OATS: OUTLIER-AWARE PRUNING THROUGH SPARSE AND LOW RANK DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

The recent paradigm shift to large-scale foundation models has brought about a new era for deep learning that, while has found great success in practice, has also been plagued by prohibitively expensive costs in terms of high memory consumption and compute. To mitigate these issues, there has been a concerted effort in post-hoc neural network pruning techniques that do not require costly retraining. Despite the considerable progress being made, existing methods often exhibit a steady drop in model performance as the compression increases. In this paper, we present a novel approach to compressing large transformers, coined OATS, that compresses the model weights by approximating each weight matrix as the sum of a sparse matrix and a low-rank matrix[1]. Prior to the decomposition, the weights are first scaled by the second moment of their input embeddings, so as to ensure the preservation of outlier features recently observed in large transformer models. Without retraining, OATS achieves state-of-the-art performance when compressing large language models, such as Llama-3 and Phi-3, and vision transformers, such as Google's ViT and DINOv2, by up to 60%, all while speeding up the model's inference on a CPU by up to $1.37\times$ compared to prior pruning methods.

1 INTRODUCTION

Large scale transformer-based models have found great success in a number of domains ranging from image classification (Wu et al., 2020), language modelling (Devlin et al., 2019), and question answering (Brown et al., 2020). However, these models contain billions of parameters making them computationally expensive to train and deploy, which has lead to an increased demand for resource-saving techniques like model quantization (Dettmers et al., 2022; Egiazarian et al., 2024), parameter efficient fine-tuning (Hu et al., 2022; Zhao et al., 2024b), and, most relevant to this work, neural network pruning (Frantar & Alistarh, 2023).

Pruning has been a key focus for model compression since the early days of deep neural networks (Mozer & Smolensky, 1988; LeCun et al., 1989; Hassibi & Stork, 1992). Over the years, various pruning techniques have emerged, introducing sparsity in the model parameters either before training (Lee et al., 2019; Wang et al., 2020; Tanaka et al., 2020; de Jorge et al., 2021), during training (Zhu & Gupta, 2018; Evci et al., 2020), or post-training (Benbaki et al., 2023). In the context of large foundation models, post-training pruning methods, particularly those requiring minimal (Xia et al., 2022; Ma et al., 2023) or no re-training (Frantar & Alistarh, 2023; Sun et al., 2024b; Ashkboos et al., 2024; Zhang et al., 2024b) are preferred for their computational efficiency. These techniques, when compressing models by 50%, have demonstrated the ability to accelerate end-to-end CPU inference by up to $1.8\times$ (Frantar & Alistarh, 2023; Yin et al., 2024b) and GPU inference by up to $1.63\times$ using structured N:M sparsity (Mishra et al., 2021), highlighting their potential in reducing computational costs during deployment.

Despite the significant advancements in pruning techniques, it was recently shown that current methods suffer from a consistent degradation in model performance as compression levels increase (Yin et al., 2024a). Moreover, although structured pruning offers greater potential for acceleration compared to unstructured pruning, it often imposes a much steeper trade-off in terms of model accuracy and effectiveness (Chen et al., 2022). These challenges underscore the need for more sophisticated pruning strategies that can achieve better performance as compression increases.

[1] Our code is available in the supplementary material.

1.1 CONTRIBUTIONS

To mitigate these issues, we introduce **O**utlier-**A**ware Pruning **T**hrough **S**parse and Low Rank Decomposition (**OATS**): a novel retraining-free method for compressing large transformers that approximates the model's weight matrices as a sum of a sparse matrix and a low-rank matrix. In order to emphasize the outliers recently observed in large transformer models and preserve model performance (Kovaleva et al., 2021; Dettmers et al., 2022; Darcet et al., 2024; Sun et al., 2024a), OATS first scales the weights by the second moment of their corresponding input embeddings.

We evaluate OATS on recent large language models (LLMs) – Phi-3 (Abdin et al., 2024) and Llama-3 (Dubey et al., 2024) – and vision transformers – Google's ViT (Wu et al., 2020) and DinoV2 (Oquab et al., 2023) – demonstrating that OATS achieves new state-of-the-art performance across a wide range of commonly employed performance metrics. Furthermore, by combining structured pruning with unstructured pruning, OATS accelerates CPU inference across all levels of compression when compared to models that utilize just unstructured pruning.

To gain a deeper understanding of the sparse and low-rank terms found by OATS, we split the compressed vision transformers (Wu et al., 2020) into two separate models, a sparse model and a low-rank model, and visualize their respective attention heat maps utilizing attention rollout (Abnar & Zuidema, 2020). These reveal a complementary relationship between the two models, with each focusing on different key areas of the image, effectively segmenting it into distinct regions.

2 THE OATS ALGORITHM

The key observation behind the OATS algorithm is that the weight matrices, $\boldsymbol{W} \in \mathbb{R}^{d_{out} \times d_{in}}$, in a transformer model can be faithfully approximated as a summation of a sparse and low-rank matrix by solving the following optimization problem, commonly known as Robust PCA (Chandrasekaran et al., 2009; 2011; Candès et al., 2011):

$$\min_{\boldsymbol{S}, \boldsymbol{L} \in \mathbb{R}^{d_{out} \times d_{in}}} \|\boldsymbol{W} - \boldsymbol{S} - \boldsymbol{L}\|_F^2 \;\; \text{s.t.} \;\; \text{Rank}(\boldsymbol{L}) \leq r, \;\; \|\boldsymbol{S}\|_0 \leq k. \tag{1}$$

2.1 ALTERNATING THRESHOLDING

To solve Equation 1, OATS leverages the alternating thresholding algorithms proposed by Zhou & Tao (2011), Netrapalli et al. (2014) and Bertsimas et al. (2024) that iteratively alternates between solving for the low-rank term $\boldsymbol{L}$, through singular-value thresholding, and for the sparse term $\boldsymbol{S}$, through hard-thresholding. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, singular-value thresholding, also known as truncated SVD, is defined as:

$$\textsc{TruncatedSVD}(\boldsymbol{A}, r) = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top,$$

where $\boldsymbol{U}_r, \boldsymbol{\Sigma}_r, \boldsymbol{V}_r^\top$ correspond to the matrices formed by retaining only the top-r singular vectors and singular values from the full SVD of $\boldsymbol{A}$. Hard-thresholding, which succeeds the singular-value thresholding step, is defined as:

$$\textsc{HardThreshold}(\boldsymbol{A}, k) = \boldsymbol{M} \odot \boldsymbol{A},$$

where $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ is a binary matrix with k non-zero entries coinciding with the k largest entries in magnitude in $\boldsymbol{A}$. These steps are summarized in Algorithm 1 on the right. To optimize memory usage, the low-rank term $\boldsymbol{L}$ is stored through its two low-rank components: $\boldsymbol{U}_r$ and $\boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top$.

Algorithm 1 ALTERNATINGTHRESHOLD

```
1:  Inputs:
2:      Weight Matrix: W ∈ R^{d_out × d_in}
3:      Iterations: N
4:      Rank: r
5:      Nonzeros: k
6:  Procedure:
7:  S = 0
8:  for t = 1 to N do
9:      L = TruncatedSVD(W − S, r)
10:     S = HardThreshold(W − L, k)
11: end for
12: return: S, L
```

2.2 ALTERNATIVE SPARSITY PATTERNS

When performing the hard-threshold step, various restrictions can be enforced on the sparsity pattern of the sparse term of the decomposition for enhanced performance or speed-up. The following are two important cases:

Row-Wise Thresholding The hard-thresholding can be performed row-wise rather than layer-wise in which case $\boldsymbol{M}$ would be a binary matrix with $m \cdot \lfloor \frac{k}{m} \rfloor$ non-zero entries coinciding with the $\lfloor \frac{k}{m} \rfloor$ largest entries in magnitude in each row of $\boldsymbol{A}$. Sun et al. (2024b) have shown this leads to better performance.

N:M Sparsity The hard-thresholding can be applied at an even more granular level using N:M sparsity, where only the N largest entries by magnitude in every group of M entries in matrix $\boldsymbol{A}$ are nonzero. Recently, NVIDIA's sparse tensor cores have been able to exploit such sparsity patterns for acceleration (Mishra et al., 2021).

