Fluctuation-Based Adaptive Structured Pruning for Large Language Models

Yongqi An[1, 2], **Xu Zhao**[1, 4, *], **Tao Yu**[1, 2], **Ming Tang**[1, 2], **Jinqiao Wang**[1, 2, 3, 4]

[1]Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of artificial intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Wuhan AI Research, Wuhan, China
[4]Objecteye Inc., Beijing, China
{yongqi.an, xu.zhao, tangm, jqwang}@nlpr.ia.ac.cn, yutao2022@ia.ac.cn

Abstract

Network Pruning is a promising way to address the huge computing resource demands of the deployment and inference of Large Language Models (LLMs). Retraining-free is important for LLMs' pruning methods. However, almost all of the existing retraining-free pruning approaches for LLMs focus on unstructured pruning, which requires specific hardware support for acceleration. In this paper, we propose a novel retraining-free structured pruning framework for LLMs, named FLAP (**FL**uctuation-based **A**daptive Structured **P**runing). It is hardware-friendly by effectively reducing storage and enhancing inference speed. For effective structured pruning of LLMs, we highlight three critical elements that demand the utmost attention: formulating structured importance metrics, adaptively searching the global compressed model, and implementing compensation mechanisms to mitigate performance loss. First, FLAP determines whether the output feature map is easily recoverable when a column of weight is removed, based on the fluctuation pruning metric. Then it standardizes the importance scores to adaptively determine the global compressed model structure. At last, FLAP adds additional bias terms to recover the output feature maps using the baseline values. We thoroughly evaluate our approach on a variety of language benchmarks. Without any retraining, our method significantly outperforms the state-of-the-art methods, including LLM-Pruner and the extension of Wanda in structured pruning. The code is released at https://github.com/CASIA-IVA-Lab/FLAP.

Introduction

Large Language Models (LLMs) (Brown et al. 2020; Touvron et al. 2023; Zhang et al. 2022; Scao et al. 2022) have recently achieved outstanding performance across various language benchmarks in NLP (Bommarito and Katz 2022; Bubeck et al. 2023; Wei et al. 2022), spurring a large number of open-source applications (Taori et al. 2023; Anand et al. 2023; Richards 2023). These remarkable capabilities typically come with a huge-scale model size with high inference costs. This makes it harder for more people to benefit from LLMs. Due to the computational resource constraints, most of the model compression methods in the pre-LLM era are no longer feasible for LLMs. Model compression methods for LLMs to date focus on model quantization (Dettmers et al. 2022; Xiao et al. 2023; Frantar et al. 2023; Dettmers et al. 2023) and unstructured pruning (Sun et al. 2023; Frantar and Alistarh 2023).

Structured pruning (He and Xiao 2023), which prunes entire rows or columns of weights, offers a promising solution to the deployment challenges of LLMs. Unlike unstructured pruning, structured pruning reduces both parameters and inference time without relying on specific hardware, making it more widely applicable (Anwar, Hwang, and Sung 2017). For effective structured pruning, it's crucial to have a metric that captures the collective significance of an entire row or column. However, current unstructured pruning techniques for LLMs, as seen in methods like (Sun et al. 2023; Frantar and Alistarh 2023), primarily focus on the importance of individual elements of each row in isolation. This absence of structured metrics that evaluate entire rows or columns makes them less suitable for structured pruning. The recent LLM-Pruner (Ma, Fang, and Wang 2023) attempted structured pruning for LLMs, but its dependence on LoRA finetuning (Hu et al. 2021) creates a tough trade-off between high computation and effective pruning, limiting its use in larger models.

Pruning essentially involves two key aspects: discovering redundancy and recovering performance. For an effective structured pruning method tailored to LLMs, three fundamental criteria must be satisfied: a) a structured importance metric to discover structured redundancy; b) a mechanism for adaptively searching the optimal global compression model structure; and c) a compensation strategy to minimize performance degradation.

In response to these three essential criteria, we introduce FLAP (**FL**uctuation-based **A**daptive Structured **P**runing), a novel structured pruning framework. We find that certain channels of hidden state features exhibit structured sample stability. This observation enables us to compensate for bias within the model using baseline values. Specifically, we design a structured pruning metric that estimates the fluctuation of each input feature relative to the baseline value, utilizing a set of calibration samples. This metric assists in determining whether the output feature map can be recovered when a column of the weight matrix is removed. We then standardize these fluctuation metric scores across lay-

*Corresponding Author

ers and modules separately, allowing for the adaptive determination of the global compressed model structure. Finally, FLAP employs the baseline values to add additional biases, recovering the output feature maps for the corresponding layers. Remarkably, our method avoids the need for the retraining process and requires only a single forward pass for both pruning and bias compensation, thereby maintaining low memory overhead.

We evaluate the effectiveness of FLAP on the LLaMA model family, and FLAP achieves remarkable performance on a variety of language benchmarks. Impressively, without any retraining, our method significantly outperforms the state-of-the-art methods, including LLM-Pruner and the extension of Wanda in structured pruning.

