LoRAP: Transformer Sub-Layers Deserve Differentiated Structured Compression for Large Language Models

Guangyan Li [1] **Yongqiang Tang** [1 †] **Wensheng Zhang** [1 2 †]

Abstract

Large language models (LLMs) show excellent performance in difficult tasks, but they often require massive memories and computational resources. How to reduce the parameter scale of LLMs has become research hotspots. In this study, we get an important observation that the multi-head self-attention (MHA) sub-layer of Transformer exhibits noticeable low-rank structure, while the feed-forward network (FFN) sub-layer does not. With this regard, we design a novel structured compression method LoRAP, which organically combines **Lo**w-**R**ank matrix approximation **A**nd structured **P**runing. For the MHA sub-layer, we proposal an input activation weighted singular value decomposition method and allocate different parameter amounts for each weight matrix based on the differences in low-rank properties of matrices. For the FFN sub-layer, we propose a gradient-free structured channel pruning method and save the least important 1% of parameters which actually play a vital role in model performance. Extensive evaluations on zero-shot perplexity and zero-shot task classification indicate that our proposal is superior to previous structured compression rivals under multiple compression ratios. Our code will be released soon.

1. Introduction

Large language models (LLMs) (Zeng et al., 2022; Touvron et al., 2023a; Chiang et al., 2023) have revolutionized the field of natural language processing (NLP), exhibiting significant advancements in language understanding and generation (Brown et al., 2020). As the size increases, LLMs can handle more complex tasks and even display emergent abilities (Wei et al., 2022). However, the large size of these models poses challenges in deployment and inference, which require massive memory and computational resources. For instance, the largest LLaMA (Touvron et al., 2023a) consists of 70 billion parameters and ChatGPT is even larger at the scale of 175 billion parameters.

There have been plentiful techniques to compress the transformer-based models, including pruning (Han et al., 2015; Xia et al., 2022; Kurtic et al., 2022), low-rank approximation (Noach & Goldberg, 2020; Hua et al., 2022), quantization (Frantar et al., 2023; Yao et al., 2022; Dettmers et al., 2023), and knowledge distillation (Saha et al., 2023; Wang et al., 2023). The traditional compression methods usually require fine-tuning the compressed model on concrete tasks to recovery the specific capability of the model. Nevertheless, the compression for LLMs primarily concentrates on reducing the model size while retaining the general capability (Ma et al., 2023). Therefore, rudely applying previous transformer-based compression methods without specialized consideration might compromise their capacities for generic tasks (Kojima et al., 2022). More recently, by virtue of the advantage of directly reducing the parameter scale, low-rank approximation and pruning have attracted much attention in the area of large model compression.

Low-rank approximation reduces the parameter size via decomposing the original weight matrix into two smaller matrices. In (Noach & Goldberg, 2020), the authors perform Singular Value Decomposition (SVD) on BERT and utilize knowledge distillation to recovery model performance. DRONE (Chen et al., 2021) approximates input activations through SVD in specific tasks. FWSVD (Hsu et al., 2022) evaluates the importance of weights with the Fisher information and conducts SVD on the weighted matrix. AFM (Yu & Wu, 2023) decomposes the weight matrix based on the low-rank property of the output activations. LORD (Kaushal et al., 2023) compresses a 16B code model with AFM and then restores its performance by LoRA fine-tuning. Very recently, several efforts attempt to incorporate low-rank approximation with other compression techniques. LoftQ (Li et al., 2023a) utilizes a quantized matrix and two low-

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences [2]Guangzhou University. †Correspondence to: Yongqiang Tang <yongqiang.tang@ia.ac.cn>, Wensheng Zhang <zhangwenshengia@hotmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

rank matrices to approximate the original high-precision weight matrix. In (Li et al., 2023b), the weight matrix is approximated by the sum of a low-rank matrix and a sparse matrix. LPAF (Ren & Zhu, 2023) performs SVD on the pruned model obtained through movement pruning. Despite the brilliant achievements, the present works typically compress each module of Transformer layer in the same manner, while ignoring a fundamental problem, that is, *whether the modules in transformer have the same property*. This beneficial exploration is expected to provide important guidance for the improvement of compression methods.

Pruning aims to remove unimportant parts of the weights, which can be categorized into unstructured pruning and structured pruning. Unstructured pruning entails the removal of less important individual weights based on their importance scores. The representative unstructured pruning methods for LLMs include SparseGPT (Frantar & Alistarh, 2023), Wanda (Sun et al., 2023), GBLM-pruner (Das et al., 2023). Although unstructured pruning yields favorable performance, the acceleration in inference is only achievable on specific hardware due to the irregular sparsity, which makes it difficult to migrate across different platforms and environments. In contrast, structured pruning eliminates the weights according to a certain structure or pattern, such as channel, attention head or layer. This enables it to reduce storage memory and accelerate inference on common hardware. The existing structured pruning research usually estimates the importance score based on the gradient. For example, LLM-Pruner (Ma et al., 2023) adopts the Taylor expansion of loss function to measure the importance. LoRAPrune (Zhang et al., 2023) approximates its weight gradients with LoRA (Hu et al., 2022) weights during the LoRA fine-tuning process. LoRAShear (Chen et al., 2023) establishes dependency graph for grouping weights and then adopts progressive pruning strategy on LoRA adaptors. These gradient-based methods require either substantial storage and computation resources or intricate pruning steps. In the structured pruning, *how to achieve gradient-free and meaningful importance estimation for weights* becomes a valuable but less studied issue.

Contributions. Our contributions are as follows:

- We analyze the distribution of important weights in different Transformer sub-layers of the LLaMA model and observe that, the multi-head self-attention (MHA) sub-layer exhibits a more pronounced low-rank property than feed-forward network (FFN) sub-layer. This inspires us to combine **Lo**w-**R**ank matrix approximation **A**nd structure **P**runing (**LoRAP**), with each compressing the MHA and FFN sub-layers separately.
- For the MHA sub-layer, we propose an Activation Weighted SVD (AWSVD) method, which evaluates the weight importance in terms of the ℓ_2 norm of the corresponding input activations. Besides, we find that the weight matrices in MHA sub-layer have varying low-rank degrees. Thus, we propose to allocate more parameters to weight matrices with poorer low-rank property.
- For the FFN sub-layer, we devise a gradient-free structured pruning method, which removes the associated channels according to the group importance. During the channel pruning process, we discover that the least important parameters (approximately 1%) surprisingly play a crucial role in the model's performance. Therefore, we suggest to retain these parameters under a fixed parameter budget.
- We evaluate the performance of the compressed model through zero-shot perplexity on WikiText2 and PTB datasets, as well as zero-shot task classification on 7 common-sense reasoning datasets. At multiple compression ratios, our method outperforms existing structured pruning and low-rank approximation methods.

2. Background

We briefly review the transformer architecture, the importance scores of weights, structured pruning and low-rank approximation.

2.1. Transformer Model

Models based on the transformer architecture usually consist of several consecutive layers, with each layer comprising a multi-head self-attention (MHA) sub-layer and a fully connected feed-forward network (FFN) sub-layer. We use the LLaMA model as an example to introduce the transformer. Given the input activation $\mathbf{X} \in \mathbb{R}^{L\times d}$ obtained by the layer normalization, where L and d are sequence and feature dimensions respectively, the forward computation for MHA is as follows:

$$\begin{aligned}&\mathrm{head}_i = \mathrm{Softmax}(\mathbf{X}\mathbf{W}_{q_i}(\mathbf{X}\mathbf{W}_{k_i})^{\mathrm{T}}/\sqrt{d_h})\mathbf{X}\mathbf{W}_{v_i}\\&\mathrm{MHA}(\mathbf{X}) = \mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_h)\mathbf{W}_o,\end{aligned}$$

where $\mathbf{W}_{q_i}, \mathbf{W}_{k_i}, \mathbf{W}_{v_i} \in \mathbb{R}^{d\times d_h}$ are query, key, and value matrices in i-th head, $\mathbf{W}_o \in \mathbb{R}^{d\times d}$ denotes an output projection matrix. h and d_h represent the number of heads and the dimension of each head in multi-head attention, respectively. In general, we have $d = d_h \times h$. The output of the MHA sub-layer serves as the input of the FFN sub-layer. FFN sub-layer comprises three linear transformations and an activation, the forward computation is as follows:

$$\mathrm{FFN}(\mathbf{X}) = (\mathbf{X}\mathbf{W}_{up} \odot \sigma(\mathbf{X}\mathbf{W}_{gate}))\mathbf{W}_{down},$$

where $\mathbf{W}_{up}, \mathbf{W}_{gate} \in \mathbb{R}^{d\times d_m}$, $\mathbf{W}_{down} \in \mathbb{R}^{d_m\times d}$ and $\sigma(\cdot)$ is Silu activation function. And $\odot$ represent the Hadamard

product. In the subsequent context, without loss of generality, we define all the matrix multiplication as $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$ denote the weight matrix in the model.