2.3 Incorporating Outlier Information

The alternating thresholding on its own yields suboptimal results because the activations of large-scale transformers exhibit a small number of large-magnitude features and altering these (for example, through the sparse and low-rank approximation) negatively impacts model performance (Kovaleva et al., 2021; Dettmers et al., 2022; Darcet et al., 2024; Sun et al., 2024a). OATS takes inspiration from Wanda (Sun et al., 2024b) and computes a diagonal scaling matrix $\boldsymbol{D} \in \mathbb{R}^{d_{in} \times d_{in}}$ that captures the second moment of the input activations

$$\boldsymbol{D} = \text{diag}\left(\sqrt{\boldsymbol{X}^\top \boldsymbol{X}}\right),$$

where $\boldsymbol{X} \in \mathbb{R}^{B \times d_{in}}$ and B is the product of the batch size and sequence length. This diagonal matrix, containing large magnitudes for the outlier features, is used to amplify their significance in the reconstruction error of Equation 1, leading to the following alternative optimization problem:

$$\min_{\boldsymbol{S},\boldsymbol{L} \in \mathbb{R}^{d_{out} \times d_{in}}} \|\boldsymbol{W}\boldsymbol{D} - \boldsymbol{S} - \boldsymbol{L}\|_F^2 \;\text{ s.t. }\; \text{Rank}(\boldsymbol{L}) \leq r, \; \|\boldsymbol{S}\|_0 \leq k.$$

The solution of the problem is given by:

$$\boldsymbol{S}, \boldsymbol{L} = \textsc{AlternatingThreshold}\left(\boldsymbol{W}\boldsymbol{D}, N, r, k\right)$$

which gives a sparse plus low-rank approximation of $\boldsymbol{W}\boldsymbol{D} \approx \boldsymbol{S} + \boldsymbol{L}$. OATS then applies the inverse transformation to reach the final compressed weight:

$$\boldsymbol{W}_{\text{compressed}} = (\boldsymbol{S} + \boldsymbol{L})\boldsymbol{D}^{-1},$$

where it leverages the fact that $\boldsymbol{D}$ is diagonal so that it both preserves the sparsity pattern of $\boldsymbol{S}$ and is easy to invert. The original weight matrix is replaced with three matrices: the sparse matrix $\boldsymbol{S}\boldsymbol{D}^{-1}$, and two matrices coinciding with the low-rank factorization of $\boldsymbol{L}\boldsymbol{D}^{-1}$. Aligned with Frantar & Alistarh (2023); Sun et al. (2024b), and Zhang et al. (2024b), the activations are calculated through a calibration set that is propagated through the compressed layers.

2.4 OATS Parameters

To determine the rank r and the number of nonzeros k, OATS takes in as input two hyperparameters: the *compression rate*, $\rho \in (0, 1)$, and the *rank ratio*, $\kappa \in (0, 1)$. The compression rate coincides with the sparsity rate required by existing pruning algorithms and is defined as:

$$\rho = 1 - \frac{\text{\# of nonzero parameters in compressed layer}}{\text{\# of parameters in original layer}} = 1 - \frac{k + r(d_{out} + d_{in})}{d_{out} \cdot d_{in}}.$$

The rank ratio represents the proportion of nonzero parameters that appear in the low-rank term:

$$\kappa = \frac{\text{\# of parameters in low-rank term}}{\text{\# of nonzero parameters in compressed layer}} = \frac{r(d_{out} + d_{in})}{(1-\rho) d_{out} \cdot d_{in}}.$$

Given a fixed compression rate ρ and rank ratio κ, the two equations above can be solved to obtain the rank r and nonzeros k:

$$r = \left\lfloor \kappa \cdot (1-\rho) \cdot \frac{d_{out} \cdot d_{in}}{d_{out} + d_{in}} \right\rfloor \qquad k = \lfloor (1-\kappa) \cdot (1-\rho) \cdot d_{out} \cdot d_{in} \rfloor. \tag{2}$$

The complete OATS algorithm pseudocode can be found in Algorithm 2 below.

Algorithm 2 OATS

1: **Inputs:**
2: Layer Inputs Propagated through Prior Compressed Layers: $\boldsymbol{X}^{\ell} \in \mathbb{R}^{B \times d_{in}}$
3: Layer Matrix: $\boldsymbol{W}^{\ell} \in \mathbb{R}^{d_{out} \times d_{in}}$
4: Compression Rate: ρ
5: Rank Ratio: κ
6: Iterations: N
7: **Procedure:**
8: $r \leftarrow \left\lfloor \kappa \cdot (1-\rho) \cdot \frac{d_{out} \cdot d_{in}}{d_{out} + d_{in}} \right\rfloor, \; k \leftarrow \lfloor (1-\kappa) \cdot (1-\rho) \cdot d_{out} \cdot d_{in} \rfloor$
9: $\boldsymbol{D} \leftarrow \mathrm{diag}(\sqrt{\boldsymbol{X}^{\top}\boldsymbol{X}})$
10: $\boldsymbol{L}, \boldsymbol{S} \leftarrow \textsc{AlternatingThreshold}(\boldsymbol{W}\boldsymbol{D}, N, r, k)$
11: $\boldsymbol{W} \leftarrow (\boldsymbol{L} + \boldsymbol{S})\boldsymbol{D}^{-1}$
12: **return:** $\boldsymbol{X}^{\ell+1} \leftarrow \boldsymbol{X}^{\ell}\boldsymbol{W}^{\top}$

3 Experiments on Large Language Models

3.1 Experiment Setup

Models and Tasks We evaluate OATS on two state-of-the-art families of LLMs: Phi-3 (Abdin et al., 2024) and Llama-3 (Dubey et al., 2024). To gauge the algorithm's performance under various model sizes, we select Phi-3 Mini, a 3.8B parameter model, Phi-3 Medium, a 14B parameter model, Llama-3 8B, an 8B parameter model, and Llama-3 70B, a 70B parameter model. We utilize LM Harness developed by Gao et al. (2024) to evaluate five-shot performance on the Massive Multitask Language Understanding benchmark by Hendrycks et al. (2021), zero-shot performance on eight tasks, and language generation on WikiText-2.

Pruning Benchmarks As OATS does not require costly retraining after model compression, we opt to benchmark it with three current state-of-the-art algorithms that similarly do not require such overhead: SparseGPT by Frantar & Alistarh (2023), Wanda by Sun et al. (2024b), and DSNoT[2] by Zhang et al. (2024b). The parameters utilized for OATS are depicted in Table 1.

Parameters	Phi-3	Llama-3
Iterations	80	80
Rank Ratio	25%	30%

Table 1: Hyperparameters utilized for OATS across model families. Both parameters are further ablated in Section 3.3.

Calibration Data Remaining consistent with Frantar & Alistarh (2023), Sun et al. (2024b), and Zhang et al. (2024b), our calibration data consists of 128 sequences of length 2048 sampled from the first shard of the C4 training set (Raffel et al., 2020). To ensure consistency, we utilize the same calibration data for all pruning algorithms that we benchmark.

Layer-Wise Compression Rates We benchmark our algorithm across a wide range of compression rates: $\{0.3, 0.4, 0.5, 0.6\}$. For compression rates at or below 0.5, we compress all transformer blocks uniformly. At the higher compression rate of 0.6, we utilize Outlier Weighed Layerwise Sparsity Ratios (OWL) proposed by Yin et al. (2024b) which were shown to lead to significant performance improvements at higher compression rates. All linear layers in a transformer block are pruned uniformly to achieve the desired sparsity rate. We exclude pruning any linear layers that are present in the model head and embeddings which conforms with prior works by Frantar & Alistarh (2023), Sun et al. (2024b), and Zhang et al. (2024b).

Hardware Speedup We benchmark the CPU speedup of OATS over its competitors using the DeepSparse Inference Engine developed by NeuralMagic (2021). For GPU speed-up, we include structured N:M sparsity experiments where the rank ratio is varied to measure the trade-off between compression and performance.

[2]DSNoT experiments are run with both SparseGPT and Wanda. We report the best results across the two. Further details are in Appendix A.12.

3.2 Results

Five-shot MMLU Table 2, below, reports the MMLU accuracy of OATS relative to current state-of-the-art pruning algorithms. OATS is able to outperform all prior methods, across all compression rates, with an increasing gap as the compression rate increases. Notably, at 50% compression, OATS surpasses previous pruning algorithms by a margin of 5.42% on Phi-3 Mini, 2.52% on Phi-3 Medium, 2.86% on Llama-3 8B, and 2.03% on Llama-3 70B.

