CFSP: An Efficient Structured Pruning Framework for LLMs with Coarse-to-Fine Activation Information

Yuxin Wang[1*], **Minghua Ma**[1*], **Zekun Wang**[1*†], **Jingchang Chen**[1]
Liping Shan[2], **Qing Yang**[2], **Dongliang Xu**[2], **Ming Liu**[1†], **Bing Qin**[1]
[1]Harbin Institute of Technology, Harbin, China
[2]Du Xiaoman Science Technology Co., Ltd, Beijing, China
{yuxinwang,mhma,zkwang,jcchen,mliu,qinb}@ir.hit.edu.cn

Abstract

The colossal parameters and computational overhead of Large Language Models (LLMs) challenge their real-world applications. Network pruning, which targets unstructured or structured sparsity by removing redundant parameters, has recently been explored for LLM acceleration. Existing LLM pruning works focus on unstructured pruning, which typically requires special hardware support for a practical speed-up. In contrast, structured pruning can reduce latency on general devices. However, it remains a challenge to perform structured pruning efficiently and maintain performance, especially at high sparsity ratios. To this end, we introduce an efficient structured pruning framework named CFSP, which leverages both **C**oarse (interblock) and **F**ine-grained (intrablock) activation information as an importance criterion to guide pruning. The pruning is highly efficient, as it only requires one forward pass to compute feature activations. Specifically, we first allocate the sparsity budget across blocks based on their importance and then retain important weights within each block. In addition, we introduce a recovery fine-tuning strategy that adaptively allocates training overhead based on coarse-grained importance to further improve performance. Experimental results demonstrate that CFSP outperforms existing methods on diverse models across various sparsity budgets. Our code will be available at https://github.com/wyxscir/CFSP.

1 Introduction

Although scaling up Large Language Models (LLMs) brings remarkable performance (Brown et al., 2020; OpenAI, 2023; Gemini Team et al., 2023; Meta, 2024; DeepSeek-AI et al., 2024; Yang et al., 2024a), increasing parameters brings more computations and memory consumption, posing a significant challenge of deploying in practical applications. To address this, various model compression methods for LLMs are proposed (Dettmers et al., 2022; Frantar et al., 2022; Lin et al., 2024; Muralidharan et al., 2024). Existing LLM pruning work (Frantar and Alistarh, 2023; Sun et al., 2024; Xu et al., 2024a; Zhang et al., 2024b) focuses mainly on unstructured or semi-structured sparsity. However, these paradigms require specific hardware to achieve practical acceleration.

In contrast, *structured pruning*, which imposes structured sparsity by removing groups of consecutive parameters (Louizos et al., 2017; Wang et al., 2020; Xia et al., 2022), is more hardware-friendly on general devices. However, there are some challenges involved in existing structured pruning methods for LLMs: (1) They typically introduce learnable masks to search (Xia et al., 2023; Dery et al., 2024) or utilize gradients to guide pruning (Ma et al., 2023; Zhang et al., 2023a). Unfortunately, they require significant computational overhead, especially for large-scale (*e.g.*, 70B) models. (2) It is also worth noting that they usually assign a uniform sparsity budget per block, which is suboptimal since LLM blocks have different significance in the representation functionality (Gromov et al., 2024a). Moreover, they usually involve a recovery fine-tuning with Low-Rank Adapter (LoRA) (Hu et al., 2022) to enhance pruned models, which also introduce training overhead and overlook the varying importance of blocks.

To this end, we propose CFSP (shown in Figure 1), an efficient structural pruning framework for LLM that takes advantage of coarse to fine-grained activation information to guide pruning. Specifically, we employ activation as the importance criterion, which is calculated for blocks (coarse-grained) and the weights within each block (fine-grained) in a single forward pass. For each block, we measure its saliency of transformations on the basis of the angular distance of the input and output

[*]Equal Contribution.
[†]Corresponding Author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 9311–9328
January 19–24, 2025. ©2025 Association for Computational Linguistics

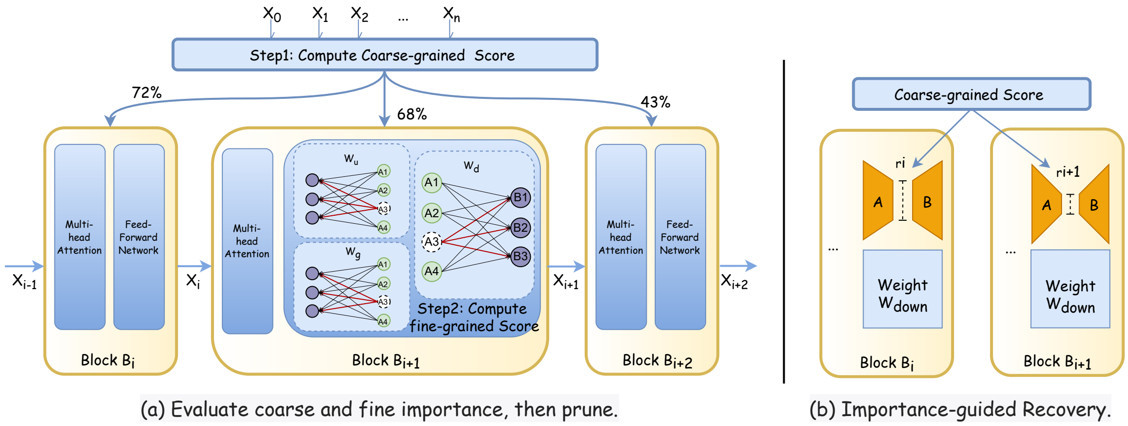

Figure 1: Illustration of our proposed CFSP framework. (a) Pruning with coarse (interblock) to fine (intrablock) activation information guidance. (b) Recovery fine-tuning with importance-guided allocation, where the rank sizes of each component are determined by coarse-grained importance.

representations. Then, we utilize this criterion as coarse-grained importance to assign the sparsity budget. Finally, for weights within each block, we take the product of their relative activations and weights as a fine-grained criterion to remove redundant parts. Since existing work typically performs recovery fine-tuning with LoRA to further improve performance, we propose a more efficient recovery method that leverages coarse-grained importance to adaptively allocate additional trainable parameters: the pruned models can achieve comparable performance while utilizing less recovery data.

Our contributions can be summarized as follows:

- We propose an efficient coarse-to-fine importance criterion for identifying redundant structures for pruning, which takes only a few minutes[1] to complete on various models.
- We introduce an efficient recovery fine-tuning method that adaptively assigns additional trainable parameters based on the coarse-grained importance score.
- Extensive experimental results indicate that CFSP surpasses existing methods across various models at different sparsity levels, demonstrating promising performance on challenging tasks even at high sparsity levels.

2 Methodology

The overview of CFSP framework is shown in Figure 1. We first introduce our preliminary analysis in Section 2.1, then give details of our pruning criterion and procedure in Section 2.2. Finally, we introduce the proposed importance-guidance recovery strategy in Section 2.3.

2.1 Preliminaries

The Transformer block (Vaswani et al., 2017) consists of multi-head attention (MHA) and feedforward network (FFN). We analyze the computational overhead and the sparsity of them. As shown in Figure 2, the parameter size and MAC of FFN are significantly larger than those of MHA. In addition, we observe that pruning MHA leads to a significant performance drop with only 10% sparsity, while pruning FFN has a more stable performance even with 50% sparsity, showing that the FFN module has a higher structural sparsity (Zhang et al., 2022) and is more friendly to structured pruning (Gunter et al., 2024). Thus, in this work, we focus on pruning the intermediate dimension of FFN.

