IMPROVE MXFP4 QUANTIZATION VIA INTER-BLOCK PERMUTATION AND MICRO-BLOCK GIVENS ROTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have achieved remarkable performance across diverse tasks, but their growing computational and memory costs pose substantial challenges for efficient deployment. Post-training quantization (PTQ) has emerged as a practical solution, while recent hardware support for microscaling formats such as MXFP4 further enables efficient low-bit inference. However, MXFP4 remains highly sensitive to massive activation outliers due to its power-of-two E8M0 shared scales and low-precision E2M1 values. Existing block-wise Hadamard transformations mitigate outliers by uniformly spreading their energy across an entire quantization block, but such magnitude reduction is not necessarily optimal for MXFP4. We observe that MXFP4 contains a set of favorable, exactly or nearly representable magnitudes, termed **lossless quantization points**, and quantization error is strongly determined by the distance of transformed values to these points. Based on this observation, we propose a novel transform framework for MXFP4. First, we propose *Inter-block permutation* to group massive outliers with low-importance channels, reducing their interference with important features under block-wise shared scaling. Second, we propose *Micro-block Givens Rotation* to adaptively control the number of participating channels and redistribute outlier energy toward favorable quantization points, instead of uniformly dispersing it across the entire block. Experiments on the LLaMA2/3 and Qwen3 model families demonstrate that our method better adapts to the quantization characteristics of MXFP4 than the Hadamard transform and achieves superior quantization performance across multiple LLM benchmarks, providing new insights into the design of quantization methods for next-generation microscaling low-precision formats.

1 INTRODUCTION

Large language models (LLMs) (Grattafiori et al., 2024; Yang et al., 2025; DeepSeekAI et al., 2024; Achiam et al., 2023) have demonstrated remarkable capabilities across a wide range of tasks, including natural language understanding, reasoning, and content generation. However, their rapidly growing model sizes also introduce substantial memory and computational overhead, posing significant challenges for efficient deployment. Among various model compression techniques (Frantar & Alistarh, 2023; Gu et al., 2024), post-training quantization (PTQ) (Frantar et al., 2023) has become one of the most practical and widely adopted approaches, as it can significantly reduce memory footprint and inference cost without requiring expensive retraining.

Early low-bit quantization methods mainly focused on integer formats such as INT8 (Xiao et al., 2023). More recently, modern accelerators, such as NVIDIA Blackwell GPUs, have introduced native support for Microscaling floating-point formats (Rouhani et al., 2023), among which MXFP4 has attracted increasing attention (Lee et al., 2025; Shao et al., 2026). MXFP4 partitions tensors into blocks of 32 elements, where each block shares an E8M0 scale and each value is represented in the E2M1 FP4 format. Such fine-grained block-wise scaling better adapts to the non-uniform distributions of LLM weights and activations and limits the influence of outliers (Dettmers et al., 2022) to a local block. Moreover, since the E8M0 scale is restricted to powers of two, scaling can be implemented efficiently through simple exponent adjustment, making MXFP4 attractive for efficient low-bit training and inference (DeepSeekAI et al., 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

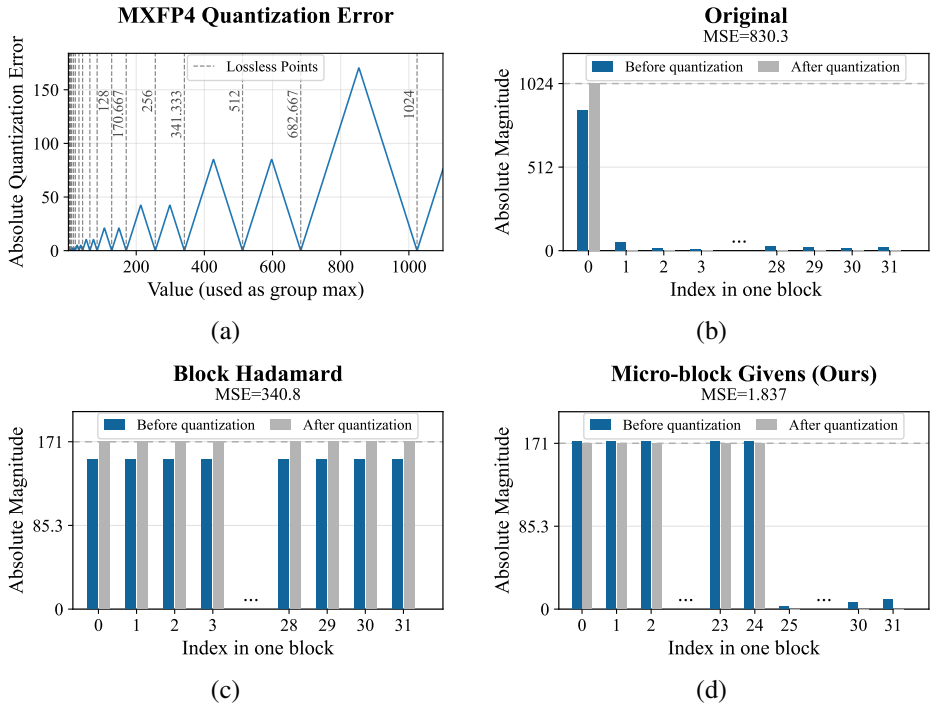


Figure 1: Motivation of our method. (a) illustrates the MXFP4 quantization error[1] when a value serves as the block-wise maximum, with the shared scale determined by this maximum. Due to the power-of-two (PoT) constraint of the scale factor, zero-error points repeatedly appear. (b) shows the distribution of 32 elements within a block containing a massive outlier. (c) presents the result after applying the Hadamard transform, which uniformly redistributes the outlier magnitude across all 32 elements and reduces the quantization error. (d) shows our method, which redistributes the outlier magnitude only within a micro-block, allowing the transformed values to approach the lossless quantization points and thereby reducing the quantization error.

Despite its fine-grained scaling, MXFP4 remains highly sensitive to extreme outliers. In particular, the E8M0 scale itself is quantized to powers of two. As the magnitude of the shared scale increases, the spacing between adjacent representable scales also becomes larger, making blocks containing massive outliers particularly difficult to quantize accurately. A common solution is to apply block-wise Hadamard rotation (Shao et al., 2026) to reduce extreme magnitudes by distributing outlier energy across the channels within a quantization block. However, for MXFP4, a smaller transformed magnitude does not necessarily imply a smaller quantization error. Due to the discrete E8M0 scale and E2M1 codebook, MXFP4 contains a set of exactly or nearly representable magnitudes, which we refer to as **lossless quantization points**, as shown in Figure 1(a). Therefore, the quantization error is determined more importantly by its distance to these representable points. This observation suggests that uniformly spreading an outlier over the entire block can be suboptimal. Instead, we aim to redistribute the energy of the outlier within an adaptively selected micro-block, such that its transformed magnitude is explicitly driven toward a favorable quantization point.

