COAT: COrrelation-Aware Orthogonal Transform for LLM Quantization

Anonymous Authors[1]

Abstract

Quantization of large language models (LLMs) has become essential to mitigate memory and computational bottlenecks, however the occurrence of massive activation outliers poses the main challenge in low-bit LLM quantization. In the recent works, rotational transforms are demonstrated to effectively improve quantization performance by suppressing outliers, as they reshape activation distributions without changing the underlying model function. Existing approaches typically rely either on fixed data-agnostic transforms, such as Hadamard rotations, or on calibration-time optimization of learned rotations. In this paper, we demonstrate that maximizing cross-dimensional correlation provides a principled criterion for constructing quantization-friendly orthogonal transforms. We propose correlation-aware orthogonal transform (COAT) as a novel and mathematically grounded transform for post-training quantization. COAT synergistically combines a robust, data-independent rotation/Hadamard backbone with an adaptive, data-dependent component, and therefore bridges the gap between data-agnostic structured rotations and learned rotations. This approach proves to be substantially effective for post-training quantization, and numerous experiments on Llama-family models show that COAT is competitive with state-of-the-art rotation methods.

1. Introduction

Large language models (LLMs) have achieved remarkable performance across a broad range of natural language processing tasks, driving rapid growth in model scale. While this scaling has enabled strong capabilities, it has also created a major deployment bottleneck due to the high cost of inference. To improve inference efficiency, a variety of approaches have been explored, including pruning, distillation, and quantization (Ma et al., 2023; Hinton et al., 2015; Han et al., 2015; Yao et al., 2022). Among these, post-training quantization is particularly attractive because it reduces memory usage and inference cost by mapping weights and activations to low-precision representations, without requiring expensive retraining or fine-tuning. However, activation outliers remain a fundamental obstacle to accurate low-bit quantization in LLMs. Because a small number of extreme values can dominate the activation dynamic range, they substantially degrade low-bit quantization accuracy, motivating methods that explicitly reshape activations into more quantization-friendly representations.

To address this challenge, previous work has proposed several outlier-mitigation techniques, including high-precision handling of extreme values and activation smoothing with diagonal transformations. More recently, rotation-based methods have emerged as a particularly effective approach. These methods rely on rotational invariance, which allows rotation matrices to be integrated into nearby weights without affecting the original network outputs. The resulting rotated basis disperses outlier magnitude across dimensions, leading to smoother activations and improved quantizability. Methods such as QuaRot (Ashkboos et al., 2024) use randomized Hadamard rotations to suppress activation outliers and improve low-bit quantization. However, such fixed random rotations are not generally optimal, and the subsequent work SpinQuant (Liu et al., 2025) shows that learned rotations can further enhance quantization performance.

Despite the empirical effectiveness of rotation-based quantization, current methods exhibit a clear trade-off. On the one hand, fixed rotations are simple and efficient, but remain entirely data-agnostic and therefore cannot adapt to the activation statistics of a given model. On the other hand, learned rotations (Hu et al., 2025; Liu et al., 2025) can better match the data, but require end-to-end optimization under orthogonality constraints, which introduces a substantial computational overhead. For large models, this overhead can be prohibitive: prior results report 238.89 GiB of memory and 42.9 GPU hours for rotation optimization on a 70B model. Such costs are difficult to reconcile with the practical appeal of post-training quantization, whose main advantage lies in enabling fast and lightweight de-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