2.2 CFSP Framework

CFSP takes activations as an importance criterion to identify redundant parts of LLMs for the following reasons: (1) Activations can be obtained with a single forward pass, resulting in significantly lower overhead compared to other metrics. (2) As pointed out in previous studies (Sun et al., 2024; Lin et al., 2024), parameter weights corresponding to larger activation magnitudes are more salient since they process more important features.

The feature activations are calculated on a small number (*e.g.*, 128) of calibration samples. We further incorporate coarse- and fine-grained importance for sparsity allocation and weight pruning.

[1] Details of time cost are shown in Table 7 in Appendix.

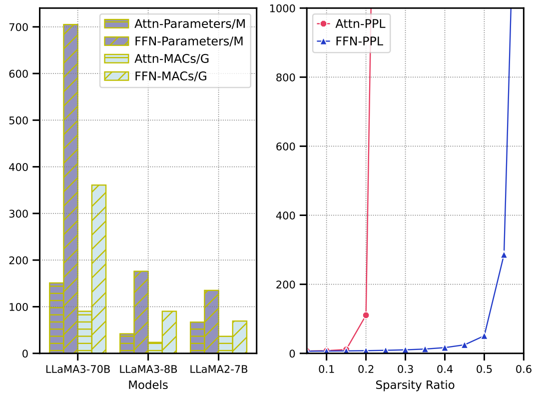

Figure 2: Preliminary analysis. (Left): Parameter size and MACs of modules. (Right): Sensitivity of pruning each module on LLaMA2-7B.

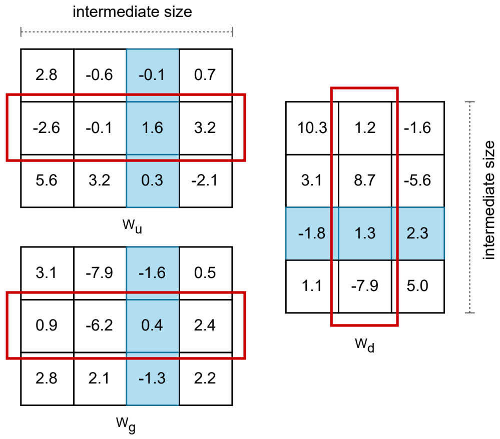

Figure 3: The structural dependencies of FFN in LLaMA3. The blue part corresponds to the minimum unit of structured pruning. The red box represents the relative size of a matrix element in its row or column.

Coarse-grained Importance Existing pruning work usually assigns the same sparsity to each block, but it is suboptimal for sparsity allocation. In fact, the blocks perform different functions and their importance varies significantly (Gromov et al., 2024b). Due to the residual structure, the effect of each block can be viewed as a transformation of the input representations.

Thus, we measure the coarse-grained importance of blocks S_g through the saliency of transformation of feature activations during the forward process. Specifically, the S_g of l-th block $\mathbf{B}^\ell$ is calculated as:

$$S_g(\mathbf{B}^\ell) = \sum_{i=0} D(\boldsymbol{x}_i^\ell, \boldsymbol{x}_i^{\ell+1}) \tag{1}$$

$$D(\boldsymbol{x}_i^\ell, \boldsymbol{x}_i^{\ell+1}) = \frac{1}{\pi}\arccos\left(\frac{\boldsymbol{x}_i^\ell \cdot \boldsymbol{x}_i^{\ell+1}}{\|\boldsymbol{x}_i^\ell\|\|\boldsymbol{x}_i^{\ell+1}\|}\right) \tag{2}$$

where $\boldsymbol{x}^\ell$ and $\boldsymbol{x}^{\ell+1}$ represent the input and output activation states of the ℓ-th block. $D(\cdot)$ can be various distance measurements of two representations. Here we select the angular distance because it performs better than the others in our experiments. We then normalize $\mathbf{B}^\ell$ with the Sigmoid as:

$$\text{Norm}(S_g(\mathbf{B}^\ell)) = \text{Sigmoid}(S_g(\mathbf{B}^\ell) - \overline{S}) \tag{3}$$

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-\alpha \cdot x}} \tag{4}$$

where $\overline{S}$ is the average importance score of all blocks. The function Sigmoid is introduced to process the scores non-linearly, which can make the distinction between blocks more significant, and α controls the intensity of significance. Finally, we assign the sparsity budgets across blocks based on the normalized importance scores as:

$$\text{Sparsity}(\mathbf{B}^\ell) = \frac{\text{Norm}(S_g(\mathbf{B}^\ell)) \cdot \gamma \cdot n}{\sum_{\ell=1}^{n} \text{Norm}(S_g(\mathbf{B}^\ell))} \tag{5}$$

where n is the number of blocks and γ represents the whole sparsity budget of the model.

Dimension Adjustment Equation 5 can assign irregularly shaped weight matrices that do not satisfy the multiples of 64 or 128, thus destroying the parallelism of the tensor cores on the GPU. To this end, we introduce a simple adjustment during pruning to adjust the final dimensions of the pruned blocks to multiples of 128. For the l-th block $\mathbf{B}^l$, the final dimension dim_f^l are computed as:

$$dim_f^l = \left(\left\lfloor \frac{dim_o \times \text{Sparsity}(\mathbf{B}^i) + 64}{128} \right\rfloor\right) \times 128 \tag{6}$$

where dim_o is the intermediate dimensions of original dense model. We present ablation results in Section 3.5 that demonstrate that the adjustment of dimensions significantly accelerates inference speed on GPUs. Notably, this enhancement is achieved with only a minimal increase in parameters and no detrimental impact on performance.

Fine-grained Importance After assigning the proper sparsity to each block, we then identify the importance of pruning units (intermediate dimensions) within the block. Figure 3 shows the structural dependencies of three matrices used in

FFN (W_u, W_g and W_d): removing the intermediate dimensions is equivalent to subtracting the corresponding columns of W_u, W_g and the corresponding rows of W_d. The weight matrix represents the connections between neurons, where each row or column of weights influences the same neuron, implying that the fine-grained importance of the weights is also related to their respective row or column. For the i-th intermediate dimension, we utilize the activation of $\mathbf{X}_d$ of W_d and all weight matrices to calculate fine-grained importance score S_l^i:

$$S_l^i = F_l^i \cdot \|\mathbf{X}_d^i\| \tag{7}$$

$$F_l^i = \sum_j \left(\frac{W_d^{ij} \cdot \|\mathbf{X}_d^i\|}{W_d^{*j} \cdot \|\mathbf{X}_d^*\|} + \frac{W_u^{ij}}{W_u^{i*}} + \frac{W_g^{ij}}{W_g^{i*}} \right) \tag{8}$$

where $\|\cdot\|$ is L2 normalization. As shown in Equation 8, the pruning structure F consists of three matrices determined by cumulative activation of the matrix W_d. The matrices W_u and W_g use a relative weight measurement, where the magnitude of the weight is proportional to the sum of the row in the matrices. An unique aspect is the matrix W_d in F, which quantifies the ratio between the weight activation magnitudes and the sum of column activation magnitudes of W_d, as shown in the first term of Equation 8.

2.3 Importance-guided Recovery Fine-tuning

In addition to the single-shot pruning scenario, we also explore the integration of recovery fine-tuning to further enhance performance at high sparsity. Our recovery setting follows Ma et al. (2023) to fine-tune with LoRA (Hu et al., 2022). Unlike the original LoRA, we propose an importance-guided method that adaptively assigns additional trainable parameters across different blocks. Specifically, for the l-th block, the rank r^l of LoRA is determined based on the coarse-grained importance scores computed during pruning:

$$r^l = \frac{\mathrm{Norm}(S_g(\mathbf{B}^\ell)) \cdot \bar{r} \cdot n}{\sum_{\ell=1}^{n} \mathrm{Norm}(S_g(\mathbf{B}^\ell))} \tag{9}$$

where $\bar{r}$ is the averaged rank allocated budget. In our experiments, we find that our recovery method is more efficient, achieving comparable performance while requiring less training data.