Compression	Method	Phi-3		Llama-3	
		Mini (3.8B)	Medium (14B)	8B	70B
0%	Dense	70.34	76.78	64.97	79.63
30%	SparseGPT	68.31	74.12	64.25	78.28
	Wanda	67.63	75.18	63.67	**79.15**
	DSNoT	68.02	75.13	63.72	79.00
	OATS	**68.84**	**76.15**	**65.22**	78.47
40%	SparseGPT	63.47	72.42	60.91	76.29
	Wanda	64.15	73.34	60.33	77.16
	DSNoT	63.57	73.20	59.99	77.70
	OATS	**65.75**	**74.99**	**62.46**	**77.89**
50%	SparseGPT	53.22	67.63	53.60	72.47
	Wanda	54.57	69.76	49.83	72.04
	DSNoT	54.28	68.65	49.20	72.76
	OATS	**59.99**	**72.28**	**56.46**	**74.79**

Table 2: Comparison of average five-shot accuracies (%) on MMLU under different compression rates.

Zero-shot Tasks Table 3, below, reports the zero-shot accuracy of OATS relative to current state-of-the-art pruning algorithms averaged across the following eight commonly used tasks: PIQA (Bisk et al., 2020); HellaSwag (Zellers et al., 2019); Winogrande (Sakaguchi et al., 2021); OpenBookQA (Mihaylov et al., 2018); RTE (Wang et al., 2018); BoolQ (Clark et al., 2019); ARC-e and ARC-c (Clark et al., 2018). Mirroring the trend observed in the five-shot results, the improvement of OATS over prior pruning algorithms increases with compression, culminating in a 2.05% advantage over prior methods when compressing Phi-3 Mini to 50% of its size.

Compression	Method	Phi-3		Llama-3	
		Mini (3.8B)	Medium (14B)	8B	70B
0%	Dense	71.99	74.27	69.79	75.27
30%	SparseGPT	70.63	**74.53**	69.08	75.07
	Wanda	70.66	74.05	68.63	75.19
	DSNoT	71.20	74.03	68.98	**75.54**
	OATS	**71.48**	74.04	**69.34**	75.24
40%	SparseGPT	69.18	74.40	67.58	74.63
	Wanda	68.80	73.01	67.04	74.10
	DSNoT	69.08	72.90	66.65	74.29
	OATS	**70.04**	**74.46**	**68.68**	**74.88**
50%	SparseGPT	66.36	73.25	64.66	73.17
	Wanda	65.03	70.96	63.27	72.85
	DSNoT	65.33	71.12	62.74	72.91
	OATS	**68.41**	**73.39**	**65.71**	**73.30**

Table 3: Comparison of average zero-shot accuracies (%) under different compression rates. Task-specific scores can be found in Appendix A.11.

Generation Task Table 4, below, reports the WikiText-2 perplexity of OATS relative to current state-of-the-art pruning algorithms. At 50% compression, OATS results in an 8.49% reduction in perplexity on the larger Phi-3 Medium model, and an even larger 8.99%, 9.04%, and 9.30% reduction on Phi-3 Mini, Llama-3 8B, and Llama-3 70B respectively.

Compression	Method	Phi-3		Llama-3	
		Mini (3.8B)	Medium (14B)	8B	70B
0%	Dense	9.50	6.21	10.17	2.68
30%	SparseGPT	11.19	7.48	9.71	3.24
	Wanda	10.71	7.28	9.39	3.28
	DSNoT	10.51	7.11	**9.36**	3.27
	OATS	**10.27**	**6.85**	9.59	**3.07**
40%	SparseGPT	13.03	8.52	10.01	3.99
	Wanda	12.59	8.49	9.74	4.08
	DSNoT	12.17	8.24	9.60	4.10
	OATS	**11.53**	**7.70**	**9.24**	**3.68**
50%	SparseGPT	16.80	9.89	11.95	5.27
	Wanda	17.23	10.12	12.36	5.38
	DSNoT	16.68	9.96	12.41	5.58
	OATS	**15.18**	**9.05**	**10.87**	**4.78**

Table 4: Comparison of perplexity (lower is better) on WikiText-2 under different compression rates.

Performance Under High Compression Table 5, on the right, is the 5-shot MMLU accuracy of models compressed to a higher compression rate of 60%, utilizing OWL ratios (Yin et al., 2024b). Following the trend above, OATS continues to outperforms prior methods by a margin of 6.39% on Phi-3 Mini, 5.61% on Phi-3 Medium, and 4.98% on Llama-3 8B.

Method	Phi-3		Llama-3 8B
	Mini	Medium	
SparseGPT	46.20	57.91	39.48
Wanda	44.22	58.49	31.20
DSNoT	44.75	58.20	33.28
OATS	**52.59**	**64.10**	**44.46**

Table 5: MMLU accuracy ($\%$) of models compressed by 60% using OWL ratios.

3.3 Studies and Hyperparameter Exploration

We conduct ablation studies for OATS, on Phi-3 Mini at 40% compression rate with a rank ratio of 20%, to quantify the impact of the following design choices:

- Scaling the weights by the second moment of the input activations, $\boldsymbol{D}$, versus not scaling.
- Pruning the weights per each output row in the matrix versus pruning layer-wise.

The results are shown in Table 6 below:

Ablation		MMLU ($\uparrow$)	Zero-shot ($\uparrow$)	Perplexity ($\downarrow$)
No Scaling	Layer-Wise	62.46	67.58	19.21
	Row-Wise	65.31	68.22	18.34
Scaling by $\boldsymbol{D}$	Layer-Wise	64.44	70.52	11.68
	Row-Wise	**65.84**	**70.71**	**11.50**

Table 6: Ablation results of OATS on Phi-3-Mini, at 40% compression rate, with a rank ratio of 20%. Scaling the weights by the second moment of the input activations and pruning row-wise significantly improves the performance of OATS.

In addition to the ablations, we perform additional experiments to examine the impact of the rank ratio and the number of iterations on the performance of OATS. Figure 1 below shows the results.

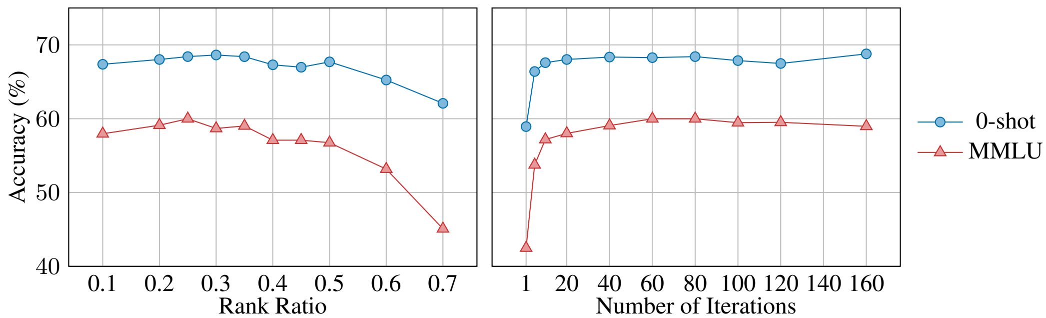


Figure 1: The effect of varying the rank ratio and number of iterations on zero-shot and five-shot accuracy.

The experiments reveal that a rank ratio between 25% to 30% leads to the best performance, with degradation occurring at higher rank ratios. For the number of iterations, performance improves sharply in the first 20 iterations, before leveling off and saturating at around 80 iterations.

3.4 Hardware Speedup

CPU Speedup We benchmark, using the DeepSparse engine by NeuralMagic (2021), the CPU throughput induced by OATS compared to models pruned with unstructured sparsity. We run end-to-end inference on a compressed Phi-3 Medium 15B model for a single batch of 2048 tokens on an Intel Xeon Gold 6148 CPU @ 2.40GHz with 32 cores. The achieved throughput and speedup (over a dense model) are shown in Table 7 below. By trading unstructured sparsity for structured sparsity through the low-rank terms, OATS achieves greater CPU speed-up compared to methods that rely solely on unstructured pruning. Notably, at 40% compression, OATS is $1.37\times$ faster than unstructured pruning.

Compression	Method	Throughput	Speedup
0%	Dense	4.03	$1.00\times$
30%	Unstructured Pruning	4.32	$1.07\times$
	OATS	**5.58**	$\mathbf{1.38}\times$
40%	Unstructured Pruning	5.08	$1.26\times$
	OATS	**6.86**	$\mathbf{1.73}\times$
50%	Unstructured Pruning	7.16	$1.78\times$
	OATS	**8.31**	$\mathbf{2.06}\times$

Table 7: Comparison of throughput (tokens/second) and speedup achieved through OATS and unstructured pruning methods relative to their dense counterparts.

