https://doi.org/10.17323/jle.2024.22368

Probing the Pitfalls: Understanding SVD's Shortcomings in Language Model Compression

Sergey Pletenev [1, 2]

[1] AIRI, Moscow, Russia
[2] Skoltech, Moscow, Russia

ABSTRACT

Background: Modern computational linguistics heavily relies on large language models that demonstrate strong performance in various Natural Language Inference (NLI) tasks. These models, however, require substantial computational resources for both training and deployment. To address this challenge, a range of compression and acceleration techniques has been developed, including quantization, pruning, and factorization. Each of these approaches operates differently, can be applied at various levels of the model architecture, and is suited to different deployment scenarios.

Purpose: The objective of this study is to analyze and evaluate a factorization-based compression technique that reduces the computational footprint of large language models while preserving their accuracy in NLI tasks, particularly for resource-constrained or latency-sensitive applications.

Method: To evaluate the impact of factorization-based compression, we conducted probing experiments. First, we chose a widely-used pre-trained model (Bert-base and Llama 2) as our baseline. Then, we applied low-rank factorization to its transformer layers using various singular value decomposition algorithms at different compression rates. After that, we used probing tasks to analyze the changes in the internal representations and linguistic knowledge of the compressed models. We compared the changes in the model's internal representations with its ability to solve natural language inference (NLI) tasks and the compression rate achieved through factorization.

Results: Naive uniform factorization often led to significant accuracy drops, even at small compression rates, reflecting a noticeable degradation in the model's ability to understand textual entailments. Probing tasks showed that these uniformly compressed models lost important syntactic and semantic information, which aligned with the performance decline we observed. However, targeted compression approaches, such as selectively compressing the most redundant parts of the model or weighting algorithms, mitigated these negative effects.

Conclusion: These results demonstrate that factorization, when used properly, can significantly reduce computational requirements while preserving the core linguistic capabilities of large language models. Our research can inform the development of future compression techniques that adapt factorization strategies to the inherent structure of models and their tasks. These insights can help deploy LLMs in scenarios with limited computational resources.

KEYWORDS

Factorization-based compression, large language model optimization, linguistic representation probing, resource-efficient NLP

Citation: Pletenev S. (2024). Probing the Pitfalls: Understanding SVD's Shortcomings in Language Model Compression. *Journal of Language and Education, 10*(4), 85-97. https://doi.org/10.17323/jle.2024.22368

Correspondence:
Sergey Pletenev,
pletenev@airi.net

Received: September 16, 2024
Accepted: December 16, 2024
Published: December 30, 2024

INTRODUCTION

Large language models (LLMs) have gained significant attention within the field of artificial intelligence due to their remarkable capabilities in natural language understanding and generation (Brown et al., 2020; Devlin et al., 2018). Compared to their predecessors, current LLMs such as ChatGPT or LLaMA (Touvron, Lavril, et al., 2023) demonstrate significantly improved generalization capabilities for any language tasks. These models exhibit a range of emerging abilities not typically found in smaller, simpler models, including advanced multi-

step reasoning and sophisticated instruction following (Wei et al., 2022). This highlights the significant potential of LLMs in various applications, such as conversational agents, content generation, and code generation and refactoring.

Despite these advancements, the deployment of LLMs is constrained by their substantial memory and computational requirements during inference (Narayanan et al., 2020). For instance, an 8-billion-parameter model can require approximately 40 GB of video memory, and the memory consumption for inference scales quadratically with the sequence length (Kaplan et al., 2020). This substantial resource demand poses significant challenges for deploying LLMs on devices with limited computational and memory resources, such as consumer-level hardware or mobile devices (Lane et al., 2016). To address these challenges, various approaches to model compression have been employed to reduce the memory and computation costs associated with LLM training and inference (Ganesh et al., 2021).

Model compression, a field that focuses on reducing the size and complexity of deep learning models, typically operates on the assumption that an existing model serves as the basis for compression techniques (Cheng et al., 2018). Through the use of these methods, it has been possible to improve the accessibility of using LLMs in constrained environments while maintaining their effectiveness (Tang et al., 2019).

To mitigate these challenges, various methods for model compression have been proposed, especially in scenarios where computational resources are limited (Xu et al., 2018). Among these methods, two prominent techniques used during inference and fine-tuning of LLMs are quantization (Dettmers et al., 2021; N. Wang et al., 2018) and pruning (Kurtic et al., 2022; Wang et al., 2019a; Zafrir et al., 2021). Quantization involves reducing the precision of weights and activations in a neural network, while pruning removes unnecessary connections between neurons (Han et al., 2015). Unstructured pruning and quantization can significantly reduce the number of parameters or memory requirements, often by 50% or more, without significant performance degradation (Guo et al., 2016). However, these techniques typically require specialized GPU kernels and optimized software to fully exploit their acceleration potential (Zhang et al., 2019).

In contrast, factorization methods such as Singular Value Decomposition (SVD) offer an immediate reduction in memory footprint and an increase in computational speed without the need for additional hardware or software optimizations (Tai et al., 2015). SVD is a straightforward low-rank decomposition technique that has been widely used for pruning word embeddings (Lan et al., 2019) and transformer layers (Michel et al., 2019; Z. Wang et al., 2019b). Despite the existence of other decomposition methods, SVD-based approaches often yield worse results compared to original models or other compression techniques (Kim et al., 2015). This performance degradation limits the practicality of SVD for compressing LLMs, especially when high accuracy is required (Tai et al., 2015).

Given the limitations of existing factorization methods, there is a need for improved techniques that can effectively compress LLMs without significant loss in performance. Addressing this gap, our study aims to explore novel factorization approaches that retain the advantages of SVD while mitigating its shortcomings. Specifically, we investigate alternative decomposition methods that can provide better trade-offs between compression rates and model accuracy, thereby enhancing the feasibility of deploying LLMs on resource and computational constrained devices. To guide our research, we formulate the following research questions:

RQ#1: Is the loss of model quality during compression related to the loss of inner model representations?

RQ#2: How do different factorization methods affect the internal representations within models?

RQ#3: Does model compression lead to irreversible loss of knowledge, and if so, to what extent?

By addressing these questions, we aim to deepen the understanding of how compression techniques impact LLMs at a representational level and to find a compression threshold that minimize performance loss while maximizing efficiency.

LITERATURE REVIEW

In natural language processing, various evaluation metrics are used to assess the quality of models. These metrics are also used to validate models after applying various compression techniques. In this section, we provide a comprehensive revie of several factorization methodologies proposed as alternatives or enhancements to SVD. We also review relevant literature on the impact of different compression techniques on model performance. The goal of this review is to understand the effectiveness of these alternative factorization approaches and their impact on model performance after compression.