Recovery Data We explore various datasets for recovery fine-tuning. We find that the quality and diversity of knowledge in the data are critical for recovery performance, especially on challenging tasks (*e.g.*, MMLU). The details of datasets and results can be found in Appendices A.2 and B.2.

3 Experiments

3.1 Experimental Setup

In this work, we target to prune intermediate dimensions of FFN in LLM and conduct experiments primarily on widely used LLM models: LLaMA3-{8B,70B} (Meta, 2024) and a middle-size LLaMA2-13B (Touvron et al., 2023b). We also conduct experiments on the latest LLaMA3.1-8B (Dubey et al., 2024) and more models from LLaMA family in Appendix B.1.

Evaluation Benchmarks Following previous work (Sun et al., 2024; An et al., 2024), we evaluate the zero-shot performance of models across 5 well-known tasks: WinoGrande (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), ARC-easy and ARC-challenge (Clark et al., 2018). Since Jaiswal et al. (2024) shows that LLM pruning methods tend to have significantly degraded performance on knowledge-intensive tasks, we also include two challenging QA tasks for zero-shot evaluation: MMLU (Hendrycks et al., 2021) and FreebaseQA (Jiang et al., 2019), which focus on factual knowledge. For language modeling performance, we evaluate models on WikiText2 (Merity et al., 2017). Following previous work, we use the LM-Evaluation-Harness (Gao et al., 2023) and LLM-Kick (Jaiswal et al., 2024) with default hyperparameters for the corresponding tasks. More details of the evaluation are shown in Appendix A.1

Implementation Details For the pruning stage, the calibration data are randomly selected from the WikiText2 (Merity et al., 2017) training set. Unless otherwise stated, the calibration set consists of 128 samples and each has approximately 1024 tokens following Sun et al. (2024). In the recovery fine-tuning stage, pruned models are trained on 0.1B tokens from the FineWeb-Edu (Lozhkov et al., 2024) dataset with the next-token prediction loss. We set the average rank budget of IG-LoRA at 8 following Ma et al. (2023). More details and ablation of the implementation are shown in Appendix A.2 and Appendix B.2, respectively.

Baselines We compare the single-shot pruning performance[2] of CFSP against the following base-

[2]Without recovery fine-tuning.

Sparsity	Method	WinoGrande	PIQA	HellaSwag	OBQA	ARC-e	ARC-c	MMLU	FreebaseQA	Average
0%	LLaMA3-8B	72.93	80.96	79.17	45.00	77.90	53.16	62.09	72.62	67.98
20%	*w/o recovery*									
	Magnitude-SP	50.99	51.31	26.18	30.20	25.43	25.76	23.71	0.52	29.26
	Wanda-SP	67.56	75.41	65.99	**42.00**	65.40	41.38	46.20	**39.11**	55.38
	FLAP	65.67	74.65	62.41	40.20	61.36	35.15	41.39	34.58	51.93
	CFSP (ours)	**70.32**	**77.64**	**72.74**	41.20	**68.10**	**43.86**	**56.43**	38.59	**58.61**
50%	*w/o recovery*									
	Magnitude-SP	51.14	50.27	26.47	29.40	25.08	26.49	23.08	0.52	29.06
	Wanda-SP	59.51	63.98	45.71	**33.00**	44.95	27.99	27.08	6.15	38.54
	FLAP	58.80	62.35	41.89	31.00	40.28	26.11	24.03	4.55	36.12
	CFSP (ours)	**62.04**	**66.76**	**49.96**	31.80	**48.74**	**30.89**	**32.39**	**10.83**	**41.67**
	w/ recovery									
	Wanda-SP	61.48	70.89	60.20	**37.60**	60.86	36.43	35.54	11.89	46.86
	CFSP (ours)	**65.51**	**72.03**	**61.45**	36.20	**62.37**	**37.54**	**40.37**	**18.32**	**49.22**

Table 1: Zero-shot performance of pruned models on LLaMA3-8B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

Sparsity	Method	WinoGrande	PIQA	HellaSwag	OBQA	ARC-e	ARC-c	MMLU	FreebaseQA	Average
0%	LLaMA3-70B	80.35	84.71	84.93	48.60	85.90	64.16	75.36	81.53	75.69
20%	*w/o recovery*									
	Magnitude-SP	51.93	58.38	32.69	29.40	32.41	27.73	24.44	0.52	32.19
	Wanda-SP	77.19	82.92	82.50	**49.20**	81.65	58.28	66.74	79.65	72.27
	FLAP	77.51	82.48	80.41	47.40	78.49	55.12	65.88	79.02	70.79
	CFSP (ours)	**80.66**	**83.51**	**83.97**	46.40	**83.46**	**61.43**	**73.04**	**80.18**	**74.08**
50%	*w/o recovery*									
	Magnitude-SP	51.22	52.72	27.04	30.00	25.46	26.62	23.52	0.55	29.64
	Wanda-SP	73.95	76.44	73.80	**44.00**	66.79	43.94	54.91	42.26	59.51
	FLAP	72.85	76.82	68.05	42.80	66.54	45.05	53.90	38.41	58.05
	CFSP (ours)	**75.06**	**78.89**	**75.95**	43.60	**71.34**	**46.67**	**59.74**	**46.42**	**62.20**
	w/ recovery									
	Wanda-SP	76.33	80.02	79.73	**47.20**	73.10	47.26	59.98	48.20	63.98
	CFSP (ours)	**78.30**	**81.01**	**80.18**	45.20	**76.65**	**51.54**	**65.52**	**54.77**	**66.65**

Table 2: Zero-shot performance of pruned models on LLaMA3-70B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

lines: **Magnitude-SP** measures the importance criterion based on the magnitude of weights (Han et al., 2016; Jaiswal et al., 2023). This baseline employs uniform sparsity across blocks. **Wanda-SP** is extended by the unstructured pruning method Wanda (Sun et al., 2024), which modifies the target pruning units to structured weights. We globally sort the pruning units across all blocks to identify redundant components, as this strategy tends to achieve better performance compared to adopting a local manner for individual blocks. **FLAP** (An et al., 2024) uses the stability of activations as an importance criterion, also applying a global sorting strategy. Notably, for a fair comparison, all baselines are implemented to prune the intermediate dimensions of the FFN, which are the same as CFSP. Details are shown in Appendix A.3.

