LatentLLM: Attention-Aware Joint Tensor Compression

Koike-Akino, Toshiaki; Chen, Xiangyu; Liu, Jing; Wang, Ye; Wang, Pu; Brand, Matthew

TR2025-075 June 07, 2025

Abstract

We propose a new framework to convert a large foundation model such as large language models (LLMs)/large multi- modal models (LMMs) into a reduced-dimension latent structure. Our method uses a global attention-aware joint tensor decomposition to significantly improve the model efficiency. We show the benefit on several benchmark including multi-modal reasoning tasks.

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop 2025

LatentLLM: Attention-Aware Joint Tensor Compression

Toshiaki Koike-Akino, Xiangyu Chen, Jing Liu, Ye Wang, Pu (Perry) Wang, Matthew Brand
Mitsubishi Electric Research Laboratories (MERL)
201 Broadway, Cambridge, MA 02139, USA.
{koike, xiachen, jiliu, yewang, puwang, brand}@merl.com

Abstract

We propose a new framework to convert a large foundation model such as large language models (LLMs)/large multi-modal models (LMMs) into a reduced-dimension latent structure. Our method uses a global attention-aware joint tensor decomposition to significantly improve the model efficiency. We show the benefit on several benchmark including multi-modal reasoning tasks.

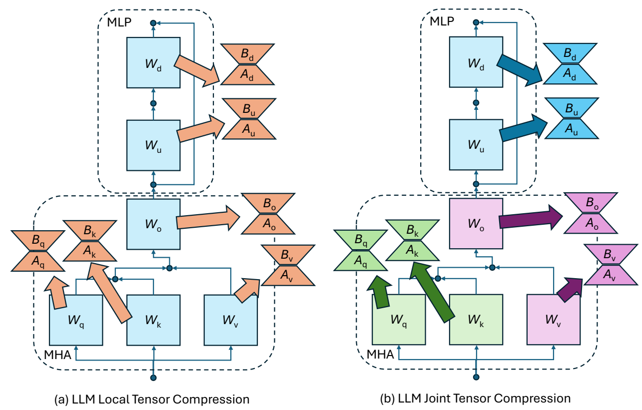


Figure 1. Reduced-dimension LLM/LMM with low-rank tensor decomposition. (a) each module is locally compressed. (b) multiple modules are globally compressed.

1. Introduction

Large language models (LLMs) [1, 27] and large multi-modal models (LMMs) [19] have shown excellent performance across a variety of general tasks [4, 14, 29]. Nonetheless, these models having billions of parameters demand significant computational resources [25]. Towards increasing the accessibility of LLMs/LMMs for limited resource devices, extensive efforts have been devoted to model compression [2, 30, 34]: e.g., partial activation [13, 16], pruning [3, 8, 10, 26], quantization [9, 17, 28], distillation [6, 11, 12], and low-rank factorization [12, 18, 32].

Recently DeepSeek-V3 [18] has attracted much attention for its high efficiency with latent reduction. It employs a multi-head latent attention (MLA) to compress the multi-head attention (MHA), realizing an efficient KV cache [5]. In this paper, we provide a novel solution to convert a pretrained LLM/LMM built with MHA into a compressed LLM/LMM with MLA. Our approach is motivated by a global compression framework [3, 28], while we adopt it for tensor rank reduction not for pruning or quantization. Our derived solution is based on a high-order tensor-rank decomposition to jointly factorize multiple layers.

The contributions of our paper are summarized below.

- We propose a novel low-rank decomposition approach called LatentLLM to compress LLMs/LMMs.
- We discuss an optimal pre-conditioning for activation-aware SVD.
- We reveal that a choice of junction matrix can significantly reduce the model size.
- We then introduce an attention-aware joint SVD framework to compress multiple weights at the same time.
- Our experiments validate that our LatentLLM approach can improve the efficiency of LLM/LMM.
- The latent LLaVA with our method offers a significant advantage in multi-modal reasoning capability.

