

EoRA: Training-free Compensation for Compressed LLM with Eigenspace Low-Rank Approximation

Shih-Yang Liu¹ Maksim Khadkevich Nai Chit Fung¹ Charbel Sakr Chao-Han Huck Yang
Chien-Yi Wang Saurav Muralidharan Hongxu Yin Kwang-Ting Cheng¹ Jan Kautz
Yu-Chiang Frank Wang Pavlo Molchanov Min-Hung Chen

Abstract: In this work, we re-formulate the model compression problem into the *customized compensation* problem: Given a compressed model, we aim to introduce residual low-rank paths to *compensate* for compression errors under customized requirements from users (e.g., tasks, compression ratios), resulting in greater flexibility in balancing accuracy and overhead (inference and model size) without being bound to fixed compression formats. However, naively applying SVD to derive residual paths causes suboptimal utilization of the low-rank representation capacity. Instead, we propose *Training-free Eigenspace Low-Rank Approximation (EoRA)*, a method that directly minimizes compression-induced errors without requiring gradient-based training, achieving fast optimization in minutes using a small amount of calibration data. EoRA projects compression errors into the eigenspace of input activations, leveraging eigenvalues to effectively prioritize the reconstruction of high-importance error components. Moreover, EoRA can be seamlessly integrated with fine-tuning and quantization to further improve effectiveness and efficiency. EoRA consistently outperforms previous methods in compensating errors for compressed LLaMA2/3 models on various tasks, such as language generation, commonsense reasoning, and math reasoning tasks (e.g., **31.31%/12.88%** and **9.69%** improvements on ARC-Easy/ARC-Challenge and MathQA when compensating LLaMA3-8B that is quantized to 4-bit and pruned to 2:4 sparsity). EoRA offers a scalable, training-free solution to compensate for compression errors, making it a powerful tool to deploy LLMs more flexibly. Code is available at <https://github.com/NVlabs/EoRA>.

1. Introduction

Although Large Language Models (LLMs) exhibit superior performance across diverse applications, their empirical deployment remains challenging due to their associated considerable model size and high inference costs. To mitigate these emerging challenges, model compression research such as post-training compression (Ashkboos et al., 2024; Ma et al., 2023) and compression-aware training (Alvarez & Salzmann, 2017; Lym et al., 2019; Liu et al., 2024, 2023c) has been extensively explored to reduce the computational resource demands of serving LLMs (Zhu et al., 2023). However, most existing methods either incur significant accuracy degradation compared to uncompressed models or have high training time. Also, their adaptability is often constrained by a limited range of hardware-supported compression formats (e.g., 2:4 sparsity, 3/4-bit quantization), making it difficult to address various user requirements for accuracy and efficiency. For example, if a user is willing to accept slightly increased inference latency to gain better accuracy, the strict 2:4 sparsity requirement on NVIDIA GPUs or existing integer quantization kernels rules out any intermediate approach, such as 2.X:4 sparsity or INT.X-bit quantization, where X can be any arbitrary integer.

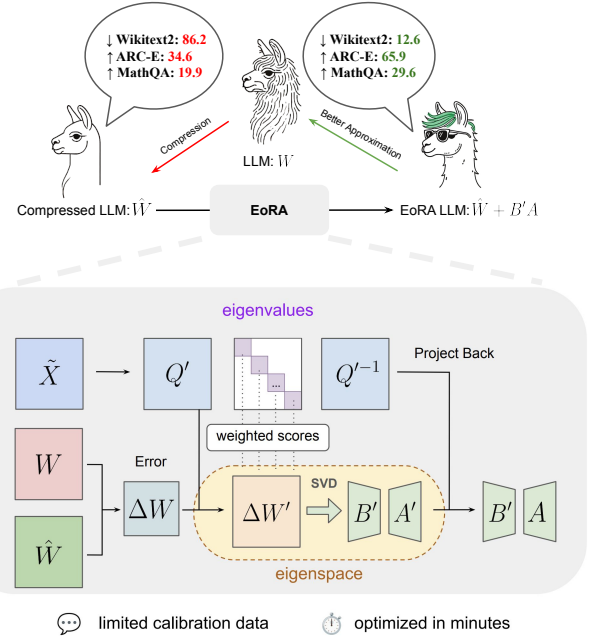


Figure 1 | An overview of our proposed EoRA, which projects the compression error into the eigenspace of input activations and performs low-rank approximation for compensating the compressed model. EoRA of rank 128 can improve the accuracy of the LLaMA3-8B model pruned to 2 : 4 structured sparsity and quantized to 4-bit by 31.3% on ARC-E and 9.7% on MathQA.

trary value.

To bypass such limited flexibility, we reframe model compression as the *customized compensation* problem: introduce residual low-rank paths to *compensate* for compression errors under diverse user needs (e.g., tasks, compression ratios). Instead of focusing solely on minimal performance drop, these paths boost capacity without being constrained by specific compression formats. A straightforward approach is using SVD (Li et al., 2024b; Yao et al., 2024) on compression errors, but this does not factor in the differing influence each weight matrix element has on the model’s final output, resulting in suboptimal utilization of the low-rank representation capacity. Other existing methods either yield limited compensation (by ignoring calibration data) or lose flexibility (via costly compression-aware fine-tuning), making quick task adaptation difficult. This leads to the question: “*How can we efficiently and effectively compensate errors for compressed large language models?*”

To address this research question, We introduce *Training-free Eigenspace Low-Rank Approximation (EoRA)*, which boosts *effectiveness* of compensation while maintaining flexibility and efficiency. By projecting compression errors into the eigenspace of each layer’s input activations, we ensure a direct relationship between the approximation error and the model compression loss. Inspired by the classical Principal Component Analysis (PCA), our method uses each activation channel’s eigenvalues as importance scores for reconstructing weight columns, maximizing low-rank capacity. As a **training-free** optimization method, EoRA requires no gradient computation and can be completed within a few minutes using minimal calibration data. It can also serve as a good starting point for fine-tuning, and remains robust to quantization to further reduce overhead.

We validate the effectiveness of EoRA for compensating the compressed LLMs (LLaMA2-7B/13B and LLaMA3-8B) on language generation, commonsense reasoning, and math tasks, comparing against an SVD-based approach. Our method consistently outperforms SVD, especially for aggressively compressed (including *pruned, quantized, and both*) models (e.g., **3.33%/2.65%** and **3.42%** improvement on ARC-Easy/ARC-Challenge and MathQA when compensating 2:4 pruned LLaMA3-8B). To address memory access overhead from naively running low-rank compensation alongside the quantized weight, we develop a custom kernel that fuses part of the low-rank matrix multiplication with the quantization kernel, yielding up to a 1.3x speedup. Moreover, EoRA remains resilient under 3/4-bit quantization with minimal accuracy drop, underscoring its practicality in compensating for compression errors.

The summary of our contributions is as follows:

- **Flexible Model Compression:** We reframe the conventional compression problem as *customized compensation* problem and propose a novel method, *Training-free Eigenspace Low-Rank Approximation (EoRA)*, a *training-free* approach that compensates compression errors in minutes using minimal calibration data.
- **Eigenspace Projection:** EoRA projects weights into the eigenspace and uses eigenvalues as importance indicators, more effectively leveraging low-rank capacity than naive SVD while directly correlating compensation with layer-wise compression loss.
- **Efficient Inference:** We develop a custom kernel that fuses part of the low-rank matrix multiplication with quantization kernel, accelerating EoRA inference by up to 1.3x. EoRA is also robust to quantization, further minimizing the memory overhead from residual low-rank compensation paths.