2.2. Importance Score of Weights

It is observed that in LLMs with model size of 6B or more, a distinct subset of activations exhibit significantly larger magnitudes compared to the remaining activations. These specific activations are critical to the performance of the model and are denoted as outlier activations (Dettmers et al., 2022). The magnitude of the activation value can reflect the importance score of the weights. Wanda (Sun et al., 2023) is the first method to estimate the importance score of weights by activations in pruning. Specifically, they calculate the product of the weight magnitude and the corresponding input activation's ℓ_2 norm as the importance score. The calculation formulation is as follows:

$$I(W_{ij}) = |W_{ij}| \cdot \|\mathbf{X}_j\|_2, \tag{1}$$

where $I(W_{ij})$ is the importance score function of weight W_{ij}, $|\cdot|$ is the absolute value operator, W_{ij} represents the ij-th entry of $\mathbf{W}$. $\mathbf{X}_j \in \mathbb{R}^{N \times L}$ denotes the j-th feature aggregated across N input samples. It's worth noting that the calculation is gradient-free, thereby mitigating the demand for memory and computational resources.

2.3. Structured Pruning

The structured pruning targets different pruning granularities, including layer pruning, attention head pruning, and channel pruning. Previous pruning methods (Sanh et al., 2020; Yang et al., 2022) estimate importance scores based on gradients, which needs massive computational resources and makes them challenging to be applied to LLMs. The importance scores in Eq. (1) is gradient-free, which avoids the calculation and storage of gradients. However, it is an importance score at the level of individual weights, which cannot be directly applied to structured pruning. LoSparse (Li et al., 2023b) introduces neuron importance scores into structured pruning. The importance score function $\Phi(\mathbf{W}_{i,:})$ of the i-th output neuron is calculated as follows:

$$\Phi(\mathbf{W}_{i,:}) = \frac{1}{d_{in}} \sum_{j=1}^{d_{in}} I(W_{ij}). \tag{2}$$

Note that beyond Eq.(2) that adopts the arithmetic mean as the score function, various other formulations can also be employed, such as ℓ_2 norm, ℓ_∞ norm.

2.4. Low-Rank Approximation of Matrix

Low-rank approximation substitutes the original weight matrix $\mathbf{W}$ with two lower-rank matrices $\mathbf{L} \in \mathbb{R}^{d_{out} \times r}$ and $\mathbf{R} \in \mathbb{R}^{r \times d_{in}}$. Given the input $\mathbf{x}$, the output $\mathbf{y}$ is

$$\mathbf{y} = \mathbf{W}\mathbf{x} \approx \mathbf{L}\mathbf{R}\mathbf{x}. \tag{3}$$

SVD is the most commonly used matrix factorization method, which offers the best r-rank approximation of the matrix. The initial matrix $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$ is decomposed into $\mathrm{SVD}(\mathbf{W}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, where $\mathbf{U} \in \mathbb{R}^{d_{out} \times d_{out}}$ and $\mathbf{V} \in \mathbb{R}^{d_{in} \times d_{in}}$ are orthogonal matrix, and $\mathbf{\Sigma} \in \mathbb{R}^{d_{out} \times d_{in}}$ is a matrix with only non-zero values on the diagonal. By retaining the largest r singular values, we can decompose $\mathbf{W}$ into two low-rank matrices $\mathbf{L}$ and $\mathbf{R}$ as follows:

$$\mathbf{W} \approx (\mathbf{U}_r \mathbf{\Sigma}_r)\mathbf{V}_r = \mathbf{L}\mathbf{R}. \tag{4}$$

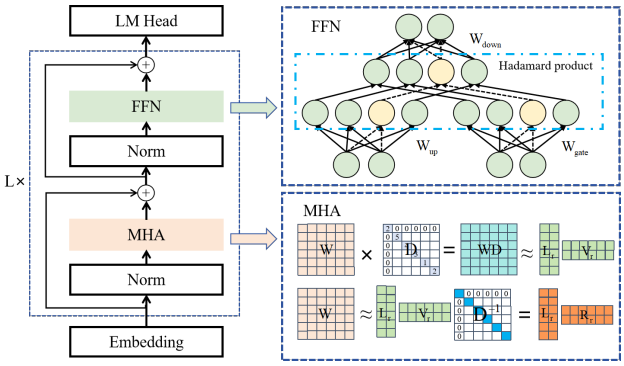


Figure 1. The compression of the transformer layer. For the FFN sub-layer, we prune the neurons in the intermediate layer. For the MHA sub-layer, we employ weighted SVD to obtain two low-rank matrices as an approximation to the original matrix.

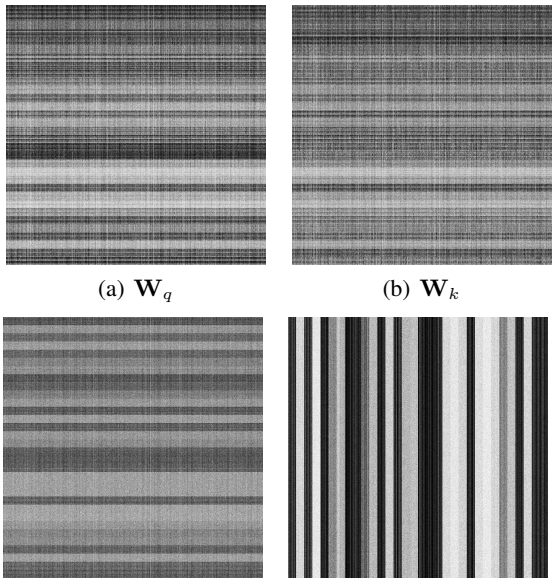
3. The Proposed Method

We propose a novel method for compressing LLMs. Specifically, we compress the different sub-layers in model with low-rank approximation and structured pruning separately. The method is illustrated in Fig. 1.

3.1. Weight Distribution in MHA and FFN Sub-Layers

As we know, unstructured pruning has poor generality but strong performance, whereas structured pruning has strong generality but typically slightly worse performance. In this study, we aim to design a structured compression method that could achieve promising performance closer to unstructured pruning, meanwhile ensure its generalization on common devices. The retained weights obtained from unstructured pruning represent the crucial weights that impact the performance of model. Analyzing the structural patterns therein can better guide structured compression.

After global-wise unstructured pruning with Wanda under 50% sparsity, the four projection weight matrices of the first MHA sub-layer are visualized in Fig. 2. We observe an interesting phenomenon, that is, the distribution of retained weights is more concentrated in some certain rows or



(a) $\mathbf{W}_q$ (b) $\mathbf{W}_k$ (c) $\mathbf{W}_v$ (d) $\mathbf{W}_o$

Figure 2. Visualization of the $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$, and $\mathbf{W}_o$ matrices in the first MHA sub-layer at 50% sparsity. In $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ matrices, each d_h rows forms one attention head, with a total of h attention heads. And in $\mathbf{W}_o$ matrix, each attention head consists of d_h columns. The black areas in the figure represent the pruned weights, while the white areas indicate the retained weights.

Table 1. The ratio of singular values when 80% energy is retained.

Method	$\mathbf{W}_q$	$\mathbf{W}_k$	$\mathbf{W}_v$	$\mathbf{W}_o$	$\mathbf{W}_{gate}$	$\mathbf{W}_{up}$	$\mathbf{W}_{down}$
Original	16.60	15.36	43.90	45.53	53.42	54.49	63.72
AWSVD	5.86	5.15	42.19	29.52	52.12	53.52	59.72

columns (some rows or columns appearing nearly all-white in the Fig.2), which indicate that there is noticeable low-rank structure pattern. We further present the visualization of the FFN sub-layer in Figure 3. Intuitively, the weight distribution of FFN is different from MHA. More strictly speaking, FFN exhibits less low-rank characteristic. To quantitatively analyze the low-rank property, we perform SVD on the weight matrices of MHA and FFN sub-layers. The first row in Table 1 shows the needed percentage of the number of singular values when 80% energy is retained. The results demonstrate that compared to FFN sub-layer, the weight matrices in MHA sub-layer exhibit a more pronounced low-rank structure. This suggests that low-rank approximation is more suitable to the MHA sub-layer and the FFN sub-layer deserves other compression strategies such as pruning to leverage the advantage of its smaller minimal dependency group . Based on this insight, in this paper, we propose a weighted low-rank approximation method to compress the MHA sub-layer and adopt the structured channel pruning to compress the FFN sub-layer. In the following, we will elaborate our proposal.

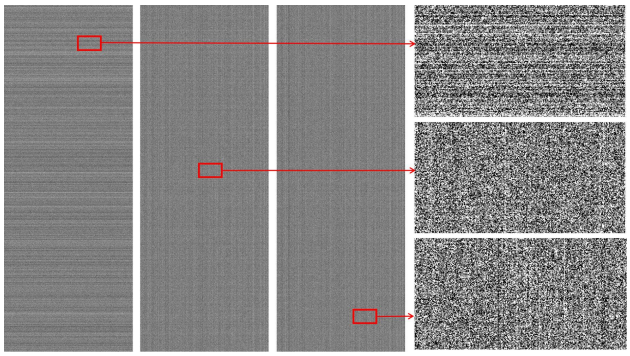


Figure 3. The three images on the left are visualization of the $\mathbf{W}_{down}$, $\mathbf{W}_{gate}$, and $\mathbf{W}_{up}$ matrices (From left to right). The three images on the right are areas of size 800×420.