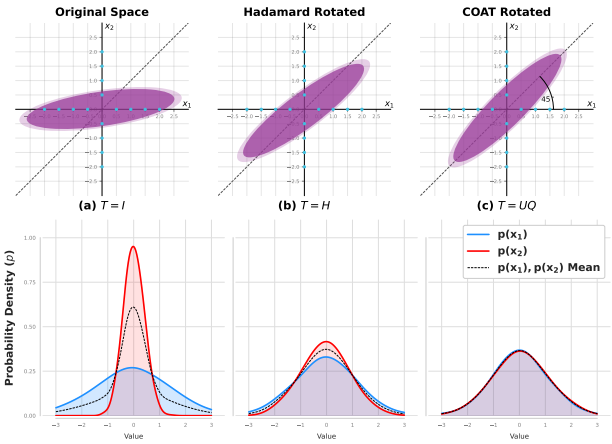

Figure 1. Geometric intuition for correlation-aware rotation design. The top row shows the original distribution, a fixed Hadamard rotation, and the proposed hybrid rotation transform, while the bottom row shows the corresponding marginals. Unlike Hadamard, which mixes coordinates without adapting to the covariance geometry, the proposed transform aligns with the dominant correlated directions before structured mixing. This yields more balanced and coherent marginals and stronger cross-dimensional correlation in the rotated space, motivating our correlation-maximization objective.

ployment. In addition, optimizing on the rotation manifold requires specialized procedures to maintain orthogonality, and the use of small calibration sets can further make the optimization susceptible to overfitting as shown in (Shao et al., 2026).

To address this gap, we propose a correlation-aware orthogonal transformation (COAT) for LLM post-training quantization. Rather than relying on either fixed data-agnostic rotations or expensive end-to-end optimization, COAT derives quantization-friendly orthogonal transforms from a correlation-based objective motivated from first principles. The resulting formulation yields a closed-form hybrid rotation that combines a fixed structured backbone with an adaptive component estimated from calibration data. Figure 1 provides the geometric intuition behind our approach. A fixed Hadamard rotation redistributes the mass of the distribution, but because it is agnostic to the covariance structure, it may not fully balance the transformed marginals. This motivates formulating rotation design as the search for orthogonal transforms that explicitly exploit cross-dimensional correlation. Experiments across multiple prominent LLM architectures demonstrate that the proposed hybrid transforms are competitive and practically efficient for low-bit post-training quantization. Taken together, these results establish a theoretically grounded calibration-based paradigm for rotation design in LLM quantization.

Our contributions are summarized as follows.

- We formulate a **correlation-based objective** for rotation design and derive a **closed-form** family of hybrid orthogonal transforms that unifies fixed structured rotations and data-dependent alignment within a single framework, showing that widely used structured rotations like Hadamard rotation arise as a special case of the proposed formulation.
- COAT delivers high quantization accuracy while remaining practically efficient. It substantially reduces the calibration cost between model sizes — for the 70 B model, COAT completes calibration in 43 minutes on a single A6000 GPU achieving a **21× speedup** in calibration time and a **9.8× reduction in memory usage** relative to SpinQuant. Notably, COAT can also be **calibrated efficiently on CPU** unlike prior rotation-based methods.

2. Related Work

2.1. Non-orthogonal Transforms

Scaling-based methods like SmoothQuant (Xiao et al., 2023) have been proposed to address activation outliers by shifting the outliers from activations to weights, reducing activation quantization errors. OmniQuant (Shao et al., 2024) introduces learnable weight clipping and uses block-wise error minimization to fine-tune quantization errors. Although these methods reduce activation quantization errors, they shift the difficulty to weight quantization, creating problems in the presence of extreme activation outliers.

2.2. Rotation Based Transforms

Among the rotation-based methods, QuIP (Chee et al., 2023) first introduces incoherent processing to reduce the impact of outliers in both the weight and activation spaces. QuIP# (Tseng et al., 2024) further improves speed by using randomized Hadamard transforms. Building on these methods, which were primarily meant for weight-only quantization, QuaRot (Ashkboos et al., 2024) extends it for activation and KV-cache quantization, relying purely on random Hadamard rotations and incoherence processing to reduce outliers. SpinQuant (Liu et al., 2025) shows that some random rotations lead to much better quantization than others, and consists of an approach that incorporates learned rotation matrices for optimal quantized network accuracy. Similar to SpinQuant, OSTQuant (Hu et al., 2025) treats the rotation along with scaling matrices as network parameters and fine-tunes them in an end-to-end fashion resulting in the transformations becoming non-orthogonal. In contrast, DartQuant (Shao et al., 2026) optimizes orthogonal rotations through a calibration-time loss, making it a calibration-based alternative rather than an end-to-end fine-tuned one.

