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

Shih-Yang Liu[1,2]*, **Maksim Khadkevich**[1], **Nai Chit Fung**[2], **Charbel Sakr**[1], **Chao-Han Huck Yang**[1], **Chien-Yi Wang**[1], **Saurav Muralidharan**[1], **Hongxu Yin**[1], **Kwang-Ting Cheng**[2], **Jan Kautz**[1], **Yu-Chiang Frank Wang**[1], **Pavlo Molchanov**[1], **Min-Hung Chen**[1]
[1]NVIDIA, [2]Hong Kong University of Science and Technology

Abstract

While post-training compression techniques effectively reduce the memory footprint, latency, and power consumption of Large Language Models (LLMs), they often result in noticeable accuracy degradation and remain limited by hardware and kernel constraints that restrict supported compression formats—ultimately reducing flexibility across a wide range of deployment scenarios. In this work, we propose ***EoRA***—a novel, **fine-tuning-free** method that augments compressed LLMs with low-rank matrices, allowing users to rapidly enhance task-specific performance and freely balance the trade-off between accuracy and computational overhead beyond the constraints of compression formats. EoRA consistently outperforms prior fine-tuning-free low-rank methods in recovering the accuracy of compressed LLMs, achieving notable accuracy improvements (e.g., $\mathbf{10.84}\%$ on ARC-Challenge, $\mathbf{6.74}\%$ on MathQA, and $\mathbf{11.45}\%$ on GSM8K for LLaMA3-8B compressed to 3-bit). We also introduce an optimized CUDA kernel, accelerating inference by up to 1.4× and reducing memory overhead through quantizing EoRA. Overall, EoRA offers a prompt solution for improving the accuracy of compressed models under varying user requirements, enabling more efficient and flexible deployment of LLMs. Code is available at `https://github.com/NVlabs/EoRA`.

1 Introduction

Although Large Language Models (LLMs) excel in various tasks, their deployment remains challenging due to high inference costs. Post-training compression methods, like quantization (Frantar et al., 2023; Lin et al., 2024; Liu et al., 2025; Tseng et al., 2024) and pruning (Ma et al., 2023; Frantar & Alistarh, 2023; Sun et al., 2024), aim to reduce computational demands but typically cause accuracy loss or face hardware/kernel constraints, limiting deployment flexibility. For instance, strict hardware-supported formats, such as 2:4 sparsity on NVIDIA GPUs or integer-only quantization kernels, prevent intermediate approaches (e.g., 2.X:4 sparsity or arbitrary-bit quantization) that could offer a more adaptable trade-off between accuracy and latency based on user needs.

To relax these format constraints and improve the accuracy of the compressed models on specified tasks, we formulate a new problem, termed *customized compensation*: Given a compressed LLM, we attach residual low-rank paths to it to *compensate* for compression errors and enhance task-specific accuracy, enabling more flexible control over the trade-off between accuracy and compression ratio to accommodate varying user requirements. For example, a user may wish to boost the accuracy of a 2:4 sparsity-pruned model on math reasoning tasks, accepting a modest increase in memory usage and inference latency in return. Importantly, in our problem setting, the weights of the compressed model are not modified during compensation. This enables deployment of a single, general compressed backbone alongside lightweight, task-specific low-rank modules that can be dynamically loaded as needed—allowing for efficient integration with existing multi-adapter inference frameworks (e.g., (The vLLM Team)) as illustrated in Figure 1. A naive solution is to apply SVD (Li et al., 2024; Yao et al., 2024) for compensation; however, this neglects calibration data and thus fails to enhance

*Work done during Shih-Yang's internship at NVIDIA.

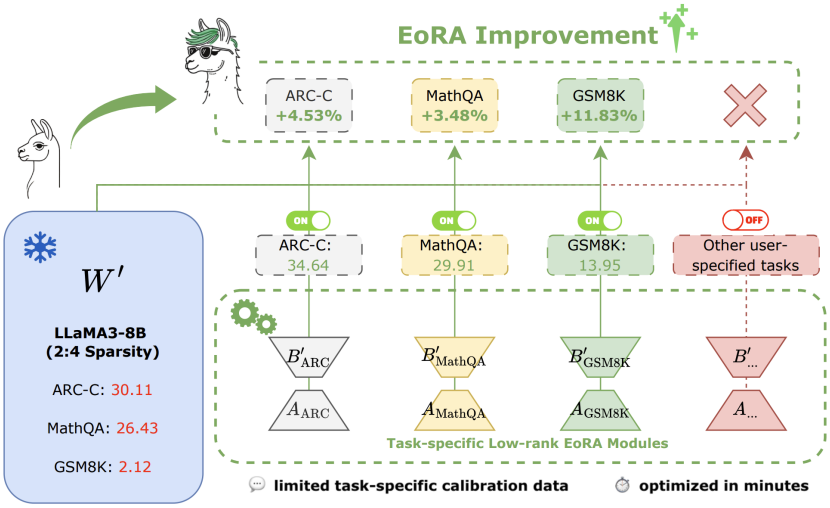


Figure 1: An overview of our proposed EoRA, which enables swift task-specific accuracy enhancement for compressed LLMs without **fine-tuning**, using only a small amount of downstream calibration data. At inference time, a single compressed backbone is loaded, while lightweight, task-specific low-rank modules can be dynamically toggled on and off on demand, enabling efficient and flexible deployment. EoRA with rank 128 boosts the accuracy of the LLaMA3-8B model pruned to 2:4 structured sparsity by 4.53%, 3.48%, and 11.83% on ARC-C, MathQA, and GSM8K, respectively—all achieved within minutes using just 64 calibration samples per task.

task-specific performance. Alternatively, LoRA-based methods, such as (Li et al., 2024; Dettmers et al., 2023) require fine-tuning, limiting their applicability for rapid task adaptation. These limitations prompt an important question: "*How can we swiftly improve the task-specific accuracy for compressed LLMs without fine-tuning?*"

To tackle this research challenge, we introduce *fine-tuning-free **E**igenspace **L**ow-**R**ank **A**pproximation (**EoRA**)*, a method designed to efficiently enhance the task-specific accuracy of compressed LLMs while offering users greater flexibility in managing the trade-off between accuracy and computational overhead. EoRA operates by projecting the compression error into the task-specific eigenspace of each layer's input activations, followed by applying SVD to approximate the projected error. This approach ensures that the SVD approximation error directly aligns with the task-specific compression loss. As a **fine-tuning-free** method, EoRA avoids backpropagation and completes in just a few minutes using minimal calibration data.

We validate the effectiveness of EoRA in boosting the accuracy of compressed LLMs (LLaMA2-7B/13B and LLaMA3-8B) on language generation, commonsense reasoning, and math tasks. Our method consistently outperforms other training-free baselines, especially for aggressively compressed (including *pruned, quantized, and both*) models (e.g., $\mathbf{2.65}\%$, $\mathbf{3.42}\%$, and $\mathbf{10.99}\%$ improvement on ARC-Challenge, MathQA, and GSM8K when compensating 2:4 pruned LLaMA3-8B). To reduce redundant memory transfer overhead from running low-rank compensation, we design a fused kernel that integrates low-rank and quantization operations, achieving up to 1.4× speedup.

The summary of our contributions is as follows:

- **Flexible and Task-specific Model Compensation**: We propose, *fine-tuning-free **E**igenspace **L**ow-**R**ank **A**pproximation (**EoRA**)*, a *training-free* approach that improves the task-specific accuracy of compressed LLMs in minutes using minimal calibration data, while supporting more flexible compression ratios unconstrained by hardware or kernel-imposed format limitations.
- **Eigenspace Projection**: EoRA leverages calibration data to project the compression error into the task-specific eigenspace and utilizes the corresponding eigenvalues as importance indicators, effectively aligning the approximation error with task-specific compression loss.
- **Efficient Inference**: We develop a custom kernel that fuses part of the low-rank matrix multiplication with a quantization kernel, accelerating EoRA inference by up to 1.4x. EoRA is also robust to quantization, further minimizing the size-overhead from low-rank compensation matrices.