Our main contributions are listed as follows:

- We propose a novel retraining-free structured pruning framework for LLMs named FLAP. To our best knowledge, this is the first work that identifies the characteristic of structured sample stability in LLMs.
- The proposed framework uses a bias compensation mechanism, a pruning performance recovery method that does not require retraining. This mechanism yields greater benefits, especially under large pruning ratios.
- Our method achieves remarkable performance on a variety of language benchmarks and outperforms the state-of-the-art method without any retraining.

Related Works

Network Pruning Methods

Network pruning is a model compression technique that identifies and eliminates redundancy in the structure or parameters of a neural network, based on specific pruning metrics, and incorporates methods to recover model performance (LeCun, Denker, and Solla 1989; Hassibi, Stork, and Wolff 1993; Han et al. 2015). Pruning methods fall into two categories: unstructured pruning and structured pruning. Unstructured pruning is performed at the individual weight level, allowing for a large sparsity but failing to achieve real inference acceleration or storage reduction (Zafrir et al. 2021; Han, Mao, and Dally 2016). Within unstructured pruning, there exists a specialized variant known as semi-structured pruning. This approach enforces exactly N non-zero values in each block of M consecutive weights (Zhou et al. 2021). This approach has gained traction recently, particularly with support on newer NVIDIA hardware (Mishra et al. 2021). Structured pruning, by contrast, operates on entire rows or columns of weights, providing a more hardware-friendly solution that reduces storage requirements and enhances inference speed (Xia, Zhong, and Chen 2022; Molchanov et al. 2017).

However, conventional structured pruning methods typically rely on retraining (sometimes iteratively) to regain the performance of the pruned model (Han et al. 2015; Tan and Motani 2020; Han, Mao, and Dally 2016). Such methods pose scalability challenges for billion-scale LLMs due to constraints on memory and computational resources. Therefore a retraining-free structured pruning method for LLMs is very critical.

Large Language Model Compression

Large Language Models usually consist of billions of parameters, and their gradient backpropagation and training stage require large amounts of memory and computational resources. Consequently, many conventional model compression techniques have become infeasible for LLMs (Frantar and Alistarh 2023). For instance, knowledge distillation (Hinton, Vinyals, and Dean 2015), once a practical approach, now faces implementation challenges due to high training costs. Existing compression methods for LLMs mainly include post-training quantization (Dettmers et al. 2022; Xiao et al. 2023; Frantar et al. 2023; Dettmers et al. 2023) and post-training pruning (Sun et al. 2023; Frantar and Alistarh 2023). Our method also falls into the category of post-training pruning. It utilizes bias compensation to recover model performance, effectively avoiding the high computational cost of retraining. Unlike the past post-training pruning methods, our method is designed for the features of structured pruning of LLMs.

Properties of LLMs

Our work is related to the distinct properties of Large Language Models (LLMs) that have inspired various model compression techniques (Sun et al. 2023; Dettmers et al. 2023, 2022). Dettmers et al. (Dettmers et al. 2022) observed the emergence of channels with abnormally large magnitudes in the hidden state features of LLMs once they exceed a certain parameter scale (e.g., 6B). They suggest that this is the reason why existing quantization methods fail on LLMs. In response, they introduced a novel mixed-precision quantization technique. Contrary to the focus of previous work on the outlier magnitudes in LLMs, our research pivots towards investigating the structured stability within the channels of input features in these models. In our study, we find that certain channels within the hidden state features demonstrate consistent structured sample stability. This discovery offers invaluable insights for crafting structured post-training pruning algorithms, laying the foundation for the method we present in this paper.

Preliminaries

Layer-Wise Pruning

Given the computational constraints, globally solving the pruning problem for Large Language Models (LLMs) is challenging. Layer-wise pruning becomes a practical solution under these constraints. Following this notion, SparseGPT (Frantar and Alistarh 2023) demonstrated that the challenge of unstructured pruning for LLMs can be tackled by decomposing it into individual layer-wise subproblems. This principle can be seamlessly extended to structured pruning within LLMs. The quality of solutions to these layer-wise subproblems can be evaluated based on the ℓ_2-error. Given an input $\mathbf{X}^\ell$ of shape (N, C_{in}, L) where N and L represent batch and sequence dimensions respectively, and a weight $\mathbf{W}^\ell$ of shape (C_{out}, C_{in}), the ℓ_2-error for structured pruning can be defined as:

$$\text{argmin}_{\mathbf{M}^\ell \in \mathbb{R}^{C_{in}}, \widehat{\mathbf{W}}^\ell} ||\mathbf{W}^\ell \mathbf{X}^\ell - (\mathbf{M}^\ell \odot \widehat{\mathbf{W}}^\ell)\mathbf{X}^\ell||_2^2 \quad (1)$$