In addition, MXFP4 performs block-wise quantization, where every 32 consecutive elements share the same scale. Our micro-block transform redistributes the energy of an outlier to several neighboring channels, inevitably perturbing the other channels participating in the transformation. Ideally, these channels should have relatively low importance so that the induced perturbation has minimal impact on the model output. This motivates a channel-reordering strategy before the micro-block rotation.

Based on the above observations, we propose a *Micro-block Givens Rotation* that adaptively controls the number of participating channels, rather than uniformly spreading outlier energy across the entire MXFP4 block, and a *Inter-block Permutation* to rearrange the channels to enable more effective outlier suppression with limited impact on important features.

[1]In this work, the MXFP4 shared scale is computed with an additional $3/4$ scaling factor to avoid clipping error following existing implementations (Lee et al., 2025; Egiazarian et al., 2025).

Our primary contributions are summarized as follows:

- We propose *Micro-block Givens* for MXFP4 to reduce the quantization error of outliers themselves. By exploiting the lossless quantization points induced by power-of-two scaling, our method redistributes outlier energy within a smaller adaptive micro-block, driving the transformed magnitude closer to favorable quantization points.
- We propose *Inter-block Permutation* for MXFP4 to reduce the adverse impact of outliers on other elements within the same quantization group. By distributing massive outliers across different groups and pairing them with low-importance channels, our method mitigates the quantization degradation introduced to neighboring elements.
- Experimental results on LLaMA2/3 and Qwen3 models demonstrate that our method improves MXFP4 quantization performance and achieves competitive or superior results compared with conventional Hadamard transforms.

2 Related Work

2.1 LLM Quantization

Model quantization converts high-precision weights and activations into low-precision representations to reduce the cost of memory and computational resources. For LLMs, activation outliers (Dettmers et al., 2022) introduce large dynamic ranges, requiring larger quantization steps and causing significant quantization errors for non-outlier values. Similar to post-training pruning (Sun et al., 2023), existing quantization research mainly focuses on retraining-free approaches to alleviate the impact of outliers, which can be broadly categorized into four directions: (1) **Data-type-based methods** design fine-grained numerical formats to better represent tensors with large dynamic ranges. Instead of using a single global scale, microscaling-based approaches adopt block-wise scaling factors, allowing each small group of elements to have an independent quantization range and reducing the influence of outliers on other quantized values (Jang & Tambe, 2025). (2) **Scaling-based methods** mitigate outlier effects by redistributing the quantization difficulty between activations and weights. Methods such as SmoothQuant (Xiao et al., 2023) migrate activation outliers to weights through channel-wise scaling factors, while AWQ (Lin et al., 2024) identifies important weight channels and applies adaptive scaling to preserve salient features under low-bit quantization. (3) **Transform-based methods** employ orthogonal transformations to redistribute outlier magnitudes across channels before quantization. QuaRot (Ashkboos et al., 2024) applies fixed Hadamard transforms (Tseng et al., 2024) to smooth activation distributions, while SpinQuant (Liu et al., 2024) and FlatQuant (Sun et al., 2025) learn optimized rotation matrices to further improve quantization robustness. (4) **Compensation-based methods** reduce quantization-induced reconstruction errors by adjusting weights without fine-tuning. Representative approaches such as GPTQ (Frantar et al., 2023) and OBC (Frantar et al., 2022) use Hessian-based compensation to update unquantized weights. Subsequent methods such as GPTAQ (Li et al., 2022) and BOA (Kim et al., 2025) further consider error propagation across layers and attention-specific characteristics, respectively.

2.2 Microscaling Formats

Compared with low-bit integer formats, floating-point provides a substantially wider dynamic range under the same bit budget, making them better suited to the highly non-uniform distributions commonly observed in LLM weights and activations (Liu et al., 2023). Microscaling formats (Rouhani et al., 2023) partition tensors into small groups and assign a separate scaling factor to each group, which further limits the adverse influence of extreme outliers on the quantization precision of other values within the tensor.MXFP4 adopts an E8M0 power-of-two (PoT) scale for each micro-block, providing a simple and hardware-friendly solution for FP4 computation (Abecassis et al., 2025). Subsequent formats further improve numerical flexibility through more sophisticated scaling schemes. For example, NVFP4 employs an E4M3 block-level scale for every 16 elements together with an additional tensor-level FP32 scale, forming a two-level scaling hierarchy. More recently, Hi-Float4 (HiF4) (Luo et al., 2026) introduces a three-level hierarchical scaling scheme together with a redesigned FP4 representation to better capture dynamic ranges.

2.3 HADAMARD TRANSFORM IN LLMS QUANTIZATION

Rotation-based transforms are a cornerstone of modern W4A4 Quantization. The most classical transform matrix is the Hadamard transform. Its ±1-valued orthogonal structure uniformly redistributes energy across dimensions, making it explicitly well-designed for low-bit INT deployment scenarios. However, recent research has identified a fundamental incompatibility between global Hadamard rotations and Microscaling (MX) formats (Shao et al., 2026; Egiazarian et al., 2025). For instance, BRQ (Shao et al., 2026) demonstrated that block-wise rotation is superior to global rotation for MXFP4. Although BRQ (Shao et al., 2026) alluded to the potential for optimized transformation matrices for MXFP4, the design of such specialized matrices remains an open challenge.

3 PREREQUISITES

Basic Principles of Quantization. Typically, quantization maps a floating-point value x to a k-bit integer grid using a scaling factor s and a zero-point z:

$$\hat{x} = \text{clamp}\left(\lfloor \frac{x}{s} \rceil + z, Q_{min}, Q_{max}\right), \tag{1}$$

where $\lfloor \cdot \rceil$ denotes the rounding operator. For symmetric quantization, $z = 0$.

MXFP4 Formats. Unlike uniform integer formats, 4-bit floating-point (FP4) formats are characterized by the $ExMy$ notation, where x and y bits are allocated to the exponent and mantissa, respectively. A general FP4 value v is decoded using a sign bit (s), exponent bits (e), and mantissa bits (m) as follows:

$$v = (-1)^s \times 2^{e-\text{bias}} \times (1 + m \cdot 2^{-M}), \tag{2}$$

where M denotes the number of mantissa bits. For example, in the standard FP4 E2M1 configuration, each element provides 7 distinct positive values $\{0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 6.0\}$ along with their negative counterparts and zero, forming a non-uniform quantization grid. MXFP4 adopts a microscaling scheme, where the tensor is partitioned into small groups and each group is associated with an independent power-of-two scaling factor. Specifically, for a group $\mathbf{x}_g$ containing G elements, MXFP4 first computes a shared scale factor s_g and quantizes the normalized values into the FP4 E2M1 format:

$$\hat{x}_{g,i} = s_g \cdot Q_{\text{E2M1}}\left(\frac{x_{g,i}}{s_g}\right), \tag{3}$$

where $Q_{\text{E2M1}}(\cdot)$ denotes the FP4 E2M1 quantizer. In the MXFP4 format, the scaling factor is represented by an 8-bit exponent-only format (E8M0), which restricts the scale to powers of two:

$$s_g = 2^{k_g - 2}, \tag{4}$$

where k_g is an integer shared by all elements within the group. Therefore, MXFP4 separates the representation of magnitude and fine-grained values: the E8M0 scale captures the coarse dynamic range of each micro-block.