2 METHOD: EORA

In this work, we introduce a new problem, termed *customized compensation*: Given an already compressed model, the objective is to add residual low-rank paths that *rapidly compensate* for compression errors and enhance task-specific accuracy according to user-defined accuracy/overhead requirements. Crucially, the compressed model's weights remain unchanged during compensation, enabling the deployment of a single, general compressed backbone with lightweight, task-specific low-rank modules that can be dynamically loaded as needed, facilitating efficient integration with existing inference frameworks, as illustrated in Figure 1.

A simple approach to obtain low-rank residual paths that compensate for compression errors is to directly apply Singular Value Decomposition (SVD) (Li et al., 2024; Yao et al., 2024; Li et al., 2025). However, naively applying SVD to optimize error approximation lossdoes not ensure minimization of the layer-wise compression loss, and ignores calibration data, making it ineffective for task-specific accuracy recovery. While LoRA-based methods (Li et al., 2024; Dettmers et al., 2023) address this issue, they require fine-tuning and are less suitable for rapid adaptation. This raises a key question: "*How can we swiftly improve the task-specific accuracy for compressed LLMs without fine-tuning?*".

To tackle the challenge of improving task-specific accuracy of compressed LLMs without fine-tuning, we introduce *fine-tuning-free **E**igenspace **Lo**w-**R**ank **A**pproximation (**EoRA**)*—a method that preserves the efficiency of existing training-free solutions while substantially improving their *effectiveness* in task-specific accuracy recovery, as shown in Algorithm 1. EoRA compensation is applied to each compressed linear layer, and the overall **fine-tuning-free** optimization across all linear layers can be completed in just a few minutes, enabling users to rapidly enhance the accuracy of compressed LLMs on their chosen downstream tasks using only a small amount of task-specific calibration data—without any need for backpropagation. EoRA can also provide better initialization for further LoRA fine-tuning, offering users the option to further improve accuracy if additional computational resources are available. Moreover, the low-rank matrices of EoRA are robust to quantization, which can further reduce the additional memory/inference cost (Section A.8 for more details). Please check Section A.1 and A.2 of the Appendix for the full EoRA explanation.

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 Eigendecompostion 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 $\Delta W'$, $B'A' = U'\Sigma'V' = \text{SVD}(\Delta W')$
6. Project the approximation back to the original space $A = A'Q'^{-1}$
7. The final forward pass of current layer becomes $\hat{W}X + B'AX$

3 EXPERIMENTS

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) and GPTQ (Frantar et al., 2023). We follow the settings from (Huang et al., 2024) to construct the calibration dataset for both SparseGPT and GPTQ. We compare EoRA with ZeroQuant-V2 (Yao et al., 2024), Act-S (Yuan et al., 2023), and ApiQ (Liao et al., 2024), on language generation (WikiText2), commonsense reasoning (ARC-C (Clark et al., 2018)), and math reasoning (MathQA (Amini et al., 2019) and GSM8K (Cobbe et al., 2021)) tasks using the LM-Evaluation-Harness framework (Gao et al., 2024). Note that the optimization time for both EoRA and Act-S is comparable, with each completing within minutes, whereas ApiQ requires over hours to optimize. Please see Sec A.3 for more experimental details.

Main Results. We evaluate EoRA on LLaMA3-8B quantized with GPTQ to 4-bit and 3-bit to assess the effectiveness of EoRA in compensating for quantization error. The ranks for all the methods are

Table 1: Perplexity and commonsense/math reasoning results for LLaMA3-8B quantized to 3/4-bits using GPTQ, with all compensation methods evaluated at rank 128.

Model	W bits	Compensation Method	Wikitext2 ↓	ARC-C ↑	MathQA ↑	GSM8K ↑
LLaMA3-8B	-	-	6.13	50.42	40.10	36.23
	W4	-	7.00	45.90	34.07	27.74
		ZeroQuant-V2	<u>6.80</u>	45.24	<u>36.51</u>	**31.23**
		Act-S	6.82	**47.86**	35.84	29.34
		ApiQ	6.87	46.58	36.18	30.09
		EoRA (Ours)	**6.80**	<u>47.44</u>	**37.21**	<u>30.70</u>
	W3	-	15.64	20.90	22.37	0.45
		ZeroQuant-V2	10.24	30.02	26.43	3.79
		Act-S	<u>10.19</u>	<u>31.28</u>	25.42	4.09
		ApiQ	10.41	30.46	<u>26.86</u>	<u>10.79</u>
		EoRA (Ours)	**10.06**	**31.74**	**29.11**	**11.90**

set to 128. From Table 1, 3-bit quantization causes significant accuracy degradation, with losses up to 29.5%/17.7%/35.8% on ARC-C, MathQA, and GSM8K, respectively. By applying EoRA, we demonstrate that the accuracy loss can be reduced to 18.7%/10.9%/24.3% on ARC-C, MathQA, and GSM8K—providing 10.8%/6.7%/11.5% improvement, outperforming all the baseline methods 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. Results for LLaMA2-7B/13B are presented in Table 2 in the Appendix (Section A.4), with a similar improvement trend. For compensating sparsity error, EoRA consistently outperforms all training-free baselines, achieving gains of 2.9%, 2.1%, and 10.7% over Act-S on ARC-C, MathQA, and GSM8K, respectively, as shown in Section A.5 of the Appendix. We also explore the feasibility of using EoRA to improve ultra-compressed models that combine both pruning and quantization, and EoRA continues to outperform all baselines on ARC-C and MathQA. Please check Section A.6 of the Appendix for more details.

EoRA Efficiency Optimization. To reduce computational overhead in practice, we propose fusing the low-bit weight quantization kernel with the matrix multiplication of B, which shares the same output, as illustrated in Figure 2 (a) of Appendix. our custom EoRA kernel substantially accelerates inference, achieving a speedup of up to 1.4x over FP16 with EoRA of rank 128 at 3-bit quantization, as shown in Table 6. Finally, EoRA can also be quantized to further reduce the additional cost of residual low-rank compensation paths. As shown in Figure 2 (b) and Table 7 of the Appendix, 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%. Please check Section A.8 of Appendix for more details.

4 Conclusion

We present EoRA, a novel fine-tuning-free approach that rapidly boosts the task-specific accuracy of compressed LLMs using minimal calibration data, while offering greater flexibility by relaxing compression format constraints. EoRA remains robust under quantization, reducing memory overhead, and can serve as a strong initialization for LoRA fine-tuning. EoRA achieves strong results across language, commonsense, and math reasoning tasks, outperforming prior low-rank methods. Overall, EoRA is a scalable, efficient solution for improving compressed LLMs across diverse deployment settings, with potential extensions to new architectures and modalities.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

For this submission, Large Language Models (LLMs) were only used for sentence polishing, grammar checking, and LaTeX error correction.

ETHICS STATEMENT

We do not collect or annotate any human subject data; all experiments use publicly available datasets under research licenses. We adhere to the terms of use specified by the original dataset creators and provide appropriate citations. Our approach does not introduce additional risks of data misuse or privacy leakage. Therefore, we do not foresee any obvious negative societal impacts. Nonetheless, we encourage responsible use and emphasize that our framework should be applied in alignment with safety and ethical guidelines.

REFERENCES

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.

Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. Low-rank quantization-aware training for llms. *arXiv preprint arXiv:2406.06385*, 2024.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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.

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 Hoefler, 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`.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations*, 2022.

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*, 2024.

Geonho Lee, Janghwan Lee, Sukjin Hong, Minsoo Kim, Euijai Ahn, Du-Seong Chang, and Jungwook Choi. Rilq: Rank-insensitive lora-based quantization error compensation for boosting 2-bit large language model accuracy. In *AAAI Conference on Artificial Intelligence*, 2025.

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. In *International Conference on Learning Representations*, 2025.

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*, 2024.

Baohao Liao, Christian Herold, Shahram Khadivi, and Christof Monz. Apiq: Finetuning of 2-bit quantized large language model. In *Empirical Methods in Natural Language Processing*, 2024.

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*, 2023.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, 2025.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *Neural Information Processing Systems*, 2023.

Mohammad Mozaffari, Amir Yazdanbakhsh, and Maryam Mehri Dehnavi. Slim: One-shot quantization and sparsity with low-rank approximation for llm weight compression. In *International Conference on Machine Learning*, 2025a.