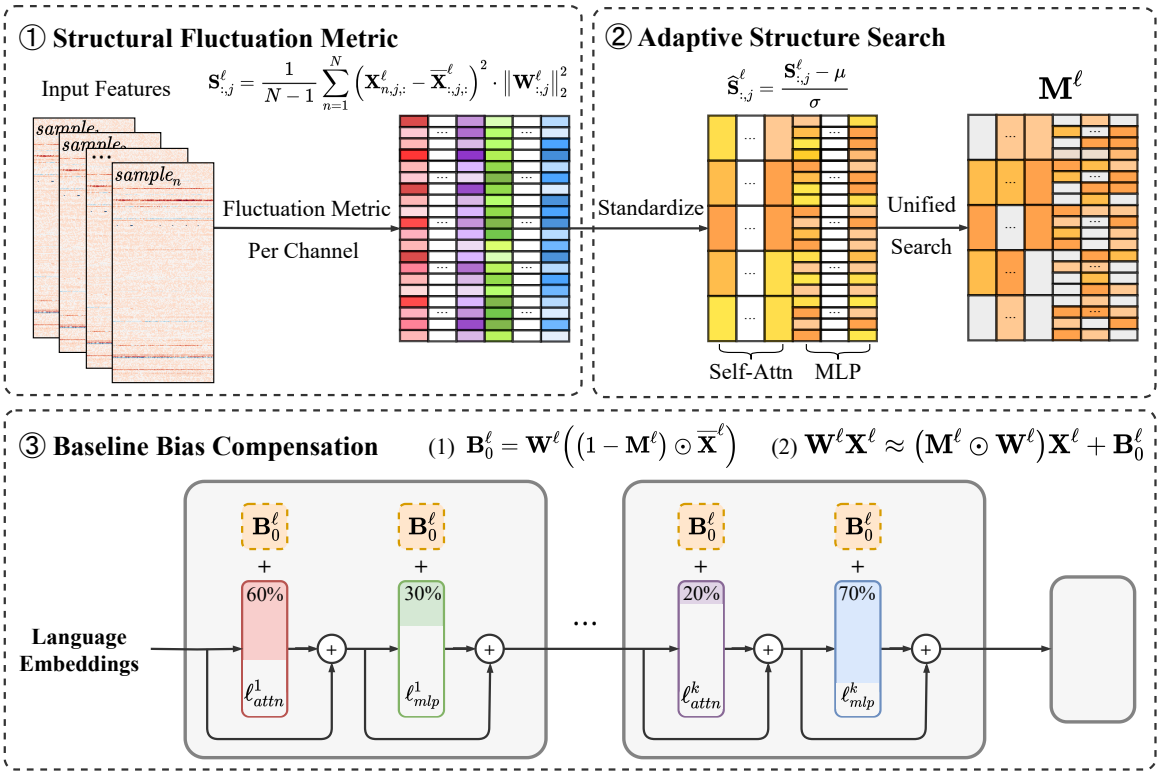

Figure 1: Framework of the proposed FLAP. ①Measure the fluctuation of each channel across different layers and modules using calibration data; ②Standardize these fluctuation measures for a unified search method; ③Implement adaptive pruning ratios for each layer and module, employing bias compensation to restore model performance.

where $\mathbf{M}^{\ell} \in \mathbb{R}^{C_{in}}$ represents the mask vector corresponding to the input channels, this vector mirrors whether each input channel is pruned or not. For the self-attention modules, these input channels are pruned in groups typically with sizes like group_size=128. The term $\widehat{\mathbf{W}}^{\ell}$ denotes the possibly updated weights for the pruned layer. The notation $||\cdot||_2^2$ represents the ℓ_2-error.

Local Pruning Metric Challenges

Regarding Eq. (1), the existing methods can be broadly categorized into two primary approaches: low-damage and easy-recoverability. These correspond to the core principles of OBD (LeCun, Denker, and Solla 1989) and OBS (Hassibi, Stork, and Wolff 1993), respectively. To illustrate, Wanda (Sun et al. 2023) uses a localized low-damage pruning metric to minimize harm to each layer's output features. In contrast, SparseGPT (Frantar and Alistarh 2023) employs an easy-recoverability metric, aiming to identify components that other weights can compensate for during pruning. These approaches are insightful but tend to focus on the importance of individual elements in the weight matrix, neglecting the broader structured context. Such an atomistic approach is misaligned with structured pruning's requirements, which demand a more global perspective that captures the collective importance of entire rows or columns in the matrix.

Methodology

In this section, we introduce FLAP, our proposed approach to structured pruning for Large Language Models (LLMs). FLAP encompasses three key components: Baseline Bias Compensation, Structured Fluctuation Metric, and Adaptive Structure Search. The overview of our method is presented in Figure 1.

Baseline Bias Compensation

In the context of structured pruning, the output of the layers of the uncompressed model can be decomposed into:

$$\mathbf{W}^{\ell}\mathbf{X}^{\ell} = \underbrace{(\mathbf{M}^{\ell} \odot \mathbf{W}^{\ell})\mathbf{X}^{\ell}}_{\text{retained}} + \underbrace{((1-\mathbf{M}^{\ell}) \odot \mathbf{W}^{\ell})\mathbf{X}^{\ell}}_{\text{removed}} \quad (2)$$

The objective of structured pruning is to minimize the impact introduced by $\Delta Y^{\ell} = ((1-\mathbf{M}^{\ell}) \odot \mathbf{W}^{\ell})\mathbf{X}^{\ell}$ in the overall output feature map, thereby reducing the reconstruction error for each layer. For structured pruning of LLMs, the constraints are stronger, so the latter components cannot be simply removed. Therefore, a compensatory mechanism is essential to recover the model's performance while adhering to the pruning structure.