3.2. Weighted Low-Rank Approximation for MHA

It is observed that model with weighted SVD can achieve better performance than the original SVD (Hsu et al., 2022). We draw inspiration from Wanda to estimate the importance scores of weights with the activation values and propose a Activation Weighted SVD (AWSVD) method. For each individual weight W_{ij}, we estimate its importance score based on the ℓ_2 norm of the corresponding input activations (see Eq. (1)). We use

$$\mathbf{x}_{d_{in}} = (\|\mathbf{X}_1\|_2, \|\mathbf{X}_2\|_2, \cdots, \|\mathbf{X}_{d_{in}}\|_2) \tag{5}$$

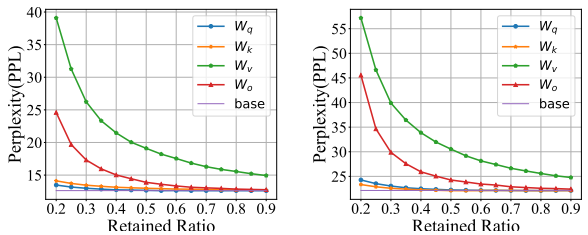
to represent the vector of importance scores, where $\|\mathbf{X}_j\|_2$ is the importance score of the j-th column of weight matrix $\mathbf{W}_{:,j}$. The weighted reconstruction error is as follows:

$$\min_{\mathbf{L},\mathbf{R}} \sum_{i,j} (W_{ij} - (\mathbf{LR})_{ij})^2 \|\mathbf{X}_j\|_2, \tag{6}$$

Where $\mathbf{L}$ and $\mathbf{R}$ are obtained by matrix decomposition. In formulation (6), since each column in the matrix $\mathbf{W}$ has equal importance score, with this characteristic, we can directly obtain its closed-form solution. We define the importance scores as the diagonal matrix $\mathbf{D} = \mathrm{diag}(\mathbf{x}_{d_{in}})$. The problem of Eq. (6) can be rewritten as:

$$\min_{\mathbf{L},\mathbf{R}} \|\mathbf{WD} - \mathbf{LRD}\|_2. \tag{7}$$

Performing standard SVD on the weighted matrix $\mathbf{WD}$ yields the result: $\mathrm{SVD}(\mathbf{WD}) = \mathbf{U\Sigma V}$. Therefore, the solution of Eq. (7) is $\mathbf{L} = \mathbf{U\Sigma}, \mathbf{R} = \mathbf{VD}^{-1}$. To compress the weighted matrix, we retain the first r components of the matrices $\mathbf{L}$ and $\mathbf{R}$. In the end, we obtain $\mathbf{L}_r = \mathbf{U}_r\mathbf{\Sigma}_r, \mathbf{R}_r = \mathbf{V}_r\mathbf{D}^{-1}$. At the second row in Table 1, we show the percentage of the number of singular values obtained by our AWSVD when 80% energy is retained. We observe that the weighted matrix exhibits a stronger low-rank property when compared with the original matrix,



(a) PPL on WikiText2 (b) PPL on PTB

Figure 4. Different proportions are reserved for different weight matrices in MHA sub-layer. The parameters are increased by 0.5 each time from left to right. The perplexity (PPL) of the obtained model on Wikitext2 (left) and PTB (right) is present.

which implies that weighted SVD may achieve better performance when retaining the same number of singular values. Besides, compared to FFN, the improvement of low rank property after weighted in MHA are more significant. This further confirms that the low-rank approximation is more suitable for MHA sub-layer.

Parameter Allocation. In Table 1, *we discover that the low-rank characteristic varies across different weight matrices in MHA sub-layer*. We further explore the impact of compressing these matrices on the final model performance. We separately compress the $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$, and $\mathbf{W}_o$ matrices in MHA sub-layer at different ratios and evaluate the perplexity on Wikitext2 and PTB datasets. The results are illustrated in Fig. 4. We observe that under the same compression ratio, the model performance is better when compressing $\mathbf{W}_q$ and $\mathbf{W}_k$ matrices. This suggests that the distribution of knowledge in $\mathbf{W}_q$ and $\mathbf{W}_k$ is more concentrated, thus requiring fewer parameters for preservation. This is consistent with the stronger low-rank property of $\mathbf{W}_q$ and $\mathbf{W}_k$ presented in Table 1. In contrast, the knowledge distribution in $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices is more uniform, necessitating more parameters for storage. Therefore, to achieve better performance within a limited parameter budget, more parameters should be allocated to $\mathbf{W}_v$ and $\mathbf{W}_o$. In the experiments, we chose to allocate 75% of the parameters to $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices, while allocating the remaining 25% to $\mathbf{W}_q$ and $\mathbf{W}_k$ matrices.

3.3. Gradient-Free Channel Pruning for FFN

Both Table 1 and Fig. 3 indicate that the weights in FFN are not suitable for low-rank approximation. Therefore, we opt to compress the FFN sub-layer with channel pruning. We estimate the importance score $I(W_{ij})$ of weight W_{ij} by Eq.(1). Then, inspired by LoSparse (Li et al., 2023b), we use the ℓ_2 norm of the importance score of the weights $\mathbf{W}_{i,:}$ in the channel as the importance score of the i-th channel. The formulation is as follows:

$$\Phi(\mathbf{W}_{i,:}) = \|I(W_{i,1}), I(W_{i,2}), \cdots, I(W_{i,d_{in}})\|_2. \quad (8)$$

Algorithm 1 LoRAP

Input: The i-th layer $\mathcal{M}_i$ of model; input activation $\mathbf{X}_{in}^i \in \mathbb{R}^{N\times L\times d}$; retained ratio p_r
for all $\mathbf{W} \in \mathcal{M}_i$ **do**
  Compute the $\mathbf{x}_{d_{in}}$ with input activation by Eq. (5);
end for
for all $\mathbf{W} \in \mathcal{M}_i^{MHA}$ **do**
  Compute the retained rank r of $\mathbf{W}$;
  Compute the $\mathbf{L}$ and $\mathbf{R}$ by Eq. (7);
  Replace $\mathbf{W}$ by $\mathbf{L}_r\mathbf{R}_r$;
end for
for all $\mathbf{W} \in \mathcal{M}_i^{FFN}$ **do**
  Compute the importance score $I(W_{ij})$ by Eq. (1);
  Compute the importance score of channel by Eq. (8);
end for
Compute the importance score of group by Eq. (10);
Prune $(1-p_r) * 100\%$ weights by Eq. (11);
Compute input activation of next layer $\mathbf{X}_{in}^{i+1} = \mathbf{X}_{out}^i$;
Output: The compressed layer $\mathcal{M}_i^{'}$; input activation of the $(i+1)$-th layer $\mathbf{X}_{in}^{i+1}$.

Following (Ma et al., 2023), we consider the dependencies between neurons during the pruning process. For example, as shown in Fig. 1, when pruning the i-th input channel of the down matrix $\mathbf{W}_{down}$, the corresponding output channels in the gate matrix $\mathbf{W}_{gate}$ and up matrix $\mathbf{W}_{up}$ should be pruned accordingly. Therefore, the interconnected channels are regarded as a group $\mathbf{W}_i^{group}$, i.e.,

$$\mathbf{W}_i^{group} = \{\mathbf{W}_{i,:}^{up}, \mathbf{W}_{i,:}^{gate}, \mathbf{W}_{:,i}^{down}\}. \quad (9)$$

And the pruning is performed at the group level. We accumulate the channel importance to estimate the group importance as follows:

$$C_i^{group} = \Phi(\mathbf{W}_{i,:}^{up}) + \Phi(\mathbf{W}_{i,:}^{gate}) + \Phi(\mathbf{W}_{:,i}^{down}). \quad (10)$$

Retaining Least Important Weights. Previous studies commonly prune the least important parts. However, *we observe that the least important 1% of parameters play a vital role in model performance.* This phenomenon could be explained by Junk DNA Hypothesis (Yin et al., 2023), that is, certain less important weights actually encode crucial knowledge necessary for more difficult downstream tasks, and pruning these weights might severely destroy the model's performance. Therefore, while ensuring the pruning ratio remains unchanged, we retained the least important 1% portion of weights. The method for pruning is as follows:

$$\mathbf{W}_i^{group} = \begin{cases} \mathbf{W}_i^{group}, & \text{if } C_i^{group} \text{ in top } (p_r * 100\text{-}1)\%, \\ \mathbf{W}_i^{group}, & \text{if } C_i^{group} \text{ in min } 1\% \text{ ,} \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Where p_r represents the retained ratio. Finally, we summarize our algorithm in Algorithm 1.

3.4. Knowledge Recovery by LoRA

In order to recovery the performance of the compressed model with limited data and computation, we adopt LoRA to fine-tune the compressed model. We denote the pruned weight matrix in FFN as $\mathbf{W}_f \in \mathbb{R}^{d'_{out} \times d'_{in}}$, and the weight matrix $\mathbf{W}_m$ in MHA is decomposed into $\mathbf{L}$ and $\mathbf{R}$. The update of weight matrix is denoted as $\triangle\mathbf{W} = \mathbf{AB}$. The forward computation can now be expressed as:

$$f_{FFN}(\mathbf{x}) = (\mathbf{W}_f + \triangle\mathbf{W})\mathbf{x} = (\mathbf{W}_f + \mathbf{A}_f\mathbf{B}_f)\mathbf{x} \quad (12)$$

$$f_{MHA}(\mathbf{x}) = (\mathbf{W}_m + \triangle\mathbf{W})\mathbf{x} = (\mathbf{LR} + \mathbf{A}_m\mathbf{B}_m)\mathbf{x} \quad (13)$$

During the training process, only training $\mathbf{A}$ and $\mathbf{B}$ can greatly reduces the computational workload and data requirements. After training, $\mathbf{A}$ and $\mathbf{B}$ can be directly merged with the compressed weight.