2. LatentLLM: Tensor compression

2.1. Reduced-dimension LLM/LMM

Fig. 1 illustrates the basic transformer architecture consisting of MHA and MLP, used in typical LLMs/LMMs. For MLP, there are up and down projections, whereas MHA has query/key/value/output projections. By transforming those dense weight matrices into low-rank decompositions, we can realize an efficient latent LLM/LMM having potential benefits: (i) fewer-parameter model size; (ii) KV cache reduction; (iii) accelerated processing; (iv) lower-power consumption. In fact, some recent LLM models such as DeepSeek-V3 [18] demonstrated efficiency and high-performance with MLA. We focus on compressing a pre-trained LLM/LMM by converting MHA into MLA in a

Table 1. Variants of pre-conditioning matrices P.

Conditioning P	Expression
Identity (Plain SVD) [7, 24]	I
Diagonal Hessian [8–10]	$\mathsf{diag}[(XX^\top + \lambda I)^{-1}]^{\frac{-1}{2}}$
Diagonal ℓ_1-norm [17, 32]	$\mathsf{diag}\big[\sum_j \lvert X_{1,j}\rvert, \ldots, \sum_j \lvert X_{d,j}\rvert\big]^\alpha$
Diagonal ℓ_2-norm [26]	$\mathsf{diag}[XX^\top]^{\frac{1}{2}}$
Covariance [31]	$XX^\top + \lambda I$
Root-Covariance (Ours)	$(XX^\top + \lambda I)^{\frac{1}{2}}$

zero-shot fashion, i.e., without any fine-tuning.

Most existing methods are based on a local optimization to approximate each weight individually. Motivated by recent global optimization [3, 28], we propose a joint tensor compression method that we call "LatentLLM." Before describing our solution, we first address activation-aware compression to provide some new insights below.

2.2. Activation-aware SVD: Pre-conditioning

A pioneering work by ASVD [32] introduced a way to compress a layer depending on the activation statistics. Consider a pretrained-weight $W \in \mathbb{R}^{d' \times d}$ to compress with a lower-rank decomposition $\hat{W} = BA$ for compression matrix $A \in \mathbb{R}^{r \times d}$ and decompression matrix $B \in \mathbb{R}^{d' \times r}$. Using the input activation $X \in \mathbb{R}^{d \times l}$ (l is the calibration sample length), ASVD aims to minimize the activation loss:

$$\mathcal{L}_1 = \mathbb{E}_X \big\lVert WX - \hat{W}X \big\rVert^2 = \mathbb{E}_X \big\lVert WX - BAX \big\rVert^2, \quad (1)$$

instead of the naïve weight-based loss: $\mathcal{L}_0 = \lVert W - \hat{W} \rVert^2$. While the optimal solution to minimize $\mathcal{L}_0$ can be given by the plain SVD of W, to minimize $\mathcal{L}_1$, ASVD introduced a pre-conditioning matrix $P \in \mathbb{R}^{d \times d}$ to whiten the statistical impact of the activation X. Specifically, ASVD uses the low-rank matrices given by whitened SVD:

$$BAP = \mathsf{svd}_r[WP], \quad (2)$$

where $\mathsf{svd}_r[\cdot]$ denotes the rank-r truncated SVD.

The optimal pre-conditioning matrix P can be given by reformulating $\mathcal{L}_1$ as follows:

$$\mathcal{L}_1 = \mathrm{tr}\big[(W - BA)\mathbb{E}_X[XX^\top](W - BA)^\top\big] \quad (3)$$
$$= \big\lVert (W - BA)C^{\frac{1}{2}} \big\rVert^2 = \big\lVert WC^{\frac{1}{2}} - BAC^{\frac{1}{2}} \big\rVert^2, \quad (4)$$

where $C = \mathbb{E}_X[XX^\top] \in \mathbb{R}^{d \times d}$ is a covariance of input activation. Hence, the above loss can be minimized by the SVD: $BAC^{\frac{1}{2}} = \mathsf{svd}_r[WC^{\frac{1}{2}}]$. Accordingly, it is found that the optimal pre-conditioner is the square-root covariance: $P = C^{\frac{1}{2}}$. Given the finite calibration data X, we can estimate the covariance as $C = XX^\top + \lambda I$, where the damping factor $\lambda \in \mathbb{R}_+$ corresponds to the shrunk estimator [15].