We add an additional bias term to compensate for the damage inflicted on the output feature maps by the removed components. This bias term is designed to mitigate the reconstruction error introduced by the pruning process, allowing the pruned model to maintain high performance. In

particular, we construct the bias term based on the baseline value, $\overline{\mathbf{X}}^{\ell}_{:,j,:}$, which represents the average of the j-th channel for all samples in layer l. As detailed in the following section, our empirical findings validate the effectiveness and feasibility of this compensatory approach. Specifically, the formulation for the baseline value is as follows:

$$\overline{\mathbf{X}}^{\ell}_{:,j,:} = \frac{1}{NL}\sum_{n=1}^{N}\sum_{k=1}^{L}\mathbf{X}^{\ell}_{n,j,k} \tag{3}$$

Once the mask $\mathbf{M}^{\ell}$ is established, the baseline value for the pruned channel can be seamlessly translated into the bias term for the linear layer as follows:

$$\begin{aligned}\mathbf{B}^{\ell}_{0} &= \mathbf{W}^{\ell}((1-\mathbf{M}^{\ell})\odot\overline{\mathbf{X}}^{\ell}) \\ \mathbf{W}^{\ell}\mathbf{X}^{\ell} &\approx (\mathbf{M}^{\ell}\odot\mathbf{W}^{\ell})\mathbf{X}^{\ell}+\mathbf{B}^{\ell}_{0}\end{aligned} \tag{4}$$

where $\mathbf{B}_0$ represents the bias of linear layer, which has a shape of $(C_{out},)$, and $\overline{\mathbf{X}}^{\ell}$ is a one-dimensional vector with dimensions $(C_{in},)$.

Structured Fluctuation Metric

Motivated by the observations from Figure 2, we note that certain channels of the hidden state features exhibit a low variation across different samples. This low fluctuation indicates that if their corresponding input feature channels are pruned, the resulted change in the output feature map can be effectively counterbalanced by the baseline value.

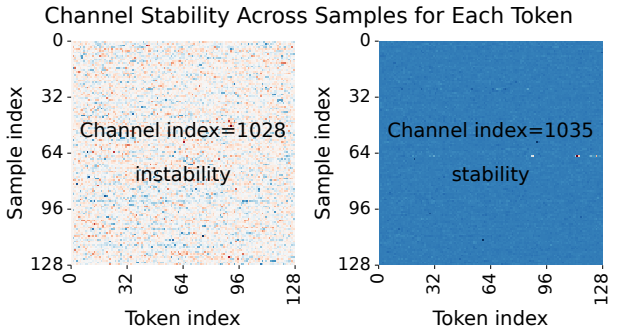

Figure 2: Certain channels of hidden state features exhibit structured sample stability. The left shows a channel with noticeable variations across samples, indicating low stability. The right displays a stable pattern common in many LLaMa channels.

As illustrated in Eq. (4), the structured easy-recoverability metric seeks to evaluate the impact on the output feature map when an input channel is substituted with its baseline value. A straightforward approach would involve individually substituting each input channel with its baseline value for the calibration samples and then computing the ℓ_2-error between the output feature maps before and after this replacement.

However, such a method poses a significant computational challenge and is impractical for LLMs. To address this, we introduce an approximate metric for structured recoverability, which termed the "fluctuation metric". Specifically, we compute the sample variance of each input feature and weight it with the squared norm of the corresponding column of the weight matrix. Concretely, the score for the group of weight $\mathbf{W}^{\ell}_{:,j}$ is defined by:

$$\mathbf{S}^{\ell}_{:,j} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{X}^{\ell}_{n,j,:}-\overline{\mathbf{X}}^{\ell}_{:,j,:})^2\cdot||\mathbf{W}^{\ell}_{:,j}||_2^2 \tag{5}$$

where $||\mathbf{W}^{\ell}_{:,j}||_2^2$ denotes the squared norm of j-th column of the weight matrix. $\frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{X}^{\ell}_{n,j,:}-\overline{\mathbf{X}}^{\ell}_{:,j,:})^2$ represents the sample variance of the j-th channel of the input feature of layer ℓ under N calibration samples. The denominator here is $\frac{1}{N-1}$. This correction is known as the Bessel correction and is used for unbiased estimation of the overall variance.

Adaptive Structure Search

The central challenge in layer-wise pruning revolves around adaptively searching the global compression model structures. Unifying different layers and modules without distinction can critically degrade performance. This issue arises because the magnitudes of the metrics across layers and modules vary greatly (Shi et al. 2023). Figure 3 demonstrates this by showing the mean values of the fluctuation metric for different modules in different layers.

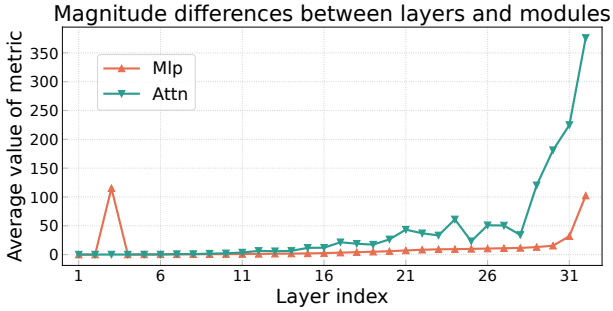

Figure 3: Comparison of the average value of the fluctuation metric across different layers for different modules.

