ROSAQ: Rotation-based Saliency-Aware Weight Quantization for Efficiently Compressing Large Language Models

Anonymous ACL submission

Abstract

Quantization has been widely studied as an effective technique for reducing the memory requirement of large language models (LLMs), potentially improving the latency time as well. Utilizing the characteristic of *rotational invariance* of transformer, we propose the rotation-based saliency-aware weight quantization (**ROSAQ**), which identifies salient channels in the projection feature space, not in the original feature space, where the projected "principal" dimensions are naturally considered as "salient" features. The proposed ROSAQ consists of 1) *PCA-based projection*, which first performs principal component analysis (PCA) on a calibration set and transforms via the PCA projection, 2) *Salient channel identification*, which selects dimensions corresponding to the K-largest eigenvalues as salient channels, and 3) *Saliency-aware quantization* with mixed-precision, which uses FP16 for salient dimensions and INT3/4 for other dimensions. Experiment results show that ROSAQ shows improvements over the baseline saliency-aware quantization on the original feature space and other existing quantization methods. With kernel fusion, ROSAQ presents about 2.3x speed up over FP16 implementation in generating 256 tokens with a batch size of 64. Our code is publicly available in https://www.github.com/XXX

1 Introduction

Large language models (LLMs) have shown remarkable performances on various NLP tasks, however, due to their huge sizes of model parameters, LLMs require significant GPU memory for inference, substantially limiting the throughput and latency time. To address these challenges, *quantization* methods (Yao et al., 2022; Dettmers et al., 2022, 2023; Wu et al., 2023b; Yao et al., 2023; Kim et al., 2024a) have been widely studied as an effective technique for reducing the memory requirement of LLMs, potentially improving the latency time as well, by representing weights and activations of LLMs using low-precision.

In quantization, one of the most challenging issues is the presence of *outliers* in weights and activations, as they widen the quantization range and increase quantization error. Recently, leveraging the *rotational invariance* of transformers (Liu et al., 2024b), rotation-based quantization has been extensively applied to mitigate outliers, motivated by the observation that outliers are reduced after rotation. Similarly, SmoothQuant (Xiao et al., 2023) exploits the *scaling invariance* of linear layer, by dividing activation values by channel-specific scaling factors, thus greatly reducing activation outliers without severely strengthening weight outliers. The scaling invariance is further utilized in activation-aware quantization (AWQ) (Lin et al., 2024), which primarily focuses on "salient channels" to reduce quantization errors, by identifying salient channels based on activation magnitude.

Without being limited in the original feature space as in (Lin et al., 2024), this paper extensively explores the rotational invariance for *saliency-aware weight quantization* by identifying salient channels based on "principal dimensions on the projection space," thereby proposing the rotation-based saliency-aware weight quantization (**ROSAQ**). By the definition, the principal dimensions resulting from the principal component analysis (PCA) maximize the variances of channel values on projected space, and accordingly substantially increase their activation magnitudes. Our key underlying expectation is that these principal channels with the largest eigenvalues are more dominant and salient than the existing magnitude-based salient channels in original space, due to their inherent properties of maximizing variance, thereby further improving the saliency-aware quantization. The proposed ROSAQ consists of three steps:

- **PCA-based projection**, which first performs

PCA projection with its eigenvectors on a calibration set to obtain the PCA-projected calibration set. For the multi-head self-attention (MHSA) layer, we further propose the use of *head-wise* PCA, where the PCA projection is applied separately to each head-specific attention representation.

- **Salient channel identification**, which selects "principal channels" corresponding to the K-largest eigenvalues as 'salient' channels, and regards the other channels as normal non-salient channels.

- **Saliency-aware quantization with mixed-precision**, which applies the *per-group* quantization, where employs FP16 for a salient group of channels, and INT3/4 for all other groups of non-salient channels, where a group consists of 128 channels.

Experiment results on Wikitext2, zero-shot common-sense reasoning, and zero-shot MMLU tasks show that the proposed ROSAQ leads to improvements over the baseline saliency-aware quantization with mixed-precision on original feature space and the existing quantization methods, with minimal performance degradation. Furthermore, with kernel fusion, ROSAQ exhibits about 2.3x speed up over FP16 implementation when generating 256 tokens with a batch size of 64, and about 2x speedup when generating 128 tokens with a batch size of 128.

Our contributions are summarized as follows: 1) we propose ROSAQ, which is a novel rotation-based saliency-aware quantization, by choosing principal channels resulting from the PCA projection as salient ones, 2) we apply the head-wise PCA projection across multiple heads for quantizing the parameters of MHSA, and 3) the proposed ROSAQ leads to improved performances over existing quantization on Wikitext2, zero-shot common-sense reasoning, and zero-shot MMLU tasks.