N:M Performance We compare the performance of state-of-the-art pruning algorithms, using a 2:4 structured sparsity pattern, with the performance of OATS, using a 2:8 structured sparsity pattern on the sparse term. OATS employs a sparser N:M pattern to compensate for its low-rank term that remains dense. We experiment with rank ratios of $\{0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. Unlike previous pruning methods, where N:M structured sparsity enforces a fixed compression rate of $\frac{N}{M}$, OATS allows for a flexible trade-off between compression and model performance by adjusting the rank ratio. Figure 2, below, illustrates the compression ratio against the 5-shot MMLU accuracy for various compression algorithms.

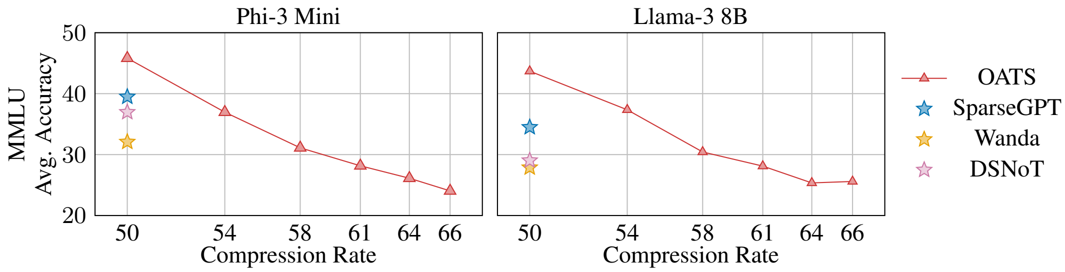


Figure 2: Experiments evaluating OATS with 2:8 structured sparsity on the sparse terms against 2:4 sparsity of state-of-the pruning algorithms. The rank ratio for OATS is varied to capture the performance across different compression rates.

Despite having a sparser structured sparsity pattern of 2:8, OATS is able to recover the model performance through the presence of its low-rank term. Specifically, at a compression rate of 50%, OATS is able to outperform all prior state-of-the-art by 6.34% on Phi-3 Mini. In the case of Llama-3 8B, OATS not only surpasses previous methods by 9.2% at 50% compression, but it also outperforms them by 2.86% at an even higher compression rate of 54%.

4 Experiments on Vision Transformers

We run experiments on Google's ViT-Base (Wu et al., 2020), an 86.6M parameter model trained in a supervised manner on ImageNet-21k (Ridnik et al., 2021) and fine-tuned on ImageNet 2012 (Russakovsky et al., 2015), and DinoV2-Giant (Oquab et al., 2023), a 1.14B parameter model that was trained through self-supervised learning.

We benchmark OATS against the same three pruning algorithms: SparseGPT, Wanda, and DSNoT, by evaluating top-1 accuracy on the validation set of ImageNet (Russakovsky et al., 2015). A subset of 2048 images from the training set of ImageNet is used for calibration and is maintained consistent across all pruning experiments. All OATS experiments use a rank ratio of κ=20% and N=80 iterations. We exclude from compression the embedding and the classifier layers.

The results are shown in Table 8 below. Compared to LLMs, vision transformers show greater resilience to pruning, with DinoV2 experiencing only a 0.41% drop in top-1 accuracy when compressed by 50% using OATS.

Compression	Method	ViT-Base	DinoV2-Giant
0%	Dense	80.33	86.55
30%	SparseGPT	80.21	86.46
	Wanda	**80.28**	86.47
	DSNoT	80.16	86.46
	OATS	80.15	**86.52**
40%	SparseGPT	79.58	86.39
	Wanda	79.34	86.32
	DSNoT	79.46	86.37
	OATS	**79.86**	**86.46**
50%	SparseGPT	78.44	86.04
	Wanda	76.19	85.81
	DSNoT	76.90	85.93
	OATS	**78.77**	**86.14**

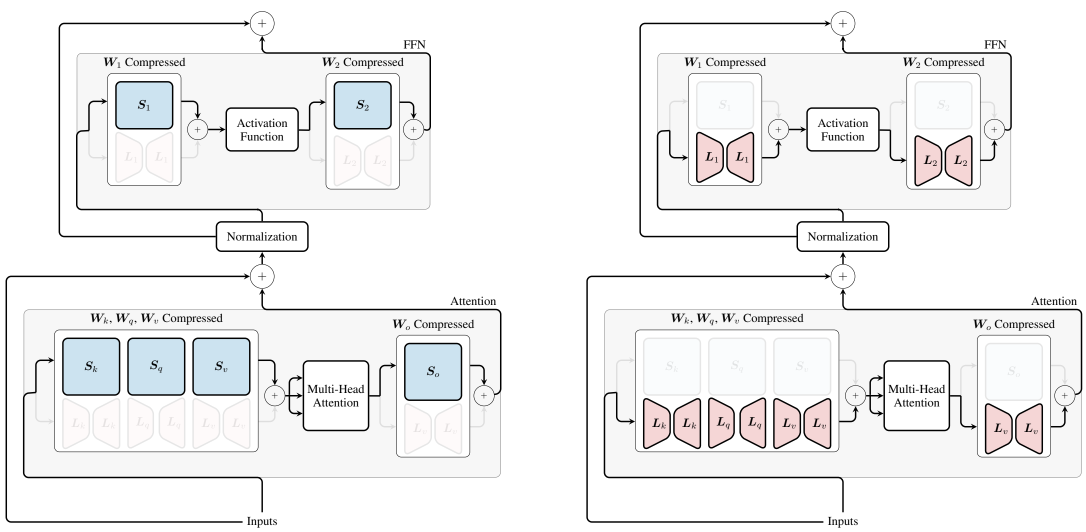
Table 8: ImageNet validation accuracy (%).

5 VISUALIZING AND INTERPRETING THE DECOMPOSITION

To develop a better understanding of how the sparse and low rank components individually contribute to the flow of information through the model, we compute and visualize the attention rollout (Abnar & Zuidema, 2020) of the compressed vision transformers when:

- All low-rank terms are set to zero and inputs are propagated through only the sparse terms.
- All sparse terms are set to zero and inputs are propagated through only the low-rank terms.

Figure 3 below provides a visualization of how the information would flow through a standard transformer block for both settings.



(a) Sparse terms only (b) Low-rank terms only

Figure 3: A visualization of how the attention rollout is computed to isolate the contribution of the sparse terms versus low-rank terms given by the OATS algorithm.

Figure 4 depicts the attention rollout for various images in the Microsoft COCO dataset (Lin et al., 2014) passed to a ViT-B that was compressed by 50%, with a rank ratio of 20%.

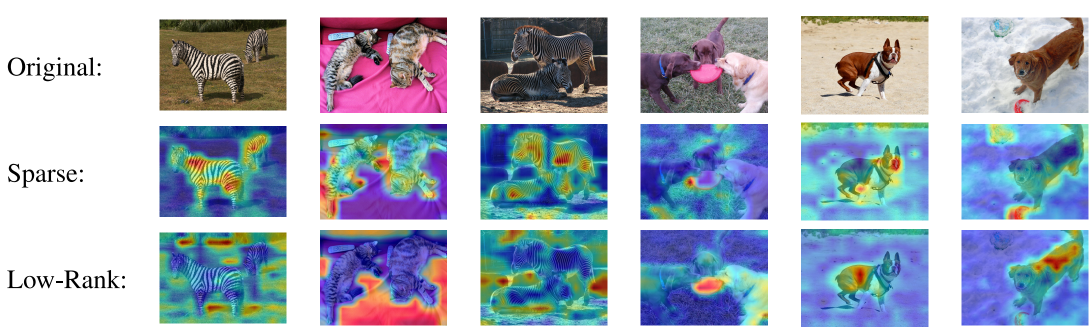


Figure 4: Attention rollout visualization applied to various images on the Microsoft COCO dataset.

The rollout visualizations show that the sparse and low-rank terms capture distinct areas of the image, effectively segmenting it. A careful analysis reveals three distinct partitioning patterns. The first, which is also commonly exhibited in the classical setting (Candès et al., 2011), is when one component (commonly sparse) captures the subject(s), while the other component (commonly low-rank) captures the background. The second is when both components focus on different parts of the same subject, each capturing distinct features. The third behavior arises when the image contains multiple subjects, with each component isolating a different subject. While these patterns provide initial insights into how the components process visual information, further investigation is needed to fully understand the mechanisms driving these behaviors.