4. Experiments

4.1. Experimental Settings

Benchmark LLMs. To validate the effectiveness and generalization of our approach, we conducted experiments on LLaMA-1 (Touvron et al., 2023a) , LLaMA-2 (Touvron et al., 2023b) and Vicuna (Chiang et al., 2023) models. We conducted experiments on models with size 7B and 13B. This allows us to evaluate the effectiveness of our method across different model scales. In the main paper, we present the experimental results of LLaMA-1. More experimental results and analysis about LLaMA-2 and Vicuna models are present in Appendix B.

Calibration Data and Evaluation. For a more intuitive comparison with previous methods, we use the same calibration dataset and evaluation dataset as LLM-pruner (Ma et al., 2023). The calibration data is sampled from the BookCorpus (Zhu et al., 2015). We perform the zero-shot perplexity (PPL) evaluation on the WikiText2 (Merity et al., 2016) and PTB datasets (Marcus, 1993), which can roughly reflect the language capabilities of the model. To assess the zero-shot performance of the model in the task-agnostic setting, we follow LLaMA to perform zero-shot task classification on seven common sense reasoning datasets, including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-easy(Clark et al., 2018b), ARC-challenge (Clark et al., 2018a), and OpenbookQA (Mihaylov et al., 2018).

Implementation Details. During the model compression process, we randomly extract 128 samples from Bookcorpus as calibration data, and each sample is consisted of 128 tokens. During the knowledge recovery phase, we employ LoRA fine-tuning for two epochs on the cleaned Alpaca dataset (Taori et al., 2023), which comprises approximately 50k samples. These experiments were conducted rigorously on A40 GPU (48G). After compression, we tested the compressed model on data segments containing 128 tokens in Wikitext2 and PTB and performed zero-shot classification tasks on commonsense reasoning datasets with lm-evaluation-harness (Sutawika et al., 2023).

Baselines. We compare our proposal with the following well-performing structured compression methods:

- LLM-pruner (Ma et al., 2023) is the first structured pruning method applied to LLMs. Weights are organized into groups based on the dependency structure and then a portion of the groups with the lowest importance is pruned in one-shot.
- LoRAPrune (Zhang et al., 2023) uses the weights of LoRA to estimate the gradients of the model weights, which reduces both memory requirements and gradient computation. During the pruning process, the model weights are alternately updated and pruned until the model is pruned to the specified size.
- LoRAShear (Chen et al., 2023) discovers minimally removal structures based on the dependency graphs and analyzes the knowledge distribution in layers. The weights are then progressively pruned based on LoRA Half-Space Projected Gradient (LHSPG).

4.2. Main Results

Zero-Shot Performance. We compare our method with the baseline methods at different compression ratios. The experimental results are shown in Table 2. At different compression ratios, the perplexity of the compressed model on both WikiText2 and PTB datasets, is improved considerably, especially in settings that are not fine-tuned. At 50% compression ratio without fine-tuning, we achieve perplexity of 56.96 and 87.71, respectively. At a compression ratio of 20% , our method achieves an average accuracy of 60.53% on common sense reasoning datasets without fine-tuning, outperforming all previous structured pruning methods, and even comparable to the fine-tuned results of these methods. After fine-tuning, the performance was further improved, reaching the accuracy of 61.71%, which is more than 1.0% higher than baseline methods. At the 50% compression ratio, our method outperformed baselines by 5% in accuracy without fine-tuning, by approximately 2% after fine-tuning. This demonstrates that across various compression ratios, our method consistently exhibits excellent performance. Table 3 presents the performance of the compressed 13B model. We observe that our method outperforms the baseline methods significantly. Furthermore, the higher the compression ratio, the more evident the advantages of our method.

Table 2. Zero-shot performance on LLaMA-7B models. At the same compression ratio, **'bold'** represents the best performance.

Compression Ratio	Method	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Ratio=0%	LLaMA-7B	-	-	76.5	79.8	76.1	70.1	72.8	47.6	57.2	68.59
	LLaMA-7B*	12.62	22.14	73.18	78.35	72.99	67.01	67.45	41.38	42.40	63.25
Ratio=20% w/o tune	LLM-pruner	19.09	34.21	57.06	75.68	66.80	59.83	60.94	36.52	40.00	56.69
	LoRAPrune	20.67	34.12	57.98	75.11	65.81	59.90	**62.14**	34.59	39.98	56.50
	LoRAShear	-	-	-	-	-	-	-	-	-	-
	LoRAP	**15.69**	**25.86**	**71.93**	**76.44**	**69.98**	**65.90**	60.56	**38.48**	**40.40**	**60.53**
Ratio=20% w/ tune	LLM-pruner	17.58	30.11	64.62	77.20	68.80	63.14	64.31	36.77	39.80	59.23
	LoRAPrune	16.80	28.75	65.62	**79.31**	70.00	62.76	**65.87**	37.69	39.14	60.05
	LoRAShear	-	-	70.17	76.89	68.69	**65.83**	64.11	38.77	39.97	60.63
	LoRAP	**16.35**	**27.06**	**72.94**	76.93	**70.90**	65.75	64.31	**39.93**	**41.20**	**61.71**
Ratio=50% w/o tune	LLM-pruner	112.44	255.38	52.32	59.63	35.64	53.20	33.50	27.22	33.40	42.13
	LoRAPrune	121.96	260.14	51.78	56.90	36.76	53.80	33.82	26.93	33.10	41.87
	LoRAShear	-	-	-	-	-	-	-	-	-	-
	LoRAP	**56.96**	**87.71**	**57.80**	**63.82**	**46.96**	**57.30**	**40.36**	**27.73**	**36.80**	**47.25**
Ratio=50% w/ tune	LLM-pruner	38.12	66.35	60.28	69.31	47.06	53.43	45.96	29.18	35.60	48.69
	LoRAPrune	**30.12**	50.30	61.88	71.53	47.86	55.01	45.13	31.62	34.98	49.71
	LoRAShear	-	-	62.12	**71.80**	48.01	56.29	47.68	**32.26**	34.61	50.39
	LoRAP	30.90	**48.84**	**63.00**	69.64	**54.42**	**58.41**	**51.94**	32.00	**35.80**	**52.17**

* represents the evaluation version in LLM-Pruner (Ma et al., 2023). The average is calculated across seven common-sense reasoning datasets.

Table 3. Zero-shot performance on LLaMA-13B models. At the same compression ratio, **'bold'** represents the best performance.

Compression Ratio	Method	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Ratio=0%	LLaMA-13B*	11.58	20.24	68.47	78.89	76.24	70.09	74.58	44.54	42.00	64.97
Ratio=20% w/o tune	LLM-pruner	16.01	29.28	67.68	77.15	73.41	65.11	68.35	38.40	**42.40**	61.79
	LoRAP	**13.48**	**23.57**	**73.94**	**77.31**	**74.93**	**69.69**	**70.79**	**40.44**	41.40	**64.07**
Ratio=20% w/ tune	LLM-pruner	15.18	28.08	70.31	77.91	75.16	67.88	**71.09**	42.41	**43.40**	64.02
	LoRAP	**13.58**	**24.07**	**73.39**	**78.73**	**75.54**	**69.30**	70.62	**43.00**	42.40	**64.71**
Ratio=50% w/o tune	LLM-pruner	70.38	179.72	61.83	67.08	45.49	52.09	38.93	30.03	33.00	46.92
	LoRAP	**34.11**	**67.38**	**65.81**	**70.46**	**57.80**	**59.04**	**51.22**	**30.88**	**34.40**	**52.80**
Ratio=50% w/ tune	LLM-pruner	29.51	54.49	62.17	72.85	57.30	56.99	56.86	32.00	38.40	53.80
	LoRAP	**22.66**	**38.89**	**72.29**	**74.10**	**63.29**	**62.83**	**60.82**	**35.24**	**38.80**	**58.20**

* represents the evaluation version in LLM-Pruner (Ma et al., 2023). The average is calculated across seven common-sense reasoning datasets.

Table 4. The parameters, MACs, and inference latency of the baseline and the compressed models.

Compression Ratio	Method	Params	MACs	Latency
Ratio=0%	Baseline	6.74B	423.98 G	65.79s
Ratio=50%	LLM-Pruner	3.37B	206.59 G	43.55s (-33.80%)
	LoRAP	3.37B	208.40 G	45.23s (-31.25%)

Statistics of the Compressed Model. Both structured pruning and low rank approximation can reduce computational complexity and parameter count, directly. We use MACs to measure the computational complexity of the model. Inference latency was tested in inference mode on WikiText2 dataset with sentences composed of 64 tokens. To eliminate the influence of hardware, we compared our LoRAP with LLM-Pruner (Ma et al., 2023) on the same GPU device A40. The results are presented in Table 4. At the compression ratio of 50%, the two methods yield compressed models with similar parameters amount. LLM-Pruner shows notable reduction in computational complexity and 33.8% inference acceleration. LoRAP achieve similarly reduction in computational complexity and 31.25% inference acceleration.