2 Related Work

Quantization methods have been studied mainly two categories – quantization-aware training (QAT) (Liu et al., 2023; Shen et al., 2024; Ma et al., 2024) and post-training quantization (PTQ) (Pan et al., 2023; Tseng et al., 2024; Wu et al., 2023a; Guan et al., 2024; Yao et al., 2024; Liu et al., 2024a; Dettmers et al., 2024; Shao et al., 2024). PTQ

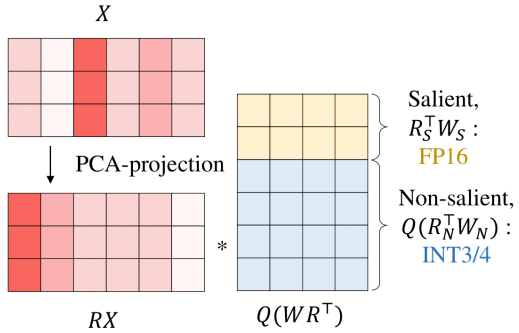

Figure 1: An overview diagram of ROSAQ that quantizes the weights of a linear layer $\mathbf{XW}$, using rotational invariance as described by Eq. (1), where $\mathbf{X}$ is the calibration data matrix. ROSAQ first applies the PCA-based projection, taking $\mathbf{Q}$ as $\mathbf{R}$, with eigenvectors obtained from Eq. (4). The salient channels denoted as $\mathbf{W}_S$, corresponding to the K largest eigenvalues, are represented in FP16, while the remaining non-salient channels $\mathbf{W}_N$ are represented in low precision, such as INT3/INT4.

is widely applied because it requires no (or minimal) training and only needs to use a small calibration set. While the "rotational invariance" has previously applied for the language model pruning (Ashkboos et al., 2024a; Hu et al., 2024), the rotation-based quantization has been extensively studied to reduce outliers, including incoherence processing based on orthogonal projections (Chee et al., 2024; Ashkboos et al., 2024b), and optimizing the rotation matrix (Liu et al., 2024b) based on Cayley SGD(Li et al., 2020).

Saliency-aware quantization has also been proposed by AWQ (Lin et al., 2024), which selects salient channels based on activation magnitudes but uses "full low-precision" quantization, leveraging the "scaling invariance" property of linear layers without mixed precision.

Unlike rotation-based quantization methods such as (Liu et al., 2024b), which aim to remove outliers, ROSAQ applies rotation to more effectively identify salient channels. Instead of the scaling invariance used in AWQ, ROSAQ exploits rotational invariance for salient channel identification.

3 Method: ROSAQ

Figure 1 presents the overall diagram of ROSAQ. Suppose that a calibration set consists of N data samples, formally presented as $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times d}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is i-th data representation. ROSAQ uses the rotational invariance for all linear layers formed with $\mathbf{XW}$ in transformer, where $\mathbf{W} \in \mathbb{R}^{d \times d'}$ is a weight matrix

of parameters, formulated as follows:

$$\mathbf{XW} = (\mathbf{XR})\left(\mathbf{R}^T\mathbf{W}\right) \quad (1)$$

where $\mathbf{R} \in \mathbb{R}^{d\times d}$ is a *rotation* matrix, which consists of orthonormal vectors.

After applying the weight quantization, Eq. (1) is approximated by:

$$\mathbf{XW} \approx (\mathbf{XR})\, Q\left(\mathbf{R}^T\mathbf{W}\right) \quad (2)$$

where Q is a quantization function, which adopts *per-group* quantization with a group size of 128 channels.

Similar to AWQ (Lin et al., 2024), ROSAQ also takes into account the assumption that weights are not equally salient. Different from AWQ which applies the quantization to all channels, but in a scale-sensitive manner, ROSAQ deploys the mixed-precision which keeps high-precision for salient channels while using low-precision non-salient channels, in order to minimize the quantization error particularly for salient channels. To formally present the mixed-precision, suppose the column vectors of $\mathbf{R}$ are sorted by their saliency degrees, and they are then divided into two groups – salient and non-salient groups of channels, $\mathbf{R} = [\mathbf{R}_S, \mathbf{R}_N]$, where $\mathbf{R}_S \in \mathbb{R}^K$ is the orthonormal vectors for salient channels, and $\mathbf{R}_N \in \mathbb{R}^{d-K}$ is one for non-salient channels. Under the mixed-precision, Eq. 2 is approximated by:

$$\mathbf{XW} \approx (\mathbf{XR}) \begin{bmatrix} \left(\mathbf{R}_S^T\mathbf{W}_S\right) \\ Q\left(\mathbf{R}_N^T\mathbf{W}_N\right) \end{bmatrix} \quad (3)$$

where $\mathbf{W}_S \in \mathbb{R}^{K\times d'}$ is the sub-block of the weight matrix for salient channels, and $\mathbf{W}_N \in \mathbb{R}^{(d-K)\times d'}$ is one for non-salient channels.