To ensure a consistent comparison of scores across different layers and modules, we standardize the metric distributions for each layer to a common mean and standard deviation. As defined in Eq. (5), the fluctuation metric captures the absolute variation in the output feature map when input features are replaced with their baseline values. In contrast, the standardized metric reflects the relative variation in the output feature map resulting from this replacement, making it suitable for a structured unified search. The standardized metric, denoted as, is formulated as follows:

$$\widehat{\mathbf{S}}^{\ell}_{:,j} = (\mathbf{S}^{\ell}_{:,j}-\mathbb{E}[\mathbf{S}^{\ell}_{:,j}])/(\mathbb{E}[[\mathbf{S}^{\ell}_{:,j}-\mathbb{E}[\mathbf{S}^{\ell}_{:,j}]]^2])^{\frac{1}{2}} \tag{6}$$

where $\mathbb{E}[\mathbf{S}^{\ell}_{:,j}]$ represents the expected value (or mean) of the vector $\mathbf{S}^{\ell}_{:,j}$. $(\mathbb{E}[[\mathbf{S}^{\ell}_{:,j}-\mathbb{E}[\mathbf{S}^{\ell}_{:,j}]]^2])^{\frac{1}{2}}$ represents the square root of the variance, which is the standard deviation.

Experiments

Experimental Settings

We conduct experiments on the LLaMA model family (LLaMA-7B/13B/30B/65B) to evaluate the efficacy of FLAP . Our evaluation focuses on language modeling performance on the WikiText2 (Merity et al. 2016) validation set and zero-shot performance across seven common sense benchmarks using the EleutherAI LM Harness (Gao et al. 2021)[1]. We compare FLAP against two previous pruning methods: Wanda-sp and LLM-Pruner. We generalize Wanda to structured pruning and name it as Wanda-sp. Detailed experimental settings, model descriptions, and evaluation protocols are provided in the Appendix A.

Method	Pruning Ratio	LLaMA			
		7B	13B	30B	65B
Dense	0%	12.62	10.81	9.11	8.21
Wanda-sp	20%	22.12	16.83	11.66	11.76
LLM-Pruner		19.77	16.01	-	-
LLM-Pruner*		17.37	15.18	-	-
FLAP (Ours)		**14.62**	**13.66**	**10.86**	**9.79**
Wanda-sp	30%	38.88	22.89	14.90	14.64
FLAP (Ours)		**17.62**	**15.65**	**12.49**	**10.90**
Wanda-sp	50%	366.43	160.49	67.46	42.85
LLM-Pruner		112.44	-	-	-
LLM-Pruner*		38.12	-	-	-
FLAP (Ours)		**31.80**	**24.20**	**19.36**	**15.30**

Table 1: WikiText2 validation perplexity of pruning methods for LLaMA model family. * means with LoRA fine-tuning.

Language Modeling

Performance Comparisons. For each of the LLaMA models, we present results at three distinct pruning ratios, as detailed in Table 1. Notably, FLAP significantly outperforms the other methods, achieving this superiority without any retraining. As the pruning ratio increases, the performance advantage of FLAP becomes more significant. To illustrate, consider the LLaMA-7B model: at a 50% pruning ratio, the LLM-Pruner exhibits a perplexity of 130.97, which improves to 39.02 after LoRA fine-tuning. In stark contrast, FLAP efficiently identifies sparse networks that yield a perplexity of 31.80, and remarkably, this is achieved without any retraining.

Remark. The FLAP method, which requires no retraining, consistently outperforms the LLM-Pruner, even when the latter is fine-tuned with LoRA. Eq (4) offers insight into the potential reason for this superior performance. In FLAP , the baseline bias $\mathbf{B}_0$ is effectively treated as a low-rank component with a rank of $r = 1$. Within the pruning framework of FLAP , bias compensation plays a pivotal role, serving a function similar to that of LoRA fine-tuning. This compensation helps to effectively recover the model's performance after pruning.

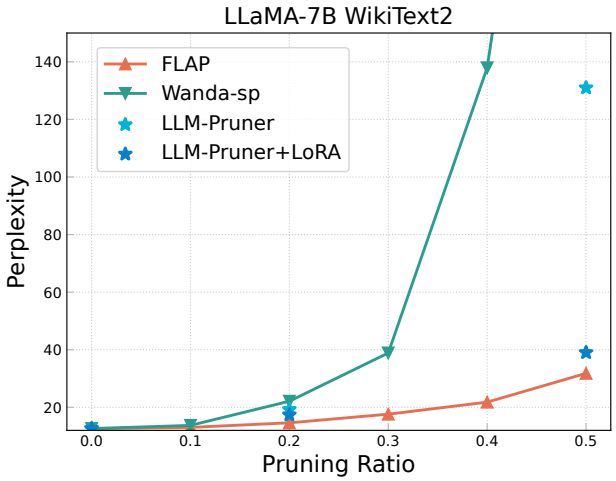

Figure 4: Results among FLAP and other structured pruning methods at varying pruning ratios on the LLaMA-7B WikiText2 dataset.