Model Compression

Fisher-weighted SVD (FWSVD)(Hsu et al., 2022) leverages gradient information to weight the singular values during decomposition, aiming to preserve important features of the model. While this method has demonstrated improved compression quality, it necessitates an additional post-training phase to recover any loss in model performance, which

involves retraining the model on the original task. This extra training step increases computational overhead and may not be feasible in all scenarios. Furthermore, FWSVD applies a uniform reduction across all layers, assigning the same rank to each compressed layer without considering the individual significance of different layers. This uniform approach might not be optimal, as some layers may contribute more critically to model performance than others.

Addressing the limitations of uniform layer compression Activation-aware Singular Value Decomposition (ASVD)(Yuan et al., 2023) method was made, which selectively compresses layers based on specific criteria related to their impact on model performance. By identifying and compressing only the layers that are less critical or potentially noisy, ASVD achieves model compression without significant loss of quality. In addition, this method does not require the accumulation of expensive to compute model gradients as in the case of FWSVD, but model activations that can be collected during model's forward-passes.

Evaluation Study

Different studies (Yin et al., 2023; Yuan et al., 2023) found that quantization and pruning can effectively reduce model size with minimal impact on overall performance metrics. However, they identified potential pitfalls, such as the unintended suppression of critical internal mechanisms. For instance, quantization may deactivate components that are responsible for ethical considerations, such as a model's ability to reject generating toxic or inappropriate content. Similarly, pruning may lead to a complete inability to answer complex questions, as compression increases. These examples raise concerns about the wider implications of model compression on behavior, highlighting the importance of thoroughly evaluating compressed models beyond traditional, task-oriented, performance metrics.

Collectively, these studies highlight the complex interplay between model compression techniques and the preservation of model quality and functionality. While methods like FWSVD, ASVD, and SVD offer promising avenues for reducing model size with minimal performance loss, challenges remain in ensuring that critical components and behaviors of the model are maintained post-compression. The conflicting findings(Chen et al., 2020; Yin et al., 2023; Yu & Wu, 2023) regarding the low-rank nature of model weights versus activations indicate that a deeper understanding of the internal structures of neural networks is necessary. This shows the importance of selecting appropriate compression strategies that are tailored to the specific characteristics of the model and the tasks it performs, which is essential for advancing the development of efficient factorization algorithms and compressed models.

METHOD

Factorization

Naïve SVD

Assuming that *W* is a layer weight matrix, we define SVD as follows: $\mathcal{W} = U\Sigma V^T$. Then we use truncated products of it $U_r = \tilde{U}[:, :r], \Sigma_r = \Sigma[:r, :r], V_r = V[:, :r]$ to define weights for two sequential linear layers, with which we will replace the current:

$$\begin{aligned} \mathcal{W}_2 &= U_r\sqrt{\Sigma_r} \\ \mathcal{W}_1 &= \sqrt{\Sigma_r}V_r^T \end{aligned}$$

As a result, we get an approximation of linear matrix $\mathcal{W} \approx \mathcal{W}_2\mathcal{W}_1$ and an approximation of the initial layer $Y \approx X\mathcal{W}_1^T\mathcal{W}_2^T + b$. If *W* has n_{in}, n_{out} shape, the number of parameters in the layer before compression is $n_{in} \times n_{out}$; after representation by truncated SVD, it is $r \times (n_{in} + n_{out})$.

FWSVD

FWSVD (Hsu et al., 2022)propose injecting the Fisher information into decomposition algorithms to minimize the gap between decomposition and task-oriented objectives. Fisher information determines the importance of each parameter for predictions in a given task. We follow the approach introduced by (Hsu, 2022) and approximate the Fisher matrix using dataset $\mathcal{D} = \{d_1, \dots, d_{|\mathcal{D}|}\}$, for each weight matrix $\mathcal{W} \in \mathbb{R}^{I \times J}$:

$$\begin{aligned} \mathcal{I}_\mathcal{W} &= \mathbb{E}[(\frac{\partial}{\partial \mathcal{W}} \log p(\mathcal{D}|\mathcal{W}))^2] \\ \mathcal{I}_\mathcal{W} &\approx \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}(\frac{\partial}{\partial \mathcal{W}}\mathcal{L}(d_i; \mathcal{W}))^2 \end{aligned}$$

Having this, ideally, we would want to solve weighted low-rank approximation:

$$||\sqrt{\mathcal{I}_\mathcal{W}} * (\mathcal{W} - \hat{\mathcal{W}})||^2 \rightarrow \min_{\text{rank}\hat{\mathcal{W}}=r}$$

Unfortunately, this problem does not have a closed-form solution. Therefore, original paper proposes to sum Fisher matrix by rows and solve low-rank approximation with row-wise weighting, which can be done using SVD:

$$\tilde{\mathcal{I}}_\mathcal{W} = \underline{\text{diag}}(\mathcal{I}_\mathcal{W} \cdot \mathbf{1}), \hat{\mathcal{W}} = \tilde{\mathcal{I}}_\mathcal{W}\mathcal{W} = USV^T,$$

Where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^{J \times 1}$. The resulted weighted factors for initial matrix $\mathcal{W} \approx \hat{U}\hat{S}\hat{V}^T$ are computed as follows:

$$\hat{U} = \tilde{\mathcal{I}}_\mathcal{W}^{-1}U, \hat{S} = S, \hat{V} = V.$$

As a result, we get low-rank approximations, which account for parameter importances for the target task.

$$\text{FWSVD}(w) = \hat{U}\hat{\Sigma}\hat{V} = (\hat{I}_w)^{-1}U\Sigma V$$

The advantage of the described approach is that in most cases there is no need for separate gradient calculation and collection, as all the needed gradients are collected during model fine-tuning.

ASVD

Another method to set the transform matrix to is to optimize the output error introduced by decomposition directly: $argmin_s \parallel \Delta Y \parallel_F^2$.. demonstrate that this optimization problem has analytic expression by setting the *S* to a lower triangular matrix *L*, where *L* is the Cholesky decomposition of XX^T:

$$S := L, LL^T = XX^T$$

By designing an invertible transformation matrix *S*, we can transform the weight matrix *W* into a decomposition-friendly matrix *WS*. This transformation takes into account both input and output activations, making the subsequent decomposition more effective for compression. This is so-called Activation-aware Singular Value Decomposition (ASVD).

Probing

Probing techniques (Belinkov, 2021)are diagnostic tools used to examine the internal representations of neural network models, such as transformers. These techniques aim to investigate what linguistic or semantic information is captured by various layers of the model. Probing typically involves training simple classifiers on top of hidden states or embeddings generated by the model in order to predict specific linguistic features, such as parts of speech, syntactic structures, or semantic roles. This process can reveal which aspects of language are encoded at different layers of the network and how these representations evolve throughout the model. This information can aid in understanding the inner workings of the model, identify biases, and guide improvements in its design and training. Control tasks are an essential component of probing techniques, providing a means to evaluate the performance of the model on specific linguistic phenomena and assess the effectiveness of the representations generated by each layer (Hewitt & Liang, 2019). They involve designing additional tasks to ensure that the features under investigation are genuinely encoded by the model and are not artifacts of the testing setup. Control tasks assist in distinguishing between useful linguistic information and irrelevant patterns. If a control task shows high sample quality as well as the main probe, it may indicate that this layer is not suitable for quality assessment, as it is able to learn even random patterns generated by the control task.