ROSAQ consists of three steps — 1) PCA-based projection, 2) Salient channel identification, and 3) Saliency-aware quantization with mixed-precision, which will be presented in the next subsections with more details.

3.1 PCA-based projection for Computing R

To obtain the rotation matrix $\mathbf{R}$ in Eq. 3, we perform PCA on the calibration set $\mathbf{X}$ as follows:

$$\mathbf{X}^T\mathbf{X} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T \quad (4)$$

where $\mathbf{R} \in \mathbb{R}^{d\times d}$ is the eigenvectors of $\mathbf{X}^T\mathbf{X}$, and $\mathbf{\Lambda}$ is the corresponding eigenvalue matrix. Without loss of generality, we assume that the column vectors of $\mathbf{R}$ are sorted by their eigenvalues.

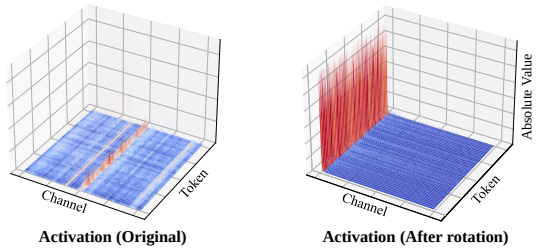

Figure 2: Magnitude of the input activation values to MHSA in LLaMA2-7B, before and after PCA-based rotation. Salient channels are more dominant in PCA-projected space than the original space. Detailed statistics are presented in Appendix F.

To check whether the PCA projection helps to identify salient channels, Fig 1 presents the activation magnitudes across all channels in both the original and PCA-projected feature spaces. It is seen that the activation values for salient channels are more dominant, making them easier to distinguish compared to those in the original feature space.

3.1.1 Head-wise PCA Projection in MHSA

In general, ROSAQ uses the layer-wise PCA projection, which is applied individually to the activation matrix $\mathbf{X}_l$ in a layer-specific manner, for each linear layer represented by $\mathbf{X}_l\mathbf{W}_l$, thereby resulting in its own rotation matrix $\mathbf{R}_l$ for each layer l. To better capture the head-specific characteristics for quantization, ROSAQ deploys a head-wise PCA projection for MHSA, where a separate PCA is performed for each head-specific attentive representation. More specifically, suppose that $\mathbf{H}_h \in \mathbb{R}^{m\times d_h}$ is h-th head-specific attentive representation resulting from the activation matrix $\mathbf{Z}_{l-1}$ at the previous layer, as follows.

$$\mathbf{H}_h = \text{Attention}\left(\mathbf{Z}_{l-1}\mathbf{W}_h^Q, \mathbf{Z}_{l-1}\mathbf{W}_h^K, \mathbf{Z}_{l-1}\mathbf{W}_h^V\right) \quad (5)$$

where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d\times d_h}$ are the weight matrices at h-th head for query, key, and value parts, respectively.

Instead of applying a global PCA on the concatenated multi-head representation concat $(\mathbf{H}_1, \cdots, \mathbf{H}_H)$, we approximate MHSA by using the head-specific PCA projection as follows:

$$\text{MHSA}\left(\mathbf{Z}_{l-1}\right) \approx \sum_{h=1}^{H} \left(\mathbf{H}_h\mathbf{R}_h\right) Q\left(\mathbf{R}_h^T\mathbf{W}_h^O\right) \quad (6)$$

where $\mathbf{R}_h$ is a head-specific PCA projection matrix, which consists of eigenvectors obtained by

Model	Method	PPL (↓)	CSR Avg. (↑)	MMLU (↑)
Llama3-8B	FP16	6.14	68.31	62.10
	GPTQ	*6.50	67.03	59.86
	SpinQuant	6.71	66.20	59.00
	AWQ	6.53	67.34	60.48
	Mixed	6.63	67.28	59.80
	ROSAQ	**6.50**	**67.72**	**60.97**
Qwen2-7B	FP16	7.14	68.92	69.42
	GPTQ	7.30	68.76	67.26
	AWQ	7.33	68.74	68.79
	Mixed	7.39	68.29	67.81
	ROSAQ	**7.29**	**68.87**	**68.82**

Table 1: Comparison results between ROSAQ and other quantization methods on LLaMA3-8b and Qwen2-7b models, when group-wise quantization is applied using INT4 with a group size of 128 (i.e., INT4 g128). PPL indicates the perplexity score on WikiText2, CSR and MMLU refer to the averaged zero-shot accuracies on zero-shot common sense reasoning and MMLU tasks, respectively. * results were quoted from AWQ (Lin et al., 2024). More detailed results are provided in Appendix E.

applying PCA on the head-specific calibration set $\mathbf{X}_h \in \mathbb{R}^{N \times d_h}$ for h-th head, and $\mathbf{W}_h^O \in \mathbb{R}^{d_h \times d}$ is the output projection matrix. In Appendix C, we present that the use of head-specific PCA leads to the decrease of perplexity, comparing to the global PCA, as in Table 3.