Remark 1 *Different pre-conditioning was introduced for pruning and quantization, as listed in Tab. 1.*

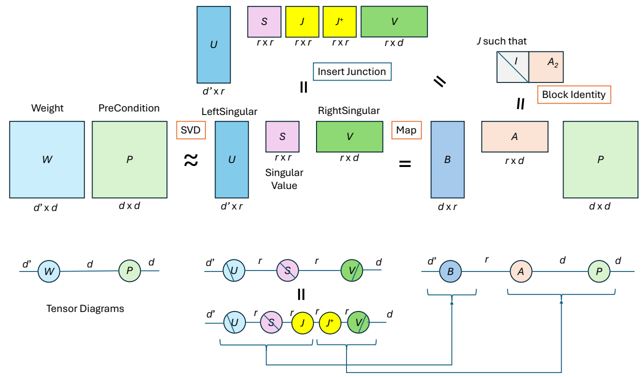


Figure 2. Activation-aware compression with pre-conditioning and junction matrix. The junction matrix J can be adjusted to save the number of parameters and inference computation.

2.3. Junction matrix for model compression

The solution of Eq. (2) has non-unique decomposition for matrices B and A. Consider the truncated SVD written as $USV = \mathsf{svd}_r[WP]$, where $U \in \mathbb{R}^{d' \times r}$, $S \in \mathbb{R}^{r \times r}$, and $V \in \mathbb{R}^{r \times d}$ are the left singular unitary matrix, singular-value diagonal matrix, and right singular unitary matrix, respectively. Hence, the matrices B and A can be expressed:

$$B = USJ, \qquad A = J^+ V P^+, \quad (5)$$

where $J \in \mathbb{R}^{r \times r}$ is a junction matrix and $[\cdot]^+$ denotes the pseudo inverse. Choosing any junction matrix that satisfies $SJJ^+ = S$ has no impact on the loss. Hence, there is few literature discussing the choice of J.

However, a certain choice of J has a noticeable advantage to reduce the number of parameters and floating-point operations (FLOPs). We can write the whitened right-singular matrix VP^+ as two sub-blocks: $VP^+ = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$, for $V_1 \in \mathbb{R}^{r \times r}$ and $V_2 \in \mathbb{R}^{r \times (d-r)}$. When we use $J = V_1$, the compression matrix A will contain an identity block as long as V_1 is non-singular:

$$A = J^+ V P^+ = V_1^+ \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} I & V_1^+ V_2 \end{bmatrix}. \quad (6)$$

This can greatly reduce the number of parameters from $r(d' + d)$ to $r(d' + d) - r^2$, as well as the FLOPs, because no computation is needed for the identity projection. Fig. 2 depicts the role of the pre-conditionning and junction matrices for the activation-aware compression, showing the flexibility of tensor mapping with the tensor diagrams.

Remark 2 *Pivoting columns solves the case when V_1 is singular, without additional FLOPs in inference.*

3. LatentLLM: Joint tensor compression

The SVD described above is optimal in the sense that the local error is minimized for the single tensor compression,

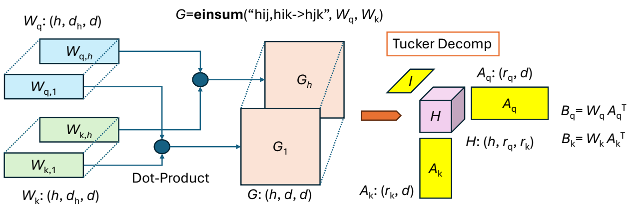


Figure 3. Tucker decomposition for joint QK compression.

whereas it does not guarantee global optimality. Motivated by SparseLLM [3], we propose a joint tensor compression technique which factorizes multiple tensors concurrently.