3.2 Main Results

Zero-shot Tasks We present a performance comparison of the LLaMA3 family in Tables 1 and 2, as well as LLaMA2-13B in Table 3. In the single-shot pruning setting, CFSP consistently demonstrates superior average performance compared to baselines across various models at both 20% and 50% sparsity. Remarkably, CFSP achieves a promising accuracy of 32.39 on MMLU with 50% sparsity on LLaMA3-8B, while other baselines regress to chance-level accuracy (~25.0). This result underscores the potential of CFSP to perform well on more challenging tasks without retraining, even at high sparsity. Furthermore, CFSP is more fa-

Sparsity	Method	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	FreebaseQA	Average
0%	LLaMA2-13B	72.22	80.52	45.20	79.38	77.48	49.06	50.51	67.57	65.24
20%	*w/o recovery*									
	Magnitude-SP	49.96	60.01	25.60	39.89	42.93	29.86	25.51	0.65	34.30
	Wanda-SP	70.01	**78.45**	43.00	73.87	72.56	44.28	41.70	40.69	58.07
	FLAP	68.27	77.58	41.40	72.58	67.47	42.58	41.15	28.63	54.96
	CFSP (ours)	**71.75**	78.29	**43.60**	**75.76**	**73.48**	**47.27**	**46.99**	**54.65**	**61.47**
50%	*w/o recovery*									
	Magnitude-SP	50.75	50.16	24.20	26.17	27.19	25.85	25.20	0.53	28.76
	Wanda-SP	64.80	71.76	38.00	57.36	59.09	37.80	27.00	3.20	44.87
	FLAP	60.54	68.50	36.60	53.95	48.78	30.97	23.00	1.50	40.48
	CFSP (ours)	**64.17**	**71.98**	**39.40**	**60.28**	**62.33**	**38.05**	**28.24**	**3.65**	**46.01**
	w/ recovery									
	Wanda-SP	66.85	74.37	**40.60**	67.15	68.31	**41.04**	35.24	25.35	52.36
	CFSP (ours)	**67.17**	**74.88**	**40.60**	**68.60**	**69.23**	40.87	**36.41**	**25.78**	**52.94**

Table 3: Zero-shot performance of pruned models on LLaMA2-13B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

Sparsity	Method	LLaMA3-8B	LLaMA3-70B
0%	Dense	6.82	5.26
20%	*w/o recovery*		
	Wanda-SP	9.39	**7.86**
	FLAP	9.40	8.21
	CFSP(ours)	**8.97**	8.02
50%	*w/o recovery*		
	Wanda-SP	19.49	13.53
	FLAP	21.06	13.37
	CFSP(ours)	**17.45**	**13.02**
	w/ recovery		
	Wanda-SP	14.52	11.75
	CFSP(ours)	**12.55**	**10.92**

Table 4: Perplexity of pruning methods for LLaMA3-8B and LLaMA3-70B on WikiText2 validation set.

Model	Parameters	Memory	MACs	Speed-up CPU	Speed-up GPU
LLaMA3-8B	8.03B	16.06GB	3.64T	1.0x	1.0x
+ CFSP	5.21B	10.42GB	2.19T	2.3x	1.6x
LLaMA2-7B	6.73B	12.61GB	3.38T	1.0x	1.0x
+ CFSP	4.57B	8.62GB	2.17T	2.1x	1.5x

Table 5: Comparison of parameter size, memory usage, MACs, and inference speed-up on CPU/GPU. The pruned models (+CFSP) are under 50% sparsity.

vorable for larger models. At the 20% and 50% sparsity on LLaMA3-70B, CFSP maintains 97.9% and 82.2% of the original performance on average, respectively. We further evaluate CFSP with recovery fine-tuning at 50% sparsity for each model. For comparison, we choose Wanda-SP, as it has the second-best average performance in single-shot pruning. We fine-tune pruned models with our proposed IG-LoRA on 0.1B tokens from the FineWeb-Edu dataset. We find that after recovery training, both pruning models are improved, especially on complex knowledge-sensitive tasks. CFSP still outperforms Wanda-SP in general, indicating the effectiveness of our proposed pruning and recovery approach.

Language Modeling Table 4 presents the perplexity on WikiText2. CFSP consistently achieves better results than baselines, except for the 20% sparsity on LLaMA3-8B, where it performs slightly worse than Wanda-SP. Additionally, the benefits of CFSP are more pronounced at higher sparsity.

3.3 Efficiency Evaluation

We assess the inference efficiency of the pruned models. The details of evaluation are shown in Appendix A.1. The results of 50% sparsity are shown in Table 5. Compared to the original dense models, CFSP reduces the parameters, memory, and MACs by 40% and achieves a speed-up over 1.5× on CPU and GPU. We also report the pruning and recovery time in Appendix A.2. In general, CFSP significantly improves efficiency, indicating its effectiveness for practical deployments of LLM.

3.4 Ablation Study

Importance Criterion We explore the effects of each component incorporated in the importance

Setting	PPL↓	HellaSwag	MMLU
coarse-grained importance ablation			
(a) Uniform	9.08	70.84	50.91
(b) Euclidean	9.11	70.11	50.19
(c) Cosine	8.98	72.52	55.79
Angular (Ours)	**8.97**	**72.74**	**56.43**
fine-grained importance ablation			
(d) Wanda	9.03	71.93	55.33
Eq (8) (Ours)	**8.97**	**72.74**	**56.43**

Table 6: Ablation of importance criterion of CFSP on LLaMA3-8B under 20% sparsity.

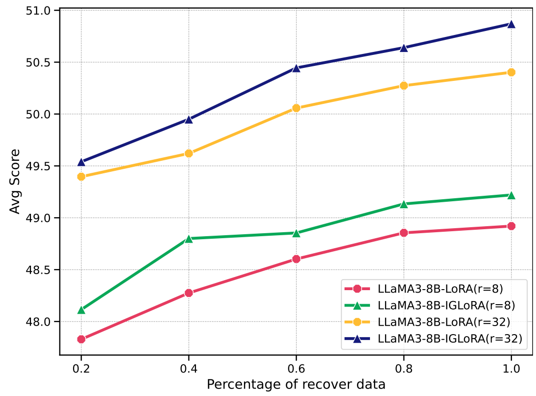

Figure 4: Results of different recovery fine-tuning methods at different data sizes. $r = 8/32$ means the average rank budget configuration is set to 8 or 32.

criterion of the proposed CFSP. Table 6 shows the ablation results under 20% sparsity of LLaMA3-8B. We first investigate the coarse-grained importance of blocks by comparing variants including: (a) uniform sparsity for each block, (b) Euclidean distance, or (c) cosine similarity as the coarse-grained importance criterion to allocation sparsity budget across blocks. As illustrated in Table 6, applying uniform sparsity or using Euclidean distance results in a notable performance decrease, particularly for zero-shot tasks. The angular distance (Eq. 2) used in CFSP achieves the best performance across tasks. For fine-grained importance ablation, as shown in Table 6, the criterion outlined in Eq. 8 also demonstrates superior performance compared to the criterion utilized in Wanda.

Recovery Fine-tuning To assess the impact of our proposed IG-LoRA for recovery, we compare it with the original LoRA. Figure 4 shows that IG-LoRA exhibits better performance than LoRA across various recovery data sizes and rank configurations. Furthermore, IG-LoRA achieves a performance comparable to LoRA trained on the full dataset while utilizing only 60% of data, highlighting the efficiency of IG-LoRA.

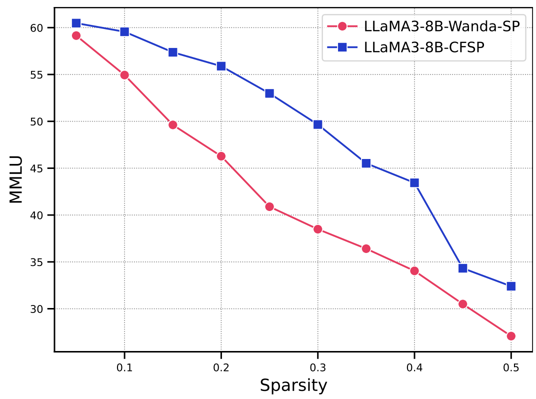

Figure 5: Performance comparison of MMLU task between CFSP and Wanda-SP on LLaMA3-8B with various sparsity.