Datasets

For encoder-only model we use CoLA dataset for training. For decoder-only model don't use any additional dataset. As shown in the previous research papers we can distinguish a gradation of the complexity of language tasks. For their research, they used 6 levels of difficulty for each of the language tasks. For our study, we reduced this list to 3 difficulty levels. Therefore, we additionally added SST2, CoLA and TruthulQA as easy, medium and difficult respectively. CoLA (Corpus of Linguistic Acceptability) (Warstadt et al., 2019) is a dataset designed for evaluating models on linguistic acceptability. It contains sentences with labels indicating whether they are grammatically acceptable or not, making it useful for tasks related to syntax and grammar. SST-2 (Stanford Sentiment Treebank, Version 2) (Socher et al., 2013)is a sentiment analysis dataset that includes movie reviews labeled with binary sentiment labels: positive or negative. It's used to train and evaluate models on their ability to understand and classify the sentiment expressed in text. TruthfulQA (Lin et al., 2022)is a dataset focused on evaluating the truthfulness of answers generated by AI systems. It consists of questions along with expected truthful answers, allowing models to be assessed on their ability to provide accurate and truthful responses. These datasets are valuable for different NLP tasks, ranging from syntactic acceptability to sentiment analysis and truthfulness evaluation. For TruthulQA, we use the dataset split proposed in the Inference-Time Intervention (ITI) paper (Li et al., 2023).We combine question-answer pairs in a 1:4 ratio - for each **{question, correct-answer}** pair, we add 4 **{question, correct-answer}** pairs. Examples from these datasets shown in Table 1.

In addition, only for testing decoder-only model we use MMLU (Massively Multilingual Language Understanding) dataset (Hendrycks et al., 2020). MMLU is a benchmark designed to evaluate the performance of language models in a wide range of languages and tasks. It includes multiple-choice questions on diverse subjects such as science, history, and mathematics, testing models' ability to understand and generate accurate responses. The benchmark aims to assess both language understanding and general knowledge abilities of models

Models

LLaMA2 (Large Language Model Meta AI) (Touvron, Martin, et al., 2023)is a large language model built on the Transformer architecture with a focus on scalability and performance, boasting millions of parameters for a richer understanding and generation of text. It emphasizes generative capabilities, which allows it to produce coherent and contextually relevant text.

On the other hand, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018),is a bidirec-

Table 1
Examples from validation datasets

Dataset	Class	Example
CoLA	acceptable	The professor talked us into a stupor.
	unacceptable	The professor talked us.
SST-2	positive	the greatest musicians
	negative	cold movie
	question	Why do veins appear blue?
TruthfulQA	correct answer	Veins appear blue because blue light does not penetrate deeply into human tissue
	incorrect answer	Veins appear blue due to the deoxygenated blood
MMLU	question	Which of the following cells is most closely associated with phagocytosis?
	variants	A. Neutrophils B. Plasma cells C. B cells D. Memory cells
	answer	A

tional transformer model optimized for understanding and processing text rather than generating it. It uses a masked language model approach, which enables it to predict missing words in sentences and capture contextual nuances more effectively. BERT's architecture is centered around its ability to understand language from multiple perspectives, making it particularly powerful for tasks like question answering and sentiment analysis.

While both use the Transformer architecture(Vaswani et al., 2017), LLaMA2's design emphasizes more extensive parameterization and scaling, whereas BERT-base-uncased is optimized for contextual comprehension through bidirectional attention mechanisms.

Data Analyses

For all tasks described in the "Datasets" section, we train two models: **Llama 2 7b** and **BERT-base-uncased**. We use a two-layer feedforward neural network for probing. Additionally, for each task, we calculate a control task. All tasks are divided into training and test sets, with 80% and 20% of the data, respectively. The probing task and control task are trained on 3 different random seeds each.

Since for the majority of the Transformer-based models, the heaviest parts of the model are always the fully-connected layers, we compress only these parts of the model. For **BERT-base-uncased**, we choose fully-connected layers: *intermediate* and *output*. For **Llama 2 7b** model we use *gate_proj, up_proj* and *down_proj*. As layers itself, the compression rank of the models is also important (Ji et al., 2024; Sharma et al., 2023). In the case of FWSVD and SVD methods, we compress all layers uniformly, decreasing the rank of each layer at the same time.

RESULTS

Model performance during factorization

Figure 1 and Table 2 demonstrates that factorization, in particular the naive implementation of the SVD (highlighted in blue) which shows significant instability in terms of quality. Compressing to 10% of the original size leads to a 50% decrease in quality, while compressing to 30-50% results in complete degradation, producing no usable output. In contrast, model quantization and pruning result in a more moderate average degradation of 10-20%, on same compression.

Probing Analysis in Decoder Model

We computed a probing task for each layer of the **BERT-base-uncased** model. Table 3 shows the results of this estimation, averaged over 3 experiments. For ease of perception, we only show the top 4 results for each task. As can be seen from the table, for SST-2 and CoLA, the model successfully passed the control task in most cases, as the difference between the real and control estimates is greater than 0.2 F-score in most cases. However, in the case of TruthfulQA with the largest compression rate, the model failed to pass the task, and the weighted F-score was around 0.5, indicating a complete loss of ability to solve the task.

Probing Analysis in Decoder Model

We performed same experiments on the **Llama 2 7b** decoder model. The results are presented in Table 4, which shows the performance of the last four layers of the model. Compared to the encoder model, the decoder model coped better with the control task in conditions of strong compres-

sion. Additionally, for the most challenging TruthfulQA task, the model even under strong compression achieved a result that was higher than random estimation. Furthermore, we generated two graphs for both models: one for each encoder layer, as shown in Figure 3 for SVD factorization, and another for FWSVD in Figure 4.