3.1. Multi-head latent attention: Joint QK SVD

First, we consider a joint compression of query (Q) and key (K) projections in MHA to convert into MLA. The attention map is the dot product of query and key features: $M_i = X^\top W_{\mathrm{q},i}^\top W_{\mathrm{k},i} X$, where $M_i \in \mathbb{R}^{l\times l}$ is the ith head attention map before softmax operation, $W_{\mathrm{q},i} \in \mathbb{R}^{d_\mathrm{h}\times d}$ is the ith head query projection matrix, and $W_{\mathrm{k},i} \in \mathbb{R}^{d_\mathrm{h}\times d}$ is the ith head key projection matrix. Here, d_h is the head dimension, which is often $d_\mathrm{h} = d/h$ for the number of heads h.

We consider minimizing the attention map error:

$$\mathcal{L}_2 = \textstyle\sum_{i=1}^{h} \big\| M_i - X^\top A_\mathrm{q}^\top B_{\mathrm{q},i}^\top B_{\mathrm{k},i} A_\mathrm{k} X \big\|^2, \tag{7}$$

where $A_\mathrm{q} \in \mathbb{R}^{r_\mathrm{q}\times d}$ is for the Q compression, $A_\mathrm{k} \in \mathbb{R}^{r_\mathrm{k}\times d}$ is for the K compression, $B_{\mathrm{q},i} \in \mathbb{R}^{d_\mathrm{h}\times r_\mathrm{q}}$ is for the ith head Q decompression, and $B_{\mathrm{k},i} \in \mathbb{R}^{d_\mathrm{h}\times r_\mathrm{k}}$ is for the ith head K decompression, respectively. Here, r_q and r_k are the latent dimensions for Q and K. Similar to Eq. (4), we can rewrite:

$$\mathcal{L}_2 = \textstyle\sum_{i=1}^{h} \big\| G_i - A_\mathrm{q}'^\top H_i A_\mathrm{k}' \big\|^2, \tag{8}$$

where $G_i = C^{\frac{1}{2}} W_{\mathrm{q},i}^\top W_{\mathrm{k}} C^{\frac{1}{2}}$, $A_\mathrm{q}' = A_\mathrm{q} C^{\frac{1}{2}}$, $A_\mathrm{k}' = A_\mathrm{k} C^{\frac{1}{2}}$, and $H_i = B_{\mathrm{q},i}^\top B_{\mathrm{k},i}$ This is known as a high-order SVD (HOSVD) problem to decompose for the 3-mode tensor $G \in \mathbb{R}^{h\times d\times d}$, whose ith slice is G_i. A_q' corresponds to the 2nd tensor plane, A_k' is the 3rd tensor plane, and $H \in \mathbb{R}^{h\times r_\mathrm{q}\times r_\mathrm{k}}$, whose ith slice is H_i, is the tensor core.

This is illustrated in Fig. 3. This Tucker tensor decomposition is typically solved by alternating SVD over each slice sequentially. Algorithm 1 shows the pseudo-code of the joint SVD compression for QK latent projections. Here, we generalize the pre-conditioning matrix P, as not necessarily the optimal $C^{\frac{1}{2}}$. In addition, we explicitly denoted any arbitrary junction matrices that do not change the error. Note that there are additional junction matrices per heads $J_i \in \mathbb{R}^{d_\mathrm{h}\times d_\mathrm{h}}$ as well as individual Q/K junctions $J_\mathrm{q} \in \mathbb{R}^{r_\mathrm{k}\times r_\mathrm{k}}$ and $J_\mathrm{k} \in \mathbb{R}^{r_\mathrm{k}\times r_\mathrm{k}}$. This suggests that we can further reduce the number of parameters by transforming into the block identity form per head.