Mohammad Mozaffari, Amir Yazdanbakhsh, Zhao Zhang, and Maryam Mehri Dehnavi. Slope: Double-pruned sparse plus lazy low-rank adapter pretraining of llms. In *International Conference on Learning Representations*, 2025b.

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.

Rajarshi Saha, Naomi Sagan, Varun Srivastava, Andrea Goldsmith, and Mert Pilanci. Compressing large language models using low rank and low precision decomposition. In *Neural Information Processing Systems*, 2024.

Charbel Sakr and Brucek Khailany. Espace: Dimensionality reduction of activations for model compression. In *Neural Information Processing Systems*, 2024.

Meyer Scetbon and James Hensman. Low-rank correction for quantized llms. *arXiv preprint arXiv:2412.07902*, 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.

The vLLM Team. MultiLoRA Inference. `https://docs.vllm.ai/en/stable/getting_started/examples/multilora_inference.html`, 2025.

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. In *International Conference on Learning Representations*, 2025.

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.

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 *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.

Cheng Zhang, Jianyi Cheng, George A Constantinides, and Yiren Zhao. Lqer: Low-rank quantization error reconstruction for llms. In *International Conference on Machine Learning*, 2024.

Cheng Zhang, Jeffrey TH Wong, Can Xiao, George A Constantinides, and Yiren Zhao. Qera: an analytical framework for quantization error reconstruction. In *International Conference on Learning Representations*, 2025.

Stephen Zhang and Vardan Papyan. Oats: Outlier-aware pruning through sparse and low rank decomposition. In *International Conference on Learning Representations*, 2025.

A Appendix

A.1 Preliminaries of EoRA

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:

$$\arg\min_{\hat{W}_l} ||W_l X_l - \hat{W}_l X_l||_F \tag{1}$$

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 accuracy/overhead requirements of different users.

To bypass the limitations of fixed compression formats and enhance the accuracy of compressed models on user-specified tasks, we introduce a new problem, termed *customized compensation*: Given an already compressed model, the objective is to add residual low-rank paths that *compensate* for compression errors and enhance task-specific accuracy according to user-defined accuracy/overhead requirements. Crucially, the compressed model's weights remain unchanged during compensation, enabling the deployment of a single, general compressed backbone with lightweight, task-specific low-rank modules that can be dynamically loaded as needed, facilitating efficient integration with existing inference frameworks, as illustrated in Figure 1.

A simple approach to obtain low-rank residual paths that compensate for compression errors is to directly apply Singular Value Decomposition (SVD) (Li et al., 2024; Yao et al., 2024; Li et al., 2025). 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 $\Delta W_l \approx U_l \Sigma_l V_l^T$, 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:

$$\arg\min_{B_l, A_l} ||\Delta W_l - B_l A_l||_F \tag{2}$$

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 ensure minimization of the layer-wise compression loss (Eq.1) and ignores calibration data, making it ineffective for task-specific accuracy recovery. While LoRA-based methods (Li et al., 2024; Dettmers et al., 2023) address this issue, they require fine-tuning and are less suitable for rapid adaptation. This raises a key question: "*How can we swiftly improve the task-specific accuracy for compressed LLMs without fine-tuning?*". For simplicity, we omit the subscript l, which corresponds to layer l in the following sections.

A.2 Method (in Details): EoRA

To tackle the challenge of improving task-specific accuracy of compressed LLMs without fine-tuning, we introduce *fine-tuning-free **E**igenspace **L**ow-**R**ank **A**pproximation (**EoRA**)*—a method that preserves the efficiency of existing training-free solutions while substantially improving their *effectiveness* in task-specific accuracy recovery.

First, we propose projecting the compression error into the eigenspace (Stewart, 2001) of the corresponding layer's input activations, ensuring a direct alignment between the error approximation loss (Eq. 2) and the overall layer-wise model compression loss (Eq. 1). 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 task-specific calibration set. The eigendecomposition $\tilde{X}\tilde{X}^T = Q\Lambda Q^T$ 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 the 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:

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

where SVD is applied to approximate $\Delta W'$ as $\text{SVD}(\Delta W') \approx U'\Sigma' V'^T$, and $\Sigma' \in \mathbb{R}^{r\times r}$ contains the top-r singular values. $U' \in \mathbb{R}^{d\times r}$ and $V' \in \mathbb{R}^{k\times r}$ are orthonormal matrices with columns representing the corresponding singular vectors. Then the low-rank matrices B' and A' are then assigned as $B' = U'\Sigma'$ and $A' = V'^T$. This loss function ensures that error columns associated with larger eigenvalues are approximated more accurately than those with smaller eigenvalues. We then multiply the low-rank approximation in the eigenspace $\Delta W'$ with $Q'^{-1} = \sqrt{\Lambda}^{-1}Q^T$ to project back to the original space, obtaining the final task-specific compression error approximation as $\Delta W = \Delta W' Q'^{-1} \approx B'A'Q'^{-1}$. Q' is invertible because $Q'^{-1} = \sqrt{\Lambda}^{-1}Q^T$, and $Q'Q'^{-1} = Q\sqrt{\Lambda}\sqrt{\Lambda}^{-1}Q^T$. Here, the middle term $\sqrt{\Lambda}\sqrt{\Lambda}^{-1}$ simplifies to the identity matrix, and since Q is an orthogonal matrix, QQ^T also yields the identity matrix. 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 one linear layer of the compressed model compensated with EoRA for the input activation X can be formulated as:

$$\hat{W}X + B'AX \tag{4}$$

Mapping EoRA loss (Eq. 3) to task-specific compression loss (Eq. 1): When Eq. 1 is conditioned on different task-specific calibration data, it also implies the compressed model's accuracy on each corresponding task. Therefore, the objective of task-specific low-rank compensation is to approximate ΔW that minimizes Eq. 1, using input activations X derived from the calibration data of different tasks. To achieve this, we reformulate the compression objective for each layer as:

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

Since the Frobenius norm of a matrix is equal to the square root of its Gram matrix (Sun, 1991; Wang et al., 2025), the minimization problem can be rewritten as:

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

Directly applying SVD on ΔW initially does not guarantee the minimization of the above equation Eq. 6. 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 1. *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 $\Delta W' = \Delta WQ\sqrt{\Lambda}$, 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

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

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

$$[\text{trace}((\Delta WQ\sqrt{\Lambda} - BAQ\sqrt{\Lambda})(\Delta WQ\sqrt{\Lambda} - BAQ\sqrt{\Lambda})^T)]^{\frac{1}{2}} \tag{8}$$

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

$$\begin{aligned} &[\text{trace}((\Delta WQ' - BAQ')(\Delta WQ' - BAQ')^T)]^{\frac{1}{2}} \\ &= [\text{trace}((\Delta W' - BAQ')(\Delta W' - BAQ')^T)]^{\frac{1}{2}} \\ &= ||\Delta W' - BAQ'||_F \end{aligned} \tag{9}$$

where the square root of the Gram matrix can be transformed back to the corresponding Frobenius norm according to (Sun, 1991). By setting $BAQ' = B'A'$, $||\Delta W' - BAQ'||_F$ 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 $\text{SVD}(\Delta W') = B'A'$. Note that the above minimization is constrained to the rank of A' and B'.

A.3 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., 2024) to construct the calibration dataset for both SparseGPT and GPTQ.

We compare EoRA with **ZeroQuant-V2** (Yao et al., 2024) which proposes using simple SVD for optimizing Eq. 2. Although Activation-aware Singular Value Decomposition (ASVD) (Yuan et al., 2023) is designed to replace the entire model with its low-rank decomposition rather than approximating the compression errors, its strategy of incorporating activation distribution variance can also be adapted for error compensation using low-rank matrices. Specifically, we scale the compression error ΔW using a diagonal scaling matrix S, where each diagonal entry S_{ii} is computed based on the average absolute value of the activations $\tilde{X}$ in the i-th channel as $S_{ii} = \left(\frac{1}{n}\sum_{j=1}^{n}|\tilde{X}_{ij}|\right)^{\frac{1}{2}}$. Here, n denotes the number of activation entries in the i-th channel. We then apply SVD to the scaled error $\Delta W" = \Delta WS$ to obtain its low-rank approximation. Since S is invertible, we can project the approximation back to the original space as $\Delta W = \Delta W"S^{-1} \approx B"A"S^{-1}$—same as how EoRA projects its compensation back to the original space. We refer to this method as **Act-S** in the remainder of this paper. We also compare EoRA with a training-based method, **ApiQ** (Liao et al., 2024), which optimizes low-rank matrices (A and B) using gradient-based training to minimize Eq. 6. In our comparison, we limit the evaluation to the layer-wise variant of ApiQ, as other variants

require substantially more memory or training time. These more resource-intensive versions align more closely with PEFT methods rather than training-free low-rank approximation approaches. For instance, when applied at the model level, ApiQ effectively becomes equivalent to training LoRA on top of a compressed model—shifting its focus toward fine-tuning rather than training-free compensation, and thus falling outside the scope of this study. Note that the optimization time for both EoRA and Act-S is comparable, with each completing within minutes, whereas ApiQ requires over hours to optimize.