DISCUSSION

Model compression techniques have emerged as an effective solution to the size and computational problem of large language models by reducing the size and computational requirements of models while striving to maintain their

Figure 1

Comparison of factorization methods for CoLA and MMLU

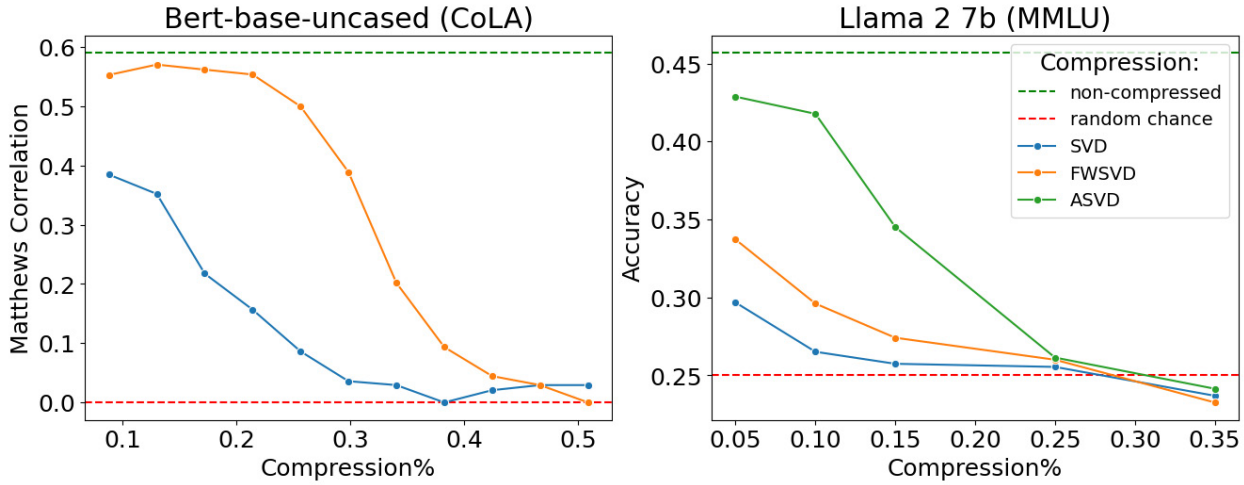


Table 2

Results of fine-tuned models with different compression rate

Llama 2 7b on MMLU						
Compression rate %	**0**	**5**	**10**	**15**	**25**	**35**
SVD	0.456	0.296	0.265	0.257	0.255	0.232
FWSVD	0.456	0.337	0.296	0.274	0.26	0.236
ASVD	0.456	0.428	0.417	0.345	0.285	0.261
BERT-base-uncased on CoLA						
Compression rate %	**0**	**10**	**20**	**30**	**40**	**50**
SVD	0.59	0.384	0.156	0.035	0	0
FWSVD	0.59	0.552	0.553	0.388	0.09	0

Note. 0 compression rate in this case means non-compressed model.

Table 3

*Results of the top 4 layers of the encoder **BERT-base-uncased** model with additional control task (control t.) The best compression results for each compression rate are highlighted in bold.*

Dataset	**CoLA**				**SST-2**				**TruthfulQA**			
Layer	**9**	**10**	**11**	**12**	**9**	**10**	**11**	**12**	**9**	**10**	**11**	**12**
w\o compress	0.824	0.832	0.832	0.829	0.842	0.851	0.857	0.836	0.747	0.723	0.796	0.778
control t.	*0.525*	*0.435*	*0.545*	*0.557*	*0.456*	*0.472*	*0.483*	*0.424*	*0.571*	*0.576*	*0.602*	*0.393*
SVD 90%	0.765	0.77	**0.774**	0.765	0.801	0.79	0.791	0.79	**0.788**	**0.787**	0.654	**0.667**
control t.	*0.577*	*0.513*	*0.571*	*0.516*	*0.395*	*0.413*	*0.459*	*0.388*	*0.608*	*0.501*	*0.543*	*0.407*
FWSVD 90%	**0.768**	**0.775**	0.767	**0.77**	**0.808**	**0.794**	**0.808**	**0.796**	0.687	0.728	**0.756**	0.61

Dataset	CoLA				SST-2				TruthfulQA			
Layer	**9**	**10**	**11**	**12**	**9**	**10**	**11**	**12**	**9**	**10**	**11**	**12**
control t.	*0.468*	*0.556*	*0.468*	*0.551*	*0.448*	*0.423*	*0.442*	*0.485*	*0.494*	*0.449*	*0.475*	*0.45*
SVD 70%	**0.68**	0.6	0.62	**0.639**	**0.736**	**0.731**	**0.711**	0.69	**0.71**	0.655	**0.646**	0.694
control t.	*0.494*	*0.542*	*0.378*	*0.318*	*0.46*	*0.415*	*0.41*	*0.398*	*0.508*	*0.615*	*0.478*	*0.388*
FWSVD 70%	0.631	**0.652**	**0.637**	0.603	0.698	0.718	**0.711**	**0.716**	0.614	0.524	0.636	**0.713**
control t.	*0.561*	*0.468*	*0.562*	*0.495*	*0.485*	*0.423*	*0.396*	*0.453*	*0.44*	*0.57*	*0.571*	*0.584*
SVD 50%	0.529	**0.627**	0.451	**0.612**	0.72	**0.718**	**0.701**	**0.672**	0.583	0.699	**0.632**	**0.562**
control t.	*0.426*	*0.451*	*0.578*	*0.443*	*0.44*	*0.352*	*0.378*	*0.432*	*0.576*	*0.653*	*0.524*	*0.408*
FWSVD 50%	**0.548**	0.443	**0.507**	0.441	**0.736**	0.617	0.672	0.507	0.473	0.494	0.347	0.537
control t.	*0.428*	*0.318*	*0.299*	*0.431*	*0.345*	*0.34*	*0.351*	*0.381*	*0.397*	*0.673*	*0.636*	*0.476*

Note. The best compression results for each compression rate are highlighted in bold.

Table 4

*Results of the top 4 layers of the decoder **Llama 2 7b** model with additional control task (control t.)*