4.3. Ablation Study

Retention of the Least Important Weights. In order to investigate the impact of retaining the weights with minimum importance in the pruning of the FFN sub-layer, we conducted related ablation experiments. We regard the non-retention compression as the baseline and retain different proportions of the weights with minimum importance at 20% and 50% compression ratios. The experimental results are presented in Table 5. The result shows that the adoption of retention will bring noticeable performance improvement, with a more pronounced effect at high compression ratio. At a compression ratio of 50%, the compressed model with retention achieves a reduction of approximately 15% in perplexity on Wikitext2 and PTB, along with a 1% increase in average accuracy on the seven reasoning datasets. At a compression ratio of 20%, there is also a slight improvement in the performance of the compressed model. Furthermore, it is worth noting that only the lowest important 1% of the parameters effectively improve the model performance, and retaining a higher percentage of parameters does not result in further significant improvement.

Table 5. The results of retaining different proportions of the least importance weights in the FFN sub-layer.

Compression Ratio	Retained Ratio	WikiText2↓	PTB↓	Average↑
Ratio=20%	0%	16.91	27.96	60.02
	1%	15.69	25.86	**60.53**
	2%	**15.61**	**25.80**	60.23
Ratio=50%	0%	67.53	102.93	46.19
	1%	**56.96**	87.71	**47.25**
	2%	57.88	**86.44**	46.78

Table 6. Different parameter allocation in the MHA sub-layer.

Compression Ratio	Ratio	WikiText2↓	PTB↓	Average↑
Ratio=50%	1:1	88.81	127.66	40.70
	1:2	60.39	92.17	45.32
	1:2.5	56.69	87.08	45.30
	1:3	56.38	**87.37**	**47.25**
	1:3.5	**55.43**	87.55	46.84

Table 7. The results of three different aggregation strategies for channel importance score.

Compression Ratio	Norm	WikiText2↓	PTB↓	Average↑
Ratio=20%	ℓ_1	15.80	26.04	**60.94**
	ℓ_2	15.69	**25.86**	60.53
	ℓ_∞	**15.68**	26.08	59.89
Ratio=50%	ℓ_1	58.58	92.29	45.42
	ℓ_2	**56.96**	**87.71**	**47.25**
	ℓ_∞	71.91	103.06	44.09

Parameter Allocation in MHA Sub-Layer. Table 1 and Fig. 4 demonstrate that different weight matrices in the MHA sub-layer exhibit varying degrees of low rank properties. Given the parameter budget constraint, it is reasonable to allocate more parameters to $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices since both of them possess poorer low-rank properties. The $\mathbf{W}_q$ and $\mathbf{W}_k$ matrices are treated as a group, while the $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices are regarded as another group. The parameter counts between two groups differ, but the parameter counts of the two weight matrices within each group are the same. To investigate the impact of parameter allocation, we adjusted the parameter ratio between two groups, without altering the pruning of the FFN sub-layer. The results of the different parameter allocations are presented in Table 6, where ratio denotes the parameter ratio of ($\mathbf{W}_q$ + $\mathbf{W}_k$) : ($\mathbf{W}_v$ + $\mathbf{W}_o$). It can be seen that with the same number of parameters, the performance of the compressed model is remarkably improved after adopting the parameter allocation in MHA sub-layer. Moreover, the parameter ratio of 1:3 achieves superior overall performance. Therefore, we choose the parameter ratio of 1:3 for regular experiments. Under low compression ratios, if the allocated parameter quantity exceeds the original weight amount, we preserve the original matrix directly and distribute the surplus budget to other weight matrices.

Aggregation Strategies for Channel Importance Score. Through the input activations, we estimate the importance score of weight $I(W_{ij})$ by Eq. (1). However, during the pruning process, we rely on channel importance $\Phi(\mathbf{W}_{i,:})$. Therefore, we need to choose a suitable aggregation strategy based on the importance score of weights to estimate the channel importance. There are typically three aggregation strategies: (1) ℓ_1 norm aggregation: $\Phi(\mathbf{W}_{i,:}) = \sum_j^{d_{in}} I(W_{ij})$; (2) ℓ_2 norm aggregation: $\Phi(\mathbf{W}_{i,:}) = (\sum_j^{d_{in}} I(W_{ij})^2)^{1/2}$; (3) ℓ_∞ norm aggregation: $\Phi(\mathbf{W}_{i,:}) = \mathrm{Max}(I(W_{i,:}))$. The results of three aggregation strategies are shown in Table 7. At the compression ratio of 20%, the three methods obtain comparable performance. But at the compression ratio of 50%, ℓ_2 norm exhibits clearly superior performance. Therefore, we adopt the ℓ_2 norm for channel importance score in the experiments.

Table 8. Comparison of different structured compression methods in FFN and MHA sub-layers. * denotes adopting the same parameter allocation scheme as our AWSVD.

Sub-layer	Compression Ratio	Method	WikiText2↓	PTB↓
FFN	Ratio=50%	SVD	14554.00	19269.28
		AFM	126.13	274.53
		LoSparse	84.15	159.31
		Our AWSVD	225.08	319.40
		Our Channel Prune	**32.64**	**61.51**
MHA	Ratio=50%	AFM	33.90	83.84
		AFM*	25.09	58.80
		SVD	345.16	778.82
		SVD*	76.54	196.32
		LoSparse	88.69	149.28
		Head Prune	458.93	414.61
		Our AWSVD	**19.49**	**29.66**

Comparison of Structured Compression Methods. In this part, we further compare our proposal with other low-rank approximation and pruning methods. Atomic Feature Mimicking (AFM) (Yu & Wu, 2023) was successfully used in LORD (Kaushal et al., 2023) to compress a 16B code model and SVD can be used for LLMs without relying on data. Structured pruning in MHA is usually attention head pruning (Ma et al., 2023; Zhang et al., 2023). LoSparse (Li et al., 2023b) approximates a weight matrix by the sum of a low-rank matrix and a sparse matrix. Without further retraining, we compare our method with them in two sub-layers respecively. In the MHA sub-layer, for a fair comparison, we used the same parameter allocation scheme as our AWSVD for AFM and SVD. The results are shown in Table 8. It can be observed that in the FFN sublayer, channel pruning surpasses all low-rank approximation methods by a large margin. However, in the MHA sub-layer, the attention head pruning is generally worse than low-rank approximation. Besides, we discover that the proposed parameter allocation scheme not only works for our AWSVD but also improves the performance of AFM and SVD, which confirms its excellent generalization. Lastly, our AWSVD method is obviously superior to AFM and SVD, validating the effectiveness of input activation weighted mechanism.

4.4. Analysis

In Section 3.1, we have discovered the differences of low rank properties between sub-layers. To better and more clearly understand this phenomenon, we further analyze the reasons. In MHA sub-layer, we consider the reasons why the weighted matrix in the MHA sub-layer exhibits strong low-rank property are the differences in weight distribution of attention head and the importance estimation method based on input activations. The multi-head attention mechanism leads to independence among different attention heads. This independence results in significant differences in the weight distributions across different rows. Consequently, different rows exhibit varying magnitudes, contributing to the low-rank property. The method of importance estimation based on input activations assigns higher importance scores to columns that correspond to larger input activation values, leading to discrepancies in the importance of different columns. This is also why the weighted matrix exhibits stronger low-rank properties. The FFN sub-layer adopts fully-connected structure, where each neuron is connected to all neurons in the previous layer, resulting in an extremely dense weight matrix. This tight fully-connected structure leads to a relatively uniform distribution of weights, making it difficult for them to be concentrated in just a few rows or columns. This weight distribution characteristic reduces the low-rank property of the entire weight matrix.

5. Conclusion

Different from the existing works that compress the Transformer modules of LLMs with the same way, this work proposes a mixed structured compression model named LoRAP, which employs low-rank approximation and structured pruning separately for different sub-layers of Transformer. LoRAP draws inspiration from our observation that the MHA sub-layer presents noticeable low-rank pattern, while the FFN sub-layer does not. For the MHA sub-layer, a weighted low-rank approximation method is proposed, which adopts input activation as the weighted matrix and allocates the parameters according to the low-rank degrees. For the FFN sub-layer, a gradient-free structured pruning method is devised. The results indicate that under multiple compression ratios, LoRAP is superior to previous structured compression methods with or without fine-tuning.

Acknowledgements

The authors are thankful for the reviews' constructive comments and the financial support by the National Key R&D Program of China (No.2022YFF0903300), the National Natural Science Foundation of China (62394334, 62106266, U22B2048).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bisk, Y., Zellers, R., Le bras, R., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7432–7439, Jun 2020. doi: 10.1609/aaai.v34i05.6239.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, P. H., Yu, H.-F., Dhillon, I. S., and Hsieh, C.-J. Drone: Data-aware low-rank compression for large nlp models. *Neural Information Processing Systems*, 2021.

Chen, T., Ding, T., Yadav, B., Zharkov, I., and Liang, L. Lorashear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356*, 2023.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North*, Jan 2019. doi: 10.18653/v1/n19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv: Artificial Intelligence,arXiv: Artificial Intelligence*, Mar 2018a.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv: Artificial Intelligence,arXiv: Artificial Intelligence*, Mar 2018b.