We evaluate EoRA and the baselines on improving the task-specific accuracy of the compressed LLMs on language generation, commonsense reasoning, and math reasoning tasks using the LM-Evaluation-Harness framework (Gao et al., 2024). We pick WikiText2 for the language generation task and perplexity as the evaluation metric. For commonsense reasoning, we select ARC-Challenge (ARC-C) (Clark et al., 2018), and for math reasoning ability, we choose MathQA (Amini et al., 2019) and GSM8K (Cobbe et al., 2021). We sample 128 concatenated sentences of length 2048 from the WikiText2 training set as the calibration set for EoRA, Act-S, and ApiQ for the language generation task. For commonsense reasoning tasks, we sample 32 concatenated sentences of length 2048 from the ARC training set and combine them with 32 concatenated sentences of the same length from C4 (Raffel et al., 2020) to construct the calibration set for EoRA, Act-S, and ApiQ. Similarly, for the math reasoning tasks, we sample 32 concatenated sentences of length 2048 from the MathQA/GSM8K training set and combine them with 32 concatenated sentences from C4 to form the calibration set for the three methods.

A.4 Quantization Error Compensation

A similar trend of improvement for LLaMA2-7B/13B with EoRA is observed in Table 2. It is worth noting that EoRA-enhanced models outperform smaller models quantized at higher precision, while high-precision models usually outperform low-precision models. For example, on GSM8K (Table 2), 3-bit LLaMA2-13B (4.62) performs worse than 4-bit LLaMA2-7B (9.93), but EoRA-enhanced 3-bit LLaMA2-13B (15.08) outperforms 4-bit LLaMA2-7B (9.93). This highlights EoRA's advantage in terms of greater flexibility: rather than shrinking the model architecture, it enables more effective compression of larger models while preserving higher accuracy.

A.5 Sparsity Error Compensation

To assess the effectiveness of EoRA in compensating for sparsity error, we compare EoRA with all the baselines on LLaMA2-7B/13B and LLaMA3-8B models pruned with SparseGPT to 2:4 sparsity—the only sparsity format that yields actual inference speedups on GPUs. Rank of all the compensation methods is set to 128, and the results of LLaMA3-8B are provided in Table 3. EoRA consistently outperforms all training-free baselines, achieving gains of 2.9%, 2.1%, and 10.7% over Act-S on ARC-C, MathQA, and GSM8K, respectively. Furthermore, it surpasses ApiQ by 0.4% on ARC-C and 1.1% on MathQA, while delivering comparable results on GSM8K—all with significantly faster optimization time (15 minutes vs. 2.5 hours). Furthermore, EoRA proves robustness across different model sizes, continuing to outperform ZeroQuant-V2, Act-S and ApiQ in boosting the accuracy of 2:4 pruned LLaMA2-7B/13B across ARC-C and MathQA as shown in Table 3. We further assess the generalizability and compatibility of EoRA with pruning methods beyond SparseGPT. Specifically, we evaluate EoRA on LLaMA3-8B pruned to 2:4 sparsity using Wanda (Sun et al., 2024), where EoRA continues to outperform all the training-free baseline methods. For additional details, please refer to Section A.7.

A.6 Sparsity & Quantization Error Compensation

Here, we examine the feasibility of applying EoRA to compensate for ultra-compressed models that undergo both pruning and quantization, as shown in Table 4. 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. 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

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

Model	W bits	Compensation Method	Wikitext2 ↓	ARC-C ↑	MathQA ↑	GSM8K ↑
LLaMA3-8B	-	-	6.13	50.42	40.10	36.23
	W4	-	7.00	45.90	34.07	27.74
		ZeroQuant-V2	<u>6.80</u>	45.24	<u>36.51</u>	**31.23**
		Act-S	6.82	**47.86**	35.84	29.34
		ApiQ	6.87	46.58	36.18	30.09
		EoRA	**6.80**	<u>47.44</u>	**37.21**	<u>30.70</u>
	W3	-	15.64	20.90	22.37	0.45
		ZeroQuant-V2	10.24	30.02	26.43	3.79
		Act-S	<u>10.19</u>	<u>31.28</u>	25.42	4.09
		ApiQ	10.41	30.46	<u>26.86</u>	<u>10.79</u>
		EoRA	**10.06**	**31.74**	**29.11**	**11.90**
LLaMA2-7B	-	-	5.47	39.84	27.67	14.85
	W4	-	5.75	38.13	26.73	9.93
		ZeroQuant-V2	<u>5.68</u>	37.62	27.06	10.15
		Act-S	<u>5.68</u>	**39.84**	**27.50**	9.86
		ApiQ	<u>5.68</u>	<u>39.59</u>	27.00	<u>11.22</u>
		EoRA	**5.68**	38.05	<u>27.13</u>	**11.45**
	W3	-	7.76	31.65	23.50	0.38
		ZeroQuant-V2	<u>6.84</u>	<u>34.47</u>	23.90	2.04
		Act-S	6.86	32.67	25.02	2.57
		ApiQ	6.86	33.70	**26.06**	<u>7.13</u>
		EoRA	**6.84**	**35.83**	<u>25.79</u>	**7.50**
LLaMA2-13B	-	-	4.88	45.56	29.91	21.37
	W4	-	5.06	44.28	29.10	21.00
		ZeroQuant-V2	<u>5.03</u>	<u>44.19</u>	28.97	19.48
		Act-S	5.04	43.60	<u>29.48</u>	18.49
		ApiQ	5.04	42.83	**29.64**	<u>21.45</u>
		EoRA	**5.03**	**44.53**	28.90	**22.36**
	W3	-	5.99	37.28	26.26	4.62
		ZeroQuant-V2	<u>5.76</u>	37.54	26.83	9.93
		Act-S	5.81	38.90	26.26	9.17
		ApiQ	5.81	**39.67**	**27.47**	<u>14.32</u>
		EoRA	**5.75**	<u>39.50</u>	<u>27.20</u>	**15.08**

Table 3: Perplexity and Commonsense/Math reasoning results of LLaMA2/3 pruned by SparseGPT to 2:4 sparsity, with low-rank compensation of rank 128.

Model	Sparsity	Compensation Method	Wikitext2 ↓	ARC-C ↑	MathQA ↑	GSM8K ↑
LLaMA3-8B	-	-	6.13	50.42	40.10	36.23
	2:4	-	12.32	30.11	26.43	2.12
		ZeroQuant-V2	11.31	31.99	26.49	2.96
		Act-S	11.32	31.74	26.73	3.26
		ApiQ	<u>11.08</u>	<u>34.21</u>	<u>28.77</u>	**14.55**
		EoRA	**11.07**	**34.64**	**29.91**	<u>13.95</u>
LLaMA2-7B	-	-	5.47	39.84	27.67	14.85
	2:4	-	8.77	30.11	24.65	1.66
		ZeroQuant-V2	8.15	30.54	24.89	1.97
		Act-S	8.22	30.20	25.09	2.73
		ApiQ	<u>8.03</u>	<u>32.67</u>	**26.36**	**7.58**
		EoRA	**7.97**	**32.67**	<u>25.59</u>	<u>6.22</u>
LLaMA2-13B	-	-	4.88	45.56	29.91	21.37
	2:4	-	7.10	34.30	25.92	2.65
		ZeroQuant-V2	6.82	33.61	25.12	3.56
		Act-S	6.92	34.12	25.69	4.09
		ApiQ	<u>6.80</u>	<u>36.68</u>	<u>27.16</u>	**12.13**
		EoRA	**6.75**	**37.54**	**27.53**	<u>10.91</u>

Table 4: 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 rank set to 128.