2. Preliminaries

Post-training compression aims to compress a well-optimized model by a targeted compression ratio utilizing only a limited set of calibration data. The compression process is often framed as a layer-wise optimization problem, aiming to minimize the layer-wise output difference between the original weight $W_l \in \mathbb{R}^{d \times k}$ and the compressed weight $\hat{W}_l \in \mathbb{R}^{d \times k}$ for each layer l . Then the *layer-wise model compression loss* can be formed as:

$$\arg \min_{\hat{W}_l} \|W_l X_l - \hat{W}_l X_l\|_F \quad (1)$$

where $X_l \in \mathbb{R}^{k \times n}$ is the input activation of layer l and F denotes the Frobenius error between the layer-wise output. Once the compression is complete, the W_l for each layer will be substituted with \hat{W}_l , resulting in a smaller model size, faster inference, or both. However, their flexibility is often limited by a discrete set of compression formats (e.g., 2:4 sparsity, 3/4-bit quantization), making it challenging to meet the diverse capacity and efficiency requirements of different users.

To remove the constraint by specific compression formats, we re-formulate the conventional model compression problem into a *customized compensation* problem: Given a compressed model, we aim to introduce residual low-rank paths to *compensate* for compression errors under customized requirements from users, such as tasks, compression ratios, etc. With these residual paths, the *compensated* model gains greater flexibility in adjusting overall capacity. To derive the low-rank residual paths that can represent compression errors, one naive method is directly adopting Singular Value Decomposition (SVD) (Li et al., 2024b; Yao et al., 2024). More specifically, this method relies on a closed-form solution by using SVD to approximate the compression error $\Delta W_l = W_l - \hat{W}_l$ as $\Delta W_l = U_l \Sigma_l V_l^T$, where $\Sigma_l \in \mathbb{R}^{r \times r}$ is a diagonal matrix containing the top- r largest singular value sorted in descending order, and $U_l \in \mathbb{R}^{d \times r}$, $V_l \in \mathbb{R}^{k \times r}$ are orthonormal matrices, with each column representing the singular vectors corresponding to the singular values in Σ_l . The product of U_l and Σ_l can then be treated as $B_l = U_l \Sigma_l$ with V_l^T being treated as A_l . Overall, the *error approximation loss* can be formulated as:

$$\arg \min_{B_l, A_l} \|\Delta W_l - B_l A_l\|_F \quad (2)$$

and SVD is applied on ΔW_l to minimize the above equation. However, naively applying SVD to optimize error approximation loss (Eq. 2) does not guarantee the minimization of layer-wise compression loss (Eq. 1), and fails to account for the varying importance of individual model weights, resulting in suboptimal utilization of the low-rank representation capacity. In the following sections, we omit the subscript l , which corresponds to layer l for simplicity.

3. Method: EoRA

Compared with standard model compression methods, *model compensation* introduces residual low-rank paths to compensate for compression errors, resulting in greater flexibility in adjusting overall capacity without being constrained by specific compression formats. However, existing methods (Li et al., 2024b; Yao et al., 2024) rely mainly on plain SVD for low-rank approximation, lacking sufficient representation capacity (Barron, 1993) to fully approximate ΔW . In other words, the target rank r remains significantly smaller than the intrinsic rank of ΔW . Therefore, it is necessary to allocate the limited representation capacity of r more effectively, focusing on reconstructing the more important weights while placing less emphasis on less important segments. Moreover, naive SVD performs the approximation in the original space, failing to ensure that minimizing the approximation error (Eq. 2) directly leads to minimizing the layer-wise compression loss (Eq. 1). Furthermore, current approaches (Li et al., 2024b; Yao et al., 2024) either offer *limited* compensation performance by neglecting calibration data or lose flexibility due to the high computational cost of compression-aware fine-tuning, making it difficult to swiftly adjust to various tasks. This phenomenon raises an emerging question: “How can we efficiently and effectively compensate for errors in compressed LLMs?”

To address this question, we propose *Training-free Eigenspace Low-Rank Approximation (EoRA)*, which retains the flexibility advantages of model compensation while enhancing both *efficiency* and *effectiveness* compared to existing approaches. First, we propose projecting the compression error into the eigenspace (Stewart, 2001) of the corresponding layer’s input activations, ensuring a direct relationship between the error approximation loss and the overall layer-wise model compression loss. Inspired by the classical Principal Component Analysis (PCA) algorithm, we leverage the eigenvalues of each activation channel as importance scores to indicate the importance of each column after the eigenprojection. This allows us to allocate more low-rank representation capacity to approximate the more critical error elements. Following PCA, we perform the eigendecomposition on $\tilde{X} \tilde{X}^T$ where $\tilde{X} \in \mathbb{R}^{k \times n}$ is the average of the input activations over the calibration set. The decomposition $\tilde{X} \tilde{X}^T = Q \Lambda Q^T$ is then used to derive the eigenspace projection matrix $Q \in \mathbb{R}^{k \times k}$ whose columns are the eigenvectors and $\Lambda \in \mathbb{R}^{k \times k}$ which is a diagonal matrix with each diagonal element being the corresponding eigenvalues of the eigenvectors in Q . We then propose to project the compression error ΔW into eigenspace with the projection matrix $Q' = Q \sqrt{\Lambda}$ to obtain the projected error $\Delta W' \in \mathbb{R}^{d \times k} = \Delta W Q'$. The proposed new error approximation loss, *EoRA loss*, can be formulated as:

$$\arg \min_{B', A'} \|\Delta W' - B' A'\|_F \quad (3)$$

where SVD is applied on $\Delta W'$ to minimize the above equation and B' and A' denote the corresponding solutions in the eigenspace. This loss function ensures that error columns associated with larger eigenvalues

are approximated more accurately than those with smaller eigenvalues, thereby facilitating a more effective allocation of the insufficient low-rank expressive power. Since Q is an orthogonal matrix, we can multiply the low-rank approximated $\Delta W'$ with $Q'^{-1} = \sqrt{\Lambda}^{-1} Q^T$ to project back to the original space after the layer-wise reconstruction, obtaining the reconstructed error $\Delta W = \Delta W' Q'^{-1}$ approximated by $B' A' Q'^{-1}$. The product of A' and Q'^{-1} can be consolidated into a single matrix with the same dimensions as the original A' , ensuring no additional inference latency as $A = A' Q'^{-1}$. Then, the forward pass of the compressed model compensated with EoRA for the input activation X can be formulated as:

$$\hat{W}X + B'AX \quad (4)$$

The overall **training-free** optimization of Eq. 3 in EoRA can be done in minutes using only a small amount of calibration data *without* any gradient computation. EoRA can also provide better initialization for fine-tuning to further enhance accuracy and offer a trade-off between accuracy and training time. Moreover, EoRA is robust to quantization which can further reduce the additional cost of residual low-rank compensation paths. Please refer to Sec. 4.5 and 4.7 for more details. The overall eigenspace projection method is depicted in Figure 1 with the detailed algorithm in Alg. 1.