The existing orthogonal rotation methods can be broadly divided into two classes: data-agnostic random transforms,

such as Hadamard-based approaches (Ashkboos et al., 2024), and data-dependent optimized transforms, either through end-to-end fine-tuning as in SpinQuant or through calibration-time loss optimization as in DartQuant. The proposed technique, COAT, takes the middle ground — it seeks a *data-dependent orthogonal transform*, but constructs it in *closed-form* from calibration statistics without the need for iterative optimization.

3. Background

The Transformer architecture, common in LLMs, consists of multi-head self-attention and feedforward network modules both composed of linear layers. Denote the output of a layer as $Y := XW^\top$, with $X \in \mathbb{R}^{T\times C_{in}}$ as the input activation, and $W \in \mathbb{R}^{C_{out}\times C_{in}}$ as the weight matrix. Based on rotational invariance, an orthogonal transformation $\mathbf{R} \in \mathbb{R}^{C_{in}\times C_{in}}$ is inserted into the linear layer without altering the output, yielding $\mathbf{Y} = \mathbf{XRR}^\top\mathbf{W}^\top = (\mathbf{XR})(\mathbf{WR})^\top$. By combining $\mathbf{R}$ with the previous weight matrix and $\mathbf{R}^\top$ with current layer's weight matrix, we can rotate the activations without introducing any additional inference cost.

Without altering the output of the model, we can insert four **orthogonal matrices** $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$, $\mathbf{R}_4$ within the Transformer block as in Fig. 2 (adapted from (Liu et al., 2025)). Multiplying $\mathbf{R}_1$ on the left side of W_q, W_k, W_v, W_{up}, W_{gate}, and $\mathbf{R}_1^\top$ on the right side of W_{out} and W_{down}, an equivalent transformation is achieved. The rotation $\mathbf{R}_2$ can be inserted between W_v and W_{out}, whereas $\mathbf{R}_3$ can be inserted between the rotated encodings of Q and K, and $\mathbf{R}_4$ can be inserted before W_{down}. Finally, $W_{embedding}$ is multiplied on the left by $\mathbf{R}_1^\top$, and $\mathbf{R}_1$ is multiplied on the right of W_{lm_head}, completing all equivalent transformations.

4. Correlation-Aware Orthogonal Transformation (COAT)

Our framework departs from standard decorrelation-based techniques (Fleury et al., 1997; Jolliffe, 2011) and instead seeks rotations that **maximize cross-dimensional correlation** in the transformed space. More precisely, for an orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$, and the transformation $\boldsymbol{Y} = \boldsymbol{XQ}$, we consider the following optimization problem:

$$J(\boldsymbol{Q}) = \max_{\boldsymbol{Q}} \Big(\sum_{i\neq j} \big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ij}^2 \Big), \quad \boldsymbol{Q}^\mathrm{T}\boldsymbol{Q} = \boldsymbol{I}, \tag{1}$$

where $\boldsymbol{\Sigma} := \frac{1}{n}\boldsymbol{X}^\mathrm{T}\boldsymbol{X}$ is the sample covariance matrix under the standard assumption that the data $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is mean-centered. Also, $\boldsymbol{\Sigma_Y}(\boldsymbol{Q})$ is the covariance matrix of the transformed data $\boldsymbol{Y}$ satisfying $\boldsymbol{\Sigma_Y}(\boldsymbol{Q}) := \frac{1}{n}\boldsymbol{Y}^\mathrm{T}\boldsymbol{Y} = \boldsymbol{Q}^\mathrm{T}\big(\frac{1}{n}\boldsymbol{X}^\mathrm{T}\boldsymbol{X}\big)\boldsymbol{Q} = \boldsymbol{Q}^\mathrm{T}\boldsymbol{\Sigma}\boldsymbol{Q}$, and $\boldsymbol{I} \in \mathbb{R}^{d\times d}$ denotes the identity matrix.