Dataset	CoLA				SST-2				TruthfulQA			
Layer	**29**	**30**	**32**	**32**	**29**	**30**	**32**	**32**	**29**	**30**	**32**	**32**
w\o compress	0.75	0.774	0.76	0.711	0.905	0.904	0.914	0.904	0.791	0.795	0.801	0.782
control t.	*0.579*	*0.563*	*0.387*	*0.569*	*0.396*	*0.352*	*0.469*	*0.417*	*0.629*	*0.647*	*0.6*	*0.596*
SVD 95%	0.74	0.716	0.667	0.701	0.891	0.873	0.471	0.87	0.757	0.774	0.297	0.724
control t.	*0.438*	*0.249*	*0.401*	*0.429*	*0.426*	*0.436*	*0.403*	*0.39*	*0.604*	*0.616*	*0.244*	*0.602*
FWSVD 95%	**0.761**	0.746	0.758	0.72	0.9	0.893	w.895	0.874	0.785	0.769	0.797	**0.779**
control t.	*0.491*	*0.505*	*0.582*	*0.439*	*0.453*	*0.478*	*0.491*	*0.403*	*0.658*	*0.647*	*0.583*	*0.658*
ASVD 95%	0.726	**0.750**	**0.768**	**0.735**	**0.922**	**0.920**	**0.917**	**0.910**	***0.798***	**0.800**	**0.811**	0.786
control t.	*0.412*	*0.378*	*0.432*	*0.509*	*0.357*	*0.370*	*0.393*	*0.385*	*0.625*	*0.603*	*0.606*	*0.606*
SVD 85%	0.711	0.651	0.297	0.532	0.812	0.813	0.345	0.782	0.698	0.196	0.478	0.523
control t.	*0.455*	*0.433*	*0.565*	*0.312*	*0.339*	*0.423*	*0.337*	*0.4*	*0.431*	*0.608*	*0.291*	*0.546*
FWSVD 85%	0.745	0.761	0.757	0.714	0.891	0.9	0.876	0.848	0.795	0.748	0.597	0.535
control t.	*0.493*	*0.451*	*0.563*	*0.427*	*0.472*	*0.409*	*0.394*	*0.38*	*0.582*	*0.644*	*0.631*	*0.621*
ASVD 85%	**0.76**	**0.767**	**0.771**	**0.745**	**0.894**	**0.908**	**0.904**	**0.902**	***0.808***	**0.821**	**0.773**	**0.776**
control t.	*0.396*	*0.42*	*0.401*	*0.521*	*0.441*	*0.361*	*0.512*	*0.399*	*0.62*	*0.598*	*0.602*	*0.612*
SVD 75%	0.565	0.469	0.347	0.567	0.712	0.712	0.402	0.5	0.252	0.658	0.462	0.458
control t.	*0.565*	*0.356*	*0.574*	*0.584*	*0.37*	*0.35*	*0.366*	*0.384*	*0.432*	*0.553*	*0.666*	*0.648*
FWSVD 75%	**0.719**	**0.704**	**0.71**	0.633	**0.841**	**0.843**	0.783	0.793	0.498	**0.711**	0.526	0.336
control t.	*0.417*	*0.462*	*0.571*	*0.3*	*0.507*	*0.397*	*0.341*	*0.336*	*0.527*	*0.577*	*0.672*	*0.546*
ASVD 75%	0.680	0.671	0.673	**0.651**	0.820	0.815	**0.813**	**0.813**	***0.777***	0.72	**0.760**	**0.750**
control t.	*0.432*	*0.421*	*0.581*	*0.499*	*0.390*	*0.401*	*0.388*	*0.410*	*0.591*	*0.566*	*0.561*	*0.615*

Note. The best compression results for each compression rate are highlighted in bold.

Figure 2

*Line graphs for each of the layers of **Llama 2 7b**. Naive SVD is used as compression method.*

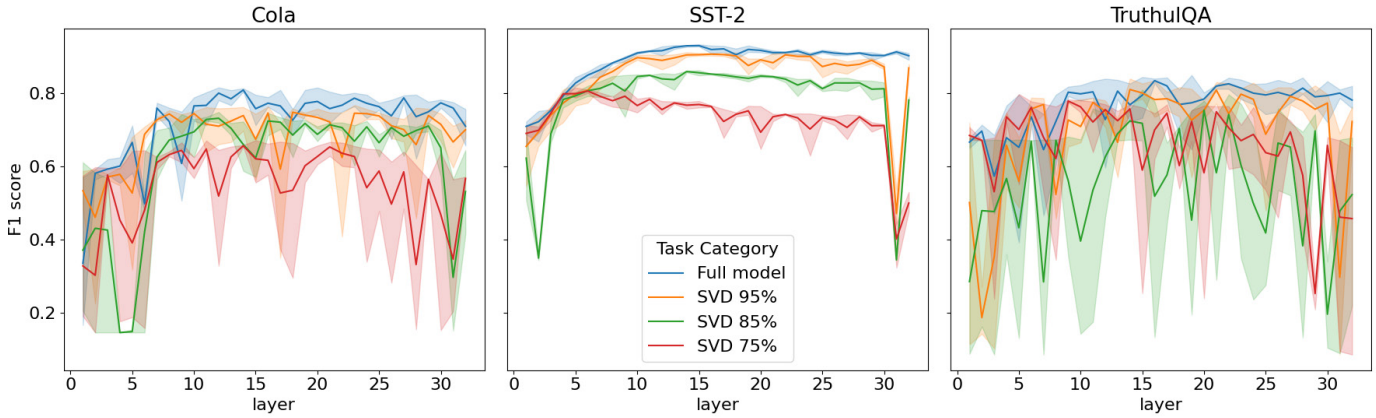
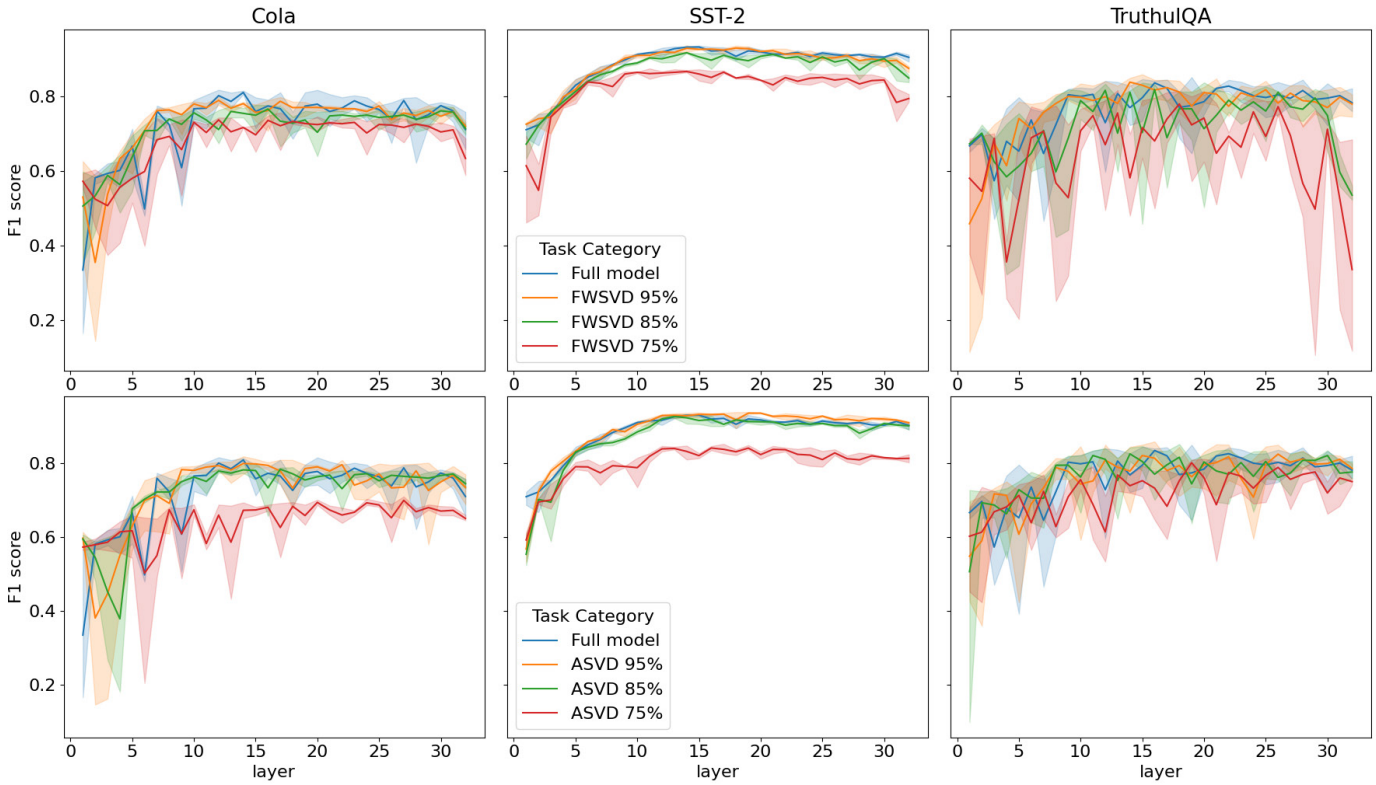


Figure 3

*Line graphs for each of the layers of **Llama 2 7b***



Note. FWSVD and ASVD is used as compression method.

performance levels. Understanding how compression affects not only the overall quality but also the internal representations within these models is crucial. This knowledge can inform the development of more efficient compression algorithms that preserve essential features necessary for complex task performance.