Algorithm 1 Eigenspace low-rank approximation (EoRA)

Input: \tilde{X} : Average of the input activations of the current layer over the calibration set, W : Full-precision Weight, \hat{W} : Compressed Weight, r : Compensation rank
Output: B', A : Two low-rank matrices for compensation.
 1. $\Delta W = W - \hat{W}$
 2. Run Eigendecomposition on $\tilde{X}\tilde{X}^T = Q\Lambda Q^T$
 3. Reformulate $Q\Lambda Q^T = (Q\sqrt{\Lambda})(\sqrt{\Lambda}Q^T) = Q'Q'^T$
 4. Project the compression error to eigenspace $\Delta W' = \Delta W Q'$
 5. Run r -rank SVD approximation on $\Delta W'$, $B' A' = U'\Sigma'V'^T = \text{SVD}(\Delta W')$
 6. Project the approximation back to the original space $A = A' Q'^{-1}$
 7. The final forward pass of current layer becomes $\hat{W}X + B'AX$

Mapping EoRA loss (Eq. 3) to compression loss (Eq. 1): The goal of low-rank compensation is to approximate ΔW such that the approximation also minimize Eq.1. To achieve this, we reformulate the compression objective for each layer as:

$$\arg \min_{B,A} \|WX - (\hat{W} + BA)X\|_F = \arg \min_{B,A} \|\Delta WX - BAX\|_F \quad (5)$$

Since the Frobenius norm of a matrix is equal to the square root of its gram matrix (Sun, 1991; Wang et al., 2024), the minimization problem can be rewritten as:

$$\begin{aligned} & \arg \min_{B,A} \|\Delta WX - BAX\|_F \\ &= \arg \min_{B,A} [\text{trace}((\Delta W - BA)XX^T(\Delta W - BA)^T)]^{\frac{1}{2}} \end{aligned} \quad (6)$$

Directly applying SVD on ΔW initially does not guarantee the minimization of the above equation Eq.6, as dropping the smallest singular values does not necessarily lead to the smallest layer-wise compression error (Eq.6) compared to discarding other singular values. To address this issue, EoRA projects ΔW into the eigenspace before performing SVD. In the following, we demonstrate that minimizing Eq.3 with SVD is the same as minimizing Eq.6.

Theorem. For an activation matrix X , whose matrix product XX^T has an eigendecomposition given by $XX^T = Q\Lambda Q^T$. By projecting the compression error ΔW into the eigenspace with $Q\sqrt{\Lambda}$ as $\Delta W' = \Delta W Q\sqrt{\Lambda}$, minimizing Eq.3 via SVD becomes equivalent to minimizing Eq.6.

Proof. First, note that $XX^T = Q\Lambda Q^T$, and by substituting this into Eq.6, we get

$$\begin{aligned} & [\text{trace}((\Delta W - BA)Q\Lambda Q^T(\Delta W - BA)^T)]^{\frac{1}{2}} \\ &= [\text{trace}((\Delta W Q - BAQ)\Lambda(\Delta W Q - BAQ)^T)]^{\frac{1}{2}} \end{aligned} \quad (7)$$

Since $\Lambda = \sqrt{\Lambda}\sqrt{\Lambda}$ and $\sqrt{\Lambda} = \sqrt{\Lambda}^T$, the above Eq. 7 can further be rewritten as:

$$[\text{trace}((\Delta W Q \sqrt{\Lambda} - B A Q \sqrt{\Lambda})(\Delta W Q \sqrt{\Lambda} - B A Q \sqrt{\Lambda})^T)]^{\frac{1}{2}} \quad (8)$$

Let $Q' = Q\sqrt{\Lambda}$, then Eq.8 becomes:

$$\begin{aligned} & [\text{trace}((\Delta W Q' - B A Q')(\Delta W Q' - B A Q')^T)]^{\frac{1}{2}} \\ &= [\text{trace}((\Delta W' - B A Q')(\Delta W' - B A Q')^T)]^{\frac{1}{2}} \\ &= \|\Delta W' - B A Q'\|_F \end{aligned} \quad (9)$$

where the square root of the gram matrix can be transformed back to the corresponding Frobenius norm according to (Sun, 1991). Let us set $B A Q' = B' A'$, then $\|\Delta W' - B A Q'\|_F$ becomes $\|\Delta W' - B' A'\|_F$. By the Eckart–Young theorem (Eckart & Young, 1936), the minimization of this Frobenius norm is achieved by running SVD on $\Delta W'$, therefore we prove that minimizing $\|\Delta W' - B' A'\|_F$ via SVD is equivalent to minimizing Eq.6, where low-rank approximation of $\Delta W'$ is $\text{SVD}(\Delta W') = B' A'$.

4. EXPERIMENTS

4.1. Experiments Details

We implement EoRA in PyTorch (Paszke et al., 2017), utilizing the Hugging Face Transformers and Datasets framework (Wolf et al., 2019). All experiments are conducted on a single NVIDIA H100 GPU. We primarily focus on evaluating EoRA for compensating LLaMA2-7B/13B and LLaMA3-8B models, compressed using SparseGPT (Frantar & Alistarh, 2023), a widely adopted pruning method, and GPTQ (Frantar et al., 2023) for quantization. Channel-wise asymmetric quantization is applied across all experiments, and we follow the settings from (Huang et al., 2024a) to construct the calibration dataset for both SparseGPT and GPTQ. In Section 4.4, we also present additional experiments using EoRA to compensate LLaMA3-8B compressed by alternative methods, such as AWQ (Lin et al., 2024) and Wanda (Sun et al., 2024).

We compare EoRA with the plain SVD low-rank compensation method and evaluate the compressed models on language generation, commonsense reasoning, and math reasoning tasks using the LM-Evaluation-Harness framework (Gao et al., 2024). For the following context, we refer to the standard SVD low-rank compensation method as SVD. We pick WikiText2 for the language generation task and perplexity as the evaluation metric. For commonsense reasoning, we select ARC-Easy and ARC-Challenge (ARC-E and ARC-C) (Clark et al., 2018), and for math reasoning ability, we choose MathQA (Amini et al., 2019). We sample 256 concatenated sentences of length 2048 from the WikiText2 training set as the calibration set for EoRA for the language generation task. For commonsense reasoning tasks, we sample 32 concatenated sentences of length 2048 from the ARC training set and combined them with 32 concatenated sentences of the same length from C4 (Raffel et al., 2020) to construct the calibration set for EoRA. Similarly, for the math reasoning task, we sample 32 concatenated sentences of length 2048 from the MathQA training set and combined them with 32 concatenated sentences from C4 to form the calibration set for EoRA. The low-rank compensation process of EoRA is entirely training-free, requiring no backpropagation. It is conducted layer-by-layer and can be completed within just a few minutes.

4.2. Main Results

4.2.1. Sparsity Error Compensation

To assess the effectiveness of EoRA in compensating for sparsity error, we compare EoRA with SVD on LLaMA2-7B/13B and LLaMA3-8B models pruned with SparseGPT to {50%, 60%, 2:4} sparsity levels. Both the ranks of EoRA and SVD are set to 128, and the results of LLaMA3-8B and LLaMA2-13B are summarized in Table 1, while the full results, including LLaMA2-7B, are provided in Table 8 in the appendix. We observe that structural pruning results in more significant accuracy degradation compared to unstructured pruning. However, EoRA consistently outperforms SVD in compensating for both types of pruning, showing improvements of 1.98%/2.22%/1.94% and 3.33%/2.65%/3.42% on the ARC and MathQA tasks for LLaMA3-8B models with 50% and 2:4 sparsity, respectively. Notably, the performance gain of EoRA over SVD is more pronounced in more challenging sparsity settings. For instance, EoRA surpasses SVD by 0.22/4.13%/4.01%/3.22% across the four tasks when compensating for LLaMA3-8B at 60% sparsity, which is a larger improvement compared to the 50% sparsity scenario. Furthermore, EoRA proves robustness across different model sizes, continuing to outperform SVD in compensating for various sparsity configurations of LLaMA2-13B.

Table 1 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 pruned by SparseGPT with different sparsity, with compensation via SVD/EoRA of rank 128.