Model	Sparsity	W bits	Compensation Method	Wikitext2 ↓	ARC-C ↑	MathQA ↑	GSM8K ↑
LLaMA3-8B	-	-	-	6.13	50.42	40.10	36.23
	2:4	W4	-	86.15	18.34	19.89	0.00
			ZeroQuant-V2	12.84	29.35	26.86	1.59
			Act-S	12.99	27.90	25.59	1.90
			ApiQ	<u>12.77</u>	<u>30.71</u>	<u>28.74</u>	**11.06**
			EoRA	**12.60**	**31.22**	**29.58**	<u>10.16</u>
LLaMA2-7B	-	-	-	5.47	39.84	27.67	14.85
	2:4	W4	-	9.37	29.43	23.88	0.99
			ZeroQuant-V2	8.42	29.94	<u>24.42</u>	1.67
			Act-S	<u>8.24</u>	28.92	24.05	1.97
			ApiQ	**8.03**	<u>30.63</u>	24.12	**7.05**
			EoRA	<u>8.24</u>	**31.14**	**25.39**	<u>4.93</u>
LLaMA2-13B	-	-	-	4.88	45.56	29.91	21.37
	2:4	W4	-	7.27	33.10	24.75	2.20
			ZeroQuant-V2	6.98	33.27	25.29	2.65
			Act-S	6.92	34.64	26.09	2.81
			ApiQ	**6.80**	**36.17**	<u>26.96</u>	**12.59**
			EoRA	<u>6.89</u>	<u>35.06</u>	**27.06**	<u>9.86</u>

improves the accuracy of these compressed models, reducing perplexity by up to 73.55 and boosting accuracy by 12.88%/9.60%/10.16% on ARC-C/MathQA/GSM8K tasks. Additionally, EoRA consistently outperforms both ZeroQuant-V2 and Act-S across LLaMA2 and LLaMA3. For instance, EoRA exceeds ZeroQuant-V2 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.

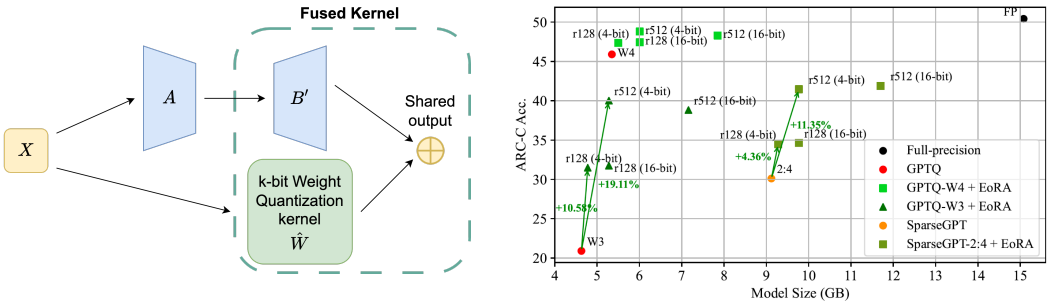
A.7 Compatibility With Various Compression Methods

Table 5: Comparison between compensation methods of rank set to 128 on compensating LLaMA3-8B models pruned to 2:4 sparsity with Wanda on Perplexity and Commonsense/Math reasoning tasks.

Compression Method	Compression Setting	Compensation Method	Wikitext2 ↓	ARC-C ↑	MathQA ↑	GSM8K ↑
Full-precision	-	-	6.13	50.42	40.10	36.23
Wanda	2:4	-	21.42	27.04	25.09	0.76
		ZeroQuant-V2	17.16	30.46	26.16	1.28
		Act-S	17.37	29.77	26.73	1.51
		ApiQ	<u>14.30</u>	<u>31.91</u>	<u>29.61</u>	**12.81**
		EoRA	**14.04**	**34.81**	**30.05**	<u>11.52</u>

In this section, we study the generalizability and compatibility of EoRA with different pruning methods beyond SparseGPT. 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 A.3, and maintain the same settings when utilizing EoRA to address compression errors. We evaluate EoRAon LLaMA3-8B pruned with Wanda to 2:4 structured sparsity. The ranks of all low-rank compensation methods are set to 128. Table 5 demonstrates that EoRAconsistently outperforms every training-free method, both ZeroQuant-V2 and Act-S, in improving accuracy across all the tasks. For example, EoRA achieves accuracy gains of 7.77%/4.96%/10.76% on ARC-C/MathQA/GSM8K which is 5.04%/3.32%/10.01% over the improvement brought by Act-s. Furthermore, EoRA outperforms ApiQ on both ARC-C and MathQA by 2.9% and 0.44%. Overall, these findings underscore the effectiveness and generalizability of EoRA across different compression techniques.

A.8 Kernel Optimization, Inference Speed Evaluation, and Memory Overhead of EoRA



(a) (b)

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

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., 2025). 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 (a). 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.

Table 6: Comparison of the average per-token latency (batch size 1) for 128-token generation on LLaMA3-70B between full-precision and GPTQ + EoRA with and without our custom EoRA kernel.

Format	EoRA Rank	EoRA Kernel	Latency	Speedup
FP-16	-	-	60ms	1x
3-bit	-	-	35ms	1.7x
	64	No	52ms	1.2x
	64	Yes	44ms	1.4x
	128	No	54ms	1.1x
	128	Yes	43ms	1.4x
	256	No	58ms	1x
	256	Yes	48ms	1.3x
4-bit	-	-	38ms	1.6x
	64	No	60ms	1x
	64	Yes	49ms	1.2x
	128	No	61ms	1x
	128	Yes	51ms	1.2x
	256	No	63ms	1x
	256	Yes	53ms	1.1x

We show the inference speedup of our proposed EoRA kernel in Table 6. The first row shows the FP16 latency (60ms), followed by the 3-bit quantized-only model (35ms). The remaining rows present EoRA latencies at different ranks, both with and without our custom CUDA kernel, alongside 3-bit and 4-bit quantization. As shown in the table, 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 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. In short, our fused EoRA kernel significantly improves inference speed for both 3-bit and 4-bit quantized models. While there remains some overhead compared to the quantized-only baseline, even with our kernel, using no kernel results in significantly higher latency—sometimes exceeding FP16—due to activation movement overhead. These results underscore the practical deployability of EoRA when paired with our optimized kernel.

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 7, while the results for LLaMA3-8B are illustrated in Figure 2 (b). 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% (11.70 GB $\rightarrow$ 9.77 GB). 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% (9.12 GB $\rightarrow$ 9.28 GB) / 7% (9.12 GB $\rightarrow$ 9.77 GB). 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% (4.63 GB $\rightarrow$ 4.78 GB) / 14% (4.63 GB $\rightarrow$ 5.28 GB). 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., 2023). 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.

Table 7: Accuracy and the Model Size of quantizing EoRA of rank {128,512} to 4/3-bit on compensating LLaMA3-8B of {2:4 sparisity, 4/3-bit}.

Compression method	Config	r	W-bit of EoRA	Model Size (GB)	ARC-C ↑	MathQA ↑
-	-	-	-	15.08	50.42	40.10
SparseGPT	2:4	-	-	9.12	30.11	26.43
		128	16	9.77	34.64	29.91
			4	9.28	34.47	29.91
			3	9.24	34.72	29.71
		512	16	11.70	41.89	34.17
			4	9.77	41.46	33.63
			3	9.64	40.35	32.66
GPTQ	W4	-	-	5.35	45.90	34.07
		128	16	6.01	47.44	37.21
			4	5.50	47.35	36.78
			3	5.46	47.18	36.52
		512	16	7.85	48.29	38.72
			4	6.01	48.80	38.92
			3	5.90	46.92	36.88
	W3	-	-	4.63	20.90	22.37
		128	16	5.28	31.74	29.11
			4	4.78	31.48	28.64
			3	4.74	29.18	26.7
		512	16	7.16	38.82	31.89
			4	5.28	40.01	31.69
			3	5.18	35.4	30.45

A.9 Compensation With Different Ranks

Since one of the advantages of using EoRA is the greater flexibility in adjusting overall model accuracy without being constrained by specific compression formats, in this section, we investigate the influence of different ranks on adopting EoRA. We vary the rank of EoRA in {64,128,256,512} on compensating LLaMA3-8B pruned to 2:4 sparsity. As shown in Figure 3, EoRA consistently outperforms the two training-free baselines (ZeroQuant-V2 and Act-S) across all tested ranks, with the performance gap becoming more prominent at higher ranks. For instance, on GSM8K, EoRA achieves improvements of 7.43%, 10.69%, 11.9%, and 14.62% at ranks 64, 128, 256, and 512, respectively. In contrast, the gains on ARC-C remain relatively stable across ranks, ranging between 2% and 4%. Additionally, EoRA begins to outperform ApiQ on GSM8K at higher ranks, with improvements of 1.21% and 2.51% observed at ranks 256 and 512, respectively. These 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 8 in the appendix.