Outlier Suppression via Orthogonal Transformation. To mitigate the impact of extreme outliers, recent rotation-based PTQ methods (Ashkboos et al., 2024; Liu et al., 2024) introduce an orthogonal matrix $\mathbf{Q}$ (where $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$) to transform the activation space. For a standard linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W}$, the transformation maintains mathematical equivalence as follows:

$$\mathbf{Y} = (\mathbf{X}\mathbf{Q})(\mathbf{Q}^T\mathbf{W}) = \tilde{\mathbf{X}}\tilde{\mathbf{W}}, \tag{5}$$

where $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{Q}$ and $\tilde{\mathbf{W}} = \mathbf{Q}^T\mathbf{W}$ represent the rotated activations and weights, respectively.

A commonly used orthogonal transform is the Hadamard transform. For a d-dimensional input, the normalized Hadamard matrix $\mathbf{H}_d$ satisfies

$$\mathbf{H}_d^T\mathbf{H}_d = \mathbf{I}, \tag{6}$$

where each entry of $\mathbf{H}_d$ has the same magnitude,

$$|\mathbf{H}_{i,j}| = \frac{1}{\sqrt{d}}. \tag{7}$$

Therefore, each transformed channel is a normalized linear combination of all input channels:

$$\tilde{\mathbf{X}}{:}, j = \frac{1}{\sqrt{d}} \sum i = 1^d s i, j \mathbf{X}_{:,i}, \qquad s_{i,j} \in -1, +1. \tag{8}$$

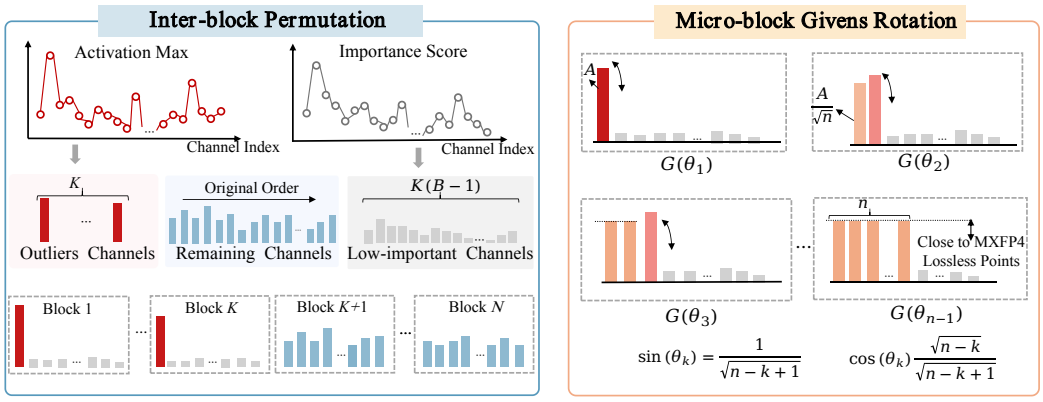


Figure 2: Overview of our method. First, *Inter-block Permutation* is performed. The Top-K channels with the largest activation magnitudes are identified as outlier channels, while $K(B-1)$ low-importance channels are selected according to the channel importance score, where B denotes the quantization block size. Each outlier channel is then grouped with $B-1$ low-importance channels to form a new block, while the original order of the remaining channels is preserved. For each outlier-containing block, *Micro-block Givens Rotation* is performed. The optimal number of participating channels n is determined by minimizing the reconstruction error. Subsequently, a sequence of Givens rotations is applied, through which the outlier energy is progressively redistributed from the dominant channel to the selected low-importance channels within the block.

4 METHODOLOGY

Our method consists of two complementary components: (1) *Inter-block Permutation*: The inter-block permutation reorganizes the channel layout by pairing outlier channels with low-importance channels, preventing extreme values from dominating the quantization behavior of important channels and reducing intra-block interference. (2) *Micro-block Givens Rotation*. For outlier-containing blocks, the proposed micro-block Givens rotation adaptively redistributes outlier energy within a compact group. By controlling the number of participating channels, the transformation moves the magnitudes closer to the lossless quantization points of MXFP4, effectively mitigating the scale rounding error introduced by power-of-two scaling.

4.1 INTER-BLOCK PERMUTATION

Given the calibration activations $\mathbf{X} \in \mathbb{R}^{S\times N}$, we first measure the global activation magnitude of the j-th channel as $a_j = \max_s |X_{s,j}|$. We then select K outlier channels:

$$\mathcal{M} = \mathrm{TopK}_j(a_j, K), \tag{9}$$

where $\mathcal{M}$ denotes the set of selected outlier channels.

For the remaining channels, we aim to populate each outlier-containing group with channels of sufficiently small importance. Since these channels are likely to be quantized to zero under low-bit quantization, we adopt a pruning-inspired importance metric based on the original activations and weights. Specifically, the global importance of channel j is defined as

$$s_j = \|\mathbf{X}_{:,j}\|_2 \, \|\mathbf{W}_{:,j}\|_1 . \tag{10}$$

We then select the $K(Blocksize-1)$ least important non-outlier channels as filler channels:

$$\mathcal{F} = \mathrm{BottomK}_{j\notin\mathcal{M}}\left(s_j, K(B-1)\right), \tag{11}$$

where B denotes the quantization group size. Each outlier channel is paired with $Blocksize-1$ filler channels to construct a new quantization group:

$$\mathcal{G}_i = [m_i, f_{i,1}, \ldots, f_{i,Blocksize-1}], \tag{12}$$

Accordingly, the reordered channel indices define a global permutation matrix $\mathbf{P}$, which rearranges the original channels into the newly constructed quantization groups.

4.2 MICRO-BLOCK GIVENS TRANSFORM

A Givens rotation is an orthogonal transformation that operates on a two-dimensional subspace. For any pair of channels i and j, its basic form can be written as

$$\mathbf{G}_{i,j}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{13}$$

The rotation only affects the i-th and j-th dimensions while leaving all other dimensions unchanged.