Model	Sparsity	Compensation Method	Wikitext2 ↓	ARC-E ↑	ARC-C ↑	MathQA ↑
LLaMA3-8B	Uncompressed	-	6.13	80.09	50.42	40.10
		-	8.25	72.13	39.84	32.69
	50%	SVD	7.99	73.90	41.38	32.96
		EoRA	7.98 (-0.01)	75.88 (+1.98)	43.60 (+2.22)	34.90 (+1.94)
	60%	-	12.00	63.38	30.54	27.00
		SVD	10.93	64.64	30.97	28.40
		EoRA	10.71 (-0.22)	68.77 (+4.13)	34.98 (+4.01)	31.62 (+3.22)
	2:4	-	12.32	62.75	30.11	26.43
		SVD	11.31	64.89	31.99	26.49
		EoRA	11.07 (-0.24)	68.22 (+3.33)	34.64 (+2.65)	29.91 (+3.42)
LLaMA2-13B	Uncompressed	-	4.88	73.23	45.56	29.91
		-	5.65	68.81	39.24	27.30
	50%	SVD	5.54	69.69	39.59	27.63
		EoRA	5.54	71.63 (+1.94)	41.97 (+2.38)	28.27 (+0.64)
	60%	-	6.93	63.21	33.70	26.86
		SVD	6.59	65.44	34.12	26.06
		EoRA	6.52 (-0.07)	67.25 (+1.81)	37.71 (+3.59)	27.16 (+1.10)
	2:4	-	7.10	66.32	34.30	25.92
		SVD	6.82	66.28	33.61	25.12
		EoRA	6.75 (-0.07)	68.47 (+2.19)	37.54 (+3.93)	27.53 (+2.41)

4.2.2. Quantization Error Compensation

We compare EoRA with SVD on LLaMA2-7B/13B and LLaMA3-8B models quantized with GPTQ to 4-bit and 3-bit to assess the effectiveness of EoRA in compensating for quantization error. The ranks for EoRA and SVD are set to 128. From Table 2, 3-bit quantization causes significant accuracy degradation, particularly for LLaMA3-8B, with losses of up to 43.31%/29.52%/17.73% on ARC-E, ARC-C, and MathQA, respectively. By applying EoRA, we demonstrate that the accuracy loss can be reduced to 19.95%/18.68%/10.99% on ARC-E, ARC-C, and MathQA, respectively—providing an improvement of 2.95%/1.72%/2.68% compared to using SVD for compensating the quantization error. On the other hand, although 4-bit quantization does not result in as much accuracy loss as 3-bit quantization, applying EoRA can still generally enhance the performance of the 4-bit model, offering up to a 2.2% and 3.14% accuracy boost on ARC-C and MathQA, respectively, for the 4-bit LLaMA3-8B model. The complete results which include LLaMA2-7B are provided in Table 9 in the appendix.

4.2.3. Sparsity & Quantization Error Compensation

Next, we examine the feasibility of applying EoRA to compensate for ultra-compressed models that undergo both pruning and quantization. Specifically, we prune LLaMA2-7B/13B and LLaMA3-8B to 2:4 sparsity and quantize them to 4-bit. We set the ranks of both EoRA and SVD to 128 to compensate for the pruning and quantization errors. The results for LLaMA3-8B and LLaMA2-13B are shown in Table 3, while the full results, which also include LLaMA2-7B, are available in Table 10 in the appendix. Similarly to our previous findings, LLaMA3-8B is the least resilient to compression, experiencing a significant drop in both perplexity for language generation and accuracy on commonsense and math reasoning tasks. Notably, the accuracy on ARC-C plummets to 18.33% and MathQA to 19.89%, which is worse than random guessing. However, compensating for the sparsity and quantization errors with EoRA significantly improves the accuracy of these compressed models, reducing perplexity by up to 73.55 and boosting accuracy by 31.31%/12.88%/9.60% on ARC and MathQA tasks. Additionally, EoRA consistently outperforms SVD across LLaMA2 and LLaMA3. For instance, EoRA exceeds SVD in compensating the compressed LLaMA2-13B on ARC-C by 1.79% and on MathQA by 1.77%, narrowing the accuracy gap with the uncompressed model to just 2.85% on MathQA. Overall, we find that EoRA tends to offer greater accuracy recovery when addressing more aggressive compression settings, ensuring the plausibility of adopting EoRA for mitigating severe compression error.

Table 2 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 quantized by GPTQ with different bit-width, with compensation via SVD/EoRA of rank 128.

Model	W-bit	Compensation Method	Wikitext2 ↓	ARC-E ↑	ARC-C ↑	MathQA ↑
LLaMA3-8B	Uncompressed	-	6.13	80.09	50.42	40.10
	W4	-	7.00	78.11	45.90	34.07
		SVD	6.80	77.48	45.24	36.51
		EoRA	6.80	78.07 (+0.59)	47.44 (+2.20)	37.21 (+0.7)
	W3	-	15.64	36.78	20.90	22.37
		SVD	10.24	57.19	30.02	26.43
		EoRA	10.06 (-0.18)	60.14 (+2.95)	31.74 (+1.72)	29.11 (+2.68)
	Uncompressed	-	4.88	73.23	45.56	29.91
LLaMA2-13B	W4	-	5.06	71.33	44.28	29.10
		SVD	5.03	71.88	44.19	28.97
		EoRA	5.03	71.80	44.53 (+0.34)	28.90
		-	5.99	63.04	37.28	26.26
	W3	SVD	5.76	64.64	37.54	26.83
		EoRA	5.75 (-0.01)	65.86 (+1.22)	39.50 (+1.96)	27.20 (+0.37)
		-	5.99	63.04	37.28	26.26
	Uncompressed	-	4.88	73.23	45.56	29.91

Table 3 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 models pruned to 2:4 using SparseGPT and quantized to 4-bit with GPTQ, with compensation via SVD/EoRA of rank 128.

Model	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
LLaMA3-8B	Uncompressed	6.13	80.09	50.42	40.10
	-	86.15	34.59	18.34	19.89
	SVD	12.84	62.12	29.35	26.86
	EoRA	12.60	65.9	31.22	29.58
LLaMA2-13B	Uncompressed	4.88	73.23	45.56	29.91
	-	7.27	64.09	33.10	24.75
	SVD	6.98	66.41	33.27	25.29
	EoRA	6.89	66.58	35.06	27.06

4.3. Compensation with Different Ranks

Since one of the advantages of using low-rank compensation for compression error is the greater flexibility in adjusting overall model capacity without being constrained by specific compression formats, in this section, we investigate the influence of different ranks on adopting EoRA. We vary the rank in $\{64, 128, 256, 512\}$ and compare it with SVD on compensating LLaMA3-8B pruned to 2:4 sparsity. As shown in Table 4, EoRA consistently outperforms SVD across different ranks, with the improvement becoming slightly more pronounced at higher ranks, particularly on Wikitext2. For example, the perplexity improvement is 0.34 at rank 512, compared to 0.09 at rank 64. The improvement across different ranks on commonsense and math reasoning tasks remains relatively steady, around 2%. The experiments prove that EoRA is robust across different rank settings, offering users a more flexible option upon existing compression configurations to effectively balance the trade-off between inference overhead and model accuracy. A similar trend is observed in the results for LLaMA2-7B/13B shown in Table 11 in the appendix.