Table 8: Results of EoRA of different rank on compensating LLaMA2/3 models pruned to 2:4 sparsity by SparseGPT on Commonsense and Math reasoning tasks.

Model	Sparsity	r	Compensation Method	ARC-C ↑	MathQA ↑	GSM8K ↑
LLaMA3-8B	-	-	-	50.42	40.10	36.23
	2:4	-	-	30.11	26.43	2.12
		64	ZeroQuant-V2	30.97	26.39	2.27
			Act-S	30.46	26.67	3.34
			ApiQ	<u>33.10</u>	<u>27.87</u>	**11.52**
			EoRA	**33.10**	**28.57**	<u>10.77</u>
		128	ZeroQuant-V2	31.99	26.49	2.96
			Act-S	31.74	26.73	3.26
			ApiQ	<u>34.21</u>	<u>28.77</u>	**14.55**
			EoRA	**34.64**	**29.91**	<u>13.95</u>
		256	ZeroQuant-V2	34.55	28.74	4.09
			Act-S	32.76	27.94	5.16
			ApiQ	<u>35.41</u>	<u>30.45</u>	<u>15.85</u>
			EoRA	**37.96**	**31.59**	**17.06**
		512	ZeroQuant-V2	<u>38.73</u>	30.38	6.75
			Act-S	36.18	29.65	8.64
			ApiQ	36.69	<u>32.63</u>	<u>20.77</u>
			EoRA	**41.89**	**34.17**	**23.28**
LLaMA2-7B	-	-	-	39.84	27.67	14.85
	2:4	-	-	30.11	24.65	1.66
		64	ZeroQuant-V2	30.20	24.48	1.97
			Act-S	30.12	25.03	1.74
			ApiQ	<u>31.83</u>	<u>25.62</u>	**5.91**
			EoRA	**32.16**	**25.62**	<u>5.08</u>
		128	ZeroQuant-V2	30.54	24.89	1.97
			Act-S	30.20	25.09	2.73
			ApiQ	<u>32.67</u>	**26.36**	**7.58**
			EoRA	**32.67**	<u>25.59</u>	<u>6.22</u>
		256	ZeroQuant-V2	31.99	25.19	2.88
			Act-S	32.59	25.39	3.26
			ApiQ	<u>34.30</u>	<u>25.99</u>	**8.79**
			EoRA	**34.47**	**26.06**	<u>7.88</u>
		512	ZeroQuant-V2	34.72	24.38	3.34
			Act-S	34.73	25.76	3.56
			ApiQ	<u>34.98</u>	**26.16**	**9.70**
			EoRA	**36.77**	<u>25.96</u>	<u>8.79</u>
LLaMA2-13B	-	-	-	45.56	29.91	21.37
	2:4	-	-	34.30	25.92	2.65
		64	ZeroQuant-V2	33.95	25.56	2.81
			Act-S	32.76	25.93	2.96
			ApiQ	<u>35.84</u>	**27.17**	**8.64**
			EoRA	**36.00**	<u>26.80</u>	<u>8.19</u>
		128	ZeroQuant-V2	33.61	25.12	3.56
			Act-S	34.12	25.69	4.09
			ApiQ	<u>36.68</u>	<u>27.16</u>	**12.13**
			EoRA	**37.54**	**27.53**	<u>10.91</u>
		256	ZeroQuant-V2	35.06	26.06	4.93
			Act-S	34.56	26.23	4.62
			ApiQ	<u>36.69</u>	<u>27.40</u>	**14.56**
			EoRA	**38.73**	**27.77**	<u>13.04</u>
		512	ZeroQuant-V2	36.51	26.39	7.28
			Act-S	36.86	26.77	6.14
			ApiQ	<u>38.57</u>	<u>27.71</u>	<u>17.21</u>
			EoRA	**40.61**	**29.17**	**17.51**

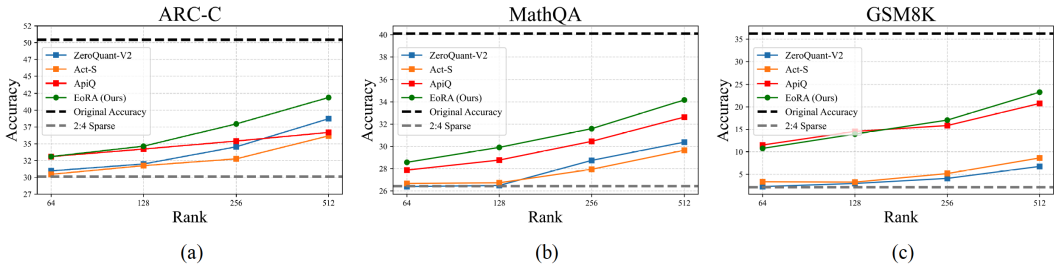


Figure 3: Results of applying EoRAand other baselines with rank set to {64,128,256,512} to improve LLaMA3-8B models pruned to 2:4 sparsity by SparseGPT on (a) ARC-C/(b) MathQA/(c) GSM8K.

A.10 Influence of Different Calibration sizes on EoRA

Table 9: Ablation studies of calibrating EoRA with different calibration sizes on compensating compressed LLaMA3-8B.

Model	Quantization Format	#Calib.	Calib. Time (mins)	MathQA ↑
LLaMA3-8B	FP16	-	-	40.10
	W4	16	6.40	36.62
		32	7.04	36.93
		64	8.03	37.21
		128	10.40	37.46
		256	14.43	37.60
		512	21.11	37.30
	W3	16	6.33	26.33
		32	7.20	27.57
		64	8.16	29.11
		128	11.33	30.34
		256	14.17	30.21
		512	20.89	30.40

We conducted ablation studies to assess how different calibration set sizes—{16, 32, 64, 128, 256, 512}—affect EoRA's performance in compensating for 4-bit and 3-bit quantized LLaMA3-8B models on MathQA. As shown in Table 9, increasing the number of calibration samples from 16 to 256 yields a moderate improvement in accuracy (from 36.62 to 37.60) for the W4 model. However, further increasing the calibration size to 512 leads to a slight decline, indicating that the accuracy gain saturates beyond a certain threshold. A similar trend is observed for the W3 model, where performance improves steadily up to 128 samples, after which the benefit plateaus. These findings suggest that while EoRA can leverage additional calibration data to improve accuracy, its performance remains stable and robust even with limited calibration.

A.11 EoRA as LoRA initialization for Fine-tuning Compressed Models

We show that, with additional computational resources, users can leverage the low-rank matrices from EoRA as initialization for LoRA fine-tuning, enabling further accuracy improvements for 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 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 following LoftQ (Li et al., 2024), and standard initialization following QLoRA (Dettmers et al., 2023). Fine-tuning is performed on the ARC training set for evaluating ARC-C, and on the MathQA training set for the math reasoning task. 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 10, initializing with EoRA substantially enhances the accuracy of compressed models, surpassing both QLoRA and LoftQ when fine-tuning 4-bit quantized LLaMA3-8B, and achieving accuracy on par with standard full-precision fine-tuning. We also observed that the improvements over QLoRA and LoftQ are more pronounced on 3-bit quantized and 2:4 pruned models, aligning with our earlier finding that EoRA is more effective when the compression error is more substantial, as shown in Table 10.

Table 10: Fine-tune the compressed LLaMA3-8B models with various compression settings and different initialization of the low-rank matrices for Commonsense/Math reasoning tasks.