Consider a local quantization group containing an outlier, whose activation can be approximately written as $[A, \epsilon_1, \epsilon_2, \epsilon_3]$, where $A \gg \epsilon_i$. For MXFP4 quantization, an excessively large value A can substantially enlarge the dynamic range of the quantization group and affect the selection of the shared scale. We therefore aim to redistribute the energy of the outlier channel to several low-magnitude channels through an orthogonal transformation. Unlike the Hadamard transform, which spreads the outlier over the entire quantization group, we only spread it over n selected channels and enforce equal outlier magnitudes among them:

$$[A, 0, \ldots, 0] \rightarrow \left[\underbrace{\frac{A}{\sqrt{n}}, \frac{A}{\sqrt{n}}, \ldots, \frac{A}{\sqrt{n}}}_{n \text{ elements}}, 0, \ldots, 0 \right]. \tag{14}$$

Therefore, $A/\sqrt{n}$ is the equal-magnitude energy allocation that satisfies the orthogonality constraint.

To achieve this objective, we apply a sequence of Givens rotations to progressively transfer the energy of the outlier channel to the remaining selected channels. Assume that the outlier channel is moved to the first position after permutation, and another $n-1$ low-magnitude channels are selected as the spreading targets. The overall transformation can be written as

$$\mathbf{G} = \mathbf{G}_{1,2}(\theta_1)\mathbf{G}_{2,3}(\theta_2)\cdots\mathbf{G}_{n-1,n}(\theta_{n-1}). \tag{15}$$

The rotation angles are chosen such that each Givens rotation extracts exactly an amplitude of $1/\sqrt{n}$ into a new channel. Before the k-th rotation, the remaining magnitude is

$$r_k = \sqrt{\frac{n-k+1}{n}}. \tag{16}$$

To generate a new component with magnitude $1/\sqrt{n}$, we require

$$r_k \cos\theta_k = \frac{1}{\sqrt{n}}. \tag{17}$$

Therefore, the rotation angles can be recursively chosen as

$$\cos\theta_k = \frac{1}{\sqrt{n-k+1}}, \qquad \sin\theta_k = \sqrt{\frac{n-k}{n-k+1}}, \qquad k = 1, \ldots, n-1. \tag{18}$$

Each rotation allocates an equal-magnitude component of $1/\sqrt{n}$ to a new channel while passing the remaining energy to the subsequent rotations, eventually distributing the original outlier uniformly over n channels. This construction can be interpreted as a progressive energy-release process. At the k-th rotation, a fixed magnitude $A/\sqrt{n}$ is retained in the current channel, while the remaining energy is propagated to the next channel. After $n-1$ rotations, the energy originally concentrated in a single outlier channel is evenly distributed over n channels while strictly preserving the overall L_2 energy.

The final transform matrix can be denoted as:

$$\mathbf{T} = \mathbf{PR}, \qquad \mathbf{R} = \operatorname{diag}\left(\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_K, \mathbf{H}_{K+1}, \ldots, \mathbf{H}_N\right). \tag{19}$$

Here, the first K outlier blocks are transformed by the proposed *Micro-block Givens Rotation*, while the remaining normal blocks follow the conventional Hadamard transform $\mathbf{H}$.

5 EXPERIMENTS

Models and Datasets. We select the current public LLMs for evaluation, including the LLaMA2 series (Touvron et al., 2023), the LLaMA3 series (Grattafiori et al., 2024), and the Qwen3 series (Yang et al., 2025). We primarily assess the performance of quantized large language models through perplexity (PPL), a widely adopted and stable metric for measuring LLM performance, and use WikiText-2-raw (Merity et al., 2016) datasets. We also conduct experiments on six standard common-sense benchmark tasks: WinoGrande (Sakaguchi et al., 2021), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018). All zero-shot tasks are performed uniformly based on the lm-eval-harness framework(Gao et al., 2021).

Compared Methods. We verify the results of different transforms. We compare with the native format and the Hadamard transform. For the Hadamard transform, we also adopt a block-diagonal transform. Additionally, our method can also be combined with the GPTQ (Frantar et al., 2023) approach. We compare it with existing PTQ methods, including: (1) QuaRot (Ashkboos et al., 2024), which uniformly applies a global Hadamard transform to smooth activations; (2) SmoothQuant (Xiao et al., 2023), which smooths the quantization difficulty by migrating it from activations to weights; (3) SpinQuant (Liu et al., 2024), which optimizes learnable transform matrix. (4) GPTQ (Frantar et al., 2023), which compensate native weights with Hessian. (5) BRQ Shao et al. (2026): which combines GPTQ with block-diagonal Hadamard transform. (5) MR-GPTQ (Egiazarian et al., 2025), which uses GPTQ with block Hadamard rotations, MSE scale optimization, and static activation reordering over the transformed weights.

Implementation Details. Our method is implemented with the PyTorch framework (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2019). Consistent with previous works (Frantar & Alistarh, 2023), the calibration data contains 128 samples randomly selected from the C4 dataset (Raffel et al., 2020). Each sample contains sequences of 2048 tokens. For normal blocks, we follow existing approaches and retain the Hadamard transform unchanged. Since outlier blocks can significantly affect the other elements within the same block, making them highly likely to be quantized to zero, we identify outlier blocks in each layer using a fixed ratio of $10/4096$. All experiments are conducted on NVIDIA RTX4090 48GB GPUs.

5.1 LAYER-WISE ANALYSIS

To demonstrate the effectiveness of our proposed method, we first perform an activation reconstruction error analysis. Figure 3 presents the activation distributions and corresponding quantization errors under three different transformation strategies: no transform, block-wise Hadamard with the block size of 32, and our proposed transformation, evaluated on different types of input activations from the Transformer layers on LLaMA2-7B.

As shown in Figure 3, block-wise Hadamard can effectively reduce the quantization error compared with the case without transformation, indicating that redistributing activation magnitudes before quantization is beneficial. Furthermore, our method achieves lower reconstruction errors than the Hadamard transformation across all evaluated activation types, demonstrating its effectiveness in reducing MXFP4 quantization errors. For activations from the Up/Down projections, the outlier magnitudes are relatively small. As the outlier magnitude increases, the advantage of our method becomes more apparent. For the activation of QKV projections, where outliers reach magnitudes of approximately 10, our method achieves a noticeable reduction in reconstruction error compared with the Hadamard transform. In the case of the activation of the Down projection, which contains massive outliers with magnitudes reaching hundreds or even thousands, the MXFP4 quantization error increases significantly due to the limited representational precision within each micro-block. By adaptively redistributing the outlier energy, our method substantially mitigates this issue and reduces the reconstruction error by several orders of magnitude compared with the Hadamard transform. We further provide an ablation analysis in Appendix B.1 to investigate the contribution of different components of our method to reducing the reconstruction error.