3.2 Salient channel identification

To identify salient channels on the PCA-based projection space, we sort projected channels according to their eigenvalues, and select a group of channels corresponding to the larger eigenvalues as salient ones. In Appedix F, Table 8 shows that the top-ranked channels also tend to have the largest average magnitudes.

3.3 Saliency-aware quantization with mixed-precision channel

After splitting weights into two groups – $\mathbf{W}_S$ and $\mathbf{W}_N$ – salient and non-salient channels, we retain FP16 precision for the salient group, while applying quantization under the INT3/INT4 to the non-salient group. Detailed settings can be found in Appendix G.

4 Experiments

4.1 Experimental Setup

We apply per-group weight quantization under an INT3/INT4, where each group consists of 128 channels. Similar to AWQ(Lin et al., 2024), we use a small calibration set from the Pile dataset(Gao et al., 2020) to prevent overfitting to any specific downstream domain. For LLMs, we employ open-source models, including the LLaMA2 and LLaMA3 (Touvron et al., 2023; AI@Meta, 2024) and Qwen-2 (Yang et al., 2024) model families.

To evaluate the quantized LLMs, we report perplexity (PPL) on the WikiText-2 dataset(Merity et al., 2016) and use standard evaluation metrics for zero-shot Common Sense Reasoning tasks and zero-shot MMLU benchmark(Hendrycks et al., 2021).

We compare ROSAQ with the existing quantization methods, such as **GPTQ**, (Frantar et al., 2023), **SpinQuant** (Liu et al., 2024b), and **AWQ** (Lin et al., 2024). We also report the "rotation-less" baseline run, denoted as **Mixed** which select salient channels according to the activation magnitudes on the original feature space (i.e., $\mathbf{R} = \mathbf{I}$ in Eq. (3)).

4.2 Main Results

Table 1 presents the performances of quantized models in INT4, in terms of the perplexity on WikiText2 and the zero-shot accuracies on Common Sense Reasoning tasks and zero-shot MMLU.

As seen in Table 1, ROSAQ exhibits slightly superior to GPTQ(Frantar et al., 2023), SpinQuant(Liu et al., 2024b) and AWQ(Lin et al., 2024) except perplexity. Notably in Table 6 which is more aggressive 3bit setting, ROSAQ outperforms other methods achieving the highest MMLU results across the various categories.

Detailed results are presented in Appendix E, D. In particular, Table 4 compares inference throughputs, reporting that ROSAQ achieves approximately 2.3x speedup over FP16 when generating 256 tokens with a batch size of 64.

5 Conclusion

In this paper, we proposed ROSAQ, a novel saliency-aware quantization method that uses PCA-based rotation to identify salient channels and preserve them in a high-precision. Experimental results on LLaMA and Qwen-2 models demonstrate that ROSAQ achieves promising outcomes in most cases. In the future, we aim to further generalize rotational invariance by incorporating scaling factors for selecting salient channels. We also would like to extend saliency-aware quantization to both weights and activations, thereby optimizing the efficiency of the retrieval-augmented generation (RAG) (Lewis et al., 2020).

Limitations

In ROSAQ, we focus solely on weight quantization, however, weight-activation quantization is often important, especially for retrieval-augmented generation (RAG). Additionally, while we currently use mixed precision by retaining FP16 for salient channels, it would be valuable to explore the use of fully low-precision formats, as in AWQ (Lin et al., 2024), by further generalizing the rotational invariance applied in our approach.

While ROSAQ achieve better performance by explicitly separating salient weights, but hardware inefficiencies still remain. In AWQ (Lin et al., 2024), they fully address the mixed-precision issue by quantizing all weights to 4 bits, making the approach more hardware-friendly. Considering the use of mixed precision, we may need to explore more hardware-friendly approaches to support mixed precision, or generalize the proposed framework by quantizing all weights into low-precision formats.

We explore rotational invariance to select salient features; however, further analysis is needed to understand how this saliency-aware quantization relates to other rotation-based approaches for mitigating outliers. Both theoretical and empirical investigations are required to establish a connection between saliency-aware quantization and outlier-free quantization when rotational operations are adopted.