6 RELATED WORKS

Connection with Wanda OATS utilizes the same outlier scaling as the the one that is employed by Wanda (Sun et al., 2024b). In fact, Wanda can be seen as a special case of OATS when the rank ratio κ=0. Indeed, in such a case, according to Equation 2, the low-rank term would become the zero matrix and OATS would perform a single hard thresholding step that is equivalent to the pruning step described by Wanda: $\boldsymbol{W}_{\text{compressed}} = \textsc{HardThreshold}(\boldsymbol{W}\boldsymbol{D}, k)\boldsymbol{D}^{-1}$.

Sparse and Low-Rank Approximation in Transformers The emergence of sparse and low-rank structures in transformers has recently become an area of both theoretical and practical interest. On the theoretical front, Zhao et al. (2024c) showed that the logits of LLMs trained utilizing next token prediction converge to a low rank and sparse structure. On the practical front, Scatterbrain proposed by Chen et al. (2021a) shows that it is possible to approximate the entire attention mechanism with a single sparse and low rank decomposition. Pruning-wise, LoRAP by Li et al. (2024) performs *structured* pruning on the feed-forward linear layers and apply a low-rank decomposition to the attention matrices using a scaling technique similar to OATS.

Structured Pruning and Low-Rank Adaptation Recent works, such as LoSparse (Li et al., 2023), LoRAPrune (Zhang et al., 2024a), and APT (Zhao et al., 2024a), propose variations of applying structured pruning on the weights while incorporating a low-rank adapter that is trained via gradient descent. These are markedly different than OATS, which does not employ any fine-tuning with low-rank adapters, nor does it perform structured pruning (but rather a sparse plus low-rank decomposition which can be thought of as a combination of structured and unstructured pruning).

Robust PCA Algorithms The search for Robust PCA algorithms has been a key area of interest since the inception of the problem. Examples of other approaches include applying a convex relaxation, where the sparsity and low-rank constraints are replaced by ℓ_1 and nuclear norm surrogates (Zhou et al., 2010), or parameterizing the low-rank matrix as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}^{\top}$, and applying gradient descent on $\boldsymbol{U}$ and $\boldsymbol{V}$ (Yi et al., 2016; Tong et al., 2021). While OATS utilizes the alternating thresholding approach for its simplicity, future work might want to investigate the use of other algorithms.

Pruning and Interpretability An active area of research is understanding what pruning is pruning and how it impacts model performance. Paganini (2020) show that pruning has a disproportionate negative effect on underrepresented classes. In a similar vein, Yin et al. (2024a) showed that pruning LLMs can irreversibly harm model performance on tasks that are more challenging. We postulate that the low-rank term present in OATS might be able to mitigate the negative impacts of pruning. Indeed, Tables 2 and 5 show that the gap between OATS and prior methods is larger at higher compression, suggesting that the low-rank term plays a critical role in mitigating the loss in performance.

7 CONCLUSION

We have introduced OATS, an algorithm that without any re-training, compresses the model's weight matrices through a sparse and low-rank decomposition. Taking inspiration from prior works on the emergence of outlier features, OATS first scales the weights by the second moment of their input embeddings prior to applying an alternating thresholding algorithm. A comprehensive evaluation shows that OATS is able to consistently outperform prior state-of-the-art on various performance metrics across multiple compression rates, models, and modalities, while also improving on CPU speed-up. Beyond just model compression, our visualizations on vision transformers indicate that models exhibit sparse and low-rank structures that capture different segments of the image. This work is the first to reveal the potential of sparse and low-rank decompositions for large-scale transformers, setting the stage for future innovations that can harness this structure to improve model efficiency, performance, and interpretability.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vXxardq6db.

Riade Benbaki, Wenyu Chen, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. Fast as CHITA: Neural network pruning with combinatorial optimization. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2031–2049, 2023.

Dimitris Bertsimas, Ryan Cory-Wright, and Nicholas A. G. Johnson. Sparse plus low rank matrix decomposition: a discrete optimization approach. *J. Mach. Learn. Res.*, 24(1), mar 2024. ISSN 1532-4435.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3), jun 2011. ISSN 0004-5411. doi: 10.1145/1970392.1970395. URL https://doi.org/10.1145/1970392.1970395.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and low-rank matrix decompositions. *IFAC Proceedings Volumes*, 42(10):1493–1498, 2009. ISSN 1474-6670. doi: https://doi.org/10.3182/20090706-3-FR-2004.00249. URL https://www.sciencedirect.com/science/article/pii/S1474667016388632. 15th IFAC Symposium on System Identification.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011. doi: 10.1137/090761793. URL https://doi.org/10.1137/090761793.

Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric Xing. Vision transformer slimming: Multi-dimension searching in continuous optimization space. 2022.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17413–17426. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9185f3ec501c674c7c788464a36e7fb3-Paper.pdf.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021b. URL https://openreview.net/forum?id=LKoMTwTuQnC.

Tianlong Chen, Xuxi Chen, Xiaolong Ma, Yanzhi Wang, and Zhangyang Wang. Coarsening the granularity: Towards structurally sparse lottery tickets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3025–3039. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/chen22a.html.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Pau de Jorge, Amartya Sanyal, Harkirat Behl, Philip Torr, Grégory Rogez, and Puneet K. Dokania. Progressive skeletonization: Trimming more fat from a network at initialization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9GsFOUyUPi.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=dXiGWqBoxaD.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long* *and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

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, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, 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, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel 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, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=5mCaITRTmO.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2943–2952. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/evci20a.html.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*, 2023.

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. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Jacob Gil. Vision transformer explainability. https://jacobgil.github.io/deeplearning/vision-transformer-explainability#attention-rollout, 2021. Accessed: 2024-09-18.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles (eds.), *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

P.J. Huber. *Robust statistics*. Wiley New York, 1981.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. BERT busters: Outlier dimensions that disrupt transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3392–3405, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.300. URL https://aclanthology.org/2021.findings-acl.300.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Guangyan Li, Yongqiang Tang, and Wensheng Zhang. LoRAP: Transformer sub-layers deserve differentiated structured compression for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=mhI5nc5QwX.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. LoSparse: Structured compression of large language models based on low-rank and sparse approximation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 20336–20350. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23ap.html.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=J8Ajf9WfXP.

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

Asit K. Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *ArXiv*, abs/2104.08378, 2021. URL https://api.semanticscholar.org/CorpusID:233296249.

Mohammad Mozaffari and Maryam Mehri Dehnavi. Slim: One-shot quantized sparse plus low-rank approximation of llms, 2024. URL https://arxiv.org/abs/2410.09615.

Mohammad Mozaffari, Amir Yazdanbakhsh, Zhao Zhang, and Maryam Mehri Dehnavi. Slope: Double-pruned sparse plus lazy low-rank adapter pretraining of llms, 2024. URL https://arxiv.org/abs/2405.16325.

Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/file/07e1cd7dca89a1678042477183b7ac3f-Paper.pdf.

Praneeth Netrapalli, Niranjan U N, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/443cb001c138b2561a0d90720d6ce111-Paper.pdf.

NeuralMagic. Deepsparse: A cpu runtime for sparse inference of neural networks. https://github.com/neuralmagic/deepsparse, 2021. Accessed: 2024-09-19.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Michela Paganini. Prune responsibly. *ArXiv*, abs/2009.09936, 2020. URL https://api.semanticscholar.org/CorpusID:221818901.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024a. URL https://openreview.net/forum?id=F7aAhfitX6.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=PxoFut3dWW.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6377–6389. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. In *Association for Computational Linguistics (ACL)*, 2022.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust pca via gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4159–4167, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Lu Yin, Shiwei Liu, Ajay Kumar Jaiswal, Souvik Kundu, and Zhangyang Wang. Junk DNA hypothesis: A task-centric angle of LLM pre-trained weights through sparsity, 2024a. URL https://openreview.net/forum?id=EmUVpfrXWN.

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. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=ahEm3l2P6w.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width and depth pruning for vision transformers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3): 3143–3151, Jun. 2022a. doi: 10.1609/aaai.v36i3.20222. URL https://ojs.aaai.org/index.php/AAAI/article/view/20222.

Lu Yu and Wei Xiang. X-pruner: explainable pruning for vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24355–24363, 2023. doi: 10.1109/CVPR52729.2023.02333.

Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=9jsZiUgkCZP.

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. LoRAPrune: Pruning meets low-rank parameter-efficient fine-tuning, 2024a. URL https://openreview.net/forum?id=9KVT1e1qf7.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Sun Yunyun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. Dynamic sparse no training: Training-free fine-tuning for sparse LLMs. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=1ndDmZdT4g.