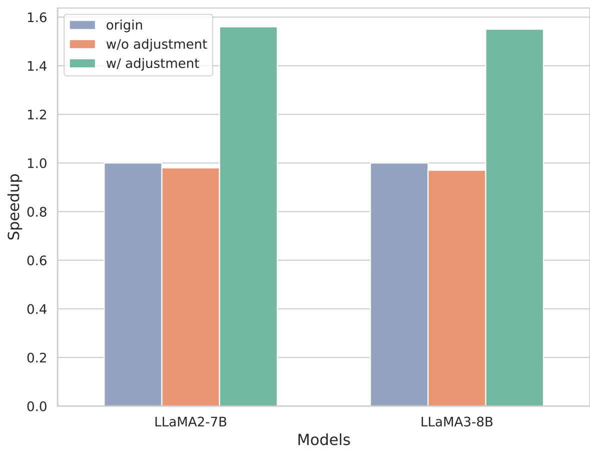

Figure 6: The effect of dimension adjustment. The speed-up is evaluated on a NVIDIA A800-80G.

3.5 Analysis

Performance with Various Sparsity Figure 5 presents the MMLU results of pruned models with sparsity from 5% to 50%. Under lower sparsity (10%), Wanda-SP is comparable to CFSP. As the sparsity increases, its performance decreases significantly, while CFSP still maintains promising performance even at 50% sparsity.

Impact of Dimension Adjustment Figure 6 compares the inference speed-up of whether to perform dimension adjustment during pruning. We observe that adjusting the intermediate dimension significantly accelerates models (1.6×). However, without the adjustment, the latency of pruned models is comparable to the original dense models. Furthermore, the cost of adjustment is negligible and does not impact performance. For instance, on LLaMA3-8B, the number of parameters

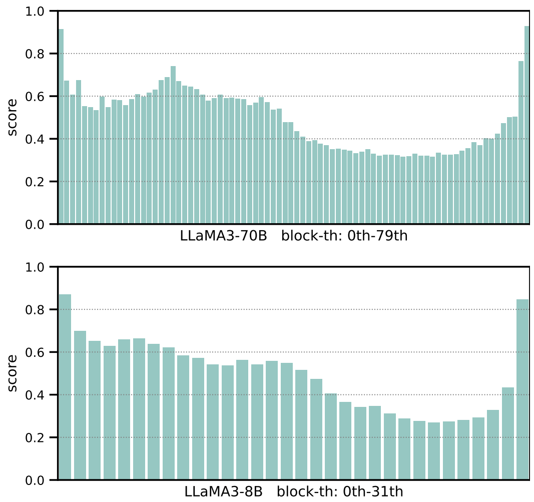

Figure 7: The visualizations of normalized coarse-grained importance scores of each block on LLaMA3-8B and LLaMA3-70B.

increased only by 0.43% after adjustment (from 5.21B to 5.23B). The average zero-shot performance remains comparable to that without adjustment (41.67 *vs* 41.61).

Visualization We present a visualization of block importance scores on LLaMA3 in Figure 7. We find that the scores vary significantly across blocks: the first and last blocks exhibit the highest scores, whereas the intermediate blocks show lower scores. The varying importance explains why the uniform sparsity allocation is suboptimal and indicates that intermediate blocks exhibit greater redundancy, allowing for more aggressive pruning, while other blocks need to retain more weights.

4 Related Work

LLM Compression The enormous computations of LLMs has prompted efforts in improving their efficiency, including quantization (Dettmers et al., 2022; Yao et al., 2022; Lin et al., 2024; Frantar et al., 2022; Xiao et al., 2023; Dettmers et al., 2023a,b; Shao et al., 2023; Xu et al., 2024b), distillation (Wang et al., 2022; Jiang et al., 2023; Wang et al., 2023; Hsieh et al., 2023; Agarwal et al., 2023; Gu et al., 2023) and KV cache compression (Sheng et al., 2023; Ge et al., 2023; Zhang et al., 2023b; Liu et al., 2024; Hooper et al., 2024; DeepSeek-AI et al., 2024). Pruning is another crucial method by eliminating redundant parameters. Most of the previous pruning work follows the *unstructured pruning*, which removes individual parameters according to their importance (Frantar and Alistarh, 2023; Sun et al., 2024; Dettmers et al., 2023b; Xu et al., 2024a; Zhang et al., 2024b). However, this paradigm requires specialized hardware support to speed up. In contrast, *structured pruning* eliminates the structural group of weights, facilitating a more convenient deployment on general hardware (Wang et al., 2020; Xia et al., 2022). Some work proposes to remove redundant layers in LLMs (Men et al., 2024; Yang et al., 2024b; Gromov et al., 2024b), while dropping entire layers leads to a significant performance drop. For pruning on more fine-grained units, some work formulates pruning as a constrained optimization problem by introducing learnable masks to search (Xia et al., 2023; Dery et al., 2024; Muralidharan et al., 2024; Gunter et al., 2024). Zhang et al. (2024a) performs iteratively to prune the coupled weights until the desired sparsity is achieved. Ma et al. (2023) and Zhang et al. (2023a) use gradient information to guide pruning. These methods incur substantial pruning overhead, particularly in the case of large-scale models. An et al. (2024) eliminates channels based on their activation fluctuations using only forward passes. In this work, we also aim to achieve efficient structured pruning using only forward passes.

Sparsity in Transformer Sparsity is a common trait in neural networks (Allen-Zhu et al., 2019; Frankle and Carbin, 2019; Jaszczur et al., 2021) and a lot of work explores sparsity in Transformer, such as attention (Voita et al., 2019; Michel et al., 2019; Hao et al., 2021; Zhu et al., 2021) or FFN (Wang et al., 2020; Zhang et al., 2022; Zuo et al., 2022). The dynamic sparsity has also garnered attention (Liu et al., 2023; Wang et al., 2024), which adaptively selects a portion of the model based on input. Yin et al. (2024) find that non-uniform sparsity yields better results for LLM unstructured pruning, which is consistent with our observation in structured pruning.

5 Conclusion

In this work, we explore structured pruning for Large Language Models (LLMs). We propose an efficient pruning framework named CFSP, which leverages coarse to fine-grained activation information as an importance criterion to determine the redundant parts to prune. For the coarse-grained importance, we measure the saliency of transformations of each block and use this criterion to assign the sparsity budget across blocks. For weights within each block, we utilize a fine-grained criterion to remove redundant parts to obtain compact models. We also introduce an efficient recovery fine-tuning method IG-LoRA that adaptively assigns additional trainable parameters based on the importance of blocks. Extensive experimental results demonstrate that CFSP outperforms existing methods across various models and sparsity levels, both in single-shot pruning and in recovery fine-tuning. Meanwhile, even at high sparsity, our method can maintain promising performance on challenging tasks such as MMLU and FreebaseQA compared to the original dense models.

Limitations

CFSP is a fast and efficient structured pruning method for large language models (LLMs), while it also has some limitations. First, our experiments focus on the LLaMA family of models (Touvron et al., 2023a,b; Meta, 2024; Dubey et al., 2024), as they are among the most advanced open-source LLMs currently. We will extend our method to a broader range of models in the future. Additionally, we do not prune attention heads, as this has been shown to cause significant performance degradation, especially for models that have grouped query attention (GQA) (Ainslie et al., 2023) like LLaMA3. Further research is needed to develop more effective pruning strategies, especially in the context of attention optimization techniques like GQA.

Acknowledgment

The research in this article is supported by the National Science Foundation of China (U22B2059, 62276083), the Human-Machine Integrated Consultation System for Cardiovascular Diseases (2023A003).