4.4. Compatibility With Various Compression Methods

This section studies the generalizability and compatibility of EoRA with different quantization and pruning methods beyond SparseGPT and GPTQ. For quantization, we consider AWQ (Lin et al., 2024), which utilizes activation statistics to enhance salient channels and reduce quantization errors. For pruning, we adopt Wanda (Sun et al., 2024), a method that prunes weights with the smallest magnitudes scaled by their corresponding input activations. For these compression methods, we adhere to the calibration set construction detailed in 4.1, and maintain the same settings when utilizing EoRA to address compression errors. We compare EoRA against SVD on LLaMA3-8B quantized with AWQ to 4-bit and 3-bit, as well as LLaMA3-8B pruned with Wanda to 50%/60% and 2:4 structured sparsity. The ranks of EoRA and SVD are set to 128. Table 5 demonstrates that EoRA consistently outperforms SVD in improving accuracy across various compression settings and methods. Notably, the benefits of EoRA become even more evident

Table 4 | Comparison between SVD and EoRA of different ranks on compensating LLaMA3-8B models pruned to 2:4 sparsity by SparseGPT on Perplexity and Commonsense/Math reasoning tasks.

r	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
-	Uncompressed	6.13	80.09	50.42	40.10
-	-	12.32	62.75	30.11	26.43
64	SVD	11.76	62.83	30.97	26.39
	EoRA	11.67	65.86	33.1	28.57
128	SVD	11.31	64.89	31.99	26.49
	EoRA	11.07	68.22	34.64	29.91
256	SVD	10.54	68.01	34.55	28.74
	EoRA	10.25	71.00	37.96	31.59
512	SVD	9.38	71.46	38.73	30.38
	EoRA	9.04	74.49	41.89	34.17

Table 5 | Comparison between SVD and EoRA on compensating LLaMA3-8B models compressed with AWQ to 4/3-bit and Wanda to { 50%, 60%, 2:4 } on Perplexity and Commonsense/Math reasoning tasks.

Compression	Config	Compensation	Wikitext2	ARC-E	ARC-C	MathQA
Uncompressed	-	-	6.13	80.09	50.42	40.10
AWQ	W4	-	7.19	76.85	45.90	36.95
		SVD	6.52	79.88	50.42	39.36
		EoRA	6.42	80.55	50.59	39.36
	W3	-	10.28	64.30	35.49	28.50
		SVD	8.22	72.89	42.41	33.86
		EoRA	7.44	74.62	43.00	35.67
Wanda	50%	-	9.38	71.25	41.29	31.52
		SVD	8.84	72.13	42.06	32.52
		EoRA	8.38	75.88	43.77	34.77
	60%	-	19.97	61.36	28.32	25.15
		SVD	15.59	66.41	32.50	27.20
		EoRA	13.46	69.10	35.49	29.74
	2:4	-	21.42	55.42	27.04	25.09
		SVD	17.16	60.64	30.46	26.16
		EoRA	14.04	65.44	34.81	30.05

under more challenging conditions, a trend also observed when compensating for models compressed with SparseGPT and GPTQ. For example, when compensating a 3-bit AWQ-quantized model, EoRA achieves accuracy gains of 1.73% and 0.59% on ARC-E and ARC-C, respectively, which are higher than the improvements of 0.67% and 0.17% observed when compensating a 4-bit quantized model. Furthermore, when applied to a 4-bit quantized LLaMA3-8B model, EoRA fully recovers the accuracy loss on ARC-E and ARC-C, even surpassing the full-precision model by 0.46% and 0.17%. These findings underscore the effectiveness and generalizability of EoRA across different compression techniques.

4.5. Fine-tuning Compressed Models with EoRA

In this section, we show that users can fine-tune EoRA to further recover the accuracy loss of the compressed models. We follow the conventional LoRA fine-tuning framework, which keeps the compressed model frozen and only tunes the low-rank residual components during fine-tuning. We conduct experiments on compressed LLaMA3-8B models

Table 6 | Fine-tune the compressed LLaMA3-8B models of various compression settings using different initialization of the low-rank matrices for Commonsense/Math reasoning tasks.

Compression	Config	Initialization	ARC-E	ARC-C	MathQA
Uncompressed	-	w/o finetuning	80.09	50.42	40.10
		Standard	84.55	56.39	53.56
SparseGPT	2:4	w/o finetuning	62.75	30.11	26.43
		Standard	73.82	41.30	45.42
		SVD	74.45	43.68	48.77
		EoRA	76.01	48.54	54.67
		w/o finetuning	78.11	45.90	34.07
GPTQ	W4	Standard	81.69	54.09	51.42
		SVD	82.49	54.52	53.96
		EoRA	83.04	55.46	56.04
		w/o finetuning	36.78	20.90	22.37
	W3	Standard	57.87	30.29	34.10
		SVD	75.54	44.70	48.17
		EoRA	76.93	47.44	53.90
		w/o finetuning	36.78	20.90	22.37

with {2:4 sparsity, 4-bit, 3-bit} compression. The rank of LoRA is set to 128 and is applied to every linear layer, initialized using EoRA, SVD, and standard Kaiming initialization. Fine-tuning is performed on the ARC training set for evaluating ARC-C and ARC-E, and on the MathQA training set for math reasoning tasks. We fine-tune the models for 3 epochs with a batch size of 64, a learning rate of $1e-5$, and a cosine learning rate scheduler. As shown in Table 6, using EoRA for initialization significantly improves the accuracy of compressed models, reducing the accuracy gap between the full-precision model and the 2:4 sparsity model from 17.34%/20.31% before fine-tuning to just 4.08%/1.88% after fine-tuning on ARC-E and ARC-C. Additionally, EoRA consistently surpasses both standard and SVD initialization by a significant margin across various compression settings, with accuracy improvements of 1.56%/4.86%/5.9% and 1.39%/2.74%/5.73% over SVD when fine-tuning 2:4 sparsity and 3-bit LLaMA3-8B models, respectively. Furthermore, fine-tuning a 4-bit model with EoRA as LoRA initialization can even surpass the accuracy of the original full-precision model, with improvements of 2.95%/5.04% on ARC-E, ARC-C, and the accuracy of the full-precision fine-tuned model on MathQA with 2.48% improvement. We conduct an ablation study to examine the impact of using different proportions of the training dataset for fine-tuning 2:4 pruned LLaMA3-8B with various low-rank matrix initialization methods. The results demonstrate that EoRA is more robust to variations in dataset sizes and consistently outperforms standard LoRA and SVD initialization. A detailed discussion is provided in Appendix A.5.

4.6. Kernel Optimization and Inference Speed Evaluation

While theoretically, compensating a compressed model with low-rank residual paths introduces minimal computational overhead, in practice, it leads to a noticeable increase in latency. This is primarily because input and output must transfer between L2 cache and DRAM twice as often compared to that without a low-rank residual path, shifting the inference process from being computation-bound to memory-bound. This phenomenon is also discussed in (Li et al., 2024a). To address this, we propose fusing the low-bit weight quantization kernel with the matrix multiplication of B , which shares the same output. By doing so, the shared output no longer needs to be offloaded and reloaded to the L2 cache, effectively reducing data transfer overhead as illustrated in Figure 2.

In language generation, the model produces tokens sequentially, making matrix-vector multiplications the primary factor impacting the inference latency. Consequently, we build our custom EoRA kernel on top of GPTQ’s low-bit quantized matrix vector product kernel, pre-allocating the shared output prior to matrix vector multiplication and integrating the full-precision matrix vector multiplication of B into the quantized kernel reducing redundant memory access. As shown in Table 7, our custom EoRA kernel substantially accelerates inference compared to using native PyTorch for the low-rank residual path on top of the low-bit quantized kernel, achieving up to a speedup of up to 1.4x over FP16 with EoRA of rank 128 at 3-bit quantization. In contrast, without the EoRA kernel, the initial 1.7x speedup provided by the 3-bit quantized kernel drops to 1.1x. Similarly, under 4-bit quantization, the EoRA kernel

delivers an extra 0.3x speedup compared to setups without the EoRA kernel.