Different Pruning Ratio. We evaluated the performance of each structured pruning method at various pruning ratios. As depicted in Figure 4, FLAP demonstrates remarkable stability in maintaining its performance as the pruning ratio increases. In contrast, Wanda-sp exhibits a sharp decrease in performance as the pruning ratio rises. Meanwhile, LLM-Pruner requires LoRA fine-tuning to maintain acceptable performance when the pruning ratio is increased to levels like 50%.

Zero-shot Tasks Performance

We assessed the zero-shot capability of the pruned model across seven downstream tasks. As illustrated in Table 2, our method consistently outperforms LLM-Pruner with LoRA Fine-Tuning, achieving superior performance across varying pruning ratios, all without the need for retraining. At a 20% pruning ratio, Wanda-sp exhibits remarkable zero-shot capabilities, even surpassing the performance of the original, unpruned model. This suggests the presence of structured redundancy within LLMs that can be pruned away without necessitating retraining, thereby potentially enhancing model efficiency. However, when the pruning ratio is increased to 50%, the performance of Wanda-sp suffers a significant degradation. In stark contrast, our method continues to excel, maintaining a clear advantage over other approaches. This finding demonstrates the efficacy of our structured pruning method in preserving the generalization capabilities of large language models (LLMs), even under stringent pruning conditions.

Ablation Study

We systematically examine three fundamental components of the FLAP method: the pruning metric, the global compression structure, and bias compensation. Additionally, we evaluate the robustness of our pruning approach in relation to calibration samples.

[1]https://github.com/EleutherAI/lm-evaluation-harness

Method	Pruning Ratio	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
LLaMA-7B	0%	73.18	78.35	72.99	67.01	67.45	41.38	42.40	63.25
Wanda-sp	20%	**71.25**	**77.09**	**72.77**	67.09	**71.09**	**42.58**	**41.60**	**63.35**
LLM-Pruner		59.39	75.57	65.34	61.33	59.18	37.18	39.80	56.82
LLM-Pruner (w/ LoRA)		69.54	76.44	68.11	65.11	63.43	37.88	40.00	60.07
FLAP (Ours)		69.63	76.82	71.20	**68.35**	69.91	39.25	39.40	62.08
Wanda-sp	50%	50.58	55.01	29.56	51.78	31.27	23.04	23.60	37.83
LLM-Pruner		52.57	60.45	35.86	49.01	32.83	25.51	34.80	41.58
LLM-Pruner (w/ LoRA)		**61.47**	**68.82**	47.56	55.09	46.46	28.24	35.20	48.98
FLAP (Ours)		60.21	67.52	**52.14**	**57.54**	**49.66**	**29.95**	**35.60**	**50.37**

Table 2: Zero-shot performance of the compressed LLaMA-7B. Bold results highlight the best performance. Underscored results denote the second-best performance for each pruning ratio.

Pruning Metric. Both the pruning metric and compressed model structure are critical factors in the pruning process. FLAP is specifically designed to address these two dimensions in the structured pruning of Large Language Models (LLMs). To evaluate their effectiveness, we conducted experiments employing various structured pruning metrics and global compression structures.

We investigated three structured pruning metrics in this study: 1) Weighted Input Feature Norm (WIFN), a low-damage metric assessing the effect of weight columns on the output feature map; 2) Input Feature Variance (IFV), used to gauge the variability among input features; and 3) Weighted Input Feature Variance (WIFV), utilized by FLAP to assist in determining the potential for recovery of the output feature map after a column of the weight matrix is removed.

To underscore the importance of global adaptive compression structure, we defined four configurations: 'UL-UM' (Uniform across Layers and Modules, employed in unstructured pruning for LLMs like Wanda); 'UL-MM' (Uniform across Layers, Manual ratio for Modules); 'AL-MM' (Adaptive across Layers, Manual for Modules); and 'AL-AM' (Adaptive across both Layers and Modules), the structure chosen by FLAP . Results in this section include bias compensation, with bias-compensated ablation experiments detailed later.

In our experiments, we structurally pruned the LLaMA-7B model with a 50% pruning ratio and evaluated the model using the perplexity metric on the WikiText2 dataset. The detailed results are presented in Table 3. Notably, the most effective pruning model was obtained using the default configuration of FLAP , achieving a perplexity of 31.80. The AL-AM global adaptive compression structure consistently outperformed other configurations under all evaluated pruning metrics, thereby effectively validating our proposed Adaptive Structure Search strategy. When analyzing the effectiveness of different global compression structures, we observed that various metrics present distinct strengths and weaknesses. Nevertheless, our proposed WIFV structured pruning metric displayed superior adaptability to the global compression structure.

Baseline Bias Compensation. In structured pruning of large language models, restoring model performance after the pruning process is a crucial aspect. Our approach uniquely leverages bias compensation as a strategy to recover the performance of pruned models, circumventing the need for expensive and time-consuming retraining procedures. Figure 5 vividly illustrates the performance of the FLAP method on the WikiText2 dataset, comparing the perplexity scores with and without bias compensation at varying pruning ratios for the LLaMA-7B model. Evident from the figure, bias compensation plays a significant role in mitigating the performance degradation associated with pruning. Furthermore, this compensatory effect becomes more pronounced as the pruning ratio increases, highlighting the growing importance of bias compensation in more aggressively pruned models.