Algorithm 1 Joint SVD for QK Projections in MHA

Input: Pre-conditioning $P \in \mathbb{R}^{d\times d}$, query projection heads $W_{\mathrm{q},i} \in \mathbb{R}^{d_\mathrm{h}\times d}$, key projection heads $W_{\mathrm{k},i} \in \mathbb{R}^{d_\mathrm{h}\times d}$, number of heads h, rank $r_\mathrm{q}, r_\mathrm{k}$, iteration N
Initialize:
$W_{\mathrm{q},i} = W_{\mathrm{q},i} P$ for $i \in \{1, \ldots, h\}$
$W_{\mathrm{k},i} = W_{\mathrm{k},i} P$ for $i \in \{1, \ldots, h\}$
$G_i = W_{\mathrm{q},i}^\top W_{\mathrm{k},i}$ for $i \in \{1, \ldots, h\}$
$A_\mathrm{q} = \mathsf{RightSingular}_{r_\mathrm{q}} \big[\sum_{i=1}^{h} G_i G_i^\top \big]$
for $n = 1$ **to** N **do**
  $A_\mathrm{k} = \mathsf{RightSingular}_{r_\mathrm{k}} \big[\sum_{i=1}^{h} G_i^\top A_\mathrm{q}^\top A_\mathrm{q} G_i \big]$
  $A_\mathrm{q} = \mathsf{RightSingular}_{r_\mathrm{q}} \big[\sum_{i=1}^{h} G_i A_\mathrm{k} A_\mathrm{k}^\top G_i^\top \big]$
end for
Output:
$B_{\mathrm{q},i} = J_i^\top W_{\mathrm{q},i} A_\mathrm{q}^\top J_\mathrm{q}$ for $i \in \{1, \ldots, h\}$
$B_{\mathrm{k},i} = J_i^+ W_{\mathrm{k},i} A_\mathrm{k}^\top J_\mathrm{k}$ for $i \in \{1, \ldots, h\}$
$A_\mathrm{q} = J_\mathrm{q}^+ A_\mathrm{q} P^+$
$A_\mathrm{k} = J_\mathrm{k}^+ A_\mathrm{k} P^+$

3.2. Multi-head latent attention: Joint VO SVD

Next, we discuss the joint SVD for value (V) and output (O) projections in MHA. For any arbitrary attention map, we may consider minimizing the loss:

$$\mathcal{L}_3 = \textstyle\sum_{i=1}^{h} \big\| W_{\mathrm{o},i} W_{\mathrm{v},i} X - \hat{W}_{\mathrm{o},i} \hat{W}_{\mathrm{v},i} X \big\|^2, \tag{9}$$

where $W_{\mathrm{o},i} \in \mathbb{R}^{d'\times d_\mathrm{h}}$ is the ith head output projection, and $W_{\mathrm{v},i} \in \mathbb{R}^{d_\mathrm{h}\times d}$ is the ith head value value projection. Here we design the low-rank compression: $\hat{W}_{\mathrm{o},i} = B_\mathrm{o} A_{\mathrm{o},i} \in \mathbb{R}^{d'\times d_\mathrm{h}}$ and $\hat{W}_{\mathrm{v},i} = B_{\mathrm{v},i} A_\mathrm{v} \in \mathbb{R}^{d_\mathrm{h}\times d}$ with $B_\mathrm{o} \in \mathbb{R}^{d'\times r_\mathrm{o}}$, $A_{\mathrm{o},i} \in \mathbb{R}^{r_\mathrm{o}\times d_\mathrm{h}}$, $B_{\mathrm{v},i} \in \mathbb{R}^{d_\mathrm{h}\times r_\mathrm{v}}$, and $A_\mathrm{v} \in \mathbb{R}^{r_\mathrm{v}\times d}$. Interestingly, this is also formulated in a similar manner of Eq. (8), and it can be solved by the joint SVD algorithm.