Our findings demonstrate a strong correlation between compression and loss of model quality, confirming RQ1. This aligns with previous research indicating that certain model layers are more vulnerable to compression-induced degradation (Chen et al., 2020). However, unlike earlier studies that largely focused on aggregate performance metrics such as overall accuracy or perplexity, our approach delved deeper into the internal representations of models and their behavior at the layer level. By examining both encoder and decoder architectures, we reveal how the internal structure of the model can become less robust as compression inten-

sifies, thus contributing to a more nuanced understanding of how and why quality degrades.

This trend is more significant in the decoder model compared to the encoder model. Different tasks exhibit varying degrees of quality degradation with compression. SST-2 remains consistent across all models, whereas CoLA demonstrates a decline in quality with model variation. This suggests that some layers entirely lose their capacity to generate outputs rather than merely degrading in quality. TruthfulQA, the most challenging task, exhibits the most substantial quality drop, with significant instability between model layers; at high compression levels, it ceases to function effectively, yielding results akin to random sampling. It is evident that compression not only diminishes the knowledge within the compressed layer but also affects other related aspects outside our specific focus. For instance, with a 30% compression rate in the BERT-base-uncased decoder using standard SVD, the model fails to produce the desired results (Figure 1.), demonstrating a correlation quality of 0.03, or an F-measure of 0.5. Still, the model retains some residual knowledge of CoLA, achieving an F-measure of approximately 0.6 on the last four layers, outperforming random responses (Table 2.).

With respect to RQ2, our results show that advanced factorization methods like ASVD and FWSVD improve model quality retention compared to standard SVD. While (Chen et al., 2020) suggested that certain layers are inherently more difficult to compress, our findings expand upon this by demonstrating that selective and refined factorization techniques can mitigate these vulnerabilities. Conversely, for the Llama 2 7b decoder model, ASVD consistently delivers superior results, as evidenced by Table 3 and illustrated in Figures 3 and 4. Notably, even with a maximum compression of 25%, the Llama model retains no more than a 10% quality loss for CoLA and SST-2 tasks, but completely forgets TruthfulQA, resulting in an MMLU benchmark score of 0.285, almost equivalent to random choice (0.25). Furthermore, Figures 2 and 3 highlight a significant difference between ASVD, FWSVD and SVD in relation to SST-2. SVD exhibits a quality drop in the final layers, which is less in FWSVD and absent in ASVD. This capability allows ASVD to achieve superior results for complex tasks. Moreover, we contribute evidence to support and refine the assertions of previous works (Ji et al., 2024; Yuan et al., 2023), who proposed alternative compression approaches but did not fully account for layer-specific sensitivities. By employing ASVD and FWSVD, we illustrate a concrete pathway towards preserving critical internal features that standard SVD often fails to maintain. This deeper analysis and interpretation of the obtained results extends previous works, offering new strategies to better control how compression impacts different parts of a model's internal structure.

Our investigation into RQ3, whether compression leads to irreversible loss of knowledge, provides both confirmation of and contrast to existing literature. Similar to prior studies reporting irreversible degradation in certain architectures or tasks (Sharma et al., 2023), we find that challenging tasks such as TruthfulQA suffer disproportionately under high compression rates. Yet, our layer-wise probing and fine-tuning experiments reveal that not all knowledge is equally affected: while some tasks all but vanish under extreme compression, simpler tasks like SST-2 remain largely intact. This more differentiated picture advances the field's understanding of knowledge retention, suggesting that the vulnerability of knowledge to compression may depend on the complexity and nature of the task, rather than reflecting a uniform process of forgetting.

Compared to previous research, our study delves deeper into the literature by confirming previous findings on the existence of "incompressible" layers (Chen et al., 2020) and expanding the scope by proposing solutions through factorization variants such as ASVD and FWSVD. While our findings do not completely solve the challenge of model compression without loss of accuracy, they represent a significant step towards balancing efficiency and model integrity, pointing to promising avenues for further exploration. For instance, Sharma et al. (2023) highlighted the cumulative impact of noise during compression, an aspect we did not specifically address. This gap suggests potential synergies between our methods and other noise mitigation strategies, encouraging future research that integrates complementary findings to achieve better compression results.

CONCLUSION

This study demonstrates that increased model compression leads to a decrease in both model performance and the quality of hidden representations. This effect is more pronounced in decoder models compared to encoder models. The decrease is dependent on the task and layer, and more complex tasks are more adversely affected by compression.

Our findings highlight the importance of considering the effect of compression on various model architectures and tasks. In particular, we found that the FWSVD method outperformed standard SVD at higher compression rates for encoder models like BERT in terms of preserving model quality. For decoder models like Llama-2 we see a similar picture, but besides FWSVD we can use additionally ASVD which shows even better results. These results suggest that both FWSVD and ASVD can effectively reduce some of the negative effects of compression by improving the compressibility of layers that would otherwise be incompressible. This helps maintain model performance, but irreversible knowl-

edge loss at the layer level continues to be a significant factor leading to performance decline, especially in more complex tasks.

Future research should focus on exploring factors such as noise during compression and developing more advanced compression techniques in order to fully address these issues. Improving methods like ASVD may lead to better preservation of model performance at higher compression rates. In addition, it may be worth to use the probing results as an estimate and threshold to prepare the model for compression.

DECLARATION OF COMPETITING INTEREST

None declared.