Model	Compression Method	Compression Setting	LoRA initialization	ARC-C ↑	MathQA ↑
LLaMA3-8B	Full-precision	-	w/o finetuning	50.42	40.10
			Standard	56.39	53.56
	SparseGPT	2:4	w/o finetuning	30.11	26.43
			QLoRA	41.30	45.42
			LoftQ	43.68	48.77
			EoRA	**48.54**	**54.67**
	GPTQ	W4	w/o finetuning	45.90	34.07
			QLoRA	54.09	51.42
			LoftQ	54.52	53.96
			EoRA	**55.46**	**56.04**
	GPTQ	W3	w/o finetuning	20.90	22.37
			QLoRA	30.29	34.10
			LoftQ	44.70	48.17
			EoRA	**47.44**	**53.90**

A.12 EoRA on More Tasks

Table 11: Comparison of 4-bit and 3-bit quantized LLaMA3-8B on the LLM summarization task.

Model	Quantization Format	Compensation Method	CNN/DailyMail (ROUGE-Lsum) ↑
LLaMA3-8B	4-bit	-	0.1672
		ZeroQuant-V2	0.1798
		Act-S	0.1786
		ApiQ	0.1804
		EoRA	**0.1812**
	3-bit	-	0.0650
		ZeroQuant-V2	0.0970
		Act-S	0.1286
		ApiQ	0.1357
		EoRA	**0.1463**

LLM Summarization. We evaluate EoRA and baseline methods on restoring the summarization capability of quantized models using the testset of CNN/DailyMail — a widely used English-language corpus containing over 300k news articles authored by CNN and Daily Mail journalists. This dataset supports both extractive and abstractive summarization, and we adopt ROUGE-Lsum as the evaluation metric, where higher values indicate better summary quality. We set the rank to 128 for all the methods. For Act-S, ApiQ, and EoRA, we use 128 calibration sentences from WikiText2. The results are shown in Table 11. Notably, EoRA consistently outperforms all baselines across both 4-bit and 3-bit quantized LLaMA3-8B models. Specifically, it improves the ROUGE-Lsum score from 0.1672 to 0.1812 in the 4-bit setting, and from 0.0650 to 0.1463 in the 3-bit setting—demonstrating substantial recovery of summarization performance. These results highlight that EoRA remains effective and competitive in practical, real-world scenarios such as summarization.

Table 12: Comparison of 4-bit and 3-bit quantized LLaMA3.2-3B on multi-task language understanding.

Model	Quantization Format	EoRA Rank	MMLU ↑
LLaMA3.2-3B	FP16	-	54.19
	4-bit	-	24.16
		32	52.53
		64	52.49
		128	52.93
	3-bit	-	22.89
		32	39.08
		64	38.83
		128	39.68

Language Understanding. We tested EoRA on 4-bit quantized LLaMA-3.2-3B, varying the rank from 8 to 128. Using 128 WikiText2 calibration samples, we observe substantial recovery—even rank 8 lifts accuracy from 24.16 to 52.20, as shown in Table 12. This result highlights both the effectiveness of EoRA in compensating for quantization errors in smaller-scale models and its robustness across a range of low-rank settings.

A.13 GPTQ: Channel-wise vs. Group-wise Quantization

Table 13: Comparison of 4-bit and 3-bit group-wise quantized LLaMA3-8B. The group size is set as 128. EoRA rank is set as 128.

Model	Quantization Format	Compensation Method	MathQA ↑
LLaMA3-8B	FP16	-	40.10
	W4-groupsize 128	-	38.34
		ZeroQuant-V2	38.92
		Act-S	38.49
		ApiQ	38.90
		EoRA	**39.16**
	W3-groupsize 128	-	32.52
		ZeroQuant-V2	32.39
		Act-S	33.33
		ApiQ	34.80
		EoRA	**35.10**

We initially adopt channel-wise quantization for GPTQ to remain consistent with the original GPTQ setup. More importantly, our goal is to showcase the robustness of EoRA in recovering from even severe quantization errors. That said, we conduct additional experiments using group-wise quantization (group size = 128) to evaluate EoRA's effectiveness in this more commonly used setting. Table 13 reports results on MathQA for 4-bit and 3-bit group-wise quantized LLaMA3-8B models, referred to as W4-groupsize 128 and W3-groupsize 128, respectively. We compare EoRA against prior compensation methods and set the rank to 128. These results highlight EoRA's consistent advantage over existing methods in compensating for group-wise quantization. In particular, for the W4-group size 128 configuration, EoRA is able to recover nearly all of the lost accuracy, achieving performance comparable to the original full-precision model.

A.14 Layer-wise Discrepancy Analysis

Table 14: Comparison of layer-wise discrepancy on q projector of LLaMA2-7B.

Method	Layer 0	Layer 5	Layer 10	Layer 15	Layer 20	Layer 25	Layer 30
GPTQ (4-bit)	2.3	11.3	13.4	12.6	10.7	10.9	11.5
ZeroQuant-V2	1.7	10.2	10.8	11.3	9.7	9.2	9.9
Act-S	2.0	9.7	9.4	10.3	8.4	7.7	8.9
ApiQ	1.5	7.8	8.2	7.4	6.2	5.9	5.1
EoRA	1.4	8.5	8.1	7.2	5.8	6.0	5.4

We follow the layer-wise discrepancy analysis setting of the ApiQ (Liao et al., 2024) and run the analysis on the q projector of LLaMA2-7B. We compare the discrepancy of layers [0, 5, 10, 15, 20, 25, 30] of different compensation methods of rank 128 and show that EoRA effectively reduces layer-wise discrepancy compared to existing baselines, as shown in Table 14. Notably, EoRA maintains consistently low output activation errors across all layers, especially in the later ones (e.g., 5.8 at layer 20 and 5.4 at layer 30), showing significant improvements over ZeroQuant-V2 and Act-S. EoRA also matches or outperforms ApiQ, demonstrating its effectiveness. Importantly, EoRA accomplishes this with dramatically less compute overhead—typically completing in just a few minutes—whereas ApiQ often requires several hours of optimization. This highlights EoRA's practical advantage as a highly efficient solution for layer-wise error minimization for recovering the error of low-bit quantization.

Table 15: Comparison of 4-bit and 3-bit quantized LLaMA3-8B for more recent low-rank approaches.

Model	Quantization Format	Compensation Method	MathQA ↑
LLaMA3-8B	-	-	40.10
	4-bit	-	34.07
		FWSVD	35.64
		ZeroQuant-V2	36.51
		Act-S	35.84
		ApiQ	36.18
		LQER	35.46
		LRC	36.40
		CALDERA	36.70
		QERA	35.90
		SLiM	35.90
		OATS	36.01
		EoRA	**37.21**
	3-bit	-	22.37
		FWSVD	26.30
		ZeroQuant-V2	26.43
		Act-S	25.42
		ApiQ	26.86
		LQER	25.60
		LRC	28.64
		CALDERA	28.10
		QERA	25.32
		SLiM	25.91
		OATS	25.30
		EoRA	**29.11**

A.15 More Comparisons with Recent Low-Rank Approaches

Recent Low-Rank approaches can be broadly categorized into three groups: 1) activation-statistics-based scaling methods, including SLiM (Mozaffari et al., 2025a), OATS (Zhang & Papyan, 2025), LQER (Zhang et al., 2024), and QERA (Zhang et al., 2025), 2) iterative low-rank compensation approaches, which are LRC (Scetbon & Hensman, 2024) and CALDERA (Saha et al., 2024), and 3) training-based methods, including LR-QAT (Bondarenko et al., 2024) and SLoPe (Mozaffari et al., 2025b).