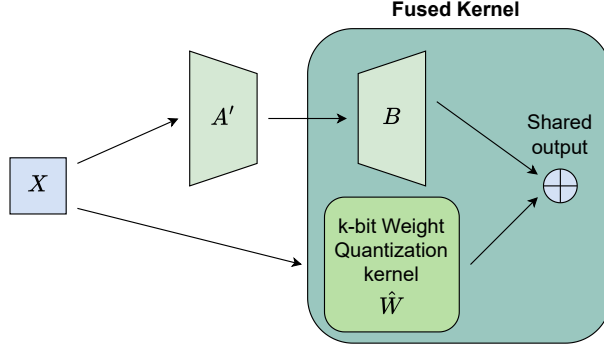


Figure 2 | We propose fusing the multiplication of B with the weight quantization kernel to minimize data movement overhead and substantially improve the inference latency.

Table 7 | Comparison of the average per-token latency (batch size 1) for 128-token generation on LLaMA3-70B between full-precision and GPTQ + EoRA (rank=128) with and without our custom EoRA kernel.

Format	EoRA Kernel	Latency	Speedup
FP-16	-	60ms	1x
3-bit	-	35ms	1.7x
	No	54ms	1.1x
3-bit	Yes	43ms	1.4x
	-	38ms	1.6x
4-bit	No	61ms	0.9x
	Yes	51ms	1.2x

4.7. Quantization of EoRA

Finally, EoRA can also be quantized to further reduce the additional cost of residual low-rank compensation paths. In this section, we quantize EoRA of rank $\{128, 512\}$ to 4/3-bit on compensating three types of compressed LLaMA3-8B models (2:4 pruned, 4-bit quantized, and 3-bit quantized). The complete results are provided in Table 13, while the results for LLaMA3-8B are illustrated in Figure 3. As shown in the figure, EoRA is robust to quantization, which means that when EoRA is quantized, the accuracy drop from full-precision EoRA is insignificant while the model size is significantly reduced. For example, when a 512-rank EoRA is quantized from 16-bits to 4-bit on 2:4 pruned LLaMA3-8B, the accuracy drops are only 0.43% on ARC-C while the total model size reduces by 16.49%. Additionally, compared to the original uncompensated 2:4 pruned model, quantizing EoRA of rank 128/512 improves accuracy by 4.4%/11.4% with a total model size increase of just 2%/7%. For 3-bit quantized LLaMA3-8B compensated with a 4-bit quantized EoRA of rank 128/512 achieves 10.6%/19.1% accuracy improvements, with a corresponding model size increase of only 3%/14%. Interestingly, we also observe that quantizing EoRA does not always result in accuracy loss; in some cases, it even slightly improves accuracy, potentially due to quantization acting as a form of regularization, as discussed in Liu et al. (2023a). Generally, we recommend users quantize EoRA to 4-bit, as this significantly reduces inference latency and model size with kernel support, without causing any noticeable drop in accuracy.

5. RELATED WORK

LLMs Compression: With the rapid expansion of LLMs in various applications, it is crucial to compress the model size to lower the computational costs for deployment. However, traditional compression-aware training methods (Liu et al., 2023b,a) are no longer practical for LLMs, as these techniques demand access to the original training datasets and significant computational resources for model retraining. To overcome these challenges, many post-training

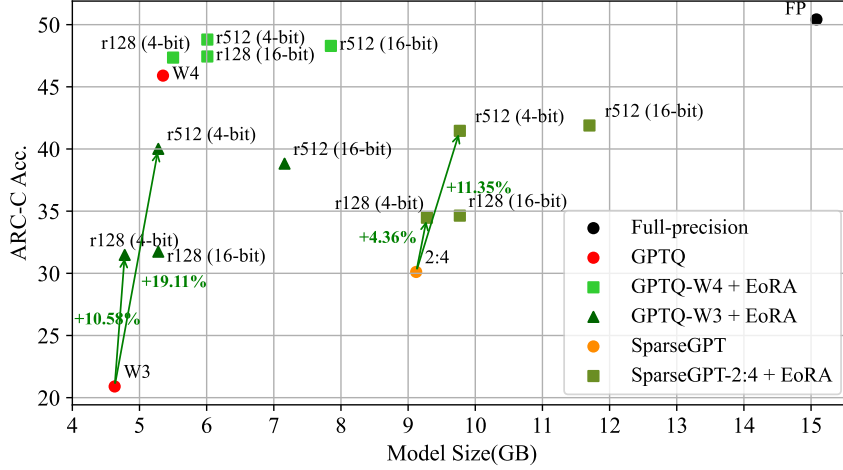


Figure 3 | The model size and ARC-C accuracy of EoRA with rank 128/512, quantized to 4-bit for compensating LLaMA3-8B quantized to 4/3-bit or pruned to 2:4 sparsity.

compression methods (Frantar et al., 2023; Frantar & Alistarh, 2023; Tseng et al., 2024; Sun et al., 2024; Wang et al., 2024) have been developed that do not require model retraining and only need a small subset of the dataset for calibration. Among these methods, Post-training Quantization (PTQ) (Frantar et al., 2023; Tseng et al., 2024) is one of the most commonly applied techniques. It reduces the model size by replacing higher bitwidth representations with lower bitwidth ones. Another popular approach is Post-training Pruning (PTP), which minimizes computation by setting the least important weight elements to zero, as demonstrated in (Frantar & Alistarh, 2023; Sun et al., 2024). Recently, a different approach to compression has been explored in studies like (Yuan et al., 2023; Wang et al., 2024), where the model’s weights are replaced with low-rank matrices. Similarly, ESPACE (Sakr & Khailany, 2024) employs activation projections to achieve dimensionality reduction in GEMM layers. These methods can reduce both inference latency and model size without the need for custom kernel support. Since our proposed EoRA is compression-agnostic, it remains compatible with all of these compression techniques.

Compression-aware Low-rank Adaptation: (Dettmers et al., 2023) proposes combining a low-rank adaptation (LoRA) parameter-efficient fine-tuning method with quantized models to further reduce training costs. Building upon this idea, LoftQ (Li et al., 2024b) suggests accounting for compression error by initializing LoRA with the SVD approximation of it, thereby enhancing fine-tuning accuracy. Another line of research, such as (Huang et al., 2024b), explores reducing quantization difficulty during fine-tuning by incorporating rotations with LoRA, while (Xu et al., 2024) introduces a new group-wise adaptation technique to increase the degrees of freedom in quantization. However, all these methods primarily focus on improving fine-tuning accuracy and are mostly compatible only with quantization. In contrast, EoRA aims to enhance the compressed model without fine-tuning and is agnostic to the used compression method. We also demonstrate that EoRA can serve as the initialization for LoRA—similar to LoftQ—in downstream fine-tuning tasks, consistently outperforming the naive SVD initialization method.

6. CONCLUSION

In this work, we propose **EoRA** (Training-free **E**igenspace **L**ow-**R**ank **A**pproximation), a novel method to efficiently and effectively compensate for compression errors in large language models. By projecting compression-induced errors into the eigenspace of model activations, EoRA leverages eigenvalues as importance indicators, enabling optimal utilization of low-rank capacity without requiring gradient-based training. Our approach demonstrates significant improvements in language generation, commonsense reasoning, and mathematical reasoning tasks, outperforming traditional low-rank approximation techniques such as SVD. The key strength of EoRA lies in its *training-free* nature, allowing for rapid optimization using only a small calibration dataset, and its robustness to quantization, making it an effective tool for deploying large models with varying capacity requirements. Moreover, EoRA provides a solid initialization for fine-tuning, further reducing accuracy degradation and, in some cases, surpassing the performance of *uncompressed models*. Overall, EoRA presents a scalable, versatile solution for model compensation, with potential applications across various domains where efficient deployment of large models is crucial. Future work may explore extending EoRA to more complex model architectures and compression scenarios, further enhancing its adaptability and effectiveness.