Acknowledgments

References

AI@Meta. 2024. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. Accessed: 2024-09-15.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari Do Nascimento, Torsten Hoefler, and James Hensman. 2024a. Slicegpt: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024b. Quarot: Outlier-free 4-bit inference in rotated llms. *Preprint*, arXiv:2404.00456.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip: 2-bit quantization of large language models with guarantees. *Preprint*, arXiv:2307.13304.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Preprint*, arXiv:2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2024. Spqr: A sparse-quantized representation for near-lossless LLM weight compression. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. 2024. APTQ: attention-aware post-training mixed-precision quantization for large language models. *CoRR*, abs/2402.14866.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Yuxuan Hu, Jing Zhang, Zhe Zhao, Chen Zhao, Xiaodong Chen, Cuiping Li, and Hong Chen. 2024. sp^3: Enhancing structured pruning via PCA projection. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3150–3170, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024a. Squeezellm: Dense-and-sparse quantization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Taesu Kim, Jongho Lee, Daehyun Ahn, Sarang Kim, Jiwoong Choi, Minkyu Kim, and Hyungjun Kim. 2024b. Quick: Quantization-aware interleaving and conflict-free kernel for efficient llm inference. *Preprint*, arXiv:2402.10076.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jun Li, Li Fuxin, and Sinisa Todorovic. 2020. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *Preprint*, arXiv:2002.01113.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org.

Jing Liu, Ruihao Gong, Xiuying Wei, Zhiwei Dong, Jianfei Cai, and Bohan Zhuang. 2024a. QLLM: accurate and efficient low-bitwidth quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models. *Preprint*, arXiv:2305.17888.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024b. Spinquant: Llm quantization with learned rotations. *Preprint*, arXiv:2405.16406.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. 2024. The era of 1-bit llms: All large language models are in 1.58 bits. *Preprint*, arXiv:2402.17764.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

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

Jiayi Pan, Chengcan Wang, Kaifu Zheng, Yangguang Li, Zhenyu Wang, and Bin Feng. 2023. Smoothquant+: Accurate and efficient 4-bit post-training weightquantization for llm. *Preprint*, arXiv:2312.03788.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xuan Shen, Zhenglun Kong, Changdi Yang, Zhaoyang Han, Lei Lu, Peiyan Dong, Cheng Lyu, Chih hsiang Li, Xuehang Guo, Zhihao Shu, Wei Niu, Miriam Leeser, Pu Zhao, and Yanzhi Wang. 2024. Edgeqat: Entropy and distribution guided quantization-aware training for the acceleration of lightweight llms on the edge. *Preprint*, arXiv:2402.10787.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Xiaoxia Wu, Haojun Xia, Stephen Youn, Zhen Zheng, Shiyang Chen, Arash Bakhtiari, Michael Wyatt, Reza Yazdani Aminabadi, Yuxiong He, Olatunji Ruwase, Leon Song, and Zhewei Yao. 2023a. Zeroquant(4+2): Redefining llms quantization with a new fp6-centric strategy for diverse generative tasks. *Preprint*, arXiv:2312.08583.

Xiaoxia Wu, Zhewei Yao, and Yuxiong He. 2023b. Zeroquant-fp: A leap forward in llms post-training w4a8 quantization using floating-point formats. *Preprint*, arXiv:2307.09782.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Preprint*, arXiv:2206.01861.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2023. Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation. *Preprint*, arXiv:2303.08302.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2024. Exploring post-training quantization in llms from comprehensive study to low rank compensation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19377–19385. AAAI Press.

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

A ROSAQ vs. AWQ: Rotational invariance vs. Scaling invariance for Selecting Salient Channels

ROSAQ and AWQ uses different types of computational invariance of a linear layer – the rotational and scaling invariance. In other words, while ROSAQ uses the rotational invariance based on Eq. (1), AWQ (Lin et al., 2024) uses the scaling invariance, formulated as follows:

$$\mathbf{XW} \approx \left(\mathbf{XD}^{-1}\right) Q\left(\mathbf{DW}\right) \tag{7}$$

where $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a positive *diagonal* matrix, where AWQ uses the power of the average magnitudes of channels.

B Impact of Salient Channel Identification

Selected	PPL (↓)
Top	**5.57**
Bottom	6.63
Random	6.59
Top & Bottom	5.58

Table 2: Comparison of variants of the salient channel identification in ROSAQ, in terms of perplexity (PPL) on WikiText-2, under Llama2-7B setting.

As in Section 3.2, after performing PCA, ROSAQ selects the principal dimensions corresponding to the K largest eigenvalues as salient features. To examine the effect of using the top-K eigenvalues, Table 2 compares different methods for identifying salient channels in the PCA-projected space, measured by perplexity (PPL) on WikiText2 under the LLaMA2-7B setting. *Top* refers to ROSAQ's method of selecting the largest eigenvalues, *Bottom* represents a scheme that selects the least-principal dimensions corresponding to the K smallest eigenvaluesl, *Random* refers to a selection scheme of a random group of K-consecutive channels, and *Top* & *Bottom* refers to a combined scheme that takes both channels selected by *Top* and *Bottom*. The results show that

Rotation	PPL (↓)
w/ Head-wise	5.57
w/o Head-wise	5.94

Table 3: Comparison results between the head-wise PCA and the global PCA (corresponds to the row named "w/o Head-wise row") in terms of PPL scores on WikiText2 under the LLaMA2-7B setting.

the *Top* method yields the lowest PPL score, confirming that ROSAQ's selection scheme, based on the largest eigenvalues, effectively identifies informative salient channels.

