5	30.7	28.4	58.9	36.6	33.3	39.1	34.4	36.9	58.4
	Ours (Layer)	24.4%	**55.9**	**34.5**	**54.5**	**67.6**	36.5	**62.5**	**62.6**	**74.5**	**64.8**	**55.9**	**64.9**	**51.5**	**57.1**	**90.6**
Llama3.1-70B	Dense	0.00%	74.8	33.0	81.6	76.5	37.5	73.0	79.9	83.9	86.8	80.5	79.3	68.8	71.3	100
	SliceGPT	29.1%	40.4	31.9	18.9	37.8	37.5	41.0	45.3	61.0	24.1	24.8	37.5	30.5	35.9	50.4
	Ours (None)	30.3%	66.1	**37.5**	58.1	69.0	**46.2**	61.8	68.4	75.7	81.7	73.2	70.4	62.0	64.2	90.0
	Ours (Layer)	30.3%	**68.9**	34.7	**70.0**	**72.5**	42.3	**68.9**	**74.4**	**79.3**	**86.8**	**81.5**	**78.6**	**68.2**	**68.8**	**96.5**

Table 22: Stability of different pruning methods on classification benchmarks by pruning Llama3.1-8B and Llama3.1-70B.



LLM	Method	Ratio	Benchmarks												Average
			C3	CMNLI	CHID	BoolQ	WSC	CoQA	HeSW	PIQA	Race-M	Race-H	MMLU	CMMLU	
Llama3.1-8B	SliceGPT	23.9%	63.2	37.0	38.5	35.6	98.1	60.7	59.3	71.2	39.0	46.1	42.1	47.4	53.2
	Ours (None)	24.4%	59.9	47.0	39.0	57.0	**100**	49.8	43.4	64.7	53.4	54.5	57.8	61.4	53.7
	Ours (Layer)	24.4%	**78.5**	**49.7**	**59.9**	**75.0**	**100**	**80.3**	**84.8**	**86.5**	**87.3**	**86.1**	**90.8**	**89.1**	**80.7**
Llama3.1-70B	SliceGPT	29.1%	55.6	45.4	32.1	36.1	**98.1**	58.8	59.6	72.0	32.9	39.6	48.7	45.9	49.3
	Ours (None)	30.3%	77.6	43.6	64.5	73.0	91.4	76.8	84.6	85.0	92.4	89.3	84.1	81.4	78.6
	Ours (Layer)	30.3%	**86.7**	**95.7**	**76.1**	**77.7**	95.2	**89.5**	**92.6**	**93.4**	**97.2**	**96.3**	**95.9**	**94.6**	**90.9**

Conclusion

In this paper, we propose LLM-Streamline, a layer pruning-and-replacement algorithm for LLMs. Extensive experiments show that this layer replacement method using a lightweight network outperforms previous state-of-the-art pruning methods and demonstrates superior effectiveness and efficiency compared to concurrent layer pruning methods.

Paper Link:
https://arxiv.org/pdf/2403.19135
Github Link:
https://github.com/RUCKBReasoning/LLM-Streamline
Open source model Link:
https://huggingface.co/XiaodongChen/Llama-2-4.7B
https://huggingface.co/XiaodongChen/Llama-3.1-5.4B