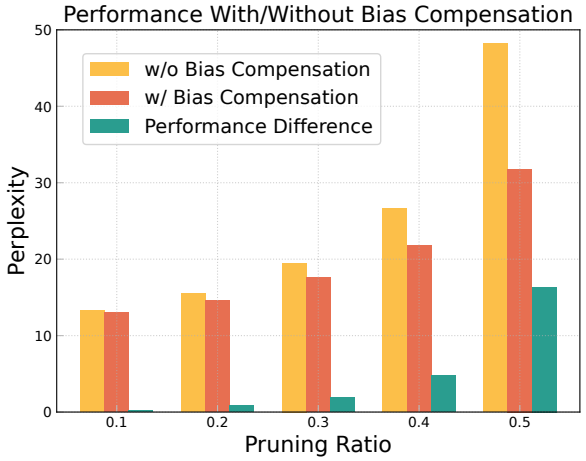

Figure 5: Performance comparison of the model with and without Bias Compensation at various pruning ratios. The yellow and orange bars represent the Perplexity of the model without and with Bias Compensation, respectively. The green bars show the performance difference between the two conditions.

Robustness to Calibration Samples. Our method utilizes a calibration dataset to estimate the input variance at each layer of the language model. This makes it critical to investigate the impact of the size of this calibration dataset on the pruning performance. Figure 6 delineates the effects of varying the number of calibration samples on the pruning outcome. For this analysis, we set a pruning ratio of 50% for the LLaMa-7B model and observed the resultant perplexity on the WikiText2 dataset. The results clearly show that

Pruning Metric	Compressed model structure			
	UL-UM	UL-MM	AL-MM	AL-AM
WIFN: $\sum_{i=1}^{C_{out}} \|\|\mathbf{X}_j^\ell\|\|_2 \cdot \|\mathbf{W}_{ij}^\ell\|$	84.79	128.75	<u>34.50</u>	**34.09**
IFV: $\frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{X}_{n,j,:}^\ell - \overline{\mathbf{X}}_{:,j,:}^\ell)^2$	<u>55.41</u>	48.87	35.72	**33.33**
WIFV: $\frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{X}_{n,j,:}^\ell - \overline{\mathbf{X}}_{:,j,:}^\ell)^2 \cdot \|\|\mathbf{W}_{:,j}^\ell\|\|_2^2$	57.57	<u>38.31</u>	34.82	**<u>31.80</u>**

Table 3: Ablation on pruning metric and compressed model structure. Bold results denote the best compressed model structure found for each pruning metric. Underscored results indicate the best pruning metric found for each compressed model structure.

FLAP 's performance improves as the size of the calibration dataset increases. In our experiments, we selected a default setting of 1024 calibration samples. Given that only a single forward propagation is required for this calculation, the computational cost associated with this sample size is minimal. Notably, the entire pruning process for the LLaMa-7B model is efficiently completed in a span of 3 to 5 minutes on a single GPU.

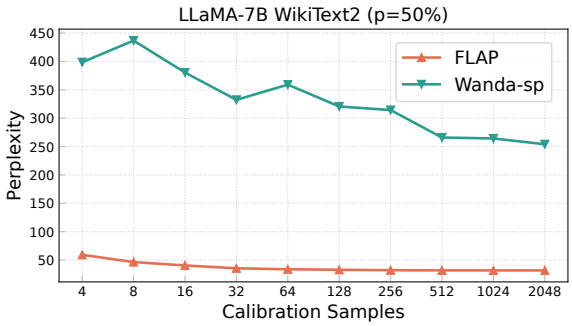

Figure 6: Robustness to Calibration Samples.

Inference Speed

Unlike unstructured pruning, structured pruning offers the dual benefit of reducing both the number of parameters and the inference time, without the need for specialized hardware. This makes structured pruning a more universally applicable approach. In this section, we empirically compare the actual parameter counts and inference speeds of different pruning methods, with the experiments conducted on NVIDIA A100 GPUs. The detailed results are presented in Table 4. Notably, Wanda, employed here as a representative of unstructured pruning, does not effectively reduce either the parameter count or the inference speed. In contrast, our method demonstrates substantial efficiency improvements: at a 20% pruning ratio, it reduces the number of parameters by 52%, and accelerates the inference speed by 66%. At a 50% pruning ratio, these improvements are further amplified, with reductions in parameter count by 25%, and an increase in speed by 31%.

Figure 7 compares the throughput of the LLaMA-7B model with a model pruned by 50% using our method, across various batch sizes. The comparison clearly shows that the pruned model benefits more at larger batch sizes, as it has not yet hit the throughput bottleneck.

Method	Pruning Ratio	Params	Memory	Tokens/s
LLaMA-7B	0%	6.74B	12916.5MiB	25.84
Wanda	20%	6.74B	12916.5MiB	25.67 (≈ 0%)
LLM-Pruner		5.42B	10387.2MiB	32.57 (↑ 26%)
FLAP (Ours)		5.07B	9726.2MiB	33.90 (↑ 31%)
Wanda	50%	6.74B	12916.5MiB	25.95 (≈ 0%)
LLM-Pruner		3.35B	6547.1MiB	40.95 (↑ 58%)
FLAP (Ours)		3.26B	6268.2MiB	42.88 (↑ 66%)

Table 4: Inference speed and memory footprint comparison.