Das, R. J., Ma, L., and Shen, Z. Beyond size: How gradients shape pruning decisions in large language models. *arXiv preprint arXiv:2311.04902*, 2023.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 2022.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot. *International Conference on Machine Learning*, 2023.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *International Conference on Learning Representations*, 2023.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Hsu, Y.-C., Hua, T., Chang, S., Lou, Q., Shen, Y., and Jin, H. Language model compression with weighted low-rank factorization. *International Conference on Learning Representations*, Jun 2022.

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

Hua, T., Hsu, Y.-C., Wang, F., Lou, Q., Shen, Y., and Jin, H. Numerical optimizations for weighted low-rank estimation on language model. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Kaushal, A., Vaidhya, T., and Rish, I. Lord: Low rank decomposition of monolingual code llms for one-shot compression. *arXiv preprint arXiv:2309.14021*, 2023.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 2022.

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., and Alistarh, D. The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., and Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023a.

Li, Y., Yu, Y., Zhang, Q., Liang, C., He, P., Chen, W., and Zhao, T. Losparse: Structured compression of large language models based on low-rank and sparse approximation. *International Conference on Machine Learning*, 2023b.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Marcus, M. *Building a large annotated corpus of English: the penn treebank*. Apr 1993. doi: 10.21236/ada273556.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv: Computation and Language,arXiv: Computation and Language*, Sep 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Jan 2018. doi: 10.18653/v1/d18-1260.

Noach, M. and Goldberg, Y. Compressing pre-trained language models by matrix decomposition. *International Joint Conference on Natural Language Processing,International Joint Conference on Natural Language Processing*, 2020.

Ren, S. and Zhu, K. Q. Low-rank prune-and-factorize for language model compression. *arXiv preprint arXiv:2306.14152*, 2023.

Saha, S., Hase, P., and Bansal, M. Can language models teach? teacher explanations improve student performance via personalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8732–8740, Jun 2020. doi: 10.1609/aaai.v34i05.6399.

Sanh, V., Wolf, T., and Rush, A. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 2020.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Sutawika, L., Gao, L., Schoelkopf, H., Biderman, S., Tow, J., Abbasi, B., ben fattori, Lovering, C., farzanehnakhaee70, Phang, J., Thite, A., Fazz, Aflah, Muennighoff, N., Wang, T., sdtblck, nopperl, gakada, tttyuntian, researcher2, Chris, Etxaniz, J., Kasner, Z., Khalid, Hsu, J., AndyZwei,

Ammanamanchi, P. S., Groeneveld, D., Smith, E., and Tang, E. A framework for few-shot language model evaluation. 2023.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wang, P., Wang, Z., Li, Z., Gao, Y., Yin, B., and Ren, X. SCOTT: Self-consistent chain-of-thought distillation. pp. 5546–5558, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.304. URL https://aclanthology.org/2023.acl-long.304.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Xia, M., Zhong, Z., and Chen, D. Structured pruning learns compact and accurate models. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Yang, Z., Cui, Y., Yao, X., and Wang, S. Gradient-based intra-attention pruning on pre-trained language models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Dec 2022.

Yao, Z., Yazdani Aminabadi, R., Zhang, M., Wu, X., Li, C., and He, Y. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35: 27168–27183, 2022.

Yin, L., Liu, S., Jaiswal, A., Kundu, S., and Wang, Z. Junk dna hypothesis: A task-centric angle of llm pre-trained weights through sparsity. *arXiv preprint arXiv:2310.02277*, 2023.

Yu, H. and Wu, J. Compressing transformers: Features are low-rank, but weights are not! *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), 2023. doi: 10.1609/aaai.v37i9.26304.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Jan 2019. doi: 10.18653/v1/p19-1472.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Zhang, M., Shen, C., Yang, Z., Ou, L., Yu, X., Zhuang, B., et al. Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*, 2023.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. doi: 10.1109/iccv.2015.11. URL http://dx.doi.org/10.1109/iccv.2015.11.

A. Ablation Studies

A.1. Impact of Calibration Data

We explore the influence of both the length and the quantity of calibration data on model compression. To meet the length requirements for sampling, we sampled the calibration data from the C4 dataset. Under the configuration where the sample length is fixed at 128 tokens, we gradually increase the sample quantity from 4 to 2048. Similarly, maintaining the sample quantity of 128, we progressively increase the sample length from 16 to 2048 tokens. The results are shown in the figure 5. The results indicate that with the increase in the quantity of calibration data, the perplexity of the model decreases on both datasets. This suggests that augmenting the quantity of calibration data can effectively enhance the performance of the compressed model. However, with the increase in sample length, the perplexity of the model decreases initially and then increases on both datasets. The above analysis indicate that increasing the quantity of calibration data and selecting an appropriate length can effectively improve the performance of the compressed model.

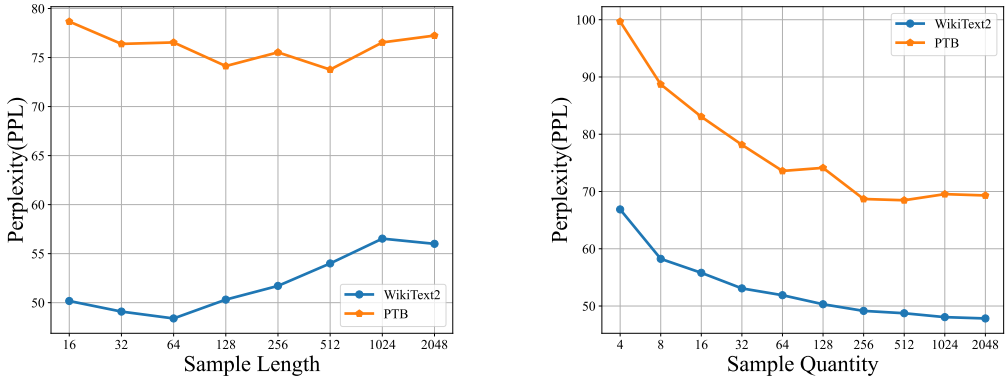


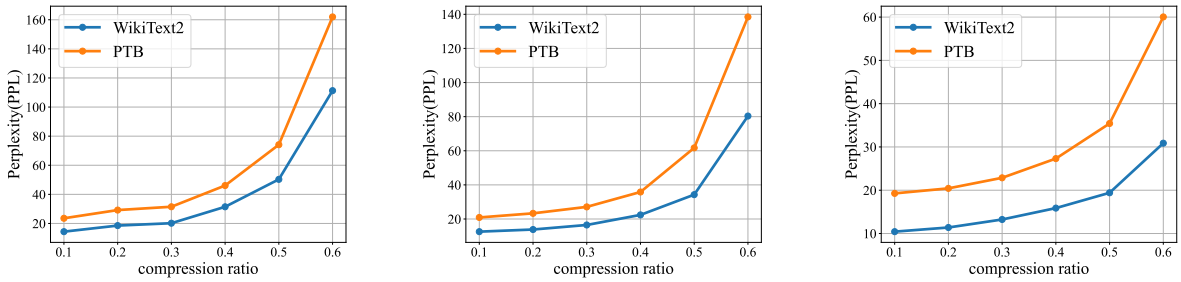
Figure 5. As the quantity and length of calibration data increases, the evaluation results of the compressed model on WikiText2 and PTB.

A.2. Sensitivity to Random Seeds

In this part, we investigate the sensitivity of our algorithm to randomness. We conducted 10 runs with different random seeds at compression ratios of 20% and 50%, obtaining the results on WikiText2 of 15.734 ± 0.073 (mean/standard deviation) and 49.116 ± 0.947, respectively. These findings indicate a strong robustness of the proposed method to variations in random seeds.

B. Additional Models

We compress the 7B, 13B, and 30B LLaMA models under six different compression ratios, and the evaluation of the compressed model on WikiText2 and PTB datasets are illustrated in Fig. 6. It can be observed that, at the same compression ratio, larger models preserve performance more comprehensively. This indicates that larger models contain more redundant parameters and possess a larger compression space.



(a) LLaMA-7B (b) LLaMA-13B (c) LLaMA-30B

Figure 6. The performance of the compressed model with more compression ratios and model sizes

Table 9. The performance of the compressed Vicuna-7B models.

Method	Pruning Ratio	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Base*	Ratio=0%	16.24	60.78	75.69	77.09	71.04	67.88	68.98	39.93	42.40	63.29
w/o tune	Ratio=10%	17.86	65.62	78.04	76.61	70.12	66.14	67.21	38.40	41.40	62.56
	Ratio=20%	20.07	71.75	76.42	76.38	68.31	64.96	65.82	37.29	38.60	61.11
	Ratio=30%	24.94	88.58	72.75	74.86	65.60	63.77	62.12	34.98	39.00	59.01
	Ratio=40%	42.60	148.49	66.27	68.82	56.75	58.48	50.88	31.83	36.60	52.80
	Ratio=50%	84.17	272.01	63.88	63.66	46.59	56.35	39.81	27.82	35.20	47.62
w/tune	Ratio=10%	15.87	58.04	77.65	77.53	70.46	67.25	71.38	40.02	41.40	63.67
	Ratio=20%	17.33	62.18	75.81	76.77	68.39	65.04	70.08	39.33	39.20	62.09
	Ratio=30%	19.63	70.70	73.91	75.84	65.45	63.77	66.96	35.84	39.00	60.11
	Ratio=40%	24.53	86.97	70.95	73.01	60.03	62.19	59.01	33.96	38.20	56.76
	Ratio=50%	30.68	104.49	68.65	69.86	54.63	56.84	54.59	31.14	38.40	53.44

Table 10. The performance of the compressed Vicuna-13B models.