7. ACKNOWLEDGMENTS

We extend our gratitude to Xing Li and other members from ModelCloud.AI for their assistance in integrating EoRA into the [GPTQModel](#) package, thus making our work more accessible to the broader public.

References

- Jose M Alvarez and Mathieu Salzmann. Compression-aware training of deep networks. In *Neural Information Processing Systems*, 2017.
- Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *North American Chapter of the Association for Computational Linguistics*, 2019.
- Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefer, and James Hensman. Slicept: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations*, 2024.
- Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 1993.
- 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. *arXiv preprint arXiv:1803.05457*, 2018.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Neural Information Processing Systems*, 2023.
- Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.
- Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, 2023.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 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, 2024. URL <https://zenodo.org/records/12608602>.
- Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. How good are low-bit quantized llama3 models? an empirical study. *arXiv preprint arXiv:2404.14047*, 2024a.
- Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. Rolora: Fine-tuning rotated outlier-free llms for effective weight-activation quantization. *arXiv preprint arXiv:2407.08044*, 2024b.
- Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdqunat: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024a.
- Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. In *International Conference on Learning Representations*, 2024b.
- 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. In *Machine Learning and Systems*, 2024.
- Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *International Conference on Machine Learning*, pp. 21813–21824. PMLR, 2023a.
- Zechun Liu, Barlas Oguz, Aasish Pappu, Yangyang Shi, and Raghuraman Krishnamoorthi. Binary and ternary natural language generation. In *Association for Computational Linguistics*, 2023b.
- Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023c.

- Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinqant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024.
- Sangkug Lym, Esha Choukse, Siavash Zangeneh, Wei Wen, Sujay Sanghavi, and Mattan Erez. Prunetrain: fast neural network training by dynamic sparse model reconfiguration. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Neural Information Processing Systems*, 2023.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Neural Information Processing Systems Autodiff Workshop*, 2017.
- 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*, 2020.
- Charbel Sakr and Brucek Khailany. Espace: Dimensionality reduction of activations for model compression. *arXiv preprint arXiv:2410.05437*, 2024.
- Gilbert W Stewart. *Matrix Algorithms: Volume II: Eigensystems*. SIAM, 2001.
- Ji-Guang Sun. Perturbation bounds for the cholesky and qr factorizations. *BIT Numerical Mathematics*, 1991.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024.
- Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning*, 2024.
- Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.
- Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2024.
- Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Exploring post-training quantization in llms from comprehensive study to low rank compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.
- 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.
- Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*, 2023.

A. Appendix

A.1. Sparsity Error Compensation

Table 8 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 pruned by SparseGPT with different sparsity, with compensation via SVD/EoRA of rank 128.

Model	Sparsity	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
LLaMA3-8B	Uncompressed	-	6.13	80.09	50.42	40.10
		-	8.25	72.13	39.84	32.69
	50%	SVD	7.99	73.90	41.38	32.96
		EoRA	7.98 (-0.01)	75.88 (+1.98)	43.60 (+2.22)	34.90 (+1.94)
	60%	-	12.00	63.38	30.54	27.00
		SVD	10.93	64.64	30.97	28.40
		EoRA	10.71 (-0.22)	68.77 (+4.13)	34.98 (+4.01)	31.62 (+3.22)
	2:4	-	12.32	62.75	30.11	26.43
		SVD	11.31	64.89	31.99	26.49
		EoRA	11.07 (-0.24)	68.22 (+3.33)	34.64 (+2.65)	29.91 (+3.42)
LLaMA2-7B	Uncompressed	-	5.47	69.31	39.84	27.67
		-	6.48	64.14	35.92	26.90
	50%	SVD	6.34	63.51	36.26	26.39
		EoRA	6.31 (-0.03)	66.45 (+2.94)	38.22 (+1.96)	27.10 (+0.71)
	60%	-	8.35	59.72	30.11	25.15
		SVD	7.81	61.61	32.42	25.09
		EoRA	7.69 (-0.12)	62.66 (+1.05)	34.12 (+1.70)	25.99 (+0.9)
	2:4	-	8.77	60.47	30.11	24.65
		SVD	8.15	60.98	30.54	24.89
		EoRA	7.97 (-0.18)	63.42 (+2.44)	32.67 (+2.13)	25.59 (+0.70)
LLaMA2-13B	Uncompressed	-	4.88	73.23	45.56	29.91
		-	5.65	68.81	39.24	27.30
	50%	SVD	5.54	69.69	39.59	27.63
		EoRA	5.54	71.63 (+1.94)	41.97 (+2.38)	28.27 (+0.64)
	60%	-	6.93	63.21	33.70	26.86
		SVD	6.59	65.44	34.12	26.06
		EoRA	6.52 (-0.07)	67.25 (+1.81)	37.71 (+3.59)	27.16 (+1.10)
	2:4	-	7.10	66.32	34.30	25.92
		SVD	6.82	66.28	33.61	25.12
		EoRA	6.75 (-0.07)	68.47 (+2.19)	37.54 (+3.93)	27.53 (+2.41)

A.2. Quantization Error Compensation

Table 9 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 quantized by GPTQ with different bit-width, with compensation via SVD/EoRA of rank 128.

Model	W-bit	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
LLaMA3-8B	Uncompressed	-	6.13	80.09	50.42	40.10
	W4	-	7.00	78.11	45.90	34.07
		SVD	6.80	77.48	45.24	36.51
		EoRA	6.80	78.07 (+0.59)	47.44 (+2.20)	37.21 (+0.7)
	W3	-	15.64	36.78	20.90	22.37
		SVD	10.24	57.19	30.02	26.43
		EoRA	10.06 (-0.18)	60.14 (+2.95)	31.74 (+1.72)	29.11 (+2.68)
LLaMA2-7B	Uncompressed	-	5.47	69.31	39.84	27.67
	W4	-	5.75	67.67	38.13	26.73
		SVD	5.68	66.96	37.62	27.06
		EoRA	5.68	68.18 (+1.22)	38.05 (+0.43)	27.13 (+0.07)
	W3	-	7.76	58.41	31.65	23.50
		SVD	6.84	63.97	34.47	23.90
		EoRA	6.84	65.69 (+1.72)	35.83 (+1.36)	25.79 (+1.89)
LLaMA2-13B	Uncompressed	-	4.88	73.23	45.56	29.91
	W4	-	5.06	71.33	44.28	29.10
		SVD	5.03	71.88	44.19	28.97
		EoRA	5.03	71.80	44.53 (+0.34)	28.90
	W3	-	5.99	63.04	37.28	26.26
		SVD	5.76	64.64	37.54	26.83
		EoRA	5.75 (-0.01)	65.86 (+1.22)	39.50 (+1.96)	27.20 (+0.37)

A.3. Sparse & Quantization Error Compensation

Table 10 | Perplexity and Commonsense/Math reasoning results of LLaMA2/3 models pruned to 2:4 using SparseGPT and quantized to 4-bit with GPTQ, with compensation via SVD/EoRA of rank 128.