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. GKD: generalized knowledge distillation for auto-regressive sequence models. *ArXiv preprint*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *Proc. of ICML*, Proceedings of Machine Learning Research.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. Fluctuation-based adaptive structured pruning for large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari Do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. In *Proc. of ICLR*.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *ArXiv preprint*.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie

Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Lucio M. Dery, Steven Kolawole, Jean-François Kagey, Virginia Smith, Graham Neubig, and Ameet Talwalkar. 2024. Everybody prune now: Structured pruning of llms with only forward passes. *ArXiv preprint*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *ArXiv preprint*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023a. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023b. Spqr: A sparse-quantized representation for near-lossless LLM weight compression. *ArXiv preprint*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *ArXiv preprint*.

Zhaoye Fei, Yunfan Shao, Linyang Li, Zhiyuan Zeng, Hang Yan, Xipeng Qiu, and Dahua Lin. 2024. Query of cc: Unearthing large scale domain-specific knowledge from public corpora. *ArXiv preprint*.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proc. of ICLR*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: accurate post-training quantization for generative pre-trained transformers. *ArXiv preprint*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2023. Model tells you what to discard: Adaptive KV cache compression for llms. *ArXiv preprint*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu

Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *ArXiv preprint*.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2024a. The unreasonable ineffectiveness of the deeper layers. *ArXiv preprint*.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2024b. The unreasonable ineffectiveness of the deeper layers. *ArXiv preprint*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *ArXiv preprint*.

Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, Deepak Gopinath, Dian Ang Yap, Dong Yin, Feng Nan, Floris Weers, Guoli Yin, Haoshuo Huang, Jianyu Wang, Jiarui Lu, John Peebles, Ke Ye, Mark Lee, Nan Du, Qibin Chen, Quentin Keunebroek, Sam Wiseman, Syd Evans, Tao Lei, Vivek Rathod, Xiang Kong, Xianzhi Du, Yanghao Li, Yongqiang Wang, Yuan Gao, Zaid Ahmed, Zhaoyang Xu, Zhiyun Lu, Al Rashid, Albin Madappally Jose, Alec Doane, Alfredo Bencomo, Allison Vanderby, Andrew Hansen, Ankur Jain, Anupama Mann Anupama, Areeba Kamal, Bugu Wu, Carolina Brum, Charlie Maalouf, Chinguun Erdenebileg, Chris Dulhanty, Dominik Moritz, Doug Kang, Eduardo Jimenez, Evan Ladd, Fangping Shi, Felix Bai, Frank Chu, Fred Hohman, Hadas Kotek, Hannah Gillis Coleman, Jane Li, Jeffrey P. Bigham, Jeffery Cao, Jeff Lai, Jessica Cheung, Jiulong Shan, Joe Zhou, John Li, Jun Qin, Karanjeet Singh, Karla Vega, Kelvin Zou, Laura Heckman, Lauren Gardiner, Margit Bowler, Maria Cordell, Meng Cao, Nicole Hay, Nilesh Shahdadpuri, Otto Godwin, Pranay Dighe, Pushyami Rachapudi, Ramsey Tantawi, Roman Frigg, Sam Davarnia, Sanskruti Shah, Saptarshi Guha, Sasha Sirovica, Shen Ma, Shuang Ma, Simon Wang, Sulgi Kim, Suma Jayaram, Vaishaal Shankar, Varsha Paidi, Vivek Kumar, Xin Wang, Xin Zheng, and Walker Cheng. 2024. Apple intelligence foundation language models. *ArXiv preprint*.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *Proc. of ICLR*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proc. of ICLR*.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length LLM inference with KV cache quantization. *ArXiv preprint*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proc. of ICLR*.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. 2024. Compressing llms: The truth is rarely pure and never simple. In *The Twelfth International Conference on Learning Representations*.

Ajay Jaiswal, Shiwei Liu, Tianlong Chen, and Zhangyang Wang. 2023. The emergence of essential sparsity in large pre-trained models: The weights that matter. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Lukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. 2021. Sparse is enough in scaling transformers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proc. of NAACL-HLT*.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. *ArXiv preprint*.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2017. Learning sparse neural networks through l_0 regularization. *ArXiv preprint*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *ArXiv preprint*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proc. of ICLR*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proc. of EMNLP*.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *ArXiv preprint*.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models. *ArXiv preprint*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. Flexgen: High-throughput generative inference of large language models with a single GPU. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *Proc. of ICLR*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proc. of ACL*.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. SCOTT: self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*.

Zekun Wang, Jingchang Chen, Wangchunshu Zhou, Haichao Zhu, Jiafeng Liang, Liping Shan, Ming Liu, Dongliang Xu, Qing Yang, and Bing Qin. 2024. Smarttrim: Adaptive tokens and attention pruning for efficient vision-language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*.

Zekun Wang, Wenhui Wang, Haichao Zhu, Ming Liu, Bing Qin, and Furu Wei. 2022. Distilled dual-encoder model for vision-language understanding. In *Proc. of EMNLP*.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proc. of EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *ArXiv preprint*.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proc. of ACL*.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research.

Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. 2024a. BESA: pruning large language models with blockwise parameter-efficient sparsity allocation. *ArXiv preprint*.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. 2024b. Onebit: Towards extremely low-bit large language models. *ArXiv preprint*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *ArXiv preprint*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024b. Laco: Large language model pruning via layer collapse. *ArXiv preprint*.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. 2024. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning llms to high sparsity. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL*.

Honghe Zhang, Xiaolong Shi, Jingwei Sun, and Guangzhong Sun. 2024a. Structured pruning for large language models using coupled components elimination and minor fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2023a. Pruning meets low-rank parameter-efficient fine-tuning. *ArXiv preprint*.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. 2024b. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. 2023b. H2O: heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Haichao Zhu, Zekun Wang, Heng Zhang, Ming Liu, Sendong Zhao, and Bing Qin. 2021. Less is more: Domain adaptation with lottery ticket for reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. MoEBERT: from BERT to mixture-of-experts via importance-guided adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

A Details of Experimental Setup

A.1 Details of Evaluation Benchmarks

Zero-shot Tasks Evaluation In this work, we consider the following tasks for evaluating zero-shot performance, along with their respective evaluation metrics: WinoGrande (Sakaguchi et al., 2020) with the accuracy, PIQA (Bisk et al., 2020) with the normalized accuracy, OBQA (Mihaylov et al., 2018) with the normalized accuracy, HellaSwag (Zellers et al., 2019) with the normalized accuracy, ARC-easy/challenge (Clark et al., 2018) with the normalized accuracy, MMLU (Hendrycks et al., 2021) with the accuracy, FreebaseQA (Jiang et al., 2019) with the exact-match score. The first 6 tasks are general common sense reasoning tasks, while the others are knowledge-intensive. We evaluate WinoGrande, PIQA, HellaSwag, BoolQ, ARC-e/c, and MMLU by LM-Evaluation-Harness (Gao et al., 2023) in multiple choice form: we compute the loglikelihood for each choice and report the accuracy for the highest choice. For FreebaseQA, the evaluation is run with LLM-Kick (Jaiswal et al., 2024).

Language Modeling Evaluation We evaluate language modeling performance on WikiText2 (Merity et al., 2017) validation set with the setting of (Gao et al., 2023). The input length is 1024 for the 8B/13B models and 256 for the 70B models.

Inference Efficiency Evaluation We evaluate the speed-up of CPU on an Intel Xeon E5-466 2640 v4 CPU and the speed-up of GPU on a single A800-80G GPU. We set the sequence length to 1024 and the batch size to 1.