C Impact of Head-Wise PCA Projection

As in Section 3.1.1, ROSAQ applies a head-wise PCA projection, where PCA is performed separately on each head-specific attention representation. To examine the effect of head-wise PCA, we explore an alternative approach using global PCA, applied to the concatenated multi-head attention representations, as follows

$$\begin{aligned}&\mathrm{MHSA}\left(\mathbf{Z}_{l-1}\right) \approx \\ &\mathrm{concat}\left[\mathbf{H}_1, \cdots, \mathbf{H}_H\right] \mathbf{R}\, Q\left(\mathbf{R}^T \mathbf{W}^O\right)\end{aligned} \quad (8)$$

where $\mathbf{R} \in \mathbb{R}^{d_h \cdot H \times d}$ is the global PCA projection matrix which is obtained by performing the PCA on the calibration set $\mathbf{X}_{multi}$ that consists of concatenated multi-head representations, and $\mathbf{W}^O \in \mathbb{R}^{d_h \cdot H \times d}$ is the global output weight matrix.

Table 3 presents a comparison between the head-wise PCA and the global PCA (corresponds to the row named "w/o Head-wise row") in terms of PPL scores on WikiText2 under the LLaMA2-7B setting. The results show that head-wise PCA achieves a PPL of 5.57, improving upon the PPL of 5.94 from global PCA, confirming the effectiveness of the head-wise PCA approach.

D Evaluation of Throughput using Kernel Fusion

To evaluate the throughput of ROSAQ for inference time, we apply the QUICK(Kim et al., 2024b) kernel to assess its advantages over the FP16 implementation in an INT4 quantization setting. Considering that most quantization methods present similar inference speeds to those using FP16 as batch size increases, we also evaluate the throughput using batch sizes ranging from 64 to 128 on an RTX A6000.

For each test, we design our experiment such that the number of tokens to be generated is the same as the number of the context tokens (e.g., 32 tokens are generated after seeing 32 context tokens generated 32 tokens, and 64 tokens are generated after seeing 64 context tokens). Generation process continues until an Out-of-Memory error is encountered, after which we calculates throughput performance as follows:

$$\mathrm{DecodeSpeed} = \frac{1}{\mathrm{MedianTimePerToken}}, \quad (9)$$

$$\mathrm{Throughput} = \mathrm{DecodeSpeed} \times \mathrm{BatchSize}, \quad (10)$$

Table 4 shows results of throughput for inference, comparing ROSAQ with AWQ[1] and GPTQ[2], evaluated on sample sets in Wikitext2, under LLaMA2-7b, with the relative speedup over the FP16 implementation. It is shown that ROSAQ achieves about 2.3x speedup for generating 256 tokens with a batch size of 64, and about 2x speedup for generating 128 tokens with a batch size of 128.

E Detailed Experiment Results

Tables 5, 6, and 7 present the full comparison results of ROSAQ, GPTQ(Frantar et al., 2023), SpinQuant(Liu et al., 2024b), AWQ(Lin et al., 2024), and Mixed, in terms of PPL on WikiText2, zero-shot accuracies on Common Sense Reasoning and MMLU(Hendrycks et al., 2021) tasks, under INT4/INT3 quantization, using LLaMA2-7b, LLaMA3-8b, and Qwen2-7b, respectively. Zero-shot commonsense reasoning tasks include PIQA(Bisk et al., 2020), WinoGrande(Sakaguchi et al., 2019), HellaSwag(Zellers et al., 2019), ARC-easy and ARC-challenge (Clark et al., 2018), CommonsenseQA(Talmor et al., 2019), OpenbookQA(Mihaylov et al., 2018).

F Activation Values of Salient channels: Original vs. PCA-Projected Spaces

To examine how the average magnitudes of the salient channels differ between the original and PCA-projected spaces, Table 8 represents the average magnitudes of top salient channels, where ten channels are selected after sorting all channels

[1] https://github.com/casper-hansen/AutoAWQ
[2] https://github.com/AutoGPTQ/AutoGPTQ

Device	Batch Size	# of given, # to generate	FP16	AWQ		GPTQ		ROSAQ	
			Decode Speed	Decode Speed	vs FP16	Decode Speed	vs FP16	Decode Speed	vs FP16
RTX A6000	64	128	1106	2323	110%	1331	20%	1667	50%
		256	708	1909	169%	800	12%	1638	131%
		512	OOM	1395	-	433	-	1235	-
	128	64	2077	2635	26%	2274	9%	2788	34%
		128	1299	2479	90%	1453	12%	2604	100%
		256	OOM	2012	-	806	-	2082	-

Table 4: Comparison of throughput for inference, between ROSAQ, AWQ, and GPTQ, computed on a RTX A6000, using sample sets in Wikitext2, under LLaMA2-7b, on RTX A6000, with the relative speedup over the FP16 implementation. For kernel funsions, AWQ, GPTQ, and ROSAQ use AWQ kernel, the marlin kernel, and QUICK kernel, respectively.