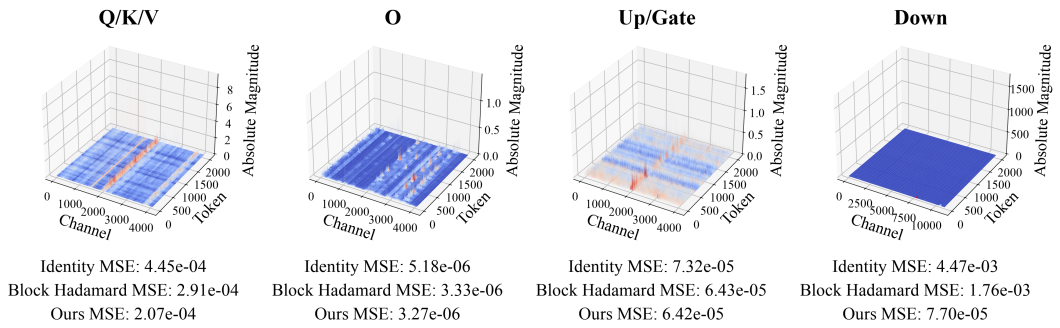


Figure 3: Illustration of activation MXFP4 quantization errors with different transforms. The activations are collected from different modules in the first Transformer block of LLaMA2-7B.The error is computed as $\|\mathbf{X} - \mathcal{Q}(\mathbf{XH})\mathbf{H}^T\|$, where $\mathcal{Q}(\cdot)$ denotes the quantizer and $\mathbf{H}$ denotes the corresponding transform matrix.

Table 1: WikiText PPL (↓) and zero-shot accuracy (↑) for various LLaMA models under W4A4 quantization.

Model	Quantizer	Transform	PPL	Zero-shot Accuracy						
				WinoG.	OBQA	PIQA	HellaS.	ARC-E	ARC-C	*Avg*
2-7B	FP16	–	5.47	69.30	44.00	78.78	76.17	73.78	44.97	64.50
	RTN	w/o	6.20	63.93	**41.20**	76.82	73.13	70.24	**41.98**	61.22
		Hadamard	6.17	66.22	39.40	**77.58**	72.81	69.23	41.04	61.05
		Ours	**6.14**	**66.85**	41.00	76.88	**73.25**	**71.59**	41.72	**61.88**
	GPTQ	w/o	6.13	66.85	40.20	77.20	73.33	69.23	41.72	61.42
		Hadamard	**5.84**	**67.72**	41.80	77.53	**73.83**	72.14	43.94	62.83
		Ours	**5.84**	67.64	**42.20**	**77.97**	73.82	**73.06**	**45.73**	**63.40**
2-13B	FP16	–	4.88	72.53	45.20	80.41	79.65	76.52	48.81	67.19
	RTN	w/o	5.37	69.69	**43.00**	78.84	76.55	73.82	**47.35**	64.88
		Hadamard	**5.30**	**70.96**	42.40	**79.00**	**76.82**	74.03	46.08	64.88
		Ours	5.35	69.69	42.80	78.84	76.72	**74.79**	46.84	**64.95**
	GPTQ	w/o	5.34	69.53	**45.00**	79.38	76.91	73.70	46.25	65.13
		Hadamard	**5.16**	**72.93**	43.60	80.03	77.56	**76.05**	**48.98**	**66.53**
		Ours	5.20	70.32	43.20	**80.09**	**77.62**	75.55	47.10	65.65
3-8B	FP16	–	6.13	73.80	44.80	80.63	79.24	77.57	54.01	68.34
	RTN	w/o	7.84	66.69	43.20	**77.69**	**75.34**	73.15	45.99	63.68
		Hadamard	7.84	**68.35**	42.20	**77.69**	75.19	73.48	**50.77**	**64.62**
		Ours	**7.62**	66.85	**43.40**	77.48	74.47	**75.67**	46.93	64.13
	GPTQ	w/o	7.59	68.35	42.00	76.44	74.31	72.01	45.05	63.03
		Hadamard	**7.10**	69.69	43.00	**78.84**	**76.41**	**76.64**	**51.19**	65.96
		Ours	7.16	**72.14**	**45.00**	78.45	75.92	75.08	49.66	**66.04**

5.2 Main Benchmark Results

Compared with the Hadamard transform. Table 1 compares our methods with no rotation and the conventional Hadamard transform across different LLaMA models. On LLaMA2-7B, it achieves the best average zero-shot accuracy for both RTN and GPTQ, while also obtaining the lowest or tied-lowest WikiText2 PPL. On LLaMA3-8B, our method increases the average accuracy from 63.03 to 66.04 under GPTQ compared with RTN, slightly outperforming Hadamard while maintaining comparable perplexity. These results indicate that the proposed methods generalize well across different model scales and quantizers.

Compared with existing PTQ methods. Table 2 compares our method with representative PTQ methods on Qwen3-8B. Our method achieves a WikiText2 PPL of 10.94, matching the best result among the compared quantized methods. Our method obtains the highest average zero-shot accuracy of 64.85, together with the best performance on PIQA and ARC-C. These results demonstrate that our method provides clear advantages over existing PTQ methods under the MXFP4 format.

Table 2: WikiText PPL (↓) and zero-shot task accuracy (↑) performance for Qwen3-8B by different PTQ methods. The $^+$ denotes combination with GPTQ.

Method	Wiki (↓)	Zero-shot Accuracy (↑)						
		WinoG.	OBQA	PIQA	HellaS.	ARC-E	ARC-C	*Avg*
FP16	7.22	73.48	44.80	81.45	79.53	79.84	55.89	69.16
RTN	11.07	64.48	39.80	74.10	70.45	75.93	52.65	62.90
QuaRot$^+$	12.03	64.71	35.42	75.08	71.54	69.02	44.41	60.03
SpinQuant$^+$	11.05	**69.18**	37.46	75.31	**73.28**	75.07	50.40	63.45
SmoothQuant	11.16	64.96	39.00	73.94	71.05	76.47	50.34	62.63
GPTQ	**10.94**	66.69	40.00	74.70	70.28	77.74	52.90	63.72
BRQ	**10.94**	66.30	39.80	75.08	71.73	77.02	53.75	63.95
MR-GPTQ	**10.94**	67.01	**40.60**	76.39	72.26	**79.08**	53.33	64.78
Ours$^+$	**10.94**	67.96	40.20	**76.71**	71.84	78.37	**54.01**	**64.85**

Table 3: Ablation study of transformation components on Qwen3-8B with RTN quantization.