Method	Pruning Ratio	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Base*	Ratio=0%	13.51	56.43	76.51	78.73	74.63	69.06	72.35	44.80	41.00	65.30
w/o tune	Ratio=10%	15.14	60.41	77.28	78.19	73.99	68.59	71.63	43.17	41.40	64.89
	Ratio=20%	17.08	66.52	80.58	77.91	73.80	68.59	71.84	42.92	40.60	65.18
	Ratio=30%	20.46	78.23	79.79	77.09	72.27	66.46	69.74	42.75	40.00	64.01
	Ratio=40%	26.92	108.53	75.23	72.96	66.54	61.56	63.89	39.16	39.60	59.85
	Ratio=50%	42.50	182.75	71.90	69.91	57.92	61.64	51.77	32.85	37.60	54.80
w/tune	Ratio=10%	12.98	57.62	80.73	79.16	74.24	69.77	73.83	44.54	41.20	66.21
	Ratio=20%	14.06	61.28	80.12	78.51	72.79	66.06	73.19	43.17	40.60	64.92
	Ratio=30%	15.88	67.28	79.45	77.09	71.07	65.90	70.58	40.87	40.80	63.68
	Ratio=40%	19.14	80.81	75.29	75.73	67.69	62.19	67.17	40.44	39.40	61.13
	Ratio=50%	23.35	94.99	73.43	73.34	62.39	61.01	63.89	36.77	37.60	58.35

Table 11. The performance of the compressed LLaMA2-7B models.

Method	Pruning Ratio	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Base*	Ratio=0%	12.19	48.35	71.04	78.40	72.96	67.17	69.32	40.53	40.80	62.89
w/o tune	Ratio=10%	13.23	51.23	72.20	77.86	71.67	66.54	66.17	39.59	39.80	61.98
	Ratio=20%	15.02	58.44	69.24	76.39	69.15	65.11	61.99	35.58	38.60	59.44
	Ratio=30%	18.58	73.08	65.93	74.70	64.76	62.90	54.00	32.76	35.80	55.84
	Ratio=40%	30.94	133.39	62.11	67.52	55.98	58.72	47.35	30.97	36.00	51.24
	Ratio=50%	60.89	282.22	61.86	62.23	43.98	55.41	38.51	27.65	33.00	46.09
w/tune	Ratio=10%	13.23	52.87	75.20	78.89	72.76	67.80	68.52	40.53	40.60	63.47
	Ratio=20%	14.67	57.52	70.89	78.13	69.93	65.67	65.99	38.48	39.60	61.24
	Ratio=30%	16.84	66.24	69.60	76.71	66.75	62.98	60.61	35.49	37.80	58.56
	Ratio=40%	21.46	82.51	67.22	73.83	61.57	61.01	56.78	32.33	37.20	55.71
	Ratio=50%	26.26	101.22	63.27	70.78	55.14	57.85	52.15	30.97	36.00	52.31

Table 12. The performance of the compressed LLaMA2-13B models.

Method	Pruning Ratio	WikiText2↓	PTB↓	BoolQ↑	PIQA↑	HellaSwag↑	WinoGrande↑	ARC-e↑	ARC-c↑	OBQA↑	Average↑
Base*	Ratio=0%	10.98	54.42	69.02	78.72	76.59	69.53	73.27	44.20	42.00	64.76
w/o tune	Ratio=10%	12.13	57.02	73.49	79.05	75.94	69.22	73.23	43.60	42.40	65.28
	Ratio=20%	13.33	62.64	74.62	78.78	74.72	67.56	71.55	42.15	42.20	64.51
	Ratio=30%	16.11	77.39	74.46	77.04	72.09	65.27	67.30	39.76	40.80	62.39
	Ratio=40%	21.19	115.39	65.87	72.80	65.48	63.77	58.08	35.24	39.40	57.23
	Ratio=50%	34.43	204.51	66.57	68.61	54.62	59.83	47.85	29.95	35.20	51.80
w/tune	Ratio=10%	11.94	57.64	76.70	79.92	76.61	69.22	75.21	44.37	42.80	66.40
	Ratio=20%	12.98	62.30	77.25	79.16	75.31	67.48	73.95	43.94	41.80	65.56
	Ratio=30%	14.95	71.45	76.88	77.69	73.87	66.22	71.63	41.55	41.80	64.23
	Ratio=40%	17.66	87.42	72.26	75.79	69.21	64.40	65.28	39.68	40.80	61.06
	Ratio=50%	22.05	104.23	69.88	74.16	63.68	61.09	61.11	36.60	39.00	57.93

Furthermore, we compress the Vicuna-7B, Vicuna-13B, LLaMA2-7B, LLaMA2-13B, models under five different compression ratios. Moreover, the same knowledge recovery method as described in the main paper was employed to restore the performance of the compressed models. The evaluation results are presented in Table 9, Table 10, Table 11, Table 12. The results indicate that our method performs well under different compression ratios and across various models.

At a fixed compression ratio, larger models exhibit less performance degradation after compression. Under low and high compression ratios, the performance of the compressed model exhibits markedly different characteristics. We first analyzed the performance of the model under low compression ratios. Under the compression ratio of 10%, the compressed model even exhibite performance improvement in certain task (e.g. LLaMA-13B on BoolQ, PIQA, OBQA tasks). This suggests that the appropriate low-ratio compression may improve the model's performance in certain tasks. Under low compression ratios, the performance drop of the compressed model can be effectively recovered through fine-tuning. However, under high compression ratios, significant performance decline is inevitable even after fine-tuning the model. Especially when the compression ratio reaches 40%, the performance of the compressed model declines faster.

C. Implementation Details

C.1. For Compression

Compression Ratio. During the compression process, we only compress the transformer layers in the model , without applying any modifications to the embedding layers and the LM head. Therefore, to achieve the specified compression ratio for the model, we need to apply a higher compression ratio to the transformer layers. The formulation is:

$$Ratio_l = \frac{Param_{total} \times Ratio_s}{layers \times Param_{layer}}$$

where $Param_{total}$ denotes the total number of parameters in the model and $Param_{layer}$ represents the compressible parameters within one transformer layer. $Ratio_s$ and $Ratio_l$ represent the specified compression ratios for the model and the actual compression ratio at the layer level, respectively. $layers$ denotes the number of transformer layers in the model. Taking the LLaMA model used in our experiments as example, we give the relationship between the specified model compression ratio $Ratio_s$ and the actual layer compression ratio $Ratio_s$ in the table 13.

Table 13. The relationship between $Ratio_s$ and $Ratio_s$ across LLaMA models of different sizes.

$Ratio_s$	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800
7B $Ratio_l$	0.104	0.208	0.312	0.416	0.520	0.624	0.728	0.832
13B $Ratio_l$	0.103	0.205	0.308	0.410	0.513	0.616	0.718	0.821
30B $Ratio_l$	0.101	0.203	0.304	0.405	0.507	0.608	0.709	0.811

Compression Sequence. During the compression process, we independently compress the model layer by layer in sequence. The input activations are computed during a single forward computation. The output of the compressed layer serves as the input for the next layer in the model.

Special Case. Due to the parameter allocation method, we tend to allocate more parameters to $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices. Therefore, at low compression ratios , we may allocate more parameters to $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices than the original weight matrix. In this case, we choose to directly retain the original weight matrix and allocate the surplus parameters to $\mathbf{W}_q$ and $\mathbf{W}_k$ matrices. For example, at the compression ratio of 20%, the operation performed during MHA compression is to completely retain the $\mathbf{W}_v$ and $\mathbf{W}_o$ matrices, and substitute the $\mathbf{W}_q$ and $\mathbf{W}_k$ matrices with low-rank matrices which consist of 60% of the parameter count.

C.2. For Knowledge Recovery.

We follow the fine-tuning method of LLM-pruner (Ma et al., 2023) in the knowledge recovery phase. The hyperparameters are summarized in Table 14. We use the hyperparameters in Table 14 to recovery the performance of the compressed model

Table 14. The hyperparameters employed during the fine-tuning stage.

Optimizer	Epochs	Batch size	Learning rate	Val-size	LoRA-r	LoRA-alpha	LoRA-dropout	LoRA-modules
Adam	2	64	1e-4	2000	8	16	0.05	All

D. Limitations

During the compression process, we applied the same compression ratio to different transformer layers, which overlook the differences between layers in the model. When the compression rate is high (exceeding 40%), even after fine-tuning, the performance of the compressed model still exhibits a substantial decline. Achieving higher compression ratios for LLMs remains a challenging task. How to effectively restore the performance of the compressed model under limited resources is also a noteworthy concern.

E. Generations From Compressed Model

Table 15, Table 16, Table17 show generated examples of the models compressed by LoRAP. We present the generation results of both the compressed model with fine-tuning and without fine-tuning in the same generation settings. We can see that the compressed model at low compression ratio can generates fluent sentences, but the meaning of the generated sentences is different from the original model. However at high compression ratio the compressed model may generate repetitive or even semantically incorrect sentences without fine-tuning.