Let $\|\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\|_F$ denote the Frobenius norm of the covariance matrix $\boldsymbol{\Sigma_Y}(\boldsymbol{Q})$, then $\|\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\|_F^2 = \sum_{i,j}\big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ij}^2 = \sum_{i\neq j}\big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ij}^2 + \sum_i \big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ii}^2$. Since the term $\|\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\|_F^2$ is a constant, the optimization problem in (1) can be recast as the minimization problem:

$$J(\boldsymbol{Q}) = \min_{\boldsymbol{Q}} \left(\sum_i \big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ii}^2 \right), \quad \boldsymbol{Q}^\mathrm{T}\boldsymbol{Q} = \boldsymbol{I}. \tag{2}$$

Considering the eigen-decomposition $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{\Lambda} = \mathrm{diag}\,(\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_d)$ is the diagonal matrix of the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{U}$ is the matrix whose columns are the eigenvectors of $\boldsymbol{\Sigma}$. Since, $\boldsymbol{U}$ is orthogonal, we write $\boldsymbol{Q} = \boldsymbol{UV}$ for some orthogonal matrix $\boldsymbol{V}$, so that $\boldsymbol{\Sigma_Y}(\boldsymbol{Q}) = \boldsymbol{Q}^\mathrm{T}\boldsymbol{\Sigma}\boldsymbol{Q} = \boldsymbol{V}^\mathrm{T}\boldsymbol{U}^\mathrm{T}\boldsymbol{\Sigma}\boldsymbol{U}\boldsymbol{V} = \boldsymbol{V}^\mathrm{T}\boldsymbol{\Lambda}\boldsymbol{V}$. Accordingly, we reformulate the optimization problem as:

$$J(\boldsymbol{Q}) = \min_{\boldsymbol{Q}} \Big(\sum_{i=1}^{d} \big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ii}^2 \Big), \quad \boldsymbol{Tr}\left(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\right) = \sum_{i=1}^{d}\lambda_i \tag{3}$$

The minimum is achieved for an orthogonal matrix $\boldsymbol{Q}$ satisfying that

$$\big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ii} = \frac{1}{d}\left(\sum_{i=1}^{d}\lambda_i\right) = \frac{\boldsymbol{Tr}\,(\boldsymbol{\Sigma})}{d} \tag{4}$$

for all $i = 1, 2, 3, \ldots, = d$. The existence of such an optimal orthogonal matrix, say $\boldsymbol{Q}^*$, is guaranteed by the Schur-Horn theorem (Horn, 1954). However, the theorem doesn't give a procedure to construct one. Interestingly, the well-known Hadamard matrices are the simplest of solutions when the dimension d is a power of 2, that is, $d = 2^k$ for some non-negative integer k. Each Hadamard matrix has entries either $+1$ or -1, and for every doubling of the dimension, the matrix is scaled by a factor of $1/\sqrt{2}$. Any member of Hadamard family $\boldsymbol{H}_{2^k}$ is an orthogonal matrix with squares of the entries equal to $1/d$. Therefore, in the matrix equation $\boldsymbol{Q} = \boldsymbol{UV}$ choosing $\boldsymbol{V}$ to be the Hadamard matrix $\boldsymbol{H}_d$, where $d = 2^k$, we have that $\boldsymbol{\Sigma_Y}(\boldsymbol{Q}) = \boldsymbol{H}_d^\mathrm{T}\boldsymbol{\Lambda}\boldsymbol{H}_d$, and

$$\big(\boldsymbol{\Sigma_Y}(\boldsymbol{Q})\big)_{ii} = \sum_{k=1}^{d}\lambda_k(\boldsymbol{H}_d)_{ij}^2 = \sum_{k=1}^{d}\frac{\lambda_k}{d} = \overline{\lambda}. \tag{5}$$

This shows that the Hadamard matrices offer the most natural solution to our final optimization problem (3), so we can choose the desired matrix $\boldsymbol{Q} = \boldsymbol{UV}$ simply as $\boldsymbol{Q} = \boldsymbol{UH}_d$.