Inter-Block Permutation	Micro-block Givens Rotation	Wiki (↓)	Avg. Acc. (↑)
✗	✗	11.07	62.90
✓	✗	10.98	63.22
✗	✓	11.29	63.38
✓	✓	10.95	63.05

Table 4: Ablation study on the outlier ratio on Qwen3-8B with RTN quantization.

Outlier Ratio	Wiki (↓)	Avg. Acc. (↑)
5 / 4096	10.94	63.10
10 / 4096	10.98	63.22
15 / 4096	11.16	62.95
20 / 4096	11.23	62.88

5.3 Ablation Study

We first evaluate the effectiveness of the two proposed components on Qwen3-8B with RTN quantization. As shown in Table 3, both components individually improve the downstream average accuracy compared with the baseline, although the changes in WikiText2 perplexity are not strictly monotonic. When the two components are combined, the resulting PPL and average accuracy do not surpass those of the individual components. Nevertheless, as shown in the Appendix B.1, the two modules exhibit a clear complementary effect in reducing the reconstruction error. This suggests that their interaction has not yet been fully translated into downstream performance gains. In future work, we will further investigate how to better coordinate these two modules to achieve more consistent downstream task improvements.

We further study the sensitivity to the outlier ratio in Table 4. A moderate outlier ratio generally provides a better trade-off between WikiText2 perplexity and downstream accuracy, while increasing the ratio does not consistently improve performance. In particular, excessively selecting outlier blocks may introduce unnecessary transformations to blocks that do not benefit from the proposed treatment. Based on these results, we adopt an outlier ratio of (10/4096) as the default setting in our experiments.

6 Conclusion

This paper presents a post-training quantization framework tailored to the unique numerical characteristics of MXFP4. We identify that quantization error is strongly related to the distance between transformed values and favorable representable magnitudes, which we refer to as lossless quantization points. Based on this observation, we propose two complementary components: *Inter-block permutation* that groups outliers with low-importance channels, and *Micro-block Givens Rotation* that adaptively controls the number of participating channels to redistribute outlier energy toward favorable quantization points. Experiments on the LLaMA2/3 and Qwen3 model families demonstrate that our method outperforms the Hadamard transform and other commonly used transformation strategies under MXFP4 quantization. These results highlight the importance of designing format-aware transformations that explicitly exploit the numerical and block-wise properties of microscaling low-precision formats.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and grammatical refinement. They were not involved in method design, experimental design, implementation, data analysis, or scientific conclusions. All methods, experiments, results, and claims were produced and verified by the authors, and the authors take full responsibility for the content of this paper.

REFERENCES

Felix Abecassis, Anjulie Agrusa, Dong Ahn, Jonah Alben, Stefania Alborghetti, Michael Andersch, Sivakumar Arayandi, Alexis Bjorlin, Aaron Blakeman, Evan Briones, et al. Pretraining large language models with nvfp4. arXiv preprint arXiv:2509.25149, 2025.

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

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In NeurIPS, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In AAAI, 2020.

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.

DeepSeekAI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J.L. Cai, and et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434, 2024.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. In NeurIPS, 2022.

Vage Egiazarian, Roberto L Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, et al. Bridging the gap between promise and performance for microscaling fp4 quantization. arXiv preprint arXiv:2509.23202, 2025.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In ICML, 2023.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh Frantar. Optimal brain compression: A framework for accurate post-training quantization and pruning. In NeurIPS, 2022.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. Zenodo, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In ICLR, 2024.

Wonsuk Jang and Thierry Tambe. Blockdialect: Block-wise fine-grained mixed format quantization for energy-efficient llm inference. In ICML, 2025.

Junhan Kim, Ho-young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. Boa: Attention-aware post-training quantization without backpropagation. In ICML, 2025.

Janghwan Lee, Jiwoong Park, Jinseok Kim, Yongjik Kim, Jungju Oh, Jinwook Oh, and Jungwook Choi. Amxfp4: Taming activation outliers with asymmetric microscaling floating-point for 4-bit llm inference. In Findings of ACL, 2025.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. In ICML, 2022.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, WeiMing Chen, WeiChen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. In MLSys, 2024.

Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. Llm-fp4: 4-bit floating-point quantized transformers. In EMNLP, pp. 592–605, 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. arXiv preprint arXiv:2405.16406, 2024.

Yuanyong Luo, Jing Huang, Yu Cheng, Ziwei Yu, Kaihua Tang, Xinda Ma, Xin Wang, Anping Tong, Guipeng Hu, Yun Xu, et al. Hifloat4 format for language model inference. arXiv preprint arXiv:2602.11287, 2026.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. arXiv preprint arXiv:1809.02789, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In NeurIPS, 2019.

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. JMLR, 21(140):1–67, 2020.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, et al. Microscaling data formats for deep learning. arXiv preprint arXiv:2310.10537, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. Communications of the ACM, 64(9):99–106, 2021.

Yuantian Shao, Peisong Wang, Yuanteng Chen, Chang Xu, Zhihui Wei, and Jian Cheng. Block rotation is all you need for mxfp4 quantization. In ICML, 2026.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. arXiv preprint arXiv:2306.11695, 2023.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for llm quantization. In ICML, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. In ICML, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In ICML, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019.

A ALGORITHMS

Algorithm 1 Inter-block Permutation

Input: Calibration activations $\mathbf{X} \in \mathbb{R}^{N \times D}$, weight matrix $\mathbf{W}$, block size B, outlier ratio α
Output: Permutation matrix $\mathbf{P}$
$a_j \leftarrow \max_n |\mathbf{X}_{n,j}|$ ▷ Channel maximum statistics
$s_j \leftarrow \|\mathbf{X}_{:,j}\|_2 \cdot \|\mathbf{W}_{:,j}\|_1$ ▷ Compute channel importance score
$K \leftarrow \max(1, \mathrm{round}(\alpha D))$
$\mathcal{O} \leftarrow \mathrm{TopK}(a_j, K)$
$\mathcal{L} \leftarrow \mathrm{BottomK}(s_j, K(B-1))$
Initialize an empty channel ordering $\mathcal{I}$
for $k = 1$ **to** K **do**
  $o_k \leftarrow \mathcal{O}[k]$
  $\mathcal{L}_k \leftarrow \mathcal{L}[(k-1)(B-1) : k(B-1)]$
  $\mathcal{G}_k \leftarrow [o_k, \mathcal{L}_k]$
  Append $\mathcal{G}_k$ to $\mathcal{I}$
end for
$\mathcal{R} \leftarrow \{1, \ldots, D\} \setminus (\mathcal{O} \cup \mathcal{L})$
Append channels in $\mathcal{R}$ to $\mathcal{I}$ while preserving their original order
Construct permutation matrix $\mathbf{P}$ according to $\mathcal{I}$
return $\mathbf{P}$