Model	Sparsity	W-bit	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
LLaMA3-8B	Uncompressed		-	6.13	80.09	50.42	40.10
			-	86.15	34.59	18.34	19.89
	2:4	W4	SVD	12.84	62.12	29.35	26.86
			EoRA	12.60 (-0.24)	65.9 (+3.78)	31.22 (+1.87)	29.58 (+2.72)
LLaMA2-7B	Uncompressed		-	5.47	69.31	39.84	27.67
			-	9.37	58.41	29.43	23.88
	2:4	W4	SVD	8.42	59.09	29.94	24.42
			EoRA	8.24 (-0.18)	62.33 (+3.24)	31.14 (+1.20)	25.39 (+0.97)
LLaMA2-13B	Uncompressed		-	4.88	73.23	45.56	29.91
			-	7.27	64.09	33.10	24.75
	2:4	W4	SVD	6.98	66.41	33.27	25.29
			EoRA	6.89 (-0.09)	66.58 (+0.17)	35.06 (+1.79)	27.06 (+1.77)

A.4. Compensation With Different Rank

Table 11 | Comparison between SVD and EoRA of different rank on compensating LLaMA2/3 models pruned to 2:4 sparsity by SparseGPT on Perplexity and Commonsense/Math reasoning tasks.

e							
Model	Sparsity	r	Compensation Method	Wikitext2	ARC-E	ARC-C	MathQA
LLaMA3-8B	Uncompressed	-	-	6.13	80.09	50.42	40.10
		-	-	12.32	62.75	30.11	26.43
	2:4	64	SVD EoRA	11.76 11.67 (-0.10)	62.83 65.86 (+3.03)	30.97 33.1 (+2.13)	26.39 28.57 (+2.18)
		128	SVD EoRA	11.31 11.07 (-0.24)	64.89 68.22 (+3.33)	31.99 34.64 (+2.65)	26.49 29.91 (+3.42)
		256	SVD EoRA	10.54 10.25 (-0.30)	68.01 71.00 (+2.99)	34.55 37.96 (+3.41)	28.74 31.59 (+2.85)
		512	SVD EoRA	9.38 9.04 (-0.34)	71.46 74.49 (+3.03)	38.73 41.89 (+3.16)	30.38 34.17 (+3.79)
	Uncompressed	-	-	5.47	69.31	39.84	27.67
		-	-	8.77	60.47	30.11	24.65
	2:4	64	SVD EoRA	8.37 8.29 (-0.08)	60.18 62.58 (+2.40)	30.2 32.16 (+1.96)	24.48 25.62 (+1.14)
		128	SVD EoRA	8.15 7.97 (-0.18)	60.98 63.42 (+2.44)	30.54 32.67 (+2.13)	24.89 25.59 (+0.70)
		256	SVD EoRA	7.74 7.45 (-0.29)	62.71 65.44 (+2.73)	31.99 34.47 (+2.48)	25.19 26.06 (+0.87)
		512	SVD EoRA	7.09 6.80 (-0.29)	65.44 66.91 (+1.47)	34.72 36.77 (+2.05)	24.38 25.96 (+1.58)
LLaMA2-7B	Uncompressed	-	-	4.88	73.23	45.56	29.91
		-	-	7.10	66.32	34.30	25.92
	2:4	64	SVD EoRA	6.95 6.92 (-0.03)	66.24 67.50 (+1.26)	33.95 36.00 (+2.05)	25.56 26.80 (+1.24)
		128	SVD EoRA	6.82 6.75 (-0.07)	66.28 68.47 (+2.19)	33.61 37.54 (+3.93)	25.12 27.53 (+2.41)
		256	SVD EoRA	6.57 6.46 (-0.11)	66.32 70.07 (+3.75)	35.06 38.73 (+3.67)	26.06 27.77 (+1.71)
		512	SVD EoRA	6.20 6.07 (-0.13)	68.72 71.54 (+2.82)	36.51 40.61 (+4.10)	26.39 29.17 (+2.78)
	Uncompressed	-	-	4.88	73.23	45.56	29.91
		-	-	7.10	66.32	34.30	25.92
	2:4	64	SVD EoRA	6.95 6.92 (-0.03)	66.24 67.50 (+1.26)	33.95 36.00 (+2.05)	25.56 26.80 (+1.24)
		128	SVD EoRA	6.82 6.75 (-0.07)	66.28 68.47 (+2.19)	33.61 37.54 (+3.93)	25.12 27.53 (+2.41)
		256	SVD EoRA	6.57 6.46 (-0.11)	66.32 70.07 (+3.75)	35.06 38.73 (+3.67)	26.06 27.77 (+1.71)
		512	SVD EoRA	6.20 6.07 (-0.13)	68.72 71.54 (+2.82)	36.51 40.61 (+4.10)	26.39 29.17 (+2.78)

A.5. Ablation: Fine-Tuning with Different Numbers of Training Data

Table 12 | Ablation study on the effect of using different proportions of the dataset for fine-tuning 2:4 pruned LLaMA3-8B models with varying low-rank matrix initializations on Commonsense/Math reasoning tasks.

Dataset Ratio	LoRA initialization	ARC-E	ARC-C	MathQA
-	-	80.09	50.42	40.10
100%	Standard	73.82	41.30	45.42
	SVD	74.45	43.68	48.77
	EoRA	76.01 (+1.56)	48.54 (+4.86)	54.67 (+5.90)
50%	Standard	71.67	38.56	40.23
	SVD	72.18	41.46	42.51
	EoRA	75.42 (+3.24)	46.41 (+4.95)	48.91 (+6.40)
30%	Standard	69.82	36.77	36.71
	SVD	72.01	39.76	40.60
	EoRA	73.86 (+1.85)	43.85 (+4.09)	44.79 (+4.19)

In this section, we show that fine-tuning with the EoRA-compensated model is robust to various ratios of training data. We follow the setting in Sec. 4.5 on compressed LLaMA3-8B models with 2:4 sparsity compression. As shown in Table 12, using EoRA for initialization consistently outperforms both standard and SVD initialization across various dataset ratios, with accuracy improvements (ARC-E/ARC-C/MathQA) of 3.24%/4.95%/6.4% and 1.85%/4.09%/4.19% over SVD when fine-tuning using 50% and 30% training data, respectively.

A.6. Quantizing EoRA with Efficiency Evaluation

Table 13 | Accuracy and the Model Size of quantizing EoRA of rank {128,512} to 4/3-bit on compensating LLaMA3-8B of {2:4 spararsity, 4/3-bit}.

Compression method	Config	r	W-bit of EoRA	Model Size (GB)	Wikitext2	ARC-E	ARC-C	MathQA
-	-	-	-	15.08	6.13	80.09	50.42	40.10
SparseGPT	2:4	128	-	9.12	12.32	62.75	30.11	26.43
			16	9.77	11.07	68.22	34.64	29.91
			4	9.28	11.15	67.55	34.47	29.91
			3	9.24	11.31	68.01	34.72	29.71
			16	11.70	9.04	74.49	41.89	34.17
		512	4	9.77	9.12	74.62	41.46	33.63
			3	9.64	9.32	72.30	40.35	32.66
			-	5.35	7.00	78.11	45.90	34.07
			16	6.01	6.80	78.07	47.44	37.21
			4	5.50	6.83	78.78	47.35	36.78
GPTQ	W4	128	3	5.46	6.90	78.24	47.18	36.52
			16	7.85	6.50	79.75	48.29	38.72
			4	6.01	6.61	78.87	48.80	38.92
			3	5.90	6.75	78.49	46.92	36.88
			-	4.63	15.64	36.78	20.90	22.37
		128	16	5.28	10.06	60.14	31.74	29.11
			4	4.78	10.26	61.53	31.48	28.64
			3	4.74	11.68	56.52	29.18	26.70
			16	7.16	8.53	71.00	38.82	31.89
			4	5.28	8.67	68.35	40.01	31.69
		512	3	5.18	10.19	66.70	35.40	30.45
			-	-	-	-	-	-
			16	-	-	-	-	-
			4	-	-	-	-	-