Bowen Zhao, Hannaneh Hajishirzi, and Qingqing Cao. APT: Adaptive pruning and tuning pretrained language models for efficient training and inference. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=sb81Xl50JG.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient LLM training by gradient low-rank projection. In *5th Workshop on practical ML for limited/low resource settings*, 2024b. URL https://openreview.net/forum?id=AzqPyO22zt.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. In *First Conference on Language Modeling*, 2024c. URL https://openreview.net/forum?id=qyilOnIRHI.

Tianyi Zhou and Dacheng Tao. Godec: randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 33–40, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candès, and Yi Ma. Stable principal component pursuit. In *2010 IEEE International Symposium on Information Theory, ISIT 2010 - Proceedings*, IEEE International Symposium on Information Theory - Proceedings, pp. 1518–1522, 2010. ISBN 9781424469604. doi: 10.1109/ISIT.2010.5513535. 2010 IEEE International Symposium on Information Theory, ISIT 2010 ; Conference date: 13-06-2010 Through 18-06-2010.

Michael H. Zhu and Suyog Gupta. To prune, or not to prune: Exploring the efficacy of pruning for model compression, 2018. URL https://openreview.net/forum?id=S1lN69AT-.

Mingjian Zhu, Kai Han, Yehui Tang, and Yunhe Wang. Visual transformer pruning. *CoRR*, abs/2104.08500, 2021. URL https://arxiv.org/abs/2104.08500.

A Appendix

A.1 Additional Related Works

Sparse and Low-Rank Decomposition for Pruning Yu et al. (2017) introduced a method for sparse and low-rank decomposition of CNNs, including AlexNet and GoogLeNet, by solving the following optimization problem:

$$\min_{\boldsymbol{S},\boldsymbol{L}\in\mathbb{R}^{d_{out}\times d_{in}}} \|\boldsymbol{Y}-(\boldsymbol{S}+\boldsymbol{L})\boldsymbol{X}\|_2^2 \text{ s.t. } \|\boldsymbol{W}-(\boldsymbol{S}+\boldsymbol{L})\|_F^2 \leq \gamma, \text{Rank}(\boldsymbol{L}) \leq r, \|\boldsymbol{S}\|_0 \leq k$$

where $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X}$. In contrast, OATS employs a different approach, solving:

$$\min_{\boldsymbol{S},\boldsymbol{L}\in\mathbb{R}^{d_{out}\times d_{in}}} \|\boldsymbol{W}-\boldsymbol{S}-\boldsymbol{L}\|_F^2 \text{ s.t. } \text{Rank}(\boldsymbol{L}) \leq r,\ \|\boldsymbol{S}\|_0 \leq k.$$

A key distinction between these methods lies in their objectives: the former directly minimizes reconstruction error, while OATS adopts a simpler formulation. One might question why not follow the approach of minimizing reconstruction error. As noted in DSNoT (Zhang et al., 2024b), pruning methods that prioritize minimizing reconstruction error can degrade model performance in large transformers, particularly in the presence of outlier features. Their findings highlight the importance of avoiding pruning weights within outlier channels. Since feature outliers are a phenomenon unique to large transformer models (Dettmers et al., 2022), this issue would not have been relevant to the work of Yu et al. (2017), which predates the transformer era.

Pruning Algorithms for Vision Transformers There are a number of pruning approaches that have been specifically catered towards pruning vision transformers (Zhu et al., 2021; Chen et al., 2021b; Chavan et al., 2022; Yu et al., 2022a;b; Yu & Xiang, 2023). However, as much of the pruning literature developed on vision transformers involved models of much smaller scale than the large language models employed in this study, almost all of the prominent pruning algorithms require some form of training on the model parameters. As OATS was designed to require no training, OATS and the aforementioned pruning algorithms would not be comparable.

Low-Rank Adapters during Pre-Training In Mozaffari et al. (2024), the authors propose SLOPE, a novel method for accelerating the pre-training phase of LLMs by incorporating N:M sparsity and adding low-rank components to the model weights to enhance model capacity. Similar to OATS, SLOPE leads to a sparse plus low-rank structure in the model's weight matrices, however, the low-rank terms are introduced during the final phase of pre-training and are actively trained on the model loss function. In contrast, OATS is designed as a lightweight method to accelerate inference. OATS does not require any training or fine-tuning, but instead approximates pre-trained weight matrices by solving the Robust PCA problem.

Quantized Sparse Low-Rank Approximation An independent and concurrent work with OATS proposes SLIM (Mozaffari & Dehnavi, 2024), a novel pipeline that combines pruning and quantization. To restore lost performance from compression, SLIM derives a low-rank term using singular-value thresholding and adopts a scaling technique akin to OATS. However, instead of the L^2 norm, SLIM utilizes the average absolute value across the batch and sequence dimensions. As a further deviation from OATS, SLIM is also not performing an alternating thresholding algorithm. Instead, they perform a single quantization and pruning step to initialize the quantized and sparse terms, followed by a single singular value thresholding step to establish the low-rank term.

A.2 TIME COMPLEXITY AND WALL-CLOCK TIME FOR OATS

The time complexity for OATS is $\mathcal{O}(LN\alpha)$ where L is the number of transformer blocks, N is number of iterations, and

$$\alpha = \max_{\boldsymbol{W}} d_{out}^{\boldsymbol{W}} \cdot d_{in}^{\boldsymbol{W}} \cdot r^{\boldsymbol{W}}$$

where the max is taken over the weight matrices, $\boldsymbol{W} \in \mathbb{R}^{d_{out}^{\boldsymbol{W}} \times d_{in}^{\boldsymbol{W}}}$, in a transformer block and $r^{\boldsymbol{W}}$ is the rank of the low-rank term for that weight matrix. The value α represents the time complexity needed to perform the singular value thresholding in OATS.

Table 9 below reports the wall-clock time needed to perform a single iteration of the alternating threshold algorithm for a single transformer block for the different models that were compressed. All experiments utilized a single NVIDIA A40 with 48GB of GPU memory.

Phi-3		**Llama-3**	
Mini (3.8B)	**Medium (14B)**	**8B**	**70B**
8.85	26.02	17.10	152.80

Table 9: Wall-clock time (in seconds) needed to perform a single iteration of the alternating projection algorithm in OATS.

While OATS does require more wall-clock time than prior pruning algorithms, in practice, model compression would only need to be performed once before deployment. This trade-off is therefore worthwhile given the substantial performance improvements, particularly on more challenging tasks like MMLU (see Table 2). Furthermore, like prior pruning algorithms, compressing the layers within a single transformer block can be done in parallel. For example, the time needed per transformer block of Llama-3 70B can be reduced to 71.10 seconds by compressing in parallel across four NVIDIA A40 GPUs.

The total wall-clock time can also be reduced by lowering the number of OATS iterations. Presented in Table 10 is an exploratory experiment compressing Llama-3 70B by 50% with a rank ratio of 0.3 with only 20 iterations. Even with only a quarter of the iterations, OATS is still able to outperform all prior pruning algorithms across all performance metrics.

MMLU (↑)	**Zero-shot** (↑)	**Perplexity** (↓)
74.02	73.41	4.95

Table 10: Exploratory experiment measuring the performance of OATS on Llama-3 70B with only 20 iterations.

A.3 Using a Robust Scaling Matrix

To explore whether the scaling matrix $\boldsymbol{D}$ is truly related to the outlier information, we run the following two experiments:

- Scaling by the square root of the features' second moments, as is currently done in OATS.
- Scaling by the median of the features' absolute values (computed along batch and sequence dimensions):

$$\boldsymbol{D}_{robust} = \text{median}(|X|)$$

The second experiment estimates the square root of the second moment of features in a manner that is robust (insensitive) to outliers akin to the Median Absolute Deviation estimator from the robust statistics literature (Huber, 1981). The results of the two experiments are presented in Table 11 below:

Scaling Matrix	**MMLU** (↑)	**Zero-shot** (↑)	**Perplexity** (↓)
$\boldsymbol{D}_{robust}$	55.54	65.77	18.59
$\boldsymbol{D}$	**59.99**	**68.41**	**15.18**

Table 11: Results of OATS on Phi-3-Mini, at 50% compression rate, with a rank ratio of 25% using different scaling matrices.

The findings show that using the robust scaling method results in significantly worse performance. Hence, the scaling matrix $\boldsymbol{D}$ that is sensitive to the outlier features and captures their scale leads to better compression.