Model	Method		PPL ↓	Common Sense Reasoning 0-shot ↑								MMLU 0-shot ↑				
				PIQA	WinoGrande	HellaSwag	ARC_e	ARC_c	CSQA	OBQA	Avg.	Humanities	Social Science	STEM	Other	Avg.
Llama2 -7B	FP16	-	5.47	79.11	69.3	76	74.58	46.42	32.84	44.2	60.35	38.83	46.12	34.48	47.02	41.26
	INT4 g128	GPTQ	*5.69	78.62	68.82	75.33	72.56	44.20	28.83	44	58.91	36.96	42.57	31.72	44.67	38.72
		SpinQuant	5.52	77.15	67.96	74.39	70.75	42.75	24.73	42.2	57.13	31.20	34.97	28.10	37.46	32.72
		AWQ	5.60	79.05	68.67	75.31	73.70	44.28	27.76	44.2	59	36.20	46.70	35.81	46.31	40.64
		Mixed	5.68	79.22	69.53	75.36	74.16	44.71	31.78	42.6	59.62	37.79	43.74	33.62	45.67	39.90
		Ours	5.57	78.62	69.38	75.32	73.19	44.80	31.12	42.6	59.29	39.09	46.47	34.51	46.25	41.26
	INT3 g128	GPTQ	*6.43	76.66	66.93	70.62	63.97	39.07	22.36	41.2	54.40	28.25	30.03	27.40	30.29	28.90
		SpinQuant	6.25	75.90	66.38	70.09	65.24	39.51	21.46	38.8	53.91	26.06	27.66	25.37	30.54	27.25
		AWQ	6.24	77.37	68.19	73.32	69.53	44.2	26.37	40	57	31.16	32.27	30.26	37.17	32.53
		Mixed	6.30	77.48	68.11	73.20	71.63	42.49	26.78	41	57.24	34.43	39.42	33.42	40.52	36.65
		Ours	6.08	78.24	66.46	73.26	70.58	42.92	28.26	39.8	57.07	34.52	40.43	32.76	42.45	37.17

Table 5: Comparison results between ROSAQ and other quantization methods on **LLaMA2-7B**, when group-wise quantization is applied using INT4/INT3 with a group size of 128 (i.e., INT4/3 g128). PPL indicates the perplexity score on WikiText2, the performances reported at Common Sense Reasoning and MMLU indicate the zero-shot accuracies across tasks. * results were quoted from AWQ (Lin et al., 2024).

Model	Method		PPL (↓)	Common Sense Reasoning 0-shot (↑)								MMLU 0-shot (↑)				
				PIQA	WinoGrande	HellaSwag	ARC_e	ARC_c	CSQA	OBQA	Avg.	Humanities	Social Science	STEM	Other	Avg.
Llama3 -8B	FP16	-	6.14	80.85	72.61	79.16	77.69	53.33	69.53	45	68.31	54.84	72.96	53.50	71.07	62.10
	INT4 g128	GPTQ	*6.5	80.03	73.88	77.64	76.81	51.45	65.03	44.4	67.03	53.73	70.39	50.17	68.55	59.86
		SpinQuant	6.71	79.22	72.61	78.42	73.7	50.26	66.01	43.2	66.2	52.20	69	50.52	68.04	59
		AWQ	6.53	80.52	70.88	77.73	78.11	49.32	80.99	43.6	68.74	53.20	71.47	52.24	68.73	60.48
		Mixed	6.63	79.81	72.45	78.19	77.61	52.22	66.09	44.6	67.28	53.43	70.30	50.59	68.39	59.80
		Ours	6.50	81.01	70.72	78.21	75.84	51.54	79.36	45.4	68.87	54.24	71.30	52.27	69.75	60.97
	INT3 g128	GPTQ	*8.2	73.18	67.56	70.95	60.48	39	34.15	37.2	54.65	41.11	49.95	35.65	48.60	43.48
		SpinQuant	7.98	78.18	68.98	74.67	73.95	47.78	50.78	40.4	62.11	43.91	49.24	38.03	51.69	45.48
		AWQ	8.24	78.13	72.14	73.97	73.4	46.16	59.87	42.8	63.78	46.48	59.70	39.52	57.32	50.21
		Mixed	10.67	74.32	68.75	71.52	64.27	41.21	47.50	38.8	58.05	42.64	55.83	40.34	58.42	48.50
		Ours	8.11	77.64	68.59	74.1	69.82	45.65	58.64	42	62.35	47.55	60.42	45.83	58.55	52.41

Table 6: Comparison results between ROSAQ and other quantization methods on **LLaMA3-8B**, when group-wise quantization is applied using INT4/INT3 with a group size of 128 (i.e., INT4/3 g128). PPL indicates the perplexity score on WikiText2, the performances reported at Common Sense Reasoning and MMLU indicate the zero-shot accuracies across tasks. * results were quoted from AWQ (Lin et al., 2024).