REFERENCES

Belinkov, Y. (2021). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, *48*(1), 207–219. https://doi.org/10.1162/COLI_a_00422

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020). Language Models are few-shot learners. *Advances in Neural Information Processing Systems*. https://arxiv.org/abs/2005.14165v4

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., & Carbin, M. (2020). The lottery ticket hypothesis for pre-trained BERT networks. *Advances in Neural Information Processing Systems*. https://arxiv.org/abs/2007.12223v2

Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, *35*(1), 126–136. https://doi.org/10.1109/MSP.2017.2765695

Dettmers, T., Lewis, M., Shleifer, S., & Zettlemoyer, L. (2021). 8-bit Optimizers via block-wise quantization. *ICLR 2022 - 10th International Conference on Learning Representations*, *8*, 105–125. Curran Associates, Inc. https://arxiv.org/abs/2110.02861v2

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. Association for Computational Linguistics. https://arxiv.org/abs/1810.04805v2

Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Sajjad, H., Nakov, P., Chen, D., & Winslett, M. (2021). Compressing large-scale transformer-based models: A case study on BERT. *Transactions of the Association for Computational Linguistics*, *9*, 1061–1080. https://doi.org/10.1162/TACL_A_00413

Guo, Y., Yao, A., & Chen, Y. (2016). Dynamic network surgery for efficient DNNs. *Advances in Neural Information Processing Systems* (pp. 1387–1395). Morgan Kaufmann Publishers Inc. https://arxiv.org/abs/1608.04493v2

Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both Weights and Connections for Efficient Neural Networks. *Advances in Neural Information Processing Systems* (pp. 1135–1143). https://arxiv.org/abs/1506.02626v3

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring Massive Multitask Language Understanding. *ICLR 2021 - 9th International Conference on Learning Representations* (pp. 1343–1355). OpenReview.net. https://arxiv.org/abs/2009.03300v3

Hewitt, J., & Liang, P. (2019). Designing and Interpreting Probes with Control Tasks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference* (pp. 2733–2743). Association for Computational Linguistics. https://doi.org/10.18653/v1/d19-1275

Hsu, Y. C., Hua, T., Chang, S. E., Lou, Q., Shen, Y., & Jin, H. (2022). Language model compression with weighted low-rank factorization. *ICLR 2022 - 10th International Conference on Learning Representations*. https://arxiv.org/abs/2207.00112v1

Ji, Y., Xiang, Y., Li, J., Chen, W., Liu, Z., Chen, K., & Zhang, M. (2024). *Feature-based low-rank compression of large language models via bayesian optimization* (pp. 844–857). OpenReview.net. https://arxiv.org/abs/2405.10616v1

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *scaling laws for neural language models*. https://arxiv.org/abs/2001.08361v1

Kim, Y. D., Park, E., Yoo, S., Choi, T., Yang, L., & Shin, D. (2015). Compression of deep convolutional neural networks for fast and low power mobile applications. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. OpenReview.net. https://arxiv.org/abs/1511.06530v2

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., & Alistarh, D. (2022). 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, EMNLP 2022*, 4163–4181. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.279

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. *8th International Conference on Learning Representations, ICLR 2020*. Curran Associates, Inc. https://arxiv.org/abs/1909.11942v6

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L., & Kawsar, F. (2016). DeepX: A software accelerator for low-power deep learning inference on mobile devices. *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2016 - Proceedings*. IEEE Press. https://doi.org/10.1109/IPSN.2016.7460664

Li, K., Patel, O., Viégas, F., Pfister, H., & Wattenberg, M. (2023). Inference-Time intervention: Eliciting truthful answers from a Language Model. *Advances in Neural Information Processing Systems*, *36*. https://arxiv.org/abs/2306.03341v6

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *1*, 3214–3252. https://doi.org/10.18653/V1/2022.ACL-LONG.229

Michel, P., Levy, O., & Neubig, G. (2019). Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1905.10650v3

Narayanan, D., Phanishayee, A., Shi, K., Chen, X., & Zaharia, M. (2020). Memory-efficient pipeline-parallel DNN training. *Proceedings of Machine Learning Research*, *139*, 7937–7947. https://arxiv.org/abs/2006.09503v3

Sharma, P., Ash, J. T., & Misra, D. (2023). The truth is in there: improving reasoning in Language Models with layer-selective rank reduction. *12th International Conference on Learning Representations, ICLR 2024*. OpenReview.net https://arxiv.org/abs/2312.13558v1

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). *Recursive deep models for semantic compositionality over a sentiment treebank* (pp. 1631–1642). ACM. https://aclanthology.org/D13-1170

Tai, C., Xiao, T., Zhang, Y., Wang, X., & Weinan, E. (2015). Convolutional neural networks with low-rank regularization. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. OpenReview.net https://arxiv.org/abs/1511.06067v3

Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., & Lin, J. (2019). *Distilling task-specific knowledge from BERT into simple neural networks*. https://arxiv.org/abs/1903.12136v1

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*. https://arxiv.org/abs/2302.13971v1

Touvron, H., Martin, L., Stone, K.R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D.M., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A.S., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I.M., Korenev, A.V., Koura, P.S., Lachaux, M., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M.H., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., & Scialom, T. (2023). *Llama 2: Open foundation and fine-tuned chat models*. https://arxiv.org/abs/2307.09288v2

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-December*, 5999–6009. https://arxiv.org/abs/1706.03762v7

Wang, N., Choi, J., Brand, D., Chen, C. Y., & Gopalakrishnan, K. (2018). Training deep neural networks with 8-bit floating point numbers. *Advances in Neural Information Processing Systems*, *2018-December*, 7675–7684. https://arxiv.org/abs/1812.08011v1

Wang, Z., Wohlwend, J., & Lei, T. (2019a). Structured pruning of Large Language Models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 6151–6162. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.496

Wang, Z., Wohlwend, J., & Lei, T. (2019b). Structured pruning of Large Language Models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 6151–6162. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.496

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641. https://doi.org/10.1162/TACL_A_00290

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). *Emergent abilities of Large Language Models*. https://arxiv.org/abs/2206.07682v2

Xu, C., Yao, J., Lin, Z., Ou, W., Cao, Y., Wang, Z., & Zha, H. (2018). Alternating multi-bit quantization for recurrent neural networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. OpenReview.net. https://arxiv.org/abs/1802.00150v1

Yin, L., Jaiswal, A., Liu, S., Kundu, S., & Wang, Z. (2023). *Pruning small pre-trained weights irreversibly and monotonically impairs "difficult" downstream tasks in LLMs*. https://arxiv.org/abs/2310.02277v2

Yu, H., & Wu, J. (2023). Compressing transformers: Features are low-rank, but weights are not! *AAAI Conference on Artificial Intelligence*, *37*, 11007–11015. https://doi.org/10.1609/AAAI.V37I9.26304

Yuan, Z., Shang, Y., Song, Y., Wu, Q., Yan, Y., & Sun, G. (2023). *ASVD: Activation-aware singular value decomposition for compressing Large Language Models*. https://arxiv.org/abs/2312.05821v4

Zafrir, O., Larey, A., Boudoukh, G., Shen, H., & Wasserblat, M. (2021). *Prune once for all: Sparse pre-trained Language Models*. https://arxiv.org/abs/2111.05754v1

Zhang, T., Lin, Z., Yang, G., & De Sa, C. (2019). QPyTorch: A low-precision arithmetic simulation framework. *Proceedings - 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing* (pp. 10–13). Curran Associates Inc. https://doi.org/10.1109/EMC2-NIPS53020.2019.00010

APPENDIX

As a verification of our conclusions in the main paper, we performed more experiments with a more modern version of llama: llama 3.1. As factorization methods, we use the standard SVD and ASVD, which has performed well in LLama 2 compression.