5. Experiments

Model and Dataset. We evaluate our method on Llama-family models, including Llama-2 (7B, 13B, and 70B) (Touvron et al., 2023) and Llama-3 (8B). We report *perplexity* on WikiText2 (Merity et al., 2016), C4 (Raffel et al., 2020), and

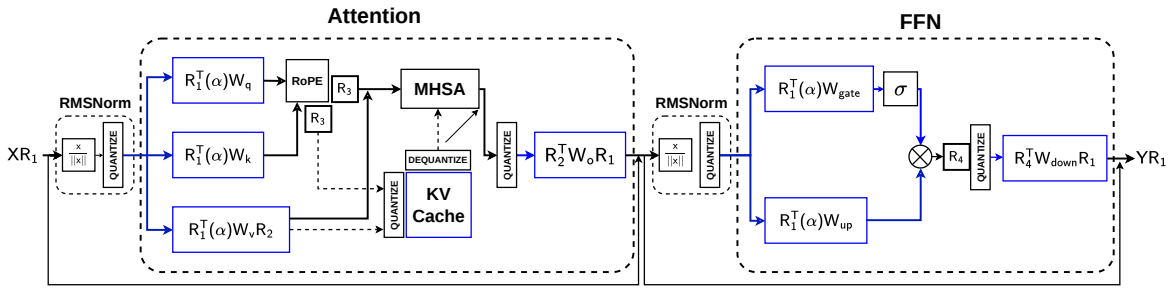

Figure 2. Rotational Invariance mechanism in LLMs. The black outlines represent the flow in FP16 format, while the blue outlines indicate the flow in INT4 format; the dashed line shows the flow in and out of the KV buffer.

PTB (Marcus et al., 1993), and further measure zero-shot performance on nine downstream tasks: WinoGrande (Sakaguchi et al., 2021), SIQA (Sap et al., 2019), PIQA (Bisk et al., 2020), OpenBookQA (Mihaylov et al., 2018), LAMBADA (Radford et al., 2019), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), ARC-E, and ARC-C (Boratko et al., 2018).

Baselines and Implementation Details. We compare our approach with existing rotation-based quantization methods, including Hadamard rotations in QuaRot (Ashkboos et al., 2024) and end-to-end learned rotations in SpinQuant (Liu et al., 2025). For the main results, we quantize weights with GPTQ (Frantar et al., 2022) under the standard setup using 128 WikiText2 calibration samples, each of length 2048, and quantize activations with per-token asymmetric quantization. We follow the same rotational invariance construction as in Figure 2 with four orthogonal matrices of which R_1 and R_2 are calibrated using COAT while R_3 and R_4 are left online as random Hadamard matrices. Our orthogonal matrices are computed offline from activation covariance statistics estimated on the same 128 WikiText2 calibration samples using the Welford algorithm (Welford, 1962) (Chan et al., 1983), which provides a numerically stable unbiased online covariance estimator.

5.1. Main Results

Table 1 compares our 4-bit weight, activation and KV-cache method against FloatingPoint, QuaRot, and SpinQuant on three WikiText-style perplexity evaluations and the average accuracy over nine zero-shot commonsense reasoning tasks. Across all four model scales, our approach remains consistently competitive and shows a favorable tradeoff between language modeling quality and downstream reasoning accuracy under aggressive quantization.

Compared with QuaRot, our method reduces perplexity by 17.1% on Llama-2 7B and 6.5% on Llama-3 8B, while improving zero-shot accuracy by 1.79 percentage points on Llama-2 13B, and 3.93 percentage points on Llama-3 8B. Relative to SpinQuant, our method improves zero-shot accuracy by 0.06 percentage points on Llama-2 13B and 0.12 percentage points on Llama-2 70B, while remaining within 0.10 percentage points on Llama-3 8B. However, SpinQuant achieves lower perplexity in several settings, including 6.7% lower perplexity on Llama-2 13B and 4.5% lower perplexity on Llama-3 8B. Overall, these results show that our method offers a strong accuracy-perplexity tradeoff at 4 bits, remaining highly competitive with SpinQuant and even outperforming it in certain settings, particularly on zero-shot reasoning for Llama-2 13B and 70B.