Model	Method		PPL (↓)	Common Sense Reasoning 0-shot (↑)								MMLU 0-shot (↑)				
				PIQA	WinoGrande	HellaSwag	ARC_e	ARC_c	CSQA	OBQA	Avg.	Humanities	Social Science	STEM	Other	Avg.
Qwen2 -7B	FP16	-	7.14	81.12	72.30	78.80	74.62	49.74	81.65	44.2	68.92	60.72	80.92	64.95	75.73	69.42
	INT4 g128	GPTQ	7.3	80.41	70.64	77.84	75.76	51.62	79.28	45.8	68.76	58.55	79.46	61.16	74.12	67.26
		AWQ	7.33	80.52	70.88	77.73	78.11	49.32	80.99	43.6	68.74	60.77	80.08	63.53	75.09	68.79
		Mixed	7.39	80.47	70.48	78.12	74.07	50.60	80.92	43.4	68.29	58.55	80.18	62.54	74.93	67.81
		Ours	7.29	81.01	70.72	78.21	75.84	51.54	79.36	45.4	68.87	60.85	79.95	63.5	75.28	68.82
	INT3 g128	GPTQ	8.04	77.69	67.48	74.44	71.55	46.5	70.93	43.2	64.54	54.35	72.77	54.17	67.36	61.22
		AWQ	8.16	79.71	68.82	74.69	74.96	50.34	77.97	43.2	67.1	55.49	75.92	56.11	71.13	63.57
		Mixed	22.25	70.40	58.96	54.77	61.36	42.24	74.94	41.8	57.78	55.47	75.37	57.98	70.10	63.63
		Ours	7.97	79.71	69.85	76.58	69.15	46.42	78.13	43.6	66.21	56.37	77.87	61.08	72.22	65.65

Table 7: Comparison results between ROSAQ and other quantization methods on **Qwen2-7B**, when group-wise quantization is applied using INT4/INT3 with a group size of 128 (i.e., INT4/3 g128). PPL indicates the perplexity score on WikiText2, the performances reported at Common Sense Reasoning and MMLU indicate the zero-shot accuracies across tasks. * results were quoted from AWQ (Lin et al., 2024).

Method		# Rank of Avg. Magnitude									
		1	2	3	4	5	6	7	8	9	10
Rotated	Magnitude	42.13	9.38	6.25	5.44	5.06	4.38	4.04	4.03	3.96	3.27
	Channel Index	0	1	2	4	3	5	6	8	7	10
	Eigenvalue	20449.56	7449.92	4075.54	2847.99	3839.25	2720.07	2446.66	2038.99	2229.96	1551.73
Original	Magnitude	7.44	2.12	1.23	1.21	0.88	0.70	0.65	0.62	0.51	0.50
	Channel Index	1512	2944	2298	2393	3135	3431	310	3893	2077	1415

Table 8: Comparison of statistics of the salient channels between the original and PCA-projected spaces, using the LLaMA2-7B model with a random 2048-token sequence from WikiText-2. 1) The average magnitudes of top salient channels, where ten channels are selected after sorting all channels based on their average magnitudes in each respective space. 2) The corresponding eigenvalues in the PCA-projected channels

based on their average magnitudes in each respective space, using the LLaMA2-7B model with a random 2048-token sequence from WikiText-2. In the PCA-projected channels, we also present their corresponding eigenvalues.

The results show that the PCA-projected space tends to have larger average magnitudes compared to those of the original space. The relative ratio of the average magnitudes between the top-1 and top-2 channels is more dominant in the PCA-projected space compared to the original space.

G Weight Splitting Section

For the attention block, we allocate 128 salient features for W_Q, W_K, and W_V. In the case of W_O, we allocate 32 salient features per head to ensure the group size for quantization is a multiple of 32. For the FFN block, 128 salient features are set for both W_U and W_G. However, since the rotation matrix between weights of FFN block cannot be absorbed into W_U and W_G, we do not rotate the W_D. Instead, we apply per-channel scaling, proposed by AWQ(Lin et al., 2024), to protect the salient features of the W_D during quantization. We also provide an ablation study in Table 2 to test whether the earlier features of the weights are truly salient.