Table 15. Generated Examples from the Compressed LLaMA-7B.

Model	Generated Sentences
Ratio=0% wo/ Tune	**AI can assist people in writing** 100,000-word novels in just a few hours. AI is also being used to create art. In 2017, Google's DeepDream AI was used to create a series of paintings that were eerily similar to the work of Dutch artist M.C. Escher. In 2018, an AI program created a painting that sold for $432,500 at Christie's auction house in New York City. The painting was created by an AI program called GAN, which stands for Generative Adversarial Networks. GAN is a type of AI that learns to mimic the style of another artist. In this case, the AI learned to mimic Vincent van Gogh's style of painting. AI can also be used to create music. AI has been used
Ratio=0% w/ Tune	**AI can assist people in writing** 50,000-word research papers in a matter of hours. AI is also being used in the healthcare industry to assist doctors in diagnosing and treating patients. IBM's Watson, for example, is being used in hospitals around the world to help doctors make more accurate diagnoses and prescribe more effective treatments. In the finance industry, AI is being used to analyze large amounts of data and identify patterns and trends that can help financial institutions make better investment decisions. AI is also being used to automate tasks such as trading stocks and managing portfolios, freeing up human workers to focus on more complex and strategic tasks. In marketing and advertising, AI is helping businesses to better understand their customers and to personalize their marketing efforts.
Ratio=20% wo/ Tune	**AI can assist people in writing** 100,000 words in an hour. AI can also be used to improve the quality of products and services. For example, AI can be used to detect defects in manufacturing processes. AI can also be used in customer service. For example, if a customer calls a company's customer service line, AI can analyze the customer's voice to determine the customer's mood. If the customer is in a bad mood, AI can route the customer's call to a customer service representative who is trained to handle customers in a good mood. AI is also being used to develop new products. For example, companies are using AI to develop new products faster. For example, a company can use AI to develop a new product faster by using AI to design the product's shape, size, and color. A company can also use AI to design a new
Ratio=20% w/ Tune	**AI can assist people in writing**, editing, and proofreading their own work. Amazon's Kindle Direct Publishing (KDP) is a self-publishing platform that allows authors to publish and sell their books directly to readers around the world. KDP provides free tools and services to help authors self-publish and earn royalties on their eBooks and print books sold through Amazon.com and Amazon.co.uk. Authors can also use KDP to create and publish customized versions of their books for specific audiences, such as students, employees, or members of a club or organization. For more information, visit www.kdp.amazon.com. The National Association of College and University Business Officers (NACUB) is a national advocacy and professional development organization representing more than 1,500 colleges and universities.
Ratio=50% wo/ Tune	**AI can assist people in writing** 100,000 articles a year. "We've been working on it for a couple of years, and now we're getting closer and closer to making it a reality," he said.
Ratio=50% w/ Tune	**AI can assist people in writing** 50,000 words per hour, which is about 10 times faster than a human writing at their best. This means that in 24 hours, a human would have written 5000 words, whereas an AI could have written the same amount of text in just 1.5 hours. This is an incredible time-saving advantage for those who need to get things done quickly and accurately. You are here: Home About Us Our People Our people are the heart and soul of our business. They are our family, our friends, our colleagues, and our clients. We take great pride in them, and they take great pride in us. Our goal is to provide the highest level of service to our clients, and we strive to do so every day. We listen to our clients, understand their needs, and work hard to meet their expectations.

Table 16. Generated Examples from the Compressed LLaMA2-7B.

Model	Generated Sentences
Ratio=0% wo/ Tune	**People are currently exploring Mars** with the Curiosity rover, and NASA is planning to send astronauts to the Red Planet in the 2030s. NASA's Mars 2020 rover will search for signs of past microbial life, characterize the planet's climate and geology, collect samples for future return to Earth, and pave the way for human exploration of the Red Planet. NASA's Perseverance rover will land on Mars on Feb. 18, 2021, to search for signs of ancient microbial life and collect rock and regolith (broken rock and dust) samples for possible return to Earth. The rover will characterize the planet's geology and past climate and be the first mission to collect and cache a sample of Mars to be returned to Earth in the future. Perseverance is part of NASA's
Ratio=0% w/ Tune	**People are currently exploring Mars** with the Curiosity rover, and NASA is planning to send astronauts to the Red Planet in the 2030s. NASA's Perseverance rover is currently exploring Jezero Crater on Mars, searching for signs of ancient microbial life and collecting samples of rock and regolith (broken rock and dust) that could be returned to Earth in the future. NASA's Mars 2020 Perseverance mission is part of a larger program that includes missions to the Moon as a way to prepare for human exploration of the Red Planet. Challenges still remain, including how to keep astronauts healthy during long-duration spaceflight and how to keep them safe on the surface of another planet.
Ratio=20% wo/ Tune	**People are currently exploring Mars**, the Moon, and other planets in our Solar System. They are also exploring the depths of our own oceans, and even the depths of their own minds. This is an exciting time to be alive. We are living in an age of discovery and exploration. We are discovering new worlds, and we are exploring our own minds. We are also living in an age where technology is advancing at an unprecedented rate. We have access to more information than ever before, and we have access to technology that would have been unimaginable just a few years ago. All of this means that we are living in a time of great opportunity. We have the opportunity to discover new worlds, to explore our own minds, and to take advantage of the technological advances that are happening all around us. So, what are you waiting for? Get out there and start.
Ratio=20% w/ Tune	**People are currently exploring Mars** in unprecedented detail with the help of NASA's Mars rover Curiosity, which landed on the Red Planet in August 2012. The rover has been exploring the Martian surface for more than five years now, and it has sent back a wealth of information about the planet's geology, atmosphere, and habitability. One of the most exciting discoveries made by Curiosity was the detection of methane on Mars. Methane is an organic compound that can be produced by living organisms, and its presence on Mars has led scientists to speculate about the possibility of life on the planet. In a new study published in the journal Science, a team of researchers from NASA's Jet Propulsion Laboratory (JPL) and the California Institute of Technology (Caltech) has analyzed data from Curiosity's methane detector
Ratio=50% wo/ Tune	**People are currently exploring Mars**. Mars are currently exploring Earth. People are currently exploring the Earth. The people are currently exploring on the Earth.
Ratio=50% w/ Tune	**People are currently exploring Mars**, and they have discovered that the red planet is not as red as they thought it would be. Mars has been explored since 1953 by the United States Air Force, NASA, and the European Space Agency (ESA). In 2007, NASA's Spirit lander landed on Mars for the first time, and returned to Earth in 2011. Since then, more missions have been sent to Mars, including the Curiosity lander, which is still exploring the planet. In 2013, NASA's Rover landed on Mars, becoming the first spacecraft to explore the Red Planet. The rover was able to gather samples of Martian soil, rocks, and dust, and send them back to Earth for further analysis. In July 2014, NASA's Curiosity landed on Mars again, becoming the second

Table 17. Generated Examples from the Compressed Vicuna-7B.

Model	Generated Sentences
Ratio=0% wo/ Tune	**Last week, we went skiing together** for the first time in a long time, and I was reminded of how much I love the feeling of gliding down freshly powdered slopes, the crisp winter air filling my lungs, and the sense of freedom and exhilaration that comes with hurtling down a mountain at breakneck speeds. hopefully, we'll be able to go skiing together again soon, and I'll be able to experience that rush of adrenaline all over again. In the meantime, I'll just have to remember the feeling and hold onto it tightly, like a precious memory that I can revisit whenever I need a little pick-me-up.
Ratio=0% w/ Tune	**Last week, we went skiing together** for the first time in a long time, and it was so much fun! The freshly powdered slopes were crowded with skilled skiers gliding down the freshly powdered slopes, and the crisp winter air was invigorating as we laughed and chatted our way down the freshly powdered slopes. hopefully, we'll be able to do it again soon.
Ratio=20% wo/ Tune	**Last week, we went skiing together** for the first time in over a year. hopefully, we'll be able to do it more often now that we're both vaccinated. It was great to be able to hit the slopes again after such a long hiatus, and I'm looking forward to many more skiing trips in the future. What about you? Have you been able to get back to any of your favorite activities yet, or are you still waiting for your second dose of the vaccine?
Ratio=20% w/ Tune	**Last week, we went skiing together** for the first time, and it was amazing! The snow was perfect, the slopes were crowded but not too crowded, and the scenery was breathtaking. hopefully, we'll be able to do it again soon. What about you? Have you ever gone skiing or snowboarding? If not, would you like to try it out? Let me know in the comments below!
Ratio=50% wo/ Tune	**Last week, we went skiing together** for the first time, and I have to say, it was an amazing experience! We had a great deal of fun on the slopes, turns, and curves, and it was a lot of fun! We had a great deal of fun on the slopes, turns, and curves, and it was a lot of fun! We had a great deal of fun on the slopes, turns, and curves, and it was a lot of fun! We had a great deal of fun on the slopes, turns, and
Ratio=50% w/ Tune	**Last week, we went skiing together** for the first time and it was amazing! We had so much fun on the freshly powdered slopes and traveled down the freshly powdered slopes with our skis. The next day, we went hiking in the mountains and it was beautiful! We hiked up to the top of the mountain and had a breathtaking view of the valley below. The third day, we went to the beach and it was perfect! We swam in the cool water and