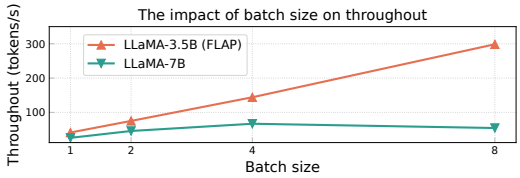

Figure 7: The impact of batch size on throughput. The hardware is the NVIDIA A100-40G.

Conclusion

In this work, we propose FLAP (**FL**uctuation-based **A**daptive Structured **P**runing), a retraining-free structured pruning framework explicitly designed for Large Language Models (LLMs). To address the challenges posed by structured pruning, we introduce a novel structured pruning metric, employ adaptive global model compression strategies, and implement robust compensation mechanisms designed to mitigate potential performance losses. Our empirical results affirm that the structured compression model crafted by FLAP can maintain perplexity and zero-shot performance without any retraining. Especially worth noting is the efficacy of FLAP in upholding model performance at both low and medium compression rates. Our work demonstrates that bias compensation can largely replace retraining or parameter efficient fine-tuning (PEFT). We hope that our work contributes to a better understanding of structured pruning and performance recovery of LLMs.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2021ZD0110400), Beijing Municipal Science and Technology Project (Z231100007423004), Zhejiang Lab (No. 2021KH0AB07), and National Natural Science Foundation of China (Grant No. 62206290, 62276260, 62176254, 61976210, 62076235).

References

Anand, Y.; Nussbaum, Z.; Duderstadt, B.; Schmidt, B.; and Mulyar, A. 2023. GPT4All: Training an Assistant-style Chatbot with Large Scale Data Distillation from GPT-3.5-Turbo. https://github.com/nomic-ai/gpt4all. Accessed: 2023-08-09.

Anwar, S.; Hwang, K.; and Sung, W. 2017. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3): 1–18.

Bommarito, M.; and Katz, D. M. 2022. GPT Takes the Bar Exam. arXiv:2212.14402.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. arXiv:2005.14165.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv:2303.12712.

Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems*.

Dettmers, T.; Svirschevski, R.; Egiazarian, V.; Kuznedelev, D.; Frantar, E.; Ashkboos, S.; Borzunov, A.; Hoefler, T.; and Alistarh, D. 2023. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. arXiv:2306.03078.

Frantar, E.; and Alistarh, D. 2023. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. arXiv:2301.00774.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. GPTQ: Accurate Post-training Compression for Generative Pretrained Transformers. In *International Conference on Learning Representations*.

Gao, L.; Tow, J.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; McDonell, K.; Muennighoff, N.; et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *International Conference on Learning Representations*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. J. 2015. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems*.

Hassibi, B.; Stork, D. G.; and Wolff, G. J. 1993. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*.

He, Y.; and Xiao, L. 2023. Structured Pruning for Deep Convolutional Neural Networks: A survey. arXiv:2303.00566.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. arXiv:2106.09685.

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*.

Ma, X.; Fang, G.; and Wang, X. 2023. LLM-Pruner: On the Structural Pruning of Large Language Models. Version 3, arXiv:2305.11627.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer Sentinel Mixture Models. arXiv:1609.07843.

Mishra, A.; Latorre, J. A.; Pool, J.; Stosic, D.; Stosic, D.; Venkatesh, G.; Yu, C.; and Micikevicius, P. 2021. Accelerating sparse deep neural networks. arXiv:2104.08378.

Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; and Kautz, J. 2017. Pruning Convolutional Neural Networks for Resource Efficient Inference. In *International Conference on Learning Representations*.

Richards, T. B. 2023. Auto-GPT: An experimental open-source attempt to make GPT-4 fully autonomous. https://github.com/Significant-Gravitas/Auto-GPT. Accessed: 2023-08-09.

Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. arXiv:2211.05100.

Shi, D.; Tao, C.; Jin, Y.; Yang, Z.; Yuan, C.; and Wang, J. 2023. Upop: Unified and progressive pruning for compressing vision-language transformers. arXiv:2301.13741.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, Z. 2023. A Simple and Effective Pruning Approach for Large Language Models. arXiv:2306.11695.

Tan, C. M. J.; and Motani, M. 2020. Dropnet: Reducing neural network complexity via iterative pruning. In *International Conference on Machine Learning*, 9356–9366. PMLR.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca. Accessed: 2023-08-09.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022. Emergent Abilities of Large Language Models. In *Transactions on Machine Learning Research*.

Xia, M.; Zhong, Z.; and Chen, D. 2022. Structured Pruning Learns Compact and Accurate Models. In *Association for Computational Linguistics (ACL)*.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *International Conference on Machine Learning*.

Zafrir, O.; Larey, A.; Boudoukh, G.; Shen, H.; and Wasserblat, M. 2021. Prune once for all: Sparse pre-trained language models. arXiv:2111.05754.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. OPT: Open pre-trained transformer language models. arXiv:2205.01068.

Zhou, A.; Ma, Y.; Zhu, J.; Liu, J.; Zhang, Z.; Yuan, K.; Sun, W.; and Li, H. 2021. Learning n: m fine-grained structured sparse neural networks from scratch. arXiv:2102.04010.