A.2 Implementation Details

We implement CFSP with Huggingface Transformer (Wolf et al., 2019). We perform experiments on NVIDIA A800-80G GPUs. The pruning stage is conducted on 1 GPU for the 7B/13B models and 8 GPUs for the 70B models. Unless otherwise stated, the calibration dataset consists of 128 samples and each has approximately 1024 tokens following Sun et al. (2024); An et al. (2024). By default, for the 7B/8B/13B models, α in Equation 3 is set to 1, whereas for the 70B model, it is set to 3. For the recovery fine-tuning stage, the average rank budget is set to 8 by default. We explore the following datasets for recovery training:

- **Slimpajama**[3] (Soboleva et al., 2023) is created by cleaning and deduplicating the RedPajama dataset (Computer, 2023).
- **Alpaca-Cleaned**[4] is a cleaned version of the original Alpaca (Taori et al., 2023), which is also used as recovery data in previous LLM structural pruning work Ma et al. (2023); Ashkboos et al. (2024).

[3] https://huggingface.co/datasets/DKYoon/SlimPajama-6B

[4] https://huggingface.co/datasets/yahma/alpaca-cleaned

Model Size	prune		recovery	
	device	time	device	time
7/8B	1xA800-80G	2min	8xA800-80G	0.5h
13B	1xA800-80G	4-5min	8xA800-80G	0.92h
70B	8xA800-80G	15min	16xA800-80G	5.42h

Table 7: Details of time cost of pruning and recovery for different model sizes.

- **Knowledge-Pile**[5] (Fei et al., 2024) is a dataset with high-quality knowledge data retrieved from public corpora.
- **FineWeb-Edu**[6] is a dataset filtered from FineWeb (Penedo et al., 2024), focusing on high-quality educational web pages using a classifier trained with annotations from LLaMA3-70B-Instruct.

In our experiments, we use FinWeb-Edu as our default recovery data since it achieves the best performance across all tasks. More experimental results are shown in Appendix B.2. We use the AdamW optimizer with a learning rate of 2e-4 for the 8B and 13B models, and 1e-4 for the 70B models. The batch size is set to 128. We use 8 GPUs to fine-tune the pruned 7B/8B/13B models and 16 GPUs for the 70G models. We also show the details of time cost for pruning and recovery in Table 7.

A.3 Details of Baselines

In this section, we present more details of baselines in comparison:

Magnitude-SP measures the importance criterion based on the magnitude of weights (Han et al., 2016; Jaiswal et al., 2023). This baseline employs with uniform sparsity across blocks.

Wanda-SP is extended by the unstructured pruning method Wanda (Sun et al., 2024), which modifies the target pruning units to structured weights. This baseline uses the product of weights and activations as an importance criterion. We globally sort the pruning units across all blocks to identify redundant components, as this strategy tends to achieve better performance compared to adopting a local manner for individual blocks.

FLAP (An et al., 2024) is a training-free structured pruning method for LLM, using the stability of activations as an importance criterion with a global sorting strategy. We follow its optimal setting: *Weighted Input Feature Variance*.

For a fair comparison, all baselines are implemented to prune the intermediate dimensions of FFN, which are the same as CFSP. Since the original FLAP paper only reports the results of both MHA and FFN pruning on LLaMA, we reimplement based on their official code and conduct on more models.

[5] https://huggingface.co/datasets/Query-of-CC/Knowledge_Pile

[6] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu

Sparsity	Method	Average	PPL
0%	Qwen2.5-7B	68.58	7.64
20%	*w/o recovery*		
	Wanda-SP	62.70	**8.82**
	FLAP	61.71	9.12
	CFSP (ours)	**63.02**	9.03

Table 8: The averaged zero-shot performance and PPL on wikitext2 of pruned models on Qwen2.5-7B under 20% sparsity. **Bold** indicates the best results.

B More Results and Analysis

B.1 More Results on LLaMA

Results of LLaMA3.1-8B In addition to the experiments presented in Section 3.2, we also conduct experiments on the latest powerful model LLaMA3.1-8B.

The results are shown in Table 9. In the single-shot pruning setting, CFSP consistently outperforms other baselines across a variety of tasks and sparsity budgets. Furthermore, we experiment with recovery fine-tuning for CFSP and Wanda-SP. As observed with previous models, our method still achieves better results.

Results of LLaMA2-7B Table 12 shows the results on LLaMA2-7B. In the single-shot pruning setting, CFSP consistently exceeds other baselines on various tasks and sparsity levels. Additionally, we perform recovery fine-tuning for both CFSP and Wanda-SP. As with other models, CFSP also provides superior performance.

Results of LLaMA1 Since the LLaMA1 family models were released earlier and no longer the best open-source LLMs, we do not include their results in Section 3.2. Here, we present the zero-shot performance comparison of LLaMA1 family in Table 10 and Table 11[7]. It can be observed

[7] The results of Wanda-SP reported by us differ from those in An et al. (2024) since we employ a global sorting strategy as described in Appendix A.3.

Sparsity	Method	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	Average
0%	LLaMA3.1-8B	73.95	81.01	44.8	78.91	80.85	53.33	62.95	69.74
20%	*w/o recovery*								
	Wanda-SP	67.96	74.65	41.00	65.95	64.44	39.85	45.21	57.01
	FLAP	64.64	73.72	40.60	61.92	61.11	35.92	38.20	53.73
	CFSP (ours)	**71.51**	**76.88**	**41.60**	**72.28**	**70.39**	**44.88**	**54.59**	**63.59**
50%	*w/o recovery*								
	Wanda-SP	58.88	63.76	32.20	46.03	46.93	29.52	26.39	43.39
	FLAP	58.09	59.96	31.00	41.98	39.56	26.02	23.15	39.97
	CFSP (ours)	**61.09**	**66.16**	**32.40**	**49.31**	**48.70**	**29.95**	**32.05**	**45.67**
	w/ recovery								
	Wanda-SP	61.88	70.78	**36.80**	59.58	61.53	36.43	36.44	51.92
	CFSP	**65.19**	**71.16**	36.40	**61.23**	**62.54**	**37.29**	**40.65**	**55.83**

Table 9: Zero-shot performance of pruned models on LLaMA3.1-8B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** results indicate the best results under the same setting.

Sparsity	Method	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	Average
0%	LLaMA-7B	70.09	79.16	44.06	76.21	72.85	44.80	29.92	59.58
20%	*w/o recovery*								
	Magnitude-SP	49.33	52.12	24.20	27.20	28.66	25.68	24.85	33.15
	Wanda-SP	67.88	**76.17**	41.00	70.54	66.67	39.85	27.63	55.68
	FLAP	66.61	75.63	**42.00**	68.91	66.33	38.82	26.95	55.04
	CFSP (ours)	**68.43**	75.90	41.20	**71.48**	**69.44**	**42.66**	**27.75**	**56.69**
50%	*w/o recovery*								
	Magnitude-SP	52.01	49.24	26.20	26.31	26.43	26.96	24.87	33.15
	Wanda-SP	63.30	65.38	37.00	52.13	47.81	29.10	24.16	45.55
	FLAP	60.14	65.56	36.00	50.23	44.82	29.01	24.46	44.32
	CFSP (ours)	**63.69**	**66.21**	**37.20**	**54.55**	**47.98**	**30.12**	24.03	**46.25**
	w/ recovery								
	Wanda-SP	65.51	**71.33**	38.20	61.29	58.42	34.04	24.53	50.47
	CFSP	**65.55**	71.22	**39.20**	**61.31**	**58.96**	**34.73**	25.35	**50.90**

Table 10: Zero-shot performance of pruned models on LLaMA-7B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

that on LLaMA-7B and LLaMA-13B, CFSP consistently achieves the best average performance at different sparsity. An interesting phenomenon is that on some challenging tasks (*e.g.* MMLU), all pruning methods exhibit performance close to chance-level accuracy at 50% sparsity. Compared to the results on the LLaMA3 herd of models, this could be attributed to the LLaMA1 family models' inherently weaker performance on these tasks, with high-sparsity pruning further degrading this aspect of their performance.