A comparison of rotation matrix computation time and memory cost of SpinQuant and COAT on an Nvidia A6000 GPU server is shown in Table 4. COAT is a simple calibration method, which has significant reduction in compute cost across various models. For the 70B model, COAT completes the calibration in 43 minutes on a single GPU and moreover after pre-saving the activations can also **calibrate on CPU in 51 minutes**(AMD EPYC 7272), achieving a **speedup of 21× in calibration and 9.8× in memory usage** compared to SpinQuant. Moreover, SpinQuant is nearly impossible to train on a CPU.

5.2. Calibration Dataset

To study sensitivity of the calibration-data, we construct COAT using WikiText2, PTB, and C4 samples and evaluate the resulting quantized models. As shown in Table 5 below, the results are broadly consistent across calibration datasets for both Llama-2 7B and 13B, suggesting that COAT is reasonably robust to the choice of calibration data. Table 6 also shows that COAT is robust to calibration sample size.

Table 1. Comparison of the average Perplexity scores across three datasets and the average accuracy on nine Zero-shot Common Sense Reasoning tasks. The results for all the comparison methods were obtained using their publicly available codebases. Full results can be found in Table 2.

Bits (W-A-KV)	Method	Llama-2 7B		Llama-2 13B		Llama-2 70B		Llama-3 8B	
		PPL ↓	0-shot[9] ↑	PPL ↓	0-shot[9] ↑	PPL ↓	0-shot[9] ↑	PPL ↓	0-shot[9] ↑
16	FloatingPoint	16.88	61.16	20.85	64.28	11.09	69.53	8.92	66.04
4	QuaRot	27.01	57.03	24.98	59.87	**11.49**	67.41	12.29	57.32
	SpinQuant	25.12	**57.55**	**23.37**	61.60	11.76	68.05	**10.99**	**61.35**
	COAT	**22.38**	56.62	24.91	**61.66**	11.87	**68.17**	11.49	61.25

Table 2. Complete comparison of the average accuracy on nine Zero-Shot Commonsense Reasoning tasks across different models.

Model	Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB	HS	ARC-E	ARC-C	MMLU	Avg ↑
Llama-2 7B	16	Full Precision	69.06	46.16	79.05	44.20	73.90	76.02	74.54	46.33	41.21	61.16
	4	QuaRot	65.27	43.65	77.04	40.60	70.25	72.81	68.43	41.64	33.56	57.03
		SpinQuant	64.17	44.73	76.44	40.60	71.16	73.40	70.50	42.32	34.67	57.55
		COAT	66.54	44.93	76.61	41.20	69.47	71.52	66.84	40.78	31.68	56.62
Llama-2 13B	16	Full Precision	72.14	47.39	80.52	45.20	76.73	79.38	77.44	49.15	50.53	64.28
	4	QuaRot	69.30	44.27	78.24	42.80	65.44	75.75	72.64	45.90	44.52	59.87
		SpinQuant	68.35	45.65	77.75	43.60	73.28	77.10	75.00	47.61	46.08	61.60
		COAT	70.17	46.32	77.75	44.60	73.98	75.73	74.16	46.42	45.79	61.66
Llama-2 70B	16	Full Precision	77.98	49.13	82.75	48.80	79.58	83.80	81.02	57.51	65.20	69.53
	4	QuaRot	76.48	46.26	81.28	46.00	79.00	81.82	79.46	55.63	60.78	67.41
		SpinQuant	75.93	47.85	81.50	47.40	79.16	82.90	79.91	55.20	62.59	68.05
		COAT	76.72	48.21	81.39	48.00	78.19	82.70	79.59	57.00	61.71	68.17
Llama-3 8B	16	Full Precision	73.32	47.19	80.96	45.00	75.63	79.14	77.82	53.24	62.06	66.04
	4	QuaRot	66.77	44.78	73.61	41.00	64.93	71.69	66.33	40.02	46.77	57.32
		SpinQuant	67.56	43.71	77.26	41.40	72.06	75.50	73.48	47.95	53.20	61.35
		COAT	67.09	45.19	76.77	42.60	70.02	74.30	73.53	50.17	51.64	61.25

Table 6. Comparison of Perplexity scores of COAT calibration results across sample sizes.