The first group—SLiM, OATS, LQER, and QERA—scales the compression error based on activation statistics prior to applying SVD for low-rank approximation. The underlying intuition is that input channels with higher magnitudes (i.e., outliers) are considered more important, and as a result, their corresponding weight entries should be prioritized over those linked to lower-magnitude activations. All four methods differ slightly in how the scaling diagonal matrix is constructed, but they are fundamentally similar to the ASVD-based scaling baseline already discussed in our paper (see Section A.3). For example, LQER (Zhang et al., 2024) mitigates quantization error by scaling the residual using activation statistics—specifically, the maximum average magnitude per input channel—before applying SVD. Another example, SLiM (Mozaffari et al., 2025a), introduces a

low-rank compensation method that also leverages activation statistics (specifically, average absolute activation values) to scale residuals. Although SLIM is proposed alongside a custom quantization scheme, we isolate and evaluate its saliency-based low-rank adapter strategy. Although these methods vary in how they scale the compression error, none of them are guaranteed to minimize the layer-wise compression error directly—an issue we discussed in Section A.1 of our paper. In contrast, EoRA is explicitly formulated to minimize this objective (as detailed in Section A.2).

The second group, including iterative low-rank compensation approaches, mainly lack the flexibility as EoRA. For example, LRC (Scetbon & Hensman, 2024) requires iterative updates to weights and low-rank modules, leading to task-specific quantized models. In contrast, EoRA only adapts the low-rank modules, allowing a shared compressed backbone and easier integration with multi-adapter frameworks (The vLLM Team). Although LRC offers a closed-form solution, it assumes is full-rank—an assumption that often fails and requires extra modification steps that may introduce noise and instability. EoRA avoids this by only requiring to be symmetric, improving numerical stability and robustness. Another example, CALDERA (Saha et al., 2024), employs an iterative optimization strategy that updates both the quantized weights and the low-rank matrices, using a closed-form solution. However, because it requires modifying the quantized weights during this process, CALDERA is less efficient for multi-task scenarios, where separate quantized models would need to be maintained for each task. EoRA, on the other hand, avoids this limitation by making only the low-rank components task-specific, while keeping the quantized backbone fixed and shared across tasks. This decoupled design allows for easy integration with existing multi-adapter inference frameworks (The vLLM Team) and significantly improves the practicality of EoRA for real-world applications.

The third group requires gradient-based training. For example, LR-QAT (Bondarenko et al., 2024) is a quantization-aware training (QAT) method. While it also uses low-rank modules, it operates in the fine-tuning regime, combining cross-entropy loss with final-layer output alignment (akin to knowledge distillation). SLoPe (Mozaffari et al., 2025b) primarily targets improving pre-training efficiency and enhancing the accuracy of compressed models through training. As such, it aligns more closely with quantization-aware training (QAT) as well. In contrast, EoRA is a post-training quantization (PTQ) method that requires no gradient updates to the quantized model. In the LLM compression community (Frantar et al., 2023; Lin et al., 2024), it is standard practice to distinguish between PTQ and QAT approaches, as they serve different purposes and are not typically benchmarked against each other.

We summarize the empirical comparison on 4-bit and 3-bit quantized LLaMA3-8B models evaluated on MathQA in Table 15. All methods use rank of 128 and identical calibration data (see Section A.3). To fairly compare CALDERA with EoRA, we adapt the CALDERA method by fixing the quantized weights and only updating the low-rank matrices. This degenerates the iterative process into a two-step approximation: first approximate the down-projection matrix, followed by the up-projection. Once the quantized weights are fixed, additional iterations do not change the approximation.

As the results show, EoRA consistently outperforms other low-rank compensation methods, particularly those based on activation statistics like LQER, SLiM, etc. This highlights the advantage of EoRA's mathematical property, which directly minimizes the layer-wise compression error rather than relying on heuristics derived from activation statistics. While LRC offers better accuracy than heuristic scaling approaches, it still lags behind EoRA due to its reliance on less stable approximations. These findings highlight EoRA's practicality and effectiveness for post-training quantized LLMs. From the results, we observe that CALDERA performs well—outperforming both Act-S and ApiQ—but EoRA still consistently achieves higher accuracy. We attribute this to EoRA's single-step optimization that jointly solves for both projection matrices, whereas CALDERA's sequential two-step process may accumulate slightly more approximation error.

Overall, EoRA consistently surpasses all the recent low-rank methods, including both activation-statistics-based approaches and iterative low-rank compensation approaches. Its effectiveness stems from its ability to directly minimize layer-wise compression error, eliminate the need for heuristic magnitude-based scaling, and utilize a single-step optimization process. These strengths underscore both the theoretical robustness and practical efficiency of EoRA compared to existing methods.

Table 16: Comparison of 2-bit GPTQ-quantized LLaMA3-8B on MathQA.

Model	Quantization Format	Fine-tuning Strategy	MathQA ↑
LLaMA3-8B	2-bit (GPTQ)	-	18.22
		LoftQ	35.80
		EoRA	36.89
		RILQ	37.60
		RILQ + EoRA	**38.90**

A.16 EoRA on 2-bit Quantization

We also evaluate a more challenging setting, 2-bit quantization, where RILQ (Lee et al., 2025) is one of the state-of-the-art error compensation methods. RILQ adopts standard backpropagation (i.e., fine-tuning), utilizing a combination of cross-entropy loss and final-layer output alignment (similar to knowledge distillation). As outlined in the RILQ paper, RILQ uses gradient descent to collectively tune all adapters, minimizing the discrepancy between full-precision and quantized activation outputs of the final layer. In addition, as quoted from the RILQ paper, RILQ also incorporates a causal language modeling objective with Ground Truth in the optimization of low-rank adapters. Therefore, it is not appropriate to directly compare EoRA to RILQ as methods in the same category. However, since EoRA can act as an effective initialization for subsequent fine-tuning (as detailed in Section A.11), it is complementary to RILQ rather than competing with it. To examine this synergy, we performed experiments where EoRA was first used for initialization—following the setup outlined in Section A.11—and then applied RILQ's fine-tuning objective to a 2-bit GPTQ-quantized LLaMA3-8B model on the MathQA dataset. The results are shown in Table 16. As expected, RILQ outperforms LoftQ in compensating for quantization error in 2-bit models. Moreover, when initialized with EoRA, further fine-tuning with RILQ's objective yields an additional 1.3% improvement over RILQ alone.

A.17 Related Works

Post-training LLM Compression: As LLMs scale, reducing their size is essential for efficient deployment. Traditional compression-aware training methods are impractical due to the need for full datasets and heavy retraining. Post-training compression methods like quantization and pruning have gained popularity as they require only minimal calibration data and no retraining. PTQ reduces model size by lowering bitwidths (Frantar et al., 2023; Tseng et al., 2024), while PTP removes less important weights to reduce computation (Frantar & Alistarh, 2023; Sun et al., 2024). Our method, EoRA, is compatible with all such compression techniques as it operates independently of the base method used.

Low-Rank Decomposition: Low-rank decomposition methods (Yuan et al., 2023; Sakr & Khailany, 2024; Wang et al., 2025; Hsu et al., 2022; Mozaffari et al., 2025a; Zhang & Papyan, 2025; Zhang et al., 2024; 2025; Saha et al., 2024) compress models by replacing weights with low-rank matrices, reducing both latency and size without special kernel support. However, they are less widely adopted due to weaker accuracy-compression trade-offs. While FWSVD (Hsu et al., 2022) is designed primarily as a compression method based on low-rank decomposition of model weights, EoRA instead leverages low-rank modules specifically to compensate for compression errors. Unlike FWSVD, EoRA provides a theoretical guarantee of minimizing layer-wise compression loss, as demonstrated in our derivation in Section A.2. Using gradient-based information, as required by FWSVD, can be prohibitively expensive for LLMs, as noted in ASVD (Yuan et al., 2023). SVD-LLM (Wang et al., 2025) is conceptually close to our work, which also tries to align the SVD compression error with the layer-wise compression loss. It relies on the matrix product of the activation being positive-definite, a condition often unmet in practice. Enforcing this condition typically requires additional modifications, which introduce noise into the approximation. In contrast, EoRA employs eigendecomposition, which only requires the matrix product of the activation to be symmetric—a property that naturally holds—avoiding such issues. Furthermore, although EoRA also utilizes

SVD-based low-rank decomposition, its core objective is fundamentally different. Whereas prior methods aim to replace pre-trained weight matrices with low-rank approximations to reduce model size and inference cost, EoRA focuses on approximating the compression error itself. This allows for improved accuracy recovery in compressed LLMs and provides greater flexibility in balancing accuracy and computational overhead by overcoming the constraints of fixed compression formats. Please refer to Section A.15 in Appendix for more detailed comparison.