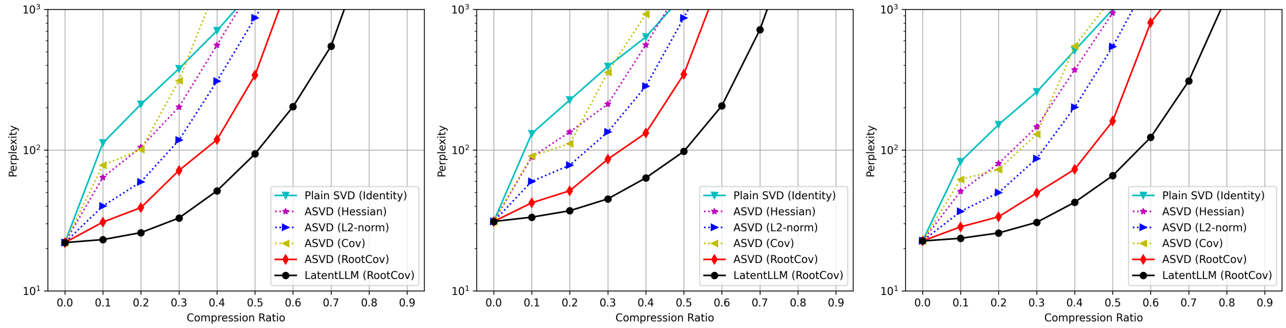
3.3. Latent MLP: Joint UD SVD

Finally, we address the joint compression of MLP layers which consists of up (U) projection and down (D) projection in typical LLMs/LMMs. Although the global optimization is generally difficult due to the nonlinear activations in the MLP layer, SparseLLM [3] provides an elegant way to approximate MLP layer. The key idea is to minimize the MLP loss in a decoupled manner by introducing auxiliary variables. Our LatentLLM exploits the same philosophy to compress MLP layers not to prune. Refer more details on the decoupled optimization in SparseLLM [3].

4. Experiments

We conduct experiments for LLM and LMM benchmarks to evaluate the effectiveness of our method, based on the same setting of SparseLLM [3] and their code base[1]. For LLM

[1] https://github.com/BaiTheBest/SparseLLM



(a) WT2 (OPT-350M) (b) PTB (OPT-350M) (c) C4 (OPT-350M)

Figure 4. Perplexity (↓) over compression ratio for OPT models.

Table 2. Accuracy in percent (↑) on ScienceQA dataset of LLaVA model with different compression methods for 10%–20% size reduction. Question subjects: natural science (NAT); social science (SOC); language science (LAN). Context modality: text (TXT); image (IMG); or no context (NO). Grades: 1–6 (G1-6); 7–12 (G7-12).

		Subject			Context Modality			Grades		
Method	Compression	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Avg
Original un-compressed	0%	72.47	69.18	65.73	73.51	68.82	65.99	72.72	65.19	70.03
Plain SVD (Identity)	10%	5.33	1.35	0.27	5.77	6.69	0.00	3.30	2.97	3.18
ASVD (Hessian)	10%	17.23	24.97	3.18	18.43	29.55	2.16	17.40	11.27	15.21
ASVD (ℓ_2-norm)	10%	16.70	18.34	2.55	17.89	24.34	2.23	16.04	8.57	13.37
ASVD (Cov)	10%	41.21	27.22	37.91	41.30	35.15	38.33	38.62	35.27	37.42
ASVD (RootCov)	10%	64.08	56.13	57.36	64.03	60.98	57.35	62.70	57.02	60.67
LatentLLM (RootCov)	10%	**68.52**	**64.23**	**61.36**	**69.06**	**65.20**	**61.53**	**68.72**	**60.45**	**65.76**
Plain SVD (Identity)	20%	0.18	0.00	0.00	0.20	0.20	0.00	0.04	0.20	0.09
ASVD (Hessian)	20%	3.82	2.81	0.00	3.62	5.30	0.14	3.01	1.91	2.62
ASVD (ℓ_2-norm)	20%	0.44	0.79	0.00	0.39	0.79	0.07	0.51	0.20	0.40
ASVD (Cov)	20%	41.39	27.22	37.55	41.45	35.35	38.12	38.69	35.14	37.42
ASVD (RootCov)	20%	61.19	53.43	53.36	61.53	59.40	52.68	58.96	54.98	57.53
LatentLLM (RootCov)	20%	**66.39**	**61.19**	**60.82**	**67.20**	**63.41**	**60.62**	**66.41**	**59.26**	**63.85**