Results of Qwen2.5 In addition to the models of the LLaMA families, we also conduct experiments on Qwen2.5-7B (Qwen Team, 2024) to verify whether our pruning framework is model-agnostic. As shown in Table 8, CFSP consistently shows better zero-shot performance on average and achieves comparable PPL.

B.2 More Analysis of CFSP

Impact of Hyperparameter α In Equation 3, we introduce a hyperparameter α to control the intensity of significance during calculating block importance. In preliminary experiments, we explore the impact of different α and the results are shown in Figure 9. We observe that for smaller models like LLaMA3-8B, a smaller α is better, while for larger models like the LLaMA3-70B model, a larger α

Sparsity	Method	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	Average
0%	LLaMA-13B	72.69	80.20	44.80	79.08	74.71	47.70	41.24	62.92
20%	*w/o recovery*								
	Magnitude-SP	48.86	58.43	27.40	33.29	33.80	29.10	23.36	36.32
	Wanda-SP	70.56	77.53	41.40	75.40	66.25	41.98	31.65	57.82
	FLAP	70.56	77.09	41.40	74.19	68.77	43.60	31.77	58.19
	CFSP (ours)	**71.43**	**78.13**	**42.80**	**76.31**	**68.81**	**45.14**	**38.07**	**60.10**
50%	*w/o recovery*								
	Magnitude-SP	50.99	50.98	26.20	27.16	27.27	27.30	23.94	33.40
	Wanda-SP	67.01	67.46	37.00	61.44	49.83	31.14	27.23	48.73
	FLAP	64.24	70.24	36.00	56.73	52.86	33.19	25.23	48.35
	CFSP (ours)	**68.43**	**71.76**	**37.60**	**63.50**	**59.85**	**37.46**	**27.38**	**52.28**
	w/ recovery								
	Wanda-SP	67.48	**75.08**	39.20	68.14	39.59	64.48	31.97	55.13
	CFSP (ours)	**68.82**	74.76	**41.40**	**68.85**	**41.30**	**67.13**	**35.63**	**56.84**

Table 11: Zero-shot performance of pruned models on LLaMA-13B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

Sparsity	Method	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	FreebaseQA	Average
0%	LLaMA2-7B	69.06	79.11	44.20	76.02	74.62	46.33	41.25	68.39	62.37
20%	*w/o recovery*									
	Magnitude-SP	48.70	52.12	24.40	28.77	30.18	24.32	25.84	0.55	29.36
	Wanda-SP	66.93	76.50	**41.80**	70.82	64.48	38.57	32.19	**44.44**	54.46
	FLAP	65.51	75.84	40.00	69.64	60.82	37.29	30.86	28.65	51.07
	CFSP (ours)	**67.25**	**76.88**	40.60	**72.05**	**68.77**	**41.47**	**36.33**	38.99	**55.29**
50%	*w/o recovery*									
	Magnitude-SP	50.20	48.20	27.00	26.32	26.52	29.10	26.84	0.53	29.34
	Wanda-SP	61.56	66.49	34.80	52.23	45.37	28.07	**25.45**	4.15	39.76
	FLAP	57.62	66.00	32.80	48.09	40.07	27.39	23.04	0.90	36.99
	CFSP (ours)	**61.64**	**67.36**	**35.20**	**53.96**	**48.61**	**30.20**	23.07	**4.35**	**40.55**
	w/ recovery									
	Wanda-SP	63.85	**71.11**	37.80	61.40	57.08	35.04	26.32	**20.90**	46.69
	CFSP (ours)	**65.11**	70.73	**37.00**	**61.96**	**58.88**	**36.26**	**29.46**	20.05	**47.43**

Table 12: Zero-shot performance of pruned models on LLaMA2-7B under 20% and 50% sparsity. For 50% sparsity, we also show the results after recovery fine-tuning. **Bold** indicates the best results under the same setting.

Model	Datasets	WinoGrande	PIQA	OBQA	HellaSwag	ARC-e	ARC-c	MMLU	Average
LLaMA2-7B	Slimpajama	64.01	70.40	36.40	60.65	56.78	33.36	24.95	47.46
	Alpaca-cleaned	64.33	70.24	**37.60**	61.22	**59.30**	34.47	24.33	50.21
	Knowledge-pile	64.80	70.13	36.80	60.45	58.42	34.81	24.55	49.99
	FineWeb-edu	**65.11**	**70.73**	37.00	**61.96**	58.88	**36.26**	**29.46**	**51.34**
LLaMA3-8B	Slimpajama	64.17	70.95	35.00	60.62	57.03	33.62	37.55	51.28
	Alpaca-cleaned	59.67	67.85	34.80	60.97	57.41	35.24	37.89	50.55
	Knowledge-pile	**66.14**	71.38	35.60	60.89	62.05	36.69	38.07	52.97
	FineWeb-edu	65.51	**72.03**	**36.20**	**61.45**	**62.37**	**37.54**	**40.37**	**53.64**

Table 13: Zero-shot performance of various datasets for recovery fine-tuning. All methods are trained with the same tokens (0.1B). **Bold** indicates the best results on each model.

is more appropriate. Finally, we set $\alpha = 1$ for the 7B/8B/13B models and $\alpha = 3$ for the 70B models.

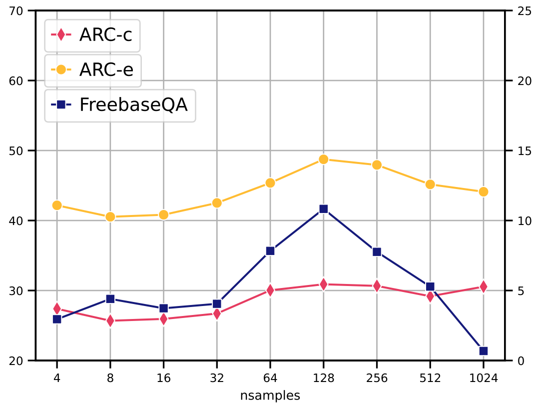

Figure 8: The impact of the size of calibration data. The models are pruned from LLaMA3-8B under 50% sparsity.

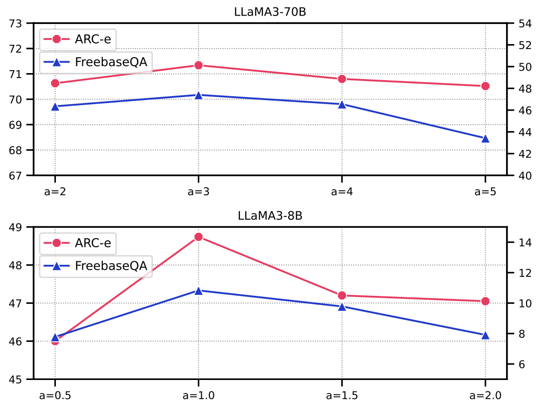

Figure 9: The effect of hyperparameter α in calculating block importance. The models are pruned under 50% sparsity.

Impact of Calibration Data Sizes We investigate the impact of calibration data sizes. Figure 8 presents the results of 3 tasks on LLaMA3-8B with 20% sparsity. We find that the data with 128 examples yield the best overall performance.

Impact of Recovery Data As described in Appendix A.2, we explore various datasets for recovery fine-tuning. As shown in Table 13, FineWeb-Edu consistently outperforms others in a variety of tasks, particularly demonstrating significant improvements in knowledge-intensive tasks such as MMLU and FreebaseQA, which is shown challenging for pruned models (Jaiswal et al., 2024). Thus, we select it for recovery fine-tuning.