Model	Sample	WikiText2	PTB	C4	Avg
Llama-2 7b	64	6.39	53.22	8.89	22.83
	128	6.38	51.93	8.83	22.38
	256	6.34	50.58	8.59	21.83
Llama-2 13b	64	5.50	59.41	7.76	24.22
	128	5.60	61.24	7.90	24.91
	256	5.56	61.84	7.81	25.07

5.3. Layerwise Quantization Error

Figure 3 compares the mean-squared quantization error across hidden dimensions for up- and q-projection activations in Llama-2 7B at layers 10 and 25 using Hadamard and COAT transformations. Across all four cases, COAT consistently shifts the error profile downward relative to Hadamard, demonstrating lower quantization error. This trend indicates that COAT provides a more effective redistribution of activation values prior to quantization, reducing distortion and improving numerical fidelity. The improvement is observed in both intermediate and deeper layers, suggesting that the benefits of COAT remain stable across different network depths and projection types.

6. Conclusion

We presented COAT, a principled framework for post-training quantization of LLMs. Starting from a first-principles objective over orthogonal transforms, we derived a closed-form solution for cross-dimensional correlation maximization with an equivalent variance-equalization objective. Our analysis reveals a family of optimal transforms, including constructions with either data-independent or data-dependent backbones. We instantiated COAT using a fixed Hadamard backbone together with a data-dependent orthogonal component. Empirical results across diverse LLMs showed consistent gains over purely data-agnostic baselines while being competitive with end-to-end learned baselines.

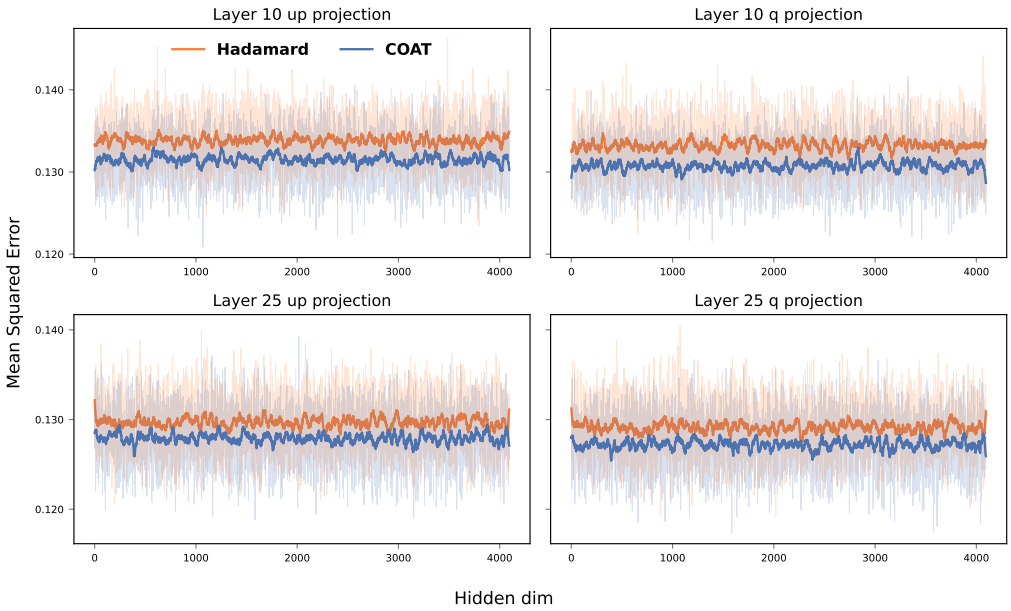

Figure 3. Comparison of mean-squared quantization error across hidden dimensions for up- and q-projection activations in Llama-2 7b at layers 10 and 25.

Table 3. Comparison of Perplexity scores across three datasets for different models.