Figure 4

*Line graphs for each of the layers of **Llama 3.1 8b**. Naive SVD and ASVD are used as compression method.*

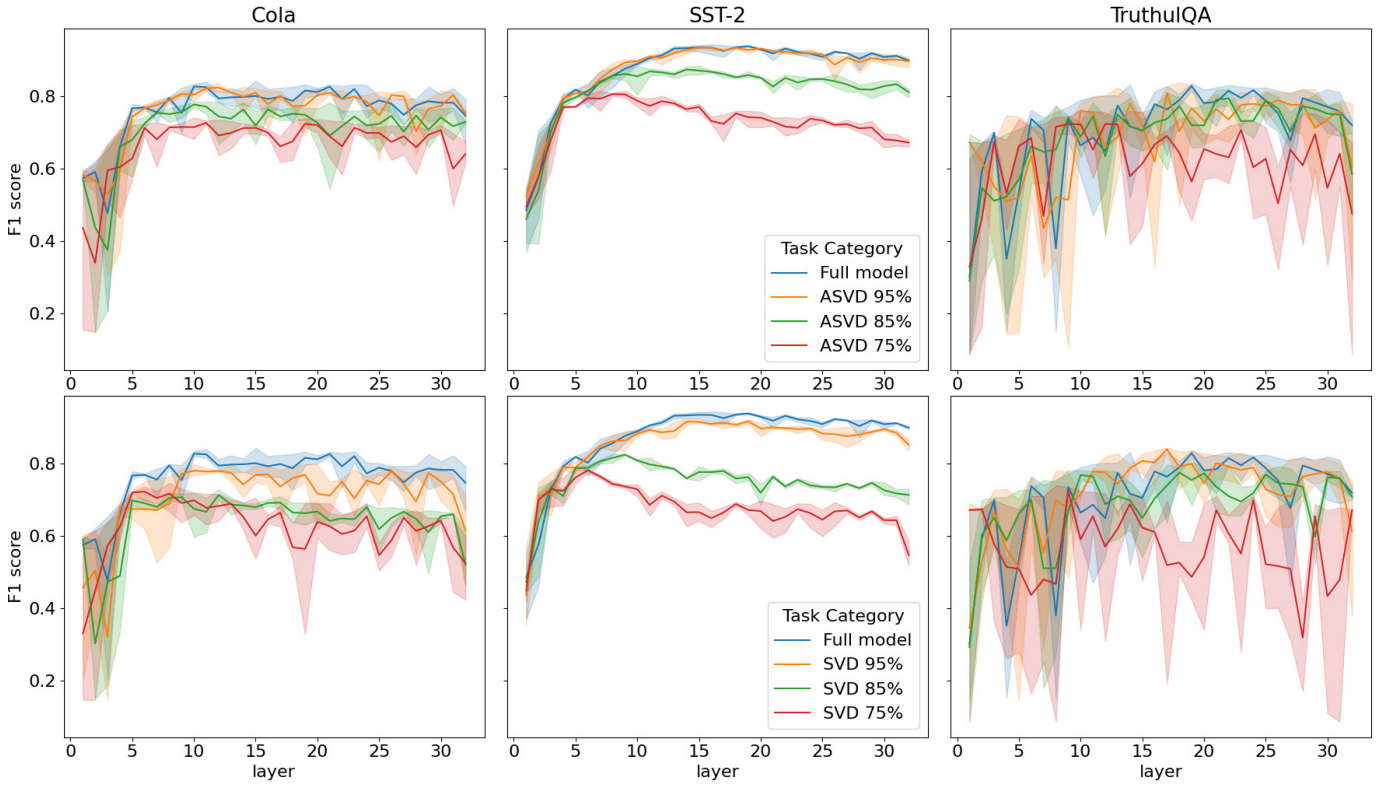


Table 5

*Results of the top 4 layers of the decoder **Llama 3.1 8b** model with additional control task (control t.). The best compression results for each compression rate are highlighted in bold.*

Dataset	CoLA				SST-2				TruthfulQA			
Layer	29	30	32	32	29	30	32	32	29	30	32	32
w\o compress	0.787	0.783	0.783	0.747	0.920	0.909	0.913	0.899	0.783	0.771	0.758	0.720
control t.	0.579	0.563	0.387	0.569	0.396	0.352	0.469	0.417	0.629	0.647	0.6	0.596
SVD 95%	**0.774**	0.751	0.715	0.615	0.888	0.896	0.885	0.853	**0.772**	**0.778**	0.759	**0.612**
control t.	0.438	0.249	0.401	0.429	0.426	0.436	0.403	0.39	0.604	0.616	0.244	0.602
ASVD 95%	0.766	**0.773**	**0.804**	**0.754**	**0.906**	**0.902**	**0.902**	**0.898**	0.712	0.737	**0.762**	0.607
control t.	0.412	0.378	0.432	0.509	0.357	0.370	0.393	0.385	0.625	0.603	0.606	0.606
SVD 85%	0.610	0.654	0.660	0.519	0.747	0.726	0.718	0.713	0.597	**0.762**	**0.760**	**0.708**
control t.	0.455	0.433	0.565	0.312	0.339	0.423	0.337	0.4	0.431	0.608	0.291	0.546
ASVD 85%	**0.708**	**0.742**	**0.718**	**0.729**	**0.819**	**0.829**	**0.834**	**0.812**	**0.766**	0.752	0.749	0.685
control t.	0.396	0.42	0.401	0.521	0.441	0.361	0.512	0.399	0.62	0.598	0.602	0.612
SVD 75%	0.627	0.642	0.568	0.524	0.668	0.643	0.643	0.547	0.655	0.433	0.479	**0.672**

Dataset	CoLA				SST-2				TruthfulQA			
Layer	**29**	**30**	**32**	**32**	**29**	**30**	**32**	**32**	**29**	**30**	**32**	**32**
control t.	0.565	0.356	0.574	0.584	0.372	0.35	0.366	0.384	0.432	0.553	0.666	0.648
ASVD 75%	**0.694**	**0.706**	**0.6**	**0.640**	**0.714**	**0.681**	**0.678**	**0.672**	**0.696**	**0.547**	**0.641**	0.476
control t.	0.432	0.421	0.581	0.499	0.390	0.401	0.388	0.410	0.591	0.566	0.561	0.615

In a result, we see a similar pattern to that observed in the research with Llama 2 7b: TruthfulQA probing performs poorly with SVD, and much better with AVD. It is also noticeable that llama 3.1 is much less compressible, as we see a rapid drop in quality on SST-2 when compressed. At the same time, a small compression of 5% under ASVD has virtually no effect on a simpler dataset such as SST-2 and CoLA. From this we can conclude that our study is scalable to other LLM models.