Algorithm 2 Micro-block Givens Rotation

Input: Permuted calibration activations $\widetilde{\mathbf{X}} = \mathbf{XP}$, block size B, MXFP4 quantizer $Q(\cdot)$
Output: Block-diagonal transform matrix $\mathbf{R}$
$G \leftarrow D/B$
for each block $g = 0$ **to** $G-1$ **do**
  $\mathbf{X}_g \leftarrow \widetilde{\mathbf{X}}[:, gB : (g+1)B]$
  if $\mathbf{X}_g$ contains an outlier channel **then**
    for $n = 1$ **to** B **do**
      $\widehat{\mathbf{R}}_g^{(n)} \leftarrow \mathbf{I}_B$
      for $k = 1$ **to** $n-1$ **do**
        $r \leftarrow n - k + 1$
        $s \leftarrow 1/\sqrt{r}, \quad c \leftarrow \sqrt{(r-1)/r}$
        Construct $\mathbf{G}(c, s)$ ▷ Eq. 15
        $\widehat{\mathbf{R}}_g^{(n)} \leftarrow \widehat{\mathbf{R}}_g^{(n)} \mathbf{G}(c, s)$
      end for
      $\widehat{\mathbf{X}}_g^{(n)} \leftarrow \mathbf{X}_g \widehat{\mathbf{R}}_g^{(n)}$
      $\overline{\mathbf{X}}_g^{(n)} \leftarrow Q\left(\widehat{\mathbf{X}}_g^{(n)}\right) \widehat{\mathbf{R}}_g^{(n)T}$
      $E(n) \leftarrow \left\| \mathbf{X}_g - \overline{\mathbf{X}}_g^{(n)} \right\|_F^2$
    end for
    $n^\star \leftarrow \arg\min_{1 \le n \le B} E(n)$
    $\mathbf{R}_g \leftarrow \widehat{\mathbf{R}}_g^{(n^\star)}$
  else
    $\mathbf{R}_g \leftarrow \mathbf{H}_B / \sqrt{B}$
  end if
end for
return $\mathbf{R} \leftarrow \mathrm{diag}(\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_{G-1})$

B More Analyses

B.1 Details of Layer-wise Error

To more comprehensively evaluate the effectiveness of the proposed components, we conduct a layer-wise analysis, since our method directly transforms the activations of each layer. Table 5 reports the ablation results of the proposed components on activations collected from different types of Transformer layers in LLaMA2-7B. We first examine the effect of applying the Hadamard transform to different blocks. For outlier-containing blocks, Hadamard provides the largest reconstruction-error reduction for $Down$ activations, while the improvement for $Q/K/V$ is limited. In contrast, for normal blocks, Hadamard benefits $Q/K/V$ the most and $Down$ the least, indicating that its effectiveness strongly depends on both activation type and the presence of extreme outliers.

Replacing Hadamard in outlier blocks with our proposed *Micro-block Givens Rotation* yields modest changes for the activation from O and $Up/Gate$, but provides clearer improvements for $Q/K/V$ and especially $Down$, where the reconstruction error is reduced by approximately two orders of magnitude. This demonstrates the effectiveness of the proposed micro-block transformation for blocks containing massive outliers.

We further evaluate the proposed *Inter-block Permutation*. Permutation alone has little effect on $Down$, slightly improves O and $Up/Gate$, and substantially reduces the error of $Q/K/V$. Combining permutation with Hadamard further reduces the error, e.g., by about 25% for the O projection compared with Hadamard alone. Finally, applying Micro-block Givens to outlier blocks after permutation achieves the lowest reconstruction errors across all evaluated activation types, demonstrating the complementary effects of the two proposed components.

Table 5: Layer-wise ablation study of reconstruction error under different strategies.

Inter-block Permutation (ours)	Outliers Blocks			Normal Blocks		Q/K/V	O	Up/Gate	Down
	Micro-block Givens (ours)	Hadamard	Identity	Hadamard	Identity				
			✓		✓	4.45e-4	5.18e-6	7.33e-5	4.47e-3
		✓			✓	4.12e-4	4.42e-6	6.87e-5	1.74e-3
		✓		✓		2.91e-4	3.34e-6	6.43e-5	1.74e-3
	✓			✓		2.23e-4	3.59e-6	6.46e-5	7.61e-5
✓			✓		✓	3.93e-4	5.12e-6	7.32e-5	4.47e-3
✓		✓		✓		2.83e-4	3.30e-6	6.44e-5	1.74e-3
✓	✓			✓		2.08e-4	3.29e-6	6.43e-5	7.61e-5

We alse explore the influence to reconstructu error by outlier ratio using the activation collected in the *QKV* projection. As shown in Table 6, the reconstruction error generally decreases as the outlier ratio increases, indicating that applying the proposed transformation to more outlier blocks can further reduce quantization-induced reconstruction error. However, reconstruction error alone may not fully reflect the importance of individual channels, since some channels with relatively small magnitudes may still carry important information for downstream prediction. Therefore, rather than aggressively increasing the outlier ratio, we adopt a conservative ratio of $10/4096$ in our experiments. In future work, we will explore an adaptive mechanism that jointly considers reconstruction error and channel importance to determine the outlier ratio for each layer.

Table 6: Ablation study of QKV activation quantization error under different outlier ratios.

Inter-Block Permutation	Micro-block Givens Rotation	Outlier Ratio			
		5/4096	10/4096	15/4096	20/4096
✗	✗	5.53e-4			
✓	✗	4.95e-4	4.74e-4	4.64e-4	4.62e-4
✗	✓	1.33e-4	1.04e-4	9.11e-5	8.46e-5
✓	✓	1.14e-4	8.25e-5	7.18e-5	6.40e-5

B.2 Standard MXFP4 Lossless Points

Figure 4 illustrates the quantization error of standard MXFP4 when the input value serves as the block-wise maximum. Standard MXFP4 does not employ the additional $3/4$ scaling factor. Its E2M1 FP4 data format has a maximum representable codeword of 6, whereas the power-of-two shared scale can result in a normalized dynamic range extending to 8. Consequently, values falling between 6 and 8 cannot be represented directly and are clipped to the largest codeword. Once the block scale increases to the next power of two, however, these values are rescaled back into the representable FP4 range, leading to an abrupt change in quantization error around the power-of-two scale boundaries. In particular, values slightly below such a boundary may be mapped close to 8 after normalization and thus suffer from clipping, whereas slightly larger values can trigger a larger shared scale and consequently incur a smaller quantization error.

Despite this characteristic discontinuity, the quantization error of standard MXFP4 remains highly non-monotonic with respect to the input magnitude. Therefore, simply increasing or decreasing the transformed magnitude does not guarantee a lower error. This further motivates our MSE-based search over the number of participating channels n, which directly selects the micro-block transform that minimizes the resulting reconstruction error.