calibration, we use 64 samples of 2048-token segments, randomly chosen from the first shard of the C4 [23] dataset. For LMM calibration, we use 64 samples, randomly chosen from the train split of the ScienceQA [20] dataset.

For LLM, we consider the OPT model family [33] as it provides a wide range of model scales from 125M to 175B. We consider the benchmark of raw-WikiText2 (WT2) [22], the Penn Treebank (PTB) [21], and the C4 [23], popular in the related literature [8, 9, 26]. For LMM, we use LLaVA 7B [19] model. We evaluate the capability of the multi-modal answer reasoning with ScienceQA, which contains 21K questions for three subjects: natural, social, and language science. Some questions have image and/or text contexts, and the problem levels range from grade 1 to 12.

We first look into the compression capability of our LatentLLM for LLM benchmarks in Fig. 4. We can see that the conventional plain SVD has a poor performance, and that ASVD with a proper pre-conditioning can significantly improve the perplexity. Further, the joint SVD used for LatentLLM offers an additional improvement for all benchmarks.

We then show the accuracy of latent LLaVA models for ScienceQA multi-modal reasoning benchmark in Tab. 2. It is verified that our LatentLLM can significantly outperform other low-rank compression methods across diverse reasoning problems over different subjects/contexts/grades.

5. Summary

We introduced LatentLLM which jointly compresses multiple tensors through the use of high-order tensor-rank decomposition. We also provided new perspectives for choosing the pre-conditioner and junction matrix. Benchmark experiments demonstrated that the model compression performance of LLM/LMM can be significantly improved.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, et al. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*, 2024. 1

[3] Guangji Bai, Yijiang Li, Chen Ling, Kibaek Kim, and Liang Zhao. SparseLLM: Towards global pruning of pre-trained language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2, 3

[4] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023. 1

[5] Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S Abdelfattah, and Kai-Chiang Wu. Palu: Compressing KV-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*, 2024. 1

[6] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. 1

[7] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014. 2

[8] Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR, 2023. 1, 2, 4

[9] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 1, 4

[10] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993. 1, 2

[11] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023. 1

[12] Injoon Hwang, Haewon Park, Youngwan Lee, Jooyoung Yang, and SunJae Maeng. PC-LoRA: Low-rank adaptation for progressive model compression with knowledge distillation. *arXiv preprint arXiv:2406.09117*, 2024. 1

[13] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. 1

[14] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. GPT-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270): 20230254, 2024. 1

[15] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004. 2

[16] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. MoE-LlaVa: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 1

[17] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang,

Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024. 1, 2

[18] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 4

[20] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 4

[21] Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. 4

[22] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 4

[23] 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. 4

[24] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6655–6659. IEEE, 2013. 2

[25] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020. 1

[26] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023. 1, 2, 4

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[28] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-VLM: Post-training quantization for large vision-language models. *arXiv preprint arXiv:2410.08119*, 2024. 1, 2

[29] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 1

[30] Canwen Xu and Julian McAuley. A survey on model compression and acceleration for pretrained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10566–10575, 2023. 1

[31] Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models. *arXiv preprint arXiv:2406.05223*, 2024. 2

[32] Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023. 1, 2

[33] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 4

[34] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024. 1