Model	Bits (W-A-KV)	Method	Wiki	PTB	C4	Avg ↓
Llama-2 7B	16	Full Precision	5.47	37.91	7.26	16.88
	4	QuaRot	6.17	66.41	8.46	27.01
		SpinQuant	5.99	61.03	8.34	25.12
		COAT	6.38	51.93	8.83	22.38
Llama-2 13B	16	Full Precision	4.88	50.94	6.73	20.85
	4	QuaRot	5.51	61.78	7.65	24.98
		SpinQuant	5.30	57.30	7.51	23.37
		COAT	5.60	61.24	7.90	24.91
Llama-2 70B	16	Full Precision	3.32	24.25	5.71	11.09
	4	QuaRot	3.81	24.53	6.14	11.49
		SpinQuant	3.69	25.52	6.07	11.76
		COAT	3.77	25.76	6.09	11.87
Llama-3 8B	16	Full Precision	5.47	37.91	7.26	16.88
	4	QuaRot	8.40	14.68	13.79	12.29
		SpinQuant	7.41	13.33	12.23	10.99
		COAT	7.96	13.50	13.00	11.49

Table 4. Comparison of calibration cost

Cost	Method	7B	13B	70B
GPU hour (h)	SpinQuant	0.41	0.84	46.40
	COAT	0.26	0.35	2.25
Mem (GiB)	SpinQuant	18.23	32.21	218.72
	COAT	4.51	7.05	22.42

Table 5. Comparison of Perplexity scores of COAT across various calibration datasets.

Model	Dataset	WikiText2	PTB	C4	Avg
Llama-2 7B	WikiText2	6.38	51.93	8.83	22.38
	PTB	6.53	53.02	9.11	22.88
	C4	6.30	49.94	8.56	21.60
Llama-2 13B	WikiText2	5.60	61.24	7.90	24.91
	PTB	5.60	59.29	7.61	24.16
	C4	5.65	62.29	7.93	25.32

Impact Statement

This paper presented an efficient approach for quantization of Large Language Models (LLMs). LLMs can be used to make academic and industry workflows efficient and effective. However, they also have the risk of being misused to generate fake content. Our contribution enables edge-deployment of LLMs more efficient and does not alleviate risks arising due to their misuse in any way.

References

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Boratko, M., Padigela, H., Mikkilineni, D., Yuvraj, P., Das, R., McCallum, A., Chang, M., Fokoue-Nkoutche, A., Kapanipathi, P., Mattei, N., et al. A systematic classification of knowledge, reasoning, and context within the arc dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pp. 60–70, 2018.

Chan, T. F., Golub, G. H., and LeVeque, R. J. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983.

Chee, J., Cai, Y., Kuleshov, V., and De Sa, C. M. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36: 4396–4429, 2023.

Fleury, M., Downton, A. C., and Clark, A. F. Karhünen-loève transform: An exercise in simple image-processing parallel pipelines. In *European Conference on Parallel Processing*, pp. 815–819. Springer, 1997.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Horn, A. Doubly stochastic matrices and the diagonal of a rotation matrix. *American Journal of Mathematics*, 76 (3):620–630, 1954.

Hu, X., Cheng, Y., Yang, D., Xu, Z., Yuan, Z., Yu, J., Xu, C., Jiang, Z., and Zhou, S. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

Jolliffe, I. *Principal Component Analysis*, pp. 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_455.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Peng, G., Qiao, Y., and Luo, P. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45472–45496, 2024.

Shao, Y., Chen, Y., Wang, P., Yu, J., Lin, J., Wei, Z., Cheng, J., et al. Dartquant: efficient rotational distribution calibration for llm quantization. *Advances in Neural Information Processing Systems*, 38:143936–143970, 2026.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E.,

Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Tseng, A., Chee, J., Sun, Q., Kuleshov, V., and De Sa, C. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *Proceedings of machine learning research*, 235:48630, 2024.

Welford, B. P. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Yao, Z., Yazdani Aminabadi, R., Zhang, M., Wu, X., Li, C., and He, Y. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in neural information processing systems*, 35: 27168–27183, 2022.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.