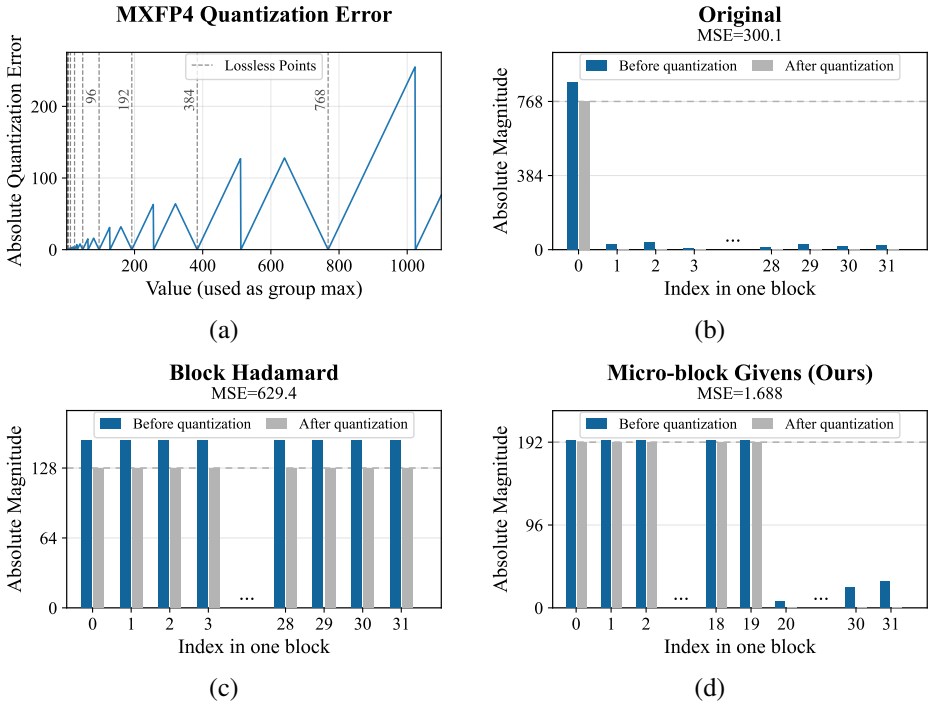
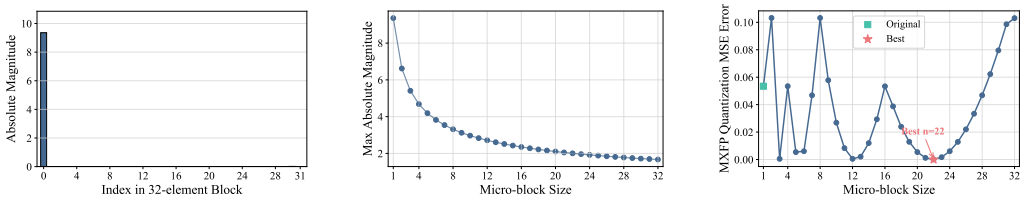


Figure 4: Standard MXFP4 quantization error and transformed weight distribution.

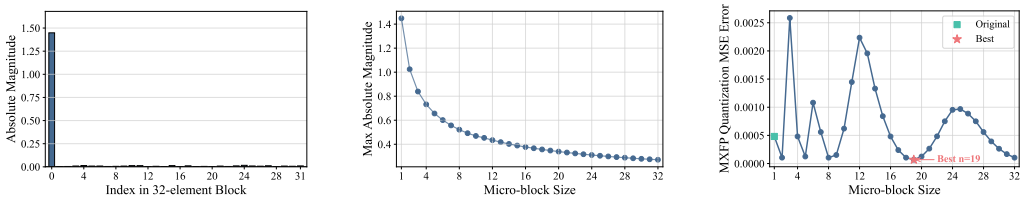
B.3 Micro-block Distribution

To further investigate the effect of the micro-block transform size, we select the 32-element block containing the largest-magnitude activation after applying inter-block permutation, and compare the MXFP4 errors under different micro-block transform sizes. The results are shown in Figure 5- 8. As the number of participating channels n increases, the transformed activation magnitudes within the block generally decrease. However, the reconstruction error does not decrease monotonically with n. Although an overall downward trend can be observed, the error exhibits substantial fluctuations across different transform sizes. In particular, spreading the outlier over all 32 elements in the block is not always optimal and can even lead to a noticeable increase in reconstruction error in some cases. These results demonstrate the necessity of adaptively selecting an appropriate micro-block size rather than uniformly distributing the outlier energy across the entire block.



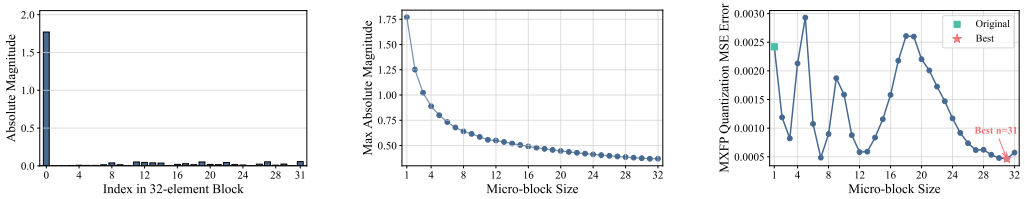
(a) Original block element (b) Transformed element maximum (c) MXFP4 error

Figure 5: The amplitude distribution within the block containing the maximum-magnitude element in the activation of *Q/K/V*.



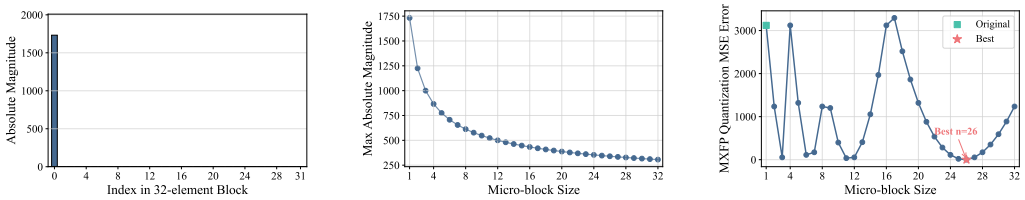
(a) Original block element (b) Transformed element maximum (c) MXFP4 error

Figure 6: The amplitude distribution within the block containing the maximum-magnitude element in the activation of *O*.



(a) Original block element (b) Transformed element maximum (c) MXFP4 error

Figure 7: The amplitude distribution within the block containing the maximum-magnitude element in the activation of *Up/Gate*.



(a) Original block element (b) Transformed element maximum (c) MXFP4 error

Figure 8: The amplitude distribution within the block containing the maximum-magnitude element in the activation of *Down*.