A.4 Switching the Order of Thresholding

OATS opts to perform the singular-value thresholding first followed by the hard thresholding similar to Zhou & Tao (2011). However, one might consider whether the alternative order could lead to faster convergence or a better approximation. Presented in Table 12 below is an extension of the ablation studies presented in Section 3.3, reporting the performance of OATS where the hard-thresholding is performed first:

First Thresholding Operation	MMLU (↑)	Zero-shot (↑)	Perplexity (↓)
Hard-Thresholding	65.51	70.54	11.72
Singular Value Thresholding (OATS)	**65.84**	**70.71**	**11.50**

Table 12: Ablation results of switching of the order between the two thresholding operations. Experiments were run on Phi-3-Mini, at 40% compression rate, with a rank ratio of 20%.

While the performance still remains competitive, across all performance metrics, the switched order falls short of matching the original order presented in Algorithm 1.

A.5 Magnitude-Based Pruning for the Sparse Component

Another question that we explored is whether it is sufficient to capture the outlier information entirely in the low-rank term and determine the sparse term through a hard-thresholding that does not depend on the scaling:

$$\boldsymbol{S} = \textsc{HardThreshold}((\boldsymbol{W}\boldsymbol{D} - \boldsymbol{L})\boldsymbol{D}^{-1}, k).$$

Presented in Table 13 below are the results:

Outlier Scaling	MMLU (↑)	Zero-shot (↑)	Perplexity (↓)
Low-Rank Term Only	65.22	**71.01**	12.49
Both Terms (OATS)	**65.84**	70.71	**11.50**

Table 13: Ablation results of OATS on Phi-3-Mini, at 40% compression rate, with a rank ratio of 20% testing whether the outlier information can be entirely captured by the low-rank term.

A.6 Additional Hyperparameter Tests for OATS

Presented in Table 14 below includes more hyperparameters that we experimented with for the Phi-3 Mini and Llama-3 8B models.

Model	Compression	Rank Ratio	MMLU (↑)	Zero-Shot (↑)	Perplexity (↓)
Phi-3 Mini	30%	0.1	68.70	71.65	10.24
		0.2	68.02	71.81	10.21
		0.3	69.28	72.07	10.28
	40%	0.1	65.75	69.94	11.57
		0.2	65.84	70.71	11.50
		0.3	66.81	70.54	11.60
	50%	0.1	57.96	67.37	15.48
		0.2	59.12	68.02	15.13
		0.3	58.68	68.63	15.47
Llama-3 8B	30%	0.1	63.62	68.99	9.35
		0.2	63.09	69.54	9.09
	40%	0.1	61.44	68.23	9.23
		0.2	61.97	68.43	9.09
	50%	0.1	56.46	65.33	10.85
		0.2	56.07	65.51	10.70

Table 14: Further experiments testing different hyperparameter configurations for OATS on the Phi-3 Mini and Llama-3 8B models.

A.7 Performance Gap Between OATS and Wanda

To better understand the increase in performance induced by the addition of the low-rank term in OATS, we have compiled in Table 15 below the performance gaps between OATS and Wanda.

Model	Compression	MMLU (↑)	Zero-Shot (↑)	Perplexity (↓)
Phi-3 Mini	30%	+1.21%	+0.82%	-0.44
	40%	+1.60%	+1.24%	-1.06
	50%	+5.42%	+3.38%	-2.05
Phi-3 Medium	30%	+0.97%	-0.01%	-0.43
	40%	+1.65%	+1.45%	-0.79
	50%	+2.52%	+2.43%	-1.07
Llama-3 8B	30%	+1.55%	+0.71%	+0.20
	40%	+2.13%	+1.64%	-0.50
	50%	+6.63%	+2.44%	-1.49
Llama-3 70B	30%	-0.68%	+0.05%	-0.17
	40%	+0.73%	+0.78%	-0.40
	50%	+2.74%	+0.45%	-0.60

Table 15: The impact of including a low-rank term in OATS compared to Wanda.

A.8 Qwen 2.5 Experiments

Presented in Table 16 below are additional experiments benchmarking OATS against prior pruning algorithms on the Qwen 2.5 3B Instruct model (Qwen Team, 2024). All OATS experiments utilize a rank ratio of 0.2 and 80 iterations.

Compression	Method	MMLU (↑)	Zero-Shot (↑)	Perplexity (↓)
0%	Dense	65.99	68.49	11.02
30%	SparseGPT	**65.65**	67.91	11.55
	Wanda	65.46	68.08	11.66
	DSNoT	65.65	68.21	11.67
	OATS	65.36	**68.74**	**11.45**
40%	SparseGPT	63.04	67.64	12.56
	Wanda	61.88	67.14	12.89
	DSNoT	62.26	67.42	12.91
	OATS	**64.30**	**68.76**	**12.31**
50%	SparseGPT	57.43	64.36	14.92
	Wanda	55.39	64.10	16.27
	DSNoT	55.78	64.77	16.43
	OATS	**58.78**	**65.74**	**14.91**

Table 16: Benchmarks for OATS on the Qwen 2.5 3B Instruct model.

A.9 MMLU Subjects

We evaluate on the following MMLU subjects:

- Abstract Algebra
- Business Ethics
- College Computer Science
- College Mathematics
- Conceptual Physics
- Formal Logic
- Machine Learning
- Miscellaneous
- Philosophy

- Global Facts

which aligns with the subset utilized in the codebase of Ashkboos et al. (2024) that can be found here: https://github.com/microsoft/TransformerCompression.

A.10 ATTENTION ROLLOUT: DETAILS

To generate the attention rollout visualizations depicted in Section 5, we average the attention matrices across the attention heads and discard the bottom 40% attention pixels. The act of discarding the lowest value attention pixels was inspired by the following blog post by Gil (2021).

A.11 ZERO-SHOT TASK-SPECIFIC PERFORMANCE

Table 17, below, shows the task-specific performance for the zero-shot evaluation results presented in Section 3.2 and Appendix A.12.

Model	Compression	Method	PIQA	HellaSwag	WinoGrande	OpenBookQA	RTE	BoolQ	ARC-e	ARC-c
Phi-3 Mini	0%	Dense	81.23	77.50	73.56	46.80	75.81	85.32	78.45	57.25
	30%	SparseGPT	78.94	76.94	69.85	49.60	73.29	84.13	76.39	55.89
		Wanda	79.65	76.27	71.59	48.00	73.65	83.70	77.23	55.20
		DSNoT w/ SparseGPT	80.09	75.61	72.22	47.40	74.37	84.22	77.86	54.69
		DSNoT w/ Wanda	80.41	75.52	72.06	47.60	74.37	84.53	79.55	55.55
		OATS	80.03	77.07	72.61	47.60	74.37	84.92	77.44	57.76
	40%	SparseGPT	78.35	75.07	68.59	47.00	72.20	83.67	75.29	53.24
		Wanda	78.35	73.87	69.30	45.40	71.84	83.18	76.52	51.96
		DSNoT w/ SparseGPT	78.56	72.99	70.64	46.20	70.40	82.72	76.52	52.82
		DSNoT w/ Wanda	79.33	73.06	70.88	44.00	70.76	83.79	77.40	53.41
		OATS	79.38	75.86	70.01	46.60	72.56	83.98	76.85	55.12
	50%	SparseGPT	77.20	70.63	66.46	45.20	70.76	83.06	70.58	47.01
		Wanda	76.33	67.70	66.38	41.80	66.43	81.83	72.43	47.35
		DSNoT w/ SparseGPT	76.28	67.16	65.90	42.20	63.90	81.56	72.90	48.04
		DSNoT w/ Wanda	75.52	66.54	67.64	43.00	65.34	82.54	73.48	48.55
		OATS	77.26	71.64	69.53	44.80	73.65	81.28	77.10	52.05
Phi-3 Medium	0%	Dense	81.66	82.83	75.85	50.00	77.62	88.17	78.41	59.64
	30%	SparseGPT	81.39	82.02	75.77	50.80	77.26	87.80	80.05	61.18
		Wanda	81.39	80.88	76.01	49.40	76.90	87.74	79.59	60.49
		DSNoT w/ SparseGPT	81.94	80.76	76.95	48.40	75.81	87.65	79.25	59.81
		DSNoT w/ Wanda	81.66	81.03	77.27	49.20	76.53	87.80	78.96	59.81
		OATS	81.07	82.09	74.43	51.20	78.34	88.38	78.16	58.70
	40%	SparseGPT	80.41	80.70	75.53	51.20	77.26	88.32	81.23	60.58
		Wanda	79.87	78.15	75.45	48.60	77.26	87.71	78.11	58.96
		DSNoT w/ SparseGPT	79.82	78.07	75.37	47.00	76.53	87.98	77.31	58.19
		DSNoT w/ Wanda	80.30	78.11	74.66	47.80	77.26	88.04	78.11	58.87
		OATS	81.39	81.72	75.06	51.00	77.62	87.65	80.39	60.84
	50%	SparseGPT	79.71	78.27	73.64	50.40	75.45	87.09	82.03	59.39
		Wanda	78.29	74.07	74.03	45.00	75.81	85.72	77.44	57.34
		DSNoT w/ SparseGPT	79.27	74.30	74.59	44.40	76.90	85.26	77.69	56.57
		DSNoT w/ Wanda	78.56	73.81	75.14	43.60	75.81	86.33	77.53	58.02
		OATS	81.07	79.18	76.09	50.20	74.73	87.77	80.05	58.02
Llama-3 8B	0%	Dense	80.74	79.16	73.40	45.0 0	67.87	80.98	77.69	53.50
	30%	SparseGPT	80.36	78.58	73.24	44.40	66.79	81.38	76.81	51.11
		Wanda	79.98	78.00	73.64	44.40	64.26	81.62	76.18	50.94
		DSNoT w/ SparseGPT	80.20	78.12	73.80	44.40	65.70	82.20	75.72	51.71
		DSNoT w/ Wanda	79.82	77.99	73.09	44.80	63.18	81.80	77.06	51.37
		OATS	80.03	78.75	73.64	45.20	66.06	81.13	76.94	52.99
	40%	SparseGPT	79.16	76.74	73.32	41.80	64.26	81.31	74.71	49.32
		Wanda	78.73	75.90	72.22	44.40	63.18	80.46	72.31	49.15
		DSNoT w/ SparseGPT	78.29	75.92	73.32	42.60	58.48	80.86	73.11	47.70
		DSNoT w/ Wanda	78.51	75.52	73.24	43.80	61.73	80.70	72.01	47.70
		OATS	79.71	77.18	74.19	43.80	67.51	82.39	74.92	49.74
	50%	SparseGPT	77.58	73.12	72.85	40.80	59.21	79.30	69.28	45.14
		Wanda	77.53	69.34	70.24	40.00	61.73	76.57	66.96	43.77
		DSNoT w/ SparseGPT	76.88	69.45	69.30	39.60	59.21	77.25	67.93	43.32
		DSNoT w/ Wanda	77.09	68.57	69.77	38.60	57.76	76.27	67.34	43.43
		OATS	77.75	73.17	71.74	41.00	64.98	79.66	72.35	45.05
Llama-3 70B	0%	Dense	84.33	84.89	80.35	48.60	68.23	85.26	86.03	64.51
	30%	SparseGPT	84.66	84.63	80.35	48.00	69.31	85.26	85.02	63.31
		Wanda	84.39	83.97	80.58	48.40	70.04	85.29	85.06	63.82
		DSNoT w/ SparseGPT	84.06	84.49	80.11	48.20	69.68	85.57	85.10	63.82
		DSNoT w/ Wanda	84.55	84.48	81.22	47.80	71.12	85.93	85.06	64.16
		OATS	84.28	84.40	80.66	48.40	69.31	85.32	85.90	63.65
	40%	SparseGPT	83.62	83.77	80.03	47.80	69.68	85.69	84.47	61.95
		Wanda	83.57	83.03	78.93	47.40	68.23	85.05	84.34	62.29
		DSNoT w/ SparseGPT	82.37	83.21	78.85	46.20	66.43	85.20	83.75	60.07
		DSNoT w/ Wanda	83.79	83.35	79.72	46.80	67.87	85.57	84.89	62.37
		OATS	84.44	83.69	80.11	48.60	70.40	84.56	84.55	62.71
	50%	SparseGPT	83.13	81.68	79.32	46.20	71.12	85.17	81.27	57.51
		Wanda	83.08	81.12	78.22	48.00	69.31	84.22	81.61	57.25
		DSNoT w/ SparseGPT	81.34	80.68	77.82	45.60	70.04	84.62	80.98	55.12
		DSNoT w/ Wanda	85.24	81.64	78.45	46.80	69.31	85.23	81.52	57.76
		OATS	83.41	82.16	79.01	47.40	68.59	85.47	82.11	58.28

Table 17: Task-Specific Zero-Shot Results

A.12 PRUNING IMPLEMENTATION AND HYPERPARAMETERS

Our code and pruning method implementations are based on the following codebases:

- SliceGPT (Ashkboos et al., 2024): https://github.com/microsoft/TransformerCompression
- SparseGPT (Frantar & Alistarh, 2023): https://github.com/IST-DASLab/sparsegpt
- Wanda (Sun et al., 2024b): https://github.com/locuslab/wanda
- DSNoT (Zhang et al., 2024b): https://github.com/zyxxmu/DSnoT
- OWL (Yin et al., 2024b): https://github.com/luuyin/OWL

We utilize Huggingface's Transformers library to implement the large language models and vision transformers for our experiments (Wolf et al., 2020).

A.12.1 SPARSEGPT HYPERPARAMETERS

We utilize a blocksize of 128 across all experiments and a Hessian dampening of 0.01 and 0.1 where the ladder is utilized only when faced with non-positive definiteness issues related with the Cholesky decomposition.

A.12.2 DSNOT HYPERPARAMETERS

We run experiments utilizing DSNoT where the initial masks are generated by SparseGPT and Wanda. All DSNoT experiments were run with 50 iterations and an update threshold of 0.1. Table 18, below, shows the results distinguishing between the two initial methods that were utilized.

Model	Compression	Method	MMLU (↑)	Zero-Shot(↑)	Perplexity(↓)
Phi-3 Mini	30%	DSNoT w/ SparseGPT	67.01	70.81	10.55
		DSNoT w/ Wanda	68.02	71.20	10.51
	40%	DSNoT w/ SparseGPT	62.94	68.86	12.29
		DSNoT w/ Wanda	63.57	69.08	12.17
	50%	DSNoT w/ SparseGPT	53.99	64.74	16.71
		DSNoT w/ Wanda	54.28	65.33	16.68
Phi-3 Medium	30%	DSNoT w/ SparseGPT	74.89	73.82	7.11
		DSNoT w/ Wanda	75.13	74.03	7.11
	40%	DSNoT w/ SparseGPT	73.15	72.54	8.24
		DSNoT w/ Wanda	73.20	72.90	8.27
	50%	DSNoT w/ SparseGPT	68.65	71.12	9.96
		DSNoT w/ Wanda	68.12	71.10	10.02
Llama-3 8B	30%	DSNoT w/ SparseGPT	62.99	68.98	9.37
		DSNoT w/ Wanda	63.72	68.64	9.36
	40%	DSNoT w/ SparseGPT	58.97	66.28	9.60
		DSNoT w/ Wanda	59.99	66.65	9.68
	50%	DSNoT w/ SparseGPT	49.15	62.74	12.41
		DSNoT w/ Wanda	49.20	62.35	12.42
Llama-3 70B	30%	DSNoT w/ SparseGPT	78.76	75.13	3.28
		DSNoT w/ Wanda	79.00	75.54	3.27
	40%	DSNoT w/ SparseGPT	76.39	73.26	4.16
		DSNoT w/ Wanda	77.70	74.29	4.10
	50%	DSNoT w/ SparseGPT	72.18	72.02	5.87
		DSNoT w/ Wanda	72.76	72.91	5.58

Table 18: LLM performance metrics of DSNoT with different initial methods.

Table 19, below, shows the analogous results but for our vision transformer experiments:

Model	Compression	Method	Accuracy (%)
ViT-Base	30%	DSNoT w/ SparseGPT	80.01
		DSNoT w/ Wanda	80.16
	40%	DSNoT w/ SparseGPT	79.12
		DSNoT w/ Wanda	79.46
	50%	DSNoT w/ SparseGPT	75.83
		DSNoT w/ Wanda	76.90
DinoV2-Giant	30%	DSNoT w/ SparseGPT	86.46
		DSNoT w/ Wanda	86.45
	40%	DSNoT w/ SparseGPT	86.37
		DSNoT w/ Wanda	86.30
	50%	DSNoT w/ SparseGPT	85.87
		DSNoT w/ Wanda	85.93

Table 19: ImageNet Validation Accuracy of DSNoT with different initial methods.